

TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN
**THIẾT KẾ KIẾN TRÚC
PHẦN MỀM**
NGÀNH CÔNG NGHỆ THÔNG TIN
WEBSITE CÔNG THỨC NẤU ĂN

Sinh viên thực hiện: Hoàng Lê Tuấn Minh_0196566

Nguyễn Mạnh Đức_0196566

Lớp: 66PM4

Giảng viên: Phạm Hữu Tùng

Hà Nội, 5/2024

MỤC LỤC

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI	6
1.1 Lý do chọn đề tài.	6
1.2. Mục tiêu đồ án.	6
1.3. Phạm vi và giới hạn.	7
1.4. Phương pháp thực hiện.	7
2. Cơ sở lý thuyết.	7
2.1 ReactJS	7
2.2. NestJS	9
2.3. Cơ sở dữ liệu MySQL.....	12
2.4. Vite	13
CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.	14
1. Mô tả bài toán.....	14
2. Yêu cầu chức năng.	14
Đối với người dùng chưa đăng nhập:	14
Đối với người dùng đã đăng nhập:	14
Đối với quản trị viên (Admin):	15
3. Yêu cầu phi chức năng.	15
4. Diagram Usecase tổng quát.	16
5. Mô tả các Use Case chính	17
5.1. Usecase: Sign in.....	17
5.2. Usecase: Login.....	17
5.3. Use Case: Add Recipe	18
5.4. Use Case: Modify Recipe.....	19
5.5. Use Case: Delete Recipe.....	20
5.6. Use Case: Review	20
5.7. Use Case: Favorite	21
5.8. Use Case: Add Ingredients	22
5.9. Use Case: Modify Ingredients	23

5.10. Use Case: Delete Ingredients	23
5.11. Use Case: Add Categories	24
5.12. Use Case: Modify Categories	25
5.13. Use Case: Delete Categories	25
5.14. Use Case: Add Users	26
5.15. Use Case: Delete Users	27
5.16. Use Case: Search	27
CHƯƠNG III: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG	28
1. Lựa chọn mô hình kiến trúc	28
2. Sơ đồ kiến trúc tổng thể.	29
3. Biểu đồ hoạt động (<i>Các chức năng chính</i>)	30
3.1. Đăng nhập	30
3.2. Đăng ký.....	30
3.3. Thêm (Công thức, Danh mục, Nguyên liệu).....	31
3.4. Sửa (Công thức, Danh mục, Nguyên liệu)	31
3.5. Xóa (Công thức, Danh mục, Nguyên liệu)	32
3.6. Tìm kiếm.....	32
3.7. Đánh giá	33
3.8. Yêu thích.....	33
4. Thiết kế lớp (Class Diagram)	34
5. Thiết kế cơ sở dữ liệu (ERD, bảng, mối quan hệ) 5.1. Sơ đồ ERD	35
5.2 Database Diagram	36
6. Thiết kế giao diện người dùng (Mockup / Wireframe)	37
6.1. Đăng nhập/ Đăng ký.....	37
6.2. Trang chủ.....	38
6.3. Công thức.....	39
6.4. Hồ sơ người dùng.....	40
6.4. Trang quản lý.....	42
CHƯƠNG IV: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG	44

1. Môi trường triển khai	44
2. Cài đặt hệ thống.....	45
2.1. Yêu cầu hệ thống.....	45
2.2. Các bước cài đặt chương trình.	45
CHƯƠNG V: KẾT QUẢ VÀ ĐÁNH GIÁ	47
1. Kết quả thử nghiệm hệ thống	47
2. Thử nghiệm chức năng toàn hệ thống (Manual Test)	47
3. Đánh giá hiệu quả hệ thống.....	48
Ưu điểm	48
Hạn chế (Cập nhật)	48
Đề xuất cải tiến	48
TÀI LIỆU THAM KHẢO	49

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

1.1 Lý do chọn đề tài.

Trong cuộc sống hiện đại, nhu cầu ăn uống không chỉ dừng lại ở việc “ăn no, mặc ấm” mà còn hướng đến “ăn ngon, sống khỏe”. Việc tự tay chuẩn bị những bữa ăn tại nhà không chỉ giúp tiết kiệm chi phí mà còn đảm bảo vệ sinh an toàn thực phẩm và mang lại giá trị tinh thần to lớn. Tuy nhiên, không phải ai cũng có sẵn kiến thức và kỹ năng nấu ăn, đặc biệt là những người bận rộn hoặc mới bắt đầu làm quen với gian bếp.

Xuất phát từ thực tế đó, đề tài “Website công thức nấu ăn” được xây dựng nhằm tạo ra một nền tảng trực tuyến giúp người dùng dễ dàng tìm kiếm, học hỏi và chia sẻ các công thức nấu ăn phong phú, đa dạng. Website sẽ cung cấp đầy đủ thông tin từ nguyên liệu, cách chế biến, thời gian nấu cho đến mẹo nhỏ trong quá trình nấu nướng. Ngoài ra, người dùng có thể lưu lại công thức yêu thích, đánh giá, bình luận và đóng góp công thức riêng, tạo nên một cộng đồng ẩm thực năng động và bổ ích.

Thông qua đề tài này, nhóm hướng đến việc kết hợp giữa công nghệ và ẩm thực, nhằm mang lại trải nghiệm tiện lợi, thân thiện và truyền cảm hứng cho mọi người trong hành trình khám phá nghệ thuật nấu ăn.

1.2. Mục tiêu đề án.

- Xây dựng một website chia sẻ công thức nấu ăn tiện lợi, dễ sử dụng cho mọi đối tượng người dùng.
- Giúp người dùng dễ dàng tìm kiếm, học hỏi và thực hành các món ăn từ đơn giản đến phức tạp.
- Tạo không gian để người dùng chia sẻ kinh nghiệm nấu nướng, đánh giá và đóng góp công thức cá nhân.
- Ứng dụng công nghệ web hiện đại vào lĩnh vực ẩm thực, nâng cao trải nghiệm người dùng trong việc học nấu ăn trực tuyến.

1.3. Phạm vi và giới hạn.

- Phạm vi: Người dùng có thể xem, tạo, chỉnh sửa, thêm vào yêu thích, bình luận, đánh giá và tìm kiếm công thức nấu ăn. Quản lý có thể quản lý xem, xoá công thức nấu ăn, tài khoản người dùng, danh mục công thức và bình luận
- Giới hạn: Không tích hợp tính năng mua nguyên liệu hoặc video call hướng dẫn trực tiếp. Chỉ hỗ trợ tiếng Việt.

1.4. Phương pháp thực hiện.

- Frontend: ReactJS
- Backend: NestJs
- Cơ sở dữ liệu: MySQL
- Triển khai: Vite

2. Cơ sở lý thuyết.

2.1 ReactJS

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,... Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm

2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

Ngoài việc hỗ trợ xây dựng giao diện nhanh, hạn chế lỗi trong quá trình code, cải thiện performance website thì những tính năng đặc biệt dưới đây có thể là lý do khiến bạn “chốt sale” với ReactJS và bắt đầu tìm hiểu nó từ bây giờ:

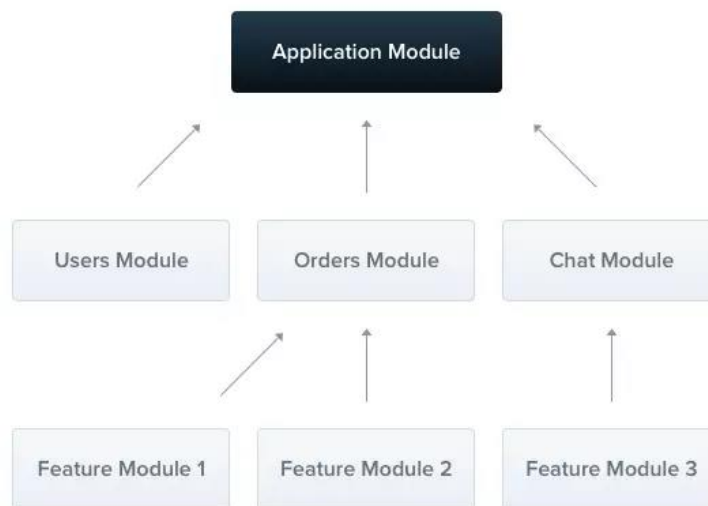
- Phù hợp với đa dạng thể loại website: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì bạn không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML và nó đã cung cấp cho bạn đủ loại “đồ chơi” để bạn có thể dùng cho nhiều trường hợp.
- Tái sử dụng các Component: Nếu bạn xây dựng các Component đủ tốt, đủ flexible để có thể thỏa các “yêu cầu” của nhiều dự án khác nhau, bạn chỉ tốn thời gian xây dựng ban đầu và sử dụng lại hầu như toàn bộ ở các dự án sau. Không chỉ riêng mỗi ReactJS mà các framework hiện nay cũng đều cho phép chúng ta thực hiện điều đó, ví dụ Flutter chẳng hạn.
- Có thể sử dụng cho cả Mobile application: Hầu hết chúng ta đều biết rằng ReactJS được sử dụng cho việc lập trình website, nhưng thực chất nó được sinh ra không chỉ làm mỗi điều đó. Nếu bạn cần phát triển thêm ứng dụng Mobile, thì hãy sử dụng thêm React Native – một framework khác được phát triển cũng chính Facebook, bạn có thể dễ dàng “chia sẻ” các Component hoặc sử dụng lại các Business Logic trong ứng dụng.
- Thân thiện với SEO: SEO là một phần không thể thiếu để đưa thông tin website của bạn lên top đầu tìm kiếm của Google. Bản chất ReactJS là một thư viện JavaScript, Google Search Engine hiện nay đã crawl và index được code JavaScript, tuy nhiên bạn cũng cần thêm một vài thư viện khác để hỗ trợ điều này nhé!
- Debug dễ dàng: Facebook đã phát hành 1 Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng. Điều đó giúp tăng tốc quá trình release sản phẩm cũng như quá trình coding của bạn.

2.2. NestJS

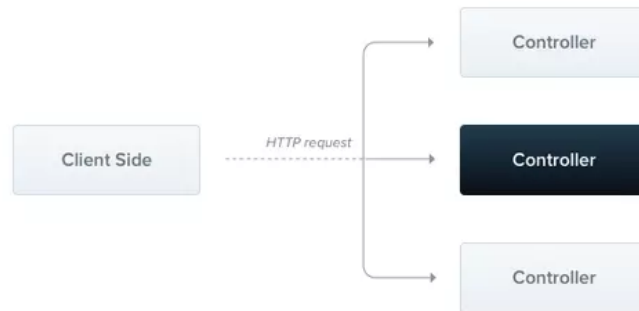
NestJS là một framework mã nguồn mở để phát triển ứng dụng server-side (backend applications) bằng ngôn ngữ TypeScript hoặc JavaScript. Nó được xây dựng trên cơ sở của Node.js và sử dụng các khái niệm từ TypeScript để tạo ra một môi trường phát triển hiện đại và mạnh mẽ cho việc xây dựng các ứng dụng web và API.

Mục tiêu chính của NestJS là cung cấp một cấu trúc ứng dụng rõ ràng và dễ quản lý, giúp tăng tính bảo trì và sự tổ chức trong mã nguồn. Để đạt được điều này, NestJS triển khai mô hình kiến trúc lõi (core architecture) dựa trên các nguyên tắc của Angular, đặc biệt là sử dụng Dependency Injection (DI) và Modules (Các module).

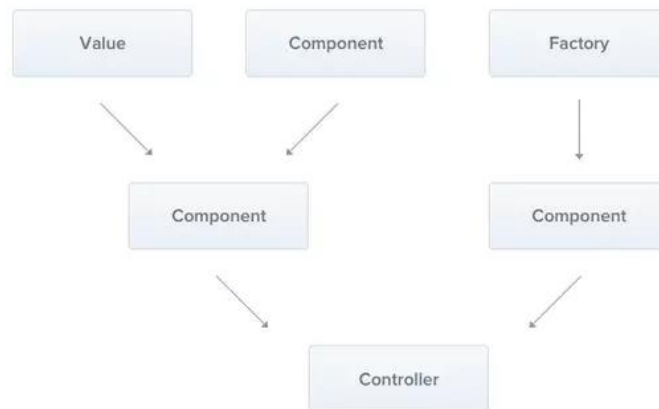
Cấu trúc của NestJS được xây dựng dựa trên mô hình kiến trúc lõi (core architecture) giúp tạo ra một ứng dụng server-side (backend application) rõ ràng, dễ quản lý và dễ mở rộng. Cấu trúc NestJS thường được tổ chức thành các phần chính sau:



Module (Các module): Module là một phần cơ bản trong cấu trúc NestJS. Mỗi ứng dụng NestJS bao gồm ít nhất một module gốc (root module) và có thể có nhiều module con. Module là nơi tổ chức các thành phần của ứng dụng như Controllers, Providers và các thành phần khác. Mỗi module đại diện cho một phần chức năng cụ thể của ứng dụng.



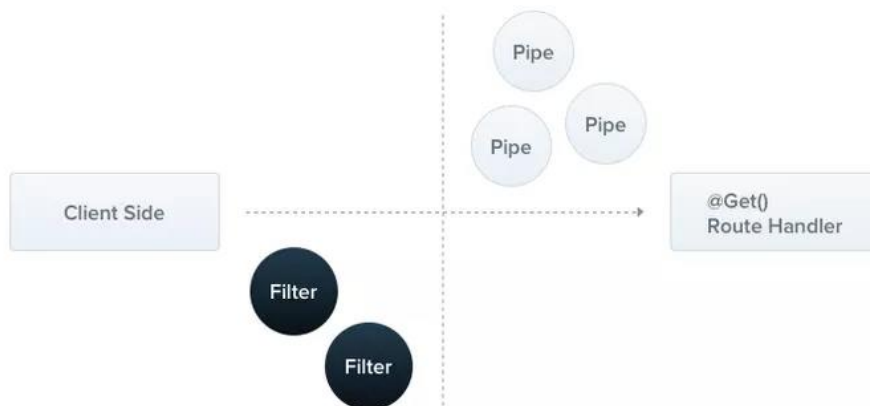
Controller (Bộ điều khiển): Controllers là thành phần chịu trách nhiệm xử lý các yêu cầu HTTP từ phía client và trả về kết quả tương ứng. Controllers là nơi xử lý các request và trả về các response. Các phương thức của controller được chú thích (decorated) bằng các decorator như `@Get()`, `@Post()`, `@Put()`, v.v., để chỉ định các route và phương thức HTTP tương ứng.



Provider (Các nhà cung cấp): Providers là thành phần chịu trách nhiệm cung cấp các dịch vụ cho ứng dụng. Đây có thể là các service, repository, logger, v.v. Providers sử dụng dependency injection để chèn vào các thành phần khác và có thể được sử dụng bởi các controllers hoặc các providers khác.



Middleware (Trung gian): Middleware là các hàm xử lý mà NestJS sử dụng để xử lý các yêu cầu HTTP trước khi chúng đến các route xử lý chính. Middleware có thể được sử dụng để thực hiện các thao tác chung như xác thực, ghi log, xử lý lỗi, v.v.



Filter (Bộ lọc): Filters được sử dụng để xử lý các exception (ngoại lệ) xảy ra trong ứng dụng. Filters cho phép bạn xử lý và thay đổi response trước khi gửi về client khi có exception xảy ra.

Guard (Bảo vệ): Guards được sử dụng để kiểm tra xem một yêu cầu có thể được xử lý hoặc không. Guards cho phép bạn thực hiện các kiểm tra xác thực hoặc kiểm tra quyền trước khi xử lý một yêu cầu.

Interceptor (Bộ chặn): Interceptors là các hàm xử lý mà NestJS sử dụng để chặn và thay đổi response trước khi nó được gửi về client. Interceptors có thể được sử dụng để thực hiện các thao tác chung trên response trước khi nó đi ra ngoài.

Exception (Ngoại lệ): Exception handling (xử lý ngoại lệ) là một phần quan trọng của cấu trúc NestJS. Exception handling cho phép bạn xử lý các exception xảy ra trong ứng dụng và trả về các thông báo lỗi thích hợp cho client.

2.3. Cơ sở dữ liệu MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về **MySQL** miễn phí từ trang chủ. **MySQL** có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng **Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...**



Đặc điểm của MySQL:

- Hiệu suất cao và khả năng mở rộng: MySQL được tối ưu hóa để xử lý một lượng lớn dữ liệu một cách nhanh chóng và hiệu quả. Hệ thống có thể dễ dàng mở rộng để đáp ứng nhu cầu lưu trữ và xử lý dữ liệu ngày càng tăng.
- Bảo mật và tính toàn vẹn dữ liệu: MySQL cung cấp các tính năng bảo mật mạnh mẽ như xác thực người dùng, phân quyền, mã hóa dữ liệu và phát hiện xâm nhập, giúp bảo vệ dữ liệu khỏi các mối đe dọa an ninh.
- Tính linh hoạt và tùy biến cao: MySQL hỗ trợ nhiều nền tảng, nhiều loại engine lưu trữ và cho phép người dùng tùy chỉnh cấu hình để phù hợp với nhu cầu cụ thể của ứng dụng.

- Cộng đồng lớn và hỗ trợ mạnh mẽ: MySQL có một cộng đồng người dùng rất lớn, sẵn sàng chia sẻ kinh nghiệm và giải đáp thắc mắc.
- Mã nguồn mở: MySQL là một phần mềm mã nguồn mở, người dùng có thể tự do sử dụng, sửa đổi và phân phối.
- Cơ chế quản lý dữ liệu: MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu (database). Mỗi cơ sở dữ liệu có thể chứa nhiều bảng (table), mỗi bảng lưu trữ một loại dữ liệu cụ thể. Các bảng được liên kết với nhau thông qua các khóa (key) để tạo thành các mối quan hệ.
- Cơ chế phân quyền: MySQL có cơ chế phân quyền người dùng rất chi tiết. Mỗi người dùng có thể được cấp quyền truy cập đến một hoặc nhiều cơ sở dữ liệu, với các quyền hạn khác nhau như đọc, ghi, sửa đổi cấu trúc bảng, v.v.
- Ngôn ngữ SQL: MySQL sử dụng ngôn ngữ SQL (Structured Query Language) để tương tác với dữ liệu. SQL là ngôn ngữ chuẩn để truy vấn, cập nhật, xóa và quản lý dữ liệu trong các hệ quản trị cơ sở dữ liệu quan hệ.

2.4. Vite

Vite (phát âm là "vit", nghĩa là "nhanh" trong tiếng Pháp) hay **Vitejs** là một công cụ build và phát triển ứng dụng web được tạo ra bởi Evan You, người đã phát triển Vue.js, Vite ra đời mục đích khắc phục những hạn chế về tốc độ và hiệu suất.

Vite mang đến cách tiếp cận mới trong việc xây dựng các dự án web bằng cách tận dụng các module ES gốc trong trình duyệt và cung cấp tính năng **Hot Module Replacement (HMR)**.

Giả sử bạn đang xây dựng một ứng dụng web đơn giản với **React**. Trước khi có **Vite**, công cụ phổ biến như **Webpack** hoặc **Create React App (CRA)** được sử dụng nhiều. Tuy nhiên, các công cụ này có thể chậm khi khởi động hoặc cập nhật code.

Đặc điểm và chức năng của Vite:

- **Phân phát tệp theo yêu cầu qua ESM gốc:** thay vì đóng gói toàn bộ mã nguồn thành một khối lớn, Vite tận dụng các modules ECMAScript (ESM) gốc của browser để tải từng tệp riêng lẻ khi cần. Cách tiếp cận này giúp loại bỏ quá trình đóng gói phức tạp, tăng tốc và tối ưu hiệu suất.
- **API plugin và API JavaScript của Vite:** bạn có thể thêm các tính năng mới, thay đổi cách Vite hoạt động, hoặc tích hợp với các công cụ khác để phù hợp với yêu cầu của dự án từ đơn giản đến phức tạp.

CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.

1. Mô tả bài toán.

Hiện nay, với sự phát triển của công nghệ và nhu cầu nấu ăn tại nhà ngày càng tăng, người dùng mong muốn có một nền tảng nơi họ có thể tìm kiếm, lưu trữ, chia sẻ và học hỏi các công thức nấu ăn một cách dễ dàng. Tuy nhiên, các nền tảng hiện tại đa số chỉ mang tính chất chia sẻ chung chung, thiếu tổ chức, khó tìm kiếm và không thân thiện với người dùng. Bài toán đặt ra là xây dựng một website công thức nấu ăn có giao diện hiện đại, trực quan, cho phép người dùng:

- Xem công thức theo danh mục, khẩu vị, thời gian nấu, thành phần, v.v.
- Tạo, chỉnh sửa và chia sẻ công thức cá nhân.
- Đánh giá, bình luận công thức của người khác.
- Lưu trữ công thức yêu thích để tham khảo sau.

2. Yêu cầu chức năng.

Hệ thống cần đáp ứng các chức năng chính sau:

Đối với người dùng chưa đăng nhập:

- Xem danh sách công thức nấu ăn.
- Tìm kiếm công thức theo tên hoặc danh mục.
- Xem chi tiết công thức: nguyên liệu, cách làm, hình ảnh minh họa, video (nếu có).
- Đăng ký tài khoản.
- Đăng nhập.

Đối với người dùng đã đăng nhập:

- Tạo công thức mới (gồm tên món, mô tả, nguyên liệu, bước làm, hình ảnh, video).
- Chỉnh sửa hoặc xóa công thức cá nhân.
- Đánh giá công thức (1 – 5 sao).

- Bình luận và xem bình luận.
- Thêm công thức vào danh sách yêu thích.
- Quản lý hồ sơ cá nhân.

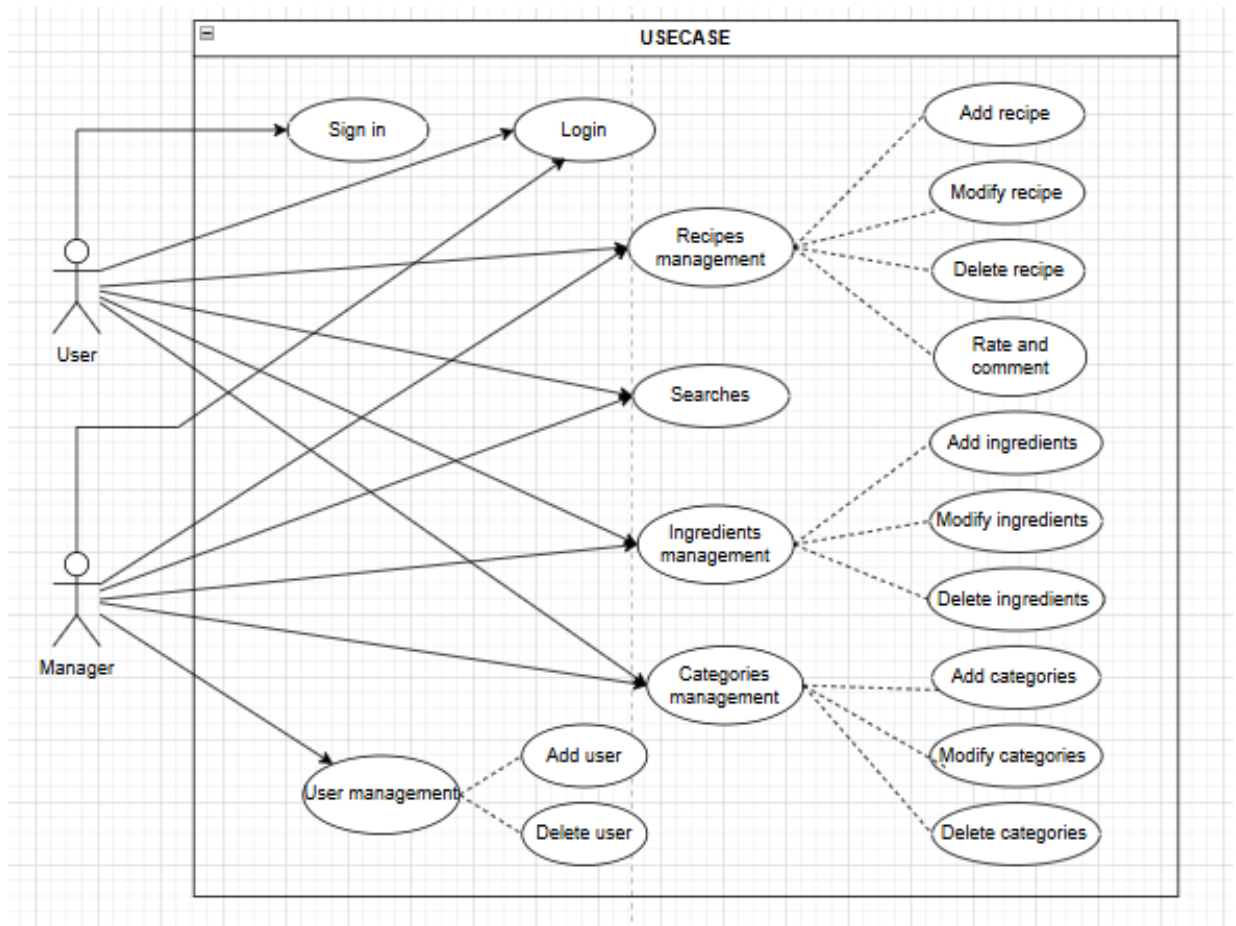
Đối với quản trị viên (Admin):

- Quản lý người dùng: danh sách, xóa người dùng vi phạm.
- Quản lý công thức: kiểm duyệt nội dung vi phạm.
- Quản lý danh mục món ăn.
- Quản lý bình luận: kiểm duyệt nội dung vi phạm.

3. Yêu cầu phi chức năng.

Loại yêu cầu	Mô tả
Hiệu năng	Hệ thống phản hồi dưới 2 giây với các thao tác chính.
Tính mở rộng	Dễ dàng mở rộng thêm chức năng như chia sẻ mạng xã hội, đa ngôn ngữ.
Tính sẵn sàng	Hệ thống hoạt động liên tục, hạn chế thời gian downtime.
Tính thân thiện	Giao diện trực quan, hỗ trợ cả máy tính và thiết bị di động.
Bảo mật	Mã hóa mật khẩu, xác thực JWT, phân quyền người dùng rõ ràng.

4. Diagram Usecase tổng quát.



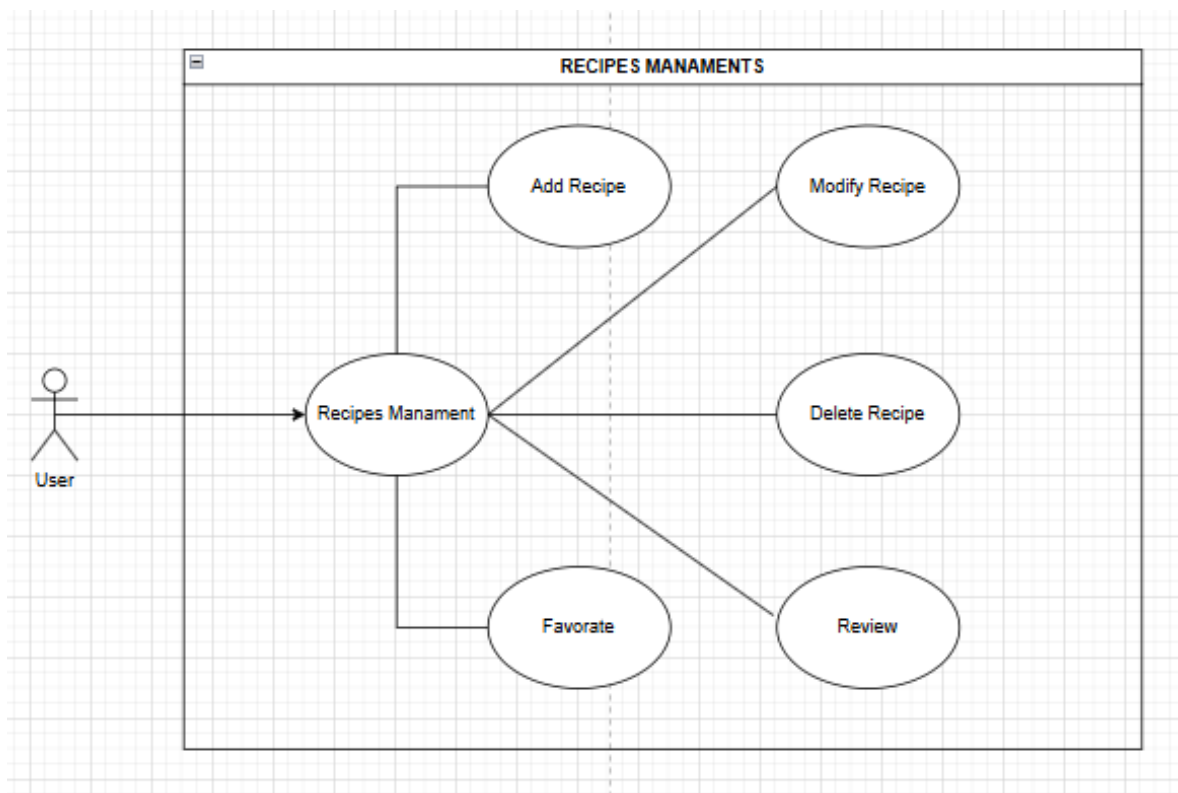
5. Mô tả các Use Case chính

5.1. Usecase: Sign in.

ID:	001
Tác nhân:	User
Mô tả :	Người dùng tạo tài khoản để sử dụng hệ thống.
Tiền điều kiện:	Người dùng chưa có tài khoản.
Hậu điều kiện:	Người dùng được đăng nhập vào hệ thống.
Luồng chính:	<ol style="list-style-type: none">1. Người dùng truy cập trang đăng nhập.2. Nhập email và mật khẩu.3. Hệ thống xác thực thông tin.4. Nếu đúng, truy cập thành công; nếu sai, hiển thị lỗi.

5.2. Usecase: Login.

ID:	002
Tác nhân:	User, Manager
Mô tả :	Người dùng đăng nhập để sử dụng chức năng hệ thống.
Tiền điều kiện:	Người dùng đã có tài khoản.
Hậu điều kiện:	Người dùng được đăng nhập vào hệ thống.
Luồng chính:	<ol style="list-style-type: none">1. Người dùng truy cập trang đăng nhập.2. Nhập email và mật khẩu.3. Hệ thống xác thực thông tin.4. Nếu đúng, truy cập thành công; nếu sai, hiển thị lỗi.



5.3. Use Case: Add Recipe

ID:	003
Tác nhân:	User, Manager
Mô tả :	Cho phép người dùng thêm một công thức nấu ăn mới vào hệ thống.
Tiền điều kiện:	Người dùng đã đăng nhập.
Hậu điều kiện:	Công thức mới được lưu và hiển thị trong danh sách công thức.
Luồng chính:	<ol style="list-style-type: none"> 1. Người dùng chọn chức năng "Thêm công thức". 2. Hệ thống hiển thị form nhập liệu. 3. Người dùng nhập các thông tin: tên món ăn, nguyên liệu, mô tả cách làm, ảnh minh họa, danh mục món ăn. 4. Người dùng nhấn nút "Lưu".

	<p>5. Hệ thống xác thực dữ liệu.</p> <p>6. Hệ thống lưu công thức và hiển thị thông báo thành công.</p>
Luồng phụ:	Nếu ảnh không hợp lệ → Hệ thống thông báo lỗi ảnh, yêu cầu chọn lại. Nếu thiếu trường bắt buộc → Hệ thống thông báo lỗi và yêu cầu nhập lại.

5.4. Use Case: Modify Recipe

ID:	004
Tác nhân:	User, Manager
Mô tả :	Cho phép sửa thông tin công thức đã đăng.
Tiền điều kiện:	Công thức phải thuộc quyền sở hữu của người dùng hoặc quyền admin.
Hậu điều kiện:	Công thức được cập nhật mới.
Luồng chính:	<p>7. Người dùng chọn công thức đã đăng.</p> <p>8. Nhấn nút “Chỉnh sửa”.</p> <p>9. Hệ thống hiển thị form chỉnh sửa.</p> <p>10. Người dùng cập nhật thông tin.</p> <p>11. Nhấn "Lưu".</p> <p>12. Hệ thống xác thực và cập nhật dữ liệu.</p>
Luồng phụ:	Nếu không có quyền chỉnh sửa → Hệ thống hiển thị lỗi "Bạn không có quyền sửa công thức này".

5.5. Use Case: Delete Recipe

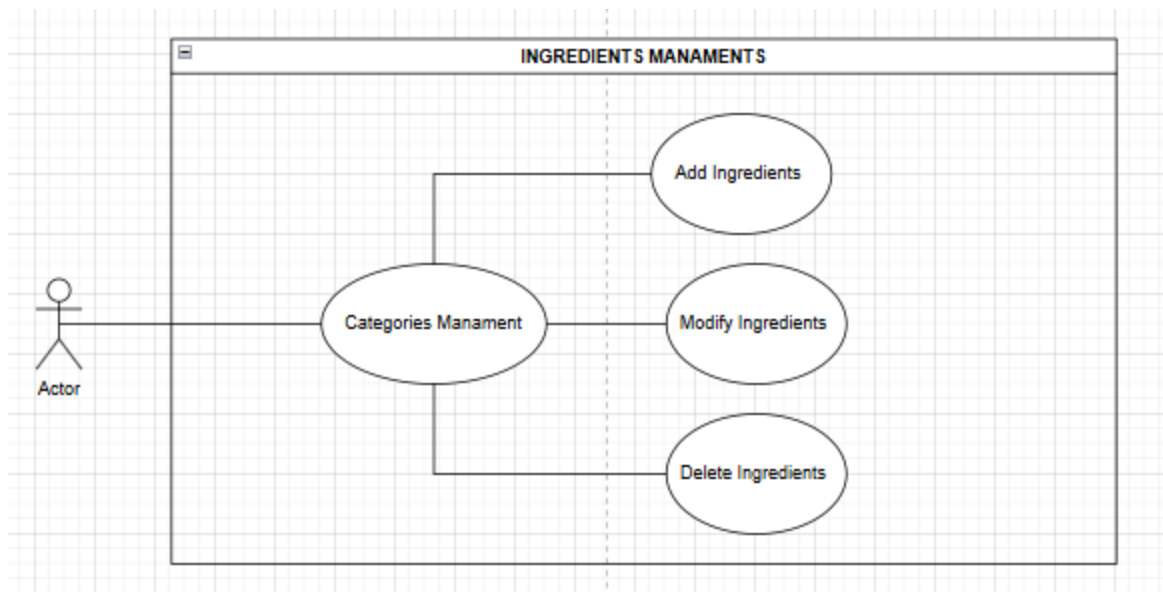
ID:	005
Tác nhân:	User, Manager
Mô tả :	Xóa công thức khỏi hệ thống.
Tiền điều kiện:	Công thức phải thuộc quyền sở hữu của người dùng hoặc quyền admin.
Hậu điều kiện:	Công thức bị xoá vĩnh viễn khỏi hệ thống.
Luồng chính:	<ol style="list-style-type: none">1. Người dùng nhấn nút “Xóa” tại công thức.2. Hệ thống hiển thị cảnh báo xác nhận.3. Người dùng xác nhận xóa.4. Hệ thống xóa công thức khỏi cơ sở dữ liệu.

5.6. Use Case: Review

ID:	006
Tác nhân:	User
Mô tả :	Người dùng có thể đánh giá và bình luận công thức.
Tiền điều kiện:	Người dùng đã đăng nhập.
Hậu điều kiện:	Bình luận và đánh giá hiển thị công khai.
Luồng chính:	<ol style="list-style-type: none">5. Người dùng mở một công thức.6. Nhập nội dung bình luận và chọn số sao đánh giá.7. Nhấn “Gửi”.8. Hệ thống lưu và hiển thị bình luận & đánh giá.

5.7. Use Case: Favorite

ID:	006
Tác nhân:	User
Mô tả :	Người dùng có thể thêm công thức vào mục yêu thích.
Tiền điều kiện:	Người dùng đã đăng nhập.
Hậu điều kiện:	Công thức được thêm vào mục yêu thích của người dùng.
Luồng chính:	<ol style="list-style-type: none">1. Người dùng mở một công thức.2. Nhấp nút thêm vào yêu thích.3. Hệ thống lưu công thức vào mục yêu thích.



5.8. Use Case: Add Ingredients

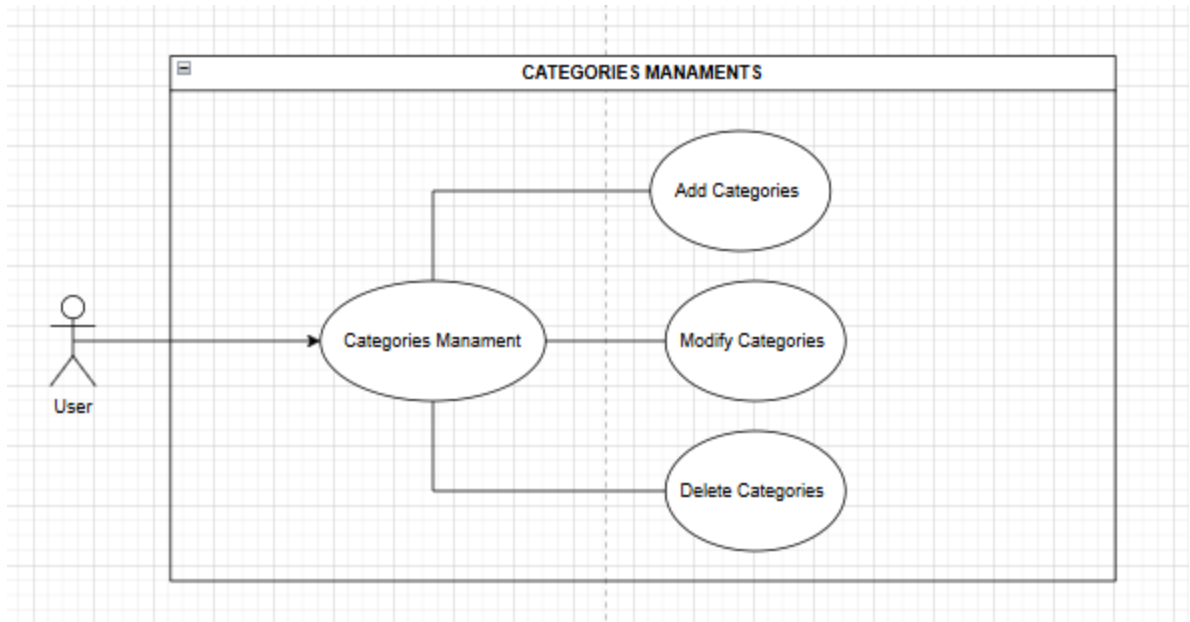
ID:	007
Tác nhân:	User, Manager
Mô tả :	Cho phép người dùng thêm nguyên liệu mới.
Tiền điều kiện:	Người dùng đã đăng nhập.
Hậu điều kiện:	Nguyên liệu được thêm mới
Luồng chính:	<ol style="list-style-type: none"> 1. Người dùng chọn mục “Thêm nguyên liệu”. 2. Nhập tên nguyên liệu, đơn vị tính,... 3. Nhấn “Lưu”. 4. Hệ thống lưu nguyên liệu vào cơ sở dữ liệu.

5.9. Use Case: Modify Ingredients

ID:	008
Tác nhân:	User, Manager
Mô tả :	Cho phép người dùng cập nhật nguyên liệu có sẵn.
Tiền điều kiện:	Nguyên liệu đã tồn tại và thuộc quyền chỉnh sửa.
Hậu điều kiện:	Cập nhật thành công nguyên liệu
Luồng chính:	<ol style="list-style-type: none">1. Người dùng chọn nguyên liệu.2. Nhấn “Chỉnh sửa”.3. Cập nhật thông tin nguyên liệu.4. Nhấn “Lưu”.

5.10. Use Case: Delete Ingredients

ID:	009
Tác nhân:	User, Manager
Mô tả :	Xoá nguyên liệu không còn dùng
Tiền điều kiện:	Người dùng đã đăng nhập
Hậu điều kiện:	Nguyên liệu bị xoá khỏi hệ thống
Luồng chính:	<ol style="list-style-type: none">1. Người dùng chọn nguyên liệu cần xoá.2. Nhấn “Xoá” và xác nhận.3. Hệ thống xoá nguyên liệu khỏi hệ thống.



5.11. Use Case: Add Categories

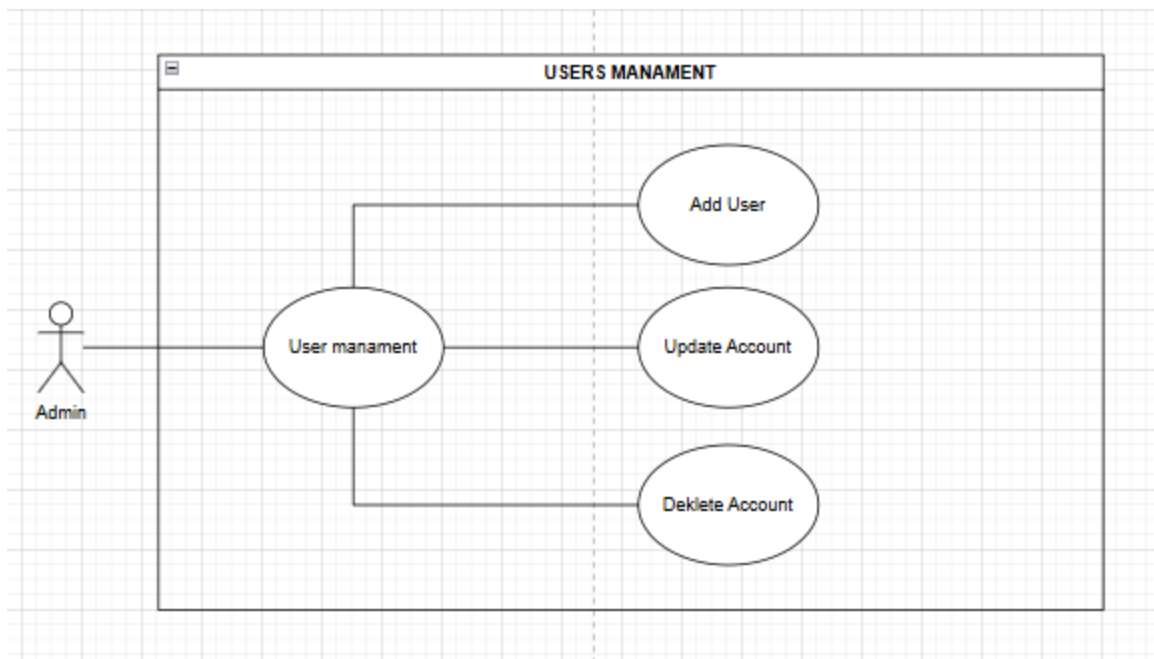
ID:	010
Tác nhân:	User, Manager
Mô tả :	Thêm một danh mục mới cho công thức (ví dụ: món chay, món mặn...).
Tiền điều kiện:	Người dùng đã đăng nhập
Hậu điều kiện:	Danh mục mới được thêm vào hệ thống, có thể sử dụng khi đăng công thức.
Luồng chính:	<ol style="list-style-type: none"> 1. Người dùng chọn chức năng “Thêm danh mục”. 2. Nhập tên danh mục, mô tả (nếu có). 3. Nhấn “Lưu”. 4. Hệ thống xác thực và lưu thông tin danh mục.

5.12. Use Case: Modify Categories

ID:	011
Tác nhân:	User, Manager
Mô tả :	Sửa thông tin danh mục đã tồn tại.
Tiền điều kiện:	Danh mục đã tồn tại và người dùng có quyền chỉnh sửa.
Hậu điều kiện:	Danh mục được cập nhật trong hệ thống.
Luồng chính:	<ol style="list-style-type: none">1. Người dùng chọn danh mục cần sửa.2. Nhấn nút “Chỉnh sửa”.3. Cập nhật thông tin (tên, mô tả...).4. Nhấn “Lưu”.5. Hệ thống xác thực và cập nhật danh mục.

5.13. Use Case: Delete Categories

ID:	012
Tác nhân:	User, Manager
Mô tả :	Xoá danh mục món ăn không còn phù hợp.
Tiền điều kiện:	Danh mục không được liên kết với bất kỳ công thức nào.
Hậu điều kiện:	Danh mục bị xóa khỏi hệ thống..
Luồng chính:	<ol style="list-style-type: none">1. Người dùng chọn danh mục cần xoá.2. Nhấn nút “Xoá”.3. Hệ thống kiểm tra ràng buộc.4. Nếu hợp lệ, danh mục được xóa vĩnh viễn.



5.14. Use Case: Add Users

ID:	013
Tác nhân:	Manager
Mô tả :	Quản trị viên tạo tài khoản cho người dùng khác.
Tiền điều kiện:	Quản trị viên đã đăng nhập.
Hậu điều kiện:	Người dùng mới có thể đăng nhập và sử dụng hệ thống.
Luồng chính:	<ol style="list-style-type: none"> 1. Manager chọn chức năng “Thêm người dùng”. 2. Nhập email, tên, vai trò và mật khẩu. 3. Nhấn “Lưu”. 4. Hệ thống tạo tài khoản mới.

5.15. Use Case: Delete Users

ID:	014
Tác nhân:	Manager
Mô tả :	Xóa tài khoản người dùng khỏi hệ thống (khi vi phạm hoặc không còn sử dụng).
Tiền điều kiện:	Manager đã đăng nhập và có quyền truy cập chức năng.
Hậu điều kiện:	Người dùng bị xóa vĩnh viễn, không thể đăng nhập lại.
Luồng chính:	<ol style="list-style-type: none">1. Manager truy cập danh sách người dùng.2. Chọn người dùng cần xóa.3. Nhấn “Xóa” và xác nhận.4. Hệ thống xóa người dùng và thông báo thành công.

5.16. Use Case: Search

ID:	015
Tác nhân:	User, Manager
Mô tả :	Cho phép người dùng tìm kiếm công thức nấu ăn theo từ khóa, nguyên liệu, danh mục, hoặc theo mức đánh giá.
Tiền điều kiện:	Người dùng đã truy cập hệ thống (không cần đăng nhập vẫn tìm kiếm được).
Hậu điều kiện:	Danh sách công thức phù hợp với tiêu chí tìm kiếm được hiển thị.
Luồng chính:	<ol style="list-style-type: none">1. Người dùng nhập từ khóa vào ô tìm kiếm (có thể là tên món ăn, tên nguyên liệu, hoặc tên danh mục).2. Người dùng nhấn nút “Tìm kiếm”.3. Hệ thống truy vấn dữ liệu dựa trên tiêu chí đã nhập và hiển thị

CHƯƠNG III: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

1. Lựa chọn mô hình kiến trúc

Sử dụng **kiến trúc phân lớp (Layered Architecture)**. Là dạng kiến trúc rất phổ biến ngày nay, chắc chắn ai cũng đã từng nghe qua về kiến trúc 3 lớp, MVC, MVVM,... Tất cả đều có chung một mục đích là tổ chức, phân rã mã nguồn theo nhiều chức năng và nhiệm vụ khác nhau trong hệ thống.

Trong các hệ thống hướng đối tượng, UI, Database và các dạng mã khác thường được viết trực tiếp vào trong logic nghiệp vụ. Và theo chiều ngược lại, mã nghiệp vụ lâu lâu lại được nhúng vào trong UI, DB code. Không có sự phân cấp rõ ràng dẫn đến sự phức tạp trong tổ chức, quá nhiều phụ thuộc lẫn nhau giữa các mã nguồn. Sở dĩ điều này diễn ra là vì lập trình nó rất dễ và nhanh.. nhưng chỉ là lúc mới bắt đầu. Khi các đoạn mã liên quan đến nghiệp vụ được dàn trải ở khắp nơi, điều này sẽ trở nên rất phức tạp và rất khó để đọc hiểu được nghiệp vụ. Thay đổi trên giao diện người dùng cũng có thể làm thay đổi business logic. Để thay đổi business logic, chúng ta phải đi ngược lên các mã UI, cơ sở dữ liệu hoặc các phần tử của chương trình khác để xem những chỗ cần phải thay đổi theo. Điều này không cần nói cũng đủ hiểu nó gây ra khó khăn như thế nào.

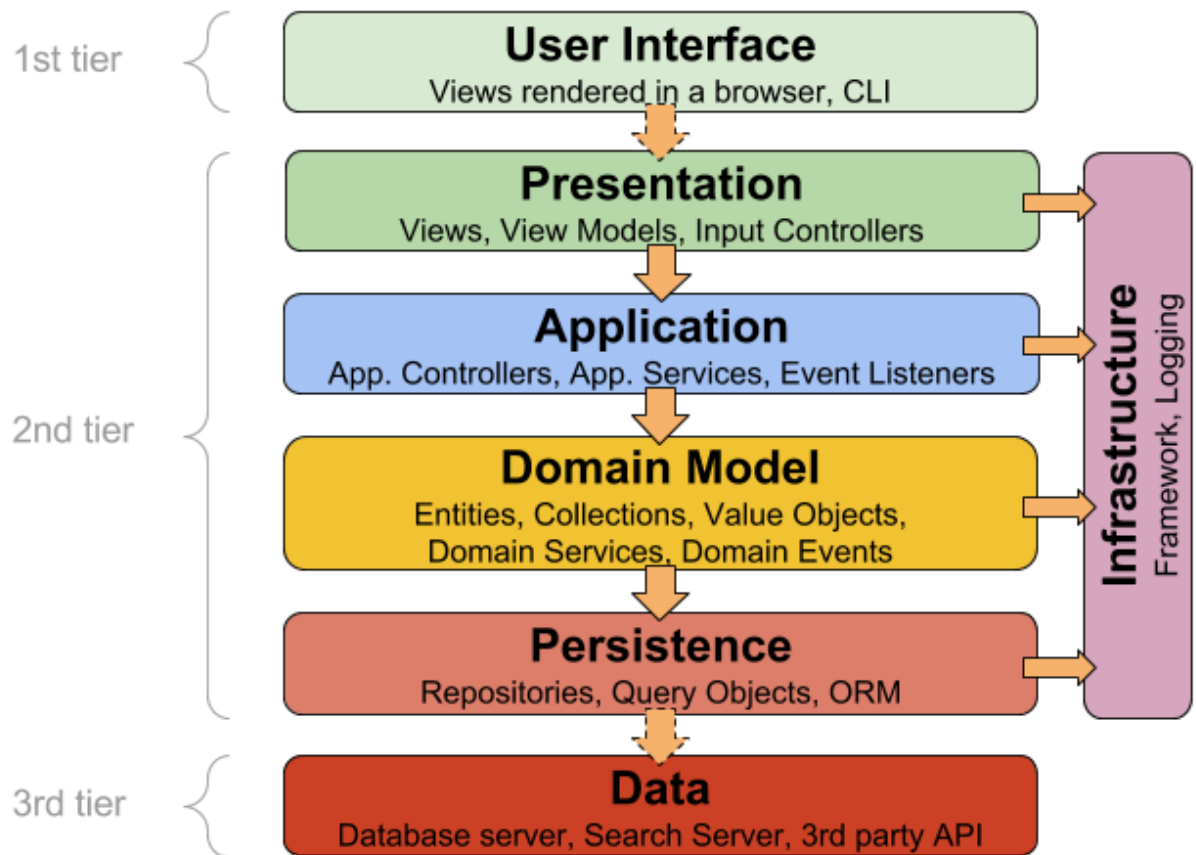
ƯU ĐIỂM:

- Chúng ta chỉ cần hiểu những lớp bên dưới lớp chúng ta đang làm;
- Mỗi lớp có thể được thay thế bởi một thể hiện tương đương (equivalent implementation), không ảnh hưởng đến các lớp khác;
- Một lớp có thể được sử dụng bởi một số lớp cấp cao khác nhau.

NHỮNG BẤT LỢI LÀ:

- Lớp không thể đóng gói (encapsulate) tất cả mọi thứ (một field được thêm vào UI, rất có thể cũng cần phải được thêm vào DB);
- Các lớp bổ sung (extra) có thể gây ảnh hưởng đến hiệu suất, đặc biệt nếu ở các tier khác nhau.

2. Sơ đồ kiến trúc tổng thể.



User Interface (Presentation): là những chương trình chạy trên desktop / CLI hoặc các web page. Các chương trình client lúc này thường là các rich client (nơi luồng công việc và kiểm tra input đầu vào đặt ở client)

Business Logic (Domain): là nơi đặt các logic về nghiệp vụ chính của hệ thống, layer này sẽ tiếp nhận các request từ phía Client, xử lý và lưu trữ data thông qua Data source layer.

Data source: nhiệm vụ chính của layer này là thực thi các tác vụ lưu trữ dữ liệu và liên lạc với các applications khác (được request từ business logic layer).

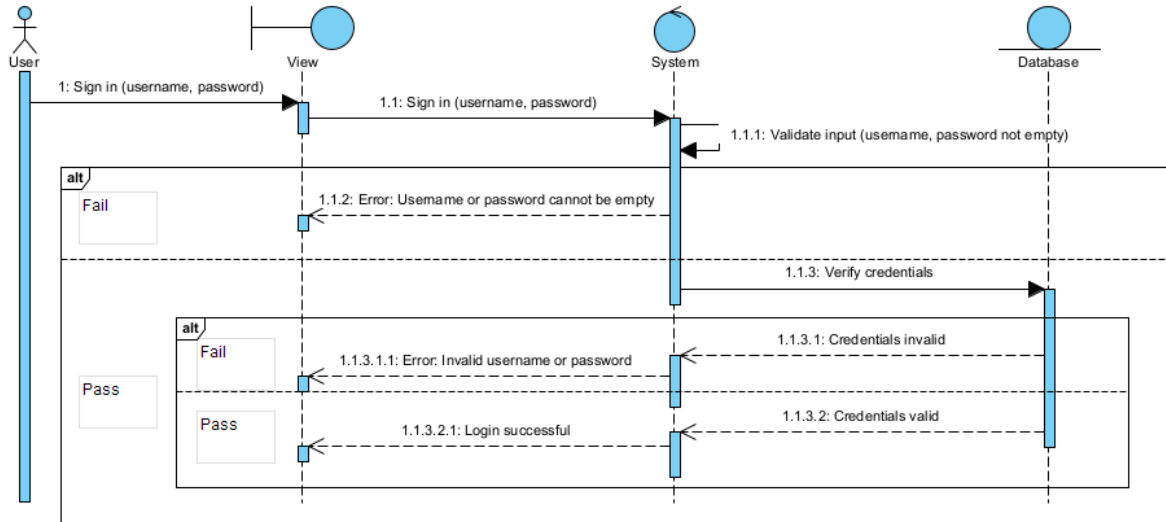
Web browser: làm nhiệm vụ render và gửi / nhận request tới server.

Application server: chứa presentation layer, the application layer, the domain layer, và the persistence layer.

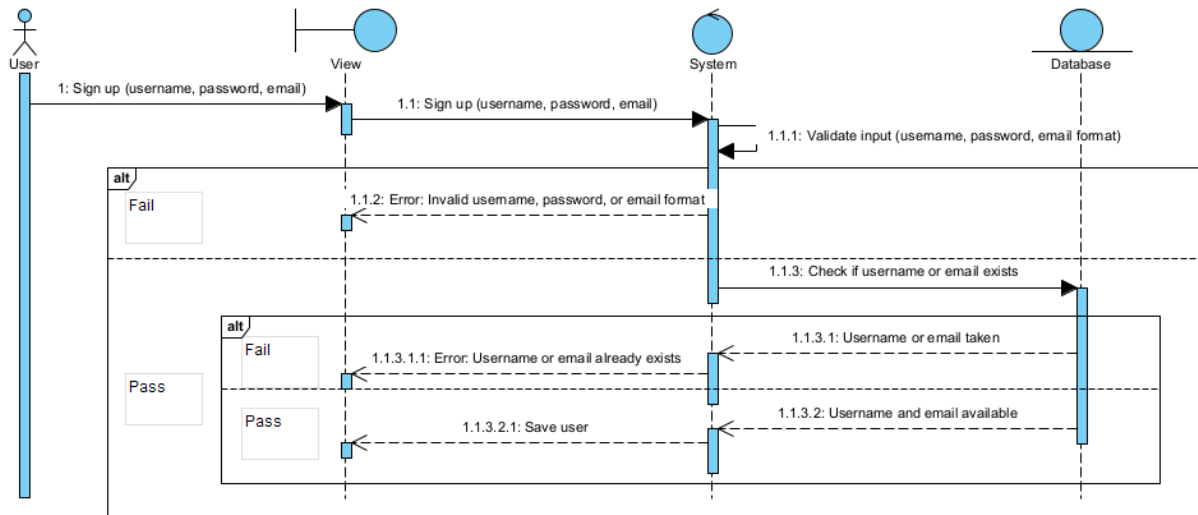
Database server: được tối ưu cho mục đích lưu trữ dữ liệu

3. Biểu đồ hoạt động (Các chức năng chính)

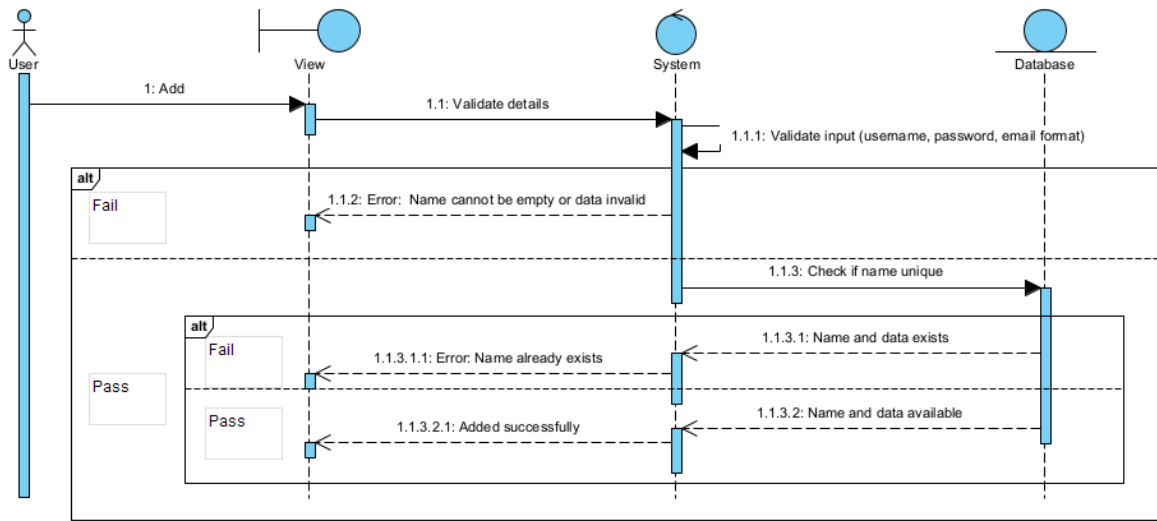
3.1. Đăng nhập



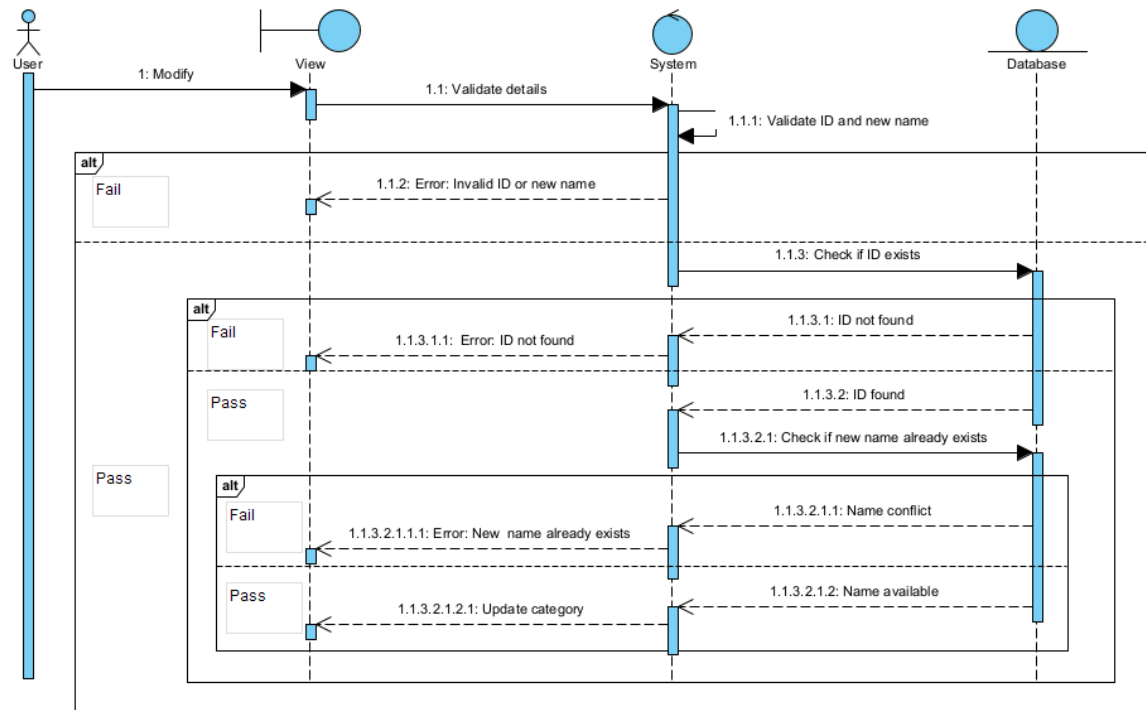
3.2. Đăng ký



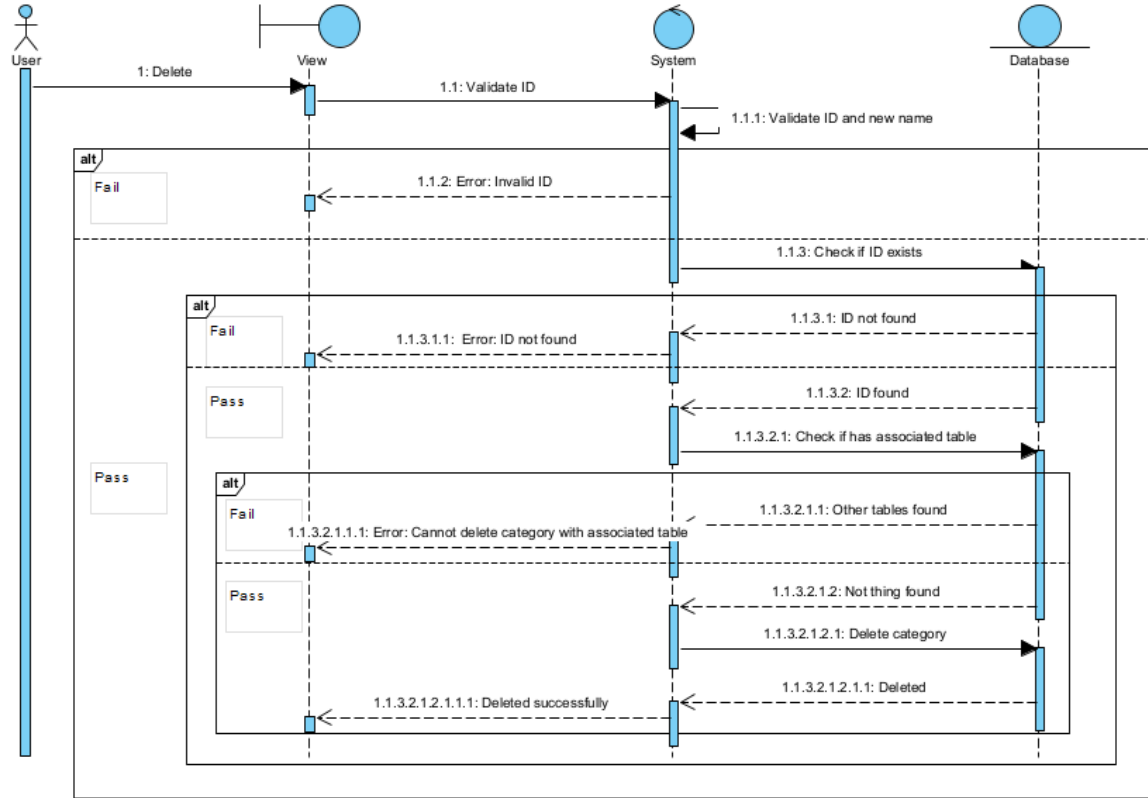
3.3. Thêm (Công thức, Danh mục, Nguyên liệu)



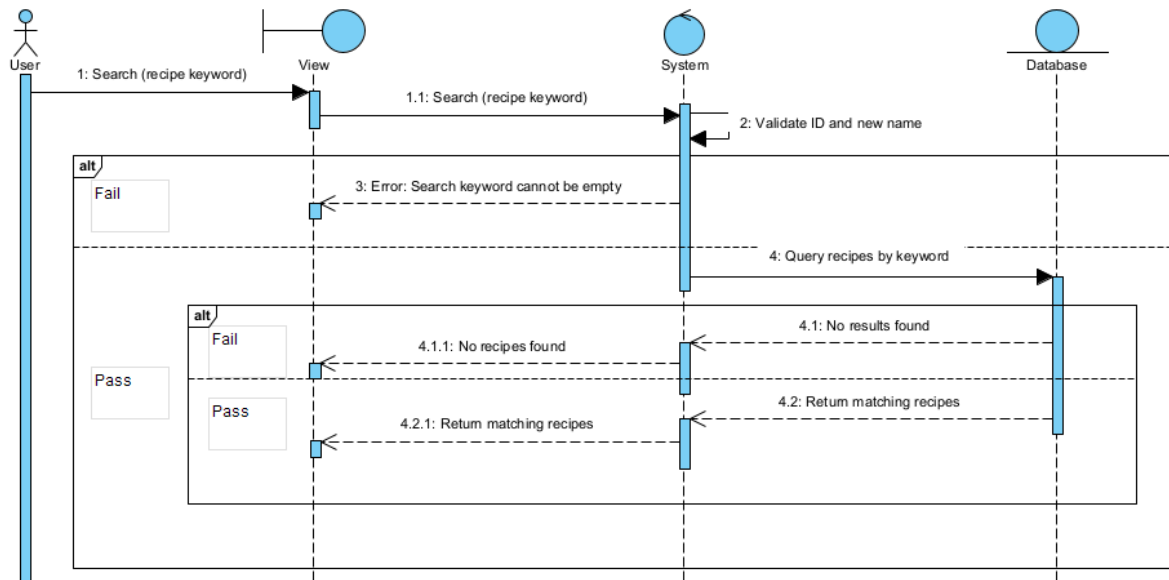
3.4. Sửa (Công thức, Danh mục, Nguyên liệu)



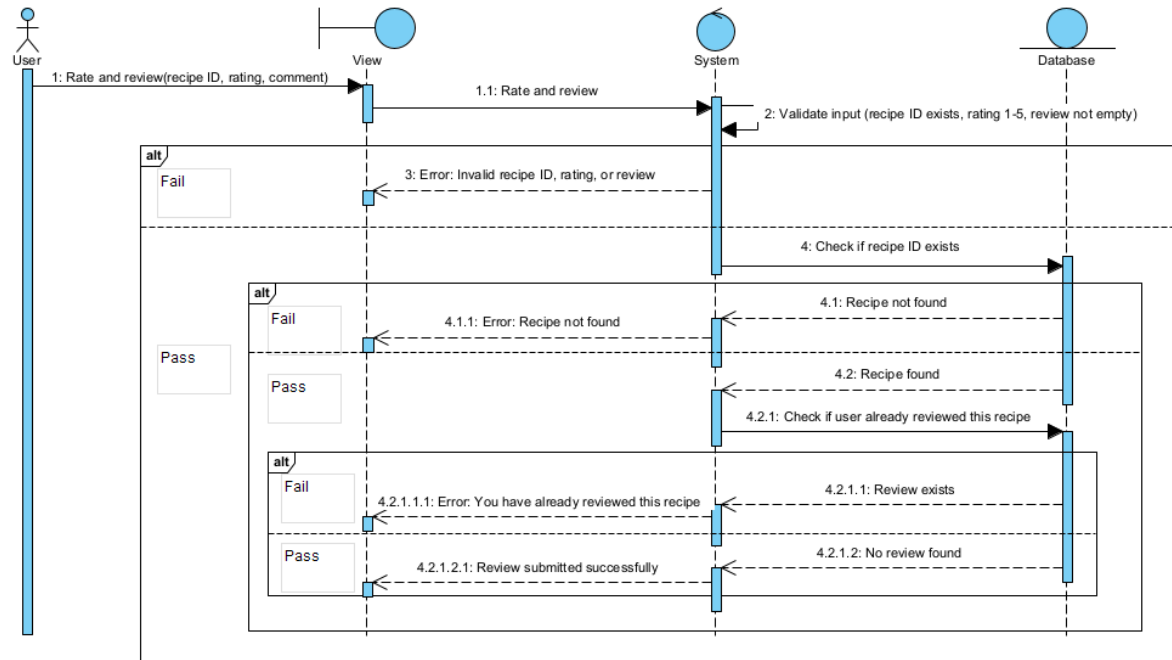
3.5. Xóa (Công thức, Danh mục, Nguyên liệu)



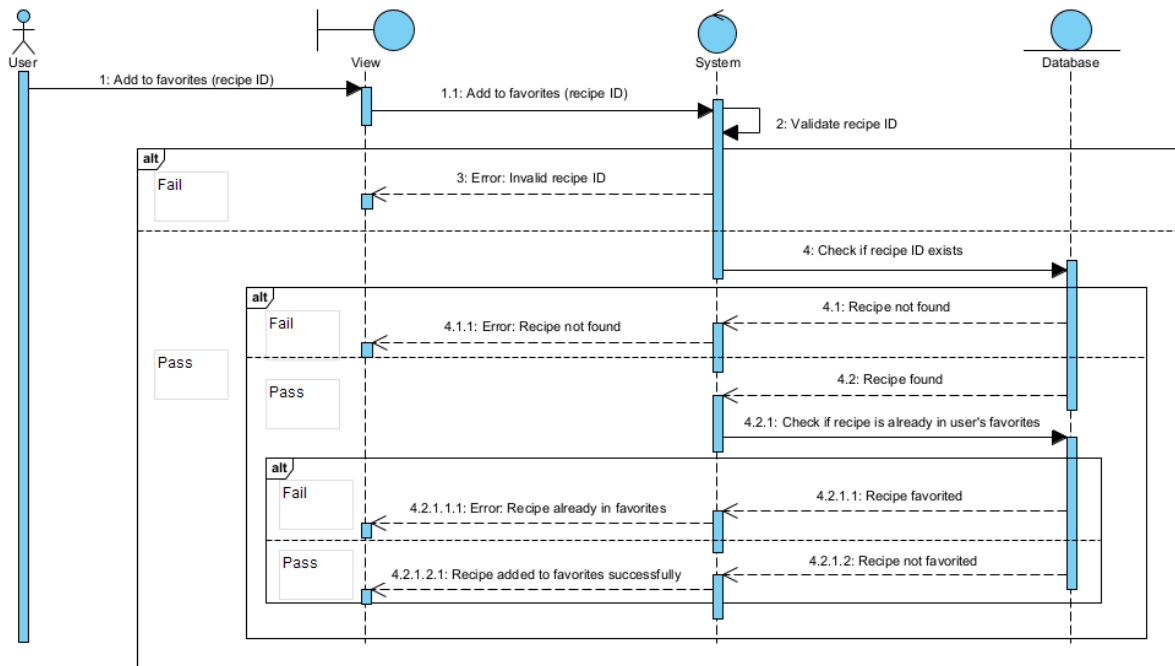
3.6. Tìm kiếm



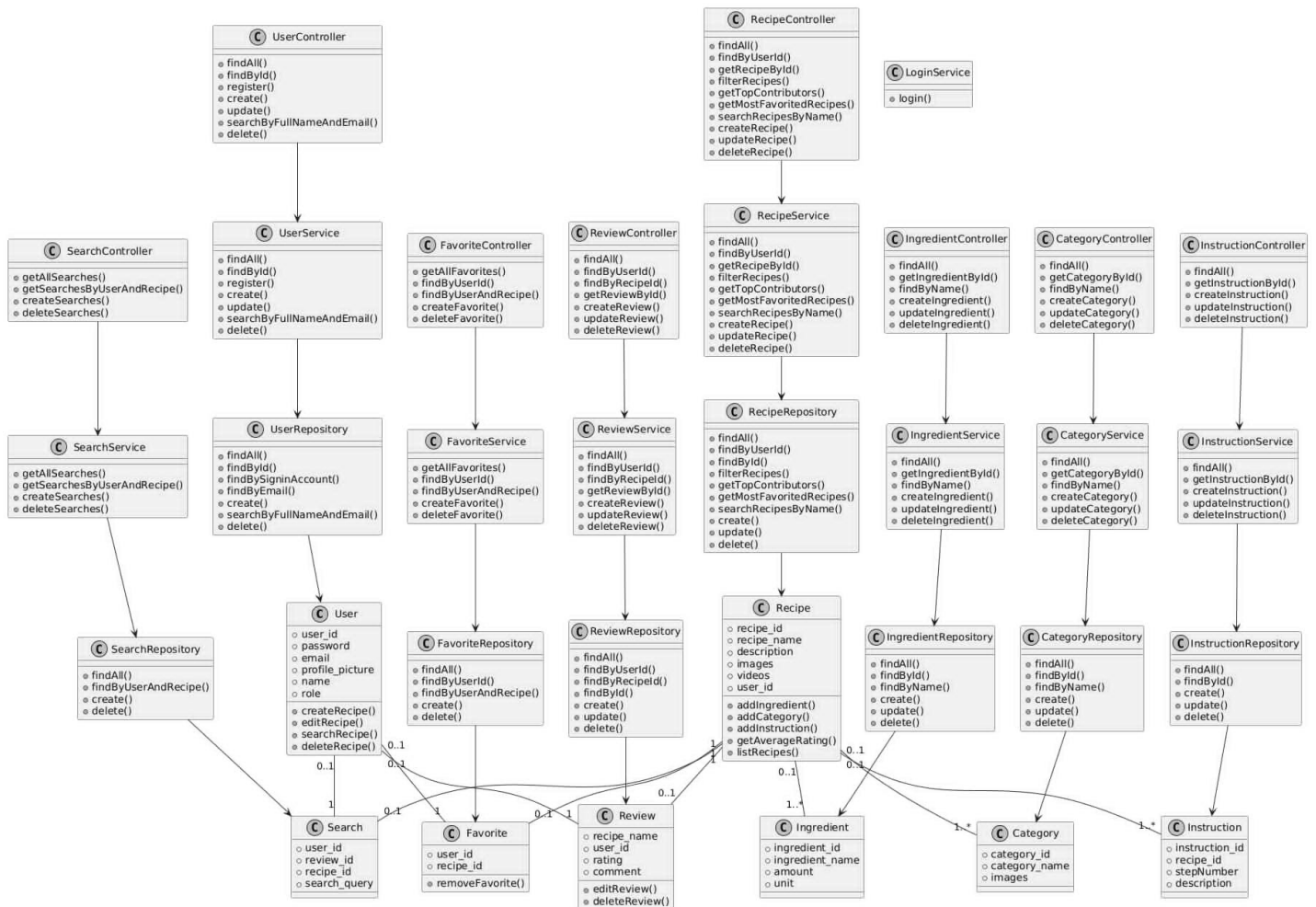
3.7. Đánh giá



3.8. Yêu thích

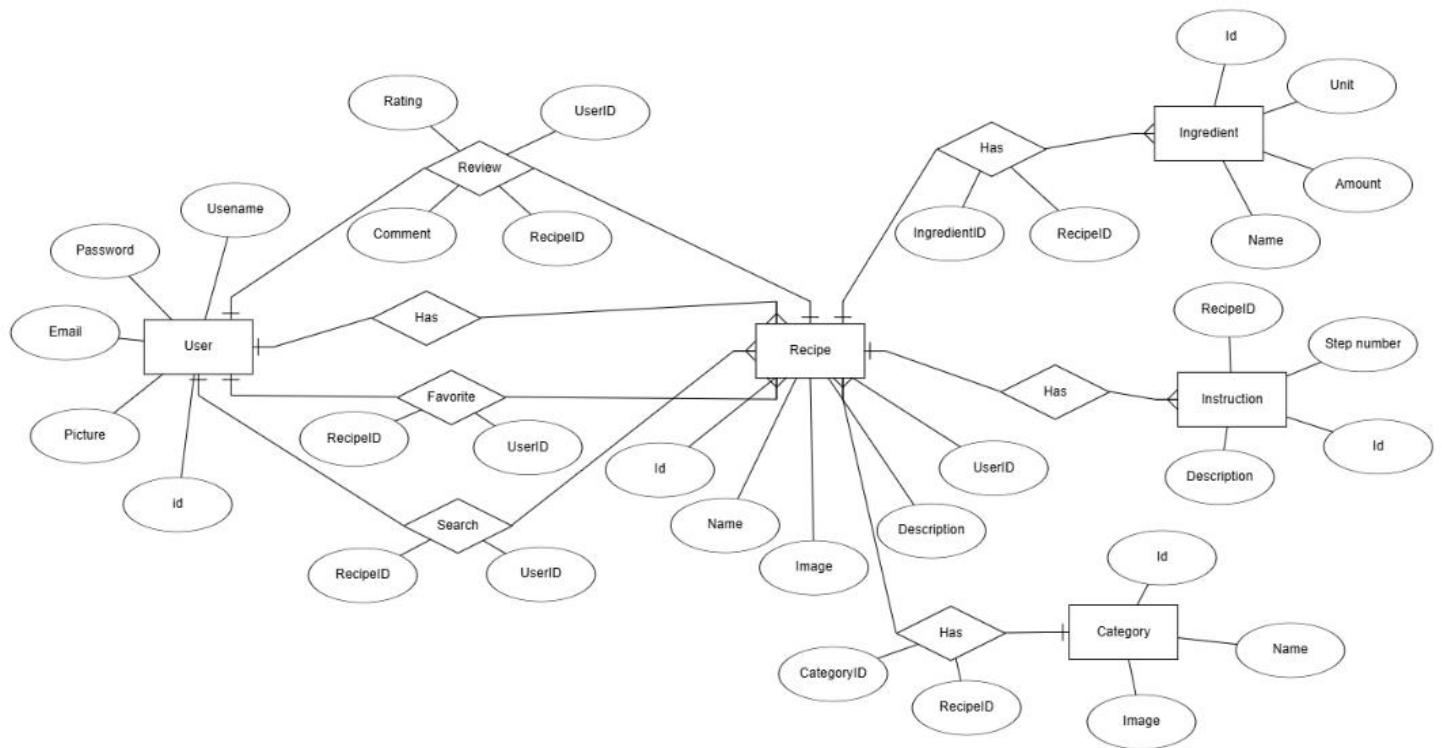


4. Thiết kế lớp (Class Diagram)

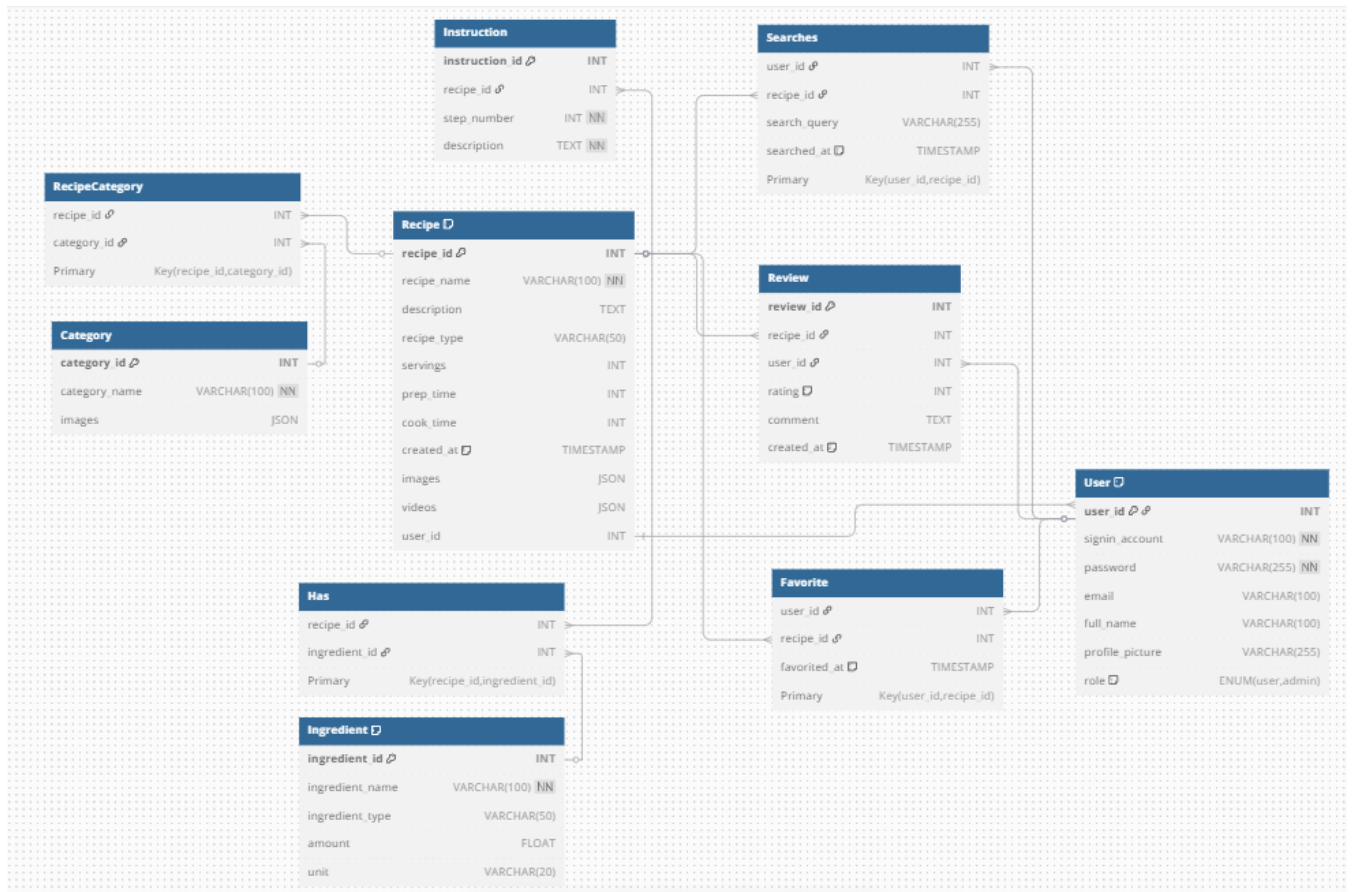


5. Thiết kế cơ sở dữ liệu (ERD, bảng, mối quan hệ)

5.1. Sơ đồ ERD

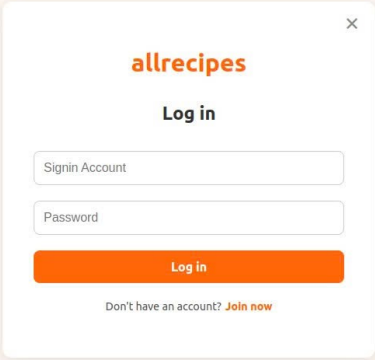


5.2 Database Diagram



6. Thiết kế giao diện người dùng (Mockup / Wireframe)

6.1. Đăng nhập/ Đăng ký



The image shows a wireframe of a login modal for the 'allrecipes' website. The modal is centered on a light orange background. It features the 'allrecipes' logo at the top, followed by the title 'Log in'. Below the title are two input fields: 'Signin Account' and 'Password'. A prominent orange 'Log in' button is positioned below the password field. At the bottom of the modal, there is a link that reads 'Don't have an account? [Join now](#)'. The modal has a close button (an 'X' icon) in the top right corner.

6.2. Trang chủ

dishare

HomeProfileRecipesSearch recipes...


Welcome to Dishare

Discover and Share Your Favorite Recipes


MEATLOAF

SALAD


STEAK



Best Chicken Salad
This creamy chicken salad recipe is the best and a family favorite! I like to use




Classic Macaroni Salad
This creamy macaroni salad always gets lots of compliments. It's an easy




Best Chicken Salad
This creamy chicken salad recipe is the best and a family favorite! I like to use


MOST FAVORITED RECIPES




Best Chicken Salad




Marinated Flank Steak




Chinese Pepper Steak



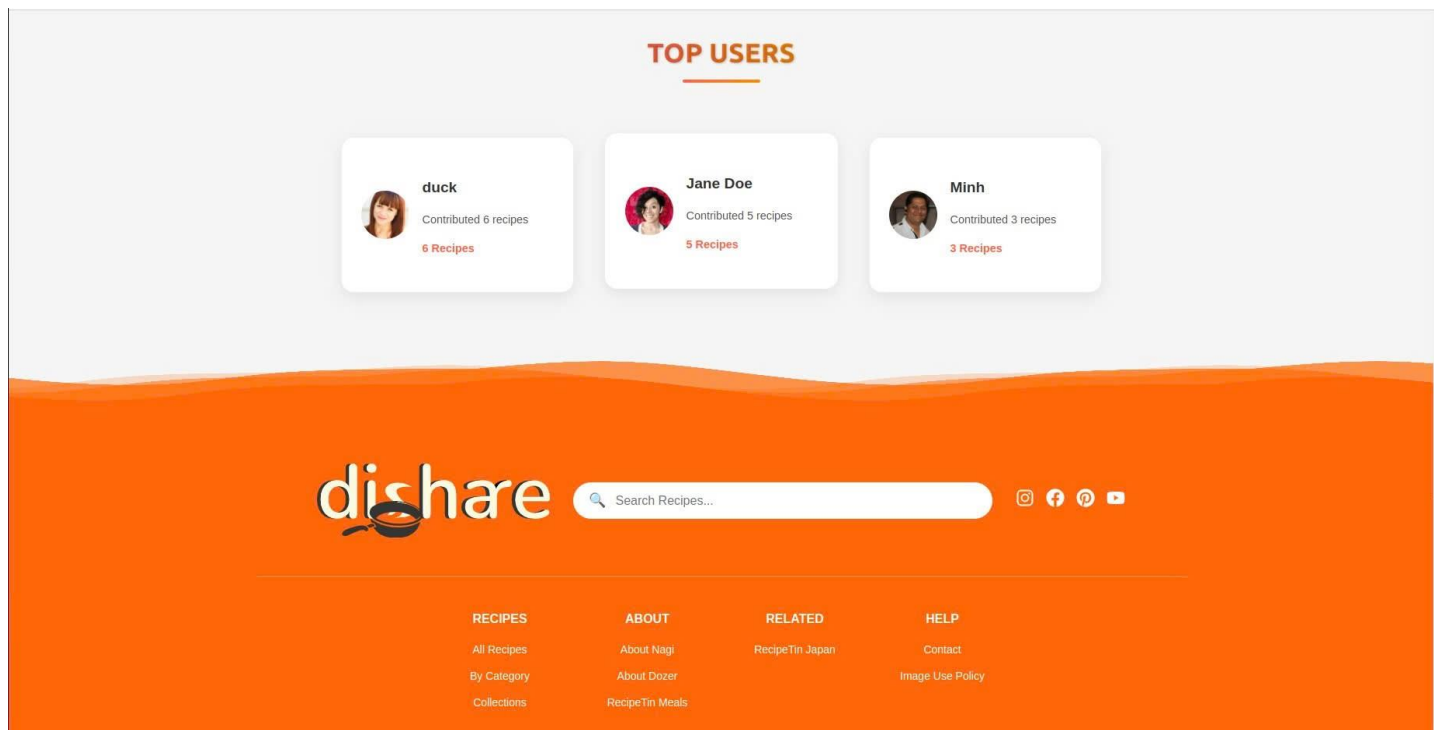
Salisbury Steak



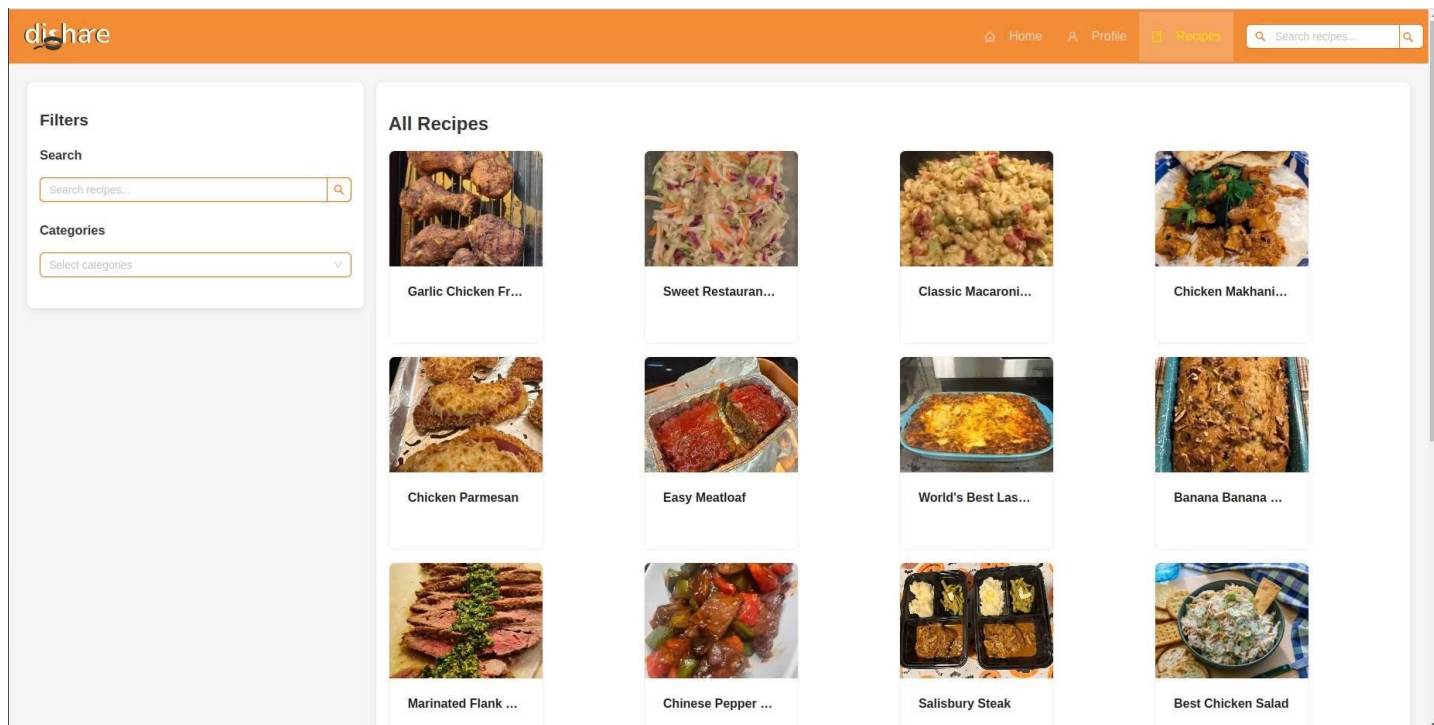
Banana Banana Bread



White Cheese Chicken Lasagna




6.3. Công thức



6.4. Hồ sơ người dùng

dishare

HomeProfileRecipesSearch recipes...



duck
duckky123@gmail.com

Profile

My Recipes

Favorite Recipes

Your Comments

Logout

Profile

duck

Last Name


duckky123@gmail.com

Edit Profile

localhost:5173/profile

dishare

HomeProfileRecipesSearch recipes...



duck
duckky123@gmail.com

Profile

My Recipes

Favorite Recipes

Your Comments

Logout

My Recipes

Marinated Flank Steak

A great flank steak marinade like this one is important if you want a tender, juicy, flavorful steak. Make sure you marinate your flank steak for at least 2 hours for best results or longer if you have time. This wonderful quick and easy recipe also works great when the steak is sliced and used for fajitas.

View

Edit

Delete

Chinese Pepper Steak

A delicious pepper steak recipe served with boiled white rice — easy and made from items I've already got in my cupboards. My mother clipped this recipe from somewhere and it became a specialty of mine; however, I've been unable to find the original source.

View

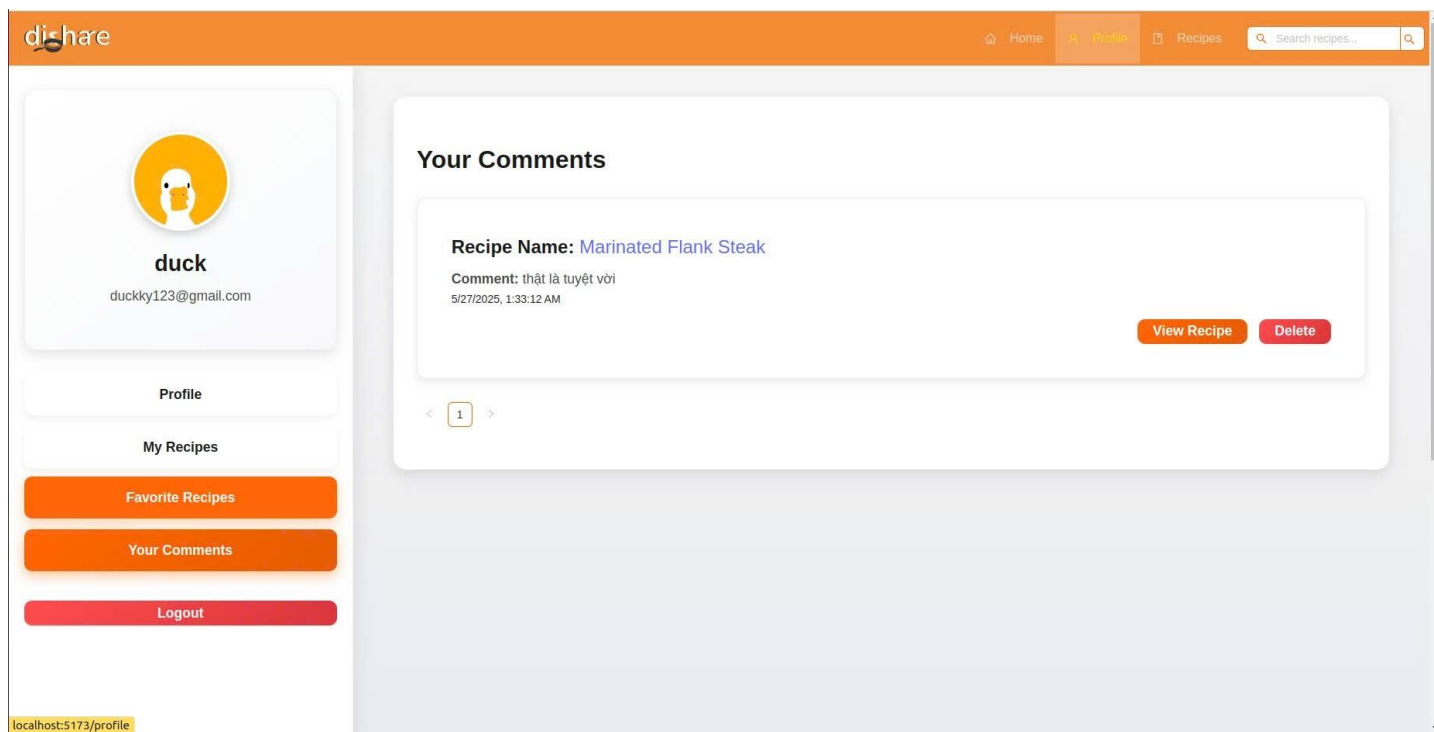
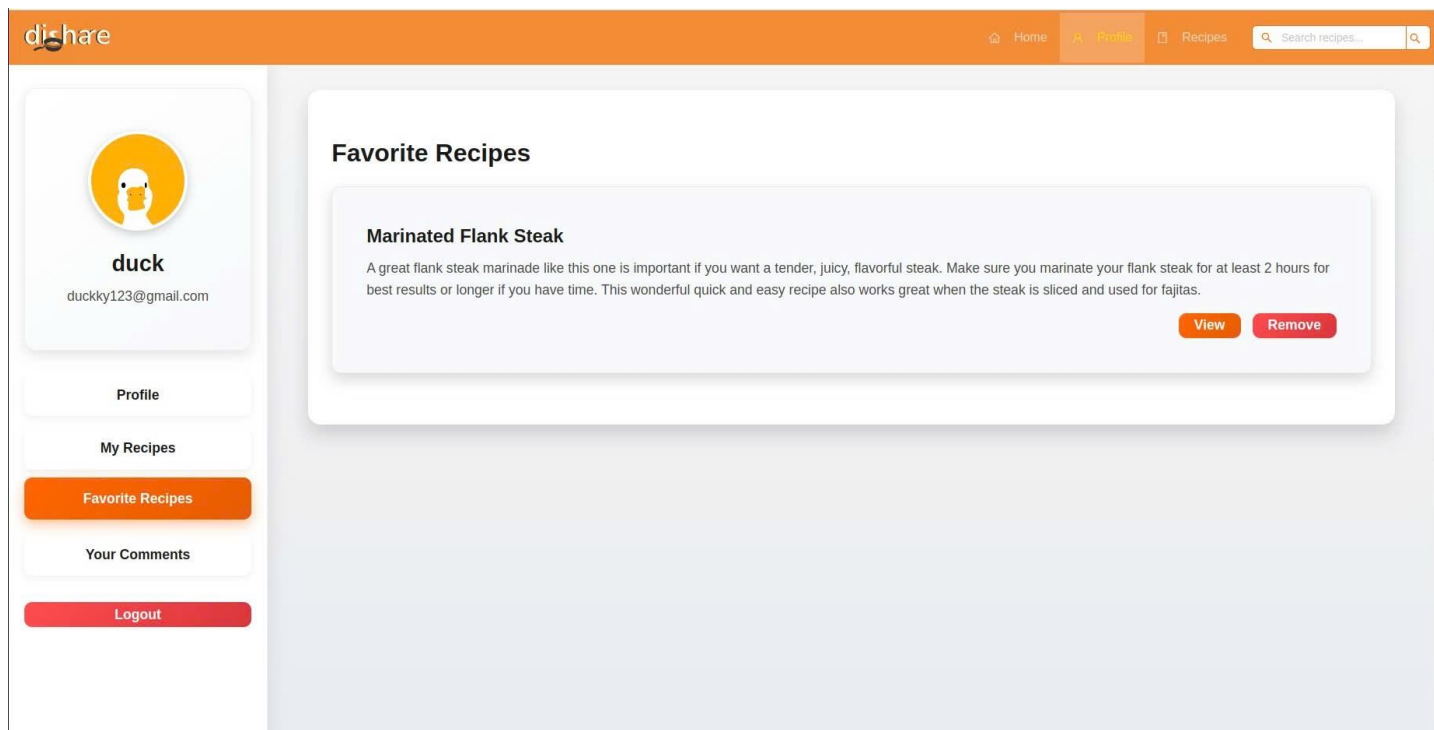
Edit

Delete

Salisbury Steak

Salisbury steak with a savory onion sauce. This yummy recipe has been in my family for years. It's easy to cook but tastes like it took hours to make! I usually make enough extra sauce to pour over mashed potatoes.

localhost:5173/profile





duck

Recipes (6) Reviews (1)



Marinated Flank Steak



Chinese Pepper Steak



Salisbury Steak



6.4. Trang quản lý

di share

Logout

Management

Recipes Management

Users Management

Categories Management

Admin Recipes

Search recipes by name:

Recipe Name	Created At	Categories	Action
Garlic Chicken Fried Chicken	May 26, 2025	Chicken, FRIED CHICKEN	VIEW DELETE MANAGE REVIEWS
Sweet Restaurant Slaw	May 26, 2025	None	VIEW DELETE MANAGE REVIEWS
Classic Macaroni Salad	May 26, 2025	SALAD	VIEW DELETE MANAGE REVIEWS
Chicken Makhani (Indian Butter Chicken)	May 26, 2025	Chicken, Indian	VIEW DELETE MANAGE REVIEWS
Chicken Parmesan	May 26, 2025	Chicken	VIEW DELETE MANAGE REVIEWS
Easy Meatloaf	May 26, 2025	Meatloaf	VIEW DELETE MANAGE REVIEWS
World's Best Lasagna	May 27, 2025	Lasagna	VIEW DELETE MANAGE REVIEWS
Banana Banana Bread	May 27, 2025	Banana bread	VIEW DELETE MANAGE REVIEWS
Marinated Flank Steak	May 27, 2025	Steak	VIEW DELETE MANAGE REVIEWS

di share

Logout

Management

Recipes Management

Users Management

Categories Management

Admin Users

Search users by username or email

Username	Full Name	Email	Action
duckky	Test User	test@example.com	<div>VIEWEDITDELETE</div>
duc123	Jane Doe	jane1231111111111411@example.com	<div>VIEWEDITDELETE</div>
duc1234	Jane Doe	jane1231111111411@example.com	<div>VIEWEDITDELETE</div>
minh123	Minh	minh123@example.com	<div>VIEWEDITDELETE</div>
duccal	duck	duckky123@gmail.com	<div>VIEWEDITDELETE</div>
duccal1	dduck	jane14111@example.com	<div>VIEWEDITDELETE</div>

<1>

10 / page

di share

Logout

Management

Recipes Management

Users Management

Categories Management

Admin Categories

Search categories by name

Category Name	Recipe Count	Action
Banana bread	1	<div>DELETE</div>
Chicken	5	<div>DELETE</div>
FRIED CHICKEN	1	<div>DELETE</div>
Indian	1	<div>DELETE</div>
Lasagna	3	<div>DELETE</div>
Meatloaf	1	<div>DELETE</div>
SALAD	2	<div>DELETE</div>
Steak	3	<div>DELETE</div>

<1>

10 / page

localhost:5173/admin/categories

CHƯƠNG IV: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

1. Môi trường triển khai

Hệ thống được phát triển và triển khai trong môi trường hiện đại, sử dụng kiến trúc client-server. Các thành phần chính bao gồm:

Thành phần	Công nghệ sử dụng
Frontend	ReactJS + Vite + Ant Design
Backend	NestJS (Node.js framework)
Cơ sở dữ liệu	MySQL
Trình quản lý CSDL	MySQL Workbench
Quản lý gói (frontend/backend)	npm
Trình duyệt	Google Chrome
Postman	Kiểm thử API REST
Hệ điều hành phát triển	Windows 10 / Ubuntu 22.04 LTS

2. Cài đặt hệ thống.

2.1. Yêu cầu hệ thống

Để chạy được hệ thống đầy đủ (backend + frontend), cần đảm bảo máy có cài đặt các công cụ sau:

Phần mềm	Phiên bản khuyến nghị
Node.js	>= 18.x
npm	>= 9.x
MySQL	>= 8.0
NestJS CLI	>= 10.x
Vite (React scaffolding)	>= 4.x
Git	>= 2.x
IDE	VS Code

2.2. Các bước cài đặt chương trình.

2.2.1. Clone source code từ Git repository

```
https://github.com/chan-gio/food-recipe.git
```

2.2.2. Cài đặt backend (NestJS)

```
cd backend
npm install
```

Cấu hình kết nối MySQL (file `.env` ví dụ):

```
DB_HOST=localhost
DB_PORT=3306
DB_USERNAME=root
DB_PASSWORD=your_password
DB_NAME=recipe_db
```

Khởi tạo database và chạy migration (nếu có):

```
npm run migration:run
```

Khởi chạy server NestJS:

```
bash
Sao chép
Chỉnh sửa
npm run start:dev
```

2.2.3. Cài đặt frontend (ReactJS + Vite)

```
cd ../frontend
npm install
```

Cấu hình kết nối backend (file `.env` ví dụ):

```
VITE_API_URL=http://localhost:3000/api
```

Chạy ứng dụng React:

```
npm run dev
```

CHƯƠNG V: KẾT QUẢ VÀ ĐÁNH GIÁ

1. Kết quả thử nghiệm hệ thống

Trong quá trình phát triển hệ thống, nhóm đã tiến hành thử nghiệm thông qua hình thức **Unit Test** để đảm bảo tính đúng đắn và ổn định của các thành phần chức năng. Các công cụ được sử dụng bao gồm:

Sử dụng thư viện **Jest** (tích hợp sẵn với NestJS) và **DOM** để viết unit test cho các service, controller.

100% test case frontend đều vượt qua.

2. Thử nghiệm chức năng toàn hệ thống (Manual Test)

Ngoài Unit Test, nhóm cũng thử nghiệm thực tế các chức năng trên trình duyệt với nhiều vai trò khác nhau:

- Người dùng có thể:
 - Đăng ký / Đăng nhập thành công
 - Duyệt và tìm kiếm công thức
 - Xem chi tiết công thức, nguyên liệu, hướng dẫn
 - Thêm công thức vào mục yêu thích
 - Gửi bình luận và đánh giá công thức
- Admin có thể:
 - Duyệt danh sách công thức
 - Xoá / sửa bài
 - Quản lý người dùng
 - Quản lý danh mục
 - Quản lý bình luận

3. Đánh giá hiệu quả hệ thống

Ưu điểm

- **Giao diện thân thiện, dễ sử dụng:** Sử dụng ReactJS kết hợp Ant Design giúp người dùng thao tác mượt mà, dễ tiếp cận.
- **Chức năng đầy đủ:** Bao gồm đăng ký, đăng nhập, thêm/sửa/xem công thức, bình luận, đánh giá, phân loại theo danh mục.
- **Hiệu suất tốt với dữ liệu lớn:**
 - Hệ thống đã triển khai **phân trang (pagination)** giúp tải dữ liệu theo từng trang, tránh dồn quá nhiều dữ liệu về client.
 - **Lazy loading** được áp dụng cho hình ảnh và component giúp cải thiện tốc độ tải trang và hiệu năng tổng thể.
- ☒ **Kiến trúc rõ ràng, dễ bảo trì:** Mô hình phân tầng (Layered Architecture) tách biệt các phần giao diện, xử lý và lưu trữ.
- ☒ **Đảm bảo độ ổn định nhờ kiểm thử:** Đã áp dụng Unit Test cho frontend để đảm bảo độ chính xác và ổn định khi phát triển.

Hạn chế (Cập nhật)

- **Chưa có phân quyền chi tiết nâng cao:** Hiện hệ thống mới chỉ chia User và Admin, chưa phân loại vai trò cụ thể hơn như "Editor", "Chef", "Moderator".
- **Chưa có hệ thống gợi ý thông minh:** Việc tìm kiếm công thức còn phụ thuộc vào từ khóa đơn giản, chưa có gợi ý theo sở thích/ngữ cảnh người dùng.

Đề xuất cải tiến

- Nâng cấp hệ thống phân quyền (RBAC – Role-Based Access Control).
- Thêm tính năng gợi ý công thức theo lịch sử nấu ăn / đánh giá.
- Triển khai **tìm kiếm nâng cao:** theo nguyên liệu, thời gian nấu, độ khó,...
- Bổ sung **bộ lọc và sắp xếp** (filter/sort) trong danh sách công thức.
- Tích hợp **CI/CD pipeline** để tự động test và triển khai hệ thống khi có cập nhật.

TÀI LIỆU THAM KHẢO