

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



CÔNG NGHỆ PHẦN MỀM (CO3001)

Đề tài:

Restaurant POS 2.0

Giảng viên:	Thầy Lê Đình Thuận
Sinh viên thực hiện:	Nguyễn Lê Hiên - 1913315
	Nguyễn Văn Thương - 1915439
	Lưu Công Định - 1913114
	Phùng Minh Khánh - 1913756
	Trần Tiến Lộc - 1914038
	Bùi Tiến Bình - 1912713

TP. HỒ CHÍ MINH, THÁNG 11/2021

Mục lục

1 Requirement elicitation	1
1.1 Defining the context of this project, relevant stakeholders and the scope of the project	1
1.2 Describe all functional and non-functional requirements of the desired system. Draw a use-case diagram for the whole system	2
1.2.1 Functional requirement	2
1.2.2 Non-functional requirement	4
1.2.3 Draw a use-case diagram for the whole system	7
1.3 Choose one specific feature, i.e. food ordering, tablereservation, customer management. Draw its use-casediagram and describe the use-case using a table format	8
1.3.1 Use-case : Đặt bàn	9
1.3.2 Use-case: Thanh toán	11
1.3.3 Use-case : Lịch sử thanh toán	13
1.3.4 Use case : Xem menu(Xem chi tiết món ăn) và Thêm món ăn vào giỏ hàng	14
1.3.5 Use case : Đăng ký	16
1.3.6 Use case : Đăng nhập	18
1.3.7 Use-case : Xem lịch sử đặt bàn	20
1.3.8 Use-case : Lọc danh sách món theo danh mục	21
2 System modelling	22
2.1 Activity diagram to capture Major (not all) functional requirements of the desired system.	22
2.1.1 Đặt bàn	22
2.1.2 Thanh toán	23
2.1.3 Lịch sử thanh toán	24
2.1.4 Xem menu, xem chi tiết món ăn và thêm món ăn vào giỏ hàng	25
2.1.5 Đăng nhập	26
2.1.6 Đăng ký	27
2.1.7 Lịch sử đặt bàn	28
2.1.8 Lọc danh sách món theo danh mục	29
2.2 Sequence diagram for use-case in Task 1.3.	30

2.2.1	Đặt bàn	30
2.2.2	Thanh toán	31
2.2.3	Lịch sử thanh toán	32
2.2.4	Xem menu và xem chi tiết món ăn	33
2.2.5	Thêm món ăn vào giỏ hàng	34
2.2.6	Đăng nhập	35
2.2.7	Đăng ký	36
2.2.8	Lịch sử đặt bàn	37
2.2.9	Lọc danh sách món theo danh mục	37
2.3	Class Diagram	38
3	Architecture design	39
3.1	Describe an architectural approach you will use to implement the desired system	39
3.1.1	Database Management	39
3.1.2	Back End	39
3.1.3	Front End	40
3.2	Draw an implementation diagram for Major (not all) functional requirements	40
3.2.1	Đặt bàn	40
3.2.2	Thanh toán	41
3.2.3	Lịch sử thanh toán	41
3.2.4	Thêm món vào giỏ hàng	42
3.2.5	Đăng nhập	42
3.2.6	Đăng ký	43
3.2.7	Lịch sử đặt bàn	43
3.2.8	Lọc danh sách món theo danh mục	44
4	Implementation	44
4.1	Setting up github	44
4.2	Giao diện đã hoàn thành	45
4.2.1	Header	45
4.2.2	Footer	45
4.2.3	Đăng nhập	45
4.2.4	Đăng ký	46
4.2.5	Tạo mật khẩu mới	47

4.2.6	Trang chủ	49
4.2.7	Menu	50
4.2.8	Chi tiết món ăn	50
4.2.9	Đặt bàn	51
4.2.10	Lịch sử đặt bàn	52
4.2.11	Thanh toán	52
4.2.12	Các phương thức thanh toán	53
4.2.13	Lịch sử thanh toán	53
4.2.14	Giỏ hàng	53
4.2.15	Flow of screen	54
4.3	Mô tả các chức năng đã hoàn thành	55
4.3.1	Layout và Trang chủ	55
4.3.2	Đăng nhập, đăng ký, lấy lại mật khẩu	55
4.3.3	Hiển thị menu món ăn, Xem chi tiết món ăn, Thêm món ăn vào giỏ hàng	55
4.3.4	Đặt bàn và xem lịch sử đặt bàn	56
4.3.5	Xem giỏ hàng và chỉnh sửa giỏ hàng	56
4.3.6	Thanh toán và xem lịch sử thanh toán	56
5	Database diagram và mô tả cách kết nối cơ sở dữ liệu	57
6	Công nghệ Trending	59
TÀI LIỆU THAM KHẢO		62

1 Requirement elicitation

1.1 Defining the context of this project, relevant stakeholders and the scope of the project

- **Ngữ cảnh:** Trong đại dịch Virus Corona, các nhà hàng đang phải đổi mới với nguy cơ lớn hơn bao giờ hết. Các hệ thống như vậy được kỳ vọng sẽ tăng cường trí tuệ kinh doanh, giảm thiểu nỗ lực lãng phí và cơ hội mở rộng quy mô thành doanh nghiệp lớn. Restaurant POS 2.0 cung cấp giải pháp cho đặt món ăn tại nhà hàng mà không cần tiếp xúc trực tiếp với nhân viên. Phần nào giúp khách hàng chủ động hơn trong việc gọi món, đưa ra những phản hồi về bữa ăn và dịch vụ nhà hàng. Restaurant POS 2.0 mang đến trải nghiệm mới cho thực khách.
- **Những đối tượng liên quan:** Khách hàng, chủ nhà hàng, nhân viên, đầu bếp, ngân hàng, người quản lý hệ thống,...
- **Phạm vi dự án:** được xác định thông qua mục đích, các module sản phẩm mong muốn, các functional requirement và Non-functional requirement, các tính năng sẽ thực hiện trong dự án, chi phí cũng như deadline của dự án. Cụ thể:
 - Mục đích: tạo ra phần mềm pos restaurant với các hành vi có thể giao tiếp qua mạng một cách thuận tiện, tránh tiếp xúc trực tiếp nhiều nhất có thể giữa các bên ứng với bối cảnh đại dịch covid 19.
 - Các module sản phẩm mong muốn:
 - + Có thể đặt món ăn, đặt bàn thông qua menu online.
 - + Thanh toán online.
 - + Nhân viên xác nhận đơn đặt hàng, giao tiếp với khách hàng online.
 - Các tính năng cần thực hiện:
 - + Đặt bàn => scope: Chỉ các thành viên VIP (thanh toán trên 10 lần trở lên) mới được phép đặt bàn.
 - + Lịch sử đặt bàn => scope: Chỉ các thành viên VIP (thanh toán trên 10

(lần) mới có tính năng xem Lịch sử đặt bàn.

- + Thanh toán => scope: Khách hàng phải là thành viên mới sử dụng được tính năng này. Thanh toán bằng thẻ ngân hàng, Visa, Mastercard, ví điện tử hoặc tiền mặt.
- + Lịch sử thanh toán => scope: khách hàng là thành viên.
- + Thêm món vào giỏ hàng => scope: Món ăn hiển thị trong menu thì số lượng còn lại ít nhất là 1. Là thành viên mới được thêm món vào giỏ hàng.
- + Đăng ký, đăng nhập => scope: chỉ đăng nhập khi đã đăng ký tài khoản (đăng nhập bằng username và password), không hỗ trợ đăng ký nhanh bằng google/facebook.
- + Đăng xuất => scope: phải đăng nhập mới xuất hiện tính năng này.
- + Tạo mật khẩu mới => scope: Email khách hàng phải tồn tại.
- + Hiện thị menu món ăn => scope: món ăn tồn tại trong database.
- + Lọc danh sách món ăn => scope: loại thức ăn phải có trong database.
- + Hiển thị chi tiết món ăn.
- + Xem giỏ hàng.

1.2 Describe all functional and non-functional requirements of the desired system. Draw a use-case diagram for the whole system

1.2.1 Functional requirement

1.2.1.1 Đối với Khách hàng

* Interactive requirement:

- Chức năng đăng ký tài khoản cho khách hàng mới: trở thành thành viên.
- Chức năng đăng nhập: Đăng nhập vào hệ thống
- Chức năng đăng xuất: Thoát khỏi hệ thống
- Chức năng tạo mật khẩu mới khi khách hàng quên mật khẩu: sử dụng email đã đăng ký để xác nhận và tạo mật khẩu mới.

- Chức năng đặt bàn cho thành viên VIP:
 - + Hiển thị các bàn còn trống.
 - + Xác nhận đặt bàn.
 - + Đặt bàn trong khoảng thời gian từ 30ph-3giờ
- Chức năng xem lịch sử đặt bàn: Xem tất cả các bàn đã đặt vào các khung giờ, số lượng, số bàn.
- Chức năng hiển thị menu: Cho phép người dùng xem tất cả các món ăn hiện có và có thể lọc món ăn theo loại món ăn và có thể thêm món ăn vào giỏ hàng với số lượng là 1.
- Hiển thị chi tiết món ăn: Khi khách hàng click vào tên món ăn ở menu thì hiển thị chi tiết về món ăn và có thể chọn số lượng để thêm vào giỏ hàng.
- Chức năng giỏ hàng: Cho phép người dùng xem và chỉnh sửa số lượng món ăn đã thêm vào giỏ hàng.
- Chức năng thanh toán: Khách hàng có thể kiểm tra hóa đơn và thanh toán dễ dàng, nhanh chóng.
- Chức năng lịch sử thanh toán: Khách hàng có thể xem lại lịch sử thanh toán.

* **Non-interactive requirement:**

- Trong quá trình đặt hàng, dùng công nghệ AI để đề xuất các đơn hàng theo nhiều loại yếu tố (xếp hạng, giá mua, sản phẩm mới, combo, tần suất đặt hàng). Đối với các user là thành viên, có thể đề xuất theo món ăn yêu thích, độ tuổi, giới tính của từng user.
- Gửi thông kê điểm tích lũy cho từng user vào đầu mỗi tháng.
- Trong quá trình thanh toán, thực hiện discount cho khách hàng vào những ngày lễ. Hoặc đối với thành viên, có thể discount theo số điểm đã tích lũy hoặc thực hiện discount vào ngày sinh nhật,...

1.2.1.2 Đối với nhân viên:

* **Interactive requirement:**

- Hiển thị các bàn đã đặt bởi khách hàng ứng với từng thời gian đến của khách hàng. (có thể nhiều khách hàng)

- Hiển thị các đơn đặt hàng của khách hàng.
- Chức năng xác nhận đơn hàng và in hóa đơn online cho khách hàng.

*** Non interactive requirement:**

- Các đơn đặt hàng hiển thị để xác nhận phải theo thứ tự hoặc độ ưu tiên (tùy vào từng thành viên).

1.2.1.3 Đối với đầu bếp:

- Đầu bếp cập nhật món ăn có còn hay không.
- Đầu bếp có thể thực hiện yêu cầu các nguyên liệu cần thiết đang thiếu.
- Đầu bếp cập nhật thời gian hoàn thành thức ăn.

1.2.1.4 Đối với các bên khác:

- Chủ nhà hàng, quản lý nhà hàng có thể xem báo cáo doanh thu tổng và doanh thu với từng khu ăn uống trong ngày.
- Cho phép các quản lý từng khu vực ăn uống xem báo cáo doanh thu của từng khu.
- Quản lý khách hàng
 - + Cho phép chủ nhà hàng, quản lý nhà hàng xem danh sách thông tin tài khoản và dữ liệu mua hàng của thành viên nhà hàng.
 - + Cho phép chủ nhà hàng, quản lý nhà hàng xem phản hồi, đánh giá khách hàng.
 - + Cho phép chủ nhà hàng, quản lý nhà hàng xem thống kê các món ăn được order trong tháng của khách hàng.

1.2.2 Non-functional requirement

*** General non- functional requirement**

- Thời gian hoạt động của hệ thống là từ 8h đến 23h30 vào tất cả các ngày trong tuần.
- Khi đầu bếp thực hiện cập nhật món ăn, hệ thống phải đồng bộ lên menu ngay lập tức tối đa khoảng 100ms.

- Thời gian hiện thị toàn bộ menu sau khi truy cập web hoặc thông qua mã QR không quá 1s.
- Thời gian tiếp nhận và xử lý đơn đặt hàng không quá 3 giây.
- Web app thân thiện. Nhân viên, đầu bếp và quản lý chủ nhà hàng có thể học được cách sử dụng hệ thống trong vòng 30 phút. User có thể sử dụng web app dễ dàng sau khi đọc các hướng dẫn chức năng trên thanh công cụ duy nhất 1 lần.
- Các chức năng chỉnh sửa hệ thống, thông tin người dùng tuyệt đối bảo mật, chỉ có chủ nhà hàng hoặc quản lý có thể thay đổi hoặc truy cập.
- Đơn vị tiền tệ được sử dụng trên hệ thống là VNĐ.
- Gửi email định kỳ đến các thành viên đã lâu rồi chưa đến quán cùng với discount ưu đãi, để thu hút.
- Web app chạy tốt, phản hồi nhanh tối đa là 1.5s với băng thông mạng 30 mbps.
- Có thể chạy trên bất cứ hệ điều hành cũng như thiết bị nào.

* Individual non-functional requirement:

- Đăng ký tài khoản: Mật khẩu phải gồm cả kí tự đặc biệt, chữ hoa, chữ thường và số
- Đăng nhập:
 - + Sau khi đăng nhập thành công, hiện thị trang chủ nhà hàng trong vòng tối đa 1s.
 - Gửi phản hồi về đơn hàng
 - + Form phản hồi phải đơn giản, đọc vào hiểu ngay.
 - + Thời gian upload phản hồi tối đa là 100ms.
- Đặt bàn:
 - + Sau khi đăng nhập thành công, hiển thị trang chủ của nhà hàng trong vòng tối đa 1s.
 - + Cập nhật bàn trống trong vòng 1s.
 - + Nếu nhiều vị khách hàng cùng đặt bàn cùng 1 thời gian và vị trí thì ưu tiên người đặt trước và báo bàn không còn trống với những người

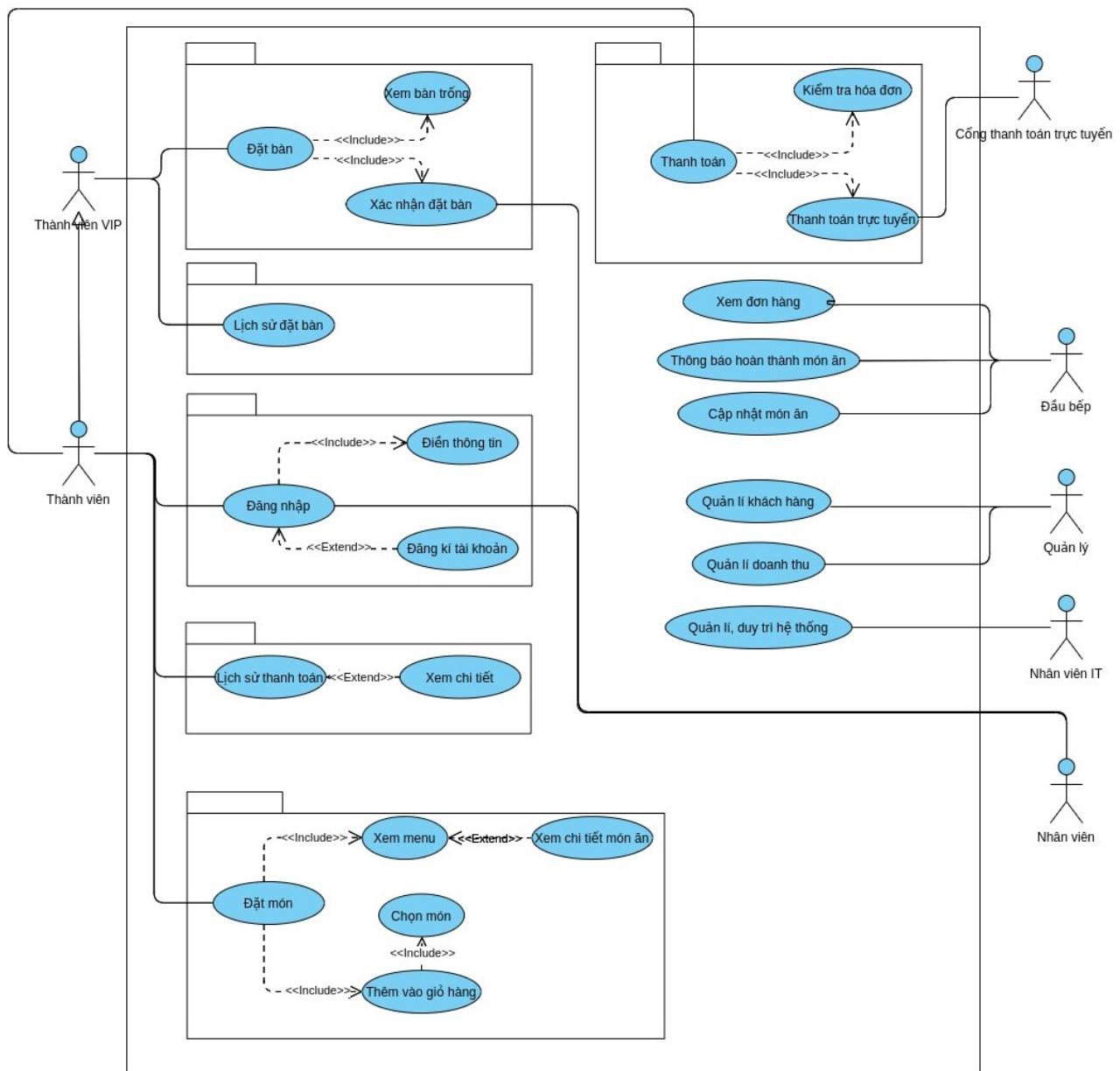
còn lại.

+ Xác nhận đặt bàn thành công tối đa 1s.

- Khi thanh toán:

- Thực hiện trừ vào số dư tài khoản ví điện tử trong vòng dưới 500ms
- Phản hồi thanh toán thành công/thất bại trong vòng dưới 200ms.

1.2.3 Draw a use-case diagram for the whole system



Hình 1.1: Use case diagram of System restaurant Pos 2.0

1.3 Choose one specific feature, i.e. food ordering, tablereservation, customer management. Draw its use-casediagram and describe the use-case using a table format

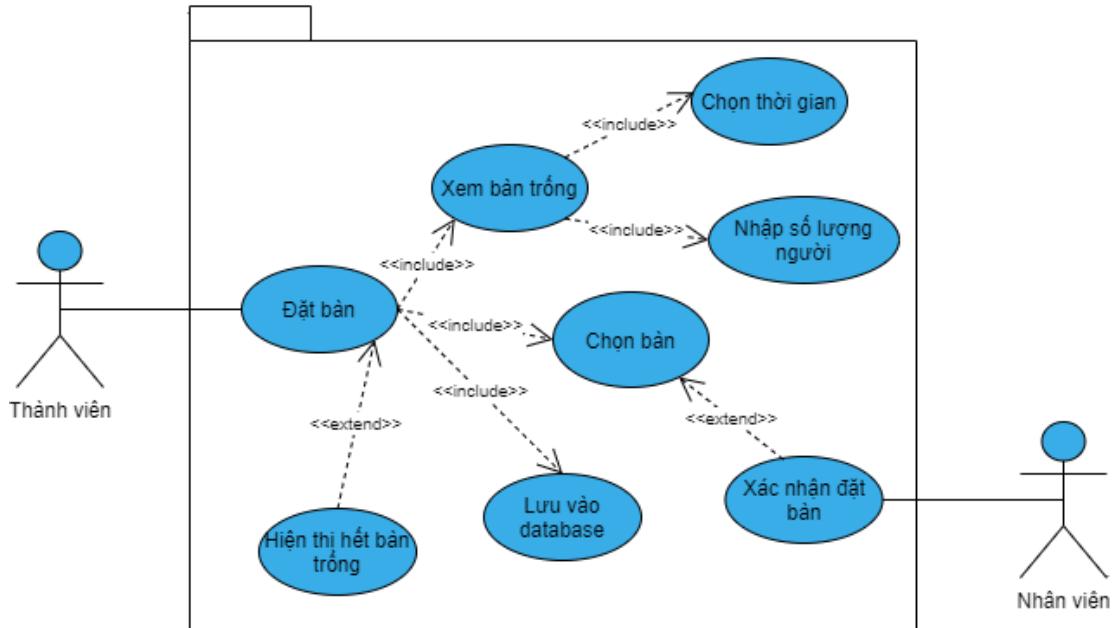
Các thành viên sẽ làm việc theo bảng phân công dưới đây xuyên suốt các task của Bài tập lớn.

Member	Feature
Nguyễn Lê Hiên	Hiển thị menu món ăn, Xem chi tiết món ăn, Thêm món vào giỏ hàng. (Code FrontEnd Trang chủ và Layout)
Nguyễn Văn Thương	Xem lịch sử đặt bàn, Lọc danh sách món theo loại, Giỏ hàng
Lưu Công Định	Đặt bàn
Phùng Minh Khánh	Xem lịch sử thanh toán
Trần Tiến Lộc	Thanh Toán, Lấy lại mật khẩu
Bùi Tiến Bình	Đăng kí, Đăng nhập, Đăng xuất

1.3.1 Use-case : Đặt bàn

Đặt bàn	
Actors	Khách hàng (Thành viên) và nhân viên nhà hàng.
Descriptions	Giúp khách hàng có bàn sẵn khi đến quán ăn, không phải chờ đợi hoặc quay về khi quán đã hết bàn tại thời điểm nào đó.
Triggers	Khách hàng phải là thành viên của nhà hàng (có tài khoản nhà hàng và là thành viên Vip (thanh toán tại nhà hàng trên 10 lần)) và bấm vào nút đặt bàn.
Precondition	Khách hàng đã đăng nhập vào hệ thống, trang web đang hoạt động trong giờ làm việc, và không bảo trì.
Postcondition	<ol style="list-style-type: none"> Đặt bàn thành công, hệ thống lưu trữ bàn đặt cho khách hàng. Đặt bàn không thành công, hiển thị "Xin lỗi quý khách, thời điểm này đã hết bàn trống".
Normal Flow	<ol style="list-style-type: none"> Khách hàng nhấn vào biểu tượng "Đặt bàn". Hệ thống hiển thị form đặt bàn. Khách hàng chọn khung giờ, số lượng người. Khách hàng nhấn nút "Tìm kiếm". Hệ thống hiển thị bàn còn trống, khách hàng nhấn vào bàn mà mình muốn đặt. Khách hàng bấm xác nhận đặt bàn. Nhân viên xác nhận đặt bàn. Hệ thống lưu trữ bàn đặt cho khách hàng, hiển thị đặt bàn hoàn tất.
Alternative Flows	None
Exceptions Flow	<p>Tại bước 5:</p> <p>Cửa hàng đã hết bàn theo yêu cầu của khách hàng. Dưa ra câu hiển thị "Xin lỗi quý khách, thời điểm này đã hết bàn trống".</p> <p>5.1 Khách hàng chọn lại thời gian đặt bàn và số lượng</p> <p>5.1a. Khách hàng chọn đặt lại.</p> <p>5.1b. Quay trở lại bước 3.</p> <p>5.2 Khách hàng kết chọn kết thúc đặt bàn.</p> <p>5.2a. Hệ thống hiển thị "Xin lỗi quý khách, hẹn quý khách lần sau".</p>

	5.2b. Kết thúc use-case.
Comments	None

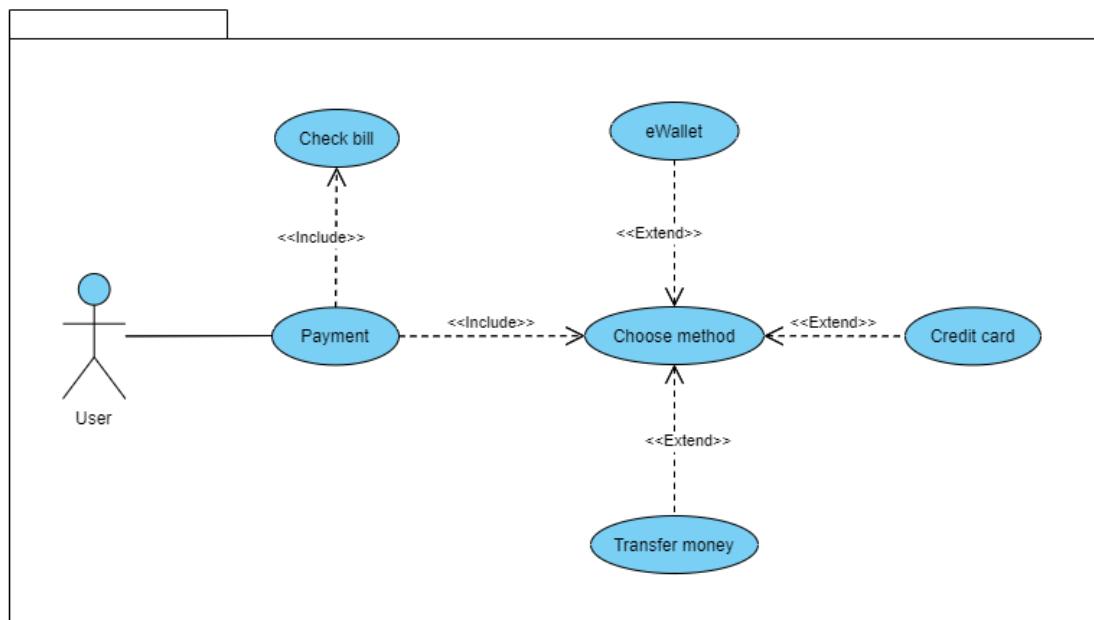


Hình 1.2: Use case diagram of table reservation

1.3.2 Use-case: Thanh toán

Thanh toán	
Actors	Khách hàng
Descriptions	Người dùng có thể sử dụng để thanh toán hóa đơn.
Triggers	Nhấn chọn vào "Thanh toán".
Precondition	Thành viên đã đăng nhập vào hệ thống thành công.
Postcondition	Khách hàng hoàn tất thanh toán hóa đơn.
Normal Flows	<ol style="list-style-type: none"> 1. Sau khi kiểm tra giỏ hàng, khách hàng nhấn vào biểu tượng "Thanh Toán". 2. Hệ thống hiển thị hóa đơn, khách hàng kiểm tra lại hóa đơn. 3. Khách hàng chọn phương thức thanh toán. 4. Người dùng hoàn thiện thông tin cá nhân. 5. Khách hàng chọn xác nhận thanh toán và nhập mã pin để xác thực. 6. Hệ thống lưu record đơn hàng vào database, đồng thời phản hồi thanh toán thành công. 7. Hoàn tất thanh toán.
Alternative Flows	<p><i>Tại bước 3:</i> Khách hàng muốn thanh toán trực tiếp bằng tiền mặt.</p> <ol style="list-style-type: none"> 3.1. Gửi yêu cầu thanh toán bằng hình thức trực tiếp qua Boxchat tới nhân viên. 3.2. Khách hàng thực hiện thanh toán. 3.3. Thanh toán hoàn tất và dữ liệu được nhân viên lưu vào database.

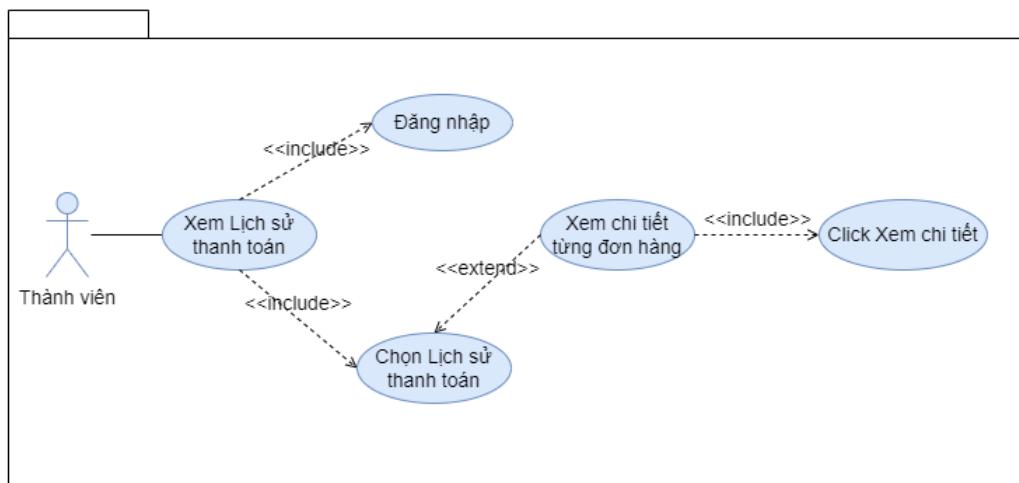
Exceptions	Tại bước 2: Hóa đơn không chính xác
Flows	<p>2.1. Gửi yêu cầu nhân viên kiểm tra lại thông tin hóa đơn.</p> <p>2.2. Hóa đơn đã được chỉnh sửa và tiếp tục bước 3.</p> <p>Tại bước 4: Loại ví điện tử đã chọn hiện đang bảo trì hệ thống.</p> <p>4.1. Ứng dụng hiển thị "Thanh toán thất bại do hệ thống ví điện tử đang được bảo trì".</p> <p>4.2. Hệ thống yêu cầu người dùng chọn lại phương thức thanh toán và tiếp tục bước 4 trong Normal Flows.</p>
	Tại bước 5: Số dư trong ví không đủ để thanh toán hóa đơn.
Comments	None



Hình 1.3: Use case diagram of Payment

1.3.3 Use-case : Lịch sử thanh toán

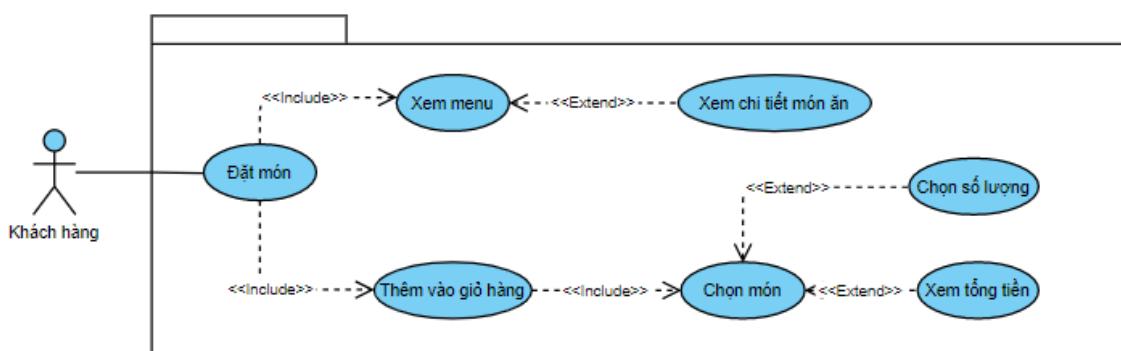
Lịch sử thanh toán	
Actors	Thành viên
Descriptions	Thành viên xem lại tất cả hoá đơn đã thanh toán và có thể xem lại chi tiết của mỗi hoá đơn.
Triggers	Thành viên cần phải có tài khoản.
Precondition	Hệ thống kích hoạt Usecase Đăng nhập.
Postcondition	Thành viên xem lịch sử các lần đã Thanh toán ở nhà hàng và chi tiết mỗi đơn hàng.
Normal Flows	<ol style="list-style-type: none"> Thành viên Đăng nhập thành công Thành viên chọn click vào "Lịch sử thanh toán" trên màn hình giao diện. Hệ thống sẽ hiển thị danh sách kết quả theo yêu cầu của thành viên theo thứ tự mới tới cũ. Thành viên click "Xem chi tiết" trên màn hình giao diện. Hệ thống sẽ hiển thị danh sách món ăn và tổng tiền thanh toán của hoá đơn đó. Usecase kết thúc.
Alternative Flows	None
Exceptions Flows	Tại bước 1: Hệ thống bảo trì
Comments	1,1. Hệ thống thông báo bảo trì.



Hình 1.4: Use case diagram of Lịch sử thanh toán

1.3.4 Use case : Xem menu(Xem chi tiết món ăn) và Thêm món ăn vào giỏ hàng

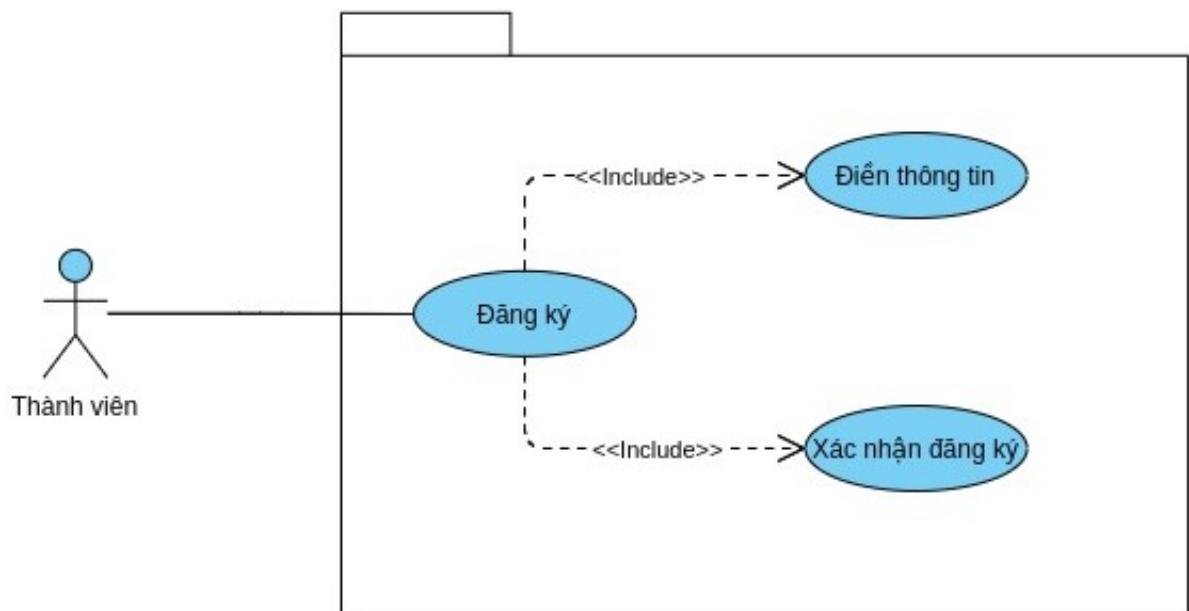
Xem menu(Xem chi tiết món ăn) và Thêm món ăn vào giỏ hàng	
Actors	Khách hàng
Descriptions	Khách hàng thêm món ăn vào giỏ hàng.
Triggers	Khách hàng bấm vào menu
Precondition	None
Postcondition	Thêm món ăn vào giỏ hàng thành công
Normal Flows	<p>1. Khách hàng xem menu.</p> <p>2. Khách hàng bấm vào ADD TO CART tại món ăn mà mình chọn để thêm món ăn vào giỏ hàng với số lượng món ăn là 1.</p> <p>3. Hệ thống tiến hành thêm món ăn vào giỏ hàng cho khách.</p> <p>4. Hoàn tất thêm món ăn vào giỏ hàng</p>
Alternative Flows	<p>Tại bước 2:</p> <p>2.1. Khách hàng nhập vào món ăn sẽ hiển thị form thông tin món ăn.</p> <p>2.1a. Trên form thông tin món ăn, khách hàng có thể xem thông tin chi tiết về món ăn, chọn số lượng ...</p> <p>2.1b. Nếu khách hàng vừa ý, chọn số lượng cần đặt, bấm nút thêm vào giỏ hàng. Use case tiếp tục bước 3.</p> <p>2.1c. Ngược lại, khách hàng bấm dấu "X", Use case quay lại bước 1.</p>
Exceptions Flows	<p>Tại bước 3:</p> <p>3.1. Chưa đăng nhập</p> <p>3.1a. Hệ thống quay về trang đăng nhập.</p> <p>3.1b. Đăng nhập vào hệ thống.</p> <p>3.1c. Quay lại bước 1.</p>
Comment	Khách hàng có thể hủy món đã chọn trong giỏ hàng trước khi thanh toán nếu không muốn chọn nữa.



Hình 1.5: Use case diagram của Xem menu(Xem chi tiết món ăn) và Thêm món ăn vào giỏ hàng

1.3.5 Use case : Đăng ký

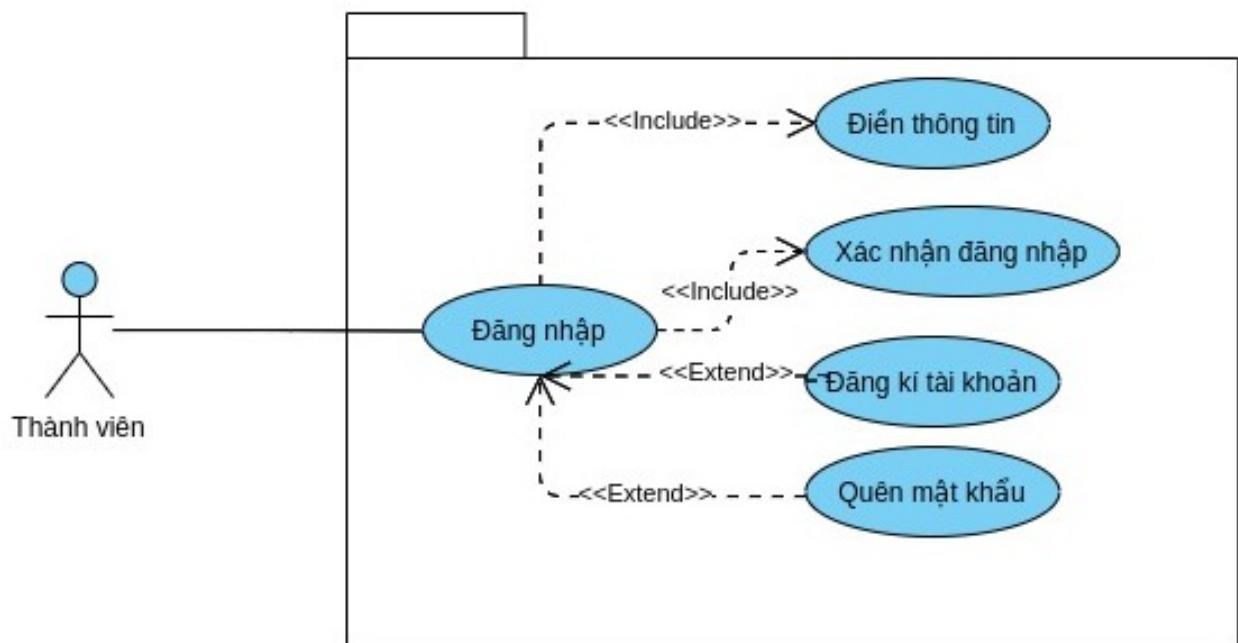
Đăng Ký	
Actors	Khách hàng, Nhân viên
Descriptions	Khách hàng muốn đăng ký tài khoản để trở thành thành viên và nhận được nhiều thông báo mới về nhà hàng và ưu đãi từ nhà hàng..
Triggers	Người dùng nhấn chọn đăng ký từ trang đăng nhập.
Precondition	Truy cập được vào trang chủ hệ thống.
Postcondition	Đăng kí tài khoản thành công.
Normal Flows	<ol style="list-style-type: none"> 1. Người dùng bấm vào "đăng ký". 2. Hệ thống hiển thị form đăng ký tài khoản. 3. Người dùng nhập thông tin tài khoản. 4. Bấm vào nút đăng ký. 5. Hệ thống kiểm tra thông tin đăng ký. 6. Hệ thống lưu thông tin tài khoản vào database 7. Hệ thống xác thực đăng kí thành công và cho phép khách hàng đăng nhập vào hệ thống. 8. Kết thúc use case.
Alternative Flows	<p><i>Tại bước 6:</i></p> <ol style="list-style-type: none"> 6. Do thông tin tài khoản không hợp lệ hoặc bị trùng nên hệ thống hiển thị lỗi tương ứng. 7. Quay lại bước 3
Exceptions Flows	<p><i>Tại bước 1::</i></p> <ol style="list-style-type: none"> 1. Hiện hệ thống đang bảo trì, nâng cấp. 2. Hệ thống hiển thị thông báo: Hệ thống đang bảo trì, vui lòng truy cập sau. 3. Người dùng thoát ra khỏi hệ thống 4. Kết thúc use case. <p><i>Tại bước 3: 3. Khách hàng click vào tùy chọn "Trở lại".</i></p> <ol style="list-style-type: none"> 4. Dưa khách hàng về trang đăng nhập. 5. Kết thúc use case.
Comment	None



Hình 1.6: Use case Đăng ký

1.3.6 Use case : Đăng nhập

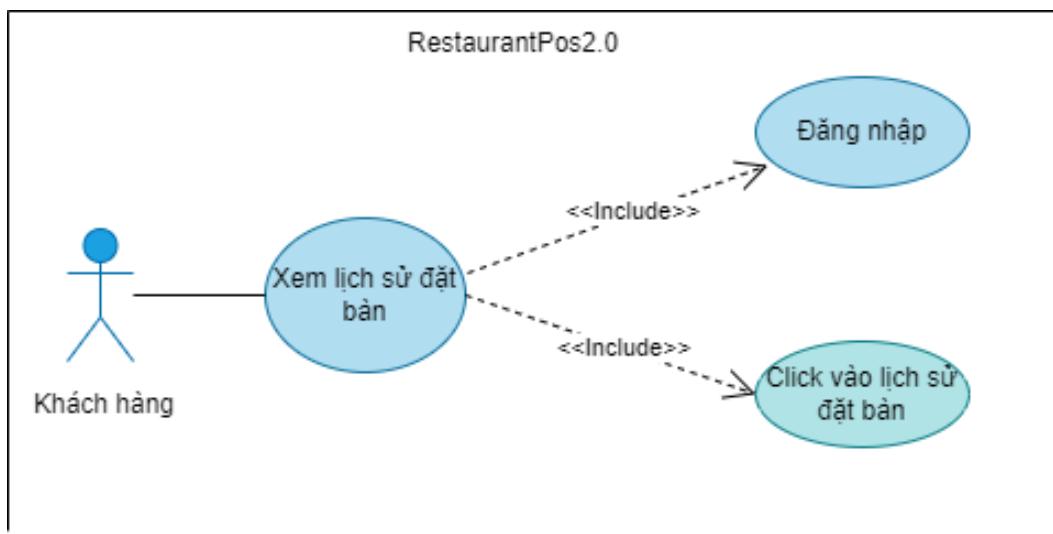
Đăng nhập	
Actors	Khách hàng, Nhân viên
Descriptions	Use case cho phép người dùng đăng nhập vào hệ thống để thực hiện những chức năng của mình.
Triggers	Người dùng chọn chức năng đăng nhập từ trang chủ của hệ thống.
Precondition	Truy cập được vào trang chủ hệ thống.
Postcondition	Đăng nhập thành công
Normal Flows	<p>1. Người dùng bấm vào "Đăng nhập"</p> <p>2. Hệ thống hiện form đăng nhập.</p> <p>3. Người dùng nhập tên tài khoản và mật khẩu.</p> <p>4. Bấm vào nút Đăng nhập.</p> <p>5. Hệ thống kiểm tra thông tin tài khoản.</p> <p>6. Hệ thống thông báo đăng nhập thành công và đưa người dùng về trang chủ.</p> <p>7. Kết thúc use case.</p>
Alternative Flows	<p>Tại bước 3:</p> <p>3. Khách hàng click vào tùy chọn "Trở lại".</p> <p>4. Dưa khách hàng về trang menu.</p> <p>5. Kết thúc use case.</p>
Exceptions Flows	<p>Tại bước 1::</p> <p>1. Hiện hệ thống đang bảo trì, nâng cấp.</p> <p>2. Hệ thống hiển thị thông báo: Hệ thống đang bảo trì, vui lòng truy cập sau.</p> <p>3. Người dùng thoát ra khỏi hệ thống</p> <p>4. Kết thúc use case.</p> <p>Tại bước 6:</p> <p>6. Hệ thống thông báo lỗi do thông tin tài khoản không hợp lệ hoặc không tồn tại.</p> <p>7. Quay lại bước 3</p>
Comment	None



Hình 1.7: Use case Đăng nhập

1.3.7 Use-case : Xem lịch sử đặt bàn

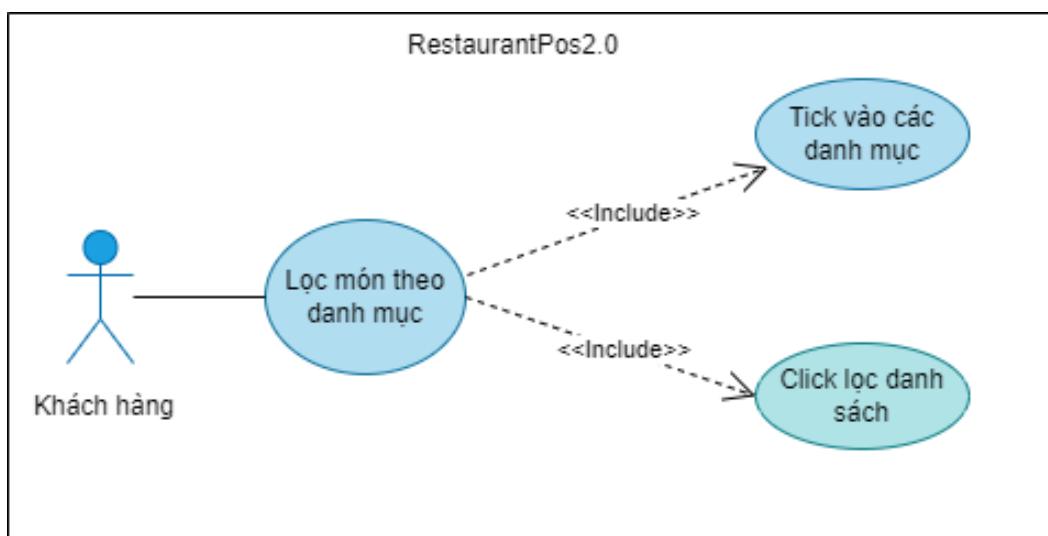
Xem lịch sử đặt bàn	
Actors	Khách hàng
Descriptions	Khách hàng có thể xem lại toàn bộ các lần đặt bàn tại nhà hàng với các mốc thời gian cụ thể .
Triggers	Khách hàng chọn vào mục "Lịch sử đặt bàn"
Precondition	Khách hàng phải đăng nhập thành công vào hệ thống .
Postcondition	Khách hàng xem lịch sử các lần đặt bàn tại nhà hàng.
Normal Flow	<ol style="list-style-type: none"> Khách hàng click vào "Lịch sử đặt bàn" trên giao diện. Hệ thống hiển thị kết quả trả về đến khách hàng sắp xếp theo thời gian gần nhất với thời điểm hiện tại. Kết thúc Usecase.
Alternative Flows	<p>Tại bước 1: Kiểm tra đăng nhập</p> <ol style="list-style-type: none"> 1.1 Hệ thống kiểm tra khách hàng đã đăng nhập chưa. 1.1.a Chưa đăng nhập. Chuyển hướng đến trang đăng nhập. 1.1.b Đã đăng nhập. Usecase tiếp tục bước 2.
Exceptions Flow	<p>Tại bước 1: Hệ thống đang bảo trì .</p> <ol style="list-style-type: none"> 1.1. Hệ thống hiển thị thông báo ra màn hình. 1.2 Kết thúc Usecase
Comments	None



Hình 1.8: Usecase diagram xem lịch sử đặt bàn

1.3.8 Use-case : Lọc danh sách món theo danh mục

Lọc danh sách món theo danh mục	
Actors	Khách hàng
Descriptions	Khách hàng có thể xem danh sách các món ăn theo loại hoặc lọc lại những loại món ăn theo sở thích.
Triggers	Khách hàng phải nhấn vào "Lọc danh sách".
Precondition	Khách hàng đã truy cập vào hệ thống.
Postcondition	Khách hàng xem danh sách món ăn theo các danh mục đã lọc.
Normal Flow	<ol style="list-style-type: none"> 1. Khách hàng tick vào các checkbox danh mục món trên giao diện. 2 Khách hàng click vào button "Lọc danh sách" 3. Hệ thống hiển thị kết quả danh sách món trả về đến khách hàng sắp xếp theo các danh mục đã chọn. 4. Usecase kết thúc.
Alternative Flows	<p>Tại bước 2: Khách hàng không tick vào danh mục nào trước khi click button "Lọc danh sách".</p> <ol style="list-style-type: none"> 2.1. Hệ thống mặc định chọn toàn bộ danh mục. 2.2 Use case tiếp tục bước 3.
Exceptions Flow	<p>Tại bước 1: Hệ thống đang bảo trì.</p> <ol style="list-style-type: none"> 1.1. Hệ thống hiển thị thông báo ra màn hình. 1.2. Kết thúc Usecase.
Comments	None



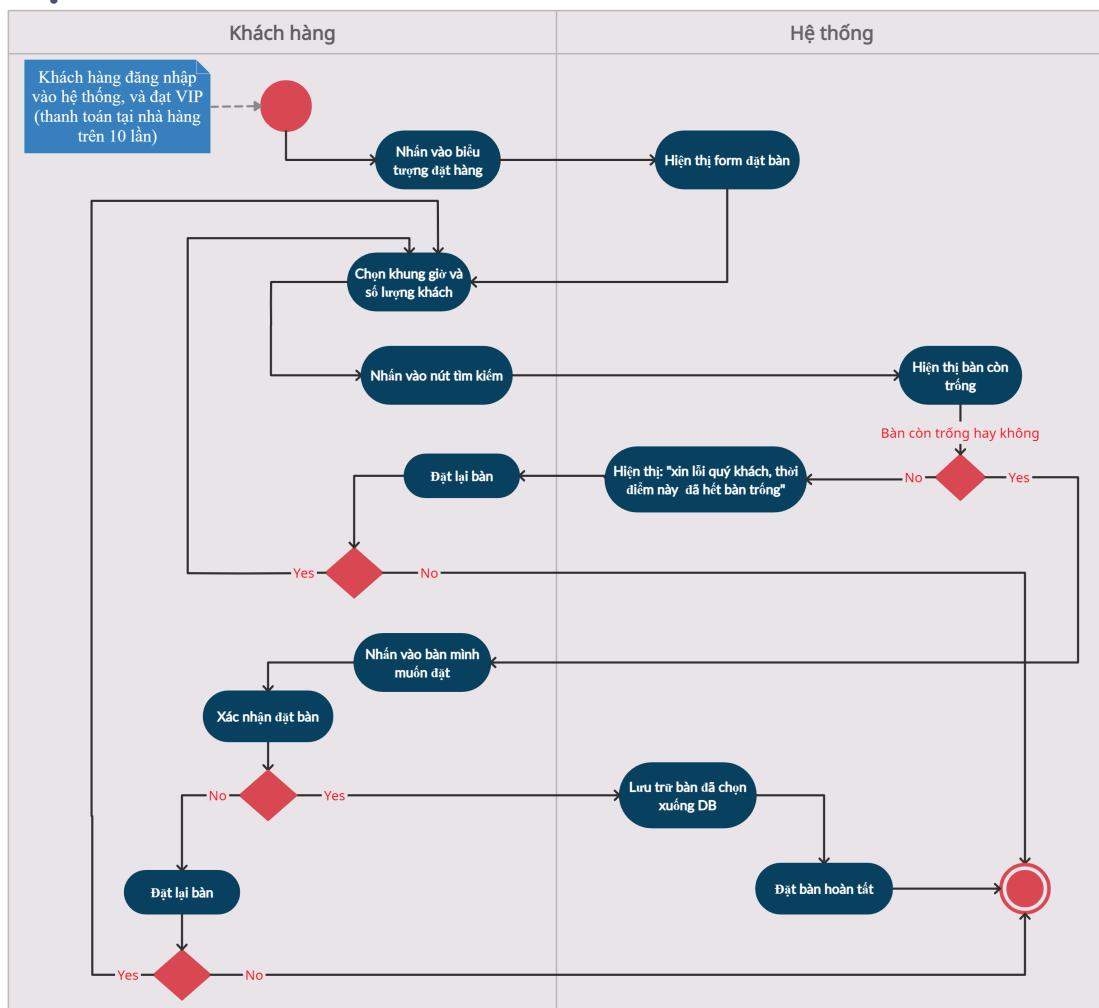
Hình 1.9: Use casediagram lọc danh sách món theo danh mục

2 System modelling

2.1 Activity diagram to capture Major (not all) functional requirements of the desired system.

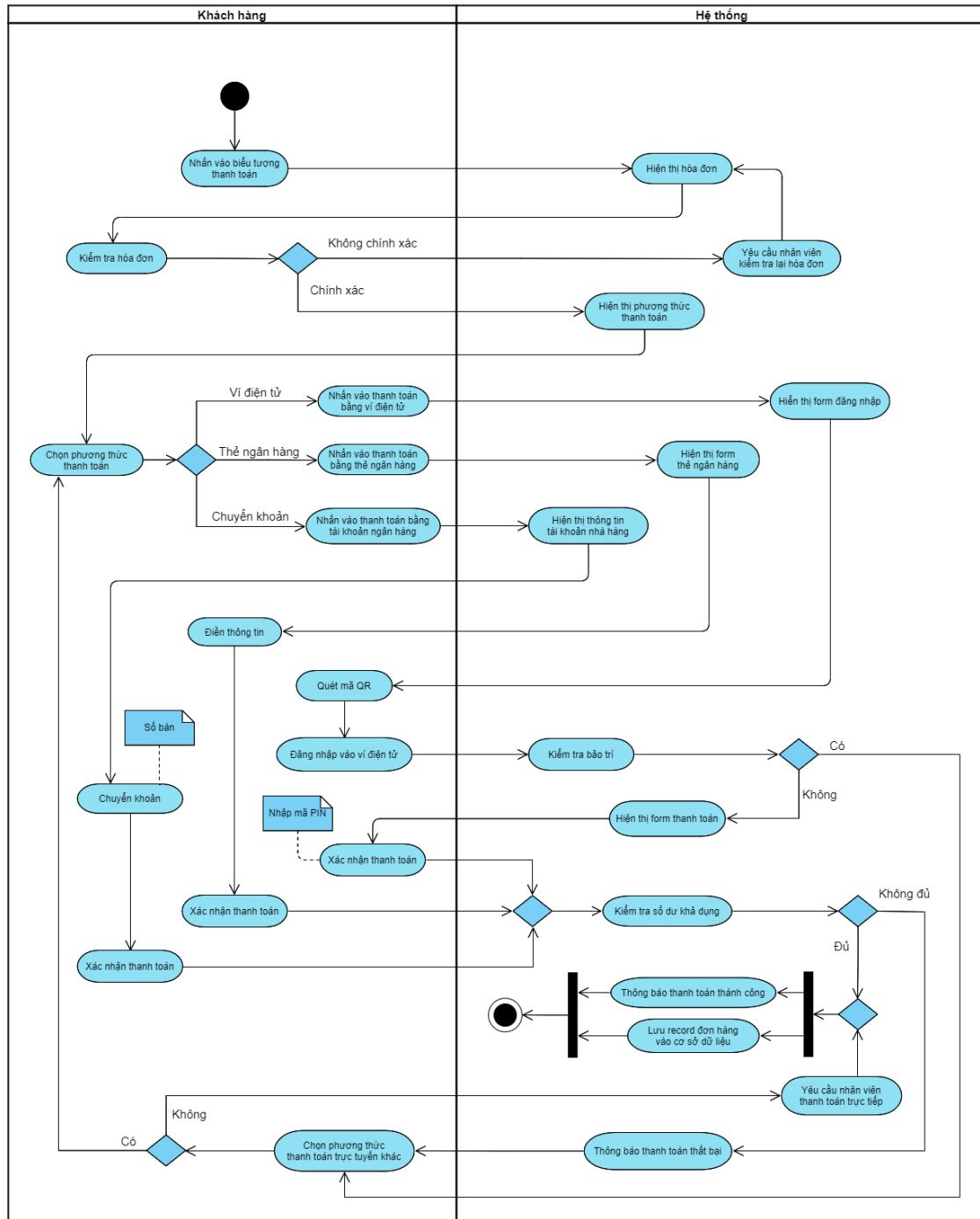
2.1.1 Đặt bàn

Đặt bàn



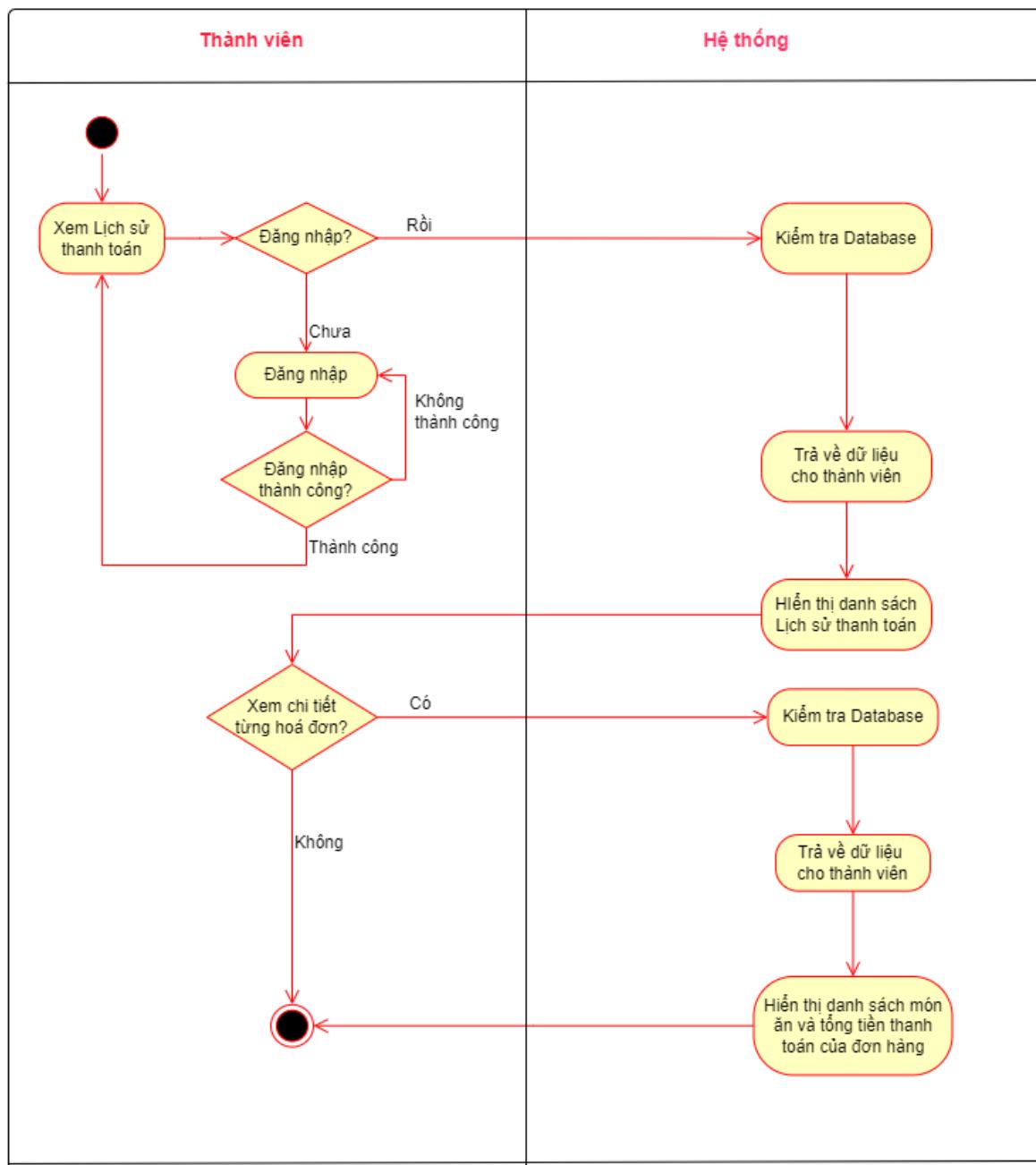
Hình 2.1: Activity diagram của use case: Đặt bàn

2.1.2 Thanh toán



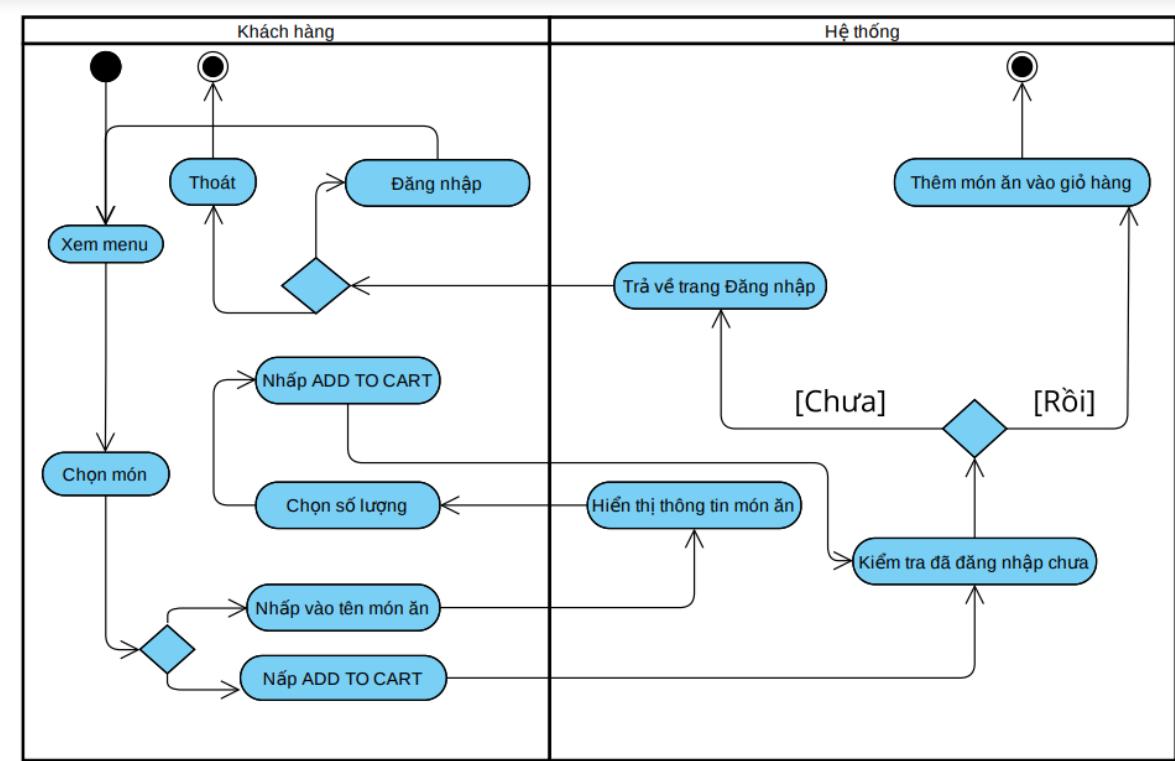
Hình 2.2: Activity diagram của use case: Thanh toán

2.1.3 Lịch sử thanh toán



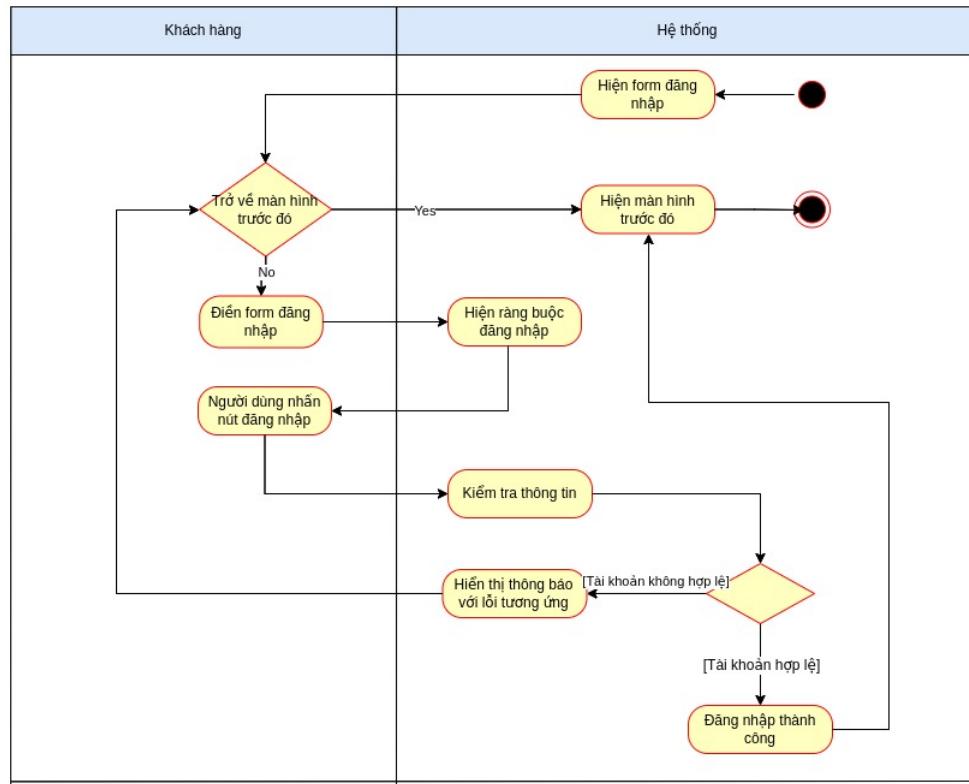
Hình 2.3: Activity diagram của use case: Lịch sử thanh toán

2.1.4 Xem menu, xem chi tiết món ăn và thêm món ăn vào giỏ hàng



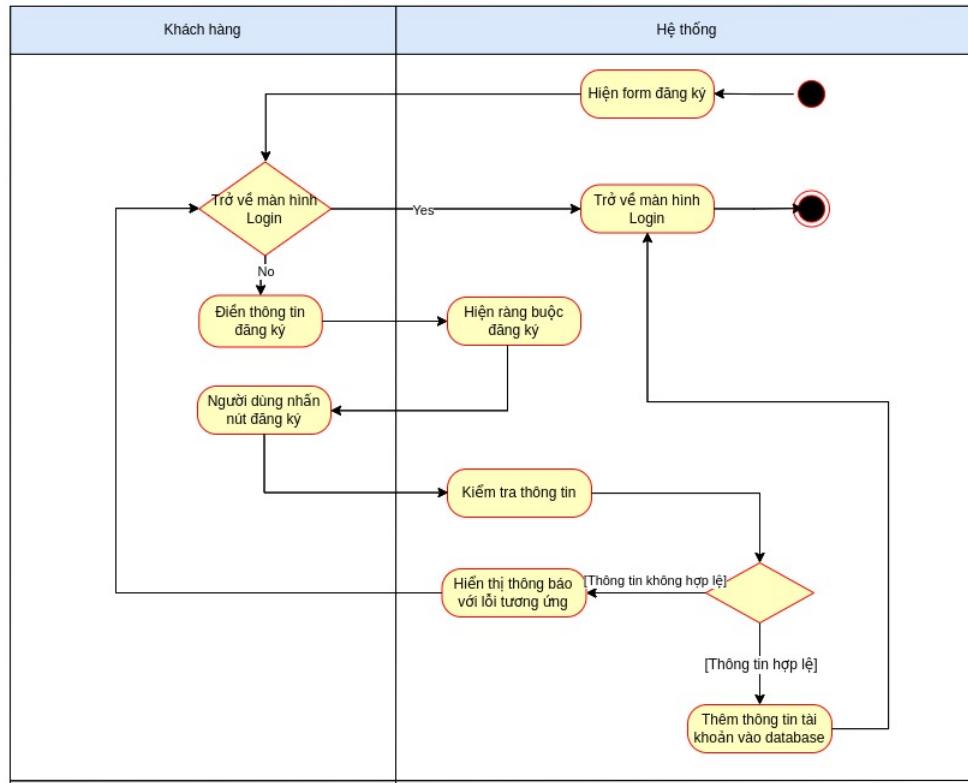
Hình 2.4: Activity diagram của use case: Xem menu, xem chi tiết món ăn và thêm món ăn vào giỏ hàng

2.1.5 Đăng nhập



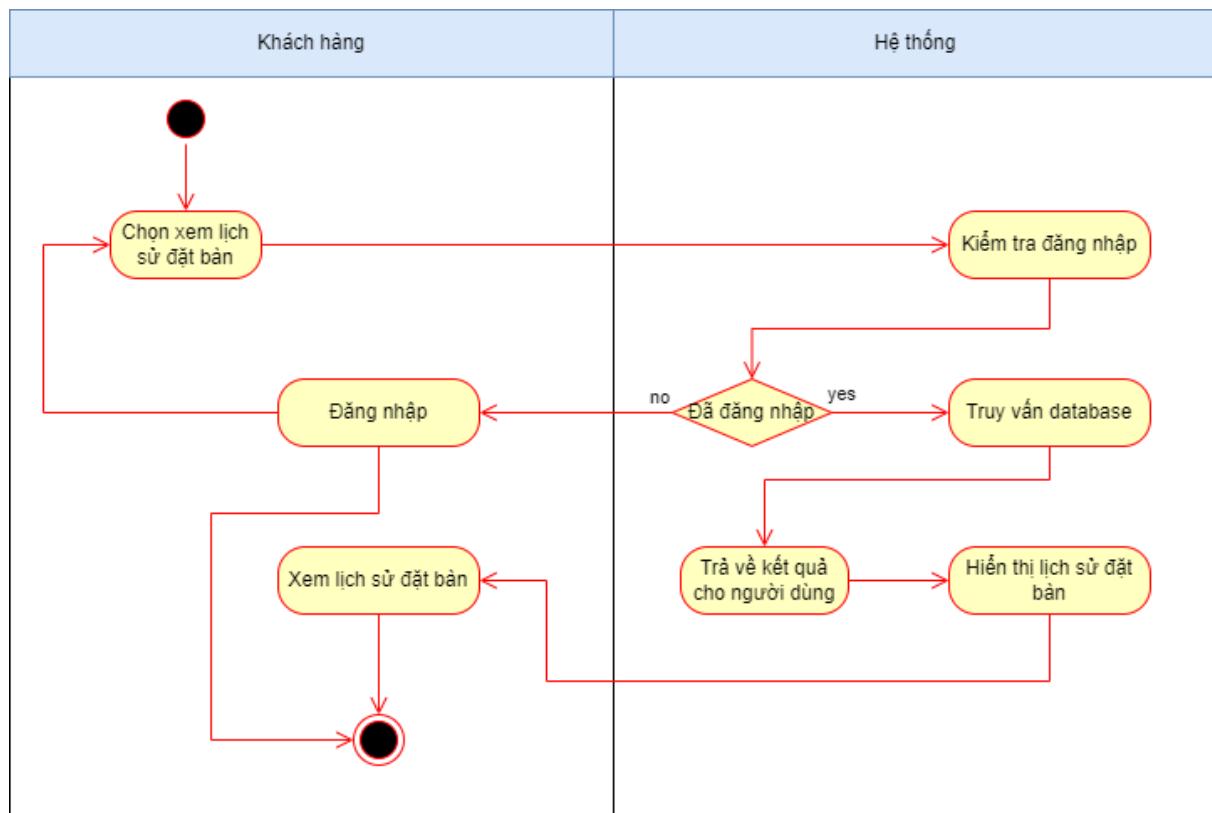
Hình 2.5: Activity diagram của use case: Đăng nhập

2.1.6 Đăng ký



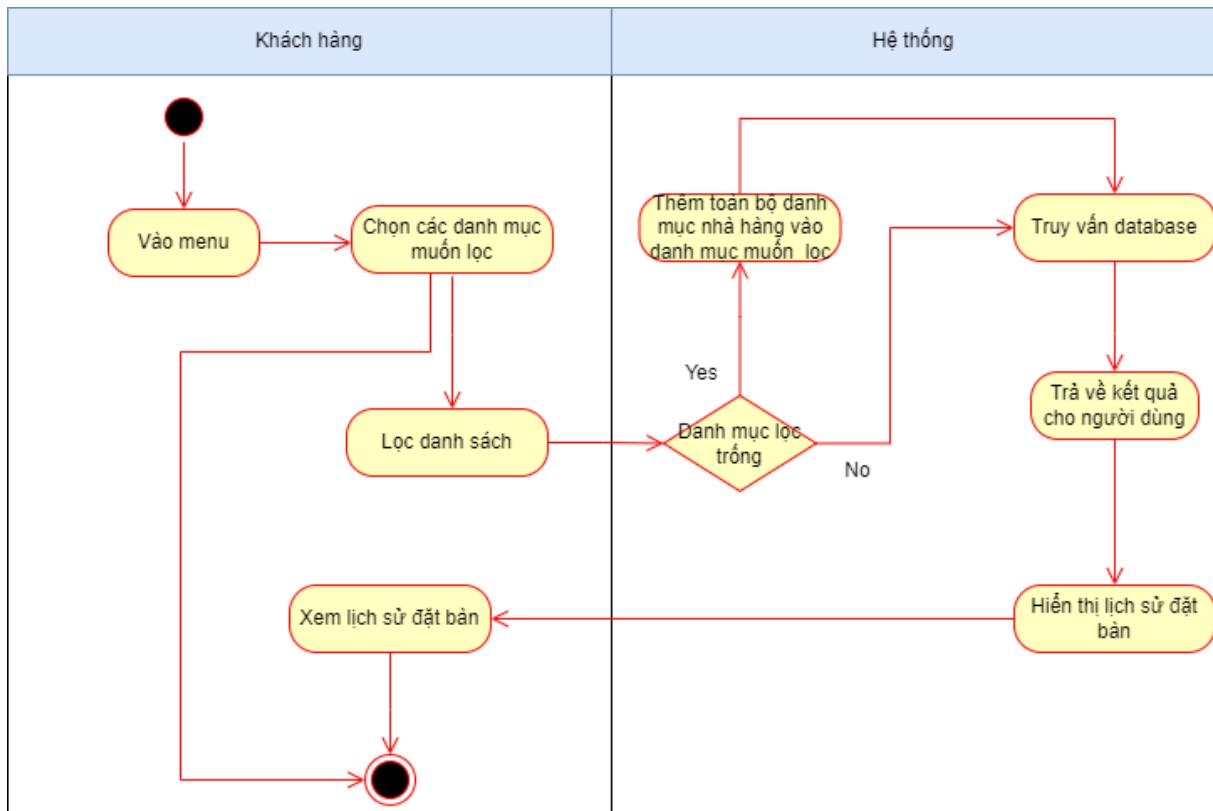
Hình 2.6: Activity diagram của use case: Đăng ký

2.1.7 Lịch sử đặt bàn



Hình 2.7: Activity diagram của use case: Lịch sử đặt bàn

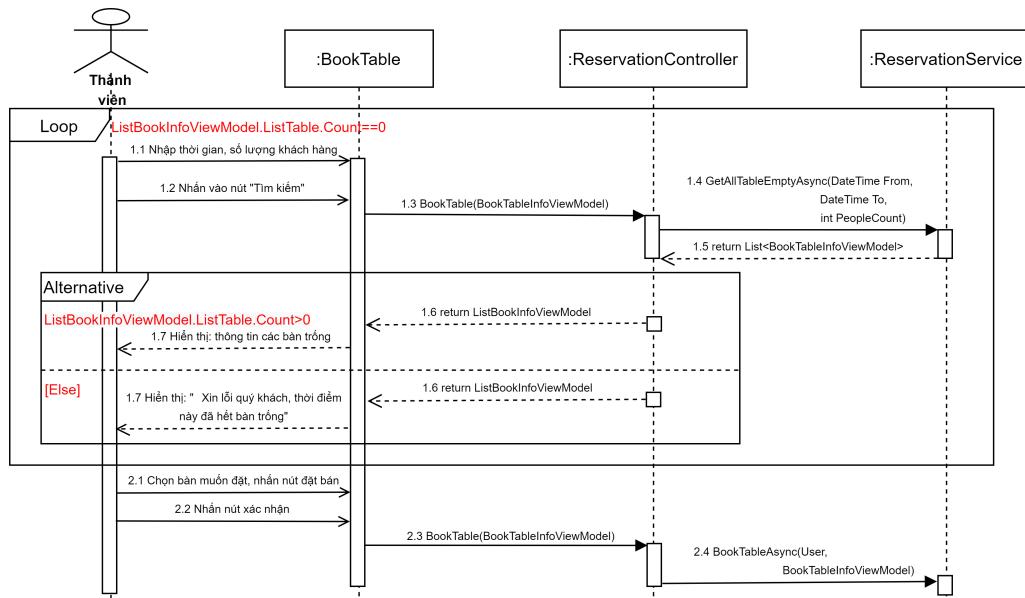
2.1.8 Lọc danh sách món theo danh mục



Hình 2.8: Activity diagram của use case: Lọc danh sách món theo danh mục

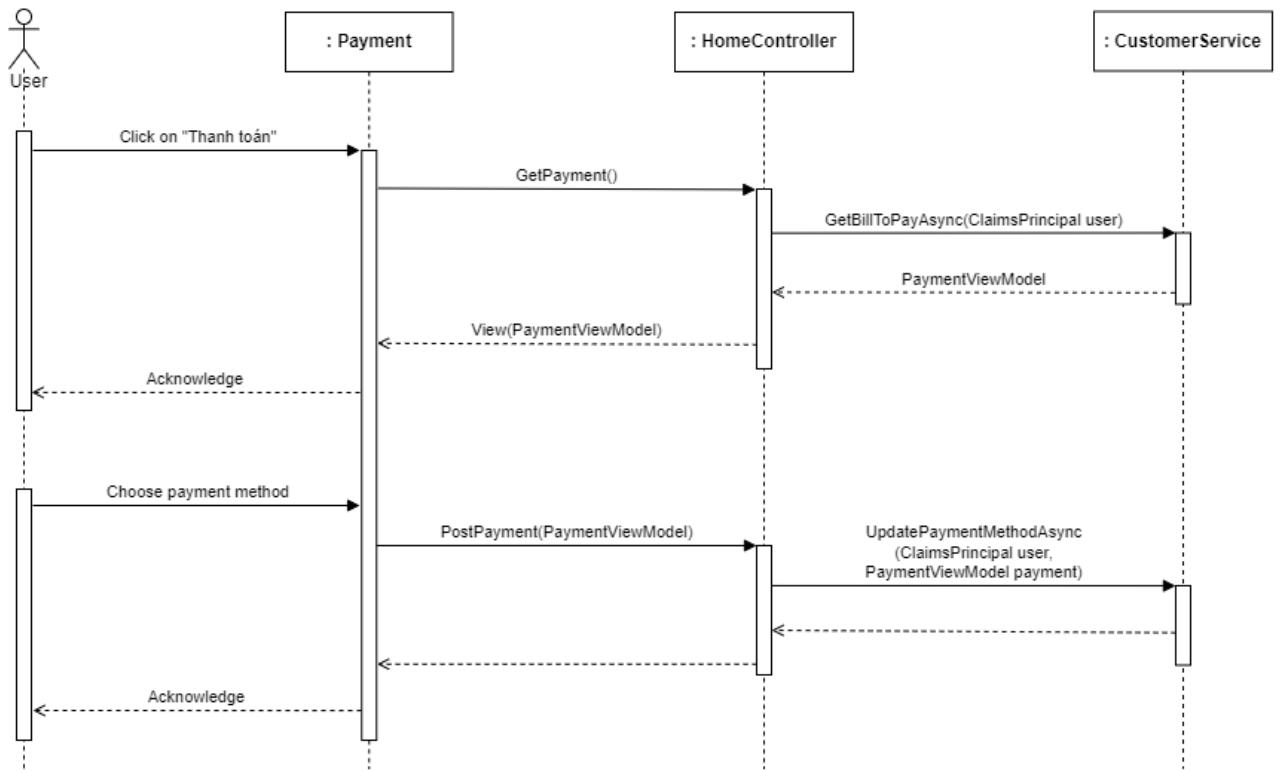
2.2 Sequence diagram for use-case in Task 1.3.

2.2.1 Đặt bàn



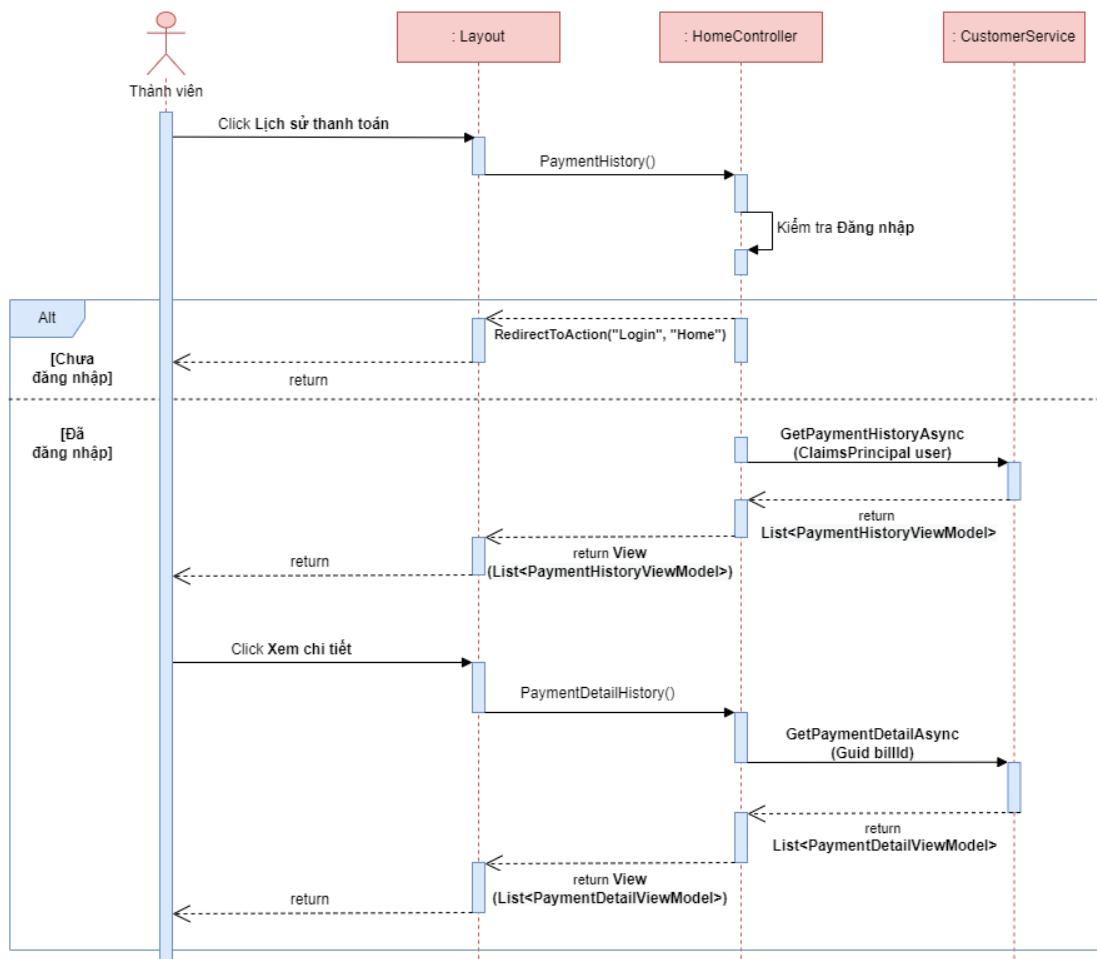
Hình 2.9: Sequence diagram của use case: Reservation

2.2.2 Thanh toán



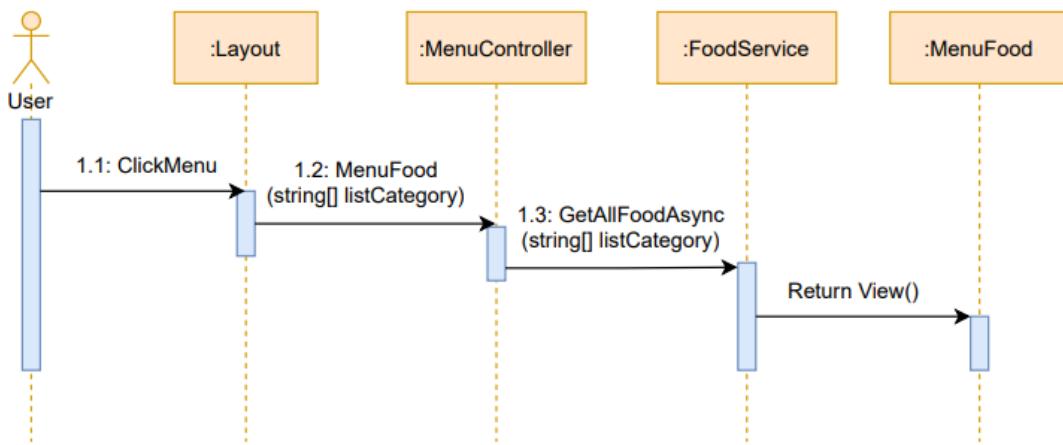
Hình 2.10: Sequence diagram của use case: Payment

2.2.3 Lịch sử thanh toán

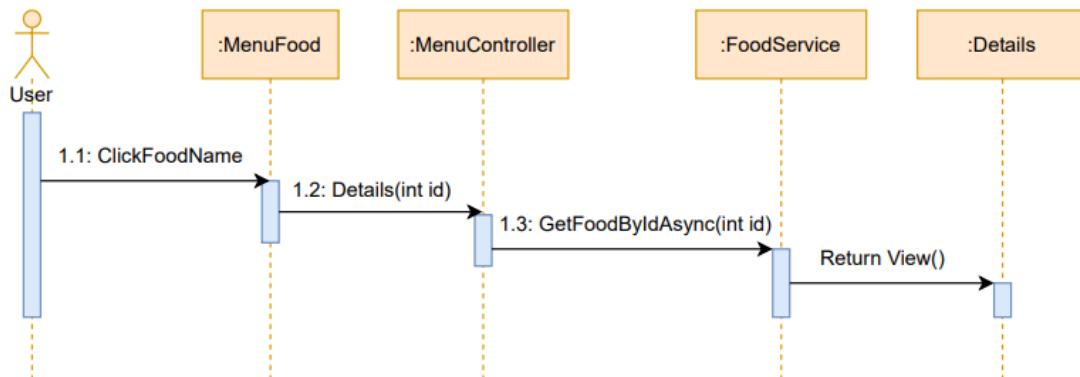


Hình 2.11: Sequence diagram của use case: Lịch sử thanh toán

2.2.4 Xem menu và xem chi tiết món ăn



Hình 2.12: Sequence diagram của use case: Xem menu



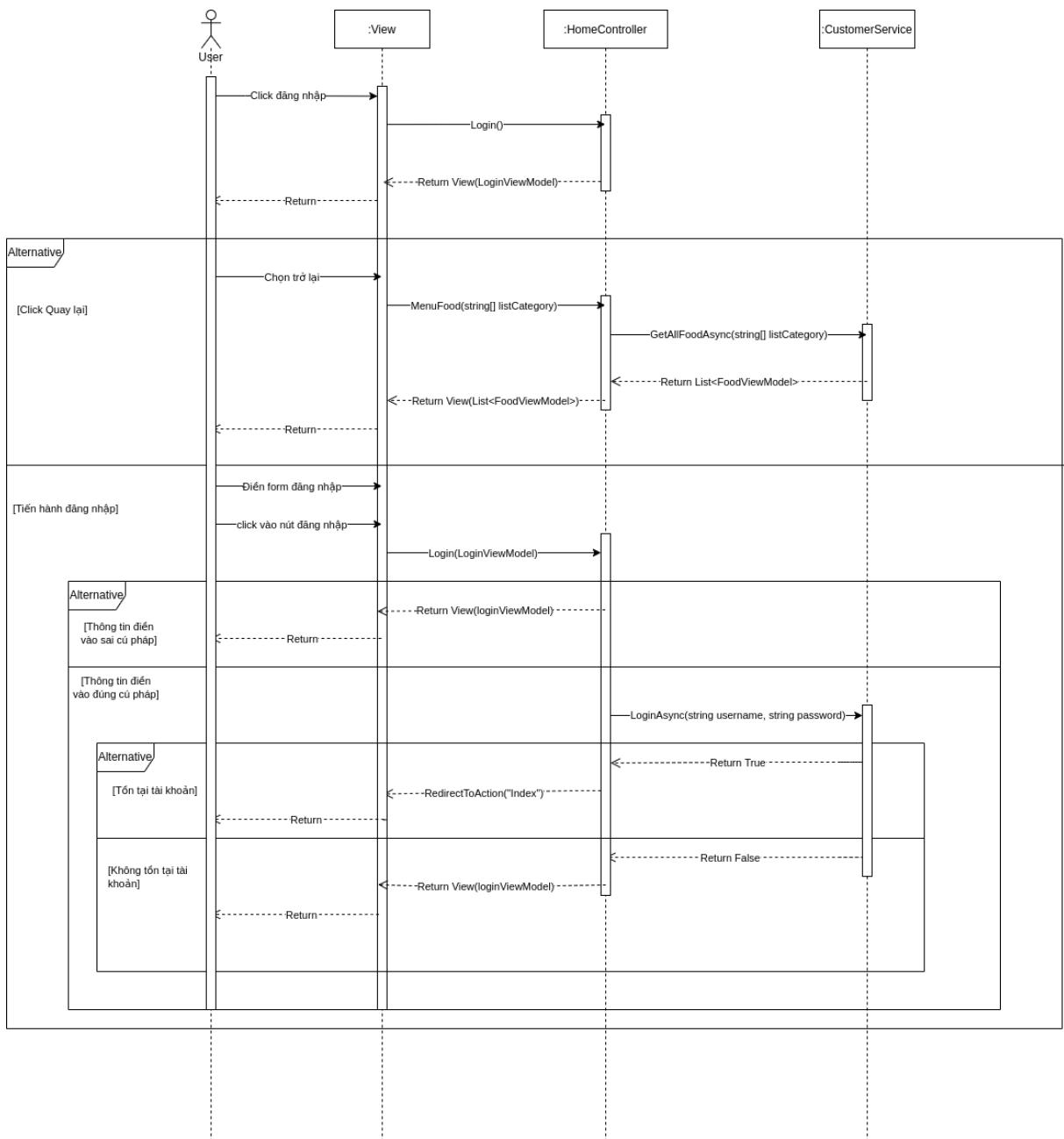
Hình 2.13: Sequence diagram của use case: Xem chi tiết món ăn

2.2.5 Thêm món ăn vào giỏ hàng



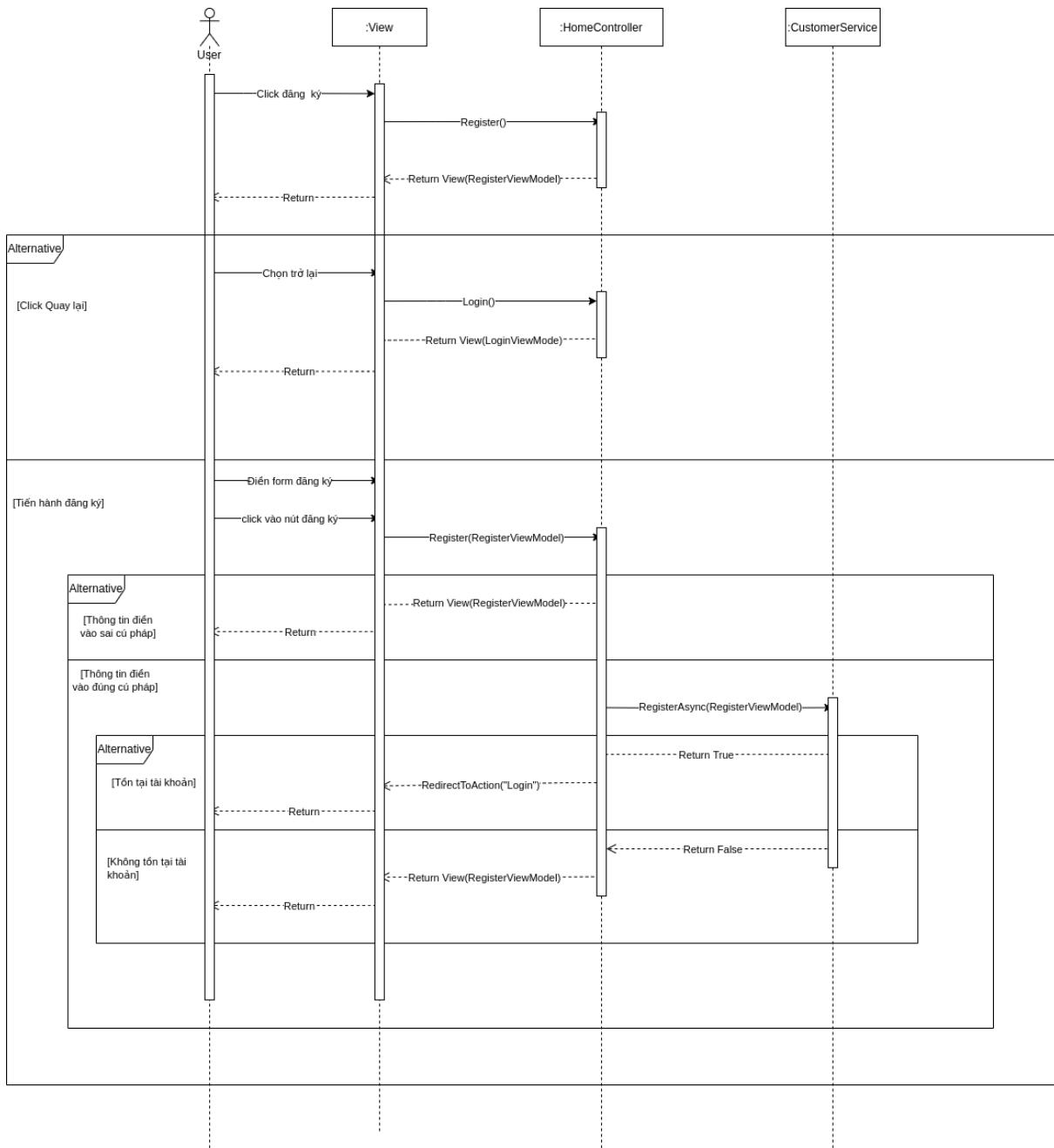
Hình 2.14: Sequence diagram của use case: Thêm món ăn vào giỏ hàng

2.2.6 Đăng nhập



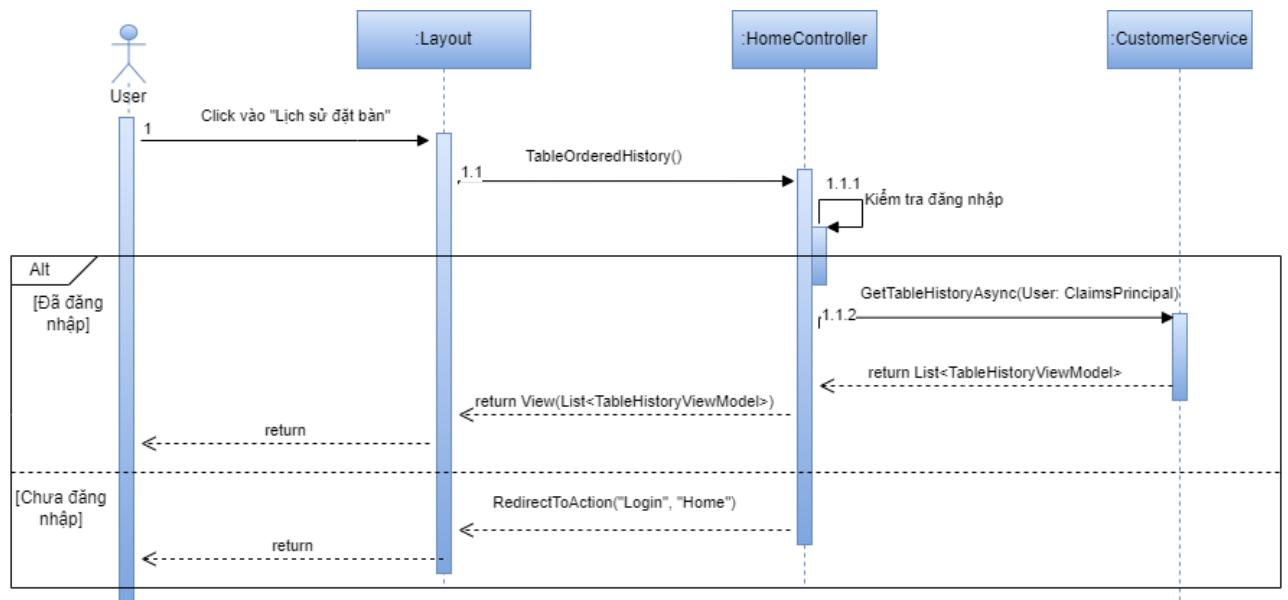
Hình 2.15: Sequence diagram của use case: Đăng nhập

2.2.7 Đăng ký



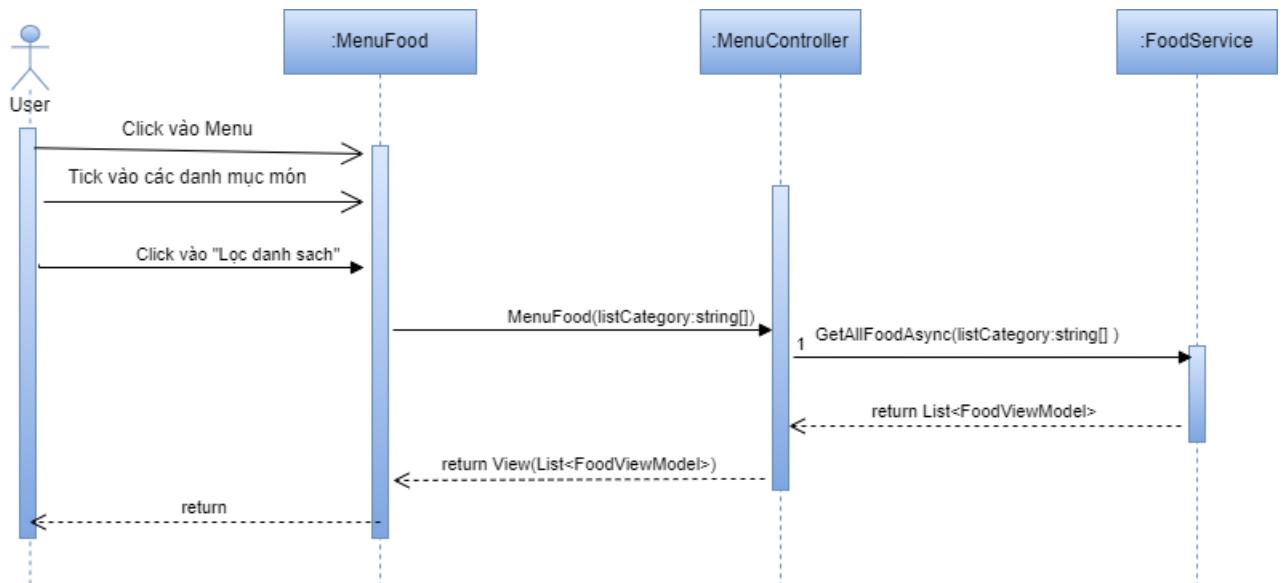
Hình 2.16: Sequence diagram của use case: Đăng ký

2.2.8 Lịch sử đặt bàn



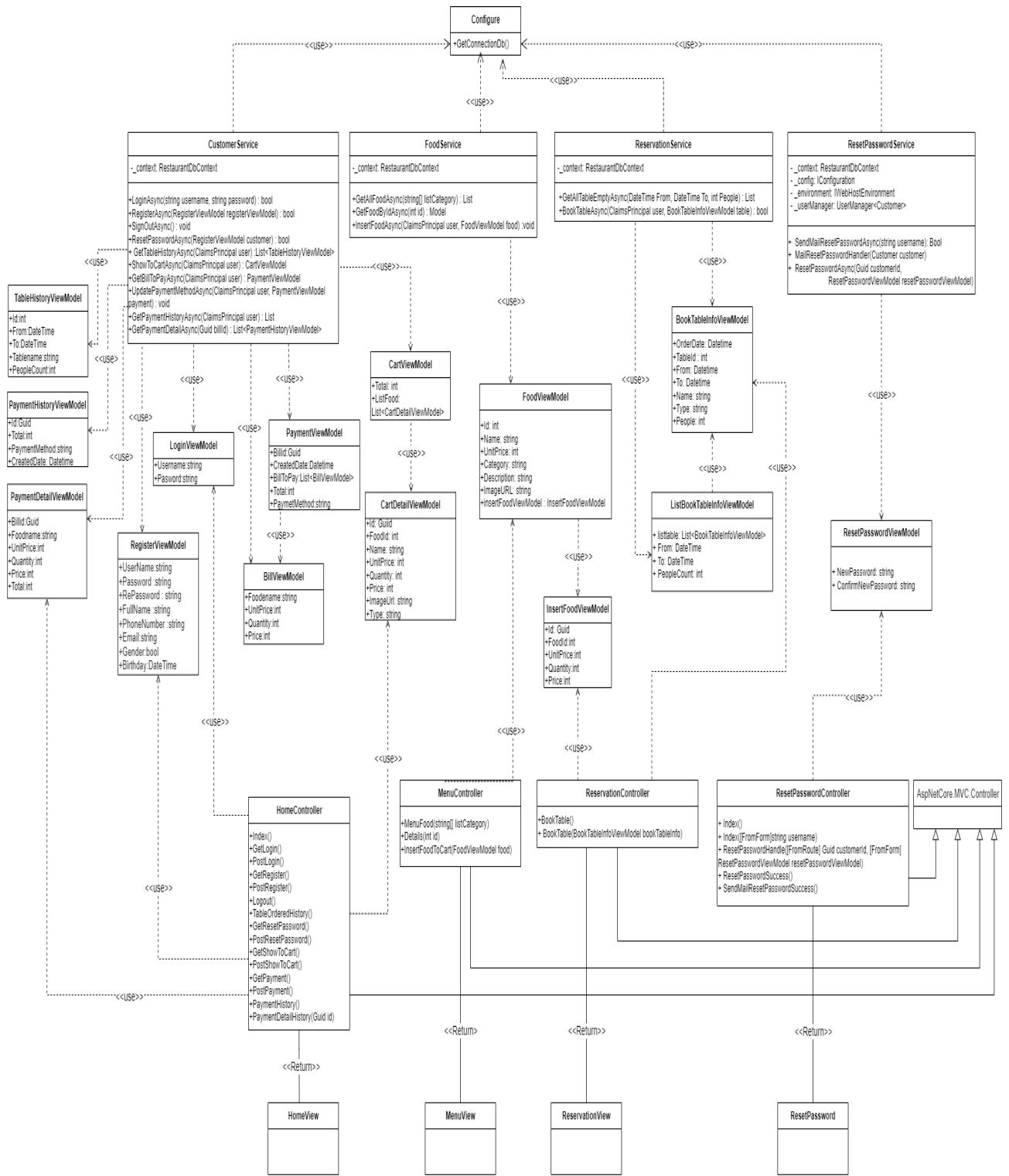
Hình 2.17: Sequence diagram của use case: Lịch sử đặt bàn

2.2.9 Lọc danh sách món theo danh mục



Hình 2.18: Sequence diagram của use case: Lọc danh sách món theo danh mục

2.3 Class Diagram



Hình 2.19: Class diagram của toàn hệ thống

Link class diagram <https://drive.google.com/file/d/1mmftqeUzWBv-iz-p-OG0qaF37QMMCgiF/view?usp=sharing>

3 Architecture design

3.1 Describe an architectural approach you will use to implement the desired system

3.1.1 Database Management

- Sử dụng SQL Sever

- Lý do chọn SQL server để quản lý cơ sở dữ liệu
 - + Dễ sử dụng, có thể sử dụng online để kết nối các thành viên lại giúp dễ dàng làm công việc nhóm một cách dễ dàng.
 - + Phù hợp với các thành viên đều sử dụng hệ điều hành Window.
 - + Tích hợp chặt chẽ với .Net Core, Framework được sử dụng làm Back End

3.1.2 Back End

- Sử dụng .Net core với mô hình MVC

- Những thành phần chính của kiến trúc được sử dụng.

- + Models

Model là bộ phận giữ vai trò nhận dữ liệu từ controller và lưu trữ toàn bộ tất cả những dữ liệu đó vào trong ứng dụng. Nó chính là cầu nối giữa hai thành phần view và controller.

- + Views

View là phần giao diện, nơi mà người dùng có thể thông qua các thao tác truy vấn khác nhau trên website để truy cập, thao tác với trang Web, đây là nơi tạo ra các thành phần của HTML. Bên cạnh đó, chức năng của view là ghi nhận các hoạt động của người dùng để tương tác với controller mà không có quan hệ trực tiếp hay lấy dữ liệu từ controller. Thay vào đó, view chỉ hiển thị những yêu cầu được chuyển cho controller.

- + Controllers

Controller là một bộ phận xử lý những yêu cầu mà người dùng thông qua view để gửi đến. Sau khi xử lý yêu cầu, Controller sẽ đưa cho người dùng những dữ liệu phù hợp. Ngoài ra, controller còn có thêm một chức năng nữa chính là kết nối với model. Controller sẽ lấy dữ liệu và gửi tới model.

+ Services

Services giúp dữ liệu kết nối giữa Model và Database. Nó giúp lấy dữ liệu từ database và các hoạt động thêm, xóa, cập nhật dữ liệu vào Database.

+ Data

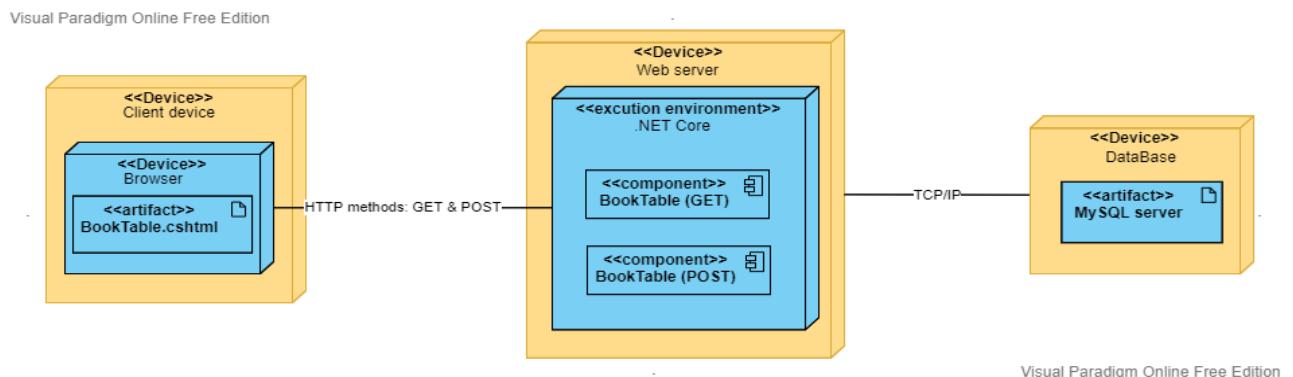
Nơi tạo các Database cho hệ thống để kết nối với SQL server.

3.1.3 Front End

- Sử dụng HTML,CSS,JAVASCRIPT.
- Các thư viện BOOTSTRAP, JQUERY.
- Các tính năng từ google font và font-anwesome.

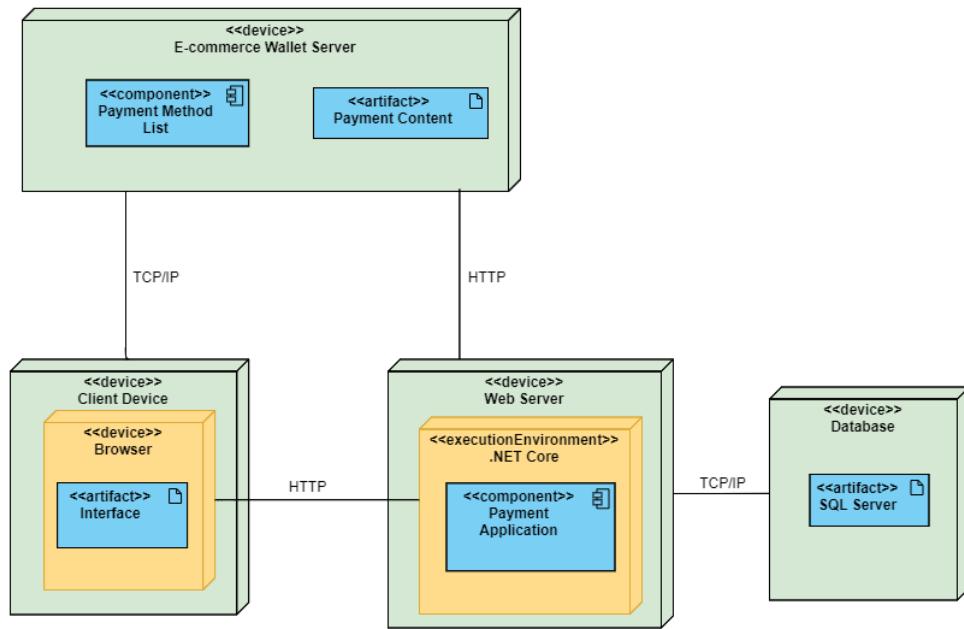
3.2 Draw an implementation diagram for Major (not all) functional requirements

3.2.1 Đặt bàn



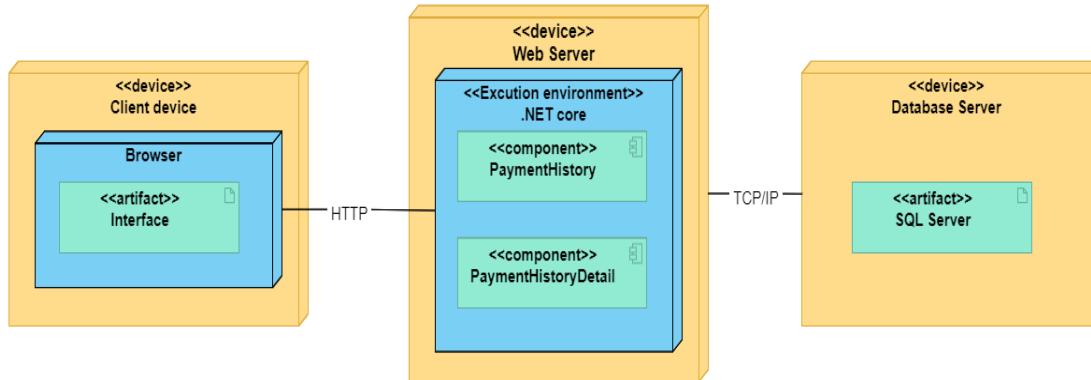
Hình 3.1: Implementation diagram for use case: Reservation

3.2.2 Thanh toán



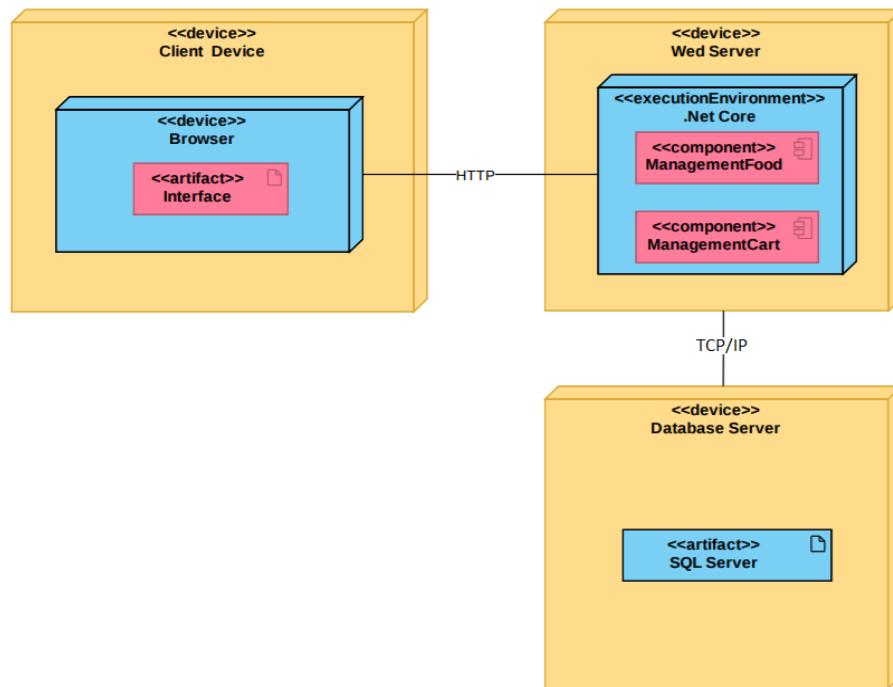
Hình 3.2: Implementation diagram for use case: Payment

3.2.3 Lịch sử thanh toán



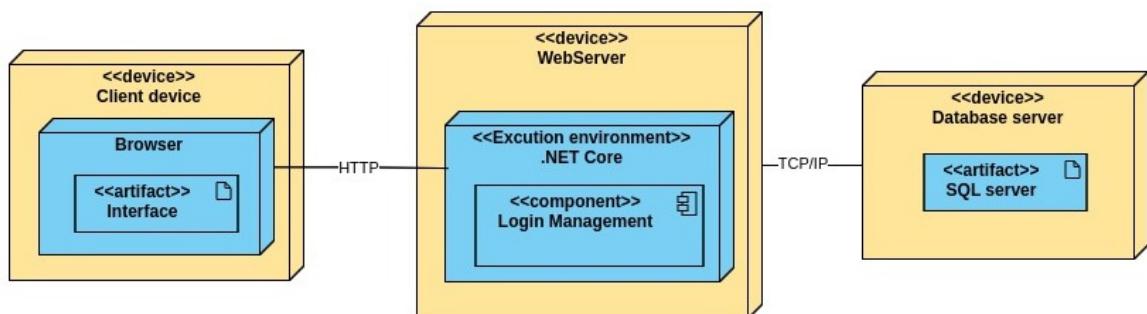
Hình 3.3: Implementation diagram for use case: Lịch sử thanh toán

3.2.4 Thêm món vào giỏ hàng



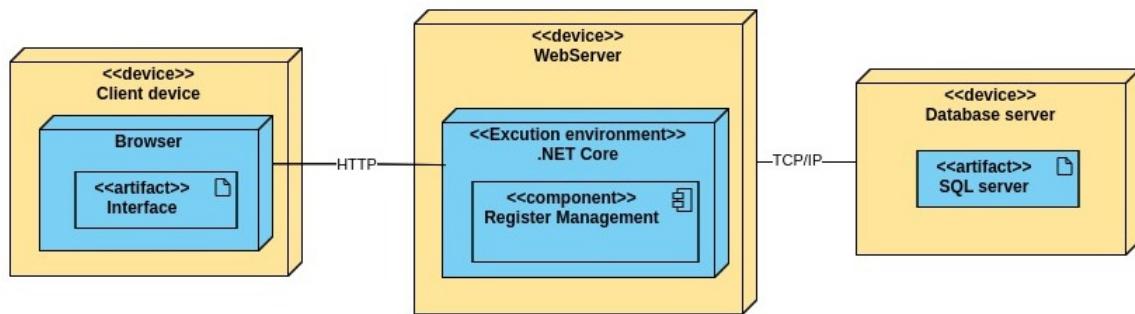
Hình 3.4: Deployment diagram cho use case: Thêm món vào giỏ hàng

3.2.5 Đăng nhập



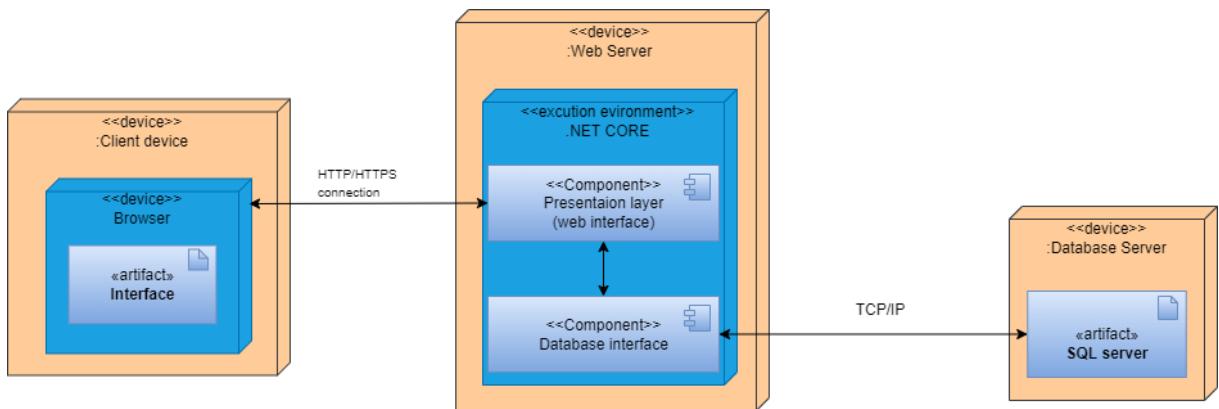
Hình 3.5: deployment diagram cho use case: đăng nhập

3.2.6 Đăng ký



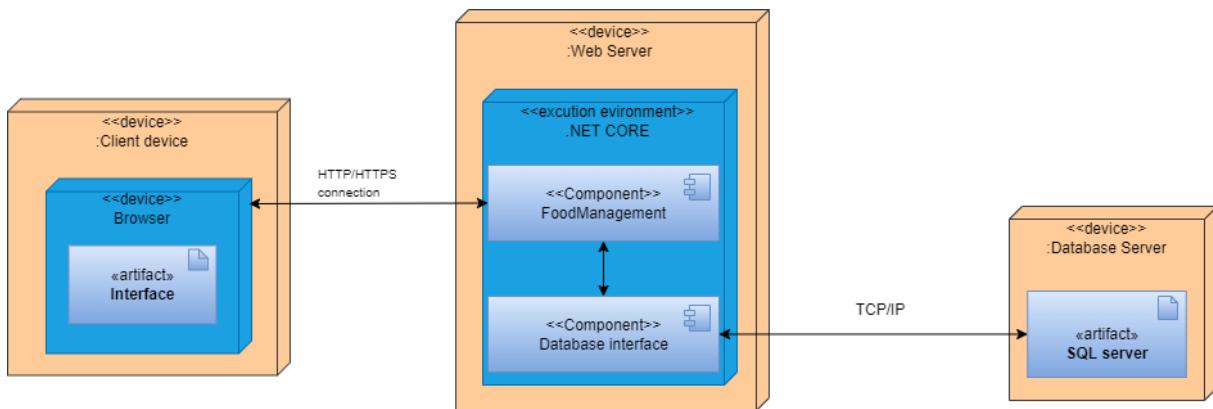
Hình 3.6: deployment diagram cho use case: đăng ký

3.2.7 Lịch sử đặt bàn



Hình 3.7: deployment diagram cho use case: Lịch sử đặt bàn

3.2.8 Lọc danh sách món theo danh mục



Hình 3.8: deployment diagram cho use case: Lọc danh sách món theo danh mục

4 Implementation

4.1 Setting up github

Các công việc đã làm với github:

- Đã tạo repository và thêm các thành viên vào.
- Chia nhánh develop làm nhánh tổng và merge vào nhánh này.
- Các thành viên chia ra code bằng cách tạo nhánh mới và sau khi code xong tạo Pull Request cho các thành viên khác vào review và approve sau đó merge vào Develop.
- Sau khi hoàn tất merge từ nhánh develop vào main.
- Link github <https://github.com/HienNguyen120201/RestaurantPOS>

4.2 Giao diện đã hoàn thành

4.2.1 Header



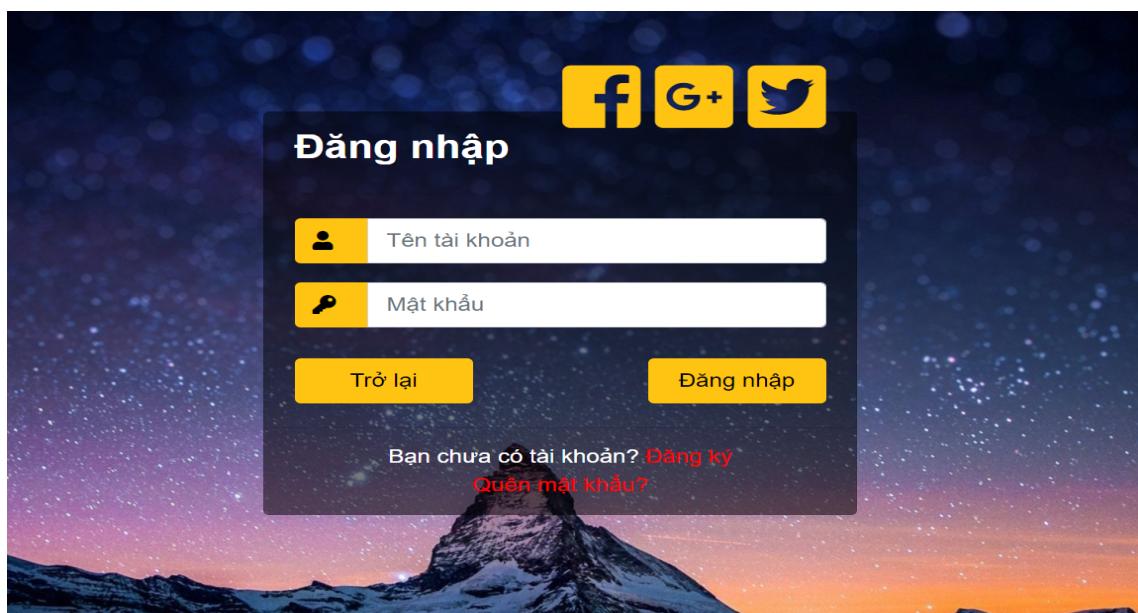
Hình 4.1: Giao diện Header

4.2.2 Footer



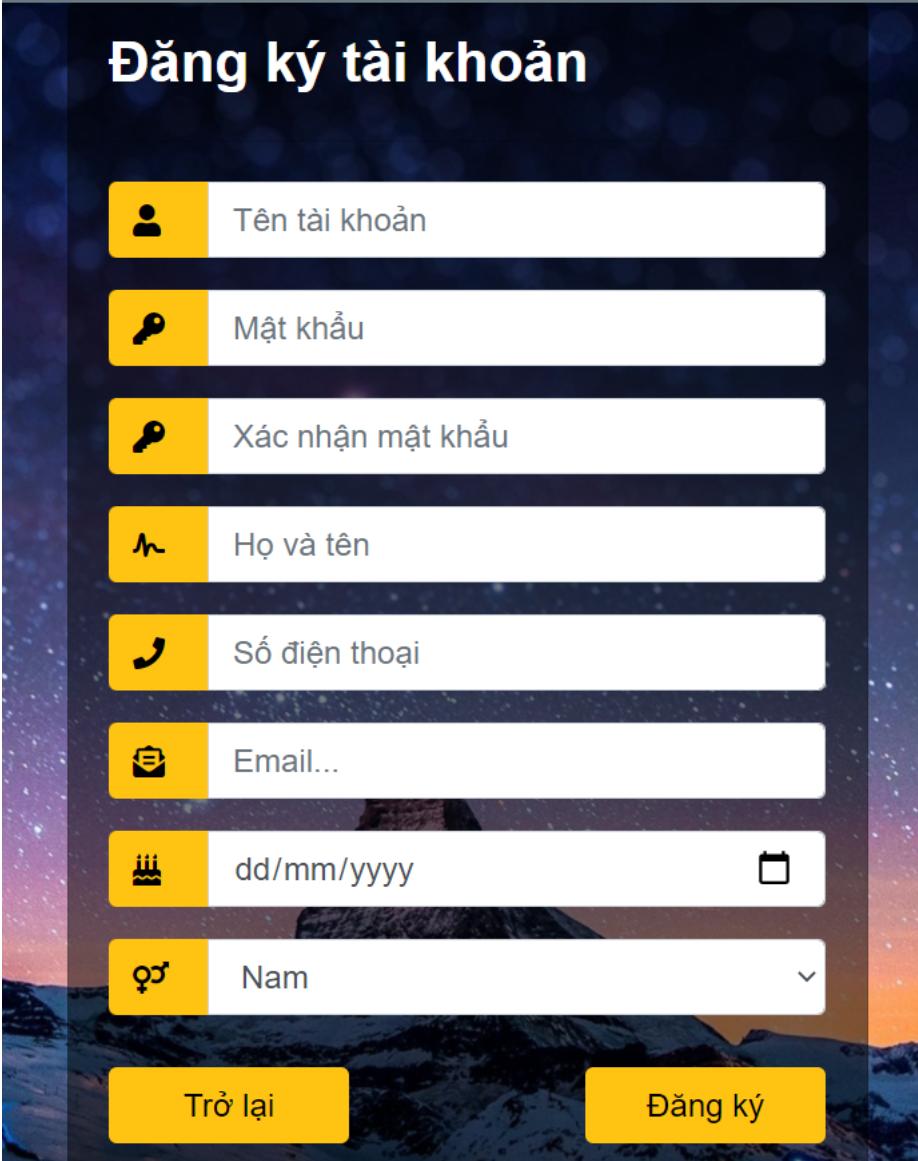
Hình 4.2: Giao diện Footer

4.2.3 Đăng nhập



Hình 4.3: Giao diện Đăng nhập

4.2.4 Đăng ký



Đăng ký tài khoản

Tên tài khoản

Mật khẩu

Xác nhận mật khẩu

Họ và tên

Số điện thoại

Email...

dd/mm/yyyy

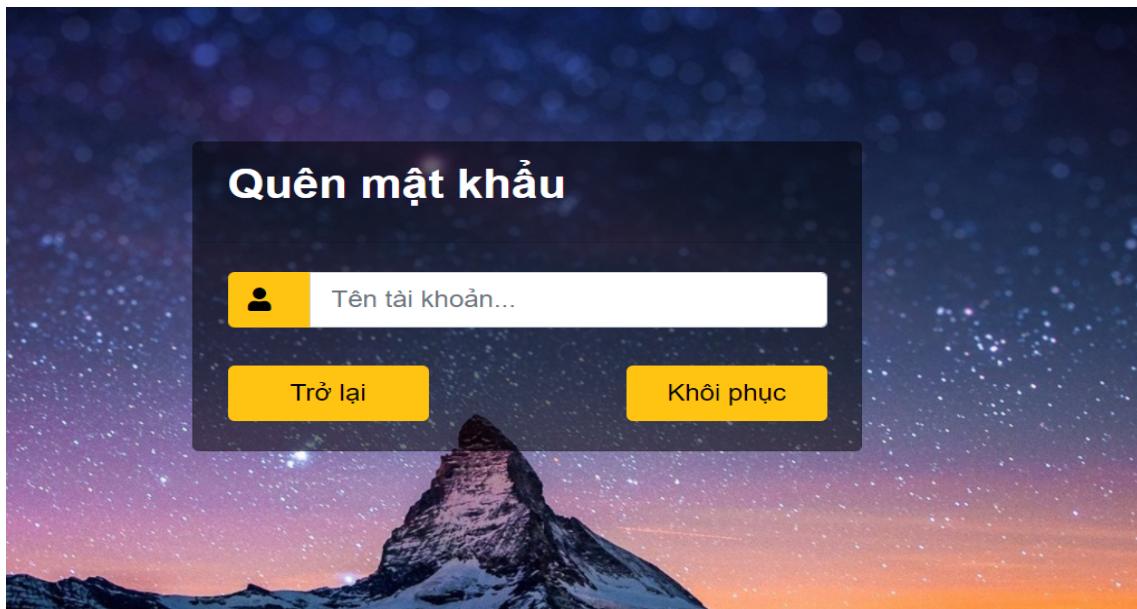
Nam

Trở lại

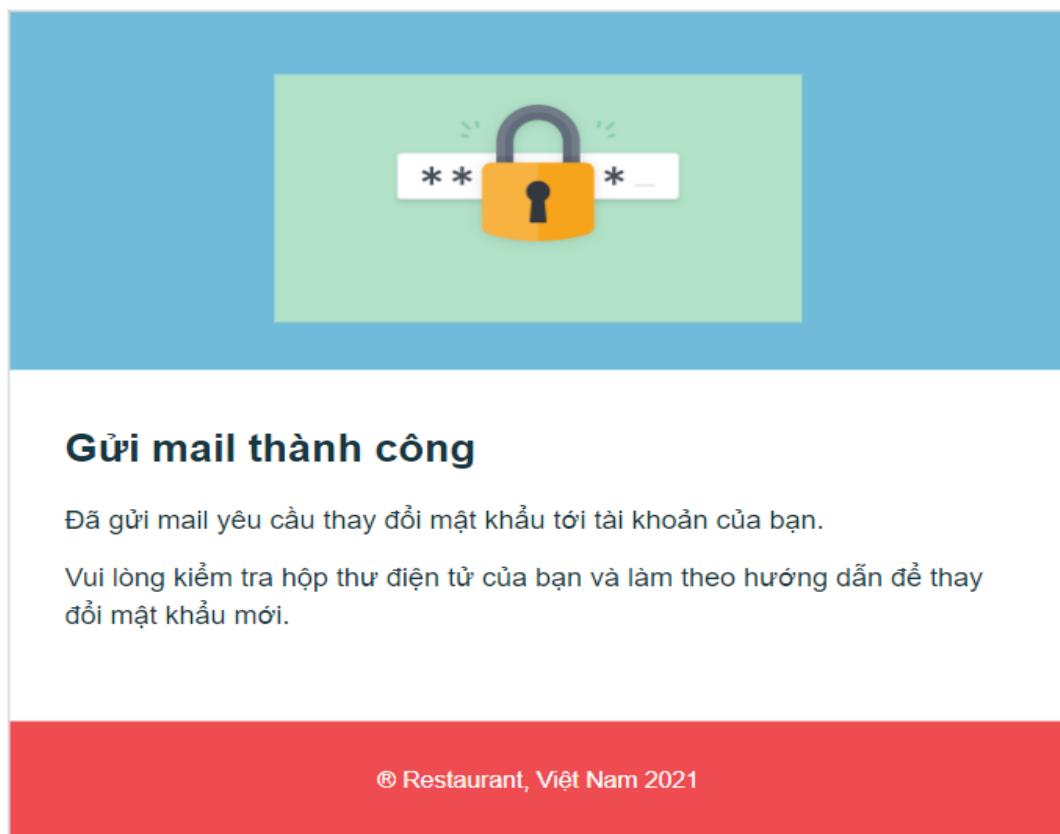
Đăng ký

Hình 4.4: Giao diện Đăng ký

4.2.5 Tạo mật khẩu mới



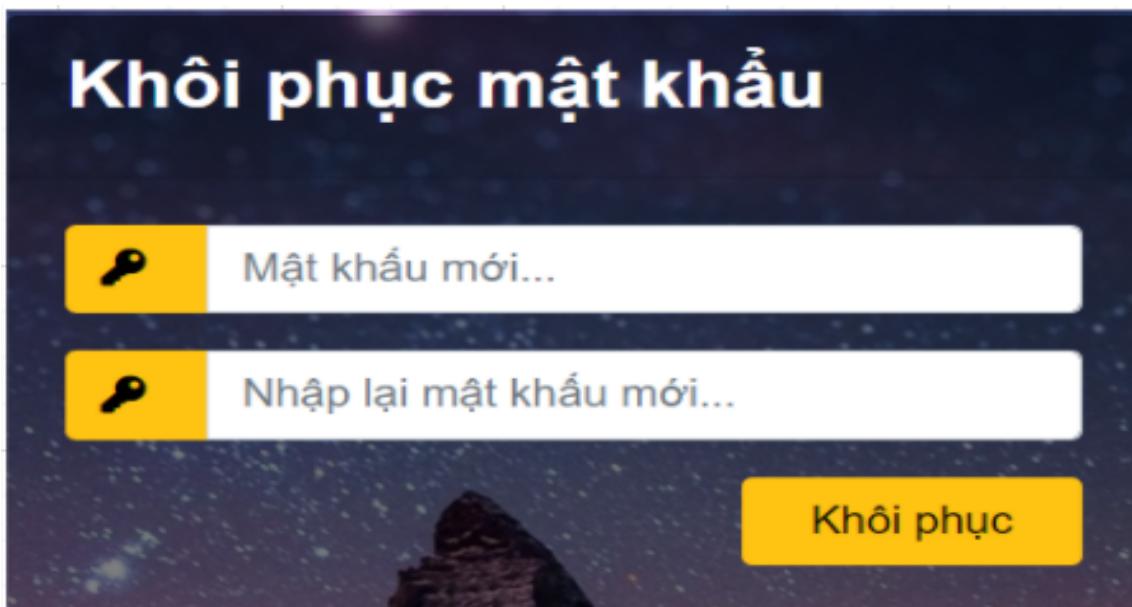
Hình 4.5: Giao diện Lấy lại mật khẩu



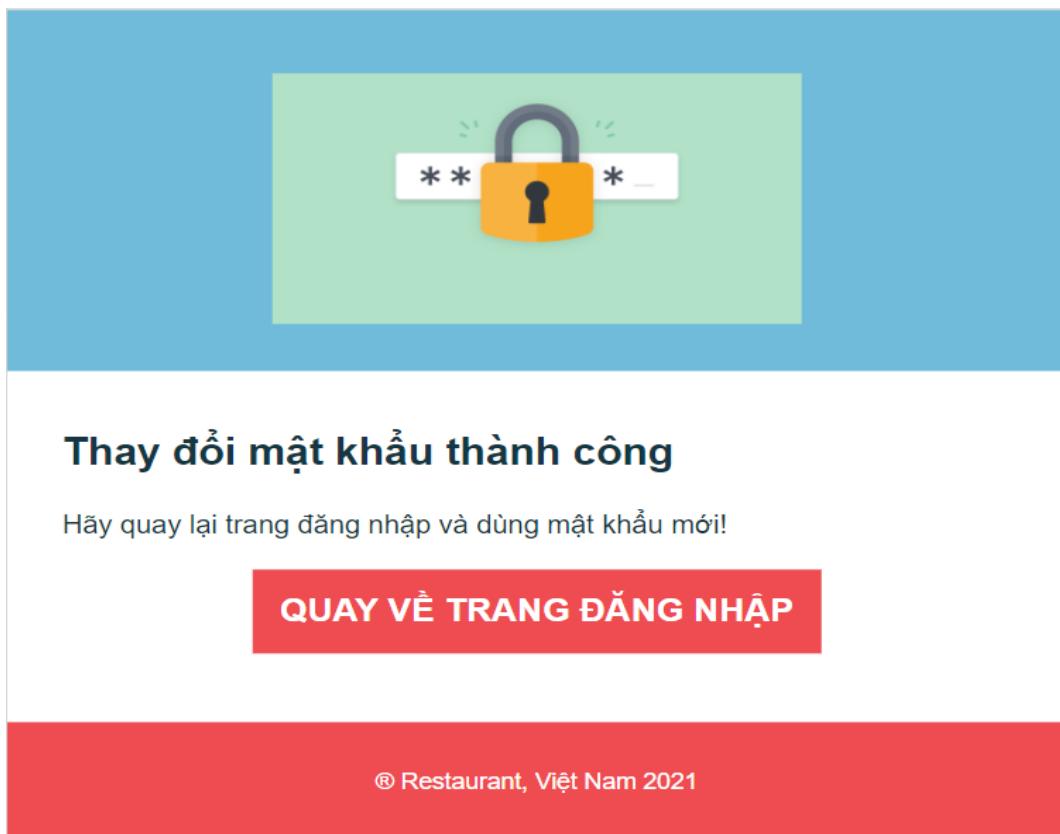
Hình 4.6: Giao diện sau khi bấm vào Khôi phục



Hình 4.7: Form nhận được trong gmail

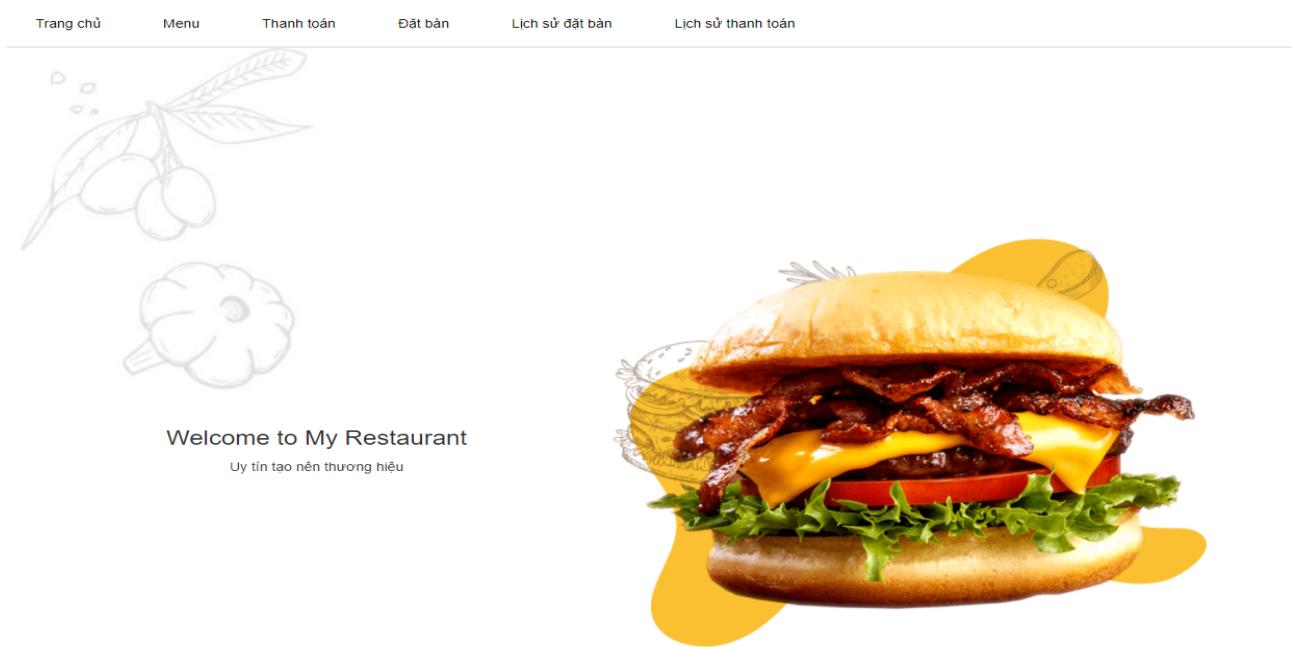


Hình 4.8: Giao diện sau khi bấm vào ĐỔI MẬT KHẨU



Hình 4.9: Giao diện sau khi đổi mật khẩu thành công

4.2.6 Trang chủ



Hình 4.10: Giao diện Trang chủ

4.2.7 Menu

Trang chủ Menu Thanh toán Đặt bàn Lịch sử đặt bàn Lịch sử thanh toán

Các Loại Món Ăn

- Salad
- Món Hấp Xào
- Món Nướng
- Món Chiên Và Bánh
- Món Nướng
- Món Tráng Miệng
- Đồ Uống
- Món Nước
- Bánh Mì, Chả

Lọc Danh Sách

Salad:



SALAD DƯA CHUỘT
30000 VND
★★★★★



SALAD CÁ TRÍCH
50000 VND
★★★★★



CAESAR SALAD
40000 VND
★★★★★

Món Hấp Xào:



HÀ CẢO THẬP CẨM
35000 VND
★★★★★



NGHÊU XAO BƠ TỎI
40000 VND
★★★★★



CUA HOÀNG ĐẾ HẤP SẢ
109900 VND
★★★★★

Hình 4.11: Giao diện Menu

4.2.8 Chi tiết món ăn

Trang chủ Menu Thanh toán Đặt bàn Lịch sử đặt bàn Lịch sử thanh toán



SALAD CÁ TRÍCH

★★★★★
10 Review(s)

Giá: 50000 VND

Salad Nga

SỐ LƯỢNG

ADD TO CART

Hình 4.12: Giao diện Chi tiết món ăn

4.2.9 Đặt bàn

Tìm kiếm bàn trống:

The screenshot shows a search interface for available tables. At the top, there are navigation links: Trang chủ, Menu, Thanh toán, Đặt bàn, Lịch sử đặt bàn, and Lịch sử thanh toán. Below these are two main sections: "Đặt trước" (Booking in advance) and "Bàn tại nhà hàng" (Tables at the restaurant). The "Bàn tại nhà hàng" section contains fields for "Ngày:" (dd/mm/yyyy), "Từ:" (From), "Đến:" (To), and "Số lượng khách:" (Number of guests). A dropdown menu for "Chọn số lượng khách" is open. An orange "TÌM KIẾM" (Search) button is located below these fields. To the right, a "Thời gian mở cửa" (Opening hours) table lists the opening times for each day of the week from Tuesday to Sunday. Below this is a table titled "Chọn bàn muốn đặt" (Select the table you want to book) with columns: STT, Tên bàn, Số lượng khách, Từ, Đến, Trạng thái, and Đặt bàn. A note at the bottom says: "Quý khách chọn thời gian đặt bàn bên trên!" (Please select the booking time above!).

Hình 4.13: Giao diện tìm kiếm bàn trống

Đặt bàn:

The screenshot shows a booking interface. At the top, it features the same search form for available tables as in Figure 4.13. Below this, a table titled "Chọn bàn muốn đặt" (Select the table you want to book) displays three available tables. Each row includes columns for STT, Tên bàn (Table name), Số lượng khách (Number of guests), Từ (From), Đến (To), Trạng thái (Status), and Đặt bàn (Book). The "Đặt bàn" column contains green "ĐẶT BÀN" (Book) buttons. The table rows are as follows:

STT	Tên bàn	Số lượng khách	Từ	Đến	Trạng thái	Đặt bàn
1	Bàn 2	10 người	18/11/2021 09:03	18/11/2021 10:03	Trống	ĐẶT BÀN
2	Bàn 9	10 người	18/11/2021 09:03	18/11/2021 10:03	Trống	ĐẶT BÀN
3	Bàn 13	10 người	18/11/2021 09:03	18/11/2021 10:03	Trống	ĐẶT BÀN

Hình 4.14: Giao diện Đặt bàn

4.2.10 Lịch sử đặt bàn

Trang chủ	Menu	Thanh toán	Đặt bàn	<u>Lịch sử đặt bàn</u>	Lịch sử thanh toán																														
Lịch sử đặt bàn																																			
<table border="1"> <thead> <tr> <th>STT</th><th>Tên bàn</th><th>Số người</th><th>Bắt đầu</th><th>Kết thúc</th></tr> </thead> <tbody> <tr> <td>1</td><td>Bàn 12</td><td>5</td><td>22/11/2021 08:27</td><td>22/11/2021 10:28</td></tr> <tr> <td>2</td><td>Bàn 13</td><td>10</td><td>18/11/2021 10:44</td><td>18/11/2021 11:44</td></tr> <tr> <td>3</td><td>Bàn 2</td><td>10</td><td>18/11/2021 10:09</td><td>18/11/2021 11:09</td></tr> <tr> <td>4</td><td>Bàn 9</td><td>10</td><td>18/11/2021 10:09</td><td>18/11/2021 11:09</td></tr> <tr> <td>5</td><td>Bàn 2</td><td>10</td><td>18/11/2021 03:46</td><td>18/11/2021 04:46</td></tr> </tbody> </table>						STT	Tên bàn	Số người	Bắt đầu	Kết thúc	1	Bàn 12	5	22/11/2021 08:27	22/11/2021 10:28	2	Bàn 13	10	18/11/2021 10:44	18/11/2021 11:44	3	Bàn 2	10	18/11/2021 10:09	18/11/2021 11:09	4	Bàn 9	10	18/11/2021 10:09	18/11/2021 11:09	5	Bàn 2	10	18/11/2021 03:46	18/11/2021 04:46
STT	Tên bàn	Số người	Bắt đầu	Kết thúc																															
1	Bàn 12	5	22/11/2021 08:27	22/11/2021 10:28																															
2	Bàn 13	10	18/11/2021 10:44	18/11/2021 11:44																															
3	Bàn 2	10	18/11/2021 10:09	18/11/2021 11:09																															
4	Bàn 9	10	18/11/2021 10:09	18/11/2021 11:09																															
5	Bàn 2	10	18/11/2021 03:46	18/11/2021 04:46																															

Hình 4.15: Giao diện Lịch sử đặt bàn

4.2.11 Thanh toán

Mã đơn hàng 29b33c77-bffe-45b8-9c27-7712e7c7e57b	Ngày: 06/11/2021 Thời gian: 14:38
STT	Tên món
1	Salad cá trích
2	Nghêu xào bơ tỏi
	Tạm tính
	90000
	VAT (10%)
	9000
Trả lại	
	Thanh toán 99000 VNĐ

Hình 4.16: Giao diện Thanh toán

4.2.12 Các phương thức thanh toán

Ví điện tử Thẻ ngân hàng Chuyển khoản

Chọn ví điện tử:

Momo Zalo Pay P

Tài khoản

Nhập tên tài khoản

Chọn ngân hàng

--Chọn ngân hàng--

Số thẻ

Nhập mã số thẻ

Hiệu lực từ ngày

--Tháng-- --Năm--

Mật khẩu

Xác nhận

Hoàn thành

Tài khoản: 123456789
Ngân hàng: ViettinBank
Số tiền: 22000 VND
Nội dung: Thanh toán hóa đơn.

Xác nhận

Hình 4.17: Giao diện 3 phương thức Thanh toán

4.2.13 Lịch sử thanh toán

Trang chủ	Menu	Thanh toán	Đặt bàn	Lịch sử đặt bàn	Lịch sử thanh toán
-----------	------	------------	---------	-----------------	--------------------

Lịch sử thanh toán

STT	Chi tiết	Tổng (VNĐ)	Phương thức thanh toán	Ngày thanh toán
1	Xem chi tiết:	1426700	Ví điện tử ZaloPay	06/11/2021 22:42
2	Xem chi tiết:	1295800	Ví điện tử AirPay	20/10/2021 17:30
3	Xem chi tiết:	159500	Ví điện tử AirPay	03/11/2021 22:17
4	Xem chi tiết:	55000	Ví điện tử ZaloPay	06/11/2021 22:50
5	Xem chi tiết:	49500	Ví điện tử ZaloPay	06/11/2021 22:50
6	Xem chi tiết:	800800	Ví điện tử AirPay	06/11/2021 22:50
7	Xem chi tiết:	800800	Ví điện tử AirPay	06/11/2021 22:50

Hình 4.18: Giao diện Lịch sử thanh toán

4.2.14 Giỏ hàng

Giỏ hàng

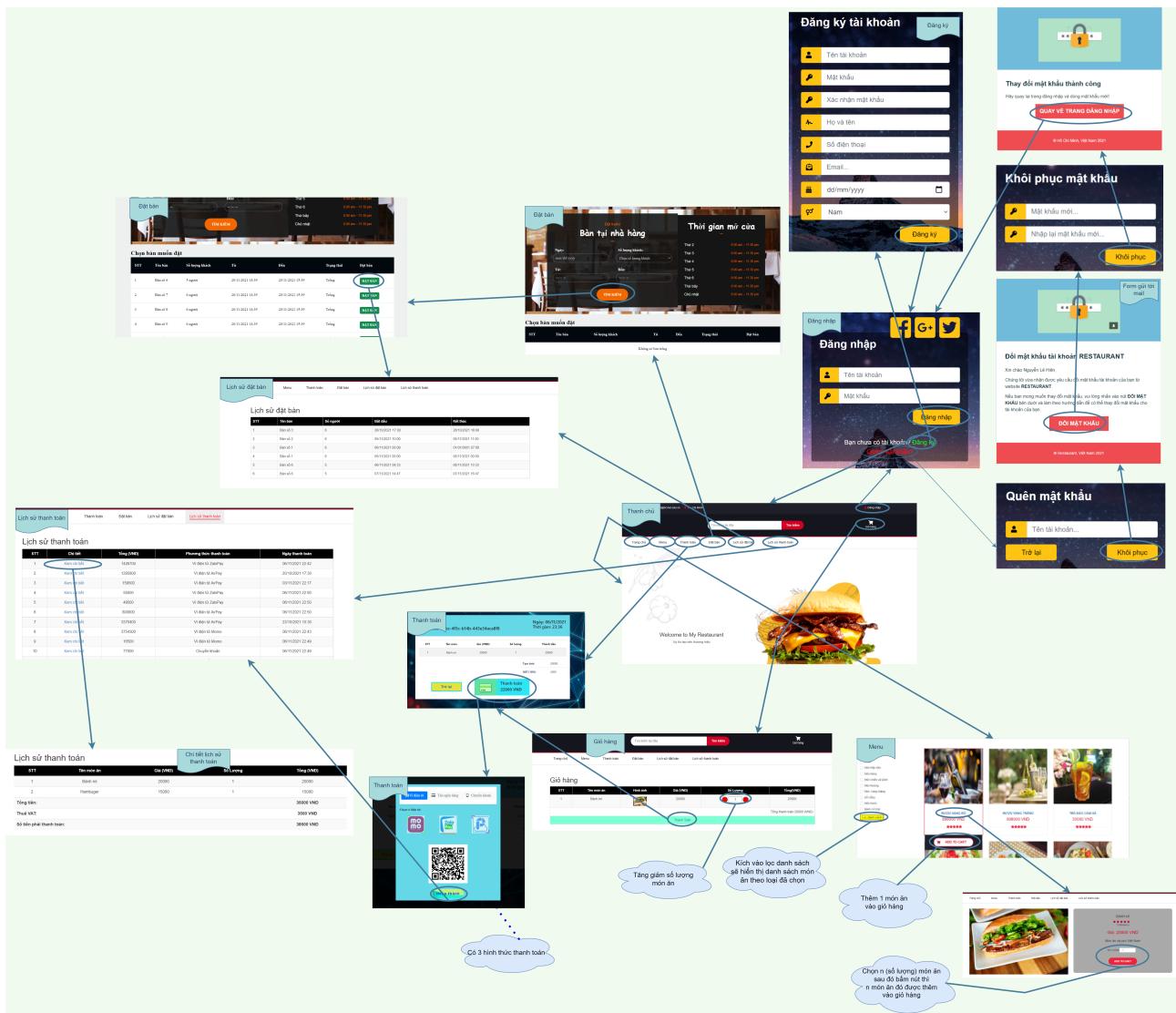
STT	Tên món ăn	Hình ảnh	Giá (VNĐ)	Số Lượng	Tổng(VND)
1	Bánh mì		20000	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	20000

Tổng thanh toán: 20000 (VND)

Thanh Toán

Hình 4.19: Giao diện Giỏ hàng

4.2.15 Flow of screen



Hình 4.20: ScreenFlow

Link Flow of screen: <https://drive.google.com/file/d/1MiKPihuM-KzW9NwNDncWSj3view?usp=sharing>

4.3 Mô tả các chức năng đã hoàn thành

4.3.1 Layout và Trang chủ

- **Header** như hình 4.1. Tại đây có thể bấm vào **Đăng nhập** để đăng nhập (nếu chưa đăng nhập) hoặc bấm **Đăng xuất** (nếu đã đăng nhập). Bấm **Giỏ hàng** để xem giỏ hàng. Ngoài ra còn có thanh chọn đường dẫn để đến với các chức năng khác: Trang chủ, Menu, Thanh toán, Lịch sử thanh toán. Hai chức năng Đặt bàn và Lịch sử đặt bàn chỉ được sử dụng khi khách hàng là thành viên VIP.
- **Footer** như hình 4.2.
- **Trang chủ** như hình 4.10. Trang chủ có slogan của nhà hàng và một ảnh động thêm đẹp mắt.

4.3.2 Đăng nhập, đăng ký, lấy lại mật khẩu

- **Đăng nhập** như hình 4.3: Nhập Tên tài khoản và Mật khẩu, sau khi điền đầy đủ bấm **Đăng nhập**. Nếu tên tài khoản và mật khẩu hợp lệ sẽ chuyển đến trang chủ. Nếu chưa có tài khoản bấm **Đăng ký** để đăng ký tài khoản. Nếu quên mật khẩu bấm **Quên mật khẩu** để lấy lại mật khẩu.
- **Đăng ký** như hình 4.4: Nhập các thông tin gồm: Tên tài khoản, Mật khẩu(gồm chữ cái in hoa, chữ thường, kí tự đặc biệt và số), Xác nhận mật khẩu, Họ và tên, Số điện thoại, Email, Ngày/Tháng/Năm sinh, Giới tính. Sau đó bấm **Đăng ký** để đăng ký tài khoản. Nếu Tên đăng nhập đã tồn tại hoặc mật khẩu không đúng mẫu thì sẽ không đăng ký được.
- **Quên mật khẩu** như hình 4.5: Tại đây người dùng nhập Tên tài khoản vào và bấm **Khôi phục** sau đó hệ thống sẽ hiển thị như hình 4.6 và hệ thống sẽ gửi một mail có from như hình 4.7 cho người dùng. Người dùng bấm **ĐỔI MẬT KHẨU** sau đó hệ thống hiển thị như hình 4.8. Tại đây người dùng nhập mật khẩu mới và xác nhận mật khẩu mới sau đó bấm **Khôi phục** và nếu tạo lại mật khẩu mới thành công hệ thống hiển thị như hình 4.9.

4.3.3 Hiển thị menu món ăn, Xem chi tiết món ăn, Thêm món ăn vào giỏ hàng

- **Menu** như trong Hình 4.11: Tại đây có thể lọc danh sách các món mà khách hàng muốn chọn để đơn giản menu bằng cách tick vào các loại món ăn và bấm

Lọc danh sách. Tại đây bấm vào **ADD TO CART** sẽ thêm món ăn vào giỏ hàng với số lượng là 1 món. Bấm vào tên món ăn để xem chi tiết hơn về món ăn.

- **Chi tiết món ăn** như hình 4.12: Tại đây khách hàng xem chi tiết về món ăn và chọn số lượng muốn thêm vào giỏ hàng. Bấm **ADD TO CART** để thêm vào giỏ hàng.
- Chức năng thêm món ăn vào giỏ hàng sẽ chia làm 2 trường hợp:
 - Nếu khách hàng đã có giỏ hàng và chưa thanh toán thì sẽ tiếp tục thêm vào giỏ hàng đó
 - Nếu khách hàng chưa có giỏ hàng thì sẽ tiến hành tạo giỏ hàng mới sau đó tạo món ăn trong giỏ hàng
- Khi đã có giỏ hàng và muốn thêm món ăn vào cũng chia làm 2 trường hợp:
 - Nếu món ăn đã có trong giỏ hàng thì cộng số lượng món ăn đã có với số lượng món ăn vừa chọn.
 - Nếu chưa có sẽ thêm món ăn đó vào giỏ hàng với số lượng đã chọn.

4.3.4 Đặt bàn và xem lịch sử đặt bàn

Đặt bàn như hình 4.13: Tại đây khách hàng chọn : Ngày/Tháng/Năm, Số lượng khách, Thời gian đặt từ mấy giờ tới mấy giờ, sau đó bấm **Tìm kiếm** để tìm bàn trống. Sau khi hiện ra bàn trống khách hàng bấm **Đặt bàn** và sẽ hiển thị bàn đã đặt trong **Lịch sử đặt bàn** như hình 4.15.

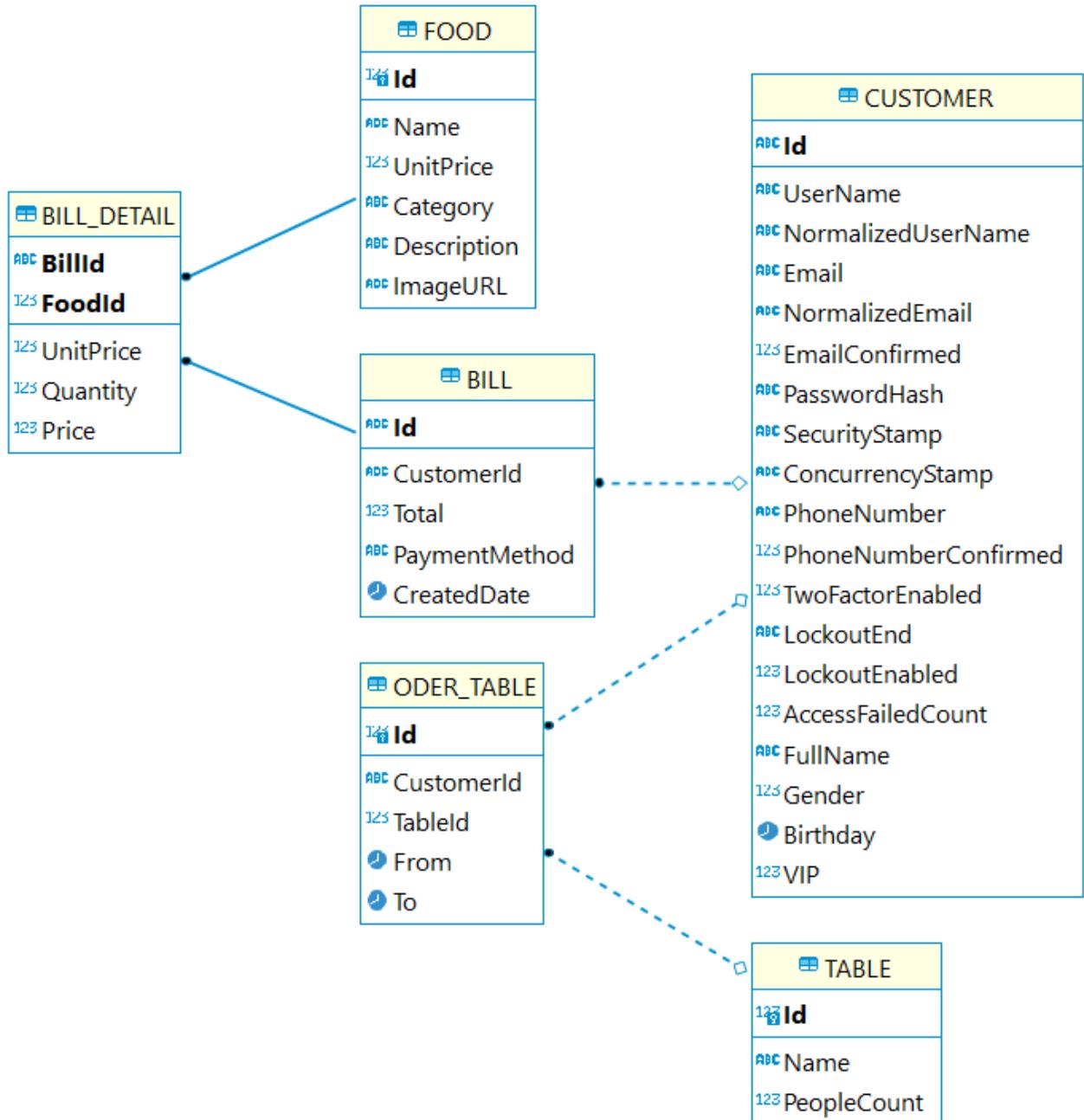
4.3.5 Xem giỏ hàng và chỉnh sửa giỏ hàng

Xem giỏ hàng như hình 4.19: Tại đây có thể bấm vào 2 nút (+) hoặc (-) để thêm hoặc bớt số lượng món ăn. Sau đó bấm **Thanh toán** để tiến hành thanh toán.

4.3.6 Thanh toán và xem lịch sử thanh toán

Thanh toán như hình 4.16: Tại đây khách hàng bấm **Thanh toán** sẽ hiển thị ra 3 phương thức thanh toán. Sau khi xác nhận phương thức thanh toán và bấm **Xác nhận** hoặc **Hoàn thành** để thanh toán. Sau đó đơn thanh toán sẽ hiển thị trong **Lịch sử thanh toán** như hình 4.18..

5 Database diagram và mô tả cách kết nối cơ sở dữ liệu



Hình 5.1: Database Diagram

- Nhóm chúng em đã đăng ký một host trên trang somee.com để có thể triển khai database cũng như website của nhóm hoạt động. Sau đây là các bước mà nhóm đã thực hiện để kết nối với cơ sở dữ liệu.
 - + Bước 1: Bởi nhóm sử dụng code first để hiện thực database nên ở bước đầu tiên, nhóm đã code các Entity, EntityConfiguration, DbContext rồi sau đó mới thực hiện kết nối database.
 - + Bước 2: Sau khi đăng ký host thành công, somee cung cấp sẵn connection-string để cấu hình trong Entity Framework Core. Lấy chuỗi này và gắn cho DefaultConnection trong file appsettings.json trong project.
 - + Bước 3: Bởi bản thân lúc khởi tạo project ASP.NET Core MVC tại Visual Studio thì ở file Startup.cs đã inject DbContext bằng cách sử dụng dependency injection rồi nên nhóm chỉ add thêm identity để hỗ trợ việc login, logout sau này.
 - + Bước 4: Nhóm tiến hành add-migration và update database để tạo database.
 - + Bước 5: Màn hình terminal hiển thị update database thành công. Từ giờ các thao tác trên database sẽ được thực hiện thông qua DbContext.

6 Công nghệ Trending

- Nhóm chúng em ứng dụng 1 mảng nhỏ của DevOps(Ci/Cd) để làm việc với nhau qua github và tự động deploy project.
 - + **Ci:** Khi thành viên nào tạo pullrequest, Ci giúp kiểm tra build thành công mới cho phép merge.
 - + **Cd:** Khi có hành động merge vào nhánh main, Cd giúp tự động deploy project lên heroku(address: <https://awesomerestaurants.herokuapp.com/>).
- Triển khai Ci/Cd thông qua github action và docker file. Tạo thư mục .github/- workflows. Trong thư mục workflows, tạo hai file **ci.yml** , **desploy.yml** .
 - + **Ci(ci.yml)**

```
21 lines (18 sloc) | 489 Bytes
Raw Blame ⌂ ⌄ ⌅ ⌆
1 name: Check build pass before merging
2
3 on:
4   pull_request:
5     branches:
6       - "*"
7
8 jobs:
9   build:
10    name: Check build pass before merging
11    runs-on: ubuntu-latest
12
13 steps:
14   - name: Clone code
15     uses: actions/checkout@v2
16   - name: Setup .NET Core 3.1
17     uses: actions/setup-dotnet@v1
18     with:
19       dotnet-version: 3.1.x
20   - name: Build Project
21     run: dotnet build
```

Hình 6.1: Nội dung file Ci.yml tương ứng với action check build pass before merge trong github Actions

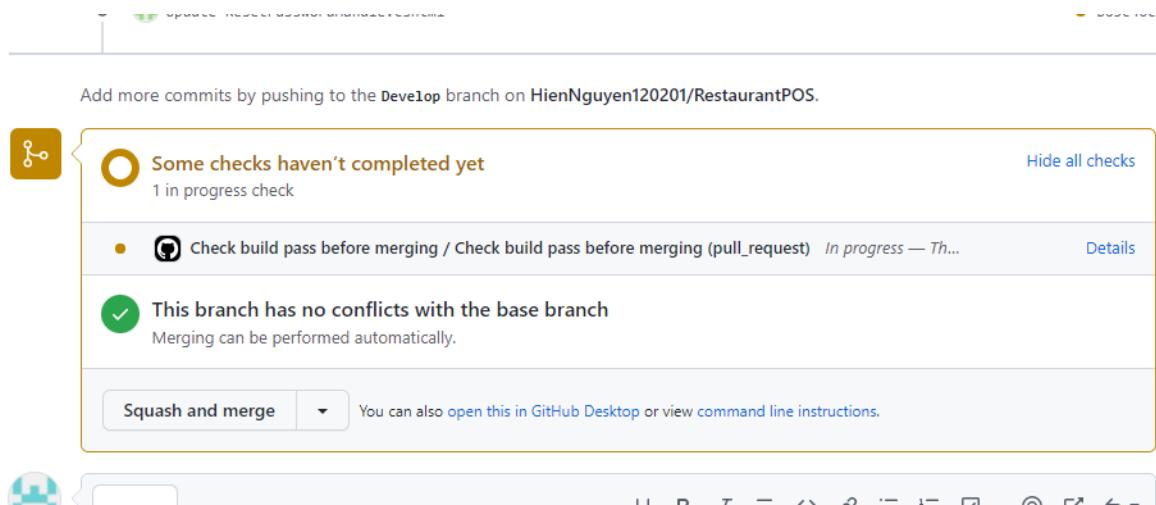
+ Cd(deploy.yml)

```
26 lines (22 sloc) | 725 Bytes
1 name: Deploy to Heroku
2
3 on:
4   push:
5     branches:
6       - main
7
8 env:
9   HEROKU_API_KEY: ${{ secrets.HEROKU_API_KEY }}
10  APP_NAME: ${{ secrets.HEROKU_APP_NAME }}
11
12 jobs:
13   build:
14     name: Deploy to Heroku
15     runs-on: ubuntu-latest
16     steps:
17       - name: Clone repository
18       uses: actions/checkout@v2
19
20       - name: Login to Heroku from Docker
21       run: |
22         docker login --username=_ --password=$HEROKU_API_KEY registry.heroku.com
23       - name: Build & Push the Docker image to Heroku Container Registry
24       run: |
25         heroku container:push web -a $APP_NAME
26         heroku container:release web -a $APP_NAME
```

Hình 6.2: Nội dung file deploy.yml tương ứng với action deploy to heroku trong github Actions

➤ Kiểm tra hoạt động CiCd.

+ Ci(Action Check build pass before merge) . Khi tạo pull request, sẽ tự động build code bằng docker build. Nếu success thì mới cho phép merge, không thì không cho phép.



Hình 6.3: Kiểm tra hoạt động action Check build pass before merge(Ci)

+ Cd(Action deploy to Heroku) . Tự động deploy lên heroku mỗi lần merge vào nhánh main.

The screenshot shows the GitHub Workflows interface under the 'All workflows' tab. It displays 20 workflow runs. The 'Deploy to Heroku' action is highlighted in blue. Five successful runs are listed:

- Cicd**: Check build pass before merging #9: Pull request #46 opened by ThuongNguyenVan158. Status: cicd. Run time: 2 hours ago, 38s.
- Cicd**: Check build pass before merging #8: Pull request #43 opened by ThuongNguyenVan158. Status: cicd. Run time: 2 hours ago, 35s.
- Origin/feature/order table**: Check build pass before merging #7: Pull request #42 opened by ThuongNguyenVan158. Status: origin/Feature/Order-Table. Run time: 3 hours ago, 37s.
- Merge pull request #41 from HienNguyen120201/Develop**: Deploy to Heroku #11: Commit f323b2a pushed by luucongdinh113. Status: main. Run time: 3 hours ago, 1m 22s.
- Develop**: Check build pass before merging #6: Pull request #41 opened by luucongdinh113. Status: Develop. Run time: 3 hours ago, 34s.

Hình 6.4: Kiểm tra hoạt động action Deploy to heroku(Cd)

Tài liệu

- [1] Requirements Engineering. *Chapter two.* Truy cập tại <https://gyires.inf.unideb.hu/GyBITT/07/ch02s04.html>.
- [2] Developing Use Case Diagrams - Use Case Scenarios. Truy cập tại <https://www.w3computing.com/systemsanalysis/use-case-diagrams-scenarios/>.