

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



BÁO CÁO BÀI TẬP LỚN
MÔN: THỰC TẬP CÔNG NGHỆ PHẦN MỀM

SFCS – SMART FOOD COURT SYSTEM

Lớp: L05

GVHD: Lê Đình Thuận

Nhóm Heroic Debuggers – Nhóm 12:

1. Lê Đức Huy – 1810166
2. Nguyễn Gia Huy – 1810173
3. Lê Trung Đan – 1810887
4. Hoàng Tấn Phát – 1811137
5. Đỗ Lê Quang Trung - 1811304

TP. HỒ CHÍ MINH, THÁNG 7/2020

MỤC LỤC

PART 1	1
I. Functional requirements	1
1. Use-case diagram for the whole system	1
2. Use-case detail/scenario for use-cases (interaction functional requirements) that the student is in-charged/ taking care of	2
3. Other non-interactive functional requirement (bonus)	25
II. Non-functional requirements	28
1. General non-functional requirements for the whole system	28
2. Other non-functional requirements related to the features that the student is in-charged	28
PART 2	31
I. Sequence diagram	31
II. Activity diagram	36
III. State Diagram (Bonus)	40
PART 3	42
I. Deployment View (Group work)	42
II. Development/Implementation View (Individual work)	43
PART 4	47
I. Module Interface	47
II. Class Diagram	50
III. Method Descriptions	51
IV. Sequence Diagram	57
V. Activity Diagram	62
VI. Design Pattern (Bonus)	67
VII. A Working Demonstration (Bonus)	69
PHÂN CÔNG CÔNG VIỆC	77
LINK GITHUB	78
TÀI LIỆU THAM KHẢO	79

DANH SÁCH HÌNH VẼ

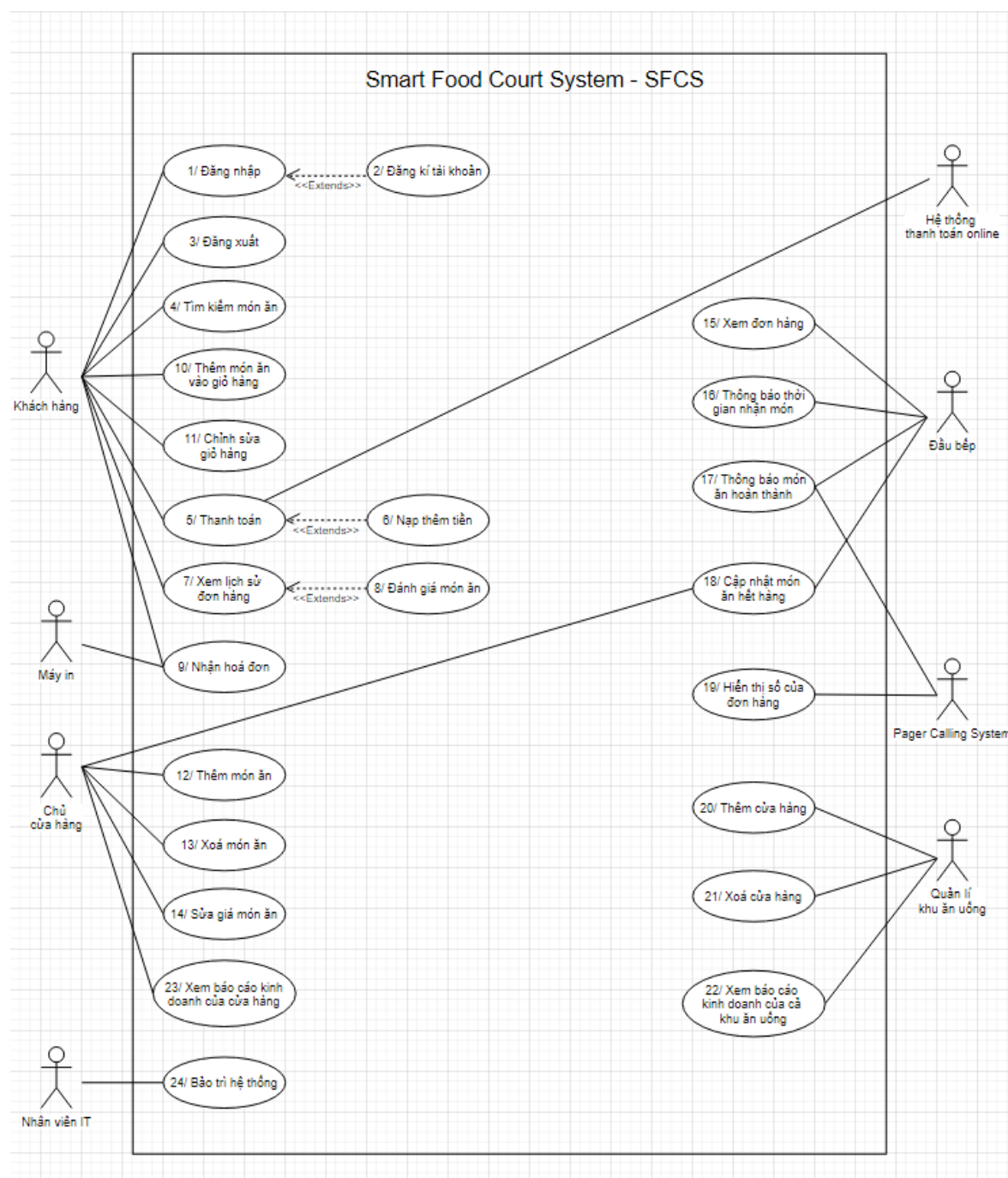
Hình 1: Use-case Diagram toàn hệ thống	1
Hình 2: Sequence diagram cho chức năng "Xử lý tác vụ đăng nhập của khách hàng"	31
Hình 3: Sequence diagram cho chức năng "Thanh toán"	32
Hình 4: Sequence diagram cho chức năng "Thêm món ăn vào giỏ hàng"	33
Hình 5: Sequence diagram cho chức năng "Thông báo món ăn hoàn thành"	34
Hình 6: Sequence diagram cho chức năng "Thêm cửa hàng"	35
Hình 7: Activity diagram cho chức năng "Xử lý tác vụ đăng nhập của khách hàng"	36
Hình 8: Activity diagram cho chức năng "Xem lịch sử và rate"	37
Hình 9: Activity diagram cho chức năng "Thêm món ăn vào giỏ hàng"	38
Hình 10: Activity diagram cho chức năng "Thông báo món ăn hoàn thành"	39
Hình 11: Activity diagram cho chức năng "Thêm cửa hàng"	40
Hình 12: State diagram cho chức năng "Thêm cửa hàng"	40
Hình 13: Deployment diagram cho toàn bộ hệ thống	42
Hình 14: Component diagram của SV Lê Đức Huy – 1810166	43
Hình 15: Package diagram của SV Nguyễn Gia Huy - 1810173	44
Hình 16: Package diagram của SV Lê Trung Đan – 1810887	45
Hình 17: Package diagram của SV Hoàng Tấn Phát - 1811137	45
Hình 18: Package diagram của SV Đỗ Lê Quang Trung - 1811304	46
Hình 19: Interface PagerCallingSystem	47
Hình 20: Interface Order	47
Hình 21: Interface IShoppingCart	48
Hình 22: Interface ICustomer	49
Hình 23: Class diagram cho toàn hệ thống	50
Hình 24: Sequence diagram chi tiết cho chức năng “Xử lý tác vụ đăng nhập, đăng ký tài khoản của khách hàng”	57
Hình 25: Sequence diagram chi tiết cho chức năng “Thanh toán”	58
Hình 26: Sequence diagram chi tiết cho chức năng “Thêm món ăn vào giỏ hàng”	59
Hình 27: Sequence diagram chi tiết cho chức năng “Thông báo món ăn hoàn thành”	60

Hình 28: Sequence diagram chi tiết cho chức năng “Thêm cửa hàng”	61
Hình 29: Activity diagram chi tiết cho chức năng “Xử lý tác vụ đăng nhập, đăng ký tài khoản của khách hàng”	62
Hình 30: Activity diagram chi tiết cho chức năng “Xem lịch sử và rate”	63
Hình 31: Activity diagram chi tiết cho chức năng “Đặt món ăn của khách hàng”	64
Hình 32: Activity diagram chi tiết cho chức năng “Thông báo món ăn hoàn thành”	65
Hình 33: Activity diagram chi tiết cho chức năng “Thêm cửa hàng”	66
Hình 34: Trang chủ SFCS	69
Hình 35: Đăng nhập tài khoản	70
Hình 36: Đăng kí tài khoản	70
Hình 37: Mua hàng thành công	71
Hình 38: Chọn “Store” ở thanh công cụ	72
Hình 39: Danh sách cửa hàng	72
Hình 40: Menu của “Quán ăn Yummy”	73
Hình 41: Tìm kiếm món “gà” và kết quả	73
Hình 42: Trang web quản lý	74
Hình 43: Thêm món “Cơm trộn Hàn Quốc”	74
Hình 44: Món ăn được thêm thành công	75
Hình 45: Chọn “Users” ở web quản lý	75
Hình 46: Danh sách tài khoản	76

PART 1

I. Functional requirements

1. Use-case diagram for the whole system



Hình 1: Use-case Diagram toàn hệ thống

2. Use-case detail/scenario for use-cases (interaction functional requirements) that the student is in-charged/ taking care of

Use-case ID:	1		
Use-case Name:	Đăng nhập		
Created By:	Hoàng Tấn Phát	Last Updated By:	Hoàng Tấn Phát
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Khách hàng		
Description:	Cho phép đăng nhập vào tài khoản để sử dụng các dịch vụ của ứng dụng		
Trigger:	Xuất hiện ngay khi người dùng mở ứng dụng		
Pre-conditions:	- Thiết bị chứa ứng dụng có kết nối internet		
Post-conditions:	- Đăng nhập thành công		
Normal Flow	1. Hệ thống hiển thị giao diện điền thông tin đăng nhập 2. Người sử dụng nhập vào màn hình tên tài khoản và mật khẩu vào 2 khung trống 3. Người sử dụng nhấn vào nút "Đăng nhập" 4. Hệ thống hiển thị xác nhận "Đang đăng nhập"		
Alternative Flows:	A. Người dùng nhập sai tên đăng nhập 4.1. Hệ thống hiển thị "Tài khoản không tồn tại" 4.2. Hệ thống quay lại bước 3 B. Người dùng nhập sai mật khẩu 4.1. Hệ thống hiển thị "Sai mật khẩu" 4.2. Hệ thống quay lại bước 3		
Exceptions:	Nếu không có kết nối internet, hệ thống hiển thị "Kiểm tra lại kết nối internet và thử lại sau"		
Notes and Issues:			
Non-functional:	- Thời gian hiển thị màn hình đăng nhập dưới 2 giây - Thời gian phản hồi của hệ thống dưới 1 giây		

Use-case ID:	2		
Use-case Name:	Đăng kí tài khoản		
Created By:	Hoàng Tấn Phát	Last Updated By:	Hoàng Tấn Phát
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Khách hàng		
Description:	Tạo một tài khoản cá nhân để có thể đăng nhập vào hệ thống		
Trigger:	Nhấn vào nút "Đăng kí"		
Pre-conditions:	- Thiết bị chứa ứng dụng có kết nối internet		
Post-conditions:	- Đăng kí một tài khoản thành công		
Normal Flow	1. Nhấn vào "Đăng kí" 2. Hiện thị giao diện đăng kí tài khoản 3. Nhập vào tên tài khoản và mật khẩu 4. Nhập mật khẩu vào mục "Xác nhận mật khẩu" 5. Nhấn nút "Tạo tài khoản" 6. Hệ thống hiển thị "Đang tạo tài khoản" 7. Hệ thống quay trở về màn hình đăng nhập		
Alternative Flows:	A. Người dùng nhập tài khoản đã tồn tại 5.1. Hệ thống hiển thị "Tài khoản đã tồn tại" 5.2. Hệ thống quay lại bước 2 B. Người dùng nhập 2 mật khẩu khác nhau 5.1. Hệ thống hiển thị "Mật khẩu sai" 5.2. Hệ thống quay lại bước 2 C. Người dùng dừng tạo tài khoản Người dùng nhấn nút "Quay về" và hệ thống nhảy đến bước 7		
Exceptions:	Nếu không có kết nối internet, hệ thống hiển thị "Kiểm tra lại kết nối internet và thử lại sau"		
Notes and Issues:			
Non-functional:	- Tài khoản tạo xong phải được lưu vào hệ thống dưới 3 giây		

Use-case ID:	3		
Use-case Name:	Đăng xuất		
Created By:	Hoàng Tấn Phát	Last Updated By:	Hoàng Tấn Phát
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Khách hàng		
Description:	Đăng xuất ra khỏi ứng dụng		
Trigger:	Nhấn vào nút "Đăng xuất"		
Pre-conditions:	- Thiết bị chứa ứng dụng có kết nối internet - Đã đăng nhập tài khoản trước đó		
Post-conditions:	- Đăng xuất thành công		
Normal Flow	1. Người dùng ấn vào nút "Đăng xuất" 2. Hệ thống xuất hiện hộp thoại "Xác nhận đăng xuất" 3. Người dùng nhấn "Có" 4. Hệ thống trở về màn hình đăng nhập		
Alternative Flows:	3.1. Người dùng nhấn "Không" 3.2. Hộp thoại tắt		
Exceptions:	Nếu không có kết nối internet, hệ thống hiển thị "Kiểm tra lại kết nối internet và thử lại sau"		
Notes and Issues:			
Non-functional:	- Thời gian đăng xuất không quá 2 giây		

Use-case ID:	4		
Use-case Name:	Tìm kiếm món ăn		
Created By:	Hoàng Tấn Phát	Last Updated By:	Hoàng Tấn Phát
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Khách hàng		
Description:	Tìm kiếm món ăn		
Trigger:	Nhấn vào mục "Tìm kiếm món ăn"		
Pre-conditions:	- Thiết bị chứa ứng dụng có kết nối internet		
Post-conditions:	- Xuất hiện kết quả tìm kiếm		
Normal Flow	1. Người dùng nhấn vào mục "Tìm kiếm món ăn"		

	2. Nhập từ khóa vào khung nhập 3. Chọn các bộ lọc 4. Nhấn vào nút "Tìm kiếm" 5. Hệ thống hiển thị kết quả
Alternative Flows:	
Exceptions:	Nếu không có kết nối internet, hệ thống hiển thị "Kiểm tra lại kết nối internet và thử lại sau"
Notes and Issues:	
Non-functional:	- Thời gian liệt kê danh sách đơn hàng dưới 1 giây

Use-case ID:	5		
Use-case Name:	Thanh toán		
Created By:	Lê Trung Đan	Last Updated By:	Lê Trung Đan
Date Created:	02/05/2020	Date Last Updated:	02/05/2020
Actors:	Khách hàng, Hệ thống thanh toán online		
Description:	Khách hàng thanh toán đơn hàng sau khi đã chọn xong (danh sách) món ăn		
Trigger:	Chọn xong (danh sách) món ăn, khách hàng ấn vào nút "Thanh toán"		
Pre-conditions:	<ul style="list-style-type: none">- Món ăn đã được chọn xong- Người dùng có tài khoản tại Food Court đã được kết nối với một dịch vụ thanh toán online- Thiết bị được kết nối mạng		
Post-conditions:	<ul style="list-style-type: none">- Thanh toán đơn hàng thành công- Giỏ hàng tự động được lưu vào lịch sử đơn hàng- Những món ăn trong giỏ được tăng thêm số lần được mua tổng cộng hiển thị trên app/vendor cho người dùng sau lựa chọn- Tài khoản dịch vụ thanh toán bị trừ khoản tiền tương ứng		
Normal Flow	1. Chọn dịch vụ thanh toán đã được kết nối (Momo, ZaloPay,...)		

	2. Xác nhận giao dịch 3. Thực hiện giao dịch
Alternative Flows:	
Exceptions:	Tại bước 2, tài khoản người dùng không còn đủ tiền. Gợi ý đi đến use-case "Nạp thêm tiền"
Notes and Issues:	
Non-functional:	<ul style="list-style-type: none"> - Hỗ trợ nhiều dịch vụ thanh toán online khác nhau - Bảo mật thông tin thanh toán - Kết nối với dịch vụ ngoài dưới 3s - Giao diện gọn gàng, dễ nhìn, tránh bị nhầm lẫn tai hại về tiền của khách hàng

Use-case ID:	6		
Use-case Name:	Nạp thêm tiền		
Created By:	Lê Trung Đan	Last Updated By:	Lê Trung Đan
Date Created:	02/05/2020	Date Last Updated:	02/05/2020
Actors:	Khách hàng, Hệ thống thanh toán online		
Description:	Hỗ trợ nạp thêm tiền vào dịch vụ được kết nối		
Trigger:	Người dùng bấm vào nút "Nạp thêm tiền" tại màn hình thanh toán		
Pre-conditions:	<ul style="list-style-type: none"> - Thiết bị có kết nối mạng - Tài khoản đã được kết nối với một dịch vụ thanh toán online - Dịch vụ thanh toán online được kết nối với ngân hàng điện tử hoặc cho phép nạp tiền từ sim điện thoại 		
Post-conditions:	- Tài khoản giao dịch online tương ứng được nạp thêm tiền		
Normal Flow	1. Chuyển hướng đến giao diện nạp tiền của dịch vụ được kết nối tương ứng		
Alternative Flows:			
Exceptions:			
Notes and Issues:	- Mọi thao tác trên tài khoản thanh toán được hiện thực bởi dịch vụ bên thứ 3		

Non-functional:	<ul style="list-style-type: none">- Tuyệt đối bảo mật thông tin đăng nhập của bên thứ 3, đặc biệt là khi kết nối với tài khoản ngân hàng- Ẩn (chỉ hiển thị *) khi người dùng đăng nhập trên vendor công cộng, kể cả tên đăng nhập
-----------------	--

Use-case ID:	7		
Use-case Name:	Xem lịch sử đơn hàng		
Created By:	Lê Trung Đan	Last Updated By:	Lê Trung Đan
Date Created:	02/05/2020	Date Last Updated:	02/05/2020
Actors:	Khách hàng		
Description:	Hiển thị danh sách những món ăn, cùng với giá tiền, ngày tháng, cửa hàng,... mà khách hàng đã từng thanh toán		
Trigger:	Người dùng nhấn vào nút "Lịch sử" trên giao diện đã đăng nhập		
Pre-conditions:	<ul style="list-style-type: none">- Thiết bị có kết nối mạng- Tài khoản Food Court đã được đăng nhập		
Post-conditions:			
Normal Flow	<ol style="list-style-type: none">1. Hệ thống lấy dữ liệu từ máy chủ có chứa thông tin tương ứng của khách hàng2. Khách hàng kéo màn hình lên xuống, để xem lại lịch sử mua hàng3. Khách hàng nhấn thoát ra		
Alternative Flows:			
Exceptions:			
Notes and Issues:	<ul style="list-style-type: none">- Cho phép khách hàng đánh giá món ăn trong lịch sử		
Non-functional:	<ul style="list-style-type: none">- Giao diện dễ nhìn, vừa đủ thông tin, không bị nhàm lẫn		

Use-case ID:	8		
Use-case Name:	Đánh giá món ăn		
Created By:	Lê Trung Đan	Last Updated By:	Lê Trung Đan
Date Created:	02/05/2020	Date Last Updated:	02/05/2020
Actors:	Khách hàng		
Description:	Khách hàng đánh giá món ăn đã từng thanh toán, ngoài ra có thể thêm bình luận,...		
Trigger:	Khách hàng nhấn vào "Cho đánh giá" trên màn hình lịch sử món ăn		
Pre-conditions:	<ul style="list-style-type: none">- Thiết bị có kết nối mạng- Tài khoản Food Court đã được đăng nhập- Lịch sử món ăn lớn hơn 0		
Post-conditions:	<ul style="list-style-type: none">- Những món ăn được đánh giá sẽ được cập nhật số lượt đánh giá và số sao trung bình, hiển thị công khai bình luận của người dùng		
Normal Flow	<ol style="list-style-type: none">1. Người dùng chọn một món ăn để cho điểm2. Cho số sao mà mình thấy xứng đáng từ 1 đến 5, ngoài ra có thể thêm góp ý và đánh giá chi tiết, trả lời các câu hỏi như về vấn đề giá cả hợp lý, hợp vệ sinh,...3. Nhấn "Xong"4. Hoàn thành đánh giá		
Alternative Flows:	Tại bước 3, sau khi đánh giá xong một món, người dùng có thể quay lại bước 1 đánh giá thêm một món nữa hoặc tiến tới bước 4 và kết thúc		
Exceptions:			
Notes and Issues:	<ul style="list-style-type: none">- Khách hàng đánh giá bằng những từ ngữ không hợp thuần phong mỹ tục, trong trường hợp đó số sao vẫn được ghi lại nhưng bình luận sẽ bị che mất		
Non-functional:	<ul style="list-style-type: none">- Giao diện trực quan, dễ nhìn, giúp người dùng đánh giá đúng trọng tâm		

Use-case ID:	9		
Use-case Name:	Nhận hoá đơn		
Created By:	Lê Trung Đan	Last Updated By:	Lê Trung Đan
Date Created:	02/05/2020	Date Last Updated:	02/05/2020
Actors:	Khách hàng, Máy in		
Description:	In ra hoá đơn cho khách hàng tại vendor hoặc tạo hoá đơn điện tử trên mobile app		
Trigger:	Thanh toán thành công		
Pre-conditions:	- Các thông tin đã được cung cấp đầy đủ		
Post-conditions:	- Hoá đơn giấy được in ra hoặc hiển thị hoá đơn điện tử trên mobile app		
Normal Flow	1. Sau khi thanh toán xong, người dùng đợi vendor in ra hoá đơn giấy hoặc ấn vào nút "Hoá đơn" trên mobile app		
Alternative Flows:			
Exceptions:			
Notes and Issues:			
Non-functional:	- Thời gian in trong vòng 4s, hoá đơn vừa đủ cầm tay không quá to quá nhỏ - Chữ trên hoá đơn vừa đủ nhìn, không quá nhỏ, ưu tiên dùng các loại font monospace để tránh nhầm lẫn chữ, số, nhưng đồng thời hỗ trợ hiển thị món ăn bằng tiếng Việt		

Use-case ID:	10		
Use-case Name:	Thêm món ăn vào giỏ hàng		
Created By:	Nguyễn Gia Huy	Last Updated By:	Nguyễn Gia Huy
Date Created:	26/04/2020	Date Last Updated:	26/04/2020
Actors:	Khách hàng		
Description:	Khách hàng thêm món ăn muốn mua cùng số lượng vào giỏ hàng		
Trigger:	Khách hàng muốn mua một món ăn nào đó trong menu		
Pre-conditions:	- Khách hàng đang ở trang danh sách các món ăn		

	- Hệ thống phải có ít nhất một món ăn có thể đặt
Post-conditions:	- Dữ liệu về giỏ hàng của khách hàng được cập nhật
Normal Flow	1. Khách hàng bấm chọn vào món ăn muốn mua 2. Giao diện sẽ hiển thị cửa sổ gồm thông tin chi tiết về món ăn và textbox để khách hàng nhập số lượng cần mua 3. Khách hàng nhập số lượng cần mua và bấm nút “Thêm món ăn vào giỏ hàng” 4. Hệ thống sẽ cập nhật món mới vào đơn hàng và tính toán tổng giá tiền đơn hàng 5. Thông báo thêm món ăn thành công cho khách hàng
Alternative Flows:	
Exceptions:	Exception 1: Tại bước 2 2a. Nếu món ăn không có sẵn, hệ thống chỉ hiện thông tin món ăn, ấn nút “Thêm món ăn vào giỏ hàng ” và hiện thông báo “Hiện tại không thể đặt món ăn này”
Notes and Issues:	- Số lượng mặc định là 1 và khách hàng chỉ được nhập số nguyên dương
Non-functional:	- Dữ liệu đơn hàng được lưu lại và tính toán giá tiền nhanh không quá 1 giây để trả lại thông báo cho khách hàng - Tính toán giá tiền chính xác

Use-case ID:	11		
Use-case Name:	Chỉnh sửa giỏ hàng		
Created By:	Nguyễn Gia Huy	Last Updated By:	Nguyễn Gia Huy
Date Created:	26/04/2020	Date Last Updated:	26/04/2020
Actors:	Khách hàng		
Description:	Khách hàng có thể chỉnh sửa số lượng của một món ăn đã thêm vào giỏ hàng		
Trigger:	Khách hàng muốn thay đổi các món ăn đã thêm vào giỏ hàng		
Pre-conditions:	- Khách hàng đang ở trang chủ của ứng dụng - Người dùng có ít nhất một món ăn đã được thêm vào giỏ		

	hàng
Post-conditions:	- Dữ liệu về đơn hàng của khách hàng được cập nhật
Normal Flow	<ol style="list-style-type: none"> 1. Khách hàng bấm vào nút “Giỏ hàng” 2. Giao diện hiển thị cửa sổ gồm các món ăn đã được khách hàng thêm vào giỏ hàng (số lượng, tổng giá tiền) 3. Khách hàng bấm vào số lượng của một món hàng và thay đổi số lượng cần mua 4. Hệ thống sẽ cập nhật lại dữ liệu đơn hàng và tính toán lại tổng giá tiền đơn hàng 5. Hệ thống hiển thị đơn hàng mới cho khách hàng
Alternative Flows:	<p>Alternative 1: tại bước thứ 3</p> <p>3a. Người dùng bấm phím “X” cạnh món ăn để bỏ chọn món đó khỏi đơn hàng</p> <p><i>Tiếp tục bước 4 trong normal flow</i></p> <p>Alternative 2: tại bước 5</p> <p>5a. Nếu khách hàng đã hủy hết món ăn trong giỏ hàng, hệ thống hiển thị cửa sổ thông báo không có món ăn nào trong giỏ hàng</p> <p><i>Kết thúc flow</i></p>
Exceptions:	<p>Exception 1: tại bước 2</p> <p>2a. Nếu không có món ăn nào trong giỏ hàng, hệ thống hiển thị cửa sổ thông báo không có món ăn nào trong giỏ hàng</p>
Notes and Issues:	- Chỉ cho phép khách hàng chỉnh số lượng là số nguyên dương
Non-functional:	<ul style="list-style-type: none"> - Dữ liệu đơn hàng được lưu lại và tính toán giá tiền nhanh chóng (không quá 0.5 giây để hiển thị giỏ hàng đã cập nhật lại cho khách hàng) - Tính toán giá tiền chính xác

Use-case ID:	12		
Use-case Name:	Thêm món ăn		
Created By:	Nguyễn Gia Huy	Last Updated By:	Nguyễn Gia Huy
Date Created:	26/04/2020	Date Last Updated:	26/04/2020
Actors:	Chủ cửa hàng		
Description:	Chủ cửa hàng thêm một món ăn mới vào danh sách các món ăn của cửa hàng mình. Thông tin món ăn thêm vào bao gồm: tên món ăn, hình ảnh mô tả, giá, trạng thái (còn hàng hay hết hàng)		
Trigger:	Chủ cửa hàng quyết định bán thêm một món ăn mới		
Pre-conditions:	- Chủ cửa hàng đang ở trang quản lý cửa hàng của mình		
Post-conditions:	- Thông tin về món ăn mới được cập nhật lên cơ sở dữ liệu của hệ thống		
Normal Flow	<ol style="list-style-type: none"> 1. Chủ cửa hàng bấm vào mục “Quản lý menu món ăn” 2. Hệ thống hiển thị danh sách các món ăn hiện có của cửa hàng 3. Chủ cửa hàng bấm vào nút “Thêm món ăn mới” 4. Hệ thống hiển thị form biểu mẫu để nhập thông tin món ăn bao gồm : tên món ăn, hình ảnh mô tả, giá cả, trạng thái (còn hàng hoặc hết hàng) 5. Chủ cửa hàng điền thông tin cho món ăn mới và thêm hình ảnh mô tả cho món ăn từ bộ nhớ của máy 6. Hệ thống load ảnh món ăn để chủ cửa hàng xem trước 7. Chủ cửa hàng bấm nút “Lưu” 8. Hệ thống lưu thông tin vào cơ sở dữ liệu và trả về thông báo hoàn thành 9. Hệ thống hiển thị lại danh sách các món ăn hiện có sau khi đã cập nhật 		
Alternative Flows:	<p>Alternative 1: tại bước 5</p> <ol style="list-style-type: none"> 5a. Chủ cửa hàng chọn thêm hình ảnh từ web 5b. Hệ thống hiển thị textbox để người dùng nhập URL của hình ảnh 5c. Chủ cửa hàng nhập/dán URL hình ảnh vào ô này 		

	<i>Tiếp tục bước 6 trong normal flow</i>
Exceptions:	<p>Exception 1: tại bước 5</p> <p>5a. Nếu chủ cửa hàng nhập sai một trường thông tin, hệ thống sẽ tô đỏ trường đó, thông báo lỗi sai và yêu cầu nhập lại ⁽¹⁾</p> <p>Exception 2: tại bước 7</p> <p>7a. Nếu chủ cửa hàng chưa nhập đủ thông tin món ăn mà đã bấm “Lưu”, hệ thống sẽ yêu cầu người dùng nhập lại</p>
Notes and Issues:	<p>(1) Yêu cầu của mỗi trường thông tin:</p> <ul style="list-style-type: none"> - Tên món ăn : Không được trùng với các món đã có - Giá tiền : Số nguyên dương ≥ 1000 (VNĐ) - Hình ảnh mô tả: file hình ảnh hợp lệ (*.png, *.jpg) - Trạng thái: là một radio-button gồm 2 lựa chọn là “Còn hàng” và “Hết hàng”
Non-functional:	<ul style="list-style-type: none"> - Hệ thống phải tạo một ID riêng biệt cho mỗi món ăn được thêm vào - Hệ thống truy xuất và lưu thông tin vào cơ sở dữ liệu trong vòng dưới 2 giây

Use-case ID:	13		
Use-case Name:	Xóa món ăn		
Created By:	Nguyễn Gia Huy	Last Updated By:	Nguyễn Gia Huy
Date Created:	26/04/2020	Date Last Updated:	26/04/2020
Actors:	Chủ cửa hàng		
Description:	Chủ cửa hàng xóa một món ăn khỏi menu của cửa hàng mình		
Trigger:	Chủ cửa hàng không kinh doanh một món ăn nào đó nữa		
Pre-conditions:	- Chủ cửa hàng đang ở trang quản lý cửa hàng của mình - Món ăn muốn xóa đang tồn tại trong dữ liệu của cửa hàng		
Post-conditions:	- Dữ liệu của món ăn bị xóa khỏi cơ sở dữ liệu của hệ thống		
Normal Flow	1. Chủ cửa hàng bấm vào mục “Quản lý menu món ăn” 2. Hệ thống hiển thị danh sách các món ăn hiện có của cửa hàng		

	3. Chủ cửa hàng bấm vào món ăn muốn xóa 4. Hệ thống hiển thị trang chứa thông tin của món ăn đó 5. Chủ cửa hàng bấm nút “Xóa món ăn” 6. Hệ thống hiển thị thông báo yêu cầu chủ cửa hàng xác nhận xóa món ăn này 7. Chủ cửa hàng bấm nút “Đồng ý” 8. Hệ thống xóa thông tin về món ăn trong cơ sở dữ liệu và trả về thông báo hoàn thành 9. Hệ thống hiển thị lại danh sách các món ăn hiện có sau khi đã cập nhật
Alternative Flows:	
Exceptions:	Exception 1: tại bước 2 2a. Nếu không có món ăn nào trong danh sách, hệ thống sẽ hiển thị trang trống cùng thông báo chưa có món ăn được thêm vào Exception 2: tại bước 7 7a. Nếu chủ cửa hàng bấm “Hủy”, hệ thống sẽ trở về trang thông tin món ăn
Notes and Issues:	
Non-functional:	- Hệ thống truy xuất và thay đổi thông tin trong cơ sở dữ liệu trong vòng dưới 2 giây

Use-case ID:	14		
Use-case Name:	Sửa giá món ăn		
Created By:	Nguyễn Gia Huy	Last Updated By:	Nguyễn Gia Huy
Date Created:	26/04/2020	Date Last Updated:	26/04/2020
Actors:	Chủ cửa hàng		
Description:	Chủ cửa hàng chỉnh sửa giá của một món ăn trong menu của cửa hàng mình		
Trigger:	Chủ cửa hàng muốn thay đổi giá tiền của một món ăn trong menu		

Pre-conditions:	- Chủ cửa hàng đang ở trang quản lý của hệ thống - Món ăn muốn sửa giá phải tồn tại trong dữ liệu của cửa hàng
Post-conditions:	- Dữ liệu của món ăn được thay đổi trên cơ sở dữ liệu của hệ thống
Normal Flow	1. Chủ cửa hàng bấm vào mục “Quản lý menu món ăn” 2. Hệ thống hiển thị danh sách các món ăn hiện có của cửa hàng 3. Chủ cửa hàng bấm vào món ăn muốn chỉnh sửa 4. Hệ thống hiển thị trang chứa thông tin của món ăn đó 5. Chủ cửa hàng bấm nút “Chỉnh sửa” 6. Hệ thống mở khóa các trường thông tin cho người dùng chỉnh sửa 7. Chủ cửa hàng thay đổi giá món ăn trong trường “Giá tiền” 8. Chủ cửa hàng bấm nút “Lưu” 9. Hệ thống cập nhật thông tin về món ăn trong cơ sở dữ liệu và trả về thông báo hoàn thành 10. Hệ thống hiển thị lại danh sách các món ăn hiện có sau khi đã cập nhật
Alternative Flows:	
Exceptions:	Exception 1: tại bước 2 2a. Nếu không có món ăn nào trong danh sách, hệ thống sẽ hiển thị trang trắng cùng thông báo chưa có món ăn được thêm vào Exception 2: tại bước 7 7a. Nếu chủ cửa hàng nhập giá không đúng quy định hoặc để trống, hệ thống sẽ tô đỏ trường đó, thông báo lỗi sai và yêu cầu nhập lại
Notes and Issues:	- Giá tiền : số nguyên dương ≥ 1000 (VNĐ)
Non-functional:	- Hệ thống truy xuất và lưu thông tin vào cơ sở dữ liệu trong vòng dưới 2 giây

Use-case ID:	15		
Use-case Name:	Xem đơn hàng		
Created By:	Lê Đức Huy	Last Updated By:	Lê Đức Huy
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Đầu bếp		
Description:	Cho phép đầu bếp xem đơn hàng để làm thức ăn		
Trigger:	Đầu bếp nhấn vào mục “Xem đơn hàng”		
Pre-conditions:	- Thiết bị có kết nối Internet		
Post-conditions:	- Đầu bếp xem đơn hàng thành công - Đơn hàng hiển thị trạng thái “Đã xem”		
Normal Flow	1. Đầu bếp nhấn vào mục “Xem đơn hàng” 2. Hệ thống liệt kê danh sách các đơn hàng đang đợi 3. Đầu bếp bấm vào đơn hàng muốn xem 4. Hệ thống hiển thị đơn hàng tương ứng 5. Đơn hàng hiển thị trạng thái “Đã xem”		
Alternative Flows:	<i>Alternative 1: tại bước 2</i> 2a. Nếu đơn hàng rỗng, hệ thống hiển thị “Không có đơn hàng nào” Kết thúc flow <i>Alternative 2: tại bước 5</i> 5a. Đơn hàng hiển thị trạng thái “Đã xem”, đầu bếp bấm nút “Quay lại” Use-case quay về bước 2 trong normal flow		
Exceptions:	<i>Exception 1: tại bước 1</i> 1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không có kết nối Internet”		
Notes and Issues:			
Non-functional:	- Thời gian liệt kê danh sách đơn hàng dưới 1s		

Use-case ID:	16
Use-case Name:	Thông báo thời gian nhận món

Created By:	Lê Đức Huy	Last Updated By:	Lê Đức Huy
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Đầu bếp		
Description:	Cho phép đầu bếp thông báo thời gian nhận món đến khách hàng sử dụng app mobile		
Trigger:	Đầu bếp nhấn vào mục “Thông báo thời gian nhận món”		
Pre-conditions:	<ul style="list-style-type: none"> - Thiết bị có kết nối Internet - Khách hàng đặt hàng bằng app mobile 		
Post-conditions:	- Thời gian nhận món được chuyển đến khách hàng		
Normal Flow	1. Đầu bếp nhấn vào mục “Thông báo thời gian nhận món” 2. Hệ thống hiển thị bàn phím số 3. Đầu bếp nhập con số phù hợp với thời gian ước tính cần để làm món ăn 4. Đầu bếp bấm “Xác nhận” 5. Hệ thống gửi thông báo “Món ăn sẽ được hoàn thành sau x phút” đến khách hàng, với x là con số tương ứng đầu bếp nhập		
Alternative Flows:	<i>Alternative 1: tại bước 4</i> 4a. Đầu bếp bấm “Nhập lại” Use-case quay về bước 2 trong normal flow		
Exceptions:	<i>Exception 1: tại bước 1</i> 1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không có kết nối Internet”		
Notes and Issues:	- Số phút đầu bếp nhập vào là số tự nhiên		
Non-functional:	- Thông báo được gửi đến khách hàng qua app mobile trong thời gian không quá 1s		

Use-case ID:	17		
Use-case Name:	Thông báo món ăn hoàn thành		
Created By:	Lê Đức Huy	Last Updated By:	Lê Đức Huy
Date Created:	25/04/2020	Date Last Updated:	25/04/2020

Actors:	Đầu bếp, Pager Calling System
Description:	Cho phép đầu bếp thông báo món ăn hoàn thành đến khách hàng
Trigger:	Đầu bếp nhấn vào mục “Thông báo món ăn hoàn thành”
Pre-conditions:	<ul style="list-style-type: none">- Thiết bị có kết nối Internet- Khách hàng đặt món tại khu ăn uống và lấy pager có số tương ứng với số trên hoá đơn. Hoặc khách hàng đặt món bằng app mobile
Post-conditions:	- Thông báo món ăn hoàn thành được gửi đến khách hàng
Normal Flow	<ol style="list-style-type: none">1. Đầu bếp nhấn vào mục “Thông báo món ăn hoàn thành”2. Hệ thống liệt kê danh sách các đơn hàng đang đợi3. Đầu bếp bấm vào đơn hàng đã làm thức ăn xong4. Đầu bếp bấm “Hoàn thành”5. Pager có số tương ứng với số hoá đơn của món ăn đầu bếp làm xong sẽ phát ra âm thanh thông báo cho khách hàng biết để họ tới nhận món6. Khi khách hàng nhận món xong, Pager sẽ xóa số hoá đơn tương ứng, quay lại trạng thái chờ
Alternative Flows:	<i>Alternative 1: tại bước 4</i> 4a. Đầu bếp bấm “Quay lại”. Use-case quay về bước 2 trong normal flow <i>Alternative 2: tại bước 5</i> 5a. Nếu đơn hàng được đặt qua app mobile, hệ thống sẽ gửi thông báo “Món ăn hoàn thành” đến khách hàng qua ứng dụng Kết thúc flow
Exceptions:	<i>Exception 1: tại bước 1</i> 1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không có kết nối Internet”
Notes and Issues:	
Non-functional:	<ul style="list-style-type: none">- Thông báo được gửi qua Pager không quá 0.5s- Thông báo được gửi qua apple mobile không quá 1s

Use-case ID:	18		
Use-case Name:	Cập nhật món ăn hết hàng		
Created By:	Lê Đức Huy	Last Updated By:	Lê Đức Huy
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Đầu bếp, Chủ cửa hàng		
Description:	Cho phép đầu bếp, chủ cửa hàng cập nhật món ăn đã hết		
Trigger:	Đầu bếp, chủ cửa hàng nhấn vào mục “Cập nhật món ăn hết hàng”		
Pre-conditions:	- Thiết bị có kết nối Internet		
Post-conditions:	- Món ăn hết hàng được cập nhật, khách hàng không thể chọn chúng trong danh sách món ăn		
Normal Flow	1. Đầu bếp, chủ cửa hàng nhấn vào mục “Cập nhật món ăn hết hàng” 2. Hệ thống liệt kê danh sách món ăn 3. Đầu bếp, chủ cửa hàng bấm vào món ăn muốn thông báo hết hàng 4. Đầu bếp, chủ cửa hàng bấm “Xác nhận” 5. Khi khách hàng chọn món, món ăn hết hàng sẽ bị làm mờ, khách hàng không thể chọn chúng		
Alternative Flows:	<i>Alternative 1: tại bước 4</i> 4a. Đầu bếp, chủ cửa hàng bấm “Quay lại” Use-case quay về bước 2 trong normal flow		
Exceptions:	<i>Exception 1: tại bước 1</i> 1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không có kết nối Internet”		
Notes and Issues:			
Non-functional:	- Sau khi đầu bếp, chủ cửa hàng hoàn thành “Cập nhật món ăn hết hàng”, trong thời gian không quá 3s, món ăn phải được làm mờ để khách hàng không chọn được		

Use-case ID:	19		
Use-case Name:	Hiển thị số của đơn hàng		
Created By:	Lê Đức Huy	Last Updated By:	Lê Đức Huy
Date Created:	25/04/2020	Date Last Updated:	25/04/2020
Actors:	Pager Calling System		
Description:	Cho phép Pager hiển thị con số tương ứng với số hoá đơn của khách hàng		
Trigger:	Khách hàng đặt món tại khu ăn uống và hoá đơn được in ra		
Pre-conditions:	<ul style="list-style-type: none"> - Thiết bị có kết nối Internet - Số lượng Pager vừa đủ dùng 		
Post-conditions:	- Pager hiển thị chính xác con số tương ứng với số hoá đơn		
Normal Flow	<ol style="list-style-type: none"> 1. Khi hoá đơn được in cho khách hàng, hệ thống sẽ gửi tín hiệu đến các Pager 2. Hệ thống lọc ra các Pager ở trạng thái chờ 3. Hệ thống chọn duy nhất một Pager, cập nhật số hoá đơn cho Pager 4. Pager hiển thị số hoá đơn tương ứng 5. Hệ thống hiển thị trên màn hình thông báo cho khách hàng sau khi hoá đơn được in xong: “Xin vui lòng lấy đúng Pager trùng với số hoá đơn” 		
Alternative Flows:	<i>Alternative 1: tại bước 3</i> 3a. Nếu không có Pager nào ở trạng thái chờ, hệ thống thông báo “Xin vui lòng cầm hoá đơn đến quầy và chờ nhận món” Kết thúc flow		
Exceptions:	<i>Exception 1: tại bước 1</i> 1a. Nếu không kết nối Internet, hệ thống không gửi được tín hiệu đến các Pager, hệ thống thông báo “Xin vui lòng cầm hoá đơn đến quầy và chờ nhận món”		
Notes and Issues:	Pager hiển thị được số từ 000 đến 999		
Non-functional:	<ul style="list-style-type: none"> - Số phải được hiển thị lên pager trong thời gian không quá 0.5s 		

Use-case ID:	20		
Use-case Name:	Thêm cửa hàng		
Created By:	Đỗ Lê Quang Trung	Last Updated By:	Đỗ Lê Quang Trung
Date Created:	27/04/2020	Date Last Updated:	27/04/2020
Actors:	Quản lý khu ăn uống		
Description:	Cho phép quản lý cập nhật thêm cửa hàng mới		
Trigger:	Quản lý nhấn vào mục “Thêm cửa hàng”		
Pre-conditions:	- Thiết bị có kết nối Internet		
Post-conditions:	- Cửa hàng mới và thông tin về nó được thêm vào hệ thống		
Normal Flow	1. Hiển thị giao diện chứa các thông tin về cửa hàng 2. Quản lý nhập thông tin về cửa hàng 3. Quản lý nhấn nút “Lưu” 4. Hệ thống thông báo “Đã thêm mới cửa hàng”		
Alternative Flows:	3a. Quản lý nhấn nút “Làm lại” 3a1. Các thông tin vừa nhập bị xóa. Use-case quay về bước 2		
Exceptions:	1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không có kết nối Internet” 4a. Quản lý nhập sai hoặc thiếu thông tin 4a1. Hệ thống báo lỗi “Dữ liệu không đúng yêu cầu”. Use-case quay về bước 2 4b. Nếu dữ liệu cửa hàng đã bị trùng 4b1. Hệ thống báo lỗi “Cửa hàng đã tồn tại”. Use-case quay về bước 2		
Notes and Issues:			
Non-functional:	- Thời gian dữ liệu được lưu lại trong hệ thống dưới 1 giây		

Use-case ID:	21		
Use-case Name:	Xoá cửa hàng		
Created By:	Đỗ Lê Quang Trung	Last Updated By:	Đỗ Lê Quang Trung
Date Created:	27/04/2020	Date Last Updated:	27/04/2020
Actors:	Quản lý khu ăn uống		

Description:	Cho phép quản lý xóa cửa hàng khỏi hệ thống
Trigger:	Quản lý nhấn vào mục “Xóa cửa hàng”
Pre-conditions:	- Thiết bị có kết nối Internet - Trong hệ thống có ít nhất 1 cửa hàng
Post-conditions:	- Cửa hàng và thông tin về nó được xóa khỏi hệ thống
Normal Flow	1. Hệ thống hiển thị danh sách các cửa hàng hiện có 2. Quản lý chọn cửa hàng cần xóa 3. Quản lý nhấn nút “Xóa” 4. Hệ thống hiển thị thông báo xác nhận xóa cửa hàng 5. Quản lý nhấn nút “Đồng ý” 6. Hệ thống hiển thị thông báo “Đã xóa”
Alternative Flows:	5a. Quản lý nhấn nút “Từ chối”. Use-case quay về bước 2
Exceptions:	1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không có kết nối Internet” 1b. Nếu không có cửa hàng nào, hệ thống thông báo “Chưa có cửa hàng nào”
Notes and Issues:	
Non-functional:	- Thời gian dữ liệu xóa khỏi trong hệ thống dưới 1 giây

Use-case ID:	22		
Use-case Name:	Xem báo cáo kinh doanh của cả khu ăn uống		
Created By:	Đỗ Lê Quang Trung	Last Updated By:	Đỗ Lê Quang Trung
Date Created:	26/04/2020	Date Last Updated:	26/04/2020
Actors:	Quản lý khu ăn uống		
Description:	Cho phép quản lý đọc các báo cáo về tình hình kinh doanh của các cửa hàng trong khu ăn uống		
Trigger:	Quản lý nhấn vào mục “Xem báo cáo kinh doanh khu ăn uống”		
Pre-conditions:	- Thiết bị có kết nối Internet		
Post-conditions:	- Quản lý biết được tình hình kinh doanh của cửa hàng		
Normal Flow	1. Hiển thị danh sách các báo cáo kinh doanh 2. Quản lý nhấn vào báo cáo muốn xem		

	3. Hiện thị bài báo cáo được chọn 4. Quản lý nhấn nút “Thoát”
Alternative Flows:	4a. Quản lý nhấn nút “Xóa” 4a1. Báo cáo được xóa khỏi hệ thống
Exceptions:	1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không có kết nối Internet” 1b. Nếu không có báo cáo nào, hệ thống thông báo “Hiện giờ không có báo cáo”
Notes and Issues:	
Non-functional:	- Dữ liệu sẽ được lưu trữ trên điện toán đám mây - Các bài báo cáo được sắp xếp theo trình tự thời gian từ gần nhất đến xa nhất

Use-case ID:	23		
Use-case Name:	Xem báo cáo kinh doanh của cửa hàng		
Created By:	Đỗ Lê Quang Trung	Last Updated By:	Đỗ Lê Quang Trung
Date Created:	26/04/2020	Date Last Updated:	26/04/2020
Actors:	Chủ cửa hàng		
Description:	Cho phép chủ cửa hàng đọc các báo cáo về tình hình kinh doanh cửa hàng của mình		
Trigger:	Chủ cửa hàng nhấn vào mục “Xem báo cáo kinh doanh cửa hàng”		
Pre-conditions:	- Thiết bị có kết nối Internet		
Post-conditions:	- Chủ cửa hàng biết được tình hình kinh doanh của cửa hàng		
Normal Flow	1. Hiện thị danh sách các báo cáo kinh doanh 2. Chủ cửa hàng nhấn vào báo cáo muốn xem 3. Hiện thị bài báo cáo được chọn 4. Chủ cửa hàng nhấn nút “Thoát”		
Alternative Flows:	4a. Chủ cửa hàng nhấn nút “Xóa” 4a1. Báo cáo được xóa khỏi hệ thống		
Exceptions:	1a. Nếu không kết nối Internet, hệ thống sẽ thông báo “Không		

	có kết nối Internet” 1b. Nếu không có báo cáo nào, hệ thống thông báo “Hiện giờ không có báo cáo”
Notes and Issues:	
Non-functional:	- Dữ liệu sẽ được lưu trữ trên điện toán đám mây - Các bài báo cáo được sắp xếp theo trình tự thời gian từ gần nhất đến xa nhất

Use-case ID:	24		
Use-case Name:	Bảo trì hệ thống		
Created By:	Đỗ Lê Quang Trung	Last Updated By:	Đỗ Lê Quang Trung
Date Created:	28/04/2020	Date Last Updated:	28/04/2020
Actors:	Nhân viên IT		
Description:	Đóng các dịch vụ trực tuyến để bảo trì hệ thống		
Trigger:	Nhân viên IT nhấn vào mục “Bảo trì hệ thống”		
Pre-conditions:			
Post-conditions:	- Không có lỗi phát sinh trong quá trình vận hành		
Normal Flow	1. Nhân viên IT nhấn nút “Tiến hành bảo trì” 2. Hiện thông báo “Hiện đang bảo trì” lên trang chủ của hệ thống 3. Tắt toàn bộ các dịch vụ trực tuyến 4. Tiến hành các hoạt động bảo trì 5. Nhân viên nhấn nút “Hoàn thành bảo trì” 6. Các dịch vụ trực tuyến được mở lại		
Alternative Flows:			
Exceptions:	3a. Nhân viên IT không tắt các dịch vụ trực tuyến 3a1. Hiện thông báo “Hiện chưa tắt các dịch vụ trực tuyến” Use-case quay lại bước 3		
Notes and Issues:			
Non-functional:	- Thời gian tắt toàn bộ dịch vụ trực tuyến dưới 3 phút - Thời gian mở lại các dịch vụ trực tuyến dưới 5 phút		

3. Other non-interactive functional requirement (bonus)

- Use-case “Đăng nhập” – ID 1:
 - + Thời gian hiển thị màn hình đăng nhập dưới 2 giây.
 - + Thời gian phản hồi của hệ thống dưới 1 giây.
- Use-case “Đăng kí tài khoản” – ID 2:
 - + Tài khoản tạo xong phải được lưu vào hệ thống dưới 3 giây.
- Use-case “Đăng xuất” – ID 3:
 - + Thời gian đăng xuất không quá 2 giây.
- Use-case “Tìm kiếm món ăn” – ID 4:
 - + Thời gian liệt kê danh sách đơn hàng dưới 1 giây.
- Use-case “Thanh toán” – ID 5:
 - + Hỗ trợ nhiều dịch vụ thanh toán online khác nhau.
 - + Bảo mật thông tin thanh toán.
 - + Kết nối với dịch vụ ngoài dưới 3s.
 - + Giao diện gọn gàng, dễ nhìn, tránh bị nhầm lẫn tai hại về tiền của khách hàng.
- Use-case “Nạp thêm tiền” – ID 6:
 - + Tuyệt đối bảo mật thông tin đăng nhập của bên thứ 3, đặc biệt là khi kết nối với tài khoản ngân hàng.
 - + Ẩn (chỉ hiển thị *) khi người dùng đăng nhập trên vendor công cộng, kể cả tên đăng nhập.
- Use-case “Xem lịch sử đơn hàng” – ID 7:
 - + Giao diện dễ nhìn, vừa đủ thông tin, không bị nhầm lẫn.
- Use-case “Đánh giá món ăn” – ID 8:
 - + Giao diện trực quan, dễ nhìn, giúp người dùng đánh giá đúng trọng tâm.

- Use-case “*Nhận hoá đơn*” – ID 9:

- + Thời gian in trong vòng 4s, hoá đơn vừa đủ cầm tay không quá to quá nhỏ.
- + Chữ trên hoá đơn vừa đủ nhìn, không quá nhỏ, ưu tiên dùng các loại font monospace để tránh nhầm lẫn chữ, số, nhưng đồng thời hỗ trợ hiển thị món ăn bằng tiếng Việt.

- Use-case “*Thêm món ăn vào giỏ hàng*” – ID 10:

- + Dữ liệu đơn hàng được lưu lại và tính toán giá tiền nhanh không quá 1 giây để trả lại thông báo cho khách hàng.
- + Tính toán giá tiền chính xác.

- Use-case “*Chỉnh sửa giỏ hàng*” – ID 11:

- + Dữ liệu đơn hàng được lưu lại và tính toán giá tiền nhanh chóng (không quá 0.5 giây để hiển thị giỏ hàng đã cập nhật lại cho khách hàng).
- + Tính toán giá tiền chính xác.

- Use-case “*Thêm món ăn*” – ID 12:

- + Hệ thống phải tạo một ID riêng biệt cho mỗi món ăn được thêm vào.
- + Hệ thống truy xuất và lưu thông tin vào cơ sở dữ liệu trong vòng dưới 2 giây.

- Use-case “*Xóa món ăn*” – ID 13:

- + Hệ thống truy xuất và thay đổi thông tin trong cơ sở dữ liệu trong vòng dưới 2 giây.

- Use-case “*Sửa giá món ăn*” – ID 14:

- + Hệ thống truy xuất và lưu thông tin vào cơ sở dữ liệu trong vòng dưới 2 giây

- Use-case “*Xem đơn hàng*” – ID 15:

- + Thời gian liệt kê danh sách đơn hàng có sẵn dưới 1s.

- Use-case “*Thông báo thời gian nhận món*” – ID 16:

- + Thông báo được gửi đến khách hàng qua app mobile không quá 1s.
- *Use-case “Thông báo món ăn hoàn thành” – ID 17:*
 - + Thông báo được gửi đến khách hàng qua Paper không quá 0.5s.
 - + Thông báo được gửi đến khách hàng qua app mobile không quá 1s.
- *Use-case “Cập nhật món ăn hết hàng” – ID 18:*
 - + Sau khi đầu bếp, chủ cửa hàng hoàn thành bước “Cập nhật món ăn hết hàng”, món ăn phải được làm mờ để khách hàng không chọn được trong thời gian không quá 3s.
- *Use-case “Hiển thị số của đơn hàng” – ID 19:*
 - + Số tương ứng với số hoá đơn của khách hàng phải được hiển thị lên Pager trong thời gian không quá 0.5s.
- *Use-case “Thêm cửa hàng” – ID 20:*
 - + Thời gian dữ liệu được lưu lại trong hệ thống dưới 1 giây.
- *Use-case “Xóa cửa hàng” – ID 21*
 - + Thời gian dữ liệu xóa khỏi trong hệ thống dưới 1 giây.
- *Use-case “Xem báo cáo kinh doanh của cả khu ăn uống” – ID 22:*
 - + Dữ liệu sẽ được lưu trữ trên điện toán đám mây.
 - + Các bài báo cáo được sắp xếp theo trình tự thời gian từ gần nhất đến xa nhất.
- *Use-case “Xem báo cáo kinh doanh của cửa hàng” – ID 23:*
 - + Dữ liệu sẽ được lưu trữ trên điện toán đám mây.
 - + Các bài báo cáo được sắp xếp theo trình tự thời gian từ gần nhất đến xa nhất.
- *Use-case “Bảo trì hệ thống” – ID 24:*
 - + Thời gian tắt toàn bộ dịch vụ trực tuyến dưới 3 phút.
 - + Thời gian mở lại các dịch vụ trực tuyến dưới 5 phút.

II. Non-functional requirements

1. General non-functional requirements for the whole system

Toàn hệ thống có những yêu cầu phi chức năng như sau:

- Usability requirement
- Installability requirement
- Performance requirement
- Environmental requirement
- Security requirement
- Space requirement
- Scalability requirement

2. Other non-functional requirements related to the features that the student is in-charged

- “*Usability Requirement*” – *Yêu cầu về khả năng dễ sử dụng:*

+ Khách hàng có thể làm quen với app mobile, thực hiện được những thao tác trên app sau khoảng thời gian sử dụng 15 phút.

+ Đầu bếp có thể thực hiện được mọi hoạt động liên quan đến đơn hàng (xem, thông báo, cập nhật món,...) sau khoảng 30 phút tiếp xúc với phần mềm.

+ Chủ cửa hàng, quản lý khu ăn uống có thể hiểu và sử dụng được những chức năng quản lý cửa hàng của mình/khu ăn uống sau 1 tiếng training.

- “*Installability Requirement*” – *Yêu cầu về khả năng cài đặt:*

+ Việc cài đặt hệ thống phải được thực hiện bởi đối tác phát triển phần mềm.

- “*Performance requirement*” – *Yêu cầu về hiệu suất:*

+ Độ trễ phản hồi thấp $< 0.5s$.

+ Hệ thống có thể truy xuất và thay đổi dữ liệu trên cơ sở dữ liệu trong vòng 5s hoặc ít hơn.

+ Tài nguyên lưu trữ chiếm dụng của hệ thống trong trạng thái hoạt động bình thường không được phép lớn hơn 80% tài nguyên lưu trữ được phép sử dụng hoặc 20% tài nguyên lưu trữ dùng chung tại mọi thời điểm.

- “*Environmental requirement*” - *Yêu cầu về hệ điều hành và phần cứng của app:*

+ App dành cho khách hàng:

* Hệ điều hành:

- Android 5.1 trở lên.
- Android Open Source Project/Android-x86 (5.1 trở lên) trên các máy màn hình cảm ứng công cộng.
- Khi quy mô Food Court được mở rộng trên 20 cửa hàng thì sẽ port phiên bản dành cho iOS.

* Phần cứng:

- 512MB RAM (khuyến khích 1G RAM)
- 128MB dung lượng trống (khuyến khích 256MB để lưu cache)

+ Web App dành cho nhân viên/đầu bếp/quản lý:

* Hệ điều hành:

- Blink Engine browsers (khuyến cáo Google Chrome/Chromium 60 trở lên)
- Mozilla Quantum Engine browsers (khuyến cáo Firefox 57 trở lên)
- Safari 10 trở lên

* Phần cứng:

- 512MB RAM (khuyến khích 2G RAM)
- 256MB dung lượng ổ đĩa trống

- “*Security requirement*” – *Yêu cầu về bảo mật:*

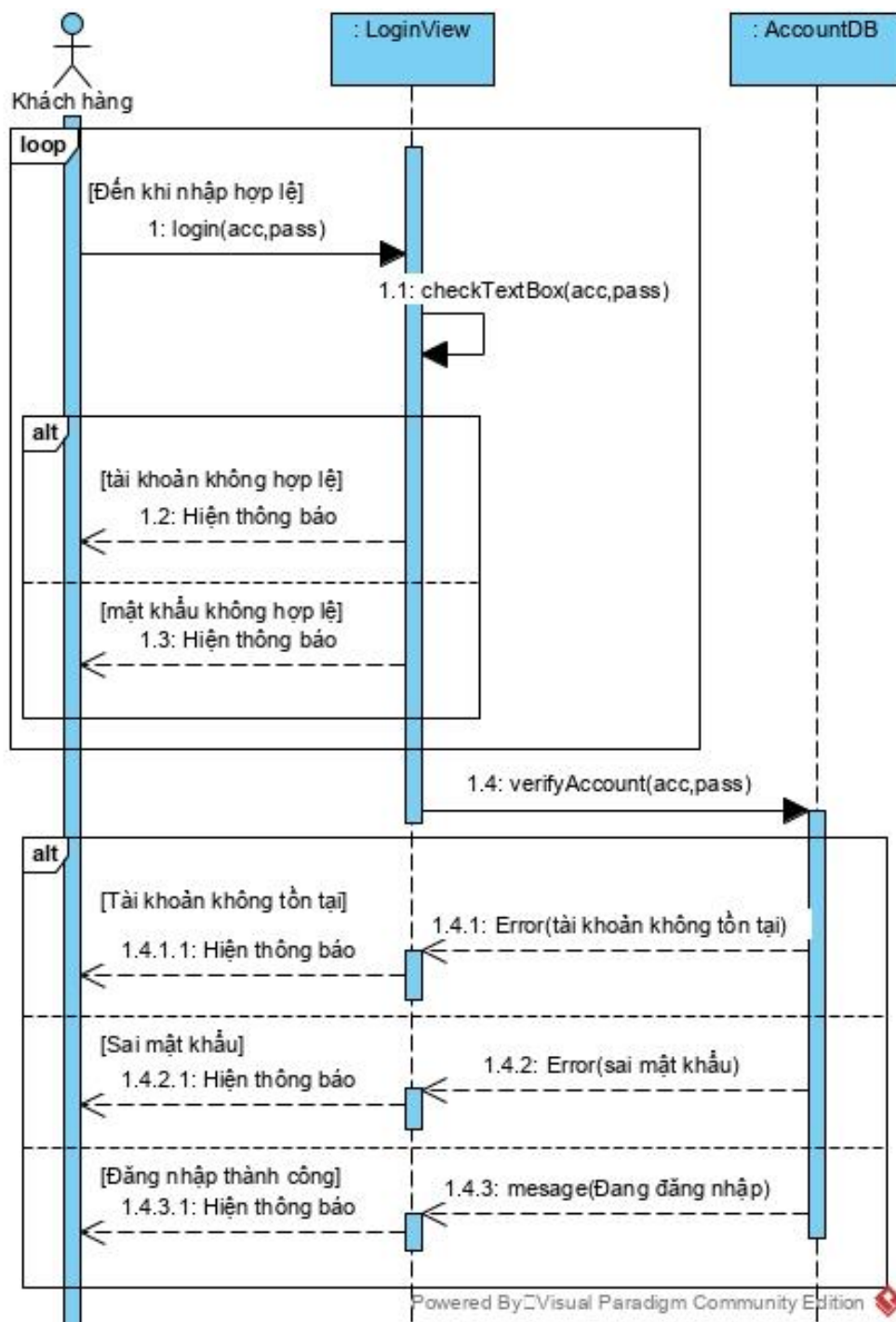
- + Người dùng muốn sử dụng ứng dụng bắt buộc phải đăng nhập tài khoản
- + Người mua hàng cần bấm mã pin thiết bị trước 2 phút khi xác nhận thanh toán.
- + Hệ thống luôn kiểm tra kết nối internet của thiết bị mỗi 5 giây.

- “*Space requirement*” – *Yêu cầu về không gian*:
 - + Hệ thống có thể đáp ứng tối đa lên đến 1000 người dùng tại thời gian cao điểm (giờ ăn trưa từ 11h đến 13h, giờ ăn tối từ 17h – 20h) và tối đa 500 người tại thời điểm khác.
- “*Scalability requirement*” – *Yêu cầu về không gian*:
 - + Dễ dàng thêm tính năng mới cho hệ thống.
 - + Do nhu cầu về việc mở rộng kinh doanh sản xuất, hệ thống cần phải đáp ứng nhu cầu tăng thêm khoảng 300 khách hàng truy cập vào hệ thống mỗi năm.

PART 2

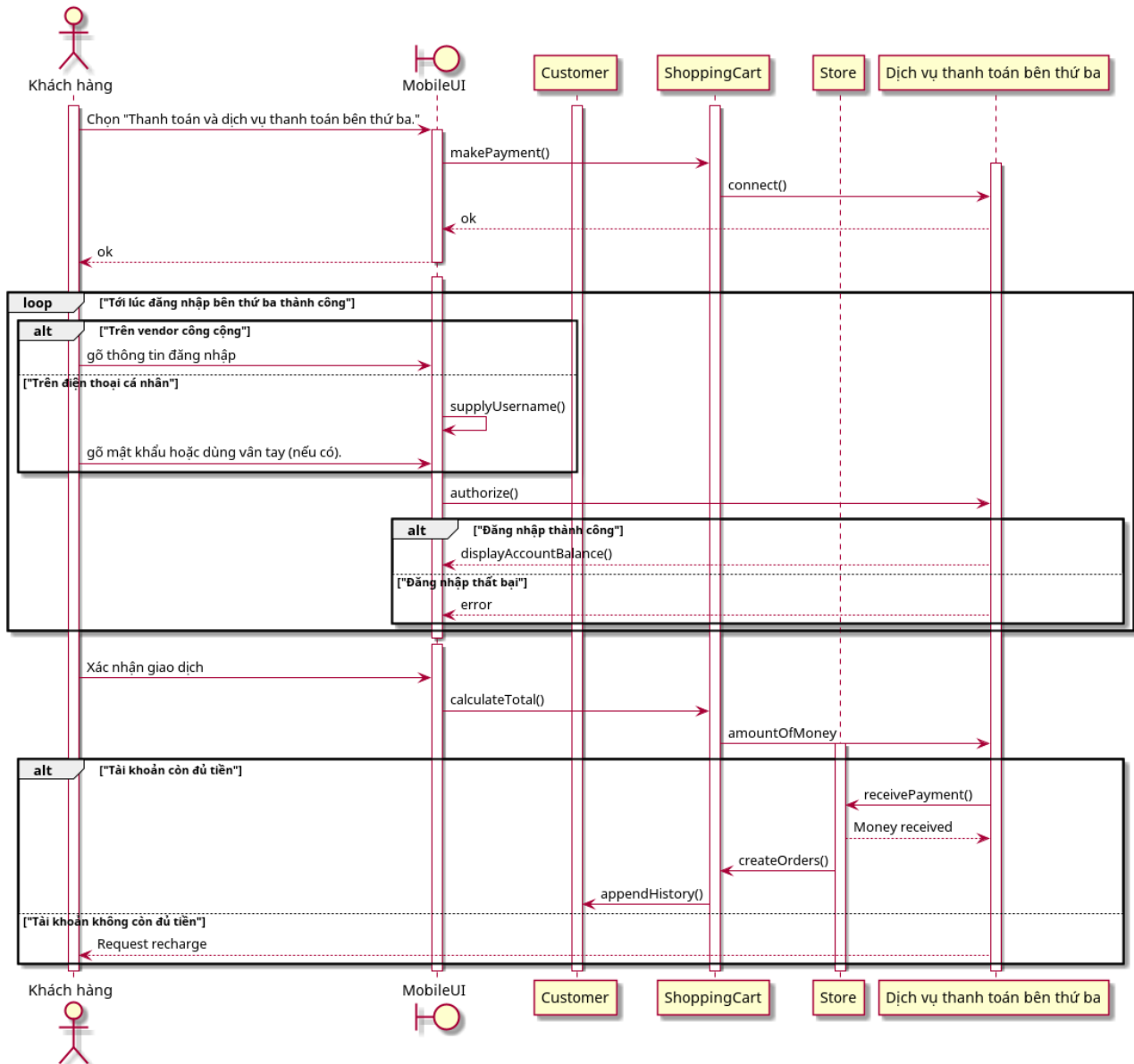
I. Sequence diagram

* Chức năng “Xử lý tác vụ đăng nhập của khách hàng”:



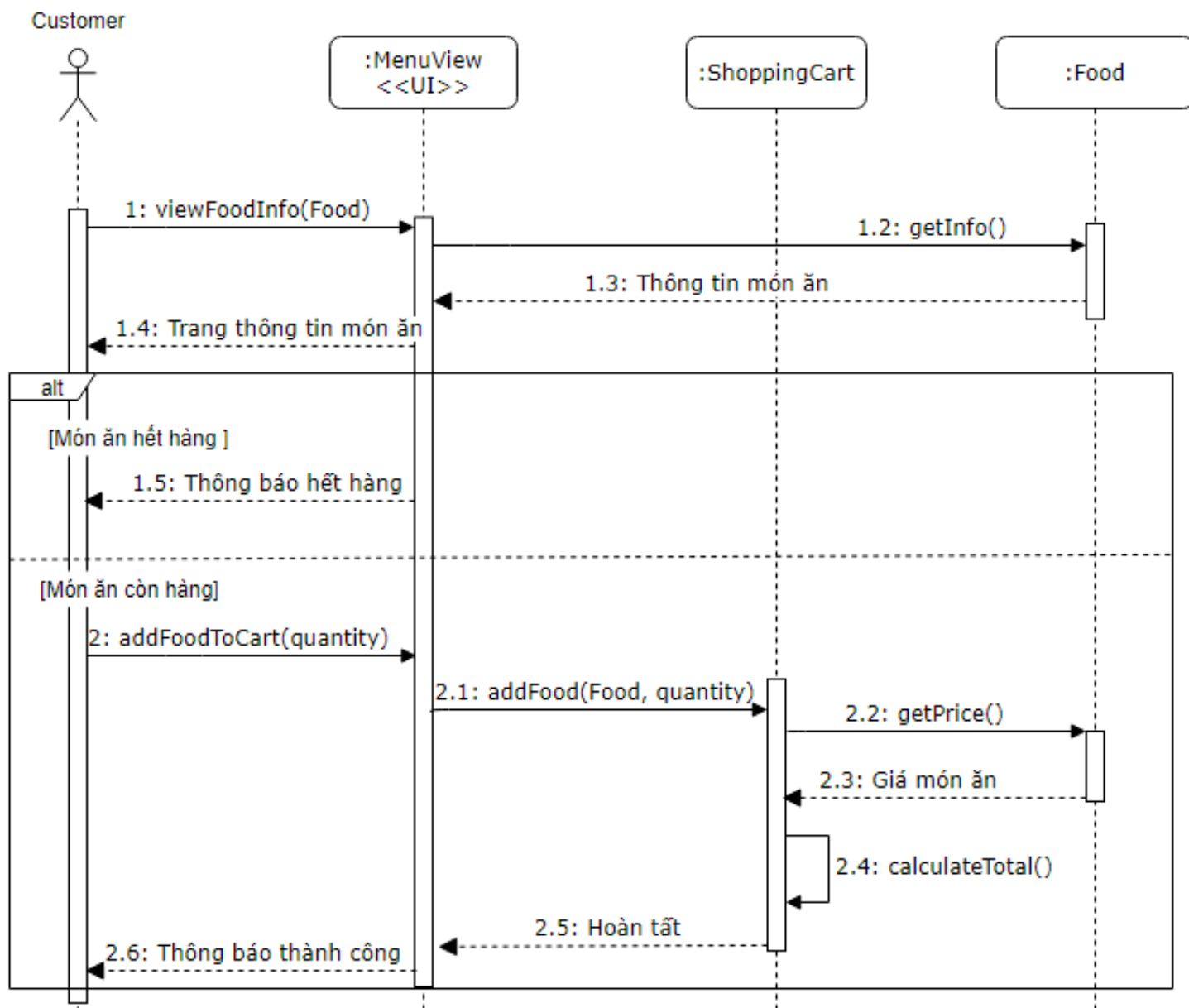
Hình 2: Sequence diagram cho chức năng "Xử lý tác vụ đăng nhập của khách hàng"

*** Chức năng “Thanh toán”:**



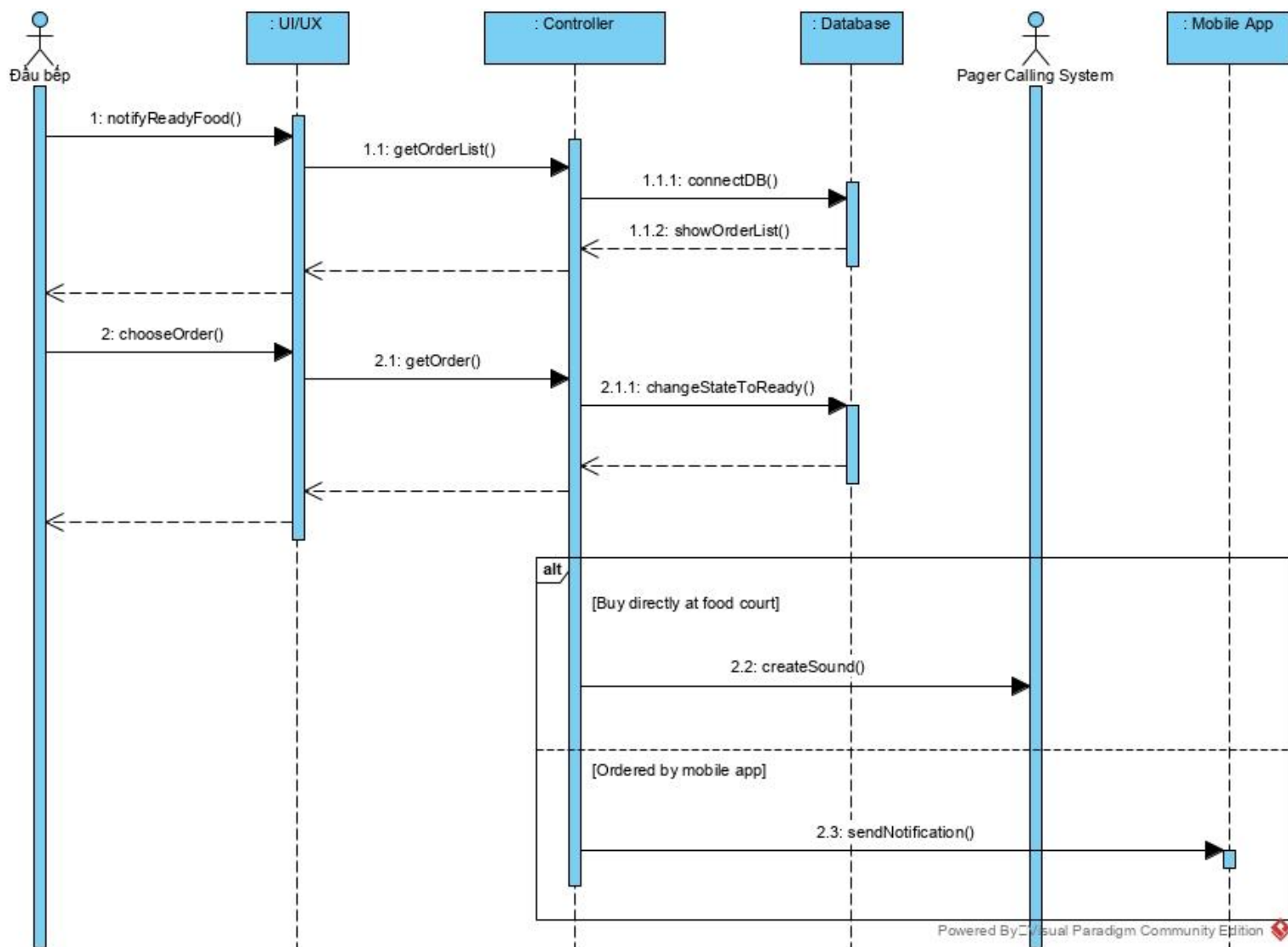
Hình 3: Sequence diagram cho chức năng "Thanh toán"

*** Chức năng “Thêm món ăn vào giỏ hàng”:**



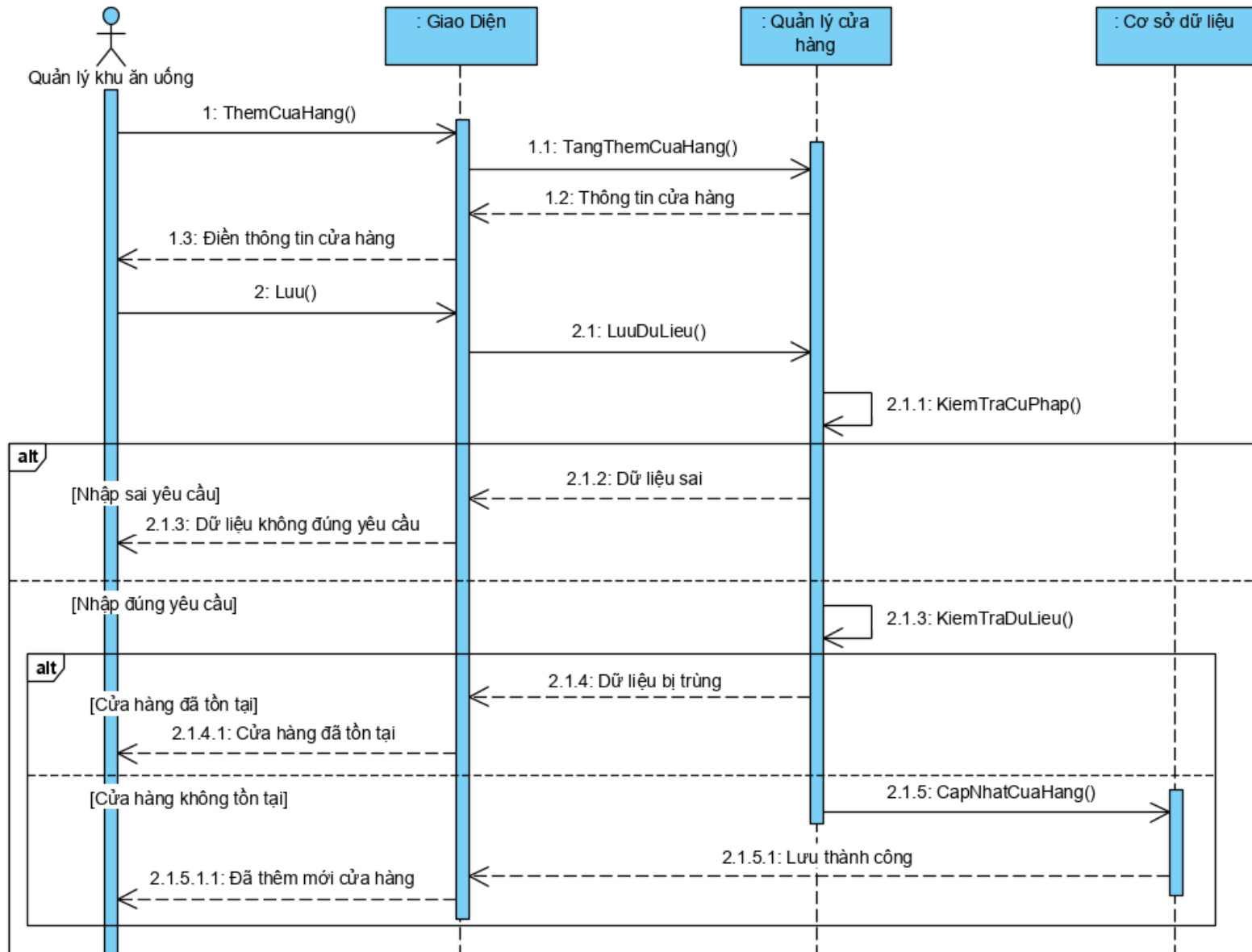
Hình 4: Sequence diagram cho chức năng "Thêm món ăn vào giỏ hàng"

* Chức năng “Thông báo món ăn hoàn thành”



Hình 5: Sequence diagram cho chức năng "Thông báo món ăn hoàn thành"

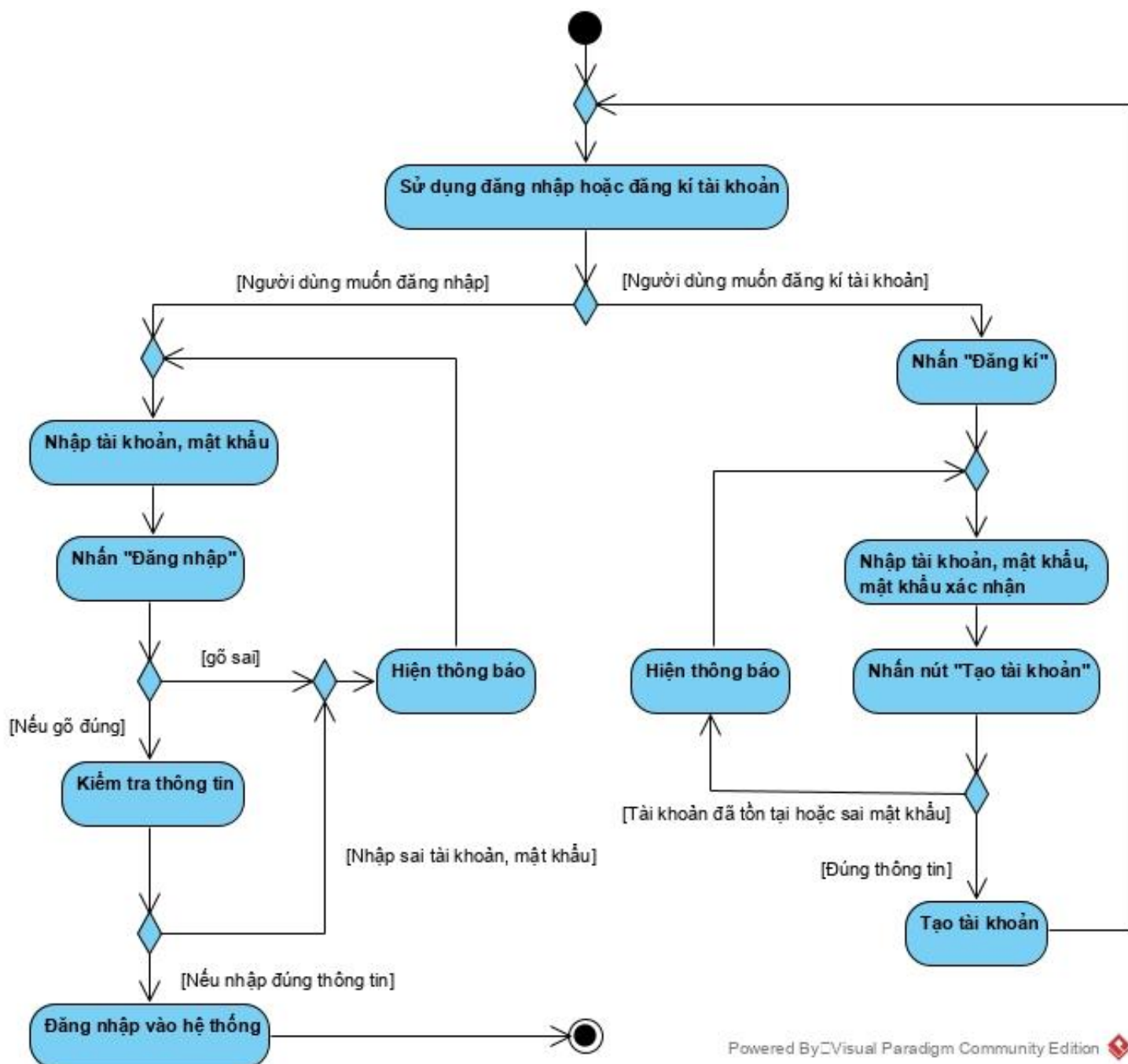
*** Chức năng “Thêm cửa hàng”:**



Hình 6: Sequence diagram cho chức năng "Thêm cửa hàng"

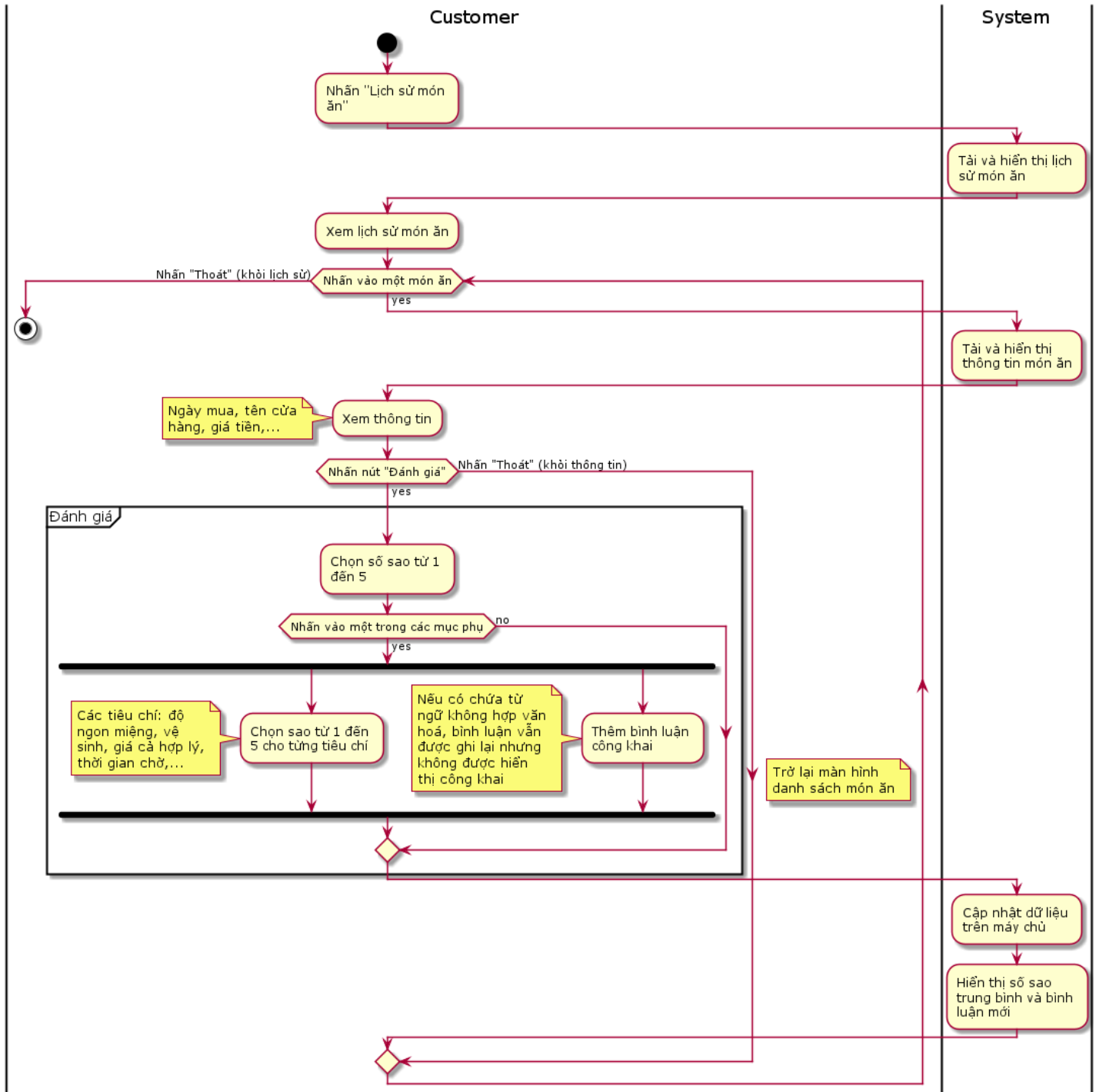
II. Activity diagram

* Chức năng “Xử lý tác vụ đăng nhập của khách hàng”:



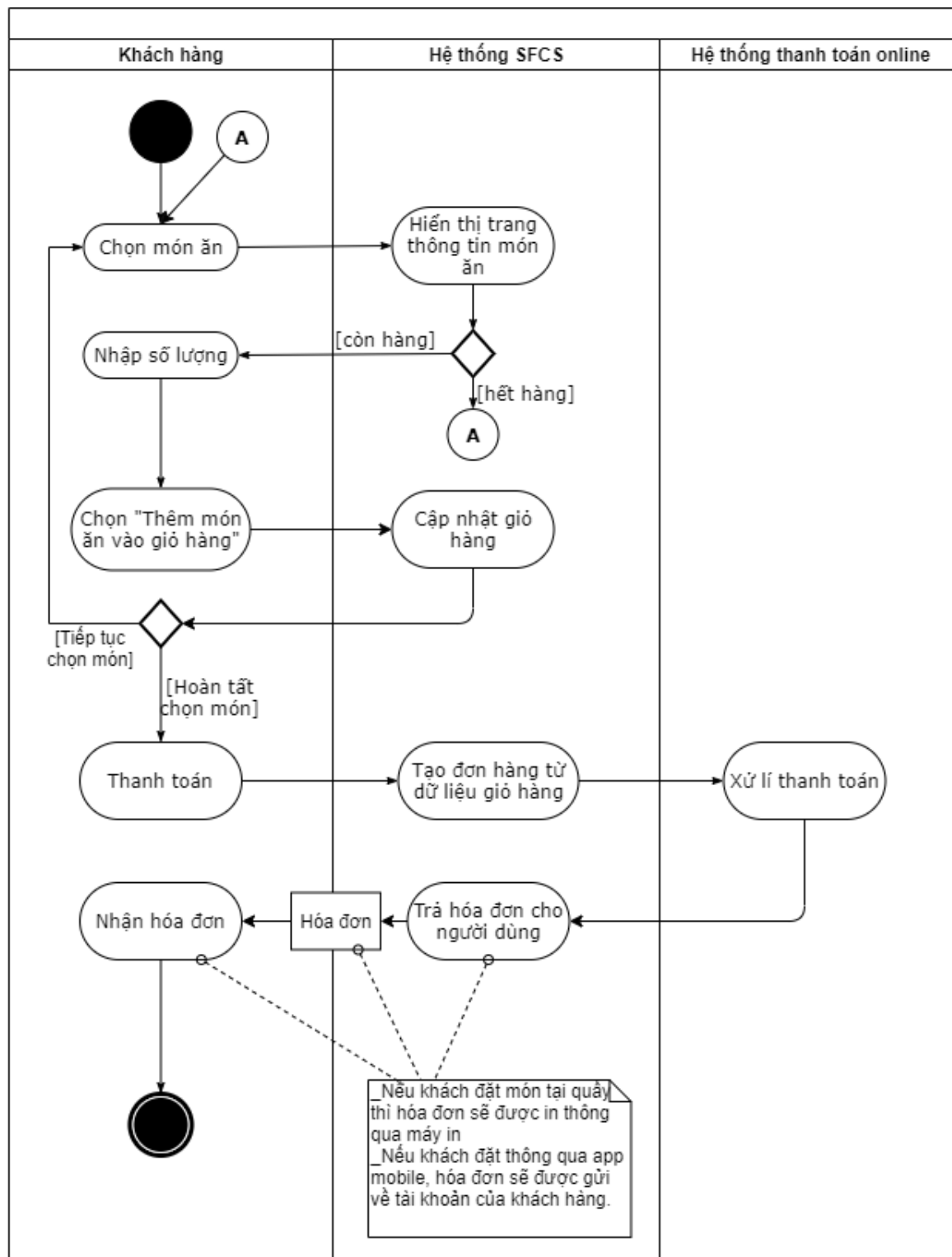
Hình 7: Activity diagram cho chức năng "Xử lý tác vụ đăng nhập của khách hàng"

*** Chức năng “Xem lịch sử và rate”:**



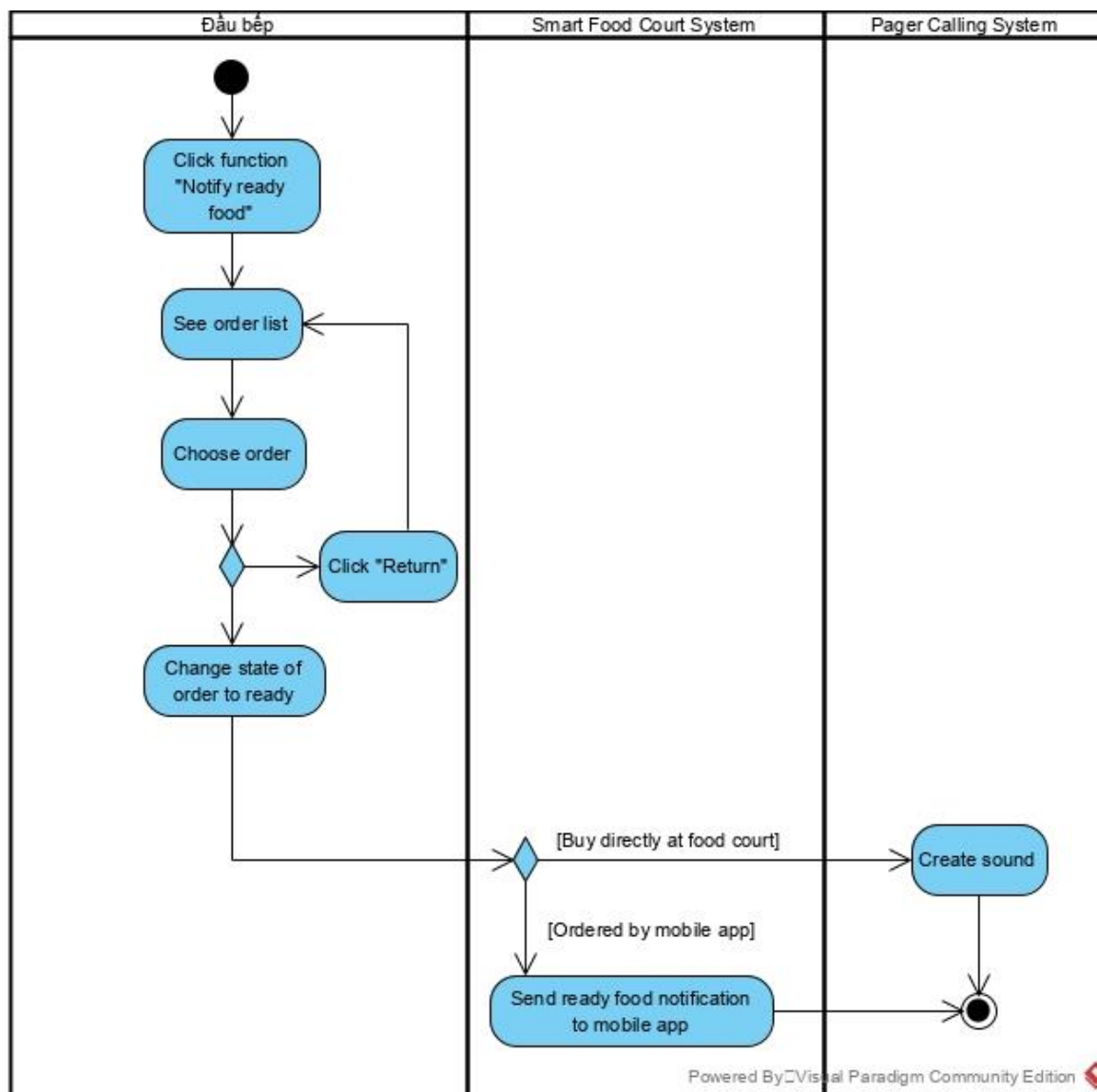
Hình 8: Activity diagram cho chức năng "Xem lịch sử và rate"

*** Chức năng “Thêm món ăn vào giỏ hàng”:**



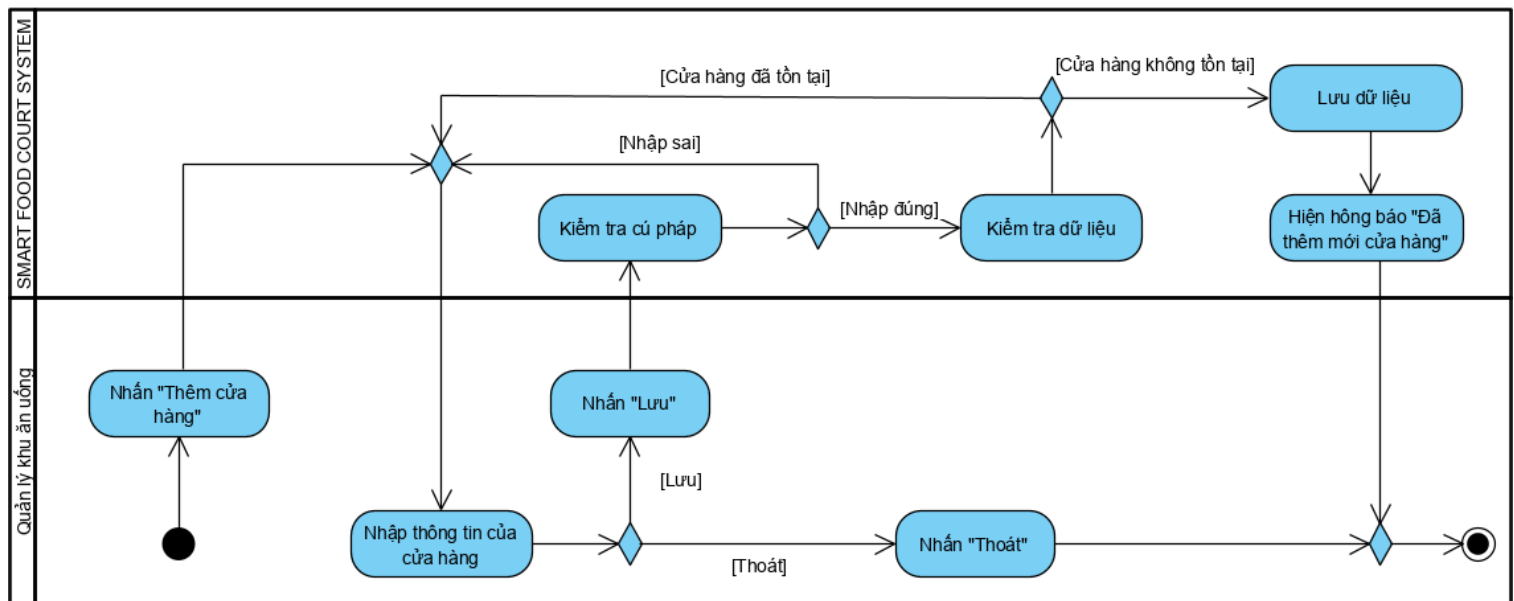
Hình 9: Activity diagram cho chức năng "Thêm món ăn vào giỏ hàng"

*** Chức năng “Thông báo món ăn hoàn thành”**



Hình 10: Activity diagram cho chức năng "Thông báo món ăn hoàn thành"

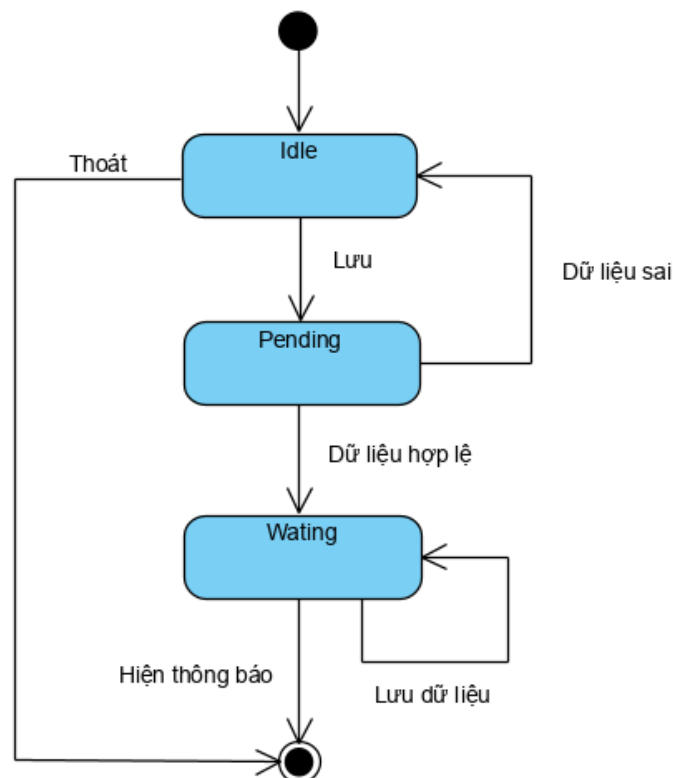
*** Chức năng “Thêm cửa hàng”:**



Hình 11: Activity diagram cho chức năng "Thêm cửa hàng"

III. State Diagram (Bonus)

*** Chức năng “Thêm cửa hàng”:**



Hình 12: State diagram cho chức năng "Thêm cửa hàng"

State tabular:

State	Description
Idle	Hệ thống chờ quản lý nhập thông tin
Pending	Hệ thống đang kiểm tra dữ liệu đầu vào
Waiting	Hệ thống đang lưu thông tin vào cơ sở dữ liệu

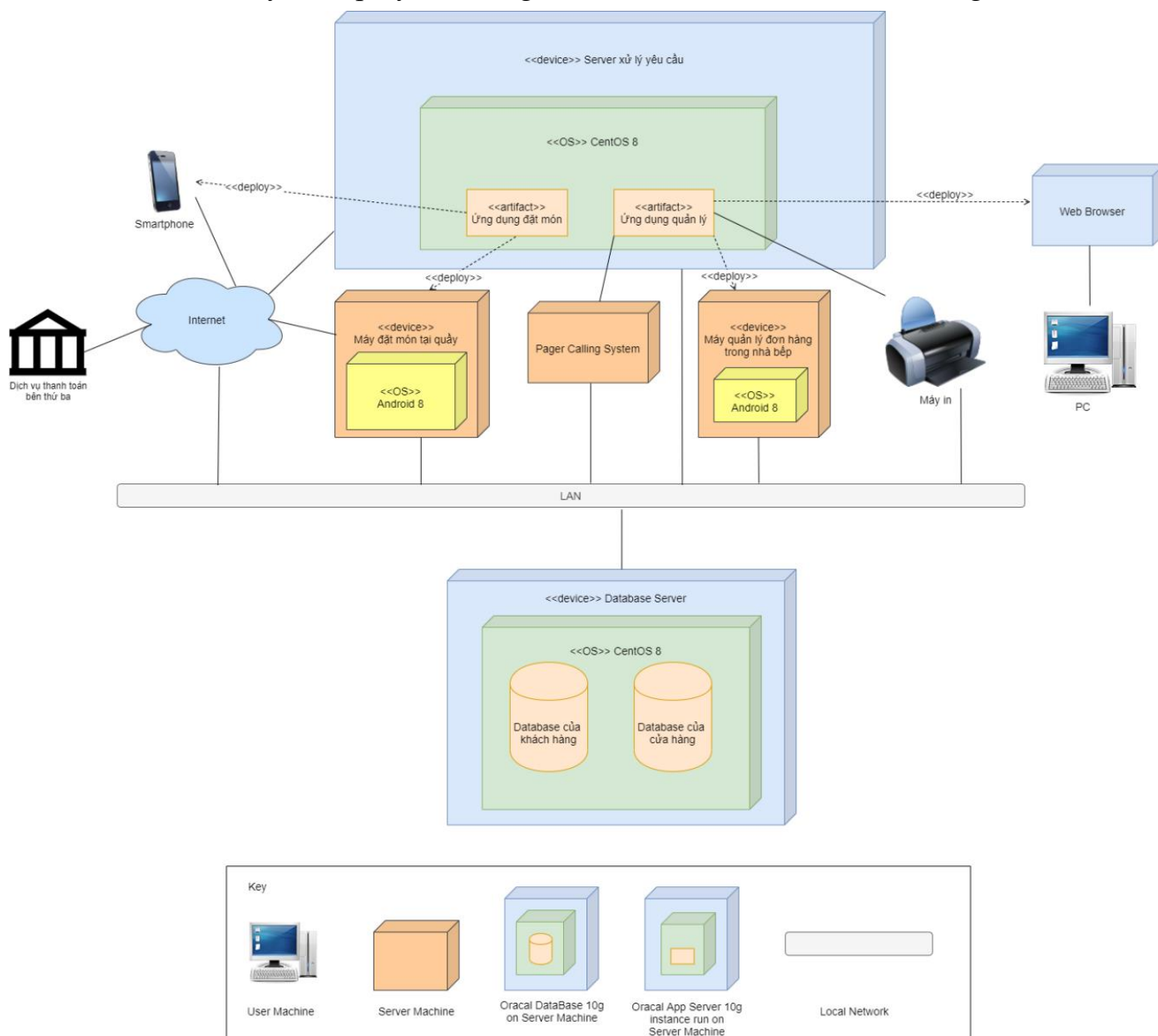
Stimulus tabular:

Stimulus	Description
Lưu	Quản lý xác nhận lưu thông tin
Thoát	Quản lý dừng việc thêm cửa hàng
Dữ liệu sai	Quản lý nhập sai thông tin
Dữ liệu hợp lệ	Quản lý nhập đúng thông tin
Lưu dữ liệu	Hệ thống lưu thông tin vào cơ sở dữ liệu
Hiện thông báo	Hệ thống hiện thông báo “Đã thêm mới cửa hàng”

PART 3

I. Deployment View (Group work)

- Deployment View được biểu diễn bằng Deployment diagram.
- Dưới đây là Deployment diagram nhóm làm cho toàn bộ hệ thống:

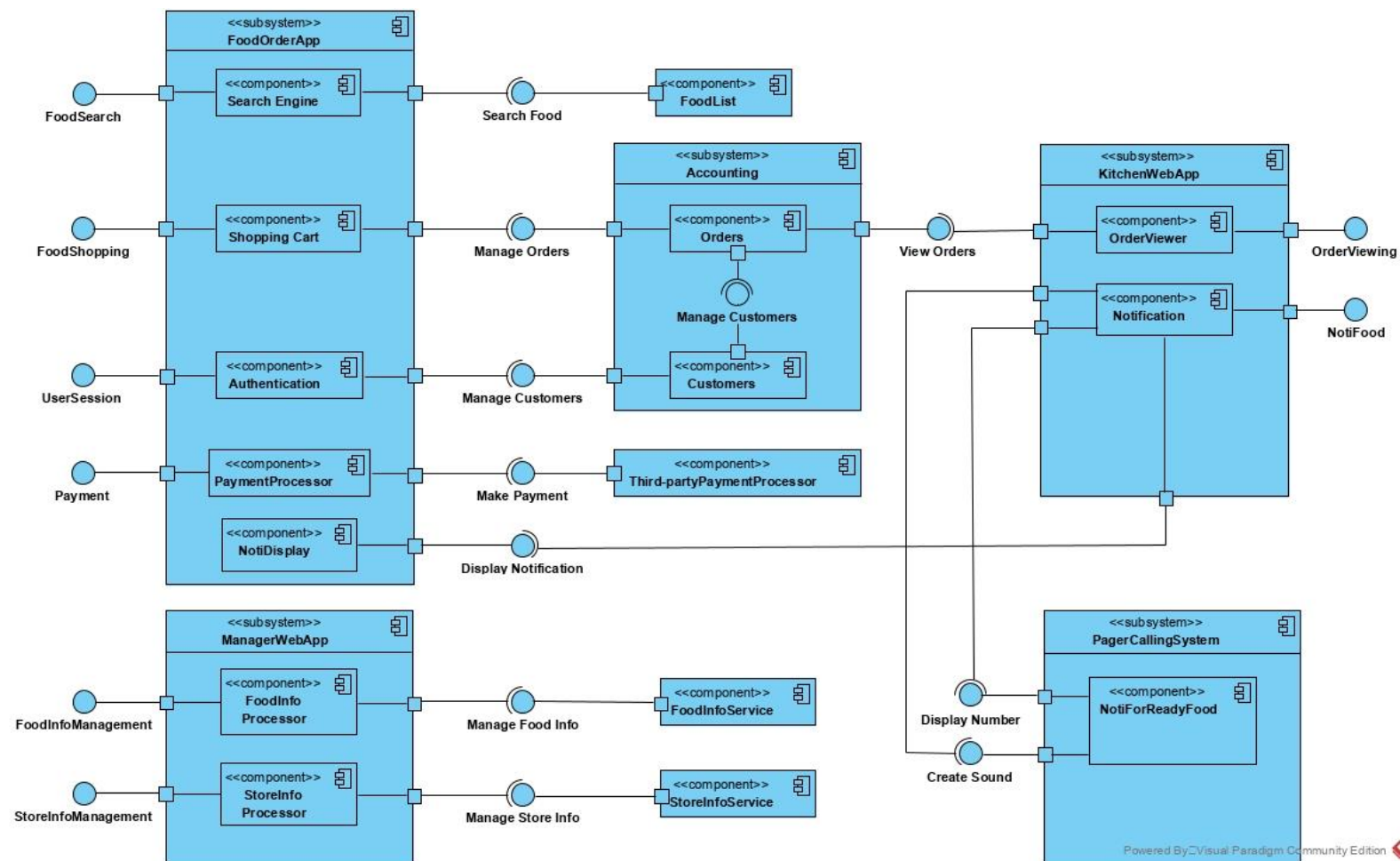


Hình 13: Deployment diagram cho toàn bộ hệ thống

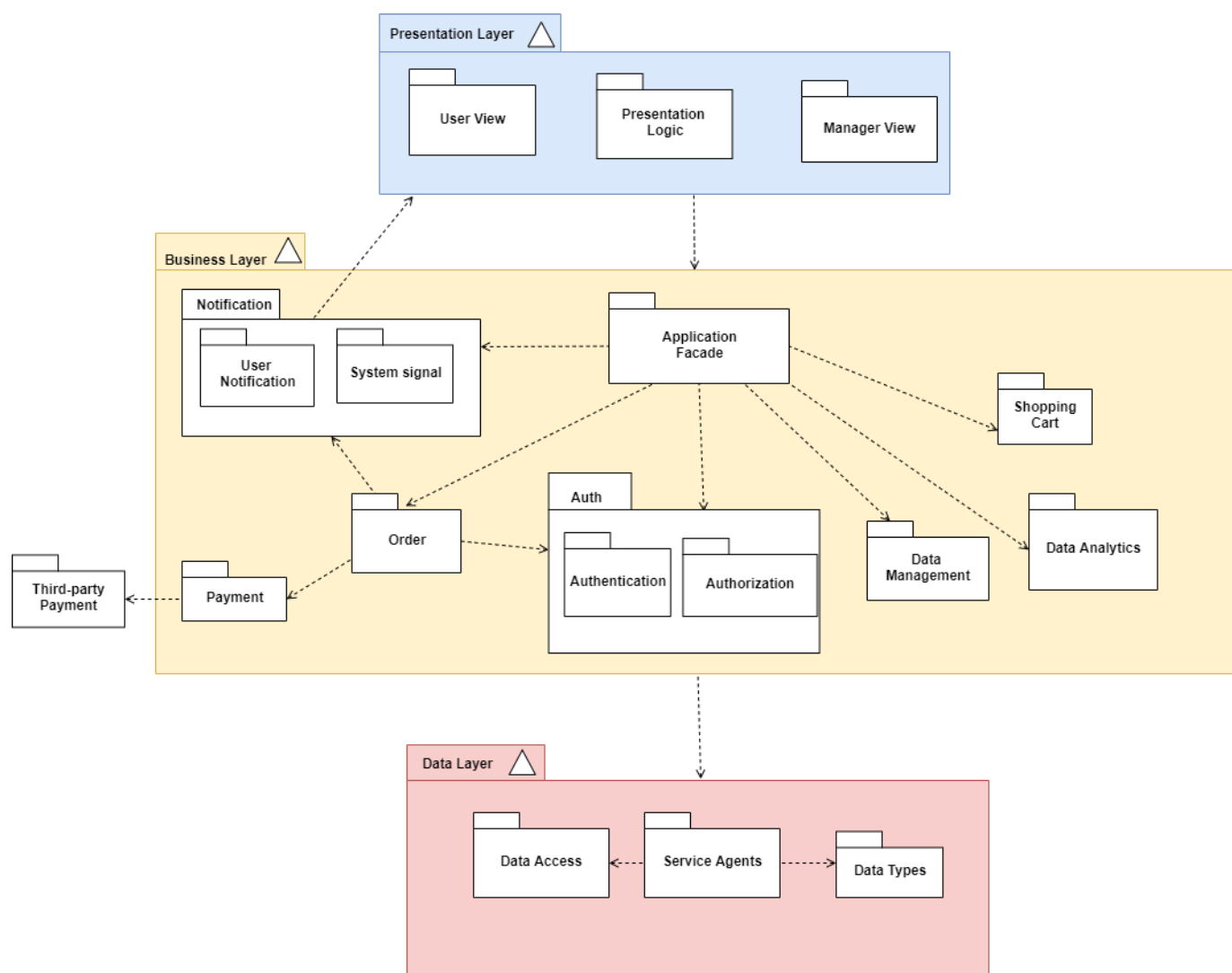
II. Development/Implementation View (Individual work)

- Development/Implementation View được biểu diễn bằng: Component diagram, Package diagram.

- Dưới đây là Component diagram/ Package diagram của mỗi cá nhân trong nhóm làm cho toàn bộ hệ thống:

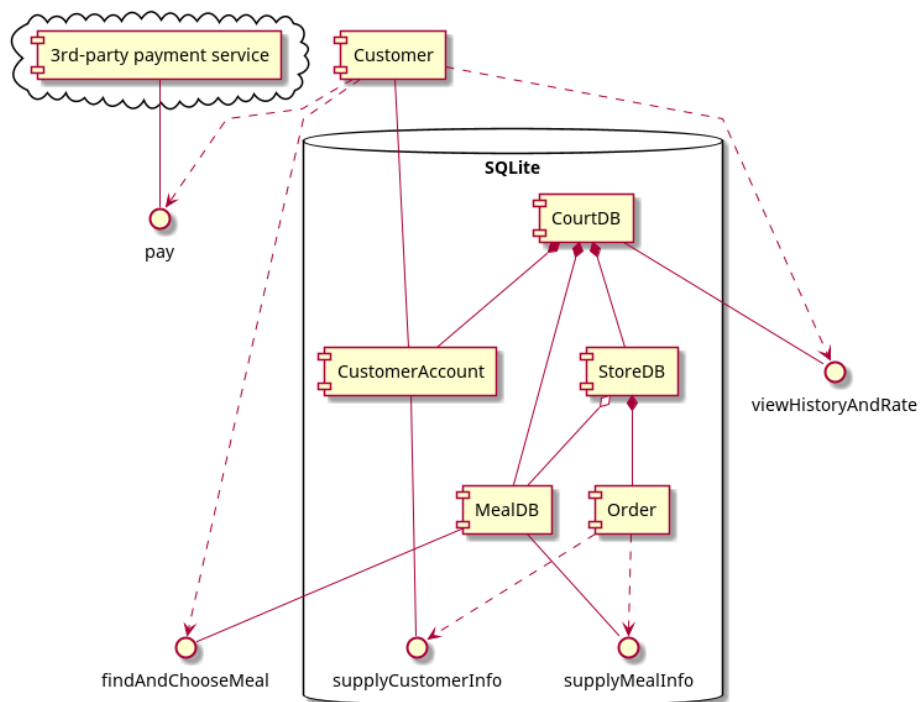


Hình 14: Component diagram của SV Lê Đức Huy – 1810166

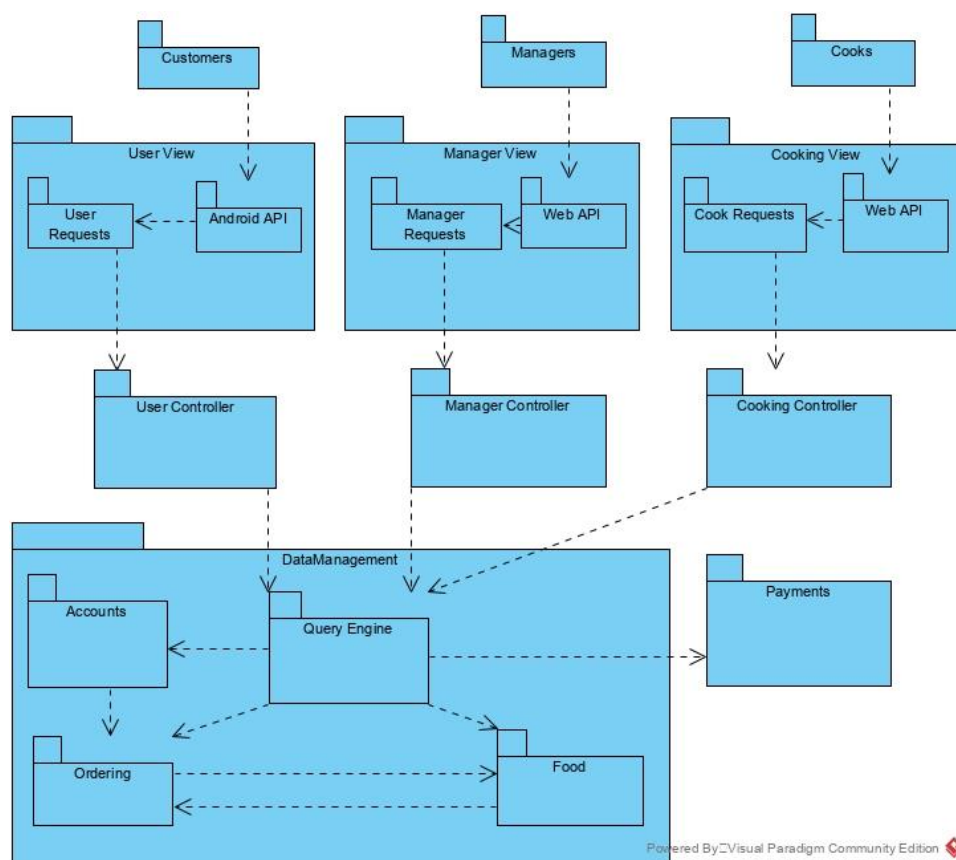


Hình 15: Package diagram của SV Nguyễn Gia Huy - 1810173

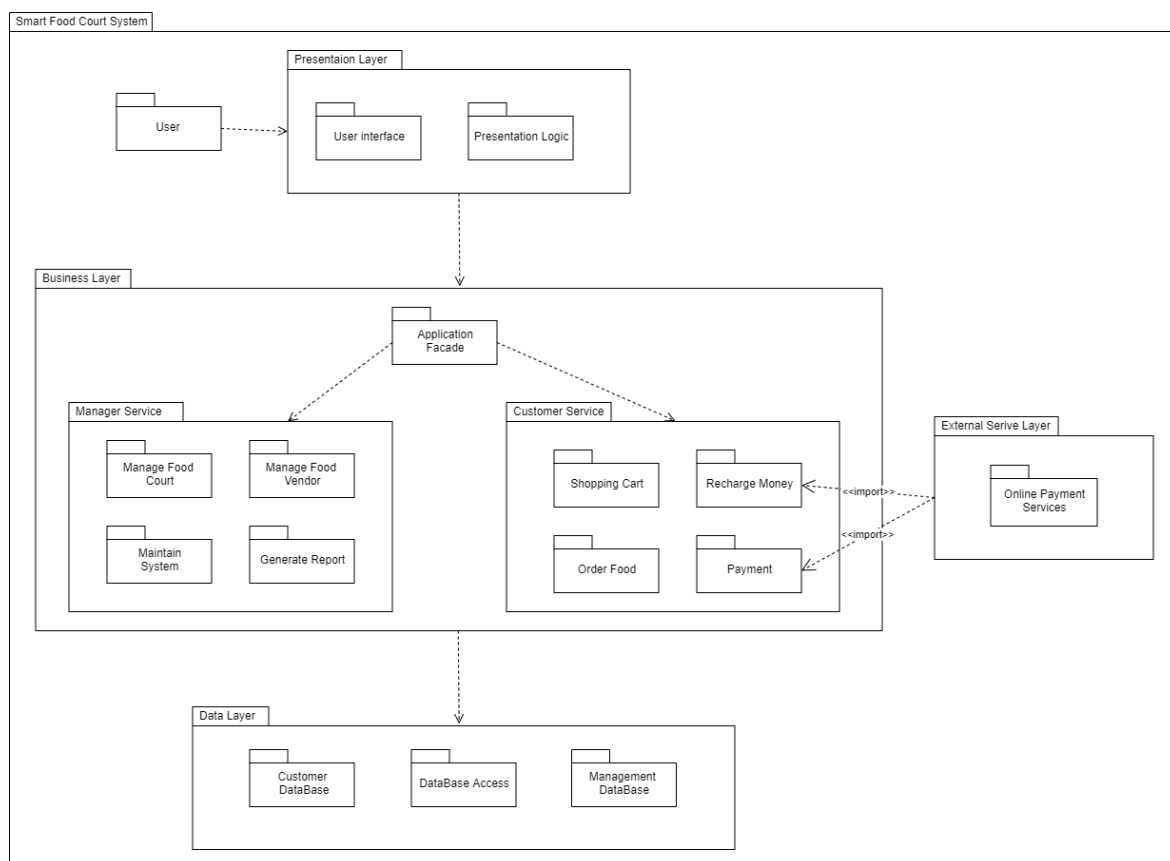
- ❖ **Presentation Layer** : Tầng layer về giao diện của người dùng bao gồm giao diện cho khách hàng và cho quản lí
- ❖ **Business Layer** : Layer xử lí các yêu cầu của người dùng và di chuyển, xử lí dữ liệu giữa các tầng xung quanh. Giải thích một số package :
 - **Data Analytics** : Package chứa những package, class phân tích số liệu cũng như vẽ các biểu đồ về báo cáo kinh doanh của các cửa hàng
 - **Notification Management**: Package quản lí các thông báo gửi tới người dùng cũng như các signal điều khiển trong hệ thống
 - **Auth** : Package phục vụ việc xác thực danh tính cũng như quản lí quyền của tài khoản
- ❖ **Data Layer** : Layer lưu trữ dữ liệu của toàn hệ thống



Hình 16: Package diagram của SV Lê Trung Đan – 1810887



Hình 17: Package diagram của SV Hoàng Tấn Phát - 1811137



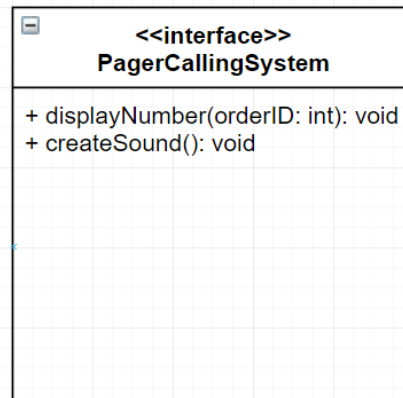
Hình 18: Package diagram của SV Đỗ Lê Quang Trung - 1811304

PART 4

I. Module Interface

Ta sử dụng các Module Interface như:

* **PagerCallingSystem:**

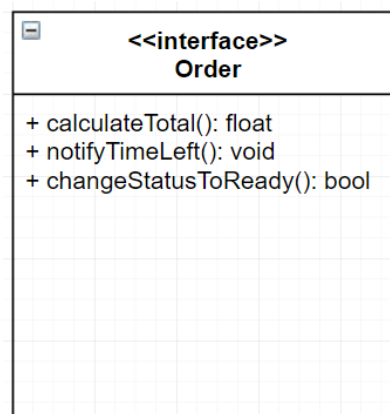


Hình 19: Interface PagerCallingSystem

Interface này sẽ cung cấp hàm:

- void displayNumber(orderID: int) để hiển thị số của đơn hàng lên màn hình hiển thị của Pager.
- void createSound() để yêu cầu Pager phát ra âm thanh thông báo cho khách hàng biết khi món ăn đã hoàn thành để họ tới quầy nhận món.

* **Order:**

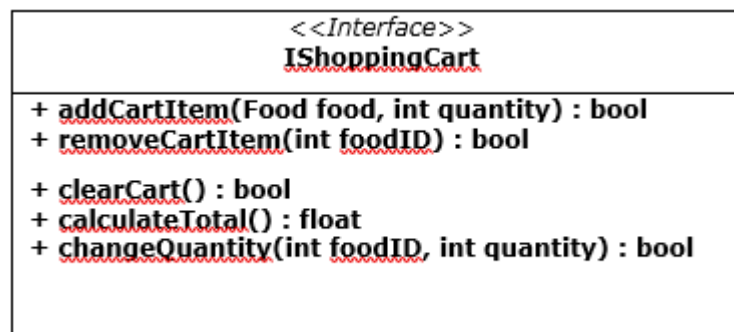


Hình 20: Interface Order

Interface này sẽ cung cấp hàm:

- float calculateTotal() để tính tổng số tiền mà khách hàng cần phải trả cho đơn hàng.
- void notifyTimeLeft(): thông báo thời gian dự định hoàn thành của đơn hàng cho khách hàng biết sau khi họ đặt hàng và thanh toán thành công.
- bool changeStatusToReady(): chuyển trạng thái của đơn hàng thành “Hoàn thành” khi món ăn hoàn tất.

* Shopping Cart:



Hình 21: Interface IShoppingCart

Thực hiện chức năng căn cơ của một giỏ hàng (thêm/xóa hoặc thay đổi số lượng của vật phẩm trong giỏ hàng)

❖ addItem(Food food, int quantity): bool

Phương thức thêm một món ăn với số lượng cho sẵn vào giỏ hàng

❖ removeCartItem(int foodID): bool

Phương thức bỏ một món ăn ra khỏi giỏ hàng

❖ clearCart(): bool

Phương thức bỏ chọn toàn bộ các món đang có trong giỏ hàng

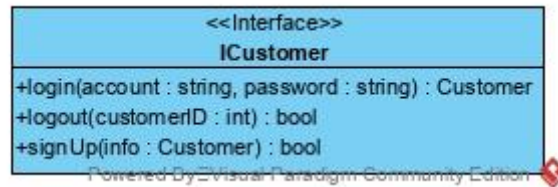
❖ calculateTotal(): float

Tính toán tổng giá tiền của giỏ hàng

❖ changeQuantity(int foodID, int quantity): bool

Phương thức thay đổi số lượng của một món ăn đã có trong giỏ hàng

*** ICustomer:**

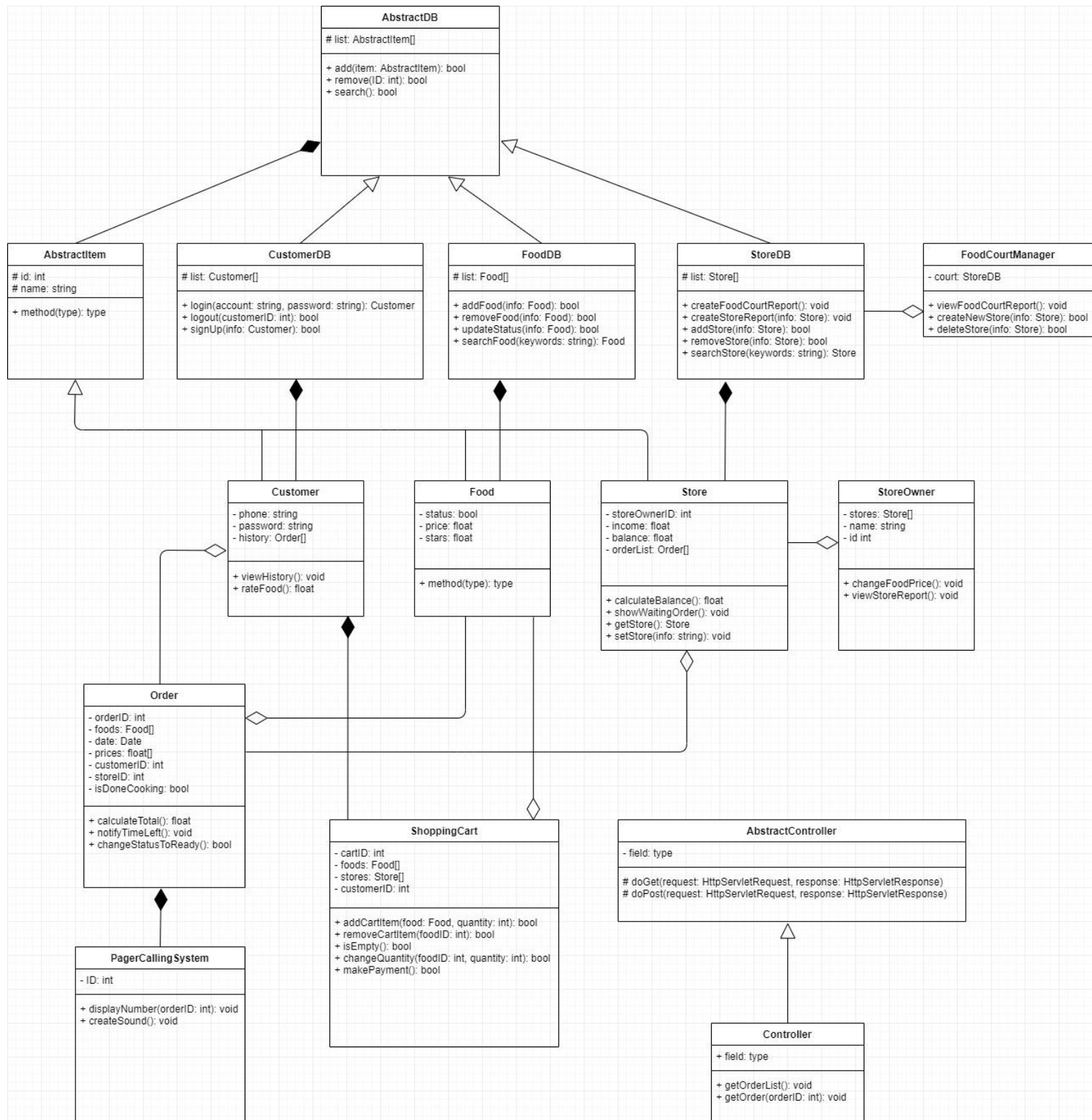


Hình 22: Interface ICustomer

- Hàm đầu tiên thực hiện chức năng “đăng nhập” vào hệ thống của khách hàng.
- Hàm thứ hai thực hiện chức năng “đăng xuất” của khách hàng
- Hàm thứ ba thực hiện chức năng “đăng kí tài khoản”

II. Class Diagram

Class diagram cho toàn bộ hệ thống:



Hình 23: Class diagram cho toàn bộ hệ thống

III. Method Descriptions

* Class AbstractDB:

Tên method	Chức năng
add(item: AbstractItem): bool	- Đối số truyền vào là biến item thuộc kiểu AbstractItem, kiểu trả về của hàm là bool - Dùng để thêm mới một mục
remove(ID: int): bool	- Đối số truyền vào là biến ID kiểu int, kiểu trả về của hàm là bool - Dùng để xoá một mục
search(): bool	- Kiểu trả về của hàm là bool - Dùng để tìm kiếm dữ liệu

* Class AbstractController:

Tên method	Chức năng
# doGet(request: HttpServletRequest, response: HttpServletResponse)	Dùng để điều khiển các chức năng của chương trình
# doPost(request: HttpServletRequest, response: HttpServletResponse)	

* Class Controller:

Tên method	Chức năng
getOrderList(): void	- Kiểu trả về của hàm là void - Dùng để ra lệnh lấy danh sách đơn hàng của cửa hàng
getOrder(orderID: int): void	- Kiểu trả về của hàm là void - Dùng để ra lệnh lấy đơn hàng cụ thể của cửa hàng

*** Class CustomerDB:**

Tên method	Chức năng
login(account: string, password: string): Customer	- Đối số truyền vào là biến account và password thuộc kiểu string, kiểu trả về của hàm là kiểu Customer - Dùng để đăng nhập vào hệ thống
logout(customerID: int): bool	- Đối số truyền vào là biến customer ID kiểu int, kiểu trả về của hàm là bool - Dùng để đăng xuất khỏi hệ thống
signUp(info: Customer): bool	- Đối số truyền vào là biến info kiểu Customer, kiểu trả về của hàm là bool - Dùng để đăng ký tài khoản để sử dụng hệ thống

*** Class FoodDB:**

Tên method	Chức năng
addFood(info: Food): bool	- Đối số truyền vào là biến info kiểu Food, kiểu trả về của hàm là kiểu bool - Dùng để thêm món ăn vào menu
removeFood(info: Food): bool	- Đối số truyền vào là biến info kiểu Food, kiểu trả về của hàm là kiểu bool - Dùng để xóa món ăn khỏi menu
updateStatus(info: Food): bool	- Đối số truyền vào là biến info kiểu Food, kiểu trả về của hàm là kiểu bool - Dùng để cập nhật tình trạng của món ăn (còn hàng hay hết hàng)
searchFood(keywords: string): Food	- Đối số truyền vào là biến keywords kiểu string, kiểu trả về của hàm là kiểu Food - Dùng để tìm kiếm món ăn theo từ khoá nhập vào

*** Class StoreDB:**

Tên method	Chức năng
createFoodCourtReport(): void	- Kiểu trả về của hàm là kiểu void - Dùng để tạo ra bản báo cáo của cả khu ăn uống
createStoreReport(info: Store): void	- Đối số truyền vào là biến info kiểu Store, kiểu trả về của hàm là kiểu void - Dùng để tạo ra bản báo cáo của cửa hàng
addStore(info: Store): bool	- Đối số truyền vào là biến info kiểu Store, kiểu trả về của hàm là kiểu bool - Dùng để thêm mới cửa hàng
removeStore(info: Store): bool	- Đối số truyền vào là biến info kiểu Store, kiểu trả về của hàm là kiểu bool - Dùng để xóa cửa hàng
searchStore(keywords: string): Store	- Đối số truyền vào là biến keywords kiểu string, kiểu trả về của hàm là kiểu Store - Dùng để tìm kiếm cửa hàng theo từ khoá nhập vào

*** Class FoodCourtManager:**

Tên method	Chức năng
viewFoodCourtReport(): void	- Kiểu trả về của hàm là kiểu void - Dùng để xem bản báo cáo của cả khu ăn uống
createNewStore(info: Store): bool	- Đối số truyền vào là biến info kiểu Store, kiểu trả về của hàm là kiểu bool

	- Dùng để thêm mới cửa hàng
deleteStore(info: Store): bool	- Đối số truyền vào là biến info kiểu Store, kiểu trả về của hàm là kiểu bool - Dùng để xoá cửa hàng

*** Class Customer:**

Tên method	Chức năng
viewHistory(): void	- Kiểu trả về của hàm là kiểu void - Dùng để xem lịch sử mua sắm
rateFood(): float	- Kiểu trả về của hàm là kiểu float - Dùng để đánh giá món ăn

*** Class Store:**

Tên method	Chức năng
calculateBalance(): float	- Kiểu trả về của hàm là kiểu float - Dùng để tính toán doanh thu, lợi nhuận của cửa hàng
showWaitingOrder(): void	- Kiểu trả về của hàm là kiểu void - Dùng để xem danh sách đơn hàng đang đợi
getStore(): Store	- Kiểu trả về của hàm là kiểu Store - Dùng để xem thông tin của cửa hàng
setStore(info: string): void	- Đối số truyền vào là biến info kiểu string, kiểu trả về của hàm là kiểu void - Dùng để cập nhật thông tin của cửa hàng

*** Class StoreOwner:**

Tên method	Chức năng
changeFoodPrice(): void	- Kiểu trả về của hàm là kiểu void - Dùng để thay đổi giá món ăn của cửa hàng
viewStoreReport(): void	- Kiểu trả về của hàm là kiểu void - Dùng để xem báo cáo kinh doanh của cửa hàng

*** Class Order:**

Tên method	Chức năng
calculateTotal(): float	- Kiểu trả về của hàm là kiểu float - Dùng để tính tổng giá tiền phải trả của đơn hàng
notifyTimeLeft(): void	- Kiểu trả về của hàm là kiểu void - Dùng để thông báo thời gian còn lại cho đến khi nhận món của đơn hàng
changeStatusToReady(): bool	- Kiểu trả về của hàm là kiểu bool - Dùng để thay đổi trạng thái của đơn hàng thành “hoàn thành” khi làm xong món ăn

*** Class Shopping Cart:**

Tên method	Chức năng
addCartItem(food: Food, quantity: int): bool	- Đối số truyền vào là biến food kiểu Food, biến quantity kiểu int, kiểu trả về của hàm là kiểu bool - Dùng để thêm món ăn vào giỏ hàng
removeCartItem(foodID: int): bool	- Đối số truyền vào là biến foodID kiểu int, kiểu trả về của hàm là kiểu bool - Dùng để xóa món ăn khỏi giỏ hàng

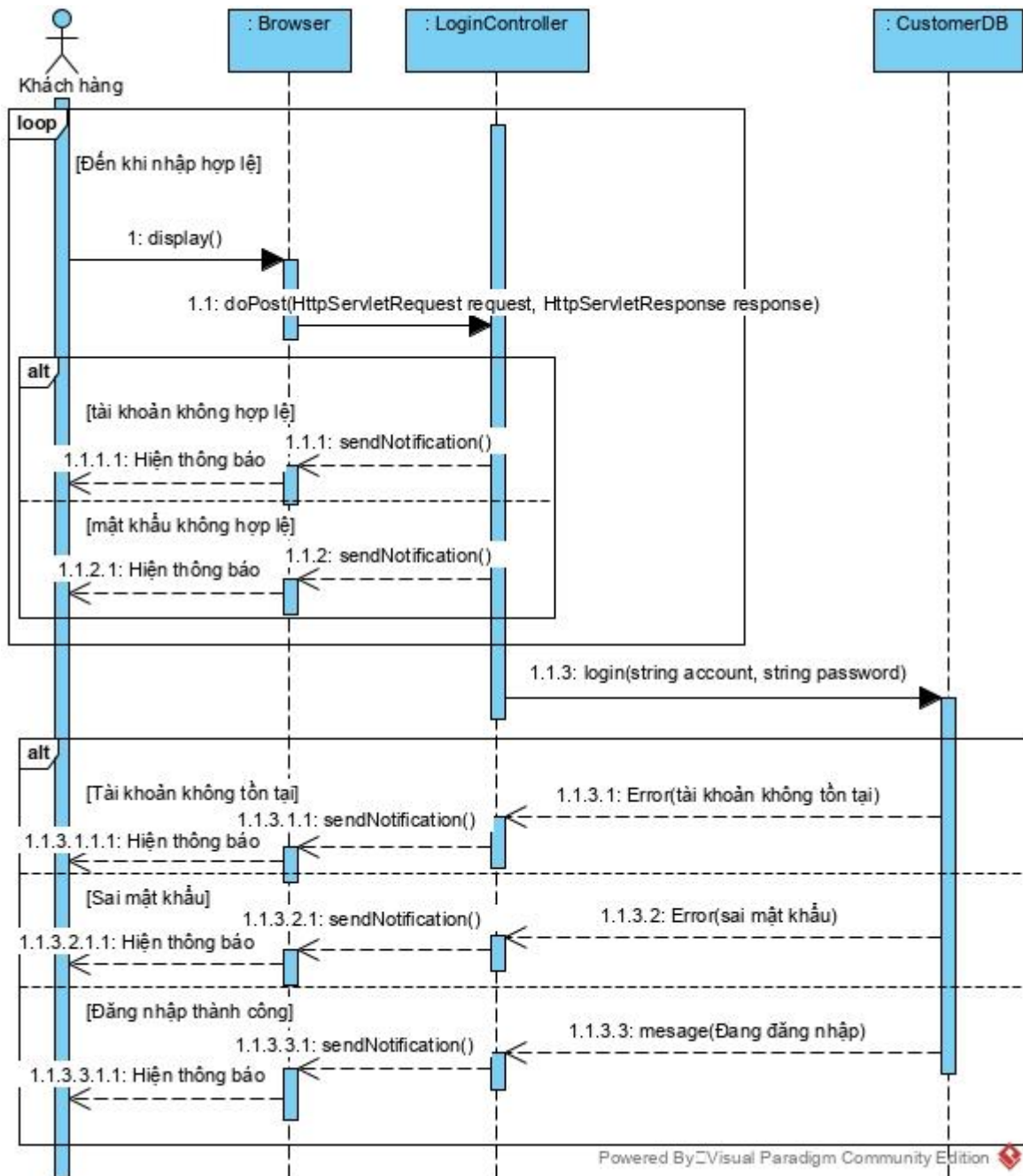
isEmpty(): bool	<ul style="list-style-type: none">- Kiểu trả về của hàm là kiểu bool- Dùng để kiểm tra xem giỏ hàng có món ăn nào hay không
changeQuantity(foodID: int, quantity: int): bool	<ul style="list-style-type: none">- Đối số truyền vào là biến foodID và biến quantity kiểu int, kiểu trả về của hàm là kiểu bool- Dùng để thay đổi số lượng của món ăn đã chọn
makePayment(): bool	<ul style="list-style-type: none">- Kiểu trả về của hàm là kiểu bool- Dùng để thanh toán giỏ hàng

*** Class PagerCallingSystem:**

Tên method	Chức năng
displayNumber(orderID: int): void	<ul style="list-style-type: none">- Đối số truyền vào là biến orderID kiểu int, kiểu trả về của hàm là kiểu void- Dùng để hiển thị số của order lên màn hình pager
createSound(): void	<ul style="list-style-type: none">- Kiểu trả về của hàm là kiểu void- Dùng để ra lệnh cho pager tạo âm thanh thông báo cho khách hàng

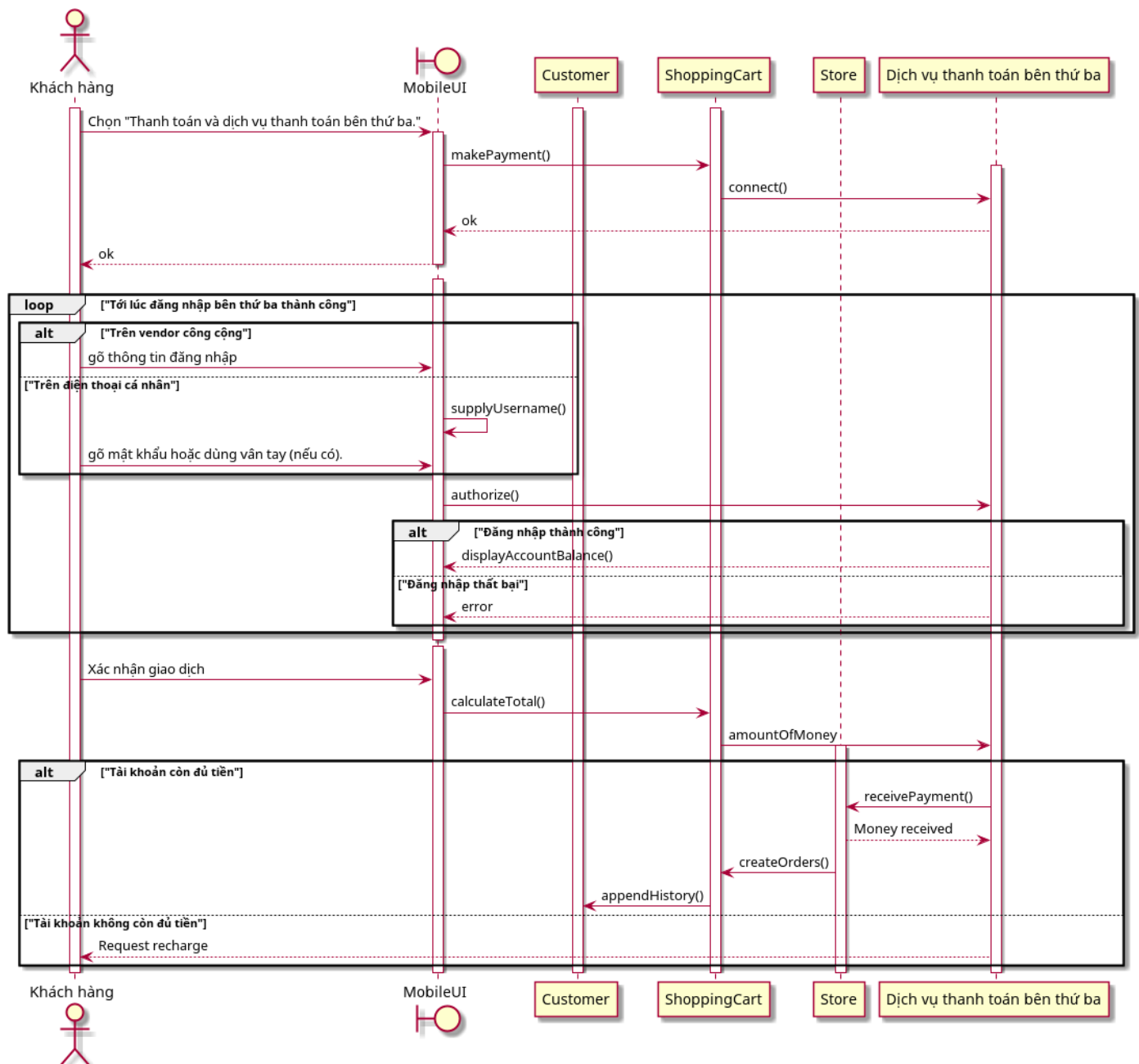
IV. Sequence Diagram

Sequence diagram ở mức chi tiết cho chức năng “Xử lý tác vụ đăng nhập, đăng ký tài khoản của khách hàng”:



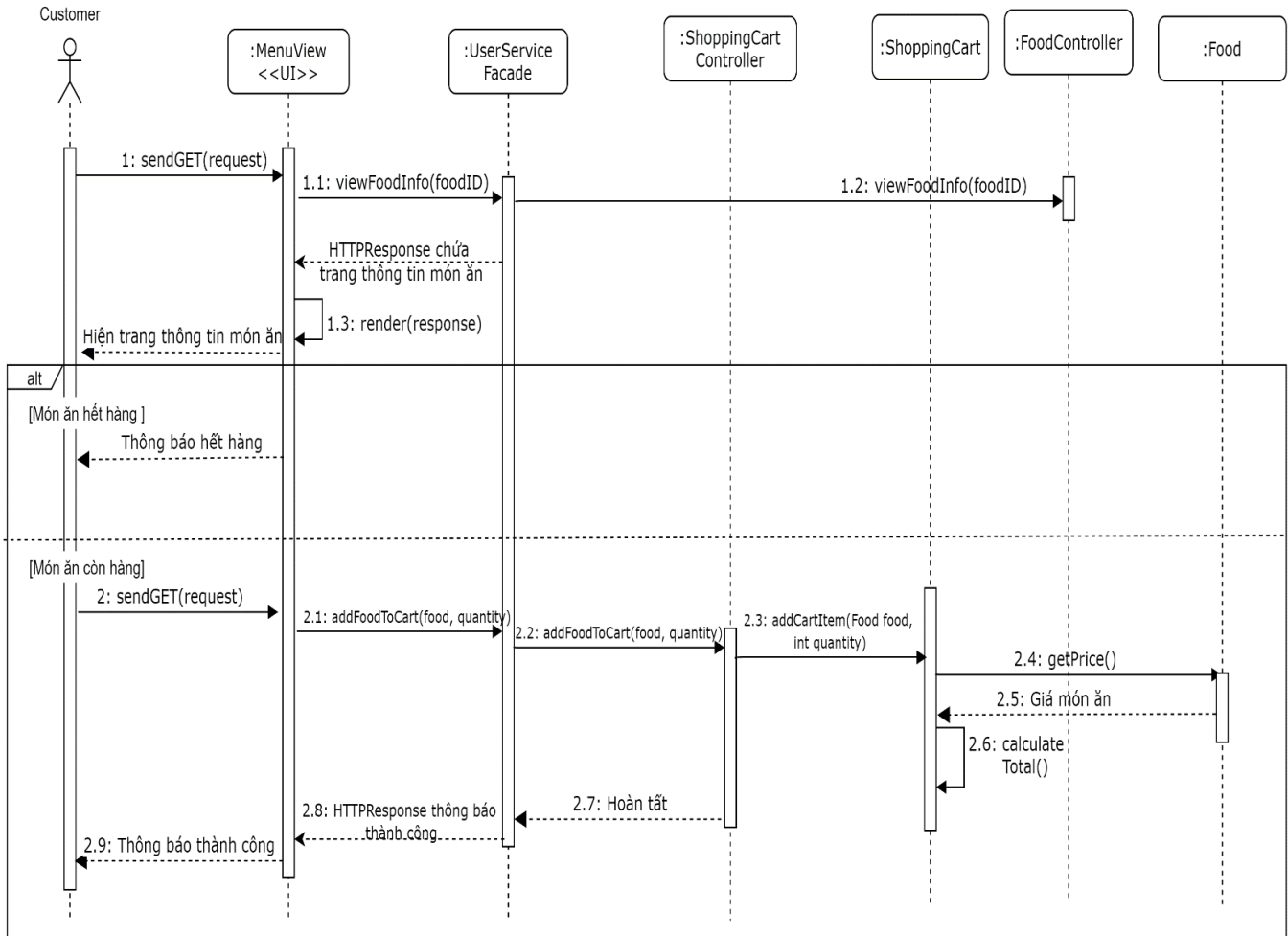
Hình 24: Sequence diagram chi tiết cho chức năng “Xử lý tác vụ đăng nhập, đăng ký tài khoản của khách hàng”

Sequence diagram ở mức chi tiết cho chức năng “Thanh toán”:



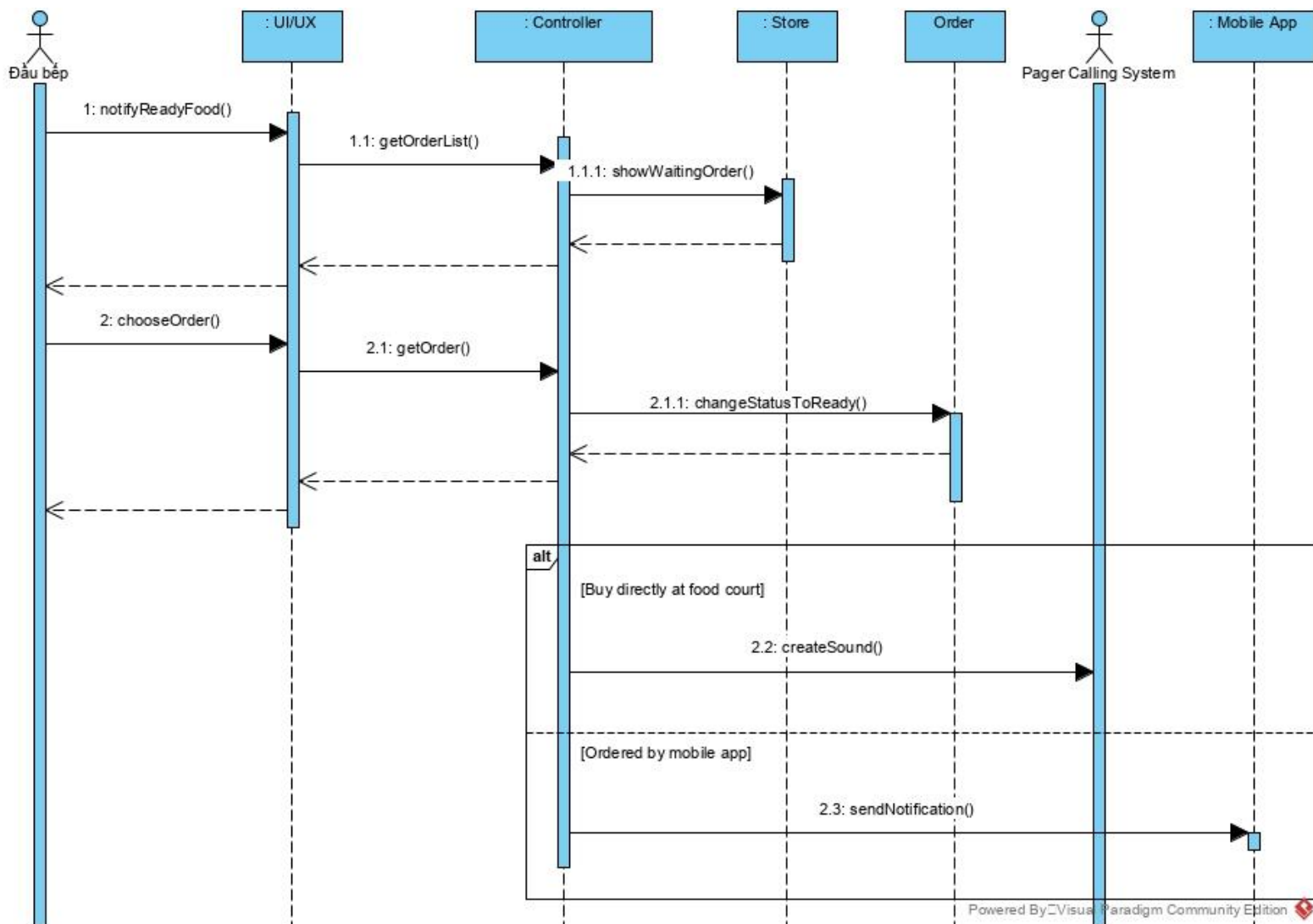
Hình 25: Sequence diagram chi tiết cho chức năng “Thanh toán”

Sequence diagram ở mức chi tiết cho chức năng “Thêm món ăn vào giỏ hàng”:



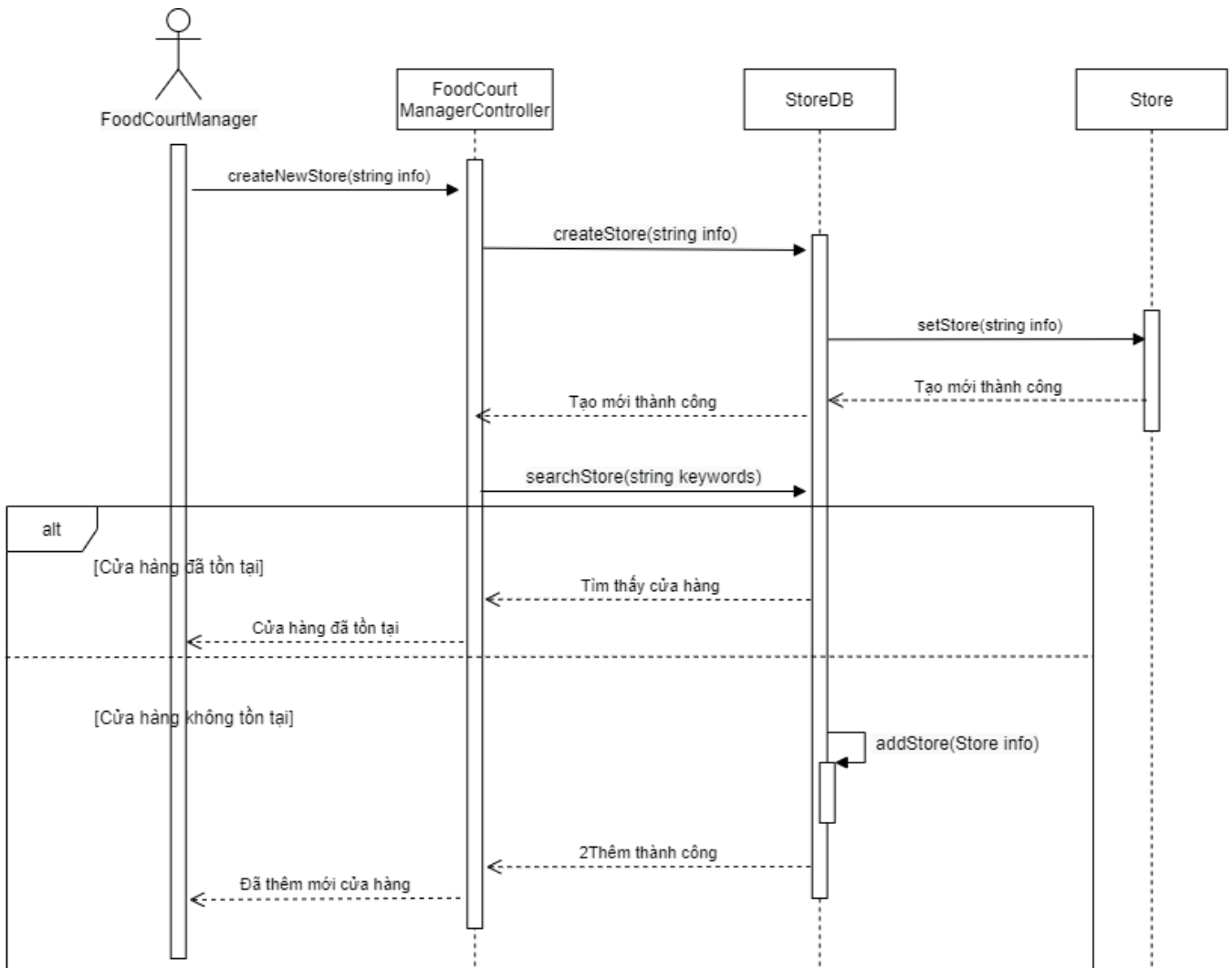
Hình 26: Sequence diagram chi tiết cho chức năng “Thêm món ăn vào giỏ hàng”

Sequence diagram ở mức chi tiết cho chức năng “Thông báo món ăn hoàn thành” của hệ thống:



Hình 27: Sequence diagram chi tiết cho chức năng “Thông báo món ăn hoàn thành”

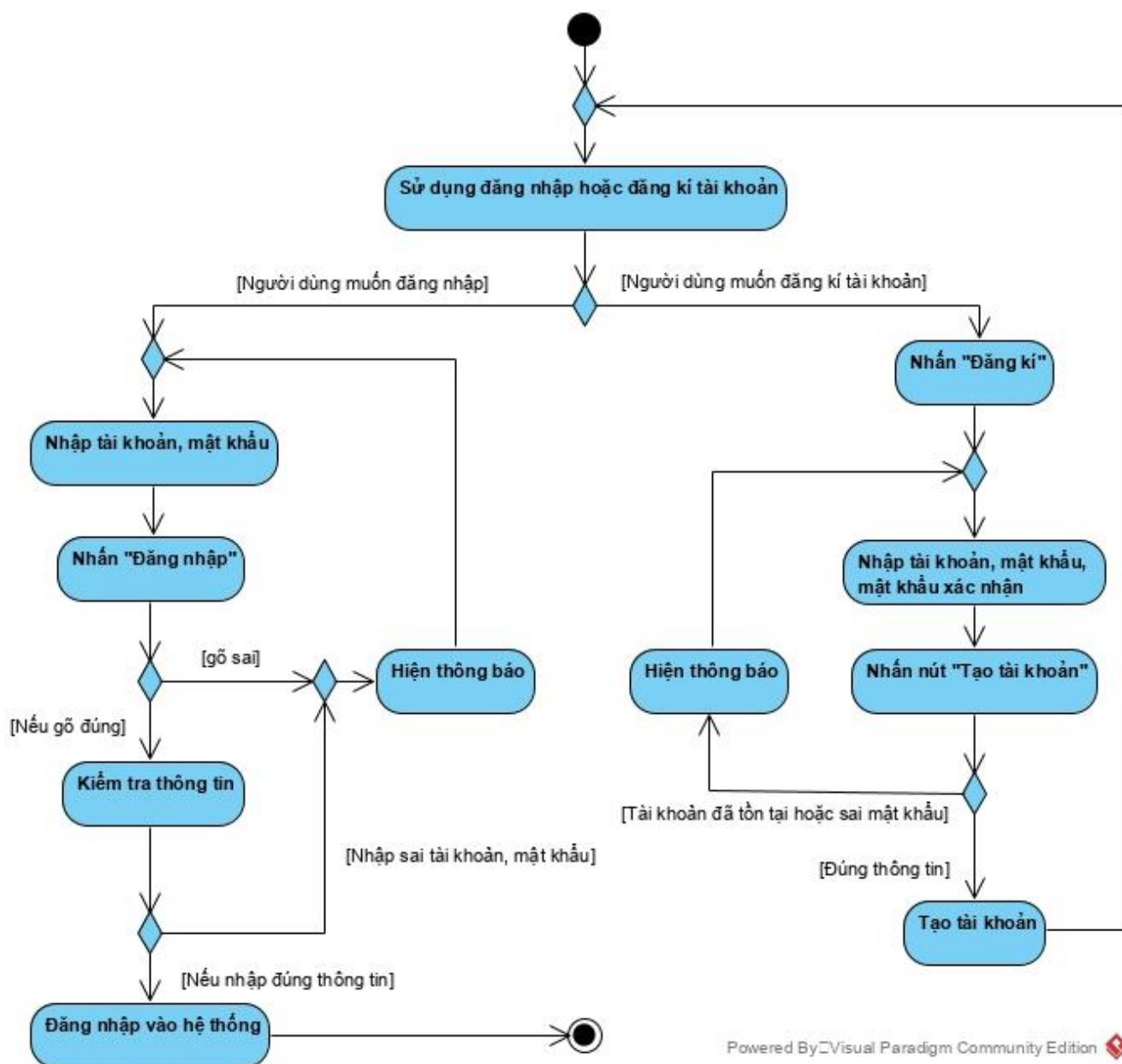
Sequence diagram ở mức chi tiết cho chức năng “Thêm cửa hàng”:



Hình 28: Sequence diagram chi tiết cho chức năng “Thêm cửa hàng”

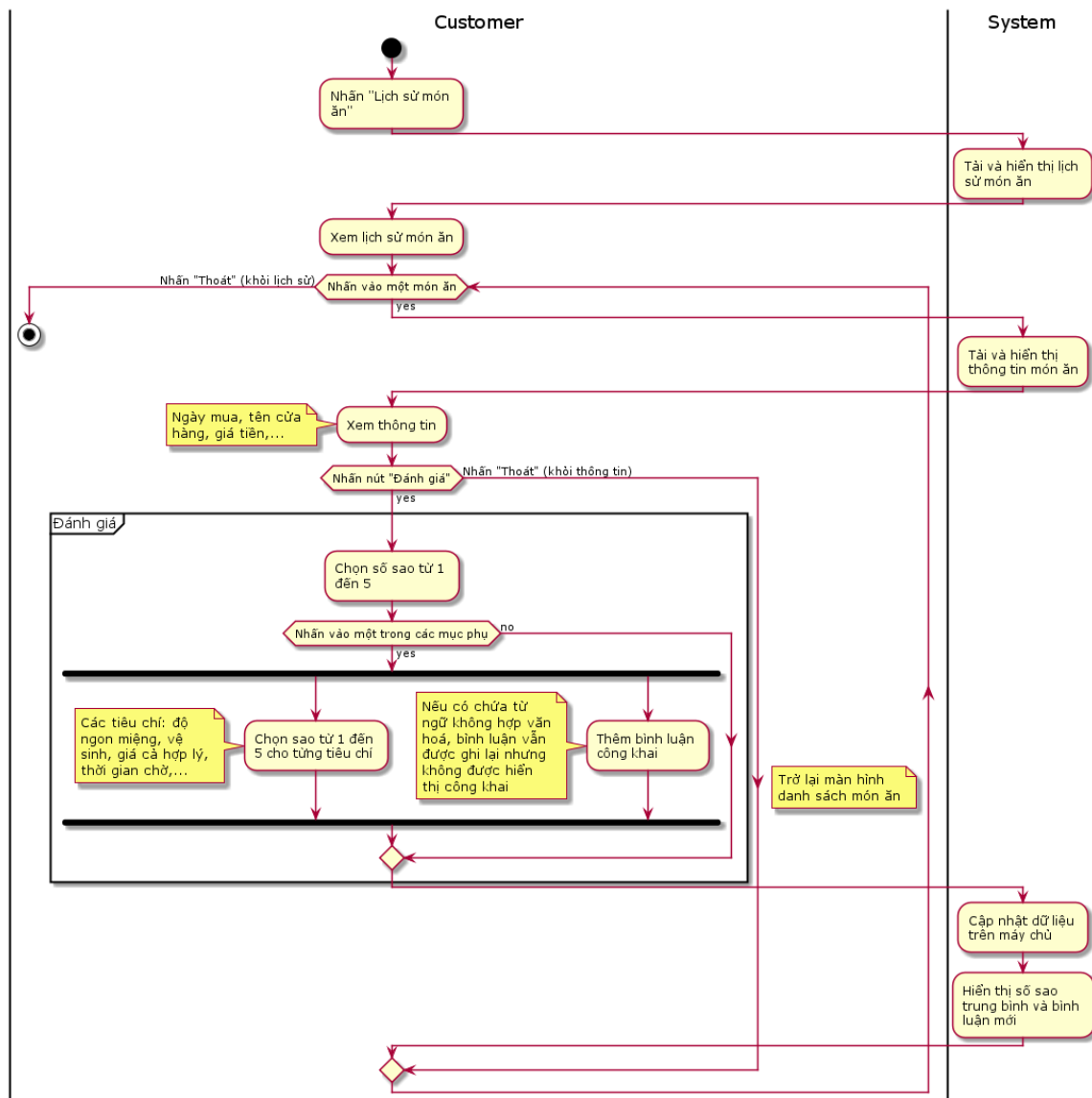
V. Activity Diagram

Activity diagram ở mức chi tiết cho chức năng “Xử lý tác vụ đăng nhập, đăng ký tài khoản của khách hàng”:



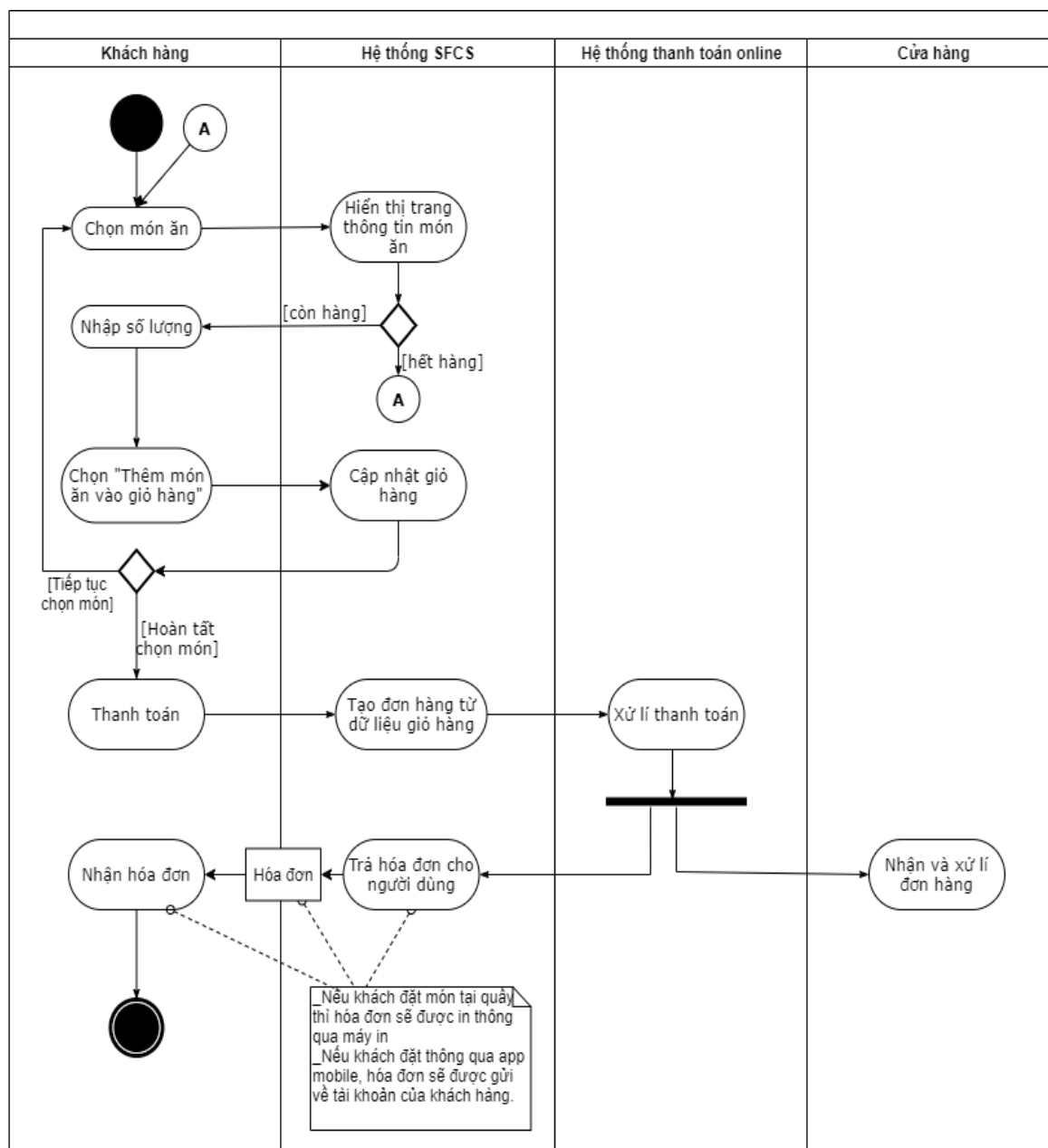
Hình 29: Activity diagram chi tiết cho chức năng “Xử lý tác vụ đăng nhập, đăng ký tài khoản của khách hàng”

Activity diagram ở mức chi tiết cho chức năng “Xem lịch sử và rate”:



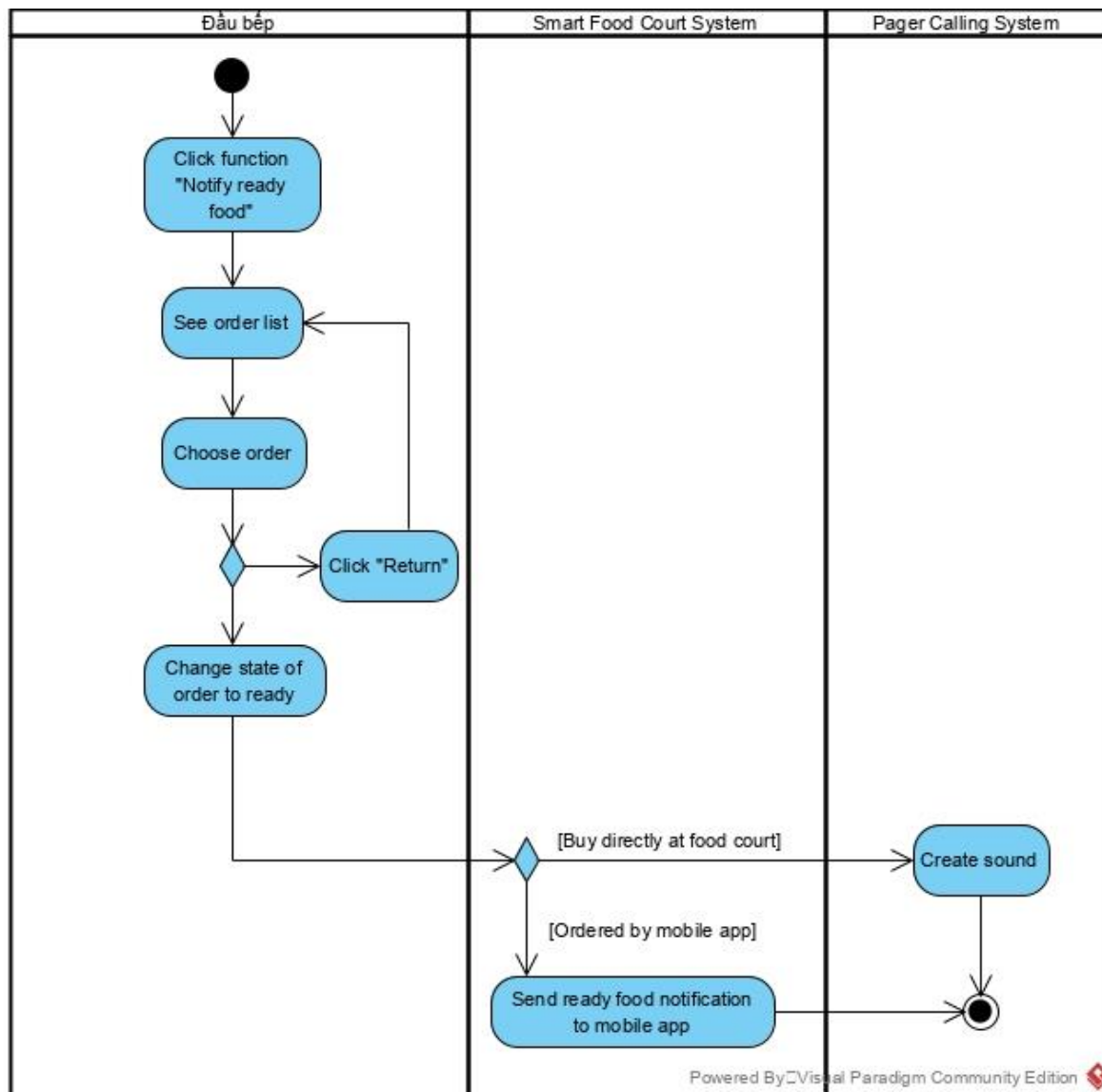
Hình 30: Activity diagram chi tiết cho chức năng “Xem lịch sử và rate”

Activity diagram ở mức chi tiết cho chức năng “Đặt món ăn của khách hàng”:



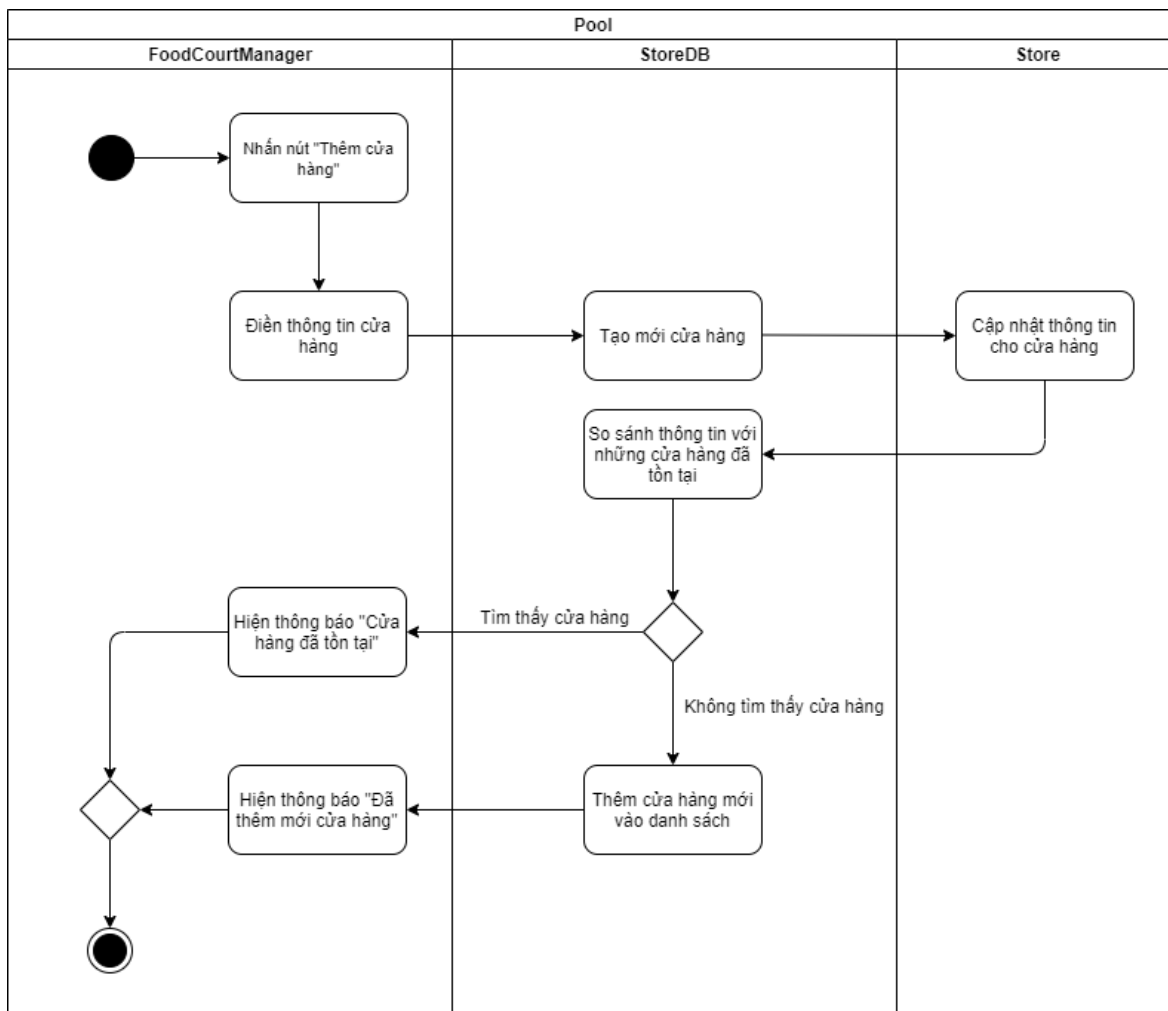
Hình 31: Activity diagram chi tiết cho chức năng “Đặt món ăn của khách hàng”

Activity diagram ở mức chi tiết cho chức năng “Thông báo món ăn hoàn thành”:



Hình 32: Activity diagram chi tiết cho chức năng “Thông báo món ăn hoàn thành”

Activity diagram ở mức chi tiết cho chức năng “Thêm cửa hàng”:



Hình 33: Activity diagram chi tiết cho chức năng “Thêm cửa hàng”

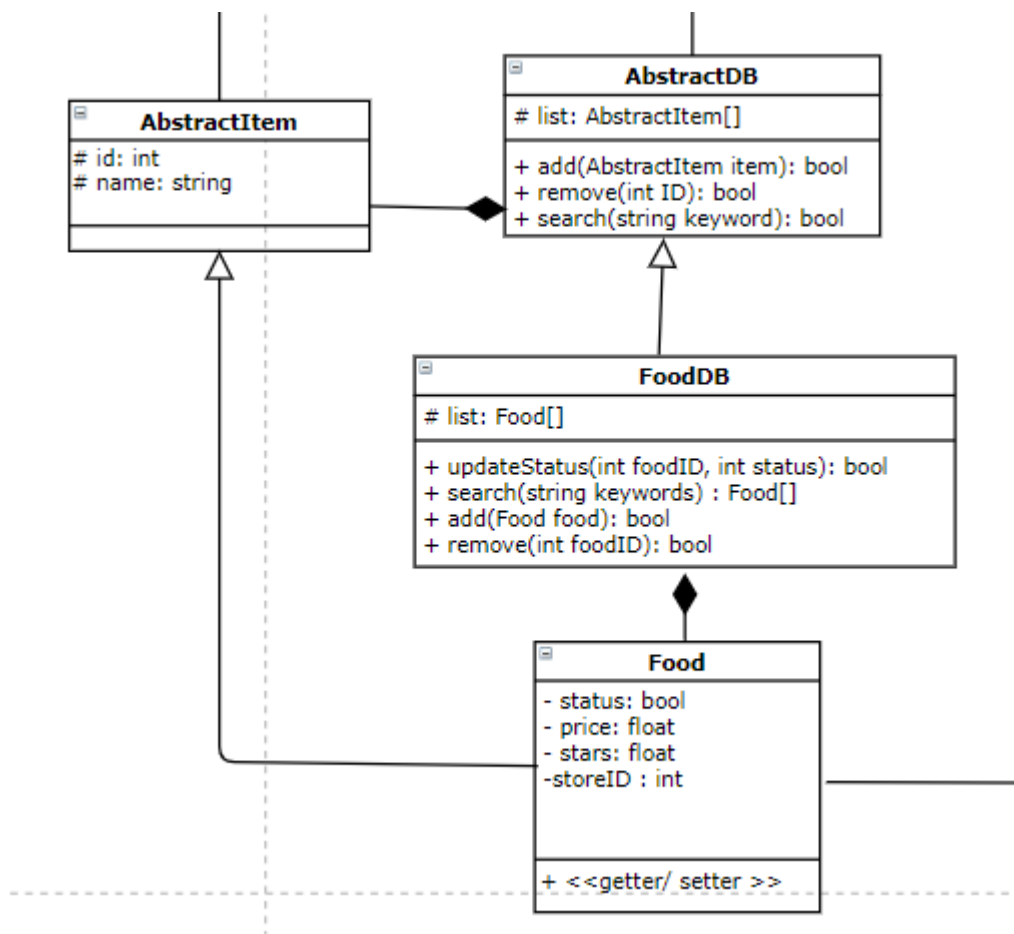
VI. Design Pattern (Bonus)

* Observer:

- Đối với mục cập nhật giao diện web, ta sẽ dùng mẫu Observer. Mẫu thiết kế này sẽ hữu hiệu khi người quản lý thay đổi thông tin về món ăn, cửa hàng, hay thông báo tình trạng món ăn còn hàng hay hết hàng, thì những thông tin đó sẽ ngay lập tức cập nhật lên web để khách hàng biết. Lý do là vì mẫu Observer định nghĩa mối quan hệ phụ thuộc một – nhiều giữa các đối tượng để khi mà một đối tượng có sự thay đổi trạng thái, tất cả các thành phần phụ thuộc của nó sẽ được thông báo và cập nhật một cách tự động.

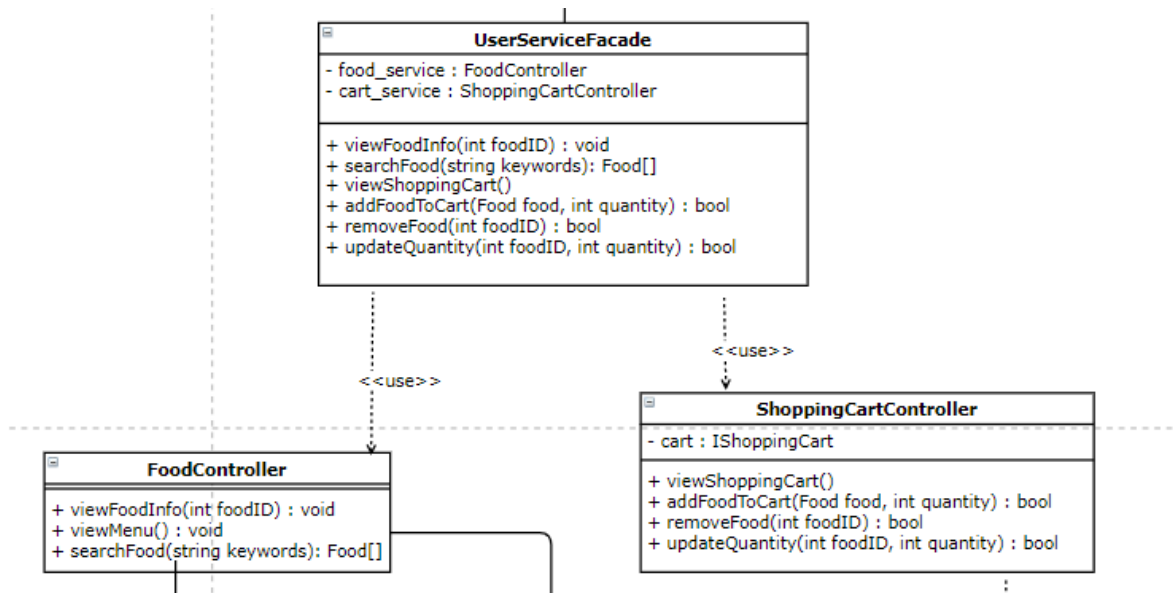
* Template method:

- Sử dụng template method để tạo các abstract class chứa các thông tin cơ bản mà các class con kế thừa đều sử dụng và một số phương thức tạo sẵn mà class con có thể sử dụng hoặc override lại class cha theo yêu cầu chức năng.



* Facade:

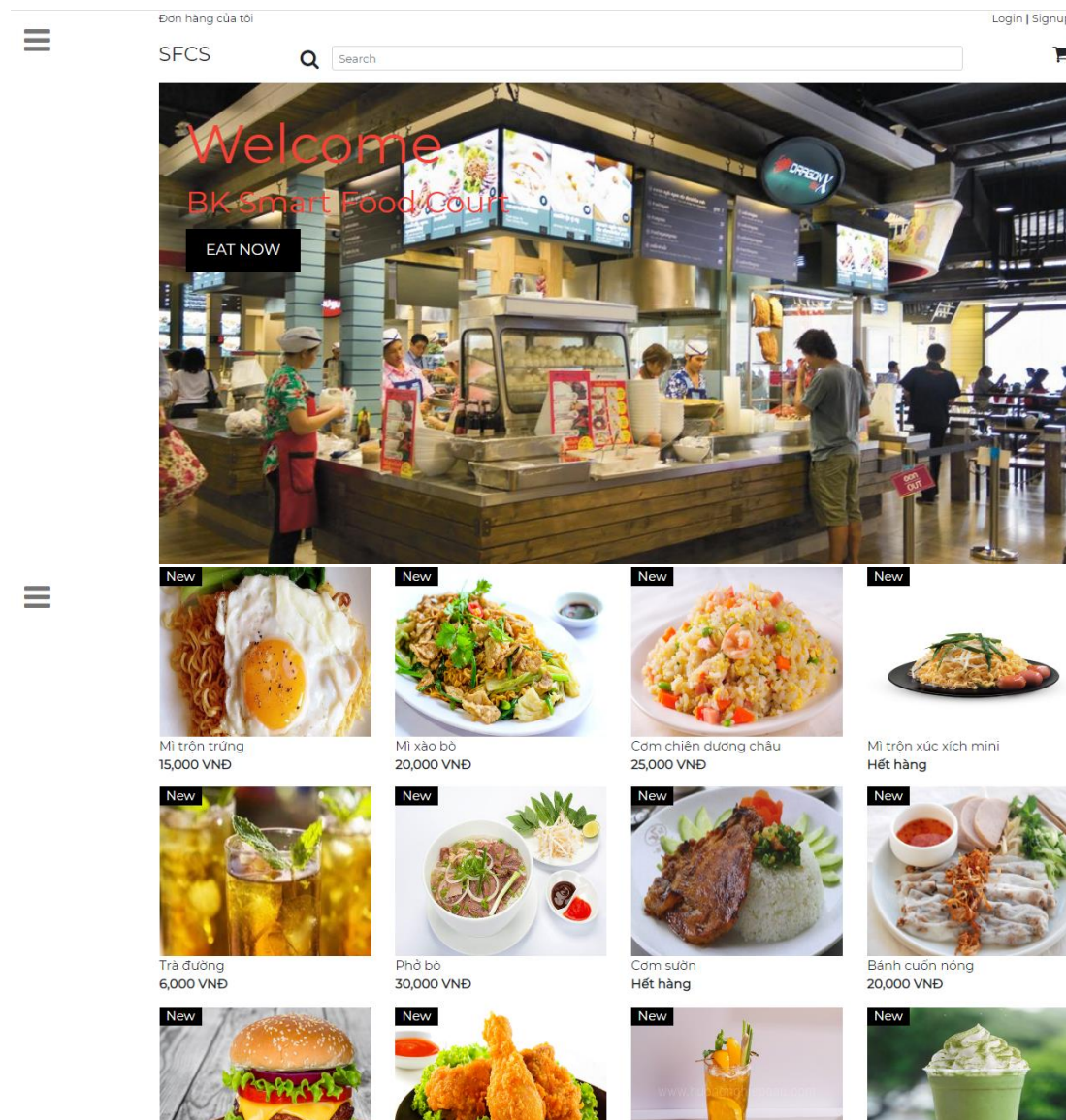
Vì hệ thống bao gồm nhiều hệ thống con phức tạp nên ta sử dụng Façade Pattern trong business layer tạo ra một interface thống nhất, đơn giản để xử lý các event, request của người dùng.



VII. A Working Demonstration (Bonus)

Sau đây là những bức ảnh demo chụp từ trang web FCFS:

- Trang chủ của Smart Food Court System:



Hình 34: Trang chủ SFCS



- Đăng nhập/Đăng ký tài khoản:

Đơn hàng của tôi

SFCS

Search

Login | Signup

WELCOME

Đăng nhập tài khoản

Username:

Password:

[Đăng nhập](#)

[Quên mật khẩu?](#)

[Chưa có tài khoản? Đăng kí](#)

Hình 35: Đăng nhập tài khoản

Đơn hàng của tôi

SFCS

Search

Login | Signup

Đăng kí tài khoản

Username:

Required: 150 characters or fewer. Letters, digits and @/!/-/_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

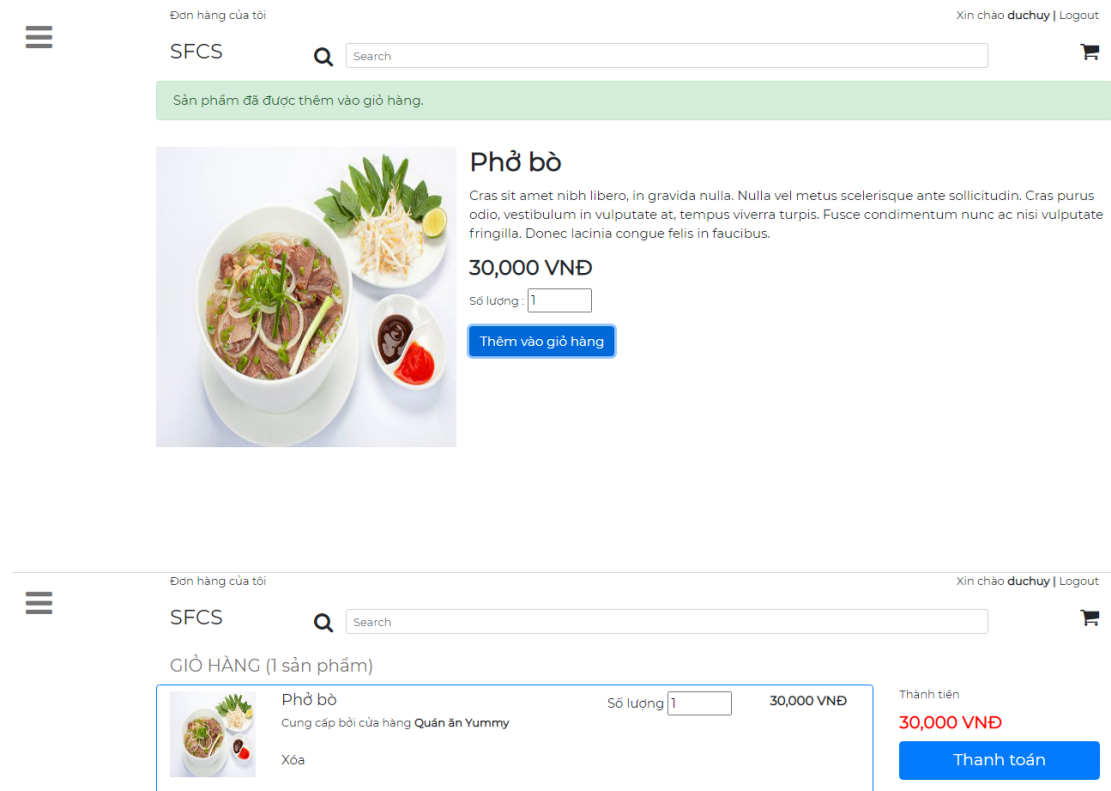
Password confirmation:

Enter the same password as before, for verification.

[Sign up](#)

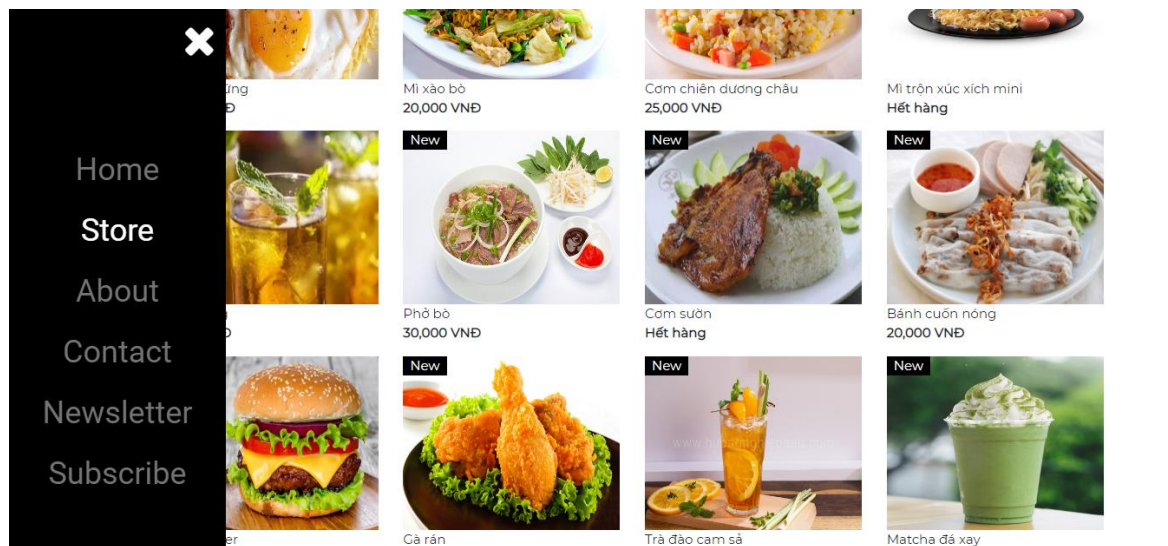
Hình 36: Đăng kí tài khoản

- Khi đã đăng nhập vào hệ thống, chọn mua một món ăn bất kỳ và món ăn đã được thêm vào giỏ hàng thành công:

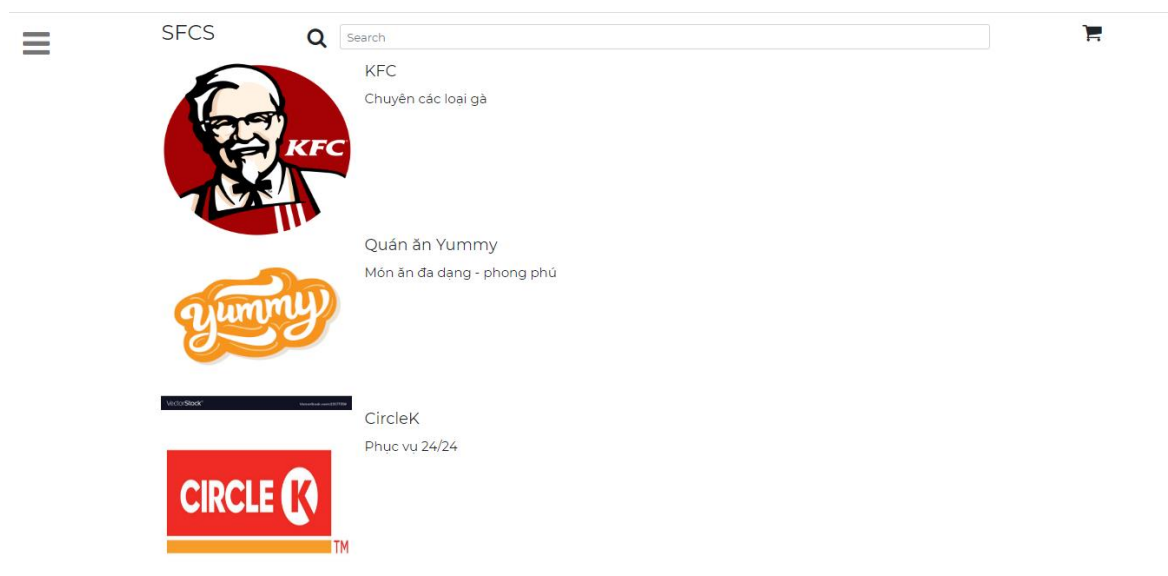


Hình 37: Mua hàng thành công

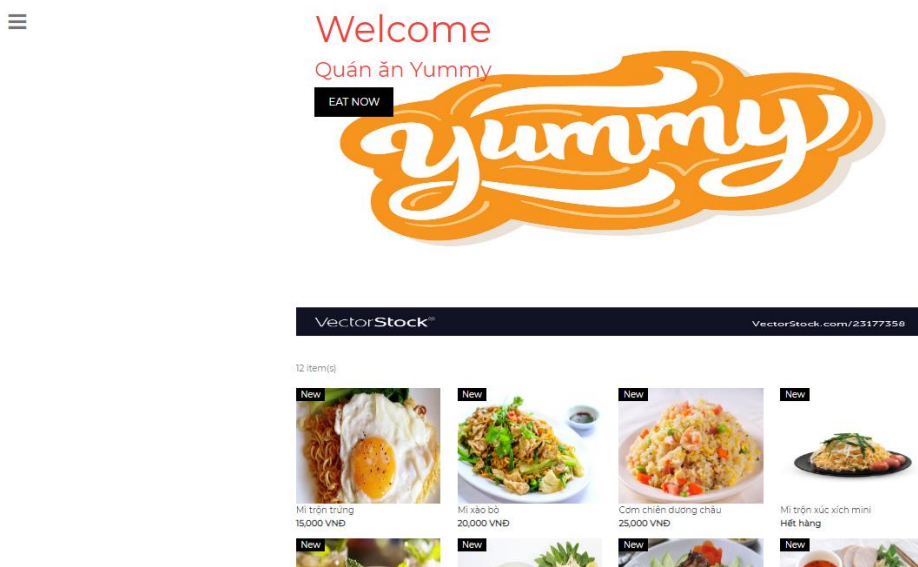
- Chọn “Store” để xem các cửa hàng có sẵn, khi bấm vào cửa hàng cụ thể sẽ hiện menu món ăn của cửa hàng đó (Giả sử ở đây ta chọn “Quán ăn Yummy”):



Hình 38: Chọn “Store” ở thanh công cụ

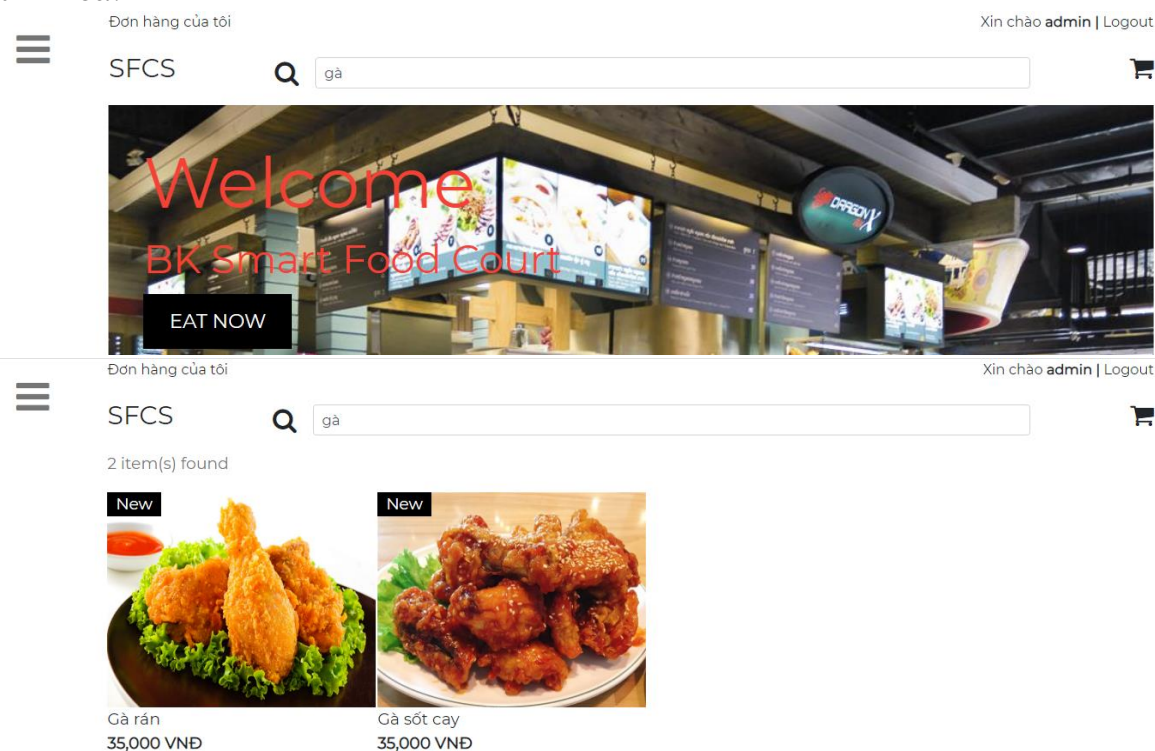


Hình 39: Danh sách cửa hàng



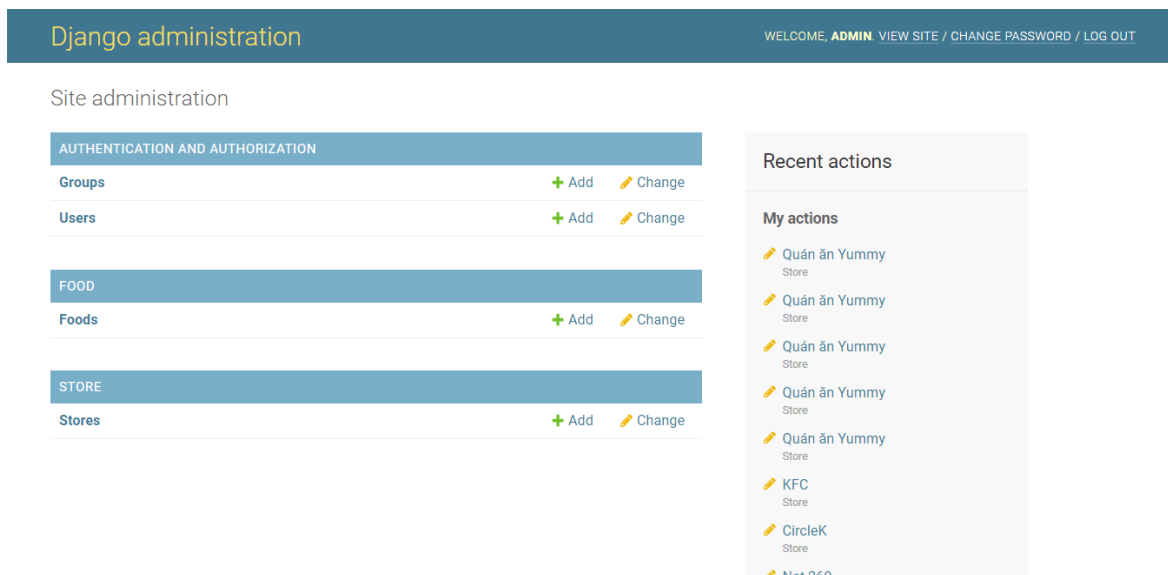
Hình 40: Menu của “Quán ăn Yummy”

- Tìm kiếm món ăn trên thanh tìm kiếm sẽ cho ra các món ăn tương ứng với từ khoá:



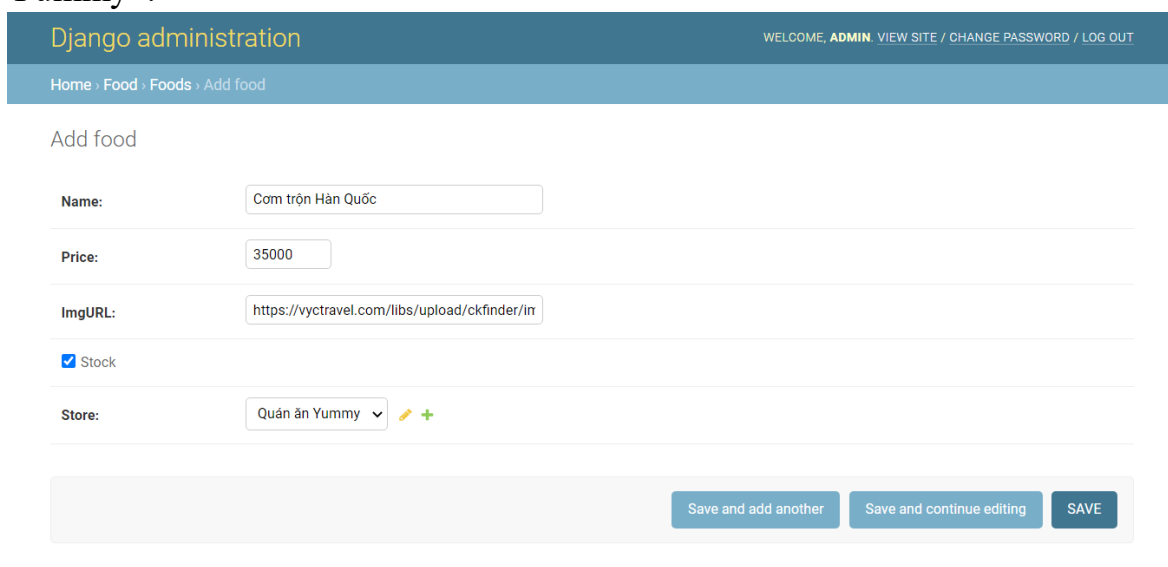
Hình 41: Tìm kiếm món “gà” và kết quả

- Sau đây là trang web quản lý dành cho chủ cửa hàng, quản lý khu ăn uống. Tại đây ta có thể chỉnh sửa, thêm/xoá món ăn, cửa hàng sao cho phù hợp với nhu cầu:



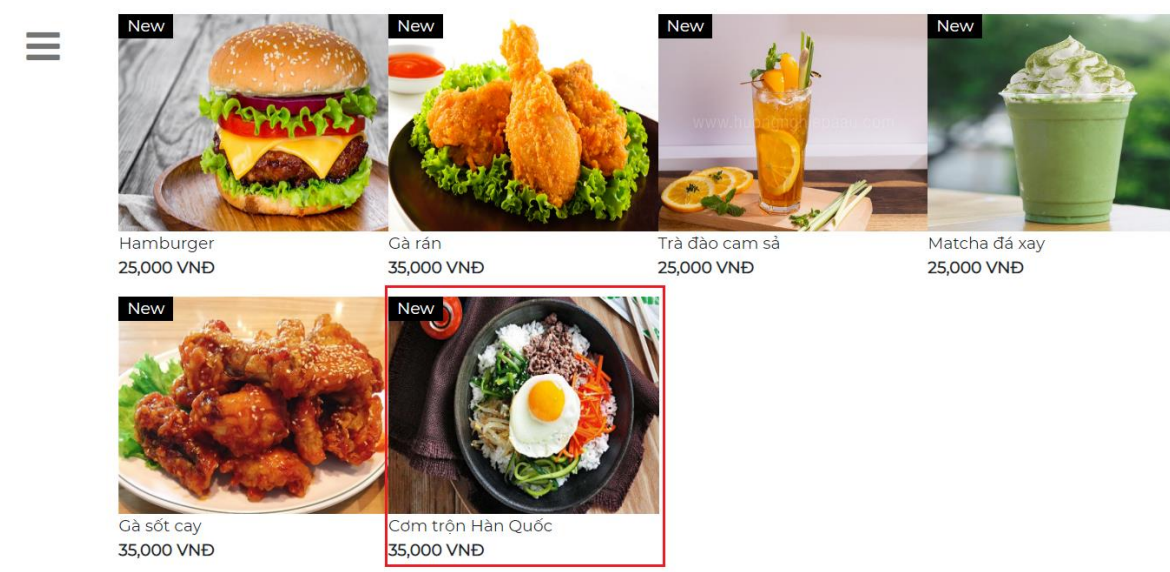
Hình 42: Trang web quản lý

- Thử thêm một món ăn “Cơm trộn Hàn Quốc” cho cửa hàng “Quán ăn Yummy”:



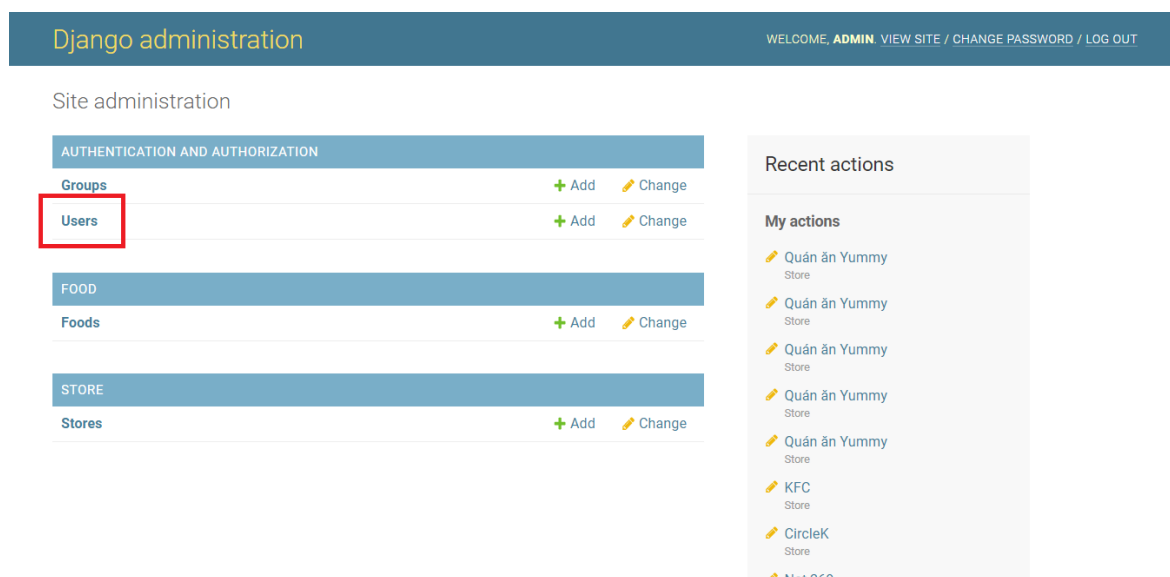
Hình 43: Thêm món “Cơm trộn Hàn Quốc”

- Món ăn đã được thêm thành công và hiển thị ở trang chủ:



Hình 44: Món ăn được thêm thành công

- Ngoài ra, ở trang quản lý còn có thể xem có bao nhiêu người đã đăng ký tài khoản trên web bằng cách nhấn vào mục “Users” ở web quản lý:



Hình 45: Chọn “Users” ở web quản lý

Django administration

WELCOME, ADMIN.

Home > Authentication and Authorization > Users

Select user to change

Q

Search

Action:

Go

 0 of 4 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	abc				✖
<input type="checkbox"/>	admin	giahuynghuyen2k@gmail.com			✔
<input type="checkbox"/>	duchuy				✖
<input type="checkbox"/>	godlike				✖

4 users

Hình 46: Danh sách tài khoản

PHÂN CÔNG CÔNG VIỆC

Họ và tên	MSSV	Nhiệm vụ	Hoàn thành
Lê Đức Huy	1810166	- Làm chức năng: Xem đơn hàng; Thông báo thời gian nhận món; Thông báo món ăn hoàn thành; Cập nhật món ăn hết hàng; Hiện thị số của đơn hàng lên Pager. - Viết báo cáo tổng hợp, Meeting minutes.	100%
Nguyễn Gia Huy	1810173	- Làm chức năng: Thêm món ăn vào giỏ hàng; Chỉnh sửa giỏ hàng; Thêm món ăn; Xoá món ăn; Sửa giá món ăn. - Viết Git reports.	100%
Lê Trung Đan	1810887	- Làm chức năng: Thanh toán; Nạp thêm tiền; Xem lịch sử đơn hàng; Nhận hoá đơn; Đánh giá món ăn. - Viết Test report.	100%
Hoàng Tấn Phát	1811137	- Làm chức năng: Đăng nhập; Đăng kí tài khoản; Đăng xuất; Tìm kiếm món ăn. - Viết Test report.	100%
Đỗ Lê Quang Trung	1811304	- Làm chức năng: Thêm cửa hàng; Xoá cửa hàng; Xem báo cáo kinh doanh của cả khu ăn uống;	100%



		Xem báo cáo kinh doanh của cửa hàng; Bảo trì hệ thống. - Viết Test report.	
--	--	---	--

LINK GITHUB

https://github.com/ghuyng/demoSFCS

TÀI LIỆU THAM KHẢO

[1] *UML Modeling*; Truy cập từ:

<https://www.visual-paradigm.com/VPGallery/diagrams/index.html>

[2] *The Unified Modeling Language*; Truy cập từ:

<https://www.uml-diagrams.org/>

[3] *Software Engineering Tutorial*; Truy cập từ:

https://www.tutorialspoint.com/software_engineering/index.htm