|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

**TRƯỜNG ĐẠI HỌC THỦY LỢI**



HOÀNG MINH HÙNG

**Xây dựng website và tích hợp Chatbot giới thiệu dịch vụ ẩm thực Hà Nội**

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI

HOÀNG MINH HÙNG

**Xây dựng website và tích hợp Chatbot giới thiệu dịch vụ ẩm thực Hà Nội**

Ngành : Công Nghệ Thông Tin

Mã số: 7480201

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | 1. TS. Nguyễn Văn Thẩm |

HÀ NỘI, NĂM 2024



**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập - Tự do - Hạnh phúc**



**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Hoàng Minh Hùng Hệ đào tạo : Đại học chính quy

Lớp: 62TH4 Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1. TÊN ĐỀ TÀI: Xây dựng website và tích hợp Chatbot giới thiệu về dịch vụ ẩm thực Hà Nội
2. CÁC TÀI LIỆU CƠ BẢN:
3. *React*. (n.d.).<https://react.dev/>
4. *Index | Node.js v21.7.3 Documentation*. (n.d.). <https://nodejs.org/docs/latest/api/>
5. *Frameworks built on Express*. (n.d.). https://expressjs.com/en/resources/frameworks.html
6. *MongoDB documentation*. (n.d.). MongoDB Documentation.https://www.mongodb.com/docs/.
7. <https://sakos.vn/10-mon-an-dac-san-ha-noi-dam-da-huong-vi-thu-do/>
8. <https://www.traveloka.com/vi-vn/explore/culinary/dac-san-ha-noi/159784/>
9. *OpenAI Platform*. (n.d.). https://platform.openai.com/docs/introduction

3. NỘI DUNG CÁC PHẦN THUYẾT MINH CẦN TÍNH TOÁN:

|  |
| --- |
| **Nội dung cần thuyết minh** |
| * Chương 1: Cơ sở lý thuyết |
| * Chương 2: Thiết kế và xây dựng hệ thống |

4. NỘI DUNG CÁC PHẦN THUYẾT MINH CẦN TÍNH TOÁN:

|  |  |
| --- | --- |
| **Phần** | **Họ và tên giáo viên hướng dẫn** |
| * Chương 1: Cơ sở lý thuyết | TS. Nguyễn Văn Thẩm |
| * Chương 2: Thiết kế và xây dựng hệ thống | TS. Nguyễn Văn Thẩm |
| * Chương 3: Kết quả và đánh giá kết luận | TS. Nguyễn Văn Thẩm |

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày ......... tháng.......năm 2024

**Trưởng Bộ môn Giáo viên hướng dẫn chính**

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . tháng. năm 2024

Chủ tịch Hội đồng

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi

Ngày. . . tháng. .....năm 2024

**Sinh viên làm Đồ án tốt nghiệp**

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

**Tên đề tài:** *Xây dựng website và tích hợp Chatbot giới thiệu về dịch vụ ẩm thực Hà nội.*

*Sinh viên thực hiện*: Hoàng Minh Hùng

*Lớp*: 62TH4

*Mã sinh viên*: 2051063705

*Số điện thoại*: 0374341002

*Email*: 2051063705@e.tlu.edu.vn

*Giáo viên hướng dẫn*: TS. Nguyễn Văn Thẩm

**TÓM TẮT ĐỀ TÀI**

1. **Bài toán cần giải quyết**

Hà Nội, thủ đô nghìn năm văn hiến, không chỉ là trung tâm văn hóa và lịch sử của Việt Nam mà còn là thiên đường ẩm thực đầy mê hoặc. Những con phố nhỏ xinh, những quán hàng truyền thống hay những nhà hàng hiện đại, tất cả đều mang trong mình hương vị đặc trưng của thành phố này. Không chỉ dừng lại ở những món ăn truyền thống, ẩm thực Hà Nội còn mang trong mình sự đa dạng và sáng tạo, với nhiều món ăn hiện đại và phong phú khác, từ bánh mì pate cho đến nộm bò khô, làm hài lòng cả những thực khách khó tính nhất. Trải qua hàng ngàn năm lịch sử và văn hóa, ẩm thực Hà Nội không chỉ là nơi để thưởng thức món ăn ngon mà còn là một phần không thể tách rời trong việc hiểu về con người và đất nước Việt Nam. Chính vì những lý do đó, em đã chọn đề tài **“Xây dựng website và tích hợp Chatbot giới thiệu dịch vụ ẩm thực của Hà Nội**

**CÁC MỤC TIÊU CHÍNH**

* Tích hợp AI Chatbot để hỗ trợ du khách
* Các báo cáo, tài liệu đặc tả theo yêu cầu.
* Thiết kế, xây dựng hệ thống đảm bảo các chức năng cơ bản như:
  + Đăng nhập, đăng kí và lưu trữ tài khoản.
  + Quản lý thông tin: Cho phép thêm, chỉnh sửa và xóa thông tin về ẩm thực Hà Nội.
  + Quản lý thông tin người dùng: Giám sát chi tiết về người dùng, cấp phép quyền truy nhập và theo dõi các hoạt động của họ trên trang web.
  + Quản lý chatbot: Cấu hình và quản lý các câu hỏi, câu trả lời và kịch bản tương tác của chatbot.
* Các tệp dữ liệu, giao diện sử dụng
* Thiết kế, xây dựng và cài đặt được website hỗ trợ giới thiệu ẩm thực Hà Nội.
* Website đảm bảo chức năng cơ bản cho người quản trị website và người dùng thông thường.
  + Dành cho người quản trị bao gồm các trang cho phép:
    - Được thêm thông tin về ẩm thực Hà Nội
    - Được chỉnh sửa các thông tin khi có sự thay đổi
    - Được xóa bài viết
    - Được xóa tài khoản các thành viên: Khi người dùng vi phạm các điều luật không gian mạng hoặc comment gây ảnh hưởng đến các người dùng khác hoặc vì một lý do nào đó yêu cầu xóa tài khoản, người quản lý có thể xóa toàn bộ thông tin liên quan đến tài khoản đó
  + Dành cho người dùng thông thường : Đăng kí , đăng nhập tài khoản , quản lý tài khoản của mình , sử dụng chatbot để tìm kiếm thông tin về ẩm thực Hà Nội.

**KẾT QUẢ DỰ KIẾN**

1. Lý thuyết:

* Hiểu biết sâu sắc về quy trình phát triển phần mềm và kiến thức vững về các công nghệ được sử dụng trong dự án, bao gồm MongoDB, React, ExpressJS và NodeJS.
* Thành thạo trong việc triển khai các nguyên lý thiết kế web và các kỹ thuật lập trình để tạo ra các trang web đáng tin cậy, hiệu quả và dễ bảo trì.
* Hoàn thiện bản đặc tả, phân tích yêu cầu thực tiễn
* Hoàn thiện báo cáo đồ án tốt nghiệp

1. Thực hiện:

* Xây dựng và triển khai trang web hoàn thiện và hoạt động một cách mượt mà và hiệu quả trên mọi thiết bị và mọi môi trường truy cập và tạo ra một trải nghiệm tương tác và thú vị cho người dùng.
* Tích hợp với AI Chatbot để để hỗ trợ du khách trong việc tìm hiểu về dịch vụ ẩm thực của thủ đô Hà Nội.

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào.Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN/KLTN**  **Hoàng Minh Hùng** |

LỜI CÁM ƠN

Trước tiên, em xin bày tỏ lòng biết ơn sâu sắc đến Ban Giám hiệu trường Đại học Thủy Lợi, các thầy cô khoa Công nghệ Thông tin đã tạo điều kiện thuận lợi và hỗ trợ em trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp này.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến TS Nguyễn Văn Thẩm, người đã tận tình hướng dẫn, động viên và cung cấp những kiến thức quý báu, giúp em hoàn thành đồ án tốt nghiệp. Sự chỉ dẫn tận tâm của thầy không chỉ giúp em giải quyết những khó khăn trong quá trình nghiên cứu mà còn định hướng cho em những phương pháp học tập và làm việc hiệu quả.

Em cũng xin gửi lời cảm ơn đến gia đình và bạn bè, những người đã luôn ở bên, ủng hộ và khích lệ em trong suốt thời gian học tập và thực hiện đồ án.

Cuối cùng, em xin cảm ơn tất cả những ai đã giúp đỡ, đóng góp ý kiến và chia sẻ kinh nghiệm để em có thể hoàn thành đồ án này. Mặc dù đã cố gắng hết sức, nhưng chắc chắn đồ án còn nhiều thiếu sót, em rất mong nhận được sự góp ý của các thầy cô và các bạn để hoàn thiện hơn.

Em xin chân thành cảm ơn!

Mục lục

[**DANH MỤC HÌNH ẢNH** 10](#_Toc185531107)

[**DANH MỤC BẢNG BIỂU** 11](#_Toc185531108)

[**DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ** 13](#_Toc185531109)

[**LỜI MỞ ĐẦU** 14](#_Toc185531110)

[**CHƯƠNG 1: Tổng Quan** 19](#_Toc185531111)

[**1.1. Tổng quan về dịch vụ ẩm thực Hà Nội** 19](#_Toc185531112)

[***1.1.1.*** ***Lịch sử và phát triển của ẩm thực Hà Nội*** 19](#_Toc185531113)

[***1.1.2.*** ***Các loại hình dịch vụ ẩm thực*** 19](#_Toc185531114)

[***1.1.3.*** ***Đặc điểm và xu hướng tiêu dùng ẩm thực*** 19](#_Toc185531115)

[***1.1.4.*** ***Vai trò của ông nghệ trong ngành ẩm thực*** 20](#_Toc185531116)

[**1.2. Mục tiêu** 22](#_Toc185531117)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VỀ WEBSITE VÀ CHATBOT** 23](#_Toc185531118)

[**2.1. Tổng quan về công nghệ website** 23](#_Toc185531119)

[***2.1.1. Tổng quan về Javascript*** 23](#_Toc185531120)

[***2.1.2. Tổng quan về React*** 25](#_Toc185531121)

[***2.1.4. Tổng quan về ExpressJS*** 32](#_Toc185531122)

[***2.1.5. Tổng quan về MongoDB*** 34](#_Toc185531123)

[**2.2. Tổng quan về Chatbot** 39](#_Toc185531124)

[***2.2.1. Khái niệm và phân loại Chatbot*** 39](#_Toc185531125)

[***2.2.2. Công nghệ và nền tảng phát triển Chatbot*** 40](#_Toc185531126)

[***2.2.3. Ứng dụng của Chatbot*** 40](#_Toc185531127)

[**CHƯƠNG 3: MÔ HÌNH CHỨC NĂNG VÀ MÔ HÌNH DỮ LIỆU CỦA WEBSITE GIỚI THIỆU ẨM THỰC HÀ NỘI** 41](#_Toc185531128)

[**3.1 Mô hình chức năng** 41](#_Toc185531129)

[***3.1.1. Xác định yêu cầu*** 41](#_Toc185531130)

[**3.1.2 Chức năng hệ thống** 42](#_Toc185531131)

[**3.1.3 Xây dựng biểu đồ usecase** 43](#_Toc185531132)

[**3.1.5 Biểu đồ phân rã chức năng** 54](#_Toc185531133)

[**3.1.6 Biểu đồ tuần tự** 55](#_Toc185531134)

# **DANH MỤC HÌNH ẢNH**

[Hình 2. 1. Javascript có thể tạo ra những chuyển động đẹp mắt cho website 25](#_Toc185781605)

[Hình 2. 2. ReactJS có thể tạo ra bất kỳ trang web nào mượt mà, nhanh chóng, khả năng mở rộng cao 26](#_Toc185781606)

[Hình 2. 3. Ưu điểm của ReactJS 28](#_Toc185781607)

[Hình 2. 4. Mô hình hoạt động của NodeJS 30](#_Toc185781608)

[Hình 2. 5. Tính năng của ExpressJS là gì? 33](#_Toc185781609)

[Hình 2. 6. MongoDB là một phần mềm cơ sở dữ liệu mã nguồn mở NoSQL 36](#_Toc185781610)

[Hình 2. 7. Truy vấn ad hoc được xem là tính năng tốt nhất 37](#_Toc185781611)

[Hình 2. 8. Lợi ích và hạn chế của Chatbot 39](#_Toc185781612)

[Hình 3. 1. Biểu đồ use-case của người đọc 43](#_Toc185781613)

[Hình 3. 2. Biểu đồ use-case của người đọc - thành viên 44](#_Toc185781614)

[Hình 3. 3. Biểu đồ use-case của người quản trị 44](#_Toc185781615)

[Hình 3. 4. Biểu đồ phân rã chức năng 54](#_Toc185781616)

[Hình 3. 5. Biểu đồ tuần tự đăng nhập cho người dùng hoặc quản trị viên 55](#_Toc185781617)

[Hình 3. 6. Biểu đồ tuần tự đăng ký thành viên cho người dùng 56](#_Toc185781618)

[Hình 3. 7. Biểu đồ tuần tự tìm kiếm thông tin các món ăn 57](#_Toc185781619)

[Hình 3. 8. Biểu đồ tuần tự xem thông tin chi tiết 58](#_Toc185781620)

[Hình 3. 9. Biểu đồ tuần tự bình luận bài viết 58](#_Toc185781621)

[Hình 3. 10. Biểu đồ tuần tự thêm mới nội dung về ẩm thực 59](#_Toc185781622)

[Hình 2. 19. Biểu đồ tuần tự chỉnh sửa nội dung về ẩm thực 59](#_Toc185781623)

[Hình 3. 11. Biểu đồ tuần tự xóa nội dung về ẩm thực 60](#_Toc185781624)

[Hình 3. 12. Biểu đồ tuần tự xóa tài khoản người dùng 61](#_Toc185781625)

[Hình 3. 13. Biểu đồ tuần tự xóa thông báo về cảm xúc và bình luận 61](#_Toc185781626)

# **DANH MỤC BẢNG BIỂU**

[Bảng 3.1. Bảng đặc tả hệ thống 42](#_Toc185781638)

[Bảng 3.2. Bảng đặc tả đăng nhập 44](#_Toc185781639)

[Bảng 3.3. Bảng đặc tả đăng ký 46](#_Toc185781640)

[Bảng 3.4. Bảng đặc tả đổi mật khẩu, cập nhật thông tin 47](#_Toc185781641)

[Bảng 3.5. Bảng đặc tả thả cảm xúc bài đăng 48](#_Toc185781642)

[Bảng 3.6. Bảng đặc tả bình luận bài đăng 49](#_Toc185781643)

[Bảng 3.7. Bảng đặc tả thêm mới bài đăng 50](#_Toc185781644)

[Bảng 3.8. Bảng đặc tả chỉnh sửa bài đăng 51](#_Toc185781645)

[Bảng 3.9. Bảng đặc tả xóa bài đăng 52](#_Toc185781646)

**DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Viết đầy đủ** |
| **API** | **A**pplication **P**rogramming **I**nterface |
| **BSON** | Binary JSON |
| **CSDL** | Cơ sở dữ liệu |
| **CPU** | Central Processing Unit |
| **DB** | Database |
| **I/O** | Input/Output |
| **JSON** | **J**ava**S**cript **O**bject **N**otation |
| **MERN** | MongoDB, Express.js, React.js, Node.js |
| **NoSQL** | non-SQL |
| **REST** | **RE**presentational **S**tate **T**ransfer |
| **RDBMS** | Relational Database Management System |
| **SQL** | Structured Query Language |
| **SEO** | **Search Engine Optimization** |
| **XSS** | Cross-Site Scripting |

**LỜI MỞ ĐẦU**

1. **Giới thiệu đề tài**

Trên thị trường hiện nay, với sự phát triển nhanh chóng của công nghệ thông tin, việc sử dụng trang web và chatbot đang trở thành một xu hướng quan trọng trong lĩnh vực truyền thông và quảng cáo. Đặc biệt, khi khám phá các địa điểm ẩm thực Hà Nội, việc kết hợp cả hai yếu tố này không chỉ giúp tăng cường hiệu quả quảng cáo mà còn mang lại trải nghiệm tương tác tốt hơn cho khách hàng.

Với sự phổ biến của điện thoại thông minh và internet, việc sử dụng trang web và chatbot trong lĩnh vực ẩm thực giúp tạo ra một kênh giao tiếp trực tuyến hiệu quả, giúp doanh nghiệp tiếp cận đến đông đảo khách hàng tiềm năng một cách nhanh chóng và tiện lợi. Đồng thời, việc áp dụng công nghệ này cũng giúp nâng cao trải nghiệm của khách hàng khi tìm hiểu thông tin về ẩm thực Hà nội và dễ dàng tương tác để có được những thông tin cần thiết.

1. **Lý do chọn đề tài**

Hiện nay, em nhận thấy các trang web thường không đáp ứng đủ nhu cầu của người dùng như việc chia sẻ cảm xúc và bình luận để trao đổi với những người khác về dịch vụ ẩm thử Hà Nội . Điều này gây khó khăn khi họ cần thông tin chi tiết từ các bài viết mà không tìm thấy câu trả lời chính xác.

Việc kết hợp chatbot thông minh trong website không chỉ giúp cung cấp thông tin mà còn tạo sự tương tác thân thiện với người dùng, nâng cao trải nghiệm sử dụng. Đây cũng là xu hướng phổ biến trong lĩnh vực công nghệ và dịch vụ hiện nay, giúp sinh viên áp dụng kiến thức học tập vào thực tiễn.

Kết hợp website giới thiệu dịch vụ ẩm thực với chatbot là một ý tưởng mới mẻ, tạo sự khác biệt so với các trang web ẩm thực truyền thống. Chatbot có thể gợi ý món ăn, trả lời các câu hỏi phổ biến về địa chỉ, giá cả, hoặc thậm chí là kể những câu chuyện văn hóa đằng sau các món ăn.

1. **Mục tiêu của đề tài**

* Xây dựng trang web hoàn chỉnh: Phát triển một trang web đầy đủ chức năng và chất lượng cao để giới thiệu dịch vụ ẩm thực Hà Nội.
* Sử dụng công nghệ hiện đại: Áp dụng kiến trúc hệ thống MERN (MongoDB, ExpressJS, ReactJS và NodeJS) để xây dựng ứng dụng web một cách hiệu quả và linh hoạt.
* Tích hợp các tính năng tương tác: Tạo ra các tính năng tương tác như tìm kiếm, lọc, đánh giá và bình luận để người dùng có thể tương tác với trang web một cách thuận tiện và sinh động.

**4.1. Phân tích yêu cầu và thiết kế hệ thống:**

Xác định và phân tích yêu cầu cụ thể cho website và chatbot nhằm mục đích truyền thông và giới thiệu các địa điểm ẩm thực Hà Nội.

Thiết kế kiến trúc hệ thống MERN (MongoDB, ExpressJS, ReactJS và NodeJS) để đáp ứng các yêu cầu và tích hợp chatbot một cách hiệu quả.

**4.2. Phát triển website:**

Sử dụng ReactJS để thiết kế một giao diện người dùng trực quan và dễ dàng sử dụng.

Xây dựng backend bằng các công nghệ NodeJS và ExpressJS để kiểm soát các luồng nghiệp vụ và tương tác với cơ sở dữ liệu MongoDB.

**4.3. Phát triển chatbot:**

Tích hợp chatbot vào website bằng cách sử dụng Chat Coze Web SDK

Huấn luận chatbot dựa vào data từ database

Lập trình chatbot để giới thiệu và hỗ trợ người dùng tra cứu thông tin về các địa điểm ẩm thực của Hà Nội một cách nhanh chóng và hiệu quả.

**4.4. Kiểm thử và đánh giá:**

Thực hiện việc kiểm tra, sự đánh giá nhằm đảm bảo rằng độ ổn định, an toàn và hiệu suất của website và chatbot luôn được duy trì.

Thu thập phản hồi từ người dùng để cải thiện và điều chỉnh sản phẩm.

1. **Đối tượng phạm vi nghiên cứu**

**Người dùng cuối:** Các độc giả, quan tâm đến ẩm thực của Hà Nội.

**Người quản trị hệ thống:** Quản trị viên chịu trách nhiệm duy trì và quản lý website và chatbot.

**Người quản lý dự án:** Những người có trách nhiệm điều hành và giám sát tiến độ của dự án.

Phạm vi của đề tài giới hạn trong khuân khổ nội dung phục vụ truyền thông và giới ẩm thực của thủ đô Hà Nội sử dụng mô hình công nghệ MERN

1. **Nhiệm vụ và kết quả đạt được**

Phân tích yêu cầu và đề xuất của phần mềm

Thiết kế đặc tả hệ thống

Xây dựng được cơ sở dữ liệu

Lập trình các module của hệ thống

1. **Bố cục của đồ án**

Dựa trên các nội dung đã khảo sát, để thực hiện được mục tiêu đề ra và đảm bảo tính logic, ngoài phần mở đầu và phần kết luận, đồ án sẽ được tổ chức theo các chương sau đây:

*Chương 1: Cơ sở lý thuyết*. Chương này trình bày các cơ sở lý thuyết về internet và website; trình bày kết quả nghiên cứu, tìm hiểu về các ngôn ngữ lập trình và kiến trúc hệ thống MERN (MongoDB, ExpressJS, ReactJS và NodeJS) được sử dụng trong việc xây dựng đề tài.

*Chương 2: Phân tích và thiết kế hệ thống*. Chương này giới thiệu bài toán và đưa ra các mô hình hệ thống; đề xuất các ý tưởng, chức năng cho phần mềm; trình bày các mô hình thực thể quan hệ, sơ đồ phân cấp chức năng. Phân tích chi tiết các nhiệm vụ của người sử dụng và người quản lý; thiết kế giao diện cho người dùng và thiết kế cấu trúc dữ liệu, các chức năng cho hệ thống.

*Chương 3: Kết quả và đánh giá kết quả.* Trong chương này, nêu kết quả thực hiện được của đề tài và trình bày cách cài đặt chương trình.

1. **Đóng góp của đồ án**

Trong quá trình thực hiện đồ án tốt nghiệp, em đã sử dụng những kiến thức đã học được từ trường để tiếp cận đề tài một cách linh động. Không chỉ vậy, em còn tự tìm hiểu, nghiên cứu thêm về các kiến thức mới và ngôn ngữ lập trình mới để phát triển dự án một cách nhanh chóng và hoàn thiện hơn:

*Phân tích rõ ràng vai trò của người dùng trong hệ thống*. Đối với người quản *trị*, khách hàng đã đăng nhập hay chưa đăng nhập đều được giới hạn một số chức năng nhất định.

*Đưa ra các chức năng điều khiển dành cho người quản trị*. Nghiên cứu và triển khai thực nghiệm các chức năng nhằm đáp ứng nhu cầu cho người quản lý như quản lý các thông tin về bài viết, thông tin các người dùng.

*Đưa ra chức năng thả cảm xúc và bình luận cho người dùng*. Nghiên cứu và triển khai thực nghiệm chức năng thả cảm xúc và bình luận cho người dùng. Chức năng này cho phép người dùng thể hiện cảm xúc và ý kiến cá nhân về các địa danh, cũng như trao đổi với người dùng khác về những địa điểm họ quan tâm.

**CHƯƠNG 1: Tổng Quan**

**1.1. Tổng quan về dịch vụ ẩm thực Hà Nội**

* + 1. ***Lịch sử và phát triển của ẩm thực Hà Nội***

Ẩm thực Hà Nội là sự kết tinh của hàng nghìn năm lịch sử, chịu ảnh hưởng sâu sắc từ văn hóa truyền thống Việt Nam và giao thoa với các nền văn hóa khác, đặc biệt là Trung Quốc và Pháp.

* **Thời kỳ phong kiến**: Ẩm thực Hà Nội phản ánh lối sống cung đình tinh tế và dân dã với các món ăn truyền thống như phở, bún thang, chả cá.
* **Thời kỳ thuộc địa**: Ảnh hưởng của Pháp đã mang đến các món bánh mì, cà phê và kỹ thuật nấu ăn mới, tạo nên sự đa dạng.
* **Thời kỳ hiện đại**: Ẩm thực Hà Nội tiếp tục phát triển, kết hợp truyền thống với hiện đại. Nhiều món ăn được sáng tạo mới, đồng thời dịch vụ nhà hàng, quán ăn được hiện đại hóa để phục vụ nhu cầu trong nước và quốc tế.

Ẩm thực Hà Nội ngày nay là một phần quan trọng của văn hóa, là biểu tượng cho sự giao thoa giữa truyền thống và đổi mới.

* + 1. ***Các loại hình dịch vụ ẩm thực***
* Ẩm thực đường phố
* Nhà hàng truyền thống
* Nhà hàng cao cấp (Fine Dining)
* Buffet
* Quán ăn nhanh (Fast Food)
* Dịch vụ giao hàng ẩm thực (Food Delivery)
* Quán bar, pub
* Dịch vụ tiệc tại gia và sự kiện (Catering)
* Ẩm thực chay (Vegetarian/Vegan Dining)
* Ẩm thực quốc tế
  + 1. ***Đặc điểm và xu hướng tiêu dùng ẩm thực***

**Đặc điểm tiêu dùng ẩm thực**

* Sự đa dạng trong nhu cầu
* Tính tiện lợi và nhanh chóng
* Định hướng trải nghiệm

**Xu hướng tiêu dùng ẩm thực**

* Ưu tiên thực phẩm lành mạnh
* Dịch vụ ẩm thực số hóa
* Trải nghiệm văn hóa qua ẩm thực
* Sự phát triển của ẩm thực thuần chay
* Ẩm thực bền vững
* Trải nghiệm cá nhân hóa
* Mô hình kinh doanh mới
  + 1. ***Vai trò của ông nghệ trong ngành ẩm thực***

Công nghệ đã và đang đóng vai trò quan trọng trong việc thúc đẩy sự phát triển, đổi mới và nâng cao hiệu quả trong ngành ẩm thực.

1. Tối ưu hóa quy trình kinh doanh và vận hành

* **Ứng dụng quản lý nhà hàng**: Các phần mềm như POS (Point of Sale) giúp quản lý bán hàng, kiểm soát nguyên liệu, và theo dõi hiệu suất kinh doanh một cách chính xác và nhanh chóng.
* **Tự động hóa trong chế biến**: Sử dụng thiết bị thông minh, máy móc tự động để giảm thời gian nấu nướng và nâng cao chất lượng món ăn.

1. Cải thiện trải nghiệm khách hàng

* **Đặt chỗ trực tuyến**: Các ứng dụng như NowTable, OpenTable cho phép khách hàng đặt bàn trước, giúp giảm thời gian chờ đợi.
* **Màn hình tương tác**: Một số nhà hàng sử dụng màn hình cảm ứng để khách hàng tự chọn món và thanh toán.

1. Phát triển dịch vụ giao hàng

* **Ứng dụng đặt món ăn**: Các nền tảng như GrabFood, ShopeeFood, Baemin đã cách mạng hóa cách thức khách hàng tiếp cận ẩm thực, tăng sự tiện lợi và nhanh chóng.
* **Theo dõi đơn hàng theo thời gian thực**: Khách hàng có thể theo dõi lộ trình giao hàng và ước tính thời gian nhận món.

1. Quảng bá và tiếp cận khách hàng

* **Tiếp thị số (Digital Marketing)**: Công nghệ giúp quảng bá thương hiệu qua mạng xã hội, email, và công cụ tìm kiếm.
* **Đánh giá trực tuyến**: Các nền tảng như Foody, TripAdvisor cho phép khách hàng chia sẻ trải nghiệm và giúp nhà hàng cải thiện chất lượng dịch vụ.

1. Hỗ trợ đổi mới sản phẩm và mô hình kinh doanh

* **Phân tích dữ liệu lớn (Big Data)**: Hiểu xu hướng thị trường, dự đoán nhu cầu tiêu dùng để đưa ra các chiến lược phù hợp.
* **Ứng dụng AI**: Sử dụng trí tuệ nhân tạo để phát triển thực đơn, tối ưu hóa quy trình hoặc tự động trả lời khách hàng.

1. Đổi mới trải nghiệm ăn uống

* **Robot phục vụ**: Nhiều nhà hàng đã sử dụng robot để mang món ăn đến bàn, tạo sự độc đáo và tăng hiệu quả.
* **Thực tế ảo (VR) và thực tế tăng cường (AR)**: Tạo trải nghiệm ăn uống tương tác, từ xem món ăn dưới dạng 3D đến khám phá không gian ẩm thực ảo.

**1.2. Mục tiêu**

* Xây dựng trang web hoàn chỉnh: Phát triển một trang web đầy đủ chức năng và chất lượng cao để giới thiệu về dịch vụ ẩm thực ở Hà Nội.
* Sử dụng công nghệ hiện đại: Áp dụng các công nghệ mới nhất như MongoDB, React, ExpressJS và NodeJS để xây dựng ứng dụng web một cách hiệu quả và linh hoạt.
* Tích hợp các tính năng tương tác: Tạo ra các tính năng tương tác như tìm kiếm, lọc, đánh giá và bình luận để người dùng có thể tương tác với trang web một cách thuận tiện và sinh động.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VỀ WEBSITE VÀ CHATBOT**

**2.1. Tổng quan về công nghệ website**

***2.1.1. Tổng quan về Javascript***

JavaScript là một ngôn ngữ lập trình phổ biến được sử dụng rộng rãi trong phát triển web để tạo ra các trang web động và tương tác. JavaScript (JS) ban đầu được phát triển để thêm tính năng tương tác vào các trang web. Ngày nay, nó đã trở thành một ngôn ngữ lập trình phổ biến không chỉ cho front-end mà còn cho back-end development (với Node.js), mobile app development (Với React Native), và nhiều ứng dụng khác.

So sánh với các ngôn ngữ khác

**Với Python:**

* **Điểm mạnh:** Python dễ học, cú pháp rõ ràng, phù hợp cho AI/ML, và có thư viện phong phú.
* **Điểm yếu:** Không phù hợp cho front-end web development mà cần phải kết hợp với JavaScript hoặc các framework khác.

**Với Java:**

* **Điểm mạnh:** Được sử dụng rộng rãi trong enterprise applications, hệ thống lớn, hỗ trợ đa luồng.
* **Điểm yếu:** Cú pháp phức tạp hơn JavaScript, khởi động chậm hơn cho các ứng dụng nhỏ.

**Với Ruby:**

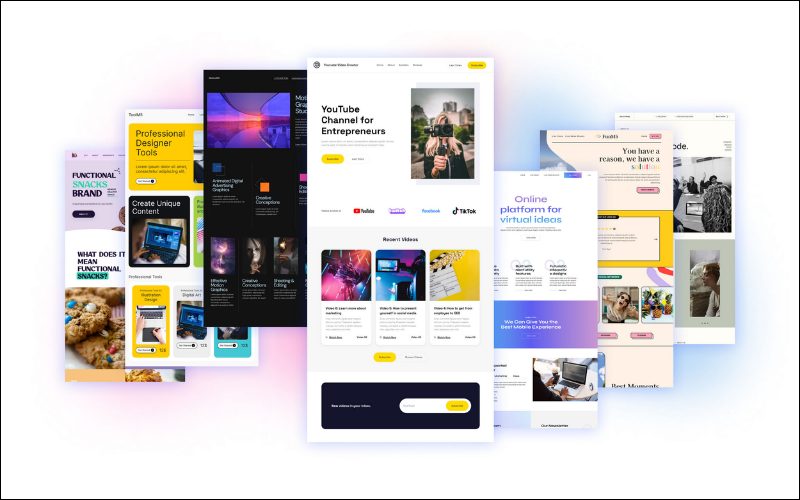
* **Điểm mạnh:** Đơn giản, dễ đọc, phù hợp cho web development.
* **Điểm yếu:** Hiệu suất không cao bằng JavaScript, hạn chế trong các ứng dụng cần xử lý nhanh.

**Điểm mạnh của JavaScript**

1. **Thiết kế cho web:** JavaScript được thiết kế đặc biệt cho việc tương tác trực tiếp trên trình duyệt.
2. **Không đồng bộ:** Xử lý không đồng bộ (asynchronous processing) làm cho việc làm web trở nên mượt mà hơn.
3. **Community lớn:** Có một cộng đồng lớn, hỗ trợ mạnh mẽ và nhiều thư viện/framework phát triển.

Điểm yếu của JavaScript

1. **Bảo mật:** JavaScript dễ bị tấn công XSS (Cross-Site Scripting) nếu không được bảo vệ cẩn thận.
2. **Không đồng nhất:** Sự khác biệt giữa các trình duyệt có thể gây ra các vấn đề tương thích.
3. **Performance:** Hiệu suất không thể so sánh với các ngôn ngữ như C++ hoặc Rust cho các ứng dụng yêu cầu tính toán nặng.



Hình 2. 1. *Javascript có thể tạo ra những chuyển động đẹp mắt cho website*

***2.1.2. Tổng quan về React***

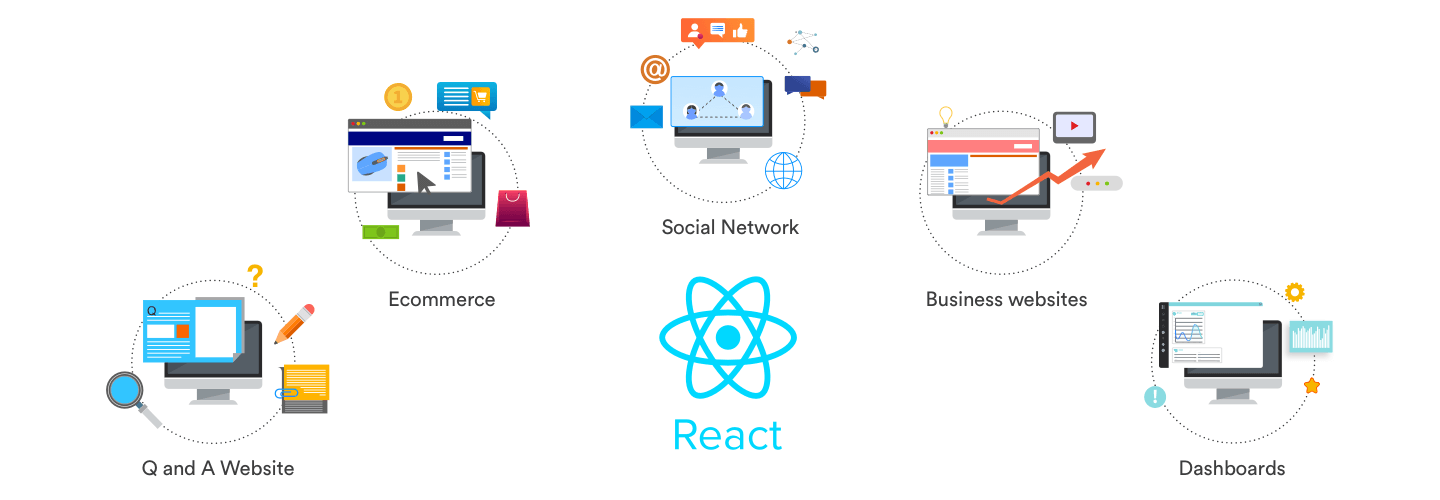
**Giới thiệu React**

ReactJS được phát triển bởi Facebook và được giới thiệu lần đầu tiên vào năm 2011. Ban đầu, ReactJS được phát triển để xây dựng giao diện người dùng trên trang web Facebook, nhằm cải thiện tốc độ và hiệu suất của ứng dụng web.

Tuy nhiên, ReactJS không được công bố cho cộng đồng phát triển cho đến năm 2013, khi Facebook công bố mã nguồn mở của nó và giới thiệu cho cộng đồng lập trình viên. Từ đó, ReactJS nhanh chóng trở thành một trong những thư viện phổ biến nhất để phát triển các ứng dụng web động.

Sau đó, vào năm 2015, Facebook giới thiệu phiên bản React Native, một framework phát triển ứng dụng di động sử dụng ReactJS. React Native cho phép các nhà phát triển xây dựng các ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn, tương tự như ReactJS trên web.

Hiện nay, ReactJS đã trở thành một trong những thư viện phát triển web phổ biến nhất, được sử dụng rộng rãi bởi các công ty lớn và nhỏ trên toàn thế giới. Facebook cũng tiếp tục đầu tư phát triển và nâng cấp ReactJS để đáp ứng nhu cầu của cộng đồng phát triển.



Hình 2. 2. ReactJS có thể tạo ra bất kỳ trang web nào mượt mà, nhanh chóng, khả năng mở rộng cao

**Lợi ích của ReactJS**

ReactJS mang đến nhiều lợi ích cho việc phát triển ứng dụng web, bao gồm:

**Hiệu suất cao:** ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM cho phép ReactJS cập nhật các thay đổi trên trang web một cách nhanh chóng và hiệu quả hơn so với cách truyền thống, giúp tăng tốc độ và hiệu suất của ứng dụng.

**Tái sử dụng:** ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

**Dễ dàng quản lý trạng thái:** ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.

**Hỗ trợ tốt cho SEO:** ReactJS cho phép các nhà phát triển xây dựng ứng dụng web với khả năng tương thích tốt với SEO. Với sự hỗ trợ của các thư viện như React Helmet, ReactJS cho phép các nhà phát triển tùy chỉnh và quản lý các phần tử meta và title cho từng trang web.

**Hỗ trợ đa nền tảng:** ReactJS không chỉ được sử dụng để phát triển các ứng dụng web, mà còn được sử dụng để phát triển các ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

Tính năng nổi bật của ReactJS

ReactJS có nhiều tính năng hữu ích cho việc phát triển ứng dụng web, bao gồm:

**Components**: ReactJS cho phép phát triển ứng dụng web theo mô hình component. Các component là các phần tử UI độc lập có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng.

**Virtual DOM**: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM được lưu trữ trong bộ nhớ và được cập nhật một cách nhanh chóng khi có thay đổi, giúp tăng tốc độ và hiệu suất của ứng dụng.

**JSX**: JSX là một ngôn ngữ lập trình phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI. JSX kết hợp HTML và JavaScript, giúp cho việc viết mã dễ hiểu và dễ bảo trì hơn.

**State và Props**: ReactJS cho phép quản lý trạng thái của các thành phần UI thông qua State và Props. State là trạng thái của một thành phần được quản lý bởi nó chính, trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.

**Hỗ trợ tốt cho SEO**: ReactJS hỗ trợ tốt cho việc tối ưu hóa SEO. Với các thư viện như React Helmet, các nhà phát triển có thể quản lý các phần tử meta và title cho từng trang web, giúp tăng khả năng tìm kiếm và tăng cường trải nghiệm người dùng.

**Hỗ trợ đa nền tảng**: ReactJS không chỉ được sử dụng để phát triển ứng dụng web, mà còn được sử dụng để phát triển ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

**Redux**: Redux là một thư viện quản lý trạng thái cho các ứng dụng ReactJS. Nó giúp quản lý trạng thái của ứng dụng một cách chính xác và dễ dàng, đồng thời giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.



Hình 2. 3. Ưu điểm của ReactJS

Các dự án sử dụng ReactJS thành công

ReactJS là một trong những công nghệ phát triển web phổ biến nhất hiện nay, được sử dụng trong nhiều dự án thành công của các công ty lớn trên thế giới. Dưới đây là một số dự án thành công sử dụng ReactJS:

* **Facebook**: ReactJS là một trong những công nghệ chính của Facebook, được sử dụng để xây dựng nhiều sản phẩm của họ như Facebook, Instagram, Messenger...
* **Netflix:** Netflix sử dụng ReactJS để xây dựng giao diện người dùng cho trang web của họ và cho các ứng dụng di động.
* **Airbnb**: Airbnb sử dụng ReactJS để xây dựng giao diện người dùng cho trang web của họ và cho các ứng dụng di động.
* **Uber**: Uber sử dụng ReactJS để xây dựng giao diện người dùng cho ứng dụng di động của họ.
* **Dropbox**: Dropbox sử dụng ReactJS để xây dựng giao diện người dùng cho trang web và cho các ứng dụng di động của họ.
* **Atlassian**: Atlassian sử dụng ReactJS để xây dựng giao diện người dùng cho nhiều sản phẩm của họ như Jira, Confluence...

Tất cả những dự án trên đều cho thấy rằng ReactJS là một công nghệ rất phổ biến và thành công trong việc xây dựng các ứng dụng web và di động.

2.1.3. Tổng quan về NodeJS

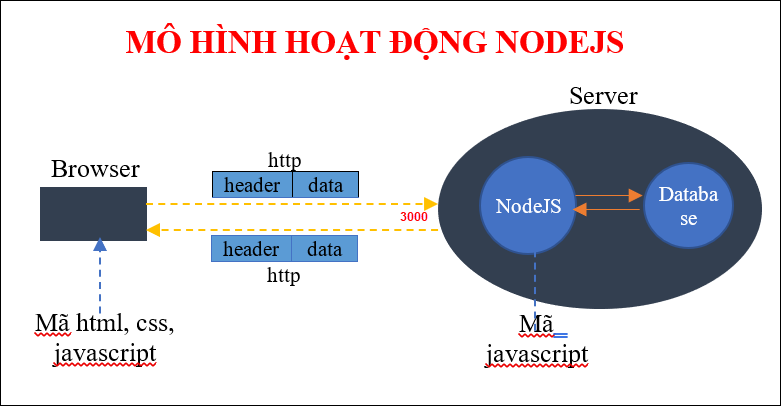
Node.js được phát triển từ Javascript vào năm 2009 bởi Ryan Dahl. NodeJS có cách thức hoạt động chủ yếu trên Server sử dụng để xây dựng cho các ứng dụng realtime. NodeJS dùng mô hình I/O lập trình dựa theo sự kiện non-blocking.

Nguyên lý hoạt động của NodeJS

Node.js là một nền tảng phát triển phía server được xây dựng trên nền tảng JavaScript. Khi một yêu cầu mạng đến từ một client, Node.js sẽ xử lý yêu cầu đó bằng cách thực hiện các bước như sau:

* Node.js tạo một event loop để theo dõi các yêu cầu mạng đến và đi.
* Khi một yêu cầu mạng đến, Node.js sẽ tạo một worker thread (luồng làm việc) để xử lý yêu cầu đó.
* Trong worker thread, Node.js sẽ thực hiện các tác vụ xử lý yêu cầu, chẳng hạn như đọc và ghi vào cơ sở dữ liệu, đọc và ghi file, tương tác với API, ...
* Khi worker thread đã hoàn thành các tác vụ, Node.js sẽ trả về kết quả cho client qua mạng.
* Nếu có yêu cầu mạng mới đến, Node.js sẽ tạo một worker thread mới để xử lý yêu cầu đó.

Các yêu cầu mạng đến và đi trong Node.js được xử lý bằng cách sử dụng các hàm callback, Promise, async/await để đảm bảo tính phi đồng bộ và tăng hiệu suất của ứng dụng. Các yêu cầu mạng được xử lý một cách độc lập, giúp tránh tình trạng "blocking" (chặn) trong quá trình xử lý yêu cầu. Node.js cũng có thể hoạt động với các module và thư viện khác để hỗ trợ cho việc phát triển ứng dụng web.



Hình 2. 4. Mô hình hoạt động của NodeJS

**Ưu điểm của NodeJS**

Hiệu suất cao: Node.js được xây dựng trên nền tảng JavaScript, được thiết kế để xử lý các yêu cầu mạng phức tạp với tốc độ nhanh và hiệu quả.

Lập trình phi đồng bộ (asynchronous programming): Node.js sử dụng các callback, Promise, async/await để xử lý các yêu cầu mạng phi đồng bộ, giúp tăng hiệu suất của ứng dụng và giảm thiểu thời gian đáp ứng cho người dùng.

Có nhiều thư viện và module hỗ trợ: Node.js có rất nhiều thư viện và module hỗ trợ cho việc phát triển ứng dụng web, giúp các nhà phát triển nhanh chóng xây dựng ứng dụng với đầy đủ tính năng.

Dễ dàng mở rộng: Node.js có khả năng mở rộng tốt, cho phép các nhà phát triển tăng hiệu suất của ứng dụng bằng cách thêm các máy chủ và máy chủ công việc (worker servers).

Đơn giản và dễ học: Vì Node.js được xây dựng trên nền tảng JavaScript, nên các nhà phát triển có thể nhanh chóng học và sử dụng nó.

**Nhược điểm của NodeJS**

Không phù hợp cho các ứng dụng CPU-bound: Node.js thường được sử dụng cho các ứng dụng I/O-bound (yêu cầu đọc và ghi vào cơ sở dữ liệu, đọc và ghi file, tương tác với API, ...) nên nó không phù hợp cho các ứng dụng CPU-bound (như xử lý ảnh, phân tích dữ liệu lớn, ...).

Có thể gây ra vấn đề khi xử lý lỗi: Khi xử lý lỗi trong Node.js, các nhà phát triển cần quản lý callback và exception để tránh gây ra vấn đề trong quá trình xử lý các yêu cầu mạng.

Phiên bản thay đổi liên tục: Node.js thường có các phiên bản thay đổi liên tục, điều này đôi khi gây khó khăn cho các nhà phát triển khi cập nhật các thư viện và module để đảm bảo tính tương thích của ứng dụng.

Những ứng dụng phổ biến của Node.js hiện nay

Node.js là một nền tảng phát triển phía server, có thể được sử dụng trong nhiều ứng dụng lập trình web, bao gồm:

Phát triển các ứng dụng web động (dynamic web applications): Node.js cho phép phát triển các ứng dụng web động như các trang web tương tác cao hoặc các ứng dụng phức tạp, cung cấp tính năng xử lý yêu cầu đồng thời và xử lý dữ liệu theo thời gian thực.

Phát triển các ứng dụng web thời gian thực (real-time web applications): Node.js cung cấp tính năng xử lý sự kiện theo thời gian thực, giúp phát triển các ứng dụng web thời gian thực như các ứng dụng chat, các trò chơi trực tuyến hoặc các ứng dụng tương tác cao.

Xử lý các yêu cầu API (APIs): Node.js là một nền tảng phát triển tuyệt vời để xử lý các yêu cầu API, cho phép phát triển các ứng dụng web dựa trên API.

Xử lý các tác vụ đồng bộ và không đồng bộ (synchronous và asynchronous tasks): Node.js cho phép phát triển các ứng dụng web đồng bộ và không đồng bộ, giúp tăng tốc độ xử lý của ứng dụng.

Phát triển các ứng dụng web đơn trang (single-page applications): Node.js cung cấp các thư viện và framework như Express.js, Sails.js, Meteor.js để phát triển các ứng dụng web đơn trang.

Phát triển các ứng dụng web liên kết nhiều hệ thống (microservices): Node.js cho phép phát triển các ứng dụng web liên kết với nhiều hệ thống khác nhau như cơ sở dữ liệu, các hệ thống phân tán, các dịch vụ bên thứ ba, giúp phát triển các ứng dụng web phức tạp và có tính mở rộng cao.

***2.1.4. Tổng quan về ExpressJS***

**Giới thiệu ExpressJS**

ExpressJS là một framework mạnh mẽ và phổ biến được xây dựng trên nền tảng **NodeJS**. Nó giúp đơn giản hóa quá trình phát triển ứng dụng web bằng cách cung cấp các công cụ và thư viện cần thiết để xử lý yêu cầu và phản hồi HTTP một cách dễ dàng.

ExpressJS cho phép bạn xây dựng các ứng dụng web độc lập, linh hoạt và có hiệu suất cao. Với cú pháp đơn giản và dễ hiểu, ExpressJS giúp lập trình viên tập trung vào việc xây dựng logic ứng dụng thay vì phải lo lắng về các chi tiết cơ bản của việc xử lý yêu cầu và phản hồi.

Điều đặc biệt về ExpressJS là khả năng sử dụng **middleware**. Middleware là các chức năng trung gian được thực thi trước khi yêu cầu của người dùng được xử lý hoặc sau khi phản hồi được gửi đi. Điều này giúp bạn thực hiện các tác vụ như xác thực người dùng, ghi log, xử lý lỗi và nhiều hơn nữa một cách dễ dàng.

**A purple and white diagram

Description automatically generated**

Hình 2. 5. *Tính năng của ExpressJS là gì?*

ExpressJS là một nền tảng lập trình web phổ biến trên NodeJS, mã nguồn mở và được áp dụng rộng rãi. Nó đi kèm với một loạt tính năng mạnh mẽ, cho phép lập trình viên xây dựng và triển khai các ứng dụng web nhanh chóng và dễ dàng. Dưới đây là một vài tính năng chính của ExpressJS:

**Phát triển máy chủ nhanh hơn:** ExpressJS tối ưu hóa cú pháp và cung cấp các phương thức và hàm tiện ích để xử lý các tác vụ phổ biến trong lập trình web. Nhờ đó, bạn có thể viết code ngắn gọn và tối giản hóa quy trình phát triển.

**Định hướng (Routing):** ExpressJS cung cấp một hệ thống định hướng hiệu quả, giúp bạn định nghĩa các đường dẫn (routes) để điều khiển các yêu cầu từ người dùng và trả lời phù hợp. Điều này giúp tổ chức và quản lý các thành phần của ứng dụng một cách dễ dàng.

**Middleware:** ExpressJS hỗ trợ middleware, cho phép bạn thêm các chức năng trung gian vào quy trình xử lý yêu cầu và phản hồi. Middleware giúp xác thực người dùng, ghi log, xử lý lỗi, nén dữ liệu và thực hiện nhiều tác vụ khác một cách linh hoạt.

**Cấu hình môi trường:** ExpressJS cung cấp một cách để cấu hình môi trường phát triển và môi trường sản xuất. Bạn có thể thiết lập các biến môi trường, cấu hình định dạng và quy tắc, tùy chỉnh ứng dụng của mình theo các môi trường khác nhau.

**Xử lý lỗi:** ExpressJS cung cấp cơ chế xử lý lỗi cho phép bạn kiểm soát và xử lý các lỗi xảy ra trong quá trình xử lý yêu cầu. Bạn có thể tạo ra các middleware để xử lý lỗi và phản hồi với các thông báo lỗi tùy chỉnh.

**Ưu điểm của ExpressJS**

Ngôn ngữ đơn giản và sử dụng dễ dàng: ExpressJS sở hữu cú pháp rõ ràng và dễ hiểu, cho phép các lập trình viên nhanh chóng hiểu rõ và triển khai những tính năng mới một cách dễ dàng.

Linh hoạt: ExpressJS không áp đặt một cấu trúc cụ thể, cho phép lập trình viên tự do tùy chỉnh và xây dựng ứng dụng theo ý muốn.

Hỗ trợ middleware: ExpressJS cung cấp hệ thống middleware mạnh mẽ, cho phép thực hiện các chức năng như xác thực, ghi log, nén dữ liệu và xử lý lỗi một cách linh hoạt và dễ dàng.

Hiệu suất cao: ExpressJS được xây dựng trên NodeJS, nền tảng có hiệu suất cao, cho phép xử lý nhanh chóng các yêu cầu web đồng thời và có khả năng mở rộng tốt.

**Nhược điểm của ExpressJS**

Thiếu cấu trúc: Do ExpressJS không áp đặt một cấu trúc nghiêm ngặt, việc tổ chức dự án và quản lý mã nguồn có thể trở nên khó khăn, đặc biệt khi ứng dụng phát triển lớn và phức tạp.

Khả năng mở rộng: Khi ứng dụng phát triển lớn và phức tạp, việc quản lý mã nguồn và mở rộng có thể trở nên khó khăn với ExpressJS. Cần có sự kiểm soát cẩn thận để tránh sự phức tạp và rối rắm trong việc quản lý các module và tương tác giữa chúng.

Cộng đồng hỗ trợ: Mặc dù ExpressJS có một cộng đồng lớn và đầy đủ tài liệu, tuy nhiên, không đạt được mức độ hỗ trợ như các framework web khác như **Angular** hoặc **React**.

***2.1.5. Tổng quan về MongoDB***

**Giới thiệu MongoDB**

MongoDB là một cơ sở dữ liệu hướng tài liệu (document), khác với các cơ sở dữ liệu quan hệ như MySQL sử dụng các bảng (table) để phân chia dữ liệu và dùng các trường (column) để định nghĩa dữ liệu, MongoDB không yêu cầu bạn phải chỉ định dạng dữ liệu của mình. Thay vào đó, nó sử dụng một schema rất linh hoạt được gọi là BSON. BSON có thể được hiểu đơn giản là dạng document JSON. Chính vì dữ liệu không bị ràng buộc nên các đơn vị lưu trữ của MongoDB, hay còn được gọi là các collection sẽ có kích thước và các document khác nhau. Ngoài ra, việc lưu trữ các document dưới dạng JSON là một lợi thế giúp tăng hiệu suất cho các quá trình truy vấn và duyệt dữ liệu.

MongoDB được sử dụng rất nhiều bởi các doanh nghiệp lớn, chúng ta có thể kể đến ở đây như: eBay, SourceForge và The New York Times. Được tin dùng như vậy bởi các tập đoàn lớn là vì DB này mang lại rất nhiều tính năng ưu việt đáp ứng được nhu cầu tải trọng cao và cần sự ổn định - các yếu tố tiên quyết để có thể xây dựng nên các hệ thống lớn.

Một số tính năng của MongoDB có thể được kể đến như:

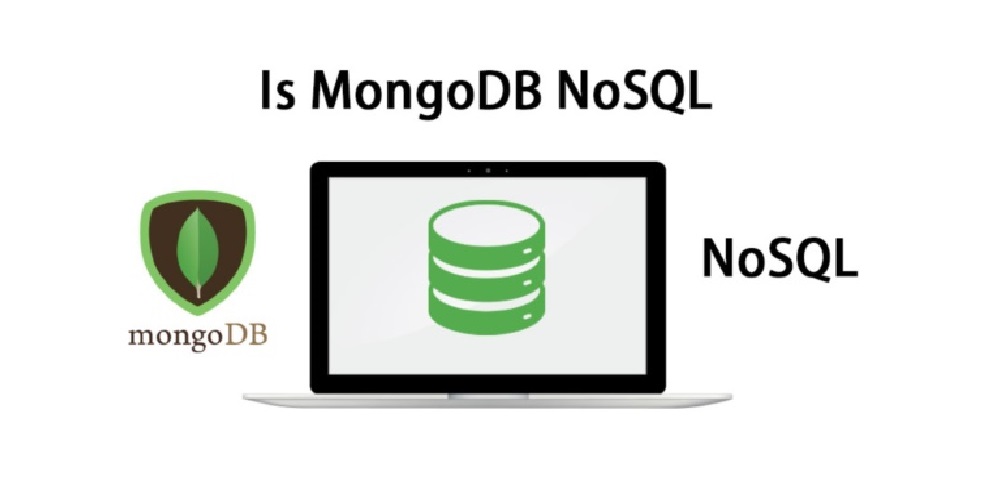
**Các ad hoc query:** hỗ trợ search bằng field, các phép search thông thường, regular expression searches, và range queries.

**Indexing:**bất kì field nào trong BSON document cũng có thể được index.

**Replication:** có ý nghĩa là “nhân bản”, là có một phiên bản giống hệt phiên bản đang tồn tại, đang sử dụng. Với cơ sở dữ liệu, nhu cầu lưu trữ lớn, đòi hỏi cơ sở dữ liệu toàn vẹn, không bị mất mát trước những sự cố ngoài dự đoán là rất cao. Vì vậy, người ta nghĩ ra khái niệm “nhân bản”, tạo một phiên bản cơ sở dữ liệu giống hệt cơ sở dữ liệu đang tồn tại, và lưu trữ ở một nơi khác, đề phòng có sự cố.

**Aggregation:**Các Aggregation operation xử lý các bản ghi dữ liệu và trả về kết quả đã được tính toán. Các phép toán tập hợp nhóm các giá trị từ nhiều Document lại với nhau, và có thể thực hiện nhiều phép toán đa dạng trên dữ liệu đã được nhóm đó để trả về một kết quả duy nhất. Trong SQL, count(\*) và GROUP BY là tương đương với Aggregation trong MongoDB.

**Lưu trữ file:**MongoDB được dùng như một hệ thống file tận dụng những function trên và hoạt động như một cách phân phối qua sharding.



Hình 2. 6. *MongoDB là một phần mềm cơ sở dữ liệu mã nguồn mở NoSQL*

**Tìm hiểu về MongoDB: Khái niệm NoSQL**

NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: Non-Relational SQL hay có nơi thường gọi là Not-Only SQL.

NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value.

NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ) về tốc độ, tính năng, khả năng mở rộng,...

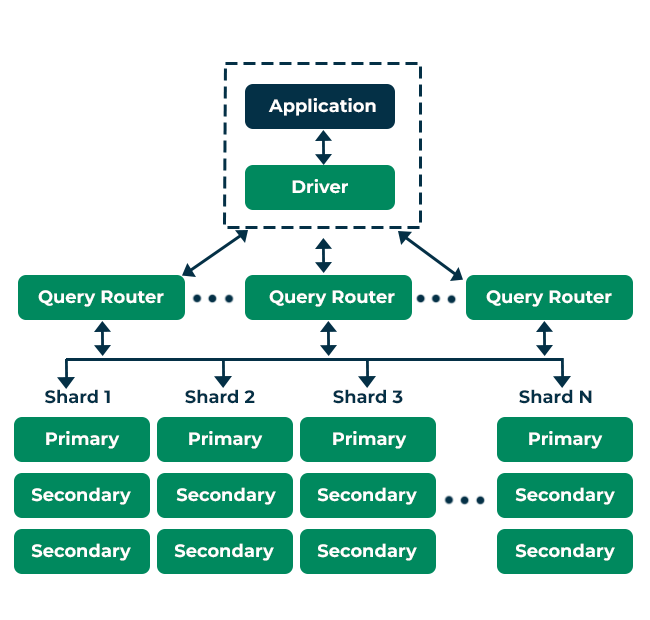
Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc ...

NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.

NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable,...

#### NoSQL hoạt động như thế nào

Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu để truy cập và quản lý dữ liệu. Các loại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữ liệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt một số hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.



Hình 2. 7. Truy vấn ad hoc được xem là tính năng tốt nhất

**Ưu điểm của MongoDB**

Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.

Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.

MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:

Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.

**Nhược điểm của MongoDB**

Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.

Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.

Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.

Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

**Nên dùng MongoDB trong trường hợp nào?**

**Khi quản lý và truyền tải content:**quản lý đa dạng nhiều sản phẩm và có nội dung khác nhau chỉ trong một kho lưu trữ dữ liệu cho phép thay đổi và phản hồi nhanh chóng, không làm phức tạp hóa thêm từ hệ thống content

**Cấu trúc mobile và social:** MongoDB cung cấp một platform có sẵn, phản xạ nhanh, và dễ mở rộng cho phép rất nhiều khả năng đột phá, phân tích real-time, và hỗ trợ toàn cầu.

**Quản lý data khách hàng:**tận dụng khả năng query nhanh chóng cho phân tích real-time trên cơ sở dữ liệu người dùng cực lớn với các mô hình data phức tạp bằng các schema linh hoạt và tự động sharding cho mở rộng chiều ngang.

**2.2. Tổng quan về Chatbot**

### ***2.2.1. Khái niệm và phân loại Chatbot***

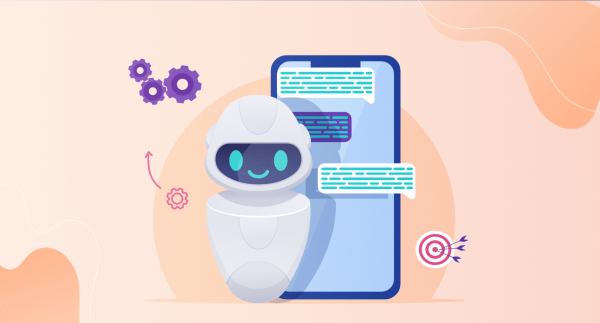
Chatbot là một chương trình trí tuệ nhân tạo được tạo ra để mô phỏng các cuộc trò chuyện với người dùng thông qua internet. Sử dụng công nghệ trí tuệ nhân tạo (AI) và xử lý ngôn ngữ tự nhiên (NLP), chatbot có khả năng hiểu câu hỏi và tự động đưa ra phản hồi.

Là một sản phẩm của công nghệ hiện đại, chatbot đóng vai trò quan trọng trong việc kết nối con người với các phần mềm tự động. Nhờ tính ứng dụng cao, chatbot trở thành công cụ hỗ trợ hiệu quả trong việc tương tác giữa doanh nghiệp và khách hàng.

Hiện nay, chatbot hiện diện trên nhiều nền tảng công nghệ khác nhau, từ loa thông minh trong gia đình đến các ứng dụng nhắn tin tại nơi làm việc. Người dùng có thể dễ dàng kết nối với chatbot thông qua các ứng dụng phổ biến như Siri của Apple, Google Assistant, hay Amazon Alexa.

Tuy nhiên, bên cạnh những lợi ích, chatbot vẫn tồn tại một số hạn chế:

* Nội dung phản hồi còn giới hạn, thiếu tính linh hoạt.
* Câu trả lời có thể gây nhàm chán vì mang tính máy móc, lặp lại.
* Việc lập trình phức tạp dẫn đến chi phí cao.
* Không phải tất cả các doanh nghiệp hay lĩnh vực đều phù hợp để sử dụng chatbot.



Hình 2. 8. Lợi ích và hạn chế của Chatbot

### ***2.2.2. Công nghệ và nền tảng phát triển Chatbot***

**Nền tảng không cần lập trình (No-code/Low-code):**

* **Dialogflow (Google):** Hỗ trợ xây dựng chatbot mạnh mẽ với AI tích hợp.
* **Microsoft Bot Framework:** Nền tảng toàn diện cho việc tạo và triển khai chatbot.
* **IBM Watson Assistant:** Cung cấp giải pháp AI để phát triển chatbot với khả năng phân tích ngôn ngữ sâu.

**Nền tảng nhắn tin xã hội:**

* **Facebook Messenger:** Cho phép tích hợp chatbot để giao tiếp với khách hàng trực tiếp.
* **WhatsApp Business API:** Tạo chatbot cho các doanh nghiệp giao tiếp hiệu quả qua WhatsApp.
* **Telegram Bot API:** Hỗ trợ phát triển chatbot với tính năng đa dạng trên nền tảng Telegram.

**Ứng dụng thoại và trợ lý ảo:**

* **Amazon Alexa:** Tạo các kỹ năng (skills) để chatbot giao tiếp qua giọng nói.
* **Google Assistant:** Phát triển chatbot dựa trên Google để hỗ trợ trên nhiều thiết bị.

**Công cụ mã nguồn mở:**

* **Rasa:** Framework mã nguồn mở mạnh mẽ để phát triển chatbot AI.
* **Botpress:** Nền tảng mã nguồn mở chuyên phát triển chatbot doanh nghiệp.

**Nền tảng tùy chỉnh:**

* Các công ty hoặc doanh nghiệp có thể xây dựng chatbot riêng bằng cách sử dụng các ngôn ngữ lập trình như Python, JavaScript, hoặc các thư viện như TensorFlow, PyTorch, và spaCy.

### ***2.2.3. Ứng dụng của Chatbot***

Chatbot được ứng dụng rộng rãi trong nhiều lĩnh vực nhờ khả năng tự động hóa tương tác, cải thiện trải nghiệm người dùng và tối ưu hóa quy trình.

**1. Dịch vụ khách hàng và hỗ trợ:**

* **Trả lời câu hỏi thường gặp (FAQ):** Chatbot có thể trả lời nhanh các câu hỏi phổ biến như thông tin sản phẩm, dịch vụ, chính sách bảo hành.
* **Hỗ trợ 24/7:** Đảm bảo khách hàng nhận được trợ giúp bất kể thời gian, giảm tải cho đội ngũ nhân viên hỗ trợ.
* **Xử lý khiếu nại:** Tiếp nhận và giải quyết vấn đề cơ bản trước khi chuyển đến nhân viên phụ trách.

**2. Thương mại điện tử và bán lẻ:**

* **Hướng dẫn mua sắm:** Gợi ý sản phẩm dựa trên sở thích, nhu cầu của khách hàng.
* **Xử lý đơn hàng:** Hỗ trợ đặt hàng, thanh toán và theo dõi tình trạng đơn hàng.

# **CHƯƠNG 3: MÔ HÌNH CHỨC NĂNG VÀ MÔ HÌNH DỮ LIỆU CỦA WEBSITE GIỚI THIỆU ẨM THỰC HÀ NỘI**

## **3.1 Mô hình chức năng**

### ***3.1.1. Xác định yêu cầu***

Hệ thống có chức năng ủy quyền người dùng, do đó:

**Đối với người dùng là người đọc - không phải thành viên:**

* Được xem thông tin về dịch vụ ẩm thực của thủ đô Hà Nội, xem thông của người đăng bài
* Được xem lượt tương tác và các bình luận của các bài đăng
* Được tìm kiếm các bài đăng và người đăng
* Được đăng ký tài khoản, đăng nhập

**Đối với người dùng là người đọc - thành viên:**

* Được xem thông tin về dịch vụ ẩm thực của thủ đô Hà Nội, xem thông của người đăng bài
* Được xem lượt tương tác và các bình luận của các bài đăng
* Được tìm kiếm các bài đăng và người đăng
* Được đăng ký tài khoản, đăng nhập
* Được thả cảm xúc và viết bình luận
* Được nhận thông báo khi người khác phản hồi lại bình luận
* Được quản lý tài khoản thành viên của mình
* Sử dụng chatbot để tìm kiếm thông tin về dịch vụ ẩm thực của thủ đô Hà Nội

**Đối với người dùng là người quản trị:**

Ngoài các tính năng của người đọc - thành viên:

* Được thêm bài viết về dịch vụ ẩm thực của thủ đô Hà Nội
* Được chỉnh sửa thông tin của bài viết: Khi có sự thay đổi về địa điểm, giá thành, người quản lý có thể thay đổi thông tin để người dùng cập nhật kịp thời
* Được xóa bài viết: Có thể vì một lý do nào đó hoặc buộc bị xóa khỏi hệ thống thì người quản lý cũng có thể xóa được bài đăng đó
* Được xóa tài khoản các thành viên: Khi người dùng vi phạm các điều luật không gian mạng hoặc comment gây ảnh hưởng đến các người dùng khác hoặc vì một lý do nào đó yêu cầu xóa tài khoản, người quản lý có thể xóa toàn bộ thông tin liên quan đến tài khoản đó
* Được xem các thống kê về các bài đăng như số lượt like, comment, lượt đọc

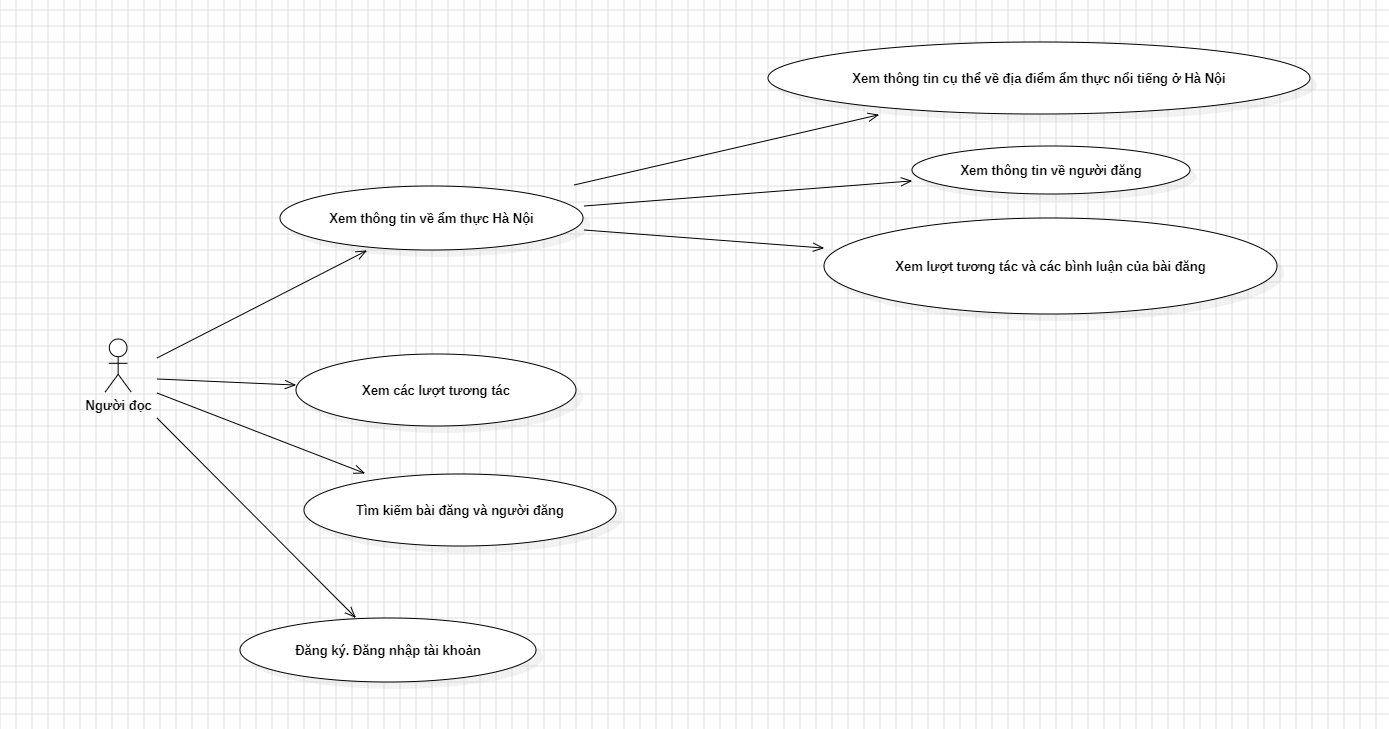
## **3.1.2 Chức năng hệ thống**

Bảng 3.1. Bảng đặc tả hệ thống

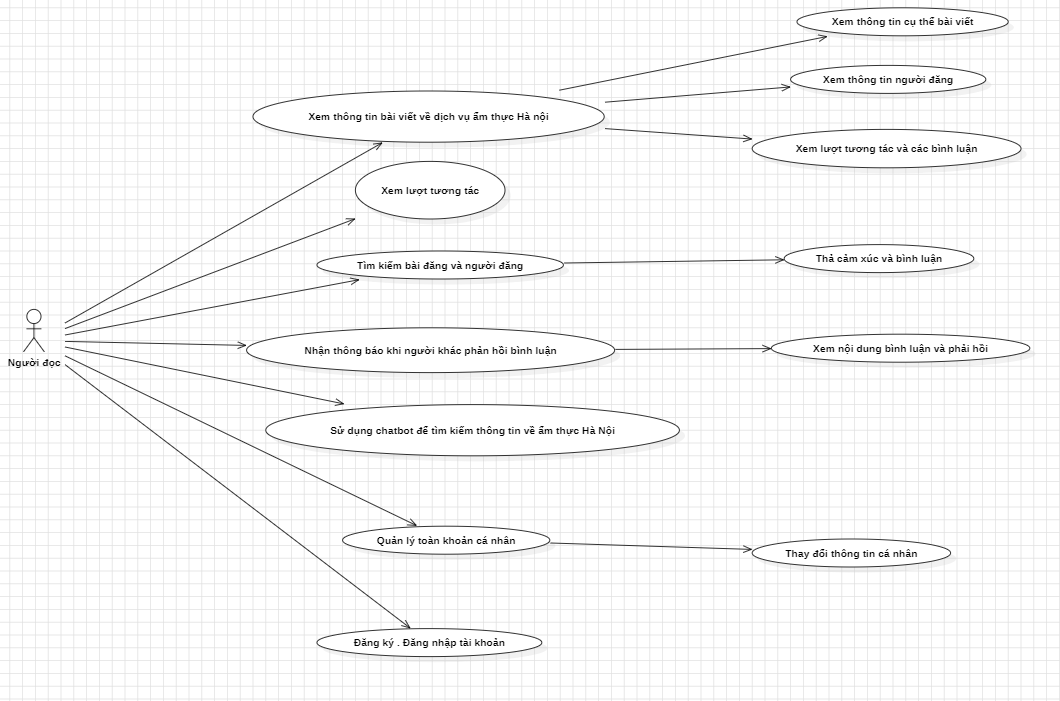
|  |  |  |
| --- | --- | --- |
| **Vai trò** | | **Mô tả** |
| Người quản trị | | Người giữ vai trò quan trọng nhất trong hệ thống, quản lý có quyền thêm, chỉnh sửa và xóa tất cả các thông tin của hệ thống. |
| Khách hàng | Không phải thành viên | Thao tác trực tiếp trên hệ thống người dùng, với các chức năng: xem các post về ẩm thực của thủ đô Hà Nội, xem lượt đánh giá và các bình luận về bài viết đó, tìm kiếm các bài đăng và người dùng khác, đăng ký tài khoản, đăng nhập |
| Thành viên | Vận hành các chức năng của người đọc - không phải thành viên, ngoài ra người đọc thành viên có thể thả cảm xúc, viết bình luận về các bài đăng mà người đọc quan tâm, có thông báo khi người khác phản hồi bình luận của mình, quản lý tài khoản thành viên của mình, sử dụng chatbot để tìm kiếm thông tin |

## **3.1.3 Xây dựng biểu đồ usecase**

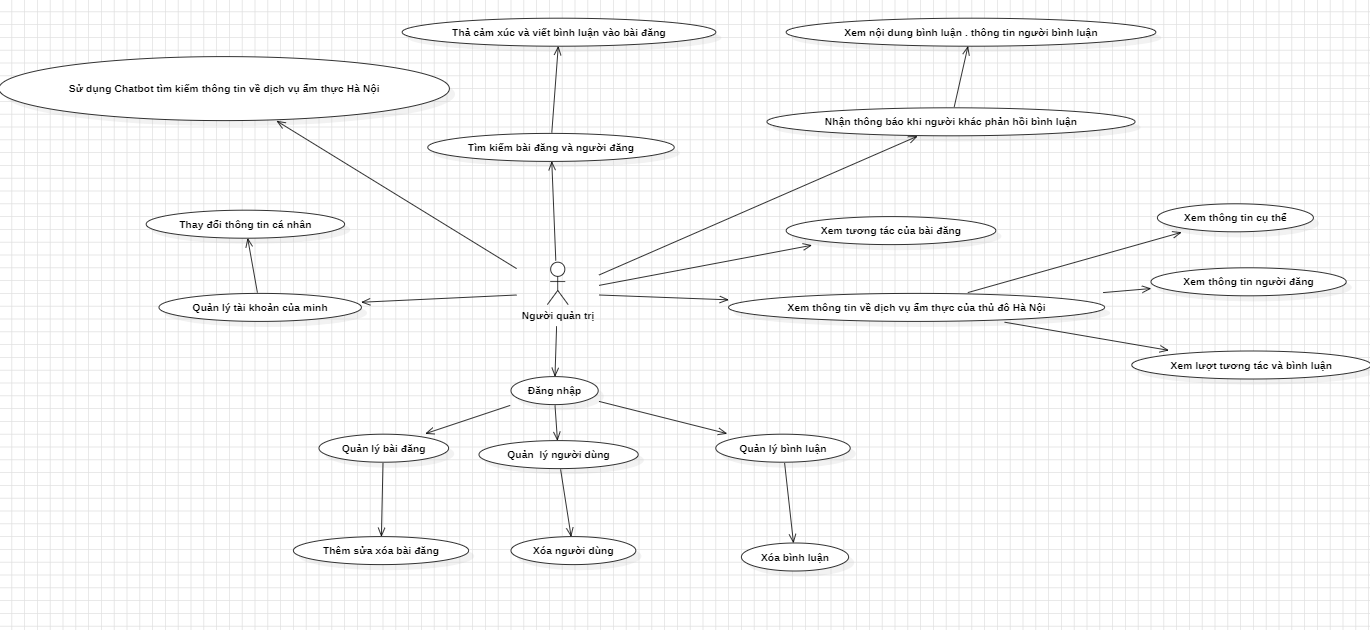
Biểu đồ trường hợp sử dụng (use-case) là một phương pháp được sử dụng trong kỹ thuật phần mềm và hệ thống để nhận diện các yêu cầu chức năng của hệ thống. Hoặc, Sơ đồ trường hợp sử dụng là một công cụ quan trọng trong lập trình phần mềm và thiết kế hệ thống, giúp xác định các yêu cầu chức năng cần thiết cho hệ thống. Use case mô tả sự tương tác đặc trưng giữa người dùng bên ngoài và hệ thống. Use case cũng mô tả các yêu cầu đối với hệ thống.



Hình 3. 1. Biểu đồ use-case của người đọc



Hình 3. 2. Biểu đồ use-case của người đọc - thành viên



Hình 3. 3. Biểu đồ use-case của người quản trị

**3.1.4 Đặc tả usecase**

Bảng 3.2. Bảng đặc tả đăng nhập

|  |  |  |
| --- | --- | --- |
| Tên use case | Đăng nhập | |
| Tác nhân chính | Quản trị viên và người dùng có tài khoản | |
| Mục đích | Người dùng hoặc quản trị viên đã đăng nhập thành công vào tài khoản tương ứng. | |
| Mức độ ưu tiên (Priority) | Bắt buộc phải có | |
| Mức | Mức 0 | |
| Điều kiện kích hoạt (Trigger) | Khi người dùng hoặc quản trị viên dùng muốn đăng nhập vào tài khoản và ấn nút Đăng Nhập | |
| Điều kiện tiên quyết (Pre-condition) | Người dùng hoặc quản trị viên phải có tài khoản | |
| Điều kiện thành công (Post-condition) | Người dùng hoặc quản trị viên đăng nhập thành công vào tài khoản của mình | |
| Điều kiện thất bại | Người dùng không đăng nhập được vào tài khoản của mình | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Người dùng nhập email và password |
| 2 | Hệ thống xác minh email và password của người dùng là đúng |
| 3 | Người dùng hoặc quản trị viên đăng nhập thành công |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Hệ thống xác minh email hoặc password bị sai | |
| 2a1. Hệ thống hiển thị thông báo và người dùng nhập lại email và password | |

Bảng 3.3. Bảng đặc tả đăng ký

|  |  |  |
| --- | --- | --- |
| Tên use case | Đăng ký | |
| Tác nhân chính | Người dùng mới | |
| Mục đích | Cho phép người dùng mới tạo tài khoản để sử dụng dịch vụ | |
| Mức độ ưu tiên (Priority) | Bắt buộc phải có | |
| Mức | Mức 0 | |
| Điều kiện kích hoạt (Trigger) | Khi người dùng mới muốn tạo tài khoản và ấn nút Đăng Ký | |
| Điều kiện tiên quyết (Pre-condition) | Người dùng mới chưa có tài khoản | |
| Điều kiện thành công (Post-condition) | Người dùng mới đăng ký tài khoản thành công và được chuyển hướng đến trang đăng nhập | |
| Điều kiện thất bại | Người dùng không thể đăng ký tài khoản | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Người dùng nhập thông tin cần thiết (fullname, email, password) |
| 2 | Hệ thống kiểm tra tính hợp lệ của thông tin |
| 3 | Hệ thống lưu thông tin người dùng và tạo tài khoản mới |
| 4 | Người dùng được chuyển hướng đến trang đăng nhập |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Hệ thống xác minh thông tin không hợp lệ (ví dụ: email đã tồn tại, password không khớp) | |
| 2a1. Hệ thống thông báo lỗi và yêu cầu người dùng cung cấp lại thông tin | |

Bảng 3.4. Bảng đặc tả đổi mật khẩu, cập nhật thông tin

|  |  |  |
| --- | --- | --- |
| Tên use case | Đổi mật khẩu, cập nhật thông tin | |
| Tác nhân chính | Người dùng có tài khoản | |
| Mục đích | Cho phép người dùng cập nhật thông tin cá nhân của mình | |
| Mức độ ưu tiên (Priority) | Không bắt buộc | |
| Mức | Mức 1 | |
| Điều kiện kích hoạt (Trigger) | Khi người dùng muốn cập nhật thông tin và ấn nút Lưu Thay Đổi | |
| Điều kiện tiên quyết (Pre-condition) | Người dùng phải đăng nhập vào tài khoản của mình và đang ở màn chỉnh sửa hồ sơ | |
| Điều kiện thành công (Post-condition) | Thông báo cập nhật thành công | |
| Điều kiện thất bại | Thông báo cập nhật thất bại | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Người dùng nhập thông tin cá nhân mới |
| 2 | Hệ thống kiểm tra tính hợp lệ của thông tin mới |
| 3 | Hệ thống cập nhật thông tin cá nhân của người dùng |
| 4 | Hệ thống hiển thị thông báo cập nhật thông tin thành công |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Dữ liệu bị bỏ trống không cập nhật được | |

Bảng 3.5. Bảng đặc tả thả cảm xúc bài đăng

|  |  |  |
| --- | --- | --- |
| Tên use case | Thả cảm xúc bài đăng | |
| Tác nhân chính | Người dùng có tài khoản | |
| Mục đích | Cho phép người dùng thể hiện cảm xúc với bài viết | |
| Mức độ ưu tiên (Priority) | Không bắt buộc | |
| Mức | Mức 1 | |
| Điều kiện kích hoạt (Trigger) | Khi người dùng muốn thả cảm xúc cho một bài viết và ấn vào biểu tượng cảm xúc | |
| Điều kiện tiên quyết (Pre-condition) | Người dùng phải đăng nhập vào tài khoản của mình | |
| Điều kiện thành công (Post-condition) | Cảm xúc của người dùng được ghi nhận thành công và hiển thị trên bài viết | |
| Điều kiện thất bại | Người dùng không thể thả cảm xúc cho bài viết | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Người dùng nhấn vào biểu tượng cảm xúc trên bài viết |
| 2 | Hệ thống ghi nhận cảm xúc của người dùng cho bài viết |
| 3 | Hệ thống cập nhật và hiển thị cảm xúc của người dùng trên các bài đăng |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Hệ thống không thể ghi nhận cảm xúc (ví dụ: lỗi kết nối) | |

Bảng 3.6. Bảng đặc tả bình luận bài đăng

|  |  |  |
| --- | --- | --- |
| Tên use case | Thả cảm xúc bài đăng | |
| Tác nhân chính | Người dùng có tài khoản | |
| Mục đích | Cho phép người dùng thêm bình luận vào bài đăng | |
| Mức độ ưu tiên (Priority) | Không bắt buộc | |
| Mức | Mức 1 | |
| Điều kiện kích hoạt (Trigger) | Khi người dùng muốn bình luận vào một bài đăng và ấn nút Bình Luận | |
| Điều kiện tiên quyết (Pre-condition) | Người dùng phải đăng nhập vào tài khoản của mình | |
| Điều kiện thành công (Post-condition) | Bình luận của người dùng được ghi nhận thành công và hiển thị dưới bài đăng | |
| Điều kiện thất bại | Người dùng không thể bình luận vào bài đăng | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Người dùng nhập nội dung bình luận vào bài đăng |
| 2 | Người dùng ấn nút Bình Luận |
| 3 | Hệ thống ghi nhận bình luận của người dùng |
| 4 | Hệ thống cập nhật và hiển thị bình luận của người dùng dưới bài đăng |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Hệ thống không thể ghi nhận bình luận (ví dụ: lỗi kết nối) | |

Bảng 3.7. Bảng đặc tả thêm mới bài đăng

|  |  |  |
| --- | --- | --- |
| Tên use case | Thêm mới bài đăng | |
| Tác nhân chính | Quản trị viên | |
| Mục đích | Cho phép quản trị viên thêm mới bài đăng trên hệ thống | |
| Mức độ ưu tiên (Priority) | Bắt buộc phải có | |
| Mức | Mức 0 | |
| Điều kiện kích hoạt (Trigger) | Khi quản trị viên muốn thêm mới một bài đăng và ấn nút Đăng Bài | |
| Điều kiện tiên quyết (Pre-condition) | Quản trị viên phải đăng nhập vào tài khoản quản trị | |
| Điều kiện thành công (Post-condition) | Bài đăng mới được thêm thành công và hiển thị trên hệ thống | |
| Điều kiện thất bại | Quản trị viên không thể thêm mới bài đăng | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Quản trị viên nhập ảnh banner, tiêu đề và nội dung bài đăng |
| 2 | Quản trị viên thêm hình ảnh, video hoặc các tài liệu đính kèm (nếu có) |
| 3 | Quản trị viên ấn nút Đăng Bài |
| 4 | Hệ thống ghi nhận thông tin bài đăng mới |
| 5 | Hệ thống cập nhật và hiển thị bài đăng mới trên hệ thống |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Hệ thống không thể ghi nhận thông tin bài đăng mới (ví dụ: lỗi kết nối) | |

Bảng 3.8. Bảng đặc tả chỉnh sửa bài đăng

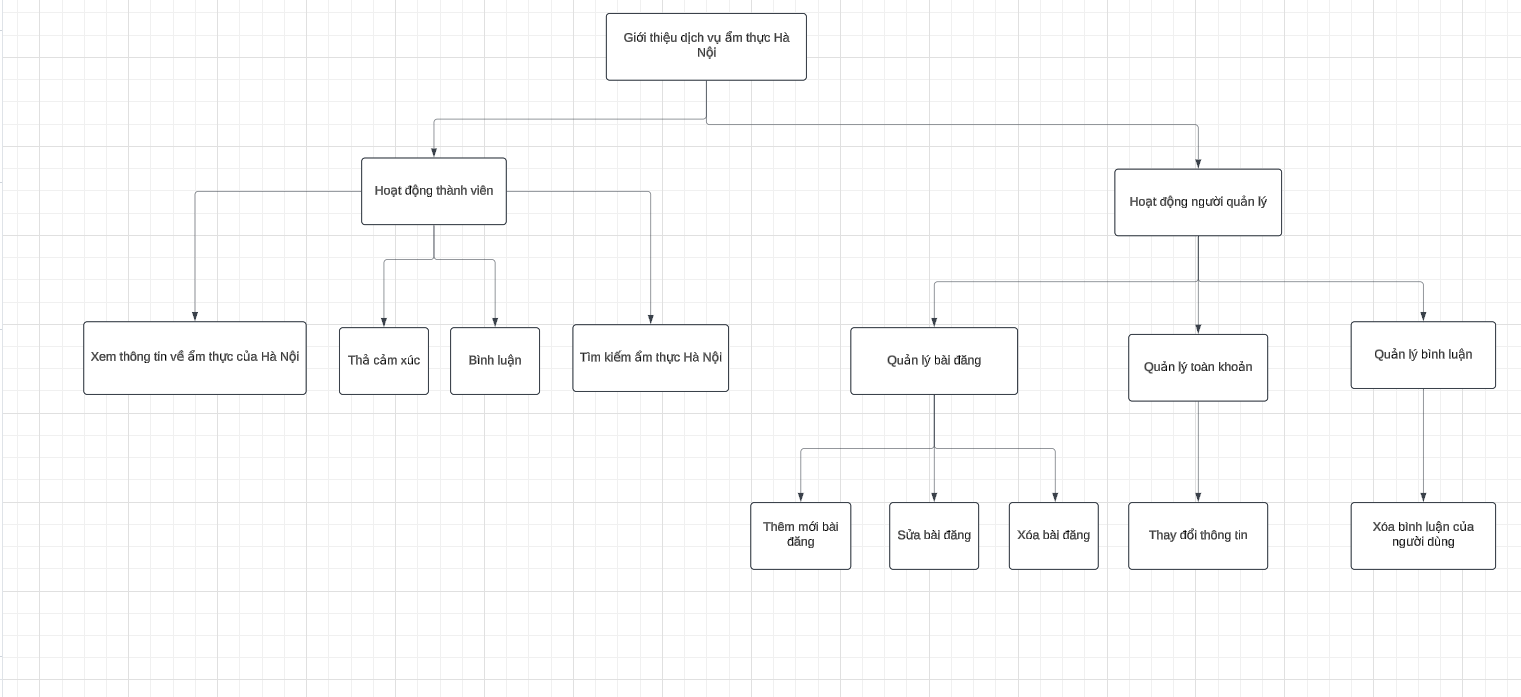
|  |  |  |
| --- | --- | --- |
| Tên use case | Chỉnh sửa bài đăng | |
| Tác nhân chính | Quản trị viên | |
| Mục đích | Cho phép quản trị viên chỉnh sửa nội dung bài đăng trên hệ thống | |
| Mức độ ưu tiên (Priority) | Không bắt buộc | |
| Mức | Mức 1 | |
| Điều kiện kích hoạt (Trigger) | Khi quản trị viên muốn chỉnh sửa bài đăng và ấn nút Lưu Chỉnh Sửa | |
| Điều kiện tiên quyết (Pre-condition) | Quản trị viên phải đăng nhập vào tài khoản quản trị và bài đăng cần chỉnh sửa đã tồn tại trên hệ thống | |
| Điều kiện thành công (Post-condition) | Bài đăng được chỉnh sửa thành công và cập nhật trên hệ thống | |
| Điều kiện thất bại | Quản trị viên không thể chỉnh sửa bài đăng | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Quản trị viên chọn bài đăng cần chỉnh sửa |
| 2 | Quản trị viên chỉnh sửa tiêu đề, nội dung bài đăng và các tài liệu đính kèm (nếu cần) |
| 3 | Quản trị viên ấn nút Lưu Chỉnh Sửa |
| 4 | Hệ thống ghi nhận thông tin chỉnh sửa mới |
| 5 | Hệ thống cập nhật và hiển thị bài đăng đã chỉnh sửa trên hệ thống |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Hệ thống không thể ghi nhận thông tin chỉnh sửa mới (ví dụ: lỗi kết nối) | |

Bảng 3.9. Bảng đặc tả xóa bài đăng

|  |  |  |
| --- | --- | --- |
| Tên use case | Xóa bài đăng | |
| Tác nhân chính | Quản trị viên | |
| Mục đích | Cho phép quản trị viên xóa bài đăng khỏi hệ thống | |
| Mức độ ưu tiên (Priority) | Không bắt buộc | |
| Mức | Mức 1 | |
| Điều kiện kích hoạt (Trigger) | Khi quản trị viên muốn xóa một bài đăng và ấn nút Xóa | |
| Điều kiện tiên quyết (Pre-condition) | Quản trị viên phải đăng nhập vào tài khoản quản trị và bài đăng cần xóa đã tồn tại trên hệ thống | |
| Điều kiện thành công (Post-condition) | Bài đăng được xóa thành công khỏi hệ thống | |
| Điều kiện thất bại | Quản trị viên không thể xóa bài đăng | |
| Luồng sự kiện chính (Basic flow) | Bước | Hành động |
| 1 | Quản trị viên chọn bài đăng cần xóa |
| 2 | Quản trị viên xác nhận việc xóa bài đăng |
| 3 | Hệ thống ghi nhận yêu cầu xóa bài đăng |
| 4 | Hệ thống xóa bài đăng khỏi hệ thống |
| 5 | Hệ thống hiển thị thông báo xóa bài đăng thành công |
| Luồng sự kiện ngoại lệ (Exception Flow) | 2a. Hệ thống không thể xóa bài đăng (ví dụ: lỗi kết nối) | |

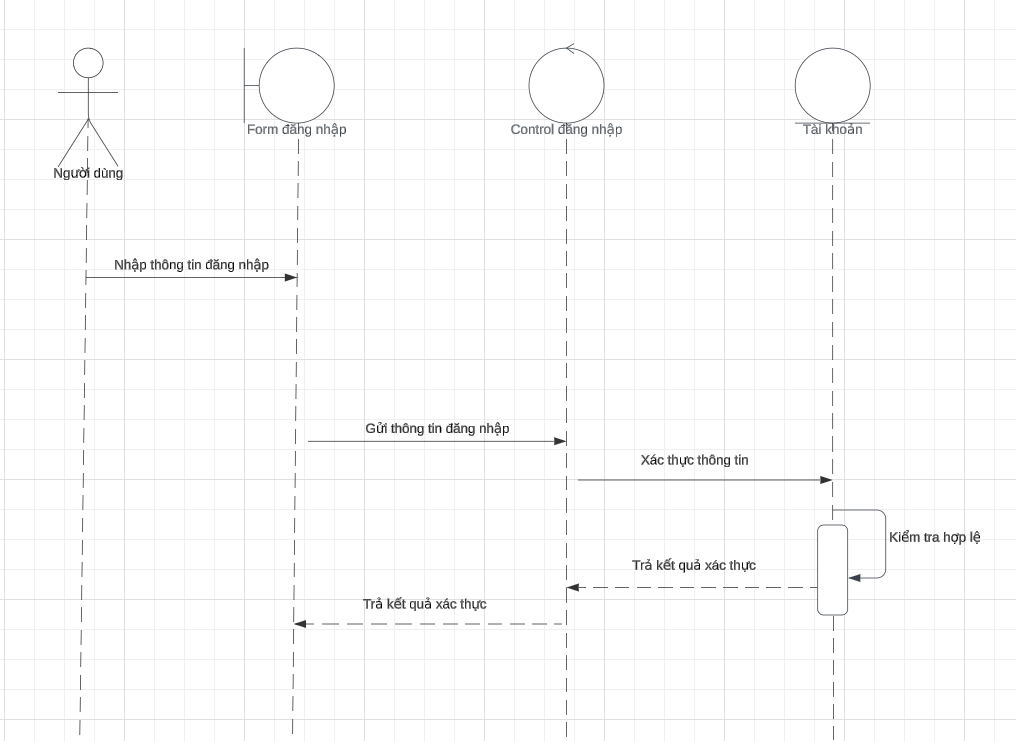
## **3.1.5 Biểu đồ phân rã chức năng**

Mô hình phân tách chức năng (BFD - Business Function Diagram) là một công cụ để biểu diễn việc phân tách một cách đơn giản các nhiệm vụ cần thực hiện. Mỗi nhiệm vụ được chia thành nhiều nhiệm vụ con, mức độ phân chia này sẽ phụ thuộc vào kích thước và độ phức tạp của hệ thống.

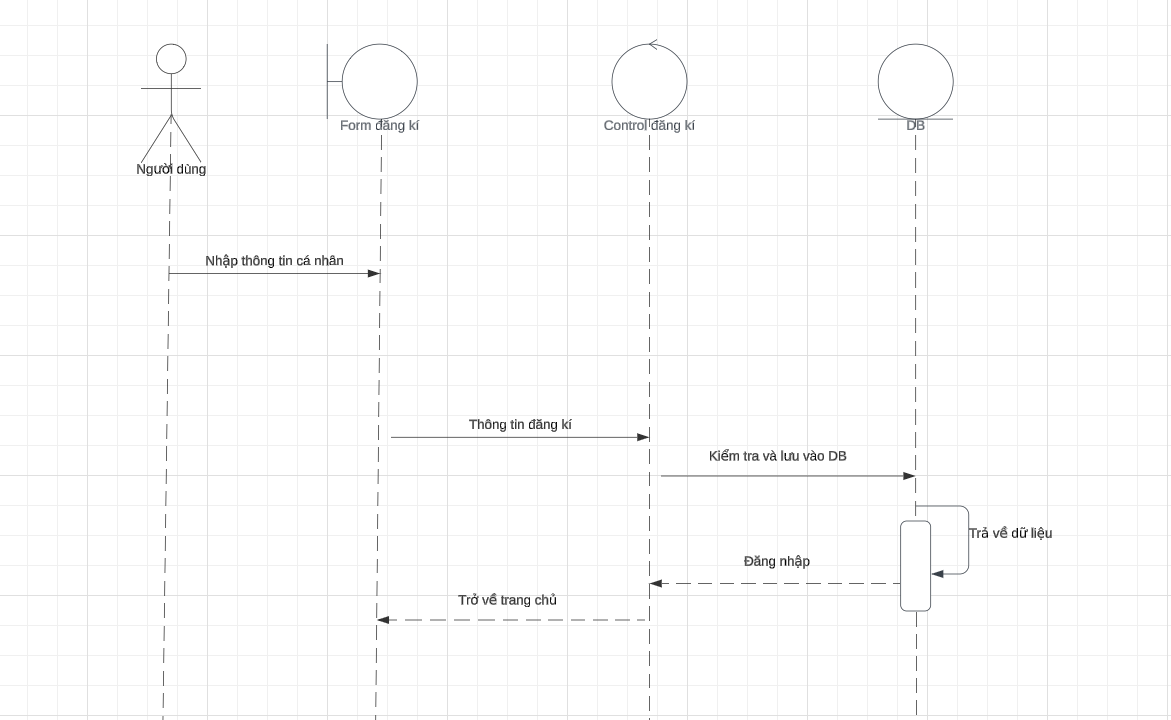


Hình 3. 4. Biểu đồ phân rã chức năng

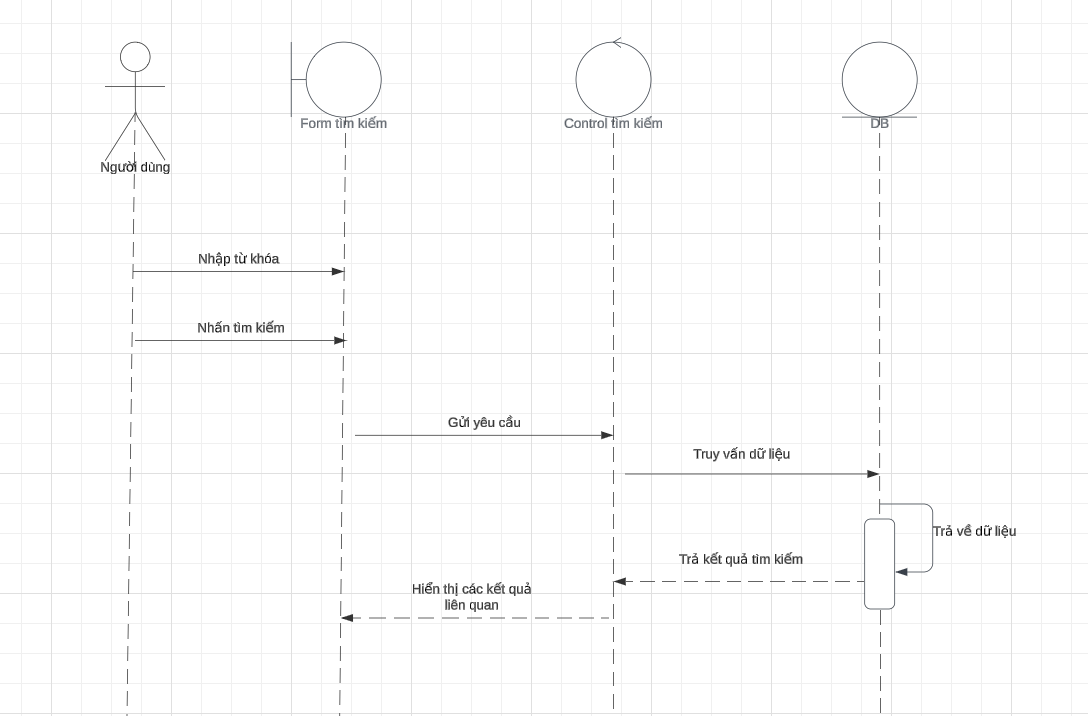
## **3.1.6 Biểu đồ tuần tự**



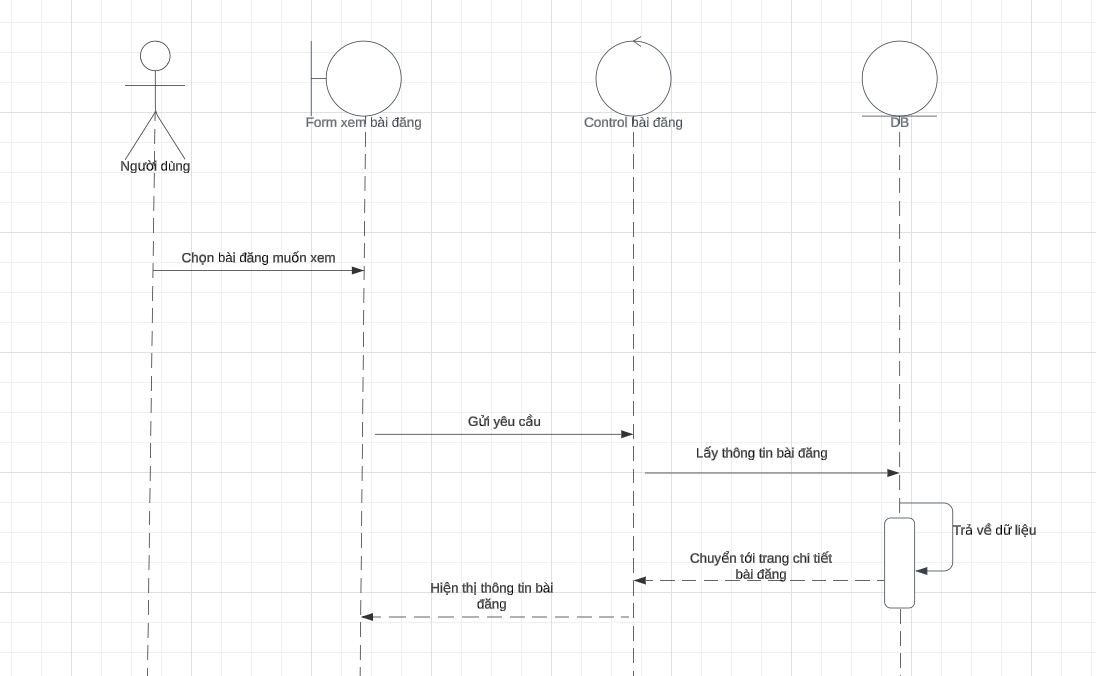
Hình 3. 5. Biểu đồ tuần tự đăng nhập cho người dùng hoặc quản trị viên

****

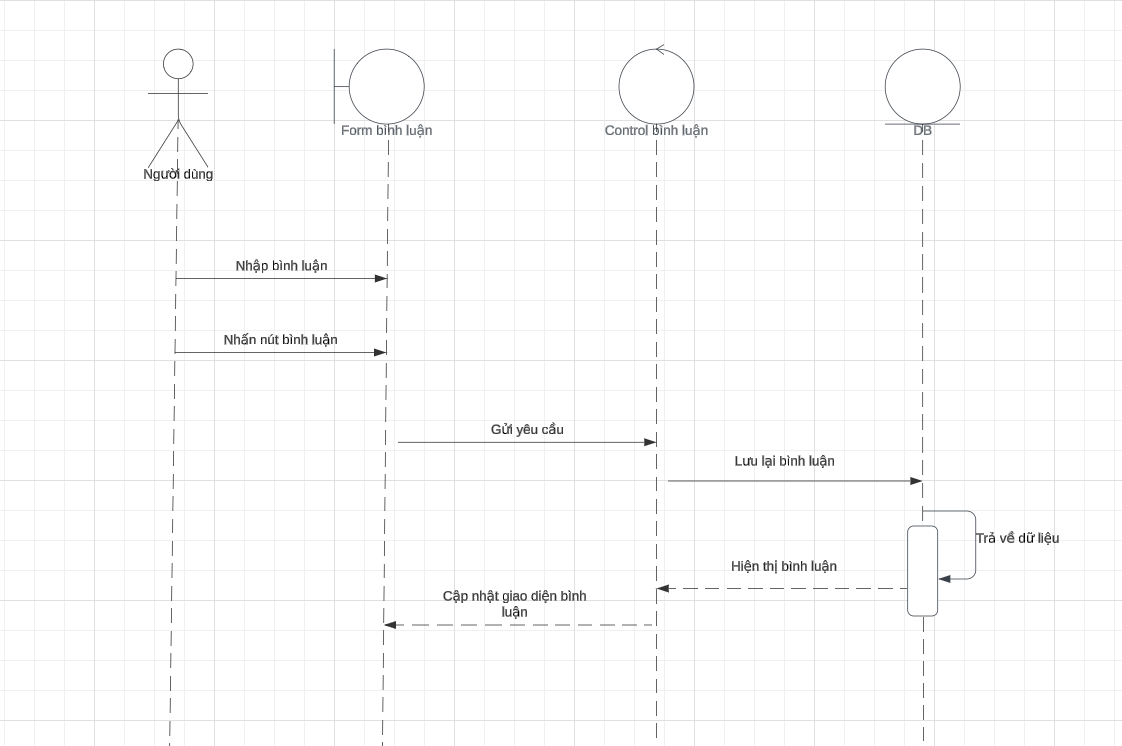
Hình 3. 6. Biểu đồ tuần tự đăng ký thành viên cho người dùng

****

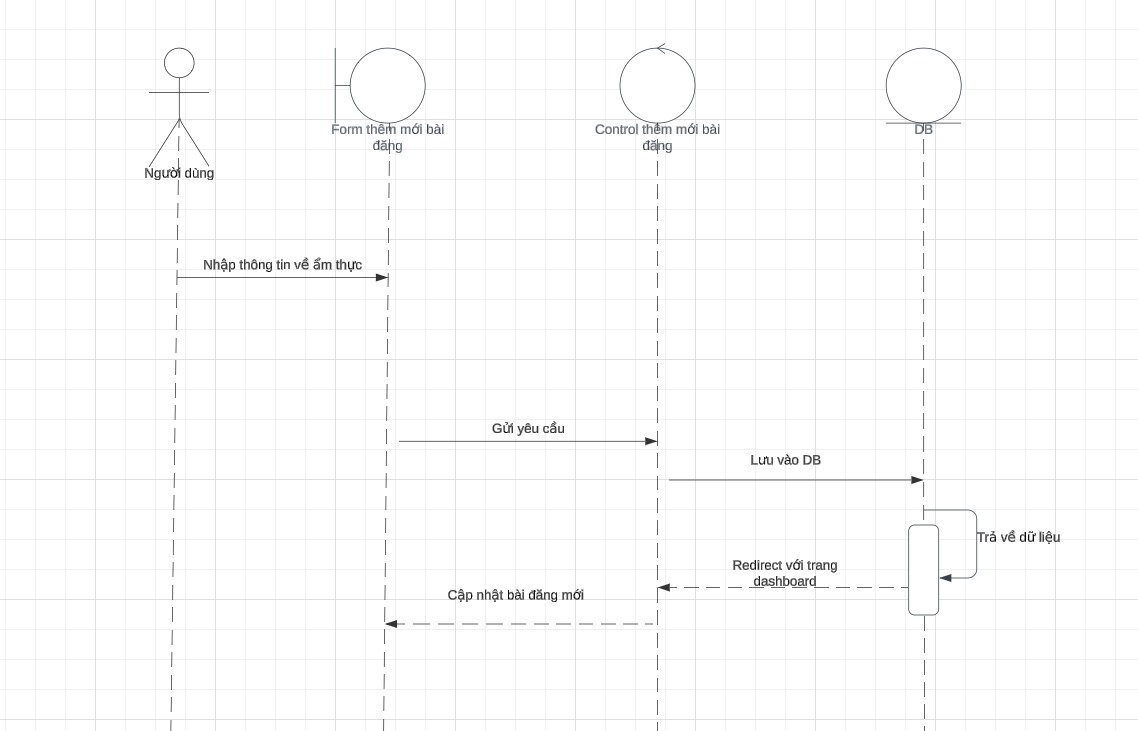
Hình 3. 7. Biểu đồ tuần tự tìm kiếm thông tin các món ăn

****

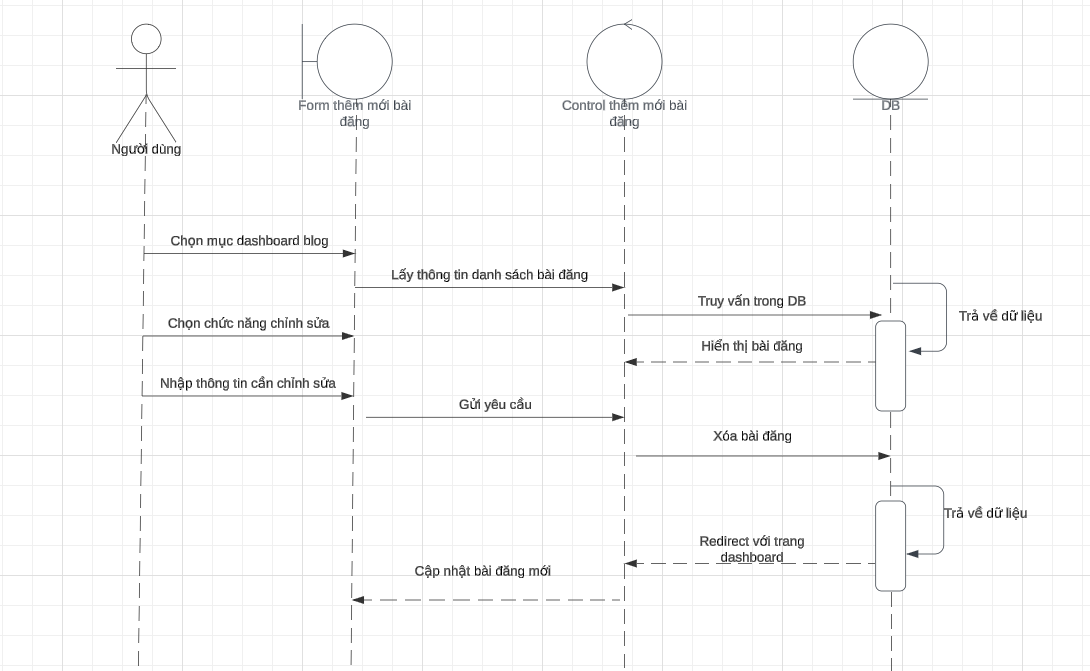
Hình 3. 8. Biểu đồ tuần tự xem thông tin chi tiết

****

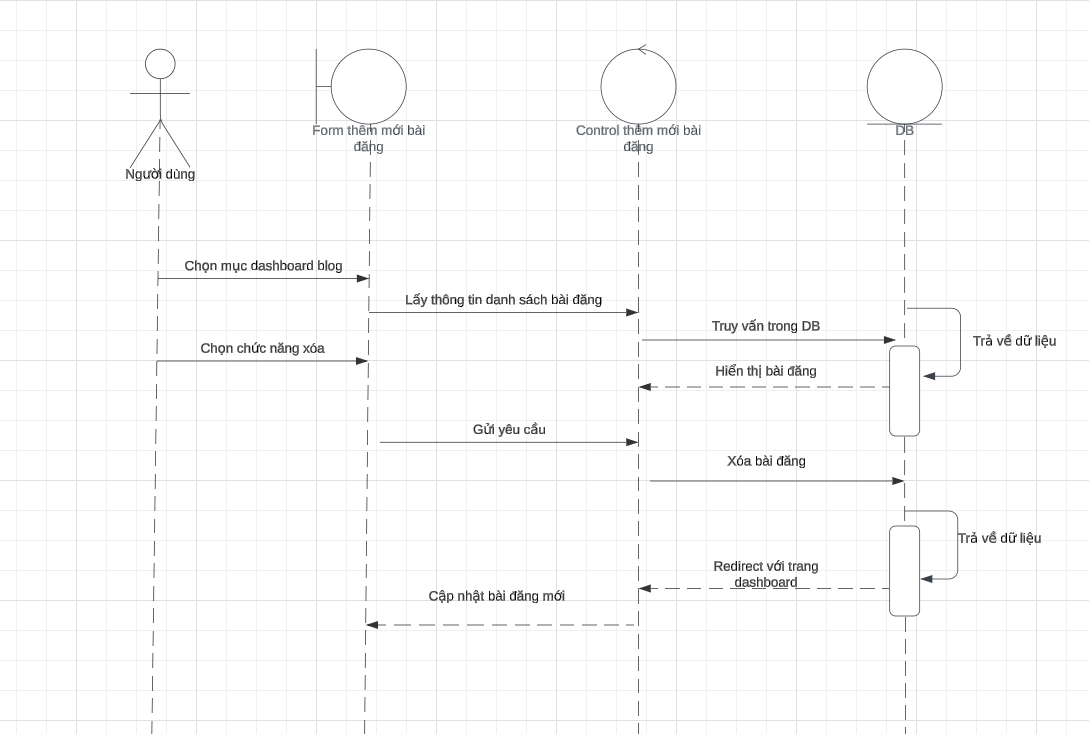
Hình 3. 9. Biểu đồ tuần tự bình luận bài viết

****

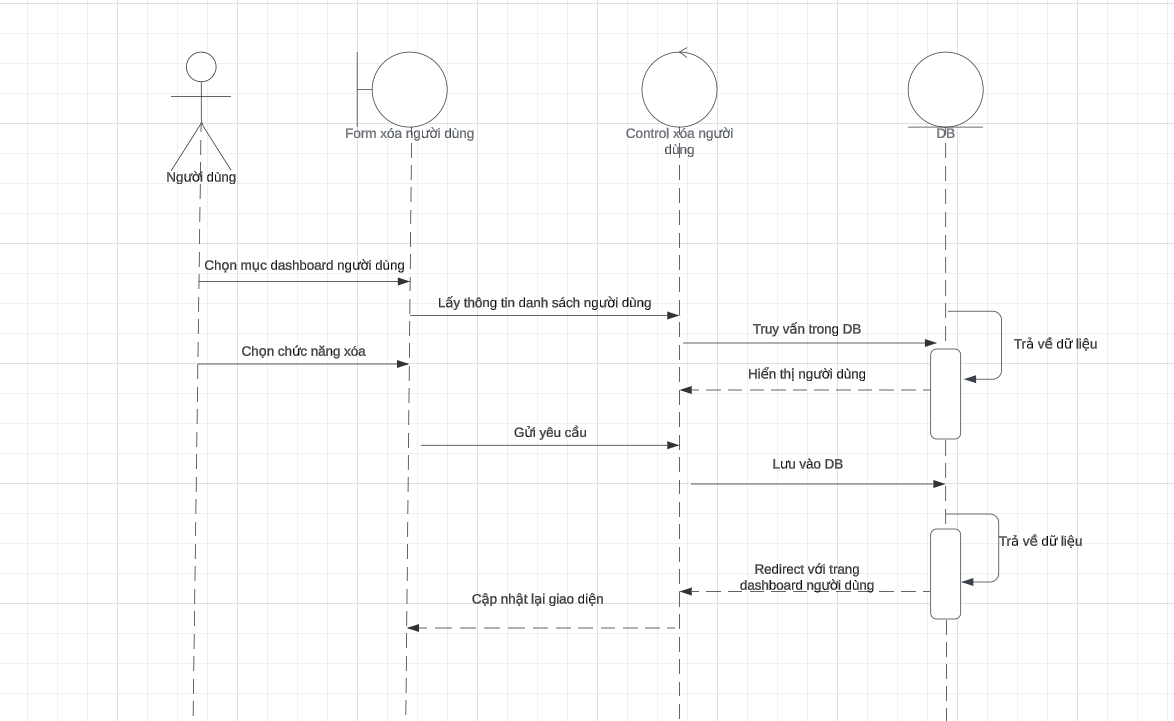
Hình 3. 10. Biểu đồ tuần tự thêm mới nội dung về ẩm thực

****

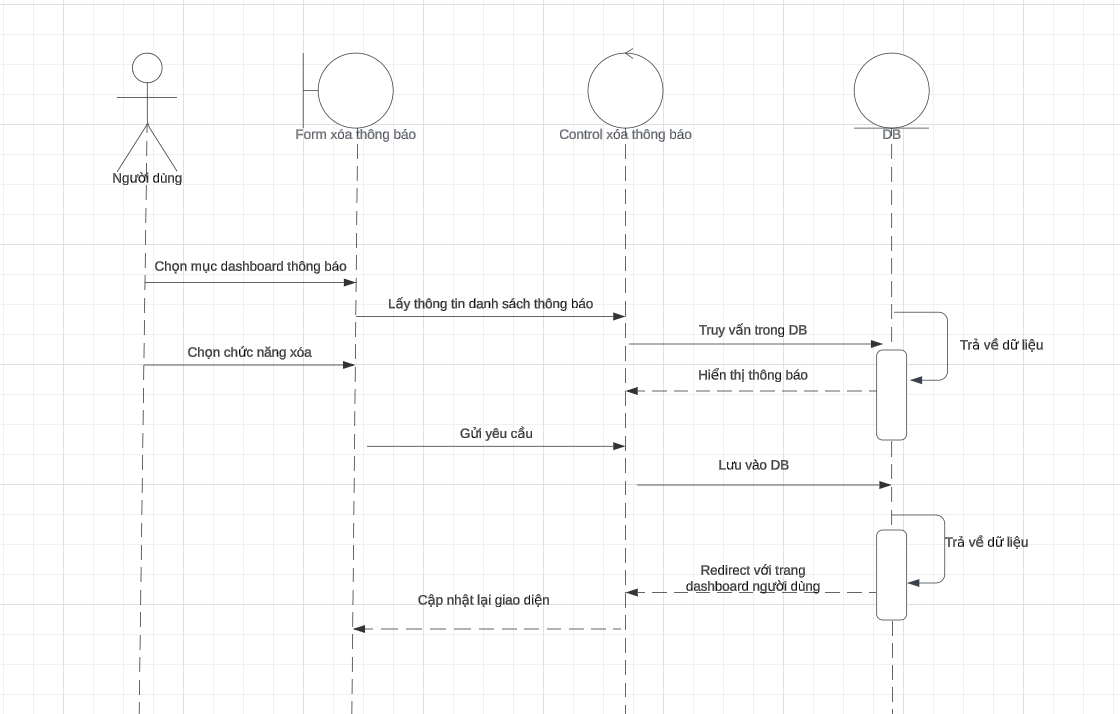
Hình 2. 19. Biểu đồ tuần tự chỉnh sửa nội dung về ẩm thực

****

Hình 3. 11. Biểu đồ tuần tự xóa nội dung về ẩm thực

****

Hình 3. 12. Biểu đồ tuần tự xóa tài khoản người dùng



Hình 3. 13. Biểu đồ tuần tự xóa thông báo về cảm xúc và bình luận