# 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 2.1 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 2.1 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 2.2 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 2.3 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 2.2 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. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Không có | | |

### Use case Tìm kiếm công thức

Chức năng “Tìm kiếm công thức” cho phép người dùng Tìm kiếm các công thức dựa vào từ khóa và bộ lọc. Biểu đồ use case mô tả chức năng này được thể hiện như sau:

Hình 2.4 Biểu đồ use case Tìm kiếm công thức

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

Hình 2.5 Biểu đồ hoạt động use case Tìm kiếm công thức

Chức năng “Tìm kiếm công thức” 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 2.3 Đặc tả chi tiết use case Tìm kiếm công thức

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC02 | **Tên Use case** | Tìm kiếm công thức |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Use case này cho phép khách hàng tìm kiếm các công thức bằng từ khóa và bộ lọc. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng nhập từ khóa vào ô tìm kiếm hoặc chọn các bộ lọc ở trang Newfeeds, Công thức. | | |
| **Tiền điều kiện** | Không. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Nhập từ khóa cần tìm kiếm vào ô tìm kiếm và chọn các bộ lọc nếu cần. | | 2 | Hệ thống | Hiển thị danh sách các công thức có liên quan đến từ khóa và bộ lọc của khách hàng. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Không có | | |

### Use case Quản lý công thức

Chức năng “Quản lý công thức” cho phép người dùng quản lý các công thức. Biểu đồ use case mô tả chức năng này được thể hiện như sau:

Hình 2.6 Biểu đồ use case Quản lý công thức

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

Hình 2.7 Biểu đồ hoạt động use case Quản lý công thức

Chức năng “Quản lý công thức” 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 2.4 Đặc tả chi tiết use case Quản lý công thức

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC03 | **Tên Use case** | Quản lý công thức |
| **Tác nhân** | Khách hàng, Quản trị viên | | |
| **Mô tả** | Use case này cho phép khách hàng quản lý các công thức của riêng mình và quản trị viên quản lý tất cả các công thức trong hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng hoặc quản trị viên kích vào Avatar ở góc trên bên phải đối với người dùng và kích vào nút “Admin Home” ở góc trên bên phải, sau đó kích vào nút “Quản lý công thức”. | | |
| **Tiền điều kiện** | Khách hàng cũng như quản trị viên phải đăng nhập vào hệ thống với tài khoản đã được phân quyền. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1a | Khách hàng | Kích vào nút Avatar ở góc trên bên phải. | | 1b | Quản trị viên | Kích vào nút “Admin Home” ở góc trên bên phải. | | 2 | Khách hàng, quản trị viên | Kích vào nút “Quản lý công thức”. | | 3 | Hệ thống | Hiển thị danh sách các công thức phù hợp với từng người dùng. | | 4 | Khách hàng, quản trị viên | Kích vào nút “Xem chi tiết” của 1 công thức. | | 5 | Hệ thống | Hiển thị chi tiết công thức được chọn. | | 6 | Khách hàng, quản trị viên | Kích vào nút “Thêm công thức”. | | 7 | Hệ thống | Hiển thị form nhập công thức mới. | | 8 | Khách hàng, quản trị viên | Điền đầy đủ thông tin công thức mới và kích vào nút “Thêm mới”. | | 9 | Hệ thống | Kiểm tra thông tin. | | 10 | Hệ thống | Thêm công thức mới vào hệ thống. | | 11 | Hệ thống | Hiển thị thông báo thành công và quay về màn hình danh sách công thức. | | 12 | Khách hàng, quản trị viên | Kích vào nút “Sửa” ở cột cuối của 1 dòng công thức bất kỳ. | | 13 | Hệ thống | Hiển thị form sửa công thức với thông tin công thức vừa được chọn. | | 14 | Khách hàng, quản trị viên | Nhập thông tin cần sửa vào form và kích vào nút “Sửa”. | | 15 | Hệ thống | Kiểm tra thông tin. | | 16 | Hệ thống | Lưu công thức vào hệ thống. | | 17 | Hệ thống | Hiển thị thông báo lưu thành công và quay lại màn hình danh sách công thức. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 8a, 14a | Khách hàng, quản trị viên` | Nếu khách hàng hoặc quản trị viên kích vào nút “Hủy”, hệ thống sẽ dừng use case và quay lại màn hình danh sách công thức. | | 9a, 15a | Hệ thống | Nếu thông tin nhập vào không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Cập nhật thông tin của công thức vào hệ thống. | | |

### Use case Quản lý danh mục

Chức năng “Quản lý danh mục” cho phép người dùng quản lý các danh mục. Biểu đồ use case mô tả chức năng này được thể hiện như sau:

Hình 2.8 Biểu đồ use case Quản lý danh mục

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

Hình 2.9 Biểu đồ hoạt động use case Quản lý danh mục

Chức năng “Quản lý danh mục” 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 2.5 Đặc tả chi tiết use case Quản lý danh mục

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC04 | **Tên Use case** | Quản lý công thức |
| **Tác nhân** | Khách hàng, Quản trị viên | | |
| **Mô tả** | Use case này cho phép khách hàng quản lý các danh mục của riêng mình và quản trị viên quản lý tất cả các danh mụctrong hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng hoặc quản trị viên kích vào Avatar ở góc trên bên phải đối với người dùng và kích vào nút “Admin Home” ở góc trên bên phải, sau đó kích vào nút “Quản lý danh mục”. | | |
| **Tiền điều kiện** | Khách hàng cũng như quản trị viên phải đăng nhập vào hệ thống với tài khoản đã được phân quyền. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1a | Khách hàng | Kích vào nút Avatar ở góc trên bên phải. | | 1b | Quản trị viên | Kích vào nút “Admin Home” ở góc trên bên phải. | | 2 | Khách hàng, quản trị viên | Kích vào nút “Quản lý danh mục”. | | 3 | Hệ thống | Hiển thị danh sách các danh mục phù hợp với từng người dùng. | | 4 | Khách hàng, quản trị viên | Kích vào nút “Xem chi tiết” của 1 danh mục. | | 5 | Hệ thống | Hiển thị chi tiết danh mục được chọn. | | 6 | Khách hàng, quản trị viên | Kích vào nút “Thêm danh mục”. | | 7 | Hệ thống | Hiển thị form nhập danh mục mới. | | 8 | Khách hàng, quản trị viên | Điền đầy đủ thông tin danh mục mới và kích vào nút “Thêm mới”. | | 9 | Hệ thống | Kiểm tra thông tin. | | 10 | Hệ thống | Thêm công thức mới vào hệ thống. | | 11 | Hệ thống | Hiển thị thông báo thành công và quay về màn hình danh sách danh mục. | | 12 | Khách hàng, quản trị viên | Kích vào nút “Sửa” ở cột cuối của 1 dòng danh mục bất kỳ. | | 13 | Hệ thống | Hiển thị form sửa công thức với thông tin danh mục vừa được chọn. | | 14 | Khách hàng, quản trị viên | Nhập thông tin cần sửa vào form và kích vào nút “Sửa”. | | 15 | Hệ thống | Kiểm tra thông tin. | | 16 | Hệ thống | Lưu danh mục vào hệ thống. | | 17 | Hệ thống | Hiển thị thông báo lưu thành công và quay lại màn hình danh sách danh mục. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 8a, 14a | Khách hàng, quản trị viên | Nếu khách hàng hoặc quản trị viên kích vào nút “Hủy”, hệ thống sẽ dừng use case và quay lại màn hình danh sách danh mục. | | 9a, 15a | Hệ thống | Nếu thông tin nhập vào không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Cập nhật thông tin của danh mục vào hệ thống. | | |

### Use case Quản lý nguyên liệu

Chức năng “Quản lý nguyên liệu” 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:

Hình 2.10 Biểu đồ use case Quản lý nguyên liệu

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

Hình 2.11 Biểu đồ hoạt động use case Quản lý nguyên liệu

Chức năng “Quản lý nguyên liệu” 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 2.6 Đặc tả chi tiết use case Quản lý nguyên liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC05 | **Tên Use case** | Quản lý nguyên liệu |
| **Tác nhân** | Khách hàng, Quản trị viên | | |
| **Mô tả** | Use case này cho phép khách hàng quản lý các nguyên liệu của riêng mình và quản trị viên quản lý tất cả các nguyên liệu trong hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng hoặc quản trị viên kích vào Avatar ở góc trên bên phải đối với người dùng và kích vào nút “Admin Home” ở góc trên bên phải, sau đó kích vào nút “Quản lý nguyên liệu”. | | |
| **Tiền điều kiện** | Khách hàng cũng như quản trị viên phải đăng nhập vào hệ thống với tài khoản đã được phân quyền. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1a | Khách hàng | Kích vào nút Avatar ở góc trên bên phải. | | 1b | Quản trị viên | Kích vào nút “Admin Home” ở góc trên bên phải. | | 2 | Khách hàng, quản trị viên | Kích vào nút “Quản lý nguyên liệu”. | | 3 | Hệ thống | Hiển thị danh sách các nguyên liệu phù hợp với từng người dùng. | | 4 | Khách hàng, quản trị viên | Kích vào nút “Xem chi tiết” của 1 nguyên liệu. | | 5 | Hệ thống | Hiển thị chi tiết nguyên liệu được chọn. | | 6 | Khách hàng, quản trị viên | Kích vào nút “Thêm nguyên liệu”. | | 7 | Hệ thống | Hiển thị form nhập nguyên liệu mới. | | 8 | Khách hàng, quản trị viên | Điền đầy đủ thông tin nguyên liệu mới và kích vào nút “Thêm mới”. | | 9 | Hệ thống | Kiểm tra thông tin. | | 10 | Hệ thống | Thêm nguyên liệu mới vào hệ thống. | | 11 | Hệ thống | Hiển thị thông báo thành công và quay về màn hình danh sách nguyên liệu. | | 12 | Khách hàng, quản trị viên | Kích vào nút “Sửa” ở cột cuối của 1 dòng nguyên liệu bất kỳ. | | 13 | Hệ thống | Hiển thị form sửa công thức với thông tin nguyên liệu vừa được chọn. | | 14 | Khách hàng, quản trị viên | Nhập thông tin cần sửa vào form và kích vào nút “Sửa”. | | 15 | Hệ thống | Kiểm tra thông tin. | | 16 | Hệ thống | Lưu danh mục vào hệ thống. | | 17 | Hệ thống | Hiển thị thông báo lưu thành công và quay lại màn hình danh sách nguyên liệu. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 8a, 14a | Khách hàng, quản trị viên | Nếu khách hàng hoặc quản trị viên kích vào nút “Hủy”, hệ thống sẽ dừng use case và quay lại màn hình danh sách nguyên liệu. | | 9a, 15a | Hệ thống | Nếu thông tin nhập vào không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Cập nhật thông tin của nguyên liệu vào hệ thống. | | |

### Use case Quản lý đơn vị tính

Chức năng “Quản lý đơn vị tính” 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:

Hình 2.12 Biểu đồ use case Quản lý đơn vị tính

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

Hình 2.13 Biểu đồ hoạt động use case Quản lý đơn vị tính

Chức năng “Quản lý đơn vị tính” 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC06 | **Tên Use case** | Quản lý đơn vị tính |
| **Tác nhân** | Khách hàng, Quản trị viên | | |
| **Mô tả** | Use case này cho phép khách hàng quản lý các đơn vị tính của riêng mình và quản trị viên quản lý tất cả các đơn vị tính trong hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng hoặc quản trị viên kích vào Avatar ở góc trên bên phải đối với người dùng và kích vào nút “Admin Home” ở góc trên bên phải, sau đó kích vào nút “Quản lý đơn vị tính”. | | |
| **Tiền điều kiện** | Khách hàng cũng như quản trị viên phải đăng nhập vào hệ thống với tài khoản đã được phân quyền. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1a | Khách hàng | Kích vào nút Avatar ở góc trên bên phải. | | 1b | Quản trị viên | Kích vào nút “Admin Home” ở góc trên bên phải. | | 2 | Khách hàng, quản trị viên | Kích vào nút “Quản lý đơn vị tính”. | | 3 | Hệ thống | Hiển thị danh sách các đơn vị tính phù hợp với từng người dùng. | | 4 | Khách hàng, quản trị viên | Kích vào nút “Xem chi tiết” của 1 nguyên liệu tính. | | 5 | Hệ thống | Hiển thị chi tiết đơn vị tính được chọn. | | 6 | Khách hàng, quản trị viên | Kích vào nút “Thêm đơn vị tính”. | | 7 | Hệ thống | Hiển thị form nhập đơn vị tính mới. | | 8 | Khách hàng, quản trị viên | Điền đầy đủ thông tin đơn vị tính mới và kích vào nút “Thêm mới”. | | 9 | Hệ thống | Kiểm tra thông tin. | | 10 | Hệ thống | Thêm đơn vị tính mới vào hệ thống. | | 11 | Hệ thống | Hiển thị thông báo thành công và quay về màn hình danh sách nguyên liệu. | | 12 | Khách hàng, quản trị viên | Kích vào nút “Sửa” ở cột cuối của 1 dòng đơn vị tính bất kỳ. | | 13 | Hệ thống | Hiển thị form sửa công thức với thông tin đơn vị tính vừa được chọn. | | 14 | Khách hàng, quản trị viên | Nhập thông tin cần sửa vào form và kích vào nút “Sửa”. | | 15 | Hệ thống | Kiểm tra thông tin. | | 16 | Hệ thống | Lưu danh mục vào hệ thống. | | 17 | Hệ thống | Hiển thị thông báo lưu thành công và quay lại màn hình danh sách đơn vị tính. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 8a, 14a | Khách hàng, quản trị viên | Nếu khách hàng hoặc quản trị viên kích vào nút “Hủy”, hệ thống sẽ dừng use case và quay lại màn hình danh sách đơn vị tính. | | 9a, 15a | Hệ thống | Nếu thông tin nhập vào không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Cập nhật thông tin của đơn vị tính vào hệ thống. | | |

### Use case Chat với ChatbotAI

Chức năng “Chat với ChatbotAI” 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:

Hình 2.20 Biểu đồ use case Chat với ChatbotAI

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

Hình 2.21 Biểu đồ hoạt động use case Chat với ChatbotAI

Chức năng “Chat với ChatbotAI” 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 2.11 Đặc tả chi tiết use case "Chat với ChatbotAI"

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC07 | **Tên Use case** | Chat với ChatbotAI |
| **Tác nhân** | Khách hàng, Quản trị viên | | |
| **Mô tả** | Use case này cho phép người dùng nói chuyện, đặt các câu hỏi thắc mắc với AI của hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi người dùng kích vào nút “Chat với Chatbot” ở góc dưới bên phải các màn hình. | | |
| **Tiền điều kiện** | Người dùng phải đăng nhập vào hệ thống để chat được với Chatbot. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1 | Người dùng | Kích vào nút “Chat với Chatbot” ở góc dưới bên phải của các màn hình. | | 2 | Hệ thống | Hiển thị giao diện nhỏ để chat với Chatbot góc dưới bên phải. | | 3 | Người dùng | Nhập câu hỏi, từ khóa và kích vào nút “Gửi”. | | 4 | Hệ thống | Xử lý dữ liệu và trả về câu trả lời phù hợp với dữ liệu đầu vào của người dùng. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Không có. | | |

### Use case Bình luận và đánh giá

Chức năng “Bình luận và đánh giá” 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:

Hình 2.14 Biểu đồ use case Bình luận và đánh giá

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

Hình 2.15 Biểu đồ hoạt động use case Bình luận và đánh giá

Chức năng “Bình luận và đánh giá” 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 2.8 Đặc tả chi tiết use case " Bình luận và đánh giá "

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC08 | **Tên Use case** | Bình luận và đánh giá |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Use case này cho phép khách hàng bình luận và đánh giá các công thức. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng nhập bình luận vào ô bình luận phía dưới của công thức bất kỳ. | | |
| **Tiền điều kiện** | Khách hàng phải đăng nhập tài khoản trước khi bình luận. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Nhập bình luận vào ô bình luận ở dưới công thức. | | 2 | Khách hàng | Chọn số sao muốn đánh giá cho công thức. | | 3 | Khách hàng | Kích vào nút “Gửi”. | | 4 | Hệ thống | Kiểm tra thông tin. | | 5 | Hệ thống | Lưu bình luận vào hệ thống. | | 6 | Hệ thống | Hiển thị lại danh sách bình luận của công thức. | | 7 | Khách hàng | Kích vào nút “Xóa” trên bình luận của khách hàng đó. | | 8 | Hệ thống | Hiển thị thông báo xác nhận xóa bình luận. | | 9 | Khách hàng | Kích vào nút “Xóa”. | | 10 | Hệ thống | Cập nhật lại thông tin của bình luận. | | 11 | Hệ thống | Hiển thị lại danh sách bình luận của công thức. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 4a | Hệ thống | Kiểm tra nếu thông tin người dùng không hợp lệ thì yêu cầu nhập lại. | | 4b | Hệ thống | Nếu khách hàng đã bình luận thì không cho phép bình luận thêm lần nữa để tránh trường hợp người dùng spam. | | 3a, 9a | Khách hàng | Nếu khách hàng kích vào nút “Hủy”, hệ thống sẽ dừng use case. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Cập nhật thông tin bình luận vào hệ thống. | | |

### Use case Quản lý thông tin cá nhân

Chức năng “Quản lý thông tin cá nhân” 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:

Hình 2.16 Biểu đồ use case Quản lý thông tin cá nhân

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

Hình 2.17 Biểu đồ hoạt động use case Quản lý thông tin cá nhân

Chức năng “Quản lý thông tin cá nhân” 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 2.9 Đặc tả chi tiết use case " Quản lý thông tin cá nhân"

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC09 | **Tên Use case** | Quản lý thông tin cá nhân |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Use case này cho phép khách hàng quản lý các thông tin cá nhân trong hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi khách hàng kích vào nút “Quản lý tài khoản cá nhân” ở góc trên bên phải màn hình. | | |
| **Tiền điều kiện** | Khách hàng phải đăng nhập tài khoản trước đó. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Kích vào nút Avatar ở góc trên bên phải, sau đó kích vào nút “Quản lý tài khoản cá nhân” ở menu hiển thị ra. | | 2 | Hệ thống | Hiển thị màn hình thông tin tài khoản cá nhân. | | 3 | Khách hàng | Nhập thông tin muốn sửa vào form rồi kích vào nút “Cập nhật”. | | 4 | Hệ thống | Kiểm tra thông tin. | | 5 | Hệ thống | Lưu thông tin vào hệ thống. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 4a | Hệ thống | Nếu người dùng nhập sai hoặc thiếu, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Cập nhật thông tin người dùng vào hệ thống. | | |

### Use case Quản lý người dùng

Chức năng “Quản lý người dùng” 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:

Hình 2.18 Biểu đồ use case Quản lý người dùng

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

Hình 2.19 Biểu đồ hoạt động use case Quản lý người dùng

Chức năng “Quản lý người dùng” 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 2.10 Đặc tả chi tiết use case " Quản lý người dùng "

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC10 | **Tên Use case** | Quản lý người dùng |
| **Tác nhân** | Quản trị viên | | |
| **Mô tả** | Use case này cho phép quản trị viên quản lý các tài khoản trong hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi quản trị viên kích vào nút “Người dùng” trên thanh điều hướng bên trái ở trang chủ của quản trị viên. | | |
| **Tiền điều kiện** | Quản trị viên phải đăng nhập với tài khoản đã được phân quyền quản trị viên. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1 | Quản trị viên | Kích vào nút “Người dùng” ở thanh bên trái trên các màn hình của quản trị viên. | | 2 | Hệ thống | Hiển thị màn hình danh sách người dùng trong hệ thống. | | 3 | Quản trị viên | Kích vào nút “Xem chi tiết” của 1 tài khoản bất kỳ. | | 4 | Hệ thống | Hiển thị thông tin chi tiết của tài khoản đó. | | 5 | Quản trị viên | Kích vào nút “Sửa” của 1 tài khoản bất kỳ. | | 6 | Hệ thống | Hiển thị form sửa thông tin tài khoản. | | 7 | Quản trị viên | Điền các thông tin cần sửa của tài khoản và kích vào nút “Cập nhật”. | | 8 | Hệ thống | Kiểm tra thông tin. | | 9 | Hệ thống | Lưu thông tin tài khoản vào hệ thống. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 7a | Quản trị viên | Nếu quản trị viên kích vào nút “Hủy”, hệ thống quay lại màn hình danh sách người dùng hệ thống. | | 8a | Hệ thống | Nếu thông tin nhập vào không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại. | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Cập nhật thông tin tài khoản vào hệ thống. | | |

### Use case Thống kê

Chức năng “Thống kê” 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:

Hình 2.20 Biểu đồ use case Thống kê

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

Hình 2.21 Biểu đồ hoạt động use case Thống kê

Chức năng “Thống kê” 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 2.11 Đặc tả chi tiết use case " Thống kê"

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case** | UC11 | **Tên Use case** | Thống kê |
| **Tác nhân** | Quản trị viên | | |
| **Mô tả** | Use case này cho phép quản trị viên xem được các thống kê về hệ thống. | | |
| **Sự kiện kích hoạt chức năng** | Use case này bắt đầu khi quản trị viên kích vào nút “Dashboard” trên thanh điều hướng bên trái ở trang chủ của quản trị viên. | | |
| **Tiền điều kiện** | Quản trị viên phải đăng nhập với tài khoản đã được phân quyền quản trị viên. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | 1 | Quản trị viên | Kích vào nút “Dashboard” ở thanh bên trái trên các màn hình của quản trị viên. | | 2 | Hệ thống | Hiển thị các thống kê liên quan đến hệ thống với các biểu đồ mặc định. | | 3 | Quản trị viên | Chọn mốc thời gian, loại biều đồ. | | 4 | Hệ thống | Hiển thị các thống kê phù hợp với thời gian và biểu đồ đã chọn. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **#** | **Thực hiện bởi** | **Hành động** | | Tại bất kỳ bước nào | Hệ thống | Nếu có lỗi xảy ra, hiển thị thông báo “Đã có lỗi xảy ra”. | | | |
| **Hậu điều kiện** | Không có | | |

## 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

Thiết kế kiến trúc phần mềm là bước quan trọng trong quá trình phát triển hệ thống website mạng xã hội chia sẻ công thức nấu ăn Cook-craft. Kiến trúc hệ thống sẽ quyết định cách các thành phần trong hệ thống tương tác với nhau, ảnh hưởng đến hiệu suất, bảo mật và khả năng mở rộng của hệ thống.

Hệ thống được xây dựng theo mô hình Client-Server hiện đại, bao gồm ba lớp chính:

### Frontend (Client side)

* Sử dụng ReactJS để phát triển giao diện người dùng (UI).
* Hỗ trợ tương tác trực tiếp với người dùng thông qua component và state management (React hooks, Context API, Redux).
* Thực hiện các yêu cầu HTTP tới backend thông qua RESTful API (sử dụng Axios/Fetch).
* Xây dựng giao diện với HTML, CSS, JavaScript (ES6+) kết hợp thư viện UI (Material UI, TailwindCSS, hoặc tương tự) để tạo trải nghiệm trực quan, thân thiện và responsive.
* Quản lý routing thông qua React Router.

### Backend (Server side)

* Sử dụng Spring Boot để phát triển API và xử lý logic nghiệp vụ.
* Cung cấp các REST API cho frontend sử dụng.
* Quản lý xác thực và phân quyền người dùng qua Spring Security.
* Sử dụng mô hình MVC (Model-View-Controller) để phân chia rõ ràng giữa logic nghiệp vụ, tầng dữ liệu và giao diện.
* Tích hợp với MySQL để lưu trữ và quản lý dữ liệu.

### Database layer

* Sử dụng MySQL làm hệ quản trị cơ sở dữ liệu chính.
* Lưu trữ thông tin công thức, khách hàng, bình luận, đánh giá và các dữ liệu quan trọng khác.
* Hỗ trợ các tính năng như backup, restore và transaction đảm bảo tính nhất quán của dữ liệu.

### Mô hình tổng thể hệ thống

Hệ thống bao gồm các thành phần chính sau:

* Client (ReactJS).
* Giao diện người dùng (UI) cho khách hàng và quản trị viên.
* Xử lý các yêu cầu HTTP và hiển thị dữ liệu từ backend.

#### API Gateway (Spring Boot):

* Xử lý tất cả các yêu cầu từ client và điều phối tới các service phù hợp.
* Đảm bảo an toàn dữ liệu qua cơ chế xác thực và phân quyền.

#### Database (MySQL):

* Lưu trữ thông tin sách, danh mục, khách hàng, đơn hàng, và lịch sử giao dịch.
* Sử dụng JPA/Hibernate để tương tác với cơ sở dữ liệu một cách tự động và hiệu quả.

### Các thành phần kiến trúc chính

#### User Interface (UI):

* Phát triển bằng ReactJS với thiết kế responsive.
* Sử dụng các component tái sử dụng và module hóa để tăng tính linh hoạt.

#### Service Layer (Backend):

* Xử lý logic nghiệp vụ như quản lý công thức, bình luận, xử lý train.
* Đảm bảo an toàn dữ liệu với các cơ chế xác thực và phân quyền.

#### Data Layer (Database):

* Quản lý dữ liệu thông qua các entity và repository trong Spring Boot.
* Sử dụng các transaction để đảm bảo tính nhất quán của dữ liệu.

#### Security Layer:

* Bảo vệ hệ thống với cơ chế xác thực (Authentication) và phân quyền (Authorization).
* Mã hóa thông tin người dùng và bảo mật truyền tải dữ liệu qua HTTPS.

### Lưu đồ kiến trúc hệ thống

Lưu đồ mô tả tổng quan kiến trúc hệ thống Cook-craft:

* Người dùng truy cập website từ trình duyệt trên máy tính hoặc thiết bị di động.
* Yêu cầu của người dùng được gửi tới API Gateway (Spring Boot) thông qua các HTTP request.
* Backend xử lý logic nghiệp vụ, bao gồm: quản lý công thức, bình luận, lượt thích, người dùng, và tương tác với AI Recommendation/Chatbot.
* Backend tương tác với cơ sở dữ liệu MySQL để lưu trữ và truy xuất dữ liệu một cách hiệu quả.
* Kết quả được trả về cho frontend (ReactJS) để hiển thị giao diện thân thiện và tương tác trực quan cho người dùng.
* Hệ thống được thiết kế mở rộng, dễ bảo trì, tối ưu hiệu suất và hỗ trợ phát triển các tính năng mới mà không ảnh hưởng đến các phần đã triển khai.

## Thiết kế cấu trúc phần mềm

## Thiết kế cơ sở dữ liệu

Để đảm bảo hệ thống hoạt động ổn định, truy xuất dữ liệu hiệu quả và hỗ trợ đầy đủ các chức năng đã đề ra, cơ sở dữ liệu được thiết kế với các bảng liên kết chặt chẽ, đảm bảo tính toàn vẹn và mở rộng. Cấu trúc cơ sở dữ liệu bao gồm các bảng chính như: người dùng, danh mục, công thức, nguyên liệu, đơn vị tính, bình luận đánh giá.

### Biểu đồ mô tả cấu trúc dữ liệu

### Mô tả cơ sở dữ liệu

Bảng Users

Bảng Categoris

Bảng Recipes

Bảng Ingredients

Bảng Units

Bảng Recipe\_Ingredient\_Details

Bảng Reviews

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Thiết kế giao diện

Giao diện người dùng được thiết kế theo hướng hiện đại, trực quan, dễ sử dụng, phù hợp với cả khách hàng và quản trị viên. Mỗi chức năng đều có bố cục hợp lý, đảm bảo trải nghiệm mượt mà trên cả máy tính và thiết bị di động. Các mẫu giao diện sau đây minh họa cụ thể các chức năng chính trong hệ thống.

### Đăng nhập

### Tìm kiếm công thức

### Quản lý công thức

### Quản lý danh mục

### Quản lý nguyên liệu

### Quản lý đơn vị tính

### Chat với ChatbotAI

### Bình luận và đánh giá

### Quản lý thông tin cá nhân

### Quản lý người dùng

### Thống kê

## Kết luận

Chương 3 đã trình bày toàn bộ kiến trúc của hệ thống từ tổng thể đến chi tiết: kiến trúc client-server, cấu trúc phân lớp, thiết kế CSDL, API và giao diện người dùng. Các biểu đồ lớp và biểu đồ trình tự giúp thể hiện rõ tương tác giữa các thành phần, góp phần đảm bảo tính mạch lạc và khả năng mở rộng trong quá trình phát triển.

# CÀI ĐẶT WEBSITE DIỄN ĐÀN CHIA SẺ CÔNG THỨC NẤU ĂN COOK-CRAFT

## Môi trường phát triển

Hệ thống website bán sách Tâm An được cài đặt và phát triển trên môi trường máy tính cá nhân với cấu hình và công cụ như sau:

*Bảng 4.**1 Cấu hình máy tính cá nhân và công cụ cài đặt phát triển*

|  |  |
| --- | --- |
| Thành phần | Thông tin sử dụng |
| Hệ điều hành | Windows 11 64-bit |
| IDE backend | IntelliJ IDEA (dành cho phát triển Spring Boot) |
| IDE frontend | Visual Studio Code (dành cho phát triển Angular) |
| Phiên bản Java | Java SE 21 |
| Công cụ build backend | Maven 3.9.x |
| Framework backend | Spring Boot 3.4.3 |
| Framework frontend | Angular 19 |
| Node.js | Phiên bản 18x |
| Cơ sở dữ liệu | MySQL 8.0.41 |
| Gửi mail | SMTP Gmail |
| Trình duyệt kiểm thử | Google Chrome, Microsoft Edge |

## Cài đặt backend (Spring Boot)

## Tạo dự án Spring Boot

Dự án backend được tạo bằng Spring Initializr với các thông số:

* + - * Ngôn ngữ: Java
      * Build tool: Maven
      * Spring Boot version: 3.4.3 Dependencies chính:
      * Spring Web
      * Spring Data JPA
      * Spring Security
      * Validation
      * MySQL Driver
      * Mail
      * Devtools
      * Lombok
      * Apache POI (xuất excel)
      * Cloudinary (upload ảnh)

## Chạy ứng dụng backend

Mở class BookstoreTamAnApplication.java, chạy trực tiếp trong IntelliJ IDEA.

Hoặc dùng dòng lệnh:

* + - * Mvn clean install
      * Java -jar target/BookstoreTamAn-0.1.1-SNAPSHOT.jar Backend khởi chạy tại địa chỉ: <http://localhost:8080/api/bookstore>

## Cài đặt frontend (Angular)

## Cài đặt môi trường

* + - * Cài Node.js phiên bản 18.x
      * Cài Angular CLI:

+ npm install -g @angular/cli

## Khởi tạo và cấu hình dự án

* + - * Clone hoặc tạo mới project BookstoreTamAn
      * Cài đặt các thư viện phụ:

+ npm install

## Chạy frontend

* + - * Clone hoặc tạo mới project BookstoreTamAn
      * Mở terminal chạy lệnh: ng serve
      * Truy cập giao diện tại địa chỉ: [http://localhost:4200](http://localhost:4200/)

## Cài đặt cơ sở dữ liệu MySQL

## Tạo cơ sở dữ liệu

Mở HeidiSQL chạy câu lệnh: CREATE DATABASE BookstoreTamAn

## Cấu trúc cơ sở dữ liệu

Cấu trúc các bảng như: user, book, category, book\_category, author, order, order\_detail, news được định nghĩa trong chương 3.3.2

## Khởi tạo dữ liệu

Dữ liệu mẫu được thêm thủ công hoặc bằng file .sql thông qua HeidiSQL. Một số bảng được liên kết bằng khóa ngoại như book → author, order → user.

## Chạy thử toàn hệ thống

1. Chạy backend bằng IntelliJ hoặc terminal.
2. Chạy frontend bằng lệnh ng serve.
3. Truy cập <http://locallhost:4200/>.

## Triển khai hệ thống

* Sau khi xây dựng và kiểm thử thành công hệ thống trên môi trường cục bộ, sau đó tiến hành triển khai hệ thống website bán sách Tâm An lên Internet để người dùng có thể truy cập và sử dụng như một website thực thụ. Việc triển khai này giúp kiểm thử hệ thống trong môi trường thật, phục vụ đánh giá tổng thể và trình diễn đồ án.
* Sử dụng các dịch vụ miễn phí hỗ trợ triển khai nhanh chóng:

+ Render: triển khai backend (Spring Boot).

+ Vercel: triển khai frontend (Angular).

## Triển khai backend trên Render

**Bước 1:** Đẩy toàn bộ mã nguồn backend Spring Boot lên GitHub trong một repository riêng.

**Bước 2:** Truy cập vào https://render.com, đăng nhập bằng tài khoản GitHub, tạo một Web Service mới từ repo bookstore-backend.

**Bước 3:** Cấu hình dịch vụ:

* Environment: Java.
* Build Command: ./mvnw clean install.
* Start Command: java -jar target/\*.jar.
* Region: Singapore.

**Bước 4:** Render tự động build và khởi động ứng dụng. Sau vài phút, hệ thống cung cấp một đường dẫn API công khai.

## Triển khai frontend trên Vercel

**Bước 1:** Đẩy mã nguồn Angular của dự án TamAnStore lên một repo GitHub riêng.

**Bước 2:** Truy cập https://vercel.com, đăng nhập bằng GitHub, chọn Add New Project, liên kết tới repo frontend.

**Bước 3:** Cấu hình dự án:

* Environment: Java.
* Framework: Angular.
* Build Command: npm run build.
* Output Directory: dist/TamAnStore.

**Bước 4:** Sau khi triển khai xong, Vercel cung cấp một đường dẫn công khai.

## Kết nối frontend với backend

* + - * Trong các file service của Angular, cấu hình lại đường dẫn API backend đã triển khai.
      * Sau đó rebuild frontend và đẩy lại lên GitHub để Vercel tự động triển khai lại với cấu hình mới.

# KIỂM THỬ WEBSITE DIỄN ĐÀN CHIA SẺ CÔNG THỨC NẤU ĂN COOK-CRAFT

## Mục tiêu kiểm thử

Quá trình kiểm thử nhằm đảm bảo rằng hệ thống:

* Đáp ứng đúng yêu cầu chức năng và phi chức năng đã đặt ra.
* Hoạt động ổn định khi người dùng truy cập thực tế qua mạng Internet.
* Phát hiện và sửa lỗi trong quá trình vận hành thử nghiệm.

## Kế hoạch kiểm thử

Quá trình kiểm thử được thực hiện thủ công (manual testing) trên các trình duyệt phổ biến và qua công cụ Postman để kiểm tra các API backend. Kiểm thử được chia làm hai nhóm đối tượng: khách hàng và quản trị viên.

## Các chức năng được kiểm thử

**Phía khách hàng:**

* + - * Đăng ký tài khoản.
      * Đăng nhập.
      * Quên mật khẩu.
      * Tìm kiếm sách.
      * Quản lý giỏ hàng.
      * Đặt hàng và thanh toán.

## Phía quản trị viên:

* + - * Quản lý sản phẩm: thêm, sửa, xóa, hiển thị.
      * Thống kê doanh thu.
      * Xuất báo cáo doanh thu ra file Excel.

## Công cụ kiểm thử

* + - * Trình duyệt: Chrome, Edge.
      * Công cụ kiểm thử API: Postman.
      * Công cụ kiểm tra dữ liệu: HeidiSQL.
      * Công cụ kiểm tra mail: Gmail.

## Kiểm thử chức năng

*Bảng 5.1* *Kiểm thử các chức năng*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mục đích kiểm thử** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** |
| TC\_01 |  | Đăng ký thành công khi nhập các trường hợp lệ | Nhập họ tên, email, tên đăng nhập, mật khẩu hợp lệ, địa chỉ, số điện thoại.  Nhấn nút Đăng kí | Đăng ký thành công | Đạt |
|  | Đăng ký |  |  |  |  |
| TC\_02 | Đăng ký không thành công khi các trường để trống | Bỏ trống tất cả các trường  Nhấn nút Đăng kí | Hiển thị thông báo lỗi ở các trường bắt buộc | Đạt |
| TC\_03 |  | Đăng kí không | Nhập email đã | Hiển thị | Đạt |
|  |  | thành công khi | đăng kí | thông |  |
|  |  | nhập email đã | 2. Nhập các trường | báo lỗi |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | tồn tại | còn lại hợp  3. Nhấn nút đăng kí | “Email đã tồn tại trong hệ thống ” |  |
| TC\_04 | Đăng nhập | Đăng nhập thành công khi nhập các trường hợp lệ | 1. Nhập tên đăng nhập và mật khẩu hợp lệ 2. Nhấn nút đăng nhập | Đăng nhập thành công | Đạt |
| TC\_05 | Đăng nhập không thành công khi để trống các trường | 1. Bỏ trống tất cả các trường 2. Nhấn nút đăng nhập | Hiển thị thông báo lỗi ở các trường bắt buộc | Đạt |
| TC\_06 | Đăng nhập không thành công khi nhập sai email | 1. Nhập tên đăng nhập không hợp lệ 2. Nhập mật hợp lệ 3. Nhấn nút đăng nhập | Hiển thị thông báo lỗi “Tên đăng nhập hoặc mật khẩu không đúng” | Đạt |
| TC\_07 | Đăng nhập  không thành | 1. Nhập tên đăng   nhập hợp lệ | Hiển thị  thông | Đạt |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | công khi nhập mật khẩu sai | 1. Nhập mật khẩu sai 2. Nhấn nút đăng nhập | báo lỗi “Tên đăng nhập hoặc mật khẩu không đúng” |  |
| TC\_08 | Tìm kiếm sách | Tìm kiếm sản phẩm khi nhập đúng toàn bộ tên sách | 1. Nhập đúng toàn bộ tên sách 2. Nhấn Enter | Hiển thị sản phẩm vừa nhập tương ứng | Đạt |
| TC\_09 | Tìm kiếm sản phẩm khi nhập đúng một phần tên sách | 1. Nhập đúng một phần tên sách 2. Nhấn Enter | Hiển thị danh sách sản phẩm trong tên có chứa các kí tự vừa nhập tương ứng | Đạt |
| TC\_10 | Tìm kiếm sản phẩm khi không nhập gì | 1. Không nhập gì 2. Nhấn Enter | Hiển thị danh sách sản  phẩm | Đạt |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | ban đầu |  |
| TC\_11 | Tìm kiếm sản phẩm khi nhập sản phẩm không có trong hệ thống | 1. Nhập tên sản phẩm không có trong hệ thống 2. Nhấn Enter | Hiển thị danh sách trống |  |
| TC12 | Tìm kiếm khi nhập khoảng trắng đầu, cuối tên sách | 1. Nhập khoảng trắng ở đầu hoặc cuối tên sách 2. Nhấn Enter | Tìm kiếm thành công, cắt bỏ khoảng trắng | Không đạt |
| TC\_13 | Thêm sản phẩm vào giỏ hàng | Thêm sản phẩm vào giỏ hàng | 1. Chọn sản phẩm muốn thêm vào giỏ hàng 2. Click button “Thêm vào giỏ” | Màn hình hiển thị thông báo “Thêm sản phẩm vào giỏ hàng thành công” | Đạt |
| TC\_14 | Cập nhật số lượng | Cập nhật số lượng giỏ hàng thành công khi tổng | 1. Click button “+”, “-“, hoặc nhập số lượng | Số lượng được cập nhật đúng | Đạt |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | giỏ hàng | số lượng trong giỏ hàng <100 |  |  |  |
| TC\_15 | Xóa sản phẩm khỏi giỏ hàng | Xóa sản phẩm thành công | 1. Click button “Xóa” , hiển thị popup xác nhận xóa 2. Click button “Có” | Xóa sản phẩm thành công | Đạt |
| TC\_16 | Xóa sản phẩm không thành công | 1. Click button “Xóa” , hiển thị popup xác nhận xóa 2. Click button “Hủy” | Hủy bỏ thao tác xóa | Đạt |
| TC\_17 | Đặt hàng | Đặt hàng thành công khi nhập thông tin hợp lệ | 1. Nhập đúng thông tin cần thanh toán 2. Nhấn nút “Xác nhận đặt hàng” | Thanh toán thành công, gửi email thông báo cho người dùng.” | Đạt |
| TC\_18 | Đặt hàng không thành công khi bỏ  trống tất cả | 1. Bỏ trống tất cả các trường 2. Nhấn nút “Xác   nhận đặt hàng” | Hiển thị thông báo lỗi ở  các | Đạt |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | các trường |  | trường bắt buộc |  |
| TC\_19 | Quản lý sản phẩm | Thêm sản phẩm thành công khi nhập các trường hợp lệ | 1. Nhập các trường hợp lệ 2. Nhấn nút Lưu | Thêm sản phẩm thành công | Đạt |
| TC\_20 | Thêm sản phẩm thành công khi bỏ trống các trường | 1. Bỏ trống tất cả các trường 2. Nhấn nút Lưu | Hiển thị thông báo lỗi ở các trường bắt buộc | Đạt |
| TC\_21 | Sửa sản phẩm thành công khi nhập các trường hợp lệ | 1. Bỏ trống tất cả các trường 2. Nhấn nút Lưu | Cập nhật sản phẩm thành công | Đạt |
| TC\_22 | Sửa sản phẩm thành công khi bỏ trống các trường | 1. Bỏ trống tất cả các trường 2. Nhấn nút Lưu | Hiển thị thông báo lỗi ở các trường bắt buộc | Đạt |
| TC\_23 | Xóa sản phẩm thành công | 1. Chọn sản phẩm cần xóa 2. Nhấn nút Xóa,   hiển thị popup xác | Xóa sản phẩm thành công, sản | Đạt |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | nhận xóa  3. Nhấn nút “Có” | phẩm được xóa khỏi hệ thống |  |
| TC\_24 | Xóa sản phẩm không thành công | 1. Chọn sản phẩm cần xóa 2. Nhấn nút Xóa, hiển thị popup xác nhận xóa 3. Nhấn nút “Hủy” | Hủy bỏ thao tác xóa | Đạt |
| TC\_25 | Thống kê doanh thu | Thống kê doanh thu thành công | 1. Chọn ngày bắt đầu và ngày kết thúc cần thống kê 2. Nhấn nút “Xem thống kê” | Phản ánh đúng kết quả thống kê | Đạt |
| TC\_26 | Xuất excel thống kê | Xuất excel thành công | 1. Nhấn nút “Xuất excel” | File excel được tải về máy và hiển thị đúng nội dung thống kê | Đạt |

## Kết quả kiểm thử

Phía khách hàng (Tổng cộng 18 test case):

* Tỷ lệ đạt: 94,5%
* Tỷ lệ không đạt: 5,5%

Phía quản trị viên (Tổng cộng 8 test case):

* Tỷ lệ đạt: 100%
* Tỷ lệ không đạt: 0,0%

## Kết luận chương 5

Hệ thống đã được kiểm thử thông qua các test case cụ thể bao gồm cả phía người dùng và quản trị viên. Tất cả chức năng đều hoạt động đúng yêu cầu, đảm bảo tính ổn định, bảo mật và hiệu quả xử lý. Việc kiểm thử trên môi trường triển khai thực tế càng làm rõ tính ứng dụng cao của hệ thống.

# KẾT LUẬN

## Kết quả đạt được

* + - * Qua quá trình thực hiện đồ án, hệ thống website bán sách Tâm An đã hoàn thiện đầy đủ các chức năng cơ bản và nâng cao, phục vụ tốt cho cả người dùng cuối và người quản trị.
      * Tích lũy được nhiều kinh nghiệm cũng như kiến thức mới:
  + Rèn luyện kỹ năng phân tích yêu cầu, thiết kế hệ thống, và xử lý các tình huống thực tế phát sinh trong quá trình xây dựng và triển khai.
  + Hiểu sâu hơn về quy trình phát triển một hệ thống web hoàn chỉnh từ frontend đến backend.
  + Nắm vững cách sử dụng các công nghệ như Angular, Spring Boot, MySQL, cũng như tích hợp các dịch vụ như email, thanh toán online, triển khai đám mây.
  + Quan trọng hơn, học được cách quản lý thời gian, làm việc tự lập, cũng như không ngừng tìm hiểu tài liệu để giải quyết các vấn đề kỹ thuật phức tạp.

## Hướng phát triển

* + - * Tích hợp các hình thức đăng nhập qua bên thứ ba (Google, Facebook).
      * Hoàn thiện hệ thống đánh giá sách và gợi ý theo hành vi người dùng.
      * Tối ưu hiệu năng và giao diện trên thiết bị di động.
      * Tích hợp các API vận chuyển để theo dõi đơn hàng.

# TÀI LIỆU THAM KHẢO

1. Đỗ Ngọc Sơn, Phạm Văn Biên, Nguyễn Phương Nga (2015), *“Hệ quản trị cơ sở dữ liệu”.* Hà Nội: NXB Khoa học và Kỹ thuật – Trường Đại học Công nghiệp Hà Nội.
2. Nguyễn Trung Phú, Trần Thị Phương Nhung, Đỗ Thị Minh Nguyệt (2019), *“Thiết kế Web”*. Hà Nội: NXB Thống kê – Trường Đại học Công nghiệp Hà Nội.
3. Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan (2015) *“Phân tích thiết kế hướng đối tượng”*. Hà Nội: NXB Khoa học và Kỹ thuật – Trường Đại học Công nghiệp Hà Nội.
4. Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thuỷ, Phạm Kim Phượng (2011), *“Phân tích thiết kế hệ thống”*. Hà Nội: NXB Giáo dục Việt Nam.
5. Nguyễn Bá Nghiễn, Ngô Văn Bình, Vương Quốc Dũng, Đỗ Sinh Trường (2020), *“Lập trình hướng đối tượng với Java”*. Hà Nội: NXB Thống kê.
6. Hoàng Quang Huy (2016), *“Giáo trình kiểm thử phần mềm”*. Hà Nội: NXB Thống kê.