

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

----------



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài: Xây dựng ứng dụng quản lý dinh dưỡng tích hợp chatbot AI

Nhóm sinh viên thực hiện:

Họ và tên Mã sinh viên Lớp

Nguyễn Khánh An B21DCCN002 D21HTTT1

Vũ Trung Lập B21DCCN479 D21HTTT6

Giảng viên hướng dẫn: PGS. TS. Nguyễn Trọng Khánh

Hệ đào tạo: Đại học chính quy

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

NGUYỄN KHÁNH AN

VŨ TRUNG LẬP

NĂM 2025

HÀ NỘI - 2025

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

-----❧❧❧❧❧-----



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài: Xây dựng ứng dụng quản lý dinh dưỡng tích hợp chatbot AI

Nhóm sinh viên thực hiện:

Họ và tên Mã sinh viên Lớp

Nguyễn Khánh An B21DCCN002 D21HTTT1

Vũ Trung Lập B21DCCN479 D21HTTT6

Giảng viên hướng dẫn: PGS. TS. Nguyễn Trọng Khánh

Hệ đào tạo: Đại học chính quy

HÀ NỘI - 2025

LỜI CẢM ƠN

Trước tiên, Nhóm thực hiện xin gửi lời cảm ơn sâu sắc nhất đến thầy giáo Nguyễn Trọng Khánh, người đã tận tình hướng dẫn trong suốt quá trình nghiên cứu và thực hiện, giúp nhóm hoàn thành tốt đồ án tốt nghiệp này.

Nhóm xin gửi lời tri ân đến quý thầy cô giáo trong khoa Công nghệ thông tin đã tận tình giảng dạy trong suốt bốn năm qua. Những kiến thức nhận được từ quý thầy cô sẽ là hành trang quý báu, tạo nền tảng vững chắc cho con đường sự nghiệp trong tương lai.

Nhóm thực hiện cũng xin bày tỏ lòng biết ơn chân thành đến gia đình và người thân đã luôn ở bên động viên, hỗ trợ các thành viên vượt qua những khó khăn trong học tập cũng như cuộc sống.

Cuối cùng, xin gửi lời cảm ơn đến tất cả bạn bè đã đồng hành trong bốn năm học tập tại Học viện, luôn giúp đỡ và khích lệ để nhóm có thể hoàn thành đồ án này một cách tốt nhất.

Xin chân thành cảm ơn!

Hà Nội, tháng 12 năm 2025

Đại diện nhóm

Nguyễn Khánh An

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của giảng viên hướng dẫn)

Điểm:.....(Bằng chữ:.....)

Đồng ý/không đồng ý cho sinh viên bảo vệ trước hội đồng chấm tốt nghiệp.

Hà Nội, tháng 12 năm 2024

CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN

NGUYỄN TRỌNG KHÁNH

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của giảng viên phản biện)

Điểm:.....(Bằng chữ:.....)

Đồng ý/không đồng ý cho sinh viên bảo vệ trước hội đồng chấm tốt nghiệp.

Hà Nội, tháng 12 năm 2024

CÁN BỘ - GIẢNG VIÊN PHẢN BIỆN

NGUYỄN TRỌNG KHÁNH

BẢNG PHÂN CHIA CÔNG VIỆC

Thành viên	Nhiệm vụ
Nguyễn Khánh An	<ul style="list-style-type: none"> - Kiểm tra, sàng lọc dữ liệu thu thập được; loại bỏ dữ liệu trùng lặp, nhiễu và xử lý lỗi định dạng. - Thiết kế kiến trúc Cơ sở dữ liệu. - Xây dựng và tối ưu hóa các API cho hệ thống. - Xây dựng giao diện người dùng (UI/UX) và các chức năng trên ứng dụng di động. - Kiểm thử chức năng toàn hệ thống. - Rà soát lỗi logic, lỗi giao diện và thực hiện các bản vá lỗi. - Soạn thảo báo cáo.
Vũ Trung Lập	<ul style="list-style-type: none"> - Thu thập, tổng hợp dữ liệu thô về nguyên liệu và món ăn từ các nguồn uy tín. - Chuẩn hóa định dạng và nhập liệu vào hệ thống cơ sở dữ liệu. - Nghiên cứu và xây dựng mô hình Chatbot tư vấn dinh dưỡng. - Phát triển thuật toán gợi ý thực đơn. - Kiểm thử chuyên sâu các tính năng AI, đánh giá độ chính xác của mô hình. - Phát hiện lỗi và tối ưu hóa hiệu năng xử lý của các thuật toán. - Soạn thảo báo cáo.

LINK GITHUB SẢN PHẨM

Link github ứng dụng mobile MyHealth:

https://github.com/nguyenkhanhan123/Mobile_MyHealth

Link github backend hệ thống:

https://github.com/nguyenkhanhan123/Backend_MyHealth

Link github service AI:

https://github.com/trunglap923/Server_AI_DATN

MỤC LỤC

DANH MỤC BẢNG BIÊU	
DANH MỤC HÌNH ẢNH	
KÝ HIỆU VÀ THUẬT NGỮ	
LỜI MỞ ĐẦU	1
Chương 1. CƠ SỞ LÝ THUYẾT	2
1.1. Kiến thức nền tảng về dinh dưỡng	2
1.1.1. Chỉ số BMI	2
1.1.2. Chỉ số BMR	3
1.1.3. Chỉ số TDEE	4
1.1.4. Sự phát triển của quy tắc Wishnofsky	5
1.1.5. Lượng nước cần thiết hàng ngày	6
1.1.6. Nhu cầu dinh dưỡng khuyến nghị	7
1.1.7. Phân mức trên nhãn thông tin dinh dưỡng	7
1.1.8. USDA FoodData Central	8
1.2. Kiến thức nền tảng về AI	9
1.2.1. LLM	9
1.2.2. RAG	11
1.2.3. Self-Query Retriever	13
1.3. Thuật toán SLSQP	15
1.3.1. Tổng quan	15
1.3.2. Dạng bài áp dụng	15
1.3.3. Mô hình hóa bài toán gợi ý khẩu phần ăn	16
1.4. Các công nghệ sử dụng	17
1.4.1. Flutter	17
1.4.2. Dart và Dio	18
1.4.3. Python	20
1.4.4. FastAPI	21
1.4.5. PostgreSQL	22
1.4.6. Firebase và FCM	23

1.4.7. LangChain	25
1.4.8. LangGraph.....	25
1.4.9. DeepSeek LLM	26
1.4.10. ElasticSearch	26
1.5. Tổng kết chương	27
Chương 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG	28
2.1. Tổng quan về nghiệp vụ.....	28
2.2. Phân tích yêu cầu chức năng.....	29
2.2.1. Tổng quan hệ thống.....	29
2.2.2. Đặc tả Usecase	30
2.3. Phân tích yêu cầu phi chức năng	39
2.3.1. Yêu cầu bảo mật.....	39
2.3.2. Yêu cầu hiệu năng	39
2.4. Phân tích biểu đồ phân tích.....	40
2.5. Thiết kế cơ sở dữ liệu hệ thống.....	43
2.6. Thiết kế biểu đồ tuần tự cho từng chức năng.....	44
2.7. Thiết kế luồng xử lý chức năng AI	55
2.7.1. Luồng hoạt động chức năng gợi ý thực đơn	55
2.7.2. Luồng hoạt động chức năng thay đổi món ăn.....	55
2.7.3. Luồng hoạt động chức năng chatbot	56
2.8. Tổng kết chương	58
Chương 3. CÀI ĐẶT VÀ KIỂM THỦ HỆ THỐNG	59
3.1. Kiến trúc hệ thống.....	59
3.2. Môi trường triển khai	59
3.3. Giao diện và chức năng đã hoàn thành	60
3.3.1. Giao diện module đăng nhập	60
3.3.2. Giao diện module đăng ký	61
3.3.3. Giao diện module quản lý hồ sơ thể chất.....	61
3.3.4. Giao diện module quản lý lịch trình tập thể dục	64
3.3.5. Giao diện module quản lý lượng nước uống.....	65
3.3.6. Giao diện module quản lý các món ăn từng bữa trong ngày	66
3.3.7. Giao diện module thống kê dinh dưỡng trong ngày	67

3.3.8. Giao diện module hỏi đáp với chuyên gia AI	68
3.3.9. Giao diện module gửi góp ý cho quản trị viên.....	69
3.3.10. Giao diện module quản lý các món ăn hiển thị cho người dùng	69
3.3.11. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn.....	72
3.3.12. Giao diện module nhận thông báo góp ý của người dùng	74
3.4. Đánh giá hệ thống	75
3.4.1. Đánh giá mô hình truy xuất.....	75
3.4.2. Đánh giá hiệu năng hệ thống.....	77
3.5. Tổng kết chương	78
Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	79
4.1. Kết luận	79
4.2. Hướng phát triển	79
4.3. Tổng kết chương	79
TÀI LIỆU THAM KHẢO	81

DANH MỤC BẢNG BIỂU

Bảng 1.1.Lượng nước cần bù thêm so với mức độ hoạt động	7
Bảng 2.1.Kịch bản usecase Đăng nhập	30
Bảng 2.2. Kịch bản usecase Đăng ký tài khoản mới	30
Bảng 2.3. Kịch bản usecase Quản lý lịch trình tập thể dục	31
Bảng 2.4. Kịch bản usecase Quản lý lượng nước uống.....	32
Bảng 2.5. Kịch bản usecase Quản lý các món ăn từng bữa trong ngày	33
Bảng 2.6.Kịch bản usecase Quản lý hồ sơ thể chất.....	34
Bảng 2.7. Kịch bản usecase Thông kê dinh dưỡng trong ngày	34
Bảng 2.8. Kịch bản usecase Hỏi đáp với chuyên gia AI	35
Bảng 2.9. Kịch bản usecase Gửi góp ý cho quản trị viên.....	36
Bảng 2.10. Kịch bản usecase Nhận thông báo góp ý của người dùng	36
Bảng 2.11. Kịch bản usecase Quản lý các món ăn hiển thị cho người dùng	37
Bảng 2.12. Kịch bản usecase Quản lý các nguyên liệu sẽ dùng cho món ăn.....	38
Bảng 2.13. Bảng chi tiết thông tin các lớp	42
Bảng 3.1. Kết quả đánh giá của mô hình theo từng query	76
Bảng 3.2. Kết quả đánh giá trung bình của mô hình	76
Bảng 3.3. Đánh giá hiệu năng chức năng gợi ý, thay đổi món	77
Bảng 3.4. Đánh giá hiệu năng chức năng chatbot	77

DANH MỤC HÌNH ẢNH

Ảnh 1.1. Các mức độ của chỉ số BMI	2
Ảnh 1.2. Kiến trúc RAG	11
Ảnh 1.3. Cách hoạt động của RAG	12
Ảnh 1.4. Cách hoạt động Self-Query Retriever	14
Ảnh 1.5. Flutter – Framework phát triển ứng dụng đa nền tảng của Google.....	17
Ảnh 1.6. Dart – Ngôn ngữ lập trình chính của Flutter	18
Ảnh 1.7. Luồng giao tiếp Web/Mobile App với Server thông qua Dio	19
Ảnh 1.8. Ngôn ngữ lập trình Python	20
Ảnh 1.9. FastAPI framework cho backend Python	21
Ảnh 1.10. Hệ quản trị cơ sở dữ liệu PostgreSQL.....	22
Ảnh 1.11. Firebase – Hỗ trợ phát triển ứng dụng nhanh chóng	23
Ảnh 1.12. Cơ chế hoạt động của FCM	24
Ảnh 1.13. Framework LangChain	25
Ảnh 1.14. Framework LangGraph.....	25
Ảnh 1.16. Deepseek.....	26
Ảnh 1.17. ElasticSearch	27
Ảnh 2.1. Biểu đồ usecase tổng quan hệ thống.....	29
Ảnh 2.2. Biểu đồ lớp của hệ thống	40
Ảnh 2.3. Cơ sở dữ liệu hệ thống.....	43
Ảnh 2.4. Biểu đồ tuần tự module Đăng nhập	44
Ảnh 2.5. Biểu đồ tuần tự module Đăng ký	45
Ảnh 2.6. Biểu đồ tuần tự module Quản lý lịch trình tập thể dục	46
Ảnh 2.7. Biểu đồ tuần tự module Quản lý lượng nước uống	47
Ảnh 2.8. Biểu đồ tuần tự module Quản lý các món ăn từng bữa trong ngày	48
Ảnh 2.9. Biểu đồ tuần tự module Quản lý hồ sơ thể chất	49
Ảnh 2.10. Biểu đồ tuần tự module Thông kê dinh dưỡng trong ngày	50
Ảnh 2.11. Biểu đồ tuần tự module Hỏi đáp với chuyên gia AI.....	51
Ảnh 2.12. Biểu đồ tuần tự module Gửi góp ý cho quản trị viên	52
Ảnh 2.13. Biểu đồ tuần tự module Nhận thông báo góp ý của người dùng.....	52
Ảnh 2.14. Biểu đồ tuần tự module Quản lý các món ăn hiển thị cho người dùng...53	53
Ảnh 2.15. Biểu đồ tuần tự module Quản lý các nguyên liệu sẽ dùng cho món ăn ..54	54
Ảnh 2.16. Luồng hoạt động chức năng gợi ý thực đơn.....	55
Ảnh 2.17. Luồng hoạt động chức năng thay đổi món ăn	55
Ảnh 2.18. Luồng hoạt động chức năng chatbot.....	56
Ảnh 3.1. Kiến trúc hệ thống	59
Ảnh 3.2. Giao diện module đăng nhập	60
Ảnh 3.3. Giao diện module đăng ký	61
Ảnh 3.4. Giao diện module quản lý hồ sơ thể chất (1).....	61
Ảnh 3.5. Giao diện module quản lý hồ sơ thể chất (2).....	62

Ảnh 3.6. Giao diện module quản lý hồ sơ thể chất (3).....	62
Ảnh 3.7. Giao diện module quản lý hồ sơ thể chất (4).....	63
Ảnh 3.8. Giao diện module quản lý hồ sơ thể chất (5).....	63
Ảnh 3.9. Giao diện module quản lý lịch trình tập thể dục (1).....	64
Ảnh 3.10. Giao diện module quản lý lịch trình tập thể dục (2).....	64
Ảnh 3.11. Giao diện module quản lý lượng nước uống (1)	65
Ảnh 3.12. Giao diện module quản lý lượng nước uống (2)	65
Ảnh 3.13. Giao diện module quản lý các món ăn từng bữa trong ngày (1)	66
Ảnh 3.14. Giao diện module quản lý các món ăn từng bữa trong ngày (2)	66
Ảnh 3.15. Giao diện module quản lý các món ăn từng bữa trong ngày (3)	67
Ảnh 3.16. Giao diện module thống kê dinh dưỡng trong ngày	67
Ảnh 3.17. Giao diện module hỏi đáp với chuyên gia AI (1)	68
Ảnh 3.18. Giao diện module hỏi đáp với chuyên gia AI (2)	68
Ảnh 3.19. Giao diện module gửi góp ý cho quản trị viên	69
Ảnh 3.20. Giao diện module quản lý các món ăn hiển thị cho người dùng bản mobile (1).....	69
Ảnh 3.21. Giao diện module quản lý các món ăn hiển thị cho người dùng bản mobile (2).....	70
Ảnh 3.22. Giao diện module quản lý các món ăn hiển thị cho người dùng bản web (1).....	70
Ảnh 3.23. Giao diện module quản lý các món ăn hiển thị cho người dùng bản web (2).....	71
Ảnh 3.24. Giao diện module quản lý các món ăn hiển thị cho người dùng bản web (3).....	71
Ảnh 3.25. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản mobile (1).....	72
Ảnh 3.26. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản mobile (2).....	72
Ảnh 3.27. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản web (1).....	73
Ảnh 3.28. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản web (2).....	73
Ảnh 3.29. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản web (3).....	74
Ảnh 3.30. Giao diện module nhận thông báo góp ý của người dùng.....	74

KÝ HIỆU VÀ THUẬT NGỮ

STT	Thuật ngữ, từ viết tắt	Ý nghĩa	Chú thích
1	BMI	Body Mass Index	Chỉ số khối cơ thể
2	BMR	Basal Metabolic Rate	Tỷ lệ trao đổi chất cơ bản
3	TDEE	Total Daily Energy Expenditure	Tổng năng lượng tiêu hao hàng ngày
4	RDA	Recommended Dietary Allowance	Nhu cầu dinh dưỡng khuyến nghị
5	DV	Daily Value	Giá trị hàng ngày
6	LLM	Large Language Model	Mô hình ngôn ngữ lớn
7	RAG	Retrieval-Augmented Generation	Tạo sinh dựa trên truy xuất tăng cường
8	SLSQP	Sequential Least Squares Programming	Thuật toán tối ưu có ràng buộc
9	SDK	Software Development Kit	Bộ công cụ phát triển phần mềm
10	IDE	Integrated Development Environment	Môi trường phát triển tích hợp
11	FCM	Firebase Cloud Messaging	Dịch vụ nhắn tin đám mây Firebase
12	HTTP	Hyper Text Transfer Protocol	Giao thức truyền tải siêu văn bản
13	AI	Artificial Intelligence	Trí tuệ nhân tạo
14	API	Application Programming Interface	Giao diện lập trình ứng dụng
15	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên

LỜI MỞ ĐẦU

Dinh dưỡng có vai trò rất quan trọng trong việc hình thành, phát triển cơ thể và giữ gìn sức khỏe của con người. Ở mỗi thời kỳ phát triển, nhu cầu về dinh dưỡng hoàn toàn khác nhau, tuy nhiên việc đáp ứng nhu cầu ấy một cách hợp lý lại luôn luôn là vấn đề rất quan trọng, vì đó là nền tảng của sức khỏe. Dinh dưỡng hợp lý là thực hiện một chế độ ăn cung cấp đầy đủ năng lượng và đầy đủ, cân đối các chất dinh dưỡng cần thiết, có vai trò cực kỳ quan trọng trong việc đảm bảo sức khỏe cho con người và phòng chống các loại bệnh tật.

Việt Nam đang đổi mới với một cuộc khủng hoảng sức khỏe kép đáng báo động. Tỷ lệ thừa cân, béo phì đã tăng gần gấp đôi trong thập kỷ qua, đặc biệt ở các thành phố lớn và trong nhóm trẻ em, với tỷ lệ hiện nay là 19%. Nguyên nhân sâu xa đến từ sự thay đổi lối sống trong quá trình đô thị hóa, với chế độ ăn nhiều thực phẩm chế biến sẵn, giàu năng lượng và thói quen ít vận động. Hệ quả trực tiếp là gánh nặng khổng lồ từ các bệnh không lây nhiễm như tim mạch, đái tháo đường, ung thư, hiện là nguyên nhân của gần 80% tổng số ca tử vong hàng năm tại Việt Nam. Thực trạng này cho thấy nhu cầu cấp thiết về một giải pháp can thiệp hiệu quả, dễ tiếp cận và có khả năng mở rộng để thúc đẩy lối sống lành mạnh trong cộng đồng.

Tầm nhìn của đề tài “Xây dựng ứng dụng quản lý dinh dưỡng tích hợp chatbot AI” hướng đến là phát triển một ứng dụng hữu ích, đóng vai trò giúp người dùng có thể tự trở thành một chuyên gia dinh dưỡng và huấn luyện viên cá nhân cho bản thân ngay trên điện thoại của người dùng. Để làm được điều này, dự án tập trung vào ba mục tiêu cốt lõi. Thứ nhất, xây dựng một cơ sở dữ liệu dinh dưỡng đầy đủ và chuẩn hóa dành riêng cho các món ăn Việt Nam, dựa trên các nguồn uy tín đồng thời hỗ trợ các đơn vị đo lường quen thuộc với người Việt. Thứ hai, phát triển một hệ thống quản lý thông minh, có khả năng cá nhân hóa thực đơn và kế hoạch luyện tập dựa trên mục tiêu, sở thích và thể trạng riêng của từng người. Mục tiêu quan trọng thứ ba, là tích hợp một chatbot AI thông minh, là nơi trả lời các câu hỏi nhanh của người dùng, giúp giải đáp các thắc mắc trở nên thuận tiện hơn.

Nội dung đồ án bao gồm có 4 chương:

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích yêu cầu và thiết kế hệ thống

Chương 3: Cài đặt và kiểm thử hệ thống

Chương 4: Kết luận và hướng phát triển

Trong quá trình tìm hiểu, phân tích, nhóm không thể tránh được những khiếm khuyết, sai sót. Mong thầy cô và các bạn góp ý để đồ án của nhóm ngày càng hoàn thiện hơn.

Chương 1. CƠ SỞ LÝ THUYẾT

1.1. Kiến thức nền tảng về dinh dưỡng

1.1.1. Chỉ số BMI

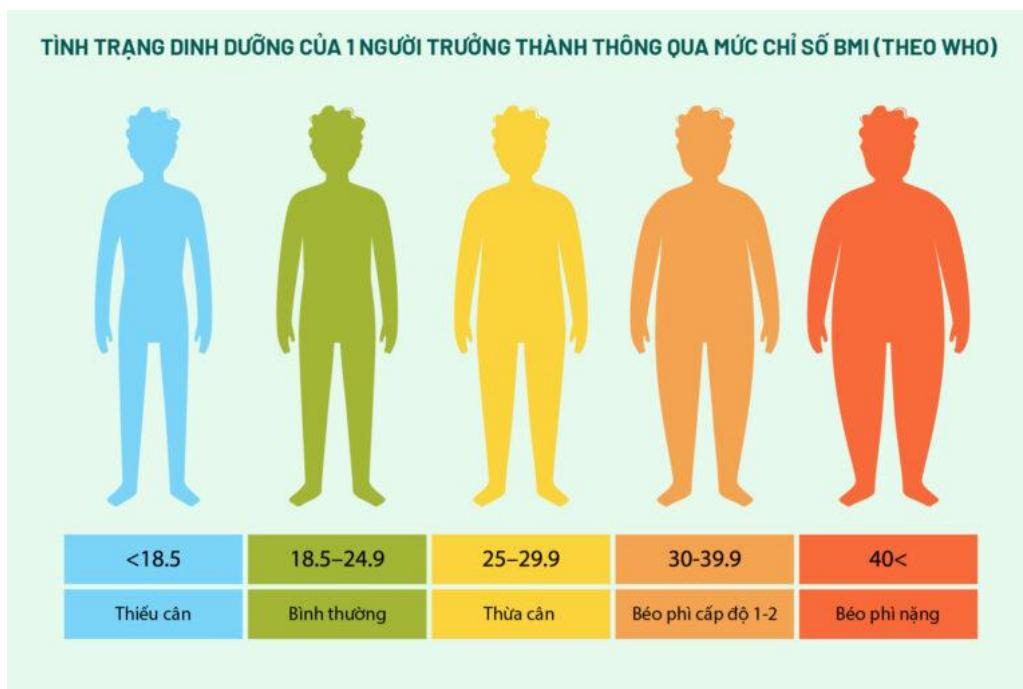
Chỉ số BMI [1] là một công cụ thường được sử dụng để đo lượng mỡ trong cơ thể. Chỉ số BMI chuẩn được tính dựa trên chiều cao và cân nặng, áp dụng cho nam và nữ trưởng thành.

Chỉ số khối cơ thể được tính theo công thức: $BMI = W / [(H)^2]$

Trong đó:

- BMI đơn vị thường dùng là kg/m²
- W là cân nặng (kg)
- H là chiều cao (m)

Một người có chỉ số BMI bình thường sẽ dao động trong khoảng 18,5 - 24,9, con số này cho thấy người đó đang ở mức cân nặng lý tưởng.



Ảnh 1.1. Các mức độ của chỉ số BMI

Chỉ số BMI tăng quá cao sẽ ảnh hưởng xấu đến sức khỏe và gây ra nhiều bệnh lý nguy hiểm. Nhiều vấn đề sức khỏe nghiêm trọng có liên quan đến tình trạng thừa cân hoặc béo phì, bao gồm:

- Bệnh tim mạch
- Bệnh tiểu đường
- Bệnh vú túi mật

- Nguy cơ gây ra một số loại ung thư, chẳng hạn như ung thư nội mạc tử cung, ung thư vú, ung thư đại tràng và túi mật. Ngoài ra, béo phì cũng có thể làm tăng nguy cơ ung thư buồng trứng.
- Chứng ngưng thở khi ngủ
- Bệnh về khớp
- Vô sinh.

1.1.2. Chỉ số BMR

Chỉ số BMR [1] là chỉ số cho biết tỷ lệ trao đổi chất cơ bản trong cơ thể con người. Đây là chỉ số cung cấp cho con người biết mức năng lượng tối thiểu mà cơ thể người cần có, từ đó giúp duy trì được sự ổn định của các chức năng sống của cơ thể như hệ tuần hoàn, hệ tiêu hóa, hệ hô hấp... Việc tính BMR phụ thuộc vào nhiều yếu tố, chỉ số BMR sẽ chiếm khoảng 70% tổng số lượng Calo tiêu hao mỗi ngày.

Quá trình cơ thể mỗi người đốt cháy Calo không phải hầu như xuất phát từ các yếu tố bên ngoài như vận động, di chuyển đi lại... mà là các hoạt động của các bộ phận như não, tim, gan, phổi, hô hấp... cũng đều tiêu hao năng lượng, ngay cả khi đang trong trạng thái nghỉ ngơi cũng tiêu tốn Calo.

Vì vậy, việc đo BMR có thể giúp mọi người theo dõi trọng lượng của cơ thể mình. Tổng lượng Calo cũng như mức độ hoạt động của cơ thể mỗi ngày đều phụ thuộc vào chỉ số BMR. Tức là, nếu muốn tăng cân thì BMR thì phải hấp thụ lượng ăn lớn hơn tổng lượng Calo cần có. Còn nếu muốn giảm cân thì cần phải có chế độ tập luyện và có chế độ ăn phù hợp.

Công thức tính BMR được sử dụng trong ứng dụng là công thức Harris – Benedict. Đây là công thức được sáng chế vào năm 1919, nhưng đã được tiến vào năm 1984, bởi vì các nhà khoa học đã chứng minh được công thức được cải tiến có độ chính xác cao hơn. Công thức Harris - Benedict đó là:

- Đối với nam giới: $BMR = 66 + (13,7 \times \text{trọng lượng cơ thể tính bằng kg}) + (5 \times \text{chiều cao tính bằng cm}) - (6,8 \times \text{tuổi tính theo năm}).$
- Đối với nữ giới: $BMR = 655 + (9,6 \times \text{trọng lượng cơ thể tính bằng kg}) + (1,8 \times \text{chiều cao tính bằng cm}) - (4,7 \times \text{tuổi tính theo năm}).$

Công thức này khá đơn giản và dễ thực hiện. Nhưng đối với người có cơ bắp to hoặc có cân nặng quá mức, thì chỉ số BMR chuẩn áp dụng công thức Harris - Benedict sẽ cho ra kết quả không quá chính xác, thường thì sẽ có kết quả thấp hơn so với thực tế. Bởi người có càng nhiều cơ bắp thì lượng Calo tiêu tốn sẽ nhiều hơn người bình thường.

1.1.3. Chỉ số TDEE

TDEE [1] là tổng năng lượng (calo) tiêu hao trong một ngày để phục vụ cho nhiều mục đích khác nhau như làm việc, học tập, vui chơi, ngủ và tập luyện.

Về cơ bản, quy tắc giảm cân là lượng calo tiêu hao nhiều hơn lượng calo nạp vào mỗi ngày. Để đạt được mức hao hụt calo, trước tiên cần phải tính toán lượng năng lượng cơ thể cần nạp vào và chỉ số TDEE sẽ cho biết chính xác lượng calo nạp trong ngày. Từ đó, sẽ có kế hoạch vận động, ăn uống phù hợp. Nếu nạp calo ít hơn chỉ số TDEE thì sẽ giảm cân. Nếu tiêu thụ nhiều calo hơn chỉ số TDEE, cân nặng sẽ tăng lên theo thời gian.

TDEE được tạo thành từ nhiều nguồn năng lượng khác nhau. Và chỉ số này còn phụ thuộc vào nhiều yếu tố. Vì vậy TDEE của mỗi người sẽ khác nhau. Ngay cả một người cũng có chỉ số TDEE mỗi ngày khác nhau tùy thuộc vào mức độ vận động trong ngày hôm đó. TDEE bao gồm các nguồn năng lượng sau: BMR, TEF, EAT và NEAT. Trong đó:

- BMR: Chỉ số này cho biết lượng năng lượng tối thiểu mà cần có để duy trì sự sống. BMR thường chiếm khoảng 70% tổng số mức năng lượng của TDEE.
- TEF (Thermic Effect of Foods): Đây là lượng năng lượng mà cơ thể sử dụng để tiêu hóa, hấp thụ và lưu trữ các chất dinh dưỡng trong thực phẩm. TEF nói chung chiếm khoảng 10% tổng mức năng lượng của TDEE.
- EAT (Exercise Activity Thermogenesis - Lượng năng lượng tiêu hao thông qua tập thể dục): EAT chỉ chiếm khoảng 5% tổng mức năng lượng của TDEE.
- NEAT (Non-exercise Activity Thermogenesis - Năng lượng tiêu hao không phải từ tập thể dục): Năng lượng trong NEAT đến từ các hoạt động nhẹ nhàng như đứng, ngồi, đi lại trong nhà,... NEAT chiếm khoảng 15% tổng mức năng lượng của TDEE.

Có hai cách tính TDEE, đó là tính thông qua hệ số luyện tập và tổng số nguồn calo được đốt cháy.

Cách tính TDEE thông qua hệ số vận động:

Để tính TDEE thông qua hệ số vận động, sử dụng công thức:

$$\text{TDEE} = \text{BMR} \times R.$$

Hệ số vận động không có tên gọi cụ thể được ký hiệu cố định là R. Hệ số này phụ thuộc vào tần suất và mức độ luyện tập của mỗi người, cụ thể:

- Người ít vận động, làm công việc nhẹ nhàng và không tập thể dục: $R = 1.2$
- Người ít vận động, tập thể dục dưới 3 lần/tuần: $R = 1.375$
- Người vận động mỗi ngày, tập thể dục 3-5 lần/tuần: $R = 1.55$
- Người thường xuyên vận động, tập thể dục 6-7 lần/tuần: $R = 1.725$
- Người rất năng động, vận động mạnh, tập 2 lần/ngày: $R = 1.9$

Cách tính TDEE thông qua năng lượng sử dụng:

Để tính TDEE theo năng lượng sử dụng, sử dụng công thức:

$$\text{TDEE} = \text{BMR} + \text{TEF} + \text{EAT} + \text{NEAT}$$

Công thức này khá phức tạp vì phải tính 4 chỉ số BMR, TEF, EAT, NEAT. Ngoài ra, cách tính này có nhược điểm là khó tính chính xác toàn bộ lượng calo đốt cháy trong ngày. Đồng thời, cần tính toán hàng ngày vì mỗi ngày sẽ có những hoạt động khác nhau. Do đó, cách đơn giản và phổ biến nhất để tính TDEE là theo hệ số vận động.

1.1.4. Sự phát triển của quy tắc Wishnofsky

Quy tắc Wishnofsky [2] được phát triển vào năm 1958 bởi Max Wishnofsky. Trong một báo cáo, ông đã đặt câu hỏi: "Tương đương calo của một pound trọng lượng cơ thể tăng hoặc mất là gì?"

Sau khi phân tích các tài liệu khoa học hiện có vào thời điểm đó, Wishnofsky đã đi đến kết luận rằng "tương đương calo của một pound trọng lượng cơ thể mất" hoặc "tăng sẽ là 3500 kcal".

Wishnofsky đưa ra kết luận này dựa trên hai nguồn chính:

1. Phân tích hóa học mô mỡ: Dựa trên một phân tích từ năm 1911 cho thấy mô của con người chứa 87% là "chất béo" (triglyceride). Wishnofsky tính toán rằng một pound (khoảng 454g) mô mỡ có hàm lượng năng lượng khoảng 3750 kcal.
2. Các nghiên cứu ăn kiêng: Ông đã xem xét các nghiên cứu về giảm cân bằng chế độ ăn rất ít calo từ những năm 1930. Từ dữ liệu này, ông tính toán được hàm lượng năng lượng của trọng lượng thay đổi là 3500 kcal/pound.

Ông nhận thấy "sự trùng khớp đáng kinh ngạc" giữa hai giá trị này và từ đó, quy tắc 3500 kcal/pound đã ra đời. Quy tắc này nhanh chóng trở nên phổ biến, được trích dẫn rộng rãi trong cả tài liệu khoa học lẫn báo chí đại chúng, với nội dung cốt lõi là: cần thâm hụt khoảng 3500 kcal để giảm 1 pound (0.45 kg) trọng lượng cơ thể.

Dựa trên những hiểu biết khoa học hiện đại ngày nay về trao đổi chất và cân bằng năng lượng, các chuyên gia đã nhận định về quy tắc Wishnofsky:

- Quy tắc Wishnofsky bị coi là thiếu vì bỏ qua các yếu tố sinh lý thực tế trong quá trình giảm cân. Khi ăn kiêng, cơ thể giảm tốc độ trao đổi chất, làm mức tiêu hao năng lượng giảm và khiến tốc độ giảm cân chậm hơn dự đoán. Ngoài ra, trọng lượng mất đi ban đầu chủ yếu là glycogen, protein và nước chứ không hoàn toàn là mỡ, nên giá trị năng lượng thực tế thấp hơn nhiều. Kết quả là quy tắc này thường dự đoán quá mức mức giảm cân, đặc biệt khi áp dụng trong thời gian dài.

- Mặc dù quy tắc Wishnofsky đơn giản, nhưng thiếu nền tảng khoa học hiện đại và thường dự đoán sai mức giảm cân. Một cách tiếp cận hiệu quả hơn là sử dụng các mô hình giảm cân động đã được xác thực, như của Thomas hay Hall, vốn tính đến thay đổi trao đổi chất, thành phần cơ thể, tuổi tác, giới tính và mức thâm hụt calo. Các mô hình này dựa trên định luật nhiệt động lực học, đưa ra dự đoán thực tế về quỹ đạo giảm cân theo thời gian. Chuyên gia dinh dưỡng nên áp dụng các mô hình này qua ứng dụng, trang web hoặc thiết bị điện tử để giúp bệnh nhân có kỳ vọng giảm cân chính xác và tránh thất vọng khi tuân thủ chế độ ăn.

Do thế, khi sử dụng quy tắc Wishnofsky như một "mục tiêu khởi động"—nhắm tới việc thâm hụt khoảng 500 calo mỗi ngày để bắt đầu quá trình. Tuy nhiên, thay vì mong đợi kết quả tuyến tính (giảm 1 pound/tuần đều đặn), hãy áp dụng khái niệm cốt lõi của "quy tắc động" như một "sự điều chỉnh thực tế". Điều này có nghĩa là phải chấp nhận rằng khi giảm cân, cơ thể sẽ thích nghi và quá trình trao đổi chất sẽ chậm lại; tốc độ giảm cân *chắc chắn* sẽ giảm dần. Vì vậy, để duy trì đà giảm cân, cần chuẩn bị tinh thần để đánh giá lại và điều chỉnh (thường là giảm thêm) lượng calo nạp vào hoặc tăng cường vận động sau mỗi vài tuần khi bạn nhận thấy cân nặng của mình bắt đầu chững lại.

1.1.5. Lượng nước cần thiết hàng ngày

Uống đủ nước mỗi ngày [3] là một trong những yếu tố quan trọng để duy trì sức khỏe tốt và cơ thể hoạt động hiệu quả. Tuy nhiên, nhiều người thường mắc rằng mỗi ngày cần uống bao nhiêu nước là đủ. Con số phổ biến “2 lít nước mỗi ngày” thực chất chỉ là một mức trung bình mang tính tham khảo.

Để tính toán chính xác lượng nước cần thiết cho cơ thể, có thể dựa vào cân nặng. Công thức đơn giản và dễ nhớ là:

$$\text{Lượng nước cơ bản hằng ngày (lít)} = \text{cân nặng (kg)} \times 0.03 \text{ lít}$$

Đối với những người thường xuyên vận động, luyện tập thể thao hoặc lao động nặng, cơ thể sẽ cần nhiều nước hơn để bù lại lượng nước mất qua mồ hôi. Theo Hệ thống Đại học Missouri (Mỹ), bạn cần bổ sung thêm 360ml nước (khoảng 12 oz) cho mỗi 30 phút luyện tập.

Ngoài ra, thai kỳ là quá trình quan trọng của mỗi người mẹ, ngoài việc bổ sung đầy đủ chất dinh dưỡng bà bầu cũng nên bổ sung đủ nước cần cho cả mẹ và bé mỗi ngày. Trong thời kỳ mang thai cơ thể mẹ cần bổ sung nhiều lượng nước hơn so với người thường bởi nguồn nước được cung cấp cho cả mẹ và bé.

Để dễ hiểu, mọi người có thể dùng công thức tổng quát sau:

$$\text{Lượng nước cần uống (lít)} = \text{Cân nặng (kg)} \times 0.03 \text{ lít} + 0.3 \text{ lít nếu mang thai} + \text{nước bù thêm.}$$

Mức độ hoạt động	Nước bù thêm
Ít vận động (1.2)	0 lít
Hoạt động nhẹ (1.375)	+0.4 lít/ngày
Hoạt động vừa (1.55)	+0.65 lít/ngày
Hoạt động nặng (1.725)	+0.9 lít /ngày
Rất nặng (1.9)	+1.25 lít/ngày

Bảng 1.1.Lượng nước cần bù thêm so với mức độ hoạt động

1.1.6. Nhu cầu dinh dưỡng khuyến nghị

Nhu cầu dinh dưỡng khuyến nghị [3] là lượng ăn vào của một chất dinh dưỡng đảm bảo đáp ứng nhu cầu hành ngày về chất đó cho hầu hết (97,5%) các cá thể trong quần thể khỏe mạnh, mức nhu cầu này được tính theo tuổi, giới, tầm vóc, tình trạng sinh lý và loại hình lao động. Bảng Nhu cầu dinh dưỡng khuyến nghị cho người dân của một quốc gia là những khuyến nghị về nhu cầu năng lượng và các dưỡng chất theo lứa tuổi, giới tính, loại hình lao động và tình trạng sinh lý và theo thói quen ăn (uống) cho người dân nước đó. Bảng Nhu cầu dinh dưỡng khuyến nghị là cơ sở khoa học cho việc xây dựng cơ cấu bữa ăn của người dân một nước.

Cơ sở khoa học để xây dựng “Nhu cầu Dinh dưỡng Khuyến nghị - RDA” của một quốc gia bao gồm:

- Dựa theo các khuyến nghị của các tổ chức uy tín trên thế giới như: Tổ chức Y tế Thế giới (World Health Organization – WHO), Tổ chức Nông nghiệp và Thực phẩm Liên Hiệp Quốc (Food and Agriculture Organization – FAO), Quỹ Nhi đồng Liên hợp quốc (United Nations Children’s Fund – UNICEF).
- Tham khảo các khuyến nghị của các nước cùng khu vực (Đông Nam Á, Châu Á...) và đặc biệt là các quốc gia lân cận.
- Căn cứ trên tình hình an ninh thực phẩm quốc gia.
- Dựa vào tầm vóc, cách ăn uống, lối sống, điều kiện lao động và mô hình các bệnh lý liên quan đến lối sống trong nước và trên thế giới.

1.1.7. Phân mức trên nhãn thông tin dinh dưỡng

% Giá trị hàng ngày (%DV) [4] cho biết mức độ đóng góp của từng chất dinh dưỡng trong một khẩu phần ăn vào tổng chế độ ăn hàng ngày. Khi gắn nhãn cho các món ăn, %DV giúp bạn xác định nhanh chóng xem một món ăn có hàm lượng chất dinh dưỡng cao hay thấp, từ đó cung cấp thông tin rõ ràng cho người tiêu dùng. Giá trị hàng ngày là lượng tham khảo (tính bằng gam, miligam hoặc microgam) chất dinh dưỡng cần tiêu thụ hoặc không được vượt quá mỗi ngày.

%DV có thể giúp bạn đưa ra lựa chọn sáng suốt về những gì bạn ăn và uống:

- So sánh Thực phẩm : %DV giúp bạn dễ dàng so sánh. Chỉ cần đảm bảo khẩu phần ăn của mỗi sản phẩm là như nhau. Hãy xem %DV của các chất dinh dưỡng khác nhau và thường xuyên chọn những thực phẩm chứa nhiều chất dinh dưỡng bạn muốn bổ sung và ít chất dinh dưỡng bạn muốn hạn chế.
- Quản lý sự cân bằng dinh dưỡng : Bạn không cần phải từ bỏ một món ăn yêu thích để có một chế độ ăn uống lành mạnh. Hãy sử dụng %DV để cân bằng dinh dưỡng với các loại thực phẩm khác trong ngày. Khi một món ăn bạn thích có hàm lượng dinh dưỡng cao mà bạn muốn ăn ít hơn—hoặc ít hơn một món ăn mà bạn muốn ăn nhiều hơn—you có thể cân bằng nó với các món ăn có hàm lượng dinh dưỡng thấp (hoặc cao) vào các thời điểm khác trong ngày.

Sử dụng %DV để xác định xem một khẩu phần ăn có hàm lượng dinh dưỡng cao hay thấp thì có một hướng dẫn chung:

- 5% DV hoặc ít hơn của một chất dinh dưỡng trên mỗi khẩu phần được coi là thấp.
- 20% DV hoặc hơn giá trị dinh dưỡng cho mỗi khẩu phần ăn được coi là cao.

Một số chất dinh dưỡng không có %DV

- Protein thường không có phần trăm Giá trị hàng ngày (%DV) được liệt kê trên nhãn, vì vậy hãy sử dụng số gam (g) làm hướng dẫn.
- *Chất béo chuyển hóa* và tổng lượng đường không có phần trăm Giá trị hàng ngày (%DV), vì vậy hãy sử dụng số gam (g) để so sánh và lựa chọn thực phẩm.

1.1.8. USDA FoodData Central

USDA FoodData Central [5] là nguồn dữ liệu tổng hợp về thành phần thực phẩm của USDA, bao gồm nhiều loại dữ liệu khác nhau.

Nguồn cung thực phẩm đang tăng với tốc độ đáng kinh ngạc, đồng thời thành phần của các sản phẩm nông nghiệp và thực phẩm chế biến cũng thay đổi liên tục. Bên cạnh đó, hiểu biết khoa học về mối quan hệ giữa chế độ ăn uống và sức khỏe đang phát triển nhanh chóng, đòi hỏi mức độ kiến thức về thực phẩm và các thành phần của chúng cao hơn. Những nhu cầu này đã làm tăng yêu cầu về thông tin minh bạch và dễ tiếp cận về chất dinh dưỡng và các thành phần khác của thực phẩm.

Các nguồn dữ liệu về thành phần thực phẩm của USDA đang phát triển để đáp ứng nhu cầu về thành phần cũng như nhu cầu của một nhóm người dùng ngày càng đa dạng, bao gồm các nhà nghiên cứu, nhà hoạch định chính sách, chuyên gia dinh dưỡng và sức khỏe, và các nhà phát triển sản phẩm. Điều này dẫn đến việc phát triển và ra mắt FoodData Central vào năm 2019. FoodData Central cung cấp một hệ thống dữ liệu dựa trên web, bao gồm năm loại dữ liệu thành phần thực phẩm và dinh dưỡng

khác nhau, mỗi loại có mục đích riêng và được thu thập theo những phương pháp khác nhau:

- Foundation Foods – Dữ liệu về chất dinh dưỡng của thực phẩm cơ bản (chưa hoặc chế biến nhẹ) với metadata chi tiết về mẫu, địa điểm, thời gian, và phương pháp phân tích.
- SR Legacy – Cơ sở dữ liệu truyền thống, tổng hợp từ phân tích, ước lượng, và tài liệu công bố; không cập nhật sau 2018.
- FNDDS (Food and Nutrient Database for Dietary Studies) – Giá trị dinh dưỡng cho thực phẩm trong khảo sát NHANES, phục vụ phân tích chế độ ăn và nghiên cứu dinh dưỡng.
- Experimental Foods – Dữ liệu về thực phẩm thử nghiệm hoặc nghiên cứu đặc biệt, cung cấp thông tin về ảnh hưởng của các yếu tố sản xuất và nghiên cứu đến thành phần dinh dưỡng.
- Branded Foods (USDA Global Branded Food Products Database) – Dữ liệu dinh dưỡng từ các sản phẩm thương hiệu và nhãn riêng, thu thập qua hợp tác công-tư với ngành thực phẩm.

1.2. Kiến thức nền tảng về AI

1.2.1. LLM

1.2.1.1. Khái niệm

LLM [6] là một loại mô hình ngôn ngữ sử dụng kỹ thuật học sâu được huấn luyện trên tập dữ liệu văn bản cực lớn, với số lượng tham số rất cao có thể lên tới hàng nghìn tỷ. Những mô hình này có khả năng hiểu và sinh ngôn ngữ tự nhiên, tức là có thể dự đoán từ hoặc chuỗi từ tiếp theo dựa trên ngữ cảnh đã học từ khối dữ liệu khổng lồ.

1.2.1.2. Kiến trúc cơ bản của LLM

Kiến trúc của LLM bao gồm nhiều lớp mạng nơ-ron riêng biệt hoạt động cùng với nhau để xử lý văn bản đầu vào và tạo nội dung đầu ra, bao gồm:

- Embedding Layer: Chuyển các token (từ hoặc phân đoạn nhỏ hơn) của văn bản đầu vào thành vector nhiều chiều. Các vector này nắm giữ thông tin ngữ nghĩa, cú pháp của từng từ hay token đó trong câu.
- Feedforward Layer: Là các lớp nơ-ron được kết nối đầy đủ, áp dụng các phép biến đổi phi tuyến trên đầu ra của các lớp trước đó để tạo ra các biểu diễn từ hay câu có chiều sâu, trừu tượng, giàu thông tin hơn.
- Recurrent Layer: Được thiết kế để xử lý thông tin tuần tự, tạo ra các biểu diễn từ có tính tuần tự, phụ thuộc vào ngữ cảnh. Từ đó cho phép mô hình hiểu và nắm bắt mối quan hệ phức tạp giữa các từ trong câu.

- Attention Mechanism: Đây là cơ chế giúp mô hình tập trung vào các phần quan trọng của đầu vào trong khi tạo đầu ra. Nó giúp mô hình chú ý đến các phần khác nhau của ngữ cảnh để tạo ra các dự đoán chính xác hơn.

1.2.1.3. Cách hoạt động của LLM

LLM hoạt động dựa trên các mạng nơ-ron sâu, trong đó phổ biến nhất là kiến trúc Transformer. Quá trình xử lý của một LLM thường bao gồm các bước: mã hóa đầu vào, xử lý nội bộ bằng các lớp Transformer, và giải mã để sinh đầu ra.

Trước tiên, văn bản đầu vào được chuyển đổi thành các đơn vị nhỏ hơn (tokens). Mỗi token được nhúng thành các vector số có ý nghĩa về mặt ngữ nghĩa. Các vector này được đưa vào nhiều lớp Transformer nhằm phân tích mối quan hệ giữa các từ trong câu. Trong các lớp Transformer, mô hình áp dụng cả phép toán tuyến tính và phi tuyến tính thông qua các lớp Feedforward và đặc biệt sử dụng cơ chế Attention để xác định những phần thông tin quan trọng trong chuỗi đầu vào.

Nhờ cơ chế Attention, mô hình có thể hiểu ngữ cảnh xa, nắm bắt mối liên hệ giữa các từ, cụm từ hoặc câu trong văn bản. Sau đó, ở giai đoạn giải mã, mô hình sẽ dự đoán token tiếp theo trên toàn bộ ngữ cảnh đã xử lý. Quá trình này được lặp lại cho đến khi tạo thành câu trả lời hoàn chỉnh.

Thông qua quá trình huấn luyện trên dữ liệu lớn và có thể được fine-tuning theo từng tác vụ cụ thể, LLM học được quy luật ngôn ngữ, cách diễn đạt và các mẫu thông tin trong văn bản. Nhờ đó, mô hình có khả năng thực hiện nhiều nhiệm vụ khác nhau như: tạo mới văn bản, tóm tắt, dịch ngôn ngữ, phân tích ngữ nghĩa,... với độ chính xác ngày càng cao.

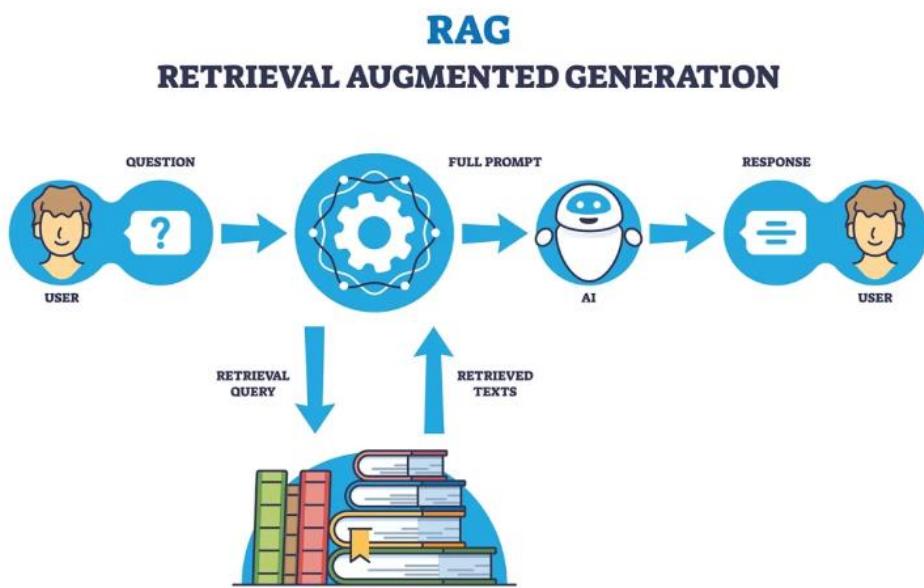
1.2.1.4. Kết luận

Trong dự án này, LLM được sử dụng nhằm đáp ứng yêu cầu hiểu và xử lý các truy vấn ngôn ngữ tự nhiên đa dạng của người dùng. Người dùng có thể đặt câu hỏi hoặc yêu cầu theo nhiều cách diễn đạt khác nhau, do đó cần một mô hình có khả năng nắm bắt ngữ cảnh và ý nghĩa của câu hỏi một cách linh hoạt. Bên cạnh đó, LLM có khả năng suy luận và tổng hợp thông tin, giúp hệ thống đưa ra các gợi ý và phản hồi mang tính logic, mạch lạc và gần với cách giao tiếp của con người. Điều này đặc biệt phù hợp với chức năng chatbot và tư vấn dinh dưỡng trong ứng dụng, góp phần nâng cao trải nghiệm người dùng và tính thực tiễn của hệ thống.

1.2.2. RAG

1.2.2.1. Khái niệm

RAG [7] là một phương pháp kết hợp LLM và hệ thống truy xuất thông tin. Thay vì để LLM dựa hoàn toàn vào kiến thức học sẵn, RAG cho phép truy xuất thông tin từ cơ sở dữ liệu bên ngoài để tạo điều kiện xác.



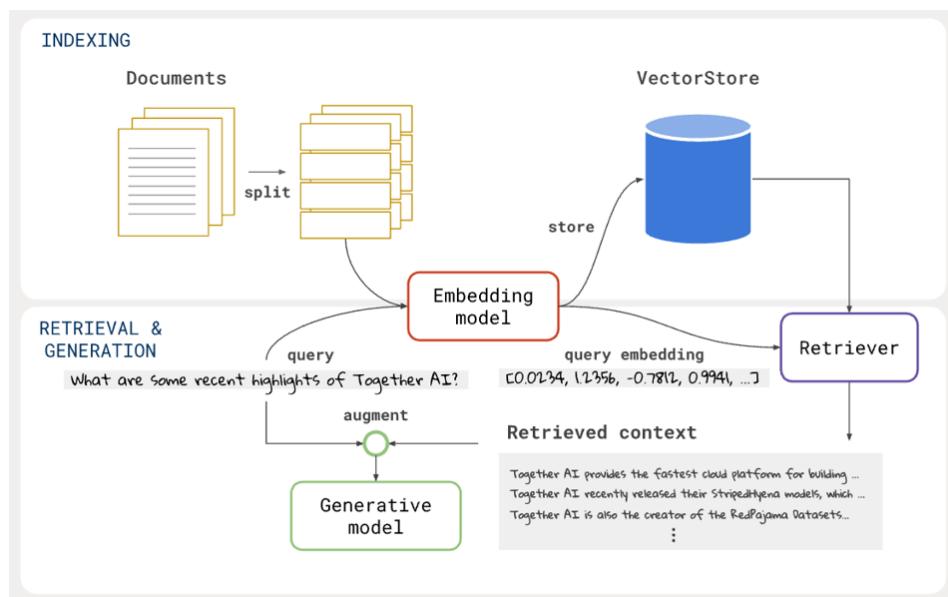
Ảnh 1.2. Kiến trúc RAG

Có thể hiểu RAG bằng cách kết hợp hai thành phần:

- Retrieval: Tìm kiếm, truy xuất thông tin từ nguồn dữ liệu bên ngoài như tài liệu hay cơ sở dữ liệu.
- Generation: Sử dụng LLM kết hợp với thông tin truy xuất được để sinh câu trả lời chính xác.

1.2.2.2. Cách hoạt động của RAG

RAG hoạt động dựa trên sự kết hợp giữa truy xuất thông tin và sinh nội dung, gồm hai giai đoạn chính: chuẩn bị dữ liệu và xử lý truy vấn người dùng.



Ảnh 1.3. Cách hoạt động của RAG

Trong giai đoạn chuẩn bị dữ liệu, các tài liệu văn bản thô được chia nhỏ thành các đoạn ngắn, sau đó được chuyển đổi thành vector embedding biểu diễn ý nghĩa ngữ nghĩa. Các vector này được lưu trữ trong cơ sở dữ liệu vector, cho phép hệ thống thực hiện tìm kiếm tương đồng nhanh và chính xác theo ngữ nghĩa.

Trong giai đoạn xử lý truy vấn, câu hỏi của người dùng được chuyển thành vector và so sánh với các vector đã lưu để truy xuất các đoạn thông tin liên quan nhất. Các thông tin này được kết hợp với câu hỏi gốc để tạo thành một prompt mở rộng, cung cấp ngữ cảnh cho mô hình ngôn ngữ lớn. Cuối cùng, mô hình ngôn ngữ sử dụng prompt này để sinh ra câu trả lời có độ chính xác cao và bám sát dữ liệu thực tế.

Nhờ cơ chế này, RAG giúp mô hình ngôn ngữ trả lời chính xác hơn, giảm phụ thuộc vào tri thức huấn luyện sẵn và tăng khả năng cập nhật, mở rộng tri thức theo dữ liệu thực tế.

1.2.2.3. Lựa chọn mô hình

Trong hệ thống RAG, việc lựa chọn mô hình Embedding và mô hình ngôn ngữ lớn (LLM) có ảnh hưởng trực tiếp đến chất lượng truy xuất và độ chính xác của câu trả lời.

Trong dự án này, mô hình Alibaba-NLP/gte-multilingual-base được sử dụng để tạo vector embedding cho văn bản. Đây là mô hình embedding đa ngôn ngữ, hỗ trợ tốt tiếng Việt và có khả năng biểu diễn ngữ nghĩa hiệu quả, giúp nâng cao độ chính xác của tìm kiếm ngữ nghĩa trong Vector Database.

Mô hình DeepSeek v3.2 được sử dụng để sinh câu trả lời trong kiến trúc RAG. Mô hình này có khả năng hiểu ngữ cảnh và tạo phản hồi mạch lạc dựa trên thông tin truy xuất được, giúp hạn chế sai lệch thông tin và cải thiện chất lượng phản hồi của hệ thống.

Việc kết hợp hai mô hình trên giúp hệ thống RAG hoạt động hiệu quả, cân bằng giữa khả năng truy xuất ngữ nghĩa và sinh ngôn ngữ tự nhiên, phù hợp với các ứng dụng chatbot và gợi ý thông minh trong dự án.

1.2.2.4. Kết luận

Trong dự án ứng dụng gợi ý thực đơn và chatbot tư vấn dinh dưỡng, việc sử dụng RAG là cần thiết nhằm khắc phục những hạn chế của mô hình ngôn ngữ lớn truyền thống khi làm việc với dữ liệu thực tế. Hệ thống cần xử lý và trả lời các câu hỏi dựa trên dữ liệu dinh dưỡng, món ăn và quy tắc tính toán được lưu trữ riêng, thay vì chỉ dựa vào kiến thức tổng quát đã được huấn luyện sẵn trong mô hình.

RAG cho phép hệ thống truy xuất trực tiếp thông tin liên quan từ cơ sở dữ liệu (như thành phần dinh dưỡng, lượng calo, nguyên liệu, quy tắc gợi ý thực đơn) và kết hợp với khả năng suy luận của LLM để tạo ra câu trả lời chính xác, có căn cứ. Điều này giúp giảm hiện tượng sinh thông tin sai lệch (hallucination), đồng thời đảm bảo nội dung phản hồi luôn nhất quán với dữ liệu của hệ thống.

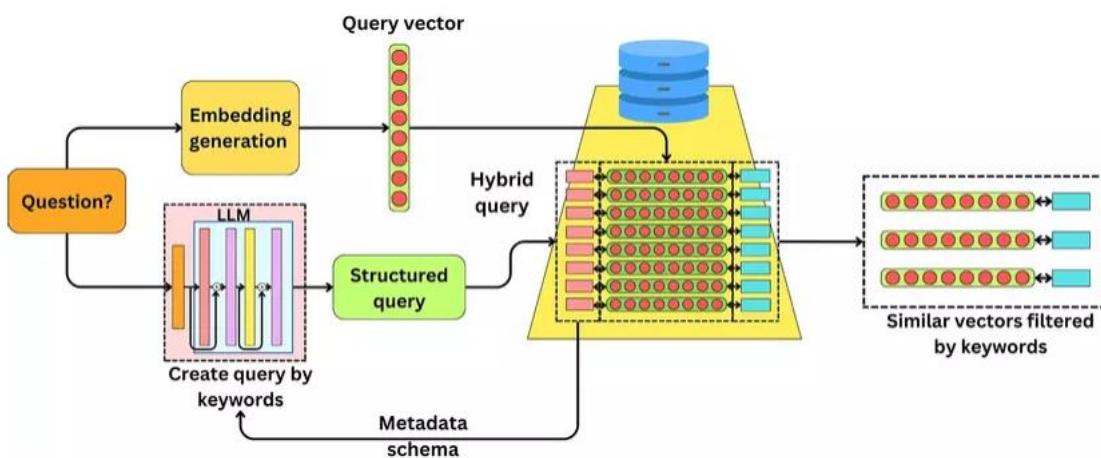
Ngoài ra, RAG giúp hệ thống linh hoạt trong việc cập nhật tri thức, khi chỉ cần thay đổi hoặc bổ sung dữ liệu trong kho lưu trữ mà không cần huấn luyện lại mô hình. Đây là yếu tố quan trọng giúp dự án dễ mở rộng, tiết kiệm tài nguyên và phù hợp với các ứng dụng AI trong môi trường thực tế.

1.2.3. Self-Query Retriever

1.2.3.1. Khái niệm

Trong các hệ thống tìm kiếm dựa trên AI, việc truy xuất thông tin chính xác từ tập dữ liệu lớn là một thách thức lớn, đặc biệt khi truy vấn của người dùng được biểu đạt dưới dạng ngôn ngữ tự nhiên. Mặc dù tìm kiếm ngữ nghĩa giúp hiểu được ý nghĩa của truy vấn, nhưng nó gặp hạn chế khi cần kết hợp với các điều kiện lọc có cấu trúc như số lượng, khoảng giá trị hoặc thời gian. Để giải quyết vấn đề này, Self-Query Retriever được đề xuất như một giải pháp hiệu quả.

Self-Query Retriever [7] là một cơ chế cho phép kết hợp sức mạnh của LLM với tìm kiếm vector và lọc metadata có cấu trúc. Thay vì chỉ dựa vào việc so khớp nội dung văn bản, Self-Query Retriever sử dụng LLM để phân tích truy vấn ngôn ngữ tự nhiên của người dùng, từ đó tự động trích xuất các điều kiện lọc liên quan đến metadata.



Ảnh 1.4. Cách hoạt động Self-Query Retriever

Cụ thể, khi người dùng đưa ra một truy vấn phức tạp, LLM sẽ xác định các thuộc tính có cấu trúc được ngầm hiểu trong câu hỏi (ví dụ: loại, khoảng giá trị, điều kiện so sánh), sau đó chuyển đổi chúng thành một truy vấn có cấu trúc mà hệ thống tìm kiếm (như Elasticsearch) có thể xử lý hiệu quả. Truy vấn này được áp dụng đồng thời lên vector embeddings và metadata của tài liệu, giúp kết quả tìm kiếm vừa đúng ngữ nghĩa vừa thỏa mãn các ràng buộc cụ thể.

Trong bối cảnh các hệ thống RAG, Self-Query Retriever đóng vai trò quan trọng trong việc nâng cao độ chính xác của bước truy xuất, giúp LLM nhận được tập tài liệu liên quan và phù hợp hơn để sinh câu trả lời. Nhờ đó, hệ thống không chỉ cải thiện chất lượng phản hồi mà còn mang lại trải nghiệm tìm kiếm tự nhiên, linh hoạt và thân thiện hơn cho người dùng.

1.2.3.2. Kết luận

Trong phạm vi dự án, dữ liệu không chỉ bao gồm nội dung văn bản thuần túy mà còn đi kèm nhiều thuộc tính có cấu trúc như lượng calo, thành phần dinh dưỡng, loại món ăn, thời điểm sử dụng hoặc các điều kiện ràng buộc cụ thể. Người dùng thường đặt câu hỏi dưới dạng ngôn ngữ tự nhiên nhưng ngầm chứa các điều kiện lọc này, ví dụ như “món ăn ít calo cho bữa tối” hoặc “thực đơn giàu protein dưới 500 kcal”.

Việc sử dụng Self-Query Retriever giúp hệ thống tự động chuyển đổi các truy vấn ngôn ngữ tự nhiên đó thành truy vấn có cấu trúc, kết hợp đồng thời tìm kiếm ngữ nghĩa và lọc metadata. Nhờ đó, hệ thống truy xuất được tập tài liệu phù hợp hơn so với việc chỉ sử dụng Semantic Search thông thường, vốn không xử lý tốt các ràng buộc số liệu và điều kiện so sánh.

Bên cạnh đó, Self-Query Retriever cho phép hệ thống mở rộng linh hoạt mà không cần xây dựng thủ công các bộ lọc cho từng loại truy vấn, giảm độ phức tạp trong thiết kế và tăng khả năng thích ứng với nhiều dạng câu hỏi khác nhau. Điều này đặc biệt phù hợp với các ứng dụng RAG, nơi chất lượng truy xuất quyết định trực tiếp đến độ chính xác của câu trả lời do LLM sinh ra.

1.3. Thuật toán SLSQP

1.3.1. Tổng quan

SLSQP [8] là một phương pháp tối ưu hóa phi tuyến có ràng buộc, thuộc họ Sequential Quadratic Programming. Thuật toán được phát triển từ những năm 1980 bởi Dieter Kraft và hiện nay vẫn được sử dụng rộng rãi trong các thư viện tối ưu như Scipy.

Ý tưởng cốt lõi của SLSQP là giải bài toán tối ưu phi tuyến ban đầu bằng cách lặp qua một chuỗi các bài toán con dạng Quadratic Programming. Ở mỗi bước lặp:

- Hàm mục tiêu được xấp xỉ bậc hai thông qua khai triển Taylor.
- Các ràng buộc phi tuyến được tuyến tính hóa quanh nghiệm hiện tại.
- Một bài toán QP được giải để tìm hướng cập nhật nghiệm.

Trong quá trình lặp, SLSQP sử dụng các phương pháp quasi-Newton để xấp xỉ ma trận Hessian của hàm Lagrangian, giúp giảm chi phí tính toán so với việc tính Hessian chính xác. Thuật toán tiếp tục lặp cho đến khi thỏa mãn các điều kiện hội tụ về tối ưu và khả thi.

SLSQP đặc biệt phù hợp cho các bài toán cỡ vừa, có số biến và ràng buộc không quá lớn, và yêu cầu các hàm mục tiêu cũng như ràng buộc phải trơn và khả vi liên tục.

1.3.2. Dạng bài áp dụng

SLSQP được thiết kế để giải bài toán Nonlinear Programming tổng quát có dạng:

$$\min_{x \in R^n} f(x)$$

Với các ràng buộc:

- Ràng buộc đẳng thức: $c_i(x) = 0, i = 1, \dots, m_e$
- Ràng buộc bất đẳng thức: $c_i(x) \geq 0, i = m_e + 1, \dots, m$
- Ràng buộc biên cho biến: $l_i \leq x_i \leq u_i, i = 1, \dots, n$

Trong đó:

- x là vector biến tối ưu liên tục.
- $f(x)$ là hàm mục tiêu phi tuyến.
- $c(x)$ là tập các hàm ràng buộc.
- l, u lần lượt là cận dưới và cận trên của biến.

1.3.3. Mô hình hóa bài toán gợi ý khẩu phần ăn

Xét một ngày gồm B bữa ăn. Hệ thống lựa chọn n món ăn cho các bữa trong ngày. Gọi $x_i^{(b)}$ là khẩu phần (số suất) của món thứ i được sử dụng trong bữa ăn b . Với mỗi món ăn i , các giá trị dinh dưỡng trên một suất bao gồm năng lượng E_i , protein P_i , carbohydrate C_i và chất béo F_i .

Tổng giá trị dinh dưỡng trong ngày được xác định như sau:

$$E_{day}(x) = \sum_{b=1}^B \sum_i E_i x_i^{(b)}, \quad P_{day}(x) = \sum_{b=1}^B \sum_i P_i x_i^{(b)},$$

$$C_{day}(x) = \sum_{b=1}^B \sum_i C_i x_i^{(b)}, \quad F_{day}(x) = \sum_{b=1}^B \sum_i F_i x_i^{(b)}.$$

Năng lượng cung cấp trong bữa ăn b được xác định bởi:

$$E_b(x) = \sum_i E_i x_i^{(b)}.$$

Mục tiêu của bài toán là xác định các khẩu phần $x_i^{(b)}$ sao cho:

- Tổng dinh dưỡng trong ngày đạt gần nhất với nhu cầu người dùng.
- Đồng thời phân bổ năng lượng hợp lý giữa các bữa ăn.

Từ đó, hàm mục tiêu được xây dựng như sau:

$$\begin{aligned} f(x) = & (E_{day}(x) - E^{req})^2 + \alpha(P_{day}(x) - P^{req})^2 + \beta(C_{day}(x) - C^{req})^2 \\ & + \gamma(F_{day}(x) - F^{req})^2 + \lambda \sum_{b=1}^B (E_b(x) - E_b^{target})^2 \end{aligned}$$

Trong đó:

- $E^{req}, P^{req}, C^{req}, F^{req}$ là nhu cầu dinh dưỡng mục tiêu trong một ngày.
- E_b^{target} là mức năng lượng mục tiêu của bữa ăn b , được xác định dựa trên tỷ lệ phân bổ năng lượng trong ngày.
- $\alpha, \beta, \gamma, \lambda$ là các trọng số điều chỉnh mức độ ưu tiên trong tối ưu dinh dưỡng tổng thể và cân đối năng lượng theo từng bữa ăn.
- Ràng buộc về khẩu phần ăn của từng loại món: $x_i^{min} \leq x_i \leq x_i^{max}$.

Như vậy, hoàn toàn có thể áp dụng thuật toán SLSQP để giải quyết bài toán này để tìm nghiệm tối ưu.

1.4. Các công nghệ sử dụng

1.4.1. Flutter

Flutter [6] là một SDK và framework do Google phát triển, cho phép xây dựng ứng dụng di động (Android & iOS) – và ngày càng mở rộng sang desktop, web và nhúng – từ một cơ sở mã nguồn duy nhất.



Ảnh 1.5. Flutter – Framework phát triển ứng dụng đa nền tảng của Google

Trong bối cảnh kỹ thuật lập trình đa nền tảng phát triển mạnh mẽ, Google đã phát triển Flutter và liên tục hoàn thiện nó theo chiều hướng đa nền tảng. Quá trình này diễn ra từng bước với những bước tiến đáng kể:

- Năm 2015, Google công bố Flutter được viết dựa trên ngôn ngữ Dart với thiết kế dễ học, dễ sử dụng để phát triển các ứng dụng. Đây là lần đầu tiên Flutter được giới thiệu với cộng đồng.
- Năm 2017 đánh dấu bước ngoặt quan trọng khi phiên bản alpha đầu tiên của Flutter (0.0.6) chính thức ra mắt, và đã trở thành lựa chọn phổ biến của các nhà phát triển nhờ hiệu suất cao, dễ sử dụng và thư viện tiện ích đựng sẵn phong phú.
- Năm 2018 chứng kiến sự phát triển mạnh mẽ của Flutter khi phiên bản 1.0 chính thức ra mắt tại sự kiện Flutter Live. Đây được coi là phiên bản ổn định đầu tiên, mở ra giai đoạn sử dụng rộng rãi hơn.
- Năm 2019, tính năng hỗ trợ Flutter cho các nền tảng desktop và web chính thức được công bố. Các công cụ phát triển cho Windows, macOS, Linux và web cũng được ra mắt.
- Từ năm 2021 đến nay, Flutter liên tục phát hành các phiên bản mới như 2.0.6, 3.0,... nhằm cải thiện trải nghiệm người dùng và tính năng.

Đặc điểm của ngôn ngữ lập trình Flutter:

- Hot Reload – JIT (Just-in-time): Cho phép cập nhật mã và xem ngay kết quả trên thiết bị/emulator mà không phải khởi động lại ứng dụng, tăng tốc độ phát triển, debug và thử nghiệm giao diện.
- AOT – Ahead-of-time Compiler: Khi build cho môi trường production, Dart biên dịch AOT thành mã máy gốc, giúp ứng dụng khởi động nhanh và chạy

mượt mà, kết hợp JIT (cho phát triển) và AOT (cho production) để cân bằng giữa trải nghiệm developer và hiệu năng runtime.

- Expressive and Flexible UI: Giao diện hoàn toàn được dựng từ widget, dễ mở rộng, customize, hỗ trợ animation/transition phong phú, API trực quan, hai thư viện thiết kế sẵn: Material & Cupertino, cho phép app trông “native” trên mọi nền tảng.

1.4.2. Dart và Dio

Dart [6] 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.



Ảnh 1.6. Dart – Ngôn ngữ lập trình chính của Flutter

Ngôn ngữ này được giới thiệu lần đầu vào ngày 10/10/2011 tại hội nghị GOTO (Đan Mạch) bởi Lars Bak và Kasper Lund. Dù ban đầu vẫn phải nhiều ý kiến trái chiều, Dart vẫn không ngừng phát triển, đạt cột mốc với phiên bản 1.0 vào năm 2013. Đặc biệt, phiên bản Dart 2.0 ra mắt tháng 8/2018 đã chuyển trọng tâm sang biên dịch JavaScript, cải tiến cú pháp và hệ thống kiểu.

Dart là một ngôn ngữ mới, được sự đầu tư phát triển bởi đội ngũ từ Google nên có nhiều tính năng của một ngôn ngữ lập trình hiện đại:

- Cú pháp đơn giản và dễ học: Dart có cú pháp tương tự Java/C++, giúp lập trình viên dễ tiếp cận và rút ngắn thời gian học.
- Hiệu suất cao: Nhờ biên dịch Ahead-Of-Time (AOT), Dart cho hiệu năng nhanh và mượt trên nhiều nền tảng, đặc biệt là di động.
- Hỗ trợ lập trình hướng đối tượng (OOP): Dart sử dụng class, kế thừa, và đơ hình giúp tổ chức và tái sử dụng mã hiệu quả.
- Null Safety: Ngăn lỗi null tại thời điểm chạy bằng cách kiểm soát chặt chẽ biến null trong quá trình biên dịch.
- Biên dịch linh hoạt (AOT + JIT): Dart hỗ trợ cả AOT (tăng hiệu năng khi chạy) và JIT (hỗ trợ debug nhanh), giúp tăng tốc quá trình phát triển và triển khai.

- Phát triển đa nền tảng: Kết hợp với Flutter, Dart cho phép viết một lần, chạy trên iOS, Android, Web, và Desktop.
- Hot Reload: Tính năng hot reload cho phép cập nhật giao diện ngay lập tức khi thay đổi mã nguồn – rất tiện lợi khi phát triển UI.
- Tích hợp công cụ mạnh mẽ: Dart hoạt động tốt với các IDE như VS Code, Android Studio, hỗ trợ gợi ý mã, debug, kiểm tra lỗi, tăng hiệu suất làm việc.



Ảnh 1.7. Luồng giao tiếp Web/Mobile App với Server thông qua Dio

Dio [6] là một thư viện HTTP client mạnh mẽ được sử dụng phổ biến trong Flutter để thực hiện các thao tác giao tiếp giữa ứng dụng và server thông qua các giao thức HTTP.

Thư viện này hỗ trợ:

- Gửi và nhận dữ liệu qua API
- Quản lý các headers, query parameters
- Thiết lập timeout và hủy yêu cầu
- Quản lý lỗi hiệu quả
- Sử dụng interceptors để xử lý trước và sau khi gửi request

Dio được đánh giá cao nhờ sự linh hoạt và khả năng mở rộng, phù hợp cho các dự án từ đơn giản đến phức tạp.

Ưu điểm:

- Đầy đủ chức năng HTTP: Hỗ trợ tất cả các phương thức HTTP: GET, POST, PUT, DELETE, PATCH...
- Hỗ trợ Interceptors: Cho phép xử lý request/response trước và sau khi gửi, rất hữu ích cho việc thêm token, log, hoặc xử lý lỗi chung.

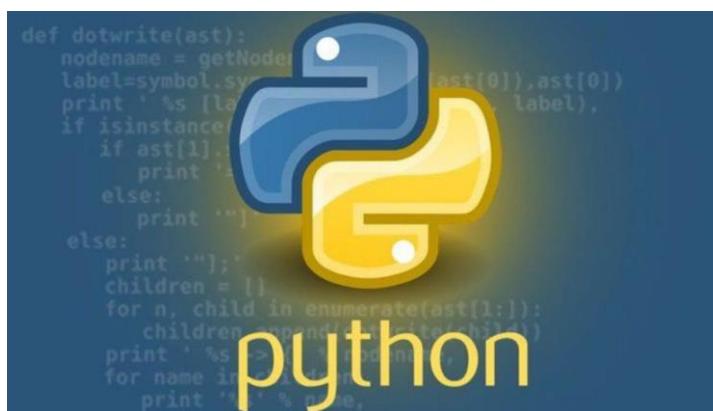
- Hỗ trợ Timeout, Retry, Cancel: Có thể thiết lập timeout, hủy request giữa chừng, hoặc retry lại khi thất bại.
- Dễ mở rộng và tùy biến cao: Có thể cấu hình base URL, headers mặc định, responseType, v.v.
- Xử lý lỗi rõ ràng: Dio phân biệt rõ các loại lỗi như kết nối, timeout, server trả lỗi...

Nhược điểm:

- Hơi nặng với ứng dụng nhỏ: Đối với những app chỉ cần vài API đơn giản, Dio có thể bị đánh giá là "thừa tính năng".
- Cần thêm code để xử lý model: Không tự động ánh xạ JSON ↔ Object
- Không mặc định hỗ trợ cache: Muốn cache response phải tự làm hoặc dùng thư viện phụ trợ khác.

1.4.3. Python

Python [7] là một ngôn ngữ lập trình bậc cao, đa năng và dễ học, được phát triển bởi Guido van Rossum vào năm 1991. Python nổi bật với cú pháp rõ ràng, ngắn gọn và gần gũi với ngôn ngữ tự nhiên, giúp người mới học dễ tiếp cận và lập trình viên chuyên nghiệp phát triển nhanh chóng các ứng dụng phức tạp.



Ảnh 1.8. Ngôn ngữ lập trình Python

Đặc điểm nổi bật của Python:

- Dễ đọc, dễ học: Cú pháp của Python được thiết kế để người đọc có thể hiểu dễ dàng, gần giống ngôn ngữ tự nhiên.
- Đa nền tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux,...
- Thư viện phong phú: Python sở hữu hệ sinh thái thư viện rất lớn, phục vụ nhiều lĩnh vực như khoa học dữ liệu (NumPy, Pandas), trí tuệ nhân tạo (TensorFlow, PyTorch), lập trình web (Django, Flask), tự động hóa,...
- Hỗ trợ lập trình hướng đối tượng, hàm và thủ tục: Giúp người dùng linh hoạt trong việc xây dựng chương trình.

Python được sử dụng rộng rãi trong nhiều lĩnh vực:

- Khoa học dữ liệu và trí tuệ nhân tạo: Dùng để phân tích, huấn luyện mô hình và xử lý dữ liệu lớn.
- Phát triển web: Với các framework như Django, Flask.
- Tự động hóa và DevOps: Viết script để tự động hóa các tác vụ hệ thống.
- Giảng dạy và nghiên cứu: Python là một trong những ngôn ngữ phổ biến nhất trong giáo dục lập trình hiện nay.

1.4.4. FastAPI

FastAPI [7] là một framework web hiện đại dùng để xây dựng API với Python 3.7+, dựa trên các tiêu chuẩn OpenAPI (Swagger) và JSON Schema. Framework này được tạo ra bởi Sebastián Ramírez và ra mắt lần đầu vào năm 2018. FastAPI được thiết kế nhằm mang lại hiệu năng cao, tốc độ phát triển nhanh, đồng thời đảm bảo dễ bảo trì và an toàn cho ứng dụng.



Ảnh 1.9. FastAPI framework cho backend Python

Đặc điểm nổi bật của FastAPI:

- **Hiệu năng cao:** FastAPI có tốc độ xử lý tương đương với các framework như Node.js và Go, nhờ sử dụng Starlette cho phần web và Pydantic cho phần xử lý dữ liệu.
- **Tự động sinh tài liệu API:** Khi xây dựng API, FastAPI tự động tạo giao diện tài liệu tương tác thông qua Swagger UI hoặc ReDoc, giúp nhà phát triển dễ dàng kiểm thử và chia sẻ API.
- **Hỗ trợ kiểu dữ liệu tĩnh (type hints):** Giúp kiểm tra lỗi, sinh tài liệu tự động và tăng độ tin cậy của mã nguồn.
- **Dễ sử dụng và phát triển nhanh:** Cú pháp ngắn gọn, trực quan và tận dụng tối đa cú pháp Python hiện đại như async / await để hỗ trợ lập trình bất đồng bộ (asynchronous).

FastAPI là lựa chọn lý tưởng cho các dự án cần hiệu năng cao, cấu trúc rõ ràng, và phát triển nhanh chóng. Với cú pháp thân thiện và khả năng tự động hóa mạnh mẽ, FastAPI ngày càng trở thành framework phổ biến hàng đầu trong hệ sinh thái Python.

1.4.5. PostgreSQL

PostgreSQL [8] là một hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (object-relational database management system) miễn phí và mã nguồn mở (RDBMS) tiên tiến, có khả năng mở rộng cao và tuân thủ các tiêu chuẩn kỹ thuật. Hệ thống được thiết kế để xử lý nhiều loại khối lượng công việc khác nhau, từ máy tính cá nhân đến các kho dữ liệu hoặc dịch vụ web có nhiều người dùng đồng thời. PostgreSQL được phát triển bởi PostgreSQL Global Development Group và được phát hành lần đầu vào ngày 08/07/1996. Hệ thống có tính linh hoạt cao, có thể chạy trên nhiều nền tảng như Mac OS X, Solaris và Windows. Là phần mềm mã nguồn mở và miễn phí, PostgreSQL cho phép người dùng tự do sử dụng, chỉnh sửa và phân phối cho mọi mục đích. Ngoài ra, PostgreSQL nổi bật với tính ổn định cao và là hệ thống quản lý cơ sở dữ liệu đầu tiên triển khai cơ chế kiểm soát đồng thời nhiều phiên bản (MVCC).



Ảnh 1.10. Hệ quản trị cơ sở dữ liệu PostgreSQL

PostgreSQL cung cấp nhiều tính năng hiện đại cho phép xử lý khối lượng dữ liệu lớn và thực hiện nhiều tác vụ phức tạp:

- Tuân thủ chuẩn SQL.
- Hỗ trợ các ràng buộc toàn vẹn (integrity constraints).
- Thực hiện toàn vẹn giao dịch (ACID).
- Kiểm soát truy cập đồng thời đa phiên bản (MVCC).
- Sao lưu và phục hồi dữ liệu (streaming replication).
- Hỗ trợ lập trình thủ tục, trigger, view, khóa ngoại, và các kiểu dữ liệu phong phú như số, chuỗi, boolean, mảng, JSON, XML, UUID,...
- Các loại chỉ mục: B-tree, Hash, GiST, GIN, BRIN, Bloom Filters.
- Hỗ trợ giao dịch song song, phân vùng bảng, và đồng bộ bản ghi.
- Hỗ trợ xác thực SSL, LDAP, SCRAM-SHA-256, Certificate, cùng nhiều cơ chế bảo mật mạnh mẽ khác.

- Hệ thống có khả năng mở rộng cao với các ngôn ngữ lập trình mở rộng như PL/pgSQL, Perl, Python, và C.

1.4.6. Firebase và FCM

Firebase [6] là một trong những BaaS (Backend as a Service), tức là một dịch vụ cung cấp các giải pháp backend cho các ứng dụng web và di động.



Ảnh 1.11. Firebase – Hỗ trợ phát triển ứng dụng nhanh chóng

Người dùng không cần phải tự thiết kế, triển khai và quản lý server hay cơ sở dữ liệu của mình, mà chỉ cần sử dụng các API và SDK do Firebase cung cấp để kết nối với các dịch vụ của nó. Bằng cách này, người dùng có thể tiết kiệm thời gian, chi phí và công sức cho việc xây dựng backend.

Firebase được ra đời vào năm 2011 bởi James Tamplin và Andrew Lee với tên gọi ban đầu là Evolve, một nền tảng cung cấp các API để tích hợp tính năng chat vào các trang web.

Sau đó, họ nhận ra rằng nền tảng này được sử dụng để truyền dữ liệu ứng dụng chứ không chỉ là chat. Họ đã phát triển Evolve thành Firebase và công bố nó vào tháng 4 năm 2012. Đến tháng 10 năm 2014, Firebase đã được Google mua lại và trở thành một phần của Google Cloud Platform.

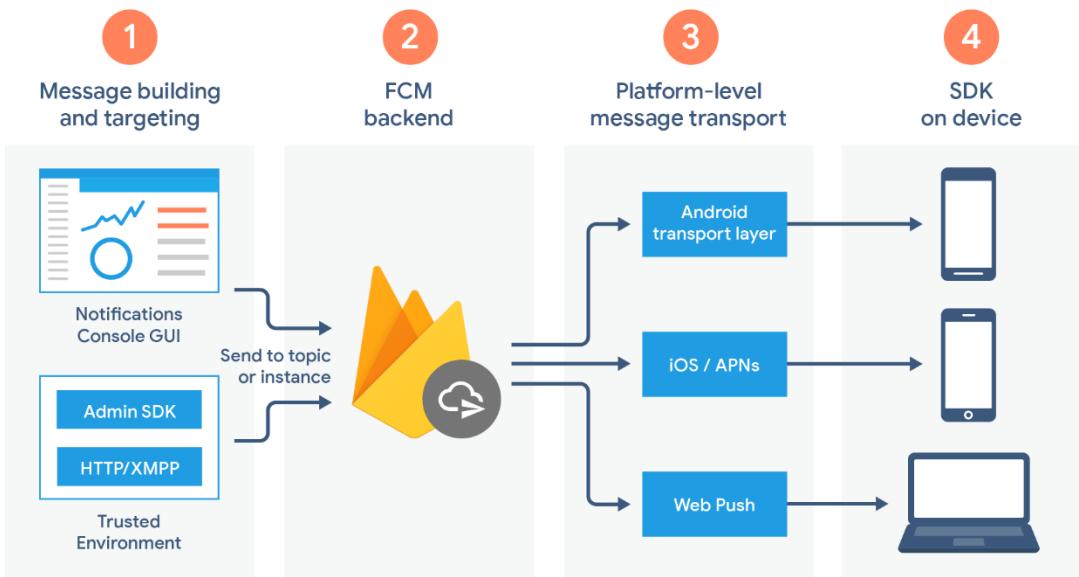
Ưu điểm của Firebase:

- Giúp tiết kiệm thời gian và chi phí phát triển ứng dụng, bằng cách cung cấp các dịch vụ và công cụ sẵn có và dễ sử dụng.
- Cho phép các nhà phát triển tập trung vào việc xây dựng giao diện và tính năng của ứng dụng, mà không cần lo lắng về việc xử lý và lưu trữ dữ liệu ở phía máy chủ.
- Có khả năng mở rộng tự động theo quy mô của ứng dụng, không cần phải quản lý cơ sở hạ tầng hay máy chủ.
- Được hưởng lợi từ hệ thống máy chủ mạnh mẽ và an toàn của Google, đảm bảo hiệu suất cao và độ tin cậy cao cho các ứng dụng.
- Hỗ trợ nhiều ngôn ngữ lập trình và nền tảng khác nhau, cho phép phát triển các ứng dụng đa nền tảng một cách hiệu quả và dễ dàng.

Hạn chế của Firebase:

- Có giới hạn về dung lượng lưu trữ và số lượng kết nối trong phiên bản miễn phí. Để sử dụng các tính năng nâng cao hoặc quy mô lớn hơn, các nhà phát triển phải trả phí theo mức sử dụng.
- Không đáp ứng được một số yêu cầu đặc biệt hoặc phức tạp của các ứng dụng. Ví dụ, Firebase Realtime Database chỉ hỗ trợ cơ sở dữ liệu NoSQL, không hỗ trợ các truy vấn phức tạp hay các thao tác với dữ liệu liên quan.
- Gặp khó khăn trong việc tích hợp với một số công nghệ hoặc dịch vụ khác. Ví dụ, Firebase không hỗ trợ GraphQL hay Apollo Client, hai công nghệ phổ biến trong việc xây dựng API hiện đại.

FCM là một dịch vụ gửi thông báo, tin nhắn đa nền tảng được cung cấp bởi Google, cho phép gửi tin nhắn, thông báo một cách đáng tin cậy và hoàn toàn miễn phí tới các thiết bị đã được đăng ký với FCM.

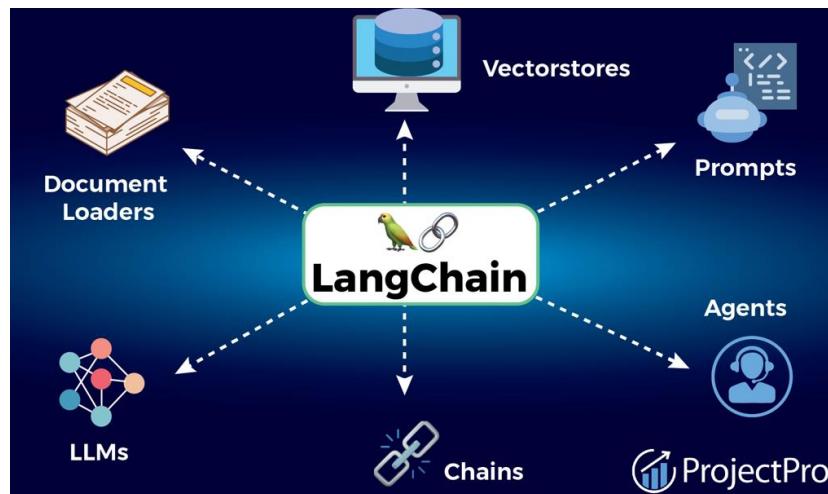


Ảnh 1.12. Cơ chế hoạt động của FCM

Nguyên tắc hoạt động: Các thiết bị client sẽ đăng ký `device_token` lên cho FCM. Các thông báo, tin nhắn được soạn và gửi từ một website, từ Notifications composer của firebase cung cấp, FCM sẽ nhận những thông báo này và xử lý gửi về các thiết bị đã đăng ký với FCM từ trước. Khi các thiết bị có kết nối mạng thì thông báo sẽ được gửi về ứng dụng thành công.

1.4.7. LangChain

Langchain [12] là một framework hỗ trợ xây dựng các ứng dụng dựa trên LLM như ChatGPT. Framework này cung cấp các thành phần giúp kết nối LLM với dữ liệu bên ngoài, các công cụ xử lý logic và bộ nhớ hội thoại, từ đó cho phép xây dựng các hệ thống có khả năng suy luận và tương tác linh hoạt hơn so với việc gọi LLM trực tiếp.



Ảnh 1.13. Framework LangChain

LangChain tổ chức ứng dụng dưới dạng các chuỗi xử lý (chains), trong đó đầu ra của một bước có thể được sử dụng làm đầu vào cho bước tiếp theo. Ngoài ra, framework còn hỗ trợ quản lý ngữ cảnh hội thoại (memory), tích hợp nguồn tri thức bên ngoài (retriever) và điều phối lời gọi mô hình theo các kịch bản xác định trước.

1.4.8. LangGraph

LangGraph [13] là một framework mở rộng của LangChain, được thiết kế để xây dựng các ứng dụng LLM theo mô hình đồ thị trạng thái (graph-based workflow). Thay vì tổ chức luồng xử lý theo dạng tuyến tính, LangGraph cho phép mô hình hóa hệ thống dưới dạng các nút xử lý và các cạnh chuyển tiếp, hỗ trợ nhánh rẽ, vòng lặp và điều kiện chuyển bước.



Ảnh 1.14. Framework LangGraph

LangGraph đặc biệt phù hợp với các hệ thống hội thoại nhiều bước, trong đó trạng thái của phiên làm việc cần được duy trì và cập nhật liên tục. Framework này giúp kiểm soát rõ ràng luồng xử lý và giảm sự phức tạp khi hệ thống mở rộng.

1.4.9. DeepSeek LLM

DeepSeek [14] là một LLM hoạt động dựa trên sự kết hợp của học máy, xử lý ngôn ngữ tự nhiên và mạng nơ-ron sâu. Mô hình được huấn luyện trên tập dữ liệu lớn và đa dạng, cho phép học được các quy luật và mối quan hệ ngữ nghĩa trong dữ liệu, từ đó hiểu và phản hồi các truy vấn ngôn ngữ của người dùng một cách tự nhiên.



Ảnh 1.15. Deepseek

Về mặt kiến trúc, DeepSeek sử dụng các mô hình hiện đại dựa trên Transformer, kết hợp với một số kỹ thuật tối ưu như Mixture-of-Experts (MoE), trong đó chỉ các “chuyên gia” phù hợp với truy vấn được kích hoạt, giúp giảm chi phí tính toán mà vẫn đảm bảo hiệu suất cao. Ngoài ra, mô hình áp dụng cơ chế Multi-Head Latent Attention, giúp tăng khả năng nắm bắt ngữ cảnh và mối quan hệ giữa các thành phần trong văn bản.

DeepSeek còn được cải thiện thông qua các kỹ thuật học tăng cường, cho phép mô hình tự điều chỉnh và nâng cao khả năng suy luận theo thời gian. Một ưu điểm quan trọng khác là khả năng xử lý ngữ cảnh dài và sinh đầu ra lớn, phù hợp với các ứng dụng yêu cầu phân tích thông tin phức tạp và hội thoại nhiều lượt.

Nhờ những đặc điểm trên, DeepSeek phù hợp để tích hợp vào các hệ thống chatbot thông minh, trong đó mô hình đảm nhiệm vai trò hiểu ngôn ngữ, suy luận và sinh phản hồi, trong khi các phép toán và xử lý chuyên biệt được thực hiện bởi các module nghiệp vụ khác.

1.4.10. ElasticSearch

Elasticsearch [15] là một hệ thống tìm kiếm và phân tích dữ liệu mã nguồn mở, được xây dựng trên nền tảng Apache Lucene. Elasticsearch được thiết kế để xử lý các tác vụ tìm kiếm với tốc độ cao trên khối lượng dữ liệu lớn, thông qua cơ chế lập chỉ mục (indexing) và truy vấn tối ưu. Hệ thống hoạt động theo kiến trúc phân tán, cho phép mở rộng linh hoạt bằng cách chia dữ liệu thành nhiều shard và sao lưu trên các node khác nhau.



elasticsearch

Ảnh 1.16. Elasticsearch

Dữ liệu trong Elasticsearch được lưu trữ dưới dạng document JSON và được tổ chức theo các index, giúp việc tìm kiếm và truy vấn trở nên nhanh chóng và hiệu quả. Elasticsearch hỗ trợ nhiều dạng truy vấn khác nhau như tìm kiếm toàn văn (full-text search), tìm kiếm theo điều kiện, sắp xếp và phân tích dữ liệu theo thời gian thực. Nhờ đó, Elasticsearch thường được sử dụng trong các hệ thống cần khả năng tìm kiếm mạnh mẽ như hệ thống log, phân tích dữ liệu, và các nền tảng tìm kiếm thông tin.

Bên cạnh các chức năng tìm kiếm truyền thống, Elasticsearch còn hỗ trợ lưu trữ và truy vấn dữ liệu vector, cho phép thực hiện tìm kiếm dựa trên độ tương đồng ngữ nghĩa. Tính năng này giúp Elasticsearch không chỉ dừng lại ở việc tìm kiếm theo từ khóa, mà còn có thể áp dụng cho các bài toán tìm kiếm ngữ nghĩa và truy xuất thông tin nâng cao.

1.5. Tổng kết chương

Như vậy, chương 1 đã trình bày các cơ sở lý thuyết làm nền tảng cho việc xây dựng ứng dụng quản lý dinh dưỡng tích hợp chatbot AI. Nội dung chương bao gồm các kiến thức cơ bản về dinh dưỡng, các chỉ số đánh giá thể trạng và nhu cầu năng lượng, cùng với những nguyên lý và công nghệ AI như LLM, RAG, Self-Query Retriever và thuật toán tối ưu SLSQP. Những kiến thức này là cơ sở khoa học để phân tích yêu cầu, thiết kế hệ thống và triển khai các chức năng gợi ý thực đơn, tối ưu khẩu phần ăn và chatbot tư vấn dinh dưỡng ở các chương tiếp theo.

Chương 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

2.1. Tổng quan về nghiệp vụ

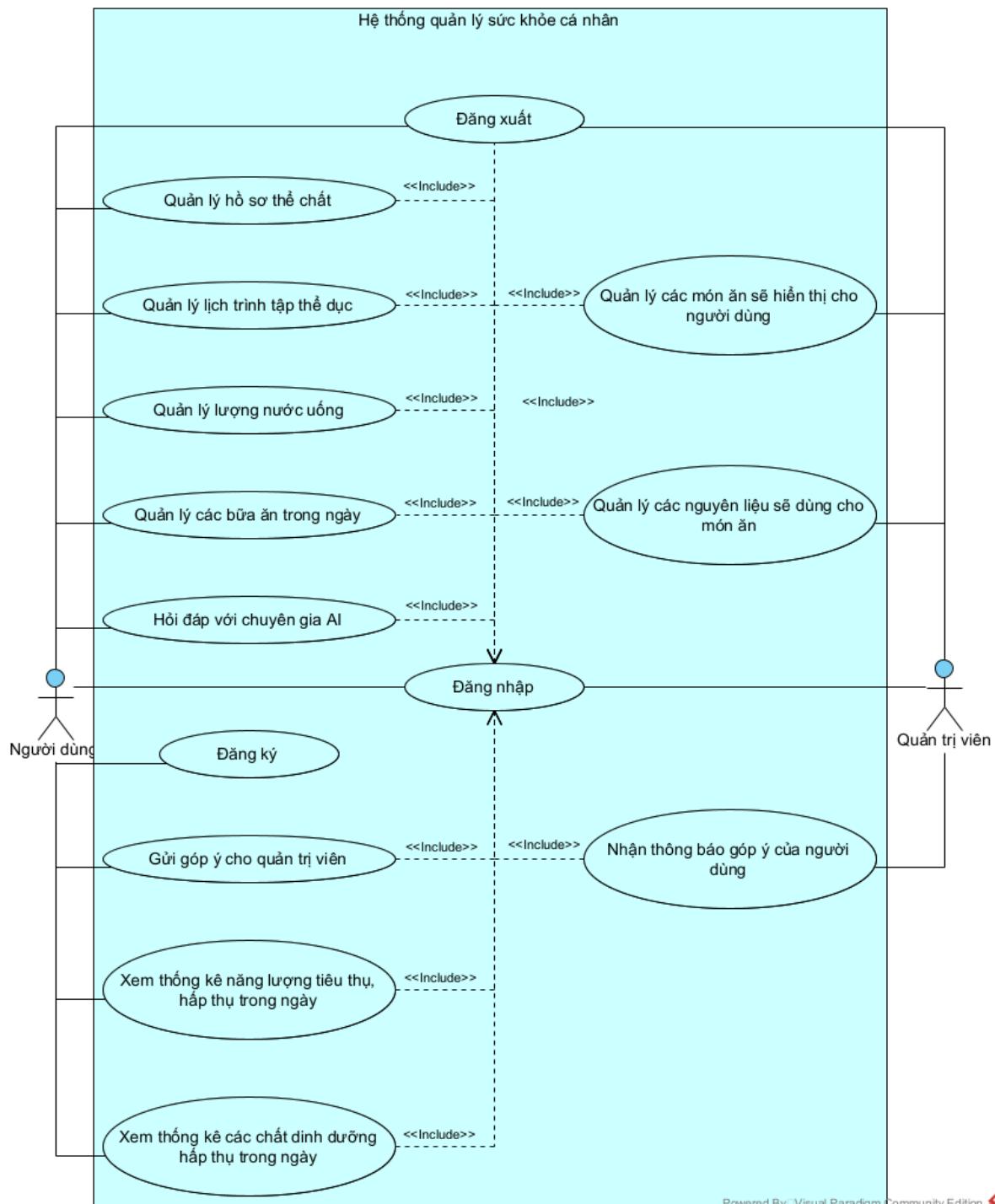
Hệ thống có 2 tác nhân chính là người dùng và quản trị viên, mỗi tác nhân đều được chia các chức năng riêng biệt trong hệ thống.

Tác nhân thứ 1 (người dùng): Là người truy cập hệ thống thông qua chức năng đăng ký, đăng nhập. Người dùng có thể sử dụng các chức năng liên quan đến việc quản lý sức khỏe cá nhân của người dùng. Người dùng có các chức năng chính như quản lý hồ sơ thể chất (xem, thêm, cập nhật), quản lý lịch trình tập thể dục (xem, thêm, xóa), quản lý lượng nước uống (xem, thêm, xóa), quản lý các món ăn từng bữa trong ngày (xem, thêm, xóa, tìm kiếm), xem thống kê năng lượng trao đổi trong ngày, các chất dinh dưỡng hấp thụ trong ngày, hỏi đáp với chuyên gia AI, gửi góp ý cho quản trị viên.

Tác nhân thứ 2 (quản trị viên): Là người quản lý dữ liệu hệ thống, lắng nghe góp ý của người dùng. Quản trị viên có các chức năng chính như quản lý các món ăn sẽ hiển thị cho người dùng (xem, thêm, sửa, xóa, tìm kiếm), quản lý các nguyên liệu sẽ dùng cho món ăn (xem, thêm, sửa, xóa, tìm kiếm), nhận thông báo góp ý của người dùng.

2.2. Phân tích yêu cầu chức năng

2.2.1. Tổng quan hệ thống



Ảnh 2.1. Biểu đồ usecase tổng quan hệ thống

2.2.2. Đặc tả Usecase

Usecase	Đăng nhập
Actor	Người dùng, quản trị viên
Pre-condition	Người dùng đã đăng ký tài khoản, quản trị viên đã được cấp tài khoản
Post condition	Đăng nhập thành công
Main events	<ol style="list-style-type: none"> 1. Người dùng, quản trị viên chọn đăng nhập. 2. Hệ thống hiển thị giao diện đăng nhập. 3. Người dùng, quản trị viên nhập tài khoản, mật khẩu rồi ấn nút Đăng nhập. 4. Đăng nhập thành công, hệ thống chuyển đến giao diện Trang chủ.
Exception	4a. Hệ thống thông báo lỗi: “Tài khoản/mật khẩu không đúng” nếu thông tin được nhập không chính xác.

Bảng 2.1. Kịch bản usecase Đăng nhập

Usecase	Đăng ký tài khoản mới
Actor	Người dùng
Pre-condition	Người dùng chưa có tài khoản
Post condition	Đăng ký thành công
Main events	<ol style="list-style-type: none"> 1. Người dùng chọn Đăng ký. 2. Hệ thống hiển thị giao diện đăng ký. 3. Người dùng nhập tài khoản, mật khẩu, xác nhận mật khẩu rồi ấn nút Đăng ký. 4. Đăng ký thành công, hệ thống chuyển đến giao diện Đăng nhập.
Exception	4a. Hệ thống thông báo lỗi: “Tài khoản đã tồn tại” nếu tài khoản được nhập đã tồn tại.

Bảng 2.2. Kịch bản usecase Đăng ký tài khoản mới

Usecase	Quản lý lịch trình tập thể dục
Actor	Người dùng
Pre-condition	Người dùng đã có tài khoản
Post condition	Xem, thêm, xóa lịch trình tập thể dục
Main events	<ul style="list-style-type: none"> 1. Sau khi đăng nhập, người dùng chọn Xem lịch trình tập thể dục 2. Hệ thống hiển thị giao diện Danh sách lịch trình tập thể dục. 3. Người dùng chọn Thêm lịch trình tập thể dục. 4. Hệ thống hiển thị giao diện Chọn bài tập thể dục. 5. Người dùng chọn 1 bài tập thể dục. 6. Hệ thống hiển thị giao diện Đèn mức độ, thời gian bài tập. 7. Người dùng chọn mức độ và đèn thời gian bài tập. 8. Thêm lịch trình tập thể dục thành công, hệ thống trở lại giao diện Danh sách lịch trình tập thể dục. 9. Người dùng chọn Xóa lịch trình tập thể dục của 1 bài tập trong danh sách. 10. Hệ thống hiển thị giao diện Xác nhận xóa bài tập. 11. Người dùng chọn Xác nhận. 12. Xóa lịch trình tập thể dục thành công, hệ thống trở lại giao diện Danh sách lịch trình tập thể dục.
Exception	Không

Bảng 2.3. Kịch bản usecase Quản lý lịch trình tập thể dục

Usecase	Quản lý lượng nước uống
Actor	Người dùng
Pre-condition	Người dùng đã có tài khoản
Post condition	Xem, thêm, xóa lượng nước uống
Main events	<ol style="list-style-type: none"> 1. Sau khi đăng nhập, người dùng chọn Xem lượng nước uống. 2. Hệ thống hiển thị giao diện Danh sách lượng nước uống. 3. Người dùng chọn Thêm lượng nước uống. 4. Hệ thống hiển thị giao diện Đơn vị lượng, đơn vị lượng nước uống. 5. Người dùng chọn đơn vị và điền số lượng của lượng nước uống. 6. Thêm lượng nước uống thành công, hệ thống trả lại giao diện Danh sách lượng nước uống. 7. Người dùng chọn Xóa lượng nước uống của 1 lần uống nước trong danh sách. 8. Hệ thống hiển thị giao diện Xác nhận xóa lượng nước. 9. Người dùng chọn Xác nhận. 10. Xóa lượng nước uống thành công, hệ thống trả lại giao diện Danh sách lượng nước uống.
Exception	Không

Bảng 2.4. Kịch bản usecase Quản lý lượng nước uống

Use case	Quản lý các món ăn từng bữa trong ngày
Actor	Người dùng
Pre-condition	Người dùng đã có tài khoản
Post condition	Xem, thêm, xóa, tìm kiếm các món ăn từng bữa trong ngày
Main events	<ol style="list-style-type: none"> 1. Sau khi đăng nhập, người dùng chọn Xem các món ăn của 1 trong 4 bữa: Bữa sáng, bữa trưa, bữa tối, bữa ăn nhẹ. 2. Hệ thống hiển thị giao diện Danh sách các món ăn của bữa đó. 3. Người dùng chọn Thêm món ăn. 4. Hệ thống hiển thị giao diện Chọn món ăn. 5. Người dùng nhập từ khóa vào ô tìm kiếm và chọn Tìm kiếm. 6. Hệ thống cập nhật danh sách món ăn trong giao diện Chọn món ăn. 7. Người dùng chọn 1 món ăn. 8. Hệ thống hiển thị giao diện Đèn khối lượng món ăn. 9. Người dùng điều chỉnh khối lượng món ăn. 10. Thêm món ăn từng bữa thành công, hệ thống trả lại giao diện Danh sách các món ăn của bữa đó. 11. Người dùng chọn Xóa món ăn của 1 món trong danh sách. 12. Hệ thống hiển thị giao diện Xác nhận xóa món ăn. 13. Người dùng chọn Xác nhận. 14. Xóa món ăn từng bữa thành công, hệ thống trả lại giao diện Danh sách các món ăn của bữa đó.
Exception	Không

Bảng 2.5. Kịch bản usecase Quản lý các món ăn từng bữa trong ngày

Usecase	Quản lý hồ sơ thẻ chất
Actor	Người dùng
Pre-condition	Người dùng đã có tài khoản
Post condition	Xem, thêm, cập nhật hồ sơ thẻ chất thành công
Main events	<ul style="list-style-type: none"> 1. Người dùng chưa thêm hồ sơ thẻ chất sau khi đăng nhập hoặc đã thêm hồ sơ chọn Sửa hồ sơ thẻ chất. 2. Hệ thống hiển thị chuỗi giao diện điền thông tin của Thêm hoặc Sửa hồ sơ thẻ chất. 3. Người dùng nhập thông tin trong các giao diện rồi ấn nút Hoàn thành. 4. Thêm hoặc Sửa hồ sơ thành công, hệ thống chuyển đến giao diện Trang chủ.
Exception	Không

Bảng 2.6. Kịch bản usecase Quản lý hồ sơ thẻ chất

Usecase	Thông kê dinh dưỡng trong ngày
Actor	Người dùng
Pre-condition	Người dùng đã có tài khoản
Post condition	Người dùng xem được các thông số thống kê
Main events	<ul style="list-style-type: none"> 1. Sau khi đăng nhập, người dùng chọn Nhật kí hàng ngày. 2. Hệ thống hiển thị giao diện Nhật kí hiện ra với các thông số: Năng lượng mục tiêu, Năng lượng tiêu thụ, Năng lượng hấp thụ. 3. Người dùng chọn Theo dõi hàng ngày. 4. Hệ thống hiển thị giao diện Theo dõi hiện ra với các thông số của các chất dinh dưỡng: Lượng dinh dưỡng đã hấp thụ, lượng dinh dưỡng mục tiêu.
Exception	Không

Bảng 2.7. Kịch bản usecase Thông kê dinh dưỡng trong ngày

Usecase	Hỏi đáp với chuyên gia AI
Actor	Người dùng
Pre-condition	Người dùng đã có tài khoản
Post condition	Người dùng nhận được câu trả lời của chuyên gia AI
Main events	<ol style="list-style-type: none"> 1. Sau khi đăng nhập, người dùng chọn Khuyến nghị. 2. Hệ thống hiển thị giao diện Khuyến nghị hiện ra với các món ăn mà chatbot đề xuất cho người dùng 1 ngày. 3. Người dùng chọn Tạo mới. 4. Hệ thống cập nhật giao diện Khuyến nghị với các món ăn đề xuất cho người dùng 1 ngày mới. 5. Người dùng chọn Đổi món. 6. Hệ thống cập nhật giao diện Khuyến nghị với một món mới tương đồng với món muốn đổi. 7. Người dùng chọn Chuyên gia. 8. Hệ thống hiển thị giao diện Chuyên gia. 9. Người dùng điền câu hỏi và ấn Gửi. 10. Giao diện cập nhật giao diện Chuyên gia với câu trả lời của chatbot.
Exception	Không

Bảng 2.8. Kịch bản usecase Hỏi đáp với chuyên gia AI

Usecase	Gửi góp ý cho quản trị viên
Actor	Người dùng
Pre-condition	Người dùng đã có tài khoản
Post condition	Người dùng gửi được gửi góp ý cho quản trị viên
Main events	<ol style="list-style-type: none"> 1. Sau khi đăng nhập, người dùng chọn Hòm thư góp ý. 2. Hệ thống hiển thị giao diện Hòm thư góp ý . 3. Người dùng điền góp ý và ấn Gửi. 4. Gửi góp ý thành công, hệ thống trả lại giao diện Trang chủ.
Exception	Không

Bảng 2.9. Kịch bản usecase Gửi góp ý cho quản trị viên

Usecase	Nhận thông báo góp ý của người dùng
Actor	Quản trị viên
Pre-condition	Quản trị viên được cấp tài khoản
Post condition	Nhận thông báo góp ý của người dùng thành công.
Main events	<ol style="list-style-type: none"> 1. Sau khi đăng nhập, quản trị viên chọn Thông báo 2. Hệ thống hiển thị giao diện Danh sách các thông báo. 3. Quản trị viên chọn 1 thông báo. 4. Hệ thống hiển thị giao diện Chi tiết thông báo.
Exception	Không

Bảng 2.10. Kịch bản usecase Nhận thông báo góp ý của người dùng

Usecase	Quản lý các món ăn hiển thị cho người dùng
Actor	Quản trị viên
Pre-condition	Quản trị viên được cấp tài khoản
Post condition	Xem, thêm, sửa, xóa, tìm kiếm thông tin các món ăn
Main events	<ul style="list-style-type: none"> 1. Sau khi đăng nhập, quản trị viên chọn Danh sách món ăn. 2. Hệ thống hiển thị giao diện Danh sách các món ăn. 3. Quản trị viên nhập từ khóa vào ô tìm kiếm và chọn Tìm kiếm. 4. Hệ thống cập nhật danh sách món ăn trong giao diện Danh sách các món ăn. 5. Quản trị viên chọn Sửa món ăn của 1 món. 6. Hệ thống hiển thị giao diện Sửa món ăn. 7. Quản trị viên sửa các trường thông tin của món ăn. 8. Sửa món ăn thành công, hệ thống trả lại giao diện Danh sách các món ăn. 9. Quản trị viên chọn Xóa món ăn của 1 món trong danh sách. 10. Hệ thống hiển thị giao diện Xác nhận xóa món ăn. 11. Quản trị viên chọn Xác nhận. 12. Xóa món ăn thành công, hệ thống trả lại giao diện Danh sách các món ăn. 13. Quản trị viên chọn Thêm món ăn. 14. Hệ thống hiển thị giao diện Thêm món ăn. 15. Quản trị viên điền các trường thông tin của món ăn mới. 16. Thêm món ăn thành công, hệ thống trả lại giao diện Danh sách các món ăn.
Exception	Không

Bảng 2.11. Kịch bản usecase Quản lý các món ăn hiển thị cho người dùng

Usecase	Quản lý các nguyên liệu sẽ dùng cho món ăn
Actor	Quản trị viên
Pre-condition	Quản trị viên được cấp tài khoản
Post condition	Xem, thêm, sửa, xóa, tìm kiếm thông tin các nguyên liệu
Main events	<ul style="list-style-type: none"> 1. Sau khi đăng nhập, quản trị viên chọn Danh sách nguyên liệu. 2. Hệ thống hiển thị giao diện Danh sách các nguyên liệu. 3. Quản trị viên nhập từ khóa vào ô tìm kiếm và chọn Tìm kiếm. 4. Hệ thống cập nhật danh sách nguyên liệu trong giao diện Danh sách các nguyên liệu. 5. Quản trị viên chọn Sửa nguyên liệu của 1 nguyên liệu. 6. Hệ thống hiển thị giao diện Sửa nguyên liệu. 7. Quản trị viên sửa các trường thông tin của nguyên liệu. 8. Sửa nguyên liệu thành công, hệ thống trả lại giao diện Danh sách các nguyên liệu. 9. Quản trị viên chọn Xóa nguyên liệu của 1 món trong danh sách. 10. Hệ thống hiển thị giao diện Xác nhận xóa nguyên liệu. 11. Quản trị viên chọn Xác nhận. 12. Xóa nguyên liệu thành công, hệ thống trả lại giao diện Danh sách các nguyên liệu. 13. Quản trị viên chọn Thêm nguyên liệu. 14. Hệ thống hiển thị giao diện Thêm nguyên liệu. 15. Quản trị viên điền các trường thông tin của nguyên liệu mới. 16. Thêm nguyên liệu thành công, hệ thống trả lại giao diện Danh sách các nguyên liệu.
Exception	Không

Bảng 2.12. Kịch bản usecase Quản lý các nguyên liệu sẽ dùng cho món ăn

2.3. Phân tích yêu cầu phi chức năng

2.3.1. Yêu cầu bảo mật

Đồ án sử dụng FastAPI (Python) làm backend đã được xây dựng với mục tiêu đảm bảo an toàn cho hệ thống thông qua hai lớp bảo vệ chính: cơ chế phân quyền chặt chẽ theo vai trò và cấu hình chia sẻ tài nguyên gốc chéo (CORS).

Về bảo mật phân quyền, hệ thống thiết lập cơ chế kiểm soát truy cập dựa trên vai trò với hai đối tượng chính: quản trị viên và người dùng.

Về kiểm soát truy cập nguồn, hệ thống áp dụng cấu hình CORS ngay tại lớp Middleware của FastAPI. Cấu hình này hoạt động như một bộ lọc, chỉ cho phép các yêu cầu HTTP (GET, POST, PUT, PATCH, DELETE) được gửi từ các nguồn (domain/origin) đã được xác định trước và tin cậy (ví dụ: từ ứng dụng client). Cơ chế này giúp ngăn chặn các hành vi truy cập trái phép hoặc giả mạo yêu cầu từ các trang web hoặc nguồn bên ngoài không thuộc phạm vi quản lý của hệ thống, giảm thiểu rủi ro tấn công từ chối dịch vụ hoặc khai thác dữ liệu trái phép.

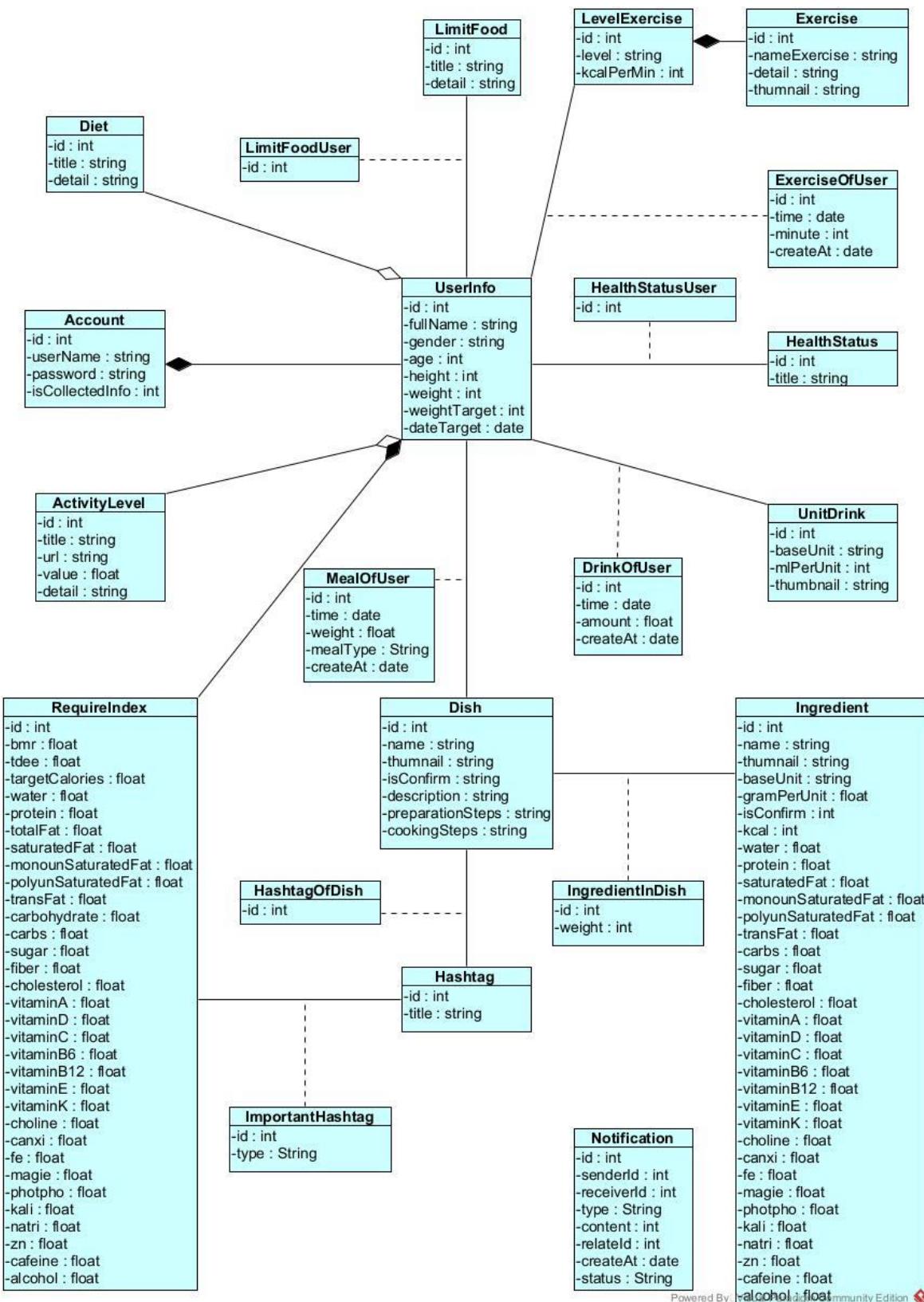
2.3.2. Yêu cầu hiệu năng

Hệ thống được xây dựng theo mô hình client-server, sử dụng Flutter cho ứng dụng di động (Mobile App) và FastAPI cho backend. Cách tiếp cận này tách biệt giao diện người dùng và logic xử lý dữ liệu, giúp tối ưu hiệu năng, khả năng mở rộng và bảo trì hệ thống.

Client (Flutter) tập trung hiển thị giao diện tương tác đa nền tảng (Android/iOS) với hiệu suất cao. Sử dụng ngôn ngữ Dart và bộ công cụ UI của Flutter để mang lại trải nghiệm người dùng mượt mà, tốc độ khung hình ổn định. Ứng dụng client sử dụng thư viện Dio hoặc Http để giao tiếp bát đồng bộ với server, giúp cập nhật dữ liệu (như danh sách món ăn, tin nhắn Chatbot) mà không làm gián đoạn trải nghiệm người dùng. Các kỹ thuật tối ưu như Lazy Loading (tải chậm nội dung) cũng được áp dụng để giảm tải cho thiết bị và tiết kiệm băng thông mạng.

Server (FastAPI) chịu trách nhiệm xử lý toàn bộ logic nghiệp vụ, từ tính toán định dường, truy vấn dữ liệu đến tích hợp mô hình AI. FastAPI được lựa chọn nhờ hiệu năng vượt trội với cơ chế xử lý bát đồng bộ (async/await), cho phép server tiếp nhận và xử lý đồng thời nhiều yêu cầu mà không bị tắc nghẽn (blocking), đặc biệt hiệu quả khi xử lý các tác vụ AI tốn thời gian. Server cung cấp các API RESTful chuẩn, trả về dữ liệu định dạng JSON giúp việc trao đổi thông tin giữa Client và Server diễn ra nhanh chóng và chính xác.

2.4. Phân tích Biểu đồ lớp



Ảnh 2.2. Biểu đồ lớp của hệ thống

Sau đây là bảng chi tiết thông tin của các lớp:

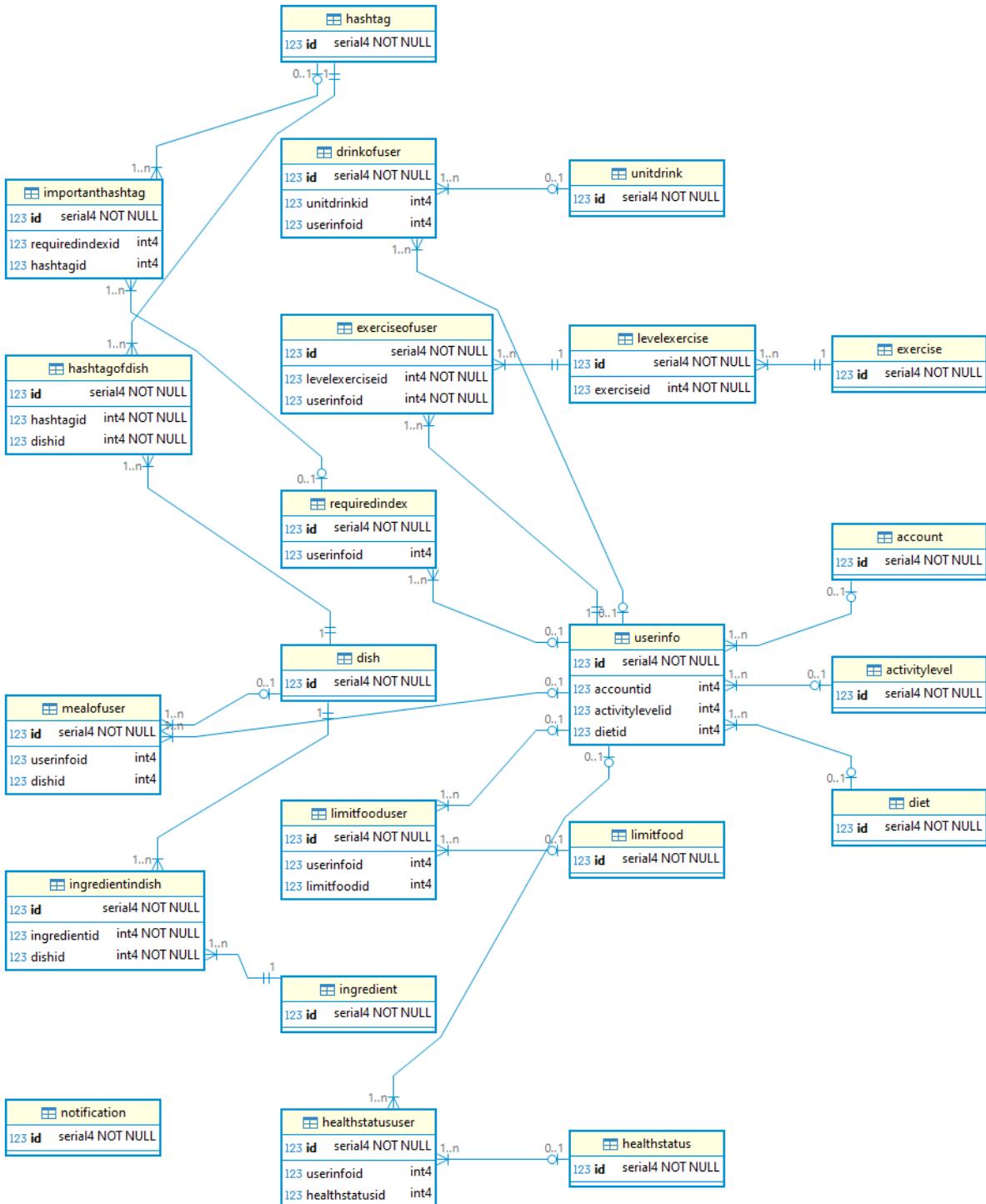
STT	Tên lớp	Mô tả
1	Account	Lớp Account dùng để lưu thông tin đăng nhập của người dùng, bao gồm tên đăng nhập, mật khẩu và trạng thái đã thu thập đủ thông tin cá nhân hay chưa.
2	UserInfo	Lớp UserInfo lưu trữ thông tin cá nhân và sức khỏe của người dùng như họ tên, giới tính, tuổi, chiều cao, cân nặng, mục tiêu cân nặng và thời gian đạt mục tiêu.
3	RequireIndex	Lớp RequireIndex lưu các chỉ số dinh dưỡng cần thiết của người dùng như BMR, TDEE, lượng calo mục tiêu, nước, protein, chất béo, vitamin và khoáng chất.
4	Diet	Lớp Diet mô tả chế độ ăn uống, bao gồm tiêu đề và mô tả chi tiết cho từng chế độ ăn phù hợp với người dùng.
5	ActivityLevel	Lớp ActivityLevel mô tả mức độ vận động của người dùng (ít vận động, trung bình, nhiều) để phục vụ tính toán nhu cầu năng lượng hàng ngày.
6	HealthStatus	Lớp HealthStatus mô tả các tình trạng sức khỏe (ví dụ: tiểu đường, huyết áp cao, béo phì...).
7	HealthStatusUser	Lớp trung gian HealthStatusUser dùng để liên kết người dùng với các tình trạng sức khỏe tương ứng.
8	LimitFood	Lớp LimitFood dùng để định nghĩa các loại thực phẩm cần hạn chế đối với người dùng trong chế độ ăn hoặc tình trạng sức khỏe nhất định.
9	LimitFoodUser	Lớp trung gian LimitFoodUser dùng để liên kết người dùng với các thực phẩm bị hạn chế.
10	Ingredient	Lớp Ingredient lưu thông tin chi tiết về nguyên liệu, bao gồm đơn vị cơ bản, giá trị dinh dưỡng và các thành phần vi chất.

11	Dish	Lớp Dish mô tả món ăn, bao gồm tên món, hình ảnh, mô tả, nguyên liệu, cách chuẩn bị và các bước nấu ăn.
12	MealOfUser	Là lớp chứa nhật ký bữa ăn từng bữa của người dùng.
13	Exercise	Lớp Exercise mô tả các bài tập thể dục, bao gồm tên bài tập, mô tả chi tiết và hình ảnh minh họa.
14	LevelExercise	Lớp LevelExercise xác định mức độ bài tập và lượng calo tiêu hao trên mỗi phút cho từng mức độ.
15	ExerciseOfUser	Lớp ExerciseOfUser lưu lịch sử tập luyện của người dùng, bao gồm thời gian tập, số phút và ngày tạo.
16	UnitDrink	Lớp UnitDrink định nghĩa đơn vị uống (cốc, chai, ml...) và lượng nước tương ứng cho mỗi đơn vị.
17	DrinkOfUser	Lớp DrinkOfUser lưu thông tin lượng nước người dùng đã uống theo thời gian.
18	Hashtag	Lớp Hashtag dùng để gắn thẻ cho món ăn nhằm hỗ trợ tìm kiếm và phân loại.
19	ImportantHashtag	Lớp ImportantHashtag dùng để xác định các hashtag quan trọng phục vụ gợi ý hoặc lọc dữ liệu cho người dùng cụ thể.
20	HashtagOfDish	Lớp trung gian HashtagOfDish liên kết món ăn với các hashtag tương ứng.
21	Notification	Lớp Notification dùng để quản lý các thông báo gửi đến người dùng, bao gồm người gửi, người nhận, nội dung, loại thông báo và trạng thái.
22	IngredientInDish	Lớp trung gian IngredientInDish dùng để xác định nguyên liệu và khối lượng nguyên liệu sử dụng trong từng món ăn.

Bảng 2.13. Bảng chi tiết thông tin các lớp

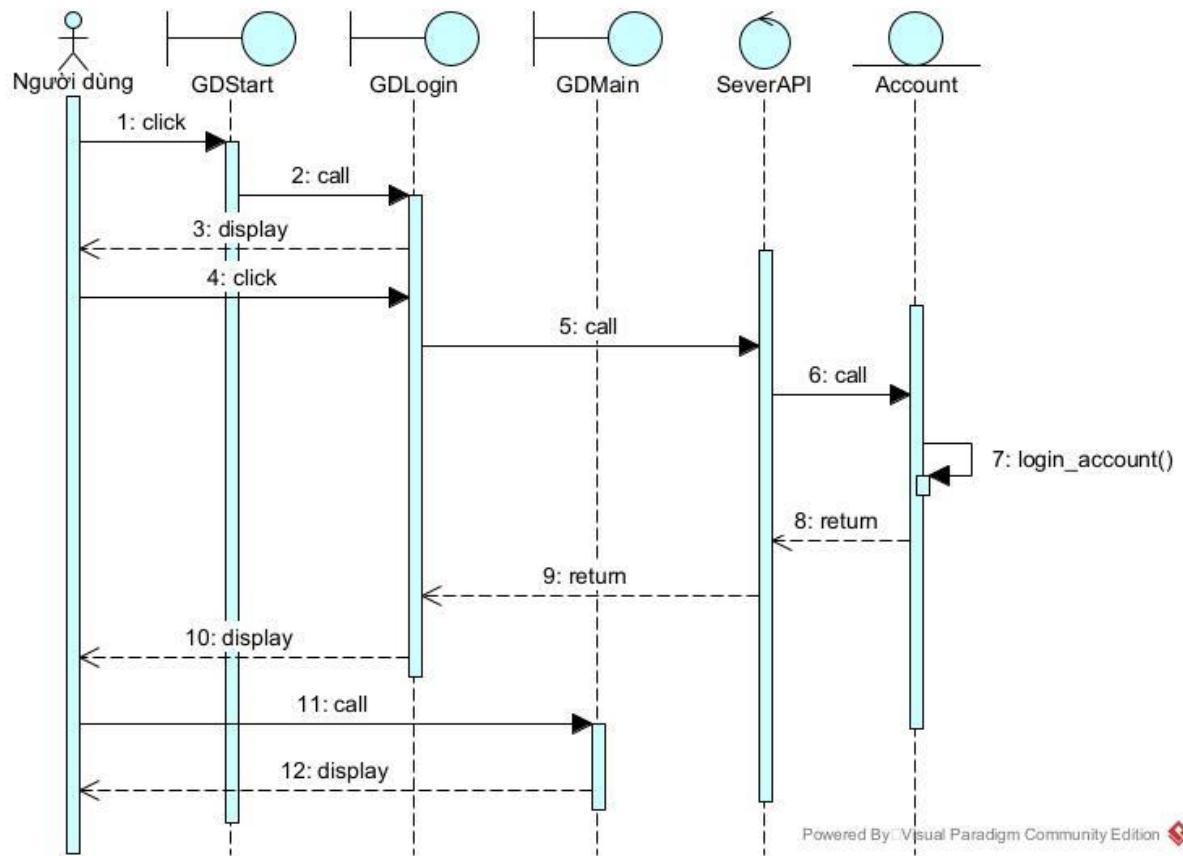
2.5. Thiết kế cơ sở dữ liệu hệ thống

Dựa trên các lớp thực thể đã mô tả ở trên, thiết kế cơ sở dữ liệu của hệ thống được xây dựng tương ứng như sau:

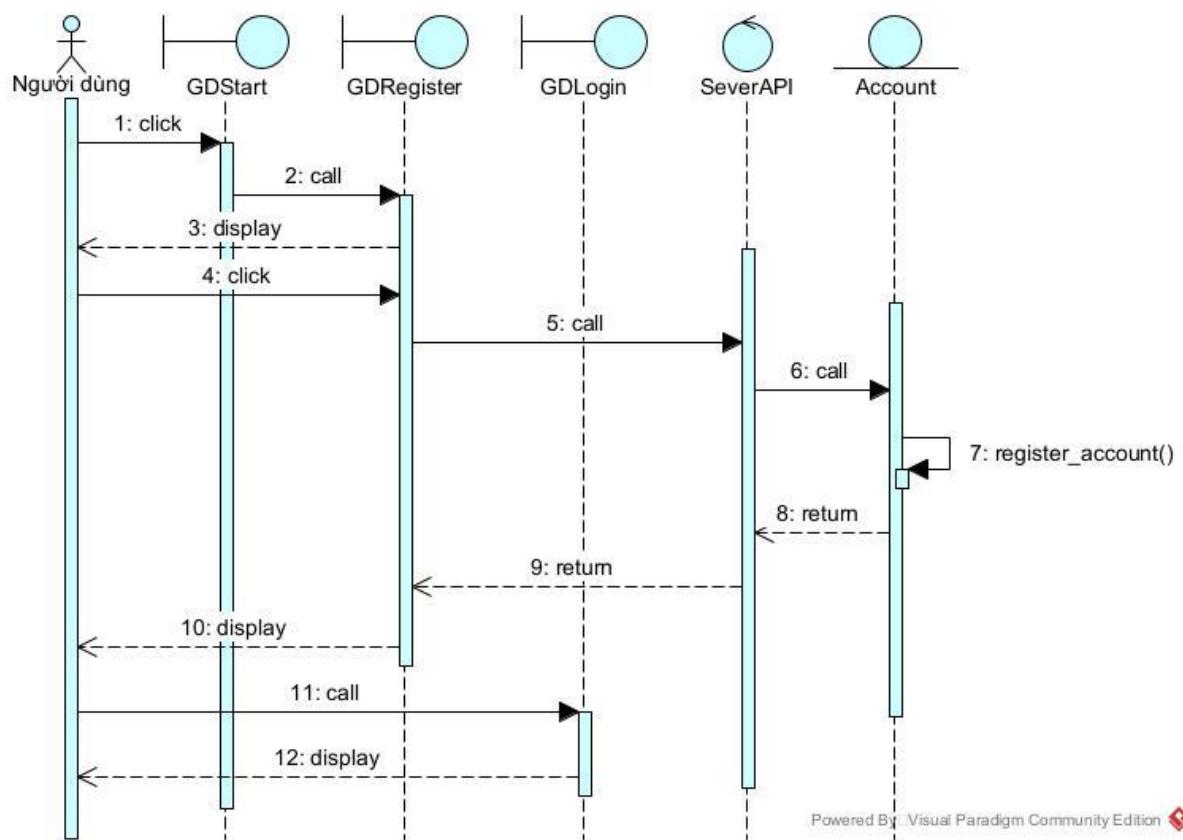


Ảnh 2.3. Cơ sở dữ liệu hệ thống

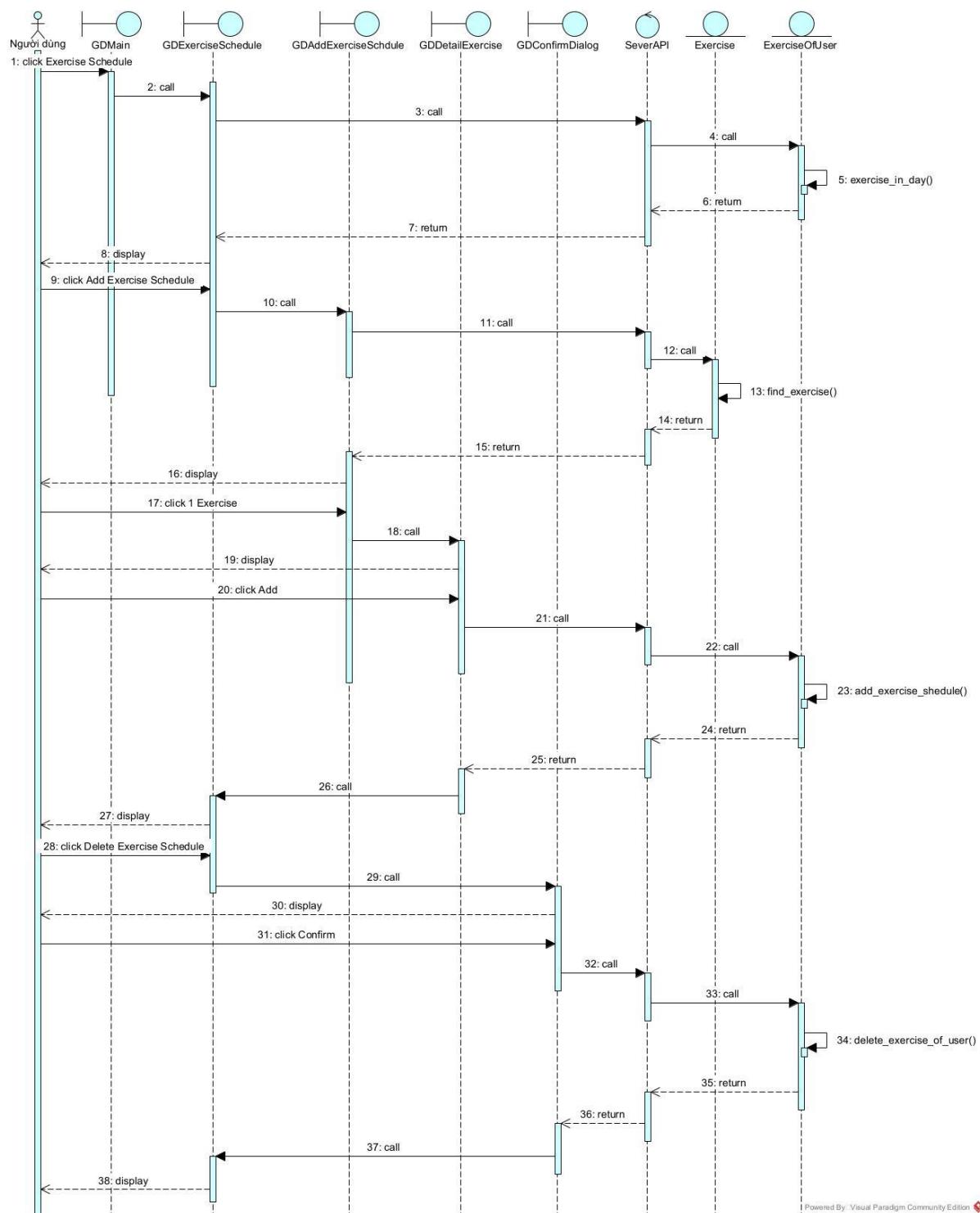
2.6. Thiết kế biểu đồ tuần tự cho từng chức năng



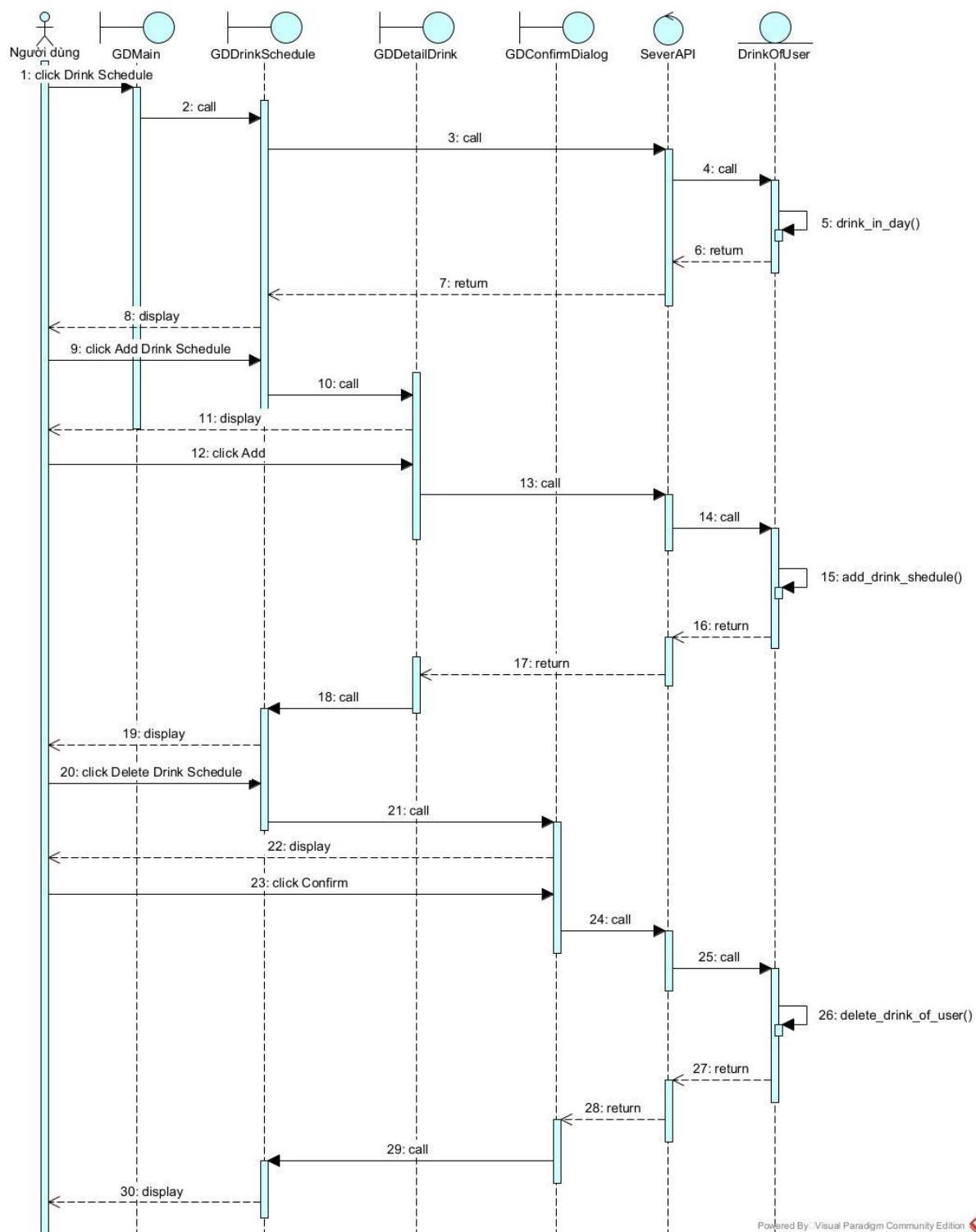
Ảnh 2.4. Biểu đồ tuần tự module Đăng nhập



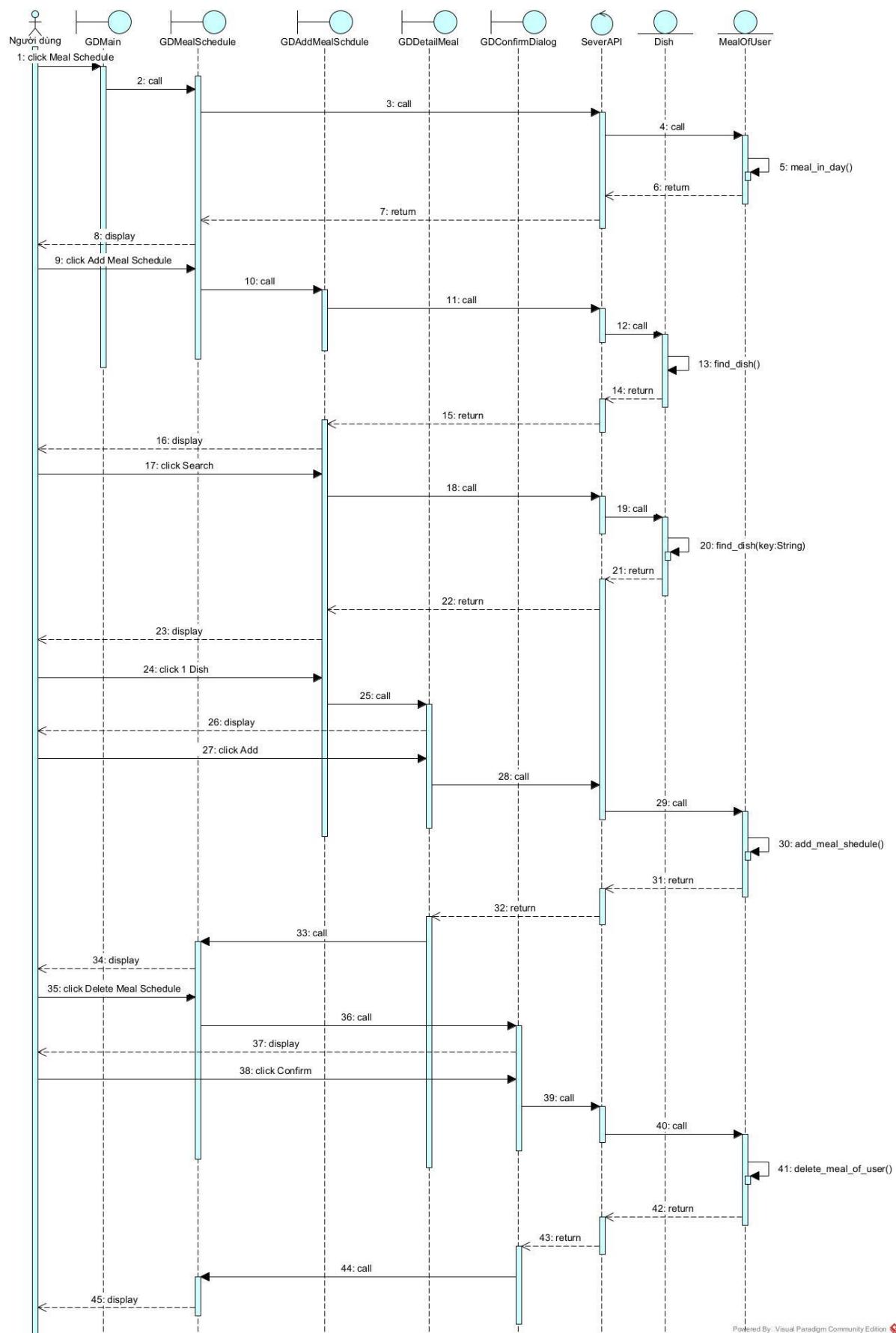
Ảnh 2.5. Biểu đồ tuần tự module Đăng ký



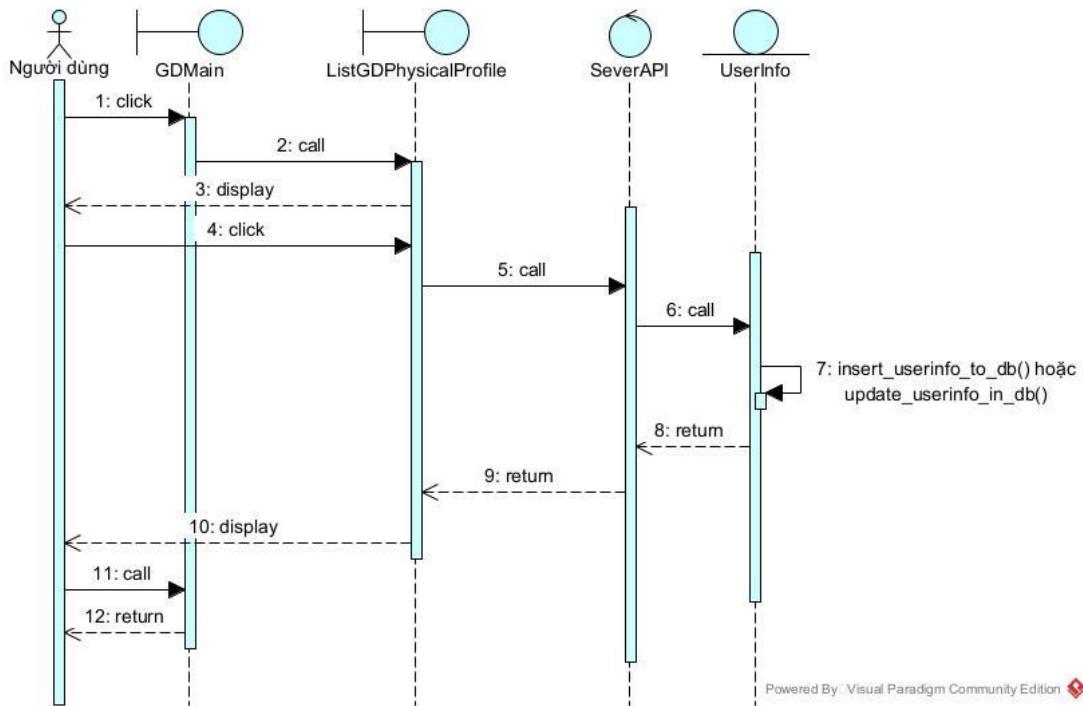
Ảnh 2.6. Biểu đồ tuần tự module Quản lý lịch trình tập thể dục



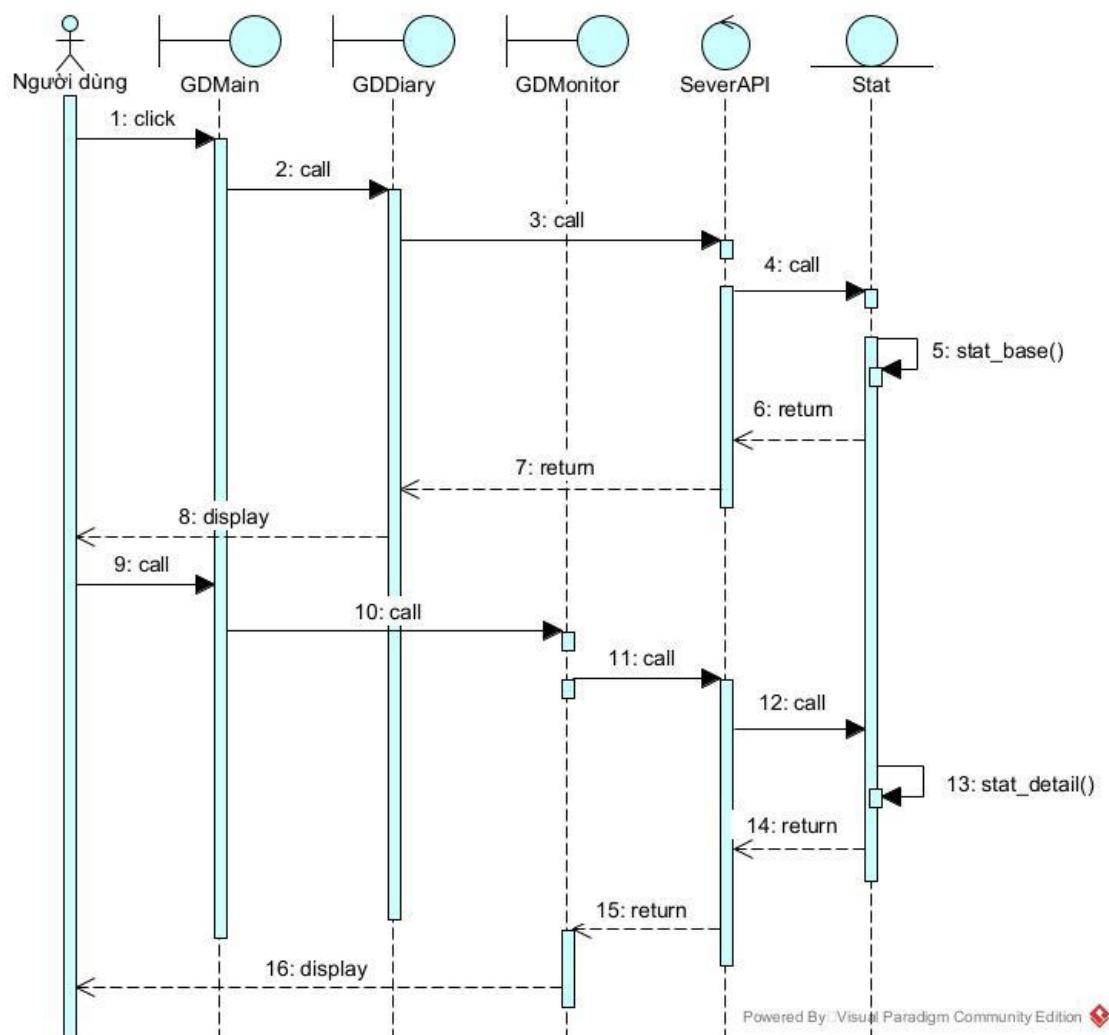
Ảnh 2.7. Biểu đồ tuần tự module Quản lý lượng nước uống



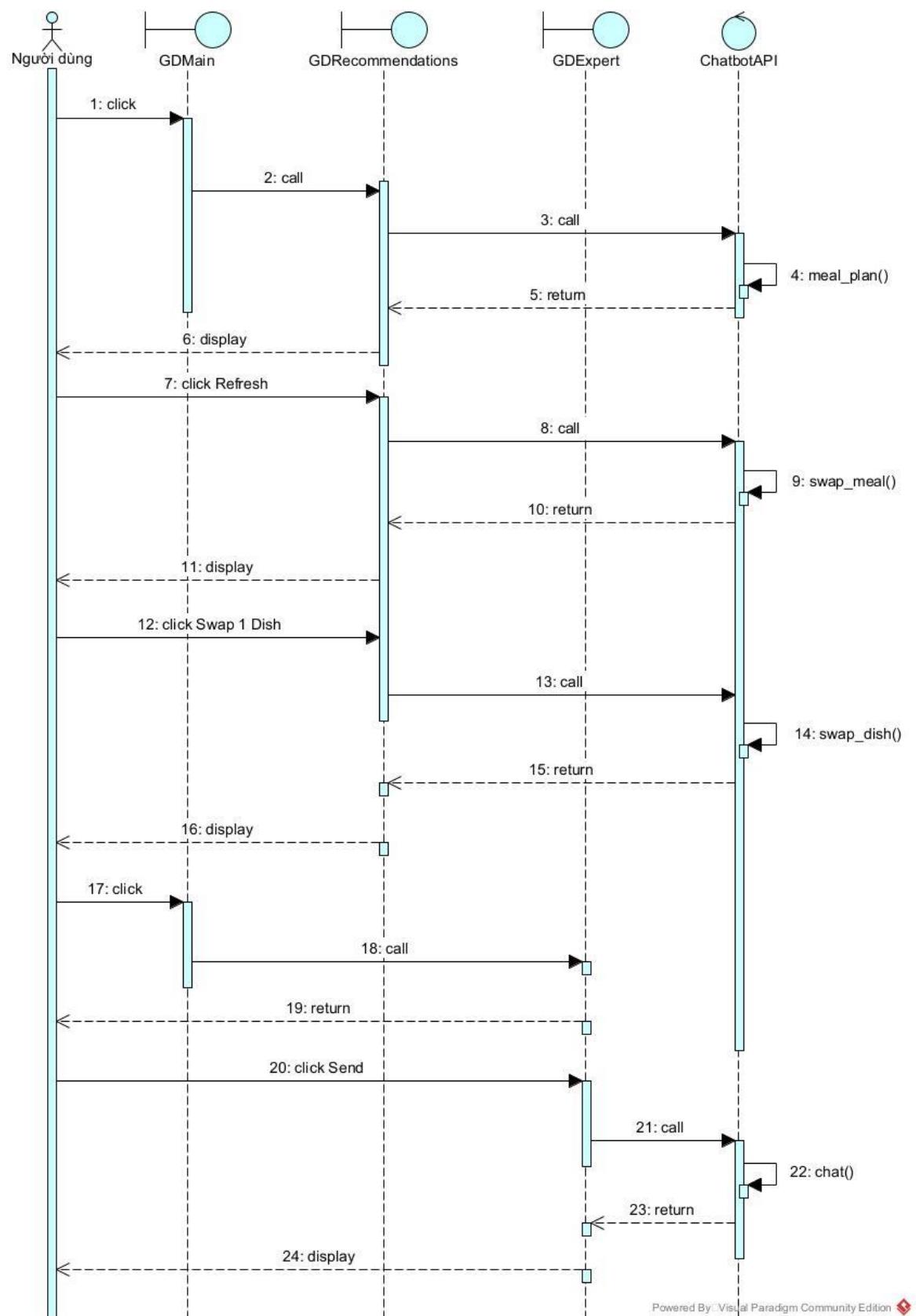
Ảnh 2.8. Biểu đồ tuần tự module Quản lý các món ăn từng bữa trong ngày



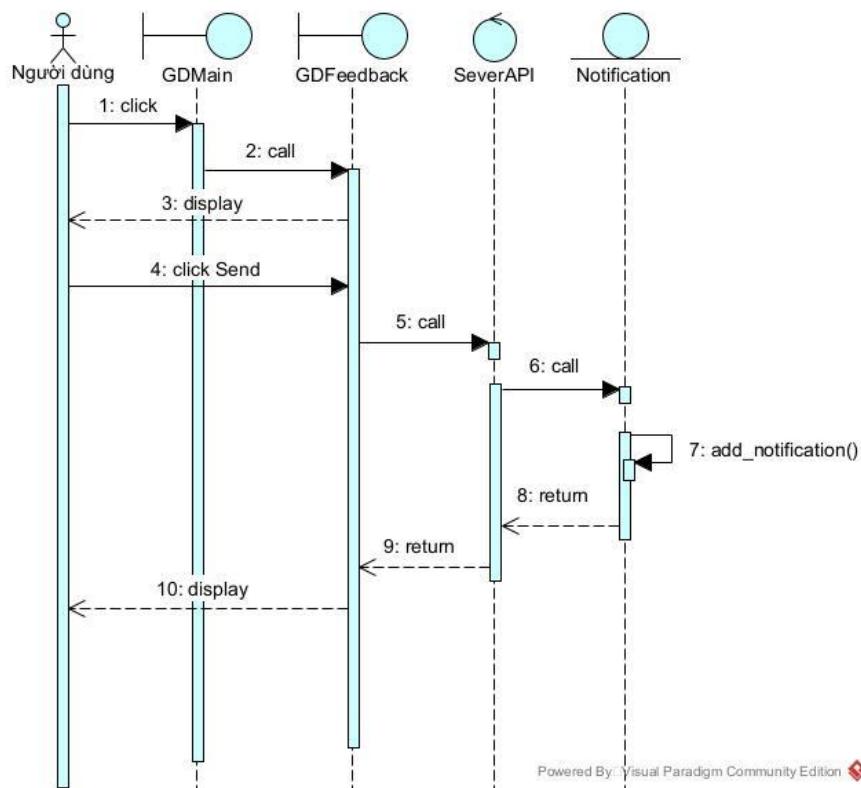
Ảnh 2.9. Biểu đồ tuần tự module Quản lý hồ sơ thể chất



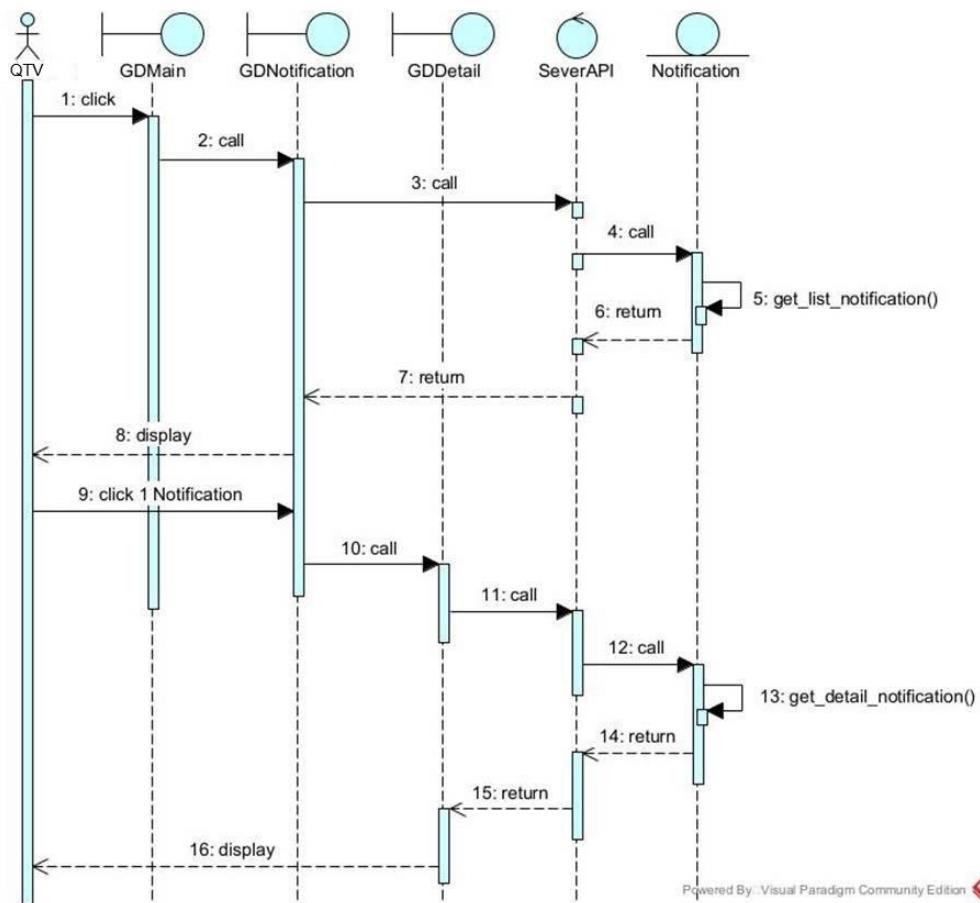
Ảnh 2.10. Biểu đồ tuần tự module Thống kê định dưỡng trong ngày



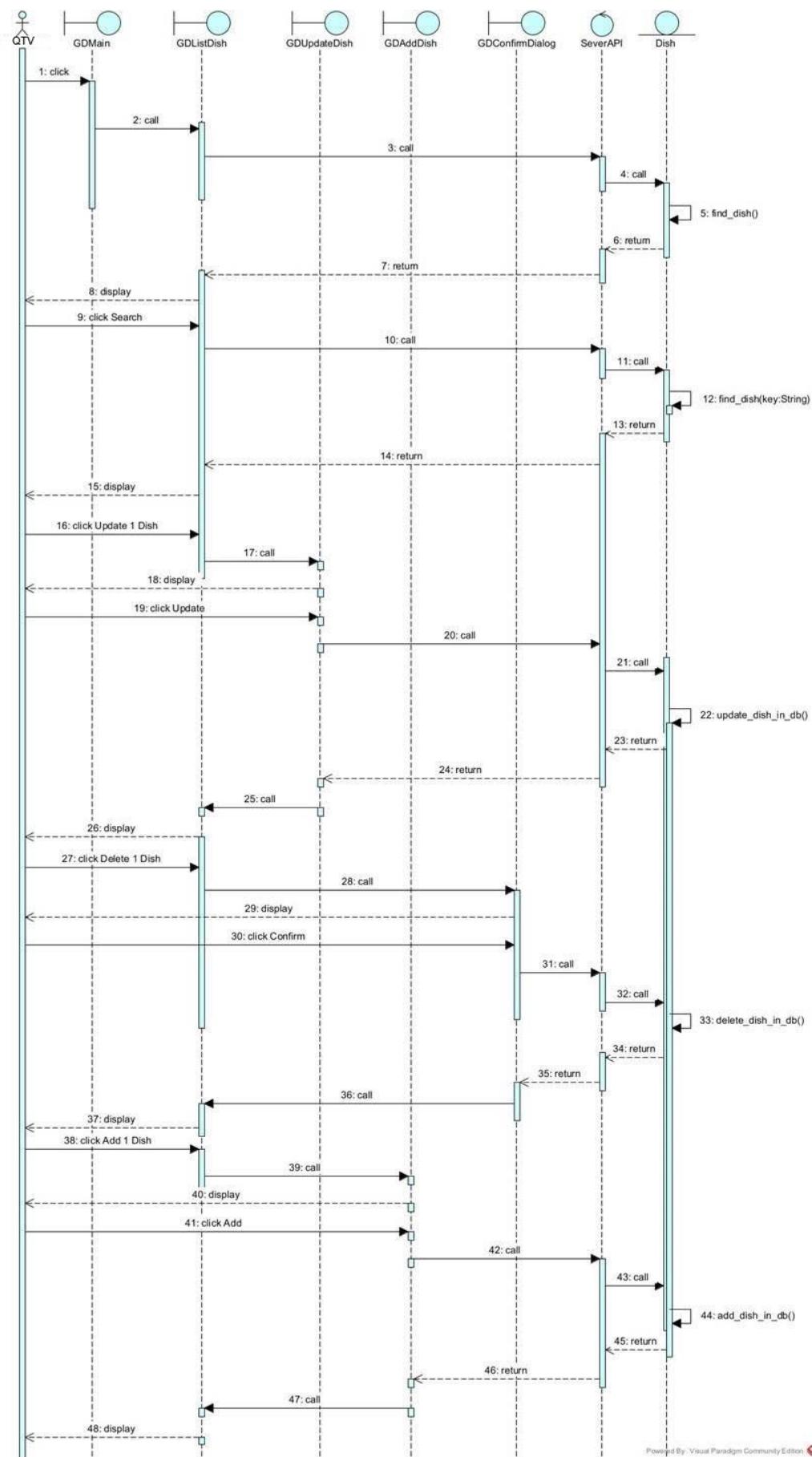
Ảnh 2.11. Biểu đồ tuần tự module Hỏi đáp với chuyên gia AI



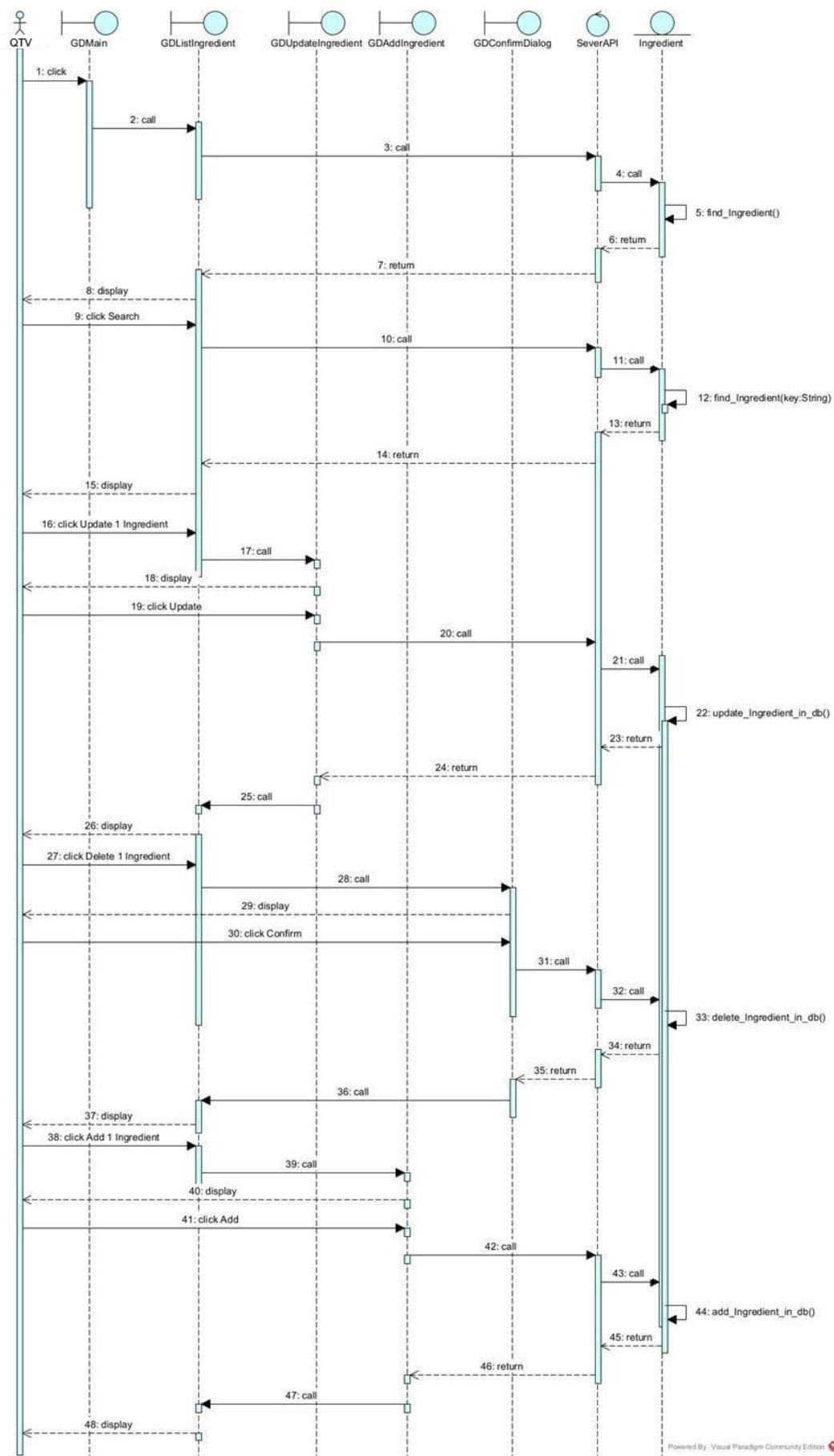
Ảnh 2.12. Biểu đồ tuần tự module Gửi góp ý cho quản trị viên



Ảnh 2.13. Biểu đồ tuần tự module Nhận thông báo góp ý của người dùng



Ảnh 2.14. Biểu đồ tuần tự module Quản lý các món ăn hiển thị cho người dùng



Ảnh 2.15. Biểu đồ tuần tự module Quản lý các nguyên liệu sẽ dùng cho món ăn

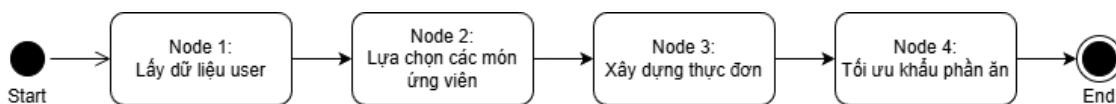
2.7. Thiết kế luồng xử lý chức năng AI

2.7.1. Luồng hoạt động chức năng gợi ý thực đơn

Chức năng gợi ý thực đơn giúp hệ thống xây dựng thực đơn phù hợp dựa trên thông tin cá nhân, mục tiêu dinh dưỡng và tình trạng sức khỏe của người dùng.

- Input: ID người dùng, danh sách các bữa cần gợi ý.
- Output: Danh sách các món gợi ý đi kèm với khẩu phần từng món.

Luồng hoạt động của chức năng gợi ý thực đơn như sau:



Ảnh 2.16. Luồng hoạt động chức năng gợi ý thực đơn

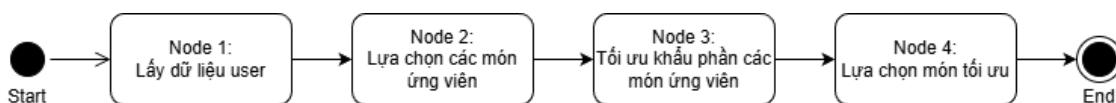
- Node 1: Lấy dữ liệu người dùng từ cơ sở dữ liệu dựa trên ID.
- Node 2: Áp dụng Retriever tới Vector Database chứa dữ liệu món ăn (gồm Embedding dữ liệu thông tin người dùng, Semantic Search kết hợp Heuristic Ranking dựa trên thông tin người dùng) để truy xuất các món ăn ứng viên.
- Node 3: Sử dụng LLM để lựa chọn tổ hợp món ăn phù hợp với từng bữa.
- Node 4: Áp dụng thuật toán SLSQP để điều chỉnh lượng ăn của từng món nhằm đạt mục tiêu dinh dưỡng, sau đó trả kết quả về cho người dùng và kết thúc luồng.

2.7.2. Luồng hoạt động chức năng thay đổi món ăn

Chức năng thay đổi món ăn cho phép người dùng thay thế một món ăn trong một bữa nào đó bằng một món ăn khác mà hệ thống gợi ý.

- Input: ID người dùng, dữ liệu món ăn người dùng muốn thay thế.
- Output: 1 món ăn khác thay thế.

Luồng hoạt động của chức năng thay đổi món ăn như sau:



Ảnh 2.17. Luồng hoạt động chức năng thay đổi món ăn

- Node 1: Lấy dữ liệu người dùng từ cơ sở dữ liệu dựa trên ID.
- Node 2: Áp dụng Retriever tới Vector Database chứa dữ liệu món ăn (gồm Embedding dữ liệu thông tin người dùng kèm thông tin món cũ và Semantic Search) để truy xuất các món ăn ứng viên.

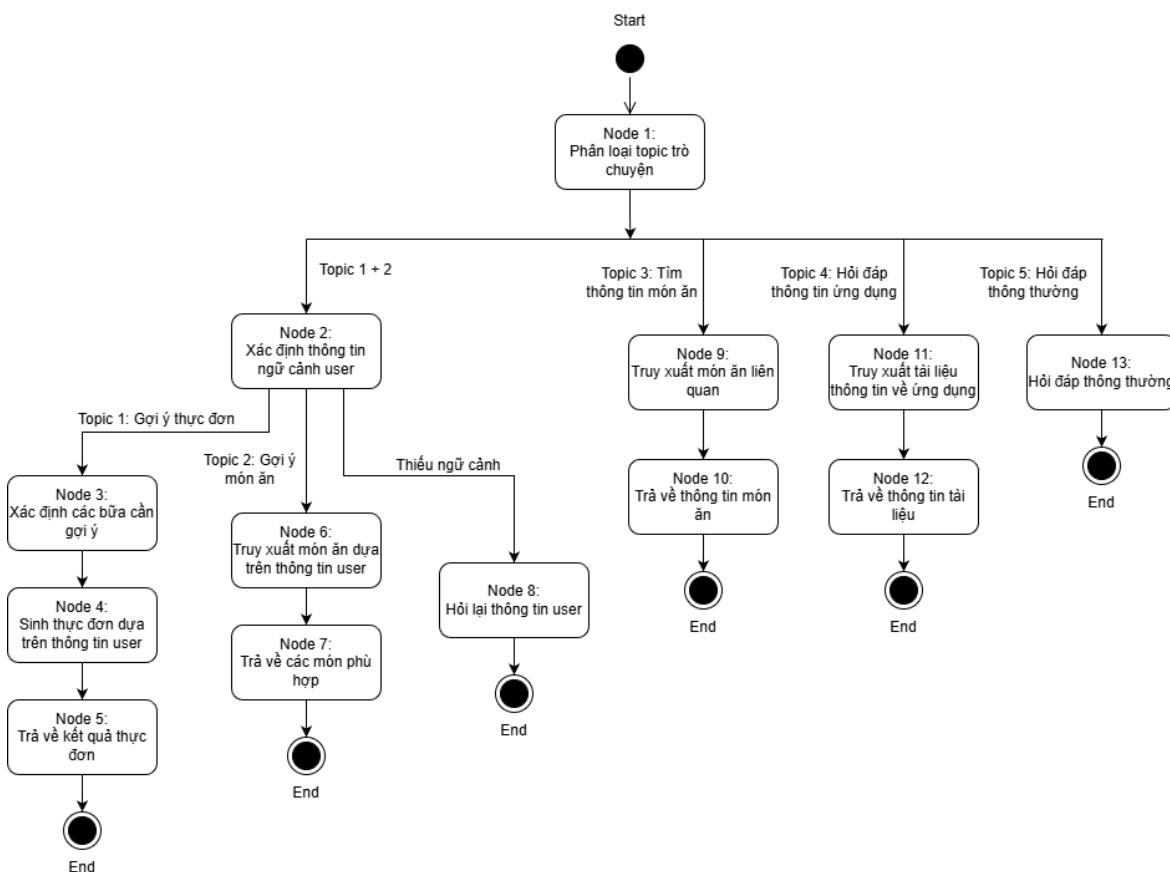
- Node 3: Áp dụng thuật toán SLSQP để điều chỉnh lượng ăn của từng món ứng viên thay thế sao cho gần dinh dưỡng với món cũ. Lọc ra các món sát với dinh dưỡng món cũ nhất.
- Node 4: Sử dụng LLM để chọn món thay thế từ danh sách món ứng viên sao cho phù hợp và kết thúc luồng.

2.7.3. Luồng hoạt động chức năng chatbot

Chức năng chatbot cho phép người dùng nói chuyện, trao đổi về các chủ đề như gợi ý thực đơn người dùng tùy biến, gợi ý món ăn, tìm kiếm thông tin món ăn, thông tin về ứng dụng cũng như hỏi đáp các thông tin về dinh dưỡng, sức khỏe, chế độ tập luyện.

- Input: ID người dùng, ID luồng trò chuyện, tin nhắn người dùng.
- Output: Tin nhắn phản hồi từ chatbot.

Luồng hoạt động của chức năng chatbot như sau:



Ảnh 2.18. Luồng hoạt động chức năng chatbot

- Node 1: Sử dụng LLM để phân loại chủ đề trò chuyện nhằm xác định mục đích của người dùng. Các chủ đề chính bao gồm:
 - Topic 1: Gợi ý thực đơn.
 - Topic 2: Gợi ý món ăn.

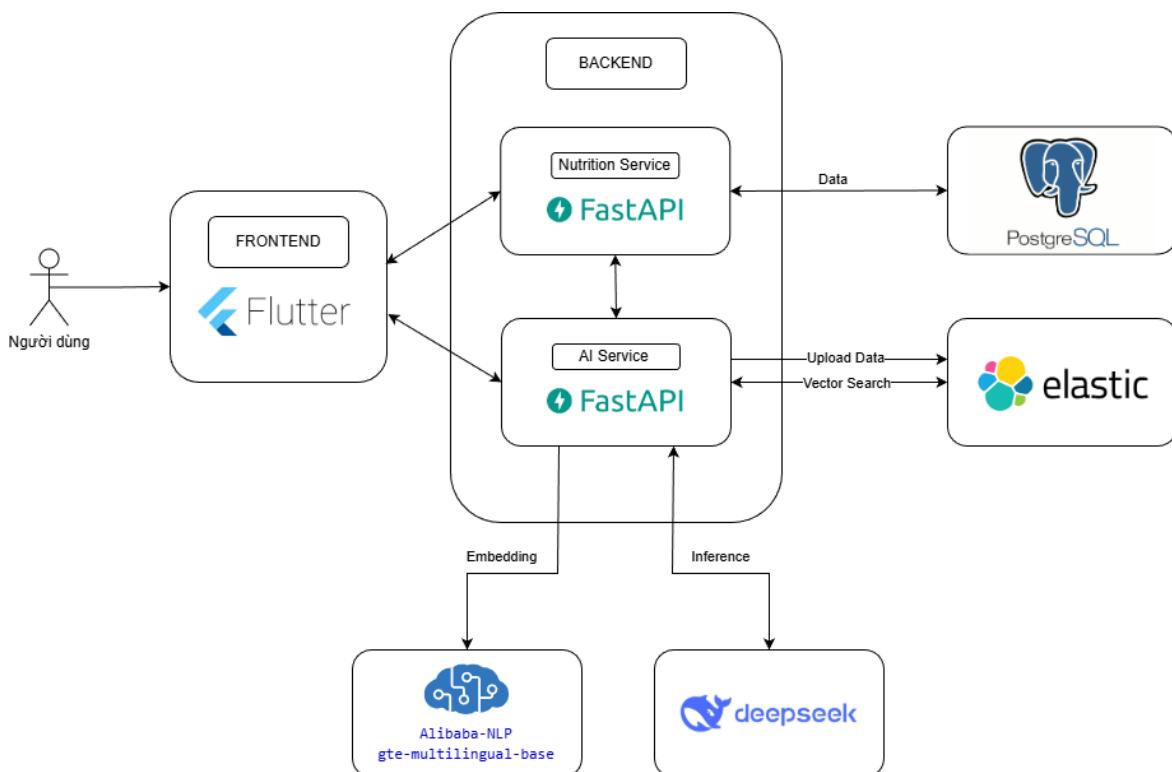
- Topic 3: Tìm kiếm thông tin về món ăn.
- Topic 4: Tìm kiếm thông tin về ứng dụng.
- Topic 5: Hỏi đáp thông thường.
- Node 2: Sử dụng LLM xác định thông tin về người dùng (tuổi, giới tính, chiều cao, ...) đang trao đổi nhằm phục vụ Topic 1 (Gợi ý thực đơn) hoặc Topic 2 (Gợi ý món ăn) có đủ hay chưa.
 - Nếu thiếu ngũ cành, hệ thống chuyển sang Node 8 để hỏi bổ sung thông tin từ người dùng.
 - Nếu đã đủ ngũ cành, hệ thống tiếp tục chuyển đến nhánh dựa trên chủ đề đã được phân ở Node 1.
- Topic 1: Gợi ý thực đơn.
 - Node 3: Sử dụng LLM xác định những bữa cần gợi ý.
 - Node 4: Sử dụng chức năng gợi ý thực đơn đã xây dựng ở mục 2.7.1 để cho ra kết quả.
 - Node 5: Trả về kết quả thực đơn hoàn chỉnh cho người dùng và kết thúc luồng.
- Topic 2: Gợi ý món ăn.
 - Node 6: Truy xuất các món ăn phù hợp dựa trên thông tin và tin nhắn của người dùng (sử dụng cơ chế Retriever của RAG tới Vector Database chứa món ăn).
 - Node 7: Sử dụng LLM kết quả 2-3 món ứng viên cho người dùng và kết thúc luồng.
- Topic 3: Tìm kiếm thông tin món ăn.
 - Node 9: Truy xuất các món ăn dựa trên tin nhắn của người dùng (sử dụng Retriever tới Vector Database chứa món ăn).
 - Node 10: Sử dụng LLM trả về thông tin món ăn mà người dùng muốn tìm kiếm và kết thúc luồng.
- Topic 4: Hỏi đáp thông tin về ứng dụng.
 - Node 11: Truy xuất các phần tài liệu chứa thông tin mà người dùng muốn tìm kiếm như thông tin, các chức năng, điều khoản của ứng dụng... (Sử dụng Retriever tới Vector Database chứa tài liệu thông tin ứng dụng).
 - Node 12: Sử dụng LLM tả về kết quả thông tin truy xuất được mà người dùng muốn tìm kiếm và kết thúc luồng.
- Topic 5: Hỏi đáp thông thường.
 - Node 13: Sử dụng LLM trả lời các câu hỏi của người dùng (khác các topic 1-4), đồng thời giới hạn chỉ trả lời các câu hỏi liên quan đến dinh dưỡng, sức khỏe, chế độ luyện tập và kết thúc luồng.

2.8. Tổng kết chương

Chương 2 đã tập trung phân tích nghiệp vụ và xác định các yêu cầu chức năng, phi chức năng của hệ thống ứng dụng quản lý dinh dưỡng tích hợp chatbot AI. Trên cơ sở đó, chương đã trình bày thiết kế tổng thể hệ thống, bao gồm các use case, sơ đồ xử lý, luồng hoạt động của các chức năng AI và thiết kế cơ sở dữ liệu. Những nội dung trong chương là cơ sở trực tiếp cho việc cài đặt, triển khai và kiểm thử hệ thống được trình bày trong Chương 3.

Chương 3. CÀI ĐẶT VÀ KIỂM THỬ HỆ THỐNG

3.1. Kiến trúc hệ thống



Ảnh 3.1. Kiến trúc hệ thống

Hệ thống được xây dựng theo kiến trúc Client–Server. Trong đó máy khách gửi các HTTP request đến máy chủ, máy chủ tiếp nhận, xử lý và trả kết quả về cho máy khách dưới dạng JSON.

Phía Frontend được phát triển bằng Flutter, đóng vai trò giao diện tương tác trực tiếp với người dùng, hiển thị thông tin thực đơn, dữ liệu dinh dưỡng và kết quả phản hồi từ hệ thống AI.

Phía Backend được triển khai bằng FastAPI và được tách thành hai dịch vụ chính. Nutrition Service chịu trách nhiệm xử lý các chức năng nghiệp vụ như quản lý thông tin người dùng, thực đơn, dữ liệu dinh dưỡng và giao tiếp với cơ sở dữ liệu PostgreSQL để lưu trữ dữ liệu quan hệ.

Bên cạnh đó, AI Service đảm nhiệm các chức năng liên quan đến trí tuệ nhân tạo như gợi ý bữa ăn, tìm kiếm món ăn và chatbot. Dịch vụ này sử dụng mô hình Alibaba-NLP/gte-multilingual-base để tạo vector embedding và lưu trữ, tìm kiếm ngữ nghĩa trong Elasticsearch. Khi cần sinh câu trả lời, AI Service gửi prompt mở rộng đến mô hình DeepSeek để thực hiện suy luận và tạo phản hồi.

3.2. Môi trường triển khai

Hệ thống được phát triển và triển khai dựa trên các công cụ và môi trường sau:

Thiết bị phần cứng:

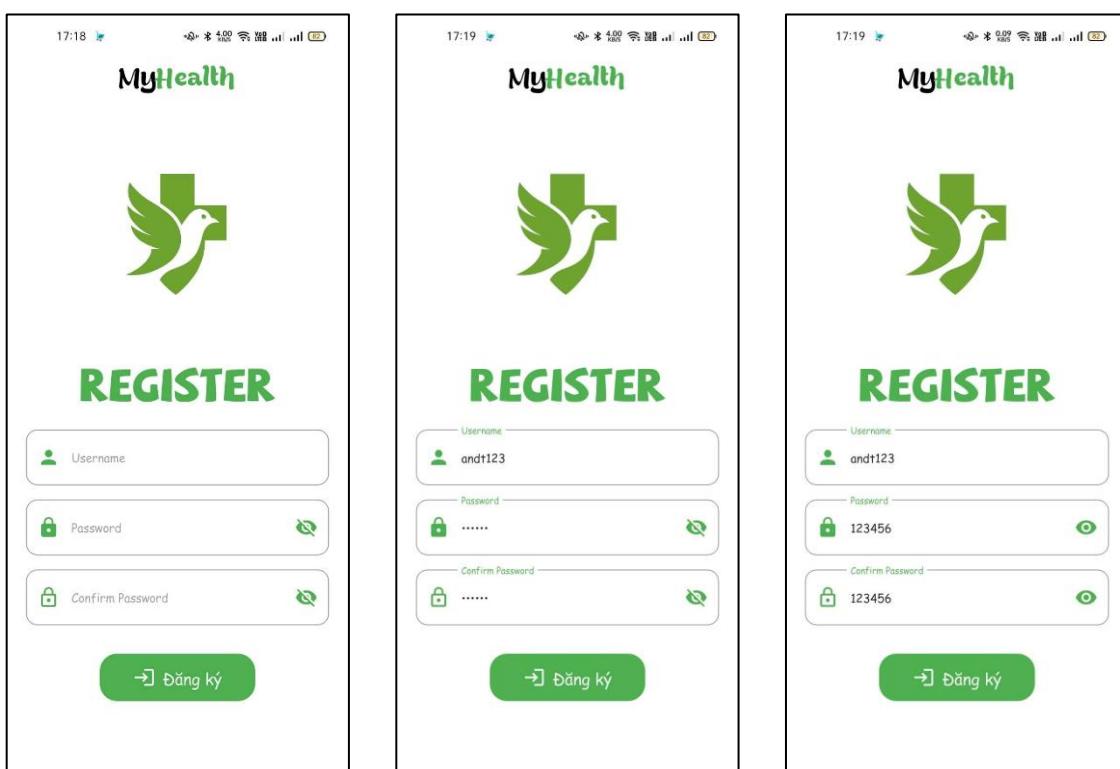
- Máy tính cá nhân: Sử dụng để lập trình, chạy server cục bộ.
- Thiết bị di động: Các điện thoại thông minh chạy hệ điều hành Android hoặc iOS để cài đặt và kiểm thử ứng dụng.

Công cụ và Phần mềm:

- Môi trường lập trình (IDE): Visual Studio Code và Android Studio.
- Nền tảng phát triển: Flutter và Python.
- Framework & Thư viện: FastAPI, LangChain, LangGraph, Dio.
- Cơ sở dữ liệu: PostgreSQL.
- Dịch vụ bên thứ ba: FCM, ElasticSearch, Deepseek API.

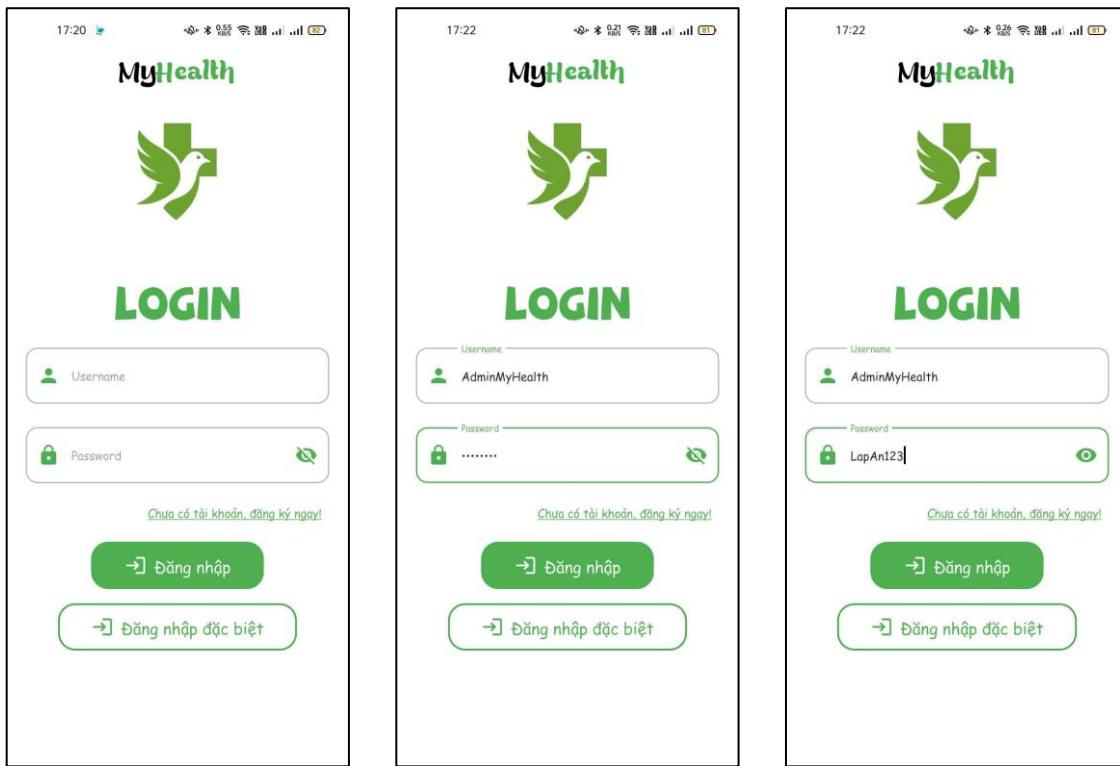
3.3. Giao diện và chức năng đã hoàn thành

3.3.1. Giao diện module đăng nhập



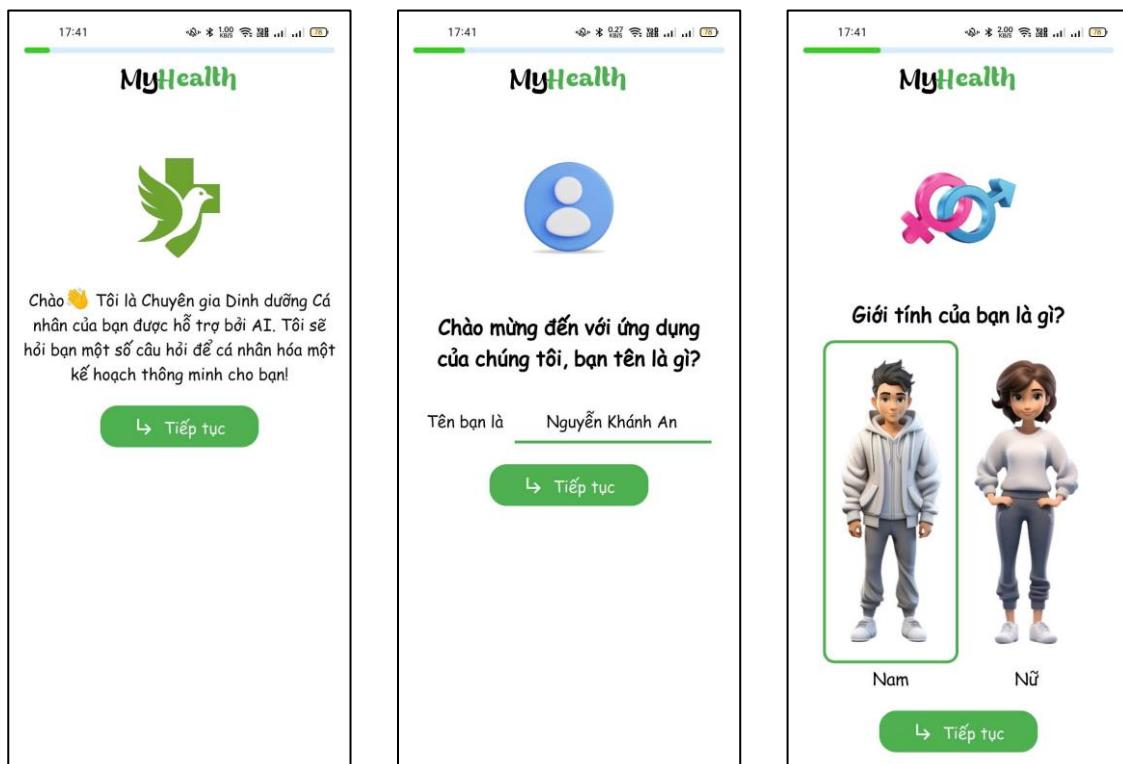
Ảnh 3.2. Giao diện module đăng nhập

3.3.2. Giao diện module đăng ký

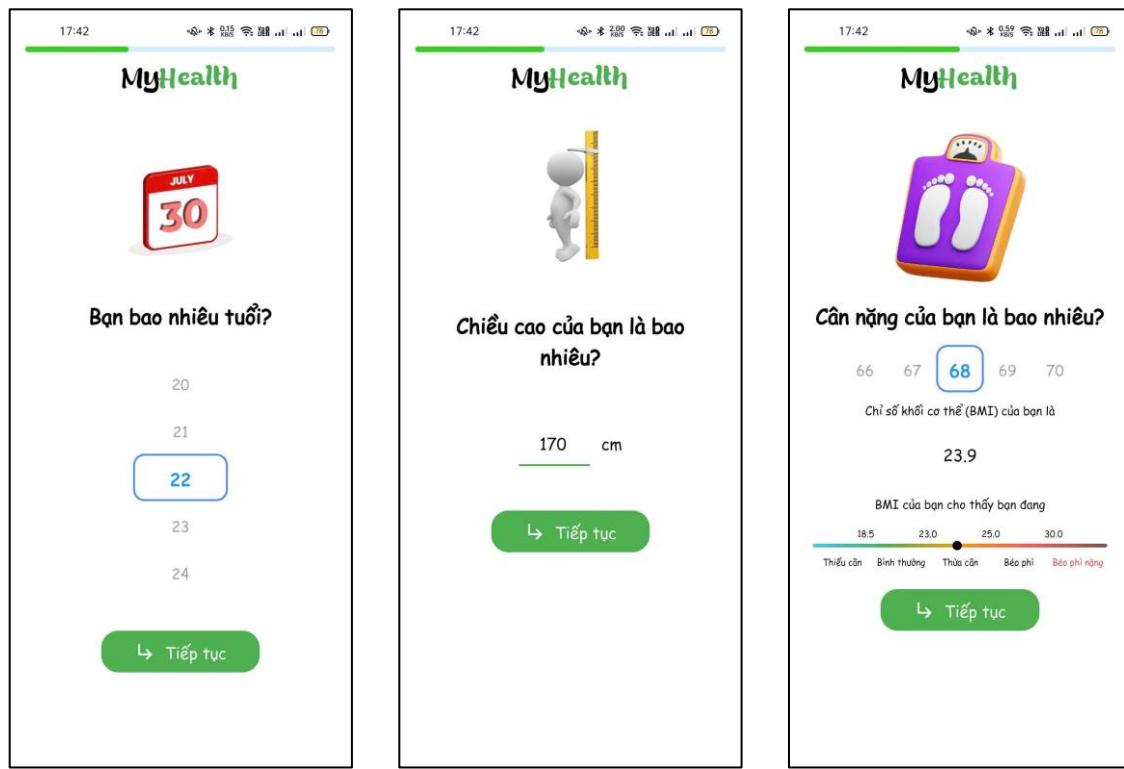


Ảnh 3.3. Giao diện module đăng ký

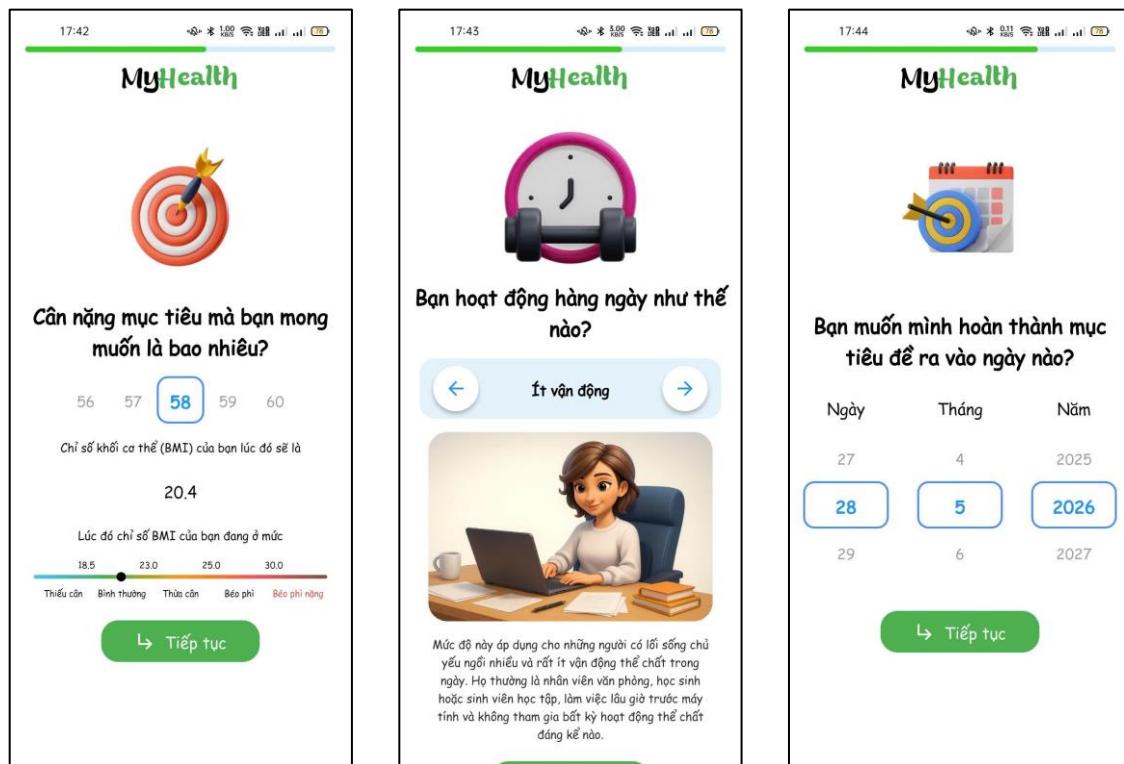
3.3.3. Giao diện module quản lý hồ sơ thể chất



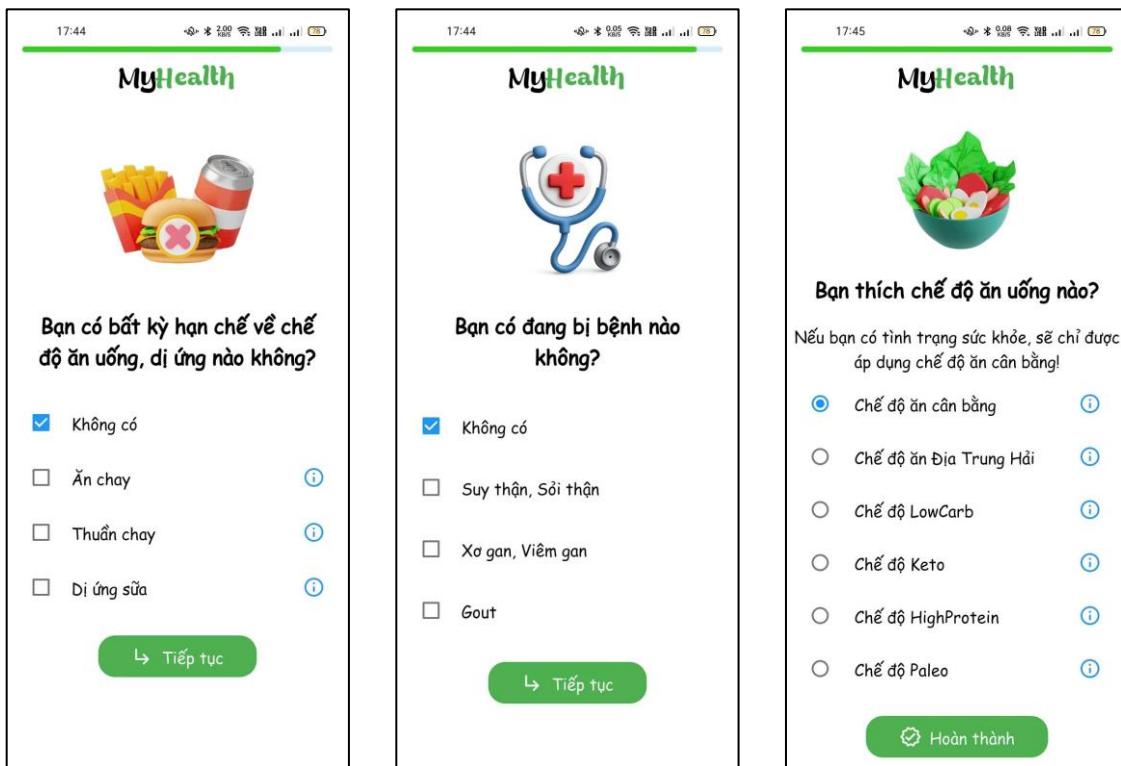
Ảnh 3.4. Giao diện module quản lý hồ sơ thể chất (1)



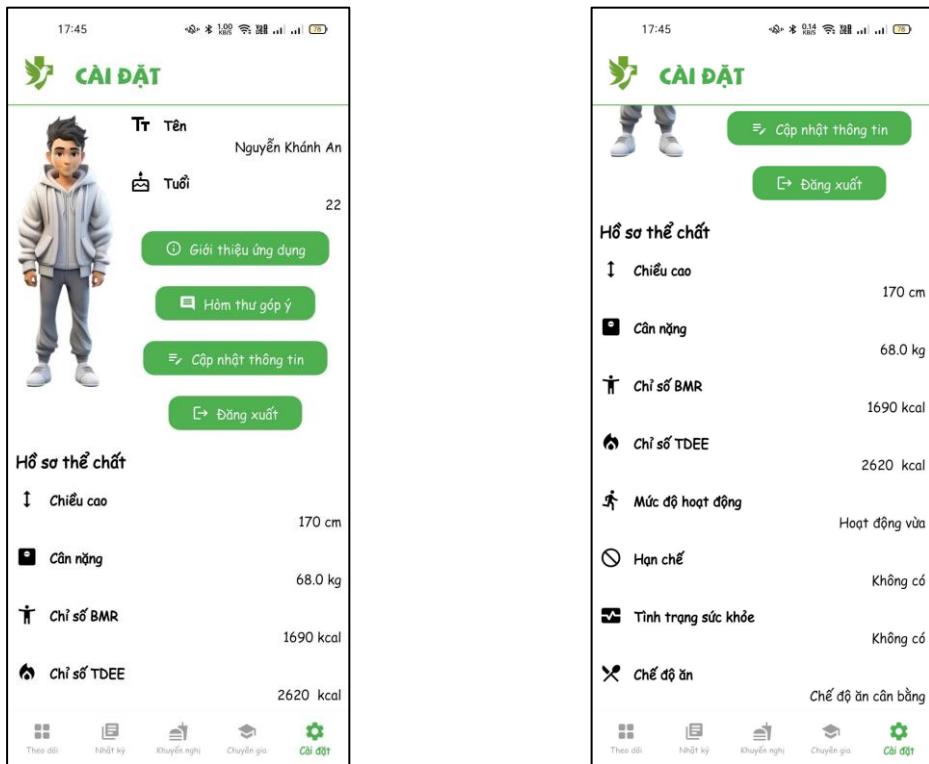
Ảnh 3.5. Giao diện module quản lý hồ sơ thể chất (2)



Ảnh 3.6. Giao diện module quản lý hồ sơ thể chất (3)

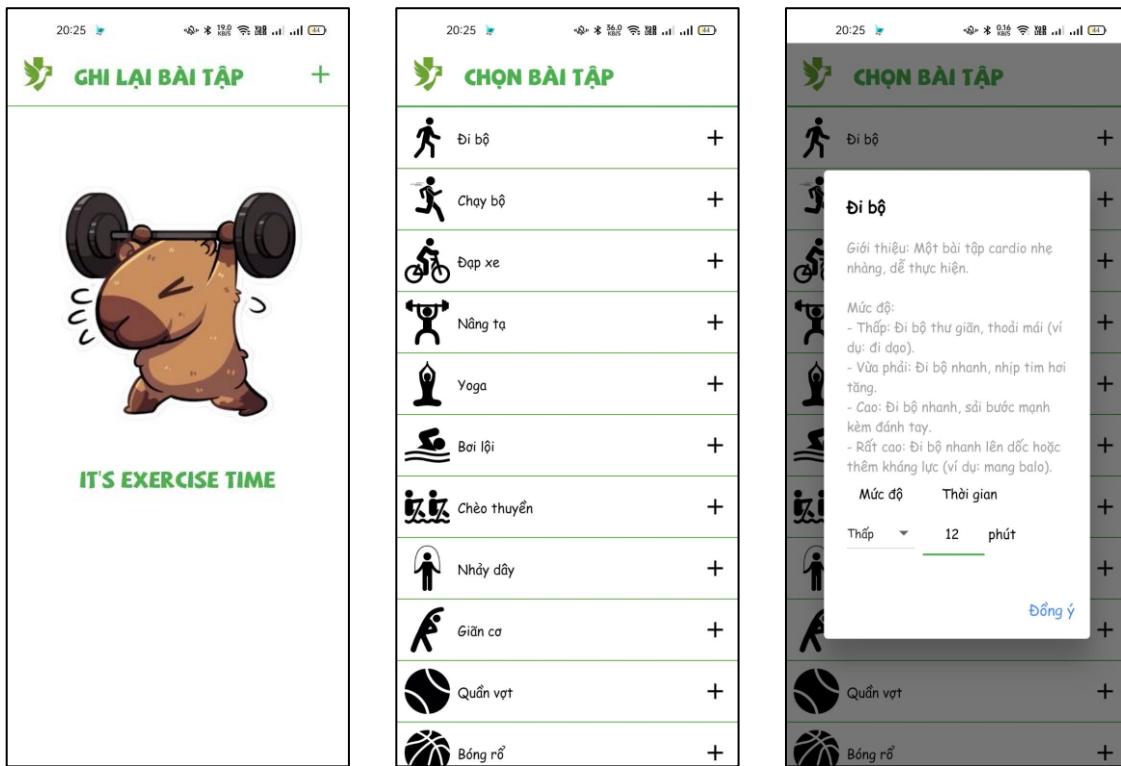


Ảnh 3.7. Giao diện module quản lý hồ sơ thể chất (4)

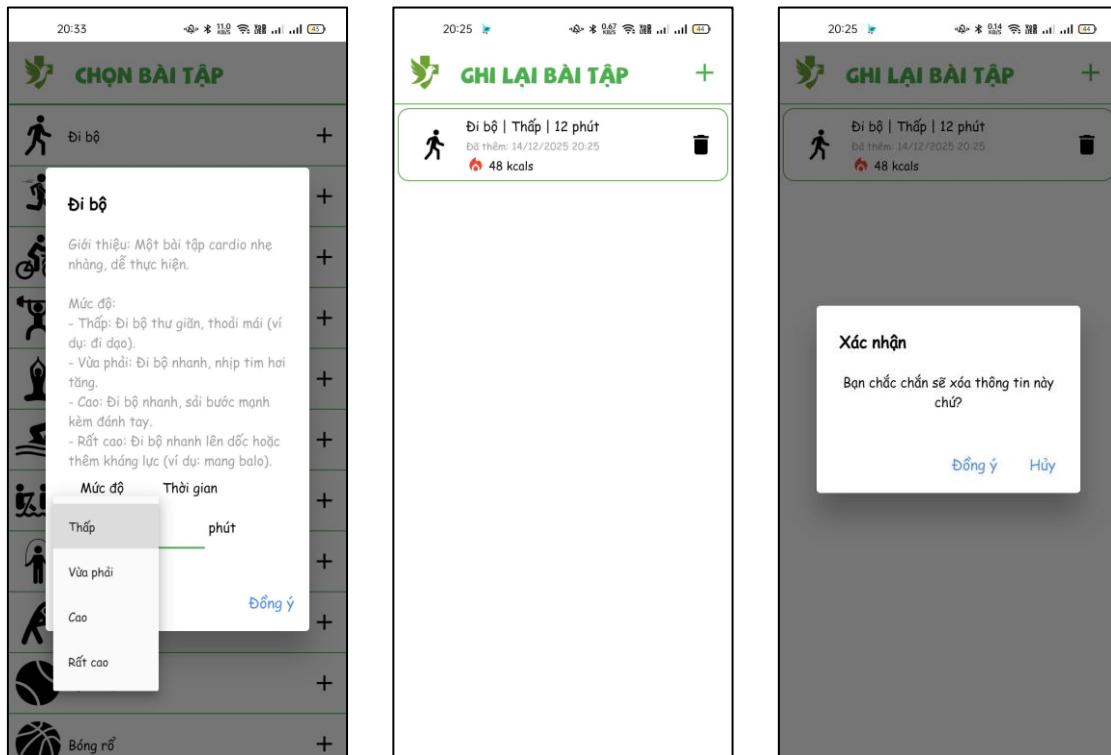


Ảnh 3.8. Giao diện module quản lý hồ sơ thể chất (5)

3.3.4. Giao diện module quản lý lịch trình tập thể dục

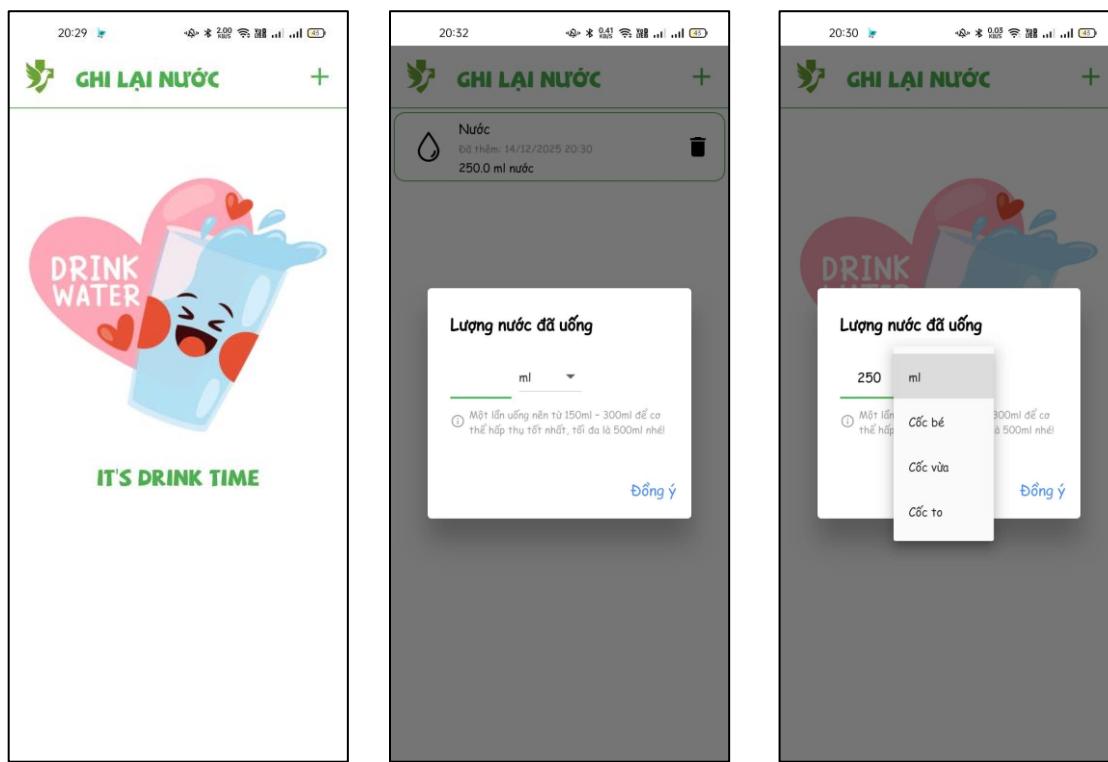


Ảnh 3.9. Giao diện module quản lý lịch trình tập thể dục (1)

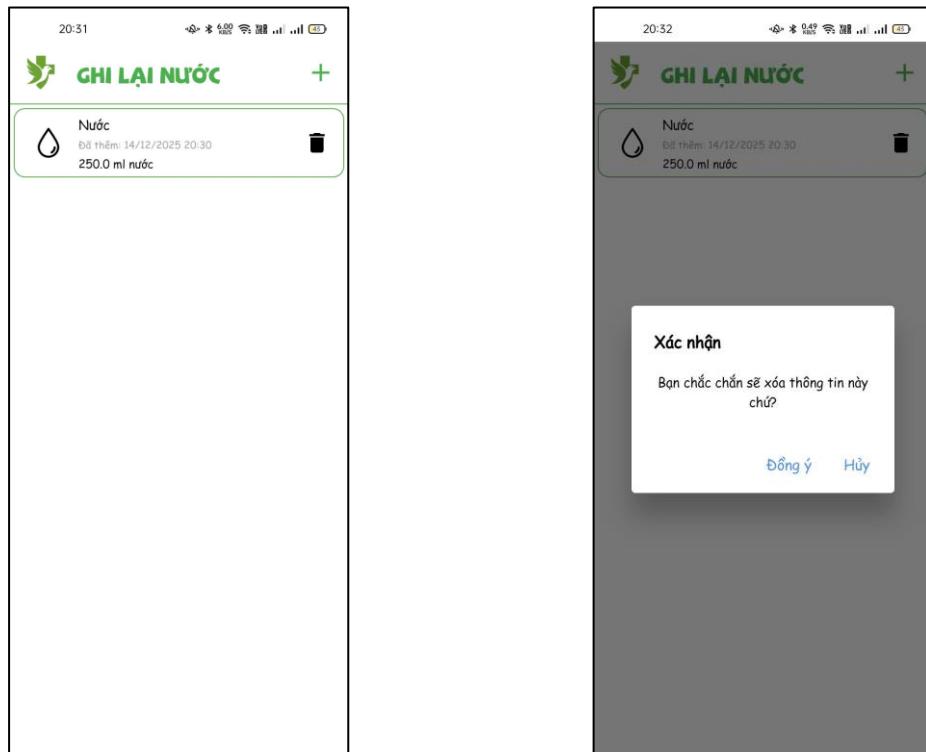


Ảnh 3.10. Giao diện module quản lý lịch trình tập thể dục (2)

3.3.5. Giao diện module quản lý lượng nước uống

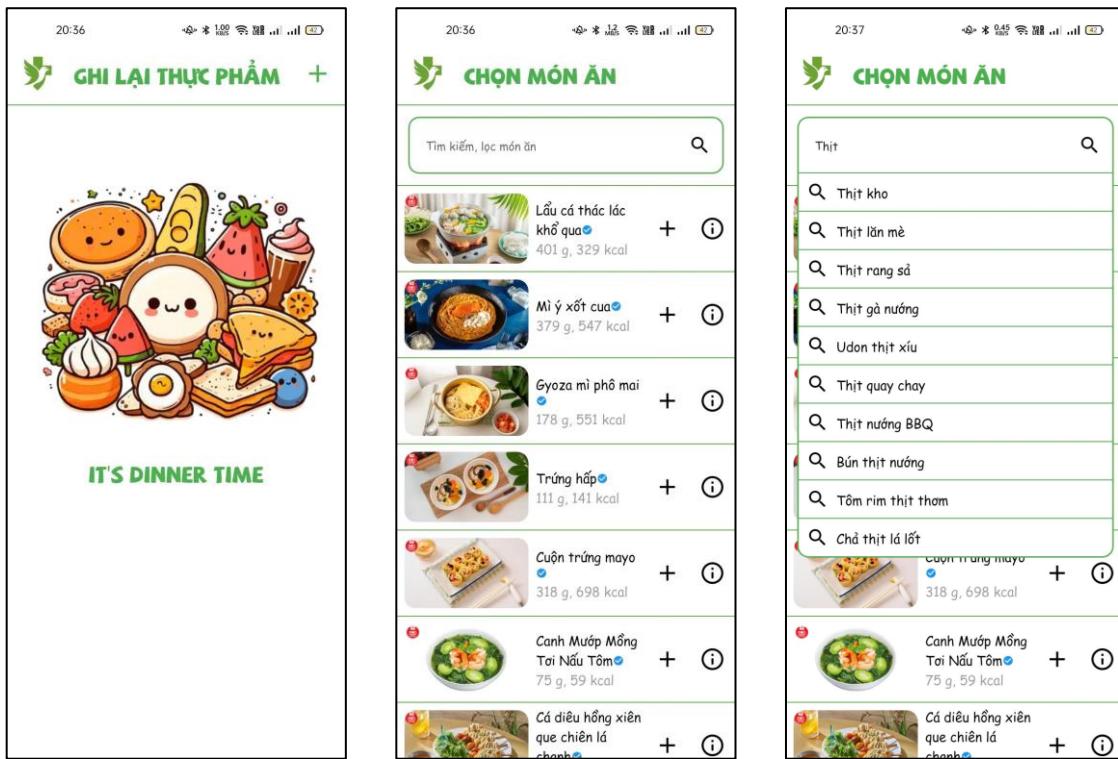


Ảnh 3.11. Giao diện module quản lý lượng nước uống (1)

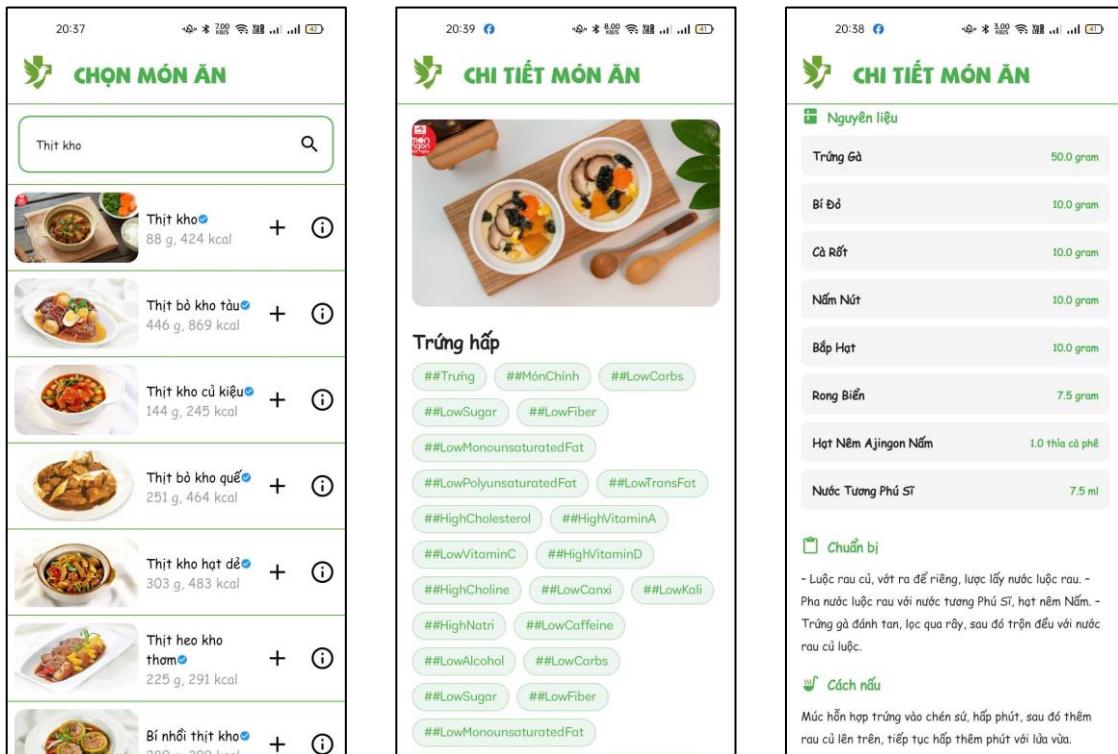


Ảnh 3.12. Giao diện module quản lý lượng nước uống (2)

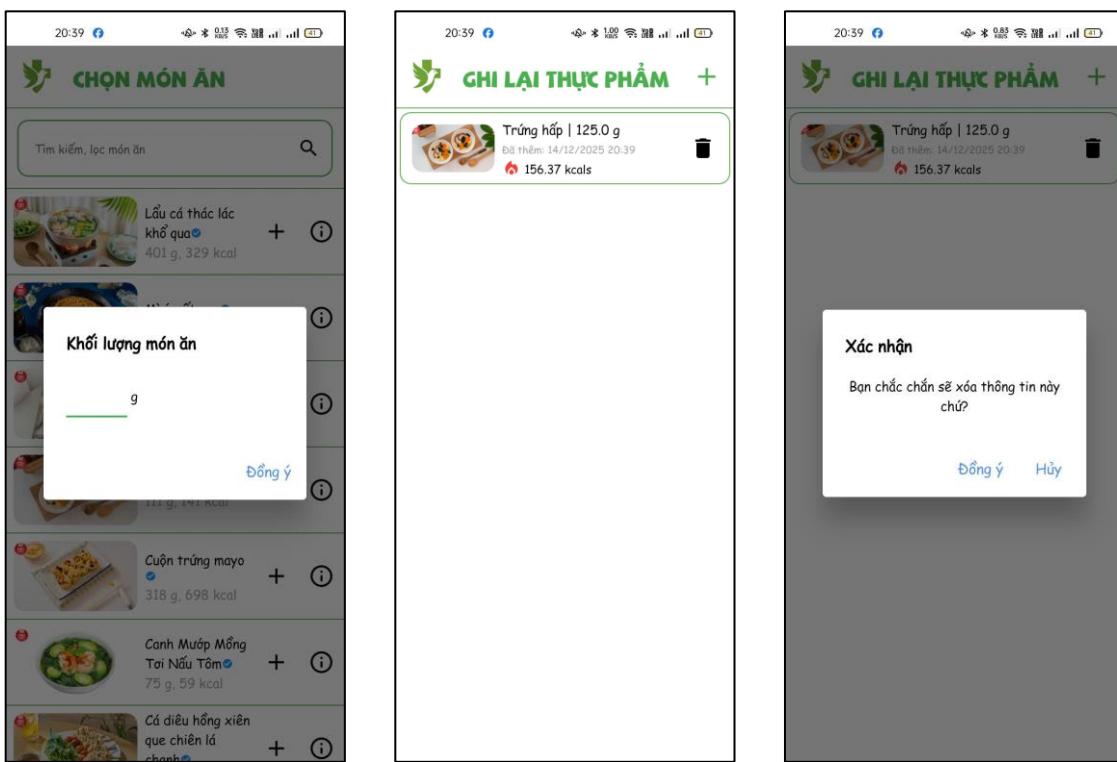
3.3.6. Giao diện module quản lý các món ăn từng bữa trong ngày



Ảnh 3.13. Giao diện module quản lý các món ăn từng bữa trong ngày (1)

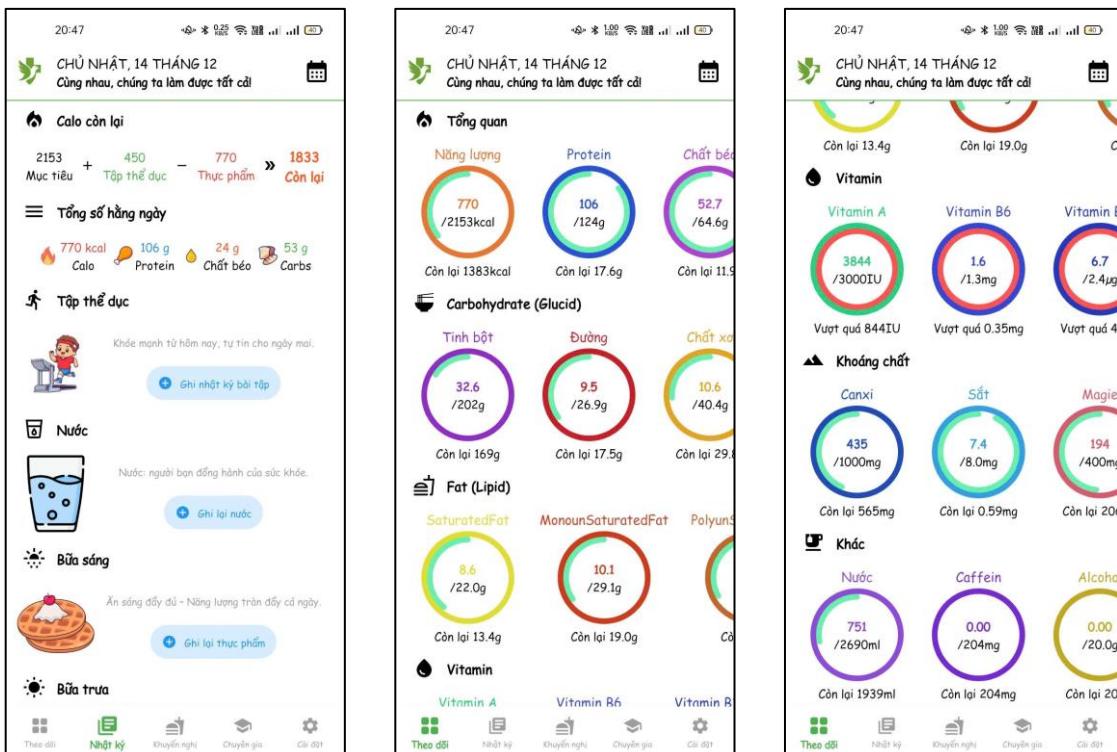


Ảnh 3.14. Giao diện module quản lý các món ăn từng bữa trong ngày (2)



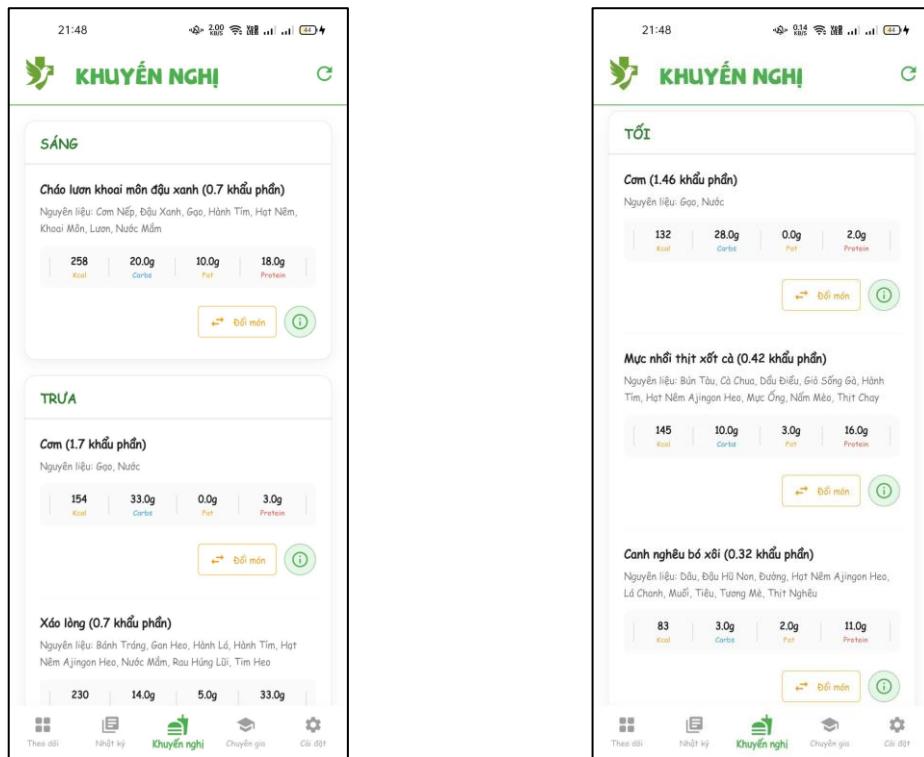
Ảnh 3.15. Giao diện module quản lý các món ăn từng bữa trong ngày (3)

3.3.7. Giao diện module thống kê dinh dưỡng trong ngày

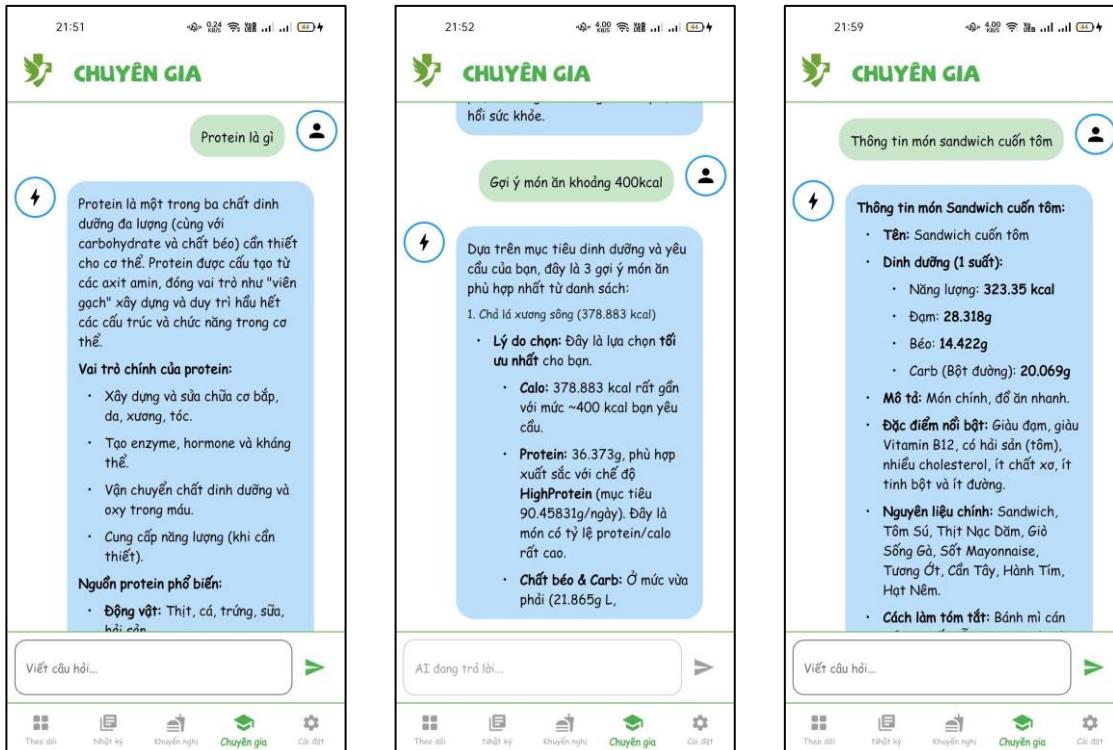


Ảnh 3.16. Giao diện module thống kê dinh dưỡng trong ngày

3.3.8. Giao diện module hỏi đáp với chuyên gia AI

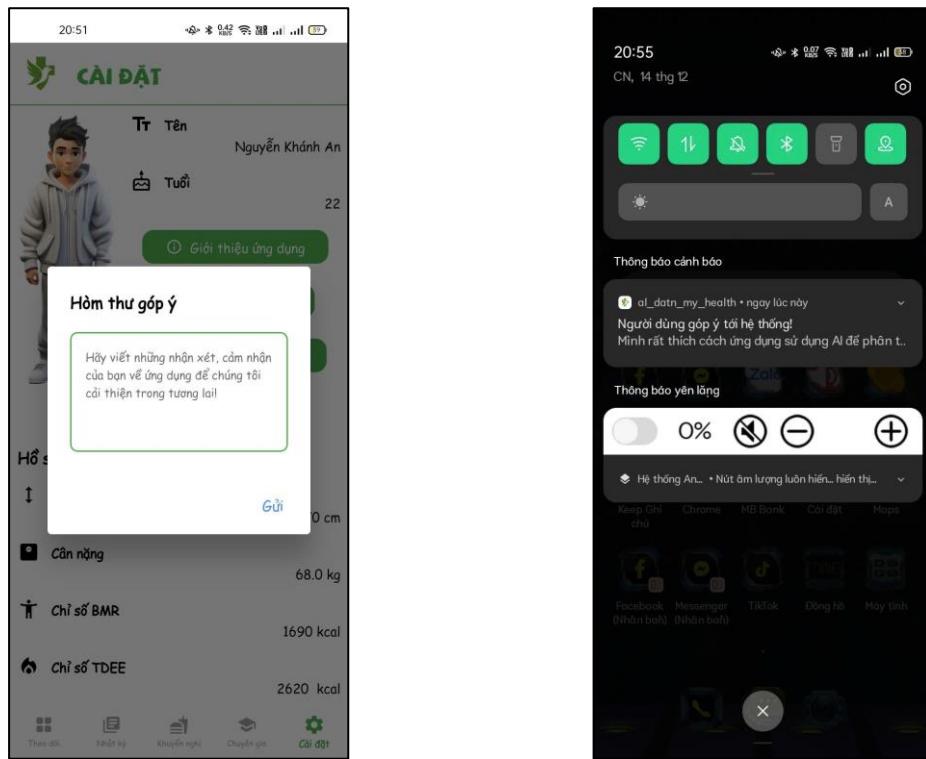


Ảnh 3.17. Giao diện module hỏi đáp với chuyên gia AI (1)



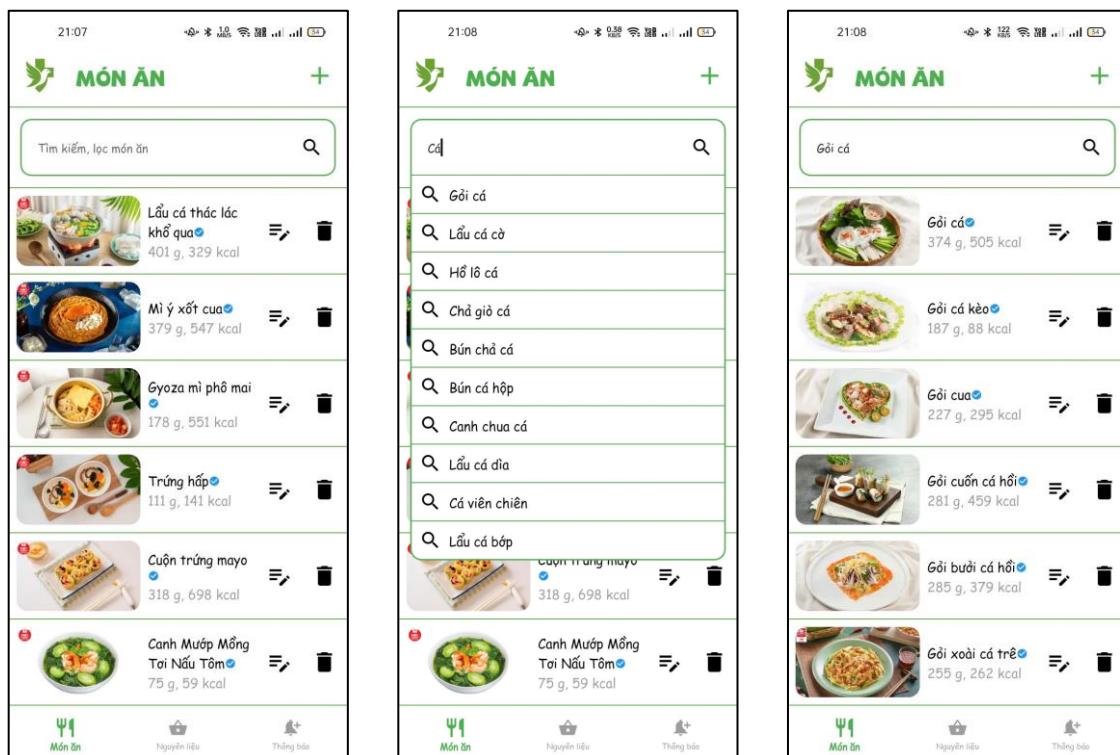
Ảnh 3.18. Giao diện module hỏi đáp với chuyên gia AI (2)

3.3.9. Giao diện module gửi góp ý cho quản trị viên

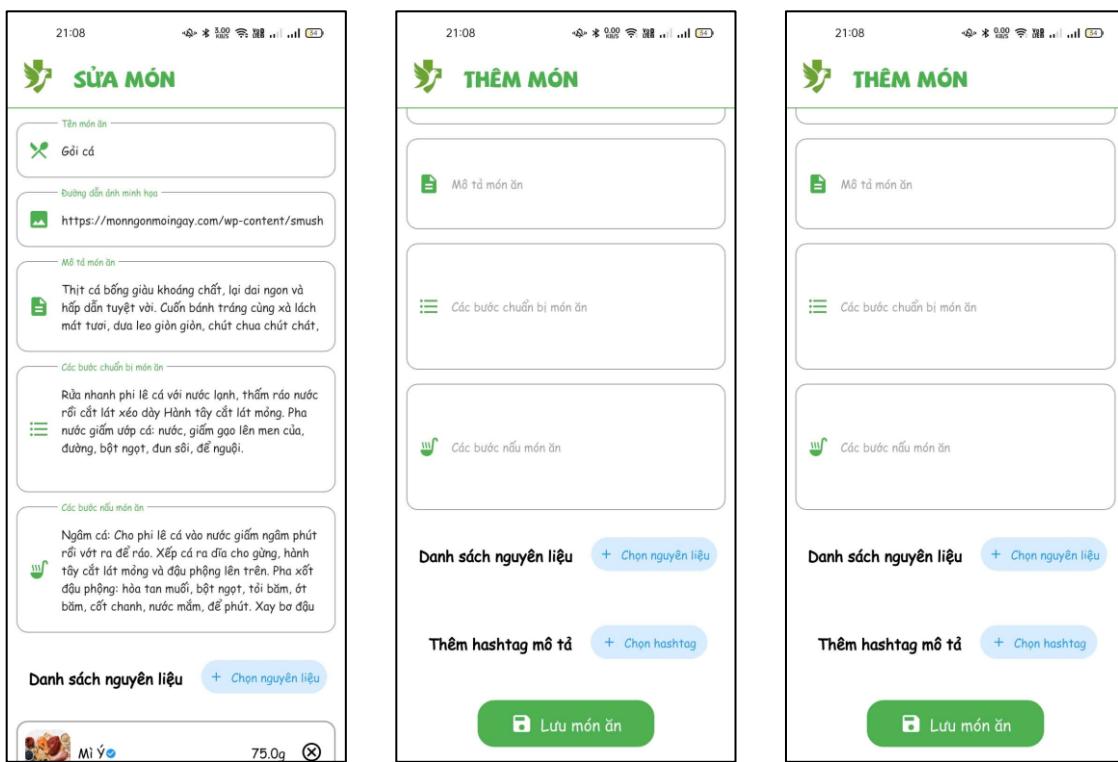


Ảnh 3.19. Giao diện module gửi góp ý cho quản trị viên

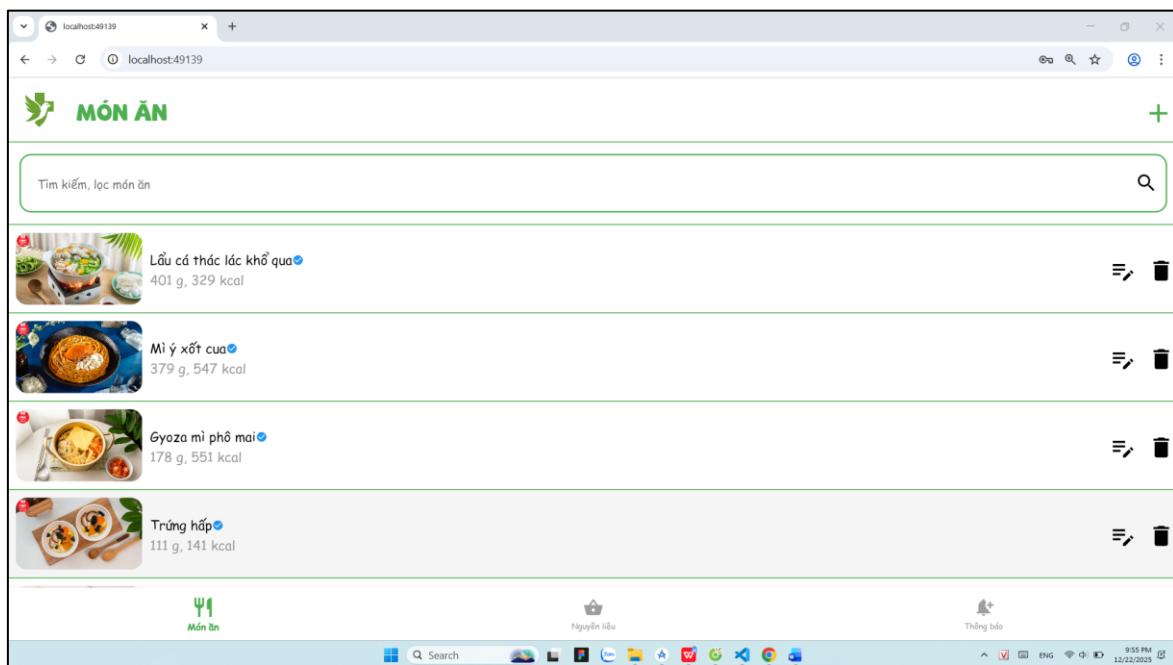
3.3.10. Giao diện module quản lý các món ăn hiển thị cho người dùng



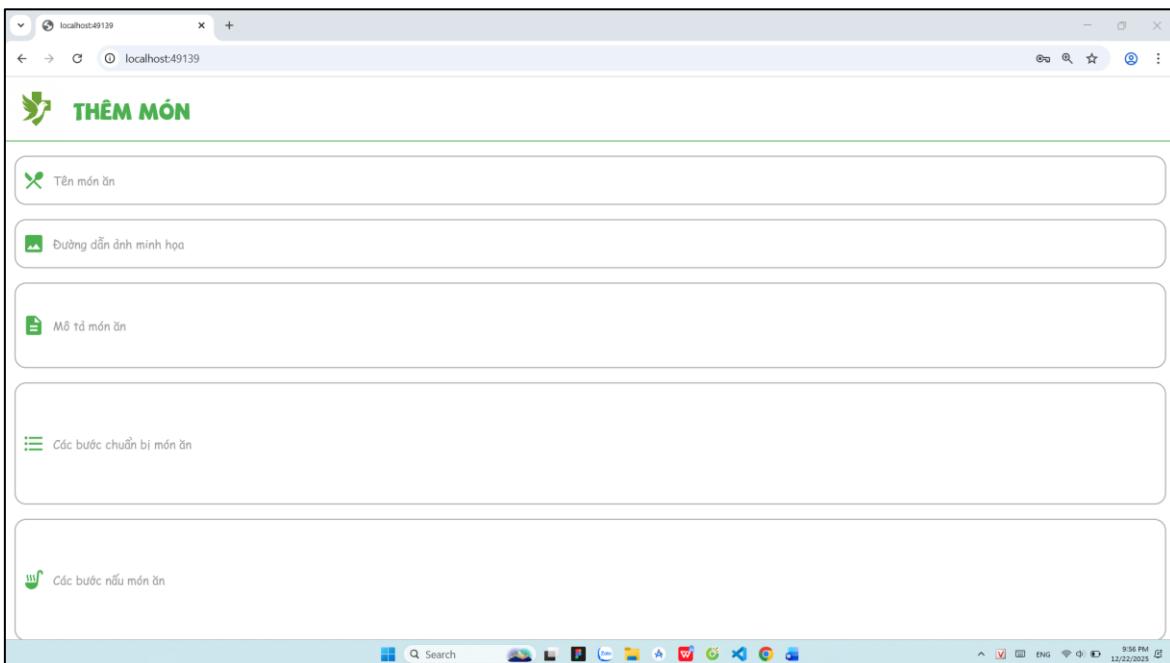
Ảnh 3.20. Giao diện module quản lý các món ăn hiển thị cho người dùng bản mobile (1)



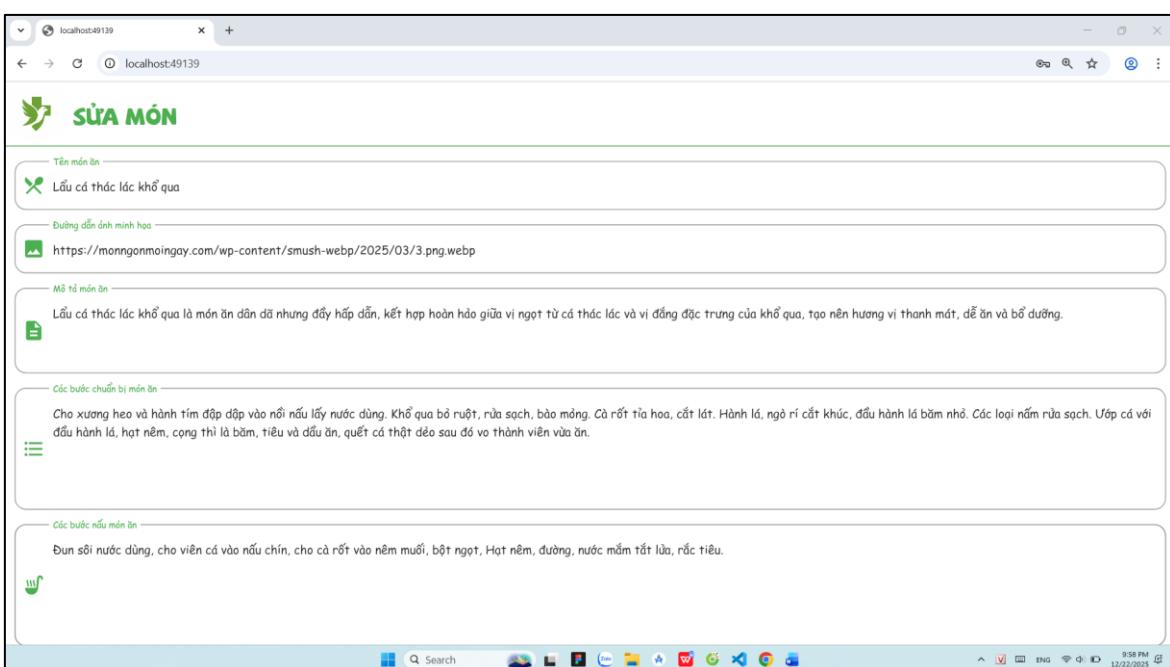
Ảnh 3.21. Giao diện module quản lý các món ăn hiển thị cho người dùng bản mobile (2)



Ảnh 3.22. Giao diện module quản lý các món ăn hiển thị cho người dùng bản web (1)

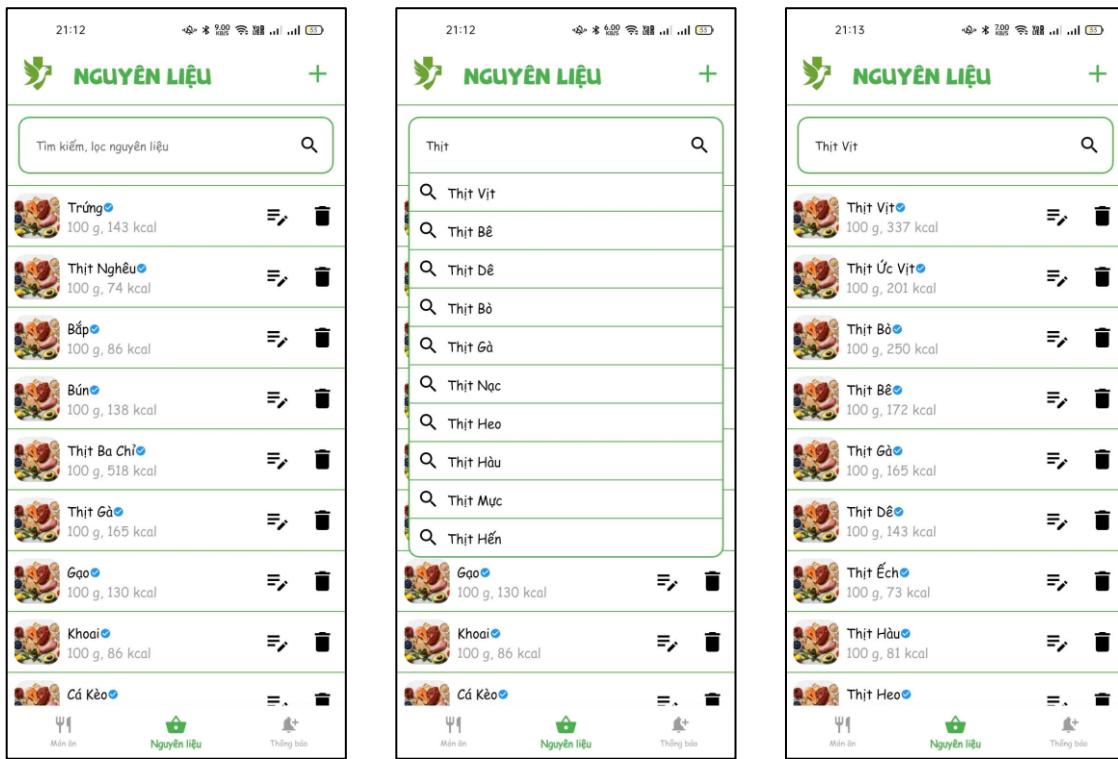


Ảnh 3.23. Giao diện module quản lý các món ăn hiển thị cho người dùng bản web (2)

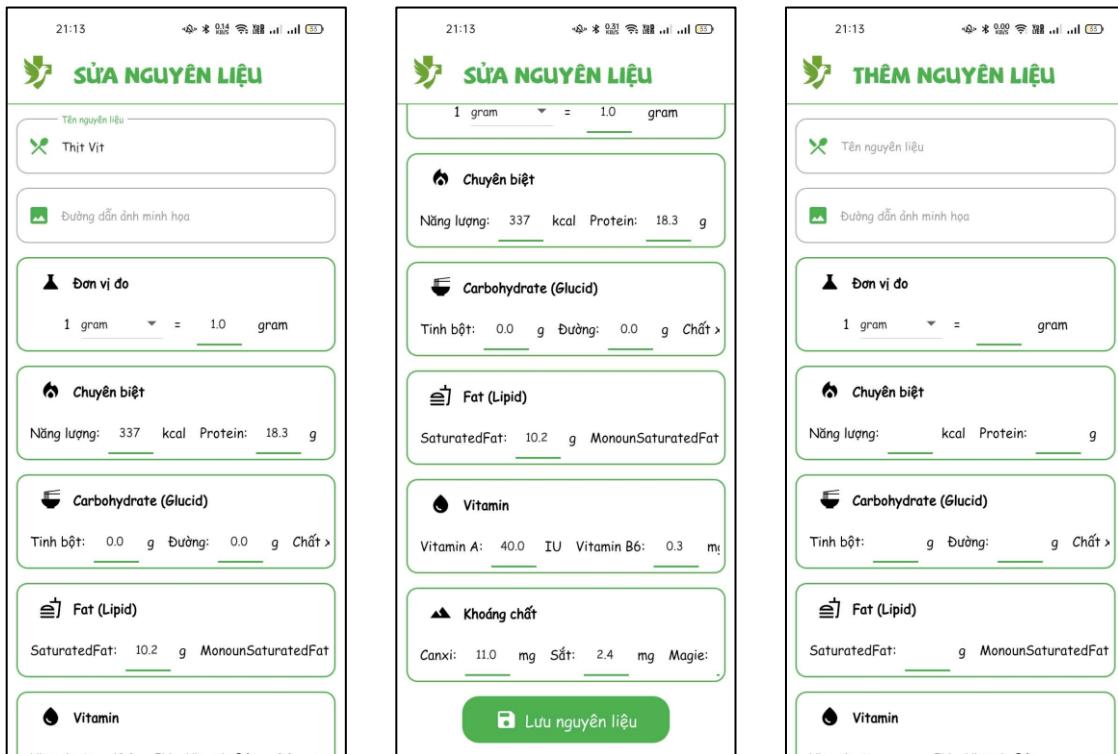


Ảnh 3.24. Giao diện module quản lý các món ăn hiển thị cho người dùng bản web (3)

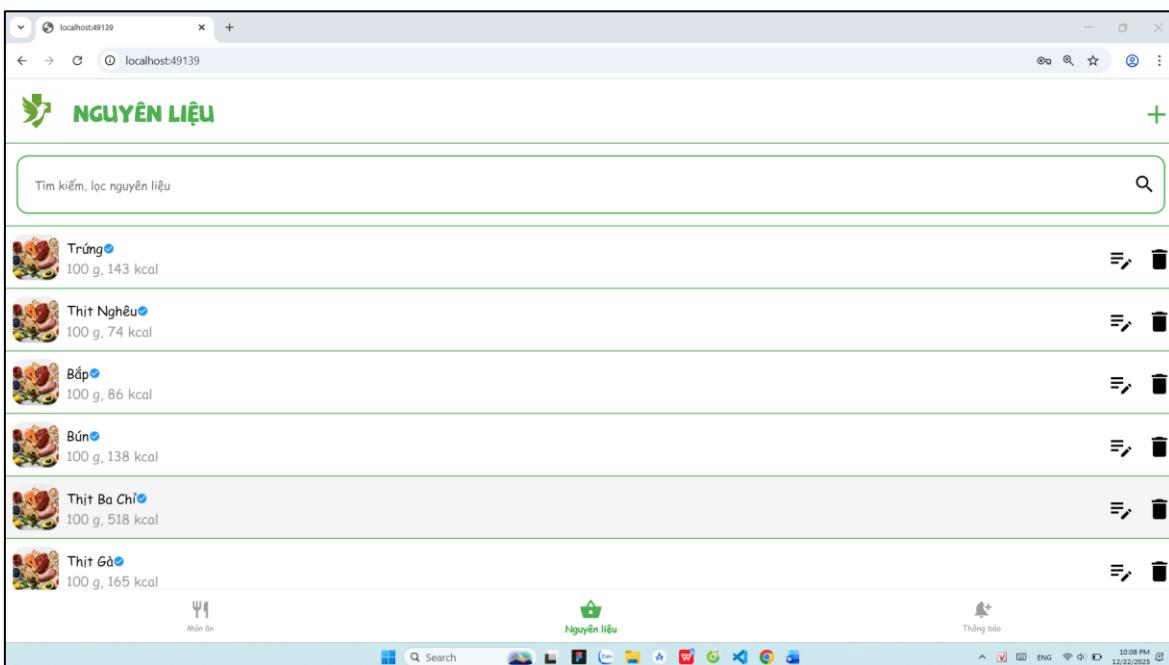
3.3.11. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn



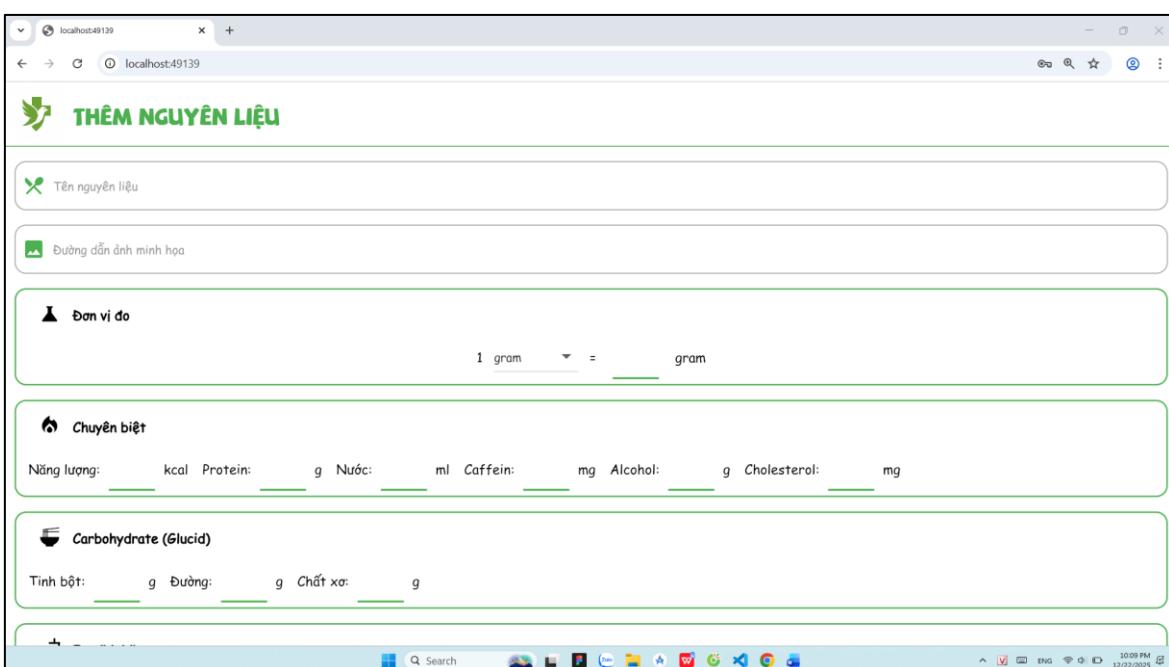
Ảnh 3.25. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản mobile (1)



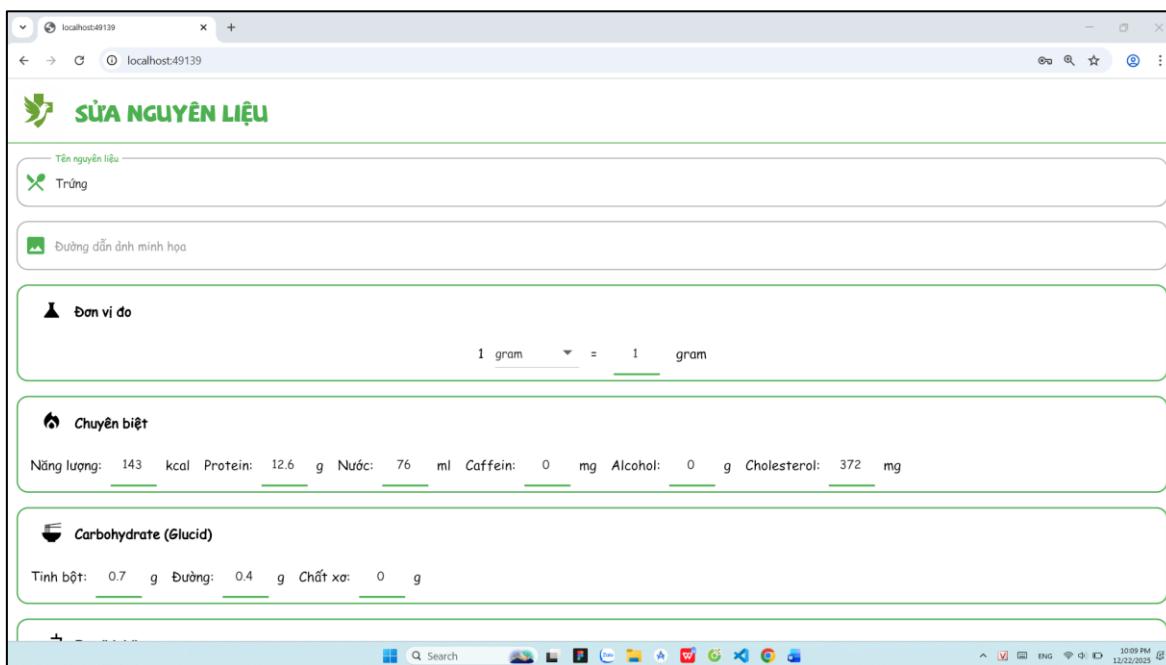
Ảnh 3.26. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản mobile (2)



Ảnh 3.27. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản web (1)

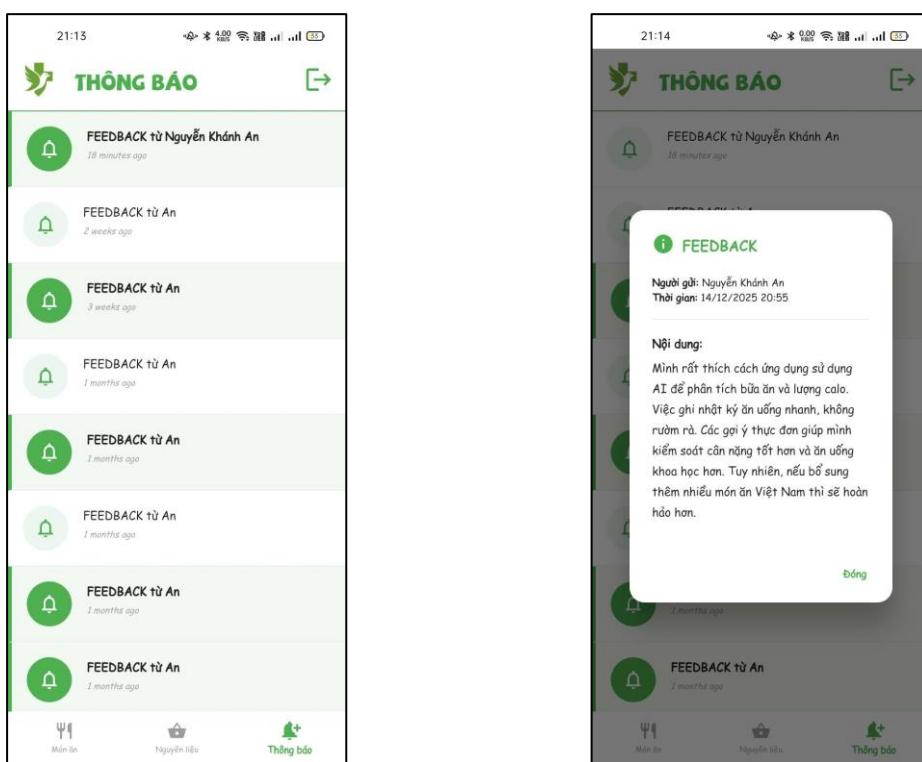


Ảnh 3.28. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản web (2)



Ảnh 3.29. Giao diện module quản lý các nguyên liệu sẽ dùng cho món ăn bản web (3)

3.3.12. Giao diện module nhận thông báo góp ý của người dùng



Ảnh 3.30. Giao diện module nhận thông báo góp ý của người dùng

3.4. Đánh giá hệ thống

3.4.1. Đánh giá mô hình truy xuất

3.4.1.1. Mô tả phương pháp

Mô hình truy xuất và gợi ý món ăn trong hệ thống được xây dựng dựa trên cơ chế tìm kiếm vector, trong đó thông tin món ăn được chuyển đổi thành vector embedding bằng mô hình *gte-multilingual-base* và lưu trữ trong cơ sở dữ liệu Elasticsearch. Khi người dùng gửi truy vấn, hệ thống thực hiện so khớp vector kết hợp với các điều kiện lọc (dinh dưỡng, nguyên liệu, nhãn) để trả về danh sách các món ăn phù hợp nhất.

Để đánh giá chất lượng mô hình truy xuất, đề tài sử dụng tập truy vấn kiểm thử bao gồm nhiều câu hỏi mô phỏng nhu cầu thực tế người dùng, ví dụ:

- Gợi ý các món ăn có chứa rau, giàu protein, ít béo.
- Gợi ý các món ăn ít calo, phù hợp giảm cân.
- Gợi ý món ăn healthy cho bữa tối.

Với mỗi truy vấn, danh sách kết quả trả về được gán nhãn mức độ liên quan và sử dụng để tính toán các chỉ số đánh giá.

3.4.1.2. Chỉ số đánh giá sử dụng

Để đánh giá chất lượng của mô hình truy xuất và gợi ý món ăn, đề tài sử dụng hai chỉ số đánh giá phổ biến trong các hệ thống tìm kiếm và gợi ý, bao gồm Precision@k và nDCG@k. Hai chỉ số này hỗ trợ cho nhau trong việc phản ánh cả độ chính xác và chất lượng sắp xếp của kết quả truy xuất.

a) Precision@k

Precision@k [16] đo lường tỉ lệ các kết quả phù hợp trong k kết quả đầu tiên mà hệ thống trả về. Chỉ số này cho biết mức độ chính xác của hệ thống trong việc loại bỏ các kết quả không liên quan ra khỏi danh sách gợi ý.

Công thức tính được xác định như sau:

$$\text{Precision}@k = \frac{\text{Số kết quả phù hợp trong top } k}{k}$$

Trong đó:

- k là số lượng kết quả được xét.
- Một kết quả được xem là phù hợp nếu món ăn thỏa mãn các điều kiện truy vấn.

b) nDCG@k

nDCG@k [16] là chỉ số đánh giá chất lượng xếp hạng kết quả, trong đó mức độ phù hợp của mỗi kết quả được gán trọng số dựa trên vị trí xuất hiện trong danh sách trả về. Các kết quả xuất hiện ở vị trí đầu sẽ có ảnh hưởng lớn hơn đến giá trị nDCG@k.

Trước hết, DCG@k được tính theo công thức:

$$DCG@k = \sum_{i=1}^k \frac{rel_i}{\log_2(i+1)}$$

Trong đó:

- rel_i là mức độ liên quan của kết quả ở vị trí thứ i .
- i là vị trí của kết quả trong danh sách.

Sau đó, nDCG@k được chuẩn hóa bằng cách chia DCG@k cho IDCG@k (giá trị DCG@k lý tưởng khi kết quả được sắp xếp theo mức độ liên quan giảm dần):

$$nDCG@k = \frac{DCG@k}{IDCG@k}$$

Giá trị của nDCG@k nằm trong khoảng từ 0 đến 1, trong đó giá trị càng gần 1 thể hiện chất lượng xếp hạng càng tốt.

3.4.1.3. Kết quả đánh giá

Dữ liệu đánh giá được xây dựng từ các truy vấn mô phỏng nhu cầu thực tế của người dùng, kết hợp với kết quả truy xuất từ hệ thống. Các kết quả được gán nhãn thủ công để đảm bảo đúng chất lượng truy xuất. Các chỉ số đánh giá được tính toán thể hiện ở các bảng sau:

Query ID	Mô tả truy vấn	Precision@10	nDCG@10
1	Gợi ý các món ăn có chứa rau, giàu protein, ít béo	1	0.9658
2	Các món ăn ít calo, phù hợp giảm cân	0.8	0.8987
3	Món ăn cho người tập gym	0.8	0.7448
...

Bảng 3.1. Kết quả đánh giá của mô hình theo từng query

Chi số	Giá trị trung bình
Precision@10	0.8492
nDCG@10	0.8111

Bảng 3.2. Kết quả đánh giá trung bình của mô hình

3.4.1.4. Nhận xét

Kết quả đánh giá cho thấy hệ thống gợi ý món ăn đạt hiệu quả tốt với Precision@10 trung bình 0.8492, chứng tỏ phần lớn các món ăn trong top-10 đều phù hợp với truy vấn người dùng. Chỉ số nDCG@10 trung bình 0.8111 cho thấy thứ tự sắp xếp kết quả tương đối hợp lý, các món ăn liên quan cao thường được ưu tiên ở vị trí đầu. Nhìn chung, mô hình truy xuất đáp ứng tốt yêu cầu đề ra và phù hợp cho ứng dụng thực tế.

3.4.2. Đánh giá hiệu năng hệ thống

Hiệu năng hệ thống được đánh giá thông qua nhiều kịch bản sử dụng thực tế của ứng dụng gợi ý bữa ăn và chatbot dinh dưỡng. Mỗi kịch bản tương ứng với một luồng xử lý khác nhau, bao gồm truy xuất dữ liệu, sinh nội dung bằng mô hình ngôn ngữ lớn và tối ưu khẩu phần ăn. Thời gian xử lý được đo nhằm đánh giá khả năng đáp ứng của hệ thống trong điều kiện sử dụng thông thường.

Các kịch bản đánh giá gồm:

- Kịch bản 1 – Sinh thực đơn trong ngày: Hệ thống thực hiện truy vấn cho 3 bữa sáng, trưa, tối bằng Elasticsearch kết hợp Self Query Retriever, sau đó sử dụng mô hình Deepseek LLM để lựa chọn món ăn và áp dụng thuật toán SLSQP nhằm tối ưu khẩu phần dinh dưỡng.
- Kịch bản 2 – Đổi món / Tìm món ăn tương đồng: Hệ thống truy xuất các món ăn có độ tương đồng so với món được chọn bằng Elasticsearch kết hợp Self Query Retriever, sau đó tối ưu khẩu phần dinh dưỡng và sử dụng Deepseek LLM để lựa chọn phương án thay thế.
- Kịch bản 3 – Chatbot dinh dưỡng: Hệ thống xử lý các câu hỏi tư vấn theo các chủ đề và đưa ra câu trả lời tùy vào luồng xử lý của từng chủ đề.

Thời gian phản hồi của từng kịch bản thể hiện qua các bảng sau:

Kịch bản	Các bước xử lý chính	Thời gian trung bình
1	Truy vấn Elasticsearch + LLM lựa thực đơn + tối ưu SLSQP	15-20s
2	Truy vấn Elasticsearch + tối ưu SLSQP + LLM lựa món	10-12s

Bảng 3.3. Đánh giá hiệu năng chức năng gợi ý, thay đổi món

Topic	Tên topic	Các bước xử lý chính	Thời gian trung bình
1	Gợi ý thực đơn	Xác định topic + ngữ cảnh + Các bước kịch bản 1	20-30s
2	Gợi ý món ăn	Xác định topic + ngữ cảnh + truy vấn Elasticsearch + LLM lựa món	12-15s
3	Tìm thông tin món ăn	Xác định topic + truy vấn Elasticsearch	7-10s
4	Hỏi thông tin ứng dụng	Xác định topic + truy vấn Elasticsearch	7-10s
5	Hỏi đáp thông thường	Xác định topic + LLM trả lời	2-3s

Bảng 3.4. Đánh giá hiệu năng chức năng chatbot

Kết quả trên cho thấy thời gian xử lý của hệ thống phụ thuộc vào mức độ phức tạp của từng kịch bản. Các tác vụ đơn giản như truy vấn hoặc tư vấn chatbot có thời gian phản hồi ngắn, trong khi các kịch bản phức tạp như sinh thực đơn hoặc đổi món, gợi ý món cần nhiều bước xử lý AI và tối ưu toán học nên có thời gian phản hồi dài hơn. Tuy nhiên, các kịch bản này đều phù hợp với mục đích sử dụng thực tế của ứng dụng, đặc biệt là các chức năng được thực hiện với tần suất thấp như sinh thực đơn theo ngày.

3.5. Tổng kết chương

Chương 3 đã trình bày kiến trúc xây dựng hệ thống và triển khai hệ thống ứng dụng quản lý dinh dưỡng tích hợp chatbot AI theo thiết kế đã đề xuất. Nội dung chương tập trung vào việc cài đặt các chức năng chính, tích hợp mô hình AI, xử lý dữ liệu dinh dưỡng và triển khai giao diện người dùng. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, đáp ứng các yêu cầu đặt ra và là cơ sở để đánh giá, hoàn thiện hệ thống trong các nghiên cứu và phát triển tiếp theo.

Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

Ứng dụng quản lý dinh dưỡng tích hợp chatbot AI đã hoàn thành các mục tiêu cơ bản đề ra:

- Về mặt lý thuyết: Nhóm đã nghiên cứu và áp dụng thành công các kiến thức về dinh dưỡng (BMI, TDEE, chế độ ăn) kết hợp với các công nghệ tiên tiến như LLM và kỹ thuật RAG để tối ưu hóa khả năng tư vấn của Chatbot.
- Về mặt sản phẩm: Ứng dụng MyHealth đã được hoàn thiện với đầy đủ các tính năng thiết yếu: Quản lý hồ sơ sức khỏe và tính toán calo mục tiêu, Tra cứu và ghi lại nhật ký ăn uống với kho dữ liệu món ăn Việt Nam, tích hợp Chatbot AI hỗ trợ giải đáp thắc mắc và tư vấn thực đơn, giao diện thân thiện, dễ sử dụng cho người dùng phổ thông.

Tuy nhiên, hệ thống vẫn còn một số điểm hạn chế:

- Dữ liệu: Kho dữ liệu món ăn tuy đa dạng nhưng chưa thể bao phủ hết toàn bộ các món ăn địa phương và các cách chế biến khác nhau.
- Hiệu năng AI: Tốc độ phản hồi của Chatbot đôi khi còn phụ thuộc vào đường truyền mạng và thời gian truy xuất dữ liệu từ hệ thống RAG.
- Tính năng nhập liệu: Người dùng vẫn phải nhập tay hoặc tìm kiếm món ăn, chưa có tính năng tự động nhận diện qua hình ảnh.

4.2. Hướng phát triển

Để nâng cao chất lượng ứng dụng trong tương lai và đưa vào sử dụng thực tế, ứng dụng có thể được phát triển thêm theo các hướng sau:

- Tích hợp AI nhận diện hình ảnh: Cho phép chụp ảnh món ăn để tự động tính toán lượng calo và dinh dưỡng.
- Kết nối thiết bị thông minh: Đóng bộ dữ liệu từ đồng hồ thông minh để theo dõi vận động chính xác hơn.
- Mở rộng cộng đồng: Xây dựng tính năng mạng xã hội để người dùng chia sẻ thực đơn và quá trình luyện tập.
- Tối ưu hóa Chatbot: Cải thiện tốc độ và độ chính xác của AI thông qua cải thiện mô hình embedding, tối ưu prompt, luồng xử lý hoạt động chức năng hệ thống.

4.3. Tổng kết chương

Chương này đã tổng hợp lại những kết quả mà ứng dụng quản lý dinh dưỡng tích hợp chatbot AI đã đạt được, đồng thời chỉ ra những hạn chế còn tồn tại và định hướng phát triển trong tương lai. Ứng dụng không chỉ đáp ứng các yêu cầu cơ bản về quản lý hồ sơ sức khỏe và theo dõi dinh dưỡng mà còn tích hợp chatbot AI hỗ trợ tư

vấn và gợi ý thực đơn, góp phần nâng cao trải nghiệm và hiệu quả sử dụng cho người dùng. Tuy nhiên, hệ thống vẫn còn một số điểm cần cải thiện như mở rộng dữ liệu món ăn, tối ưu hiệu năng và độ chính xác của chatbot AI, cũng như bổ sung các tính năng tự động hóa trong quá trình nhập liệu. Trong tương lai, với các định hướng phát triển rõ ràng, ứng dụng sẽ tiếp tục được hoàn thiện, đáp ứng tốt hơn nhu cầu thực tiễn và mang lại giá trị thiết thực cho người dùng.

TÀI LIỆU THAM KHẢO

- [1] PGS.TS. Phạm Duy Tường, Dinh dưỡng và an toàn thực phẩm, Hà Nội: Nhà xuất bản giáo dục Việt Nam, 2012.
- [2] Thomas, Diana M.; Gonzalez, M. Cristina; Pereira, Andrea Z.; Redman, Leanne M.; Heymsfield, Steven B, "Time to Correctly Predict the Amount of Weight Loss with Dieting," *Journal of the Academy of Nutrition and Dietetics*, 2014.
- [3] Bộ Y Tế, Nhu cầu dinh dưỡng khuyến nghị Việt Nam, Hà Nội: Nhà xuất bản Y Học, 2016.
- [4] Administration, U.S. Food and Drug, "The Lows and Highs of Percent Daily Value on the Nutrition Facts Label," 2024. [Online]. Available: <https://www.fda.gov/food/nutrition-facts-label/lows-and-highs-percent-daily-value-nutrition-facts-label>.
- [5] U.S. Department of Agriculture, Agricultural Research Service, "FoodData Central – About Us," 2019. [Online]. Available: <https://fdc.nal.usda.gov/about-us>.
- [6] ThS. Bùi Phú Khuyên, Giáo Trình CMP177 - Lập Trình Trên Thiết Bị Di Động Cơ Bản và Nâng Cao, TP.HCM: ĐẠI HỌC CÔNG NGHỆ TP.HCM, 2024 .
- [7] Nguyễn Hoàng Anh, Đặng Ngọc Hùng, Bài giảng Lập trình với Python, Hà Nội: Học viện công nghệ Bưu chính Viễn thông, 2022.
- [8] Nguyễn Văn Thịịnh, "Tìm hiểu hệ quản trị cơ sở dữ liệu PostgreSQL," Viblo, 21/8/2021. [Online]. Available: <https://viblo.asia/p/tim-hieu-he-quan-tri-co-so-du-lieu-postgresql-m68Z0eLdlkG>.
- [9] Nickname Hướng Nội, "RAG (Retrieval-Augmented Generation) là gì? Giải thích dễ hiểu cho Developer," 200Lab, 11 Mar 2025. [Online]. Available: <https://200lab.io/blog/rag-la-gi>.
- [10] Dieter Kraft, A Software Package for Sequential Quadratic Programming, Germany, 1988.
- [11] Anugrah Jo Joshy và John T. Hwang, PySLSQP: A transparent Python package for the SLSQP optimization algorithm modernized with utilities for visualization and post-processing, University of California San Diego, 2024.
- [12] Amazon Web Services, "What is a Large Language Model (LLM)?," AWS, [Online]. Available: [What is a Large Language Model \(LLM\)?](#)

- [13] Josh Asres, "Self-querying retrievers with Elasticsearch: Unleashing your metadata," Elastic, 13 2 2025. [Online]. Available: <https://www.elastic.co/search-labs/blog/self-querying-retrievers>.
- [14] Amazon Web Services, "What is LangChain?," AWS, [Online]. Available: <https://aws.amazon.com/vi/what-is/langchain/>.
- [15] AbeGauld, "Tìm hiểu về LangGraph," Viblo, 15 3 2025. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-langgraph-0gdJzzKjJz5>.
- [16] Nimrod Kramer, "DeepSeek: Everything you need to know about this new LLM in one place," Daily Dev, 22 1 2025. [Online]. Available: <https://daily.dev/blog/deepseek-everything-you-need-to-know-about-this-new-llm-in-one-place>.
- [17] Khiem, "Elasticsearch là gì? Elasticsearch hoạt động như thế nào?," 200Lab, 12 10 2023. [Online]. Available: <https://200lab.io/blog/elasticsearch-la-gi>.
- [18] Wikipedia contributors, "Evaluation measures (information retrieval)," Wikipedia, The Free Encyclopedia, [Online]. Available: [https://en.wikipedia.org/wiki/Evaluation_measures_\(information_retrieval\)](https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)).