**BỘ CÔNG THƯƠNG**

**TRƯỜNG CAO ĐẲNG CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG WEB BÁN MỸ PHẨM HOÀNG VŨ**

**Sinh viên thực hiện:**

Nguyễn Hoàng Vũ

2121110317 - CCQ2111J

Trần khánh Hằng

2121110309 - CCQ2111I

**Giảng viên hướng dẫn: ThS. Hồ Diên Lợi**

TP. Hồ Chí Minh – Tháng 06/2024

# LỜI NHẬN XÉT

# LỜI CẢM ƠN

Lời đầu tiên, xin trân trọng cảm ơn thầy Hồ Diên Lợi đã tận tình hướng dẫn em trong quá trình học tập cũng như trong việc hoàn thành đồ án.

Xin chân thành cảm ơn các Thầy, Cô thuộc khoa Công Nghệ Thông Tin cũng như những khoa đang công tác tại trường Cao Đẳng Công Thương TP.HCM đã tận tình giảng dạy cho em trong thời gian học tập.

Chúng em cũng xin gửi cảm ơn chân thành nhất tới các thầy cô giáo trong Trường Cao Đẳng Công Thương TP.HCM nói chung, các thầy cô trong Khoa Công nghệ thông tin nói riêng đã truyền đạt cho chúng em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình em tham gia học tập.

Cuối cùng, chúng em xin chân thành cảm ơn gia đình và bạn bè, những người luôn ở bên cạnh đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên chúng em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp.

Với điều kiện về thời gian cũng như lượng kiến thức về đề tài rất rộng mà kinh nghiệm còn hạn chế của một sinh viên, đề án này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao kiến thức của mình, phục vụ tốt hơn công tác thực tế sau này.

TP Thủ Đức, ngày …. tháng …. năm 2024

Sinh viên thực hiện

# MỤC LỤC

[**LỜI NHẬN XÉT i**](#_heading=h.30j0zll)

[**LỜI CẢM ƠN ii**](#_heading=h.3znysh7)

[**MỤC LỤC iii**](#_heading=h.2et92p0)

[**Chương 1.**](#_heading=h.3dy6vkm) **TỔNG QUAN VỀ WEB BÁN MỸ PHẨM 1**

[**1.1**](#_heading=h.1t3h5sf) **Lý do chọn đề tài 1**

[**1.2 Mục đích nghiên cứu 1**](#_heading=h.4d34og8)

[**1.3 Phương pháp nghiên cứu 1**](#_heading=h.2s8eyo1)

[**1.4 Đối tượng nghiên cứu 1**](#_heading=h.17dp8vu)

[**1.5 Bố cục đề tài 1**](#_heading=h.3rdcrjn)

[**Chương 2-**](#_heading=h.26in1rg) **CƠ SỞ LÝ THUYẾT 2**

[**2.1 ReactJS 2**](#_heading=h.lnxbz9)

[2.1.1](#_heading=h.35nkun2) Giới thiệu tổng quan về ReactJs 2

[2.1.2](#_heading=h.44sinio) Các tính năng chính của ReactJS 2

[2.1.3](#_heading=h.z337ya) Ưu điểm khi sử dụng ReactJS 4

[2.1.4](#_heading=h.3j2qqm3) Nhược điểm khi sử dụng ReactJS 4

[**2.2 Restful API 5**](#_heading=h.4i7ojhp)

[2.2.1](#_heading=h.2xcytpi) Tổng quan về Restful API 5

[**2.3 Tìm hiểu về JAVASCRIPT 6**](#_heading=h.2bn6wsx)

[2.3.1](#_heading=h.qsh70q) Tổng quan về Javascript 6

[2.3.2](#_heading=h.3as4poj) Cách thức hoạt động của Javascript 6

[2.3.3](#_heading=h.1pxezwc) Đặc điểm của JavaScript 7

[**2.4 Node.js 7**](#_heading=h.2p2csry)

[2.4.1](#_heading=h.147n2zr) Khái niệm về Node.js 7

[2.4.2](#_heading=h.23ckvvd) Một số ưu điểm khi sử dụng NodeJS 8

[**2.5 Tìm hiểm về Express Framework 10**](#_heading=h.32hioqz)

[2.5.1](#_heading=h.1hmsyys) Giới thiệu về Express Framework 10

[**2.6 Hệ quản trị cơ sở dữ liệu MongoDB 10**](#_heading=h.2grqrue)

[2.6.1](#_heading=h.vx1227) Giới thiệu về MongoDB 10

[2.6.2](#_heading=h.3fwokq0) Thuật ngữ sử dụng trong MongoDB 11

[**Chương 3-**](#_heading=h.1v1yuxt) **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 11**

[**3.1 Khảo sát và phân tích hệ thống 12**](#_heading=h.4f1mdlm)

[3.1.1](#_heading=h.2u6wntf) Khảo sát yêu cầu 12

[3.1.2](#_heading=h.19c6y18) Giới thiệu về hệ thống 12

[3.1.3](#_heading=h.3tbugp1) Mục tiêu sử dụng hệ thống 12

[3.1.4](#_heading=h.28h4qwu) Các vai trò của người dùng trong hệ thống 13

[3.1.5](#_heading=h.nmf14n) Các yêu cầu và chức năng của hệ thống 13

[3.1.6](#_heading=h.46r0co2) Các yêu cầu phi chức năng 15

[3.1.7](#_heading=h.2lwamvv) Các chức năng phía giao diện người dùng 16

[3.1.8](#_heading=h.111kx3o) Các chức năng phía giao diện người quản trị 16

[**3.2 Đặc tả và thiết kế hệ thống 17**](#_heading=h.206ipza)

[3.2.1](#_heading=h.4k668n3) Biểu đồ use case tác nhân khách hàng 17

[3.2.2](#_heading=h.1egqt2p) Biểu đồ use case tác nhân quản trị 18

[**3.3 Biểu đồ lớp 18**](#_heading=h.2dlolyb)

[**3.4 Biểu đồ hoạt động 19**](#_heading=h.3cqmetx)

[3.4.1](#_heading=h.1rvwp1q) Biểu đồ hoạt động tổng quát 19

[3.4.2](#_heading=h.1664s55) Biểu đồ hoạt động chức năng đăng nhập, đăng ký 22

[3.4.3](#_heading=h.25b2l0r) Biểu đồ hoạt động chức năng đăng nhập của quản trị viên 22

[3.4.4](#_heading=h.34g0dwd) Biểu đồ hoạt động chức năng quản lý sản phẩm 23

[3.4.5](#_heading=h.43ky6rz) Biểu đồ hoạt động chức năng quản lý đơn hàng 24

[**3.5 Biểu đồ tuần tự 26**](#_heading=h.xvir7l)

[3.5.1](#_heading=h.3hv69ve) Biểu đồ tuần tự Đăng nhập 26

[3.5.2](#_heading=h.4h042r0) Biểu đồ tuần tự Đăng xuất 26

[3.5.3](#_heading=h.1baon6m) Biểu đồ tuần tự Đăng ký 27

[3.5.4](#_heading=h.2afmg28) Biểu đồ tuần tự Xem thông tin tài khoản 28

[3.5.5](#_heading=h.39kk8xu) Biểu đồ tuần tự Thêm sản phẩm vào giỏ hàng 29

[3.5.6](#_heading=h.48pi1tg) Biểu đồ tuần tự Xem giỏ hàng 30

[3.5.7](#_heading=h.1302m92) Biểu đồ tuần tự Cập nhật giỏ hàng 31

[3.5.8](#_heading=h.2250f4o) Biểu đồ tuần tự Đặt hàng 32

[3.5.9](#_heading=h.319y80a) Biểu đồ tuần tự Thêm sản phẩm 33

[3.5.10](#_heading=h.40ew0vw) Biểu đồ tuần tự Cập nhật sản phẩm 34

[3.5.11](#_heading=h.upglbi) Biểu đồ tuần tự Xóa sản phẩm 35

[3.5.12](#_heading=h.1tuee74) Biểu đồ tuần tự Thay đổi tình trạng đơn hàng 36

[**3.6 Tổ chức cơ sở dữ liệu 38**](#_heading=h.2szc72q)

[3.6.1](#_heading=h.184mhaj) Database Diagram 38

[3.6.2](#_heading=h.279ka65) Mô tả chi tiết một số đối tượng chính 39

[**Chương 4-**](#_heading=h.1yyy98l) **TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 43**

[**4.1 Giao diện trang người dùng 43**](#_heading=h.4iylrwe)

[4.1.1](#_heading=h.2y3w247) Trang chủ 43

[4.1.2](#_heading=h.1d96cc0) Đăng nhập 43

[4.1.3](#_heading=h.3x8tuzt) Đăng kí 43

[4.1.4](#_heading=h.2ce457m) Giỏ hàng 43

[4.1.5](#_heading=h.rjefff) Trang bài viết 43

[4.1.6](#_heading=h.3bj1y38) Trang mua hàng 43

[4.1.7](#_heading=h.1qoc8b1) Tìm kiếm 43

[4.1.8](#_heading=h.4anzqyu) Thông tin người dùng 43

[4.1.9](#_heading=h.2pta16n) Thông tin giỏ hàng 43

[4.1.10](#_heading=h.14ykbeg) Trang chi tiết sản phẩm 43

[**4.2 Giao diện trang quản trị 43**](#_heading=h.3oy7u29)

[4.2.1](#_heading=h.243i4a2) Quản lý người dùng 43

[4.2.2](#_heading=h.j8sehv) Quản lý sản phẩm 43

[4.2.3](#_heading=h.338fx5o) Thêm Sản phẩm 43

[4.2.4](#_heading=h.1idq7dh) Quản lý thương hiệu 43

[4.2.5](#_heading=h.42ddq1a) Thêm thương hiệu 43

[**TỔNG KẾT 44**](#_heading=h.wnyagw)

[**Kết quả đạt được 44**](#_heading=h.3gnlt4p)

[**Hạn chế 44**](#_heading=h.1vsw3ci)

[**Hướng phát triển 44**](#_heading=h.4fsjm0b)

[**TÀI LIỆU THAM KHẢO 45**](#_heading=h.1a346fx)

**DANH MỤC HÌNH**

[*Hình 1: ReactJS* 3](#_heading=h.1ksv4uv)

[*Hình 2: Các tính năng của ReactJs* 4](#_heading=h.2jxsxqh)

[*Hình 3: Nhược điểm của ReactJS* 6](#_heading=h.1y810tw)

[Hình 4: *RESTFUL* API 6](#_heading=h.1ci93xb)

[*Hình 5: Các thành phần của RESTFUL API* 7](#_heading=h.3whwml4)

[*Hình 6: JavaScript* 8](#_heading=h.49x2ik5)

[*Hình 7: NodeJs* 9](#_heading=h.3o7alnk)

[*Hình 8: Ưu điểm của NodeJs* 10](#_heading=h.ihv636)

[*Hình 9: Express Framework* 11](#_heading=h.41mghml)

[*Hình 10: Biểu đồ tác nhân khách hàng* 15](#_heading=h.37m2jsg)

[*Hình 11:Biểu đồ tác nhân quản trị* 16](#_heading=h.1mrcu09)

[*Hình 12: Biểu đồ use case tác nhân khách hàng* 19](#_heading=h.2zbgiuw)

[*Hình 13: Biểu đồ tác nhân người quản trị* 19](#_heading=h.3ygebqi)

[*Hình 14: Biểu đồ lớp* 20](#_heading=h.sqyw64)

[*Hình 15: Biểu đồ tổng quát tác nhân khách hàng* 21](#_heading=h.4bvk7pj)

[*Hình 16: Biểu đồ tổng quát tác nhân quản trị* 22](#_heading=h.2r0uhxc)

[*Hình 17: Biểu đồ hoạt động chức năng đăng ký, đăng nhập của khách hàng* 23](#_heading=h.3q5sasy)

[*Hình 18: Biểu đồ hoạt động chức năng của người quản trị* 24](#_heading=h.kgcv8k)

[*Hình 19: Biểu đồ hoạt động chức năng quản lý sản phẩm* 25](#_heading=h.1jlao46)

[*Hình 20: Biểu đồ hoạt động chức năng quản lý đơn hàng* 26](#_heading=h.2iq8gzs)

[*Hình 21: Biểu đồ tuần tự đăng nhập* 27](#_heading=h.1x0gk37)

[*Hình 22: Biểu đồ tuần tự đăng xuất* 28](#_heading=h.2w5ecyt)

[*Hình 23: Biểu đồ tuần tự đăng ký* 29](#_heading=h.3vac5uf)

[*Hình 24: Biểu đồ tuần tự xem thông tin cá nhân* 30](#_heading=h.pkwqa1)

[*Hình 25: Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng* 31](#_heading=h.1opuj5n)

[*Hình 26: Biểu đồ tuần tự xem giỏ hàng* 32](#_heading=h.2nusc19)

[*Hình 27: Biểu đồ tuần tự cập nhật giỏ hàng* 33](#_heading=h.3mzq4wv)

[*Hình 28: Biểu đồ tuần tự đặt hàng* 34](#_heading=h.haapch)

[*Hình 29: Biểu đồ tuần tự thêm sản phẩm* 35](#_heading=h.1gf8i83)

[*Hình 30: Biểu đồ tuần tự cập nhật sản phẩm* 36](#_heading=h.2fk6b3p)

[*Hình 31: Biểu đồ tuần tự xóa sản phẩm* 37](#_heading=h.3ep43zb)

[*Hình 32: Biểu đồ tuần tự thay đổi tình trạng đơn hàng* 38](#_heading=h.4du1wux)

[*Hình 33: Database Diagram* 39](#_heading=h.3s49zyc)

**DANH MỤC BẢNG**

[Bảng 1.Các yêu cầu chức năng phía giao diện người dùng 18](#_heading=h.111kx3o)

[Bảng 2.Các chức năng phía giao diện người quản trị 19](#_heading=h.3l18frh)

[Bảng 3. Bảng thuộc tính product 41](#_heading=h.meukdy)

[Bảng 4. Bảng thuộc tính category 42](#_heading=h.36ei31r)

[Bảng 5. Bảng thuộc tính customer 42](#_heading=h.1ljsd9k)

[Bảng 6. Bảng thuộc tính order 43](#_heading=h.45jfvxd)

[Bảng 7. Bảng thuộc tính cart 43](#_heading=h.2koq656)

[Bảng 8. Bảng thuộc tính brand 44](#_heading=h.zu0gcz)

[Bảng 9. Bảng thuộc tính slider 44](#_heading=h.3jtnz0s)

# Chương 1. TỔNG QUAN VỀ WEB BÁN MỸ PHẨM

## Lý do chọn đề tài

Hiện nay xã hội đã bước và thời kỳ công nghiệp hóa hiện đại hóa thì nhu cầu về công nghệ của con người ngày một tăng cao, nhờ vào sự tiện lợi và phổ biến của nó đã đưa nền kinh tế tăng nhanh và đạt những kết quả tốt nhất

Và việc áp dụng công nghệ thông tin vào việc bán hàng là điều vô cùng hợp lý, khi chúng ta bán hàng online sẽ giúp chúng ta tiếp cận nhiều khách hàng hơn, từ mọi miền đất nước, giúp khách hàng dễ dàng tìm kiếm, so sánh giá cả thị trường mà không cần phải ra khỏi nhà, cũng nhờ công nghệ cũng sẽ giúp sản phẩm của chúng ta được nhiều khách hàng biết đến nhiều hơn. Và việc đặt hàng online giúp cho khách hàng giảm bớt nhiều chi phí và thời gian, việc đặt hàng cũng nhanh chóng và thuận tiện hơn rất nhiều

Áp dụng các kỹ năng thiết kế giao diện và sử dụng các công nghệ như: React, NodeJS, JavaScript để làm một web bán hàng hoàn chỉnh, cũng như học hỏi về các sàn thương mại điện tử nhằm năng cao kỹ năng chuyên môn của bản thân về mảng web

## Mục đích nghiên cứu

Mục đích nghiên cứu nhằm tạo ra một trang web bán hàng hoàn chỉnh, giúp người dùng mua sắm một cách trực tuyến và dễ dàng hơn. Các kết quả và kinh nghiệm khi thực hiện sẽ được sửa đổi và phát triển hơn

## Phương pháp nghiên cứu

Sử dụng các công cụ và mạng xã hội để tìm hiểu và tiếp thu các kiến thức, thu thập các hành vi và hoạt động của khách hàng trên các trang thương mại điện tử khác

Thực hiện các cuộc khảo sát trên các nền tảng mạng xã hội như: FaceBook, Zalo, Tiktok, … để hiểu được các nhu cầu và mong muốn của khách hàng về trang web bán hàng.

Phân tích các nguyên lý thiết kế UI/UX và đặc biệt là tập trung nhất vào người tiêu dùng để tạo cho người dùng cảm nhận được tính linh hoạt và dễ dàng khi sử dụng trang web bán hàng

## Đối tượng nghiên cứu

Đối tượng nghiên cứu chính là các công nghệ: HTML, CSS, JavaScript, ReactJS, NodeJS.

Ứng dụng: MongoDB, Postman, Visual Studio Code, ...

## Bố cục đề tài

Chương 1: Tổng quan về web bán hàng

Chương 2: Cơ sở lý thuyết

Chương 3: Phân tích và thiết kế hệ thống

Chương 4: Triển khai và đánh giá kết quả

Tổng kết

Tài liệu tham khảo

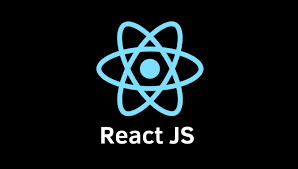
# CƠ SỞ LÝ THUYẾT

## ReactJS

### Giới thiệu tổng quan về ReactJs

ReactJS là một thư viện chứa nhiều JavaScript mã nguồn mở được Facebook xây dựng và phát triển. Thư viện này được sử dụng để tạo ra các ứng dụng trang web hấp dẫn với hiệu quả cao, tốc độ load nhanh và mã tối thiểu. Mỗi web sử dụng ReactJS phải chạy nhanh, mượt và có khả năng mở rộng cao, thao tác thực hiện đơn giản.

Có thể nói, hầu hết các tính năng hay sức mạnh của ReactJS thường bắt đầu từ việc tập trung vào các phần riêng lẻ. Do đó thay vì làm việc trên toàn bộ ứng dụng của web thì ReactJS lại cho phép các Developer phá vỡ giao diện phức tạp của người dùng trở nên đơn giản hơn nhiều. Điều này có nghĩa là các Render dữ liệu không chỉ được thực hiện ở vị trí server mà còn có thể thực hiện ở vị trí Client khi sử dụng ReactJS.



*Hình 1: ReactJS*

### Các tính năng chính của ReactJS

JSX: Viết tắt của JavaScript extension, nó là React extension, giúp cho việc thay đổi cây DOM dễ dàng hơn bằng HTML-style code đơn giản. Nó là một trong những tính năng tốt và dễ sử dụng.

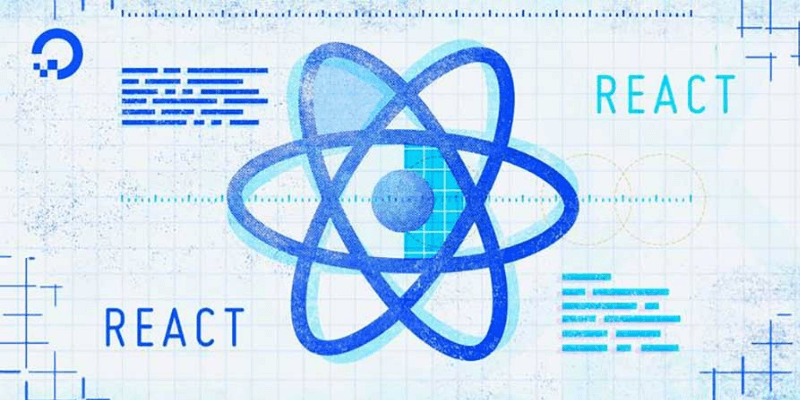
Components: Một trang web được xây dựng bằng ReactJS là một sự kết hợp nhiều component lại với nhau chứ không phải chung một Template như bình thường. Các component cũng như các hàm JavaScript bình thường, giúp tạo ra các code dễ dàng bằng cách tách các logic ra thành các đoạn code độc lập có thể tái sử dụng. Chúng ta có thể sử dụng component dưới dạng function hoặc class, ngoài ra các component còn có state và props.

Redux: Là một phần cực kỳ quan trọng đối với ReactJS và được sử dụng phổ biến. Trong một ReactJS không có các module chuyên dụng để xử lý dữ liệu nên nó được thiết lập một cách độc lập bằng việc chia nhỏ view thành các thành phần khác nhau giúp chúng liên kết chặt chẽ với nhau tốt hơn. Sự liên kết và các mối quan hệ giữa các component trong ReactJS cần được quan tâm đặc biệt bởi chỉ có một luồng dữ liệu duy nhất là luồng dữ liệu từ cha sang con. Việc sử dụng luồng dữ liệu một chiều này có phần khó khăn đối với những người mới tìm hiểu sử dụng và ứng dụng vào các dự án. Bên cạnh các mặt hạn chế, ReactJS có thể phát huy được tất cả chức năng cũng như vai trò của mình trong quá trình sử dụng cơ chế một chiều này. Bởi các chức năng của view trở nên phức tạp hơn rất nhiều.

Virtual Dom: là một phần quan trọng mà hầu như các framework đều sử dụng nó như một ReactJS. Người dùng không cần thao tác trực tiếp trên Dom mà vẫn có thể thấy được view và các thay đổi đó. Bởi Virtual Dom đóng vai trò là một model và kiêm cả vai trò là một view nên sự thay đổi một trong hai nhân tố sẽ kéo theo các nhân tố khác thay đổi. Ngược lại, nếu bạn không thao tác trực tiếp trên các phân tử Dom, bạn vẫn có thể thực hiện được các cơ chế Data Binding.

Javascript Expressions: Biểu thức JS có thể sử dụng trong file.jsx hoặc .js bằng cách sử dụng cặp dấu ngoặc nhọn “{}”

Props và State

Props là một tham số được chuyển qua lại giữa các React Components, các props này được truyền qua các component với cú pháp giống như là HTML attributes.

*Hình 2: Các tính năng của ReactJs*

State là một object mà lưu trữ giá trị của các thuộc tính bên trong components và chỉ tồn tại trong phạm vi của component đó. Mỗi khi bạn thay đổi giá trị của một state thì component đó sẽ được render lại.

### Ưu điểm khi sử dụng ReactJS

ReactJS tạo ra cho bản thân nó một ảo, nơi các Component được tồn tại trên đó và việc tạo ra Dom giúp cải thiện hiệu suất làm việc tốt hơn. Khi bạn cần cập nhật các thông tin lên Dom hoặc thay đổi gì đó, ReactJS đều có thể tính toán trước và thực hiện chúng. Nhờ đó mà ReactJS sẽ tránh được các thao tác cần có trên Dom và không tốn thêm bất cứ hành động nào khác.

Việc viết code trở nên dễ dàng hơn bởi nó sử dụng một cú pháp đặc biệt là JSX, cho phép trộn được giữa code HTML và JavaScript. Bên cạnh đó người dùng có thể sử dụng đoạn code này để thêm vào hàm Render mà không cần thực hiện việc nối chuỗi. Điều này được đánh giá là một trong những đặc tính mới cực kỳ thú vị của ReactJS. Đồng thời, việc chuyển đổi các đoạn HTML thành các hàm khởi động đều được thực hiện từ bộ biến đổi chính là JSX.

 ReactJS có nhiều công cụ phát triển nhờ cài đặt thêm ứng dụng mở rộng của Chrome chuyên sử dụng cho ReactJS. Các lập trình viên có thể debug code một cách dễ dàng hơn, giúp bạn quan sát trực tiếp vào Virtual Dom.

Thư viện ReactJS thân thiện với SEO, được sự hỗ trợ từ các Render, trả về trình duyệt dưới dạng khi bạn chạy ReactJS trên server và Virtual Dom.

### Nhược điểm khi sử dụng ReactJS

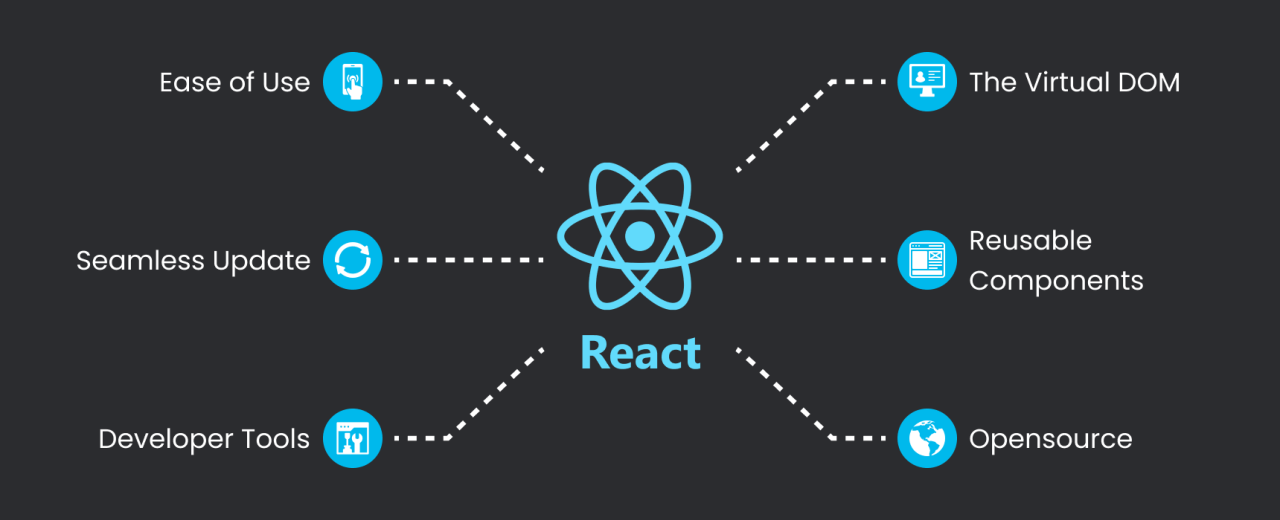
Nhược điểm chính của reactjs đó là:

Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax

Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

ReactJs khá nặng nếu so với các framework khác React có kích thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh

Khó tiếp cận cho người mới học Web



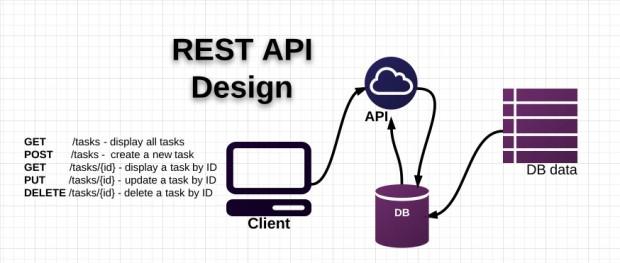
*Hình 3: Nhược điểm của ReactJS*

## Restful API

### Tổng quan về Restful API

#### . Khái niệm

RESTFUL API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 4: *RESTFUL* API

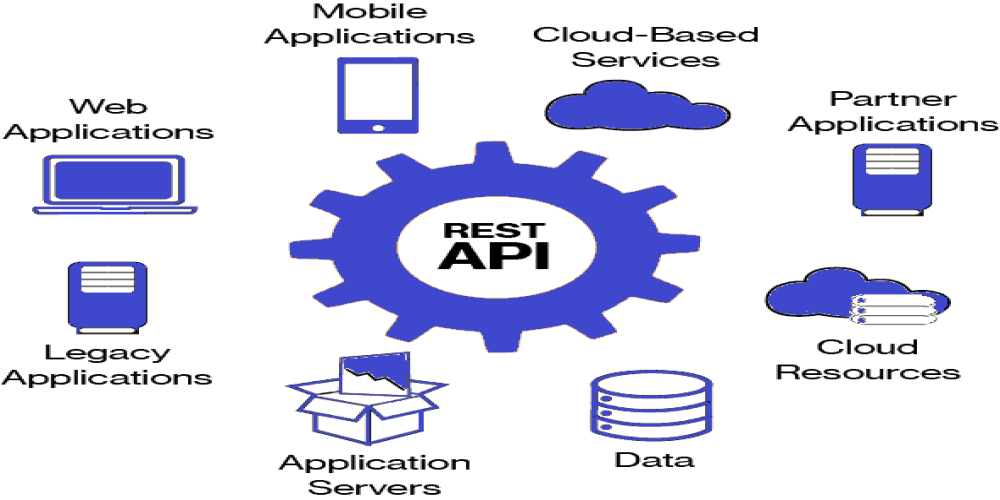
#### Các thành phần

**API** (**A**pplication Programming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

**REST** (REpresentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.



*Hình 5: Các thành phần của RESTFUL API*

## Tìm hiểu về JAVASCRIPT

### Tổng quan về Javascript

JavaScript viết tắt là JS là ngôn ngữ lập trình phổ biến dùng để tạo ra các trang web tương tác. Được tích hợp và nhúng vào HTML giúp web trở nên sống động hơn. JavaScript đóng vai trò như một phần của trang web, thực thi cho phép Client-Side Script từ phía người dùng cũng như phía máy chủ (Nodejs) tạo ra các trang web động.

### Cách thức hoạt động của Javascript

Thông thường, JavaScript sẽ được nhúng trực tiếp vào một web hoặc chúng được tham chiếu qua file .js hoặc JavaScript.

Đây là một ngôn ngữ đến từ phía Client nên Script sẽ được download về máy client khi truy cập.

Tại đây, chúng sẽ được hệ thống xử ý. Vì vậy, bạn không cần phải tải về máy server rồi chờ cho chúng xử lý xong mới phản hồi được kết quả đến client.

### Đặc điểm của JavaScript

Thay đổi nội dung HTML: Một trong số nhiều phương thức HTML JavaScript chính là getElementById (). Chúng được sử dụng để tìm một phần tử của HTML với id = ” demo” và dùng để thay đổi nội dung của phần từ (Internal HTML) sang thành “Hello JavaScript”

Thay đổi giá trị thuộc tính HTML: Tổng quan về javascript còn có thể sử dụng để thay đổi các giá trị của thuộc tính. Ví dụ: thay đổi thuộc tính src (source) của tag<img>.



*Hình 6: JavaScript*

Thay đổi kiểu HTML: Đây chính là một hoạt động biến thể của việc thay đổi thuộc tính của HTML ở trên. Ví dụ: document.getElementById(‘demo’).style.fontSize = ’35px;

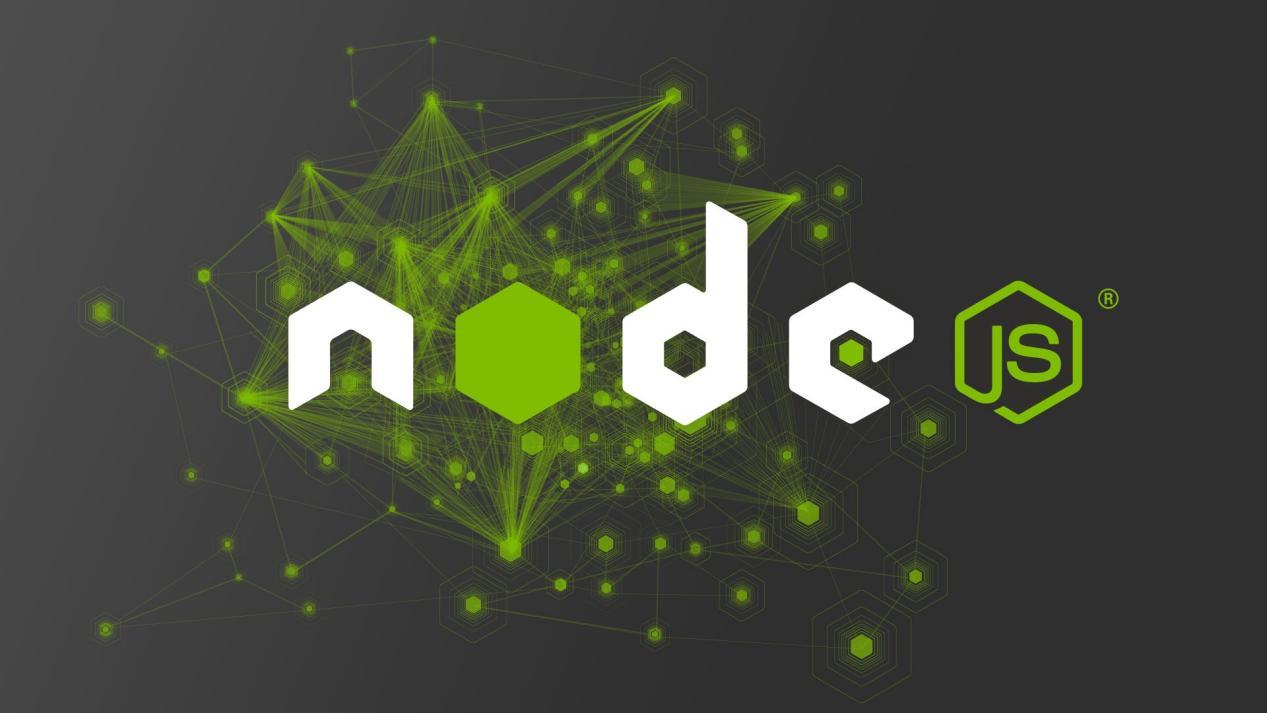
Ẩn các phần tử HTML: Một hoạt động tiếp theo là Javascript có thể ẩn được các phần tử HTML. Chúng có thể được thực hiện thông qua hoạt động thay đổi kiểu hiển thị các phần tử HTML.

Hiển thị các phần tử HTML: Một điểm đặc biệt là JavaScript có thể hiển thị được các yếu tố HTML ẩn. Đồng thời, cũng có thể thực hiện được thông qua cách thay đổi kiểu hiển thị phần tử.

## Node.js

### Khái niệm về Node.js

Được phát hành vào năm 2009, NodeJS, hay còn được biết với tên gọi chính thức là Node.js, là môi trường thời gian chạy (runtime environment) JavaScript đa nền tảng và mã nguồn mở. NodeJS cho phép các lập trình viên tạo cả ứng dụng front-end và back-end bằng JavaScript.



*Hình 7: NodeJs*

#### NodeJS là mã nguồn mở

Điều này có nghĩa là mã nguồn của NodeJS được cung cấp công khai. Và được duy trì bởi những người đóng góp từ khắp nơi trên thế giới.

#### NodeJS hỗ trợ đa nền tảng

NodeJS không phụ thuộc vào bất kỳ phần mềm hệ điều hành nào mà đều có thể hoạt động trên Linux, mac hoặc Windows.

#### NodeJS là môi trường thời gian chạy mã JavaScript

Khi bạn viết code JavaScript trong trình soạn thảo văn bản, code đó không thể thực hiện bất kỳ tác vụ nào trừ khi bạn thực thi (hoặc chạy) nó. Và để chạy code, bạn cần có môi trường thời gian chạy.

Các trình duyệt như Chrome và Firefox có môi trường thời gian chạy. Đó là lý do tại sao họ có thể chạy code JavaScript. Trước khi NodeJS được tạo, JavaScript chỉ có thể chạy trên trình duyệt và chỉ được sử dụng để xây dựng các ứng dụng front-end.

NodeJS cung cấp môi trường thời gian chạy bên ngoài trình duyệt. Nó cũng được xây dựng trên công cụ JavaScript của Chrome (V8 Engine). Điều này giúp bạn có thể xây dựng các ứng dụng backend bằng cách sử dụng cùng ngôn ngữ lập trình JavaScript mà bạn quen thuộc.

### Một số ưu điểm khi sử dụng NodeJS

#### NodeJS có thể mở rộng

Các ứng dụng NodeJS có khả năng mở rộng cao vì chúng hoạt động không đồng bộ vì các yêu cầu đồng thời có thể được xử lý rất hiệu quả bằng NodeJS.

NodeJS hoạt động trên một luồng đơn nên khi có một yêu cầu đến, NodeJS sẽ bắt đầu xử lý yêu cầu đó và sẵn sàng xử lý yêu cầu tiếp theo.

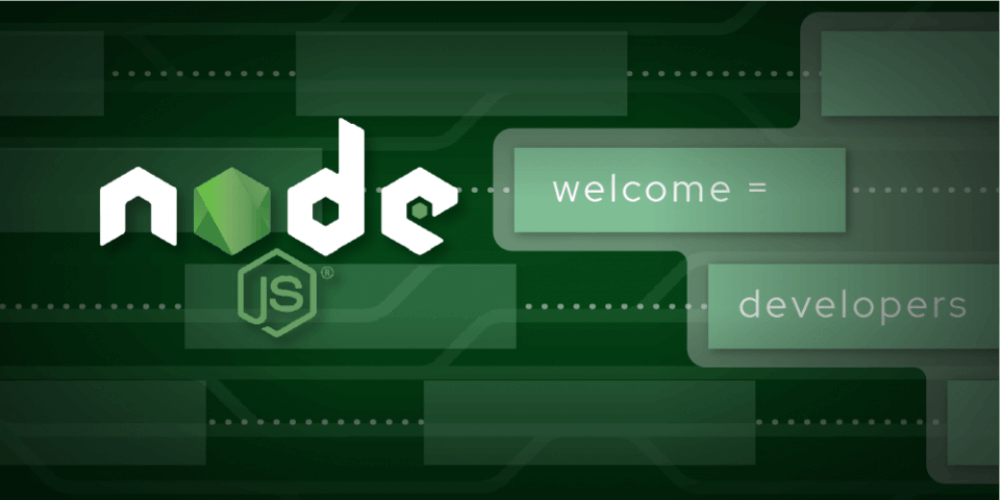
Tính năng hấp dẫn nhất của NodeJS là khả năng phân vùng các ứng dụng theo chiều ngang và quy trình phân vùng này chủ yếu đạt được nhờ sử dụng các tiến trình con. Bằng cách sử dụng tính năng này, các phiên bản ứng dụng riêng biệt được cung cấp cho các đối tượng mục tiêu khác nhau và cho phép NodeJS đáp ứng sở thích của khách hàng.

#### Thời gian thực thi code nhanh

Công cụ thời gian chạy (runtime motor) JavaScript V8 được NodeJS sử dụng và cũng được Google Chrome sử dụng. Một trình bao bọc được trung tâm cung cấp cho JavaScript và vì lý do đó, công cụ thời gian chạy trở nên nhanh hơn.

Công cụ JavaScript V8 của Google Chrome là nền tảng của NodeJS, cho phép thực thi mã nhanh hơn. Công cụ này biên dịch code JavaScript thành code máy, giúp code được triển khai hiệu quả và dễ dàng hơn và nhanh hơn.

Đồng thời, việc sử dụng các khái niệm như lập trình không đồng bộ và cách vận hành non-blocking trên các hoạt động I/O cũng giúp nâng cao hiệu suất của NodeJS.



*Hình 8: Ưu điểm của NodeJs*

#### Khả năng tương thích trên nhiều nền tảng

Các loại hệ điều hành khác nhau như Windows, UNIX, [LINUX](https://itviec.com/blog/linux/), MacOS và các thiết bị di động khác đều có thể sử dụng NodeJS.

#### Sử dụng JavaScript

NodeJS sử dụng JavaScript. Hầu hết các lập trình viên đều quen thuộc với JavaScript, vì vậy đối với họ, việc hiểu NodeJS trở nên rất dễ dàng hơn.

#### Truyền dữ liệu nhanh

Thời gian xử lý những dữ liệu đã được truyền đến các luồng khác nhau thường sẽ mất nhiều thời gian. Trong khi đó, để xử lý dữ liệu, NodeJS chỉ mất một khoảng thời gian rất ngắn và thực hiện với tốc độ nhanh.

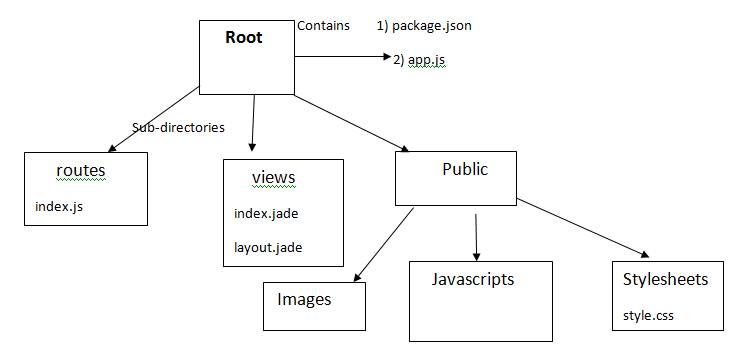
NodeJS tiết kiệm rất nhiều thời gian vì các tệp được NodeJS xử lý và tải lên đồng thời. Do đó, tốc độ tổng thể của truyền dữ liệu và video được cải thiện nhờ NodeJS.

## Tìm hiểm về Express Framework

### Giới thiệu về Express Framework

Express là một framework dành cho nodejs. Nó cung cấp cho chúng ta rất nhiều tính năng mạnh mẽ trên nền tảng web cũng như trên các ứng dụng di động. Express hỗ trợ các phương thức HTTP và middleware tạo ra một API vô cùng mạnh mẽ và dễ sử dụng. Có thể tổng hợp một số chức năng chính của express như sau:

* Định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL
* Cho phép trả về các trang HTML dựa vào các tham số.



*Hình 9: Express Framework*

## Hệ quản trị cơ sở dữ liệu MongoDB

### Giới thiệu về MongoDB

MongoDBlà một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng, MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON. MongoDB được phát triển bởi MongoDB Inc. và được cấp phép theo Giấy phép Công cộng phía Máy chủ (SSPL).

### Thuật ngữ sử dụng trong MongoDB

* ***\_id***

Là trường bắt buộc có trong mỗi document. Trường \_id đại diện cho một giá trị duy nhất trong document MongoDB. Trường \_id cũng có thể được hiểu là khóa chính trong document. Nếu bạn thêm mới một document thì MongoDB sẽ tự động sinh ra một \_id đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.

* ***Collection***

Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong cơ sở dữ liệu RDBMS (Relational Database Management System). Collection nằm trong một cơ sở dữ liệu duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.

* ***Cursor***

Đây là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.

* ***Database***

Nơi chứa các Collection, giống với cơ sở dữ liệu DMS chúng chứa các bảng. Mỗi Database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một máy chủ MongoDB có thể chứa nhiều Database.

* ***Field***

Là một cặp name – value trong một document. Một document có thể có không hoặc nhiều trường. Các trường giống các cột ở cơ sở dữ liệu quan hệ.

* ***JSON***

Viết tắt của JavaScript Object Notation. Con người có thể đọc được ở định dạng văn bản đơn giản thể hiện cho các dữ liệu có cấu trúc. Hiện tại JSON đang hỗ trợ rất nhiều ngôn ngữ lập trình.

* ***Index***

Là những cấu trúc dữ liệu đặc biệt, dùng để chứa một phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Chỉ số lưu trữ giá trị của một fields cụ thể hoặc thiết lập các fields, sắp xếp theo giá trị của các fields này. Index hỗ trợ độ phân tích một cách hiệu quả các truy vấn. Nếu không có chỉ mục, MongoDB sẽ phải quét tất cả các documents của collection để chọn ra những document phù hợp với câu truy vấn. Quá trình quét này là không hiệu quả và yêu cầu MongoDB để xử lý một khối lượng lớn dữ liệu.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Khảo sát và phân tích hệ thống

### Khảo sát yêu cầu

* ***Yêu cầu từ khách hàng***

Đầu tiên tính thẩm mỹ của web có vai trò tạo ấn tượng và thu hút khách hàng, nhưng để giữ chân khách hàng và khiến họ trở lại với web những lần sau được quyết định bởi nội dung trình diễn trên web. Khách hàng sẽ cảm thấy thỏa mãn khi cửa hàng trực tuyến có khả năng đáp ứng các nhu cầu sau đây: tìm sản phẩm nhanh, sản phẩm có đủ thông tin để lựa chọn, mua hàng nhanh chóng.

* ***Yêu cầu từ người quản trị***

Tạo nội dung, cập nhật dữ liệu phù hợp: người quản trị web sau khi tiếp nhận phải là người tạo nội dung hay chỉnh sửa, cập nhật những nội dung cho hệ thống.

Kiểm tra tình trạng đơn hàng và quản lý người dùng: Người quản trị có thể kiểm tra được tình trạng đơn hàng để có thể xử lý các yêu cầu người dùng về đơn hàng.

Thống kê doanh thu: Doanh thu thể hiện sự phát triển của doanh nghiệp. Từ đó tập trung phát triển các mặt hàng được khách hàng tiêu dùng yêu cầu hoặc sử dụng nhiều nhất.

### Giới thiệu về hệ thống

Hiện nay với sự phát triển nhanh chóng của công nghệ thông tin ngày càng hiện đại, đã dẫn đến việc mua sắm của con người cũng ngày càng dễ dàng hơn, chính vì thế mà xu hướng quản lý bán hàng ngày càng phổ biến hơn. Việc quản lý bán hàng không hề đơn giản, rất dễ nhầm lẫn cũng chính vì thế mà đã ra đời hệ thống quản lý bán hàng.

Hệ thống được dựa trên ứng dụng các công nghệ (Nodejs, React JS, JavaScript...) để xây dựng một web bán hàng. Với các chứng năng theo nhu cầu của người sử dụng (khách hàng, quản trị).

Khách hàng có thể truy cập web bằng các thiết bị di động, laptop, ...đến trang web xem các sản phẩm và đặt hàng, tra cứu tin tức, liên hệ người bán.

Quản trị có thể quản lý được sản phẩm, danh mục, tin tức, chỉnh sửa giao diện cho trang web của mình. Thêm, xóa, sửa nội dung cho web.

### Mục tiêu sử dụng hệ thống

Bán sản phẩm, dịch vụ: Với một trang web thương mại điện tử, mục đích chính của web có thể bán được nhiều sản phẩm, gói dịch vụ cùng với nhiều tiện ích cho khách hàng.

Tăng khả năng tiếp cận và phạm vi tiếp cận với khách hàng: Khách hàng có thể truy cập web mọi nơi để đáp ứng nhu cầu mua sắm.

Xây dựng thương hiệu: Xây dựng thương hiệu cũng giúp thu hút khách hàng mới và khiến khách hàng hiện tại trung thành hơn.

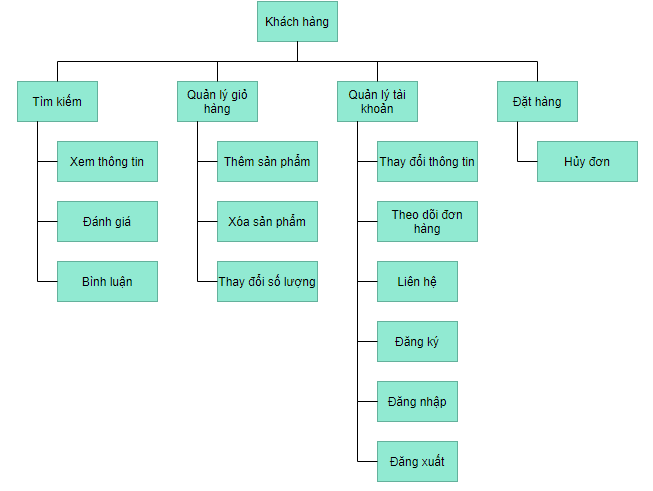
Quản lý tài nguyên: Người dùng dễ dàng quản lý dữ liệu.

### Các vai trò của người dùng trong hệ thống

* ***Khách hàng***
* Xem thông tin và mua sắm các sản phẩm được đăng bán trên web.
* Đăng nhập, đăng ký thành viên.
* Quản lý thông tin cá nhân.
* Đánh giá về các sản phẩm, dịch vụ, ...tương tác với người bán.
* ***Quản trị***
* Quản lý thông tin sản phẩm, danh mục, tin tức, ...
* Cập nhật giao diện web
* Thêm, xóa, cập nhật dữ liệu.
* Quản lý nội dung web.

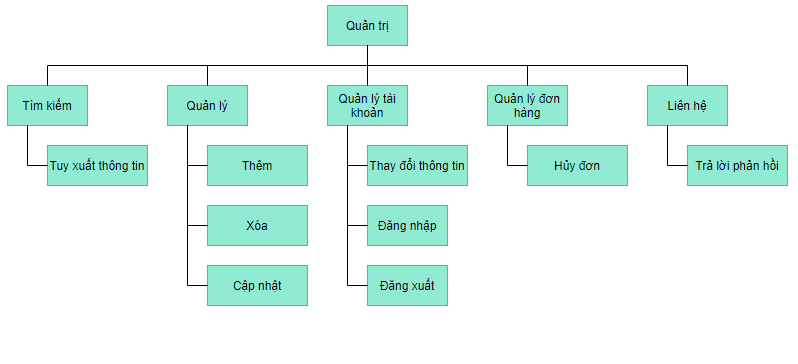
### Các yêu cầu và chức năng của hệ thống

* ***Người dùng khách hàng***
* Hiển thị: hiện thể sản phẩm theo danh mục.
* Xem thông tin: xem thông tin sản phẩm, thông tin cá nhân.
* Tìm kiếm: Tìm kiếm sản phẩm theo từ khóa.
* Đánh giá: Bình luận đánh giá về sản phẩm, dịch vụ,...
* Đăng nhập, đăng ký: đăng nhập hệ thống bằng tài khoản đã đăng ký.
* Giỏ hàng: Thêm xóa sửa giỏ hàng
* Đặt hàng: cho phép khách hàng đặt hàng trực tuyến trên web.
* Liên hệ: cho phép khách hàng liên hệ quản trị.
* Hủy đặt hàng: cho phép khách hàng hủy đơn hàng đã đặt.



*Hình 10: Biểu đồ tác nhân khách hàng*

* ***Người dùng quản trị***
* Quản lý sản phẩm: Cho phép thêm, sửa, xóa sản phẩm trong mỗi danh mục sản phẩm.
* Quản lý tin tức: Cho phép thêm, sửa, xóa tin tức trong mỗi danh mục tin tức.
* Quản lý danh mục: Cho phép thêm, sửa, xóa tin tức trong mỗi danh mục.
* Quản lý thương hiệu: Cho phép thêm, sửa, xóa tin tức trong mỗi danh mục thương hiệu.
* Quản lý danh mục: Cho phép thêm, sửa, xóa tin tức trong mỗi danh mục.



*Hình 11:Biểu đồ tác nhân quản trị*

### Các yêu cầu phi chức năng

* Hiển thị thông tin đầy đủ (hình ảnh, giá, tên,..).
* Màu sắc phù hợp.
* Bảo mật thông tin.
* Font chữ, cỡ chữ phù hợp.
* Thông tin chính xác.
* Dễ dàng sử dụng, khai thác.
* Phù hợp nhu cầu.

### Các chức năng phía giao diện người dùng

| **STT** | **Chức năng** | **Mô tả** |
| --- | --- | --- |
| 1 | Đăng ký | Khách hàng đăng ký thành viên để thực hiện thao tác giao dịch. |
| 2 | Đăng nhập | Khách hàng đăng nhập vào web bằng tài khoản mật khẩu đã đăng ký trước đó. |
| 3 | Tìm kiếm sản phẩm | Khách hàng nhập thông tin cần tìm vào thanh tìm kiếm hoặc tìm kiếm theo danh mục phân loại. |
| 4 | Xem chi tiết sản phẩm | Khách hàng có thể xem chi tiết sản phẩm bao gồm như: Tên sản phẩm, giá, thương hiệu, mô tả, tình trạng… |
| 5 | Đánh giá | Người xem có thể bình luận về sản phẩm, nội dung và thông tin người đánh giá được lưu vào cơ sở dữ liệu. |
| 6 | Giỏ hàng | Cho phép người dùng xem các sản phẩm đã chọn, thay đổi số lượng mua hoặc xóa sản phẩm khỏi giỏ hàng. Bao gồm tính tổng giá sản phẩm khách đã chọn. |
| 7 | Đặt hàng | Khách hàng click đặt hàng các sản phẩm nằm trong giỏ hàng, hệ thống báo đặt hàng thành công và gửi thông báo xác nhận đặt hàng qua email khách hàng. |
| 8 | Tin tức | Người xem có thể xem các tin tức mới về hệ thống bán hàng. |
| 9 | Quản lý tài khoản | Khách hàng có thể xem thông tin tài khoản, thay đổi thông tin tài khoản và xem các đơn hàng đã đặt. |
| 10 | Xem sản phẩm | Hiển thị các sản phẩm mới, nổi bật, giảm giá,. . . lên web. |

Bảng 1.Các yêu cầu chức năng phía giao diện người dùng

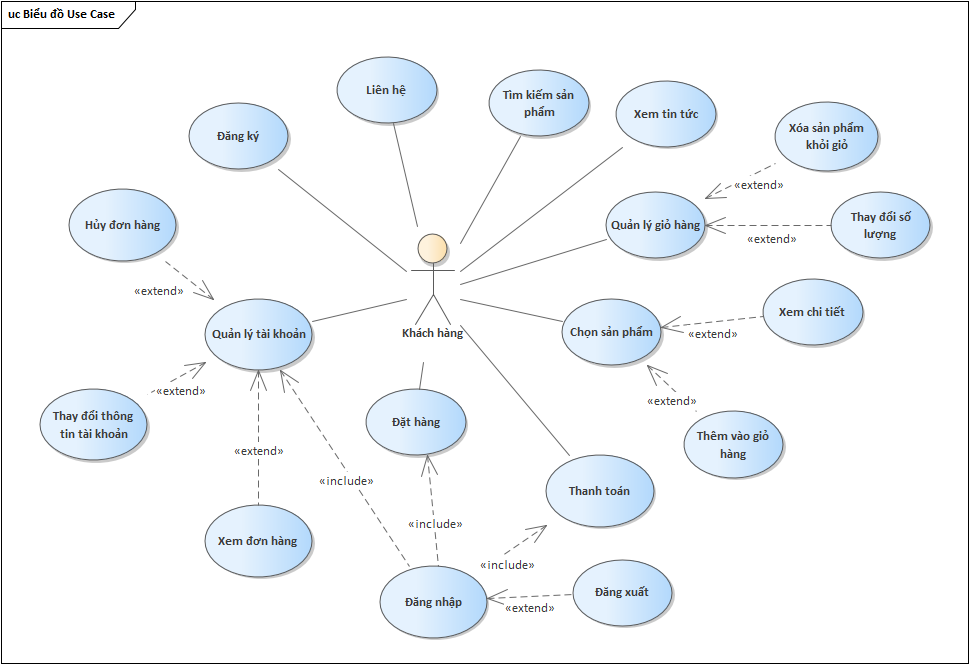
### Các chức năng phía giao diện người quản trị

| **STT** | **Chức năng** | **Mô tả** |
| --- | --- | --- |
| 1 | Đăng nhập | Đăng nhập vào hệ thống bằng tài khoản mật khẩu đã thiết lập quản trị để thao tác các chức năng quản lý |
| 2 | Quản lý sản phẩm | Quản lý thông tin sản phẩm bao gồm: Thêm, xóa, cập nhật các thông tin sản phẩm |
| 3 | Quản lý danh mục sản phẩm | Quản lý danh mục bao gồm các chức năng: Thêm, xóa, sửa, cập nhập |
| 4 | Quản lý đơn hàng | Thống kê các đơn hàng, xem tình trạng (Đã giao/ chưa giao) và chi tiết đơn hàng |
| 5 | Chi tiết đơn hàng | Hiển thị chi tiết đơn hàng bao gồm thông tin khách hàng, mã đơn hàng, các sản phẩm đặt và giá. |
| 6 | Quản lý người dùng | Thêm, xóa, sửa thông tin người dùng |
| 7 | Quản lý mã giảm giá | Quản lý mã giảm giá bao gồm các chức năng: Thêm, xóa, sửa, cập nhập |
| 8 | Quản lý tin tức | Quản lý tin tức bao gồm các chức năng: Thêm, xóa, sửa, cập nhập tin tức |
| 9 | Thống kê | Thống kê doanh thu sản phẩm. |

Bảng 2.Các chức năng phía giao diện người quản trị

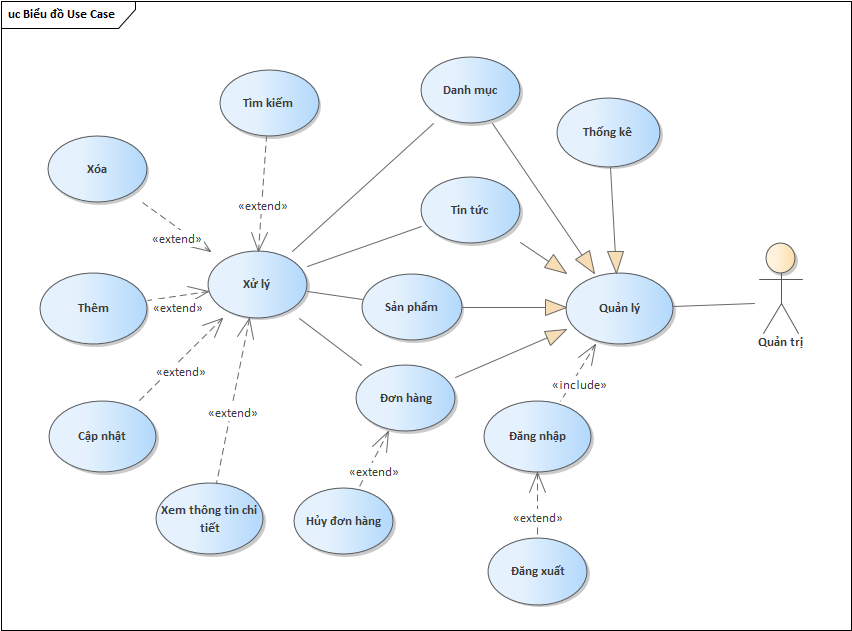
## Đặc tả và thiết kế hệ thống

### Biểu đồ use case tác nhân khách hàng



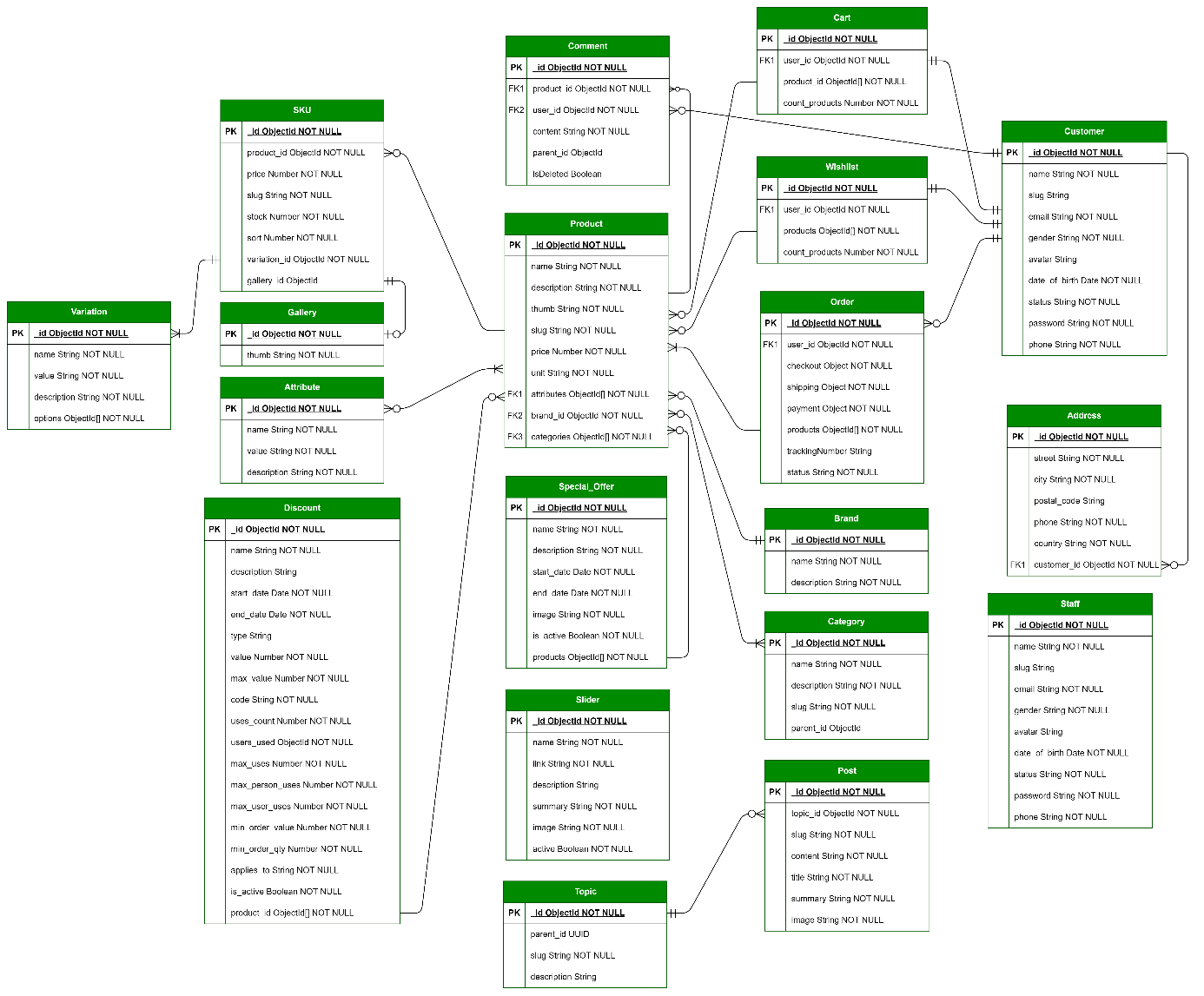
*Hình 13: Biểu đồ use case tác nhân khách hàng*

### Biểu đồ use case tác nhân quản trị

**

*Hình 14: Biểu đồ tác nhân người quản trị*

## Biểu đồ lớp

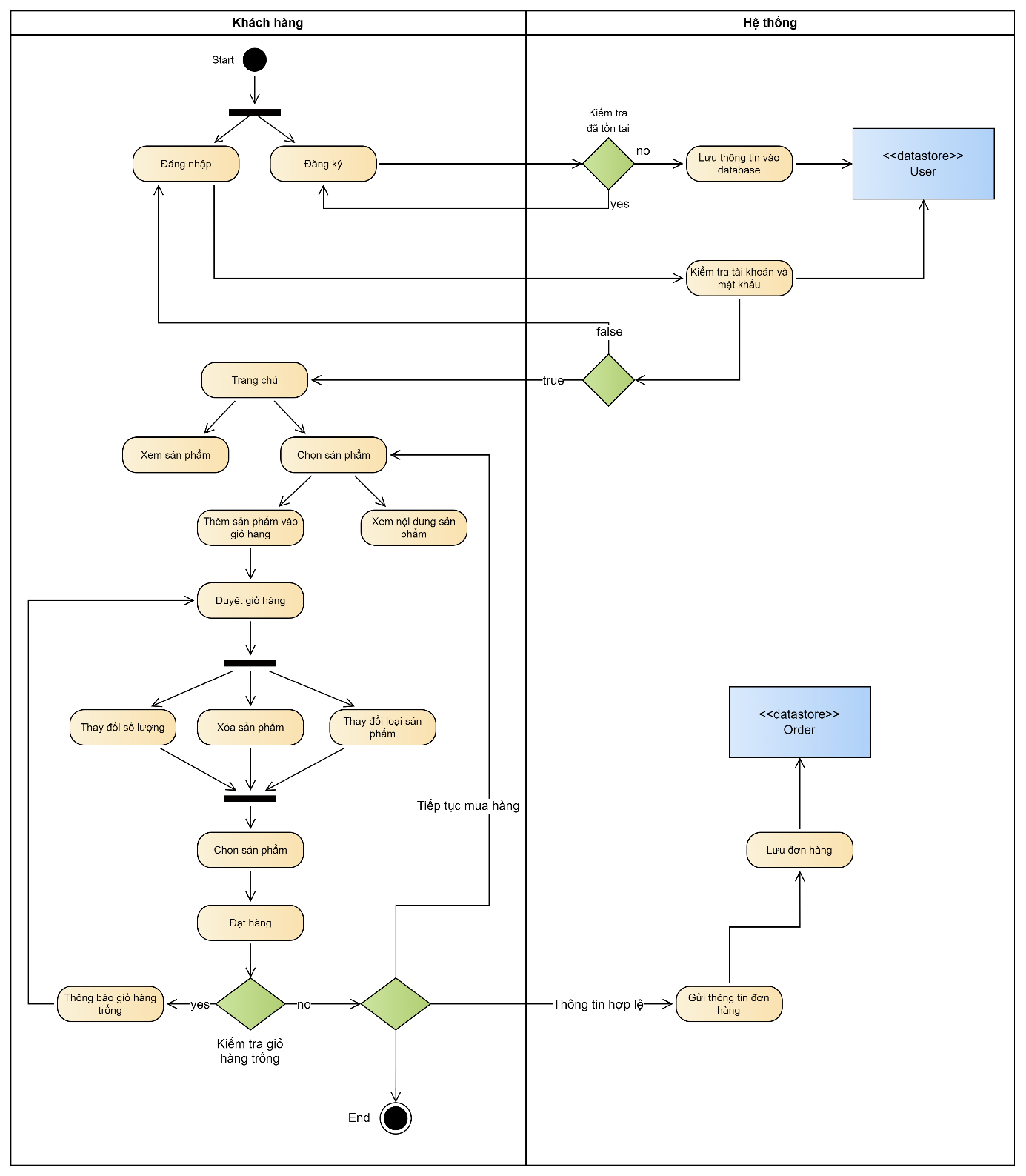


*Hình 15: Biểu đồ lớp*

## Biểu đồ hoạt động

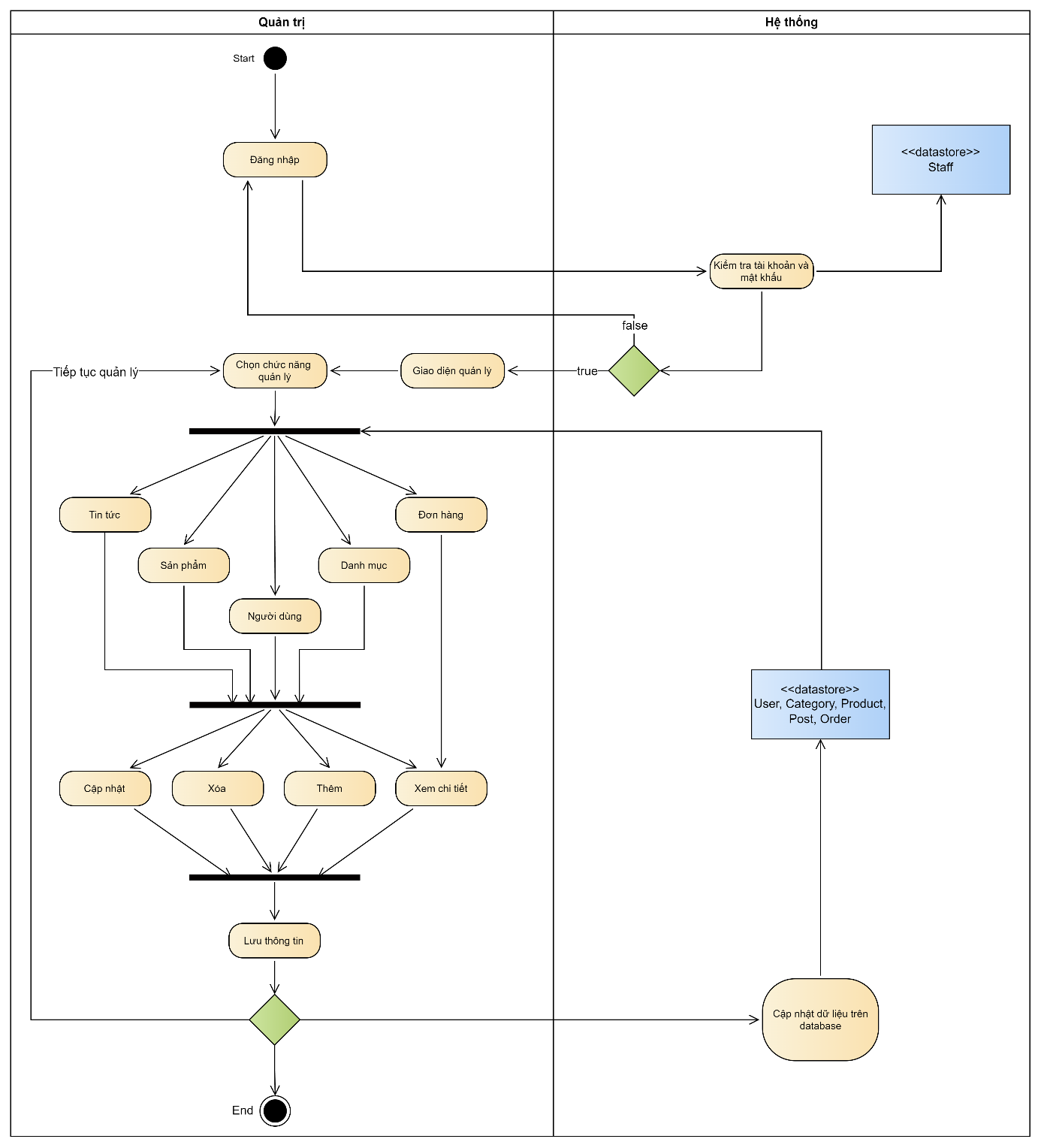
### Biểu đồ hoạt động tổng quát

* ***Tác nhân Khách hàng***



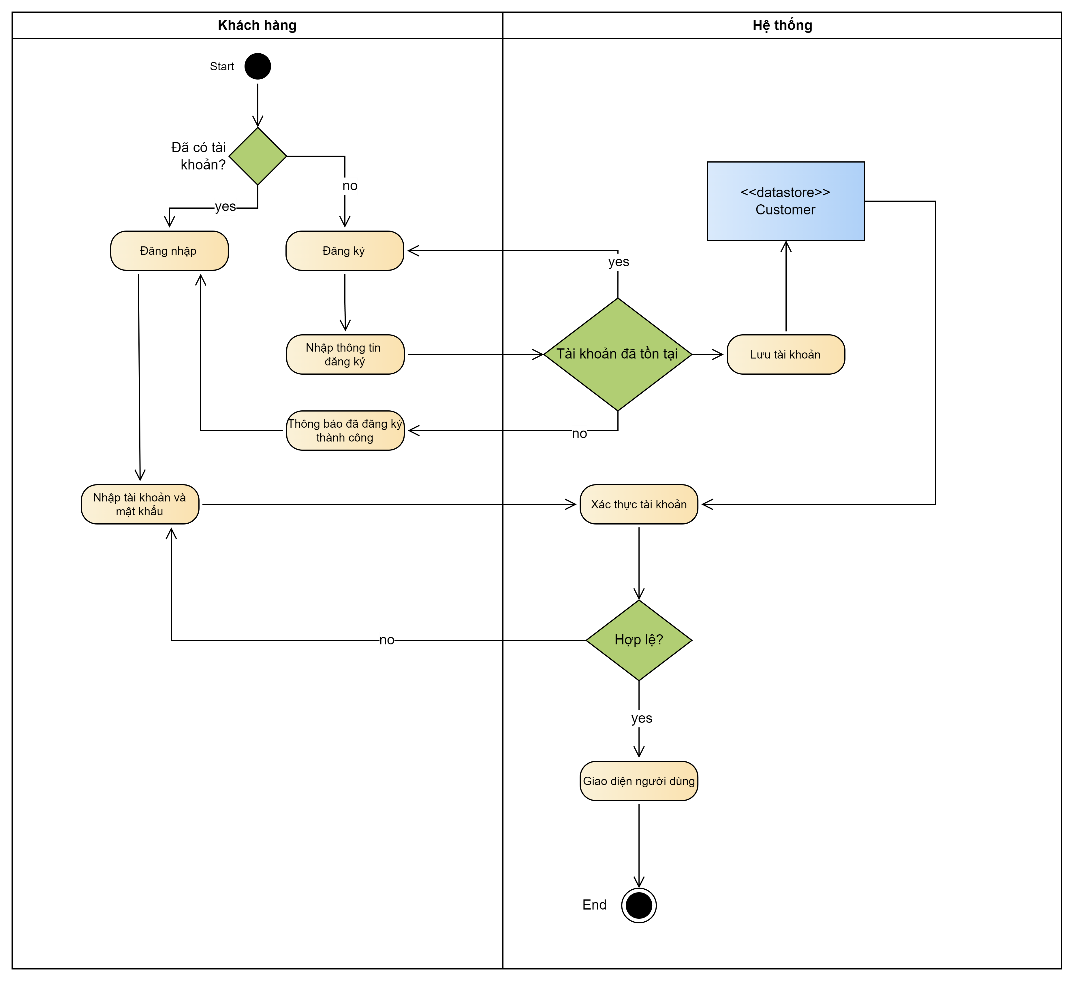
*Hình 16: Biểu đồ tổng quát tác nhân khách hàng*

* ***Tác nhân Quản trị***



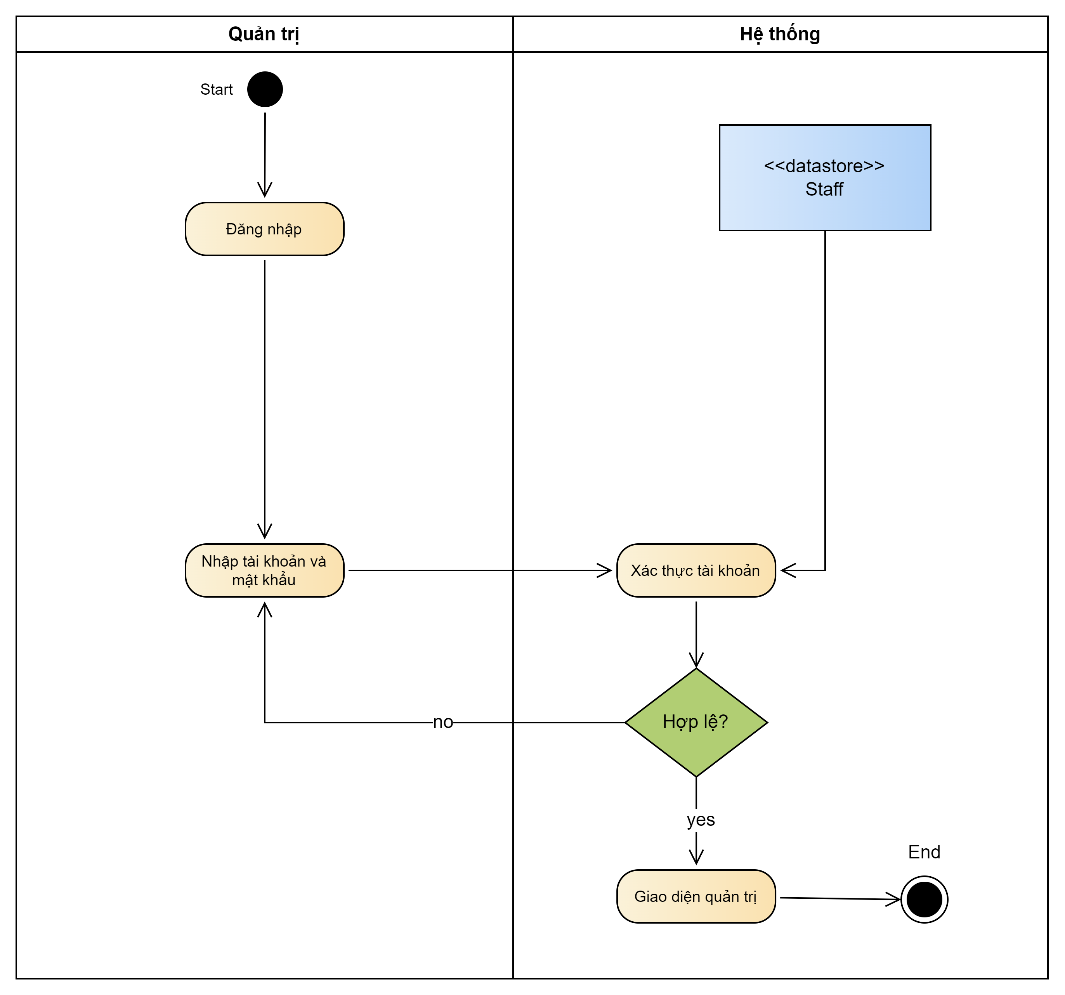
*Hình 17: Biểu đồ tổng quát tác nhân quản trị*

### Biểu đồ hoạt động chức năng đăng nhập, đăng ký



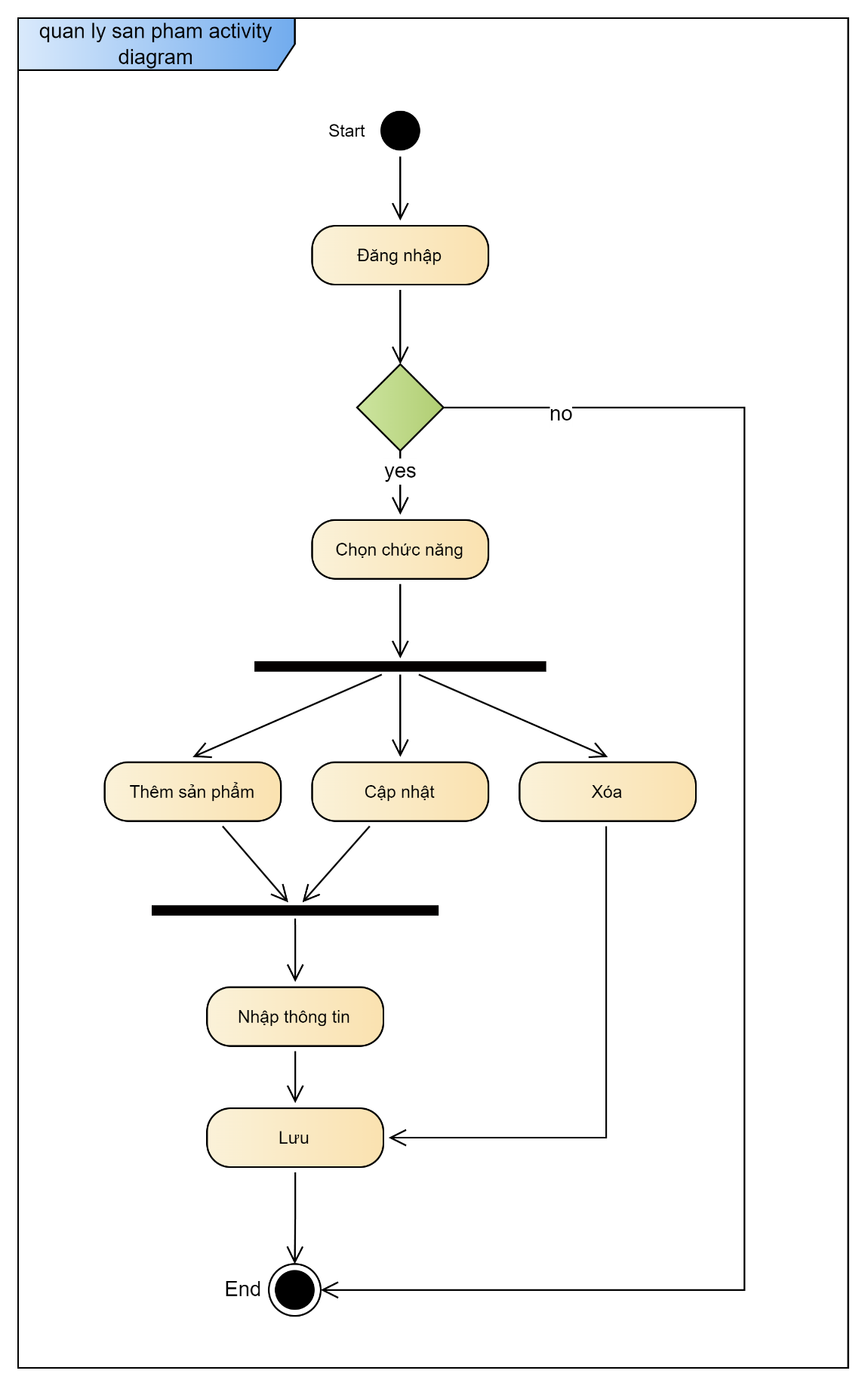
*Hình 18: Biểu đồ hoạt động chức năng đăng ký, đăng nhập của khách hàng*

### Biểu đồ hoạt động chức năng đăng nhập của quản trị viên



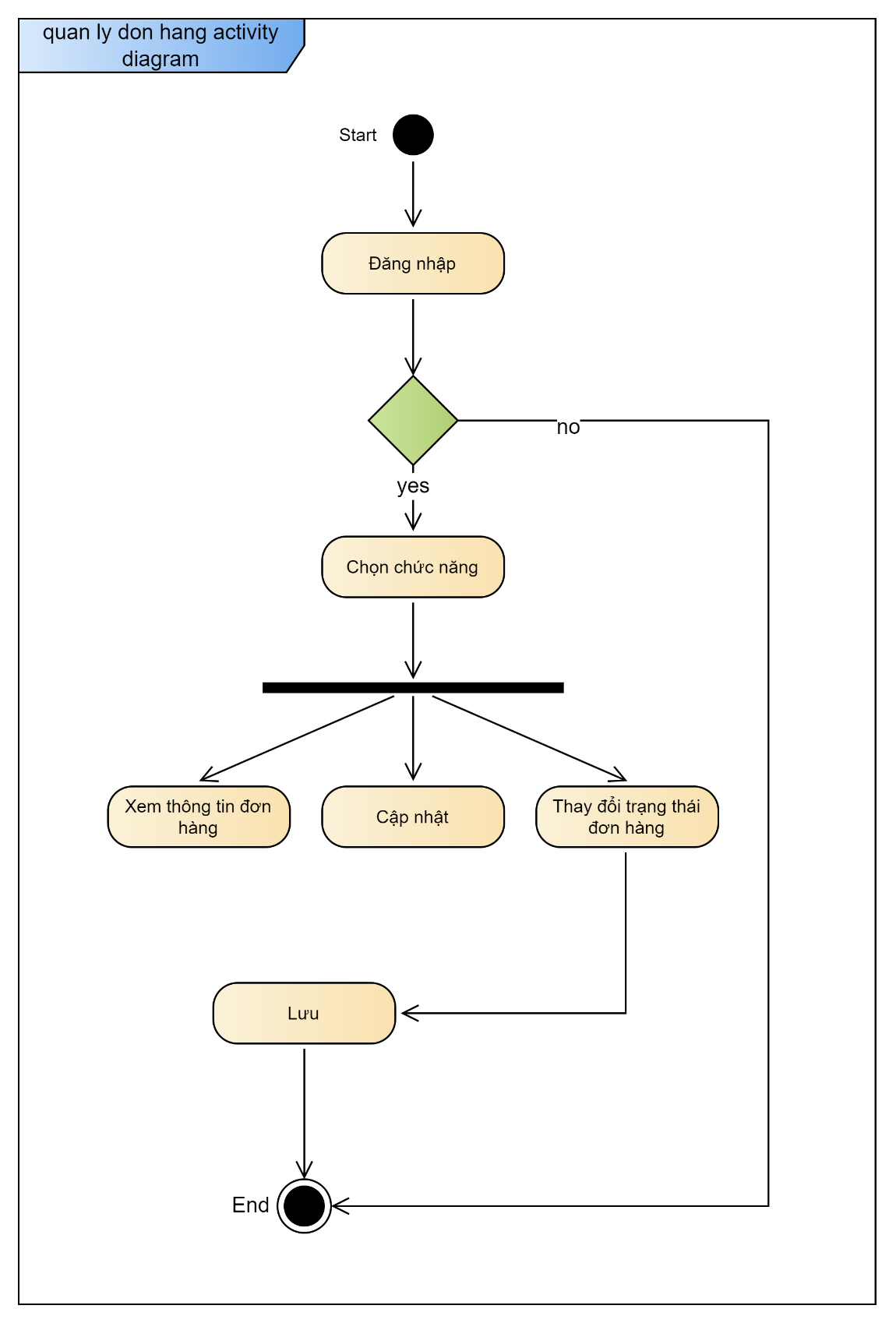
*Hình 19: Biểu đồ hoạt động chức năng của người quản trị*

### Biểu đồ hoạt động chức năng quản lý sản phẩm



*Hình 20: Biểu đồ hoạt động chức năng quản lý sản phẩm*

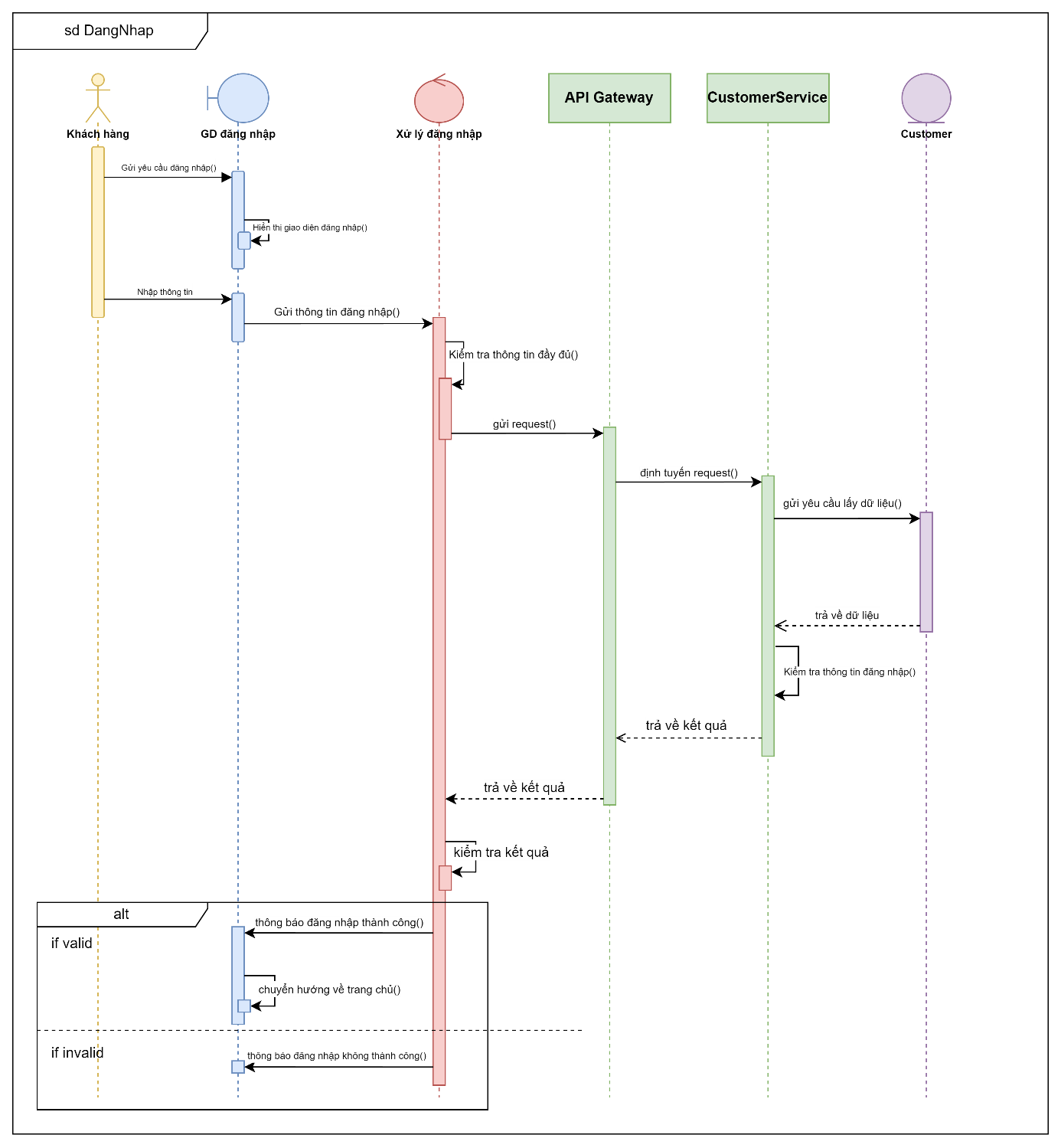
### Biểu đồ hoạt động chức năng quản lý đơn hàng



*Hình 21: Biểu đồ hoạt động chức năng quản lý đơn hàng*

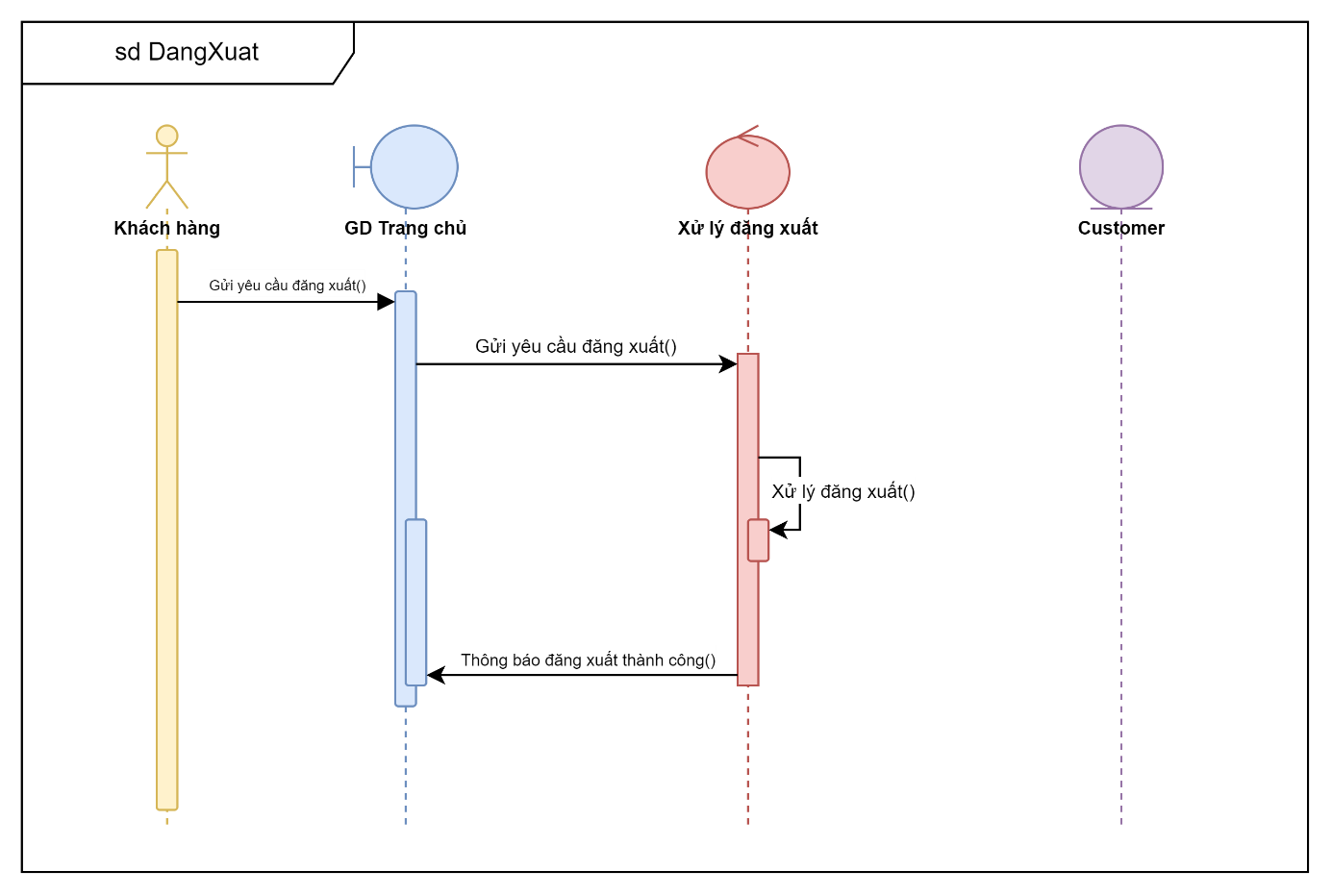
## Biểu đồ tuần tự

### Biểu đồ tuần tự Đăng nhập



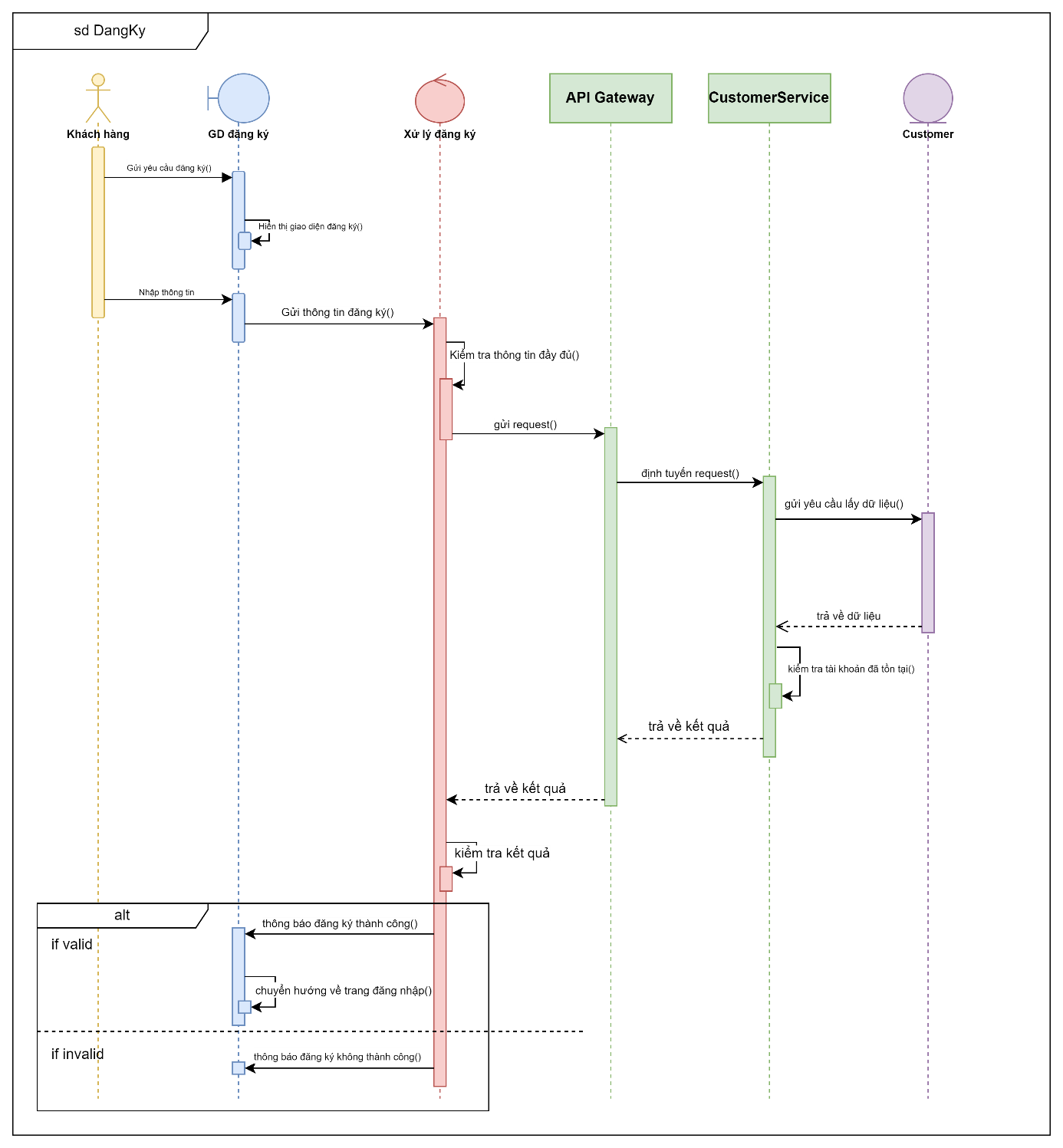
*Hình 22: Biểu đồ tuần tự đăng nhập*

### Biểu đồ tuần tự Đăng xuất



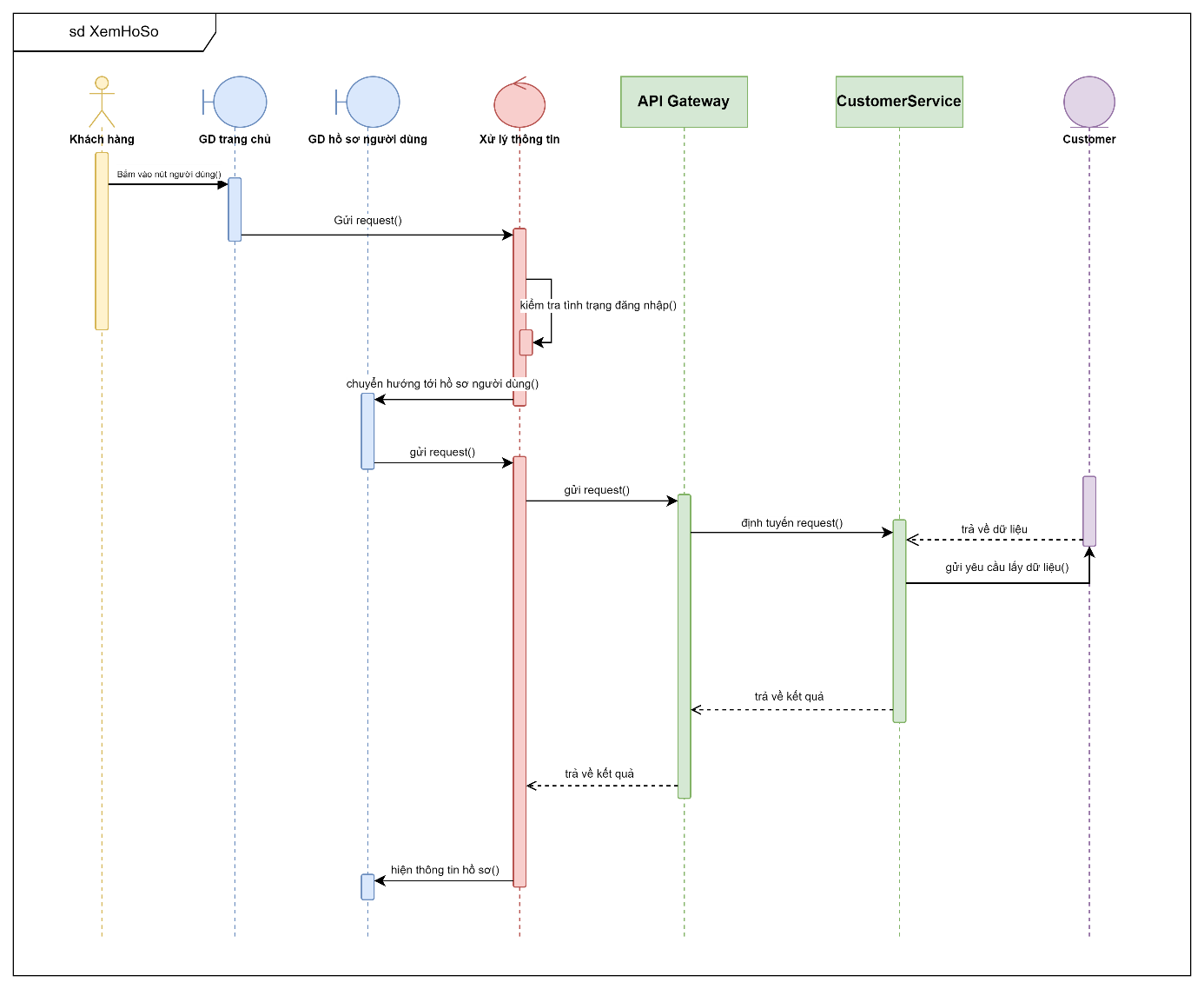
*Hình 23: Biểu đồ tuần tự đăng xuất*

### Biểu đồ tuần tự Đăng ký



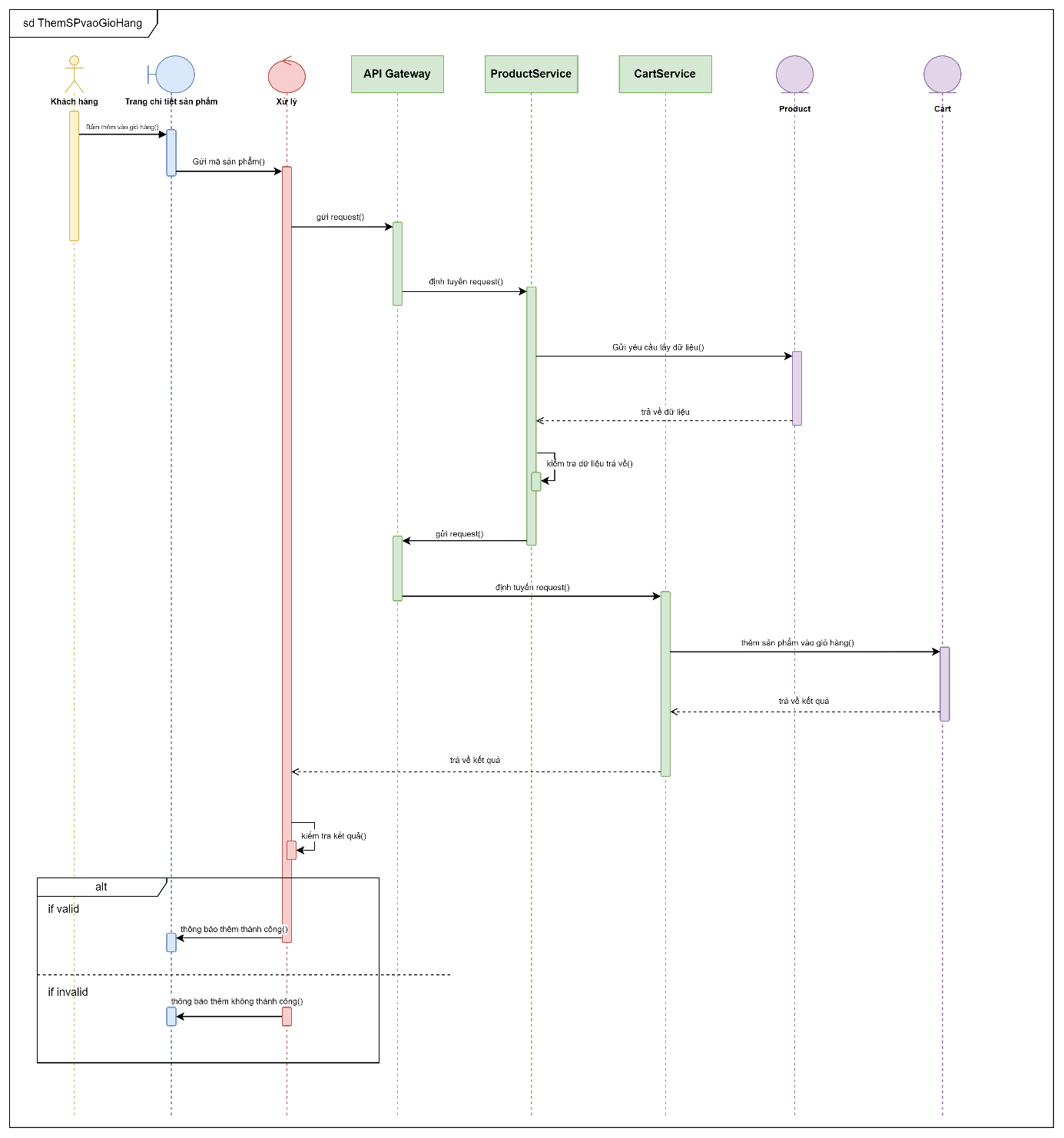
*Hình 24: Biểu đồ tuần tự đăng ký*

### Biểu đồ tuần tự Xem thông tin tài khoản



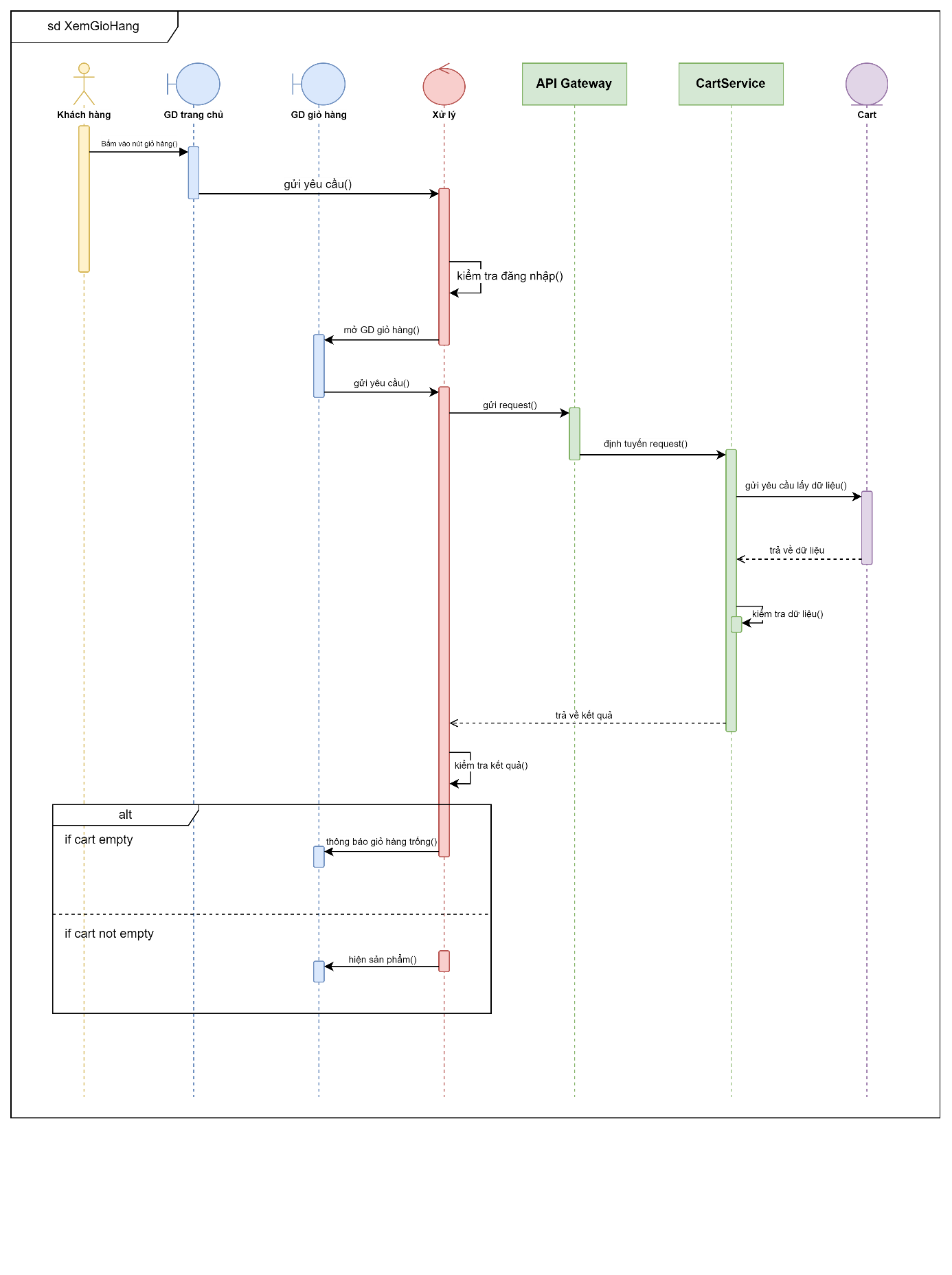
*Hình 25: Biểu đồ tuần tự xem thông tin cá nhân*

### Biểu đồ tuần tự Thêm sản phẩm vào giỏ hàng



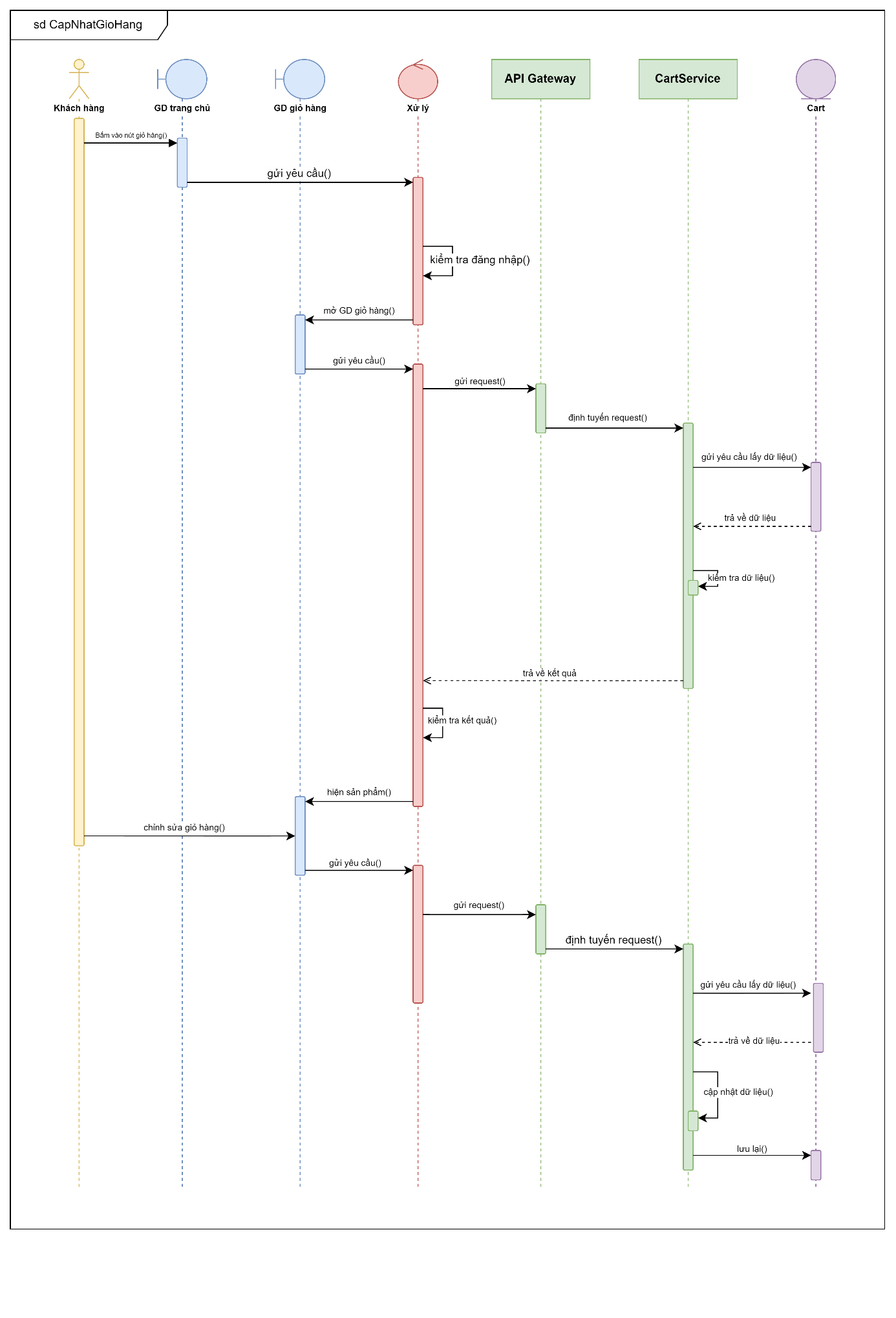
*Hình 26: Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng*

### Biểu đồ tuần tự Xem giỏ hàng



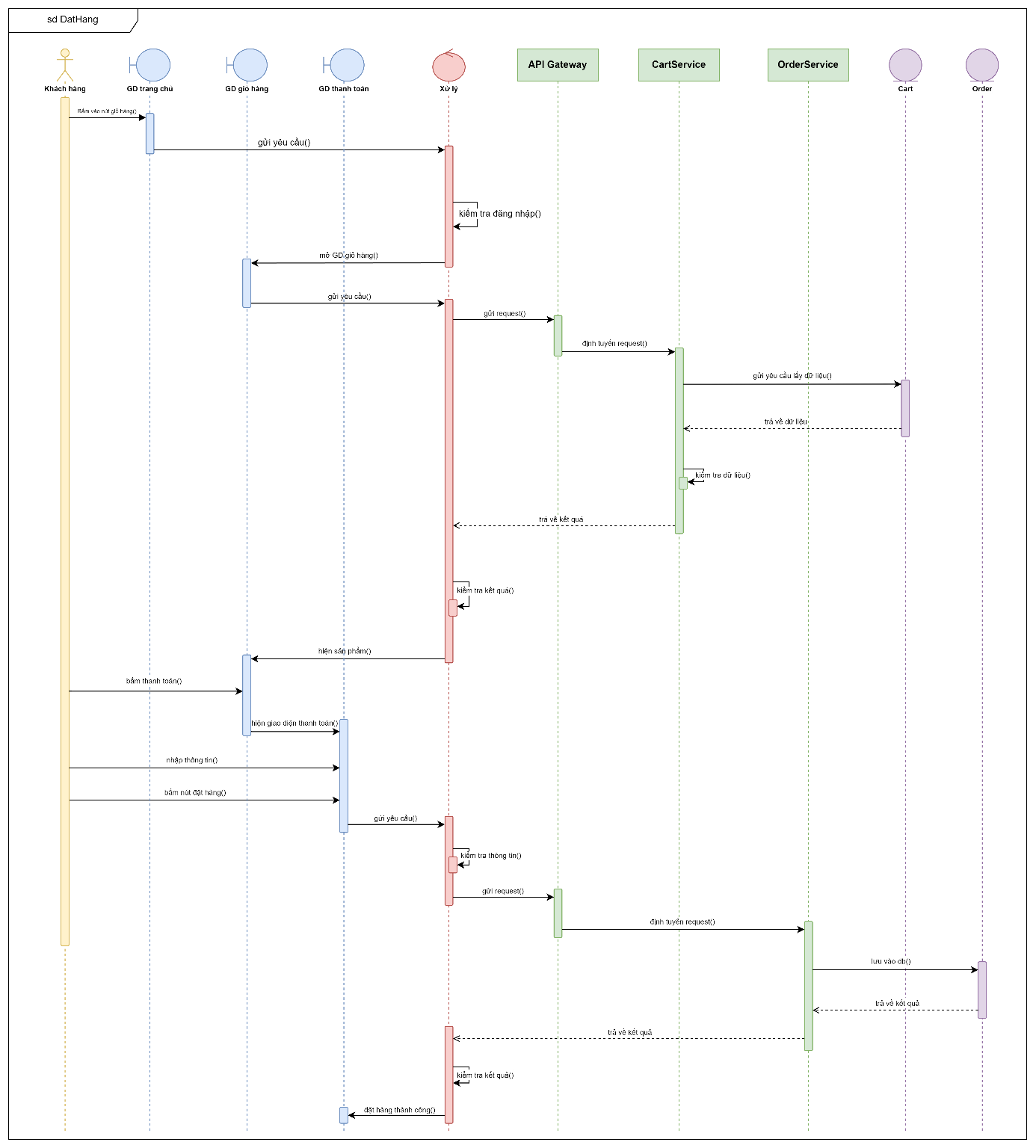
*Hình 27: Biểu đồ tuần tự xem giỏ hàng*

### Biểu đồ tuần tự Cập nhật giỏ hàng



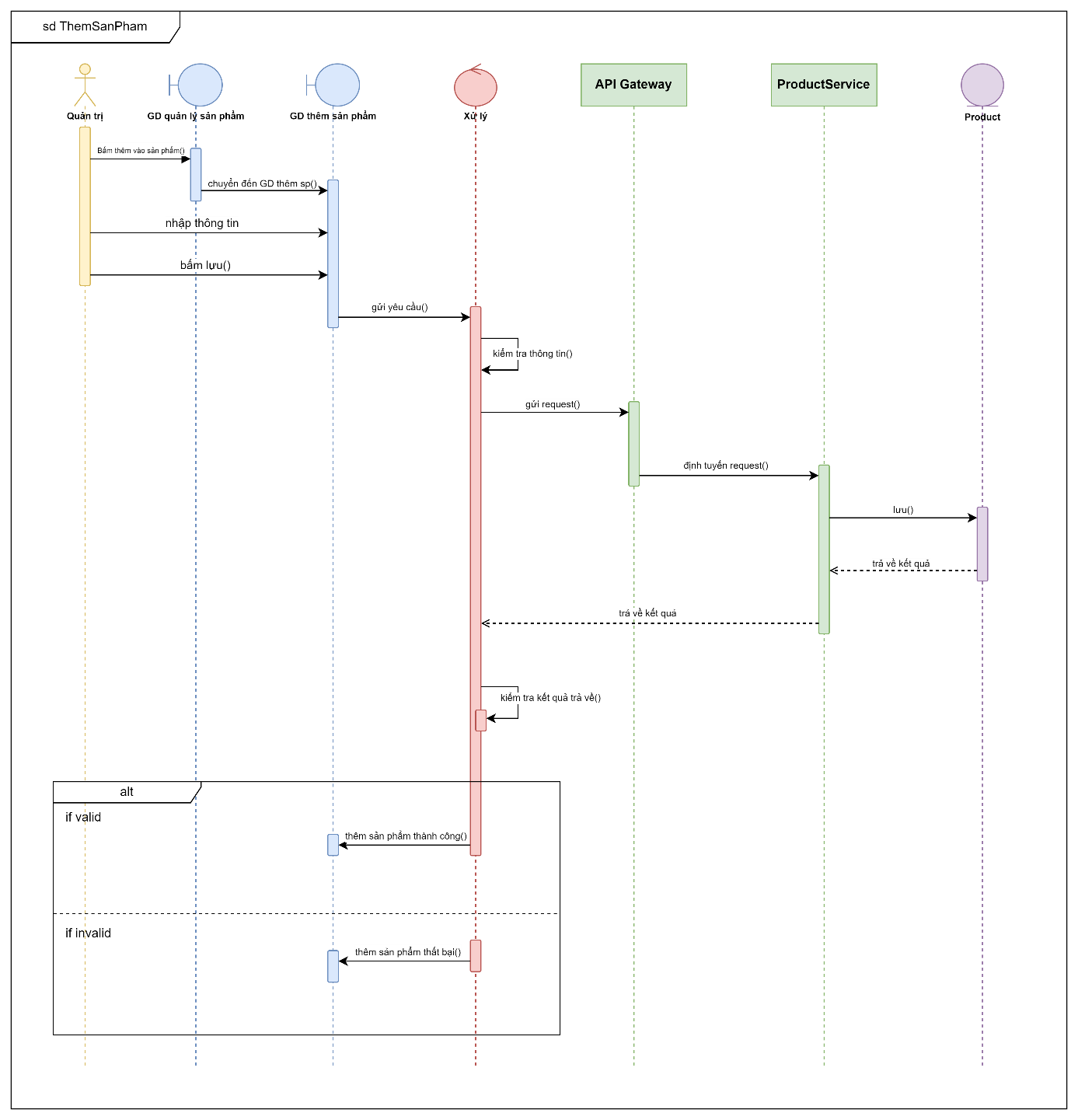
*Hình 28: Biểu đồ tuần tự cập nhật giỏ hàng*

### Biểu đồ tuần tự Đặt hàng



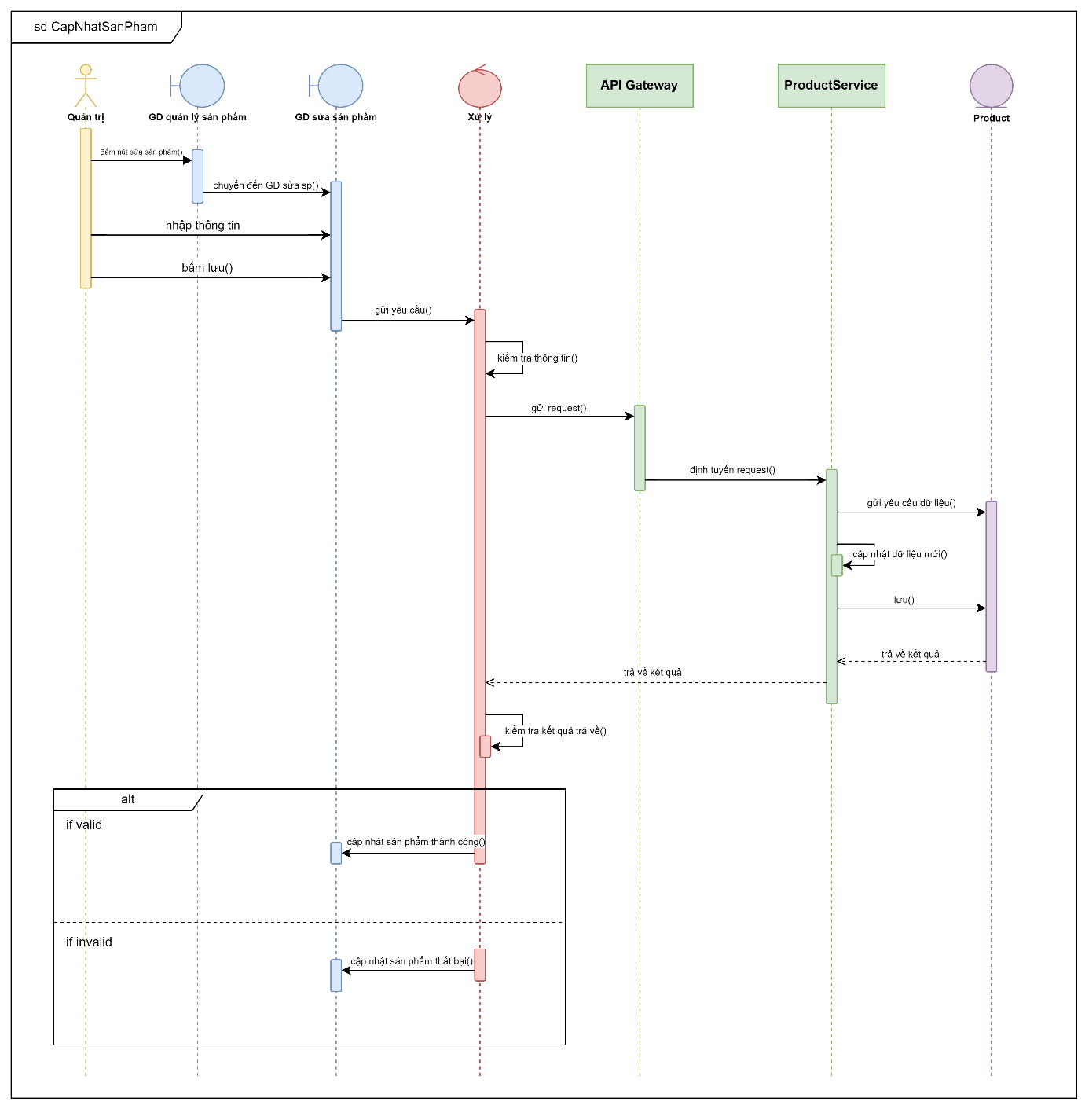
*Hình 29: Biểu đồ tuần tự đặt hàng*

### Biểu đồ tuần tự Thêm sản phẩm



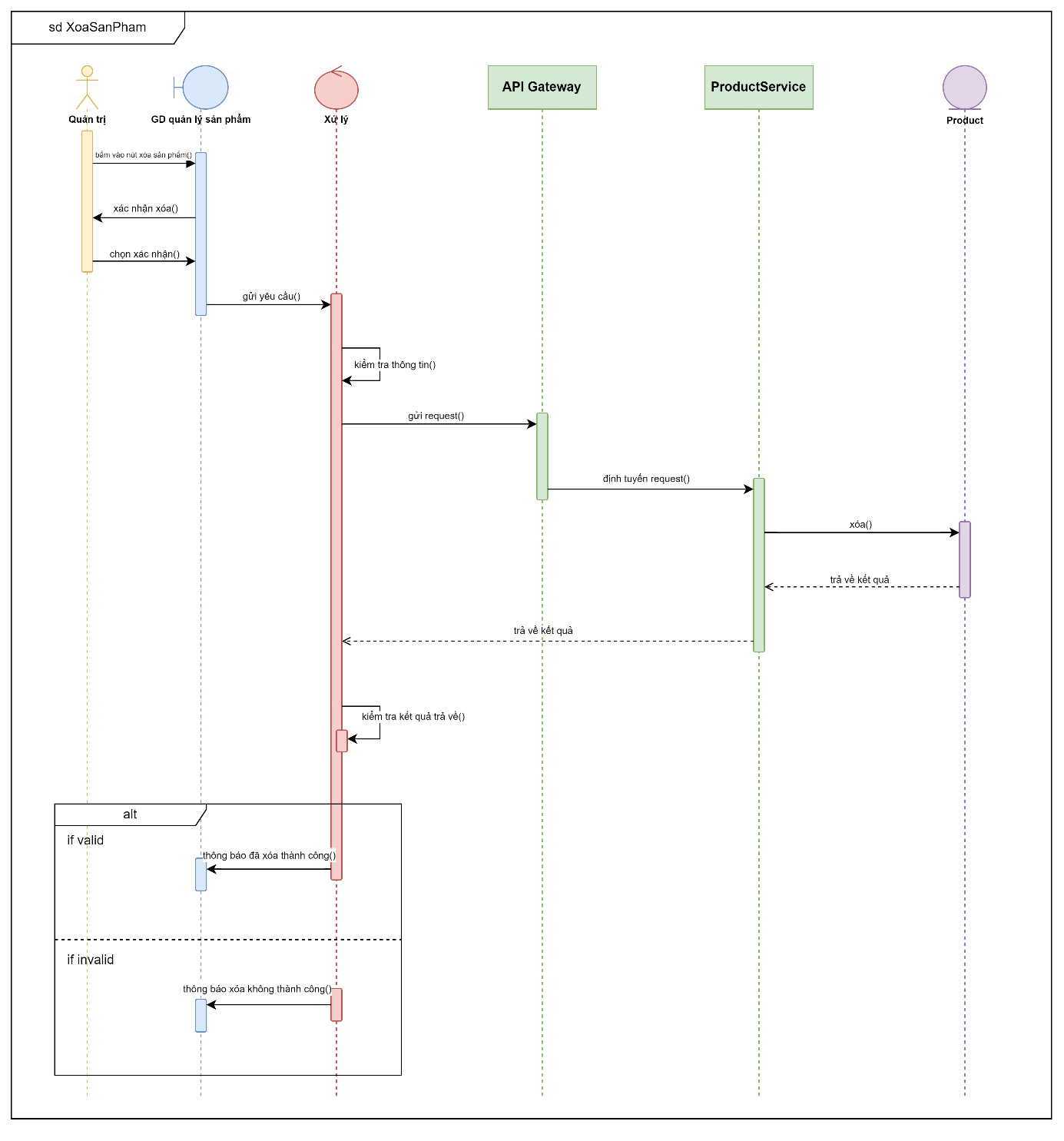
*Hình 30: Biểu đồ tuần tự thêm sản phẩm*

### Biểu đồ tuần tự Cập nhật sản phẩm



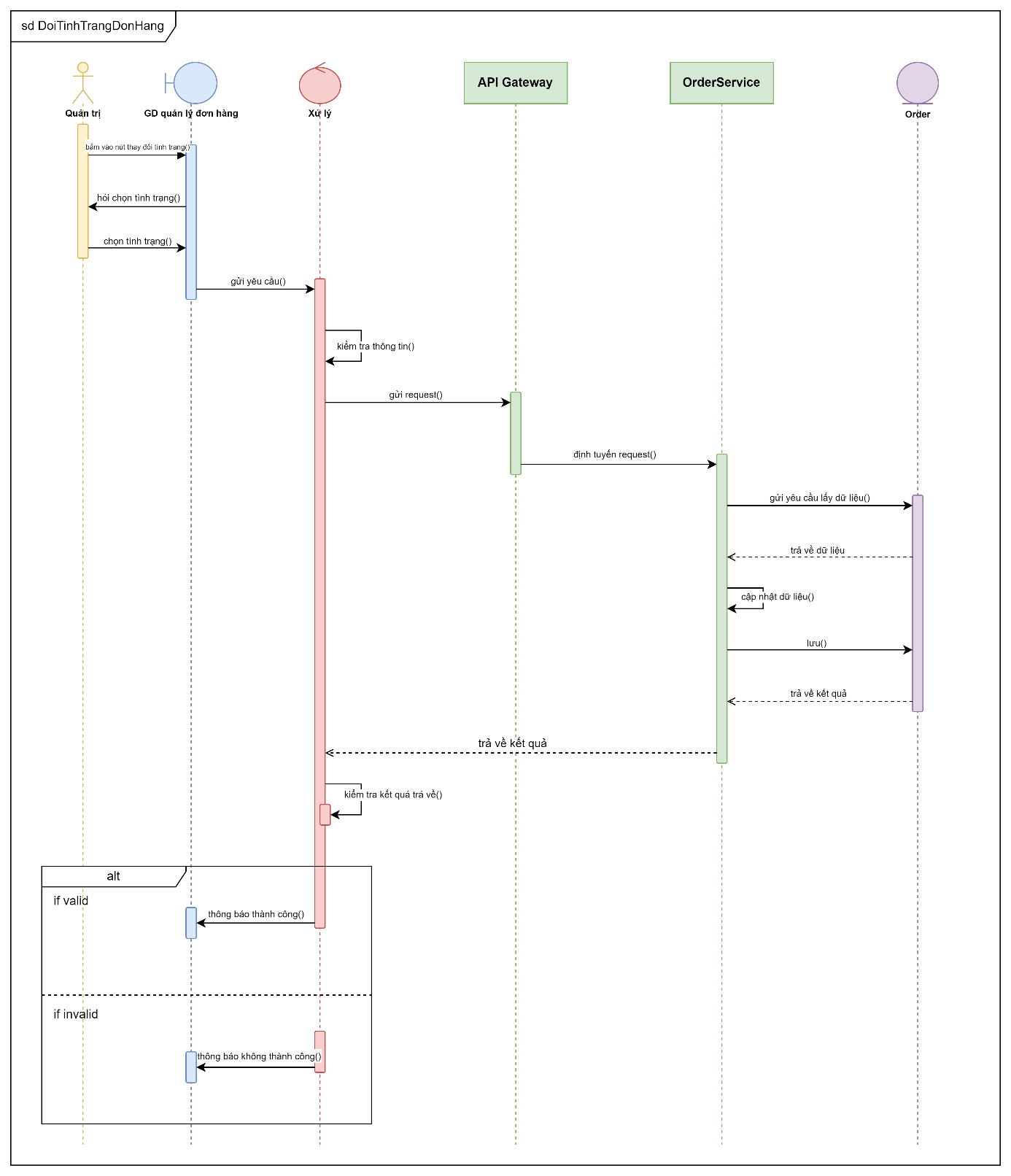
*Hình 31: Biểu đồ tuần tự cập nhật sản phẩm*

### Biểu đồ tuần tự Xóa sản phẩm



*Hình 32: Biểu đồ tuần tự xóa sản phẩm*

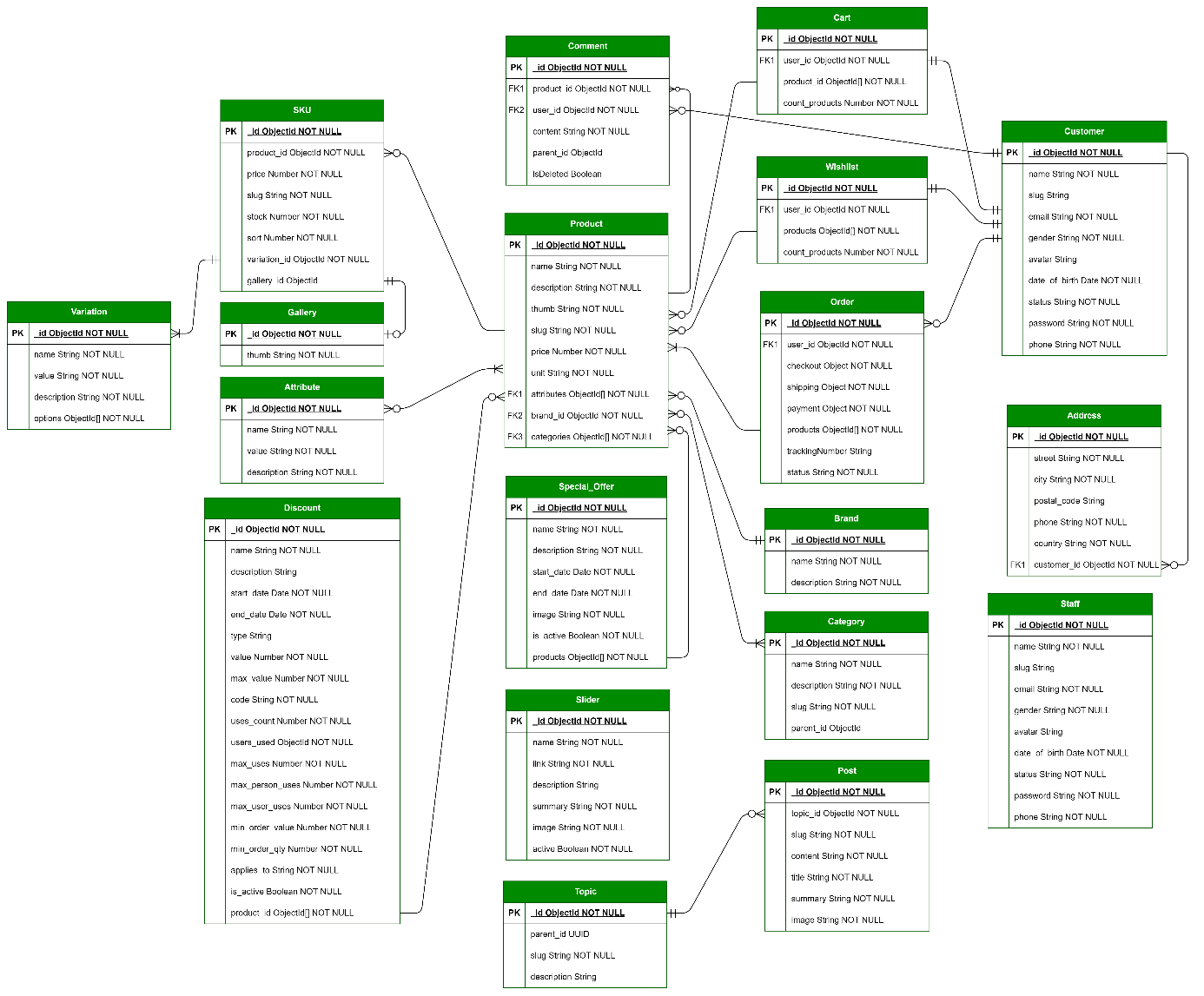
### Biểu đồ tuần tự Thay đổi tình trạng đơn hàng



*Hình 33: Biểu đồ tuần tự thay đổi tình trạng đơn hàng*

## Tổ chức cơ sở dữ liệu

### Database Diagram



*Hình 34: Database Diagram*

### Mô tả chi tiết một số đối tượng chính

* ***Product***

| **STT** | **Thuộc tính** | **Loại** | **Kiểu** | **Ràng buộc** | **Diễn giải** |
| --- | --- | --- | --- | --- | --- |
| 1 | \_id | primary\_key | ObjectId | NOT NULL | Mã sản phẩm |
| 2 | name |  | String | NOT NULL | Tên sản phẩm |
| 3 | description |  | String | NOT NULL | Mô tả |
| 4 | thumb |  | String |  | Đường link hình ảnh |
| 4 | price |  | Number | NOT NULL | Giá của sản phẩm |
| 5 | unit |  | String | NOT NULL | Đơn vị sản phẩm |
| 6 | attributes | foreign\_key | ObjectId[ ] | NOT NULL | Thuộc tính sản phẩm |
| 7 | brand\_id |  | ObjectId | NOT NULL | Nhãn hàng sản phẩm |
| 8 | isPublished |  | Boolean |  | Tình trạng |
| 8 | categories |  | ObjectId[ ] | NOT NULL | Danh mục sản phẩm |

Bảng 3. Bảng thuộc tính product

* ***Category***

| **STT** | **Thuộc tính** | **Loại** | **Kiểu** | **Ràng buộc** | **Diễn giải** |
| --- | --- | --- | --- | --- | --- |
| 1 | \_id | primary\_key | ObjectId | NOT NULL | Mã danh mục |
| 2 | name |  | String | NOT NULL | Tên danh mục |
| 3 | description |  | String | NOT NULL | Mô tả |
| 4 | slug |  | String | NOT NULL | slug |
| 5 | isPublished |  | Boolean |  | Tình trạng |

Bảng 4. Bảng thuộc tính category

* ***Customer***

| **STT** | **Thuộc tính** | **Loại** | **Kiểu** | **Ràng buộc** | **Diễn giải** |
| --- | --- | --- | --- | --- | --- |
| 1 | \_id | primary\_key | ObjectId | NOT NULL | Mã người dùng |
| 2 | name |  | String | NOT NULL | Tên người dùng |
| 3 | email |  | String | NOT NULL | Email người dùng |
| 4 | slug |  | String | NOT NULL | slug |
| 5 | gender |  | String | NOT NULL | Giới tính |
| 6 | avatar |  | String |  | Ảnh đại diện |
| 7 | date\_of\_birth |  | Date | NOT NULL | Ngày sinh |
| 8 | status |  | String | NOT NULL | Tình trạng |
| 9 | password |  | String | NOT NULL | Mật khẩu |
| 10 | phone |  | String | NOT NULL | Số điện thoại |

Bảng 5. Bảng thuộc tính customer

* ***Order***

| **STT** | **Thuộc tính** | **Loại** | **Kiểu** | **Ràng buộc** | **Diễn giải** |
| --- | --- | --- | --- | --- | --- |
| 1 | \_id | primary\_key | ObjectId | NOT NULL | Mã người dùng |
| 2 | user\_id | foreign\_key | ObjectId | NOT NULL | Mã người dùng |
| 3 | checkout |  | Object | NOT NULL | checkout |
| 4 | shipping |  | Object | NOT NULL | shipping |
| 5 | payment |  | Object | NOT NULL | payment |
| 6 | products |  | ObjectId[] | NOT NULL | Sản phẩm |
| 7 | trackingNumber |  | String | NOT NULL | trackingNumber |
| 8 | status |  | String | NOT NULL | Tình trạng |

Bảng 6. Bảng thuộc tính order

* ***Cart***

| **STT** | **Thuộc tính** | **Loại** | **Kiểu** | **Ràng buộc** | **Diễn giải** |
| --- | --- | --- | --- | --- | --- |
| 1 | \_id | primary\_key | ObjectId | NOT NULL | Mã giỏ hàng |
| 2 | user\_id | foreign\_key | ObjectId | NOT NULL | Mã người dùng |
| 3 | products |  | ObjectId[] | NOT NULL | Sản phẩm |
| 4 | count\_products |  | Number | NOT NULL | Số lượng sản phẩm |

Bảng 7. Bảng thuộc tính cart

* ***Brand***

| **STT** | **Thuộc tính** | **Loại** | **Kiểu** | **Ràng buộc** | **Diễn giải** |
| --- | --- | --- | --- | --- | --- |
| 1 | \_id | primary\_key | ObjectId | NOT NULL | Mã nhãn hàng |
| 2 | name |  | String | NOT NULL | Tên nhãn hàng |
| 3 | description |  | String | NOT NULL | Mô tả |

Bảng 8. Bảng thuộc tính brand

* ***Slider***

| **STT** | **Thuộc tính** | **Loại** | **Kiểu** | **Ràng buộc** | **Diễn giải** |
| --- | --- | --- | --- | --- | --- |
| 1 | \_id | primary\_key | ObjectId | NOT NULL | Mã slider |
| 2 | name |  | String | NOT NULL | Tên |
| 3 | description |  | String | NOT NULL | Mô tả |
| 4 | link |  | String | NOT NULL | Đường dẫn |
| 5 | summary |  | String | NOT NULL | Tóm tắt |
| 6 | image |  | String | NOT NULL | Hình ảnh |
| 7 | active |  | Boolean | NOT NULL | Tình trạng |

Bảng 9. Bảng thuộc tính slider

# TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

## Giao diện trang người dùng

### Trang chủ

### Đăng nhập

### Đăng kí

### Giỏ hàng

### Trang bài viết

### Trang mua hàng

### Tìm kiếm

### Thông tin người dùng

### Thông tin giỏ hàng

### Trang chi tiết sản phẩm

## Giao diện trang quản trị

### Quản lý người dùng

### Quản lý sản phẩm

### Thêm Sản phẩm

### Quản lý thương hiệu

### Thêm thương hiệu

# TỔNG KẾT

## Kết quả đạt được

Đề tài “**Xây dựng web bán hàng mỹ phẩm Hoàng Vũ**” cũng xuất phát từ thực tế ngày nay nhằm tạo một nền tảng cơ sở ban đầu để có thể hỗ trợ thêm cho các bạn muốn thiết kế một Web cho doanh nghiệp hay tư nhân có yêu cầu.

Mặc dù đã có nhiều cố gắng, tìm hiểu các kiến thức đã học, kết hợp tra cứu các tài liệu chuyên ngành nhưng do hạn chế về thời gian, khả năng và kinh nghiệm nên không tránh khỏi những thiếu sót nhất định và kết quả thu được như sau:

* **Áp dụng các công nghệ**: NodeJs, React JS,...
* **Xây dựng web bán hàng mỹ phẩm Hoàng Vũ**: Sữa rửa mặt, kem dưỡng, tẩy trang, ….
* **Hoàn thành các chức năng cho khách hàng**: Đăng nhập, đăng ký, tìm kiếm, đặt hàng, thêm giỏ hàng, xem thông tin (sản phẩm, tin tức,...), quản lý thông tin cá nhân( thay đổi thông tin, theo dõi đơn hàng), liên hệ.
* **Hoàn thành các chức năng cho quản trị**: Thêm, xóa, cập nhật thông tin (sản phẩm, tin tức, danh mục,...), kiểm soát đơn hàng.

## Hạn chế

* Một số giao diện còn đơn giản, không cuốn hút người xem, hạn chế thông tin.
* Chưa xây dựng quảng cáo,….
* Không có phần bình luận sản phẩm.

## Hướng phát triển

* Tìm hiểu sâu hơn về NodeJs cụ thể là Express Framework để có thể đáp ứng nhiều hơn nữa nhu cầu của người sử dụng, phát triển và tối ưu hóa hệ thống.
* Tìm hiểu thêm về React JS và các thành phần thư viện ứng dụng để nâng cao việc thiết kế và xây dựng giao diện đẹp mắt, thân thiện hơn với người dùng.
* Xây dựng trang Web quy mô lớn hơn với nhiều ứng dụng, chức năng để đáp ứng đầy đủ nhu cầu người sử dụng.
* Áp dụng nhiều công nghệ mới vào ứng dụng web.

# TÀI LIỆU THAM KHẢO

1. https://topdev.vn/blog/react-la-gi-lo-trinh/ :ReactJS
2. https://www.w3schools.com/html/html\_scripts.asp: JAVASCRIPT
3. https://topdev.vn/blog/restful-api-la-gi/ : Restful API
4. [Express - Node.js web application framework (expressjs.com)](https://expressjs.com/) / : Express Framework
5. https://www.postman.com/ : Postman
6. https://nodejs.org/en/learn/getting-started/introduction-to-nodejs : NodeJS