# TỔNG QUAN VỀ DỰ ÁN XÂY DỰNG XÂY DỰNG DIỄN ĐÀN CÔNG THỨC NẤU ĂN COOK-CRAFT TÍCH HỢP AI

## Giới thiệu về dự án

Trong bối cảnh số hóa diễn ra mạnh mẽ, nhu cầu tìm kiếm, chia sẻ và cá nhân hóa tri thức ẩm thực ngày càng gia tăng. Các công thức nấu ăn hiện phân tán trên nhiều nền tảng, thiếu tính hệ thống và ít được điều chỉnh theo khẩu vị, dinh dưỡng hay nguyên liệu sẵn có của từng người dùng. Từ thực tiễn đó, dự án **Cook-Craft** được đề xuất nhằm xây dựng một **diễn đàn công thức nấu ăn** kết hợp **trí tuệ nhân tạo (AI)** để nâng cao khả năng tìm kiếm, gợi ý và tương tác trong cộng đồng ẩm thực.

Hệ thống hướng đến hai nhóm người dùng chính:

* Người dùng: đăng tải và quản lý công thức; tìm kiếm, lọc theo danh mục/nguyên liệu; bình luận, đánh giá, thêm vào yêu thích, theo dõi tác giả.
* Quản trị viên: quản lý người dùng, công thức, danh mục, nguyên liệu, đơn vị đo, phân quyền; theo dõi hoạt động hệ thống qua các chỉ số tổng hợp.

Về công nghệ, dự án áp dụng mô hình client–server hiện đại: **ReactJS** cho giao diện người dùng (UI) hướng trải nghiệm, **Spring Boot** cho lớp **backend API** xử lý nghiệp vụ và tích hợp AI, **MySQL** cho **cơ sở dữ liệu** đảm bảo lưu trữ nhất quán và truy vấn hiệu quả. Kiến trúc dịch vụ tuân thủ nguyên tắc RESTful, cơ chế xác thực/ủy quyền an toàn (JWT) và sẵn sàng mở rộng khi lưu lượng tăng.

Thành phần **AI** được tích hợp nhằm:

* Gợi ý công thức theo nguyên liệu sẵn có, khẩu vị và mục tiêu dinh dưỡng.
* Hỗ trợ tóm tắt điểm chính của công thức, và cải thiện khả năng tìm kiếm bằng ngôn ngữ tự nhiên.

Bên cạnh yêu cầu chức năng, dự án chú trọng các yêu cầu phi chức năng: **hiệu năng** (tốc độ phản hồi), **bảo mật** (bảo vệ dữ liệu người dùng), **khả năng mở rộng**, **tính sẵn sàng triển khai** và **trải nghiệm người dùng** mạch lạc, nhất quán trên nhiều thiết bị.

Với các mục tiêu trên, **Cook-Craft** kỳ vọng hình thành một cộng đồng ẩm thực trực tuyến năng động, tạo môi trường học hỏi và sáng tạo, đồng thời mang đến trải nghiệm tìm – chọn – chia sẻ công thức nấu ăn **thông minh, cá nhân hóa và đáng tin cậy**.

### Bối bảnh bài toán

Với sự phát triển mạnh mẽ của Internet, nhu cầu chia sẻ và tìm kiếm kiến thức trong lĩnh vực ẩm thực ngày càng cao. Người dùng không chỉ quan tâm đến việc học nấu ăn mà còn muốn trao đổi kinh nghiệm, sáng tạo công thức mới và tham khảo đánh giá từ cộng đồng. Tuy nhiên, phần lớn các công thức hiện nay chỉ tồn tại trên blog cá nhân, video mạng xã hội hoặc các website đơn lẻ, thiếu sự kết nối giữa người dùng và thiếu các công cụ hỗ trợ thông minh.

Mặt khác, sự phát triển của trí tuệ nhân tạo (AI) mở ra cơ hội để cải thiện trải nghiệm trong lĩnh vực này. AI có thể giúp người dùng lựa chọn công thức phù hợp với nguyên liệu họ có sẵn, đưa ra gợi ý theo chế độ ăn uống hoặc hỗ trợ tìm kiếm nhanh chóng hơn bằng ngôn ngữ tự nhiên. Trong bối cảnh đó, việc xây dựng một diễn đàn công thức nấu ăn tích hợp AI vừa đáp ứng nhu cầu cộng đồng, vừa ứng dụng công nghệ mới là cần thiết và mang tính thực tiễn cao.

### Nghiệp vụ hoạt động

Hệ thống **Cook-Craft** được định hướng phát triển với các nghiệp vụ hoạt động chính như sau:

* Đối với người dùng:
* Khi truy cập vào website, người dùng sẽ thấy giao diện thân thiện, trực quan, hiển thị danh sách các công thức nổi bật, mới đăng và công thức được AI gợi ý dựa trên nguyên liệu hoặc sở thích.
* Người dùng có thể thêm mới, chỉnh sửa hoặc ngưng/mở hoạt động các công thức mà họ tạo ra, cũng như quản lý danh mục, nguyên liệu và đơn vị (unit) của riêng mình.
* Hệ thống cho phép người dùng xem, đánh giá và bình luận các công thức của người khác. Mỗi công thức chỉ cho phép một bình luận từ mỗi người dùng; nếu muốn thêm bình luận mới, người dùng có thể xóa bình luận cũ trước khi đăng bình luận mới, nhằm tránh spam.
* Tất cả công thức hiển thị đầy đủ thông tin: nguyên liệu, các bước thực hiện, hình ảnh minh họa, đơn vị đo lường, thời gian nấu, chế độ ăn hoặc gợi ý AI.
* Người dùng có thể theo dõi công thức yêu thích, lưu bộ sưu tập và theo dõi các tác giả khác để cập nhật công thức mới.
* Đối với quản trị viên:
* Quản trị viên có quyền quản lý toàn bộ người dùng, bao gồm tạo, chỉnh sửa, khóa hoặc xóa tài khoản vi phạm.
* Quản lý toàn bộ công thức, danh mục, nguyên liệu và đơn vị trong hệ thống, bao gồm các thao tác thêm mới, chỉnh sửa, ngưng/mở hoạt động hoặc xóa.
* Quản trị viên có thể xem và đánh giá, bình luận công thức như người dùng thông thường, với quyền xóa bình luận để chỉnh sửa nội dung.
* Hệ thống cung cấp các thống kê và báo cáo về hoạt động của người dùng: số lượng công thức, số lượt đánh giá, mức độ tương tác, số lượng bình luận, danh mục/nhóm nguyên liệu được sử dụng nhiều nhất.
* Quản trị viên đồng thời có thể quản lý tin tức, bài viết hướng dẫn, thông báo hoặc thông tin bổ sung về diễn đàn, giúp duy trì tính cập nhật và thúc đẩy tương tác người dùng.

### Các yêu cầu ràng buộc của hệ thống

#### Các yêu cầu chức năng

Dành cho khách hàng:

* Đăng nhập/Đăng ký tài khoản: người dùng sẽ sử dụng tài khoản để đăng nhập vào hệ thống sử dụng các chức năng.
* Đăng nhập: người dùng sử dụng tài khoản để truy cập vào hệ thống và sử dụng các chức năng nâng cao.
* Đăng ký: cho phép người dùng tạo tài khoản mới với thông tin cơ bản (họ tên, email, mật khẩu).
* Quên mật khẩu: hỗ trợ lấy lại mật khẩu thông qua xác thực email.
* Cập nhật thông tin cá nhân: người dùng có thể thay đổi mật khẩu, ảnh đại diện, thông tin liên hệ.
* Quản lý công thức:
* Thêm công thức mới: người dùng có thể đăng tải công thức với đầy đủ nguyên liệu, đơn vị, các bước nấu, hình ảnh minh họa.
* Sửa công thức: cho phép chỉnh sửa nội dung công thức đã tạo.
* Ngưng/Mở hoạt động công thức: cho phép ẩn hoặc hiển thị công thức trên hệ thống.
* Quản lý dữ liệu cá nhân:
* Quản lý danh mục cá nhân: thêm, sửa, xóa hoặc tạm ngưng các danh mục công thức của riêng mình.
* Quản lý nguyên liệu và đơn vị (unit): thêm mới, chỉnh sửa hoặc ngưng/mở hoạt động.
* Tương tác công thức:
* Xem công thức: người dùng có thể xem chi tiết công thức của mình hoặc của người khác.
* Đánh giá và bình luận: người dùng có thể chấm điểm và viết bình luận cho công thức.
* Quản lý bình luận: mỗi công thức chỉ cho phép một bình luận từ một người dùng. Người dùng có thể xóa bình luận cũ để thêm bình luận mới.
* AI hỗ trợ:
* Gợi ý công thức: hệ thống AI đề xuất công thức dựa trên nguyên liệu có sẵn, chế độ ăn hoặc sở thích.

Dành cho quản trị viên:

* Quản lý người dùng: thêm mới, khóa/mở khóa, chỉnh sửa hoặc xóa tài khoản.
* Quản lý công thức: xem, duyệt, chỉnh sửa, ngưng/mở hoạt động.
* Quản lý danh mục, nguyên liệu, đơn vị: thêm, sửa, xóa hoặc thay đổi trạng thái hoạt động.
* Quản lý nội dung diễn đàn: duyệt và quản lý các bài viết, bình luận, đánh giá để đảm bảo môi trường lành mạnh.
* Thống kê và báo cáo:
* Thống kê số lượng công thức, bình luận, lượt đánh giá, mức độ tương tác của người dùng.
* Thống kê danh mục/nhóm nguyên liệu được sử dụng nhiều nhất.

#### Các yêu cầu phi chức năng

* Thiết kế thân thiện với người dùng, phù hợp với người không chuyên về công nghệ.
* Hiển thị thông tin rõ ràng, dễ tìm và dễ truy cập.
* Tốc độ tải trang nhanh để cải thiện trải nghiệm người dùng.
* Tính khả dụng cao, hỗ trợ nhiều thiết bị khác nhau (mobile, tablet, desktop).
* Phản hồi người dùng rõ ràng và chính xác khi thực hiện các hành động.
* Hiệu suất và khả năng mở rộng:
* Đảm bảo hiệu suất cao khi có nhiều người truy cập đồng thời.
* Hỗ trợ lưu trữ trang và nén dữ liệu để tăng tốc độ tải trang.
* Khả năng mở rộng hệ thống dễ dàng để đáp ứng nhu cầu phát triển trong tương lai.
* Tính bảo mật:
* Xác thực và phân quyền rõ ràng để bảo vệ thông tin người dùng.
* Mã hóa dữ liệu nhạy cảm khi lưu trữ và truyền tải.

## Công cụ, phương pháp và kỹ thuật sử dụng

### Giới thiệu về thư viện ReactJS

* Khái niệm:
* ReactJS (hay React) là một thư viện JavaScript mã nguồn mở do Facebook (nay là Meta) phát triển, dùng để xây dựng giao diện người dùng (UI). React tập trung vào việc tạo ra các giao diện tương tác cao và có thể tái sử dụng, giúp quá trình phát triển trở nên hiệu quả và dễ dàng bảo trì hơn. React không phải là một framework MVC (Model-View-Controller) hoàn chỉnh, mà chỉ tập trung vào tầng "View" (hiển thị), vì vậy nó thường được kết hợp với các thư viện khác để xây dựng một ứng dụng đầy đủ.
* Ưu điểm:
* Hiệu suất cao với Virtual DOM: React sử dụng một mô hình DOM ảo (Virtual DOM) để so sánh và chỉ cập nhật những thay đổi cần thiết lên DOM thực, giúp tăng tốc độ và hiệu suất của ứng dụng, đặc biệt với các ứng dụng có dữ liệu thay đổi liên tục.
* Component-based (Thiết kế dựa trên thành phần): React cho phép bạn chia giao diện người dùng thành các component nhỏ, độc lập và có thể tái sử dụng. Điều này giúp mã nguồn trở nên gọn gàng, dễ quản lý và dễ bảo trì.
* Cú pháp JSX: JSX (JavaScript XML) cho phép bạn viết mã HTML trực tiếp trong JavaScript, làm cho việc xây dựng giao diện trở nên trực quan và dễ đọc hơn.
* Thân thiện với SEO: Với sự hỗ trợ của các thư viện render dữ liệu phía máy chủ, React có thể giúp các trang web được tối ưu hóa tốt hơn cho các công cụ tìm kiếm.
* Cộng đồng lớn và mạnh mẽ: Với sự phổ biến của mình, React có một cộng đồng lập trình viên rất lớn, cung cấp nhiều tài liệu, công cụ và sự hỗ trợ khi bạn gặp vấn đề.
* Nhược điểm:
* Học các khái niệm mới: Người mới bắt đầu có thể gặp khó khăn với các khái niệm như JSX, Virtual DOM, Props, State và luồng dữ liệu một chiều.
* Chỉ là thư viện View: React chỉ tập trung vào tầng View, vì vậy bạn cần kết hợp với các thư viện khác như Redux để quản lý trạng thái phức tạp, hoặc các thư viện routing để xử lý điều hướng.
* Thay đổi liên tục: React có tốc độ cập nhật khá nhanh, điều này đôi khi yêu cầu các nhà phát triển phải liên tục học hỏi và thích nghi với những thay đổi mới.
* Cấu trúc và thành phần chính:
* Thành phần (Components): Đây là khối xây dựng cơ bản của React. Một component là một phần của giao diện, có thể chứa HTML, CSS và JavaScript riêng. Có hai loại component chính:
* Class Component: Viết dưới dạng một class JavaScript.
* Function Component: Viết dưới dạng một hàm, được sử dụng phổ biến hơn hiện nay nhờ React Hooks.
* JSX (JavaScript XML): JSX là một cú pháp mở rộng của JavaScript, cho phép bạn viết mã giống HTML ngay trong file JavaScript. Nó giúp việc tạo và quản lý giao diện trở nên dễ dàng và trực quan.
* Props (Properties): Props là cách để truyền dữ liệu từ component cha xuống component con. Dữ liệu này chỉ có một chiều và không thể thay đổi từ component con, đảm bảo tính nhất quán của luồng dữ liệu.
* State (Trạng thái): State là một đối tượng JavaScript chứa dữ liệu hoặc thông tin về trạng thái hiện tại của một component. Khi state thay đổi, React sẽ tự động cập nhật lại giao diện tương ứng.
* Virtual DOM (DOM ảo): Virtual DOM là một bản sao nhẹ của DOM thực tế. Khi có bất kỳ thay đổi nào về dữ liệu, React sẽ so sánh Virtual DOM hiện tại với phiên bản trước đó và chỉ cập nhật những thay đổi cần thiết lên DOM thật. Điều này giúp giảm thiểu số lượng thao tác trực tiếp lên DOM và tối ưu hóa hiệu suất.

### Giới thiệu về framework Spring Boot

* Khái niệm:
* Spring Boot là nền tảng phát triển backend mạnh mẽ, loại bỏ cấu hình phức tạp của Spring truyền thống, giúp triển khai API nhanh chóng và hiệu quả.
* Ưu điểm:
* Tự động cấu hình, không cần XML.
* Hỗ trợ server nhúng như Tomcat giúp triển khai nhanh.
* Tích hợp dễ dàng với JPA, bảo mật, gửi mail.
* Hệ sinh thái Spring mạnh, cập nhật liên tục.
* Nhược điểm:
* Tốn nhiều RAM khi chạy ứng dụng lớn.
* Khởi động chậm hơn so với các ứng dụng nhẹ.
* Cấu trúc và các thành phần chính
* Controller: tiếp nhận yêu cầu HTTP.
* Service: xử lý logic nghiệp vụ.
* Repository: thao tác dữ liệu với JPA.
* Entity: ánh xạ dữ liệu với bảng trong cơ sở dữ liệu.

### Hệ quản trị cơ sở dữ liệu quan hệ MySQL

* Khái niệm:
* MySQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, nổi tiếng vì tính ổn định, hiệu năng cao và khả năng tích hợp với nhiều nền tảng.
* Các thành phần cơ bản của MySQL:
* Server: Quản lý kết nối, xử lý truy vấn và lưu trữ dữ liệu.
* Databases: Chứa các bảng, chỉ số và ràng buộc, có thể tạo nhiều cơ sở dữ liệu trên một server
* Tables: Thành phần cơ bản nhất, chứa các hàng (rows) và cột (columns).
* Columns: Thuộc tính của bảng, có tên và kiểu dữ liệu.
* Rows: Bản ghi của bảng, gồm các giá trị cho các cột tương ứng.
* Indexes: Tăng tốc độ truy vấn dữ liệu.
* Views: Tạo “cửa sổ” ảo của dữ liệu từ các bảng khác nhau.
* Triggers: Chương trình tự động kích hoạt khi có sự kiện xảy ra.
* Ưu điểm:
* Miễn phí và mã nguồn mở, có cộng đồng lớn hỗ trợ.
* Hiệu suất tốt, xử lý hàng ngàn truy vấn mỗi giây an toàn và bảo mật
* Linh hoạt và dễ sử dụng, hỗ trợ nhiều ngôn ngữ lập trình
* Nhược điểm:
* Không hỗ trợ tính năng NoSQL.
* Giới hạn khi xử lý dữ liệu lớn.
* Chậm áp dụng các tính năng mới.
* Sự tương tác giữa MySQL và Spring Boot
* Spring Boot sử dụng Spring Data JPA kết hợp Hibernate để thao tác cơ sở dữ liệu.
* Dự liệu được ánh xạ thông qua các Entity class.
* Cấu hình đơn giản qua application.yaml, application.properties.
* Sử dụng JpaRepository để thao tác dữ liệu mà không cần viết SQL thuần.

### Giới thiệu mô hình Client-Server

* Khái niệm:
* Mô hình Client – Server là một kiến trúc phần mềm phổ biến trong phát triển ứng dụng, đặc biệt là ứng dụng web. Trong mô hình này, hệ thống được chia thành hai phần chính: Client (máy khách) và Server (máy chủ). Client gửi yêu cầu, còn Server xử lý và trả về kết quả.
* Cấu trúc mô hình Client – Server:
* Client: là phía người dùng, nơi diễn ra các thao tác giao diện như nhập liệu, nhận kết quả, và tương tác với người dùng. Thường được xây dựng bằng HTML, CSS, JavaScript hoặc framework như Angular, React, Vue, …
* Server: là phía xử lý nghiệp vụ, quản lý cơ sở dữ liệu và phản hồi lại yêu cầu từ client. Thường dùng các ngôn ngữ như Java (Spring Boot), Node.js, PHP, ...
* Giao tiếp: client và server giao tiếp với nhau thông qua giao thức HTTP/HTTPS, với định dạng dữ liệu phổ biến là JSON hoặc XML.
* Áp dụng trong ReactJS và Spring Boot:
* Client (ReactJS): Gửi request HTTP thông qua các service, hiển thị dữ liệu bằng data binding.
* Server (Spring Boot): Nhận request, xử lý nghiệp vụ, truy vấn CSDL và trả response về cho client.
* Ưu điểm:
* Phân tách rõ ràng: Client và server có thể phát triển, triển khai và nâng cấp độc lập.
* Tái sử dụng và mở rộng: Server có thể phục vụ nhiều client khác nhau như web, mobile, ...
* Bảo mật và kiểm soát tốt hơn: Dữ liệu và logic nghiệp vụ nằm ở phía server, giúp kiểm soát quyền truy cập hiệu quả.
* Nhược điểm:
* Phụ thuộc vào kết nối mạng: Không thể hoạt động nếu không có mạng.
* Tăng độ phức tạp khi triển khai: Cần triển khai cả server lẫn client, cấu hình CORS, bảo mật, phân quyền.

## Kết luận

Trong chương này, chúng ta đã trình bày tổng quan về dự án mạng xã hội chia sẻ công thức nấu ăn Cook-Craft, bao gồm mục tiêu phát triển, bối cảnh bài toán, nghiệp vụ hoạt động, các yêu cầu chức năng và phi chức năng, cũng như công cụ và kỹ thuật sẽ sử dụng. Hệ thống được định hướng không chỉ là nơi chia sẻ công thức nấu ăn mà còn trở thành một cộng đồng tương tác, nơi người dùng có thể học hỏi, đánh giá, thảo luận và nhận gợi ý công thức thông minh từ AI.

Dự án lựa chọn ReactJS cho frontend, Spring Boot cho backend và MySQL cho hệ quản trị cơ sở dữ liệu nhằm đảm bảo hiệu suất cao, tính bảo mật tốt, khả năng mở rộng, đồng thời dễ dàng triển khai trong thực tế. Đây đều là các công nghệ phổ biến, mạnh mẽ và phù hợp cho việc xây dựng một nền tảng trực tuyến hiện đại, thân thiện với người dùng.

# **PHÂN TÍCH ĐẶC TẢ WEBSITE MẠNG XÃ HỘI CHIA SẺ CÔNG THỨC NẤU ĂN COOK-**CRAFT

## Các tác nhân hệ thống

Trong hệ thống website mạng xã hội Cook-craft, các tác nhân chính là con người tương tác trực tiếp với hệ thống, được phân thành hai vai trò chính như sau:

Bảng . Bảng vai trò các tác nhân hệ thống

|  |  |  |
| --- | --- | --- |
| **Tác nhân** | **Khách hàng (User)** | **Quản trị viên (Admin)** |
| **Vai trò** | Khách hàng là đối tượng sử dụng hệ thống với mục đích tìm kiếm và chia sẻ các công thức nấu ăn.  Họ tương tác với giao diện người dùng để tìm kiếm các công thức, tham khảo thông tin về các món ăn, nguyên liệu thực phẩm và đánh giá, bình luận cho công thức. | Quản trị viên là người theo dõi, quản lý toàn bộ hệ thống, có quyền kiểm soát và thao tác với tất cả các dữ liệu, nhằm đảm bảo hoạt động của website diễn ra ổn định và hiệu quả. |

## Biểu đồ use case mức hệ thống

Biểu đồ ca sử dụng ở mức hệ thống được trình bày trong Hình 2.1 nhằm mô tả tổng quan các chức năng chính mà hệ thống cung cấp cho người dùng.

A diagram of a diagram

AI-generated content may be incorrect.

Hình . Biểu đồ use case mức hệ thống

## Đặc tả chi tiết các use case

### Use case Đăng nhập

Chức năng “Đăng nhập” cho phép người dùng xác thực tài khoản để truy cập vào hệ thống. Biểu đồ use case mô tả chức năng này được thể hiện như sau:

A diagram of a diagram

AI-generated content may be incorrect.

Hình . Biểu đồ use case Đăng nhập

Tiếp theo là biểu đồ hoạt động (Activity Diagram) minh họa quy trình xử lý của use case “Đăng nhập”, nhằm hỗ trợ cho phần đặc tả chi tiết. Biểu đồ này được thể hiện như sau:

A screenshot of a diagram

AI-generated content may be incorrect.

Hình . Biểu đồ hoạt động use case Đăng nhập

Chức năng “Đăng nhập” được mô tả chi tiết trong bảng 2.1 thông qua đặc tả use case. Nội dung bảng thể hiện rõ các bước thực hiện, điều kiện và luồng xử lý chức năng này.

Bảng . Đặc tả chi tiết use case "Đăng nhập"

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC01 | **Tên Use case** | Đăng nhập |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Use case này cho phép khách hàng đăng nhập tài khoản vào hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng kích vào nút “Đăng nhập” ở góc trên bên phải màn hình. | | |
| **Tiền điều kiện** | Khách hàng chưa đăng nhập vào hệ thống hoặc phiên đăng nhập đã hết hạn. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Kích vào nút “Đăng nhập” ở góc trên bên phải màn hình. | | 2 | Hệ thống | Hiển thị form nhập thông tin đăng nhập. | | 3 | Khách hàng | Nhập thông tin đăng nhập gồm tên đăng nhập, mật khẩu rồi kích nút “Đăng nhập”. | | 4 | Hệ thống | Kiểm tra thông tin và cho phép người dùng đăng nhập vào hệ thống. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 4a | Hệ thống | Khi tên đăng nhập hoặc mật khẩu không chính xác, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại. | | | |
| **Hậu điều kiện** | Không có | | |

### Use case Tìm kiếm công thức

### Use case Quản lý công thức

### Use case Quản lý danh mục

### Use case Quản lý nguyên liệu

### Use case Quản lý đơn vị tính

### Use case Bình luận và đánh giá

### Use case Quản lý thông tin cá nhân

### Use case Thống kê

### Use case Quản lý tài khoản người dùng

## Kết luận

Trong chương này, hệ thống được phân tích dựa trên các tác nhân chính: khách hàng và quản trị viên. Các chức năng cốt lõi được mô tả thông qua use case và biểu đồ hoạt động. Việc đặc tả chi tiết từng use case giúp làm rõ luồng xử lý của hệ thống, tạo tiền đề cho quá trình thiết kế phần mềm ở chương sau.

# THIẾT KẾ WEBSITE DIỄN ĐÀN CHIA SẺ CÔNG THỨC NẤU ĂN COOK-CRAFT

## Thiết kế kiến trúc phần mềm

### Frontend (Client side)

### Backend (Server side)

### Database layer

### Mô hình tổng thể hệ thống

### Các thành phần kiến trúc chính

### Lưu đồ kiến trúc hệ thống

## Thiết kế cấu trúc phần mềm

## Thiết kế cơ sở dữ liệu

### Biểu đồ mô tả cấu trúc dữ liệu

### Mô tả cơ sở dữ liệu

## Thiết kế giao diện

## Kết luận