

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO

MÔN PHÂN TÍCH, THIẾT KẾ, ĐÁM BẢO CHẤT LUỢNG PHẦN MỀM

Đề tài:

“ HỆ THỐNG QUẢN LÝ CỦA HÀNG BÁNH NGỌT”

Người hướng dẫn : ThS Nguyễn Anh Hào

Sinh viên thực hiện: TRẦN VIỆT ANH

N20DCCN087

LUƠNG THANH QUÝ

N20DCCN058

NGUYỄN QUANG NGHĨA

N20DCCN045

Lớp : D20CQCNPM01-N

Khoa : 2020 - 2025

Ngành : CÔNG NGHỆ PHẦN MỀM

Hệ : ĐẠI HỌC CHÍNH QUY

TP.HCM, tháng 12/2024

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



BÁO CÁO

**MÔN PHÂN TÍCH, THIẾT KẾ, ĐÁM
BẢO CHẤT LUỢNG PHẦN MỀM**

Đề tài: “ HỆ THỐNG QUẢN LÝ CỦA HÀNG BÁNH NGỌT”

Người hướng dẫn : ThS Nguyễn Anh Hào

Sinh viên thực hiện:

TRẦN VIỆT ANH

N20DCCN087

LƯƠNG THANH QUÝ

N20DCCN058

NGUYỄN QUANG NGHĨA

N20DCCN045

Lớp : D20CQCNPM01-N

Khoá : 2020 - 2025

Ngành : CÔNG NGHỆ PHẦN MỀM

Hệ : ĐẠI HỌC CHÍNH QUY

TP.HCM, tháng 12/2025

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến **Thầy Nguyễn Anh Hào** thuộc Khoa Công nghệ Thông tin, Trường Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở TP. HCM. Thầy đã tận tình hướng dẫn, chỉ bảo và dành nhiều thời gian quý báu để hỗ trợ nhóm trong suốt quá trình nghiên cứu và thực hiện đề tài “**Hệ thống quản lý cửa hàng bánh ngọt**”. Những chỉ dẫn của Thầy đã giúp chúng em hoàn thành luận văn một cách tốt đẹp.

Chúng em xin gửi lời cảm ơn đến Khoa Công nghệ Thông tin, Trường Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở TP. HCM, lãnh đạo và các anh chị đang công tác tại trường đã tạo điều kiện thuận lợi để nhóm có thể hoàn thành đề tài này.

Cuối cùng, nhóm chúng em xin bày tỏ lòng biết ơn sâu sắc đến gia đình và bạn bè, những người luôn ủng hộ, động viên và chia sẻ cùng chúng em trong suốt quá trình học tập và hoàn thành đề tài này.

Thành phố Hồ Chí Minh, Tháng 12 năm 2024

Sinh viên thực hiện

Trần Việt Anh

Nguyễn Thanh Quý

Nguyễn Quang Nghĩa

MỤC LỤC

CHƯƠNG 1. : GIỚI THIỆU	2
1.1. Mục đích nghiên cứu.....	2
1.2. Mục tiêu nghiên cứu	2
1.3. Phương pháp nghiên cứu	2
CHƯƠNG 2. CƠ SỞ KHOA HỌC.....	3
2.1. Bối cảnh và nhu cầu sử dụng	3
2.2. Công nghệ sử dụng.....	4
2.2.1. Java & Spring Boot.....	4
2.2.2. TypeScript & React	5
2.2.3. MySQL.....	6
CHƯƠNG 3. : PHÂN TÍCH HỆ THỐNG	8
3.1. Bối cảnh hiện trạng của hệ thống.....	8
3.1.1. Định nghĩa vấn đề (mục tiêu) mà đề tài sẽ giải quyết.....	9
3.1.2. Hiện trạng trước khi sử dụng phần mềm.....	10
3.1.3. Giải pháp của đề tài	12
3.2. Định nghĩa các tương tác cần thiết cho phần mềm	16
3.2.1. Usecase quản lý hạn sử dụng bánh.....	16
3.2.2. Usecase thêm bánh vào giỏ hàng.....	23
3.2.3. Usecase quản lý kho nguyên liệu.....	26
3.2.4. Usecase gọi món	30
CHƯƠNG 4. : THIẾT KẾ PHẦN MỀM	33
4.1. Lượt đồ use-case cho thiết kế phần mềm	33
4.1.1. Use-case quản lý hạn sử dụng bánh.....	33
4.1.2. Use-case chi tiết sản phẩm.....	37
4.1.3. Use-case quản lý kho nguyên liệu	39
4.1.4. Use-case gọi món	41
4.1.5. Use-case thanh toán	41
4.2. Thiết kế phần mềm để xử lý use-case.....	43
4.2.1. Form.....	43
4.2.2. API	73
4.3. Sơ đồ lớp	95
4.4. Thiết kế CSDL cho phần mềm.....	95
4.4.1. Xác định thực thể	95
4.4.2. Mô hình thực thể ERD.....	96
4.4.3. Thiết kế chi tiết	96
4.5. Bảng tham chiếu	101
CHƯƠNG 5. KẾT LUẬN.....	103
5.1. Kết quả đạt được	103
5.2. Hạn chế	103
5.3. Hướng phát triển	103

LUẬC ĐỒ

Lược đồ 1. 1 quản lý hạn sử dụng tổng quát	3
Lược đồ 1. 2 quản lý kho tổng quát.....	3
Lược đồ 1. 3 gọi món tổng quát	4
Lược đồ 3. 1 quản lý hạn sử dụng bánh	10
Lược đồ 3. 2 quản ký kho nguyên liệu.....	11
Lược đồ 3. 3 gọi món	12
Lược đồ 3. 4 giải pháp quản lý hạn sử dụng bánh	13
Lược đồ 3. 5 giải pháp quản lý kho	14
Lược đồ 3. 6 giải pháp gọi món	15
Lược đồ 3. 7 usecase quản lý hạn sử dụng.....	16
Lược đồ 3. 8 tuần tự quản lý hạn sử dụng.....	18
Lược đồ 3. 9 trạng thái lô bánh	21
Lược đồ 3. 10 hoạt động của trạng thái lô bánh.....	22
Lược đồ 3. 11 usecase thêm bánh vào giỏ hàng.....	23
Lược đồ 3. 12 tuần tự thêm bánh vào giỏ hàng.....	25
Lược đồ 3. 13 usecase quản lý kho	26
Lược đồ 3. 14 tuần tự quản lý kho	28
Lược đồ 3. 15 trạng thái nguyên liệu	29
Lược đồ 3. 16 usecase gọi món	30
Lược đồ 3. 17 tuần tự gọi món	31
Lược đồ 4. 1 usecase quản lý hạn sử dụng.....	33
Lược đồ 4. 2 tuần tự hạn sử dụng.....	34
Lược đồ 4. 3 usecase thêm bánh vào giỏ hàng.....	37
Lược đồ 4. 4 tuần tự thêm bánh vào giỏ hàng	37
Lược đồ 4. 5 usecase quản lý kho nguyên liệu.....	39
Lược đồ 4. 6 tuần tự quản lý kho nguyên liệu.....	40
Lược đồ 4. 7 usecase gọi món	41
Lược đồ 4. 8 tuần tự gọi món	41
Lược đồ 4. 9 usecase thanh toán	42
Lược đồ 4. 10 tuần tự thanh toán	42
Lược đồ 4. 11 lớp	95
Lược đồ 4. 12 ERD	96

HÌNH ẢNH

Hình ảnh 4. 1 form đăng nhập.....	43
Hình ảnh 4. 2 form chi tiết sản phẩm	44
Hình ảnh 4. 3 Form quản lý Bill.....	48
Hình ảnh 4. 4 Form giỏ hàng đặt hàng	50
Hình ảnh 4. 5 Form thanh toán mã qr	52
Hình ảnh 4. 6 Form quản lý kho.....	53
Hình ảnh 4. 7 form điền thông tin nguyên liệu	54

Hình ảnh 4. 8 form thêm đơn vị	54
Hình ảnh 4. 9 form sửa nguyên liệu	55
Hình ảnh 4. 10 form xác nhận xoá	55
Hình ảnh 4. 11 combobox trạng thái nguyên liệu	56
Hình ảnh 4. 12 combobox lọc nguyên liệu.....	56
Hình ảnh 4. 13 form nhập nguyên liệu.....	57
Hình ảnh 4. 14 form chọn nguyên liệu để thêm	58
Hình ảnh 4. 15 combobox nhà cung cấp	58
Hình ảnh 4. 16 form xuất nguyên liệu.....	60
Hình ảnh 4. 17 form chọn các sản phẩm để thêm	60
Hình ảnh 4. 18 form thông tin nguyên liệu	61
Hình ảnh 4. 19 form lịch sử nhập nguyên liệu	62
Hình ảnh 4. 20 form chi tiết nhập nguyên liệu.....	63
Hình ảnh 4. 21 form lịch sử xuất nguyên liệu	64
Hình ảnh 4. 22 form chi tiết xuất nguyên liệu.....	65
Hình ảnh 4. 23 form quản lý hạn sử dụng	65
Hình ảnh 4. 24 form huỷ nhanh.....	67
Hình ảnh 4. 25 form lịch sử huỷ bánh.....	69
Hình ảnh 4. 26 form chi tiết huỷ bánh.....	70
Hình ảnh 4. 27 form giảm giá nhanh.....	71

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

Tùy viết tắt	Nghĩa đầy đủ	Ý nghĩa
CSDL	Cơ sở dữ liệu (Database)	Hệ thống lưu trữ và quản lý dữ liệu có cấu trúc. Ví dụ: MySQL, PostgreSQL, MongoDB...
ERD	Entity Relationship Diagram	Sơ đồ thực thể - mối quan hệ, dùng để mô tả cấu trúc logic của cơ sở dữ liệu quan hệ. Bao gồm các thực thể, thuộc tính và mối quan hệ giữa chúng.
API	Application Programming Interface	Giao diện lập trình ứng dụng - tập hợp các quy tắc và cơ chế cho phép các phần mềm khác nhau giao tiếp với nhau.
UC	Use Case	Trường hợp sử dụng - mô tả tương tác giữa người dùng với hệ thống để đạt được một mục tiêu cụ thể.
SD	Sequence Diagram	Sơ đồ trình tự - biểu diễn luồng tương tác giữa các đối tượng theo thời gian trong một kịch bản sử dụng cụ thể.

LỜI MỞ ĐẦU

Trong bối cảnh thị trường dịch vụ ngày càng cạnh tranh gay gắt, các cửa hàng bánh ngọt đang đối mặt với nhiều thách thức lớn trong việc duy trì hiệu quả hoạt động và thu hút khách hàng. Những khó khăn phổ biến như thiếu hụt nguyên liệu, thất thoát trong quá trình chế biến, kiểm soát không hiệu quả hạn sử dụng sản phẩm, thời gian phục vụ chậm trễ, và quản lý nhân sự chưa tối ưu đang ảnh hưởng nghiêm trọng đến doanh thu cũng như chất lượng dịch vụ. Đồng thời, việc thiếu các quy trình kiểm kê và dự trữ nguyên liệu hợp lý khiến cửa hàng gặp khó khăn trong việc chuẩn bị nguyên liệu kịp thời, từ đó ảnh hưởng đến chất lượng sản phẩm.

Trước thực trạng này, việc ứng dụng công nghệ thông tin vào quản lý và vận hành cửa hàng bánh ngọt đã trở thành một giải pháp không thể thiếu. Một hệ thống quản lý hiệu quả không chỉ giúp giảm thiểu sai sót trong hoạt động thường ngày mà còn cải thiện đáng kể trải nghiệm của khách hàng, từ đó thúc đẩy doanh thu và lợi nhuận. Những công cụ như hệ thống quản lý kho tự động, kiểm kê nguyên liệu, và đặt hàng trực tuyến sẽ hỗ trợ cửa hàng duy trì nguồn cung ổn định, giảm thiểu lãng phí và tối ưu hóa quá trình phục vụ.

Đề tài “Hệ thống quản lý cửa hàng bánh ngọt” được lựa chọn nhằm nghiên cứu và phát triển giải pháp công nghệ cho việc tối ưu hóa quy trình quản lý tại cửa hàng. Mục tiêu của đề tài là đóng góp vào việc nâng cao hiệu quả hoạt động và chất lượng dịch vụ cho cửa hàng bánh ngọt, đồng thời cung cấp một giải pháp tham khảo cho các hoạt động kinh doanh tương tự.

Phạm vi nghiên cứu của đề tài tập trung vào cửa hàng “Phương Chi,” với việc thu thập dữ liệu thực tế từ quá trình kinh doanh của cửa hàng, kết hợp tham khảo các hệ thống quản lý tương tự. Hệ thống sẽ được phát triển dựa trên phương pháp hướng đối tượng, sử dụng MySQL Server làm hệ quản trị cơ sở dữ liệu và ngôn ngữ lập trình Java. Chương trình sẽ hỗ trợ tối ưu hóa việc quản lý các danh mục sản phẩm, giúp cửa hàng tiết kiệm thời gian, nguồn lực và tăng cường hiệu quả kinh doanh, từ đó phục vụ tốt hơn nhu cầu của khách hàng.

CHƯƠNG 1. : GIỚI THIỆU

1.1. Mục đích nghiên cứu

- Tối ưu hóa quy trình quản lý kho và nguyên liệu
 - Xây dựng hệ thống kiểm soát tồn kho tự động
 - Theo dõi và quản lý hạn sử dụng nguyên liệu
 - Tự động hóa quy trình đặt hàng và nhập kho
 - Giảm thiểu thất thoát và lãng phí nguyên liệu
 - Nâng cao hiệu quả quy trình phục vụ và trải nghiệm khách hàng
- Tối ưu hóa thời gian xử lý đơn hàng
 - Cải thiện quy trình đặt hàng và thanh toán
 - Phát triển hệ thống đặt hàng trực tuyến
 - Nâng cao khả năng tương tác với khách hàng
- Đảm bảo chất lượng sản phẩm
 - Theo dõi và đánh giá chất lượng sản phẩm
 - Duy trì tiêu chuẩn sản phẩm đồng nhất
 - Tối ưu hóa quy trình bảo quản sản phẩm

1.2. Mục tiêu nghiên cứu

Với người quản lý:

- Cung cấp công cụ quản lý toàn diện về kho, nguyên liệu và doanh thu
- Hỗ trợ ra quyết định thông qua báo cáo và thống kê chi tiết
- Tối ưu hóa chi phí vận hành
- Nâng cao hiệu quả quản lý nhân sự và phân công công việc

Với nhân viên:

- Đơn giản hóa quy trình làm việc hàng ngày
- Giảm thiểu sai sót trong quá trình thực hiện công việc
- Tăng cường khả năng phối hợp giữa các bộ phận
- Cải thiện hiệu suất làm việc thông qua công cụ hỗ trợ

Với khách hàng:

- Cung cấp trải nghiệm mua sắm thuận tiện và nhanh chóng
- Đảm bảo chất lượng sản phẩm ổn định
- Nâng cao mức độ hài lòng thông qua dịch vụ chuyên nghiệp
- Tạo kênh tương tác và phản hồi hiệu quả

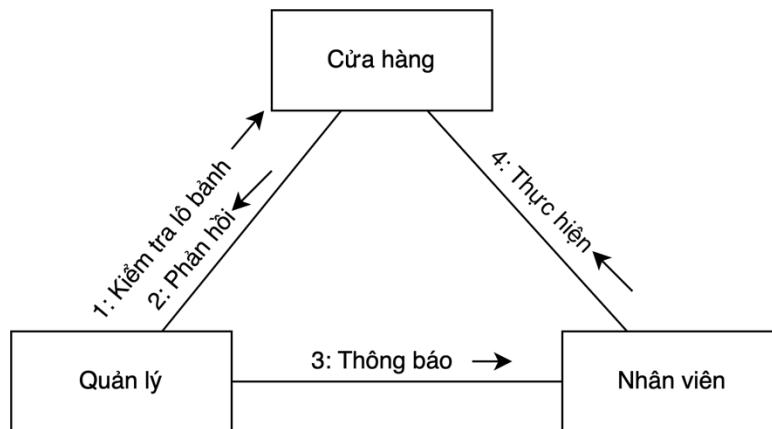
1.3. Phương pháp nghiên cứu

Phân tích và thiết kế theo hướng đối tượng.

CHƯƠNG 2. CƠ SỞ KHOA HỌC

2.1. Bối cảnh và nhu cầu sử dụng

Quy trình quản lý hạn sử dụng sản phẩm tổng quát

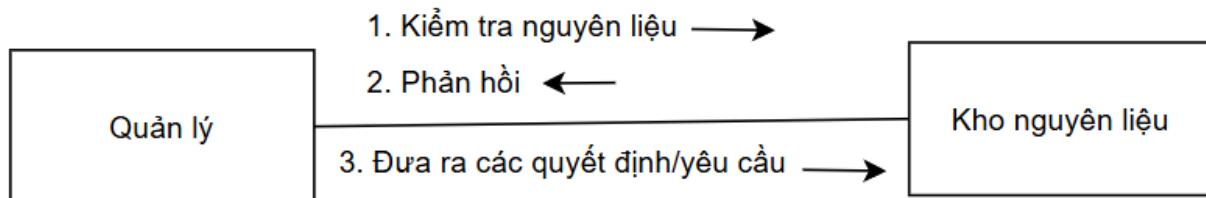


Lược đồ 1. 1 quản lý hạn sử dụng tổng quát

Mô tả quy trình:

1. Người quản lý sẽ kiểm tra toàn bộ các lô bánh hiện đang được bày bán trên cửa hàng
2. Sau khi kiểm tra xong sẽ được phản hồi 3 loại lô bánh: Còn hạn dài, Sắp hết hạn, Hết hạn.
3. Tuỳ thuộc vào từng loại quản lý sẽ thông báo làm công việc tương ứng với lô bánh
4. Nhân viên sẽ thực hiện công việc tương ứng

Quy trình quản lý kho tổng quát



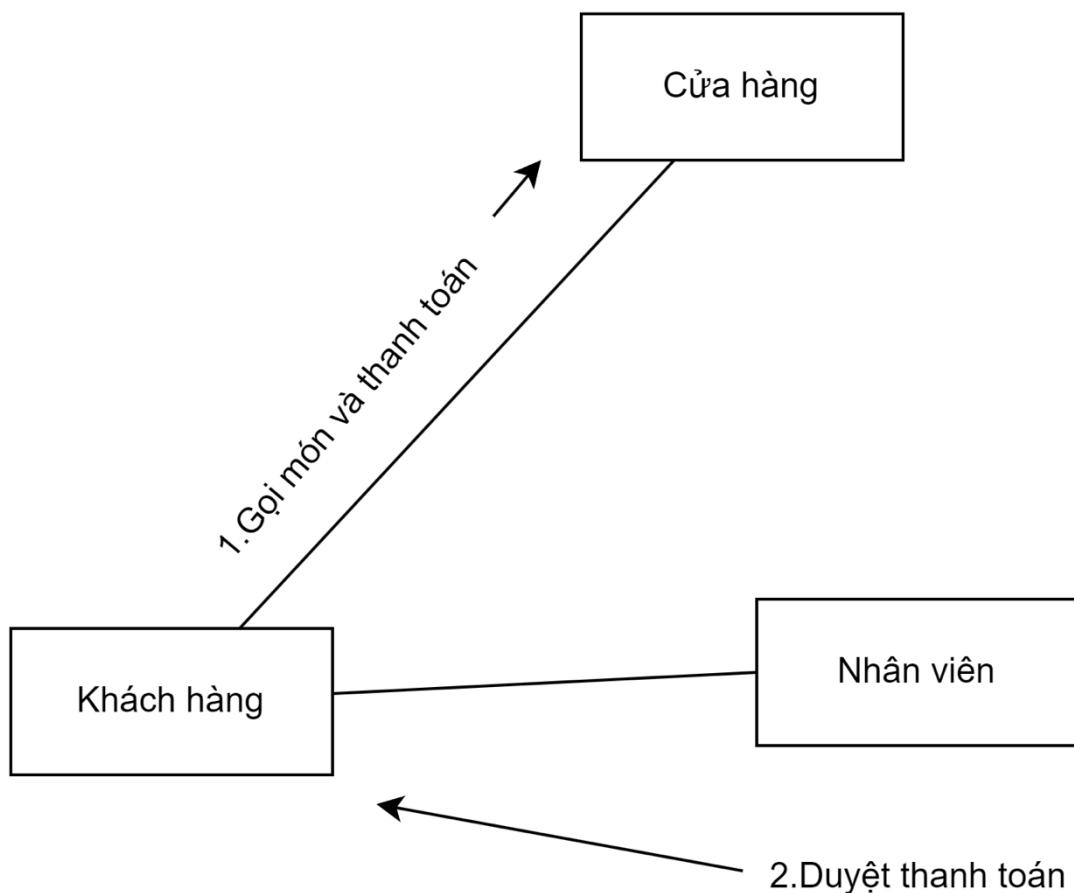
Lược đồ 1. 2 quản lý kho tổng quát

Mô tả quy trình:

1. Người quản lý sẽ kiểm tra tất cả các nguyên liệu trong kho
2. Sau khi kiểm tra, người quản lý sẽ biết được nguyên liệu nào còn nhiều, nguyên liệu nào sắp hết hay đã hết

3. Tùy vào số lượng các nguyên liệu còn lại mà người quản lý sẽ đưa ra quyết định nhập thêm nguyên liệu hay xuất kho các nguyên liệu để làm bánh

Quy trình gọi món tổng quát



Lược đồ 1. 3 gọi món tổng quát

Mô tả quy trình:

1. Khách hàng sẽ quét mã để xem danh sách bánh, thực hiện đặt bánh sau đó sẽ tiến hành thanh toán. Thanh toán gồm quét mã hoặc thanh toán bằng tiền mặt.
2. Nhân viên sẽ xác nhận đã thu tiền của khách khi khách cung cấp mã hóa đơn khi chọn thanh toán bằng tiền mặt.

2.2. Công nghệ sử dụng

2.2.1. Java & Spring Boot

Java là ngôn ngữ lập trình hướng đối tượng, mạnh mẽ và linh hoạt với các đặc điểm:

Độc lập nền tảng thông qua Java Virtual Machine (JVM)

Hỗ trợ đa luồng (multithreading) và xử lý đồng thời

Tính đóng gói (encapsulation), kế thừa (inheritance) và đa hình (polymorphism)

Garbage collection tự động quản lý bộ nhớ

Spring Boot là framework phát triển từ Spring Framework với các đặc điểm:

Auto-configuration giúp cấu hình tự động các beans và dependencies

Embedded server (Tomcat, Jetty) tích hợp sẵn

Starter dependencies đơn giản hóa quá trình cấu hình Maven/Gradle

Actuator hỗ trợ monitoring và metrics

Theo mô hình MVC (Model-View-Controller):

Model: Đại diện cho dữ liệu và business logic

View: Hiển thị thông tin cho người dùng

Controller: Xử lý requests và điều hướng

Dependency Injection (DI) và Inversion of Control (IoC):

Giảm sự phụ thuộc giữa các components

Dễ dàng test và maintain code

Quản lý vòng đời của beans

Lý do lựa chọn:

Cấu trúc project được tổ chức rõ ràng theo các module, dễ dàng quản lý và mở rộng

Tích hợp sẵn nhiều tính năng quan trọng như dependency injection, security, JPA giúp tiết kiệm thời gian phát triển

Hiệu suất cao và ổn định, phù hợp cho ứng dụng enterprise

Cộng đồng developer lớn mạnh, nguồn tài liệu tham khảo phong phú

Khả năng tích hợp linh hoạt với các công nghệ khác như REST API, WebSocket

2.2.2. TypeScript & React

TypeScript mở rộng từ JavaScript với các tính năng:

Static typing system cho phép khai báo kiểu dữ liệu

Interface và Type aliases định nghĩa cấu trúc dữ liệu

Generics cho phép tái sử dụng code với nhiều kiểu dữ liệu

Decorators hỗ trợ meta-programming

Enhanced IDE support với IntelliSense

Access modifiers (public, private, protected)

React là thư viện JavaScript với cơ chế:

Component-based architecture:

Functional Components và Class Components

Props và State management

Lifecycle methods

Lý do lựa chọn:

TypeScript cung cấp type checking giúp phát hiện lỗi sớm trong quá trình phát triển

Code base dễ đọc, dễ bảo trì và refactor nhờ type system
React có cơ chế component tái sử dụng cao, giúp phát triển nhanh và nhất quán
Virtual DOM của React tối ưu hiệu năng render của ứng dụng
Ecosystem phong phú với nhiều thư viện UI components và state management
Tương thích tốt với REST API từ Spring Boot backend

2.2.3. MySQL

Hệ quản trị CSDL quan hệ với các đặc điểm:

ACID Properties:

Atomicity: Tính nguyên tử của transactions
Consistency: Tính nhất quán của dữ liệu
Isolation: Tính độc lập giữa các transactions
Durability: Tính bền vững của dữ liệu

Kiến trúc:

Client-server architecture
Query optimization engine
Storage engines (InnoDB, MyISAM)
Buffer pool management
Index structures (B-tree, Hash)

Tính năng nâng cao:

Stored Procedures và Functions
Triggers và Events
Views và Temporary Tables
Partitioning và Sharding
Replication và Clustering
Transaction management
Full-text search

Security features:

Access control và User privileges
SSL/TLS encryption
Backup và Recovery mechanisms
Audit logging

Lý do lựa chọn:

Giải pháp mã nguồn mở, miễn phí với khả năng mở rộng cao
Hiệu suất ổn định và độ tin cậy cao trong xử lý giao dịch
Tích hợp seamless với Spring Boot thông qua JPA/Hibernate
Công cụ quản trị trực quan như MySQL Workbench giúp dễ dàng thao tác

Khả năng scale theo chiều ngang và chiều dọc linh hoạt
Cộng đồng hỗ trợ lớn, nhiều tài liệu tham khảo và best practices

CHƯƠNG 3. : PHÂN TÍCH HỆ THỐNG

3.1. Bối cảnh hiện trạng của hệ thống

Quy trình quản lý hạn sử dụng sản phẩm bánh tại cửa hàng

Hiện trạng quản lý

- Việc quản lý số lượng sản phẩm đang được thực hiện thủ công, không có hệ thống ghi chép cụ thể
- Kiểm tra hạn sử dụng được thực hiện bằng cách kiểm tra trực tiếp từng lô bánh
- Sử dụng phương pháp dán nhãn trực tiếp lên bao bì để đánh dấu trạng thái sản phẩm

Quy trình xử lý theo hạn sử dụng

Đối với bánh hết hạn

- Người quản lý trực tiếp kiểm tra và xác định các sản phẩm hết hạn
- Chỉ đạo nhân viên thu hồi và tiến hành huỷ bỏ
- Loại bỏ khỏi khu vực bày bán

Đối với bánh sắp hết hạn

- Xác định các sản phẩm gần đến hạn sử dụng
- Thực hiện dán nhãn giảm giá
- Sắp xếp ở vị trí dễ thấy để khách hàng lựa chọn

Đối với bánh còn hạn sử dụng dài

- Giữ nguyên giá bán
- Duy trì vị trí bày bán thông thường

Quy trình quản lý kho nguyên liệu

Hiện trạng:

- Việc quản lý các nguyên liệu tại một số cửa hàng vẫn còn đang được thực hiện một cách thủ công, không sử dụng hệ thống để lưu trữ và ghi chép
- Kiểm tra số lượng tồn của các nguyên liệu bằng việc trực tiếp kiểm tra từng nguyên liệu có trong kho

Các quy trình xử lý

- Các nguyên liệu sắp hết và đã hết
 - Người quản lý sẽ tiến hành nhập nguyên liệu từ một nhà cung cấp nào đó
- Các nguyên liệu còn nhiều
 - Kiểm tra và sử dụng như bình thường
- Khi cần làm bánh
 - Lấy các nguyên liệu trong kho theo loại bánh cần làm và số lượng bánh muốn làm
 - Ước tính số lượng nguyên liệu cần dùng để xuất kho để làm bánh
 - Lấy nguyên liệu theo ước tính hoặc tính toán thủ công

Quy trình đặt bánh

Hiện trạng:

- Hiện trạng hiện nay của việc quản lý gọi món tại các cửa hàng bánh thường dựa vào phương thức thủ công, chẳng hạn như ghi chép trên giấy hoặc nhân viên tự ghi nhớ trong đầu. Tuy nhiên, điều này tiềm ẩn nhiều rủi ro sai sót khi lượng khách hàng tăng cao, chẳng hạn như ghi nhầm món, thiếu sót đơn hàng, hoặc nhầm lẫn trong việc sắp xếp thứ tự phục vụ. Việc ghi nhớ và xử lý các yêu cầu của khách hàng hoàn toàn bằng tay cũng có thể ảnh hưởng đến độ chính xác trong khâu thanh toán, dẫn đến tình trạng tính nhầm số tiền phải thu từ khách hàng hoặc bỏ sót các mục trong hóa đơn. Điều này không chỉ làm mất thời gian của khách hàng mà còn ảnh hưởng đến hình ảnh chuyên nghiệp của cửa hàng, dễ gây ra những trải nghiệm không hài lòng cho khách hàng và giảm hiệu quả kinh doanh.

Các quy trình xử lý

- Khách hàng quét mã QR tại bàn hoặc tài quầy để thực hiện gọi món
- Sau khi gọi món xong khách có thể chọn thanh toán quét mã ngay trên màn hình điện thoại hoặc có thể thanh toán tiền mặt
- Khách hàng nếu thanh toán tiền mặt sẽ cung cấp tên hoặc số điện thoại cho nhân viên để nhân viên xem hóa đơn bao nhiêu tiền để thu cũng như duyệt thanh toán.

3.1.1. Định nghĩa vấn đề (mục tiêu) mà đề tài sẽ giải quyết

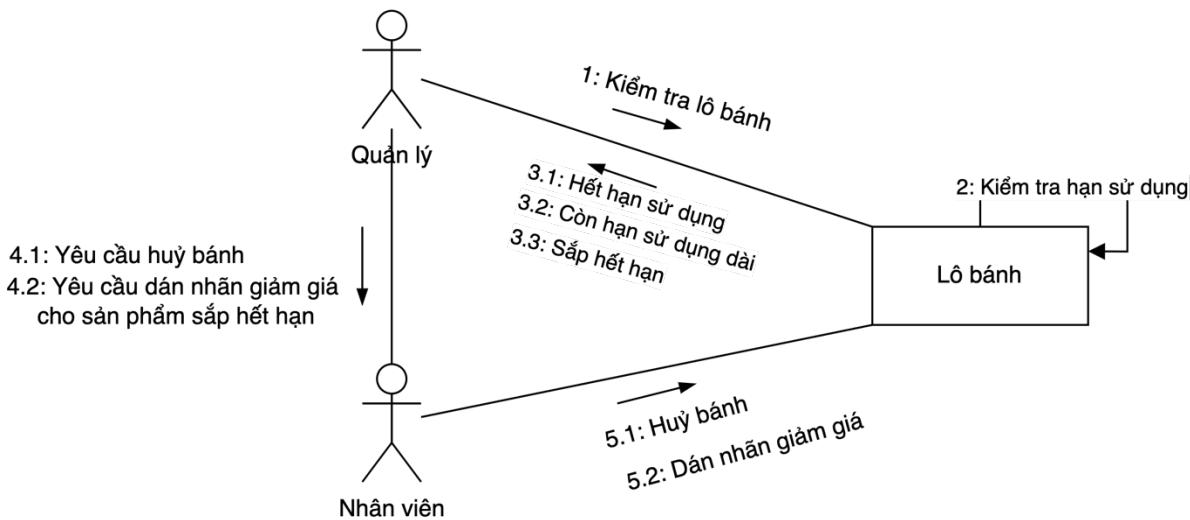
- Đề tài sẽ xây dựng một hệ thống thân thiện, tiện dụng để quản lý có thể tiết kiệm trong quá trình tìm kiếm món ăn quản lý hạn sử dụng dễ dàng.
- Thay vì phải đi kiểm tra từng lô hàng có trong tủ đang bày bán xem lô nào gần hết để bày ra trước giảm giá hoặc lô nào hết để huỷ bỏ lô bánh. Thịt nay hệ thống sẽ giúp người quản lý tự động liệt kê ra các sản phẩm sắp hết hạn để dễ dàng áp mã giảm giá, cũng như lên danh sách loại bỏ các sản phẩm hết hạn.
- Quản lý, kiểm tra các nguyên liệu đang có trong kho một cách dễ dàng và nhanh chóng, không tốn nhiều thời gian để kiểm tra tất cả các nguyên liệu trong kho.
- Hệ thống cảnh báo các nguyên liệu cần nhập thêm (sắp hết hoặc đã hết) để có thể nhanh chóng nhập thêm nguyên liệu để tránh bị ảnh hưởng đến công việc của cửa tiệm.
- Khi cần xuất nguyên liệu để làm bánh, ta có thể dựa vào công thức của bánh và số lượng cần làm để xuất kho nguyên liệu một cách chính xác, dễ dàng hơn, từ đó có thể kiểm soát chất lượng của bánh đã làm.
- Khách hàng quét mã QR tại bàn hoặc quầy để truy cập vào menu và thực hiện gọi món trực tiếp trên điện thoại mà không cần tương tác với nhân viên. Khi thanh toán

qua mã QR, hóa đơn sẽ được cập nhật tự động trên hệ thống, giảm thời gian xử lý cho nhân viên.

- Với khách thanh toán tiền mặt, hệ thống tạo mã hóa đơn để khách hàng cung cấp cho nhân viên, giúp xác minh số tiền cần thu và giảm nhầm lẫn trong quá trình thanh toán.
- Nhân viên dễ dàng tra cứu hóa đơn qua mã do khách hàng cung cấp khi thanh toán tiền mặt, giúp xác minh chính xác số tiền cần thu và trạng thái thanh toán.
- Hóa đơn được xử lý nhanh chóng, đảm bảo trải nghiệm liền mạch và tránh sai sót trong tính tiền.

3.1.2. Hiện trạng trước khi sử dụng phần mềm

- Quản lý hạn sử dụng bánh



Lược đồ 3.1 quản lý hạn sử dụng bánh

1. Người quản lý sẽ kiểm tra các lô bánh khác nhau đang được bày bán
2. Kiểm tra hạn sử dụng của lô bánh
3. Sau khi kiểm tra phản hồi lại
 - 3.1. Lô bánh đã hết hạn sử dụng
 - 3.2. Lô bánh còn hạn sử dụng dài
 - 3.3. Lô bánh sắp hết hạn sử dụng
4. Quản lý thông báo
 - 4.1. Người quản lý yêu cầu nhân viên huỷ lô bánh khi nó đã hết hạn sử dụng không tiếp tục bày bán
 - 4.2. Người quản lý yêu cầu dán nhãn giảm giá lên lô bánh đó khi nó gần hết hạn và mang trưng bày gần mắt người mua nhất

5. Nhân viên tiến hành thực hiện

5.1. Người nhân viên tiến hành huỷ bánh hết hạn

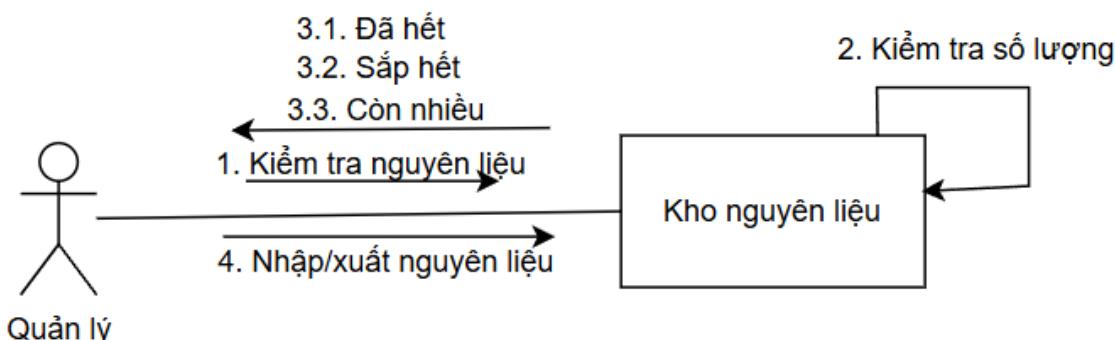
5.2. Người nhân viên dán nhãn và trưng bày bánh gần hết hạn

Kết luận: Có thể thấy được sự giao tiếp giữa những lô bánh, người quản lý và nhân viên có rất nhiều công việc khác nhau có thể gây ra sai sót chậm trễ thông tin.

Người quản lý phải thực hiện nhiều công việc thủ công như kiểm tra và gửi yêu cầu lại cho nhân viên chưa kể đến khi quản lý thủ công như vậy rất dễ sai sót bị sót lại sản phẩm hoặc thiếu chính xác.

Nhân viên phải xem người quản lý muốn giảm giá bao nhiêu phần trăm cho sản phẩm đó, hay huỷ bỏ sản phẩm nào hết hạn, làm giảm sự thuận tiện và nhân viên phải làm quá nhiều việc khác nhau. Do đó, cần có một giải pháp dùng ứng dụng để giải quyết trường hợp này.

- Quản lý kho nguyên liệu



Lược đồ 3.2 quản lý kho nguyên liệu

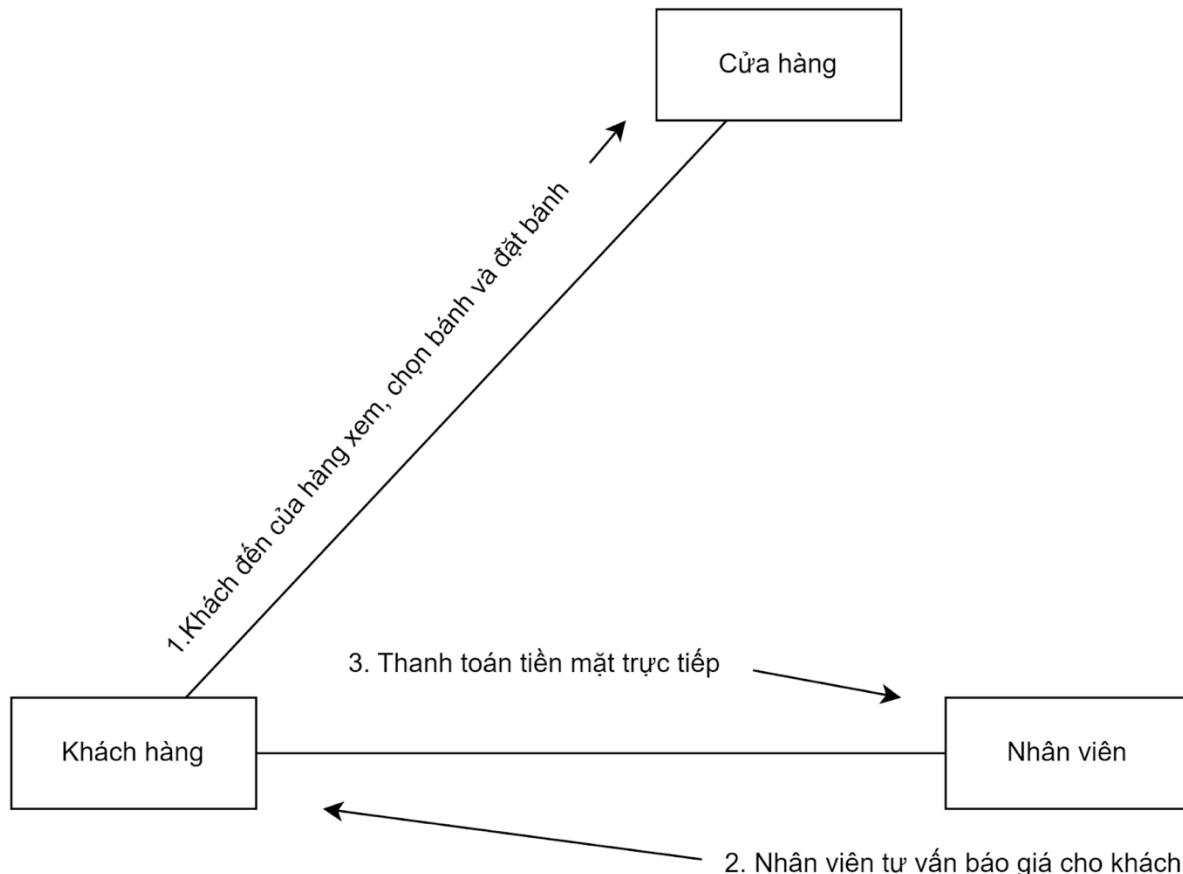
1. Người quản lý sẽ kiểm tra các nguyên liệu trong kho
2. Kiểm tra số lượng tồn của các nguyên liệu
3. Sau khi kiểm tra ta có:
 - 3.1. Các nguyên liệu đã hết
 - 3.2. Các nguyên liệu sắp hết
 - 3.3. Các nguyên liệu vẫn còn nhiều
4. Sau khi kiểm tra số lượng tồn của các nguyên liệu:
 - Người quản lý có thể nhập thêm nguyên liệu vào kho dựa trên danh sách các nguyên liệu sắp hết hoặc đã hết, có thể nhập thêm các nguyên liệu khác nếu cần
 - Người quản lý có thể yêu cầu xuất kho để làm bánh nếu số lượng nguyên liệu vẫn còn đủ để đáp ứng

Kết luận: Việc quản lý kho có thể tốn rất nhiều thời gian để kiểm tra thủ công và khó khăn trong việc kiểm soát nguyên liệu nếu không có hệ thống hỗ trợ.

Việc kiểm tra chậm trễ có thể ảnh hưởng đến các hoạt động của cửa hàng nếu nguyên liệu bị thiếu hụt nhưng chưa kịp thời nhập thêm.

Xuất kho nguyên liệu theo các loại bánh cần làm cần phải tính toán các nguyên liệu bằng tay dẽ gây sai sót, có thể ảnh hưởng đến chất lượng của bánh làm ra nếu nguyên liệu không đúng lượng cần thiết.

- Gọi món



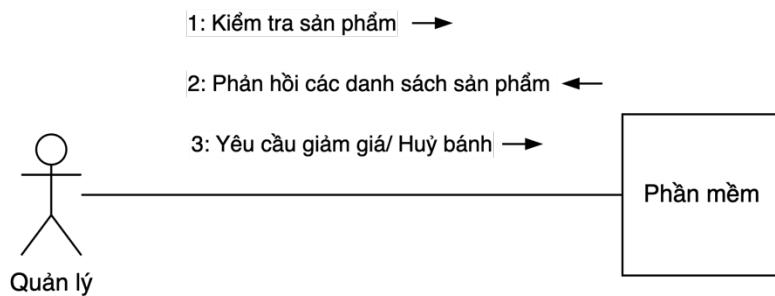
Lược đồ 3.3 gọi món

1. Khách hàng phải trực tiếp đến cửa hàng để xem và chọn sản phẩm
2. Nhân viên tư vấn trực tiếp và báo giá cho khách hàng
3. Việc thanh toán được thực hiện bằng tiền mặt trực tiếp với nhân viên

Kết luận: Quy trình rất đơn giản nhưng gây ít tiện lợi cho khách hàng. Nhân viên phải một mình xử lý nhiều công việc.

3.1.3. Giải pháp của đề tài

Quản lý hạn sử dụng bánh



Lược đồ 3.4 giải pháp quản lý hạn sử dụng bánh

Mô tả chi tiết từng bước

1. Kiểm tra sản phẩm

- Người quản lý truy cập vào phần mềm
- Gửi yêu cầu kiểm tra tình trạng sản phẩm trong hệ thống
- Đây là bước khởi đầu của quy trình

2. Phản hồi các danh sách sản phẩm

- Phần mềm tổng hợp và gửi lại cho quản lý:
- Danh sách bánh hết hạn
- Danh sách bánh sắp hết hạn
- Danh sách bánh còn hạn sử dụng dài
- Giúp quản lý có cái nhìn tổng quan về tình trạng sản phẩm

3. Yêu cầu giảm giá/Huỷ bánh

- Quản lý xem xét danh sách và đưa ra quyết định

Kết luận: Quy trình này thể hiện sự cải tiến so với cách làm thủ công có thể thấy một cách rõ ràng đã giảm được nhiều bước và trở nên đơn giản hơn.

Quản lý không cần phải kiểm tra thủ công hoặc đích thân bảo nhân viên tương tác với lô bánh.

Quy trình quản lý kho nguyên liệu

Hiện trạng:

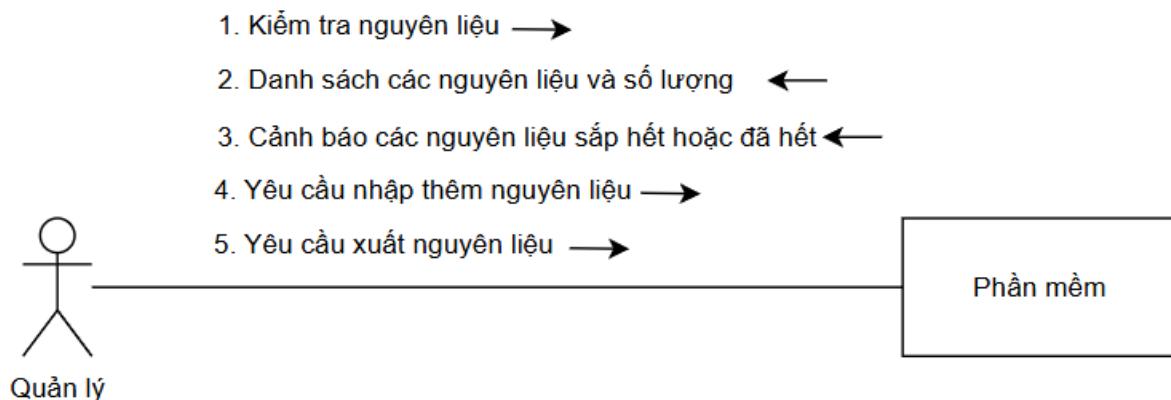
- Việc quản lý các nguyên liệu tại một số cửa hàng vẫn còn đang được thực hiện một cách thủ công, không sử dụng hệ thống để lưu trữ và ghi chép
- Kiểm tra số lượng tồn của các nguyên liệu bằng việc trực tiếp kiểm tra từng nguyên liệu có trong kho

Các quy trình xử lý

- Các nguyên liệu sắp hết và đã hết
 - Người quản lý sẽ tiến hành nhập nguyên liệu từ một nhà cung cấp nào đó
- Các nguyên liệu còn nhiều
 - Kiểm tra và sử dụng như bình thường

- Khi cần làm bánh
 - Lấy các nguyên liệu trong kho theo loại bánh cần làm và số lượng bánh muốn làm
 - Ước tính số lượng nguyên liệu cần dùng để xuất kho để làm bánh
 - Lấy nguyên liệu theo ước tính hoặc tính toán thủ công

Quản lý kho nguyên liệu



Lược đồ 3. 5 giải pháp quản lý kho

1. Kiểm tra nguyên liệu
 - Người quản lý đăng nhập vào phần mềm
 - Truy cập vào mục quản lý kho
 - Đây là bước khởi đầu của quy trình
2. Danh sách các nguyên liệu và số lượng
 - Phần mềm sẽ hiển thị danh sách các nguyên liệu trong kho và số lượng tương ứng
 - Cảnh báo các nguyên liệu sắp hết hoặc đã hết
 - Phần mềm sẽ dựa trên số lượng của các nguyên liệu để xem xét và cảnh báo nếu các nguyên liệu đã xuống dưới ngưỡng cảnh báo
 - Thống kê một danh sách các nguyên liệu đang cảnh báo
3. Yêu cầu nhập thêm nguyên liệu
 - Người quản lý có thể gửi yêu cầu nhập thêm nguyên liệu vào kho
 - Dựa trên danh sách các nguyên liệu đang cảnh báo để nhanh chóng nhập thêm nguyên liệu, tránh sự chậm trễ
 - Sau khi nhập nguyên liệu thành công, phần mềm sẽ tự động cập nhật lại số lượng các nguyên liệu trong kho
4. Yêu cầu xuất nguyên liệu
 - Để làm bánh, người quản lý cần đưa ra yêu cầu xuất kho nguyên liệu
 - Người quản lý sẽ chọn công thức bánh và nhập số lượng bánh cần làm

- Phần mềm sẽ tự động dựa vào công thức để tính ra được số lượng nguyên liệu để xuất kho

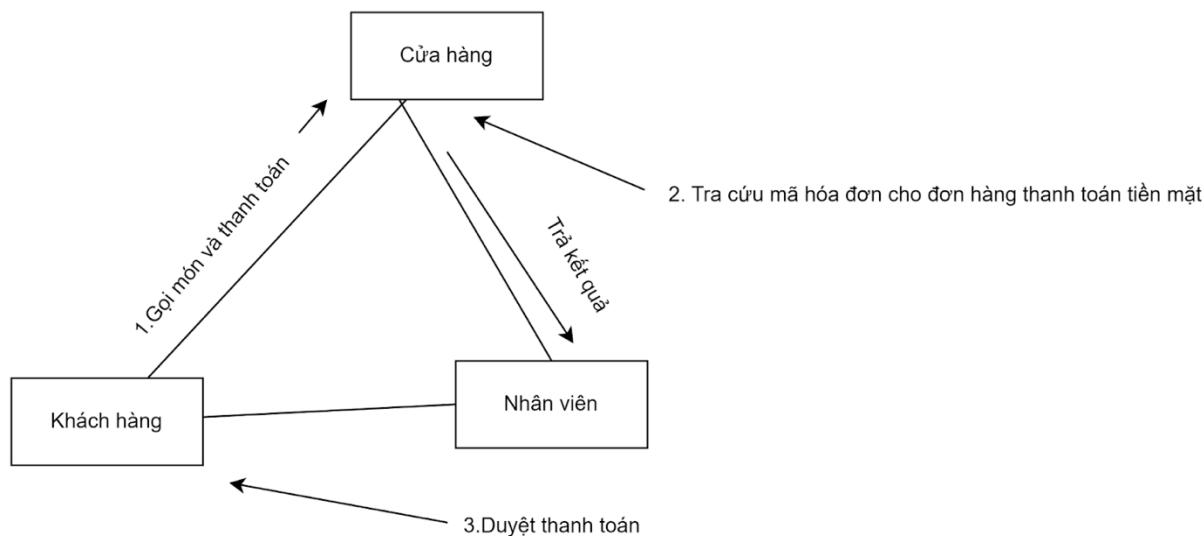
- Phần mềm sẽ cập nhật lại số lượng nguyên liệu khi xuất kho thành công

Kết luận: Phần mềm giúp người quản lý dễ dàng kiểm tra số lượng các nguyên liệu trong kho một cách nhanh chóng và dễ dàng.

Giúp người quản lý đưa ra quyết định nhập thêm nguyên liệu một cách nhanh chóng, tránh được sự chậm trễ khi thiếu hụt nguyên liệu

Vì việc quản lý qua phần mềm cũng giúp việc xuất nguyên liệu để làm bánh được chính xác và nhanh chóng hơn, có thể kiểm soát chất lượng bánh làm ra bằng cách tự động tính toán số lượng nguyên liệu theo công thức.

Gọi món



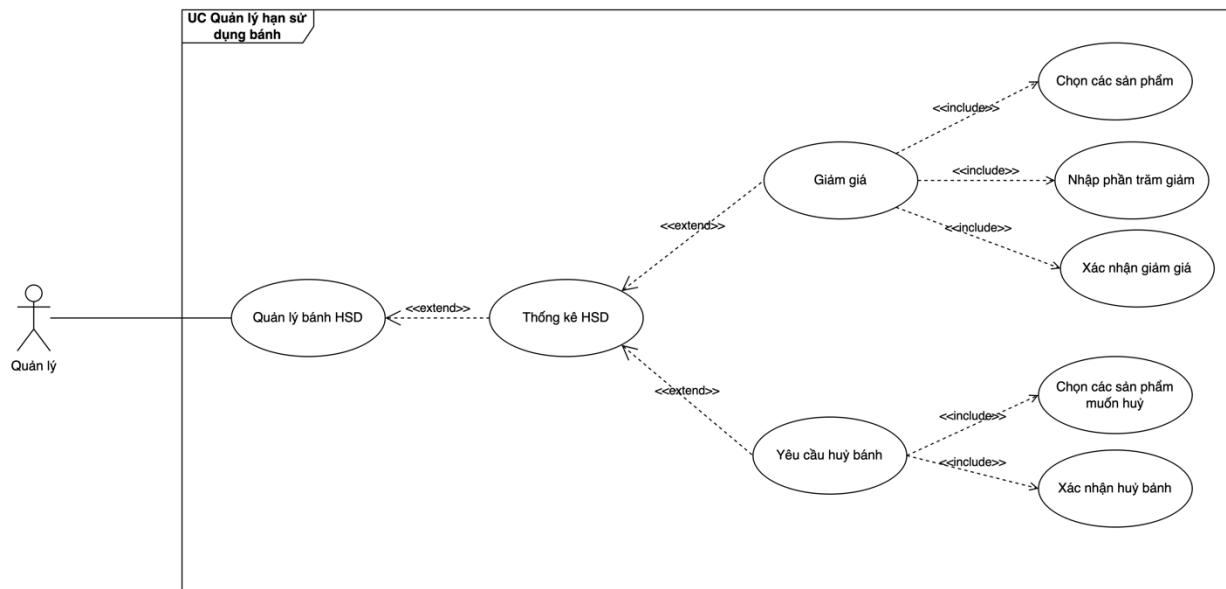
Lược đồ 3.6 giải pháp gọi món

1. Khách hàng quét mã để hiện menu để gọi món sau đó nhấn nút chọn thanh toán. Khách hàng có thể chọn thanh toán bằng cách quét mã hoặc chọn thanh toán tiền mặt.
2. Nhân viên sẽ tra cứu mã hóa đơn nếu khách hàng thanh toán bằng tiền mặt. Kết quả trả về số tiền cũng như các bánh mà khách mua.
3. Nhân viên nhấn xác nhận thanh toán cho khách hàng là đã thanh toán.

Kết luận: Việc mua bán trở nên thuận tiện hơn rất nhiều. Khách hàng có thể lựa chọn thoải mái mà không cần đến nhân viên cũng như có thể có nhiều kiểu thanh toán. Nhân viên dễ dàng phục vụ khách hàng mà không phải ghi nhớ nhiều. Việc thu chi không gây nhầm lẫn.

3.2. Định nghĩa các tương tác cần thiết cho phần mềm

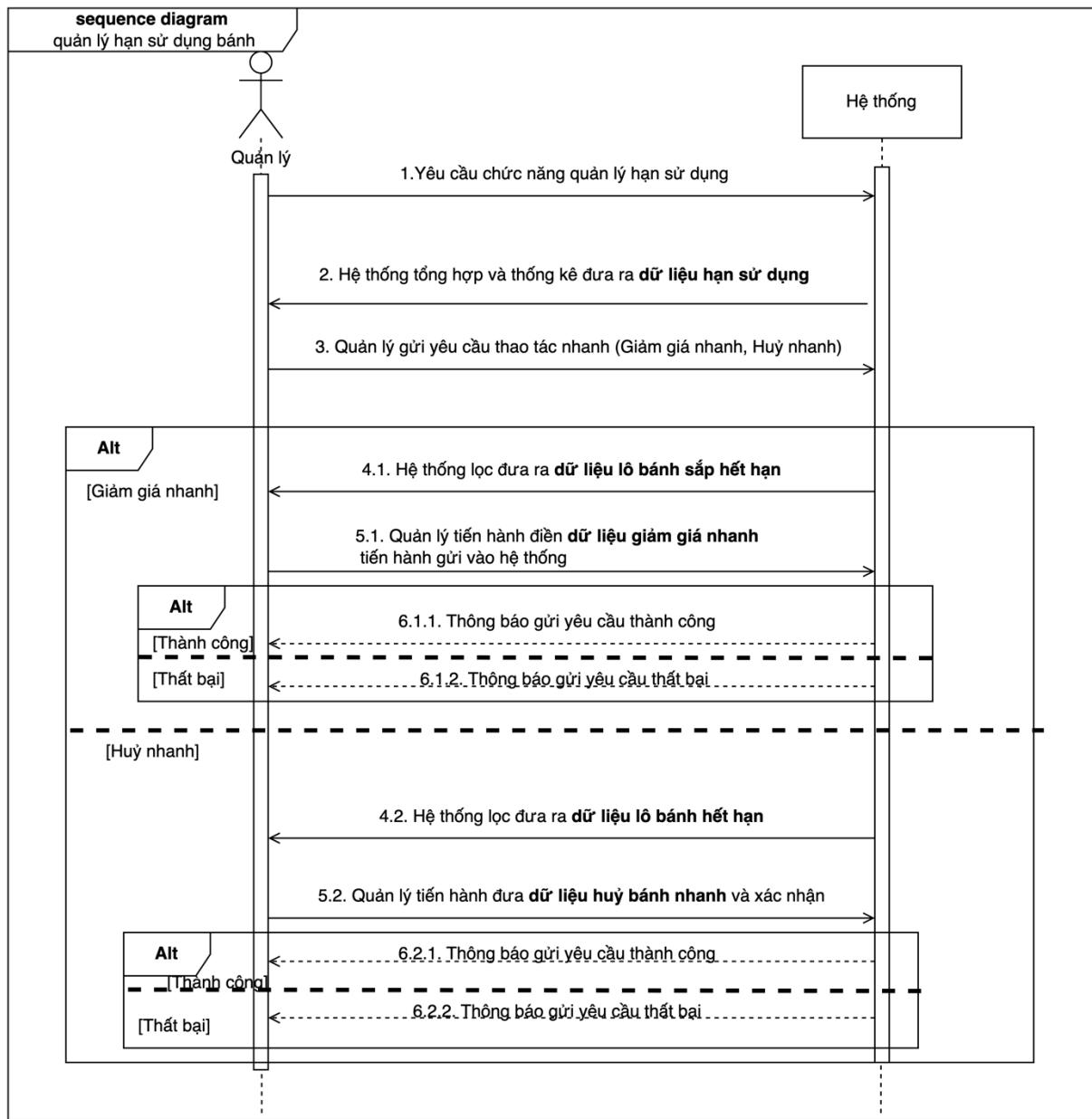
3.2.1. Usecase quản lý hạn sử dụng bánh



Lược đồ 3.7 usecase quản lý hạn sử dụng

USECASE NAME	Quản lý hạn sử dụng bánh
USECASE ID	UC-01
SCENARIO	Quản lý quản lý hạn sử dụng trên hệ thống
ACTOR	Quản lý
DESCRIPTION	Use case mô tả quy trình quản lý hạn sử dụng bánh trong hệ thống, bao gồm các tương tác từ quản lý
TRIGGER	<ul style="list-style-type: none"> Quản lý cần kiểm tra thông tin bánh Cập nhật trạng thái bánh trong hệ thống
PRECONDITION	<ul style="list-style-type: none"> Người dùng (quản lý) đã đăng nhập vào hệ thống Có quyền truy cập vào chức năng quản lý hạn sử dụng bánh Có sản phẩm bánh trong hệ thống cần xử lý

MAIN FLOW	<ol style="list-style-type: none"> 1. Quản lý yêu cầu chức năng quản lý hạn sử dụng 2. Hệ thống tổng hợp và thống kê hạn sử dụng của các lô bánh 3. Quản lý sẽ thực hiện chọn thao tác nhanh trên loại mong muốn <ol style="list-style-type: none"> 3.1. Giảm giá nhanh <ol style="list-style-type: none"> 3.1.1. Hệ thống đưa ra thông tin sản phẩm gần hết hạn 3.1.2. Quản lý nhập phần trăm muốn giảm, chọn các lô muốn giảm 3.1.3. Nhấn xác nhận 3.2. Huỷ nhanh <ol style="list-style-type: none"> 3.2.1. Hệ thống đưa ra các sản phẩm đã hết hạn 3.2.2. Quản lý chọn sản phẩm hết hạn muốn huỷ 3.2.3. Xác nhận huỷ
POSTCONDITION	<ul style="list-style-type: none"> • Thông tin bánh được cập nhật trong hệ thống
EXCEPTIONS	<ul style="list-style-type: none"> • Lỗi hệ thống • Hiển thị thông báo lỗi cụ thể • Thông báo cho người dùng • Dữ liệu không hợp lệ • Hiển thị thông báo lỗi chi tiết • Yêu cầu kiểm tra và nhập lại • Gợi ý định dạng đúng



Lược đồ 3. 8 tuần tự quản lý hạn sử dụng

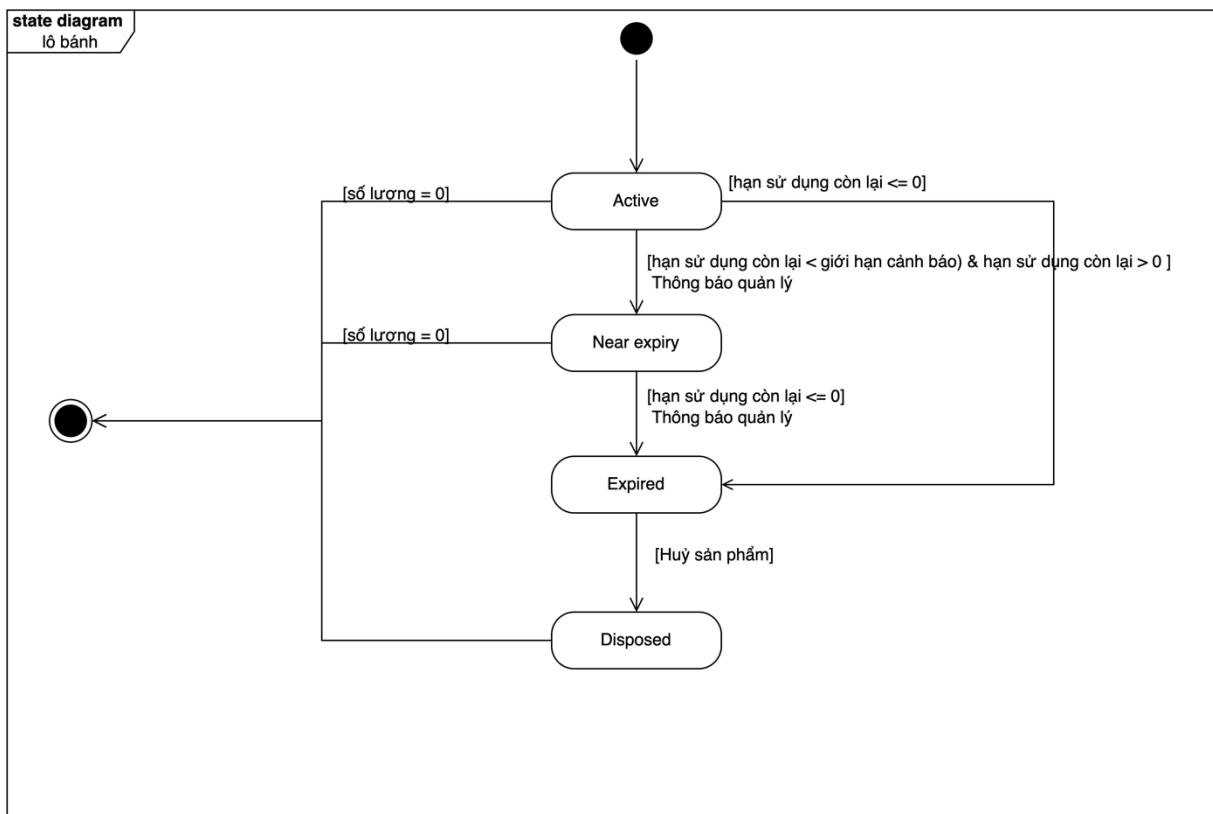
Từ điển dữ liệu [sequence diagram quản lý sử dụng bánh]

Tên	Chi tiết	Nguồn
Dữ liệu lô bánh	Id (Id bánh), Name (Tên bánh), Status (trạng thái lô bánh), Expiration_date (Ngày hết hạn), Quantity (số lượng còn lại của lô bánh)	Dữ liệu được lấy từ các bảng sau: (Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham

		chiếu bằng khoá ngoại product_id từ product_batches
Dữ liệu hạn sử dụng	Dữ liệu lô bánh đã lọc có trạng thái còn hạn dài (ACTIVE), sắp hết hạn (NEAR_EXPIRY), đã hết hạn (EXPIRED).	Dữ liệu được lấy từ các bảng sau:(Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham chiếu bằng khoá ngoại product_id từ product_batches thêm điều kiện lọc trạng thái
Dữ liệu lô bánh sắp hết hạn	Dữ liệu lô bánh đã lọc có trạng thái sắp hết hạn (NEAR_EXPIRY).	Dữ liệu được lấy từ các bảng sau:(Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham chiếu bằng khoá ngoại product_id từ product_batches thêm điều kiện lọc trạng thái
Dữ liệu lô bánh đã hết hạn	Dữ liệu lô bánh đã lọc có trạng thái đã hết hạn (EXPIRED).	Dữ liệu được lấy từ các bảng sau:(Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham chiếu bằng khoá ngoại product_id từ product_batches thêm điều kiện lọc trạng thái
Dữ liệu giảm giá nhanh	Discount (phần trăm giảm giá), isDefaultDiscount(có giảm giá mặc định đã định sẵn trong sản phẩm không),	Dữ liệu nhập từ tác nhân Quản lý thông qua form tương ứng

	endDate(ngày kết thúc giảm giá), isGetEndDate(lấy ngày kết thúc cuối cùng của lô bánh), batch_ids (id của các lô bánh được lấy từ danh sách các lô bánh hiển thị)	
Dữ liệu huỷ bánh nhanh	Batch_ids (id của các lô bánh được lấy từ danh sách các lô bánh hiển thị), note(ghi chú nội dung khi huỷ)	Dữ liệu nhập từ tác nhân Quản lý thông qua form tương ứng

Biểu đồ trạng thái lô bánh

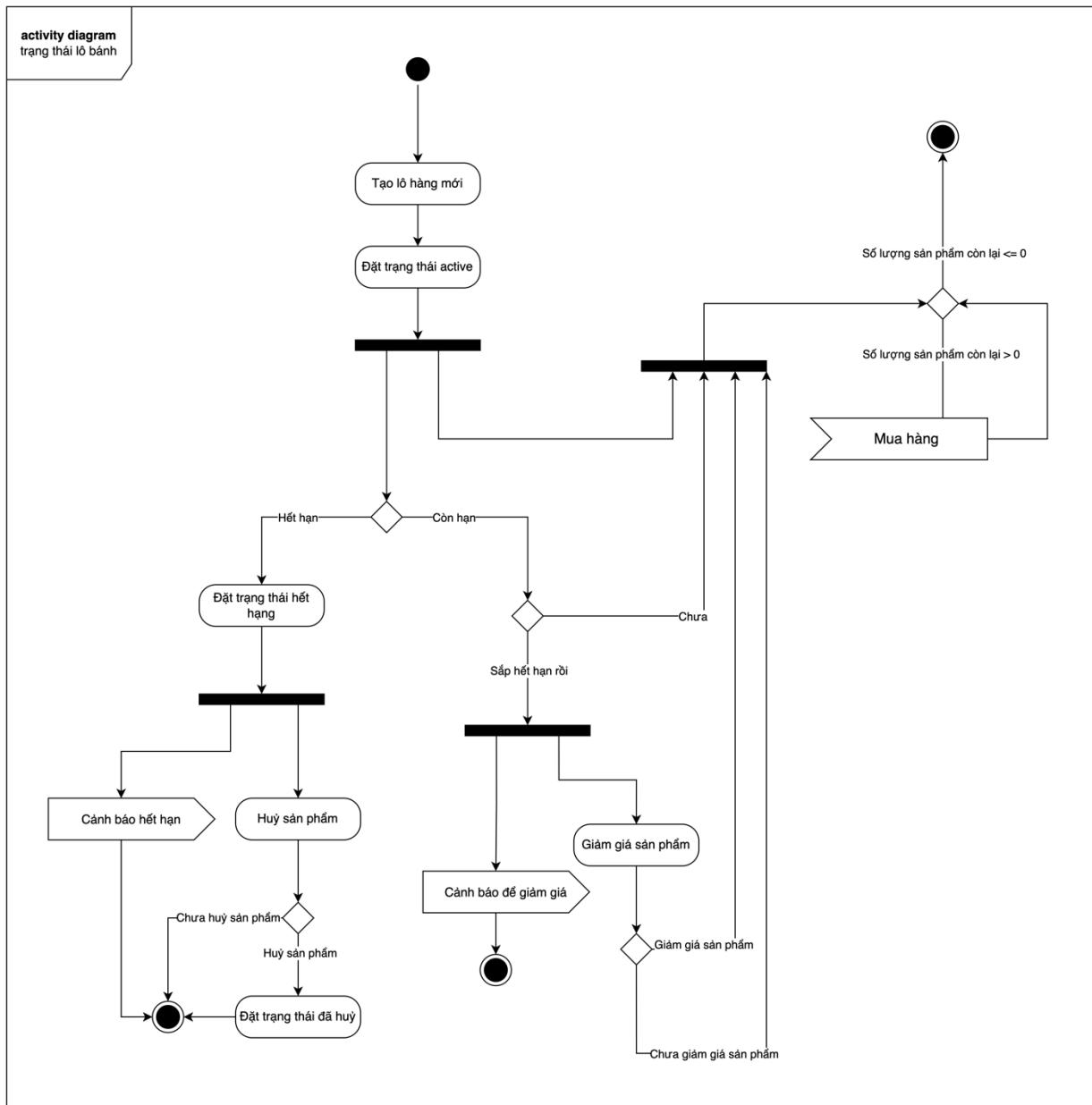


Lược đồ 3. 9 trạng thái lô bánh

Từ điển dữ liệu [state diagram lô bánh]

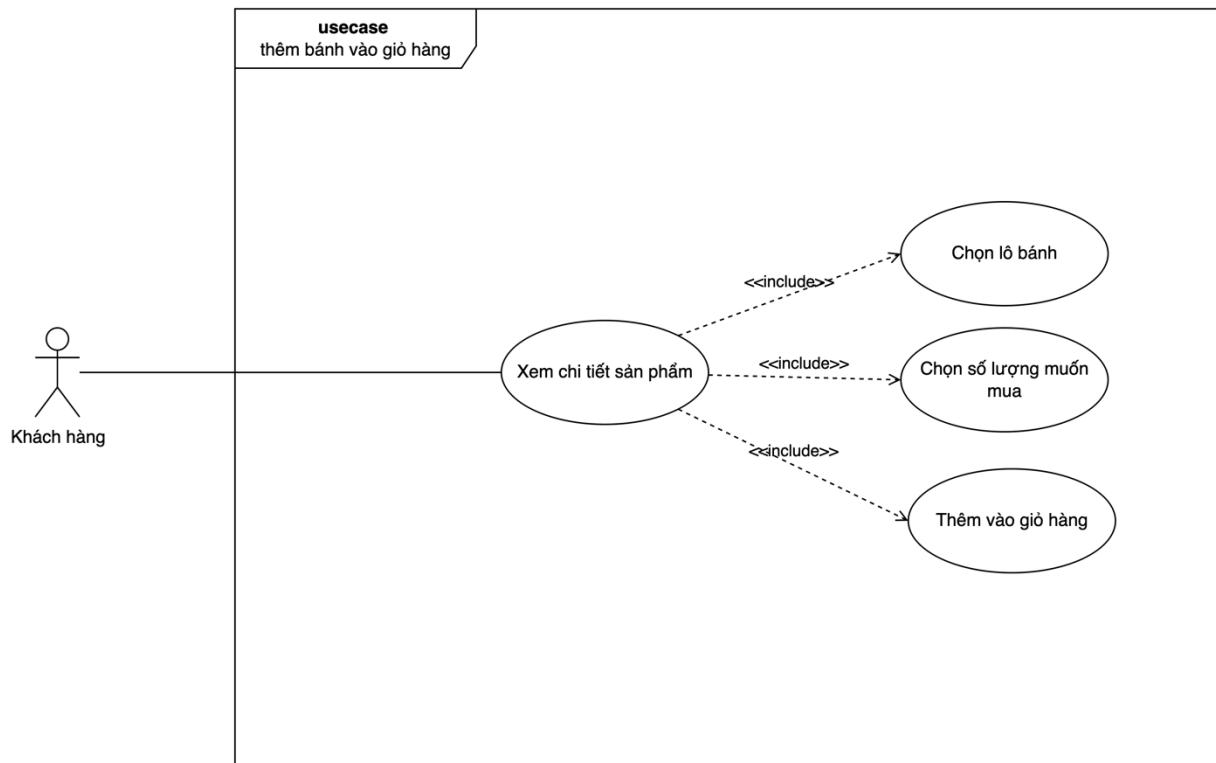
Tên	Chi tiết	Nguồn
Số lượng	Quantity(số lượng bánh còn lại của lô bánh)	Bảng product_batches
Ngày hết hạn	Expiration_date(ngày hết hạn của lô bánh)	Bảng product_batches
Giới hạn cảnh báo	Shelf_life_days_warning (ngày cảnh báo hạn sử dụng)	Bảng product
Hạn sử dụng còn lại	Expiration_date – date(now) (ngày hết hạn trừ ngày hiện tại = hạn sử dụng còn lại)	Bảng product_batches

Biểu đồ hoạt động cho trạng thái lô bánh



Lược đồ 3. 10 hoạt động của trạng thái lô bánh

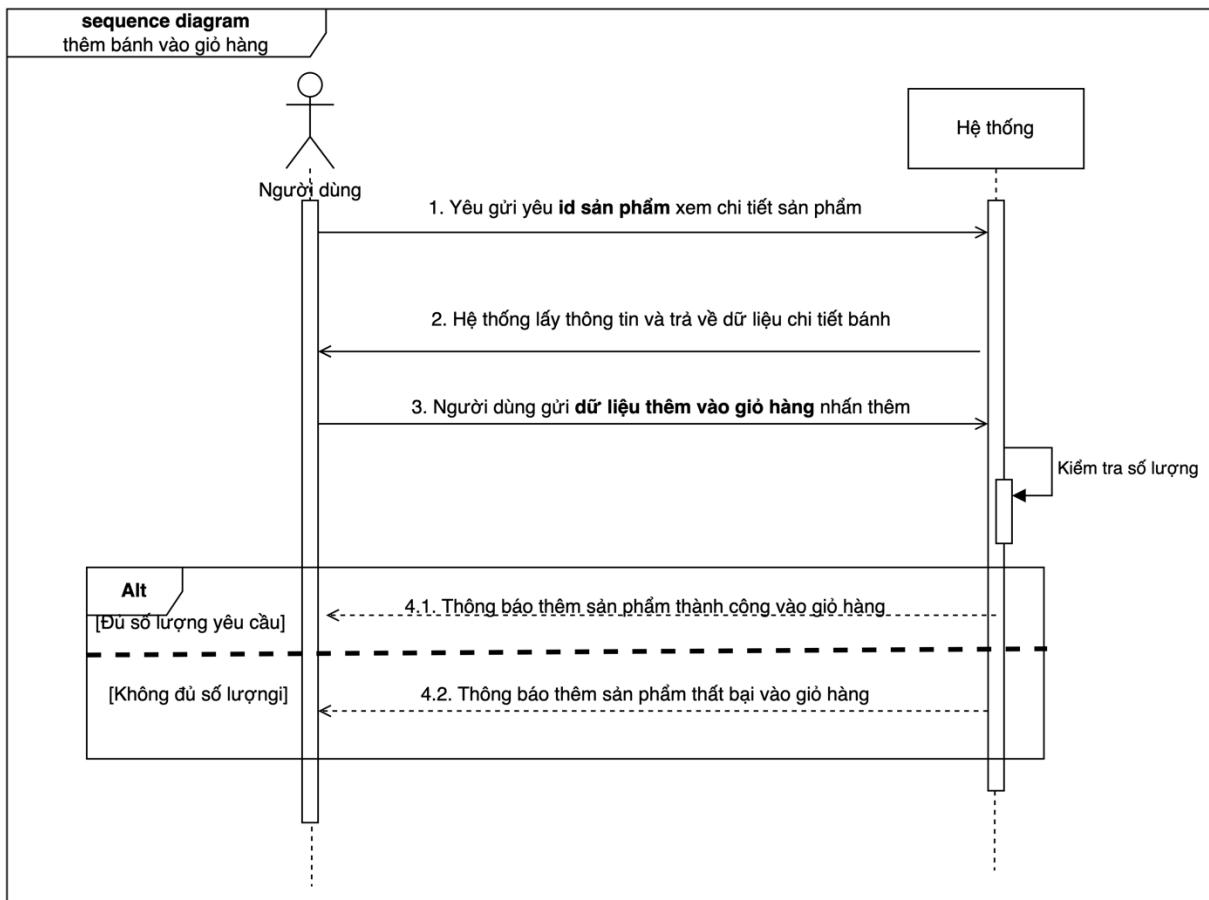
3.2.2. Usecase thêm bánh vào giỏ hàng



Lược đồ 3. 11 usecase thêm bánh vào giỏ hàng

USECASE NAME	Thêm bánh vào giỏ hàng
USECASE ID	UC-02
SCENARIO	Khách hàng xem chi tiết sản phẩm
ACTOR	Khách hàng
DESCRIPTION	Use case mô tả quy trình thêm sản phẩm vào giỏ hàng bao gồm các tương tác từ khách hàng:
TRIGGER	<ul style="list-style-type: none"> Người dùng cần xem chi tiết bánh Thêm vào giỏ hàng
PRECONDITION	<ul style="list-style-type: none"> Người dùng (khách hàng) đã vào hệ thống Có sản phẩm bánh trong hệ
MAIN FLOW	<ol style="list-style-type: none"> Khách hàng yêu cầu xem chi tiết sản phẩm

	<p>2. Hệ thống lấy dữ liệu chi tiết sản phẩm gửi ra cho người dùng</p> <p>3. Người dùng tiến hành gửi dữ liệu thêm vào giỏ hàng nhấn thêm</p> <p>4. Thông báo</p> <p>4.1. Thêm giỏ hàng thành công</p> <p>4.2. Thêm giỏ hàng thất bại</p>
POSTCONDITION	<ul style="list-style-type: none"> Bánh được thêm vào giỏ hàng
EXCEPTIONS	<ul style="list-style-type: none"> Lỗi hệ thống Hiển thị thông báo lỗi cụ thể Thông báo cho người dùng Dữ liệu không hợp lệ Hiển thị thông báo lỗi chi tiết Yêu cầu kiểm tra và nhập lại Gợi ý định dạng đúng

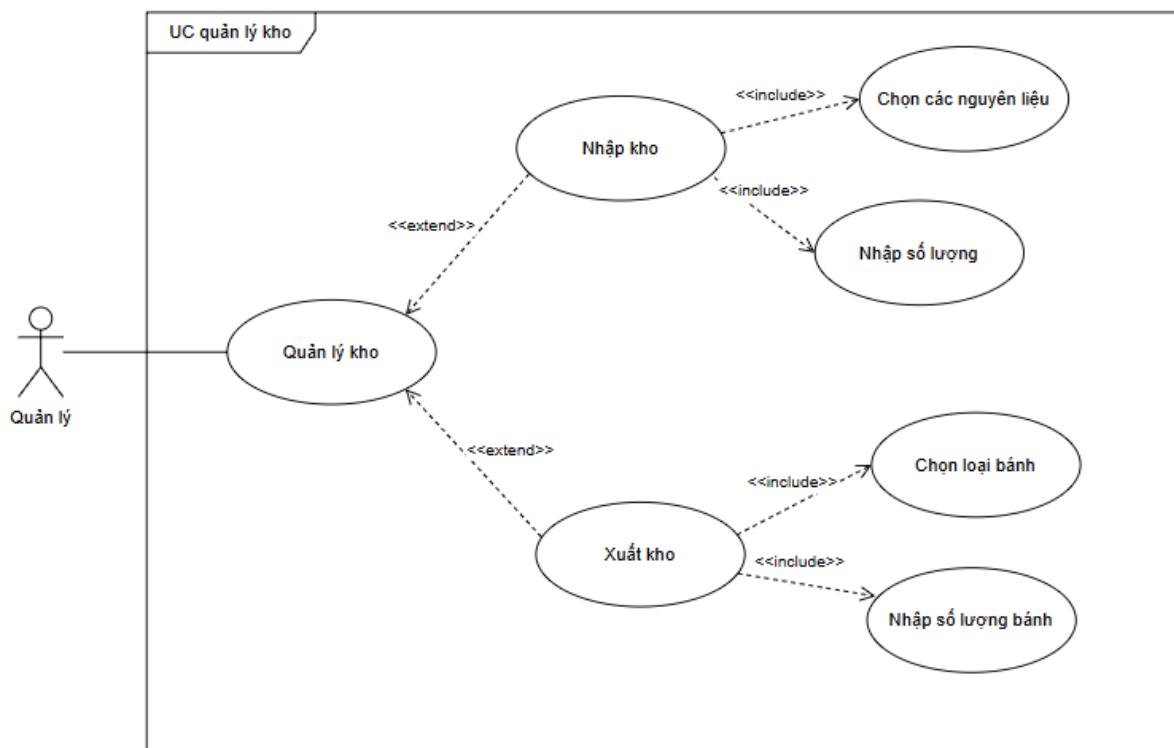


Lược đồ 3. 12 tuần tự thêm bánh vào giỏ hàng

Từ điển dữ liệu [sequence diagram thêm bánh vào giỏ hàng]

Tên	Chi tiết	Nguồn
Dữ liệu thêm vào giỏ hàng	Product_batch_id (id lô bánh), quantity (số lượng)	Dữ liệu nhập từ tác nhân khách hàng từ form chi tiết sản phẩm với id lô bánh được lấy từ lô bánh người dùng chọn của chi tiết sản phẩm người dùng đang xem và số lượng là số lượng đang chọn

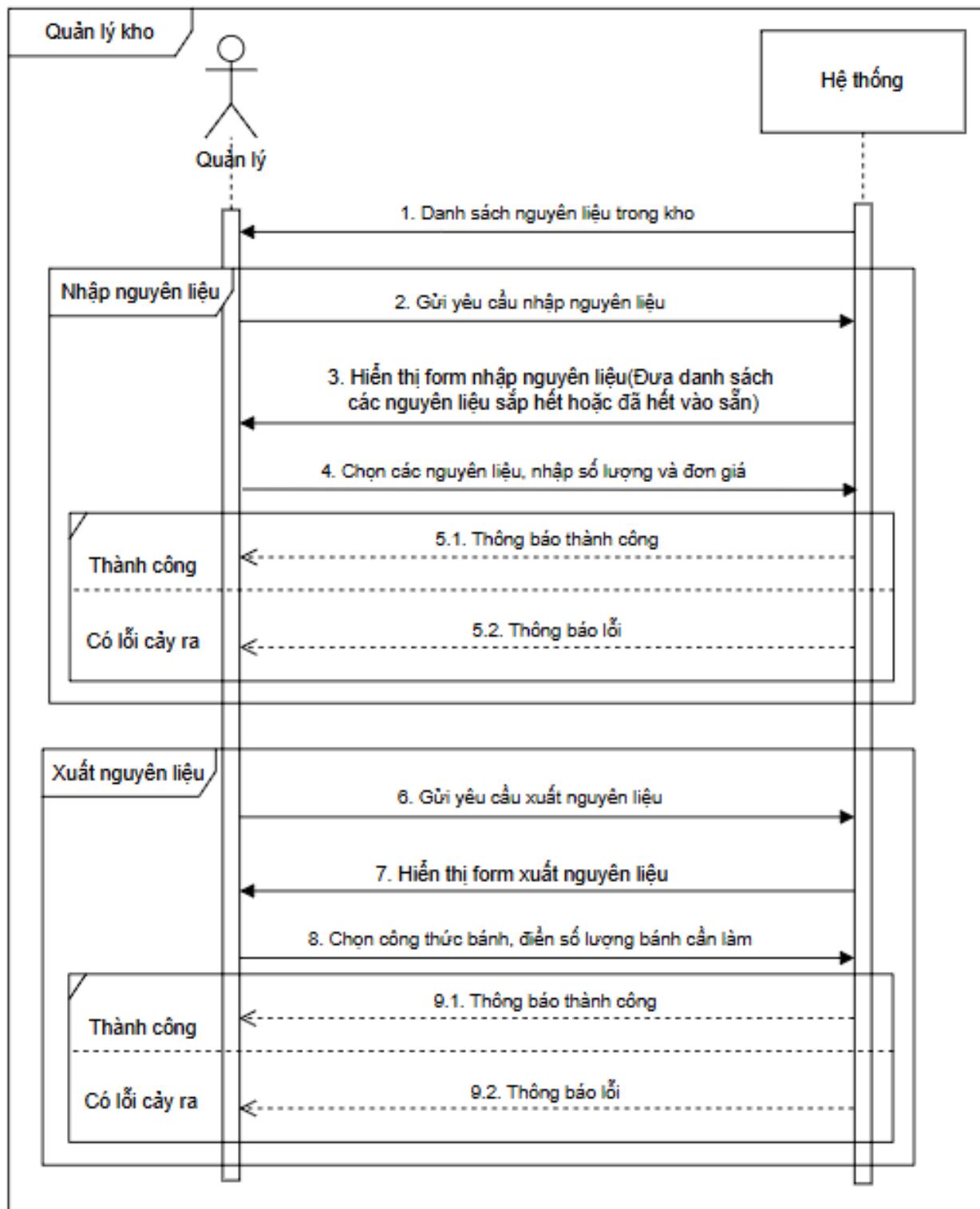
3.2.3. Usecase quản lý kho nguyên liệu



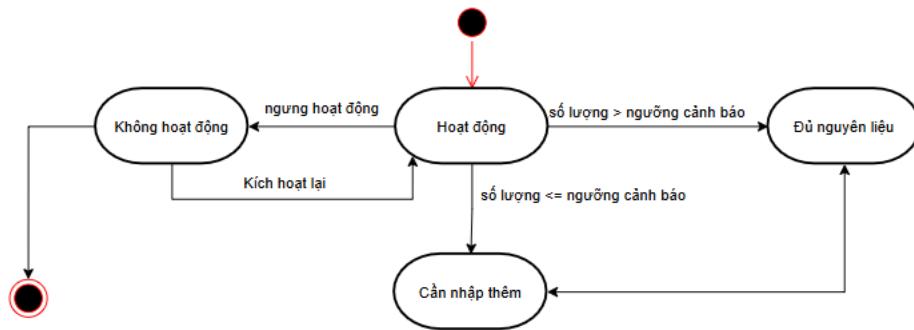
Lược đồ 3. 13 usecase quản lý kho

USECASE NAME	Quản lý kho
USECASE ID	UC-03
SCENARIO	Quản lý sẽ quản lý các nguyên liệu trong kho
ACTOR	Quản lý
DESCRIPTION	Use case mô tả quy trình quản lý các nguyên liệu trong kho
TRIGGER	Quản lý nhập thêm nguyên liệu vào kho Quản lý muốn xuất nguyên liệu trong kho để làm bánh
PRECONDITION	Người quản lý đã đăng nhập vào hệ thống Có quyền truy cập vào chức năng quản lý kho
MAIN FLOW	Xem và kiểm tra các nguyên liệu trong kho Xem các nguyên liệu đang có cảnh báo(sắp hết hoặc đã hết) Kiểm tra số lượng tồn mỗi nguyên liệu Thêm các nguyên liệu mới, xóa hoặc sửa các nguyên liệu đã có trong hệ thống Nhập nguyên liệu

	Xuất nguyên liệu để làm bánh(dựa theo công thức) Xem lịch sử nhập nguyên liệu Xem lịch sử xuất nguyên liệu
POSTCONDITION	Thông tin nguyên liệu được lưu lại hoặc cập nhật trong hệ thống Lịch sử nhập hoặc xuất nguyên liệu được lưu lại
EXCEPTIONS	Lỗi hệ thống Hiển thị thông báo lỗi cụ thể Ghi log lỗi Hướng dẫn thao tác thay thế Xung đột quyền hạn Kiểm tra và hiển thị quyền hạn hiện tại Thông báo cho người dùng Hướng dẫn liên hệ cấp trên Dữ liệu không hợp lệ Hiển thị thông báo lỗi chi tiết Yêu cầu kiểm tra và nhập lại Gợi ý định dạng đúng

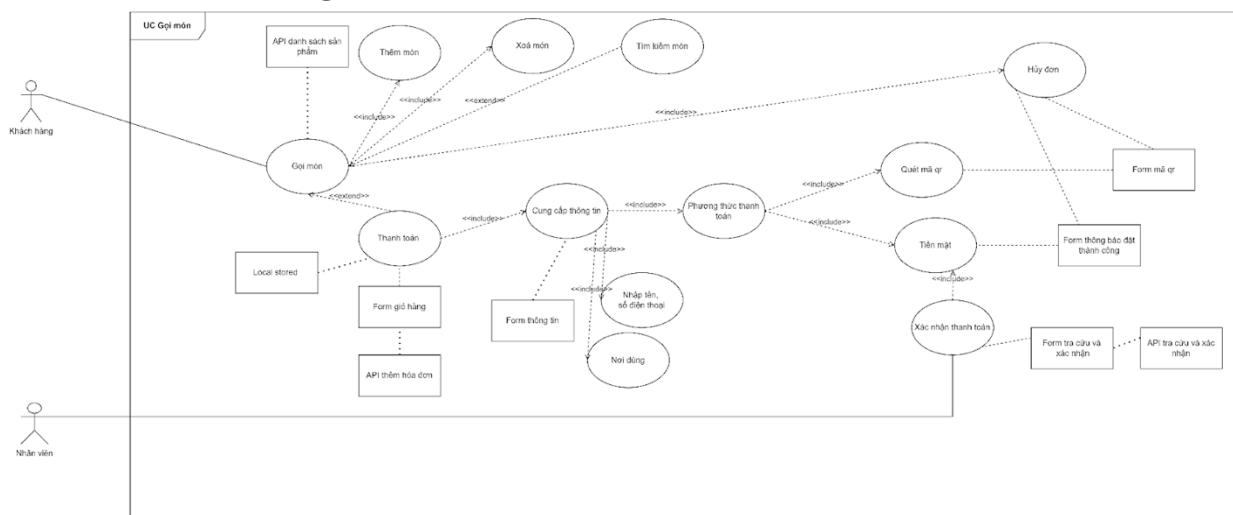


Lược đồ 3. 14 tuần tự quản lý kho

Biểu đồ trạng thái nguyên liệu

Lược đồ 3. 15 trạng thái nguyên liệu

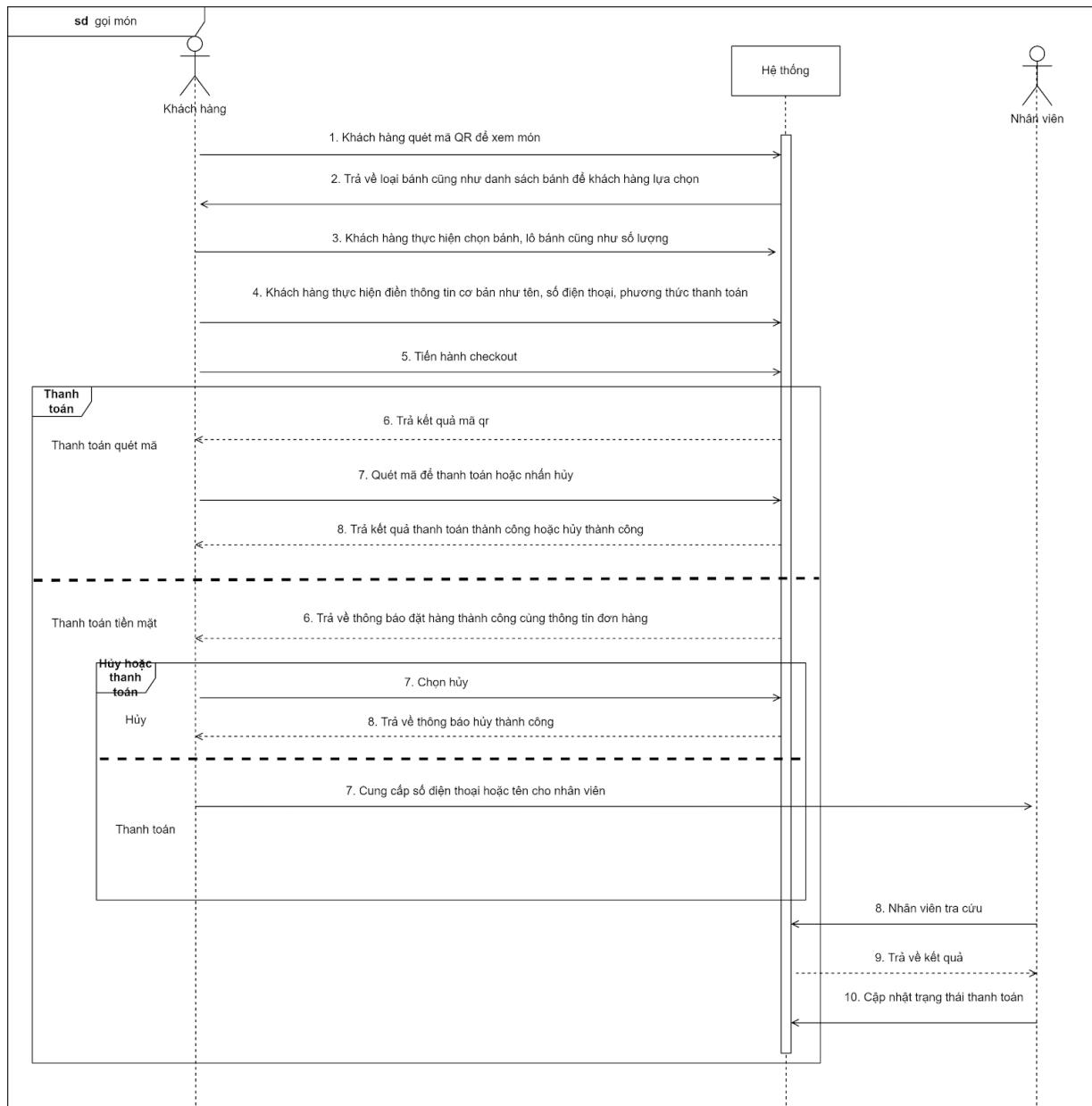
3.2.4. Usecase gọi món



Lược đồ 3. 16 usecase gọi món

USECASE NAME	Gọi món
USECASE ID	UC-04
SCENARIO	Khách hàng sẽ thực hiện gọi món
ACTOR	Quản lý
DESCRIPTION	Use case mô tả quy trình khách hàng gọi món
TRIGGER	
PRECONDITION	Có camera để quét mã qr
MAIN FLOW	Khách hàng thực hiện quét mã qr trên bàn hoặc tại quầy Lựa chọn thêm bột sản phẩm Lựa chọn thanh toán tiền mặt hoặc quét mã qr để thanh toán Thực hiện thanh toán Có thể hủy nếu chưa thanh toán Nhân viên xác nhận thanh toán cho thanh toán tiền mặt
POSTCONDITION	Thông tin hóa đơn được lưu lại trong hệ thống
EXCEPTIONS	Lỗi hệ thống Hiển thị thông báo lỗi cụ thể Ghi log lỗi Hướng dẫn thao tác thay thế Xung đột quyền hạn

	Kiểm tra và hiển thị quyền hạn hiện tại Thông báo cho người dùng Dữ liệu không hợp lệ Hiển thị thông báo lỗi chi tiết Yêu cầu kiểm tra và nhập lại Gợi ý định dạng đúng
--	--



Lược đồ 3.17 tuần tự gọi món

3.3. Định nghĩa yêu cầu và ràng buộc đối với phần mềm

3.3.1. Yêu cầu từ môi trường nghiệp vụ (Business)

UseCase	Req-ID	Nội dung yêu cầu	Stack-Holder
UC-01	B01.1	Cung cấp rõ phần trăm giảm giá	Quản lý
	B01.2	Không có sản phẩm hết hạn	Quản lý
	B01.3	Không có sản phẩm sắp tới hạn	Quản lý

3.3.2. Yêu cầu từ môi trường vận hành

UseCase	Req-ID	Nội dung yêu cầu	Stack-Holder
UC-01 UC-03	NF01.1	Hệ thống cần đăng nhập và phân quyền	
	NF01.2	Giao diện rõ ràng, đơn giản, hiển thị đầy đủ thông tin	
UC-02 UC-04	NF02.1	Hệ thống cần thiết kế với giao diện đơn giản dễ sử dụng cho khách hàng	

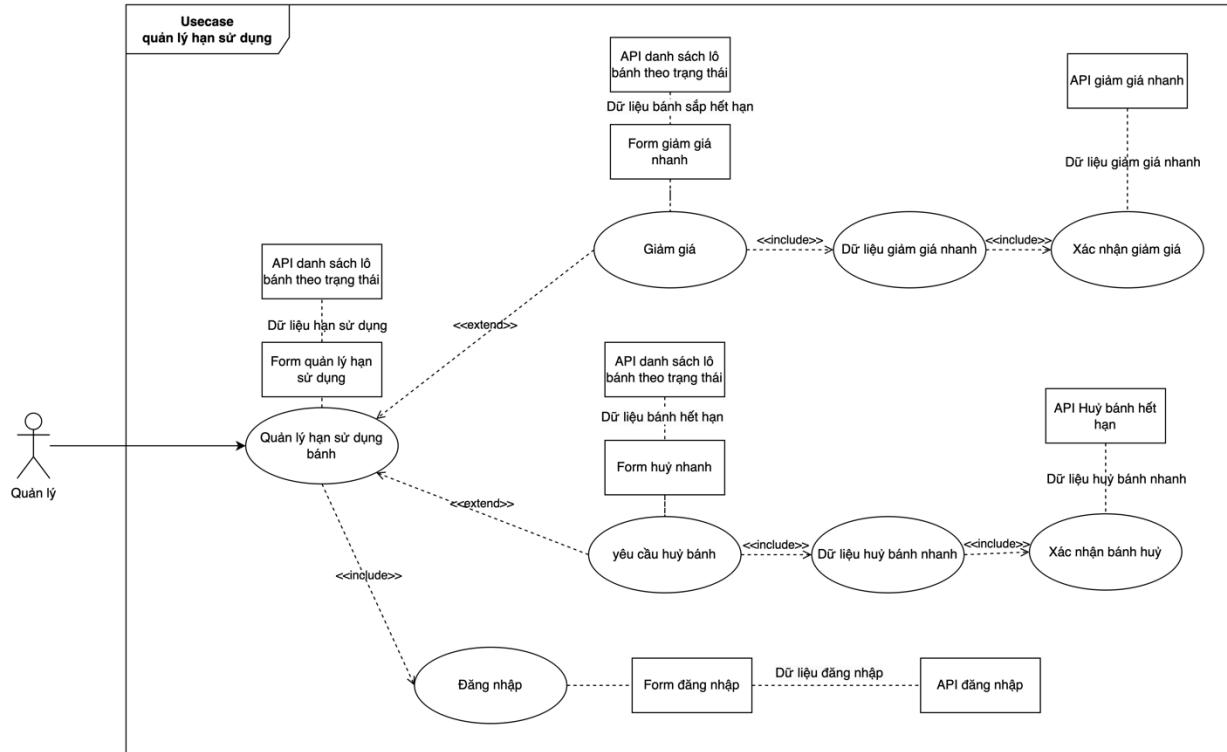
3.3.3. Yêu cầu từ môi trường phát triển. (Development)

ID	Đối tượng	Nội dung yêu cầu	Stack-Holder
TD01	Software	Viết code trên IntelliJ và WebStorm	Dev team
TD02	Software	Dùng Java 23, TypeScript, ReactTS, SpringBoot	Dev team

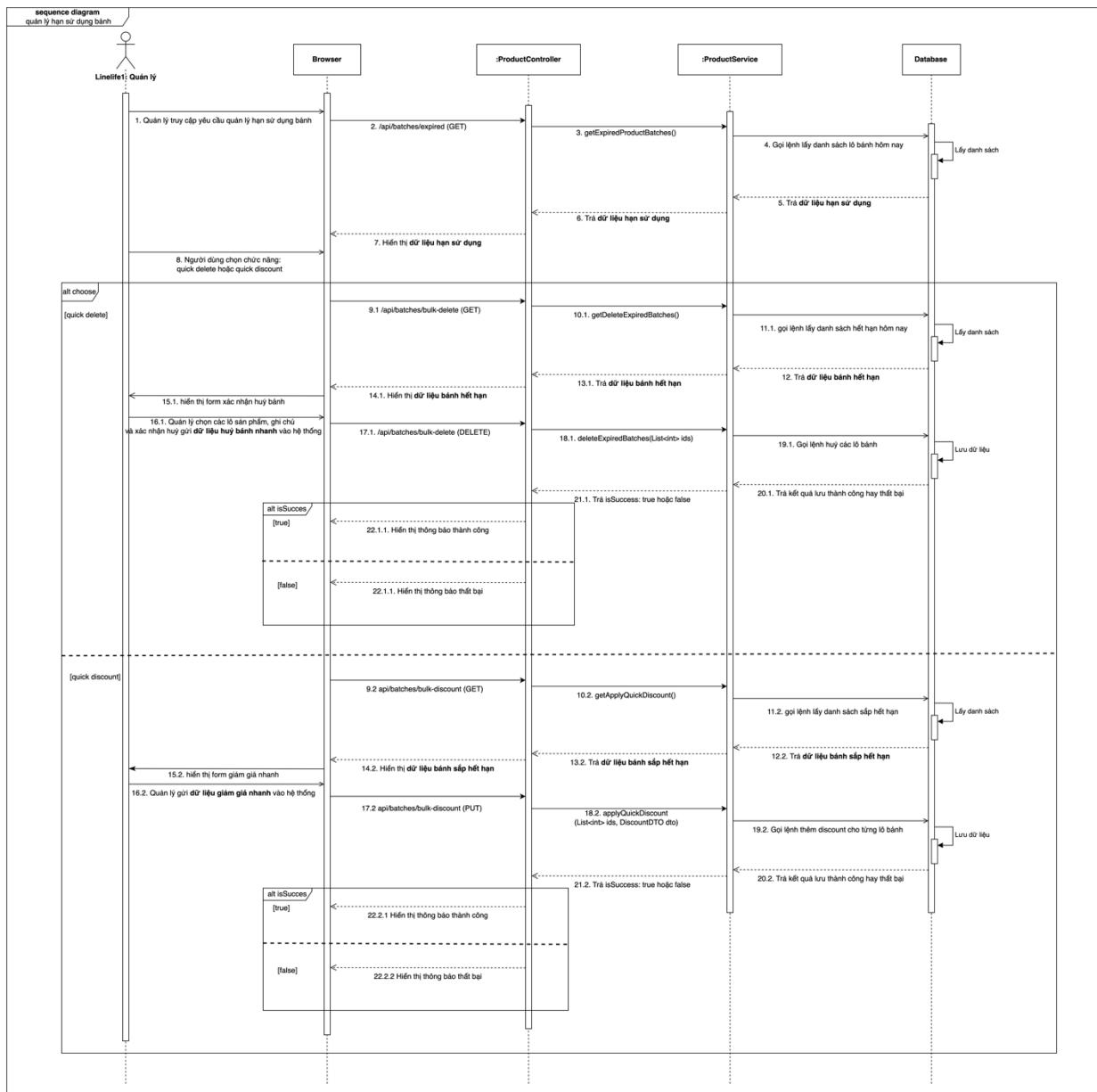
CHƯƠNG 4. :THIẾT KẾ PHẦN MỀM

4.1. Lượt đồ use-case cho thiết kế phần mềm

4.1.1. Use-case quản lý hạn sử dụng bánh



Lược đồ 4. 1 usecase quản lý hạn sử dụng



Lược đồ 4. 2 tuần tự hạn sử dụng

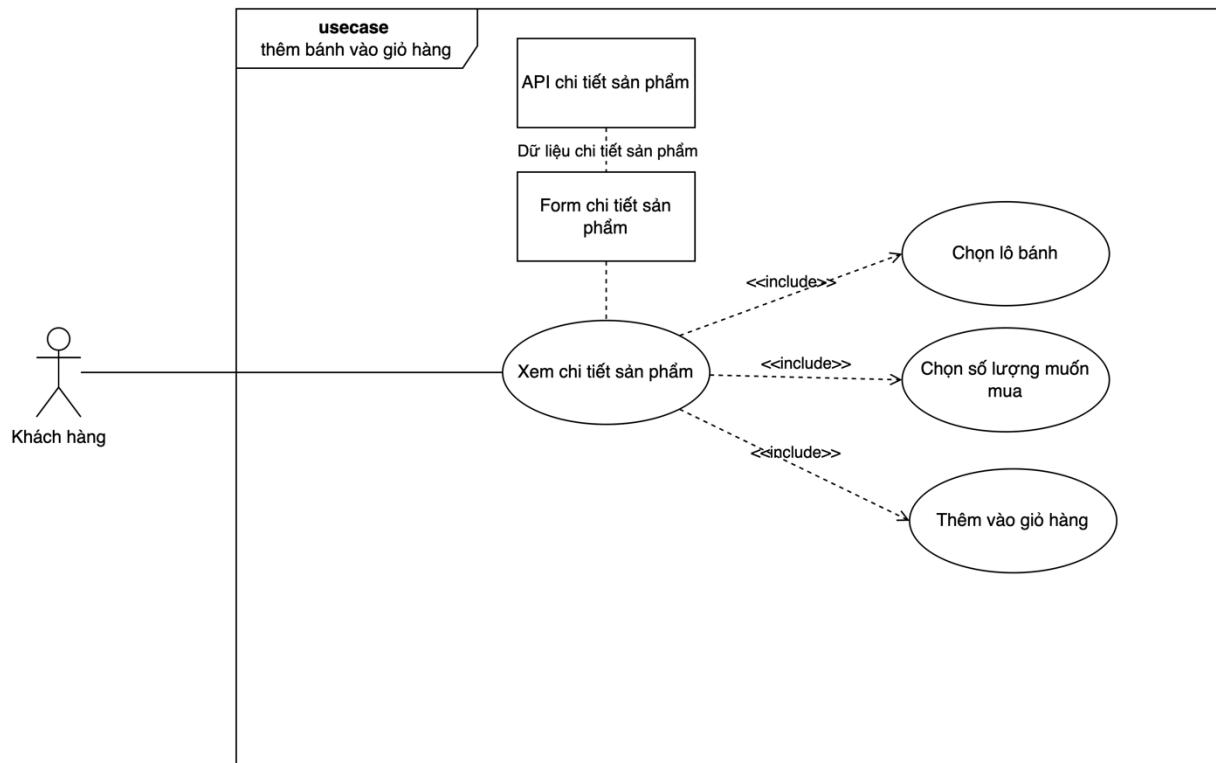
Từ điển dữ liệu [Usecase quản lý hạn sử dụng]

Tên	Chi tiết	Nguồn
Dữ liệu lô bánh	Id (Id bánh), Name (Tên bánh), Status (trạng thái lô bánh), Expiration_date (Ngày hết hạn), Quantity (số lượng còn lại của lô bánh)	Dữ liệu được lấy từ các bảng sau: (Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham chiếu bằng khoá ngoại

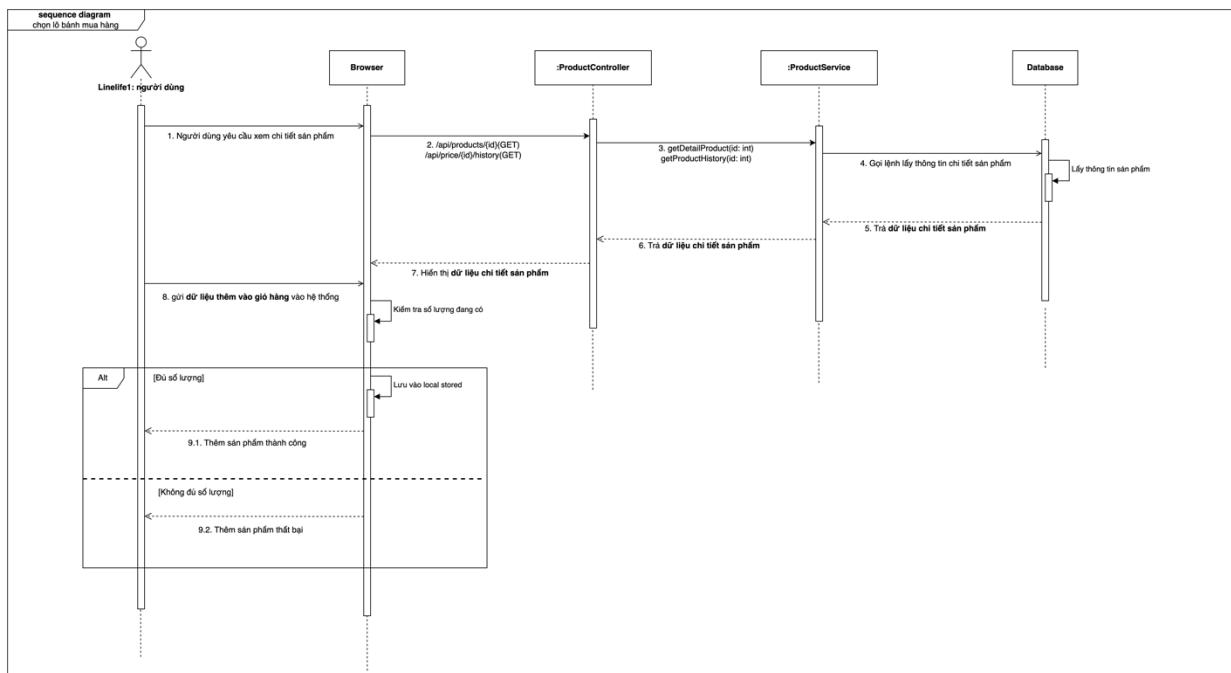
		product_id từ product_batches
Dữ liệu hạn sử dụng	Dữ liệu lô bánh đã lọc có trạng thái còn hạn dài (ACTIVE), sắp hết hạn (NEAR_EXPIRY), đã hết hạn (EXPIRED).	Dữ liệu được lấy từ các bảng sau:(Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham chiếu bằng khoá ngoại product_id từ product_batches thêm điều kiện lọc trạng thái
Dữ liệu lô bánh sắp hết hạn	Dữ liệu lô bánh đã lọc có trạng thái sắp hết hạn (NEAR_EXPIRY).	Dữ liệu được lấy từ các bảng sau:(Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham chiếu bằng khoá ngoại product_id từ product_batches thêm điều kiện lọc trạng thái
Dữ liệu lô bánh đã hết hạn	Dữ liệu lô bánh đã lọc có trạng thái đã hết hạn (EXPIRED).	Dữ liệu được lấy từ các bảng sau:(Id, Status, Expiration_date, quantity) được lấy từ bảng product_batches, Name được lấy từ products tham chiếu bằng khoá ngoại product_id từ product_batches thêm điều kiện lọc trạng thái
Dữ liệu giảm giá nhanh	Discount (phần trăm giảm giá), isDefaultDiscount(có giảm giá mặc định đã định sẵn trong sản phẩm không), endDate(ngày kết thúc	Dữ liệu nhập từ tác nhân Quản lý thông qua form giảm giá nhanh

	giảm giá), isGetEndDate(lấy ngày kết thúc cuối cùng của lô bánh), batch_ids (id của các lô bánh được lấy từ danh sách các lô bánh hiển thị)	
Dữ liệu huỷ bánh nhanh	Batch_ids (id của các lô bánh được lấy từ danh sách các lô bánh hiển thị), note(ghi chú nội dung khi huỷ)	Dữ liệu nhập từ tác nhân Quản lý thông qua form huỷ nhanh
Dữ liệu đăng nhập	Email (email đăng nhập), password(mật khẩu)	Dữ liệu nhập từ tác nhân Quản lý thông qua form đăng nhập

4.1.2. Use-case chi tiết sản phẩm



Lược đồ 4. 3 usecase thêm bánh vào giỏ hàng

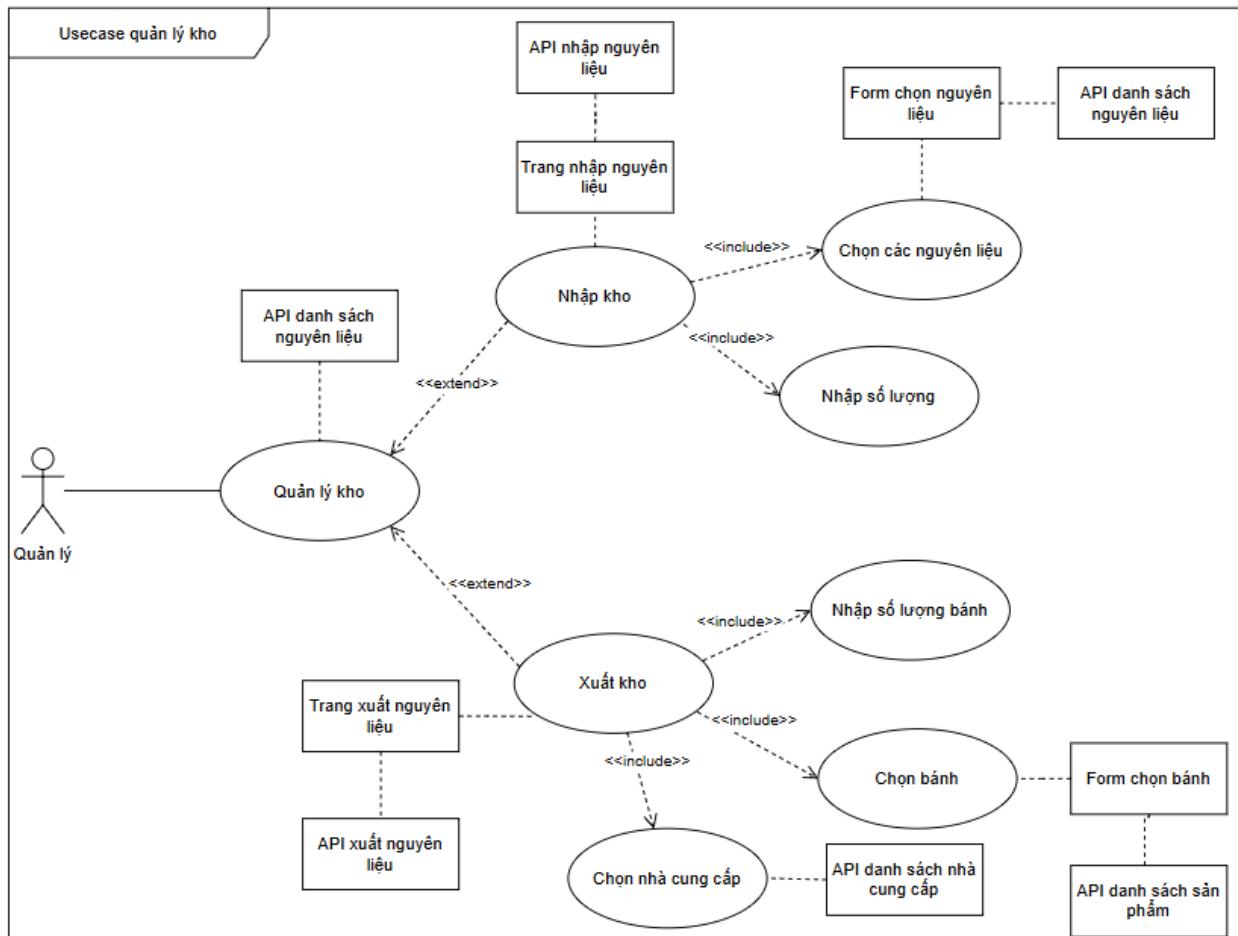


Lược đồ 4. 4 tuần tự thêm bánh vào giỏ hàng

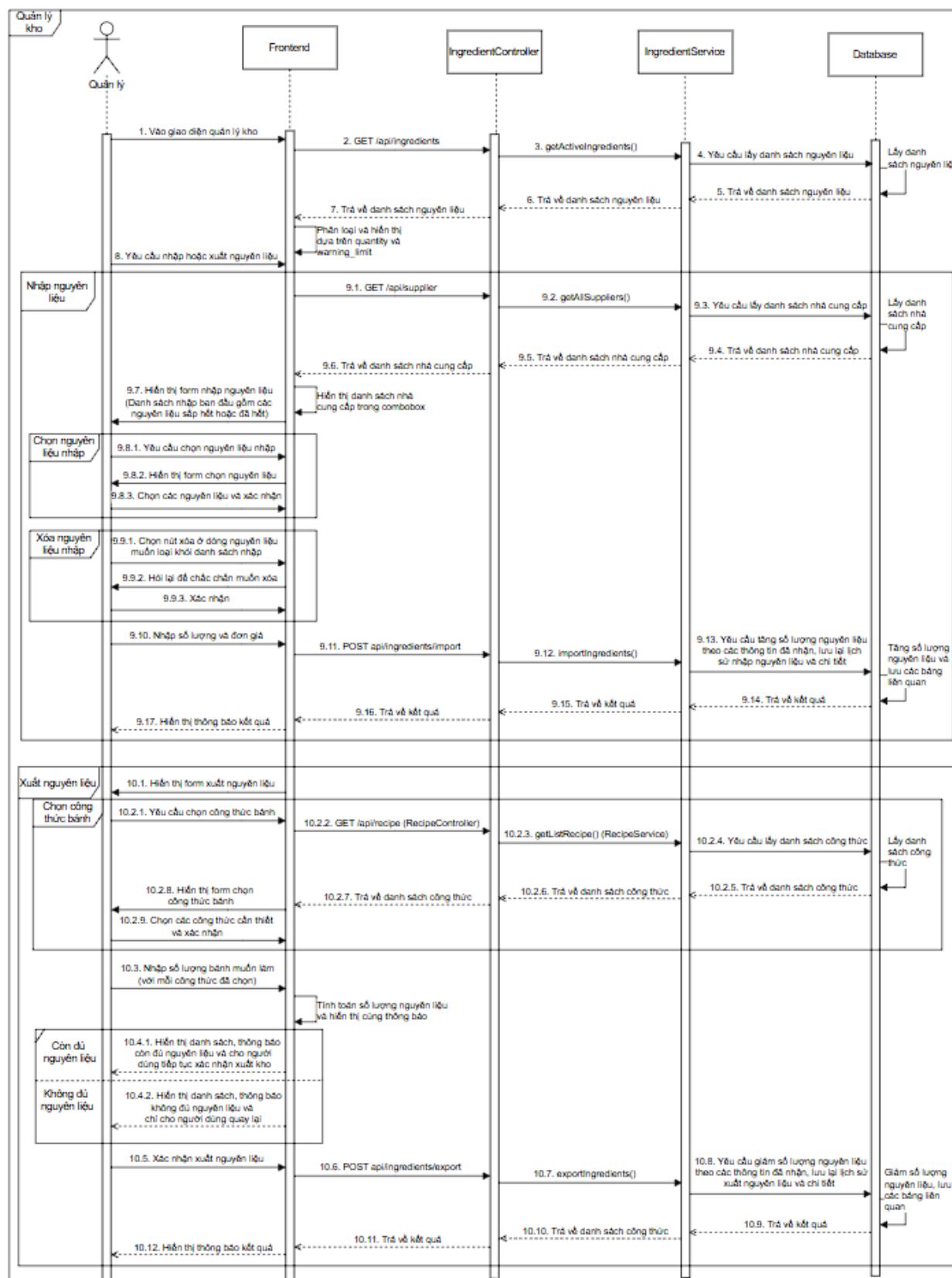
Tùy biến dữ liệu [Usecase chi tiết đơn sản phẩm]

Tên	Chi tiết	Nguồn
Dữ liệu chi tiết sản phẩm	Id (id sản phẩm), name(tên sản phẩm), category { id (id danh mục), name (tên danh mục)}, productBatches[{id (id lô bánh), expiration_date(ngày hết hạn), discount(giảm giá của lô bánh)]} imageUrl[imageUrl(url ảnh),] priceHistory[{id (id lịch sử), Effect_date(ngày áp dụng)}]	Dữ liệu được lấy từ các bảng: products, categories, images, product_batches, product_history được lấy và đi qua API tương ứng
Dữ liệu thêm vào giỏ hàng	Product_batch_id (id lô bánh), quantity (số lượng)	Dữ liệu nhập từ tác nhân khách hàng từ form chi tiết sản phẩm với id lô bánh được lấy từ lô bánh người dùng chọn của chi tiết sản phẩm người dùng đang xem và số lượng là số lượng đang chọn

4.1.3. Use-case quản lý kho nguyên liệu

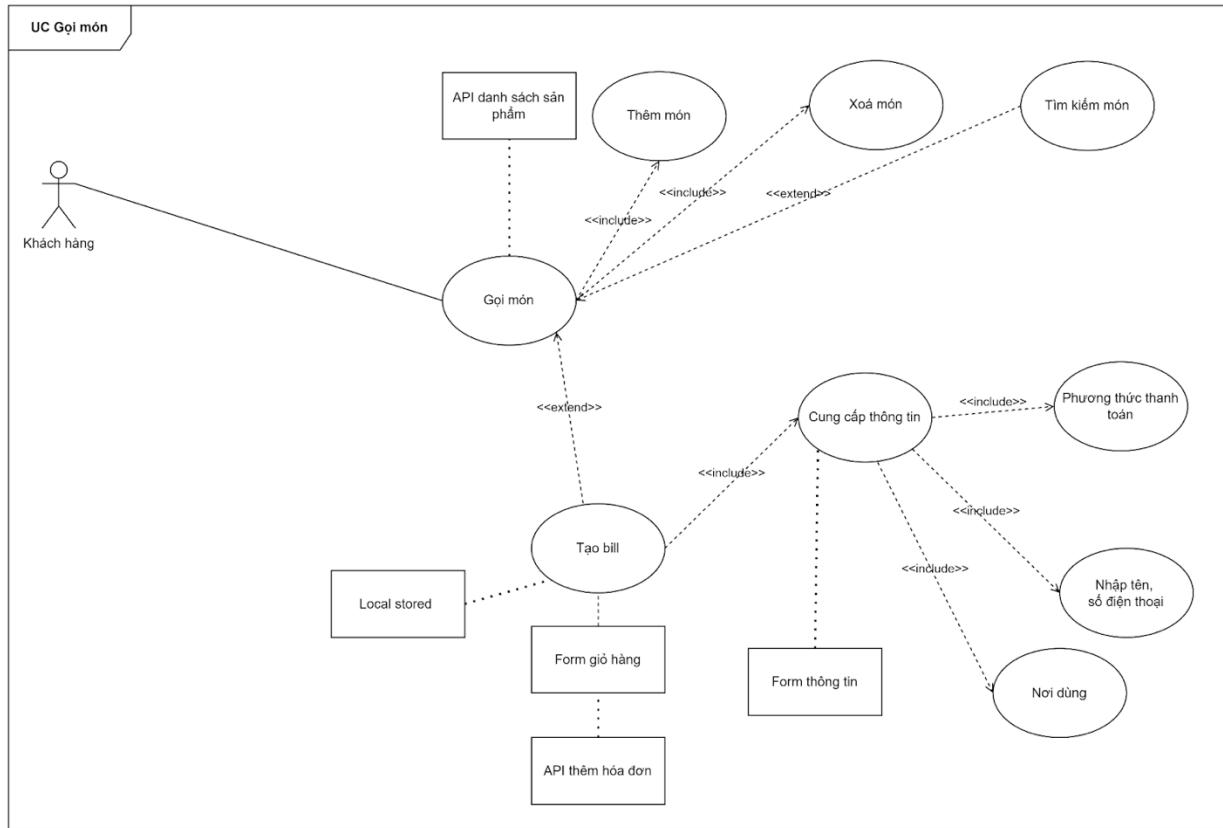


Lược đồ 4.5 usecase quản lý kho nguyên liệu

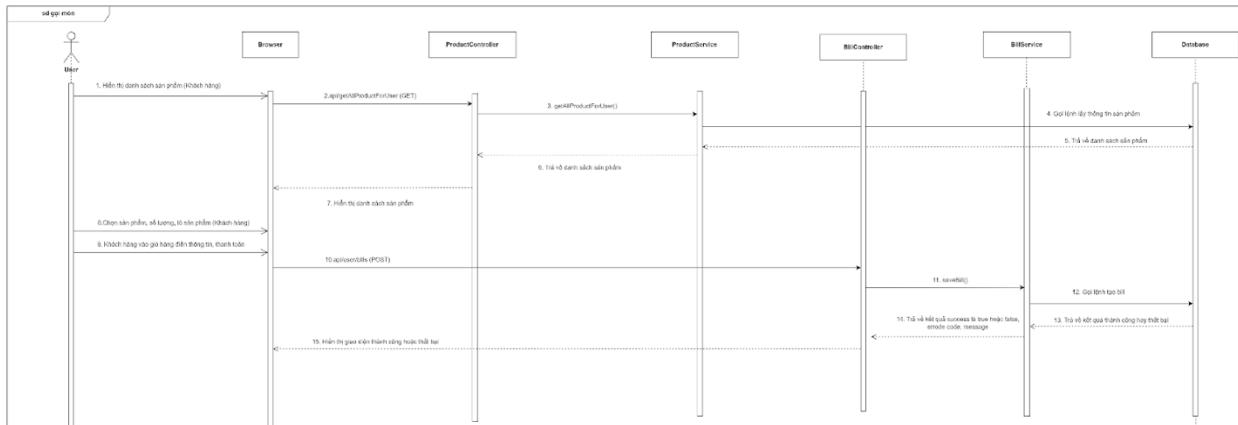


Lược đồ 4. 6 tuần tự quản lý kho nguyên liệu

4.1.4. Use-case gọi món

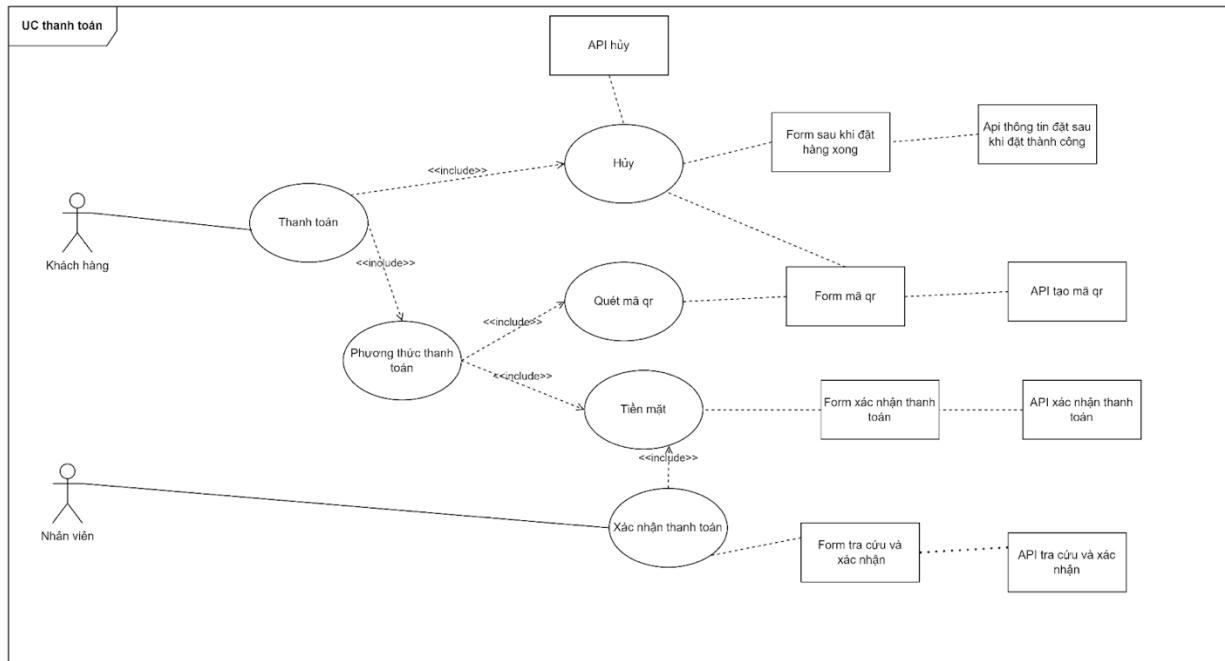


Lược đồ 4. 7 usecase gọi món

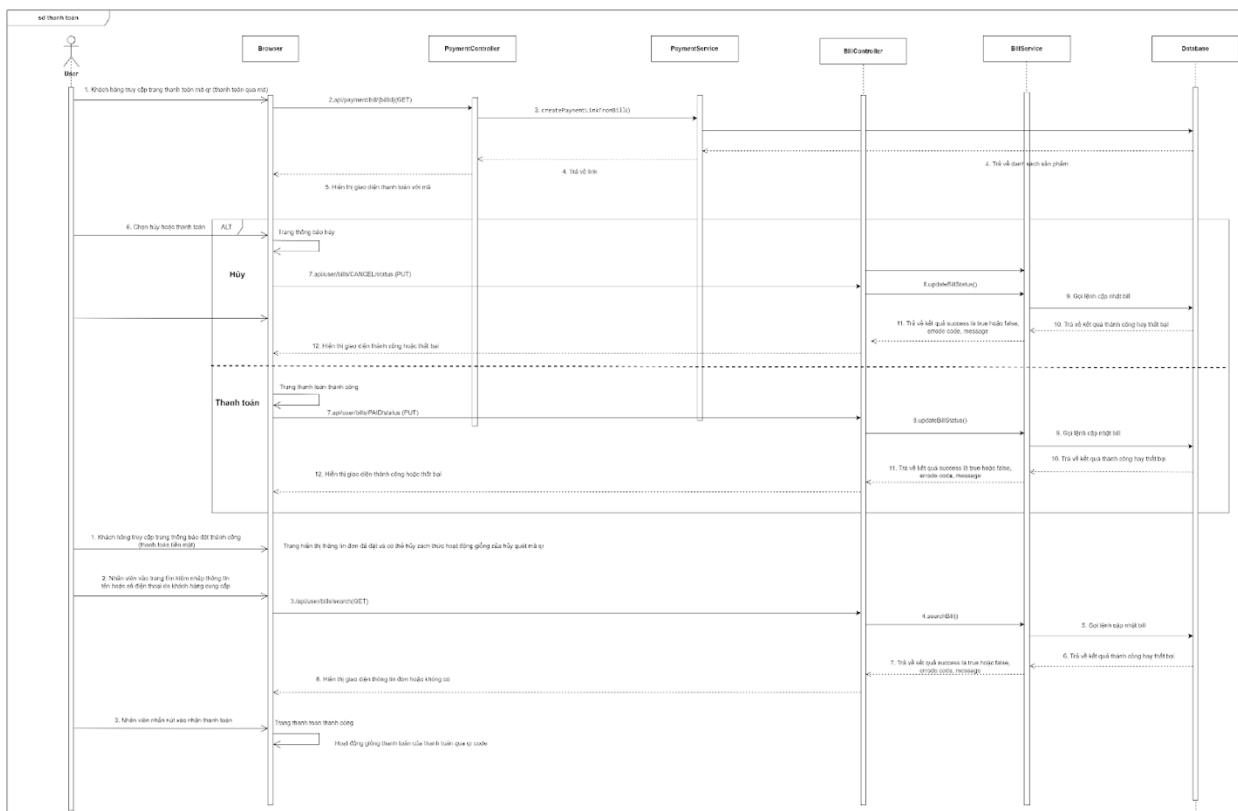


Lược đồ 4. 8 tuần tự gọi món

4.1.5. Use-case thanh toán



Lược đồ 4.9 usecase thanh toán



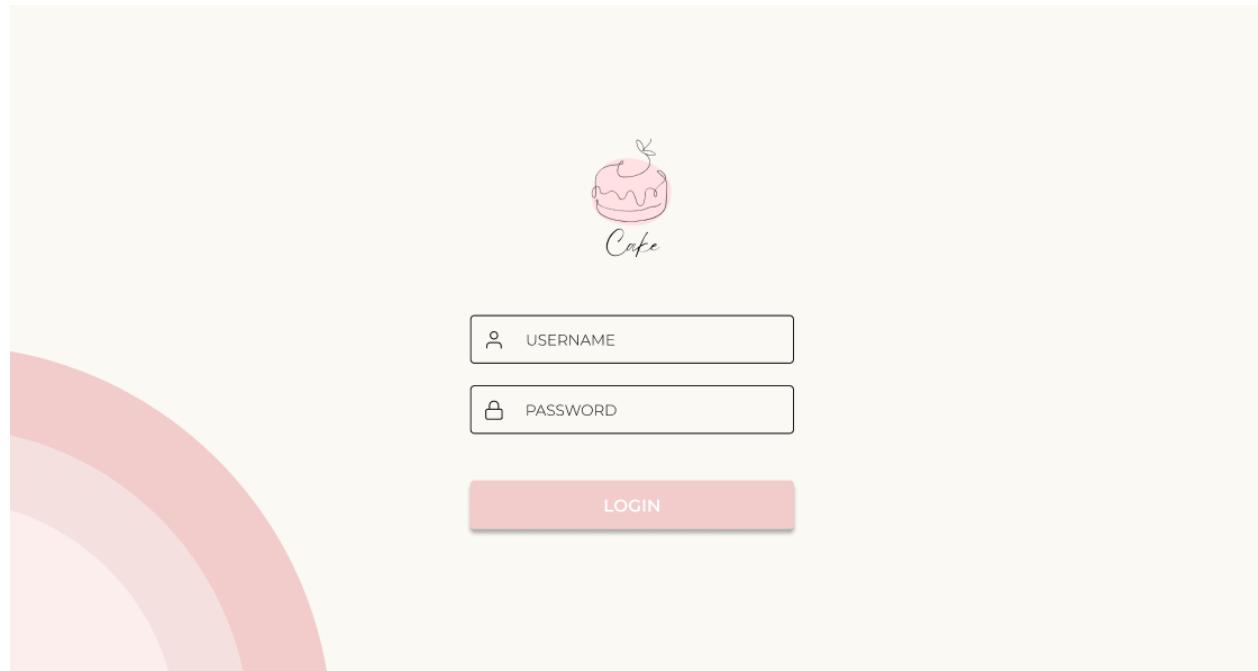
Lược đồ 4. 10 tuần tự thanh toán

4.2. Thiết kế phần mềm để xử lý use-case

4.2.1. Form

4.2.1.1. Form đăng nhập

- Tên: Form đăng nhập
- Ảnh:



Hình ảnh 4. 1 form đăng nhập

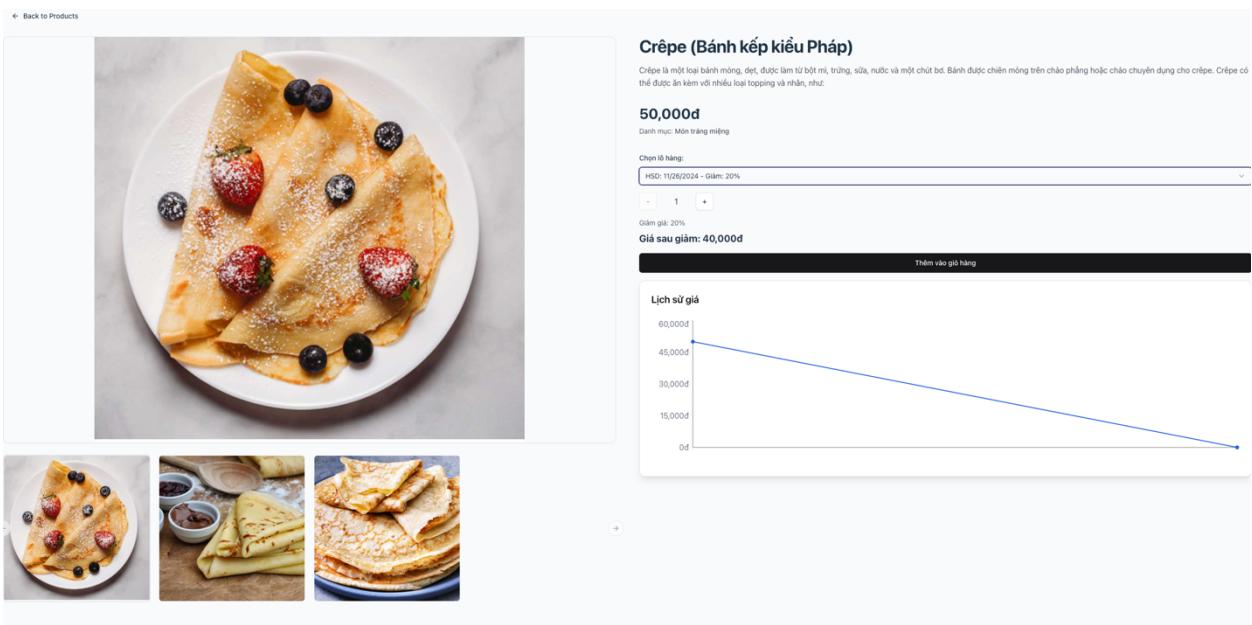
- Users: Quản lý, nhân viên.
- Control chính trên form:
 - o Control: Text-Field tên đăng nhập
 - Inputs: nhập tên đăng nhập từ bàn phím
 - Outputs: dữ liệu tên đăng nhập được lấy từ inputs của control
 - o Control: Text-Field mật khẩu
 - Inputs: mật khẩu nhập từ bàn phím
 - Outputs: dữ liệu mật khẩu được lấy từ inputs của control
 - o Control: Button đăng nhập
 - Inputs: sự kiện click chuột
 - Outputs:
 - Thông báo thành công, thông tin user và chuyển tới form đúng với quyền.
 - Thông báo thất bại
 - o Xử lý:

- Gọi API: Sau khi người dùng **dữ liệu đăng nhập** và nhấn button Đăng nhập, form gọi [API đăng nhập] để xác thực thông tin đăng nhập users.
- Kiểm tra dữ liệu: Form kiểm tra tính hợp lệ của mật khẩu trước khi thực hiện gọi API. Nếu dữ liệu không hợp lệ, thông báo lỗi sẽ được hiển thị trên form.

Tên	Chi tiết	Nguồn
Dữ liệu đăng nhập	Email (email đăng nhập), password(mật khẩu)	Dữ liệu nhập từ tác nhân Quản lý hoặc nhân viên thông qua form đăng nhập

4.2.1.2. Form chi tiết sản phẩm

- **Tên: Form chi tiết sản phẩm**
- **Ảnh:**



Hình ảnh 4. 2 form chi tiết sản phẩm

- **Users: Khách hàng**
- Control chính trên form:
 - Control: Text tên sản phẩm
 - Inputs: Tên sản phẩm lấy từ database thông qua api
 - Outputs: hiển thị dữ liệu inputs của control
 - Control: Text mô tả sản phẩm
 - Inputs: Mô tả sản phẩm lấy từ database thông qua api
 - Outputs: hiển thị dữ liệu inputs của control
 - Control: Text Giá sản phẩm
 - Inputs: Giá sản phẩm lấy từ database thông qua api

- Outputs: hiển thị dữ liệu inputs của control
- Control: Text danh mục
 - Inputs: Danh mục lấy từ database thông qua api
 - Outputs: hiển thị dữ liệu inputs của control
- Control: Combobox chọn lô bánh
 - Inputs: Chọn từ danh sách các lô bánh với thông tin ngày hết hạn và giảm giá lấy từ database thông qua api
 - Outputs:
 - hiển thị dữ liệu inputs của control
 - Hiển thị phần trăm giảm giá nếu có giảm giá trong lô bánh
 - Hiển thị tổng tiền của lô bánh sau khi giảm qua công thức “(Số lượng * Giá * (1 - Phần trăm giảm)) /100”
- Control: Text số lượng bánh
 - Inputs: Số lượng tăng giảm thông qua hai button “+” và “-”
 - Outputs: hiển thị dữ liệu inputs của control
- Control: Button tăng số lượng bánh
 - Inputs: Nhấn vào nút
 - Outputs:
 - Tăng số lượng của text số lượng bánh hiển thị
 - Tính toán lại tổng tiền của lô bánh sau khi giảm qua công thức “(Số lượng * Giá * (1 - Phần trăm giảm)) /100”
- Control: Button giảm số lượng bánh
 - Inputs: Nhấn vào nút
 - Outputs:
 - Giảm số lượng của text số lượng bánh hiển thị
 - Tính toán lại tổng tiền của lô bánh sau khi giảm qua công thức “(Số lượng * Giá * (1 - Phần trăm giảm)) /100”
- Control: LineChart giá của sản phẩm theo thời gian
 - Inputs: Dữ liệu giá theo thời gian được lấy từ database từ api
 - Outputs: Hiển thị line chart biến động giá với x là thời gian, y là giá
- Control: Text phần trăm giảm giá
 - Inputs: Dữ liệu được lấy từ lựa chọn combobox chọn lô bánh
 - Outputs: hiển thị dữ liệu inputs của control
- Control: Text tổng giá
 - Inputs: Dữ liệu được lấy từ lựa chọn combobox chọn lô bánh và số lượng bánh mua và được tính toán lại theo công thức “(Số lượng * Giá * (1 - Phần trăm giảm)) /100”

- Outputs: hiển thị dữ liệu inputs của control
- Control: Button thêm vào giỏ hàng
 - Inputs: Sự kiện nhấp nút
 - Outputs:
 - Chỉ khi lựa chọn lô hàng mới được nhấn nút
 - Hiển thị thông báo thêm thành công
 - Hiển thị thông báo thêm thất bại
- Control: Button quay về
 - Inputs: Sự kiện nhấp nút
 - Outputs: Quay về màn hình home trước đó
- Control: Image gallery ảnh sản phẩm
 - Inputs: Dữ liệu url ảnh lấy từ database từ api
 - Outputs:
 - Hiển thị ảnh chính
 - Hiển thị các ảnh phụ nhỏ hơn phía dưới
 - Nhấn ảnh phụ chuyển ảnh phụ lên làm ảnh chính và ảnh chính xuống làm ảnh phụ
- Xử lý:
 - Đầu tiên khi vào trang form thực hiện gọi [API thông tin chi tiết sản phẩm] và [API lịch sử giá] api trả về **dữ liệu sản phẩm** và **dữ liệu giá**
 - Thành công hiển thị nội dung sản phẩm
 - Số lượng sản phẩm chọn không được vượt quá sản phẩm còn lại trong lô hàng
 - Chọn lô hàng mới được nhấn nút thêm vào giỏ hàng
 - Chọn lô hàng hiển thị giá tính tổng phần trăm giảm giá
 - Nhấn nút thêm sản phẩm
 - Lưu thông tin id lô hàng và số lượng vào local stored
 - Lưu thành công hiển thị thông báo
 - Lưu thất bại hiển thị thông báo lưu thất bại
 - Thất bại thông báo lỗi ra.

Tùy chỉnh dữ liệu [Form đăng nhập]

Tên	Chi tiết	Nguồn
Dữ liệu sản phẩm	Id (id sản phẩm),	Dữ liệu được lấy từ các bảng: products, categories,

	name(tên sản phẩm), category { id (id danh mục), name (tên danh mục)}, productBatches[{id (id lô bánh), expiration_date(ngày hết hạn), discount(giảm giá của lô bánh)] imageUrls[imageUrl(url ảnh),] Effect_date(ngày áp dụng)}]	images, product_batches được lấy và đi qua API tương ứng
Dữ liệu giá	priceHistory[{id (id lịch sử),	Dữ liệu được lấy từ bảng product_history với id của sản phẩm được lấy và trả dữ liệu thông qua API lịch sử giá
Dữ liệu thêm vào giỏ hàng	Product_batch_id (id lô bánh), quantity (số lượng)	Dữ liệu nhập từ tác nhân khách hàng từ form chi tiết sản phẩm với id lô bánh được lấy từ lô bánh người dùng chọn của chi tiết sản phẩm người dùng đang xem và số lượng là số lượng đang chọn

4.2.1.3. Form quản lý Bill

- **Tên:** Form quản lý bill
- **Ảnh:**

Danh sách hóa đơn

Tìm kiếm hóa đơn

 Tim kiếm

Chọn trạng thái thanh toán

Mã hóa đơn	Tên khách hàng	Số điện thoại	Phương thức thanh toán	Hình thức dùng	Trạng thái	Tổng tiền	Thao tác
8	John Doe	123456789	CASH	DINE_IN	PAID	19 ₫	Xem chi tiết
10	John Doe	123456789	QR_CODE	TAKE_AWAY	PAID	19 ₫	Xem chi tiết
16	Quy Luong	0385258727	CASH	DINE_IN	PAID	19 ₫	Xem chi tiết
22	Quy Luong	0385258727	CASH	DINE_IN	PAID	19 ₫	Xem chi tiết
25	Quy Luong	0385258727	QR_CODE	DINE_IN	PAID	18 ₫	Xem chi tiết
8	John Doe	123456789	CASH	DINE_IN	PAID	19 ₫	Xem chi tiết
10	John Doe	123456789	QR_CODE	TAKE_AWAY	PAID	19 ₫	Xem chi tiết
16	Quy Luong	0385258727	CASH	DINE_IN	PAID	19 ₫	Xem chi tiết
22	Quy Luong	0385258727	CASH	DINE_IN	PAID	19 ₫	Xem chi tiết
25	Quy Luong	0385258727	QR_CODE	DINE_IN	PAID	18 ₫	Xem chi tiết

Hiển thị 5 trên tổng số 0 danh mục

Trước

Trang 1 / 1

Sau

10

Hình ảnh 4. 3 Form quản lý Bill

- Users: Quản lý- nhân viên
- Control chỉnh trên form:
 - o Control: Button tìm kiếm
 - Inputs: Sự kiện nhấn vào
 - Outputs: Hiển thị thông tin bill của đối tượng được tìm kiếm
 - o Control: Button Xem chi tiết
 - Inputs: Sự kiện nhấn vào
 - Outputs: Hiển thị thông tin về đơn hàng
 - o Control: Text-field nhập các thông tin để tìm kiếm

- Inputs:
 - Dữ liệu được nhập từ bàn phím người dùng
 - Placeholder text: "Mã hóa đơn", "Tên khách hàng", "Số điện thoại"
- Outputs:
 - Chuỗi dữ liệu sau khi nhập
- Control: Dropdown trạng thái bill
 - Inputs:
 - Chưa thanh toán
 - Đã thanh toán
 - Hoàn thành
 - Hủy
 - Outputs:
 - Hiện thị các hóa đơn theo trạng thái
- Xử lý:
 - Đầu tiên gọi **api lấy danh sách bill** theo trạng thái đã thanh toán vì xem đã thanh để nhân viên chuẩn bị bánh cho khách hàng
 - Nhập các thông tin để tìm kiếm nếu cần xác nhận thanh toán cho khách thanh toán tiền mặt. Dùng api tìm kiếm hóa đơn
 - Xem chi tiết đơn có thẻ nhán hoàn thành với đơn đã thanh toán khi phục vụ bánh cho khách hàng hoàn tất. Dùng api cập nhật trạng thái bill
 - Chỉ quản lý được hủy khi đơn chưa thanh toán, nút hủy chỉ hiện khi có quyền quản lý. Dùng **api cập nhật trạng thái bill**

4.2.1.4. Form giỏ hàng đặt hàng

Tên: Form giỏ hàng đặt hàng

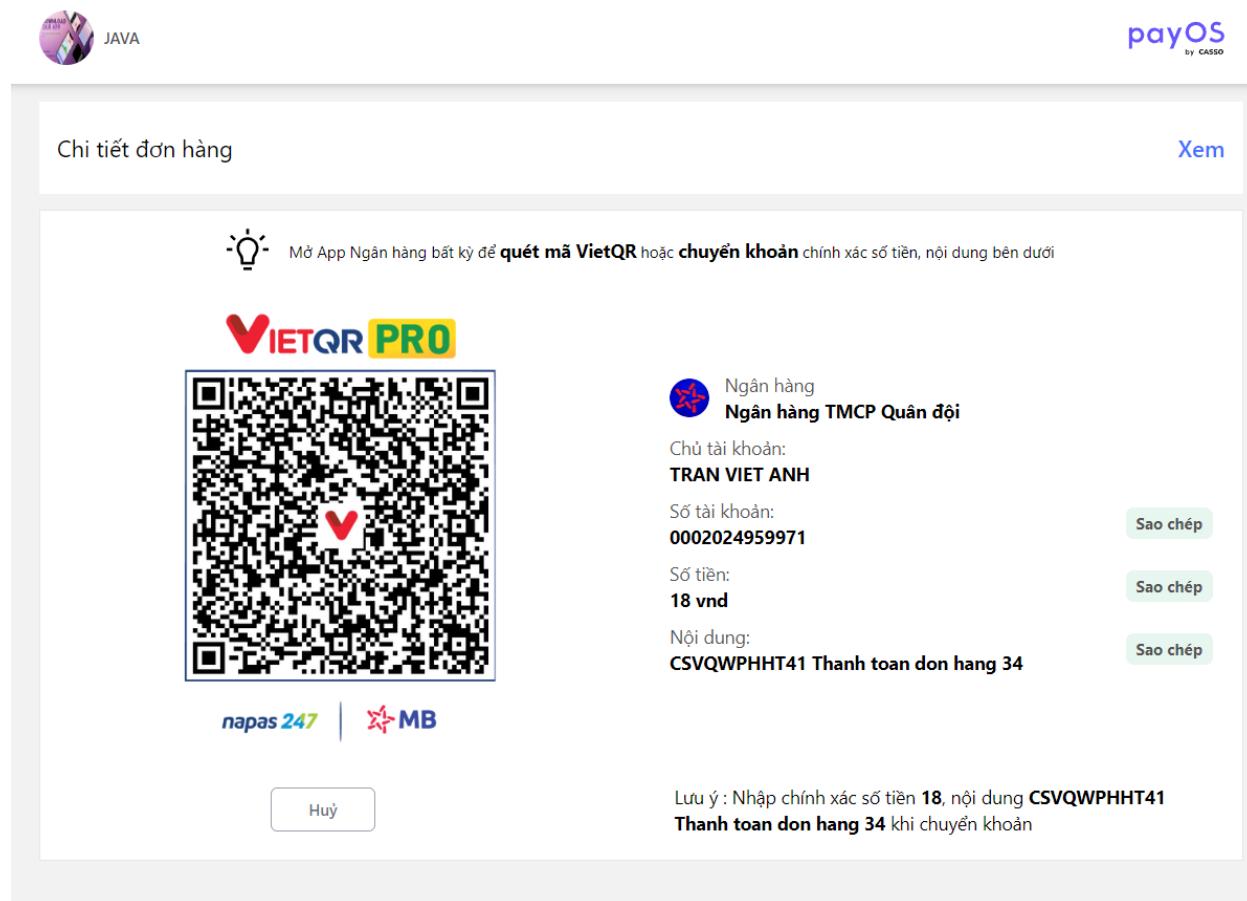
Ảnh:

The screenshot shows a user interface for a shopping cart. On the left, there's a card for a "Lemon Cake" listed as "Food" at \$27.99 with a 20% discount. To the right is an "Order Summary" form with fields for Customer Name (Quy Luong), Phone Number (0385258727), Payment Method (Cash), Dining Option (Dine In), and a Discount Code input field. The total amount is listed as \$111.96, and a "Proceed to Checkout" button is at the bottom.

Hình ảnh 4. 4 Form giỏ hàng đặt hàng

- Users: Khách hàng
- Control chính trên form:
 - o Control: Button trở về trang trước
 - Inputs: Sự kiện nhấn vào
 - Outputs: trở về màn hình trước đó
 - o Control: Button tăng giảm số lượng
 - Inputs: Sự kiện nhấn vào
 - Outputs: Hiển thị số lượng sau khi tăng giảm
 - o Control: Button xóa sản phẩm khỏi giỏ hàng
 - Inputs: Sự kiện nhấn vào
 - Outputs: Cập nhật hiển thị sản phẩm trong giỏ hàng
 - o Control: Text-field tìm kiếm
 - Inputs:
 - Dữ liệu được nhập từ bàn phím người dùng
 - Placeholder text: "Customer Name", "Phone Number"
 - Outputs:
 - Chuỗi dữ liệu sau khi nhập
 - o Control: Dropdown Payment Method
 - Inputs:
 - Cash
 - Card
 - Outputs:
 - Giá trị lựa chọn được chọn

- Control: Dropdown Dining Option
 - Inputs:
 - Dine In
 - Take Away
 - Outputs:
 - Giá trị lựa chọn được chọn
 - Control: Button Checkout
 - Inputs: Sự kiện nhấn vào
 - Outputs: Tùy vào giá trị của Payment Method mà chuyển đến trang thích hợp
 - Xử lý:
 - Đầu tiên lấy dữ liệu bánh mà khách chọn trong local stored
 - Thêm bớt số lượng sản phẩm
 - Xóa sản phẩm
 - Nhập các thông tin cơ bản của khách hàng
 - Checkout gọi api tạo hóa đơn
- 4.2.1.5. Form thanh toán mã qr**
- Tên: Form thanh toán mã qr
 - Ảnh:



Hình ảnh 4. 5 Form thanh toán mã qr

- Users: Khách hàng
- Control chỉnh trên form:
 - o Control: Button xem chi tiết đơn hàng
 - Inputs: Sự kiện nhấn vào
 - Outputs: Hiển thị mã đơn hàng kèm số tiền thanh toán
 - o Control: Button Hủy
 - Inputs: Sự kiện nhấn vào
 - Outputs: Trở về trang hủy thành công
- Xử lý:
 - o Đầu tiên sau khi gọi API checkout ở bước trước đó sẽ chuyển sang trang này gọi api tạo link thanh toán
 - o Nhấn nút xem để xem mã đơn hàng cùng với số tiền thanh toán
 - o Khi thanh toán thành công sẽ chuyển sang trang thanh toán thành công
 - o Nhấn hủy chuyển sang trang hủy thành công
- 4.2.1.6. Form quản lý kho**
- Tên: Form quản lý kho

- Ảnh:

Quản lý nguyên liệu

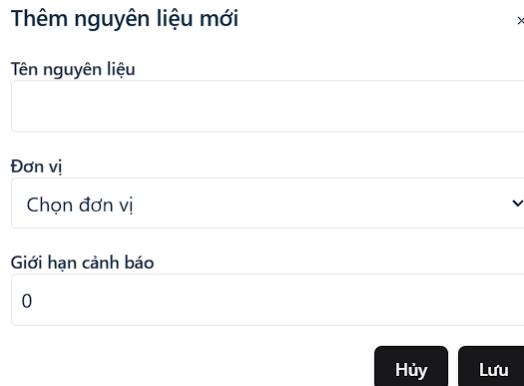
Tên nguyên liệu	Số lượng	Đơn vị	Trạng thái	Hành động
Bột gạo 1	100.3	kilogram	Đủ nguyên liệu	<input checked="" type="checkbox"/> <input type="checkbox"/>
bột mì	16.6	kilogram	Đủ nguyên liệu	<input checked="" type="checkbox"/> <input type="checkbox"/>
nguyên liệu 1	30	quả	Đủ nguyên liệu	<input checked="" type="checkbox"/> <input type="checkbox"/>
đường	8	đơn vị 1	Cần nhập thêm	<input checked="" type="checkbox"/> <input type="checkbox"/>

Hiển thị 4 trên tổng số 4 danh mục

Trước Trang 1 / 1 Sau

Hình ảnh 4. 6 Form quản lý kho

- Users: Quản lý
- Control:
 - + Button Nhập nguyên liệu
 - o Input: Sự kiện nhấn vào
 - o Output: Chuyển sang trang nhập nguyên liệu
 - + Button Xuất nguyên liệu
 - o Input: Sự kiện nhấn vào
 - o Output: Chuyển sang trang xuất nguyên liệu
 - + Button Lịch sử nhập nguyên liệu
 - o Input: Sự kiện nhấn vào
 - o Output: Chuyển sang trang lịch sử nhập nguyên liệu
 - + Button Lịch sử xuất nguyên liệu
 - o Input: Sự kiện nhấn vào
 - o Output: Chuyển sang trang lịch sử xuất nguyên liệu
 - + Button Thêm nguyên liệu
 - o Input: Sự kiện nhấn vào
 - o Output:
 - Hiển thị form để điền thông tin nguyên liệu mới



Thêm nguyên liệu mới

Tên nguyên liệu

Đơn vị

Chọn đơn vị

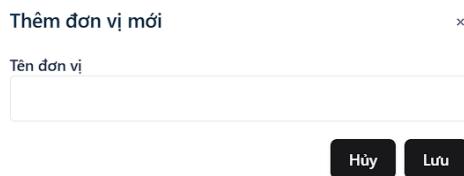
Giới hạn cảnh báo

0

Hủy Lưu

Hình ảnh 4. 7 form điền thông tin nguyên liệu

- Sau khi xác nhận thêm nguyên liệu mới, hiển thị thông báo và cập nhật lại danh sách nguyên liệu
- + Button Thêm đơn vị
 - Input: Sự kiện nhấn vào
 - Output:
 - Hiển thị form điền thông tin đơn vị mới



Thêm đơn vị mới

Tên đơn vị

Hủy Lưu

Hình ảnh 4. 8 form thêm đơn vị

- Sau khi xác nhận thêm đơn vị mới, hiển thị thông báo và cập nhật lại danh sách đơn vị
- + Button Sửa nguyên liệu
 - Input: Sự kiện nhấn vào
 - Output:
 - Hiển thị form sửa thông tin nguyên liệu

Sửa nguyên liệu

Tên nguyên liệu
bột mì

Đơn vị
kilogram

Warning Limits
5

Hủy Lưu

Hình ảnh 4. 9 form sửa nguyên liệu

- Sau khi xác nhận sửa thông tin nguyên liệu, hiển thị thông báo và cập nhật lại thông tin của nguyên liệu
- + Button Xóa nguyên liệu
 - Input: Sự kiện nhấn vào
 - Output:

Xác nhận xóa

Bạn có chắc chắn muốn xóa nguyên liệu này không?

Hủy Xóa

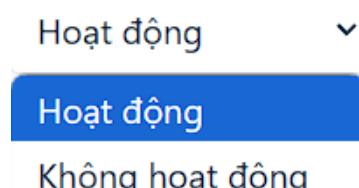
Hình ảnh 4. 10 form xác nhận xoá

- Hỏi lại để chắc chắn người dùng muốn xóa nguyên liệu
- Nếu người dùng xác nhận xóa, hiển thị thông báo và cập nhật danh sách nguyên liệu
- + Text Tên nguyên liệu:
 - Input: Lấy tên của nguyên liệu trong danh sách nguyên liệu từ api
 - Output: Hiển thị tên của nguyên liệu
- + Text Số lượng:
 - Input: Lấy số lượng của nguyên liệu trong danh sách nguyên liệu từ api
 - Output: Hiển thị số lượng tồn của nguyên liệu
- + Text Đơn vị:
 - Input:
 - Lấy id của đơn vị của nguyên liệu trong danh sách nguyên liệu từ api
 - Từ id đơn vị, lấy tên của đơn vị tương ứng theo danh sách đơn vị lấy được từ api
 - Output: Hiển thị tên của đơn vị tương ứng với nguyên liệu

+ Text Trạng thái:

- Input: Lấy số lượng và ngưỡng cảnh báo của nguyên liệu trong danh sách nguyên liệu từ api
- Output: Hiển thị trạng thái của nguyên liệu dựa trên số lượng tồn và ngưỡng cảnh báo của nguyên liệu đó
 - Số lượng > Ngưỡng cảnh báo: Đủ nguyên liệu
 - Số lượng < Ngưỡng cảnh báo: Cần nhập thêm

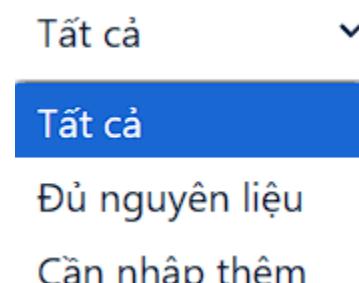
+ Combobox lọc nguyên liệu Hoạt động/Không hoạt động:



Hình ảnh 4. 11 combobox trạng thái nguyên liệu

- Input: Chọn mục nguyên liệu cần xem: Hoạt động/Không hoạt động
- Output: Thay đổi danh sách nguyên liệu đang hiển thị theo điều kiện đã chọn

+ Combobox lọc nguyên liệu theo trạng thái:



Hình ảnh 4. 12 combobox lọc nguyên liệu

- Input: Chọn mục nguyên liệu cần xem: Tất cả/Đủ nguyên liệu/Cần nhập thêm
- Output: Thay đổi danh sách nguyên liệu đang hiển thị theo điều kiện đã chọn

- Xử lý:

- + Lấy danh sách nguyên liệu và danh sách đơn vị từ api
- + Hiển thị danh sách nguyên liệu ban đầu sẽ dựa trên danh sách lấy được và lọc theo điều kiện: “Hoạt động” và “Tất cả”
- + Thông báo lỗi khi không lấy được dữ liệu
- + Khi Thêm nguyên liệu:
 - Kiểm tra dữ liệu và báo lỗi trước khi gọi api thêm nguyên liệu

- Gọi api thêm nguyên liệu nếu dữ liệu hợp lệ và thông báo kết quả
- + Khi Thêm đơn vị:
 - Kiểm tra dữ liệu và báo lỗi trước khi gọi api thêm đơn vị
 - Gọi api thêm đơn vị nếu dữ liệu hợp lệ và thông báo kết quả
- + Khi Sửa nguyên liệu:
 - Kiểm tra dữ liệu và báo lỗi trước khi gọi api sửa nguyên liệu
 - Gọi api sửa nguyên liệu nếu dữ liệu hợp lệ và thông báo kết quả
- + Khi Xóa nguyên liệu:
 - Kiểm tra dữ liệu và báo lỗi trước khi gọi api xóa nguyên liệu
 - Gọi api xóa nguyên liệu nếu dữ liệu hợp lệ và thông báo kết quả

4.2.1.7. Form nhập nguyên liệu

- Tên: Form nhập nguyên liệu
- Ảnh:

Nhập Nguyên Liệu

Tên Nguyên Liệu	Số Lượng	Giá	Đơn Vị	Thao Tác
đường	1	10000	đơn vị 1	Xóa

Tổng Tiền: 10.000đ

Xác Nhận

Hình ảnh 4. 13 form nhập nguyên liệu

- Control:
 - + Button Quay lại
 - Input: Sự kiện nhấn vào
 - Output: Quay lại trang trước đó
 - + Button Chọn nguyên liệu
 - Input: Sự kiện nhấn vào
 - Output:
 - Hiển thị form chọn nguyên liệu: Có các thông tin của danh sách nguyên liệu chưa thêm vào danh sách nhập

Chọn Nguyên Liệu

Tên Nguyên Liệu	Số Lượng Tồn	Đơn Vị	Chọn
Bột gạo 1	100.3	kilogram	<input type="checkbox"/>
bột mì	16.6	kilogram	<input type="checkbox"/>
nguyên liệu 1	30	quả	<input type="checkbox"/>

Hủy Thêm

Hình ảnh 4. 14 form chọn nguyên liệu để thêm

- Chọn các nguyên liệu cần thêm vào danh sách nhập bằng ô Chọn bên phải
- Khi nhấn nút Thêm, tất cả nguyên liệu được chọn sẽ được thêm vào danh sách nhập với số lượng mặc định là 1 và giá mặc định là 0, cập nhật lại danh sách nhập

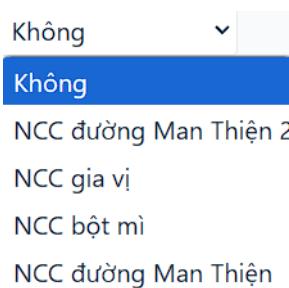
+ Button Xóa:

- Input: Sự kiện nhấn vào
- Output: Xóa nguyên liệu ra khỏi danh sách nhập và cập nhật lại danh sách nhập

+ Button Xác nhận:

- Input: Sự kiện nhấn vào
- Output:
 - Lấy danh sách nguyên liệu nhập và kiểm tra trước khi gọi api nếu tất cả dữ liệu hợp lệ
 - Hiển thị thông báo kết quả

+ Combobox Nhà cung cấp:



Hình ảnh 4. 15 combobox nhà cung cấp

- Input: Chọn một nhà cung cấp dựa theo tên hoặc chọn “Không”
 - Output: Hiển thị tên nhà cung cấp đã chọn
- + Text tên nguyên liệu

- Input: Lấy tên nguyên liệu từ danh sách nhập
- Output: Hiển thị tên nguyên liệu
- + EditText Số lượng
 - Input: Điện số lượng của nguyên liệu tương ứng cần nhập
 - Output:
 - Kiểm tra tính hợp lệ của dữ liệu điền vào
 - Thay đổi hiển thị số lượng nhập và tính toán lại Tổng giá khi dữ liệu hợp lệ
- + EditText Giá
 - Input: Điện giá(số tiền mỗi đơn vị) của nguyên liệu tương ứng cần nhập
 - Output:
 - Kiểm tra tính hợp lệ của dữ liệu điền vào
 - Thay đổi hiển thị giá của nguyên liệu và tính toán lại Tổng giá khi dữ liệu hợp lệ
- + Text Tổng tiền:
 - Input: Tính toán tổng số tiền của lần nhập nguyên liệu này dựa trên số lượng và đơn giá của mỗi nguyên liệu trong danh sách nhập
 - Công thức: Tổng của tất cả số lượng nhập * giá nhập của mỗi nguyên liệu trong danh sách nhập
 - Output: Hiển thị tổng số tiền đã tính được
- Xử lý:
 - + Lấy danh sách nguyên liệu, danh sách đơn vị từ api lấy danh sách nguyên liệu và api lấy danh sách đơn vị
 - + Lấy danh sách nhà cung cấp bằng api lấy danh sách nhà cung cấp và đưa vào combobox nhà cung cấp, thêm vào đó 1 lựa chọn “Không”
 - + Lọc danh sách nguyên liệu theo điều kiện “Hoạt động”
 - + Lọc danh sách nguyên liệu theo điều kiện “Cần nhập thêm” và thêm chúng vào danh sách nhập để hiển thị ban đầu
 - + Thông báo lỗi nếu lấy dữ liệu không thành công
 - + Khi nhấn nút “Xác nhận”, lấy dữ liệu trong danh sách nhập để thực hiện nhập nguyên liệu:
 - Kiểm tra dữ liệu trước khi gọi api và báo lỗi nếu dữ liệu không hợp lệ
 - Nếu dữ liệu hợp lệ, gọi api nhập nguyên liệu và hiển thị thông báo kết quả
- Tên: Form xuất nguyên liệu
- Ảnh:

Xuất Nguyên Liệu

[Quay lại](#)

[Chọn sản phẩm](#)

Tên Sản Phẩm	Số Lượng	Thao Tác
bánh 1	1	Xóa

Tổng Số Bánh: 1

[Xác Nhận](#)

Hình ảnh 4. 16 form xuất nguyên liệu

- Control:

+ Button Quay lại:

- o Input: Sự kiện nhấn vào
- o Output: Quay lại trang trước đó

+ Button Chọn sản phẩm:

- o Input: Sự kiện nhấn vào
- o Output:

- Mở form chọn các sản phẩm cần làm để thêm vào danh sách

Chọn Sản Phẩm

Tên Sản Phẩm	Chọn
bánh 2	<input type="checkbox"/>

[Hủy](#) [Thêm](#)

Hình ảnh 4. 17 form chọn các sản phẩm để thêm

- Form này chỉ hiển thị các sản phẩm chưa được chọn vào danh sách
- Chọn sản phẩm cần làm bằng ô chọn và nhấn nút Thêm để thêm sản phẩm đã chọn vào danh sách hiển thị và cập nhật lại danh sách
- Các sản phẩm mới được chọn sẽ có số lượng mặc định là 1

+ Button Xóa:

- o Input: Sự kiện nhấn vào
- o Output: Xóa sản phẩm ra khỏi danh sách và cập nhật lại danh sách sản phẩm đã chọn

+ Button Xác nhận:

- o Input: Sự kiện nhấn vào
- o Output:
 - Lấy danh sách sản phẩm đã chọn và kiểm tra dữ liệu, thông báo lỗi nếu dữ liệu không hợp lệ

- Nếu dữ liệu hợp lệ, dựa vào các sản phẩm đã chọn để lấy các công thức bằng api lấy công thức theo id sản phẩm
- Tính số lượng các nguyên liệu cần dùng dựa trên các công thức đã lấy được và số lượng sản phẩm tương ứng cần làm và hiển thị

Nguyên Liệu Cần Dùng

Tên Nguyên Liệu	Số Lượng Cần Dùng	Đơn Vị	Trạng Thái
Bột gạo 1	2.5	kilogram	Đủ
bột mì	1.9	kilogram	Đủ

Hủy
Xuất Nguyên Liệu

Hình ảnh 4. 18 form thông tin nguyên liệu

- Sau khi tính toán số nguyên liệu cần dùng, kiểm tra với số nguyên liệu tồn có còn đủ hay không, nếu đủ thì để trạng thái là “Đủ” và nếu không đủ thì trạng thái sẽ là “Thiếu”
- Nếu có 1 nguyên liệu nào bị thiếu, ấn nút “Xuất nguyên liệu” để không cho người dùng gọi api xuất nguyên liệu
- Nếu tất cả nguyên liệu vẫn đủ và chọn “Xuất nguyên liệu”, gọi api xuất nguyên liệu và hiển thị kết quả

+ Text Tên sản phẩm:

- Input: Lấy tên sản phẩm từ danh sách sản phẩm đã chọn để hiển thị
- Output: Hiển thị tên của sản phẩm đã chọn

+ EditText Số lượng:

- Input: Nhập vào số lượng cần làm của sản phẩm tương ứng
- Output:
 - Kiểm tra tính hợp lệ của dữ liệu nhập vào và báo lỗi nếu dữ liệu không hợp lệ
 - Nếu dữ liệu nhập vào hợp lệ, tính toán lại tổng số bánh, hiển thị số lượng và tổng số bánh mới

+ Text Tổng số bánh:

- Input: Tính toán tổng số bánh cần làm bằng cột tổng số trong danh sách sản phẩm đã chọn
- Output: Hiển thị tổng số bánh cần làm

- Xử lý:

- + Lấy danh sách sản phẩm từ api danh sách sản phẩm, lọc ra các sản phẩm hoạt động
- + Lấy danh sách nguyên liệu từ api danh sách nguyên liệu
- + Lấy danh sách đơn vị từ api danh sách đơn vị
- + Khi nhấn nút “Xác nhận”, lấy các công thức dựa trên id của sản phẩm đã chọn từ api lấy công thức theo id sản phẩm
- + Sau khi nhấn nút “Xuất nguyên liệu”, gọi api xuất nguyên liệu và hiển thị thông báo kết quả

4.2.1.9. Form lịch sử nhập nguyên liệu

- Tên: form lịch sử nhập nguyên liệu
- Ảnh:

Người nhập	Nhà cung cấp	Ngày nhập	Tổng số tiền	Hành động
Nguyễn Văn A		19:14:40 27/11/2024	0 ₫	Xem chi tiết
Nguyễn Văn A	NCC gia vị	18:35:16 27/11/2024	1.111 ₫	Xem chi tiết
Nguyễn Văn A		16:54:21 27/11/2024	11.111 ₫	Xem chi tiết
Nguyễn Văn A		16:54:01 27/11/2024	4.000 ₫	Xem chi tiết
Nguyễn Văn A		16:52:47 27/11/2024	1.000 ₫	Xem chi tiết
Nguyễn Văn A		16:52:03 27/11/2024	2.000 ₫	Xem chi tiết
Nguyễn Quang Nghĩa	NCC gia vị	17:58:09 26/11/2024	1.000 ₫	Xem chi tiết
Nguyễn Quang Nghĩa		17:57:32 26/11/2024	111.000 ₫	Xem chi tiết
Nguyễn Quang Nghĩa	NCC đường Man Thiện	17:49:39 24/11/2024	275.000 ₫	Xem chi tiết
Nguyễn Quang Nghĩa	NCC gia vị	23:47:27 20/11/2024	692.000 ₫	Xem chi tiết

Hình ảnh 4. 19 form lịch sử nhập nguyên liệu

- Control:
 - + Button Quay lại:
 - o Input: Sự kiện nhấn vào
 - o Output: Quay lại trang trước đó
 - + Text Người nhập:
 - o Input: Lấy tên người nhập tương ứng bằng id người nhập trong lịch sử nhập nguyên liệu
 - o Output: Hiển thị tên của người đã nhập nguyên liệu
 - + Text Nhà cung cấp:
 - o Input: Lấy tên nhà cung cấp tương ứng bằng id nhà cung cấp trong lịch sử nhập nguyên liệu

- Output: Hiển thị tên của nhà cung cấp tương ứng
- + Text Ngày nhập:
 - Input: Lấy thời gian nhập trong lịch sử nhập nguyên liệu
 - Output: Hiển thị thời gian nhập nguyên liệu tương ứng
- + Text Tổng số tiền:
 - Input: Lấy tổng tiền trong lịch sử nhập nguyên liệu
 - Output: Hiển thị tổng tiền tương ứng
- + Button Xem chi tiết:
 - Input: Sự kiện nhấn vào
 - Output:
 - Hiển thị chi tiết của lần nhập nguyên liệu đã chọn

Chi tiết nhập nguyên liệu ×

Danh sách nguyên liệu:

Nguyên liệu	Số lượng	Đơn vị	Giá
Bột gạo 1	30	kilogram	15.000 đ
bột mì	10	kilogram	20.000 đ
bột bắp	2	quả	21.000 đ

Đóng

Hình ảnh 4. 20 form chi tiết nhập nguyên liệu

- Xử lý:
 - + Lấy danh sách lịch sử nhập nguyên liệu và chi tiết từ api lấy danh sách nhập nguyên liệu
 - + Lấy danh sách nguyên liệu và đơn vị từ api danh sách nguyên liệu và api danh sách đơn vị
 - + Lấy danh sách người dùng từ api danh sách người dùng
 - + Thông báo lỗi nếu không lấy được dữ liệu
 - + Hiển thị tên người nhập bằng id người nhập và tìm trong danh sách người dùng
 - + Khi chọn “Xem chi tiết”
 - Hiển thị tên nguyên liệu bằng id nguyên liệu trong chi tiết lịch sử nhập và tìm trong danh sách nguyên liệu
 - Hiển thị tên đơn vị bằng cách từ nguyên liệu tương ứng, lấy id nguyên liệu và tìm trong danh sách nguyên liệu

4.2.1.10. Form lịch sử xuất nguyên liệu

- Tên: Trang lịch sử xuất nguyên liệu
- Ảnh:

Người xuất kho	Ngày xuất	Tổng số lượng	Hành động
Nguyễn Văn A	17:16:29 Th 4, 04/12/2024	5	Xem chi tiết
Nguyễn Văn A	16:52:25 Th 4, 27/11/2024	10	Xem chi tiết
John Doe	14:32:56 Th 4, 27/11/2024	23	Xem chi tiết
Nguyễn Quang Nghĩa	13:52:36 Th 4, 27/11/2024	20	Xem chi tiết
Nguyễn Quang Nghĩa	23:42:03 Th 2, 25/11/2024	1	Xem chi tiết
Nguyễn Quang Nghĩa	18:30:11 CN, 24/11/2024	1	Xem chi tiết
Nguyễn Quang Nghĩa	02:02:13 Th 5, 21/11/2024	5	Xem chi tiết
Nguyễn Quang Nghĩa	02:00:47 Th 5, 21/11/2024	5	Xem chi tiết
Nguyễn Quang Nghĩa	23:54:20 Th 4, 20/11/2024	5	Xem chi tiết
Nguyễn Quang Nghĩa	23:49:17 Th 4, 20/11/2024	5	Xem chi tiết

Hình ảnh 4. 21 form lịch sử xuất nguyên liệu

- Control:
 - + Button Quay lại:
 - o Input: Sự kiện nhấn vào
 - o Output: Quay lại trang trước đó
 - + Text Người xuất kho:
 - o Input: Lấy tên người xuất nguyên liệu tương ứng bằng id người xuất trong lịch sử xuất nguyên liệu
 - o Output: Hiển thị tên của người đã xuất nguyên liệu
 - + Text Ngày xuất:
 - o Input: Lấy thời gian xuất nguyên liệu trong lịch sử xuất nguyên liệu
 - o Output: Hiển thị thời gian xuất nguyên liệu tương ứng
 - + Text Tổng số lượng:
 - o Input: Lấy tổng số lượng bánh trong lịch sử nhập nguyên liệu
 - o Output: Hiển thị tổng số lượng bánh đã làm tương ứng
 - + Button Xem chi tiết:
 - o Input: Sự kiện nhấn vào
 - o Output:
 - Hiển thị chi tiết của lần xuất nguyên liệu đã chọn

Chi tiết xuất nguyên liệu

Danh sách nguyên liệu:

Nguyên liệu	Số lượng	Đơn vị
Bột gạo 1	4	kilogram
bột mì	5	kilogram

Đóng

Hình ảnh 4. 22 form chi tiết xuất nguyên liệu

- Xử lý:
 - + Lấy danh sách lịch sử xuất nguyên liệu và chi tiết từ api lấy danh sách xuất nguyên liệu
 - + Lấy danh sách nguyên liệu và đơn vị từ api danh sách nguyên liệu và api danh sách đơn vị
 - + Lấy danh sách người dùng từ api danh sách người dùng
 - + Thông báo lỗi nếu không lấy được dữ liệu
 - + Hiển thị tên người xuất bằng cách lấy id người xuất và tìm trong danh sách người dùng
 - + Khi chọn “Xem chi tiết”
 - o Hiển thị tên nguyên liệu bằng id nguyên liệu trong chi tiết lịch sử xuất và tìm trong danh sách nguyên liệu
 - o Hiển thị tên đơn vị bằng cách từ nguyên liệu tương ứng, lấy id nguyên liệu và tìm trong danh sách nguyên liệu

4.2.1.11. Form quản lý hạn sử dụng

- **Tên:** Form quản lý hạn sử dụng
- **Ảnh:**

ID	Name	Status	Expiry Date	Daily Discount	Countdown	Quantity
11	Crêpe (Bánh kếp kiểu Pháp)	NEAR_EXPIRY	2024-11-26	20%	0 days	100
13	Crêpe (Bánh kếp kiểu Pháp)	ACTIVE	2024-11-27	0%	1 days	100
14	Crêpe (Bánh kếp kiểu Pháp)	EXPIRED	2024-11-04	0%	-22 days	10

Hình ảnh 4. 23 form quản lý hạn sử dụng

- Users: quản lý
- Control chỉnh trên form:

- Control: Button trở về trang trước
 - Inputs: Sự kiện nhấn vào
 - Outputs: trở về màn hình trước đó
- Control: Text-field tìm kiếm
 - Inputs:
 - Dữ liệu được nhập từ bàn phím người dùng
 - Placeholder text: "Search products"
 - Outputs:
 - Chuỗi dữ liệu tìm kiếm khi nhập
 - Trigger event tìm kiếm khi nhập
- Control: Dropdown lọc
 - Inputs:
 - Danh sách các lựa chọn lọc
 - Giá trị mặc định “ALL”
 - Lựa chọn
 - ALL
 - NEAR_EXPIRY
 - ACTIVE
 - EXPIRED
 - Outputs:
 - Giá trị lựa chọn được chọn
 - Trigger lọc khi có thay đổi
- Control: Table lô bánh
 - Inputs:
 - Dữ liệu danh sách các lô bánh được lấy từ database thông qua API
 - Có các column:
 - Id
 - Name
 - Status
 - Daily discount
 - Countdown
 - Quantity
 - Outputs: Hiển thị danh sách sản phẩm lên trên table
- Control: Button huỷ nhanh
 - Inputs: Sự kiện nhấn nút
 - Outputs: chuyển hướng trang tới trang huỷ sản phẩm

- Control: Button giảm giá nhanh
 - Inputs: Sự kiện nhấn nút
 - Outputs: Chuyển hướng tới trang giảm giá nhanh
- Xử lý:
 - Đầu tiên gọi **[API danh sách các lô bánh theo trạng thái]** API trả về **dữ liệu lô bánh**
 - Trigger gọi api lại và truyền **param tìm kiếm** khi người dùng search hoặc lọc
 - Thông báo lỗi khi không tải được dữ liệu

Tùy biến dữ liệu [Form quản lý hạn sử dụng]

Tên	Chi tiết	Nguồn
Dữ liệu lô bánh	Id (Id bánh), Name(Tên bánh), Status (trạng thái lô bánh), Expiration_date (Ngày hết hạn), Quantity (số lượng còn lại của lô bánh)	Dữ liệu lấy từ database bảng products, product_batches và được trả thông qua api danh sách lô bánh theo trạng thái
Param tìm kiếm	Tên, số trang, sắp xếp, trạng thái	Dữ liệu được lấy từ tác nhân quản lý

4.2.1.12. Form huỷ nhanh

- Tên: Form huỷ nhanh
- Ảnh:

The screenshot shows a table with columns: ID, Name, Status, Expiry Date, Daily Discount, Countdown, and Quantity. The rows list various items like Crêpe and Bánh Cupcake Vanilla, all marked as EXPIRED. A red 'Delete Selected' button is visible at the top right.

ID	Name	Status	Expiry Date	Daily Discount	Countdown	Quantity
13	Crêpe (Bánh kếp kiểu Pháp)	EXPIRED	2024-11-27	0%	-13 days	100
15	Crêpe (Bánh kếp kiểu Pháp)	EXPIRED	2024-11-26	0%	-14 days	100
16	Crêpe (Bánh kếp kiểu Pháp)	EXPIRED	2024-11-29	0%	-11 days	10
17	Crêpe (Bánh kếp kiểu Pháp)	EXPIRED	2024-12-05	20%	-5 days	10
19	Bánh Cupcake Vanilla	EXPIRED	2024-12-07	0%	-3 days	8
20	Bánh Mì Gối	EXPIRED	2024-12-09	0%	-1 days	10

Hình ảnh 4. 24 form huỷ nhanh

- Users: Quản lý
- Control chính trên form:
 - Control: Button quay về
 - Inputs: sự kiện nhấn nút
 - Outputs: quay về màn hình trước đó
 - Control: Table lô bánh

- Inputs:
 - Dữ liệu danh sách các lô bánh hết hạn được lấy từ database thông qua API
 - Có các column:
 - Select
 - Id
 - Name
 - Status
 - Daily discount
 - Countdown
 - Quantity
- Outputs:
 - Hiển thị danh sách sản phẩm lên trên table
 - Chọn các lô bánh mong muốn huỷ
- Control: Button lịch sử huỷ bánh
 - Inputs: sự kiện nhấn nút
 - Outputs: chuyển tới form lịch sử huỷ bánh
- Control: Button huỷ bánh
 - Inputs: sự kiện nhấn nút
 - Outputs:
 - Hiển thị man hình dialog cho người dùng xác nhận huỷ các lô bánh mà mình đã chọn
- Xử lý:
 - Đầu tiên gọi [API lấy lô bánh hết hạn] **dữ liệu các lô bánh hết hạn** hiển thị lên table
 - Khi nhấn nút xoá gọi [API huỷ bánh] truyền **dữ liệu huỷ bánh** mà người dùng đã chọn
 - Thông báo thành công
 - Thông báo thất bại hiển thị lỗi

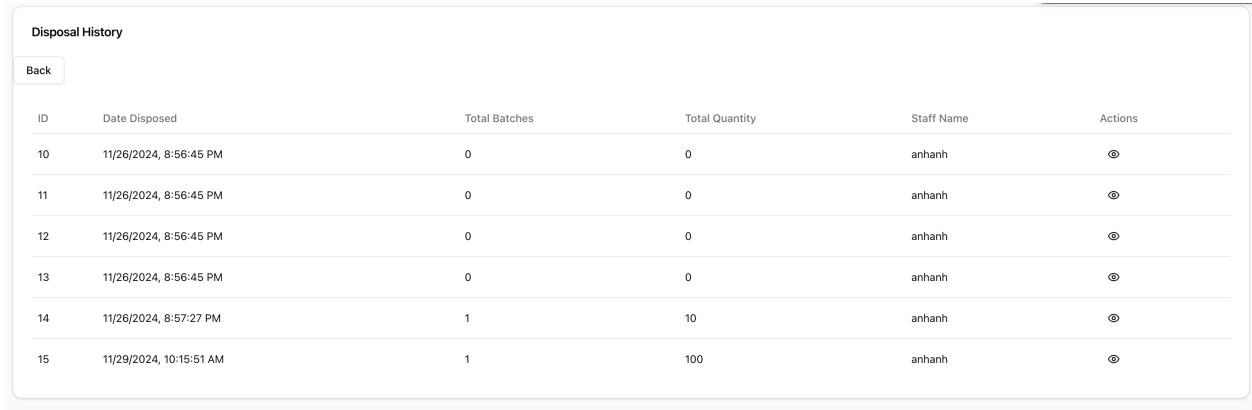
Tùy chỉnh dữ liệu [Form huỷ nhanh]

Tên	Chi tiết	Nguồn
Dữ liệu các lô bánh hết hạn	Id (Id bánh), Name(Tên bánh), Status (trạng thái lô bánh), Expiration_date (Ngày hết hạn), Quantity (số lượng còn lại của lô bánh)	Dữ liệu lấy từ database bảng products, product_batches và được trả thông qua api danh sách lô bánh theo trạng thái với status là Expired

Dữ liệu huỷ bánh	Productbatch_ids (id lô bánh), note (nội dung)	Dữ liệu được lấy từ tác nhân quản lý nhập vào thông qua form huỷ nhanh
------------------	--	--

4.2.1.13. Form lịch sử huỷ bánh

- Tên: Form lịch sử huỷ bánh
- Ảnh:



The screenshot shows a table titled 'Disposal History' with the following columns: ID, Date Disposed, Total Batches, Total Quantity, Staff Name, and Actions. The data in the table is as follows:

ID	Date Disposed	Total Batches	Total Quantity	Staff Name	Actions
10	11/26/2024, 8:56:45 PM	0	0	anhanh	⊕
11	11/26/2024, 8:56:45 PM	0	0	anhanh	⊕
12	11/26/2024, 8:56:45 PM	0	0	anhanh	⊕
13	11/26/2024, 8:56:45 PM	0	0	anhanh	⊕
14	11/26/2024, 8:57:27 PM	1	10	anhanh	⊕
15	11/29/2024, 10:15:51 AM	1	100	anhanh	⊕

Hình ảnh 4. 25 form lịch sử huỷ bánh

- Users: Quản lý
- Control chính trên form:
 - o Control: Button quay về
 - Inputs: sự kiện nhấn nút
 - Outputs: quay về màn hình trước đó
 - o Control: Table lịch sử huỷ bánh
 - Inputs:
 - **Dữ liệu danh sách huỷ bánh** được lấy qua API khi vào trang
 - Có các column:
 - o Id
 - o Date disposed
 - o Total batches
 - o Totals quantity
 - o Staff
 - Outputs:
 - Hiển thị **dữ liệu danh sách huỷ bánh** lên trên table
- Xử lý:
 - o Đầu tiên gọi [API lịch sử huỷ bánh] **Dữ liệu danh sách huỷ bánh** hiển thị lên table

Tùy biến dữ liệu [Form lịch sử huỷ bánh]

Tên	Chi tiết	Nguồn
dữ liệu danh sách huỷ bánh	Id (id huỷ), date_disposed(ngày huỷ), totalBatch(tổng số lô bánh huỷ, totalQuantity(tổng số lượng huỷ), staffName(tên người chịu trách nhiệm huỷ)	Dữ liệu lấy từ database bảng disposed_products, disposed_product_detail

4.2.1.14. Form chi tiết huỷ bánh

- Tên: Form lịch sử huỷ bánh
- Ảnh:

Batch ID	Product Name	Quantity Disposed	Manufacturing Date
11	Crêpe (Bánh kếp kiểu Pháp)	100	11/26/2024

Hình ảnh 4. 26 form chi tiết huỷ bánh

- Users: Quản lý
- Control chính trên form:
 - Control: Button x
 - Inputs: sự kiện nhấn nút
 - Outputs: quay về màn hình trước đó
 - Control: Text-field ngày huỷ bánh
 - Inputs: lấy từ **dữ liệu danh sách lô bánh huỷ**
 - Outputs: hiển thị ngày huỷ lên màn hình
 - Control: Text-field người chịu trách nhiệm huỷ
 - Inputs: lấy từ **dữ liệu danh sách lô bánh huỷ**
 - Outputs: hiển thị tên người huỷ huỷ lên màn hình
 - Control: Text-field ghi chú
 - Inputs: lấy từ **dữ liệu danh sách lô bánh huỷ**
 - Outputs: hiển thị ghi chú lên màn hình

- Control: Table lịch sử huỷ bánh
 - Inputs:
 - **Dữ liệu danh sách lô bánh huỷ** được lấy qua API khi vào trang
 - Có các column:
 - Id
 - Date disposed
 - Total batches
 - Totals quantity
 - Staff
 - Outputs:
 - Hiển thị **dữ liệu danh sách huỷ bánh** lên trên table
- Xử lý:
 - Đầu tiên gọi [API chi tiết huỷ bánh] **dữ liệu danh sách lô bánh huỷ** hiển thị lên table
- **Tùy biến dữ liệu [Form chi tiết huỷ bánh]**

Tên	Chi tiết	Nguồn
dữ liệu danh sách lô bánh huỷ	Id(id huỷ), date_disposed(ngày huỷ), staffName(người chịu trách nhiệm), note(ghi chú), productBatches[] danh sách lô bánh(id, tên, số lượng huỷ)	Dữ liệu lấy từ database bảng disposed_products, disposed_product_detail

4.2.1.15. Form giảm giá nhanh

- Tên: Form giảm giá nhanh
- Ảnh:

The screenshot shows a user interface for managing promotions. On the left, there's a search bar with placeholder text 'Search products...'. Below it is a list of products with a checkbox next to each name. One product, 'Crêpe (Bánh kếp kiểu Pháp)', is selected and highlighted with a yellow background and the status 'NEAR_EXPIRY'. To the right of the list are several status indicators: 'Status' (checkboxes for Name and Crêpe), 'Expiry Date' (2024-11-26), 'Daily Discount' (20%), and 'Countdown' (0 days). On the far right, there's a 'Promotion Settings' panel. It contains a 'Discount (%)' input field set to 0, a 'Skip Default Discount' toggle switch (unchecked), and a 'Get Latest Date' toggle switch (unchecked). There's also an 'End Date' input field with a date picker icon and a 'Create Promotion' button.

Hình ảnh 4. 27 form giảm giá nhanh

- Users: Quản lý
- Control chính trên form:
 - o Control: Button quay về
 - Inputs: sự kiện nhấn nút
 - Outputs: quay về màn hình trước đó
 - o Control: Table lô bánh
 - Inputs:
 - Dữ liệu danh sách các lô bánh hết hạn được lấy từ database thông qua API
 - Có các column:
 - o Select
 - o Id
 - o Name
 - o Status
 - o Daily discount
 - o Countdown
 - o Quantity
 - Outputs:
 - Hiển thị danh sách sản phẩm lên trên table
 - Chọn các lô bánh mong muốn giảm giá
 - o Control: Text-field phần trăm giảm giá
 - Inputs: dữ liệu người dùng nhập từ bàn phím
 - Outputs: hiển thị dữ liệu từ input lên control
 - o Control: toggle bỏ qua giảm giá mặc định
 - Inputs: sự kiện nhấn nút
 - Outputs: hiển thị giá trị của inputs lên control
 - o Control: toggle lấy ngày cuối cùng
 - Inputs: sự kiện nhấn nút
 - Outputs: hiển thị giá trị của inputs lên control và hiển thị control date time nếu như nó là false và ngược lại
 - o Control: Datetime Picker chọn ngày cuối
 - Inputs: Sự kiện nhấn nút chọn ngày trên
 - Outputs: hiển thị giá trị của inputs lên control và hiển thị control date time nếu như nó là false và ngược lại
 - o Control: Button huỷ bánh
 - Inputs: sự kiện nhấn nút
 - Outputs: Thêm giảm giá tương ứng với sản phẩm đã chọn

- Xử lý:
 - o Đầu tiên gọi [API lấy danh sách lô bánh sắp hết hạn] **dữ liệu lô bánh sắp hết hạn** hiển thị lên table
 - o Khi người dùng nhấn nút gửi [API giảm giá nhanh] với **dữ liệu giảm giá nhanh**
 - Thông báo nếu thành công
 - Thông báo lỗi nếu thất bại

Tùy biến dữ liệu [Form giảm giá nhanh]

Tên	Chi tiết	Nguồn
Dữ liệu các lô bánh hết hạn	Id (Id bánh), Name (Tên bánh), Status (trạng thái lô bánh), Expiration_date (Ngày hết hạn), Quantity (số lượng còn lại của lô bánh)	Dữ liệu lấy từ database bảng products, product_batches và được trả thông qua api danh sách lô bánh theo trạng thái với status là Near_expiry
Dữ liệu giảm giá nhanh	Ids lô bánh, phần trăm giảm giá, có bỏ qua giảm giá mặc định(là sẽ lấy giảm giá quy ước sẵn trong product limit_discount), ngày hết hạn, lấy ngày hết hạn cuối cùng(sẽ bỏ qua ngày hết hạn mà lấy ngày cuối cùng của lô bánh)	Dữ liệu được lấy từ tác nhân quản lý nhập vào

4.2.2. API

4.2.2.1. API Đăng nhập

Tên	API đăng nhập
Ý nghĩa	Nhân viên hoặc quản lý đăng nhập vào website.
Method	Post
Inputs	Dữ liệu đăng nhập
Outputs	<ul style="list-style-type: none"> • isSuccess: Trạng thái xác thực, cho biết liệu đăng nhập đã thành công hay không. • message: Thông điệp trả về. • Data: Dữ liệu đăng nhập thành công

Xử lý	<ul style="list-style-type: none"> Tiền truy cập: API kiểm tra định dạng; Login: số điện thoại gồm 10 chữ số hoặc email đúng định dạng email. Mật khẩu: Mật khẩu tối thiểu 6 kí tự, có chứa chữ hoa, chữ thường và kí tự số. Truy cập CSDL: truy vấn đến bảng User để kiểm tra thông tin đăng nhập của người dùng. Xác thực: Dựa trên kết quả xác thực CSDL sẽ được xác định đăng nhập thành công hay không nếu đăng nhập thành công truy vấn CSDL vào bảng user, role, role_detail lấy thông tin người dùng và quyền hạn hành mã hoá token JWT. Sau đó trả về thông tin cho người dùng Trường hợp thành công <pre>{ "success": true, "message": null, "errorcode": null, "data": { "user": { "id": 1, "email": "email@gmail.com", "roles": ["role người dùng"], "token": "token" } } }</pre> Trường hợp thất bại <pre>{ "success": false, "message": "Invalid login credentials", "errorcode": "1001", "data": null }</pre>
-------	---

Tùy biến dữ liệu [API đăng nhập]

Tên	Chi tiết	Nguồn
Dữ liệu đăng nhập	Tên đăng nhập, mật khẩu	Dữ liệu được nhập từ tác nhân quản lý hoặc nhân viên
Dữ liệu đăng nhập thành công	Dữ liệu người dùng(id , tên, email, role), token JWT	Dữ liệu được lấy ra từ database bảng users, roles, role_user, và token được mã hoá theo SHA512

4.2.2.2. API chi tiết sản phẩm

Tên	API chi tiết sản phẩm
Ý nghĩa	Người dùng muốn xem chi tiết sản phẩm
Method	GET
Inputs	<ul style="list-style-type: none"> Truyền id sản phẩm theo param
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết liệu đăng nhập đã thành công hay không. message: Thông điệp trả về. Data: dữ liệu sản phẩm
Xử lý	<p>Bước 1: Truy vấn bảng Products</p> <ul style="list-style-type: none"> Từ ID sản phẩm được truyền vào, truy vấn bảng Products để lấy thông tin cơ bản Lấy các trường: id, name, description, current_price <p>Bước 2: Lấy thông tin danh mục</p> <ul style="list-style-type: none"> Dùng category_id từ bảng Products để join với bảng Categories Lấy id và name của danh mục <p>Bước 3: Lấy danh sách lô hàng</p> <ul style="list-style-type: none"> Join bảng Products với Product_batches qua product_id Chỉ lấy các lô có status là "ACTIVE" hoặc "NEAR_EXPIRY" Với mỗi lô lấy: id, expiration_date, quantity, status, daily_discount <p>Bước 4: Lấy hình ảnh sản phẩm</p> <ul style="list-style-type: none"> Join bảng Products với Images qua product_id Lấy các url hình ảnh của sản phẩm <p>Bước 5: Ghép nối dữ liệu</p> <ul style="list-style-type: none"> Sau khi có đủ thông tin, ghép nối theo format JSON: Phần chính chứa thông tin sản phẩm Phần category chứa thông tin danh mục Mảng productBatches chứa các lô hàng Mảng imageUrls chứa đường dẫn hình ảnh

Kết quả trả về

- API sẽ trả về JSON có dạng như ví dụ bạn đã cho, bao gồm đầy đủ các thông tin:

- Thông tin sản phẩm (id, name, description, currentPrice)

- Thông tin danh mục (id, name)

- Danh sách lô hàng còn active/near_expiry

- Danh sách hình ảnh

- Trường hợp thành công

{

```
"success": true,
```

```
"message": null,
```

```
"errorcode": null,
```

```
"data": {
```

```
    "id": 11,
```

```
    "name": "Crêpe (Bánh kếp kiểu Pháp)",
```

```
    "description": "Crêpe là một loại bánh mỏng, dẹt, được làm từ bột mì, trứng, sữa, nước và một chút bơ. Bánh được chiên mỏng trên chảo phẳng hoặc chảo chuyên dụng cho crêpe. Crêpe có thể được ăn kèm với nhiều loại topping và nhân, như:",
```

```
    "currentPrice": 50000.0,
```

```
    "category": {
```

```
        "id": 13,
```

```
        "name": "Món tráng miệng"
```

```
    },
```

```
    "productBatches": [
```

```
    {
```

```
        "id": 11,
```

```
        "expirationDate": "2024-11-26T14:17:30",
```

```
        "quantity": 100,
```

```
        "status": "NEAR_EXPIRY",
```

```
        "dailyDiscount": 20
```

```
    },
```

```
    ],
```

```
    "imageUrls": [
```

```
        "/uploads/Screenshot 2024-11-26 at
```

```
        09.14.53.png_518_3575bc99-8962-4e7e-970c-177ad83fc72e",
```

```
    ]
```

```
}
```

	<pre> } • Trường hợp thất bại { "success": false, "message": "Cant not find product", "errorcode": "P011", "data": null } </pre>
--	--

Tùy biến dữ liệu [API chi tiết sản phẩm]

Tên	Chi tiết	Nguồn
Dữ liệu sản phẩm	Id sản phẩm, Tên sản phẩm, giá, mô tả, hình ảnh,các lô hàng(id, giảm giá, số lượng)	Dữ liệu được lấy từ database từ các bảng products, product_batches, images

4.2.2.3. API Lịch sử giá

Tên	API lịch sử giá
Ý nghĩa	Người dùng muốn xem lịch sử giá của sản phẩm
Method	GET
Inputs	<ul style="list-style-type: none"> Truyền id sản phẩm theo param
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết liệu đăng nhập đã thành công hay không. message: Thông điệp trả về. Data: dữ liệu lịch sử giá.
Xử lý	<p>Bước 1: Xác định input</p> <ul style="list-style-type: none"> API nhận vào product_id của sản phẩm cần xem lịch sử giá <p>Bước 2: Truy vấn dữ liệu</p> <ul style="list-style-type: none"> Truy vấn bảng Product_histories để lấy lịch sử giá Lọc theo product_id được truyền vào Sắp xếp theo thời gian mới nhất lên đầu (ORDER BY effective_date DESC) Với mỗi record lấy các thông tin: <ul style="list-style-type: none"> id: ID của lần thay đổi giá price: Giá tại thời điểm đó effective_date: Thời điểm áp dụng giá mới Trường hợp thành công

	<pre>{ "success": true, "message": null, "errorcode": null, "data": [{ "id": 20, "price": 50000.0, "effectiveDate": "2024-11-26T09:16:29" }, { "id": 19, "price": 0.0, "effectiveDate": "2024-11-26T09:16:04" }] }</pre> <ul style="list-style-type: none"> Trường hợp thất bại không có <pre>{ "success": true, "message": null, "errorcode": null, "data": [] }</pre>

Từ điển dữ liệu [API lịch sử giá]

Tên	Chi tiết	Nguồn
Dữ liệu lịch sử giá	Id, giá, ngày áp dụng	Dữ liệu được lấy từ database từ các bảng product_history

4.2.2.4. API Danh sách lô bánh theo trạng thái

Tên	API danh sách lô bánh
Ý nghĩa	Người quản lý muốn danh sách các lô bánh
Method	GET

Inputs	Request param là các trạng thái người dùng muốn xem {"ACTIVE", "NEAR_EXPIRY", "EXPIRED", "DISPOSED"}
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết liệu đăng nhập đã thành công hay không. message: Thông điệp trả về. Data: Dữ liệu lô bánh.
Xử lý	<p>Bước 1: Nhận input</p> <ul style="list-style-type: none"> Nhận product_id để xác định sản phẩm cần lấy thông tin Nhận status để lọc trạng thái lô bánh (VD: EXPIRED, ACTIVE, NEAR_EXPIRY...) <p>Bước 2: Truy vấn thông tin sản phẩm</p> <ul style="list-style-type: none"> Truy vấn bảng Products để lấy thông tin cơ bản của sản phẩm Lấy các trường: id, name <p>Bước 3: Truy vấn lô bánh</p> <ul style="list-style-type: none"> Join với bảng Product_batches qua product_id Lọc theo status được truyền vào Lấy các thông tin của lô: <ul style="list-style-type: none"> id status (trạng thái) daily_discount (giảm giá theo ngày) quantity (số lượng) expiration_date (ngày hết hạn) <p>Bước 4: Tính countdown</p> <ul style="list-style-type: none"> Với mỗi lô, tính số ngày từ hiện tại đến ngày hết hạn Nếu đã hết hạn thì countdown sẽ là số âm Nếu chưa hết hạn thì countdown là số dương <p>Bước 5: Xử lý kết quả</p> <ul style="list-style-type: none"> Gom nhóm thông tin sản phẩm và lô bánh Format ngày tháng theo yêu cầu Thêm thông tin success = true nếu thành công Trả về null cho message và errorcode nếu không có lỗi Trường hợp thành công <ul style="list-style-type: none"> "success": true, "message": null, "errorcode": null, "data": [<ul style="list-style-type: none"> <ul style="list-style-type: none"> "id": 14,

	<pre> "name": "Crêpe (Bánh kếp kiểu Pháp)", "status": "EXPIRED", "dailyDiscount": 0, "quantity": 10, "dateExpiry": "2024-11-04", "countDown": -22 }] } </pre> <ul style="list-style-type: none"> Trường hợp thất bại không có <pre> { "success": true, "message": null, "errorcode": null, "data": [] } </pre>
--	--

Từ điển dữ liệu [API lịch sử giá]

Tên	Chi tiết	Nguồn
Dữ liệu lịch sử giá	Id, giá, ngày áp dụng	Dữ liệu được lấy từ database từ các bảng product_history

4.2.2.5. API Huỷ nhanh

Tên	API huỷ nhanh
Ý nghĩa	Người quản lý muốn huỷ nhanh các lô bánh
Method	POST
Inputs	Dữ liệu huỷ nhanh
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết liệu đăng nhập đã thành công hay không. message: Thông điệp trả về. Data: thành công hay thất bại.
Xử lý	<p>Bước 1: Xác thực và lấy thông tin user</p> <ul style="list-style-type: none"> Lấy token JWT từ request header Decode token để lấy user_id của người đang thực hiện Kiểm tra quyền của user có được phép huỷ sản phẩm không

	<p>Bước 2: Nhận input</p> <ul style="list-style-type: none"> • Nhận note (ghi chú về việc huỷ) • Nhận danh sách ids của các lô bánh cần huỷ <p>Bước 3: Kiểm tra dữ liệu lô bánh</p> <ul style="list-style-type: none"> • Truy vấn bảng Product_batches để lấy thông tin các lô theo ids • Kiểm tra các lô có tồn tại không • Lấy số lượng hiện tại của mỗi lô <p>Bước 4: Tạo bản ghi huỷ sản phẩm</p> <ul style="list-style-type: none"> • Insert vào bảng Disposed_products: • staff_id = user_id đã lấy được • note = ghi chú từ input • date_disposed = thời gian hiện tại • Lưu lại disposed_id vừa tạo <p>Bước 5: Thêm chi tiết huỷ sản phẩm</p> <ul style="list-style-type: none"> • Với mỗi lô bánh: • Insert vào Disposed_product_details: • disposed_product_id = disposed_id vừa tạo • product_batch_id = id của lô • disposed_quantity = số lượng hiện tại của lô <p>Bước 6: Cập nhật lô bánh</p> <ul style="list-style-type: none"> • Update bảng Product_batches với các lô đã huỷ: • Set quantity = 0 • Set status = "DISPOSED" <p>Bước 7: Xử lý transaction</p> <ul style="list-style-type: none"> • Đảm bảo tất cả các thao tác DB được thực hiện trong cùng 1 transaction • Nếu có lỗi thì rollback toàn bộ • Nếu thành công thì commit <p>Bước 8: Trả về kết quả</p> <p>Nếu thành công:</p> <pre>success = true message = null errorcode = null data = "Success"</pre> <p>Nếu thất bại:</p> <pre>success = false message = thông báo lỗi errorcode = mã lỗi • Trường hợp thành công { "success": true,</pre>
--	--

	<pre> "message": null, "errorcode": null, "data": "Success" } • Trường hợp thất bại không có { "success": false, "message": null, "errorcode": null, "data": "Fail" } </pre>
--	--

Từ điển dữ liệu [API huỷ nhanh]

Tên	Chi tiết	Nguồn
Dữ liệu huỷ nhanh	Ghi chú, ids(id các lô bánh muốn huỷ)	Dữ liệu được lấy từ tác nhân quản lý

4.2.2.6. API Thêm giảm giá nhanh

Tên	API thêm giảm giá nhanh
Ý nghĩa	Người quản lý muốn thêm giảm giá nhanh cho các lô bánh
Method	POST
Inputs	<ul style="list-style-type: none"> Discount phần trăm giảm giá productBatchIds danh sách các lô sản phẩm muốn giảm giá endDate ngày kết thúc trong trường hợp không lấy ngày cuối cùng của lô skipDefaultDiscount(boolean) bỏ qua phần trăm giảm mặc định của sản phẩm getLastDate(boolean) lấy ngày cuối cùng của lô hàng
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết liệu đăng nhập đã thành công hay không. message: Thông điệp trả về. Data: thành công hay thất bại.
Xử lý	<p>Bước 1: Nhận input</p> <ul style="list-style-type: none"> Danh sách ids của các lô bánh cần giảm giá Phần trăm giảm giá người dùng nhập

- Flag bỏ qua giảm giá mặc định
- Ngày hết hạn giảm giá
- Flag lấy ngày hết hạn cuối cùng

Bước 2: Xử lý dữ liệu đầu vào

- Kiểm tra quyền của user có được phép thêm giảm giá
- Validate các input:
- Phần trăm giảm giá hợp lệ
- Ngày hết hạn phải lớn hơn ngày hiện tại
- Các lô bánh tồn tại và còn active

Bước 3: Xác định phần trăm giảm giá

- Nếu flag bỏ qua giảm giá mặc định = false:
- Lấy discount_limit từ bảng Products
- So sánh với phần trăm người dùng nhập
- Chọn giá trị nhỏ hơn

Bước 4: Xác định ngày hết hạn

- Nếu flag lấy ngày hết hạn cuối cùng = true:
- Tim lô có expiration_date muộn nhất trong các lô được chọn
- Sử dụng làm ngày hết hạn giảm giá
- Ngược lại sử dụng ngày hết hạn từ input

Bước 5: Tạo bản ghi giảm giá

- Insert vào bảng Daily_discount_product:
- product_batch_ids = danh sách ids
- discount = phần trăm giảm đã xác định
- start_date = ngày hiện tại
- end_date = ngày hết hạn đã xác định

Bước 6: Cập nhật lô sản phẩm

- Update bảng Product_batches:
- Set daily_discount = phần trăm giảm đã xác định
- Cho tất cả các lô trong danh sách ids

Bước 7: Xử lý transaction

- Đảm bảo các thao tác DB trong cùng transaction
- Rollback nếu có lỗi
- Commit nếu thành công

Bước 8: Trả về kết quả

- Nếu thành công:
 - success = true
 - message = null
 - errorcode = null
 - data = "Success"

	<ul style="list-style-type: none"> Trường hợp thành công <pre>{ "success": true, "message": null, "errorcode": null, "data": "Success" }</pre> Trường hợp thất bại không có <pre>{ "success": false, "message": null, "errorcode": null, "data": "Fail" }</pre>
--	---

Tùy biến dữ liệu [API giảm giá nhanh]

Tên	Chi tiết	Nguồn
Dữ liệu giảm giá nhanh	Ids lô bánh, phần trăm giảm giá, có bỏ qua giảm giá mặc định(là sẽ lấy giảm giá quy ước sẵn trong product limit_discount), ngày hết hạn, lấy ngày hết hạn cuối cùng(sẽ bỏ qua ngày hết hạn mà lấy ngày cuối cùng của lô bánh)	Dữ liệu được lấy từ tác nhân quản lý

4.2.2.7. API lấy danh sách hóa đơn theo trạng thái

Tên	API danh sách hóa đơn
Ý nghĩa	Xem các đơn khách hàng đặt
Method	Get
Inputs	<ul style="list-style-type: none"> Trạng thái hóa đơn
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết lấy đơn thành công hay không

	<ul style="list-style-type: none"> message: Thông điệp trả về. Data: Dữ liệu trả về.
Xử lý	<ul style="list-style-type: none"> Truy cập CSDL: API thực hiện lấy các thông tin từ bảng bills,billdetails Trường hợp thành công <pre>{ "success": true, "message": "Q2000", "errorcode": "Operation successful", "data": [{ "content": [{ "billId": 8, "customerName": "John Doe", "customerPhone": "123456789", "paymentMethod": "CASH", "diningOption": "DINE_IN", "billStatus": "PAID", "totalAmount": 19.487299999999998 }] }] }</pre> Trường hợp thất bại do yếu tố nào đó <pre>{ "success": false, "message": "Q1002", "errorcode": "", "data": null }</pre>

4.2.2.8. API tìm kiếm hóa đơn

Tên	API tìm kiếm hóa đơn
Ý nghĩa	Xem các đơn dựa vào thông tin tiêm kiém

Method	Get
Inputs	<ul style="list-style-type: none"> Mã hóa đơn Tên khách hàng Số điện thoại khách hàng
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết lấy thông tin thành công hay không message: Thông điệp trả về. Data: Dữ liệu trả về.
Xử lý	<ul style="list-style-type: none"> Truy cập CSDL: API thực hiện lấy các thông tin từ bảng bills,billdetails Trường hợp thành công Trả về giống lấy danh sách hóa theo trạng thái Trường hợp thất bại do yếu tố nào đó <pre>{ "success": false, "message": "Q1002", "errorcode": "", "data": null }</pre>

4.2.2.9. API cập nhật trạng thái hóa đơn

Tên	API cập nhật trạng thái hóa đơn
Ý nghĩa	Cập nhật trạng thái của hóa đơn
Method	Put
Inputs	<ul style="list-style-type: none"> Mã hóa đơn Trạng thái mới
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết cập nhật thông tin thành công hay không message: Thông điệp trả về. Data: Dữ liệu trả về.
Xử lý	<ul style="list-style-type: none"> Truy cập CSDL: API thực hiện cập nhật thông tin từ bảng bills

	<ul style="list-style-type: none"> Trường hợp thành công <pre>{ "success": true, "message": "Q2000", "errorcode": "Operation successful", "data": { "billId": 26, "oldStatus": "COMPLETED", "newStatus": "PAID" } }</pre>
	<ul style="list-style-type: none"> Trường hợp thất bại do yếu tố nào đó <pre>{ "success": false, "message": "Q1017", "errorcode": "Bill status already update", "data": null }</pre>

4.2.2.10. API tạo link thanh toán

Tên	API tạo link thanh toán
Ý nghĩa	Tạo link truy cập đến trang thanh toán
Method	Get
Inputs	<ul style="list-style-type: none"> Mã hóa đơn
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết tạo link thông tin thành công hay không message: Thông điệp trả về. Data: Dữ liệu trả về.

Xử lý	<ul style="list-style-type: none"> Truy cập CSDL: API thực hiện cập nhật thông tin từ bảng bills Trường hợp thành công <pre>{ "success": true, "message": "Q2000", "errorcode": "Operation successful", "data": { "bin": "970422", "accountNumber": "0002024959971", "accountName": "TRAN VIET ANH", "amount": 19, "description": "CSPVX0XFC69 Thanh toán đơn hàng 21", "orderCode": 21, "currency": "VND", "paymentLinkId": "62fc486585c2486cb19344e344d6c110", "status": "PENDING", "checkoutUrl": "https://pay.payos.vn/web/62fc486585c2486cb19344e344d6c110", "qrCode": "00020101021238570010A00000072701270006970422011300020249599 710208QRIBFTTA53037045402195802VN62380834CSPVX0XFC69 Thanh toán đơn hàng 216304C877" } }</pre> Trường hợp thất bại do yếu tố nào đó <pre>{ "success": false, "message": "", "errorcode": "", "data": null }</pre>
-------	---

4.2.2.11. API nhập nguyên liệu

Tên	API nhập nguyên liệu
Ý nghĩa	Người quản lý muốn nhập thêm nguyên liệu vào kho

Method	POST
Inputs	<ul style="list-style-type: none"> • user_id: id người nhập nguyên liệu • id_supplier: id nhà cung cấp • ingredients: danh sách nguyên liệu cần nhập, gồm id nguyên liệu, số lượng và đơn giá <ul style="list-style-type: none"> - ingredient_id: id nguyên liệu nhập - quantity: số lượng nhập - price: giá nhập
Outputs	<ul style="list-style-type: none"> • Success: Trạng thái xác thực, cho biết nhập nguyên liệu thành công hay thất bại. • message: Thông điệp trả về. • Errorcode: Mã lỗi • Data: Dữ liệu đã truyền vào
Xử lý	<ul style="list-style-type: none"> • Truy cập CSDL: API thực hiện truy vấn đến các bảng ingredients, import_ingredients, import_ingredient_details để lưu lại lịch sử nhập nguyên liệu và chi tiết, sau đó tăng số lượng tồn của các nguyên liệu cần nhập. • Trường hợp thành công: <pre>{ "success": true, "message": null, "errorcode": null, "data": { "id": 30, "user_id": 3, "import_date": "2024-12-10T21:40:32.2347567", "total_amount": 125000.0, "id_supplier": 4, "details": [{ "id": { "import_ingredient_id": 30, "ingredient_id": 1 }, "quantity": 5.0, "price": 10000.0 }] } }</pre>

	<pre> }, { "id": { "import_ingredient_id": 30, "ingredient_id": 2 }, "quantity": 5.0, "price": 15000.0 }] } } </pre> <ul style="list-style-type: none"> • Trường hợp thất bại: <pre> { "success": false, "message": "null", "errorcode": "IMPORT_INGREDIENT_FAIL", "data": null } </pre>
--	--

4.2.2.12. API xuất nguyên liệu

Tên	API xuất nguyên liệu
Ý nghĩa	Người quản lý muốn xuất kho nguyên liệu để làm bánh
Method	POST
Inputs	<ul style="list-style-type: none"> • sender_id: id người xuất nguyên liệu • total_amount: tổng số bánh cần làm • ingredients: danh sách nguyên liệu cần xuất, gồm id nguyên liệu và số lượng <ul style="list-style-type: none"> - ingredient_id: id nguyên liệu cần xuất - quantity: số lượng xuất • products: Danh sách sản phẩm cần làm, gồm id sản phẩm và số lượng <ul style="list-style-type: none"> - product_id: id sản phẩm cần làm - quantity: số lượng sản phẩm cần làm

Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết nhập nguyên liệu thành công hay thất bại. message: Thông điệp trả về. Errorcode: Mã lỗi Data: Dữ liệu đã truyền vào
Xử lý	<ul style="list-style-type: none"> Truy cập CSDL: API thực hiện truy vấn đến các bảng ingredients, export_ingredients, export_ingredient_details, daily_production, product_batches. Đầu tiên kiểm tra xem có daily_production nào là ngày hôm nay hay không, nếu có thì sử dụng id của nó, nếu không thì tạo một dòng mới để sử dụng. Sau đó lưu lại trong product_batches lô sản phẩm mới và lưu lại lịch sử xuất nguyên liệu cùng chi tiết, sau đó trừ đi số lượng tồn của các nguyên liệu cần xuất kho. Trường hợp thành công: <pre>{ "success": true, "message": null, "errorcode": null, "data": { "id": 24, "sender_id": 3, "export_date": "2024-12-10T21:43:47.2742016", "daily_production_id": 11, "total_amount": 1.0, "details": [{ "id": { "export_ingredient_id": 24, "ingredient_id": 1 }, "quantity": 0.5 }, { "id": { "export_ingredient_id": 24, "ingredient_id": 2 }, "quantity": 0.5 }] } }</pre>

	<pre> "quantity": 0.4 }] } } • Trường hợp thất bại: { "success": false, "message": "null", "errorcode": "EXPORT_INGREDIENT_FAIL", "data": null } </pre>
--	---

4.2.2.13. API lịch sử huỷ bánh

Tên	API lịch sử huỷ bánh
Ý nghĩa	Người quản lý muốn xem lịch sử huỷ bánh
Method	GET
Inputs	
Outputs	<ul style="list-style-type: none"> Success: Trạng thái xác thực, cho biết nhập nguyên liệu thành công hay thất bại. message: Thông điệp trả về. Errorcode: Mã lỗi Data: Dữ liệu huỷ bánh
Xử lý	<ul style="list-style-type: none"> Truy cập CSDL: API thực hiện truy vấn đến các bảng disposed_product, disposed_product_details, users. Đầu tiên kiểm tra từ bảng huỷ bánh lấy ra các sản đơn huỷ bánh và từ đó lấy sang bảng chi tiết để lấy được số lượng sản phẩm, số lượng lô bánh đã huỷ và tham chiếu đến bảng users lấy ra tên người huỷ. Trường hợp thành công: <pre> { "success": true, "message": null, "errorcode": null, "data": [{ "id": 1, "date": "2023-10-01T10:00:00Z", "product": "Bánh mì", "details": [{ "product": "Bánh mì", "quantity": 10 }], "user": { "name": "Trần Việt Anh" } }] } </pre>

	<pre> "id": 10, "dateDisposed": "2024-11-26T20:56:45", "totalBatches": 10, "totalQuantityDisposed": 100, "note": null, "staffName": "anhanh" }, { "id": 11, "dateDisposed": "2024-11-26T20:56:45", "totalBatches": 0, "totalQuantityDisposed": 0, "note": null, "staffName": "anhanh" },] } </pre> <ul style="list-style-type: none"> • Trường hợp thất bại: <pre> { "success": false, "message": "null", "errorcode": "GET_ALL_FAIL", "data": null } </pre>
--	--

4.2.2.14. API chi tiết huỷ bánh

Tên	API chi tiết huỷ bánh
Ý nghĩa	Người quản lý muốn xem chi tiết huỷ bánh
Method	GET
Inputs	Id của huỷ bánh truyền vào param
Outputs	<ul style="list-style-type: none"> • Success: Trạng thái xác thực, cho biết nhập nguyên liệu thành công hay thất bại. • message: Thông điệp trả về. • Errorcode: Mã lỗi • Data: Dữ liệu chi tiết huỷ bánh

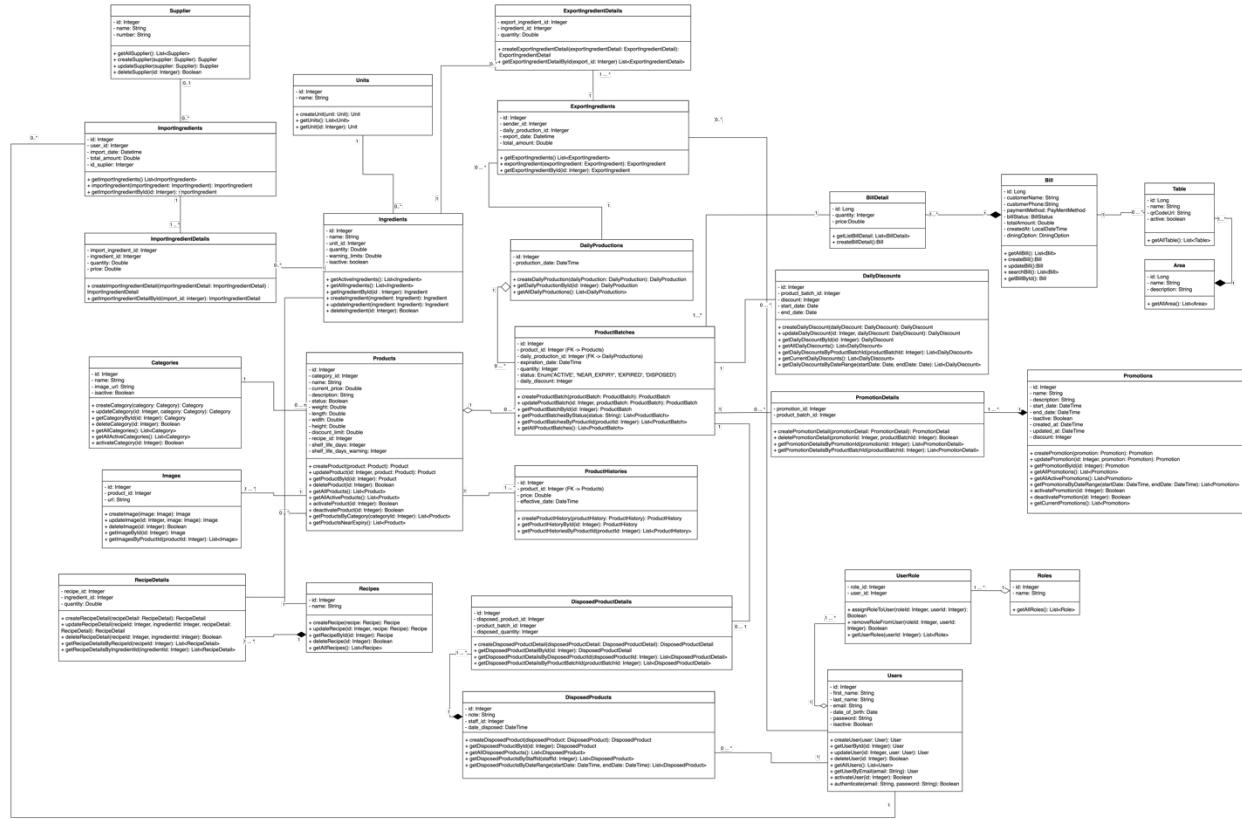
Xử lý	<ul style="list-style-type: none"> Truy cập CSDL: API thực hiện truy vấn đến các bảng disposed_product, disposed_product_details, users. Đầu tiên kiểm tra từ bảng huỷ bánh lấy ra các sản đơn huỷ bánh và từ đó lấy sang bảng chi tiết để lấy được chi tiết các lô bánh và số lượng bánh và tham chiếu đến bảng users lấy ra tên người huỷ. Trường hợp thành công: <pre>{ "success": true, "message": null, "errorcode": null, "data": { "id": 14, "dateDisposed": "2024-11-26T20:57:27", "note": "a", "staffName": "anhanh", "disposedBatches": [{ "batchId": 14, "productName": "Crêpe (Bánh kếp kiểu Pháp)", "disposedQuantity": 10, "manufacturingDate": "2024-11-04T14:44:50" }] } }</pre> <ul style="list-style-type: none"> Trường hợp thất bại: <pre>{ "success": false, "message": "null", "errorcode": "GET_ALL_FAIL", "data": null }</pre>
-------	---

Từ điển dữ liệu [API chi tiết huỷ bánh]

Tên	Chi tiết	Nguồn
Dữ liệu chi tiết huỷ bánh	Id huỷ bánh, ghi chú, ngày huỷ, tên người huỷ, các lô	Dữ liệu được lấy từ bảng products,

	bánh huỷ (id lô bánh, số lượng huỷ, tên bánh)	disposed_products, disposed_product_detail, users
--	---	---

4.3. Sơ đồ lớp



Lược đồ 4. 11 lớp

4.4. Thiết kế CSDL cho phần mềm

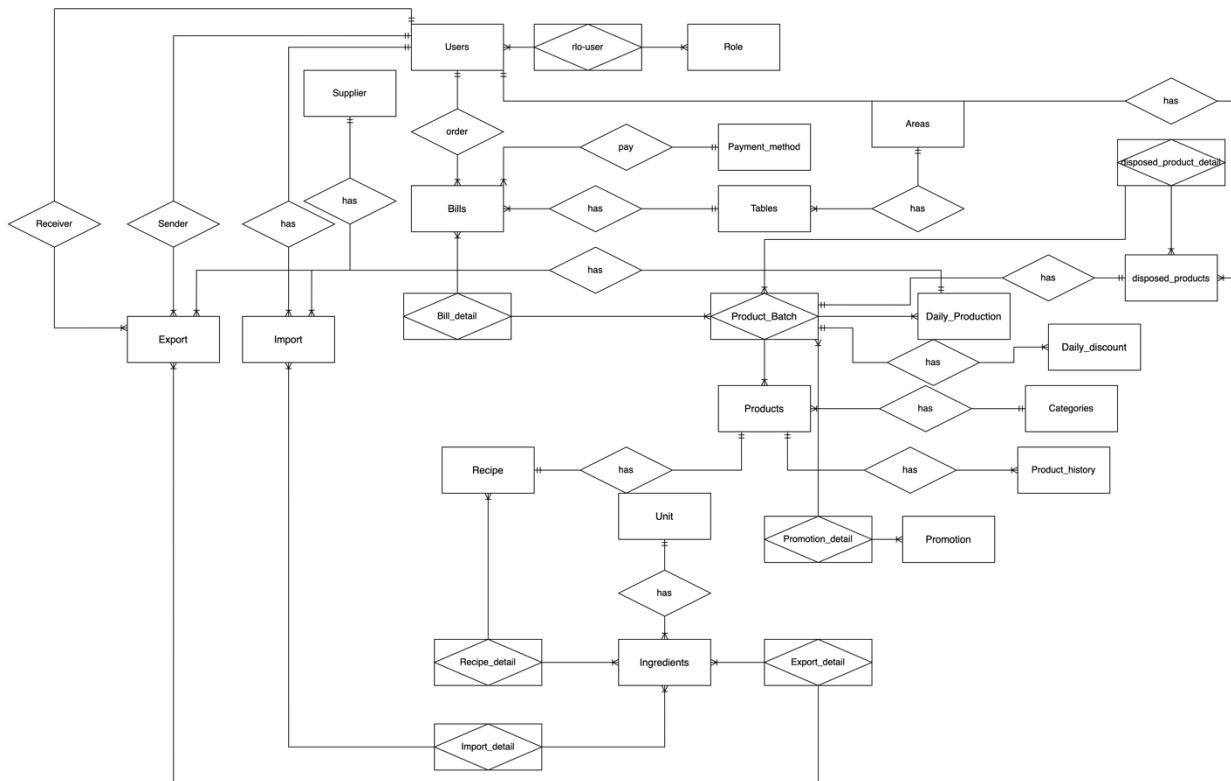
4.4.1. Xác định thực thể

Ghi chú: **Khoá chính**

- **Users**(**user_id**, first_name, last_name, email, password, is_active)
- **Roles**(**role_id**, name)
- **Categories**(**category_id**, name)
- **Product**(**product_id**, name, description, status, weight, length, height, width, shelf_life_day, shelf_life_day_warning)
- **Product_batches**(**product_batch_id**, expiration_date, quantity, status)
- **Promotions**(**promotion_id**, name, description, start_date, end_date, discount)
- **Recipes**(**recipe_id**, name)

- Ingredients(**ingredient_id**, name, quantity, warning_limits, isactive)
- Supplier(**supplier_id**, name, number)
- Daily_production(**daily_production_id**, production_date)
- Bill(**bill_id**, customer_name, customer_phone, total, date_order)
- Table(**table_id**, name)
- Area(**area_id**, name)

4.4.2. Mô hình thực thể ERD



Lược đồ 4. 12 ERD

4.4.3. Thiết kế chi tiết

Ghi chú: Khoá chính, Khoá ngoại, Khoá chính + khoá ngoại

- Users(**id**, first_name, last_name, email, date_of_birth, password, isactive)
- Roles(**id**, name)
- Role_user(**role_id**, **user_id**)
- Areas(**id**, description, name)
- Tables(**id**, active, name, qr_code_url, **area_id**)
- Images(**id**, **product_id**, url)
- Categories(**id**, name, image_url, isactive)
- Units(**id**, name)

- Ingredients(id, name, **unit_id**, warning_limits, isactive, quantity)
- Recipes(id, name)
- Recipe_details(**recipe_id**, **ingredient_id**, quantity)
- Products(id, **category_id**, name, current_price, description, status, weight, length, width, height, discount_limit, **recipe_id**, shelf_life_days, shelf_life_days_warning)
- Product_histories(id, **product_id**, price, effective_date)
- Daily_productions(id, production_date)
- Product_batches(id, **product_id**, **daily_production_id**, expiration_date, quantity, status, daily_discount)
- Promotions(id, name, description, start_date, end_date, isactive, created_at, updated_at, discount)
- Promotion_details(**promotion_id**, **product_batch_id**)
- Daily_discounts(id, **product_batch_id**, discount, start_date, end_date)
- Disposed_products(id, note, **staff_id**, date_disposed)
- Disposed_product_details(id, **disposed_product_id**, **product_batch_id**, disposed_quantity)
- Bills(id, created_at, customer_name, customer_phone, payment_method, payment_status, total_amount, **table_id**, dining_option)
- Bill_details(id, price, quantity, **bill_id**, **product_batch_id**)
- Supplier(id, name, number)
- Import_ingredients(id, **user_id**, import_date, total_amount, created_at, updated_at, **id_supplier**)
- Import_ingredient_details(**import_ingredient_id**, **ingredient_id**, quantity, price)
- Export_ingredients(id, **sender_id**, **daily_production_id**, export_date, total_amount, created_at)
- Export_ingredient_details(**export_ingredient_id**, **ingredient_id**, quantity)

4.4.4. Thiết kế mức hiện thực

Tên bảng	Mô tả	Roles được phép truy cập	Triggers/stored procedure/View
Users	Quản lý thông tin người dùng	Manager: RW	-
Roles	Quản lý vai trò	Manager: R	-
Role_user	Phân quyền người dùng	Manager: RW	-

Areas	Quản lý khu vực	Manager: RW, Employee: R, Customer: R	-
Tables	Quản lý bàn	Manager: RW, Employee: R, Customer: R	-
Images	Lưu trữ hình ảnh sản phẩm	Manager: RW, Employee: R, Customer: R	-
Categories	Danh mục sản phẩm	Manager: RW, Employee: R, Customer: R	-
Units	Đơn vị tính	Manager: RW	-
Ingredients	Quản lý nguyên liệu	Manager: RW	-
Recipes	Công thức	Manager: RW	-
Recipe_details	Chi tiết công thức	Manager: RW	-
Products	Sản phẩm	Manager: RW, Employee: R, Customer: R	-
Product_histories	Lịch sử giá sản phẩm	Manager: RW, Employee:	-

		R, Customer: R	
Daily_productions	Sản xuất hàng ngày	Manager: RW	-
Product_batches	Lô sản phẩm	Manager: RW, Employee: R, Customer: R	Event_update_trang_thai_lo_banh, Event_kiem_tra_hsd, Event_kiem_tra_giam_gia, Trigger_dat_ngay_het_han
Promotions	Khuyến mãi	Manager: RW, Employee: R, Customer: R	-
Promotion_details	Chi tiết khuyến mãi	Manager: RW, Employee: R, Customer: R	-
Daily_discounts	Giảm giá hàng ngày	Manager: RW, Employee: R, Customer: R	-
Disposed_products	Sản phẩm hủy	Manager: RW	-
Disposed_product_details	Chi tiết sản phẩm hủy	Manager: RW	
Bills	Hóa đơn	Manager: RW, Employee: RW, Customer: RW	-
Bill_details	Chi tiết hóa đơn	Manager: RW, Employee:	-

		RW, Customer: RW	
Bill_status_history	Lịch sử trạng thái hóa đơn	Manager: RW, Employee: RW, Customer: RW	-
Supplier	Nhà cung cấp	Manager: RW	-
Import_ingredients	Nhập nguyên liệu	Manager: RW	-
Import_ingredient_details	Chi tiết nhập nguyên liệu	Manager: RW	
Export_ingredients	Xuất nguyên liệu	Manager: RW	-
Export_ingredient_details	Chi tiết xuất nguyên liệu	Manager: RW	-

Event_update_trang_thai_lo_banh

- Inputs: Tự động mỗi tiếng thì chạy một lần
- Output: Tự động cập nhật trạng thái khi **hạn sử dụng còn lại** bé hơn 0 đổi sang trạng thái hết hạn, **hạn sử dụng còn lại** lớn hơn 0 và bé hơn **giới hạn hạn sử dụng** thì đổi trạng thái là gần hết hạn
- Table: Product_batches, products

Bảng từ điển dữ liệu [Event_update_trang_thai_lo_banh]

Tên	Chi tiết	Nguồn
Hạn sử dụng còn lại	Expiration_date – date(now)	Dữ liệu được lấy từ bảng product_batches
Giới hạn hạn sử dụng	Shelf_life_days_warning	Dữ liệu lấy từ bảng products

Event_kiem_tra_han_su_dung

- Inputs: Tự động mỗi 30 phút thì chạy một lần
- Output: tự động xem hết một loạt các lô bánh lô bánh nào hết hạn, hoặc gần hết hạn thì gửi thông báo đến người quản lý đang hoạt động thông qua websocket

- Table: Product_batches

Event_update_product_batch_discounts

- Inputs: Tự động chạy mỗi 30 phút, bắt đầu từ ngày 5/12/2024
- Output: Cập nhật giảm giá cho các lô sản phẩm dựa trên bảng daily_discounts
- Tables:
 - product_batches (UPDATE)
 - daily_discounts (READ)
- Mô tả:
 - Lấy ra các product_batch_id và discount từ bảng daily_discounts mà có ngày hiện tại (CURDATE()) nằm trong khoảng start_date và end_date
 - Cập nhật trường daily_discount trong bảng product_batches với giá trị discount tương ứng
 - Nếu một product_batch không có discount hợp lệ, sẽ set daily_discount = 0
 - Chỉ cập nhật những product_batch có trong bảng daily_discounts

Trigger_dat_ngay_het_han

- Inputs: khi insert vào bảng product_batches
- Output: tự thêm ngày hết hạn vào database
- Tables:
 - product_batches (INSERT)
 - products (READ)
- Mô tả:
 - Trước khi thêm mới một lô sản phẩm (product_batch)
 - Lấy số ngày bảo quản (shelf_life_days) từ bảng products dựa trên product_id của lô sản phẩm mới
 - Tự động tính toán ngày hết hạn (expiration_date) bằng cách:
 - Lấy ngày hiện tại (CURDATE())
 - Cộng thêm số ngày bảo quản (shelf_life_days)
 - Gán giá trị expiration_date cho lô sản phẩm mới trước khi insert

4.5. Bảng tham chiếu

Use-case	Form	API	Table
UC-01 Quản lý hạn sử dụng bánh	- Form đăng nhập. - Form quản lý hạn sử dụng	- API lấy danh sách lô bánh theo trạng thái. - API huỷ nhanh - API thêm giảm giá nhanh - API đăng nhập.	Users, Roles, Role_detail, Products, Product_batches, Daily_discounts, disposed_products, disposed_product_detail

	<ul style="list-style-type: none"> - Form huỷ nhanh - Form giảm giá nhanh 		
UC-02 Chi tiết sản phẩm	<ul style="list-style-type: none"> - Form chi tiết sản phẩm 	<ul style="list-style-type: none"> - API chi tiết sản phẩm - API lịch sử giá 	Products, Product_batches, Product_history, Images, Daily_discounts
UC-03 Quản lý kho nguyên liệu	<ul style="list-style-type: none"> - Form nhập nguyên liệu - Form chọn nguyên liệu - Form xuất nguyên liệu - Form chọn sản phẩm 	<ul style="list-style-type: none"> - API nhập nguyên liệu - API xuất nguyên liệu - API danh sách nguyên liệu - API danh sách đơn vị - API danh sách sản phẩm - API lấy công thức theo sản phẩm - API lấy danh sách nhà cung cấp 	Ingredients, Units, Supplier, Products, Import_ingredients, Import_ingredient_details, Export_ingredients, Export_ingredient_details, Daily_production, Product_batches
UC-04 Gọi món	<ul style="list-style-type: none"> Form chi tiết sản phẩm Form giờ hàng đặt hàng Form quản lý bill Form thanh toán mã qr 	<ul style="list-style-type: none"> - API chi tiết sản phẩm - API tạo hóa đơn - API lấy danh sách hóa đơn theo trạng thái - API cập nhật trạng thái hóa đơn 	Products, Product_batches, Product_history, Images, Daily_discounts, bills, billdetails, table, area

CHƯƠNG 5. KẾT LUẬN

5.1. Kết quả đạt được

Về mặt lý thuyết Qua quá trình thực hiện đề tài, tôi đã đạt được những tiến bộ đáng kể trong việc nghiên cứu và phát triển:

- Nắm vững kiến trúc và công nghệ hiện đại trong phát triển web:
 - Front-end: ReactJS với TypeScript giúp code chặt chẽ và dễ bảo trì hơn
 - Back-end: Spring Boot (Java) với các design pattern và best practices
- Hiểu sâu về quy trình phân tích thiết kế hướng đối tượng (OOP) trong phát triển phần mềm enterprise
- Nắm rõ quy trình nghiệp vụ quản lý cửa hàng bánh ngọt từ khâu nhập liệu đến báo cáo
- Phát triển kỹ năng tự nghiên cứu và học hỏi công nghệ mới

Về mặt thực tiễn Đã xây dựng thành công hệ thống quản lý cửa hàng bánh ngọt với các chức năng chính:

- Quản lý người dùng: Phân quyền admin/nhân viên, đăng ký/đăng nhập/quản lý tài khoản
- Quản lý sản phẩm: CRUD bánh ngọt, phân loại, giá bán, tồn kho
- Quản lý đơn hàng: Tạo đơn, theo dõi trạng thái, thanh toán
- Báo cáo thống kê: Doanh thu, bán hàng, tồn kho theo thời gian

5.2. Hạn chế

Dù đã hoàn thành các chức năng cơ bản, hệ thống vẫn còn một số điểm cần cải thiện:

- Performance: Cần tối ưu hóa queries và caching để tăng tốc độ xử lý
- UI/UX: Giao diện cần được thiết kế responsively hơn cho các thiết bị
- Testing: Cần bổ sung unit test và integration test để đảm bảo chất lượng
- Security: Tăng cường bảo mật, rate limiting và validation chặt chẽ hơn

5.3. Hướng phát triển

Để nâng cao chất lượng và tính năng của hệ thống, các hướng phát triển trong tương lai bao gồm:

Hướng phát triển trong tương lai:

- Cải thiện giao diện người dùng

- Tối ưu hóa trải nghiệm người dùng dựa trên phản hồi thực tế
- Thiết kế giao diện responsive tốt hơn cho nhiều loại thiết bị
- Bổ sung các tính năng tương tác thông minh và trực quan
- Phát triển thêm các báo cáo và biểu đồ trực quan
- Thêm nhiều phương thức thanh toán
 - Tích hợp các cổng thanh toán điện tử phổ biến
 - Tích hợp hệ thống tích điểm và ưu đãi khách hàng
- **Tích hợp trí tuệ nhân tạo**
 - Phát triển hệ thống dự báo nhu cầu nguyên liệu
 - Tự động đề xuất các chiến lược khuyến mãi
 - Phân tích hành vi khách hàng để cá nhân hóa dịch vụ
 - Tối ưu hóa quy trình vận hành dựa trên dữ liệu
- **Mở rộng phạm vi của đế tài**
 - Tích hợp hệ thống quản lý quan hệ khách hàng (CRM)
 - Mở rộng khả năng quản lý nhiều chi nhánh

Tầm nhìn dài hạn:

- Hướng tới xây dựng một hệ thống quản lý toàn diện
- Tạo nền tảng cho việc mở rộng kinh doanh
- Nâng cao khả năng cạnh tranh trong thời đại số
- Đáp ứng nhu cầu ngày càng cao của khách hàng và doanh nghiệp