

**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

Restaurant POS 2.0

LỚP L02 - HK212

GVHD: PGS.TS Bùi Hoài Thắng
PGS.TS Quản Thành Thơ
GV Bùi Công Tuấn
Sinh viên thực hiện: Trịnh Việt Hoàng - 1752216
Bùi Thanh Duy - 1912871
Nguyễn Trọng Anh Tuấn - 2010756
Nguyễn Lê Gia Hinh - 2011213
Trần Quốc Thái - 2010616

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 2 NĂM 2022



Mục lục

1 Requirement elicitation	5
1.1 Identifying the context of this project, relevant stakeholders and the scope of the project	5
1.2 Description of all functional and non-functional requirements of the desired system. Use-case diagram of the whole system	7
1.2.1 Functional requirements	7
1.2.2 Non-functional requirements	9
1.2.3 Use-case diagram of the whole system	11
1.3 Use-case diagrams and description of some features	12
1.3.1 Use-case : Đặt bàn	12
1.3.2 Use-case : Thanh toán	14
1.3.3 Use-case : Đăng Ký	16
1.3.4 Use-case : Đăng Nhập	18
1.3.5 Use-case : Xem lịch sử thanh toán	20
1.3.6 Use-case : Tạo đơn hàng	22
1.3.7 Use-case : Duyệt lịch sử đặt bàn	24
1.3.8 Use-case : Duyệt món ăn theo danh mục món ăn	25
2 System Modeling	27
2.1 Activity diagram to capture Major (not all) functional requirements of the desired system.	27
2.1.1 Xem Menu, chi tiết món ăn và thêm vào giỏ hàng	27
2.1.2 Đặt bàn	28
2.1.3 Xem lịch sử thanh toán	29
2.1.4 Thanh toán	30
2.1.5 Đăng ký	31
2.1.6 Đăng nhập	32
2.1.7 Kiểm tra lịch sử đặt bàn	33
2.1.8 Kiểm tra lịch sử đặt bàn	34
2.2 Sequence diagram for use-case in Task 1.3.	35
2.2.1 Đặt bàn	35
2.2.2 Thanh toán	36
2.2.3 Lịch sử thanh toán	37
2.2.4 Xem menu và xem chi tiết món ăn	37
2.2.5 Thêm món ăn vào giỏ hàng	38
2.2.6 Đăng ký	39
2.2.7 Đăng nhập	41
2.2.8 Lịch sử đặt bàn	42
2.2.9 Lọc danh sách món ăn theo danh mục	43



2.3 Class Diagram	44
3 Architecture design	45
3.1 Describe an architectural approach you will use to implement the desired system	45
3.1.1 Database Management: SQL Server	45
3.1.2 Back End: .NET Core Framework and MVC Design Pattern	45
3.1.3 Front End	45
3.2 Draw an implementation diagram for Major (not all) functional requirements	46
3.2.1 Đặt bàn	46
3.2.2 Thanh toán	46
3.2.3 Lịch sử thanh toán	47
3.2.4 Thêm món ăn vào giỏ hàng	48
3.2.5 Đăng nhập	48
3.2.6 Đăng ký	49
3.2.7 Lịch sử đặt bàn	49
3.2.8 Lọc danh sách món theo danh mục	50
4 Implementation - Sprint 1	50
4.1 Setting up github	50
4.2 Giao diện đã hoàn thành	51
4.2.1 Header	51
4.2.2 Footer	51
4.2.3 Đăng nhập	52
4.2.4 Đăng ký	53
4.2.5 Tạo mật khẩu mới	54
4.2.6 Trang chủ	59
4.2.7 Menu	60
4.2.8 Chi tiết món ăn	61
4.2.9 Đặt bàn	61
4.2.10 Lịch sử đặt bàn	62
4.2.11 Thanh toán	63
4.2.12 Các phương thức thanh toán	63
4.2.13 Lịch sử thanh toán	63
4.2.14 Giỏ hàng	64
4.2.15 Flow of screen	64
4.3 Mô tả các tính năng	65
4.3.1 Layout và Trang chủ	65
4.3.2 Đăng nhập, đăng ký, quên mật khẩu	66



4.3.3	Hiển thị Menu, xem thông tin món ăn, thêm món ăn vào giỏ hàng	66
4.3.4	Đặt bàn, xem lịch sử đặt bàn	67
4.3.5	Xem giờ hàng, chỉnh sửa đơn hàng	67
4.3.6	Thanh toán, lịch sử thanh toán	67
5	Implementation - Sprint 2	68
5.1	Database diagram	68
5.2	Triển khai công nghệ CI/CD	69



Danh sách thành viên và phân chia công việc

STT	Họ và tên	MSSV	Công việc
1	Trịnh Việt Hoàng	1752216	Dăng ký và đăng nhập
2	Bùi Thanh Duy	1912871	Đặt bàn và thanh toán
3	Nguyễn Trọng Anh Tuấn	2010756	Xem lịch sử thanh toán và tạo đơn hàng.
4	Nguyễn Lê Gia Hinh	2011213	Duyệt lịch sử đặt bàn và duyệt món ăn theo danh mục
5	Trần Quốc Thái	2010616	Xác định ngữ cảnh, đặc tả các yêu cầu cho hệ thống



1 Requirement elicitation

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

- **Ngữ cảnh:** Trong đại dịch Covid-19, lệnh giãn cách được áp đặt trên nhiều quốc gia khiến cho các nhà hàng, quán ăn, các nhà đầu tư trong lĩnh vực liên quan phải đối diện với việc dừng hoạt động. Hệ thống Restaurant POS 2.0 được đề xuất với kỳ vọng sẽ giúp các nhà đầu tư, các chủ nhà hàng, quán ăn có thể tiếp tục kinh doanh, mở rộng quy mô và giảm thiểu hao phí trong việc duy trì hoạt động kinh doanh giữa đại dịch. Hệ thống này cung cấp giải pháp cho việc đặt và thanh toán món ăn cho nhà hàng gián tiếp thông qua hệ thống, giúp khách hàng thực hiện tốt lệnh giãn cách cũng như tiện lợi hơn trong việc đặt món, góp ý, phản ánh thực nhất về chất lượng của nhà hàng.
- **Các đối tượng liên quan đến hệ thống:** Khách hàng, chủ nhà hàng, nhân viên, đầu bếp, ngân hàng, người quản lý hệ thống, ...
- **Phạm vi dự án:**
 - Mục đích: Phần mềm Restaurant POS 2.0 có khả năng thực hiện các giao dịch mua, đặt hàng đồ ăn thông qua mạng internet một cách thuận tiện nhất và tránh tiếp xúc trực tiếp nhiều nhất cho Khách hàng
 - Các module sản phẩm cần có:
 - * Thanh toán online thông qua hệ thống
 - * Đặt món, đặt bàn thông qua hệ thống
 - * Nhận phản hồi của khách hàng thông qua hệ thống
 - * Nhân viên tương tác với khách hàng, xác nhận đơn hàng thông qua hệ thống
 - Các tính năng cần thực hiện
 - * Đặt bàn: Các khách hàng sử dụng hệ thống được phép đặt bàn
 - * Lịch sử đặt bàn: Các khách hàng sử dụng hệ thống được xem lịch sử đặt bàn
 - * Thanh toán: Khách hàng sử dụng hệ thống phải thanh toán bằng thẻ ngân hàng, liên kết ngân hàng số hoặc tiền mặt
 - * Lịch sử thanh toán: Các khách hàng sử dụng hệ thống được xem lịch sử thanh toán
 - * Thêm món vào giỏ hàng: Các khách hàng sử dụng hệ thống được thêm món ăn trên menu vào giỏ hàng



- * Đăng ký: Các khách hàng chưa sử dụng hệ thống
- * Đăng nhập: Các khách hàng đã sử dụng hệ thống
- * Đăng xuất: Các khách hàng được sử dụng hệ thống
- * Thay đổi mật khẩu: Các khách hàng đã đăng ký sử dụng hệ thống
- * Hiển thị menu món ăn: Các khách hàng sử dụng hệ thống có thể thấy được các món ăn được lưu trong hệ thống
- * Lọc danh sách món ăn: Các khách hàng sử dụng hệ thống có thể lựa chọn các món ăn được lưu trong hệ thống theo một số tiêu chí nhất định của khách hàng đưa ra
- * Hiển thị chi tiết món ăn: Các khách hàng sử dụng hệ thống có thể thấy chi tiết thông tin về món ăn khách hàng đã chọn
- * Xem giờ hàng: Các khách hàng sử dụng hệ thống có thể xem giờ hàng của mình
- * Gửi đánh giá, phản hồi: Các khách hàng vừa nhận đơn món ăn hoặc đã dùng bữa tại nhà hàng sử dụng hệ thống để đặt bàn có thể đánh giá và gửi phản hồi



1.2 Description of all functional and non-functional requirements of the desired system. Use-case diagram of the whole system

1.2.1 Functional requirements

Đối với khách hàng:

- Interactive requirements:

- Chức năng đặt bàn: Gửi thông tin đặt bàn cho nhân viên và xác nhận thông tin đặt bàn cho khách
- Chức năng lịch sử đặt bàn: Xuất danh sách các lần đặt bàn của khách thông qua hệ thống
- Chức năng thanh toán: Xuất thông tin hóa đơn về giỏ hàng của khách hàng và thực hiện giao dịch bằng thẻ ngân hàng, liên kết ngân hàng số hoặc tiền mặt
- Chức năng lịch sử thanh toán: Xuất danh sách về thông tin chi tiết về những lần thanh toán của khách hàng thông qua hệ thống
- Chức năng thêm món vào giỏ hàng: Thêm thông tin về món ăn khách hàng lựa vào giỏ hàng của khách
- Chức năng xem giỏ hàng: Xuất danh sách các món ăn mà khách hàng đã chọn
- Chức năng đăng ký tài khoản: Tạo tài khoản mới cho khách hàng
- Chức năng đăng nhập: Đăng nhập vào hệ thống
- Chức năng đăng xuất: Đăng xuất khỏi hệ thống
- Chức năng thay đổi mật khẩu: Thay đổi mật khẩu của khách hàng thông qua đăng nhập chính hoặc email xác thực
- Chức năng hiện thị menu món ăn: Xuất danh sách cách món ăn của nhà hàng trên hệ thống theo một thứ tự do khách hàng lựa chọn
- Chức năng lọc danh sách món ăn: Xuất danh sách món ăn của nhà hàng trên hệ thống thỏa tiêu chí của khách hàng đề ra
- Hiển thị chi tiết món ăn: Xuất thông tin về chi tiết món ăn.
- Đánh giá, phản hồi: Gửi đánh giá và phản hồi của khách hàng lên hệ thống

- Non-interactive requirements



- Trong quá trình hiển thị món ăn, công nghệ AI sẽ dựa trên sở thích khách hàng (được phân tích và rút ra thông qua lịch sử đặt món của khách hàng) và thị hiếu của khách hàng sử dụng hệ thống
- Thông kê, lưu trữ các thông tin liên quan đến khách hàng, lịch sử thành toán, lịch sử đơn hàng
- Trong quá trình thanh toán, giảm giá cho khách hàng sử dụng hệ thống vào ngày sinh nhật của khách và ngày khách đăng ký thành công trên hệ thống

Đối với nhân viên:

- Interactive requirements:
 - Hiển thị yêu cầu đặt bàn của khách hàng, xác nhận hoặc từ chối đặt bàn của khách
 - Hiển thị đơn đặt hàng của khách hàng.
 - Xác nhận, phản hồi chỉnh sửa hoặc từ chối đơn đặt hàng của khách hàng
- Non-interactive requirements: Các đơn đặt hàng, đặt bàn được hiển thị theo thứ tự độ ưu tiên và thời điểm khách hàng xác nhận đặt hàng

Đối với đầu bếp:

- Xuất thông tin đơn món ăn được xác nhận bởi nhân viên
- Gửi yêu cầu nguyên liệu cần thiết đang thiếu lên hệ thống để có thể được cung cấp nguyên liệu
- Cập nhật hoàn thành đơn món ăn

Đối với các stakeholders khác:

- Chủ nhà hàng có thể xem và xuất báo cáo doanh thu tổng và doanh thu các nhà hàng trong ngày, trong tháng và trong năm
- Quản lý nhà hàng có thể xem và xuất báo cáo doanh thu tổng và doanh thu nhà hàng của mình trong ngày, trong tháng và trong năm
- Chủ nhà hàng và quản lý nhà hàng có thể xem danh sách thông tin tài khoản, các lịch sử liên quan của khách hàng



- Chủ nhà hàng và quản lý nhà hàng có thể xem phản hồi, đánh giá của khách hàng
- Chủ nhà hàng và quản lý nhà hàng có thể xem thống kê các món ăn được order trong ngày, trong tháng và trong năm của khách hàng

1.2.2 Non-functional requirements

• Non-functional requirements:

- Hệ thống phải hoạt động được 24/7
- Các thao tác xác nhận, từ chối của khách hàng, nhân viên và đầu bếp phải được thực hiện trong tối đa 100ms
- Đầu bếp cập nhật tình trạng món ăn lên hệ thống trong tối đa 100ms
- Thời gian truy cập hệ thống thông qua url hoặc QR code không quá 1s
- Giao diện thân thiện: Nhân viên có thể sử dụng thành thạo các chức năng liên quan của mình trong vòng 1 phút. User có thể sử dụng để đạt được mục đích mình muốn sau các hướng dẫn 1 lần
- Hiệu chỉnh, thay đổi hệ thống chỉ có thể được thực hiện bởi chủ nhà hàng, quản lý nhà hàng hoặc quản lý hệ thống. Các stakeholders khác không thể can thiệp hay hiệu chỉnh hệ thống
- Đơn vị tiền tệ sử dụng cho hệ thống là VND
- Thời gian thực hệ thống được lấy theo GMT+7
- Gửi email, tin nhắn quảng cáo đến khách hàng mỗi tháng
- Có thể chạy trên các trình duyệt web nooit tiếng (Safari, Chrome, Opera, Microsoft Edge,...)

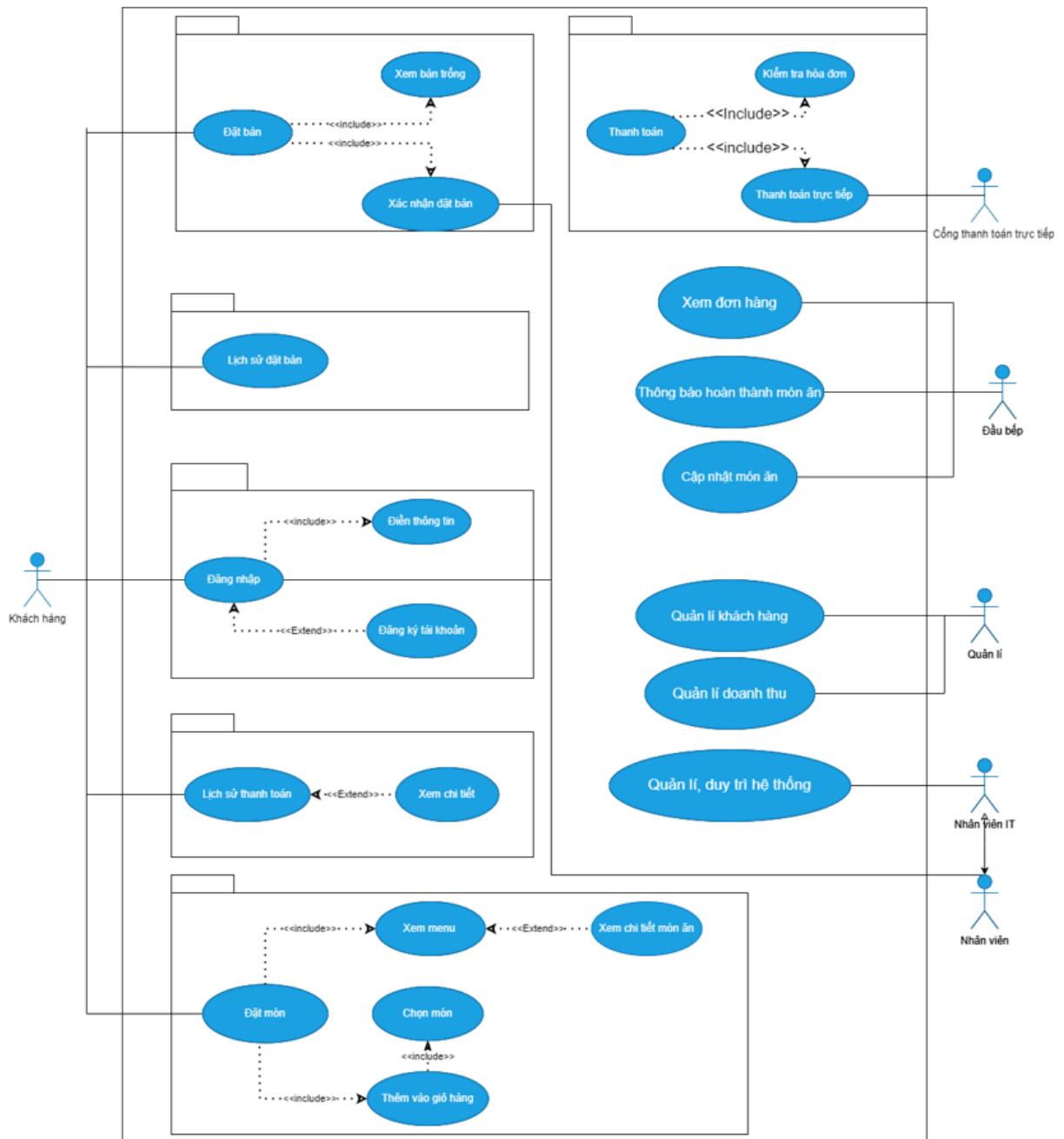
• Individual Non-functional requirements:

- Đăng ký tài khoản: Mật khẩu phải có 8 số, ít nhất một ký tự viết hoa, một ký tự bình thường, một ký tự số, một ký tự đặc biệt
- Đăng nhập: Xác thực đăng nhập tối đa 500ms và trả về trang chủ trong 0.5s
- Đánh giá, phản hồi:
 - * Form phản hồi dễ hiểu, dễ thực hiện. Khách hàng có thể viết phản hồi và gửi ngay khi mới sử dụng
 - * Thời gian gửi form về hệ thống tối đa 5s
- Đặt bàn:



- * Sau khi gửi đặt bàn thành công trở về trang chủ trong vòng 0.5s
 - * Cập nhật thông tin về bàn trống trong 0.5s
 - * Xác nhận đặt bàn thành công tối đa 1s
- Thanh toán:
- * Thực hiện giao dịch với ngân hàng trong tối đa 500ms
 - * Phản hồi thành công hoặc thất bại giao dịch trong tối đa 500ms

1.2.3 Use-case diagram of the whole system



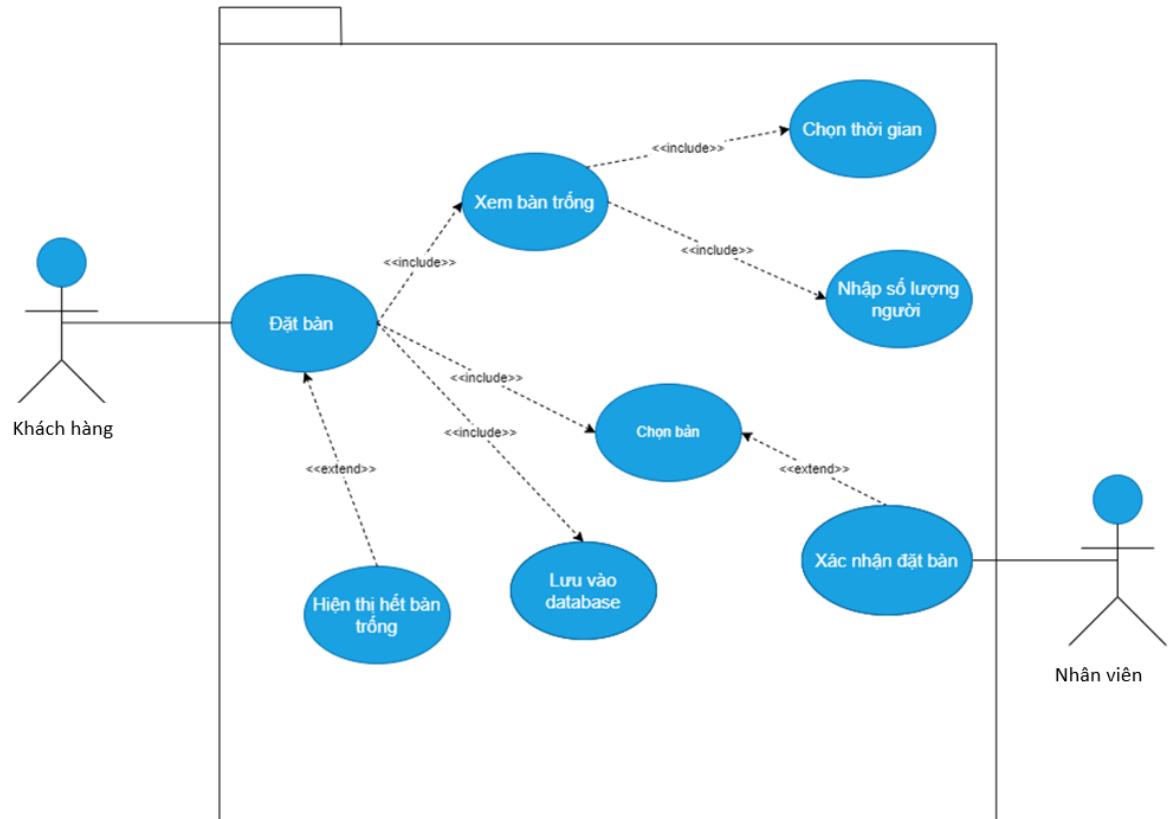
Hình 1: Use case diagram of whole system



1.3 Use-case diagrams and description of some features

1.3.1 Use-case : Đặt bàn

Đặt bàn	
Actors	Khách hàng và nhân viên nhà hàng.
Descriptions	Giúp khách hàng có bàn sẵn khi đến quán ăn, giúp khách không phải lãng phí thời gian để chờ đợi hoặc 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 (thanh toán tại nhà hàng trên 10 lần)) và ấn 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	1. Đặt bàn thành công, hệ thống lưu trữ bàn đặt cho khách hàng. 2. Đặt bàn không thành công, hiển thị "Xin lỗi quý khách, đã hết bàn trống".
Normal Flow	1. Khách hàng ấn vào biểu tượng "Đặt bàn". 2. Hệ thống hiện thị form đặt bàn. 3. Khách hàng chọn thời gian, số lượng khách. 4. Khách hàng nhấn nút "Tìm kiếm". 5. Hệ thống hiển thị bàn còn trống, khách hàng ấn vào vị trí bàn mong muốn. 6. Khách hàng bấm xác nhận đặt bàn. 7. Nhân viên xác nhận đặt bàn. 8. Hệ thống lưu trữ bàn đặt cho khách hàng, hiển thị đặt bàn hoàn tất.
Alternative Flows	None
Exceptions Flow	Tại bước 5: Cửa hàng đã hết bàn theo yêu cầu của khách hàng. Dưa ra câu hiển thị "Xin lỗi quý khách, đã hết bàn trống". 5.1 Khách hàng chọn lại thời gian đặt bàn và số lượng 5.1a. Khách hàng chọn đặt lại. 5.1b. Quay trở lại bước 3. 5.2 Khách hàng kết chọn kết thúc đặt bàn. 5.2a. Hệ thống hiển thị "Xin lỗi quý khách, hẹn quý khách lần sau". 5.2b. Kết thúc use-case.
Comments	None

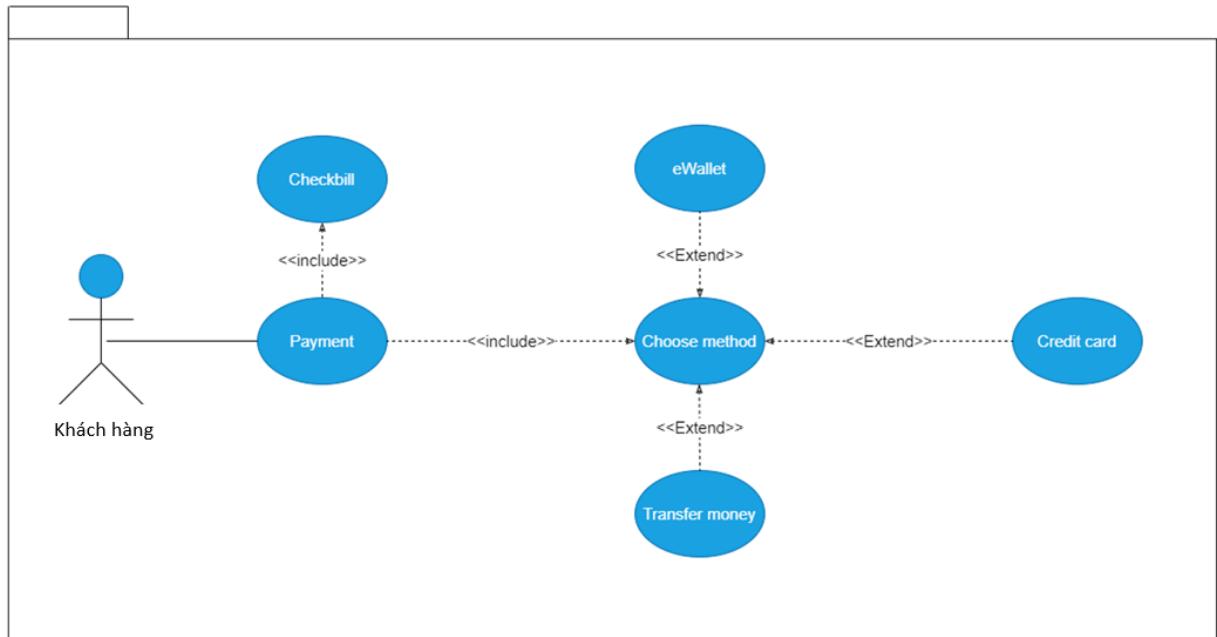


Hình 2: Use case diagram of table reservation



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	Khách hàng đã đăng nhập vào hệ thống thành công.
Postcondition	Khách hàng hoàn tất thanh toán thành công.
Normal Flow	<ol style="list-style-type: none">Sau khi kiểm tra giỏ hàng, khách hàng nhấn vào biểu tượng "Thanh Toán".Hệ thống hiển thị hóa đơn, khách hàng kiểm tra lại hóa đơn.Khách hàng chọn phương thức thanh toán.Người dùng điền những thông tin cá nhân.Khách hàng chọn xác nhận thanh toán và nhập mã pin để xác thực.Hệ thống lưu record đơn hàng vào database, đồng thời phản hồi thanh toán thành công.Hoàn tất thanh toán.
Alternative Flows	Tại bước 3: Khách hàng muốn thanh toán trực tiếp bằng tiền mặt. 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 Flow	Tại bước 2: Hóa đơn không chính xác 2.1. Gửi yêu cầu nhân viên kiểm tra lại thông tin hóa đơn. 2.2. Hóa đơn đã được chỉnh sửa và tiếp tục bước 3. Tại bước 4: Loại ví điện tử đã chọn hiện đang bảo trì hệ thống. 4.1. Ứng dụng hiển thị "Thanh Toán thất bại do hệ thống ví điện tử đang được bảo trì". 4.2. Hệ thống yêu cầu người dùng chọn lại phương thức thanh toán và tiếp tục bước 4 trong Normal Flows. Tại bước 5: Số dư trong ví không đủ để thanh toán hóa đơn. 5.1. Ứng dụng hiển thị "Thanh toán thất bại do số dư không đủ". 5.2. Hệ thống yêu cầu người dùng chọn lại phương thức thanh toán và tiếp tục bước 4 trong Normal Flows.
Comments	None

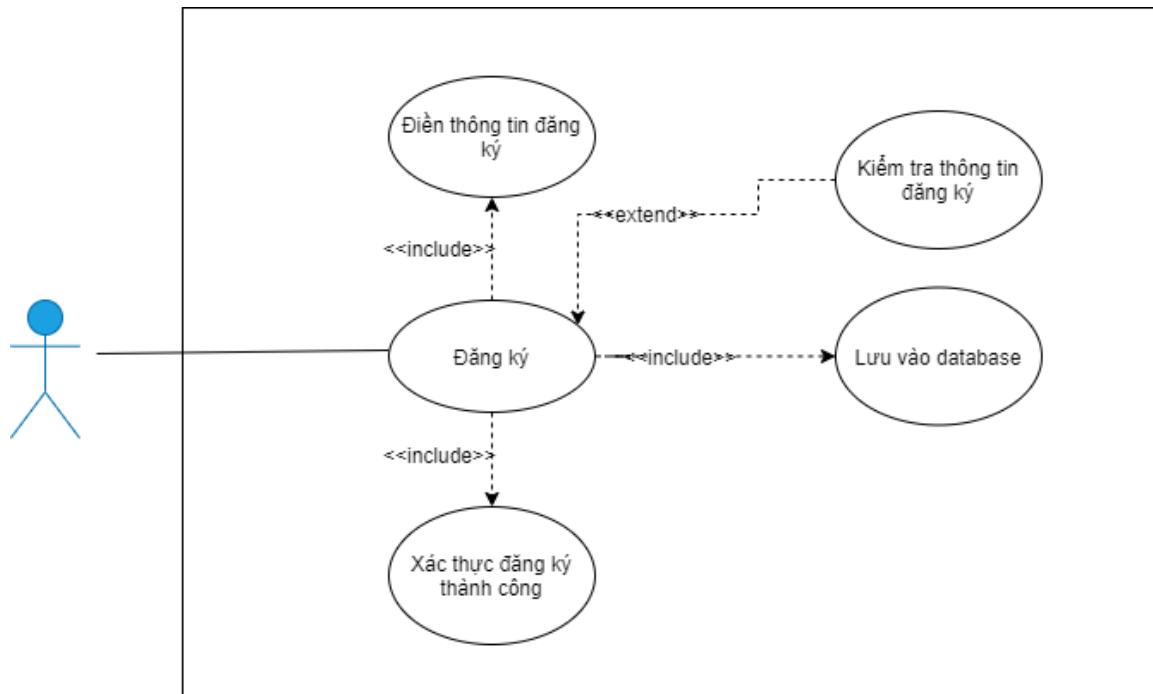


Hình 3: Use case diagram of Payment



1.3.3 Use-case : Đăng Ký

Đăng Ký	
Actors	Khách hàng và nhân viên nhà hàng.
Descriptions	Khách hàng đăng ký để sử dụng hệ thống và nhận nhiều ưu đãi cũng như thông báo mới từ nhà hàng.
Triggers	Khách hàng nhấn chọn đăng ký từ trang đăng nhập.
Precondition	Truy cập được vào trang chủ hệ thống, trang web đang hoạt động trong giờ làm việc.
Postcondition	Đăng ký tài khoản thành công.
Normal Flow	<ol style="list-style-type: none">Khách hàng ấn vào biểu tượng "Đăng ký".Hệ thống hiện thị form đăng ký tài khoản.Khách hàng nhập thông tin tài khoản.Khách hàng ấn vào biểu tượng "Hoàn thành đăng ký".Hệ thống kiểm tra thông tin đăng ký.Hệ thống lưu thông tin tài khoản vào database.Hệ thống xác thực đăng ký thành công, cho phép tài khoản khách hàng vừa đăng ký có thể đăng nhập vào hệ thống.Kết thúc use case
Alternative Flows	Tại bước 6: Do thông tin tài khoản đăng ký bị thiếu, chứa ký tự đặc biệt hoặc đã được sử dụng nên hệ thống sẽ hiển thị lỗi tương ứng. 6.1 Quay lại bước 3.
Exceptions Flow	Tại bước 1: Hệ thống đang được bảo trì và nâng cấp. 1.1 Hiển thị: Hiện tại hệ thống đang được bảo trì, vui lòng thử lại sau. 1.2 Người dùng thoát ra khỏi hệ thống. 1.3 Kết thúc use-case.
Comments	None

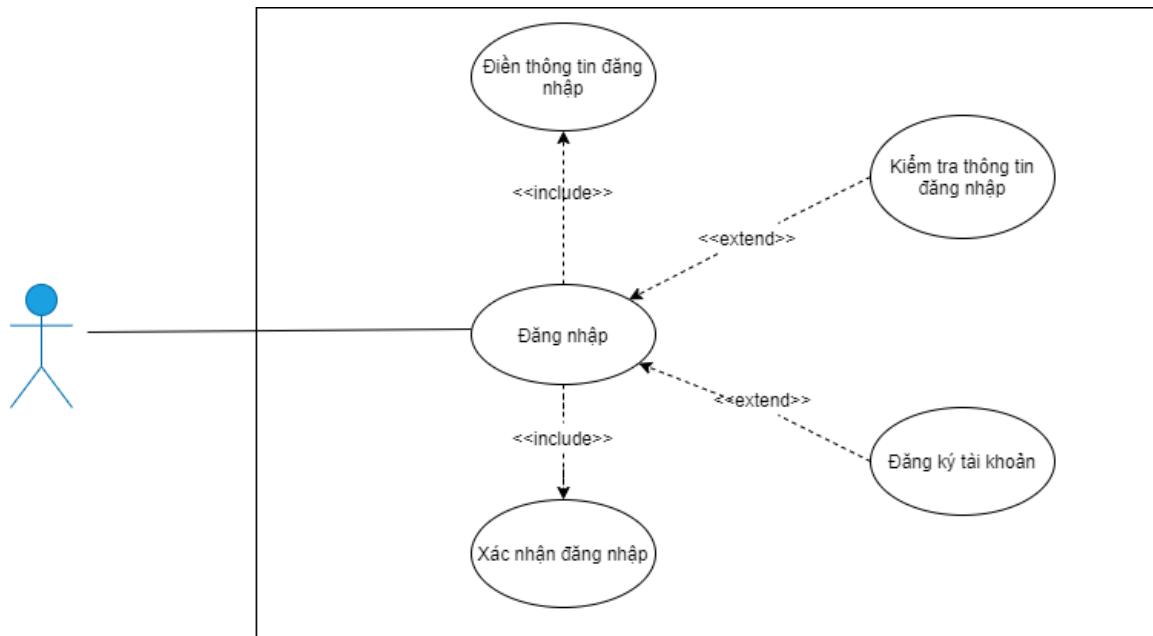


Hình 4: Use case diagram of sign up



1.3.4 Use-case : Đăng Nhập

Đăng Nhập	
Actors	Khách hàng và nhân viên nhà hàng.
Descriptions	Khách hàng đăng nhập vào hệ thống để được sử dụng những dịch vụ của nhà hàng.
Triggers	Khách hàng ấn vào biểu tượng "Đăng nhập" từ trang chủ của hệ thống.
Precondition	Truy cập được vào trang chủ hệ thống, trang web đang hoạt động trong giờ làm việc.
Postcondition	Đăng nhập thành công.
Normal Flow	<ol style="list-style-type: none">Khách hàng ấn vào biểu tượng "Đăng nhập".Hệ thống hiện thị form đăng nhập.Khách hàng nhập tên tài khoản và mật khẩu.Khách hàng ấn vào biểu tượng "Đăng nhập".Hệ thống kiểm tra thông tin tài khoản.Hệ thống thông báo đăng nhập thành công và đưa người dùng về trang chủ.Kết thúc use case
Alternative Flows	Tại bước 3: Khách hàng ấn vào biểu tượng "Trở lại". 3.1 Dưa khách hàng về trang menu. 3.2 Kết thúc use case
Exceptions Flow	Tại bước 1: Hệ thống đang được bảo trì và nâng cấp. 1.1 Hiển thị: Hiện tại hệ thống đang được bảo trì, vui lòng thử lại sau. 1.2 Người dùng thoát ra khỏi hệ thống. 1.3 Kết thúc use-case. Tại bước 6: Do thông tin đăng nhập không chính xác. 6.1 Hiển thị: Mật khẩu không hợp lệ hoặc tài khoản không tồn tại. 6.2 Quay lại bước 3.
Comments	None

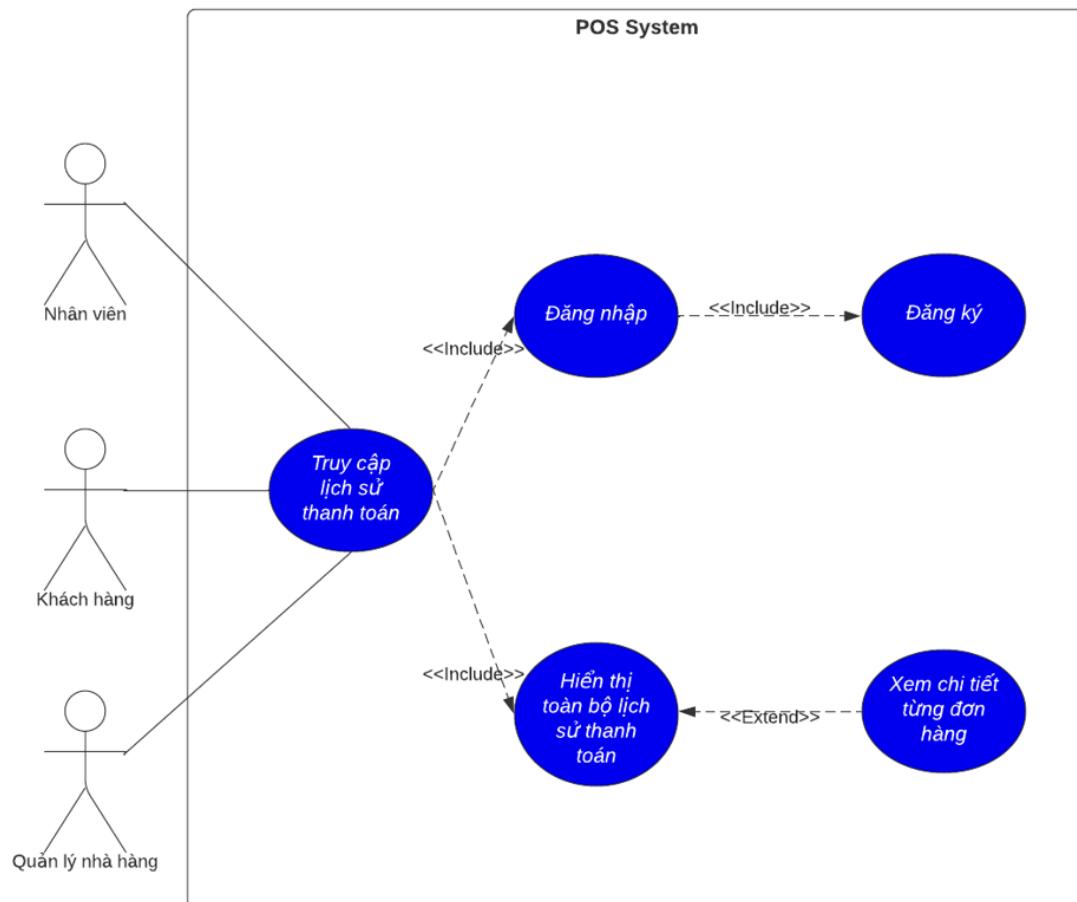


Hình 5: Use case diagram of sign in



1.3.5 Use-case : Xem lịch sử thanh toán

Truy cập lịch sử thanh toán	
Actors	Khách hàng, nhân viên, quản lý nhà hàng. .
Descriptions	<p>Khách hàng có thể truy cập lịch sử thanh toán của các đơn hàng họ đã đặt.</p> <p>Nhân viên và quản lý nhà hàng có thể truy cập lịch sử thanh toán của tất cả các đơn hàng.</p>
Triggers	Người muốn truy cập cần phải có tài khoản.
Precondition	Hệ thống kích hoạt Usecase Đăng nhập.
Postcondition	Xem được lịch sử thanh toán, chi tiết và ngày giờ của mỗi đơn hàng trong quá khứ.
Normal Flow	<ol style="list-style-type: none">Người dùng Đăng nhập thành côngNgười dùng 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.Người dùng nhấp vào đơn hàng mà họ muốn xem chi tiết.Hệ thống sẽ hiển thị danh sách món ăn và tổng tiền thanh toán của hoá đơn đó.
Alternative Flows	Tại bước 1 : Nếu người dùng chưa có tài khoản thì sẽ cần phải đăng ký tài khoản.
Exceptions Flow	Tại bước 1: Hệ thống bảo trì. 1.1. Hệ thống thông báo bảo trì...
Comments	Không.

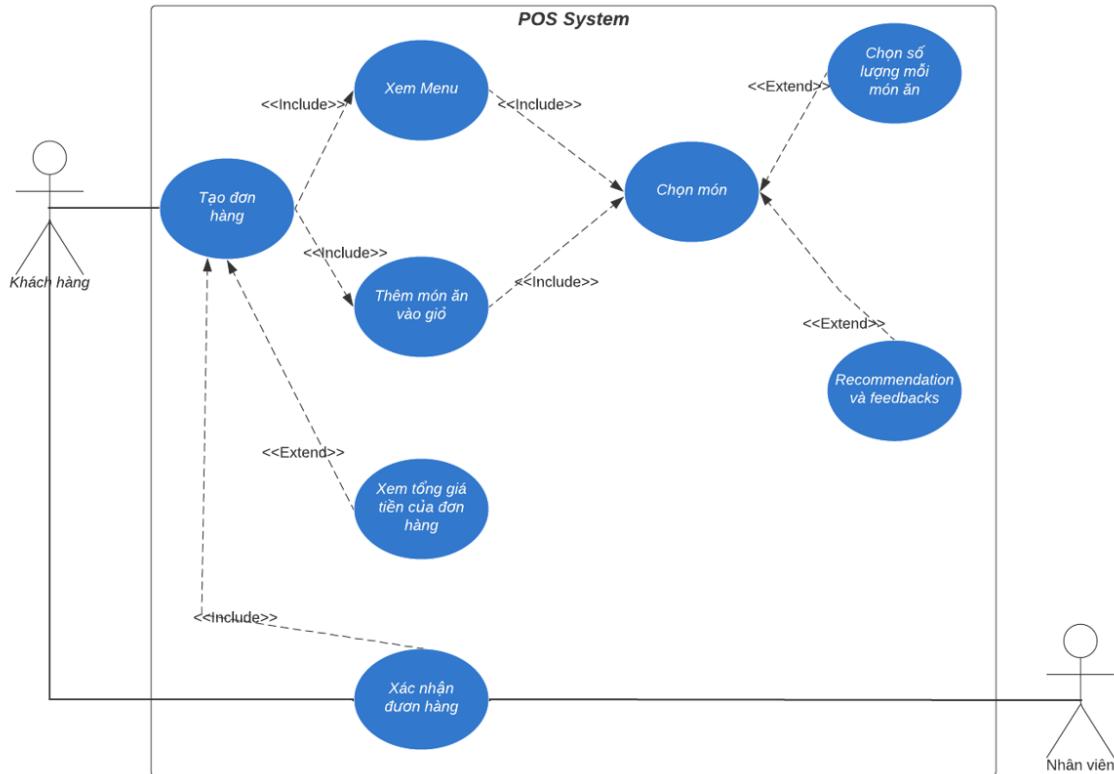


Hình 6: Use case diagram of payment history viewing



1.3.6 Use-case : Tạo đơn hàng

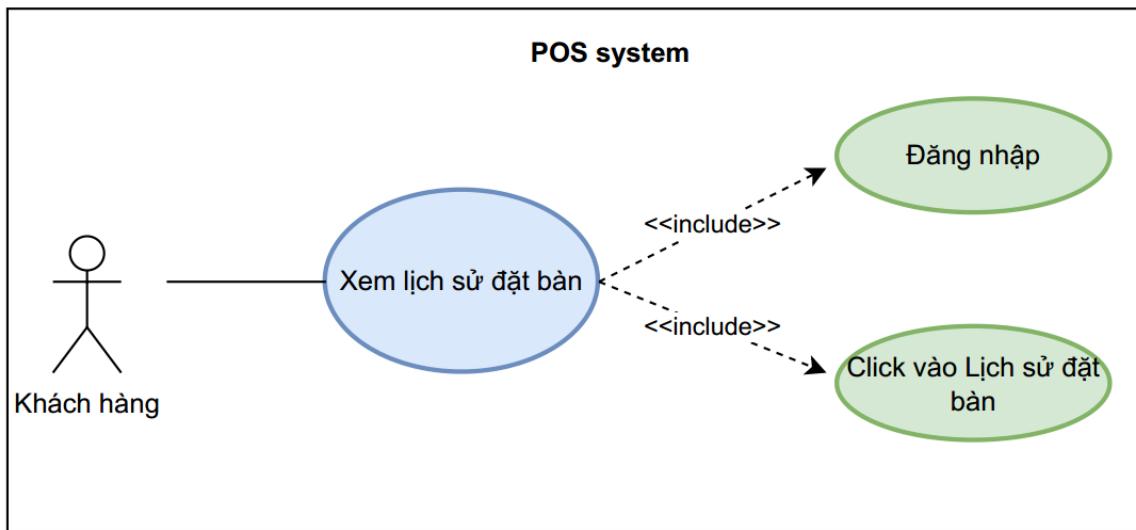
Xem Menu và thêm món ăn vào giỏ	
Actors	Khách hàng .
Descriptions	Khách hàng xem và thêm món ăn vào giỏ hàng.
Triggers	Khách hàng nhấp vào “Menu”.
Precondition	Không.
Postcondition	Khách hàng sẽ được thông báo nếu đặt hàng thành công.
Normal Flow	<ol style="list-style-type: none">Khách hàng chọn món ăn dựa trên Menu của nhà hàng.Khách hàng nhấp vào “Thêm vào giỏ” tại món ăn mà mình chọn.Hệ thống tiến hành thêm món ăn vào giỏ và tính tiền các món ăn.Khách hàng và nhân viên cùng xác nhận đơn hàng.
Alternative Flows	Tại bước 2: 2,1. Khi đang chọn món thì khách hàng có thể xem các món ăn được đề cử, best-seller hay feedbacks từ những người khác. 2,2. Trên form thông tin món ăn, khách hàng có thể xem thông tin chi tiết về món ăn, chọn số lượng . . . 2,3. Nếu không hài lòng với đơn đặt hàng, khách hàng có thể ấn dấu ‘X’ ở góc trên bên phải màn hình để quay lại Menu.
Exceptions Flow	Tại bước 3: Nếu nhà hàng không phục vụ món khách hàng đã chọn thì khách hàng sẽ nhận được thông báo và được chuyển về Menu để chọn lại món ăn.
Comments	Khách hàng có thể hủy món đã chọn trong giỏ hàng trước khi thanh toán nếu không muốn chọn nữa.



Hình 7: Use case diagram of food ordering

1.3.7 Use-case : Duyệt lịch sử đặt bàn

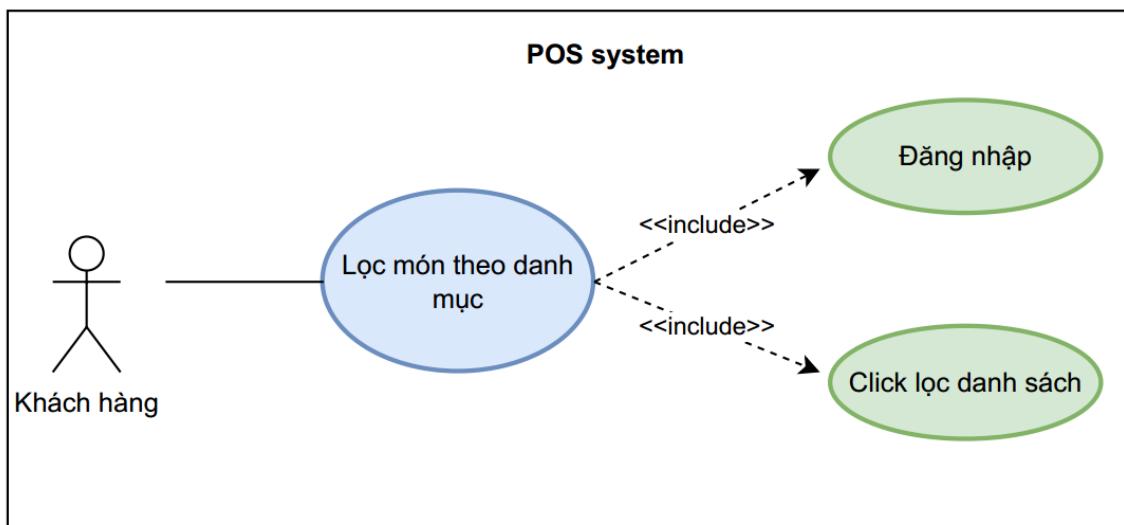
Duyệt lịch sử đặt bàn	
Actors	Khách hàng
Descriptions	Khách hàng có thể kiểm tra thông tin các lần đặt bàn tại nhà hàng với các mốc thời gian cụ thể.
Triggers	Khách hàng nhấp vào "Lịch sử đặt bàn".
Precondition	Khách hàng đăng nhập thành công vào hệ thống.
Postcondition	Khách hàng xem được lịch sử các lần đặt bàn.
Normal Flow	<ol style="list-style-type: none"> Khách hàng click vào "Lịch sử đặt bàn" trên UI. Hệ thống hiển thị kết quả trả về đến khách hàng sắp xếp theo thời gian gần nhất với thời điểm hiện tại. Use case kết thúc.
Alternative Flows	Tại bước 1: Kiểm tra đăng nhập <ol style="list-style-type: none"> Hệ thống kiểm tra khách hàng đã đăng nhập chưa <ol style="list-style-type: none"> Chưa đăng nhập. Chuyển hướng đến trang đăng nhập. Đã đăng nhập. Use case tiếp tục bước 2.
Exceptions Flow	Tại bước 1: Hệ thống bảo trì <ol style="list-style-type: none"> Hệ thống hiển thị thông báo ra màn hình. Kết thúc Use-case.
Comments	None



Hình 8: Use case diagram of view history of booking

1.3.8 Use-case : Duyệt món ăn theo danh mục món ăn

Duyệt món ăn theo danh mục món ăn	
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 truy cập thành công vào trang chủ.
Postcondition	Khách hàng xem danh sách món ăn theo các danh mục đã lọc.
Normal Flow	<ol style="list-style-type: none"> Khách hàng tick vào các checkbox danh mục món trên giao diện. Khách hàng click vào button "Lọc danh sách" Hệ thống hiển thị kết quả danh sách món trả về đến khách hàng sắp xếp theo các danh mục đã chọn. Use-case kết thúc.
Alternative Flows	Tại bước 2: Khách hàng không chọn vào bất cứ danh mục nào trước khi bấm nút "Lọc danh sách" <ol style="list-style-type: none"> Hệ thống mặc định chọn toàn bộ danh mục. Use-case tiếp tục bước 3.
Exceptions Flow	Tại bước 1: Hệ thống bảo trì <ol style="list-style-type: none"> Hệ thống hiển thị thông báo ra màn hình. Kết thúc Use-case.
Comments	None



Hình 9: Use case diagram of filter dishes by category

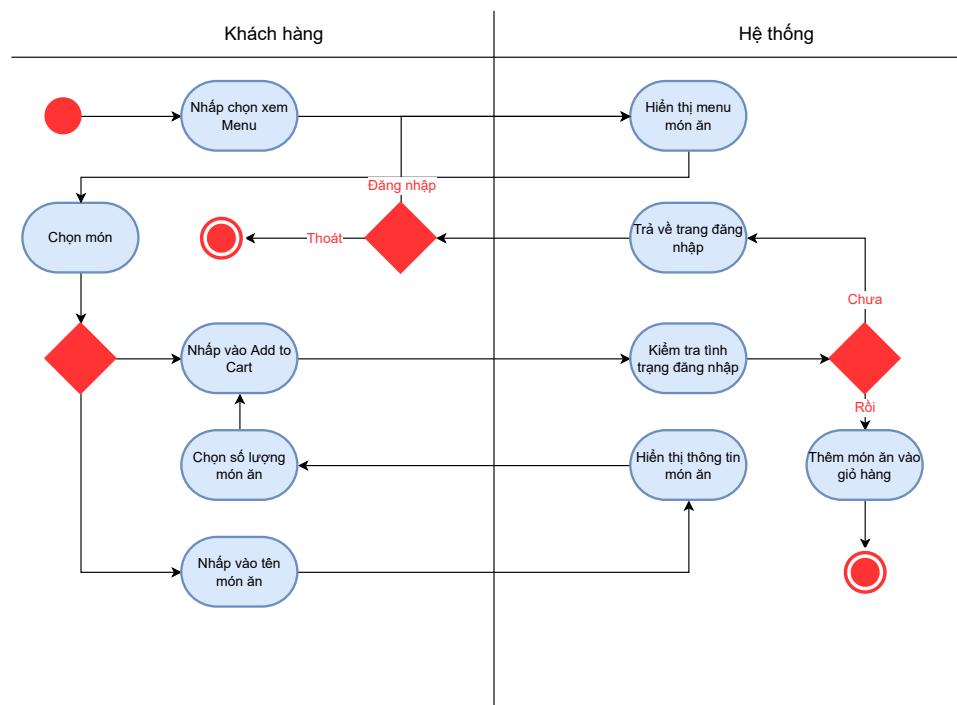


Trường Đại học Bách Khoa, TP. Hồ Chí Minh
Khoa Khoa Học & Kỹ Thuật Máy Tính

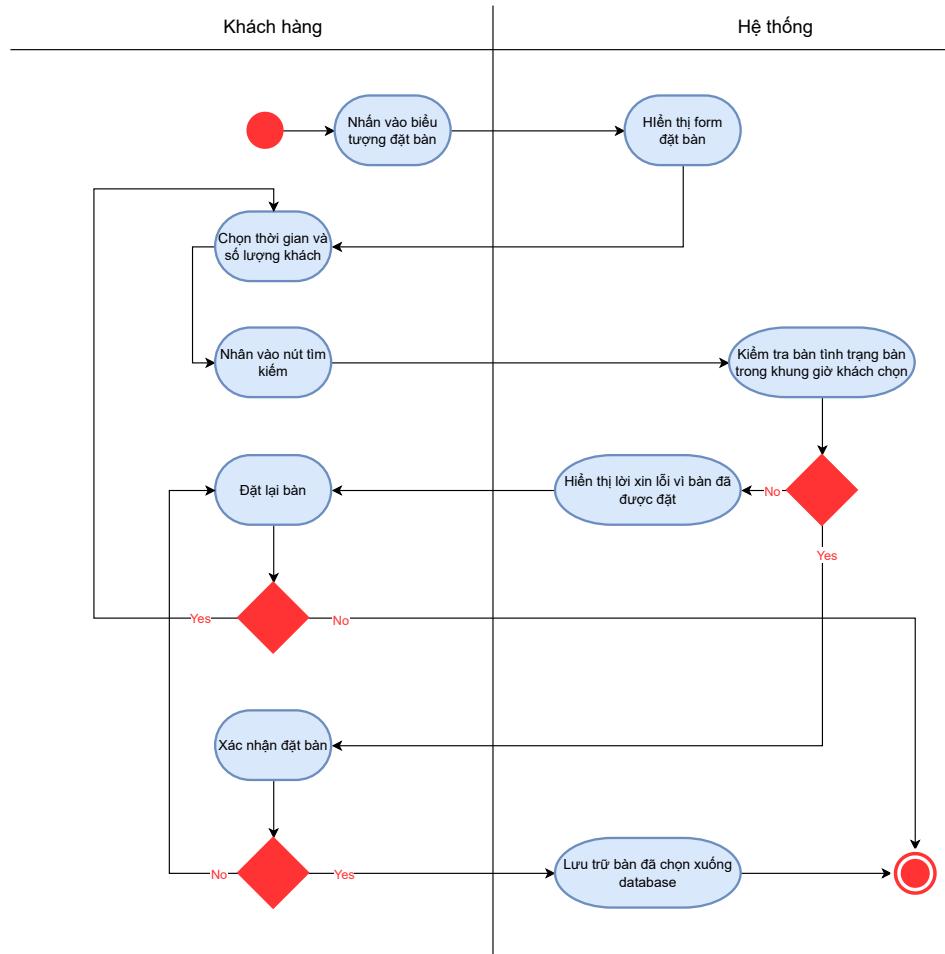
2 System Modeling

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

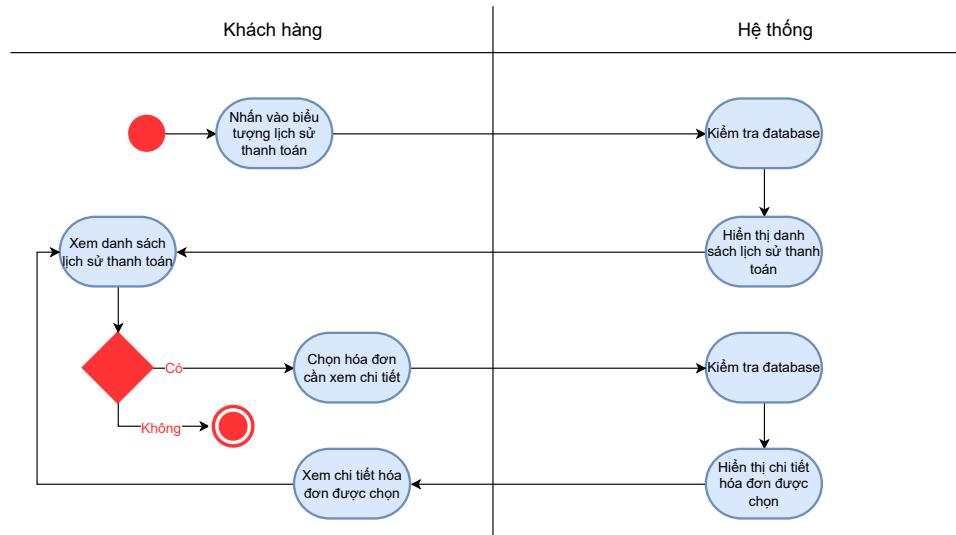
2.1.1 Xem Menu, chi tiết món ăn và thêm vào giỏ hàng



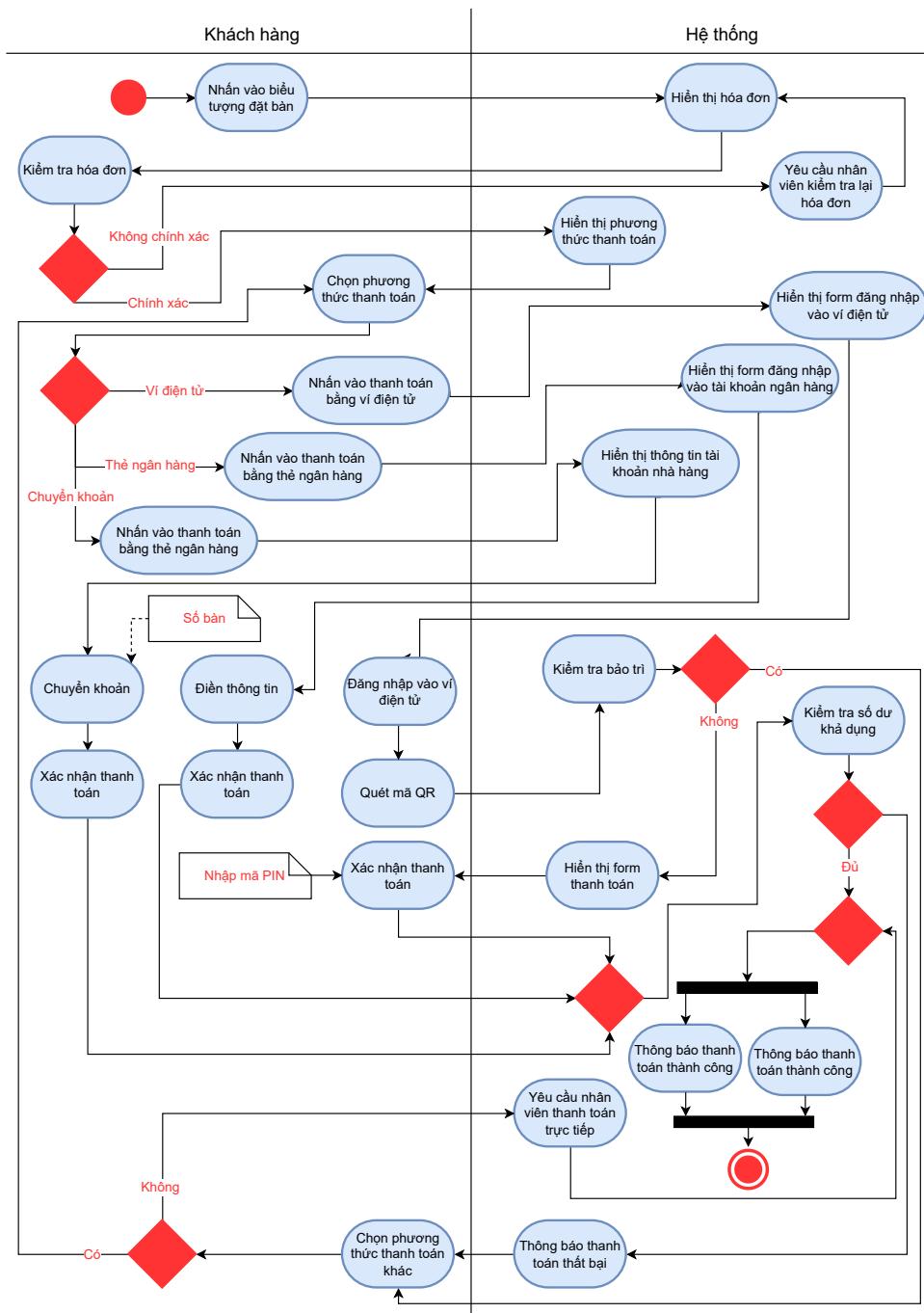
2.1.2 Đặt bàn



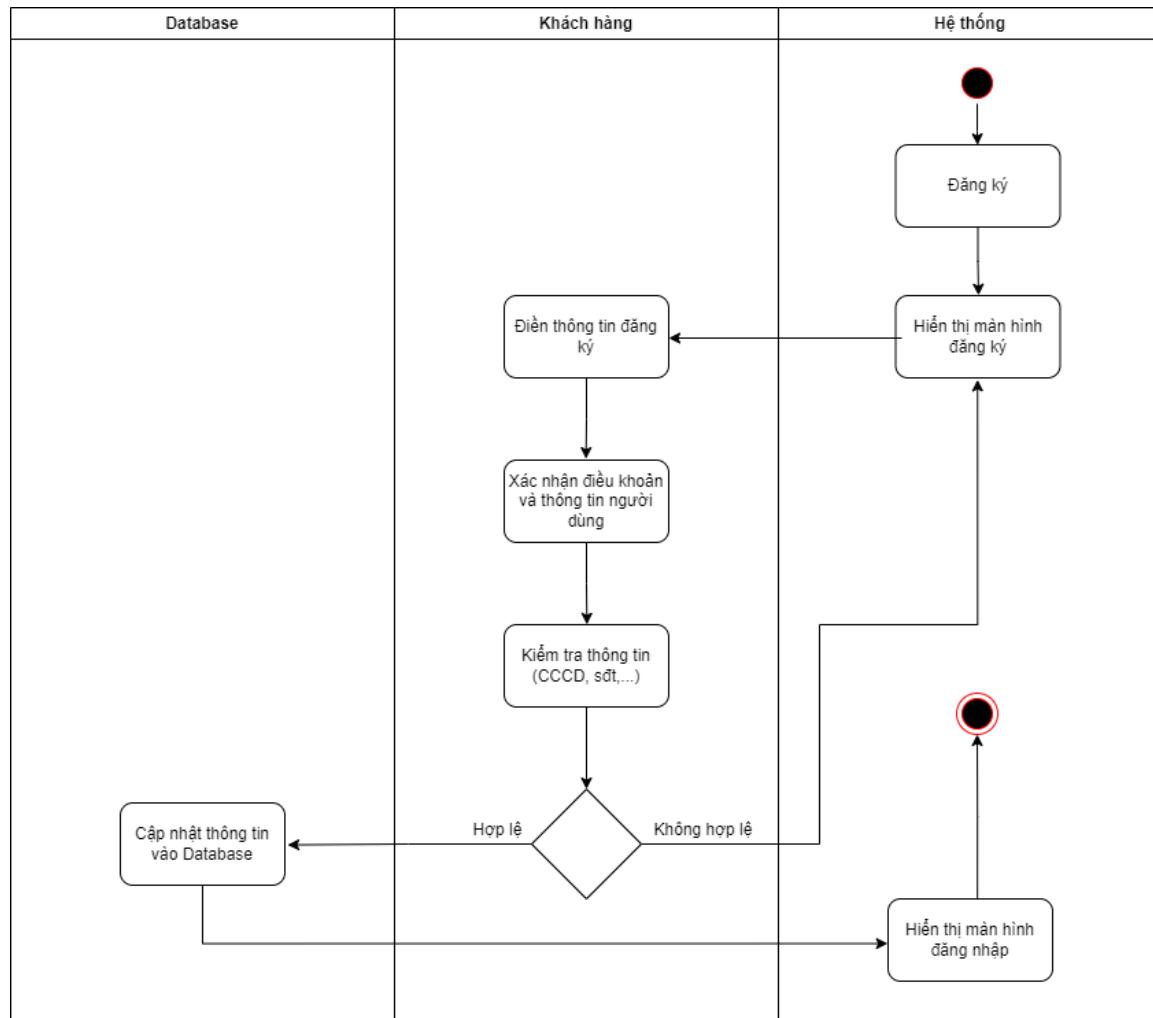
2.1.3 Xem lịch sử thanh toán



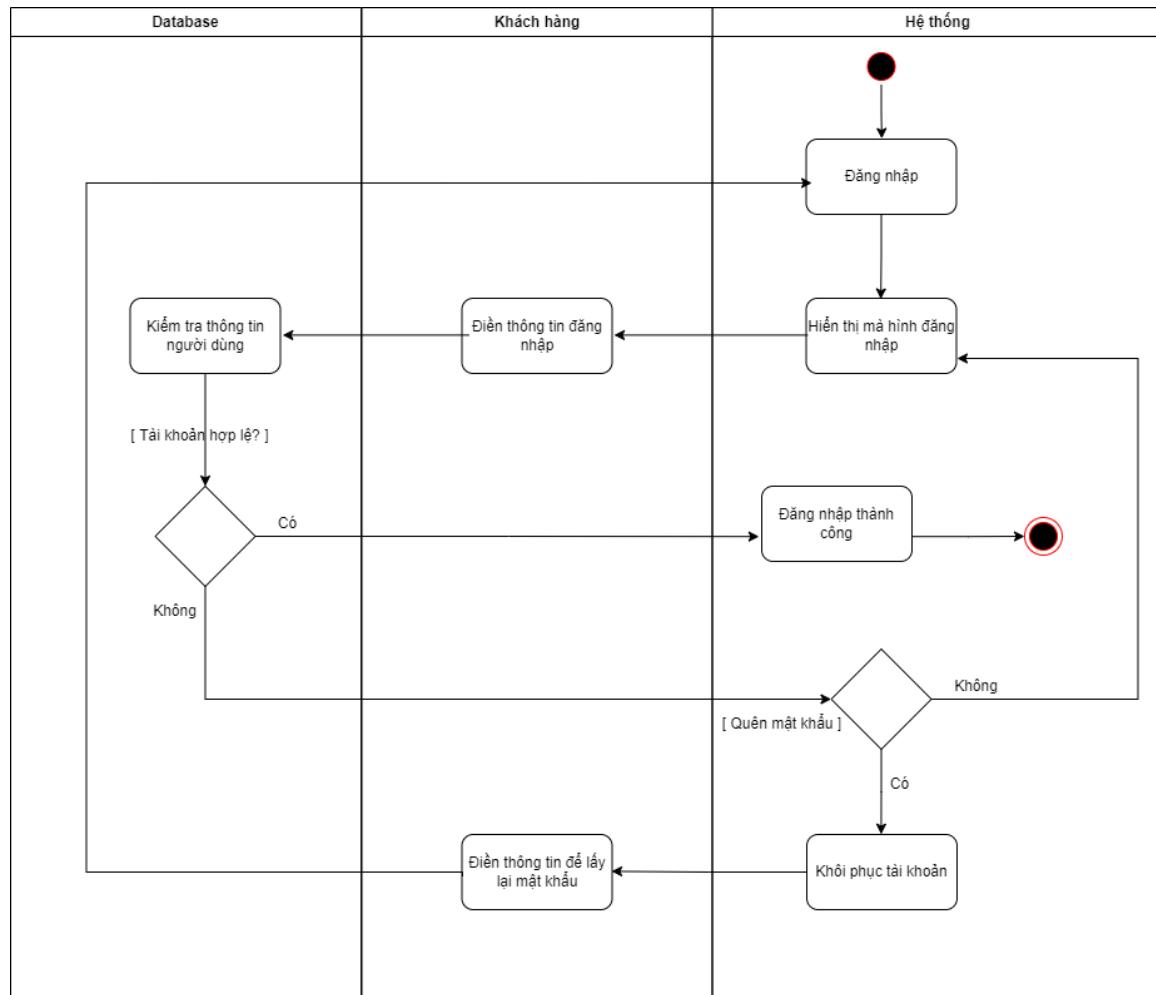
2.1.4 Thanh toán



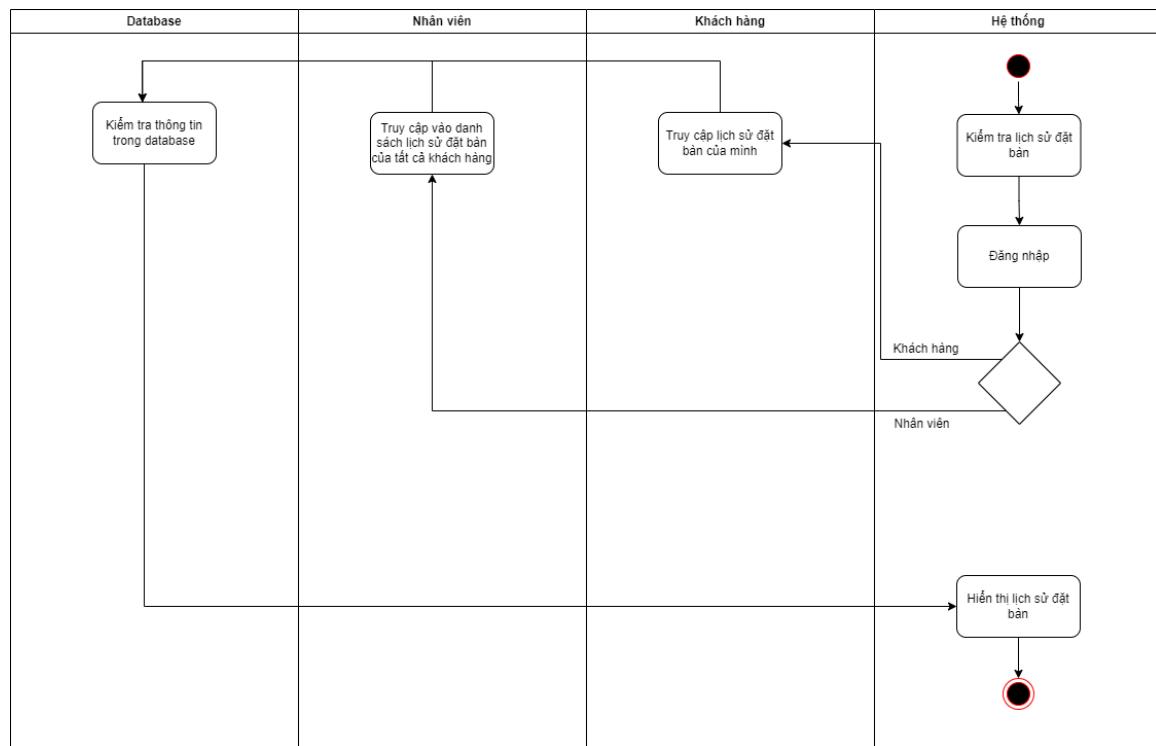
2.1.5 Đăng ký



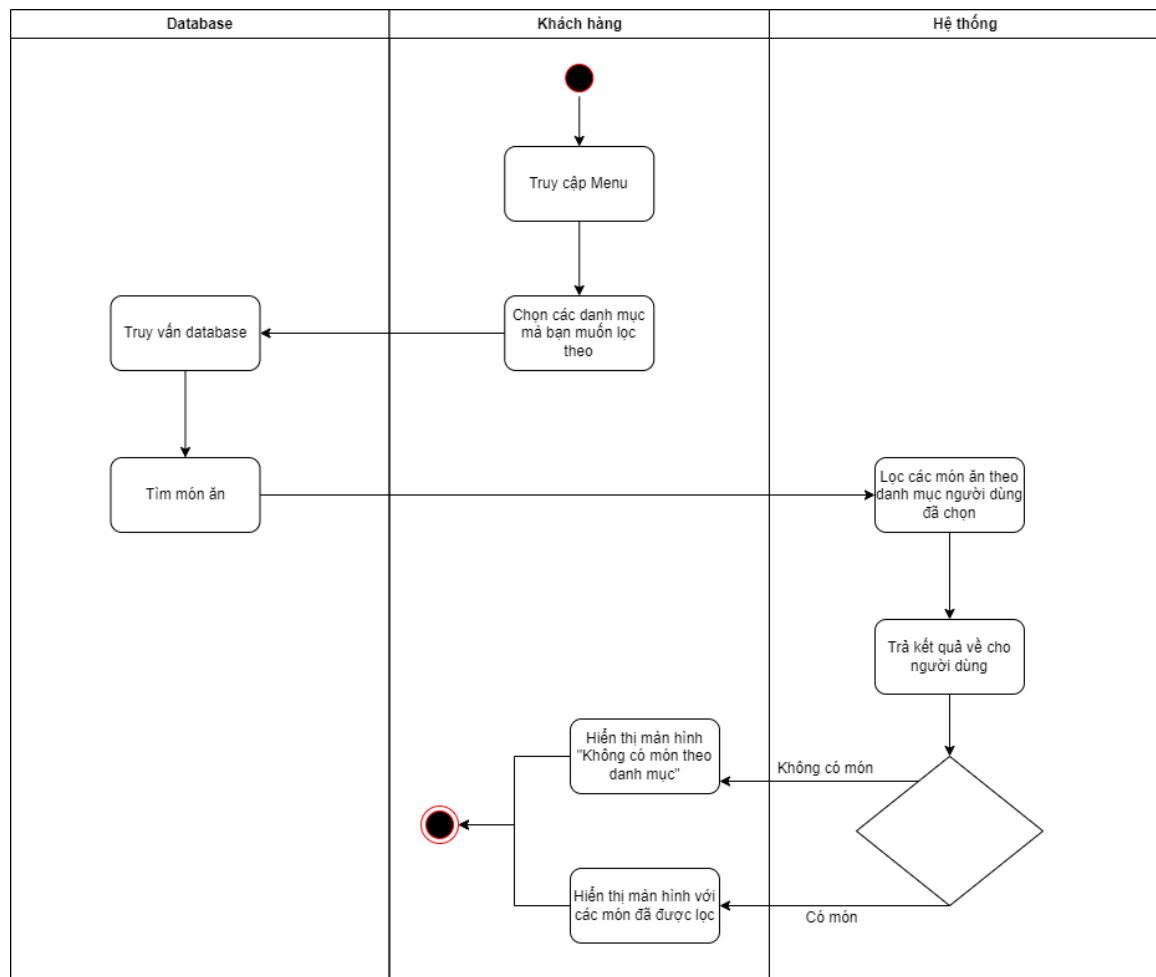
2.1.6 Đăng nhập



2.1.7 Kiểm tra lịch sử đặt bàn

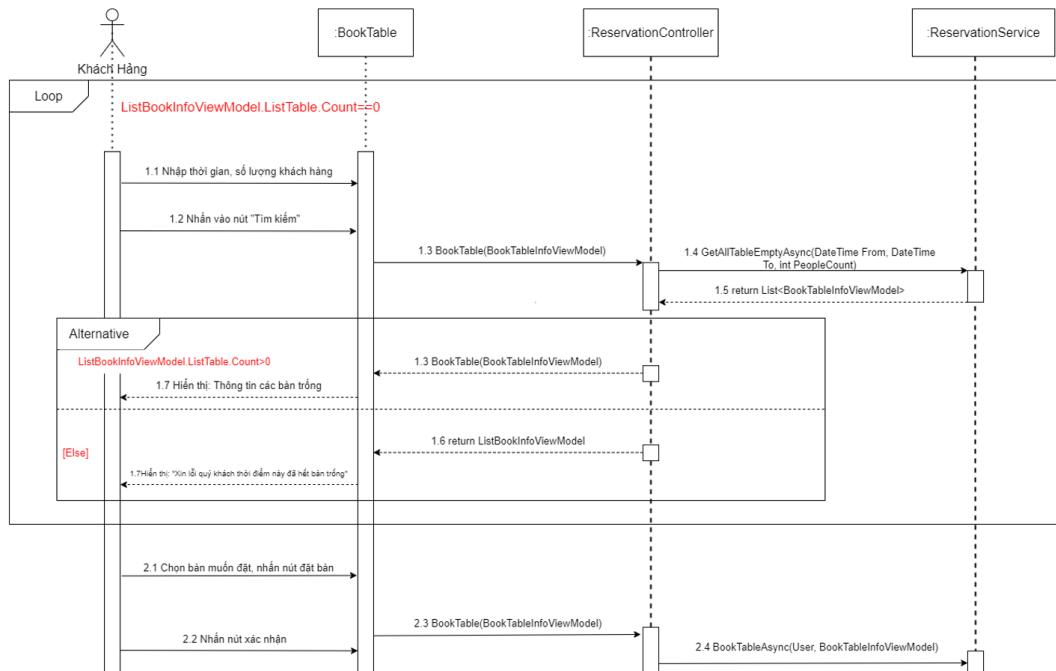


2.1.8 Kiểm tra lịch sử đặt bàn



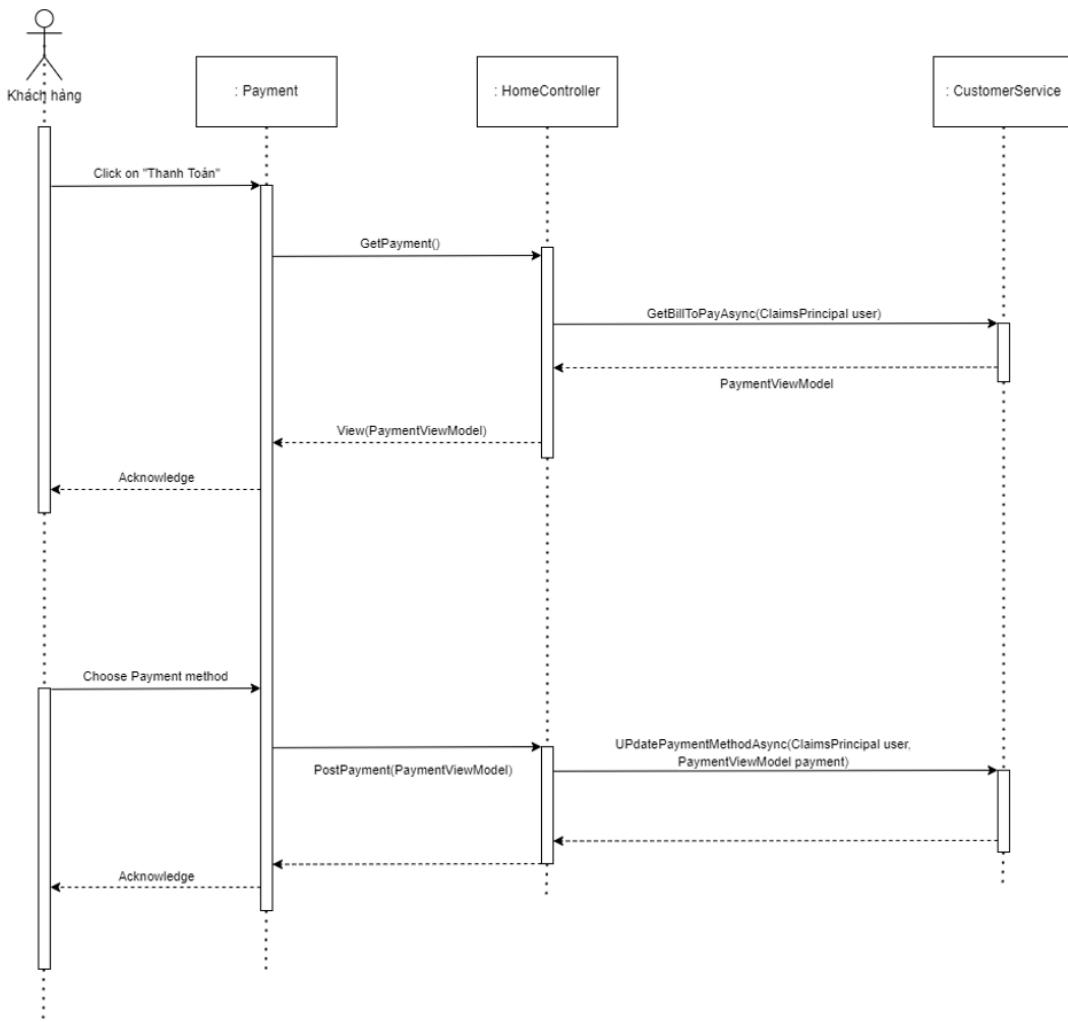
2.2 Sequence diagram for use-case in Task 1.3.

2.2.1 Đặt bàn



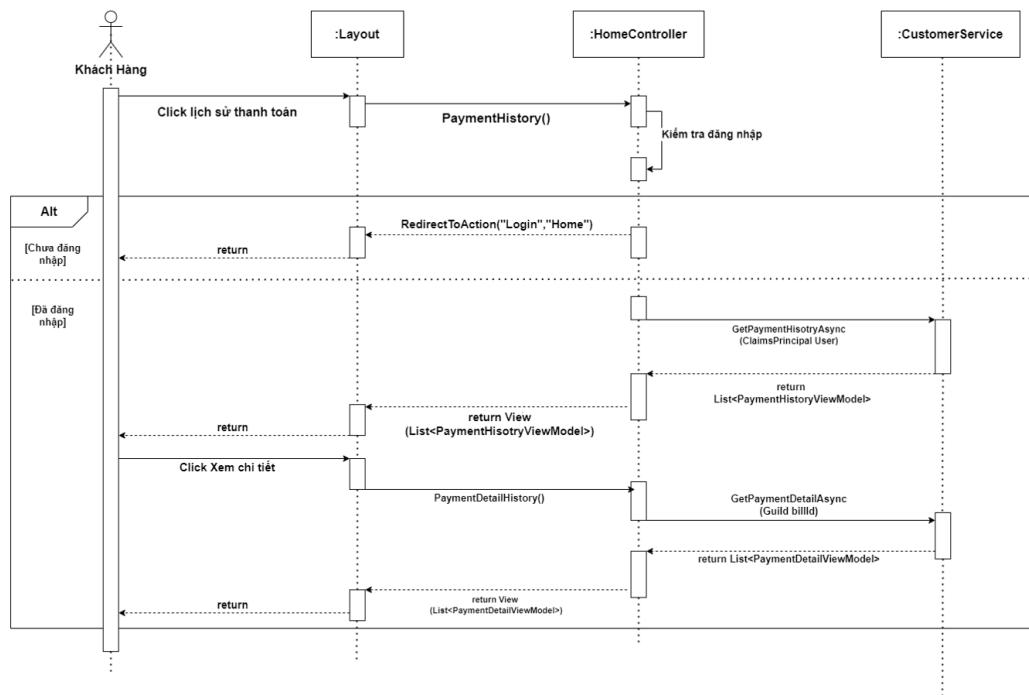
Hình 10: Sequence diagram của use case: Reservation

2.2.2 Thanh toán



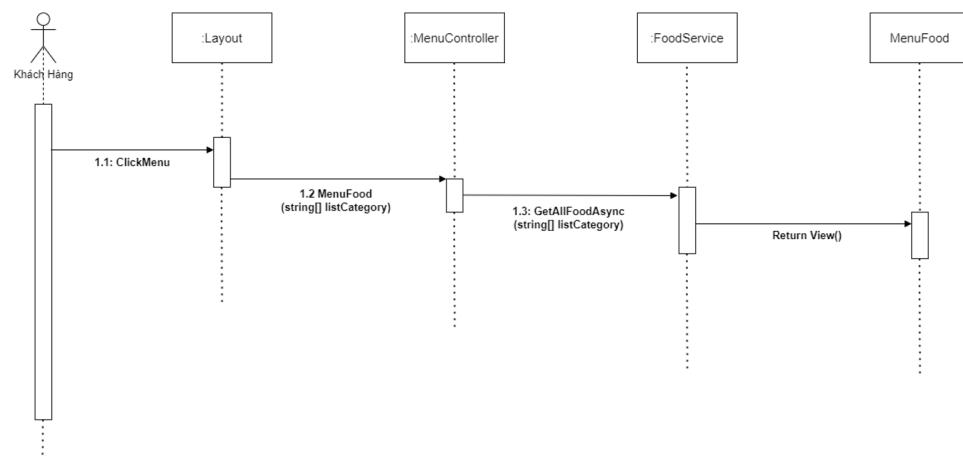
Hình 11: Sequence diagram của use case: Payment

2.2.3 Lịch sử thanh toán

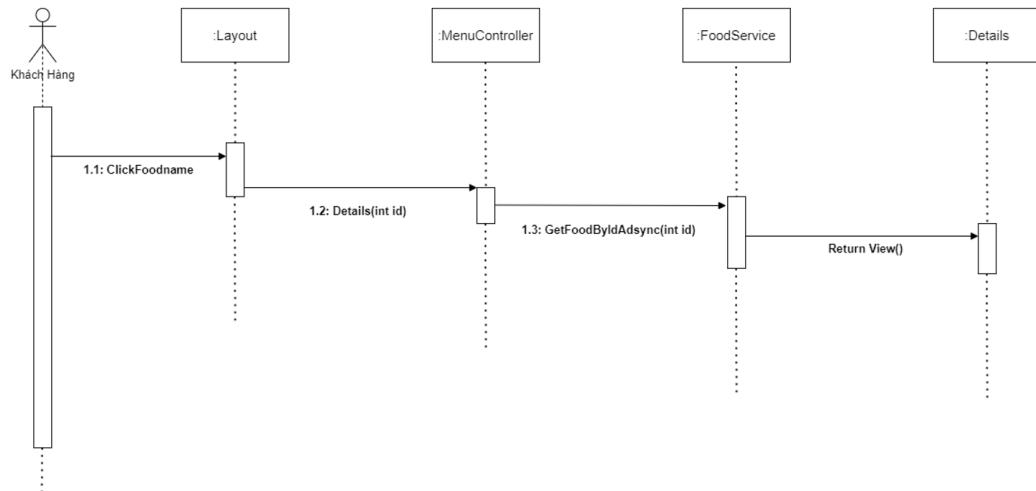


Hình 12: Sequence diagram của use case: Lịch sử thanh toán

2.2.4 Xem menu và xem chi tiết món ăn

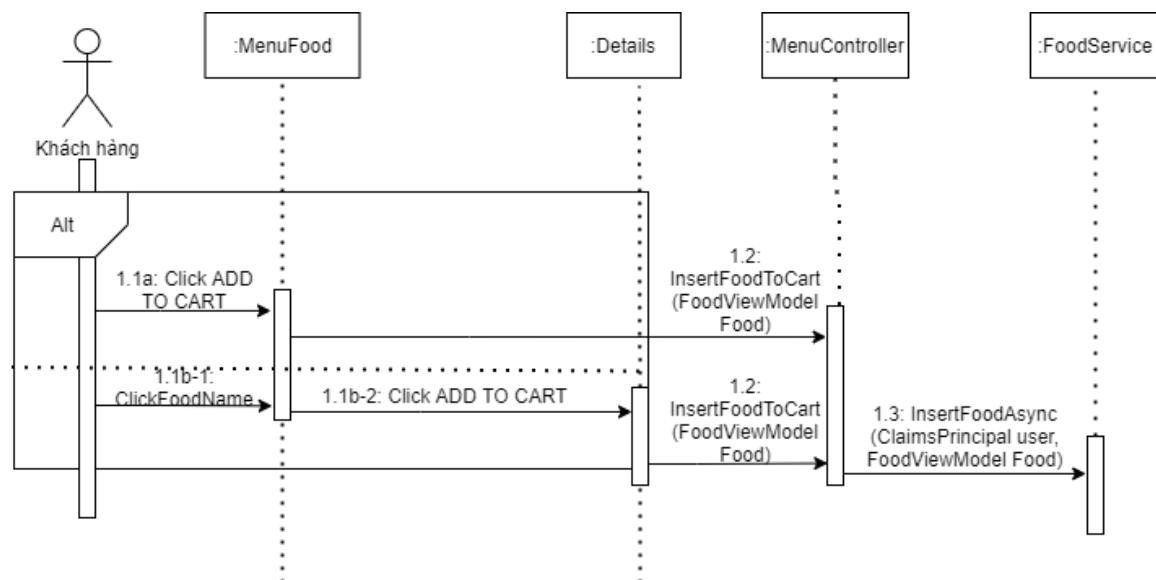


Hình 13: Sequence diagram của use case: Xem menu



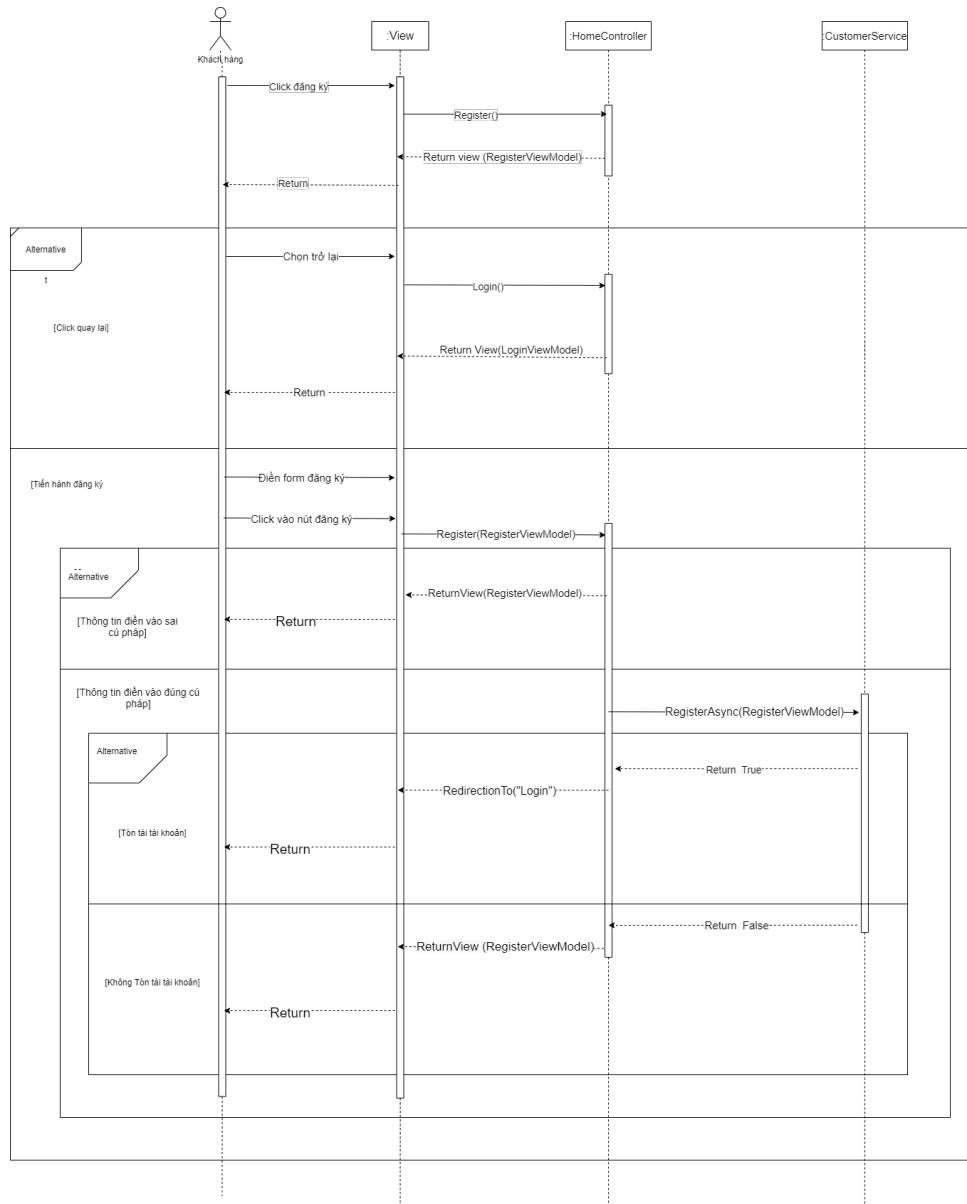
Hình 14: Sequence diagram của use case: Xem chi tiết món ăn

2.2.5 Thêm món ăn vào giỏ hàng



Hình 15: Sequence diagram của use case: Thêm món ăn vào giỏ hàng

2.2.6 Đăng ký

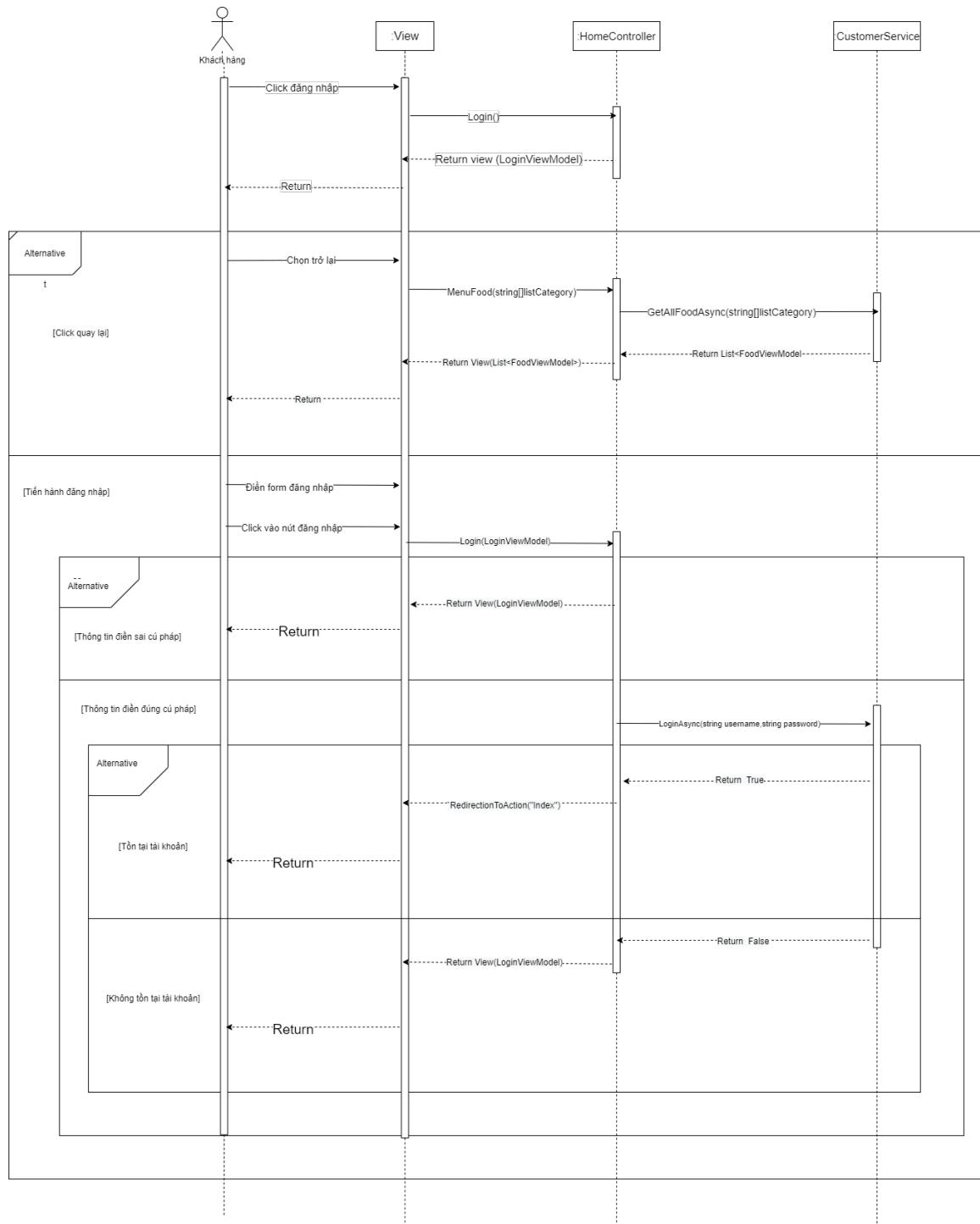


Hình 16: Sequence diagram của use case: Đăng ký



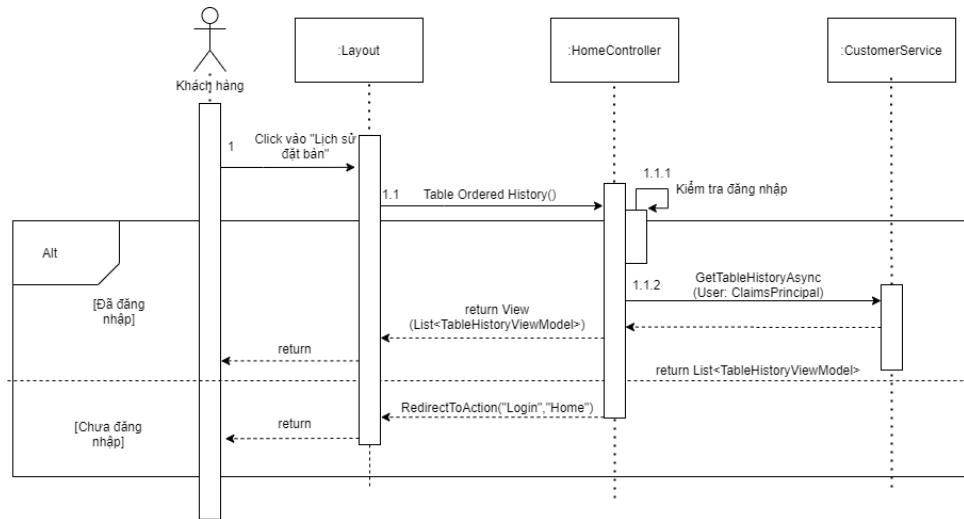
Trường Đại học Bách Khoa, TP. Hồ Chí Minh
Khoa Khoa Học & Kỹ Thuật Máy Tính

2.2.7 Đăng nhập



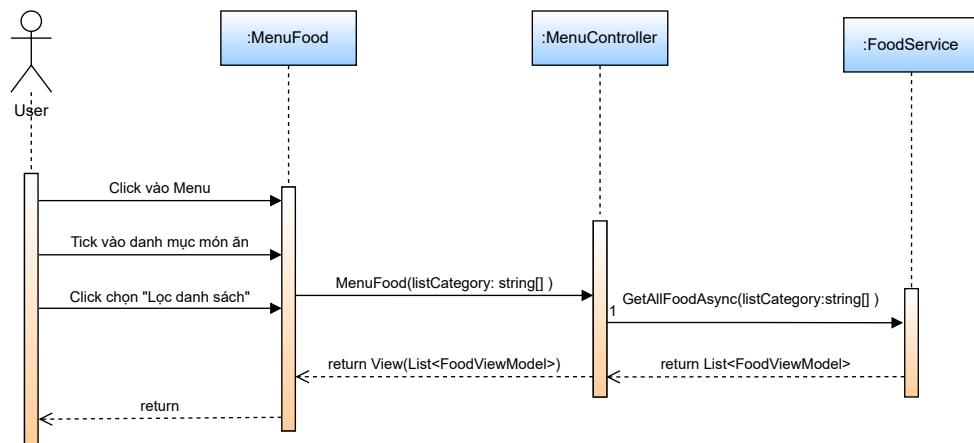
Hình 17: Sequence diagram của use case: Đăng nhập

2.2.8 Lịch sử đặt bàn



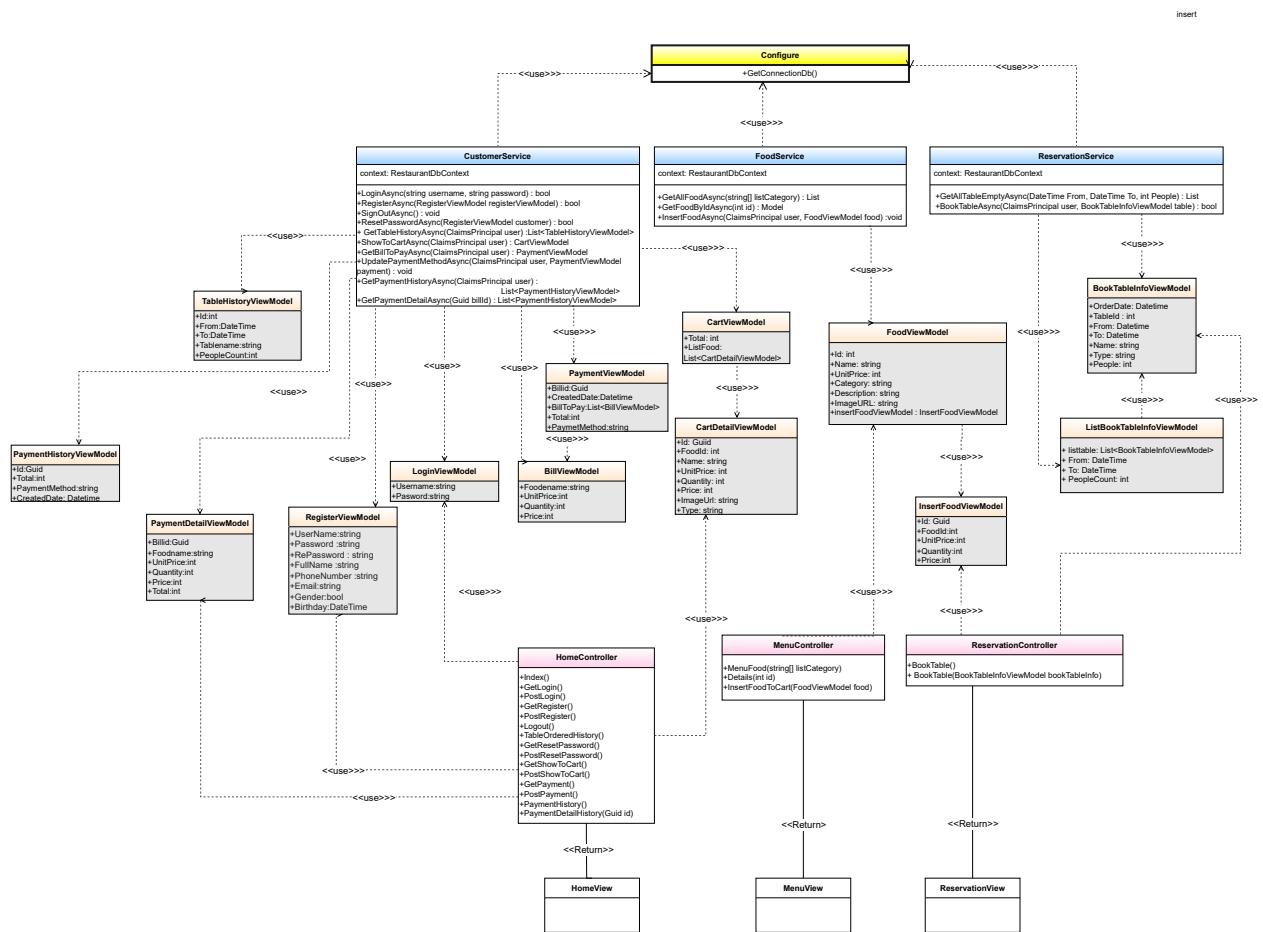
Hình 18: Sequence diagram của use case: Lịch sử đặt bàn

2.2.9 Lọc danh sách món ăn theo danh mục



Hình 19: Sequence diagram của use case: Lọc danh sách món ăn theo danh mục

2.3 Class Diagram



Hình 20: Class diagram cho toàn hệ thống



3 Architecture design

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

3.1.1 Database Management: SQL Server

- Phù hợp với hệ điều hành Window (được sử dụng bởi tất cả các thành viên trong nhóm và đa số người dùng trên thế giới)
- Được tích hợp với .NET Core (Framework được nhóm chọn làm Back End)
- Dễ lập trình, sử dụng, có các server online miễn phí hoặc giá phải chăng để cho phép các thành viên kết nối giúp dễ dàng thiết kế và quản lý cơ sở quản trị dữ liệu

3.1.2 Back End: .NET Core Framework and MVC Design Pattern

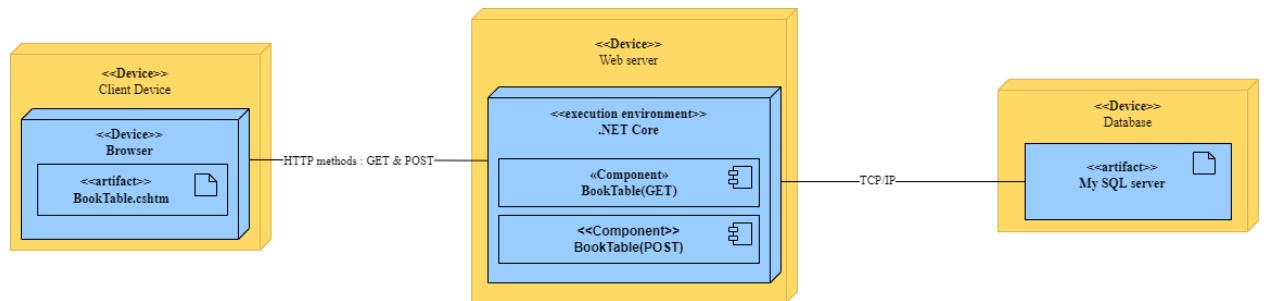
- Controller: tiếp nhận, xử lý và trả lại trả lời các yêu cầu của người dùng. Khi được yêu cầu truy vấn dữ liệu, controller sẽ liên hệ với model để lấy dữ liệu. Khi được yêu cầu trả về các giao diện, controller sẽ liên hệ với view để lấy về các giao diện của web
- Views: tạo giao diện của app. Với view, nhóm sẽ viết các thành phần HTML của app.
- Model: lưu trữ toàn bộ dữ liệu của ứng dụng. Model sẽ nhận yêu cầu truy vấn dữ liệu từ controller và trả lại dữ liệu phù hợp cho controller
- Services: giúp kết nối Model với Database. Services giúp thực hiện các thao tác trên cơ sở dữ liệu như thêm, xóa và cập nhật dữ liệu.
- Data: tạo ra các database để hệ thống kết nối với SQL Server

3.1.3 Front End

- Sử dụng HTML, CSS và JAVASCRIPT để tạo cấu trúc cơ bản, thiết kế hình ảnh và các hiệu ứng cho trang web
- Sử dụng thư viện BOOTSTRAP, GOOGLE FONT, FONT AWESOME và JQUERY để có sẵn các form trang web nổi tiếng để thân thiện hơn với người dùng hiện nay

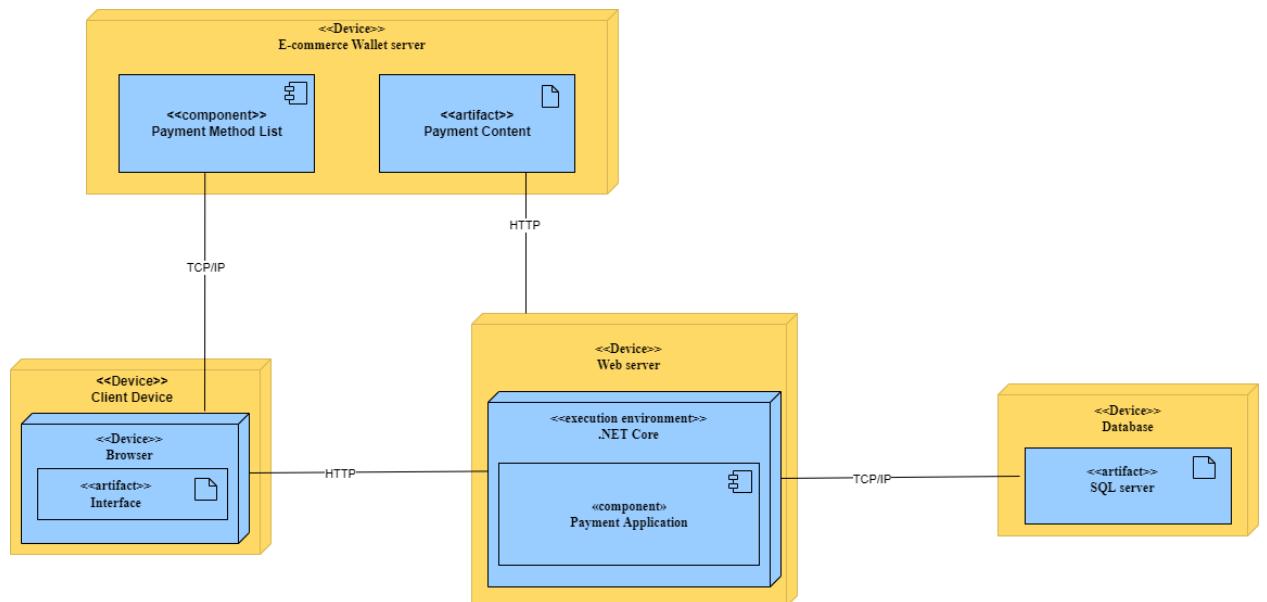
3.2 Draw an implementation diagram for Major (not all) functional requirements.

3.2.1 Đặt bàn



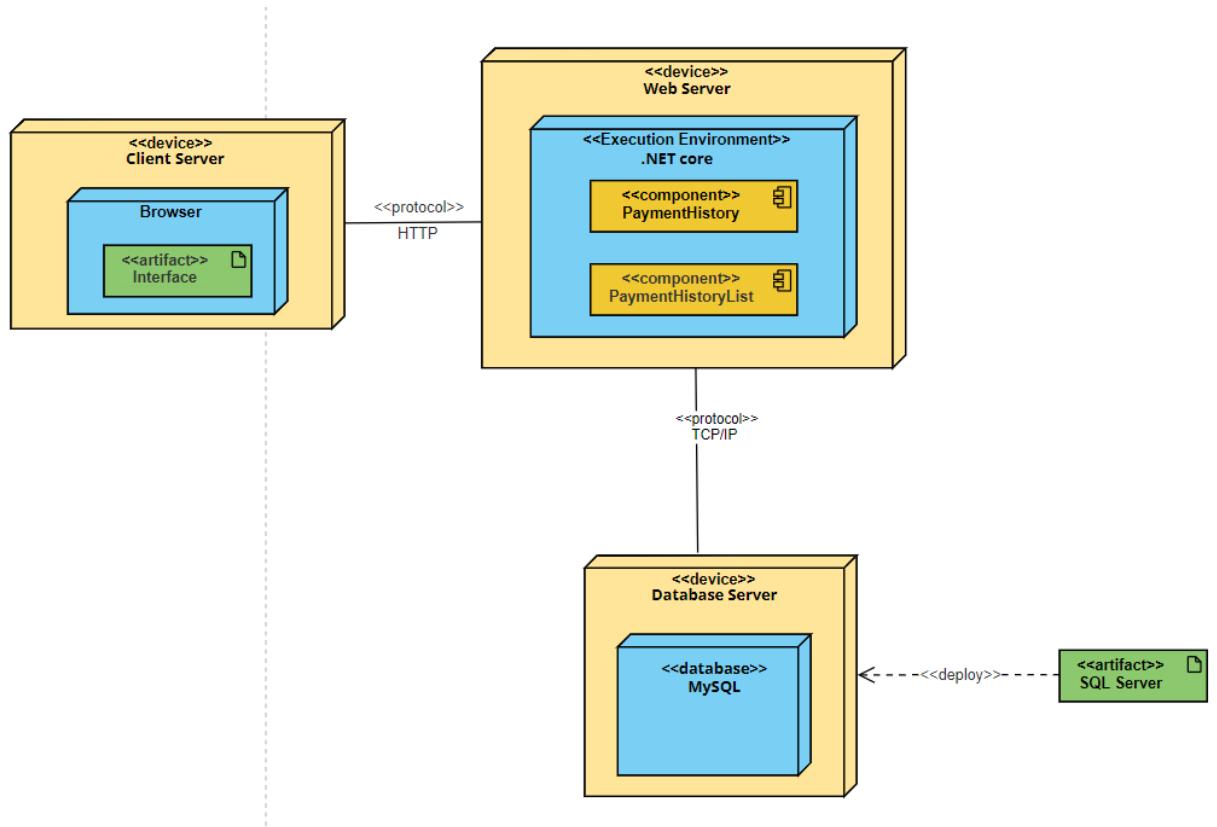
Hình 21: Implementation diagram for use case: Reservation

3.2.2 Thanh toán



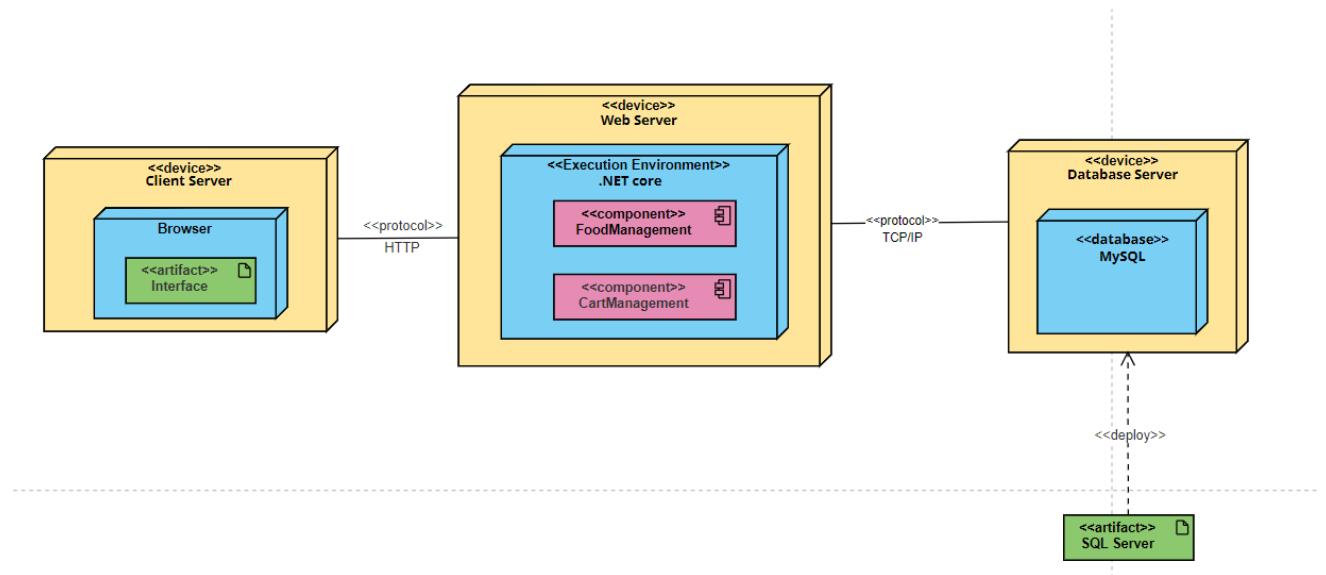
Hình 22: Implementation diagram for use case: Payment

3.2.3 Lịch sử thanh toán



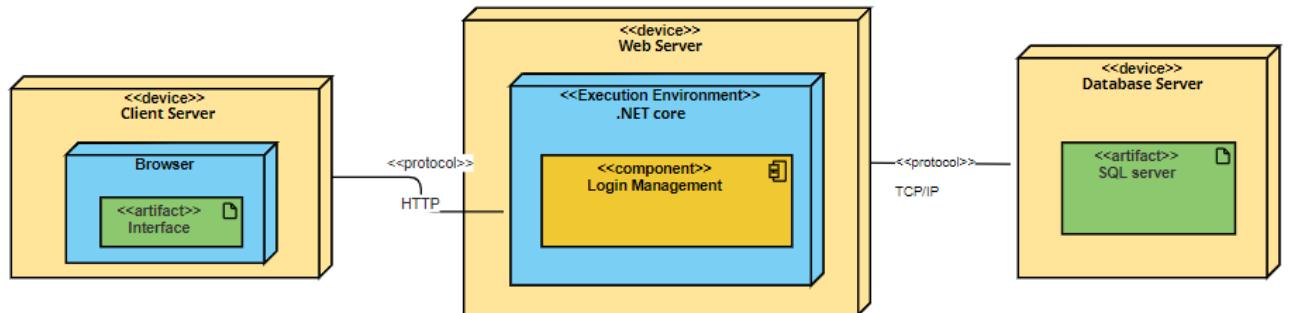
Hình 23: Implementation diagram for use case: PaymentHistory

3.2.4 Thêm món ăn vào giỏ hàng



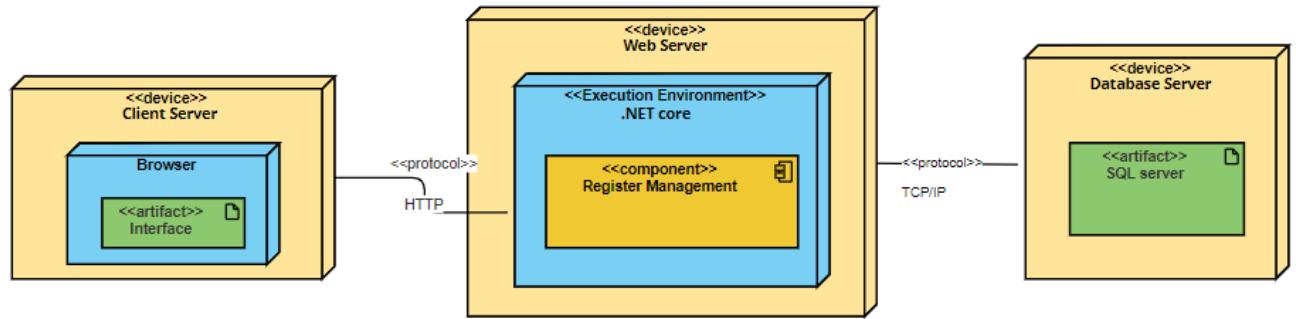
Hình 24: Implementation diagram for use case: Add foods to cart

3.2.5 Đăng nhập



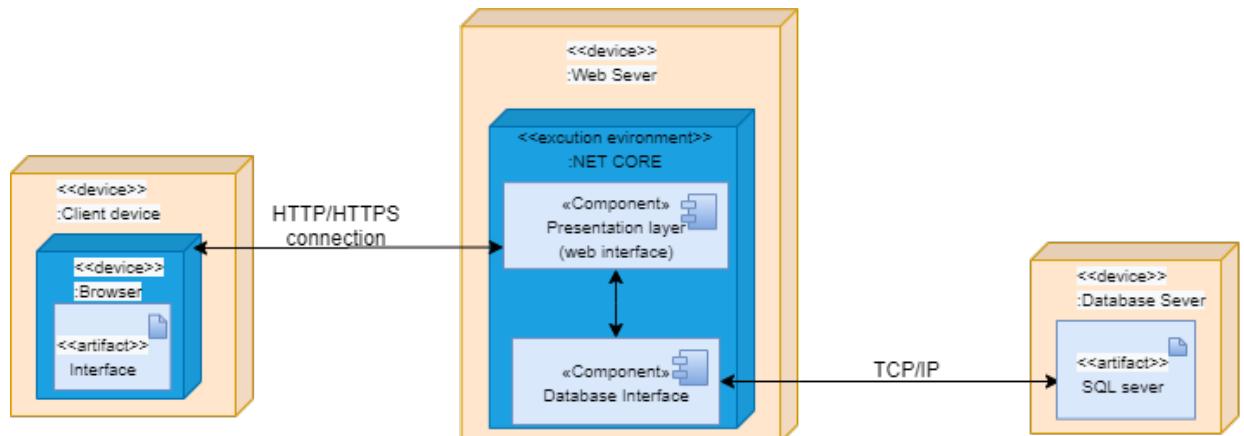
Hình 25: Implementation diagram for use case: Login

3.2.6 Đăng ký



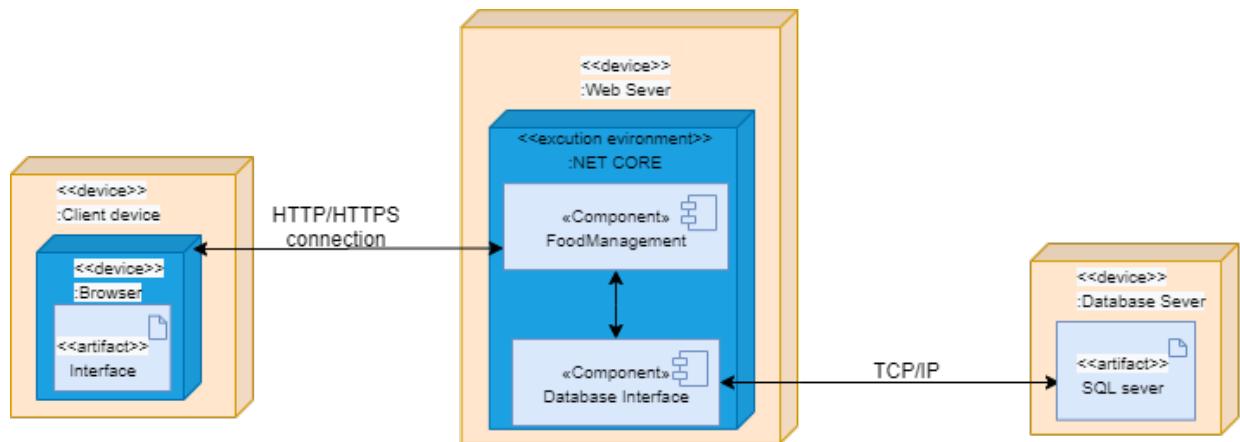
Hình 26: Implementation diagram for use case: Register

3.2.7 Lịch sử đặt bàn



Hình 27: Implementation diagram for use case: Lịch sử đặt bàn

3.2.8 Lọc danh sách món theo danh mục



Hình 28: Implementation diagram for use case: Lọc danh sách món theo danh mục

4 Implementation - Sprint 1

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/Thaigitub/SoftwareEngineering.git>



4.2 Giao diện đã hoàn thành

4.2.1 Header



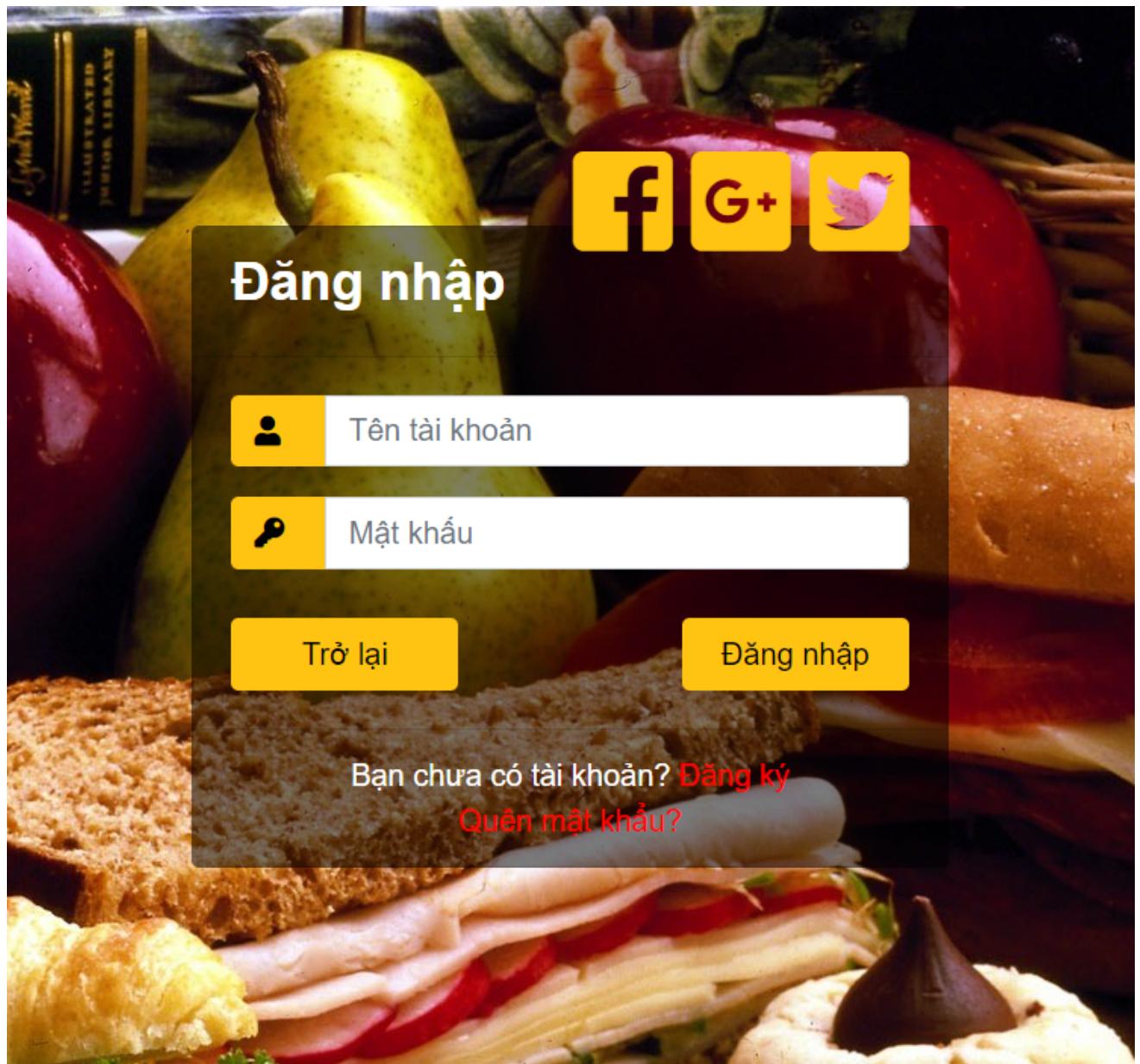
Hình 29: Giao diện Header

4.2.2 Footer



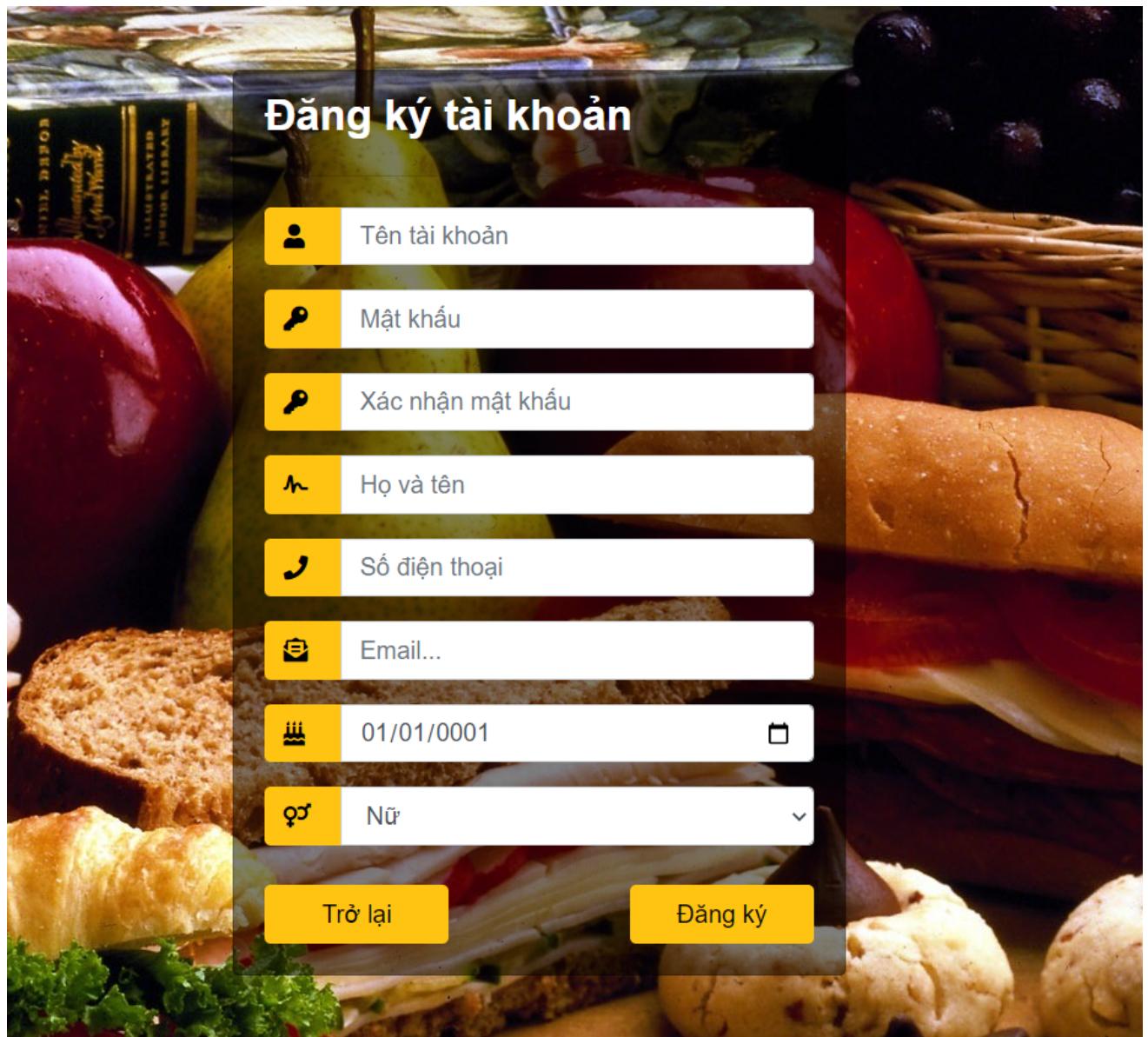
Hình 30: Giao diện Footer

4.2.3 Đăng nhập



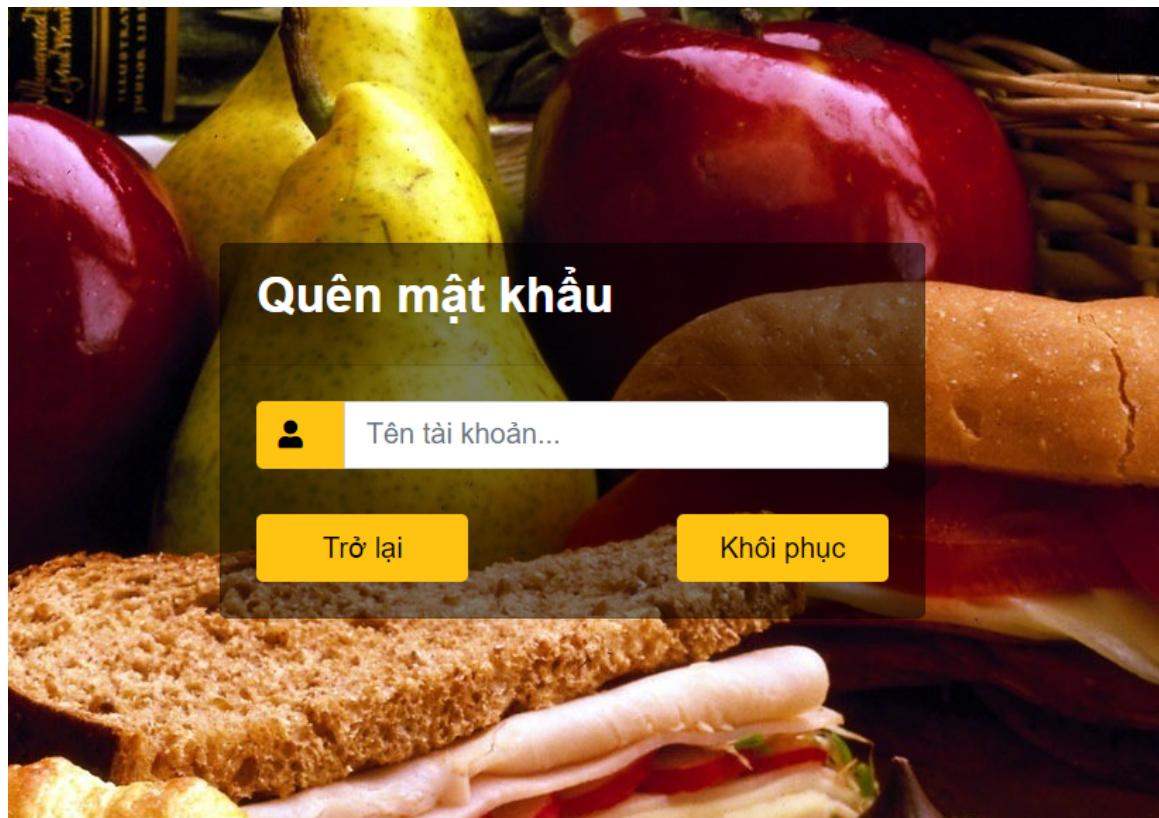
Hình 31: Giao diện Đăng nhập

4.2.4 Đăng ký

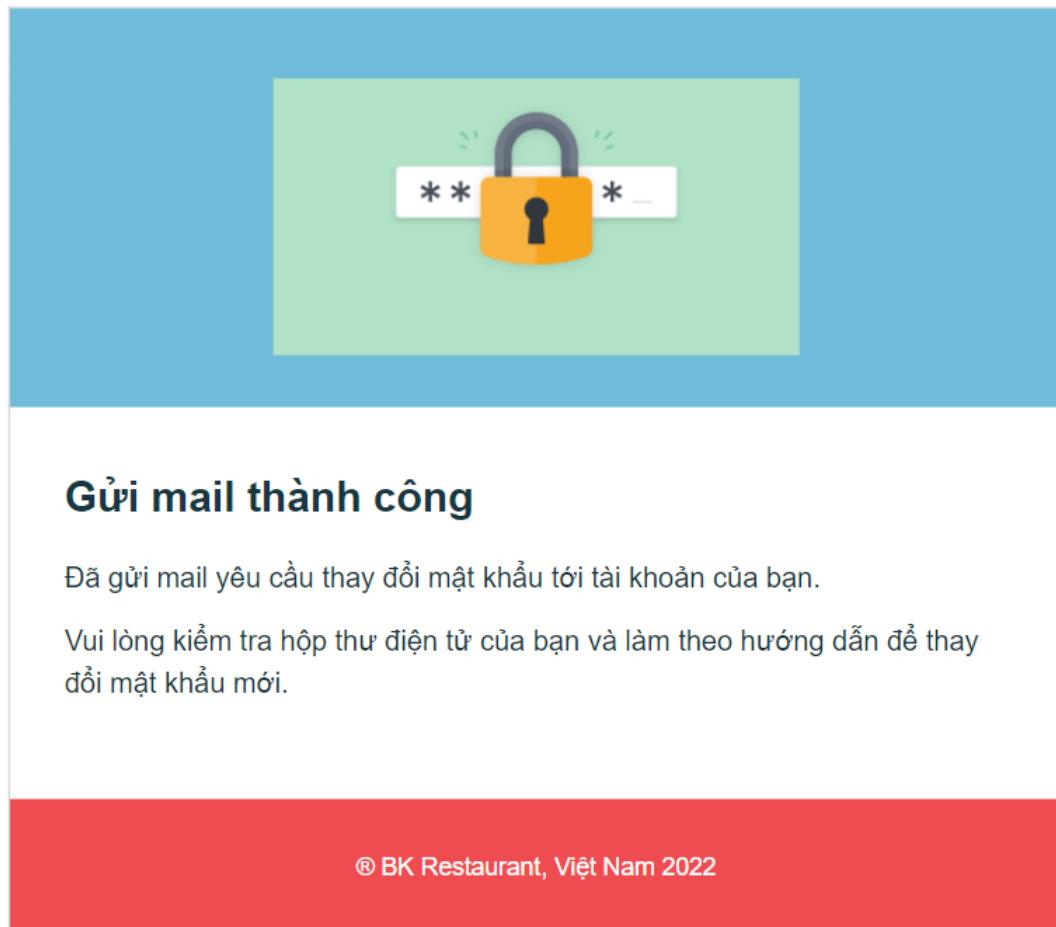


Hình 32: Giao diện Đăng ký

4.2.5 Tạo mật khẩu mới



Hình 33: Giao diện Lấy lại mật khẩu



Hình 34: Giao diện sau khi bấm vào Khôi phục



Đổi mật khẩu tài khoản RESTAURANT

Xin chào Bùi Thanh Duy,

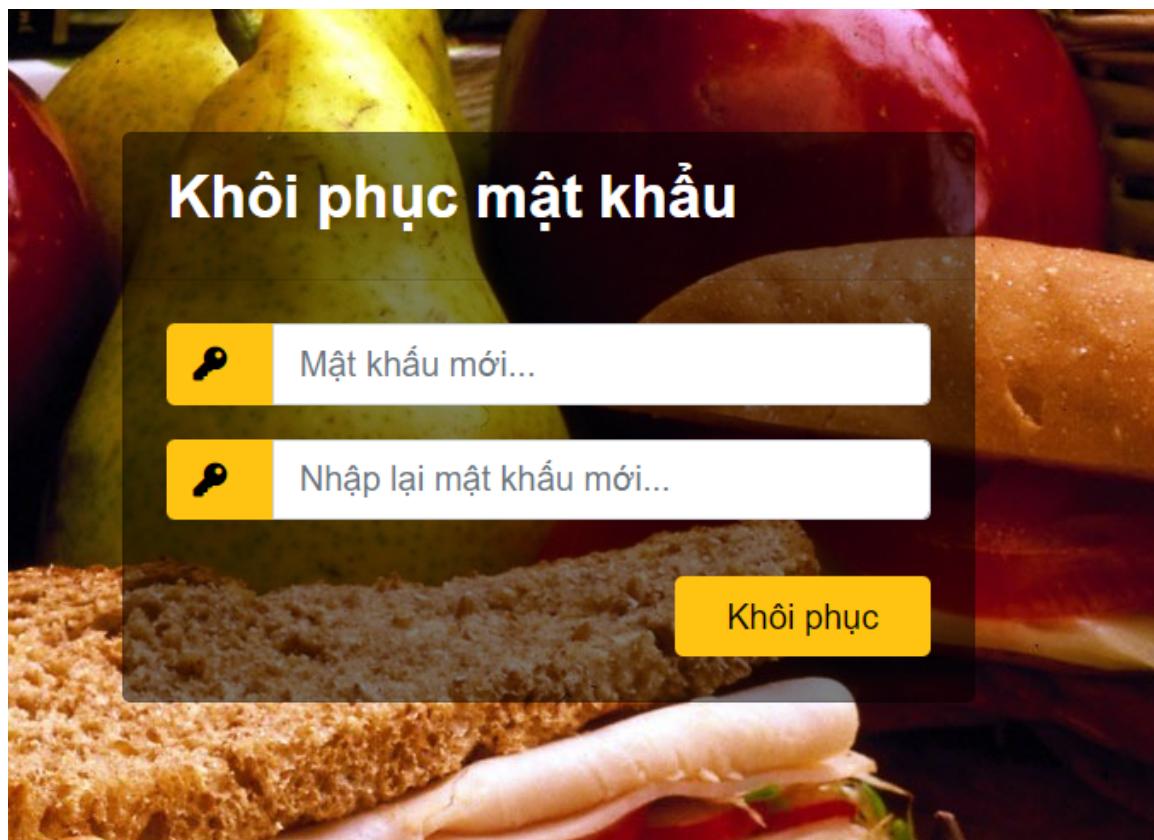
Chúng tôi vừa nhận được yêu cầu đổi mật khẩu tài khoản của bạn từ website **RESTAURANT**.

Nếu bạn mong muốn thay đổi mật khẩu, vui lòng nhấn vào nút **ĐỔI MẬT KHẨU** bên dưới và làm theo hướng dẫn để có thể thay đổi mật khẩu cho tài khoản của bạn.

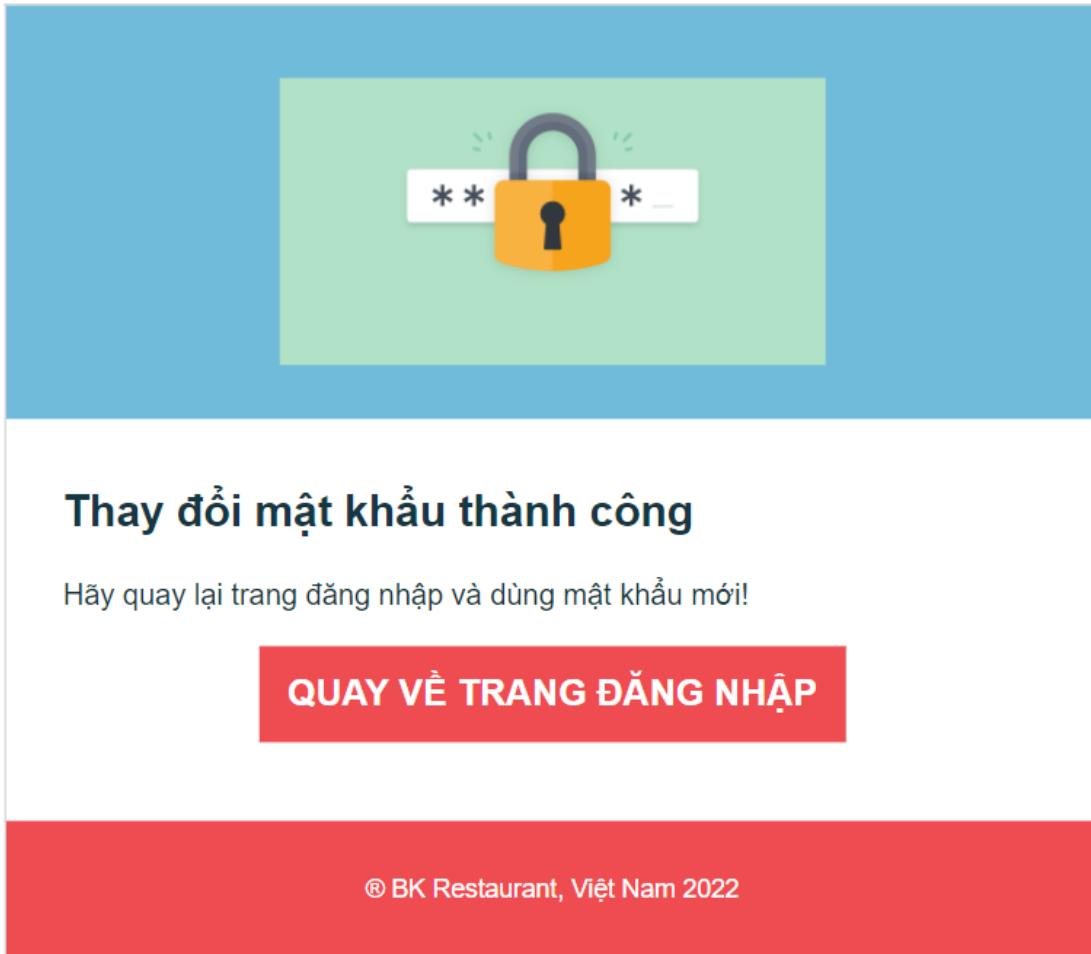
ĐỔI MẬT KHẨU

® Restaurant, Việt Nam 2022

Hình 35: Form nhận được trong gmail



Hình 36: Giao diện sau khi bấm vào ĐỔI MẬT KHẨU



Hình 37: Giao diện sau khi đổi mật khẩu thành công



Trường Đại học Bách Khoa, TP. Hồ Chí Minh
Khoa Khoa Học & Kỹ Thuật Máy Tính

4.2.6 Trang chủ



Welcome to BK Restaurant
Khách hàng là thương đế



Restaurant
Food & Drinks

Hình 38: Giao diện trang chủ



4.2.7 Menu

Danh Mục Món

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

Lọc Danh Sách

Salad:



SALAD DƯA CHUỐT
30000 VND

★★★★★



SALAD CÁ TRÍCH
50000 VND

★★★★★



CAESAR SALAD
40000 VND

★★★★★

Món Hấp Xào:



HÀ CẢO THẬP CẨM
35000 VND

★★★★★



NGHÊU XÀO BƠ TỎI
40000 VND

★★★★★



CUA HOÀNG ĐẾ HẤP SÀ
1099000 VND

★★★★★

Hình 39: Giao diện Menu



4.2.8 Chi tiết món ăn

HÀ CẢO THẬP CẨM

★★★★★
10 Review(s)

Giá: 35000 VND

Cách chế biến tinh tế mang đến hương vị tuyệt hảo

SỐ LƯỢNG

ADD TO CART

Hình 40: Giao diện Chi tiết món ăn

4.2.9 Đặt bàn

Tìm kiếm bàn trống:

Đặt trước

Bàn tại nhà hàng

Ngày: mm/dd/yyyy

Từ: -:-

Số lượng khách: Chọn số lượng khách

Đến: -:-

TÌM KIẾM

Thời gian mở cửa

Thứ	Thời gian
Thứ 2	8:00 am – 11:30 pm
Thứ 3	8:00 am – 11:30 pm
Thứ 4	8:00 am – 11:30 pm
Thứ 5	8:00 am – 11:30 pm
Thứ 6	8:00 am – 11:30 pm
Thứ bảy	8:00 am – 11:30 pm
Chủ nhật	8:00 am – 11:30 pm

Chọn bàn muốn đặt

STT	Tên bàn	Số lượng khách	Từ	Đến	Trạng thái	Đặt bàn
Quý khách chọn thời gian đặt bàn bên trên!						

Hình 41: Giao diện tìm kiếm bàn trống

Đặt bàn:



The screenshot shows a restaurant booking interface. On the left, there's a search form titled "Đặt trước" (Reserve) with fields for "Ngày:" (Date), "Từ:" (From), "Số lượng khách:" (Number of guests), and "Đến:" (To). Below these is a dropdown for "Chọn số lượng khách" (Select number of guests) and an orange "TÌM KIẾM" (Search) button. To the right, a sidebar titled "Thời gian mở cửa" (Opening hours) lists the days of the week from Tuesday to Sunday, each with a time range from 8:00 am to 11:30 pm. At the bottom, there's a section titled "Chọn bàn muốn đặt" (Select the table you want to book) with a table showing two available tables: Table 4 and Table 7, both for 4 people, starting at 16:00 on December 23, 2022.

STT	Tên bàn	Số lượng khách	Từ	Đến	Trạng thái	Đặt bàn
1	Bàn 4	4 người	23/12/2022 16:00	23/12/2022 16:30	Trống	ĐẶT BÀN
2	Bàn 7	4 người	23/12/2022 16:00	23/12/2022 16:30	Trống	ĐẶT BÀN

Hình 42: Giao diện Đặt bàn

4.2.10 Lịch sử đặt bàn

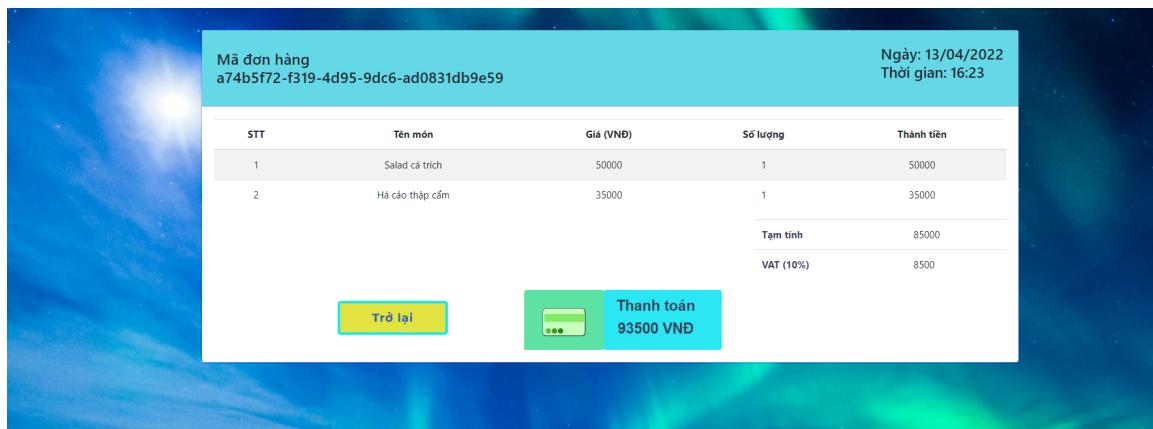
Lịch sử đặt bàn

STT	Tên bàn	Số người	Bắt đầu	Kết thúc
1	Bàn 4	4	23/12/2022 16:00	23/12/2022 16:30

Hình 43: Giao diện Lịch sử Đặt bàn

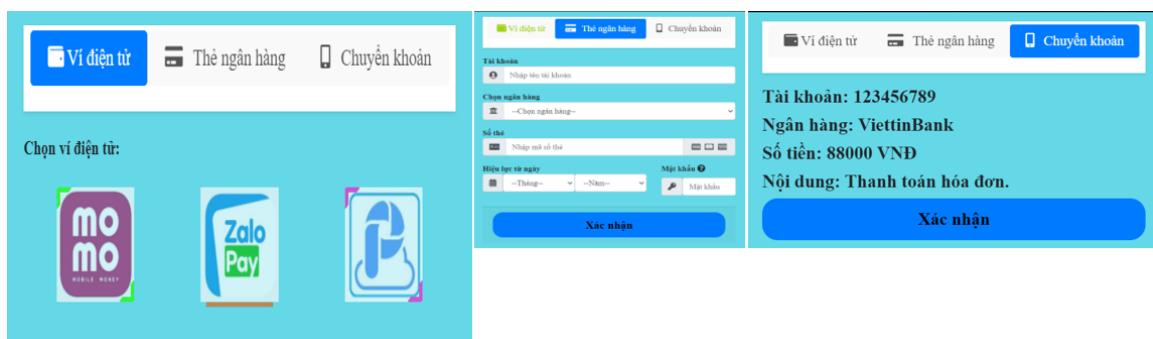


4.2.11 Thanh toán



Hình 44: Giao diện Thanh toán

4.2.12 Các phương thức thanh toán



Hình 45: Giao diện 3 phương thức thanh toán

4.2.13 Lịch sử thanh toán

Lịch sử thanh toán

STT	Chi tiết	Tổng (VND)	Phương thức thanh toán	Ngày thanh toán
1	Xem chi tiết:	1285900	Ví điện tử Momo	13/04/2022 21:28
2	Xem chi tiết:	88000	Ví điện tử Momo	13/04/2022 21:26

Hình 46: Giao diện lịch sử thanh toán



4.2.14 Giỏ hàng

Giỏ hàng

STT	Tên món ăn	Hình ảnh	Giá (VND)	Số Lượng	Tổng(VND)
1	Salad cá trích		50000	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	50000
2	Caesar salad		40000	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	40000
3	Cua hoàng đế hấp sả		1099000	<input type="button" value="-"/> <input type="text" value="1"/> <input type="button" value="+"/>	1099000

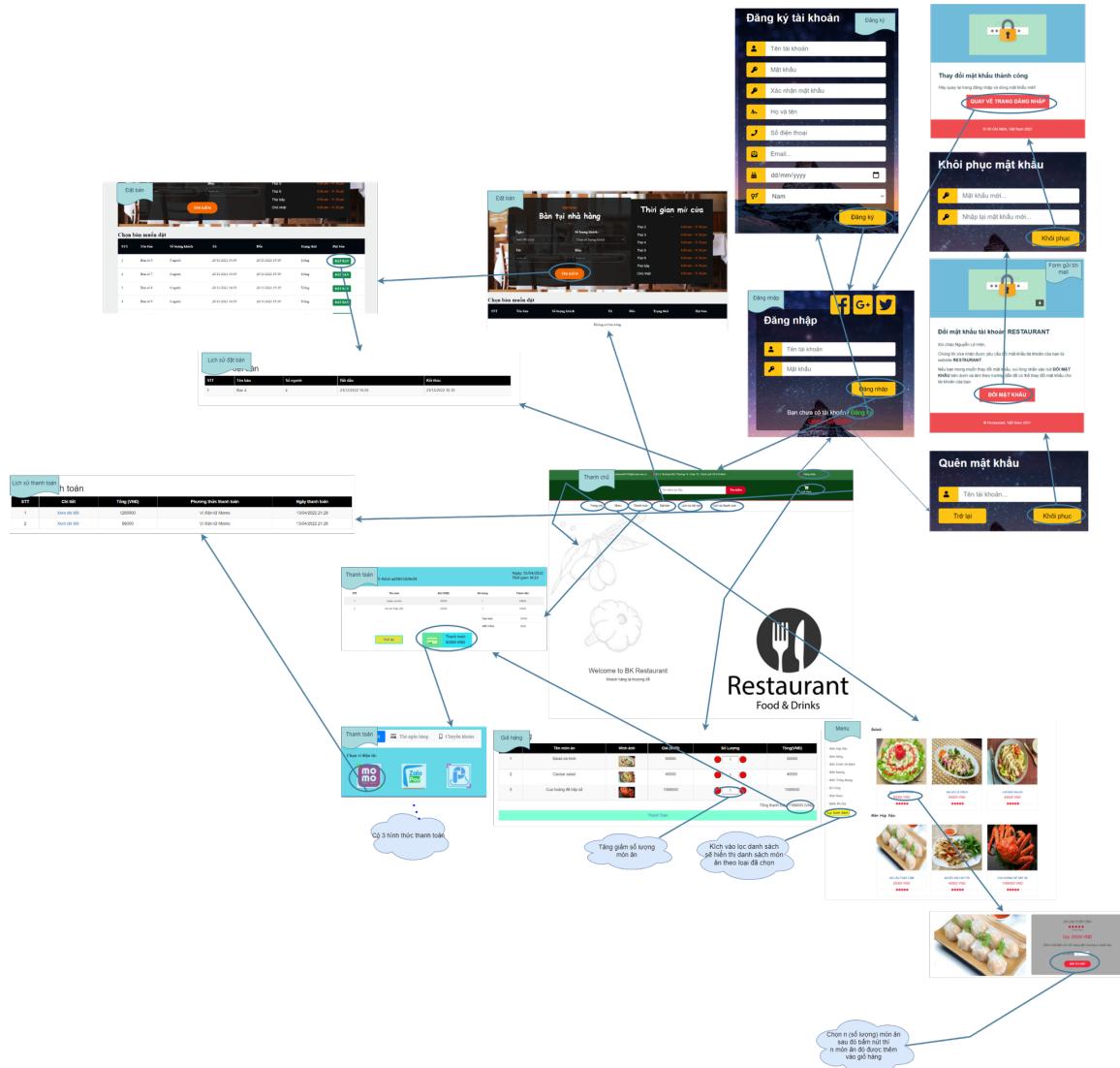
Tổng thanh toán: 1189000 (VND)

Thanh Toán

Hình 47: Giao diện giỏ hàng

4.2.15 Flow of screen

Link Flow of Screen : <https://drive.google.com/file/d/12YJ3GHkMb4rZKqSpEiC24qbqO7pYK/view?usp=sharing>



Hình 48: ScreenFlow

4.3 Mô tả các tính năng

4.3.1 Layout và Trang chủ

- **Header** như hình 4.1. Tại đây có thẻ bấm vào **Đăng nhập** để đăng nhập (nếu chưa đăng nhập) hoặc bấm **Đăng xuất** (nếu đã từng 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.

- **Footer** như hình 4.2



- **Trang chủ** như hình 4.10. Trang chủ có slogan của nhà hagnf và một ảnh động thêm đẹp mắt.

4.3.2 Đăng nhập, đăng ký, quên mật khẩu

- Từ góc trên bên phải của màn hình trang chủ, bấm chọn **Dăng nhập**. **Dăng nhập** như hình 4.2.3: Nhập Tên tài khoản và Mật khẩu, sau khi điền đầy đủ bấm **Dăng 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 4.2.4: 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 4.2.9.

4.3.3 Hiển thị Menu, xem thông tin món ăn, thêm món ăn vào giỏ hàng

- **Menu** như trong Hình 4.2.11: 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.s
- **Chi tiết món ăn** như hình 4.2.12: 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, xem lịch sử đặt bàn

Đặt bàn như hình 4.2.13: 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 4.2.15.

4.3.5 Xem giỏ hàng, chỉnh sửa đơn hàng

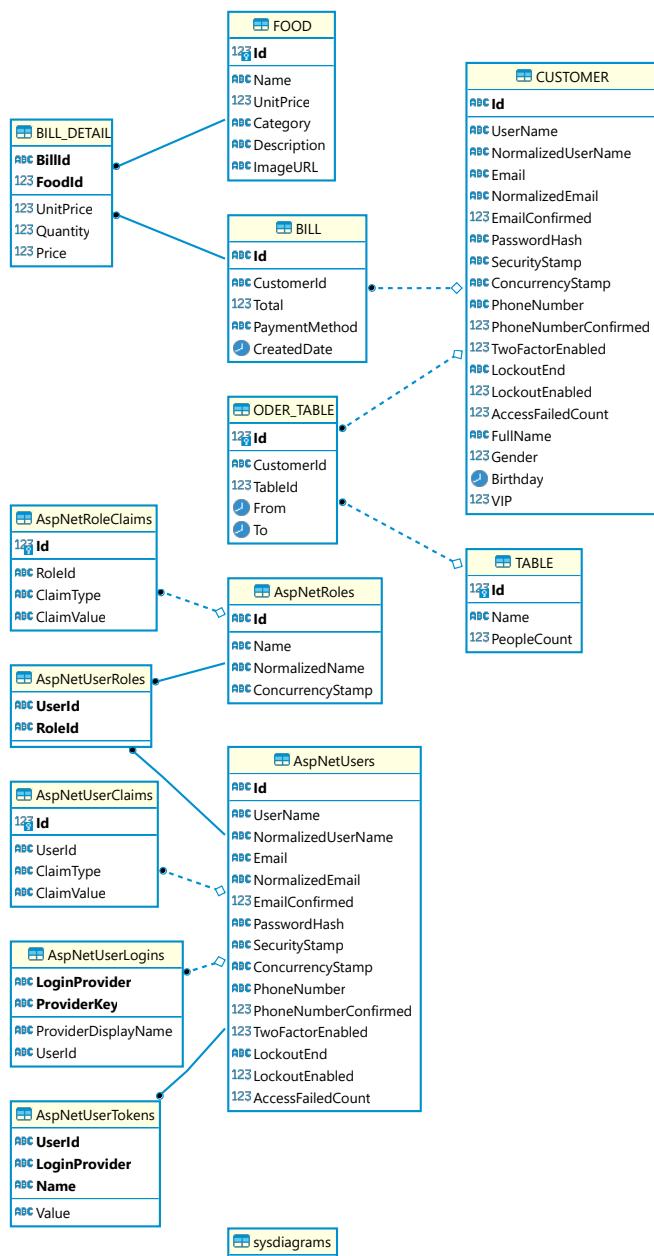
Xem giỏ hàng như hình 4.2.19: Tại đây có thể bấm vào 2 nút (+) hoặc (-) để thêm hoặc bớt số lượng món ăn. Sau đó bấm **Thanh toán** để tiến hành thanh toán.

4.3.6 Thanh toán, lịch sử thanh toán

Thanh toán như hình 4.2.16: 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 4.2.18..

5 Implementation - Sprint 2

5.1 Database diagram



Hình 49: Database diagram



- Nhóm chúng em thực hiện đăng kí host trên trang somee.com để triển khai database cũng như website của nhóm hoạt động thông qua các bước sau:
 - Bước 1: Nhóm đã code các Entity, EntityConfiguration, DbContext rồi thực hiện kết nối database.
 - Bước 2: Sau khi đăng kí host thành công, **somee** cung cấp sẵn connection-string để cấu hình trong Entity Framework Core. Dùng chuỗi này để cấu hình cho DefaultConnection trong file *appsetting.json* của dự án.
 - Bước 3: Vì khi khởi tạo project ASP.NET Core MVC tại Visual Studio, file *Startup.cs* đã inject DbContext bằng cách sử dụng dependency injection rồi nên nhóm chỉ thêm identity để hỗ trợ login, logout sau này.
 - Bước 4: Tiến hành add-migration và update database để tạo ra database mới.
 - Bước 5: Terminal hiển thị việc kết nối và updata database thành công. Từ giờ các thao tác trên database được thực hiện thông qua DbContext.

5.2 Triển khai công nghệ CI/CD

CI/CD là một bộ đôi công việc, bao gồm **CI** (Continuous Integration) và **CD** (Continuous Delivery), ý nói là quá trình tích hợp (integration) thường xuyên, nhanh chóng hơn khi code cũng như thường xuyên cập nhật phiên bản mới (delivery).

CI/CD thu hẹp khoảng cách giữa các hoạt động và nhóm phát triển và vận hành bằng cách thực thi tự động hóa trong việc xây dựng, thử nghiệm và triển khai các ứng dụng.

- CI: Mỗi khi có *pull request*, CI giúp kiểm tra build thành công mới cho phép tích hợp.
- CD: Khi có hành động *merge* vào main, CD giúp tự động deploy lên heroku (địa chỉ: <https://bkrestaurantpos.herokuapp.com/>)



```
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 50: File ci.yml được dùng để check build pass before merge trong github Actions

```
26 lines (22 sloc) | 725 Bytes
Raw Blame ⌂ ⌄ ⌅ ⌆
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 51: File deploy.yml được dùng để deploy lên heroku trong github Actions



U	Deploy to Heroku #7: Commit 46ca79f pushed by Thaigithub	main	14 days ago 1m 17s	...
U	Deploy to Heroku #6: Commit a3bf397 pushed by Thaigithub	main	14 days ago 1m 10s	...
Up	Deploy to Heroku #5: Commit 679535e pushed by Thaigithub	main	14 days ago 1m 19s	...
Update	Deploy to Heroku #4: Commit b8f7974 pushed by Thaigithub	main	14 days ago 25s	...

Hình 52: Kiểm tra hoạt động *deploy to heroku*



Trường Đại học Bách Khoa, TP. Hồ Chí Minh
Khoa Khoa Học & Kỹ Thuật Máy Tính
