

ĐẠI HỌC QUỐC GIA TPHCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

---♦♦♦---



CÔNG NGHỆ PHẦN MỀM

Giáo viên hướng dẫn

Huỳnh Tường Nguyên

Bùi Hoài Thắng

Sinh viên thực hiện

Hồ Trần Nhật Nguyên 1914371

Nguyễn Thành Đạt 1913060

Huỳnh Đinh Quang Khải 1812612

Phạm Quang Sơn 1914973

Huỳnh Gia Khoa 2011414

Lớp

L02

Nhóm

Nhóm 9

Hồ Chí Minh, tháng 2 năm 2022

Mục lục

1. Task 1. Requirement elicitation	3
Task 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?	3
Task 1.2: Describe all functional and non-functional requirements of the desired system. Draw a use-case diagram for the whole system.	6
Task 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.	10
2. Task 2	13
Task 2.1. Draw an activity diagram to capture Major (not all) functional requirements of the desired system.	13
Task 2.2. Draw a sequence diagram for use-case in Task 1.3.	14
Task 2.3. Draw a class diagram.	15
3. Task 3	15
Task 3.1. Describe an architectural approach you will use to implement the desired system	15
Task 3.2. Draw an implementation diagram for Major (not all) functional requirements	18
4. Task 4	19
Task 4.1. Setting up. The team creates an online repository (github, bitbucket, etc) for version control.	19
Task 4.2. Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files/ folders	19
Task 4.3. Implement a Minimum Viable Product (MVP) for the menu screen in Figure 2 and demonstrate the result. MVP means that do the least to be able to demonstrate. That means at this stage, no need for a database to store all menu items, customers, etc. Data can be hard coded in code files.	19
5. Task 5	19

1. Task 1. Requirement Elicitation

Task 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?

a. Bối cảnh dự án (context):

- Điểm bán hàng (Point of sale - POS) hay điểm mua hàng là thời gian và địa điểm mà tại đó hoàn thành giao dịch bán lẻ. Tại điểm bán hàng, người bán tính toán số tiền khách hàng nợ, cho biết số tiền đó, có thể lập hóa đơn cho khách hàng và chỉ ra các tùy chọn để khách hàng thanh toán. Trong kinh doanh nhà hàng, hệ thống POS thường bao gồm đặt bàn, đặt đồ ăn, cảnh báo, thanh toán, xử lý thẻ tín dụng và quản lý khách hàng... Hệ thống này còn được gọi là Point of Service vì nó có thể xử lý các đơn đặt hàng và điểm bán hàng. Những hệ thống như vậy được kỳ vọng làm tăng cường trí tuệ kinh doanh, giảm thiểu lãng phí và tăng cường quy mô trở thành một doanh nghiệp lớn.
- Trước đại dịch COVID-19, hệ thống POS đã và đang phổ biến trong nhiều ngành nghề khác nhau. Trong bối cảnh COVID 19 hoành hành kết hợp với các chỉ thị giãn cách, các nhà hàng đổi mới với nguy cơ lớn hơn bao giờ hết. Vì vậy tính thông minh, tự động, tương tác cao, tiện dụng, bảo mật và có tính mở rộng (scalable) của hệ thống POS của nhà hàng phải được đề cao hơn bao giờ hết.

- Với thời đại 4.0 hiện nay, việc áp dụng các hệ thống POS sẽ giúp tăng cường trí tuệ kinh doanh (business intelligence), giảm thiểu chi phí không đáng có, xử lý giao dịch một cách tự động. Từ đó mở rộng quy mô kinh doanh.

b. Các bên liên quan (stakeholders):

- Khách hàng (Customer)
- Nhân viên bán hàng (Clerk)
- Bếp trưởng điều hành (Head chef)
- Quản lý nhà hàng (Restaurant manager)
- Giám đốc tài chính (Finance manager)
- Chủ nhà hàng (Restaurant owner)
- Hệ thống thanh toán online (Online payment system)

c. User story:

Khách hàng	<p>Không muốn tải ứng dụng mới khi đặt và gọi món.</p> <p>Không muốn phải đứng xếp hàng.</p> <p>Muốn có dịch vụ đặt bàn trước.</p> <p>Muốn được đánh giá chất lượng dịch vụ và món ăn của nhà hàng.</p> <p>Muốn hạn chế tối đa việc tiếp xúc với nhân viên và bề mặt vật thể.</p> <p>Muốn có nhiều hình thức thanh toán ngoài tiền mặt.</p> <p>Muốn xem và chọn trong thực đơn được sắp xếp, phân loại rõ ràng.</p> <p>Muốn được ghi chú yêu cầu đặc biệt khi gọi món ăn.</p>
Nhân viên	<p>Muốn nắm rõ thông tin về số giờ làm và lương mình nhận.</p> <p>Muốn quản lý các đơn đặt hàng của khách hàng.</p> <p>Muốn được tương tác từ xa với khách hàng trong trường hợp cần thiết.</p> <p>Muốn được tương tác nhanh với quản lý.</p>
Bếp trưởng điều hành	<p>Muốn được tương tác nhanh với các đầu bếp.</p> <p>Muốn được thông báo nhanh các món ăn được đặt.</p> <p>Muốn thông tin nhanh cho nhân viên về tình trạng của các món ăn.</p> <p>Muốn gửi thông báo đến nhân viên về tình trạng đơn hàng của khách hàng.</p> <p>Muốn được nắm bắt rõ thông tin về nguồn cung cấp nguyên liệu.</p>

Quản lý nhà hàng	Muốn giao tiếp từ xa với toàn bộ nhân viên của nhà hàng. Muốn quản lý toàn bộ thực đơn của nhà hàng. Muốn quản lý toàn bộ các đơn hàng của khách hàng. Muốn tương tác với bếp trưởng điều hành về tình trạng của các món.
Giám đốc tài chính	Muốn có thông tin về các giao dịch, số liệu thanh toán và có thể thống kê chi phí và lợi nhuận.
Người dùng hệ thống	Không muốn lộ thông tin cá nhân của mình. Muốn có tài khoản riêng để lưu trữ, bảo mật thông tin bản thân. Muốn tạo tài khoản và chỉnh sửa thông tin cá nhân. Muốn được hỗ trợ từ xa. Muốn thay đổi và tìm lại mật khẩu khi cần thiết.

d. Danh sách các tính năng (features) của hệ thống (What are expected to be done?):

- Quản lý tài khoản (Account management)
- Quản lý khách hàng (Customer management)
- Quản lý thực đơn (Menu management)
- Đặt món ăn (Food ordering)
- Đặt bàn (Table reservation)
- Thanh toán online (Online payment)
- Dịch vụ phản hồi, đánh giá (Feedback service)
- Truy cập bằng QR (QR accession)

e. Phạm vi (Scope) của hệ thống:

- Hệ thống cung cấp tình trạng món ăn, bàn trống và các khuyến mãi cho khách hàng.
- Hỗ trợ việc quản lý thông qua tài khoản cấp cho bếp trưởng điều hành. Cung cấp thông tin về nguồn gốc, số lượng tồn của nguyên liệu.
- Hỗ trợ tương tác thời gian thực giữa các user với nhau.
- Hệ thống chỉ hỗ trợ thanh toán online.
- Đối với tính năng truy cập bằng QR:
 - Chỉ cần có ảnh QR thì khách hàng có thể truy cập.
 - QR chỉ hỗ trợ truy cập trang web, không hỗ trợ việc truy cập vào tài khoản, xác thực thông tin người dùng hay thanh toán.
- Đối với đặt món ăn:
 - Hệ thống chỉ phục vụ khi khách hàng đã đến nhà hàng đặt món, chỉ hỗ trợ đặt hàng từ xa khi đã đăng ký thành viên.
- Đối với thanh toán online:

- Không hỗ trợ việc ghi nợ.
- Hỗ trợ thanh toán qua thẻ VISA, Master và ví điện tử (chỉ dùng Momo).
- Không hỗ trợ thanh toán bằng các loại tiền ảo, ví dụ: Bitcoin,...

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

a. Functional requirements:

- Đối với khách hàng (Customer):
 - Khách hàng có thể quản lý tài khoản cá nhân trên hệ thống (đăng ký, đăng nhập, điều chỉnh thông tin cá nhân)
 - Khách hàng có thể quét mã QR code để truy cập vào trang web của nhà hàng thay vì phải nhập trực tiếp đường dẫn liên kết đến trang web.
 - Khách hàng có thể đặt bàn trước khi đến nhà hàng
 - Khách hàng có thể xem menu của nhà hàng.
 - Khách hàng có thể xem thông tin chi tiết về từng món ăn được chọn.
 - Các món ăn trong menu cung cấp bởi hệ thống được phân loại theo từng loại thực phẩm khác nhau.
 - Khi chọn món ăn, khách hàng có thể ghi chú thêm các yêu cầu đặc biệt đối với món ăn.
 - Khách hàng có thể theo dõi danh sách các món ăn đã chọn.
 - Khách hàng có thể thêm hoặc xóa các món ăn đã chọn.
 - Hệ thống cần cung cấp lựa chọn cho khách hàng ăn tại nhà hàng hoặc mang về.
 - Hệ thống cung cấp các phương thức thanh toán online (thẻ ngân hàng và ví điện tử) cho khách hàng.
 - Khách hàng có thể gửi đánh giá về chất lượng phục vụ và món ăn của nhà hàng.
- Đối với nhân viên bán hàng (Clerk):
 - Nhân viên có thể đăng nhập vào hệ thống để làm việc.
 - Nhân viên có thể quản lý (xem, thêm, sửa, xóa) thông tin của khách hàng.
 - Nhân viên có thể xác nhận hóa đơn của khách hàng trên hệ thống.
 - Nhân viên có thể gửi thông báo cho khách hàng trong trường hợp món ăn khách hàng chọn không còn được phục vụ.
 - Nhân viên có thể cập nhật hóa đơn của khách hàng trên hệ thống.
- Đối với bếp trưởng điều hành (Executive chef):
 - Bếp trưởng có thể đăng nhập vào hệ thống để làm việc.

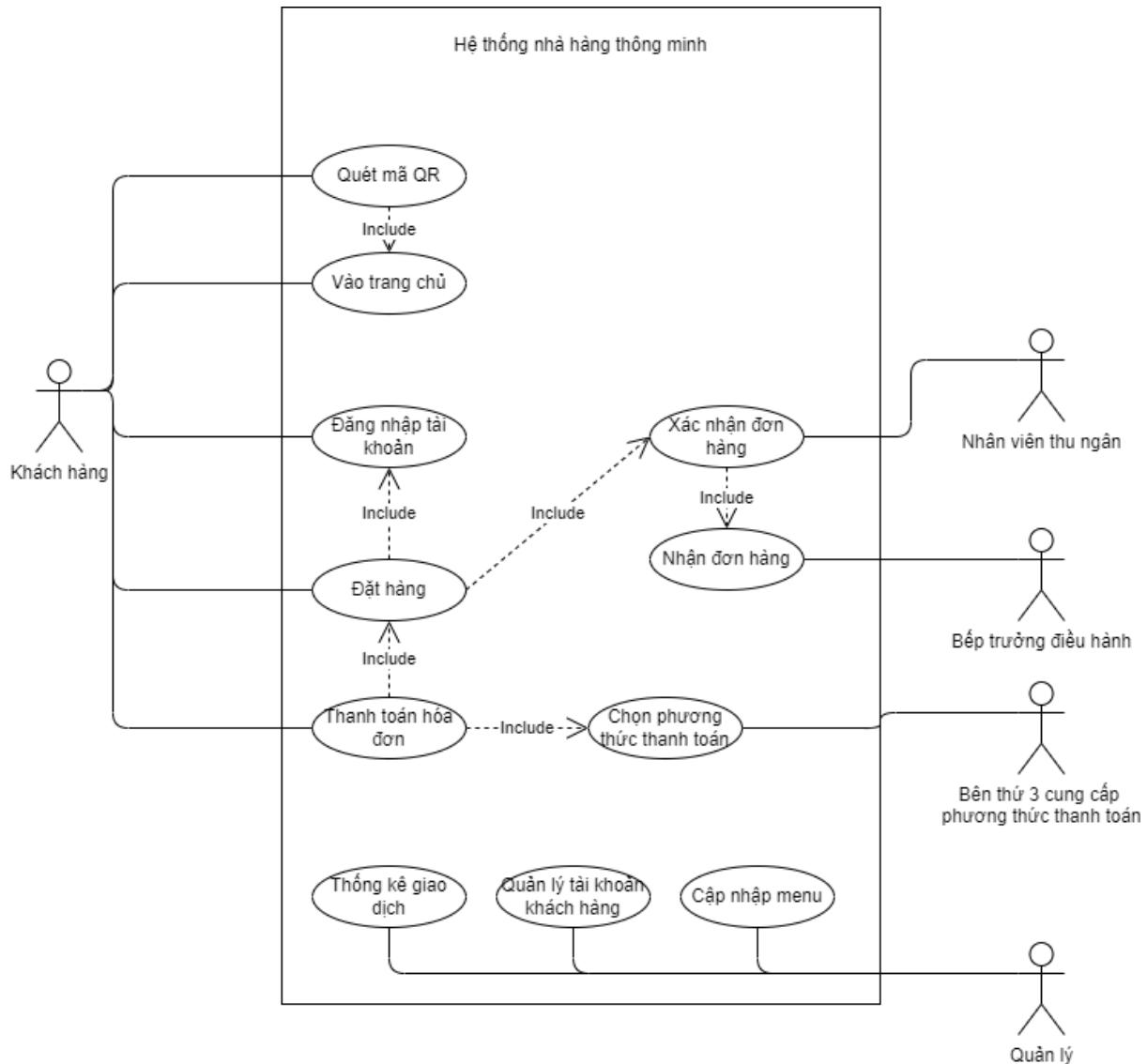
- Bếp trưởng điều hành có thể nhận và xem các đơn hàng đã được xác nhận.
- Bếp trưởng có thể gửi thông báo về tình trạng món ăn để người quản lý nhà hàng cập nhật vào hệ thống.
- Đối với quản lý nhà hàng (Restaurant manager):
 - Người quản lý có thể đăng nhập vào hệ thống để làm việc.
 - Người quản lý có thể quản lý (xem, thêm, sửa, xóa) thông tin của khách hàng.
 - Người quản lý có thể nhận và xem các đơn hàng đã được xác nhận.
 - Người quản lý có thể quản lý và cập nhật (tạo, thêm, xóa, sửa) thực đơn của nhà hàng.
- Một số yêu cầu khác cần xử lý đối với hệ thống POS:
 - Hệ thống có thể nhận thông tin món ăn từ đầu bếp để hiển thị menu thích hợp.
 - Hệ thống cần lưu trữ thông tin từng tài khoản của người dùng.
 - Hệ thống cần có khả năng phân quyền truy cập cho từng vai trò khác nhau của người dùng.
 - Hệ thống cần có khả năng kiểm tra trạng thái (còn phục vụ hay không còn phục vụ) của các món ăn trong thực đơn được người quản lý cập nhật.
 - Hệ thống cần ghi nhận lại lịch sử các giao dịch và đơn hàng đã được thanh toán của người dùng.
 - Hệ thống có thể thống kê tổng doanh thu trong một tháng, quý hoặc năm của nhà hàng.

b. Non-functional requirements:

- Tốc độ, hiệu năng (Performance):
 - Các trang của ứng dụng web load không quá 2 giây
 - Quá trình gửi dữ liệu giữa các người dùng không quá 3 giây
 - Phản hồi của hệ thống đối với các thao tác còn lại của người dùng không quá 1 giây.
 - Mỗi giây có thể xử lý 10 đơn hàng cùng lúc.
 - Thực hiện tối đa 500 đơn hàng mỗi ngày.
- Tính dễ sử dụng (Ease of use)
 - Mỗi chức năng thực hiện dưới 4 thao tác.
 - Một nhân viên bán hàng, người quản lý nhà hàng, bếp trưởng điều hành thành thạo các thao tác trung bình sau 15 phút training.
 - Khách hàng có thể thành thạo các thao tác sau 5 phút training.
- Kích thước (size)
 - Ứng dụng web không yêu cầu người dùng download bất cứ file gì.

- Ứng dụng quản lý hệ thống ở máy chủ kích thước không quá 500mb.
- Độ tin cậy (Reliability)
 - Trung bình số lần truy cập hệ thống thất bại là 2 trong 1000 lần truy cập.
 - Xác suất hệ thống không khả dụng là dưới 0.05%.
 - Tỷ lệ xảy ra lỗi là dưới 0.05%.
- Tính bền vững của hệ thống (Robustness)
 - Thời gian khởi động lại hệ thống không quá 1 phút.
 - Xác suất hư hỏng dữ liệu (data corruption) là dưới 2%.
- Bảo mật, an ninh (Security)
 - Chức năng thanh toán phải áp dụng hoặc sử dụng lại hệ thống có các cơ chế bảo mật: SSL, SET, PCI DSS.
 - Tài khoản Admin được nhập sai tối đa 5 lần.
 - Cảnh báo nếu như có IP máy chủ khác xâm nhập.
 - Trang Web có thể ngăn ngừa tấn công DDOS.
- Tính mở rộng của hệ thống (Scalability)
 - Hệ thống sẽ có thể mở rộng để sử dụng ở nhiều hơn 2 nhà hàng trong tương lai.
 - Đa nền tảng (Multi-platform)
 - Hệ thống có thể sử dụng hiệu quả trên điện thoại di động (Android, IOS), máy tính bảng hay máy tính bàn, laptop (Windows, Linux, Mac) với các trình duyệt (Chrome, Firefox, Safari, Opera).
- Tính bản địa (Localization)
 - Hỗ trợ tiếng Anh và tiếng Việt
- Tính bảo trì (Maintainability)
 - Mỗi lần nâng cấp, bảo trì hệ thống định kỳ (theo quý 3 tháng) thì không mất quá 30 phút
- Tính toàn vẹn, chính xác (Integrity)
 - Kết quả tính toán giá tiền của đơn hàng luôn chính xác 100%
- Tính sao lưu (Backup)
 - Sao lưu kết quả kinh doanh 1 ngày/lần.

c. Use case diagram cho toàn hệ thống:



Bảng danh sách Actor

ID	Tên actor
1	Khách hàng
2	Nhân viên thu ngân
3	Bếp trưởng điều hành
4	Quản lý
5	Bên thứ ba cung cấp phương thức thanh toán

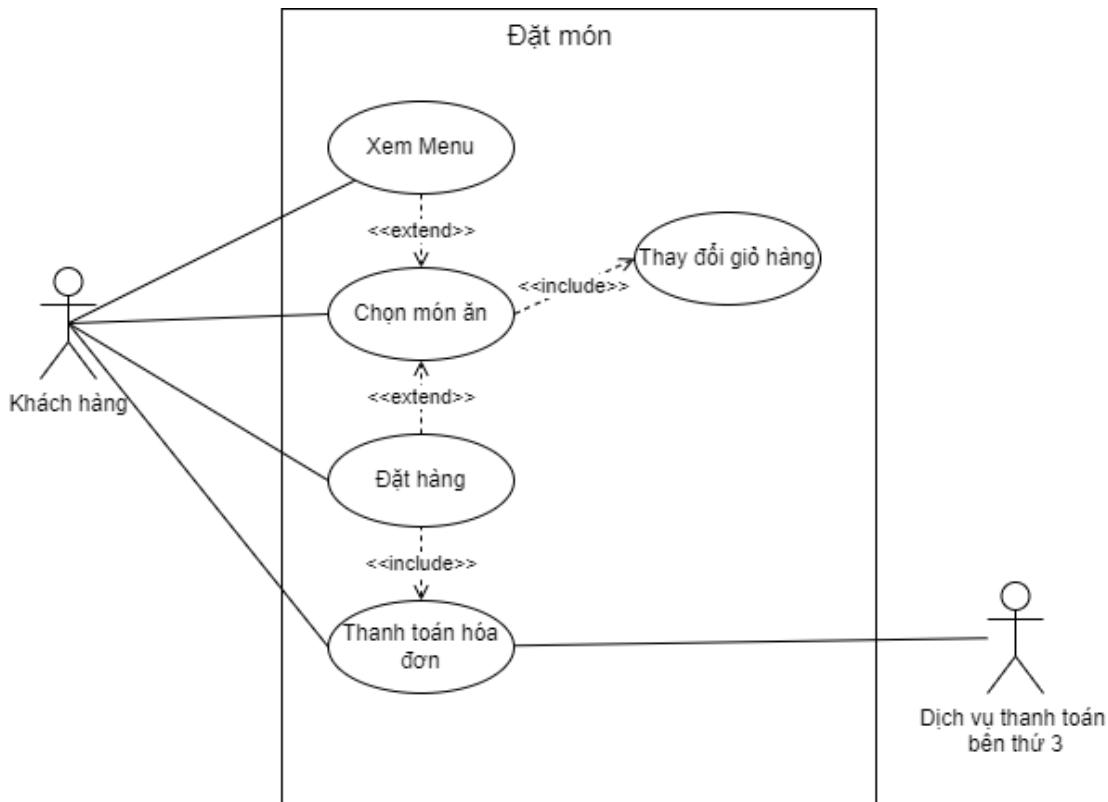
Bảng mô tả các use case chính của hệ thống:

ID	Tên Use case
1	Quét mã QR
2	Vào trang chủ
3	Đăng nhập tài khoản
4	Đặt hàng
5	Thanh toán hóa đơn
6	Lưu giao dịch
7	Xác nhận đơn hàng
8	Nhận đơn hàng
9	Thống kê giao dịch
10	Quản lý tài khoản khách hàng
11	Cập nhật menu

Task 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.

- a. Use-case diagram for a specific feature

Chức năng đặt món được chúng tôi lựa chọn để mô tả.



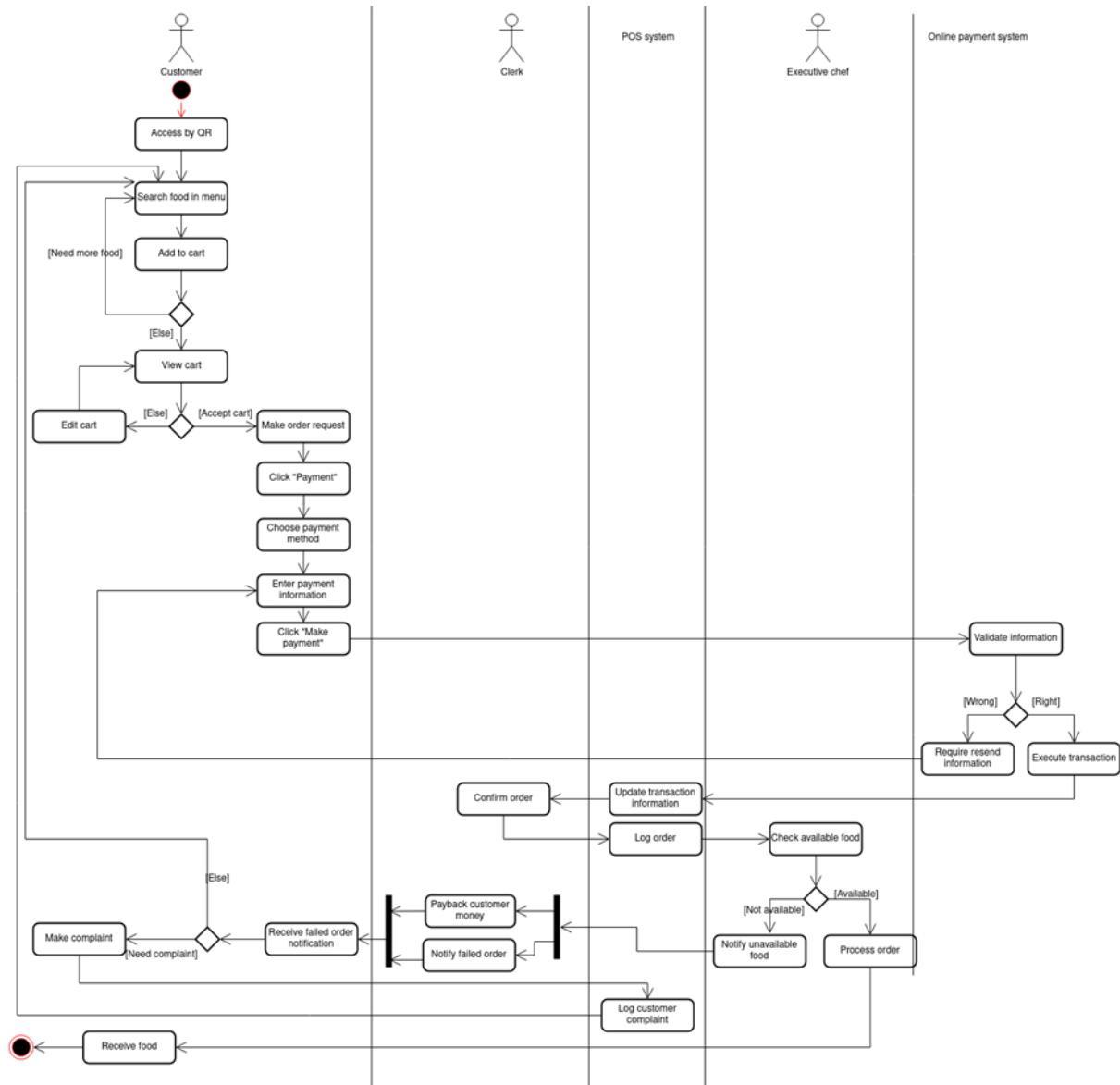
b. Use-case scenario

Use-case Name	Đặt hàng
Actor	Khách hàng
Description	Khách hàng đặt các món ăn trong Menu thông qua web.
Trigger	Khách hàng trực tiếp thực hiện
Preconditions	<ul style="list-style-type: none"> 1. Khách hàng phải sử dụng thiết bị có kết nối Internet vào được website của nhà hàng. 2. Khách hàng phải đăng nhập sử dụng tài khoản đã được đăng ký dưới quyền Guest.
Postconditions	<ul style="list-style-type: none"> 1. Đơn hàng được ghi nhận thành công lên hệ thống. 2. Khách hàng đặt được món ăn mình muốn. 3. Khách hàng chuyển sang bước thanh toán.
Normal Flows	<ul style="list-style-type: none"> 1. Khách hàng bấm chọn vào món ăn mình muốn đặt trên màn hình. 2. Hệ thống hiển thị thông tin món ăn đó cùng với hình ảnh và giá. 3. Khách hàng bấm chọn số lượng cho món ăn được đặt và

	<p>thêm vào giỏ hàng.</p> <ol style="list-style-type: none"> 4. Hệ thống hiển thị các món mà khách hàng chọn trong giỏ hàng. 5. Khách hàng bấm Payment để xác nhận các lựa chọn của mình. 6. Hệ thống thông báo “Đặt hàng thành công” và chuyển tiếp tới trang web thanh toán.
Alternative Flows	<p>Ở bước 1:</p> <ol style="list-style-type: none"> 1a. Người dùng bấm chọn mục thẻ loại đồ ăn trên màn hình để lọc món ăn theo ý của khách hàng. 2a. Hệ thống hiển thị danh sách món ăn được lọc theo loại. <p><i>Tiếp tục bước 1 trong flow.</i></p> <p>Ở bước 5:</p> <ol style="list-style-type: none"> 5a. Người dùng nhấn nút Back quay về tiếp tục chọn món ăn. <p><i>Tiếp tục bước 1 trong flow.</i></p>
Exceptions	Ở bước 5 nếu trong giỏ hàng không có món thì hệ thống hiển thị chưa có món, không nhấn XÁC NHẬN được và chỉ có thể nhấn Back quay trở về Menu.
Note and issues	Giao diện Menu sẽ loại bỏ hoặc đánh dấu những món đã hết và cả thời gian khi hết bàn. Hệ thống phải luôn cập nhật dữ liệu Menu và tình trạng số bàn trong ngày và tuần.

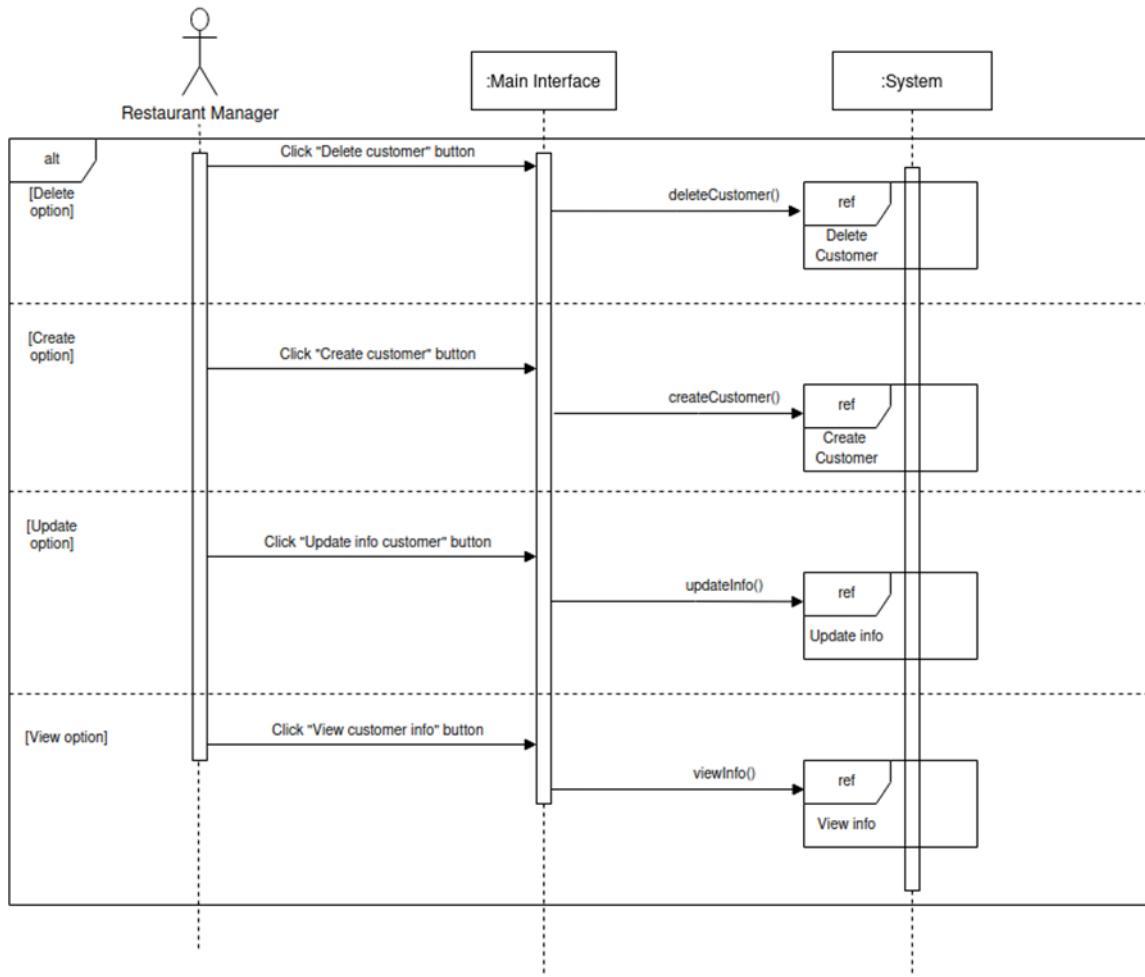
2. Task 2. System Modelling

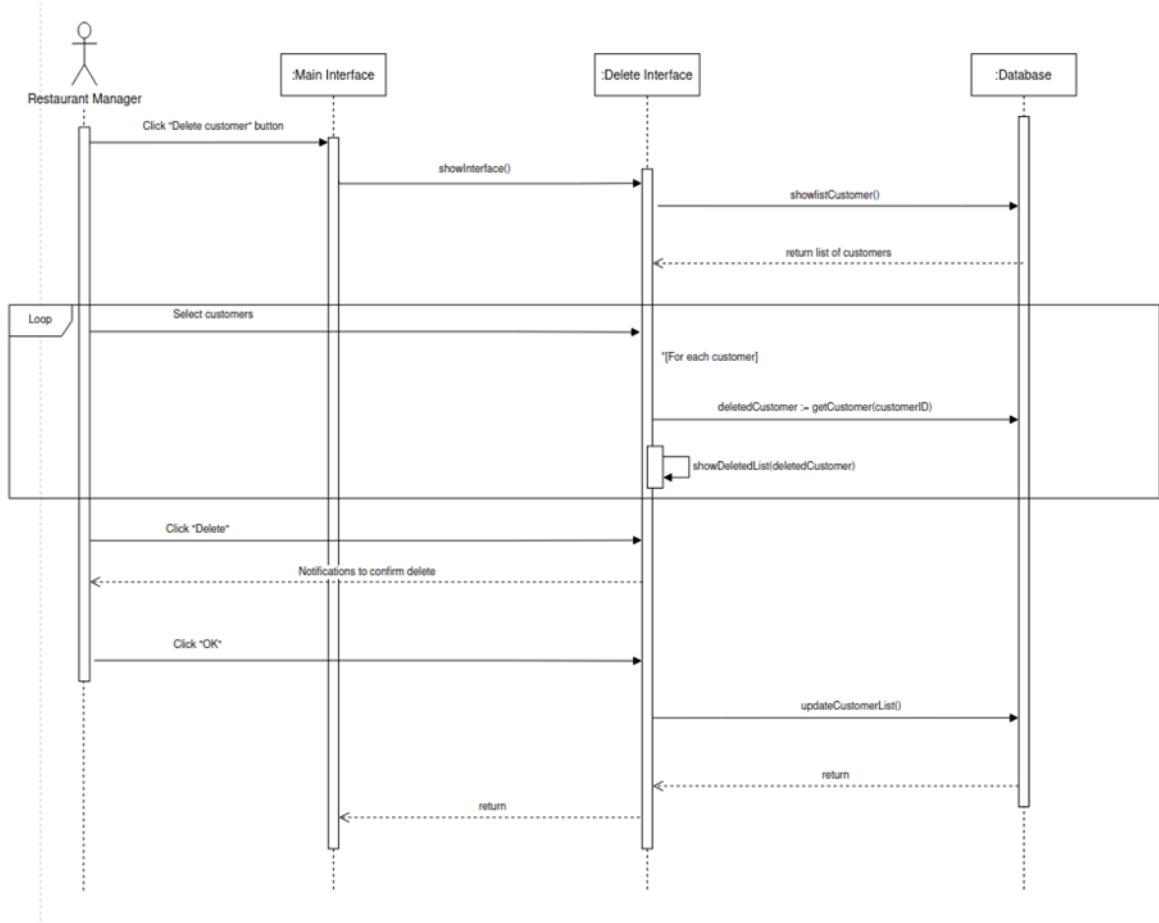
Task 2.1. Draw an activity diagram to capture Major (not all) functional requirements of the desired system.

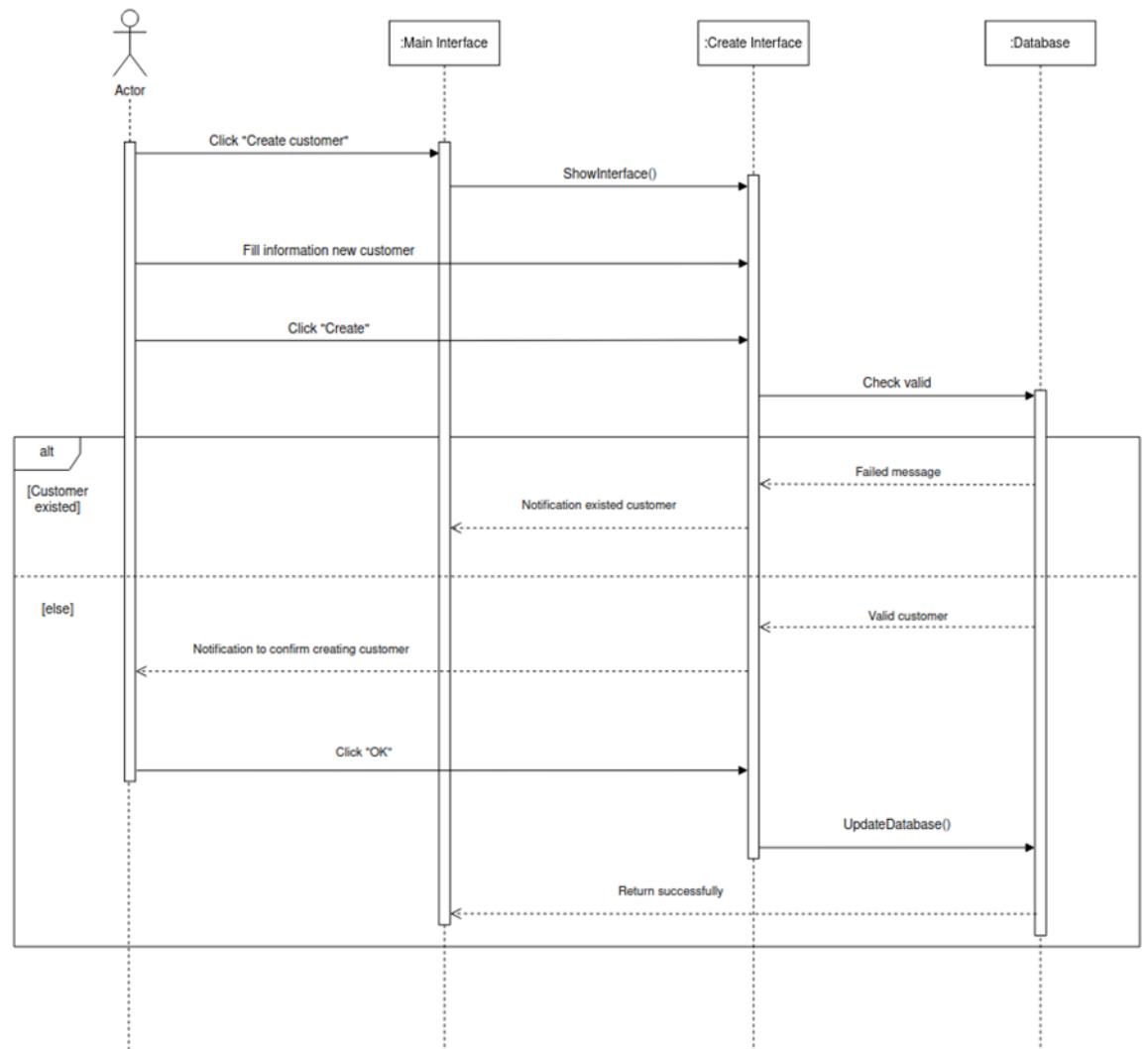


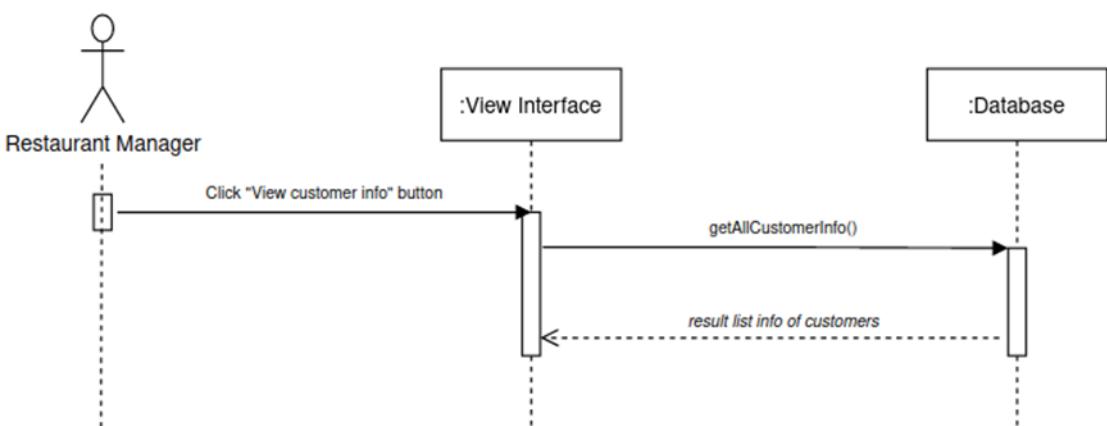
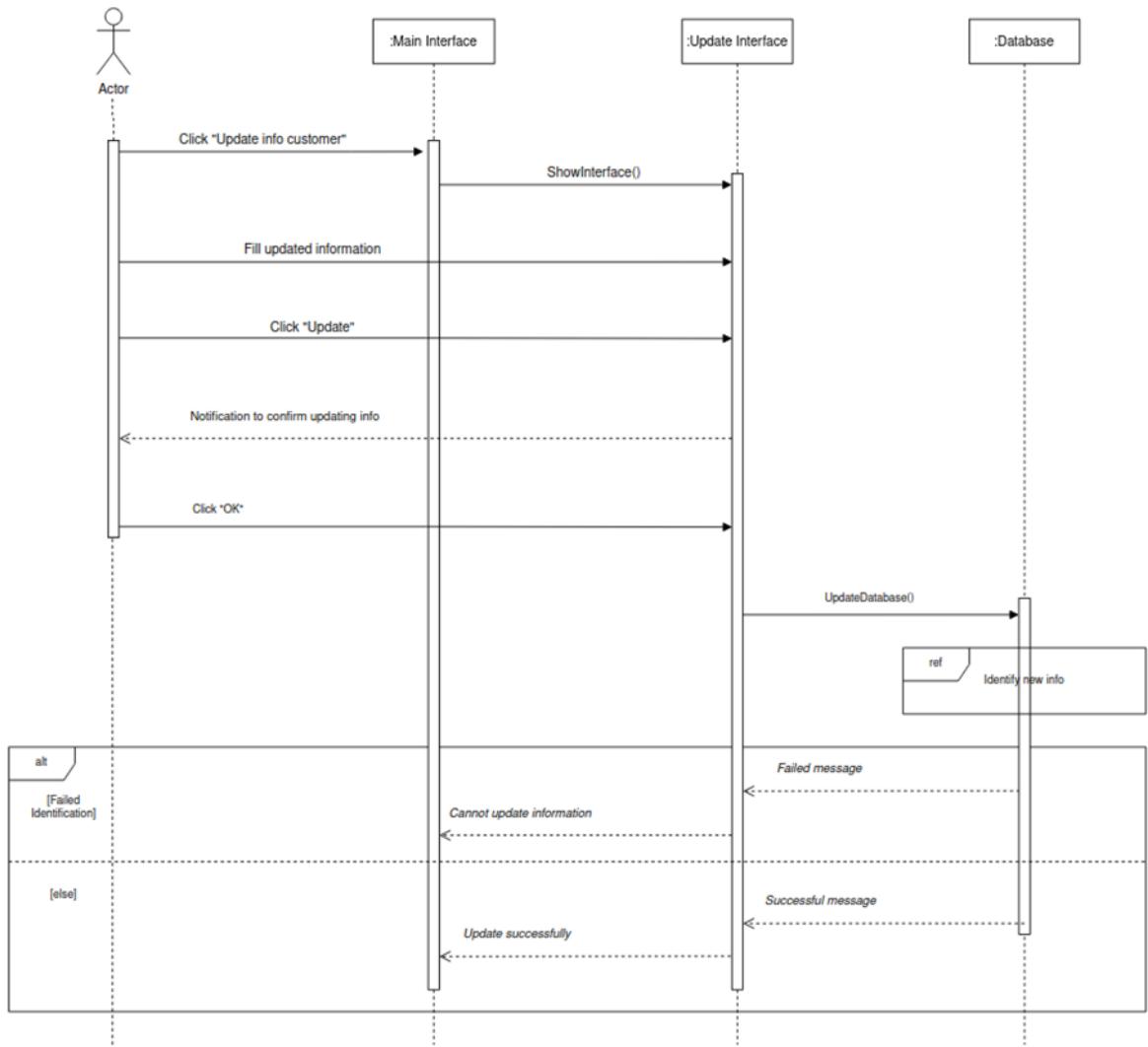
Task 2.2. Draw a sequence diagram for use-case in Task 1.3.

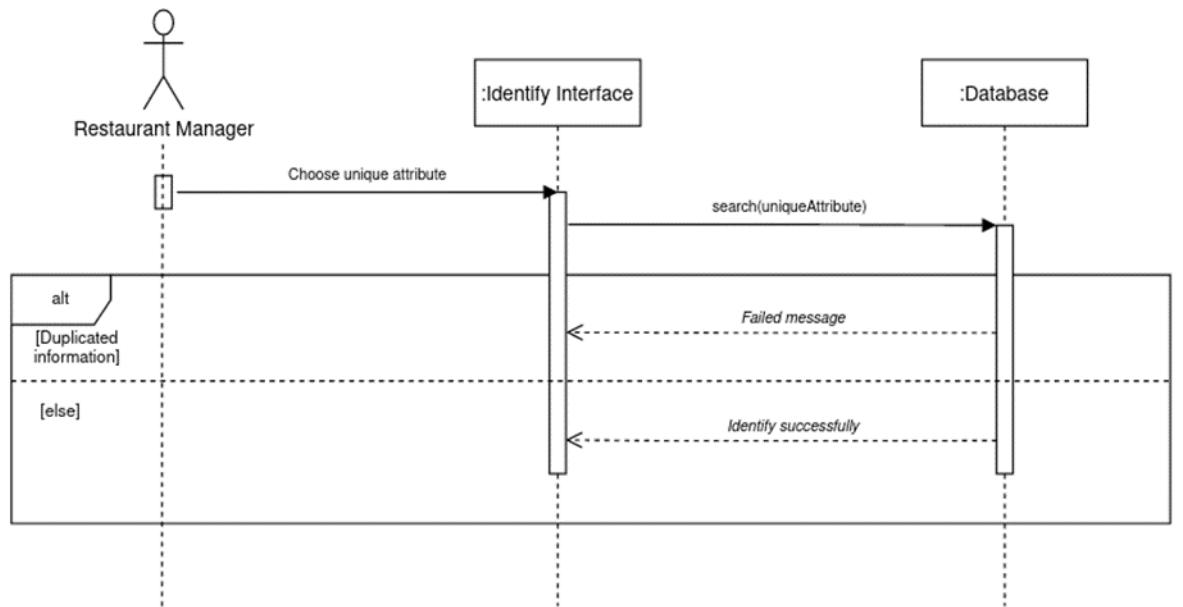
+ Feature: Customer management:



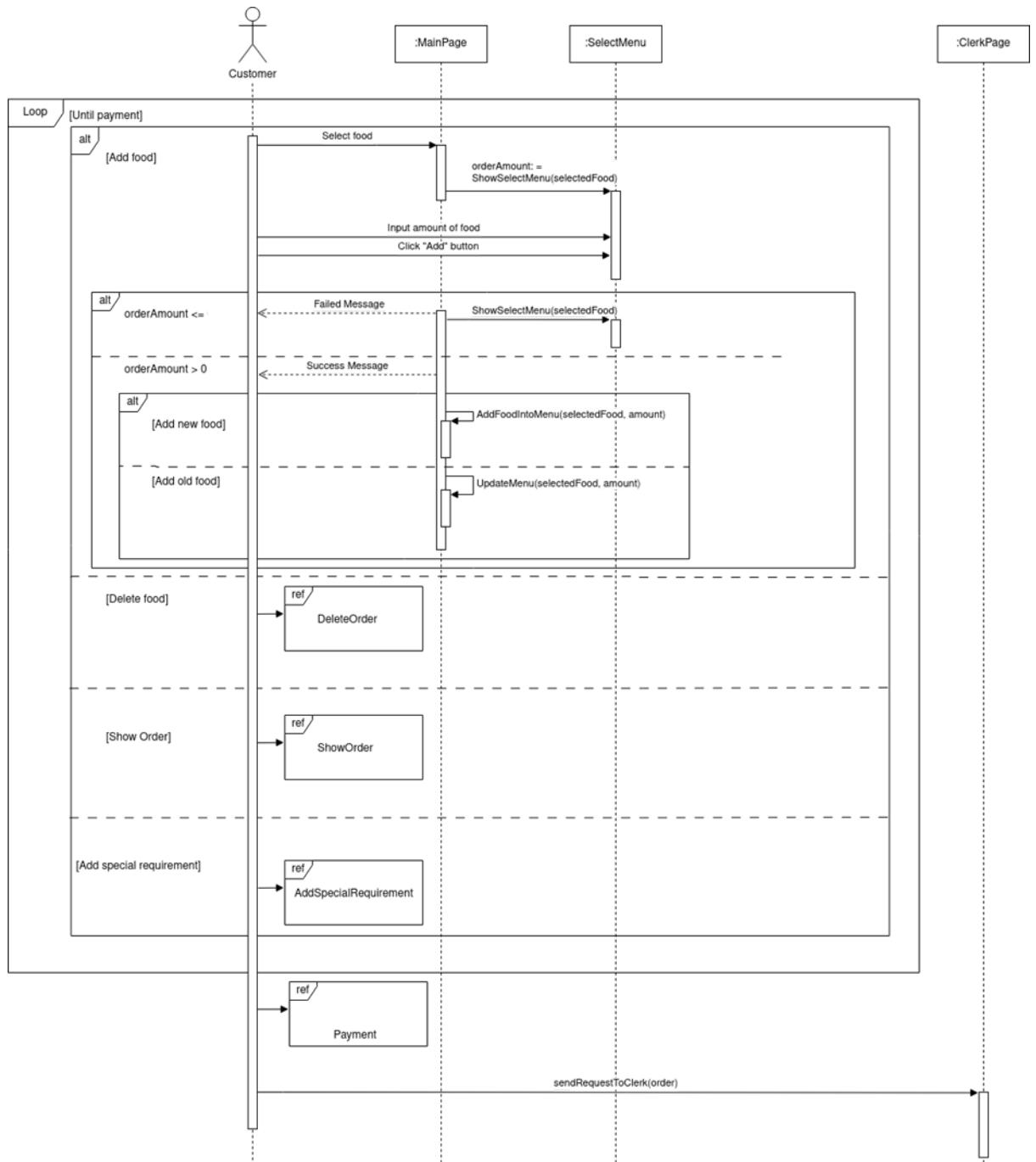


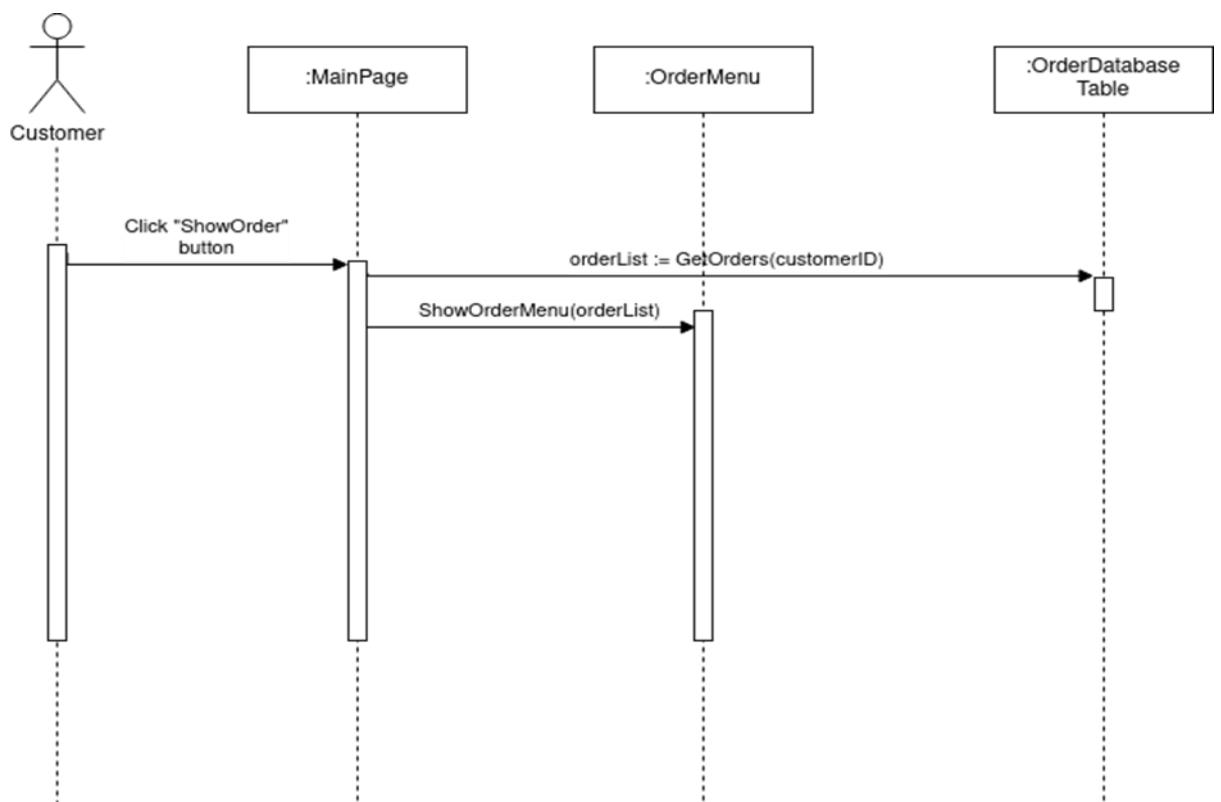
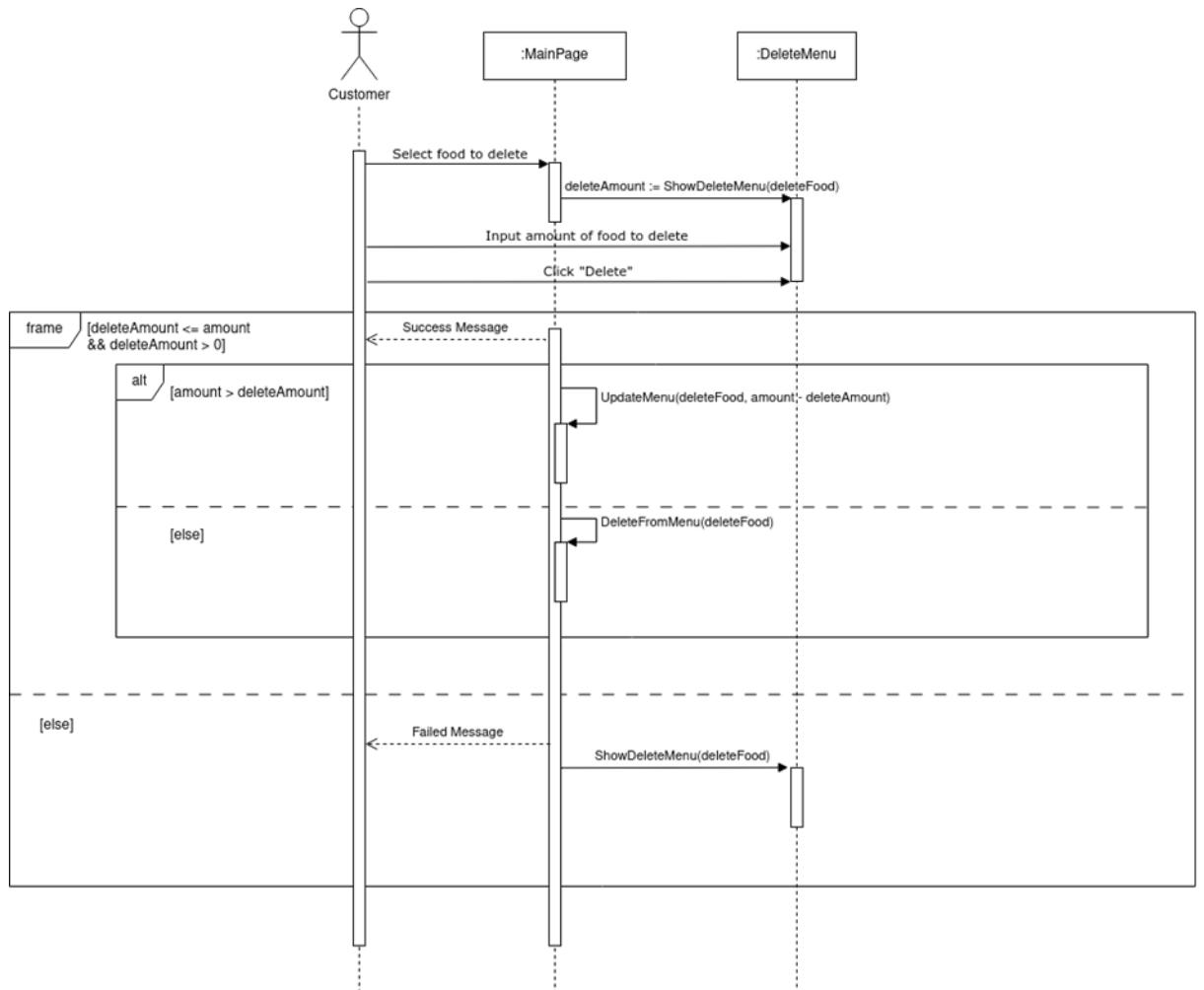


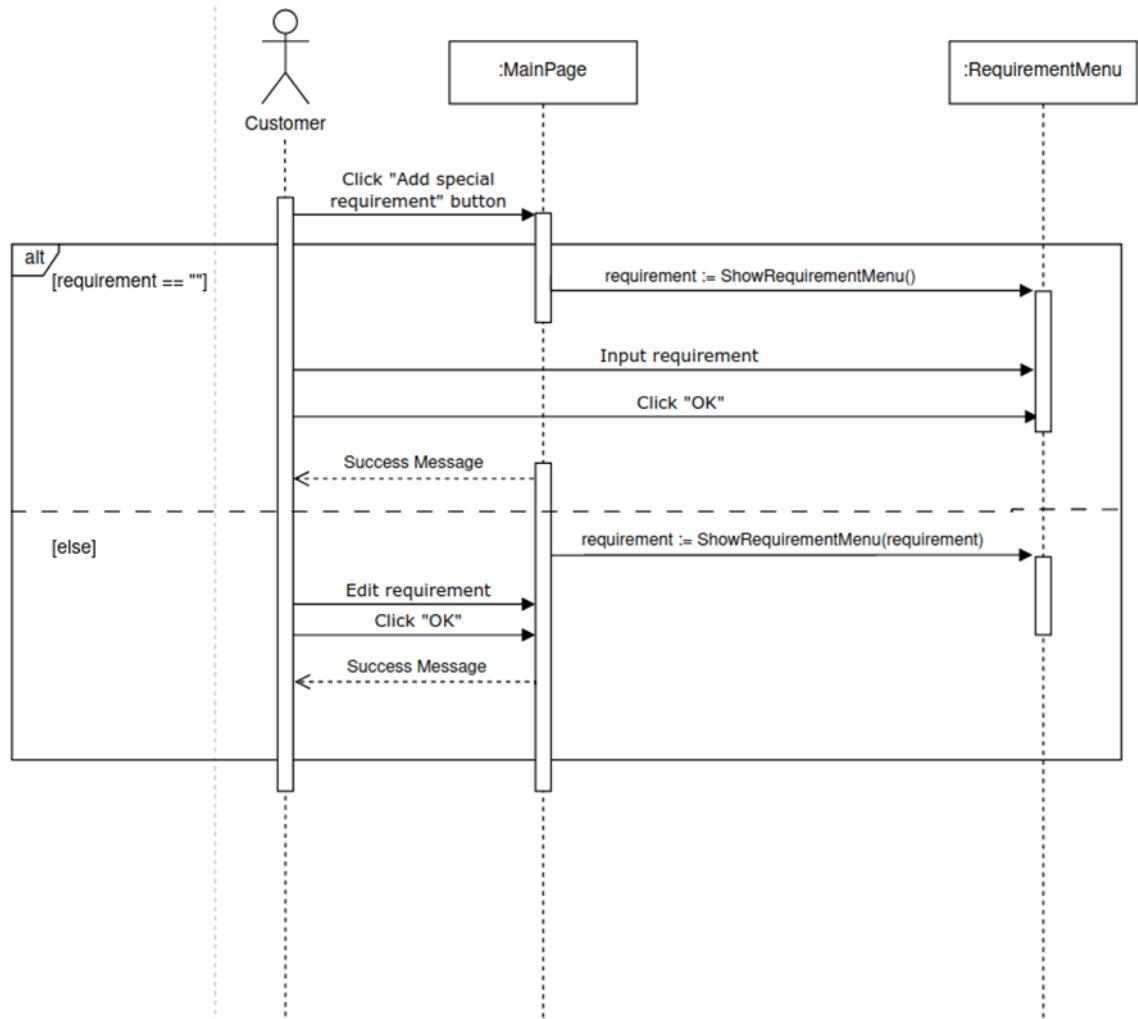


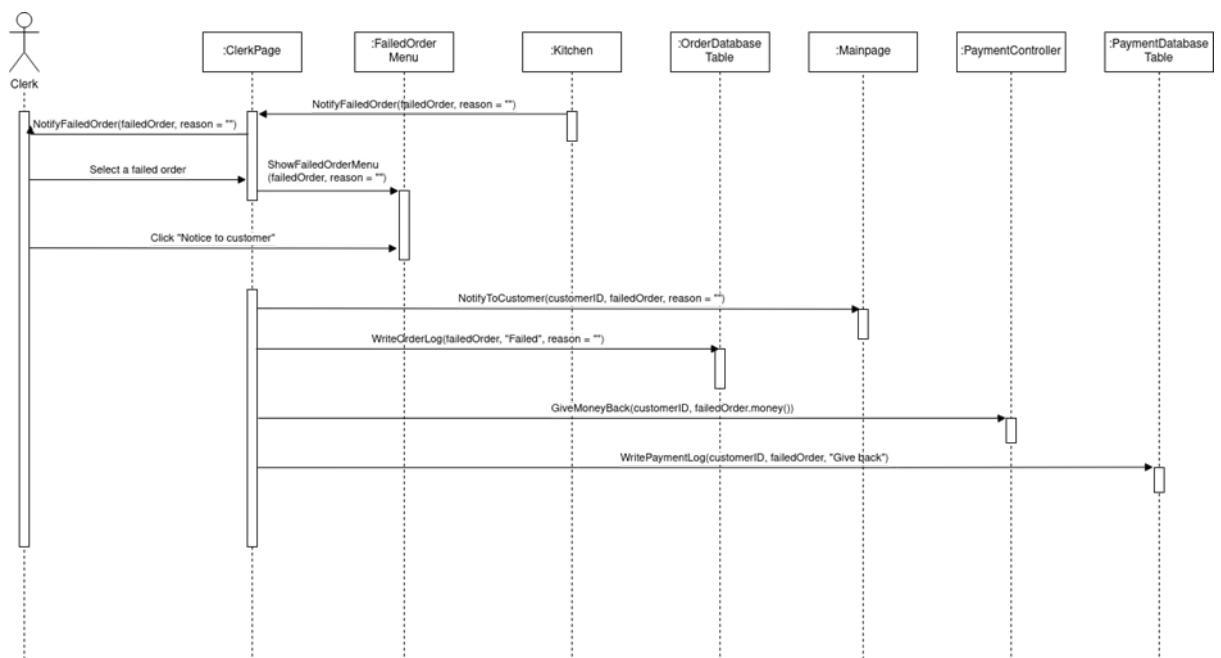
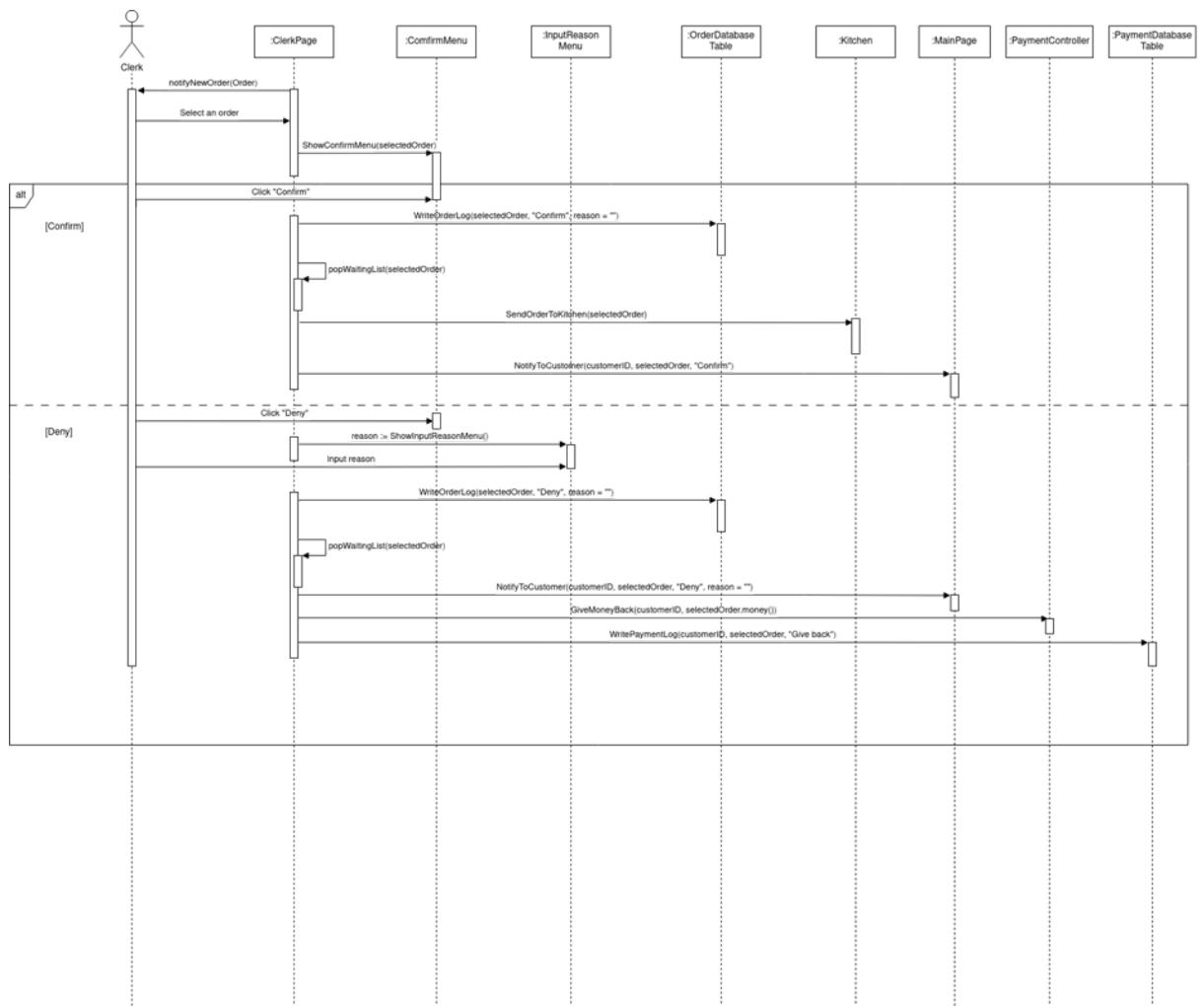


+ Feature: Food ordering:

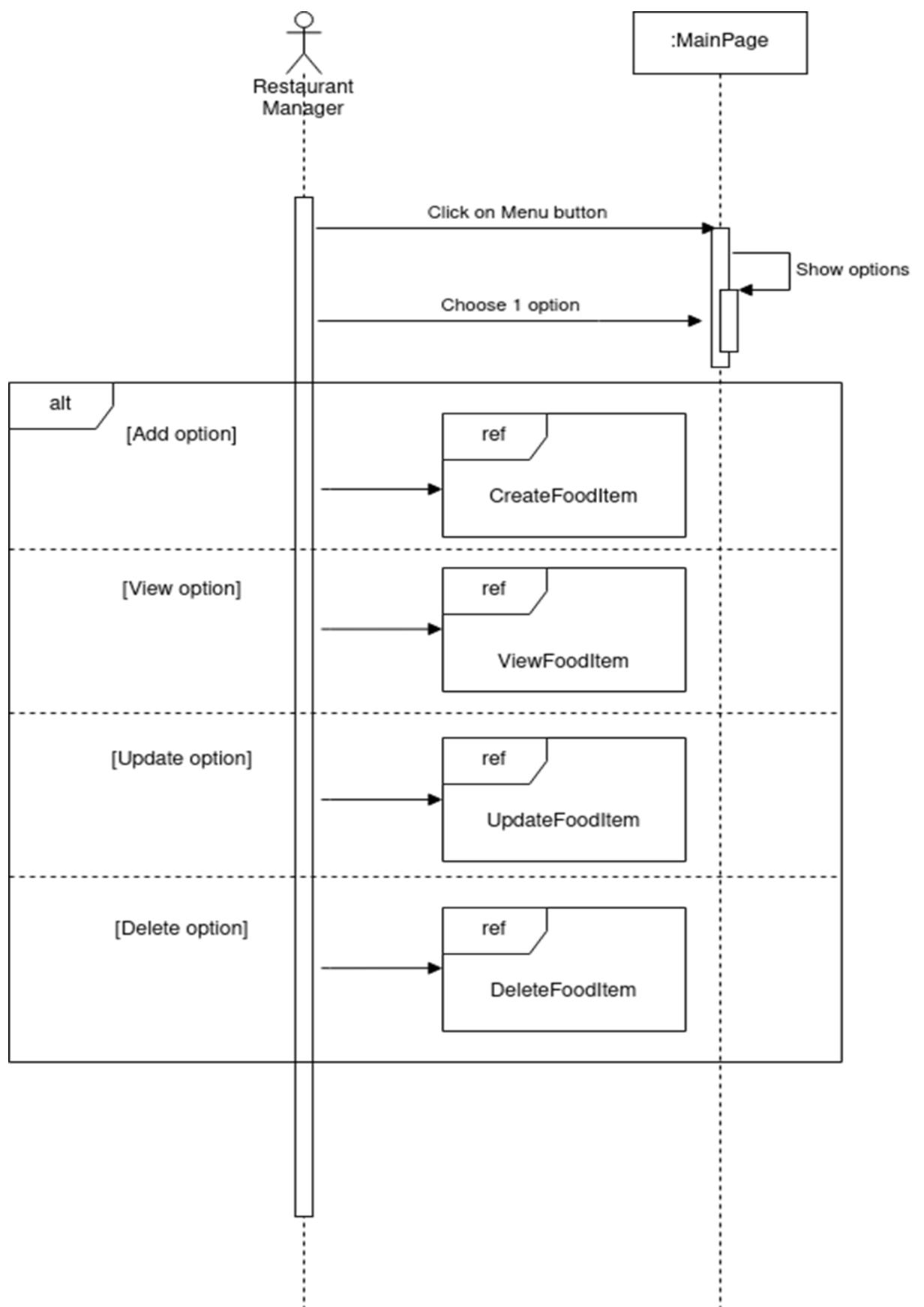


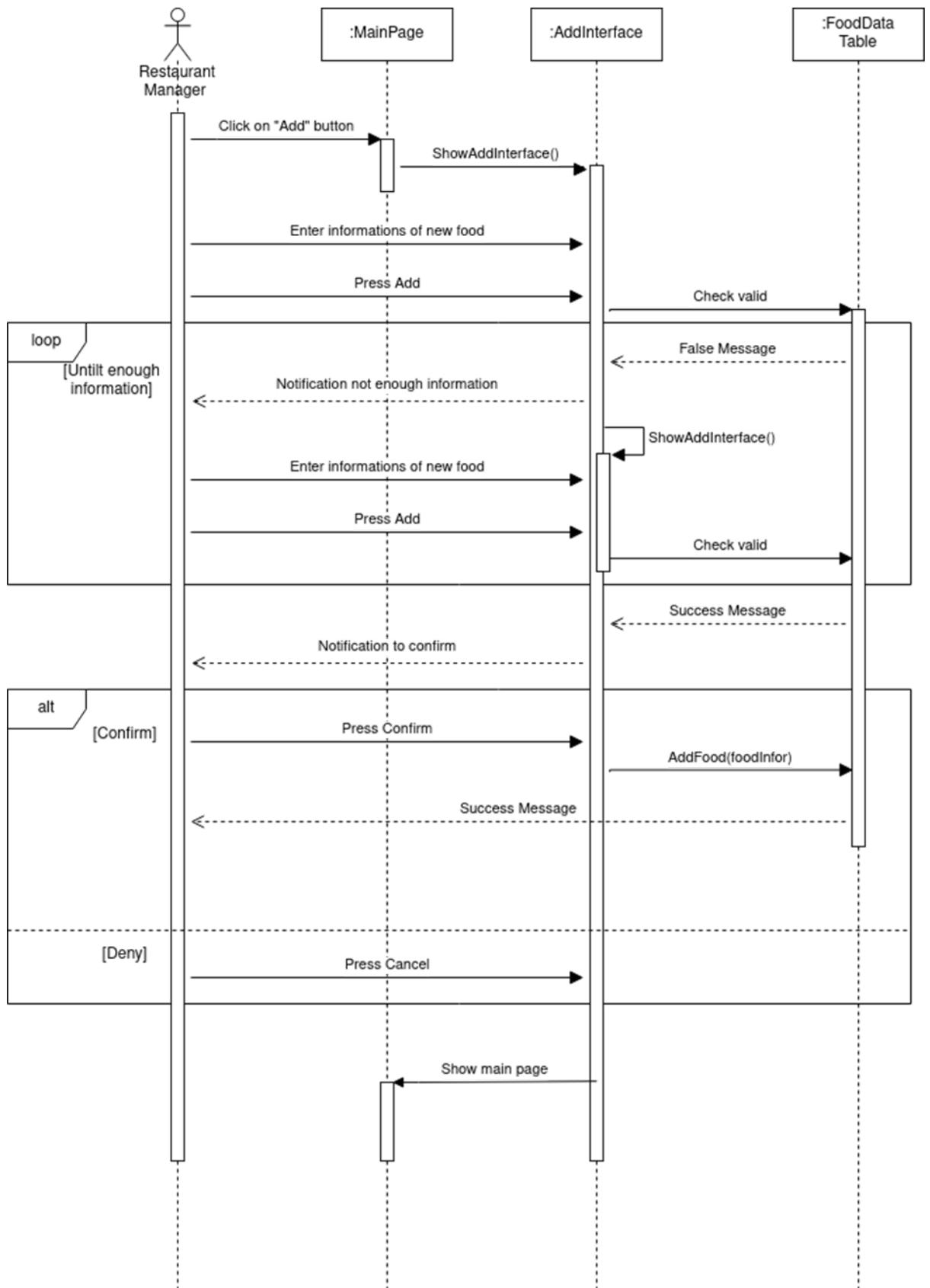


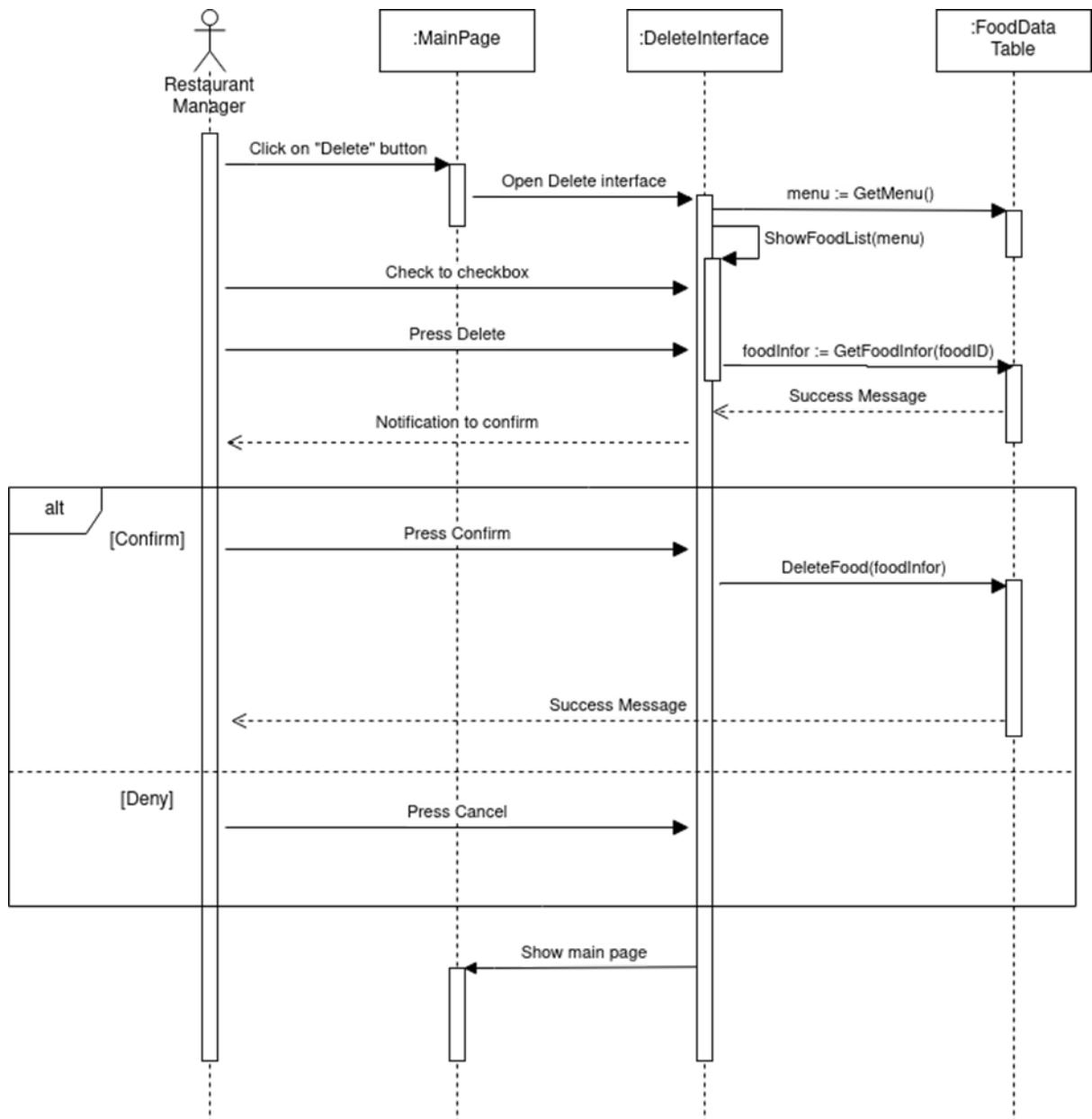


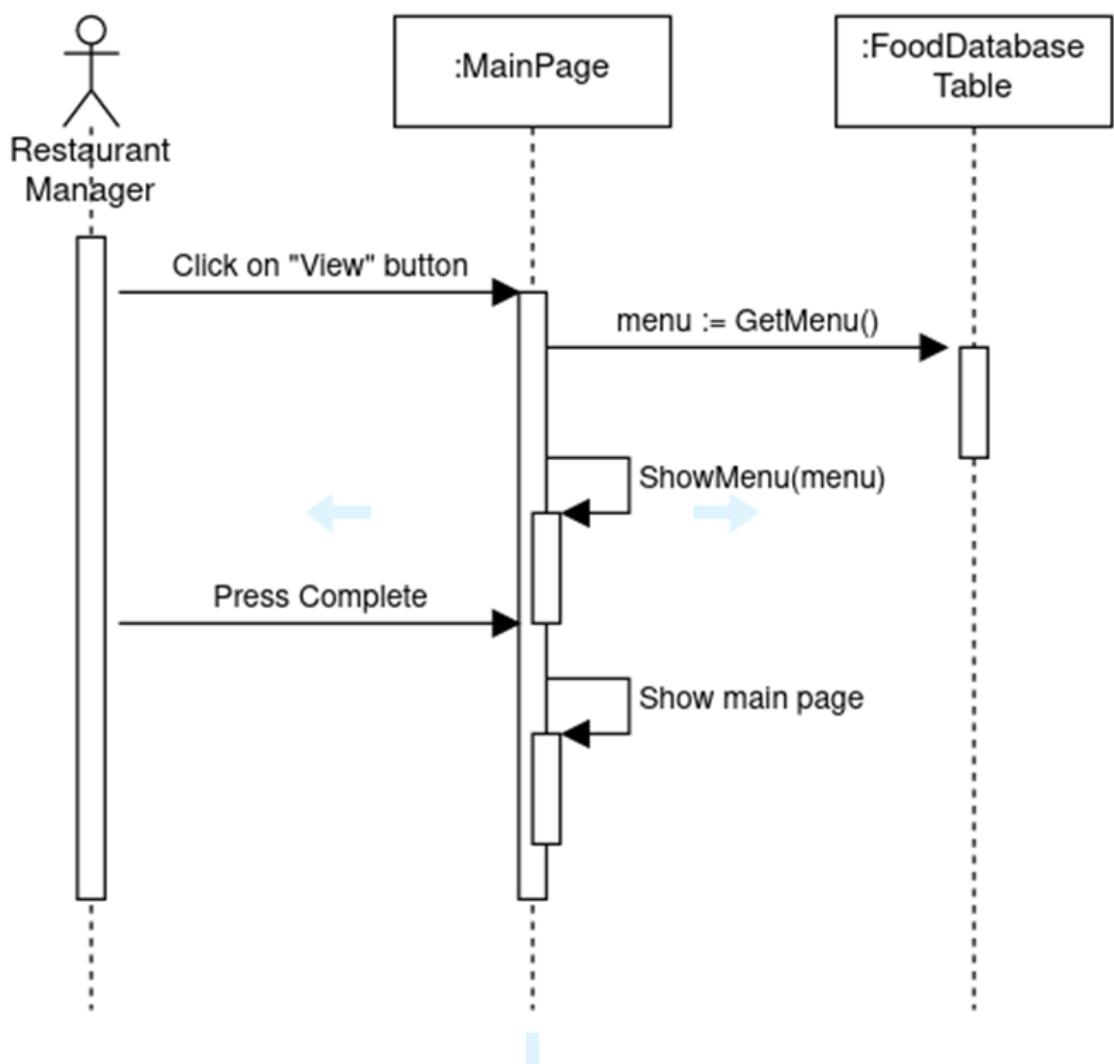


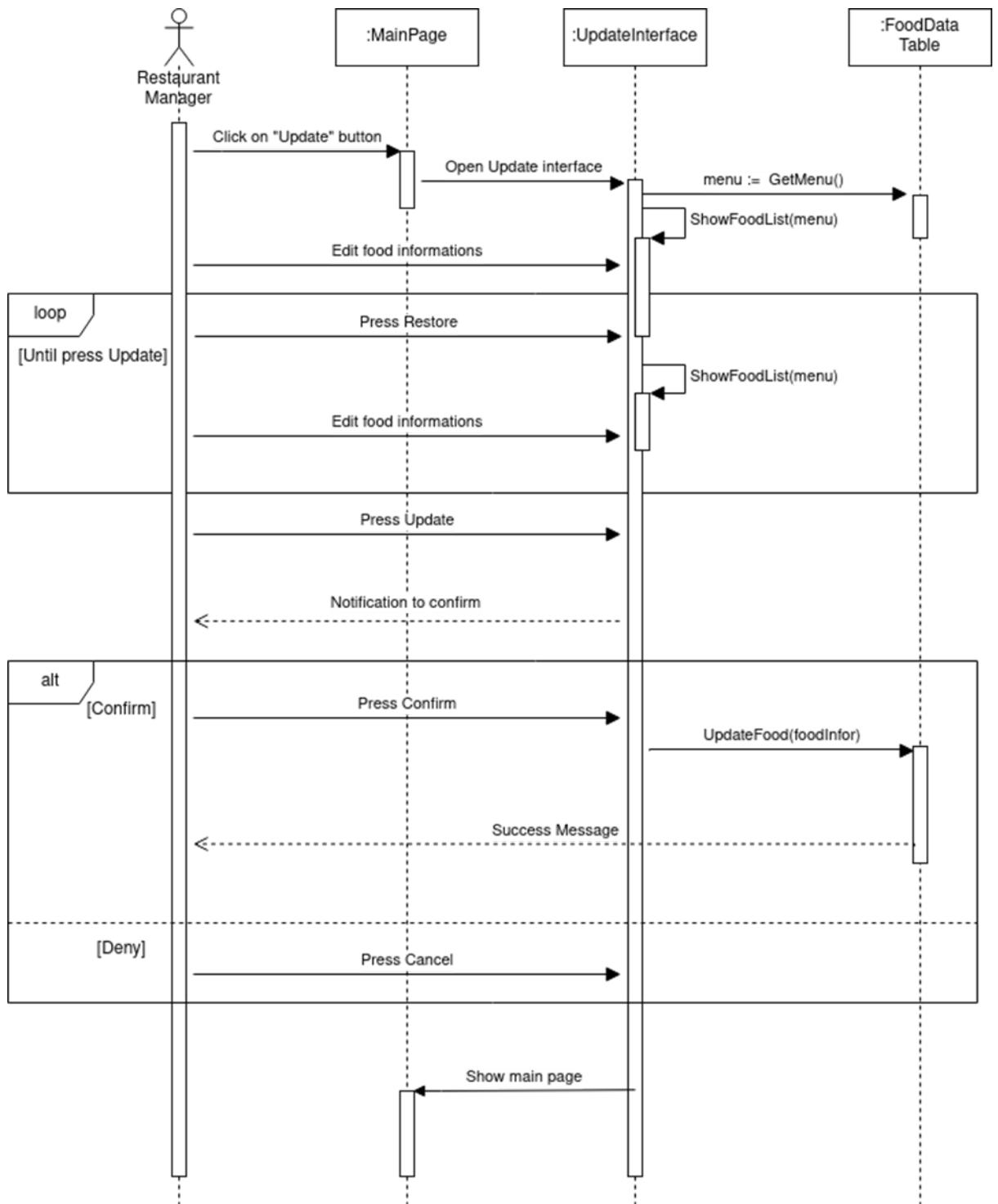
+ Feature: Menu management:



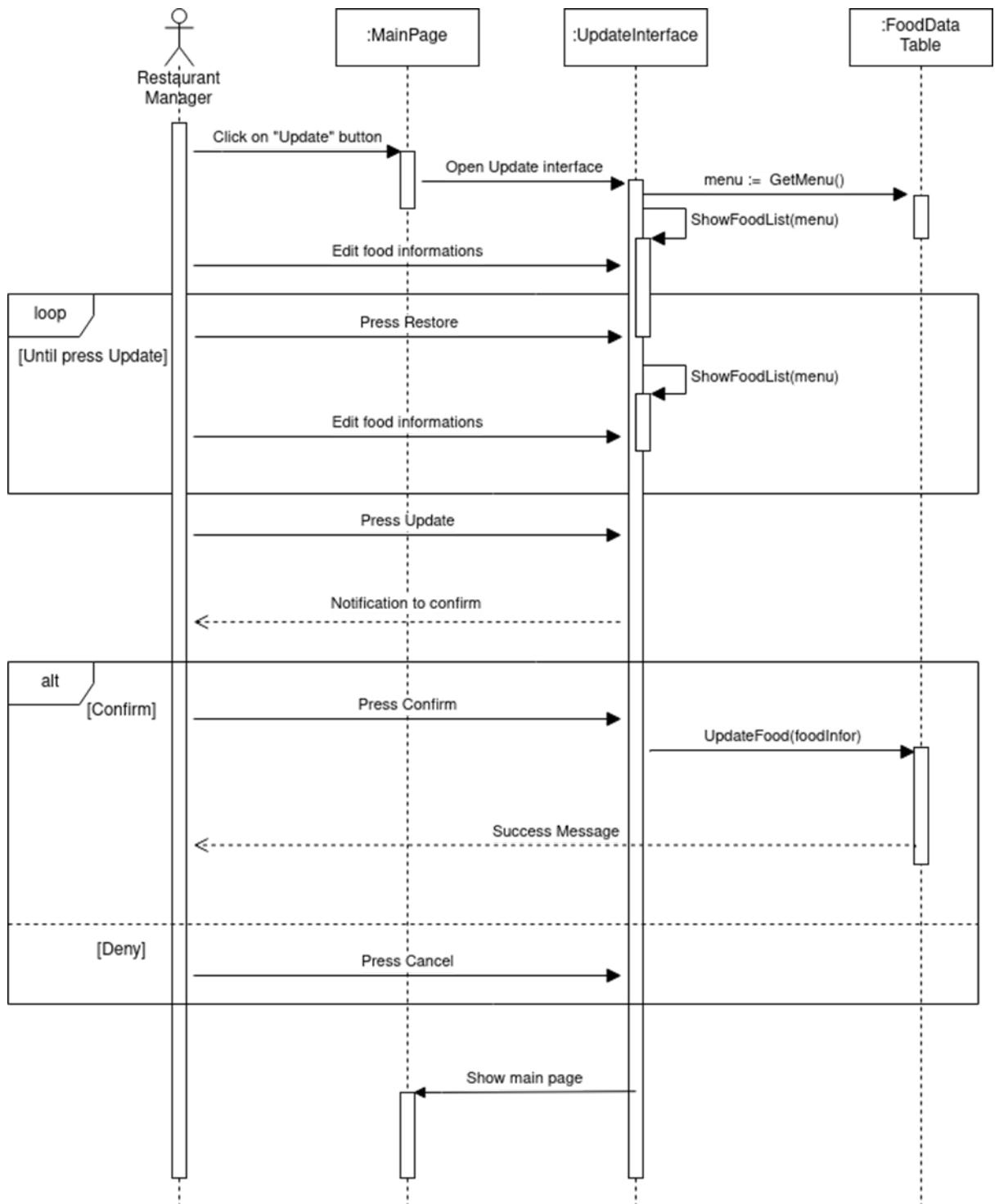


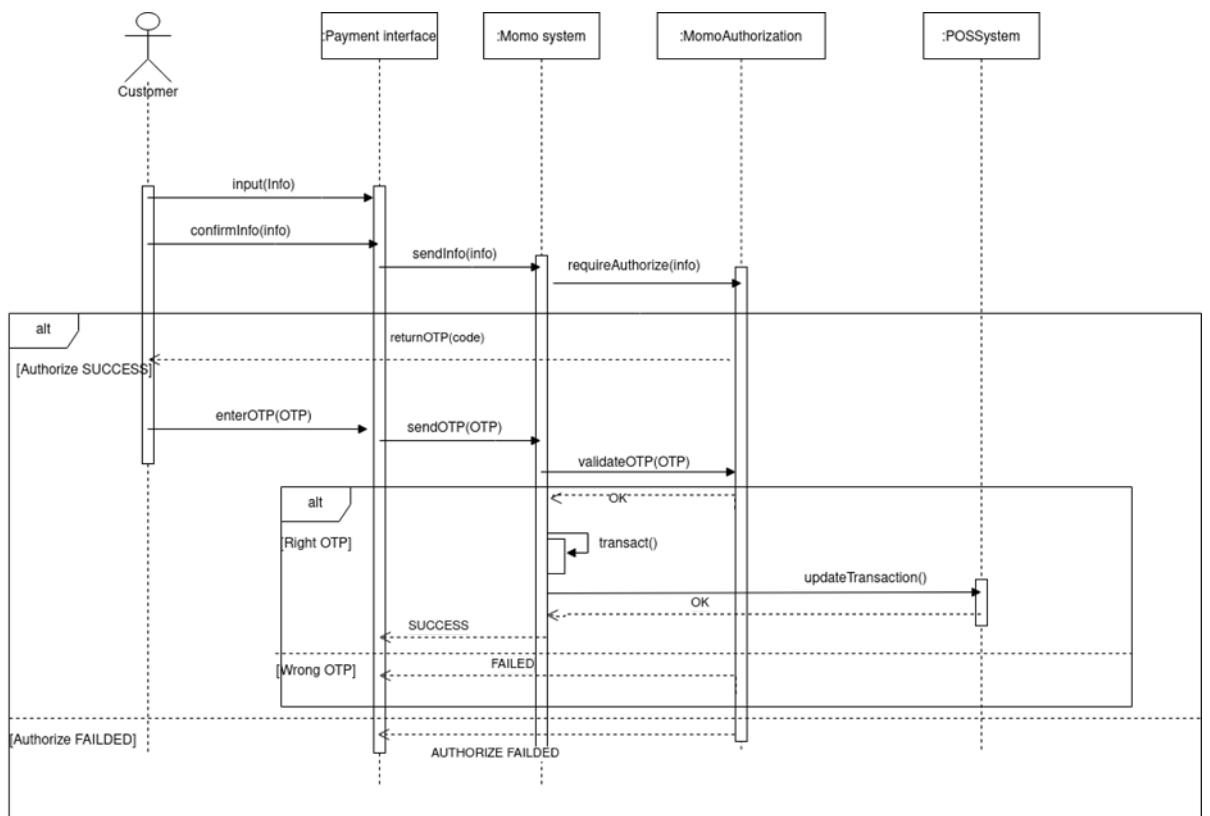
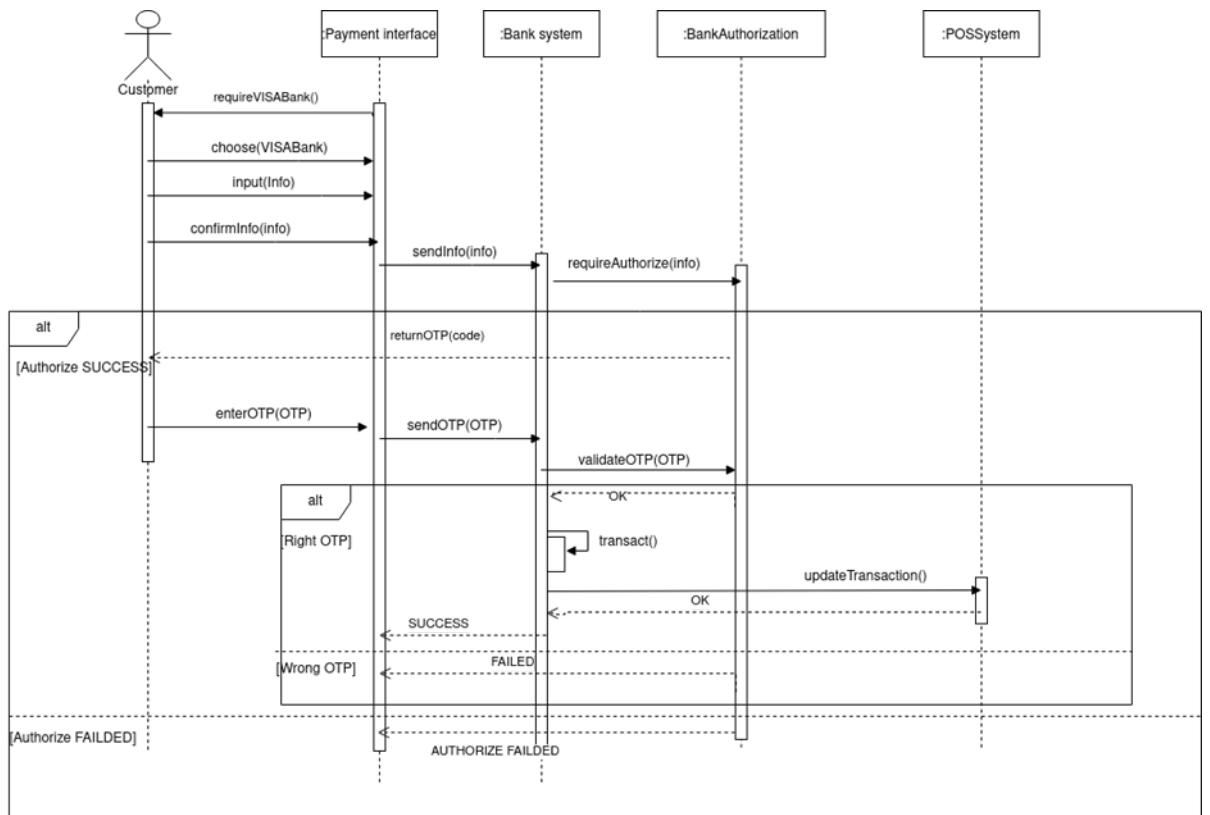






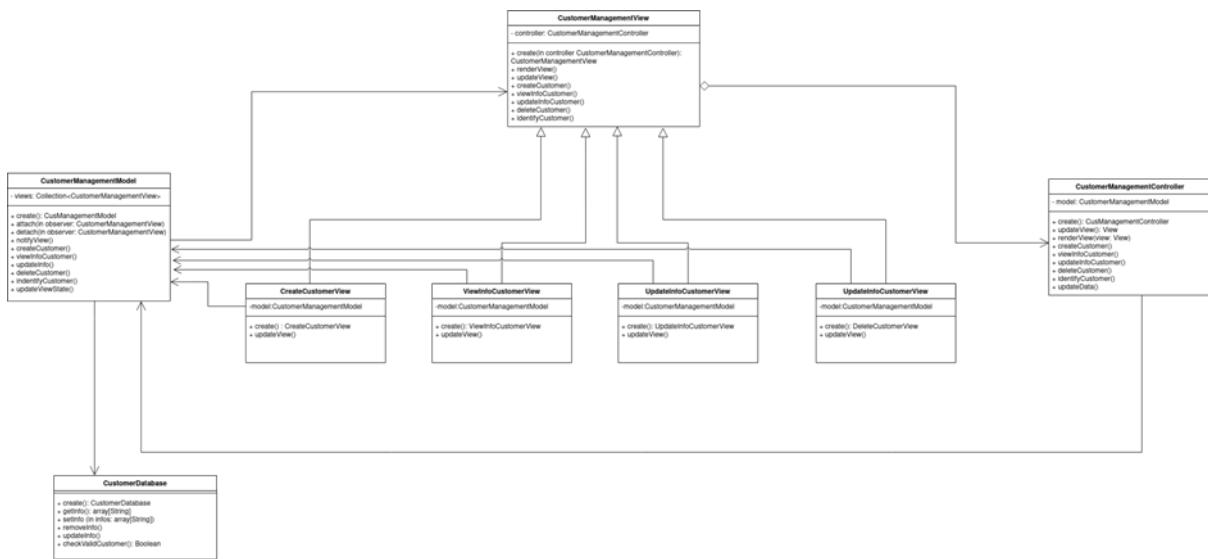
+ Feature: Online payment:



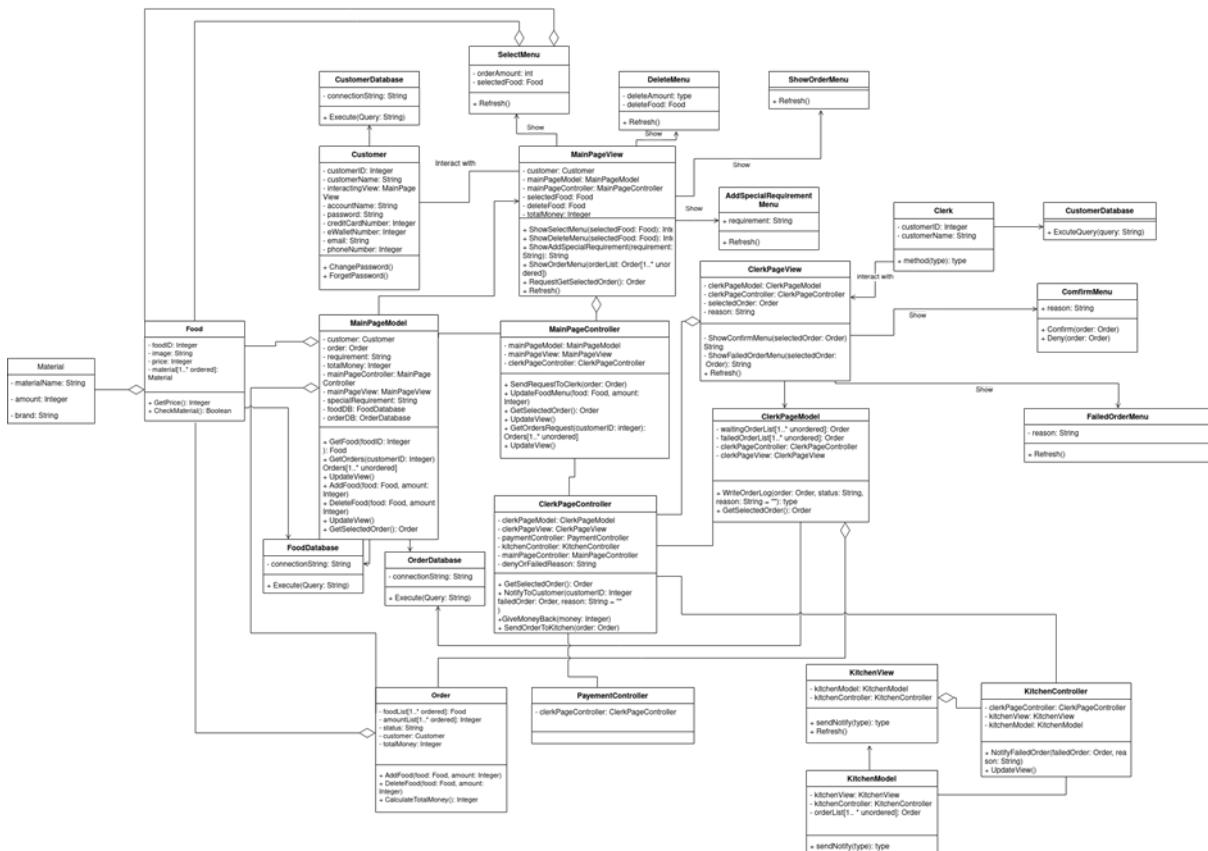


Task 2.3. Draw a class diagram.

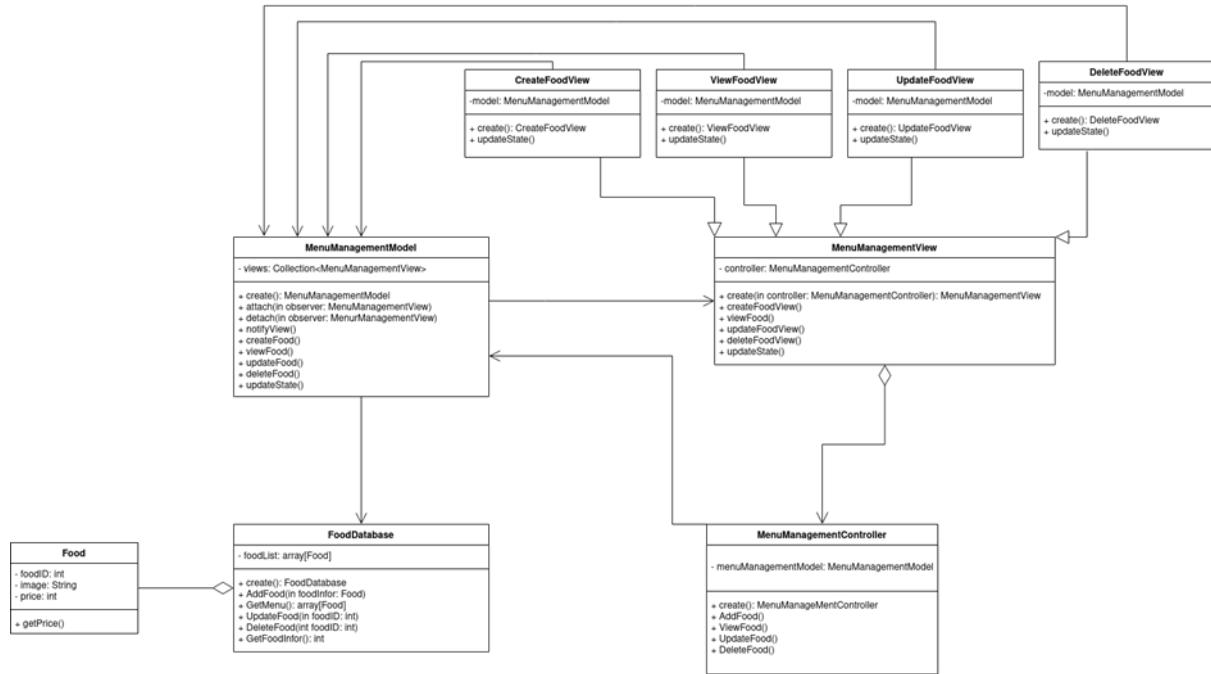
+ Feature: Customer management



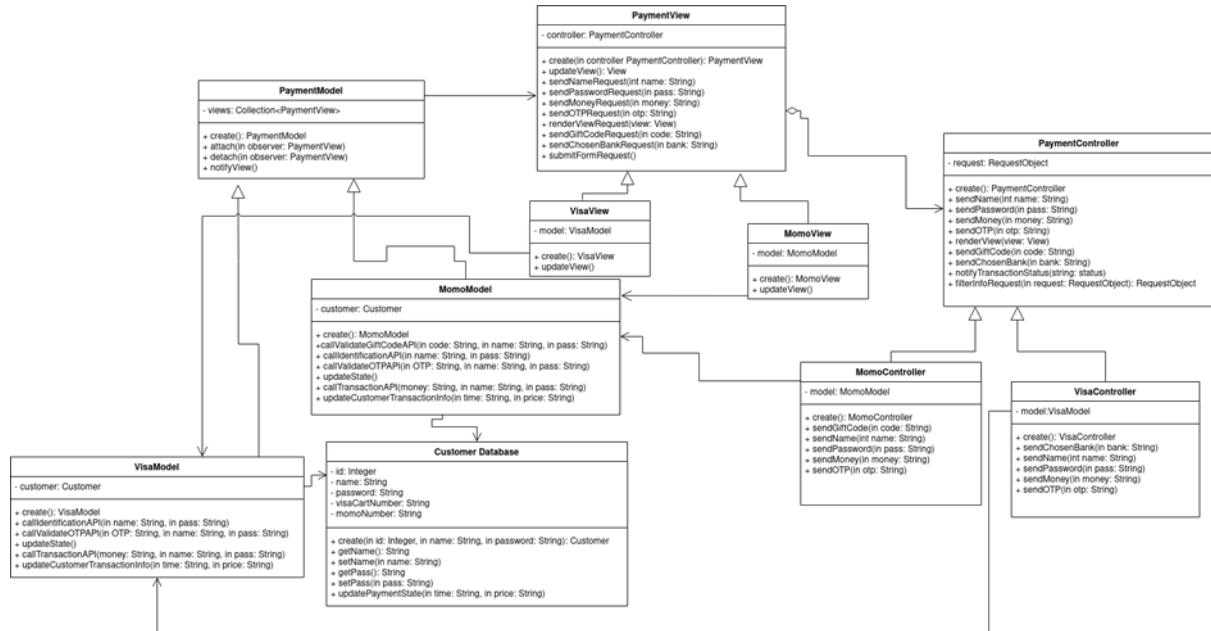
+ Feature: Food ordering



+ Feature: Menu management



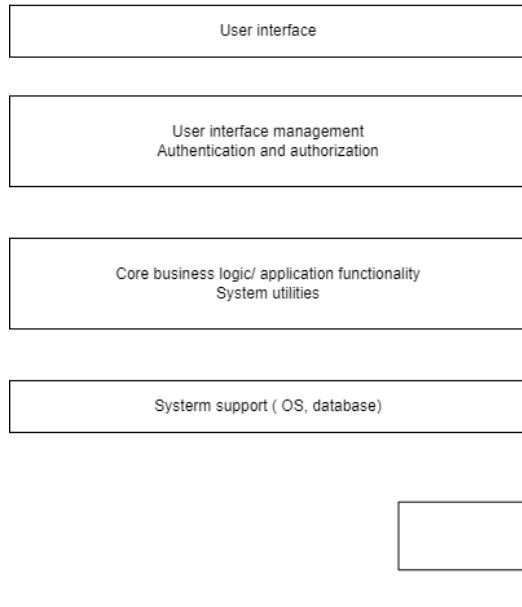
+ Feature: Online payment



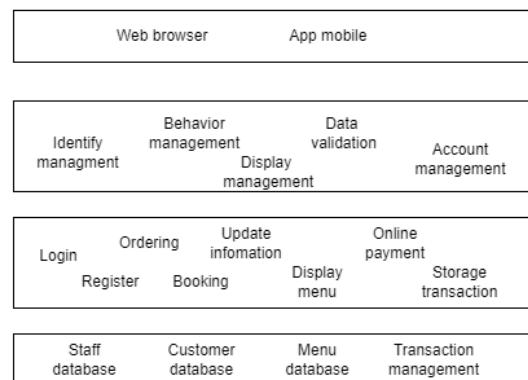
3. Task 3. Architecture Design

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

A generic layered architecture



The architecture of the POS system

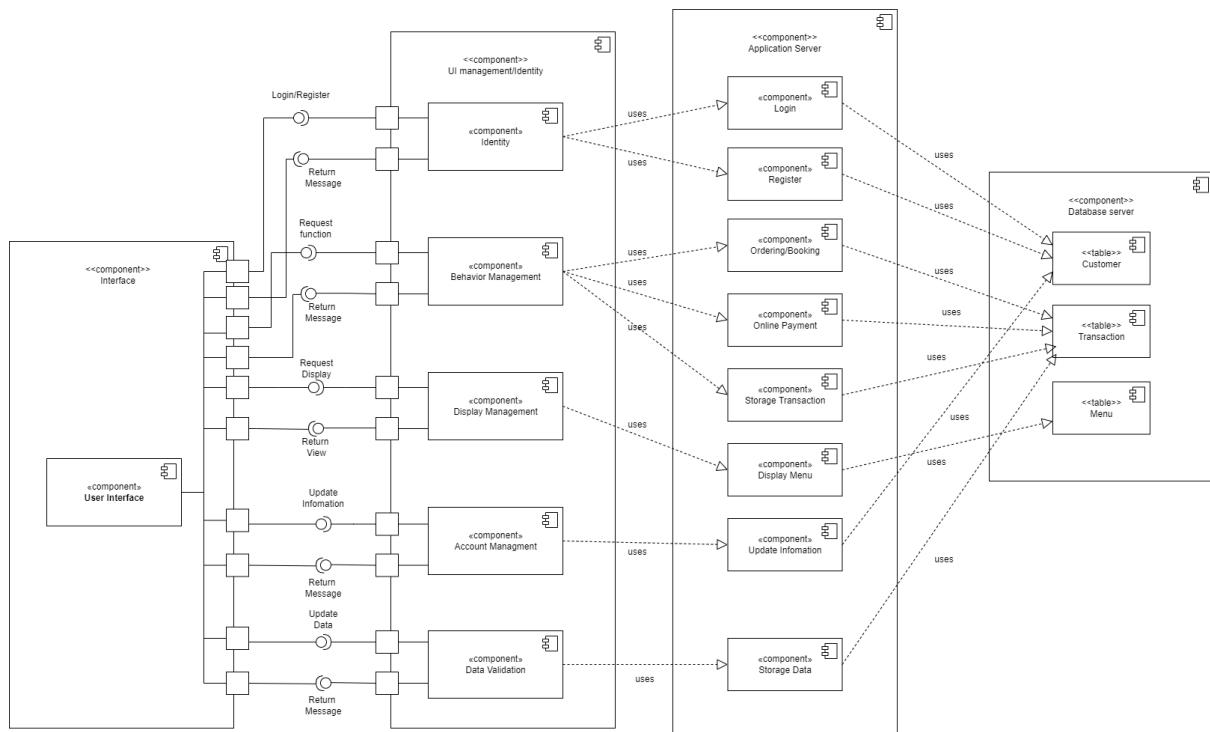


Name	The layered architecture pattern
Description	Tổ chức hệ thống thành các lớp với chức năng liên quan được liên kết với mỗi lớp. Một lớp cung cấp các dịch vụ cho lớp phía trên nó nên các lớp cấp thấp nhất sẽ đại diện cho các dịch vụ cốt lõi có khả năng được sử dụng trong toàn hệ thống. User interface: hiển thị hệ thống bằng đồ họa để người dùng thao tác các chức năng.

	<p>kiện ở trong view trên các trang:</p> <ul style="list-style-type: none"> ○ Hiển thị Dropdown menu khi xử lý sự kiện click vào menu của user. ○ Hiển thị slide tự động <ul style="list-style-type: none"> ● Gửi các tín hiệu xử lý sự kiện cho lớp tiếp theo. ● Truyền data vào các request của người dùng và gửi cho lớp tiếp theo ● Khi được thông báo các thay đổi trạng thái từ lớp dưới, thực hiện truy xuất dữ liệu để thực hiện refresh lại view và update view mới. <p>User interface management/ authentication and authorization: nhận các request cùng data từ lớp phía trên. Sau đó kiểm tra và xác nhận yêu cầu của request. Từ đó truyền request và data xuống lớp tiếp theo để gọi function tương ứng và thao tác với dữ liệu đó. Cuối cùng thông báo cho lớp trên khi có sự thay đổi trạng thái cùng với truyền dữ liệu đã được xử lý tương ứng với view được đề xuất.</p> <p>Core business logic/ application functionality</p> <p>System utilities:</p> <ul style="list-style-type: none"> ● Trả về response từ các request được gửi tới. ● Gọi các hàm tương ứng để thực hiện các thao tác trên dữ liệu: <ul style="list-style-type: none"> ○ CRUD: món ăn, đơn hàng, thông tin khách hàng, giao dịch thanh toán. ○ Sử dụng các phương thức để thực hiện các thao tác đăng ký, đăng nhập, quản lý tài khoản người dùng. ● Cung cấp các phương thức tương tác với database. ● Cung cấp các phương thức để xử lý gọi API để thực hiện xác thực cho việc thực hiện thanh toán với các hệ thống thanh toán. <p>System support (OS, database):</p> <ul style="list-style-type: none"> ● Cung cấp môi trường để vận hành ứng dụng. ● Cơ sở dữ liệu để lưu trữ và truy vấn dữ liệu.
Advantages	<ul style="list-style-type: none"> ● Dễ dàng thay thế toàn bộ các lớp miễn sao giao diện được duy trì. ● Có thể xây dựng các cơ sở dự phòng ở mỗi lớp để tăng độ tin cậy cũng như an toàn cho hệ thống. ● Quản lý và quản trị rủi ro đơn giản do mỗi lớp có các module riêng và được phân chia rõ ràng.
Disadvantages	<ul style="list-style-type: none"> ● Việc cung cấp sự phân tách rõ ràng giữa các lớp thường rất khó và một lớp cấp cao có thể phải tương tác trực tiếp với các lớp cấp thấp hơn thay vì thông qua lớp

- ngay bên dưới nó.
- Hiệu năng là một vấn đề do hiệu suất giữa các lớp khác nhau là khác nhau do sự yêu cầu xử lý khác nhau.
 - Cần đảm bảo tương tác hiệu quả giữa các lớp.

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



Mô tả component diagram: Tập trung vào các component liên quan đến khách hàng

- + Hệ thống có 4 component lớn tương đương với 4 layer đã đặc tả
 - Interface Component: Đảm nhận các nhiệm vụ như đã được mô tả trong task 3.1. Chứa UI mà khách hàng tương tác trực tiếp.
 - UI Management Component: Đảm nhận các nhiệm vụ như đã được mô tả trong task 3.1. Chứa các component dùng để quản lý giao diện của người dùng. Từ đó sẽ gửi request cùng với data theo yêu cầu của người dùng xuống lớp tiếp theo để gọi các hàm tương ứng.
 - Application Server: Đảm nhận các nhiệm vụ như đã được mô tả trong task 3.1. Chứa các web service, các hàm dùng để xử lý request có data của người dùng và reponse về cho lớp UI Management Component. Các component ở lớp này phụ thuộc trực tiếp vào component ở lớp trước.

- Database Server: Đảm nhận các nhiệm vụ như đã được mô tả trong task 3.1. Các component ở lớp này là các bảng dữ liệu, cung cấp công cụ thao tác sửa đổi, lưu trữ và truy xuất dữ liệu của hệ thống. Các bảng này cũng phụ thuộc trực tiếp vào các component của lớp trước.

4. Task 4. Implementation - Sprint 1

Task 4.1. Setting up. The team creates an online repository (github, bitbucket, etc) for version control.

```
commit cdcc04bda77b55462f75173715339da8b05c35cc
Merge: d65501c 9216800
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Wed Apr 13 23:42:44 2022 +0700
```

Merge Nguyen Commit

Update App.css ??

```
commit 9216800c0284dcf931f9274f18af01af558f9f65
Author: nhatnguyen1101
<58659298+nhatnguyen1101@users.noreply.github.com>
Date:   Wed Apr 13 23:38:04 2022 +0700
```

Update App.css

```
commit d65501c2dca5e649462ab9986558cb0f23705241
Merge: b8e7ef7 8042a2b
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Wed Apr 13 23:19:53 2022 +0700
```

Merge Khai Commit

Khai branch

```
commit 8042a2bf7d03b7389561e2474f7819b72c934d07
Author: KhaiHuynh-111 <khai.huynhmt1805@hcmut.edu.vn>
Date:   Wed Apr 13 23:15:16 2022 +0700
```

Fix index.css

```
commit 145e64766eeb00e148a52a8540c5592d36554815
Author: KhaiHuynh-111 <khai.huynhmt1805@hcmut.edu.vn>
Date:   Wed Apr 13 23:13:39 2022 +0700
```

Fix index.js

```
commit b8e7ef74f30528542ba7917426409a5f3317fc64
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Wed Apr 13 23:06:50 2022 +0700
```

Delete index.html

```
commit 1ca9bbc3dd6494bd5a3efd250879825b1458d8e9
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Wed Apr 13 14:49:35 2022 +0700
```

add data

```
commit a2fa73c669d3e58c4c190fa1916a795f22bb0fb1
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Tue Apr 12 07:40:41 2022 +0700
```

button click

```
commit 2e51fdbbe3215864d1ac522d8680166f7a027d0d6
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Mon Apr 11 17:57:34 2022 +0700
```

filter

```
commit ed3da5ff328bc16598b042bbdd65e5349affaba2
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Mon Apr 11 11:32:48 2022 +0700
```

slider

```
commit faelcabd44499e8b6ab991850ca60f112d96ffff
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Fri Apr 8 17:30:20 2022 +0700
```

okasda

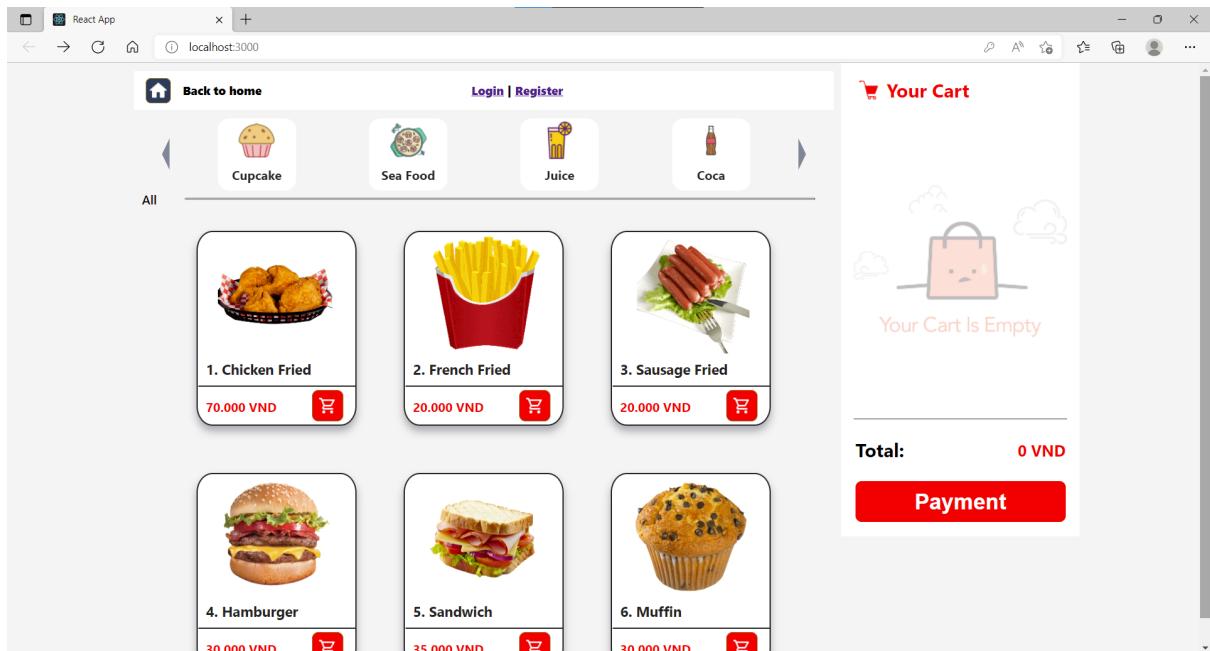
```
commit af32c866781467bc435342007e59eb180b81ffef
Author: datnt165 <93573049+datnt165@users.noreply.github.com>
Date:   Fri Apr 8 17:24:28 2022 +0700
```

First commit

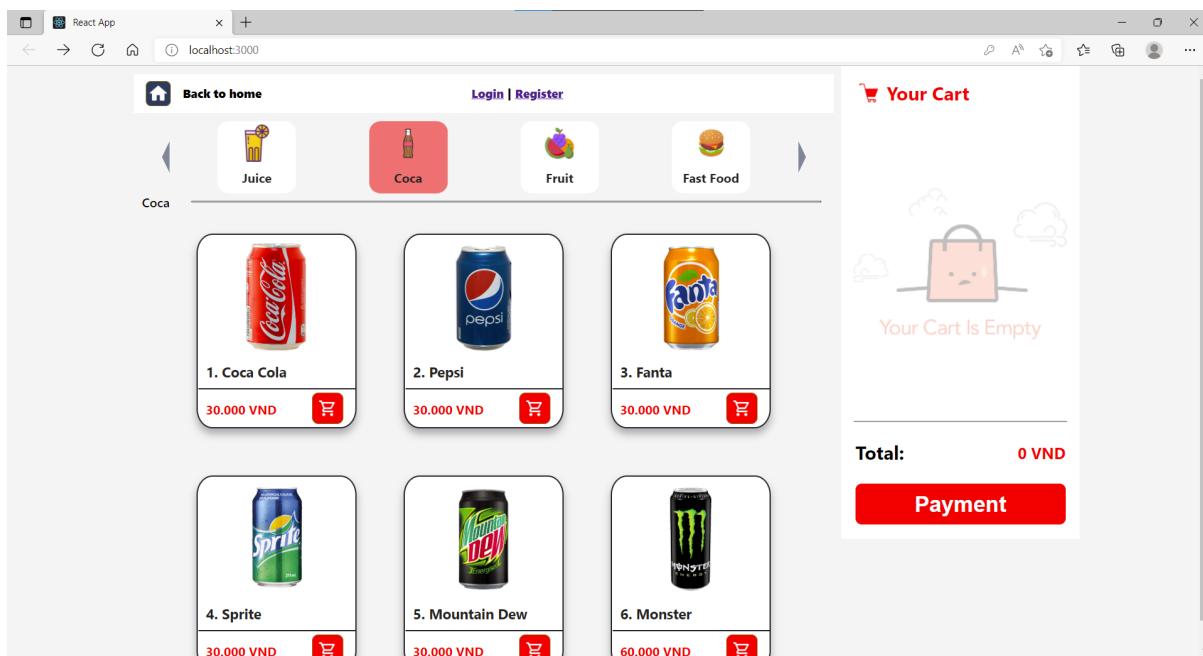
Task 4.2. Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files/folders

Link github: <https://github.com/datnt165/MVC-Test>

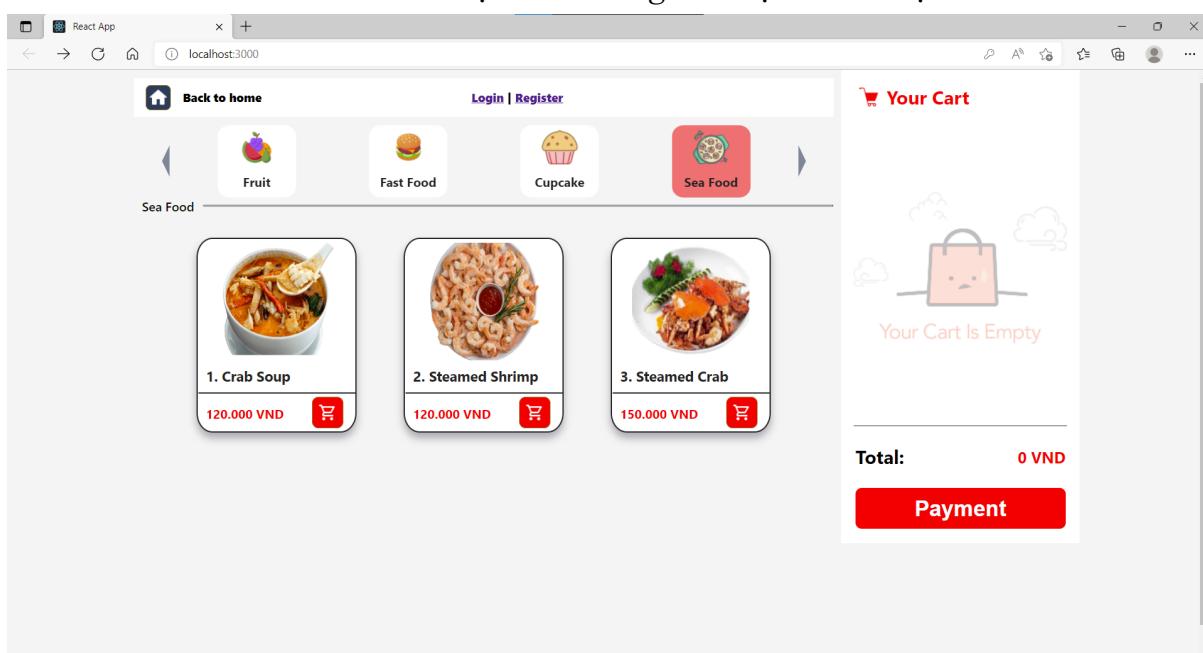
Task 4.3. Implement a Minimum Viable Product (MVP) for the menu screen in Figure 2 and demonstrate the result. MVP means that do the least to be able to demonstrate. That means at this stage, no need for a database to store all menu items, customers, etc. Data can be hard coded in code files.



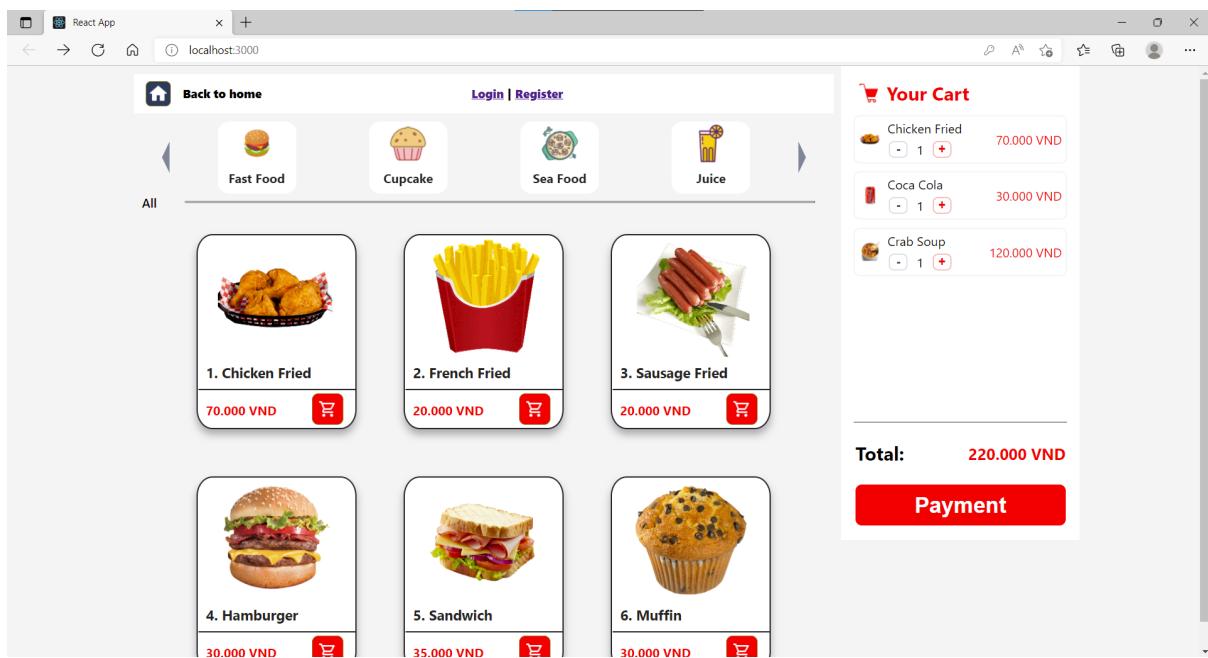
Hình 1: Menu chưa filter Danh mục



Hình 2: Menu hiển thị các đồ uống khi chọn Danh mục Coca



Hình 3: Menu hiển thị các món ăn liên quan đến Danh mục Sea Food



Hình 4: Chọn các món ăn mong muốn vào giỏ hàng.

5. Task 5. Implementation - Sprint 2

Task 5.1. Implement Figure 2

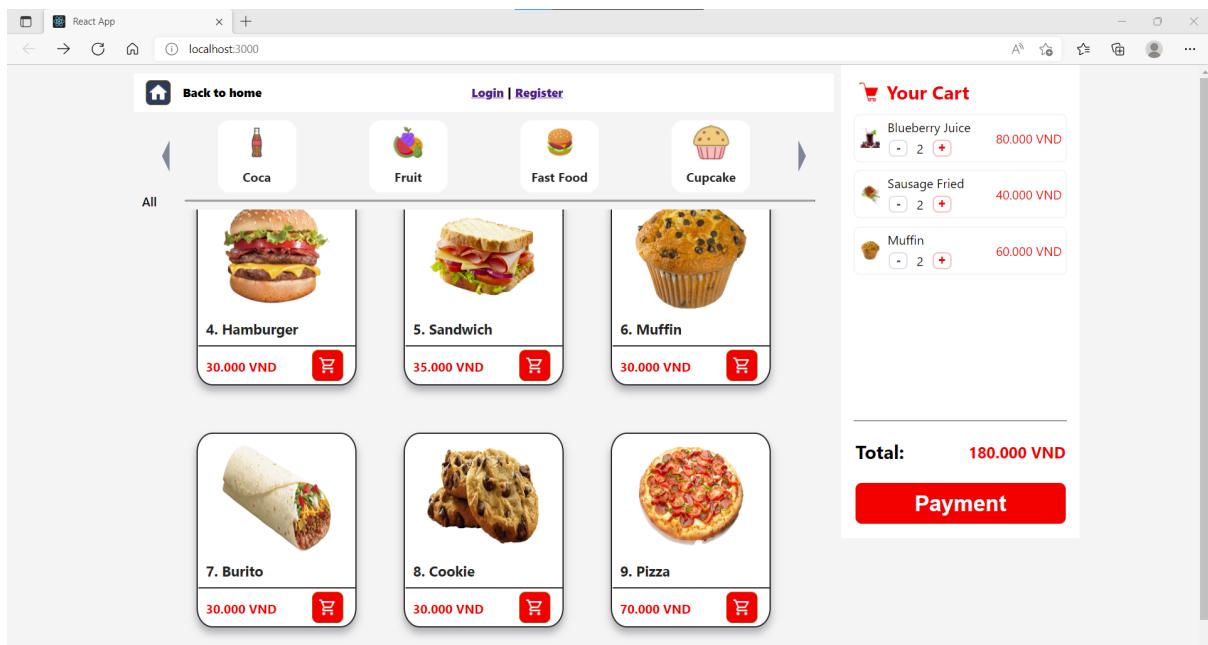
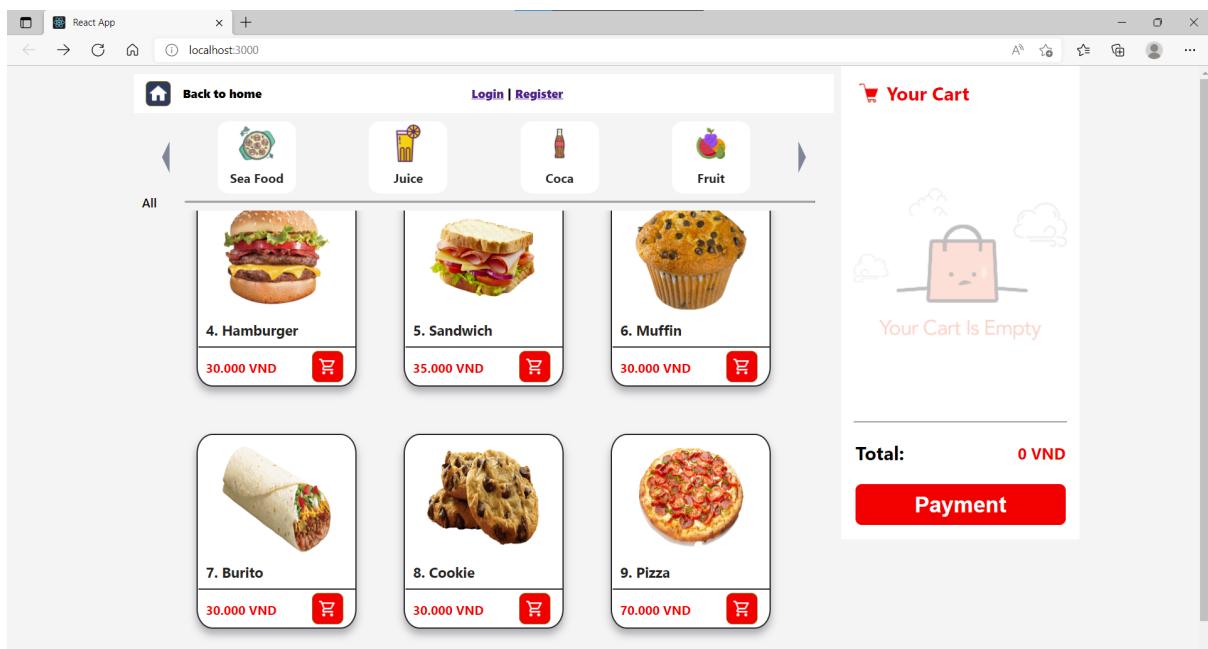
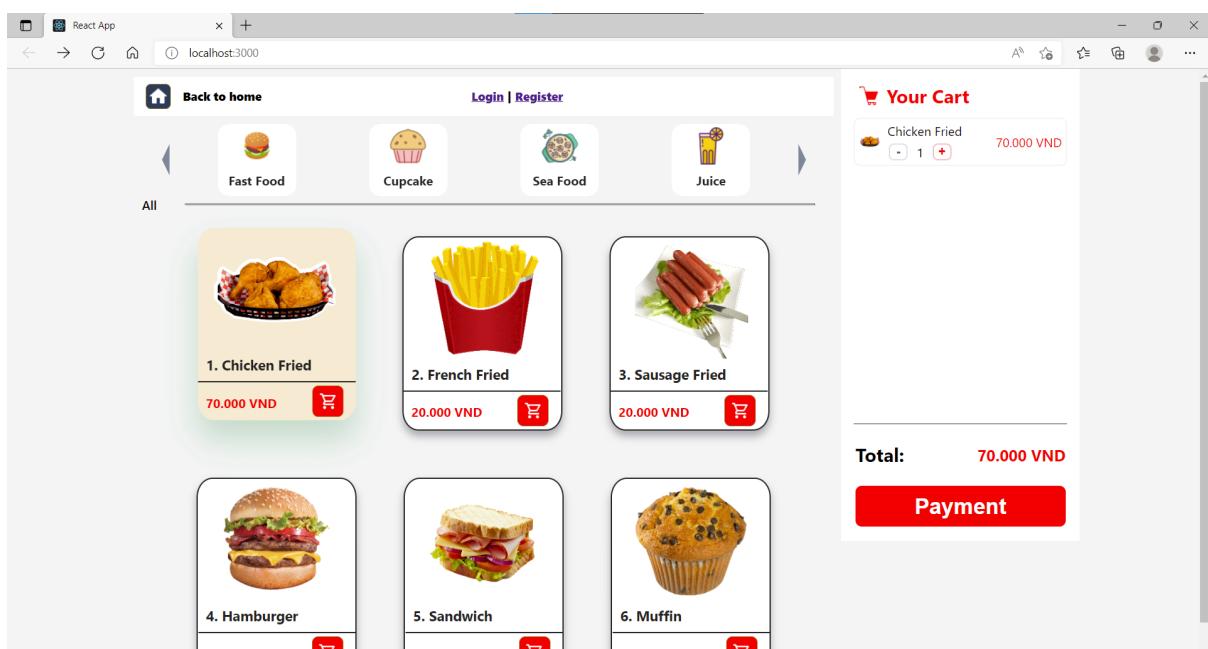


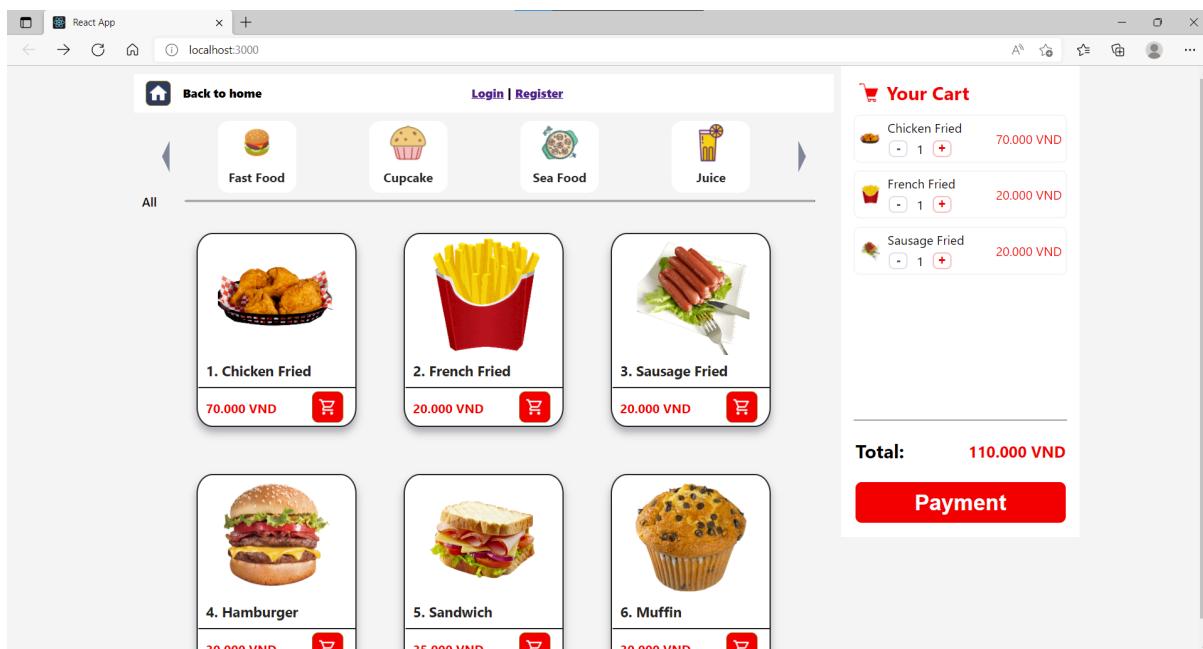
Figure 2. The menu screen for take-away or dine-in customers.



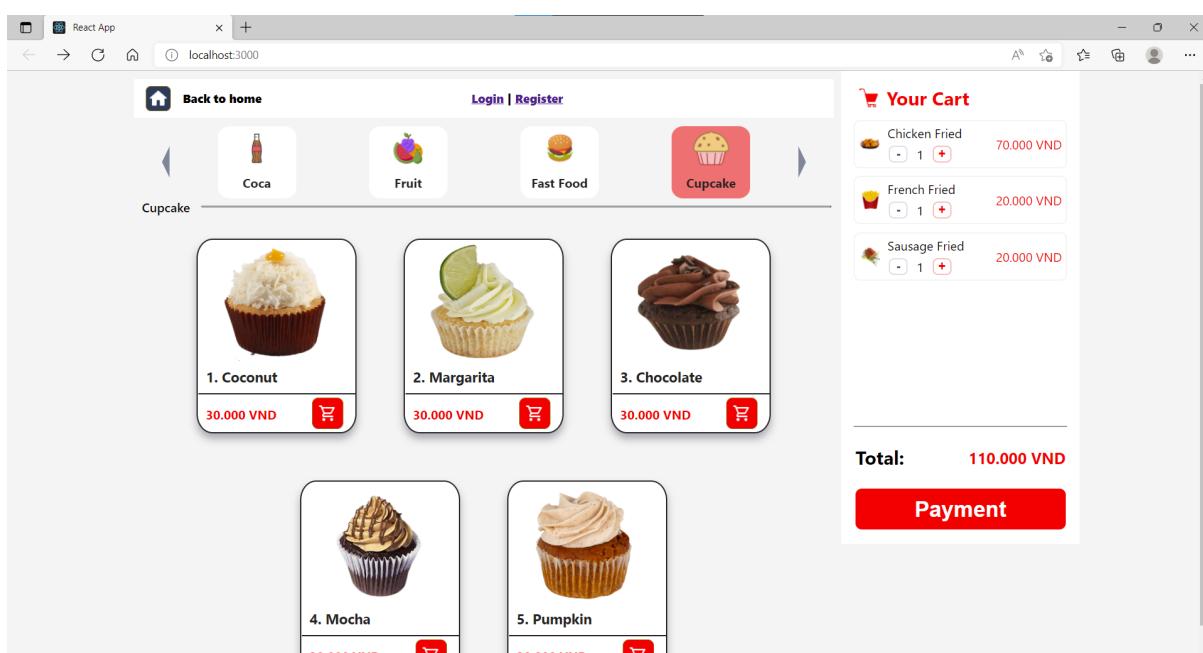
Hình 5.1.1. Giao diện thực đơn khi chưa thêm gì vào giỏ hàng



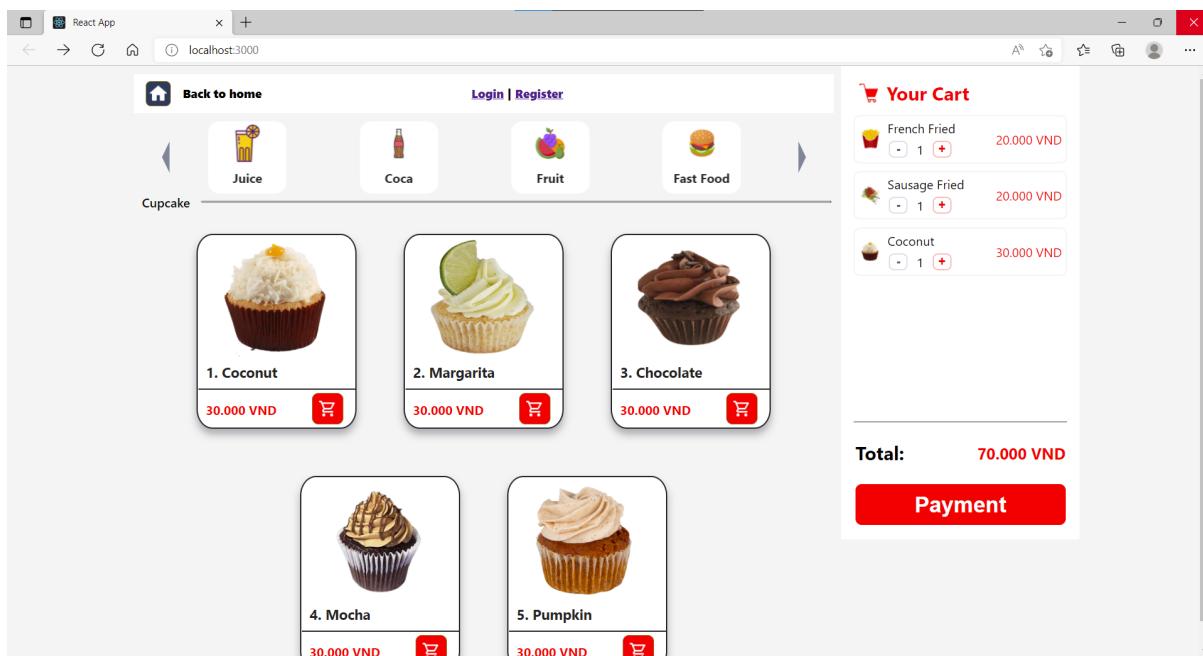
Hình 5.1.2. Giao diện thực đơn khi đã thêm 1 món vào giỏ hàng



Hình 5.1.3. Giao diện thực đơn khi đã thêm một số món mới vào giỏ hàng



Hình 5.1.4. Giao diện thực đơn khi filter món ăn theo ý thích của khách hàng



Hình 5.1.5. Giao diện thực đơn khi xóa món ăn đầu tiên chọn và thêm món ăn theo ý thích của khách hàng ở mục filter.

Task 5.2. Implement Figure 3 và 4

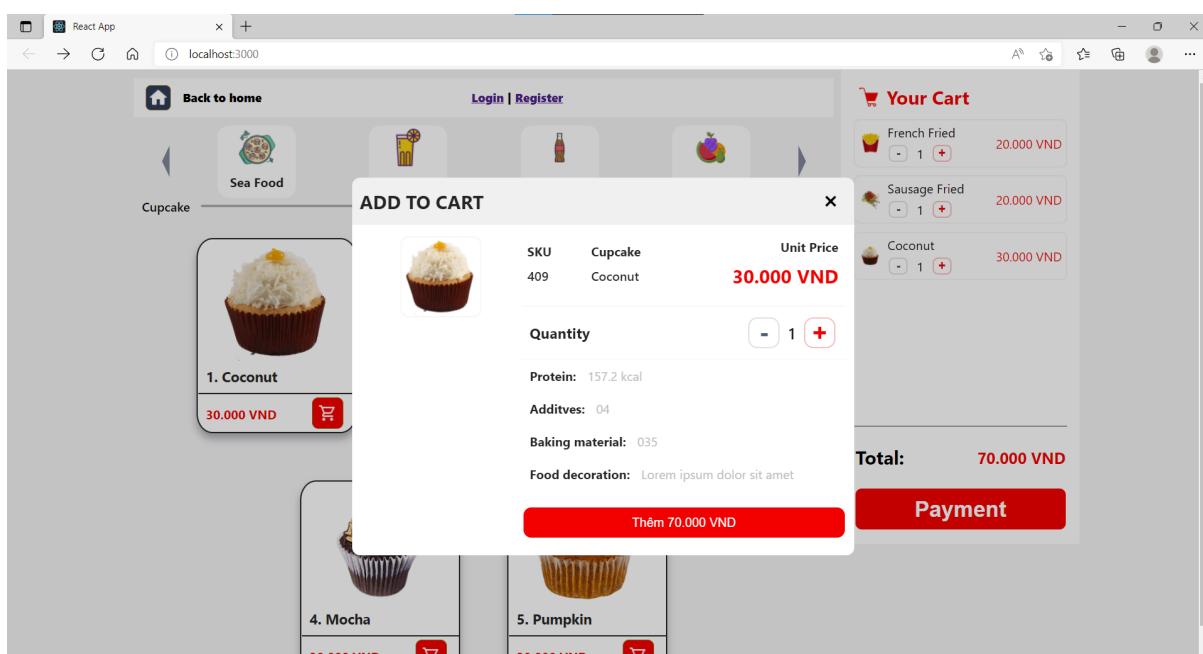
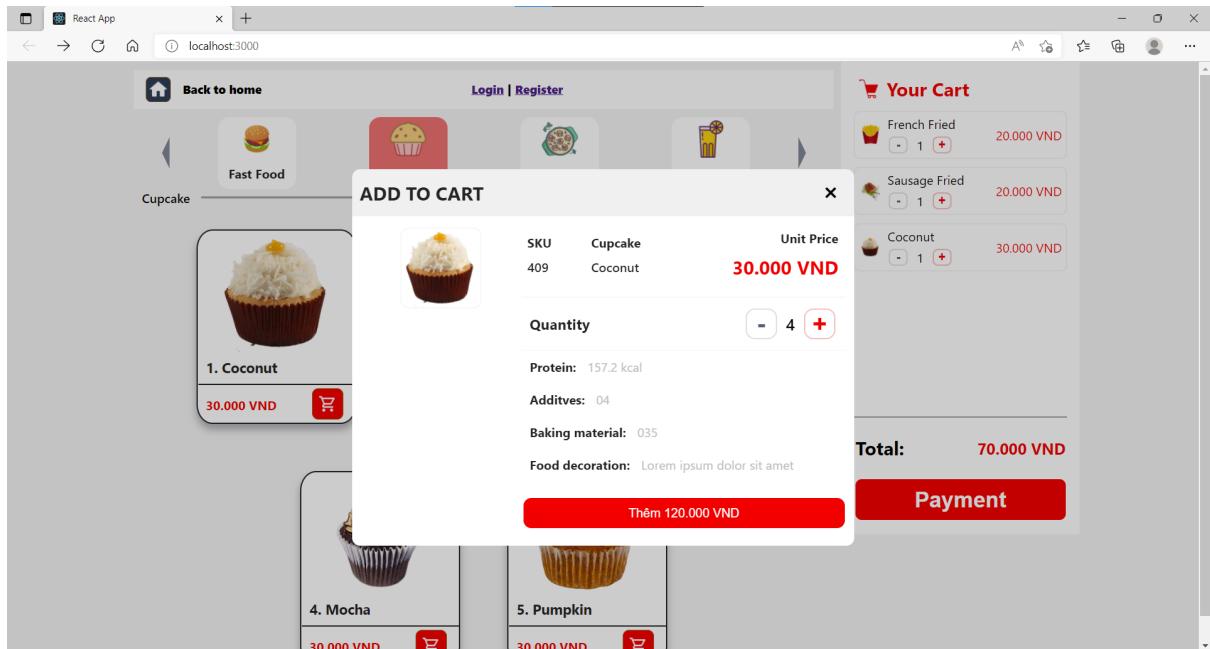
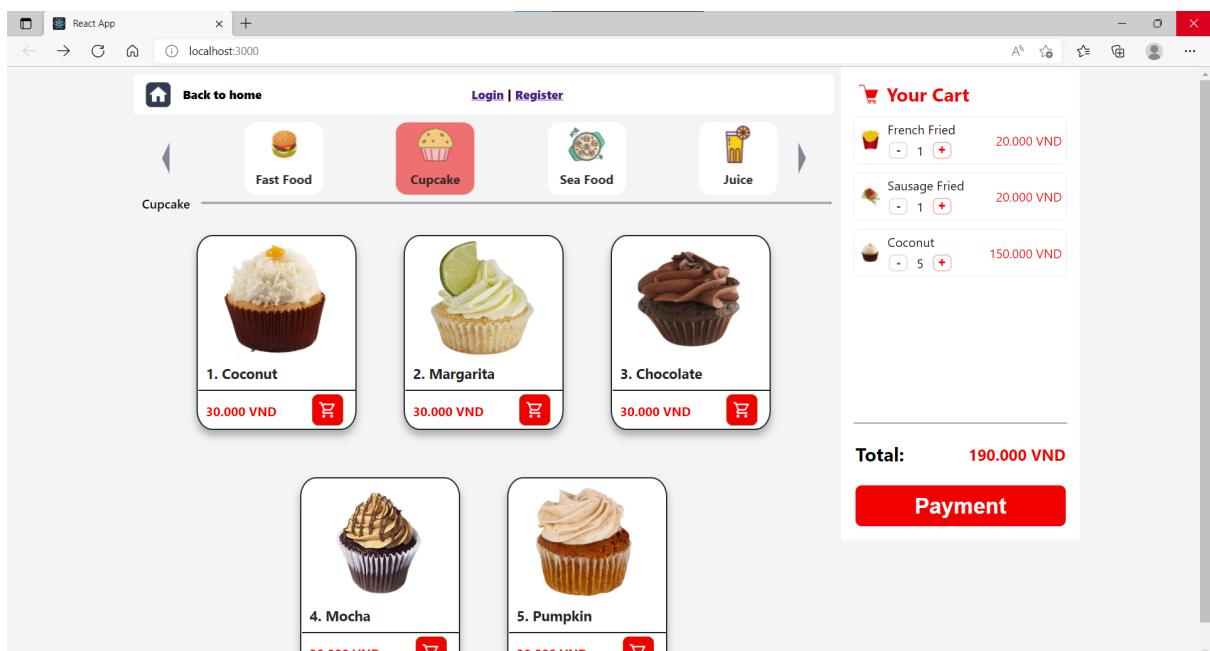


Figure 3: The detail item screen when clicking to select an item in Figure 2. When changing the Quantity or click on the shopping button, returning to the Menu screen and update the sum-up Your Cart in the right hand side



Hình 5.2.1. Giao diện figure 3 khi đã tăng số lượng



Hình 5.2.2. Giao diện menu khi bấm “thêm” ở Pop up thông tin thức ăn.

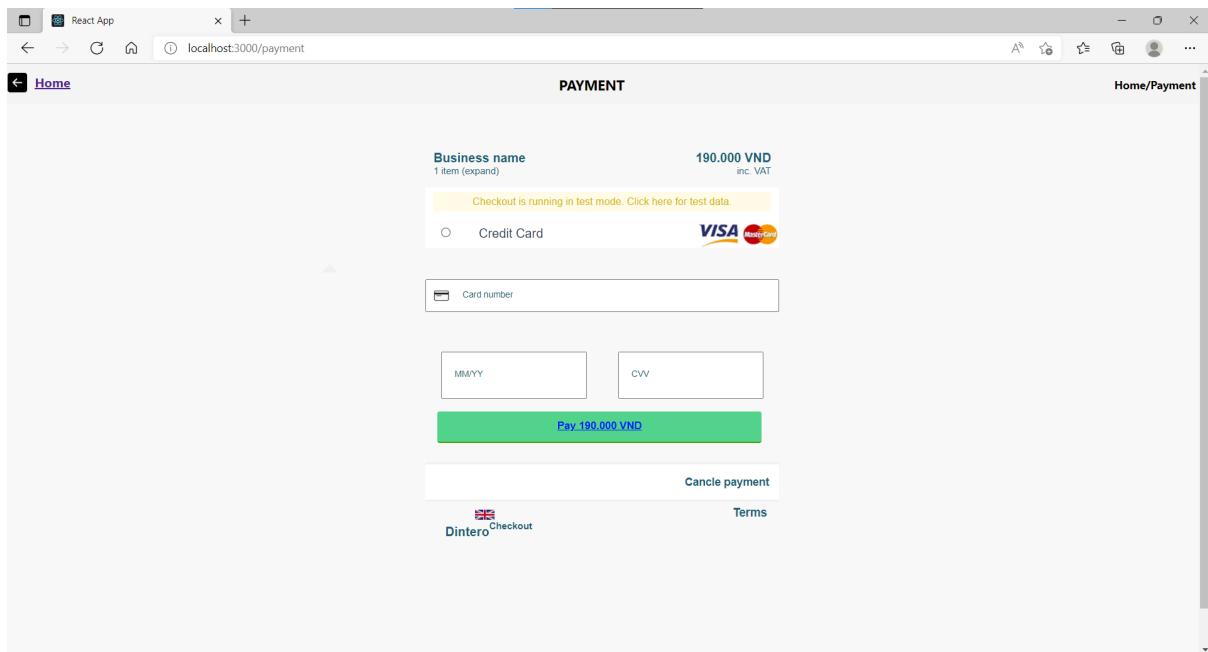


Figure 4: The payment page when clicking on the button Payment in Figure 2

Task 5.3. Test Case for Implementation

Black-box testing: kiểm tra các requirement

Total: 11 testcase

Passed: 10

Not Passed: 1

ID	Name	Test case description	Precondition	Step name	Step description	Expected Result	Microsoft Edge	
							Step	Case
1.1	Food 1.1	Thêm thức ăn vào giỏ hàng tại icon cart của món ăn khi giỏ hàng	Menu hiển thị các món ăn và giỏ hàng trống	Step 1	Click vào icon cart tại món ăn mà muốn thêm vào giỏ hàng	Món ăn được thêm vào giỏ hàng bên phải	Passed	Passed

		trống						
1.2	Food 1.2	Thêm món ăn vào giỏ hàng trống (tại modal xem chi tiết món ăn)	Menu hiển thị các món ăn và giỏ hàng trống	Step 1	Click vào món ăn muốn thêm	Hiển thị cửa sổ nhỏ chứa thông tin chi tiết của món ăn	Passed	Passed
				Step 2	Nhấn vào dấu “+” “-” để tăng giảm số lượng phần của món ăn muốn thêm vào giỏ hàng	Số lượng của món ăn đó được tăng giảm đúng theo mong muốn của khách hàng	Passed	
				Step 3	Nhấn button thêm	Món ăn được thêm vào giỏ hàng đúng với số lượng đã chỉnh	Passed	
2.1	Cart 2.1	Chỉnh sửa thông tin giỏ hàng (số lượng trực tiếp trên giỏ hàng)	Món ăn được thêm vào giỏ	Step 1	Nhấn dấu “+” “-” để chỉnh sửa số lượng mua	Số lượng được tăng giảm đúng theo mong muốn	Passed	
2.2	Cart 2.2	Xóa món ăn trên giỏ hàng	Món ăn đã được thêm vào giỏ hàng	Step 1	Nhấn dấu “-” tới khi món ăn đó không còn hiện trên giỏ hàng nữa	Món ăn biến mất khỏi giỏ hàng	Passed	Passed
3.1	Button 3.1	Kiểm tra tính inactive của nút nhấn cart trên các món ăn	Menu hiển thị các món ăn	Step 1	Nhấn icon “cart” trên các món ăn	Món ăn được thêm hoặc tăng số lượng trên giỏ hàng	Passed	Passed
4.1	Function 4.1	Di chuyển slider danh mục để	Slider hiển thị các danh mục và 2 mũi tên ở 2 bên	Step 1	Nhấn mũi tên bên phải	Các danh mục di chuyển sang trái, danh mục ngoài cùng bên trái bị ẩn và có 1 danh mục mới bên	Passed	Passed

		tìm kiếm				phải		
				Step 2	Nhấp mũi tên bên trái	Các danh mục di chuyển sang phải. Danh mục bên trái ngoài cùng ở Step 1 bị biến mất thì bây giờ lại xuất hiện ở chỗ cũ và Danh mục mới xuất hiện ngoài cùng bên phải ở Step 1 biến mất	Passed	
4.2	Function 4.2	Hiển thị các món ăn của danh mục bất kỳ	Các danh mục được hiển thị trên Slider	Step 1	Chọn Danh mục món ăn mình mong muốn	Hiển thị các món ăn thuộc Danh mục món ăn mà mình chọn	Passed	Passed
4.3	Function 4.3	Hiển thị chi tiết thông tin của món ăn bất kỳ	Các món ăn được hiển thị trên menu	Step 1	Chọn món ăn mình muốn biết thông tin chi tiết	Một cửa sổ nhỏ hiện lên với thông tin chi tiết của món ăn đó	Passed	Passed
4.4	Function 4.4	Chức năng xác thực đăng nhập	Hiển thị trang điền thông tin đăng nhập	Step 1	Nhấn vào Login trên thanh bar	Chuyển hướng đến trang đăng nhập	Passed	Passed
				Step 2	Điền Username và password	Thông tin được cập nhật vào form	Passed	
				Step 3	Nhấn nút Login	Hiển thị “ Username or Password is not correct! ” khi điền không đúng thông tin tài khoản	Passed	
4.5	Function 4.5	Chức năng đăng ký	Hiển thị trang điền thông tin đăng ký	Step 1	Nhấn vào Register trên thanh bar	Hiển thị trang điền thông tin tạo tài khoản	Passed	Not Passed
				Step 2	Điền	Các thông tin được	Passed	

					Username, Password và Confirm Password	điền vào form. Password hiển thị * để an toàn thông tin		
				Step 3	Nhấn Register trên thanh bar	Tạo tài khoản thành công	Not Run	
4.6	Function 4.6	Thanh toán đơn hàng	Đã thêm đầy đủ món ăn cần mua vào giỏ hàng	Step 1	Nhấn Payment	Chuyển tiếp tới trang payment	Passed	Passed