

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BTL PHÁT TRIỂN HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ

Giảng viên hướng dẫn	: KIM NGỌC BÁCH
Danh sách thành viên	:
Phạm Minh Đức	B21DCCN254
Lê Trí Đức	B21DCCN242
Cao Minh Sơn	B21DCCN650
Nhóm	: G1_18_final

Hà Nội – 2025

Mục lục

I. Tổng quan về hệ thống:	5
1. Giới thiệu đề tài:	5
2. Mô hình kinh doanh:	5
3. Mô hình doanh thu:	5
II. Cơ sở lý thuyết:	5
1. Mô hình kiến trúc:	5
2. Ngôn ngữ và thư viện:	6
a) Front-End:	6
b) Back-End:	7
1) Node.js	7
2) Express	8
c) Database:	9
d) Một số thư viện khác:	10
III. Các tính năng sẽ thiết kế và triển khai:	11
1. Đối với người dùng (Customer):	11
2. Đối với quản trị (Admin):	13
IV. Thiết kế hệ thống:	14
1. Biểu đồ Use Case	14
1.1) Use Case tổng quát	14
1.2) Use Case chi tiết	15
1.2.1) Đối với người dùng	15
a) Đăng nhập	15
b) Tìm kiếm	16
c) nhắn tin	16
1.2.2) Đối với admin	17
a) quản lý nhân viên	17
b) quản lý khách hàng	17
c) quản lý sản phẩm	17
d) quản lý đơn hàng	18
1.2.3) Đối với khách hàng	18

a) Quản lý thông tin cá nhân.....	18
b) xem, tìm kiếm, đánh giá,thêm sản phẩm vào giỏ hàng.....	19
c) đặt hàng, hoàn hàng, trả hàng	19
2) Biểu đồ tuần tự.....	20
2.1) Đối với người dùng.....	20
a)Đăng nhập.....	20
b) Tìm kiếm.....	20
c)Nhấn tin.....	20
2.2) Đối với người quản lý	21
a)Quản lý nhân viên	21
b)Quản lý sản phẩm	21
c) Quản lý Khách hàng	21
d)Quản lý đơn hàng	22
2.3. Đối với khách hàng.....	22
a)Quản lý thông tin cá nhân.....	22
b) xem, tìm kiếm, đánh giá,thêm sản phẩm vào giỏ hàng.....	22
c) đặt hàng, hoàn hàng, trả hàng	23
IV. Kế hoạch triển khai:.....	24
4.1.Tổng quan về phân chia công việc.....	24
4.2.Bản phân công công việc chi tiết.....	24
4.3Giao diện hệ thống	28
V.Kết luận và định hướng phát triển	30

Thành viên	Tỷ lệ hoàn thành công việc
Lê Trí Đức	100%
Cao Minh Sơn	100%
Phạm Minh Đức	100%

I. Tổng quan về hệ thống:

1. Giới thiệu đề tài:

Trong bối cảnh công nghệ ngày càng phát triển và nhu cầu tiêu dùng trực tuyến ngày càng tăng cao, việc xây dựng một hệ thống đặt đồ ăn trực tuyến đa tính năng là cần thiết nhằm đáp ứng nhu cầu thực tế của người dùng. Hệ thống này đóng vai trò là một nền tảng trung gian giữa khách hàng (người tiêu dùng), nhà hàng (doanh nghiệp cung cấp món ăn) và quản trị viên (admin) nhằm đảm bảo quy trình đặt – chuẩn bị – giao hàng được vận hành trơn tru.

Người dùng có thể dễ dàng truy cập website, tìm kiếm và lựa chọn các món ăn yêu thích, sau đó tiến hành đặt hàng một cách nhanh chóng và tiện lợi. Nhà hàng có thể tiếp nhận, xử lý đơn hàng và cập nhật trạng thái giao hàng theo thời gian thực. Quản trị viên có quyền kiểm soát toàn bộ hệ thống như theo dõi doanh thu, quản lý người dùng, nhà hàng và phân tích hoạt động kinh doanh.

2. Mô hình kinh doanh:

Hệ thống được phát triển dựa trên mô hình B2C (Business to Consumer), trong đó doanh nghiệp (các nhà hàng, quán ăn) cung cấp dịch vụ ẩm thực trực tiếp đến người tiêu dùng cuối thông qua nền tảng web.

- Người tiêu dùng có thể xem thực đơn, đặt món và theo dõi quá trình xử lý đơn hàng.
- Nhà hàng nhận thông tin đơn hàng, chuẩn bị món ăn, phối hợp với đơn vị vận chuyển để giao hàng đến tay người tiêu dùng.
- Toàn bộ giao dịch, tương tác giữa các bên được thực hiện trên nền tảng trực tuyến, giúp tối ưu hóa trải nghiệm người dùng.

3. Mô hình doanh thu:

Hệ thống tạo doanh thu từ hai nguồn chính:

- Bán hàng trực tiếp: Người tiêu dùng trả tiền cho các món ăn được đặt thông qua hệ thống. Một phần lợi nhuận sẽ được chia cho nền tảng (nếu hệ thống là trung gian).
 - Phí giao hàng: Tính phí vận chuyển đối với người tiêu dùng, hệ thống có thể hưởng một phần hoa hồng từ đơn vị vận chuyển hoặc nhà hàng.
- Trong tương lai, hệ thống có thể mở rộng thêm các mô hình doanh thu khác như: đăng quảng cáo món ăn, gói thành viên ưu đãi cho nhà hàng, chương trình khuyến mãi tính phí, hoặc tính phí quản lý theo tháng.

II. Cơ sở lý thuyết:

1. Mô hình kiến trúc:

Hệ thống được xây dựng dựa trên mô hình kiến trúc Client – Server kết hợp với mô hình MVC (Model – View – Controller) nhằm phân tách rõ ràng các thành phần trong ứng dụng, tăng khả năng mở rộng và bảo trì.

Client (Frontend)

- Được xây dựng bằng ReactJS – một thư viện JavaScript phổ biến, mạnh mẽ, hỗ trợ tạo giao diện người dùng theo hướng component.
- Gửi yêu cầu HTTP hoặc HTTPS (REST API) đến phía server.
- Nhận dữ liệu JSON từ server và hiển thị trực quan cho người dùng.

Server (Backend - theo MVC)

Model:

- Chịu trách nhiệm tương tác trực tiếp với cơ sở dữ liệu như MongoDB thông qua thư viện Mongoose.
- Thực hiện các thao tác thêm, sửa, xóa, truy vấn dữ liệu.

View:

- Trong hệ thống sử dụng React ở phía client, View phía backend chủ yếu trả về JSON để frontend xử lý và hiển thị.
- Trong trường hợp cần thiết, có thể dùng template engine (như EJS) để render giao diện phía server.

Controller:

- Nhận và xử lý các yêu cầu từ phía client.
- Gọi Model để truy xuất dữ liệu, xử lý logic nghiệp vụ và trả lại dữ liệu cho client.

2. Ngôn ngữ và thư viện:

a) Front-End:

Trang web sử dụng thư viện ReactJs để phát triển giao diện người dùng. ReactJS là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web.

Một số đặc điểm nổi bật của ReactJs:

- **Component-based:** UI được chia nhỏ thành các thành phần (components), dễ quản lý và tái sử dụng.
- **Virtual DOM:** Tăng hiệu năng bằng cách so sánh sự thay đổi và cập nhật DOM một cách tối ưu.

- One-way data binding: Dữ liệu truyền từ component cha xuống component con giúp kiểm soát luồng dữ liệu dễ hơn.
- JSX (JavaScript XML): Cú pháp mở rộng cho phép viết HTML trong JavaScript.

b) Back-End:

Được xây dựng dựa trên NodeJs và ExpressJs.

1) Node.js

Node.js là một môi trường thực thi JavaScript phía máy chủ (server-side runtime environment), cho phép bạn chạy JavaScript bên ngoài trình duyệt. Ban đầu, JavaScript chỉ chạy được trên trình duyệt, nhưng Node.js đã mở rộng nó thành một ngôn ngữ dùng được cả cho backend. Điều này cho phép lập trình viên sử dụng JavaScript cho cả frontend và backend, giúp thống nhất ngôn ngữ và tăng hiệu quả phát triển.

Đặc điểm nổi bật của Node.js:

Nguồn mở (Open-source)

- Mã nguồn Node.js được công khai.
- Bất kỳ ai cũng có thể xem, sử dụng, đóng góp.

Đa nền tảng (Cross-platform)

- Node.js chạy được trên Windows, Linux, macOS mà không cần chỉnh sửa mã nguồn.

Dựa trên V8 JavaScript Engine

- Node.js sử dụng V8 Engine của Google (dùng trong Chrome) để thực thi mã JavaScript nhanh chóng và hiệu quả.

Môi trường thực thi JavaScript ngoài trình duyệt

- Node.js mở rộng khả năng của JavaScript, cho phép bạn chạy code ngoài môi trường trình duyệt (trên máy chủ, máy tính cá nhân).

Node.js hoạt động như thế nào?

Kiến trúc Non-blocking I/O và Event-Driven

- Các thao tác I/O (đọc file, gọi API, truy vấn DB) không làm chặn chương trình.
- Node.js sử dụng mô hình sự kiện (event-driven) để xử lý bất đồng bộ.

Single-threaded nhưng hiệu quả

- Node.js sử dụng một luồng chính để xử lý logic người dùng.
- Dưới tầng thấp, Node.js dùng libuv để xử lý I/O bất đồng bộ bằng các luồng nền (background threads).

Event Loop (Vòng lặp sự kiện)

- Là trái tim của Node.js.
- Nó liên tục lắng nghe và xử lý các sự kiện I/O mà không chặn luồng chính.

Trigger Callback

- Khi thao tác I/O hoàn tất, Node.js gọi lại (callback) các hàm xử lý kết quả.

2) Express

Express là một framework giúp đơn giản hóa quá trình xây dựng máy chủ HTTP bằng Node.js. Thay vì phải sử dụng trực tiếp module http của Node để xử lý request, response, routing,... Express cung cấp một lớp trừu tượng tiện lợi giúp viết code ngắn gọn và rõ ràng hơn.

Cách thức hoạt động

Express hoạt động dựa trên các thành phần sau:

- **Middleware:** Là các hàm được thực thi tuần tự trong quá trình xử lý request. Mỗi middleware có thể sửa đổi request, response hoặc kết thúc chuỗi xử lý.
- **Router:** Xác định các tuyến đường (route) để xử lý các yêu cầu HTTP (GET, POST, PUT, DELETE...) đến từ client.
- **Request/Response Object:** Express mở rộng các đối tượng req và res của Node để dễ thao tác hơn, như req.body, res.json(), res.send()...
- **Template engine (tùy chọn):** Hỗ trợ render HTML từ dữ liệu nếu cần tạo giao diện động, ví dụ như sử dụng EJS, Pug.

Đặc điểm nổi bật của Express

- **Đơn giản và dễ sử dụng**
Cung cấp API đơn giản giúp xây dựng web server hoặc REST API nhanh chóng.
- **Nhẹ và linh hoạt**
Không ràng buộc theo cấu trúc cứng nhắc, cho phép bạn tự do thiết kế ứng dụng theo ý muốn.
- **Hệ thống middleware mạnh mẽ**
Có thể dễ dàng thêm các lớp xử lý như kiểm tra xác thực, xử lý lỗi, phân tích body request,...

- Hỗ trợ routing rõ ràng
Cung cấp cách định nghĩa tuyến đường (route) cụ thể cho các loại yêu cầu HTTP khác nhau.
- Hệ sinh thái phong phú
Có thể kết hợp với hàng loạt thư viện và middleware như cors, body-parser, morgan, jsonwebtoken, multer,...
- Tích hợp tốt với các template engine
Hỗ trợ render HTML động thông qua các công cụ như EJS, Pug, Handlebars,...
- Dễ mở rộng và bảo trì
Có thể chia nhỏ ứng dụng thành nhiều route module, giúp tổ chức và bảo trì code tốt hơn.

c) Database:

Mongoose giúp bạn định nghĩa schema (lược đồ) cho các tài liệu trong MongoDB, quản lý mối quan hệ giữa các dữ liệu, kiểm tra dữ liệu đầu vào, và trừu tượng hóa các thao tác CRUD (Create, Read, Update, Delete).

Nói cách khác, thay vì làm việc với dữ liệu MongoDB thô thông qua MongoClient, bạn sử dụng các mô hình (models) để thao tác dữ liệu một cách có tổ chức và dễ kiểm soát hơn.

Cách thức hoạt động

- Schema
Là bản thiết kế cho tài liệu MongoDB. Schema định nghĩa các trường (field), kiểu dữ liệu, ràng buộc (validation), giá trị mặc định,...
- Model
Là một constructor được tạo ra từ schema. Model đại diện cho một collection trong MongoDB, và là nơi để thực hiện các thao tác như tạo, đọc, cập nhật và xóa.
- Document
Là một bản ghi cụ thể trong collection, được tạo từ model. Bạn có thể thao tác với nó như một đối tượng JavaScript.
- Query
Mongoose cung cấp cú pháp đơn giản để truy vấn dữ liệu, hỗ trợ promise hoặc async/await.

Đặc điểm nổi bật của Mongoose

Tạo schema rõ ràng và có tổ chức

Giúp định nghĩa cấu trúc dữ liệu một cách chặt chẽ, kiểm soát tốt dữ liệu đầu vào.

Hỗ trợ validation (ràng buộc dữ liệu)

Có thể dễ dàng kiểm tra dữ liệu hợp lệ trước khi lưu vào cơ sở dữ liệu.

Hỗ trợ middleware (hooks)

Có thể thêm các hành động trước hoặc sau khi lưu, cập nhật hoặc xóa tài liệu, ví dụ như pre('save'), post('remove').

Query đơn giản và mạnh mẽ

Truy vấn theo điều kiện dễ dàng, hỗ trợ lọc, sắp xếp, phân trang,...

Tích hợp tốt với Node.js và Express

Là sự lựa chọn phổ biến khi kết hợp với Express để xây dựng các RESTful API.

Hỗ trợ population (liên kết dữ liệu)

Cho phép liên kết giữa các collection, giống như foreign key trong SQL.

d) Một số thư viện khác:

JWT (JSON Web Token) : Sau khi người dùng đăng nhập thành công, server sẽ trả về token để client lưu trữ. Các yêu cầu sau đó được xác thực bằng cách đính kèm token.

Bcrypt : Dùng thuật toán băm để mã hóa mật khẩu, tránh lộ thông tin nhạy cảm khi database bị rò rỉ.

multer: Dùng để xử lý việc upload tệp (ví dụ: ảnh món ăn, ảnh đại diện người dùng, v.v.). Multer phân tích các biểu mẫu có enctype="multipart/form-data" và lưu trữ file tạm thời hoặc lâu dài theo cấu hình.

cors: Cho phép ứng dụng client (ReactJS) gửi request đến server (NodeJS) trên một domain khác (hữu ích khi phát triển frontend và backend độc lập).

dotenv: Hỗ trợ quản lý biến môi trường. Thông tin nhạy cảm như chuỗi kết nối cơ sở dữ liệu, khóa API... được lưu trong file .env và không bị đẩy lên Git.

validator: Cung cấp các hàm xác thực dữ liệu đầu vào như email, URL, số điện thoại... giúp tăng độ tin cậy và bảo mật của hệ thống.

uuid: Tạo mã định danh (ID) duy nhất cho các đối tượng trong hệ thống (đơn hàng, người dùng, hóa đơn...) nhằm giảm khả năng trùng lặp.

moment: Thư viện thao tác ngày giờ mạnh mẽ, hỗ trợ định dạng, so sánh, tính toán khoảng thời gian... rất phù hợp cho các tính năng như thống kê, lịch sử đặt món, trạng thái đơn hàng theo thời gian.

nodemailer: Hỗ trợ gửi email từ server. Dùng để gửi thông báo đơn hàng, mã xác thực email, hoặc mã khôi phục mật khẩu.

socket.io: Cho phép giao tiếp thời gian thực giữa server và client. Hệ thống có thể dùng để:

- Cập nhật trạng thái đơn hàng theo thời gian thực.
- Chat giữa người dùng và quản trị viên hoặc nhà hàng.

ngrok: Dùng để tạo đường hầm (tunnel) từ máy tính cục bộ lên internet. Hữu ích trong giai đoạn phát triển để kiểm thử webhook (Stripe, VNPay...) và truy cập server nội bộ từ xa.

@fortawesome/react-fontawesome và các gói biểu tượng: Cung cấp bộ icon phong phú và đẹp mắt để hiển thị trên giao diện React, giúp nâng cao trải nghiệm người dùng.

III. Các tính năng sẽ thiết kế và triển khai:

1. Đối với người dùng (Customer):

Người dùng cuối là các khách hàng có nhu cầu đặt món ăn trực tuyến. Họ sẽ có quyền truy cập và sử dụng các chức năng sau:

Đăng ký / Đăng nhập / Đăng xuất

- Tạo tài khoản bằng email, mật khẩu.
- Hệ thống xác thực bằng token JWT, đảm bảo an toàn khi đăng nhập.
- Hỗ trợ đăng xuất, xóa token khỏi localStorage.

Duyệt món ăn

- Hiển thị danh sách các món ăn đang được bán.
- Hỗ trợ tìm kiếm theo từ khóa.
- Lọc theo các tiêu chí: loại món ăn (cơm, nước, chay, v.v.), giá, đánh giá

Xem thông tin món ăn

- Trang chi tiết món ăn với thông tin như: tên, mô tả, giá, hình ảnh, đánh giá từ người dùng khác.

Quản lý giỏ hàng

- Thêm, xóa, cập nhật số lượng món ăn.
- Tính tổng tiền tạm tính, thuế, giảm giá.

Đặt hàng

- Thực hiện quy trình xác nhận giỏ hàng và tạo đơn hàng mới.
- Gửi thông báo xác nhận đơn hàng qua email (sử dụng nodemailer).

Thanh toán

- **Trực tuyến:** tích hợp cổng thanh toán như Stripe, VNPay hoặc Momo.
- **COD (Cash on Delivery):** thanh toán khi nhận hàng.

Quản lý đơn hàng

- Theo dõi tình trạng đơn hàng theo các trạng thái:
- "Đang xử lý"
- "Đang giao"
- "Đã giao"
- Cho phép hủy đơn hàng nếu chưa giao.
- Đặt lại món từ đơn hàng cũ (reorder).

Xem lịch sử đơn hàng: ngày đặt, tổng tiền, trạng thái, chi tiết.

Đánh giá sản phẩm

- Gửi đánh giá với sao (1–5) và nội dung.
- Lọc đánh giá theo mức sao.

Quản lý thông tin cá nhân

- Cập nhật thông tin: tên, email, số điện thoại, địa chỉ.
- Đổi mật khẩu.
- Giao diện thân thiện, xác thực bảo mật khi thay đổi thông tin nhạy cảm.

Đổi trả – Hoàn tiền

- Gửi yêu cầu đổi trả hàng trong thời gian cho phép.
- Gửi yêu cầu hoàn tiền nếu đơn hàng có lỗi hoặc giao sai món.
- Theo dõi trạng thái xử lý hoàn tiền.

Chat với nhân viên hỗ trợ

- Giao tiếp thời gian thực với quản trị viên hoặc nhân viên qua socket.io.
- Gửi thắc mắc, hỗ trợ nhanh về món ăn, đơn hàng, vấn đề giao hàng.

Tìm kiếm thông minh

- Gợi ý khi nhập từ khóa.

- Kết hợp từ khóa với bộ lọc (lọc nhanh theo danh mục, giá...).

2. Đối với quản trị (Admin):

Quản trị viên đóng vai trò là người kiểm soát toàn bộ hệ thống, từ quản lý dữ liệu đến xử lý yêu cầu và tương tác với khách hàng. các chức năng chính bao gồm:

quản lý món ăn

- thêm món ăn mới: nhập tên, mô tả, giá, loại, hình ảnh, tình trạng khuyến mãi.
- chỉnh sửa món ăn: cập nhật thông tin món đã có.
- xóa món ăn: gỡ bỏ món ăn không còn phục vụ khỏi hệ thống.
- tìm kiếm, phân loại món ăn: theo tên, loại, giá, số lượt đặt hàng, trạng thái bán.

quản lý đơn hàng

- xem danh sách đơn hàng: bao gồm thông tin khách hàng, món ăn, tổng tiền, trạng thái, thời gian đặt.
- cập nhật trạng thái đơn hàng:
 - "đang xử lý" → "đang giao" → "đã giao".
 - hủy đơn nếu có sự cố

quản lý tài khoản

- đối với nhân viên:
 - tạo tài khoản nhân viên mới.
 - gán quyền (admin / nhân viên thường).
 - xóa hoặc khóa tài khoản khi cần.
- đối với khách hàng:
 - xem danh sách khách hàng đã đăng ký.

thống kê – báo cáo

- thống kê doanh thu: theo ngày, tuần, tháng, năm.

xử lý đổi trả – hoàn tiền

- xem yêu cầu từ khách hàng.
- phê duyệt hoặc từ chối yêu cầu.
- cập nhật trạng thái xử lý đổi trả.
- gửi phản hồi cho khách hàng.

quản lý sản phẩm khuyến nghị

- tự thêm sản phẩm khuyến nghị trên trang chủ .

hệ thống chatbox với khách hàng

- trò chuyện thời gian thực với khách hàng thông qua chatbox (tích hợp websocket).
- lưu lịch sử trò chuyện để tiện theo dõi và hỗ trợ sau.

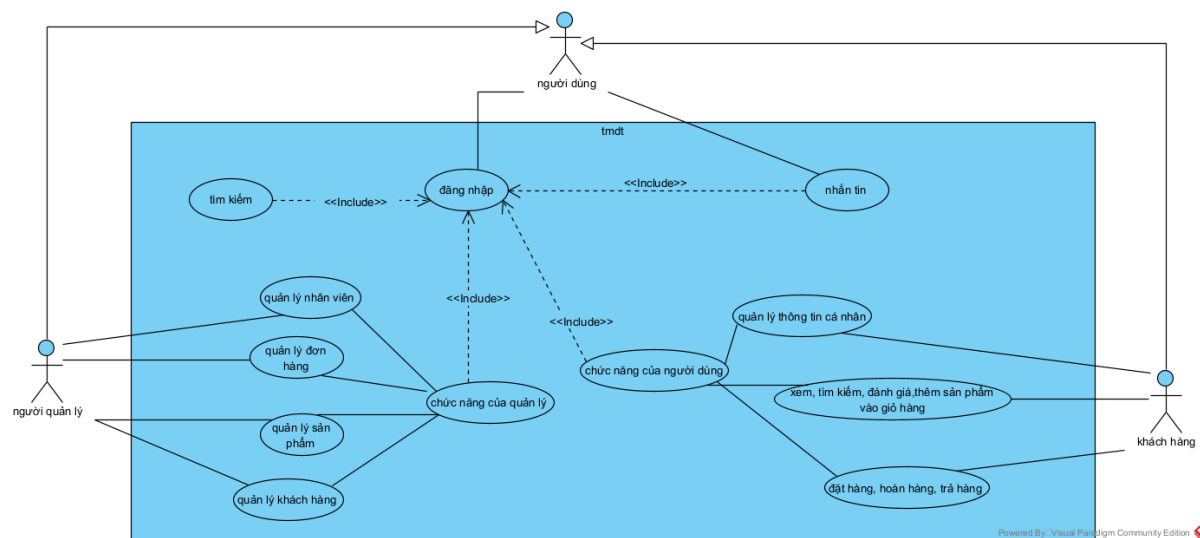
tìm kiếm nâng cao

- tìm kiếm nhanh: món ăn, đơn hàng, người dùng.
- tìm kiếm kết hợp bộ lọc: trạng thái đơn hàng, ngày đặt, loại món ăn,...

IV.Thiết kế hệ thống

1.Biểu đồ Use Case

1.1)Use Case tổng quát



Mô tả Use Case:

Người dùng: Use Case đại diện cho từng thành viên trong hệ thống bao gồm cả nhân viên, quản lý và khách hàng

Người quản lý: Use Case đại diện cho người quản lý

Khách hàng: Use Case đại diện cho khách hàng

Đăng nhập: Use Case đại diện cho chức năng đăng nhập

Tìm kiếm: Use Case đại diện cho chức năng tìm kiếm

Nhắn tin: Use Case đại diện cho chức năng nhắn tin

Quản lý đơn hàng: Use Case đại diện cho chức năng quản lý đơn hàng của admin

Quản lý sản phẩm: Use Case đại diện cho chức năng quản lý sản phẩm, của admin

Quản lý nhân viên: Use Case đại diện cho chức năng quản lý nhân viên của admin

Quản lý khách hàng: Use Case đại diện cho chức năng quản lý toàn bộ khách hàng của admin

Xem, tìm kiếm, đánh giá, thêm sản phẩm vào giỏ hàng: Use Case đại diện cho chức năng về sản phẩm của khách hàng như xem, tìm kiếm, thêm sản phẩm vào giỏ hàng, đánh giá sản phẩm

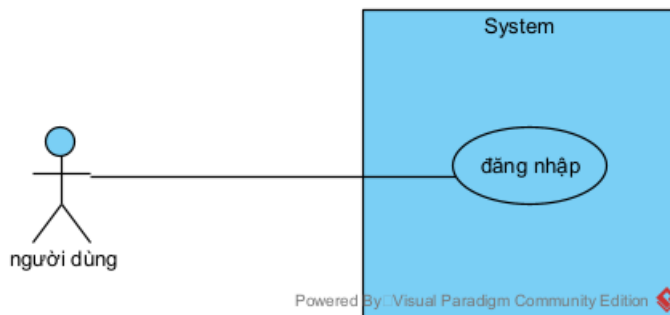
Đặt hàng, hoàn hàng, trả hàng: Use Case đại diện cho chức năng về đơn hàng của khách hàng

Quản lý thông tin cá nhân: Use Case đại diện cho chức năng quản lý thông tin của khách hàng

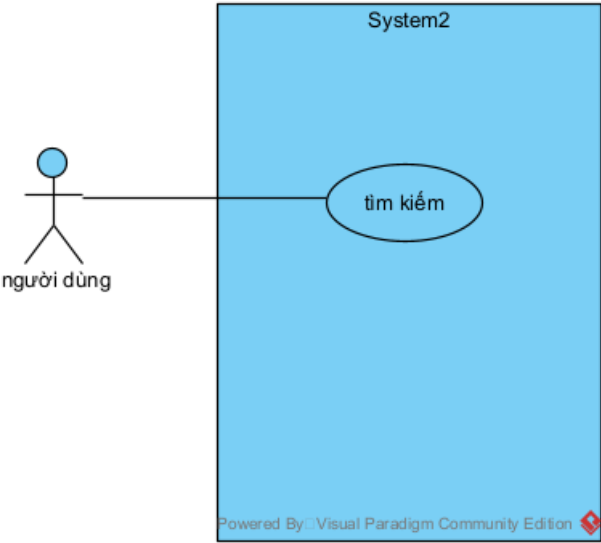
1.2) Use Case chi tiết

1.2.1) Đối với người dùng

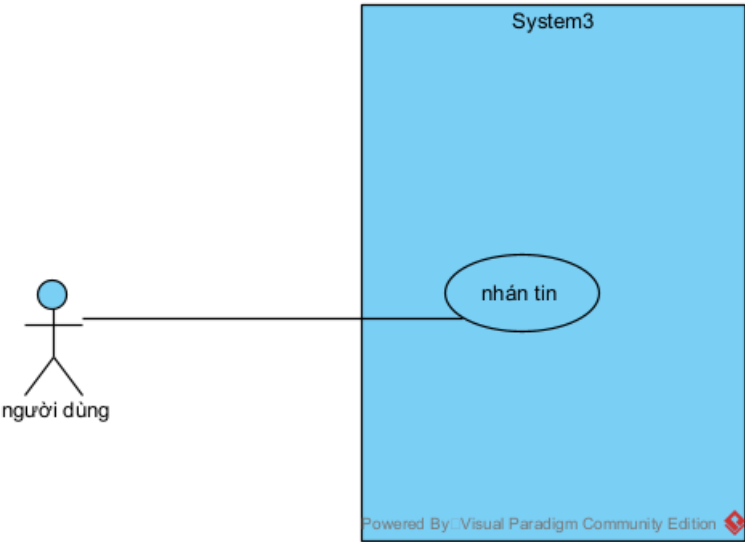
a) Đăng nhập



b) Tìm kiếm

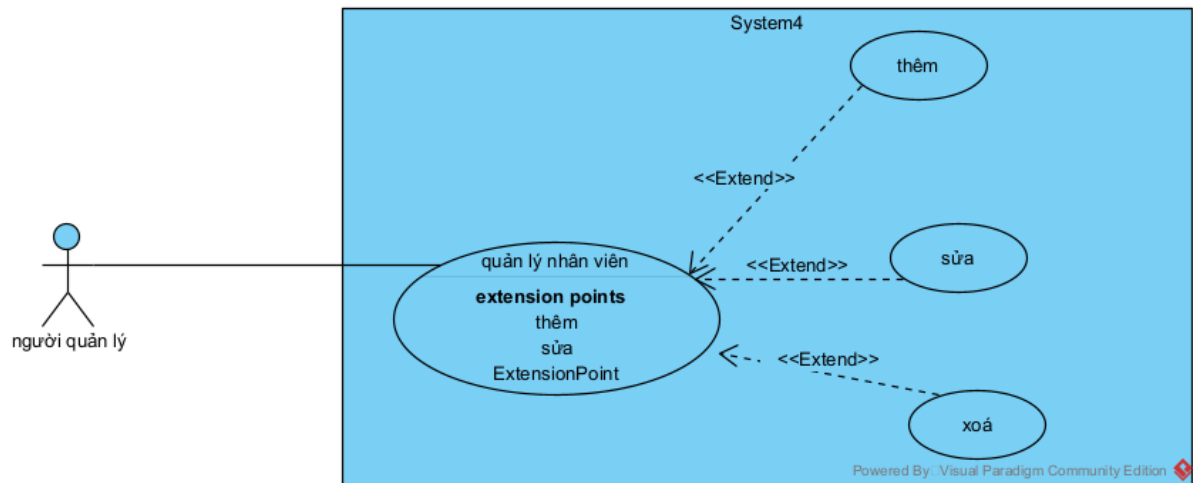


c) nhắn tin

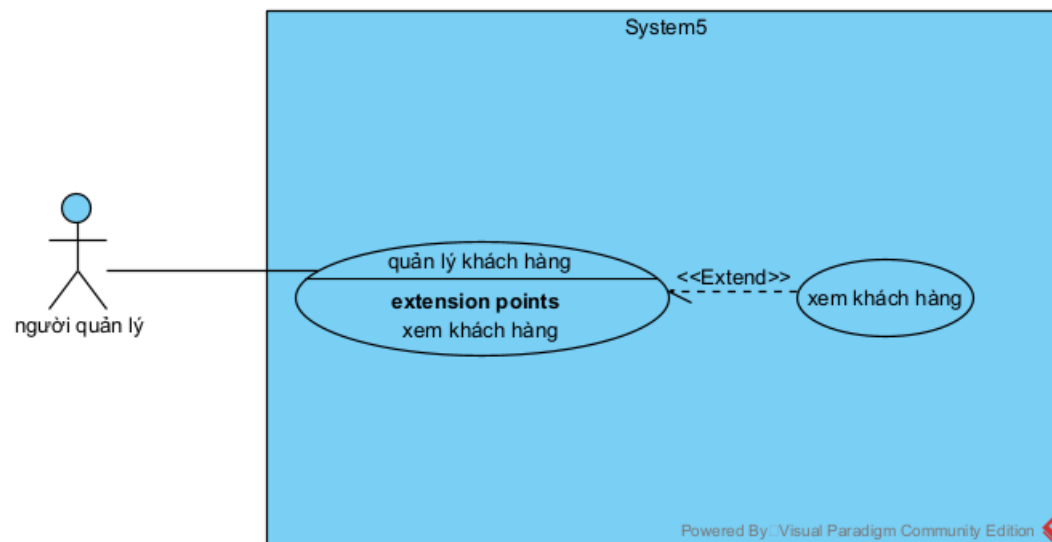


1.2.2) Đối với admin

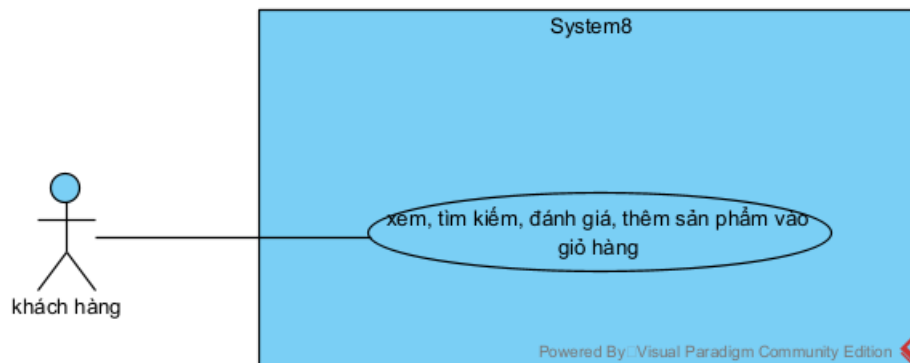
a) quản lý nhân viên



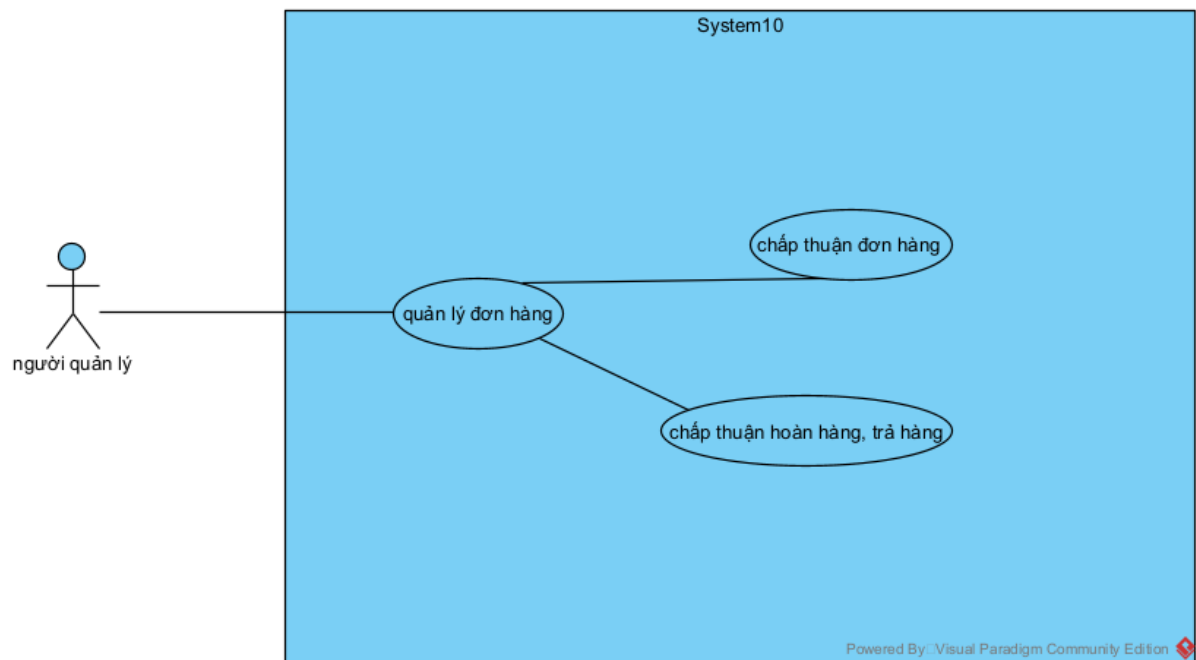
b) quản lý khách hàng



c) quản lý sản phẩm

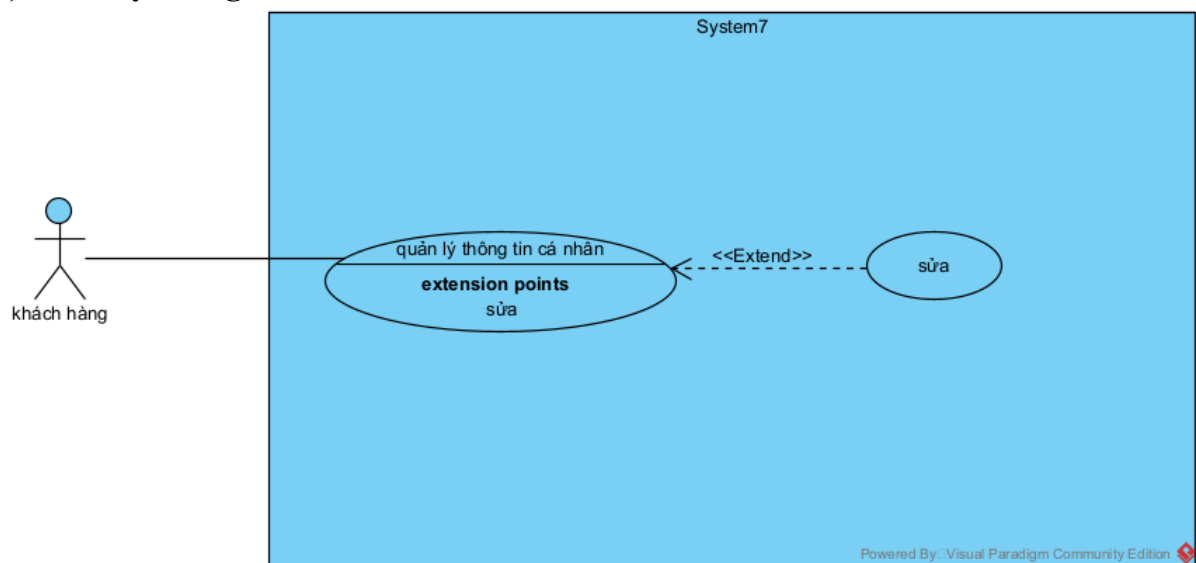


d) quản lý đơn hàng

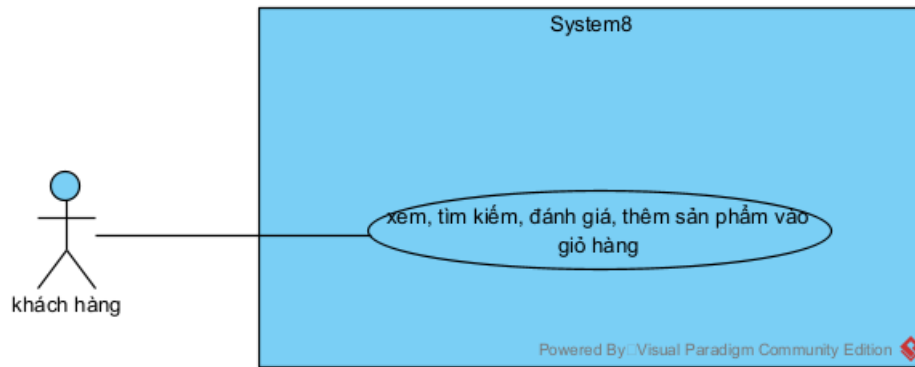


1.2.3) Đối với khách hàng

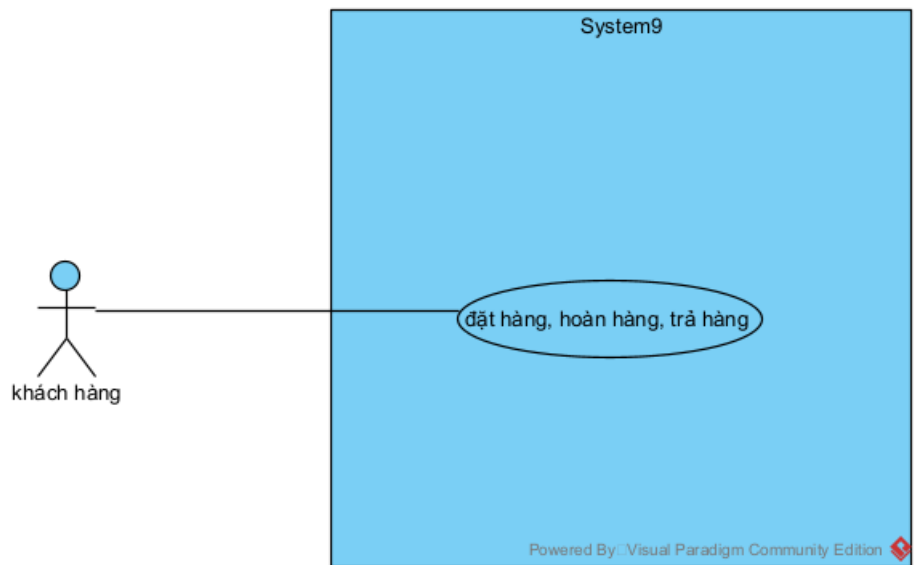
a) Quản lý thông tin cá nhân



b) xem, tìm kiếm, đánh giá, thêm sản phẩm vào giỏ hàng



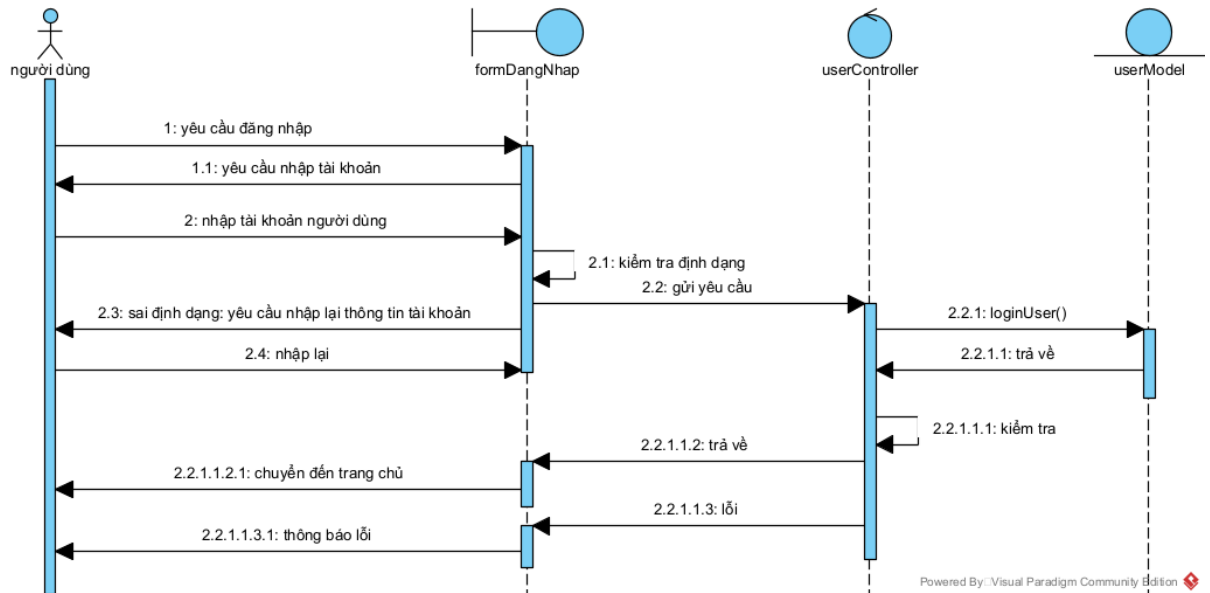
c) đặt hàng, hoàn hàng, trả hàng



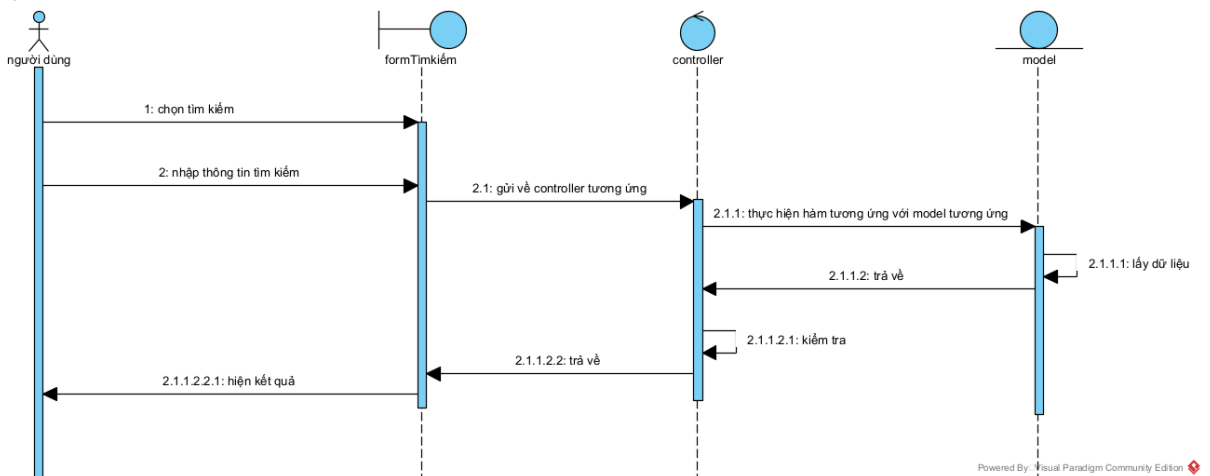
2) Biểu đồ tuần tự

2.1) Đối với người dùng

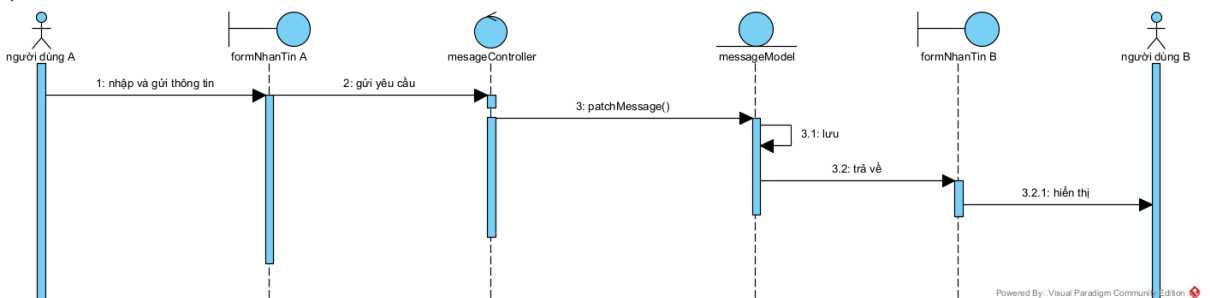
a) Đăng nhập



b) Tìm kiếm

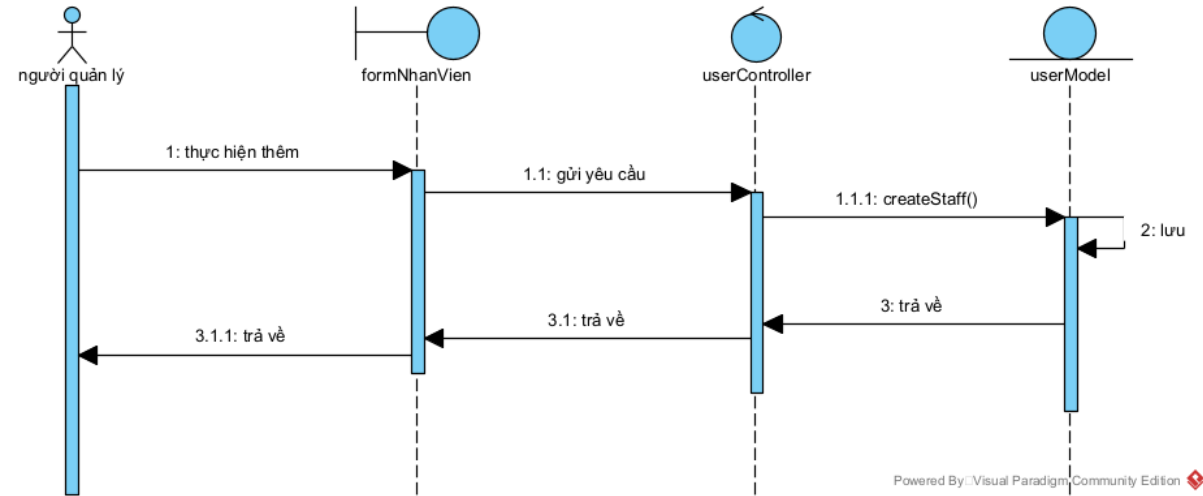


c) Nhắn tin

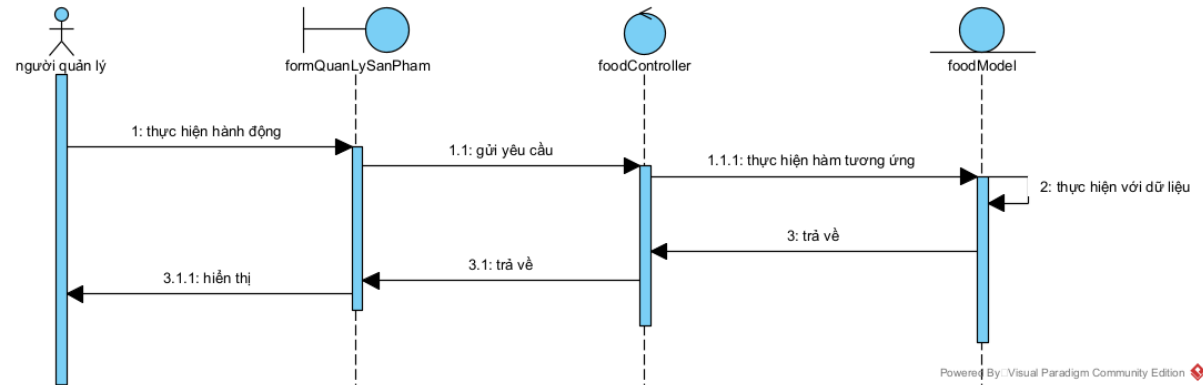


2.2) Đối với người quản lý

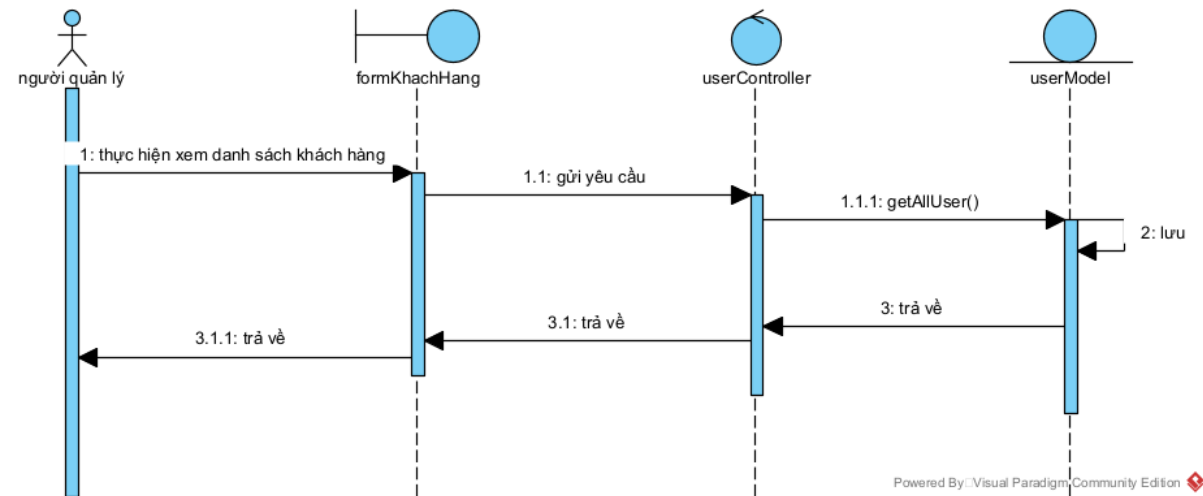
a) Quản lý nhân viên



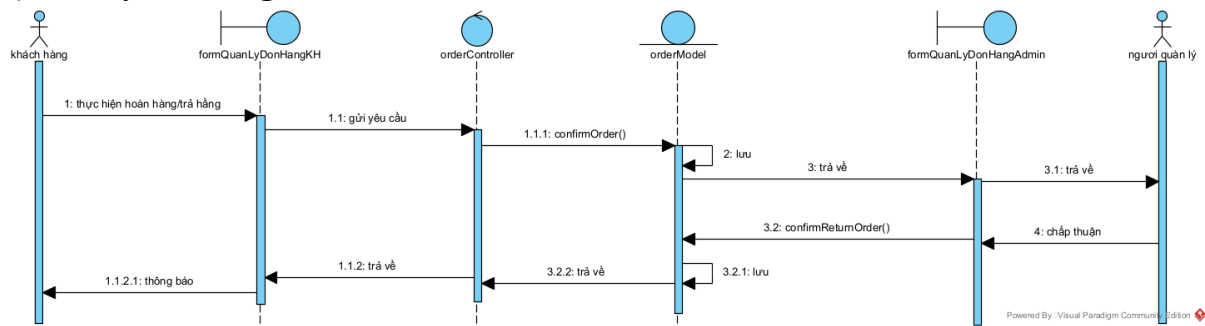
b) Quản lý sản phẩm



c) Quản lý Khách hàng

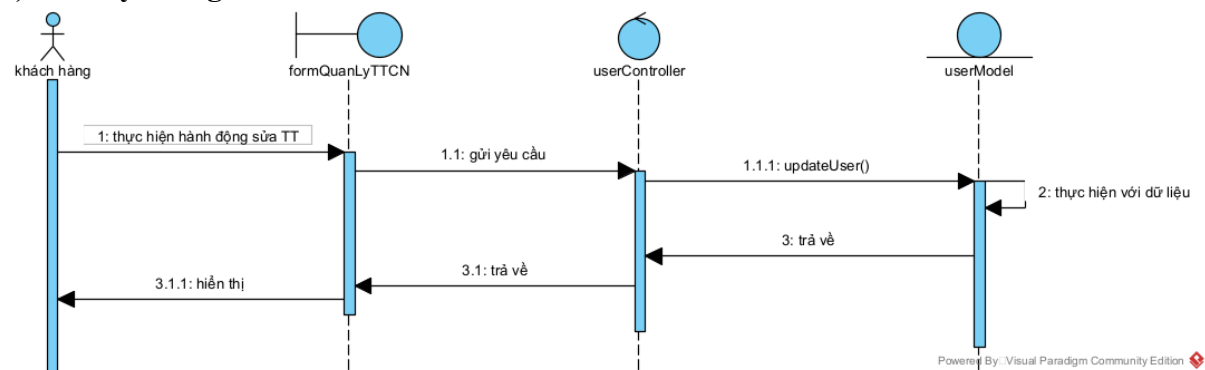


d) Quản lý đơn hàng



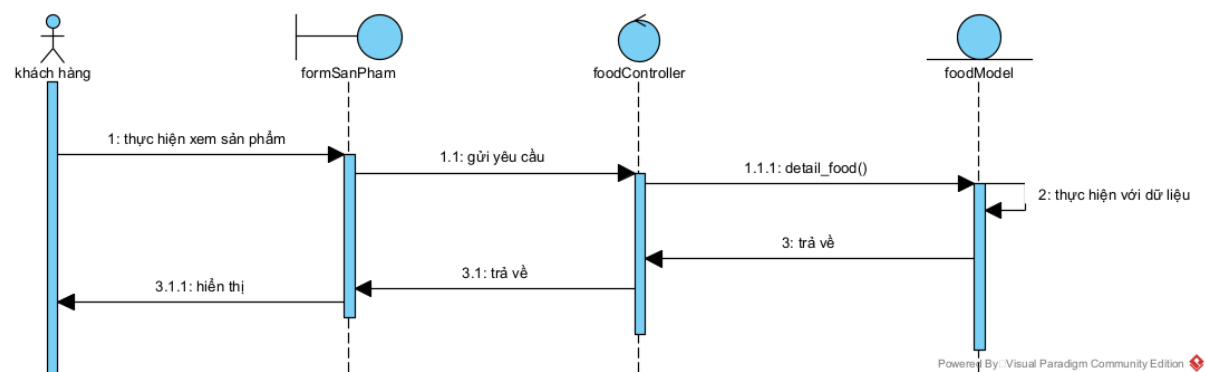
2.3. Đối với khách hàng

a) Quản lý thông tin cá nhân

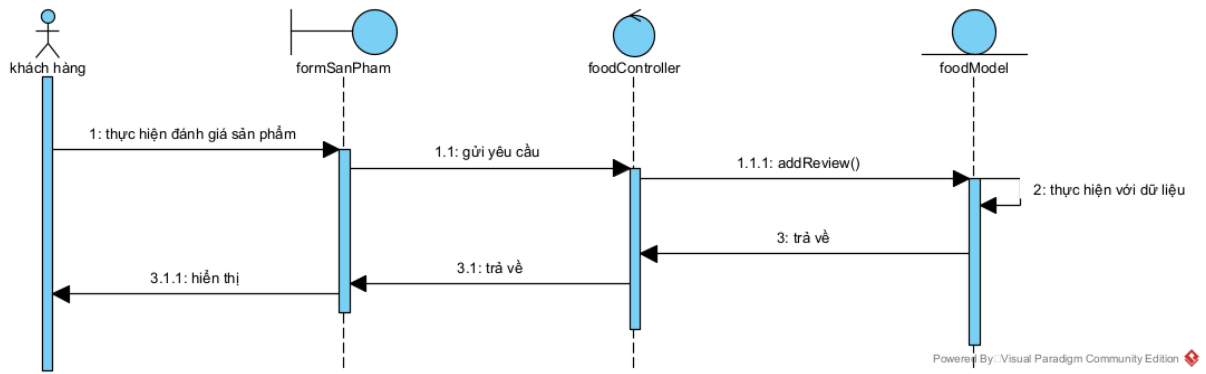


b) xem, tìm kiếm, đánh giá, thêm sản phẩm vào giỏ hàng

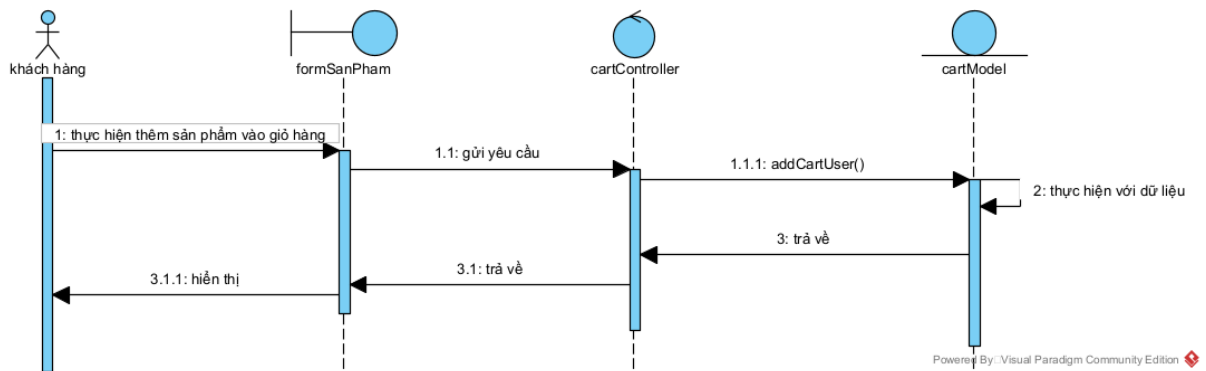
xem



Đánh giá

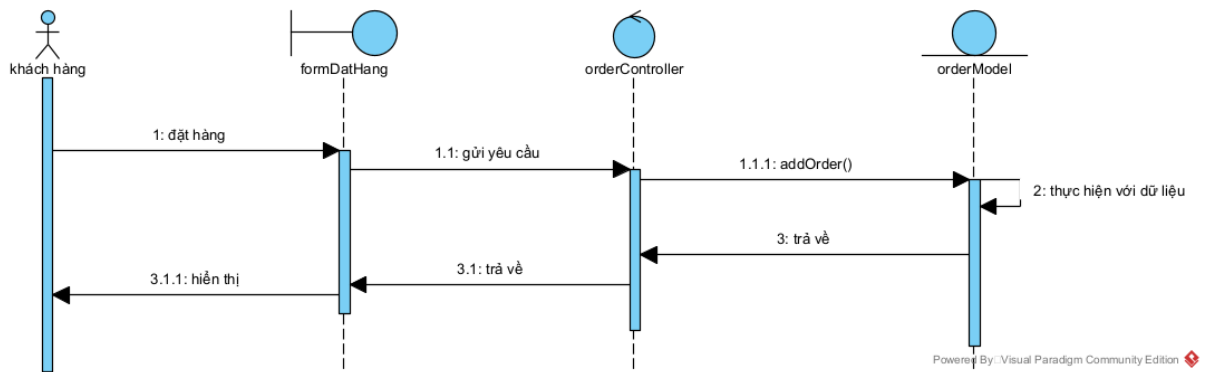


Thêm sản phẩm vào giỏ hàng

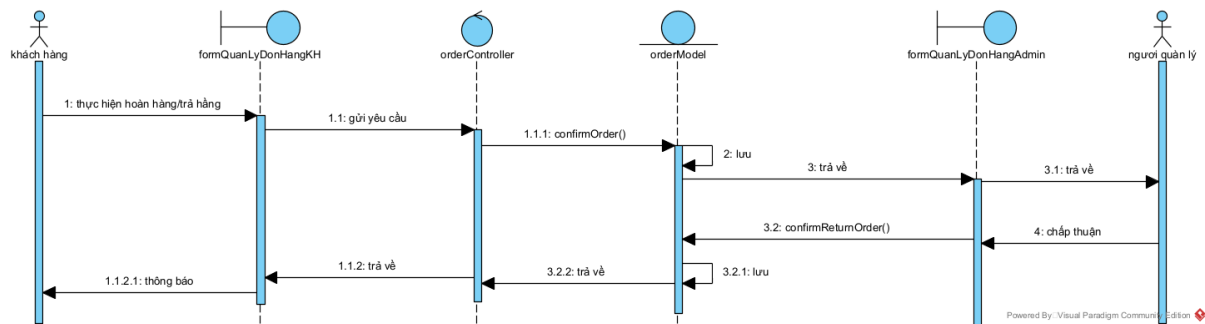


c) đặt hàng, hoàn hàng, trả hàng

đặt hàng



Hoàn hàng/trả hàng



IV. Kế hoạch triển khai:

4.1. Tổng quan về phân chia công việc

Phân tích thiết kế hệ thống: cả nhóm

Xây dựng FE : Minh Đức, Sơn

Xây dựng BE: cả nhóm

Quản lý code: git, Github

4.2. Bản phân công công việc chi tiết

STT	Nội dung công việc	Người thực hiện	Thời gian bắt đầu	Thời gian kết thúc	Trạng thái công việc
1	Xác định về đề tài muốn thực hiện (web hay app, mô hình kinh doanh, mô hình doanh thu, xác định chức năng, công nghệ sử dụng, đối tượng sử dụng, kiến trúc hệ thống)	cả nhóm	7/3/2025	7/3/2025	hoàn thành

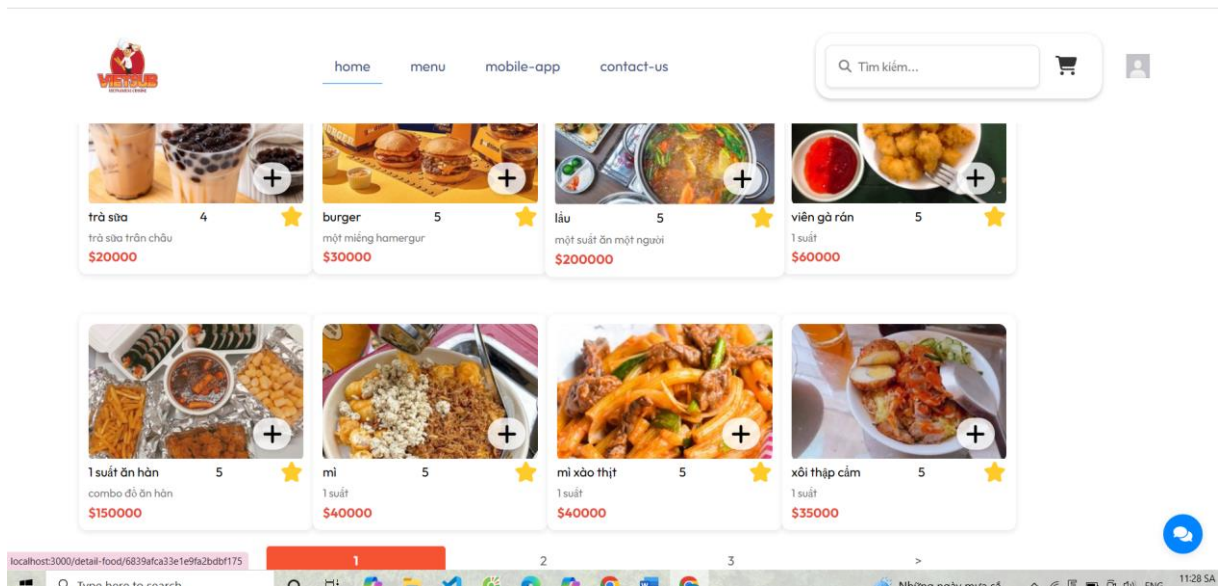
2	Họp phân chia công việc cụ thể	cả nhóm	8/3/2025	8/3/2025	hoàn thành
3	Vẽ Use Case, đặc tả Use Case	Minh Đức	10/3/2025	11/3/2025	hoàn thành
4	Viết Scenario	Minh Đức	12/3/2025	14/3/2025	hoàn thành
5	Vẽ biểu đồ lớp	Minh Đức	14/3/2025	15/3/2025	hoàn thành
6	Vẽ biểu đồ hoạt động	Son	12/3/2025	15/3/2025	hoàn thành
7	Vẽ biểu đồ trình tự	Son	16/3/2025	19/3/2025	hoàn thành
8	Thiết kế cơ sở dữ liệu	Trí Đức	16/3/2025	19/3/2025	hoàn thành
9	Thiết kế giao diện	Trí Đức	20/3/2025	23/3/2025	hoàn thành
10	Họp phân chia xây dựng code	cả nhóm	25/3/2025	25/3/2025	hoàn thành
11	Xây dựng Backend	cả nhóm	1/4/2025	15/5/2025	chưa hoàn thành
11.1	Xây dựng chức năng đăng nhập, đăng ký	Son	1/4/2025	15/5/2025	hoàn thành

11.2	Xây dựng chức năng thêm vào giỏ hàng	Son	1/4/2025	15/5/2025	hoàn thành
11.3	Xây dựng chức năng đặt hàng	Minh Đức	1/4/2025	15/5/2025	hoàn thành
11.4	Xây dựng chức năng tìm kiếm sản phẩm, đơn hàng	Minh Đức	1/4/2025	15/5/2025	hoàn thành
11.5	Xây dựng chức năng chỉnh sửa thông tin cá nhân	Minh Đức	1/4/2025	15/5/2025	hoàn thành
11.6	Xây dựng chức năng phân quyền, phân cấp (admin có thể tạo và chỉnh sửa quyền cho nhân viên)	Trí Đức	1/4/2025	15/5/2025	hoàn thành
11.7	Xây dựng chức năng thêm, xoá, sửa sản phẩm cho admin	Trí Đức	1/4/2025	15/5/2025	hoàn thành
11.8	Xây dựng chức năng quản lý đơn hàng được đặt	Trí Đức	1/4/2025	15/5/2025	hoàn thành

	cho cả người dùng và nhân viên				
11.9	Thống kê doanh thu	Trí Đức	1/4/2025	15/5/2025	hoàn thành
11.10	Khuyến nghị sản phẩm theo logic, và theo admin cài đặt	Trí Đức	1/4/2025	15/5/2025	hoàn thành
11.11	Kết nối với mess	Trí Đức	1/4/2025	15/5/2025	hoàn thành
11.12	Thanh toán QR hoặc thẻ	Trí Đức	1/4/2025	15/5/2025	hoàn thành
12	Xây dựng frontend	Son, Minh Đức	1/4/2025	15/5/2025	hoàn thành
12.1	Xây dựng trang đăng nhập, đăng ký	Son	26/4/2025	15/5/2025	hoàn thành
12.2	Xây dựng trang giao diện chính	Son	26/4/2025	15/5/2025	hoàn thành
12.3	Xây dựng trang giỏ hàng	Son	26/4/2025	15/5/2025	hoàn thành
12.4	Xây dựng trang đặt hàng	Son	26/4/2025	15/5/2025	hoàn thành
12.5	Xây dựng trang quản lý thông tin(thông tin	Son	26/4/2025	15/5/2025	hoàn thành

	cá nhân, thông tin đơn hàng)				
12.6	Xây dựng trang chi tiết đơn hàng	Minh Đức	26/4/2025	15/5/2025	hoàn thành
12.7	Xây dựng trang chi tiết sản phẩm	Minh Đức	26/4/2025	15/5/2025	hoàn thành
12.8	Xây dựng logic sắp xếp, và phân loại	Minh Đức	26/4/2025	15/5/2025	hoàn thành
13	Kết nối FE với BE	Cả nhóm	15/5/2025	17/5/2025	hoàn thành
14	Kiểm thử hệ thống	Cả nhóm	18/5/2025	19/5/2025	hoàn thành

4.3 Giao diện hệ thống



Hình 4.3.1. Giao diện chính của website khách

Delivery Information

First name Last name

Street

City State

Zip code Country

Phone

Items	Title	Price	Quantity	Total
	burger	30000	1	30000
	mi	40000	1	40000

Subtotal

Phí vận chuyển

Tổng giá trị

Chọn phương thức thanh toán

☐ Thanh toán khi nhận hàng (COD)

☒ Thanh toán online

Chọn cổng thanh toán

☐ Stripe

☐ VnPay

☐ MoMo

Bạn đã chọn: Thanh toán online (chưa chọn cổng)

Submit

Hình 4.3.2. Giao diện đặt hàng của khách

My account

Orders

Tất Cả | Chờ Xác Nhận | Đang Chuẩn Bị | Đang Giao Hàng | Đã Giao | Đã Hủy | Đã Hoàn Trả

Returned ORD-227277d6

Items	Title	Price	Quantity	Total
	mi	120000	3	120020

22:29:32 3/6/2025

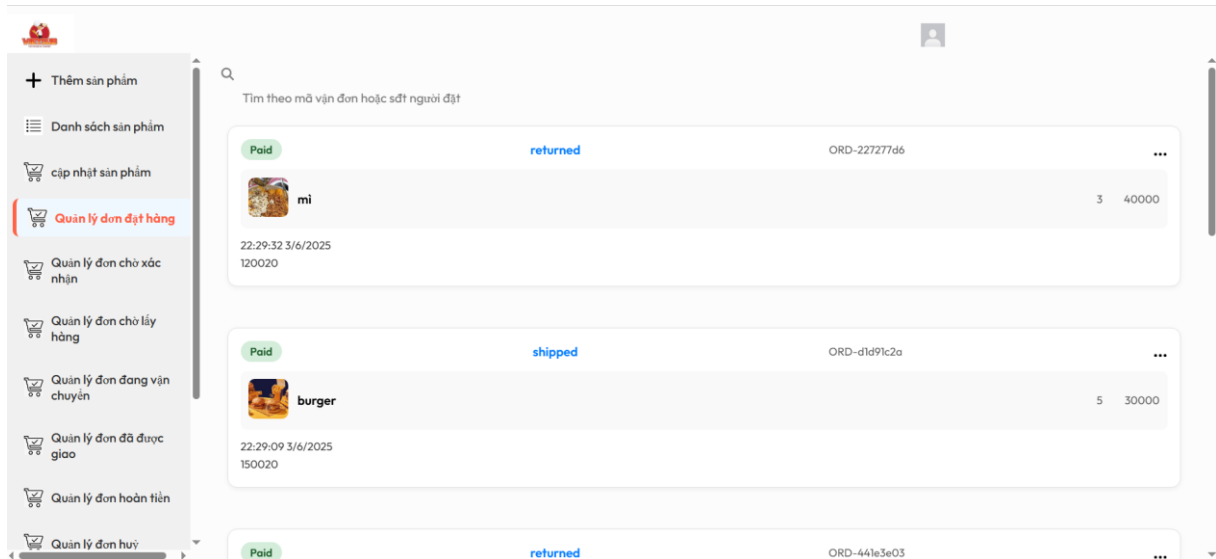
Buy Again

Shipped ORD-d1d91c2a

Items	Title	Price	Quantity	Total
	burger	150000	5	150020

22:29:09 3/6/2025

Hình 4.3.3. Giao diện quản lý đơn tài khoản của khách



Hình 4.3.4. Giao diện và các chức năng của admin

V. Kết luận và định hướng phát triển

Tóm tắt những gì nhóm đã làm được:

- Hoàn thiện hệ thống đặt món ăn trực tuyến gồm frontend (ReactJS) và backend (Node.js, Express).
- Xây dựng chức năng đăng ký, đăng nhập và xác thực người dùng bằng JWT.
- Triển khai các tính năng cho người dùng như: tìm kiếm món ăn, lọc theo loại/giá/đánh giá, quản lý giỏ hàng, đặt hàng, thanh toán (COD và Stripe/VNPay/MoMo), đánh giá sản phẩm.
- Hoàn thành các chức năng quản lý đơn hàng: theo dõi, hủy đơn, đặt lại, lịch sử đơn hàng.
- Tích hợp tính năng đổi trả hàng và hoàn tiền.
- Xây dựng hệ thống chat thời gian thực giữa khách hàng và nhân viên bằng WebSocket (Socket.IO).
- Tạo giao diện quản trị cho admin để quản lý món ăn, đơn hàng, tài khoản nhân viên và khách hàng.
- Thống kê doanh thu, số lượng đơn hàng, xử lý yêu cầu đổi trả và thêm sản phẩm khuyến nghị.

Những khó khăn gặp phải:

- Việc tích hợp nhiều phương thức thanh toán (Stripe, VNPay, MoMo) gây khó khăn trong việc xử lý callback, webhook và đồng bộ trạng thái đơn hàng.
- Quản lý trạng thái đơn hàng và cập nhật real-time đòi hỏi xử lý phức tạp về logic và socket.
- Một số lỗi phát sinh do bất đồng giữa frontend (React) và backend (API Node.js), đặc biệt trong truyền dữ liệu dạng JSON hoặc FormData.
- Thời gian học tập và triển khai giới hạn nên chưa thể tối ưu UI/UX hoặc bảo mật cao cho hệ thống.
- Quản lý phân quyền giữa các vai trò (admin, nhân viên, khách hàng) đôi khi gặp lỗi logic.

Đề xuất hướng phát triển hoặc mở rộng hệ thống:

- Tối ưu giao diện người dùng trên cả desktop và mobile để nâng cao trải nghiệm người dùng.
- Nâng cao bảo mật hệ thống bằng xác thực 2 bước, hạn chế brute-force và lọc dữ liệu đầu vào.
- Tích hợp AI để gợi ý món ăn theo hành vi người dùng (đề xuất theo lịch sử đặt hàng).
- Thêm tính năng mã giảm giá, khuyến mãi và quản lý voucher trong hệ thống.
- Mở rộng báo cáo thống kê cho admin như thống kê theo ngày, tháng, năm, so sánh doanh thu.
- Phân hệ quản lý kho để liên kết món ăn với số lượng nguyên liệu còn lại.
- Tích hợp thêm chatbot AI để hỗ trợ khách hàng 24/7 thay vì chỉ chat với nhân viên.