



## Software Engineering

---

### Assignment

# RESTAURANT POS 2.0

---

Teacher: Lê Đình Thuận  
Member List: Nguyễn Văn Nhật Tiến - 1915486  
Võ Đình Thanh - 1912041  
Đặng Quốc Thắng - 1912084  
Nguyễn Hữu Ngọc Vinh - 1915939  
Mai Đức Long - 1913996  
Lê Bảo Khánh - 1911363

TP Hồ Chí Minh, 10/2021

# Contents

|  |           |
|--|-----------|
| <b>Introduction to project</b>                       | <b>2</b>  |
| <b>1 Requirement Elicitation</b>                     | <b>3</b>  |
| 1.1 Context of this project . . . . .                | 4         |
| 1.1.1 Identify the context of this project . . . . . | 4         |
| 1.1.2 Who are relevant stakeholders? . . . . .       | 4         |
| 1.1.3 What are expected to be done? . . . . .        | 4         |
| 1.1.4 What are the scope of the project? . . . . .   | 4         |
| 1.2 Gerenal features of project . . . . .            | 5         |
| 1.2.1 Functional requirements . . . . .              | 5         |
| 1.2.2 Non-functional requirements . . . . .          | 5         |
| 1.2.3 Use-case diagram . . . . .                     | 6         |
| 1.3 Detailed features . . . . .                      | 7         |
| 1.3.1 Work assignment . . . . .                      | 7         |
| 1.3.2 Detailed features . . . . .                    | 7         |
| <b>2 System Modeling</b>                             | <b>21</b> |
| 2.1 Activity Diagram . . . . .                       | 22        |
| 2.2 Sequence Diagram . . . . .                       | 28        |
| 2.3 Class Diagram . . . . .                          | 34        |
| <b>3 Architecture Design</b>                         | <b>35</b> |
| 3.1 An architectural approach - MVC . . . . .        | 36        |
| 3.2 Package Diagram . . . . .                        | 37        |

# Introduction to project

- Point of sale (POS) or point of purchase is the time and place where a retail transaction is completed. At the point of sale, the merchant calculates the amount owed by the customer, indicates that amount, may prepare an invoice for the customer, and indicates the options for the customer to make payment. In restaurant business, POS systems often include table reservation, ordering food, alerts, billing, credit card processing and customer management. Even before the COVID-19 crisis, POS systems had gained traction across the industry.
- During the coronavirus pandemic, restaurants face greater peril than ever. Such systems are expected to increase business intelligence, reduce wasted effort and opportunity to scale to a large business. Moreover, the systems should support take-away options. Our customers have multiple restaurants and have a need to develop a responsive web-based POS system that implement the current business flow as described in below figure

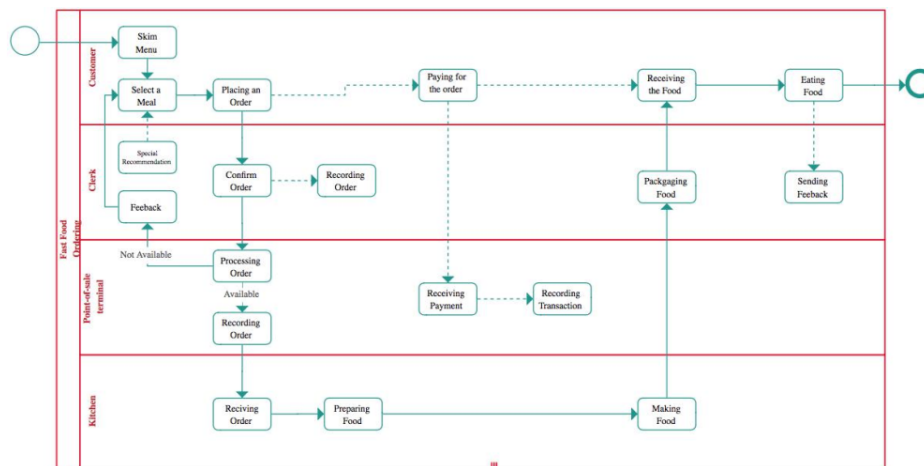


Figure 1: Customer-drawing workflow

- The restaurant owners express several special demands for the new system:
  - The system should allow non-direct contact between Clerks and Customers
  - The system should be implemented using Web technology and QR code, so customers will not have to install apps
  - The system should be usable from a mobile device, a tablet device or a normal computer/ laptop
  - The system should be extendable to use in multiple restaurants in the future
  - The current transactions is about 300 orders per day.

## Chapter 1

# Requirement Elicitation

## 1.1 Context of this project

### 1.1.1 Identify the context of this project

- Trong thời kỳ đại dịch, cần một ứng dụng web để mua hàng với mà không cần tiếp xúc trực tiếp giữa Nhân viên và Khách hàng. Trong dự án này, chúng ta tạo một ứng dụng web đơn giản có thể sử dụng được từ thiết bị di động, máy tính bảng hoặc máy tính / máy tính xách tay thông thường. Vì vậy, Khách hàng không phải mất thời gian tải xuống bất cứ thứ gì mà vẫn có thể đặt món.
- Để phục vụ mục đích không tương tác trực tiếp giữa nhân viên với khách hàng, tại bàn ăn có mã QR để Khách hàng quét và sau đó chuyển trực tiếp họ đến trang Web. Họ có thể chọn đồ ăn, nước uống trong thực đơn sau đó đặt món.
- Khách hàng có thể thanh toán bằng Ví điện tử hoặc Thẻ tín dụng, có thể đăng ký làm Thành viên và được giảm giá sau một số điều kiện nhất định.
- Bên cạnh đó, Khách hàng có thể đánh giá, xem profile cá nhân, thông tin nhà hàng, đặt bàn...

### 1.1.2 Who are relevant stakeholders?

1. Customer(Khách hàng) có thể đăng kí để trở thành Member(Thành viên) - Sử dụng dịch vụ của nhà hàng
2. Manager(Quản lý) - Có trách nhiệm quản lý nhà hàng
3. Chef(Đầu bếp) - Phục vụ món ăn, nhận đánh giá từ customer
4. Staff(Nhân viên) - Tham gia vào các hoạt động của nhà hàng, phục vụ customer

### 1.1.3 What are expected to be done?

1. Chức năng Register
2. Chức năng Login
3. Chức năng View Menu
4. Chức năng Food Ordering
5. Chức năng Make payment
6. Chức năng Collecting and Use Discount points
7. Chức năng Feedback

### 1.1.4 What are the scope of the project?

1. Hệ thống có thể sử dụng trên điện thoại di động, máy tính bảng, hoặc laptop.
2. Khi đăng ký người dùng cung cấp số điện thoại, email, họ tên.
3. Member đăng nhập vào hệ thống bằng số điện thoại.
4. Người dùng có thể xem những món có thể đặt được tại thời điểm đặt món.
5. Người dùng có thể đặt được những món xuất hiện trong Menu.
6. Payment method: Chấp nhận tiền mặt, ví điện tử (Momo, ZaloPay ...), thẻ thanh toán nội địa (OCB, Agribank ...), thẻ thanh toán quốc tế VISA, Master Card.
7. Discount points(Điểm khuyến mại) chỉ dùng để đổi khuyến mại, giảm giá khi Đặt món chứ không có chức năng khác.
8. Khi đánh giá người dùng có thể rating từ 1 đến 5 sao và ghi đánh giá dưới 1000 từ.
9. Chỉ những khách hàng đã đăng kí thành viên thì mới có thể thực hiện đặt bàn.

## 1.2 Gerenal features of project

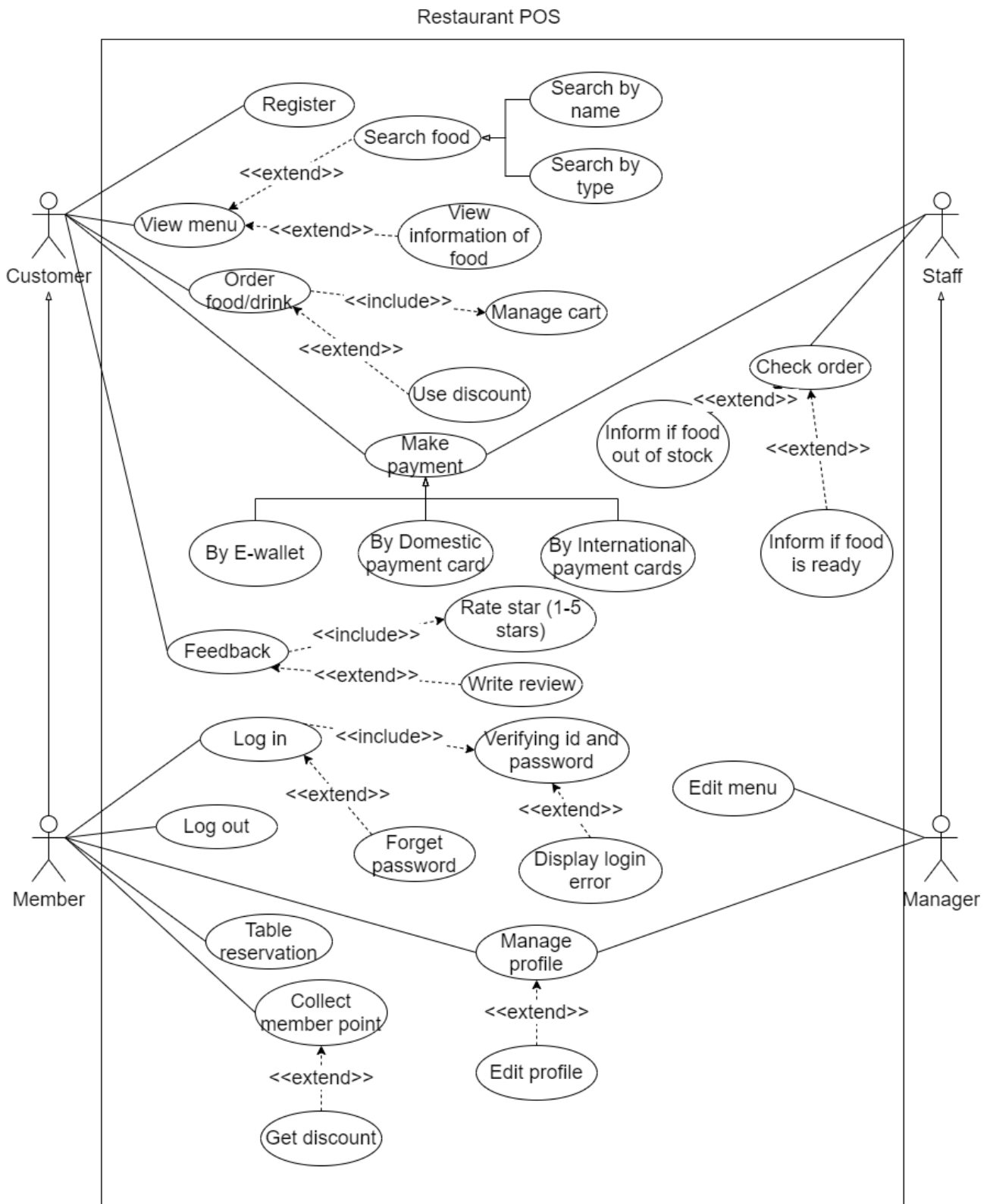
### 1.2.1 Functional requirements

1. Register: Customer chưa có tài khoản của nhà hàng có thể đăng kí để trở thành Member của nhà hàng, từ đó nhận được nhiều lợi ích hơn
2. Sign in: Members (Customer đã có tài khoản) có thể đăng nhập vào hệ thống.
3. View menu: người dùng có thể xem menu, xem thông tin món ăn, tìm món ăn bằng tên, loại món ăn,...
4. Order food: khi đặt món ăn, người dùng phải thêm món ăn vào giỏ hàng, sử dụng discount (nếu có/nếu muốn) và thanh toán.
5. Make payment:
  - Customer và Member có thể thanh toán bằng phương thức tiền mặt, thẻ tín dụng hoặc sử dụng ví điện tử.
  - Staff hoặc Manager của nhà hàng cùng tham gia vào chức năng này. Họ nhận tiền và xác minh việc thanh toán của khách hàng.
6. Table reservation: Member có thể đặt bàn trước khi họ đến nhà hàng.
7. Feedback: Customer và Member có thể đánh giá cho nhà hàng bằng việc rate sao và viết review về trải nghiệm của họ tại nhà hàng
8. Collect point: Members thu thập điểm thành viên sau mỗi đơn hàng. Nếu có đủ điểm, họ có thể đổi điểm để lấy discount (có thể sử dụng cho những đơn hàng tiếp theo để giảm giá)
9. Manage profile: Khách hàng có thể quản lý hồ sơ tài khoản của mình (xem thông tin, đổi tên, đổi email,...)
10. Manage menu: Manager có thể quản lý, sửa đổi menu của nhà hàng
11. Manage customer: Staff và Manager có thể quản lý các thông tin của khách hàng như hồ sơ, đơn đặt hàng, đặt bàn,...

### 1.2.2 Non-functional requirements

1. Hệ thống cho phép giao dịch trực tiếp hoặc không tương tác trực tiếp.
2. Hệ thống có thể sử dụng trên điện thoại di động, máy tính bảng, hoặc laptop.
3. Hệ thống có thể mở rộng để sử dụng cho nhiều nhà hàng trong tương lai..
4. Có thể thực hiện ít nhất 300 đơn một ngày.
5. Users có thể dễ dàng sử dụng sau nhiều nhất là 1 tiếng training.
6. Web-based.
7. Mở cửa từ 7h sáng đến 9h tối
8. Giao diện thân thiện với người dùng.
9. Thời gian phản hồi của các chức năng không quá 3 giây (trừ những chức năng cần đợi xác minh của Staff như thanh toán, đặt bàn,...)

### 1.2.3 Use-case diagram



## 1.3 Detailed features

### 1.3.1 Work assignment

| Member               | Feature           |
|----------------------|-------------------|
| Mai Đức Long         | Login             |
| Nguyễn Văn Nhật Tiến | Table Reservation |
| Nguyễn Hữu Ngọc Vinh | Register          |
| Võ Đình Thanh        | Collect Point     |
| Đặng Quốc Thắng      | Food ordering     |
| Lê Bảo Khánh         | Make payment      |

### 1.3.2 Detailed features

#### a. Register

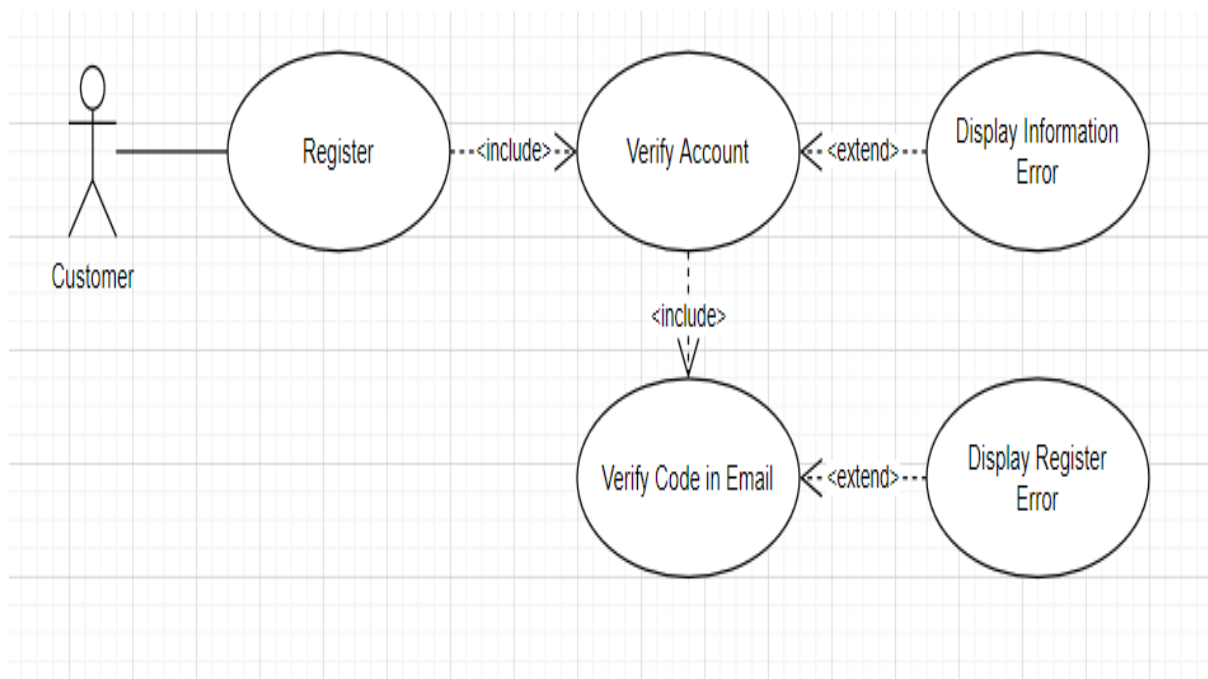


Figure 1.1: Register use-case diagram

| Actor                    | System user                  |
|--------------------------|------------------------------|
| <i>Use case name</i>     | Register                     |
| <i>Use case overview</i> | Người dùng đăng ký tài khoản |
| <i>Actor(s)</i>          | Customer                     |
| <i>Trigger</i>           | Người dùng đăng ký tài khoản |
| <i>Preconditions</i>     | Người dùng đăng ký tài khoản |
| <i>Postconditions</i>    | Bạn đã đăng ký thành công    |



*Normal Flow*

1. Hệ thống hiển thị màn hình để người dùng đăng ký.
2. Người dùng nhập thông tin cơ bản.
3. Hệ thống kiểm tra thông tin.
4. Hệ thống gửi mã xác nhận đến người dùng qua email để xác thực tài khoản.
5. Người dùng nhập mã xác nhận và hệ thống xác nhận.
6. Nếu đăng ký thành công, hệ thống xuất ra màn hình đăng nhập cho người dùng.
7. Kết thúc use-case.

---

*Exception Flow*

E1: Nếu thiếu thông tin

- Thông báo "Thông tin không đủ" sẽ hiển thị.
- Quay lại bước 2 trong Normal Flow.

E2: Nếu tên đăng nhập đã tồn tại

- Thông báo "Tài khoản đã tồn tại" sẽ hiển thị.
- Quay lại bước 2 trong Normal Flow.

E3: Nếu người dùng nhập sai mã xác thực

- Thông báo "Mã không hợp lệ. Vui lòng thử lại." sẽ hiển thị.
  - Quay lại bước 5 trong Normal Flow.
-

b. Login

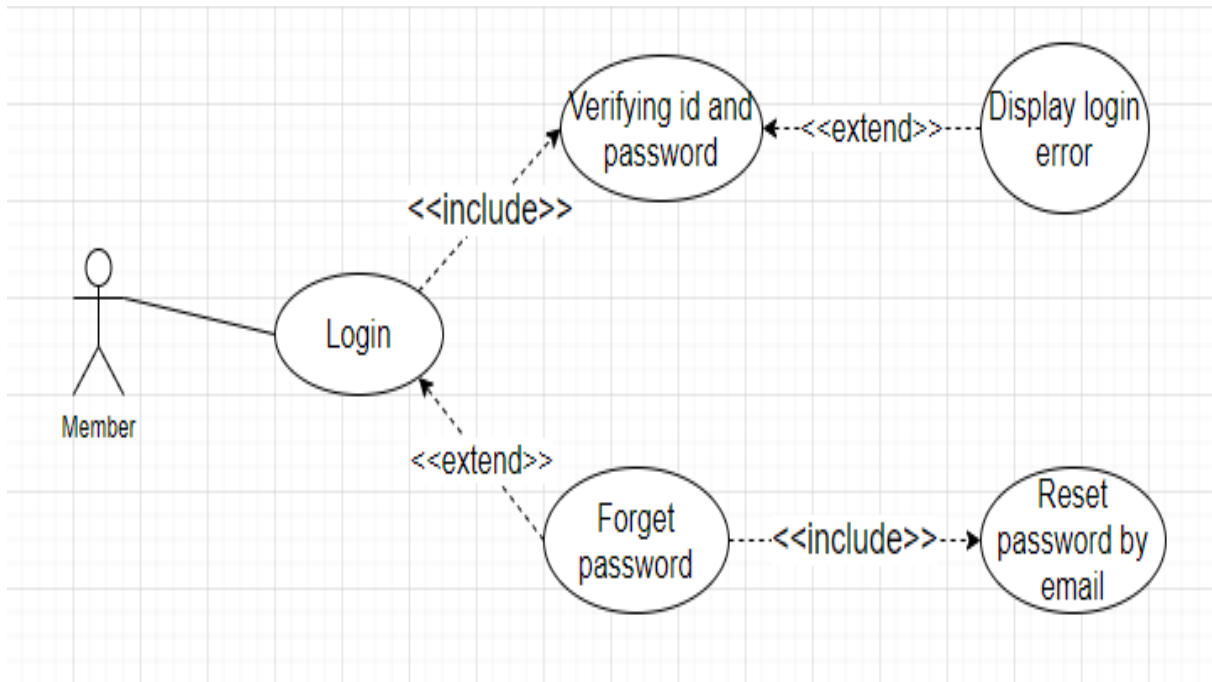


Figure 1.2: Login use-case diagram

| Actor             | System user  |
|-------------------|--|
| Use case name     | Log in   |
| Use case overview | Người dùng đã có tài khoản có thể đăng nhập vào hệ thống để thực hiện các chức năng của họ.  |
| Actor(s)          | Member   |
| Trigger           | <ul style="list-style-type: none"> <li>Thiết bị của người dùng có kết nối mạng</li> <li>Người dùng chưa đăng nhập</li> <li>Người dùng là Member (đã có tài khoản)</li> </ul>   |
| Preconditions     | Người dùng đăng ký tài khoản   |
| Postconditions    | Người dùng đăng nhập thành công vào hệ thống   |
| Normal Flow       | <ol style="list-style-type: none"> <li>Hệ thống hiển thị màn hình đăng nhập.</li> <li>Người dùng nhập tên đăng nhập và mật khẩu và nhấn nút "Login".</li> <li>Hệ thống kiểm tra thông tin đăng nhập.</li> <li>Nếu thành công, hệ thống cập nhật lại giao diện theo thông tin tài khoản của người dùng</li> <li>Kết thúc use-case.</li> </ol> |

*Exception Flow*

E1: Người dùng nhập sai tài khoản hoặc mật khẩu:

- Hệ thống hiển thị lại màn hình đăng nhập kèm theo thông báo “Tên đăng nhập hoặc mật khẩu bị sai”.
- Quay lại bước 2 trong Normal Flow

E2: Quên mật khẩu: Người dùng chọn nút “Forget password” khi quên mật khẩu

- Người dùng nhập tên đăng nhập và email xác thực, sau đó chọn nút “Reset password”
  - Hệ thống kiểm tra tên đăng nhập và email xác thực.
  - Nếu đúng, hệ thống gửi link xác thực tới email của người dùng.
  - Người dùng nhấn vào link, nhập mật khẩu mới và nhấn “Reset password”
  - Hệ thống hiện màn hình thông báo “Reset password success”
  - Use-case kết thúc.
  - Người dùng nhập tên đăng nhập và email xác thực, sau đó chọn nút “Reset password”
-

c. Collect Point

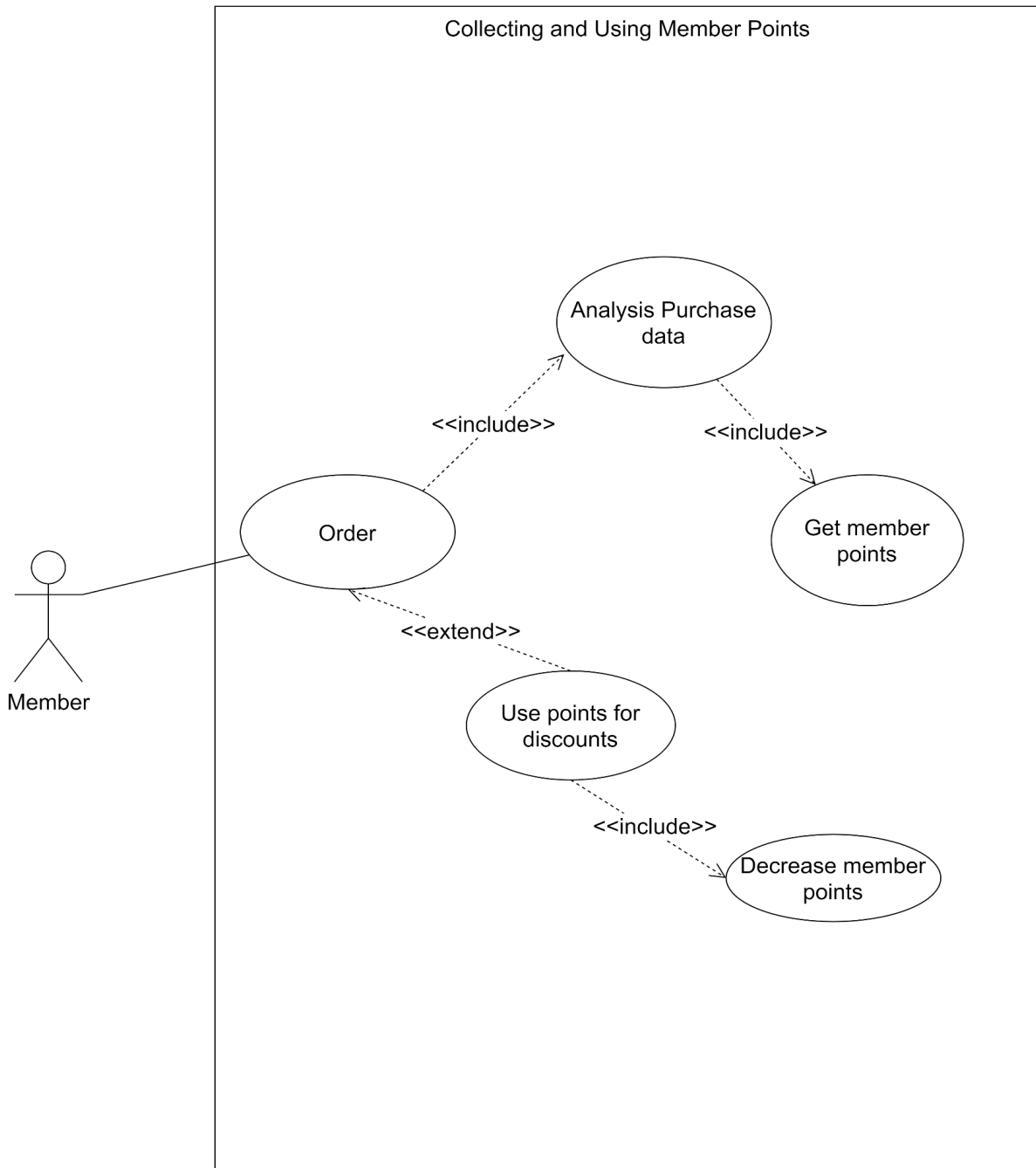


Figure 1.3: Collect Point use-case diagram

| Actor                    | System user  |
|--------------------------|--|
| <i>Use case name</i>     | Collect and Use Point  |
| <i>Use case overview</i> | Members thu thập Điểm giảm giá cho mỗi lần đặt món. Nếu Members có đủ điểm có thể dùng Điểm để đổi Giảm giá. |
| <i>Actor(s)</i>          | Member   |

|                       |   |
|-----------------------|---|
| <i>Trigger</i>        | <ul style="list-style-type: none"><li>– Đặt món</li><li>– Members nhấn vào nút “Exchange Discount” (optional)</li></ul>   |
| <i>Preconditions</i>  | <ul style="list-style-type: none"><li>– Website, Cơ sở dữ liệu và Internet có sẵn.</li><li>– Người dùng là Member.</li></ul>  |
| <i>Postconditions</i> | <ul style="list-style-type: none"><li>– Điểm giảm giá được lưu vào tài khoản của Members.</li><li>– Nếu Members sử dụng Điểm thì được giảm giá và mất điểm.</li></ul>   |
| <i>Normal Flow</i>    | <ol style="list-style-type: none"><li>1. Members đăng nhập.</li><li>2. Members đặt món.</li><li>3. Điểm của Members được cập nhật tự động.</li><li>4. Thông báo về Điểm hiện có.</li><li>5. Use-case kết thúc.</li><li>6. Optional: Members sử dụng Điểm thì được giảm giá và mất điểm.</li><li>7. Members nhấn vào nút “Exchange Discount”</li><li>8. Người dùng chọn lựa chọn Giảm giá mà mình muốn.</li><li>9. Nhấn vào nút “OK”. Nếu người dùng đủ điểm họ sẽ được giảm giá ở bước 2.</li><li>10. Điểm giảm giá sẽ bị giảm.</li><li>11. Thông báo về Điểm hiện có.</li><li>12. Use-case kết thúc.</li></ol> |
| <i>Exception Flow</i> | <p>E1: Ở bước 7, Members không có đủ điểm nhưng vẫn sử dụng chức năng giảm giá</p> <ul style="list-style-type: none"><li>– Thông báo rằng họ không có đủ điểm.</li><li>– Kết thúc Use-case</li></ul>  |

d. Food Ordering

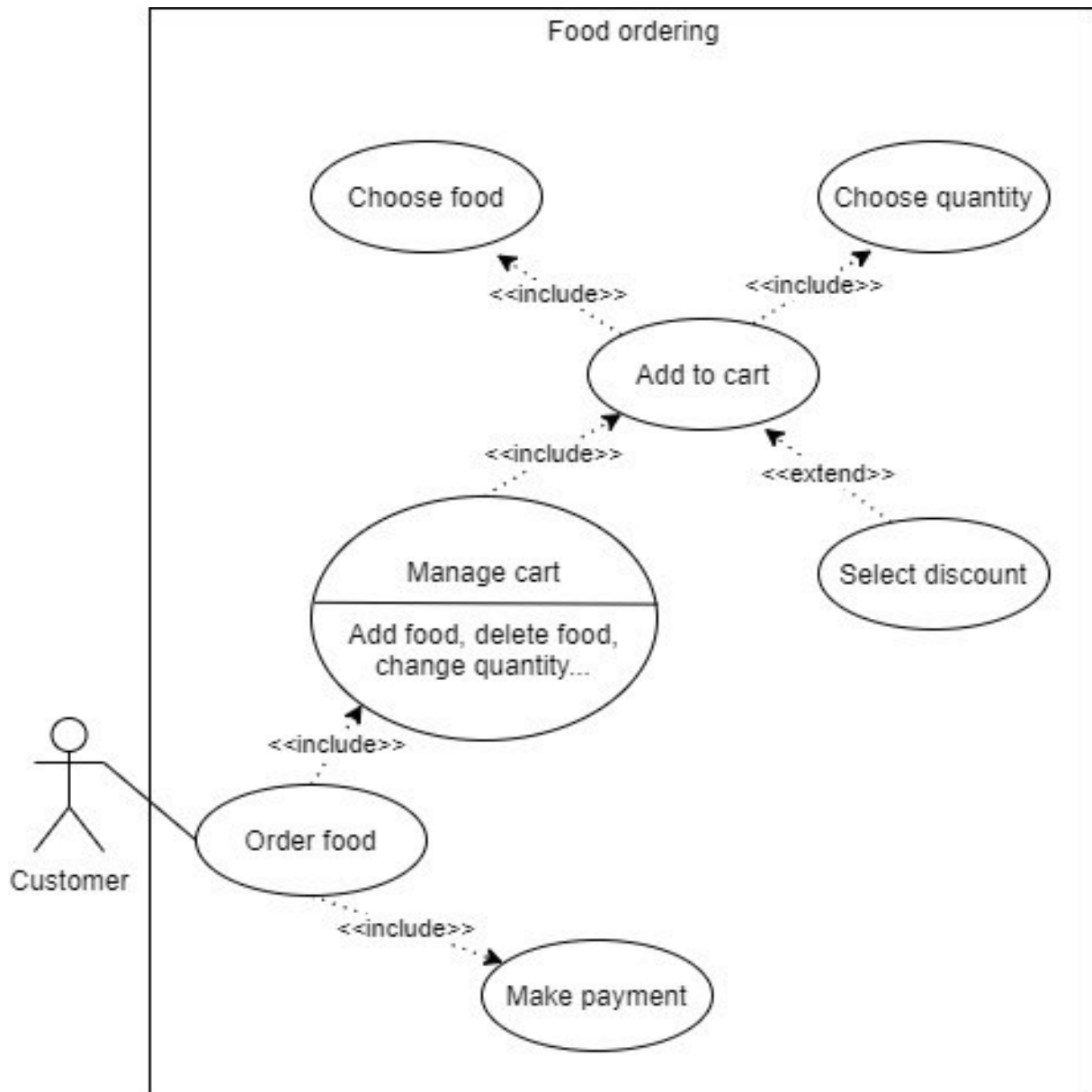


Figure 1.4: Food ordering use-case diagram

| Actor             | System user   |
|-------------------|---|
| Use case name     | Food Ordering   |
| Use case overview | Customer đặt đồ ăn từ menu  |
| Actor(s)          | Customer  |
| Trigger           | Customer nhấn vào nút “Order food”  |
| Preconditions     | <ul style="list-style-type: none"> <li>– Website, Database có sẵn</li> <li>– Thiết bị của customer có Internet</li> </ul> |
| Postconditions    | Customer đặt đồ ăn thành công   |

---

*Normal Flow*

1. Hệ thống hiển thị thông tin món ăn trong menu
2. Customer chọn món ăn muốn đặt
3. Customer chọn số lượng, discount(nếu có) và nhấn nút “Add to cart”
4. Customer nhấn vào nút “Your cart”
5. Customer quản lý và điều chỉnh giỏ hàng
6. Customer nhấn vào nút “Pay now” để thanh toán
7. Use-case kết thúc

---

*Exception Flow*

- E1: Customer nhấn vào nút “Back to home”:
- Quay trở về trang chủ
  - Use-case kết thúc
-

e. Make Payment

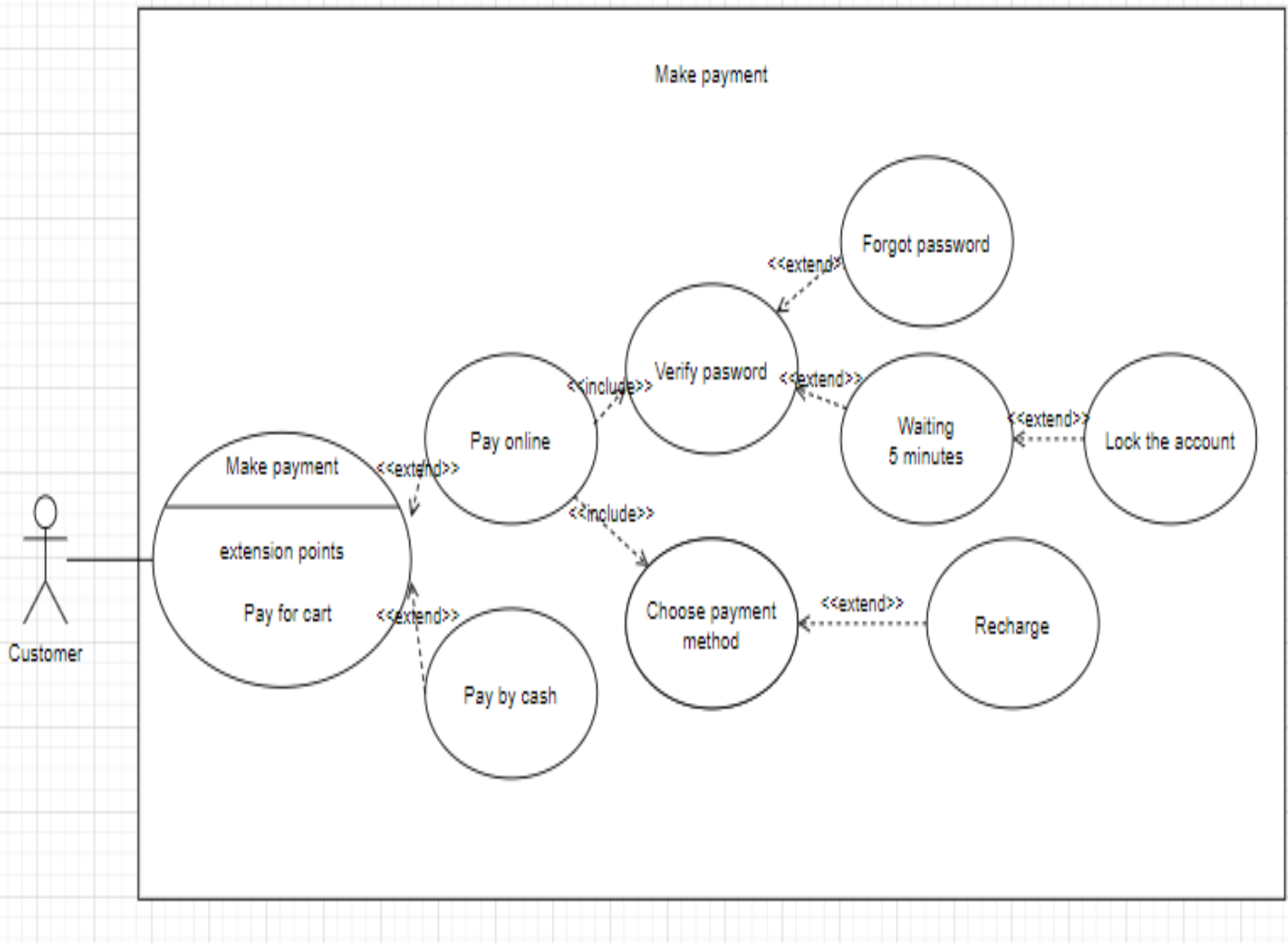


Figure 1.5: Make payment use-case diagram

| Actor             | System user   |
|-------------------|---|
| Use case name     | Make payment  |
| Use case overview | Cho phép customer thanh toán online cho đơn hàng  |
| Actor(s)          | Customer  |
| Trigger           | <ul style="list-style-type: none"> <li>– Sau khi đặt hàng</li> <li>– Chọn “Pay now” trên app</li> </ul>   |
| Preconditions     | <ul style="list-style-type: none"> <li>– Customer phải có account</li> <li>– Customer phải login trên app</li> <li>– Thiết bị của customer phải có kết nối Internet</li> <li>– Customer đã hoàn tất đặt hàng</li> </ul> |



*Postconditions*

- Tài khoản của customer bị trừ tiền tương ứng với đơn hàng
- Hệ thống thông báo “Payment successful”
- Bộ phận cashier nhận được tiền

---

*Normal Flow*

1. Customer chọn “Pay now” trên app
2. Hiện thị thông tin đơn hàng trên màn hình thiết bị
3. Customer chọn “Confirm” (xác nhận đơn hàng)
4. Customer chọn hình thức thanh toán (bằng tiền mặt, credit card / e-wallet ...)
5. Yêu cầu customer nhập lại mật khẩu để xác thực thanh toán
6. Hiện thị: “Payment successful” trên màn hình thiết bị

---

*Alternative Flow*

- A1: Ở bước 3, Customer muốn edit đơn hàng
- Customer chọn “Edit” đơn hàng
  - Customer chỉnh sửa lại đơn hàng đã đặt
  - Customer xác nhận đặt hàng
  - Tiếp tục bước 4 của Normal Flow
- A2: Ở bước 4, Hình thức thanh toán không đủ tiền
- Hình thức thanh toán customer chọn không đủ tiền cho đơn hàng
  - Hệ thống yêu cầu customer phải nạp thêm tiền
  - Tiếp tục tại use case “Recharge”
- A3: Ở bước 4, Customer muốn trả tiền mặt
- Customer chọn “Pay by cash”
  - Cashier thu tiền và thối lại cho khách
  - Tiếp tục tại bước 6 trong Normal Flow
-

*Exception Flow*

E1: Ở bước 3, Customer muốn cancel đơn hàng

- Customer chọn “Cancel” đơn hàng
- Kết thúc use case

E2: Ở bước 4, Hình thức thanh toán không hợp lệ

- Payment method được chọn không hợp lệ
- Tiếp tục tại bước 4 trong Normal Flow

E3: Ở bước 5, Customer nhập sai mật khẩu quá 5 lần

- Customer nhập sai mật khẩu quá 5 lần
- Hệ thống hỏi customer muốn sử dụng “Forget password”
- Nếu chọn: Tiếp tục tại use case “Forgot password”
- Nếu không: Đợi 5 phút. Quay lại bước 5 của Normal Flow
- Nếu tiếp tục sai sau lần đợi thứ 3 thì tài khoản sẽ bị khóa

f. Table Reservation

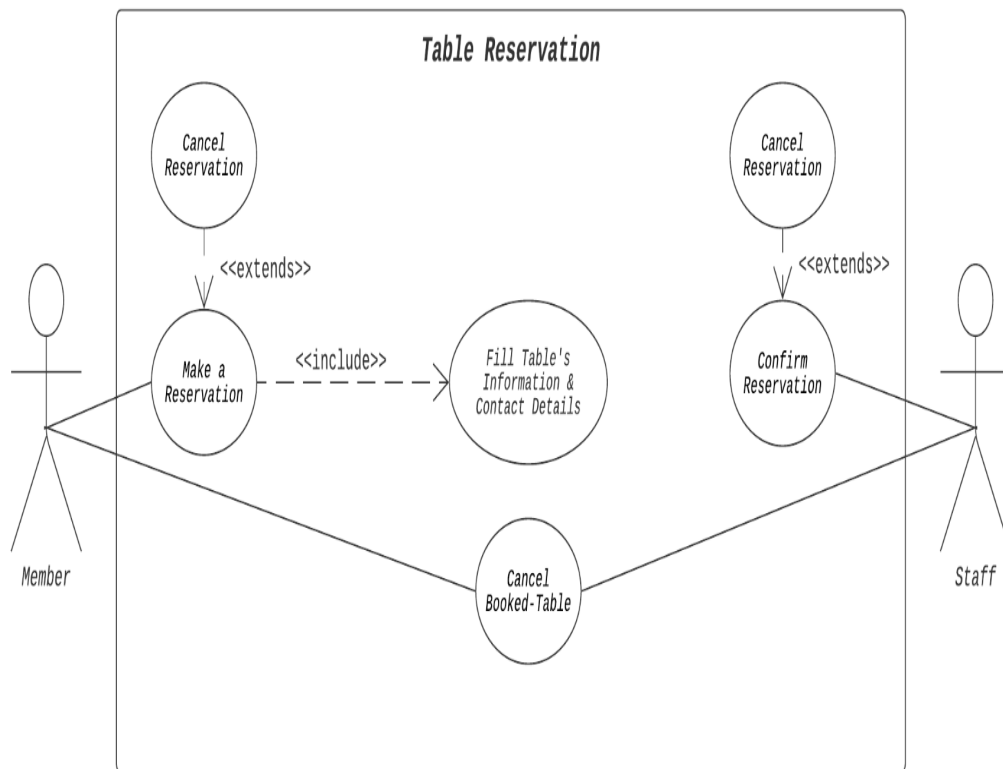


Figure 1.6: Table Reservation use-case diagram

| Actor             | System user   |
|-------------------|---|
| Use case name     | Make a Reservation  |
| Use case overview | Member thực hiện việc đặt bàn   |
| Actor(s)          | Member  |
| Trigger           | Member nhấn chọn “Đặt bàn” trên giao diện   |
| Preconditions     | <ul style="list-style-type: none"> <li>– Member phải có tài khoản đăng ký từ trước</li> <li>– Member đã đăng nhập tài khoản thành công</li> </ul> |
| Postconditions    | Member đặt bàn thành công   |

*Normal Flow*

1. Member nhấn chọn “Đặt bàn” trên giao diện
2. Member điền số lượng khách vào ô “Party Size”; chọn ngày và giờ đặt bàn ở ô “Date” và “Time”. Rồi nhấn "Search" để tìm kiếm bàn
3. Dựa vào ngày, giờ và số lượng khách, màn hình hiển thị danh sách các bàn trong đó có các bàn đã được đặt (hiển thị màu đỏ) và các bàn chưa được đặt (hiển thị màu đen)
4. Member chọn 1 trong số các bàn chưa được đặt
5. Ở các ô thông tin của khách bao gồm: Họ và tên đệm, tên, email và số điện thoại
6. Sau khi xác nhận các thông tin đã chính xác thì Member nhấn “Tiếp theo” và hiển thị màn hình xác nhận thông tin
7. Sau khi xác nhận thông tin đặt bàn, nhấn vào checkbox đồng ý với các điều khoản đặt bàn. Member nhấn "Xác nhận"
8. Hiển thị màn hình đặt bàn hoàn tất
9. Đợi phản hồi từ nhà hàng
10. Nhà hàng xác nhận đặt bàn thành công, Use-case kết thúc

---

*Alternative Flow*

- A1: Ở bước 7, Member đọc điều khoản đặt bàn
- Member nhấn chọn "Điều khoản đặt bàn" hoặc "Tiếp tục" và hiển thị màn hình Điều khoản đặt bàn
  - Sau khi đọc điều khoản đặt bàn, nếu member nhấn chọn "Xác nhận"
  - Tiếp tục bước 8 ở Normal Flow
  - Nếu member nhấn chọn "Quay lại"
  - Quay lại bước 7 ở Normal Flow
- A2: Ở bước 7, Member quay lại màn hình thông tin đặt bàn
- Member nhấn chọn "Quay lại"
  - Quay lại bước 2 ở Normal Flow

---

*Exception Flow*

- E1: Bước 2 - 6 Member muốn hủy đặt bàn
- Member nhấn chọn “Hủy đặt bàn”
  - Use-case kết thúc
- E2: Bước 9 Nhà hàng không chấp nhận đặt bàn
- Nhà hàng không chấp nhận việc đặt bàn của Member
  - Use-case kết thúc



## Chapter 2

# System Modeling

## 2.1 Activity Diagram

### a. Login

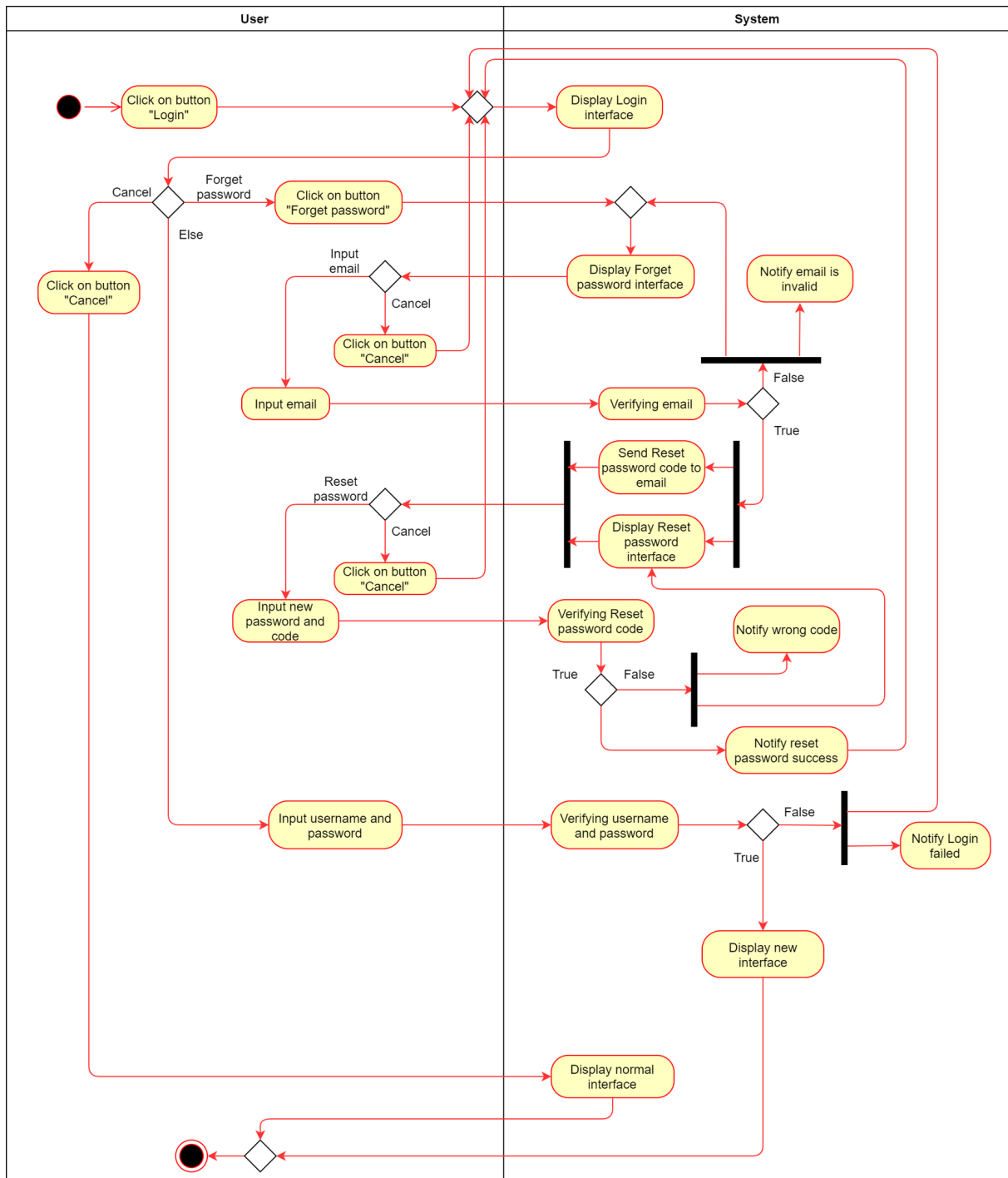


Figure 2.1: Activity Diagram - Login

b. Table Reservation

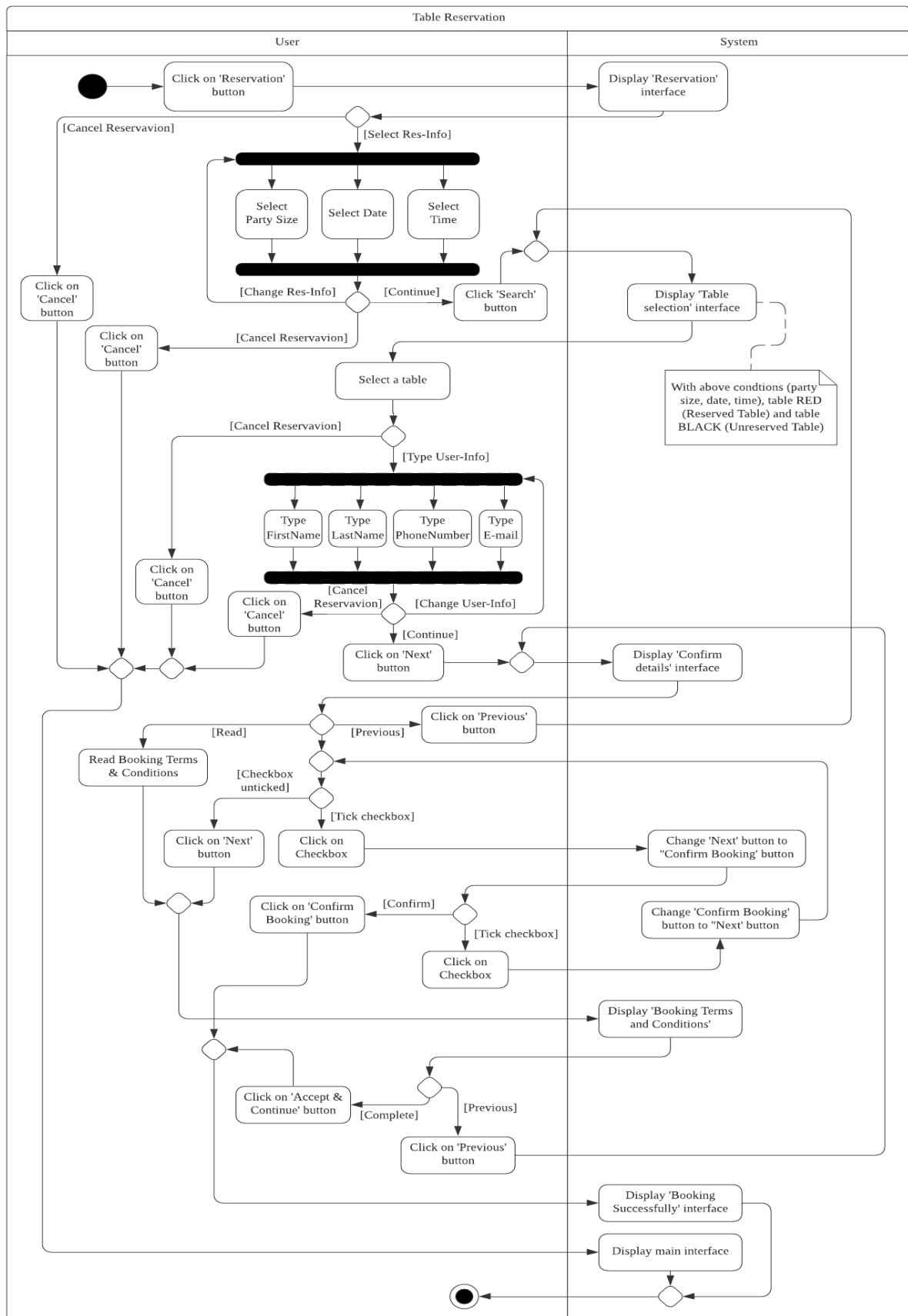


Figure 2.2: Activity Diagram - Table Reservation



c. Register

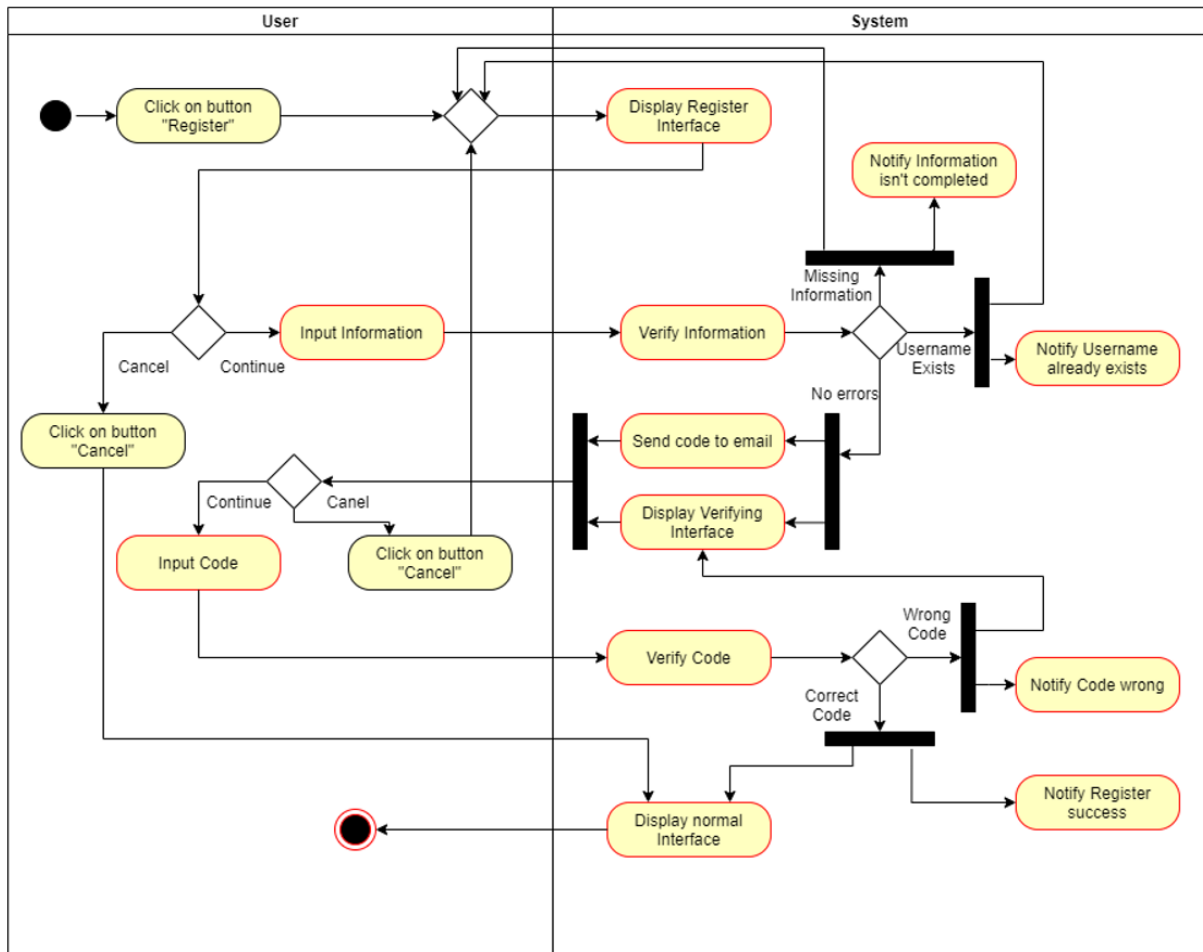


Figure 2.3: Activity Diagram - Register

#### d. Collect Point

Discount Points Collecting and Using Feature .

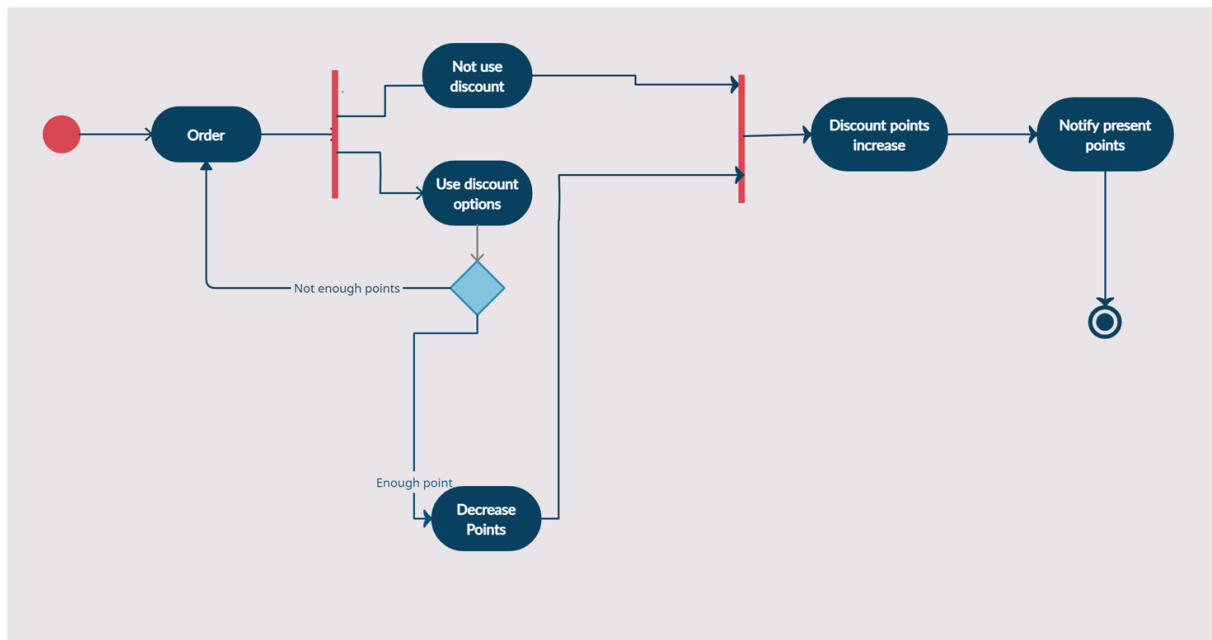


Figure 2.4: Activity Diagram - Collect Point

e. Food Ordering

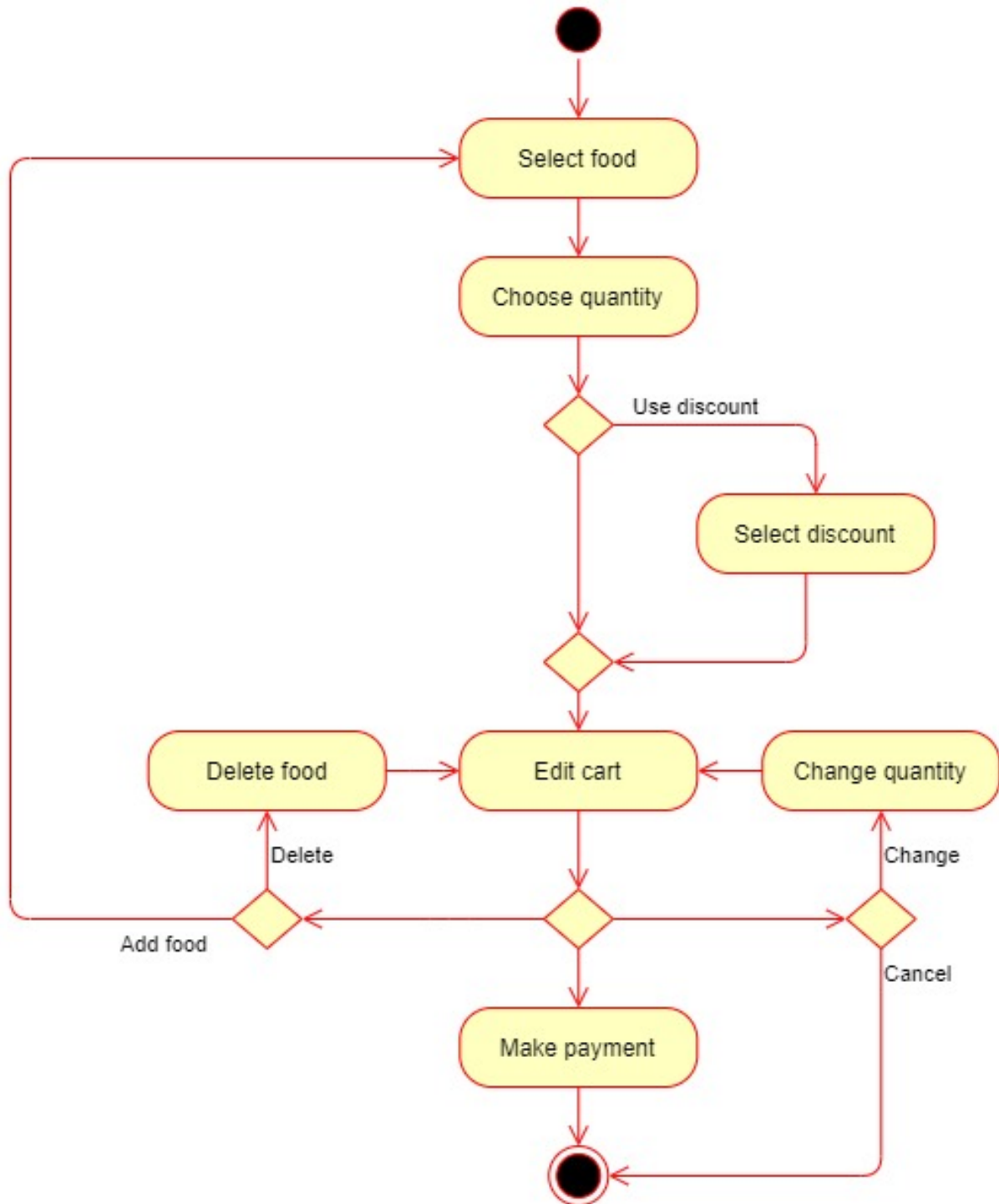


Figure 2.5: Activity Diagram - Food Ordering

f. Make Payment

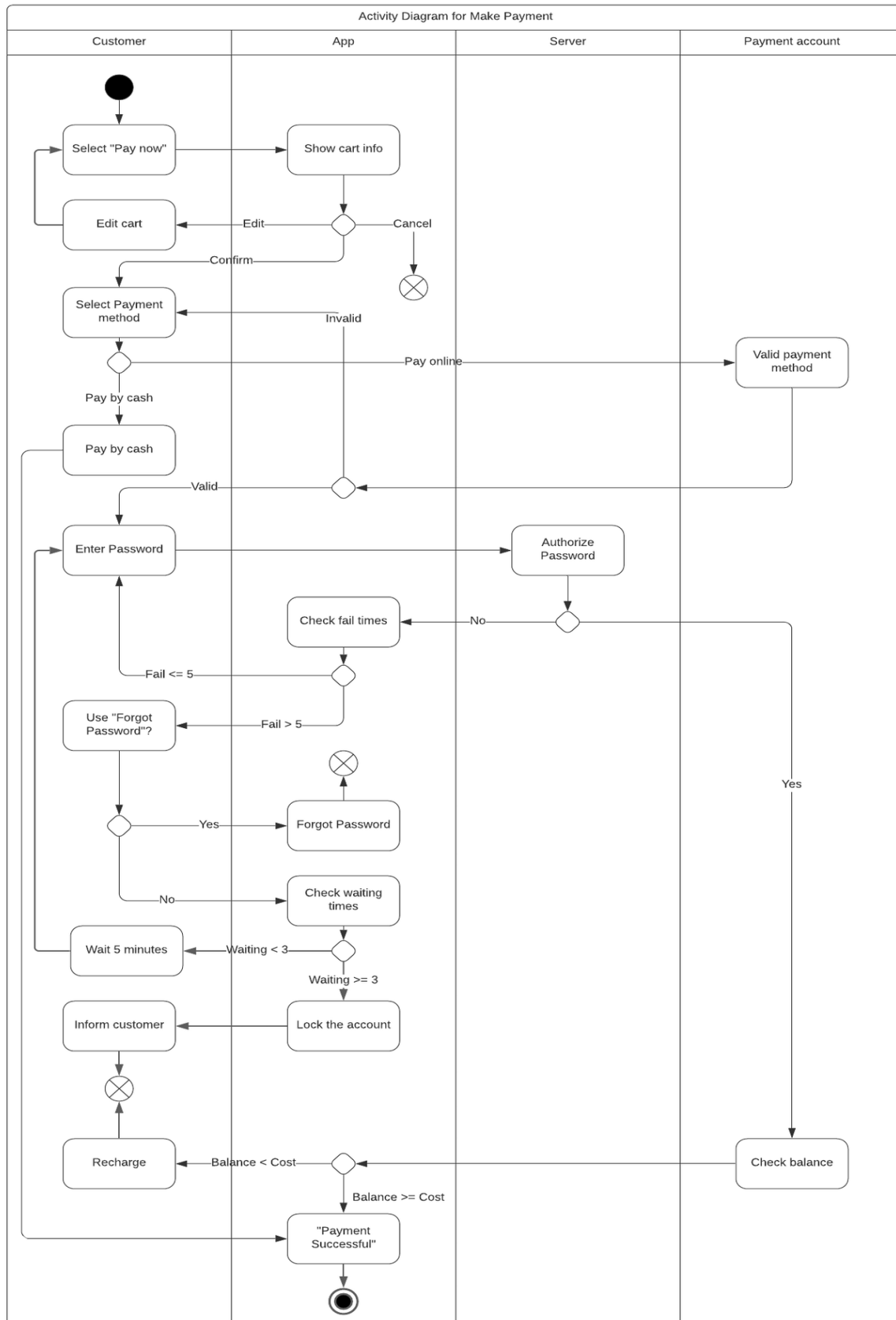


Figure 2.6: Activity Diagram - Make Payment

## 2.2 Sequence Diagram

### a. Login

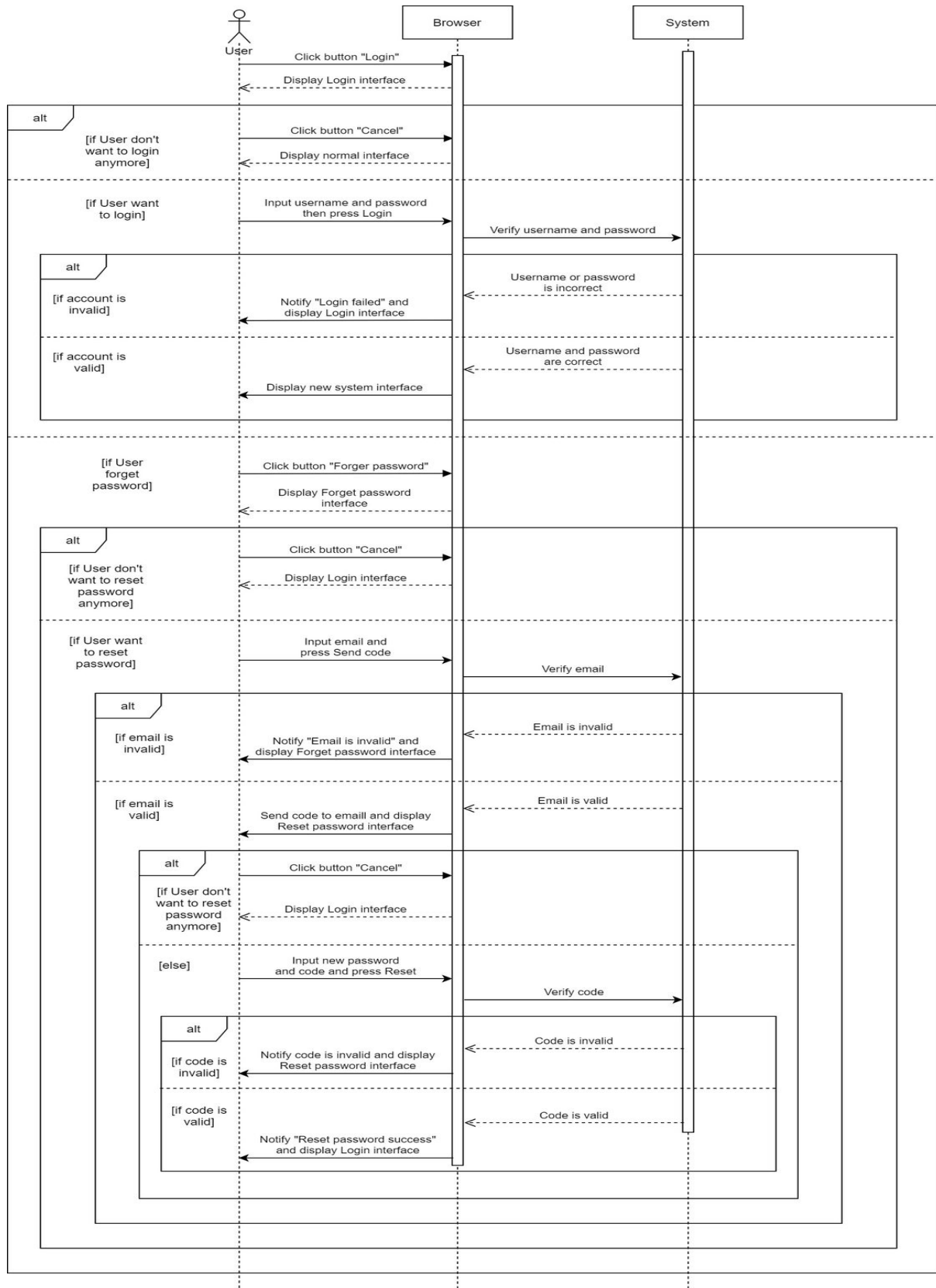


Figure 2.7: Sequence Diagram - Login

## b. Table Reservation

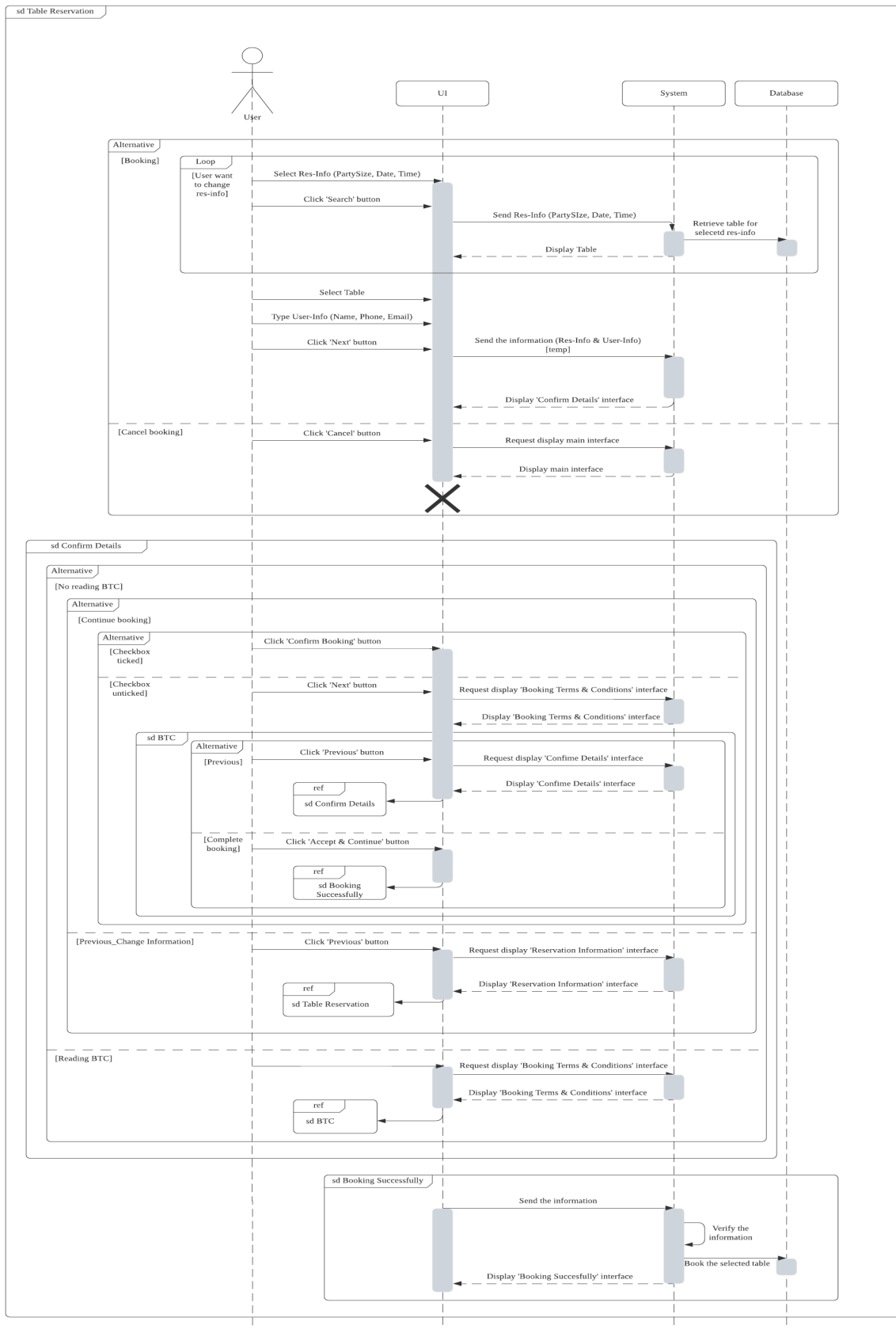


Figure 2.8: Sequence Diagram - Table Reservation

## c. Register

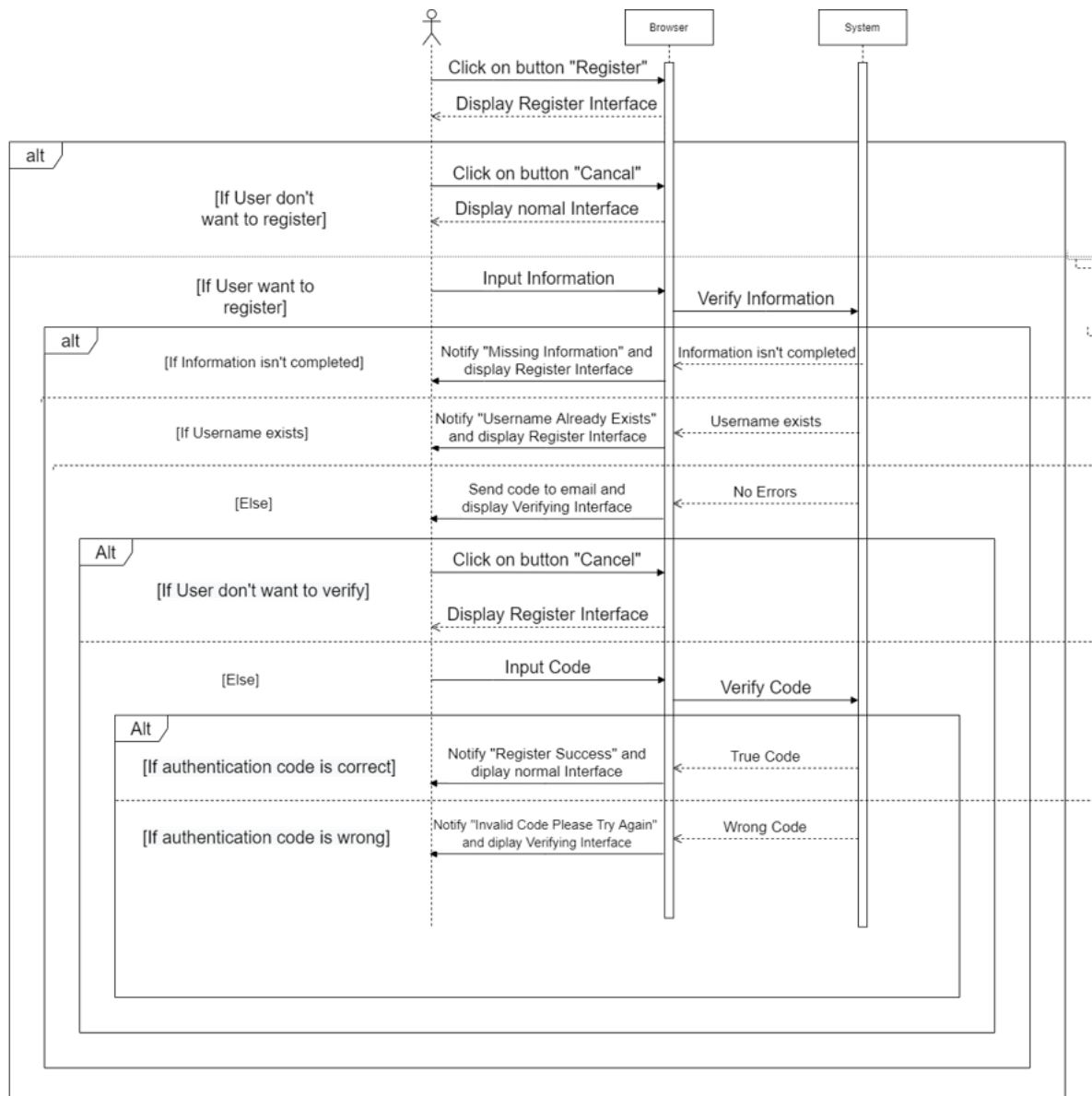


Figure 2.9: Sequence Diagram - Register

d. Collect Point

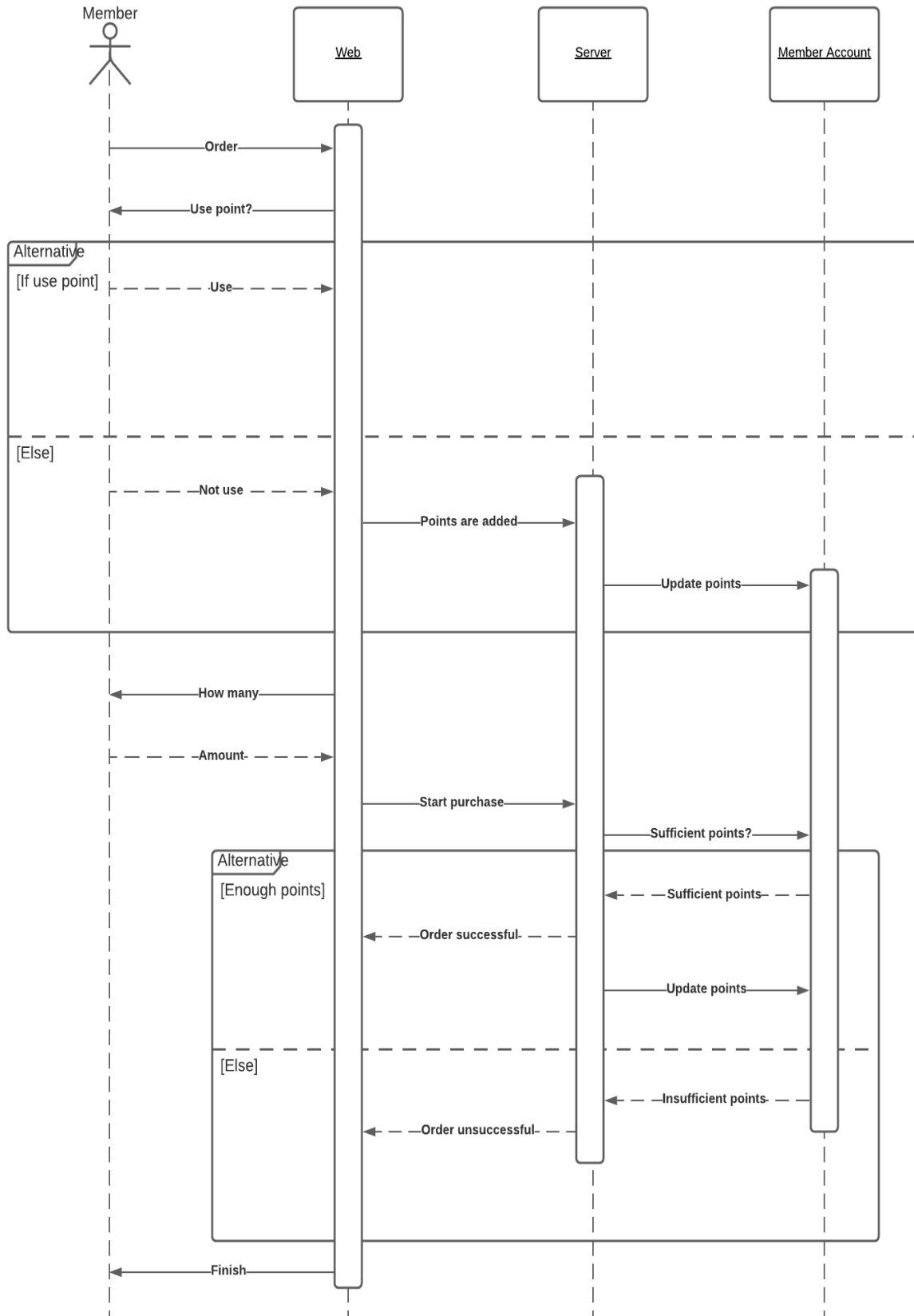


Figure 2.10: Sequence Diagram - Collect Point



e. Food Ordering

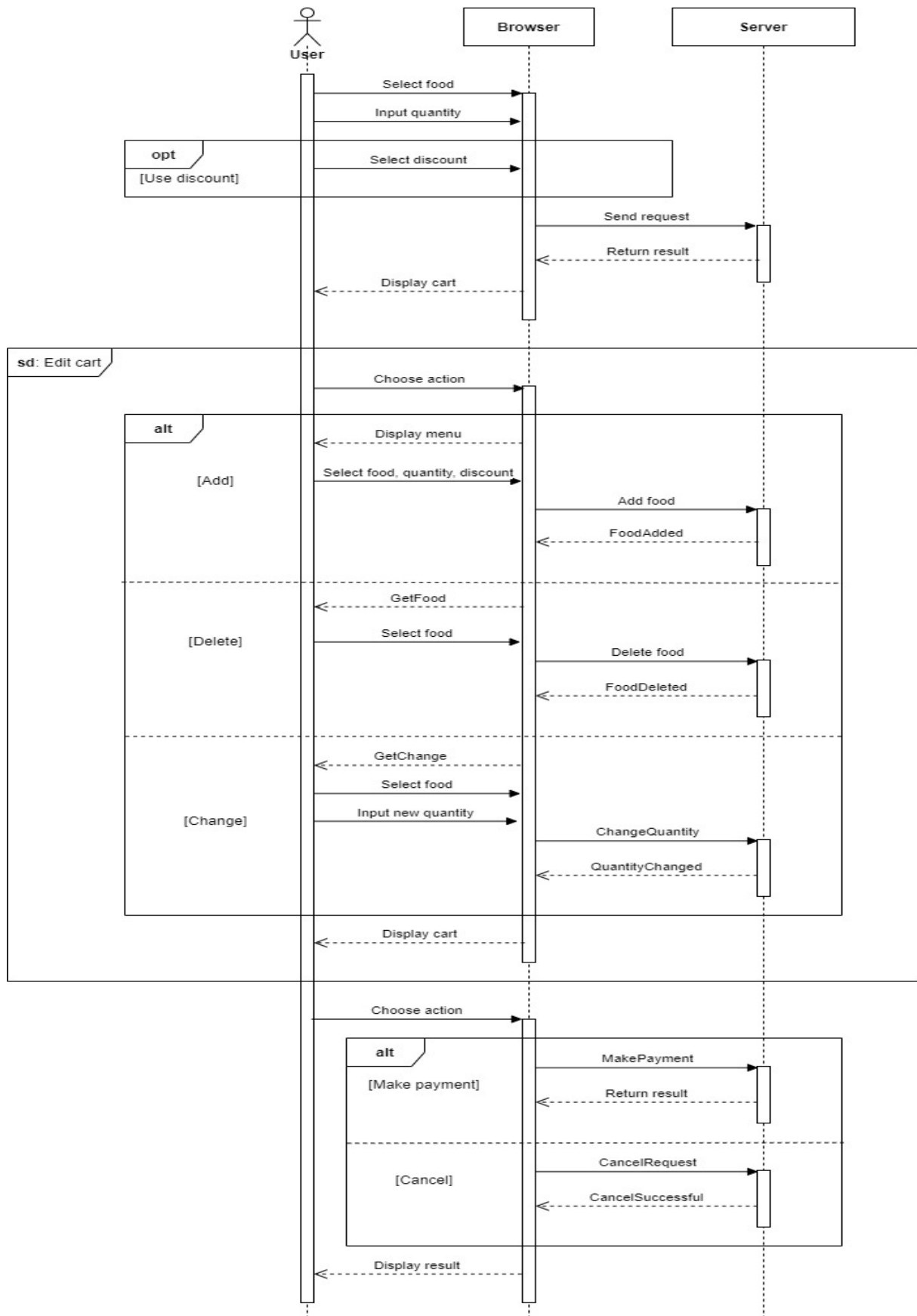


Figure 2.11: Sequence Diagram - Food Ordering

f. Make Payment

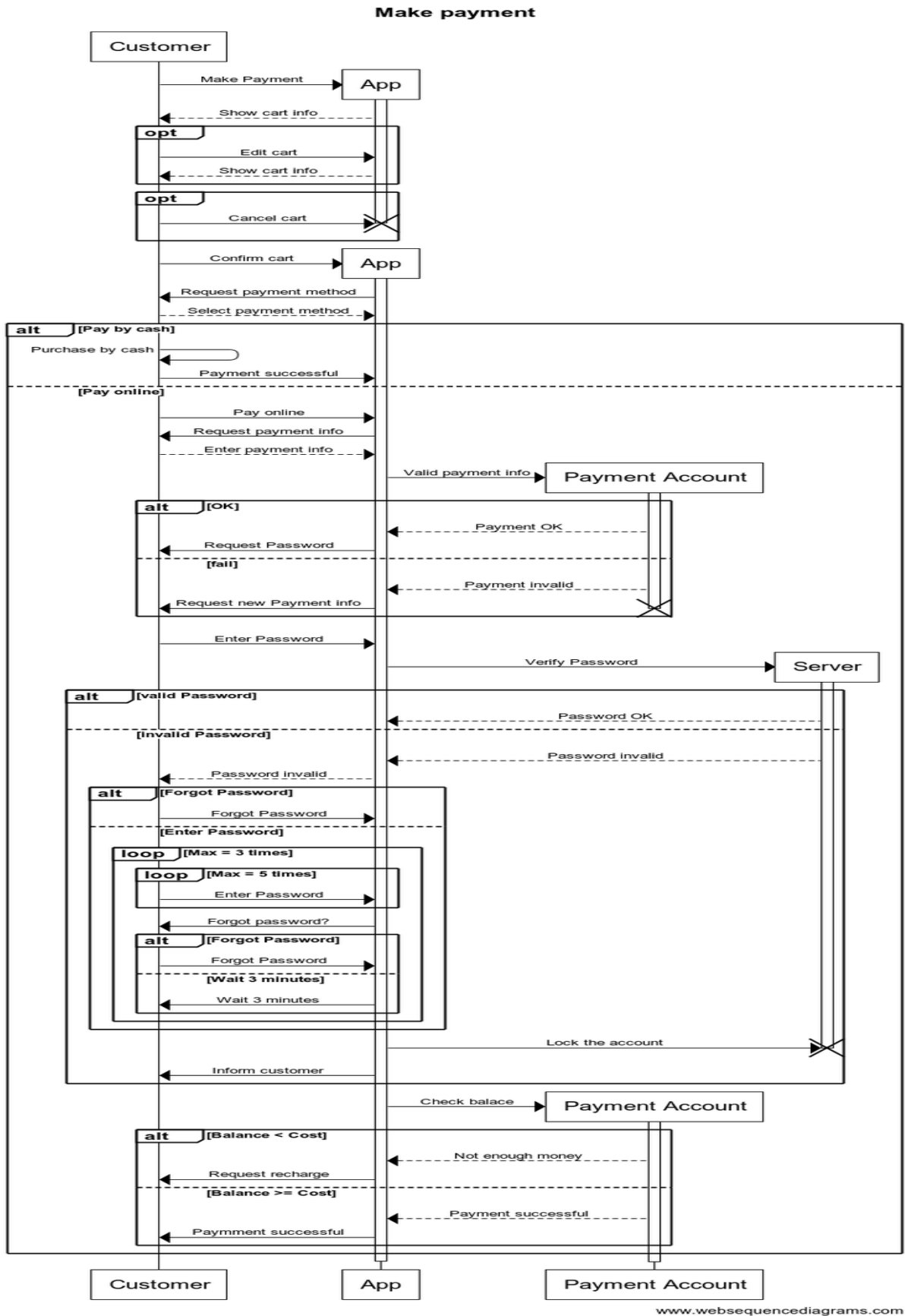


Figure 2.12: Sequence Diagram - Make Payment

## 2.3 Class Diagram

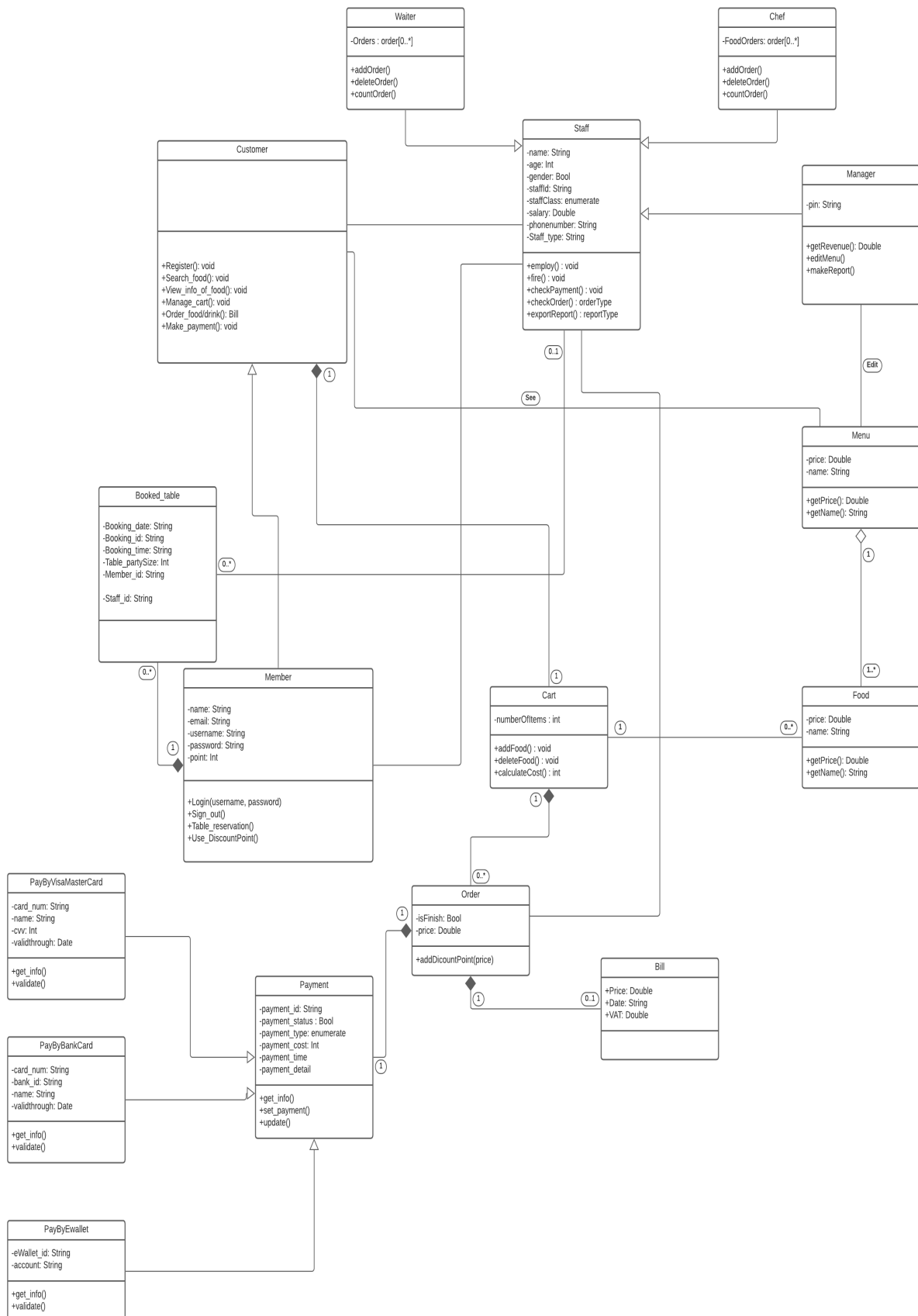


Figure 2.13: Class Diagram

## Chapter 3

# Architecture Design

### 3.1 An architectural approach - MVC

Model-View-Controller (MVC) phân tách một ứng dụng thành ba thành phần logic chính: mô hình, khung nhìn và bộ điều khiển. Mỗi thành phần này được xây dựng để xử lý các khía cạnh phát triển cụ thể của một ứng dụng.

MVC bao gồm 3 thành phần:

- Model: Thành phần Model tương ứng với tất cả logic liên quan đến dữ liệu mà người dùng làm việc với. Điều này có thể đại diện cho dữ liệu đang được chuyển giữa các thành phần View và Controller hoặc bất kỳ dữ liệu nào khác liên quan đến logic nghiệp vụ.
- View: Thành phần View được sử dụng cho tất cả logic UI của ứng dụng.
- Controller: Bộ điều khiển hoạt động như một giao diện giữa các thành phần Model và View để xử lý tất cả logic nghiệp vụ và các yêu cầu đến, thao tác dữ liệu bằng thành phần Model và tương tác với Views để hiển thị kết quả cuối cùng.

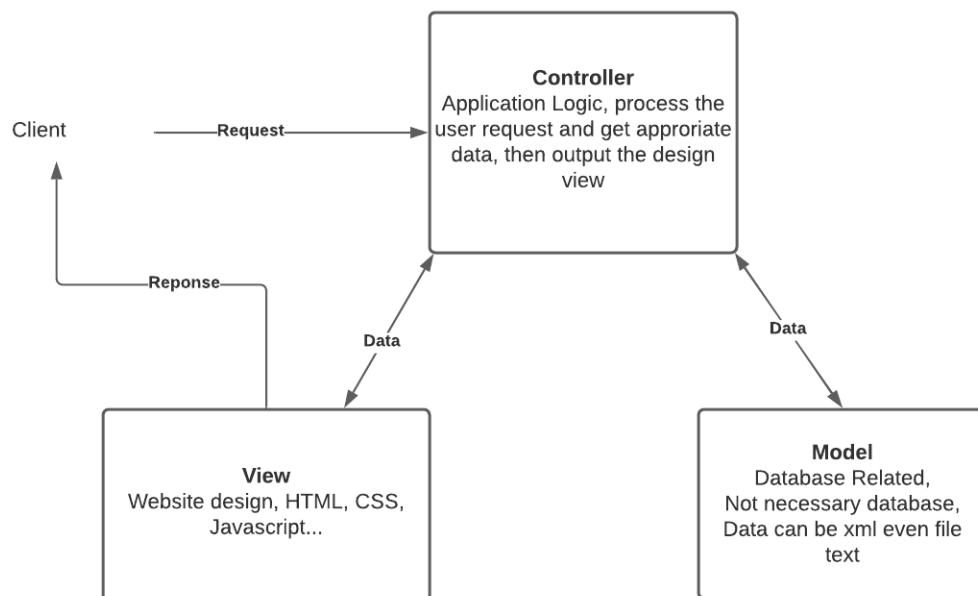


Figure 3.1: MVC pattern for our system

## 3.2 Package Diagram

### a. Login

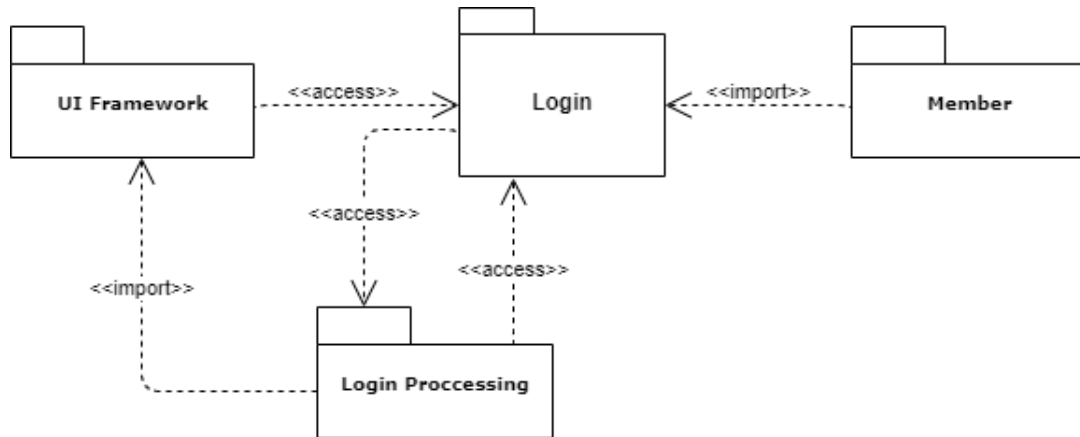


Figure 3.2: Package Diagram - Login

### b. Table Reservation

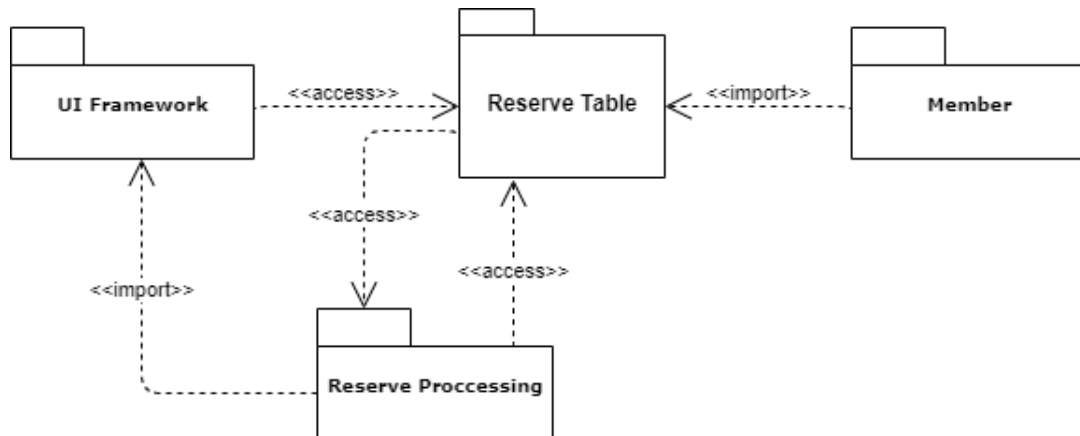


Figure 3.3: Package Diagram - Table Reservation

c. Register

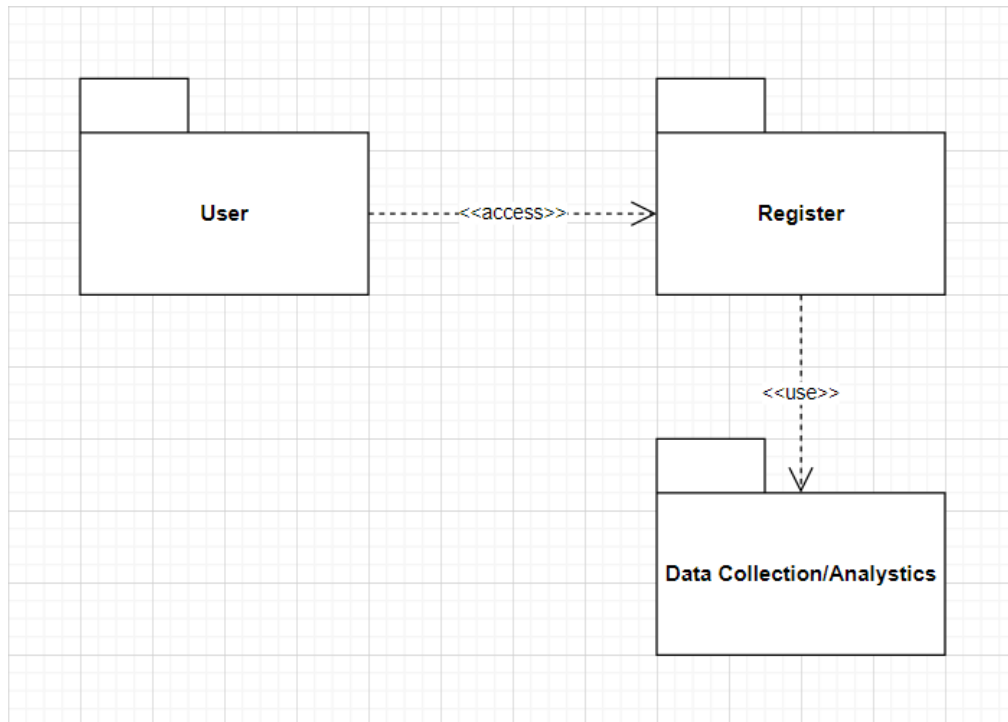


Figure 3.4: Package Diagram - Register

d. Collect Point

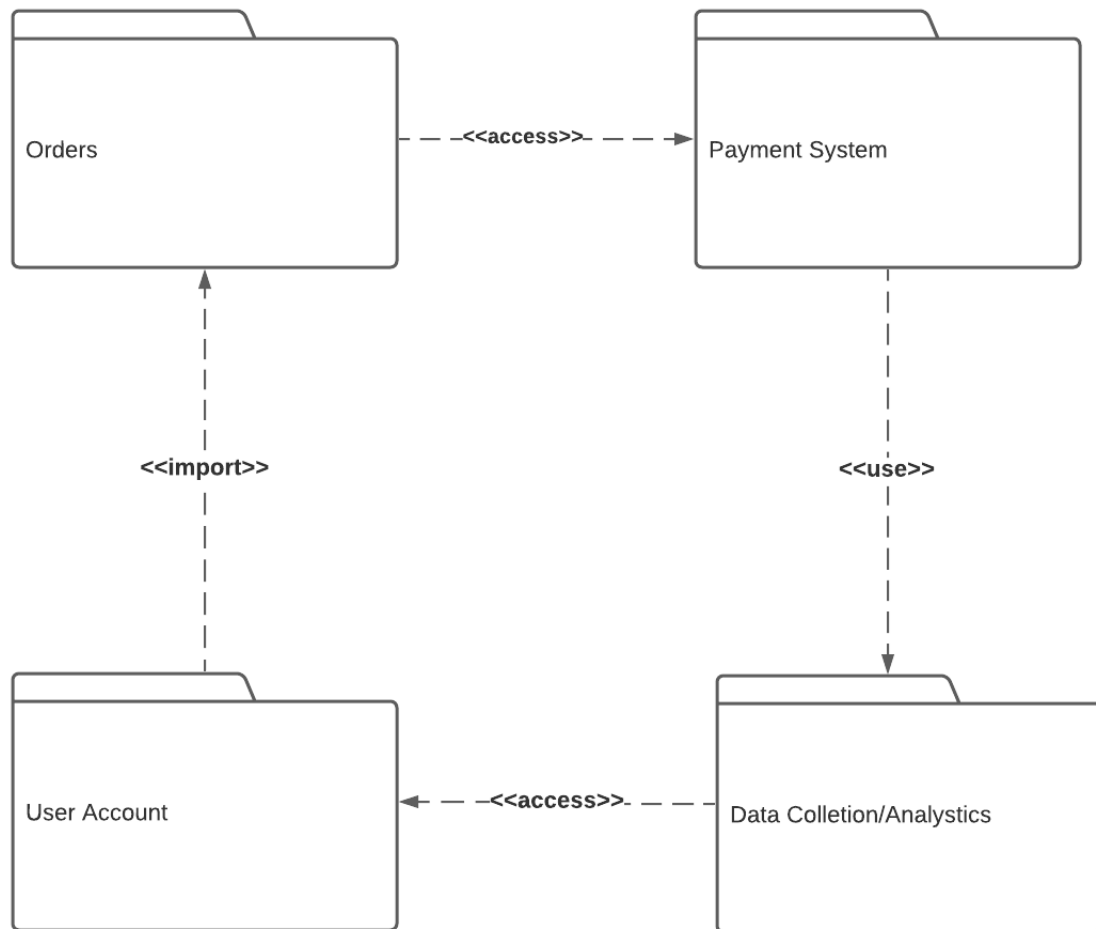


Figure 3.5: Package Diagram - Collect Point



e. Food Ordering

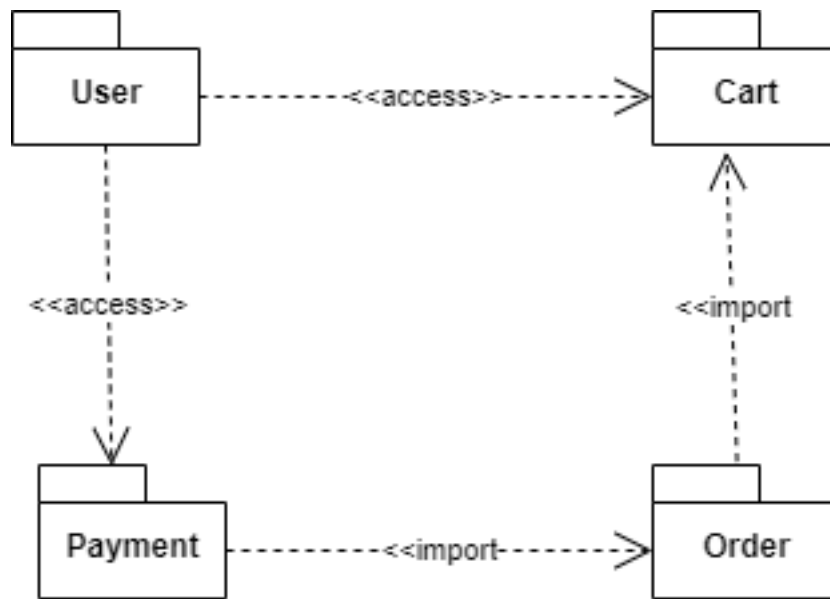


Figure 3.6: Package Diagram - Food Ordering

f. Make Payment

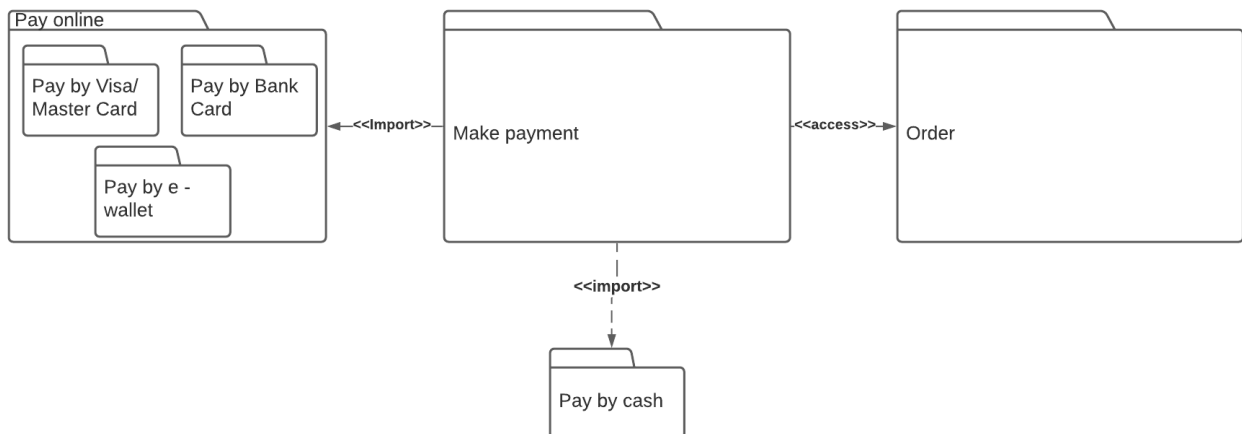


Figure 3.7: Package Diagram - Make Payment