

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



Công nghệ phần mềm (CO3001)

RESTAURANT POS 2.0

GVHD: PGS.TS Quản Thành Thơ

SV thực hiện: Nguyễn Công Đoàn - 1910131
Dương Thanh Thương - 191218
Huỳnh Hữu Hạnh - 1910161
Nguyễn Lê Thanh Lộc - 1912871
Võ Minh Chánh - 1910057

Mục lục

1 Requirement elicitation	2
1.1 Identify the context of this project	2
1.2 Functional and non-functional requirements of the desired system. Use-case diagram for the whole system	3
1.3 Use-case diagram and description for features	5
2 System modeling	6
2.1 Activity diagram for functional requirements of the desired system	6
2.2 Sequence diagram for use-case	10
2.3 Class diagram	11
3 Architecture design	12
3.1 Describe an architectural approach	12
3.1.1 Mô hình MVC NodeJS-MySQL	12
3.1.2 Mô hình MVC cho hệ thống	13
3.1.3 Những công nghệ sử dụng	13
3.2 Implementation diagram for functional requirements	14
4 Task 4: Implementation - Sprint 1	14
4.1 Task 4.1	14
4.2 Task 4.2	14
4.3 Task 4.3	15
5 Task 5: Implementation - Sprint 2	16
Tài liệu	16

1 Requirement elicitation

1.1 Identify the context of this project

- **Ngữ cảnh của đề tài**

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. Ngay cả trước cuộc khủng hoảng COVID-19, hệ thống POS đã đạt được sức hút trong toàn ngành. Trong đại dịch coronavirus, các nhà hàng đối mặt với nguy cơ lây lan dịch bệnh trong môi trường làm việc lớn hơn bao giờ hết. Những hệ thống tự động được kỳ vọng sẽ giúp nhiều doanh nghiệp giãn cách và cắt giảm nhân sự, không tốn nhiều công sức và cơ hội để mở rộng quy mô thành một doanh nghiệp lớn. Một nhà hàng chuyên bán các loại thức ăn nhanh như hamburger, gà rán ... ở Việt Nam đã yêu cầu thiết kế một hệ thống Restaurant POS 2.0 tự động, hệ thống này cho phép nhân viên và khách hàng không cần phải trao đổi trực tiếp khi mua hàng. Khách hàng sẽ tự động mua và thanh toán sản phẩm bằng cách truy cập vào website của cửa hàng hoặc dùng QR code để đến website của cửa hàng, ngoài ra khách hàng còn có thể đưa ra phản hồi đánh giá cho sản phẩm. Website này có thể truy cập bằng nhiều thiết bị khác nhau. Số lượng giao dịch mà website có thể đáp ứng ít nhất 300 giao dịch mỗi ngày và website có thể mở sẵn sàng mở rộng thêm trong tương lai.

- **Những đối tượng liên quan**

1. Khách hàng: có thể đặt hàng, thanh toán, đánh giá các sản phẩm trên website.
2. Chủ nhà hàng và người quản lý hệ thống: có thể tùy chỉnh menu, giá sản phẩm, xem tổng số đơn và tiền hàng.
3. Nhân viên: có thể xem số đơn đã hoàn thành và số đơn còn chờ xử lý để thống kê giao người giao hàng.
4. Nhân viên nhà bếp: có thể xem số đơn đang cần xử lý và đánh dấu đã nhận đơn, cũng như hoàn thành xong.

- **Sau khi hoàn thành, hệ thống Restaurant POS 2.0 có thể**

1. Cho phép khách hàng đặt món và thanh toán thông qua website của nhà hàng được truy cập bằng QR code.
2. Nhân viên có thể thống kê và xử lý các đơn hàng.
3. Hệ thống có thể sử dụng được trên: thiết bị di động, máy tính bảng, máy tính/máy tính xách tay.
4. Cho phép thực hiện 300 giao dịch một ngày.

- **Phạm vi dự án**

Được xác định thông qua mục đích, các module sản phẩm mong muốn, các functional requirement và non-functional requirement, các tính năng sẽ thực hiện trong dự án, chi phí cũng như deadline của dự án. Cụ thể:

1. Mục đích: tạo ra website với các hành vi có thể giao tiếp trực tuyến một cách thuận tiện, tránh tiếp xúc trực tiếp nhiều nhất có thể giữa các bên ứng với bối cảnh đại dịch covid 19.
2. Loại cửa hàng: cửa hàng phục vụ thức ăn nhanh, chỉ mang về.
3. Khu vực của nhà hàng : Việt Nam.
4. Loại thiết bị có thể truy cập website : smartphone, laptop, tablet, PC.
5. Loại thiết bị có thể dùng QR code : smartphone, tablet.
6. Loại trình duyệt có thể sử dụng: chrome 14.0, edge 18.0, safari 10.1, firefox 20.0.
7. Phạm vi cho các tính năng của hệ thống
 - (a) Đặt món: chỉ đặt món khi có tài khoản.
 - (b) Đăng nhập/ đăng ký tài khoản: đăng ký sử dụng số điện thoại, tài khoản google, facebook.
 - (c) Xem lịch sử đơn hàng, và danh sách món order khi có tài khoản.

- (d) Phương thức thanh toán: có thể thanh toán trực tiếp khi nhận hàng hoặc thanh toán bằng ví điện tử momo.
- (e) Update thông tin website: chỉ có tài khoản quản lý.
- (f) Thống kê đơn hàng: xem đơn hàng trong vòng 3 tháng.

1.2 Functional and non-functional requirements of the desired system. Use-case diagram for the whole system

• Functional requirement

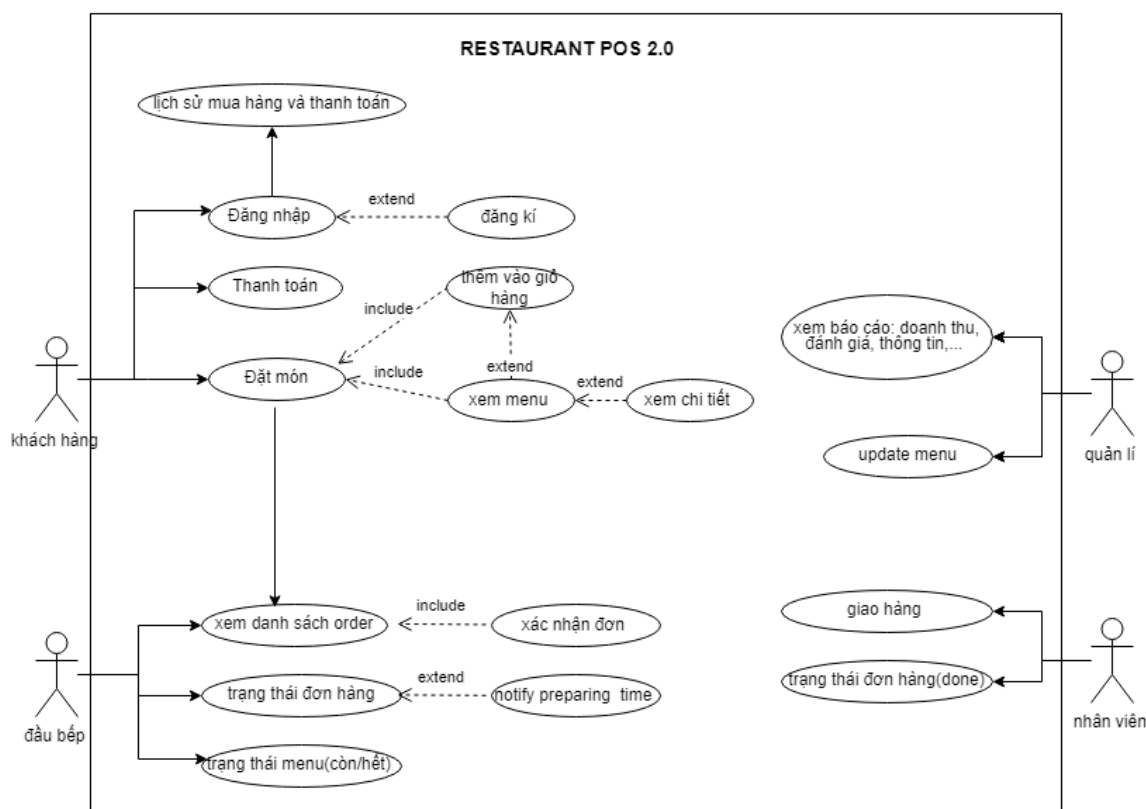
- Đối với khách hàng
 - * Đăng nhập
 - Chức năng đăng nhập với tư cách Guest.
 - Chức năng đăng ký tài khoản cho khách hàng.
 - Chức năng đăng nhập vào hệ thống (User).
 - Chức năng đăng xuất.
 - Chức năng tạo mật khẩu mới khi khách hàng quên mật khẩu: sử dụng email đã đăng kí để xác nhận và tạo mật khẩu mới.
 - Chức năng thay đổi thông tin cá nhân.
 - * Đặt món
 - Chức năng menu: cho phép người dùng xem tất cả các món và có thể lọc theo loại món ăn.
 - Chức năng xem thông tin chi tiết món ăn.
 - Chức năng giỏ hàng: cho phép xem và chỉnh sửa các món đã chọn.
 - Chức năng đơn hàng: xem được đơn hàng đã được tình trạng đơn hàng đã được xử lí hay chưa (set time cố định cho từng loại món ăn).
 - Chức năng hủy đơn hàng: cho phép hủy đơn khi hệ thống chưa xác nhận nhận đơn hàng.
 - Chức năng xem lịch sử đặt món và lịch sử thanh toán.
 - * Thanh toán: khách hàng có thể thanh toán dễ dàng bằng 2 phương thức đã đề cập ở trên.
- Đối với quản lý và nhân viên
 - * Đối với quản lý
 - Xem báo cáo: chức năng xem được thống kê đơn hàng trong tháng, chức năng xem được thông tin khách hàng, chức năng quản lý tài khoản và quyền truy cập của nhân viên, chức năng xem được phản hồi và đánh giá của khách hàng.
 - Chỉnh sửa và cập nhật menu: chức năng thêm, sửa, xóa các món ăn trong menu.
 - * Đối với nhân viên giao hàng
 - Giao hàng: xem số đơn nhà bếp đã hoàn thành và giao cho khách hàng, shipper.
 - Cập nhật trạng thái đơn hàng (done): sau khi giao hàng, cập nhật trạng thái đơn hàng thành đã giao (done).
 - * Đối với nhân viên nhà bếp
 - Xem danh sách đơn hàng đang chờ chế biến (to-do).
 - Cập nhật trạng thái đơn hàng (cook-done).
 - Cập nhật trạng thái món ăn trong menu (available).

• Non-functional Requirements

- Hệ thống phải có thể sử dụng bằng thiết bị điện thoại thông minh, máy tính bảng và máy tính cá nhân.
- Giao dịch đáp ứng được khoảng 300 đơn/ngày (mở rộng 1000 đơn trên ngày).
- Thời gian hệ thống phản hồi nhanh (delay nhỏ hơn hoặc bằng 5s).
- Hệ thống hoạt động tốt, ổn định trong khung giờ mở cửa: 9am -> 10pm.

- Hệ thống dễ sử dụng, giao diện thân thiện với người dùng (với người dùng bình thường sẽ chỉ mất khoảng 5 phút để làm quen).
- Hệ thống có độ bảo mật thông tin người dùng cao (Ví dụ: khi user thực hiện thanh toán online, hệ thống không được phép lưu trữ thông tin thẻ tín dụng của users, mật khẩu tài khoản khách hàng được lưu sẽ được hash bằng MD5).
- Khi đầu bếp thực hiện cập nhật món ăn, hệ thống phải đồng bộ lên menu ngay lập tức tối đa khoảng 100ms.
- Hệ thống lưu trữ được nhiều dữ liệu giao dịch trong khoảng từ 6 tháng - 1 năm mà không bị quá tải hay mất dữ liệu.
- Thời gian hiện thị toàn bộ menu sau khi truy cập web hoặc thông qua mã QR không quá 1s.
- Hệ thống đều phải hỏi xác nhận (Y/N) cho các thao tác đặt hàng, hủy đơn hàng, ...
- Có thể chạy trên 3 loại trình duyệt web : chrome, edge, firefox.

Draw a use-case diagram for the whole system



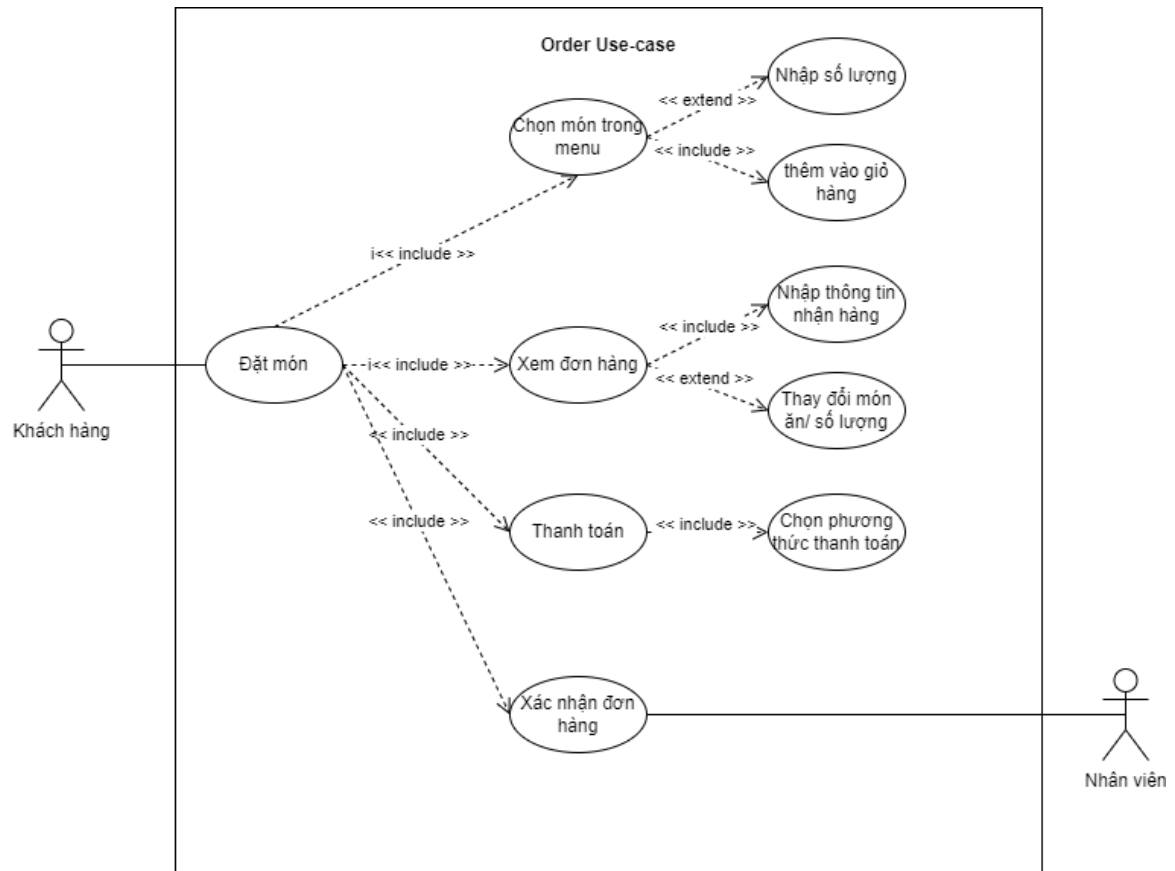
Hình 1: Use-case diagram for the whole system

1.3 Use-case diagram and description for features

Order use-case description:

Thanh toán trực tuyến	
Actors	Khách hàng, nhân viên
Descriptions	Khách hàng đặt các món trong menu qua trang web
Pre-conditions	1. Khách hàng có kết nối internet và đăng nhập vào được website của nhà hàng 2. Các món customer muốn đặt phải có trong menu 3. Cửa hàng vẫn đang tiếp nhận order (chưa đóng cửa)
Post-conditions	1. Đơn hàng được ghi nhận thành công lên hệ thống 2. Khách hàng đặt được món ăn mình muốn
Normal Flow	1. Sau khi đăng nhập vào hệ thống Customer nhấn chọn vào menu 2. Giao diện hiển thị menu các món, giá 3. Người dùng chọn các món muốn đặt 4. Người dùng thêm món vào giỏ hàng 5. Khách hàng click vào cart để xem các món đã chọn và tổng số tiền 6. Hệ thống hiển thị các món khách hàng đã chọn, số lượng, số tiền của mỗi món, tổng số tiền của Order. Khách hàng có thể chỉnh sửa giỏ hàng 7. Customer chọn thanh toán 8. Hệ thống chuyển sang phần Payment
Alternative Flow	<i>Ở bước 3:</i> Khách hàng có thể chọn filter hiển thị các món ăn theo Category, Vendor hoặc tìm kiếm món ăn theo tên <i>Ở Bước 7:</i> Nếu khách hàng đang tại cửa hàng thì có thể chọn ô Tại cửa hàng trong ô địa chỉ nhận <i>Ở Bước 7:</i> Khách hàng có thể tăng giảm số lượng của mỗi món ăn hoặc xóa món ăn đó khỏi Order <i>Ở Bước 7:</i> Khách hàng có thể chọn Back để quay trở lại Menu để trở về bước 1
Exceptions	<i>Ở bước 3:</i> Nếu cửa hàng tạm đóng sẽ hiển thị thông báo cửa hàng tạm đóng vào không tiếp nhận Order <i>Ở bước 3:</i> Khách hàng chưa chọn món nhưng click vào Cart sẽ hiển thị thông báo chưa có món được chọn và quay về Menu <i>Ở Bước 7:</i> Nếu khách hàng xóa hết món ăn thì sẽ hiện thị thông báo không có món nào được chọn và quay về Menu

Order use-case diagram:

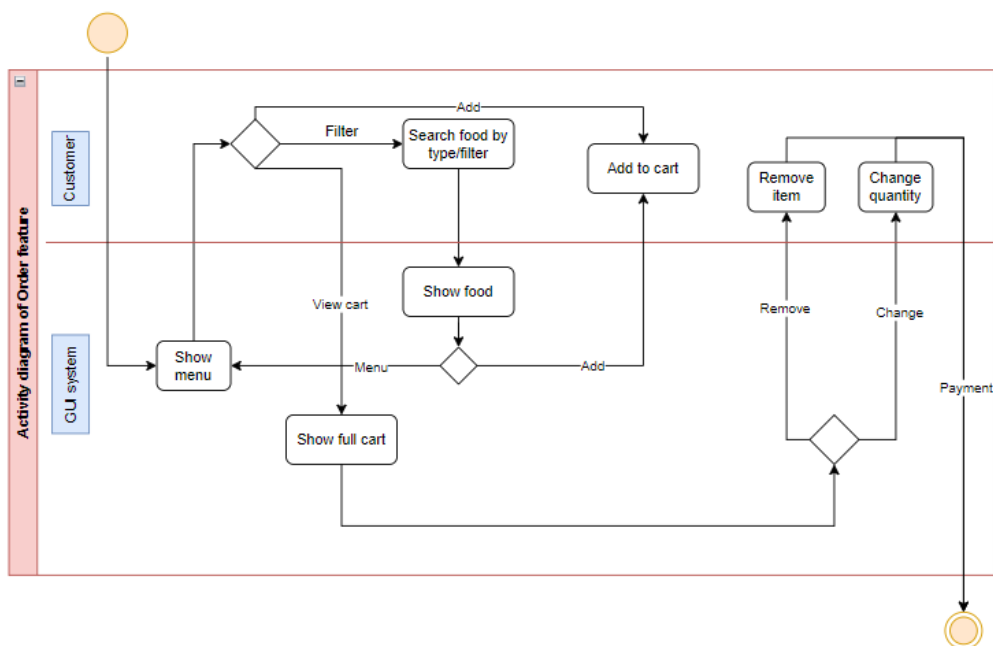


Hình 2: Order use-case diagram

2 System modeling

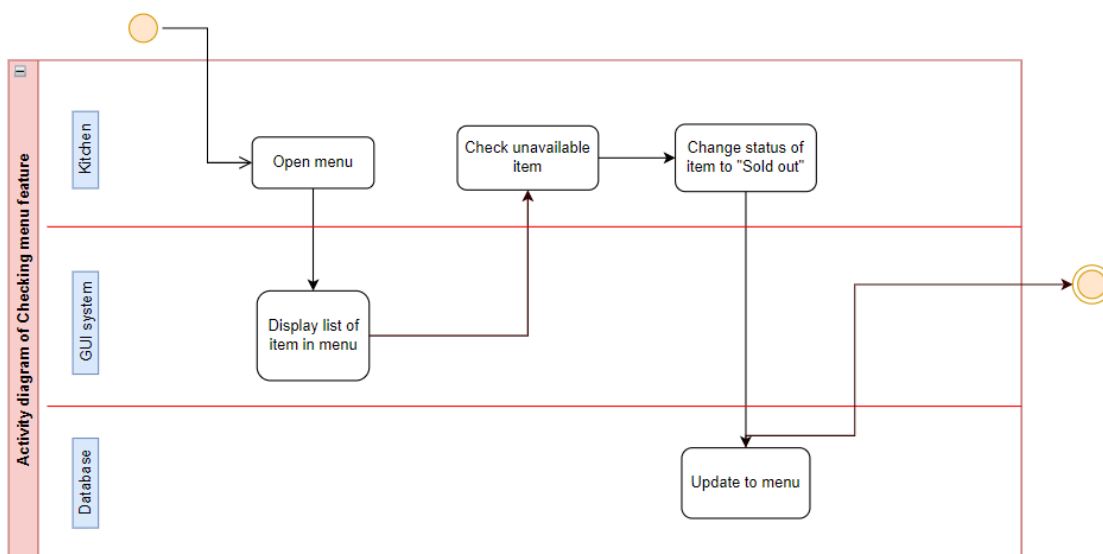
2.1 Activity diagram for functional requirements of the desired system

1. Chức năng đặt hàng



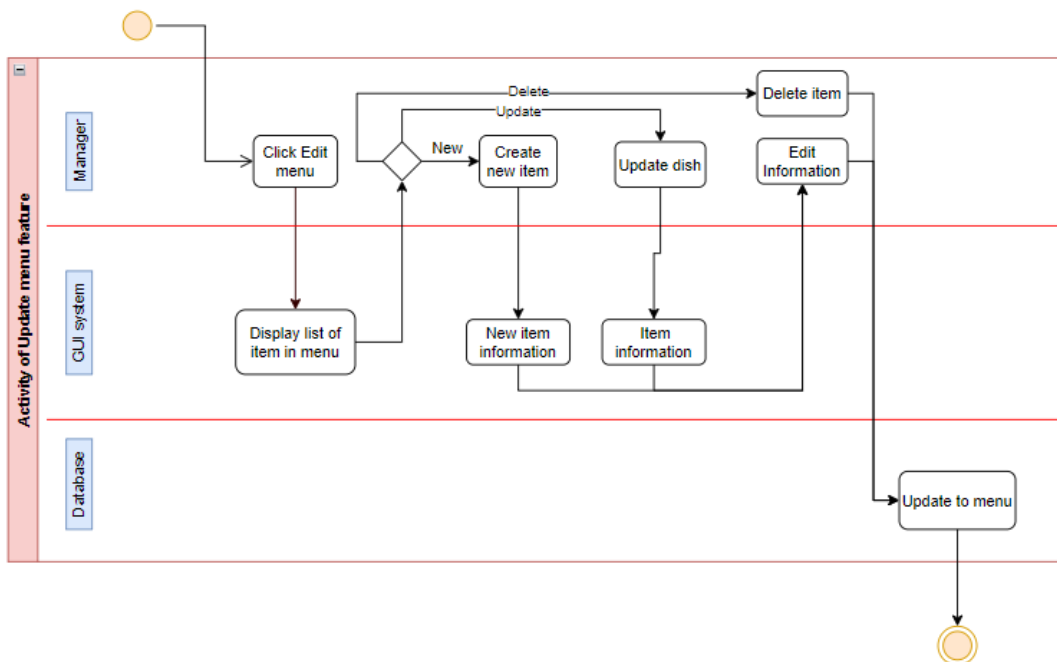
Hình 3: Activity diagram của use-case: Đặt hàng

2. Chức năng kiểm tra thực đơn



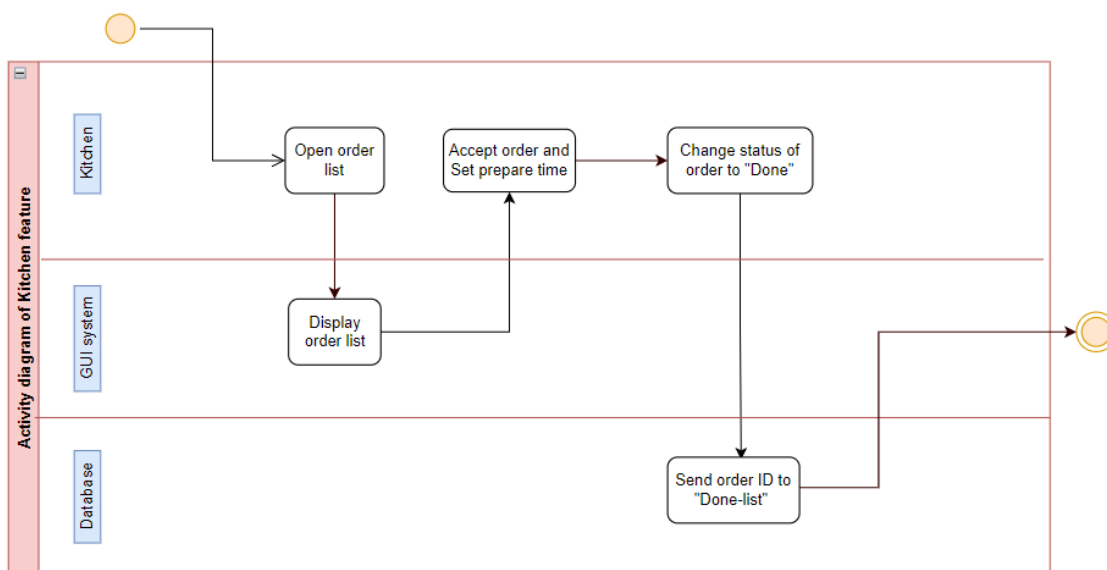
Hình 4: Activity diagram của use-case: Kiểm tra thực đơn

3. Chức năng cập nhật thực đơn



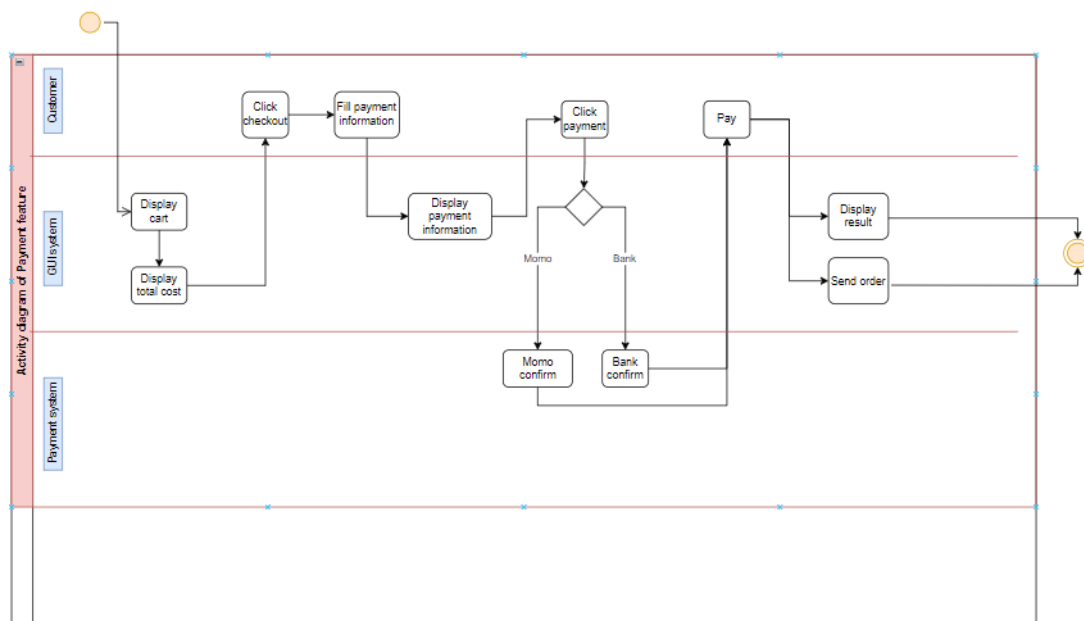
Hình 5: Activity diagram của use-case: Cập nhật thực đơn

4. Chức năng tình trạng đơn hàng



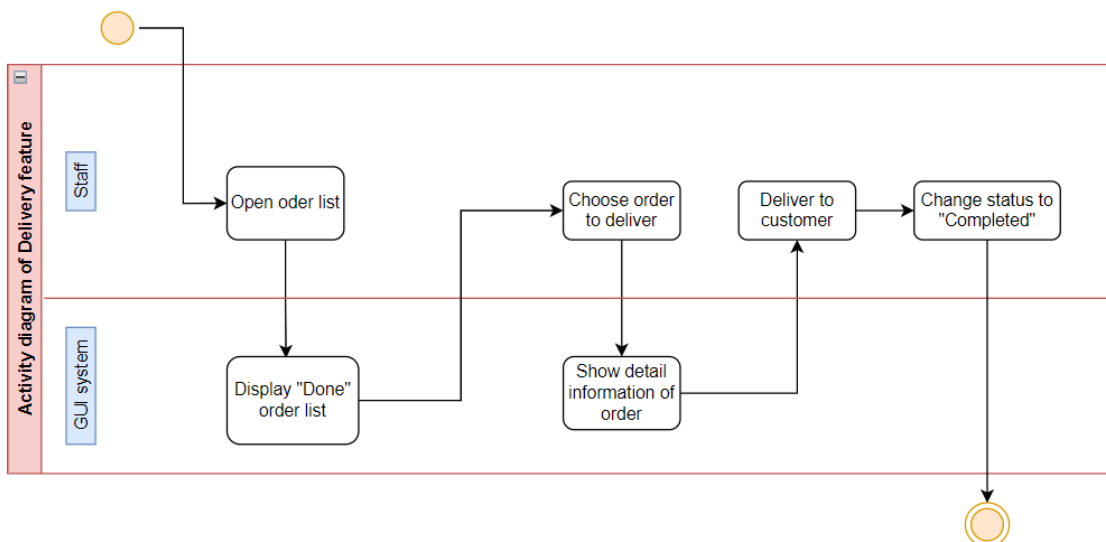
Hình 6: Activity diagram của use-case: Tình trạng đơn hàng

5. Chức năng thanh toán



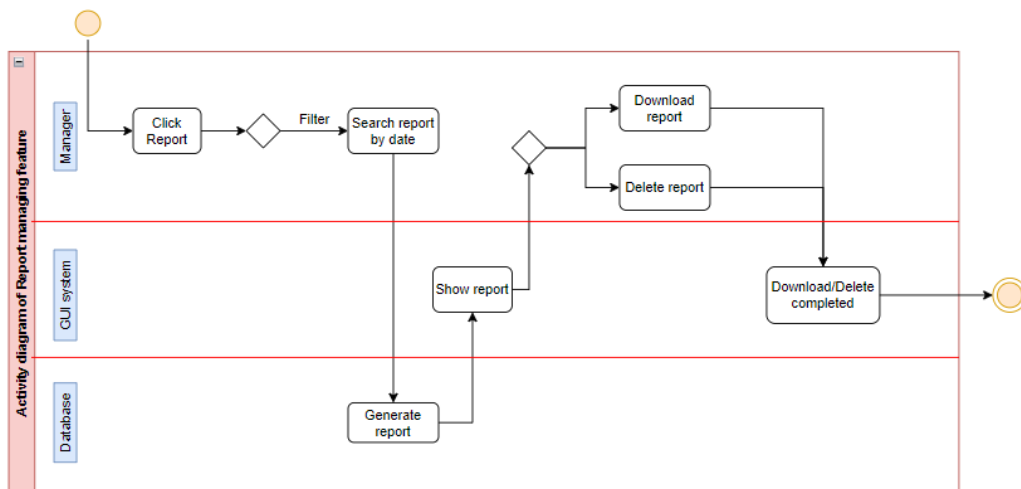
Hình 7: Activity diagram của use-case: Thanh toán

6. Chức năng giao hàng



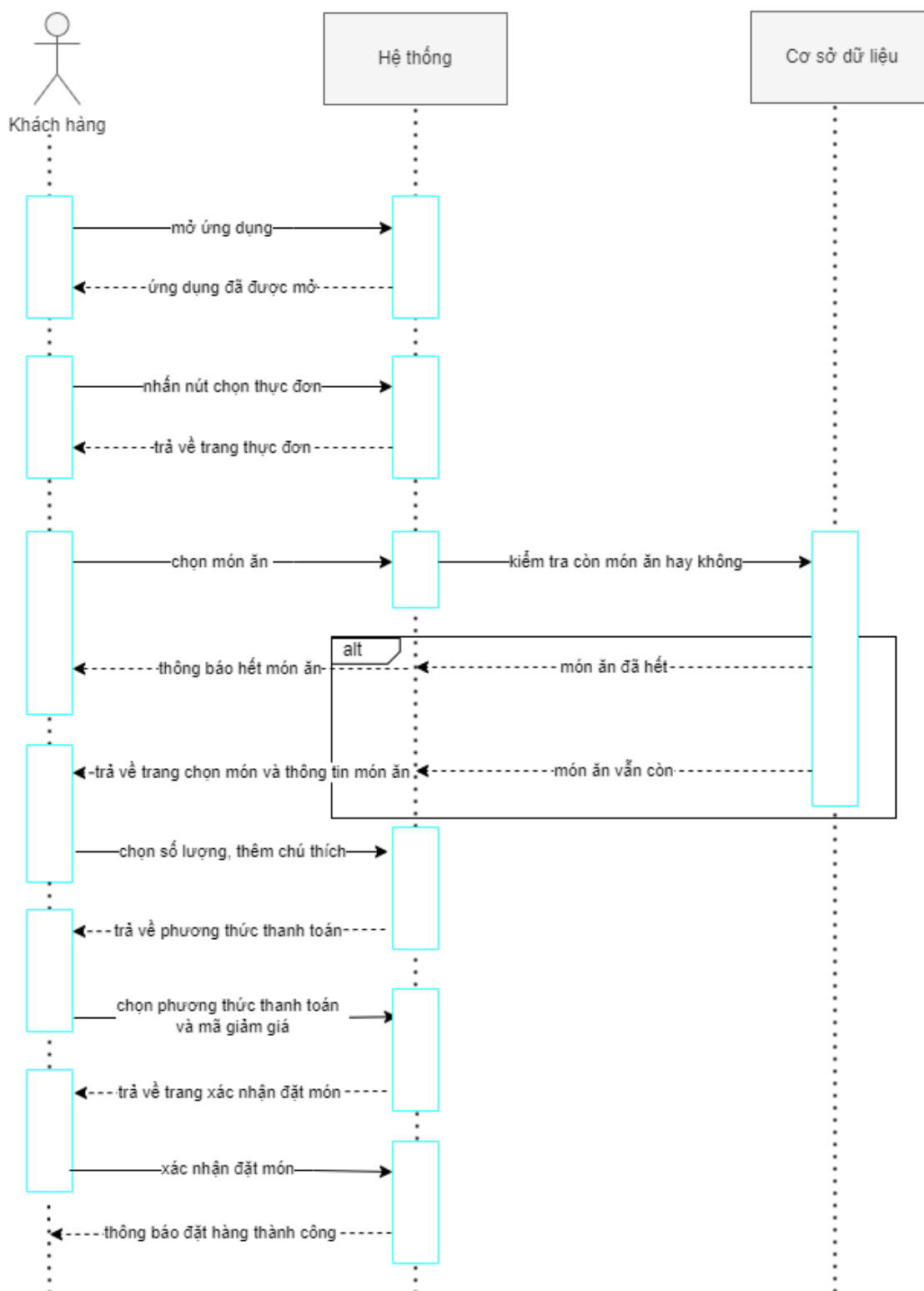
Hình 8: Activity diagram của use-case: Giao hàng

7. Chức năng báo cáo (quản lí)



Hình 9: Activity diagram của use-case: Báo cáo

2.2 Sequence diagram for use-case

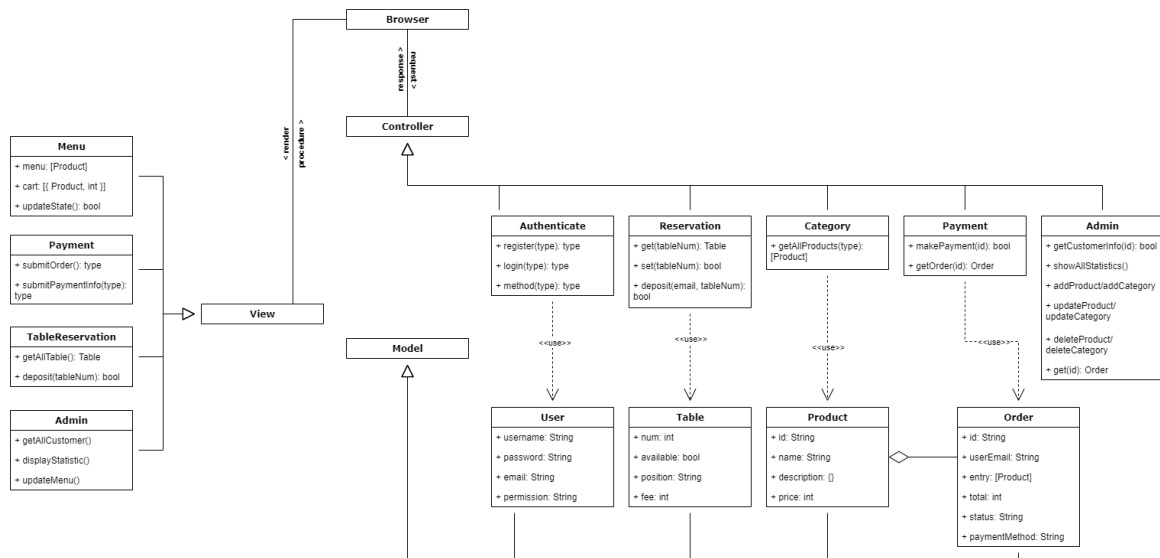


Hình 10: Sequence diagram for use-case: Đặt món

2.3 Class diagram

Draw a class diagram

Link diagram: https://app.diagrams.net/#G1y07RLNQjB3FN0rXw77h2BQkChLUk_Hk



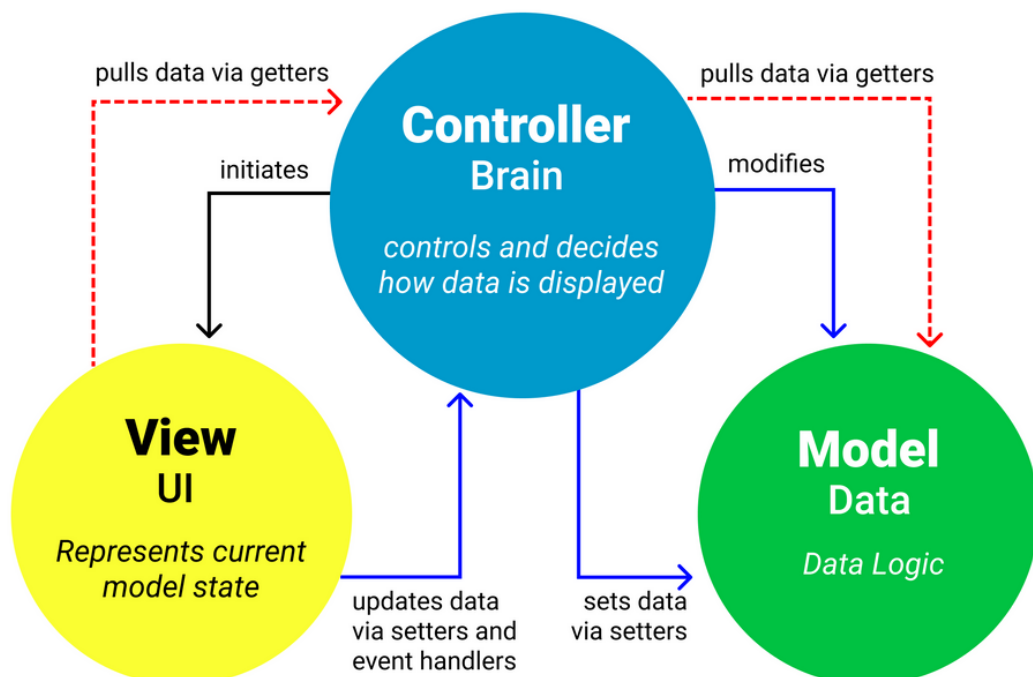
Hình 11: Class diagram của toàn bộ hệ thống

3 Architecture design

3.1 Describe an architectural approach

3.1.1 Mô hình MVC NodeJS-MySQL

MVC Architecture Pattern



Hình 12: Mô hình kiến trúc MVC

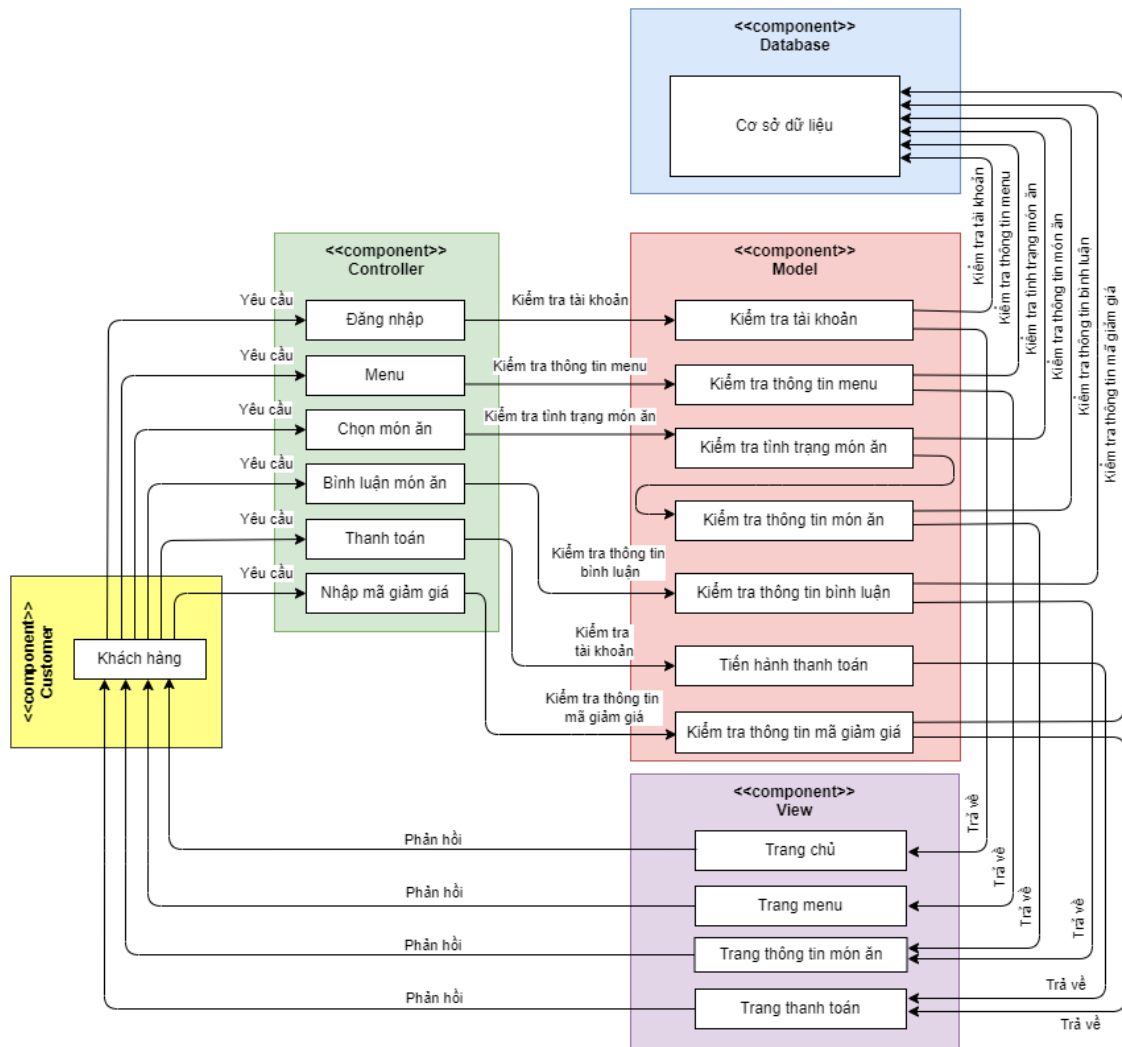
Trong đó:

- Model: giúp quản lý dữ liệu. Cho dù dữ liệu là từ cơ sở dữ liệu, API hay đối tượng JSON, thì mô

hình chịu trách nhiệm quản lý nó. Trong trường hợp này là API.

- Views: quyết định xem người dùng sẽ nhìn thấy gì trên màn hình của họ và như thế nào. Trong trường hợp này là HTML, CSS.
- Controller: lấy, sửa đổi và cung cấp dữ liệu cho người dùng. Về cơ bản, controller liên kết giữa views và model. Trong trường hợp này là NodeJS và các framework liên quan.

3.1.2 Mô hình MVC cho hệ thống



Hình 13: Mô hình MVC của hệ thống

3.1.3 Những công nghệ sử dụng

1. Front End

- Sử dụng HTML, CSS, JavaScript.
- Sử dụng framework Bootstrap, JQuery.
- Sử dụng các bộ Google Fonts, Font Awesome.

2. Back End: Sử dụng NodeJs với mô hình MVC. Lí do chọn Framework ExpressJS:

- Module express-generator giúp tạo project gần như tương tự với mô hình MVC.
- Sự phân chia các thành phần giúp dễ kiểm tra lỗi, dễ chia việc.

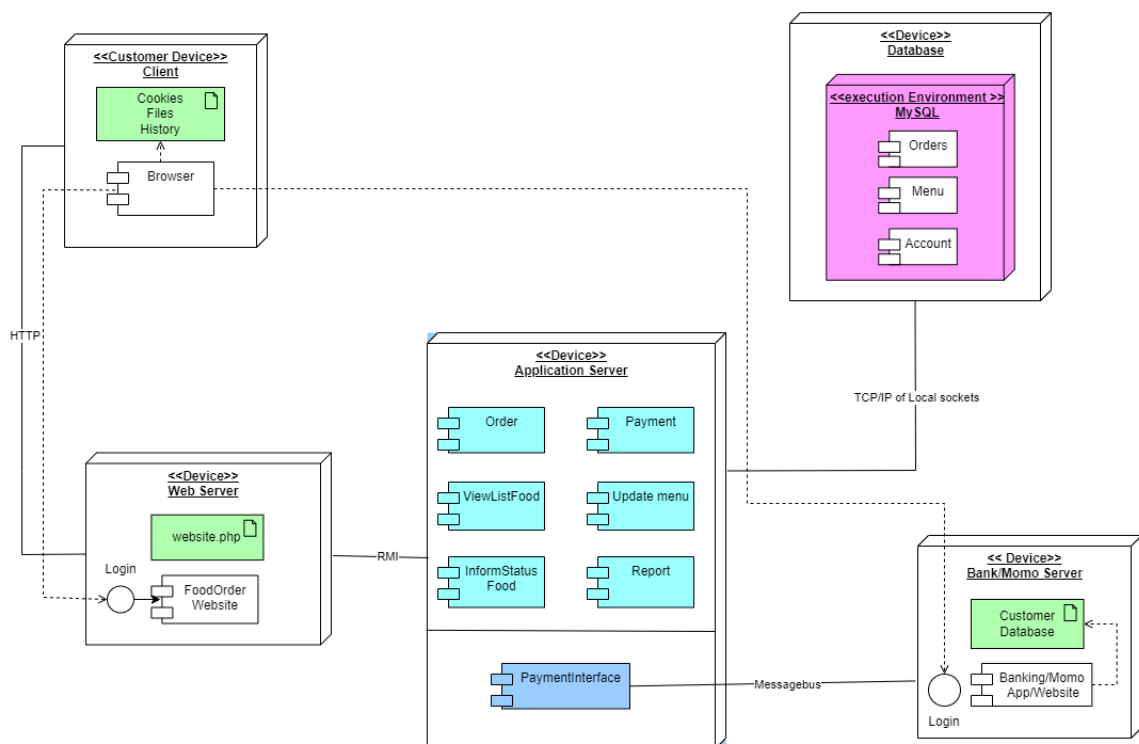
- Hỗ trợ REST API.

3. **Data Management:** Sử dụng MySQL Server. Lí do chọn MySQL để quản lý cơ sở dữ liệu:

- Dễ sử dụng, do MySQL là hệ thống quản trị cơ sở dữ liệu quan hệ nên dễ quản lý, dễ sử dụng.
- Phù hợp với các thành viên đều sử dụng hệ điều hành Windows.
- Có thể tích hợp với NodeJs, Framework được sử dụng làm Back End.
- MySQL thích hợp cho các dự án vừa và nhỏ.

3.2 Implementation diagram for functional requirements

HỆ THỐNG POS 2.0



Hình 14: Implementation diagram cho toàn bộ hệ thống

4 Task 4: Implementation - Sprint 1

4.1 Task 4.1

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

Link Github nhóm làm việc: <https://github.com/HuuHanh31/POS-2.0-Restaurant.git>

4.2 Task 4.2

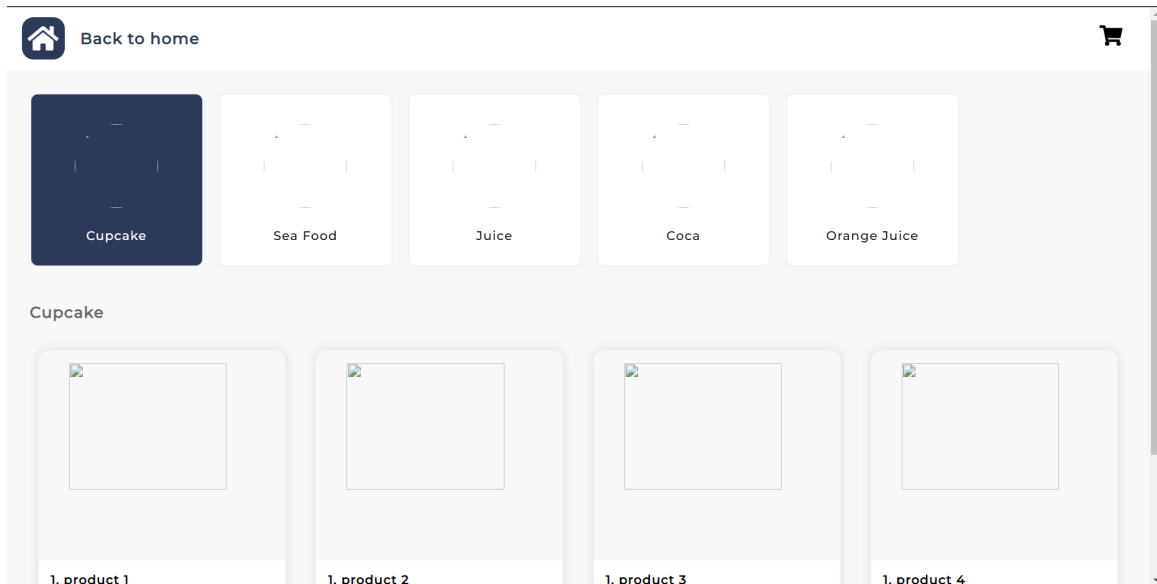
Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files/ folders

File database và file mô tả yêu cầu nằm trong link github: <https://github.com/HuuHanh31/POS-2.0-Restaurant.git>

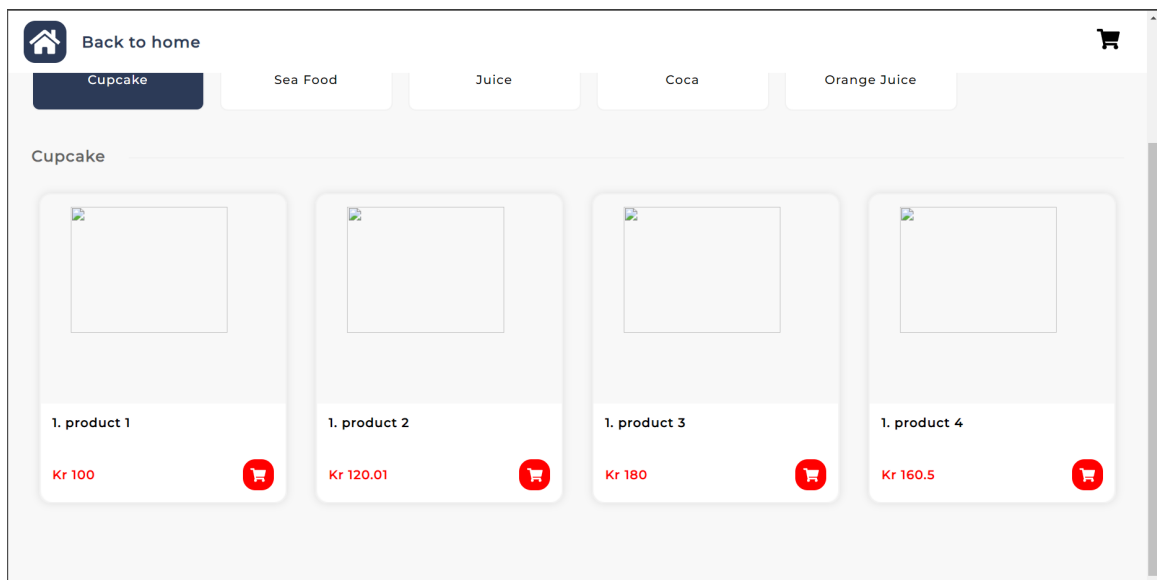
4.3 Task 4.3

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. MVP cho food-ordering:

- Danh sách các món ăn:

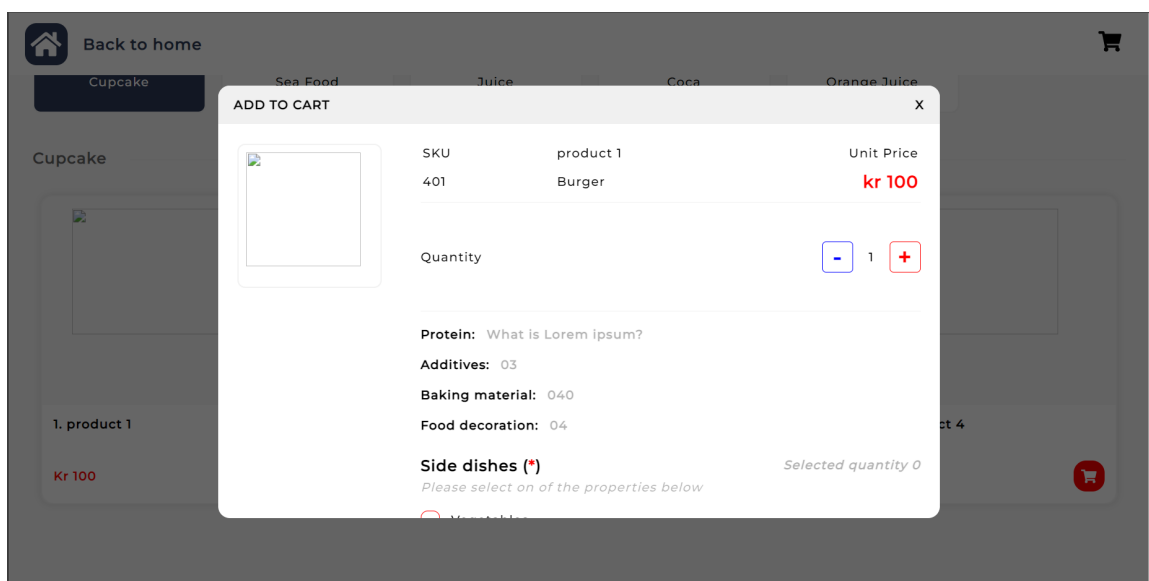


Hình 15: Danh sách món ăn 1



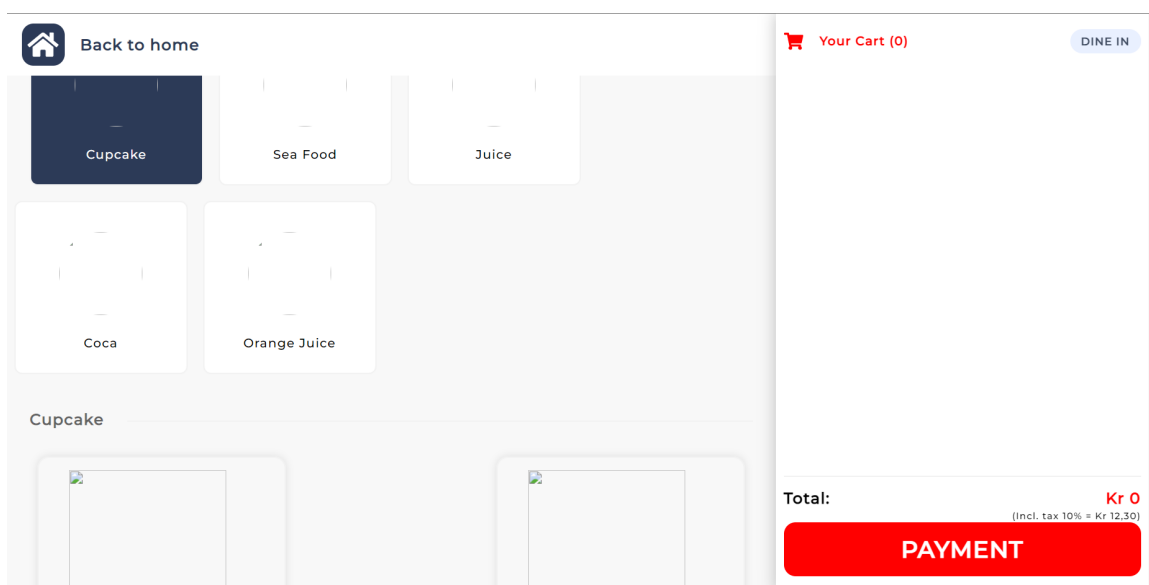
Hình 16: Danh sách món ăn

- Click vào món ăn và xem chi tiết món ăn:



Hình 17: Click vào món ăn và xem chi tiết

- Click vào giỏ hàng:



Hình 18: Giỏ hàng

5 Task 5: Implementation - Sprint 2

Tài liệu