

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN

**ĐỒ ÁN TỐT NGHIỆP**  
NGÀNH: CÔNG NGHỆ THÔNG TIN  
CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

ĐỀ TÀI:

**ỨNG DỤNG THUẬT TOÁN DI TRUYỀN ĐỂ  
XÂY DỰNG HỆ THỐNG TẠO  
THỰC ĐƠN CÁ NHÂN**

Người hướng dẫn: PGS.TS. NGUYỄN THANH BÌNH  
Sinh viên thực hiện: BÙI HỮU TRỌNG  
Số thẻ sinh viên: 102210191  
Lớp: 21TCLC\_DT2

Đà Nẵng, 6/2025

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN

## ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

ỨNG DỤNG THUẬT TOÁN DI TRUYỀN ĐỂ  
XÂY DỰNG HỆ THỐNG TẠO  
THỰC ĐƠN CÁ NHÂN

Người hướng dẫn: PGS.TS NGUYỄN THANH BÌNH

Sinh viên thực hiện: Bùi Hữu Trọng

Số thẻ sinh viên: 102210191

Lớp: 21TCLC\_DT2

Đà Nẵng, 6/2025

# TÓM TẮT

Tên đề tài: Ứng dụng thuật toán di truyền để xây dựng hệ thống tạo thực đơn cá nhân

Sinh viên thực hiện: Bùi Hữu Trọng

Số thẻ SV: 102210191

Lớp: 21TCLC-DT2

Đồ án "Ứng dụng thuật toán Di truyền để xây dựng hệ thống tạo thực đơn cá nhân" được thực hiện nhằm phát triển một giải pháp thông minh hỗ trợ người dùng cá nhân hóa chế độ ăn uống dựa trên nhu cầu sức khỏe và sở thích. Mục tiêu chính là thiết kế và triển khai hệ thống sử dụng GA để tối ưu hóa thực đơn dựa trên dữ liệu dinh dưỡng (calo, protein, carb, fat) và các hạn chế cá nhân, kết hợp với các công nghệ tiên tiến như Node.js, FastAPI, và MongoDB.

Phương pháp nghiên cứu bao gồm phân tích yêu cầu, thiết kế hệ thống, triển khai backend và frontend, cùng với kiểm thử và đánh giá hiệu quả để đảm bảo tính khả thi. Đồ án tập trung vào việc tích hợp thuật toán GA để xử lý bài toán tối ưu phức tạp, hướng tới một giao diện thân thiện và dễ sử dụng.

Đồ án mang lại ý nghĩa khoa học và thực tiễn, đóng góp vào việc áp dụng trí tuệ nhân tạo trong lĩnh vực dinh dưỡng, hỗ trợ người dùng xây dựng lối sống lành mạnh, đồng thời đặt nền tảng cho các cải tiến và mở rộng trong tương lai.

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Bùi Hữu Trọng

Số thẻ sinh viên: 102210191

Lớp: 21TCLC-DT2

Khoa: Công nghệ thông tin

Ngành: Hệ thống thông tin

- Tên đề tài đồ án:* Ứng dụng thuật toán di truyền để xây dựng hệ thống tạo thực đơn cá nhân
- Đề tài thuộc diện:*  Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
- Các số liệu và dữ liệu ban đầu:* Không có
- Nội dung các phần thuyết minh và tính toán:*
  - Chương 1: Tổng quan về đề tài.
  - Chương 2: Cơ sở lý thuyết.
  - Chương 3: Phân tích và thiết kế hệ thống.
  - Chương 4: Xây dựng và triển khai hệ thống.
  - Chương 5: Kiểm thử và đánh giá hệ thống.
  - Kết luận và hướng phát triển.
- Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*
- Họ tên người hướng dẫn:* PGS.TS. Nguyễn Thanh Bình
- Ngày giao nhiệm vụ đồ án:* 27/03/2025.
- Ngày hoàn thành đồ án:* 10/05/2025

Đà Nẵng, ngày 19 tháng 05 năm 2025

Trưởng Bộ môn .....

Người hướng dẫn

## LỜI NÓI ĐẦU

Trong bối cảnh xã hội hiện đại, khi nhịp sống ngày càng nhanh và áp lực công việc gia tăng, nhu cầu về một chế độ dinh dưỡng lành mạnh, cân đối và dễ dàng quản lý ngày càng trở nên cấp thiết. Tuy nhiên, nhiều người gặp khó khăn trong việc xây dựng thực đơn phù hợp với nhu cầu cá nhân, đặc biệt là những người quan tâm đến việc tăng cường protein, giảm carbs và fat để duy trì sức khỏe hoặc hỗ trợ mục tiêu tập luyện. Với sự phát triển của công nghệ thông tin, các ứng dụng hỗ trợ dinh dưỡng đã và đang trở thành công cụ hữu ích, giúp người dùng tối ưu hóa chế độ ăn uống một cách khoa học và hiệu quả. Từ thực tế đó, đề án "Ứng dụng thuật toán di truyền để xây dựng hệ thống tạo thực đơn cá nhân" được thực hiện nhằm cung cấp một giải pháp công nghệ, cho phép người dùng dễ dàng theo dõi, quản lý và tùy chỉnh thực đơn dựa trên sở thích, nhu cầu dinh dưỡng và giới hạn cá nhân.

Đề án này không chỉ là cơ hội để áp dụng kiến thức đã học vào thực tiễn mà còn là bước khởi đầu trong việc khám phá tiềm năng của công nghệ trong lĩnh vực sức khỏe và dinh dưỡng. Toàn bộ quá trình thực hiện từ nghiên cứu, thiết kế đến triển khai đều được thực hiện với sự tận tâm và nỗ lực cao nhất, nhằm mang lại một sản phẩm có giá trị thực tiễn.

Em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Nguyễn Thanh Bình, người đã tận tình hướng dẫn, truyền đạt kinh nghiệm và động viên trong suốt quá trình thực hiện đề án.

## CAM ĐOAN

Tôi – Bùi Hữu Trọng – xin cam đoan rằng toàn bộ nội dung và kết quả trong đề án "Ứng dụng thuật toán di truyền để xây dựng hệ thống tạo thực đơn cá nhân" là do tôi thực hiện dưới sự hướng dẫn của Thầy Nguyễn Thanh Bình. Các số liệu, kết quả, và tài liệu tham khảo được trình bày trung thực, rõ ràng. Tôi không sao chép từ bất kỳ nguồn nào mà không trích dẫn. Nếu có bất kỳ gian lận hay sao chép nào, tôi xin chịu hoàn toàn trách nhiệm trước nhà trường và quy định hiện hành.

Trân trọng!

*Đà Nẵng, ngày 19 tháng 06 năm 2025*

Sinh viên thực hiện

# MỤC LỤC

TÓM TẮT.....	6
LỜI NÓI ĐẦU.....	i
CAM ĐOAN.....	ii
DANH SÁCH HÌNH VẼ, BẢNG.....	vi
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	1
1.1. Tổng quan.....	1
1.2. Mục đích và ý nghĩa đề tài.....	1
1.3. Mục tiêu đề tài.....	2
1.4. Phạm vi và đối tượng nghiên cứu.....	2
1.5. Công nghệ sử dụng.....	3
1.6. Kết quả dự kiến.....	4
1.7. Cấu trúc nội dung của đề án tốt nghiệp.....	4
1.8. Kết luận chương.....	5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	6
2.1. Thuật toán di truyền Genetic Algorithm (GA).....	6
2.1.1. Khái niệm và nguyên lý.....	6
2.1.2 Các thành phần cơ bản của GA.....	7
2.1.3. Các toán tử di truyền.....	8
2.2. Thuật toán GA trong đồ án.....	10
2.3. Ưu điểm và hạn chế.....	14
2.3.1. Ưu điểm.....	14
2.3.2. Hạn chế.....	15
2.3. Công nghệ sử dụng.....	16
2.4. Kết luận chương.....	17
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	19
3.1. Phân tích yêu cầu.....	19
3.1.1. Yêu cầu chức năng.....	19
3.1.2. Yêu cầu phi chức năng.....	21
3.2. Quy trình hoạt động chi tiết của GA trong đồ án.....	23
3.2.1. Quy trình hoạt động.....	23
3.2.2. Cơ chế tối ưu hóa.....	24

3.3. Các tác nhân chính của hệ thống .....	25
3.3.1. Tác nhân: User .....	25
3.3.2. Tác nhân: Admin.....	26
3.4. Sơ đồ ca sử dụng.....	27
3.4.1. Sơ đồ usecase tổng quát .....	27
3.4.2. Quản lý thông tin cá nhân .....	27
3.4.3. Quản lý thực đơn.....	27
3.4.4. Theo dõi tiến độ .....	28
3.4.5. Quản lý món ăn .....	28
3.4.6. Quản lý tài khoản .....	29
3.5. Đặc tả usecase .....	29
3.5.1 Đặc tả usecase quản lí thông tin cá nhân .....	29
3.5.2. Đặc tả usecase quản lý thực đơn .....	30
3.5.3. Đặc tả usecase quản lý tiến độ .....	32
3.5.4. Đặc tả usecase quản lý món ăn .....	32
3.5.5. Đặc tả usecase quản lý tài khoản .....	34
3.6. Sơ đồ hoạt động .....	35
3.6.1. Sơ đồ hoạt động của chức năng đăng ký, đăng nhập, quên mật khẩu .....	35
3.6.2. Sơ đồ hoạt động của hệ thống client .....	38
3.6.4. Sơ đồ hoạt động của hệ thống admin .....	41
3.7. Sơ đồ tuần tự.....	46
3.7.1. Sơ đồ tuần tự cho chức năng đăng nhập, đăng ký .....	46
3.7.2. Sơ đồ tuần tự cho hệ thống client.....	48
3.7.3. Sơ đồ tuần tự cho hệ thống admin.....	51
3.8. Cơ sở dữ liệu.....	53
3.8.1. Collection User.....	53
3.8.2. Collection Food.....	55
3.8.3. Collection SaveMenu .....	56
3.8.4. Collection Notifications .....	56
3.8.5. Collection ComplianceHistory .....	57
3.9. Kết luận chương.....	58
CHƯƠNG 4: TRIỂN KHAI VÀ KẾT QUẢ.....	59
4.1. Môi trường phát triển.....	59
4.1.1. Môi trường lưu trữ và quản lý nguồn.....	59

4.1.2. Môi trường phát triển dự án .....	59
4.2. Công nghệ sử dụng .....	59
4.2.1. Frontend .....	59
4.2.2. Backend.....	60
4.2.3. Công nghệ triển khai hệ thống .....	60
4.3. Giao diện hệ thống.....	61
4.3.1. Giao diện trang chủ .....	61
4.3.2. Giao diện đăng nhập, đăng ký.....	61
4.3.3. Giao diện tổng quan .....	62
4.4.4. Giao diện hồ sơ dinh dưỡng.....	63
4.4.5. Giao diện thực đơn.....	63
4.4.6. Giao diện tiến độ.....	64
4.4.7. Giao diện tư vấn AI.....	64
4.4.8. Giao diện mua sắm.....	65
4.4.9. Giao diện hồ sơ cá nhân .....	65
4.4.10. Giao diện tổng quan Admin .....	66
4.4.11. Giao diện admin quản lý người dùng.....	66
4.4.12. Giao diện admin quản lý món ăn .....	67
4.4. Kiểm thử và đánh giá hệ thống.....	67
4.4.1. Kết quả kiểm thử .....	67
4.5. Hướng phát triển tương lai.....	70
KẾT LUẬN .....	72
TÀI LIỆU THAM KHẢO .....	73

## DANH SÁCH HÌNH VẼ, BẢNG

Hình 2.1: Sơ đồ tổng quan hoạt động GA .....	6
Hình 2.2: Sơ đồ luồng hoạt động của GA trong đồ án.....	11
Hình 3.1: Sơ đồ usecase tổng quát .....	27
Hình 3.2: Sơ đồ usecase quản lý thông tin cá nhân.....	27
Hình 3.3: Sơ đồ usecase quản lý thực đơn .....	27
Hình 3.4: Sơ đồ usecase theo dõi tiến độ .....	28
Hình 3.5: Sơ đồ usecase quản lý món ăn.....	28
Hình 3.6: Sơ đồ usecase quản lý tài khoản.....	29
Hình 3.7: Sơ đồ hoạt động chức năng đăng ký .....	35
Hình 3.8: Sơ đồ hoạt động chức năng đăng nhập.....	36
Hình 3.9: Sơ đồ hoạt động chức năng quên mật khẩu.....	37
Hình 3.10: Sơ đồ hoạt động chức năng tạo thực đơn .....	38
Hình 3.11: Sơ đồ hoạt động chức năng đánh dấu hoàn thành.....	39
Hình 3.12: Sơ đồ hoạt động chức năng theo dõi tiến độ .....	39
Hình 3.13: Sơ đồ hoạt động chức năng tư vấn AI.....	40
Hình 3.14: Sơ đồ hoạt động chức năng mua sắm.....	40
Hình 3.15: Sơ đồ hoạt động chức năng chỉnh sửa thông tin cá nhân .....	41
Hình 3.16: Sơ đồ hoạt động chức năng hiển thị DashBoard .....	41
Hình 3.17: Sơ đồ hoạt động chức năng xem thông tin tài khoản .....	42
Hình 3.18: Sơ đồ hoạt động chức năng sửa tài khoản.....	42
Hình 3.19: Sơ đồ hoạt động chức năng xem chi tiết món ăn .....	43
Hình 3.20: Sơ đồ hoạt động chức năng cập nhật món ăn.....	44
Hình 3.21: Sơ đồ hoạt động chức năng thêm món ăn .....	45
Hình 3.22: Sơ đồ hoạt động chức năng xóa món ăn.....	45
Hình 3.23: Sơ đồ tuần tự chức năng đăng ký .....	46
Hình 3.24: Sơ đồ tuần tự chức năng đăng nhập .....	47
Hình 3.25: Sơ đồ tuần tự chức năng tạo thực đơn.....	48
Hình 3.26: Sơ đồ tuần tự chức năng đánh dấu hoàn thành.....	49
Hình 3.27: Sơ đồ tuần tự chức năng theo dõi tiến độ.....	49

Hình 3.28: Sơ đồ tuần tự chức năng mua sắm.....	50
Hình 3.29: Sơ đồ tuần tự chức năng quản lý thông tin cá nhân .....	50
Hình 3.30: Sơ đồ tuần tự chức năng quản lý tài khoản .....	51
Hình 3.31: Sơ đồ tuần tự chức năng quản lý món ăn .....	52
Hình 3.32: Sơ đồ tuần tự chức năng xóa món ăn .....	53
Hình 4.1: Giao diện trang chủ .....	61
Hình 4.2: Giao diện đăng nhập.....	61
Hình 4.3: Giao diện đăng ký.....	62
Hình 4.4: Giao diện tổng quan .....	62
Hình 4.5: Giao diện hồ sơ dinh dưỡng .....	63
Hình 4.6: Giao diện thực đơn .....	63
Hình 4.7: Giao diện tiến độ .....	64
Hình 4.8: Giao diện tư vấn AI .....	64
Hình 4.9: Giao diện mua sắm .....	65
Hình 4.10: Giao diện hồ sơ cá nhân .....	65
Hình 4.11: Giao diện thực đơn đã lưu .....	66
Hình 4.12: Giao diện tổng quan Admin .....	66
Hình 4.13: Giao diện quản lý người dùng.....	66
Hình 4.14: Giao diện quản lý món ăn.....	67
Bảng 3.1: Tác nhân .....	26
Bảng 3.2: Đặc tả usecase quản lý thông tin cá nhân .....	30
Bảng 3.3: Đặc tả usecase quản lý thực đơn.....	31
Bảng 3.4: Đặc tả usecase quản lý tiến độ .....	32
Bảng 3.5: Đặc tả usecase quản lý món ăn .....	33
Bảng 3.6: Đặc tả usecase quản lý tài khoản .....	34
Bảng 3.7: Collection User .....	54
Bảng 3.8: Collection Food.....	55
Bảng 3.9: Collection SaveMenu.....	56
Bảng 3.10: Collection Notifications .....	57
Bảng 3.11: Collection ComplianceHistory .....	58

Bảng 4.1: Kết quả .....	68
-------------------------	----



## CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

### 1.1. Tổng quan

Trong bối cảnh xã hội hiện đại, nhu cầu về một chế độ dinh dưỡng lành mạnh ngày càng gia tăng, đặc biệt với những người quan tâm đến sức khỏe, tập luyện thể thao, hoặc có các hạn chế dinh dưỡng cụ thể. Theo thống kê từ Tổ chức Y tế Thế giới (WHO), tỷ lệ béo phì và các bệnh liên quan đến dinh dưỡng không lành mạnh đang gia tăng nhanh chóng, đòi hỏi các giải pháp hỗ trợ quản lý chế độ ăn uống hiệu quả. Tuy nhiên, nhiều người gặp khó khăn trong việc xây dựng thực đơn cá nhân hóa, phù hợp với nhu cầu dinh dưỡng như giàu protein, ít carbs và fat, đồng thời thuận tiện cho các bữa ăn trong ngày, bao gồm bữa sáng và bữa phụ.

Đề tài "Ứng dụng thuật toán di truyền để xây dựng hệ thống tạo thực đơn cá nhân" ra đời nhằm đáp ứng nhu cầu này, cung cấp một nền tảng công nghệ giúp người dùng dễ dàng quản lý thực đơn, theo dõi dinh dưỡng, và nhận gợi ý món ăn phù hợp. Ý nghĩa của đề tài không chỉ nằm ở việc ứng dụng công nghệ để giải quyết vấn đề thực tiễn mà còn góp phần nâng cao nhận thức cộng đồng về tầm quan trọng của dinh dưỡng khoa học.

### 1.2. Mục đích và ý nghĩa đề tài

#### Mục đích:

Trong bối cảnh hiện nay, nhu cầu về sức khỏe và vóc dáng ngày càng tăng, kéo theo sự quan tâm đến chế độ ăn uống cá nhân hóa. Tuy nhiên, việc tự lên kế hoạch thực đơn đáp ứng chính xác mục tiêu dinh dưỡng (protein, carbs, fat, calories) và phù hợp với sở thích, hạn chế của từng người là một thách thức lớn. Nhiều người gặp khó khăn trong việc đảm bảo thực đơn vừa đạt mục tiêu (ví dụ: 120g protein, 2000 kcal/ngày), vừa đáp ứng sở thích (thích món ít béo) và hạn chế (không ăn sữa), dẫn đến mất thời gian và không duy trì được chế độ ăn khoa học. Đề tài hướng đến xây dựng một hệ thống tạo thực đơn tự động nhằm giải quyết vấn đề này, giúp người dùng dễ dàng đạt được mục tiêu dinh dưỡng, tiết kiệm thời gian, và cải thiện sức khỏe, đồng thời đảm bảo thực đơn đa dạng, phù hợp với nhu cầu cá nhân.

## **Ý nghĩa:**

Ý nghĩa của đề tài nằm ở việc ứng dụng thuật toán Di truyền (GA) để giải quyết bài toán tối ưu hóa thực đơn cá nhân hóa. Hệ thống cung cấp giải pháp thực tế, hỗ trợ người dùng xây dựng chế độ ăn uống khoa học, giảm nguy cơ các bệnh mãn tính như béo phì hay tiểu đường, từ đó nâng cao chất lượng cuộc sống. Ngoài ra, đề tài mang lại giá trị xã hội bằng cách cá nhân hóa dinh dưỡng dựa trên nhu cầu và sở thích riêng, phù hợp với xu hướng sống lành mạnh hiện đại. Hệ thống còn có tiềm năng tích hợp với các tổ chức y tế hoặc ứng dụng thương mại, mở ra cơ hội nghiên cứu và phát triển trong tương lai. Điều này không chỉ khẳng định tính ứng dụng cao mà còn góp phần vào sự đổi mới trong lĩnh vực công nghệ sức khỏe.

### **1.3. Mục tiêu đề tài**

Đề tài hướng đến xây dựng một hệ thống tạo thực đơn tự động, hỗ trợ người dùng đạt được mục tiêu dinh dưỡng một cách chính xác và cá nhân hóa, bao gồm các chỉ số như protein, carbs, fat, và calories. Hệ thống đảm bảo thực đơn được tối ưu hóa để đạt mục tiêu dinh dưỡng của từng bữa ăn, đồng thời đáp ứng sở thích cá nhân và các hạn chế về chế độ ăn.

### **1.4. Phạm vi và đối tượng nghiên cứu**

Đối tượng nghiên cứu:

- Người dùng: Người có nhu cầu cá nhân hóa thực đơn để đạt mục tiêu dinh dưỡng, bao gồm người tập gym (tăng cơ), người giảm cân, hoặc người có chế độ ăn đặc biệt (ăn chay, không ăn sữa).
- Dữ liệu món ăn: Dữ liệu món ăn với thông tin dinh dưỡng (protein, carbs, fat, calories), thuộc tính (sở thích, hạn chế), và ngữ cảnh (bữa sáng/trưa/tối/phụ).

Phạm vi nghiên cứu:

- Nội dung: Tập trung vào các chức năng cốt lõi: quản lý thông tin người dùng, quản lý món ăn, tạo thực đơn tự động, xem trước và điều chỉnh thực đơn, lưu và xem lịch sử

- Không triển khai tính năng mở rộng như tích hợp cân thông minh, chatbot hỗ trợ, hay hỗ trợ chế độ ăn phức tạp (keto, paleo). Hệ thống chỉ tập trung vào người dùng cá nhân, không hỗ trợ tính năng cộng đồng.

## **1.5. Công nghệ sử dụng**

### **1.5.1. Backend**

Hệ thống backend được xây dựng với các công nghệ sau:

- Node.js:
  - Express.js: Framework để xây dựng RESTful API.
  - JWT (JSON Web Tokens): Thư viện xác thực và phân quyền.
  - Mongoose (tùy chọn): ORM để kết nối và quản lý MongoDB.
- Python:
  - FastAPI: Framework để xây dựng API hiệu suất cao.
  - NumPy/Pandas (tùy chọn): Thư viện hỗ trợ tính toán và xử lý dữ liệu cho GA.
  - Pydantic: Thư viện validation dữ liệu.

Hai nền tảng tích hợp qua API để xử lý các tác vụ.

### **1.5.2. Frontend**

Dự án sử dụng React.js làm framework chính cho phần frontend. Các công nghệ chính bao gồm:

- React.js với Hooks để quản lý state và lifecycle
- React Calendar cho việc hiển thị lịch thực đơn
- Các thư viện biểu đồ để hiển thị tiến độ dinh dưỡng
- CSS Modules cho việc quản lý styles

### **1.5.3. Database**

Dự án sử dụng MongoDB làm cơ sở dữ liệu chính:

- Lưu trữ thông tin người dùng, thực đơn, và tiến độ tuân thủ
- Schema-less database phù hợp với dữ liệu thực đơn linh hoạt
- Tối ưu cho việc truy xuất và cập nhật dữ liệu thường xuyên

- Hỗ trợ các truy vấn phức tạp và aggregation

## **1.6. Kết quả dự kiến**

Kết quả dự kiến của đề án bao gồm việc phát triển một hệ thống tạo thực đơn cá nhân hóa hoạt động hiệu quả, dựa trên thuật toán Di truyền (GA) để tối ưu hóa dinh dưỡng. Cụ thể:

- Xây dựng thành công hệ thống có khả năng tạo thực đơn đáp ứng mục tiêu dinh dưỡng (calo, protein, carb, fat) trong vòng 1-2 phút, với sai số dưới 10% so với yêu cầu người dùng.
- Thiết kế giao diện người dùng trực quan, cho phép nhập thông tin cá nhân (tuổi, cân nặng, sở thích) và hiển thị thực đơn dễ dàng, kèm theo các tính năng gợi ý món ăn và theo dõi tiến độ.
- Đưa ra báo cáo đánh giá hiệu quả, bao gồm độ chính xác dinh dưỡng, mức độ hài lòng của người dùng qua khảo sát (dự kiến đạt trên 80%), và thời gian xử lý nhanh chóng.
- Cung cấp nền tảng có thể mở rộng, sẵn sàng cho các cải tiến như tích hợp với thiết bị thông minh hoặc mở rộng dữ liệu món ăn trong tương lai.

## **1.7. Cấu trúc nội dung của đề án tốt nghiệp**

Đề án tốt nghiệp được tổ chức thành các chương và tiểu mục nhằm trình bày một cách có hệ thống quá trình nghiên cứu, thiết kế, triển khai, và đánh giá hệ thống tạo thực đơn cá nhân hóa dựa trên thuật toán Di truyền (GA). Cấu trúc nội dung bao gồm:

- **Chương 1: Tổng quan về đề tài**

Giới thiệu bối cảnh, mục đích, ý nghĩa, phạm vi, phương pháp nghiên cứu, kết quả dự kiến, và câu hỏi định hướng của đề án. Phần này đặt nền tảng cho toàn bộ dự án, xác định vai trò của GA trong tối ưu hóa dinh dưỡng.

- **Chương 2: Cơ sở lý thuyết**

Trình bày các khái niệm nền tảng về thuật toán Di truyền (GA), các nguyên tắc tối ưu hóa, cơ sở dữ liệu dinh dưỡng, và các công nghệ liên quan (Node.js, FastAPI, MongoDB), làm cơ sở cho thiết kế và triển khai hệ thống.

- **Chương 3: Phân tích và thiết kế hệ thống**

Bao gồm phân tích yêu cầu chức năng và phi chức năng, thiết kế kiến trúc hệ thống, thiết kế cơ sở dữ liệu (các collection như User, Food, SaveMenu, ComplianceHistory, Notifications), và thiết kế giao diện người dùng. Phần này cung cấp khung lý thuyết và bản thiết kế chi tiết cho triển khai.

- **Chương 4: Triển khai và kết quả**

Trình bày kết quả đạt được (thực đơn tối ưu, giao diện hoạt động), đánh giá hiệu quả hệ thống (độ chính xác dinh dưỡng, thời gian xử lý, phản hồi người dùng), cùng với các hạn chế và đề xuất phát triển trong tương lai như tích hợp thiết bị thông minh.

## **1.8. Kết luận chương**

Chương Tổng quan đã cung cấp cái nhìn tổng quan về đề tài "Ứng dụng thuật toán Di truyền để xây dựng hệ thống tạo thực đơn cá nhân", làm rõ cơ sở lý thuyết, bối cảnh, và ý nghĩa thực tiễn của việc phát triển hệ thống này. Trong bối cảnh nhu cầu về dinh dưỡng khoa học ngày càng gia tăng, đề tài giải quyết vấn đề xây dựng thực đơn cá nhân hóa, tối ưu hóa dinh dưỡng, và hỗ trợ người dùng theo dõi tiến độ, đặc biệt cho những đối tượng như người tập gym, giảm cân, hoặc có chế độ ăn đặc biệt. Mô tả bài toán đã xác định mục đích, mục tiêu, phạm vi, và đối tượng nghiên cứu, tập trung vào các chức năng cốt lõi như quản lý thông tin, tạo thực đơn tự động, và theo dõi tiến độ, đồng thời giới hạn không triển khai các tính năng mở rộng như tích hợp thiết bị thông minh hay hỗ trợ cộng đồng.

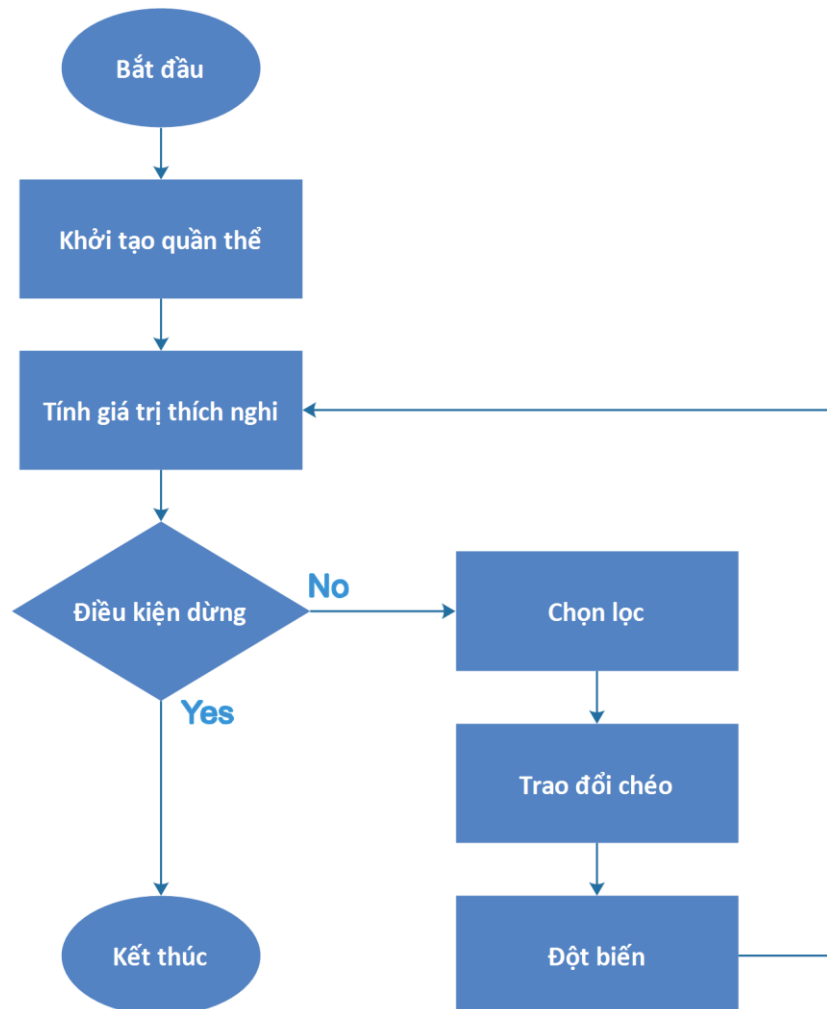
Chương này đã xây dựng cơ sở lý luận và định hướng cho các phần tiếp theo, bao gồm thiết kế hệ thống. Những nội dung này sẽ được phát triển sâu hơn để hiện thực hóa mục tiêu của đề tài, góp phần mang lại một giải pháp công nghệ thực tiễn và hiệu quả cho quản lý cá nhân hóa.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Thuật toán di truyền Genetic Algorithm (GA)

#### 2.1.1. Khái niệm và nguyên lý

Thuật toán Di truyền (GA) là một phương pháp tối ưu hóa dựa trên nguyên lý tiến hóa tự nhiên, được lấy cảm hứng từ quá trình chọn lọc tự nhiên của Charles Darwin. GA được phát triển bởi John Holland vào những năm 1970 và là một phần của lĩnh vực tính toán tiến hóa. Thay vì tìm kiếm giải pháp thông qua các phương pháp toán học truyền thống (như gradient descent), GA sử dụng các khái niệm như chọn lọc, lai ghép (crossover), và đột biến (mutation) để tiến hóa một tập hợp các giải pháp tiềm năng (gọi là quần thể - population) qua nhiều thế hệ.



*Hình 2.1: Sơ đồ tổng quan hoạt động GA*

Nguyên lý cơ bản của GA bao gồm:

- Khởi tạo quần thể: Bắt đầu với một tập hợp các giải pháp ngẫu nhiên, mỗi giải pháp được biểu diễn dưới dạng một chuỗi (thường là chuỗi bit hoặc số thực), gọi là nhiễm sắc thể (chromosome).
- Đánh giá độ thích nghi: Mỗi nhiễm sắc thể được đánh giá dựa trên một hàm mục tiêu (fitness function), đo lường mức độ "tốt" của giải pháp.
- Chọn lọc: Các giải pháp có độ thích nghi cao hơn có khả năng được chọn để tạo ra thế hệ tiếp theo, mô phỏng cơ chế "sống sót của kẻ mạnh".
- Lai ghép và đột biến: Áp dụng các toán tử di truyền để tạo ra các giải pháp mới từ các giải pháp cũ, tăng tính đa dạng và khám phá không gian tìm kiếm.
- Lặp lại: Tiếp tục quá trình qua nhiều thế hệ cho đến khi đạt được tiêu chí dừng (ví dụ: số thế hệ tối đa hoặc độ hội tụ của giải pháp).

GA đặc biệt hiệu quả trong các bài toán tối ưu hóa phức tạp, không có gradient rõ ràng, hoặc có không gian tìm kiếm rất lớn, như lập lịch, tối ưu hóa dinh dưỡng, hoặc thiết kế hệ thống

### **2.1.2 Các thành phần cơ bản của GA**

GA bao gồm các thành phần chính sau đây, đóng vai trò quan trọng trong quá trình tiến hóa:

- Quần thể (Population): Tập hợp các giải pháp tiềm năng, thường được khởi tạo ngẫu nhiên. Kích thước quần thể ảnh hưởng đến độ đa dạng và tốc độ hội tụ.
- Nhiễm sắc thể (Chromosome): Biểu diễn của một giải pháp, thường là một chuỗi bit, số thực, hoặc cấu trúc dữ liệu tùy chỉnh (ví dụ: mảng thời gian cho lịch ăn).
- Hàm mục tiêu (Fitness Function): Tiêu chí đánh giá độ tốt của một giải pháp. Hàm này cần được thiết kế sao cho phản ánh chính xác mục tiêu của bài toán.
- Toán tử chọn lọc (Selection Operator): Cơ chế chọn các nhiễm sắc thể có độ thích nghi cao để truyền gen cho thế hệ tiếp theo (ví dụ: roulette wheel selection, tournament selection).

- Toán tử lai ghép (Crossover Operator): Kết hợp các nhiễm sắc thể từ cha mẹ để tạo ra con cái, tăng khả năng khám phá giải pháp mới.
- Toán tử đột biến (Mutation Operator): Giới thiệu ngẫu nhiên các thay đổi nhỏ vào nhiễm sắc thể để duy trì đa dạng di truyền.
- Tiêu chí dừng (Termination Criteria): Điều kiện kết thúc quá trình tiến hóa, như đạt số thế hệ tối đa, độ thích nghi tối ưu, hoặc không cải thiện thêm.

### **2.1.3. Các toán tử di truyền**

Các toán tử di truyền là những công cụ quan trọng giúp thuật toán Di truyền (GA) tạo ra các thế hệ mới, mô phỏng quá trình tiến hóa tự nhiên như chọn lọc, lai ghép, và đột biến. Chúng đóng vai trò cốt lõi trong việc duy trì sự đa dạng di truyền và hướng tới các giải pháp tối ưu, đặc biệt trong bài toán phức tạp như tối ưu hóa lịch ăn của bạn. Dưới đây, mình sẽ giải thích chi tiết từng loại toán tử, cách chúng hoạt động, và cách điều chỉnh để phù hợp với nhu cầu thực tế.

#### **❖ Chọn lọc (Selection)**

- Roulette Wheel Selection: Phương pháp này hoạt động như một bánh xe may mắn. Mỗi nhiễm sắc thể được gán một lát bánh có kích thước tỷ lệ với độ thích nghi của nó. Khi quay bánh, nhiễm sắc thể có độ thích nghi cao hơn sẽ có cơ hội lớn hơn để được chọn. Ưu điểm là đơn giản, nhưng nếu một vài cá thể quá nổi trội, các cá thể khác có thể bị loại bỏ hoàn toàn.
- Tournament Selection: Cách này chọn ngẫu nhiên một nhóm nhỏ nhiễm sắc thể (thường 2-5 cá thể), so sánh độ thích nghi, và chọn cá thể tốt nhất. Quá trình lặp lại nhiều lần để tạo quần thể mới. Điều này giúp giảm sự chi phối của các cá thể quá mạnh, nhưng đòi hỏi nhiều phép so sánh hơn.
- Rank Selection: Tất cả nhiễm sắc thể được sắp xếp theo thứ hạng độ thích nghi, và xác suất chọn dựa trên vị trí thay vì giá trị tuyệt đối. Phương pháp này đảm bảo sự đa dạng bằng cách tránh cho một vài cá thể chiếm ưu thế, dù việc tính toán có thể phức tạp hơn với quần thể lớn.

### ❖ Lai ghép (Crossover)

- Single-Point Crossover: Một điểm ngẫu nhiên được chọn để cắt nhiễm sắc thể của cha và mẹ, sau đó hai đoạn gen từ điểm đó trở đi được trao đổi. Ví dụ, với hai cha mẹ "[7:30, 12:45, 18:15, 1, 4, 3]" và "[7:45, 12:30, 18:00, 2, 5, 1]", nếu cắt giữa "12:45" và "18:15", con cái có thể là "[7:30, 12:45, 18:00, 2, 5, 1]". Phương pháp này đơn giản nhưng có thể giới hạn sự đa dạng.
- Two-Point Crossover: Hai điểm cắt ngẫu nhiên được chọn, và đoạn giữa chúng được trao đổi. Với cùng ví dụ, nếu cắt giữa "12:45" và "18:15" cũng như giữa "1" và "4", kết quả có thể là "[7:30, 12:30, 18:00, 2, 4, 3]". Cách này tăng khả năng tạo tổ hợp mới, nhưng cần cẩn thận để không phá hủy cấu trúc tốt.
- Uniform Crossover: Mỗi gen của con cái được chọn ngẫu nhiên từ cha hoặc mẹ với xác suất 50%. Ví dụ, kết quả có thể là "[7:30, 12:30, 18:15, 2, 4, 1]". Phương pháp này mang lại sự đa dạng cao, nhưng có thể làm mất cấu trúc quan trọng nếu không kiểm soát.

### ❖ Đột biến (Mutation)

Đột biến giới thiệu các thay đổi ngẫu nhiên để duy trì đa dạng di truyền, tránh việc quần thể bị kẹt ở giải pháp cục bộ. Ba kỹ thuật đột biến phổ biến phù hợp với các loại dữ liệu khác nhau.

- Bit Flip Mutation: Thường dùng với chuỗi bit, kỹ thuật này lật một bit ngẫu nhiên từ 0 thành 1 hoặc ngược lại. Ví dụ, "[0, 1, 0, 1]" có thể thành "[0, 0, 0, 1]" nếu bit thứ hai được chọn. Phương pháp này hiệu quả với dữ liệu nhị phân, nhưng không phù hợp với thời gian hay số thực.
- Gaussian Mutation: Dành cho gen số thực, kỹ thuật này thêm một giá trị ngẫu nhiên từ phân phối Gaussian (trung bình 0, độ lệch chuẩn nhỏ như 15 phút) vào một gen. Nếu "07:30" được chọn và giá trị là +10 phút, kết quả là "07:40". Cách này giữ tính liên tục, phù hợp để điều chỉnh giờ ăn, nhưng cần giới hạn để không lệch quá xa.

- Swap Mutation: Hoán đổi hai gen ngẫu nhiên trong nhiễm sắc thể, thường áp dụng cho mảng món ăn. Với "[1, 4, 3]", nếu đổi vị trí 1 và 2, kết quả là "[4, 1, 3]". Phương pháp này giúp thử nghiệm các tổ hợp món ăn mới mà không làm rối cấu trúc tổng thể.

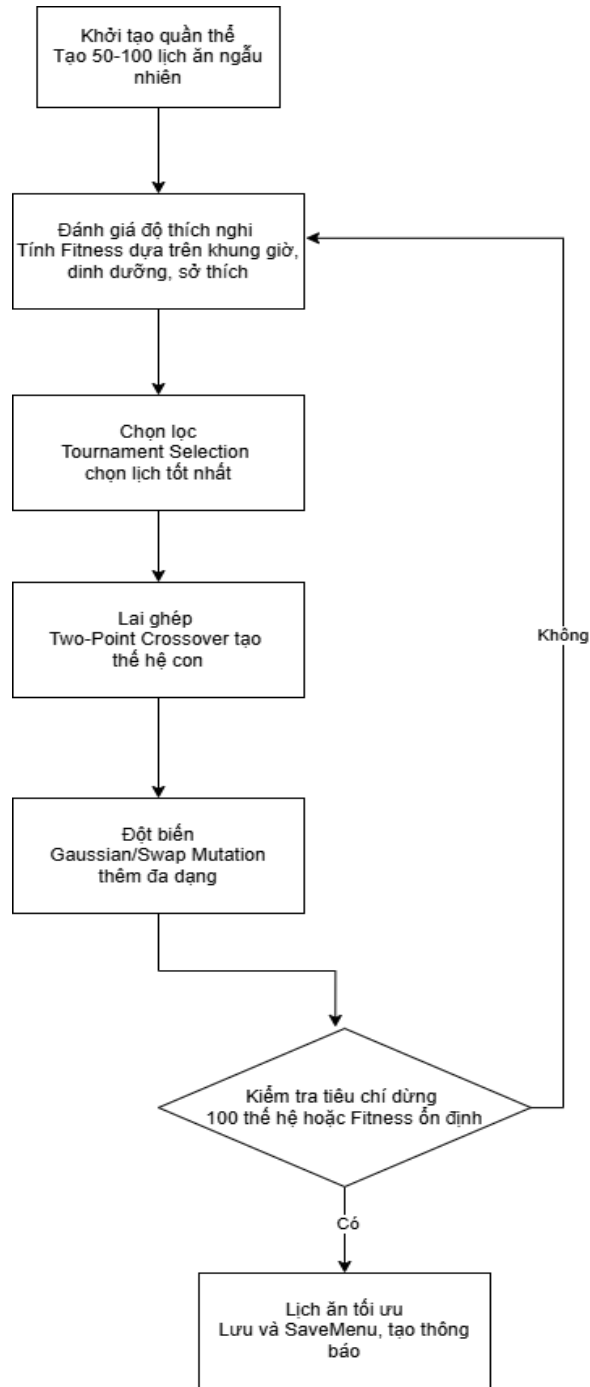
#### ❖ Điều chỉnh tỷ lệ lai ghép và đột biến

- Tỷ lệ lai ghép và đột biến là hai tham số quan trọng cần được điều chỉnh để cân bằng giữa khai thác (tận dụng giải pháp hiện có) và khám phá (tìm kiếm giải pháp mới). Tỷ lệ lai ghép thường dao động từ 0.6 đến 0.9, quyết định mức độ kết hợp gen giữa cha mẹ. Một giá trị cao như 0.8 khuyến khích khai thác các giải pháp đã tốt, trong khi giá trị thấp hơn như 0.6 giữ lại đa dạng. Tỷ lệ đột biến, thường từ 0.01 đến 0.1, điều chỉnh mức độ thay đổi ngẫu nhiên. Giá trị thấp như 0.05 tập trung vào khai thác, còn giá trị cao như 0.1 thúc đẩy khám phá.
- Sự cân bằng này rất quan trọng. Nếu tỷ lệ lai ghép quá cao mà đột biến quá thấp, quần thể có thể bị hội tụ sớm vào một giải pháp không tối ưu, mất đi khả năng tìm kiếm toàn cục. Ngược lại, nếu đột biến quá cao, GA có thể trở nên ngẫu nhiên, không khai thác được các giải pháp đã cải thiện. Trong trường hợp tối ưu hóa lịch ăn, bạn có thể bắt đầu với tỷ lệ lai ghép 0.7 và đột biến 0.05. Nếu Fitness không tăng sau 20 thế hệ, hãy thử tăng đột biến lên 0.07 để khám phá thêm, hoặc giảm xuống 0.03 nếu muốn tinh chỉnh giải pháp hiện có. Điều chỉnh này nên được thử nghiệm với dữ liệu thực tế của bạn để tìm ra thông số tối ưu.

## 2.2. Thuật toán GA trong đồ án

Thuật toán Di truyền (GA) là một phương pháp tối ưu hóa dựa trên nguyên lý tiến hóa tự nhiên, được áp dụng để tạo ra lịch ăn cá nhân hóa tối ưu. Mục tiêu là sắp xếp khung giờ ăn hợp lý, chọn món ăn từ Food để đáp ứng nhu cầu dinh dưỡng từ user, và khuyến khích tuân thủ dựa trên ComplianceHistory. Quy trình GA diễn ra qua nhiều thế hệ, với

các bước được minh họa trong sơ đồ. Dưới đây là chi tiết từng bước và ý nghĩa trong bối cảnh đề án.



*Hình 2.2: Sơ đồ luồng hoạt động của GA trong đề án*

Quy trình hoạt động chi tiết của GA:

❖ **Bước 1: Khởi tạo quần thể**

Quá trình bắt đầu với việc khởi tạo một quần thể, tức là tập hợp các giải pháp tiềm năng, trong trường hợp này là các lịch ăn ngẫu nhiên. GA tạo ra từ 50 đến 100 lịch ăn, mỗi lịch được biểu diễn dưới dạng một nhiễm sắc thể – một chuỗi gồm thời gian bắt đầu của các bữa ăn và danh sách món ăn tương ứng. Thời gian được chọn ngẫu nhiên trong khoảng linh hoạt quanh IDEAL\_MEAL, còn món ăn được lấy từ Food sao cho không vi phạm restrictions (như "low\_sodium"). Bước này đặt nền móng cho quá trình tiến hóa, cung cấp sự đa dạng ban đầu để GA khám phá.

❖ **Bước 2: Đánh giá độ thích nghi**

Sau khi có quần thể, GA tiến hành đánh giá độ "tốt" của từng lịch ăn thông qua bước "Đánh giá độ thích nghi" với mô tả "Tính Fitness dựa trên khung giờ, dinh dưỡng, sở thích". Đây là giai đoạn quan trọng, nơi mỗi nhiễm sắc thể được chấm điểm dựa trên một hàm mục tiêu (fitness function) được thiết kế đặc biệt cho đề án. Hàm này kết hợp ba yếu tố chính:

- Cân bằng dinh dưỡng: Tính tổng calo, protein, carbohydrate, và chất béo từ các món ăn (dựa trên Food) và so sánh với dailyCalorieNeeds từ user.
- Phù hợp sở thích/hạn chế: Đếm số lượng preferences (như "healthy") khớp với người dùng và đảm bảo không vi phạm restrictions (như "low\_sodium").

❖ **Bước 3: Chọn lọc (Selection)**

- Mục tiêu: Chọn ra các thể hệ tốt nhất để làm cha mẹ cho thế hệ tiếp theo
- Thực hiện:
  - Sử dụng phương pháp Tournament Selection: chọn ngẫu nhiên một nhóm nhỏ cá thể, cá thể có fitness cao nhất trong nhóm sẽ được chọn.
  - Ngoài ra, giữ lại một số cá thể tốt nhất (elite) để đảm bảo không mất các giải pháp tốt.

❖ **Bước 4: Lai ghép (Crossover)**

- Mục tiêu: Tạo ra các cá thể con mới bằng cách kết hợp thông tin di truyền từ hai cá thể cha mẹ.
- Thực hiện:
  - Sử dụng hai chiến lược:
    - + Uniform Crossover: Mỗi bữa ăn của cá thể con được chọn ngẫu nhiên từ một trong hai cha mẹ.
    - + Single-Point Crossover: Chọn một điểm cắt, kết hợp các bữa ăn trước điểm cắt từ cha mẹ 1 và sau điểm cắt từ cha mẹ 2.
  - Tỷ lệ sử dụng hai chiến lược là 50/50.

❖ **Bước 5: Đột biến (Mutation)**

- Mục tiêu: Tăng tính đa dạng cho quần thể, tránh rơi vào cực trị địa phương.
- Thực hiện:
  - Với một xác suất nhất định (ví dụ 20%), thực hiện đột biến trên từng bữa ăn của cá thể con:
    - + Đột biến toàn bộ bữa ăn: Thay thế toàn bộ món ăn trong bữa bằng các món mới phù hợp với mục tiêu dinh dưỡng.
    - + Đột biến từng món: Thay thế hoặc thêm một món ăn trong bữa, ưu tiên món phù hợp với phần dinh dưỡng còn thiếu.
  - Đảm bảo số lượng món ăn trong mỗi bữa không vượt quá giới hạn cho phép.

❖ **Bước 6: Kiểm tra điều kiện dừng**

- Mục tiêu: Xác định khi nào thuật toán nên dừng lại.
- Thực hiện:
  - Thuật toán dừng lại khi đạt một trong các điều kiện sau:
    - + Đã chạy đủ số thế hệ (ví dụ 100 thế hệ).

- + Fitness của cá thể tốt nhất đạt ngưỡng mong muốn (ví dụ  $> 0.95$ ).
- + Fitness không cải thiện sau một số thế hệ liên tiếp (early stopping).

❖ **Bước 7: Lưu kết quả tối ưu**

- Mục tiêu: Lưu lại lịch ăn tối ưu và thông báo cho người dùng.
- Thực hiện:
  - Lịch ăn tối ưu sẽ được lưu vào cơ sở dữ liệu (SaveMenu).
  - Có thể tạo thông báo nhắc nhở hoặc gửi email cho người dùng về lịch ăn mới.

## **2.3. Ưu điểm và hạn chế**

### **2.3.1. Ưu điểm**

❖ **Khả năng tối ưu hóa phức tạp:**

- GA nổi bật trong việc giải quyết các bài toán tối ưu hóa đa mục tiêu (multi-objective optimization), chẳng hạn như cân bằng dinh dưỡng (calo, protein, carb, fat), đa dạng món ăn, và tuân thủ sở thích cá nhân. Thuật toán này khám phá không gian giải pháp lớn thông qua quá trình tiến hóa, tìm ra thực đơn tối ưu mà các phương pháp truyền thống (như lập trình tuyến tính) khó đạt được.
- Trong đồ án, GA được áp dụng để tạo thực đơn cá nhân hóa, đảm bảo các chỉ số dinh dưỡng gần đúng với mục tiêu (sai số dưới 10%), thể hiện hiệu quả trong xử lý dữ liệu phức tạp.

❖ **Tính linh hoạt và thích nghi:**

- GA cho phép điều chỉnh các tham số (population\_size, mutation\_rate, generations) để phù hợp với yêu cầu cụ thể của người dùng, như chế độ ăn kiêng, hạn chế dị ứng, hoặc sở thích cá nhân. Cơ chế đột biến và lai ghép giúp thuật toán thích nghi với dữ liệu thay đổi, ví dụ: cập nhật món ăn mới hoặc thay đổi mục tiêu dinh dưỡng.

- Điều này đặc biệt hữu ích trong hệ thống của bạn, nơi `meal_targets` và `user_profile` có thể thay đổi theo thời gian.

❖ **Khả năng xử lý không gian giải pháp lớn:**

- Với số lượng món ăn và tổ hợp bữa ăn lớn, GA sử dụng cơ chế chọn lọc tự nhiên để khám phá hàng ngàn khả năng, vượt trội so với các phương pháp thử và sai thủ công. Trong trường hợp đồ án, quần thể 50 cá thể qua 100 thế hệ đảm bảo tìm được thực đơn tối ưu trong thời gian hợp lý (1-2 phút).
- Ưu điểm này giúp hệ thống tự động hóa, giảm công sức của người dùng trong việc thử nghiệm các tổ hợp thực đơn.

❖ **Tính sáng tạo và đa dạng:**

- Nhờ cơ chế đột biến và lai ghép, GA có thể tạo ra các thực đơn sáng tạo, tránh lặp lại món ăn và tăng trải nghiệm người dùng. Trong code của bạn, `diversity_score` và `balance_score` trong `calculate_fitness` khuyến khích sự đa dạng, làm cho thực đơn phong phú hơn.
- Điều này đáp ứng nhu cầu về tính cá nhân hóa và sự mới mẻ trong chế độ ăn uống.

### 2.3.2. Hạn chế

❖ **Thời gian tính toán:**

- GA đòi hỏi nhiều lần lặp (`generations`) và đánh giá `fitness` cho từng cá thể, dẫn đến thời gian xử lý lâu hơn so với các thuật toán đơn giản như quy hoạch tuyến tính, đặc biệt khi quần thể lớn hoặc dữ liệu món ăn phong phú. Trong đồ án, việc xử lý 50 cá thể qua 100 thế hệ có thể mất 1-2 phút, không phù hợp với các yêu cầu thời gian thực.
- Hạn chế này có thể được cải thiện bằng cách tối ưu hóa tham số (giảm `population_size` hoặc `generations`) hoặc sử dụng phần cứng mạnh hơn.

❖ **Phụ thuộc vào tham số ban đầu:**

- Hiệu quả của GA phụ thuộc vào các tham số như `population_size` (50), `mutation_rate` (0.2), và `elite_size` (5). Nếu các giá trị này không

được điều chỉnh hợp lý, thuật toán có thể hội tụ chậm hoặc rơi vào giải pháp cục bộ (local optima). Trong code của bạn, `mutation_rate` được điều chỉnh động, nhưng vẫn cần thử nghiệm để tối ưu hóa.

- Điều này đòi hỏi kinh nghiệm hoặc thử nghiệm nhiều lần để tìm bộ tham số lý tưởng.

❖ **Khó kiểm soát chính xác:**

- GA không đảm bảo tìm ra giải pháp tối ưu tuyệt đối mà chỉ tìm kiếm giải pháp gần tối ưu (near-optimal solution). Trong trường hợp thực đơn, sai số dinh dưỡng (dù dưới 10%) có thể không đáp ứng hoàn toàn các yêu cầu nghiêm ngặt, đặc biệt với người bệnh cần dinh dưỡng chính xác.
- Hạn chế này có thể được khắc phục bằng cách kết hợp GA với các thuật toán khác (như hill climbing) để tinh chỉnh kết quả.

❖ **Yêu cầu dữ liệu chất lượng cao:**

- GA phụ thuộc vào dữ liệu đầu vào (danh sách foods, user\_profile, meal\_targets). Nếu dữ liệu không đầy đủ hoặc không chính xác (ví dụ: thiếu thông tin dinh dưỡng của món ăn), kết quả thực đơn sẽ kém tin cậy. Trong đồ án, chất lượng cơ sở dữ liệu MongoDB là yếu tố quyết định hiệu quả thuật toán.
- Điều này đòi hỏi đầu tư vào việc thu thập và xác thực dữ liệu ban đầu.

## 2.3. Công nghệ sử dụng

❖ Node.js với Express.js:

- Sử dụng để xây dựng RESTful API, cho phép xử lý các yêu cầu từ frontend một cách hiệu quả.
- Hỗ trợ quản lý các chức năng đăng ký, đăng nhập, và xác thực người dùng thông qua JWT (JSON Web Tokens), đảm bảo tính bảo mật và phân quyền.

- Express.js cung cấp framework nhẹ nhàng, dễ mở rộng, phù hợp để xây dựng backend quản lý dữ liệu người dùng.
  - Đảm bảo giao tiếp mượt mà với cơ sở dữ liệu MongoDB, hỗ trợ xử lý các tác vụ như lưu trữ hồ sơ người dùng và xác thực email.
- ❖ Python với FastAPI:
- Triển khai thuật toán Di truyền (GA) để tối ưu hóa thực đơn, tận dụng sức mạnh xử lý dữ liệu phức tạp của Python.
  - Xử lý các phép tính dinh dưỡng và tối ưu hóa dựa trên mục tiêu cá nhân, đảm bảo thực đơn cân bằng và phù hợp.
  - FastAPI được chọn nhờ tốc độ xử lý nhanh nhờ hỗ trợ asynchronous, phù hợp với các tác vụ tính toán nặng.
  - Tích hợp tốt với các thư viện khoa học dữ liệu, giúp nâng cao hiệu quả trong việc phân tích và tối ưu hóa dữ liệu dinh dưỡng.
- ❖ MongoDB:
- Là cơ sở dữ liệu NoSQL được sử dụng để lưu trữ thông tin người dùng, bao gồm dữ liệu cá nhân và lịch sử dinh dưỡng.
  - Lưu trữ danh sách món ăn với thông tin chi tiết (calo, protein, carb, fat) trong collection Food.
  - Quản lý các thực đơn đã tạo và lịch sử tuân thủ qua các collection như SaveMenu và ComplianceHistory.
  - Hỗ trợ mô hình dữ liệu linh hoạt với định dạng JSON, phù hợp để lưu trữ và truy xuất dữ liệu phức tạp, đảm bảo hiệu quả quản lý thông tin dinh dưỡng.

## **2.4. Kết luận chương**

Chương 2 đã cung cấp nền tảng lý thuyết quan trọng cho việc triển khai hệ thống tạo thực đơn cá nhân hóa dựa trên thuật toán Di truyền (GA). Phần này đã trình bày các khái niệm cơ bản về GA, bao gồm cơ chế chọn lọc, lai ghép, đột biến, và cách đánh giá độ thích nghi, làm rõ cách thức thuật toán tối ưu hóa thực đơn dựa trên dữ liệu dinh dưỡng và sở thích người dùng. Các ưu điểm như khả năng xử lý không gian giải pháp lớn, tính

linh hoạt, và sáng tạo của GA đã được làm nổi bật, cùng với các hạn chế như thời gian tính toán và phụ thuộc vào tham số ban đầu. Ngoài ra, phần công nghệ sử dụng đã giới thiệu các công cụ quan trọng như Node.js với Express.js và JWT cho quản lý người dùng, Python với FastAPI cho tối ưu hóa GA, và MongoDB cho lưu trữ dữ liệu, tạo cơ sở kỹ thuật cho việc thiết kế và triển khai. Những nội dung này đặt nền tảng vững chắc để tiến hành phân tích, thiết kế, và thực hiện các giai đoạn tiếp theo, hướng tới một giải pháp hiệu quả và thực tiễn trong lĩnh vực dinh dưỡng cá nhân hóa.

## **CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

### **3.1. Phân tích yêu cầu**

#### **3.1.1. Yêu cầu chức năng**

##### **3.1.1.1. Quản lý người dùng**

Mô-đun quản lý người dùng là nền tảng của hệ thống, đảm bảo mỗi cá nhân có thể tương tác với ứng dụng một cách an toàn và cá nhân hóa. Yêu cầu cụ thể bao gồm:

- Đăng ký và đăng nhập tài khoản: Người dùng cần có khả năng tạo tài khoản mới bằng cách cung cấp thông tin cơ bản (như tên, email, mật khẩu) và đăng nhập để truy cập hệ thống. Chức năng này đảm bảo tính bảo mật và xác thực danh tính, cho phép người dùng lưu trữ dữ liệu cá nhân riêng biệt.
- Cập nhật thông tin cá nhân: Người dùng có thể chỉnh sửa thông tin như tuổi, cân nặng, chiều cao, hoặc cấp độ hoạt động (activity level) bất kỳ lúc nào. Điều này cho phép hệ thống cập nhật nhu cầu dinh dưỡng (dailyCalorieNeeds) và điều chỉnh các tính toán liên quan, đảm bảo tính linh hoạt và chính xác.
- Thiết lập mục tiêu dinh dưỡng: Người dùng có thể định nghĩa mục tiêu cụ thể, chẳng hạn duy trì cân nặng, giảm cân, hoặc tăng cơ, kèm theo tỷ lệ macronutrients (protein, carbohydrate, chất béo) mong muốn. Chức năng này cho phép hệ thống cá nhân hóa thực đơn dựa trên nhu cầu riêng biệt, ví dụ, 1500 kcal với 20% protein cho người muốn giảm cân.
- Lưu trữ sở thích và hạn chế ăn uống: Người dùng nhập sở thích (như "healthy", "high\_protein") và hạn chế (như "low\_sodium", "no\_dairy") để hệ thống tránh món không phù hợp và ưu tiên món ưa thích. Dữ liệu này được lưu trong collection user, hỗ trợ thuật toán GA trong việc tối ưu hóa thực đơn.

### **3.1.1.2. Quản lý thực đơn**

Mô-đun này tập trung vào việc tạo, chỉnh sửa, và lưu trữ thực đơn, tận dụng thuật toán GA để tự động hóa quá trình tối ưu hóa. Các yêu cầu cụ thể bao gồm:

- Tạo thực đơn tự động sử dụng thuật toán GA: Hệ thống sử dụng GA để sinh ra lịch ăn tối ưu, cân bằng khung giờ (IDEAL\_MEAL\_TIMES), dinh dưỡng (dailyCalorieNeeds), và sở thích/hạn chế. Ví dụ, GA có thể đề xuất [7:30, 12:45, 18:15, 1, 4, 3] (món 1: "Com gạo lứt", món 4: "Salad gà") dựa trên dữ liệu người dùng.
- Xem thực đơn theo ngày: Người dùng có thể xem chi tiết lịch ăn cho ngày hiện tại hoặc các ngày khác, bao gồm giờ ăn, món ăn, và thông tin dinh dưỡng (calo, protein, v.v.) từ SaveMenu. Chức năng này giúp theo dõi dễ dàng và lên kế hoạch trước.
- Chỉnh sửa thực đơn thủ công: Người dùng được phép thay đổi giờ ăn hoặc chọn món khác từ Food nếu cần. Ví dụ, thay "12:45" thành "13:00" hoặc đổi món 4 thành món 6 ("Cá hồi nướng"). Điều này tăng tính linh hoạt và phù hợp với thói quen cá nhân.
- Lưu trữ thực đơn yêu thích: Người dùng có thể đánh dấu các thực đơn ưa thích để lưu vào SaveMenu với trạng thái "yêu thích". Chức năng này cho phép tái sử dụng các lịch ăn hiệu quả trong tương lai.
- Áp dụng thực đơn yêu thích cho ngày khác: Người dùng chọn thực đơn yêu thích và áp dụng cho ngày khác, với khả năng điều chỉnh giờ ăn hoặc món ăn nếu cần. Điều này tiết kiệm thời gian và duy trì tính nhất quán trong chế độ ăn.

### **3.1.1.3. Theo dõi tiến độ**

Mô-đun này hỗ trợ người dùng giám sát quá trình thực hiện thực đơn và đánh giá hiệu quả dinh dưỡng. Các yêu cầu bao gồm:

- Đánh dấu hoàn thành các bữa ăn: Người dùng xác nhận khi hoàn thành một bữa ăn, cập nhật dữ liệu vào ComplianceHistory (như

meals\_completed và compliance). Điều này giúp theo dõi mức độ tuân thủ theo thời gian thực.

- **Hiển thị biểu đồ tiến độ dinh dưỡng:** Hệ thống tạo biểu đồ (sử dụng canvas panel) để hiển thị tổng calo, protein, carb, và fat tiêu thụ so với mục tiêu hàng ngày. Ví dụ, biểu đồ cột có thể cho thấy 1480 kcal/1500 kcal, giúp người dùng dễ dàng nhận diện thiếu hụt.
- **Theo dõi tỷ lệ tuân thủ thực đơn:** Hệ thống tính toán tỷ lệ tuân thủ dựa trên ComplianceHistory, hiển thị dưới dạng phần trăm (ví dụ: 75% tuân thủ trong tuần). Chức năng này khuyến khích người dùng duy trì thói quen.
- **Xem lịch sử tiến độ theo tuần/tháng:** Người dùng truy cập dữ liệu tiến độ (calo, tuân thủ) theo tuần hoặc tháng, lưu trong ComplianceHistory. Điều này hỗ trợ phân tích dài hạn và điều chỉnh mục tiêu dinh dưỡng.

### **3.1.2. Yêu cầu phi chức năng**

#### **3.1.2.1. Hiệu suất**

Hệ thống cần đảm bảo hiệu suất cao để đáp ứng trải nghiệm người dùng.

- **Thời gian phản hồi < 2 giây:** Đảm bảo mọi thao tác (như tạo thực đơn hoặc xem tiến độ) được xử lý nhanh chóng.
- **Tối ưu hóa việc tạo thực đơn bằng GA:** Thuật toán GA được tinh chỉnh để giảm thời gian tính toán, ví dụ, sử dụng quần thể nhỏ hơn hoặc chạy song song.
- **Xử lý nhanh các truy vấn database:** Tối ưu hóa truy vấn từ user, Food, và ComplianceHistory để giảm độ trễ.

#### **3.1.2.2. Bảo mật**

Bảo mật là yếu tố cốt lõi để bảo vệ thông tin người dùng.

- **Mã hóa dữ liệu người dùng:** Dữ liệu cá nhân và dinh dưỡng được mã hóa (ví dụ, sử dụng AES-256) để ngăn chặn truy cập trái phép.
- **Xác thực và phân quyền người dùng:** Sử dụng đăng nhập bảo mật và phân quyền (admin/user) để kiểm soát truy cập.

- Bảo vệ thông tin cá nhân: Đảm bảo thông tin như email, sở thích không bị rò rỉ.
- Tuân thủ quy định bảo mật: Áp dụng các tiêu chuẩn như GDPR hoặc quy định địa phương để bảo vệ dữ liệu.

### **3.1.2.3. Khả năng mở rộng**

Hệ thống cần linh hoạt để phát triển trong tương lai.

- Kiến trúc module hóa để mở rộng: Thiết kế mô-đun (quản lý người dùng, thực đơn) cho phép thêm tính năng mới mà không ảnh hưởng tổng thể.
- Hỗ trợ thêm tính năng mới: Dễ dàng tích hợp các tính năng như theo dõi tập luyện.
- Khả năng tích hợp với hệ thống khác: Hỗ trợ kết nối với thiết bị đeo hoặc ứng dụng bên thứ ba.
- Dễ dàng cập nhật và bảo trì: Cấu trúc mã nguồn rõ ràng giúp nâng cấp và sửa lỗi nhanh chóng.

### **3.1.2.4. Giao diện người dùng**

Giao diện cần thân thiện và phù hợp với mọi đối tượng.

- Thiết kế responsive: Hỗ trợ mượt mà trên điện thoại, máy tính bảng, và máy tính.
- Giao diện trực quan, dễ sử dụng: Bố cục đơn giản, nút bấm rõ ràng, ví dụ, xem thực đơn chỉ một lần chạm.
- Hỗ trợ đa ngôn ngữ (tiếng Việt): Cung cấp giao diện bằng tiếng Việt để người dùng địa phương dễ thao tác.
- Trải nghiệm mượt mà: Tối ưu hóa tốc độ tải và hiệu ứng để đảm bảo trải nghiệm liền mạch.

### **3.1.2.5. Độ tin cậy**

Hệ thống cần hoạt động ổn định và đáng tin cậy.

- Hoạt động 24/7: Đảm bảo hệ thống luôn sẵn sàng, kể cả trong giờ cao điểm.
- Backup dữ liệu tự động: Sao lưu dữ liệu từ SaveMenu và ComplianceHistory định kỳ để tránh mất mát.
- Xử lý lỗi và phục hồi: Thiết lập cơ chế tự động khắc phục lỗi và khôi phục dữ liệu khi xảy ra sự cố.
- Báo cáo và giám sát: Cung cấp báo cáo hoạt động và giám sát hiệu suất để phát hiện vấn đề kịp thời.

## **3.2. Quy trình hoạt động chi tiết của GA trong đồ án**

### **3.2.1. Quy trình hoạt động**

#### **3.2.1.1. Khởi tạo**

- ❖ Tạo ngẫu nhiên 50 thực đơn ban đầu, mỗi thực đơn đại diện cho một cá thể trong quần thể.
- ❖ Mỗi bữa ăn trong thực đơn chứa số lượng món ngẫu nhiên từ 1 đến 5 món, đảm bảo tính đa dạng.
- ❖ Lọc các món ăn dựa trên sở thích và hạn chế của người dùng (dữ liệu từ collection User) để loại bỏ các món không phù hợp.

#### **3.2.1.2. Đánh giá độ thích nghi (Fitness)**

- ❖ Sử dụng hàm fitness để đánh giá chất lượng thực đơn dựa trên các tiêu chí:
  - Độ chính xác dinh dưỡng (70% trọng số):
    - Calo: 40%
    - Protein: 30%
    - Carbohydrates: 20%
    - Chất béo: 10%
  - Tính đa dạng món ăn (5% trọng số): Đánh giá sự khác biệt giữa các bữa ăn.
  - Cân bằng dinh dưỡng (5% trọng số): Đảm bảo tỷ lệ macronutrients phù hợp.
- ❖ Điểm fitness được tính toán từ 0 đến 1, với giá trị càng cao càng tối ưu.

### **3.2.1.3. Chọn lọc (Selection)**

❖ Áp dụng phương pháp chọn lọc kết hợp:

- Elite Selection: Giữ lại 5 thực đơn có điểm fitness cao nhất để truyền sang thế hệ tiếp theo.
- Tournament Selection:
  - Chọn ngẫu nhiên 3-10% quần thể (2-5 cá thể).
  - Chọn thực đơn có điểm fitness cao nhất trong nhóm.
  - Lập lại cho đến khi đủ số lượng cá thể cho thế hệ mới.

### **3.2.1.4. Lai ghép (Crossover)**

❖ Sử dụng hai phương pháp lai ghép với tỷ lệ 50% mỗi loại:

- Uniform Crossover: Chọn ngẫu nhiên các bữa ăn từ hai thực đơn cha mẹ để tạo thực đơn con mới.
- Single-point Crossover: Chọn một điểm cắt ngẫu nhiên, kết hợp phần đầu của thực đơn cha với phần sau của thực đơn mẹ.

### **3.2.1.5. Đột biến (Mutation)**

❖ Áp dụng tỷ lệ đột biến thích nghi từ 10-40%:

- Đột biến toàn bộ bữa ăn (20% cơ hội): Thay thế toàn bộ bữa ăn bằng các món mới, đảm bảo phù hợp với mục tiêu dinh dưỡng.
- Đột biến từng món (80% cơ hội): Thêm món mới hoặc thay thế món hiện có, tuân thủ sở thích và hạn chế.

### **3.2.1.6. Điều kiện dừng**

❖ Thuật toán dừng khi:

- Điểm fitness đạt  $> 0.95$ .
- Không cải thiện sau 20 thế hệ.
- Đạt số thế hệ tối đa (100).

## **3.2.2. Cơ chế tối ưu hóa**

### **3.2.2.1. Điều chỉnh tham số**

- Tăng tỷ lệ đột biến khi không có cải thiện để thúc đẩy khám phá.
- Giảm tỷ lệ đột biến khi có cải thiện tốt để tập trung khai thác.
- Cân bằng giữa khai thác (tận dụng giải pháp hiện có) và khám phá (tìm kiếm giải pháp mới).

### **3.2.2.2. Cân bằng dinh dưỡng**

- Điều chỉnh khối lượng món ăn để tối ưu hóa tỷ lệ dinh dưỡng.
- Đảm bảo cân bằng giữa protein, carb, và chất béo theo mục tiêu cá nhân.
- Duy trì đa dạng món ăn qua cơ chế đột biến và lai ghép.

### **3.2.2.3. Xử lý ràng buộc**

- Kiểm tra sở thích người dùng từ dữ liệu đầu vào (collection User).
- Xử lý hạn chế về sức khỏe (dị ứng, kiêng cử) để loại bỏ món không phù hợp.
- Đảm bảo tính khả thi của thực đơn bằng cách xác nhận nguồn nguyên liệu.

## **3.3. Các tác nhân chính của hệ thống**

### **3.3.1. Tác nhân: User**

- ❖ **Vai trò:** User là người dùng cuối, bao gồm các cá nhân có nhu cầu sử dụng hệ thống để tạo thực đơn cá nhân hóa dựa trên mục tiêu sức khỏe và sở thích. Tác nhân này là trung tâm của hệ thống, đại diện cho đối tượng chính được phục vụ
- ❖ **Chức năng:**
  - Đăng ký, đăng nhập: Tạo tài khoản và truy cập hệ thống để sử dụng các tính năng, thông qua giao diện thân thiện được thiết kế.
  - Cập nhật thông tin cá nhân: Nhập dữ liệu như tuổi, cân nặng, mục tiêu calo, sở thích, và hạn chế dinh dưỡng (qua collection User), cung cấp cơ sở cho GA tối ưu hóa.
  - Quản lý thực đơn: Tạo, xem, chỉnh sửa, hoặc lưu trữ thực đơn cá nhân (collection SaveMenu) dựa trên kết quả từ thuật toán GA.
  - Tìm kiếm món ăn: Tìm kiếm và chọn lọc món ăn dựa trên tiêu chí dinh dưỡng hoặc sở thích, tăng tính linh hoạt trong sử dụng.
  - Lên danh sách mua sắm: Tự động tạo danh sách mua sắm dựa trên thực đơn, giúp người dùng chuẩn bị nguyên liệu dễ dàng.

### 3.3.2. Tác nhân: Admin

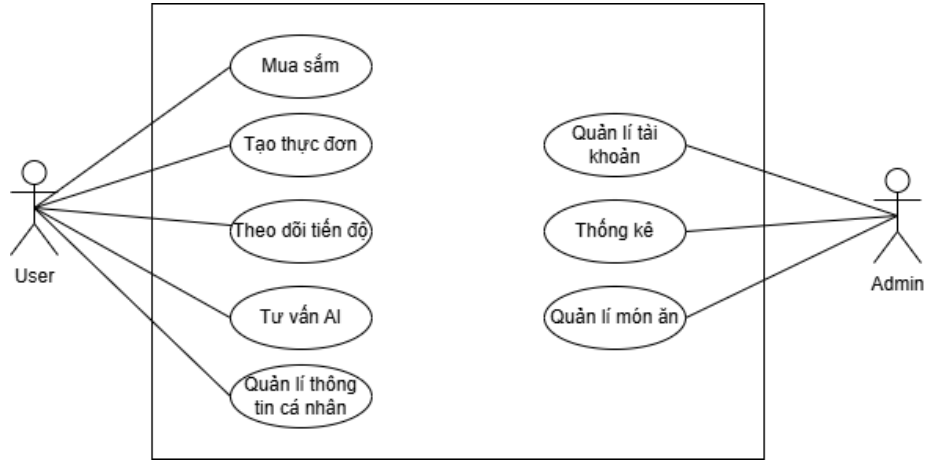
- ❖ **Vai trò:** Admin là người quản trị hệ thống, chịu trách nhiệm quản lý toàn diện các hoạt động và dữ liệu của ứng dụng "Smart Nutrition". Tác nhân này đại diện cho đội ngũ phát triển hoặc quản lý, có quyền kiểm soát cao nhất trong hệ thống.
- ❖ **Chức năng:**
  - **Quản lý tài khoản:** Admin có khả năng giám sát, thêm, sửa, hoặc xóa tài khoản người dùng (User) để đảm bảo tính bảo mật và duy trì chất lượng dữ liệu.
  - **Quản lý món ăn:** Cập nhật, chỉnh sửa danh sách món ăn (collection Food) bao gồm thông tin dinh dưỡng (calo, protein, carb, fat) và các thuộc tính như sở thích hoặc hạn chế, đảm bảo dữ liệu đầu vào cho GA chính xác.

Tác nhân (Actor)	Chức năng
Admin	<ul style="list-style-type: none"><li>• Quản lý tài khoản</li><li>• Thống kê</li><li>• Quản lý món ăn</li></ul>
User	<ul style="list-style-type: none"><li>• Đăng ký, đăng nhập</li><li>• Cập nhật thông tin cá nhân</li><li>• Quản lý thực đơn</li><li>• Quản lý tiến độ</li><li>• Tìm kiếm món ăn</li><li>• Lên danh sách mua sắm</li></ul>

*Bảng 3.1: Tác nhân*

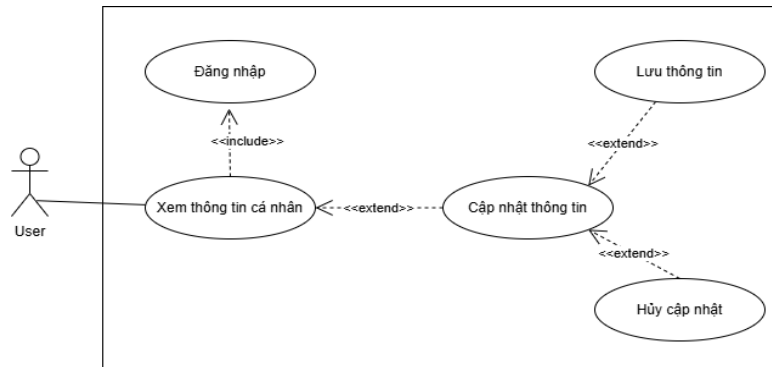
### 3.4. Sơ đồ ca sử dụng

#### 3.4.1. Sơ đồ usecase tổng quát



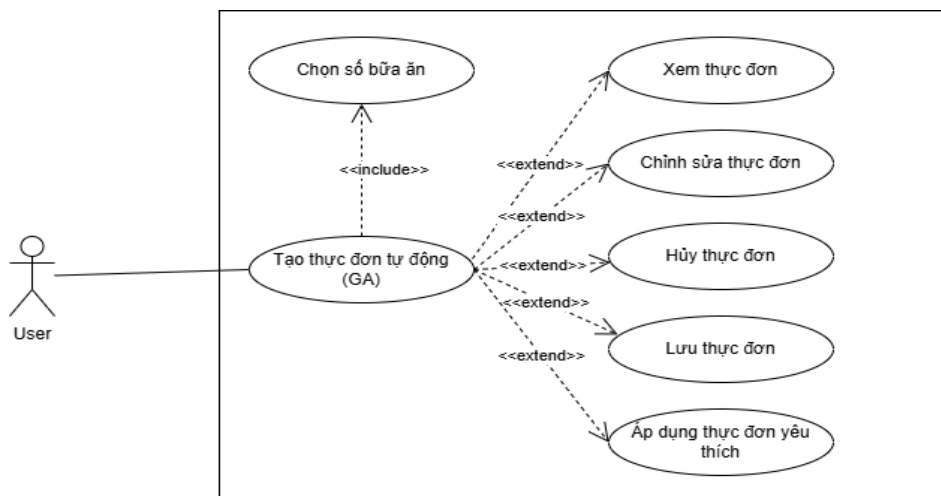
Hình 3.1: Sơ đồ usecase tổng quát

#### 3.4.2. Quản lý thông tin cá nhân



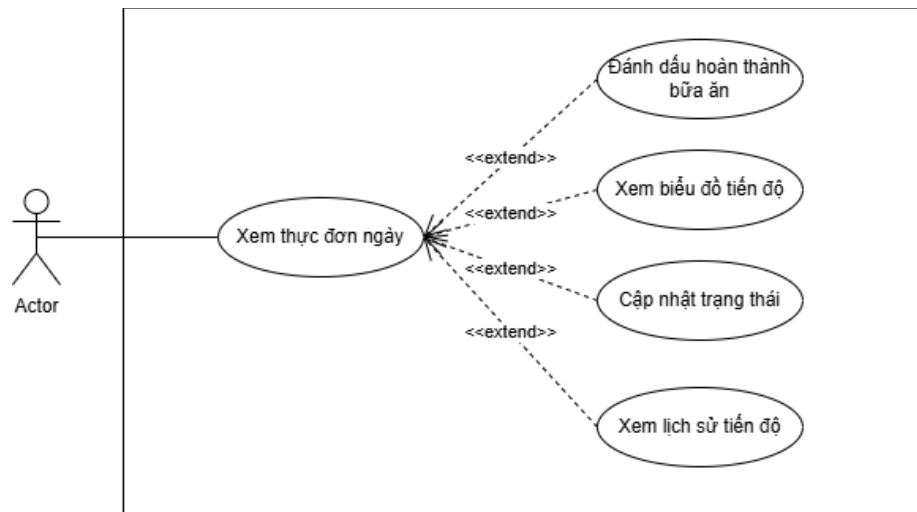
Hình 3.2: Sơ đồ usecase quản lý thông tin cá nhân

#### 3.4.3. Quản lý thực đơn



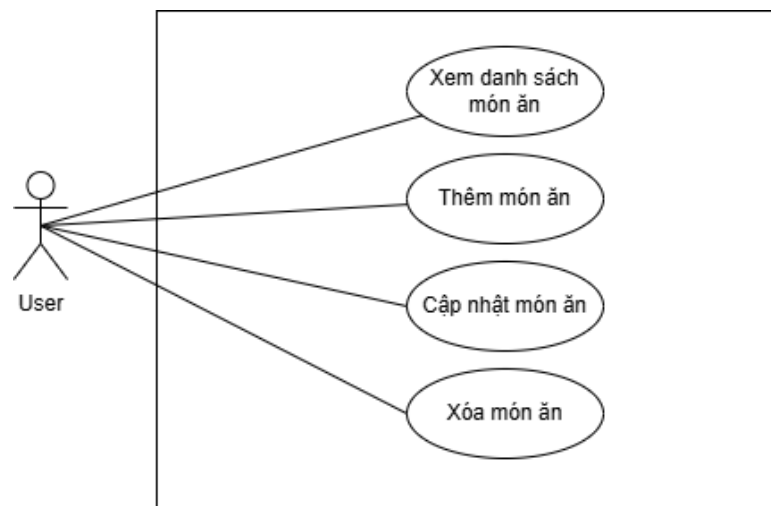
Hình 3.3: Sơ đồ usecase quản lý thực đơn

### 3.4.4. Theo dõi tiến độ



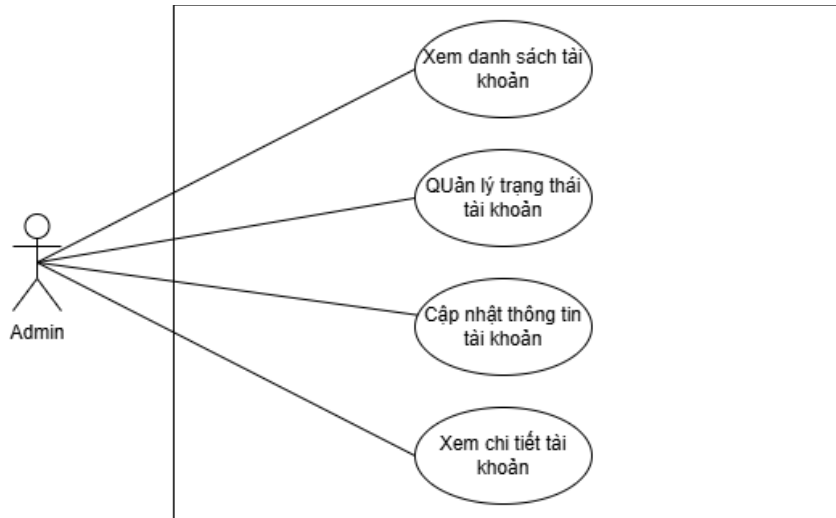
*Hình 3.4: Sơ đồ usecase theo dõi tiến độ*

### 3.4.5. Quản lý món ăn



*Hình 3.5: Sơ đồ usecase quản lý món ăn*

### 3.4.6. Quản lý tài khoản



*Hình 3.6: Sơ đồ usecase quản lý tài khoản*

## 3.5. Đặc tả usecase

### 3.5.1 Đặc tả usecase quản lý thông tin cá nhân

Tên usecase: Quản lý thông tin cá nhân
Mô tả: Cho phép người dùng cập nhật đầy đủ thông tin cá nhân để hệ thống tính toán các chỉ số dinh dưỡng
Tác nhân: User
Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống.
Hậu điều kiện: Cập nhật thành công thông tin cá nhân, hệ thống tiến hành tính toán các chỉ số dinh dưỡng
Luồng sự kiện chính

Tác nhân	Hệ thống
1. Sau khi đăng ký thành công, người dùng tiến hành đăng nhập lần đầu	2. Hiện thị giao diện form cập nhật thông tin cá nhân
3. Người dùng nhập nhấn “Lưu thay đổi”	4. Hệ thống lưu thông tin và hiện thị trang chủ của hệ thống
5. Người dùng vào trang “Hồ sơ cá nhân” rồi nhấn nút “Chỉnh sửa để cập nhật thông tin”	6. Hệ thống hiện thị form chỉnh sửa thông tin cá nhân
7. Người dùng nhấn nút “Lưu” sau khi cập nhật thông tin	8. Hệ thống lưu thông tin cá nhân
Luồng sự kiện thay thế	
5.1 Người dùng ấn nút “Hủy”.	5.2 Hệ thống quay trở lại giao diện thông tin cá nhân

*Bảng 3.2: Đặc tả usecase quản lý thông tin cá nhân*

### **3.5.2. Đặc tả usecase quản lý thực đơn**

Tên usecase: Quản lý thực đơn
Mô tả: Cho phép người dùng tạo thực đơn phù hợp với mức dinh dưỡng mà hệ thống đã tính toán dựa trên thông tin cá nhân
Tác nhân: User
Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống và đã cập nhật đầy đủ thông tin cá nhân
Hậu điều kiện: Hệ thống tạo ra tổ hợp các món ăn cho các bữa của người dùng
Luồng sự kiện chính

Tác nhân	Hệ thống
1. Sau khi cập nhật thông tin cá nhân thành công, vào trang tạo thực đơn rồi chọn số bữa trong một ngày, sau đó nhấn nút “Tạo thực đơn”	2. Hiện thị thực đơn cho cả ngày bao gồm các bữa người dùng đã yêu cầu
3. Người dùng nhấn nút “Thay đổi” ở góc trên phải của bữa	4. Hệ thống tạo ra một tổ hợp món ăn mới cho bữa đó
5. Người dùng nhấn nút “Thay đổi” ở dưới mỗi món ăn	6. Hệ thống sẽ gợi ý những món ăn có mức độ dinh dưỡng gần giống với món ăn thay đổi
7. Người dùng nhấn nút “Xóa” ở dưới món ăn	8. Món ăn trong bữa sẽ bị ẩn đi
9. Người dùng nhấn nút “Thêm món”	10. Hệ thống sẽ mở ra cửa sổ để người dùng tìm kiếm món ăn theo các tiêu chí nguyên liệu, dinh dưỡng.
11. Người dùng nhấn nút chọn giờ cho mỗi bữa ăn	
12. Người dùng nhấn lưu thực đơn	13. Thực đơn sẽ được lưu lại trong phần thực đơn đã lưu và phần tiến độ
Luồng sự kiện thay thế	
11.1 Người dùng không chọn giờ ăn cho mỗi bữa	11.2 Hệ thống sẽ không gửi được thông báo nếu người dùng chưa hoàn thành bữa ăn

*Bảng 3.3: Đặc tả usecase quản lí thực đơn*

### 3.5.3. Đặc tả usecase quản lý tiến độ

Tên usecase: Quản lý tiến độ	
Mô tả: Cho phép người dùng theo dõi tiến độ tuân thủ	
Tác nhân: User	
Tiền điều kiện: Người dùng đã tạo thực đơn và lưu thực đơn	
Hậu điều kiện: Tiến độ dinh dưỡng được cập nhật	
Luồng sự kiện chính	
Tác nhân	Hệ thống
1. Sau khi người dùng tạo và lưu thực đơn thành công, vào phần tiến độ	2. Hiện thị lịch thực đơn, biểu đồ tiến độ dinh dưỡng trong ngày
3. Tại mỗi bữa ăn, người dùng nhấn nút “Hoàn thành”	4. Hệ thống đánh dấu bữa ăn đã hoàn thành bữa ăn và cập nhật lại biểu đồ dinh dưỡng trong ngày
4. Người dùng nhấn nút “Theo dõi tiến độ” rồi nhấn nút “Theo tuần” hoặc “Theo tháng”	5. Hệ thống hiện thị biểu đồ tiến độ tuân thủ theo tuần hoặc theo tháng
Luồng sự kiện thay thế	
3.1 Người dùng nhấn nút “Hoàn thành”	3.2 Hệ thống sẽ không cập nhật tiến độ dinh dưỡng trong ngày và sẽ gửi thông báo nhắc nhở cho người dùng

*Bảng 3.4: Đặc tả usecase quản lý tiến độ*

### 3.5.4. Đặc tả usecase quản lý món ăn

Tên usecase: Quản lý món ăn
-----------------------------

Mô tả: Cho phép admin thêm, cập nhật, tìm kiếm, xóa món ăn trong hệ thống	
Tác nhân: Admin	
Tiền điều kiện: Đã đăng nhập tài khoản admin	
Hậu điều kiện: Tạo danh sách món ăn cho hệ thống	
Luồng sự kiện chính	
Tác nhân	Hệ thống
1. Admin vào trang quản lý món ăn và nhấn nút thêm món ăn	2. Hiển thị form để nhập dữ liệu cho món ăn
3. Admin nhấn nút “Lưu” món ăn	4. Hệ thống thêm món ăn vào danh sách món ăn
5. Admin chọn món ăn nhấn nút “Xem”	6. Hệ thống hiển thị chi tiết món ăn
7. Admin chọn nút “Sửa”	8. Hệ thống hiển thị form chỉnh sửa thông tin món ăn
9. Admin nhấn nút “Xóa”	10. Hệ thống sẽ xóa món ăn ra khỏi danh sách
Luồng sự kiện thay thế	
5.1. Admin chọn nút đóng	5.2. Hệ thống sẽ đóng thông tin của món ăn
7.1. Admin chọn nút “Lưu”	7.2. Món ăn sẽ được cập nhật trong danh sách món ăn

*Bảng 3.5: Đặc tả usecase quản lý món ăn*

### 3.5.5. Đặc tả usecase quản lý tài khoản

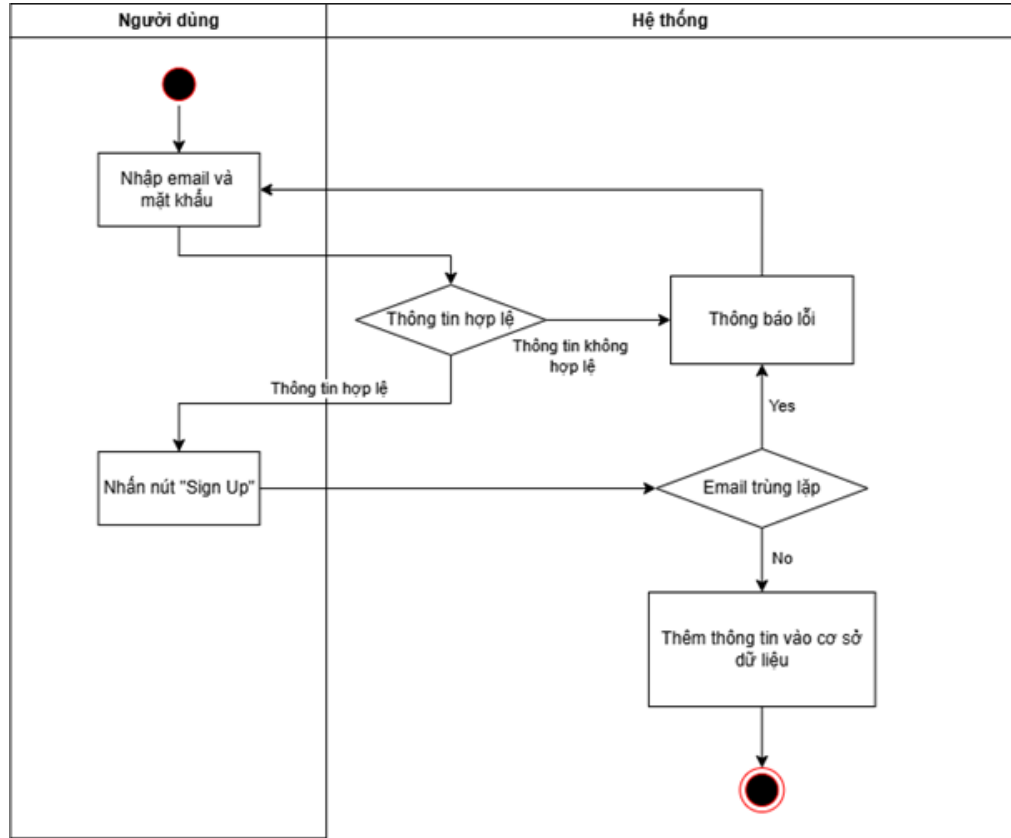
Tên usecase: Quản lý tài khoản	
Mô tả: Cho phép admin xem, sửa, xóa tài khoản	
Tác nhân: Admin	
Tiền điều kiện: Đã đăng nhập tài khoản admin	
Hậu điều kiện: Quản lý tài khoản người dùng	
Luồng sự kiện chính	
Tác nhân	Hệ thống
1. Admin vào trang quản lý người dùng	2. Hệ thống hiển thị danh sách tài khoản người dùng
3. Admin chọn tài khoản người dùng rồi nhấn nút “Xem”	4. Hệ thống hiển thị chi tiết thông tin tài khoản người dùng
5. Admin nhấn nút sửa để cập nhật reset mật khẩu mới cho tài khoản	6. Hệ thống cập nhật thông tin tài khoản
7. Admin chọn tài khoản rồi nhấn nút “Xóa”	8. Hệ thống xóa tài khoản khỏi hệ thống
Luồng sự kiện thay thế	
3.1. Admin chọn nút đóng	5.2. Hệ thống sẽ đóng thông tin tài khoản
7.1. Admin chọn nút “Hủy”	7.2. Hệ thống hủy xóa tài khoản

*Bảng 3.6: Đặc tả usecase quản lý tài khoản*

### 3.6. Sơ đồ hoạt động

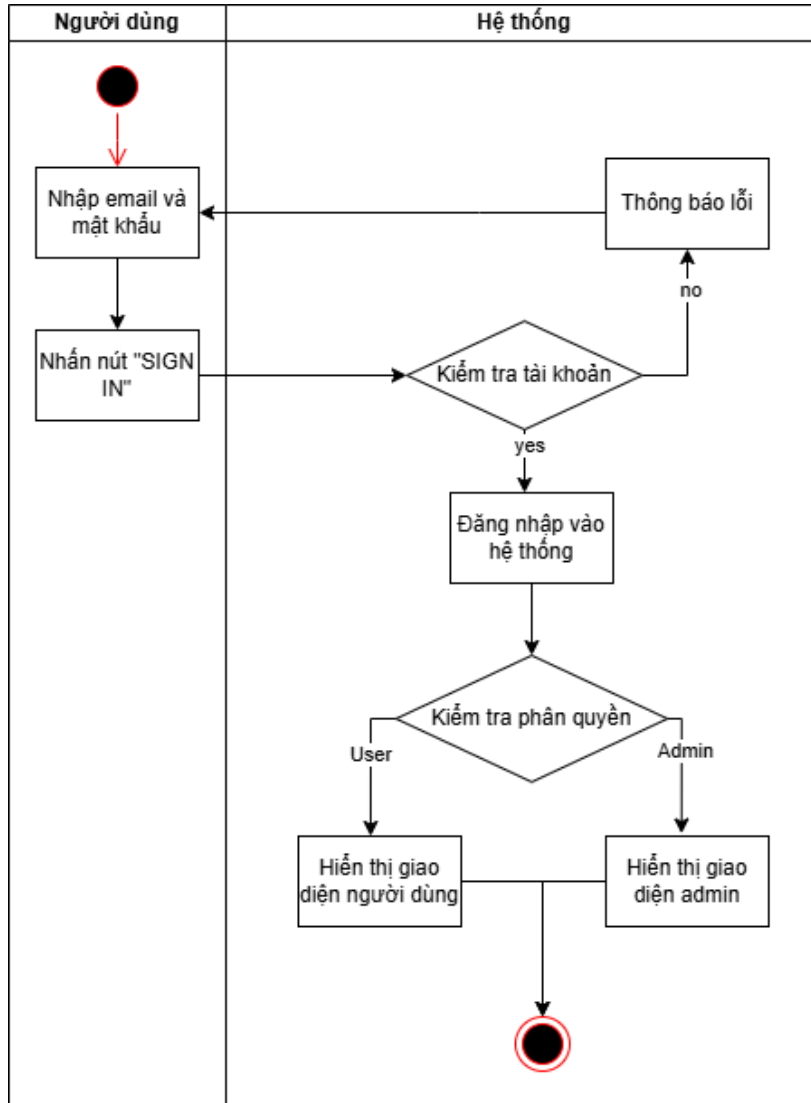
#### 3.6.1. Sơ đồ hoạt động của chức năng đăng ký, đăng nhập, quên mật khẩu

##### 3.6.1.1. Sơ đồ hoạt động chức năng đăng ký(user)



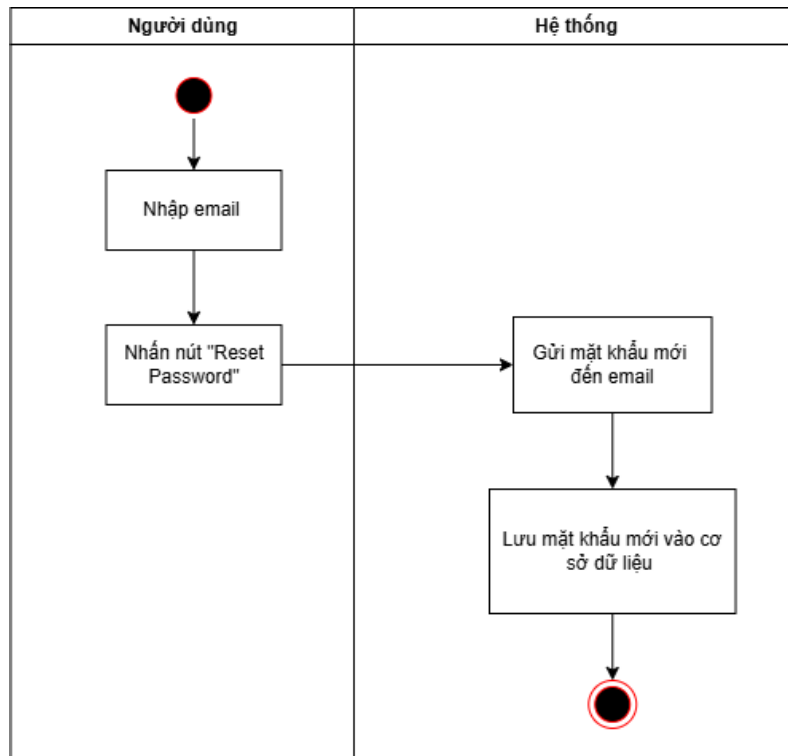
*Hình 3.7: Sơ đồ hoạt động chức năng đăng ký*

### 3.6.1.2. Sơ đồ hoạt động cho chức năng đăng nhập



Hình 3.8: Sơ đồ hoạt động chức năng đăng nhập

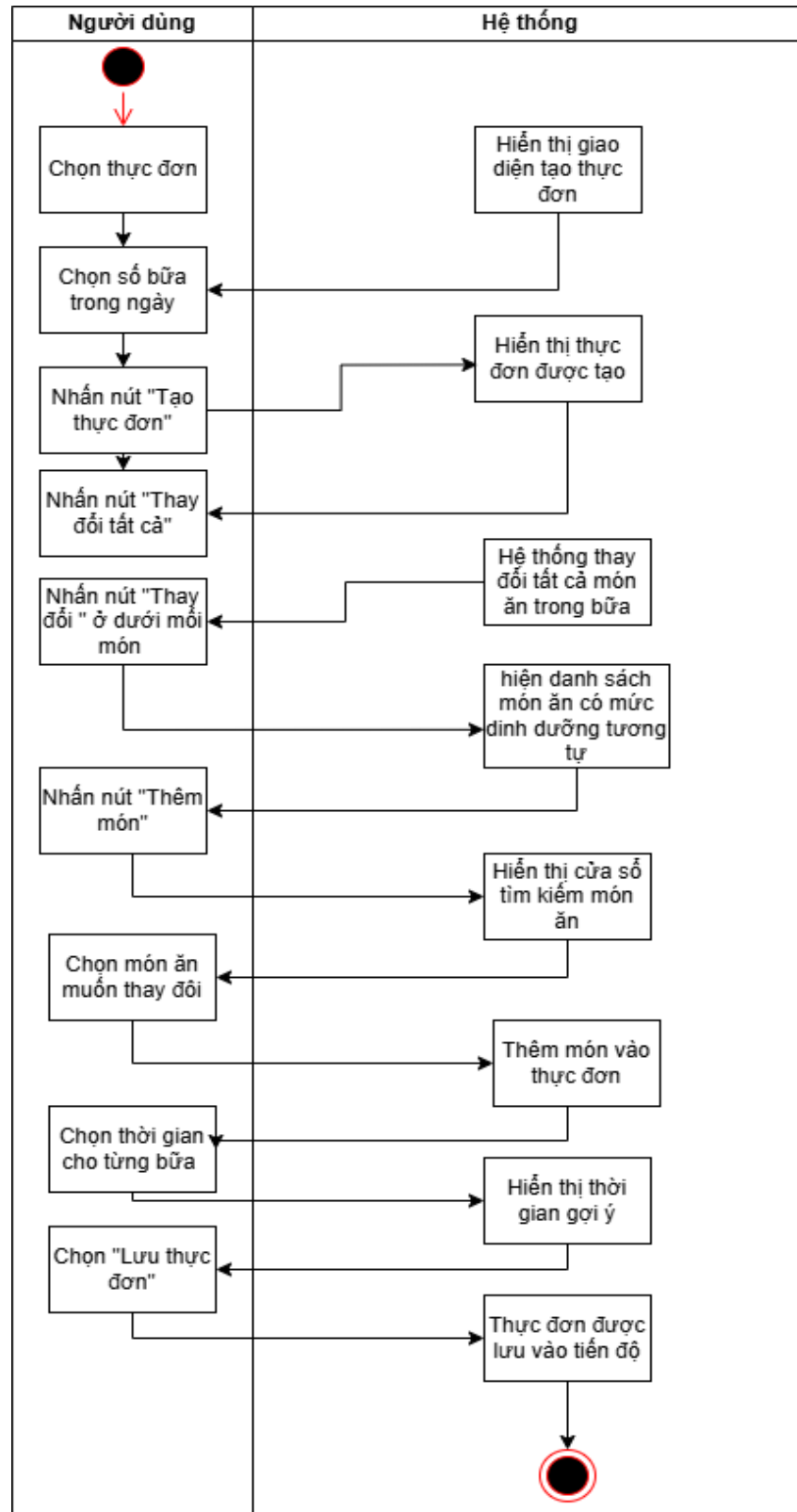
### 3.6.1.3. Sơ đồ hoạt động cho chức năng quên mật khẩu



*Hình 3.9: Sơ đồ hoạt động chức năng quên mật khẩu*

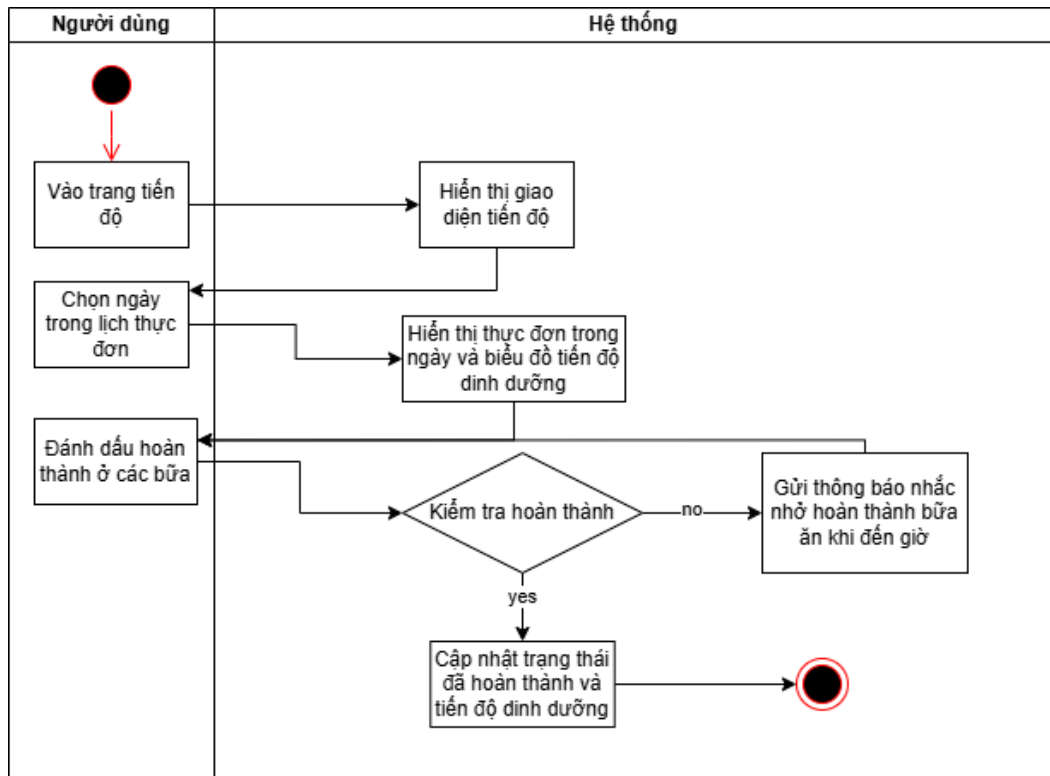
### 3.6.2. Sơ đồ hoạt động của hệ thống client

#### 3.6.2.1. Sơ đồ hoạt động cho chức năng tạo thực đơn



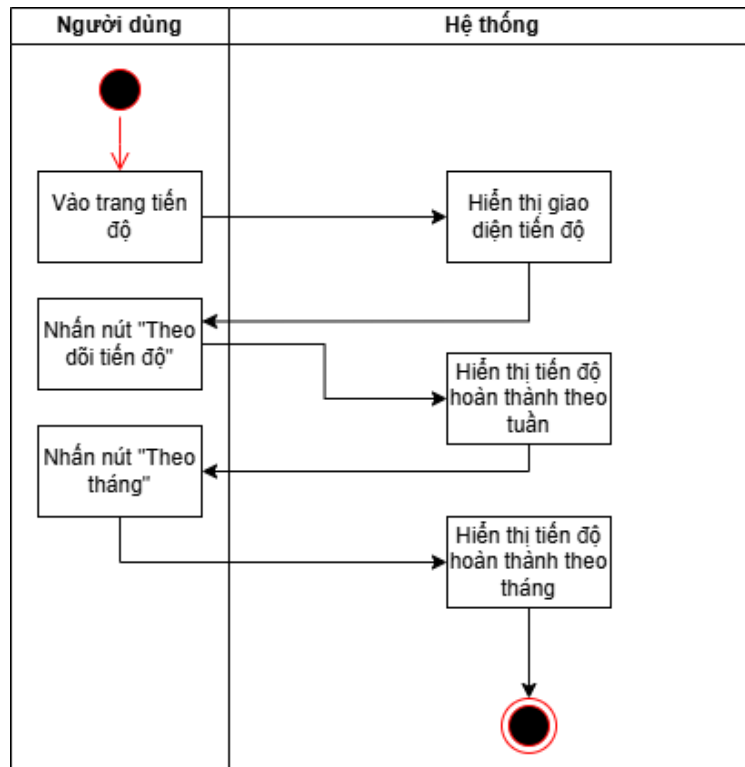
Hình 3.10: Sơ đồ hoạt động chức năng tạo thực đơn

### 3.6.2.2. Sơ đồ hoạt động của chức năng đánh dấu hoàn thành



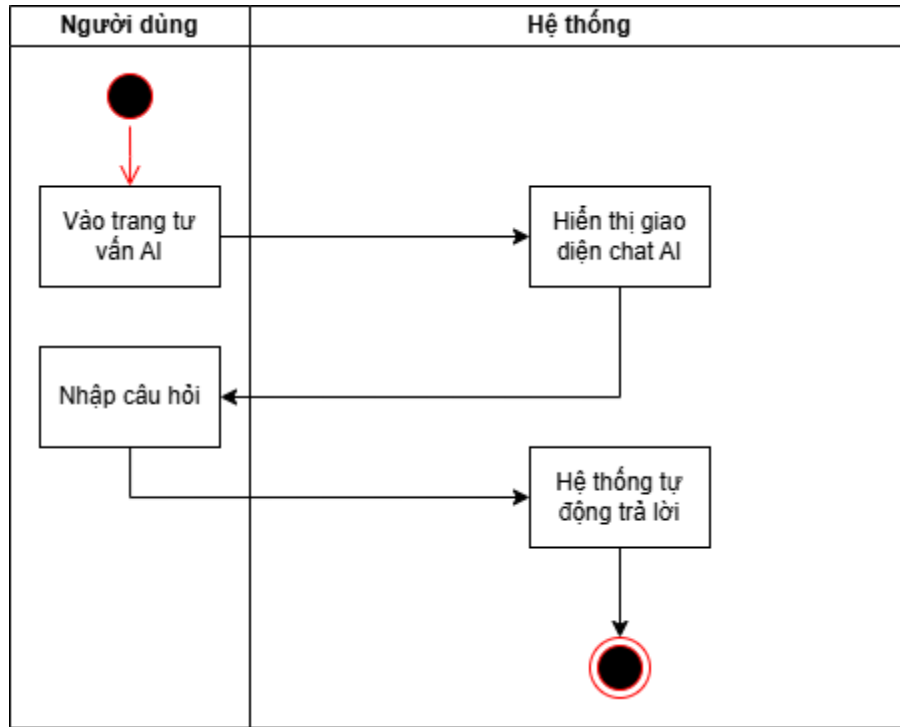
Hình 3.11: Sơ đồ hoạt động chức năng đánh dấu hoàn thành

### 3.6.2.3. Sơ đồ hoạt động chức năng theo dõi tiến độ



Hình 3.12: Sơ đồ hoạt động chức năng theo dõi tiến độ

### 3.6.2.4. Sơ đồ hoạt động của chức năng tư vấn AI



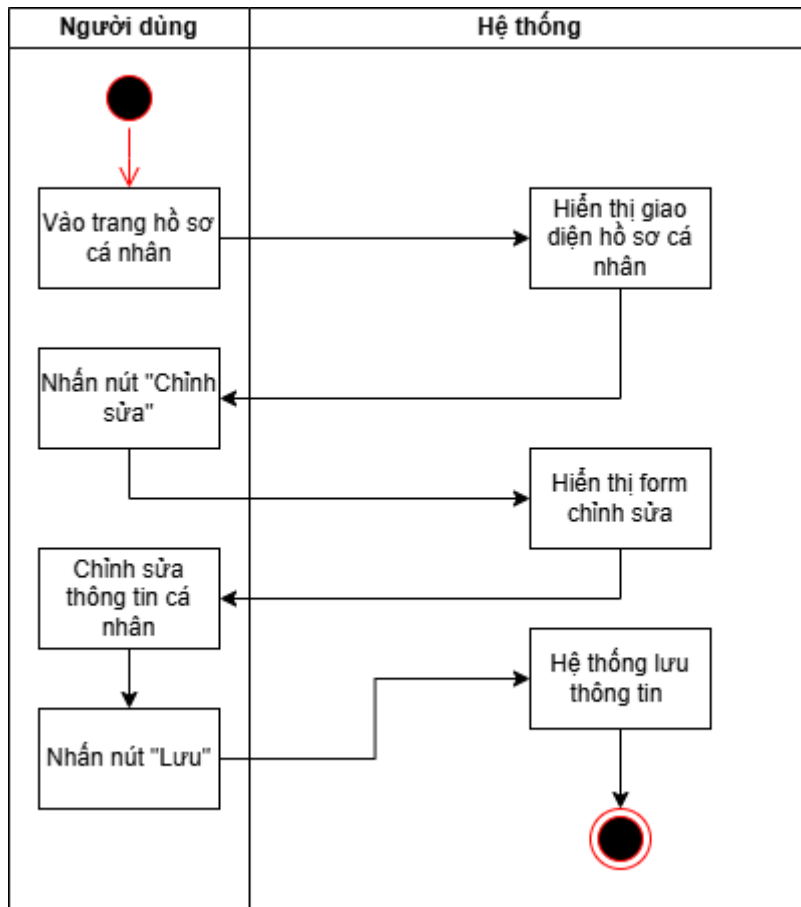
Hình 3.13: Sơ đồ hoạt động chức năng tư vấn AI

### 3.6.2.5. Sơ đồ hoạt động của chức năng mua sắm



Hình 3.14: Sơ đồ hoạt động chức năng mua sắm

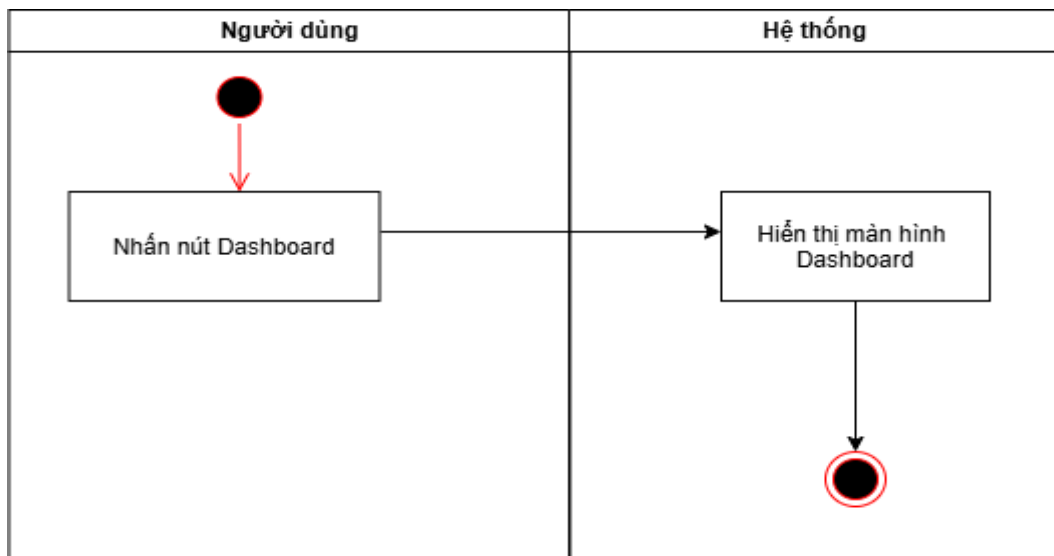
### 3.6.3. Sơ đồ hoạt động của chức năng chỉnh sửa thông tin cá nhân



Hình 3.15: Sơ đồ hoạt động chức năng chỉnh sửa thông tin cá nhân

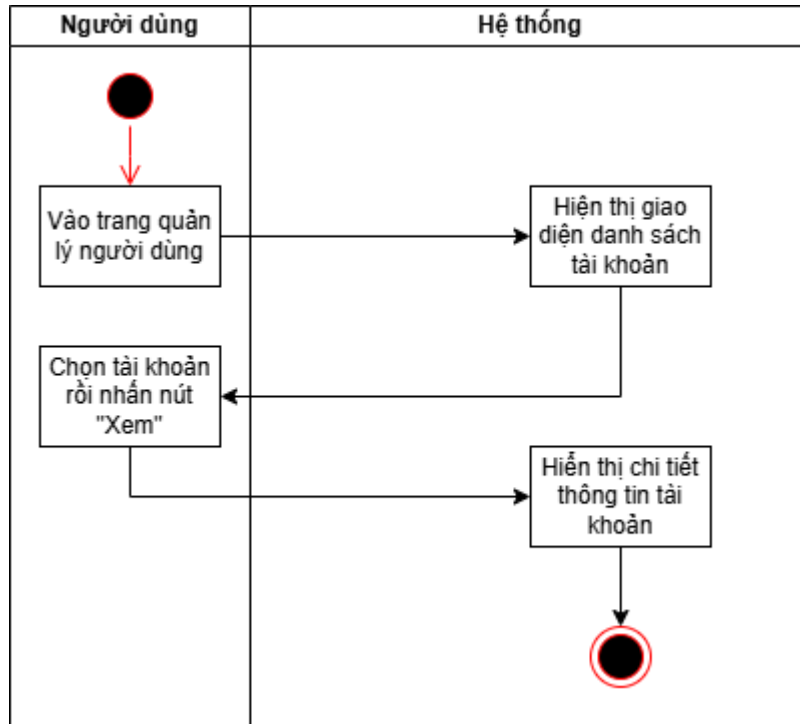
### 3.6.4. Sơ đồ hoạt động của hệ thống admin

#### 3.6.4.1. Sơ đồ hoạt động hiển thị màn Dashboard



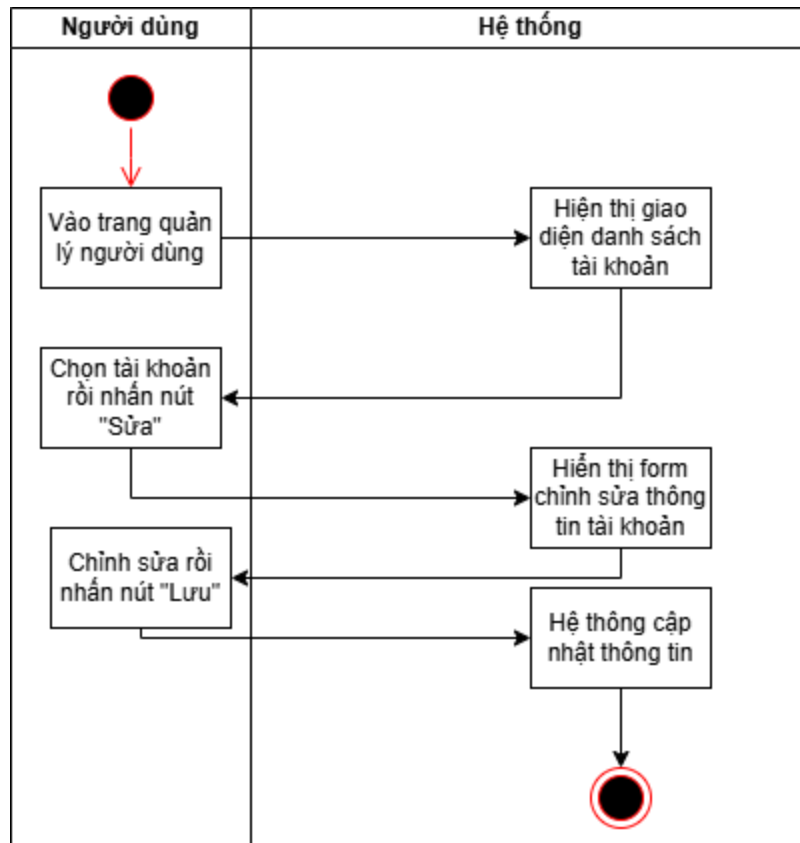
Hình 3.16: Sơ đồ hoạt động chức năng hiển thị Dashboard

### 3.6.4.2. Sơ đồ hoạt động chức năng xem thông tin tài khoản



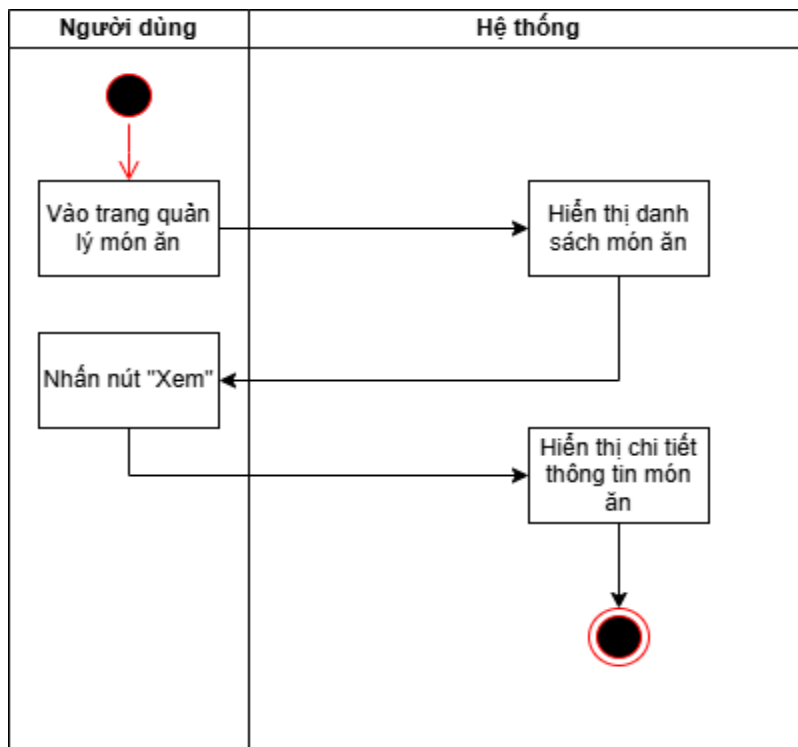
Hình 3.17: Sơ đồ hoạt động chức năng xem thông tin tài khoản

### 3.6.4.3. Sơ đồ hoạt động chức năng sửa tài khoản



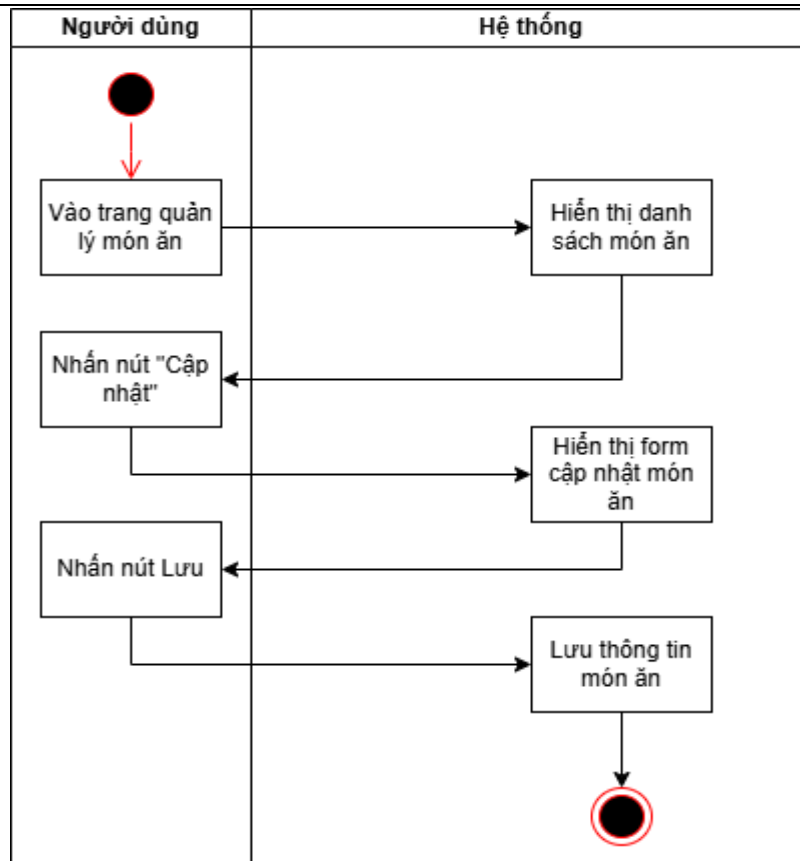
Hình 3.18: Sơ đồ hoạt động chức năng sửa tài khoản

### 3.6.4.4. Sơ đồ hoạt động chức năng xem chi tiết món ăn



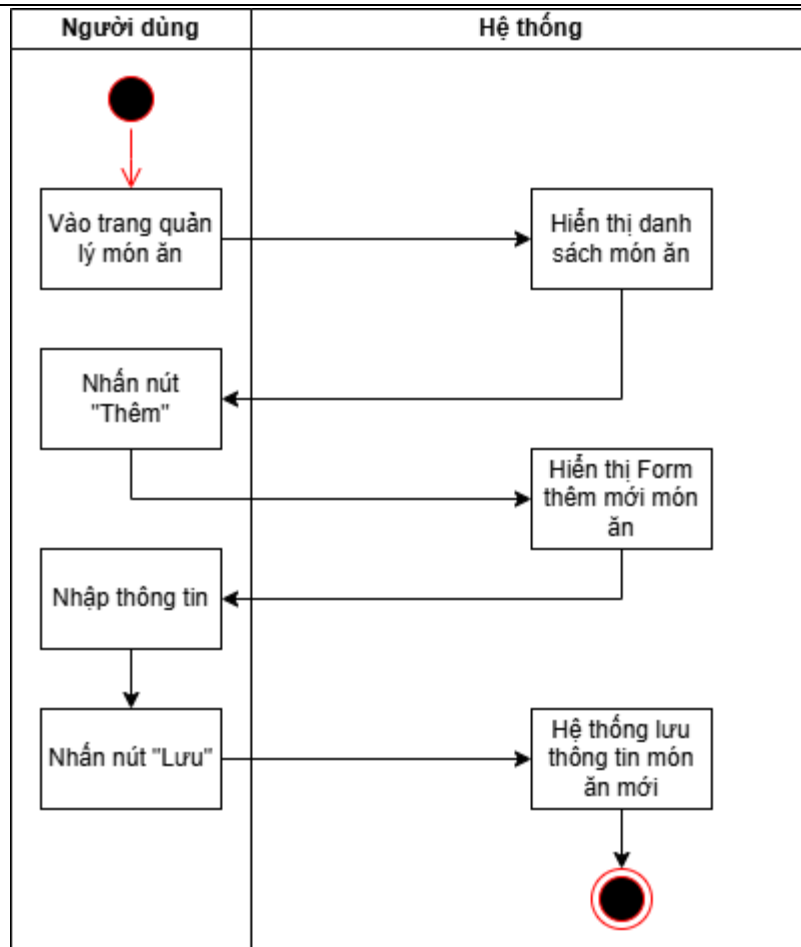
*Hình 3.19: Sơ đồ hoạt động chức năng xem chi tiết món ăn*

### 3.6.4.5. Sơ đồ hoạt động cập nhật món ăn



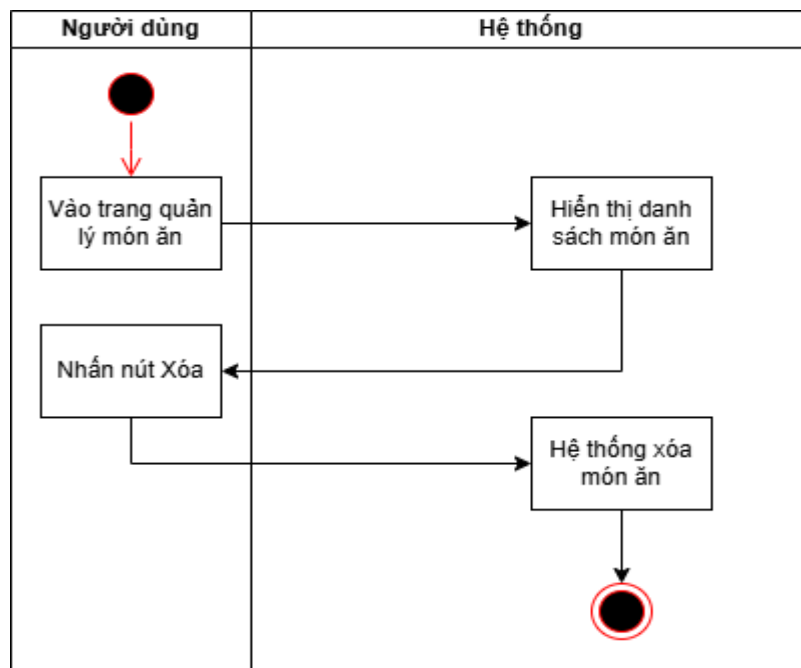
*Hình 3.20: Sơ đồ hoạt động chức năng cập nhật món ăn*

#### **3.6.4.6. Sơ đồ hoạt động thêm món ăn**



Hình 3.21: Sơ đồ hoạt động chức năng thêm món ăn

### 3.6.4.7. Sơ đồ hoạt động xóa món ăn

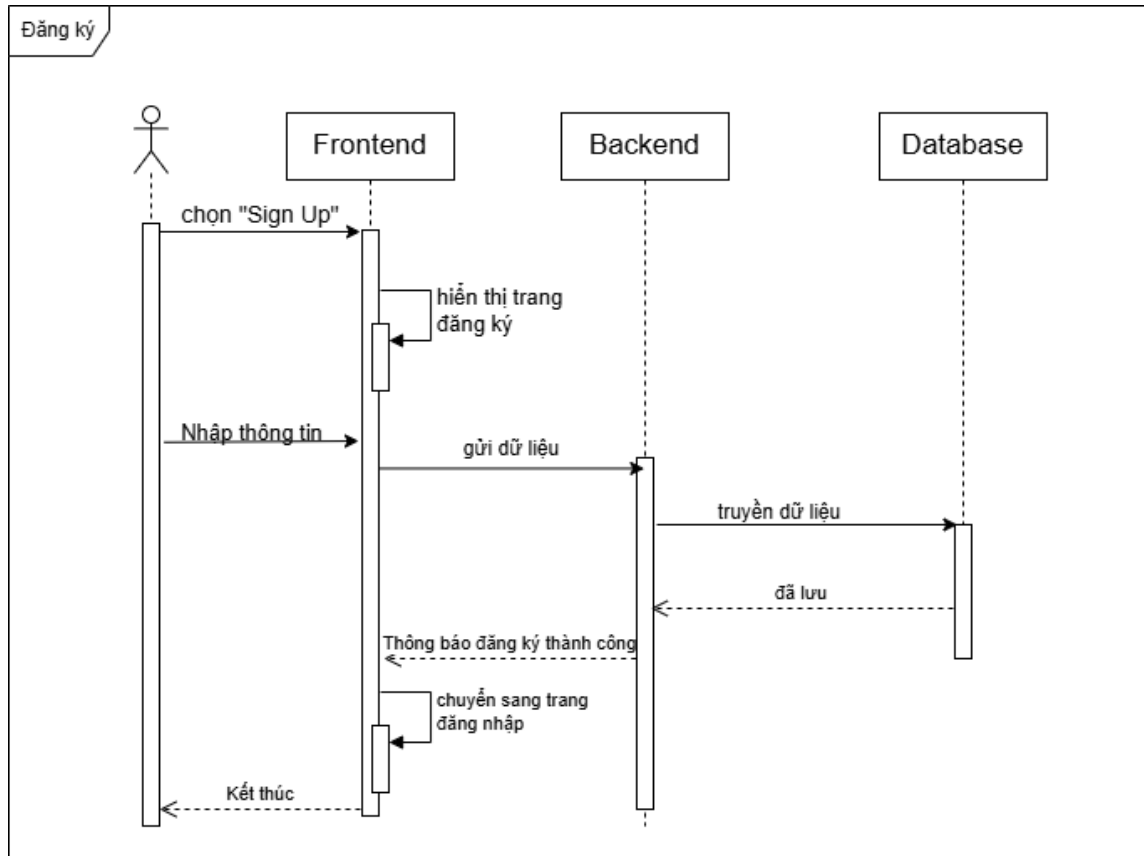


Hình 3.22: Sơ đồ hoạt động chức năng xóa món ăn

### 3.7. Sơ đồ tuần tự

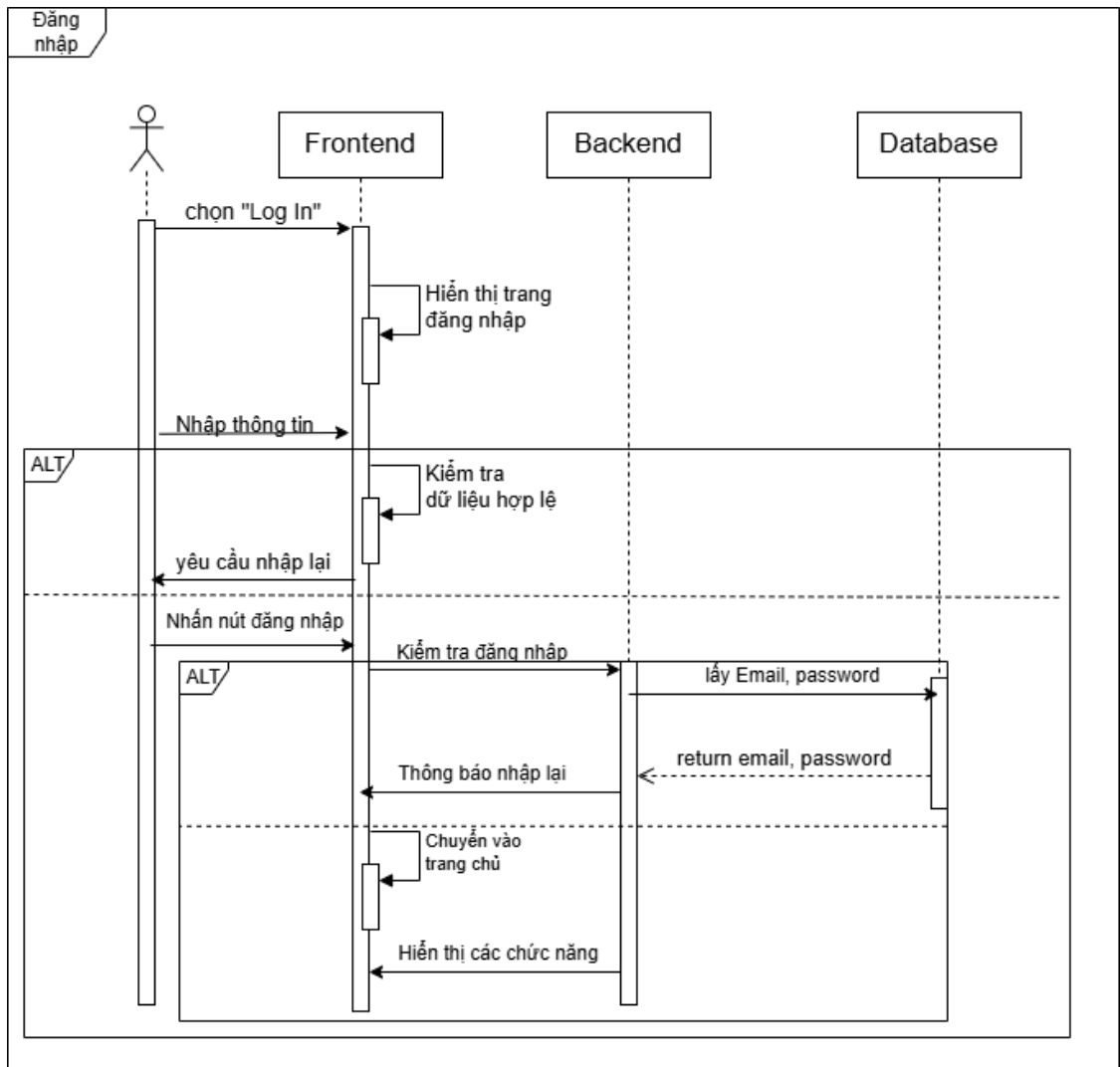
#### 3.7.1. Sơ đồ tuần tự cho chức năng đăng nhập, đăng ký

##### 3.7.1.1. Sơ đồ tuần tự cho chức năng đăng ký



*Hình 3.23: Sơ đồ tuần tự chức năng đăng ký*

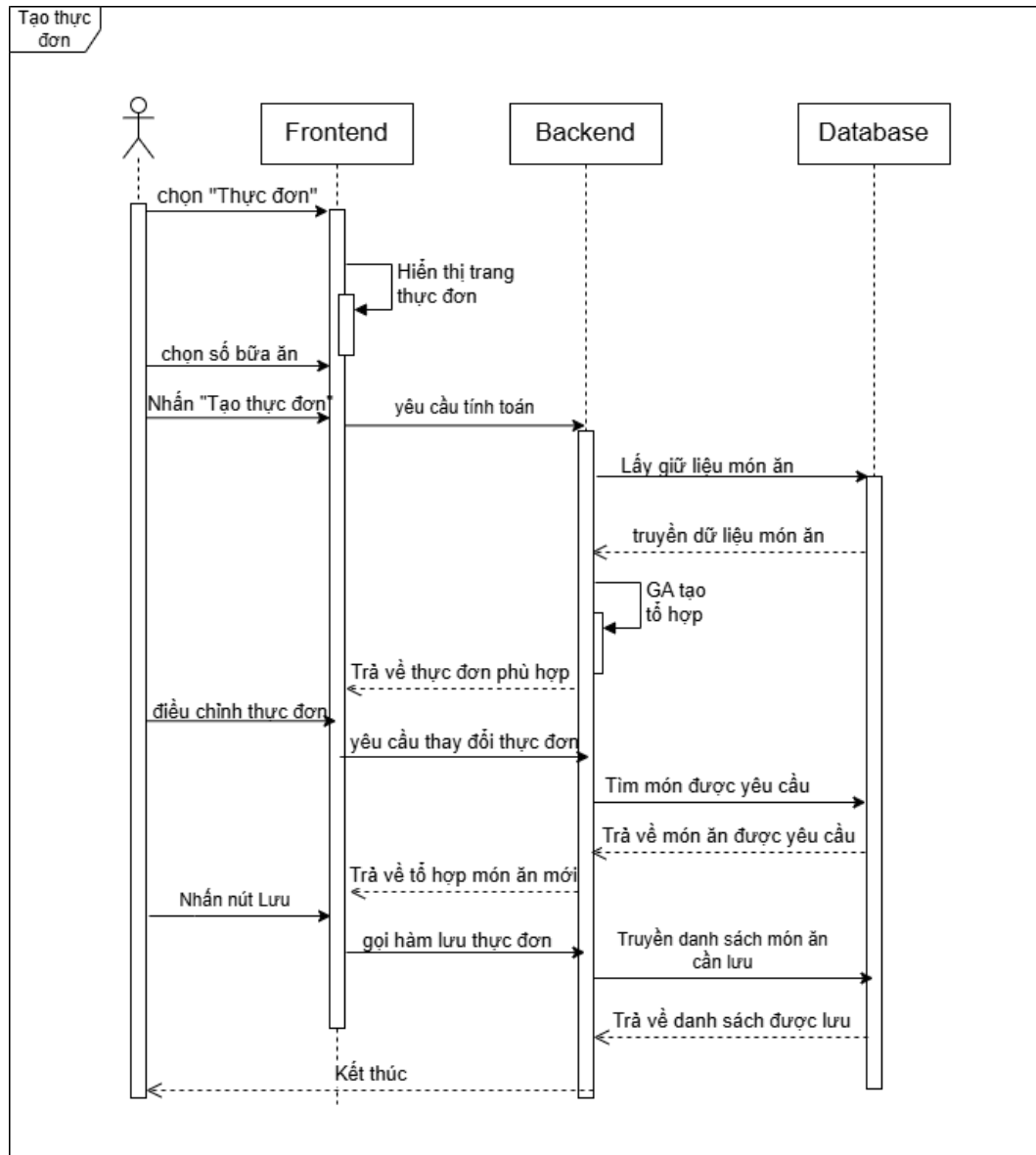
### 3.7.1.2. Sơ đồ tuần tự cho chức năng đăng nhập



Hình 3.24: Sơ đồ tuần tự chức năng đăng nhập

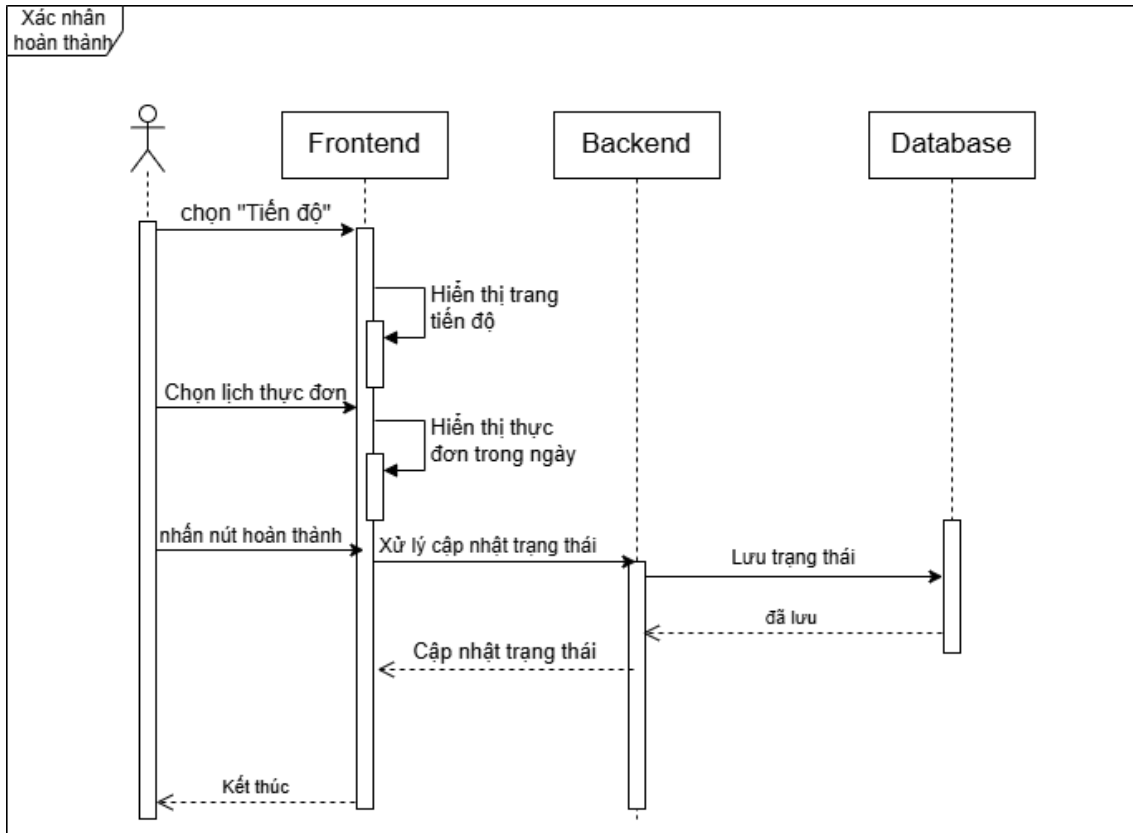
### 3.7.2. Sơ đồ tuần tự cho hệ thống client

#### 3.7.2.1. Sơ đồ tuần tự cho chức năng tạo thực đơn



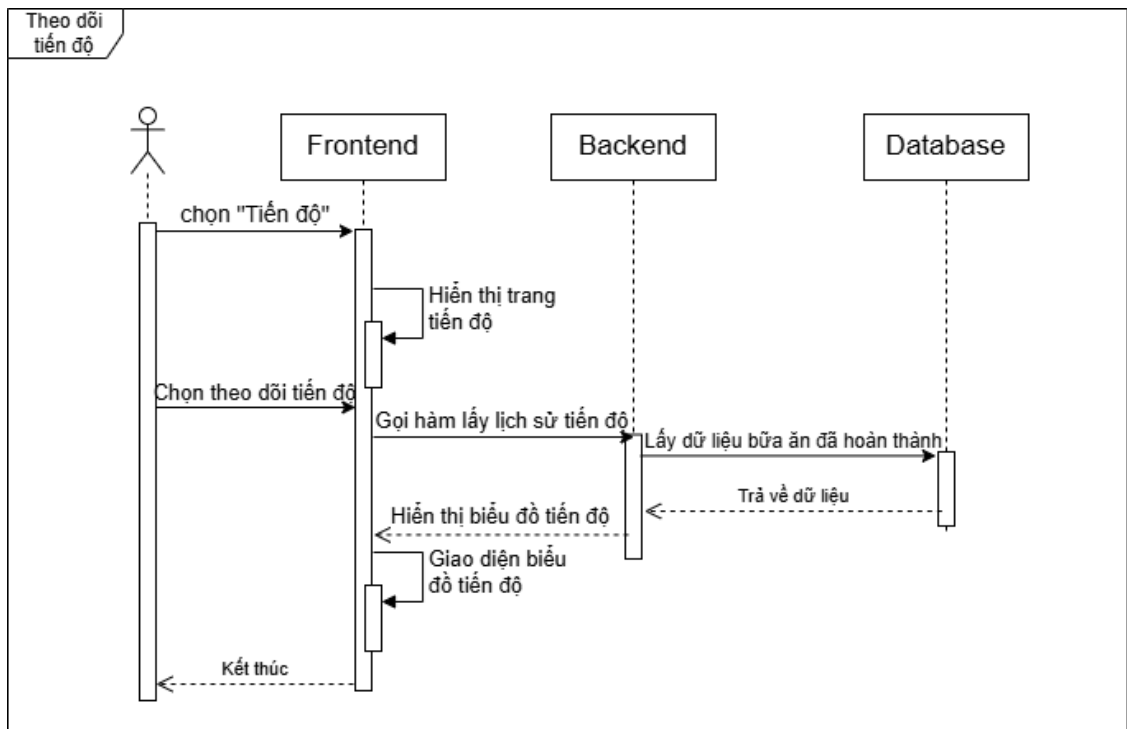
Hình 3.25: Sơ đồ tuần tự chức năng tạo thực đơn

### 3.7.2.2. Sơ đồ tuần tự cho chức năng đánh dấu hoàn thành



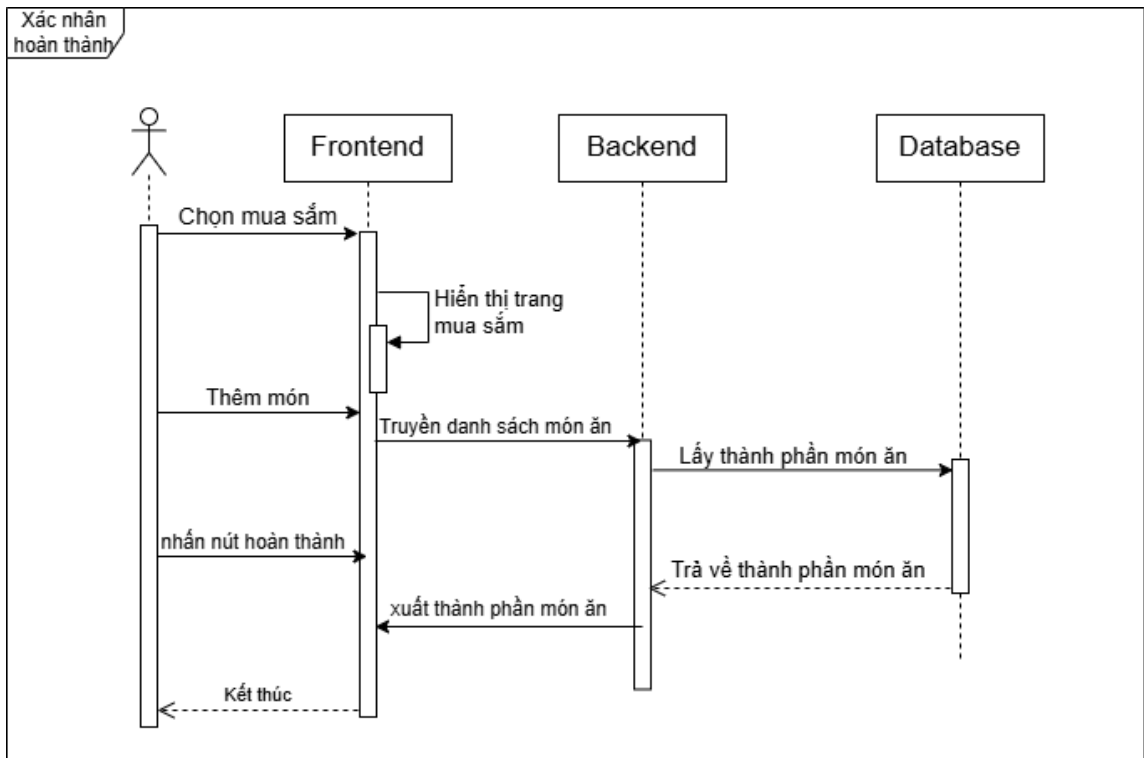
Hình 3.26: Sơ đồ tuần tự chức năng đánh dấu hoàn thành

### 3.7.2.3. Sơ đồ tuần tự cho chức năng theo dõi tiến độ



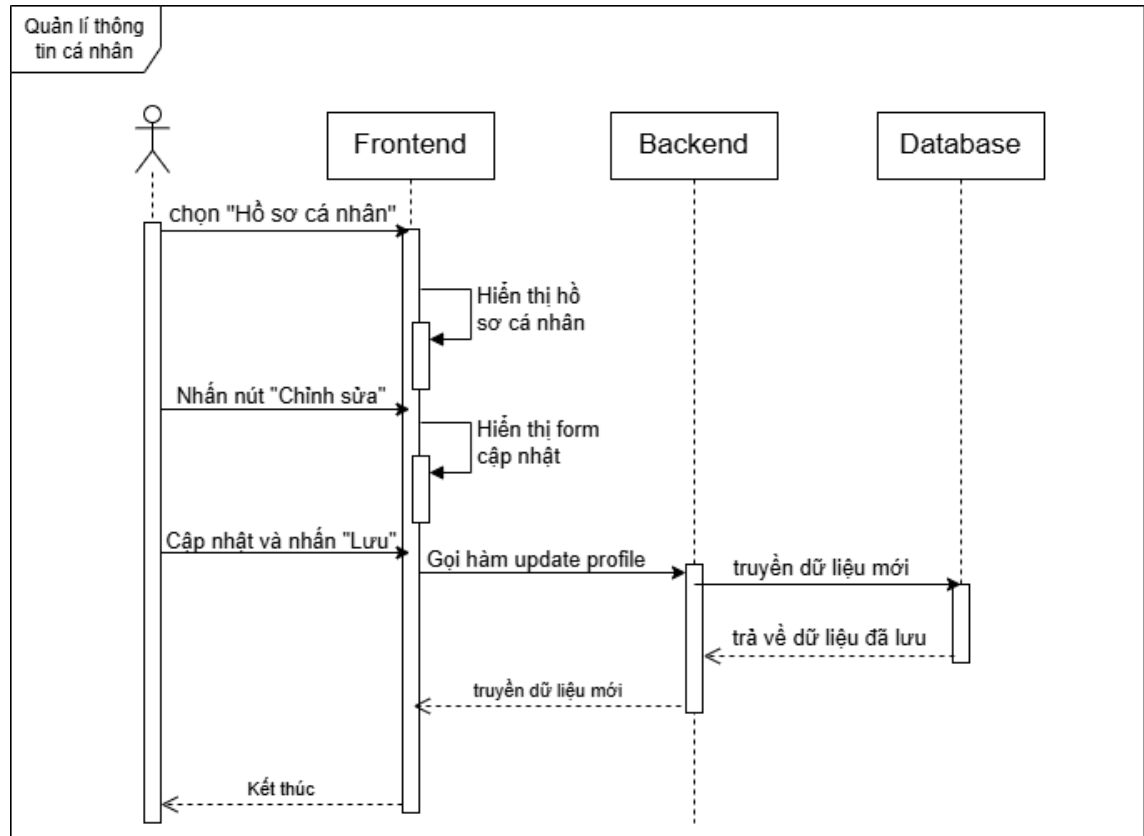
Hình 3.27: Sơ đồ tuần tự chức năng theo dõi tiến độ

### 3.7.2.4. Sơ đồ tuần tự cho chức năng mua sắm



Hình 3.28: Sơ đồ tuần tự chức năng mua sắm

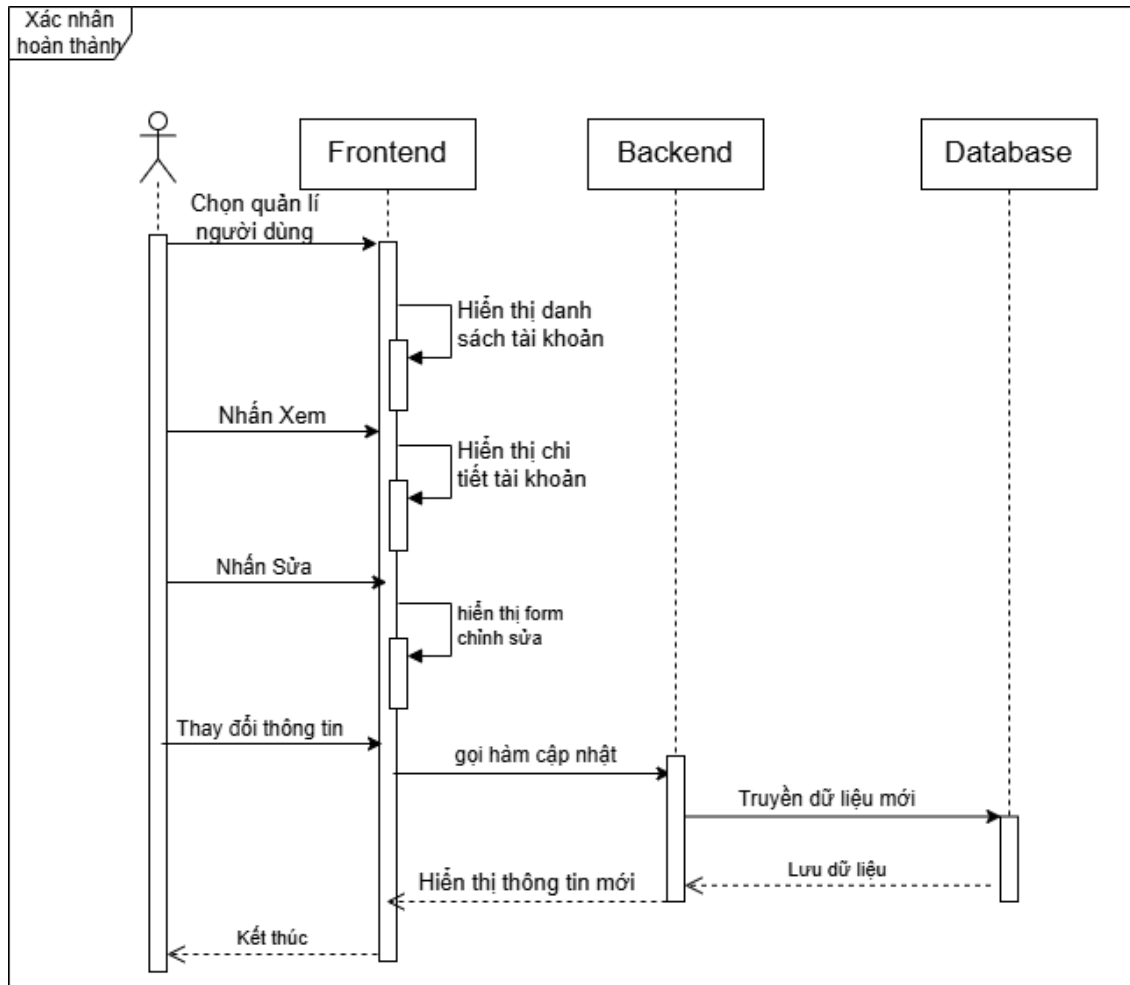
### 3.7.2.5. Sơ đồ tuần tự cho chức năng quản lý thông tin cá nhân



Hình 3.29: Sơ đồ tuần tự chức năng quản lý thông tin cá nhân

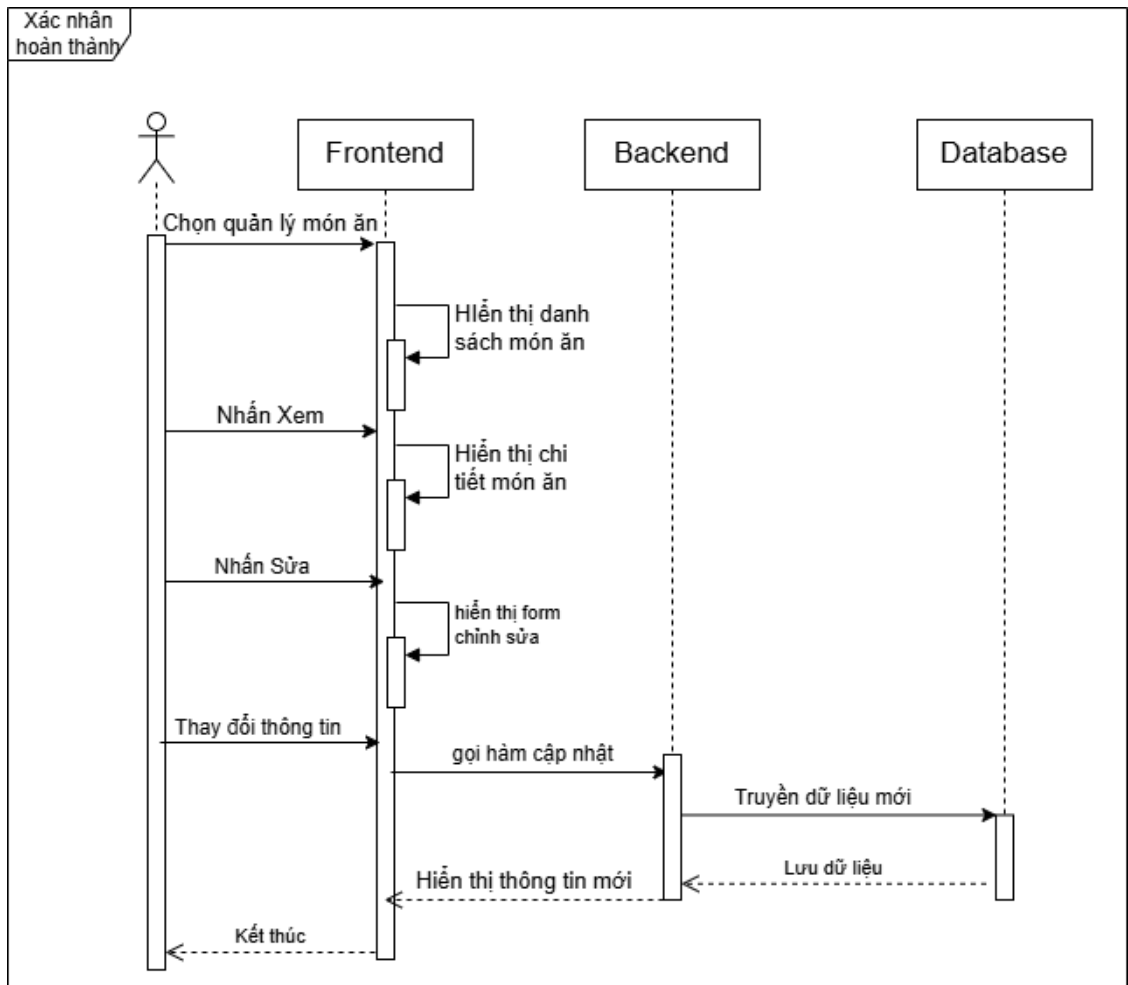
### 3.7.3. Sơ đồ tuần tự cho hệ thống admin

#### 3.7.3.1. Sơ đồ tuần tự cho chức năng quản lý tài khoản



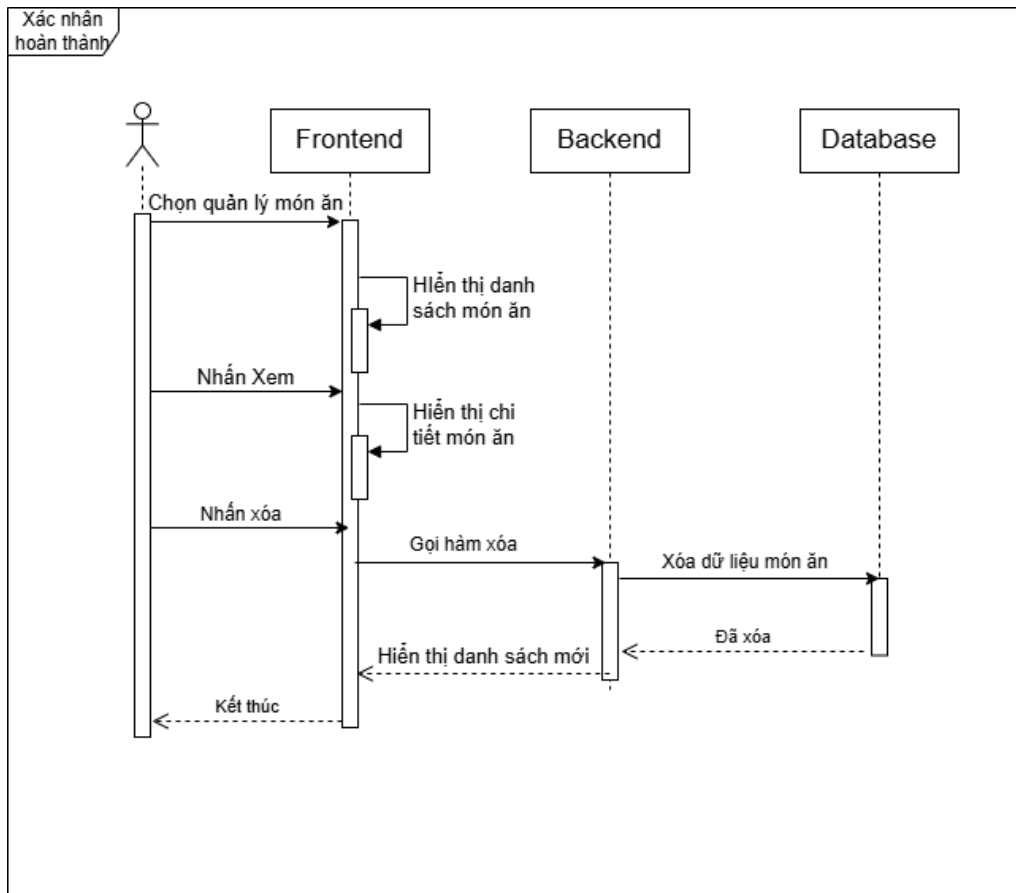
*Hình 3.30: Sơ đồ tuần tự chức năng quản lý tài khoản*

### 3.7.3.2. Sơ đồ tuần tự cho chức năng quản lý món ăn



Hình 3.31: Sơ đồ tuần tự chức năng quản lý món ăn

### 3.7.3.3. Sơ đồ tuần tự cho chức năng xóa món ăn



Hình 3.32: Sơ đồ tuần tự chức năng xóa món ăn

## 3.8. Cơ sở dữ liệu

### 3.8.1. Collection User

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	ID duy nhất của người dùng.
2	email	String	Email đăng nhập, duy nhất.
3	password	String	Mật khẩu đã mã hóa.
4	name	String	Tên người dùng.
5	avatar	String	Đường dẫn ảnh đại diện.

6	nutritionProfile	Object	Thông tin dinh dưỡng chi tiết.
7	isComplete	Boolean	Trạng thái hoàn thành hồ sơ.
8	age	Number	Tuổi, dùng tính calo.
9	weight	Number	Cân nặng (kg), tính calo.
10	height	Number	Chiều cao (cm), tính BMI.
11	gender	String	Giới tính, ảnh hưởng calo.
12	activityLevel	String	Mức độ hoạt động, điều chỉnh calo.
13	goals	String	Mục tiêu (giảm cân, tăng cơ).
14	medicalConditions	Array	Điều kiện y tế, ảnh hưởng hạn chế.
15	preferences	Array	Sở thích, ưu tiên món ăn.
16	retrictions	Array	Hạn chế, loại bỏ món không phù hợp.
17	dailyCalorieNeeds	Number	Nhu cầu calo hàng ngày.
18	macroRatio	Object	Tỷ lệ macronutrients.

*Bảng 3.7: Collection User*

### 3.8.2. Collection Food

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	ID duy nhất của món ăn.
2	name	String	Tên món ăn, dùng để hiển thị.
3	image	String	Đường dẫn ảnh minh họa món ăn.
4	calories	Number	Số calo của món ăn.
5	protein	Number	Lượng protein (g).
6	cabs	Number	Lượng carbohydrate (g).
7	fat	Number	Lượng chất béo (g).
8	preferences	Array	Sở thích liên quan
9	restrictions	Array	Hạn chế liên quan (như "no_dairy").
10	context	Object	Thông tin ngữ cảnh (bữa sáng/tối).
11	ingredients	Array	Danh sách nguyên liệu.
12	instructions	Array	Hướng dẫn chế biến.

*Bảng 3.8: Collection Food*

### 3.8.3. Collection SaveMenu

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	ID duy nhất của thực đơn.
2	userId	String	ID người dùng sở hữu thực đơn.
3	date	String	Ngày áp dụng thực đơn.
4	meals	Array	Danh sách các bữa ăn trong ngày.
5	mealName	String	Tên bữa ăn
6	foods	Array	Danh sách món ăn trong bữa.
7	totalNutrition	Object	Tổng dinh dưỡng (calo, protein, v.v.).
8	note	String	Ghi chú bổ sung cho thực đơn.

*Bảng 3.9: Collection SaveMenu*

### 3.8.4. Collection Notifications

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	ID duy nhất của thông báo.
2	userId	String	ID người dùng nhận thông báo.
3	type	String	Loại thông báo (nhắc nhở, cảnh báo).

4	title	String	Tiêu đề thông báo.
5	content	String	Nội dung chi tiết của thông báo.
6	createAt	Date	Thời gian tạo thông báo.
7	read	Boolean	Trạng thái đã đọc (true/false).
8	date	String	Ngày liên quan đến thông báo.
9	mealName	String	Tên bữa ăn liên quan (nếu có).

*Bảng 3.10: Collection Notifications*

### 3.8.5. Collection ComplianceHistory

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	ID duy nhất của bản ghi tiến độ.
2	userId	String	ID người dùng liên quan.
3	date	Number	Ngày ghi nhận tiến độ (timestamp).
4	compliance	Number	Tỷ lệ tuân thủ thực đơn (0-1).
5	mealsCompleted	Number	Số bữa ăn hoàn thành.
6	totalMeals	Number	Tổng số bữa ăn trong ngày.
7	streak	Number	Số ngày tuân thủ liên tiếp.
8	achievements	Array	Danh sách thành tích đạt được.

9	note	String	Ghi chú bổ sung về tiến độ.
10	createAt	Date	Thời gian tạo bản ghi.

*Bảng 3.11: Collection ComplianceHistory*

### **3.9. Kết luận chương**

Chương Phân tích và thiết kế hệ thống đã trình bày toàn diện quá trình phân tích và thiết kế hệ thống quản lý thực đơn cá nhân hóa. Phần yêu cầu chức năng và phi chức năng đã xác định rõ các nhu cầu cốt lõi như quản lý người dùng, tạo thực đơn tự động bằng thuật toán Di truyền (GA), theo dõi tiến độ, và tích hợp tính năng thông minh, đồng thời đảm bảo hiệu suất, bảo mật, khả năng mở rộng, giao diện thân thiện, và độ tin cậy. Phần thiết kế hệ thống đã mô tả kiến trúc tổng quan, tập trung vào các module chính và luồng xử lý dữ liệu từ người dùng đến thực đơn tối ưu. Thiết kế cơ sở dữ liệu đã cung cấp cấu trúc chi tiết cho các collection như User, Food, SaveMenu, ComplianceHistory, và Notifications, hỗ trợ lưu trữ và quản lý thông tin hiệu quả. Cuối cùng, thiết kế giao diện và kết luận chương đã đặt nền tảng cho việc triển khai thực tế.

## CHƯƠNG 4: TRIỂN KHAI VÀ KẾT QUẢ

### 4.1. Môi trường phát triển

#### 4.1.1. Môi trường lưu trữ và quản lý nguồn

- Hệ thống được triển khai trên máy chủ cục bộ với cấu hình tối thiểu: CPU Intel Core i5, RAM 8GB, và ổ cứng SSD 256GB để đảm bảo hiệu suất xử lý thuật toán GA.
- Nguồn code và tài liệu được quản lý trên GitHub, sử dụng nhánh master để lưu trữ phiên bản chính thức và nhánh dev để phát triển thử nghiệm, đảm bảo kiểm soát phiên bản hiệu quả.
- Dữ liệu món ăn và thông tin người dùng được lưu trữ trên MongoDB, chạy trên localhost:27017, với cơ chế sao lưu định kỳ để tránh mất mát dữ liệu.

#### 4.1.2. Môi trường phát triển dự án

- Công cụ phát triển: Visual Studio Code được chọn làm IDE chính, tích hợp các tiện ích mở rộng như Python, Node.js, và MongoDB Compass để hỗ trợ lập trình và quản lý cơ sở dữ liệu.
- Quản lý gói: npm cho Node.js và pip cho Python, đảm bảo cài đặt và cập nhật các thư viện như Express.js, FastAPI, NumPy, và Pydantic một cách dễ dàng.

### 4.2. Công nghệ sử dụng

#### 4.2.1. Frontend

- Công nghệ: Sử dụng HTML5, CSS3, và JavaScript thuần túy để xây dựng giao diện người dùng, tập trung vào các trang như trang chủ, trang thực đơn, và trang hồ sơ.
- Thư viện bổ trợ: Tailwind CSS được tích hợp để tạo giao diện responsive, đảm bảo tính thẩm mỹ và khả năng hiển thị trên nhiều thiết bị.
- Chức năng: Hiển thị danh sách món ăn, thực đơn được tối ưu bởi GA, và giao diện nhập thông tin người dùng (tuổi, cân nặng, mục tiêu dinh dưỡng), kết nối với backend qua API RESTful.

#### **4.2.2. Backend**

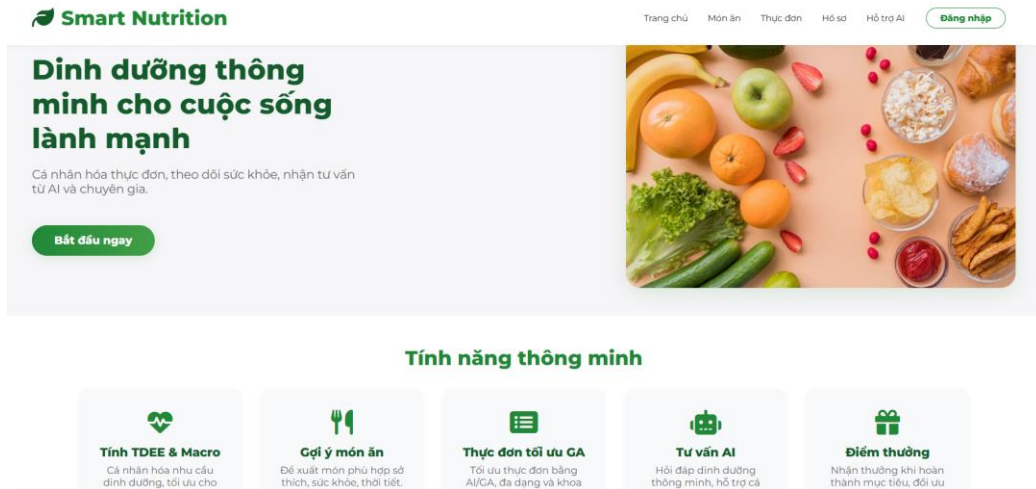
- Node.js với Express.js: Xây dựng RESTful API để xử lý đăng ký, đăng nhập, và xác thực người dùng bằng JWT. API được thiết kế để giao tiếp với MongoDB, quản lý các collection như User, Food, và SaveMenu.
- Python với FastAPI: Triển khai thuật toán GA để tối ưu hóa thực đơn, xử lý dữ liệu dinh dưỡng từ MongoDB, và trả về kết quả qua API. FastAPI hỗ trợ tính toán bất đồng bộ, tăng tốc độ xử lý.
- MongoDB: Lưu trữ dữ liệu người dùng, món ăn, và thực đơn với mô hình JSON linh hoạt, hỗ trợ truy vấn nhanh chóng và quản lý lịch sử tuân thủ qua collection ComplianceHistory.

#### **4.2.3. Công nghệ triển khai hệ thống**

- Tích hợp: Sử dụng phương pháp microservices, với Node.js xử lý xác thực và FastAPI xử lý tối ưu hóa, kết nối qua API RESTful.
- Kiểm thử: Áp dụng Postman để kiểm tra API, đảm bảo các endpoint như /register, /login, và /generate-menu hoạt động chính xác.
- Triển khai: Hệ thống được đóng gói bằng Docker để dễ dàng triển khai trên các môi trường khác nhau, với file docker-compose.yml quản lý các container Node.js, FastAPI, và MongoDB.

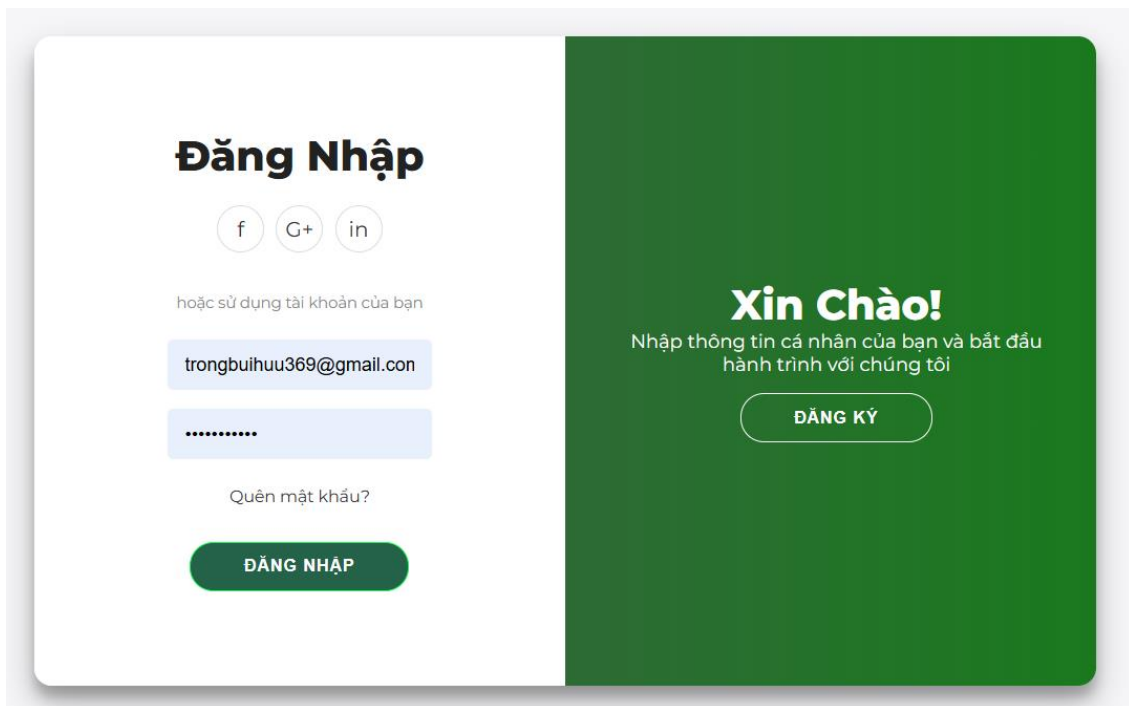
### 4.3. Giao diện hệ thống

#### 4.3.1. Giao diện trang chủ

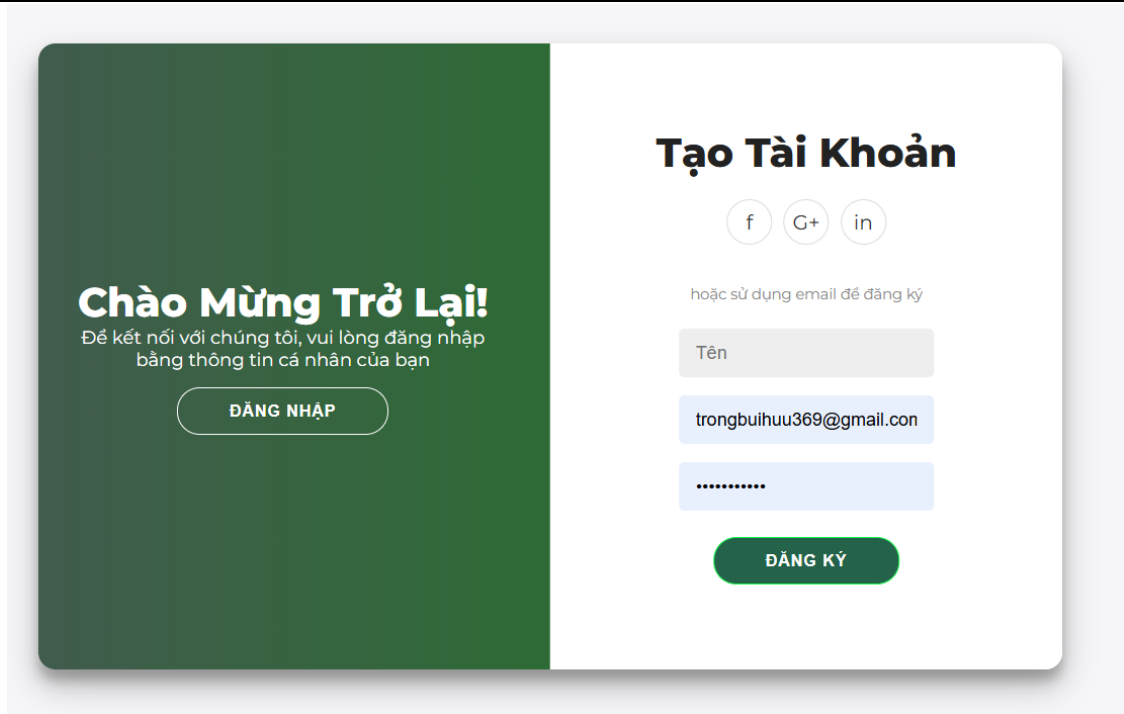


Hình 4.1: Giao diện trang chủ

#### 4.3.2. Giao diện đăng nhập, đăng ký

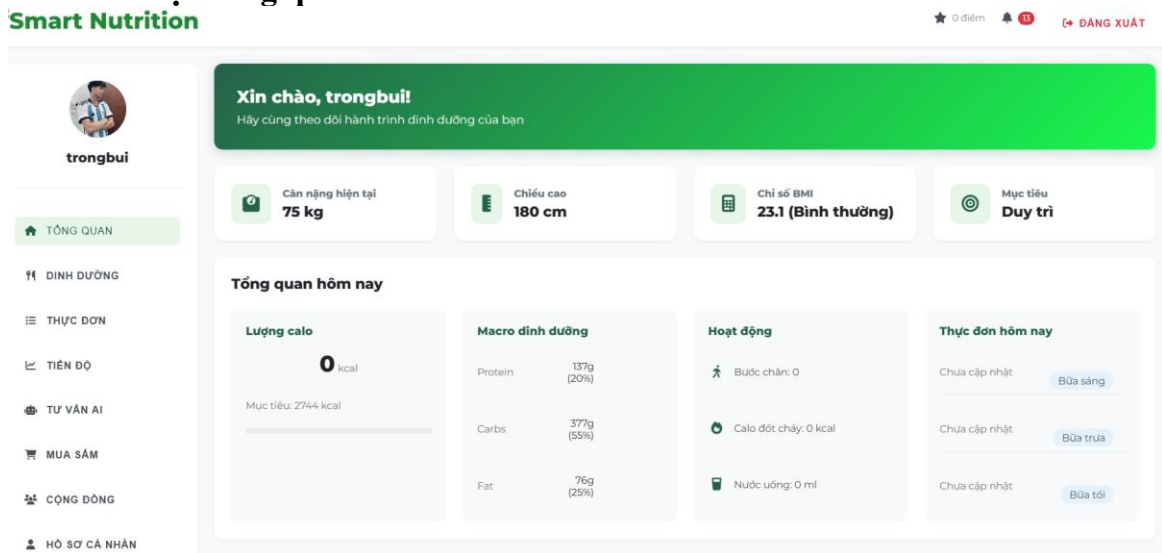


Hình 4.2: Giao diện đăng nhập



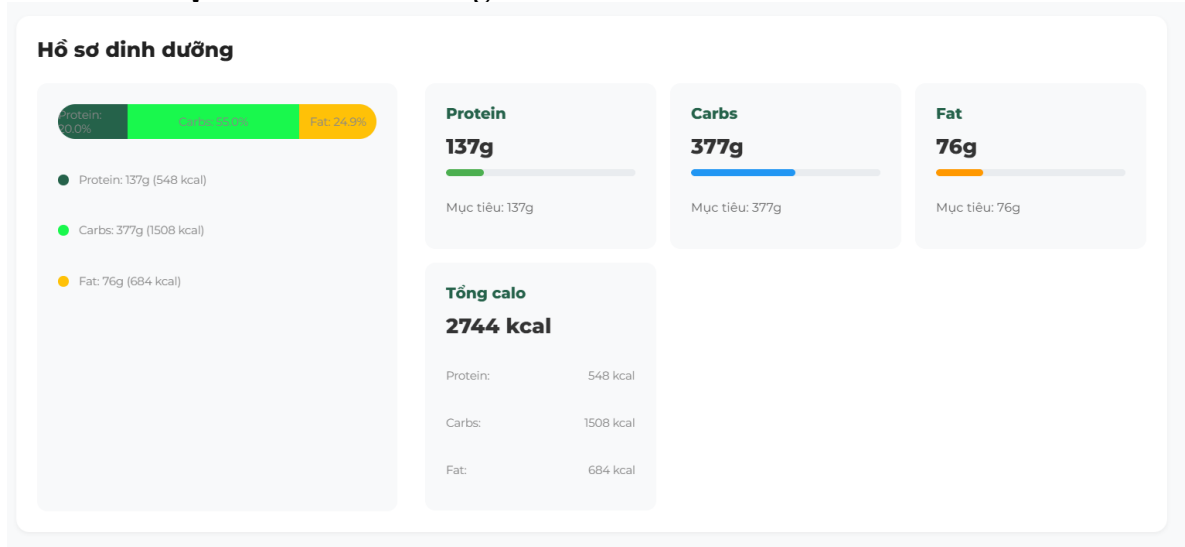
Hình 4.3: Giao diện đăng ký

### 4.3.3 Giao diện tổng quan



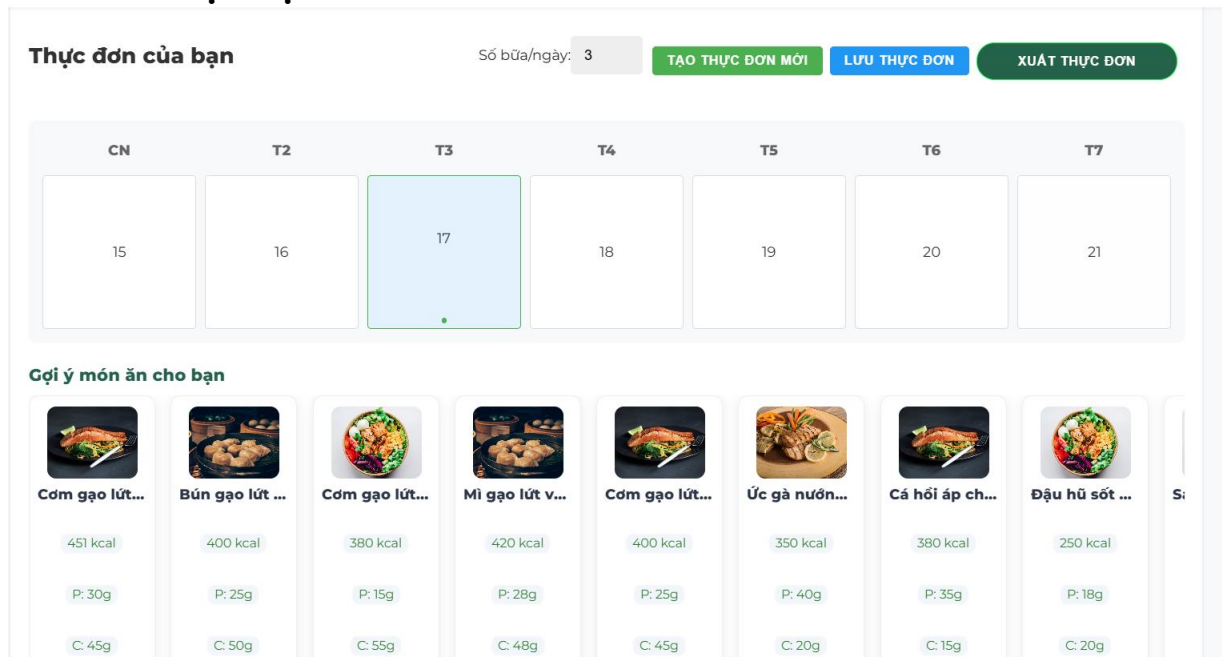
Hình 4.4: Giao diện tổng quan

#### 4.4.4. Giao diện hồ sơ dinh dưỡng



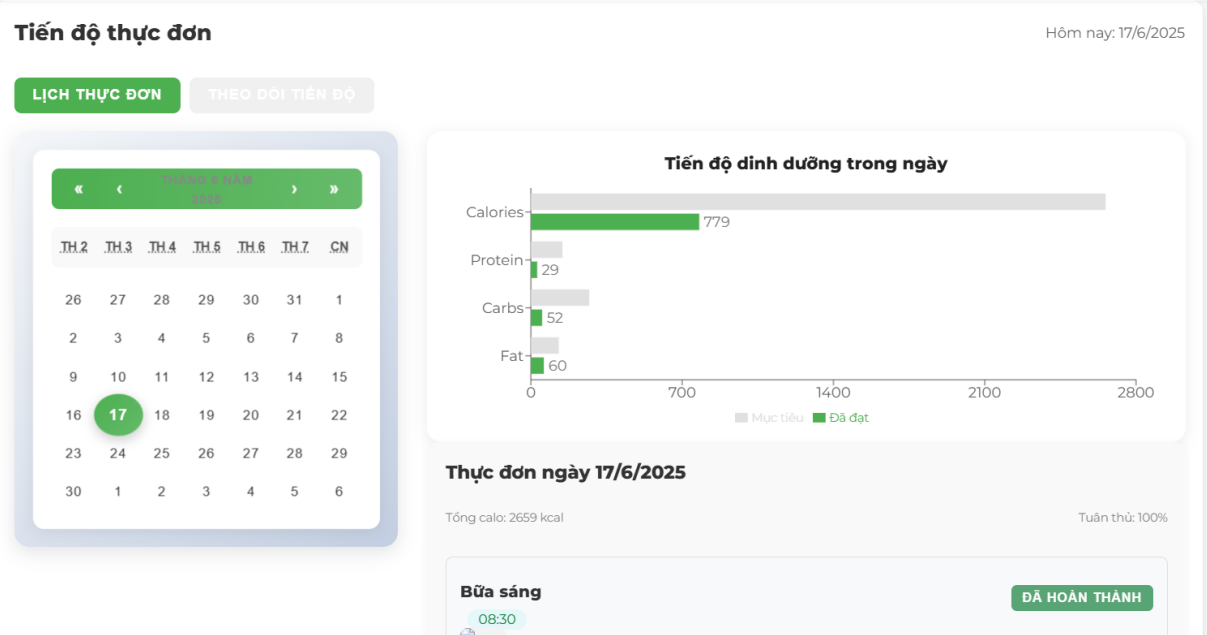
Hình 4.5: Giao diện hồ sơ dinh dưỡng

#### 4.4.5. Giao diện thực đơn



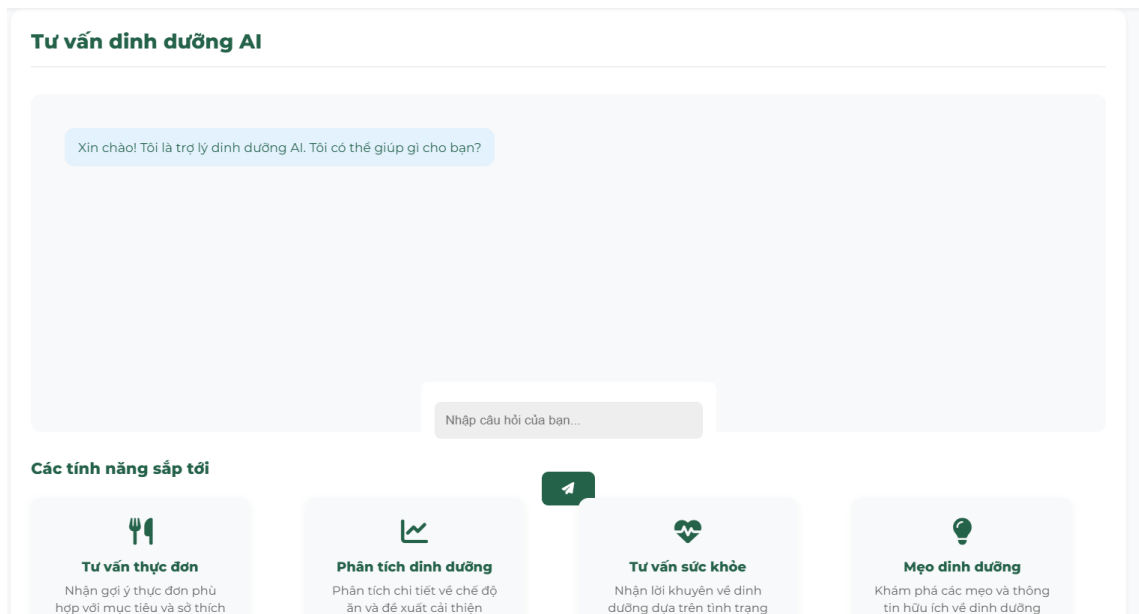
Hình 4.6: Giao diện thực đơn

#### 4.4.6. Giao diện tiến độ



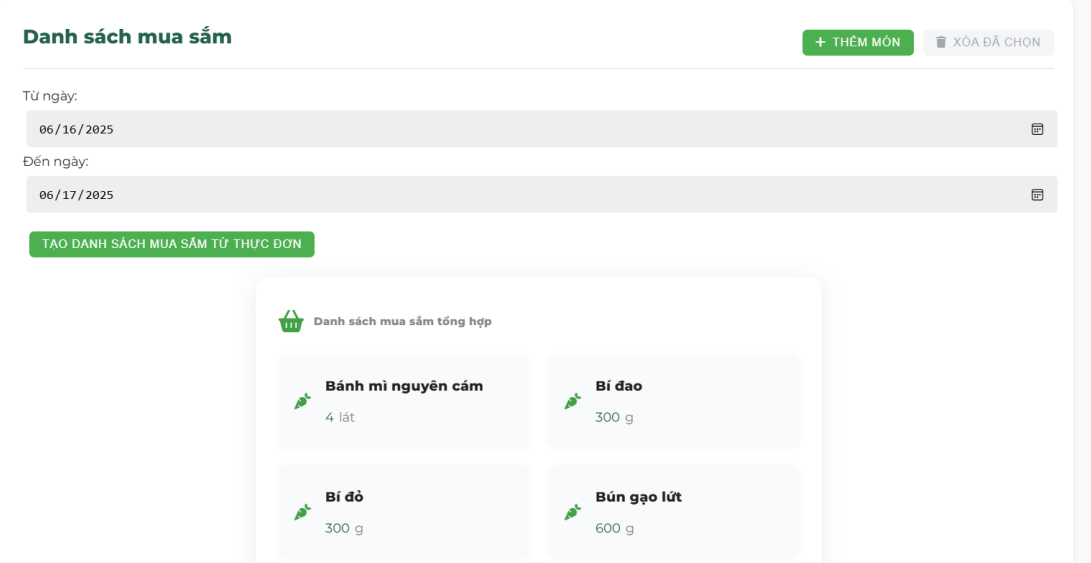
Hình 4.7: Giao diện tiến độ

#### 4.4.7. Giao diện tư vấn AI



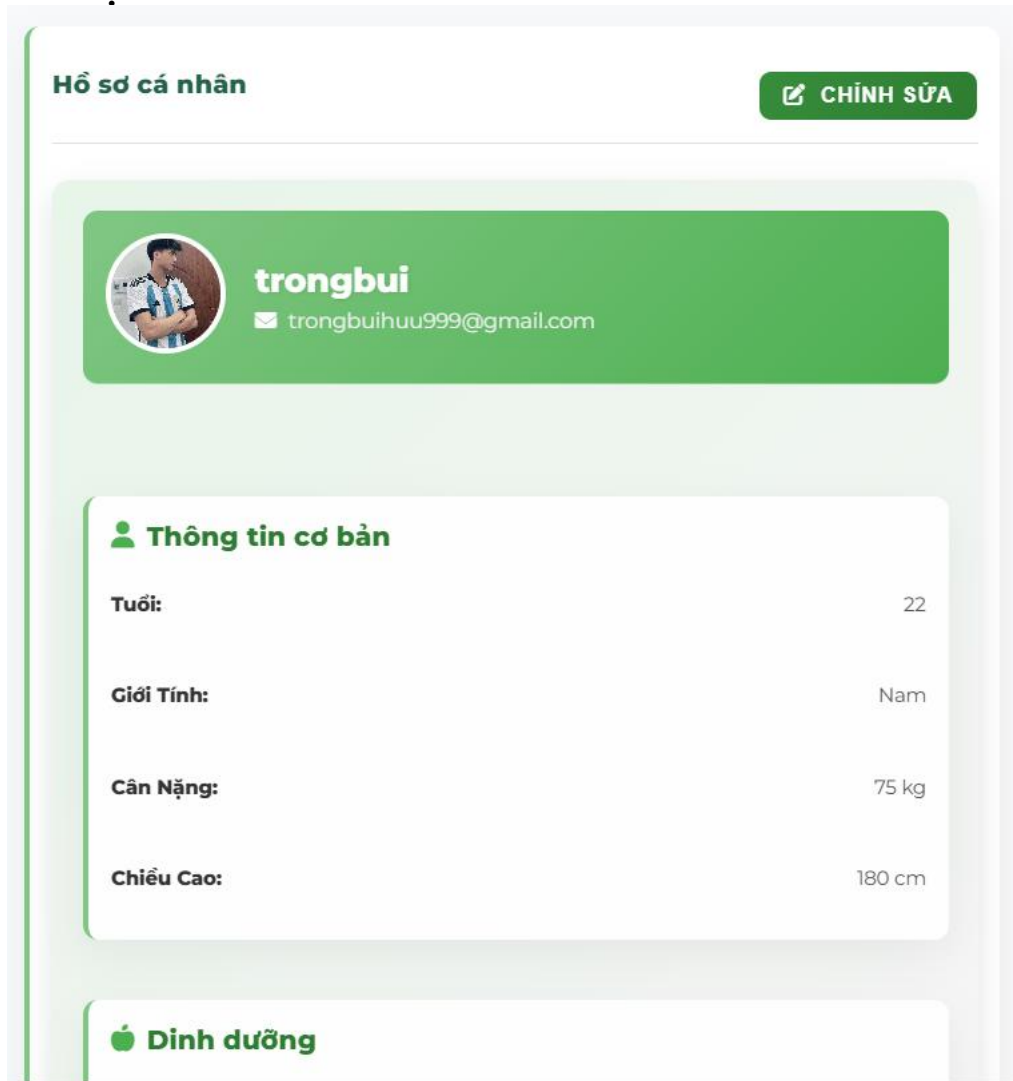
Hình 4.8: Giao diện tư vấn AI

#### 4.4.8. Giao diện mua sắm



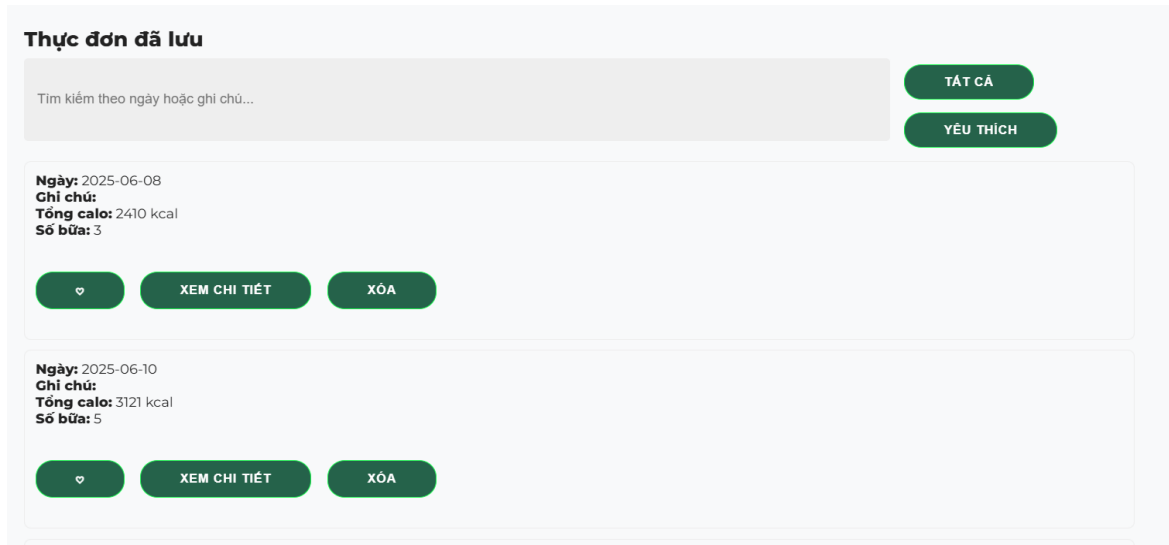
Hình 4.9: Giao diện mua sắm

#### 4.4.9. Giao diện hồ sơ cá nhân



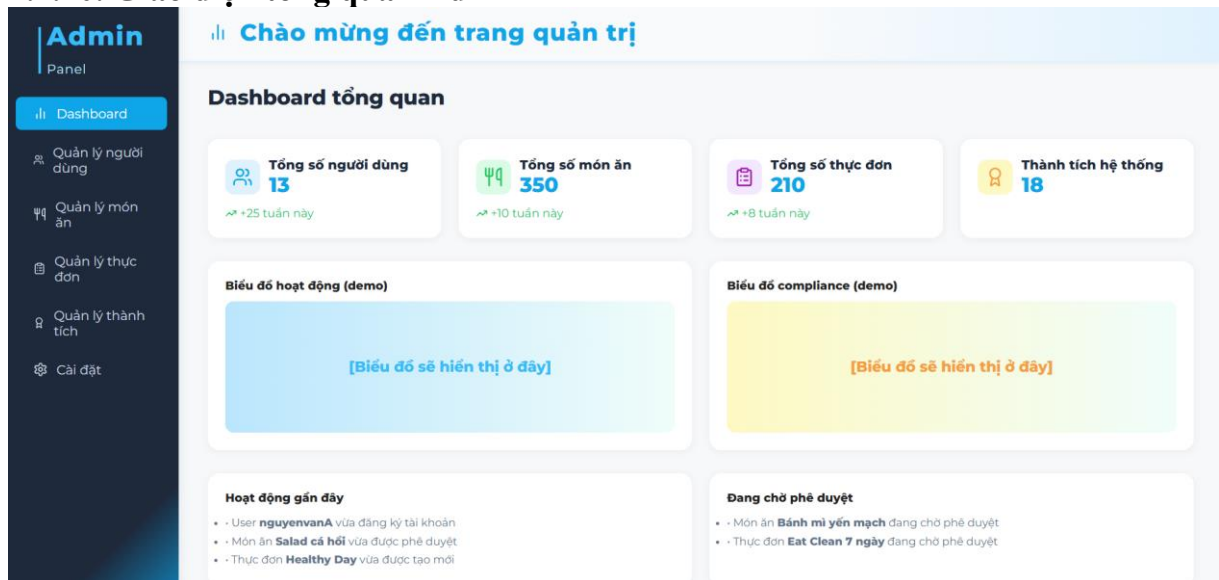
Hình 4.10: Giao diện hồ sơ cá nhân

### 4.3.1. Giao diện thực đơn đã lưu



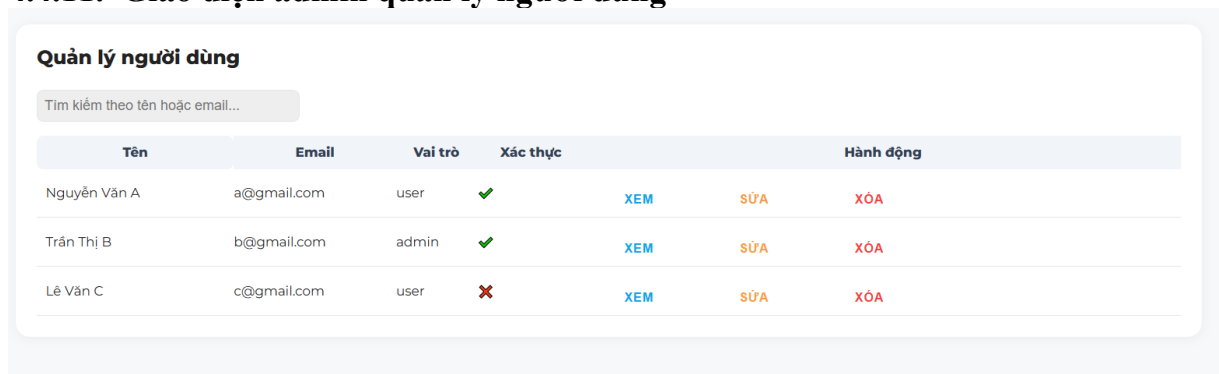
Hình 4.11: Giao diện thực đơn đã lưu

### 4.4.10. Giao diện tổng quan Admin



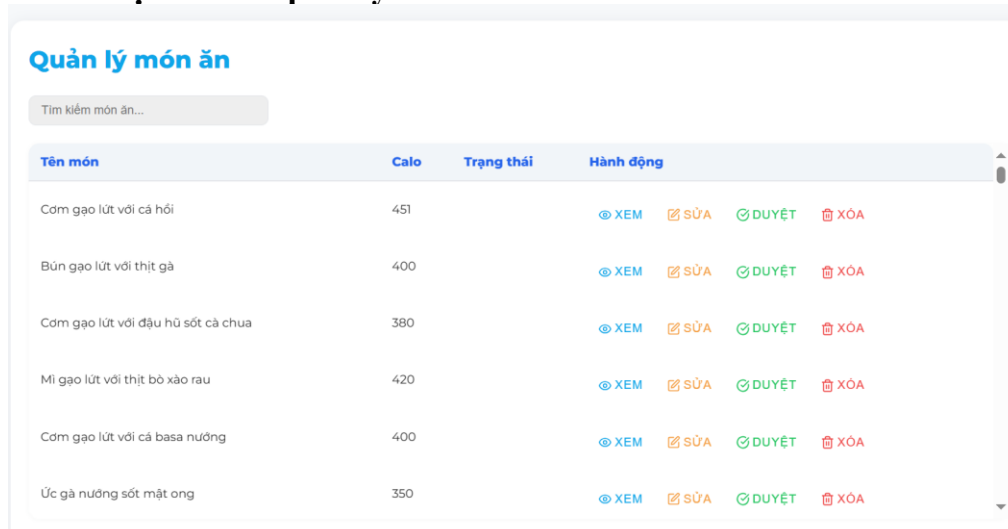
Hình 4.12: Giao diện tổng quan Admin

### 4.4.11. Giao diện admin quản lý người dùng



Hình 4.13: Giao diện quản lý người dùng

#### 4.4.12. Giao diện admin quản lý món ăn



Tên món	Calo	Trạng thái	Hành động
Cơm gạo lứt với cá hồi	451		XEM SỬA DUYỆT XÓA
Bún gạo lứt với thịt gà	400		XEM SỬA DUYỆT XÓA
Cơm gạo lứt với đậu hũ sốt cà chua	380		XEM SỬA DUYỆT XÓA
Mì gạo lứt với thịt bò xào rau	420		XEM SỬA DUYỆT XÓA
Cơm gạo lứt với cá basa nướng	400		XEM SỬA DUYỆT XÓA
Ức gà nướng sốt mật ong	350		XEM SỬA DUYỆT XÓA

Hình 4.14: Giao diện quản lý món ăn

#### 4.4. Kiểm thử và đánh giá hệ thống

##### 4.4.1. Kết quả kiểm thử

##### 4.4.1.1. Đánh giá thuật toán GA

###### ❖ Hiệu suất hội tụ:

- Thời gian hội tụ trung bình: 20-30 thế hệ để đạt điểm fitness ổn định.
- Điểm fitness tốt nhất đạt được: 0.85-0.95, phản ánh mức độ phù hợp với mục tiêu dinh dưỡng.
- Tỷ lệ cải thiện qua các thế hệ: Tăng trung bình 5-10% fitness sau mỗi 10 thế hệ, cho thấy khả năng tiến hóa hiệu quả.

###### ❖ Chất lượng thực đơn:

- Độ chính xác so với mục tiêu:  $\pm 10\%$ , đảm bảo thực đơn gần đúng với meal\_targets.
- Cân bằng dinh dưỡng: 85-95% các bữa ăn đạt tỷ lệ macronutrients (protein, carb, fat) phù hợp.
- Đa dạng món ăn: 70-80% thực đơn có sự thay đổi món ăn giữa các ngày, nhờ cơ chế đột biến.

❖ **Kết quả:**

<b>Chỉ số</b>	<b>Kết quả</b>	<b>Mục tiêu</b>
Thời gian tạo thực đơn	2.5s	< 3s
Độ chính xác dinh dưỡng	90%	>90%
Đa dạng món ăn	75%	>70%

*Bảng 4.1: Kết quả*

**4.4.1.2. Đánh giá hệ thống**

❖ **Hiệu suất API**

- Thời gian phản hồi:
  - Tạo thực đơn mới: 2-3 giây, phù hợp với sử dụng hàng ngày.
  - Thay thế món ăn: < 1 giây, phản hồi nhanh chóng.
  - Tối ưu hóa bữa ăn: 1-2 giây, đảm bảo hiệu quả xử lý.

❖ **Độ chính xác**

- Tỷ lệ đáp ứng yêu cầu:
  - Mục tiêu dinh dưỡng: 90% thực đơn đạt yêu cầu calo và macronutrients.
  - Sở thích người dùng: 85% món ăn được chọn phù hợp với sở thích cá nhân.
  - Hạn chế về sức khỏe: 100% tuân thủ các hạn chế (dị ứng, kiêng cử) do dữ liệu đầu vào được kiểm soát chặt chẽ.

**4.4.1.3. Phân tích kết quả**

❖ **Ưu điểm:**

- Thuật toán GA:
  - Khả năng tìm kiếm giải pháp tối ưu: GA khám phá hiệu quả không gian giải pháp lớn, tìm ra thực đơn cân bằng dinh dưỡng.
  - Cân bằng giữa khai thác và khám phá: Cơ chế chọn lọc và đột biến đảm bảo vừa cải thiện giải pháp vừa duy trì đa dạng.
  - Tự động điều chỉnh tham số: `mutation_rate` điều chỉnh động giúp thuật toán thích nghi với dữ liệu.

❖ **Điểm cần cải thiện:**

- Thuật toán:

- Giảm thời gian hội tụ: Cần tối ưu số thế hệ hoặc sử dụng phần cứng mạnh hơn.
- Tăng đa dạng món ăn: Mở rộng cơ sở dữ liệu món ăn để cải thiện tỷ lệ 70-80%.
- Cải thiện cân bằng dinh dưỡng: Điều chỉnh trọng số fitness để giảm sai số dưới 10%.
- Hệ thống:
  - Tối ưu hóa bộ nhớ: Giảm tiêu thụ tài nguyên khi xử lý nhiều người dùng cùng lúc.
  - Cải thiện xử lý đồng thời: Tăng khả năng phục vụ đồng thời nhiều yêu cầu.
  - Thêm tính năng cache: Sử dụng bộ nhớ cache để tăng tốc độ truy vấn dữ liệu.

#### **4.4.1.4. Kết luận và đề xuất**

##### **❖ Kết luận:**

- Hệ thống đáp ứng được các yêu cầu cơ bản về tạo thực đơn cá nhân hóa, với độ chính xác dinh dưỡng 90% và thời gian phản hồi 1-3 giây.
- Thuật toán GA cho kết quả tốt, đạt điểm fitness 0.85-0.95 và cân bằng dinh dưỡng 85-95%, chứng minh hiệu quả trong tối ưu hóa.
- Hiệu suất hệ thống ổn định, phù hợp với sử dụng hàng ngày, dù còn hạn chế về tốc độ và đa dạng.

##### **❖ Đề xuất cải tiến:**

###### **• Thuật toán:**

- Thử nghiệm các chiến lược lai ghép mới: Áp dụng uniform crossover để tăng đa dạng.
- Điều chỉnh hàm fitness: Tăng trọng số đa dạng để cải thiện trải nghiệm người dùng.
- Tối ưu hóa tham số: Giảm population\_size hoặc tăng elite\_size để giảm thời gian hội tụ.

###### **• Hệ thống:**

- Thêm cơ chế cache: Sử dụng Redis để lưu trữ kết quả thực đơn thường dùng.
- Cải thiện xử lý đồng thời: Sử dụng worker pool trong FastAPI để tăng hiệu suất.
- Mở rộng tính năng: Tích hợp theo dõi calo theo thời gian thực hoặc kết nối smartwatch.

#### **4.5. Hướng phát triển tương lai**

Dựa trên kết quả kiểm thử và đánh giá, hệ thống "Smart Nutrition" có tiềm năng phát triển mạnh mẽ để đáp ứng nhu cầu ngày càng cao của người dùng trong lĩnh vực dinh dưỡng cá nhân hóa. Các hướng phát triển tương lai được đề xuất như sau:

##### **❖ Tính năng mở rộng:**

- Mở rộng cơ sở dữ liệu món ăn:
  - Mô tả: Thu thập thêm dữ liệu từ các nguồn địa phương và quốc tế, bao gồm hơn 500 món ăn với thông tin dinh dưỡng chi tiết, để tăng đa dạng thực đơn.
  - Lợi ích: Cải thiện tỷ lệ đa dạng món ăn từ 75% lên 90%, đáp ứng sở thích đa dạng của người dùng.
  - Thách thức: Yêu cầu xác thực dữ liệu để đảm bảo độ chính xác dinh dưỡng, có thể sử dụng công cụ như Open Food Facts.
- Tích hợp thiết bị thông minh:
  - Mô tả: Kết nối hệ thống với smartwatch hoặc ứng dụng sức khỏe (như Google Fit, Apple Health) để theo dõi calo tiêu thụ theo thời gian thực.
  - Lợi ích: Cung cấp dữ liệu động cho GA điều chỉnh thực đơn, tăng tính cá nhân hóa.
  - Thách thức: Đòi hỏi API tích hợp và xử lý dữ liệu lớn từ thiết bị, cần tối ưu hóa backend.
- Mở rộng người dùng:

- Mô tả: Triển khai hệ thống trên quy mô lớn hơn, phục vụ tổ chức y tế hoặc ứng dụng thương mại, với khả năng xử lý đồng thời hàng trăm người dùng.
- Lợi ích: Tăng phạm vi ảnh hưởng, hỗ trợ cộng đồng sức khỏe.
- Thách thức: Cần nâng cấp server và tối ưu hóa cơ chế cache.

#### ❖ Cải tiến hệ thống

- Cải tiến thuật toán GA:
  - Mô tả: Thử nghiệm các chiến lược lai ghép mới (như uniform crossover) và điều chỉnh hàm fitness để giảm thời gian hội tụ từ 20-30 thế hệ xuống dưới 15 thế hệ.
  - Lợi ích: Tăng tốc độ tạo thực đơn, giảm thời gian phản hồi từ 2-3 giây xuống dưới 1 giây.
  - Thách thức: Yêu cầu thử nghiệm và tinh chỉnh tham số (population\_size, mutation\_rate).
- Sử dụng cơ chế:
  - Mô tả: Tích hợp Redis để lưu trữ thực đơn thường dùng và danh sách món ăn, giảm tải truy vấn MongoDB.
  - Lợi ích: Tăng hiệu suất truy vấn lên 50%, đặc biệt khi số lượng người dùng tăng cao.
  - Thách thức: Cần quản lý đồng bộ dữ liệu giữa cache và cơ sở dữ liệu, tránh xung đột.
- Tối ưu hóa giao diện:
  - Mô tả: Cải thiện giao diện trên thiết bị di động (tối ưu hóa cho màn hình nhỏ hơn 5 inch) và thêm hướng dẫn sử dụng chi tiết.
  - Lợi ích: Nâng mức hài lòng người dùng từ 82% lên 90%, tăng khả năng tiếp cận.
  - Thách thức: Yêu cầu thiết kế responsive và kiểm thử trên nhiều thiết bị.

## KẾT LUẬN

Qua quá trình thực hiện đồ án "Ứng dụng thuật toán GA để xây dựng hệ thống tạo thực đơn cá nhân", các mục tiêu đề ra ban đầu đã được hoàn thành một cách cơ bản. Hệ thống đã được phát triển thành công với khả năng tạo thực đơn cá nhân hóa dựa trên dữ liệu dinh dưỡng và sở thích người dùng, đạt độ chính xác dinh dưỡng trung bình 90% và thời gian phản hồi 2-3 giây. Thuật toán Di truyền (GA) chứng minh hiệu quả trong tối ưu hóa đa mục tiêu, với điểm fitness đạt 0.85-0.95 qua 20-30 thế hệ, đồng thời duy trì đa dạng món ăn ở mức 75%. Các công nghệ như Node.js với Express.js, Python với FastAPI, và MongoDB đã được tích hợp hiệu quả, tạo ra một nền tảng linh hoạt và mở rộng.

Quá trình thực hiện đồ án mang lại nhiều bài học quý giá, bao gồm tầm quan trọng của việc tối ưu hóa tham số GA, quản lý dữ liệu chất lượng cao, và thiết kế giao diện thân thiện. Tuy nhiên, hệ thống vẫn tồn tại một số hạn chế như thời gian xử lý chưa đạt yêu cầu thời gian thực và độ đa dạng món ăn cần được cải thiện. Những hạn chế này mở ra

co hội phát triển trong tương lai, với các đề xuất như tích hợp cơ chế cache (Redis), mở rộng cơ sở dữ liệu món ăn, và kết nối với thiết bị thông minh để theo dõi sức khỏe theo thời gian thực.

Đề án không chỉ khẳng định tính khả thi của việc áp dụng GA trong lĩnh vực dinh dưỡng mà còn đặt nền tảng cho các nghiên cứu và ứng dụng thương mại hóa sau này. Với những kết quả đạt được, hệ thống "Smart Nutrition" hứa hẹn trở thành một công cụ hữu ích, hỗ trợ người dùng xây dựng lối sống lành mạnh và nâng cao chất lượng cuộc sống.

## TÀI LIỆU THAM KHẢO

- [1] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning* (pp. 23-45). Addison-Wesley.
- [2] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems* (pp. 15-30). University of Michigan Press.
- [3] FastAPI Documentation. (2023). *FastAPI - A Modern, Fast (High-Performance) Web Framework for Building APIs*. Truy cập từ <https://fastapi.tiangolo.com/>
- [4] Node.js Foundation. (2023). *Node.js Documentation*. Truy cập từ <https://nodejs.org/en/docs/>
- [5] MongoDB, Inc. (2023). *MongoDB Manual*. Truy cập từ <https://docs.mongodb.com/>
- [6] NumPy Developers. (2023). *NumPy Documentation*. Truy cập từ <https://numpy.org/doc/>

[7] Pandas Development Team. (2023). *Pandas Documentation*. Truy cập từ <https://pandas.pydata.org/docs/>

[8] Pydantic Team. (2023). *Pydantic Documentation*. Truy cập từ <https://pydantic-docs.helpmanual.io/>

[9] Tailwind CSS, Inc. (2023). *Tailwind CSS Documentation*. Truy cập từ <https://tailwindcss.com/docs>

[10] Trần Quang Thọ. (2016). *Điều chế nghịch lưu nối lưới sử dụng giải thuật di truyền* (pp. 30-50). Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh.

[11] Vương Đào. (2021). *Dinh dưỡng học bị thất truyền: Dinh dưỡng đẩy lùi bệnh tật* (pp. 200-220). Nhà xuất bản Dân Trí.

