

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH

**PHÁT TRIỂN PHẦN MỀM
QUẢN LÝ CỦA HÀNG BÁNH NGOT**

Chuyên ngành: Khoa học máy tính

GVHD: THS. MAI ĐỨC TRUNG

HỘI ĐỒNG: 2 CN Đồ án chuyên ngành

—o0o—

SVTH: Nguyễn Khánh Huy (2052496)

TP. HỒ CHÍ MINH, THÁNG 12 /2025

Chữ ký của GVHD

Ngày: 22/12/2025

ThS. Mai Đức Trung

Khoa Khoa học và Kỹ thuật máy tính

Lời cam đoan

Tác giả Nguyễn Khánh Huy, xin cam đoan rằng đồ án tốt nghiệp và báo cáo đồ án tốt nghiệp được thực hiện bởi chính bản thân, dưới sự hướng dẫn và chỉ bảo của ThS. Mai Đức Trung thuộc Khoa Khoa học và Kỹ thuật Máy tính, Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh.

Trong lúc thực hiện đồ án, tác giả có tham khảo nhiều trang web, nguồn tài liệu và các bài nghiên cứu khác nhau từ các tác giả trong và ngoài nước. Những số liệu trong các bảng biểu và hình ảnh phục vụ cho việc phân tích, nhận xét, đánh giá được thu thập từ các nguồn khác nhau có ghi trong phần tài liệu tham khảo. Toàn bộ những tài liệu được tham khảo đều được ghi rõ ở phần *Danh mục tài liệu tham khảo*.

Nếu phát hiện có bất kỳ sự gian lận hoặc tương đồng nào, tác giả xin hoàn toàn chịu trách nhiệm trước Hội đồng, cũng như nhận hoàn toàn trách nhiệm về kết quả đồ án cuối cùng.

Tp. Hồ Chí Minh, ngày 22 tháng 12 năm 2025

*Tác giả
Nguyễn Khánh Huy*

Lời cảm ơn

Trong suốt quá trình nghiên cứu và thực hiện đồ án tốt nghiệp này, tác giả luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình từ các thầy cô trong Khoa Khoa học và Kỹ thuật Máy tính, cùng với sự động viên từ các bạn cùng lớp.

Tác giả xin bày tỏ lòng biết ơn sâu sắc đến Ban giám hiệu Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh, các thầy cô, giảng viên của Trường Đại học Bách khoa nói chung, và các thầy cô, giảng viên trực thuộc Khoa Khoa học và Kỹ thuật Máy tính nói riêng đã tận tình giúp đỡ và chỉ bảo trong quá trình thực hiện đồ án tốt nghiệp này. Với sự tâm huyết và nhiệt tình của mình trong công việc giảng dạy, thầy cô đã truyền tải đầy đủ những kiến thức, tạo nên nền móng vững chắc cho tác giả có thể hoàn thành đề tài này.

Ngoài ra, tác giả cũng xin bày tỏ lòng biết ơn chân thành tới ThS. Mai Đức Trung, người đã dấn dắt trong suốt những tuần qua. Nhờ có sự chỉ bảo tận tình, cách làm việc và phân công khoa học, giải thích cặn kẽ của thầy mỗi khi gặp khó khăn, những lời chỉ dẫn và động viên của thầy đã tiếp thêm động lực, giúp tác giả trở nên tự tin hơn và chủ động hơn trong thời gian hoàn thành đề tài đồ án này. Nếu thiếu đi những sự chỉ bảo đó, chắc chắn rằng đề tài này sẽ không được xây dựng một cách hoàn thiện như ngày hôm nay.

Cuối cùng, tác giả cảm ơn những người bạn đã đồng hành trong hành trình cùng phần đầu trong những năm đại học vừa qua. Với những kỹ năng và điểm mạnh khác nhau, mọi người đã cùng nhau giúp đỡ, chỉ bảo vượt qua khó khăn trong suốt quá trình theo học. Sự chân thành đó đã góp phần không nhỏ vào sự thành công của đề tài này.

Tác giả xin chân thành cảm ơn.

Tp. Hồ Chí Minh, ngày 22 tháng 12 năm 2025

Tác giả

Nguyễn Khánh Huy

Tóm tắt đề tài

Ngành kinh doanh bánh ngọt đòi hỏi khả năng quản lý đồng bộ từ khâu nguyên liệu, công thức chế biến, sản xuất theo ca, cho đến bán hàng trực tiếp và đặt hàng trực tuyến. Phần lớn các cửa hàng nhỏ hiện vẫn quản lý thủ công bằng sổ sách hoặc bảng tính rời rạc, dễ dẫn đến sai lệch tồn kho, thất thoát nguyên liệu, khó kiểm soát hiệu suất nhân viên và thiếu cơ sở dữ liệu chính xác để phân tích doanh thu – chi phí.

Đề tài tập trung nghiên cứu, phân tích và phát triển hệ thống phần mềm quản lý tiệm bánh ngọt tích hợp các chức năng chính: quản lý sản phẩm và nguyên liệu; quản lý đơn hàng, hóa đơn và thanh toán POS; theo dõi nhập – xuất – tồn kho; hỗ trợ đặt bánh trực tuyến; quản lý nhà cung cấp và đơn mua; quản lý khách hàng thân thiết, khuyến mãi; cùng các báo cáo thống kê doanh thu – tồn kho – hiệu suất hoạt động theo thời gian thực.

Hệ thống được thiết kế với giao diện thân thiện (Vue.js + TailwindCSS), kiến trúc backend gọn nhẹ (Express.js + PostgreSQL) nhằm đảm bảo hiệu năng, bảo mật, và khả năng vận hành thực tế cho cả front-of-house (bán hàng) và back-of-house (sản xuất, kho). Giải pháp có khả năng mở rộng trong tương lai để hỗ trợ đa chi nhánh, tích hợp thanh toán điện tử và phân tích dữ liệu nâng cao phục vụ quản trị.

Mục lục

| | |
|---|------|
| Chữ ký của GVHD | i |
| Lời cam đoan | ii |
| Lời cảm ơn | iii |
| Tóm tắt đề tài | iv |
| Mục lục | v |
| Danh sách hình vẽ | viii |
| 1 Tổng quan | 1 |
| 1.1 Giới thiệu đề tài | 1 |
| 1.2 Mục tiêu đề tài | 2 |
| 1.3 Khảo sát các giải pháp hiện có | 2 |
| 1.3.1 Quản lý thủ công (Excel và Sổ sách) | 2 |
| 1.3.2 Phần mềm quản lý bán hàng thương mại (KiotViet) | 3 |
| 1.3.3 So sánh và đề xuất giải pháp | 4 |
| 1.4 Bố cục bài báo cáo | 5 |
| 2 Công nghệ và phần mềm sử dụng | 7 |
| 2.1 Công nghệ ở Frontend | 7 |
| 2.1.1 Vue.js | 7 |
| 2.1.2 TailwindCSS | 9 |
| 2.2 Công nghệ ở Backend | 11 |
| 2.2.1 Express.js | 11 |
| 2.2.2 PostgreSQL | 13 |
| 3 Phân tích thiết kế hệ thống | 15 |

Mục lục

| | | |
|--------|--|----|
| 3.1 | Các Actor có trong hệ thống | 15 |
| 3.2 | Phân tích nhu cầu (User Stories) | 16 |
| 3.3 | Yêu cầu chức năng (Functional Requirements) | 17 |
| 3.4 | Yêu cầu phi chức năng (Non-Functional Requirements) | 18 |
| 3.4.1 | Hiệu năng (Performance) | 18 |
| 3.4.2 | Bảo mật (Security) | 18 |
| 3.4.3 | Độ tin cậy và Khả dụng (Reliability & Availability) | 18 |
| 3.4.4 | Tương thích và Môi trường (Compatibility) | 19 |
| 3.5 | Thiết kế tổng quan hệ thống | 20 |
| 3.6 | Use Case Diagram | 22 |
| 3.7 | Activity Diagram | 24 |
| 3.7.1 | UC-01: Đăng nhập (User Login) | 24 |
| 3.7.2 | UC-02: Đăng ký (User Register) | 26 |
| 3.7.3 | UC-03: Duyệt Menu (Browse Menu) | 28 |
| 3.7.4 | UC-04: Đặt hàng (Place Order) | 30 |
| 3.7.5 | UC-05: Xem lịch sử đơn hàng (View Order History) | 32 |
| 3.7.6 | UC-06: Nhận phiếu order (Receive Order Ticket) | 33 |
| 3.7.7 | UC-07: Cập nhật trạng thái đơn hàng (Update Order Status) | 34 |
| 3.7.8 | UC-08: Quản lý tồn kho (Manage Inventory) | 36 |
| 3.7.9 | UC-09: Tạo báo cáo (Generate Reports) | 37 |
| 3.7.10 | UC-10: Quản lý tài khoản người dùng (Manage User Accounts) | 39 |
| 3.8 | Sequence Diagram | 41 |
| 3.8.1 | UC-01: Đăng nhập (User Login) | 41 |
| 3.8.2 | UC-02: Đăng ký (User Register) | 43 |
| 3.8.3 | UC-03: Duyệt Menu (Browse Menu) | 44 |
| 3.8.4 | UC-04: Đặt hàng (Place Order) | 45 |
| 3.8.5 | UC-05: Xem lịch sử đơn hàng (View Order History) | 46 |
| 3.8.6 | UC-06: Nhận phiếu order (Receive Order Ticket) | 47 |
| 3.8.7 | UC-07: Cập nhật trạng thái đơn hàng (Update Order Status) | 48 |
| 3.8.8 | UC-08: Quản lý tồn kho (Manage Inventory) | 49 |
| 3.8.9 | UC-09: Tạo báo cáo (Generate Reports) | 50 |
| 3.8.10 | UC-10: Quản lý tài khoản (Manage User Accounts) | 51 |
| 3.9 | Class Diagram | 52 |
| 3.10 | Data Flow Diagram (DFD) | 53 |
| 3.11 | ERD (Entity Relationship Diagram) | 55 |
| 3.12 | Component Diagram | 57 |

| | |
|--|-----------|
| 3.13 Deployment Diagram | 58 |
| 4 Kết quả thiết kế giao diện | 59 |
| 4.1 Phân hệ Khách hàng (Public & Customer Portal) | 59 |
| 4.1.1 Trang chủ và Giới thiệu (Landing Page) | 60 |
| 4.1.2 Trang danh sách sản phẩm và Chi tiết | 62 |
| 4.2 Phân hệ Xác thực và Tài khoản (Authentication) | 64 |
| 4.2.1 Đăng nhập (Login) | 64 |
| 4.2.2 Đăng ký tài khoản mới (Register) | 65 |
| 4.3 Phân hệ Quản trị (Admin Dashboard) | 66 |
| 4.3.1 Tổng quan hệ thống (Dashboard) | 66 |
| 4.3.2 Quản lý Đơn hàng và Người dùng | 67 |
| 4.3.3 Quản lý Kho và Sản phẩm | 69 |
| 4.3.4 Báo cáo thông kê và Thông báo | 71 |
| 5 Định hướng phát triển | 73 |
| 5.1 Kết luận | 73 |
| 5.2 Kế hoạch hiện thực | 73 |
| Danh mục tài liệu tham khảo | 75 |

Danh sách hình vẽ

| | | |
|--------|---|----|
| 2.1.1 | Logo Vue.js | 7 |
| 2.1.2 | Logo TailwindCSS | 9 |
| 2.2.1 | Logo Express.js | 11 |
| 2.2.2 | Logo PostgreSQL | 13 |
| 3.5.1 | Sơ đồ kiến trúc tổng thể và các module chức năng của hệ thống quản lý tiệm bánh | 20 |
| 3.6.1 | Biểu đồ Use Case của hệ thống | 23 |
| 3.7.1 | Biểu đồ hoạt động chức năng Đăng ký | 26 |
| 3.7.2 | Biểu đồ hoạt động chức năng Duyệt Menu | 28 |
| 3.7.3 | Biểu đồ hoạt động chức năng Đặt hàng | 30 |
| 3.7.4 | Biểu đồ hoạt động chức năng Xem lịch sử đơn hàng | 32 |
| 3.7.5 | Biểu đồ hoạt động chức năng Nhận phiếu order | 33 |
| 3.7.6 | Biểu đồ hoạt động chức năng Cập nhật trạng thái đơn hàng | 34 |
| 3.7.7 | Biểu đồ hoạt động chức năng Quản lý tồn kho | 36 |
| 3.7.8 | Biểu đồ hoạt động chức năng Tạo báo cáo | 37 |
| 3.7.9 | Biểu đồ hoạt động chức năng Quản lý tài khoản | 39 |
| 3.8.1 | Sequence Diagram - Đăng nhập | 41 |
| 3.8.2 | Sequence Diagram - Đăng ký tài khoản | 43 |
| 3.8.3 | Sequence Diagram - Duyệt Menu | 44 |
| 3.8.4 | Sequence Diagram - Đặt hàng | 45 |
| 3.8.5 | Sequence Diagram - Xem lịch sử đơn hàng | 46 |
| 3.8.6 | Sequence Diagram - Nhận phiếu order | 47 |
| 3.8.7 | Sequence Diagram - Cập nhật trạng thái đơn hàng | 48 |
| 3.8.8 | Sequence Diagram - Quản lý tồn kho | 49 |
| 3.8.9 | Sequence Diagram - Tạo báo cáo | 50 |
| 3.8.10 | Sequence Diagram - Quản lý tài khoản người dùng | 51 |
| 3.9.1 | Biểu đồ lớp (Class Diagram) của hệ thống | 52 |
| 3.10.1 | DFD cấp 1 của hệ thống | 53 |

Danh sách hình vẽ

| | |
|---|----|
| 3.11.1 Sơ đồ ERD của cơ sở dữ liệu | 56 |
| 3.12.1 Component Diagram của hệ thống | 57 |
| 3.13.1 Deployment Diagram – Mô hình triển khai hệ thống | 58 |
| 4.1.1 Giao diện trang chủ với thông điệp chào mừng và hình ảnh chủ đạo | 60 |
| 4.1.2 Danh sách các sản phẩm nổi bật trong ngày | 61 |
| 4.1.3 Phần giới thiệu cam kết chất lượng và thông tin liên hệ nhanh | 61 |
| 4.1.4 Giao diện danh sách sản phẩm dành cho khách hàng | 62 |
| 4.1.5 Cửa sổ chi tiết sản phẩm hiển thị thành phần và cảnh báo dị ứng | 63 |
| 4.2.1 Màn hình đăng nhập với giao diện tối giản và thông tin demo | 64 |
| 4.2.2 Giao diện đăng ký thành viên mới | 65 |
| 4.3.1 Màn hình Dashboard quản trị với các chỉ số kinh doanh và cảnh báo tồn kho . . . | 66 |
| 4.3.2 Giao diện quản lý danh sách đơn hàng | 67 |
| 4.3.3 Giao diện quản lý tài khoản và phân quyền người dùng | 68 |
| 4.3.4 Bảng quản lý nguyên liệu và trạng thái tồn kho | 69 |
| 4.3.5 Giao diện chỉnh sửa thông tin sản phẩm và giá vốn | 70 |
| 4.3.6 Giao diện báo cáo phân tích doanh thu và xu hướng bán hàng | 71 |
| 4.3.7 Trung tâm thông báo và cảnh báo hệ thống | 72 |
| 5.2.1 Biểu đồ Gantt kế hoạch triển khai dự án | 74 |

Chương 1

Tổng quan

1.1 Giới thiệu đề tài

Trong bối cảnh hiện nay, ngành bán lẻ thực phẩm, đặc biệt là kinh doanh bánh ngọt, đang chứng kiến tốc độ tăng trưởng nhanh chóng cùng sự cạnh tranh ngày càng khốc liệt. Đặc thù của mô hình này không chỉ dừng lại ở nghiệp vụ bán hàng tại quầy mà còn đòi hỏi sự quản lý chặt chẽ về định lượng nguyên liệu, công thức chế biến (Recipe), quy trình sản xuất và dịch vụ giao hàng đa kênh. Tuy nhiên, phần lớn các cửa hàng quy mô vừa và nhỏ hiện vẫn duy trì phương thức quản lý thủ công qua sổ sách hoặc các bảng tính rời rạc. Hạn chế này thường dẫn đến sai lệch số liệu, khó kiểm soát nhập xuất tồn, thiếu dữ liệu phân tích kinh doanh và ảnh hưởng trực tiếp đến trải nghiệm khách hàng cũng như khả năng mở rộng quy mô.

Xuất phát từ nhu cầu cấp thiết đó, đề tài tập trung nghiên cứu và xây dựng hệ thống "Quản lý tiệm bánh ngọt". Đây là một giải pháp phần mềm toàn diện giúp số hóa quy trình vận hành, kết nối liền mạch giữa bộ phận bán hàng (Front-of-house) và bộ phận sản xuất - kho (Back-of-house). Hệ thống tích hợp các chức năng cốt lõi bao gồm: quản lý sản phẩm và công thức, kiểm soát nguyên liệu tồn kho theo thời gian thực, bán hàng POS tại quầy, hỗ trợ đặt hàng trực tuyến, quản lý nhà cung cấp và các chương trình chăm sóc khách hàng.

Về mặt kỹ thuật, ứng dụng được thiết kế với giao diện trực quan, sử dụng các công nghệ hiện đại nhằm đảm bảo hiệu năng và trải nghiệm người dùng tốt nhất, bao gồm: Vue.js và Tailwind-CSS cho giao diện (Frontend), Express.js cho xử lý nghiệp vụ (Backend) và PostgreSQL cho hệ quản trị cơ sở dữ liệu.

1.2. Mục tiêu đề tài

1.2 Mục tiêu đề tài

Với định hướng trên, hệ thống được xây dựng nhằm đạt được các mục tiêu sau:

- Cung cấp công cụ quản lý sản phẩm, đơn hàng, nguyên liệu và nhân viên một cách tập trung và hiệu quả.
- Hỗ trợ quy trình bán hàng tại quầy (POS) và đặt bánh trực tuyến đơn giản, nhanh chóng.
- Tích hợp tính năng báo cáo – thống kê doanh thu, chi phí và tồn kho để hỗ trợ quản lý ra quyết định.
- Thiết kế giao diện thân thiện, dễ sử dụng cho nhân viên và quản lý.
- Đảm bảo tính bảo mật, độ tin cậy và khả năng mở rộng cho nhiều chi nhánh trong tương lai.

1.3 Khảo sát các giải pháp hiện có

Để xác định rõ phạm vi và yêu cầu của hệ thống "Quản lý cửa hàng bánh ngọt", nhóm thực hiện đã tiến hành khảo sát và phân tích các phương thức quản lý đang tồn tại trên thị trường, bao gồm phương pháp thủ công và các giải pháp phần mềm thương mại (SaaS) phổ biến như KiotViet hay Sapo FnB.

1.3.1 Quản lý thủ công (Excel và Sổ sách)

Đây là phương pháp truyền thống vẫn được sử dụng tại nhiều tiệm bánh quy mô nhỏ hoặc hộ gia đình.

Ưu điểm:

- Chi phí đầu tư ban đầu thấp, gần như bằng không.
- Linh hoạt, dễ dàng tùy chỉnh các cột, bảng tính theo ý muốn chủ cửa hàng mà không cần kiến thức lập trình.
- Quen thuộc, dễ tiếp cận với người lớn tuổi hoặc nhân viên không rành công nghệ.

Nhược điểm:

- **Sai sót dữ liệu:** Dễ xảy ra nhầm lẫn khi nhập liệu, tính toán sai doanh thu hoặc thất thoát hàng hóa do không có cơ chế kiểm soát chặt chẽ.

1.3. Khảo sát các giải pháp hiện có

- **Khó quản lý định lượng (Recipe):** Không thể tự động trừ nguyên liệu (bột, đường, trứng) khi bán thành phẩm (bánh), dẫn đến việc lêch kho nghiêm trọng.
- **Thiếu tính thời gian thực:** Chủ cửa hàng không thể xem báo cáo doanh thu hay tồn kho ngay lập tức khi không có mặt tại cửa hàng.
- **Bảo mật kém:** Dữ liệu lưu trên máy cá nhân dễ bị mất mát do hỏng hóc thiết bị hoặc virus.

1.3.2 Phần mềm quản lý bán hàng thương mại (KiotViet)

Các nền tảng "Software as a Service"(SaaS) đang chiếm lĩnh thị trường nhờ tính năng đa dạng và hỗ trợ phần cứng tốt.

Ưu điểm:

- Hệ sinh thái đầy đủ: Hỗ trợ tích hợp máy in hóa đơn, máy quét mã vạch, thanh toán QR.
- Ổn định và chịu tải tốt: Được vận hành bởi các công ty công nghệ lớn với hạ tầng server mạnh mẽ.
- Đa nền tảng: Có ứng dụng trên điện thoại giúp chủ cửa hàng quản lý từ xa.

Nhược điểm:

- **Chi phí duy trì:** Phải trả phí thuê bao hàng tháng (subscription), chi phí này có thể cao đối với các tiệm bánh khởi nghiệp quy mô nhỏ.
- **Dư thừa tính năng:** Được thiết kế cho nhiều ngành hàng (tạp hóa, thời trang, nhà hàng), nên giao diện thường phức tạp và có nhiều chức năng không cần thiết cho mô hình tiệm bánh.
- **Hạn chế tùy biến quy trình bếp:** Tính năng hiển thị đơn hàng cho bếp (Kitchen Display System) và quản lý hạn sử dụng theo lô (FEFO) thường chỉ có ở các gói cao cấp hoặc chưa được tối ưu sâu cho đặc thù làm bánh.

1.3. Khảo sát các giải pháp hiện có

1.3.3 So sánh và đề xuất giải pháp

Dưới đây là bảng so sánh các tiêu chí chính giữa giải pháp thủ công, phần mềm thương mại và hệ thống được xây dựng trong đồ án:

Bảng 1.3.1: So sánh các giải pháp quản lý tiệm bánh

| Tiêu chí | Quản lý thủ công (Excel) | Phần mềm thương mại (KiotViet) | Hệ thống của dự án (Matcha Bakery) |
|---|-----------------------------|-----------------------------------|--|
| Chi phí sử dụng | Thấp/Miễn phí | Trả phí hàng tháng | Chi phí xây dựng một lần (Self-hosted) |
| Quản lý công thức & Định lượng (BOM) | Thủ công, khó khăn | Có, nhưng tổng quát | Chuyên sâu (Tự động trừ kho theo công thức) |
| Quy trình Bếp (Kitchen View) | Dùng giấy/Order ticket | Màn hình KDS (Gói cao cấp) | Tích hợp sẵn màn hình riêng cho thợ bánh |
| Nguyên tắc tồn kho | Không kiểm soát | LIFO/FIFO | Hỗ trợ FEFO (Hết hạn trước xuất trước) |
| Giao diện (UI/UX) | Rối rắm | Phức tạp, nhiều thao tác | Tối giản , tập trung vào nghiệp vụ cốt lõi |
| Khả năng mở rộng | Kém | Tốt | Linh hoạt (Mã nguồn mở) |

Kết luận: Từ bảng phân tích trên, hệ thống "Matcha Bakery Management" được phát triển nhằm mục đích lấp đầy khoảng trống giữa hai giải pháp hiện có. Hệ thống tập trung giải quyết các vấn đề cốt lõi của tiệm bánh:

1. Tự động hóa quy trình trừ kho nguyên liệu dựa trên thành phẩm bán ra.
2. Cung cấp giao diện chuyên biệt cho từng bộ phận (Thu ngân, Bếp, Quản lý) để tối ưu hiệu suất làm việc.
3. Loại bỏ chi phí thường niên, phù hợp với định hướng triển khai nội bộ cho các cửa hàng vừa và nhỏ.

1.4 Bô cục bài báo cáo

Bài báo cáo được chia thành các phần sau:

1. Giới thiệu đề tài
2. Công nghệ sử dụng
3. Phân tích hệ thống
4. Kết quả
5. Định hướng phát triển
6. Danh mục tài liệu tham khảo

Chương 2

Công nghệ và phần mềm sử dụng

2.1 Công nghệ ở Frontend

2.1.1 Vue.js



Hình 2.1.1: Logo Vue.js

Giới thiệu

Vue.js là một framework JavaScript lũy tiến để xây dựng giao diện người dùng. Nó được thiết kế từ đầu để có thể được áp dụng tăng dần. Core library chỉ tập trung vào tầng View, rất dễ dàng tích hợp với các thư viện khác hoặc các project hiện có. Vue.js đặc biệt được ưa chuộng nhờ sự dễ học, hiệu suất tốt và khả năng mở rộng linh hoạt, phù hợp cho nhiều loại dự án từ các ứng dụng đơn giản đến các SPA (Single Page Application) phức tạp.

Ưu điểm nổi bật của Vue.js:

- **Dễ học và sử dụng:** Vue có cú pháp đơn giản, rõ ràng và tài liệu hướng dẫn rất chi tiết, giúp các nhà phát triển nhanh chóng làm quen và bắt đầu dự án.
- **Kích thước nhỏ gọn và hiệu suất cao:** Core library của Vue.js rất nhẹ, giúp ứng dụng có thời gian tải nhanh. Nó sử dụng Virtual DOM để cập nhật giao diện người dùng một cách hiệu quả, tối ưu hóa hiệu suất.

2.1. Công nghệ ở Frontend

- **Tính linh hoạt:** Vue.js cho phép tích hợp dễ dàng vào các dự án hiện có. Nó có thể được sử dụng để phát triển các thành phần UI nhỏ hoặc xây dựng toàn bộ ứng dụng phức tạp.
- **Hệ sinh thái phong phú:** Mặc dù nhỏ gọn, Vue.js có một hệ sinh thái mạnh mẽ với các thư viện chính thức và bên thứ ba như Vue Router (quản lý định tuyến), Vuex (quản lý trạng thái), Vue CLI (công cụ dòng lệnh), và Nuxt.js (framework SSR).
- **Cộng đồng lớn và hỗ trợ tốt:** Vue.js có một cộng đồng toàn cầu lớn mạnh, cung cấp sự hỗ trợ và tài nguyên phong phú, bao gồm các diễn đàn, nhóm chat và các khóa học.

Nhược điểm của Vue.js:

- **Quy mô cộng đồng so với React/Angular:** Mặc dù cộng đồng đang phát triển nhanh, nhưng vẫn có thể nhỏ hơn một chút so với React hoặc Angular, dẫn đến ít tài nguyên hoặc giải pháp cho các vấn đề hiếm gặp hơn.
- **Khó khăn với các dự án lớn:** Đối với các dự án rất lớn và phức tạp, việc quản lý trạng thái có thể trở nên khó khăn hơn nếu không sử dụng Vuex một cách hiệu quả.
- **Ít được các tập đoàn lớn sử dụng:** So với React được Facebook hậu thuẫn và Angular được Google hậu thuẫn, Vue.js đôi khi ít được các tập đoàn lớn tin tưởng cho các dự án quy mô doanh nghiệp.
- **Hạn chế về tài nguyên việc làm:** Mặc dù phổ biến, số lượng việc làm yêu cầu kỹ năng Vue.js có thể ít hơn so với React hoặc Angular ở một số thị trường.

Trong đề tài, Vue.js được sử dụng để xây dựng giao diện cho hệ thống quản lý tiệm bánh, bao gồm các chức năng: bán hàng POS, quản lý đơn hàng, tồn kho, và hiển thị báo cáo.

2.1.2 TailwindCSS



Hình 2.1.2: Logo TailwindCSS

Giới thiệu

Tailwind CSS là một framework CSS utility-first để xây dựng các thiết kế tùy chỉnh một cách nhanh chóng mà không cần rời khỏi HTML. Thay vì các lớp được định nghĩa sẵn cho các thành phần như nút (button) hoặc thẻ (card), Tailwind cung cấp các lớp tiện ích cấp thấp (utility classes) cho phép kiểm soát từng thuộc tính CSS nhỏ nhất như `flex`, `pt-4`, `text-center`, `rotate-90`.

Ưu điểm của TailwindCSS:

- **Tốc độ phát triển nhanh:** Với Tailwind, bạn có thể xây dựng giao diện người dùng mà không cần viết CSS tùy chỉnh. Việc này giúp tăng tốc quá trình phát triển đáng kể vì bạn không phải chuyển đổi giữa các tệp HTML và CSS.
- **Thiết kế linh hoạt và tùy biến cao:** Tailwind không áp đặt bất kỳ quy tắc thiết kế nào. Bạn có toàn quyền kiểm soát giao diện và cảm nhận của ứng dụng. Bằng cách kết hợp các utility classes, bạn có thể tạo ra mọi kiểu dáng mà không bị giới hạn bởi các thành phần được định nghĩa sẵn.
- **Kích thước tệp CSS nhỏ gọn:** Tailwind sử dụng PurgeCSS (hoặc các công cụ tương tự) để loại bỏ tất cả các CSS không sử dụng trong quá trình biên dịch production, đảm bảo rằng tệp CSS cuối cùng của bạn cực kỳ nhỏ gọn và chỉ chứa những gì bạn thực sự dùng.
- **Tính nhất quán trong thiết kế:** Mặc dù cho phép tùy biến, Tailwind khuyến khích việc sử dụng một hệ thống thiết kế có cấu trúc bằng cách cung cấp các giá trị mặc định cho màu sắc, khoảng cách, font chữ, giúp duy trì tính nhất quán trong toàn bộ dự án.

2.1. Công nghệ ở Frontend

- **Dễ bảo trì:** Khi các kiểu dáng được định nghĩa trực tiếp trong HTML, việc thay đổi hoặc bảo trì một thành phần trở nên dễ dàng hơn. Bạn có thể nhanh chóng xác định các thuộc tính CSS được áp dụng cho một phần tử mà không cần phải tìm kiếm trong các tệp CSS lớn.

Nhược điểm của TailwindCSS:

- **HTML có thể trở nên dài dòng:** Việc sử dụng nhiều lớp tiện ích trong HTML có thể làm cho mã HTML trở nên dài và khó đọc, đặc biệt đối với các thành phần phức tạp.
- **Đường cong học tập ban đầu:** Đối với những người mới làm quen với Tailwind, có thể mất một thời gian để học và làm quen với tất cả các utility classes và cách sử dụng chúng một cách hiệu quả.
- **Không phù hợp với các dự án nhỏ:** Đối với các dự án rất nhỏ hoặc các trang web tĩnh đơn giản, việc thiết lập và cấu hình Tailwind có thể tốn thời gian hơn so với việc chỉ viết CSS truyền thống.
- **Khó khăn trong việc tìm kiếm lớp CSS:** Nếu bạn không quen thuộc với cú pháp, việc tìm kiếm một lớp tiện ích cụ thể có thể mất thời gian.

Trong hệ thống, TailwindCSS giúp tạo một giao diện đơn giản, trực quan, dễ sử dụng cho nhân viên và quản lý cửa hàng.

2.2 Công nghệ ở Backend

2.2.1 Express.js



Hình 2.2.1: Logo Express.js

Giới thiệu

Express.js (thường được gọi tắt là Express) là một framework web tối giản và linh hoạt cho Node.js, cung cấp một bộ tính năng mạnh mẽ để phát triển các ứng dụng web và API. Nó là một trong những framework phổ biến nhất trong hệ sinh thái Node.js và là một phần quan trọng của "MEAN stack" hoặc "MERN stack"(MongoDB, Express, React/Angular/Vue, Node.js).

Ưu điểm

- **Tốc độ phát triển nhanh:** Express.js được thiết kế để tối ưu hóa tốc độ, cung cấp các công cụ và cấu trúc để xây dựng API và ứng dụng web một cách hiệu quả.
- **Linh hoạt và tối giản:** Express cung cấp một lớp mỏng các tính năng cốt lõi cho web và API, cho phép các nhà phát triển tự do lựa chọn các thư viện và công cụ khác phù hợp với nhu cầu cụ thể của dự án.
- **Cộng đồng lớn và tài liệu phong phú:** Với một cộng đồng người dùng và nhà phát triển khổng lồ, Express.js có vô số tài liệu, hướng dẫn và giải pháp cho mọi vấn đề có thể gặp phải.
- **Khả năng mở rộng (Scalability):** Express.js dễ dàng tích hợp với các middleware (phần mềm trung gian) và module của bên thứ ba, cho phép mở rộng chức năng một cách linh hoạt.
- **Hiệu suất cao:** Express được xây dựng trên Node.js, vốn nổi tiếng về hiệu suất cao và khả năng xử lý các I/O không đồng bộ một cách hiệu quả, giúp Express trở thành lựa chọn tốt cho các ứng dụng cần xử lý nhiều yêu cầu đồng thời.

2.2. Công nghệ ở Backend

Nhược điểm

- **Tính chất tối giản có thể là nhược điểm:** Đối với các dự án lớn, việc thiếu các cấu trúc được định nghĩa sẵn có thể đòi hỏi nhiều công việc hơn trong việc tổ chức mã nguồn và quản lý các module.
- **Xử lý lỗi (Error Handling) không dễ dàng:** Việc xử lý lỗi trong Express.js có thể trở nên phức tạp đối với các nhà phát triển mới, đặc biệt là trong các ứng dụng có nhiều middleware.
- **Bảo mật:** Mặc dù Express cung cấp một số tính năng bảo mật cơ bản, nhưng việc triển khai các biện pháp bảo mật nâng cao thường đòi hỏi các thư viện và cấu hình bổ sung.
- **Ít quy ước hơn các framework đầy đủ tính năng:** So với các framework full-stack như Ruby on Rails hay Laravel, Express.js ít có các "opinioned defaults" (quy ước mặc định), yêu cầu nhà phát triển phải đưa ra nhiều quyết định về cấu trúc hơn.

Trong đề tài, Express.js đảm nhận vai trò backend chính, xử lý logic nghiệp vụ như: quản lý đơn hàng, quản lý sản phẩm-nguyên liệu, xác thực người dùng, và tích hợp với PostgreSQL.

2.2.2 PostgreSQL



Hình 2.2.2: Logo PostgreSQL

Giới thiệu

PostgreSQL (thường được phát âm là "Post-Gres-Q-L") là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS) mã nguồn mở mạnh mẽ. Nó nổi tiếng về độ tin cậy, mạnh mẽ về tính năng và hiệu suất cao. Được phát triển từ dự án POSTGRES tại Đại học California, Berkeley, từ năm 1986, PostgreSQL đã trải qua nhiều năm phát triển và cải tiến liên tục bởi một cộng đồng toàn cầu, biến nó thành một trong những cơ sở dữ liệu tiên tiến nhất hiện nay.

Ưu điểm

- **Độ tin cậy và tính toàn vẹn dữ liệu:** PostgreSQL được biết đến với khả năng bảo vệ dữ liệu mạnh mẽ, tuân thủ nghiêm ngặt các tiêu chuẩn ACID (Atomicity, Consistency, Isolation, Durability) đảm bảo tính toàn vẹn của giao dịch.
- **Mã nguồn mở và miễn phí:** Là phần mềm mã nguồn mở, PostgreSQL hoàn toàn miễn phí để sử dụng, sửa đổi và phân phối. Điều này giúp giảm đáng kể chi phí cấp phép cho các dự án.
- **Khả năng mở rộng và hiệu suất cao:** PostgreSQL có khả năng mở rộng tốt để xử lý khối lượng dữ liệu lớn và số lượng người dùng đồng thời cao. Nó cung cấp các kỹ thuật như index, tối ưu hóa truy vấn và phân vùng dữ liệu để đạt được hiệu suất tối ưu.
- **Hỗ trợ đa dạng các kiểu dữ liệu:** Ngoài các kiểu dữ liệu quan hệ truyền thống, PostgreSQL còn hỗ trợ nhiều kiểu dữ liệu nâng cao như JSON/JSONB, XML, Hstore (key-value store), mảng, hình học và kiểu dữ liệu tùy chỉnh, làm cho nó trở nên linh hoạt cho nhiều ứng dụng khác nhau.
- **Tính năng phong phú và tiên tiến:** PostgreSQL cung cấp hàng loạt tính năng hiện đại như stored procedures, triggers, views, khóa ngoại, thừa kế bảng, kiểm soát đa phiên bản đồng thời (MVCC) và full-text search.

2.2. Công nghệ ở Backend

- **Cộng đồng mạnh mẽ và hỗ trợ chuyên nghiệp:** PostgreSQL có một cộng đồng toàn cầu lớn và năng động, cung cấp sự hỗ trợ rộng rãi thông qua các diễn đàn, danh sách gửi thư và tài liệu chi tiết. Nhiều công ty cũng cung cấp dịch vụ hỗ trợ chuyên nghiệp.
- **Tương thích SQL tiêu chuẩn:** PostgreSQL tuân thủ hầu hết các tiêu chuẩn SQL, giúp các nhà phát triển dễ dàng chuyển đổi từ các hệ quản trị cơ sở dữ liệu khác.

Nhược điểm

- **Quản trị phức tạp hơn một số RDBMS khác:** Mặc dù mạnh mẽ, việc quản trị và tối ưu hóa PostgreSQL có thể đòi hỏi kiến thức chuyên sâu hơn so với một số hệ quản trị cơ sở dữ liệu khác như MySQL, đặc biệt đối với các tính năng nâng cao.
- **Tài nguyên cần thiết:** PostgreSQL có thể tiêu thụ nhiều tài nguyên hệ thống (RAM, CPU) hơn so với các cơ sở dữ liệu nhẹ hơn, đặc biệt khi xử lý các truy vấn phức tạp hoặc khối lượng dữ liệu lớn.
- **Đường cong học tập:** Đối với người mới bắt đầu, việc làm quen với tất cả các tính năng và cấu hình của PostgreSQL có thể có một đường cong học tập tương đối dốc.

Trong đề tài, PostgreSQL được dùng để lưu trữ toàn bộ dữ liệu của hệ thống: thông tin sản phẩm, công thức (recipe/BOM), tồn kho, đơn hàng, khách hàng, nhà cung cấp và báo cáo tài chính.

Chương 3

Phân tích thiết kế hệ thống

3.1 Các Actor có trong hệ thống

- **Khách hàng (Customer):** Người mua hàng, thực hiện đặt món, thanh toán và theo dõi đơn hàng.
- **Thu ngân (Cashier):** Nhân viên tại quầy, chịu trách nhiệm xử lý đơn hàng trực tiếp và xác nhận thanh toán.
- **Thợ làm bánh (Baker):** Nhân viên bộ phận bếp, nhận yêu cầu chế biến và cập nhật trạng thái món ăn.
- **Quản lý (Manager):** Người chịu trách nhiệm vận hành chung, quản lý kho, nhân sự và xem báo cáo kinh doanh.
- **Quản trị hệ thống (Admin):** Chịu trách nhiệm về kỹ thuật, bảo mật, cấu hình hệ thống và quản lý tài khoản người dùng.

3.2. Phân tích nhu cầu (User Stories)

3.2 Phân tích nhu cầu (User Stories)

Dựa trên khảo sát thực tế, nhu cầu người dùng được mô hình hóa dưới dạng các câu chuyện người dùng (User Stories) để làm rõ mối quan hệ giữa tính năng và lợi ích kinh doanh.

| ID | Tác nhân | User Story (Nhu cầu & Lợi ích) |
|------|--------------|--|
| US01 | Khách hàng | Là một Khách hàng , tôi muốn xem menu trực quan và đặt hàng online để tiết kiệm thời gian chờ đợi tại cửa hàng. |
| US02 | Khách hàng | Là một Khách hàng , tôi muốn xem lịch sử đơn hàng để có thể đặt lại các món yêu thích một cách nhanh chóng. |
| US03 | Thu ngân | Là một Thu ngân , tôi muốn hệ thống tự động tính tổng tiền và in hóa đơn để giảm thiểu sai sót tính toán trong giờ cao điểm. |
| US04 | Thợ làm bánh | Là một Thợ làm bánh , tôi muốn nhận phiếu order trên màn hình kỹ thuật số để không bị thất lạc giấy tờ và biết chính xác thứ tự ưu tiên. |
| US05 | Quản lý | Là một Quản lý , tôi muốn xem báo cáo doanh thu và tồn kho theo thời gian thực để ra quyết định nhập hàng kịp thời. |
| US06 | Admin | Là một Admin , tôi muốn phân quyền chi tiết cho từng vai trò để đảm bảo nhân viên chỉ truy cập đúng chức năng phận sự. |

Bảng 3.2.1: Bảng phân tích User Stories

3.3 Yêu cầu chức năng (Functional Requirements)

Hệ thống phiên bản 1.0 cung cấp các nhóm chức năng chính sau:

- **Quản lý tài khoản:** Đăng ký, Đăng nhập (SSO/Local), Quản lý profile cá nhân.
- **Quản lý danh mục & Sản phẩm:** Hiển thị menu, tìm kiếm, lọc sản phẩm, hiển thị trạng thái còn/hết hàng.
- **Quản lý quy trình đặt hàng:** Tạo đơn hàng, xử lý giỏ hàng, chọn phương thức thanh toán, xuất hóa đơn điện tử.
- **Quản lý bếp (Kitchen Display System):** Hiển thị hàng chờ (Queue), cập nhật trạng thái chế biến (Pending → Cooking → Ready).
- **Quản lý kho (Inventory):** Nhập/xuất nguyên liệu, cảnh báo tồn kho thấp, đồng bộ số lượng sau khi bán.
- **Quản trị hệ thống:** Quản lý người dùng (CRUD), Phân quyền (RBAC), Cấu hình tham số hệ thống.
- **Báo cáo & Thông kê:** Báo cáo doanh thu ngày/tháng, Báo cáo mặt hàng bán chạy, Báo cáo hiệu suất nhân viên.

3.4. Yêu cầu phi chức năng (Non-Functional Requirements)

3.4 Yêu cầu phi chức năng (Non-Functional Requirements)

Hệ thống được thiết kế để đáp ứng các tiêu chuẩn kỹ thuật và chất lượng dịch vụ (QoS) cho Phiên bản 1.0 như sau:

3.4.1 Hiệu năng (Performance)

- **Thời gian phản hồi (Response Time):** Hệ thống đảm bảo thời gian phản hồi trung bình dưới 2 giây cho các tác vụ truy vấn dữ liệu thông thường trong điều kiện mạng ổn định.
- **Khả năng chịu tải (Throughput):** Hệ thống hỗ trợ xử lý đồng thời tối thiểu 100 người dùng (concurrent users) mà không gây gián đoạn dịch vụ.
- **Đồng bộ dữ liệu:** Dữ liệu giữa các phân hệ (Thu ngân - Bếp - Kho) phải được đồng bộ gần như tức thời (near real-time) để đảm bảo tính chính xác của tồn kho.

3.4.2 Bảo mật (Security)

- **Mã hóa dữ liệu:** Mật khẩu người dùng phải được mã hóa một chiều (hashing bằng Bcrypt hoặc Argon2) trước khi lưu vào cơ sở dữ liệu. Dữ liệu truyền tải qua mạng phải sử dụng giao thức HTTPS (TLS 1.2 trở lên).
- **Kiểm soát truy cập:** Áp dụng cơ chế phân quyền dựa trên vai trò (RBAC - Role Based Access Control) chặt chẽ. Ngăn chặn các hành vi leo thang đặc quyền (Privilege Escalation).
- **Nhật ký hệ thống (Logging):** Mọi hành động nhạy cảm (Đăng nhập, Xóa dữ liệu, Thay đổi cấu hình) đều phải được ghi log (Audit Trail) để phục vụ tra soát.

3.4.3 Độ tin cậy và Khả dụng (Reliability & Availability)

- **Thời gian hoạt động (Uptime):** Hệ thống cam kết thời gian hoạt động đạt 99.5% trong khung giờ hành chính của cửa hàng.
- **Sao lưu dữ liệu (Backup):** Cơ sở dữ liệu phải được tự động sao lưu định kỳ hàng ngày (Daily Backup) và có cơ chế khôi phục (Recovery) khi gặp sự cố.

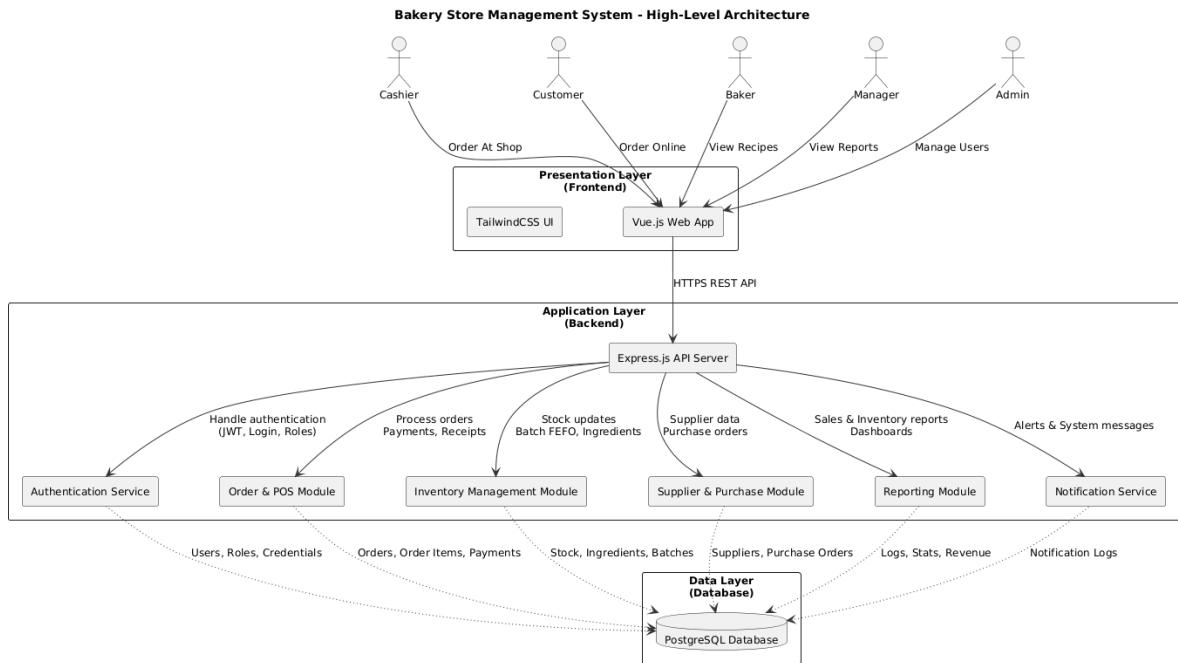
3.4. Yêu cầu phi chức năng (Non-Functional Requirements)

3.4.4 Tương thích và Môi trường (Compatibility)

- **Nền tảng hỗ trợ:** Hệ thống vận hành tốt trên các trình duyệt web phổ biến hiện đại (Google Chrome v90+, Edge, Safari, Firefox).
- **Thiết bị:** Giao diện hỗ trợ Responsive Design, hiển thị tốt trên:
 - Desktop/Laptop (Dành cho Quản lý/Admin).
 - Máy tính bảng/POS (Dành cho Thu ngân/Bếp).
 - Điện thoại di động (Dành cho Khách hàng).

3.5. Thiết kế tổng quan hệ thống

3.5 Thiết kế tổng quan hệ thống



Hình 3.5.1: Sơ đồ kiến trúc tổng thể và các module chức năng của hệ thống quản lý tiệm bánh

Dựa trên sơ đồ kiến trúc mức cao (High-Level Architecture), hệ thống quản lý tiệm bánh ngọt được xây dựng theo mô hình kiến trúc ba lớp (3-tier architecture), đảm bảo sự phân tách rõ ràng giữa giao diện, logic nghiệp vụ và dữ liệu. Chi tiết các thành phần như sau:

1. Lớp Giao diện (Presentation Layer - Frontend):

- Sử dụng **Vue.js** làm framework chính cho ứng dụng web và **TailwindCSS** để thiết kế giao diện (UI).
- Cung cấp các giao diện tương tác đặc thù cho từng nhóm người dùng: Thu ngân (Cashier - bán hàng tại quầy), Khách hàng (Customer - đặt online), Thợ bánh (Baker - xem công thức), Quản lý (Manager - xem báo cáo) và Quản trị viên (Admin - quản lý người dùng).
- Giao tiếp với Backend thông qua giao thức HTTPS REST API.

2. Lớp Ứng dụng (Application Layer - Backend):

Đây là trung tâm xử lý logic của hệ thống, vận hành trên **Express.js API Server**. Lớp này bao gồm các module chức năng chuyên biệt:

3.5. Thiết kế tổng quan hệ thống

- **Authentication Service:** Quản lý xác thực và phân quyền người dùng. Sử dụng JWT (JSON Web Token) để xử lý đăng nhập và kiểm soát quyền truy cập dựa trên vai trò (Roles).
- **Order & POS Module:** Xử lý quy trình bán hàng (POS), tiếp nhận đơn hàng (tại quầy và online), xử lý thanh toán và xuất hóa đơn.
- **Inventory Management Module:** Quản lý kho hàng, cập nhật biến động tồn kho nguyên liệu. Đặc biệt, module này hỗ trợ quản lý theo lô (Batch) và áp dụng nguyên tắc FEFO (First Expired, First Out - Hết hạn trước xuất trước).
- **Supplier & Purchase Module:** Quản lý thông tin nhà cung cấp và xử lý các đơn nhập hàng (Purchase orders).
- **Reporting Module:** Tổng hợp dữ liệu để tạo các báo cáo doanh thu, báo cáo tồn kho và hiển thị các bảng điều khiển (Dashboards) cho quản lý.
- **Notification Service:** Xử lý các cảnh báo hệ thống và gửi tin nhắn thông báo đến người dùng.

3. Lớp Dữ liệu (Data Layer - Database): Sử dụng hệ quản trị cơ sở dữ liệu **PostgreSQL** để lưu trữ bền vững và đảm bảo tính toàn vẹn dữ liệu. Các nhóm dữ liệu chính bao gồm:

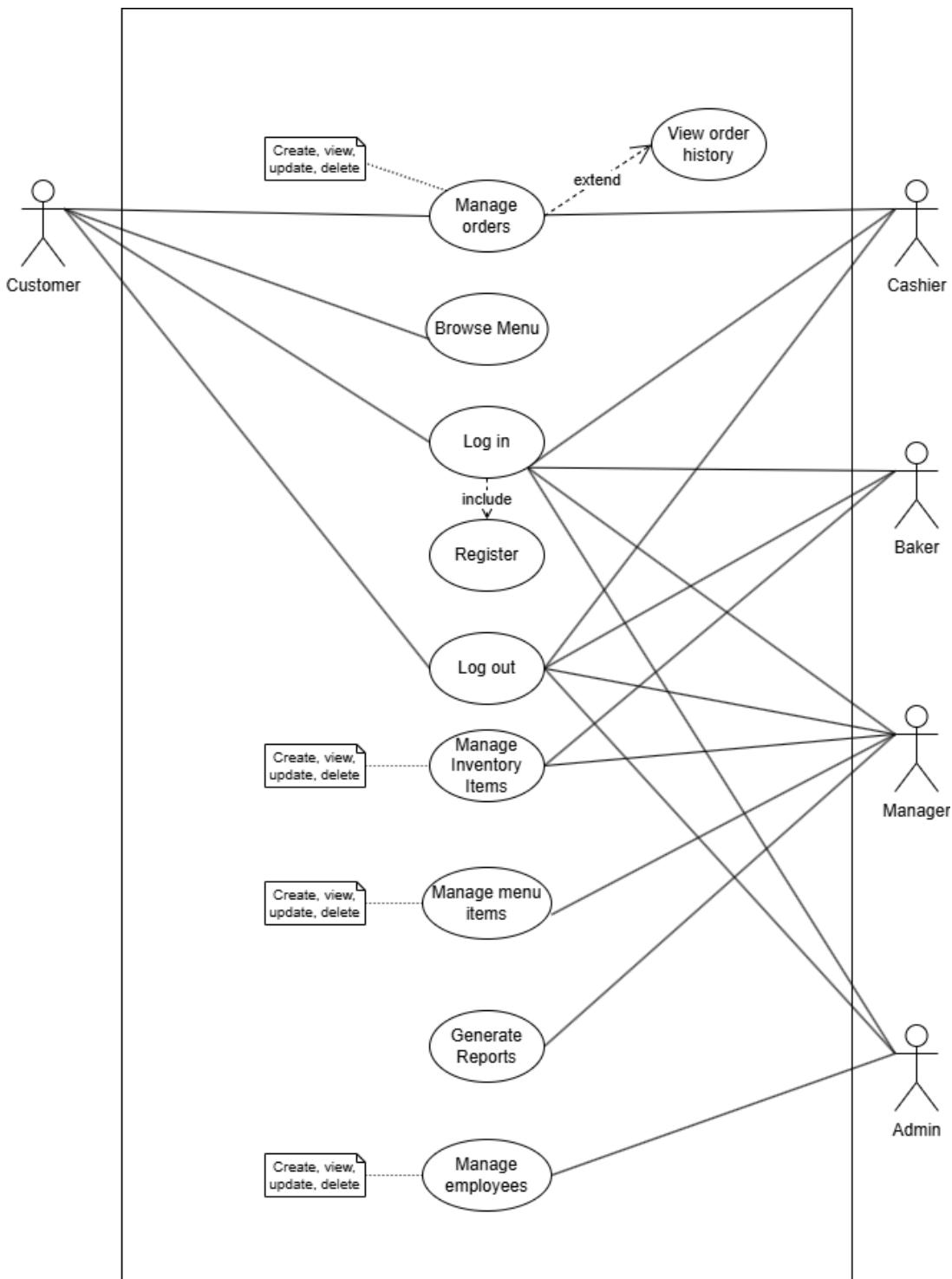
- *Users, Roles, Credentials:* Dữ liệu người dùng và phân quyền.
- *Orders, Order Items, Payments:* Dữ liệu giao dịch bán hàng.
- *Stock, Ingredients, Batches:* Dữ liệu kho, nguyên liệu và lô hàng.
- *Suppliers, Purchase Orders:* Dữ liệu nhập hàng.
- *Logs, Stats, Revenue:* Dữ liệu nhật ký và thông kê tài chính.
- *Notification Logs:* Lịch sử thông báo.

3.6. Use Case Diagram

3.6 Use Case Diagram

Biểu đồ Use Case mô tả cách các tác nhân như *Customer*, *Cashier*, *Baker*, *Manager* và *Admin* tương tác với các chức năng của hệ thống. Các chức năng chính bao gồm: đăng nhập, đăng ký, duyệt menu, đặt hàng, quản lý tồn kho, quản lý menu, quản lý nhân viên và tạo báo cáo. Sơ đồ cung cấp cái nhìn tổng quan về toàn bộ hành vi hệ thống từ góc độ người dùng.

3.6. Use Case Diagram



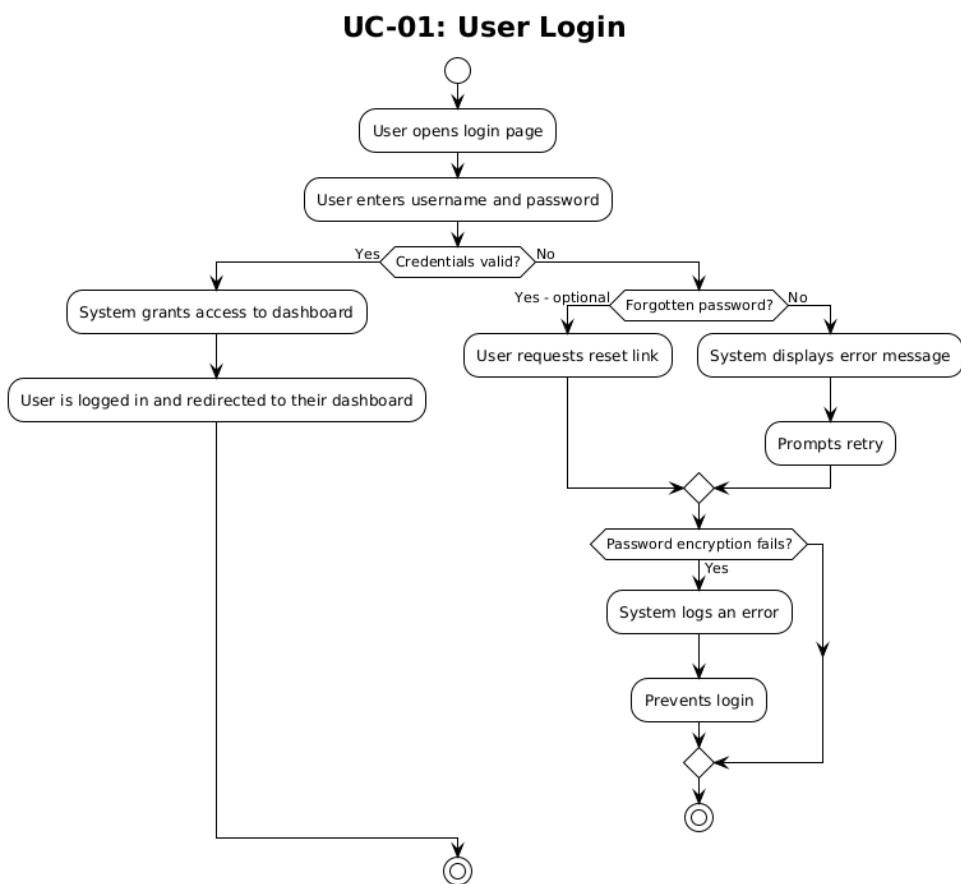
Hình 3.6.1: Biểu đồ Use Case của hệ thống

3.7. Activity Diagram

3.7 Activity Diagram

Biểu đồ hoạt động (Activity Diagram) mô tả chi tiết dòng chảy công việc (workflow) của các chức năng chính trong hệ thống, bao gồm các luồng hoạt động bình thường, các luồng rẽ nhánh và cách hệ thống xử lý các ngoại lệ.

3.7.1 UC-01: Đăng nhập (User Login)



3.7. Activity Diagram

| Thành phần | Mô tả chi tiết |
|-----------------|---|
| Tác nhân | Customer, Manager, Staff |
| Sự kiện bắt đầu | Người dùng mở trang đăng nhập. |
| Hoạt động chính | Nhập username/password → Hệ thống xác thực → Cấp quyền truy cập Dashboard. |
| Luồng rẽ nhánh | - Thông tin sai: Hiển thị lỗi, yêu cầu nhập lại. - Quên mật khẩu: Yêu cầu liên kết đặt lại mật khẩu. |
| Ngoại lệ | Lỗi mã hóa (Encryption Failure): Ghi log và chặn đăng nhập. |

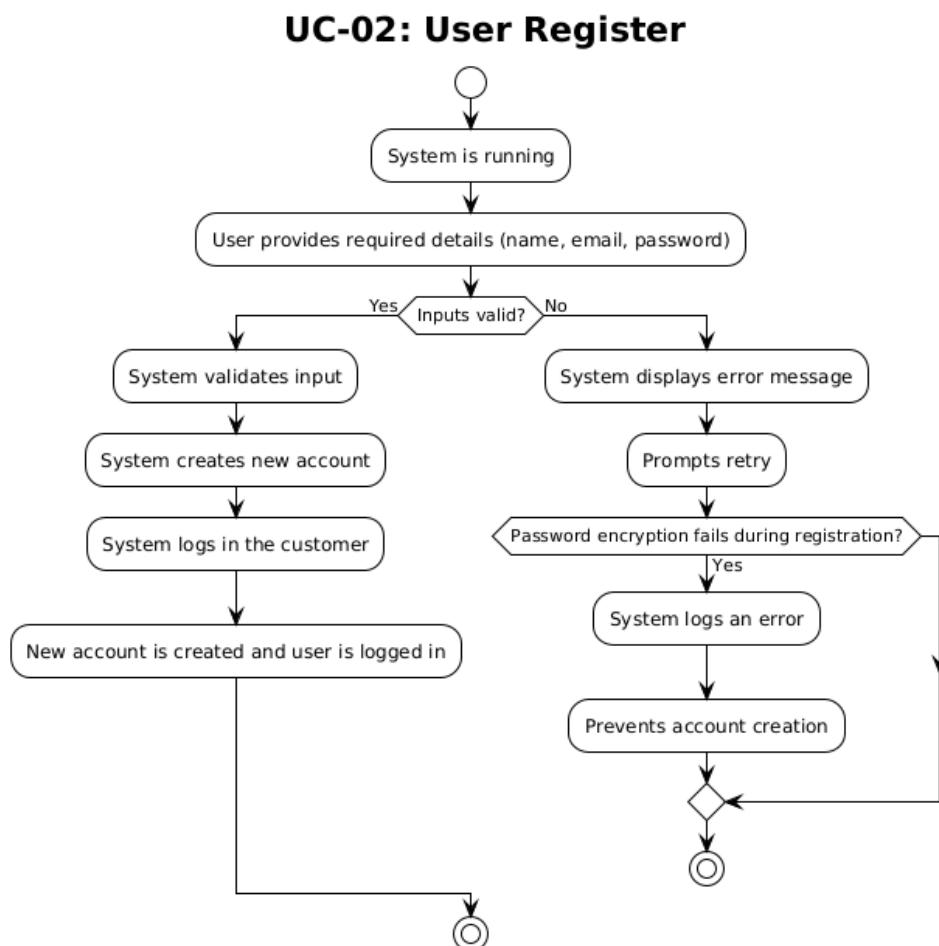
Bảng 3.7.1: Mô tả luồng hoạt động UC-01

Quy trình đăng nhập bắt đầu khi người dùng truy cập trang đăng nhập và nhập thông tin xác thực. Hệ thống thực hiện xác thực và điều hướng người dùng dựa trên kết quả.

- **Luồng chính:** Hệ thống kiểm tra thông tin. Nếu hợp lệ, người dùng được cấp quyền truy cập và chuyển hướng đến trang Dashboard tương ứng.
- **Luồng phụ:** Nếu thông tin sai, hệ thống hiển thị lỗi và yêu cầu nhập lại hoặc hỗ trợ lấy lại mật khẩu (Forgot password).
- **Ngoại lệ:** Nếu xảy ra lỗi mã hóa mật khẩu, hệ thống sẽ ghi log lỗi và ngăn chặn đăng nhập để đảm bảo an toàn.

3.7. Activity Diagram

3.7.2 UC-02: Đăng ký (User Register)



Hình 3.7.1: Biểu đồ hoạt động chức năng Đăng ký

| Thành phần | Mô tả chi tiết |
|------------------------|---|
| Tác nhân | Customer, Manager, Staff |
| Sự kiện bắt đầu | Người dùng cung cấp thông tin (tên, email, mật khẩu). |
| Hoạt động chính | Xác thực đầu vào → Tạo tài khoản mới → Tự động đăng nhập. |
| Luồng rẽ nhánh | Dữ liệu không hợp lệ: Hiển thị thông báo lỗi và yêu cầu nhập lại. |
| Ngoại lệ | Lỗi mã hóa khi đăng ký: Ghi log và ngăn tạo tài khoản. |

Bảng 3.7.2: Mô tả luồng hoạt động UC-02

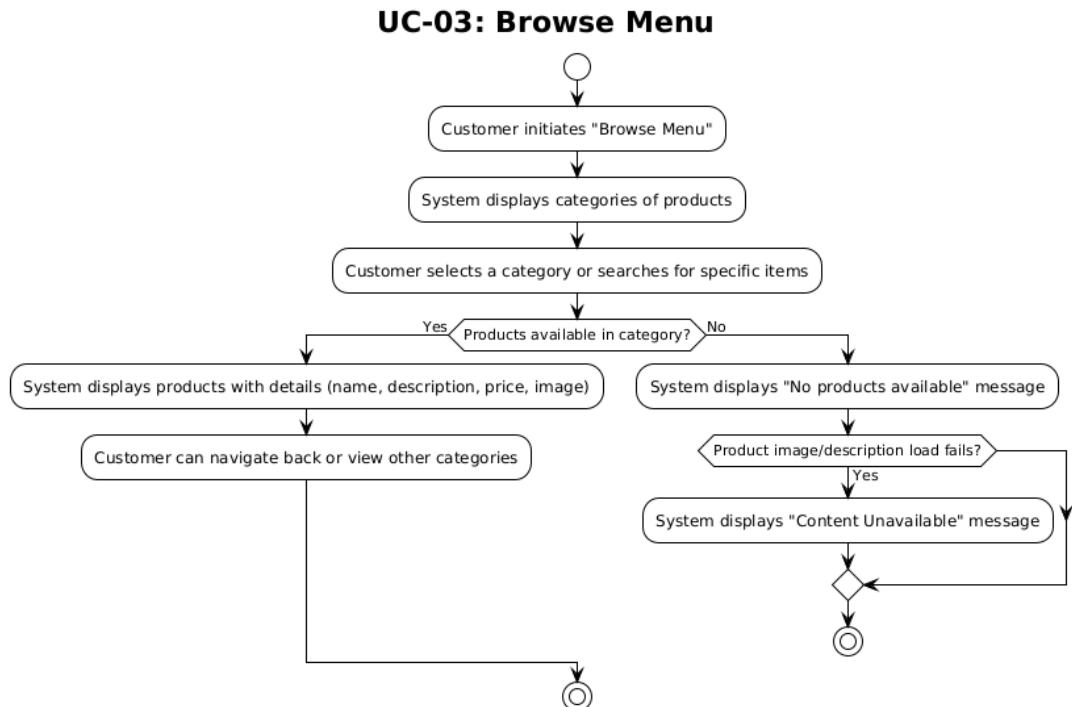
- Luồng chính:** Người dùng cung cấp thông tin (tên, email, mật khẩu). Hệ thống xác thực dữ liệu đầu vào. Nếu hợp lệ, tài khoản mới được tạo và người dùng tự động đăng nhập.

3.7. Activity Diagram

- **Ngoại lệ:** Nếu dữ liệu không hợp lệ hoặc quá trình mã hóa mật khẩu thất bại, hệ thống sẽ hiển thị thông báo lỗi và ngăn việc tạo tài khoản rác.

3.7. Activity Diagram

3.7.3 UC-03: Duyệt Menu (Browse Menu)



Hình 3.7.2: Biểu đồ hoạt động chức năng Duyệt Menu

| Thành phần | Mô tả chi tiết |
|-----------------|---|
| Tác nhân | Customer |
| Sự kiện bắt đầu | Khách hàng chọn "Browse Menu". |
| Hoạt động chính | Hiển thị danh mục → Chọn danh mục/Tìm kiếm → Xem chi tiết sản phẩm. |
| Luồng rẽ nhánh | Danh mục trống: Hiển thị thông báo "No products available". |
| Ngoại lệ | Lỗi tải ảnh/mô tả: Hiển thị "Content Unavailable" cho sản phẩm lỗi. |

Bảng 3.7.3: Mô tả luồng hoạt động UC-03

Khách hàng tìm kiếm và xem danh sách sản phẩm, kiểm tra chi tiết giá và hình ảnh trước khi đặt hàng.

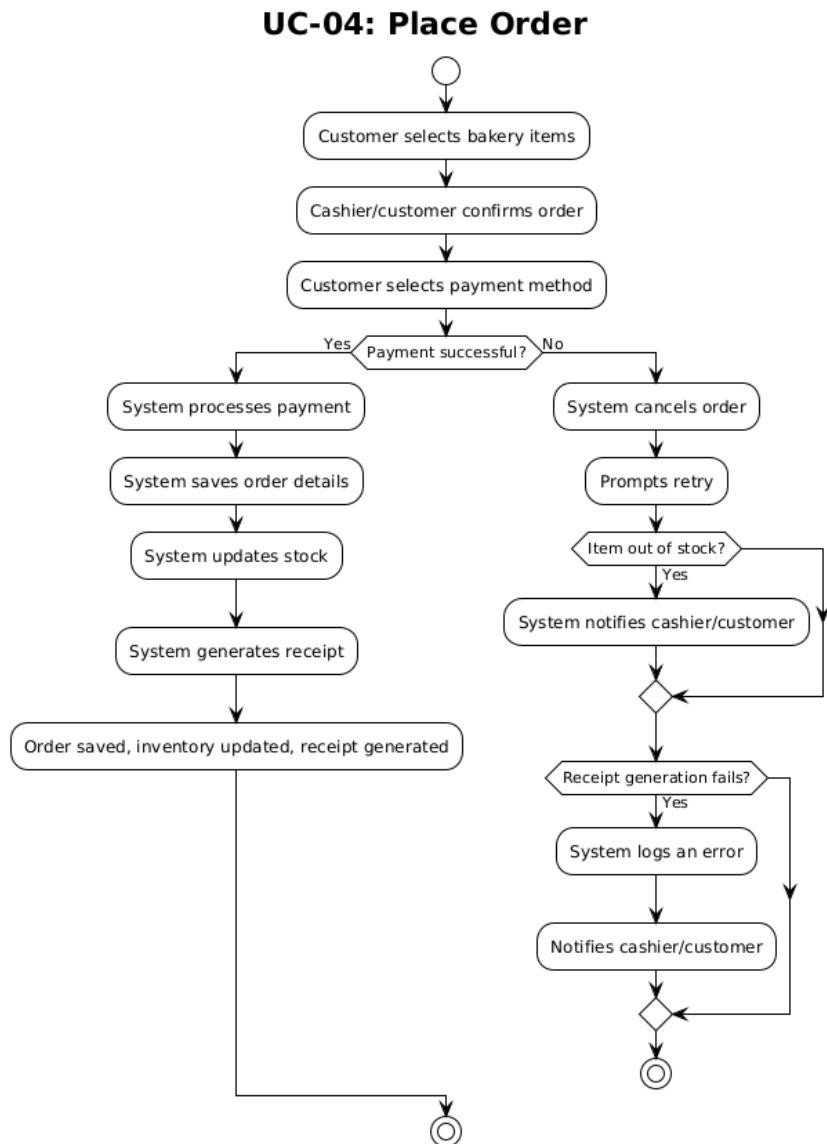
- **Luồng xử lý:** Hệ thống hiển thị các danh mục sản phẩm. Khách hàng chọn danh mục hoặc tìm kiếm.
- **Kiểm tra dữ liệu:** Hệ thống kiểm tra tính sẵn có của sản phẩm trong danh mục. Nếu có, hiển thị chi tiết (giá, mô tả, hình ảnh).

3.7. Activity Diagram

- **Xử lý lỗi:** Nếu tải hình ảnh hoặc nội dung thất bại, hệ thống hiển thị thông báo "Content Unavailable" thay vì để giao diện bị vỡ.

3.7. Activity Diagram

3.7.4 UC-04: Đặt hàng (Place Order)



Hình 3.7.3: Biểu đồ hoạt động chức năng Đặt hàng

3.7. Activity Diagram

| Thành phần | Mô tả chi tiết |
|-----------------|--|
| Tác nhân | Customer, Cashier |
| Sự kiện bắt đầu | Chọn món bánh và nước. |
| Hoạt động chính | Xác nhận đơn → Thanh toán → Lưu đơn → Trừ kho → Xuất hóa đơn. |
| Luồng rẽ nhánh | - Thanh toán thất bại: Hủy đơn, yêu cầu thử lại. - Hết hàng (Out of stock): Thông báo cho khách/thu ngân. |
| Ngoại lệ | Lỗi tạo hóa đơn: Ghi log lỗi, thông báo người dùng nhưng vẫn lưu đơn. |

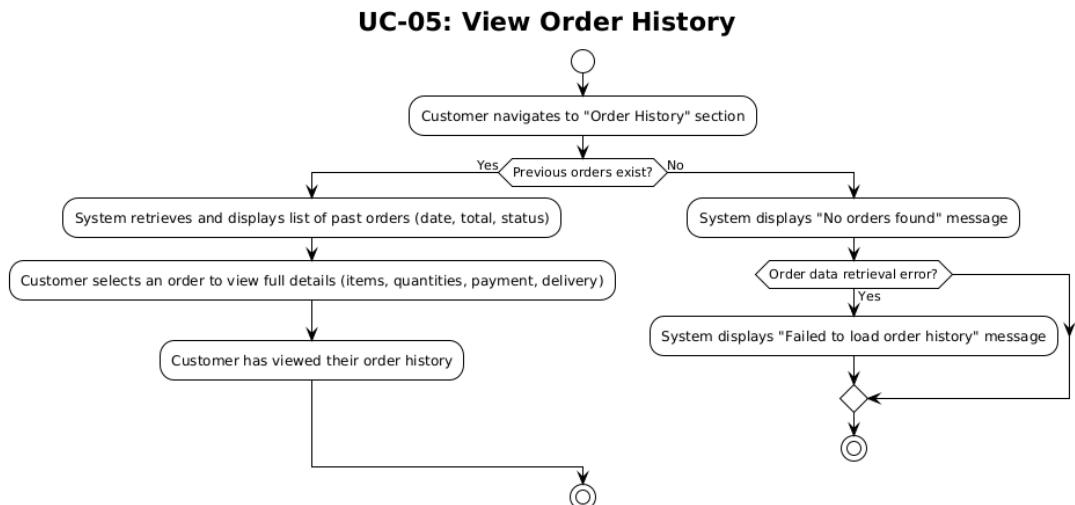
Bảng 3.7.4: Mô tả luồng hoạt động UC-04

Quy trình cốt lõi từ lúc chọn món, thanh toán đến khi xuất hóa đơn và cập nhật kho hàng.

- **Luồng chính:** Khách hàng chọn món → Xác nhận đơn → Chọn phương thức thanh toán.
Nếu thanh toán thành công, hệ thống lưu đơn, cập nhật kho và xuất hóa đơn.
- **Xử lý ngoại lệ:**
 - Nếu thanh toán thất bại: Hủy đơn hàng và yêu cầu thử lại.
 - Nếu hết hàng (Out of stock): Thông báo ngay lập tức cho thu ngân/khách hàng.
 - Lỗi xuất hóa đơn: Ghi log lỗi nhưng vẫn thông báo hoàn tất đơn hàng để không gián đoạn trải nghiệm.

3.7. Activity Diagram

3.7.5 UC-05: Xem lịch sử đơn hàng (View Order History)



Hình 3.7.4: Biểu đồ hoạt động chức năng Xem lịch sử đơn hàng

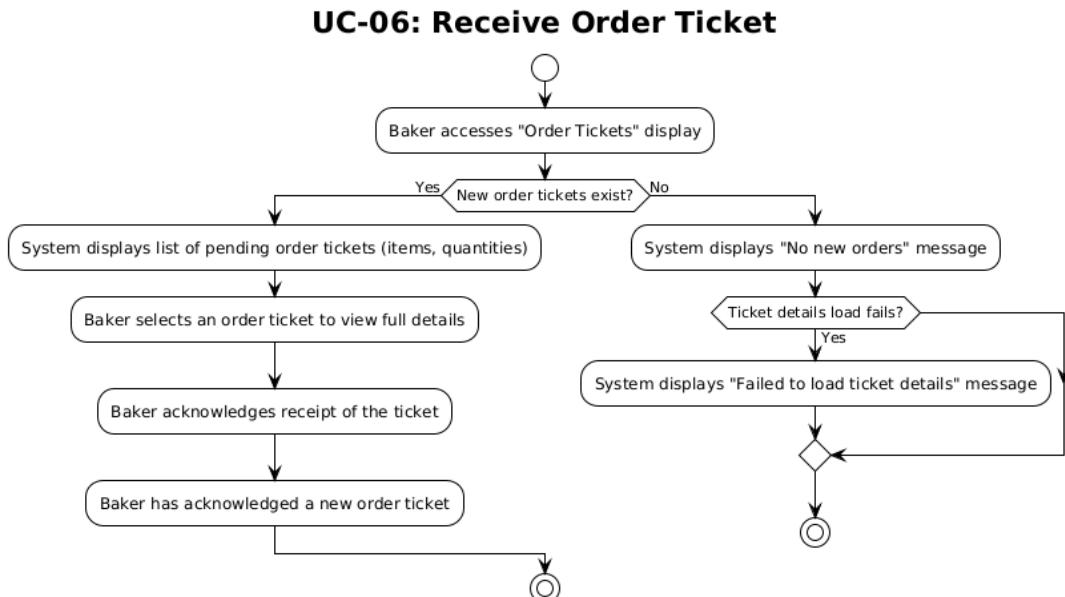
| Thành phần | Mô tả chi tiết |
|------------------------|---|
| Tác nhân | Customer |
| Sự kiện bắt đầu | Truy cập mục "Order History". |
| Hoạt động chính | Truy xuất danh sách đơn → Hiển thị tóm tắt → Xem chi tiết đơn chọn. |
| Luồng rẽ nhánh | Không có đơn hàng cũ: Hiển thị "No orders found". |
| Ngoại lệ | Lỗi truy xuất dữ liệu: Hiển thị thông báo "Failed to load order history". |

Bảng 3.7.5: Mô tả luồng hoạt động UC-05

Cho phép khách hàng tra cứu lại các đơn hàng đã đặt trong quá khứ.

- Luồng xử lý:** Hệ thống truy xuất danh sách đơn hàng dựa trên ID người dùng. Nếu có dữ liệu, hiển thị danh sách tóm tắt. Người dùng chọn xem chi tiết để thấy các món đã mua, tổng tiền và trạng thái.
- Ngoại lệ:** Nếu xảy ra lỗi truy vấn cơ sở dữ liệu (data retrieval error), hệ thống hiển thị thông báo lỗi thân thiện thay vì crash ứng dụng.

3.7.6 UC-06: Nhận phiếu order (Receive Order Ticket)



Hình 3.7.5: Biểu đồ hoạt động chức năng Nhận phiếu order

| Thành phần | Mô tả chi tiết |
|-----------------|--|
| Tác nhân | Baker |
| Sự kiện bắt đầu | Truy cập màn hình "Order Tickets". |
| Hoạt động chính | Hệ thống hiển thị danh sách vé chờ → Baker xem chi tiết → Xác nhận nhận đơn. |
| Luồng rẽ nhánh | Không có vé mới: Hiển thị thông báo "No new orders". |
| Ngoại lệ | Lỗi tải chi tiết vé: Hiển thị thông báo lỗi. |

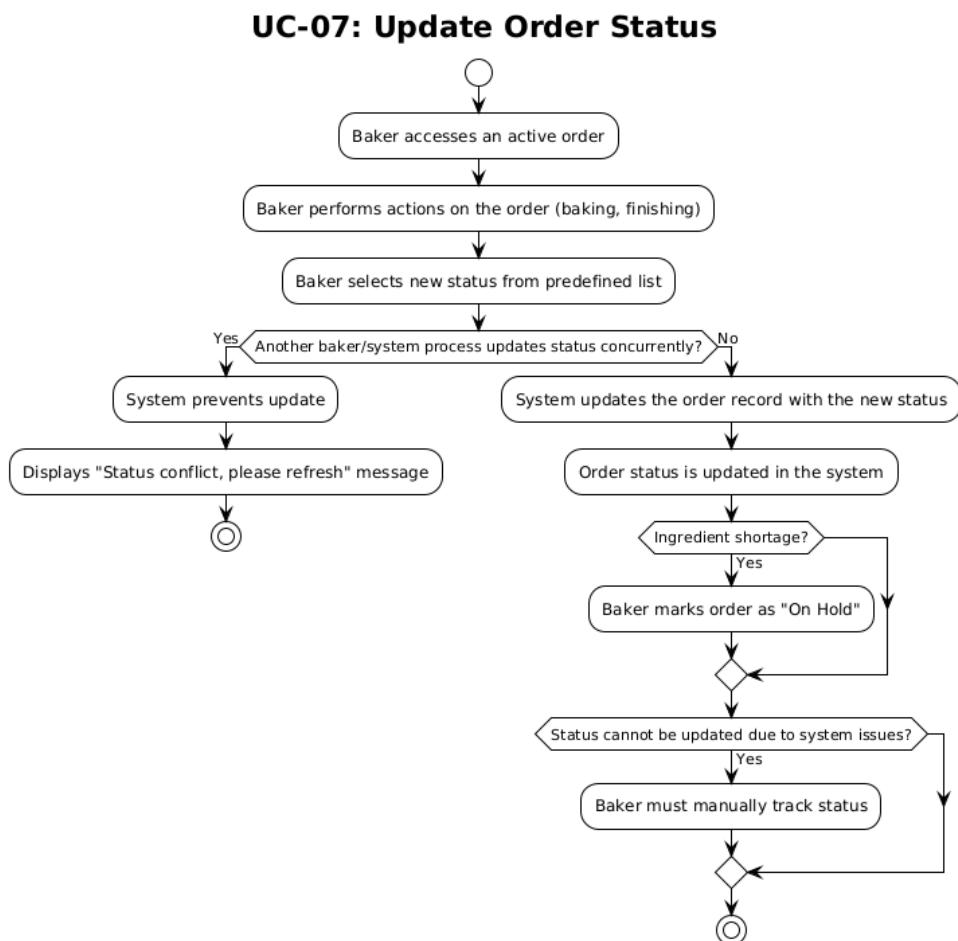
Bảng 3.7.6: Mô tả luồng hoạt động UC-06

Quy trình dành cho thợ làm bánh (Baker) để tiếp nhận yêu cầu chế biến từ hệ thống.

- Luồng xử lý:** Baker truy cập màn hình hiển thị đơn (Order Tickets). Hệ thống kiểm tra đơn mới. Nếu có, hiển thị danh sách món cần làm.
- Tương tác:** Baker chọn đơn để xem chi tiết và xác nhận (acknowledge) đã nhận đơn, giúp đồng bộ trạng thái với thu ngân.

3.7. Activity Diagram

3.7.7 UC-07: Cập nhật trạng thái đơn hàng (Update Order Status)



Hình 3.7.6: Biểu đồ hoạt động chức năng Cập nhật trạng thái đơn hàng

| Thành phần | Mô tả chi tiết |
|-----------------|---|
| Tác nhân | Baker |
| Sự kiện bắt đầu | Chọn một đơn hàng đang hoạt động. |
| Hoạt động chính | Thực hiện làm bánh → Chọn trạng thái mới → Hệ thống cập nhật. |
| Luồng rẽ nhánh | Thiếu nguyên liệu: Đánh dấu đơn là "On Hold". |
| Ngoại lệ | Xung đột cập nhật (Concurrency): Ngăn chặn và yêu cầu làm mới nếu có người khác đang cập nhật cùng lúc. |

Bảng 3.7.7: Mô tả luồng hoạt động UC-07

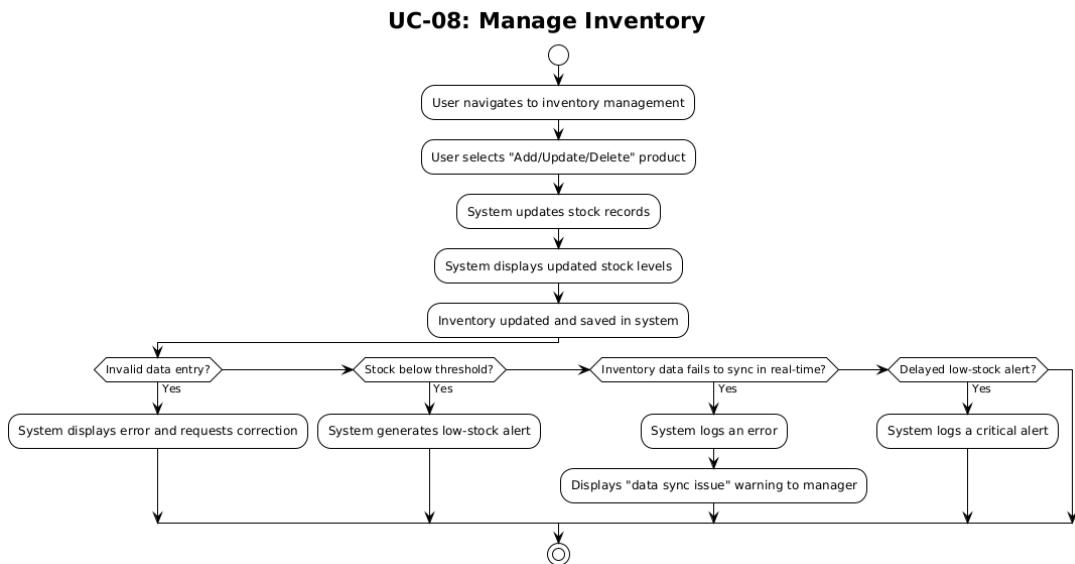
Thợ làm bánh cập nhật tiến độ trên hệ thống theo dõi.

3.7. Activity Diagram

- **Luồng chính:** Baker thực hiện làm bánh → Chọn trạng thái mới (đang làm, hoàn thành). Hệ thống cập nhật trạng thái vào CSDL.
- **Xử lý xung đột (Concurrency):** Nếu có 2 người cùng cập nhật một đơn hàng, hệ thống sẽ chặn và yêu cầu làm mới dữ liệu để tránh sai lệch.
- **Ngoại lệ:** Nếu thiếu nguyên liệu, Baker có thể đánh dấu đơn là "On Hold"(Tạm giữ).

3.7. Activity Diagram

3.7.8 UC-08: Quản lý tồn kho (Manage Inventory)



Hình 3.7.7: Biểu đồ hoạt động chức năng Quản lý tồn kho

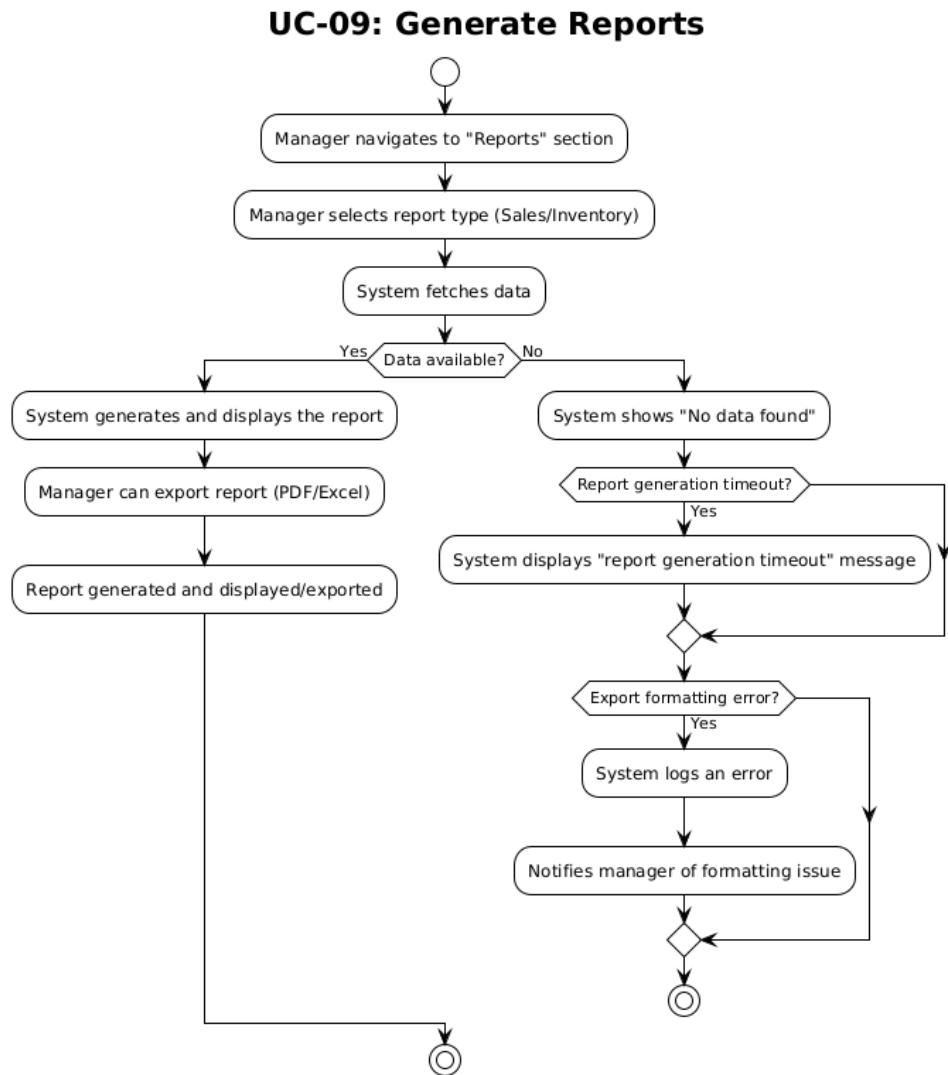
| Thành phần | Mô tả chi tiết |
|-----------------|---|
| Tác nhân | Manager |
| Sự kiện bắt đầu | Truy cập quản lý kho → Chọn Add/Update/Delete. |
| Hoạt động chính | Cập nhật dữ liệu kho → Hệ thống lưu và hiển thị mức tồn kho mới. |
| Luồng rẽ nhánh | Tồn kho dưới mức tối thiểu: Hệ thống tự động tạo cảnh báo "Low-stock alert". |
| Ngoại lệ | - Lỗi đồng bộ (Real-time Sync Failure). - Cảnh báo chậm (Delayed Alert): Ghi log lỗi nghiêm trọng. |

Bảng 3.7.8: Mô tả luồng hoạt động UC-08

Quản lý thực hiện thêm, sửa, xóa sản phẩm và nguyên liệu trong kho.

- **Luồng xử lý:** Quản lý chọn thêm/sửa/xóa sản phẩm. Hệ thống cập nhật kho và hiển thị số lượng mới.
- **Tự động hóa:**
 - Kiểm tra ngưỡng tồn kho an toàn (threshold). Nếu thấp hơn mức quy định, hệ thống tự động tạo cảnh báo "Low-stock alert".
 - Nếu nhập liệu sai (số âm, sai định dạng), hệ thống báo lỗi yêu cầu nhập lại.

3.7.9 UC-09: Tạo báo cáo (Generate Reports)



Hình 3.7.8: Biểu đồ hoạt động chức năng Tạo báo cáo

| Thành phần | Mô tả chi tiết |
|-----------------|---|
| Tác nhân | Manager |
| Sự kiện bắt đầu | Chọn loại báo cáo (Sales/Inventory). |
| Hoạt động chính | Hệ thống lấy dữ liệu → Hiển thị báo cáo → Xuất file (PDF/Excel). |
| Luồng rẽ nhánh | Không có dữ liệu: Hiển thị "No data found". |
| Ngoại lệ | - Timeout: Quá thời gian tạo báo cáo. - Lỗi định dạng xuất file: Ghi log và thông báo lỗi. |

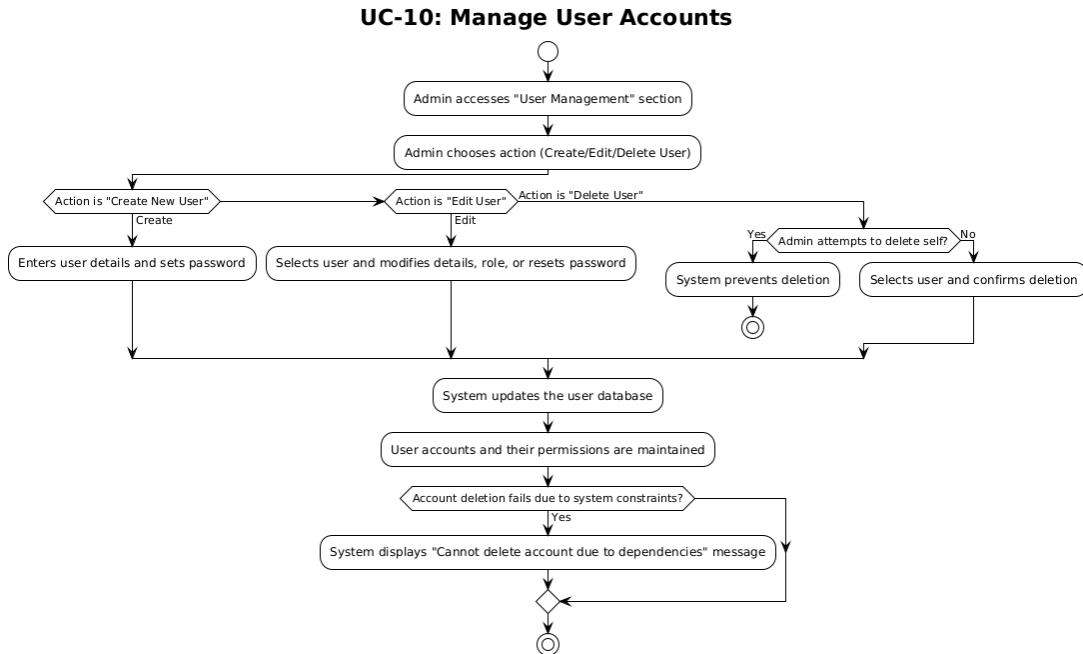
Bảng 3.7.9: Mô tả luồng hoạt động UC-09

3.7. Activity Diagram

Quản lý trích xuất các báo cáo doanh thu hoặc tồn kho để phân tích.

- **Luồng xử lý:** Chọn loại báo cáo (Doanh thu/Tồn kho) → Hệ thống lấy dữ liệu → Hiển thị và cho phép xuất file (PDF/Excel).
- **Xử lý lỗi:**
 - Không có dữ liệu: Thông báo "No data found".
 - Timeout (quá thời gian chờ): Thông báo lỗi đường truyền.
 - Lỗi định dạng xuất file: Ghi log lỗi và thông báo cho quản lý kiểm tra lại.

3.7.10 UC-10: Quản lý tài khoản người dùng (Manage User Accounts)



Hình 3.7.9: Biểu đồ hoạt động chức năng Quản lý tài khoản

| Thành phần | Mô tả chi tiết |
|-----------------|---|
| Tác nhân | Admin |
| Sự kiện bắt đầu | Truy cập User Management. |
| Hoạt động chính | Tạo mới (Create), Chính sửa (Edit), hoặc Xóa (Delete) người dùng → Cập nhật CSDL. |
| Luồng rẽ nhánh | Cố gắng tự xóa tài khoản (Delete Self): Hệ thống ngăn chặn hành động này. |
| Ngoại lệ | Xóa thất bại do ràng buộc dữ liệu: Hiển thị thông báo lỗi dependencies. |

Bảng 3.7.10: Mô tả luồng hoạt động UC-10

Quy trình dành cho Admin để quản trị hệ thống người dùng. Admin thực hiện các thao tác quản trị đối với tài khoản của nhân viên và khách hàng.

- **Chức năng:** Tạo mới, Chính sửa thông tin, hoặc Xóa người dùng.
- **Ràng buộc bảo mật:**

– Nếu Admin cố gắng tự xóa tài khoản của chính mình (Delete self), hệ thống sẽ chặn hành động này để tránh mất quyền quản trị.

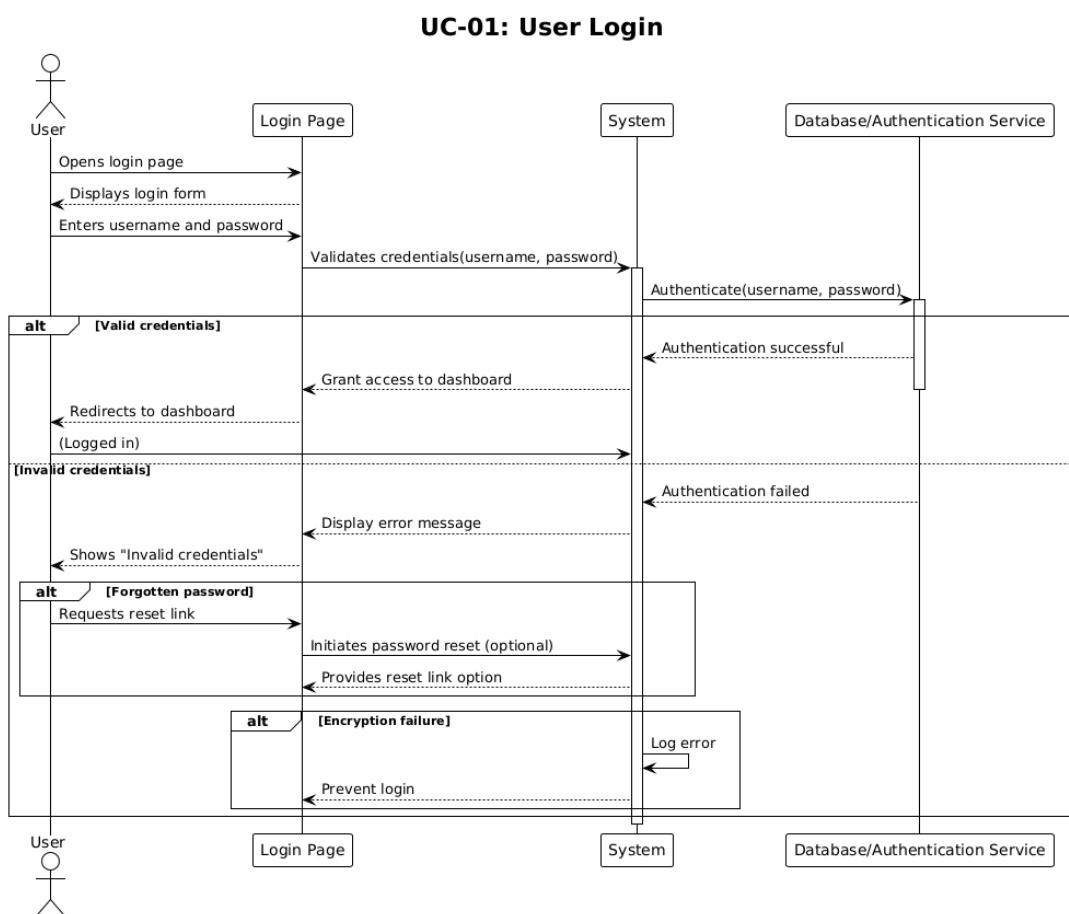
3.7. Activity Diagram

- Nếu tài khoản cần xóa đang có dữ liệu ràng buộc (ví dụ: nhân viên đang phụ trách đơn hàng), hệ thống sẽ hiển thị thông báo không thể xóa do ràng buộc dữ liệu.

3.8 Sequence Diagram

Biểu đồ tuần tự (Sequence Diagram) mô tả chi tiết sự tương tác theo thời gian giữa các đối tượng trong hệ thống (Actor, UI, System/Controller, Service, Database) để thực hiện một chức năng cụ thể.

3.8.1 UC-01: Đăng nhập (User Login)



Hình 3.8.1: Sequence Diagram - Đăng nhập

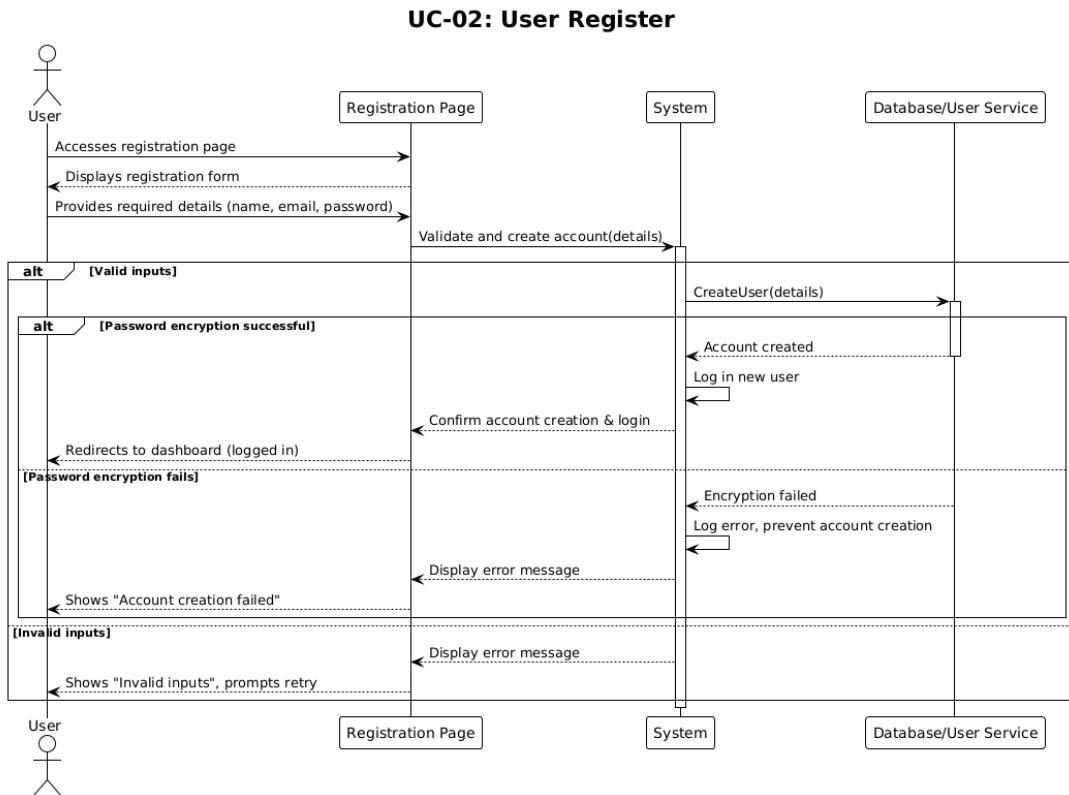
Quy trình xác thực người dùng bao gồm việc tương tác giữa người dùng, trang đăng nhập và dịch vụ xác thực.

- **Khởi tạo:** Người dùng (User) nhập tên đăng nhập và mật khẩu tại *Login Page*.
- **Xử lý:** Hệ thống gửi thông tin đến *Database/Authentication Service* để đổi chiều.
- **Các kịch bản:**

3.8. Sequence Diagram

- *Valid credentials*: Nếu thông tin đúng, hệ thống cấp quyền truy cập và chuyển hướng về Dashboard.
- *Invalid credentials*: Nếu sai, hệ thống trả về thông báo lỗi.
- *Encryption failure*: Nếu quá trình mã hóa gặp lỗi kỹ thuật, hệ thống ghi log và ngăn chặn đăng nhập.
- *Forgotten password*: Người dùng có thể yêu cầu gửi liên kết đặt lại mật khẩu.

3.8.2 UC-02: Đăng ký (User Register)



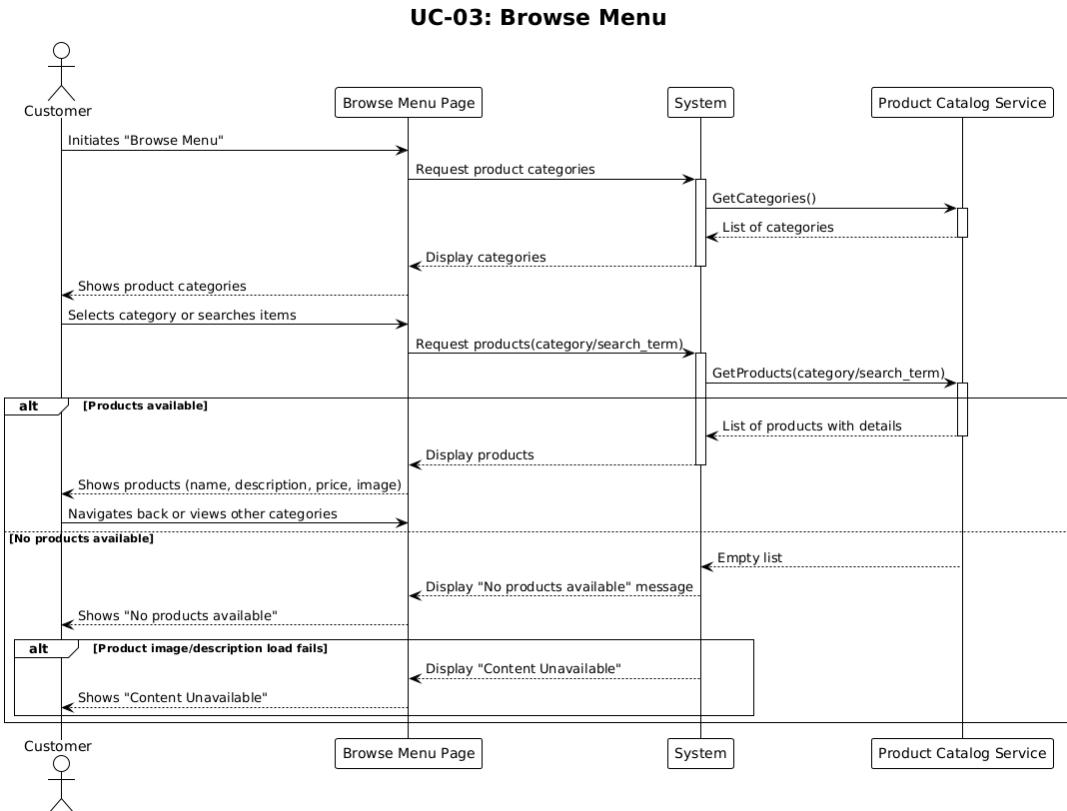
Hình 3.8.2: Sequence Diagram - Đăng ký tài khoản

Mô tả trình tự tạo tài khoản mới và lưu trữ vào cơ sở dữ liệu.

- **Khởi tạo:** Người dùng nhập thông tin cá nhân (tên, email, mật khẩu) tại *Registration Page*.
- **Xử lý:** Hệ thống xác thực dữ liệu đầu vào.
- **Các kịch bản:**
 - *Valid inputs:* Nếu dữ liệu hợp lệ và mã hóa mật khẩu thành công, *Database/User Service* tạo bản ghi mới. Người dùng được tự động đăng nhập.
 - *Encryption fails:* Nếu lỗi mã hóa, hệ thống ghi log và hủy bỏ thao tác tạo tài khoản.
 - *Invalid inputs:* Hệ thống báo lỗi định dạng ngay lập tức và yêu cầu nhập lại.

3.8. Sequence Diagram

3.8.3 UC-03: Duyệt Menu (Browse Menu)

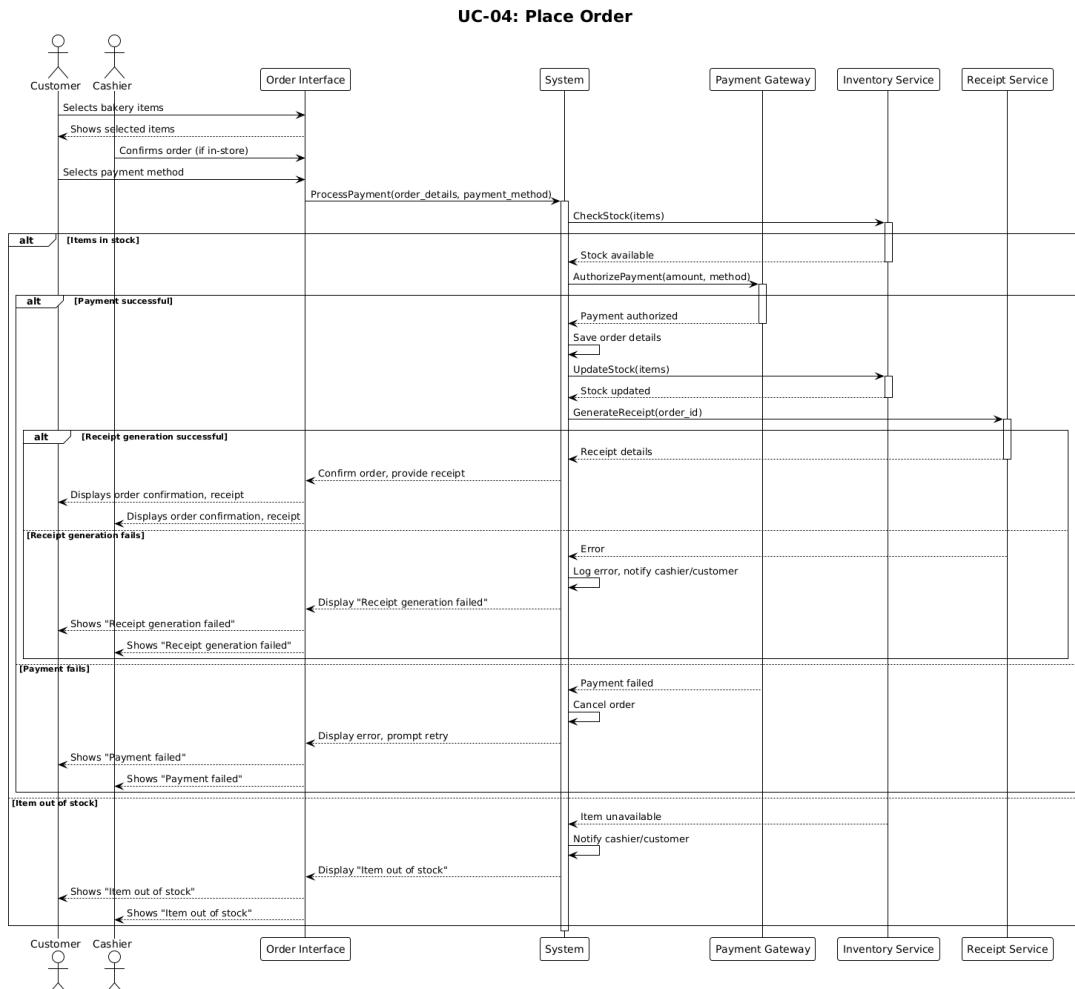


Hình 3.8.3: Sequence Diagram - Duyệt Menu

Mô tả cách hệ thống lấy dữ liệu sản phẩm để hiển thị cho khách hàng.

- **Luồng chính:** Khách hàng truy cập trang Menu. Hệ thống gọi `GetCategories()` và `GetProducts()` từ `Product Catalog Service`.
- **Phản hồi:**
 - Nếu có sản phẩm (*Products available*), hệ thống hiển thị danh sách chi tiết (tên, giá, hình ảnh).
 - Nếu không tìm thấy (*No products available*), hệ thống hiển thị thông báo trống.
 - Nếu tải hình ảnh thất bại, giao diện hiển thị thông báo "Content Unavailable".

3.8.4 UC-04: Đặt hàng (Place Order)



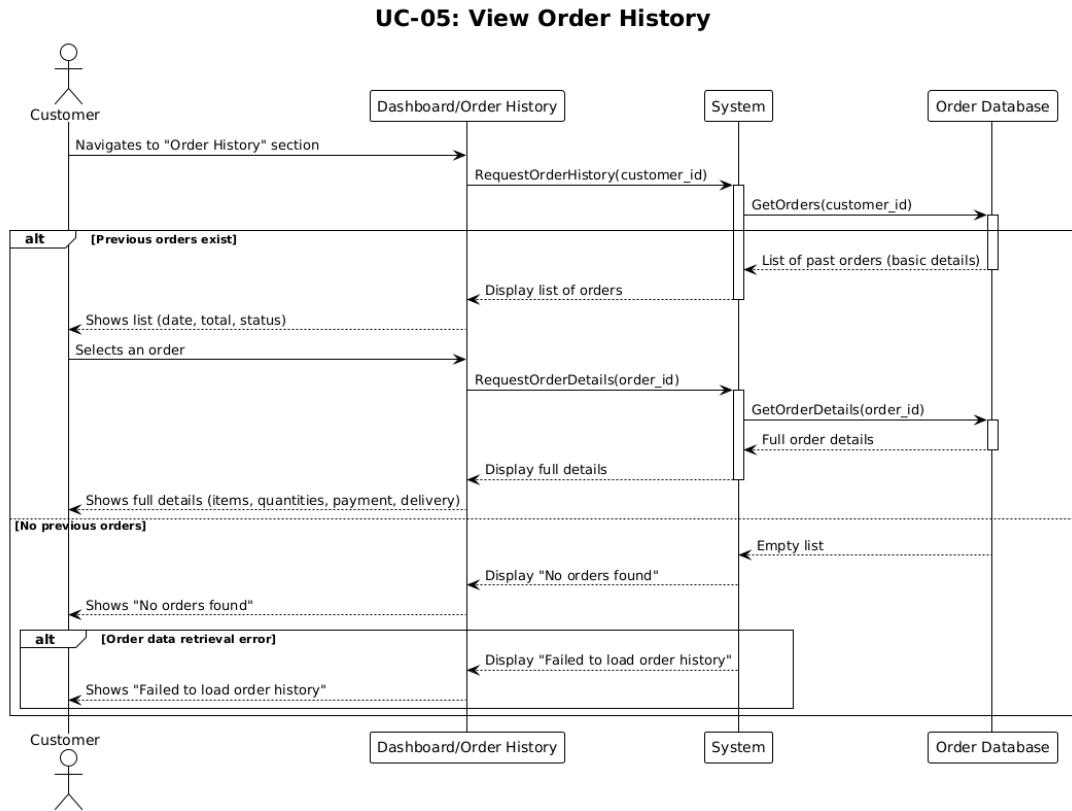
Hình 3.8.4: Sequence Diagram - Đặt hàng

Đây là quy trình phức tạp nhất, liên kết nhiều dịch vụ: Kho, Thanh toán và Hóa đơn.

- **Kiểm tra kho:** Khi khách chọn món, hệ thống gọi *CheckStock()* tới *Inventory Service*. Nếu hết hàng (*Item out of stock*), quy trình dừng lại và báo lỗi.
- **Thanh toán:** Nếu còn hàng, hệ thống gọi *Payment Gateway*.
 - Nếu thanh toán thành công: Lưu đơn hàng, gọi *UpdateStock()* để trừ kho, và gọi *GenerateReceipt()* từ *Receipt Service*.
 - Nếu thanh toán thất bại: Hủy đơn và yêu cầu thử lại.
- **Xuất hóa đơn:** Nếu tạo hóa đơn lỗi, hệ thống vẫn ghi nhận đơn hàng thành công nhưng ghi log lỗi hóa đơn.

3.8. Sequence Diagram

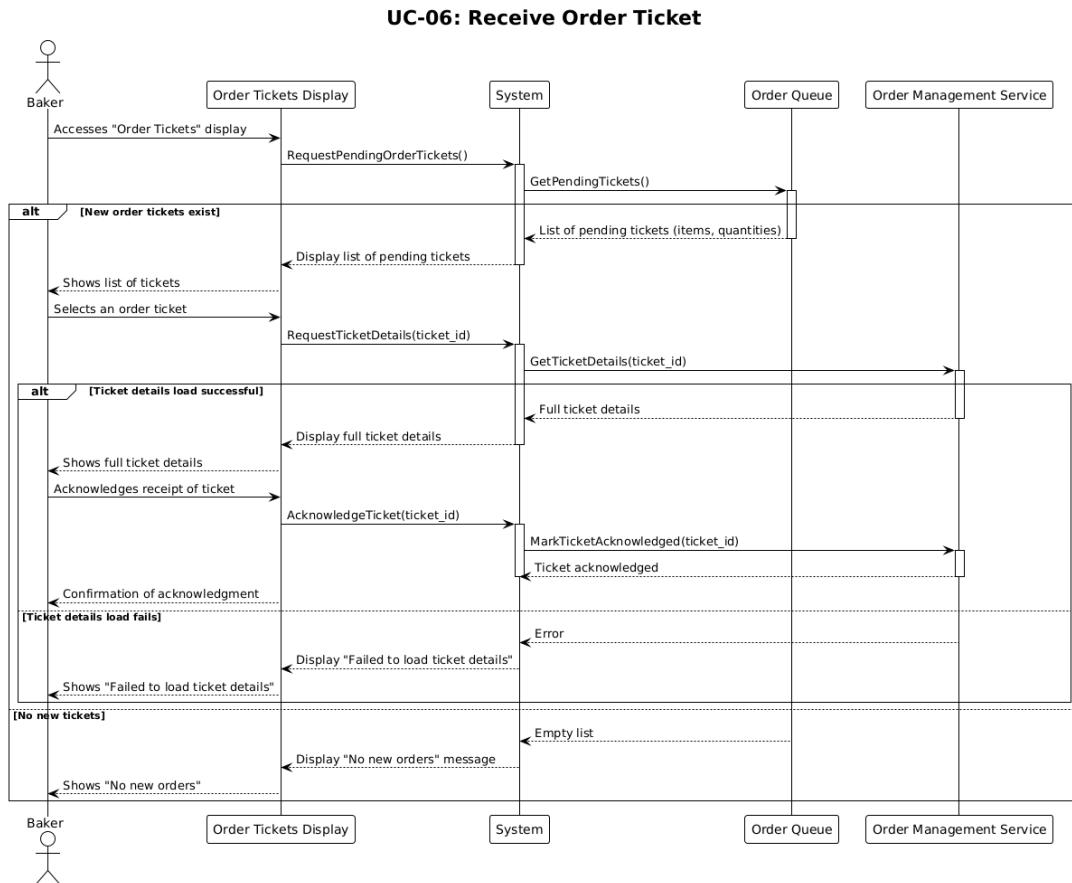
3.8.5 UC-05: Xem lịch sử đơn hàng (View Order History)



Hình 3.8.5: Sequence Diagram - Xem lịch sử đơn hàng

- **Luồng chính:** Khách hàng yêu cầu xem lịch sử. Hệ thống gửi *RequestOrderHistory()* tới *Order Database*.
- **Chi tiết đơn hàng:** Khi khách chọn một đơn cụ thể, hệ thống gửi *RequestOrderDetails()* để lấy thông tin chi tiết.
- **Xử lý lỗi:** Nếu xảy ra lỗi truy vấn (*Order data retrieval error*), hệ thống hiển thị thông báo lỗi thay vì dữ liệu.

3.8.6 UC-06: Nhận phiếu order (Receive Order Ticket)



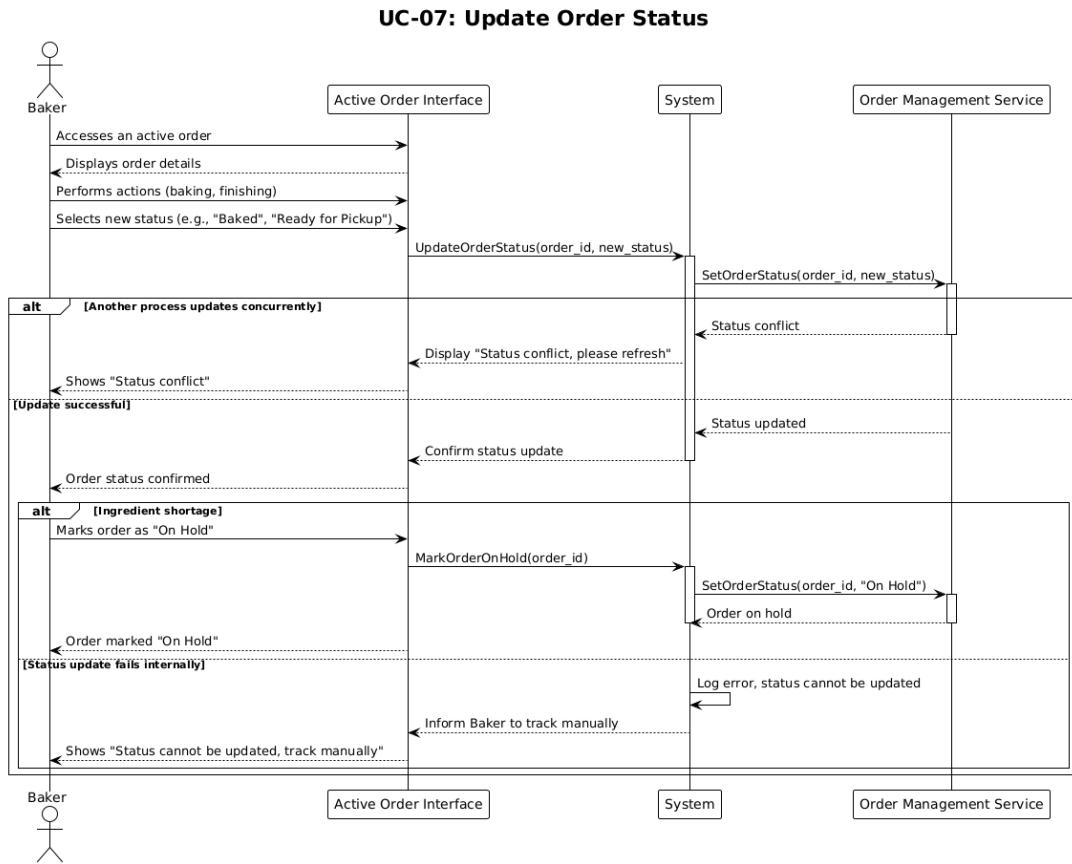
Hình 3.8.6: Sequence Diagram - Nhận phiếu order

Quy trình đồng bộ thông tin giữa Bếp (Baker) và Hệ thống quản lý đơn hàng.

- Kiểm tra:** Hệ thống định kỳ hoặc theo sự kiện kiểm tra *Order Queue*.
- Hiển thị:** Nếu có đơn mới (*New order tickets exist*), danh sách được tải về giao diện của Baker.
- Xác nhận:** Baker chọn xem chi tiết và xác nhận (*Acknowledge*). Hệ thống gọi *MarkTicketAcknowledged()* tới Service để cập nhật trạng thái đã nhận.

3.8. Sequence Diagram

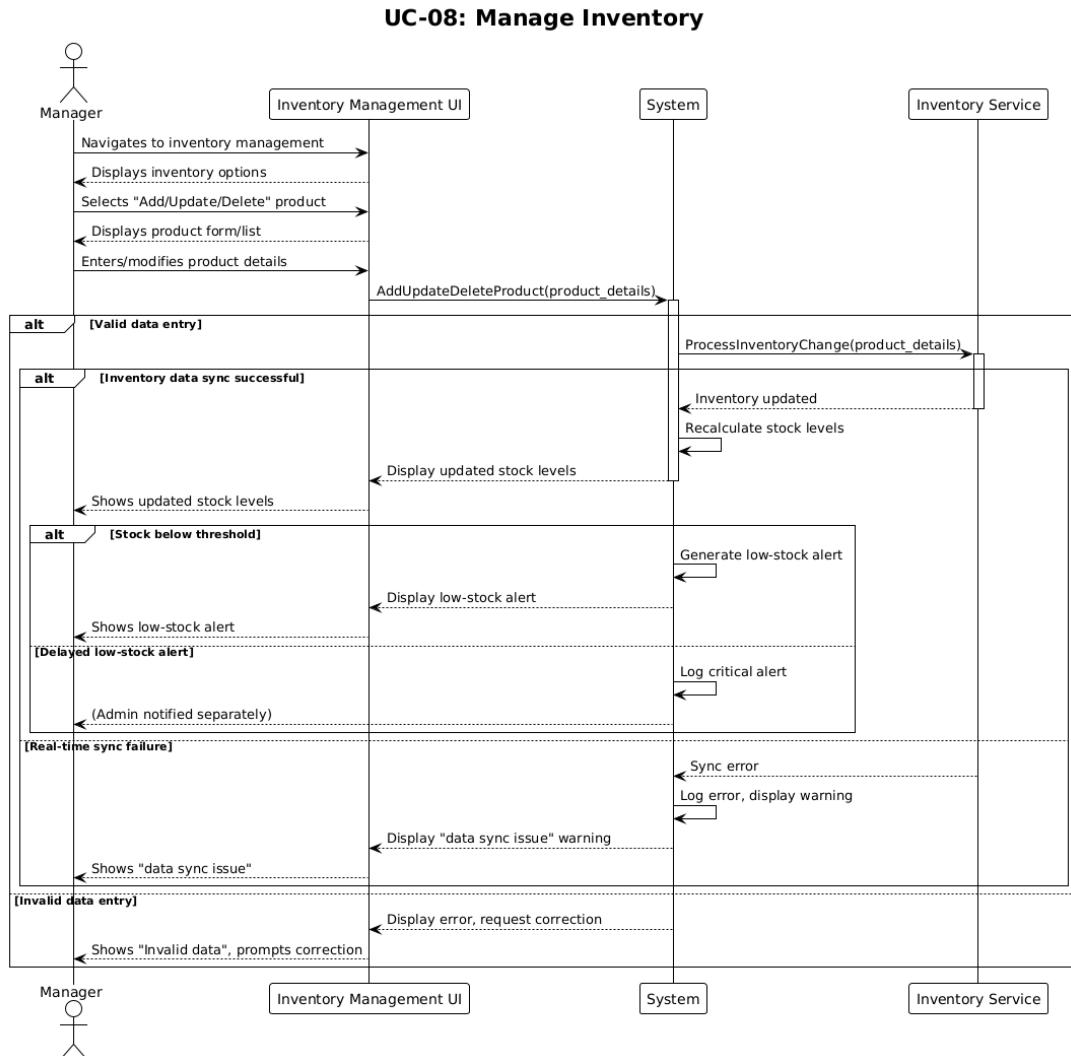
3.8.7 UC-07: Cập nhật trạng thái đơn hàng (Update Order Status)



Hình 3.8.7: Sequence Diagram - Cập nhật trạng thái đơn hàng

- **Hành động:** Baker cập nhật trạng thái (ví dụ: "Ready"). Hệ thống gọi *UpdateOrderStatus()*.
- **Xử lý đồng thời (Concurrency):** Hệ thống kiểm tra xem có tiến trình nào khác đang cập nhật cùng đơn hàng không. Nếu có xung đột, hệ thống báo lỗi yêu cầu làm mới.
- **Ngoại lệ:** Nếu thiếu nguyên liệu (*Ingredient shortage*), Baker đánh dấu đơn là "On Hold".

3.8.8 UC-08: Quản lý tồn kho (Manage Inventory)

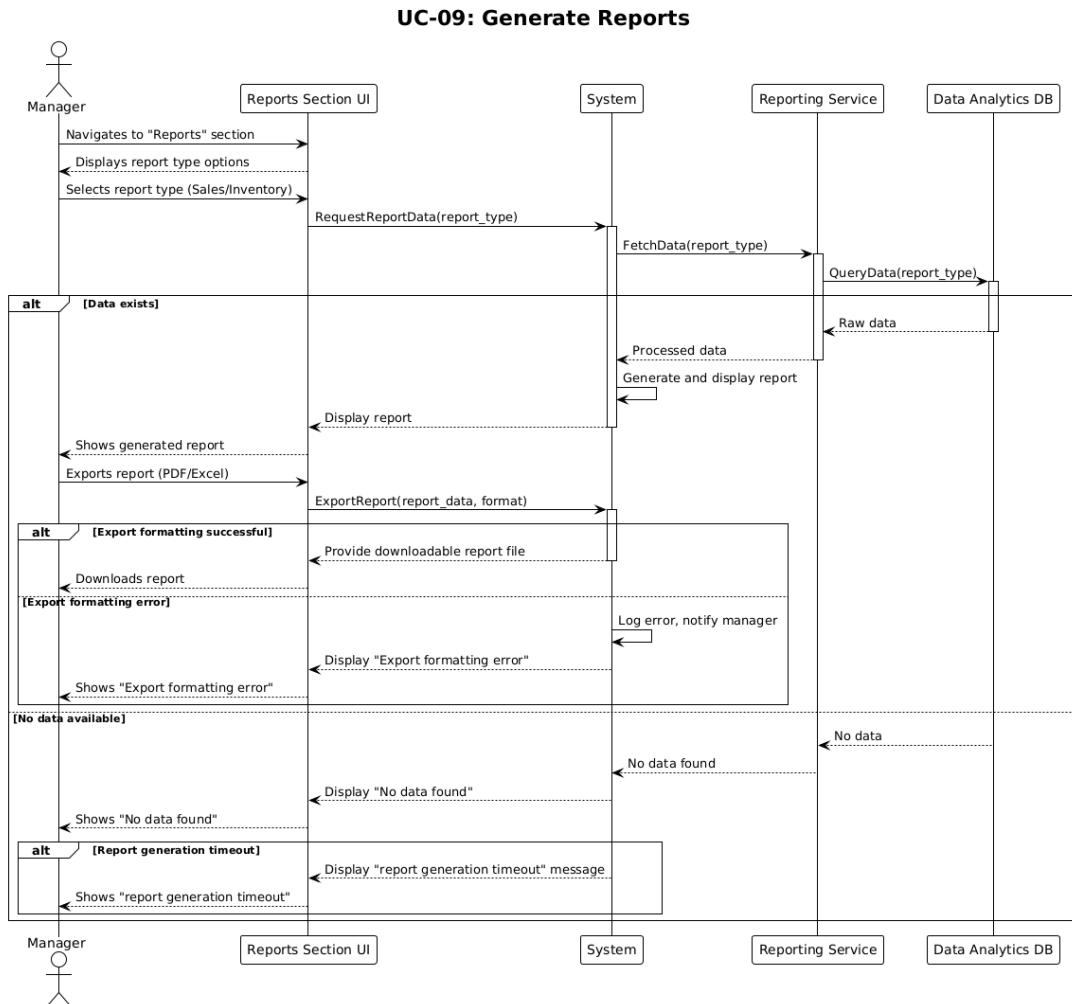


Hình 3.8.8: Sequence Diagram - Quản lý tồn kho

- Cập nhật:** Quản lý thêm/sửa sản phẩm. Hệ thống gọi *ProcessInventoryChange()* tới *Inventory Service*.
- Cảnh báo:** Sau khi cập nhật, hệ thống tính toán lại mức tồn kho. Nếu dưới ngưỡng quy định (*Stock below threshold*), hệ thống tự động tạo cảnh báo (*Generate low-stock alert*).
- Đồng bộ:** Nếu việc đồng bộ dữ liệu thất bại (*Real-time sync failure*), hệ thống hiển thị cảnh báo cho quản lý.

3.8. Sequence Diagram

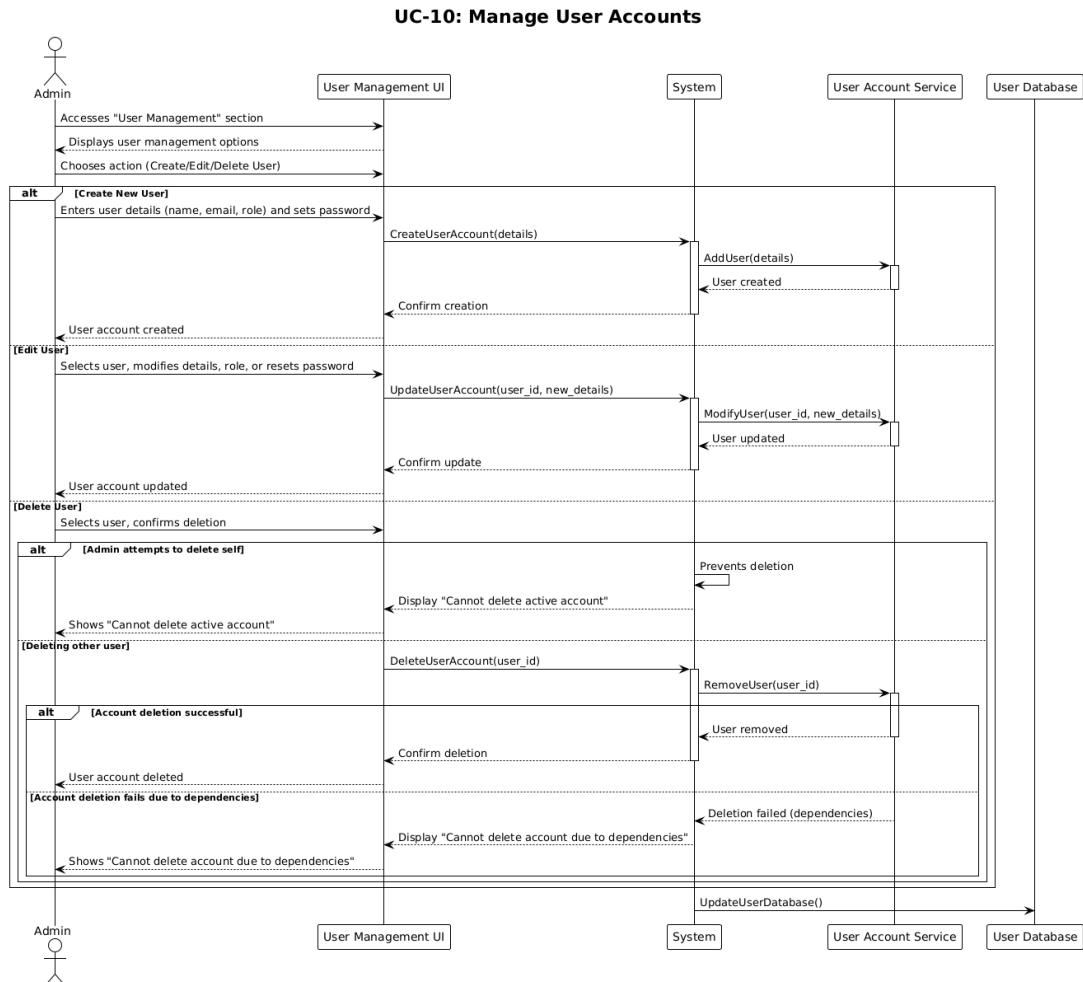
3.8.9 UC-09: Tạo báo cáo (Generate Reports)



Hình 3.8.9: Sequence Diagram - Tạo báo cáo

- **Truy xuất dữ liệu:** Quản lý chọn loại báo cáo. Hệ thống gọi *FetchData()* từ *Reporting Service* và truy vấn *Data Analytics DB*.
- **Xuất file:** Nếu dữ liệu tồn tại, hệ thống xử lý và cho phép xuất ra PDF/Excel.
- **Xử lý lỗi:** Các trường hợp như không có dữ liệu (*No data found*), lỗi định dạng khi xuất file (*Export formatting error*), hoặc quá thời gian chờ (*Timeout*) đều được xử lý với thông báo cụ thể.

3.8.10 UC-10: Quản lý tài khoản (Manage User Accounts)



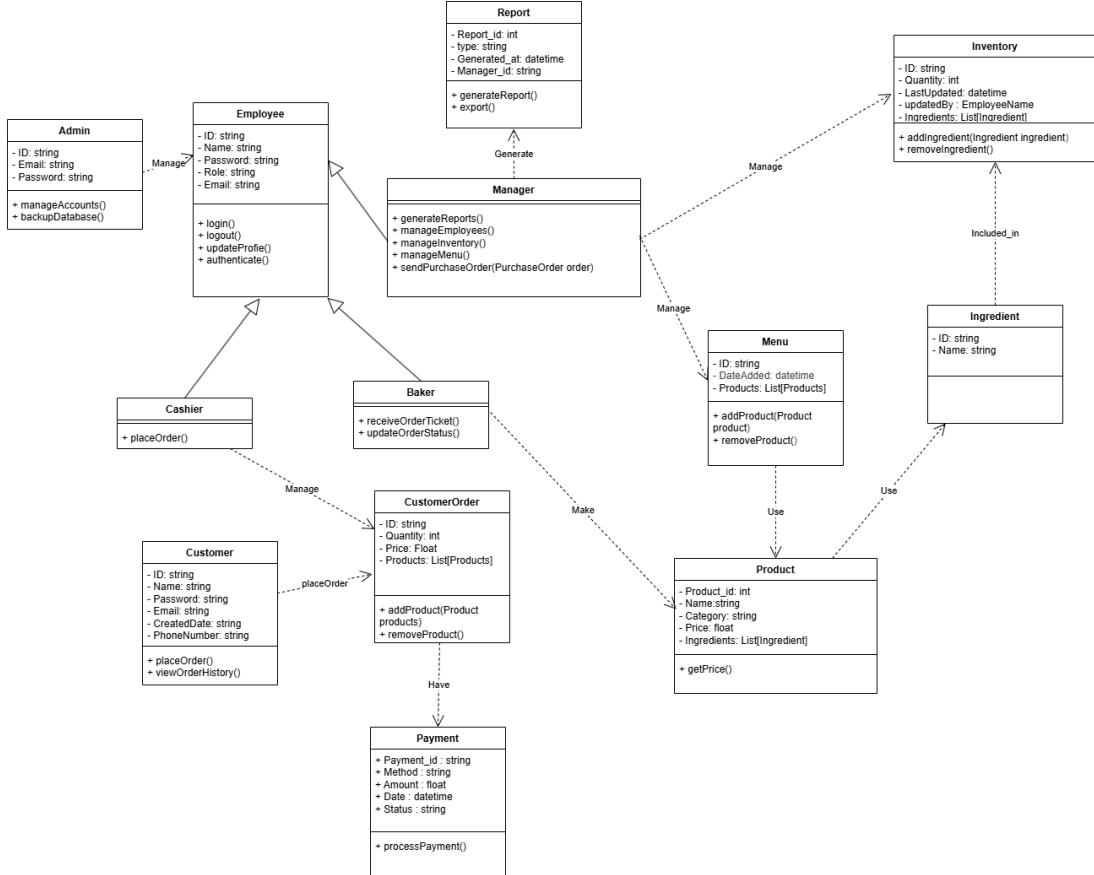
Hình 3.8.10: Sequence Diagram - Quản lý tài khoản người dùng

Mô tả các thao tác CRUD (Tạo, Sửa, Xóa) của Admin đối với tài khoản người dùng.

- **Create/Edit:** Admin nhập liệu, hệ thống gọi *AddUser()* hoặc *ModifyUser()* tới *User Account Service*.
- **Delete:**
 - Nếu Admin xóa chính mình: Hệ thống chặn (*Prevents deletion*).
 - Nếu xóa người dùng khác: Hệ thống gọi *RemoveUser()*.
 - Nếu có ràng buộc dữ liệu (ví dụ: User đang có đơn hàng chưa xong): Hệ thống báo lỗi "Cannot delete account due to dependencies".

3.9. Class Diagram

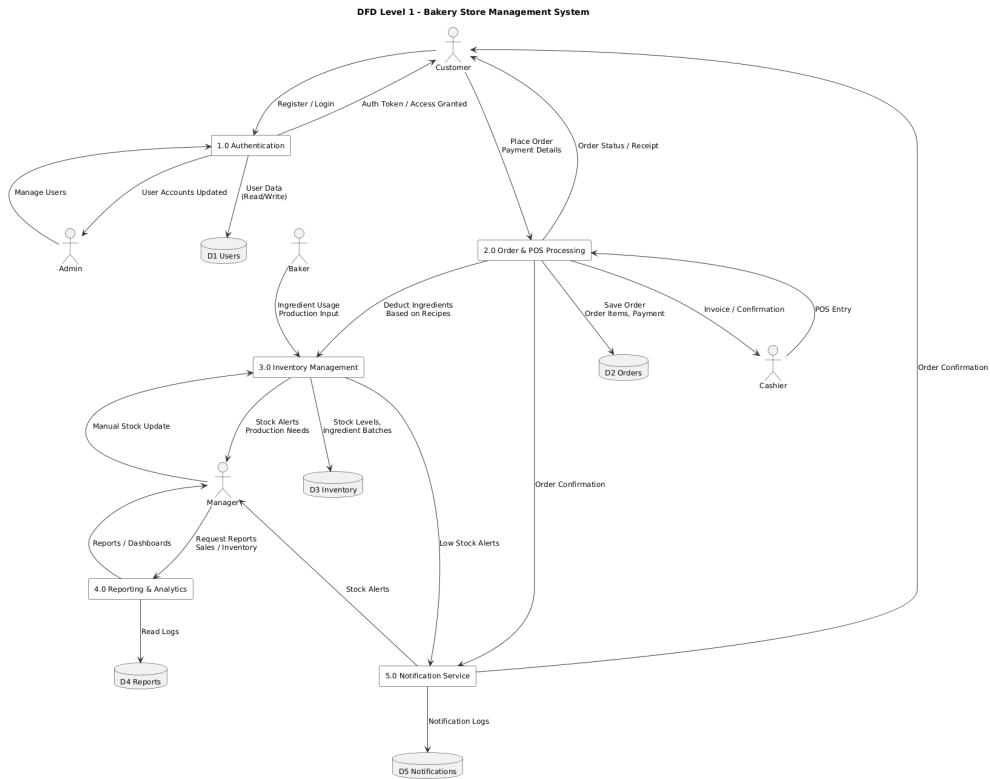
3.9 Class Diagram



Hình 3.9.1: Biểu đồ lớp (Class Diagram) của hệ thống

Biểu đồ lớp mô tả cấu trúc tổng thể của hệ thống, bao gồm các lớp chính như *Customer*, *Employee*, *Manager*, *Admin*, *Cashier*, *Baker*, *Product*, *Ingredient*, *Inventory*, *Menu*, *Order* và *Payment*. Mỗi lớp thể hiện các thuộc tính dữ liệu và các phương thức xử lý tương ứng. Các mối quan hệ giữa các lớp thể hiện sự kế thừa, kết hợp và phụ thuộc, giúp mô tả rõ cách các thành phần trong hệ thống tương tác với nhau.

3.10 Data Flow Diagram (DFD)



Hình 3.10.1: DFD cấp 1 của hệ thống

Biểu đồ luồng dữ liệu cấp 1 (DFD Level 1) chi tiết hóa các quy trình xử lý chính và dòng chảy thông tin trong hệ thống quản lý tiệm bánh. Hệ thống được chia thành 5 tiến trình (process) trung tâm tương tác với các kho dữ liệu (data stores) và các tác nhân (actors):

- 1.0 Authentication (Xác thực):** Nhận thông tin đăng ký/đăng nhập từ người dùng, đối chiếu với kho dữ liệu *D1 Users* để cấp quyền truy cập (Auth Token).
- 2.0 Order and POS Processing (Xử lý đơn hàng và POS):** Là trung tâm của hoạt động kinh doanh. Nhận yêu cầu đặt hàng từ khách hàng hoặc nhân viên thu ngân, lưu thông tin vào *D2 Orders*, đồng thời kích hoạt quy trình thanh toán và gửi yêu cầu cập nhật kho.
- 3.0 Inventory Management (Quản lý kho):** Tương tác với Thợ làm bánh (Baker) và Quản lý (Manager). Hệ thống tự động trừ nguyên liệu dựa trên công thức khi có đơn hàng, cập nhật mức tồn kho vào *D3 Inventory* và phát cảnh báo nhập hàng khi nguyên liệu sắp hết.

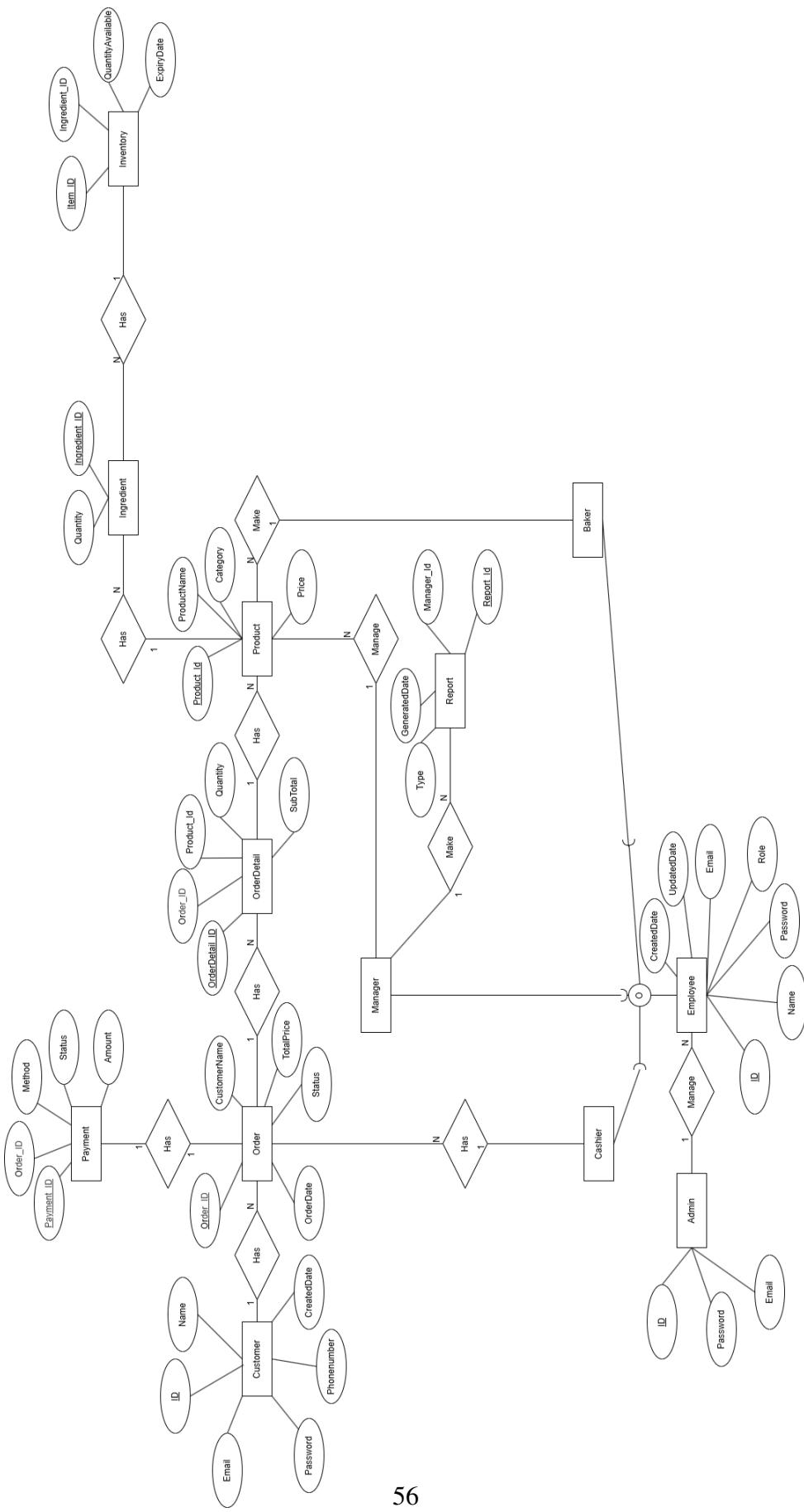
3.10. Data Flow Diagram (DFD)

- **4.0 Reporting and Analytics (Báo cáo và Phân tích):** Tổng hợp dữ liệu từ các bản ghi (logs) và doanh số để xuất ra các báo cáo doanh thu, tồn kho lưu tại *D4 Reports* phục vụ cho Quản lý.
- **5.0 Notification Service (Dịch vụ thông báo):** Nhận tín hiệu từ các quy trình khác (như đơn hàng thành công, cảnh báo tồn kho) để gửi thông báo đến người dùng hoặc nhân viên, lưu vết tại *D5 Notifications*.

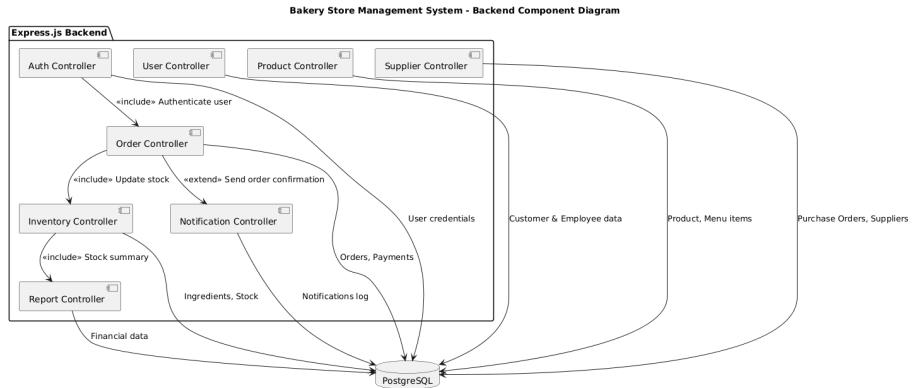
3.11 ERD (Entity Relationship Diagram)

Sơ đồ ERD mô tả mô hình dữ liệu của hệ thống, bao gồm các thực thể như *Customer*, *Order*, *OrderDetail*, *Product*, *Ingredient*, *Inventory*, *Payment* và *Report*. Các quan hệ một-nhiều và nhiều-nhiều giữa các thực thể hiện cách dữ liệu được tổ chức trong hệ quản trị cơ sở dữ liệu. ERD giúp đảm bảo tính nhất quán của dữ liệu và là cơ sở để xây dựng các bảng trong PostgreSQL.

3.11. ERD (Entity Relationship Diagram)



3.12 Component Diagram



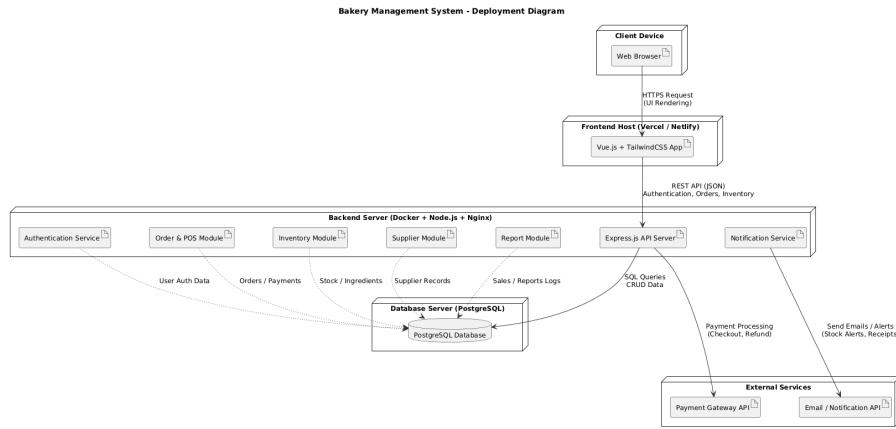
Hình 3.12.1: Component Diagram của hệ thống

Biểu đồ thành phần tập trung vào kiến trúc nội bộ của phía Backend (máy chủ), được xây dựng trên nền tảng Node.js với Express.js. Hệ thống được module hóa thành các *Controller* riêng biệt để đảm bảo tính độc lập và dễ bảo trì:

- **Controller Layer:** Bao gồm các thành phần xử lý nghiệp vụ như *Auth Controller* (xác thực), *Order Controller* (xử lý đơn hàng), *Inventory Controller* (kho), và *Report Controller* (báo cáo).
- **Mối quan hệ phụ thuộc:**
 - *Order Controller* có mối quan hệ «include» với *Inventory Controller* (để cập nhật kho sau khi bán) và «extend» với *Notification Controller* (để gửi xác nhận đơn hàng).
 - *Inventory Controller* cung cấp dữ liệu tóm tắt kho cho *Report Controller*.
- **Data Access:** Tất cả các Controller đều kết nối tập trung về cơ sở dữ liệu PostgreSQL để thực hiện các thao tác CRUD (Thêm, Đọc, Sửa, Xóa) dữ liệu liên quan đến người dùng, sản phẩm, và tài chính.

3.13. Deployment Diagram

3.13 Deployment Diagram



Hình 3.13.1: Deployment Diagram – Mô hình triển khai hệ thống

Biểu đồ triển khai mô tả cấu hình vật lý và môi trường vận hành của hệ thống, tuân theo kiến trúc Client-Server hiện đại:

- **Client Device:** Người dùng cuối (Khách hàng, Nhân viên) truy cập hệ thống thông qua trình duyệt web.
- **Frontend Host:** Ứng dụng giao diện (Vue.js + TailwindCSS) được lưu trữ (host) trên các nền tảng đám mây như Vercel hoặc Netlify, chịu trách nhiệm render giao diện và gửi HTTPS Request.
- **Backend Server:** Máy chủ ứng dụng chạy trong môi trường Docker container với Node.js và Nginx làm reverse proxy. Tại đây, *Express.js API Server* tiếp nhận các yêu cầu REST API (JSON) từ Frontend.
- **Database Server:** Sử dụng hệ quản trị cơ sở dữ liệu PostgreSQL để lưu trữ bền vững toàn bộ dữ liệu nghiệp vụ. Backend kết nối với Database thông qua các câu truy vấn SQL.
- **External Services:** Hệ thống tích hợp với các dịch vụ bên thứ ba:
 - *Payment Gateway API*: Xử lý thanh toán trực tuyến và hoàn tiền.
 - *Email / Notification API*: Gửi email xác nhận và cảnh báo hệ thống.

Chương 4

Kết quả thiết kế giao diện

Sau quá trình phân tích nghiệp vụ, nhóm đã hoàn thiện bản thiết kế giao diện chi tiết (High-fidelity Prototype) cho hệ thống "Matcha Bakery Management". Giao diện được thiết kế theo phong cách hiện đại, tối giản, lấy tông màu xanh lá (Matcha Green) làm chủ đạo để tạo dấu ấn thương hiệu mạnh mẽ, mang lại cảm giác tươi mới và sạch sẽ phù hợp với ngành thực phẩm.

Chương này sẽ trình bày chi tiết các màn hình chức năng đã được hiện thực hóa trên công cụ Figma.

4.1 Phân hệ Khách hàng (Public & Customer Portal)

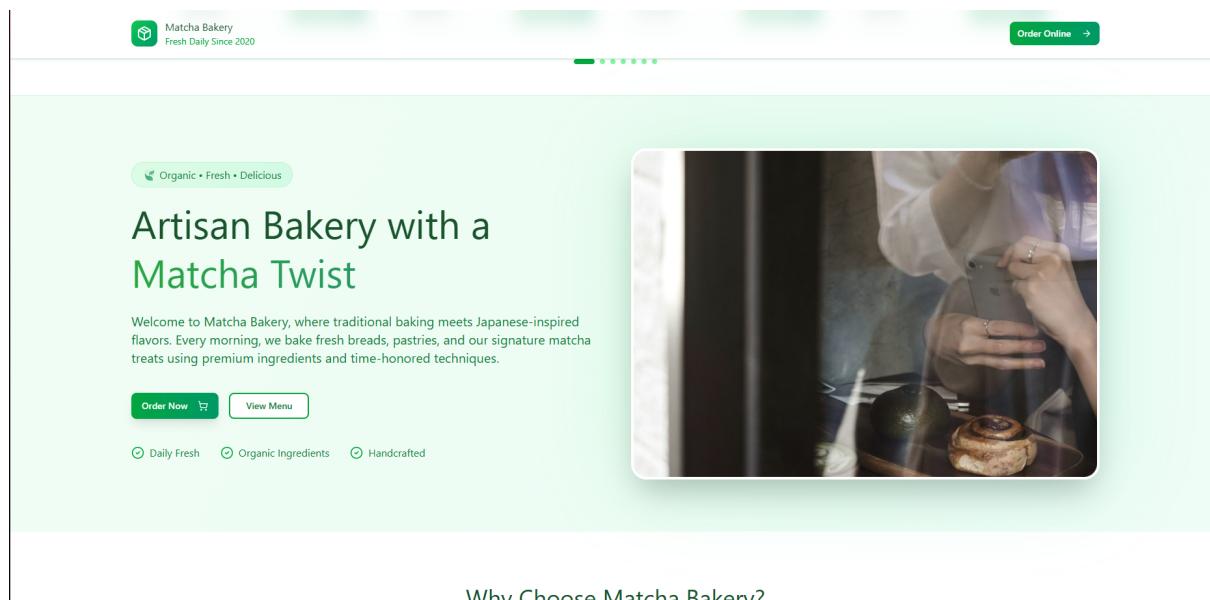
Đây là cổng giao tiếp chính với khách hàng, nơi thể hiện hình ảnh thương hiệu và thúc đẩy hành vi mua hàng.

4.1. Phân hệ Khách hàng (Public & Customer Portal)

4.1.1 Trang chủ và Giới thiệu (Landing Page)

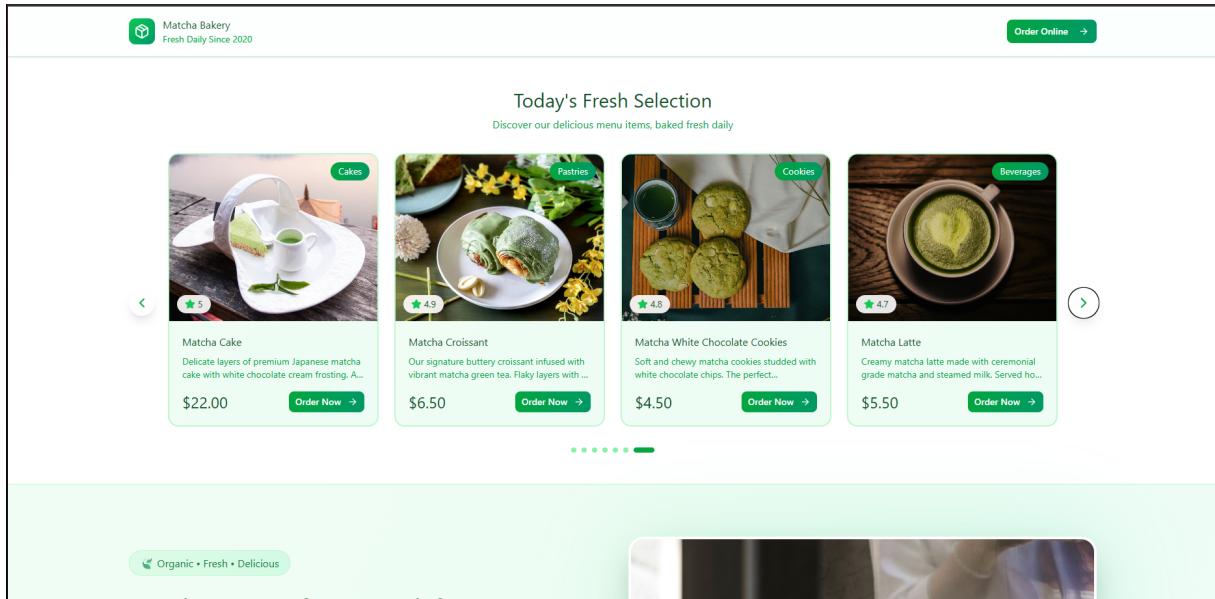
Trang chủ được thiết kế nhằm gây ấn tượng thị giác ngay lập tức với các banner khổ lớn và hình ảnh sản phẩm chất lượng cao.

- Banner Section:** Sử dụng hình ảnh "Artisan Bakery" với lời chào mừng và nút kêu gọi hành động (CTA) "Order Now" nổi bật.
- Sản phẩm trong ngày:** Slider trình diễn các món bánh mới nhất (Cakes, Pastries, Cookies) giúp khách hàng dễ dàng lựa chọn.
- Thông tin cửa hàng (Store Info):** Mục "Baked Fresh Every Morning" kết hợp khéo léo giữa câu chuyện thương hiệu về quy trình làm bánh tươi mỗi ngày và các thông tin liên hệ thiết yếu. Bốn thẻ thông tin (Cards) với icon trực quan cung cấp nhanh: Giờ mở cửa (6 AM - 8 PM), Địa điểm, Hotline và Email, giúp khách hàng dễ dàng kết nối mà không cần tìm kiếm lâu.



Hình 4.1.1: Giao diện trang chủ với thông điệp chào mừng và hình ảnh chủ đạo

4.1. Phân hệ Khách hàng (Public & Customer Portal)



Hình 4.1.2: Danh sách các sản phẩm nổi bật trong ngày

A screenshot of the Matcha Bakery website featuring a promotional section. The title "Baked Fresh Every Morning" is displayed above a paragraph: "Step into our bakery and experience the aroma of freshly baked bread, the warmth of our ovens, and the smiles of our dedicated team." Below the text are four contact boxes: "Open Daily 6 AM - 8 PM", "Location Downtown District", "Call Us (555) 123-4567", and "Email hello@matcha.cafe". To the right of the text are two images of bread: a loaf on a wooden board and a basket of sliced bread.

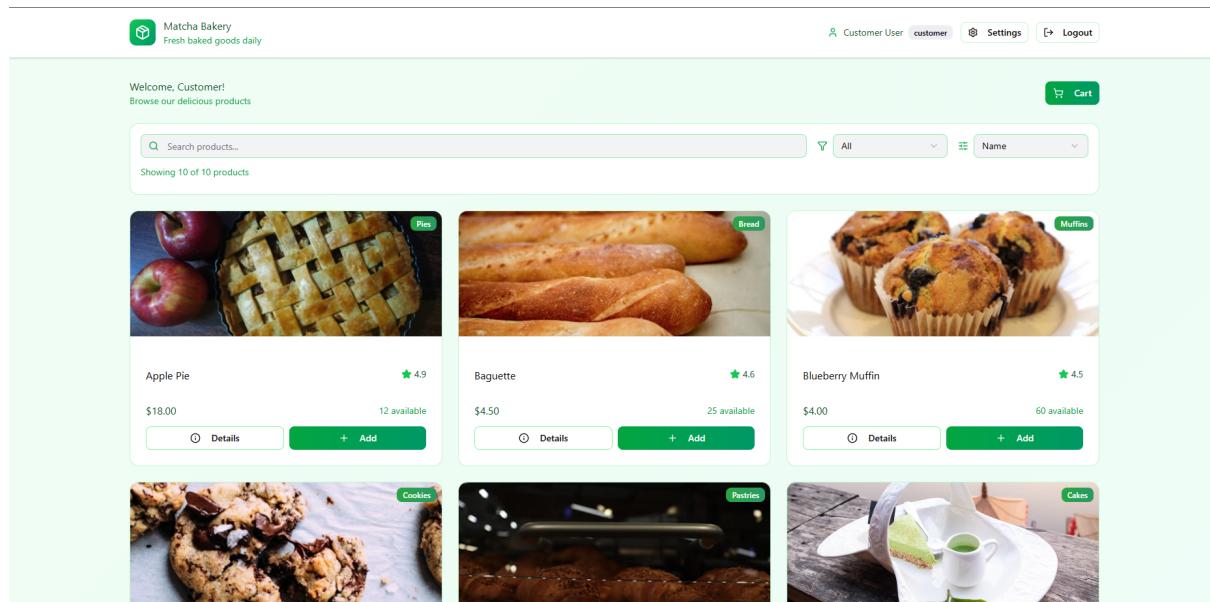
Hình 4.1.3: Phần giới thiệu cam kết chất lượng và thông tin liên hệ nhanh

4.1. Phân hệ Khách hàng (Public & Customer Portal)

4.1.2 Trang danh sách sản phẩm và Chi tiết

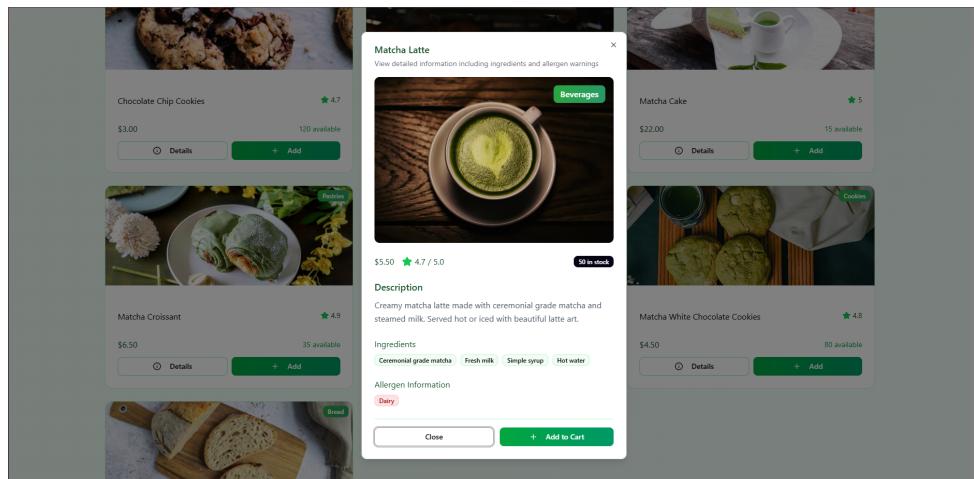
Giao diện mua hàng tập trung vào tính tiện dụng và trực quan.

- **Danh mục sản phẩm:** Hiển thị dạng lưới (Grid layout) với bộ lọc theo loại (Pies, Bread, Muffins) và giá cả. Mỗi thẻ sản phẩm hiển thị rõ tên, giá, đánh giá sao (Rating) và trạng thái tồn kho.
- **Chi tiết sản phẩm (Modal View):** Khi khách hàng chọn một món (ví dụ: Matcha Latte), một cửa sổ nổi (Modal) sẽ hiện ra cung cấp thông tin chi tiết về thành phần (Ingredients), cảnh báo dị ứng (Allergens) và mô tả hương vị, giúp khách hàng yên tâm khi đặt món.



Hình 4.1.4: Giao diện danh sách sản phẩm dành cho khách hàng

4.1. Phân hệ Khách hàng (Public & Customer Portal)



Hình 4.1.5: Cửa sổ chi tiết sản phẩm hiển thị thành phần và cảnh báo dị ứng

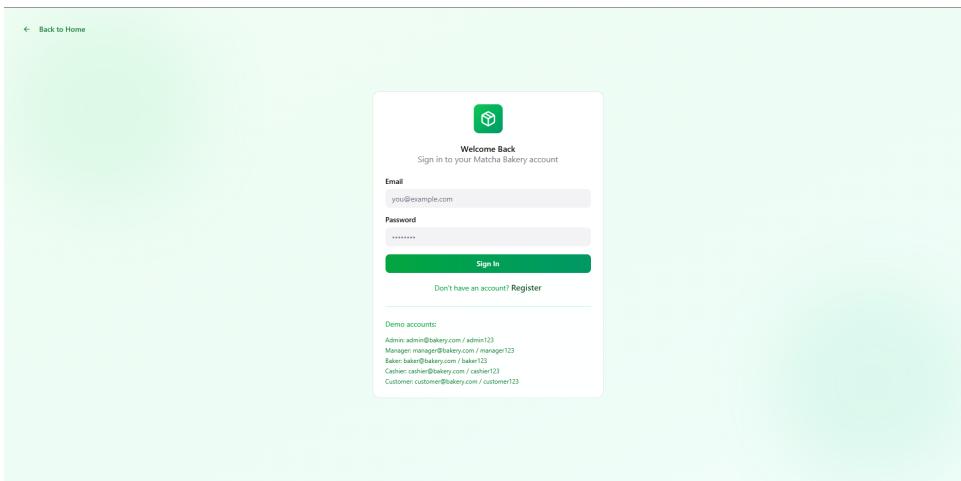
4.2. Phân hệ Xác thực và Tài khoản (Authentication)

4.2 Phân hệ Xác thực và Tài khoản (Authentication)

Đây là cổng kiểm soát truy cập an toàn, đảm bảo phân quyền chính xác giữa Khách hàng, Nhân viên và Quản trị viên.

4.2.1 Đăng nhập (Login)

Giao diện đăng nhập được thiết kế tối giản, loại bỏ các yếu tố gây xao nhãng để người dùng tập trung vào việc nhập thông tin xác thực. Hệ thống hỗ trợ hiển thị gợi ý các tài khoản demo (Admin/Manager/Baker) giúp việc kiểm thử và trải nghiệm sản phẩm thuận tiện hơn.



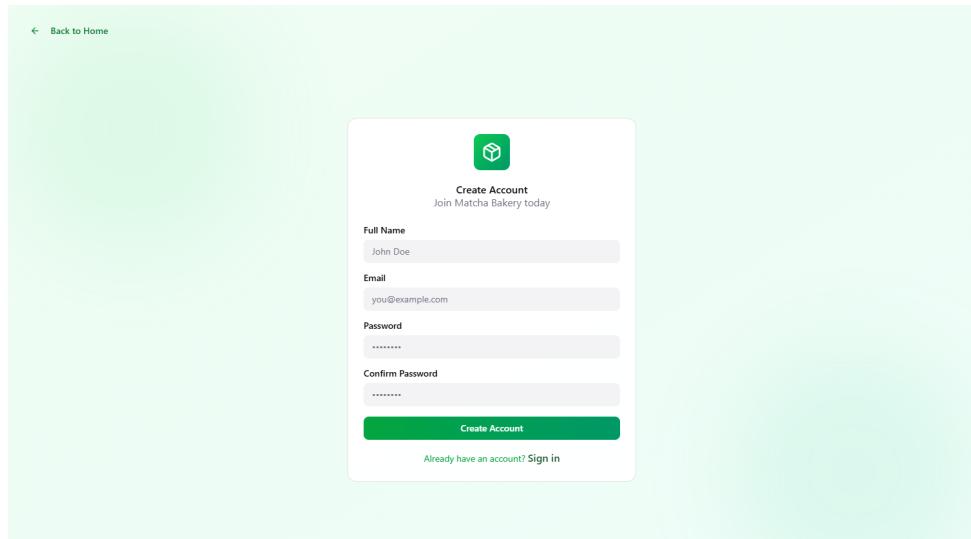
Hình 4.2.1: Màn hình đăng nhập với giao diện tối giản và thông tin demo

4.2. Phân hệ Xác thực và Tài khoản (Authentication)

4.2.2 Đăng ký tài khoản mới (Register)

Đối với khách hàng mới, hệ thống cung cấp màn hình tạo tài khoản nhanh chóng.

- Form đăng ký:** Yêu cầu các thông tin thiết yếu như Họ tên đầy đủ, Email và Mật khẩu. Trường "Confirm Password"(Xác nhận mật khẩu) giúp giảm thiểu sai sót khi nhập liệu.
- Điều hướng:** Tích hợp các liên kết điền hướng nhanh như "Back to Home"(Về trang chủ) hoặc "Sign in"(Đăng nhập) nếu người dùng đã có tài khoản, tạo sự linh hoạt trong trải nghiệm.
- Thiết kế:** Giữ nguyên ngôn ngữ thiết kế với logo thương hiệu và nút bấm màu xanh chủ đạo, tạo cảm giác tin cậy và đồng bộ.



Hình 4.2.2: Giao diện đăng ký thành viên mới

4.3. Phân hệ Quản trị (Admin Dashboard)

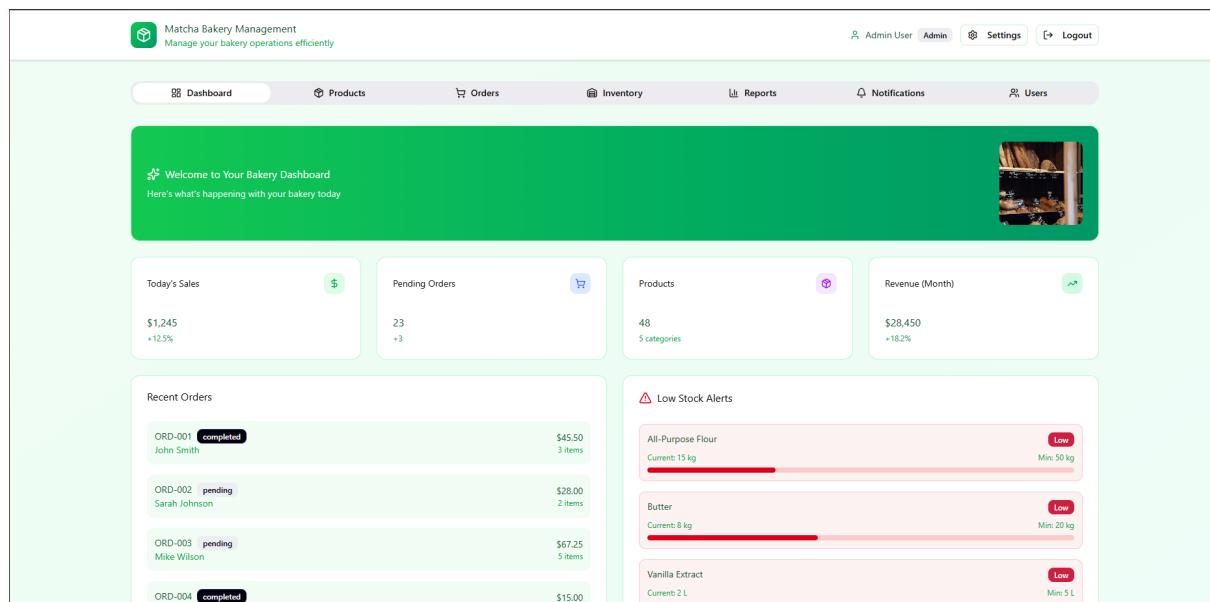
4.3 Phân hệ Quản trị (Admin Dashboard)

Giao diện dành cho Quản lý và Nhân viên được thiết kế tối ưu cho các thao tác nghiệp vụ, sử dụng nhiều biểu đồ và bảng dữ liệu (Data Tables).

4.3.1 Tổng quan hệ thống (Dashboard)

Màn hình Dashboard là trung tâm điều điều hành, cung cấp cái nhìn toàn cảnh về hoạt động kinh doanh theo thời gian thực.

- **Thẻ chỉ số (KPI Cards):** Hiển thị doanh thu trong ngày, số đơn chờ xử lý, tổng sản phẩm.
- **Cảnh báo tồn kho (Low Stock Alerts):** Một panel riêng biệt liệt kê các nguyên liệu đang ở mức báo động (ví dụ: All-Purpose Flour, Butter) giúp quản lý kịp thời nhập hàng.
- **Đơn hàng gần đây:** Danh sách tóm tắt các đơn hàng vừa phát sinh để xử lý nhanh.



Hình 4.3.1: Màn hình Dashboard quản trị với các chỉ số kinh doanh và cảnh báo tồn kho

4.3. Phân hệ Quản trị (Admin Dashboard)

4.3.2 Quản lý Đơn hàng và Người dùng

- Quản lý Đơn hàng (Orders):** Bảng dữ liệu chi tiết với trạng thái được mã hóa màu sắc (Completed - Xanh, Pending - Vàng) giúp nhân viên dễ dàng nhận biết và thao tác (Duyệt/Hủy).
- Quản lý Người dùng (Users):** Giao diện thẻ (Card view) hiển thị danh sách nhân viên và khách hàng, kèm theo vai trò (Admin, Manager, Baker) và trạng thái hoạt động.

The screenshot shows the Matcha Bakery Management Admin Dashboard. At the top, there's a header with the logo 'Matcha Bakery Management' and the tagline 'Manage your bakery operations efficiently'. On the right side of the header are links for 'Admin User', 'Admin', 'Settings', and 'Logout'. Below the header is a navigation bar with tabs: 'Dashboard', 'Products', 'Orders' (which is currently selected), 'Inventory', 'Reports', 'Notifications', and 'Users'. A sub-menu for 'Orders Management' is open, showing a table with three rows of order data. The table columns are: Order ID, Customer, Phone, Items, Total, Date, Status, and Actions. The first row (ORD-001) has a status of 'completed' (green circle) and actions for 'Edit' (pencil) and 'Delete' (trash). The second row (ORD-002) has a status of 'pending' (yellow circle) and actions for 'Edit' (pencil), 'Approve' (green checkmark), and 'Reject' (red X). The third row (ORD-003) also has a 'pending' status and similar actions. A 'New Order' button is located at the top right of the orders table.

Hình 4.3.2: Giao diện quản lý danh sách đơn hàng

4.3. Phân hệ Quản trị (Admin Dashboard)

The screenshot shows the Matcha Bakery Management Admin Dashboard. At the top, there is a header with the logo, the text "Matcha Bakery Management", "Manage your bakery operations efficiently", and navigation links for "Admin User", "Admin", "Settings", and "Logout". Below the header is a navigation bar with links for "Dashboard", "Products", "Orders", "Inventory", "Reports", "Notifications", and "Users". The main content area is titled "User Management" and contains a search bar with placeholder text "Search by name or email...". There is a dropdown menu labeled "All Roles". Below the search bar, there are five user profiles displayed in cards:

| User | Role | Email | Status | Joined | Action Buttons |
|---------------|----------|---------------------|--------|-------------------|---|
| Admin User | admin | admin@bakery.com | active | Joined: 1/15/2024 | Edit Delete |
| John Manager | manager | manager@bakery.com | active | Joined: 2/10/2024 | Edit Delete |
| Sarah Cashier | cashier | cashier@bakery.com | active | Joined: 3/5/2024 | Edit Delete |
| Mike Baker | baker | baker@bakery.com | active | Joined: 3/12/2024 | Edit Delete |
| Emma Customer | customer | customer@bakery.com | active | Joined: 4/20/2024 | Edit Delete |

Hình 4.3.3: Giao diện quản lý tài khoản và phân quyền người dùng

4.3. Phân hệ Quản trị (Admin Dashboard)

4.3.3 Quản lý Kho và Sản phẩm

Đây là chức năng quan trọng nhất để kiểm soát chi phí và nguyên liệu.

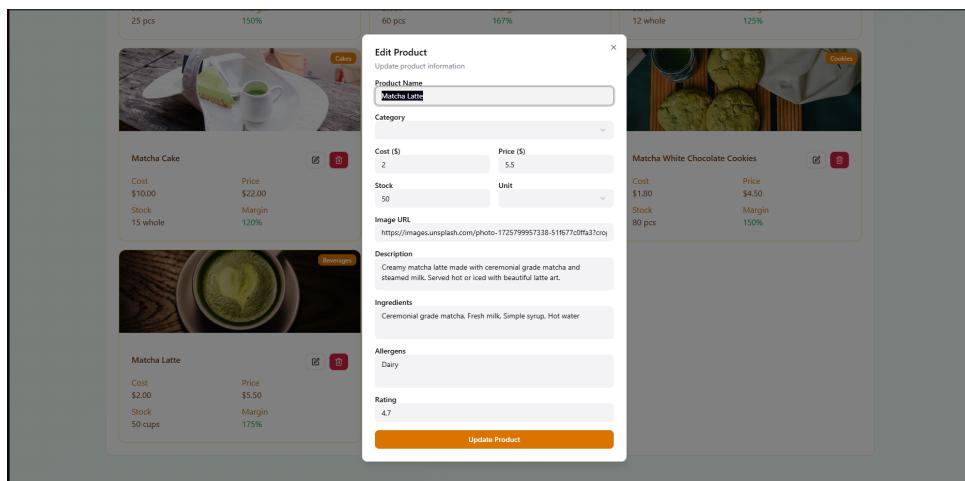
- Danh sách sản phẩm:** Hiển thị hình ảnh, giá vốn (Cost), giá bán (Price) và biên lợi nhuận (Margin) ngay trên thẻ sản phẩm.
- Quản lý tồn kho (Inventory):** Theo dõi chi tiết từng nguyên liệu, định mức tối thiểu/tối đa (Min/Max Stock). Hệ thống tự động đánh dấu trạng thái "Critical"(Nguy cấp) hoặc "Good"(Tốt).
- Chỉnh sửa chi tiết:** Form nhập liệu (Edit Product) cho phép cập nhật thông tin chi tiết, đường dẫn ảnh và công thức định lượng.

The screenshot shows the Matcha Bakery Management Admin Dashboard. At the top, there is a navigation bar with links for Dashboard, Products, Orders, Inventory, Reports, Notifications, and Users. The Inventory tab is active. Below the navigation bar, there are three summary boxes: 'Total Items' (6 categories), 'Low Stock Alerts' (3 items need restocking), and 'Inventory Value' (\$611.50 current stock value). The main area is titled 'Inventory Management' and contains a table of inventory items. The table has columns for Item Name, Category, Current Stock, Min / Max, Unit Cost, Total Value, Last Restocked, Status, and Actions. The items listed are All-Purpose Flour, Butter, Sugar, Eggs, Vanilla Extract, and Chocolate Chips. The status column uses color-coded indicators: red for critical low stock (Butter, Vanilla Extract), green for good stock (Sugar, Chocolate Chips), and yellow for low stock (Eggs). The last column contains edit and delete icons for each item.

| Item Name | Category | Current Stock | Min / Max | Unit Cost | Total Value | Last Restocked | Status | Actions |
|-------------------|------------|---------------|---------------|-----------|-------------|----------------|----------|---------|
| All-Purpose Flour | Flour | 15 kg | 50 / 200 kg | \$2.50 | \$37.50 | 2025-10-28 | Critical | |
| Butter | Dairy | 8 kg | 20 / 50 kg | \$8.00 | \$64.00 | 2025-10-29 | Critical | |
| Sugar | Sweeteners | 75 kg | 30 / 100 kg | \$1.20 | \$90.00 | 2025-10-25 | Good | |
| Eggs | Dairy | 120 pcs | 100 / 300 pcs | \$0.25 | \$30.00 | 2025-10-31 | Low | |
| Vanilla Extract | Flavoring | 2 L | 5 / 15 L | \$45.00 | \$90.00 | 2025-10-20 | Critical | |
| Chocolate Chips | Toppings | 25 kg | 15 / 40 kg | \$12.00 | \$300.00 | 2025-10-30 | Good | |

Hình 4.3.4: Bảng quản lý nguyên liệu và trạng thái tồn kho

4.3. Phân hệ Quản trị (Admin Dashboard)

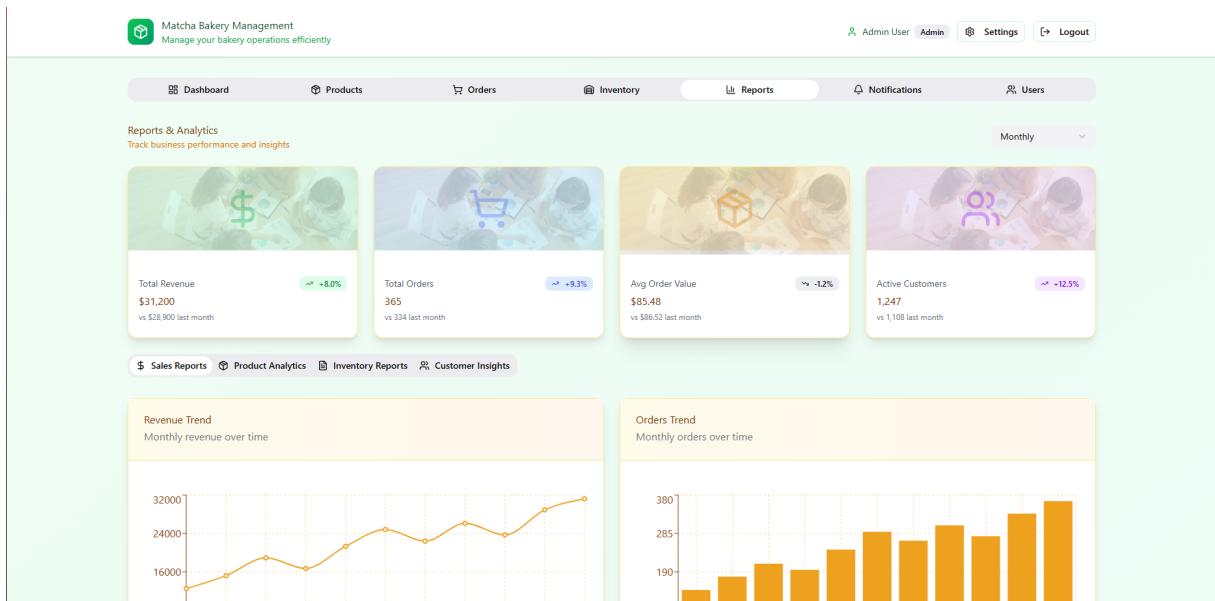


Hình 4.3.5: Giao diện chỉnh sửa thông tin sản phẩm và giá vốn

4.3. Phân hệ Quản trị (Admin Dashboard)

4.3.4 Báo cáo thống kê và Thông báo

- Analytics:** Sử dụng biểu đồ đường (Line chart) và biểu đồ cột (Bar chart) để phân tích xu hướng doanh thu và số lượng đơn hàng theo tháng, giúp chủ cửa hàng ra quyết định chiến lược.
- Trung tâm thông báo (Notifications):** Tập hợp các cảnh báo hệ thống, đơn hàng mới và nhắc nhở nhập kho tại một nơi duy nhất.



Hình 4.3.6: Giao diện báo cáo phân tích doanh thu và xu hướng bán hàng

4.3. Phân hệ Quản trị (Admin Dashboard)

The screenshot shows the Matcha Bakery Management Admin Dashboard. At the top, there is a navigation bar with icons for User, Admin, Settings, and Logout. Below the navigation bar is a horizontal menu with links for Dashboard, Products, Orders, Inventory, Reports, Notifications (which is highlighted in yellow), and Users.

The main content area is titled "Notifications" and includes a sub-instruction: "Stay updated with alerts and system messages". It displays four categories with counts: Total Notifications (10), Unread (5), Action Required (4), and Today (0). A red button at the top right indicates "5 Unread" and "4 Action Required".

Below this, there is a section titled "All Notifications" with a sub-instruction: "Manage your notifications and alerts". It shows a list of notifications with filters: All (10), Unread (5), Inventory (3), Orders (4), and System (3). Two notifications are visible:

- Critical Stock Alert** (New) Action Required: Butter stock is critically low (45kg remaining). Minimum threshold is 60kg. Immediate restocking required.
🕒 Inventory - ⏰ Nov 15, 2025
- Low Stock Warning** (New) Action Required: Flour stock is below minimum level (85kg remaining). Consider placing a purchase order soon.
🕒 Inventory - ⏰ Nov 15, 2025

Hình 4.3.7: Trung tâm thông báo và cảnh báo hệ thống

Chương 5

Định hướng phát triển

5.1 Kết luận

Thông qua đồ án này, nhóm đã xây dựng thành công bộ thiết kế giao diện (UI/UX) hoàn chỉnh cho hệ thống "Bakery Management". Bản thiết kế trên Figma không chỉ đáp ứng đầy đủ các yêu cầu chức năng (quản lý đơn hàng, kho, báo cáo) mà còn chú trọng vào trải nghiệm người dùng thông qua việc sử dụng màu sắc thương hiệu nhất quán, bố cục thông tin khoa học và hệ thống cảnh báo trực quan.

Các tính năng nổi bật đã được thiết kế bao gồm:

- Dashboard quản trị trực quan với các chỉ số thời gian thực.
- Quy trình đặt hàng mượt mà dành cho khách hàng cuối.
- Báo cáo phân tích doanh thu chi tiết dưới dạng biểu đồ.

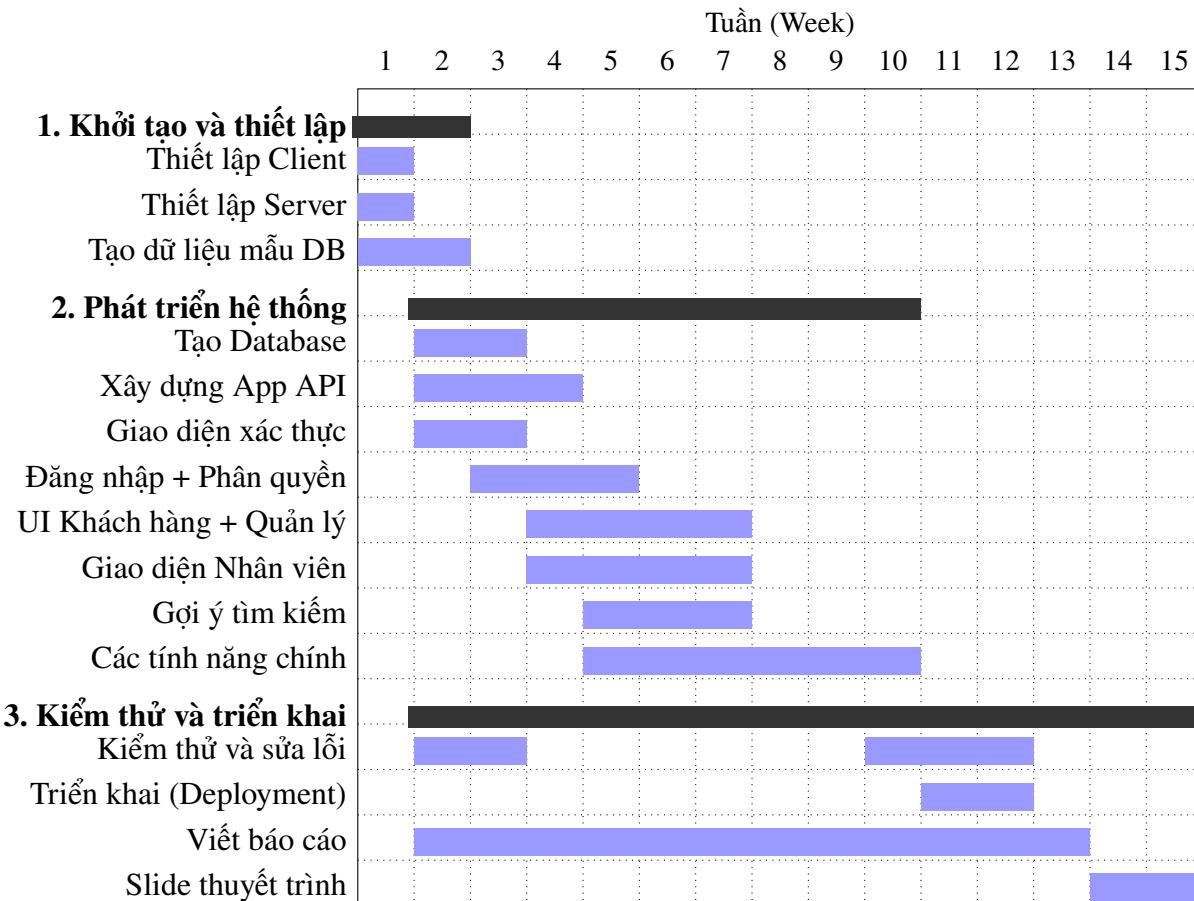
5.2 Kế hoạch hiện thực

Dựa trên bản thiết kế chi tiết này, lộ trình phát triển phần mềm được đề xuất như sau:

1. **Frontend (Vue.js + TailwindCSS):** Chuyển đổi các Component từ Figma (Cards, Modals, Tables) thành mã nguồn Vue.js. Tận dụng TailwindCSS để đảm bảo màu sắc và khoảng cách (Spacing) giống hệt thiết kế.
2. **Backend (Node.js/Express):** Xây dựng các API để phục vụ dữ liệu cho các biểu đồ báo cáo và bảng quản lý kho.
3. **Database (PostgreSQL):** Lưu trữ thông tin sản phẩm, công thức (Ingredients) và lịch sử đơn hàng.

5.2. Kế hoạch hiện thực

Dưới đây là bảng kế hoạch chi tiết phân chia công việc và thời gian thực hiện dự kiến trong 15 tuần:



Hình 5.2.1: Biểu đồ Gantt kế hoạch triển khai dự án

Danh mục tài liệu tham khảo

- [1] Jim Arlow and Ila Neustadt. *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*. Addison-Wesley Professional, 2nd edition, 2005.
- [2] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems*. Pearson, Boston, 7th edition, 2015.
- [3] OpenJS Foundation. Express - fast, unopinionated, minimalist web framework for node.js. <https://expressjs.com/>, 2025. Truy cập: 01/12/2025.
- [4] OpenJS Foundation. Node.js javascript runtime. <https://nodejs.org/>, 2025. Truy cập: 01/12/2025.
- [5] Ian Sommerville. *Software Engineering*. Pearson, London, 10th edition, 2015.
- [6] Tailwind Labs. Tailwind css - rapidly build modern websites without ever leaving your html. <https://tailwindcss.com/>, 2025. Truy cập: 01/12/2025.
- [7] The PostgreSQL Global Development Group. Postgresql: The world's most advanced open source relational database. <https://www.postgresql.org/>, 2025. Truy cập: 01/12/2025.
- [8] Vue.js Team. Vue.js - the progressive javascript framework. <https://vuejs.org/>, 2025. Truy cập: 01/12/2025.