HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

**FACULTY OF COMPUTER SCIENCE AND ENGINEERING**

****

**BÁO CÁO BÀI TẬP LỚN**

**CÔNG NGHỆ PHẦN MỀM - CO3001**

**Group 47**

**RESTAURANT POS 2.0**

**GVHD: GV. Lê Đình Thuận**

**Thành viên thực hiện: Bành Thái Phương 1732027**

**Phạm Ngọc Thân 1915262**

**Lê Phú Thiện 1814132**

**Nguyễn Ngọc Đức Tâm 1613055**

**Nguyễn Thế Duy 1912912**

**Tháng 9 năm 2021**

# **Mục Lục**

[**Mục Lục**](#_721x00kesdtn) **2**

[**#1: Requirement elicitation**](#_6d6mji7bu2yq) **4**

[1.1. Identify the context of this project. Who are relevant stakeholders? What are expected to be done? What are the scope of the project?](#_8fw8qq7on1jh) 4

[1.1.1 Identify the context of this project](#_yr48mrizhrkg) 4

[1.1.2 Relevant stakeholders in system](#_4hr1u8ddlft) 4

[1.1.3 What are expected to be done?](#_ka63ow12q0c) 5

[1.1.4 The scope of the project](#_je4a45mc9nwy) 5

[1.2. Describe all functional and non-functional requirements of the desired system. Draw a use-case diagram for the whole system](#_i2nzfmsp87nu) 6

[1.2.1 Functional requirements](#_lqnt7n84jq7k) 6

[1.2.1.a Use-case diagram](#_lzc32xmw767) 6

[1.2.1.b Descriptions](#_ydqg4vmeoc8i) 7

[1.2.2 Non-Functional requirements](#_s3kx4ajxy48g) 8

[1.3. Choose one specific feature, i.e. food ordering, table reservation, customer management. Draw its use-case diagram and describe the use-case using a table format](#_wi95nosh7ibt) 8

[1.3.1 Đặt món (Tâm)](#_u2bqiwb5efuk) 8

[1.3.2 Xác nhận đơn hàng (Tâm)](#_96hwxoygsxid) 11

[1.3.3 Thanh toán online (Thiện)](#_xfg1xb31rfr8) 12

[1.3.4 Theo dõi đơn hàng (Thiện)](#_acdyv2x8h090) 13

[1.3.5 Thay đổi trạng thái đơn hàng (Duy)](#_8zuo4vjfxacw) 13

[1.3.6 Đánh giá (Duy)](#_tiscm3ppk51n) 14

[1.3.7 Xem thông tin nhà hàng (Duy)](#_f71bp6moqika) 14

[1.3.8 Quản lý tài khoản (Thân)](#_l270e7yfkwsk) 15

[1.3.9 Quản lý tài khoản nhân viên (Thân)](#_k7xz4pko4b3h) 18

[1.3.10 Xem thông tin lịch sử đơn hàng (Thân)](#_xc8aec1yabvf) 19

[1.3.11 Quản lý menu (Phương)](#_jsg6grattr0v) 19

[1.3.12 Quản lý QR code (Phương)](#_6287zhkh6r3x) 21

[1.3.13 Quản lý thông tin giới thiệu (Group work)](#_ber8wu11qrd) 22

[1.3.14 Quản lý đánh giá khách hàng (Group work)](#_didr17351og4) 23

[**#2: System modeling**](#_pujh5vjqh2yv) **24**

[2.1. Draw an activity diagram to capture Major (not all) functional requirements of the desired system](#_o9sy20jkyzqg) 24

[2.2. Draw a sequence diagram for use-case in Task 1.3.](#_ef1hvxaumbq8) 27

[2.2.1 Đặt món](#_s5e19bhmxjk7) 27

[2.2.2 Xác nhận đơn hàng](#_2mxw4xrjbhca) 28

[2.2.3 Thanh toán online](#_47tvjx7zll53) 28

[2.2.4 Theo dõi đơn hàng](#_1vindlky44sx) 29

[2.2.5 Thay đổi trạng thái đơn hàng](#_jhbpyvzf70ta) 29

[2.2.6 Đánh giá](#_j28qvcb30mho) 30

[2.2.7 Xem thông tin nhà hàng](#_jnl1397w3ife) 30

[2.2.8 Quản lý tài khoản khách hàng](#_3gha7g1pmtgc) 31

[2.2.9 Quản lý tài khoản nhân viên](#_w0igso9ohvx7) 31

[2.2.10 Xem thông tin lịch sử đơn hàng](#_ltbhko888dss) 34

[2.2.11 Quản lý menu](#_yid8xtriorf5) 35

[2.2.12 Quản lý QR code](#_p6wdzjfqfcx) 36

[2.2.13 Quản lý thông tin giới thiệu](#_deskjsafy21s) 37

[2.2.14 Quản lý đánh giá khách hàng](#_uys6cmkwladu) 37

[2.3. Draw a class diagram](#_cw6s47l4kl68) 38

[**#3: Architecture design**](#_swgah3y6yn5j) **38**

[3.1. Describe an architectural approach you will use to implement the desired system](#_auwvuisoddw3) 38

[3.1.1 Technology](#_nbrunoblihq) 38

[3.1.2 Overall architecture](#_mm2moukb20of) 41

[3.2. Draw an implementation diagram for Major (not all) functional requirements](#_4txlz9ogyitg) 44

# **#1: Requirement elicitation**

**The team needs to perform the following tasks:**

## **1.1. Identify the context of this project. Who are relevant stakeholders? What are expected to be done? What are the scope of the project?**

### **1.1.1 Identify the context of this project**

**Restaurant POS 2.0** là hệ thống sử dụng trong các nhà hàng giúp khách hàng dễ dàng và tiện dụng trong việc chọn thực đơn, đặt đồ ăn và thanh toán trực tiếp trên website nhà hàng thông qua điện thoại của khách hàng. Khách hàng chỉ cần quét mã QR được đặt tại các bàn ăn thông qua camera điện thoại là có thể gọi món.

* Project:
* Project type: phát triển từ đầu - Project developed from scratch
* Duration (Thời gian phát triển): Ngắn, khoảng 1 tháng cho nhóm 5 người
* Problem knowledge: ambiguous - Khách hàng thể hiện được yêu cầu, tuy nhiên cần được làm rõ thêm những tính năng cụ thể
* Client involvement: high - liên quan trực tiếp đến khách hàng (ở đây là chủ nhà hàng)
* Team:
* Team size: very restricted: nhóm chỉ có 5 người
* Team expertise: regular - nhóm có kiến thức cơ bản về thực hiện dự án
* Business knowledge: unknown - nhóm chưa có kiến thức về kinh doanh
* Product knowledge: affordable -
* Yêu cầu product:
* Technical complexity: low -
* Quality: minimum - hệ thống chỉ cần chạy với những tính năng cơ bản
* Process:
* Process focus: final product

### **1.1.2 Relevant stakeholders in system**

Stakeholders của hệ thống bao gồm:

* Khách hàng: sử dụng tính năng của hệ thống về tiến hành đặt món, đánh giá từ nhà hàng.
* Nhân viên: họ sẽ cho biết họ muốn thực hiện các tính năng xử lý đơn hàng và một số tính năng khác của hệ thống.
* Chủ cửa hàng: quản lý một số tính năng, quản lý các thông tin nhân viên, thực đơn, … của nhà hàng.
* Nhà cung cấp dịch vụ thanh toán trung gian: họ sẽ cho biết họ muốn thực hiện thanh toán đơn hàng của hệ thống như thế nào.

### **1.1.3 What are expected to be done?**

* **Đặt món**: Đối với khách hàng, để thuận tiện cho việc gọi đồ ăn, khách hàng chỉ cần sử dụng camera từ điện thoại của mình, quét mã QR được đặt trên các bàn ăn và hệ thống sẽ đề xuất và hiển thị website menu của nhà hàng trên điện thoại của bạn, tiến hành đặt món ăn mà có thể không cần nhờ đến trợ giúp của phục vụ. Sau khi lướt menu, danh sách các món ăn và khách hàng tiến hành chọn món ăn và số lượng mong muốn, khách hàng vào giỏ hàng tiến hành xác nhận đặt món.
* **Thanh toán**: Sau khi chờ phục vụ chấp nhận đơn hàng, khách hàng tiến hành thanh toán của mình và chờ đợi, cuối cùng là có thể thưởng thức những món ăn ngon của nhà hàng.
* **Đánh giá**: Khách hàng có thể phản hồi, đánh giá về phục vụ, món ăn, … của nhà hàng và xem đánh giá khác.
* **Theo dõi đơn hàng**: Khách hàng theo dõi tình trạng đơn hàng của mình trong khi chờ. Các trạng thái bao gồm: đang chờ, chấp nhận và yêu cầu thanh toán/từ chối, đang chờ nấu, đang nấu, đang giao, hoàn thành.
* **Thay đổi trạng thái đơn hàng**: Sau khi khách hàng đặt món thành công, phục vụ sẽ xem đơn hàng có thể chấp nhận được không (đưa thông tin số lượng món hợp lý?, nguyên liệu, số lượng món có đủ không?...) và sau đó chấp nhận nó (xác nhận đơn hàng), tiếp đến là chờ khách hàng thanh toán xong, phục vụ sẽ cập nhật trạng thái đơn hàng và tự động chuyển đến phía đầu bếp, đầu bếp xác nhận đơn hàng thành trạng thái đang nấu, thực hiện nấu và sau khi nấu xong, cập nhật lại trạng thái và phục vụ sẽ chuyển đồ ăn cho khách hàng.

Đối với trường hợp khách hàng muốn thanh toán bằng tiền mặt, phục vụ sẽ gặp trực tiếp và thu tiền từ khách hàng, sau đó phục vụ sẽ xác nhận đơn hàng đã thanh toán và hệ thống tự động chuyển cho đầu bếp như thường lệ.

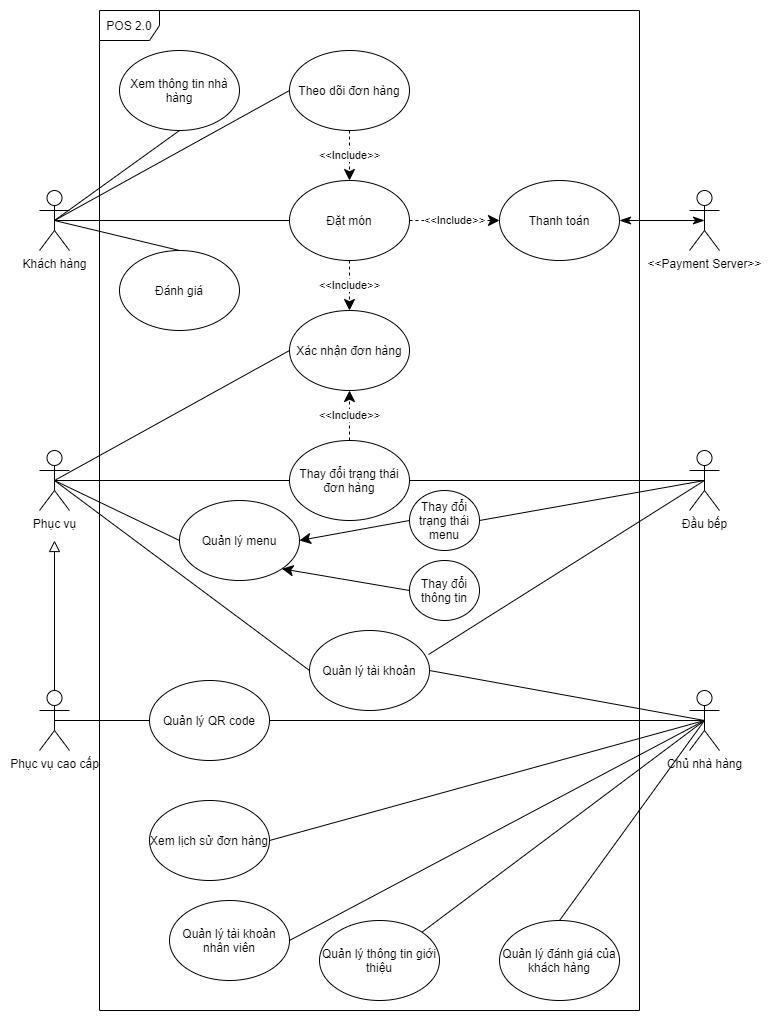
### **1.1.4 The scope of the project**

* Hệ thống hỗ trợ ngôn ngữ: tiếng Việt, tiếng Anh
* Tương thích trên các loại trình duyệt của máy tính như: Chrome, IE; điện thoại người dùng như Safari, Chrome, Opera, Mi browser (Redmi), SIB (Samsung).
* Tương thích giao diện website (Responsive website) trên hầu hết các thiết bị điện thoại của người dùng (kích thước màn hình từ 4.5 inch trở lên).
* Hỗ trợ thực hiện nhiều loại giao dịch, thanh toán: Trực tiếp, ví điện tử Momo, thẻ ATM ngân hàng, thẻ tín dụng.
* Khách hàng có thể sử dụng số điện thoại, tài khoản Google, Facebook để tiến hành đặt món.

## **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 requirements**

#### **1.2.1.a Use-case diagram**

****

#### **1.2.1.b Descriptions**

| **Use-case ID** | **Use-case name** | **Description** |
| --- | --- | --- |
| **UC01** | Xem menu | Khách hàng lướt xem menu, bao gồm danh sách các món ăn theo chủ đề của nhà hàng |
| **UC02** | Lựa chọn món ăn | Khách hàng lựa chọn món ăn, thêm bớt các món ăn mong muốn cho giỏ hàng của mình |
| **UC03** | Xác nhận đặt món ăn | Khách hàng tiến hành đặt món ăn từ giỏ hàng của mình |
| **UC04** | Xác nhận đơn hàng | Phục vụ chấp nhận đơn hàng của khách hàng, có thể từ chối nếu cần thiết |
| **UC05** | Thanh toán | Khách hàng lựa chọn phương thức thanh toán và thực hiện thanh toán |
| **UC06** | Theo dõi đơn hàng | Sau khi thanh toán, khách hàng theo dõi trạng thái của đơn hàng |
| **UC07** | Quản lý và thay đổi trạng thái đơn hàng | Nhân viên nhận order thức ăn và theo dõi cập nhật đơn hàng cho khách |
| **UC08** | Đánh giá | Khách hàng đánh giá, phản hồi về chế độ phục vụ, trang web, độ hài lòng về món ăn, nhà hàng, nhân viên |
| **UC09** | Xem thông tin nhà hàng | Khách hàng xem thông tin liên quan đến nhà hàng, chủ sở hữu, nhân viên, các liên kết, chứng nhận, các chi nhánh |
| **UC10** | Login cho khách hàng | Customer đăng nhập vào hệ thống |
| **UC11** | Login cho nhân viên | Restaurant Owner, Clerk và Chef đăng nhập vào hệ thống |
| **UC12** | Account management | Restaurant Owner, Clerk, Chef quản lý tài khoản: xem thông tin tài khoản, thay đổi thông tin tài khoản, thay đổi mật khẩu hoặc đăng xuất |
| **UC13** | Quản lý tài khoản nhân viên | Restaurant Owner xem và quản lý thông tin của các Employee (Clerk, Chef) |
| **UC14** | Xem lịch sử đơn hàng | Restaurant Owner, Clerk xem thông tin lịch sử đơn hàng của nhà hàng |
| **UC15** | Edit/Add/Remove menu | Restaurant Owner chỉnh sửa, thêm và xóa menu |
| **UC16** | Menu design management | Chủ cửa hàng quản lý định dạng menu sẽ hiển thị cho khách hàng |
| **UC17** | Quản lý QR code | Chủ cửa hàng quản lý QR code: Tạo QR code, in QR code, xem QR code |
| **UC18** | Quản lý thông tin giới thiệu | Restaurant Owner quản lý thông tin của nhà hàng để hiển thị cho khách hàng xem qua |
| **UC19** | Quản lý đánh giá khách hàng | Restaurant Owner, Clerk xem, xóa đánh giá của khách hàng |

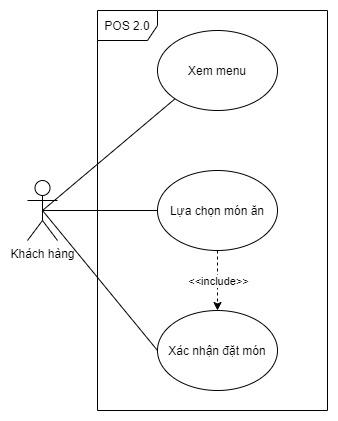
### **1.2.2 Non-Functional requirements**

* Hệ thống khởi động phải thực hiện hoàn toàn trong vòng chưa đầy 120s.
* Hệ thống xử lý và hoạt động tốt tối thiểu 25 đơn hàng cùng một thời điểm.
* Hệ thống xử lý truy cập website, hiển thị dữ liệu menu, phản hồi yêu cầu, chuyển trang không quá 2s.
* Hệ thống xử lý thông báo/hiển thị trạng thái đơn hàng cho người dùng từ yêu cầu thay đổi của phục vụ, đầu bếp là không quá 5s.
* Khi người dùng thực hiện thanh toán online, hệ thống không được phép lưu trữ thông tin tài khoản ngân hàng, mã số thẻ của người dùng.
* Khi người dùng truy cập hệ thống thông qua quét mã QR, hệ thống chỉ hiển thị trang menu của hệ thống, không gợi ý hay truy cập đến trang đăng nhập của hệ thống (phần của nhân viên, chủ nhà hàng).

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

### **1.3.1 Đặt món (Tâm)**

**1.3.1.a Use case diagram**



**1.3.1.b Scenario**

* Xem menu

| **Use-Case ID** | UC01 |
| --- | --- |
| **Use-Case Name** | Xem menu |
| **Actor** | Khách hàng |
| **Description** | Khách hàng lướt xem menu, bao gồm danh sách các món ăn theo chủ đề của nhà hàng |
| **Trigger** | Khách hàng bật camera quét mã QR trên bàn ăn của nhà hàng |
| **Preconditions** | Khách hàng đang ngồi tại bàn ăn của nhà hàng hoặc đang tại mã QR mang về |
| **Normal Flow** | 1. Khách hàng nhấn vào link web được đề xuất từ việc quét mã QR trên bàn ăn  2. Hệ thống hiển thị trang menu các món ăn hiện có của nhà hàng và số bàn hiện tại đang ngồi |
| **Exception** |  |
| **Alternative flow** |  |

* Lựa chọn món ăn

| **Use-Case ID** | UC02 |
| --- | --- |
| **Use-Case Name** | Lựa chọn món ăn |
| **Actor** | Khách hàng |
| **Description** | Khách hàng lựa chọn món ăn, thêm bớt các món ăn mong muốn cho giỏ hàng của mình |
| **Trigger** | Khách hàng nhấn vào link web |
| **Preconditions** | Khách hàng đang ở trang menu của nhà hàng |
| **Normal Flow** | 1.1. Để thêm món ăn vào giỏ hàng, khách hàng lựa chọn số lượng của món ăn và nhấn vào nút “Thêm vào giỏ hàng”  1.2. Hệ thống xác nhận và thêm thông tin món ăn vào giỏ hàng  2.1. Để hủy món ăn không còn mong muốn đặt nữa, khách hàng nhất vào nút Giỏ hàng  2.2 Hệ thống hiển thị trang giỏ hàng của khách hàng  2.3 Khách hàng nhấn vào nút “Xóa” của món ăn muốn hủy  2.4 Hệ thống hiển thị Message thông báo xác nhận  2.5 Khách hàng nhấn vào nút “OK”  2.6 Hệ thống xác nhận và xóa thông tin món ăn được chọn khỏi giỏ hàng |
| **Exception** | Exception 1: Tại bước 2.3  1a. Khách hàng chưa có bất kỳ món nào trong giỏ hàng  1b. Hệ thống hiển thị “Chưa có món ăn nào” trong giỏ hàng |
| **Alternative flow** |  |

* Xác nhận đặt món ăn

| **Use-Case ID** | UC03 |
| --- | --- |
| **Use-Case Name** | Xác nhận đặt món ăn |
| **Actor** | Khách hàng |
| **Description** | Khách hàng tiến hành đặt món ăn từ giỏ hàng của mình |
| **Trigger** | Khách hàng nhấn vào giỏ hàng |
| **Preconditions** | Khách hàng đang ở trang giỏ hàng |
| **Normal Flow** | 1. Khách hàng xem thông tin, giá các món ăn trong giỏ hàng, thêm thông tin số người tại bàn ăn, hình thức thanh toán (online/tiền mặt) và xem xét lựa chọn option mang về  2. Khách hàng nhấn vào nút “Đặt hàng”  3. Hệ thống xác nhận và gửi đơn hàng đến phục vụ, hệ thống cập nhật trạng thái giỏ hàng là “Đang chờ chấp nhận” cho khách hàng |
| **Exception** | Exception 1: Tại bước 1  1a. Khách hàng chưa có bất kỳ món nào trong giỏ hàng  1b. Hệ thống hiển thị “Chưa có món ăn nào” trong giỏ hàng và khách hàng không thể nhấn vào nút “Đặt hàng” được |
| **Alternative flow** | Alternative 1: Tại bước 3  1a. Để hủy đơn hàng, khách hàng nhấn vào nút “Hủy đơn hàng”  1b. Hệ thống xác nhận và hủy đơn hàng đến phía phục vụ |

### **1.3.2 Xác nhận đơn hàng (Tâm)**

**1.3.2.a Use case diagram**

**1.3.2.b Scenario:**

| **Use-Case ID** | UC04 |
| --- | --- |
| **Use-Case Name** | Xác nhận đơn hàng |
| **Actor** | Phục vụ |
| **Description** | Phục vụ chấp nhận đơn hàng của khách hàng, có thể từ chối nếu cần thiết |
| **Trigger** | Phục vụ nhấn vào “Danh sách đơn hàng” |
| **Preconditions** | Phục vụ đã đăng nhập thành công vào hệ thống và đang ở trang bất kỳ |
| **Normal Flow** | 1. Hệ thống hiển thị danh sách các đơn hàng đang có, các dòng đơn hàng bao gồm số (id) bàn ăn, tổng tiền đơn hàng, tên tài khoản khách hàng, hình thức thanh toán  2. Phục vụ nhấn vào nút “Chấp nhận” trên dòng đơn hàng  3. Hệ thống xác nhận và thông báo cho khách hàng |
| **Exception** | Exception 1: Tại bước 1  1a. Chưa có bất kỳ đơn hàng nào  1b. Hệ thống hiển thị “Chưa có đơn hàng nào” |
| **Alternative flow** | Alternative 1: Tại bước 2  1a. Phục vụ nhấn vào dòng đơn hàng để xem chi tiết  1b. Hệ thống hiển thị thêm chi tiết các món ăn và các thông tin khác của đơn hàng đó  1c. Phục vụ xem xét và nhấn vào nút “Chấp nhận” trên dòng đơn hàng  1d. Hệ thống xác nhận và thông báo cho khách hàng  Alternative 2: Tại bước 2  2a. Phục vụ nhấn vào nút “Từ chối” trên dòng đơn hàng nếu xảy ra vấn đề đặc biệt (hết món đột xuất, đơn hàng phá, …)  2b. Hệ thống hiển thị TextField để phục vụ nhập lý do từ chối  2c. Phục vụ điền lý do từ chối và nhấn “Xác nhận”  2d. Hệ thống xác nhận, thông báo cho khách hàng và hủy đơn hàng |

### **1.3.3 Thanh toán online (Thiện)**

| **Use-Case ID** | UC05 |
| --- | --- |
| **Use-Case Name** | Thanh toán |
| **Actor** | Khách hàng |
| **Description** | Khách hàng lựa chọn phương thức thanh toán và thực hiện thanh toán |
| **Trigger** | Khách hàng nhấn vào nút “Thanh toán” |
| **Preconditions** | Khách hàng đang ở trang giỏ hàng và đã được phục vụ xác nhận |
| **Normal Flow** | 1. Hệ thống hiển thị trang thanh toán online  2. Khách hàng lựa chọn phương thức thanh toán (bằng momo, thẻ ATM hoặc thẻ tín dụng) và nhấn nút “Xác nhận”  3. Hệ thống hiển thị trang điền thông tin để thanh toán phù hợp cho phương thức thanh toán đã chọn  4. Khách hàng xác nhận thanh toán  5. Hệ thống xác nhận, lưu thông tin đơn hàng và chuyển đến trang Theo dõi đơn hàng |
| **Exception** | Exception 1: Tại bước 4  1a. Thông tin về dịch vụ thanh toán của khách hàng không đúng  1b. Hệ thống thông báo cho khách hàng và quay lại bước 1 |
| **Alternative flow** |  |

### **1.3.4 Theo dõi đơn hàng (Thiện)**

| **Use-Case ID** | UC06 |
| --- | --- |
| **Use-Case Name** | Theo dõi đơn hàng |
| **Actor** | Khách hàng |
| **Description** | Sau khi thanh toán, khách hàng theo dõi trạng thái của đơn hàng |
| **Trigger** | Khách hàng nhấn vào nút xem thông tin đơn hàng |
| **Preconditions** | Khách hàng đã thanh toán đơn hàng bất kỳ thành công |
| **Normal Flow** | 1. Khách hàng nhấn vào nút trạng thái đơn hàng  2. Hệ thống hiển thị trạng thái của các đơn hàng(chưa làm, đang làm , đã xong) và thời gian dự kiến đơn hàng sẽ xong. |
| **Exception** | Exception 1: tại bước 1  1a. Không có đơn hàng nào đã thanh toán  1b. Hệ thống hiển thị “chưa có đơn hàng đã thanh toán”. |
| **Alternative flow** |  |

### **1.3.5 Thay đổi trạng thái đơn hàng (Duy)**

| **Use-Case ID** | UC07 |
| --- | --- |
| **Use-Case Name** | Quản lý và thay đổi trạng thái đơn hàng |
| **Actor** | Nhân viên (clerk) |
| **Description** | Nhân viên nhận order thức ăn và theo dõi cập nhật đơn hàng cho khách  Phục vụ:   * Trạng thái thanh toán (Chưa thanh toán trực tiếp, Đã thanh toán online, Đã thanh toán trực tiếp) * Trạng thái đơn hàng (đang nấu, đang giao, đã giao ) |
| **Trigger** | Nhân viên chọn vào change order status |
| **Preconditions** | Khách hàng đã đặt món |
| **Normal Flow** | 1. Nhân viên chọn Order cần thay đổi trạng thái  Hệ thống hiển thị 3 trạng thái của order và id, số bàn, số khách  2. - Trường hợp KH thanh toán online, phục vụ không thể thay đổi bất kỳ giá trị nào của trạng thái đơn hàng  3. - Trường hợp KH thanh toán trực tiếp, nhân viên có thể thay đổi trạng thái 1 (Chưa thanh toán TT, Đã thanh toán TT)  4. Sau đó nhân viên lưu lại kết quả hoặc hủy nếu không lưu  5. Nhân viên theo dõi và cập nhật trạng thái 2 ( đang nấu, đang giao, đã giao ) đơn cho đơn hàng của khách  6. Hệ thống ghi nhận kết quả cập nhật |
| **Exception** | Tại bước 1:  Nếu phục vụ chưa đăng nhập vào hệ thống, thông báo chưa đăng nhập  Quay lại màn hình login |
| **Alternative flow** | Alternative 1: tại bước 3  1a. Trường hợp khách hàng không thanh toán hay thanh toán bị lỗi, hệ thống sẽ thông báo thanh toán thất bại  1b. Hệ thống sẽ hủy đơn hàng vừa đặt  1c. Quay trở về trang menu  Alternative 2: tại bước 3  2a. Trường hợp khách hàng không thanh toán nhưng muốn giữ thông tin lịch sử đặt món có thể chọn lưu vào lịch sử đặt món  2b. Hệ thống lưu các món đã đặt và quay trở lại trang menu |

### **1.3.6 Đánh giá (Duy)**

| **Use-Case ID** | UC08 |
| --- | --- |
| **Use-Case Name** | Đánh giá |
| **Actor** | Khách hàng |
| **Description** | Khách hàng đánh giá, phản hồi về chế độ phục vụ, trang web, độ hài lòng về món ăn, nhà hàng, nhân viên |
| **Trigger** | Khách hàng nhấn vào nút “Đánh giá” |
| **Preconditions** | Khách hàng đang ở trang đánh giá của web |
| **Normal Flow** | 1. Khách hàng nhập bình luận thông tin cá nhân tùy chọn của mình, lựa chọn rating và nhấn nút “Đăng”  2. Hệ thống xác nhận, lưu thông tin đánh giá của khách hàng |
| **Exception** |  |
| **Alternative flow** |  |

### **1.3.7 Xem thông tin nhà hàng (Duy)**

| **Use-Case ID** | UC09 |
| --- | --- |
| **Use-Case Name** | Xem thông tin nhà hàng |
| **Actor** | Khách hàng |
| **Description** | Khách hàng xem thông tin liên quan đến nhà hàng, chủ sở hữu, nhân viên, các liên kết, chứng nhận, các chi nhánh |
| **Trigger** | Khách hàng nhấn vào nút “Giới thiệu” |
| **Preconditions** | Khách hàng đang ở trang thông tin của nhà hàng |
| **Normal Flow** | 1. Hệ thống hiển thị chi tiết các thông tin của nhà hàng được viết bởi chủ cửa hàng |
| **Exception** |  |
| **Alternative flow** |  |

### **1.3.8 Quản lý tài khoản (Thân)**

1.3.7.a Use-case diagram

1.3.7.b Scenario

* Hệ thống đăng nhập cho khách hàng

| **Use-Case ID** | UC10 |
| --- | --- |
| **Use-Case Name** | Login cho khách hàng (Customer) |
| **Actor** | Customer |
| **Description** | Customer đăng nhập vào hệ thống |
| **Trigger** | Customer nhấn nút đăng nhập |
| **Preconditions** | Customer đã nhấn vào nút đặt order |
| **Normal Flow** | 1. Customer chọn nút “Login”  2. Hệ thống hiển thị form đăng nhập bao gồm 3 cách: bằng số điện thoại (mặc định), bằng tài khoản Google và bằng tài khoản Facebook  3. Người dùng nhập thông tin số điện thoại  4. Hệ thống kiểm tra thông tin, gửi mã xác thực cho khách hàng  5. Người dùng nhập mã xác thực nhận được  6. Hệ thống xác nhận thông tin chính xác và hiển thị form để khách hàng điền thêm số người, …  7. Khách hàng điền thông tin và bấm vào nút xác nhận  8. Hệ thống thông báo đặt hàng thành công và quay về trang chủ |
| **Exception** | Exception 1: Tại bước 5  1a. Khách hàng nhập mã xác thực không chính xác  1.b Hệ thống gửi thông báo nhập lại và quay về màn hình login |
| **Alternative Flow** | Alternative 1: Tại bước 2  1a. Khách hàng chọn nút “Đăng nhập bằng Google”  1b. Hệ thống hiển thị form điền email  1c. Khách hàng điền email và bấm gửi mã xác thực  1d. Hệ thống email chứa mã xác thực vào tài khoản Google của khách hàng  Alternative 2: Tại bước 2  2a. Khách hàng chọn nút “Đăng nhập bằng Facebook”  2b. Hệ thống mở ra liên kết tới Facebook  3c. Khách hàng đăng nhập Facebook của mình và cho phép hệ thống xác thực tài khoản  3d. Hệ thống xác nhận thông tin và chuyển sang bước 6 |

* Hệ thống đăng nhập cho nhân viên nhà hàng

| **Use-Case ID** | UC11 |
| --- | --- |
| **Use-Case Name** | Login cho nhân viên |
| **Actor** | Employee (E) bao gồm Restaurant Owner, Clerk và Chef |
| **Description** | E đăng nhập vào hệ thống |
| **Trigger** | E nhấn nút đăng nhập |
| **Preconditions** | E đã vào trang web riêng dành cho nhân viên của hệ thống |
| **Normal Flow** | 1. E chọn nút “Login”  2. Hệ thống hiển thị form đăng nhập  3. Người dùng nhập thông tin tài khoản và mật khẩu  4. Hệ thống kiểm tra thông tin đăng nhập  5. Người dùng đăng nhập thành công và được chuyển về trang chủ |
| **Exception** | Exception 1: Tại bước 3  1a. E nhập tài khoản hoặc mật khẩu không chính xác  1.b Hệ thống gửi thông báo nhập lại và quay về màn hình login |
| **Alternative Flow** | Alternative 1: Tại bước 3  1a. E chọn nút “Quên mật khẩu”  1b. Hệ thống hiển thị form quên mật khẩu  1c. E điền các thông bảo mật  1d. Hệ thống kiểm tra thông tin bảo mật hợp lệ  A. Thông tin bảo mật chính xác  1e. Hệ thống hiển thị form thay đổi mật khẩu  1f. E nhập mật khẩu mới và nhập lại mật khẩu mới  1g. Hệ thống lưu lại thay đổi và quay về trang đăng nhập  B. Thông tin bảo mật không chính xác  1h. Hệ thống gửi thông báo không hợp lệ và quay về trang đăng nhập |

* Quản lý các thông tin cơ bản

| **Use-Case ID** | UC12 |
| --- | --- |
| **Use-Case Name** | Account management |
| **Actor** | Employee (E) bao gồm Restaurant Owner, Clerk, Chef |
| **Description** | E quản lý tài khoản: xem thông tin tài khoản, thay đổi thông tin tài khoản, thay đổi mật khẩu hoặc đăng xuất |
| **Trigger** | E nhấn nút hình ảnh tài khoản |
| **Preconditions** | E đang ở trang quản lý tài khoản người dùng |
| **Normal Flow** | 1. E chọn nút “Account Info”  2. Hệ thống hiển thị thông tin tài khoản của người dùng |
| **Exception** | Exception 1: Tại bước 1  1a. E chưa đăng nhập, E nhập thông tin đăng nhập và nhấn vào nút “Login”  1.b Hệ thống xác nhận và hiển thị trang thông tin tài khoản người dùng |
| **Alternative Flow** | Alternative 1: Tại bước 1  1a. E chọn nút “Change Password”  1b. Hệ thống hiển thị form thay đổi mật khẩu  1c. E điền các thông tin muốn thay đổi mật khẩu (mật khẩu cũ, mật khẩu mới, nhập lại mật khẩu mới) và nhấn nút “Update”  1d. Hệ thống xác nhận và hiển thị Message “Cập nhật mật khẩu thành công”  Alternative 2: Tại bước 1  2a. E chọn nút “Logout”  2b. Hệ thống đăng xuất tài khoản và hiển thị trang login  Alternative 3: Tại bước 1  3a. E chọn nút “Account Info”  3b. Hệ thống hiển thị form thay đổi thông tin tài khoản  3c. E điền các thông tin muốn thay và nhấn nút “Update”  3d. Hệ thống xác nhận và hiển thị Message “Cập nhật thông tin thành công” |

### **1.3.9 Quản lý tài khoản nhân viên (Thân)**

Scenario

| **Use-Case ID** | UC13 |
| --- | --- |
| **Use-Case Name** | Quản lý tài khoản nhân viên |
| **Actor** | Restaurant Owner (RO) |
| **Description** | RO xem và quản lý thông tin của các Employee (Clerk, Chef) |
| **Trigger** | RO nhấn nút “Thông tin nhân viên” |
| **Preconditions** | RO đã đăng nhập vào trang web riêng của hệ thống |
| **Normal Flow** | 1. RO nhấn vào nút “Thông tin nhân viên”  2. Hệ thống hiển thị thông tin nhân viên  3. RO nhấn vào nút chỉnh sửa  4. Hệ thống hiển thị form chỉnh sửa thông tin nhân viên  5. RO nhập thông tin cần sửa vào form và nhấn Update  6. Hệ thống lưu lại thay đổi và trở về trang thông tin nhân viên |
| **Exception** | Exception 1: Tại bước 1  1a. RO chưa đăng nhập  1.b Hệ thống quay về màn hình login |
| **Alternative Flow** | Alternative 1: Tại bước 3  1a. RO nhấn vào nút “Thêm nhân viên”  1b. Hệ thống hiển thị form yêu cầu nhập thông tin cho nhân viên mới  1c. RO nhập thông tin cơ bản cho nhân viên mới và nhấn Thêm  1d. Hệ thống lưu lại thông tin và quay về trang thông tin nhân viên  Alternative 2: Tại bước 3  2a. RO nhấn vào nút “Xóa nhân viên”  2b. Hệ thống hiển thị nút “Xóa” bên cạnh thông tin mỗi nhân viên  2c. RO nhấn vào nút Xóa của nhân viên  2d. Hệ thống gửi thông báo xác nhận cho RO  2e. RO bấm vào nút Xác nhận  2f. Hệ thống xóa nhân viên đã chọn và quay lại trang thông tin nhân viên |

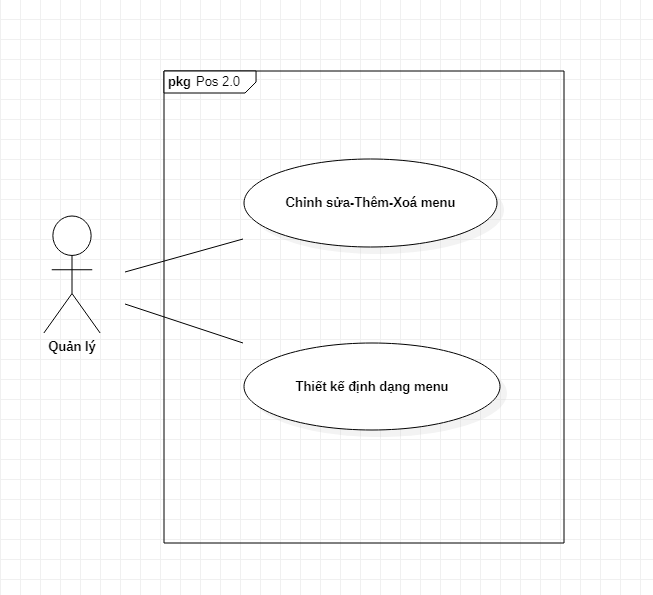
### **1.3.10 Xem thông tin lịch sử đơn hàng (Thân)**

Scenario

| **Use-Case ID** | UC14 |
| --- | --- |
| **Use-Case Name** | Xem lịch sử đơn hàng |
| **Actor** | Employee (E) bao gồm Restaurant Owner, Clerk |
| **Description** | E xem thông tin lịch sử đơn hàng của nhà hàng |
| **Trigger** | E nhấn nút “Lịch sử đơn hàng” |
| **Preconditions** | RO đã đăng nhập vào trang web riêng của hệ thống |
| **Normal Flow** | 1. E nhấn vào nút “Lịch sử”  2. Hệ thống hiển thị các đơn hàng theo thứ tự thời gian  3. E chọn vào đơn hàng  4. Hệ thống hiển thị thông tin chi tiết của đơn hàng (bao gồm món ăn, giá mỗi món, …) |
| **Exception** | Exception 1: Tại bước 1  1a. RO chưa đăng nhập  1.b Hệ thống quay về màn hình login |
| **Alternative Flow** |  |

### **1.3.11 Quản lý menu (Phương)**

1.3.10.a Use case



1.3.10.b Scenario

* Chỉnh sửa, thêm, xóa thông tin menu

| **Use-Case ID** | UC15 | |
| --- | --- | --- |
| **Use-Case Name** | Edit/Add/Remove menu | |
| **Actor** | Chủ cửa hàng (RO) | |
| **Description** | RO chỉnh sửa, thêm và xóa menu | |
| **Trigger** | RO đã nhấn vào nút “Chỉnh sửa menu” | |
| **Preconditions** | RO đang ở trang danh sách các menu | |
| **Normal Flow** | 1.1 RO nhấn vào tên các menu  1.2 Hệ thống hiển thị trang chỉnh sửa menu  1.3 RO lựa chọn thay đổi thông tin từ các trường thông tin của menu và RO nhấn vào nút “Lưu”  1.4 Hệ thống lưu lại thay đổi của menu  2.1 RO nhấn vào nút “Thêm menu”  2.2 Hệ thống hiển thị trang tạo menu mới “Tạo menu”  2.3 RO điền thông tin cần thiết vào các trường thông tin của menu và RO nhấn vào nút “Lưu”  2.4 Hệ thống lưu lại thông tin menu vào dữ liệu  3.1 RO nhấn vào nút “Xóa menu”  3.2 Hệ thống hiển thị Message “Bạn có chắc chắn xóa menu #A không?”  3.3 RO nhấn vào nút “Xác nhận”  3.4 Hệ thống xác nhận và xóa menu #A khỏi dữ liệu | |
| **Exception** | Exception 1: Tại bước 1.1 và 3.1  1a. Không có bất kỳ menu nào được tạo, hệ thống sẽ hiển thị “Empty” | |
| **Alternative Flow** |  | |

* Thiết kế định dạng menu

| **Use-Case ID** | UC16 | |
| --- | --- | --- |
| **Use-Case Name** | Menu design management | |
| **Actor** | Chủ cửa hàng (RO) | |
| **Description** | Chủ cửa hàng quản lý định dạng menu sẽ hiển thị cho khách hàng | |
| **Trigger** | RO đã nhấn vào nút “Giao diện menu” | |
| **Preconditions** | RO đang ở trang thiết kế menu | |
| **Normal Flow** | 1. RO lựa chọn layout và theme cho menu theo ý muốn và nhấn nút “Update”  2. Hệ thống xác nhận và hiển thị giả lập trên điện thoại ở màn hình theo kết quả đã được RO lựa chọn | |
| **Exception** |  | |
| **Alternative Flow** |  | |

### **1.3.12 Quản lý QR code (Phương)**

#### 

| **Use-Case ID** | UC17 | |
| --- | --- | --- |
| **Use-Case Name** | Quản lý QR code | |
| **Actor** | Restaurant Owner (RO) | |
| **Description** | Chủ cửa hàng quản lý QR code: Tạo QR code, in QR code, xem QR code | |
| **Trigger** | RO nhấn vào nút “QR code” trên thanh navbar | |
| **Preconditions** | RO đang ở trang quản lý QR code | |
| **Normal Flow** | 1.1 Để xem QR code RO lựa chọn nhấn vào nút trên danh sách mã  1.2 Hệ thống hiển thị thông tin QR code (bao gồm tên QR code, QR code và nút “Remove QR code”)  2.1 Để tạo QR code RO nhấn vào nút “Add new QR”  2.2 Hệ thống hiển thị form tạo QR mới  2.3 RO nhập tên QR code cần tạo và nhấn OK  2.4 Hệ thống xác nhận và tạo một nút QR mới  3.1 Để in QR code RO nhấn vào link in theo kích cỡ mong muốn  3.2 Hệ thống mở liên kết đến phần in của máy tính  3.3 RO nhấn “Print”  3.4 Hệ thống xác nhận và in QR code  4.1 Để xóa QR code RO lựa chọn nhấn vào nút trên danh sách mã  4.2 Hệ thống hiển thị thông tin QR code (bao gồm tên QR code, QR code và nút “Remove QR code”)  4.3 Nhấn nút “Remove QR code”  4.4 Hệ thống xác nhận, xóa thông tin mã QR code đã chọn và hiển thị trang chủ QR code | |
| **Exception** | Exception 1: Tại bước 1.1  1a. Chưa có bất kỳ QR code nào được tạo, hệ thống hiển thị “Chưa có bất kỳ QR code nào” | |
| **Alternative Flow** |  | |

#### 

### **1.3.13 Quản lý thông tin giới thiệu (Group work)**

| **Use-Case ID** | UC18 |
| --- | --- |
| **Use-Case Name** | Đánh giá |
| **Actor** | Chủ cửa hàng (RO) |
| **Description** | RO quản lý thông tin của nhà hàng để hiển thị cho khách hàng xem qua |
| **Trigger** | RO nhấn vào nút “Quản lý thông tin” |
| **Preconditions** | RO đang ở trang quản lý thông tin |
| **Normal Flow** | 1. RO thay đổi, ghi các thông tin cần thiết muốn hiển thị cho khách hàng bao gồm chủ sở hữu, nhân viên, các liên kết, chứng nhận, các chi nhánh, …  2. RO nhấn vào nút lưu thông tin  2. Hệ thống xác nhận, lưu thông tin của nhà hàng |
| **Exception** |  |
| **Alternative flow** |  |

### **1.3.14 Quản lý đánh giá khách hàng (Group work)**

| **Use-Case ID** | UC19 |
| --- | --- |
| **Use-Case Name** | Quản lý đánh giá |
| **Actor** | Restaurant Owner, Clerk (RO/C) |
| **Description** | RO/C xem, xóa đánh giá của khách hàng |
| **Trigger** | RO/C nhấn vào nút quản lý đánh giá |
| **Preconditions** | RO/C đang ở trang web quản lý đánh giá |
| **Normal Flow** | 1. RO/C nhấn vào nút quản lý đánh giá  2. Hệ thống hiển thị số lượng đánh giá (cho menu) trong ngày  3. RO/C nhấn vào món ăn để xem chi tiết  4. Hệ thống hiển thị đánh giá theo trình tự thời gian của món ăn  5. RO/C nhấn vào nút xóa đánh giá bên cạnh đánh giá của khách hàng  6. Hệ thống xóa đánh giá đã chọn |
| **Exception** | Exception 1: Tại bước 3:  3a. Nếu món ăn chưa có đánh giá, hệ thống thông báo chưa có đánh giá nào |
| **Alternative flow** |  |

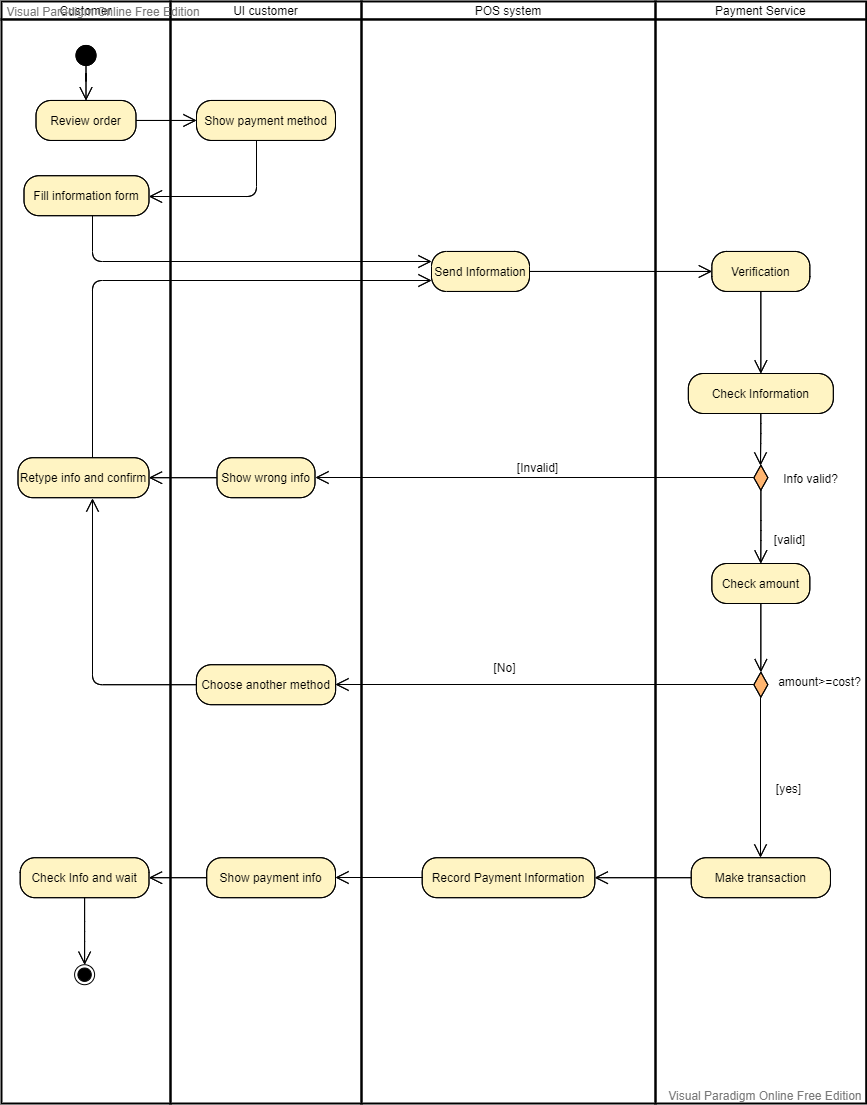
# **#2: System modeling**

## **2.1. Draw an activity diagram to capture Major (not all) functional requirements of the desired system**

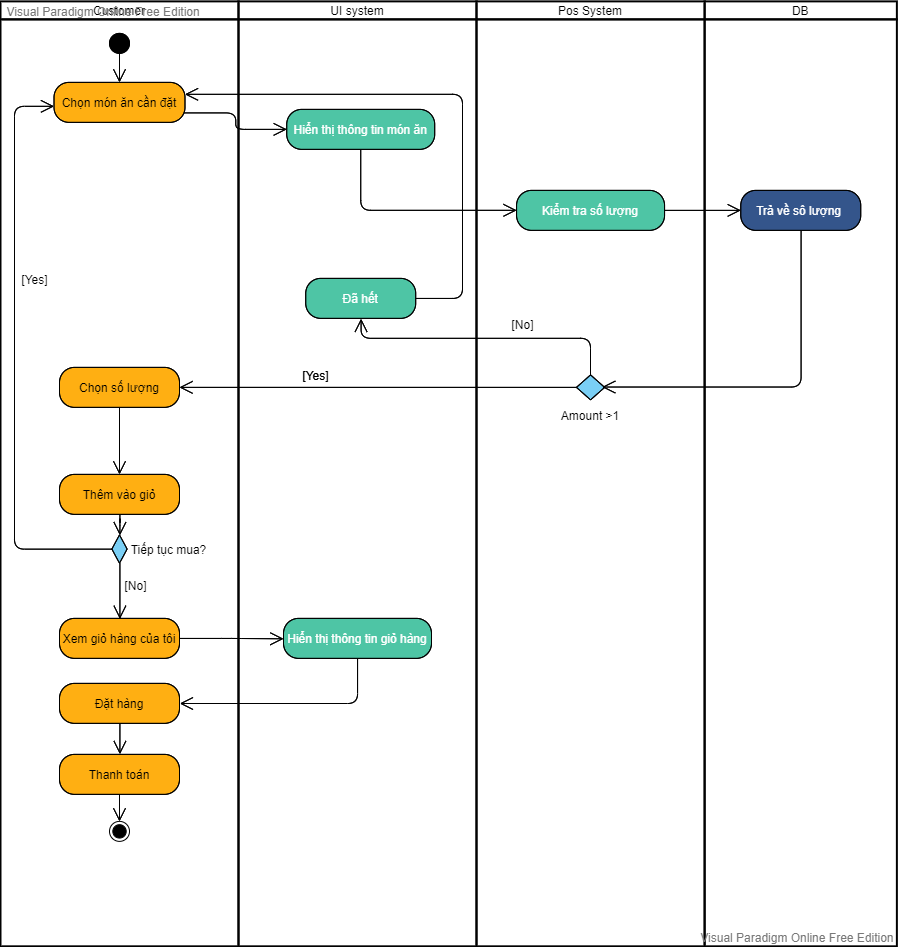
* Whole system

## 

* Payment function

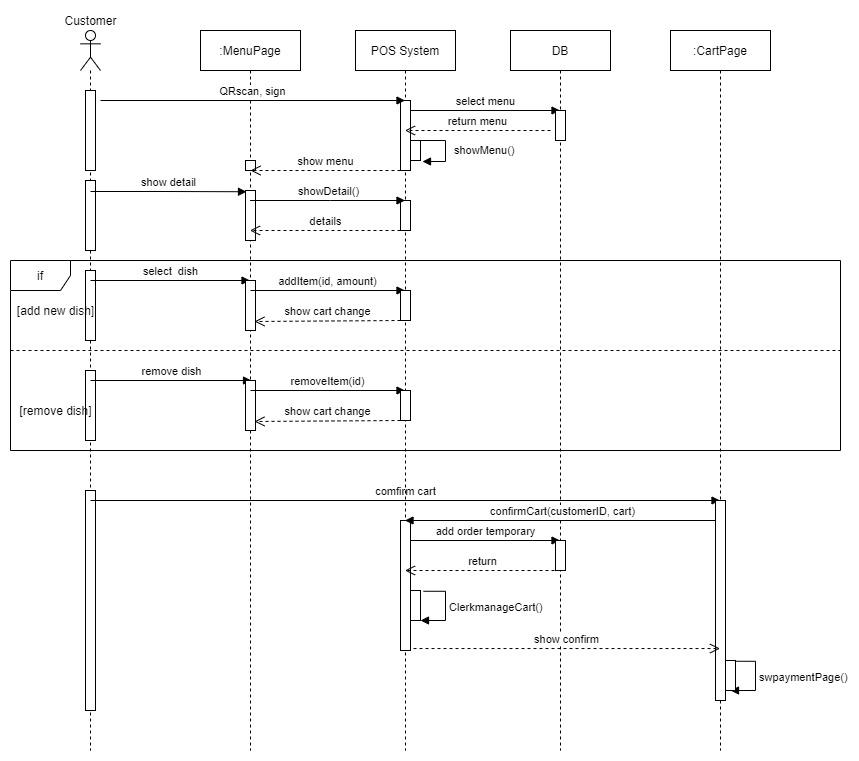


* Order Function

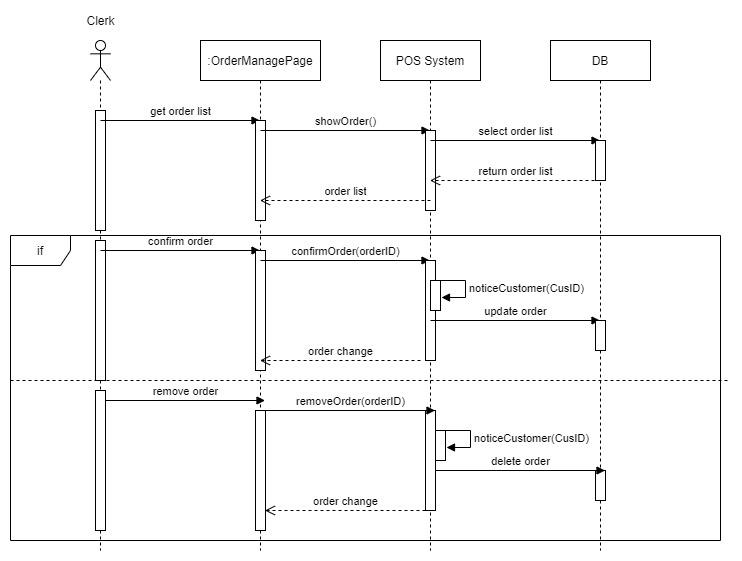


## **2.2. Draw a sequence diagram for use-case in Task 1.3.**

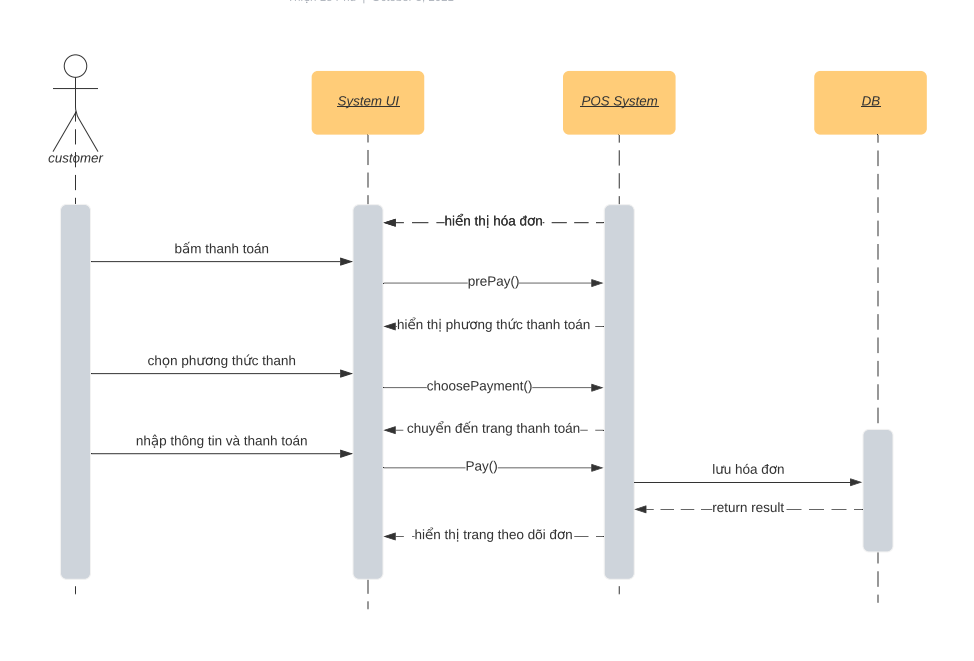
### **2.2.1 Đặt món**



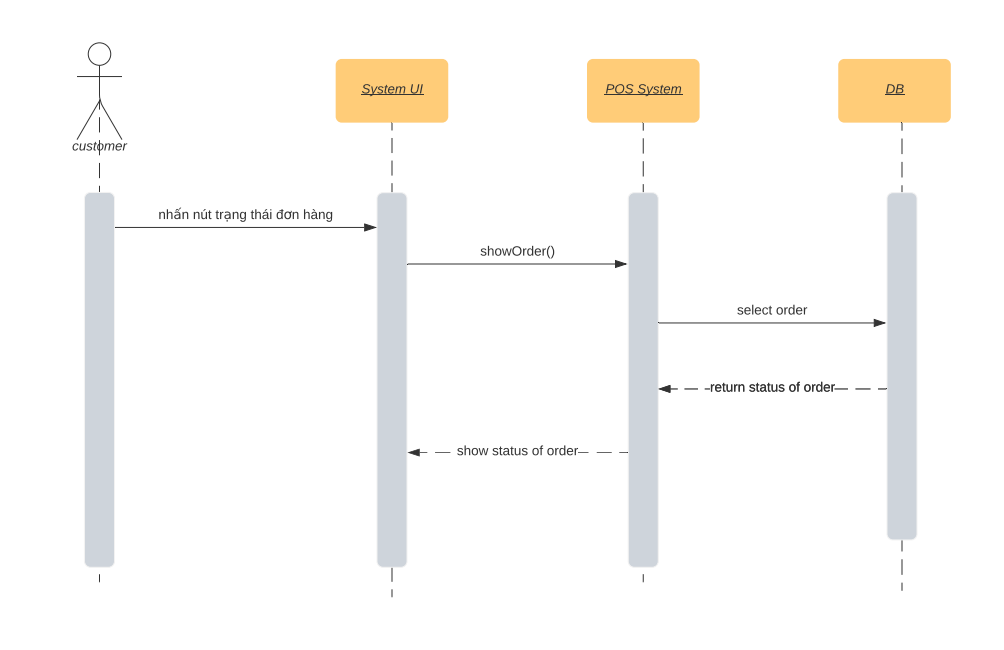
### **2.2.2 Xác nhận đơn hàng**



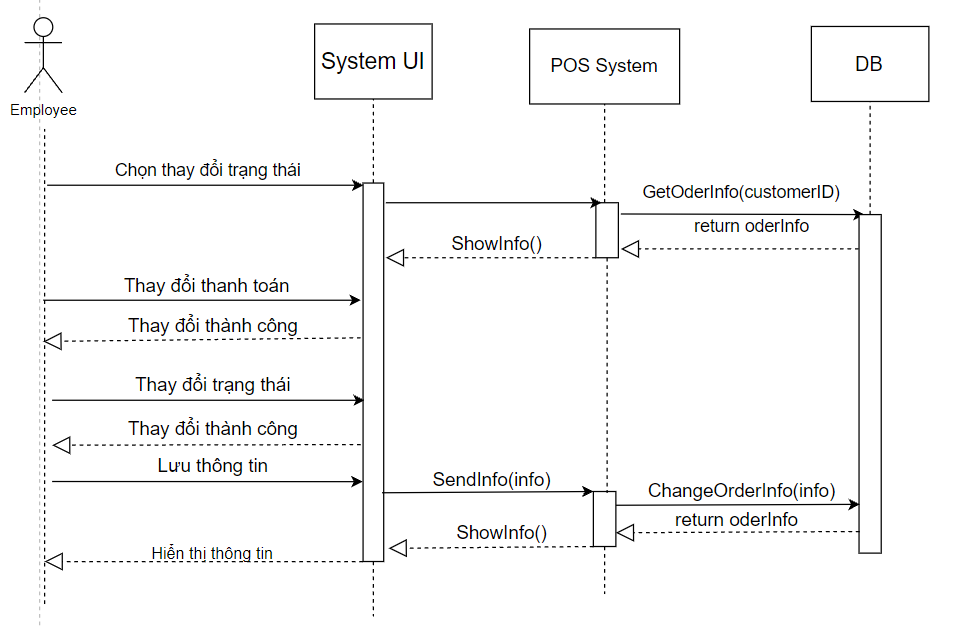
### **2.2.3 Thanh toán online**



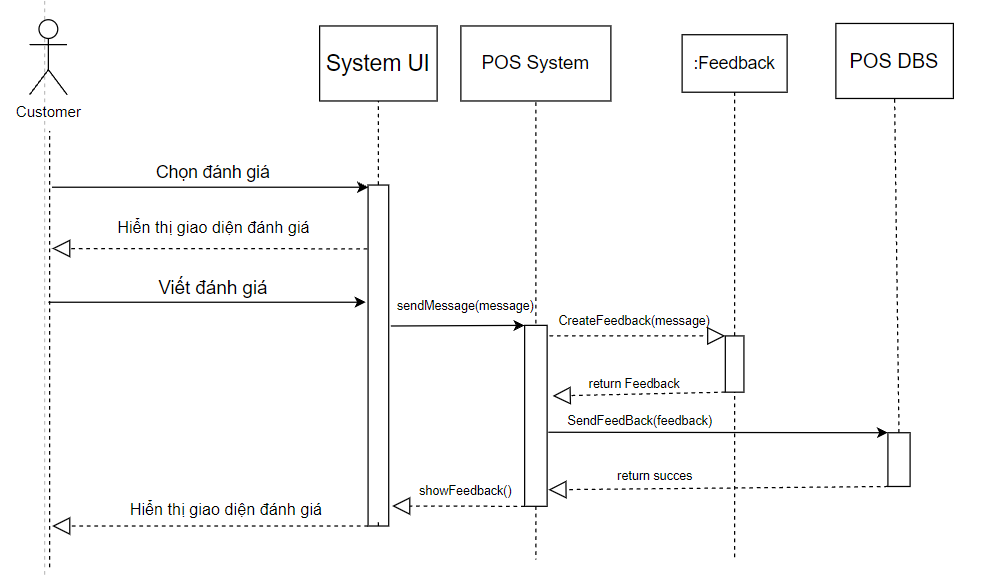
### **2.2.4 Theo dõi đơn hàng**



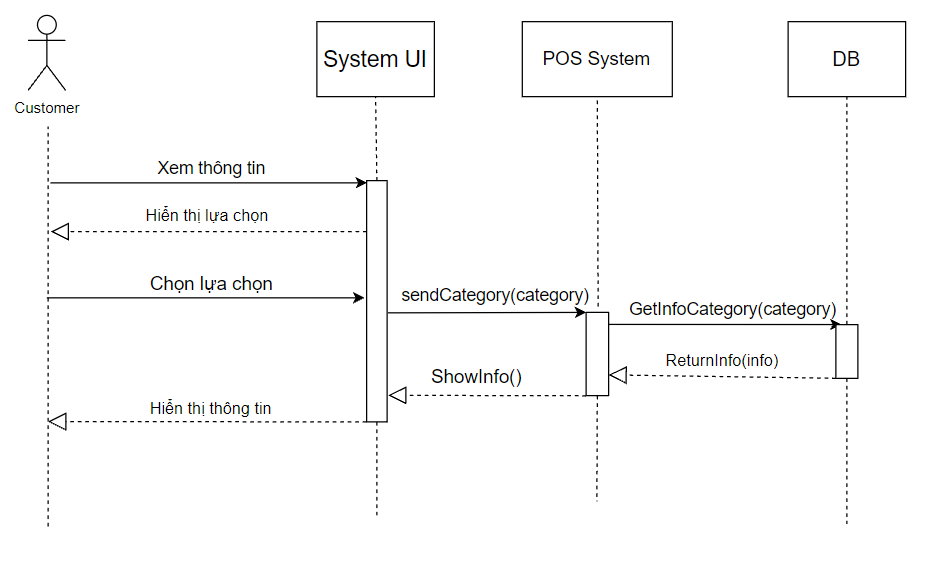
### **2.2.5 Thay đổi trạng thái đơn hàng**



### **2.2.6 Đánh giá**

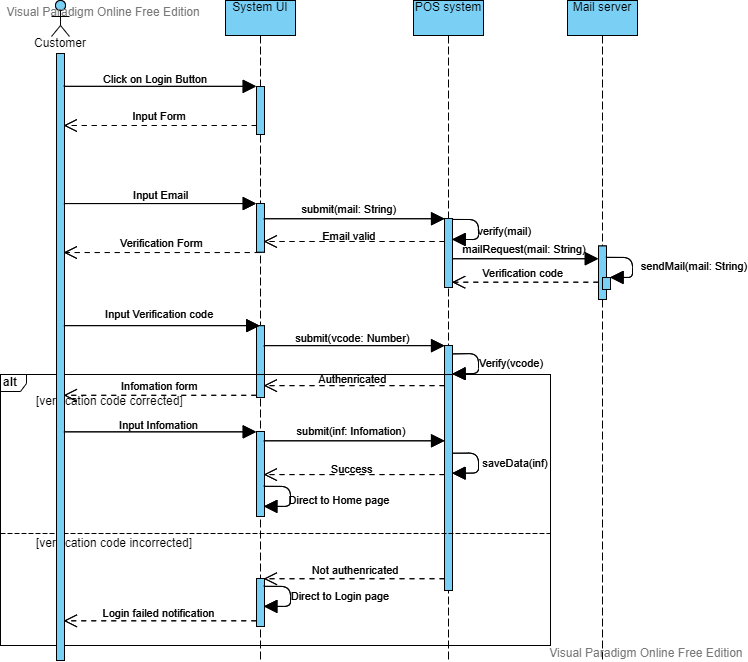


### **2.2.7 Xem thông tin nhà hàng**



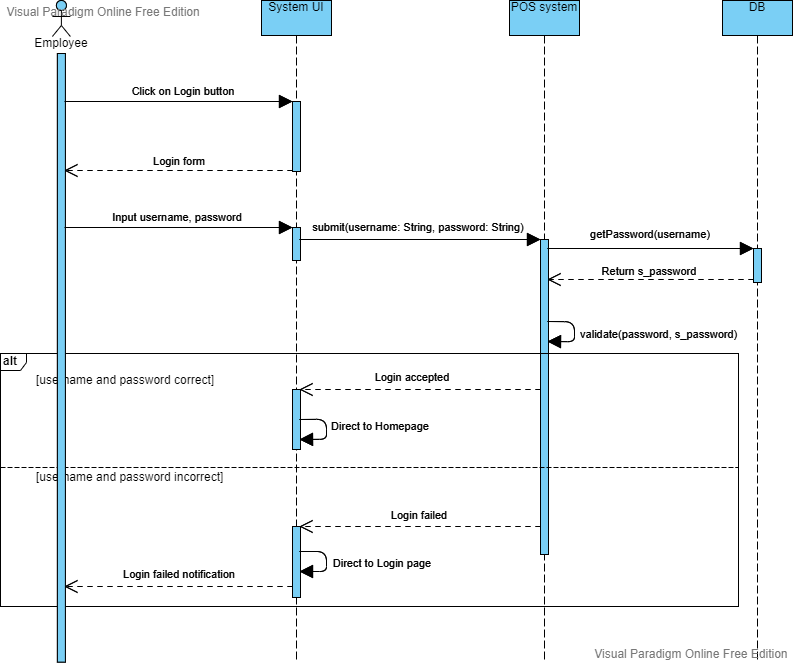
### **2.2.8 Quản lý tài khoản khách hàng**

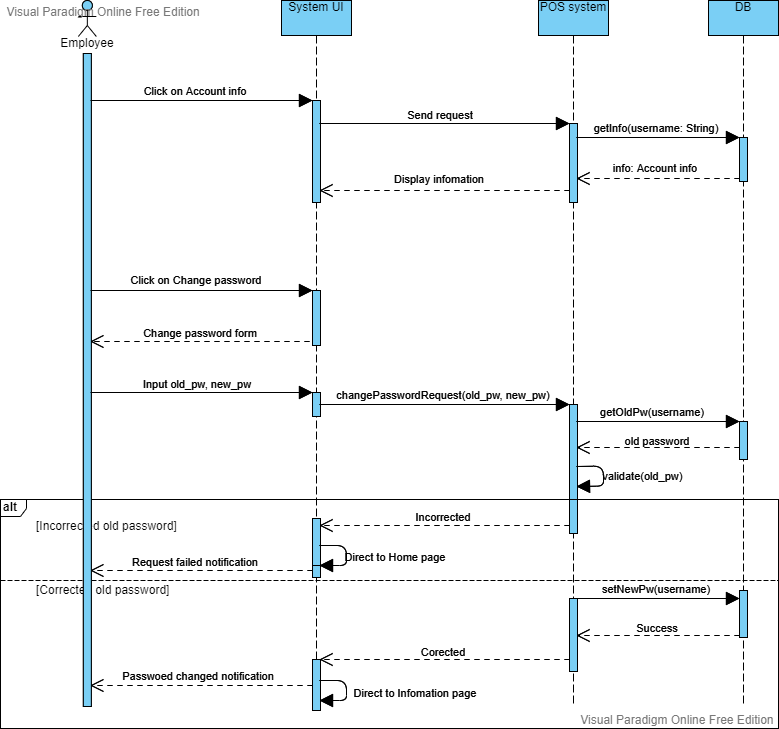
Sequence diagram cho Login của khách hàng



### **2.2.9 Quản lý tài khoản nhân viên**

Sequence diagram cho Login của nhân viên



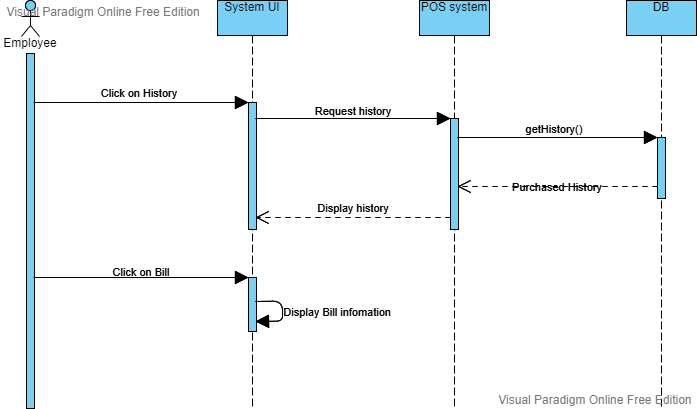
Sequence diagram cho Quản lý thông tin của Employee (gồm có Restaurant Owner, Clerk, Chef)

Sequence diagram cho Quản lý thông tin nhân viên của Restaurant Owner

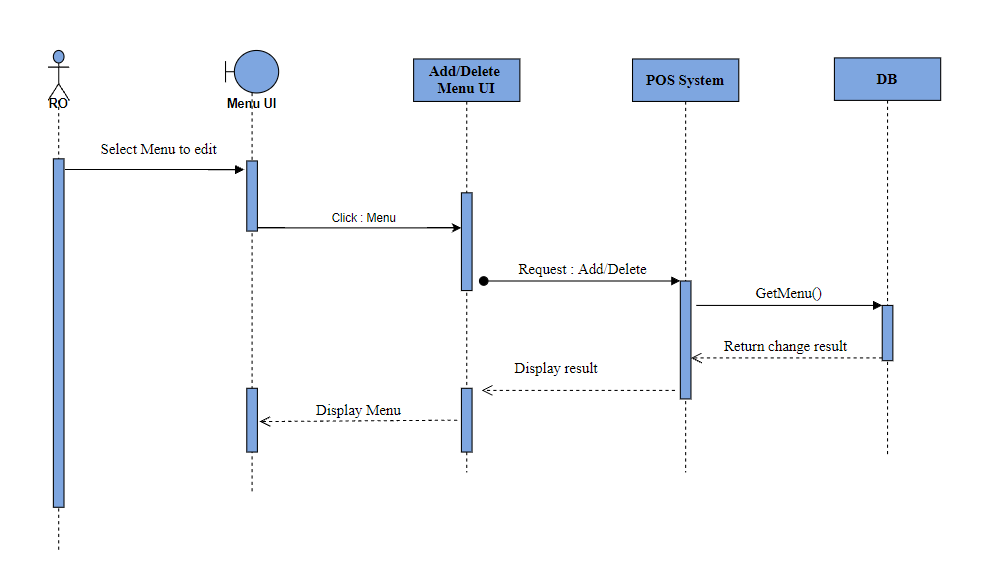
### 

### **2.2.10 Xem thông tin lịch sử đơn hàng**

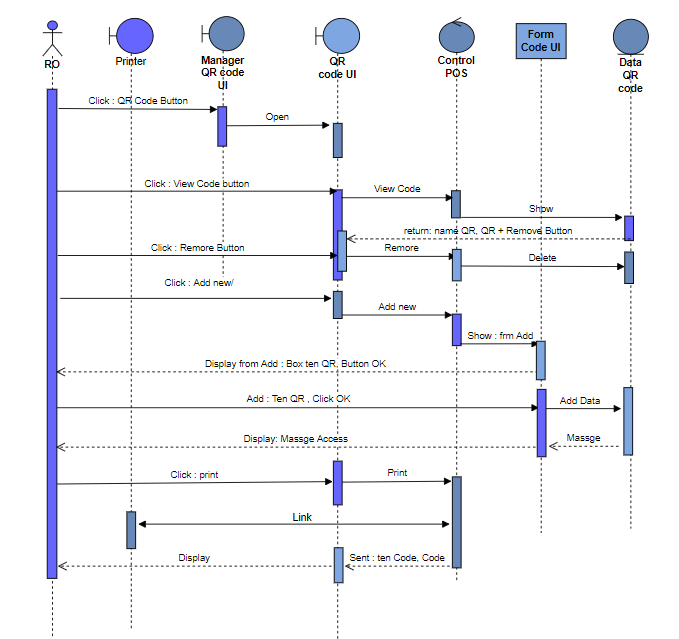
Sequence diagram cho Xem thông tin lịch sử đơn hàng



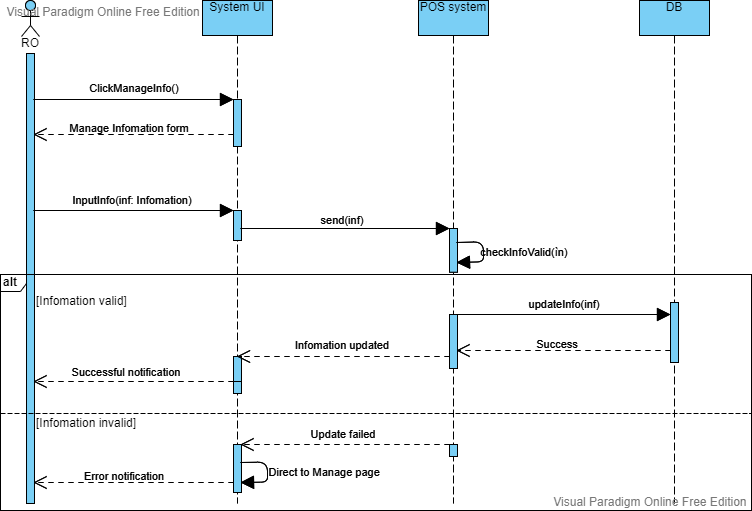
### **2.2.11 Quản lý menu**



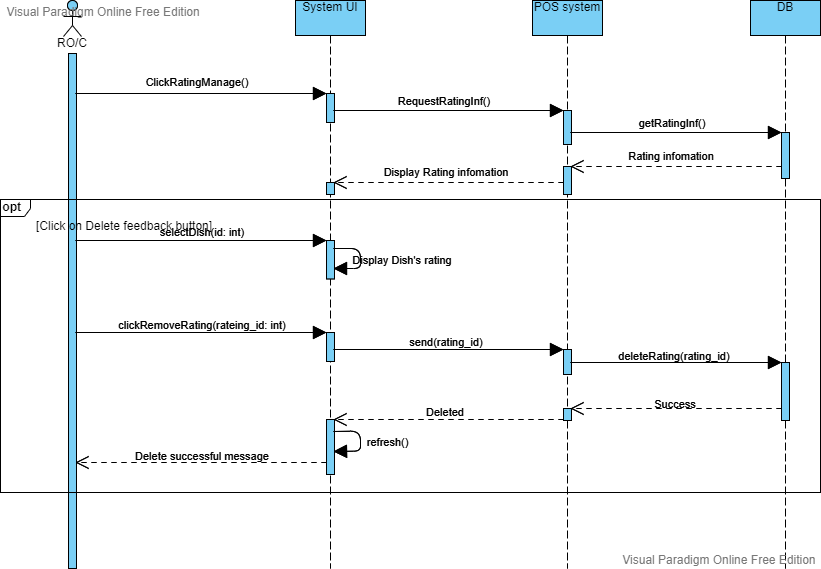
### **2.2.12 Quản lý QR code**



### **2.2.13 Quản lý thông tin giới thiệu**

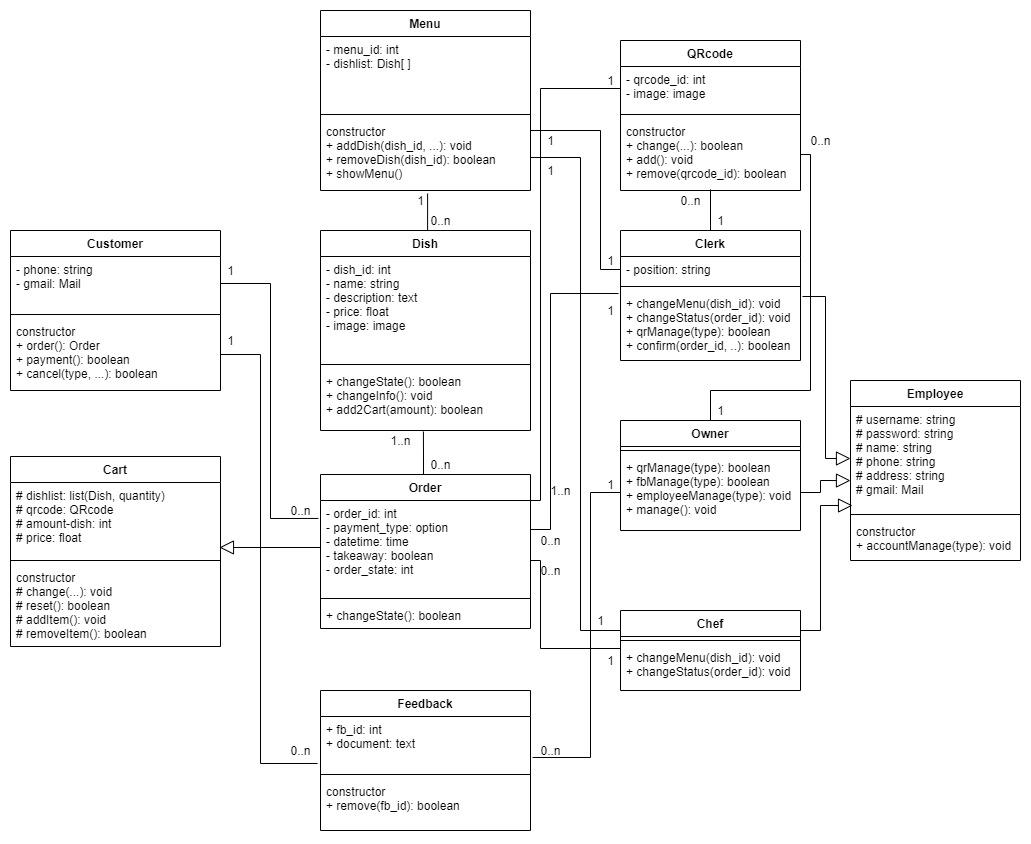


### **2.2.14 Quản lý đánh giá khách hàng**



## **2.3. Draw a class diagram**

Class diagram for whole system



# **#3: Architecture design**

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

### **3.1.1 Technology**

**3.1.1.1 NodeJS**

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

* NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh. Nếu bạn cần hỗ trợ gì về NodeJS, sẽ nhanh chóng có người hỗ trợ bạn.
* Tốc độ xử lý nhanh. Nhờ cơ chế xử lý bất đồng độ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào.
* Dễ dàng mở rộng. Nếu bạn có nhu cầu phát triển website thì tính năng dễ dàng mở rộng của NodeJS là một lợi thế cực kỳ quan trọng.
* NodeJs còn đi kèm với **Node package manager (NPM)**  là một công cụ tạo và quản lý các thư viện lập trình Javascript cho Node.js. Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework.

**3.1.1.2 ExpressJS**

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

ExpressJS sẽ giúp bạn tổ chức kiến ​​trúc backend của mình. Các nhà phát triển web thường sử dụng ExpressJS để triển khai kiến ​​trúc MVC, điều này cho phép họ viết một codebase back-end bảo trì tương đối dễ dàng.

Tổng hợp một số chức năng chính của Expressjs như sau:

* Thiết lập các lớp trung gian để trả về các HTTP request.
* Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số.

**3.1.1.3 GraphQL**

GraphQL, viết tắt của **Graph Query Language**, là một ngôn ngữ truy vấn dữ liệu cho API hiệu quả, mạnh mẽ và mang tính linh hoạt cao.

Cụ thể với GraphQL, khi client gửi yêu cầu cần dữ liệu đến server thì cấu trúc dữ liệu trả về sẽ không khô cứng mà sẽ linh hoạt thay đổi theo cách phía client yêu cầu những dữ liệu gì. Trước đó, những thông tin định dạng và định nghĩa sẽ được mô tả sẵn ở server rồi nên khi gửi yêu cầu dữ liệu lên server thì chỉ cần 1 endpoint duy nhất.

Nói cách khác, GraphQL cung cấp mô tả đầy đủ và dễ hiểu về dữ liệu trong API của bạn, cung cấp cho phía khách (client) khả năng yêu cầu chính xác những gì họ cần mà không cần dư thừa, giúp việc phát triển API dễ dàng hơn theo thời gian và cho phép các công cụ mạnh mẽ dành cho nhà phát triển.

GraphQL API có 3 phần chính :

* **Queries (Truy vấn):** Truy vấn là yêu cầu mà phía client đưa ra, cụ thể là Các câu lệnh lấy dữ liệu (tương tự method GET trong REST API)
* **Resolvers:** Resolver là thứ giúp phía server hiểu được cần làm gì/đưa ra thông tin gì, ở đâu, để phản hồi lại yêu cầu truy vấn phía trên của client.Ngoài ra, với GraphQL, schema của API và schema của cơ sở dữ liệu bạn sử dụng được tách rời. Điều này cho phép bạn sử dụng chúng như **mutation resolvers** để sửa đổi nội dung cơ sở dữ liệu của bạn.
* **Schema (Lược đồ):** Một GraphQL schema mô tả các chức năng mà phía client có thể sử dụng khi chúng kết nối với server GraphQL. Mỗi đơn vị cấu thành schema được gọi là 1 type.

**3.1.1.4 PostGresql**

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ và đối tượng (object- relational database management system) dựa trên POSTGRES, bản 4.2.[1]

* PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.
* PostgreSQL là một chương trình mã nguồn mở xây dựng trên mã nguồn ban đầu của đại học Berkeley. Nó theo chuẩn SQL99 và có nhiều đặc điểm hiện đại:
* Câu truy vấn phức hợp (complex query)
* Khóa ngoại (foreign key)
* Thủ tục sự kiện (trigger)
* Các khung nhìn (view)
* Tính toàn vẹn của các giao dịch (integrity transactions)
* Việc kiểm tra truy cập đồng thời đa phiên bản (multi version concurrency

control)

* PostgreSQL là hệ quản trị cơ sở dữ liệu hỗ trợ mạnh trong việc lưu trữ dữ liệu không gian. PostgreSQL kết hợp với module Postgis cho phép người dùng lưu trữ các lớp dữ liệu không gian. Khi sử dụng PostgreSQL, Postgis kết hợp với các phần mềm GIS hỗ trợ hiển thị, truy vấn, thống kê hoặc xử lý dữ liệu không gian.
* PostgreSQL là hệ thống quản lý cơ sở dữ liệu đầu tiên triển khai tính năng kiểm soát đồng thời nhiều phiên bản (MVCC) trước cả Oracle. Tính năng MVCC cũng tương tự với các snapshot riêng biệt trong Oracle.

**3.1.1.5 ReactJS**

React Js là một thư viện viết bằng **javascript**, dùng để **xây dựng giao diện người dùng** (UI). React được sử dụng rộng rãi và có hệ sinh thái đa dạng phong phú. React được sử dụng bởi hàng trăm công ty lớn trên thế giới, bao gồm **Netflix**, **Airbnb**, **American Express**, **Facebook**, **WhatsApp**, **eBay**, và **Instagram**. Đây là bằng chứng cho thấy công cụ này có một số lợi thế cạnh tranh so với các đối thủ khác.

* React component dễ viết hơn vì nó sử dụng **JSX**, mở rộng cú pháp tùy chọn cho JavaScript cho phép bạn kết hợp HTML với JavaScript.
* JSX là một sự pha trộn tuyệt vời của JavaScript và **HTML**. Nó làm rõ toàn bộ quá trình viết cấu trúc trang web. Ngoài ra, phần mở rộng cũng giúp render nhiều lựa chọn dễ dàng hơn.
* JSX có thể không là phần mở rộng cú pháp phổ biến nhất, nhưng nó được chứng minh là hiệu quả trong việc phát triển components đặc biệt hoặc các ứng dụng có khối lượng lớn.
* ReactJS cho phép Developer phá vỡ những cấu tạo UI phức tạp thành những component độc lập. Dev sẽ không phải lo lắng về tổng thể ứng dụng web, giờ đây Developer dễ dàng chia nhỏ các cấu trúc UI/UX phức tạp thành từng component đơn giản hơn.
* Đi kèm với ReactJS là rất nhiều các công cụ phát triển giúp cho việc debug code một cách dễ dàng hơn.

**3.1.1.6 Bootstrap**

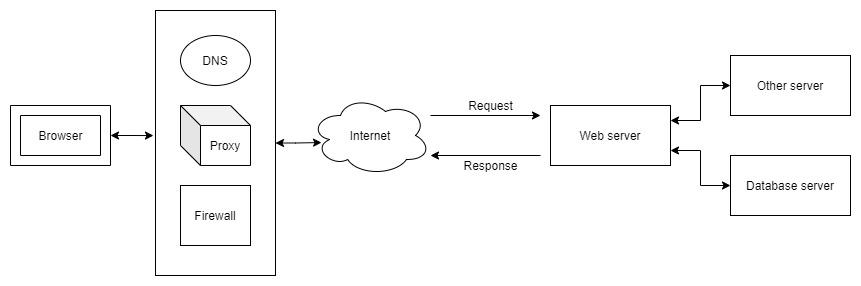
Bootstrap là framework HTML, CSS, và JavaScript phổ biến nhất để phát triển các trang web có yếu tố tính năng phản hồi và tích hợp trên thiết bị di động. Nếu như bạn đang có nhu cầu muốn sở hữu một website chuẩn responsive, có khả năng tương thích trên mọi trình duyệt và thiết bị di động thì chắc chắn công nghệ này sẽ là thành phần lý tưởng không thể thiếu để dùng đến.

Bootstrap chứa các tập tin JavaScript, CSS và fonts đã được biên dịch và nén lại. Ngoài ra, Bootstrap được thiết kế dưới dạng các mô-đun. Do đó, dễ dàng tích hợp với hầu hết các mã nguồn mở như WordPress, Joomla, Magento, …Trong đó, Bootstrap mang đến nhiều chức năng nổi bật.

* Bootstrap cho phép người dùng truy cập vào thư viện “khổng lồ” các thành tố dùng để tạo nên giao diện của một website hoàn chỉnh như font, typography, form, table, grid…
* Bootstrap cho phép bạn tùy chỉnh framework của website trước khi tải xuống và sử dụng nó tại trang web của khung.
* Tái sử dụng các thành phần lặp đi lặp lại trên trang web.
* Bootstrap được tích hợp jQuery. Bạn chỉ cần khai báo chính xác các tính năng trong quá trình lập trình web của bạn.
* Định nghĩa glyphicons nhằm giảm thiểu việc sử dụng hình ảnh làm biểu tượng và tăng tốc độ tải trang.

### **3.1.2 Overall architecture**

**3.1.2.1 Một số thành phần cần thiết**

****

**a. Browser**

Trình duyệt *(browser)* là một phần mềm giúp người dùng giao tiếp với server. Công việc chính của nó là biên dịch code HTML và hiển thị *(rendering)* trang web cho user từ hệ thống được phát triển bằng ReactJS, NodeJs. Đồng thời giúp tăng khả năng tương tác và trải nghiệm người dùng.

**b. DNS**

DNS Server hay còn được gọi là Domain Name System là một hệ thống đặt tên theo thứ tự cho máy tính, dịch vụ tham gia vào Internet. Nó liên kết nhiều thông tin đa dạng với tên miền được gán cho chúng sao cho người dùng có thể sử dụng tên miền đó để tìm hiểu các thông tin mà họ cần biết. Điều quan trọng là phải lựa chọn một tên miền có ý nghĩa cho người dùng, có liên kết với các thiết bị mạng khác để định vị và cung cấp thông tin cho người dùng trên toàn thế giới.

**c. Proxy server**

Web Management/Proxy server thực hiện giao tiếp với client người dùng bằng cách nhận request từ client và chuyển tiếp request đến application server để tiến hành xử lý, đồng thời trả response về client sau khi xử lý.

Web server còn tiến hành lưu trữ các file như HTML, file ảnh, file CSS, JavaScript….của website đó. Khi lưu trữ các file lên máy chủ sẽ đem lại những lợi ích như: luôn luôn sẵn sàng, luôn luôn kết nối với mạng internet, địa chỉ IP cố định, được bảo vệ bởi nhà cung cấp.

Sử dụng một máy chủ Server là điều cần thiết trong việc xây dựng website. Trong hệ thống POS này sẽ sử dụng Apache HTTP Server.

**d. Web server**

Application server lưu trữ toàn bộ hệ thống POS, nhận và xử lý các request từ phía người dùng, cung cấp các API để thực hiện logic nghiệp vụ và tương tác với Database server, Payment server.

Các module sẽ đảm nhiệm các chức năng cụ thể tương ứng với các request: Quản lý menu, Quản lý đơn hàng, giỏ hàng và thanh toán, Quản lý tài khoản, Quản lý mã QR, Phản hồi.

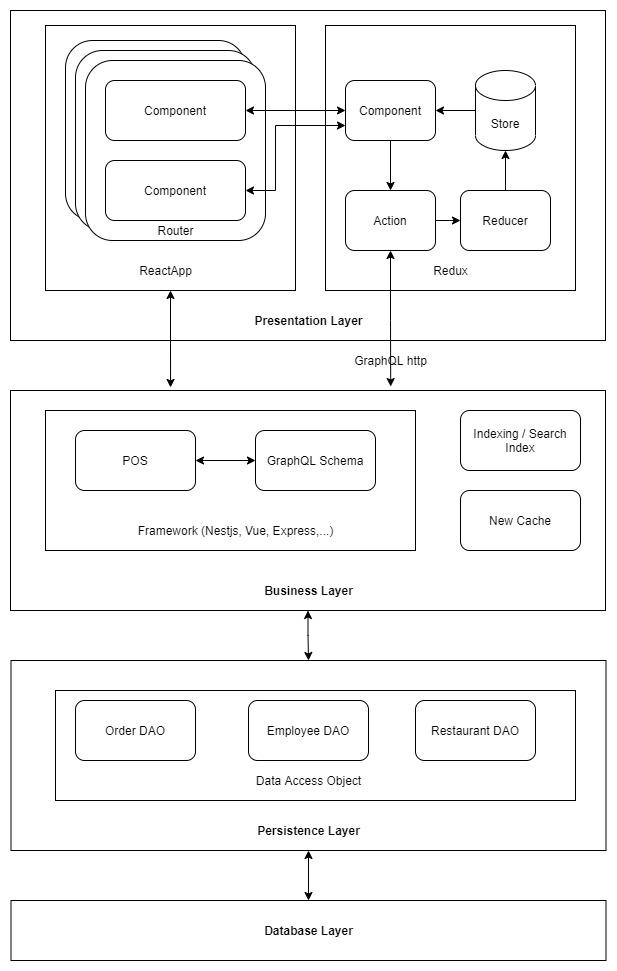
**e. Database Server**

Tổ chức lưu trữ và truy xuất dữ liệu, hệ cơ sở dữ liệu được cài đặt trên server này. Server sử dụng hệ cơ sở dữ liệu là Postgresql.

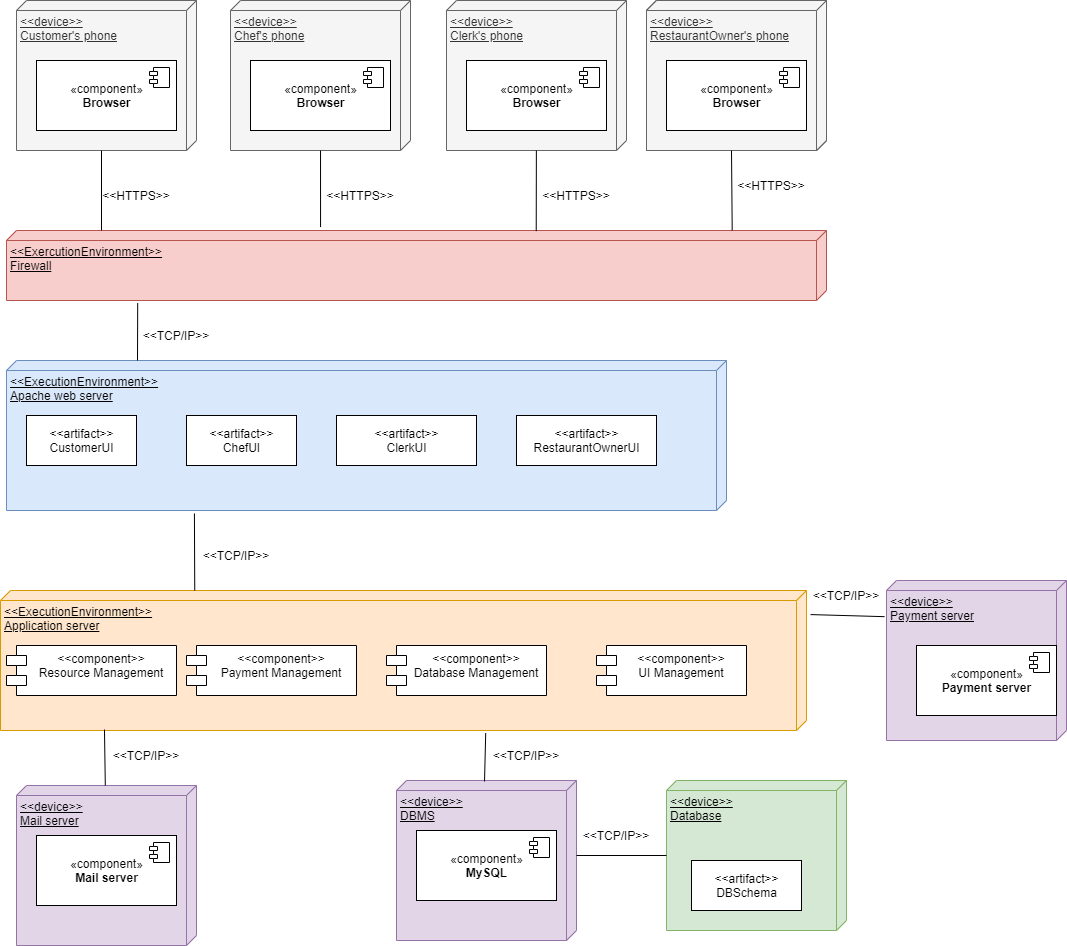
**f. Các dịch vụ, server khác**

* Hệ thống sẽ sử dụng các dịch vụ bên thứ ba để phục vụ cho chức năng thanh toán (Ví điện tử và Internet Banking)
* Người dùng đăng nhập bằng tài khoản Facebook hay số điện thoại thì cần truy cập đến các server cần thiết khác.

**3.1.2.2 Kiến trúc Layer cho ứng dụng**

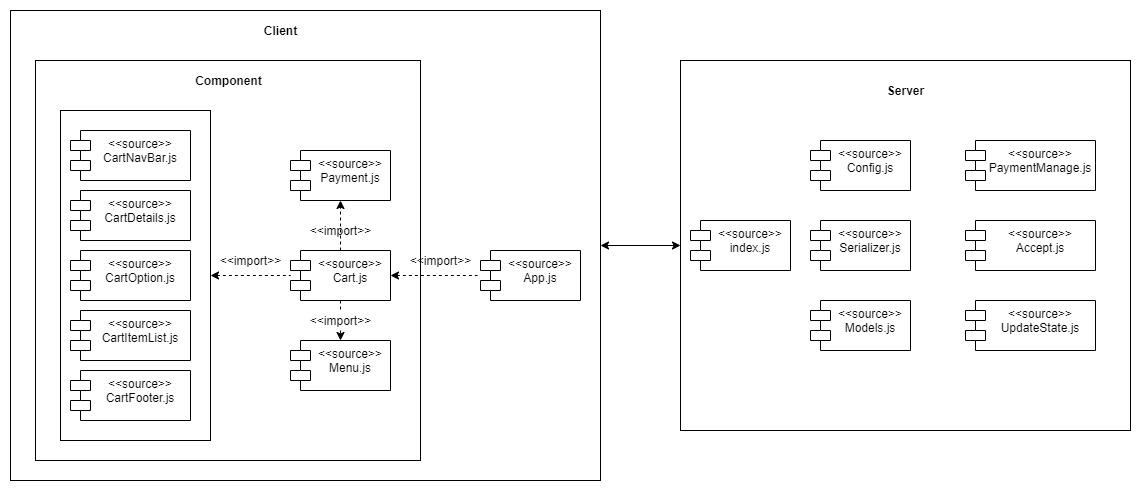


**3.1.3 Deployment view**

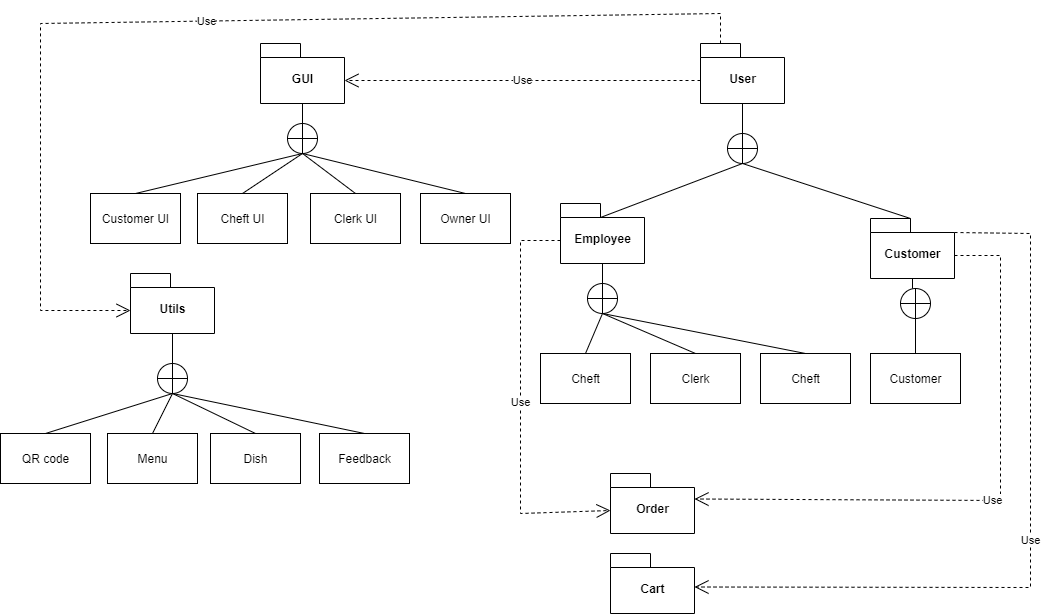


## **3.2. Draw an implementation diagram for Major (not all) functional requirements**

* Implement diagram for Order (Cart Component)



* Package Diagram for system



# **#4: Implementation – Sprint 1**

**4.1. Setting up:**

* Nhóm sử dụng công cụ git và tạo một git repository để quản lý các file hiện thực của chương trình.

Đường dẫn đến trang github của project của nhóm: https://github.com/than2k1/se\_assignment