

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

---

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
NGÀNH KỸ THUẬT PHẦN MỀM

XÂY DỰNG ỨNG DỤNG WEB NHÀ HÀNG YAYA



CBHD : ThS. Vũ Thị Dương

Sinh viên : Nguyễn Văn Chung

Mã số sinh viên: 2021600963

NGUYỄN VĂN CHUNG

KỸ THUẬT PHẦN MỀM

Hà Nội – Năm 2025

## LỜI MỞ ĐẦU

Em xin được phép bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài “*Xây dựng ứng dụng web nhà hàng Yaya*”. Em cũng xin chân thành cảm ơn các thầy cô ở khoa công nghệ thông tin đã truyền đạt vốn kiến thức quý báu cho em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy, các cô nên đề tài tốt nghiệp: “*Xây dựng ứng dụng web nhà hàng Yaya*” của em mới có thể hoàn thiện. Một lần nữa, em xin chân thành cảm ơn ThS. Vũ Thị Dương - người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này trong thời gian qua. Bài báo cáo đồ án của em thực hiện trong khoảng thời gian 9 tuần. Bài báo cáo của em bao gồm 5 chương:

*Chương 1: Trình bày tổng quan về đề tài, công nghệ và cơ sở lý thuyết.*

*Chương 2: Trình bày về việc khảo sát và đặc tả ứng dụng web đặt món ăn cho nhà hàng*

*Chương 3: Đưa ra thiết kế chức năng, thiết kế cơ sở dữ liệu và thiết kế giao diện của hệ thống.*

*Chương 4: Triển khai ứng dụng web và kết quả đạt được.*

*Chương 5: Kiểm thử hệ thống*

Bước đầu đi vào thực tế của em còn hạn chế và gặp nhiều bỡ ngỡ nên không khỏi gặp nhiều thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để kiến thức của em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao kỹ năng nghề nghiệp của em.

Em xin chân thành cảm ơn!

Hà Nội, ngày ... tháng ... năm 2025

Sinh Viên

Nguyễn Văn Chung

## MỤC LỤC

<b>LỜI MỞ ĐẦU.....</b>	i
<b>MỤC LỤC.....</b>	ii
<b>DANH MỤC CÁC CHỮ VIẾT TẮT.....</b>	v
<b>DANH MỤC BẢNG BIỂU.....</b>	vi
<b>DANH MỤC HÌNH ẢNH.....</b>	vii
<b>MỞ ĐẦU.....</b>	1
1.1.    Lý do chọn đề tài .....	1
1.2.    Mục tiêu đề tài .....	2
1.3.    Nội dung nghiên cứu .....	2
1.4.    Phạm vi đề tài .....	2
1.5.    Bố cục đề tài .....	3
<b>CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....</b>	4
1.1.    Tổng quan về đề tài .....	4
1.2.    Tổng quan NextJS .....	5
1.2.1.    Giới thiệu về NextJS .....	5
1.2.2.    Các tính năng chính của NextJS .....	6
1.2.3.    Ưu điểm của NextJS .....	8
1.2.4.    Nhược điểm của NextJS .....	9
1.3.    Tổng quan MySQL .....	10
1.3.1.    MySQL là gì? .....	10
1.3.2.    Lịch sử phát triển của MySQL .....	11
1.3.3.    Ưu điểm của MySQL .....	11
1.3.4.    Nhược điểm của MySQL .....	11
1.4.    Tổng quan .NET Core và ASP.NET Core .....	12
1.4.1.    .NET Core là gì? .....	12
1.4.2.    ASP.NET Core .....	14
1.5.    Kiến trúc Monolithic .....	15
1.5.1.    Kiến trúc phần mềm là gì? .....	15
1.5.2.    Định nghĩa .....	16

1.5.3.	Kiến trúc phần mềm trong kỹ thuật phần mềm.....	16
1.5.4.	Kiến trúc nguyên khối (Monolithic) là gì? .....	16
1.5.5.	Hiểu kiến trúc nguyên khối với sản phẩm phần mềm .....	17
1.5.6.	Lợi ích của ứng dụng nguyên khối.....	18
1.5.7.	Hạn chế của ứng dụng nguyên khối.....	18
1.6.	Clean architecture.....	19
1.6.1.	Clean Architecture là gì? .....	19
1.6.2.	Lợi ích của Clean Architecture.....	20
1.6.3.	CQRS Pattern.....	21
<b>CHƯƠNG 2: PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM.....</b>		23
2.1.	Khảo sát bài toán.....	23
2.1.1.	Khảo sát nhà hàng Yaya.....	23
2.1.2.	Khảo sát các ứng dụng web .....	24
2.2.	Khảo sát hệ thống.....	25
2.2.1.	Giới thiệu chung về hệ thống.....	25
2.2.2.	Các tác nhân hệ thống.....	26
2.3.	Yêu cầu hệ thống.....	26
2.3.1.	Yêu cầu chức năng.....	26
2.3.2.	Yêu cầu phi chức năng.....	28
2.4.	Biểu đồ use case .....	29
2.4.1.	Biểu đồ use case tổng quan .....	29
2.4.2.	Biểu đồ phân rã một số use case .....	30
2.5.	Mô tả chi tiết một số use case .....	32
<b>CHƯƠNG 3: THIẾT KẾ PHẦN MỀM .....</b>		43
3.1.	Thiết kế chức năng chính.....	43
3.1.1.	Chức năng đăng nhập .....	43
3.1.2.	Chức năng xem món ăn .....	44
3.1.3.	Chức năng quản lý món ăn .....	46
3.1.4.	Chức năng quản lý đơn hàng .....	49
3.1.5.	Chức năng đặt hàng .....	51

<b>3.2.</b>	<b>Thiết kế cơ sở dữ liệu.....</b>	<b>52</b>
3.2.1.	Biểu đồ Entity Relationship Diagram.....	52
3.2.2.	Thiết kế các bảng trong cơ sở dữ liệu.....	53
<b>3.3.</b>	<b>Thiết kế giao diện .....</b>	<b>55</b>
<b>CHƯƠNG 4: KẾT QUẢ XÂY DỰNG.....</b>		<b>59</b>
4.1.	Cài đặt.....	59
4.1.1.	Frontend.....	59
4.1.2.	Backend (Xử lý nghiệp vụ).....	59
4.1.3.	Cơ sở dữ liệu .....	59
4.1.4.	Giao tiếp giữa các thành phần.....	60
4.1.5.	Triển khai Clean Architecture.....	60
4.2.	Giao diện.....	62
<b>CHƯƠNG 5: KIỂM THỬ PHẦN MỀM.....</b>		<b>68</b>
5.1.	Kế hoạch kiểm thử .....	68
5.1.1.	Giới thiệu.....	68
5.1.2.	Thực hiện kiểm thử.....	72
5.2.	Kết quả kiểm thử.....	76
5.3.	Kết luận .....	77
<b>KẾT LUẬN.....</b>		<b>78</b>
1.	Kết quả đạt được.....	78
2.	Hướng phát triển.....	78
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>79</b>

## DANH MỤC CÁC CHỮ VIẾT TẮT

Tù viết tắt	Tiếng anh	Tiếng Việt
API	Application Programming Interface	Giao diện lập trình ứng dụng
CQRS	Command and Query Responsibility Segregation	Phân tách Lệnh và Truy vấn

## **DANH MỤC BẢNG BIỂU**

Bảng 2.1. Bảng mô tả chi tiết use case đăng nhập.....	33
Bảng 2.2. Bảng mô tả chi tiết use case xem món ăn.....	34
Bảng 2.3. Bảng mô tả chi tiết use case đặt món ăn.....	35
Bảng 2.4. Bảng mô tả chi tiết use case xem đơn hàng.....	36
Bảng 2.5. Bảng mô tả chi tiết use case thêm món ăn.....	38
Bảng 2.6. Bảng mô tả chi tiết use case sửa món ăn.....	39
Bảng 2.7. Bảng mô tả chi tiết use case xóa món ăn.....	40
Bảng 2.8. Bảng mô tả chi tiết use case Quản lý đơn hàng.....	41
Bảng 2.9. Bảng mô tả chi tiết use case Quản lý trạng thái món ăn.....	42
Bảng 5.1. Bảng định nghĩa và các từ viết tắt.....	68
Bảng 5.2. Bảng lịch trình công việc.....	69
Bảng 5.3. Bảng cấu hình phần cứng.....	69
Bảng 5.4. Bảng thông tin phần mềm.....	69
Bảng 5.5. Bảng công cụ kiểm thử.....	70
Bảng 5.6. Bảng nhân sự.....	70
Bảng 5.7. Bảng loại kiểm thử hộp đen.....	71
Bảng 5.8. Bảng loại phân loại lỗi.....	72
Bảng 5.9. Bảng điều kiện ràng buộc cho Đăng nhập.....	73
Bảng 5.10. Bảng điều kiện ràng buộc cho Đổi mật khẩu.....	74

## **DANH MỤC HÌNH ẢNH**

Hình 1.1. Framework Next.js.....	5
Hình 1.2. Cơ chế Server-Side Rendering.....	7
Hình 1.3. Cơ chế Client-Side Rendering.....	8
Hình 1.4. Logo MySQL .....	10
Hình 1.5. Logo .NET Core.....	13
Hình 1.6. Logo ASP.NET Core.....	14
Hình 1.7. Kiến trúc Monolith.....	16
Hình 1.8. Clean Architecture.....	19
Hình 1.9. CQRS Pattern.....	21
Hình 2.1. Biểu đồ use case tổng quan.....	29
Hình 2.2. Biểu đồ phân rã use case phía frontend.....	30
Hình 2.3. Biểu đồ phân rã use case quản lý món ăn.....	30
Hình 2.4. Biểu đồ phân rã use case quản lý đơn hàng.....	31
Hình 2.5. Biểu đồ phân rã use case phía backend.....	31
Hình 2.6. Biểu đồ minh họa use case đăng nhập.....	33
Hình 2.7. Biểu đồ minh họa use case xem món ăn.....	34
Hình 2.8. Biểu đồ minh họa use case đặt món ăn.....	35
Hình 2.9. Biểu đồ minh họa use case xem đơn hàng.....	37
Hình 2.10. Biểu đồ minh họa use case thêm món ăn.....	38
Hình 2.11. Biểu đồ minh họa use case sửa món ăn.....	39
Hình 2.12. Biểu đồ minh họa use case xóa món ăn.....	40
Hình 2.13. Biểu đồ minh họa use case quản lý đơn hàng.....	41
Hình 2.14. Biểu đồ minh họa use case quản lý trạng thái món ăn.....	42
Hình 3.1. Biểu đồ lớp tham gia ca sử dụng chức năng đăng nhập.....	43
Hình 3.2. Biểu đồ trình tự chức năng đăng nhập.....	43
Hình 3.3. Biểu đồ xác định lớp thực thể chức năng đăng nhập.....	44
Hình 3.4. Biểu đồ lớp tham gia ca sử dụng chức năng xem món ăn.....	44
Hình 3.5. Biểu đồ trình tự chức năng xem món ăn.....	45
Hình 3.6. Biểu đồ xác định lớp thực thể chức năng xem món ăn.....	45

Hình 3.7. Biểu đồ lớp tham gia ca sử dụng chức năng quản lý món ăn.....	46
Hình 3.8. Biểu đồ trình tự chức năng quản lý món ăn.....	48
Hình 3.9. Biểu đồ xác định lớp thực thể chức năng quản lý món ăn.....	49
Hình 3.10. Biểu đồ lớp tham gia ca sử dụng chức năng quản lý đơn hàng....	49
Hình 3.11. Biểu đồ trình tự chức năng quản lý đơn hàng.....	50
Hình 3.12. Biểu đồ xác định lớp thực thể chức năng quản lý đơn hàng.....	50
Hình 3.13. Biểu đồ lớp tham gia ca sử dụng chức năng đặt hàng.....	51
Hình 3.14. Biểu đồ trình tự chức năng đặt hàng.....	51
Hình 3.15. Biểu đồ xác định lớp thực thể chức năng đặt hàng.....	52
Hình 3.16. Biểu đồ Entity Relationship Diagram.....	52
Hình 3.17. Bảng Tables.....	53
Hình 3.18. Bảng Accounts .....	53
Hình 3.19. Bảng Dishes.....	53
Hình 3.20. Bảng DishSnapshots.....	54
Hình 3.21. Bảng Guests.....	54
Hình 3.22. Bảng Orders.....	54
Hình 3.23. Bảng RefreshTokens .....	55
Hình 3.24. Giao diện mockup Đăng nhập.....	55
Hình 3.25. Giao diện mockup Trang chủ.....	56
Hình 3.26. Giao diện mockup Quản lý đơn hàng.....	56
Hình 3.27. Giao diện mockup Quản lý bàn ăn.....	57
Hình 3.28. Giao diện mockup Thông kê.....	57
Hình 3.29. Giao diện mockup Menu.....	58
Hình 4.1. Giao diện đăng nhập.....	62
Hình 4.2. Màn hình trang chủ .....	62
Hình 4.3. Màn hình quản lý đơn hàng.....	63
Hình 4.4. Màn hình chi tiết quản lý đơn hàng tại một bàn.....	63
Hình 4.5. Màn hình chi tiết cập nhật đơn hàng.....	64
Hình 4.6. Màn hình quản lý bàn ăn.....	64
Hình 4.7. Màn hình quản lý món ăn.....	65

Hình 4.8. Màn hình thống kê doanh thu.....	65
Hình 4.9. Màn hình quản lý nhân viên.....	66
Hình 4.10. Màn hình chi tiết menu món ăn.....	66
Hình 4.11. Màn hình đơn đặt hàng.....	67
Hình 5.1. Giao diện Đăng nhập.....	72
Hình 5.2. Danh sách Test case Đăng nhập.....	73
Hình 5.3. Kết quả kiểm thử Đăng nhập bằng Selenium.....	73
Hình 5.4. Giao diện Đổi mật khẩu.....	74
Hình 5.5. Danh sách Test case Đổi mật khẩu.....	75
Hình 5.6. Kết quả kiểm thử Đổi mật khẩu bằng Selenium .....	75
Hình 5.7. Giao diện. thử thông báo realtime cho người dùng về trạng thái đơn đặt hàng.....	75
Hình 5.8. Danh sách Test case Kiểm thử thông báo realtime cho người dùng về trạng thái đơn đặt hàng.....	76
Hình 5.9. Kết quả kiểm thử Thông báo realtime bằng Selenium.....	76

## MỞ ĐẦU

### 1.1. Lý do chọn đề tài

Hiện nay công nghệ thông tin không ngừng phát triển mạnh mẽ và hiện đại, các nền tảng số ngày càng tiếp cận tới người dùng dễ dàng hơn, trong đó có các hệ thống đặt đồ món ăn dần được phổ biến trên thị trường hiện nay với nhiều nền tảng công nghệ.

Đề tài " Xây dựng ứng dụng web nhà hàng Yaya" nhằm mục đích xây dựng một nền tảng hiện đại, thân thiện với người dùng, và đáp ứng tốt nhu cầu thực tế của các nhà hàng. Đề tài kết hợp giữa Next.js - một framework phát triển giao diện người dùng nhanh và tối ưu, cùng .NET Core - nền tảng mạnh mẽ cho việc xây dựng backend, ứng dụng sẽ cung cấp một hệ thống toàn diện bao gồm các chức năng như đặt món trực tuyến, quản lý thực đơn, xử lý thanh toán, và theo dõi trạng thái đơn hàng.

Bên cạnh đó, việc đặt món qua mạng giúp cửa hàng có thể tiết kiệm ngân sách nhân viên, tối ưu quản lý nghiệp vụ. Hơn nữa, mỗi khách hàng chúng ta hiện nay ngày càng ưu chuộng việc tiện lợi, nhanh chóng và mất ít công sức hơn. Đối với thị trường Việt Nam, việc có các ứng dụng đặt món ngay tại cửa hàng online đang phổ biến hơn vì sự tiện lợi và nhanh chóng của nó.

Với mong muốn cung cấp và nâng cao kiến thức về lập trình, áp dụng kiến thức để xây dựng một ứng dụng giúp người dùng có thể dễ dàng sử dụng đặt món ăn, tham khảo, lựa chọn các thực phẩm mình mong muốn với mức giá chuẩn xác, sử dụng hệ thống với công nghệ mới nhanh và mượt mà hơn em đã chọn đề tài “Xây dựng ứng dụng web nhà hàng Yaya”.

Đề tài mang tính thực tiễn cao, phù hợp với xu hướng hiện tại và mở ra cơ hội nghiên cứu sâu hơn về việc ứng dụng công nghệ trong ngành dịch vụ. Với những lợi ích thiết thực mà đề tài mang lại, nó không chỉ là một bài tập tốt nghiệp mà còn có thể triển khai thực tế, mang lại giá trị lâu dài cho cả nhà hàng và khách hàng.

## 1.2. Mục tiêu đề tài

Làm chủ Nextjs và .NET Core, phân tích, thiết kế, phát triển ứng dụng web đặt đồ ăn cho nhà hàng, đáp ứng các yêu cầu cơ bản của một trang đặt đồ ăn.

Ứng dụng sẽ bao gồm hai phân hệ chính:

Phân hệ khách hàng: Cho phép người dùng xem thực đơn, chọn món, đặt hàng và theo dõi trạng thái đơn hàng. Giao diện được thiết kế trực quan, thân thiện, tương thích với nhiều loại thiết bị như máy tính và điện thoại.

Phân hệ quản trị: Dành cho quản lý nhà hàng để theo dõi và xử lý đơn hàng, cập nhật thực đơn, quản lý tài khoản nhân viên, và xem báo cáo doanh thu.

Mục tiêu chính của đề tài là cung cấp một giải pháp hoàn chỉnh, dễ sử dụng và có khả năng mở rộng trong tương lai. Việc sử dụng các công nghệ tiên tiến như Next.js và .NET Core không chỉ đảm bảo hiệu suất và tính bảo mật cao, mà còn giúp ứng dụng dễ dàng tích hợp với các hệ thống khác hoặc nâng cấp để đáp ứng các yêu cầu mới.

## 1.3. Nội dung nghiên cứu

Trong phạm vi đề tài và bài báo cáo, em tiến hành các nội dung sau:

- Khảo sát và phân tích yêu cầu quy trình đặt món ăn.
- Phân tích thiết kế hệ thống.
- Thiết kế cơ sở dữ liệu.
- Lập trình backend API với .NET Core.
- Lập trình giao diện sử dụng Nextjs và kết nối tới API.
- Có kế hoạch và đánh giá kiểm thử hệ thống

## 1.4. Phạm vi đề tài

Đề tài "Xây dựng ứng dụng web nhà hàng Yaya" tập trung vào việc xây dựng hệ thống đặt đồ ăn trực tuyến dành cho nhà hàng.

- Các tính năng chính như hiển thị thực đơn, đặt món, quản lý đơn hàng và theo dõi trạng thái đơn hàng, quản lý nhân viên, quản lý món ăn.

- Ứng dụng chỉ hỗ trợ thanh toán trực tiếp tại nhà hàng khi khách nhận món, chưa tích hợp các cổng thanh toán trực tuyến.
- Phản báo cáo thống kê trong hệ thống giới hạn ở việc hiển thị doanh thu theo ngày, tuần và tháng mà không đi sâu vào các chỉ số phân tích nâng cao như hành vi người dùng hay xu hướng tiêu dùng.
- Ngoài ra, ứng dụng được thiết kế hướng đến quy mô nhỏ và vừa, chưa hỗ trợ tích hợp đa nhà hàng hoặc hệ thống quản lý chuỗi.

### **1.5. Bố cục đề tài**

Ngoài các phần Mở đầu, Kết luận và Tài liệu tham khảo, báo cáo đồ án được bố cục thành 5 chương chính sau.

- Chương 1: Trình bày tổng quan về đề tài, công nghệ và cơ sở lý thuyết.
- Chương 2: Trình bày về việc khảo sát và đặc tả ứng dụng web đặt món ăn cho nhà hàng.
- Chương 3: Đưa ra thiết kế chức năng, thiết kế cơ sở dữ liệu và thiết kế giao diện của hệ thống.
- Chương 4: Triển khai ứng dụng web và kết quả đạt được
- Chương 5: Kiểm thử hệ thống

## CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. Tổng quan về đề tài

Ngày nay, với sự phát triển không ngừng của công nghệ, các doanh nghiệp trong ngành dịch vụ ăn uống đang dần chuyển đổi sang mô hình số hóa để đáp ứng nhu cầu của khách hàng và nâng cao hiệu quả hoạt động. Một trong những xu hướng phổ biến nhất là xây dựng các ứng dụng web cho phép khách hàng dễ dàng đặt đồ ăn trực tuyến. Ứng dụng không chỉ giúp tiết kiệm thời gian, tạo sự tiện lợi cho khách hàng mà còn giúp nhà hàng quản lý đơn hàng và dữ liệu một cách hiệu quả hơn.

Đề tài "Xây dựng ứng dụng web nhà hàng Yaya" nhằm mục đích xây dựng một nền tảng hiện đại, thân thiện với người dùng, và đáp ứng tốt nhu cầu thực tế của các nhà hàng. Với sự kết hợp giữa Next.js - một framework phát triển giao diện người dùng nhanh và tối ưu, cùng .NET Core - nền tảng mạnh mẽ cho việc xây dựng backend, ứng dụng sẽ cung cấp một hệ thống toàn diện bao gồm các chức năng như đặt món trực tuyến, quản lý thực đơn, xử lý thanh toán, và theo dõi trạng thái đơn hàng.

Ứng dụng sẽ bao gồm hai phân hệ chính:

Phân hệ khách hàng: Cho phép người dùng xem thực đơn, chọn món, đặt hàng và theo dõi trạng thái đơn hàng. Giao diện được thiết kế trực quan, thân thiện, tương thích với nhiều loại thiết bị như máy tính và điện thoại.

Phân hệ quản trị: Dành cho quản lý nhà hàng để theo dõi và xử lý đơn hàng, cập nhật thực đơn, quản lý tài khoản nhân viên, và xem báo cáo doanh thu.

Mục tiêu chính của đề tài là cung cấp một giải pháp hoàn chỉnh, dễ sử dụng và có khả năng mở rộng trong tương lai. Việc sử dụng các công nghệ tiên tiến như Next.js và .NET Core không chỉ đảm bảo hiệu suất và tính bảo mật cao, mà còn giúp ứng dụng dễ dàng tích hợp với các hệ thống khác hoặc nâng cấp để đáp ứng các yêu cầu mới.

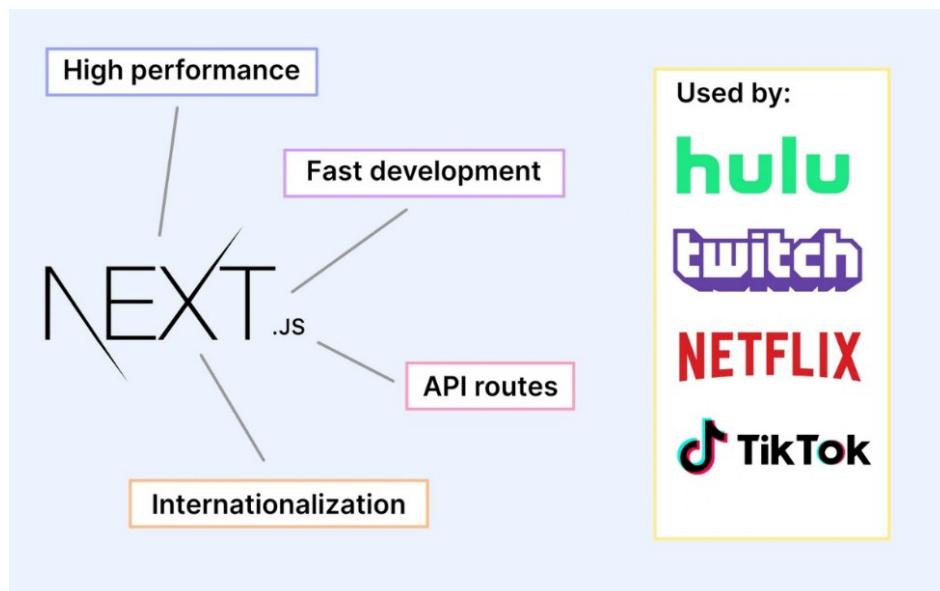
Đề tài mang tính thực tiễn cao, phù hợp với xu hướng hiện tại và mở ra cơ hội nghiên cứu sâu hơn về việc ứng dụng công nghệ trong ngành dịch vụ. Với những lợi ích thiết thực mà đề tài mang lại, nó không chỉ là một bài tập tốt nghiệp mà còn có thể triển khai thực tế, mang lại giá trị lâu dài cho cả nhà hàng và khách hàng.

## 1.2. Tổng quan NextJS

### 1.2.1. Giới thiệu về NextJS

NextJS là một framework có mã nguồn mở được xây dựng trên nền tảng của React, cho phép chúng ta xây dựng các trang web tĩnh có tốc độ siêu nhanh và thân thiện với người dùng, cũng như xây dựng các ứng dụng web React.

NextJS được ra đời vào năm 2016, thuộc sở hữu của Vercel. NextJS bắt đầu trở nên phổ biến vào năm 2018 và tiếp tục tăng trưởng mạnh mẽ trong cộng đồng phát triển web. Sự kết hợp của các tính năng như Server-side Rendering (SSR) với Static Site Generation (SSG) đã giúp NextJS trở thành sự lựa chọn hấp dẫn cho nhiều dự án.



Hình 1.1. Framework Next.js

### 1.2.2. Các tính năng chính của NextJS

#### 1. Routing trong NextJS

- Automatic Routing: NextJS sẽ tự động tạo các router dựa trên cấu trúc thư mục của chúng ta. Ví dụ, nếu bạn tạo một file có tên là about.js ở thư mục pages. NextJS sẽ tạo router là /about.
- Nested Routing: Chúng ta có thể tạo các thư mục con để tạo các router lồng nhau. Ví dụ, nếu bạn tạo một folder có tên blog nằm trong folder pages, bên trong folder blog lại có file post.js, đường dẫn sẽ là pages/blog/post.js, thì router mà NextJS tạo ra sẽ là /blog/post.
- Dynamic Routes: Có thể tạo các router động bằng cách sử dụng cặp dấu [] trong tên file. Ví dụ nếu đường dẫn là pages/blog/[slug].js thì NextJS sẽ tạo ra các router như /blog/blog-dau-tien hoặc /blog/blog-thu-hai. Với slug là một giá trị bắt kì do bạn truyền vào.
- Link Component: Để tạo liên kết giữa các trang, ta sử dụng component Link được cung cấp sẵn bởi NextJS ở thư viện next/link. Sử dụng Link thay cho thẻ a giúp tránh việc tải lại trang và tối ưu hóa hiệu suất.
- Query Parameter: Có thể truyền dữ liệu giữa các trang sử dụng query parameters trong router bằng cách sử dụng ký tự dấu chấm hỏi ? trong tên file. Ví dụ, pages/product.js có thể có các router như /product?productId=0001.

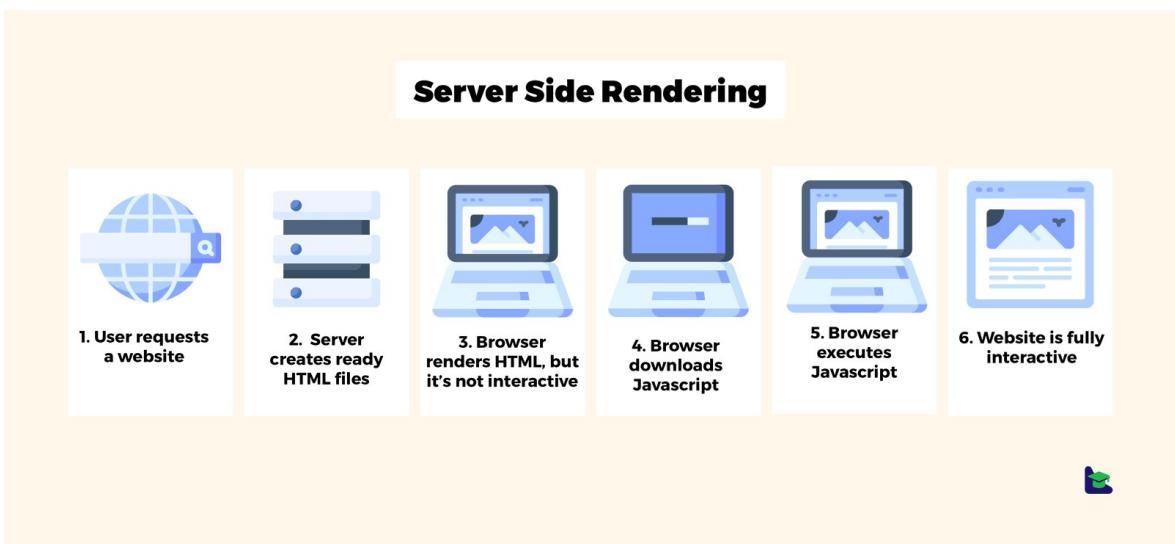
1.

#### 2. Rendering trong NextJS

- Server-side Rendering (SSR): Từ những năm 2000, SSR đã được sử dụng rất phổ biến, gọi nó là SSR vì hầu hết các logic phức tạp trên trang web sẽ được xử lý ở phía server.
  - Khi user truy cập vào một trang sử dụng SSR của NextJS, browser sẽ gửi một request đến server
  - Trước khi gửi kết quả ra trình duyệt, NextJS chạy hàm getServerSideProps() để lấy dữ liệu cần thiết cho trang. Hàm này

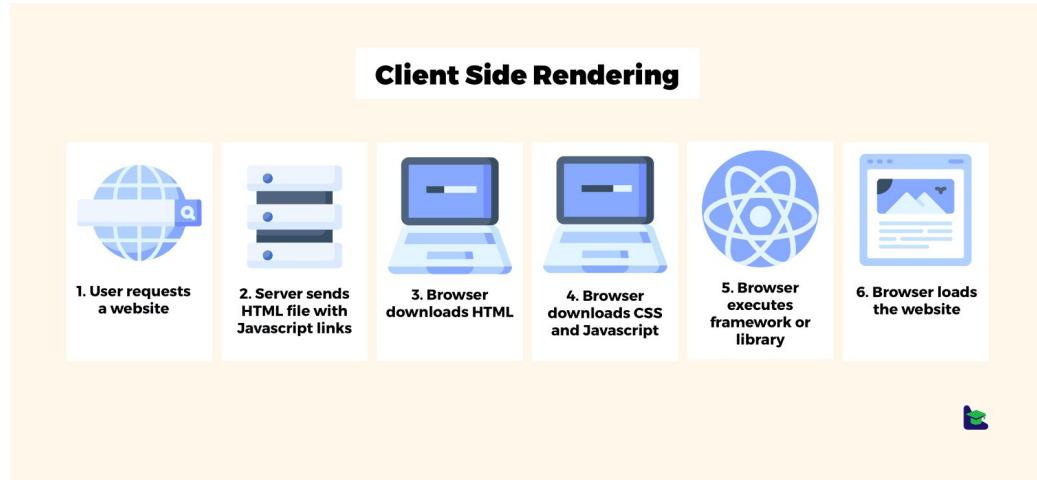
thường sẽ dùng để gọi API hoặc truy vấn vào cơ sở dữ liệu để lấy dữ liệu cần thiết.

- Với dữ liệu cần thiết đã được lấy từ bước trước, NextJS tạo ra một phiên bản đã render đầy đủ cả nội dung và dữ liệu. Sau đó trả phiên bản này ra browser dưới dạng HTML
- Browser nhận được code HTML từ server và hiển thị lên giao diện. Nội dung của trang sẽ được hiển thị ngay lập tức, sau đó JS được tải xuống và đổ lên để trang có thể tương tác được, ví dụ như hàm onClick().



*Hình 1.2. Cơ chế Server-Side Rendering*

- Client-side Rendering (CSR): Sở dĩ nó được gọi là CSR vì việc render HTML sẽ được thực thi ở phía client. Hay chúng ta còn gọi là Single Page App (SPA).
  - Chuyển việc xử lý dữ liệu sang cho client giúp server nhẹ việc hơn.
  - Trang chỉ cần load một lần duy nhất, khi user muốn lấy dữ liệu mới từ server chỉ cần gọi đến server thông qua AJAX
  - Trang web không cần load lại nhiều khi user chuyển trang, đem đến trải nghiệm tốt hơn cho người dùng



*Hình 1.3. Cơ chế Client-Side Rendering*

### 3. Styling trong NextJS

- CSS Modules: Để style cho ứng dụng NextJS, cách dễ nhất là bạn có thể tạo các file CSS/SCSS riêng lẻ cho từng component hoặc sử dụng file chung cho toàn dự án.
- CSS-in-JS Libraries: Ngoài ra bạn cũng có thể sử dụng các thư viện CSS-in-JS như Styled-Components để viết trực tiếp css vào code.
- CSS Frameworks: NextJS cũng hỗ trợ sử dụng cùng các CSS framework như TailwindCSS, Bootstrap hoặc MaterialUI.

#### 1.2.3. Ưu điểm của NextJS

NextJS hiện tại đang được sử dụng ở rất nhiều dự án khác nhau, sở dĩ NextJS được tin dùng như vậy là vì một số lý do sau:

- Sử dụng SSR và SSG: Giúp cải thiện tốc độ tải trang và khả năng SEO.
- Có nhiều tính năng giúp tối ưu hóa hiệu suất như Code Splitting, Lazy Loading, Image Optimization,...
- Fast Refresh: Tính năng giúp tự động làm mới giao diện mà không cần load lại toàn bộ trang.
- Tự động tạo file CSS dành riêng cho mỗi trang, giúp tránh xung đột trong việc sử dụng và quản lý các file CSS.

- Hỗ trợ TypeScript: NextJS cũng hỗ trợ sử dụng Typescript giúp cải thiện tính rõ ràng cho code và thuận tiện cho việc debug về sau.
- Cộng đồng lớn: NextJS có một cộng đồng sử dụng đông đảo, điều này được chứng minh ở trên chính trang Github của NextJS khi nó đang đạt khoảng hơn 100k sao. Điều này giúp cho NextJS có thêm nhiều nguồn tài liệu phong phú và các plugin hữu ích.
- Hệ sinh thái mạnh mẽ: NextJS kết hợp tốt với các thư viện và công cụ như Redux, React Query, Apollo Client và nhiều thư viện khác nằm trong hệ sinh thái của React.
- Tích hợp tốt với React: Nếu bạn đã quen với việc sử dụng React trước đó thì việc làm quen với NextJS sẽ đơn giản hơn rất nhiều.

#### **1.2.4. Nhược điểm của NextJS**

Mặc dù có nhiều ưu điểm nêu trên, NextJS vẫn có những khuyết điểm nên chúng ta cần xem xét trước khi sử dụng nó:

- Khó học cho người mới: Nếu chưa có hiểu biết cơ bản về Web Fundamentals, JavaScript và React thì việc học NextJS sẽ hơi khó khăn. Nhất là khi bạn gặp các khái niệm như SSR hay SSG.
- Khó khăn trong việc tích hợp với một số thư viện bên ngoài: Một số thư viện và plugin có thể cần phải điều chỉnh hoặc tùy chỉnh để hoạt động tốt với Next.js. Ví dụ như để sử dụng Redux trong ứng dụng NextJS, các bạn cần cài thêm thư viện next-redux-wrapper để quản lý state trên cả server và client.
- Phụ thuộc vào hệ sinh thái của React: Next.js phụ thuộc vào React, vì vậy nếu bạn không quen thuộc với React hoặc không muốn sử dụng React thì NextJS không phải là lựa chọn tốt.
- Đòi hỏi chạy trên server NodeJS: Để deploy ứng dụng NextJS, bạn cần có một máy chủ NodeJS, việc này có thể làm tăng chi phí và quá trình triển khai sẽ trở nên phức tạp hơn.

- Cấu trúc dự án phức tạp: Với các dự án lớn, việc quản lý cấu trúc dự án không cẩn thận lúc ban đầu sẽ dẫn đến việc khó quản lý sau này.
- Tuỳ vào tính chất của dự án và yêu cầu mà các bạn có thể cân nhắc có nên sử dụng NextJS trong dự án của mình hay không. Một số dự án thường được sử dụng NextJS để triển khai như: Trang tin tức, Blog, Landing Page,...

### **1.3. Tổng quan MySQL**

#### **1.3.1. MySQL là gì?**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mã nguồn mở, sử dụng ngôn ngữ truy vấn cấu trúc (SQL) để quản lý và thao tác dữ liệu. MySQL tổ chức và lưu trữ dữ liệu dưới dạng các bảng, trong đó các bảng có thể liên kết với nhau thông qua các khóa (keys).

MySQL được sử dụng rộng rãi trong việc phát triển các ứng dụng web. Nhiều công ty lớn như: Facebook, Twitter, và YouTube sử dụng MySQL để quản lý dữ liệu.



*Hình 1.4. Logo MySQL*

### **1.3.2. Lịch sử phát triển của MySQL**

MySQL được tạo ra bởi ba nhà phát triển: Michael Widenius, David Axmark và Allan Larsson vào năm 1995. Họ muốn tạo ra một hệ quản trị cơ sở dữ liệu nhanh, đáng tin cậy và dễ sử dụng cho các dự án của mình.

Năm 2008, công ty Sun Microsystems mua lại MySQL AB, công ty phát triển MySQL, với giá 1 tỷ USD. Sau đó, vào năm 2010, Oracle Corporation mua lại Sun Microsystems, và từ đó, MySQL trở thành một phần của Oracle.

### **1.3.3. Ưu điểm của MySQL**

- MySQL là phần mềm mã nguồn mở, cho phép người dùng sử dụng và tùy chỉnh mà không cần phải trả phí.
- Được thiết kế để xử lý dữ liệu nhanh chóng và hiệu quả, MySQL phù hợp với các ứng dụng web đòi hỏi tốc độ cao và khả năng phản hồi nhanh.
- Có nhiều tài liệu, hướng dẫn về cú pháp SQL phù hợp với cả người mới bắt đầu.
- Cung cấp nhiều tính năng bảo mật như xác thực người dùng, mã hóa dữ liệu và quản lý quyền truy cập chi tiết, giúp bảo vệ dữ liệu quan trọng.
- MySQL có thể xử lý từ các cơ sở dữ liệu nhỏ đến các hệ thống lớn với hàng triệu bản ghi, dễ dàng mở rộng khi nhu cầu tăng lên mà không cần thay đổi cấu trúc hệ thống.
- Hỗ trợ các tính năng transaction đảm bảo tính toàn vẹn và nhất quán của dữ liệu, giúp quản lý các thao tác dữ liệu phức tạp.

### **1.3.4. Nhược điểm của MySQL**

- So với các hệ quản trị cơ sở dữ liệu như PostgreSQL, MySQL thiếu một số tính năng nâng cao như hỗ trợ đầy đủ cho các truy vấn phức tạp, hệ thống kiểu dữ liệu phong phú hơn, và các tính năng phân tích dữ liệu tiên tiến.
- MySQL có thể gặp vấn đề về hiệu suất, thời gian phản hồi chậm khi xử lý lượng dữ liệu cực kỳ lớn hoặc các truy vấn rất phức tạp,

- Một số tính năng không thể tùy chỉnh hoặc mở rộng theo nhu cầu cụ thể của doanh nghiệp, hạn chế khả năng thích ứng với các yêu cầu đặc thù.
- Việc quản lý quyền truy cập chi tiết cho nhiều người dùng và vai trò khác nhau có thể trở nên phức tạp và khó khăn.
- Mặc dù có cộng đồng lớn, nhưng việc không có hỗ trợ chính thức miễn phí từ nhà phát triển có thể gây khó khăn khi gặp các vấn đề phức tạp hoặc lỗi hệ thống nghiêm trọng.
- MySQL có một số mở rộng và khác biệt so với chuẩn SQL, điều này có thể gây khó khăn khi chuyển đổi hoặc tích hợp với các hệ quản trị cơ sở dữ liệu khác.

## 1.4. Tổng quan .NET Core và ASP.NET Core

### 1.4.1. .NET Core là gì?

.NET Core là một nền tảng phát triển phần mềm đa nền tảng, mã nguồn mở do Microsoft phát triển. Đây là bước tiến vượt bậc từ .NET Framework truyền thống, mang lại sự linh hoạt và hiệu năng cao hơn, phù hợp với các xu hướng phát triển phần mềm hiện đại. .NET Core được thiết kế để hoạt động trên nhiều hệ điều hành như Windows, Linux và macOS, đáp ứng nhu cầu phát triển ứng dụng trên nhiều môi trường khác nhau và có thể được sử dụng để xây dựng các ứng dụng thiết bị, đám mây và IoT.



Hình 1.5. Logo .NET Core

## 1. Đặc điểm của .NET Core:

- Đa nền tảng: Chạy trên các hệ điều hành Windows, macOS và Linux.
- Nhât quán trên các kiến trúc: có thể chạy mã nguồn của bạn với cùng một hành vi trên nhiều kiến trúc hệ thống, bao gồm x64, x86 và ARM.
- Các công cụ dòng lệnh: Bao gồm các công cụ dòng lệnh dễ sử dụng, có thể được sử dụng để phát triển cục bộ và trong các tình huống tích hợp liên tục.
- Triển khai linh hoạt: có thể cài đặt song song (cài đặt toàn người dùng hoặc toàn hệ thống). Có thể được sử dụng với các container Docker.
- Tương thích: .NET Core tương thích với .NET Framework, Xamarin và Mono, thông qua .NET Standard.
- Nguồn mở: Nền tảng .NET Core là nguồn mở, sử dụng giấy phép MIT và Apache 2. .NET Core là một dự án .NET Foundation.
- Được hỗ trợ bởi Microsoft: .NET Core được Microsoft hỗ trợ, theo Hỗ trợ .NET Core.

## 2. Thành phần của .NET Core

- .NET Core runtime: cung cấp một hệ thống kiểu, tải lấp ráp, trình thu gom rác, interop gốc và các dịch vụ cơ bản khác. Các thư viện khung .NET Core cung cấp các kiểu dữ liệu nguyên thủy, các kiểu thành phần ứng dụng và các tiện ích cơ bản.
- ASP.NET Core runtime: cung cấp khung để xây dựng các ứng dụng kết nối internet, điện toán đám mây hiện đại, chẳng hạn như ứng dụng web, ứng dụng IoT và phụ trợ di động.
- .NET Core SDK và trình biên dịch ngôn ngữ (Roslyn và F #) cho phép trải nghiệm nhà phát triển .NET Core.
- Dotnet command, được sử dụng để khởi chạy các ứng dụng .NET Core và các lệnh CLI. Nó chọn thời gian chạy và lưu trữ thời gian chạy, cung cấp chính sách tải lấp ráp và khởi chạy các ứng dụng và công cụ.

### 1.4.2. ASP.NET Core



*Hình 1.6. Logo ASP.NET Core*

#### 1. Khái niệm

Nó là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (cloud) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

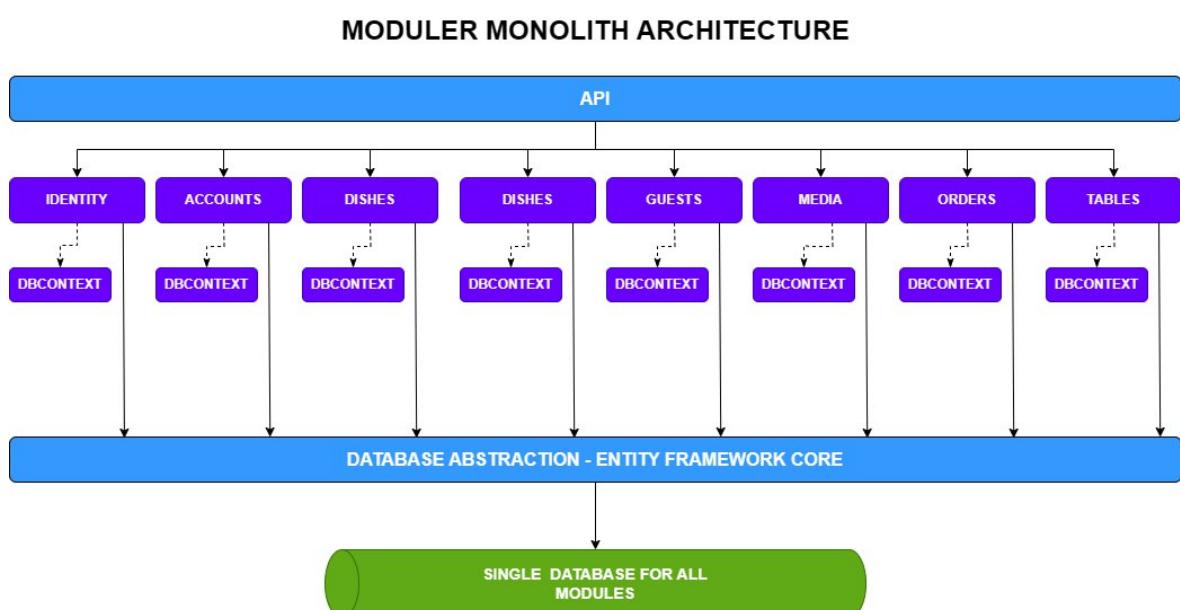
#### 2. Đặc điểm

- Hợp nhất việc xây dựng web UI và web APIs
- Tích hợp những client-side frameworks hiện đại và những luồng phát triển
- Hệ thống cấu hình dựa trên môi trường đám mây thật sự
- Dependency injection được xây dựng sẵn
- HTTP request được tối ưu nhẹ hơn
- Có thể host trên IIS hoặc self-host trong process của riêng bạn
- Được xây dựng trên .NET Core, hỗ trợ thực sự app versioning
- Chuyển các thực thể, thành phần, module như những NuGet packages
- Những công cụ mới để đơn giản hóa quá trình phát triển web hiện đại
- Xây dựng và chạy đa nền tảng (Windows, Mac và Linux)
- Mã nguồn mở và tập trung vào cộng đồng

## 1.5. Kiến trúc Monolithic

### 1.5.1. Kiến trúc phần mềm là gì?

Trong ngành công nghiệp phần mềm, khi chúng ta muốn phát triển một sản phẩm phần mềm thì chúng ta cần kiến trúc tốt bằng cách sử dụng kiến trúc đó, chúng ta có thể phát triển một sản phẩm chất lượng cao, sản phẩm có khả năng kiểm tra cao, khả năng mở rộng cao và khả năng bảo trì cao.



### *Hình 1.7. Kiến trúc Monolith*

#### **1.5.2. Định nghĩa**

“Kiến trúc phần mềm đơn giản là tổ chức của hệ thống, tổ chức này bao gồm tất cả các thành phần và cách chúng sẽ tương tác với nhau với môi trường mà chúng hoạt động và các nguyên tắc được sử dụng để xây dựng ứng dụng. Trong nhiều trường hợp, nó thường mang nghĩa sự phát triển của sản phẩm phần mềm trong tương lai”.

Kiến trúc phần mềm được thiết kế với sứ mệnh cụ thể rằng sản phẩm phải có khả năng kiểm thử cao, có khả năng mở rộng và bảo trì cao, đồng thời hành vi và cấu trúc của phần mềm phải tuân theo tiêu chuẩn và mang lại cho chúng ta kết quả tốt nhất có thể.

#### **1.5.3. Kiến trúc phần mềm trong kỹ thuật phần mềm**

Kiến trúc phần mềm trong công nghệ phần mềm trình bày cấu trúc của hệ thống. Kiến trúc tập trung vào các thành phần của hệ thống và mối quan hệ của chúng và cách các thành phần này sẽ tương tác với nhau và mang lại cho chúng ta sản phẩm phần mềm với chất lượng tốt nhất

#### **1.5.4. Kiến trúc nguyên khối (Monolithic) là gì?**

Kiến trúc nguyên khối (monolithic) là mô hình truyền thống trong thiết kế các sản phẩm phần mềm, kiến trúc nguyên khối có nghĩa là kết hợp ứng dụng thành một đơn vị (khối) và các thành phần ứng dụng được liên kết chặt chẽ với nhau và phụ thuộc lẫn nhau (nghĩa là để ứng dụng chạy trơn tru các thành phần đó phải có mặt).

Ứng dụng nguyên khối là ứng dụng một tầng, nghĩa là nhiều thành phần của ứng dụng được kết hợp thành một ứng dụng lớn duy nhất và kiến trúc của ứng dụng được kết hợp chặt chẽ

Trong ứng dụng Nguyên khối nếu chúng ta muốn thay đổi một thành phần thì các thành phần phụ thuộc khác cũng yêu cầu viết lại các đoạn code sau khi thay đổi, ứng dụng của chúng ta cần được biên dịch lại. Hơn nữa, chúng ta

chạy đi chạy lại quy trình kiểm tra và chúng ta lặp lại quy trình này sau mỗi lần thay đổi. Toàn bộ quá trình làm tăng chi phí và hạn chế sự linh hoạt và tốc độ của quá trình phát triển sản phẩm phần mềm.

### **1.5.5. Hiểu kiến trúc nguyên khôi với sản phẩm phần mềm**

Giả sử chúng ta có ứng dụng cung cấp dịch vụ cho người dùng cuối và các dịch vụ này như Cyber Security, Khoa học dữ liệu, Phát triển phần mềm và Kỹ thuật DevOps, đồng thời các dịch vụ có thể được bán cho người dùng cuối theo các tiêu chí nhất định và người dùng cuối trả tiền cho dịch vụ sử dụng phương thức thanh toán trực tuyến và trong ứng dụng, chúng ta cung cấp thông tin đăng nhập của người dùng và lịch sử giao dịch, lịch sử dự án và trong dự án, chúng ta có tính năng cuộc gọi video trực tuyến sẽ được sử dụng để xử lý cuộc gọi giữa các kỹ sư và tất cả các bên liên quan của dự án

Các thành phần của ứng dụng có thể được liệt kê như sau:

- Authentication
- Services
- Video Calling
- Online Payments
- Reports
- Jira Project API Integration

Giả sử rằng chúng ta phát triển các dự án như vậy bằng cách sử dụng kiến trúc nguyên khôi mang lại cho chúng ta sản phẩm được kết hợp chặt chẽ và xem xét dự án nếu chúng ta muốn thay đổi mô-đun Authentication thì các thành phần khác phụ thuộc vào mô-đun này sau khi thay đổi, thành phần phụ thuộc cần viết lại chức năng được sử dụng để duy trì trạng thái của người dùng và vì một thay đổi mà chúng ta cần kiểm tra lại toàn bộ ứng dụng và toàn bộ quá trình này làm tăng chi phí và thời gian phát triển và có thể làm giảm hiệu quả của ứng dụng.

### 1.5.6. Lợi ích của ứng dụng nguyên khôi

- Ứng dụng nguyên khôi có thể có thông lượng tốt hơn ứng dụng mô-đun.
- Một ứng dụng nhỏ với ít mô-đun hơn sẽ dễ kiểm tra và gỡ lỗi hơn từ đó có thể giảm chi phí phát triển.
- Kiến trúc nguyên khôi tốt hơn cho các ứng dụng nhẹ quy mô nhỏ.

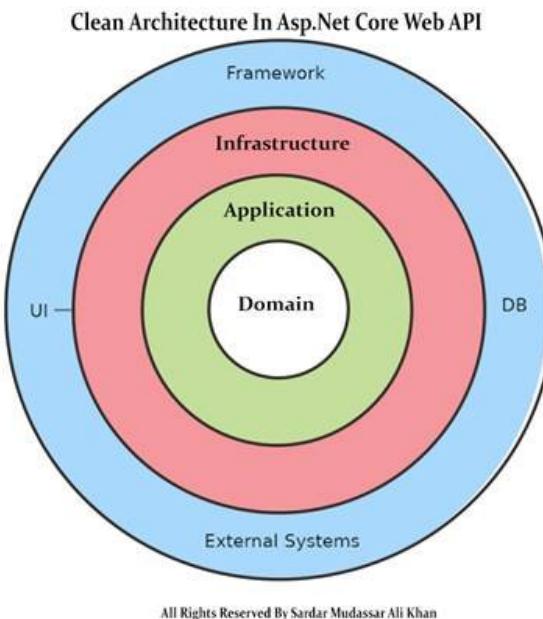
### 1.5.7. Hạn chế của ứng dụng nguyên khôi

- Kiến trúc nguyên khôi không phù hợp với các ứng dụng phức tạp.
- Khi độ phức tạp của ứng dụng tăng lên thì quy mô nhóm phát triển cũng tăng theo.
- Codebase của ứng dụng thường sẽ khó hiểu và khó sửa đổi
- Đối với bất kỳ ứng dụng nguyên khôi nào, nhà phát triển cần kiểm tra và triển khai toàn bộ ứng dụng sau một thay đổi nhỏ.
- Với ứng dụng nguyên khôi, chi phí và thời gian phát triển cũng tăng theo những thay đổi trong cấu trúc ứng dụng.
- Các ứng dụng nguyên khôi hạn chế về khả năng mở rộng và độ tin cậy
- Một module nhỏ, lớn có thể khiến cả ứng dụng bị sập.

## 1.6. Clean architecture

### 1.6.1. Clean Architecture là gì?

Clean Architecture bao gồm các tầng Domain Layer, Application Layer, Infrastructure Layer và Framework Layer. Trong đó, Domain Layer và Application Layer luôn là trung tâm của thiết kế và được gọi là lõi (core) của hệ thống. Phần lõi này độc lập với việc truy cập dữ liệu và các mối quan tâm về hạ tầng. Mục tiêu này có thể đạt được bằng cách sử dụng các Interface và sự trừu tượng trong lõi hệ thống, nhưng việc triển khai sẽ được thực hiện bên ngoài lõi hệ thống.



*Hình 1.8. Clean Architecture*

Các tầng trong Clean Architecture:

Clean Architecture bao gồm các tầng Domain Layer, Application Layer, Infrastructure Layer và Presentation Layer. Domain Layer và Application Layer luôn là trung tâm của thiết kế và được gọi là lõi của hệ thống.

Trong Clean Architecture, tất cả các phụ thuộc trong ứng dụng đều là độc lập/hướng vào trong (Independent/Inwards), và phần lõi của hệ thống không phụ thuộc vào bất kỳ tầng nào khác. Vì vậy, trong tương lai, nếu muốn thay đổi giao diện người dùng (UI) hoặc framework của hệ thống, bạn có thể thực hiện điều đó một cách dễ dàng vì các thành phần khác không phụ thuộc vào phần lõi của hệ thống.

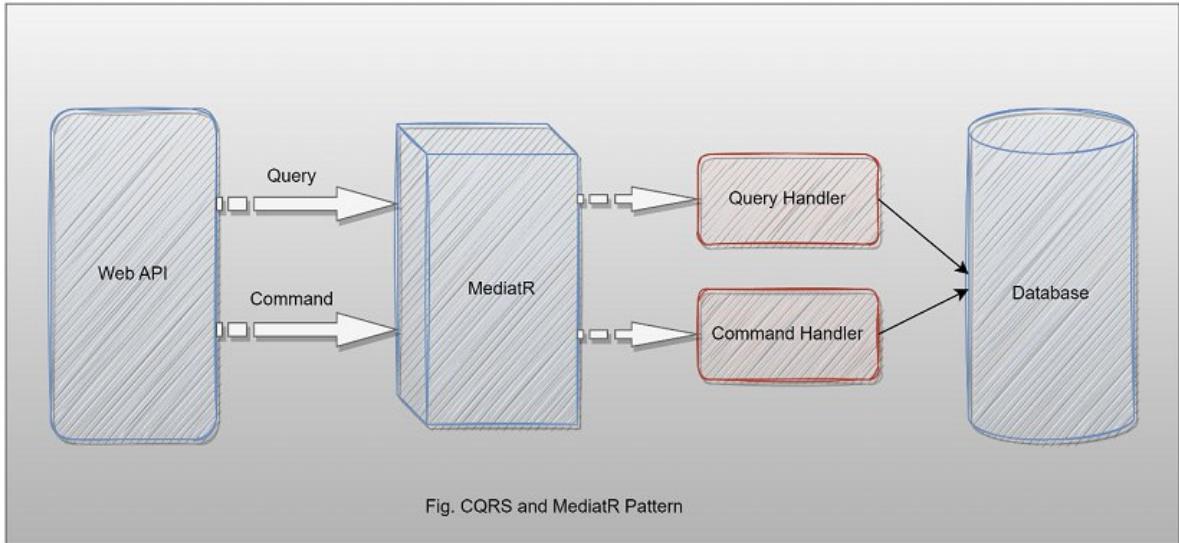
1. Domain Layer trong Clean Architecture chứa logic nghiệp vụ chính (enterprise logic), như các entity và các quy tắc của chúng. Tầng này nằm ở trung tâm của kiến trúc, nơi chứa các thực thể ứng dụng (application entities), bao gồm các lớp mô hình ứng dụng hoặc mô hình cơ sở dữ liệu. Sử dụng cách tiếp cận Code-First trong phát triển ứng dụng với ASP.NET Core, các thực thể này sẽ được dùng để tạo bảng trong cơ sở dữ liệu.

2. Application Layer chứa logic nghiệp vụ (business logic). Tất cả logic nghiệp vụ sẽ được viết trong tầng này. Ở đây, các interface của dịch vụ (service interfaces) được tách biệt khỏi phần triển khai (implementation) nhằm đảm bảo kết hợp lỏng lẻo (loose coupling) và tách biệt trách nhiệm (separation of concerns).
3. Infrastructure Layer chứa các đối tượng mô hình (model objects), nơi quản lý toàn bộ migration cơ sở dữ liệu và các đối tượng DbContext. Trong tầng này, các repository cho các đối tượng mô hình domain cũng được đặt.
4. Presentation Layer
  - Đối với API, Presentation Layer là nơi cung cấp dữ liệu đối tượng từ cơ sở dữ liệu thông qua HTTP request dưới dạng JSON Object.
  - Đối với các ứng dụng giao diện người dùng (UI), tầng này sẽ hiển thị dữ liệu bằng cách tiêu thụ các API.

#### **1.6.2. Lợi ích của Clean Architecture**

- Triển khai dễ dàng: Bạn có thể triển khai kiến trúc này với bất kỳ ngôn ngữ lập trình nào.
- Tính độc lập: Domain Layer và Application Layer luôn là trung tâm của thiết kế và không phụ thuộc vào các hệ thống bên ngoài.
- Thay đổi dễ dàng: Cho phép thay đổi các hệ thống bên ngoài mà không ảnh hưởng đến lõi của hệ thống.
- Môi trường kiểm thử cao: Dễ dàng kiểm thử mã nguồn của bạn nhanh chóng và hiệu quả.
- Sản phẩm chất lượng cao và có khả năng mở rộng tốt: Bạn có thể tạo ra một sản phẩm có chất lượng cao và dễ dàng mở rộng.

### 1.6.3. CQRS Pattern



*Hình 1.9. CQRS Pattern*

CQRS là viết tắt của Command and Query Responsibility Segregation (Phân tách trách nhiệm lệnh và truy vấn) và được sử dụng để tách biệt các thao tác đọc (queries) và ghi (commands).

- Queries thực hiện các thao tác đọc dữ liệu.
- Commands thực hiện các thao tác ghi dữ liệu như tạo (create), cập nhật (update), xóa (delete) và trả về dữ liệu.

#### 1. CQRS Pattern

Thông thường, trong các ứng dụng, chúng ta sử dụng một mô hình dữ liệu duy nhất để đọc và ghi dữ liệu, điều này hoạt động tốt trong các ứng dụng nhỏ và giúp thực hiện các thao tác CRUD dễ dàng. Tuy nhiên, khi ứng dụng trở nên phức tạp hơn, các truy vấn có thể trả về nhiều loại dữ liệu khác nhau dưới dạng các đối tượng, điều này khiến việc quản lý trở nên khó khăn với các DTO khác nhau. Đồng thời, cùng một mô hình cũng được sử dụng cho các thao tác ghi dữ liệu, dẫn đến mô hình trở nên phức tạp.

Ngoài ra, khi sử dụng cùng một mô hình cho cả thao tác đọc và ghi, việc quản lý bảo mật cũng trở nên khó khăn khi ứng dụng lớn, vì có thể xảy ra tình trạng lộ dữ liệu không đúng ngữ cảnh do khối lượng công việc trên cùng một mô hình.

CQRS giúp tách biệt các thao tác, làm cho ứng dụng dễ mở rộng và linh hoạt hơn ở quy mô lớn.

## 2. Khi nào nên sử dụng CQRS

- **Ứng dụng lớn và phức tạp:** Chúng ta nên sử dụng CQRS khi ứng dụng lớn và cần truy cập dữ liệu song song. CQRS giúp giảm xung đột hợp nhất (merge conflict) khi thực hiện nhiều thao tác với dữ liệu.
- **Mô hình dữ liệu phức tạp:** Trong ngữ cảnh Domain-Driven Design (DDD), nếu mô hình dữ liệu domain phức tạp và cần thực hiện nhiều thao tác ưu tiên như xác thực (validation) và thực thi logic nghiệp vụ, thì CQRS cung cấp tính nhất quán và linh hoạt cần thiết.

## CHƯƠNG 2 : PHÂN TÍCH ĐẶC TẢ YÊU CẦU PHẦN MỀM

### 2.1. Khảo sát bài toán

#### 2.1.1. Khảo sát nhà hàng Yaya

Phương pháp khảo sát: Quan sát thực tế

Ưu điểm:

- Hiểu đúng nhu cầu thực tế,
- Nắm bắt được quy trình nghiệp vụ
- Phát hiện các vấn đề ngầm
- Hiểu bối cảnh thực tế sử dụng
- Tăng tin cậy cho yêu cầu thiết kế

Nhược điểm:

- Mất nhiều thời gian chuẩn bị và thực hiện:
- Có thể bị hạn chế quyền quan sát:
- Khó thu thập đầy đủ tất cả tình huống:
- Cần kỹ năng phân tích hành vi cao
- Chi phí tốn kém nếu quy mô lớn

Kết quả khảo sát:

Trong quá trình quan sát hoạt động tại nhà hàng Yaya, chúng tôi ghi nhận nhiều bất cập trong công tác quản lý và phục vụ khách hàng:

*Order đồ ăn:* Khi khách đến, nhân viên ghi order bằng giấy tay hoặc điện thoại cá nhân, dẫn đến dễ xảy ra nhầm lẫn món, thất lạc đơn hàng hoặc mất thời gian xác nhận lại.

*Đặt bàn:* Việc đặt bàn chủ yếu qua điện thoại hoặc trao đổi trực tiếp, không có hệ thống kiểm soát chỗ ngồi, gây ra tình trạng trùng lịch, thiếu bàn hoặc bỏ sót đặt chỗ.

*Quản lý nhân viên:* Nhà hàng gặp khó khăn trong việc theo dõi ca làm, giờ giấc, và hiệu suất làm việc của từng nhân viên. Việc phân chia nhiệm vụ cũng thiếu minh bạch và đồng bộ.

*Lưu trữ hóa đơn và dữ liệu khách hàng:* Hóa đơn được ghi chép thủ công hoặc lưu trữ rời rạc trên nhiều file, gây khó khăn trong việc tra cứu, tổng hợp doanh thu và chăm sóc khách hàng sau này.

## Kết luận

Từ những bất cập trên, rõ ràng nhà hàng Yaya rất cần một phần mềm web hỗ trợ toàn diện. Phần mềm này sẽ giúp tự động hóa việc order, đặt bàn, quản lý nhân viên, lưu trữ hóa đơn và dữ liệu khách hàng một cách khoa học, nhanh chóng và chính xác. Qua đó, nhà hàng có thể nâng cao hiệu suất làm việc, giảm sai sót, cải thiện trải nghiệm khách hàng và phát triển bền vững hơn.

### 2.1.2. Khảo sát các ứng dụng web đặt món ăn ngay tại nhà hàng, quán ăn trên thị trường hiện nay.

Hiện nay, ứng dụng web đặt món ăn trực tiếp tại nhà hàng đang ngày càng phổ biến trên thị trường, đáp ứng nhu cầu hiện đại hóa và nâng cao trải nghiệm của khách hàng. Các ứng dụng này cho phép thực khách dễ dàng duyệt menu, đặt món ăn, và thanh toán trực tuyến ngay tại bàn, tiết kiệm thời gian và tăng cường sự tiện lợi. Một số ứng dụng còn tích hợp các tính năng như gọi phục vụ, đánh giá món ăn, và chương trình khuyến mãi, giúp tăng sự tương tác giữa khách hàng và nhà hàng. Trên thị trường, các nhà phát triển thường tập trung vào thiết kế giao diện thân thiện, tích hợp công nghệ QR code, và đảm bảo bảo mật thông tin thanh toán. Tuy nhiên, thách thức lớn nhất vẫn là khả năng tương thích với hệ thống hiện tại của nhà hàng và việc đào tạo nhân viên sử dụng hiệu quả. Sự phát triển của loại hình này không chỉ giúp nhà hàng tối ưu hóa quy trình phục vụ mà còn nâng cao sự hài lòng và trung thành của khách hàng.

## **2.2. Khảo sát hệ thống.**

### **2.2.1. Giới thiệu chung về hệ thống.**

Ứng dụng web nhà hàng Yaya là giải pháp hiện đại, được thiết kế nhằm nâng cao hiệu quả quản lý và mang lại trải nghiệm tiện lợi, nhanh chóng cho cả khách hàng và nhân viên. Với giao diện thân thiện và nhiều tính năng nổi bật, ứng dụng giúp nhà hàng tối ưu hóa quy trình vận hành và phục vụ.

Ứng dụng cho phép khách hàng có thể đăng ký tài khoản, đăng nhập, đăng xuất, đổi mật khẩu và dễ dàng quét mã QR tại bàn để xem thực đơn, đặt món ăn mà không cần sự hỗ trợ trực tiếp từ nhân viên.

Đối với nhà hàng, ứng dụng hỗ trợ quản lý bàn ăn, cho phép thêm, sửa, xóa bàn, , ẩn bàn, và chuyển đổi trạng thái sử dụng linh hoạt. Tính năng quản lý món ăn giúp cập nhật thực đơn nhanh chóng, từ thêm, sửa, xóa đến chuyển đổi trạng thái món ăn. Ngoài ra, quản lý đơn hàng hỗ trợ theo dõi và thay đổi trạng thái đơn hàng, thậm chí đặt đơn hộ khách khi cần.

Ứng dụng còn cung cấp biểu đồ phân tích doanh thu, giúp nhà hàng dễ dàng theo dõi hiệu quả kinh doanh theo thời gian thực. Quản lý tài khoản nhân viên với phân quyền chi tiết giúp đảm bảo hoạt động ổn định và bảo mật thông tin. Hơn nữa, tính năng realtime trạng thái món ăn cho phép khách hàng biết được tình trạng chuẩn bị của món, mang lại trải nghiệm dịch vụ chuyên nghiệp và minh bạch.

Ứng dụng đặt món ăn này không chỉ hỗ trợ hoạt động quản lý nhà hàng hiệu quả mà còn giúp khách hàng tận hưởng bữa ăn một cách thuận tiện, hiện đại và thoải mái hơn bao giờ hết.

## 2.2.2. Các tác nhân hệ thống

### 2.2.2.1. Giới thiệu các tác nhân

- Khách hàng có vai trò là người sử dụng chính của hệ thống, khách hàng có thể đăng nhập, đặt món, xem đơn hàng, trạng thái món ăn.
- Quản trị viên có vai trò là người quản trị hệ thống, quản trị viên có quyền quản lý tất cả các dữ liệu trong hệ thống.

### 2.2.2.2 Tác nhân và vai trò tương ứng

Tác nhân	Hoạt động
Khách hàng	1. Đăng nhập 2. Đặt món 3. Xem món ăn 4. Xem đơn hàng
Quản trị viên	1. Đăng nhập 2. Quản lý món ăn 3. Quản lý bàn ăn 4. Quản lý đơn hàng 5. Quản lý tài khoản 6. Xem doanh thu

## 2.3. Yêu cầu hệ thống

### 2.3.1. Yêu cầu chức năng

- Đăng nhập - Đăng xuất - Đổi mật khẩu
  - Cung cấp chức năng đăng nhập cho nhân viên, quản trị viên và khách hàng.
  - Hỗ trợ đăng xuất an toàn để bảo vệ thông tin người dùng.
  - Cho phép người dùng thay đổi mật khẩu cá nhân sau khi xác thực thông tin.

- Quản lý đơn hàng
  - Hiển thị danh sách các đơn hàng, bao gồm trạng thái hiện tại.
  - Cho phép nhân viên thay đổi trạng thái đơn hàng (đang chuẩn bị, đã phục vụ, đã thanh toán,...).
  - Hỗ trợ tính năng đặt đơn hàng hộ khách, bao gồm nhập thông tin và xác nhận đơn hàng.
- Quản lý bàn ăn
  - Cung cấp các chức năng thêm, sửa, xóa thông tin bàn ăn.
  - Cho phép đổi mã QR của bàn ăn và ẩn bàn khi không sử dụng.
  - Hỗ trợ thay đổi trạng thái bàn (có sẵn, ẩn).
- Quản lý món ăn
  - Hỗ trợ thêm mới món ăn với thông tin chi tiết như tên, giá, mô tả và hình ảnh.
  - Cung cấp các chức năng sửa, xóa món ăn.
  - Cho phép thay đổi trạng thái món ăn (hiển thị/ẩn trên thực đơn).
- Biểu đồ phân tích doanh thu
  - Hiển thị báo cáo trực quan về doanh thu theo thời gian (ngày, tuần, tháng).
  - Phân tích số lượng đơn hàng và món ăn được bán ra.
- Quản lý tài khoản nhân viên
  - Hỗ trợ thêm, sửa, xóa tài khoản nhân viên.
  - Cung cấp tính năng phân quyền để kiểm soát các chức năng mà nhân viên được phép truy cập.
- Cập nhật trạng thái món ăn
  - Cập nhật trạng thái món ăn theo thời gian thực.
  - Cho phép nhân viên theo dõi trạng thái để phối hợp tốt hơn trong quy trình phục vụ.
- Khách hàng: Đăng ký tài khoản, đặt món ăn
  - Cho phép khách hàng tạo tài khoản để lưu thông tin đặt hàng.

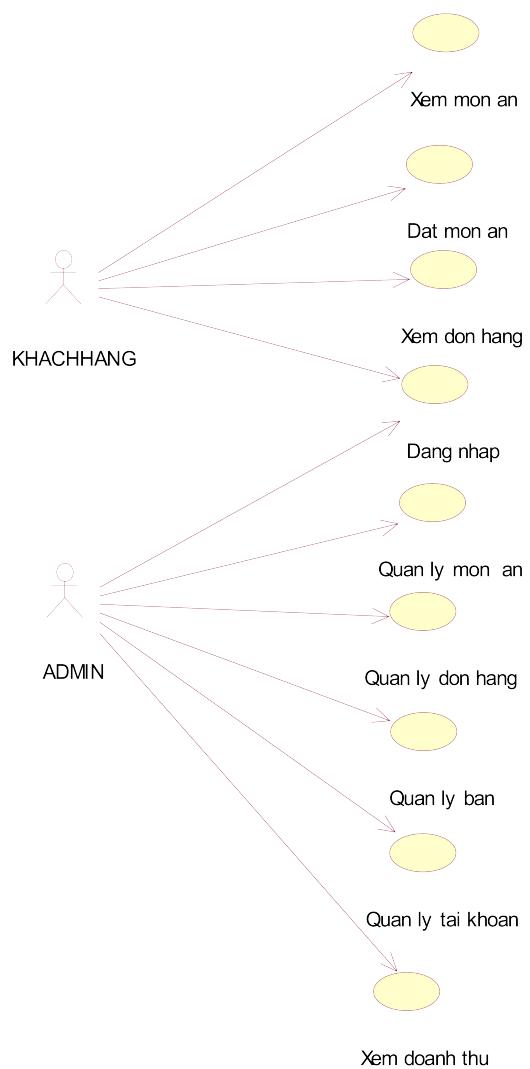
- Hỗ trợ quét mã QR tại bàn để xem thực đơn và đặt món ăn trực tiếp.
- Đăng tải hình ảnh liên quan
  - Cho phép nhân viên và quản trị viên tải lên hình ảnh minh họa món ăn, bàn ăn hoặc các tài liệu liên quan.
  - Hỗ trợ lưu trữ và quản lý hình ảnh.

### **2.3.2. Yêu cầu phi chức năng**

- Ngôn ngữ trên hệ thống: Sử dụng ngôn ngữ tiếng Việt.
- Môi trường cài đặt: Hệ thống được thiết kế để tương thích với các trình duyệt web phổ biến và hệ điều hành khác nhau, đảm bảo rằng người dùng có thể truy cập và sử dụng hệ thống một cách dễ dàng : Windows, MacOS,Linux và tương thích với các thiết bị di động thông minh như iphone, ipad và các loại smartphone.
- Độ tin cậy và bảo mật: Hệ thống đảm bảo rằng thông tin của khách hàng được bảo mật và không bị rò rỉ hoặc bị truy cập trái phép.
- Tốc độ và hiệu suất: Hệ thống được thiết kế để hoạt động nhanh chóng và đáp ứng nhu cầu của người dùng trong thời gian ngắn nhất có thể .
- Thiết kế đẹp và dễ sử dụng: Giao diện website phải đẹp, gần gũi và dễ sử dụng, đảm bảo rằng người dùng có thể tìm thấy thông tin và sản phẩm một cách dễ dàng và thuận tiện.
- Độ khả dụng: Hệ thống nên được thiết kế để có thể hoạt động 24/7 mà không gặp sự cố, đảm bảo rằng người dùng có thể truy cập và sử dụng hệ thống bất cứ khi nào họ muốn.
- Khả năng hồi phục: Khả năng hồi phục từ đê cập đến khả năng vượt qua những thời điểm quan trọng và thích nghi sau khi trải qua một số tình huống bất ngờ. Nó cũng chỉ ra việc trở lại bình thường.

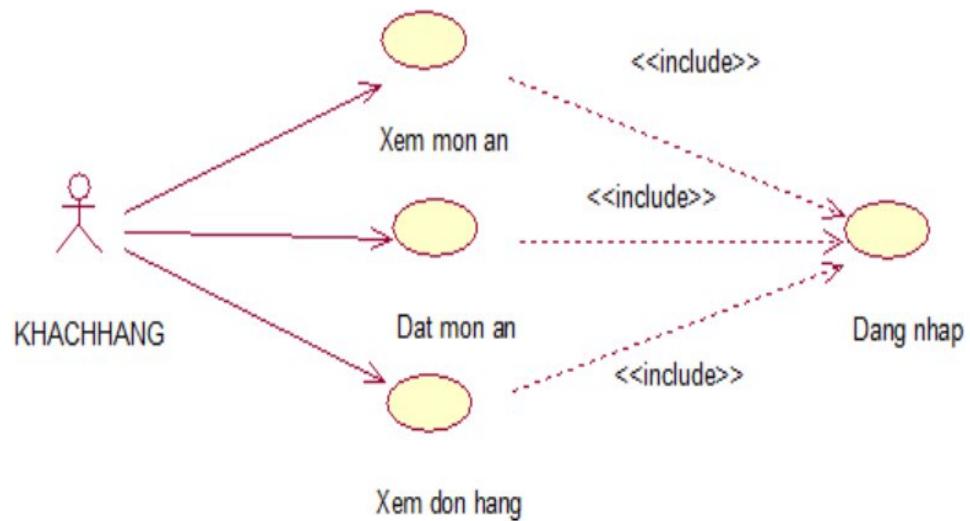
## **2.4. Biểu đồ use case.**

### **2.4.1. Biểu đồ use case tổng quan**

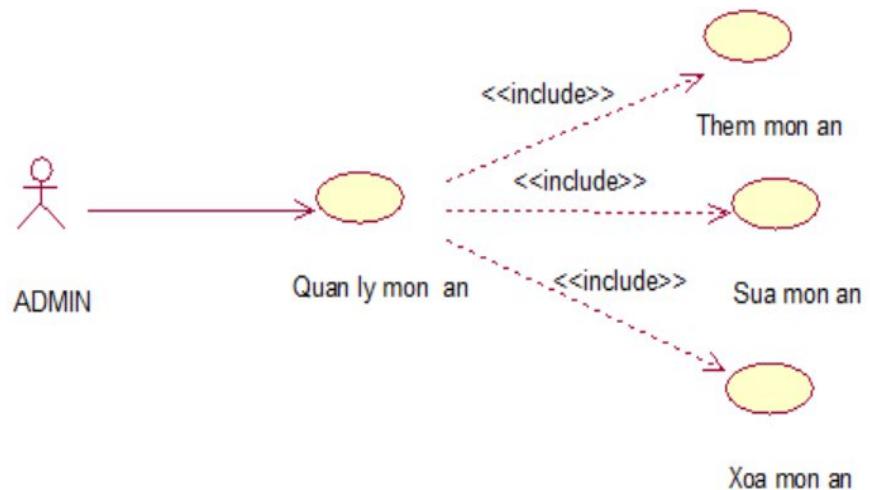


Hình 2.1. Biểu đồ use case tổng quan.

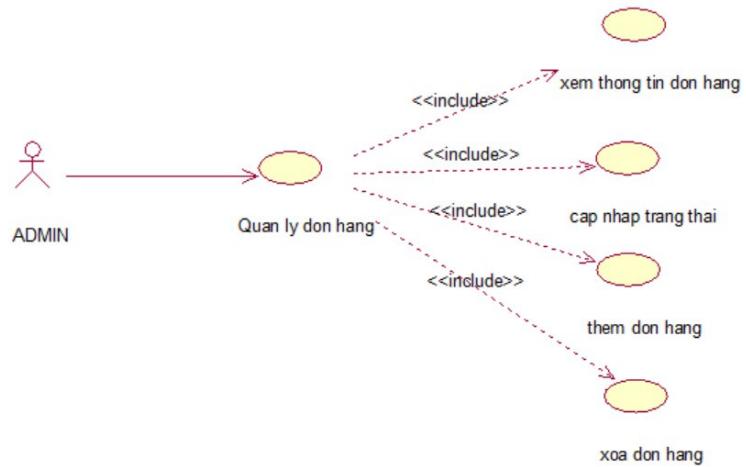
### 2.4.2. Biểu đồ phân rã một số use case.



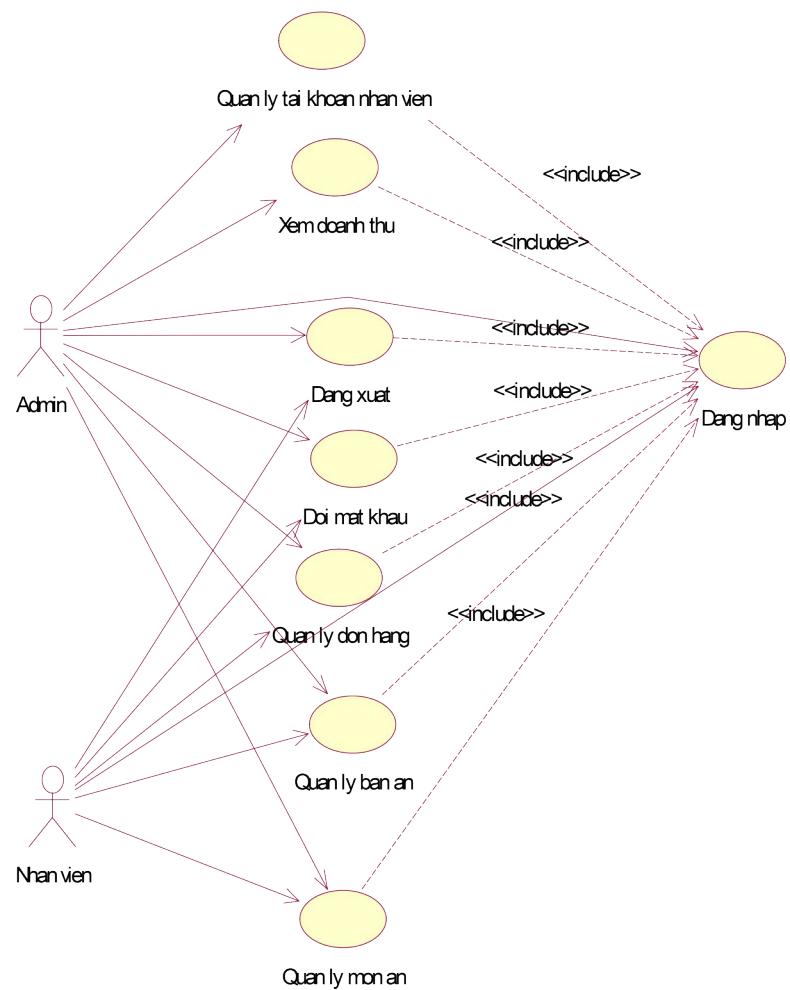
Hình 2.2. Biểu đồ phân rã use case phía frontend.



Hình 2.3. Biểu đồ phân rã use case quản lý món ăn.



Hình 2.4. Biểu đồ phân rã use case quản lý đơn hàng.



Hình 2.5. Biểu đồ phân rã use case phía backend.

## 2.5. Mô tả chi tiết một số use case.

### 2.5.1. Mô tả chi tiết use case đăng nhập

Mã Use case	UC00	Tên Use case	Đăng nhập																	
Tác nhân	Khách hàng																			
Mô tả	UC này cho phép khách hàng đăng nhập tài khoản vào hệ thống																			
Sự kiện kích hoạt chức năng	UC bắt đầu khi khách hàng kích vào ô đăng nhập ở phía trên giao diện trang chủ																			
Tiền điều kiện	Không có																			
Luồng sự kiện chính	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Khách hàng</td><td>Truy cập vào trang đăng nhập của website.</td></tr> <tr> <td>2</td><td>Hệ thống</td><td>Hiển thị giao diện đăng nhập yêu cầu người dùng nhập tên đăng nhập và mật khẩu.</td></tr> <tr> <td>3</td><td>Khách hàng</td><td>Nhập thông tin tên đăng nhập và mật khẩu của mình.</td></tr> <tr> <td>4</td><td>Khách hàng</td><td>Nhấn nút “Đăng nhập”.</td></tr> <tr> <td>5</td><td>Hệ thống</td><td>Kiểm thông tin của người dùng và cho phép đăng nhập vào hệ thống nếu thông tin hợp lệ</td></tr> </tbody> </table>	#	Thực hiện bởi	Hành động	1	Khách hàng	Truy cập vào trang đăng nhập của website.	2	Hệ thống	Hiển thị giao diện đăng nhập yêu cầu người dùng nhập tên đăng nhập và mật khẩu.	3	Khách hàng	Nhập thông tin tên đăng nhập và mật khẩu của mình.	4	Khách hàng	Nhấn nút “Đăng nhập”.	5	Hệ thống	Kiểm thông tin của người dùng và cho phép đăng nhập vào hệ thống nếu thông tin hợp lệ	
#	Thực hiện bởi	Hành động																		
1	Khách hàng	Truy cập vào trang đăng nhập của website.																		
2	Hệ thống	Hiển thị giao diện đăng nhập yêu cầu người dùng nhập tên đăng nhập và mật khẩu.																		
3	Khách hàng	Nhập thông tin tên đăng nhập và mật khẩu của mình.																		
4	Khách hàng	Nhấn nút “Đăng nhập”.																		
5	Hệ thống	Kiểm thông tin của người dùng và cho phép đăng nhập vào hệ thống nếu thông tin hợp lệ																		

Luồng sự kiện thay thế	#	Thực hiện bởi	Hành động
	5a	Hệ thống	Nếu người dùng nhập sai tên đăng nhập hoặc mật khẩu, hệ thống sẽ hiển thị thông báo lỗi “Thông tin không hợp lệ!” và hệ thống đưa người dùng quay lại bước 3.
Hậu điều kiện	Không		

Bảng 2.1 Bảng mô tả chi tiết use case đăng nhập



Hình 2.6 Biểu đồ minh họa use case đăng nhập

### 2.5.2. Mô tả chi tiết use case Xem món ăn

Mã Use case	UC01	Tên Use case	Xem món ăn
Tác nhân	Khách hàng		
Mô tả	UC này cho phép khách hàng xem món ăn.		
Sự kiện kích hoạt	UC bắt đầu khi khách hàng kích vào hình món ăn ở giao diện menu		
Tiền điều kiện	Không có		
Luồng sự kiện chính		#	Thực hiện bởi
		1	Khách hàng
			Chọn món ăn muốn xem.

	2	Hệ thống	Hiển thị thông tin món ăn.
Luồng sự kiện thay thế	#	Thực hiện bởi	Hành động
	2a	Hệ thống	Hiển thị một thông báo lỗi và use case kết thúc nếu không thể kết nối với cơ sở dữ liệu
Hậu điều kiện	Không		

Bảng 2.2 Bảng mô tả chi tiết use case xem món ăn



Hình 2.7 Biểu đồ minh họa use case xem món ăn

### 2.5.3. Mô tả chi tiết use case đặt món ăn

Mã Use case	UC02	Tên Use case	Đặt món ăn						
Tác nhân	Khách hàng								
Mô tả	Khách hàng tiến hành đặt món ăn cho đơn hàng								
Sự kiện kích hoạt chức năng	Người dùng kích vào nút “Menu” trên giao diện.								
Tiền điều kiện	Khách hàng đã đăng nhập vào tài khoản của mình.								
Luồng sự kiện chính	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Khách hàng</td> <td>Chọn món ăn muốn mua và</td> </tr> </tbody> </table>			#	Thực hiện bởi	Hành động	1	Khách hàng	Chọn món ăn muốn mua và
#	Thực hiện bởi	Hành động							
1	Khách hàng	Chọn món ăn muốn mua và							

		chọn số lượng sản phẩm.						
2	Khách hàng	Nhấn nút “Đặt hàng”						
3	Hệ thống	Hiện thị giao diện đặt hàng cho khách hàng						
4	Khách hàng	Cung cấp thông tin đặt hàng cần thiết và chọn phương thức thanh toán						
5	Hệ thống	Xác nhận thông tin khách hàng và đưa khách hàng đến giao diện xem đơn hàng						
Luồng sự kiện thay thế	<table border="1"> <thead> <tr> <th>#</th><th>Thực hiện bởi</th><th>Hành động</th></tr> </thead> <tbody> <tr> <td>5a</td><td>Hệ thống</td><td>Nếu hệ thống gặp sự cố hoặc ngân hàng bảo trì đưa ra thông báo lỗi</td></tr> </tbody> </table>		#	Thực hiện bởi	Hành động	5a	Hệ thống	Nếu hệ thống gặp sự cố hoặc ngân hàng bảo trì đưa ra thông báo lỗi
#	Thực hiện bởi	Hành động						
5a	Hệ thống	Nếu hệ thống gặp sự cố hoặc ngân hàng bảo trì đưa ra thông báo lỗi						
Hậu điều kiện	Hệ thống đã ghi nhận đơn hàng cập nhật trạng thái đơn hàng và lưu thông tin thanh toán tương ứng.							

Bảng 2.3 Bảng mô tả chi tiết use case đặt món ăn



Hình 2.8 Biểu đồ minh họa use case đặt món ăn

#### 2.5.4. Mô tả chi tiết use case xem đơn hàng

Mã Use case	UC03	Tên Use case	Xem đơn hàng									
Tác nhân	Khách hàng											
Mô tả	UC này cho phép khách hàng xem đơn hàng.											
Sự kiện kích hoạt chức năng	UC bắt đầu khi khách hàng kích vào hình món ăn ở giao diện menu											
Tiền điều kiện	Khách hàng đã đăng nhập vào tài khoản của mình											
Luồng sự kiện chính	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Khách hàng</td><td>Chọn đơn hàng trên thanh công cụ.</td></tr> <tr> <td>2</td><td>Hệ thống</td><td>Hiển thị thông tin trạng thái các đơn hàng.</td></tr> </tbody> </table>			#	Thực hiện bởi	Hành động	1	Khách hàng	Chọn đơn hàng trên thanh công cụ.	2	Hệ thống	Hiển thị thông tin trạng thái các đơn hàng.
#	Thực hiện bởi	Hành động										
1	Khách hàng	Chọn đơn hàng trên thanh công cụ.										
2	Hệ thống	Hiển thị thông tin trạng thái các đơn hàng.										
Luồng sự kiện thay thế	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>2a</td><td>Hệ thống</td><td>Hiển thị một thông báo lỗi và use case kết thúc nếu không thể kết nối với cơ sở dữ liệu</td></tr> <tr> <td>2b</td><td>Hệ thống</td><td>Nếu không có đơn hàng nào thì hiển thị thông báo</td></tr> </tbody> </table>			#	Thực hiện bởi	Hành động	2a	Hệ thống	Hiển thị một thông báo lỗi và use case kết thúc nếu không thể kết nối với cơ sở dữ liệu	2b	Hệ thống	Nếu không có đơn hàng nào thì hiển thị thông báo
#	Thực hiện bởi	Hành động										
2a	Hệ thống	Hiển thị một thông báo lỗi và use case kết thúc nếu không thể kết nối với cơ sở dữ liệu										
2b	Hệ thống	Nếu không có đơn hàng nào thì hiển thị thông báo										
Hậu điều kiện	Không											

Bảng 2.4 Bảng mô tả chi tiết use case xem đơn hàng



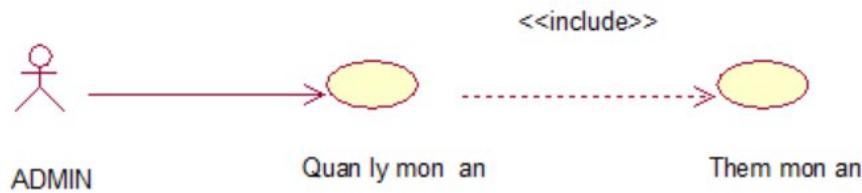
Hình 2.9 Biểu đồ minh họa use case xem đơn hàng

### 2.5.5. Mô tả chi tiết use case thêm món ăn

Mã Use case	UC04	Tên Use case	Thêm món ăn																		
Tác nhân	Admin																				
Mô tả	UC này cho phép người quản trị quản lý hệ thống thêm sản phẩm món ăn.																				
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người quản trị kích vào nút “Thêm món ăn” ở trang quản trị món ăn																				
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống, có quyền admin																				
Luồng sự kiện chính	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Admin</td><td>Chọn Thêm món ăn</td></tr> <tr> <td>2</td><td>Hệ thống</td><td>Hiển thị form nhập thông tin món ăn</td></tr> <tr> <td>3</td><td>Admin</td><td>Thêm thông tin món ăn</td></tr> <tr> <td>4</td><td>Admin</td><td>Nhấn nút “Thêm”</td></tr> <tr> <td>5</td><td>Hệ thống</td><td>Kiểm tra tính hợp lệ của thông tin</td></tr> </tbody> </table>	#	Thực hiện bởi	Hành động	1	Admin	Chọn Thêm món ăn	2	Hệ thống	Hiển thị form nhập thông tin món ăn	3	Admin	Thêm thông tin món ăn	4	Admin	Nhấn nút “Thêm”	5	Hệ thống	Kiểm tra tính hợp lệ của thông tin		
#	Thực hiện bởi	Hành động																			
1	Admin	Chọn Thêm món ăn																			
2	Hệ thống	Hiển thị form nhập thông tin món ăn																			
3	Admin	Thêm thông tin món ăn																			
4	Admin	Nhấn nút “Thêm”																			
5	Hệ thống	Kiểm tra tính hợp lệ của thông tin																			
Luồng sự kiện thay thế	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>5a</td><td>Hệ thống</td><td>Nếu thông tin hợp lệ, hệ thống lưu thông tin món ăn vào cơ sở dữ liệu rồi hiển thị thông báo</td></tr> <tr> <td>5b</td><td>Hệ thống</td><td>Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.</td></tr> </tbody> </table>	#	Thực hiện bởi	Hành động	5a	Hệ thống	Nếu thông tin hợp lệ, hệ thống lưu thông tin món ăn vào cơ sở dữ liệu rồi hiển thị thông báo	5b	Hệ thống	Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.											
#	Thực hiện bởi	Hành động																			
5a	Hệ thống	Nếu thông tin hợp lệ, hệ thống lưu thông tin món ăn vào cơ sở dữ liệu rồi hiển thị thông báo																			
5b	Hệ thống	Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.																			

Hậu điều kiện	Không có
---------------	----------

Bảng 2.5 Bảng mô tả chi tiết use case thêm món ăn



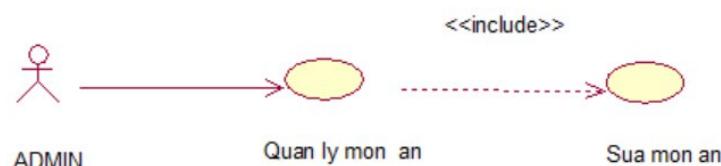
Hình 2.10 Biểu đồ minh họa use case thêm món ăn

### 2.5.6. Mô tả chi tiết use case sửa món ăn

Mã Use case	UC05	Tên Use case	Sửa món ăn																	
Tác nhân	Admin																			
Mô tả	UC này cho phép người quản trị quản lý hệ thống sửa thông tin món ăn.																			
Sự kiện kích hoạt	UC này bắt đầu khi người quản trị kích vào nút “Sửa” ở thông tin món ăn																			
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống, có quyền admin																			
Luồng sự kiện chính	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Admin</td><td>Chọn món ăn cần sửa</td></tr> <tr> <td>2</td><td>Hệ thống</td><td>Hiển thị form thông tin món ăn</td></tr> <tr> <td>3</td><td>Admin</td><td>Chỉnh sửa thông tin món ăn</td></tr> <tr> <td>4</td><td>Admin</td><td>Nhấn nút “Sửa”</td></tr> <tr> <td>5</td><td>Hệ thống</td><td>Kiểm tra tính hợp lệ của thông tin</td></tr> </tbody> </table>	#	Thực hiện bởi	Hành động	1	Admin	Chọn món ăn cần sửa	2	Hệ thống	Hiển thị form thông tin món ăn	3	Admin	Chỉnh sửa thông tin món ăn	4	Admin	Nhấn nút “Sửa”	5	Hệ thống	Kiểm tra tính hợp lệ của thông tin	
#	Thực hiện bởi	Hành động																		
1	Admin	Chọn món ăn cần sửa																		
2	Hệ thống	Hiển thị form thông tin món ăn																		
3	Admin	Chỉnh sửa thông tin món ăn																		
4	Admin	Nhấn nút “Sửa”																		
5	Hệ thống	Kiểm tra tính hợp lệ của thông tin																		
Luồng sự kiện thay thế	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>5a</td><td>Hệ thống</td><td>Nếu thông tin hợp lệ, hệ thống cập nhật thông tin món ăn</td></tr> </tbody> </table>	#	Thực hiện bởi	Hành động	5a	Hệ thống	Nếu thông tin hợp lệ, hệ thống cập nhật thông tin món ăn													
#	Thực hiện bởi	Hành động																		
5a	Hệ thống	Nếu thông tin hợp lệ, hệ thống cập nhật thông tin món ăn																		

	5b	Hệ thống	Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.
Hậu điều kiện	Không có		

Bảng 2.6 Bảng mô tả chi tiết use case sửa món ăn



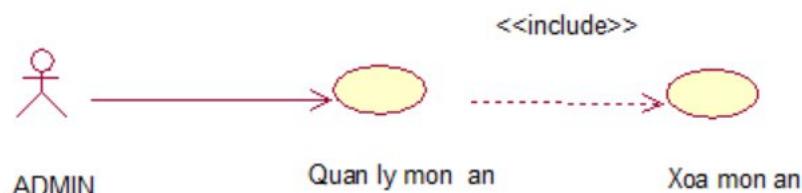
Hình 2.11 Biểu đồ minh họa use case sửa món ăn

### 2.5.7. Mô tả chi tiết use case xóa món ăn

Mã Use case	UC06	Tên Use case	Xóa món ăn
Tác nhân	Admin		
Mô tả	UC này cho phép người quản trị quản lý hệ thống xóa thông tin món ăn.		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người quản trị kích vào nút “Xóa” ở thông tin món ăn		
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống, có quyền admin		
Luồng sự kiện chính	#	Thực hiện bởi	Hành động
	1	Admin	Chọn sản phẩm cần xóa
	2	Hệ thống	Hiển thị form xác nhận xóa
	3	Admin	Chọn “Xóa”
	4	Hệ thống	Xóa sản phẩm khỏi cơ sở dữ liệu
Luồng sự kiện thay thế	#	Thực hiện bởi	Hành động
	4a	Hệ thống	Nếu xóa thành công, hệ thống hiển thị thông báo

	4b	Hệ thống	Nếu xóa không thành công hệ thống hiển thị thông báo lỗi
Hậu điều kiện	Không có		

Bảng 2.7 Bảng mô tả chi tiết use case xóa món ăn



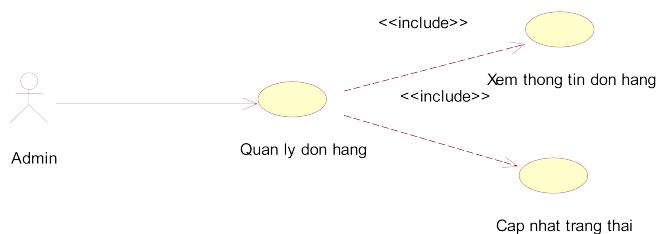
Hình 2.12 Biểu đồ minh họa use case xóa món ăn

### 2.5.8. Mô tả chi tiết use case quản lý đơn hàng

Mã Use case	UC07	Tên Use case	Quản lý đơn hàng															
Tác nhân	Admin																	
Mô tả	UC này cho phép người quản trị quản lý hệ thống xem và cập nhật trạng thái đơn hàng																	
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người quản trị kích vào nút “Quản lý đơn hàng” ở menu admin																	
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống, có quyền admin																	
Luồng sự kiện chính	<table border="1"> <thead> <tr> <th>#</th> <th>Thực hiện bởi</th> <th>Hành động</th> </tr> </thead> <tbody> <tr> <td>1</td><td>Admin</td><td>Chọn đơn hàng cần xem</td></tr> <tr> <td>2</td><td>Hệ thống</td><td>Hiển thị chi tiết hóa đơn bao gồm các thông tin sản phẩm được đặt và thông tin thanh toán</td></tr> <tr> <td>3</td><td>Admin</td><td>Cập nhật trạng thái đơn hàng</td></tr> <tr> <td>4</td><td>Hệ thống</td><td>Thay đổi trạng thái đơn hàng tương ứng và thông báo</td></tr> </tbody> </table>	#	Thực hiện bởi	Hành động	1	Admin	Chọn đơn hàng cần xem	2	Hệ thống	Hiển thị chi tiết hóa đơn bao gồm các thông tin sản phẩm được đặt và thông tin thanh toán	3	Admin	Cập nhật trạng thái đơn hàng	4	Hệ thống	Thay đổi trạng thái đơn hàng tương ứng và thông báo		
#	Thực hiện bởi	Hành động																
1	Admin	Chọn đơn hàng cần xem																
2	Hệ thống	Hiển thị chi tiết hóa đơn bao gồm các thông tin sản phẩm được đặt và thông tin thanh toán																
3	Admin	Cập nhật trạng thái đơn hàng																
4	Hệ thống	Thay đổi trạng thái đơn hàng tương ứng và thông báo																

Luồng sự kiện thay thế Luồng sự kiện	#	Thực hiện bởi	Hành động
	3a	Admin	Nếu đơn hàng đã được thanh toán thì bấm “Thanh toán tất cả” và cập nhật trạng thái thanh toán thành “đã thanh toán”.
Hậu điều kiện	Không có		

Bảng 2.8 Bảng mô tả chi tiết use case Quản lý đơn hàng



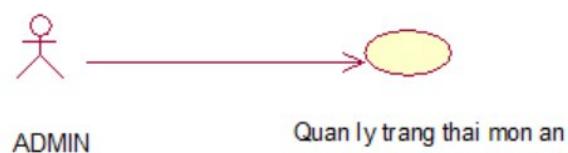
Hình 2.13 Biểu đồ minh họa chi tiết use case quản lý đơn hàng

### 2.5.9. Mô tả chi tiết use case quản lý trạng thái món ăn

Mã Use case	UC08	Tên Use case	Quản lý trạng thái món ăn
Tác nhân	Admin		
Mô tả	UC này cho phép người quản trị quản lý hệ thống xem và cập nhật trạng thái món ăn trong đơn hàng.		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người quản trị kích vào nút “Quản lý đơn hàng” ở menu admin và kíc “Trạng thái”		
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống, có quyền admin		
Luồng sự kiện chính	#	Thực hiện bởi	Hành động
	1	Admin	Chọn trạng thái món ăn cần xem
	2	Hệ thống	Hiển thị chi tiết đơn đặt hàng bao gồm các thông tin sản phẩm và trạng thái của món ăn

	3	Admin	Cập nhật trạng thái món ăn
	4	Hệ thống	Thay đổi trạng thái món ăn tương ứng và thông báo
Luồng sự kiện thay thế Luồng sự kiện thay thế	#	<b>Thực hiện bởi</b>	<b>Hành động</b>
	3a	Admin	Nếu món ăn đang được nấu , admin sẽ chọn “Đang nấu” để lưu trạng thái món ăn thành đang nấu
	3b	Admin	Nếu món ăn không thể nấu , admin sẽ chọn “Từ chối” để lưu trạng thái món ăn thành đang nấu
Hậu điều kiện		Không có	

Bảng 2.9 Bảng mô tả chi tiết use case Quản lý trạng thái món ăn

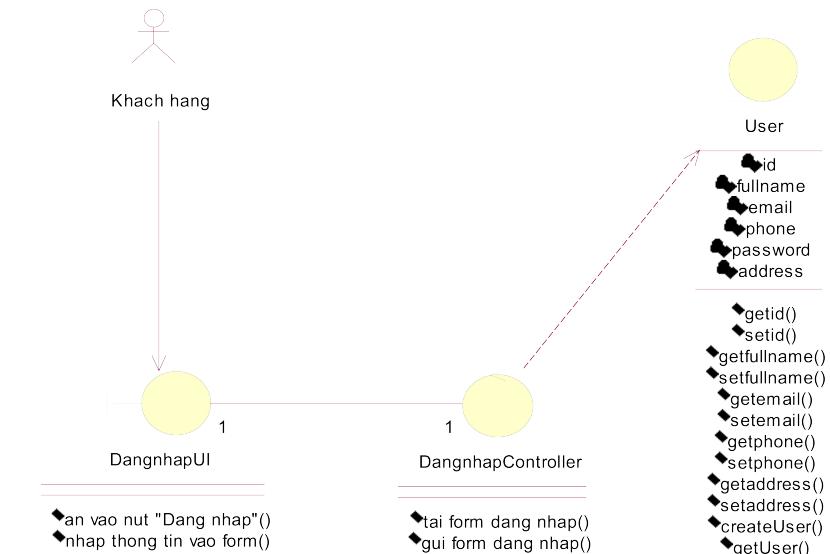


Hình 2.14 Biểu đồ minh họa chi tiết use case quản lý trạng thái món ăn

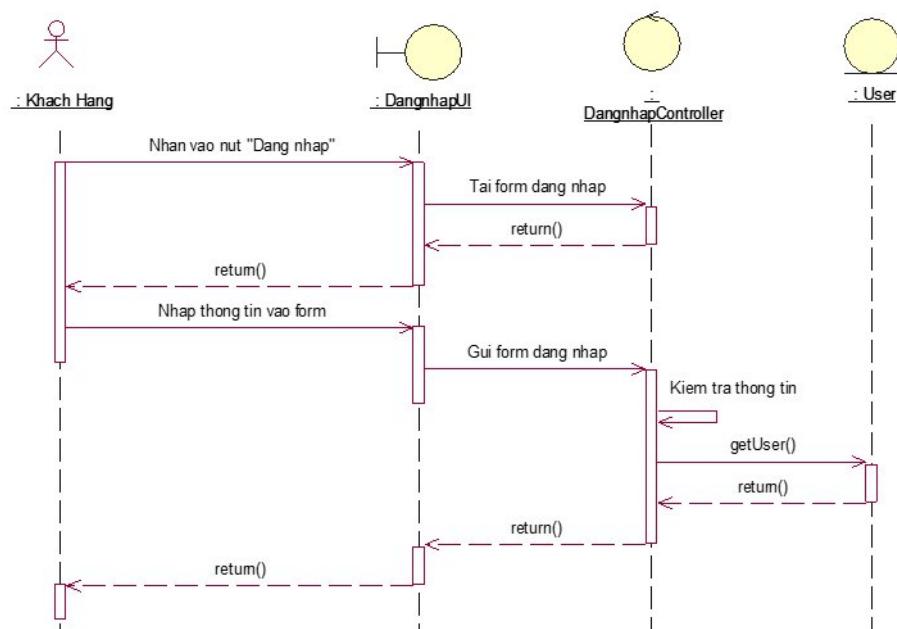
## CHƯƠNG 3 : THIẾT KẾ PHẦN MỀM

### 3.1. Thiết kế chức năng chính

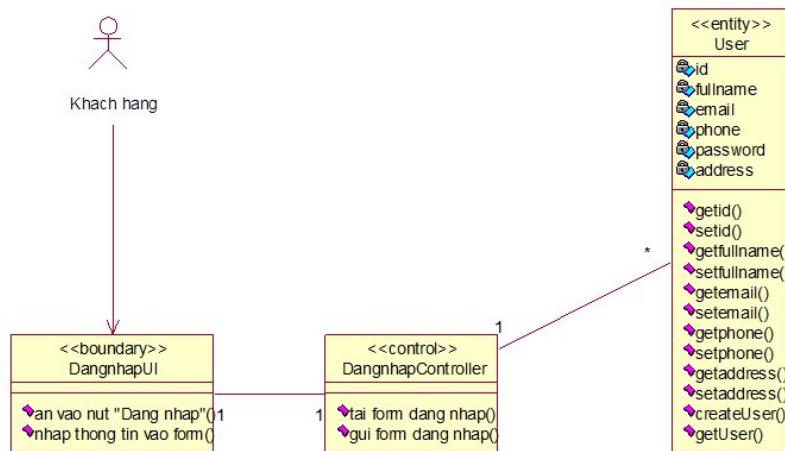
#### 3.1.1. Chức năng đăng nhập



Hình 3.1 Biểu đồ lớp tham gia ca sử dụng chức năng đăng nhập

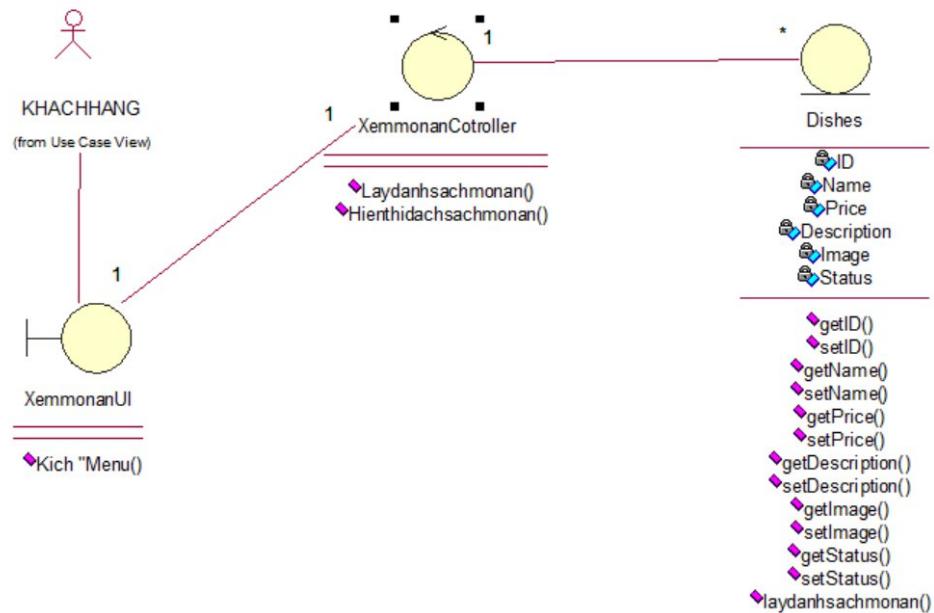


Hình 3.2 Biểu đồ trình tự chức năng đăng nhập

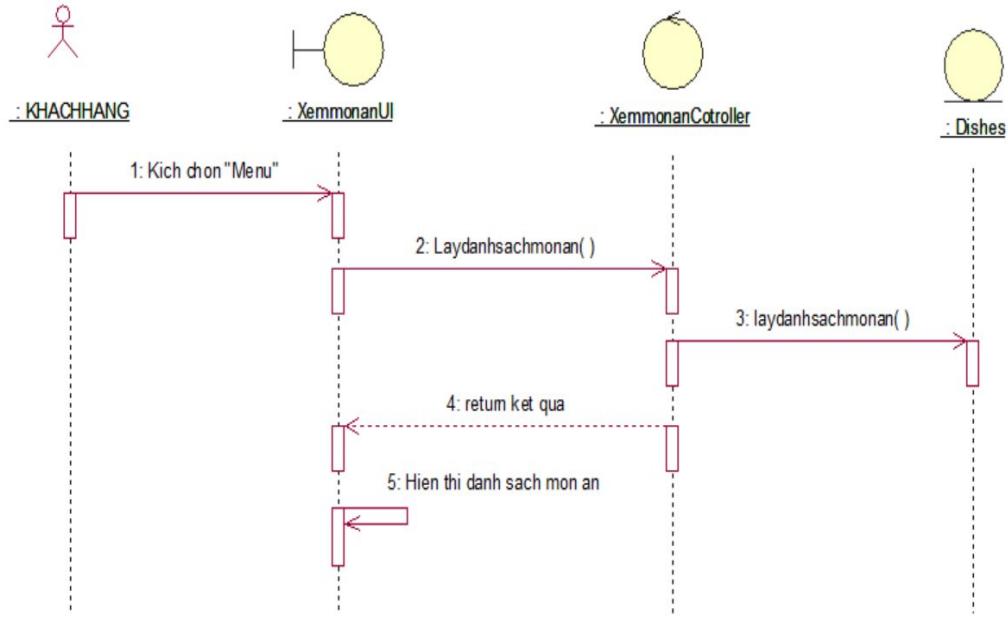


Hình 3.3 Biểu đồ xác định lớp thực thể chức năng đăng nhập

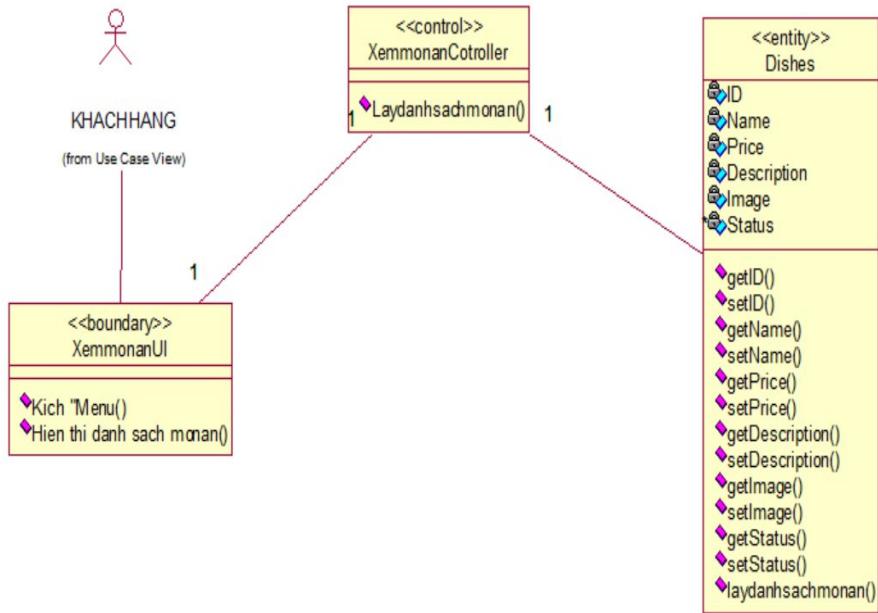
### 3.1.2. Chức năng xem món ăn



Hình 3.4 Biểu đồ lớp tham gia ca sử dụng chức năng xem món ăn

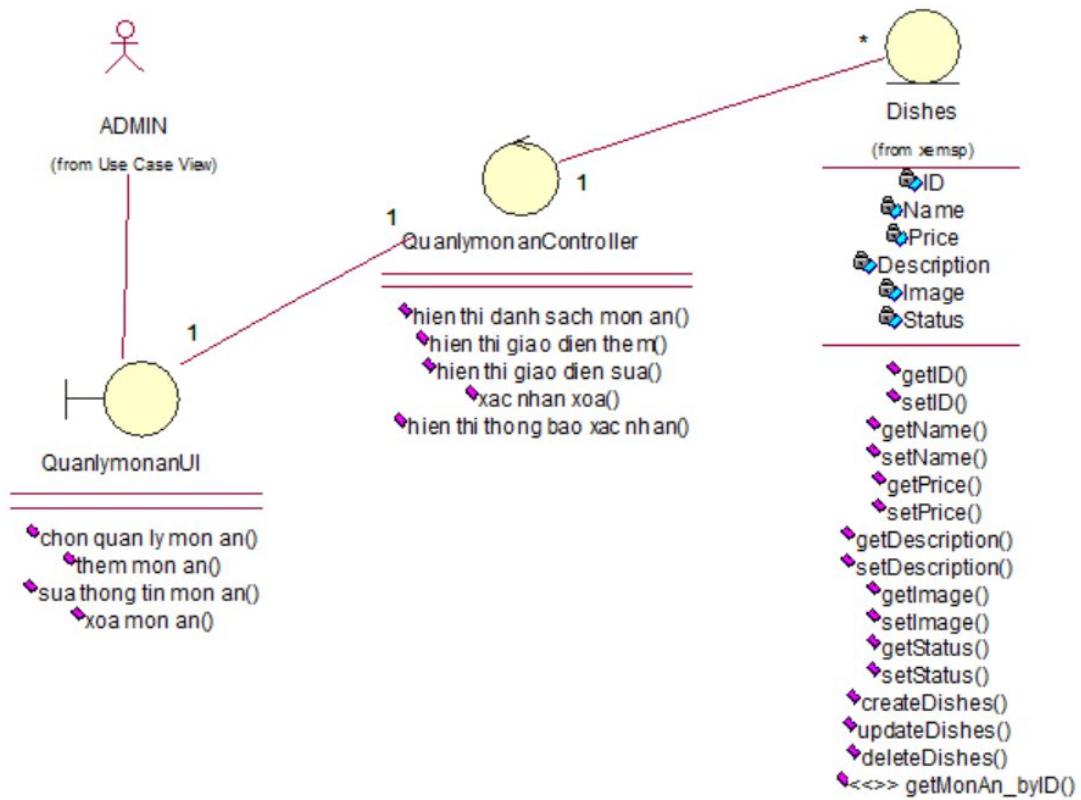


Hình 3.5. Biểu đồ trình tự chức năng Xem món ăn

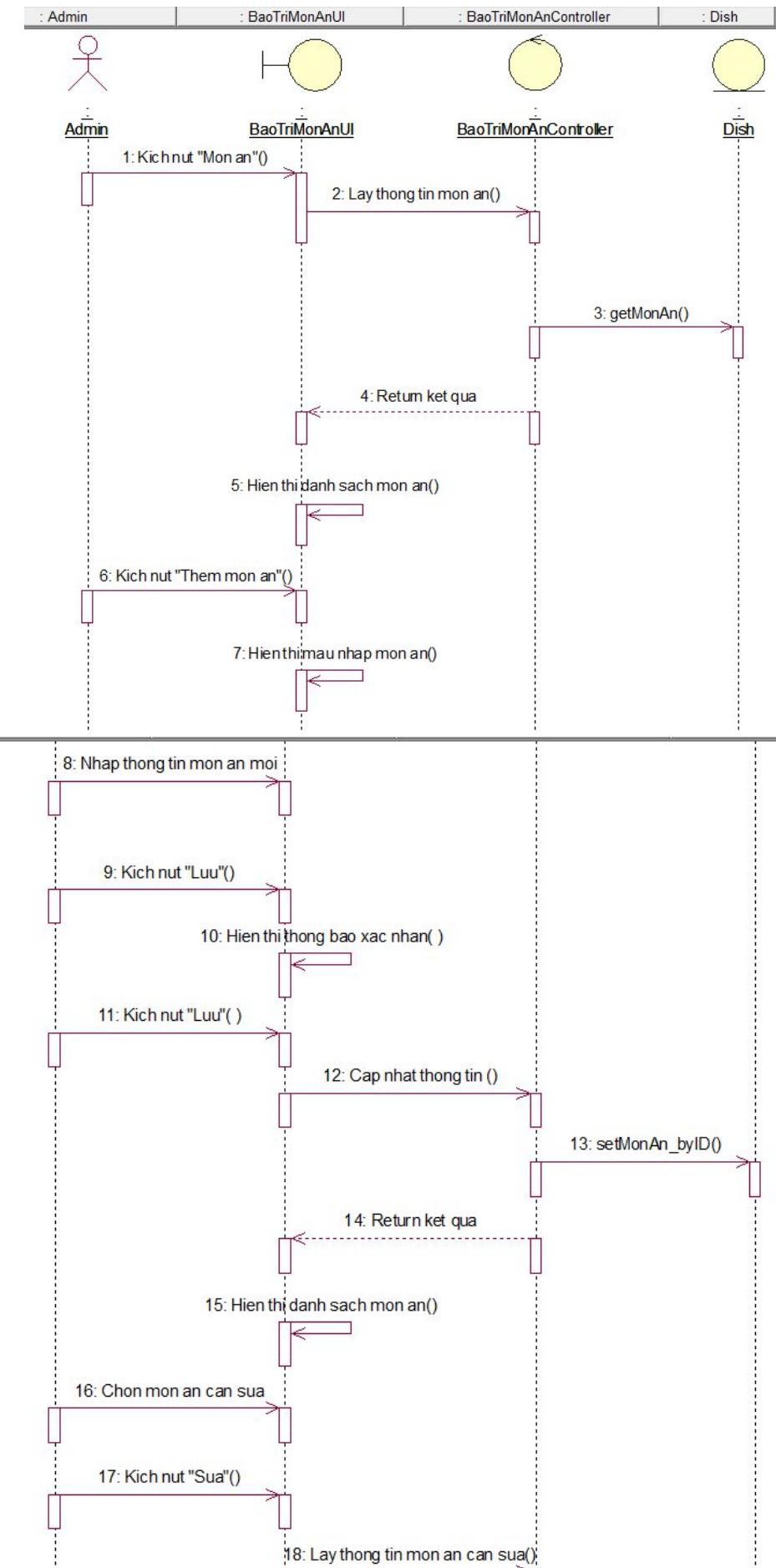


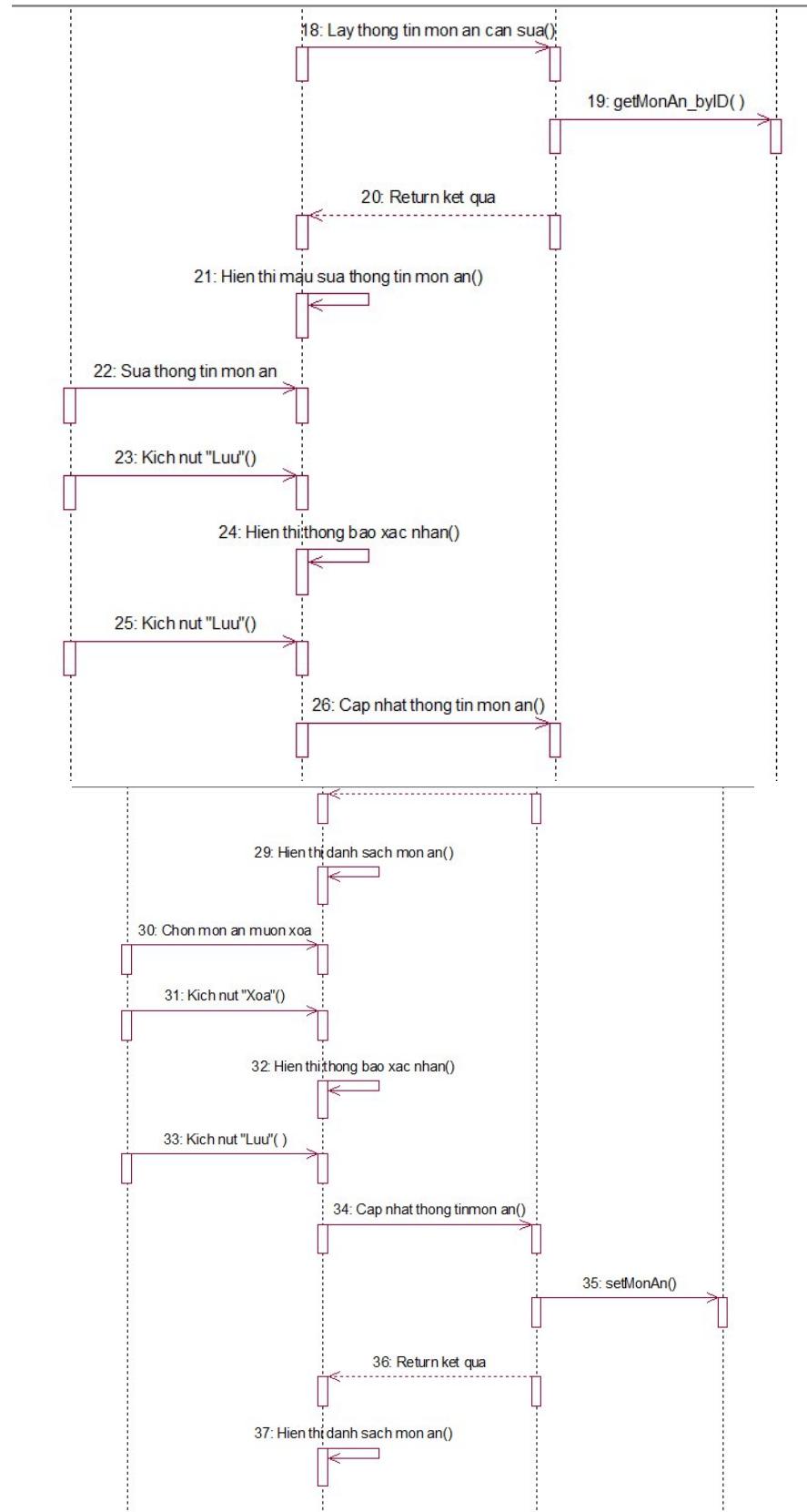
Hình 3.6 Biểu đồ xác định lớp thực thể chức năng Xem món ăn

### 3.1.3. Chức năng quản lý món ăn

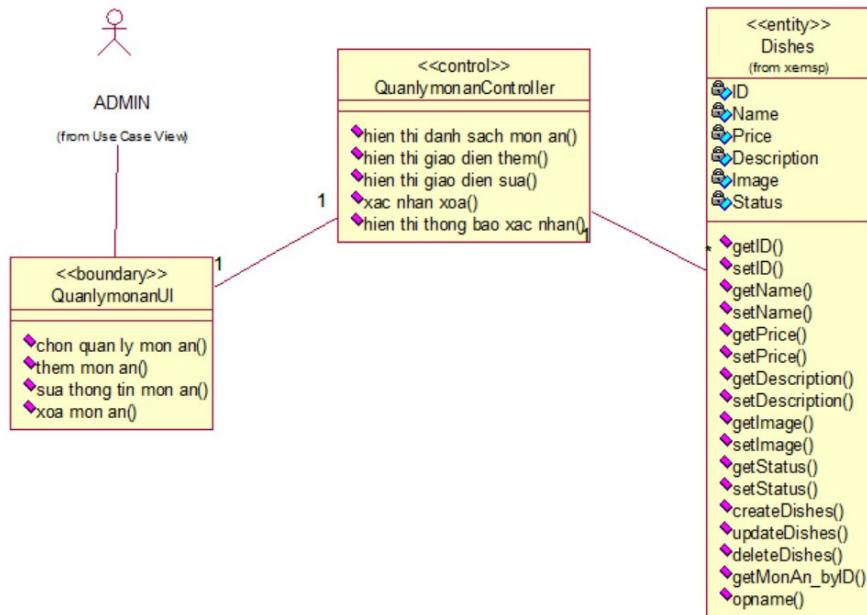


Hình 3.7 Biểu đồ lớp tham gia ca sử dụng chức năng quản lý món ăn



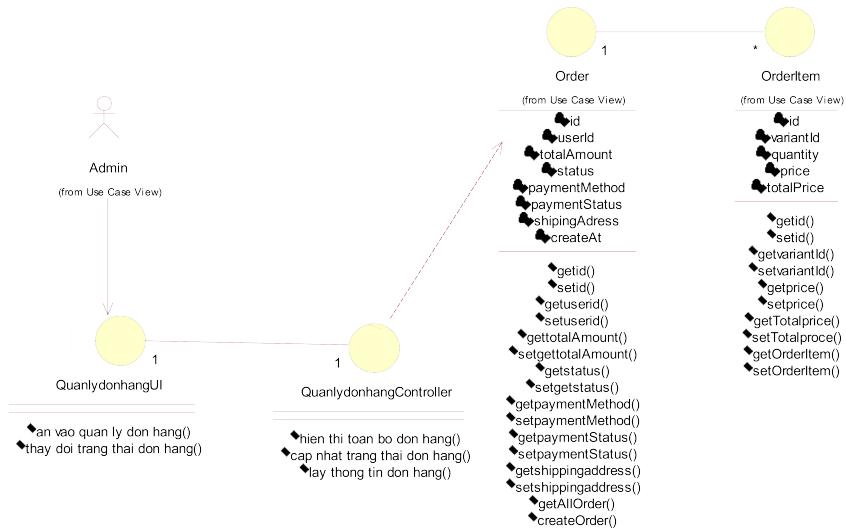


Hình 3.8. Biểu đồ trình tự chức năng quản lý món ăn

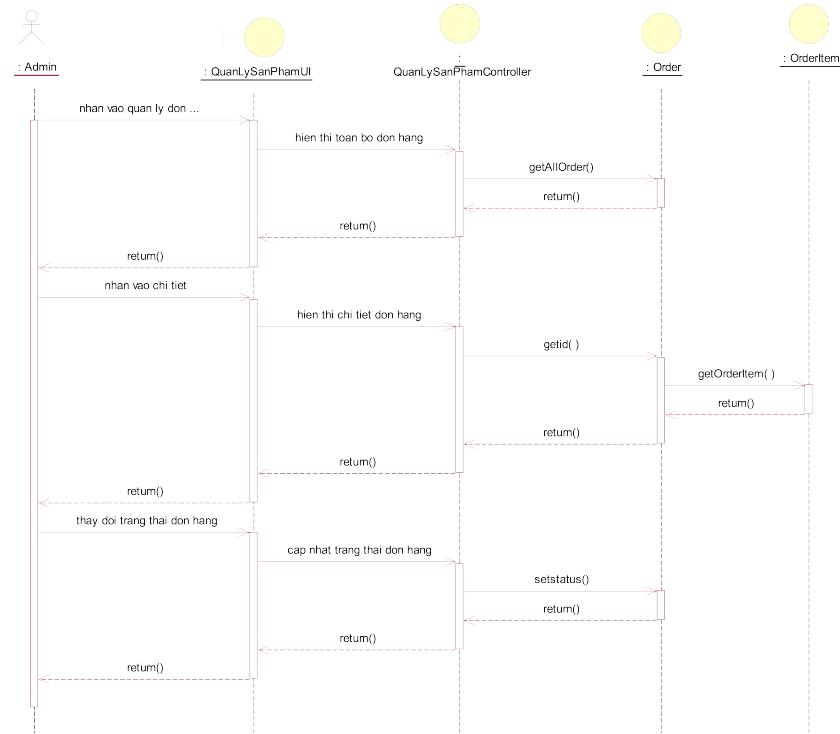


Hình 3.9 Biểu đồ xác định lớp thực thể chức năng quản lý món ăn

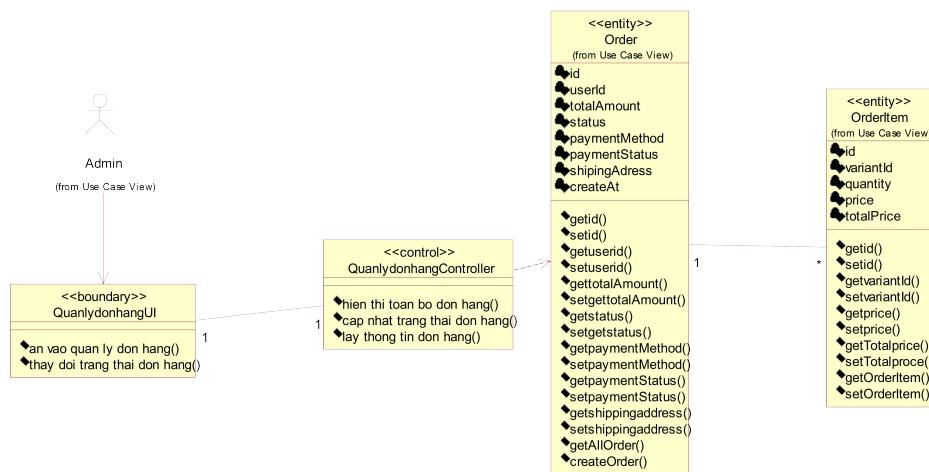
### 3.1.4. Chức năng quản lý đơn hàng



Hình 3.10 Biểu đồ lớp tham gia ca sử dụng chức năng quản lý đơn hàng

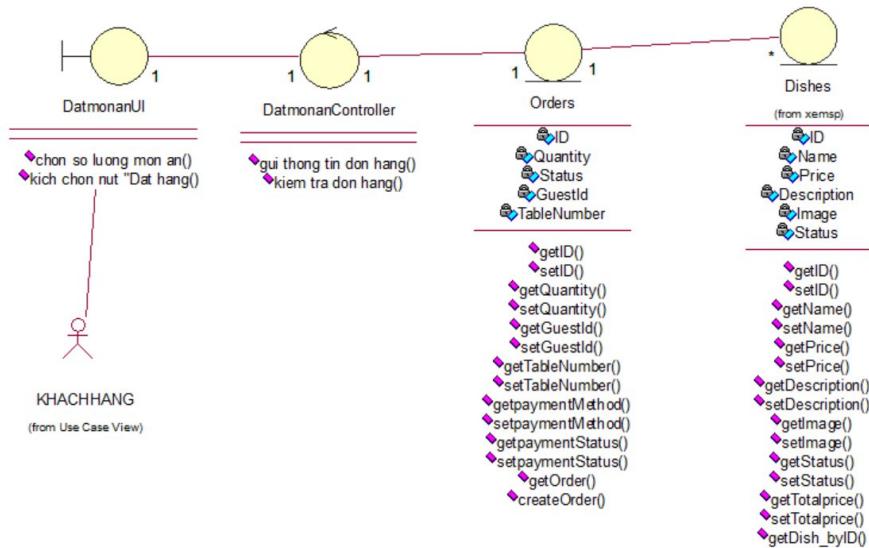


Hình 3.11 Biểu đồ trình tự chức năng quản lý đơn hàng

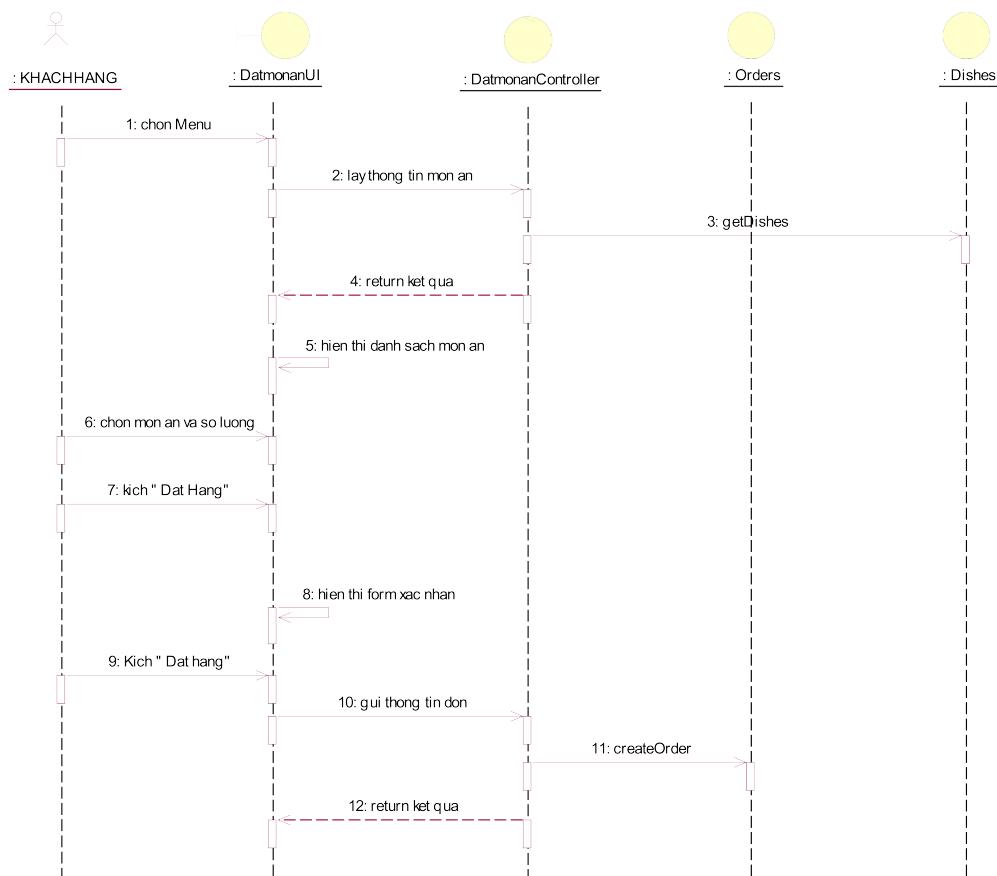


Hình 3.12 Biểu đồ xác định lớp thực thể chức năng quản lý đơn hàng

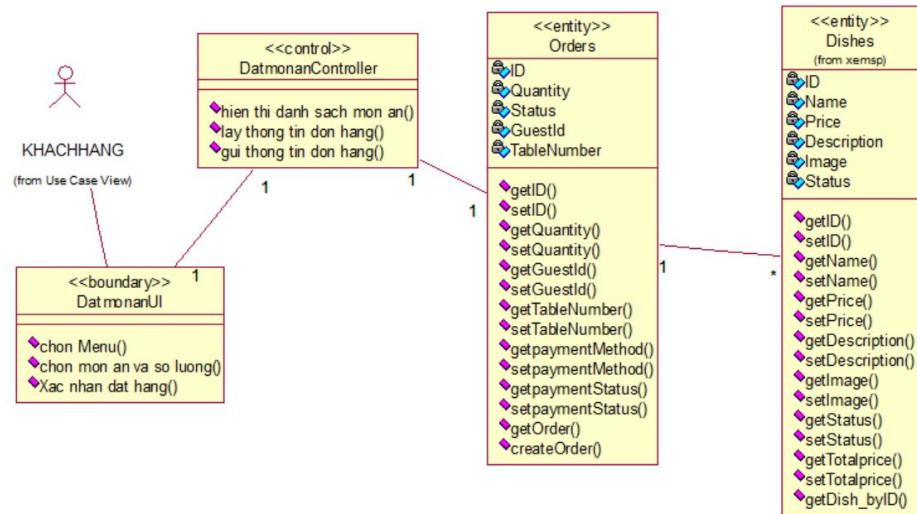
### 3.1.5. Chức năng đặt hàng



Hình 3.13 Biểu đồ lớp tham gia ca sử dụng chức năng đặt hàng



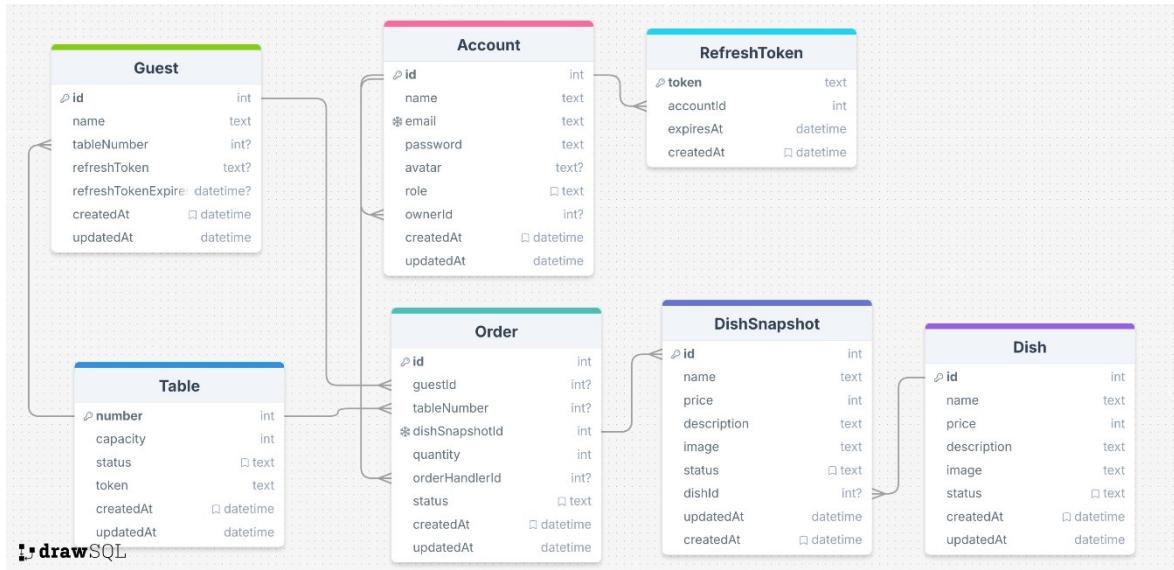
Hình 3.14 Biểu đồ trình tự chức năng đặt hàng



Hình 3.15 Biểu đồ xác định lớp thực thể chức năng đặt hàng

## 3.2. Thiết kế cơ sở dữ liệu

### 3.2.1. Biểu đồ Entity Relationship Diagram



Hình 3.16. Biểu đồ Entity Relationship Diagram

### 3.2.2. Thiết kế các bảng trong cơ sở dữ liệu

Name	Type	Length	Decimals	Not null	Virtual	Key
Id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
Number	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Capacity	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Status	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Token	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
CreatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
UpdatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	

Hình 3.17. Bảng Tables

Name	Type	Length	Decimals	Not null	Virtual	Key
Id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
Name	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Email	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Password	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Avatar	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Role	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
OwnerId	int			<input type="checkbox"/>	<input type="checkbox"/>	
CreatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
UpdatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	

Hình 3.18. Bảng Accounts

Name	Type	Length	Decimals	Not null	Virtual	Key
Id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
Name	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Price	int			<input type="checkbox"/>	<input type="checkbox"/>	
Description	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Image	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Status	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
CreatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
UpdatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	

Hình 3.19. Bảng Dishes

Name	Type	Length	Decimals	Not null	Virtual	Key
<b>Id</b>	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
Name	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Price	int			<input type="checkbox"/>	<input type="checkbox"/>	
Description	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Image	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
Status	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
DishId	int			<input type="checkbox"/>	<input type="checkbox"/>	
CreatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
UpdatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	

*Hình 3.20. Bảng DishSnapshots*

Name	Type	Length	Decimals	Not null	Virtual	Key
<b>Id</b>	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
Name	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
RefreshToken	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
RefreshTokenExpiresAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
TableNumber	int			<input type="checkbox"/>	<input type="checkbox"/>	
CreatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
UpdatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	

*Hình 3.21. Bảng Guests*

Name	Type	Length	Decimals	Not null	Virtual	Key
<b>Id</b>	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
Quantity	int			<input type="checkbox"/>	<input type="checkbox"/>	
Status	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
GuestId	int			<input type="checkbox"/>	<input type="checkbox"/>	
TableNumber	int			<input type="checkbox"/>	<input type="checkbox"/>	
DishSnapshotId	int			<input type="checkbox"/>	<input type="checkbox"/>	
OrderHandlerId	int			<input type="checkbox"/>	<input type="checkbox"/>	
CreatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
UpdatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	

*Hình 3.22. Bảng Orders*

Name	Type	Length	Decimals	Not null	Virtual	Key
Id	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
AccountId	int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Token	longtext			<input type="checkbox"/>	<input type="checkbox"/>	
IsValid	tinyint	1		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
ExpiresAt	datetime	6		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CreatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
UpdatedAt	datetime	6		<input type="checkbox"/>	<input type="checkbox"/>	
JwtTokenId	longtext			<input type="checkbox"/>	<input type="checkbox"/>	

Hình 3.23. Bảng RefreshTokens

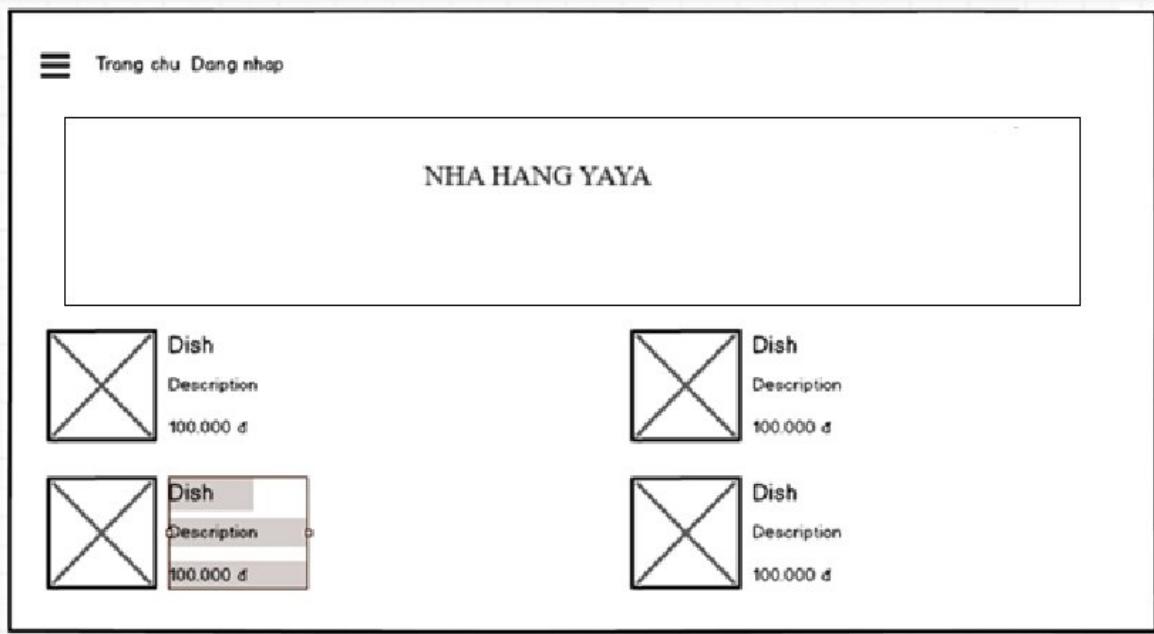
### 3.3. Thiết kế giao diện

Giao diện mockup Đăng nhập:

The mockup shows a rectangular form with a title 'Đăng nhập' at the top. Below it is a paragraph of text containing a red 'text' placeholder and a blue 'unassigned link'. Underneath is another row of text with a blue 'web link'. The form includes two input fields: one labeled 'Email' and another labeled 'Password', both represented by long rectangular boxes. At the bottom is a large rectangular button labeled 'Đăng nhập'.

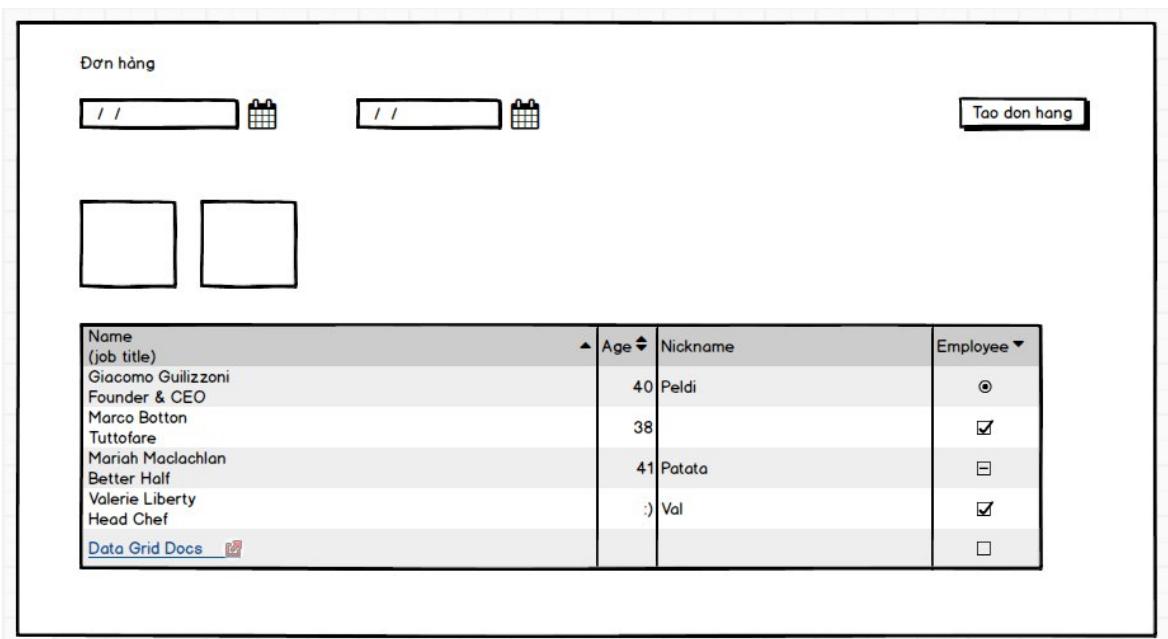
Hình 3.24. Giao diện mockup Đăng nhập

Giao diện mockup Trang chủ:

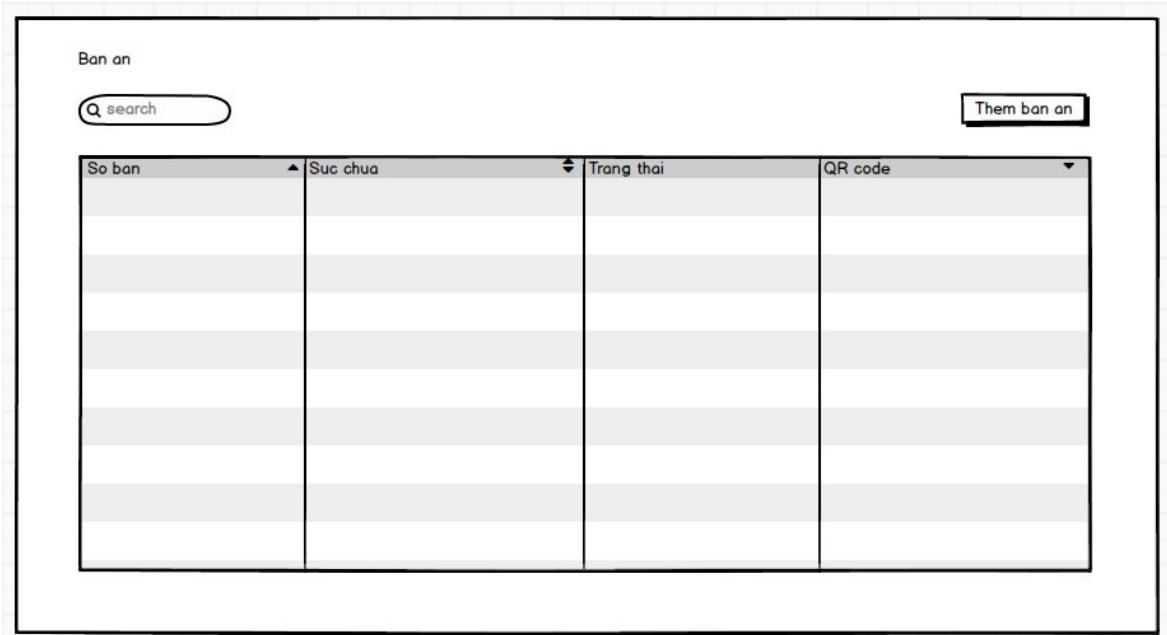


Hình 3.25. Giao diện mockup Trang chủ

Giao diện mockup Quản lý đơn hàng:

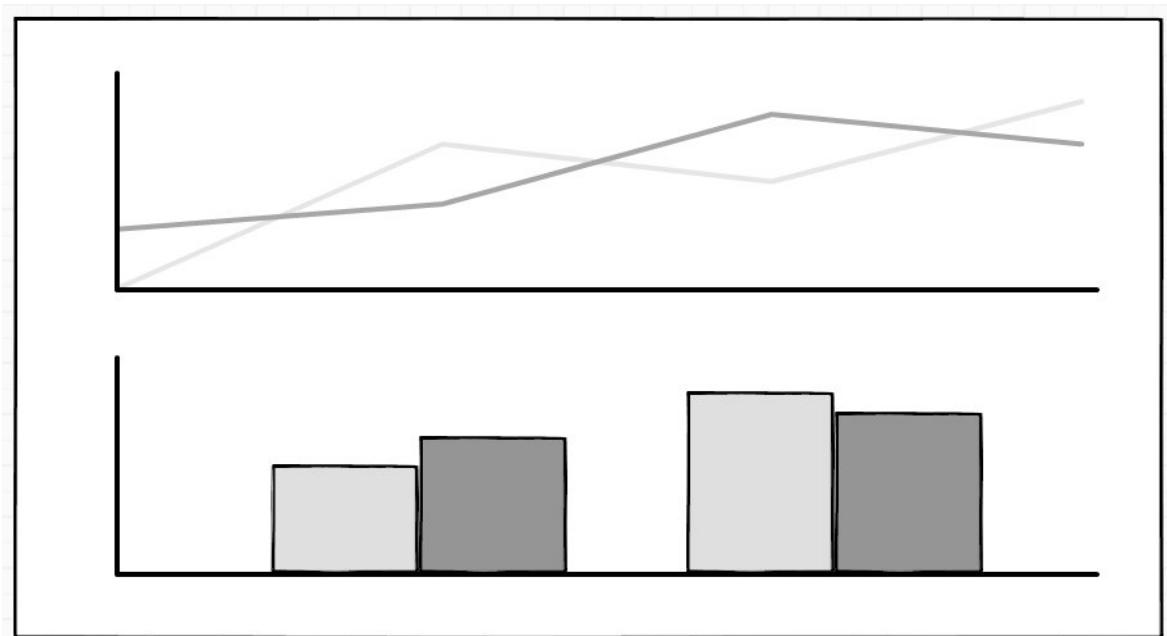


Hình 3.26. Giao diện mockup Quản lý đơn hàng



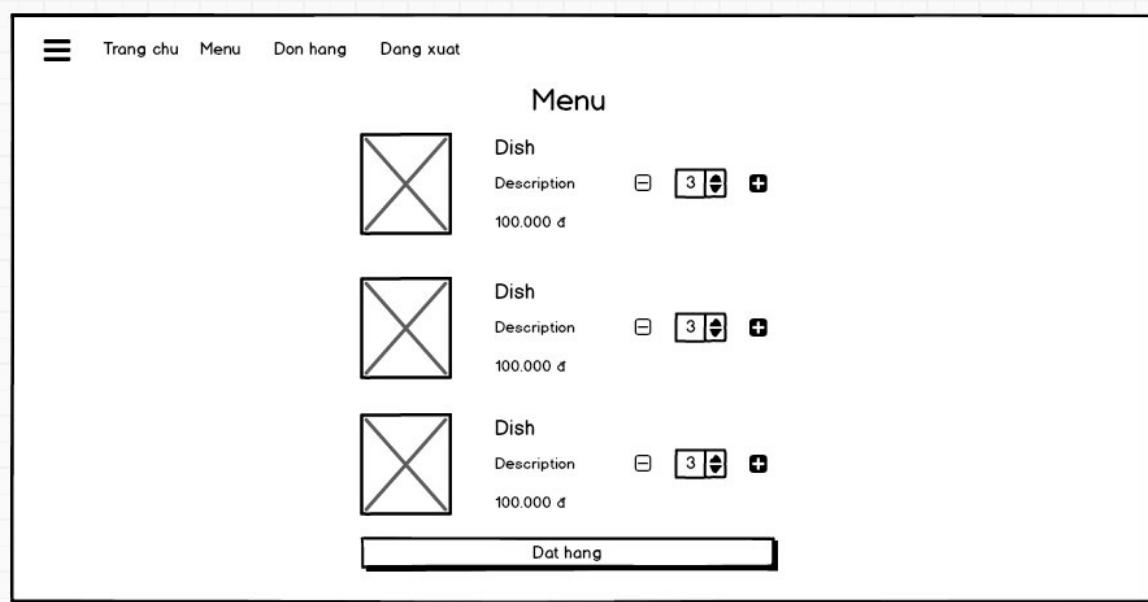
Hình 3.27. Giao diện mockup Quản lý bàn ăn

Giao diện mockup Thông kê:



Hình 3.28. Giao diện mockup Thông kê

Giao diện mockup Menu:



Hình 3.29. Giao diện mockup Menu

## CHƯƠNG 4: KẾT QUẢ XÂY DỰNG

### 4.1. Cài đặt

#### 4.1.1. Frontend

- Được xây dựng bằng **Next.js**, một framework React.js mạnh mẽ cho phát triển giao diện người dùng:
  - **SSR (Server-Side Rendering)**: Tối ưu hiển thị giao diện nhằm tăng tốc độ tải trang.
  - **Static Site Generation (SSG)**: Dùng cho những trang tính như thực đơn nhà hàng.
  - **Client-Side Rendering (CSR)**: Hỗ trợ tương tác nhanh khi thao tác người dùng.
- Frontend giao tiếp với Backend qua các API REST.
- Thiết kế giao diện thân thiện, hỗ trợ responsive trên nhiều thiết bị (PC, tablet, mobile).

#### 4.1.2. Backend (Xử lý nghiệp vụ)

Backend được phát triển bằng **.NET Core** nhờ vào sự mạnh mẽ về tính module hóa, độ bền và hiệu năng cao:

- **Xử lý logic nghiệp vụ**: Quản lý danh mục món ăn, đặt món, thanh toán.
- **API RESTful**: Cung cấp các endpoint để Frontend gửi request.
- **Bảo mật**: Triển khai Bearer token cho xác thực người dùng.
- **Middleware**: Xử lý request/response.

#### 4.1.3. Cơ sở dữ liệu

Sử dụng MySQL làm cơ sở dữ liệu chính:

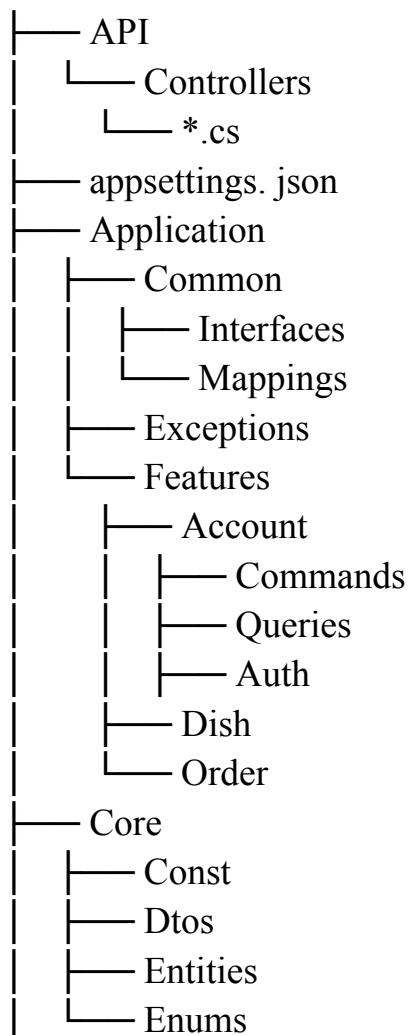
- Quản lý dữ liệu về thực đơn, đơn hàng, khách hàng.

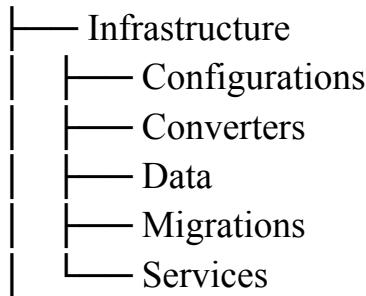
- Tính quan hệ logic giữa các bảng như: Khách hàng, Đơn hàng, Món ăn, ...
- Tối ưu hiệu năng truy vấn bằng các công nghệ như caching và indexing.

#### 4.1.4. Giao tiếp giữa các thành phần

- **Frontend - Backend:**
  - Sử dụng các API RESTful để truyền request.
  - Dữ liệu request/response được định dạng theo JSON.
- **Backend - Database:**
  - Backend tương tác với MySQL qua Entity Framework Core, cho phép xử lý ORM một cách dễ dàng.

#### 4.1.5. Triển khai Clean Architecture





## Chi tiết các thư mục

- **API:** Chứa các controller xử lý yêu cầu HTTP từ người dùng.
  - Controllers/: Các lớp API Controller (ví dụ: OrderController, DishController).
  - **appsettings.json:** File cấu hình cho ứng dụng, chứa thông tin như connection string, key bảo mật.
- **Application:** Tầng logic nghiệp vụ chính của ứng dụng.
  - Common/Interfaces: Các interface để định nghĩa hợp đồng giữa các tầng.
  - Common/Mappings: Các ánh xạ DTO hoặc Entity.
  - Exceptions/: Các lớp xử lý ngoại lệ.
  - Features/: Tách biệt theo tính năng (Account, Dish, Order) với các lớp con như Commands, Queries, Auth.
- **Core:** Tầng chứa các định nghĩa cốt lõi không phụ thuộc vào framework.
  - Const/: Các hằng số sử dụng chung trong ứng dụng.
  - Dtos/: Các đối tượng truyền dữ liệu.
  - Entities/: Các thực thể cốt lõi như Dish, Order.
  - Enums/: Các enum định nghĩa trạng thái hoặc kiểu dữ liệu.
- **Infrastructure:** Tầng triển khai các dịch vụ phụ thuộc framework.
  - Configurations/: Cấu hình cho database, email, hoặc logging.
  - Converters/: Các bộ chuyển đổi dữ liệu hoặc định dạng.
  - Data/: Các lớp truy cập dữ liệu, repository.

- `Migrations/`: Các file migration cho cơ sở dữ liệu.
- `Services/`: Các dịch vụ như gửi email, ...

## 4.2. Giao diện

### 1. Giao diện phía quản trị

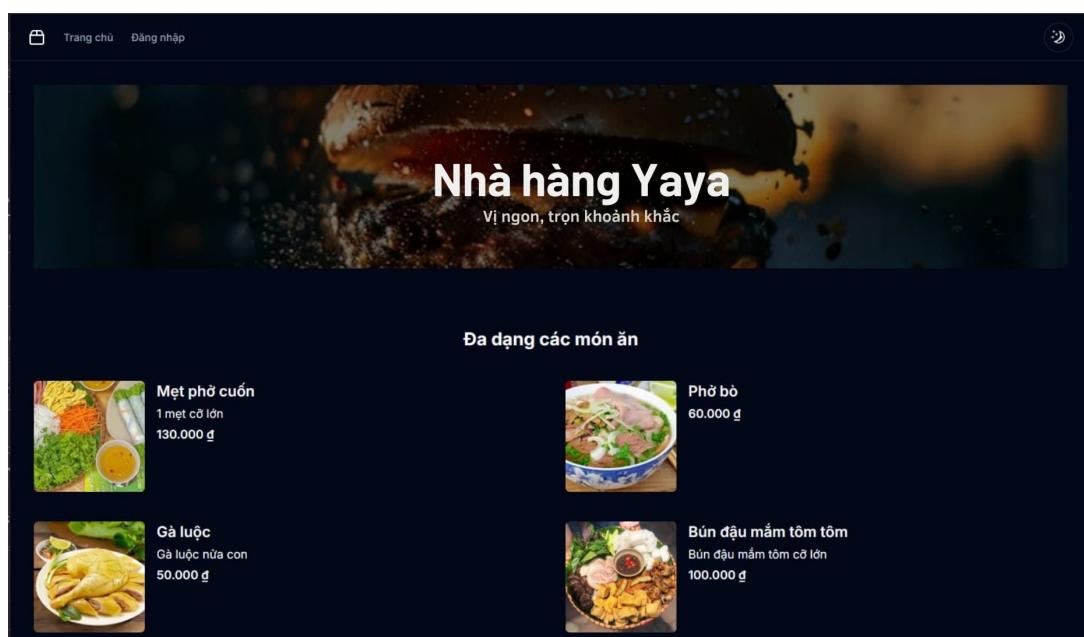
Giao diện đăng nhập:

The screenshot shows a login page with the following elements:

- Header:** Trang chủ | Đăng nhập | A sun icon.
- Title:** Đăng nhập
- Text:** Nhập email và mật khẩu của bạn để đăng nhập vào hệ thống
- Email Input:** m@example.com
- Password Input:** (empty)
- Login Button:** Đăng nhập
- Google Login:** Đăng nhập bằng Google

Hình 4.1. Giao diện đăng nhập

Màn hình trang chủ:



Hình 4.2. Màn hình trang chủ

### Màn hình quản lý đơn hàng:

**Đơn hàng**  
Quản lý đơn hàng

Từ 12/26/2024 12:00 AM Đến 12/27/2024 11:59 PM Reset + Tạo đơn hàng

Tên khách Số bàn Trạng thái

Bàn	Khách hàng	Món ăn	Trạng thái	Người xử lý	Tạo/Cập nhật
4	Phạm Văn Hà(#99)	Cơm rang dưa bò x1 50.000đ	Đã thanh toán	admin	09:51:33 26/12/2024 09:52:05 26/12/2024
4	Phạm Văn Hà(#99)	Gà luộc x3 150.000đ	Đã thanh toán	admin	09:51:33 26/12/2024 09:52:05 26/12/2024
4	Phạm Văn Hà(#99)	Bún chả Hà Nội x1 50.000đ	Đã thanh toán	admin	09:51:33 26/12/2024 09:52:05 26/12/2024

Chờ xử lý: 0 Đang nấu: 2 Từ chối: 0 Đã phục vụ: 0 Đã thanh toán: 10

Hình 4.3. Màn hình quản lý đơn hàng

Màn hình chi tiết quản lý đơn hàng tại một bàn:

**Đơn hàng**  
Quản lý đơn hàng

Từ 12/26/2024 12:00 AM Đến 12/27/2024 11:59 PM Reset + Tạo đơn hàng

Tên khách Số bàn

Bàn	Khách hàng	Món ăn	Trạng thái	Người xử lý	Tạo/Cập nhật
4	Phạm Văn Hà(#99)	Cơm rang dưa bò x1 50.000đ	Đã thanh toán	admin	09:51:33 26/12/2024 09:52:05 26/12/2024
4	Phạm Văn Hà(#99)	Gà luộc x3 150.000đ	Đã thanh toán	admin	09:51:33 26/12/2024 09:52:05 26/12/2024

**Khách đang ngồi tại bàn 2**

Tên: Rosa (#100) | Bàn: 2  
Ngày đăng ký: 09:50:28 26/12/2024

Đơn hàng:

- 1 ⚡ Cơm rang dưa bò x1 50.000đ 09:50:47 26/12/2024
- 2 ⚡ Bún chả Hà Nội x1 50.000đ 09:50:47 26/12/2024

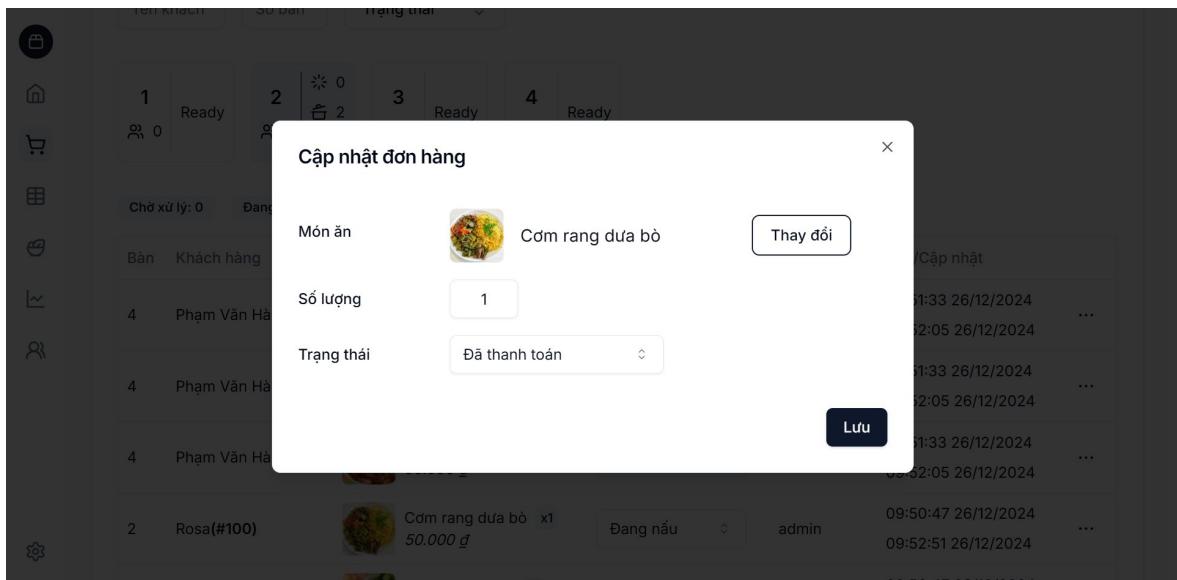
Chưa thanh toán: **100.000đ**

Đã thanh toán: **0đ**

Thanh toán tất cả (2 đơn) Từ chối tất cả (2 đơn)

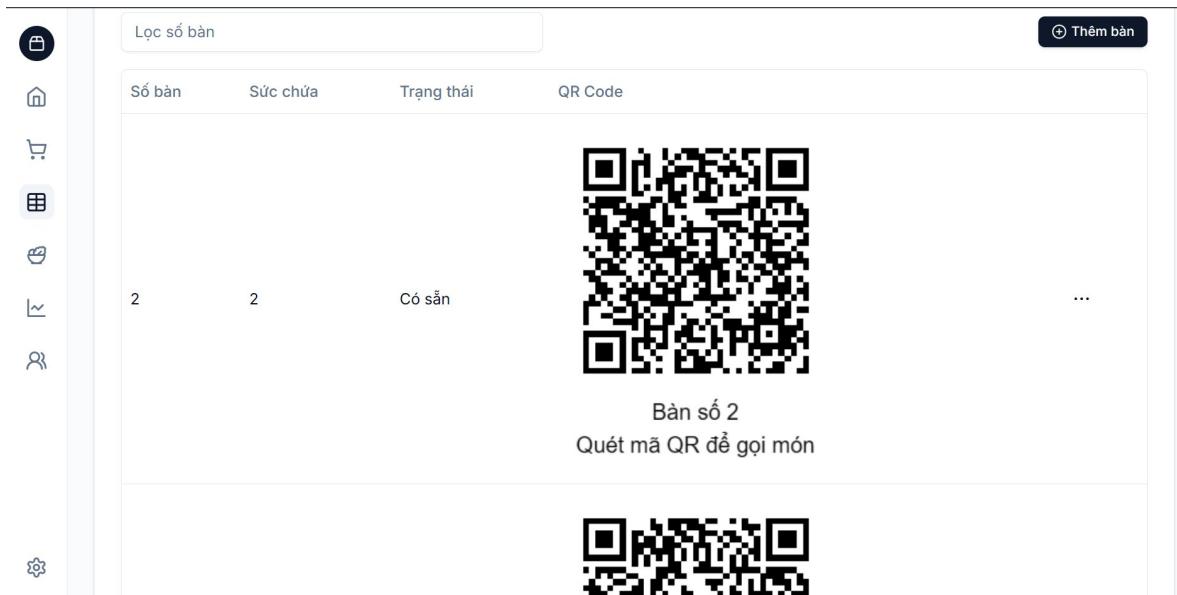
Hình 4.4. Màn hình chi tiết quản lý đơn hàng tại một bàn

Màn hình chi tiết cập nhật đơn hàng



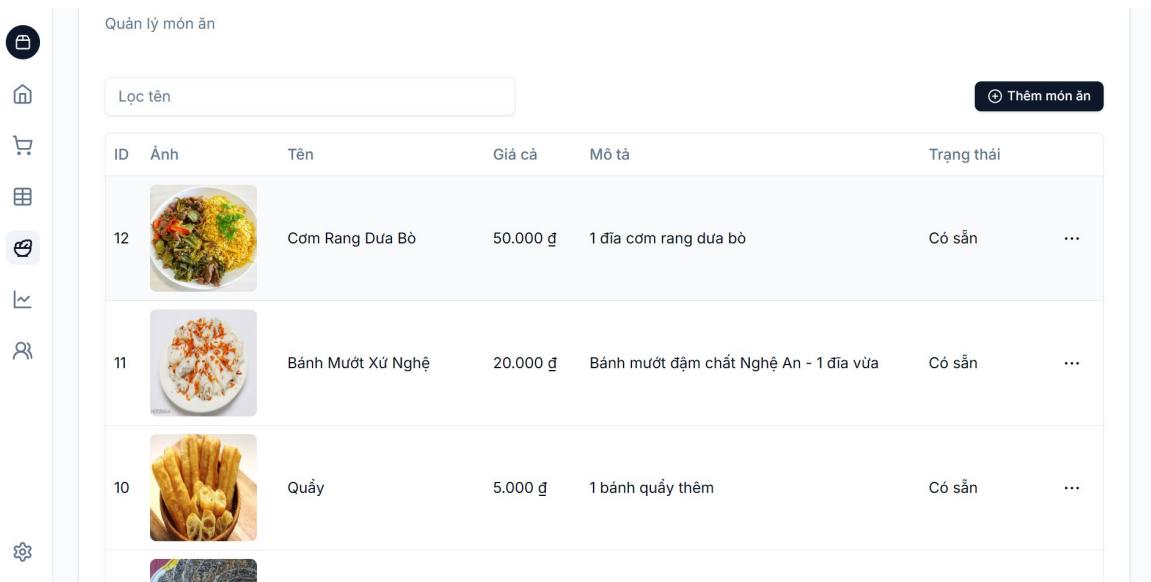
Hình 4.5. Màn hình chi tiết cập nhật đơn hàng

#### Màn hình quản lý bàn ăn:



Hình 4.6. Màn hình quản lý bàn ăn

#### Màn hình quản lý món ăn:

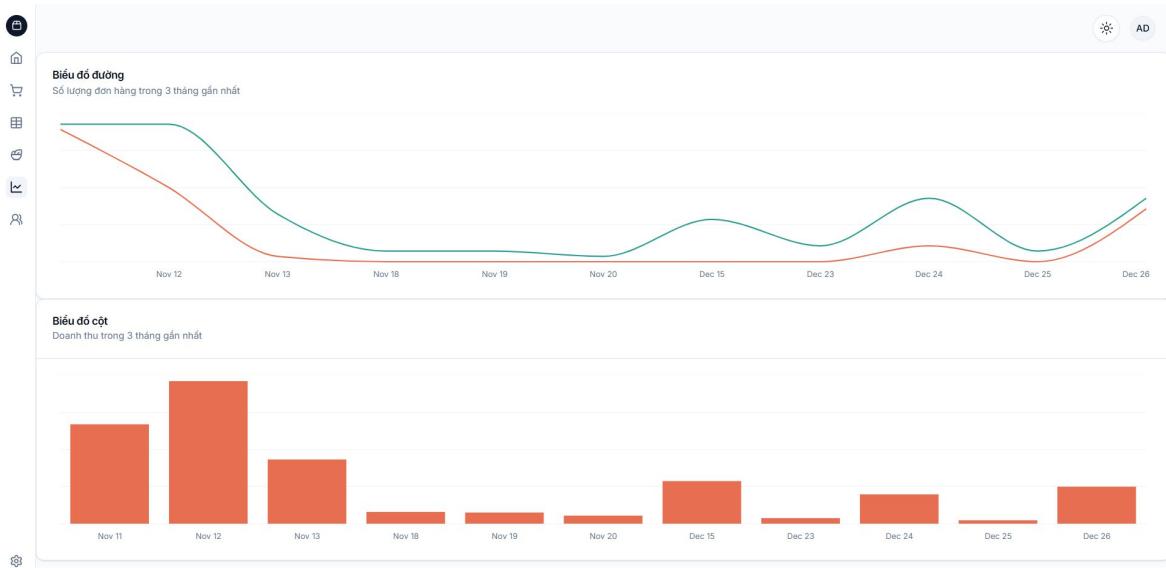


The screenshot shows a software interface for managing a restaurant's menu. On the left is a vertical sidebar with icons for home, categories, products, reports, users, and settings. The main area has a title "Quản lý món ăn" (Manage Menu Items). It includes a search bar labeled "Lọc tên" (Filter by name) and a button "Thêm món ăn" (Add new dish). A table lists three menu items:

ID	Ảnh	Tên	Giá cả	Mô tả	Trạng thái
12		Cơm Rang Dưa Bò	50.000 ₫	1 đĩa cơm rang dưa bò	Có sẵn
11		Bánh Mướt Xứ Nghệ	20.000 ₫	Bánh mướt đậm chất Nghệ An - 1 đĩa vừa	Có sẵn
10		Quẩy	5.000 ₫	1 bánh quẩy thêm	Có sẵn

Hình 4.7. Màn hình quản lý món ăn

Màn hình thống kê doanh thu:



Hình 4.8. Màn hình thống kê doanh thu

Màn hình quản lý nhân viên:

Tài khoản  
Quản lý tài khoản nhân viên

Filter emails... Tạo tài khoản

ID	Avatar	Tên	Email
4		Nhân Viên 2	nhanvien2@gmail.com
3		Nhân Viên 1	nhanvien@gmail.com
2		Nhân Viên 1	emp@gmail.com

Hiển thị 3 trong 3 kết quả Previous 1 Next >

Hình 4.9. Màn hình quản lý nhân viên

## 2. Giao diện phía khách hàng

Màn hình chi tiết menu món ăn:

Trang chủ Menu Đơn hàng Đăng xuất

	Lòng lợn Mết lợn lợn lớn 100.000 ₫	- 0 +
	Bún chả Hà Nội Bún chả Hà Nội nhiều chả 50.000 ₫	- 0 +
	Phở bò Phở bò tái 60.000 ₫	- 0 +
	Gà luộc Gà luộc nửa con 50.000 ₫	- 0 +
	Bún đậu mắm tôm tôm Bún đậu mắm tôm cỡ lớn 100.000 ₫	- 0 +

Đặt hàng - 0 món 0 ₫

Hình 4.10. Màn hình chi tiết menu món ăn

Màn hình đơn đặt hàng:

Trang chủ Menu Đơn hàng Đăng xuất

Đơn hàng

1		Lòng lợn 100.000 ₫ x 1	Chờ xử lý
2		Bún chả Hà Nội 50.000 ₫ x 1	Chờ xử lý

Đơn chưa thanh toán · 2 món 150.000 ₫

Hình 4.11. Màn hình đơn đặt hàng

## CHƯƠNG 5: KIỂM THỬ PHẦN MỀM

### 5.1. Kế hoạch kiểm thử

#### 5.1.1. Giới thiệu

##### 1. Mục đích

Tài liệu kế hoạch kiểm thử này đưa ra các mục đích sau:

- Xác định thông tin cơ bản về dự án và các thành phần chức năng được kiểm thử và không được kiểm thử.
- Liệt kê những yêu cầu cho việc kiểm thử (Test Requirements).
- Những chiến lược kiểm thử nên được sử dụng.
- Ước lượng những yêu cầu về tài nguyên và chi phí cho việc kiểm thử.
- Những tài liệu được lập sau khi hoàn thành việc kiểm thử.

##### 2. Các định nghĩa và các từ viết tắt.

Các định nghĩa:

Thuật ngữ	Định nghĩa
http://localhost:3000/	Địa chỉ của trang web
URL	Cách gọi chung địa chỉ trang Web
Test Designer/ Tester	Người thiết kế kiểm thử/ Kiểm thử viên

Bảng 5.1. Bảng định nghĩa và các từ viết tắt

Các từ viết tắt:

Từ viết tắt	Mô tả
UC	Use case
TC	Test case
GUI	Giao diện người dùng
ST	System test
UAT	User acceptance test
TR	Test report

### 3. Lịch trình công việc

Môc công việc	Sản phẩm	Bắt đầu	Kết thúc
Lập kế hoạch kiểm thử	Test plan	25/5/2025	25/5/2025
Xem lại các tài liệu	Test plan	25/5/2025	25/5/2025
Thiết kế các test case	Test case	25/5/2025	25/5/2025
Viết các test case	Test case	25/5/2025	25/5/2025
Xem lại các testcase	Test case	25/5/2025	25/5/2025
Thực thi các testcase	Test case	25/5/2025	25/5/2025
Ghi nhận và đánh giá kết quả kiểm thử	Test report	27/5/2025	28/5/2025

Bảng 5.2. Bảng lịch trình công việc

### 4. Phạm vi kiểm thử

Các chức năng được kiểm thử:

- Đăng nhập: Kiểm tra chức năng đăng nhập vào hệ thống với tài khoản vừa tạo.
- Đổi mật khẩu: Kiểm tra chức năng thay đổi mật khẩu của tài khoản đã có.
- Thông báo realtime cho người dùng về trạng thái đơn đặt hàng.

### 5. Tài nguyên kiểm thử

#### a) Phần cứng

Máy tính cá nhân có kết nối mạng Internet.

CPU	RAM	SSD	Architecture
Intel Core i5, 2.3 GHz	16 GB	512 GB	64 bits

Bảng 5.3. Bảng cấu hình phần cứng

#### b) Phần mềm.

Tên phần mềm	Phiên bản	Loại
Google Chrome	19.0.1084.56	Trình duyệt web
Microsoft Windows 11	10	Hệ điều hành

Bảng 5.4. Bảng thông tin phần mềm

c) Công cụ kiểm thử.

Hoạt động	Công cụ	Nhà cung cấp	Phiên bản
Quản lý Test Case	Microsoft Office Excel	Microsoft	2021
Quản lý phiên bản	Microsoft Office Word	Microsoft	2021
Theo dõi lỗi	Microsoft Office Excel	Microsoft	2021

Bảng 5.5. Bảng công cụ kiểm thử

d) Nhân sự

Thành viên	Vai trò
Nguyễn Văn Chung	Test Manager / Test Designer / Tester

Bảng 5.6. Bảng nhân sự

6. Chiến lược kiểm thử

a) Các giai đoạn kiểm thử.

Kiểm thử ở mức hệ thống (ST) và kiểm thử chấp nhận (UAT):

- Dùng kiểu kiểm thử tự động để kiểm thử từng chức năng.
- Việc kiểm thử chỉ bắt đầu khi đã hoàn thiện bộ test case để kiểm thử chức năng.
- Thiết kế test case theo phương pháp phân vùng tương đương.
- Chỉ thực hiện kiểm thử hồi quy, không thực hiện kiểm thử lại.

b) Loại kiểm thử

Kiểm thử hộp đen

Mục đích kiểm tra	Đảm bảo rằng tất cả các chức năng được kiểm tra hoạt động đúng theo đặc tả yêu cầu, không phụ thuộc vào cách thức mà phần mềm được triển khai.
Kỹ thuật	Thực thi tất cả các trường hợp có thể có cho mỗi nhóm chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để xác định: <ul style="list-style-type: none"> <li>- Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng.</li> </ul>

	<ul style="list-style-type: none"> <li>- Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng.</li> </ul>
Tiêu chuẩn dùng	<ul style="list-style-type: none"> <li>- Tất cả các test case đã được thiết kế đều được thực thi.</li> <li>- Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho lập trình viên khắc phục.</li> </ul>
Chịu trách nhiệm kiểm thử	Test Designer / Tester
Cách kiểm thử	Kiểm thử tự động bằng công cụ Selenium, tuần tự theo các bước được định nghĩa trong test case
Xử lý ngoại lệ	Liệt kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử.

Bảng 5.7. Bảng loại kiểm thử hộp đen

#### 7. Điều kiện chấp nhận kiểm thử.

- Tỉ lệ test case đạt(passed): 100%
- Tỉ lệ test case không đạt (failed): 0%
- Hệ thống vẫn chạy ổn định trên các trình duyệt web khác nhau (Google Chrome, Microsoft Edge)

#### 8. Phân loại lỗi

Mức độ nghiêm trọng	Đặc tả lỗi
High	<ul style="list-style-type: none"> <li>- Không thể đăng nhập</li> <li>- Không đặt khám được kể cả bác sĩ và phòng khám.</li> <li>- Không đổi mật khẩu được.</li> </ul>
Medium	<ul style="list-style-type: none"> <li>- Đăng nhập sai không báo lỗi, không trả về kết quả</li> <li>- Thay đổi thông tin hoặc thêm hồ sơ bệnh nhân không</li> </ul>

	được.
Low	<ul style="list-style-type: none"> <li>- Tìm kiếm trung tâm tiêm chủng sai.</li> <li>- Tìm kiếm bài viết sai.</li> </ul>

Bảng 5.8. Bảng loại phân loại lỗi

### 5.1.2. Thực hiện kiểm thử

#### 1. Kiểm thử chức năng đăng nhập

a) Giao diện chức năng:

**Đăng nhập**  
Nhập email và mật khẩu của bạn để đăng nhập vào hệ thống

Email  
m@example.com

Password

Đăng nhập

Đăng nhập bằng Google

Hình 5.1. Giao diện Đăng nhập.

b) Mô tả giao diện và các điều kiện ràng buộc:

- Tiến hành nhập email và mật khẩu rồi nhấn nút “Đăng nhập”.
- Điều kiện ràng buộc: Không được để trống email và mật khẩu.

Sử dụng phương pháp phân hoạch tương đương thiết lập các ca kiểm thử:

Điều kiện đầu vào	Vùng hợp lệ	Vùng không hợp lệ
Email	Email hợp lệ	Email không hợp lệ

	VD: email@domain	
		Không có ký tự nào
Mật khẩu	Bất kì ký tự nào	Nhập không đúng mật khẩu hiện tại

Bảng 5.9. Bảng điều kiện ràng buộc cho Đăng nhập

## c) Danh sách các Test case:

STT Test case	Tên Test case	Loại kiểm thử	Kịch bản kiểm thử	Các bước thực hiện	Kết quả mong đợi	Pass (X) or Fail	Ghi chú
<b>Đăng nhập</b>							
1	TC_1	F	Email không hợp lệ. VD: email@	1. Người dùng nhấn mục Đăng nhập trên trang chủ 2. Người dùng nhập ô input email: email@	Hệ thống hiển thị cảnh báo "Invalid email"	x	
2	TC_2	F	Email hợp lệ nhưng không đúng trong database	1. Người dùng nhấn mục Đăng nhập trên trang chủ 2. Người dùng nhập email hợp lệ nhưng không có trong database	Hệ thống hiển thị cảnh báo "Email không tồn tại"	x	
3	TC_3	F	Email có tồn tại trong database và mật khẩu sai	1. Người dùng nhấn mục Đăng nhập trên trang chủ 2. Người dùng nhập email có tồn tại 3. Nhập mật khẩu sai	Hệ thống hiển thị cảnh báo "Email hoặc mật khẩu không đúng"	x	
4	TC_4	F	Email và mật khẩu đều đúng	1. Người dùng nhấn mục Đăng nhập trên trang chủ 2. Người dùng nhập email có tồn tại 3. Nhập mật khẩu có tồn tại	Hệ thống hiển thị thông báo "Đăng nhập thành công"	x	

Hình 5.2. Danh sách Test case Đăng nhập

## d) Thực hiện kiểm thử tự động bằng Selenium.

Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE. Ở đây ta có kịch bản kiểm thử với các test case được test tự động.

The screenshot shows the Selenium IDE interface with the project 'DATN\_DangNhap'. The test case 'TC\_4' is selected. The command table shows the following steps:

Command	Target	Value
6. ✓ type	id=password	123123
7. ✓ click	css=bg-primary	
8. ✓ mouse over	css=bg-primary	
9. ✓ mouse out	css=bg-primary	

Below the table, there are fields for 'Command' (open), 'Target' (http://localhost:3000/), 'Value', and 'Description'. The log pane at the bottom shows the execution results:

```

5. type on id=email with value admin@123.com OK
6. type on id=password with value 123123 OK
7. click on css=bg-primary OK
8. mouseOver on css=bg-primary OK
9. mouseOut on css=bg-primary OK
10. assertText on css=opacity-90 with value Đăng nhập thành công OK
'TC_4' completed successfully

```

Hình 5.3. Kết quả kiểm thử Đăng nhập bằng Selenium

## 2. Kiểm thử chức năng đổi mật khẩu

### a) Giao diện chức năng:

**Đổi mật khẩu**

Mật khẩu cũ

Mật khẩu mới

Nhập lại mật khẩu mới

Hủy Lưu thông tin

*Hình 5.4. Giao diện Đổi mật khẩu.*

### b) Mô tả giao diện và các điều kiện ràng buộc:

- Tiến hành nhập mật khẩu cũ và mật khẩu mới rồi nhấn nút “Lưu thông tin”.
- Điều kiện ràng buộc: Không được để trống 3 trường nêu trên.

Sử dụng phương pháp phân hoạch tương đương thiết lập các ca kiểm thử:

Điều kiện đầu vào	Vùng hợp lệ	Vùng không hợp lệ
Mật khẩu cũ	Đủ 6 ký tự	Không đủ 6 ký tự
Mật khẩu mới	Đủ 6 ký tự	Không đủ 6 ký tự
		Không khớp trường nhập lại mật khẩu
Nhập lại mật khẩu mới	Đủ 6 ký tự	Không đủ 6 ký tự
		Không khớp trường mật khẩu

*Bảng 5.10. Bảng điều kiện ràng buộc cho Đổi mật khẩu*

### c) Danh sách các Test case:

Đổi mật khẩu						
Tiền điều kiện: người dùng đã đăng nhập thành công						
1	TC_1	F	Nhập mật khẩu cũ hoặc mới 1 hoặc mới 2 không đủ 6 ký tự	1. Người dùng kích vào nút "Cài đặt" hoặc icon Cài đặt trên menu quản trị.	Hệ thống hiển thị cảnh báo "String must contain at least 6 character(s)"	x
2	TC_2	F	Nhập 2 mật khẩu mới không trùng nhau và hợp lệ	1. Người dùng kích vào nút "Cài đặt" hoặc icon Cài đặt trên menu quản trị.	Hệ thống hiển thị cảnh báo "Mật khẩu không trùng nhau"	x
3	TC_3	F	Nhập sai mật khẩu cũ và nhập 2 mật khẩu mới giống nhau và hợp lệ	1. Người dùng kích vào nút "Cài đặt" hoặc icon Cài đặt trên menu quản trị.	Hệ thống hiển thị cảnh báo "Mật khẩu cũ không đúng"	x
4	TC_4	F	Nhập đúng mật khẩu cũ và nhập 2 mật khẩu mới giống nhau và hợp lệ	1. Người dùng kích vào nút "Cài đặt" hoặc icon Cài đặt trên menu quản trị.	Hệ thống hiển thị thông báo "Đổi mật khẩu thành công"	x

*Hình 5.5. Danh sách Test case Đổi mật khẩu*

d) Thực hiện kiểm thử tự động bằng Selenium:

Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE. Ở đây ta có kịch bản kiểm thử với các test case được test tự động.

Test Step	Command	Target	Value
TC_1*	✓ click	css=opacity-90	
TC_2*	✓ double click	css=opacity-90	
TC_3*	✓ click	css=opacity-90	
TC_4*	✓ assert text	css=opacity-90	Đổi mật khẩu thành công

The 'Log' section at the bottom shows the execution details of the test steps:

```

9. click on css=.items-center:nth-child(4) > .bg-primary OK
10. click on css=.opacity-90 OK
11. click on css=.opacity-90 OK
12. doubleClick on css=.opacity-90 OK
13. click on css=.opacity-90 OK
14. assertText on css=.opacity-90 with value Đổi mật khẩu thành công OK
'TC_4' completed successfully

```

*Hình 5.6. Kết quả kiểm thử Đổi mật khẩu bằng Selenium*

3. Kiểm thử thông báo realtime cho người dùng về trạng thái đơn đặt hàng.

a) Giao diện chức năng

*Hình 5.7. Giao diện thông báo cho người dùng về trạng thái đơn đặt hàng*

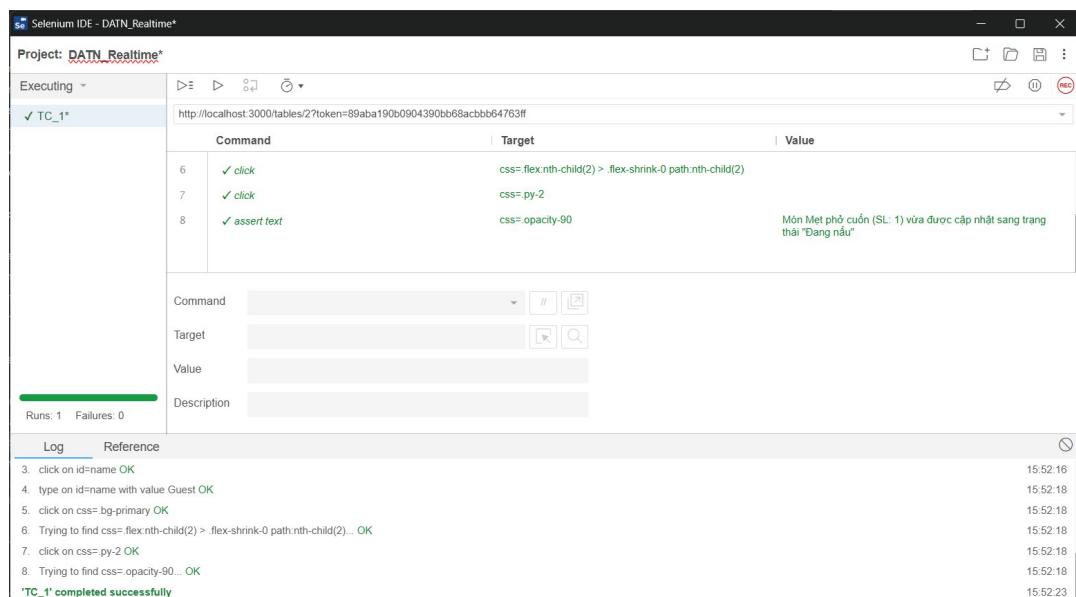
b) Danh sách các Test case:

Thông báo realtime cho người dùng về trạng thái đơn đặt hàng						
	TC_1	F	Nhân viên hoặc admin đổi trạng thái đơn và đặt món ăn 2. Nhân viên hoặc admin đổi trạng thái đơn	1. Người dùng đăng ký tài khoản và đặt món ăn	Hệ thống hiển thị thông báo "Món ăn ... đã được ..."	x
1						

*Hình 5.8. Danh sách Test case Kiểm thử thông báo cho người dùng về trạng thái đơn đặt hàng*

c) Thực hiện kiểm thử tự động bằng Selenium:

Công việc kiểm thử tự động được thực hiện bằng cách sử dụng Selenium IDE. Ở đây ta có kịch bản kiểm thử với các test case được test tự động.



*Hình 5.9. Kết quả kiểm thử Thông báo trực tiếp bằng Selenium*

## 5.2. Kết quả kiểm thử

- Không phát hiện lỗi trên website ở các tính năng có khả năng bị ảnh hưởng sau khi thêm các tính năng mới, tỷ lệ pass 100%.
- Quá trình kiểm thử hồi quy diễn ra suôn sẻ và không gặp khó khăn đáng kể.
- Các test case đều chạy ổn định trên công cụ Selenium.

### 5.3. Kết luận

Dựa trên kết quả kiểm thử của các chức năng cần thiết và các chức năng mới trên trang web, ta có một vài lời kết luận:

- Đánh giá về tính ổn định: Trang web có cho thấy tính ổn định tốt trong quá trình kiểm thử hồi quy. Không có các lỗi nghiêm trọng được phát hiện, như các lỗi giao diện người dùng (UI) cơ bản hay lỗi hoạt động của các tính năng chính.
- Độ phủ kiểm thử: Tỷ lệ phủ của các ca kiểm thử đối với các chức năng cần thiết và các chức năng mới là đủ để đảm bảo chất lượng. Các kịch bản kiểm thử đã bao quát được các trường hợp thông thường cũng như trường hợp biên của từng tính năng.
- Hiệu suất và thời gian phản hồi: Trang web đáp ứng nhanh và không có vấn đề đáng kể về hiệu suất trong quá trình kiểm thử. Thời gian phản hồi của các tương tác người dùng là hợp lý và không gây ngạt ngào.
- Cải tiến tiềm năng: Dựa trên những phản hồi từ kiểm thử, có một số đề xuất về cải tiến có thể áp dụng để cải thiện trải nghiệm người dùng hoặc tối ưu hóa hiệu suất. Tuy nhiên, không có vấn đề cấp bách cần phải giải quyết ngay lập tức.

## KẾT LUẬN

### 1. Kết quả đạt được.

Dự án *Phát triển ứng dụng web đặt đồ ăn cho nhà hàng sử dụng Next.js và .NET Core* đã được hoàn thành với các chức năng chính: người dùng có thể duyệt thực đơn, đặt món, và nhận thông báo trạng thái đơn hàng,... Ứng dụng được xây dựng dựa trên Clean Architecture, đảm bảo cấu trúc phần mềm rõ ràng, dễ bảo trì và mở rộng. Công nghệ Next.js mang lại hiệu suất cao, SEO tối ưu, trong khi .NET Core đảm bảo tính bảo mật và ổn định cho API. Hệ thống cũng hỗ trợ quản lý nhà hàng như theo dõi đơn hàng, quản lý thực đơn và khách hàng. Quá trình phát triển đã tuân thủ các nguyên tắc kiểm thử và đảm bảo chất lượng, đáp ứng nhu cầu thực tế.

### 2. Hướng phát triển.

Trong tương lai, ứng dụng có thể được mở rộng với các tính năng nâng cao như gợi ý món ăn dựa trên sở thích, tích hợp AI để tối ưu hóa quản lý nhà hàng, và triển khai ứng dụng di động nhằm nâng cao trải nghiệm người dùng. Đồng thời, hệ thống có thể mở rộng để hỗ trợ đa nhà hàng trên cùng nền tảng, tích hợp ví điện tử hoặc cổng thanh toán quốc tế, và tối ưu hóa hiệu suất với công nghệ serverless nhằm đáp ứng số lượng lớn người dùng. Việc tăng cường bảo mật và tuân thủ các tiêu chuẩn an toàn dữ liệu cũng sẽ là ưu tiên hàng đầu để đảm bảo hệ thống hoạt động ổn định và bảo vệ thông tin người dùng.

## TÀI LIỆU THAM KHẢO

- [1] Thạc Bình Cường – Nguyễn Đức Mận. (2023). *Kiểm thử và đảm bảo chất lượng phần mềm*. ĐH Bách Khoa Hà Nội.
- [2] Microsoft. (2023). *ASP.NET Core Documentation*.  
<https://learn.microsoft.com/en-us/aspnet/core/>
- [3] Dan Abramov & Next.js Team. (2023). *Next.js Documentation*.  
<https://nextjs.org/docs>.
- [4] Robert C. Martin. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
- [5] Mark Seemann. (2011). *Dependency Injection in .NET*. Manning Publications.