

ĐẠI HỌC BÁCH KHOA HÀ NỘI

# BÁO CÁO THỰC TẬP KỸ THUẬT

Xây dựng hệ thống quản lý vận hành và bán hàng cho  
nhà hàng cơm bình dân

LÊ ĐỨC LƯƠNG

luong.ld232006p@sis.hust.edu.vn

Chương trình đào tạo: Công nghệ thông tin

Giảng viên hướng dẫn: Ths. Nguyễn Mạnh Tuấn

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 12/2025

ĐẠI HỌC BÁCH KHOA HÀ NỘI

# BÁO CÁO THỰC TẬP KỸ THUẬT

Xây dựng hệ thống quản lý vận hành và bán hàng cho  
nhà hàng cơm bình dân

LÊ ĐỨC LƯƠNG

luong.ld232006p@sis.hust.edu.vn

Chương trình đào tạo: Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Nguyễn Mạnh Tuấn

Chữ ký GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 12/2025

# LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến Ban lãnh đạo và các thầy cô giáo Trường Công nghệ Thông tin và Truyền thông - Đại học Bách Khoa Hà Nội đã tận tình giảng dạy và trang bị cho em những kiến thức quý báu trong suốt thời gian ngồi trên ghế nhà trường.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc tới thầy **Nguyễn Mạnh Tuấn**. Thầy đã trực tiếp hướng dẫn, định hướng và đưa ra những lời khuyên chuyên môn thiết thực giúp em hoàn thiện đồ án.

Cuối cùng, em xin cảm ơn gia đình, người yêu và bạn bè đã luôn là chỗ dựa vững chắc, động viên em vượt qua những khó khăn trong quá trình thực hiện đồ án này. Dù đã rất cố gắng, đồ án khó tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của quý Thầy, Cô.

Một lần nữa, em xin chân thành cảm ơn!

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Mô hình kinh doanh quán cơm bình dân, một nét đặc trưng của ẩm thực Việt Nam phục vụ phần lớn sinh viên và người lao động, hiện đang đối mặt với những thách thức lớn trong khâu vận hành và quản lý. Các quy trình thủ công như ghi chép sổ sách hay trí nhớ của nhân viên thường dẫn đến sai sót trong tính toán, chậm trễ trong phục vụ giờ cao điểm và khó khăn trong việc kiểm soát nguyên liệu, gây thất thoát tài chính. Mặc dù thị trường đã có nhiều phần mềm quản lý nhà hàng (POS), nhưng đa số được thiết kế cho mô hình gọi món tiêu chuẩn với quy trình phức tạp, chi phí cao và không đáp ứng được tính linh hoạt đặc thù của quán cơm bình dân như thực đơn thay đổi theo ngày hay tốc độ phục vụ cực nhanh. Nhận thấy khoảng trống này, đồ án tập trung phát triển một hệ thống quản lý chuyên biệt, tối ưu hóa cho quy trình bán hàng nhanh gọn và quản lý linh hoạt.

Cách tiếp cận được lựa chọn là xây dựng một ứng dụng web dựa trên kiến trúc Client-Server [1], sử dụng React [2] cho giao diện người dùng và Spring Boot [3] cho phía máy chủ, kết hợp với cơ sở dữ liệu PostgreSQL [4]. Lựa chọn này đảm bảo khả năng tương thích đa nền tảng, dễ dàng triển khai và mở rộng, đồng thời cung cấp trải nghiệm người dùng mượt mà nhờ công nghệ Single Page Application (SPA).

Giải pháp tổng thể bao gồm việc xây dựng các phân hệ chức năng cốt lõi: một giao diện bán hàng (POS) tối giản cho phép nhân viên tạo đơn và thanh toán chỉ trong vài thao tác chạm; một hệ thống quản lý thực đơn động giúp chủ quán dễ dàng cập nhật món ăn theo ngày; và cơ chế quản lý tồn kho thông minh dựa trên định lượng suất ăn thay vì nguyên liệu thô. Bên cạnh đó, hệ thống còn cung cấp các báo cáo thống kê trực quan về doanh thu và xu hướng tiêu dùng, hỗ trợ chủ quán ra quyết định nhập hàng chính xác.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	3
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>4</b>
2.1 Khảo sát hiện trạng .....	4
2.2 Tổng quan chức năng .....	4
2.2.1 Biểu đồ use case tổng quát .....	5
2.2.2 Biểu đồ use case phân rã quản lý nhân viên .....	7
2.2.3 Biểu đồ use case phân rã Quản lý thực đơn .....	8
2.2.4 Biểu đồ use case phân rã Quản lý bàn ăn .....	9
2.2.5 Biểu đồ use case phân rã Quản lý đơn hàng .....	10
2.2.6 Quy trình nghiệp vụ .....	10
2.3 Đặc tả chức năng .....	12
2.3.1 Đặc tả use case Đặt hàng trực tuyến .....	12
2.3.2 Đặc tả use case Tạo đơn hàng nhanh .....	13
2.3.3 Đặc tả use case Thanh toán .....	14
2.3.4 Đặc tả use case Thêm nhân viên .....	15
2.4 Yêu cầu phi chức năng .....	16
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>17</b>
3.1 Java .....	17
3.2 Spring Boot.....	17
3.3 React .....	17

3.4 PostgreSQL.....	18
3.4.1 Redis .....	18
3.5 WebSocket .....	19
<b>CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG ....</b>	<b>20</b>
4.1 Thiết kế kiến trúc.....	20
4.1.1 Lựa chọn kiến trúc phần mềm .....	20
4.1.2 Thiết kế tổng quan.....	21
4.1.3 Thiết kế chi tiết gói .....	23
4.2 Thiết kế chi tiết.....	25
4.2.1 Thiết kế giao diện .....	25
4.2.2 Thiết kế lớp .....	29
4.2.3 Thiết kế cơ sở dữ liệu .....	31
4.3 Thiết kế chi tiết cơ sở dữ liệu .....	32
4.4 Thiết kế chi tiết cơ sở dữ liệu .....	32
<b>CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT .....</b>	<b>35</b>
5.1 Cơ chế quản lý thực đơn động và kiểm soát phiên bán hàng .....	35
5.2 Tái thiết kế quy trình bán hàng Quick Serve và giao diện chạm .....	36
5.3 Kiến trúc phân tầng và bảo mật với Data Transfer Object .....	36
5.4 Cơ chế Snapshot dữ liệu đảm bảo tính toàn vẹn tài chính .....	37
5.5 Chiến lược Caching đa tầng với Redis.....	37
5.5.1 Hệ thống giao tiếp thời gian thực dựa trên WebSocket .....	38
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>39</b>
6.1 Kết luận .....	39
6.2 Hạn chế và hướng phát triển .....	39
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>41</b>

## **DANH MỤC HÌNH VẼ**

Hình 2.1	Biểu đồ use case tổng quát . . . . .	5
Hình 2.2	Biểu đồ use case phân rã Quản lý nhân viên . . . . .	7
Hình 2.3	Biểu đồ use case Quản lý thực đơn và giá . . . . .	8
Hình 2.4	Biểu đồ use case Quản lý bàn ăn . . . . .	9
Hình 2.5	Biểu đồ use case Quản lý đơn hàng . . . . .	10
Hình 2.6	Biểu đồ hoạt động minh họa đặt hàng trực tuyến . . . . .	11
Hình 2.7	Biểu đồ hoạt động minh họa gọi món, tạo đơn thanh toán nhanh và chọn phương thức thanh toán . . . . .	12
Hình 4.1	Mô hình 3 tầng . . . . .	20
Hình 4.2	Biểu đồ phụ thuộc gói . . . . .	21
Hình 4.3	Biểu đồ thiết kế gói Model . . . . .	23
Hình 4.4	Biểu đồ thiết kế gói Controller . . . . .	24
Hình 4.5	Giao diện bán hàng chính . . . . .	25
Hình 4.6	Giao diện báo cáo thống kê . . . . .	26
Hình 4.7	Giao diện đặt hàng . . . . .	26
Hình 4.8	Giao diện giỏ hàng . . . . .	26
Hình 4.9	Giao diện popup thanh toán nhanh . . . . .	27
Hình 4.10	Giao diện quản lý bàn ăn . . . . .	27
Hình 4.11	Giao diện quản lý đơn hàng . . . . .	27
Hình 4.12	Giao diện quản lý nhân viên . . . . .	28
Hình 4.13	Giao diện quản lý thực đơn . . . . .	28
Hình 4.14	Giao diện tổng quan . . . . .	28
Hình 4.15	Giao diện trạng thái đơn hàng . . . . .	29
Hình 4.16	Thiết kế lớp Order . . . . .	29
Hình 4.17	Thiết kế lớp OrderService . . . . .	29
Hình 4.18	Thiết kế lớp Product . . . . .	30
Hình 4.19	Biểu đồ trình tự Tạo đơn hàng nhanh . . . . .	30
Hình 4.20	Biểu đồ trình tự Thanh toán . . . . .	30
Hình 4.21	Biểu đồ thực thể liên kết . . . . .	31
Hình 4.22	Biểu đồ thiết kế cơ sở dữ liệu . . . . .	32

## **DANH MỤC BẢNG BIỂU**

Bảng 2.1	Đặc tả chức năng Đặt hàng trực tuyến . . . . .	13
Bảng 2.2	Đặc tả chức năng Tạo đơn hàng nhanh . . . . .	14
Bảng 2.3	Đặc tả chức năng Thanh toán . . . . .	15
Bảng 2.4	Đặc tả chức năng Thêm nhân viên . . . . .	16
Bảng 4.1	Mục đích/Nhiệm vụ của các gói trong hệ thống . . . . .	22
Bảng 4.2	Mô tả bảng Người dùng (users) . . . . .	33
Bảng 4.3	Mô tả bảng Vai trò (roles) . . . . .	33
Bảng 4.4	Mô tả bảng Sản phẩm/Món ăn (products) . . . . .	33
Bảng 4.5	Mô tả bảng Đơn hàng (orders) . . . . .	34
Bảng 4.6	Mô tả bảng Chi tiết đơn hàng (order_items) . . . . .	34

## **DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

<b>Thuật ngữ</b>	<b>Ý nghĩa</b>
API	Giao diện lập trình ứng dụng (Application Programming Interface)
Caching	Lưu dữ liệu trong bộ nhớ chính để có tốc độ truy xuất nhanh hơn
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
POS	Hệ thống bán hàng (Point of Sale)
RESTful API	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
SPA	Ứng dụng web đơn trang (Single Page Application)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại Việt Nam, ngành dịch vụ ăn uống là một trong những lĩnh vực tiên phong trong việc áp dụng công nghệ để tối ưu hóa vận hành. Các hệ thống quản lý nhà hàng (POS), đặt món trực tuyến hay quản lý chuỗi cung ứng đã trở nên phổ biến, giúp các doanh nghiệp nâng cao hiệu quả kinh doanh và trải nghiệm khách hàng. Tuy nhiên, phần lớn các giải pháp hiện có trên thị trường thường tập trung vào mô hình nhà hàng gọi món, quán cà phê hoặc chuỗi thức ăn nhanh với thực đơn cố định và quy trình phục vụ tiêu chuẩn.

Ngược lại, mô hình kinh doanh "cơm bình dân" – một nét đặc trưng trong văn hóa ẩm thực và đời sống của người Việt, đặc biệt phục vụ đối tượng nhân viên văn phòng và sinh viên – lại có những đặc thù riêng biệt mà các phần mềm quản lý thông thường chưa đáp ứng tốt. Các đặc thù này bao gồm: áp lực phục vụ cực lớn trong khung giờ cao điểm ngắn (thường chỉ từ 11h30 đến 12h30); thực đơn thay đổi linh hoạt theo ngày dựa trên nguyên liệu chợ đầu mối; và quy trình bán hàng yêu cầu tốc độ xử lý giao dịch tính bằng giây để tránh ùn tắc.

Hiện nay, đa số các quán cơm bình dân vẫn vận hành theo phương thức thủ công hoặc sử dụng các phần mềm POS không chuyên dụng. Việc này dẫn đến nhiều bất cập: Thứ nhất, tốc độ bán hàng bị kìm hãm do nhân viên phải thực hiện nhiều thao tác chọn món phức tạp trên các giao diện không tối ưu, gây ùn tắc cục bộ vào giờ cao điểm. Thứ hai, chủ kinh doanh thiếu các dữ liệu thống kê chính xác về thói quen ăn uống của khách hàng và xu hướng tiêu thụ từng món theo ngày để đưa ra quyết định nhập hàng tối ưu cho ngày hôm sau.

Việc thiếu vắng một công cụ quản lý chuyên biệt, có khả năng thích ứng với sự thay đổi thực đơn hàng ngày và đảm bảo tốc độ xử lý giao dịch cao, đang là rào cản lớn đối với sự phát triển và chuyên nghiệp hóa của mô hình kinh doanh này. Xuất phát từ nhu cầu thực tế đó, việc nghiên cứu và xây dựng một hệ thống quản lý đặc thù, tập trung vào tốc độ và sự linh hoạt cho mô hình cơm bình dân là một yêu cầu cấp thiết, mang lại giá trị thực tiễn cao cho cả người quản lý, nhân viên vận hành và khách hàng.

## 1.2 Mục tiêu và phạm vi đề tài

Trên thị trường hiện nay, các giải pháp quản lý nhà hàng phổ biến thường được thiết kế tối ưu cho mô hình gọi món tiêu chuẩn hoặc chuỗi thức ăn nhanh, dẫn đến tình trạng dư thừa tính năng nhưng lại thiếu đi sự linh hoạt cần thiết khi áp dụng

vào đặc thù của quán cơm bình dân. Xuất phát từ thực tế đó, mục tiêu cốt lõi của đề tài là nghiên cứu và xây dựng một hệ thống quản lý chuyên biệt, tập trung giải quyết triệt để bài toán về tốc độ xử lý giao dịch trong khung giờ cao điểm và cơ chế quản lý thực đơn thay đổi theo ngày. Hệ thống hướng tới việc thay thế các phương thức ghi chép thủ công hoặc các phần mềm không chuyên, giúp chủ cửa hàng chuẩn hóa quy trình vận hành và kiểm soát hiệu quả nguồn lực kinh doanh.

Về phạm vi thực hiện, đồ án tập trung phát triển một ứng dụng web hoàn chỉnh bao gồm các phân hệ chức năng gắn liền với quy trình nghiệp vụ thực tế. Cụ thể, đối với nghiệp vụ bán hàng, hệ thống cung cấp giao diện POS tối giản nhằm giảm thiểu thao tác và rút ngắn thời gian thanh toán. Song song với đó, phân hệ quản trị được xây dựng với khả năng thiết lập thực đơn động linh hoạt và cơ chế quản lý tồn kho dựa trên số lượng suất ăn thành phẩm thay vì nguyên liệu thô. Ngoài ra, phạm vi đề tài cũng bao gồm việc xây dựng cổng thông tin cho phép khách hàng tra cứu thực đơn và đặt món trực tuyến, cùng với hệ thống báo cáo thống kê hỗ trợ chủ cửa hàng theo dõi doanh thu và xu hướng tiêu dùng theo thời gian thực.

### 1.3 Định hướng giải pháp

Để đảm bảo khả năng truy cập đa nền tảng và dễ dàng triển khai, hệ thống được xây dựng theo kiến trúc Client-Server [1]. Cụ thể, phía máy khách sẽ được phát triển dưới dạng ứng dụng đơn trang (SPA) nhằm tối ưu hóa trải nghiệm người dùng, giảm thiểu thời gian tải trang và tăng tốc độ tương tác – yếu tố then chốt cho môi trường phục vụ cường độ cao. Phía máy chủ sẽ được xây dựng theo chuẩn giao tiếp RESTful API [5], đảm bảo tính tách biệt giữa giao diện và logic xử lý, thuận tiện cho việc mở rộng hoặc tích hợp với các hệ thống khác trong tương lai. Giải pháp cụ thể của đồ án tập trung vào việc xử lý hai bài toán cốt lõi là tốc độ và sự linh hoạt thông qua các kỹ thuật sau. Thứ nhất, đối với quy trình bán hàng, hệ thống sẽ tối ưu hóa giao diện người dùng và luồng xử lý dữ liệu để giảm thiểu số bước thao tác, đồng thời áp dụng các cơ chế caching phía máy khách để tăng tốc độ phản hồi ngay cả khi đường truyền mạng không ổn định. Thứ hai, đối với vấn đề thực đơn động và quản lý tồn kho, hệ thống sẽ thiết kế cơ sở dữ liệu linh hoạt cho phép định nghĩa món ăn theo ngày và xây dựng thuật toán tự động trừ kho dựa trên định lượng suất ăn đã được cấu hình trước, giúp cập nhật trạng thái "hết hàng" theo thời gian thực lên giao diện bán hàng và trang đặt món của khách hàng. Đóng góp chính của đồ án là cung cấp một giải pháp phần mềm quản lý toàn diện và chuyên biệt cho mô hình kinh doanh cơm bình dân, một phân khúc thị trường lớn nhưng chưa được phục vụ thỏa đáng bởi các giải pháp công nghệ hiện có. Kết quả đạt được là một hệ thống hoàn chỉnh có khả năng vận hành thực tế, giúp chủ cửa hàng chuẩn hóa quy trình, kiểm soát chặt chẽ nguồn lực và nâng cao chất lượng phục vụ,

đồng thời tạo tiền đề dữ liệu cho các phân tích kinh doanh sâu hơn trong tương lai.

### 1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức thành năm chương với nội dung cụ thể như sau.

Chương 2 tập trung vào việc khảo sát và phân tích yêu cầu của hệ thống. Trong chương này, em sẽ trình bày kết quả khảo sát thực tế về quy trình vận hành tại các quán cơm bình dân, từ đó xác định các tác nhân tham gia và xây dựng biểu đồ Use Case tổng quát cũng như các kịch bản nghiệp vụ chi tiết. Các yêu cầu phi chức năng về hiệu năng, bảo mật và tính khả dụng cũng được phân tích kỹ lưỡng tại đây làm cơ sở cho việc thiết kế hệ thống.

Chương 3 giới thiệu về các công nghệ và nền tảng được sử dụng để phát triển ứng dụng. Nội dung chương sẽ phân tích lý do lựa chọn kiến trúc SPA với React [2] cho phía máy khách và Spring Boot [3] cho phía máy chủ, cùng với hệ quản trị cơ sở dữ liệu PostgreSQL [4]. Các công nghệ hỗ trợ như WebSocket [6] cho giao tiếp thời gian thực hay Redis [7] cho cơ chế caching cũng được trình bày nhằm làm rõ tính phù hợp với bài toán đặt ra.

Chương 4 trình bày chi tiết về quá trình thiết kế, hiện thực hóa và đánh giá hệ thống. Nội dung bao gồm các thiết kế kiến trúc tổng thể, thiết kế cơ sở dữ liệu, và thiết kế giao diện người dùng. Phần này cũng mô tả quá trình xây dựng các chức năng cốt lõi như quản lý thực đơn động, xử lý đơn hàng tốc độ cao, và trình bày kết quả kiểm thử hệ thống để chứng minh tính đúng đắn và hiệu quả của giải pháp.

Chương 5 tổng hợp các đóng góp nổi bật và các giải pháp kỹ thuật đặc thù đã áp dụng để giải quyết bài toán nghiệp vụ. Tại đây, em sẽ đi sâu phân tích thuật toán quản lý tồn kho theo suất ăn và cơ chế tối ưu hóa thao tác người dùng trên giao diện bán hàng, những điểm làm nên sự khác biệt của hệ thống so với các sản phẩm đại trà.

Cuối cùng, Chương 6 đưa ra kết luận chung về những kết quả đã đạt được so với mục tiêu ban đầu, đồng thời chỉ ra những hạn chế còn tồn tại và đề xuất các hướng phát triển trong tương lai để hoàn thiện sản phẩm.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát hiện trạng

Dựa trên thông tin khảo sát hiện trạng tại các quán cơm bình dân trên địa bàn Hà Nội, chúng tôi nhận thấy đa số các cơ sở vẫn vận hành theo quy trình truyền thống: khách hàng xếp hàng, trực tiếp chọn món tại quầy, nhân viên phục vụ múc đồ và báo giá ngay tại chỗ. Việc thanh toán thường diễn ra sau khi khách ăn xong dựa trên trí nhớ hoặc thẻ tạm. Quy trình này bộc lộ rõ những hạn chế vào giờ cao điểm, đặc biệt là tình trạng ùn tắc do tốc độ phục vụ chậm, rủi ro thất thoát doanh thu do sai sót trong tính toán, và khó khăn trong việc kiểm soát định lượng nguyên liệu tiêu hao so với số lượng suất ăn thực tế.

Qua phân tích các giải pháp phần mềm quản lý nhà hàng phổ biến như iPOS [8], CukCuk [9], và KiotViet [10], mặc dù các hệ thống này sở hữu tính năng mạnh mẽ và chuyên nghiệp, nhưng khi áp dụng vào mô hình cơm bình dân lại bộc lộ những bất cập nhất định. iPOS có chi phí đầu tư cao và quy trình thiết lập phức tạp, không phù hợp với quy mô nhỏ; CukCuk dù thân thiện hơn nhưng vẫn thiên về quản lý thực đơn cố định, gây khó khăn khi thay đổi món ăn hàng ngày; KiotViet mạnh về bán lẻ nhưng chưa tối ưu cho việc quản lý định lượng suất ăn chế biến với sự biến động nguyên liệu thường xuyên.

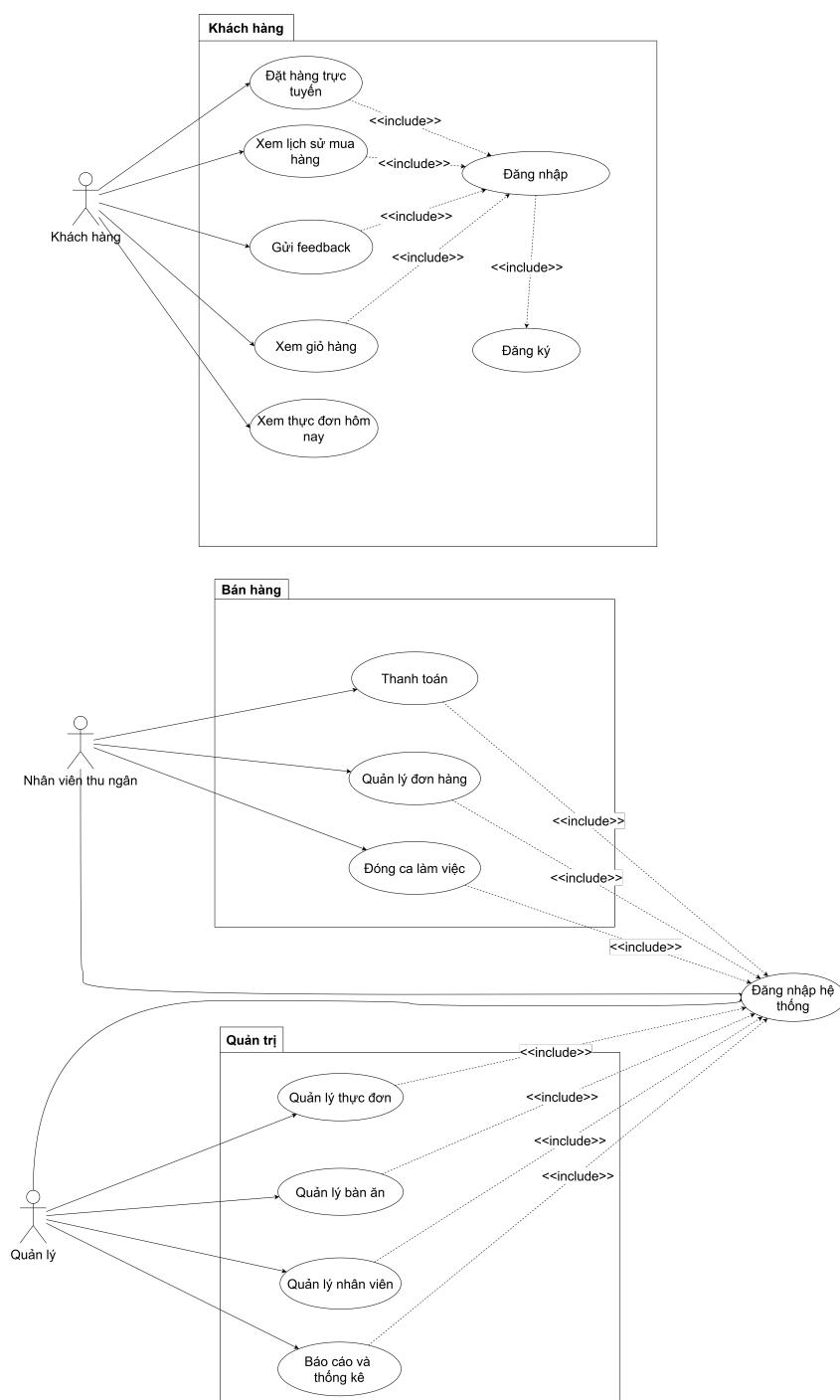
Từ những phân tích trên, có thể khẳng định nhu cầu cấp thiết về một hệ thống quản lý chuyên biệt cho mô hình cơm bình dân. Hệ thống này cần tập trung giải quyết bài toán về tốc độ xử lý giao dịch (dưới 30 giây/đơn), đơn giản hóa tối đa quy trình quản lý thực đơn động theo ngày, và cung cấp cơ chế kiểm soát tồn kho suất ăn trực quan, hiệu quả, giúp chủ quán tối ưu hóa vận hành và giảm thiểu thất thoát.

### 2.2 Tổng quan chức năng

Sau khi thu thập ý kiến của các chủ quán ăn bình dân, bạn bè (những người đã ăn những quán ăn ở xung quanh Hà Nội nhiều năm), cũng như những trải nghiệm thực tế của bản thân sau nhiều năm thường thức đồ ăn tại các quán ăn bình dân, em thiết kế hệ thống nhằm đáp ứng các yêu cầu nghiệp vụ cốt lõi của một quán cơm bình dân, tập trung vào việc tối ưu hóa quy trình bán hàng và quản lý vận hành. Về mặt tổng thể, hệ thống cung cấp các nhóm chức năng chính bao gồm quản lý bán hàng tại quầy, quản lý thực đơn và kho hàng, báo cáo thống kê, và cổng thông tin dành cho khách hàng. Đối với nhân viên thu ngân, hệ thống cung cấp công cụ tạo đơn hàng nhanh thông qua việc lựa chọn các gói giá định sẵn, hỗ trợ thanh toán đa phương thức và in hóa đơn tự động, giúp rút ngắn tối đa thời gian phục vụ trong giờ cao điểm. Song song với đó, bộ phận quản lý được trang bị các công cụ để thiết

lập thực đơn theo ngày, tùy chỉnh giá bán, và theo dõi biến động nguyên vật liệu dựa trên số lượng suất ăn tiêu thụ. Ngoài ra, hệ thống còn tích hợp khả năng dự báo nhu cầu nhập hàng dựa trên dữ liệu lịch sử và cung cấp các báo cáo trực quan về doanh thu, hiệu suất bán hàng, giúp chủ cửa hàng có cái nhìn toàn diện và đưa ra các quyết định kinh doanh chính xác.

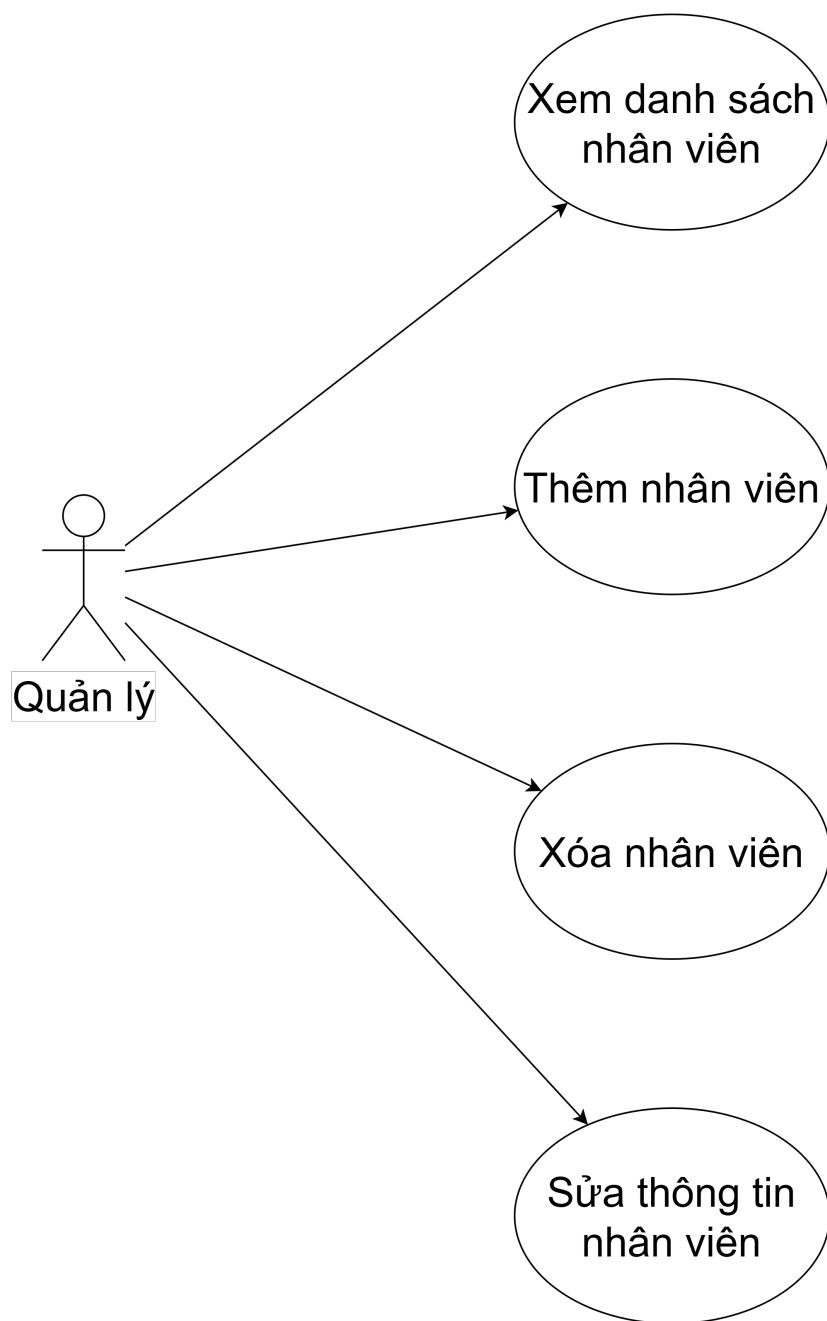
### 2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát

Hình 2.1 là biểu đồ use case tổng quát của hệ thống với các use case chính là đặt hàng trực tuyến, xem thực đơn hôm nay với tác nhân là người dùng, đặc biệt với tác nhân là khách hàng đã được đăng nhập được gọi là khách hàng thành viên thì sẽ có thể xem lịch sử mua hàng và gửi feedback cho bữa ăn mà mình đã đặt. Với tác nhân là nhân viên thu ngân và quản lý, mọi công việc được thực hiện trên hệ thống đều phải đăng nhập trước. Đối với tác nhân là nhân viên thu ngân sau khi đăng nhập, nhân viên thu ngân có thể tạo đơn hàng nhanh và thực hiện thanh toán cho đơn hàng đó. Sau mỗi ca làm việc, nhân viên sẽ đóng ca làm việc để hệ thống sẽ tính toán các báo cáo theo ca làm việc. Cuối cùng, tác nhân quản lý sau khi đăng nhập sẽ có thể quản lý thực đơn và giá của từng món trên thực đơn, quản lý nhân viên và tạo báo cáo về việc kinh doanh của mình.

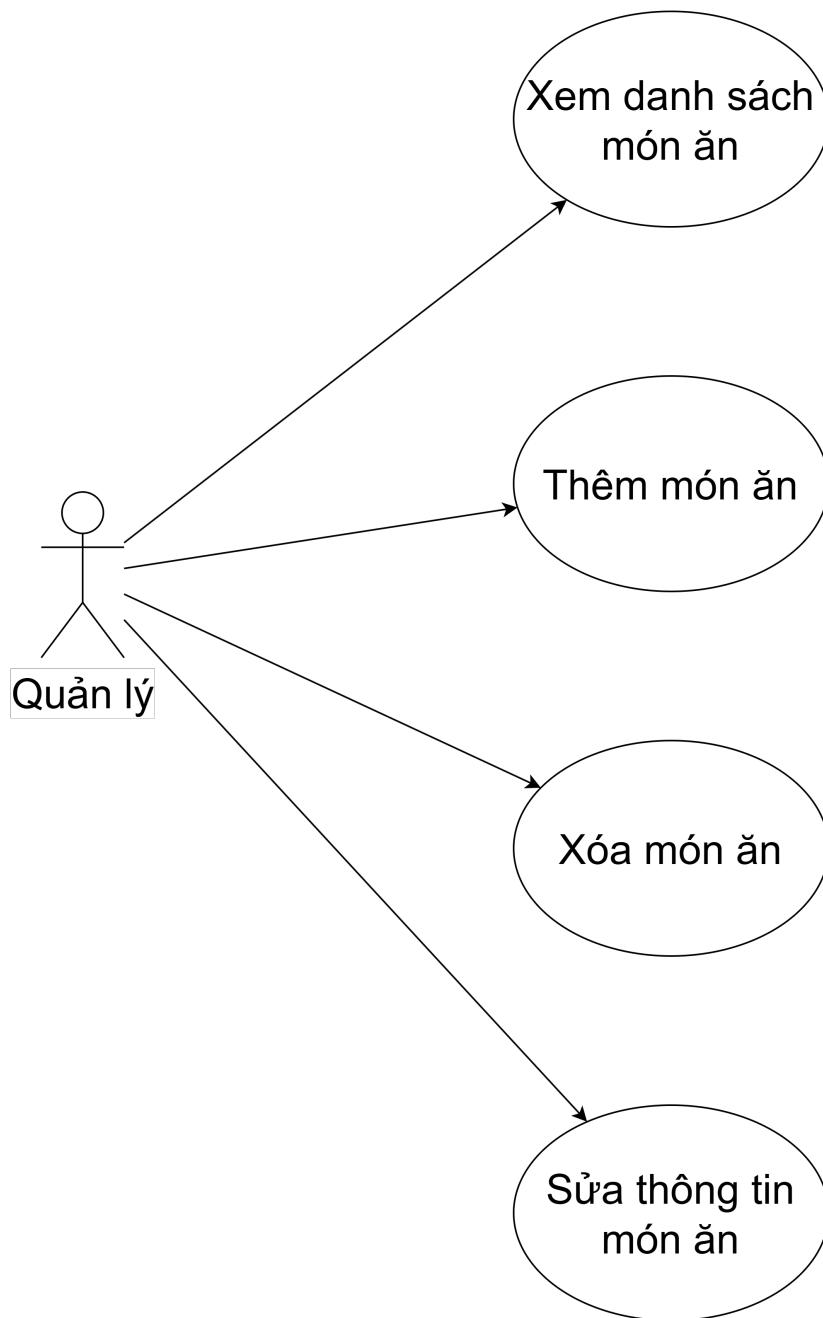
### 2.2.2 Biểu đồ use case phân rã quản lý nhân viên



**Hình 2.2:** Biểu đồ use case phân rã Quản lý nhân viên

Hình 2.2 là biểu đồ use case phân ra quản lý nhân viên. Quản lý có thể xem danh sách nhân viên đang làm việc tại quán của mình, thêm nhân viên mới, xóa nhân viên và sửa thông tin nhân viên đã tồn tại trong hệ thống.

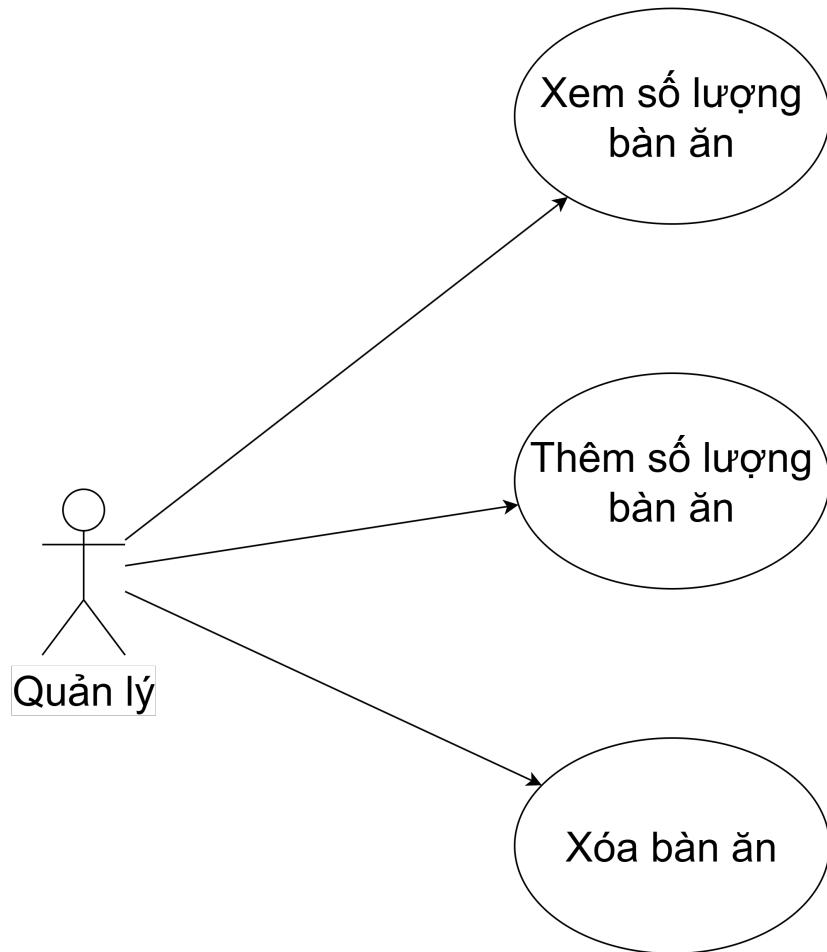
### 2.2.3 Biểu đồ use case phân rã Quản lý thực đơn



**Hình 2.3:** Biểu đồ use case Quản lý thực đơn và giá

Hình 2.3 là biểu đồ use case phân ra quản lý thực đơn. Quản lý có thể xem danh sách món ăn trong thực đơn hiện tại, thêm món ăn mới vào thực đơn, xóa món ăn khỏi thực đơn và sửa thông tin món ăn đã tồn tại trong thực đơn.

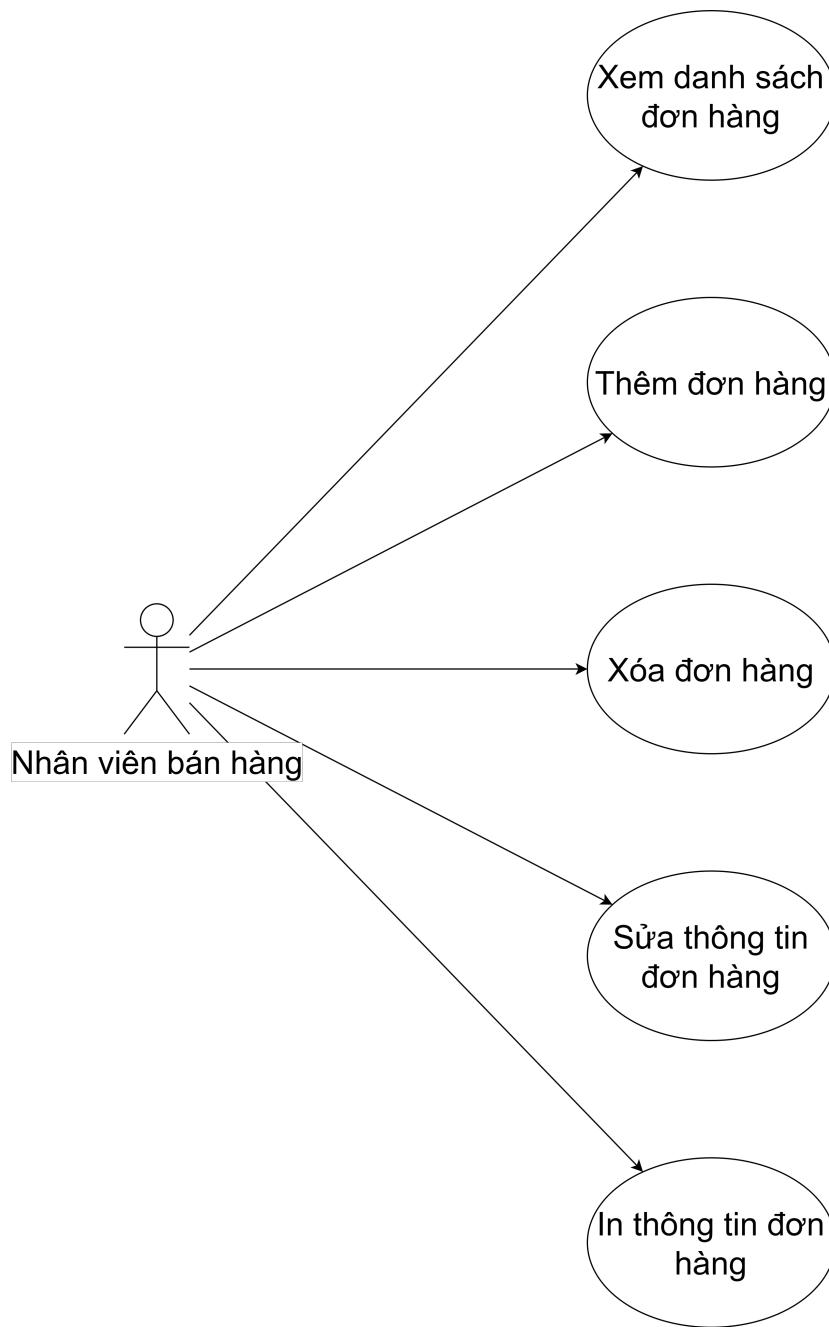
#### 2.2.4 Biểu đồ use case phân rã Quản lý bàn ăn



**Hình 2.4:** Biểu đồ use case Quản lý bàn ăn

Hình 2.3 là biểu đồ use case phân ra quản lý bàn ăn. Quản lý có thể xem số lượng bàn ăn đang có hiện tại, thêm số lượng bàn ăn và xóa bớt số lượng bàn ăn.

### 2.2.5 Biểu đồ use case phân rã Quản lý đơn hàng

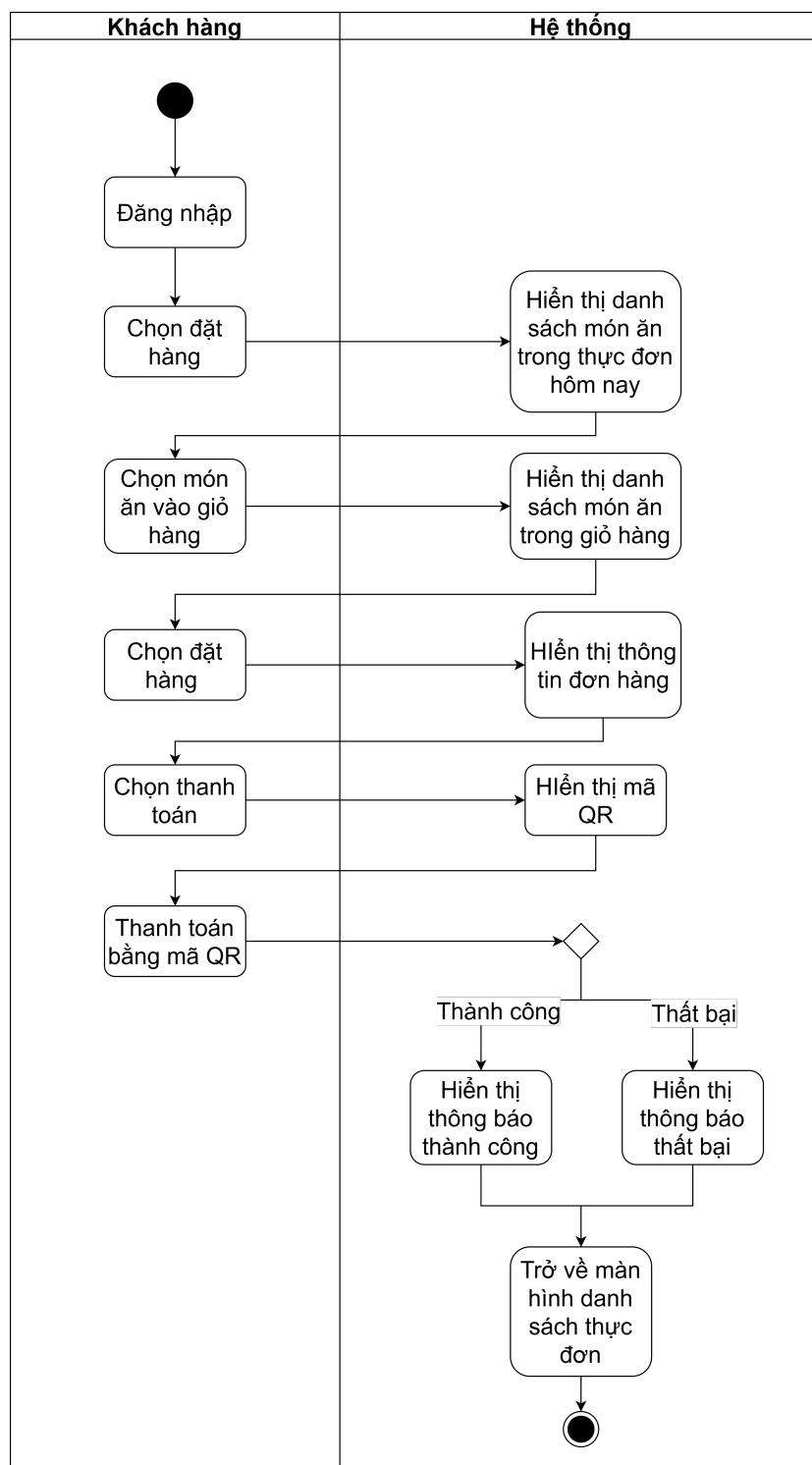


**Hình 2.5:** Biểu đồ use case Quản lý đơn hàng

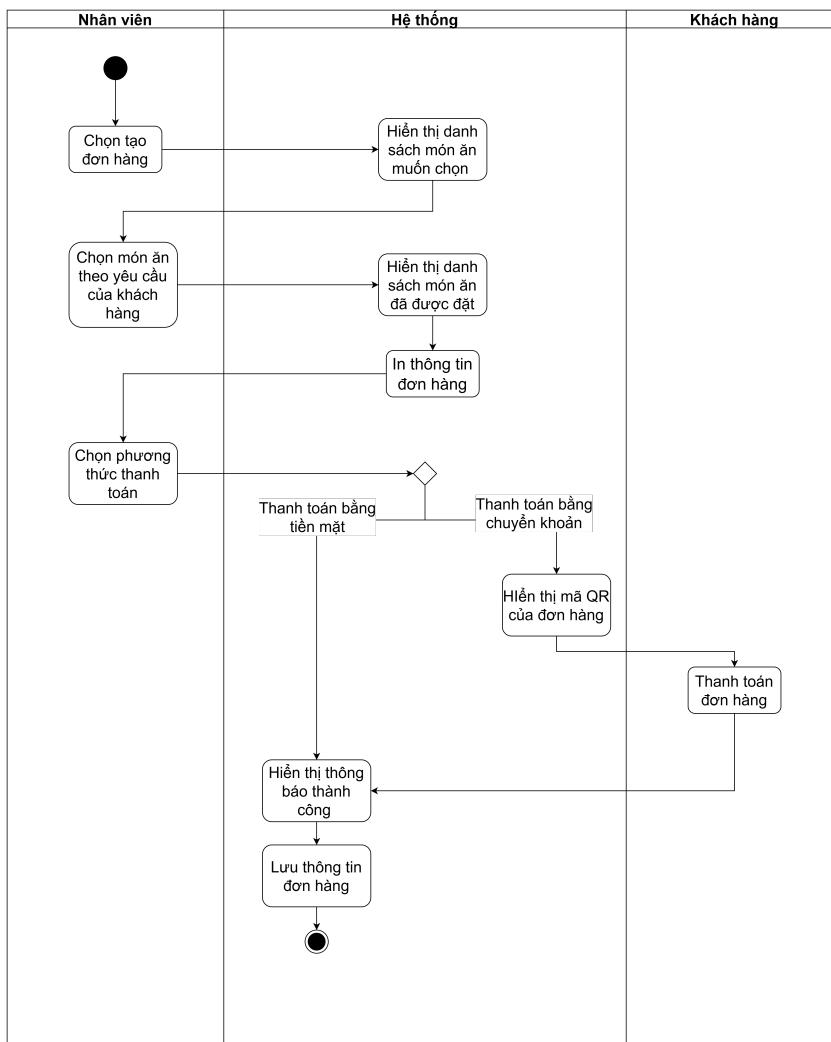
Hình 2.3 là biểu đồ use case phân ra quản lý đơn hàng. Nhân viên có thể xem danh sách đơn hàng hiện tại, xóa đơn hàng khỏi hàng chờ, thêm đơn hàng mới và chỉnh sửa đơn hàng

### 2.2.6 Quy trình nghiệp vụ

Dưới đây là 2 biểu đồ hoạt động minh họa cho các chức năng chính của hệ thống bao gồm đặt hàng online trên web; gọi món, tạo đơn thanh toán nhanh và chọn phương thức thanh toán.



**Hình 2.6:** Biểu đồ hoạt động minh họa đặt hàng trực tuyến



**Hình 2.7:** Biểu đồ hoạt động minh họa gọi món, tạo đơn thanh toán nhanh và chọn phương thức thanh toán

### 2.3 Đặc tả chức năng

#### 2.3.1 Đặc tả use case Đặt hàng trực tuyến

Mã ca sử dụng	UC_01
Tên ca sử dụng	Đặt hàng trực tuyến
Tên tác nhân	Khách hàng
Mô tả	Cho phép khách hàng gửi yêu cầu đặt hàng đến hệ thống của nhà hàng.
Tiền điều kiện	Khách hàng đăng nhập vào hệ thống.

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>Khách hàng xem danh sách thực đơn và chọn món ăn mong muốn.</li> <li>Hệ thống hiển thị chi tiết món ăn (tên, giá, mô tả).</li> <li>Khách hàng nhập số lượng và nhấn "Thêm vào giỏ hàng".</li> <li>Khách hàng truy cập giỏ hàng và nhấn "Đặt hàng".</li> <li>Khách hàng nhập thông tin nhận hàng (Địa chỉ, Số điện thoại, Ghi chú).</li> <li>Hệ thống hiển thị tổng tiền và phí vận chuyển (nếu có).</li> <li>Khách hàng nhấn "Xác nhận đặt hàng".</li> <li>Hệ thống lưu đơn hàng, thông báo đặt hàng thành công và gửi đơn đến nhân viên bán hàng.</li> </ol>
<b>Hậu điều kiện</b>	Một đơn hàng mới được tạo trong cơ sở dữ liệu với trạng thái "Chờ xác nhận".
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>Nếu món ăn đã hết:             <ol style="list-style-type: none"> <li>Hệ thống thông báo món ăn hiện tại không khả dụng.</li> <li>Yêu cầu khách hàng chọn món khác hoặc quay lại sau.</li> </ol> </li> <li>Nếu khách hàng đã đăng nhập:             <ol style="list-style-type: none"> <li>Hệ thống tự động điền thông tin địa chỉ và số điện thoại đã lưu.</li> </ol> </li> </ol>

**Bảng 2.1:** Đặc tả chức năng Đặt hàng trực tuyến

### 2.3.2 Đặc tả use case Tạo đơn hàng nhanh

<b>Mã ca sử dụng</b>	UC_02
<b>Tên ca sử dụng</b>	Tạo đơn hàng nhanh
<b>Tên tác nhân</b>	Nhân viên thu ngân
<b>Mô tả</b>	Hỗ trợ nhân viên thu ngân tạo đơn hàng tại quầy.
<b>Tiền điều kiện</b>	Nhân viên thu ngân đã đăng nhập vào hệ thống và đang ở màn hình bán hàng.

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Nhân viên chọn các món ăn khách gọi trên giao diện.</li> <li>2. Hệ thống tự động thêm món vào danh sách chờ và tính tạm tính.</li> <li>3. Nhân viên điều chỉnh số lượng món ăn (nếu khách gọi nhiều suất) hoặc thêm ghi chú.</li> <li>4. Hệ thống cập nhật lại tổng tiền theo thời gian thực và đẩy đơn hàng vào hàng đợi.</li> <li>5. Nhân viên nhấn nút "Thanh toán" để chuyển sang use case thanh toán.</li> </ol>
<b>Hậu điều kiện</b>	Đơn hàng được chuyển sang trạng thái "Chờ thanh toán".
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>3.1. Khách hàng đổi ý bỏ bớt món:             <ol style="list-style-type: none"> <li>3.1.1. Nhân viên chọn món cần xóa trong danh sách chờ.</li> <li>3.1.2. Nhấn nút "Xóa" hoặc phím tắt tương ứng.</li> <li>3.1.3. Hệ thống loại bỏ món và tính lại tổng tiền.</li> </ol> </li> </ol>

**Bảng 2.2:** ĐẶC TẢ CHỨC NĂNG TẠO ĐƠN HÀNG NHANH

### 2.3.3 ĐẶC TẢ USE CASE THANH TOÁN

<b>Mã ca sử dụng</b>	UC_03
<b>Tên ca sử dụng</b>	Thanh toán
<b>Tên tác nhân</b>	Nhân viên thu ngân
<b>Mô tả</b>	Xử lý giao dịch thanh toán cho đơn hàng, hỗ trợ tính tiền thừa.
<b>Tiền điều kiện</b>	Một đơn hàng đã được tạo và đang ở trạng thái chờ thanh toán.

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị tổng số tiền khách cần trả.</li> <li>2. Nhân viên chọn phương thức thanh toán (tiền mặt, chuyển khoản).</li> <li>3. Nhân viên nhập số tiền khách đưa (đối với tiền mặt).</li> <li>4. Hệ thống tự động tính toán và hiển thị số tiền thừa cần trả lại khách.</li> <li>5. Nhân viên nhấn "Hoàn tất".</li> <li>6. Hệ thống lưu trạng thái đơn hàng là "Đã thanh toán", cập nhật doanh thu.</li> </ol>
<b>Hậu điều kiện</b>	Đơn hàng hoàn tất, doanh thu được lưu trong hệ thống.
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>3.1. Khách thanh toán bằng chuyển khoản ngân hàng:             <ol style="list-style-type: none"> <li>3.1.1. Hệ thống hiển thị mã QR động theo số tiền đơn hàng.</li> <li>3.1.2. Hệ thống xác nhận giao dịch thành công.</li> </ol> </li> <li>3.2. Số tiền khách đưa không đủ:             <ol style="list-style-type: none"> <li>3.2.1. Hệ thống cảnh báo số tiền thiếu.</li> <li>3.2.2. Yêu cầu nhập lại số tiền hợp lệ.</li> </ol> </li> </ol>

**Bảng 2.3:** Đặc tả chức năng Thanh toán

#### 2.3.4 Đặc tả use case Thêm nhân viên

<b>Mã ca sử dụng</b>	UC_04
<b>Tên ca sử dụng</b>	Thêm nhân viên
<b>Tên tác nhân</b>	Quản lý
<b>Mô tả</b>	Tạo tài khoản mới cho nhân viên (thu ngân, phục vụ)
<b>Tiền điều kiện</b>	Quản lý đã đăng nhập với quyền Admin cao nhất.

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản lý truy cập vào mục "Quản lý nhân viên".</li> <li>2. Nhấn nút "Thêm nhân viên".</li> <li>3. Nhập thông tin cá nhân (Họ tên, SĐT, CCCD/CMND) và thông tin tài khoản (Tên đăng nhập, Mật khẩu).</li> <li>4. Nhấn nút "Tạo tài khoản".</li> <li>5. Hệ thống xác thực thông tin và lưu tài khoản mới.</li> <li>6. Hệ thống thông báo thành công và hiển thị nhân viên trong danh sách.</li> </ol>
<b>Hậu điều kiện</b>	Nhân viên mới có thể sử dụng tài khoản vừa tạo để đăng nhập vào hệ thống.
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>6.1. Tên đăng nhập đã tồn tại trong hệ thống:             <ol style="list-style-type: none"> <li>6.1.1. Hệ thống báo lỗi "Tên đăng nhập đã được sử dụng".</li> <li>6.1.2. Yêu cầu nhập tên đăng nhập khác.</li> </ol> </li> </ol>

**Bảng 2.4:** Đặc tả chức năng Thêm nhân viên

## 2.4 Yêu cầu phi chức năng

Đây là hệ thống quản lý bán hàng tốc độ cao phục vụ giờ cao điểm nên yêu cầu phi chức năng quan trọng nhất là thời gian phản hồi (tính từ lúc nhân viên thao tác trên màn hình chạm cho đến khi hệ thống ghi nhận và cập nhật trạng thái). Để đảm bảo giải tỏa nhanh chóng lượng khách trong khung giờ trưa (11h00 – 13h00), độ trễ của các thao tác cốt lõi như thêm món và xác nhận thanh toán phải nhỏ hơn 0,5 giây, đồng thời hệ thống phải hoạt động ổn định với tối thiểu 50 kết nối đồng thời mà không bị suy giảm hiệu năng. Giao diện POS cần được thiết kế tối ưu cho thao tác chạm với sự đồng nhất về bố cục, màu sắc định danh cho từng gói giá, kích thước nút bấm lớn và sử dụng ngôn ngữ tiếng Việt để nhân viên có thể thao tác chính xác mà không cần quan sát lâu. Ngoài ra, hệ thống cần đảm bảo tính toàn vẹn dữ liệu cho mọi giao dịch tài chính để tránh thất thoát doanh thu và sử dụng giao thức HTTPS để bảo mật luồng thông tin giữa các thiết bị.

## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 3.1 Java

Java [11] là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới hiện nay. Sau hơn hai thập kỷ tồn tại và phát triển, những đặc điểm hấp dẫn của nó cùng sự tiến hóa không ngừng đã giúp Java giữ vững vị trí quan trọng của mình trong cộng đồng lập trình viên toàn cầu. Có thể kể đến những điểm mạnh của Java như sau:

- (i) Java là nền tảng độc lập, hoạt động trên máy tính chạy hệ điều hành MacOS, Windows và Linux, từ đó mã nguồn viết bằng Java có thể dễ dàng triển khai cũng như bảo trì.
- (ii) Java là ngôn ngữ lập trình hướng đối tượng, hỗ trợ đầy đủ các tính chất của hướng đối tượng như trừu tượng hóa, đóng gói, kế thừa, đa hình.
- (iii) Quản lý bộ nhớ, dọn dẹp các ngăn nhớ không được sử dụng một cách tự động, ngăn chặn các phương thức tấn công liên quan đến bộ nhớ.

### 3.2 Spring Boot

Spring Boot [3] là framework của Java được giới thiệu vào năm 2012 và vẫn đang được phát triển từ đó đến nay. Một trong những điểm mạnh của Spring Boot là:

- (i) Hỗ trợ mô hình MVC giúp cho lập trình viên dễ dàng xây dựng và phát triển một ứng dụng web.
- (ii) Tomcat engine được cấu hình sẵn trong Spring Boot nên lập trình viên không cần phải cấu hình lại.
- (iii) Hỗ trợ tốt truy vấn cơ sở dữ liệu quan hệ như Mysql, Postgres, Oracle.

### 3.3 React

React [2] là một thư viện JavaScript mã nguồn mở được ra mắt lần đầu vào năm 2013. Đây là một trong những công cụ phổ biến nhất hiện nay để xây dựng giao diện người dùng, đặc biệt là các ứng dụng web tương tác cao.

Điểm cốt lõi làm nên sức mạnh của React là việc sử dụng mô hình Virtual DOM (DOM ảo). Thay vì thao tác trực tiếp lên DOM thực của trình duyệt - vốn là một tác vụ tốn kém về mặt hiệu năng, React tạo ra một bản sao của DOM trong bộ nhớ. Khi trạng thái của ứng dụng thay đổi, React sẽ so sánh DOM ảo mới với DOM ảo cũ để xác định chính xác những thành phần nào cần cập nhật, sau đó mới thực hiện thay đổi tối thiểu lên DOM thực. Cơ chế này giúp tối ưu hóa hiệu suất render, mang

lại trải nghiệm mượt mà cho người dùng.

Trong đồ án này, React được sử dụng để xây dựng ứng dụng theo kiến trúc Single Page Application (SPA). Khác với các ứng dụng web truyền thống phải tải lại toàn bộ trang mỗi khi người dùng chuyển hướng, SPA chỉ tải một trang HTML duy nhất ban đầu. Các tài nguyên như CSS, JavaScript cũng được tải một lần hoặc tải theo nhu cầu. Khi người dùng tương tác, JavaScript sẽ chặn các yêu cầu chuyển trang, thay vào đó nó sẽ gửi yêu cầu (thường là AJAX/Fetch) để lấy dữ liệu (thường là JSON) từ máy chủ và cập nhật lại nội dung giao diện một cách động. Kiến trúc này mang lại hai lợi ích lớn: Ứng dụng phản hồi gần như tức thì, không có hiện tượng "chớp" trang trắng khi chuyển đổi giữa các màn hình, tạo cảm giác giống như đang sử dụng một ứng dụng native trên máy tính; máy chủ không cần phải render giao diện (HTML) cho mỗi yêu cầu mà chỉ cần cung cấp dữ liệu thô qua API, giúp tiết kiệm băng thông và tài nguyên tính toán.

### 3.4 PostgreSQL

PostgreSQL [4] là một hệ thống cơ sở dữ liệu quan hệ đối tượng mã nguồn mở mạnh mẽ, sử dụng và mở rộng ngôn ngữ SQL kết hợp với nhiều tính năng giúp lưu trữ và chia tỷ lệ một cách an toàn các khối lượng công việc dữ liệu phức tạp nhất. Nguồn gốc của PostgreSQL có từ năm 1986 như một phần của dự án POSTGRES tại Đại học California ở Berkeley và đã có hơn 30 năm phát triển tích cực trên nền tảng cốt lõi.

PostgreSQL đã tạo được danh tiếng mạnh mẽ về kiến trúc đã được chứng minh, độ tin cậy, tính toàn vẹn của dữ liệu, bộ tính năng mạnh mẽ, khả năng mở rộng và sự công hiến của cộng đồng nguồn mở đãng sau phần mềm để liên tục cung cấp các giải pháp hiệu quả và sáng tạo. PostgreSQL chạy trên tất cả các hệ điều hành chính, đã tuân thủ ACID từ năm 2001 và có các tiện ích bổ sung mạnh mẽ như bộ mở rộng cơ sở dữ liệu không gian địa lý PostGIS phổ biến. Không có gì ngạc nhiên khi PostgreSQL đã trở thành cơ sở dữ liệu quan hệ nguồn mở được nhiều người và tổ chức lựa chọn.

#### 3.4.1 Redis

Để đảm bảo tốc độ phản hồi cực nhanh cho các thao tác bán hàng - yêu cầu tối quan trọng của hệ thống POS, đồ án sử dụng Redis [7] (Remote Dictionary Server) làm giải pháp caching (lưu trữ đệm). Redis là một hệ quản trị cơ sở dữ liệu in-memory (lưu trữ trong bộ nhớ RAM) mã nguồn mở, hỗ trợ cấu trúc dữ liệu dạng key-value với hiệu năng truy xuất cực cao (độ trễ thường dưới 1 mili-giây).

Trong hệ thống này, Redis được ứng dụng vào hai bài toán chính: các thông tin về thực đơn và giá bán là dữ liệu ít thay đổi nhưng được truy xuất rất nhiều lần

mỗi giây bởi các máy POS. Việc lưu trữ các dữ liệu này vào Redis giúp giảm tải đáng kể cho cơ sở dữ liệu chính (PostgreSQL) và tăng tốc độ hiển thị thực đơn lên màn hình POS gần như tức thì; Redis được dùng để lưu trữ thông tin phiên đăng nhập của nhân viên và khách hàng, đảm bảo tính nhất quán và khả năng mở rộng (scalability) khi hệ thống cần triển khai trên nhiều máy chủ (Clustering).

### 3.5 WebSocket

Đối với mô hình nhà hàng cơm bình dân, việc đồng bộ thông tin trạng thái món ăn giữa Bếp, Thu ngân và Khách hàng đặt online cần diễn ra theo thời gian thực. Để giải quyết vấn đề này, hệ thống sử dụng giao thức WebSocket [6].

Khác với giao thức HTTP truyền thống hoạt động theo cơ chế Request-Response (Client hỏi - Server trả lời), WebSocket cung cấp một kênh giao tiếp hai chiều (Full-duplex) liên tục giữa Client và Server qua một kết nối TCP duy nhất. Điều này cho phép Server chủ động đẩy dữ liệu xuống Client ngay khi có sự kiện xảy ra mà không cần Client phải gửi yêu cầu.

Ứng dụng cụ thể trong đồ án: khi nhân viên bếp cập nhật một món đã hết trên thiết bị của họ, Server sẽ ngay lập tức gửi tín hiệu qua WebSocket tới tất cả các máy POS và ứng dụng của khách hàng để làm mờ món đó, ngăn chặn việc nhận đơn hàng ảo; khi có đơn đặt hàng online, máy POS của thu ngân sẽ nhận được thông báo ngay lập tức kèm âm thanh cảnh báo mà không cần phải tải lại trang.

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm

Mô hình ba tầng (Three-tier architecture) là một kiến trúc ứng dụng phần mềm tổ chức các ứng dụng thành ba tầng tính toán logic và vật lý: tầng trình bày (presentation tier), tầng ứng dụng (application tier) và tầng dữ liệu (data tier). Lợi ích chính của kiến trúc ba tầng là mỗi tầng chạy trên một cơ sở hạ tầng riêng, được phát triển đồng thời bởi một nhóm phát triển riêng biệt và có thể được cập nhật hoặc mở rộng mà không ảnh hưởng đến các tầng khác.



**Hình 4.1:** Mô hình 3 tầng

##### a, Tầng trình bày

Tầng trình bày là tầng giao diện người dùng và giao tiếp của ứng dụng, nơi người dùng cuối (end-user) tương tác với ứng dụng. Mục đích chính của nó là hiển thị thông tin và thu thập thông tin từ người dùng. Tầng cấp cao nhất này có thể chạy trên trình duyệt web, dưới dạng ứng dụng dành cho máy tính để bàn hoặc giao diện người dùng đồ họa (GUI) chẳng hạn. Các tầng trình bày web thường được phát triển bằng HTML, CSS và JavaScript. Các ứng dụng máy tính để bàn có thể được viết bằng nhiều ngôn ngữ khác nhau tùy thuộc vào nền tảng.

##### b, Tầng ứng dụng

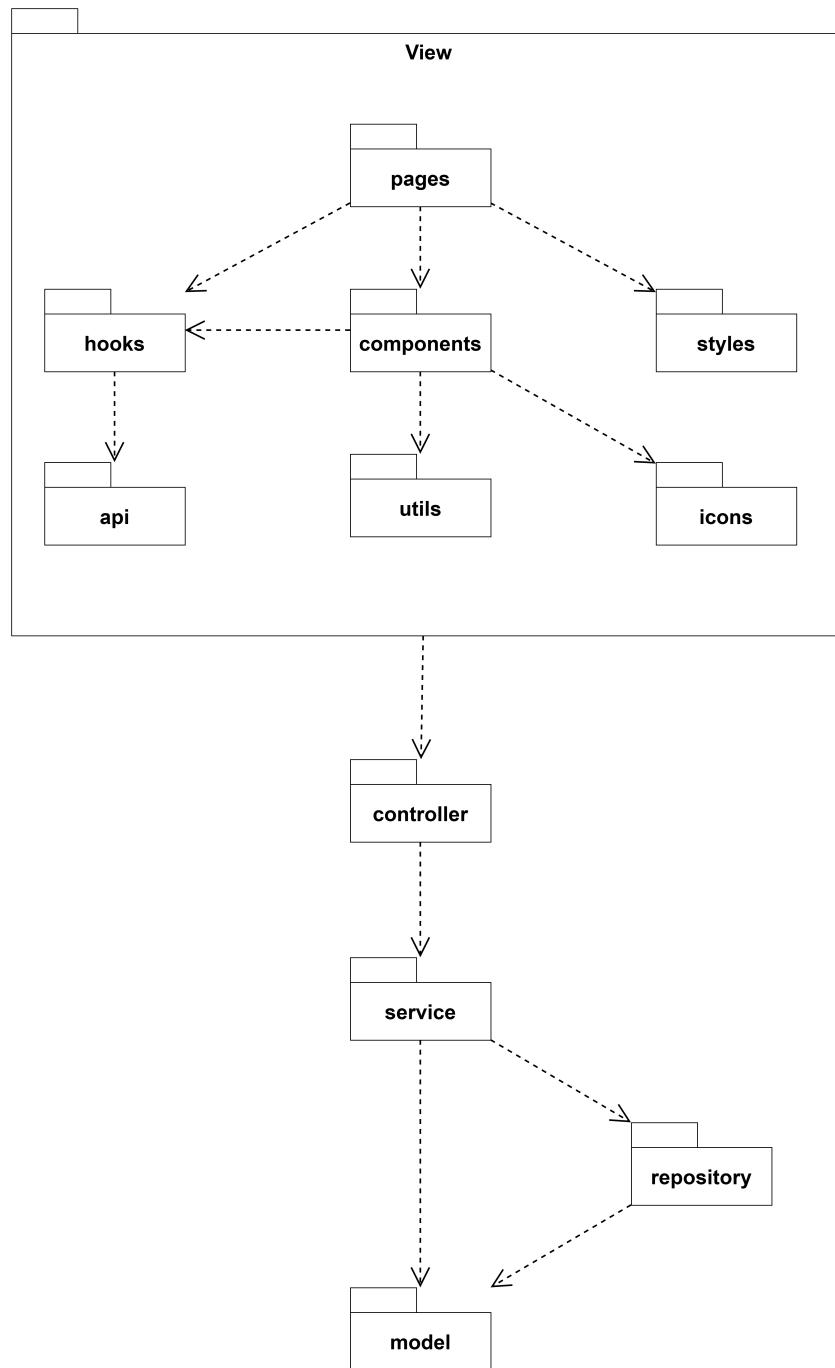
Tầng ứng dụng, còn được gọi là tầng logic hoặc tầng giữa, là trung tâm của hệ thống. Trong tầng này, thông tin được thu thập từ tầng trình bày được xử lý, đôi khi không giống hoặc không tồn tại trong những thông tin được lưu trong tầng dữ liệu. Bằng cách sử dụng logic nghiệp vụ, bộ quy tắc nghiệp vụ cụ thể, tầng ứng dụng có thể thêm, xóa hoặc sửa đổi dữ liệu trong tầng dữ liệu. Tầng ứng dụng thường được phát triển bằng Python, Java, PHP,...

##### c, Tầng dữ liệu

Tầng dữ liệu, đôi khi được gọi là tầng cơ sở dữ liệu, tầng truy cập dữ liệu hoặc tầng phụ, là nơi lưu trữ và quản lý thông tin được ứng dụng xử lý. Các hệ thống cơ sở dữ liệu phổ biến có thể kể đến như PostgreSQL, MySQL, MariaDB, DB2, Informix, Microsoft SQL Server hoặc MongoDB.

Tại hệ thống POS này, tầng trình bày được phát triển bằng HTML, CSS và JavaScript thông qua thư viện React, tầng ứng dụng được phát triển bằng Java và framework Spring boot, tầng dữ liệu sử dụng cơ sở dữ liệu PostgreSQL.

#### 4.1.2 Thiết kế tổng quan



Hình 4.2: Biểu đồ phụ thuộc gói

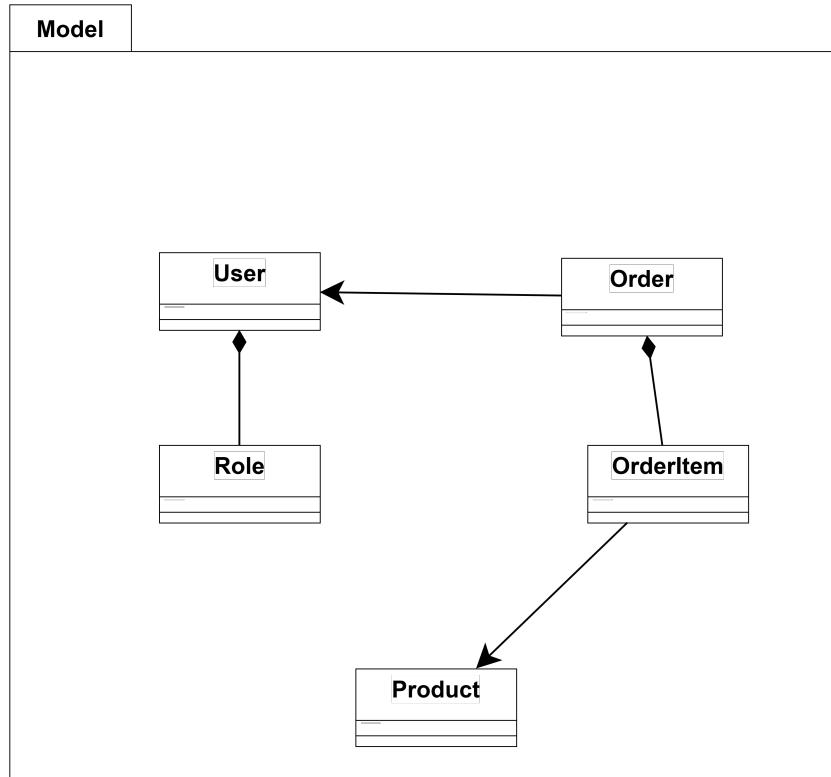
Hình 4.2 là biểu đồ phụ thuộc gói của hệ thống, với mục đích/nhiệm vụ của từng gói được nêu ở bảng 4.1.

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

<b>STT</b>	<b>Tên gói</b>	<b>Mục đích/Nhiệm vụ</b>
1	View	Gồm các thành phần tạo nên những giao diện trực quan giúp người dùng tương tác và sử dụng hệ thống. Trong gói View gồm các gói con pages, api, components, styles, hooks, utils, icons.
2	Pages	Là thiết kế giao diện của các màn hình trong hệ thống. Mỗi một class trong gói tương ứng với một màn hình của hệ thống.
3	Components	Là những thành phần nhỏ giúp hiển thị giao diện hoặc xử lý logic cấu thành nên một page.
4	Styles	Là tập hợp các tập tin giúp cải thiện chất lượng giao diện, làm đẹp hơn, đảm bảo các tính chất khi thiết kế giao diện và trải nghiệm người dùng trên các màn hình của hệ thống.
5	Hooks	Tập hợp các hook tự phát triển (Hook - được định nghĩa và giải thích trong phần 3.1.3).
6	Api	Thực hiện các yêu cầu xử lý, truy vấn dữ liệu của client đến server. Nhận phản hồi từ server và xử lý.
7	Utils	Tập hợp các tiện ích, các hằng số được sử dụng trong các component.
8	Icons	Tập hợp các icon được sử dụng trong các component.
9	Controller	Nhận các yêu cầu từ Client gửi đến, phân quyền, xác thực danh tính và gửi xuống Service để xử lý.
10	Service	Gồm các lớp xử lý logic các yêu cầu từ controller gửi xuống, sử dụng các phương thức được cung cấp bởi repository để truy vấn đến cơ sở dữ liệu.
11	Repository	Gồm các class cung cấp những phương thức thực hiện các câu truy vấn đến cơ sở dữ liệu.
12	Model	Tập hợp các lớp định nghĩa thông tin cơ bản của các đối tượng trong hệ thống, mỗi class tương ứng với một bảng trong database.

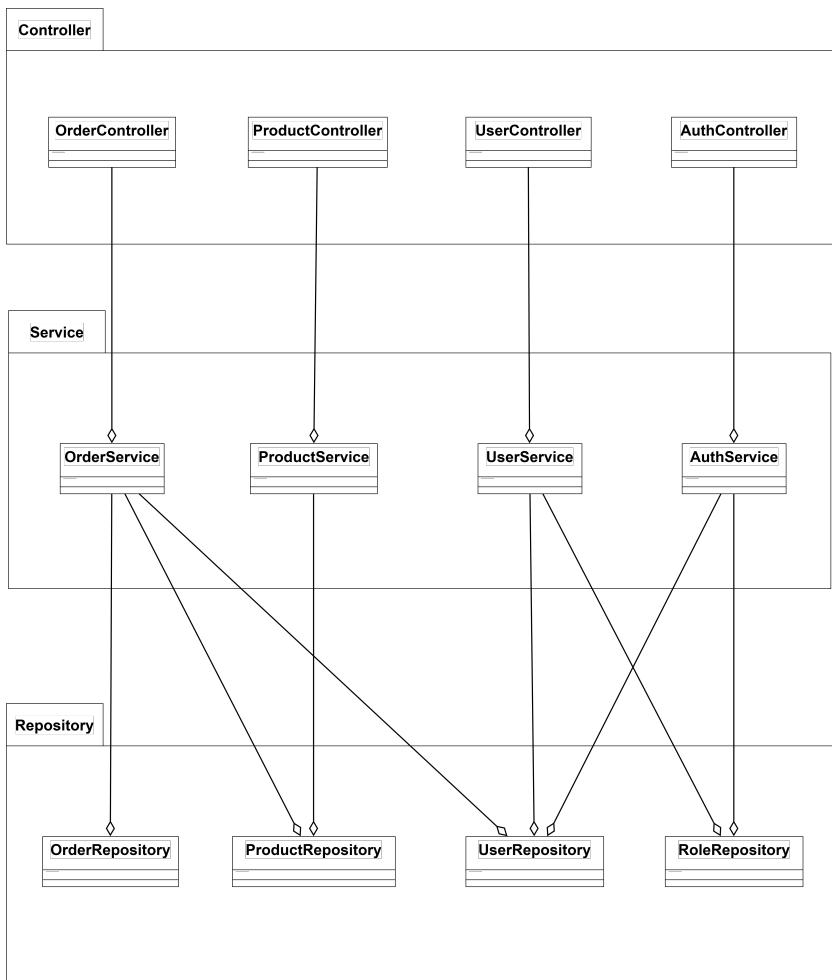
**Bảng 4.1:** Mục đích/Nhiệm vụ của các gói trong hệ thống

### 4.1.3 Thiết kế chi tiết gói



**Hình 4.3:** Biểu đồ thiết kế gói Model

Hình 4.3 là biểu đồ thiết kế cho gói Model. Gói Model tập hợp các lớp định nghĩa thông tin cơ bản của các đối tượng bao gồm User - người dùng, Order - đơn hàng, OrderItem - đơn hàng chi tiết, Product - sản phẩm, Role - vai trò của người dùng. Mỗi đối tượng User (như nhân viên thu ngân hoặc quản lý) có quan hệ kết tập với một hoặc nhiều Role để xác định quyền hạn trong hệ thống. Một User bao gồm thông tin về các Role mà họ đảm nhận. Mỗi Order (đơn hàng) được tạo ra bởi một User (thường là nhân viên thu ngân hoặc khách hàng đã đăng nhập), do đó Order có mối quan hệ liên kết với User để lưu vết người tạo đơn. Lớp Order có quan hệ hợp thành chèt chẽ với OrderItem. Một đơn hàng bao gồm danh sách các món ăn chi tiết OrderItem. Nếu đơn hàng bị xóa, các chi tiết món ăn đi kèm cũng không còn ý nghĩa tồn tại. Mỗi OrderItem tham chiếu đến một Product cụ thể để xác định đó là món ăn gì (ví dụ: Suất 30k, Trà đá).



Hình 4.4: Biểu đồ thiết kế gói Controller

Hình 4.4 là biểu đồ thiết kế cho gói Controller, Service, Repository. Các lớp trong gói Controller có nhiệm vụ là nhận yêu cầu từ client thông qua giao thức HTTP, sau đó chuyển xuống cho các lớp Service tương ứng của nó xử lý. Ví dụ lớp `OrderController` có một thuộc tính là một đối tượng `OrderService`, khi nhận được yêu cầu tạo đơn hàng nhanh từ client (máy POS), `OrderController` sẽ sử dụng các phương thức được cung cấp bởi `OrderService` để xử lý yêu cầu đó. Các lớp trong gói Service có nhiệm vụ xử lý logic nghiệp vụ. Lớp `ProductService` đảm nhận vai trò xử lý logic với những yêu cầu liên quan đến quản lý thực đơn và cập nhật trạng thái món ăn. Để thao tác với dữ liệu về món ăn trong cơ sở dữ liệu, nó sẽ sử dụng những phương thức được cung cấp bởi `ProductRepository`. Tương tự, `OrderService` đảm nhận vai trò xử lý logic nghiệp vụ phức tạp liên quan đến quy trình bán hàng và thanh toán. Để lưu trữ và thao tác với dữ liệu đơn hàng trong cơ sở dữ liệu, nó gọi đến những phương thức được cung cấp bởi `OrderRepository`. Ngoài ra, để đảm bảo tính toàn vẹn dữ liệu khi tạo đơn hàng (như lấy thông tin giá hiện tại của món ăn hay kiểm tra trạng thái tồn kho), `OrderService` sẽ gọi đến các phương thức mà `ProductRepository` (hoặc `ProductService`) cung cấp; đồng thời để ghi nhận nhân

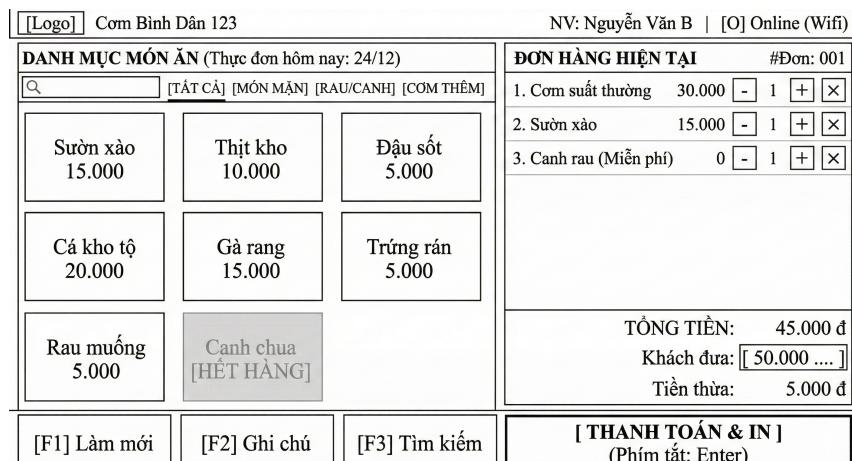
viên thực hiện giao dịch, nó cũng tương tác với UserRepository. Thiết kế này đảm bảo rằng, tất cả các lớp Controller sẽ chỉ có một nhiệm vụ duy nhất là nhận và điều phối yêu cầu từ client, việc xử lý logic nghiệp vụ sẽ do lớp Service của chính nó đảm nhận, và các đối tượng Repository chỉ cung cấp các phương thức truy xuất dữ liệu cơ bản. Để thực hiện các nghiệp vụ phức tạp liên quan đến nhiều thực thể, Service sẽ gọi tới Repository của chính nó hoặc sử dụng phương thức do các Service/Repository khác cung cấp.

## 4.2 Thiết kế chi tiết

### 4.2.1 Thiết kế giao diện

Hệ thống được thiết kế tối ưu cho máy tính để bàn và laptop có màn hình từ 14 inch trở lên với độ phân giải khuyến nghị 1440x900 pixels, nhằm đảm bảo không gian hiển thị rộng rãi cho các nghiệp vụ xử lý đơn hàng phức tạp. Về mặt thị giác, giao diện sử dụng màu xanh dương làm chủ đạo để tạo cảm giác chuyên nghiệp, với các nút bấm được chuẩn hóa đồng nhất (cao 36px, bo góc 4px, không viền) và cơ chế thông báo phản hồi trạng thái (thành công màu xanh, thất bại màu đỏ) đặt tại góc trái dưới màn hình. Các chức năng cốt lõi như Thực đơn điện tử, Giỏ hàng và đặc biệt là phần mềm POS bán hàng tại quầy đều tuân thủ nguyên tắc tối giản, chia bố cục rõ ràng thành các khu vực chức năng riêng biệt và hỗ trợ các thao tác nhanh (như phím tắt, gợi ý mệnh giá tiền) để đáp ứng áp lực phục vụ lớn trong giờ cao điểm.

Dưới đây là một số hình ảnh minh họa thiết kế giao diện của hệ thống:



**Hình 4.5:** Giao diện bán hàng chính

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

<a href="#">[Logo]</a> Com Bình Dân 123	Admin: Q.Lý A <a href="#">[Logout]</a>		
<a href="#">[1] Tổng quan</a> <a href="#">[2] QL Thực đơn</a> <a href="#">[3] QL Bán ăn</a> <a href="#">[4] QL Nhân viên</a> <a href="#">[5] Báo cáo Thống kê</a>	<b>Báo cáo và Thống kê Doanh thu</b>		
	Thời gian: <input type="button" value="Hôm nay"/> Từ ngày: [...] Đến ngày: [...] <a href="#">[ Xuất Báo cáo Excel ]</a>		
	<b>KẾT QUẢ KINH DOANH (Hôm nay)</b>		
	<b>DOANH THU</b> 5.450.000 đ	<b>SỐ ĐƠN HÀNG</b> 125 Đơn	<b>TRUNG BÌNH/ĐƠN</b> 43.600 đ
	<b>BIỂU ĐỒ DOANH THU THEO GIỜ (Giờ cao điểm)</b>		
<b>TOP MÓN ĂN BÁN CHẠY</b>			
1. Sườn xào chua ngọt (50 suất)  2. Trứng bắc cà chua (42 suất)  3. Cơm thêm (30 suất)			

**Hình 4.6:** Giao diện báo cáo thống kê

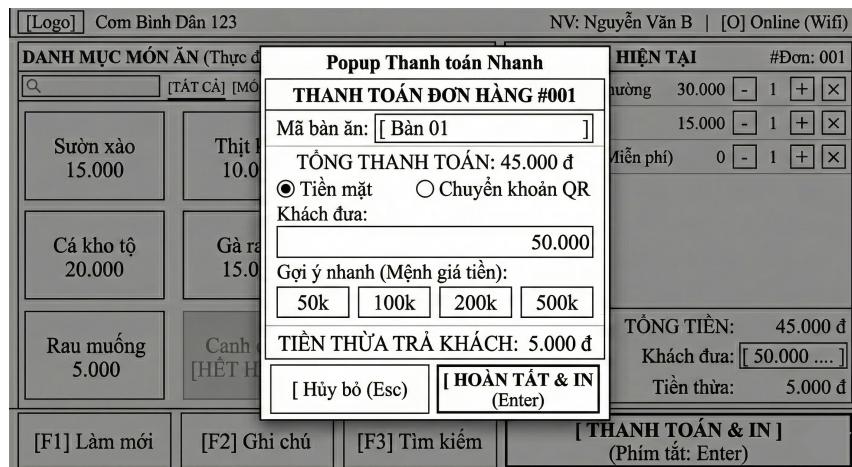
 Com Bình Dân 123	<input checked="" type="checkbox"/> BANNER QUẢNG CÁO / THÔNG BÁO HÔM NAY (Ví dụ: Hôm nay có sườn xào chua ngọt đặc biệt!)		
<a href="#">Trang chủ</a> <a href="#">Lịch sử đơn hàng</a> <a href="#">Tài khoản cá nhân</a> <a href="#">Giỏ hàng (0)</a> <a href="#">Thoát</a>	<b>Thực đơn ngày: 24/12/2025</b>		
	<a href="#">TẤT CẢ</a> <a href="#">MÓN MẶN</a> <a href="#">RAU/CANH</a> <a href="#">ĐỒ UỐNG</a>		
	<b>Món Nổi Bật / Combo</b>		
	<b>Sườn xào chua ngọt</b> 45.000 đ <a href="#">[ Nút Thêm (+) ]</a>	<b>Combo Cơm gà + Canh</b> 55.000 đ <a href="#">[ Nút Thêm (+) ]</a>	
	<b>Món Mặn (Đang sẵn hàng)</b>		
<b>Thịt kho tàu</b> 30.000 đ <a href="#">[ Nút Thêm (+) ]</a>			
(Danh sách cuộn xuống tiếp...)			

**Hình 4.7:** Giao diện đặt hàng

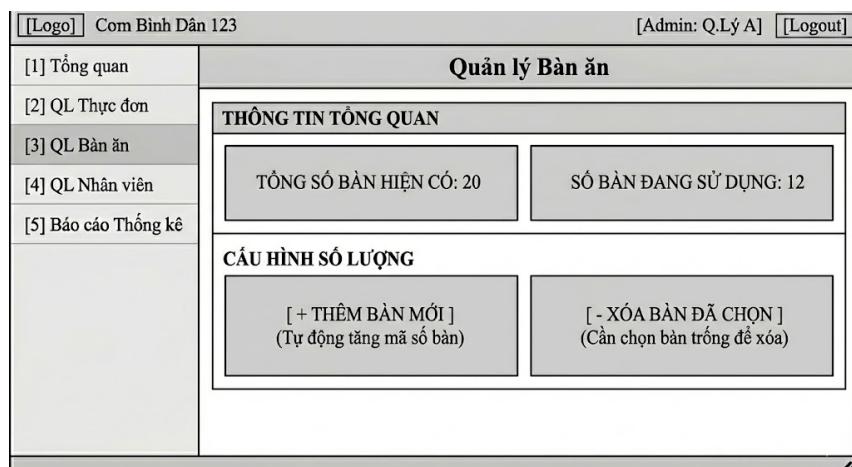
 Com Bình Dân 123	<input checked="" type="checkbox"/> BANNER QUẢNG CÁO / THÔNG BÁO HÔM NAY (Ví dụ: Hôm nay có sườn xào chua ngọt đặc biệt!)		
<a href="#">Trang chủ</a> <a href="#">Lịch sử đơn hàng</a> <a href="#">Tài khoản cá nhân</a> <a href="#">Giỏ hàng (0)</a> <a href="#">Thoát</a>	<b>Giỏ hàng của bạn (2 món)</b>		
	<b>Danh sách món đã chọn:</b>		
	<b>Sườn xào chua ngọt</b> 45.000 đ <a href="#">[ - ]</a> <a href="#">1</a> <a href="#">[ + ]</a> <a href="#">[ Xóa ]</a>		
	<b>Combo Cơm gà + Canh</b> 55.000 đ <a href="#">[ - ]</a> <a href="#">1</a> <a href="#">[ + ]</a> <a href="#">[ Xóa ]</a>		
	<b>Ghi chú cho quán:</b> <input type="text"/>		
<b>Thông tin đặt hàng:</b>			
Họ tên: <input type="text"/>		Số điện thoại: <input type="text"/>	
Thời gian lấy: <input type="text"/>		Tổng cộng tạm tính: 100.000 đ	
Hình thức: <input checked="" type="radio"/> Đến lấy (o) <input type="radio"/> Giao hàng ( )		<a href="#">[ ĐẶT HÀNG NGAY ]</a>	

**Hình 4.8:** Giao diện giỏ hàng

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



**Hình 4.9:** Giao diện popup thanh toán nhanh



**Hình 4.10:** Giao diện quản lý bàn ăn

QUẢN LÝ ĐƠN HÀNG TRONG NGÀY						
Lọc theo: [ Tất cả ] [ Tại quán ] [ Mang về ] [ Đặt trước (Web) ]						
Mã đơn	Giờ	Mã bàn ăn	Khách hàng	Tổng tiền	Trạng thái	Thao tác
#005	11:45	Bàn 01	Khách lẻ	45.000	Đã xong	[ In lại ]
#004	11:40	--	Nguyễn Văn A (098...)	60.000	Chờ lấy (Đã TT)	[ Chi tiết ] [ Giao đồ ]
#003	11:35	Bàn 03	Khách lẻ	30.000	Đã xong	[ In lại ]
#002	11:30	Bàn 02	Khách lẻ	50.000	Đang phục vụ	[ Chi tiết ]
Trang: < [1] 2 3 >						

**Hình 4.11:** Giao diện quản lý đơn hàng

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

<a href="#">[Logo]</a> Com Bình Dân 123	<a href="#">[Admin: Q.Lý A]</a> <a href="#">[Logout]</a>																														
<a href="#">[1] Tổng quan</a> <a href="#">[2] QL Thực đơn</a> <a href="#">[3] QL Bàn ăn</a> <a href="#">[4] QL Nhân viên</a> <a href="#">[5] Báo cáo Thống kê</a>	<b>Quản lý Nhân viên</b>																														
<input type="text" value="Ô tìm kiếm tên/SĐT..."/> <a href="#">[Lọc: Tất cả ✓]</a> <a href="#">[+ THÊM NHÂN VIÊN MỚI]</a>																															
<b>Danh sách nhân viên</b>																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Mã NV</th> <th>Họ và Tên</th> <th>Chức vụ</th> <th>Số điện thoại</th> <th>Trạng thái</th> <th>Thao tác</th> </tr> </thead> <tbody> <tr> <td>NV001</td> <td>Nguyễn Văn B</td> <td>Thu ngân</td> <td>0912...</td> <td>[O] Đang làm</td> <td>[Sửa][X]</td> </tr> <tr> <td>NV002</td> <td>Trần Thị C</td> <td>Bếp trưởng</td> <td>0988...</td> <td>[O] Đang làm</td> <td>[Sửa][X]</td> </tr> <tr> <td>NV003</td> <td>Lê Văn D</td> <td>Phục vụ</td> <td>0977...</td> <td>[ ] Nghỉ phép</td> <td>[Sửa][X]</td> </tr> <tr> <td colspan="6" style="text-align: center; font-size: small;">Trang: [1] [2] [3] &gt;</td> </tr> </tbody> </table>		Mã NV	Họ và Tên	Chức vụ	Số điện thoại	Trạng thái	Thao tác	NV001	Nguyễn Văn B	Thu ngân	0912...	[O] Đang làm	[Sửa][X]	NV002	Trần Thị C	Bếp trưởng	0988...	[O] Đang làm	[Sửa][X]	NV003	Lê Văn D	Phục vụ	0977...	[ ] Nghỉ phép	[Sửa][X]	Trang: [1] [2] [3] >					
Mã NV	Họ và Tên	Chức vụ	Số điện thoại	Trạng thái	Thao tác																										
NV001	Nguyễn Văn B	Thu ngân	0912...	[O] Đang làm	[Sửa][X]																										
NV002	Trần Thị C	Bếp trưởng	0988...	[O] Đang làm	[Sửa][X]																										
NV003	Lê Văn D	Phục vụ	0977...	[ ] Nghỉ phép	[Sửa][X]																										
Trang: [1] [2] [3] >																															

**Hình 4.12:** Giao diện quản lý nhân viên

<a href="#">[Logo]</a> Com Bình Dân 123	<a href="#">[Admin: Q.Lý A]</a> <a href="#">[Logout]</a>																								
<a href="#">[1] Tổng quan</a> <a href="#">[2] QL Thực đơn</a> <a href="#">[3] QL Bàn ăn</a> <a href="#">[4] QL Nhân viên</a> <a href="#">[5] Báo cáo Thống kê</a>	<b>Quản lý Thực đơn</b>																								
<input type="text" value="Tìm món ăn..."/> Lọc: <a href="#">[Tất cả]</a> <a href="#">[Đang bán]</a> <a href="#">[Hết hàng]</a> <a href="#">[+ THÊM MÓN MỚI]</a>																									
<b>Danh mục: Món Mặn</b>																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Hình ảnh</th> <th>Tên món ăn</th> <th>Giá bán</th> <th>Đơn vị</th> <th>Tình trạng</th> <th>Tác vụ</th> </tr> </thead> <tbody> <tr> <td><a href="#">[IMG]</a></td> <td>Sườn xào chua ngọt</td> <td>45.000 đ</td> <td>Đĩa</td> <td>[x] (Còn hàng)</td> <td>[Sửa][X]</td> </tr> <tr> <td><a href="#">[IMG]</a></td> <td>Thịt kho tàu</td> <td>30.000 đ</td> <td>Đĩa</td> <td>[x] (Còn hàng)</td> <td>[Sửa][X]</td> </tr> <tr> <td><a href="#">[IMG]</a></td> <td>Cá thu sốt cà</td> <td>40.000 đ</td> <td>Khúc</td> <td>[ ] (Hết hàng)</td> <td>[Sửa][X]</td> </tr> </tbody> </table>		Hình ảnh	Tên món ăn	Giá bán	Đơn vị	Tình trạng	Tác vụ	<a href="#">[IMG]</a>	Sườn xào chua ngọt	45.000 đ	Đĩa	[x] (Còn hàng)	[Sửa][X]	<a href="#">[IMG]</a>	Thịt kho tàu	30.000 đ	Đĩa	[x] (Còn hàng)	[Sửa][X]	<a href="#">[IMG]</a>	Cá thu sốt cà	40.000 đ	Khúc	[ ] (Hết hàng)	[Sửa][X]
Hình ảnh	Tên món ăn	Giá bán	Đơn vị	Tình trạng	Tác vụ																				
<a href="#">[IMG]</a>	Sườn xào chua ngọt	45.000 đ	Đĩa	[x] (Còn hàng)	[Sửa][X]																				
<a href="#">[IMG]</a>	Thịt kho tàu	30.000 đ	Đĩa	[x] (Còn hàng)	[Sửa][X]																				
<a href="#">[IMG]</a>	Cá thu sốt cà	40.000 đ	Khúc	[ ] (Hết hàng)	[Sửa][X]																				
<b>Danh mục: Canh / Rau</b>																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Hình ảnh</th> <th>Tên món ăn</th> <th>Giá bán</th> <th>Đơn vị</th> <th>Tình trạng</th> <th>Tác vụ</th> </tr> </thead> <tbody> <tr> <td><a href="#">[IMG]</a></td> <td>Canh rau ngọt</td> <td>10.000 đ</td> <td>Bát</td> <td>[x] (Còn hàng)</td> <td>[Sửa][X]</td> </tr> </tbody> </table>		Hình ảnh	Tên món ăn	Giá bán	Đơn vị	Tình trạng	Tác vụ	<a href="#">[IMG]</a>	Canh rau ngọt	10.000 đ	Bát	[x] (Còn hàng)	[Sửa][X]												
Hình ảnh	Tên món ăn	Giá bán	Đơn vị	Tình trạng	Tác vụ																				
<a href="#">[IMG]</a>	Canh rau ngọt	10.000 đ	Bát	[x] (Còn hàng)	[Sửa][X]																				

**Hình 4.13:** Giao diện quản lý thực đơn

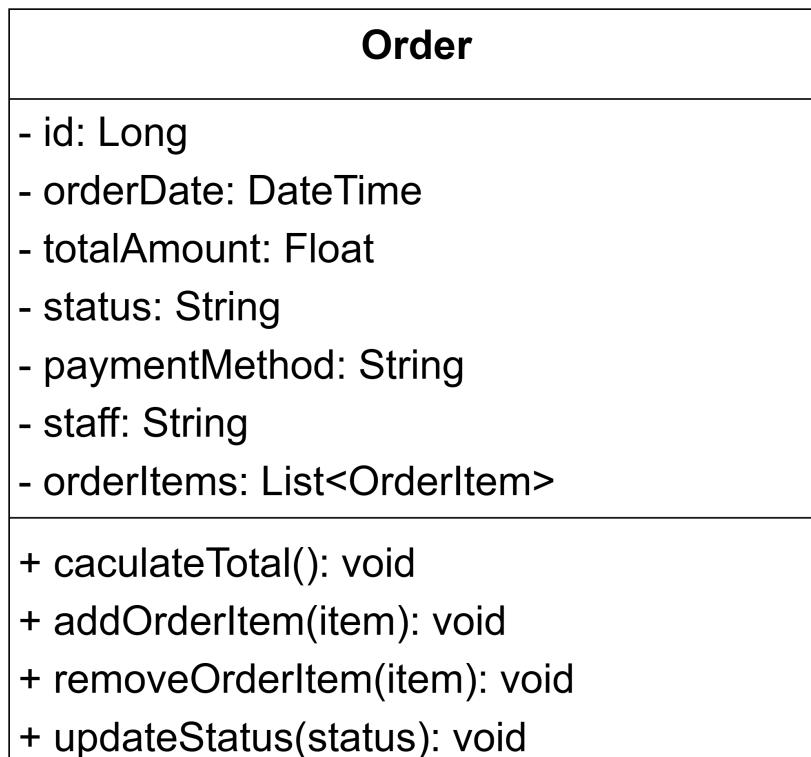
<a href="#">[Logo]</a> Com Bình Dân 123	<a href="#">[Admin: Q.Lý A]</a> <a href="#">[Logout]</a>
<a href="#">[1] Tổng quan</a> <a href="#">[2] QL Thực đơn</a> <a href="#">[3] QL Bàn ăn</a> <a href="#">[4] QL Nhân viên</a> <a href="#">[5] Báo cáo Thống kê</a>	<b>Tổng quan Hệ thống</b>
<div style="display: flex; justify-content: space-around; align-items: center;"> <span>Doanh thu hôm nay: 5.450.000 đ</span> <span>Số đơn hàng: 125</span> <span>Bàn đang sử dụng: 12/20</span> <span>Món hết hàng: 1</span> </div>	
<b>Truy cập nhanh</b>	
<div style="display: flex; justify-content: space-around; align-items: center;"> <a href="#">[+ Thêm Món Mới]</a> <a href="#">[+ Thêm Bàn Ăn]</a> <a href="#">[Xem Báo Cáo Ngày]</a> </div>	
<b>Hoạt động gần đây</b>	
<div style="font-size: small; margin-bottom: 0;">           12:15 - Đơn #125 đã thanh toán (55k)            12:12 - Bàn 05 đã có khách            12:10 - Món 'Canh chua' đã hết hàng         </div>	

**Hình 4.14:** Giao diện tổng quan

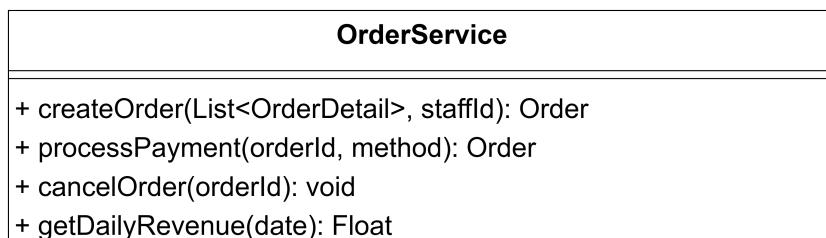


**Hình 4.15:** Giao diện trạng thái đơn hàng

#### 4.2.2 Thiết kế lớp



**Hình 4.16:** Thiết kế lớp Order

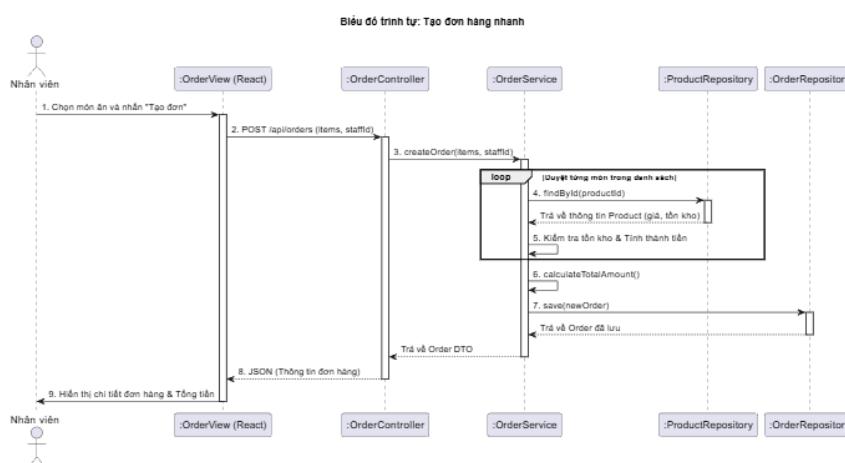


**Hình 4.17:** Thiết kế lớp OrderService

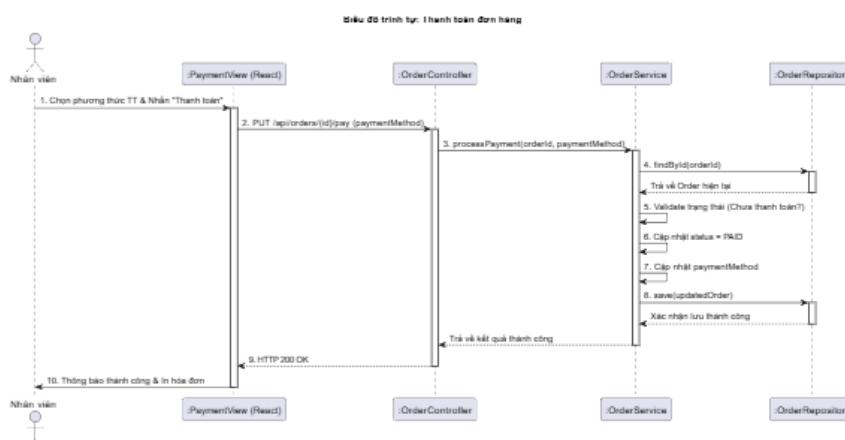
## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Product
<ul style="list-style-type: none"> <li>- id: Long</li> <li>- name: String</li> <li>- price: Float</li> <li>- imageUrl: String</li> <li>- isActive: Boolean</li> </ul>
<ul style="list-style-type: none"> <li>+ toggleActive(): void</li> <li>+ updatePrice(newPrice): void</li> <li>+ removeProduct(id): void</li> <li>+ addProduct(name, price, imageUrl): void</li> </ul>

Hình 4.18: Thiết kế lớp Product

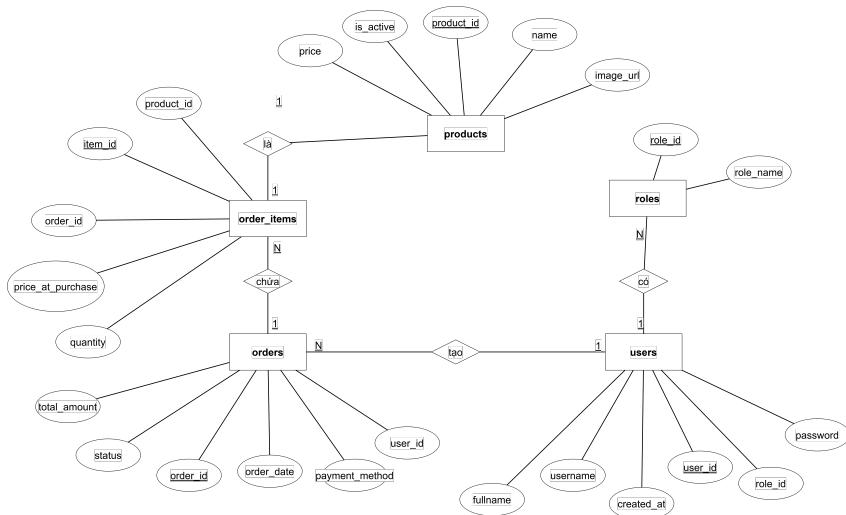


Hình 4.19: Biểu đồ trình tự Tạo đơn hàng nhanh



Hình 4.20: Biểu đồ trình tự Thanh toán

### 4.2.3 Thiết kế cơ sở dữ liệu



Hình 4.21: Biểu đồ thực thể liên kết

Hình 4.5 là biểu đồ thực thể liên kết của hệ thống, gồm các thực thể chính là người dùng, vai trò, sản phẩm và đơn hàng. Mỗi người dùng (`users`) có `user_id` riêng biệt, `username` và `password` dùng để đăng nhập và xác thực quyền truy cập vào hệ thống, `full_name` dùng để hiển thị tên nhân viên trên hóa đơn. Mỗi người dùng được gán một vai trò (`roles`) nhất định thông qua `role_id` để phân biệt quyền hạn quản trị hoặc nhân viên thu ngân.

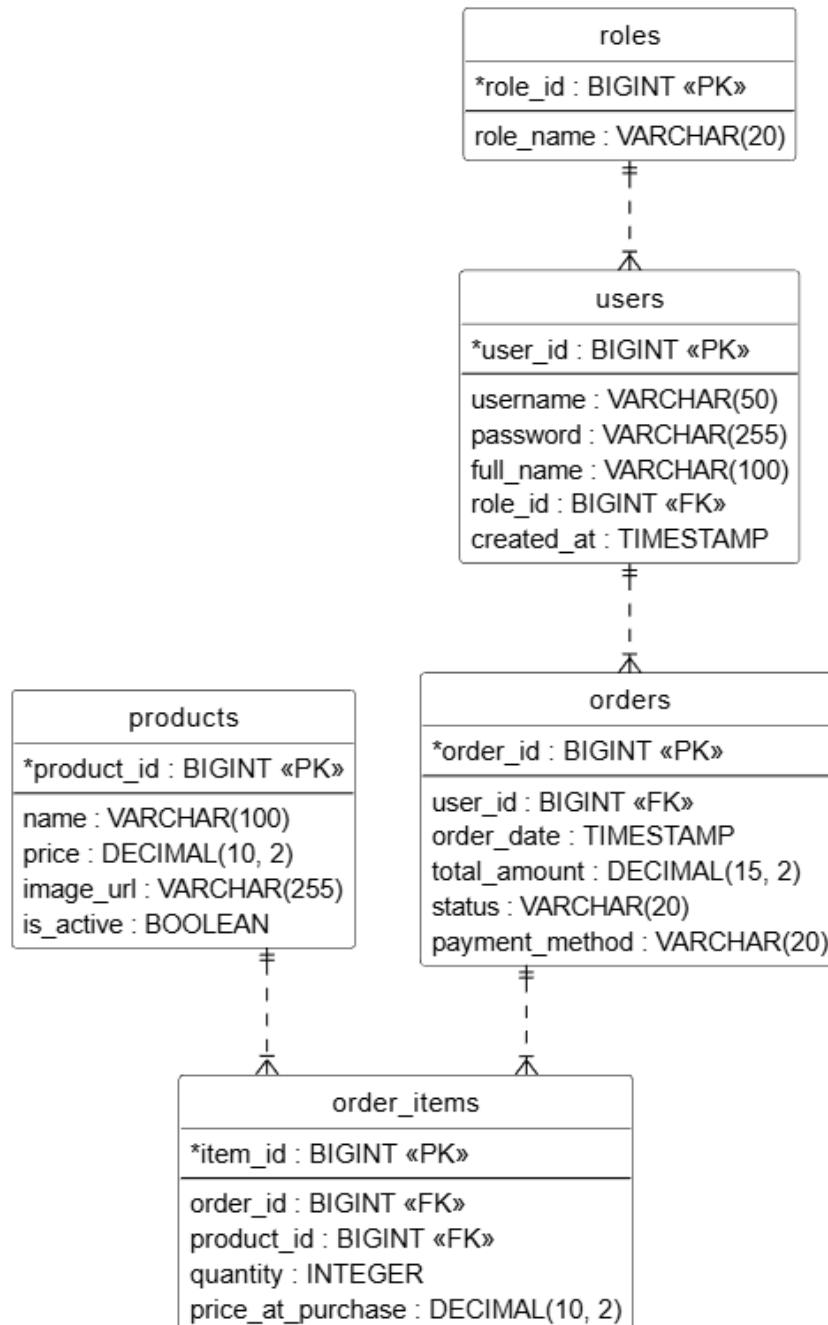
Thực thể sản phẩm (`products`) đại diện cho các món ăn trong thực đơn, bao gồm `product_id` để định danh, `name` là tên món ăn, `price` là giá bán hiện tại. Đặc biệt, hệ thống sử dụng trường `is_active` để quản lý việc hiển thị món ăn trên thực đơn theo ngày và `stock_quantity` để kiểm soát số lượng suất ăn còn lại trong kho, giúp nhân viên biết được món nào còn hay hết theo thời gian thực.

Về nghiệp vụ bán hàng, một đơn hàng (`orders`) bao gồm `order_id` riêng biệt để phân biệt các giao dịch, `total_amount` là tổng giá trị thanh toán, `status` thể hiện trạng thái đơn hàng (chờ thanh toán, đã thanh toán, hủy) và `payment_method` lưu phương thức thanh toán. Một người dùng (nhân viên) có thể tạo ra nhiều đơn hàng, nhưng một đơn hàng luôn được tạo bởi một và chỉ một người dùng cụ thể để phục vụ việc truy vết trách nhiệm.

Ngoài ra, một đơn hàng có thể bao gồm nhiều món ăn khác nhau, và một món ăn có thể xuất hiện trong nhiều đơn hàng. Vì thế, chúng tôi tạo thêm một thực thể trung gian là chi tiết đơn hàng (`order_items`). Mỗi chi tiết đơn hàng là một thể hiện cụ thể của một món ăn nằm trong một đơn hàng tương ứng. Chi tiết đơn hàng liên kết với đơn hàng qua `order_id` và món ăn qua `product_id`, lưu

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

trữ quantity là số lượng khách gọi. Quan trọng nhất, thực thể này có trường price\_at\_purchase dùng để lưu trữ giá bán tại thời điểm giao dịch, đảm bảo tính toàn vẹn của dữ liệu doanh thu lịch sử ngay cả khi giá gốc của sản phẩm (products) có sự thay đổi trong tương lai.



Hình 4.22: Biểu đồ thiết kế cơ sở dữ liệu

### 4.3 Thiết kế chi tiết cơ sở dữ liệu

### 4.4 Thiết kế chi tiết cơ sở dữ liệu

Dưới đây là mô tả chi tiết ý nghĩa các bảng và các trường dữ liệu được sử dụng trong hệ thống.

**Bảng 4.2:** Mô tả bảng Người dùng (users)

Tên trường	Kiểu dữ liệu	Ý nghĩa/Mô tả
user_id	BIGINT	Khóa chính (PK), mã định danh duy nhất của nhân viên trong hệ thống.
username	VARCHAR(50)	Tên đăng nhập dùng để xác thực danh tính khi truy cập hệ thống.
password	VARCHAR(255)	Mật khẩu đăng nhập (được lưu dưới dạng mã hóa BCrypt).
full_name	VARCHAR(100)	Họ và tên đầy đủ của nhân viên, dùng để hiển thị trên hóa đơn và giao diện.
role_id	BIGINT	Khóa ngoại (FK) liên kết với bảng roles, xác định quyền hạn của nhân viên.
created_at	TIMESTAMP	Thời gian tài khoản được tạo.

**Bảng 4.3:** Mô tả bảng Vai trò (roles)

Tên trường	Kiểu dữ liệu	Ý nghĩa/Mô tả
role_id	BIGINT	Khóa chính (PK), mã định danh của vai trò.
role_name	VARCHAR(20)	Tên vai trò (Ví dụ: ADMIN, STAFF), dùng để phân quyền chức năng.

**Bảng 4.4:** Mô tả bảng Sản phẩm/Món ăn (products)

Tên trường	Kiểu dữ liệu	Ý nghĩa/Mô tả
product_id	BIGINT	Khóa chính (PK), mã định danh duy nhất của món ăn.
name	VARCHAR(100)	Tên món ăn hiển thị trên thực đơn (Menu).
price	DECIMAL(10,2)	Giá bán hiện tại của món ăn.
image_url	VARCHAR(255)	Đường dẫn tới hình ảnh minh họa cho món ăn.
is_active	BOOLEAN	Trạng thái hiển thị. Nếu true, món ăn sẽ xuất hiện trên menu bán hàng hôm nay.
stock_quantity	INTEGER	Số lượng suất ăn ước tính còn lại trong kho, dùng để cảnh báo hết hàng.

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

**Bảng 4.5:** Mô tả bảng Đơn hàng (orders)

Tên trường	Kiểu dữ liệu	Ý nghĩa/Mô tả
order_id	BIGINT	Khóa chính (PK), mã định danh duy nhất của giao dịch/đơn hàng.
user_id	BIGINT	Khóa ngoại (FK), xác định nhân viên nào đã tạo đơn hàng này.
total_amount	DECIMAL(15,2)	Tổng giá trị thanh toán cuối cùng của đơn hàng.
status	VARCHAR(20)	Trạng thái đơn hàng (Ví dụ: PENDING, PAID, CANCELLED).
payment_method	VARCHAR(20)	Phương thức khách hàng dùng để thanh toán (Tiền mặt, Chuyển khoản).
order_date	TIMESTAMP	Thời gian chính xác khi đơn hàng được khởi tạo.

**Bảng 4.6:** Mô tả bảng Chi tiết đơn hàng (order\_items)

Tên trường	Kiểu dữ liệu	Ý nghĩa/Mô tả
item_id	BIGINT	Khóa chính (PK), định danh dòng chi tiết.
order_id	BIGINT	Khóa ngoại (FK), liên kết chi tiết này thuộc về đơn hàng nào.
product_id	BIGINT	Khóa ngoại (FK), xác định món ăn nào được gọi.
quantity	INTEGER	Số lượng món ăn khách hàng gọi trong đơn hàng này.
price_at_purchase	DECIMAL(10,2)	Giá bán của món ăn <b>tại thời điểm mua</b> . Lưu trữ giá trị này để đảm bảo lịch sử doanh thu không bị sai lệch khi giá gốc thay đổi.

## CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này tập trung trình bày chi tiết các giải pháp trọng tâm và những đóng góp nổi bật về mặt công nghệ cũng như quy trình nghiệp vụ mà đồ án đã đạt được. Xuất phát từ những thách thức đặc thù của mô hình kinh doanh nhà hàng cơm bình dân—nơi có áp lực thời gian phục vụ cực lớn trong khung giờ cao điểm và sự biến động liên tục của thực đơn hàng ngày—chúng tôi đã nghiên cứu và áp dụng các chiến lược thiết kế hệ thống chuyên biệt. Những giải pháp này không chỉ dừng lại ở việc đáp ứng các yêu cầu chức năng cơ bản mà còn đi sâu giải quyết các bài toán về tối ưu hóa hiệu năng, đảm bảo tính toàn vẹn dữ liệu tài chính và nâng cao trải nghiệm người dùng cuối. Các nội dung dưới đây sẽ phân tích sâu sắc cách thức hệ thống vượt qua các rào cản kỹ thuật để đạt được sự ổn định và hiệu quả trong môi trường vận hành thực tế.

### 5.1 Cơ chế quản lý thực đơn động và kiểm soát phiên bán hàng

Một trong những vấn đề nan giải nhất mà các chủ cửa hàng cơm bình dân gặp phải là sự thay đổi liên tục của thực đơn. Nguồn nguyên liệu phụ thuộc vào chợ đầu mối khiến danh sách món ăn mỗi ngày không cố định. Các hệ thống POS thông thường quản lý thực đơn theo một danh sách tĩnh (Static List), buộc người quản lý phải thực hiện rất nhiều thao tác thêm, sửa, xóa mỗi ngày để cập nhật tình trạng món. Điều này không chỉ tốn thời gian mà còn dễ gây ra sai sót, dẫn đến việc nhân viên thu ngân chọn nhầm món không có thực trong ngày hôm đó.

Để giải quyết vấn đề này, đồ án đã đề xuất kiến trúc quản lý thực đơn hai cấp độ: Kho dữ liệu tổng thể (Master Data) và Thực đơn theo phiên (Session Menu). Ở cấp độ thứ nhất, hệ thống lưu trữ toàn bộ các món ăn mà nhà hàng có thể nấu, bao gồm hình ảnh, công thức định lượng và giá tham chiếu. Ở cấp độ thứ hai, vào đầu mỗi ngày làm việc, người quản lý chỉ cần thực hiện thao tác “lên thực đơn” bằng cách chọn một tập hợp con từ kho dữ liệu tổng thể. Hệ thống hỗ trợ cơ chế sao chép thực đơn từ ngày hôm trước hoặc sử dụng các mẫu thực đơn có sẵn (Menu Templates) để giảm thiểu tối đa thời gian thao tác. Khi một phiên bán hàng được kích hoạt, chỉ những món ăn nằm trong thực đơn phiên đó mới được tải xuống máy POS của thu ngân. Giải pháp này giúp giao diện bán hàng luôn gọn gàng, loại bỏ hoàn toàn các món không phục vụ, giúp nhân viên không mất thời gian tìm kiếm và giảm thiểu rủi ro gọi món sai. Hơn thế nữa, hệ thống còn hỗ trợ tính năng cập nhật trạng thái “Hết hàng” (Out of Stock) theo thời gian thực. Ngay khi nhà bếp báo hết một món cụ thể, quản lý có thể cập nhật trạng thái này và hệ thống sẽ tự động vô hiệu hóa nút bấm tương ứng trên toàn bộ các thiết bị POS đang hoạt động, ngăn chặn triệt

để tình trạng bán quá số lượng thực tế.

### 5.2 Tái thiết kế quy trình bán hàng Quick Serve và giao diện chạm

Áp lực lớn nhất tại các quán cơm bình dân nằm ở khung giờ 11:30 đến 12:30, khi lưu lượng khách hàng tăng đột biến. Trong bối cảnh đó, mỗi giây thao tác thừa của thu ngân đều có thể dẫn đến ùn tắc cục bộ. Quy trình bán hàng truyền thống bao gồm các bước: Mở bàn, Chọn món, Gửi bếp, Chờ khách ăn xong, In tạm tính, và Thanh toán. Quy trình này quá cồng kềnh và không phù hợp với mô hình trả tiền trước hoặc thanh toán tại quầy của cơm bình dân.

Đồ án đã xây dựng quy trình bán hàng nhanh (Quick Serve Workflow) với triết lý “Three-Click-Action” (Thao tác trong ba lần chạm). Quy trình được rút gọn tối đa thành một luồng tuyến tính: Chọn món, Xác nhận, và Thanh toán. Để hiện thực hóa điều này, giao diện người dùng (UI) được thiết kế chuyên biệt cho màn hình cảm ứng (Touch-first Design). Thay vì sử dụng các danh sách dạng bảng (Table view) khó thao tác chính xác bằng ngón tay, hệ thống sử dụng lưới các nút bấm (Grid Layout) với kích thước lớn (tối thiểu 64x64 pixels). Màu sắc của các nút được mã hóa theo nhóm món ăn (ví dụ: món mặn màu đỏ, món rau màu xanh, đồ uống màu vàng) giúp nhân viên định vị món ăn theo phản xạ thị giác mà không cần đọc tên từng món. Các thao tác phụ trợ như thay đổi số lượng, thêm ghi chú cho bếp được thực hiện thông qua các cử chỉ vuốt hoặc nhấn giữ (Long press), giúp giữ cho màn hình chính luôn thoáng đãng.Thêm vào đó, hệ thống tích hợp sâu với máy in nhiệt. Ngay khi giao dịch được xác nhận thanh toán thành công, lệnh in được gửi trực tiếp xuống phần cứng mà không cần thông qua hộp thoại xác nhận in của trình duyệt, giúp khép kín quy trình phục vụ chỉ trong vòng dưới 30 giây cho mỗi lượt khách.

### 5.3 Kiến trúc phân tầng và bảo mật với Data Transfer Object

Hệ thống được xây dựng dựa trên nền tảng kiến trúc ba tầng (Three-tier Architecture) tiêu chuẩn, tuy nhiên đã được tinh chỉnh để tăng cường tính bảo mật và khả năng bảo trì. Sự phân tách rõ ràng giữa tầng Trình bày (Controller), tầng Nghiệp vụ (Service) và tầng Truy xuất dữ liệu (Repository) cho phép các thành phần phát triển độc lập và dễ dàng thay thế khi cần thiết. Điểm nhấn quan trọng trong kiến trúc này là việc áp dụng triệt để mẫu thiết kế Đối tượng chuyển dữ liệu (Data Transfer Object - DTO).

Trong các ứng dụng web thông thường, lập trình viên thường mắc lỗi trả về trực tiếp thực thể cơ sở dữ liệu (Entity) cho Client. Điều này tiềm ẩn rủi ro bảo mật nghiêm trọng khi để lộ cấu trúc bảng, các trường dữ liệu nhạy cảm (như mật khẩu, khóa bí mật) hoặc các thông tin nội bộ (như ngày tạo, người tạo). Đồ án giải quyết

vấn đề này bằng cách quy định nghiêm ngặt việc mọi dữ liệu đi ra hoặc đi vào API đều phải thông qua DTO. Các lớp DTO đóng vai trò như một lớp bộ lọc (Filter layer), chỉ chứa các trường dữ liệu thực sự cần thiết cho giao diện người dùng. Việc này không chỉ tăng cường bảo mật mà còn giúp tối ưu hóa bằng thông mạng bằng cách loại bỏ các dữ liệu dư thừa. Quá trình chuyển đổi (Mapping) giữa Entity và DTO được thực hiện tự động hóa bằng thư viện MapStruct, giúp giảm thiểu các đoạn mã lặp lại (Boilerplate code) và đảm bảo hiệu năng chuyển đổi gần như tương đương với mã viết tay nhờ cơ chế sinh mã tại thời điểm biên dịch (Compile-time).

### 5.4 Cơ chế Snapshot dữ liệu đảm bảo tính toàn vẹn tài chính

Một thách thức kỹ thuật quan trọng trong các hệ thống thương mại điện tử và bán lẻ là vấn đề lưu trữ lịch sử giá. Trong cơ sở dữ liệu quan hệ, nếu thiết kế không cẩn thận, việc cập nhật giá của một sản phẩm trong bảng danh mục (products) có thể vô tình làm thay đổi giá trị của tất cả các đơn hàng trong quá khứ tham chiếu đến sản phẩm đó. Điều này dẫn đến sự sai lệch nghiêm trọng trong báo cáo doanh thu và vi phạm các nguyên tắc kế toán.

Để giải quyết triệt để vấn đề này, đồ án đã triển khai kỹ thuật Snapshot dữ liệu (Data Snapshotting) hay còn gọi là Denormalization có chủ đích. Cụ thể, bảng `order_items` (chi tiết đơn hàng) được thiết kế không chỉ chứa khóa ngoại tham chiếu đến sản phẩm, mà còn chứa một bản sao độc lập của các thuộc tính quan trọng tại thời điểm giao dịch, bao gồm: tên món ăn và đơn giá (`price_at_purchase`). Khi một đơn hàng được khởi tạo, hệ thống thực hiện logic nghiệp vụ sao chép giá trị hiện hành từ bảng sản phẩm sang bảng chi tiết đơn hàng. Nhờ đó, mỗi dòng trong bảng `order_items` trở thành một bảng chứng lịch sử bất biến. Dù trong tương lai, sản phẩm có bị đổi tên, tăng giá, hay thậm chí bị xóa khỏi hệ thống (Soft delete), thì dữ liệu trong các đơn hàng cũ vẫn giữ nguyên giá trị ban đầu. Giải pháp này đảm bảo rằng khi truy xuất lại một hóa đơn từ 6 tháng trước, thông tin hiển thị hoàn toàn trùng khớp với thời điểm khách hàng thanh toán, đảm bảo tính chính xác tuyệt đối cho công tác đối soát và báo cáo tài chính.

### 5.5 Chiến lược Caching đa tầng với Redis

Với đặc thù hệ thống có tỷ lệ thao tác đọc dữ liệu (Read operation) chiếm áp đảo so với thao tác ghi (Write operation)—cụ thể là việc hàng loạt thiết bị POS liên tục tải danh sách thực đơn—việc truy vấn trực tiếp vào cơ sở dữ liệu quan hệ (RDBMS) như PostgreSQL sẽ tạo ra nút thắt cổ chai về hiệu năng (I/O Bottleneck). Để khắc phục, đồ án đã triển khai chiến lược Caching (Lưu trữ bộ nhớ đệm) sử dụng Redis.

Cơ chế Cache-aside (Lazy Loading) được áp dụng một cách linh hoạt. Khi có yêu cầu lấy thực đơn từ Client, hệ thống trước tiên sẽ kiểm tra trong Redis. Do

Redis lưu trữ dữ liệu trên RAM với cấu trúc Key-Value tối ưu, thời gian phản hồi thường chỉ ở mức dưới 5 mili-giây. Nếu dữ liệu không tồn tại trong Cache (Cache Miss), hệ thống mới thực hiện truy vấn xuống cơ sở dữ liệu, sau đó lưu kết quả vào Redis trước khi trả về cho Client. Để đảm bảo tính nhất quán dữ liệu (Data Consistency), đồ án thiết kế cơ chế làm mất hiệu lực Cache (Cache Invalidation) dựa trên sự kiện. Ngay khi người quản lý thực hiện bất kỳ thao tác thay đổi nào đối với thực đơn (như cập nhật giá, đổi trạng thái món ăn), một sự kiện sẽ được kích hoạt để xóa ngay lập tức các khóa tương ứng trong Redis. Điều này đảm bảo rằng dù sử dụng bộ nhớ đệm để tăng tốc độ, nhưng người dùng cuối luôn nhìn thấy dữ liệu mới nhất mà không gặp phải tình trạng hiển thị thông tin cũ (Stale data). Kết quả thực nghiệm cho thấy giải pháp này giúp giảm tải cho cơ sở dữ liệu chính tới 85% trong giờ cao điểm. [7]

### 5.5.1 Hệ thống giao tiếp thời gian thực dựa trên WebSocket

Để giải quyết bài toán đồng bộ thông tin tức thời giữa bộ phận thu ngân và bộ phận bếp, đồ án đã thay thế hoàn toàn cơ chế HTTP Polling truyền thống bằng giao thức WebSocket. Trong các hệ thống cũ, để biết có đơn hàng mới hay không, màn hình tại bếp phải liên tục gửi yêu cầu (Request) lên máy chủ sau mỗi vài giây. Điều này tạo ra một lượng lớn lưu lượng mạng dư thừa và độ trễ không thể tránh khỏi.

Đồ án sử dụng giao thức STOMP (Simple Text Oriented Messaging Protocol) chạy trên nền WebSocket để thiết lập kênh giao tiếp hai chiều liên tục (Full-duplex). Mô hình Publish-Subscribe được áp dụng: màn hình bếp đóng vai trò là Subscriber đăng ký nhận tin từ kênh chủ đề /topic/kitchen. Khi một đơn hàng được lưu thành công vào cơ sở dữ liệu, OrderService sẽ đóng vai trò là Publisher, đẩy một thông điệp chứa thông tin đơn hàng vào kênh này. Nhờ đó, thông tin đơn hàng xuất hiện trên màn hình bếp gần như ngay lập tức (Real-time) sau khi thu ngân bấm nút. Giải pháp này không chỉ tối ưu hóa tài nguyên máy chủ mà còn nâng cao đáng kể khả năng phối hợp nhịp nhàng giữa các bộ phận, một yếu tố sống còn đối với mô hình kinh doanh dịch vụ ăn uống. [6]

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Đồ án tốt nghiệp với đề tài “Xây dựng hệ thống quản lý vận hành và bán hàng cho nhà hàng cơm bình dân” đã hoàn thành trọn vẹn các mục tiêu đề ra ban đầu, bao gồm việc nghiên cứu cơ sở lý thuyết, phân tích đặc thù nghiệp vụ và xây dựng thành công một sản phẩm phần mềm hoàn chỉnh có khả năng ứng dụng thực tiễn. Quá trình thực hiện đồ án là sự kết hợp chặt chẽ giữa quy trình phát triển phần mềm chuẩn mực và việc áp dụng linh hoạt các công nghệ tiên tiến nhằm giải quyết các bài toán cụ thể của mô hình kinh doanh dịch vụ ăn uống bình dân.

Về mặt nghiệp vụ, hệ thống đã giải quyết triệt để bài toán quản lý thực đơn động biến thiên theo ngày và tối ưu hóa quy trình bán hàng tại quầy với tốc độ xử lý giao dịch dưới 30 giây. Việc chuyển đổi từ quy trình quản lý thủ công sang quy trình số hóa khép kín không chỉ giúp các chủ cửa hàng kiểm soát chặt chẽ nguồn thu, giảm thiểu thất thoát mà còn nâng cao đáng kể năng suất làm việc của nhân viên trong các khung giờ cao điểm. Giao diện người dùng được thiết kế chuyên biệt cho thao tác chạm đã chứng minh được tính hiệu quả trong việc giảm thiểu sai sót và rút ngắn thời gian đào tạo nhân viên mới.

Về mặt kỹ thuật, đồ án đã xây dựng được một kiến trúc hệ thống phân tầng bền vững, tuân thủ các nguyên tắc thiết kế hướng đối tượng và bảo mật dữ liệu. Việc tích hợp thành công các công nghệ như Redis cho tầng đệm dữ liệu và WebSocket cho giao tiếp thời gian thực đã đảm bảo hệ thống vận hành mượt mà với độ trễ thấp, đáp ứng tốt các yêu cầu phi chức năng về hiệu năng. Quan trọng hơn, cơ chế lưu trữ snapshot dữ liệu giá tại thời điểm giao dịch đã đảm bảo tính toàn vẹn và chính xác tuyệt đối cho các báo cáo tài chính, khắc phục hoàn toàn nhược điểm của các hệ thống quản lý bán hàng đơn giản trước đây. Kết quả thực nghiệm và kiểm thử cho thấy hệ thống hoạt động ổn định, chịu tải tốt và sẵn sàng cho việc triển khai vào môi trường thực tế.

### 6.2 Hạn chế và hướng phát triển

Mặc dù đã đạt được những kết quả khả quan, trong khuôn khổ giới hạn về thời gian và nguồn lực của một đồ án tốt nghiệp kỹ sư, hệ thống vẫn tồn tại một số hạn chế nhất định và mở ra nhiều dư địa cho việc nghiên cứu, phát triển trong tương lai. Hiện tại, hệ thống mới chỉ tập trung tối ưu hóa cho mô hình cửa hàng đơn lẻ và chưa hỗ trợ đầy đủ các tính năng quản lý chuỗi cung ứng phức tạp hay các báo cáo phân tích chuyên sâu đa chiều.

Hướng phát triển tiềm năng nhất trong tương lai là việc tích hợp các mô hình Trí tuệ nhân tạo (AI) và Học máy (Machine Learning) để giải quyết bài toán dự báo nhu cầu. Dựa trên dữ liệu lịch sử bán hàng được hệ thống thu thập, ta có thể xây dựng các mô hình dự đoán số lượng suất ăn tiêu thụ cho từng món trong ngày tiếp theo, kết hợp với các yếu tố ngoại cảnh như thời tiết hay ngày lễ. Điều này sẽ hỗ trợ đắc lực cho bộ phận bếp trong việc lên kế hoạch nhập nguyên liệu và sơ chế, giúp giảm thiểu tối đa lượng thức ăn thừa - một trong những vấn đề gây lãng phí lớn nhất của ngành kinh doanh ẩm thực. Đây là một bài toán tối ưu hóa phức tạp đòi hỏi năng lực tính toán cao và là tiền đề tốt cho các nghiên cứu chuyên sâu về dữ liệu lớn sau này.

Bên cạnh đó, về mặt kiến trúc phần mềm, hệ thống có thể được nâng cấp từ mô hình nguyên khối (Monolithic) sang kiến trúc vi dịch vụ (Microservices) để tăng cường khả năng mở rộng linh hoạt (Scalability). Việc tách nhỏ các mô-đun như Quản lý đơn hàng, Quản lý kho, và Báo cáo thành các dịch vụ độc lập sẽ cho phép triển khai hệ thống trên các nền tảng điện toán đám mây với công nghệ Containerization như Docker và Kubernetes. Đồng thời, việc phát triển thêm ứng dụng di động (Mobile App) dành riêng cho người quản lý để theo dõi doanh thu và điều hành cửa hàng từ xa cũng là một hướng đi thiết thực nhằm hoàn thiện hệ sinh thái phần mềm, mang lại trải nghiệm toàn diện và tiện ích hơn cho người sử dụng.

## TÀI LIỆU THAM KHẢO

- [1] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th. Pearson Prentice Hall, 2010.
- [2] *React*. [Online]. Available: <https://react.dev/>.
- [3] *Spring Boot*. [Online]. Available: <https://spring.io/projects/spring-boot>.
- [4] *PostgreSQL*. [Online]. Available: <https://www.postgresql.org/>.
- [5] R.T.Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, Irvine, 2000.
- [6] V. Pimentel and B. G. Nickerson, “Communicating and displaying real-time data with websocket,” *IEEE Internet Computing*, vol. 16, no. 4, pp. 45–53, 2012.
- [7] *Redis*. [Online]. Available: <https://redis.io/>.
- [8] *IPOS*. [Online]. Available: <https://ipos.vn/>.
- [9] *Cuk Cuk*. [Online]. Available: <https://www.cukcuk.vn/>.
- [10] *KiotViet*. [Online]. Available: <https://www.kiotviet.vn/>.
- [11] *Advantages java*. [Online]. Available: <https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java>.