**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**A logo with a person holding a book

AI-generated content may be incorrect.**

**ĐỒ ÁN CHUYÊN NGÀNH**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG ỨNG DỤNG**

**ĐẶT ĐỒ ĂN TRÊN ĐIỆN THOẠI VÀ TÍCH HỢP AI**

Giảng viên hướng dẫn: **ThS. Từ Thanh Trí**

Sinh viên thực hiện: **Hồ Huỳnh Bảo Thiên**

Mã số sinh viên: 1050080158

Lớp: **10\_ĐH\_CNPM2**

Khoá: **2021 - 2025**

**TP. Hồ Chí Minh, tháng 09 năm 2025**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**A logo with a person holding a book

AI-generated content may be incorrect.**

**ĐỒ ÁN CHUYÊN NGÀNH**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG ỨNG DỤNG   
ĐẶT ĐỒ ĂN TRÊN ĐIỆN THOẠI VÀ TÍCH HỢP AI**

Giảng viên hướng dẫn: **ThS. Từ Thanh Trí**

Sinh viên thực hiện: **Hồ Huỳnh Bảo Thiên**

Mã số sinh viên: 1050080158

Lớp: **10\_ĐH\_CNPM2**

Khoá: **2021 – 2025**

**TP. Hồ Chí Minh, tháng 09 năm 2025**

**LỜI CẢM ƠN**

Trước hết, em xin gửi lời cảm ơn chân thành đến tập thể quý thầy Cô Trường Đại học Tài nguyên và Môi trường TP.HCM cùng quý Thầy Cô khoa Công Nghệ Thông Tin đã tạo điều kiện giúp em trang bị những kiến thức cơ bản vững chắc, làm nền tảng cho việc thực hiện đề tài này.

Đặc biệt, em xin tri ân và bày tỏ lòng biết ơn sâu sắc nhất tới Thầy ThS. Từ Thanh Trí, người đã tận tình hỗ trợ và dẫn dắt em hoàn thành báo cáo đồ án chuyên ngành của mình. Trong suốt một học kỳ thực hiện đề tài, em đã không ngừng vận dụng những kiến thức nền tảng đã tích lũy, cùng với việc học hỏi và nghiên cứu thêm những kiến thức mới. Từ đó, em đã cố gắng tối đa hóa việc áp dụng những gì đã thu thập được để hoàn thiện báo cáo đồ án một cách tốt nhất có thể.

Tuy nhiên, em cũng ý thức rằng trong quá trình thực hiện, khó tránh khỏi những thiếu sót. Vì thế, em rất mong nhận được những góp ý quý báu từ phía các Thầy Cô, nhằm hoàn thiện hơn nữa kiến thức mà em đã học tập, đồng thời làm hành trang cho những đề tài nghiên cứu tiếp theo trong tương lai.

Em xin chân thành cảm ơn quý Thầy Cô!

**LỜI CAM ĐOAN**

Em, Hồ Huỳnh Bảo Thiên, sinh viên lớp 10 – ĐH – CNPM2, khoa công nghệ thông tin, Trường Đại học Tài nguyên và Môi trường TP.HCM, xin cam đoan rằng công trình nghiên cứu này là kết quả của sự nỗ lực và sáng tạo cá nhân dưới sự hướng dẫn của thầy ThS. Từ Thanh Trí. Em khẳng định rằng tất cả thông tin, dữ liệu và kết quả trình bày trong báo cáo đều được thực hiện một cách trung thực và độc lập, không sao chép từ bất kỳ nguồn nào mà không có trích dẫn hợp lệ. Em chịu hoàn toàn trách nhiệm về tính chính xác và đáng tin cậy của nội dung trong báo cáo này. Nếu có bất kỳ vi phạm nào liên quan đến tính trung thực của công trình, em sẵn sàng chịu mọi hình thức kỷ luật theo quy định của nhà trường.

Trân trọng,

**Sinh viên thực hiện**

(ký tên và ghi rõ họ tên)

**NHẬN XÉT**

**(giảng viên hướng dẫn)**

**Kết luận: (Đồng ý hoặc Không đồng ý cho sinh viên nộp báo cáo)**

TP. Hồ Chí Minh, ngày… tháng…. năm 2025

**GIẢNG VIÊN HƯỚNG DẪN**

(Ký và ghi rõ học hàm, học vị họ tên)

**Ths. Từ Thanh**

**NHẬN XÉT**

**(giảng viên phản biện)**

**Kết luận: (Đồng ý hoặc Không đồng ý cho sinh viên nộp báo cáo)**

TP. Hồ Chí Minh, ngày… tháng…. năm 2025

**GIẢNG VIÊN PHẢN BIỆN**

(Ký và ghi rõ học hàm, học vị họ tên)

**LỊCH LÀM VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tuần | Nội dung thực hiện | Tự nhận xét về mức độ hoàn thành | Nhận xét của  GV hướng dẫn | Chữ ký của  GV hướng dẫn |
| **1**  (25/08 – 31/08) | Đăng ký giảng viên hướng dẫn và chọn đề tài | 100% |  |  |
| **2**  (01/09 – 07/09) | Phân tích và thiết kế hệ thống | 100% |  |  |
| **3**  (08/09 – 14/09) | Xây dựng giao diện ứng dụng | 100% |  |  |
| **4**  (15/09 – 21/09) | Xây dựng giao diện ứng dụng | 100% |  |  |
| **5**  (22/09-28/09) | Xây dựng chức năng | 100% |  |  |
| **6**  (29/09 – 05/10) | Xây dựng chức năng | 100% |  |  |
| **7**  (06/10 – 12/10) | Xây dựng chức năng | 100% |  |  |
| **8**  (13/10 – 19/10) | Kiểm thử và sửa lỗi | 80% |  |  |
| **9**  (20/10 – 26/10) | Kiểm thử và sửa lỗi | 100% |  |  |
| **10**  (27/10 – 02/11) | Kiểm thử và sửa lỗi | 100% |  |  |
| **11**  (03/11– 16/11) | Viết báo cáo | 100% |  |  |
| **12**  (17/11 – 23/11) | Hoàn thành | 100% |  |  |

**TÓM TẮT ĐỒ ÁN**

Đồ án xây dựng ứng dụng đặt đồ ăn tích hợp AI nhằm mang lại trải nghiệm tiện lợi, nhanh chóng và cá nhân hóa cho người dùng trên nền tảng thiết bị di động. Ứng dụng được phát triển đa nền tảng bằng Flutter với ngôn ngữ lập trình Dart, cho phép triển khai trên cả Android và iOS từ một mã nguồn duy nhất.

Hệ thống phía máy chủ backend được xây dựng bằng ASP.NET Core Web API, kết nối với cơ sở dữ liệu SQL Server để lưu trữ thông tin người dùng, đơn hàng và dữ liệu món ăn. Dữ liệu được quản lý thông qua Entity Framework Core, giúp truy xuất dữ liệu linh hoạt và hiệu quả.

Điểm nổi bật của đồ án là tích hợp chatbot AI sử dụng mô hình Gemini 2.0 Flash. Chatbot có khả năng hiểu ngôn ngữ tự nhiên để hỗ trợ người dùng tìm kiếm món ăn, tư vấn.

Ngoài ra, hệ thống còn áp dụng các thuật toán gợi ý món ăn dựa trên phân tích lịch sử đặt món, đánh giá người dùng và mô hình lọc cộng tác để đưa ra đề xuất phù hợp. Các công cụ hỗ trợ phát triển như Git quản lý mã nguồn, Android Studio môi trường phát triển và Postman kiểm thử API cũng được sử dụng xuyên suốt quá trình thực hiện.

Đồ án không chỉ cung cấp giải pháp đặt món ăn thông minh, mà còn thể hiện khả năng tích hợp AI vào ứng dụng thực tế, góp phần nâng cao trải nghiệm người dùng và tối ưu hóa dịch vụ trong lĩnh vực ẩm thực trực tuyến.

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc197583320)

[CHƯƠNG 1: TỔNG QUAN TÀI LIỆU 2](#_Toc197583321)

[1.1 Các công việc và dự án liên quan 2](#_Toc197583322)

[1.2 Cơ sở lý thuyết và kỹ thuật 3](#_Toc197583323)

[1.2.1 Cơ sở lý thuyết 3](#_Toc197583324)

[1.2.2 Kỹ thuật 9](#_Toc197583325)

[CHƯƠNG 2: PHƯƠNG PHÁP THỰC HIỆN 10](#_Toc197583326)

[2.1 Mô hình Usecase 11](#_Toc197583327)

[2.1.1 Đặc tả usecase “đăng nhập” 11](#_Toc197583328)

[2.1.2 Đặc tả usecase “đăng ký” 12](#_Toc197583329)

[2.1.3 Đặc tả usecase “đặt món” 13](#_Toc197583330)

[2.1.4 Đặc tả usecase “đánh giá” 14](#_Toc197583331)

[2.1.5 Đặc tả usecase “thanh toán” 14](#_Toc197583332)

[2.1.6 Đặc tả usecase “quản lý đơn hàng” 15](#_Toc197583333)

[2.1.7 Đặc tả usecase “quản lý món ăn” 16](#_Toc197583334)

[2.1.8 Đặc tả usecase “quản lý nhà hàng” 18](#_Toc197583335)

[2.1.9 Đặc tả usecase “ChatBotAI” 19](#_Toc197583336)

[2.2 Sơ đồ lớp 20](#_Toc197583337)

[2.3 Sơ đồ trạng thái 21](#_Toc197583338)

[2.4 Sơ đồ luồng 23](#_Toc197583339)

[2.5 Sơ đồ tuần tự 26](#_Toc197583340)

[2.5.1 Sơ đồ tuần tự “đăng nhập” 26](#_Toc197583341)

[2.5.2 Sơ đồ tuần tự “đăng ký” 26](#_Toc197583342)

[2.5.3 Sơ đồ tuần tự “đặt món” 27](#_Toc197583343)

[2.5.4 Sơ đồ tuần tự “thanh toán” 28](#_Toc197583344)

[2.5.5 Sơ đồ tuần tự “đơn hàng” 29](#_Toc197583345)

[2.5.6 Sơ đồ tuần tự “món ăn” 30](#_Toc197583346)

[2.5.7 Sơ đồ tuần tự “nhà hàng” 31](#_Toc197583347)

[2.5.8 Sơ đồ tuần tự “đánh giá” 32](#_Toc197583348)

[2.5.9 Sơ đồ tuần tự “ChatBotAI” 33](#_Toc197583349)

[2.5 Thiết kế cơ sở dữ liệu 34](#_Toc197583350)

[2.5.1 Mô tả chi tiết “NguoiDung” 34](#_Toc197583351)

[2.5.2 Mô tả chi tiết “NhaHang” 35](#_Toc197583352)

[2.5.3 Mô tả chi tiết “MonAn” 36](#_Toc197583353)

[2.5.4 Mô tả chi tiết “DonHang” 37](#_Toc197583354)

[2.5.5 Mô tả chi tiết “ChiTietDonHang” 38](#_Toc197583355)

[2.5.6 Mô tả chi tiết “GioHang” 39](#_Toc197583356)

[2.5.7 Mô tả chi tiết “ThanhToan” 39](#_Toc197583357)

[2.5.8 Mô tả chi tiết “DanhGia” 40](#_Toc197583358)

[2.5.9 Mô tả chi tiết “ChatHistory” 41](#_Toc197583359)

[CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM 43](#_Toc197583360)

[3.1 Mô tả bài toán 43](#_Toc197583361)

[3.2 Quá trình thực hiện 43](#_Toc197583362)

[3.3 Kết quả đạt được 45](#_Toc197583363)

[3.3.1 Giao diện đăng nhập 45](#_Toc197583364)

[3.3.2 Giao diện đăng ký 46](#_Toc197583365)

[3.3.3 Giao diện trang chủ 47](#_Toc197583366)

[3.3.4 Giao diện món ăn 48](#_Toc197583367)

[3.3.5 Giao diện chi tiết món ăn 49](#_Toc197583368)

[3.3.6 Giao diện giỏ hàng 50](#_Toc197583369)

[3.3.7 Giao diện thanh toán 51](#_Toc197583370)

[3.3.9 Giao diện đơn hàng 52](#_Toc197583371)

[3.3.10 Giao diện thông tin cá nhân 53](#_Toc197583372)

[3.3.11 Giao diện hộp thoại chat bot 54](#_Toc197583373)

[CHƯƠNG 4: KẾT LUẬN 55](#_Toc197583374)

[4.1 Đánh giá kết quả đạt được 55](#_Toc197583375)

[4.2 Kiến thức lý thuyết đã được củng cố 55](#_Toc197583376)

[4.3 Kỹ năng thực hành đã học được 55](#_Toc197583377)

[4.4 Kinh nghiệm thực tiễn tích lũy được 56](#_Toc197583378)

[4.5 Thảo luận và các vấn đề chưa giải quyết 56](#_Toc197583379)

[4.6 Kết luận 56](#_Toc197583380)

[4.7 Đề nghị rút ra từ kết quả đề tài 57](#_Toc197583381)

[4.8 Hướng phát triển tương lai 57](#_Toc197583382)

[TÀI LIỆU THAM KHẢO 58](#_Toc197583383)

**DANH MỤC CÁC HÌNH ẢNH**

[Hình 1: ngôn ngữ dart 3](#_Toc197583430)

[Hình 2: flutter logo 4](#_Toc197583431)

[Hình 3: Ngôn ngữ C# 5](#_Toc197583432)

[Hình 4: asp.net core web api 6](#_Toc197583433)

[Hình 5: Microsoft SQL Server 6](#_Toc197583434)

[Hình 6: mô hình usecase 11](#_Toc197583435)

[Hình 7: sơ đồ lớp 20](#_Toc197583436)

[Hình 8: sơ đồ trạng thái đặt hàng 21](#_Toc197583437)

[Hình 9: sơ đồ trạng thái thanh thoán 22](#_Toc197583438)

[Hình 10: sơ đồ trạng thái chat bot AI 23](#_Toc197583439)

[Hình 11: sơ đồ luồng 25](#_Toc197583440)

[Hình 12: sơ đồ tuần tự đăng nhập 26](#_Toc197583441)

[Hình 13: sơ đồ tuần tự đăng ký 26](#_Toc197583442)

[Hình 14: sơ đồ tuần tự đặt món 27](#_Toc197583443)

[Hình 15: sơ đồ tuần tự thanh toán 28](#_Toc197583444)

[Hình 16: sơ đồ tuần tự đơn hàng 29](#_Toc197583445)

[Hình 17: sơ đồ tuần tự món ăn 30](#_Toc197583446)

[Hình 18: sơ đồ tuần tự nhà hàng 31](#_Toc197583447)

[Hình 19: sơ đồ tuần tự đánh giá 32](#_Toc197583448)

[Hình 20: sơ đồ tuần tự chatbotAI 33](#_Toc197583449)

[Hình 21: cơ sở dữ liệu 34](#_Toc197583450)

[Hình 22: giao diện đăng nhập 45](#_Toc197583451)

[Hình 23: giao diện đăng ký 46](#_Toc197583452)

[Hình 24: giao diện trang chủ 47](#_Toc197583453)

[Hình 25: giao diện món ăn 48](#_Toc197583454)

[Hình 26: giao diện chi tiết món ăn 49](#_Toc197583455)

[Hình 27: giao diện giỏ hàng 50](#_Toc197583456)

[Hình 28: giao diện thanh toán 51](#_Toc197583457)

[Hình 29: giao diện đơn hàng 52](#_Toc197583458)

[Hình 30: giao diện thông tin cá nhân 53](#_Toc197583459)

[Hình 31: giao diện hộp thoại chat bot AI 54](#_Toc197583460)

**DANH MỤC CÁC BẢNG**

[Bảng 1: đặc tả usecase đăng nhập 12](#_Toc197583477)

[Bảng 2: đặc tả usecase đăng ký 13](#_Toc197583478)

[Bảng 3: đặc tả usecase đặt món 13](#_Toc197583479)

[Bảng 4: đặc tả usecase đánh giá 14](#_Toc197583480)

[Bảng 5: đặc tả usecase thanh toán 15](#_Toc197583481)

[Bảng 6: đặc tả usecase quản lý đơn hàng 16](#_Toc197583482)

[Bảng 7: đặc tả usecase quản lý món ăn 18](#_Toc197583483)

[Bảng 8: đặc tả usecase quản lý nhà hàng 19](#_Toc197583484)

[Bảng 9: đặc tả usecase chatbotAI 20](#_Toc197583485)

[Bảng 10: mô tả chi tiết người dùng 35](#_Toc197583486)

[Bảng 11: mô tả chi tiết nhà hàng 36](#_Toc197583487)

[Bảng 12: mô tả chi tiết món ăn 37](#_Toc197583488)

[Bảng 13: mô tả chi tiết đơn hàng 38](#_Toc197583489)

[Bảng 14: mô tả chi tiết chi tiết đơn hàng 39](#_Toc197583490)

[Bảng 15: mô tả chi tiết giỏ hàng 39](#_Toc197583491)

[Bảng 16: mô tả chi tiết thanh toán 40](#_Toc197583492)

[Bảng 17: mô tả chi tiết đánh giá 41](#_Toc197583493)

[Bảng 18: mô tả chi tiết chathistory 42](#_Toc197583494)

LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như ngày nay, cách chúng ta đặt đồ ăn đã thay đổi đáng kể nhờ những tiện ích mà công nghệ mang lại. Chỉ với một chiếc điện thoại thông minh, người dùng có thể dễ dàng chọn món ăn yêu thích mà không cần rời khỏi nhà hay mất thời gian gọi điện cho từng quán. Xu hướng này không chỉ đáp ứng nhu cầu tiện lợi của con người mà còn mở ra cơ hội phát triển cho các nhà hàng và doanh nghiệp công nghệ. Việc xây dựng một ứng dụng đặt đồ ăn trên điện thoại tích hợp trí AI chính là một bước tiến quan trọng, để nâng cao trải nghiệm người dùng.

Ứng dụng đặt đồ ăn trên điện thoại mang lại nhiều lợi ích thiết thực cho cả người dùng và nhà hàng. Trước hết, đối với người dùng, sự tiện lợi là yếu tố nổi bật nhất. Chỉ cần vài thao tác đơn giản, bạn có thể lướt qua thực đơn món ăn, xem giá cả, đọc đánh giá và đặt món mà không cần di chuyển.

Việc tích hợp AI vào ứng dụng đặt đồ ăn không chỉ dừng lại ở việc xử lý đơn hàng mà còn mang đến những trải nghiệm thông minh và tối ưu hóa vượt trội. AI có thể phân tích sở thích cá nhân để đưa ra gợi ý phù hợp hoặc tìm kiếm mòn ăn mà khách hàng yêu cầu.

CHƯƠNG 1: TỔNG QUAN TÀI LIỆU

1.1 Các công việc và dự án liên quan

GrabFood: GrabFood là dịch vụ đặt đồ ăn thuộc hệ sinh thái Grab, cho phép người dùng chọn món từ nhiều nhà hàng và nhận giao hàng tận nơi. Ứng dụng sử dụng định vị GPS, hỗ trợ thanh toán qua GrabPay và theo dõi đơn hàng theo thời gian thực. Ưu điểm: Số lượng nhà hàng lớn, tích hợp tốt với dịch vụ vận chuyển. Hạn chế: Phí dịch vụ cao, tập trung chủ yếu ở khu vực đô thị.

ShopeeFood: ShopeeFood (tiền thân là Now) cung cấp dịch vụ đặt đồ ăn với nhiều chương trình khuyến mãi, tích hợp thanh toán qua ví ShopeePay. Ưu điểm: Giao diện thân thiện, nhiều ưu đãi hấp dẫn. Hạn chế: Phạm vi hoạt động chưa rộng, chưa tối ưu ở vùng sâu vùng xa.

Baemin: Ứng dụng đến từ Hàn Quốc này nổi bật với giao diện trẻ trung và dịch vụ giao hàng nhanh. Ưu điểm: Thiết kế bắt mắt, phù hợp với người dùng trẻ. Hạn chế: Số lượng nhà hàng đối tác còn hạn chế ở một số khu vực.

Foodpanda: Foodpanda là ứng dụng quốc tế, hoạt động ở nhiều quốc gia, cung cấp đa dạng lựa chọn nhà hàng và món ăn. Ưu điểm: Quy mô lớn, hỗ trợ đa ngôn ngữ. Hạn chế: Chi phí vận hành cao, cạnh tranh lớn với các ứng dụng địa phương.

Những ứng dụng này đều cung cấp các tính năng cơ bản như xem thực đơn, đặt hàng và thanh toán trực tuyến. Tuy nhiên, chúng chủ yếu tập trung vào tối ưu hóa quy trình giao dịch mà chưa khai thác sâu các công nghệ AI để hỗ trợ người dùng trong việc tìm kiếm món ăn theo sở thích hoặc tận dụng dữ liệu khuyến mãi một cách thông minh.

Đề tài của em có điểm tương đồng với các dự án trên ở việc cung cấp nền tảng đặt đồ ăn tiện lợi. Tuy nhiên, sự khác biệt nằm ở việc tích hợp AI với mục tiêu cụ thể:

* Tìm kiếm món ăn theo số lượng đơn đặt nhiều người dùng.
* Liệt kê món ăn hiện có.

Những tính năng này không chỉ nâng cao trải nghiệm người dùng mà còn tạo sự khác biệt so với các ứng dụng hiện có.

1.2 Cơ sở lý thuyết và kỹ thuật

1.2.1 Cơ sở lý thuyết

1.2.1.1 Ngôn ngữ Dart

Dart, là một ngôn ngữ lập trình đa mục đích mã nguồn mở, đã được đặt nền móng bởi Google. Đây là một khía cạnh của ngôn ngữ lập trình hướng đối tượng, đặc trưng bởi cú pháp kiểu C. Tinh thần lập trình hướng đối tượng thể hiện qua việc hỗ trợ giao diện và lớp, mở ra khả năng sáng tạo không giới hạn khi đặt lên bàn cân với những ngôn ngữ khác. Sự đa dạng của Dart thể hiện thông qua khả năng phát triển ứng dụng web, di động, máy chủ và máy tính để bàn[1].



Hình 1: ngôn ngữ dart

1.2.1.2 Giới thiệu flutter

Flutter là một framework mã nguồn mở do Google phát triển, được thiết kế để xây dựng các ứng dụng giao diện người dùng (UI) đẹp mắt và hiệu suất cao trên nhiều nền tảng từ một cơ sở mã duy nhất. Với Flutter, lập trình viên có thể phát triển ứng dụng cho Android, iOS, web, desktop (Windows, macOS, Linux) mà không cần viết mã riêng biệt cho từng nền tảng[2].

Một trong những điểm nổi bật của Flutter là việc sử dụng ngôn ngữ Dart – ngôn ngữ tối ưu cho việc xây dựng giao diện người dùng hiện đại. Flutter cung cấp một bộ widget phong phú và có khả năng tùy biến cao, giúp tạo nên giao diện hấp dẫn, linh hoạt theo chuẩn Material Design của Google hoặc Cupertino của Apple.

Flutter còn nổi bật với tính năng "hot reload", cho phép lập trình viên thấy ngay thay đổi mà không cần biên dịch lại toàn bộ ứng dụng, từ đó tăng tốc quá trình phát triển và thử nghiệm.

Với cộng đồng phát triển ngày càng lớn mạnh và được sử dụng trong nhiều dự án thực tế, Flutter đang trở thành một trong những công nghệ phát triển ứng dụng đa nền tảng phổ biến nhất hiện nay.



Hình 2: flutter

1.2.1.3 Ngôn ngữ C#

C#:(hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java. C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng[3].

C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).

A logo of a company

Description automatically generated

Hình 3: Ngôn ngữ C#

1.2.1.4 ASP.NET Core Web API

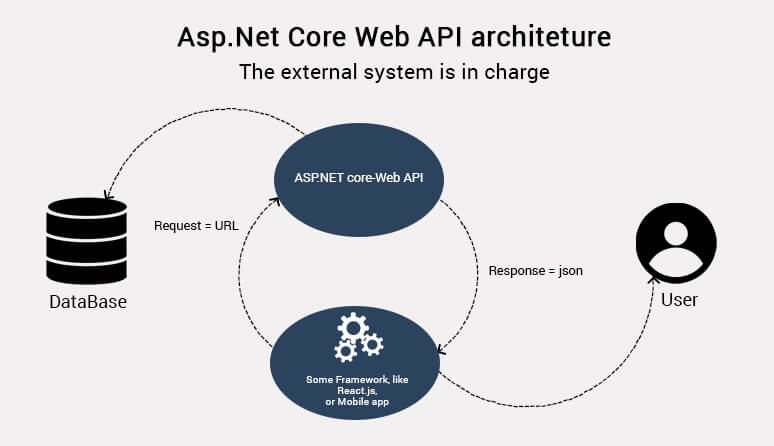
ASP.NET Core Web API là một phần của ASP.NET Core, một framework đa nền tảng để phát triển các ứng dụng web hiện đại. Web API trong ASP.NET Core giúp xây dựng các dịch vụ web theo mô hình RESTful, cho phép giao tiếp giữa client và server thông qua HTTP. Điểm mạnh của nó là hiệu suất cao, dễ dàng mở rộng, và khả năng chạy trên nhiều nền tảng như Windows, Linux, và macOS. Ngoài ra, ASP.NET Core Web API hỗ trợ việc tích hợp với các dịch vụ cloud và microservices, làm cho nó trở thành lựa chọn tối ưu cho các dự án hiện đại[4].

ASP.NET Core Web API hoạt động dựa trên mô hình **client-server**, nơi ứng dụng client sẽ gửi các yêu cầu HTTP (request) tới server và nhận lại các phản hồi (response) từ server. Đây là cách thức cơ bản mà các API hoạt động trong các ứng dụng web ngày nay.

Dưới đây là quy trình cơ bản về cách hoạt động của ASP.NET Core Web API:

* Client gửi yêu cầu: Khi người dùng hoặc một ứng dụng khác gửi một yêu cầu HTTP (GET, POST, PUT, DELETE) từ trình duyệt hoặc một ứng dụng di động, yêu cầu này được gửi tới API.
* ASP.NET Core xử lý yêu cầu: Yêu cầu này sau đó sẽ được ASP.NET Core Web API tiếp nhận và xử lý. API sẽ xác định endpoint dựa trên URL và HTTP method mà client gửi.
* Routing và Mapping: ASP.NET Core sử dụng hệ thống routing để ánh xạ yêu cầu tới đúng *controller* và *action* tương ứng. Controller này sẽ xử lý logic của yêu cầu.
* Thao tác với dữ liệu: Nếu cần, controller có thể thực hiện các thao tác với cơ sở dữ liệu thông qua Entity Framework Core hoặc các công cụ ORM khác.
* Trả về phản hồi: Sau khi xử lý xong, API sẽ trả về dữ liệu phản hồi (thường là dưới dạng JSON hoặc XML) cho client, giúp hiển thị kết quả hoặc xác nhận hành động đã thành công.

ASP.NET Core Web API cung cấp một cách thức hiệu quả để xây dựng các dịch vụ web hiện đại, với khả năng hỗ trợ nhiều ngôn ngữ lập trình và tích hợp dễ dàng với các công nghệ khác.



Hình 4: asp.net core web api

1.2.1.5 Cơ sở dữ liệu SQL Server

**A close-up of a logo

Description automatically generated**

Hình 5: Microsoft SQL Server

SQL Server(hay Microsoft SQL Server)là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft[5].

SQL Server cung cấp cho người dùng các công cụ và tính năng để quản lý, lưu trữ, xử lý các truy vấn dữ liệu, kiểm soát truy cập, xử lý giao dịch và hỗ trợ tích hợp dữ liệu từ nhiều nguồn khác nhau.

Ngoài ra, SQL Server cũng cung cấp các công cụ để tạo báo cáo, phân tích và quản lý cơ sở dữ liệu trực quan thông qua giao diện người dùng hoặc các script lệnh SQL.

SQL Server được xây dựng dựa trên SQL, một ngôn ngữ lập trình tiêu chuẩn để tương tác với cơ sở dữ liệu quan hệ. SQL Server được liên kết với Transact-SQL hoặc T-SQL, triển khai SQL của Microsoft có bổ sung một tập hợp các cấu trúc lập trình độc quyền.

SQL Server hoạt động độc quyền trên môi trường Windows trong hơn 20 năm. Vào năm 2016, Microsoft đã cung cấp SQL Server trên Linux. SQL Server 2017 ra mắt vào tháng 10 năm 2016 chạy được trên cả Windows và Linux.

***SQL Server thường đi kèm với việc thực hiện riêng các ngôn ngữ SQL, T-SQL,... Cụ thể như sau:***

* T-SQL là một trong những loại ngôn ngữ thuộc quyền sở hữu của Microsoft và được gọi với cái tên Transact-SQL. Nó thường cung cấp thêm rất nhiều cho các  khả năng khai báo biến, thủ tục lưu trữ và xử lý ngoại lệ,...
* SQL Server Management Studio là một loại công cụ giao diện chính cho máy chủ cơ sở của chính dữ liệu SQL, thông thường thì nó hỗ trợ cho cả môi trường 64 bit và 32 bit.

**Database Engine của SQL Server**

Thành phần cốt lõi của SQL Server là database engine. Thành phần này bao gồm một công cụ quan hệ có chức năng xử lý các lệnh và truy vấn, một công cụ lưu trữ quản lý các tệp, bảng, trang, index, bộ đệm và giao dịch cơ sở dữ liệu.

Các nhiệm vụ, trigger, trình xem và các đối tượng dữ liệu lưu trữ khác cũng được Database Engine khởi tạo và xử lý.

1.2.1.6. Tích hợp AI trong ứng dụng

* **Ứng dụng gợi ý (Recommendation System):**  
  Ứng dụng đề xuất là một kỹ thuật AI phổ biến dùng để cá nhân hóa trải nghiệm người dùng. Trong đề tài, ứng dụng gợi ý được áp dụng để đề xuất món ăn dựa trên lịch sử đặt hàng, đánh giá, và sở thích của người dùng. Các thuật toán có thể áp dụng bao gồm:
* Gợi ý món ăn dựa trên các đặc điểm (giá, nhà hàng,...).
* Dựa trên hành vi của những người dùng có sở thích tương đồng.
* Phân tích dữ liệu đơn hàng: Một phần quan trọng trong việc đề xuất món ăn chính là phân tích số lượng đơn hàng đã đặt.
* **Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP):**  
  NLP được sử dụng để xử lý các câu hỏi từ người dùng trong chatbot, ví dụ như: “Tôi muốn ăn đồ cay”, “món ăn phù hợp trong 50.000đ”,… Chatbot AI cần hiểu ngôn ngữ tự nhiên để phản hồi phù hợp.
* **Chatbot và mô hình đối thoại:**  
  Gemini 2.0 Flash là một mô hình ngôn ngữ mạnh mẽ do Google phát triển, có khả năng hiểu và phản hồi tự nhiên trong hội thoại. Gemini được tích hợp để đảm nhận vai trò chatbot, hỗ trợ người dùng:
  + Tìm kiếm món ăn bằng ngôn ngữ tự nhiên (VD: “Tôi muốn ăn cơm gà dưới 50.000đ”).
  + Tư vấn món ăn.
  + Giải đáp thắc mắc.
  + Việc sử dụng mô hình Gemini 2.0 Flash giúp chatbot phản hồi nhanh, tự nhiên và chính xác hơn so với kịch bản cố định (rule-based). Chatbot hoạt động thông qua API kết nối với ứng dụng backend để truy xuất dữ liệu và trả về câu trả lời phù hợp.
    1. Kỹ thuật
* Ngôn ngữ và framework:
  + Flutter: dùng để xây dựng giao diện ứng dụng. Flutter giúp tôi viết mã nguồn chung cho cả Android và iOS. Nó sử dụng ngôn ngữ Dart và có các widget để tạo giao diện.
  + Android Studio: Môi trường phát triển (IDE) để viết code, chạy thử và kiểm tra ứng dụng Flutter.
  + ASP.NET Core Web API: Xây dựng các API để kết nối ứng dụng di động với cơ sở dữ liệu, đảm bảo tính bảo mật và khả năng mở rộng.
* Cơ sở dữ liệu: dùng SQL Server để lưu trữ và quản lý dữ liệu, dùng Entity Framework Core để kết nối mã C# với SQL Server.
* Tích hợp AI:
  + Gemini 2.0 Flash: Được tích hợp thông qua API để xử lý hội thoại trong chatbot AI.
  + Xử lý dữ liệu lịch sử đặt món: Sử dụng thuật toán thống kê, lọc cộng tác và kết hợp để đưa ra gợi ý phù hợp.
* Công cụ hỗ trợ:
  + Git: Quản lý mã nguồn.
  + Postman: Kiểm tra API trong quá trình phát triển.

CHƯƠNG 2: PHƯƠNG PHÁP THỰC HIỆN

**Các yêu cầu kỹ thuật**

* **Yêu cầu chức năng:**
  + Cho phép người dùng đăng ký, đăng nhập.
  + Tìm kiếm món ăn theo tên, giá, nhà hàng.
  + Đặt món, xem giỏ hàng, thanh toán đơn hàng.
  + Xem đơn hàng.
  + Chatbot hỗ trợ người dùng tìm món ăn bằng ngôn ngữ tự nhiên.
* **Yêu cầu phi chức năng:**
  + Giao diện sử dụng trên Android/iOS.
  + Phản hồi nhanh, hoạt động ổn định.
  + Bảo mật dữ liệu người dùng.
  + Dễ dàng mở rộng và bảo trì.
* **Yêu cầu tích hợp:**
  + Kết nối với cơ sở dữ liệu SQL Server.
  + Giao tiếp API giữa frontend Flutter và backend ASP.NET Core.
  + Tích hợp mô hình AI Gemini 2.0 Flash qua API.

2.1 Mô hình Usecase



Hình 6: mô hình usecase

2.1.1 Đặc tả usecase “đăng nhập”

|  |  |
| --- | --- |
| **Tên chức năng** | Đăng nhập |
| **Tóm tắt** | Chức năng cho phép người dùng đăng nhập vào ứng dụng. |
| **Dòng sự kiện chính** | 1. Ứng dụng hiển thị form đăng nhập.  2. Người dùng nhập tên đăng nhập và mật khẩu (cả hai trường đều bắt buộc) rồi nhấn “Đăng nhập”.  3. Ứng dụng kiểm tra thông tin đăng nhập.  4. Ứng dụng hiển thị giao diện chính nếu thông tin hợp lệ. |
| **Dòng sự kiện khác** | 1. Nếu thông tin đăng nhập sai:   - Ứng dụng hiển thị thông báo "Tài khoản đăng nhập không hợp lệ". |
| **Các yêu cầu đặc biệt** | Không có |
| **Trạng thái ứng dụng trước khi thực hiện use case** | - Actor: Khách hàng, Quản trị viên.  - Điều kiện: Người dùng chưa đăng nhập vào ứng dụng. |
| **Trạng thái ứng dụng sau khi thực hiện use case** | Người dùng đăng nhập thành công và có thể sử dụng các quyền được cấp bởi ứng dụng. |

Bảng 1: đặc tả usecase đăng nhập

2.1.2 Đặc tả usecase “đăng ký”

|  |  |
| --- | --- |
| **Tên chức năng** | Đăng ký |
| **Tóm tắt** | Chức năng cho phép người dùng tạo tài khoản mới trên ứng dụng. |
| **Dòng sự kiện chính** | 1. Ứng dụng hiển thị form đăng ký.  2. Người dùng nhập các thông tin bắt buộc: tên đăng nhập, mật khẩu, email, họ tên, số điện thoại, địa chỉ.  3. Người dùng nhấn nút "Đăng ký".  4. Ứng dụng kiểm tra thông tin đăng ký.  5. Nếu thông tin hợp lệ, ứng dụng tạo tài khoản mới và hiển thị thông báo đăng ký thành công. |
| **Dòng sự kiện khác** | 1. Nếu tên đăng nhập đã tồn tại: hiển thị thông báo "Tên đăng nhập đã được sử dụng."  2. Nếu email đã tồn tại: hiển thị thông báo "Email đã được sử dụng."  3. Nếu mật khẩu không đáp ứng yêu cầu độ mạnh: hiển thị thông báo "Mật khẩu không đủ mạnh."  4. Nếu các trường bắt buộc không được điền đầy đủ: hiển thị thông báo "Vui lòng điền đầy đủ thông tin." |
| **Các yêu cầu đặc biệt** | - Mật khẩu phải có ít nhất 8 ký tự, bao gồm chữ hoa, chữ thường, số và ký tự đặc biệt.  - Email phải có định dạng hợp lệ. |
| **Trạng thái ứng dụng trước khi thực hiện Use Case** | - **Actor:** Người dùng chưa có tài khoản.  - **Điều kiện:** Ứng dụng đang ở trạng thái cho phép đăng ký. |
| **Trạng thái ứng dụng sau khi thực hiện Use Case** | Người dùng đã có tài khoản và có thể đăng nhập vào ứng dụng. |

Bảng 2: đặc tả usecase đăng ký

2.1.3 Đặc tả usecase “đặt món”

|  |  |
| --- | --- |
| **Tên chức năng** | Đặt món |
| **Tóm tắt** | Cho phép khách hàng chọn và đặt món ăn từ một nhà hàng trên ứng dụng. |
| **Dòng sự kiện chính** | 1. Đăng nhập.  2. Chọn nhà hàng.  3. Xem menu.  4. Chọn món và số lượng.  5. Xác nhận giỏ hàng.  6. Nhận thông tin đơn hàng. |
| **Dòng sự kiện khác** | - Món ăn hết hàng.  - Yêu cầu đăng nhập. |
| **Yêu cầu đặc biệt** | - Đăng nhập bắt buộc.  - Cập nhật số lượng món ăn sau khi đặt. |
| **Trạng thái ứng dụng trước khi thực hiện use case** | Khách hàng đã đăng nhập, ứng dụng sẵn sàng. |
| **Trạng thái ứng dụng sau khi thực hiện use case** | Đơn hàng được tạo, chờ xác nhận. |

Bảng 3: đặc tả usecase đặt món

2.1.4 Đặc tả usecase “đánh giá”

|  |  |
| --- | --- |
| **Tên chức năng** | Đánh giá |
| **Tóm tắt** | Cho phép khách hàng đánh giá và bình luận. |
| **Dòng sự kiện chính** | 1. Đăng nhập.  2. Truy cập trang đánh giá.  3. Nhập điểm và bình luận.  4. Gửi đánh giá.  5. Nhận thông báo xác nhận. |
| **Dòng sự kiện khác** | - Chưa hoàn tất đơn hàng.  - Đánh giá không hợp lệ.  - Yêu cầu đăng nhập. |
| **Yêu cầu đặc biệt** | - Chỉ đánh giá sau khi hoàn tất đơn hàng.  - Điểm từ 1 đến 5.  - Bình luận tuân thủ quy định. |
| **Trạng thái ứng dụng trước khi thực hiện use case** | Khách hàng đã đăng nhập và có đơn hàng hoàn tất. |
| **Trạng thái ứng dụng sau khi thực hiện use case** | Đánh giá được lưu và hiển thị trên trang nhà hàng. |

Bảng 4: đặc tả usecase đánh giá

2.1.5 Đặc tả usecase “thanh toán”

|  |  |
| --- | --- |
| **Tên chức năng** | Thanh toán |
| **Tóm tắt** | Chức năng cho phép khách hàng thanh toán cho đơn hàng đã đặt trên ứng dụng. |
| **Dòng sự kiện chính** | 1. Khách hàng đăng nhập vào ứng dụng.  2. Khách hàng xem giỏ hàng và chọn "Thanh toán".  3. Ứng dụng hiển thị các phương thức thanh toán.  4. Khách hàng chọn phương thức thanh toán và nhập thông tin cần thiết.  5. Khách hàng xác nhận thanh toán.  6. Ứng dụng xử lý thanh toán và hiển thị thông báo kết quả. |
| **Dòng sự kiện khác** | 1. Nếu giỏ hàng trống: Ứng dụng thông báo "Giỏ hàng của bạn trống."  2. Nếu phương thức thanh toán không hợp lệ: Ứng dụng thông báo "Phương thức thanh toán không được hỗ trợ."  3. Nếu thanh toán thất bại: Ứng dụng thông báo "Thanh toán không thành công, vui lòng thử lại." |
| **Các yêu cầu đặc biệt** | - Khách hàng phải đăng nhập để thực hiện thanh toán.  - Ứng dụng phải hỗ trợ ít nhất hai phương thức thanh toán: thanh toán tiền mặt hoặc thanh toán trực tuyến. |
| **Trạng thái ứng dụng trước khi thực hiện Use Case** | - Khách hàng đã đăng nhập.  - Giỏ hàng của khách hàng có ít nhất một món ăn. |
| **Trạng thái ứng dụng sau khi thực hiện Use Case** | - Nếu thanh toán thành công, đơn hàng được xác nhận và chuyển sang trạng thái "Đang chuẩn bị".  - Nếu thanh toán thất bại, đơn hàng vẫn ở trạng thái "Chưa thanh toán". |

Bảng 5: đặc tả usecase thanh toán

2.1.6 Đặc tả usecase “quản lý đơn hàng”

|  |  |
| --- | --- |
| **Tên chức năng** | Quản lý đơn hàng |
| **Tóm tắt** | Chức năng cho phép khách hàng, quản trị viên xem, cập nhật và quản lý trạng thái của các đơn hàng trên ứng dụng. |
| **Dòng sự kiện chính** | 1. Người dùng đăng nhập vào ứng dụng.  2. Người dùng truy cập trang quản lý đơn hàng.  3. Ứng dụng hiển thị danh sách các đơn hàng theo vai trò của người dùng.  4. Người dùng chọn một đơn hàng để xem chi tiết.  5. Người dùng thực hiện các hành động như cập nhật trạng thái, hủy đơn hàng (nếu có quyền).  6. Ứng dụng cập nhật và hiển thị thông tin mới của đơn hàng. |
| **Dòng sự kiện khác** | 1. Nếu người dùng không có quyền xem đơn hàng: Ứng dụng thông báo "Bạn không có quyền truy cập đơn hàng này."  2. Nếu đơn hàng không tồn tại: Ứng dụng thông báo "Đơn hàng không tồn tại."  3. Nếu cập nhật trạng thái không hợp lệ: Ứng dụng thông báo "Trạng thái cập nhật không hợp lệ." |
| **Các yêu cầu đặc biệt** | - Khách hàng chỉ có thể xem và hủy đơn hàng của chính mình.  - Quản trị viên có thể xem và quản lý tất cả đơn hàng trên ứng dụng. |
| **Trạng thái ứng dụng trước khi thực hiện Use Case** | - Người dùng đã đăng nhập.  - Ứng dụng có ít nhất một đơn hàng để quản lý. |
| **Trạng thái ứng dụng sau khi thực hiện Use Case** | - Thông tin đơn hàng được cập nhật theo hành động của người dùng.  - Trạng thái đơn hàng được thay đổi nếu có cập nhật hợp lệ. |

Bảng 6: đặc tả usecase quản lý đơn hàng

2.1.7 Đặc tả usecase “quản lý món ăn”

|  |  |
| --- | --- |
| **Tên chức năng** | Quản lý món ăn |
| **Tóm tắt** | Chức năng cho phép quản trị viên thêm, sửa, xóa và xem danh sách các món ăn trong ứng dụng. |
| **Dòng sự kiện chính** | 1. Quản trị viên đăng nhập vào ứng dụng.  2. Quản trị viên truy cập trang quản lý món ăn.  3. Ứng dụng hiển thị danh sách các món ăn hiện có.  4. Quản trị viên có thể thực hiện các thao tác:  - Thêm món ăn mới.  - Sửa thông tin món ăn.   - Xóa món ăn.  5. Ứng dụng cập nhật và hiển thị danh sách món ăn sau khi thực hiện hành động. |
| **Dòng sự kiện khác** | 1. Nếu người dùng không có quyền quản lý món ăn: Ứng dụng thông báo "Bạn không có quyền thực hiện hành động này."  2. Nếu món ăn không tồn tại: Ứng dụng thông báo "Món ăn không tồn tại."  3. Nếu thông tin món ăn không hợp lệ: Ứng dụng thông báo "Thông tin món ăn không hợp lệ." |
| **Các yêu cầu đặc biệt** | Quản trị viên có thể quản lý món ăn của tất cả các nhà hàng.  - Mỗi món ăn phải có tên, giá và mô tả. |
| **Trạng thái ứng dụng trước khi thực hiện Use Case** | - Quản trị viên đã đăng nhập.  - Ứng dụng có ít nhất một nhà hàng để quản lý món ăn. |
| **Trạng thái ứng dụng sau khi thực hiện Use Case** | Danh sách món ăn được cập nhật theo hành động của quản trị viên (thêm, sửa, xóa). |
| **Điểm mở rộng** | - Ứng dụng có thể cho phép thêm hình ảnh cho món ăn.  - Tích hợp tính năng tìm kiếm hoặc lọc món ăn theo tiêu chí như giá, loại món. |

Bảng 7: đặc tả usecase quản lý món ăn

2.1.8 Đặc tả usecase “quản lý nhà hàng”

|  |  |
| --- | --- |
| **Tên chức năng** | Quản lý nhà hàng |
| **Tóm tắt** | Chức năng cho phép quản trị viên thêm, sửa, xóa và xem danh sách các nhà hàng trong ứng dụng. |
| **Dòng sự kiện chính** | 1. Quản trị viên đăng nhập vào ứng dụng.  2. Quản trị viên truy cập trang quản lý nhà hàng.  3. Ứng dụng hiển thị danh sách các nhà hàng hiện có.  4. Quản trị viên có thể:  - Thêm nhà hàng mới.  - Sửa thông tin nhà hàng.  - Xóa nhà hàng.  5. Ứng dụng cập nhật và hiển thị danh sách nhà hàng sau khi thực hiện hành động. |
| **Dòng sự kiện khác** | 1. Nếu người dùng không có quyền quản lý nhà hàng: Ứng dụng thông báo "Bạn không có quyền thực hiện hành động này."  2. Nếu nhà hàng không tồn tại: Ứng dụng thông báo "Nhà hàng không tồn tại."  3. Nếu thông tin nhà hàng không hợp lệ: Ứng dụng thông báo "Thông tin nhà hàng không hợp lệ." |
| **Các yêu cầu đặc biệt** | - Quản trị viên có thể quản lý tất cả các nhà hàng.  - Mỗi nhà hàng phải có tên, địa chỉ và số điện thoại. |
| **Trạng thái ứng dụng trước khi thực hiện Use Case** | - Quản trị viên đã đăng nhập.  - Ứng dụng có ít nhất một nhà hàng để quản lý. |
| **Trạng thái ứng dụng sau khi thực hiện Use Case** | Danh sách nhà hàng được cập nhật theo hành động của quản trị viên (thêm, sửa, xóa). |

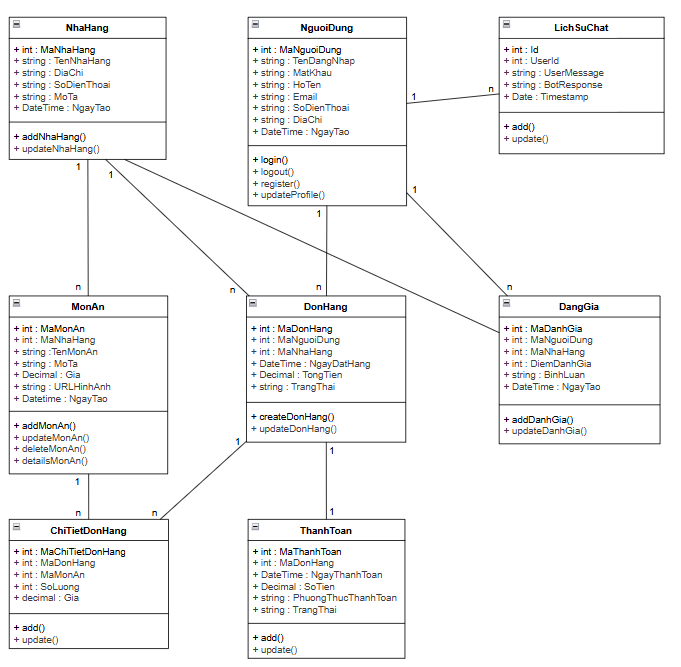
Bảng 8: đặc tả usecase quản lý nhà hàng

2.1.9 Đặc tả usecase “ChatBotAI”

|  |  |
| --- | --- |
| **Tên chức năng** | Chat bot AI |
| **Tóm tắt** | Chức năng hỗ trợ khách hàng tương tác qua giao diện hội thoại, cho phép người dùng đặt câu hỏi về món ăn, nhà hàng hoặc các thông tin liên quan |
| **Dòng sự kiện chính** | 1. Người dùng truy cập ứng dụng và mở giao diện chat.  2. Người dùng nhập câu hỏi hoặc yêu cầu.  3. Chat bot nhận và xử lý dữ liệu.  4. Chat bot trả về kết quả hoặc đề xuất phù hợp với câu hỏi của người dùng.  5. Người dùng nhận được phản hồi ngay lập tức từ ứng dụng. |
| **Dòng sự kiện khác (Trường hợp ngoại lệ)** | - Nếu chat bot không hiểu yêu cầu: Ứng dụng hiển thị thông báo "Xin lỗi, tôi không hiểu câu hỏi của bạn. Vui lòng thử lại."  - Nếu dữ liệu không đủ để đưa ra đề xuất: Ứng dụng thông báo "Chưa có đủ dữ liệu để đề xuất."  - Nếu người dùng chọn "Không quan tâm": Ứng dụng ghi nhận phản hồi để cải thiện thuật toán trong tương lai. |
| **Các yêu cầu đặc biệt** | - Chat bot cần được huấn luyện với dữ liệu từ ứng dụng.  - Ứng dụng phải bảo vệ thông tin cá nhân và tuân thủ các tiêu chuẩn bảo mật.  - Hỗ trợ xử lý ngôn ngữ tự nhiên để hiểu và phản hồi các câu hỏi của người dùng. |
| **Trạng thái ứng dụng trước khi thực hiện** | - Người dùng đã truy cập vào ứng dụng và mở giao diện chat. |
| **Trạng thái ứng dụng sau khi thực hiện** | - Người dùng nhận được phản hồi và đề xuất phù hợp từ chat bot, giúp tìm kiếm món ăn theo yêu cầu. |

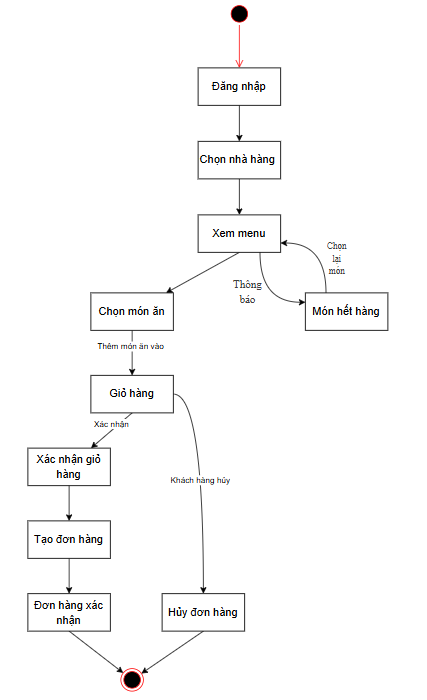
Bảng 9: đặc tả usecase chatbotAI

2.2 Sơ đồ lớp

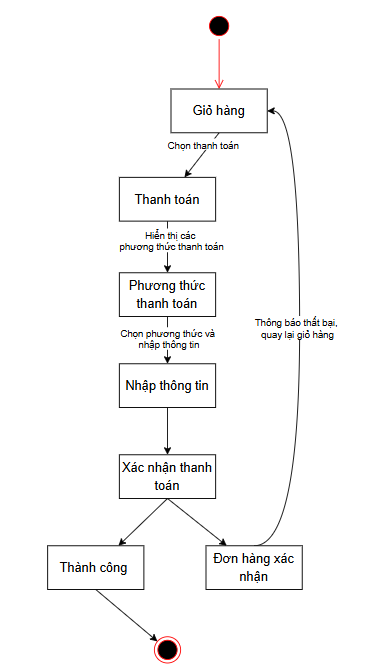


Hình 7: sơ đồ lớp

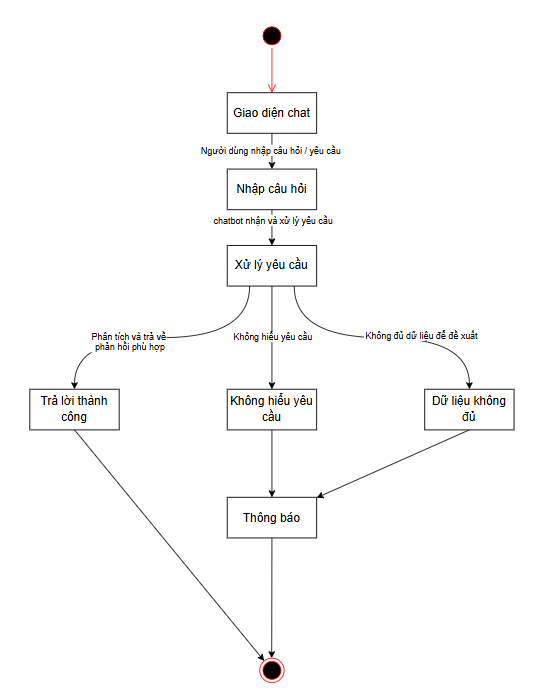
2.3 Sơ đồ trạng thái



Hình 8: sơ đồ trạng thái đặt hàng



Hình 9: sơ đồ trạng thái thanh thoán



Hình 10: sơ đồ trạng thái chat bot AI

2.4 Sơ đồ luồng

Ứng dụng được chia thành ba lớp chính:

* **Ứng dụng khách (Frontend – Flutter):**
* Giao diện người dùng cho đăng nhập, tìm món, đặt hàng, xem lịch sử, chat với AI.
* Tương tác với Web API để gửi/nhận dữ liệu.
* **Máy chủ trung gian (Backend – ASP.NET Core Web API):**
* Xử lý logic nghiệp vụ: xác thực, lưu đơn hàng, phản hồi API.
* Cung cấp API RESTful cho ứng dụng di động.
* Kết nối cơ sở dữ liệu thông qua Entity Framework Core.
* **Cơ sở dữ liệu (SQL Server):**
* Lưu thông tin người dùng, món ăn, nhà hàng, đơn hàng, đánh giá,...
* Cấu trúc dữ liệu được thiết kế để dễ truy xuất và xử lý.
* **AI (Gemini 2.0 Flash):**
* Đảm nhận vai trò chatbot hỗ trợ tìm kiếm và tư vấn món ăn.
* Tương tác qua API với backend để xử lý câu hỏi ngôn ngữ tự nhiên.

A diagram of a structure

AI-generated content may be incorrect.

Hình 11: sơ đồ luồng

2.5 Sơ đồ tuần tự

2.5.1 Sơ đồ tuần tự “đăng nhập”

A diagram of a diagram

AI-generated content may be incorrect.

Hình 12: sơ đồ tuần tự đăng nhập

2.5.2 Sơ đồ tuần tự “đăng ký”

A diagram of a diagram

AI-generated content may be incorrect.

Hình 13: sơ đồ tuần tự đăng ký

2.5.3 Sơ đồ tuần tự “đặt món”

A diagram with text and arrows

AI-generated content may be incorrect.

Hình 14: sơ đồ tuần tự đặt món

2.5.4 Sơ đồ tuần tự “thanh toán”

A diagram of a diagram

AI-generated content may be incorrect.

Hình 15: sơ đồ tuần tự thanh toán

2.5.5 Sơ đồ tuần tự “đơn hàng”

A diagram with text and arrows

AI-generated content may be incorrect.

Hình 16: sơ đồ tuần tự đơn hàng

2.5.6 Sơ đồ tuần tự “món ăn”

A screenshot of a diagram

AI-generated content may be incorrect.

Hình 17: sơ đồ tuần tự món ăn

2.5.7 Sơ đồ tuần tự “nhà hàng”

A screenshot of a diagram

AI-generated content may be incorrect.

Hình 18: sơ đồ tuần tự nhà hàng

2.5.8 Sơ đồ tuần tự “đánh giá”

A diagram of a diagram

AI-generated content may be incorrect.

Hình 19: sơ đồ tuần tự đánh giá

2.5.9 Sơ đồ tuần tự “ChatBotAI”

A diagram of a diagram

AI-generated content may be incorrect.

Hình 20: sơ đồ tuần tự chatbotAI

2.5 Thiết kế cơ sở dữ liệu

A screenshot of a computer

AI-generated content may be incorrect.

Hình 21: cơ sở dữ liệu

2.5.1 Mô tả chi tiết “NguoiDung”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng buộc** | **Mô Tả** |
| **MaNguoiDung** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất cho mỗi người dùng. Tự động tăng. |
| **TenDangNhap** | NVARCHAR(50) | NOT NULL, UNIQUE | Tên đăng nhập duy nhất của người dùng. |
| **MatKhau** | NVARCHAR(255) | NOT NULL | Mật khẩu đã được mã hóa. |
| **HoTen** | NVARCHAR(100) |  | Họ và tên đầy đủ của người dùng. |
| **Email** | NVARCHAR(100) | UNIQUE | Địa chỉ email (không trùng lặp). |
| **SoDienThoai** | NVARCHAR(20) |  | Số điện thoại của người dùng. |
| **DiaChi** | NVARCHAR(255) |  | Địa chỉ cư trú hoặc giao hàng. |
| **NgayTao** | DATETIME | DEFAULT GETDATE() | Ngày tạo tài khoản (mặc định là thời gian hiện tại). |

Bảng 10: mô tả chi tiết người dùng

2.5.2 Mô tả chi tiết “NhaHang”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **MaNhaHang** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất cho mỗi nhà hàng. Tự động tăng. |
| **TenNhaHang** | NVARCHAR(100) | NOT NULL | Tên của nhà hàng. |
| **DiaChi** | NVARCHAR(255) |  | Địa chỉ cụ thể của nhà hàng. |
| **SoDienThoai** | NVARCHAR(20) |  | Số điện thoại liên hệ của nhà hàng. |
| **MoTa** | NVARCHAR(500) |  | Mô tả ngắn gọn về nhà hàng (phong cách, đặc điểm, loại món ăn,...). |
| **NgayTao** | DATETIME | DEFAULT GETDATE() | Ngày nhà hàng được thêm vào ứng dụng (tự động lấy ngày hiện tại). |

Bảng 11: mô tả chi tiết nhà hàng

2.5.3 Mô tả chi tiết “MonAn”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **MaMonAn** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất của món ăn. Tự động tăng. |
| **MaNhaHang** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại liên kết đến bảng NhaHang, xác định nhà hàng cung cấp món ăn. |
| **TenMonAn** | NVARCHAR(100) | NOT NULL | Tên món ăn (ví dụ: "Phở bò", "Cơm gà"). |
| **MoTa** | NVARCHAR(500) |  | Mô tả chi tiết món ăn, nguyên liệu, cách chế biến,... |
| **Gia** | DECIMAL(10,2) | NOT NULL | Giá bán của món ăn. |
| **URLHinhAnh** | NVARCHAR(255) |  | Đường dẫn đến hình ảnh minh họa của món ăn. |
| **NgayTao** | DATETIME | DEFAULT GETDATE() | Ngày món ăn được thêm vào ứng dụng (tự động lấy thời gian hiện tại). |

Bảng 12: mô tả chi tiết món ăn

2.5.4 Mô tả chi tiết “DonHang”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **MaDonHang** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất cho đơn hàng. Tự động tăng. |
| **MaNguoiDung** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại đến NguoiDung, xác định người đặt đơn. |
| **MaNhaHang** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại đến NhaHang, xác định nhà hàng cung cấp đơn hàng. |
| **NgayDatHang** | DATETIME | DEFAULT GETDATE() | Thời gian đặt hàng (mặc định là thời gian hiện tại). |
| **TongTien** | DECIMAL(10,2) | NOT NULL | Tổng số tiền của đơn hàng. |
| **TrangThai** | NVARCHAR(50) | NOT NULL | Trạng thái đơn hàng (ví dụ: "Đang xử lý", "Đã giao", "Đã hủy"). |
| **DiaChiGiaoHang** | NVARCHAR(200) | NULL | Địa chỉ giao hàng cụ thể của người dùng. |
| **SoDienThoai** | VARCHAR(20) | NULL | Số điện thoại liên hệ khi giao hàng. |
| **GhiChu** | NVARCHAR(500) | NULL | Ghi chú bổ sung từ người dùng (ví dụ: "Giao sau 18h", "Không lấy hành"). |

Bảng 13: mô tả chi tiết đơn hàng

2.5.5 Mô tả chi tiết “ChiTietDonHang”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **MaChiTietDonHang** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất cho mỗi dòng chi tiết đơn hàng. |
| **MaDonHang** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại liên kết đến bảng DonHang, xác định đơn hàng. |
| **MaMonAn** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại liên kết đến bảng MonAn, xác định món ăn được đặt. |
| **SoLuong** | INT | NOT NULL | Số lượng món ăn được đặt trong đơn. |
| **Gia** | DECIMAL(10,2) | NOT NULL | Đơn giá của món ăn tại thời điểm đặt hàng (không phụ thuộc vào giá hiện tại trong bảng MonAn). |

Bảng 14: mô tả chi tiết chi tiết đơn hàng

2.5.6 Mô tả chi tiết “GioHang”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **MaGioHang** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất cho mỗi dòng trong giỏ hàng. |
| **MaNguoiDung** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại đến NguoiDung, xác định người dùng sở hữu giỏ hàng. |
| **MaMonAn** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại đến MonAn, xác định món ăn được thêm vào giỏ. |
| **SoLuong** | INT | NOT NULL, CHECK (SoLuong > 0) | Số lượng món ăn được chọn. Phải lớn hơn 0. |
| **NgayThem** | DATETIME | DEFAULT GETDATE() | Ngày giờ món ăn được thêm vào giỏ (mặc định là thời điểm hiện tại). |

Bảng 15: mô tả chi tiết giỏ hàng

2.5.7 Mô tả chi tiết “ThanhToan”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **MaThanhToan** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất cho mỗi giao dịch thanh toán. |
| **MaDonHang** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại liên kết đến bảng DonHang, xác định đơn hàng được thanh toán. |
| **NgayThanhToan** | DATETIME | DEFAULT GETDATE() | Ngày giờ thực hiện thanh toán (mặc định là thời gian hiện tại). |
| **SoTien** | DECIMAL(10,2) | NOT NULL | Số tiền đã thanh toán cho đơn hàng. |
| **PhuongThucThanhToan** | NVARCHAR(50) | NOT NULL | Phương thức thanh toán (ví dụ: "Tiền mặt", "Thẻ tín dụng", "Ví điện tử"). |
| **TrangThai** | NVARCHAR(50) | NOT NULL | Trạng thái thanh toán (ví dụ: "Thành công", "Thất bại", "Đang xử lý"). |

Bảng 16: mô tả chi tiết thanh toán

2.5.8 Mô tả chi tiết “DanhGia”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **MaDanhGia** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất của mỗi đánh giá. |
| **MaNguoiDung** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại đến bảng NguoiDung, xác định ai đã đánh giá. |
| **MaNhaHang** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại đến bảng NhaHang, xác định nhà hàng được đánh giá. |
| **DiemDanhGia** | INT | NOT NULL, CHECK (DiemDanhGia >= 1 AND <= 5) | Số điểm đánh giá từ 1 đến 5 sao. |
| **BinhLuan** | NVARCHAR(500) |  | Nhận xét chi tiết của người dùng về nhà hàng. |
| **NgayTao** | DATETIME | DEFAULT GETDATE() | Ngày đánh giá được tạo (mặc định là thời điểm hiện tại). |

Bảng 17: mô tả chi tiết đánh giá

2.5.9 Mô tả chi tiết “ChatHistory”

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Cột** | **Kiểu Dữ Liệu** | **Ràng Buộc** | **Mô Tả** |
| **Id** | INT | PRIMARY KEY, IDENTITY(1,1) | Mã định danh duy nhất cho mỗi dòng trong lịch sử trò chuyện. |
| **UserId** | INT | NOT NULL, FOREIGN KEY | Khóa ngoại đến bảng NguoiDung, xác định người dùng tham gia trò chuyện. |
| **UserMessage** | NVARCHAR(500) | NOT NULL | Tin nhắn người dùng gửi cho chatbot. |
| **BotResponse** | NVARCHAR(2000) | NOT NULL | Phản hồi của chatbot sau khi nhận được tin nhắn từ người dùng. |
| **Timestamp** | DATETIME | DEFAULT GETDATE() | Thời gian gửi tin nhắn từ người dùng và nhận phản hồi từ chatbot (mặc định là thời điểm hiện tại). |

Bảng 18: mô tả chi tiết chathistory

CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM

3.1 Mô tả bài toán

Dự án ứng dụng đặt đồ ăn này được phát triển với kiến trúc đa tầng, bao gồm ứng dụng di động Flutter, WebAPI ASP.NET Core và cơ sở dữ liệu SQL Server. Ứng dụng cho phép người dùng đăng ký tài khoản, duyệt danh sách nhà hàng và món ăn, đặt hàng, thanh toán và đánh giá nhà hàng. Cơ sở dữ liệu được thiết kế với các bảng chính như NguoiDung, NhaHang, MonAn, DonHang, ChiTietDonHang, GioHang, ThanhToan, DanhGia và ChatHistory. Mỗi bảng được liên kết với nhau thông qua các khóa ngoại, tạo nên một ứng dụng quan hệ hoàn chỉnh. Người dùng có thể đăng ký, đăng nhập, cập nhật thông tin cá nhân và quản lý địa chỉ giao hàng. Họ có thể xem danh sách nhà hàng, tìm kiếm theo tên hoặc địa chỉ, xem chi tiết nhà hàng và danh sách món ăn. Khi đã chọn được món ăn, người dùng có thể thêm vào giỏ hàng, điều chỉnh số lượng và tạo đơn hàng. Ứng dụng hỗ trợ nhiều phương thức thanh toán và cho phép theo dõi trạng thái đơn hàng. Sau khi nhận được đồ ăn, người dùng có thể đánh giá nhà hàng với thang điểm từ 1-5 sao và viết bình luận. Ngoài ra, ứng dụng còn tích hợp chatbot để hỗ trợ người dùng với các câu hỏi thường gặp, với lịch sử chat được lưu lại để cải thiện trải nghiệm. WebAPI cung cấp các endpoint cho tất cả các chức năng này, trong khi ứng dụng Flutter mang đến giao diện người dùng thân thiện với các màn hình đăng nhập, trang chủ, chi tiết nhà hàng, giỏ hàng, thanh toán, lịch sử đơn hàng, đánh giá và chat. Dự án sử dụng Entity Framework Core để tương tác với cơ sở dữ liệu, đảm bảo hiệu suất và bảo mật. Tổng thể, đây là một giải pháp toàn diện cho việc đặt đồ ăn trực tuyến, với giao diện người dùng thân thiện và ứng dụng backend mạnh mẽ, đảm bảo trải nghiệm người dùng tốt nhất.

3.2 Quá trình thực hiện

Ứng dụng được phát triển theo mô hình client-server, chia làm 2 phần chính: ứng dụng di động Flutter (client) và API backend ASP.NET Core (server). Dữ liệu được lưu trữ và xử lý qua SQL Server với thiết kế quan hệ chặt chẽ giữa các bảng như: người dùng, món ăn, đơn hàng, giỏ hàng, đánh giá,…

Một trong những điểm đặc biệt của ứng dụng là tích hợp chatbot AI sử dụng mô hình Gemini 2.0 Flash để hỗ trợ người dùng tìm kiếm món ăn và tư vấn khi đặt hàng.

Các bước triển khai chính bao gồm:

* Thiết kế cơ sở dữ liệu.
* Xây dựng API phục vụ ứng dụng di động.
* Phát triển giao diện người dùng bằng Flutter.
* Tích hợp mô hình Gemini 2.0 Flash để xây dựng chatbot AI.
* Kiểm thử các chức năng chính của ứng dụng trên thiết bị giả lập và thực tế.

Khó khăn gặp phải:

* Tích hợp mô hình AI đòi hỏi hiểu rõ cách truyền ngữ cảnh, giữ phiên hội thoại, đồng thời xử lý được tiếng Việt tốt.
* Đồng bộ hóa dữ liệu giỏ hàng và đơn hàng cần được kiểm tra kỹ để tránh sai sót khi nhiều người dùng tương tác cùng lúc.

3.3 Kết quả đạt được

3.3.1 Giao diện đăng nhập

A screenshot of a phone

AI-generated content may be incorrect.

Hình 22: giao diện đăng nhập

3.3.2 Giao diện đăng ký

A screenshot of a phone

AI-generated content may be incorrect.

Hình 23: giao diện đăng ký

3.3.3 Giao diện trang chủ

A screen shot of a phone

AI-generated content may be incorrect.

Hình 24: giao diện trang chủ

3.3.4 Giao diện món ăn

A screenshot of a phone

AI-generated content may be incorrect.

Hình 25: giao diện món ăn

3.3.5 Giao diện chi tiết món ăn

A screenshot of a phone

AI-generated content may be incorrect.

Hình 26: giao diện chi tiết món ăn

3.3.6 Giao diện giỏ hàng

A screenshot of a cell phone

AI-generated content may be incorrect.

Hình 27: giao diện giỏ hàng

3.3.7 Giao diện thanh toán

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Hình 28: giao diện thanh toán

3.3.9 Giao diện đơn hàng

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a cell phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Hình 29: giao diện đơn hàng

3.3.10 Giao diện thông tin cá nhân

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Hình 30: giao diện thông tin cá nhân

3.3.11 Giao diện hộp thoại chat bot

A cell phone with text messages

AI-generated content may be incorrect. A cell phone with text on it

AI-generated content may be incorrect.

Hình 31: giao diện hộp thoại chat bot AI

CHƯƠNG 4: KẾT LUẬN

4.1 Đánh giá kết quả đạt được

Xây dựng ứng dụng đặt đồ ăn trên điện thoại và tích hợp AI đã cơ bản đạt được các mục tiêu đặt ra trong đề cương ban đầu. Hệ thống cho phép người dùng dễ dàng đăng nhập, tìm kiếm món ăn, thêm vào giỏ hàng, đặt món, thanh toán, đánh giá nhà hàng và đặc biệt là giao tiếp với chatbot AI để nhận được gợi ý món ăn phù hợp. Việc tích hợp chatbot đã giúp cải thiện trải nghiệm người dùng, mang tính cá nhân hóa, phù hợp với xu hướng ứng dụng AI vào dịch vụ khách hàng.

Hệ thống đã giải quyết được phần lớn yêu cầu của bài toán đặt ra, bao gồm:

* Tạo môi trường giao tiếp thân thiện giữa người dùng và hệ thống.
* ChatBot hỗ trợ người dùng trong việc tìm món ăn dựa trên giá, tên hoặc mô tả.

4.2 Kiến thức lý thuyết đã được củng cố

Thông qua quá trình thực hiện đề tài, các kiến thức lý thuyết sau đã được củng cố và vận dụng hiệu quả:

* Mô hình client-server và lập trình đa lớp.
* Kiến thức về cơ sở dữ liệu quan hệ, thiết kế hệ thống quản lý đơn hàng, giỏ hàng, người dùng.
* Nguyên lý tích hợp AI vào hệ thống thực tế.

Hiểu rõ hơn về xử lý ngôn ngữ tự nhiên cơ bản thông qua việc sử dụng mô hình Gemini để trò chuyện với người dùng.

4.3 Kỹ năng thực hành đã học được

Trong quá trình làm đồ án, em đã học hỏi và rèn luyện được nhiều kỹ năng thực hành quan trọng:

* Kỹ năng lập trình ứng dụng di động với Flutter và Dart.
* Thiết kế và triển khai API backend với ASP.NET Core.
* Thiết kế cơ sở dữ liệu với SQL Server một cách chặt chẽ, khoa học.
* Tích hợp mô hình AI vào sản phẩm thực tế.
* Sử dụng các công cụ hỗ trợ phát triển như Git, Postman, Android Studio, Visual Studio.

4.4 Kinh nghiệm thực tiễn tích lũy được

Thông qua đề tài, em đã tích lũy được nhiều kinh nghiệm thực tiễn như:

* Quản lý thời gian, phân chia công việc theo giai đoạn và ưu tiên.
* Tư duy giải quyết vấn đề khi gặp lỗi trong lập trình, giao diện, hoặc kết nối AI.
* Làm việc theo hướng sản phẩm thực tế chứ không chỉ là mô hình lý thuyết.
* Giao tiếp API giữa client - server hiệu quả, xử lý bất đồng bộ và phản hồi nhanh.

4.5 Thảo luận và các vấn đề chưa giải quyết

Mặc dù hệ thống đã hoạt động ổn định với các chức năng cơ bản, nhưng vẫn còn một số vấn đề chưa được giải quyết triệt để:

* Chatbot AI hiện tại hoạt động chủ yếu dựa vào phản hồi theo ngữ cảnh đơn giản, chưa xử lý được các cuộc hội thoại phức tạp, kéo dài.
* Chưa tích hợp hệ thống thông báo đẩy (push notification) cho cập nhật đơn hàng.
* Chưa có hệ thống quản lý ưu đãi khuyến mãi động, liên kết trực tiếp với AI để tư vấn theo chương trình ưu đãi mới.

4.6 Kết luận

Đề tài đã hoàn thành đúng theo định hướng ban đầu, xây dựng thành công một ứng dụng đặt đồ ăn có tích hợp AI, mang lại trải nghiệm thân thiện, thuận tiện và cá nhân hóa cho người dùng. Qua quá trình làm đồ án, em không chỉ củng cố kiến thức chuyên môn mà còn nâng cao kỹ năng thực tế, sẵn sàng cho các công việc phát triển phần mềm trong tương lai.

4.7 Đề nghị rút ra từ kết quả đề tài

Việc tích hợp AI vào ứng dụng đặt đồ ăn là hoàn toàn khả thi và đem lại nhiều giá trị thực tiễn, đặc biệt trong việc gợi ý món ăn thông minh và giao tiếp với người dùng tự nhiên hơn.

Cần tiếp tục nghiên cứu và cập nhật các mô hình AI mới, để nâng cao khả năng hiểu ngữ cảnh và đa dạng hóa phản hồi.

4.8 Hướng phát triển tương lai

Trong tương lai, đề tài có thể được phát triển thêm theo các hướng sau:

* Tích hợp AI phân tích giọng nói để người dùng có thể đặt hàng bằng giọng nói.
* Cải tiến chatbot với khả năng hiểu ngữ cảnh dài hạn, cá nhân hóa hơn nữa dựa trên thói quen và thời gian sử dụng.
* Thêm chức năng quản lý khuyến mãi linh hoạt, tự động cập nhật và nhắc nhở người dùng.
* Mở rộng hệ thống quản lý cho nhiều nhà.
* Tích hợp các phương thức thanh toán đa dạng hơn như ví điện tử MoMo, ZaloPay.

TÀI LIỆU THAM KHẢO

[1] R. N. Jeminiwa, N. S. Hohmann, and B. I. Fox, “Developing a theoretical framework for evaluating the quality of mHealth apps for adolescent users: A systematic review,” *J. Pediatr. Pharmacol. Ther.*, vol. 24, no. 4, pp. 254–269, 2019.

[2] N. G. Ugur and A. H. Turan, “Mobile applications acceptance: A theoretical model proposal and empirical test,” *Int. J. E-Adoption*, vol. 11, no. 2, pp. 13–30, 2019.

[3] E. Masi, G. Cantone, M. Mastrofini, G. Calavaro, and P. Subiaco, “Mobile Apps Development: A Framework for Technology Decision Making,” in *Lecture Notes in Comput. Sci.*, vol. 110, D. Uhler, K. Mehta, and J. L. Wong, Eds., Springer, 2013, pp. 64–79.

[4] B. G. Jayatilleke, G. R. Ranawaka, C. Wijesekera, and M. C. B. Kumarasinha, “Development of mobile application through design-based research,” *Asian Assoc. Open Univ. J.*, vol. 13, no. 2, pp. 145–168, 2019.

[5] Y. Xu, X. Liu, Z. E. Mao, and J. Zhou, “Mobile food ordering apps, restaurant performance, and customer satisfaction,” *Cornell Hosp. Q.*, vol. 55, no. 4, pp. 365–375, 2023.

[6] P. Kaur, A. Dhir, S. Talwar, and K. Ghuman, “The value proposition of food delivery apps from the perspective of theory of consumption value,” *Int. J. Contemp. Hosp. Manag.*, vol. 33, no. 4, pp. 1129–1159, 2021.

[7] M. Keeble, J. Adams, G. Sacks, L. Vanderlee, C. M. White, D. Hammond, and T. Burgoine, “Use of online food delivery services to order food prepared away-from-home and associated sociodemographic characteristics: A cross-sectional, multi-country analysis,” *Int. J. Environ. Res. Public Health*, vol. 17, no. 14, 5190, 2020.

[8] “Mobile Application Development with Artificial Intelligence,” *Int. J. Comput. Sci.Mobile Comput.*, vol. 9, no. 3, 2020.

[9] M. L. Napoli, *Beginning Flutter: A Hands-On Guide to App Development*, Wrox Press, 2019.

[10] M. Spasojevic and V. Pecanac, *Ultimate ASP.NET Core Web API*, 2nd ed., CodeMaze, 2021.