

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống quản lý dữ liệu đa sàn thương mại điện tử

NGUYỄN XUÂN KHOA

Khoa.nx152019@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Nguyễn Mạnh Tuấn

Chữ ký của GVHD

Bộ môn:

Công nghệ phần mềm

Viện:

Công nghệ thông tin và Truyền thông

HÀ NỘI, 6/2020

Lời cam kết

Họ và tên sinh viên: Nguyễn Xuân Khoa

Điện thoại liên lạc: 0975275297

Email: khoa1997hn@gmail.com

Lớp: IS1 – K60

Hệ đào tạo: CNTT Việt Nhật

Tôi – *Nguyễn Xuân Khoa* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Nguyễn Mạnh Tuấn*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 01 tháng 06 năm 2020

Tác giả ĐATN

Nguyễn Xuân Khoa

Lời cảm ơn

Đến được những ngày cuối cùng trong 5 năm học ở Đại học Bách Khoa Hà Nội, em đã nhận được sự quan tâm giúp đỡ từ rất nhiều người, em muốn gửi lời cảm ơn tới tất cả mọi người. Lời đầu tiên, em xin gửi lời cảm ơn chân thành tới các thầy cô giáo trong trường Đại học Bách Khoa Hà Nội nói chung cùng các thầy cô trong Viện Công nghệ Thông tin và Truyền thông và các thầy cô bộ môn tiếng Nhật đã truyền dạy cho em những kiến thức bổ ích, những bài học về cách sống và cách làm người, lắng nghe những chia sẻ và tư vấn cho em trong những lúc khó khăn, tạo dựng cho em nền tảng kiến thức vững chắc, đây chính là hành trang quan trọng trong sự nghiệp và trong cuộc sống của em suốt những năm về sau.

Em xin gửi lời cảm ơn đặc biệt tới giảng viên hướng dẫn, ThS. Nguyễn Mạnh Tuấn - Giảng viên Viện CNTT&TT Đại học Bách Khoa Hà Nội. Thầy đã hướng dẫn và giúp đỡ em khi mới bắt đầu thực hiện nghiên cứu những đề án đầu tiên và đặc biệt trong đề án tốt nghiệp. Chính nhờ sự giúp đỡ và định hướng tận tình của thầy về những vấn đề em gặp phải cũng như đánh giá về những tính năng đã làm mà em đã tìm ra được hướng đi để hoàn thiện đề án tốt nhất có thể.

Cuối cùng, em xin gửi lời cảm ơn tới gia đình, người thân và bạn bè đã luôn bên cạnh, giúp đỡ em vượt quá khó khăn học hành và bệnh tật. Gia đình luôn là chỗ dựa tinh thần vững chắc. Còn bạn bè là những người tuyệt vời nhất luôn bên em khi vui khi buồn, là những người bên nhau đa số thời gian từ học hành, ăn uống và giải trí, tình bạn chân thành giúp chúng ta cùng nhau vượt qua tất cả. Nếu không có các bạn mình sẽ không thể có được ngày hôm nay cả về tinh thần và kiến thức. Chúng ta đã có một thanh xuân tuyệt vời ở Đại học Bách Khoa Hà Nội. Một lần nữa xin cảm ơn tất cả mọi người với sự chân thành nhất.

Tóm tắt nội dung đồ án

Ngày nay việc bán hàng trên các sàn thương mại điện tử ngày càng trở nên phổ biến. Cùng với đó số lượng nhà bán hàng cũng tăng lên rất nhiều. Có rất nhiều nhà bán hàng sở hữu nhiều gian hàng trên cùng một sàn thương mại điện tử hoặc trên nhiều sàn thương mại điện tử. Việc quản lý thông số bán hàng trên cùng lúc nhiều gian hàng như vậy vô cùng khó khăn. Để giải quyết vấn đề này, một số công ty đã đưa ra giải pháp, tiêu biểu là Powersell, nhanh.vn, Sapo, nhưng mỗi sản phẩm đều có những hạn chế riêng. Một trong số những điểm hạn chế là không quản lý được giá nhập sản phẩm, chi phí đóng gói, không đồng bộ giá bán, không có các cảnh báo đơn hàng khi đơn hàng không được vận chuyển hoặc thanh toán, không thể quét mã vận chuyển để trừ hàng trong kho, không thể kết nối sản phẩm dạng combo ... Những điểm hạn chế đó hầu hết đều do các công ty này ở nước ngoài, không khảo sát kỹ thị trường Việt Nam và những nhà bán hàng Việt Nam.

Để giải quyết những vấn đề trên, em đã khảo sát rất nhiều nhà bán hàng lớn trong quá trình xây dựng hệ thống để hệ thống gần gũi với người Việt Nam và chi phí sử dụng thấp hơn các hệ thống khác của nước ngoài. Hệ thống sẽ cho phép người dùng nhập danh sách sản phẩm trong kho lên hệ thống, rồi kết nối các sản phẩm trong kho với các sản phẩm trên các sàn thương mại điện tử để đồng bộ việc tồn kho và tính toán các thông số về việc bán hàng, đơn hàng trên các gian hàng cũng được đồng bộ về hệ thống để người dùng dễ dàng quản lý. Hệ thống được xây dựng dưới dạng một trang web PHP/Laravel dành cho nhà bán hàng sử dụng. Ngoài ra hệ thống còn có một server xây dựng bằng Node.js liên tục lấy dữ liệu về từ các sàn thương mại điện tử phục vụ cho việc tổng hợp thông tin của hệ thống.

Hệ thống sẽ giúp người bán tăng năng suất bán hàng, dễ dàng có được số liệu chính xác được về việc kinh doanh trên các gian hàng thương mại điện tử. Từ đó sẽ đem lại cho các nhà bán hàng một giải pháp quản lý việc bán hàng trên đa sàn thương mại hiệu quả, phù hợp với môi trường thương mại điện tử của Việt Nam, tiết kiệm thời gian và chi phí thấp.

Sinh viên thực hiện

Ký và ghi rõ họ tên

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án.....	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU	4
2.1 Khảo sát hiện trạng.....	4
2.2 Tổng quan chức năng	6
2.2.1 Biểu đồ use case tổng quan	6
2.2.2 Biểu đồ phân rã chức năng báo cáo	9
2.2.3 Biểu đồ use case phân rã chức năng Đơn hàng.....	10
2.2.4 Biểu đồ use case phân rã chức năng Kho hàng.....	11
2.2.5 Biểu đồ use case phân rã chức năng Tài khoản	12
2.2.6 Quy trình nghiệp vụ	12
2.3 Đặc tả chức năng.....	14
2.3.1 Đặc tả use case “Kết nối sản phẩm trong kho”	14
2.3.2 Đặc tả use case “Hoàn hàng/Xuất kho”	16
2.3.3 Đặc tả use case “Đồng bộ giá”	17
2.3.4 Đặc tả use case “Cảnh báo vận chuyển”	18
2.4 Yêu cầu phi chức năng.....	19
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG	20
3.1 Công nghệ thu thập dữ liệu từ website.....	20
3.1.1 Kỹ thuật thu thập dữ liệu từ API.....	20
3.1.2 Server thu thập dữ liệu sử dụng NodeJS.....	20
3.2 Công nghệ lập trình web	21
3.2.1 Công nghệ HTML, CSS, Javascript.....	21
3.2.2 Laravel Framework	22
3.2.3 Hệ quản trị cơ sở dữ liệu MySQL.....	22
3.2.4 Công nghệ RESTful API.....	23
CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG	24
4.1 Thiết kế kiến trúc	24
4.1.1 Lựa chọn kiến trúc phần mềm MVC	24

4.1.2	Thiết kế tổng quan.....	25
4.1.3	Thiết kế chi tiết gói	26
4.2	Thiết kế chi tiết.....	27
4.2.1	Thiết kế giao diện.....	27
4.2.2	Thiết kế lớp	30
4.2.3	Thiết kế cơ sở dữ liệu.....	31
4.3	Xây dựng ứng dụng.....	35
4.3.1	Thư viện và công cụ sử dụng	35
4.3.2	Kết quả đạt được	35
4.3.3	Minh họa các chức năng chính	36
4.4	Kiểm thử.....	39
4.5	Triển khai	41
	CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	42
5.1	Khó khăn khi thu thập dữ liệu từ các sàn TMDT	42
5.1.1	Vấn đề gặp phải.....	42
5.1.2	Giải pháp đưa ra.....	42
5.1.3	Kết quả đạt được	46
5.2	Chuyển đổi luật của sàn TMDT thành cảnh báo vận chuyển	47
5.2.1	Vấn đề gặp phải.....	47
5.2.2	Giải pháp đưa ra	48
5.2.3	Kết quả đạt được	48
5.3	Kết nối sản phẩm trong kho hàng và sản phẩm trên sàn và tính toán lượng hàng tồn kho.	48
5.3.1	Vấn đề gặp phải.....	48
5.3.2	Giải pháp đưa ra	49
5.3.3	Kết quả đạt được	49
5.4	Tối giản tính năng Hoàn hàng/Xuất kho	49
5.4.1	Vấn đề gặp phải.....	49
5.4.2	Giải pháp đưa ra	50
5.4.3	Kết quả đạt được	50
	CHƯƠNG 6. KẾT LUẬN VÀ PHÁT TRIỂN	51
6.1	Kết luận	51
6.2	Hướng phát triển	52
	TÀI LIỆU THAM KHẢO	54

PHỤ LỤC..... 55

DANH MỤC HÌNH VẼ

Hình 2.1	Giao diện cơ bản của Powersell	4
Hình 2.2	Giao diện cơ bản của Sapo	5
Hình 2.3	Biểu đồ use case tổng quan	7
Hình 2.4	Biểu đồ use case phân rã chức năng báo cáo	9
Hình 2.5	Biểu đồ use case phân rã chức năng đơn hàng	10
Hình 2.6	Biểu đồ use case phân rã chức năng kho hàng	11
Hình 2.7	Biểu đồ use case phân rã chức năng tài khoản	12
Hình 2.8	Sơ đồ quy trình nghiệp vụ kết nối sản phẩm	13
Hình 2.9	Đặc tả use case kết nối sản phẩm trong kho	15
Hình 2.10	Đặc tả use case hoàn hàng/xuất kho	17
Hình 3.1	Ví dụ về API danh sách sản phẩm của sàn Tiki	20
Hình 3.2	Màn hình console thông báo việc lấy dữ liệu từ Lazada hoàn tất	21
Hình 4.1	Mô hình MVC trong Laravel	25
Hình 4.2	Biểu đồ tổng quan gói	25
Hình 4.3	Biểu đồ gói chi tiết cho package Model	26
Hình 4.4	Biểu đồ gói chi tiết cho package Controller	26
Hình 4.5	Biểu đồ gói chi tiết cho package View	27
Hình 4.6	Thiết kế giao diện màn hình báo cáo	28
Hình 4.7	Thiết kế giao diện màn hình Hoàn hàng/Xuất kho	28
Hình 4.8	Thiết kế giao diện màn hình Đơn hàng	29
Hình 4.9	Thiết kế giao diện màn hình Kho hàng	29
Hình 4.10	Thiết kế giao diện màn hình Thêm kết nối	30
Hình 4.11	Thiết kế lớp của use case kết nối sản phẩm	30
Hình 4.12	Thiết kế lớp của use case cảnh báo vận chuyển	31
Hình 4.13	Thiết kế lớp của use case hoàn hàng/xuất kho	31
Hình 4.14	Mô hình thực thể liên kết cơ sở dữ liệu hệ thống	32
Hình 4.15	Màn hình báo cáo kinh doanh	36
Hình 4.16	Màn hình hoàn hàng/xuất kho	37
Hình 4.17	Màn hình danh sách đơn hàng	37
Hình 4.18	Màn hình kho hàng	38
Hình 4.19	Màn hình thêm kết nối	38
Hình 4.20	Màn hình danh sách sản phẩm trên sàn	39
Hình 4.21	Sơ đồ triển khai của máy chủ AWS	41
Hình 5.1	Cách lấy API của Tiki (1)	43
Hình 5.2	Cách lấy API của Tiki (2)	43
Hình 5.3	Cách lấy API của Tiki (3)	44

Hình 5.4 Cách lấy API của Tiki (4).....	44
Hình 5.5 Cách lấy API của Tiki (5).....	45
Hình 5.6 Cách lấy API của Tiki (6).....	45
Hình 5.7 Lấy thành công danh sách sản phẩm bằng Postman	46
Hình 5.8 Đồng bộ trạng thái đơn hàng giữa sàn Tiki và hệ thống	47
Hình 5.9 Mô tả quy trình thay đổi trạng thái nội bộ.....	50
Hình 6.1 Giao diện báo cáo kinh doanh mới.....	53
Hình 6.2 Giao diện Hoàn hàng/Xuất kho mới.....	53

DANH MỤC BẢNG

Bảng 2.1	So sánh các sản phẩm đang có với đồ án	5
Bảng 2.2	Mô tả tóm tắt chức năng cho người dùng	8
Bảng 2.3	Mô tả chức năng use case báo cáo	9
Bảng 2.4	Mô tả chức năng use case đơn hàng.....	10
Bảng 2.5	Mô tả chức năng use case chức năng kho hàng	11
Bảng 2.6	Mô tả chức năng use case tài khoản.....	12
Bảng 2.7	Mô tả tóm tắt use case kết nối sản phẩm trong kho	14
Bảng 2.8	Mô tả tóm tắt use case hoàn hàng/xuất kho	16
Bảng 2.9	Mô tả tóm tắt use case đồng bộ giá.....	17
Bảng 2.10	Mô tả tóm tắt use case cảnh báo vận chuyển	18
Bảng 3.1	Các phương thức chính của RESTful.....	23
Bảng 4.1	Mô tả các bảng trong CSDL.....	33
Bảng 4.2	Danh sách công cụ, thư viện quan trọng	35
Bảng 4.3	Kết quả đóng gói sản phẩm.....	36
Bảng 4.4	Mô tả kịch bản kiểm thử	39
Bảng 4.5	Kết quả kiểm thử	41

DANH MỤC CÁC TỪ VIẾT TẮT

API	Application Programming Interface Giao diện lập trình ứng dụng
HTML	HyperText Markup Language Ngôn ngữ đánh dấu siêu văn bản
B2C	Business to Customer
TMDT	Thương mại điện tử
CNTT	Công nghệ thông tin
ĐATN	Đồ án tốt nghiệp
SV	Sinh viên
AWS	Amazon Web Services

DANH MỤC THUẬT NGỮ

Browser	Trình duyệt
Cache memory	Bộ nhớ đệm
E-commerce	Thương mại điện tử
Shipper	Người vận chuyển

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Trong chương đầu của đồ án tốt nghiệp, em sẽ giới thiệu về đề tài Quản lý dữ liệu đa sàn thương mại điện tử. Đầu tiên sẽ là vấn đề dẫn đến việc em thực hiện đề tài này, sau đó sẽ là những mục tiêu em nhắm tới, để giải quyết những mục tiêu đó em sẽ đặt ra giải pháp và cuối cùng là bố cục các chương tiếp theo trong đồ án.

1.1 Đặt vấn đề

Thương mại điện tử ở Việt Nam đã xuất hiện từ rất lâu, từ những năm 2003 đã bắt đầu manh nha với những website như vatgia, 5giay, enbac, ... Và bắt đầu hoạt động quy củ hơn từ năm 2012 khi Lazada đặt chân vào Việt Nam. Qua thời gian phát triển, việc mua hàng trực tuyến từ việc bị người dùng nghi ngờ, e dè khi mua hàng dần nhận được sự tin tưởng. Tới nay việc bán hàng trên các nền tảng thương mại điện tử đã thật sự bùng nổ với các trang thương mại điện tử như Lazada, Tiki, Shopee và Sendo.

Để tiếp cận với nhiều khách hàng hơn, người bán sẽ phải bán trên nhiều trang thương mại điện tử hơn. Và để phục vụ việc đa dạng hóa mặt hàng, có tính đặc trưng cho từng gian hàng thì một người bán có thể sở hữu nhiều tài khoản bán hàng trên cùng một nền tảng thương mại điện tử.

Điều này dẫn đến những khó khăn về việc quản lý các tài khoản. Và khiến cho những người bán hàng có nhu cầu về một hệ thống giúp việc quản lý nhiều tài khoản bán hàng trở nên đơn giản, chính xác, tốn ít thời gian hơn. Khi nhu cầu này được đáp ứng sẽ đem lại những lợi thế về kinh doanh như biết được sản phẩm đang bán chạy, biết được lượng hàng trong kho để nhập hàng, kiểm tra đơn hàng dễ dàng và chính xác. Ngoài ra còn có thể cảnh báo người bán hàng để tránh vi phạm vào những điều luật của các trang thương mại điện tử như chậm giao hàng cho shipper, hàng hết trong kho nhưng vẫn nhận đơn hàng, ... giúp họ giảm thiểu số tiền bị phạt, và tạo sự uy tín với người mua hàng.

1.2 Mục tiêu và phạm vi đề tài

Nắm bắt được những nhu cầu trên, một số công ty nước ngoài đã đưa ra các hệ thống về đồng bộ dữ liệu đa sàn TMDT dành cho những nhà bán hàng B2C. Những hệ thống đó cơ bản đã đáp ứng được nhu cầu của nhà bán hàng, bao gồm các tính năng: kết nối sản phẩm ở trên sàn và trong kho, quản lý đơn hàng, đưa ra các thống kê, ... Tuy nhiên do không phải sản phẩm của Việt Nam, nên các sản phẩm đã có đều được xây dựng theo những nhu cầu của người nước ngoài, dẫn tới việc chưa phù hợp với người dùng Việt Nam và giá thành sử dụng còn cao. Một trong những hạn chế lớn là việc dữ liệu tồn kho sẽ được tính theo trạng thái đơn hàng trên sàn, nếu đơn hàng ở trạng thái vận chuyển sẽ trừ số lượng sản phẩm tồn kho, dẫn đến việc khi trạng thái đơn hàng chưa cập nhật kịp hoặc khi hàng đã đóng gói đang giao cho shipper thì số lượng tồn kho sẽ bị sai.

Vậy nên, em đã tiến hành khảo sát và xây dựng hệ thống dựa trên những nhu cầu thực tế của các nhà bán hàng Việt Nam để hệ thống đạt được những mục tiêu sau:

Thứ nhất, xây dựng quy trình đồng bộ dữ liệu đơn hàng và kho hàng sao cho dễ dàng quản lý đơn hàng dù đơn hàng từ nhiều nguồn khác nhau, kho hàng được quản lý chính xác, có lịch sử nhập kho và bắt buộc ghi chú mỗi khi nhập kho.

Thứ hai, có đầy đủ các cảnh báo khi nhà bán hàng có nguy cơ vi phạm điều lệ của các sàn TMDT dẫn đến bị phạt tiền. Vì mỗi sàn có một hệ thống quy định riêng, cần quy chuẩn các quy định đó và đưa ra cảnh báo phù hợp, chính xác.

Thứ ba, khi sản phẩm thương mại hóa sẽ có mức chi phí sử dụng rẻ hơn các sản phẩm khác đã có mặt trên thị trường để dễ dàng tiếp cận tới những nhà bán hàng nhỏ lẻ.

Thứ tư, có những điểm khác biệt và cải tiến những tính năng cho phù hợp với yêu cầu từng nhà bán hàng.

Đối tượng người dùng là những người có bán hàng trên nhiều sàn thương mại điện tử hoặc có nhiều gian hàng trên cùng một sàn thương mại điện tử, những người bán nhỏ không có đủ khả năng để sử dụng những hệ thống lớn, hoặc cảm thấy khó dùng và khó hiểu với những hệ thống lớn.

1.3 Định hướng giải pháp

Từ những vấn đề đặt ra ở trên, em đã xây dựng hệ thống đồng bộ dữ liệu đa sàn TMDT. Ứng với từng mục tiêu đã đặt ra ở trên, em sẽ mô tả phương án như sau:

Thứ nhất, Quy trình đồng bộ dữ liệu thông qua API hiển thị thông tin của các sàn TMDT và đưa dữ liệu lên các sàn TMDT thông qua API cập nhật dữ liệu. Dữ liệu được lấy về sẽ được lọc ra lấy những dữ liệu quan trọng để lưu vào cơ sở dữ liệu, từ đó thống kê ra cho người sử dụng. Có các tính năng kết nối sản phẩm giữa kho hàng và sản phẩm trên sàn TMDT, có lưu lại lịch sử nhập kho để dễ dàng tra cứu. Có trang báo cáo thông số bán hàng cho người sử dụng để dễ dàng quản lý một cách tổng quan.

Thứ hai, tìm hiểu các luật của các sàn TMDT với nhà bán hàng, từ đó đưa ra mức thời gian cảnh báo chung cho tất cả các sàn để cảnh báo nhà bán hàng khi có những đơn có nguy cơ bị phạt. Cảnh báo thanh toán với những đơn khách hàng đã nhận hàng nhưng nhà bán hàng chưa nhận được tiền.

Thứ ba, hệ thống được định hướng thương mại theo cách bán hoàn toàn cho nhà bán hàng theo gói 3 năm thay vì cho thuê hàng tháng, với chi phí tính ra 1 tháng rẻ hơn các hệ thống đã có trên thị trường.

Thứ tư, khảo sát các nhà bán hàng xem xét những tính năng cần thiết được đưa vào hệ thống. Tiến tới việc mỗi người dùng sẽ được lựa chọn những tính năng hay yêu cầu họ muốn và hệ thống của mỗi người sẽ khác nhau.

Hệ thống sẽ bao gồm hai phần: phần Nodejs làm nhiệm vụ lấy dữ liệu đơn hàng liên tục và lưu vào database, phần PHP dùng framework Laravel dùng để đồng bộ dữ liệu sản phẩm và hiển thị các dữ liệu sau tính toán. Frontend sử dụng template AdminLTE 2.

Quy trình thực hiện được khảo sát qua các nhà bán hàng diễn ra tại Hà Nội. Các nhà bán hàng sẽ được dùng thử . Từ đó đưa ra những nhận xét để xây dựng, cải tiến hệ thống phù hợp với người Việt Nam. Kết quả khảo sát và dùng thử được trình bày trong Phụ lục.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được em tổ chức như sau.

Chương 2 em trình bày về Khảo sát và phân tích yêu cầu, đây là quá trình tìm hiểu vấn đề, xây dựng phương án đồng bộ, khảo sát ý kiến của các nhà bán hàng và kịch bản sử dụng từ đó trình bày biểu đồ use case và các kịch bản sử dụng. Sau đó em sẽ chọn ra những use case đặc trưng để phân tích và nói về các yêu cầu phi nghiệp vụ

Trong chương 3, em nói về công nghệ sử dụng, giới thiệu và phân tích các công nghệ nền tảng được sử dụng trong đồ án từ những công nghệ đó sẽ giải quyết được những vấn đề gì đã đặt ra cho đồ án này.

Chương 4, đây là chương quan trọng mà em sẽ trình bày về quá trình tạo nên ứng dụng, phân tích thiết kế SQL, bản vẽ thiết kế giao diện, kịch bản và kết quả kiểm thử, hướng triển khai hệ thống tới người dùng cuối.

Chương 5 là những khó khăn gặp phải xuyên suốt quá trình xây dựng hệ thống và phương án xử lý khó khăn đó, đồng thời cũng nói về những đóng góp nổi bật của đồ án đã đạt được.

Các vấn đề về định hướng phát triển của hệ thống và kết luận sẽ được trình bày kỹ ở chương 6.

Chương 1 này em đã giới thiệu sơ lược về đồ án, những vấn đề đặt ra, những gì đồ án sẽ giải quyết và phương án thực hiện nó, tiếp theo em sẽ làm rõ những vấn đề đã đặt ra ở chương 1 trong các chương 2.

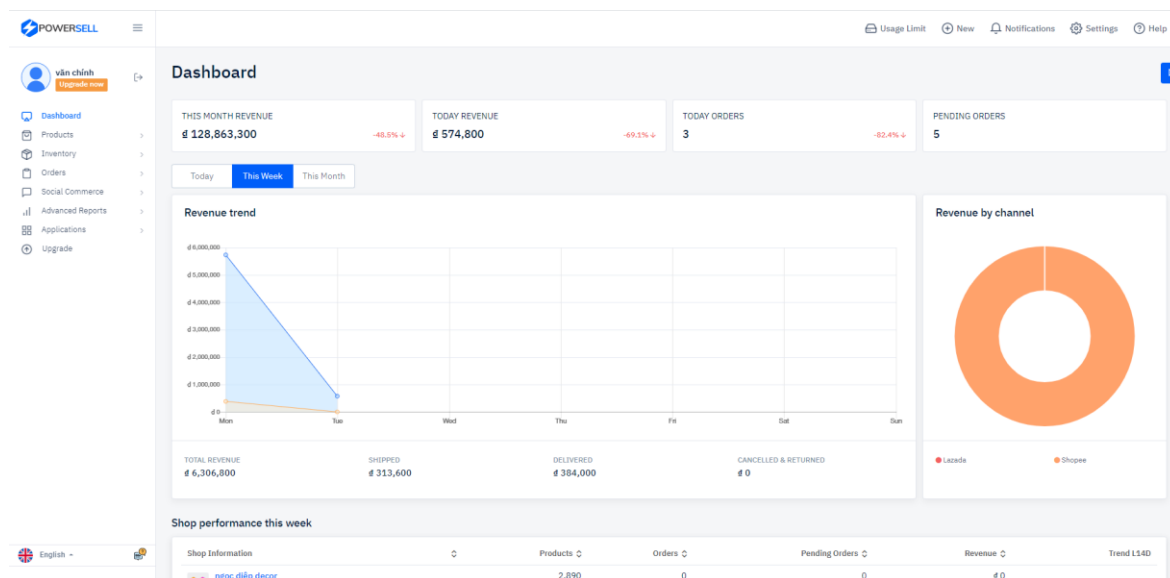
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Trước khi đi vào xây dựng và triển khai hệ thống trong đề án này em đã tìm hiểu và khảo sát các hệ thống cũng như ứng dụng có định hướng phần nào tương đồng với những gì em sẽ thực hiện trong đề án, đồng thời tham khảo từ những người sử dụng thực tế từ đó đưa ra những chức năng quan trọng cần có trong đề án này. Dựa trên những chức năng cần thiết em sẽ đi sâu vào phát triển các tính năng cũng như nghiệp vụ liên quan để tạo thành một hệ thống hoàn chỉnh. Dựa vào đó, trong chương 2 này em sẽ đưa ra những đặc tả về chức năng và các biểu đồ use case mô tả những chức năng quan trọng nhất của hệ thống.

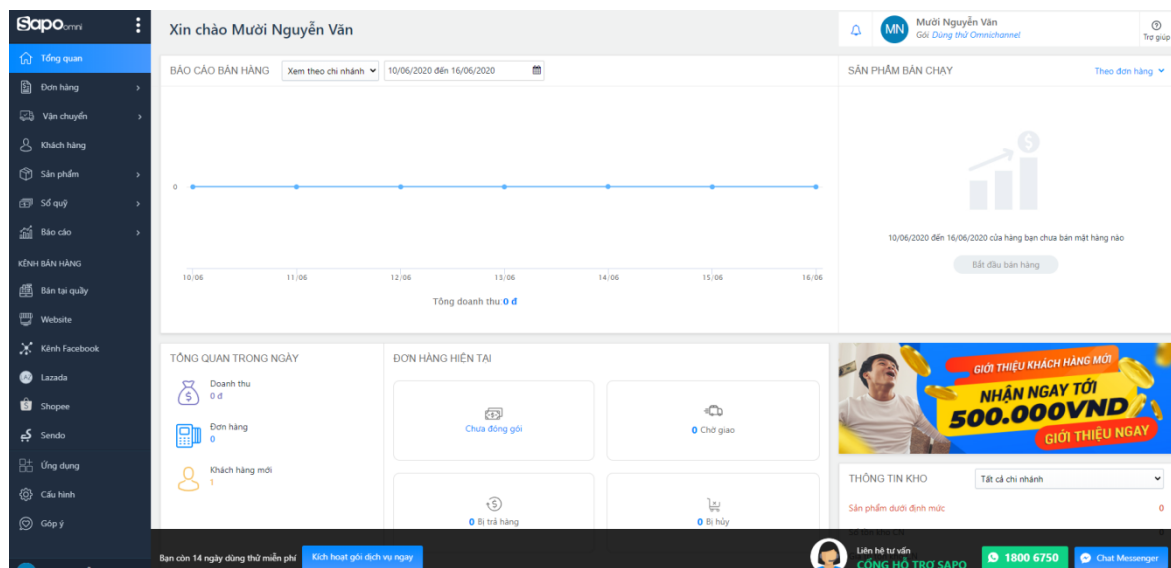
2.1 Khảo sát hiện trạng

Với việc một người sử dụng nhiều sàn TMDT sẽ rất khó khăn trong việc quản lý thông tin từ tất cả các sàn, vì mỗi sàn có một trang quản trị riêng với tính năng riêng và cách sử dụng riêng. Ví dụ số tiền chưa rút, ở Tiki sẽ là sao kê, còn ở Sendo sẽ nằm ở ví senpay. Việc phạt các đơn hàng sai quy định cũng khác nhau ở mỗi sàn. Nắm bắt nhu cầu đó, nhằm tạo ra một hệ thống giúp các nhà bán hàng dễ dàng quản lý, nâng cao hiệu suất bán hàng, các hệ thống quản lý dữ liệu đa sàn TMDT đã ra đời.

Để hiểu rõ về mục tiêu và phạm vi sử dụng của hệ thống, em đã tiến hành khảo sát các sản phẩm có trước có tính năng đồng bộ dữ liệu đa sàn TMDT và các nhà bán hàng. Từ đó giúp em đưa ra quyết định những tính năng nào quan trọng cần có trong sản phẩm của mình. Ở Việt Nam có ba sản phẩm chính là Powersell, Nhanh.vn và Sapo.



Hình 2.1 Giao diện cơ bản của Powersell



Hình 2.2 Giao diện cơ bản của Sapo

Dưới đây là bảng so sánh các nền tảng đang có với đồ án:

Bảng 2.1 So sánh các sản phẩm đang có với đồ án

	Powersell	Nhanh.vn	Sapo	Đồ án
Tính năng nổi bật	Đồng bộ dữ liệu kho hàng, đơn hàng. Báo cáo thông số bán hàng. Xử lý đơn hàng trực tiếp.	Đồng bộ dữ liệu kho hàng, đơn hàng. Báo cáo thông số bán hàng. Xử lý đơn hàng trực tiếp. Copy gian hàng.	Đồng bộ dữ liệu kho hàng, đơn hàng. Báo cáo thông số bán hàng. Quản lý phiếu chi phiếu thu.	Đồng bộ dữ liệu kho hàng, đơn hàng. Báo cáo thông số bán hàng. Cảnh báo vi phạm.
Ưu điểm	Hệ thống có nhiều tính năng nhỏ và rất nhiều loại thông số cho người dùng tham khảo. Cập nhật tính năng mới liên tục.	Có thể đồng bộ việc bán hàng trên cả các nền tảng cũ như vatgia, copy nhanh toàn bộ 1 gian hàng.	Có thể đồng bộ cả việc bán hàng trên Facebook, Zalo, Web và bán hàng offline. Có đội ngũ hỗ trợ nhanh chóng.	Có khả năng quét/nhập mã vận đơn để quản lý việc vận chuyển hàng hóa và tồn kho.
Nhược điểm	Việc kết nối thông qua API được các trang TMDT cung cấp hạn chế điều này	Rất khó để đăng ký sử dụng, muốn sử dụng phải có mã giới thiệu	Không có khả năng kết nối với Tiki. Việc kết nối thông qua API được	Hệ thống còn thô sơ và chưa có khả năng mở rộng nhanh. Phụ

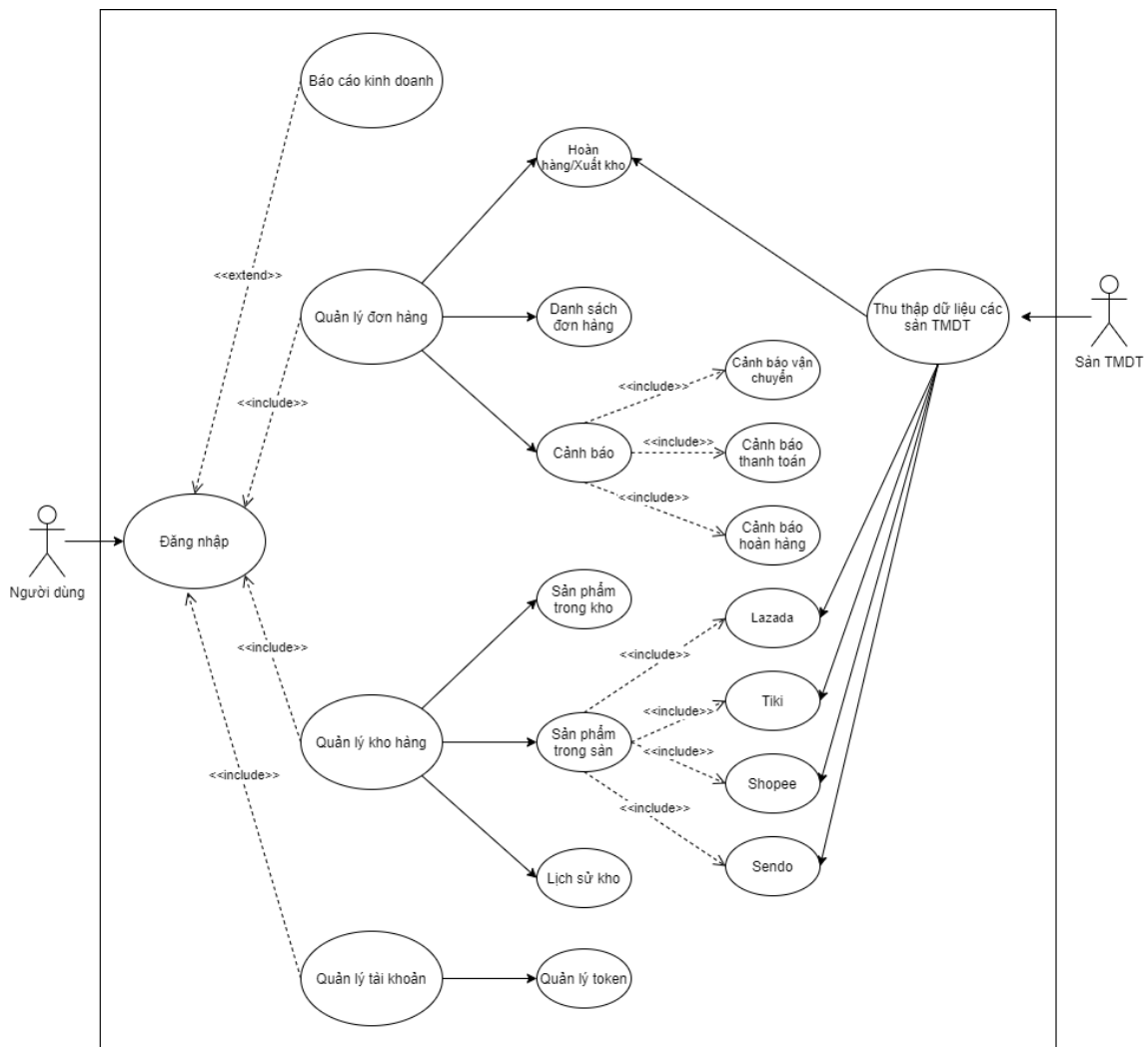
	và làm việc kết nối khó khăn với người dùng khi người dùng phải thực hiện rất nhiều thao tác. Có quá nhiều loại thông số mà người dùng không cần dùng đến.	của người khác. Không có các cảnh báo cho người sử dụng	các trang TMDT cung cấp hạn chế điều này và làm việc kết nối khó khăn với người dùng khi người dùng phải thực hiện rất nhiều thao tác.	thuộc vào việc API của các sản phẩm có thay đổi không. Nếu có thì sẽ mất thời gian cập nhật lại API.
Mức giá	Dùng thử miễn phí kết nối 2 gian hàng. 319.000đ/tháng cho 5 gian hàng. Có giới hạn số lượng sản phẩm và đơn hàng được dùng 1 tháng.	100.000đ/gian hàng/tháng	119.000đ/tháng cho gói cơ bản. 229.000đ/tháng cho gói nâng cao. Có giới hạn số gian hàng được kết nối.	3.0000.000 đồng /3 năm. 100.000 đồng/lần cập nhật API Không giới hạn số gian hàng được kết nối.

Từ việc khảo sát người dùng và các hệ thống đã có em đánh giá các tính năng mà hệ thống trong đề án tốt nghiệp này cần giải quyết sẽ bao gồm: đồng bộ dữ liệu đơn hàng, kết nối sản phẩm trong kho và các sản phẩm, cập nhật dữ liệu tồn kho, cảnh báo đơn hàng với nhà bán hàng và báo cáo kinh doanh. Từ đó em xây dựng biểu đồ use case cho hoạt động của hệ thống.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quan

Với các chức năng đã được đề xuất ở chương 1, em biểu diễn các chức năng chính cho hệ thống sẽ thực hiện trong đề án một cách tổng quan với biểu đồ use case như hình vẽ dưới đây.



Hình 2.3 Biểu đồ use case tổng quan

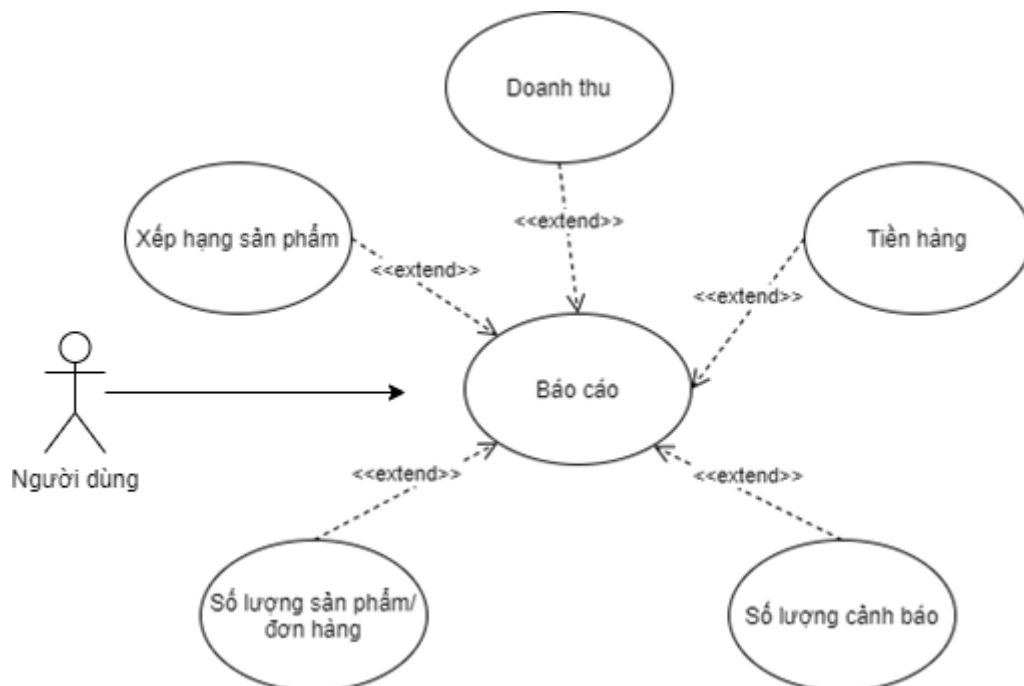
Trong biểu đồ use case ở trên ta có thể thấy, trong hệ thống này sẽ có hai tác nhân chính là người dùng với vai trò là chủ quản lý các gian hàng sử dụng các tính năng, tiện ích được tạo ra bởi hệ thống và các sàn TMDT với vai trò cung cấp dữ liệu cho hệ thống. Người dùng sẽ tương tác trực tiếp với các use case lớn như báo cáo kinh doanh, quản lý đơn hàng, quản lý kho hàng và quản lý tài khoản. Còn các sàn TMDT sẽ cung cấp dữ liệu cho các tính năng như danh sách đơn hàng, sản phẩm trong các sàn, ở đồ án này, dữ liệu mới chỉ đi một chiều từ các sàn TMDT và chưa đi ngược lại lên các sàn TMDT.

Đây là hệ thống hoạt động đơn người dùng, trong đó chỉ có 1 người sử dụng quản lý và sử dụng tất cả thông tin từ các sàn TMDT, với đặc trưng về yếu tố đơn người dùng thì thông tin đăng nhập vào các sàn TMDT sẽ giúp hệ thống lấy được thông tin riêng biệt của từng người dùng và mỗi người sẽ sử dụng một hệ thống riêng trên một server riêng. Vì các chức năng được tạo ra phục vụ cho người dùng, còn sàn TMDT chỉ cung cấp dữ liệu nên sau đây em xin mô tả về các chức năng cho người dùng.

Bảng 2.2 Mô tả tóm tắt chức năng cho người dùng

Tên use case	Mô tả tóm tắt chức năng
Báo cáo	Người dùng sau khi đăng nhập sẽ thấy ngay báo cáo kinh doanh tại màn hình báo cáo. Tại đây có thể xem số cảnh báo đơn hàng (đơn hàng chưa thanh toán, chưa xác nhận, chưa vận chuyển ...), xem doanh thu (theo sản phẩm, xếp hạng doanh thu, doanh thu tổng, ...), xem danh sách sản phẩm bán chạy. Có thể xem số tiền chưa rút (số dư ví) của các tài khoản trên các sàn.
Đơn hàng	Ở mục đơn hàng sẽ quản lý tất cả những vấn đề liên quan đến đơn hàng. Người dùng có thể xem danh sách đơn hàng, duyệt đơn hàng đang chờ, in phiếu vận đơn để dán lên hộp hàng. Tính năng hoàn hàng/xuất kho để người dùng nhập mã đơn vào, sẽ thay đổi trạng thái của đơn hàng thành xuất kho hay hoàn hàng, từ đó tính toán lượng hàng còn lại trong kho và cập nhật ngược lên các sàn. Tính năng cảnh báo sẽ thống kê các đơn hàng đang bị cảnh báo và cần xử lý.
Kho hàng	Ở mục kho hàng sẽ quản lý tất cả vấn đề liên quan tới sản phẩm. Sản phẩm trong kho là tất cả những sản phẩm thực tế có trong kho của người bán, được thống kê bao gồm giá, chi phí đóng gói, và số lượng, có thể thêm – cập nhật sản phẩm trực tiếp hoặc nhập vào từ file Excel từ mẫu có sẵn. Người dùng chọn 1 sản phẩm trong kho và liên kết nó tới các sản phẩm tương ứng trong các sàn thương mại điện tử, 1 sản phẩm trong kho có thể liên kết tới nhiều sản phẩm trên cùng 1 sàn. Sản phẩm trong sàn là danh sách các sản phẩm trên từng sàn thương mại điện tử. Lịch sử kho sẽ thể hiện lịch sử xuất – nhập kho.
Tài khoản	Ở mục tài khoản – quản lý token. Người dùng sẽ nhập username và password của từng sàn vào, hệ thống sẽ giả đăng nhập tài khoản đó với sàn và lấy những thông tin về đơn hàng, sản phẩm, số dư ví.

2.2.2 Biểu đồ phân rã chức năng báo cáo



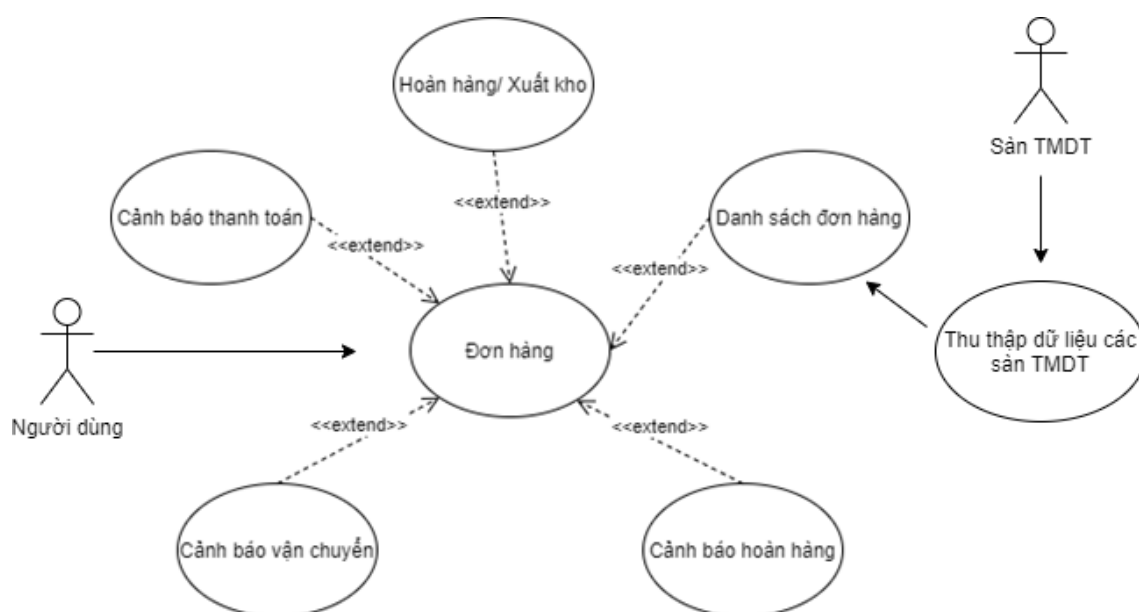
Hình 2.4 Biểu đồ use case phân rã chức năng báo cáo

Ở use case này, hệ thống cần phải xử lý việc hiển thị thông số báo cáo. Trong đó chú ý tới số lượng đơn hàng cần xử lý, số lượng này chỉ tính trong phạm vi 1 tuần gần đây, vì những đơn chưa xử lý thì sau 1 tuần ở tất cả các sản phẩm đều thành bị hủy. Tiền hàng trong kho là tổng giá trị nhập kho nhập kho của lượng hàng trong kho. Khi người dùng lọc ngày tháng thì chỉ các thông số về doanh thu mới thay đổi. Doanh thu lý thuyết là khi có đơn sẽ tính, doanh thu thực tế là khi có đơn được vận chuyển sẽ tính còn doanh thu hiệu suất là doanh thu thực tế trừ đi doanh thu những đơn phải hoàn về.

Bảng 2.3 Mô tả chức năng use case báo cáo

Tên use case	Mô tả tóm tắt tính năng
Doanh thu	Người dùng có thể xem xếp hạng doanh thu từng gian hàng, doanh thu theo từng sản, doanh thu tổng, doanh thu lý thuyết, doanh thu thực tế.
Tiền hàng	Người dùng có thể xem tiền hàng trong kho, tiền hàng chưa thanh toán, tiền trong các ví trên các sản.
Xếp hạng sản phẩm	Xếp hạng sản phẩm bán chạy theo doanh thu.
Số lượng cảnh báo	Thông báo số lượng của từng loại cảnh báo, nhấn vào có thể xem chi tiết các cảnh báo của đơn hàng nào.
Số lượng đơn hàng, sản phẩm	Thông báo số lượng sản phẩm đang có, số đơn hàng đang đã hoàn thành hoặc đang chờ duyệt.

2.2.3 Biểu đồ use case phân rã chức năng Đơn hàng



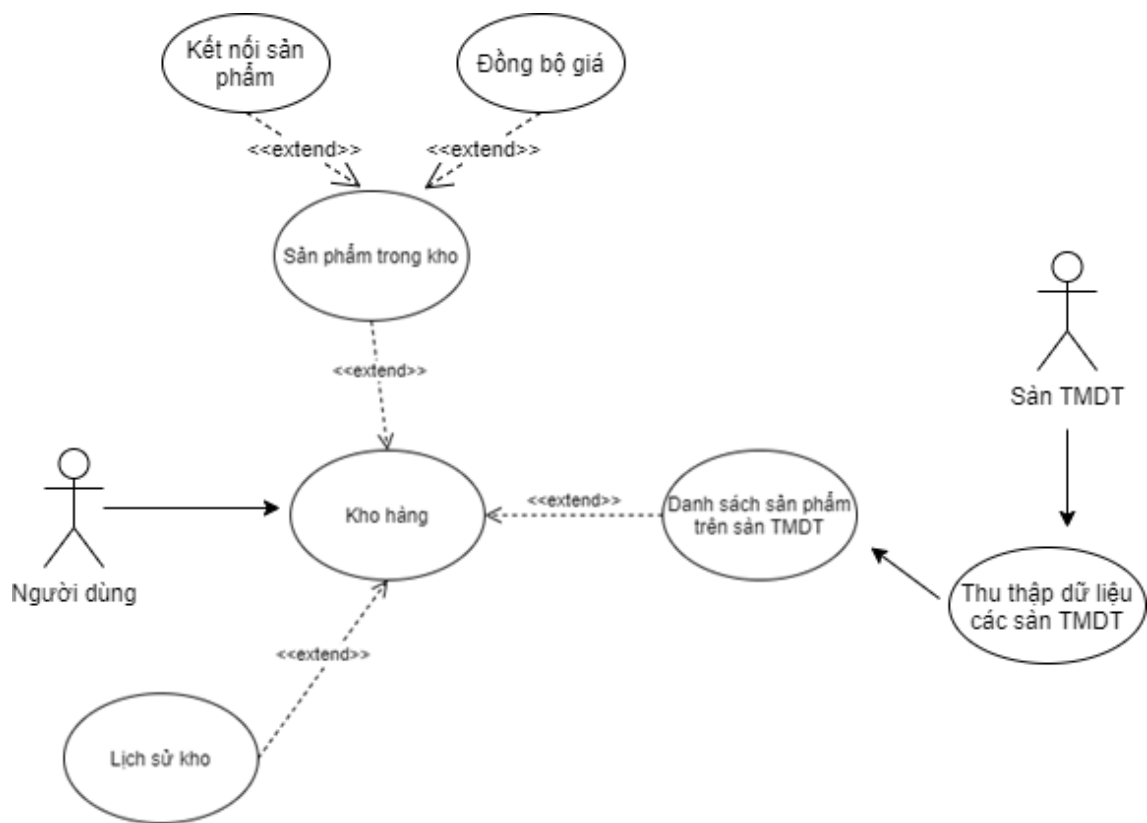
Hình 2.5 Biểu đồ use case phân rã chức năng đơn hàng

Use case này cần chú ý tới các loại cảnh báo vì mỗi sản phẩm có một luật riêng cần đồng nhất các luật đó và đưa ra cảnh báo. Tính năng hoàn hàng/xuất kho cần phân biệt giữa quét mã vạch của sản phẩm hoàn hàng tạm thời do người dùng không có nhà hay là hoàn hàng do hủy đơn. Nếu đơn hàng ở trạng thái hoàn hàng hoặc hủy thì tính là hàng được hoàn về kho, còn lại thì sẽ tính là hàng giao không thành công chỉ hoàn tạm thời.

Bảng 2.4 Mô tả chức năng use case đơn hàng

Tên use case	Mô tả tóm tắt tính năng
Hoàn hàng/Xuất kho	Người dùng chọn chế độ hoàn hàng hay xuất kho cho đơn hàng, nhập mã vận đơn để chuyển trạng thái của hàng hóa trong đơn hàng đó.
Danh sách đơn hàng	Danh sách tất cả đơn hàng của mọi gian hàng được tổng hợp lại ở màn hình này. Có thể xem chi tiết nội dung đơn hàng
Cảnh báo hoàn hàng	Danh sách các đơn hàng bị hoàn hàng về.
Cảnh báo vận chuyển	Danh sách các đơn hàng gần đến thời hạn cuối để chuyển hàng cho shipper.
Cảnh báo thanh toán	Danh sách các đơn hàng khách hàng đã nhận hàng nhưng vẫn chưa nhận được thanh toán.

2.2.4 Biểu đồ use case phân rã chức năng Kho hàng



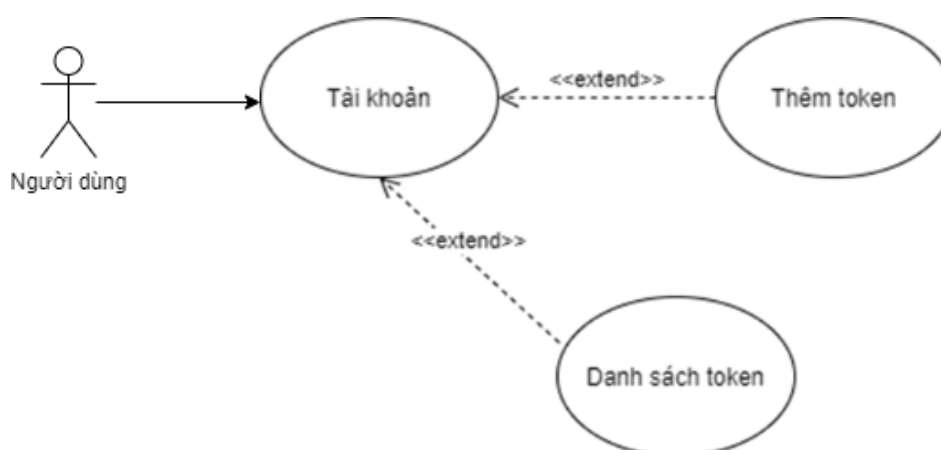
Hình 2.6 Biểu đồ use case phân rã chức năng kho hàng

Use case này cần chú ý tới việc khi lấy dữ liệu sản phẩm từ các sàn về có những sản phẩm có nhiều thuộc tính khác nhau. Lúc này cần tách mỗi sản phẩm với thuộc tính khác nhau thành 1 sản phẩm riêng biệt dựa vào mã SKU.

Bảng 2.5 Mô tả chức năng use case chức năng kho hàng

Tên use case	Mô tả tóm tắt tính năng
Sản phẩm trong kho	Danh sách các sản phẩm có trong kho hàng của người dùng. Kết nối các sản phẩm trong kho với các sản phẩm trên sàn. Tạo thêm sản phẩm mới và cập nhật số lượng tồn kho.
Danh sách sản phẩm trên các sàn TMDT	Danh sách các sản phẩm theo từng sàn TMDT được lấy về từ tài khoản của người dùng.
Lịch sử kho	Danh sách thay đổi tồn kho của người dùng.
Đồng bộ giá	Cập nhật giá của các sản phẩm trên sàn đã được kết nối với sản phẩm trong kho.
Kết nối sản phẩm	Kết nối giữa sản phẩm trên sàn với sản phẩm trong kho

2.2.5 Biểu đồ use case phân rã chức năng Tài khoản



Hình 2.7 Biểu đồ use case phân rã chức năng tài khoản

Use case này người dùng sẽ quản lý danh sách các tài khoản dùng để hệ thống đăng nhập vào các sàn TMDT và lấy dữ liệu từ đó.

Bảng 2.6 Mô tả chức năng use case tài khoản

Tên use case	Mô tả tóm tắt tính năng
Thêm token	Người dùng nhập tên đăng nhập, mật khẩu, chọn sàn TMDT vào hệ thống.
Danh sách token	Hiển thị danh sách token đã được nhập vào hệ thống

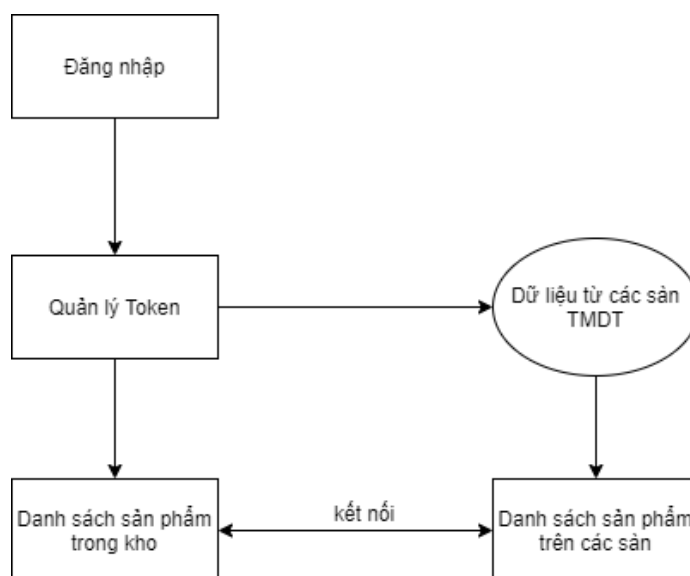
2.2.6 Quy trình nghiệp vụ

Sau đây em sẽ trình bày về quy trình kết nối sản phẩm giữa sàn TMDT với kho.

Người dùng đăng nhập hệ thống, sau đó vào phần Quản lý Token sẽ nhập tên đăng nhập và tài khoản của sàn TMDT, chọn sàn TMDT, lặp lại thao tác đó nếu có nhiều tài khoản. Dữ liệu sau đó sẽ bắt đầu được lấy về và hiển thị vào phần danh sách sản phẩm tương ứng với từng sàn Tiki, Shopee, Lazada, Sendo.

Người dùng vào Kho, tạo sản phẩm mới thủ công hoặc tải file excel về và nhập liệu vào theo mẫu rồi upload lên hệ thống, sản phẩm sẽ được nhập vào kho. Sau đó người dùng chọn sản phẩm cần đồng bộ. Một màn hình mới hiện ra, người dùng chọn những sản phẩm trên sàn tương ứng và số lượng sản phẩm trong kho ứng với sản phẩm trên sàn. Ví dụ 1: sản phẩm trong kho là A, sản phẩm trên sàn là 1 gói 5A, người dùng vào sản phẩm A trong kho chọn 5A và nhập số lượng là 5. Ví dụ 2: sản phẩm trong kho là A và B, sản phẩm trên sàn là gói AB, người dùng vào sản phẩm A trong kho, chọn AB, số lượng là 1 rồi vào sản phẩm B trong kho, chọn AB, chọn số lượng là 1. Lặp lại thao tác khi có nhiều gian hàng trong cùng 1 tài khoản.

Sau đó sản phẩm trong kho và trên sàn sẽ được kết nối với nhau. Khi có đơn hàng mới ở sàn và được quét mã thì sản phẩm trong kho sẽ bị trừ tương ứng.



Hình 2.8 Sơ đồ quy trình nghiệp vụ kết nối sản phẩm

Quy trình quét mã vận đơn của tính năng hoàn hàng/xuất kho như sau. Người dùng khi có đơn hàng mới từ sàn TMDT, đơn hàng sẽ ở trạng thái Chờ xử lý, sau khi duyệt đơn hàng, đơn hàng sẽ chuyển sang trạng thái Đang xử lý, người dùng chọn xuất kho và quét mã vận đơn của đơn hàng, và giao hàng cho shipper hoặc gửi tới kho của nhà vận chuyển, đơn hàng được nhà vận chuyển chuyển sang trạng thái vận chuyển, số lượng hàng tồn kho sẽ trừ đi dựa theo đơn hàng xuất kho. Nếu đơn hàng giao không thành công, hàng được trả lại kho, người dùng chọn hoàn hàng và quét mã vận đơn của đơn hàng lúc này số lượng hàng hóa sẽ cộng thêm vào từ đơn hàng được hoàn.

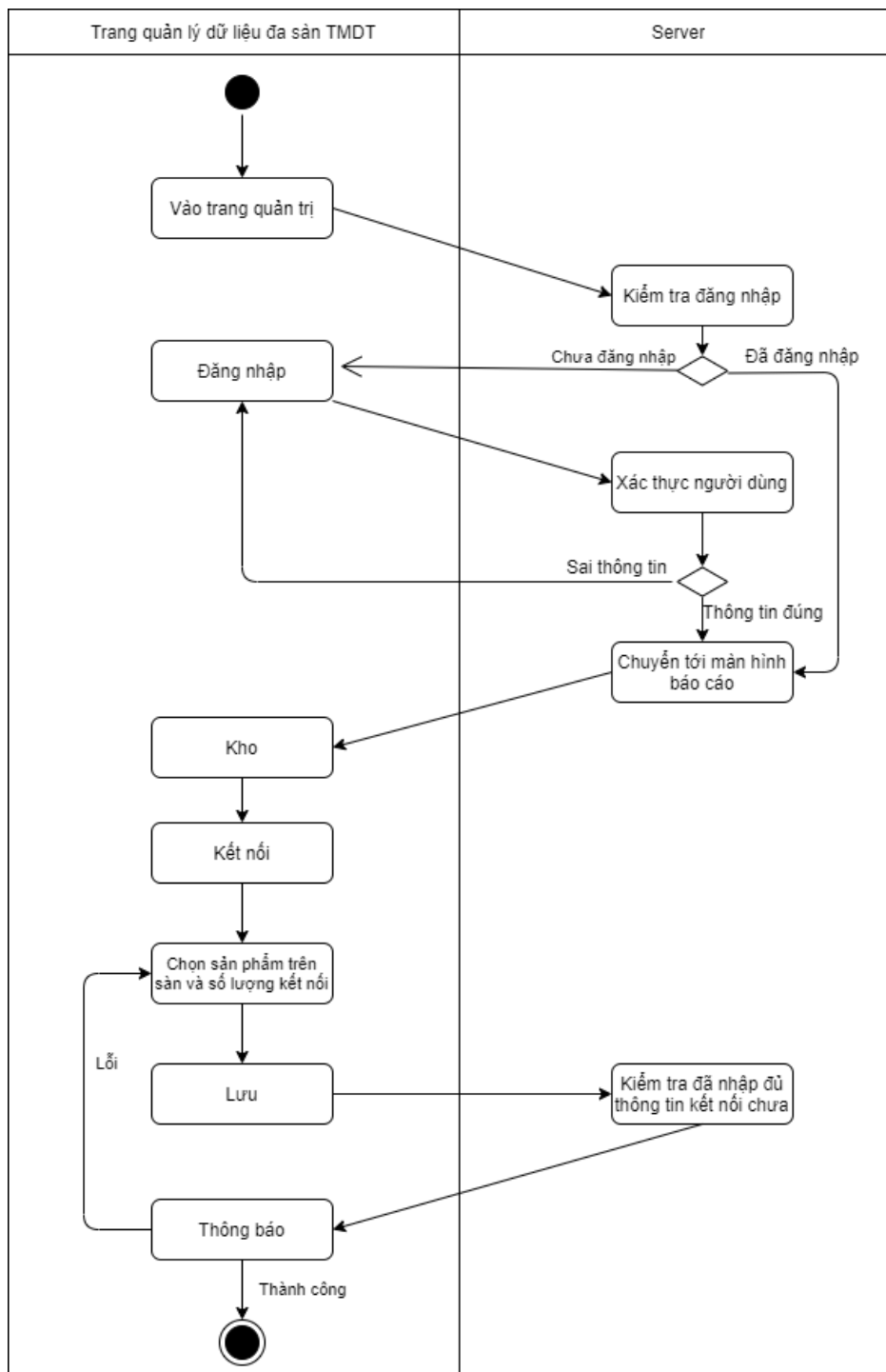
2.3 Đặc tả chức năng

Trong phần này, em sẽ chọn ra những use case là kết nối sản phẩm trong kho và hoàn hàng/xuất kho, đồng bộ giá và cảnh báo vận chuyển, đó là những chức năng chính, và mô tả cụ thể hơn về các chức năng này.

2.3.1 Đặc tả use case “Kết nối sản phẩm trong kho”

Bảng 2.7 Mô tả tóm tắt use case kết nối sản phẩm trong kho

Tên use case	Kết nối sản phẩm trong kho
ID	UC_001
Mô tả	Use case này sẽ giúp người dùng kết nối sản phẩm từ trong kho với các sản phẩm trên sàn để đồng bộ số lượng tồn kho.
Các tác nhân tham gia	Người dùng
Điều kiện đầu vào	Yêu cầu đã đăng nhập Yêu cầu đã nhập tài khoản sàn TMDT vào quản lý token. Yêu cầu sản phẩm trên sàn đã được đồng bộ về mới nhất. Yêu cầu đã có sản phẩm trong kho.
Kết quả đầu ra	Các sản phẩm trong kho và sàn liên kết nối với nhau.
Luồng sự kiện chính	1. Người dùng truy cập vào màn hình quản lý kho. 2. Người dùng chọn sản phẩm cần kết nối và vào màn hình kết nối sản phẩm 3. Người dùng chọn những sản phẩm trên từng sàn và số lượng sản phẩm trong kho ứng với nó. 4. Hệ thống lưu lại kết nối.
Luồng sự kiện khác	3b. Nếu sản phẩm trên sàn và kho đã kết nối sẽ hiển thị ở khu vực đã kết nối.
Tần suất sử dụng	Cao

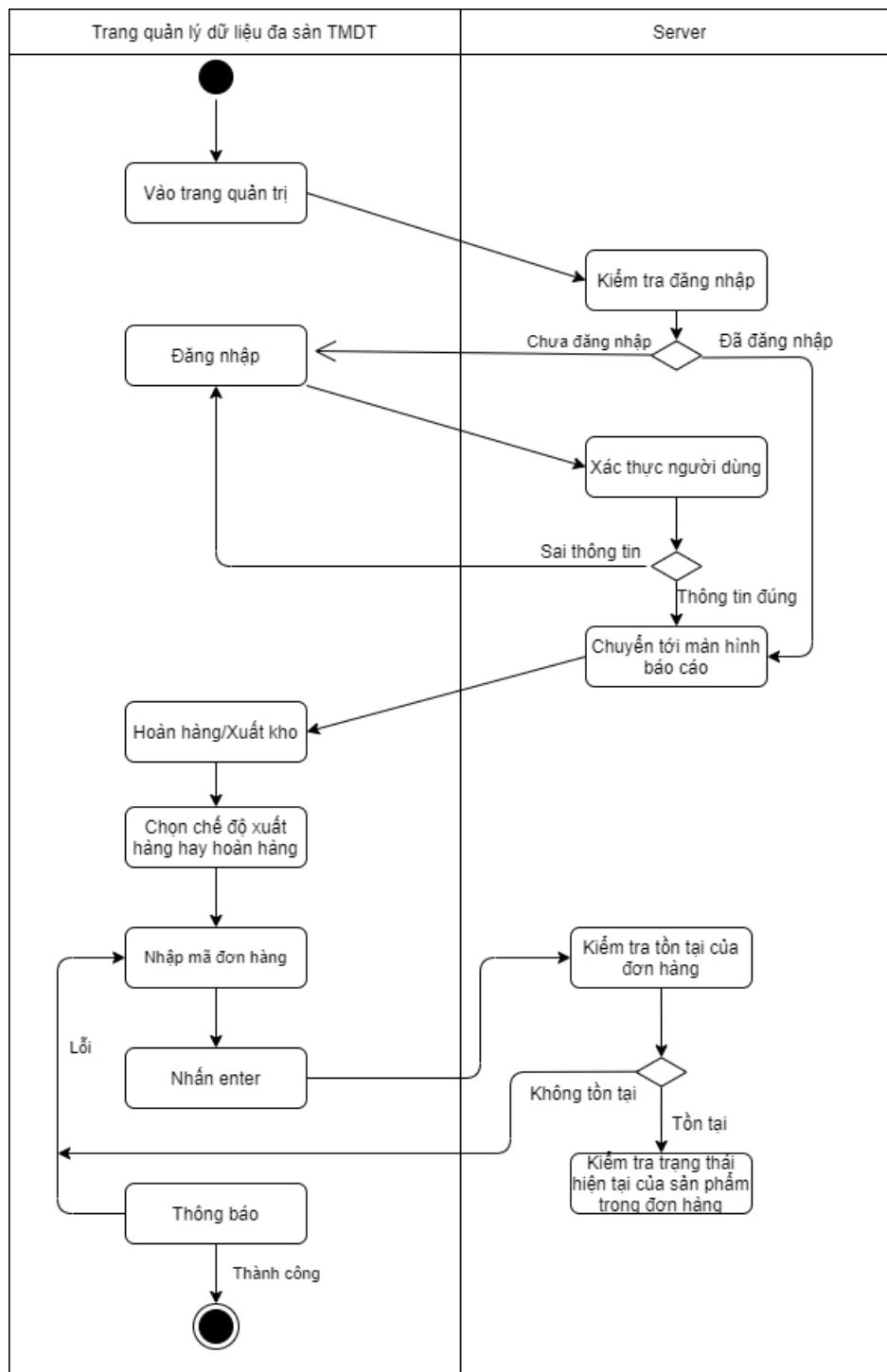


Hình 2.9 Đặc tả use case kết nối sản phẩm trong kho

2.3.2 Đặc tả use case “Hoàn hàng/Xuất kho”

Bảng 2.8 Mô tả tóm tắt use case hoàn hàng/xuất kho

Tên use case	Hoàn hàng/Xuất kho
ID	UC_002
Mô tả	Use case này để người dùng chuyển trạng nội bộ của thái đơn hàng, lấy mã vận đơn của đơn hàng, chọn trạng thái thay đổi của đơn, nhập mã vận đơn vào bằng bàn phím hoặc máy quét mã vạch. Sản phẩm trong đơn hàng sẽ được chuyển trạng thái.
Các tác nhân tham gia	Người dùng
Điều kiện đầu vào	Yêu cầu đã đăng nhập Yêu cầu đã nhập tài khoản sàn TMDT vào quản lý token. Yêu cầu đã có đơn hàng.
Kết quả đầu ra	Trạng thái nội bộ của đơn hàng được thay đổi. Tồn kho của sản phẩm thay đổi.
Luồng sự kiện chính	1. Người dùng truy cập vào màn hình hoàn hàng xuất kho 2. Dùng bàn phím hoặc máy quét mã vạch để nhập mã vận đơn của đơn hàng, chọn chế độ hoàn hàng hoặc xuất hàng. 3. Trạng thái đơn hàng thay đổi, thông báo cho người dùng. 4. Số lượng tồn kho thay đổi.
Luồng sự kiện khác	3b. Đơn không tồn tại hoặc đã ở trạng thái đó sẽ thông báo lỗi.
Tần suất sử dụng	Cao



Hình 2.10 Đặc tả use case hoàn hàng/xuất kho

2.3.3 Đặc tả use case “Đồng bộ giá”

Bảng 2.9 Mô tả tóm tắt use case đồng bộ giá

Tên use case	Đồng bộ giá
ID	UC_003
Mô tả	Use case này để người dùng đồng bộ giá nhanh giữa tất

	cả các sản phẩm được bán trên một sàn có chung một sản phẩm trong kho một cách nhanh chóng.
Các tác nhân tham gia	Người dùng
Điều kiện đầu vào	Yêu cầu đã đăng nhập Yêu cầu đã nhập tài khoản sàn TMDT vào quản lý token. Yêu cầu đã kết nối sản phẩm trong kho với sản phẩm trên sàn.
Kết quả đầu ra	Giá các sản phẩm trên sàn đã kết nối với sản phẩm trong kho được chọn thay đổi về cùng một giá.
Luồng sự kiện chính	1. Người dùng truy cập vào màn hình kho 2. Người dùng chọn sản phẩm cần đồng bộ giá 3. Nhập mức giá 4. Thay đổi giá thành công quay lại màn hình kho
Luồng sự kiện khác	3b. Nếu chưa kết nối sản phẩm sẽ thông báo chưa kết nối sản phẩm
Tần suất sử dụng	Thấp

2.3.4 Đặc tả use case “Cảnh báo vận chuyển”

Bảng 2.10 Mô tả tóm tắt use case cảnh báo vận chuyển

Tên use case	Cảnh báo vận chuyển
ID	UC_004
Mô tả	Use case này để cảnh báo người dùng về đơn hàng đang cần được vận chuyển. Người dùng cần giao hàng ngay cho đơn vị vận chuyển.
Các tác nhân tham gia	Người dùng
Điều kiện đầu vào	Yêu cầu đã đăng nhập Yêu cầu đã nhập tài khoản sàn TMDT vào quản lý token. Yêu cầu có đơn hàng mới ở trạng thái vận chuyển. Yêu cầu quá 1 ngày đơn hàng mới chưa được quét mã ở hoàn hàng/xuất kho
Kết quả đầu ra	Danh sách đơn hàng được cảnh báo
Luồng sự kiện chính	1. Người dùng truy cập vào màn cảnh báo vận chuyển 2. Đơn hàng mới sau 24 giờ nếu chưa được quét mã vận đơn sang xuất hàng sẽ được cảnh báo.
Luồng sự kiện	2b. Đơn hàng quá 15 ngày nếu chưa được quét mã sẽ

khác	không cảnh báo nữa, vì lúc này đơn chắc chắn đã bị hủy.
Tần suất sử dụng	Trung bình

2.4 Yêu cầu phi chức năng

Ngoài những tính năng chính của hệ thống, thì để phát triển hệ thống có đầy đủ tính năng dành cho người dùng và hoạt động ổn định thì hệ thống bao gồm các yêu cầu phi chức năng sau đây:

1. Tính tương thích: Trang web hoạt động tốt với tất cả các trình duyệt phổ biến hiện nay như Chrome, Firefox, Safari có thể dễ dàng xem ở các kích thước màn hình như điện thoại máy tính bảng.
2. Tính tiện lợi: Hệ thống được thiết kế đơn giản, các đề mục rõ ràng, dễ thao tác. Người dùng có thể nhanh chóng xem được thông tin mình cần, các thao tác hợp lý tốn ít thời gian.
3. Tính ổn định: Hệ thống chạy ổn định với tốc độ tải dưới 1s dù có nhiều đơn hàng hay sản phẩm. Giảm cách thời gian request tới các sản phẩm TMDT ở mức 30 phút/lần để tránh bị các sản phẩm TMDT chặn request dẫn đến hệ thống bị sập.
4. Tính riêng tư: Mỗi người dùng được sử dụng hệ thống cài đặt trên một server riêng và không công khai địa chỉ server cho người khác.

Ở chương này em đã phân tích về các tính năng cơ bản và biểu đồ use case của hệ thống quản lý dữ liệu đa sản phẩm TMDT dựa trên những đặc điểm mà hệ thống cần có sau khi đã phân tích các hệ thống đã có và tổng hợp ý kiến từ người dùng. Trong chương 3, em sẽ giới thiệu về các công nghệ cũng như kỹ thuật sẽ được sử dụng trong suốt quá trình phát triển hệ thống trong đồ án này.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Hiện nay, công nghệ phần mềm đang là một ngành quan trọng của công nghệ thông tin, trực tiếp ảnh hưởng đến việc sử dụng các sản phẩm công nghệ thông tin của người dùng trở thành một trong những ngành mũi nhọn trong xã hội. Kéo theo đó là rất nhiều các công nghệ và công cụ trợ giúp việc xây dựng các hệ thống, phần mềm được sinh ra với mục đích giúp các nhà phát triển có thể rút gọn thời gian xây dựng ứng dụng. Trong đồ án này em có sử dụng những công nghệ sau:

3.1 Công nghệ thu thập dữ liệu từ website

3.1.1 Kỹ thuật thu thập dữ liệu từ API

Trong đồ án này, việc lấy dữ liệu danh sách sản phẩm của nhà bán hàng được viết bằng PHP. Đầu tiên sẽ xác định API danh sách sản phẩm của sàn TMDT, sau đó lấy dữ liệu đó về và bóc tách lấy các dữ liệu cần thiết như tên, ID sản phẩm, số lượng tồn kho, thumbnail, giá. Sau đó lưu dữ liệu đó vào database sao cho tất cả các sản phẩm TMDT cùng một chuẩn dữ liệu khi lưu vào database.

Chi tiết hơn về quá trình xác định API em sẽ trình bày ở chương 5.

The screenshot shows the Tiki website's product list interface. The top navigation bar includes links for 'Đăng bán', 'Đề nghị duyệt', 'Chỉnh sửa sản phẩm mẫu', and 'Tạo sản phẩm'. Below this is a form to add a new product with fields for 'Tên sản phẩm', 'ID sản phẩm', 'SKU', and 'Thương hiệu'. The main content area displays a table of products with columns for 'Sản phẩm', 'Tiki NOW', 'Điểm nội dung', 'Nhà cung cấp', 'Giá bán', 'Phi Tiki thu', and 'Lợi nhuận'. The first product listed is 'Kệ sách bảng gỗ hình xương cá 5 tầng - ĐEN' with a price of 210,000 and a Tiki fee of 30,500. To the right of the product list, the browser's developer tools are open, showing a network request to the Tiki API. The request is a GET to 'https://api.tiki.vn/v1/products?seller_id=7187' with a response containing a list of product details in JSON format.

Hình 3.1 Ví dụ về API danh sách sản phẩm của sàn Tiki

3.1.2 Server thu thập dữ liệu sử dụng NodeJS

Ngoài việc đồng bộ dữ liệu sản phẩm, đồ án này còn sử dụng 1 server viết bằng NodeJS để đồng bộ liên tục dữ liệu đơn hàng từ các sản phẩm TMDT, 30 phút 1 lần và gửi dữ liệu đơn hàng đó về server PHP qua API để lưu vào database. Do tính chất đồng bộ đơn hàng rất phức tạp và đã có người làm trước, em đã sử dụng sản phẩm đã được phát triển sẵn cho việc thu thập dữ liệu đơn hàng từ các sản phẩm TMDT để làm server phục vụ việc lấy dữ liệu đơn hàng cho hệ thống.


```
Page 136
Page 137
Page 138
Page 139
Page 140
Page 141
Page 142
Page 143
Page 144
Page 145
Page 146
Page 147
Page 148
Page 149
Page 150
Page 151
Page 152
Page 153
Page 154
Page 155
Page 156
Page 157
Page 158
Page 159
Page 160
Page 161
Page 162
Page 163
Page 164
Page 165
Page 166
Page 167
Page 168
Page 169
Exported all lazada data. Process data
Sync total 2808 in 85521ms
```

Hình 3.2 Màn hình console thông báo việc lấy dữ liệu từ Lazada hoàn tất

3.2 Công nghệ lập trình web

3.2.1 Công nghệ HTML, CSS, Javascript

HTML viết tắt của HyperText Markup Language hay còn được dịch là Ngôn ngữ đánh dấu siêu văn bản, được dùng để tạo ra các siêu văn bản là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web.

CSS viết tắt của Cascading Style Sheets hay còn được dịch là các tập tin định kiểu theo tầng, được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML và XHTML. Ngoài ra CSS cũng có thể dùng cho XML, SVG, XUL. Các đặc điểm kỹ thuật của CSS được duy trì bởi World Wide Web Consortium (W3C).

JavaScript là ngôn ngữ lập trình kịch bản theo hướng đối tượng, được sử dụng để tạo ra các trang web động hay tạo các hiệu ứng trên trang web. Ngoài ra, trong các phiên bản phát triển ngày nay còn có các phiên bản JavaScript phát triển phía máy chủ nâng cao hơn như Node.js cho phép bạn có thể phát triển một trang web động hoàn chỉnh với nhiều tính năng mà chỉ cần sử dụng ngôn ngữ lập trình duy nhất Javascript. Trong môi trường máy chủ JavaScript có thể kết nối với các giao diện của môi trường để cung cấp quyền điều khiển hệ thống cho phần mềm phía máy chủ giống các ngôn ngữ lập trình phía máy chủ khác như PHP, Ruby.

Trong đồ án này, em sử dụng template AdminLTE 2, đây là một admin template sử dụng HTML, CSS, JS, thư viện Bootstrap để thuận tiện cho việc xây

dụng giao diện thân thiện với người sử dụng, thích hợp trên nhiều kích thước màn hình.

3.2.2 Laravel Framework

Laravel là một PHP framework mã nguồn mở và miễn phí, được sử dụng để phát triển các ứng dụng web theo kiến trúc model-view-controller (MVC). Laravel thành công và phổ biến nhờ vào việc có cú pháp dễ hiểu, rõ ràng, có khả năng đóng gói thành các module và quản lý gói phụ thuộc. Laravel cho phép nhiều cách giúp truy cập vào các CSDL quan hệ, đồng thời cũng cấp các tiện ích khác nhau phục vụ việc triển khai vào bảo trì ứng dụng. Laravel có một cộng đồng người dùng vô cùng đông đảo.

Laravel có tính module, vì vậy đồ án sử dụng cấu trúc module, để có thể thiết kế và mở rộng một ứng dụng một cách dễ dàng, có thể phân chia logic nghiệp vụ thành các module khác nhau, tất cả đều hoạt động cùng nhau để làm cho ứng dụng web hoạt động. Laravel cung cấp các hướng dẫn rất đơn giản để tạo các module hoặc package trong Laravel.

Phiên bản Laravel em đang sử dụng là 5.8.36 và phiên bản PHP là 7.2.31. Laravel sẽ được sử dụng để xây dựng backend cho hệ thống.

3.2.3 Hệ quản trị cơ sở dữ liệu MySQL

MySQL là hệ quản trị CSDL được đông đảo các lập trình viên tin dùng cho việc phát triển ứng dụng. Đây là hệ quản trị cơ sở dữ liệu có thể hoạt động trên nhiều hệ điều hành với nhiều điểm mạnh và cung cấp một hệ thống lớn các tiện ích cho người dùng và hoàn toàn miễn phí.

MySQL có nhiều điểm mạnh. Thứ nhất, MySQL rất nhanh, là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn. Thứ hai, MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Thứ ba, nó có thể truy cập MySQL bằng cách sử dụng các ứng dụng hỗ trợ ODBC (Open Database Connectivity - một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft). Thứ tư, nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó có thể chia sẻ dữ liệu với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu nào đó thì không thể nhìn được. Thứ năm, MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với tất cả phần cứng có thể có, từ các máy PC ở nhà cho đến các máy server. Có thể dễ dàng tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ.

Với đặc trưng là miễn phí, tốc độ và tính bảo mật cao, em đã lựa chọn MySQL làm hệ quản trị cơ sở dữ liệu cho hệ thống trong đồ án tốt nghiệp này. Việc nó có rất nhiều người sử dụng cũng hỗ trợ em rất nhiều trong quá trình sử dụng nó.

3.2.4 Công nghệ RESTful API

Công nghệ RESTful API được sử dụng để thiết kế các API sử dụng cho các ứng dụng web, tiện lợi trong việc quản lý các resource. Là tiêu chuẩn được xây dựng dựa trên giao thức HTTP do đó trọng tâm của REST quy định cách sử dụng các HTTP method cho các API. Với các ứng dụng web được thiết kế sử dụng RESTful, lập trình viên có thể dễ dàng biết được URL và HTTP method để quản lý một tài nguyên. Bản thân RESTful không quy định logic của ứng dụng và RESTful cũng không giới hạn bởi ngôn ngữ lập trình ứng dụng, do đó bất kỳ ngôn ngữ lập trình (hoặc framework) nào cũng có thể áp dụng RESTful trong việc thiết kế API cho ứng dụng web của mình. Với những điểm nổi bật thì RESTful là một trong những kiểu thiết kế API được tin tưởng sử dụng bởi các lập trình viên và trở nên phổ biến nhất ngày nay.

Có 4 phương thức chính của RESTful tương ứng với các phương thức phía server như sau:

Bảng 3.1 Các phương thức chính của RESTful

RESTful	Web server
POST: Tạo mới dữ liệu	Create
GET: Lấy dữ liệu	Read
PUT: Cập nhật hoặc thay đổi dữ liệu	Update
DELETE: Xóa dữ liệu	Delete

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

Trong đồ án này RESTful API dùng để kết nối giữa server PHP với server NodeJs. Dữ liệu từ server NodeJS sẽ được gửi qua API để lưu vào database.

Trên đây em đã giới thiệu xong những công nghệ được sử dụng trong đồ án. Tiếp theo em sẽ phân tích kỹ hơn việc sử dụng những công nghệ đó để phát triển và triển khai ứng dụng.

CHƯƠNG 4. PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG

Ở chương 3, em đã giới thiệu về các công nghệ tảng cho quá trình xây dựng hệ thống. Để tiếp tục triển khai ý tưởng của mình, trong chương 4 này em xin giới thiệu về kiến trúc tổng quan, mô hình cũng như các thiết kế chi tiết xuyên suốt toàn bộ hệ thống. Đồng thời, làm rõ mối quan hệ, phương pháp giao tiếp để các thành phần trong hệ. Phần cuối chương này, em xin trình bày những tính năng quan trọng đã hoàn thiện, cũng như kết quả đạt được thông qua một số hình ảnh minh họa, kết quả kiểm thử khả năng vận hành của hệ thống và việc triển khai hệ thống.

4.1 Thiết kế kiến trúc

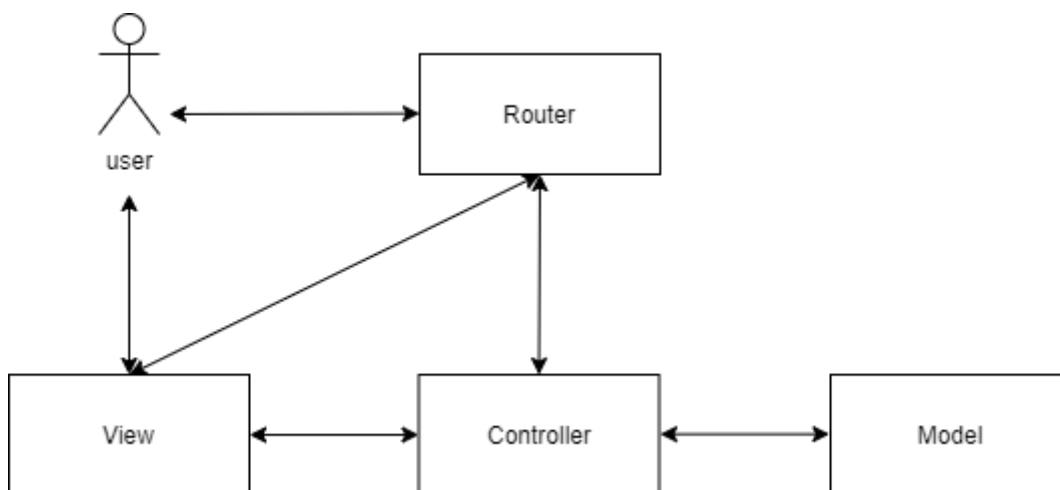
4.1.1 Lựa chọn kiến trúc phần mềm MVC

Hệ thống trong đề án này được triển khai dựa trên kiến trúc phần mềm MVC. MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình chia source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Model (M): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,...

View (V): Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. Thông thường, các ứng dụng web sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi. Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấn vào nút đó sẽ có một hành động trong Controller.

Controller (C): Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

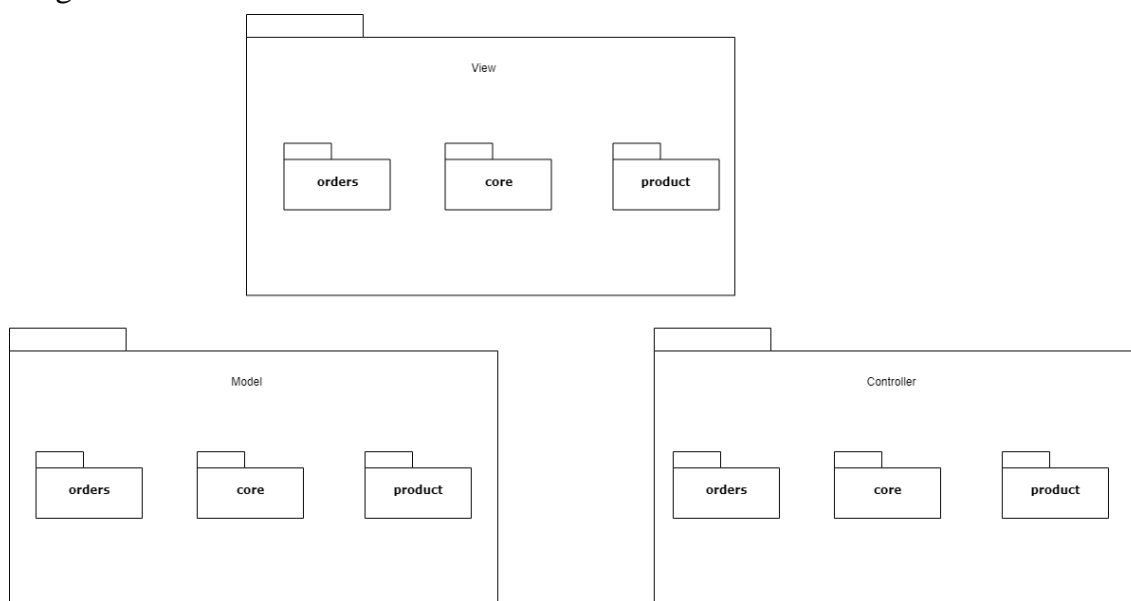


Hình 4.1 Mô hình MVC trong Laravel

Trong Laravel, mọi request từ phía người dùng đều phải qua Route, dữ liệu được gửi xuống Controller để xử lý, cần dữ liệu sẽ lấy từ Model lên hoặc cập nhật dữ liệu xuống Model, kết quả gửi ra View cho người sử dụng.

4.1.2 Thiết kế tổng quan

Sau đây em sẽ mô tả chi tiết sự phân chia các gói phần mềm trong hệ thống bằng UML



Hình 4.2 Biểu đồ tổng quan gói

Hệ thống này được xây dựng dựa trên khả năng module hóa của Laravel framework và mô hình MVC. Mỗi module lại là một mô hình MVC nhỏ hơn.

Với từng module, các thành phần trong View, Model, Controller là các class trong thư mục

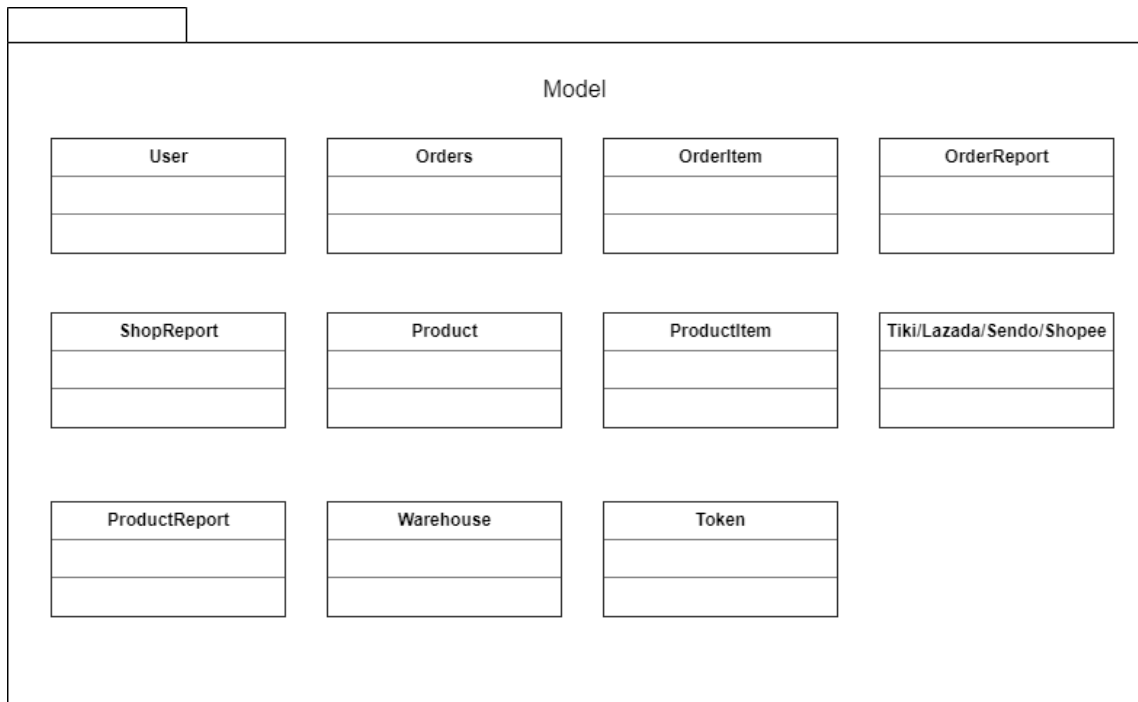
Modules/{Core|Orders|Product}/Resources/View

Modules/{Core|Orders|Product}/Entities

Modules/{Core|Orders|Product}/Http/Controller

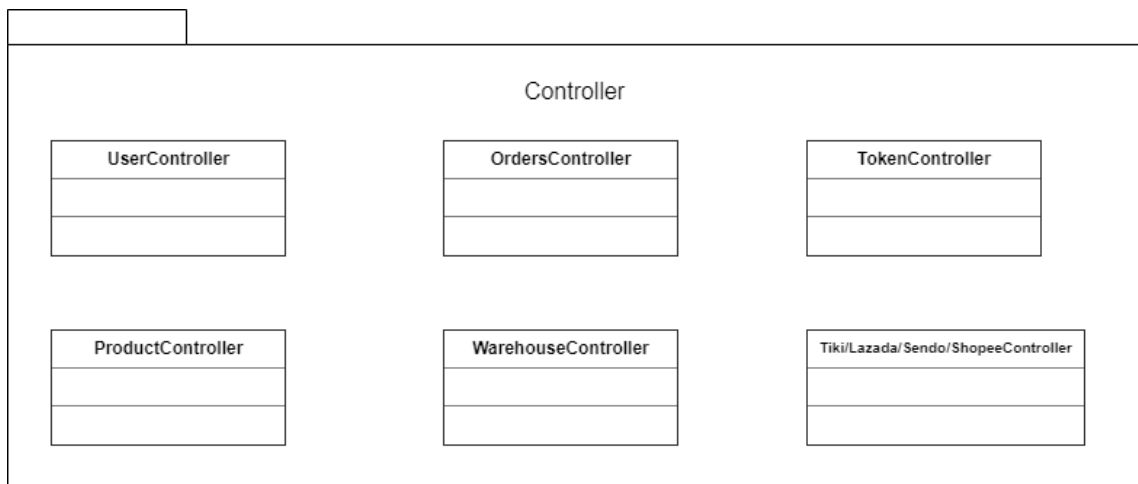
Các chức năng thuộc cùng một nhóm sẽ được nhóm lại thành một module để việc quản lý trở nên dễ dàng hơn.

4.1.3 Thiết kế chi tiết gói



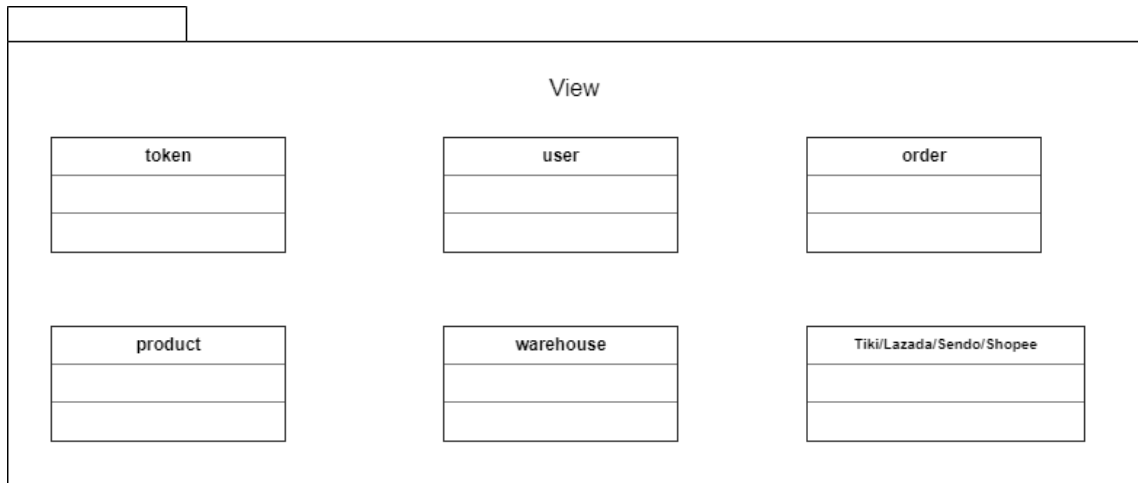
Hình 4.3 Biểu đồ gói chi tiết cho package Model

Gói model chứa các class model như user, orders, product ... là các class tương ứng tham chiếu tới các bảng tương ứng bên trong CSDL.



Hình 4.4 Biểu đồ gói chi tiết cho package Controller

Gói controller chứa các class controller như UserController, OrdersControllers, ProductController ... là các class xử lý logic từ view cũng như lấy dữ liệu từ model.



Hình 4.5 Biểu đồ gói chi tiết cho package View

Gói view chứa các view như user, order, product ... hiển thị giao diện đã được đổ dữ liệu ra cho người sử dụng.

4.2 Thiết kế chi tiết

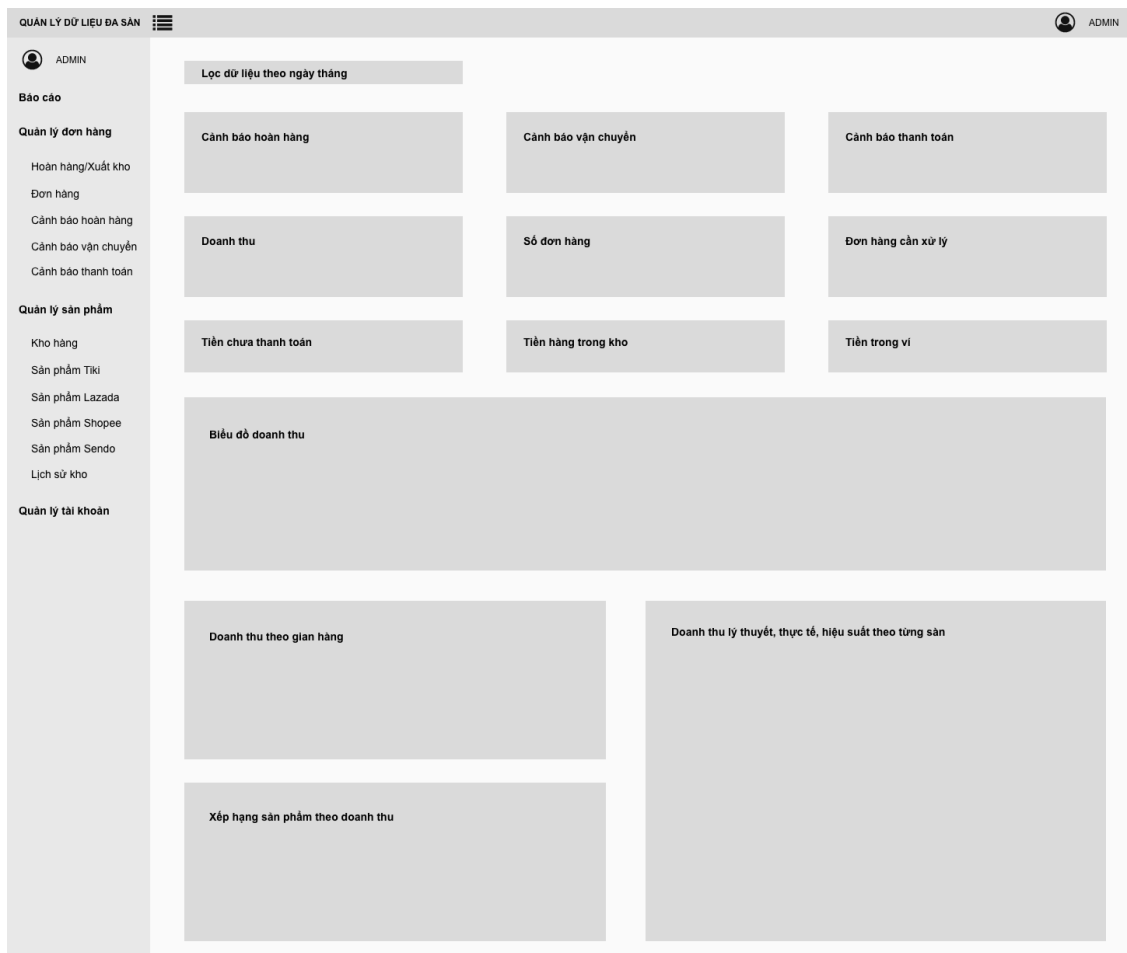
4.2.1 Thiết kế giao diện

Giao diện trang web hệ thống được thiết kế với quy tắc tối giản mọi chi tiết, tạo cảm giác thoải mái cho mắt người dùng, tối ưu về UX. Tuy tối giản nhưng nội dung trên trang web vẫn được hiển thị với đầy đủ thông tin cho người dùng. Các tính năng được tối giản số nút bấm sao cho người dùng dễ dùng.

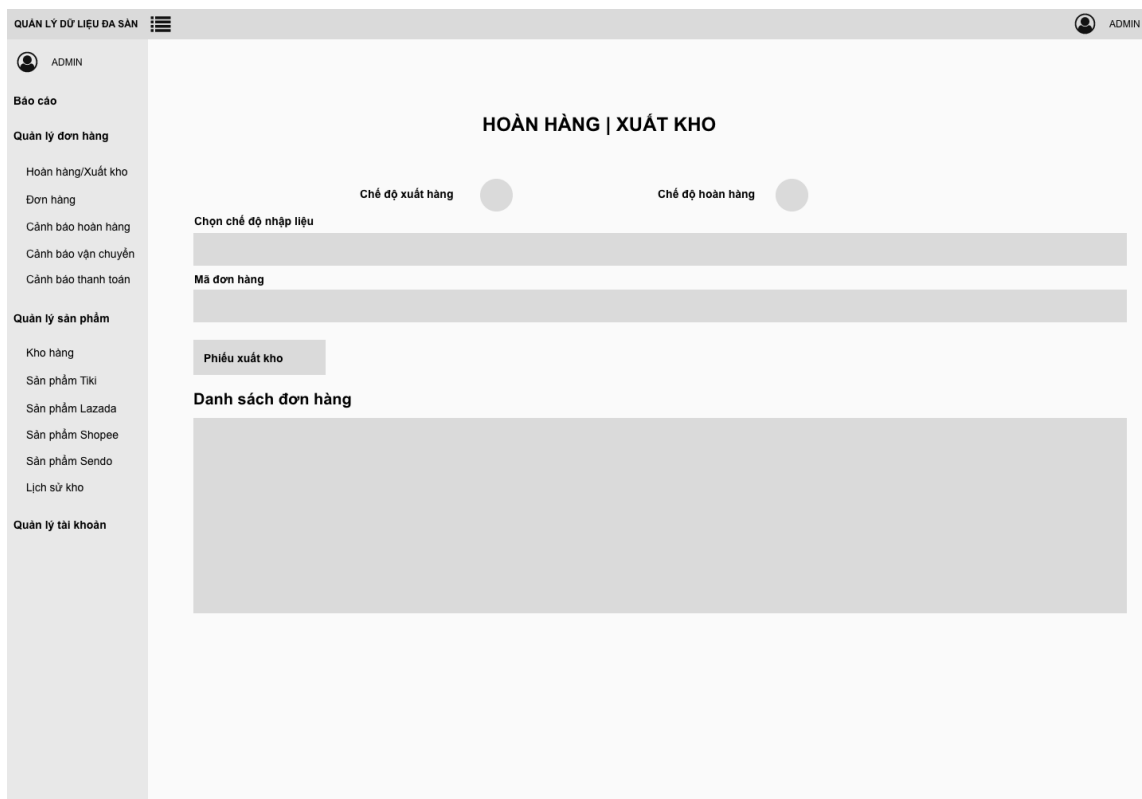
Hiện tại trang web đã được chạy thử trên tất cả các trình duyệt trên máy tính và các thiết bị di động với các kích thước màn hình 1024X768 px, 1280X800 px, 1360X768 px, 1366X768 px, 800x600 px.

Là một hệ thống quản lý dữ liệu bán hàng, em ưu tiên việc phát triển tính ổn định của hệ thống trước rồi mới đến cải tiến giao diện, nên giao diện hiện tại tuy cũ nhưng cố gắng tối ưu UI/UX và sẽ cập nhật mới giao diện về sau.

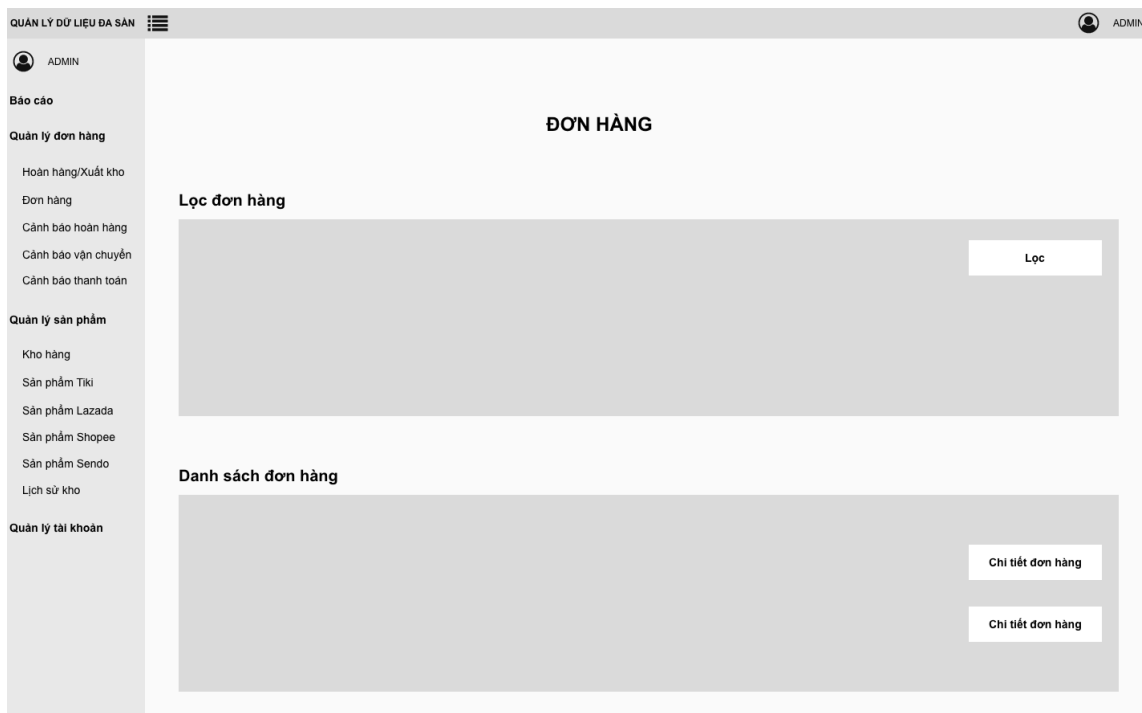
Sau đây sẽ là bản vẽ thiết kế giao diện cho hệ thống:



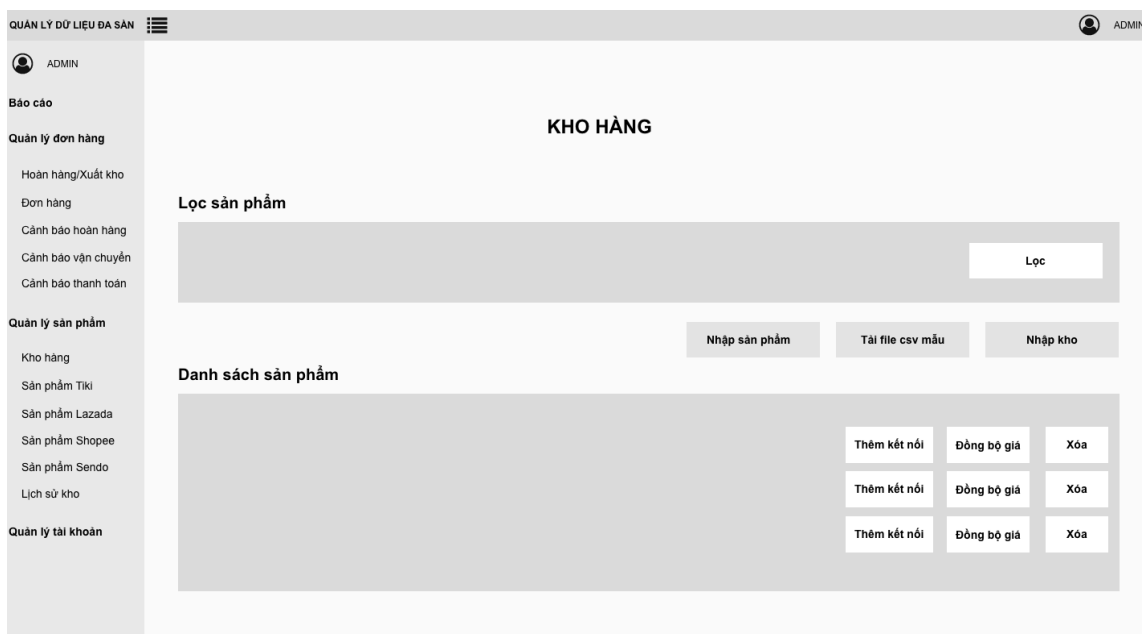
Hình 4.6 Thiết kế giao diện màn hình báo cáo



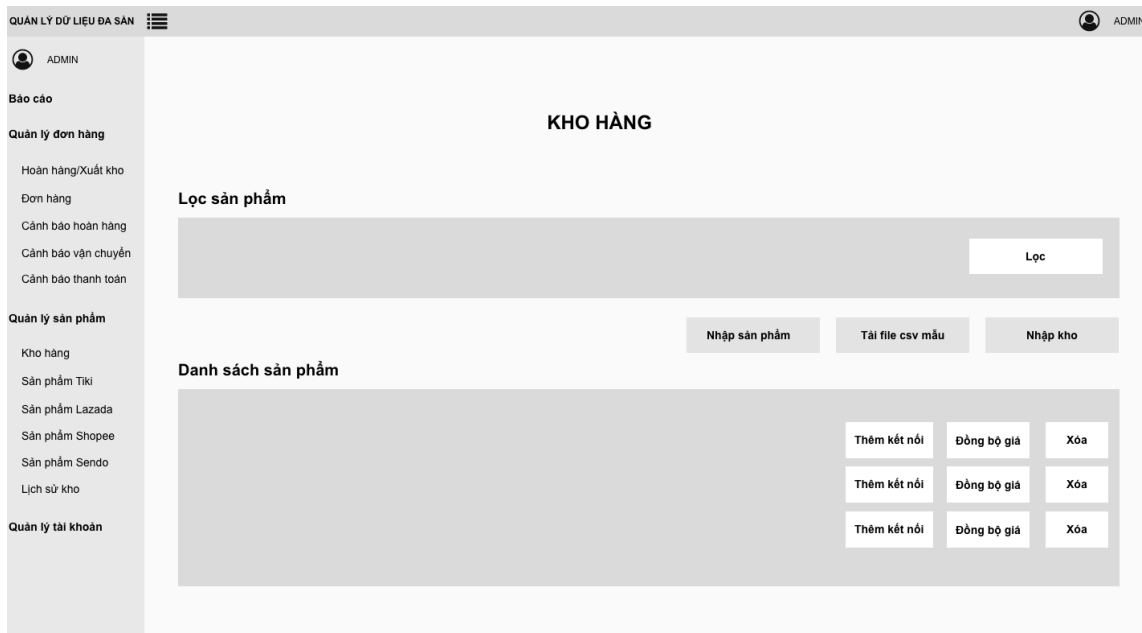
Hình 4.7 Thiết kế giao diện màn hình Hoàn hàng/Xuất kho



Hình 4.8 Thiết kế giao diện màn hình Đơn hàng

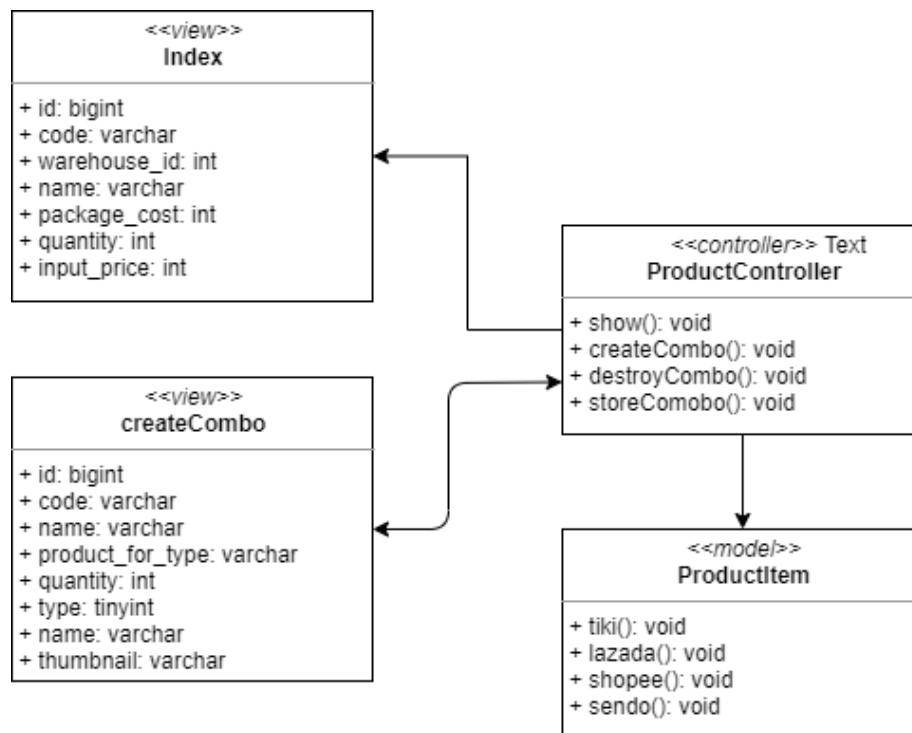


Hình 4.9 Thiết kế giao diện màn hình Kho hàng



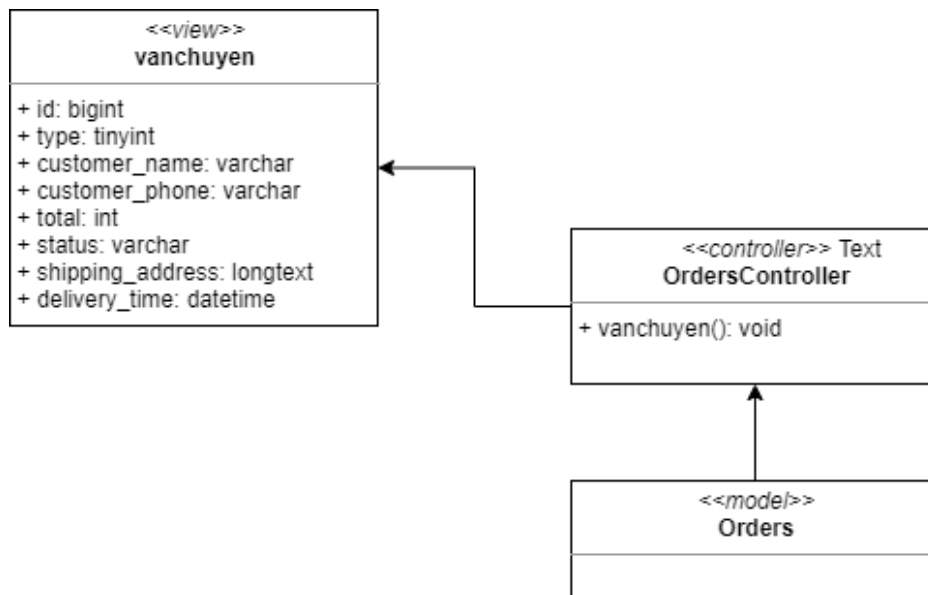
Hình 4.10 Thiết kế giao diện màn hình Thêm kết nối

4.2.2 Thiết kế lớp



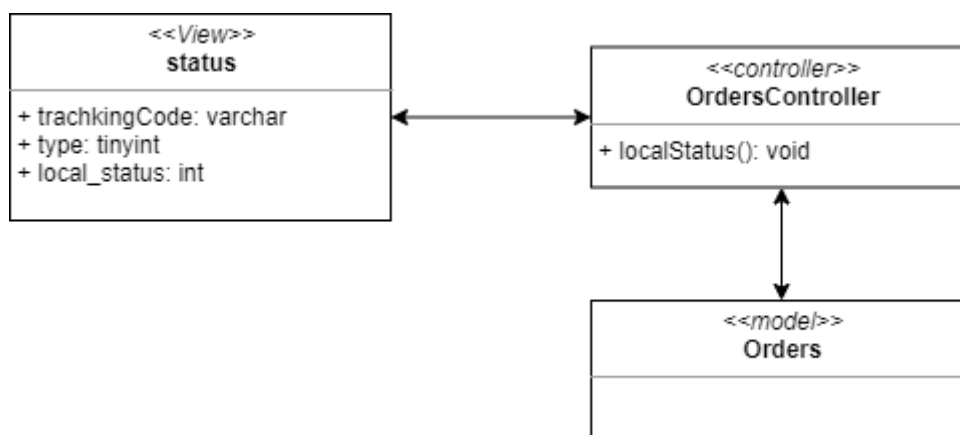
Hình 4.11 Thiết kế lớp của use case kết nối sản phẩm

Use case kết nối sản phẩm: View Index hiển thị danh sách sản phẩm trong kho được xử lý từ ProductController. Khi người dùng chọn kết nối sản phẩm, view createCombo hiện ra, lấy dữ liệu được xử lý từ ProductController, khi kết nối xong dữ liệu được gửi qua ProductController và qua ProductItem để lưu vào cơ sở dữ liệu.



Hình 4.12 Thiết kế lớp của use case cảnh báo vận chuyển

Use case cảnh báo vận chuyển: Đơn hàng được lấy từ cơ sở dữ liệu qua Orders, được xử lý để tách ra những đơn hàng cần cảnh báo ở OrdersController và hiển thị ra view vanchuyen cho người dùng.

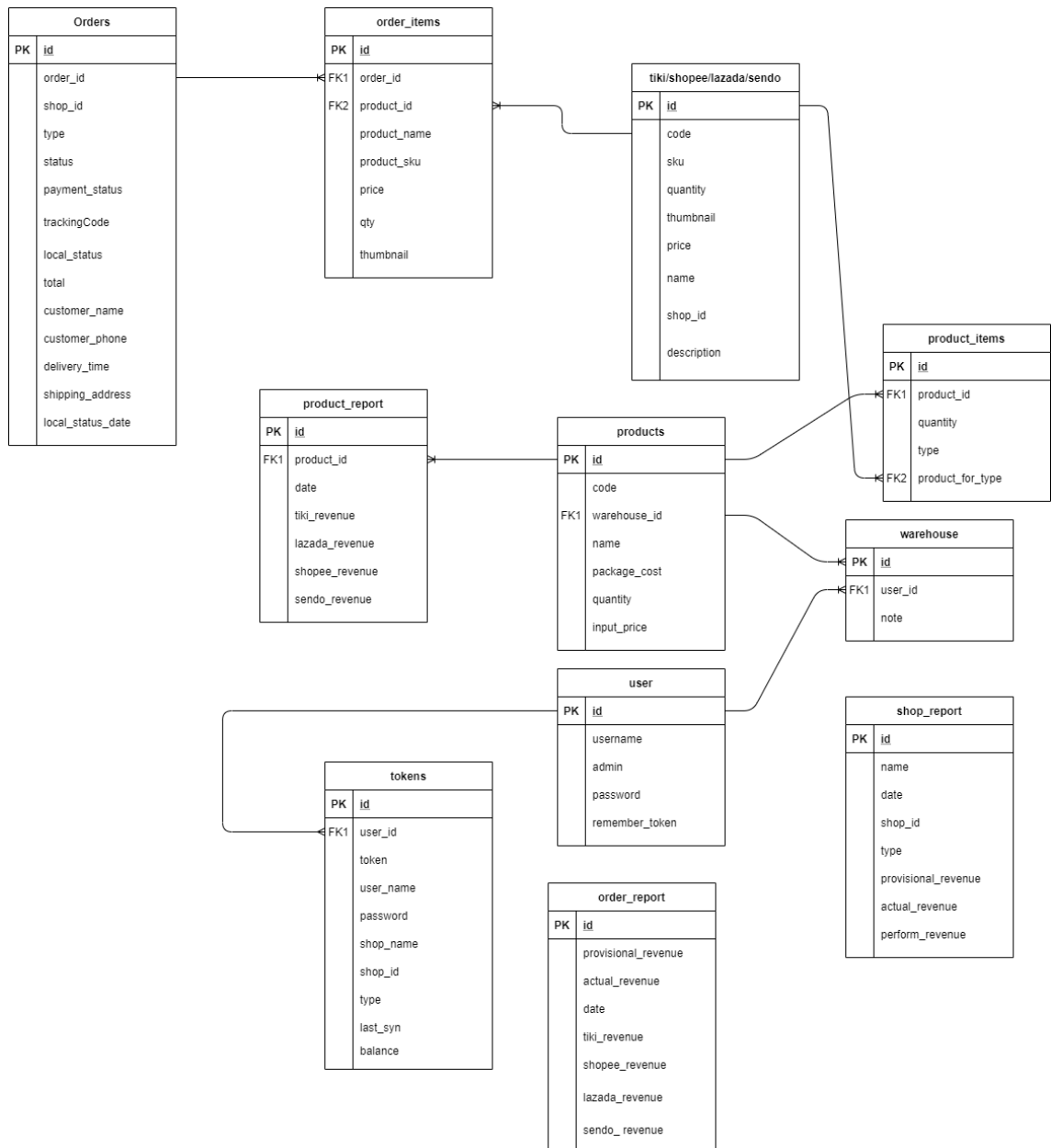


Hình 4.13 Thiết kế lớp của use case hoàn hàng/xuất kho

Use case hoàn hàng/xuất kho: Đơn hàng được người dùng nhập vào qua view, và xử lý việc thay đổi local_status ở OrdersController rồi lưu dữ liệu vào bảng Orders trong CSDL.

4.2.3 Thiết kế cơ sở dữ liệu

Trước khi đi vào mô tả cụ thể các thuộc tính của các lớp cũng như mối quan hệ của chúng trong cơ sở dữ liệu em sẽ đưa ra biểu đồ thực thể liên kết:



Hình 4.14 Mô hình thực thể liên kết cơ sở dữ liệu hệ thống

Thông qua biểu đồ thực thể liên kết em sẽ tiếp tục mô tả cụ thể từng bảng dữ liệu tương ứng được lưu trong hệ quản trị cơ sở dữ liệu như sau:

Bảng 4.1 Mô tả các bảng trong CSDL

Tên bảng	Tên trường	Kiểu dữ liệu	Mô tả
users	id	int	id của user, được sử dụng làm khoá chính.
	username	varchar	Tên đăng nhập của user
	admin	tinyint	Đánh dấu tài khoản này có phải là tài khoản admin không
	remember_token	varchar	Lưu token ghi nhớ việc đăng nhập khi người dùng chọn ghi nhớ đăng nhập
	password	varchar	Mật khẩu của user, đã được mã hóa
tokens	id	int	id của token, được sử dụng làm khoá chính.
	user_id	int	id bảng users, dùng làm khóa ngoài
	token	longtext	Token đăng nhập vào tài khoản trang quản trị bán hàng các sàn TMDT
	user_name	varchar	Tên đăng nhập vào sàn TMDT
	password	password	Mật khẩu đăng nhập vào sàn TMDT
	shop_name	shop_name	Tên gian hàng ứng với tài khoản đã nhập
	shop_id	int	ID gian hàng do sàn TMDT cấp
	type	type	Phân loại gian hàng là shopee, tiki, sendo hay lazada
	balance	float	Số dư tài khoản trong ví của các tài khoản trên sàn TMDT
Tiki/ Shopee/ Lazada/ Sendo	id	int	id của tiki/shopee/lazada/sendo, được sử dụng làm khoá chính.
	name	varchar	Tên sản phẩm
	code	varchar	Mã số của sản phẩm
	quantity	int	Số lượng tồn kho
	thumbnail	varchar	Đường dẫn ảnh thumbnail của sản phẩm
	price	int	Giá sản phẩm
	description	varchar	Mô tả của sản phẩm
	shop_id	int	ID shop sở hữu sản phẩm
	sku	varchar	Mã SKU do sàn TMDT tạo cho sản phẩm
warehouse	id	int	id của warehouse, được sử dụng làm khoá chính.
	user_id	int	id bảng users, dùng làm khóa ngoài
	note	varchar	Nội dung ghi chú trong lịch sử kho
products	id	int	id của products, được sử dụng làm khoá chính.
	code	varchar	Mã sản phẩm do người dùng đặt
	warehouse_id	int	id của bảng warehouse, được sử dụng làm khoá ngoài.
	name	varchar	Tên sản phẩm
	package_cost	int	Chi phí đóng gói
	quantity	int	Số lượng tồn kho
	input_price	int	Giá nhập vào của sản phẩm
product_items	id	int	id của product_items, được sử dụng làm khoá chính.
	product_id	int	id của bảng products, được sử dụng làm khoá ngoài.
	quantity	int	Số lượng sản phẩm trong kết nối
	type	tinyint	Loại sản của sản phẩm
	product_for_type	varchar	ID của sản phẩm trên sàn
orders	id	bigint	id của orders, được sử dụng làm khoá chính.

	order_id	varchar	ID của đơn hàng do sản TMDT tạo ra
	shop_id	int	ID của gian hàng có đơn hàng
	type	tinyint	Loại sản của đơn hàng
	status	varchar	Trạng thái của đơn hàng
	payment_status	tinyint	Trạng thái thanh toán
	trackingCode	varchar	Mã vận đơn
	local_status	int	Trạng thái nội bộ của sản phẩm trong đơn được thay đổi ở Hoàng hàng/Xuất kho
	total	int	Giá trị đơn hàng
	customer_name	varchar	Tên khách hàng
	customer_phone	varchar	Số điện thoại khách hàng
	delivery_time	datetime	Thời gian giao hàng
	shipping_address	longtext	Địa chỉ giao hàng
order_items	id	int	id của order_items, được sử dụng làm khoá chính.
	type	tinyint	Loại sản của sản phẩm trong đơn hàng
	order_id	varchar	ID của đơn hàng
	product_id	varchar	ID của sản phẩm trong đơn hàng
	product_name	varchar	Tên của sản phẩm trong đơn hàng
	product_sku	varchar	Mã SKU của sản phẩm trong đơn hàng
	price	float	Giá trị sản phẩm
	qty	int	Số lượng sản phẩm trong đơn
	thumbnail	varchar	Đường dẫn ảnh của sản phẩm
product_report	id	int	id của product_report, được sử dụng làm khoá chính.
	product_id	Int	ID của sản phẩm trong kho
	date	date	Ngày mà dữ liệu được tổng hợp
	tiki_revenue	int	Doanh thu tính theo sản tiki
	lazada_revenue	int	Doanh thu tính theo sản lazada
	shopee_revenue	int	Doanh thu tính theo sản shopee
	sendo_revenue	int	Doanh thu tính theo sản sendo
order_report	id	int	id của order_report, được sử dụng làm khoá chính.
	provisional_revenue	int	Doanh thu lý thuyết
	actual_revenue	int	Doanh thu thực tế
	date	date	Ngày mà dữ liệu được tổng hợp
	tiki_revenue	int	Doanh thu tính theo sản tiki
	lazada_revenue	int	Doanh thu tính theo sản lazada
	shopee_revenue	int	Doanh thu tính theo sản shopee
	sendo_revenue	int	Doanh thu tính theo sản sendo
shop_report	id	int	id của shop_report, được sử dụng làm khoá chính.
	provisional_revenue	int	Doanh thu lý thuyết
	actual_revenue	int	Doanh thu thực tế
	date	date	Ngày mà dữ liệu được tổng hợp
	name	int	Tên của shop
	perform_revenue	int	Doanh thu hiệu suất
	type	tinyint	Loại sản của gian hàng
	Shop_id	int	ID của shop do sản cấp

Lưu ý: Tất cả các bảng đều có trường created_at, updated_at, deleted_at mặc định của Laravel.

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Để thực hiện nghiên cứu, thiết kế và triển khai đồ án này em đã sử dụng các công cụ và thư viện được liệt kê trong bảng dưới đây:

Bảng 4.2 Danh sách công cụ, thư viện quan trọng

Mục đích	Công cụ	Địa chỉ URL
Tạo giao diện trang admin	AdminLTE2	https://adminlte.io/themes/AdminLTE/index2.html
IDE lập trình	PHP Storm	https://www.jetbrains.com/phpstorm/
Lưu trữ database	AWS RDS	https://aws.amazon.com/vi/rds/
Máy chủ triển khai ứng dụng web	AWS Elastic Beanstalk	https://aws.amazon.com/vi/elasticbeanstalk/
Thư viện load dữ liệu phân trang và không load lại trang.	Thư viện yajra	https://github.com/yajra/laravel-datatables

4.3.2 Kết quả đạt được

Kết quả đạt được là một hệ thống dạng website application gồm 2 thành phần chính là server PHP và server NodeJS.

Server Node JS chạy độc lập trên một máy chủ liên tục 30 phút lại lấy dữ liệu đơn hàng gửi về cho server PHP trên một máy chủ khác để lưu dữ liệu vào cơ sở dữ liệu. Người dùng tương tác với server PHP thông qua giao diện trang quản trị.

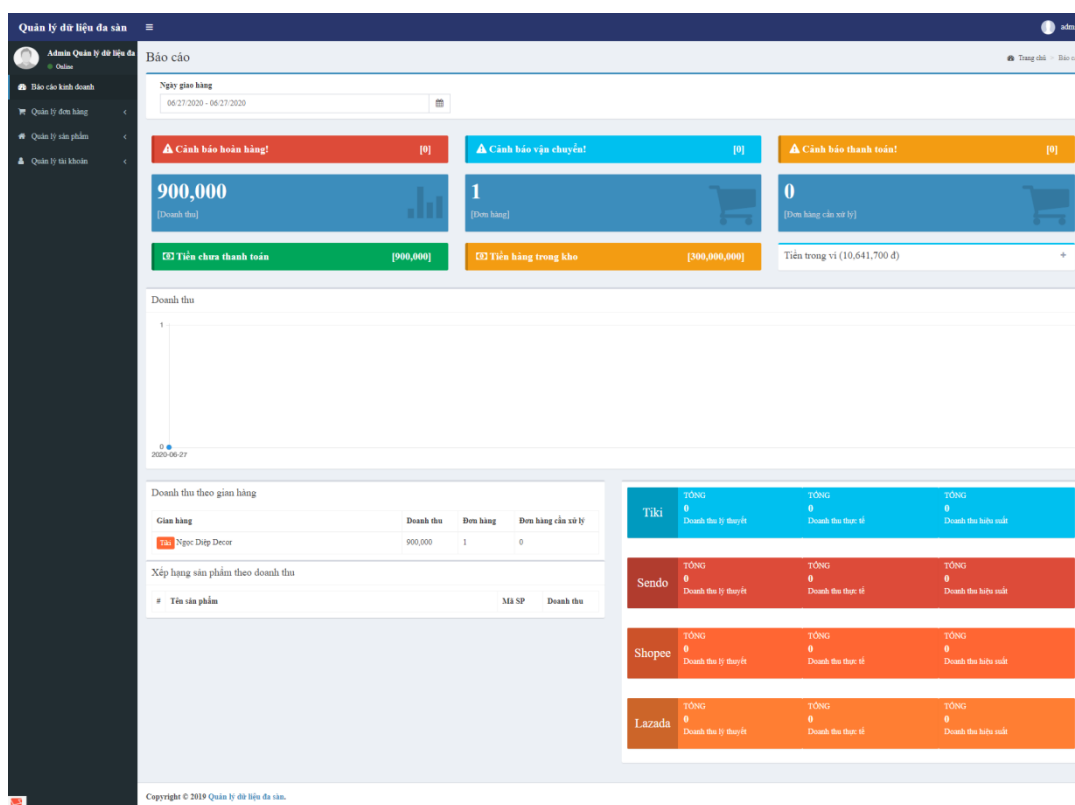
Các máy chủ em sử dụng dịch vụ của AWS.

Bảng 4.3 Kết quả đóng gói sản phẩm

Module	Số gói	Dung lượng mã nguồn	Sản phẩm đóng gói	Dung lượng sản phẩm đóng gói
PHP	1	270MB	Không có	Không có
NodeJS	1	298MB	Không có	Không có

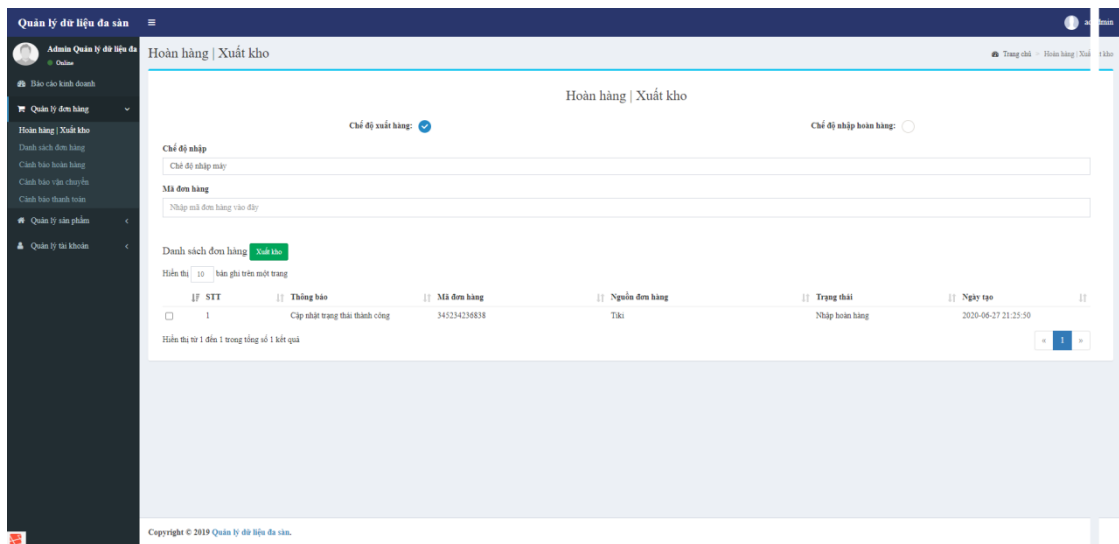
4.3.3 Minh họa các chức năng chính

Sau đây em sẽ minh họa bằng hình ảnh các chức năng chính:



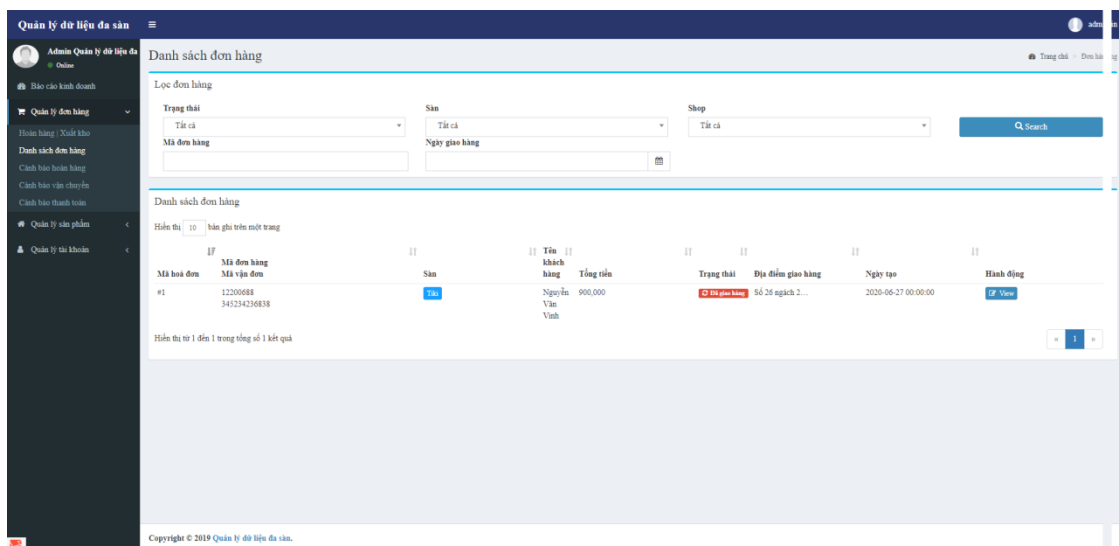
Hình 4.15 Màn hình báo cáo kinh doanh

Màn hình báo cáo kinh doanh hiển thị các thông tin về việc kinh doanh, doanh thu doanh số, hiển thị các cảnh báo nhanh và truy cập nhanh vào trang cảnh báo.



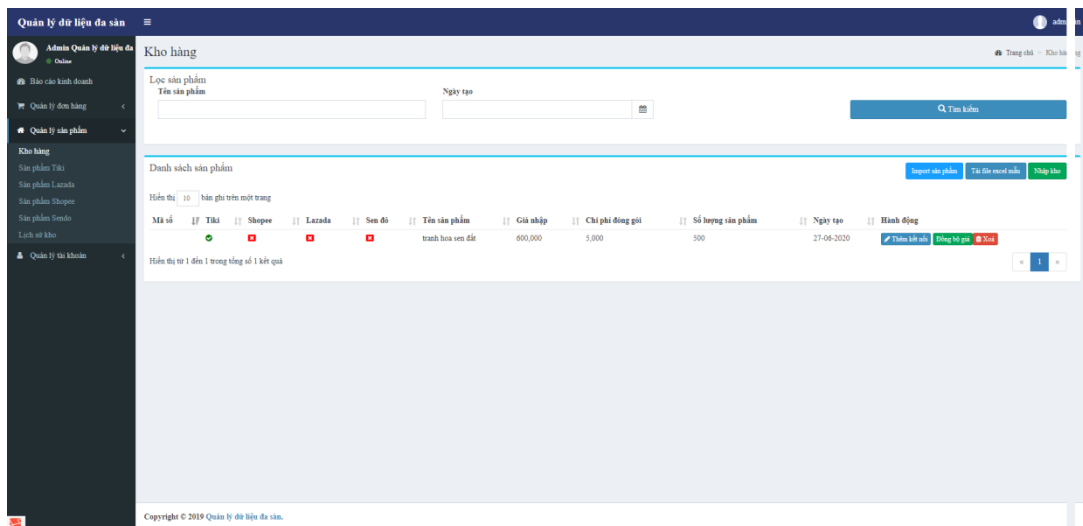
Hình 4.16 Màn hình hoàn hàng/xuất kho

Màn hình hoàn hàng/xuất kho cho phép người dùng nhập mã vận đơn để thay đổi trạng thái nội bộ của đơn hàng, từ đó thay đổi số lượng tồn kho.



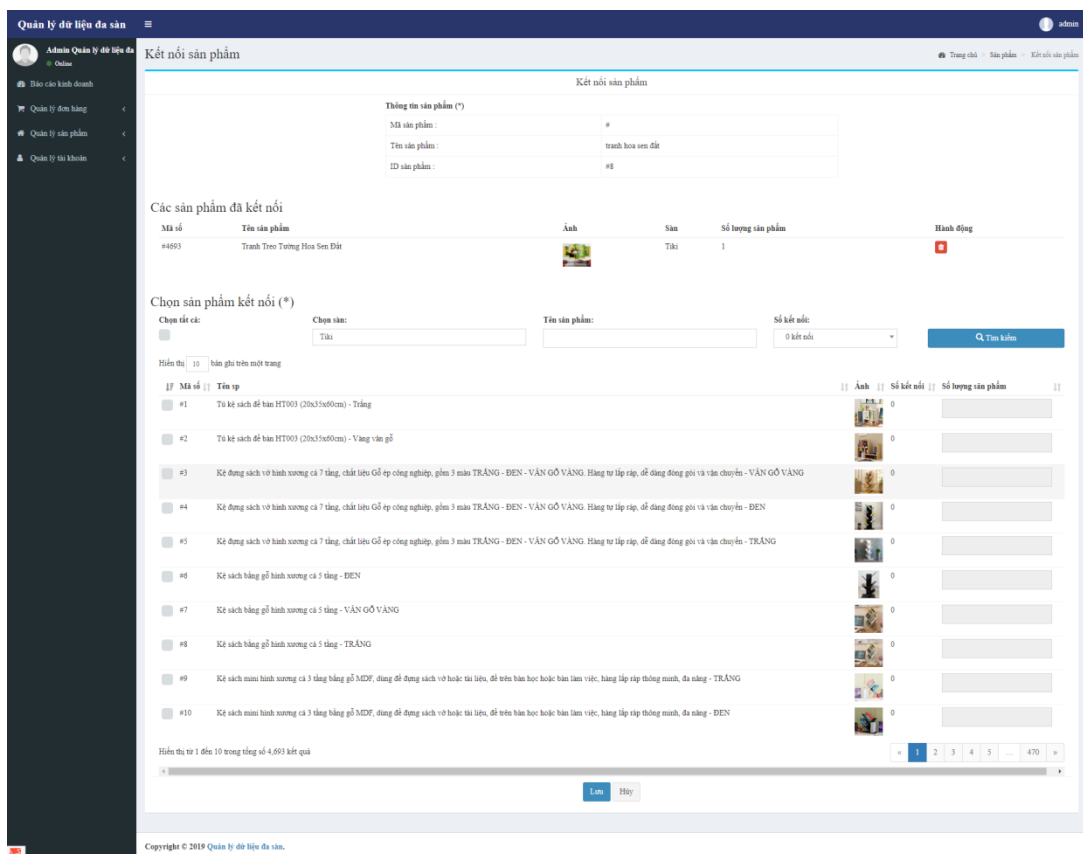
Hình 4.17 Màn hình danh sách đơn hàng

Màn hình danh sách đơn hàng hiển thị tất cả các đơn hàng của người sử dụng, cho phép xem chi tiết thông tin đơn hàng.



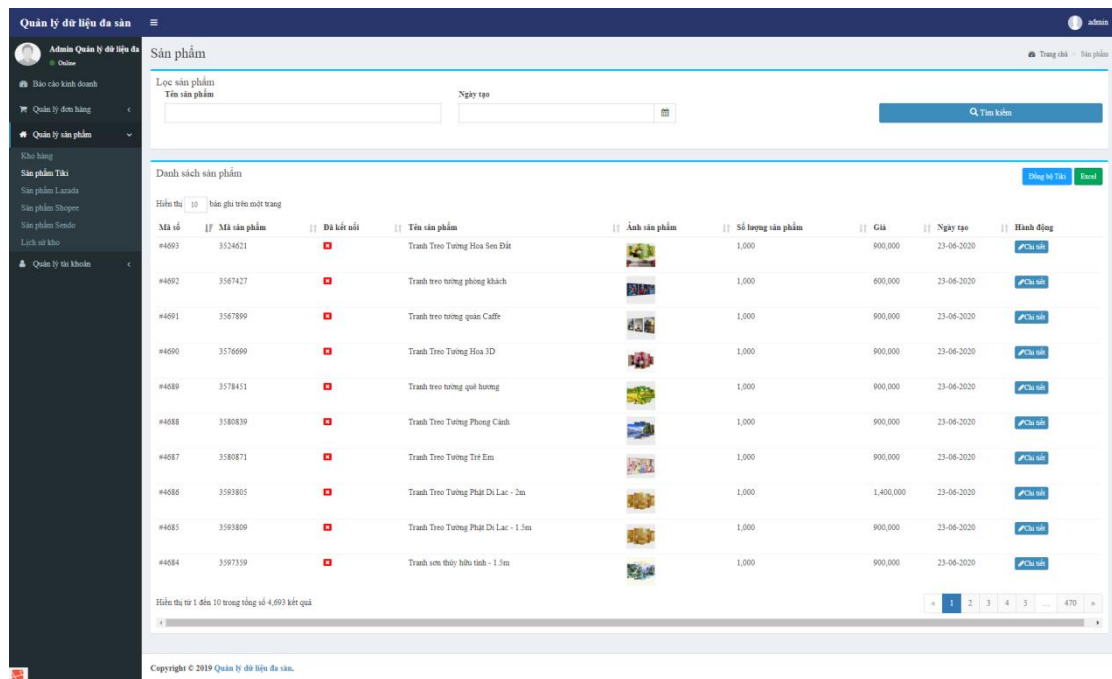
Hình 4.18 Màn hình kho hàng

Màn hình kho hàng quản lý các sản phẩm trong kho của người, thêm sản phẩm hoặc nhập kho, kết nối với sản phẩm trong các sàn.



Hình 4.19 Màn hình thêm kết nối

Người dùng chọn sản phẩm trong kho và kết nối với sản phẩm trên các sàn trong màn hình thêm kết nối sau khi điền số lượng sản phẩm. Có thể xóa sản phẩm đã kết nối



Hình 4.20 Màn hình danh sách sản phẩm trên sàn

Màn hình danh sách sản phẩm trên sàn hiển thị tất cả các sản phẩm đã được đồng bộ về, với các phân loại đã được tách ra và đặt thành tên.

4.4 Kiểm thử

Kịch bản kiểm thử của hệ thống gồm có 7 chức năng lớn, dưới đây là mô tả kịch bản kiểm thử.

Bảng 4.4 Mô tả kịch bản kiểm thử

STT	Chức năng	Kịch bản kiểm thử	Kết quả mong đợi
1	Đăng nhập	1. Điền thông tin người dùng 2. Đăng nhập 3. Tích nút nhớ mật khẩu	1. Hiện thị màn hình đăng nhập. 2. Kiểm tra thông tin đăng nhập có chính xác không và thông báo tới người dùng. 3. Nếu thành công sẽ chuyển tới màn hình báo cáo nếu thất bại sẽ không báo cho người dùng.
2	Đăng xuất	1. Đăng xuất tài khoản 2. Không cho xem hệ thống sau khi đăng xuất.	1. Sau khi đăng xuất hệ thống chuyển về màn hình đăng nhập. 2. Nếu vào đường dẫn một màn hình của hệ thống thì sẽ ra màn hình đăng nhập
3	Xem báo cáo	1. Chọn lọc báo cáo theo khoảng ngày	1. Các báo cáo lọc ra đúng như khoảng thời gian đã

		tháng năm	chọn. Chỉ các thông số liên quan đến doanh thu mới thay đổi theo bộ lọc
4	Hoàn hàng/Xuất kho	1. Chuyển trạng thái đơn hàng từ mặc định sang xuất kho, từ xuất kho sang hoàn hàng. 2. Tạo phiếu xuất kho	1. Trạng thái chuyển thành công nếu đơn hàng tồn tại và trạng thái phù hợp, nếu không sẽ thông báo lỗi. 2. Tải về phiếu xuất kho gồm lịch sử quét mã vận đơn dạng xls.
5	Đơn hàng	1. Lọc đơn hàng 2. Xem chi tiết đơn hàng	1. Kết quả lọc đúng như đã chọn 2. Hiện thị màn hình chi tiết đơn hàng
6	Kho	1. Nhập kho 2. Tải file excel mẫu 3. Nhập file excel sản phẩm 4. Kết nối sản phẩm	1. Kiểm tra chỉ cho nhập kho khi nhập đủ các ô input. Thông báo khi nhập kho thành công. 2. Tải được file excel mẫu thành công. 3. Kiểm tra nội dung file excel hợp lệ không, nếu hợp lệ nhập kho theo nội dung file. Nếu không thông báo ra cho người dùng. 4. Kiểm tra thông tin kết nối, nếu hợp lệ cho phép kết nối, nếu không thông báo cho người dùng.
7	Quản lý token	1. Nhập tên đăng nhập, mật khẩu và chọn sần	1. Thử lấy token dựa trên thông tin đăng nhập đã được nhập, nếu lấy thành công hoặc không thành công hiện thông báo. Nếu thành công bắt đầu đồng bộ dữ liệu đơn hàng và sản phẩm.

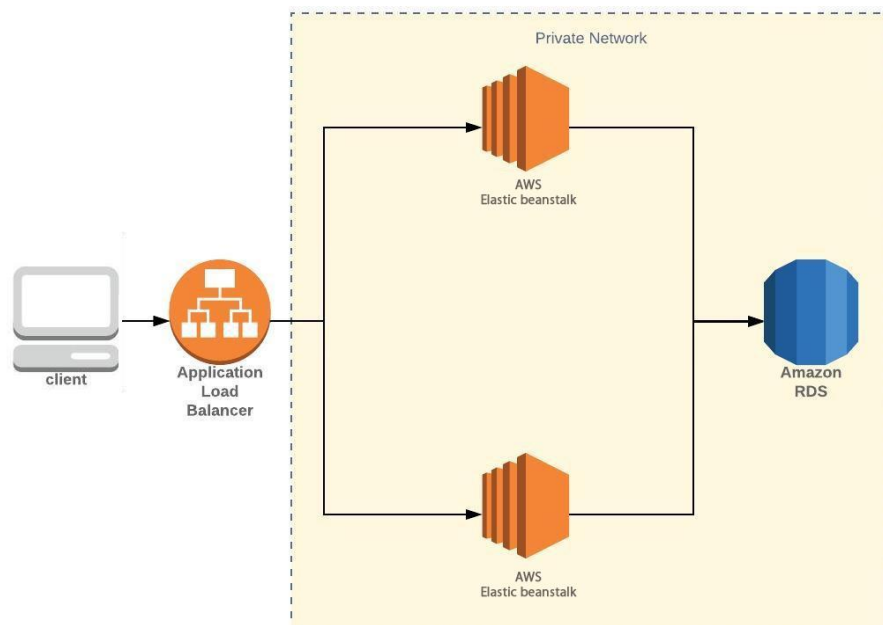
Dưới đây là kết quả kiểm thử của hệ thống:

Bảng 4.5 Kết quả kiểm thử

Chức năng	Chrome	Edge	Firefox	Safari
1	Pass	Pass	Pass	Pass
2	Pass	Pass	Pass	Pass
3	Pass	Pass	Pass	Pass
4	Pass	Pass	Pass	Pass
5	Pass	Pass	Pass	Pass
6	Pass	Pass	Pass	Pass
7	Pass	Pass	Pass	Pass

4.5 Triển khai

Để phục vụ mục đích mở rộng hệ thống hơn sẽ được trình bày ở chương 6, em đã thử sử dụng với nhiều hệ thống máy chủ như share hosting, VPS và AWS. Sau khi sử dụng thử AWS em đã quyết định sử dụng AWS để lưu trữ database và làm máy chủ của hệ thống, cụ thể là dịch vụ AWS RDS và Elastic Beanstalk. Sử dụng các dịch vụ AWS sẽ phục vụ dễ dàng cho hệ thống đơn người dùng, với 1 người/2 máy chủ. AWS cũng hỗ trợ dễ dàng cho việc nâng cấp và mở rộng hệ thống trong tương lai.



Hình 4.21 Sơ đồ triển khai của máy chủ AWS

Cấu hình của AWS RDS em chọn là 1vCPUs và 1GiB RAM bộ nhớ phân bổ là 20GiB và không autoscaling, gói sử dụng là Free Tier.

Cấu hình của AWS Elastic beanstalk sẽ tự được AWS cài đặt khi tải source code lên.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Trong chương 4 em đã giới thiệu một cách tổng quan về các thành phần cũng như kiến trúc triển khai, để mọi người có cái nhìn bao quát hơn về hệ thống trong đồ án tốt nghiệp này. Sang chương 5 em xin giới thiệu những tính năng nổi bật nhất, cũng như những đóng góp được đưa vào ứng dụng để xây dựng hệ thống.

5.1 Khó khăn khi thu thập dữ liệu từ các sàn TMDT

5.1.1 Vấn đề gặp phải

Khi thực hiện đồ án này, em đã tìm hiểu và biết rằng các sàn TMDT đều có gói API dành cho lập trình viên, cung cấp khá đầy đủ dữ liệu, em đã thử xin các sàn TMDT cấp cho API phục vụ việc lập trình các ứng dụng liên quan tới TMDT, các API cần bao gồm thông tin cá nhân người dùng, API danh sách sản phẩm, API danh sách đơn hàng, API cập nhật tồn kho, API in vận đơn, API kiểm tra số dư ví ...

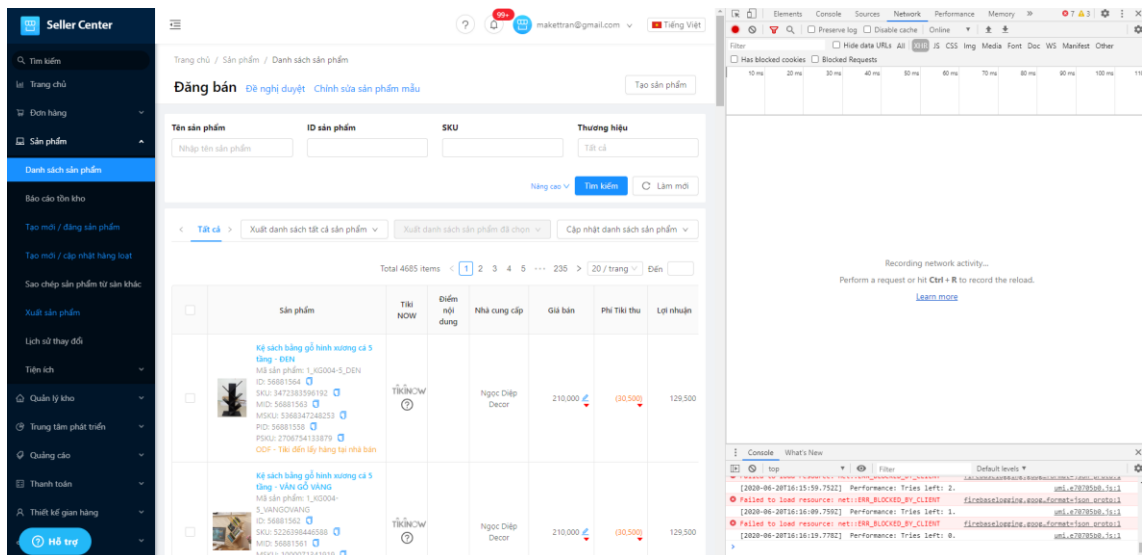
Tuy nhiên vì chỉ là một cá nhân nên em đã không được chấp thuận việc sử dụng API lập trình của họ. Vì vậy đây chính là một khó khăn rất lớn lúc muốn thực hiện đồ án, tuy nhiên cũng là một gợi ý để em hiểu ra vì sao nhiều sản phẩm khác trên thị trường không có đủ cả 4 sàn TMDT lớn nhất hiện nay, vì họ cũng không xin được đủ API của cả 4 sàn.

Em đã chuyển sang hướng tìm API của các sàn để phục vụ cho đồ án, các API này là API hiển thị dữ liệu hoặc API cập nhật dữ liệu. Hướng đi này tuy không cần sự phê duyệt dùng API của các sàn TMDT nhưng mỗi khi các sàn thay đổi API thì sẽ cần cập nhật lại hệ thống, tuy nhiên API là một thứ các sàn TMDT rất ít khi thay đổi lớn.

5.1.2 Giải pháp đưa ra

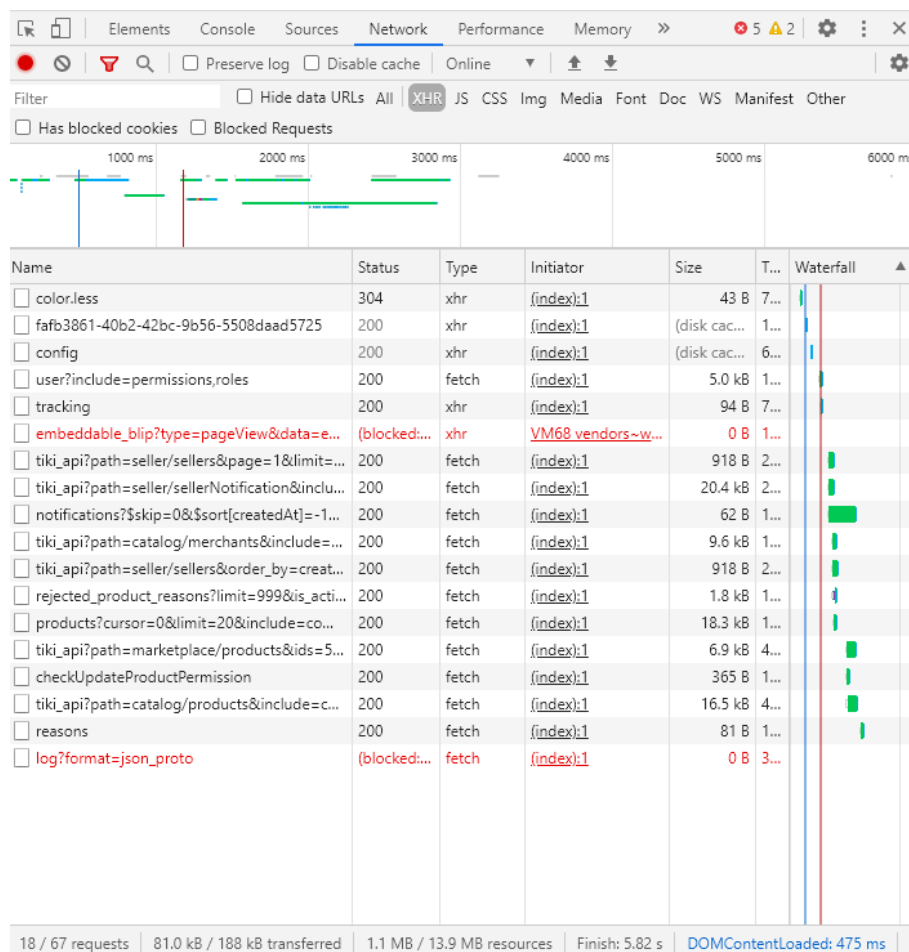
Dưới đây em sẽ trình bày cách lấy API của một sàn là Tiki, các sàn còn lại cũng làm tương tự.

Đầu tiên, đăng nhập vào trang quản lý bán hàng của Tiki là: <https://sellercenter.tiki.vn/>. Ví dụ cần lấy API danh sách sản phẩm. Em chọn “Danh sách sản phẩm” trên menu và mở Chrome DevTools (F12), chọn sang tab Network.



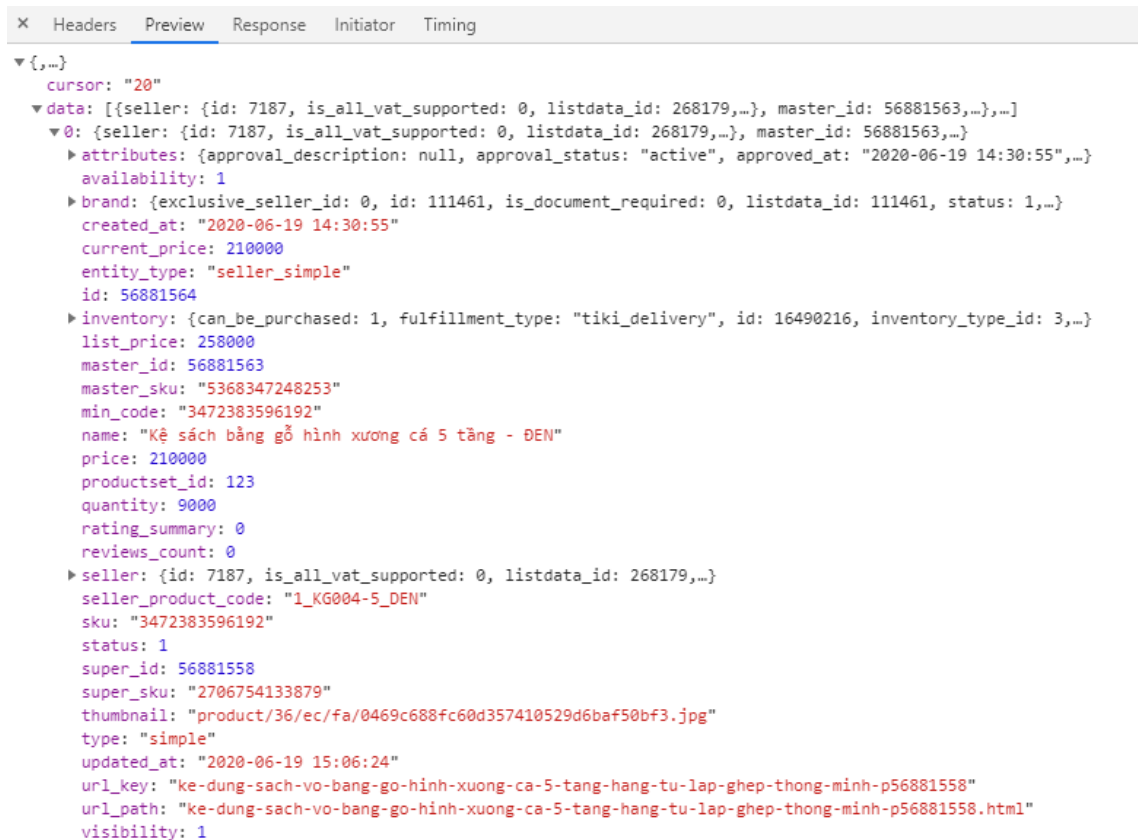
Hình 5.1 Cách lấy API của Tiki (1)

Nhấn nút clear để xóa hết các bản ghi nếu có, rồi nhấn Ctrl + R để tải lại trang. Chuyển sang tab XHR cạnh tab ALL để chỉ load ra các đối tượng XMLHttpRequest.



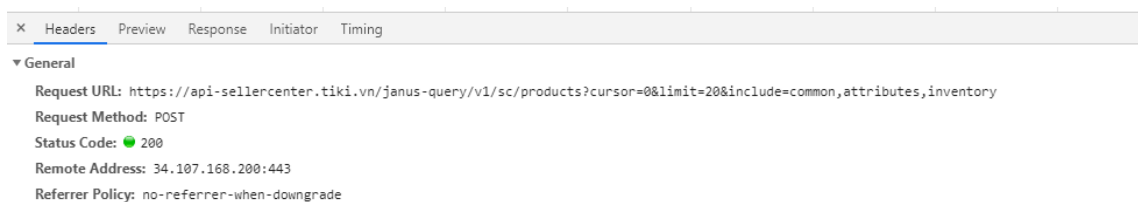
Hình 5.2 Cách lấy API của Tiki (2)

Em thử “*products?cursor=0&limit=20&include=common,attributes,inventory*” để tìm API danh sách sản phẩm. Chọn sang tab Preview. Em thấy được đây chính là dữ liệu trả về của danh sách sản phẩm mà em cần tìm.



Hình 5.3 Cách lấy API của Tiki (3)

Em quay trở lại tab Headers, vậy là em đã lấy được endpoint của API và method của API.



Hình 5.4 Cách lấy API của Tiki (4)

Tiếp theo em cần đi tìm data header và body của API. Vẫn ở tab Header, em kéo xuống mục Request Headers. Trong đó em sẽ chú ý tới trường authorization, em thấy API được bảo vệ bằng JWT Token và trường x-source có giá trị là janus-query-prod.


```

▼ Request Headers
authority: api-sellercenter.tiki.vn
method: POST
path: /janus-query/v1/sc/products?cursor=0&limit=20&include=common,attributes,inventory
scheme: https
accept: application/json
accept-encoding: gzip, deflate, br
accept-language: vi-VN,vi;q=0.9,en-US;q=0.8,en;q=0.7,fr-FR;q=0.6,fr;q=0.5
authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJyOTAsImZlcy16IiwiaWF0IjoiMTY4NjY5MzAudG1raS5zZXJ2aWwucy9hcGkvZD9rZW4vcnVxdWVzdC1sIm1hdC16MTU5MjUzODM3CwI2XhwiJ3oxMTkzMTczMTcwLjUyYyY1OjE1OT
content-length: 146
content-type: application/json; charset=utf-8
origin: https://sellercenter.tiki.vn
referer: https://sellercenter.tiki.vn/new/
sec-fetch-dest: empty
sec-fetch-mode: cors
sec-fetch-site: same-site
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.106 Safari/537.36
x-source: janus-query-prod

```

Hình 5.5 Cách lấy API của Tiki (5)

Cuối cùng, em cần tìm body của API, em xem mục Request Payload trong tab Headers và có được body của API.

```

▼ Request Payload view source
{seller_ids: [7187], ids: [], skus: [], brand_listdata_id: [], productset_ids: [], master_ids: [],...}
  brand_listdata_id: []
  entity_type: "seller_simple"
  ids: []
  master_ids: []
  master_skus: []
  productset_ids: []
  ▶ seller_ids: [7187]
  skus: []

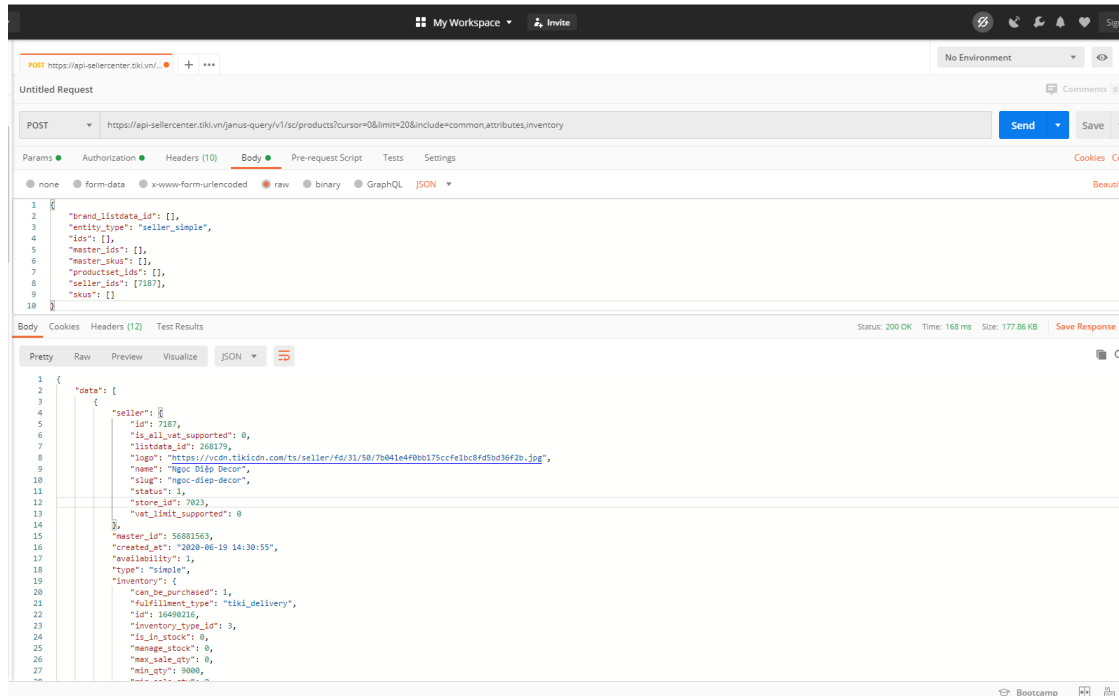
```

Hình 5.6 Cách lấy API của Tiki (6)

Trong body có những trường sau, brand_listdata_id: [], entity_type: “seller_simple”, ids: [], master_ids: [], master_skus: [], productset_ids: [], seller_ids: [7187], skus[]. Trong đó seller_ids có giá trị là 7187 là id của nhà bán hàng được tìm được khi gọi API lấy thông tin người dùng. API lấy thông tin người dùng cũng được tìm tương tự như trên với endpoint là “https://sellercenter.tiki.vn/api/user?include=permissions,roles” và method là GET, được bảo mật bằng JWT token và cookie.

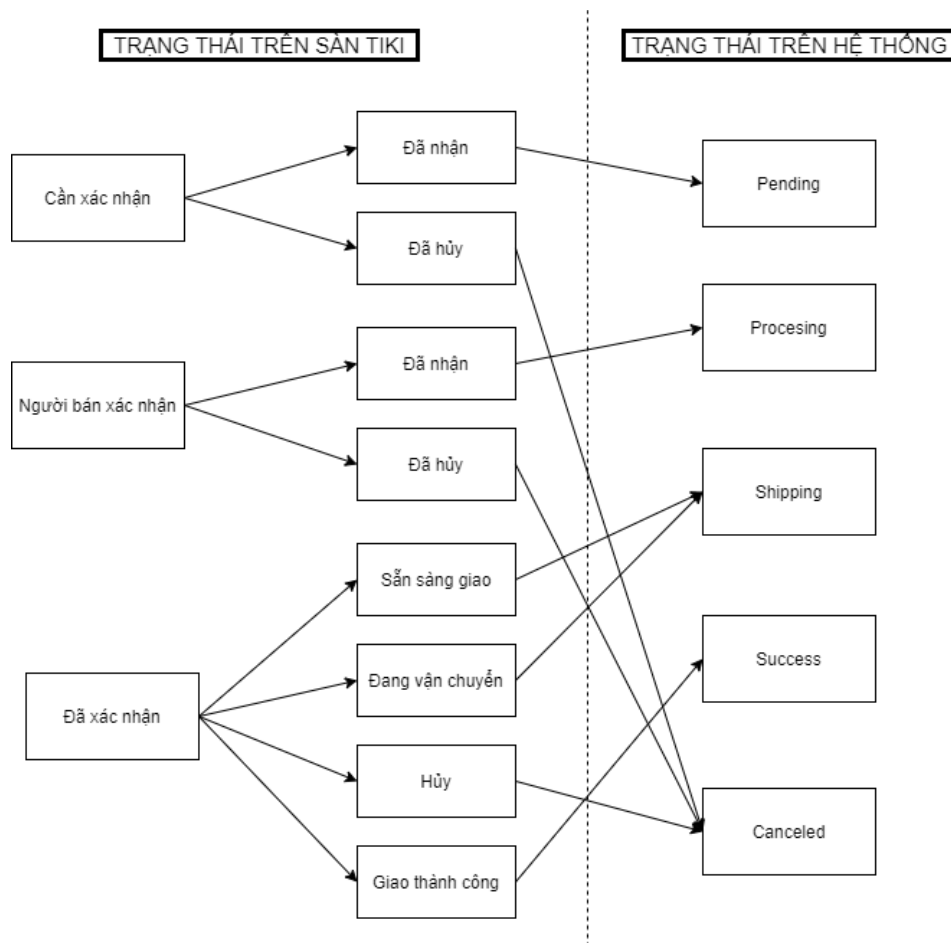
5.1.3 Kết quả đạt được

Sau khi có đủ các thông tin cần cho API lấy danh sách sản phẩm, em sẽ dùng POSTMAN để thử lấy danh sách sản phẩm. Em chọn method, điền endpoint, headers, body dạng json và authorization là Bearer Token và đã lấy được danh sách sản phẩm ở dạng Json.



Hình 5.7 Lấy thành công danh sách sản phẩm bằng Postman

Việc lấy danh sách sản phẩm khá dễ dàng, nhưng lấy danh sách đơn hàng sẽ phức tạp hơn ở chỗ để lấy trạng thái đơn hàng, em cần có một lượng lớn đơn hàng để nắm được cách đặt tên cho trạng thái của đơn hàng của từng sàn rồi đồng bộ nó với hệ thống của em. Khó khăn nhất ở đơn hàng của Tiki khi có 2 lớp trạng thái. Sau khi mượn được tài khoản của một nhà bán hàng có một số lượng đơn lớn và có đơn mới liên tục em đã tìm ra được cách đặt trạng thái cho đơn hàng của tất cả các sàn. Dưới đây là mô hình đồng trạng thái của Tiki thành trạng thái của hệ thống.



Hình 5.8 Đồng bộ trạng thái đơn hàng giữa sàn Tiki và hệ thống

5.2 Chuyển đổi luật của sàn TMDT thành cảnh báo vận chuyển

5.2.1 Vấn đề gặp phải

Mỗi sàn TMDT có bộ quy tắc bán hàng dành cho các nhà bán hàng riêng. Việc người bán hàng xác nhận đơn và đóng gói giao cho đơn vị vận chuyển là điều vô cùng cẩn thận để các sàn TMDT có thể giao hàng đến người tiêu dùng đúng thời hạn đã hẹn.

Vì vậy các sàn TMDT đều có mức phạt với người bán hàng nếu không giao hàng cho đơn vị vận chuyển. Cụ thể như sau:

Sàn Tiki: thời gian tối đa là 48 giờ, mức phạt 200.000đ/sản phẩm.

Sàn Lazada: thời gian tối đa là 48 giờ, mức phạt 200.000đ/sản phẩm, có thể bị hủy đơn hàng.

Sàn Shopee: thời gian 2 ngày với đơn hàng bình thường và 7 ngày với đơn hàng đặt trước, mức phạt sẽ là bị phạt điểm trong hệ thống Sao quả tạ dựa trên tỉ lệ đơn hàng giao trễ.

Sàn Sendo: nhà vận chuyển sẽ lấy hàng 3 lần trong 3 ngày, nếu không giao hàng sẽ bị hủy đơn hàng đó.

5.2.2 Giải pháp đưa ra

Sau khi tìm hiểu được mốc thời gian đó, tính năng cảnh báo vận chuyển được xây dựng với quy trình như sau.

Đơn hàng sau khi được đồng bộ về. Nếu sau 24 giờ kể từ khi đồng bộ về đơn hàng chưa được thay đổi trạng thái nội bộ (`local_status`) thì sẽ được đưa vào phần cảnh báo vận chuyển.

Đơn hàng sau khi được đổi trạng thái sẽ được gỡ khỏi phần cảnh báo vận chuyển.

5.2.3 Kết quả đạt được

Người dùng có thể dễ dàng biết được đơn hàng nào mình chưa vận chuyển cho nhà vận chuyển để tránh được việc bị phạt bằng cách truy cập màn hình Cảnh báo vận chuyển hoặc màn hình báo cáo.

5.3 Kết nối sản phẩm trong kho hàng và sản phẩm trên sàn và tính toán lượng hàng tồn kho.

5.3.1 Vấn đề gặp phải

Kết nối sản phẩm trong kho hàng và sản phẩm trên sàn là tính năng quan trọng bậc nhất của hệ thống, nó giúp tính năng đồng bộ tồn kho giữa các sàn hoạt động được. Việc kết nối như thế nào để chính xác và thuận tiện cho người dùng là một bài toán cần tính toán kỹ. Tiếp theo là vấn đề một sản phẩm trong kho không kết nối 1-1 với sản phẩm trên sàn, mà một sản phẩm trong kho có thể xuất hiện ở nhiều sản phẩm trên cùng một gian hàng, hoặc xuất hiện nhiều lần trên cùng một sản phẩm trên cùng một gian hàng ở cùng một sàn. Ví dụ sản phẩm *khung tranh hoa mai* có thể nằm trong sản phẩm *khung tranh hoa mai*, *3 khung tranh hoa mai* và *bộ tranh tứ quý*. Một vấn đề nữa là một sản phẩm có nhiều phân loại khác nhau, làm sao để phân biệt các sản phẩm đó để kết nối với đúng sản phẩm trong kho, vì sản phẩm màu xanh hay màu đỏ là một sản phẩm riêng biệt trong kho nhưng khi bán sẽ tính là 1 sản phẩm nhưng có 2 phân loại là xanh và đỏ.

Với tồn kho của sản phẩm sẽ xảy ra vấn đề có nhiều đơn hàng chưa phê duyệt cùng có một sản phẩm, dẫn tới sản phẩm trong kho tuy chưa bị trừ nhưng đã không đủ cho tất cả số đơn mới. Vấn đề này các hệ thống khác chưa giải quyết triệt để dẫn đến khi kiểm kê kho vẫn còn hàng, nhưng trên hệ thống báo hết hàng và người dùng không hiểu việc tính toán tồn kho bị sai ở đâu, đó là do các hệ thống khác tự trừ số lượng tồn kho khi có đơn hàng mới và cộng thêm vào khi đơn đó bị hủy.

Với những sản phẩm trên sàn dạng combo của nhiều loại sản phẩm trong kho. Người bán phải tự tính toán số lượng tồn kho để nhập vào các sàn dựa trên số lượng combo có thể có từ các sản phẩm trong kho, người bán cần một giải pháp để giải quyết vấn đề đó.

5.3.2 Giải pháp đưa ra

Đã có rất nhiều phương án được đưa ra để kết nối sản phẩm. Em sẽ trình bày một vài phương án đã được tính tới như sau:

1. Trong danh sách sản phẩm trong kho, người dùng chọn kết nối và nhập mã SKU các sản phẩm trên sàn sẽ kết nối, chọn sàn của sản phẩm và nhập số lượng.

2. Người dùng vào danh sách sản phẩm trên sàn, chọn sản phẩm đó bao gồm những sản phẩm nào trong kho với số lượng bao nhiêu.

3. Người dùng vào danh sách sản phẩm trong kho, chọn sản phẩm cần kết nối, hiển thị danh sách sản phẩm trên sàn, chọn sản phẩm trên sàn mà sản phẩm trong kho có xuất hiện cùng với số lượng sản phẩm trong kho có trong đó.

Sau khi phân tích và thử nghiệm với các phương án trên, em đã quyết định sử dụng phương án 3 để kết nối sản phẩm.

Để giải quyết vấn đề về số lượng tồn kho sản phẩm khi có quá nhiều đơn hàng mới về cùng một sản phẩm. Phương án em đưa ra là lượng tồn kho ảo, khi có đơn hàng mới, lượng tồn kho ảo sẽ tự trừ sản phẩm đi dù đơn hàng chưa được phê duyệt, lượng tồn kho ảo này chỉ phục vụ việc đồng bộ số lượng sản phẩm để cho các đơn đặt hàng mới không làm số lượng sản phẩm cần vượt quá số lượng sản phẩm trong kho có thể có.

Với việc tính toán lượng sản phẩm combo, các combo sẽ được tính số lượng tồn kho dựa trên các sản phẩm thành phần, các sản phẩm thành phần sẽ chia cho số lượng sản phẩm đó trong kết nối, so sánh phần nguyên của kết quả phép chia và lấy số bé nhất làm số lượng tồn kho cho combo.

5.3.3 Kết quả đạt được

Sau khi thực hiện theo giải pháp đề ra ở trên thì việc kết nối sản phẩm theo phương án 3 nhanh chóng và dễ sử dụng hơn phương án 1 và 2. Tuy nhiên người dùng vẫn phải kết nối thủ công tất cả sản phẩm trong kho. Việc tính toán số lượng combo đã chính xác và tự động, người bán hàng không cần tính toán thủ công nữa. Số lượng tồn kho ảo giúp cập nhật số hàng còn lại có thể nhận đơn, còn số lượng tồn kho sẽ giúp thống kê số sản phẩm trong kho phục vụ cho việc kiểm kê kho.

5.4 Tối giản tính năng Hoàn hàng/Xuất kho

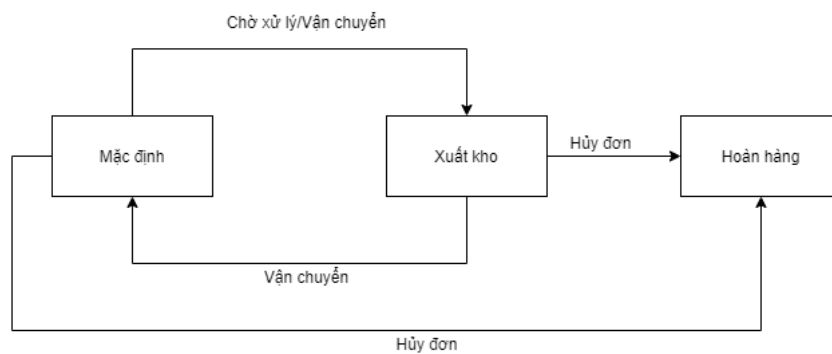
5.4.1 Vấn đề gặp phải

Khi xây dựng tính năng hoàn hàng/xuất kho, em đã xác định tính năng này phải tối giản, dễ sử dụng, nhanh gọn vì số lượng đơn hàng của 1 nhà bán hàng một ngày rất lớn, nếu hệ thống xây dựng ra mà khiến quy trình bán hàng rườm rà hơn thì hệ thống không có ý nghĩa. Các trạng thái nội bộ của đơn hàng ban đầu bao gồm hoàn hàng và xuất kho, nhưng một vấn đề phát sinh trong quá trình làm đó là hàng tạm hoàn (do giao hàng không thành công) và hàng hoàn lại do đơn hàng bị hủy.

5.4.2 Giải pháp đưa ra

Để giải quyết vấn đề quét mã của nhiều đơn hàng, em xây dựng tính năng với 2 khả năng nhập liệu là nhập máy và nhập tay. Trong đó, nhập tay cho phép người dùng nhập nhiều đơn hàng một lúc. Còn nhập máy tận dụng máy quét mã vạch của các nhà bán hàng, kết nối với máy tính và quét mã vận đơn của đơn hàng để thay đổi trạng thái của đơn.

Việc đơn hàng phân ra đơn hàng tạm hoàn và hoàn do đơn bị hủy yêu cầu cần phân ra 3 trạng thái nội bộ của đơn là Mặc định, Xuất kho, Hoàn hàng. Tuy nhiên để cho việc sử dụng dễ dàng. Sẽ chỉ phân ra 2 loại quét mã là Xuất kho và hoàn hàng. Hệ thống sẽ dựa vào trạng thái hiện tại của đơn hàng để quyết định đây là hàng hoàn tạm thời hay hàng hoàn do đơn bị hủy. Nếu đơn hàng chưa được quét mã, sẽ ở trạng thái nội bộ là Mặc định, sau khi được quét Xuất kho nó sẽ chuyển thành Xuất kho. Nếu đơn hàng vẫn ở trạng thái vận chuyển, mà được quét mã Hoàn hàng, thì sẽ được tính là hàng tạm hoàn, trạng thái lúc này là Mặc định. Nếu đơn hàng ở trạng thái hủy, mà được quét mã Hoàn hàng, sẽ được tính là hàng hoàn do đơn bị hủy, trạng thái lúc này sẽ là Hoàn hàng.



Hình 5.9 Mô tả quy trình thay đổi trạng thái nội bộ

5.4.3 Kết quả đạt được

Tính năng hoàn hàng/xuất kho sau khi được xử lý theo hướng ở trên đã trở nên đơn giản, dễ dàng sử dụng hơn cho người dùng. Quy trình quét mã vận đơn cũng trở nên chi tiết hơn giúp việc tổng kết và thống kê chuẩn xác hơn.

Vậy là em đã trình bày khó khăn và phương án giải quyết của em trong chương 5 này. Tiếp theo ở chương cuối cùng, chương 6, em sẽ kết luận và nêu lên định hướng phát triển của đề án này trong tương lai.

CHƯƠNG 6. KẾT LUẬN VÀ PHÁT TRIỂN

Để thực hiện hết ý tưởng và mong muốn của người dùng trong một đồ án là một điều bất khả thi, vì vậy trong chương 6 này, em sẽ kết luận lại những vấn đề đã làm được và đặt ra định hướng phát triển cho hệ thống trong tương lai.

6.1 Kết luận

Việc TMDT phát triển dẫn tới có rất nhiều dịch vụ ăn theo hỗ trợ việc bán hàng, đây là một thị trường rất tiềm năng, rất nhiều công ty phát triển những dịch vụ đó đã thành công kiếm được doanh thu vô cùng lớn. Nhưng khi phát triển càng lớn họ càng xa rời chính những nhu cầu thực tế của người dùng. Hệ thống của em chỉ là một trong rất nhiều hệ thống hỗ trợ cho TMDT và còn rất nhiều thiếu sót, nhưng em luôn thực hiện theo tiêu chí làm theo những gì người dùng cần trước.

Tuy nhiên, trong phạm vi một đồ án khó có thể thực hiện hết những gì người dùng cần, vì thế em đã ưu tiên thực hiện những chức năng cốt lõi mà những hệ thống quản lý dữ liệu đa sàn TMDT nào cũng có như đồng bộ đơn hàng, đồng bộ kho, tạo báo cáo. Ngoài ra cũng có những tính năng mới như cảnh báo đơn hàng, quét mã vận đơn bằng máy quét mã vạch. Hệ thống cũng được tạo ra mới mục đích cho người dùng thực tế sử dụng, mỗi người sẽ được sử dụng một máy chủ riêng để phục vụ việc có thể có những tính năng riêng phù hợp với nhu cầu từng người. Chi phí sử dụng dự kiến cũng rẻ hơn các hệ thống khác.

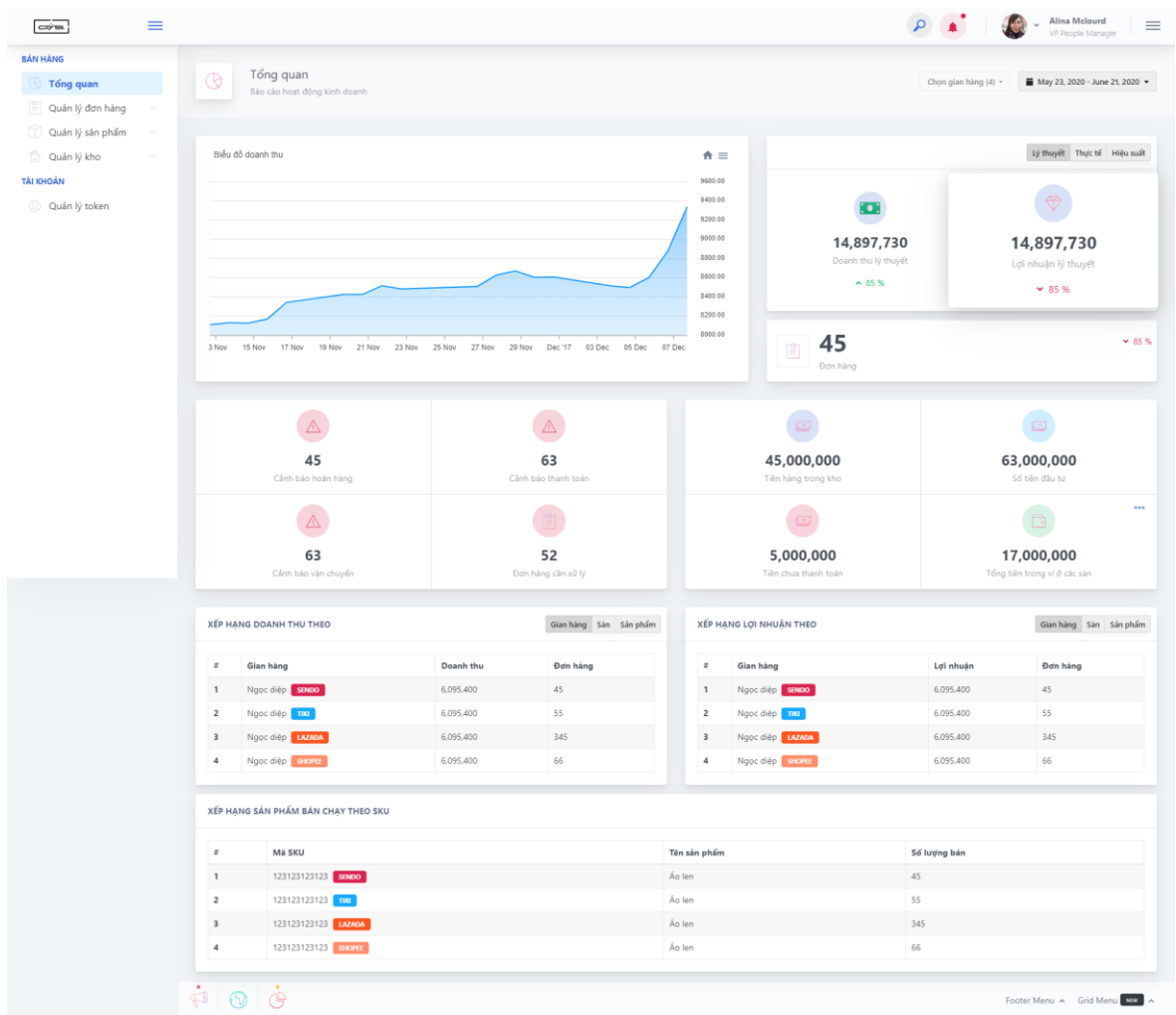
Do nhiều hạn chế về thời gian và năng lực nên hệ thống này chắc chắn còn tồn tại rất nhiều điểm hạn chế. So với các hệ thống đã có thì những điểm hạn chế có thể liệt kê như giao diện như đẹp và hiện đại, chưa có nhiều loại biểu đồ báo cáo và khả năng dự đoán doanh thu và sản phẩm bán chạy trong tương lai, tính năng phê duyệt đơn hàng và in vận đơn chưa hoàn chỉnh, thời gian bảo trì để cập nhật API của các sàn cũng lâu hơn bình thường, muốn sử dụng hệ thống phải đợi cài đặt máy chủ riêng, chưa có đồng bộ dữ liệu từ các nền tảng bán hàng offline hoặc online khác.

Quá trình thực hiện đồ án tốt nghiệp này em đã học hỏi được nhiều điều từ thầy cô, bạn bè không chỉ là về kiến thức chuyên ngành mà còn cả về những khả năng khác như khả năng phân tích yêu cầu người dùng, khả năng giao tiếp với người dùng, tầm quan trọng của việc khảo sát người dùng, các kỹ năng trong việc thương mại hóa một sản phẩm phần mềm cũng như rút ra được cho mình nhiều kinh nghiệm quý báu, có thể sẽ giúp ích được cho bản thân trong cuộc sống cũng như công việc sau này. Đồng thời giúp em nâng cao khả năng xây dựng những hệ thống tương tự trong tương lai.

6.2 Hướng phát triển

Với những thiếu sót đã được nhắc ở phần trước, và để sản phẩm có thể chính thức thương mại, trong tương lai em sẽ đặt mục tiêu hoàn thiện các tính năng chưa được ổn định và xây dựng thêm những tính năng mới như:

1. Thêm nhiều loại báo cáo hỗ trợ người dùng việc quản lý chi phí đầu vào, chi phí nhân công và doanh thu.
2. Xây dựng thành công tính năng phê duyệt đơn hàng và in vận đơn dựa trên những phần đã tìm hiểu trong quá trình làm đồ án tốt nghiệp.
3. Thêm các nền tảng bán hàng khác như Facebook, bán hàng offline. Nguyên lý hoạt động cũng tương tự như đó là 1 sàn thương mại điện tử.
4. Chuẩn bị cho việc sáp nhập Tiki và Sendo, sẽ có những sự thay đổi về mặt công nghệ của 2 sàn này, và một trong 2 sàn sẽ biến mất.
5. Xây dựng thêm trang landing page giúp người dùng mới đăng ký thông tin sử dụng hệ thống và thanh toán chi phí sử dụng.
6. Nghiên cứu phát triển đồng bộ chat với khách hàng và bình luận của khách hàng
7. Chuyển hệ thống sang giao diện mới hiện đại hơn.



Hình 6.1 Giao diện báo cáo kinh doanh mới

Hoàn hàng - Xuất kho
Quản lý đơn hàng

XUẤT HÀNG

MÃ ĐƠN HÀNG:
Nhập mã đơn hàng vào đây

Tải về dạng excel

Show 10 entries

STT	Thông báo	Mã đơn hàng	Nguồn đơn hàng	Trạng thái	Ngày tạo
-----	-----------	-------------	----------------	------------	----------

Hình 6.2 Giao diện Hoàn hàng/Xuất kho mới

Kết thúc chương 6 cũng là kết thúc nội dung chính của đề án tốt nghiệp này, em đã trình bày kết luận về đề án, chỉ ra những điểm thiếu sót và định hướng trong tương lai với mong muốn đưa hệ thống ra thương mại. Với việc có hơn 20 nhà bán hàng đang chờ đợi hệ thống này được đưa vào triển khai, hi vọng nó sẽ đáp ứng được nhu cầu thực tế của thị trường.

TÀI LIỆU THAM KHẢO

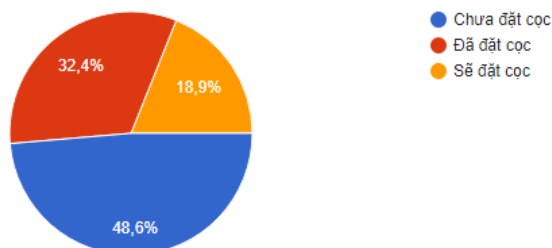
- [1] Laravel document: <https://laravel.com/docs/7.x>
- [2] AWS document: <https://docs.aws.amazon.com/>
- [3] AdminLTE document: <https://adminlte.io/themes/AdminLTE/documentation/>
- [4] Tài liệu bán hàng Tiki: <https://hocvien.tiki.vn/tai-lieu-tham-khao/>
- [5] Lazada seller center support: <https://lazadavnpsc.zendesk.com/>
- [6] Chính sách người bán Sendo: <https://ban.sendo.vn/chinh-sach-nguoi-ban/>
- [7] Hướng dẫn bán hàng cùng Shopee: <https://banhang.shopee.vn/edu/>

PHỤ LỤC

A1. Kết quả khảo sát nhà bán hàng

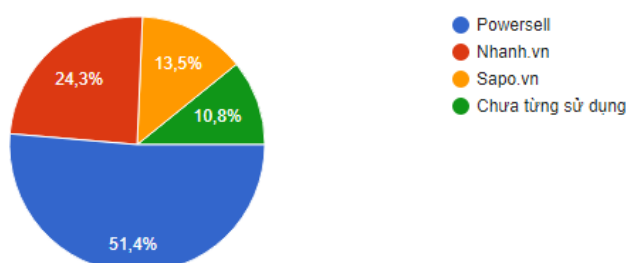
Bạn đã đặt cọc sẵn để sử dụng hệ thống chưa?

37 câu trả lời



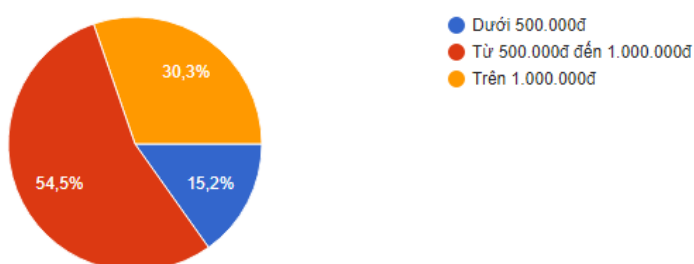
Bạn đã từng sử dụng hệ thống quản lý dữ liệu đa sàn TMDT nào?

37 câu trả lời



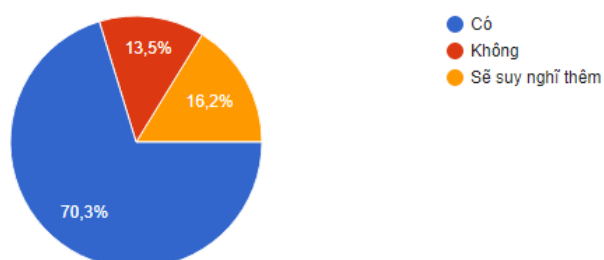
Số tiền bạn phải bỏ ra cho hệ thống quản lý dữ liệu đa sàn TMDT 1 tháng là bao nhiêu?

33 câu trả lời



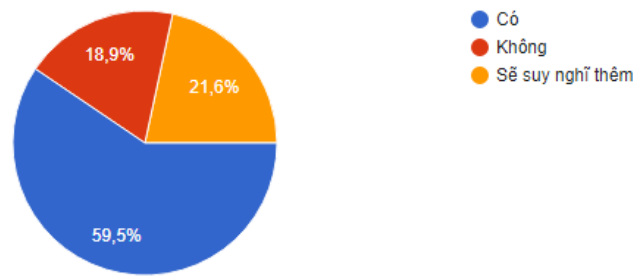
Nếu có một hệ thống bán theo gói nhiều năm với giá bằng giá thuê 1 năm của các hệ thống khác và tính phí cập nhật bạn có muốn sử dụng không?

37 câu trả lời



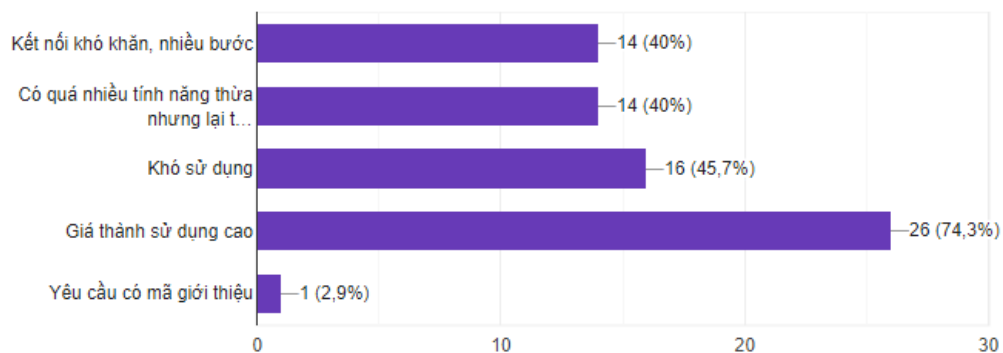
Bạn có sẵn sàng bỏ thêm tiền để có những tính năng riêng cho hệ thống.

37 câu trả lời



Bạn thấy những hệ thống bạn đã sử dụng có nhược điểm gì

35 câu trả lời



Hãy chọn những tính năng bạn cho rằng cần thiết nhất để hỗ trợ cho hệ thống quản lý dữ liệu đa sàn TMDT

37 câu trả lời

