

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

---oOo---



CÔNG NGHỆ PHẦN MỀM (CO3001)

BÁO CÁO BÀI TẬP LỚN TASK 1,2,3,4

Lớp L01 - nhóm 9

GVHD:

Bùi Công Tuấn

SVTH:

Lê Minh Nhật - 1914477

Nguyễn Hoàng Nhiên - 1911796

Hồ Công Phúc - 2120057

Trần Hoàng Quân - 1813715

Nguyễn Phúc Thanh - 1915094

TP. Hồ Chí Minh, 11/2021

Task 1

Task 1.1

a. Ngữ cảnh của dự án

Trong thời kỳ kinh tế và xã hội khủng hoảng vì đại dịch COVID-19, nền kinh tế và mọi hoạt động kinh doanh bị ảnh hưởng, bao gồm cả việc kinh doanh tại các nhà hàng ăn uống. Với mục tiêu đảm bảo thực hiện các biện pháp an toàn chống dịch cho khách hàng khi đến tại cửa hàng, hệ thống Point of sale (POS) được xây dựng và phát triển hơn nữa để đối phó với tình hình dịch bệnh.

POS là thời gian và địa điểm hoàn thành giao dịch bán lẻ, nó cho biết số tiền trên hóa đơn đến khách hàng có thể lập hóa đơn cho khách hàng và chỉ ra các tùy chọn để khách hàng thanh toán.

User Story:

- Là khách hàng, tôi muốn nhanh chóng xem tất cả món ăn của cửa hàng
- Là khách hàng, tôi muốn có thể tìm kiếm các món ăn theo tên hoặc lọc ra theo loại
- Là khách hàng, tôi muốn thêm bình luận về món ăn
- Là khách hàng, tôi muốn nhanh chóng thanh toán
- Là nhân viên bán hàng, tôi muốn xem danh sách đơn hàng cần giao
- Là nhân viên, tôi muốn tạo đơn hàng khi khách order trực tiếp
- Là đầu bếp, tôi muốn xem danh sách món ăn cần được chế biến
- Là đầu bếp, tôi muốn xem đơn hàng đang được chế biến
- Là đầu bếp, tôi muốn khóa món ăn khi hết nhiên liệu
- Là quản lý tôi muốn thêm, sửa xóa thông tin về món ăn.
- Là quản lý tôi muốn thêm, sửa xóa các chương trình khuyến mãi của cửa hàng.
- Là quản lý tôi muốn thêm, xóa, sửa thông tin tài khoản của đầu bếp và nhân viên bán hàng.
- Là quản lý tôi muốn thêm, sửa, xóa các mã QR ứng với ban ăn cho cửa hàng.
- Là quản lý tôi muốn nhanh chóng xem thống kê thu nhập của cửa hàng.

Nhóm sẽ tập trung phần thiết kế cho mô hình căn tin chế biến thức ăn tại chỗ:

- Thực phẩm được chế biến tại chỗ
- Khách hàng có thể ăn tại quán hoặc mang đi
- Khách hàng phải thanh toán trước để căn tin tiến hành chế biến
- Hỗ trợ hình thức thanh toán trực tiếp hoặc online thông qua internet banking hoặc dịch vụ thanh toán thứ ba như MOMO

b. Các bên liên quan

Khách hàng: bị ảnh hưởng trực tiếp bởi phần mềm, các thông tin về tài khoản, các giao dịch của họ được hệ thống ghi lại.

Nhân viên bán hàng: quản lí đơn hàng từ khách hàng.

Đầu bếp: chịu trách nhiệm duyệt các đơn hàng mới và đang được thực hiện để hoàn thành công việc.

Quản lí nhà hàng: bao gồm quản lí thông tin từ hệ thống.

Chủ nhà hàng (chủ đầu tư): chịu trách nhiệm về dòng tiền, đóng vai trò quyết định tiếp tục hay hủy bỏ dự án.

Bên chấp nhận thanh toán: thanh toán tiền và thông báo tình trạng thanh toán cho khách hàng.

Nhóm phát triển phần mềm: xây dựng phần mềm dựa trên những phản hồi yêu cầu từ các bên liên quan khác, các nhóm phát triển phải được liên kết với nhau để tạo ra được một sản phẩm thống nhất.

Nhóm trưởng quản lí dự án: định hình quá trình phát triển, theo dõi và kết nối tất cả các bộ phận phát triển phần mềm nhằm tối ưu hóa và đưa ra hệ thống khiến khách hàng hài lòng.

Kỹ sư phát triển phần mềm: chịu trách nhiệm về cài đặt và bảo trì hệ thống.

c. Những mong đợi từ hệ thống

Hệ thống được mong đợi giúp làm tăng trí tuệ trong kinh doanh thời kì đại dịch, giảm sự lãng phí thời gian và tăng cơ hội mở rộng quy mô kinh doanh.

Cho phép không có sự tiếp xúc trực tiếp từ khách hàng và nhân viên phục vụ.

Các tính năng của hệ thống:

- Thanh toán online
- Tiến hành được giao dịch ngay trên web
- Sử dụng trên nhiều thiết bị khác nhau
- Thực hiện nhiều giao dịch trên hệ thống cùng 1 thời điểm

d. Phạm vi của dự án

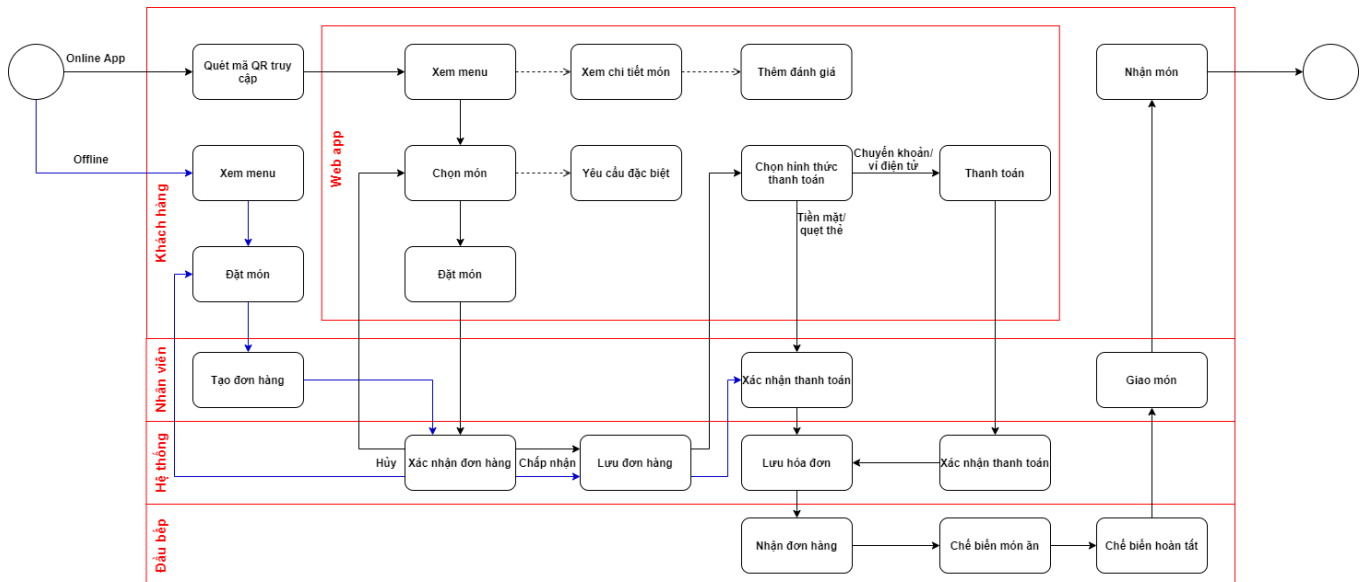
Chỉ cho phép sử dụng master card/VISA để thanh toán online

Nên được phát triển trên web và sử dụng QR code, không cần cài đặt app trên điện thoại.

Hệ thống phải có khả năng sử dụng trên điện thoại di động, máy tính bảng, laptop và máy tính để bàn.

Cho phép thực hiện đến 300 giao dịch một ngày.

e. Workflow



Task 1.2. Yêu cầu chức năng và phi chức năng

a. Yêu cầu chức năng

- Khách hàng:

Đăng nhập: vào tài khoản thành viên đã đăng kí

Đăng kí: đăng kí tài khoản thành viên mới

Quản lí tài khoản khách hàng: chỉnh sửa thông tin cá nhân, xem và xóa lịch sử đặt hàng, xóa tài khoản.

Tìm món: tìm kiếm món ăn theo tên, chủ đề tìm kiếm

Xem chi tiết món ăn: thông tin về món ăn, xem bình luận và đánh giá về món ăn

Đặt món; chọn món vào giỏ, chọn số lượng, xóa món, có thể tùy chọn mang đi

Thanh toán: thanh toán bằng tiền mặt, thẻ credit, qua ngân hàng.

- Nhân viên:

Đăng nhập: đăng nhập vào tài khoản nhân viên bán hàng

Tạo hóa đơn mới: khi khách hàng đặt trực tiếp mà không qua web

Xem danh sách chờ xác nhận thanh toán: xác nhận thanh toán

Xem lịch sử hóa đơn cũ: xem các hóa đơn bán hàng đã hoàn thành

Xem đơn hàng chờ giao: xác nhận đã giao hàng

- Đầu bếp:

Đăng nhập: đăng nhập tài khoản đầu bếp

Duyệt danh sách đơn hàng mới; chấp nhận hoặc hủy, nêu lí do

Xem danh sách món ăn (xem menu): thông báo trạng thái nấu ăn còn hàng hay đã hết hàng

Duyệt danh sách đơn hàng đang thực hiện: xác nhận hoàn tất đơn hàng

- Quản lí:

Đăng nhập: đăng nhập vào tài khoản quản lí

Quản lí tài khoản nhân viên: chỉnh sửa, thêm xóa tài khoản của nhân viên

Quản lí chương trình khuyến mãi: thêm chương trình hay hủy chương trình

Quản lí món ăn: sửa thông tin, thêm hay xóa món ăn

Quản lí mã QR: thêm mới, sửa hay xóa mã QR cũ

b. Yêu cầu phi chức năng

- Hiệu suất

Thời gian tải trang web sau khi quét mã QR dưới 2s

Hoạt động 24/7

Thời gian đặt món và đặt bàn dưới 5s

Thời gian phản hồi hủy món và hủy đặt bàn cho khách hàng dưới 5s

- Khả năng mở rộng

Có thể sử dụng cho nhiều cửa hàng khác nhau cùng lúc

- Sức chứa

Cho phép 100 khách hàng sử dụng cùng một thời điểm

Xử lý được ít nhất 300 đơn trên ngày

- Tính khả dụng

Sử dụng được trên hệ điều hành thiết bị di động là IOS, android hay hệ điều trên máy tính để bàn Window 7 trở lên

Tương thích với các kích cỡ màn hình khác nhau

- Độ tin cậy

Bảo mật thông tin người dùng

Ẩn mã số thẻ ngân hàng khi khách hàng đăng nhập

Mật khẩu khách hàng được mã hóa

Thông báo khi thông tin của tài khoản được sửa hay bị xóa

- Sử dụng-tương tác

Hoàn thành thao tác với 3 click chuột

Nhân viên có thể dùng được phần mềm sau 30p training

Khách hàng mới có thể thao tác sau 5 phút tìm hiểu

Duyệt xem menu ngay trang chủ

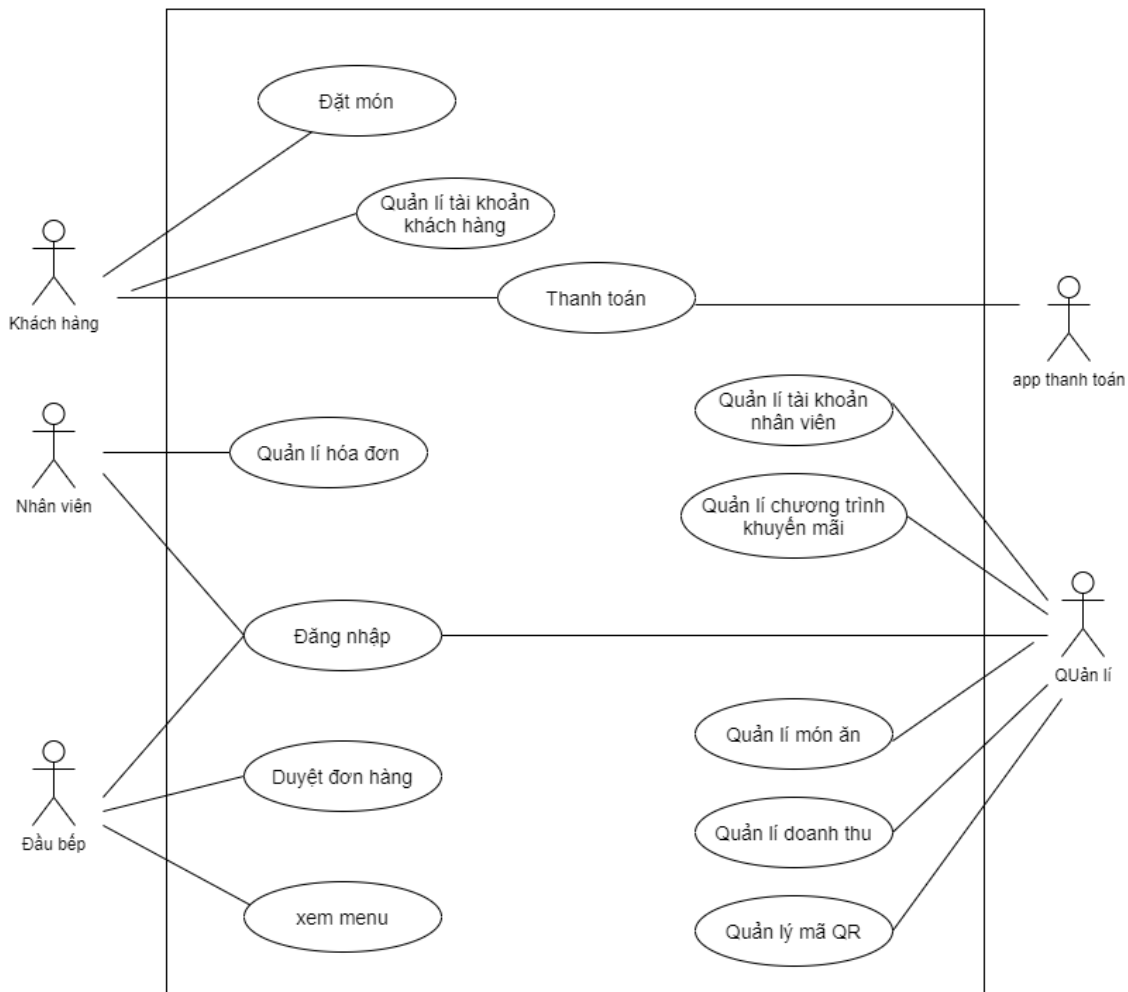
Thông tin khuyến mãi được hiển thị ngay khi khách hàng đăng nhập

Khuyến nghị tạo tài khoản cho khách hàng mới

Ưu tiên hiển thị loại món ăn phù hợp theo thời gian dùng bữa (ăn sáng, ăn trưa, ăn tối)

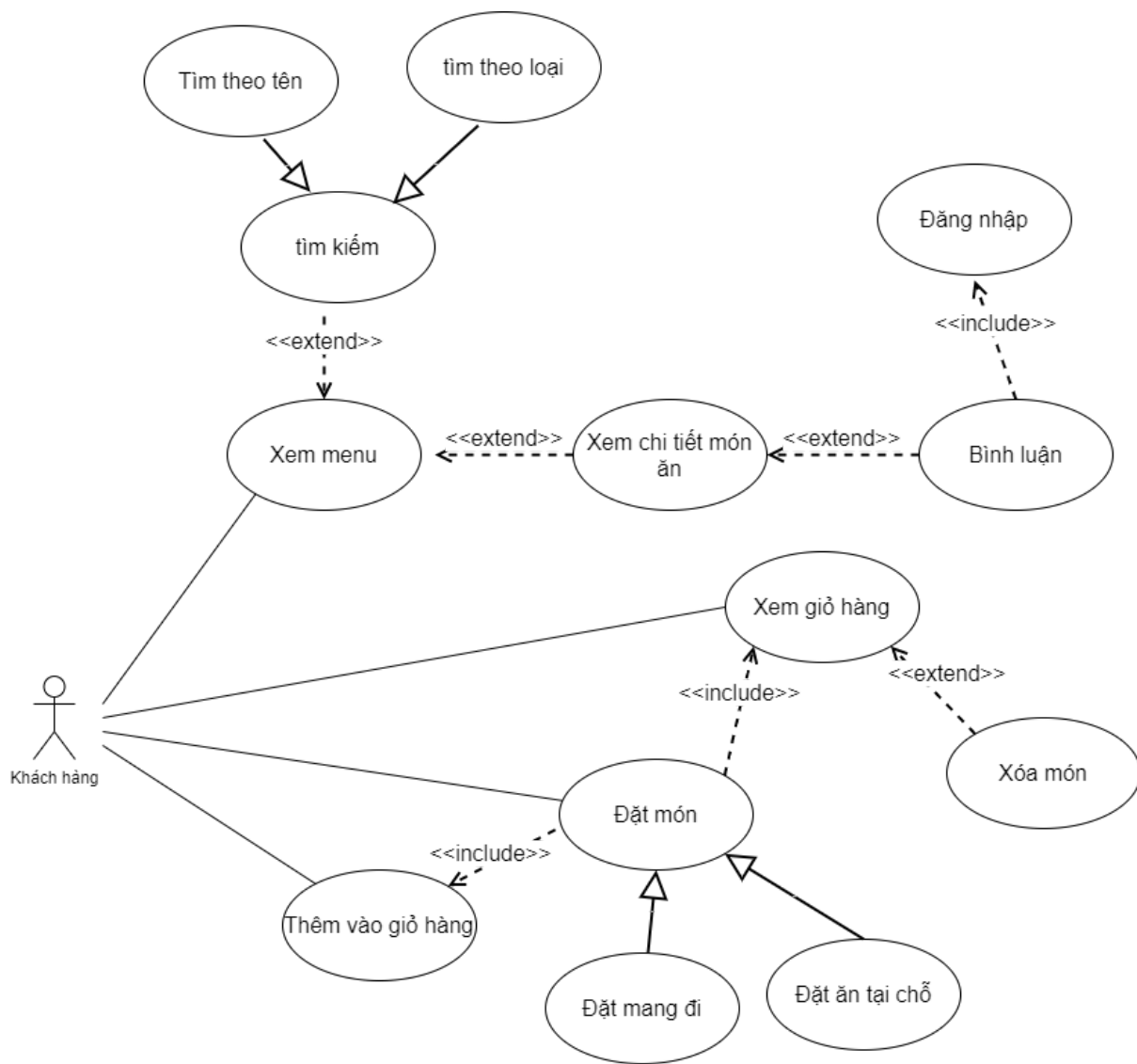
c. Use-case

POINT OF SALE SYSTEM



Task 1.3. Mô tả use-case

a. Đặt món



Đặc tả use-case

Use case name	Tìm món
Created by	HỒ Công Phúc
Date created	22/9/2021
Actor	Khách hàng
Description	Khách hàng tìm kiếm món ăn
Trigger	Khách hàng nhấn vào biểu tượng tìm kiếm hoặc chọn các mục tùy chọn loại thức ăn được liệt kê
Pre-conditions	Khách hàng vào trang Home
Post-condition	Khách hàng tìm thấy món ăn yêu thích
Normal Flow	<ol style="list-style-type: none"> 1. Tại trang Home, khách hàng nhấn vào biểu tượng tìm kiếm, gõ từ khóa và nhấn nút Tìm kiếm 2. Hệ thống trả về món ăn phù hợp 3. Khách hàng nhấn nút Xem bình luận 4. Hệ thống trả về những bình luận của món ăn
Alternative Flows	1.a Trong thanh hiển thị danh sách loại thức ăn, bấm chọn loại thức ăn mong muốn.
Exception	Tại bước 2 nếu không có món ăn phù hợp, hệ thống trả về dòng trạng thái “không có món ăn”
Notes and Issues	
Non-functional	Chỉ được chọn 1 lần 1 loại thức ăn trong trường danh sách được hiển thị

Use case name	Xem chi tiết món ăn
Created by	HỒ Công Phúc
Date created	22/09/2021
Actor	Khách hàng
Description	Khách hàng xem miêu tả và bình luận của món ăn
Trigger	Khách hàng nhấn vào biểu tượng hoặc tên món ăn được hiển thị
Pre-conditions	Khách hàng ở tại trang Home
Post-condition	Hiển thị trang thông tin chi tiết món ăn
Normal Flow	<ol style="list-style-type: none"> 1. Tại trang Home, khách hàng nhấn vào biểu tượng hoặc tên món ăn. 2. Hệ thống chuyển đến trang Thông tin chi tiết món ăn
Alternative Flows	<ol style="list-style-type: none"> 1. Tại bước 3, nếu khách hàng chưa đăng nhập, hệ thống sẽ cảnh báo yêu cầu chuyển về trang Đăng nhập để đăng nhập 2. Nhấn nút Chọn để đến trang đăng nhập hoặc nhấn nút Hủy để hủy bình luận.
Exception	
Notes and Issues	
Non-functional	- Hiển thị 10 bình luận gần nhất

Use case name	Thêm vào giỏ hàng
Created by	HỒ Công Phúc
Date created	22/9/2021
Actor	Khách hàng
Description	Khách hàng thêm vào giỏ hàng những món ăn mong muốn để đặt món sau này
Trigger	Khách hàng nhấn nút thêm vào giỏ hàng tại trang Home hoặc trang chi tiết món ăn
Pre-conditions	Khách hàng ở trang Home hoặc trang Chi tiết món ăn
Post-condition	Món ăn được thêm vào giỏ và lưu xuống database
Normal Flow	<ol style="list-style-type: none"> 1. Tại trang Home hoặc trang chi tiết món ăn, khách hàng nhấn vào nút thêm vào giỏ hàng 2. Hệ thống hiển thị thông báo nhập số lượng 3. Khách hàng nhấn OK để thêm vào giỏ hàng. 4. Hệ thống hiển thị thông báo thêm vào giỏ thành công
Alternative Flows	<p>3a. Khách hàng nhấn hủy</p> <p>4a. Hệ thống tắt dialog xác nhận thêm vào giỏ hàng</p>
Exception	Tại bước 4, nếu hệ thống không thể thêm món ăn vào giỏ hàng sẽ hiển thị thông báo thêm vào giỏ thất bại
Notes and Issues	
Non-functional	

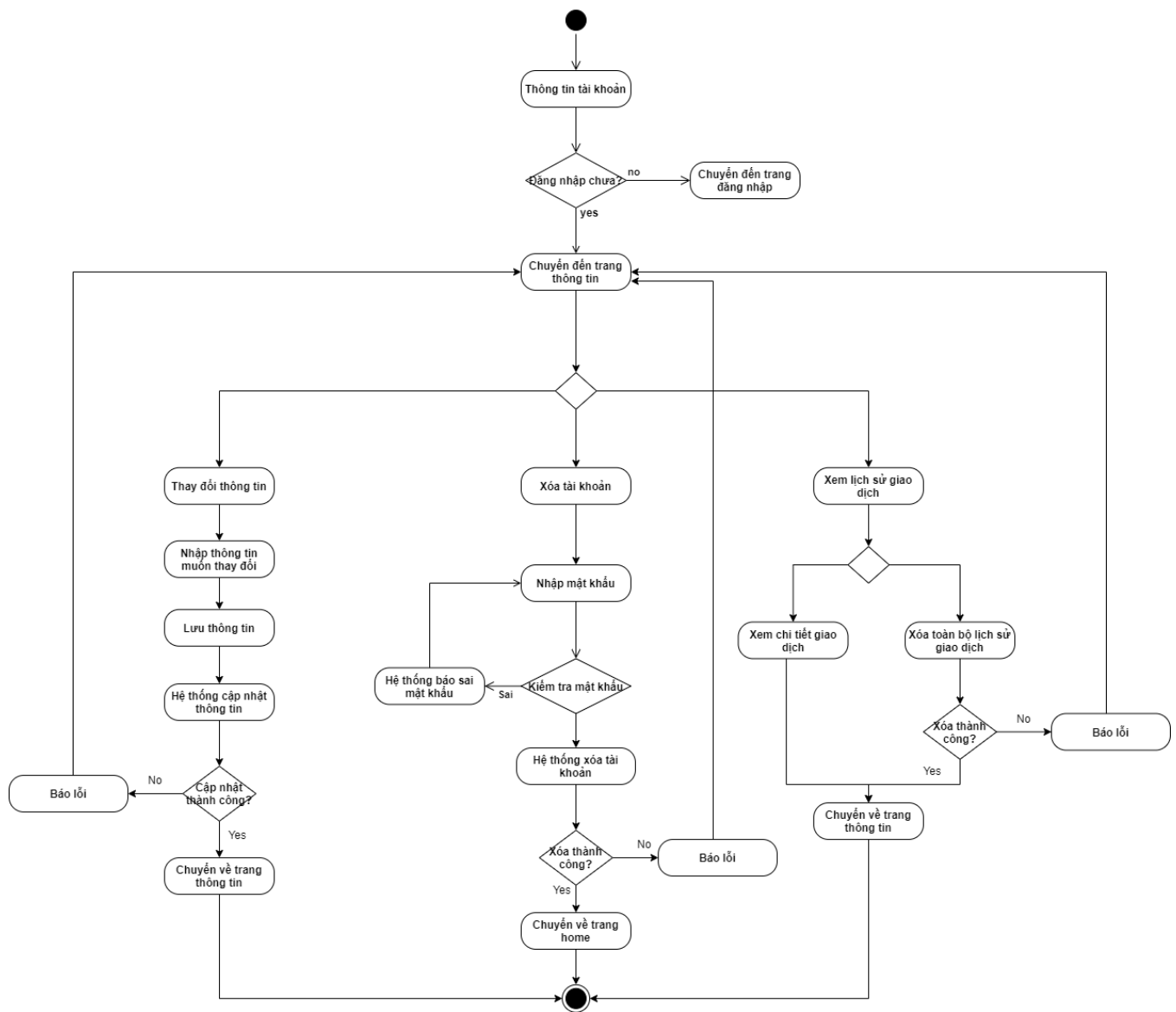
Use case name	Đặt món
---------------	---------

Created by	HỒ Công Phúc
Date created	22/9/2021
Actor	Khách hàng
Description	Khách hàng chọn món và đặt món ăn mong muốn
Trigger	Khách hàng nhấn nút đặt hàng tại trang giỏ hàng
Pre-conditions	Khách hàng đang ở trang Giỏ hàng và giỏ hàng ít nhất có 1 món ăn
Post-condition	Chuyển đến đầu bếp để duyệt đơn hàng
Normal Flow	<ol style="list-style-type: none"> 1. Khách hàng nhấn nút đặt hàng tại trang giỏ hàng 2. Hệ thống hiển thị dialog để chọn phương thức đặt hàng 3. Chọn phương thức đặt hàng: “tại chỗ” hay “mang đi” 4. Khách hàng chọn số lượng món 5. Nhấn OK để tiến hành đặt món 6. Hệ thống hiển thị thông báo đặt hàng thành công
Alternative Flows	5a. Khách hàng nhấn Hủy và hệ thống tắt dialog 6a. Hệ thống báo lỗi khi đặt hàng không thành công
Exception	Các món ăn đã hết được làm mờ và không chọn được
Notes and Issues	
Non-functional	Thời gian chờ xác nhận từ nhà bếp không quá 5 phút

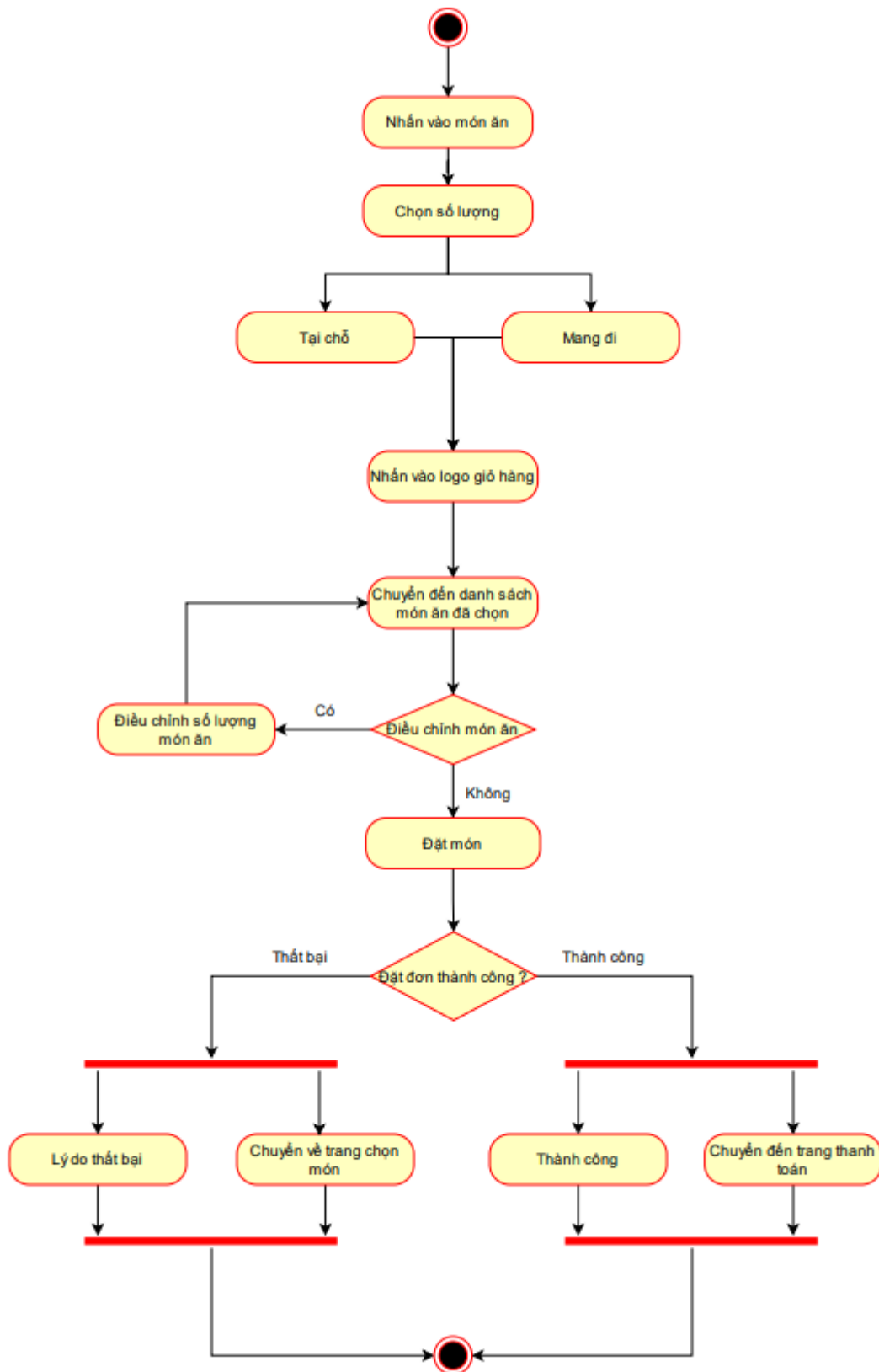
Task 2

Task 2.1 Activity Diagram

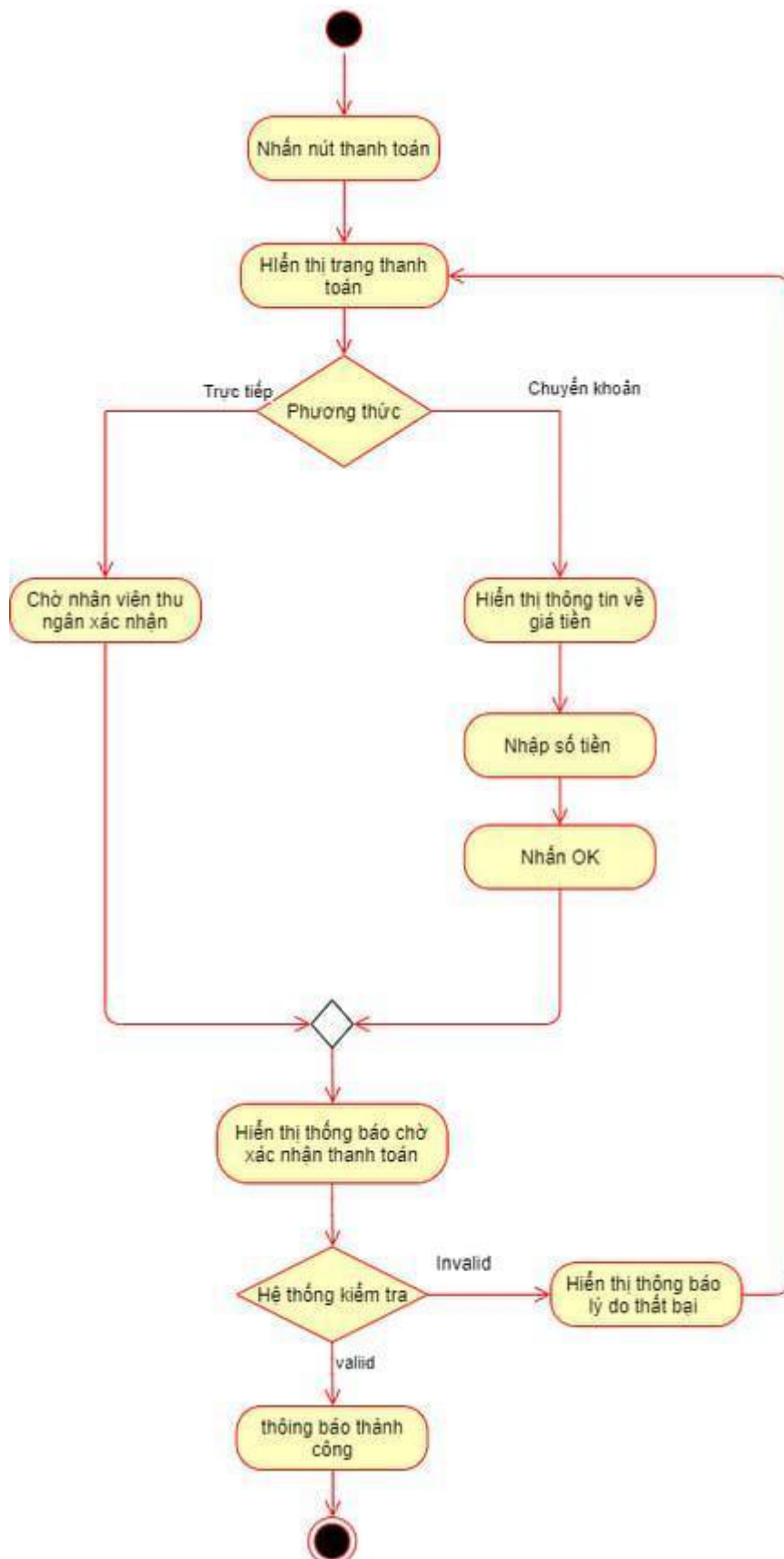
1. Quản lý tài khoản



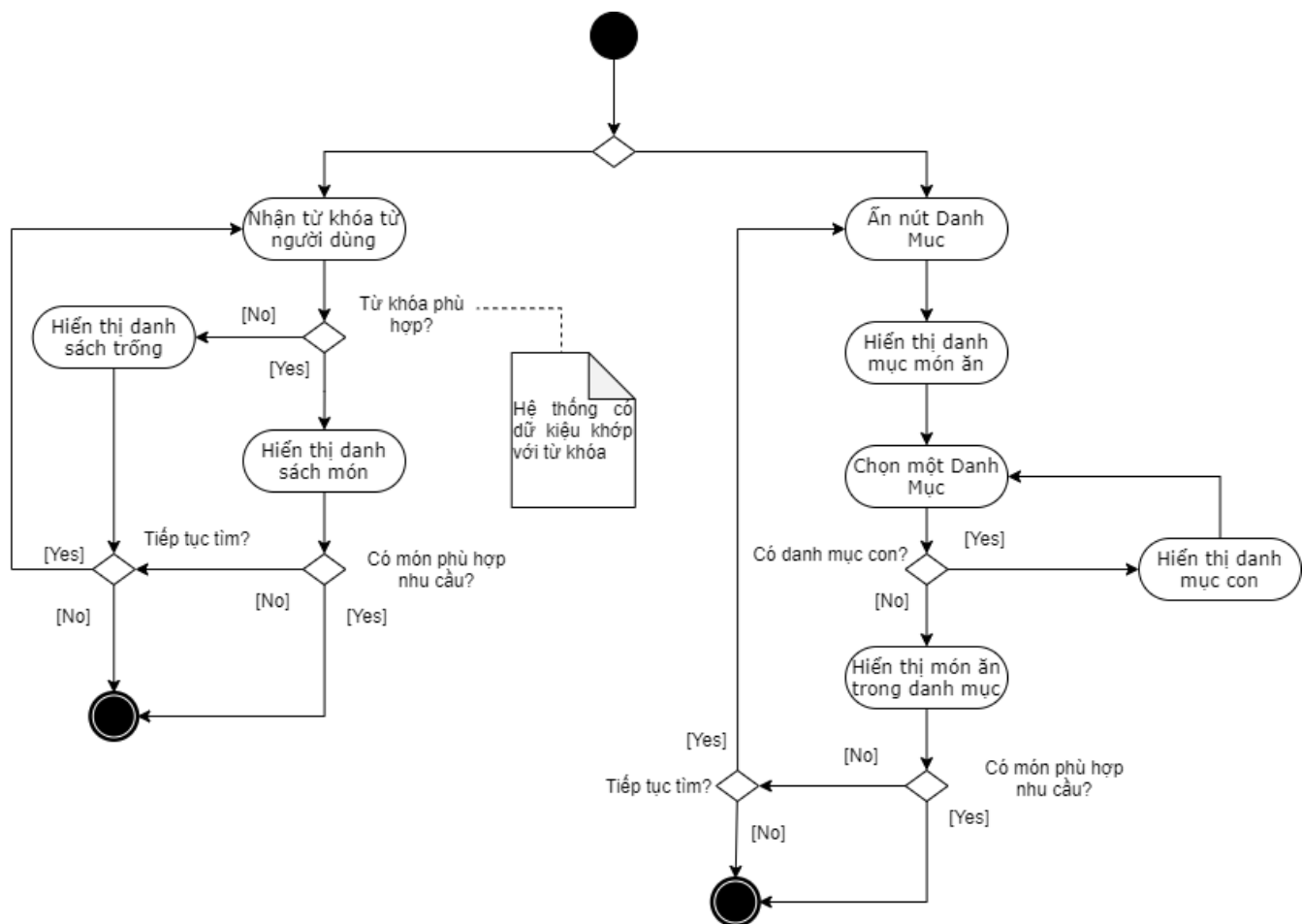
2. Đặt món



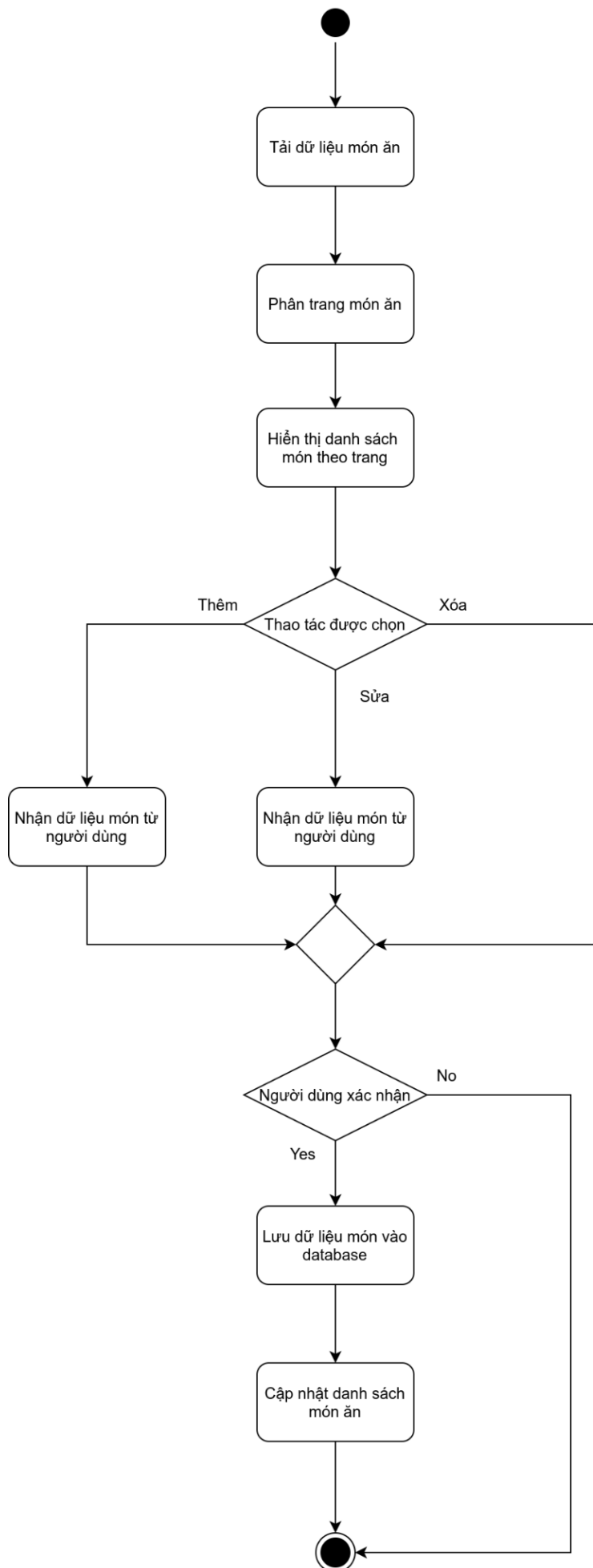
3. Thanh toán



4. Tìm kiếm món ăn

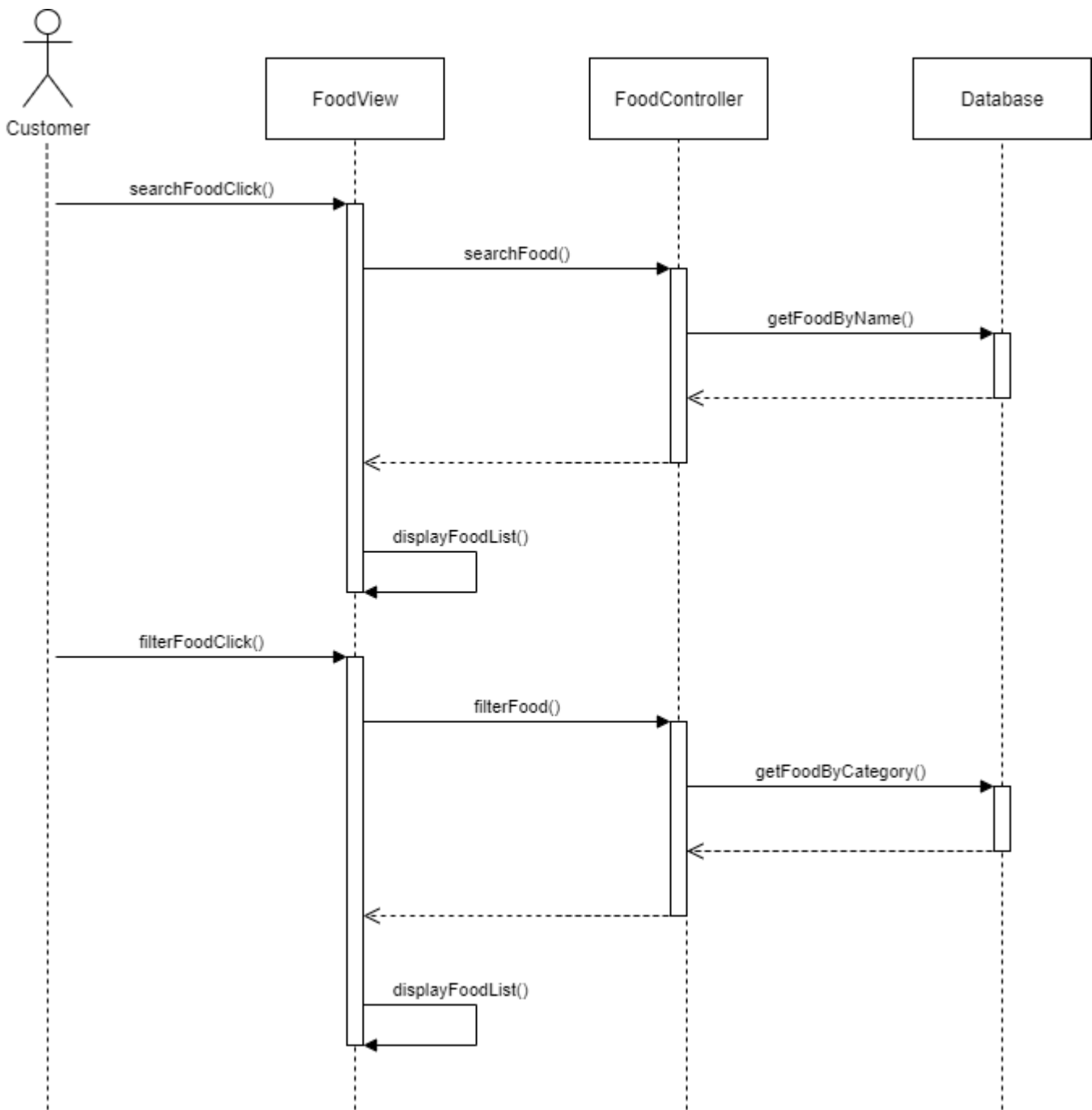


5. Quản lý món ăn

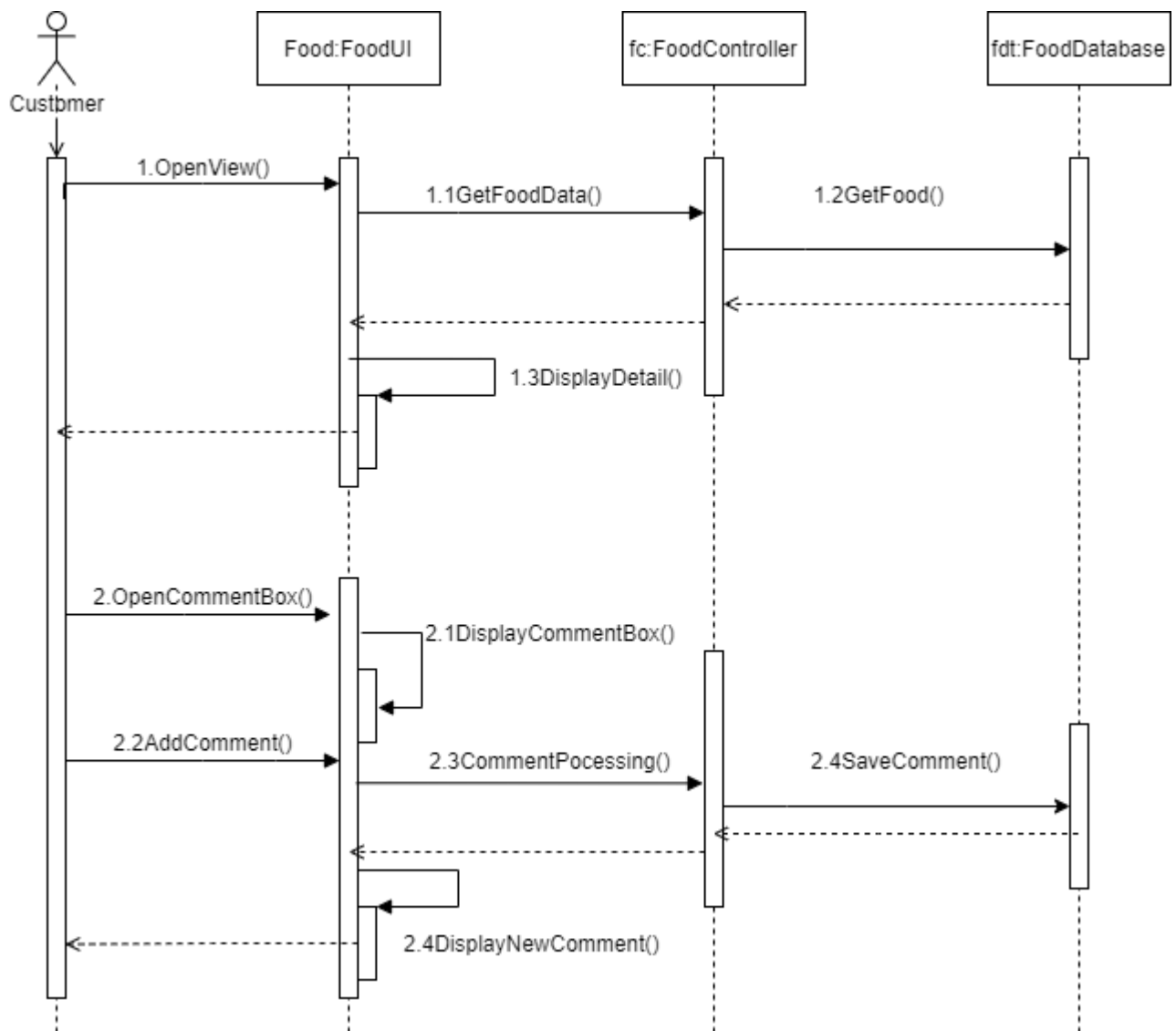


Task 2.2 Sequence Diagram

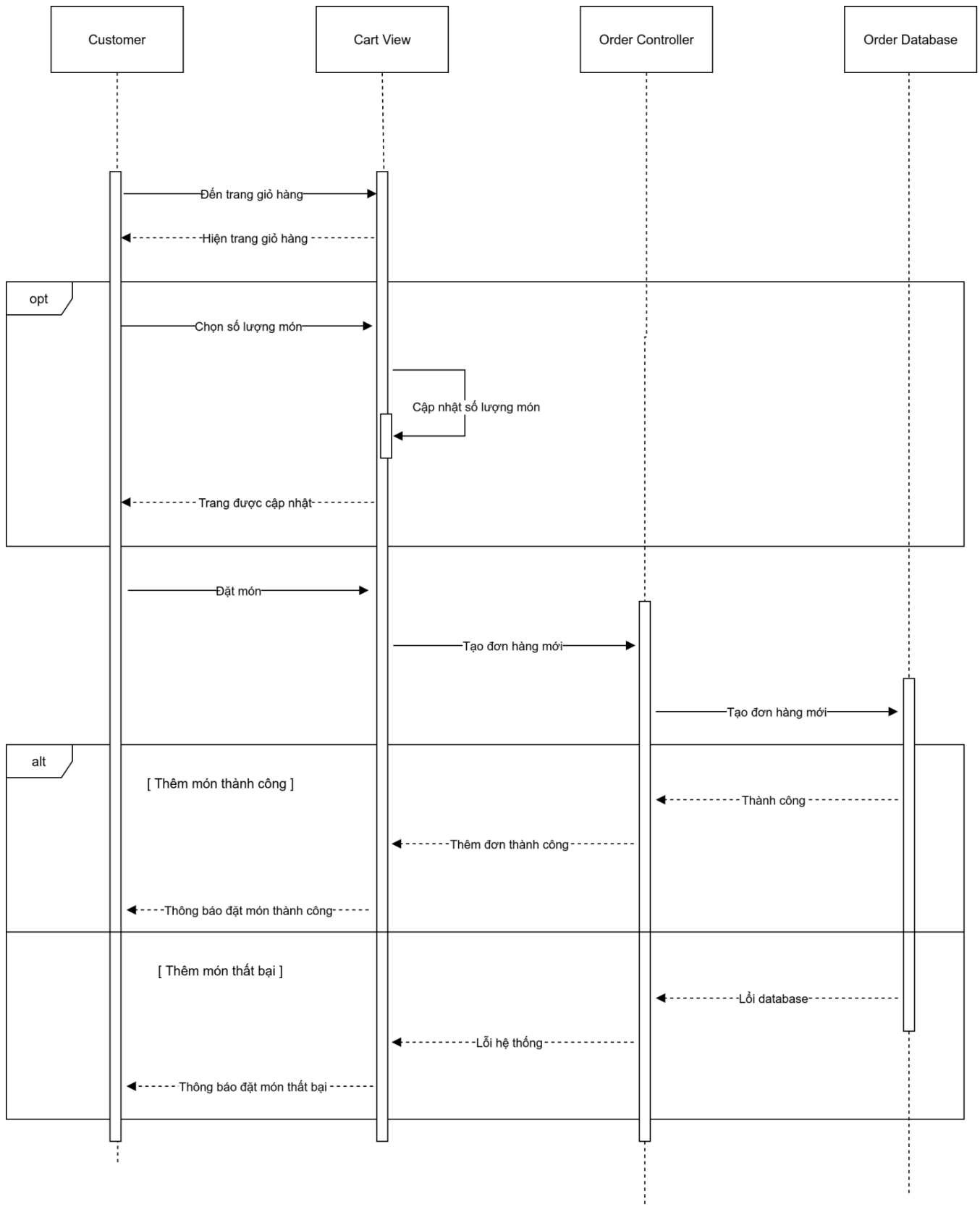
1. Tìm món



2. Xem chi tiết món ăn và bình luận



3. Đặt món



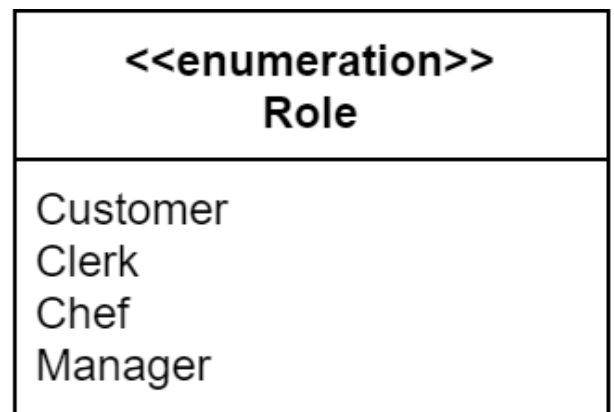
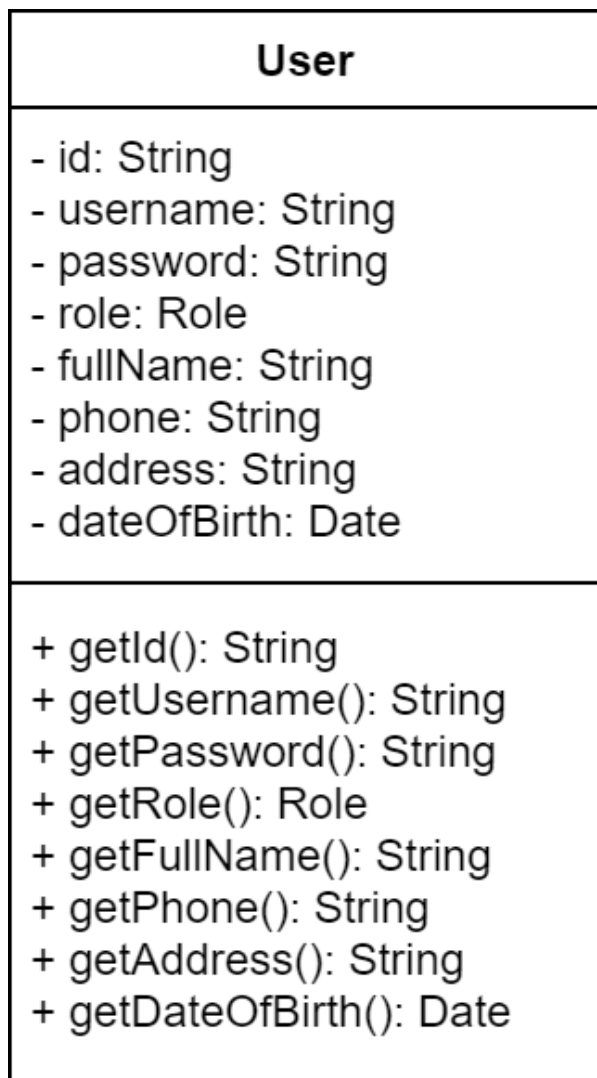
Task 2.2 Class diagram

1. Tổng quan

1.1 Entity Class

Class User dùng để lưu trữ các thông tin đăng nhập (username, password) và thông tin cá nhân của người dùng (phone, address,...)

Ngoài ra class User còn chứa thông tin về vai trò của người dùng thể hiện bởi enumeration Role (khách hàng, nhân viên, đầu bếp, quản lý), vai trò là cách nhận biết những chức năng mà người dùng được sử dụng trong hệ thống.



Class Food dùng để lưu trữ thông tin món ăn và loại món ăn, gồm các trường tên, mô tả món, loại món, đường dẫn tới hình minh họa và giá món ăn.

Enumeration FoodType thể hiện loại món của món ăn.

Food
<ul style="list-style-type: none">- id: String- name: String- description: String- type: FoodType- imageUrl: String- price: Double
<ul style="list-style-type: none">+ getId(): String+ getName(): String+ getDescription(): String+ getFoodType(): FoodType+ getImageUrl(): String+ getPrice(): Double

<<enumeration>> FoodType
<ul style="list-style-type: none">StarterMainDishDessertBeverageOther

Class Order dùng để lưu trữ thông tin đơn đặt hàng sau khi người dùng đặt hàng, gồm các trường: món đã đặt, số lượng, người đặt, tình trạng, ngày đặt, lý do hủy đơn, ghi chú thêm.

Enumeration OrderState thể hiện các tình trạng của đơn hàng

- Pending: món đang được làm
- Delivering: đang giao cho khách
- Finished: đã giao cho khách
- Cancelled: đơn hàng bị hủy

Order
<ul style="list-style-type: none">- id: String- food: Food- amount: Int- buyer: User- state: OrderState = Pending- date: Date- reason: String = ""- note: String = ""
<ul style="list-style-type: none">+ getId(): String+ getFood(): Food+ getAmount(): Int+ getBuyer(): User+ getState(): OrderState+ getDate(): Date+ getReason(): String+ getNote(): String

<<enumeration>> OrderState
<div>Pending</div> <div>Delivering</div> <div>Finished</div> <div>Cancelled</div>

Class Bill lưu trữ thông tin về hóa đơn thanh toán, gồm các trường các đơn hàng đã đặt, tổng tiền, phương thức trả tiền (gồm tiền mặt, online banking, ví điện tử MOMO)

Bill
<ul style="list-style-type: none"> - id: String - orders: List<Order> - payment: Double - paymentMethod: PaymentMethod - date: Date
<ul style="list-style-type: none"> + getId(): String + getOrders(): List<Order> + getPayment(): Double + getPaymentMethod(): PaymentMethod + getDate(): Date

<<enumeration>> PaymentMethod
Cash Momo OnlineBanking

Class QRCode lưu trữ về đường link dẫn tới menu của quán ăn

QRCode
<ul style="list-style-type: none"> - id: String - link: String
<ul style="list-style-type: none"> + getId() + getLink(): String

Class Sale lưu trữ thông tin về chương trình khuyến mãi của quán ăn, gồm các trường các món được khuyến mãi, tỷ lệ giảm giá, thời gian bắt đầu, thời gian kết thúc

Sale
<ul style="list-style-type: none">- foods: List<Food>- percentage: Double- startTime: DateTime- endTime: DateTime
<ul style="list-style-type: none">+ getFoods(): List<Food>+ getPercentage(): Double+ getStartTime(): DateTime+ getEndTime(): DateTime

1.2 Entry Point Class

Vì nhóm thiết kế hệ thống theo mô hình Client – Server nên phân hệ thống gồm các server Frontend và Backend.

1.2.1 Frontend

Class FrontendApp lưu trữ, hiển thị các View và làm entry point cho Frontend Server

FrontendApp
- foodView: FoodView ...
+ render() + handleUrl(url: String) + run()

Method render sẽ thực hiện thị các View cần thiết tại đường dẫn đang xử lý

Method handleUrl sẽ xử lý việc chuyển đổi giữa View tương ứng tại đường dẫn truyền vào

Method run: khởi động Frontend Server

1.2.2 Backend

Class BackendApp lưu trữ các module chức năng của hệ thống, xử lý logic business và làm entry point cho Backend Server

BackendApp
- foodRouter: FoodRouter - foodController: FoodController - databaseDriver: DatabaseDriver ...
+ handleUrl(url: String, request: Request) + run()

Method handleUrl: nhận request tại đường dẫn tương ứng và chuyển xuống các module tương ứng để xử lý và trả response

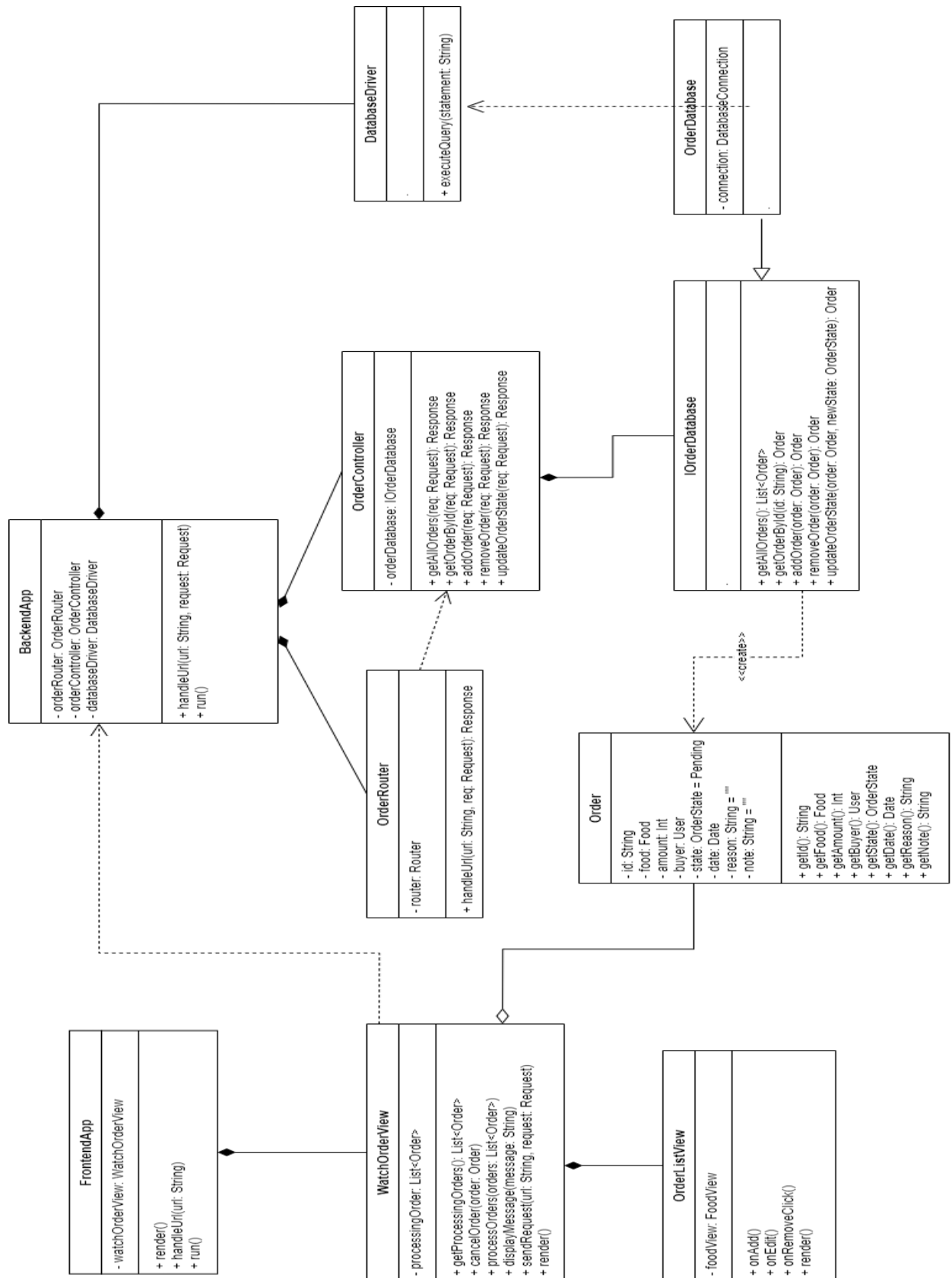
Method run: khởi động Backend Server

Một chức năng của hệ thống thường các class sau đảm nhận:

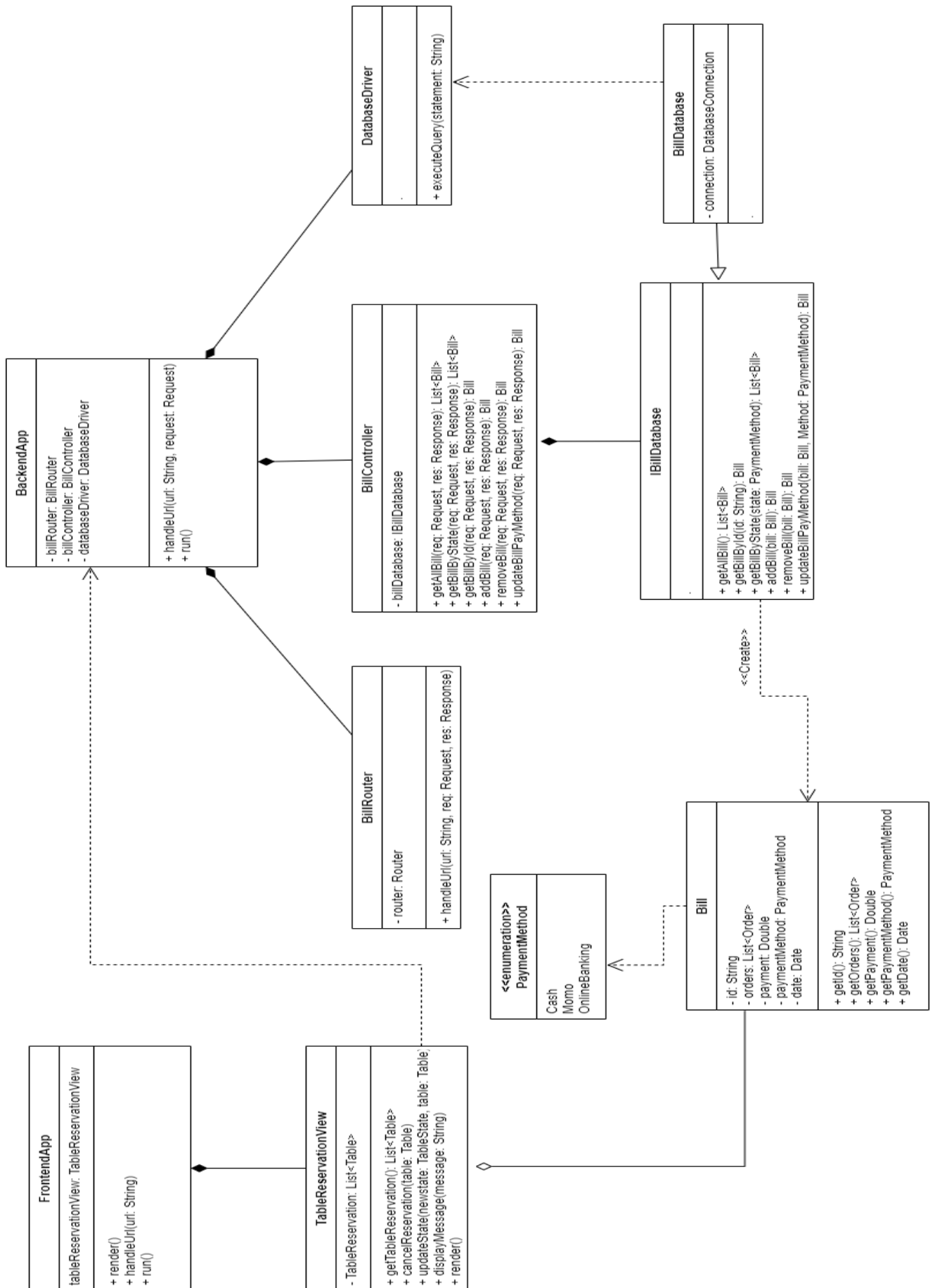
- Router: nhận request và chọn Controller tương ứng để xử lý
- Controller: nhận request từ Router và kiểm tra tính hợp lệ của request, xử lý logic business, giao tiếp với database
- Database: cung cấp các hàm để Controller có thể lấy các dữ liệu từ database

2. Class Diagram

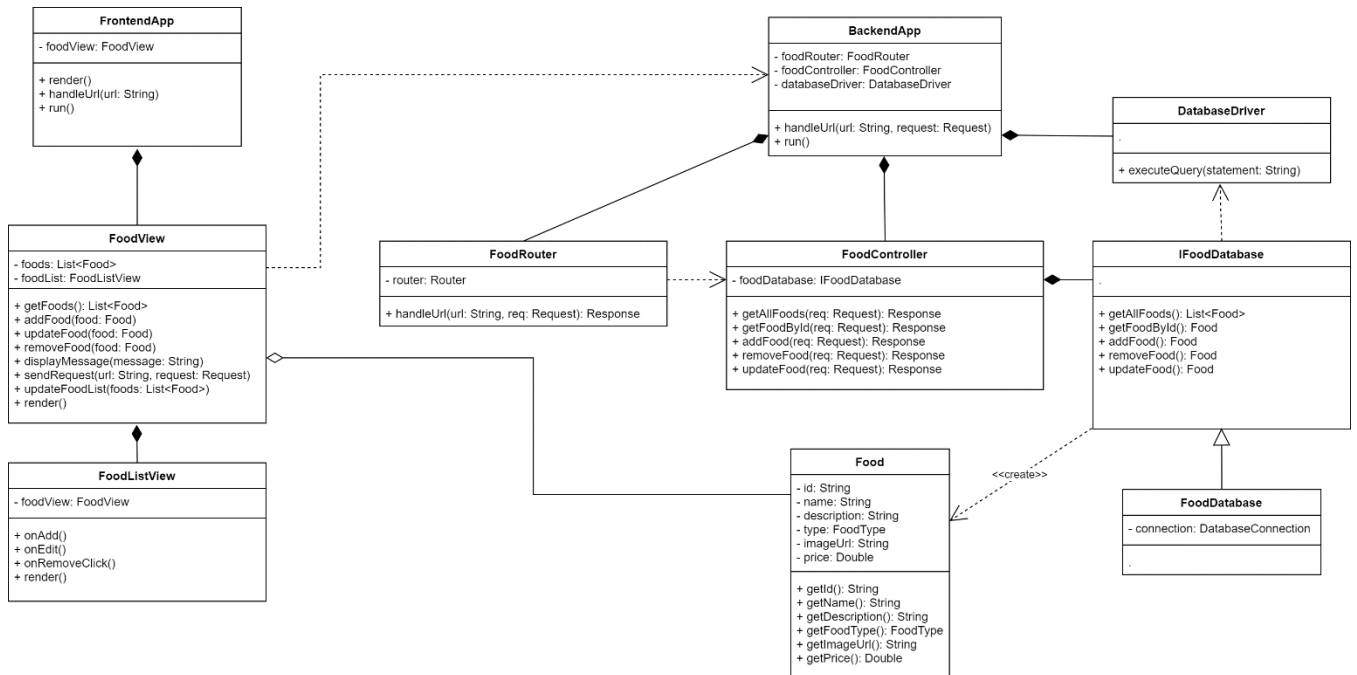
1. Xem đơn hàng cũ



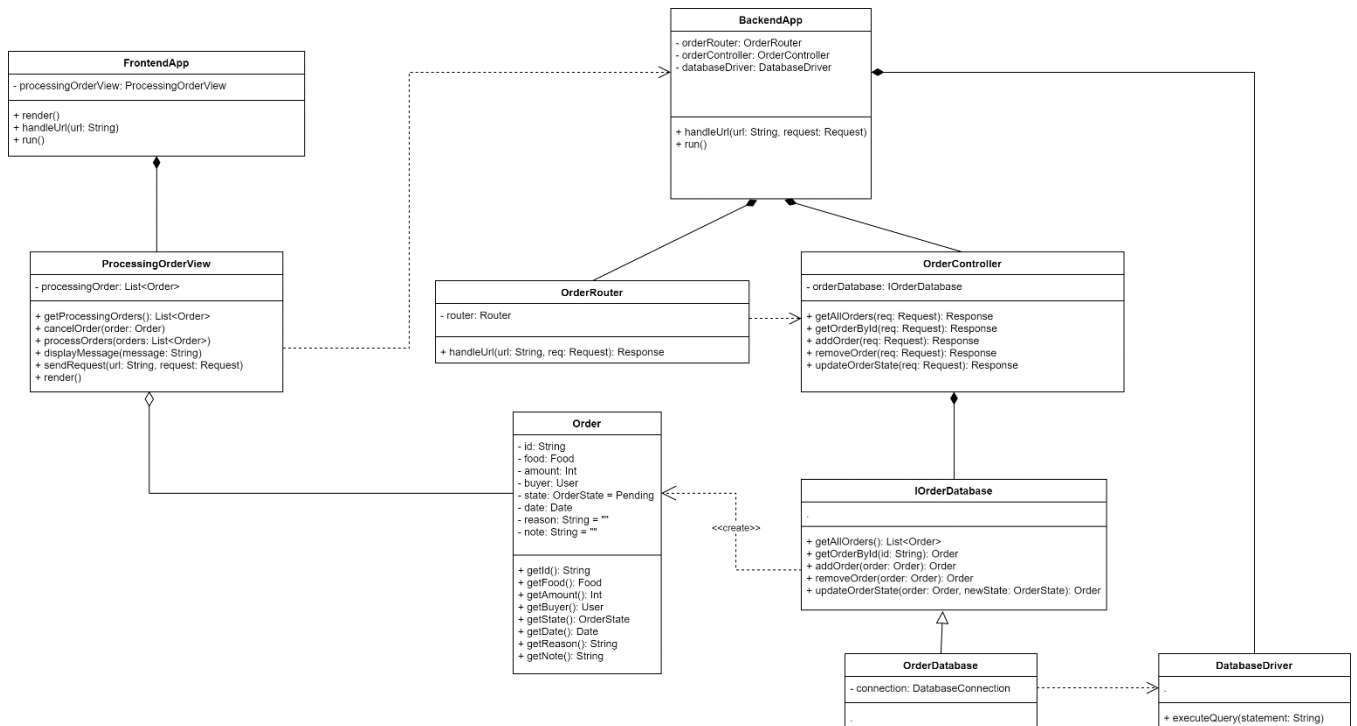
2. Tạo hóa đơn mới



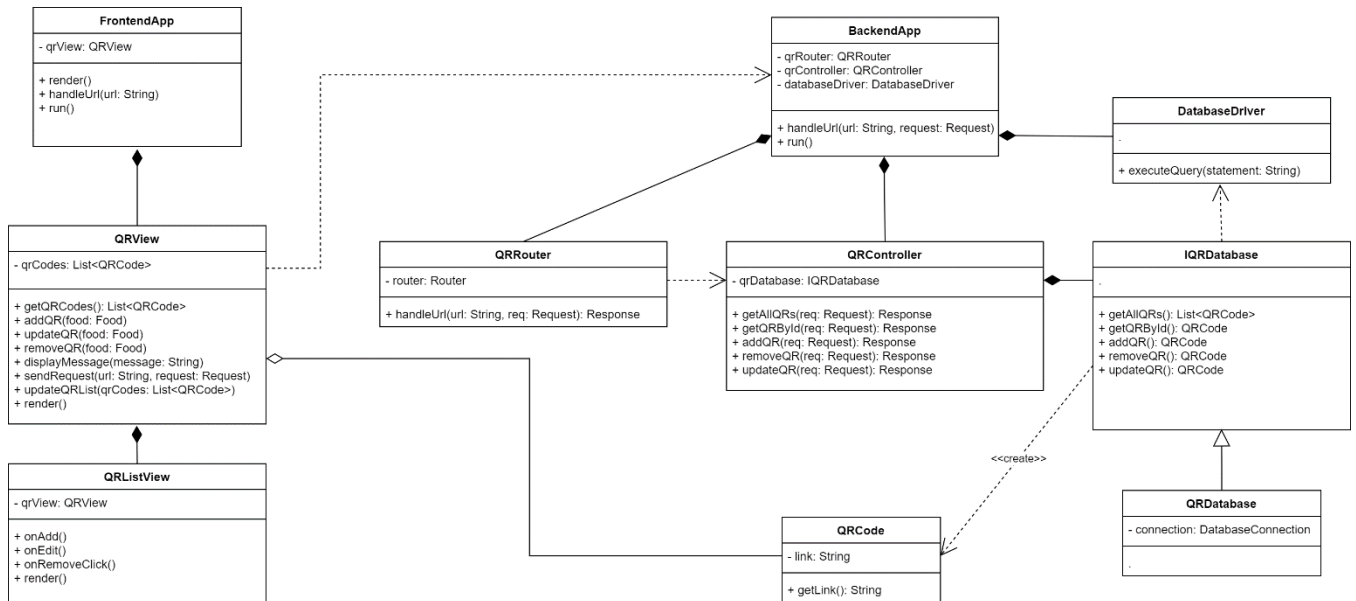
3. Quản lý món ăn



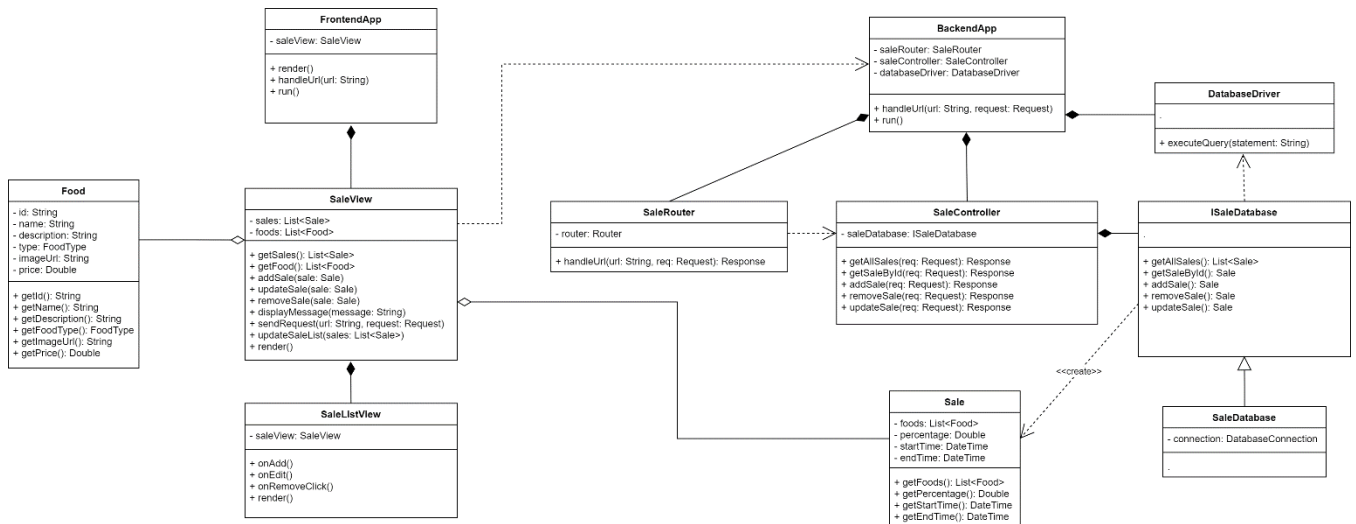
4. Quản lý đơn hàng



5. Quản lý mã QR

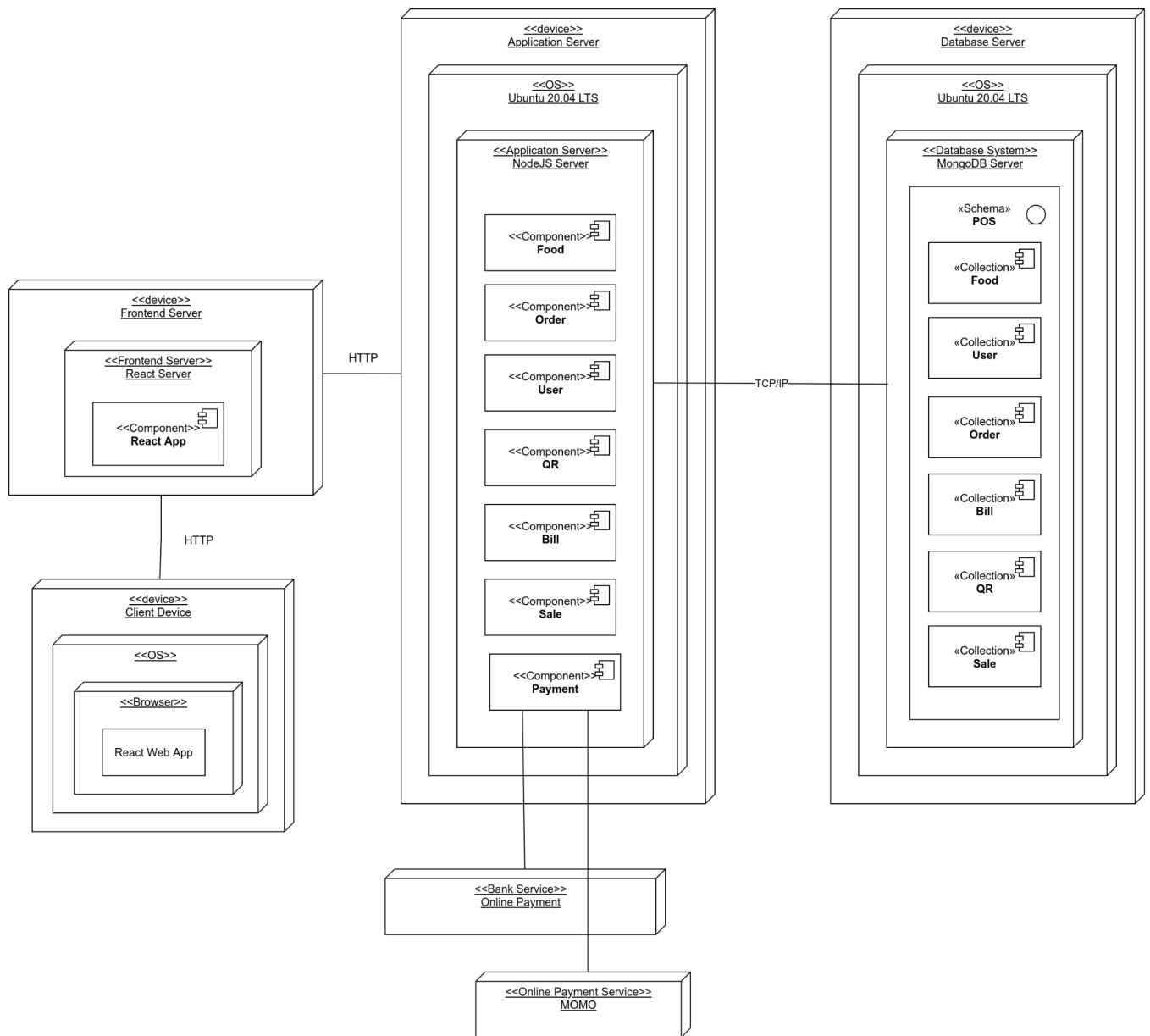


6. Quản lý chương trình khuyến mãi



Task 3

1. Kiến trúc hệ thống



Mô tả hệ thống:

Hệ thống được thiết kế theo mô hình Client - Server, gồm các thành phần:

Frontend Server: xử lý view, hiển thị giao diện để người dùng tương tác với các chức năng của hệ thống. Phần lớn request liên quan đến việc sử dụng các chức năng sẽ thông qua front-end server trước khi được đưa vào application server xử lý. Được phát triển trên nền ReactJS.

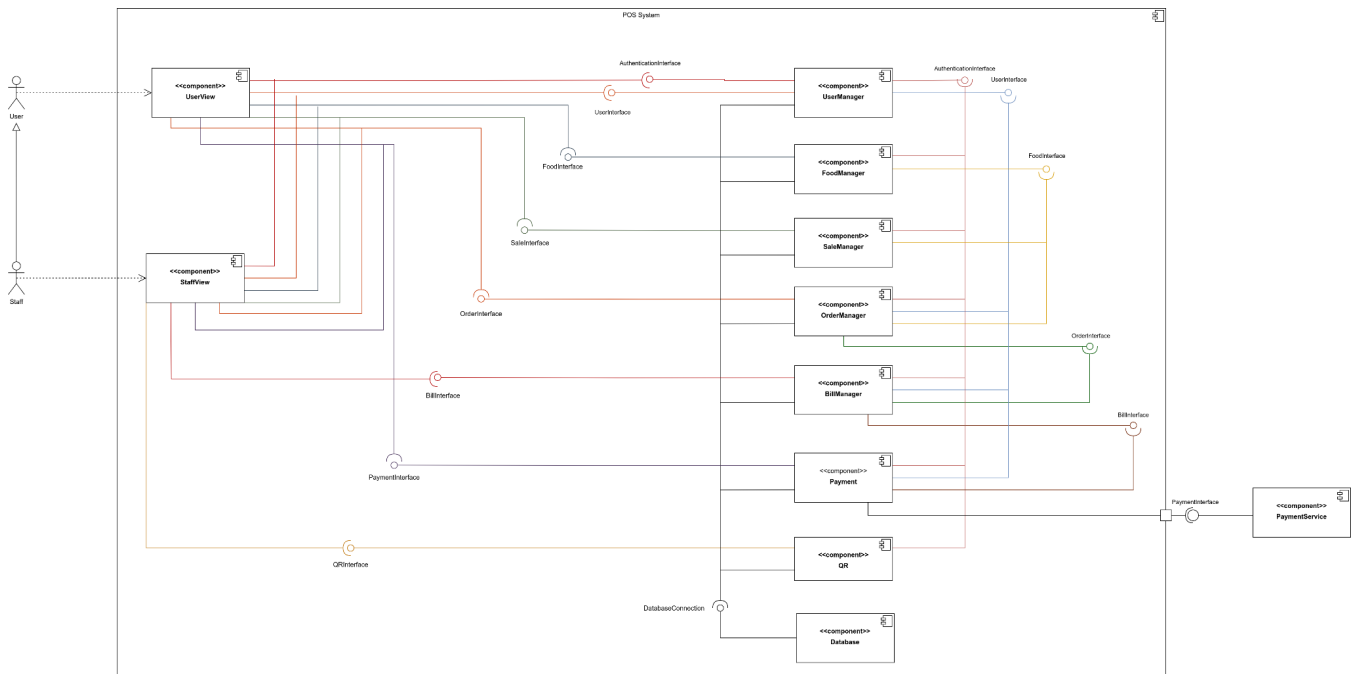
Application Server (Backend Server): xử lý business logic của các chức năng hệ thống, cung cấp các API của các chức năng, tương tác với database server. Được phát triển trên nền NodeJS.

Database Server: nơi cài đặt database system, tổ chức lưu trữ và truy xuất dữ liệu. Được phát triển trên nền MongoDB.

Client: người dùng sẽ sử dụng các chức năng thông qua React Web App thông qua các trình duyệt web. Có thể chạy trên phần lớn các hệ điều hành Windows, Android, IOS,...

Các dịch vụ Online Payment: gồm Internet Banking và Online Payment của MOMO.

2. Component Diagram



Mô tả các component chính của hệ thống:

Component xử lý logic nghiệp vụ:

- UserManager (quản lý người dùng),
- FoodManager (quản lý món)
- OrderManager (quản lý đơn hàng)
- BillManager (quản lý hóa đơn)
- Payment (quản lý thanh toán)
- QR (quản lý mã QR)

Component Payment yêu cầu API các dịch vụ thanh toán thứ ba thông qua PaymentInterface.

Component cung cấp API tương tác với database thông qua DatabaseConnection: Database.

Component xử lý view được nhóm vào 2 nhóm:

- UserView: cho khách hàng thông thường.
- StaffView: cho nhân viên của quán ăn.

Mô tả các interface:

- AuthenticationInterface: cung cấp API cho việc xác thực các request từ client có hợp lệ hay không.
- UserInterface: cung cấp API cho việc lấy, thêm, cập nhật, xóa người dùng.
- FoodInterface: cung cấp API cho việc lấy, thêm, cập nhật, xóa món ăn.
- OrderInterface: cung cấp API cho việc lấy, thêm, cập nhật đơn hàng.
- BillInterface: cung cấp API cho việc lấy, thêm, cập nhật hóa đơn.
- PaymentInterface: cung cấp API trung gian cho việc thanh toán qua bên thứ ba.
- QRInterface: cung cấp API cho việc lấy, thêm, cập nhật, xóa mã QR.
- SaleInterface: cung cấp API cho việc lấy, thêm, cập nhật, xóa chương trình khuyến mãi.

Task 4

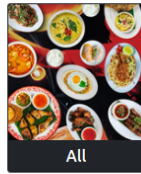
1. Project Versioning

Nhóm sử dụng git để quản lý phiên bản ứng dụng

Link repository: <https://github.com/Cyberproton/pos-system>

2. Một số hình ảnh về ứng dụng

Categories



All



Starter



Main



Dessert



Drink

Category: All



Hamburger

Price: \$200

Add to Cart



Coca

Price: \$50

Add to Cart



Flan

Price: \$200

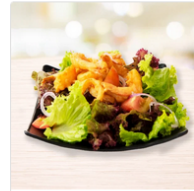
Add to Cart



Beef Steak

Price: \$10000

Add to Cart



Salad

Price: \$100

Add to Cart

1. Trang chọn món - Tất cả món

Categories



All



Starter



Main



Dessert



Drink

Category: Drink



Coca

Price: \$50

Add to Cart













Avocado Smoothie

Price: \$100

Add to Cart

2. Trang chọn món - Lọc theo category Drink

YOUR CART











ORDERED DISHES		PRICE	QUANTITY			TOTAL	
1. Hamburger <i>Starter</i>		200\$	—	1	+	200\$	
2. Coca <i>Drink</i>		200\$	—	1	+	200\$	
3. Beef Steak <i>Main</i>		10000\$	—	1	+	10000\$	
4. Crab Soup <i>Starter</i>		100\$	—	1	+	100\$	
5. Avocado Smoothie <i>Drink</i>		100\$	—	1	+	100\$	

Subtotal: 10600\$

CHECKOUT

3. Trang giỏ hàng

YOUR CART

ORDERED DISHES		PRICE	QUANTITY			TOTAL	
1. Hamburger <i>Starter</i>		200\$	—	2	+	400\$	
2. Coca <i>Drink</i>		200\$	—	5	+	1000\$	
3. Beef Steak <i>Main</i>		10000\$	—	1	+	10000\$	
4. Crab Soup <i>Starter</i>		100\$	—	4	+	400\$	
5. Avocado Smoothie <i>Drink</i>		100\$	—	2	+	200\$	

Subtotal: 12000\$

CHECKOUT

4. Trang giỏ hàng - Đã thay đổi số lượng món