**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**Kỹ thuật lập trình hướng đối tượng**

***Đề tài:* “Xây dựng ứng dụng đặt bàn tiệc tại nhà hàng”**

**Người hướng dẫn: HUỲNH TRUNG TRỤ**

**Lớp: D19CQCNPM01-N**

**Sinh viên thực hiện: LÊ MẬU ANH ĐỨC N19DCCN038**

**TRẦN THƯ ĐẠT N19DCCN036**

**Khoá: 2019**

**Hệ: ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 11/2023**

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**Kỹ thuật lập trình hướng đối tượng**

***Đề tài:* “Xây dựng ứng dụng đặt bàn tiệc tại nhà hàng”**

**Người hướng dẫn: HUỲNH TRUNG TRỤ**

**Lớp: D19CQCNPM01-N**

**Sinh viên thực hiện: LÊ MẬU ANH ĐỨC N19DCCN038**

**TRẦN THƯ ĐẠT N19DCCN036**

**Khoá: 2019**

**Hệ: ĐẠI HỌC CHÍNH QUY**

**TP.HCM, tháng 11/2023**

**LỜI CẢM ƠN**

Trong thời đại công nghệ thông tin ngày nay, việc ứng dụng Công nghệ thông tin vào cuộc sống cũng như các ngành khoa học ngày càng trở nên quan trọng. Quá trình làm đề tài này là bước đầu tiên đi vào thực tiễn và cũng chính là bước đầu thực hành cho những môn học trên ghế nhà trường nói chung và môn học chuyên ngành nói riêng. Nó cũng chính là quá trình nhận xét đánh giá và rút ra ưu, nhược điểm, để từ đó tìm ra phương án tối ưu nhất cho việc học và đi ra thực tế sau này của bản thân.

Em xin chân thành cảm ơn sự hướng dẫn của thầy Huỳnh Trung Trụ đã tạo mọi điều kiện và luôn giúp đỡ, hướng dẫn em tận tình để em hoàn thành đề tài môn thay thế này. Em chân thành cảm ơn các thầy cô luôn sẵn sàng giúp đỡ và tạo mọi điều kiện tốt nhất cho em trong quá trình học tập tại trường.

Mặc dù có nhiều cố gắng bằng toàn bộ kiến thức để hoàn thành công việc, song thời gian và kinh nghiệm của bản thân chưa được trau dồi nhiều nên việc trình bày, phân tích, xây dựng chương trình còn nhiều thiếu sót cần được bổ sung. Vì vậy chúng em rất mong nhận được ý kiến đóng góp của thầy cô để sản phẩm này có thể hoàn thiện, được ứng dụng vào thực tiễn.

Một lần nữa, chúng em xin chân thành cám ơn!

TP. Hồ Chí Minh, tháng 11 năm 2023

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc150721527)

[Chương 1. Giới thiệu 2](#_Toc150721528)

[1.1. Mục đích nghiên cứu 2](#_Toc150721529)

[1.2. Mục tiêu nghiên cứu 2](#_Toc150721530)

[1.3. Phương pháp tiến hành 2](#_Toc150721531)

[Chương 2. Cơ sở khoa học của đề tài 3](#_Toc150721532)

[2.1. Bối cảnh và nhu cầu của đề tài. 3](#_Toc150721533)

[2.2. Công nghệ sử dụng. 3](#_Toc150721534)

[2.2.1. Flutter 3](#_Toc150721535)

[2.2.2. Nodejs 4](#_Toc150721536)

[2.2.3. Socket.io 4](#_Toc150721537)

[2.2.4. Hệ quản trị cơ sở dữ liệu MySQL 4](#_Toc150721538)

[Chương 3. Phân tích hệ thống (PM là một công cụ hỗ trợ) 5](#_Toc150721539)

[3.1. Bối cảnh/ hiện trạng của hệ thống 5](#_Toc150721540)

[3.1.1. Định nghĩa vấn đề (mục tiêu) mà đề tài sẽ giải quyết 5](#_Toc150721541)

[3.1.2. Hiện trạng trước khi sử dụng phần mềm 5](#_Toc150721542)

[3.1.3. Giải pháp của đề tài 6](#_Toc150721543)

[3.2. Định nghĩa các tương tác cần thiết trên PM 7](#_Toc150721544)

[3.2.1. Use-case khách hàng đặt bàn 7](#_Toc150721545)

[3.2.2. Use-case khách hàng thanh toán cọc 9](#_Toc150721546)

[3.2.3. Use-case khách hàng đổi bàn 12](#_Toc150721547)

[3.2.4. Use-case khách hàng huỷ bàn 14](#_Toc150721548)

[3.2.5. Use-case quản lý tư vấn cho khách hàng 15](#_Toc150721549)

[3.2.6. Use-case quản lý tạo menu theo ngày 17](#_Toc150721550)

[3.3. Định nghĩa yêu cầu và ràng buộc đối với phần mềm 19](#_Toc150721551)

[3.3.1. Yêu cầu từ môi trường nghiệp vụ (business) 19](#_Toc150721552)

[3.3.2. Yêu cầu từ môi trường vận hành (operation) 21](#_Toc150721553)

[3.3.3. Yêu cầu từ môi trường phát triển (development) 21](#_Toc150721554)

[Chương 4. Thiết kế phần mềm 22](#_Toc150721555)

[4.1. Lược đồ use-case cho thiết kế phần mềm 22](#_Toc150721556)

[4.1.1. Use-case khách hàng đặt bàn 22](#_Toc150721557)

[4.1.2. Use-case khách hàng thanh toán cọc 23](#_Toc150721558)

[4.1.3. Use-case khách hàng đổi bàn 24](#_Toc150721559)

[4.1.4. Use-case khách hàng huỷ bàn 25](#_Toc150721560)

[4.1.5. Use-case quản lý tư vấn cho khách hàng 25](#_Toc150721561)

[4.1.6. Use-case quản lý tạo menu theo ngày 27](#_Toc150721562)

[4.2. Thiết kế phần mềm để xử lý use-case 28](#_Toc150721563)

[4.2.1. Form 28](#_Toc150721564)

[4.2.2. API 37](#_Toc150721565)

[4.3. Thiết kế cơ sở dữ liệu cho phần mềm 48](#_Toc150721566)

[4.3.1. Mô hình thực thể ERD 48](#_Toc150721567)

[4.3.2. Mô hình thực thể kết hợp ERD 49](#_Toc150721568)

[4.3.3. Thiết kế chi tiết thực thể: 49](#_Toc150721569)

[4.4. Bảng tham chiếu 54](#_Toc150721570)

[Chương 5. Demo chương trình 56](#_Toc150721571)

[5.1 . Đăng ký : 56](#_Toc150721572)

[5.2. Đăng nhập : 56](#_Toc150721573)

[5.3. Danh sách món 57](#_Toc150721574)

[5.4. Danh sách đồ uống 57](#_Toc150721575)

[5.5. Danh sách dịch vụ 58](#_Toc150721576)

[5.6 Danh sách lịch sử đặt bàn 58](#_Toc150721577)

[5.7. Chi tiết đặt bàn. 59](#_Toc150721578)

[5.8. Đặt bàn 59](#_Toc150721579)

[5.9 Nhắn tin 60](#_Toc150721580)

[5.10. Thanh toán tiền đặt cọc : 60](#_Toc150721581)

[Chương 6. KẾT LUẬN 61](#_Toc150721582)

[6.1. Kết quả đạt được 61](#_Toc150721583)

[6.2. Hạn chế 61](#_Toc150721584)

[6.3. Hướng phát triển 61](#_Toc150721585)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1. Quy trình đặt bàn tổng quát 3](#_Toc150372628)

[Hình 3.1. Lược đồ cộng tác yêu cầu đặt bàn trước khi sử dụng phần mềm. 5](#_Toc150372629)

[Hình 3.2. Lược đồ cộng tác yêu cầu đặt bàn sau khi sử dụng phần mềm. 6](#_Toc150372630)

[Hình 3.3. Use-case khách hàng đặt bàn 7](#_Toc150372631)

[Hình 3.4 Mô tả Use-case khách hàng đặt bàn 9](#_Toc150372632)

[Hình 3.5. Use-case khách hàng thanh toán cọc 9](#_Toc150372633)

[Hình 3.6. Mô tả tương tác Use-case thanh toán cọc 11](#_Toc150372634)

[Hình 3.7. Use-case khách hàng đổi bàn 12](#_Toc150372635)

[Hình 3.8 Mô tả tương tác Use-case khách hàng đổi bàn 13](#_Toc150372636)

[Hình 3.9. Use-case khách hàng hủy bàn 14](#_Toc150372637)

[Hình 3.10. Mô tả tương tác Use-case khách hàng hủy bàn 15](#_Toc150372638)

[Hình 3.11. Use-case quản lý tư vấn cho khách hàng 15](#_Toc150372639)

[Hình 3.12. Mô tả tương tác Use-case quản lý tư vấn cho khách hàng 17](#_Toc150372640)

[Hình 3.13. Use-case quản lý tạo menu theo ngày 17](#_Toc150372641)

[Hình 3.14. Mô tả tương tác Use-case quản lý tạo menu theo ngày 19](#_Toc150372642)

[Hình 4.1 Use-case khách hàng đặt bàn 22](#_Toc150372643)

[Hình 4.2 Mô tả tương tác Use-case khách hàng đặt bàn 22](#_Toc150372644)

[Hình 4.3 Use-case khách hàng thanh toán cọc 23](#_Toc150372645)

[Hình 4.4 Mô tả tương tác Use-case khách hàng thanh toán cọc 23](#_Toc150372646)

[Hình 4.5 Use-case khách hàng đổi bàn 24](#_Toc150372647)

[Hình 4.6 Mô tả tương tác Use-case khách hàng đổi bàn 24](#_Toc150372648)

[Hình 4.7 Use-case khách hàng huỷ bàn 25](#_Toc150372649)

[Hình 4.8 Mô tả tương tác Use-case khách hàng huỷ bàn 25](#_Toc150372650)

[Hình 4.9 Use-case quản lý tư vấn cho khách hàng 26](#_Toc150372651)

[Hình 4.10 Mô tả tương tác Use-case quản lý tư vấn cho khách hàng 26](#_Toc150372652)

[Hình 4.11 Use-case quản lý tạo menu theo ngày 27](#_Toc150372653)

[Hình 4.12 Mô tả tương tác Use-case quản lý tạo menu theo ngày 27](#_Toc150372654)

[Hình 4.13 Form đăng nhập 28](#_Toc150372655)

[Hình 4.14. Form đặt bàn 29](#_Toc150372656)

[Hình 4.15. Form danh sách yêu cầu đặt bàn 31](#_Toc150372657)

[Hình 4.16. Form xác nhận thanh toán 33](#_Toc150372658)

[Hình 4.17. Form nhắn tin 36](#_Toc150372659)

[Hình 4.18. Mô hình thực thể kết hợp ERD 49](#_Toc150372660)

**DANH MỤC BẢNG VẼ**

[Bảng 3.1. Mô tả chi tiết Use-case khách hàng đặt bàn 8](#_Toc150372603)

[Bảng 3.2. Mô tả chi tiết Use-case khách hàng thanh toán cọc 10](#_Toc150372604)

[Bảng 3.3. Mô tả chi tiết Use-case khách hàng đổi bàn 12](#_Toc150372605)

[Bảng 3.4. Mô tả chi tiết Use-case khách hàng huỷ bàn 14](#_Toc150372606)

[Bảng 3.5. Mô tả chi tiết Use-case quản lý tư vấn cho khách hàng 16](#_Toc150372607)

[Bảng 3.6. Mô tả chi tiết Use-case quản lý tạo menu theo ngày 18](#_Toc150372608)

[Bảng 3.7 Yêu cầu từ môi trường nghiệp vụ(business) 20](#_Toc150372609)

[Bảng 3.8 Yêu cầu từ môi trường vận hành(operation) 21](#_Toc150372610)

[Bảng 3.9 Yêu cầu từ môi trường phát triển(development) 21](#_Toc150372611)

[Bảng 4.1. API đăng nhập 38](#_Toc150372612)

[Bảng 4.2. API danh sách loại bàn 38](#_Toc150372613)

[Bảng 4.3. API xác nhận đặt bàn. 40](#_Toc150372614)

[Bảng 4.4. API danh sách yêu cầu đặt bàn 41](#_Toc150372615)

[Bảng 4.5. API chi tiết yêu cầu đặt bàn 42](#_Toc150372616)

[Bảng 4.6. API xác nhận thanh toán 42](#_Toc150372617)

[Bảng 4.7. API huỷ đặt bàn 43](#_Toc150372618)

[Bảng 4.8. API đổi bàn 44](#_Toc150372619)

[Bảng 4.9. API chấp nhận tin nhắn 45](#_Toc150372620)

[Bảng 4.10. API danh sách hội thoại 46](#_Toc150372621)

[Bảng 4.11. API danh sách tin nhắn 46](#_Toc150372622)

[Bảng 4.12. API nhắn tin 47](#_Toc150372623)

[Bảng 4.13. API tạo menu theo ngày 48](#_Toc150372624)

[Bảng 4.14. Bảng tham chiếu 51](#_Toc150372625)

LỜI MỞ ĐẦU

Nhu cầu thực tế của xã hội đòi hỏi con người phải luôn năng động và sáng tạo để tạo ra ngày càng nhiều sản phẩm cho xã hội. Các nhà hàng luôn luôn phát triển, các yêu cầu đặt bàn sẽ nhiều lên vì vậy đòi hỏi phải cần rất nhiều nơi lưu trữ yêu cầu đặt bàn tại nhà hàng. Để giảm đi những bất cập trong công tác quản lý món ăn, bàn, dịch vụ và yêu cầu đặt bàn trong nhà hàng, giải pháp hiệu quả nhất hiện nay là đầu tư công nghệ và thiết bị hiện đại, ứng dụng công nghệ thông tin, tự động hoá trong điều hành và quản lý để nâng cao chất lượng phục vụ, đổi mới phương thức quản lý, tăng năng suất hiệu quả. Đó là những nội dung cơ bản đề cập đến trong đề tài “**Xây dựng website đặt bàn tiệc tại nhà hàng**”.

Đề tài “**Xây dựng website đặt bàn tiệc tại nhà hàng**” là một đề tài không phải là mới, hiện tại trên thị trường cũng có rất nhiều các website của các Công ty phần mềm có chức năng tương tự và chuyên nghiệp, nhưng vì mong muốn có thể tìm hiểu một phần kiến thức nhỏ nhoi trong một sa mạc Công nghệ thông tin nên em đã đi vào tìm hiểu về cơ cấu tổ chức và phát triển nhà hàng để có thể viết ra một ứng dụng web quản lý đặt tiệc nhằm phục vụ cho lợi ích của nhà hàng. Ứng dụng web của em có thể chưa hoàn chỉnh và còn nhiều thiếu sót nhưng đó là sự cố gắng của em trong bước đầu hoà nhập vào môi trường làm việc bên ngoài và tiếp xúc chập chững với Công nghệ thông tin. Chúng em mong quý thầy cô trong khoa có thể tạo điều kiện giúp đỡ chúng em hoàn thành đề tài tốt hơn.

# Giới thiệu

## 1.1. Mục đích nghiên cứu

- Tự động hoá quy trình đặt bàn: Không còn cảnh các nhân viên phải tất bật nghe điện thoại, hay email qua lại để tư vấn, mà ứng dụng đã giúp định hình cho khách hàng những thứ mà họ cần và những thứ mà nhà hàng có thể đem lại.

- Tiện ích và linh hoạt từ các công nghệ trực tuyến: Việc có một website đặt bàn sẽ mang lại cho khách hàng một trải nghiệm thuận tiện hơn. Họ có thể xem menu, dịch vụ, đặt bàn và thanh toán trực tuyến, tiết kiệm thời gian và công sức di chuyển.

- Tăng cường quảng cáo và tiếp cận khách hàng: Một ứng dụng chuyên nghiệp không chỉ giúp quảng bá nhà hàng và thương hiệu, mà còn cho phép tiếp cận được khách hàng tiềm năng từ nhiều khu vực khác nhau. Điều này mở ra cơ hội kinh doanh rộng lớn, thu hút đối tượng khách hàng mang tính đa dạng.

- Tăng khả năng cạnh tranh: Trong ngành kinh doanh nhà hàng, việc có một ứng dụng chuyên nghiệp giúp tạo ra sự khác biệt và thu hút khách hàng. Nếu không có mặt trên internet, có thể bị thụt lùi so với các đối thủ khác đã khéo léo tận dụng công nghệ để thu hút khách hàng mới.

## 1.2. Mục tiêu nghiên cứu

Với người quản lý nhà hàng: phần mềm sẽ giúp người quản lý tạo mới, quản lý danh sách món(theo ngày), dịch vụ cộng thêm, tiếp nhận và phản hồi yêu cầu đặt bàn, đổi bàn. Ngoài ra, giúp quản lý lập hoá đơn, tiền trả trước một cách tự động.

Với người đặt tiệc: phần mềm giúp người đặt tiệc xem được menu, dịch vụ của khách hàng cũng như đặt bàn và trao đổi với người quản lý từ xa mà không cần tới nhà hàng.

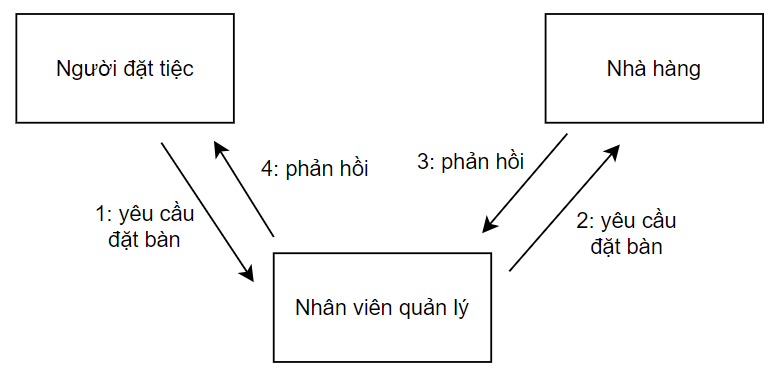
## 1.3. Phương pháp tiến hành

Phân tích và thiết kế theo hướng đối tượng.

# Cơ sở khoa học của đề tài

## 2.1. Bối cảnh và nhu cầu của đề tài.

Quy trình đặt bàn tổng quát:



Hình 2.1. Quy trình đặt bàn tổng quát

Mô tả các bước thực hiện:

1: Yêu cầu đặt bàn (từ người đặt tiệc, menu, số lượng, thời gian).

2: Yêu cầu đặt bàn: được nhân viên quản lý duyệt và phân công cho nhà hàng.

3: Phản hồi (từ nhà hàng): nhà hàng phản hồi đến quản lý nhà hàng: được hoặc không được (phải nêu lý do).

4: Phản hồi: từ nhân viên quản lý đến người đặt tiệc.

## 2.2. Công nghệ sử dụng.

### 2.2.1. Flutter

#### 2.2.1.1. Tổng quan.

Flutter là một mã nguồn mở do Google phát triển và hỗ trợ. Ngôn ngữ dùng để lập trình với Flutter là Dart, là một ngôn ngữ thuần OOP. Mọi thứ trong Flutter đều có một Widget của nó. Có một Stateful-Widget đặc biệt, với sự giúp đỡ của nó mà chúng ta có thể quản lý trạng thái widget và thay đổi nó một cách năng động.

#### 2.2.1.2 Sử dụng.

Như ở trên, mọi thứ trong Flutter đều là widget. Để viết được một ứng dụng Flutter, bạn phải tìm hiểu về:

* Ngôn ngữ Dart, cú pháp: biến, kiểu dữ liệu, hàm, lớp, câu điều kiện, vòng lặp…
* Widget: Có 2 loại widget trong Flutter là stateless-widget và stateful-widget.
* Vòng đời của Widget.
* Nhúng tệp vào chương trình, ví dụ: ảnh, font.
* Sử dụng những “pub”- thư viện được phát triển bởi cộng đồng flutter, nhằm mục đích phục vụ những mục đích nhất định: xây dựng UI, hỗ trợ cấu hình ứng dụng…
* Cách gửi yêu cầu và nhận phản hồi từ API: sử dụng pub “http” hoặc “dio”.

### 2.2.2. Nodejs

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine. JavaScript là một ngôn ngữ hướng sự kiện, do đó bất cứ điều gì xảy ra trên server đều tạo ra một sự kiện non-blocking, do vậy, một website sử dụng NodeJS có thể đáp ứng được việc truy cập cùng lúc một lượng user cực kì lớn.

Được sử dụng để xây dựng website (Back-end) liên kết các role như khách hàng, cửa hàng, người dùng. Kết nối với cơ sở dữ liệu MySQL qua thư viện Sequelize và liên kết với giao diện người dùng các các API.

### 2.2.3. Socket.io

Socket.IO là một thư viện JavaScript mã nguồn mở được sử dụng để xây dựng ứng dụng web thời gian thực (real-time). Nó cung cấp một cách dễ dàng để thiết lập kết nối hai chiều (bidirectional) và truyền thông tin trong thời gian thực giữa máy chủ (server) và khách hàng (client). Socket.IO sử dụng giao thức WebSocket để tạo kết nối giữa máy chủ và khách hàng. Giao thức WebSocket cho phép truyền thông tin trong thời gian thực mà không cần các yêu cầu HTTP mới liên tục. Điều này giúp giảm thiểu độ trễ (latency) và tăng hiệu suất cho các ứng dụng yêu cầu truyền thông nhanh và liên tục.

Trong dự án, socket.io được ứng dụng để làm chức năng nhắn tin giữa người quản lý và khách hàng realtime để có thể kịp thời giải đáp thắc mắc của khách hàng qua đó quá trình đặt bàn sẽ diễn ra nhanh hơn và bớt phân vân cho khách hàng.

### 2.2.4. Hệ quản trị cơ sở dữ liệu MySQL

**MySQL** là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. **MySQL** là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (**SQL**).

Được dùng để thiết kế cơ sở dữ liệu cho phần mềm và lưu trữ những thông tin trên website thông qua NodeJs. Được kết nối với NodeJs thông qua thư viện Sequelize.

# Phân tích hệ thống (PM là một công cụ hỗ trợ)

## 3.1. Bối cảnh/ hiện trạng của hệ thống

Nhà hàng hiện tại đang có nghiệp vụ đặt bàn trước thông qua việc khách hàng gọi điện cho nhà hàng hoặc đến trực tiếp nhà hàng và cung cấp các thông tin cần thiết như số người, ngày đặt, thời gian diễn ra, danh sách món ăn cũng như thứ tự dọn món…

Sau khi nhận được thông tin của khách hàng, quản lí sẽ kiểm tra các món của khách, trạng thái bàn xem có đầy đủ như các yêu cầu khách hàng mong muốn hay không. Nếu như đáp ứng được các yêu cầu đó quản lí sẽ:

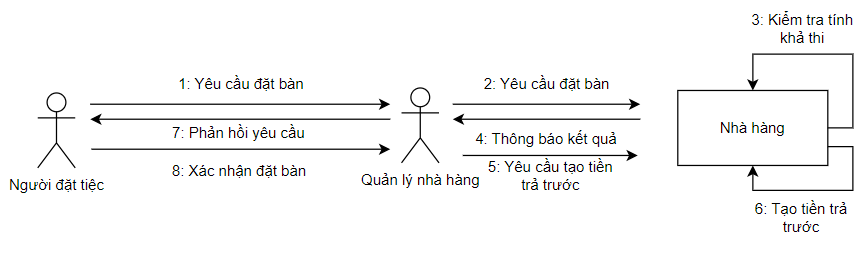
* Thêm lịch hẹn của khách và các thông tin như các món ăn, ngày đặt, giờ đến, số người, đặt bàn số mấy vào Excel.
* Sau khi thêm thì sẽ gọi điện hoặc nhắn tin cho khách hàng đã kiểm tra xác nhận đơn đặt bàn của khách hàng. Nếu như tổng hoá đơn trên 1 triệu đồng thì sẽ yêu cầu khách hàng thanh toán phí cọc trước 30%.
* Sau khi kiểm tra khách thanh toán cọc, quản lý sẽ sửa trạng thái cọc trong Excel. Gọi điện xác nhận là đã thanh toán thành công cho khách hàng.

### 3.1.1. Định nghĩa vấn đề (mục tiêu) mà đề tài sẽ giải quyết

Đề tài sẽ xây dựng một hệ thống thân thiện, tiện dụng để khách hàng có thể tiết kiệm thời gian trong quá trình tìm kiếm món ăn, đồ uống, dịch vụ và tiến hành đặt bàn trong thời gian cụ thể.

Thay vì phải gọi cho nhà hàng cung cấp rất nhiều thông tin và phải đợi nhân viên tính toán tìm kiếm và gọi lại cho mình, thì với hệ thống này,sẽ tự động tự động tiến hành các bước sau đó như kiểm tra bàn đang trống, món ăn trong menu, các dịch vụ có sẵn mà khách hàng yêu cầu và tính toán tiền mà khách hàng phải trả trước và tổng hoá đơn. Quá trình này diễn ra hoàn toàn tự động do hệ thống xử lý vì vậy sẽ tránh sai sót đến từ nhân viên tiếp nhận thông tin từ khách hàng của nhà hàng.

### 3.1.2. Hiện trạng trước khi sử dụng phần mềm



Hình 3.1. Lược đồ cộng tác yêu cầu đặt bàn trước khi sử dụng phần mềm.

1: Yêu cầu đặt bàn (từ người đặt tiệc): phải bao gồm những thông tin: menu, số lượng, thời gian, dịch vụ thêm (nếu có).

2: Yêu cầu đặt bàn (từ quản lý nhà hàng): gửi yêu cầu đặt bàn từ (1) đến nhà hàng để kiếm tra liệu có đáp ứng được không?

3: Kiểm tra tính khả thi: nhà hàng kiểm tra chỗ trống, thời điểm xem có phù hợp với yêu cầu đặt bàn từ (2) hay không?

4: Thông báo kết quả (từ nhà hàng): Nhà hàng thông báo kết quả của (3) đến quản lý nhà hàng: được hoặc không được (phải nêu lý do).

5: Yêu cầu tạo tiền trả trước (từ quản lý nhà hàng): Nếu kết quả trả về từ nhà hàng là thuận lợi, thì người quản lý yêu cầu nhà hàng tính toán tiền trả trước. Một số yêu cầu đặt bàn với số lượng lớn, cần nhiều thời gian chuẩn bị, thì phía người đặt tiệc phải đặt cọc trước một khoản.

6: Tạo tiền trả trước: Nhà hàng tính toán tiền trả trước và thông báo với quản lý nhà hàng.

7: Phản hồi yêu cầu(từ quản lý nhà hàng): Quản lý nhà hàng duyệt và phản hồi lại tới người đặt tiệc dựa trên kết quả kiểm tra từ (4) và tiền trả trước từ (6).

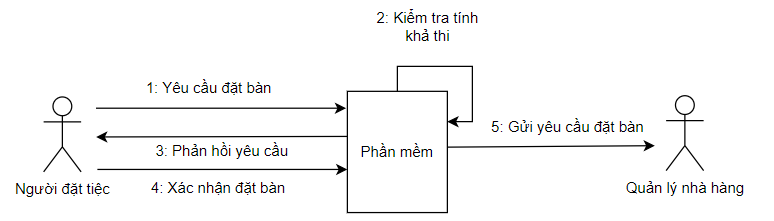
8: Xác nhận đặt bàn (từ người đặt tiệc): Người đặt tiệc xác nhận đặt bàn sau khi nhận phản hồi từ phía nhân viên quản lý, đồng thời thanh toán tiền trả trước (nếu có) hoặc từ chối đặt bàn (nếu thấy phía nhà hàng không đáp ứng đủ yêu cầu).

Kết luận: Có thể thấy rằng, quá trình giao tiếp giữa khách hàng, nhà hàng và người quản lý diễn ra qua nhiều kênh khác nhau, có thể gây ra sự mất mát, sai sót hoặc chậm trễ thông tin.

Người quản lý phải thực hiện nhiều công việc thủ công, như tiếp nhận và phản hồi yêu cầu đặt bàn, đổi bàn, giao việc cho các bên liên quan, lập hoá đơn, … có thể gây ra sai sót hoặc thiếu chính xác.

Khách hàng phải liên lạc trực tiếp với quản lý nhà hàng để đặt bàn, và phải chờ khá lâu để nhận được phản hồi lại. Điều này làm giảm sự thuận tiện và linh hoạt cho khách hàng. Do đó, cần có một giải pháp dùng ứng dụng để cải thiện cách cộng tác này.

### 3.1.3. Giải pháp của đề tài



Hình 3.2. Lược đồ cộng tác yêu cầu đặt bàn sau khi sử dụng phần mềm.

Mô tả yêu cầu đặt bàn:

1: Yêu cầu đặt bàn (từ người đặt tiệc): phải bao gồm những thông tin: menu, số lượng, thời gian, dịch vụ thêm(nếu có).

2: Kiểm tra tính khả thi: phần mềm kiểm tra chỗ trống, thời điểm xem có phù hợp với yêu cầu đặt bàn từ (1) hay không?

3: Phản hồi yêu cầu (từ phần mềm): Sau khi kiểm tra, phần mềm phản hồi lại người đặt tiệc. Nếu kết quả thuận lợi, phải bao gồm tiền trả trước (nếu có), nếu kết quả bất lợi, phải bao gồm lí do.

4: Xác nhận đặt bàn (từ người đặt tiệc): Người đặt tiệc xác nhận đặt bàn sau khi nhận phản hồi từ phần mềm, đồng thời thanh toán tiền trả trước (nếu có).

5: Gửi yêu cầu đặt bàn (từ phần mềm): Sau khi người đặt tiệc xác nhận đặt bàn, phần mềm sẽ gửi thông tin yêu cầu đặt bàn đến với người quản lý nhà hàng.

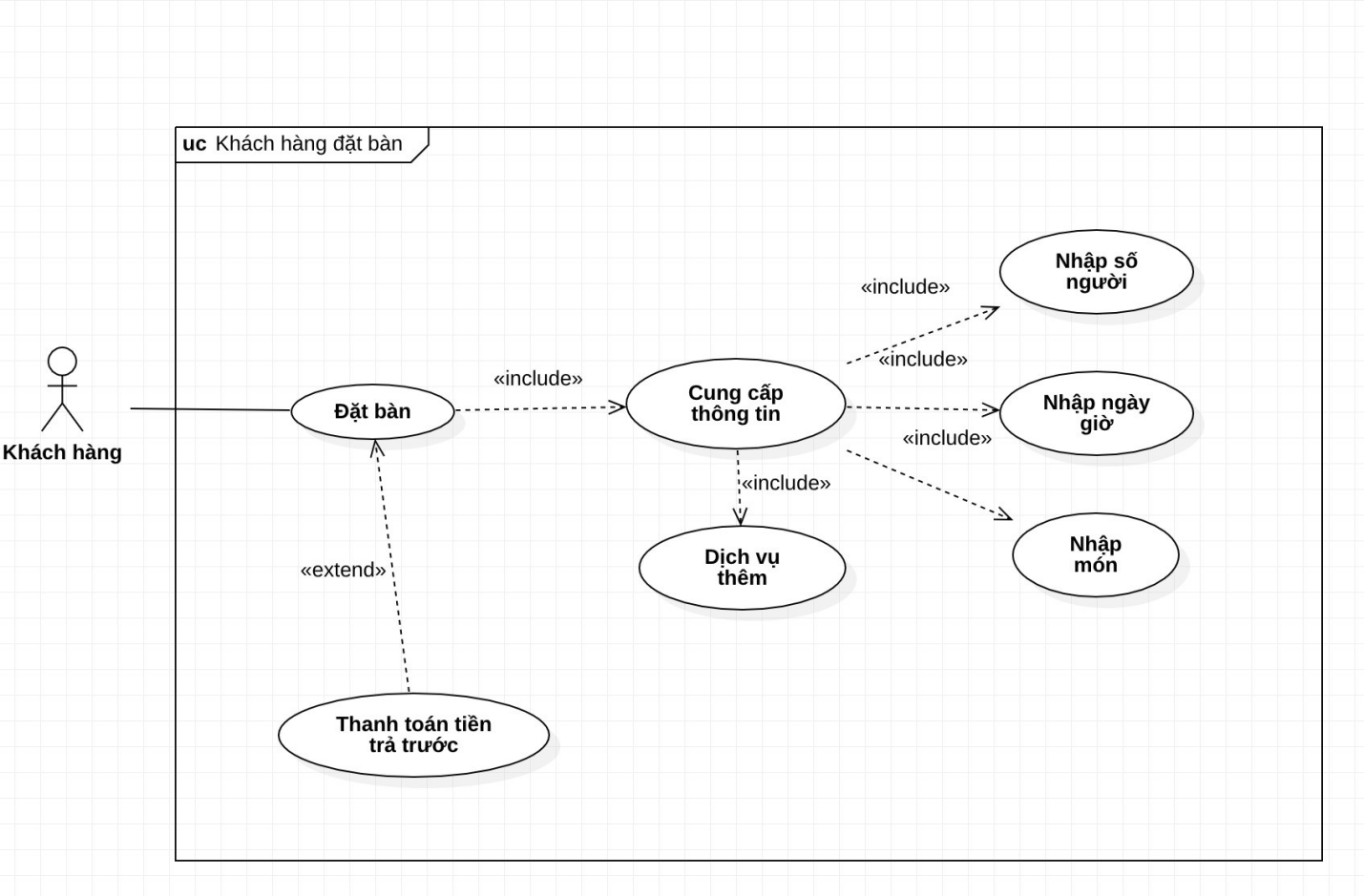
Kết luận: Có thể thấy rằng, quá trình giao tiếp giữa khách hàng, nhà hàng và người quản lý đã trở nên đơn giản hơn, không phải thông qua nhiều bước.

Người quản lý không phải tự thực hiện nhiều công việc thủ công, tránh việc gây ra sai sót hoặc thiếu chính xác.

Khách hàng cũng không cần phải liên lạc trực tiếp với quản lý nhà hàng để đặt bàn, hay phải chờ khá lâu để nhận được phản hồi lại. Điều này làm tăng sự thuận tiện, chủ động, nhanh chóng và linh hoạt cho khách hàng trong việc đặt hay đổi bàn.

## 3.2. Định nghĩa các tương tác cần thiết trên PM

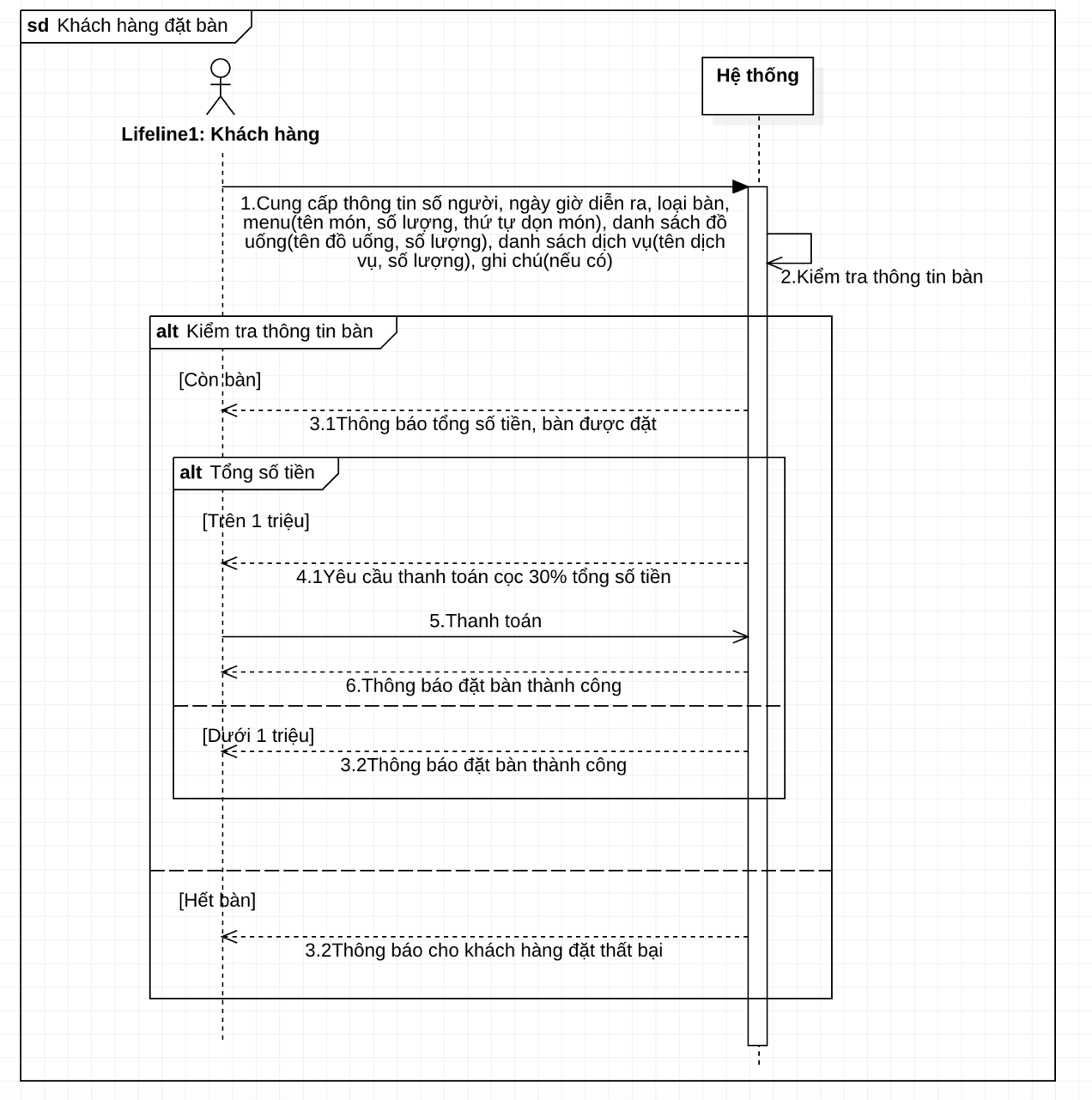
### 3.2.1. Use-case khách hàng đặt bàn



Hình 3.3. Use-case khách hàng đặt bàn

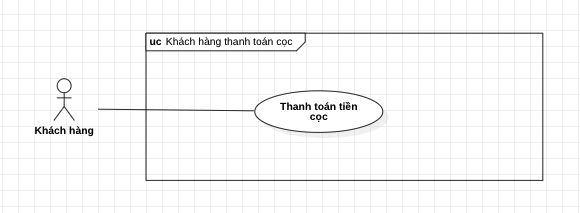
|  |  |
| --- | --- |
| **USECASE NAME** | Khách hàng đặt bàn |
| **USECASE ID** | UC-01 |
| **SCENARIO** | Khách hàng đặt bàn trên hệ thống. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khi khách hàng muốn đặt bàn qua hệ thống, khách hàng cung cấp số lượng, ngày giờ diễn ra, danh sách món, đồ uống, dịch vụ. |
| **TRIGGER** | Khách hàng thực hiện gửi yêu cầu đặt bàn. |
| **PRECONDITION** |  |
| **POSTCONDITION** | Tạo yêu cầu đặt bàn với trạng thái “Xác nhận đặt bàn”. |
| **MAIN FLOW** | 1. Khách hàng cung cấp thông tin đến hệ thống: số người, ngày giờ diễn ra, loại bàn, menu(tên món, số lượng, thứ tự dọn món), danh sách đồ uống(tên đồ uống, số lượng), danh sách dịch vụ(tên dịch vụ, số lượng), ghi chú(nếu có). 2. Hệ thống kiểm tra còn bàn và thông báo thành công và phí trả trước(nếu có) đến khách hàng. 3. Khách hàng thanh toán phí trả trước.(UC-02) |
| **EXCEPTIONS** | 1a. Khách hàng cung cấp thiếu thông tin: số người, ngày giờ diễn ra, loại bàn, menu. Trở về [Main Flow 1].  2a. Hệ thống kiểm tra hết bàn. Hệ thống hiển thị thông báo. Trở về [Main Flow 1]. |

Bảng 3.1. Mô tả chi tiết Use-case khách hàng đặt bàn



Hình 3.4 Mô tả Use-case khách hàng đặt bàn

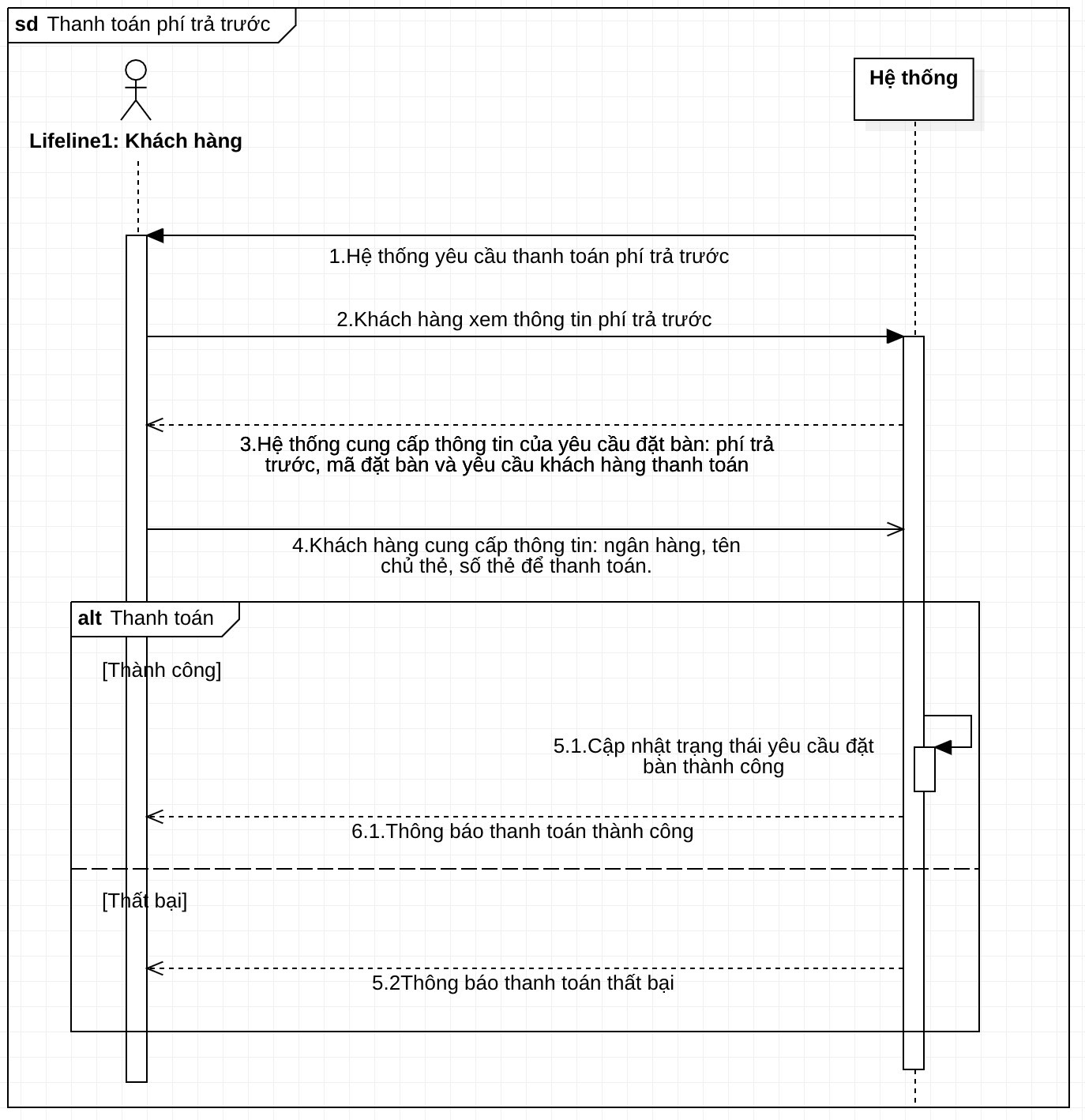
### 3.2.2. Use-case khách hàng thanh toán cọc



Hình 3.5. Use-case khách hàng thanh toán cọc

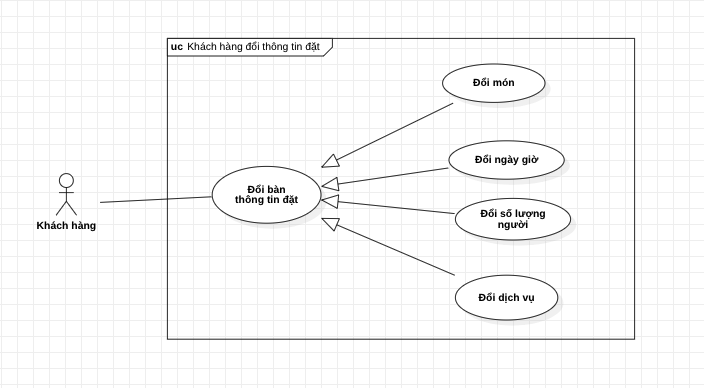
|  |  |
| --- | --- |
| **USECASE NAME** | Khách hàng thanh toán cọc. |
| **USECASE ID** | UC-02. |
| **SCENARIO** | Khách hàng thanh toán cọc qua hệ thống. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Đối với những yêu cầu đặt bàn có thông báo cọc, khách hàng phải thanh toán cọc trước deadline do hệ thống đặt ra. |
| **TRIGGER** | Khách hàng thực hiện thanh toán cọc của yêu cầu đặt bàn. |
| **PRECONDITION** | Yêu cầu đặt bàn đang ở trạng thái “Chưa thanh toán”. |
| **POSTCONDITION** | Yêu cầu đặt bàn chuyển trạng thái “Xác nhận đặt bàn”. |
| **MAIN FLOW** | 1. Hệ thống yêu cầu thanh toán phí trả trước đến khách hàng. 2. Khách hàng gửi yêu cầu xem thông tin: phí trả trước, mã đặt bàn đến hệ thống. 3. Hệ thống cung cấp thông tin của yêu cầu đặt bàn: phí trả trước, mã đặt bàn và yêu cầu khách hàng cung cấp thông tin: ngân hàng, tên chủ thẻ, số thẻ để thanh toán. 4. Khách hàng cung cấp thông tin: ngân hàng, tên chủ thẻ, số thẻ. 5. Hệ thống kiểm tra và thông báo về khách hàng. |
| **EXCEPTIONS** | 4a. Khách hàng cung cấp thông tin sai: Trở về [Main Flow 3]. |

Bảng 3.2. Mô tả chi tiết Use-case khách hàng thanh toán cọc



Hình 3.6. Mô tả tương tác Use-case thanh toán cọc

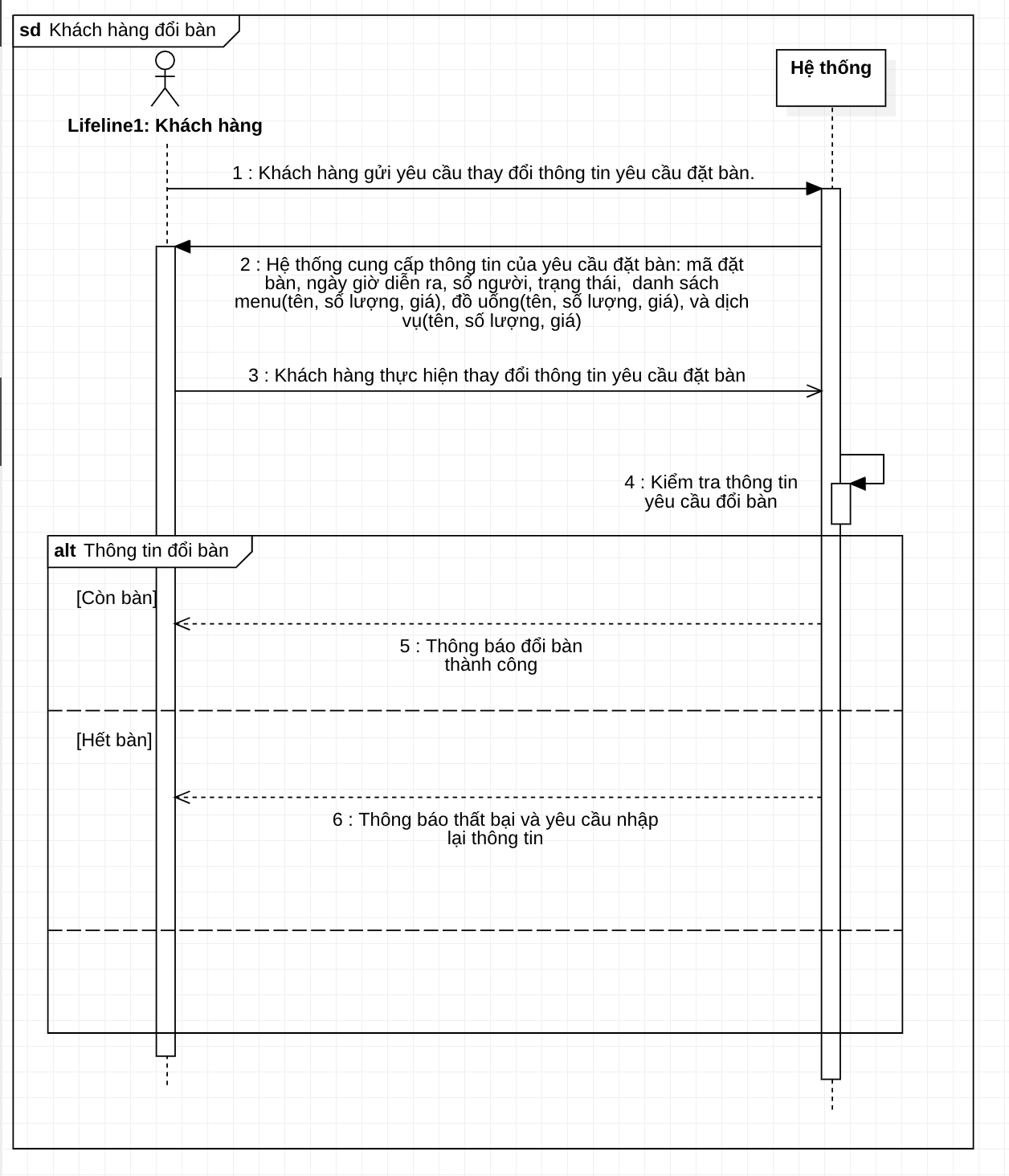
### 3.2.3. Use-case khách hàng đổi bàn



Hình 3.7. Use-case khách hàng đổi bàn

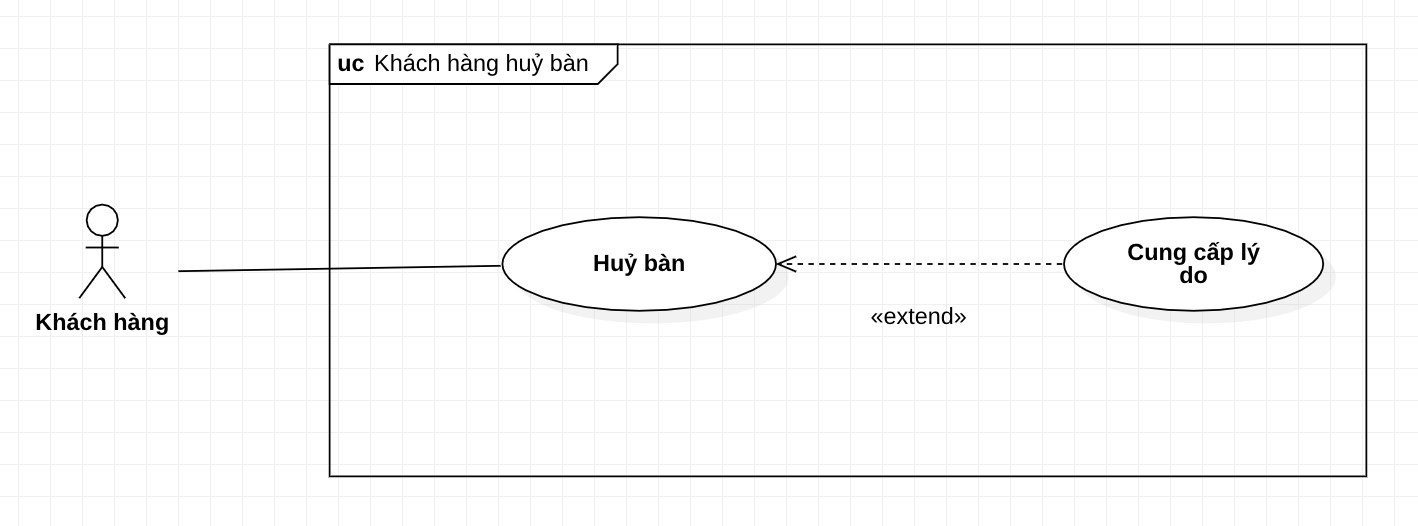
|  |  |
| --- | --- |
| **USECASE NAME** | Khách hàng đổi bàn. |
| **USECASE ID** | UC-03. |
| **SCENARIO** | Khách hàng đổi bàn trên hệ thống. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khách hàng muốn thay đổi thông tin đặt bàn. |
| **TRIGGER** | Khách hàng thực hiện yêu cầu đổi bàn. |
| **PRECONDITION** | Yêu cầu đặt bàn không ở trạng thái “Chưa kết thúc”. |
| **POSTCONDITION** | Yêu cầu đặt bàn được cập nhật. |
| **MAIN FLOW** | 1. Khách hàng gửi yêu cầu thay đổi thông tin yêu cầu đặt bàn.  2. Hệ thống cung cấp thông tin của yêu cầu đặt bàn: mã đặt bàn, ngày giờ diễn ra, số người, trạng thái, ngày tạo yêu cầu, danh sách menu(tên, số lượng, giá, đơn vị tính), đồ uống(tên, số lượng, giá, đơn vị tính), và dịch vụ(tên, số lượng, giá, đơn vị tính).  3. Khách hàng thực hiện thay đổi thông tin yêu cầu đặt bàn.  4. Hệ thống kiểm tra còn bàn vào thời gian khách đổi và thông báo về cho khách hàng. |
| **EXCEPTIONS** | 4a. Hệ thống kiểm tra thông tin: thời gian diễn ra ở quá khứ, hết bàn, thông báo về cho khách hàng. Quay về [Main Flow 3]. |

Bảng 3.3. Mô tả chi tiết Use-case khách hàng đổi bàn



Hình 3.8 Mô tả tương tác Use-case khách hàng đổi bàn

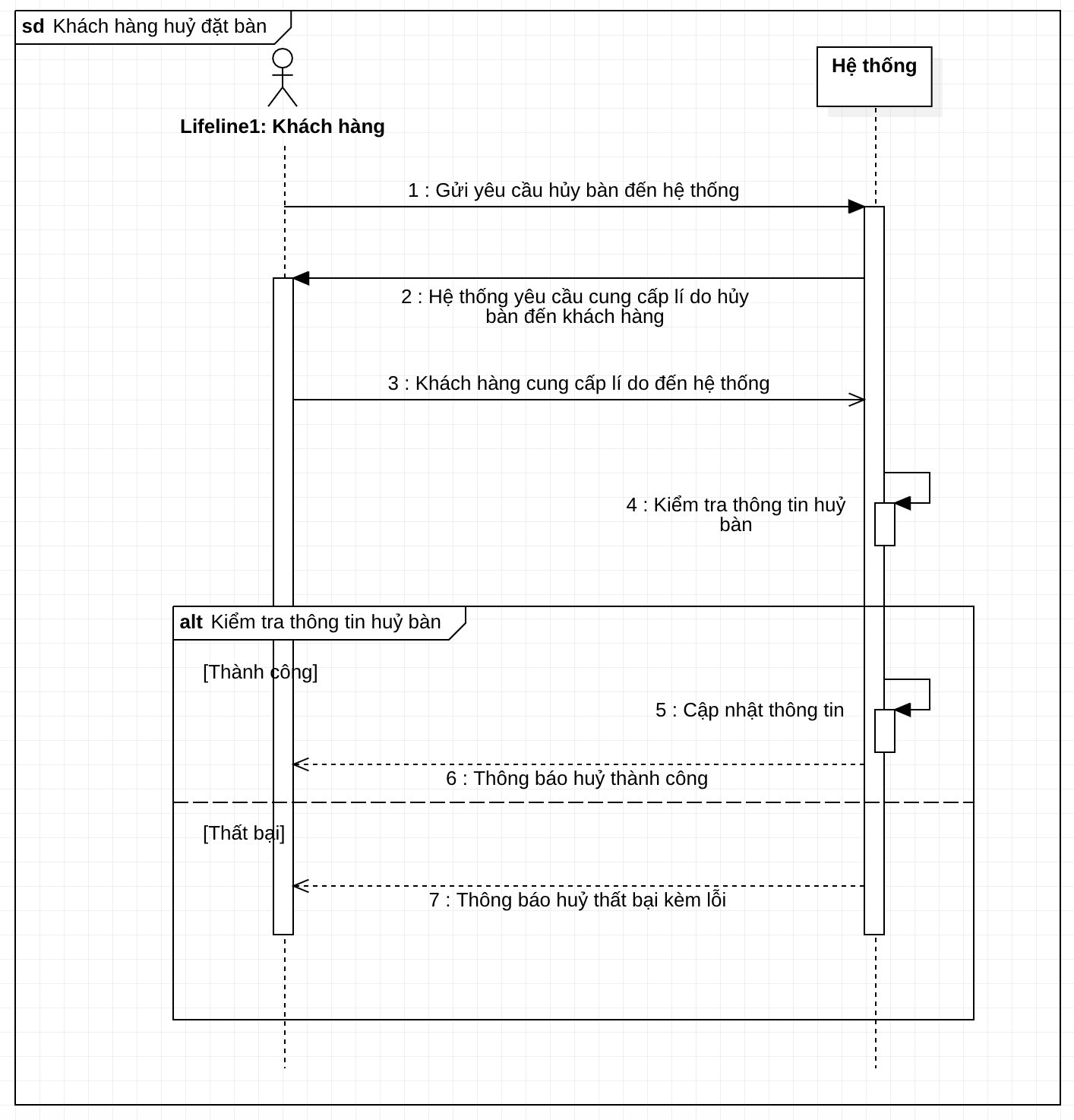
### 3.2.4. Use-case khách hàng huỷ bàn



Hình 3.9. Use-case khách hàng hủy bàn

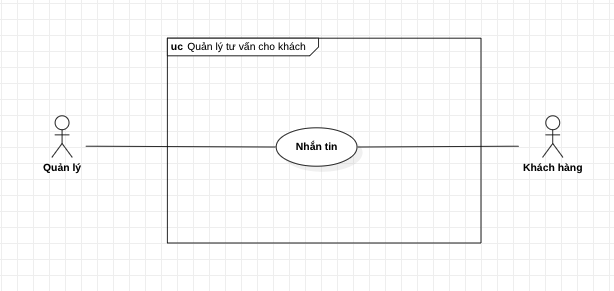
|  |  |
| --- | --- |
| **USECASE NAME** | Khách hàng huỷ bàn. |
| **USECASE ID** | UC-04. |
| **SCENARIO** | Khách hàng huỷ yêu cầu đặt bàn trên hệ thống. |
| **ACTOR** | Khách hàng. |
| **DESCRIPTION** | Khách hàng có thể huỷ bàn sau khi đặt bàn, kèm lý do(nếu có), nhưng mất tiền cọc(nếu có). |
| **TRIGGER** | Khách hàng huỷ bàn. |
| **PRECONDITION** | Yêu cầu đặt bàn không ở trạng thái “Đã kết thúc”. |
| **POSTCONDITION** | Yêu cầu đặt bàn đang chuyển trạng thái “Đã huỷ”. |
| **MAIN FLOW** | 1. Khách hàng gửi yêu cầu huỷ bàn đến hệ thống.  2. Hệ thống yêu cầu cung cấp lí do huỷ bàn đến khách hàng.  3. Khách hàng cung cấp lí do đến hệ thống.  4. Hệ thống kiểm tra và thông báo về cho khách hàng. |
| **EXCEPTIONS** |  |

Bảng 3.4. Mô tả chi tiết Use-case khách hàng huỷ bàn



Hình 3.10. Mô tả tương tác Use-case khách hàng hủy bàn

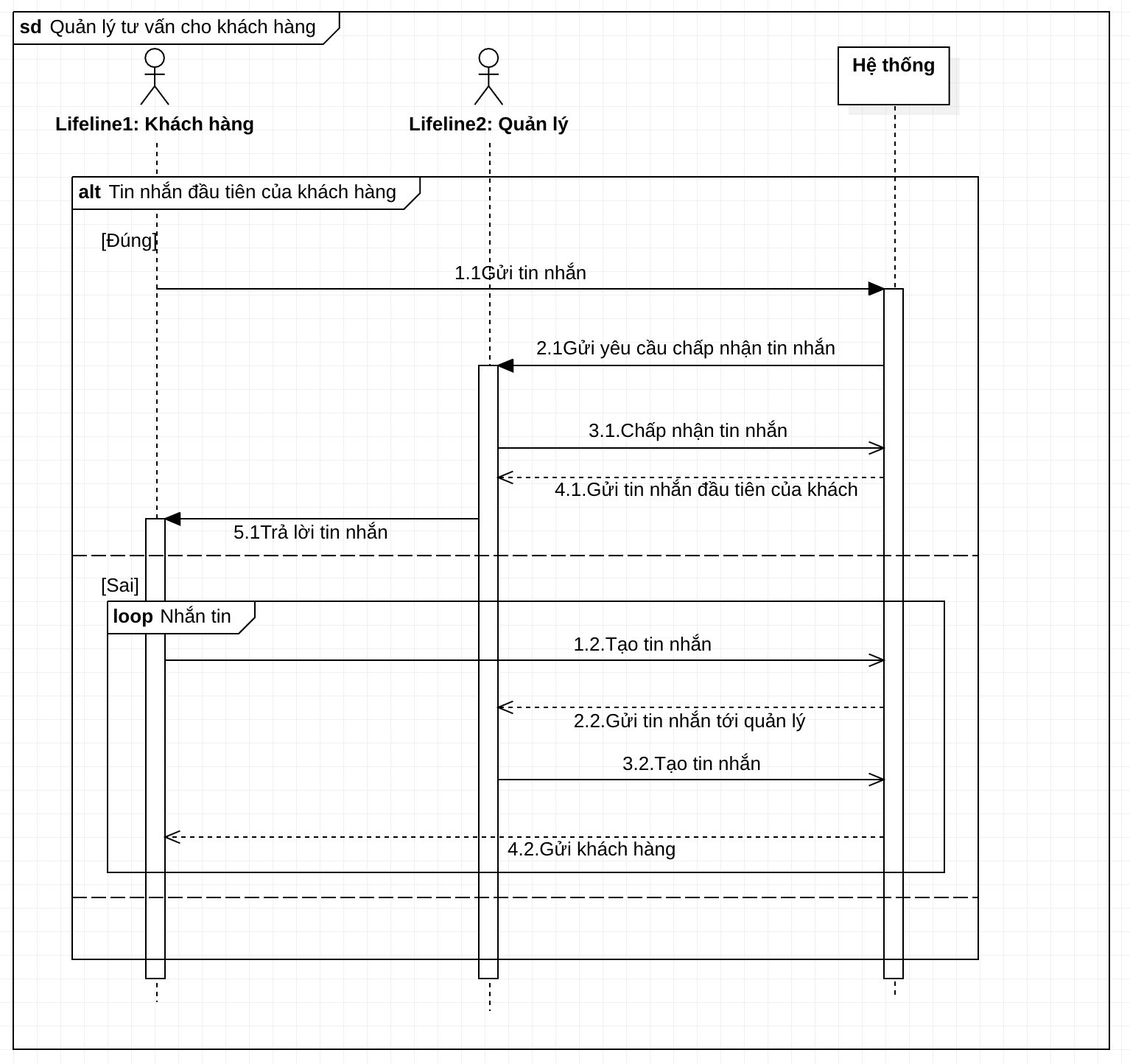
### 3.2.5. Use-case quản lý tư vấn cho khách hàng



Hình 3.11. Use-case quản lý tư vấn cho khách hàng

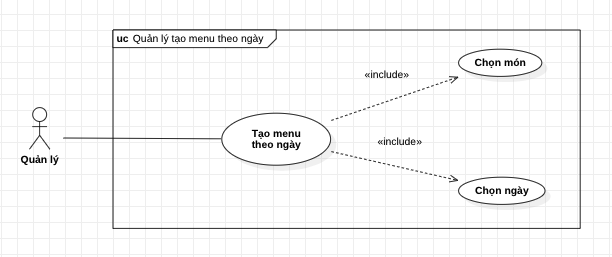
|  |  |
| --- | --- |
| **USECASE NAME** | Quản lý tư vấn cho khách hàng |
| **USECASE ID** | UC-05. |
| **SCENARIO** | Quản lý nhắn tin, trao đổi thông tin với khách hàng qua hệ thống. |
| **ACTOR** | Quản lý, khách hàng. |
| **DESCRIPTION** | Khi khách hàng và quản lý có nhu cầu trao đổi thông tin, giải đáp thắc mắc, khách hàng và quản lý sử dụng tính năng nhắn tin của hệ thống. |
| **TRIGGER** | Khách hàng, quản lý nhắn tin. |
| **PRECONDITION** |  |
| **POSTCONDITION** | Tin nhắn được tạo. |
| **MAIN FLOW** | 1. Khách hàng gửi tin nhắn(nội dung) đến hệ thống. Nếu không phải là tin nhắn đầu tiên, tới bước 4. 2. Hệ thống gửi yêu cầu chấp nhận tin nhắn đến quản lý. 3. Quản lý chấp nhận tin nhắn(khách hàng) từ hệ thống. 4. Hệ thống kiểm tra, gửi tin nhắn(nội dung, thời gian, khách hàng) đến quản lý. 5. Quản lý tạo tin nhắn(nội dung, khách hàng) đến hệ thống. 6. Hệ thống kiểm tra, gửi tin nhắn(nội dung, thời gian) đến khách hàng. |
| **EXCEPTIONS** | 4a. Hệ thống kiểm tra tin nhắn rỗng, quay lại [Main Flow 2]. |

Bảng 3.5. Mô tả chi tiết Use-case quản lý tư vấn cho khách hàng



Hình 3.12. Mô tả tương tác Use-case quản lý tư vấn cho khách hàng

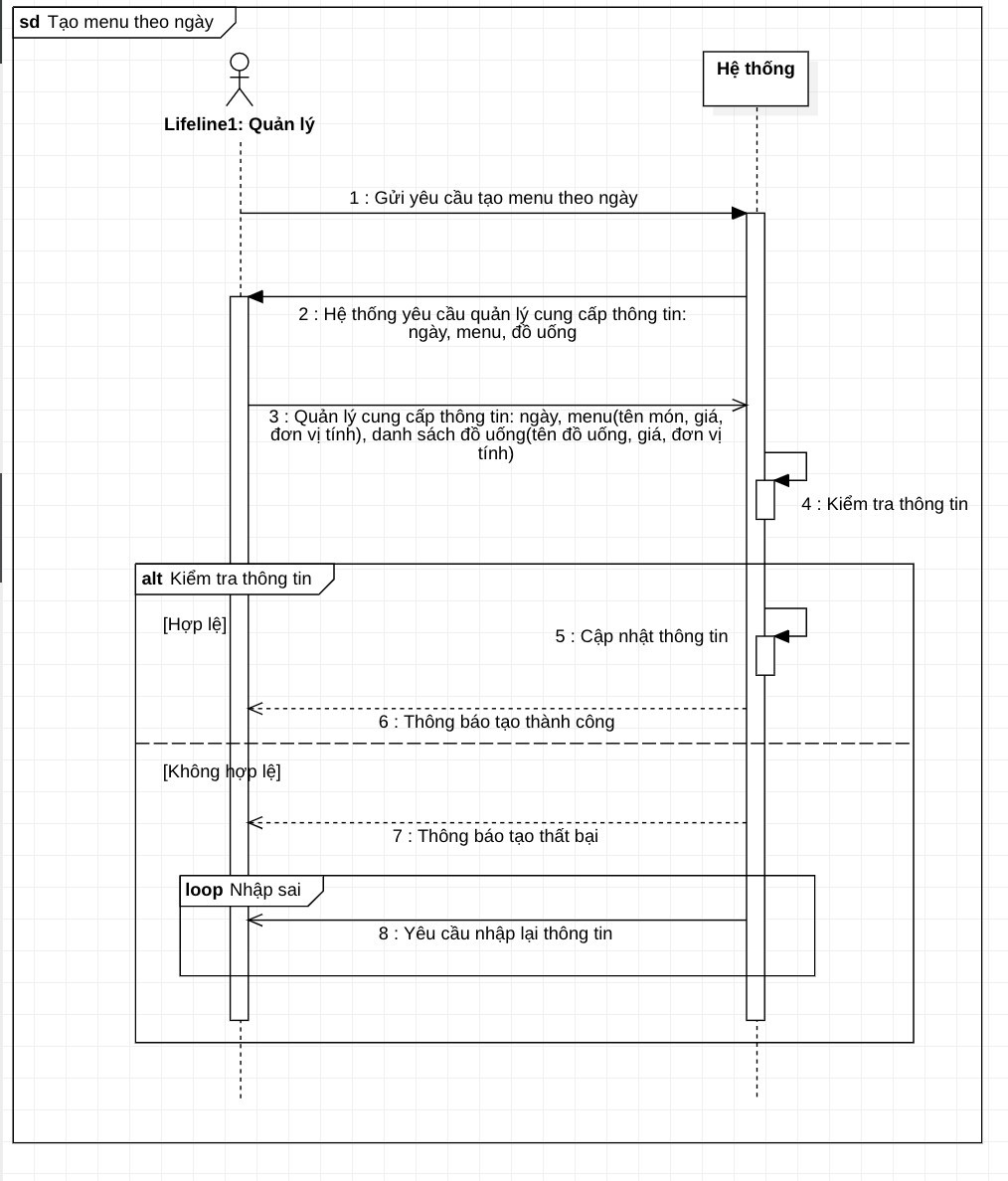
### 3.2.6. Use-case quản lý tạo menu theo ngày



Hình 3.13. Use-case quản lý tạo menu theo ngày

|  |  |
| --- | --- |
| **USECASE NAME** | Quản lý tạo menu theo ngày |
| **USECASE ID** | UC-06 |
| **SCENARIO** | Quản lý tạo menu theo ngày trên hệ thống. |
| **ACTOR** | Quản lý. |
| **DESCRIPTION** | Quản lý tạo menu theo ngày là những món ăn. |
| **TRIGGER** | Quản lý thực hiện yêu cầu tạo menu. |
| **PRECONDITION** | Ngày chưa tồn tại menu. |
| **POSTCONDITION** | Tạo menu mới theo ngày. |
| **MAIN FLOW** | 1. Quản lý gửi yêu cầu tạo Menu theo ngày.   2. Hệ thống yêu cầu quản lý cung cấp thông tin: ngày, menu, đồ uống.  3. Quản lý cung cấp thông tin: ngày, menu(tên món, giá, đơn vị tính), danh sách đồ uống(tên đồ uống, giá, đơn vị tính).  4. Hệ thống kiểm tra, thông báo về quản lý. |
| **EXCEPTIONS** | 4a. Ngày được chọn bị trùng trong quá khứ. Trở về [Main Flow 3].  4b. Hệ thống kiểm tra ngày được chọn đã có menu. Trở về [Main Flow 3]. |

Bảng 3.6. Mô tả chi tiết Use-case quản lý tạo menu theo ngày



Hình 3.14. Mô tả tương tác Use-case quản lý tạo menu theo ngày

## 3.3. Định nghĩa yêu cầu và ràng buộc đối với phần mềm

### 3.3.1. Yêu cầu từ môi trường nghiệp vụ (business)

|  |  |  |  |
| --- | --- | --- | --- |
| **Usecase** | **Req-ID** | **Nội dung yêu cầu** | **Stack-Holder** |
| U01 | B01.1 | Món chưa tồn tại. | Quản lý |
| B01.2 | Quản lý phải cung cấp rõ thông tin món ăn. | Quản lý |
| U02 | B02.1 | Món đã tồn tại. | Quản lý |
| B02.2 | Quản lý phải cung cấp rõ thông tin món ăn. | Quản lý |
| U03 | B03.1 | Menu là duy nhất trong ngày. | Quản lý |
| B03.2 | Menu không được rỗng. | Quản lý |
| U04 | B04.1 | Menu đã được tạo. | Quản lý |
| B04.2 | Menu không được rỗng. | Quản lý |
| U06 | B05.1 | Dịch vụ chưa tồn tại. | Quản lý |
| B05.2 | Quản lý phải cung cấp rõ thông tin dịch vụ. | Quản lý |
| U06 | B06.1 | Dịch vụ đã tồn tại. | Quản lý |
| B06.2 | Quản lý phải cung cấp rõ thông tin dịch vụ. | Quản lý |
| U07 | B07.1 | Nếu yêu cầu không hợp lệ hoặc không khả dụng, quản lý phải liên hệ với người đặt tiệc để giải quyết vấn đề và đề xuất giải pháp. | Quản lý, người đặt tiệc |
| B07.2 | Không thay đổi trạng thái sau khi diễn ra tiệc | Quản lý |
| U08 | B08.1 | Người đặt tiệc cần cung thông tin về yêu cầu đặt bàn: số lượng, thời gian, menu, loại bàn, dịch vụ | Người đặt tiệc |
| B08.2 | Hệ thống cần hiển thị tính khả thi của yêu cầu đặt bàn theo thông tin người đặt tiệc. | Nhà hàng |
| B08.3 | Hệ thống cần giữ bàn trong thời gian diễn ra tiệc trên dưới 4 tiếng. | Nhà hàng |
| B08.4 | Hệ thống không tính yêu cầu đặt bàn không thanh toán phí trả trước(nếu có). | Nhà hàng |
| U09 | B09.1 | Người đặt tiệc đã đặt bàn trước đó. | Người đặt tiệc |
| B09.2 | Thời gian thực hiện đổi bàn phải trước ngày diễn ra lúc đặt bàn ít nhất 3 ngày. | Người đặt tiệc |
| B09.3 | Hệ thống yêu cầu thanh toán thêm tiền trả trước. | Nhà hàng |

Bảng 3.7 Yêu cầu từ môi trường nghiệp vụ(business)

### 3.3.2. Yêu cầu từ môi trường vận hành (operation)

|  |  |  |  |
| --- | --- | --- | --- |
| **Usecase** | **ID** | **Nội dung yêu cầu** | **Stack-Holder** |
| U01, U02, U03, U04, U05, U06 | NF01.1 | Hệ thống cần đăng nhập và phân quyền: chỉ quản lý được tạo, chỉnh sửa. | Nhà hàng |
| NF01.2 | Giao diện rõ ràng, đơn giản, hiển thị đầy đủ thông tin. | Nhà hàng |
| U07 | NF07.1 | Hệ thống cần thiết kế với giao diện đơn giản dễ sử dụng cho khách hàng | Nhà hàng |
| NF07.2 | Hệ thống khoá yêu cầu đặt bàn sau khi diễn ra tiệc | Nhà hàng |
| U08 | NF08.1 | Hệ thống cần thiết kế với giao diện đơn giản dễ sử dụng cho người đặt tiệc. | Nhà hàng |
| NF08.2 | Hệ thống cần xác thực và kiểm tra thông tin khách hàng để đảm bảo chính xác và bảo mật thông tin khách hàng. | Nhà hàng |
| U09 | NF09.1 | Hệ thống cần thiết kế với giao diện đơn giản dễ sử dụng cho người đặt tiệc. | Nhà hàng |
| NF09.2 | Hệ thống cần xác thực và kiểm tra thông tin khách hàng để đảm bảo chính xác và bảo mật thông tin khách hàng. | Nhà hàng |

Bảng 3.8 Yêu cầu từ môi trường vận hành(operation)

### 3.3.3. Yêu cầu từ môi trường phát triển (development)

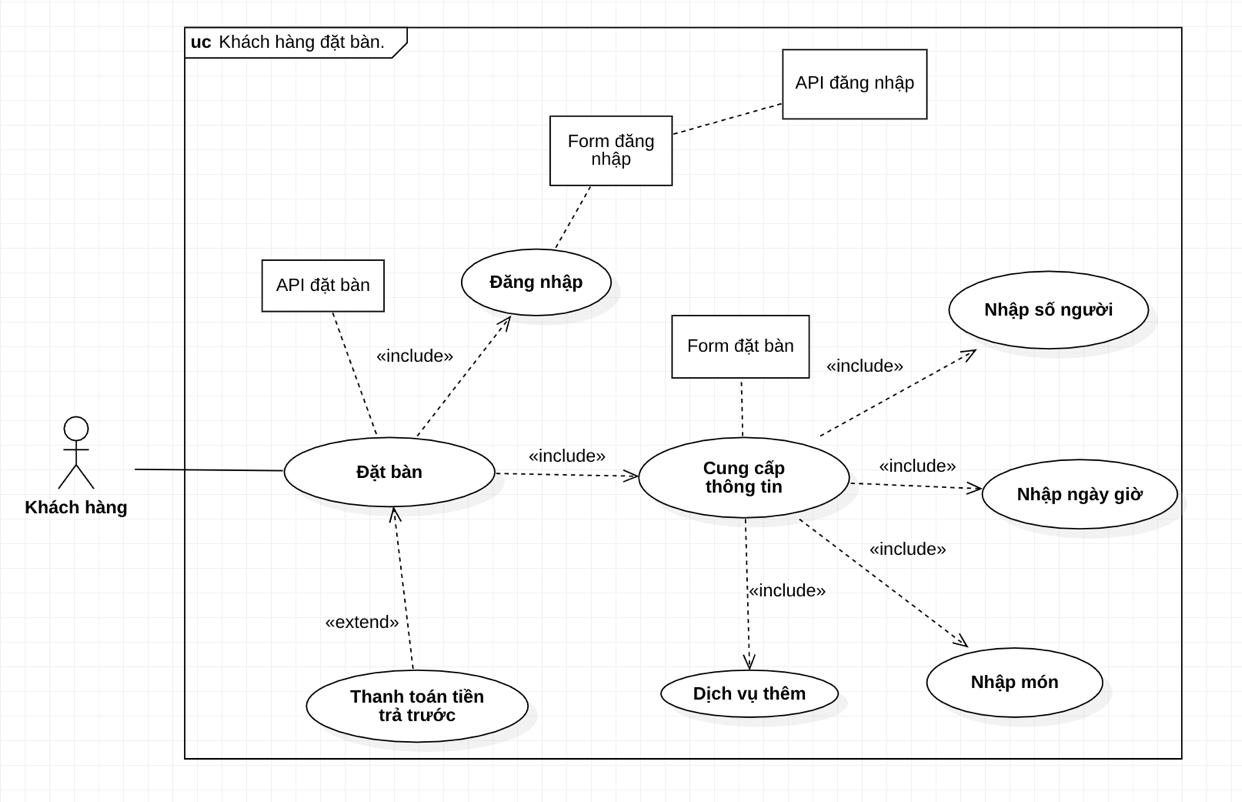
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Đối tượng** | **Nội dung yêu cầu** | **Stack-Holder** |
| TD01 | Software | Viết code trên Visual studio code | Dev team |
| TD02 | Software | Dùng ExpressJS, NodeJs, Flutter | Dev team |

Bảng 3.9 Yêu cầu từ môi trường phát triển(development)

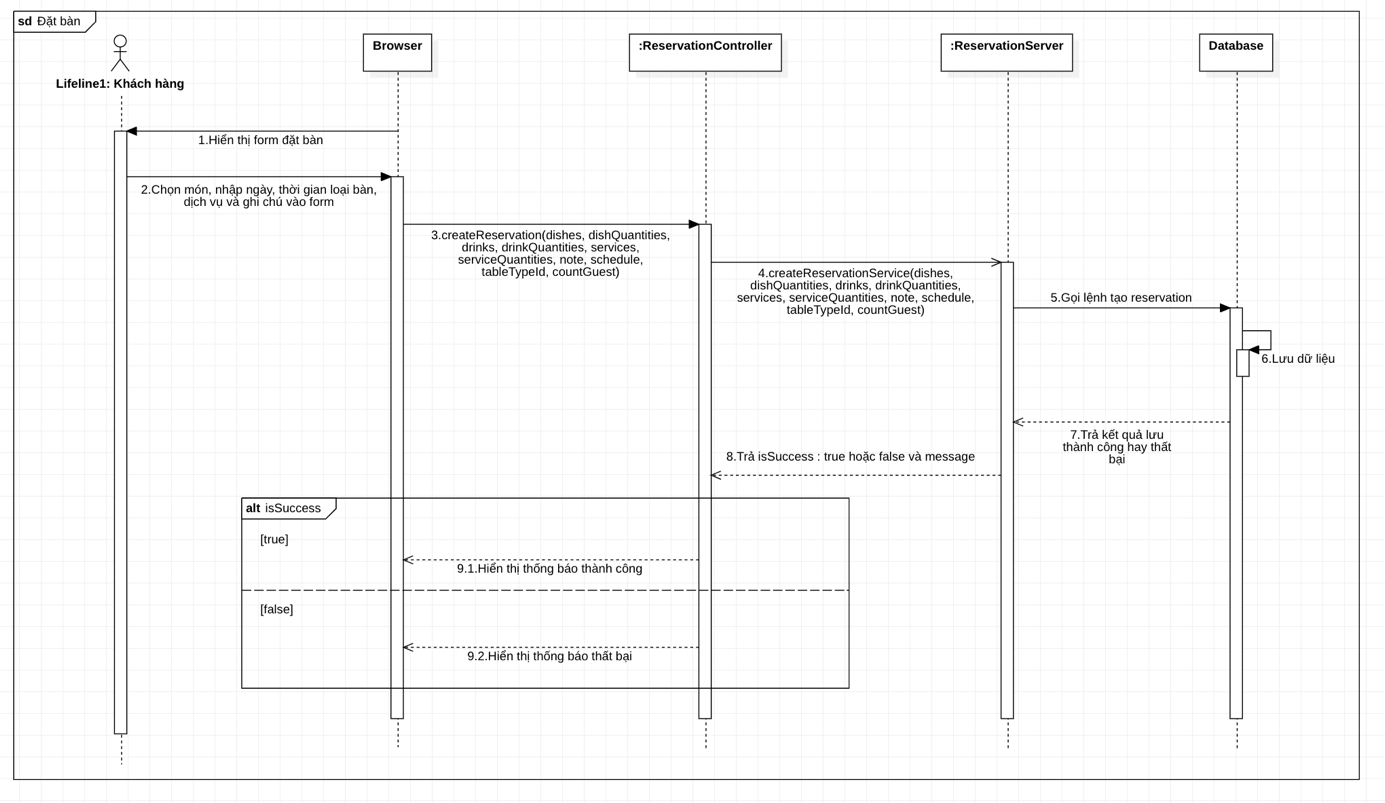
# Thiết kế phần mềm

## 4.1. Lược đồ use-case cho thiết kế phần mềm

### 4.1.1. Use-case khách hàng đặt bàn

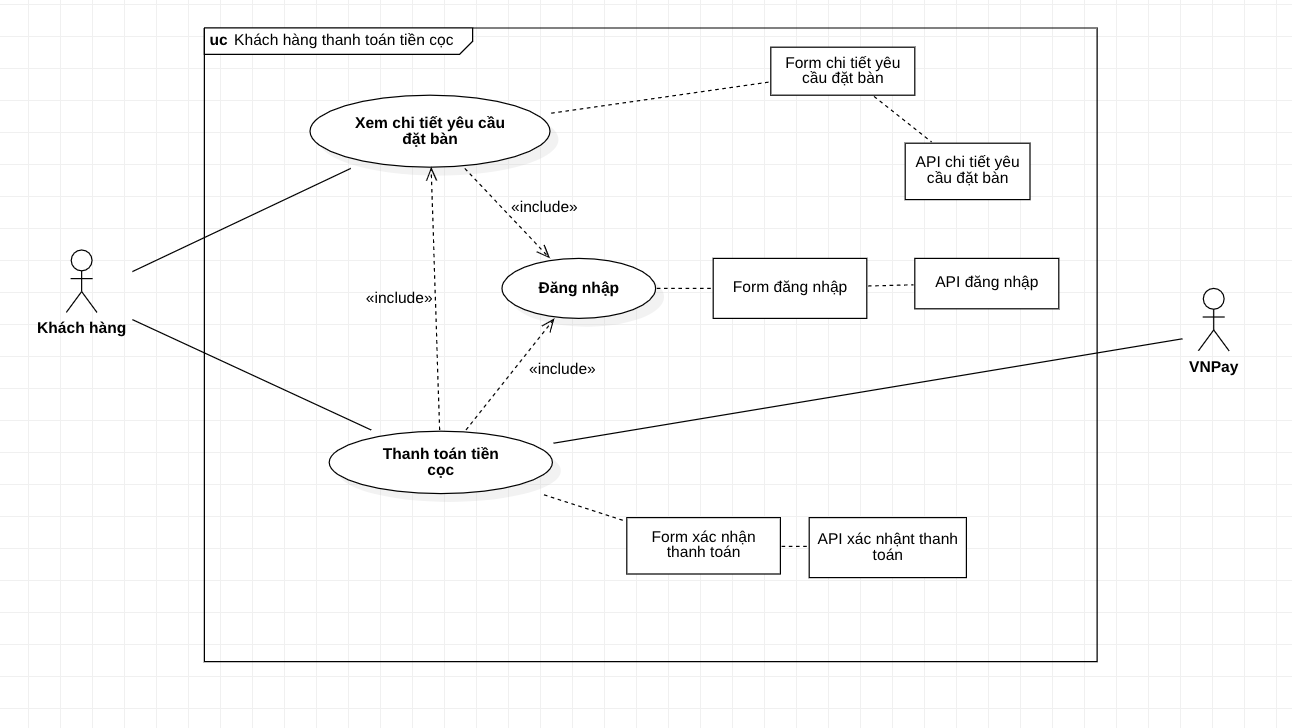


Hình 4.1 Use-case khách hàng đặt bàn

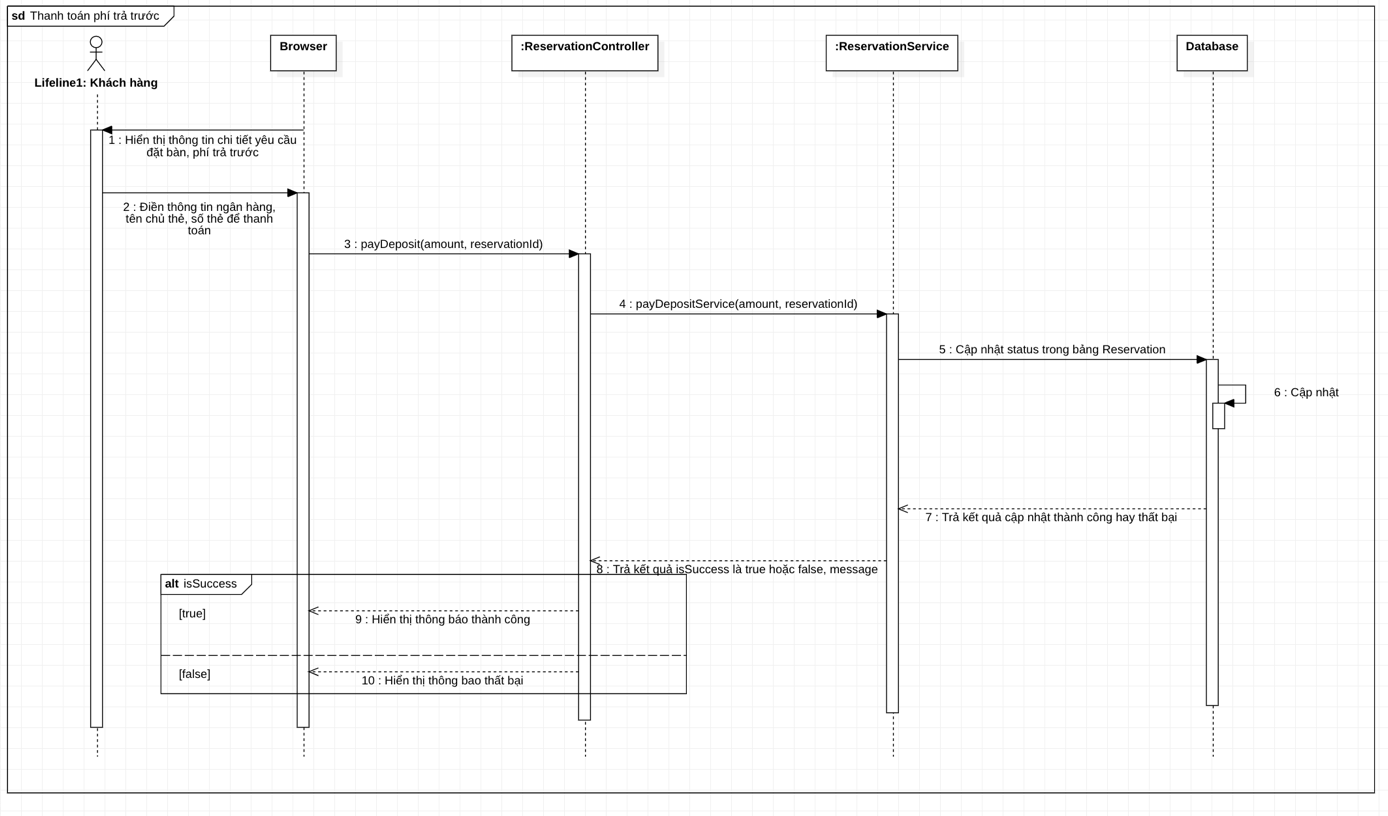


Hình 4.2 Mô tả tương tác Use-case khách hàng đặt bàn

### 4.1.2. Use-case khách hàng thanh toán cọc

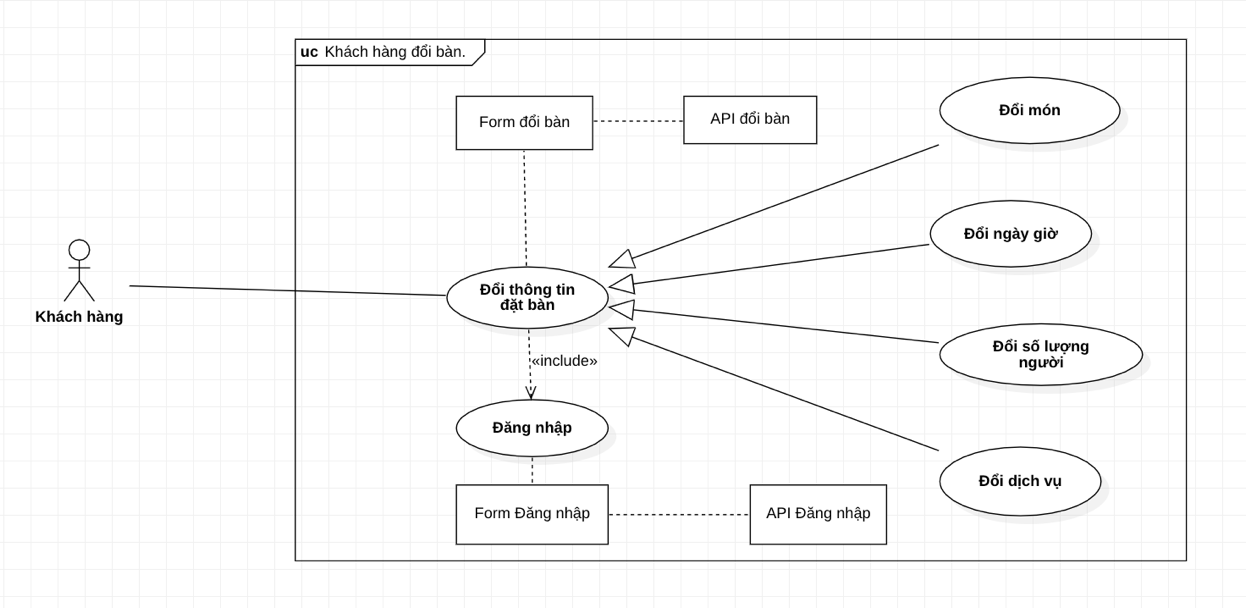


Hình 4.3 Use-case khách hàng thanh toán cọc

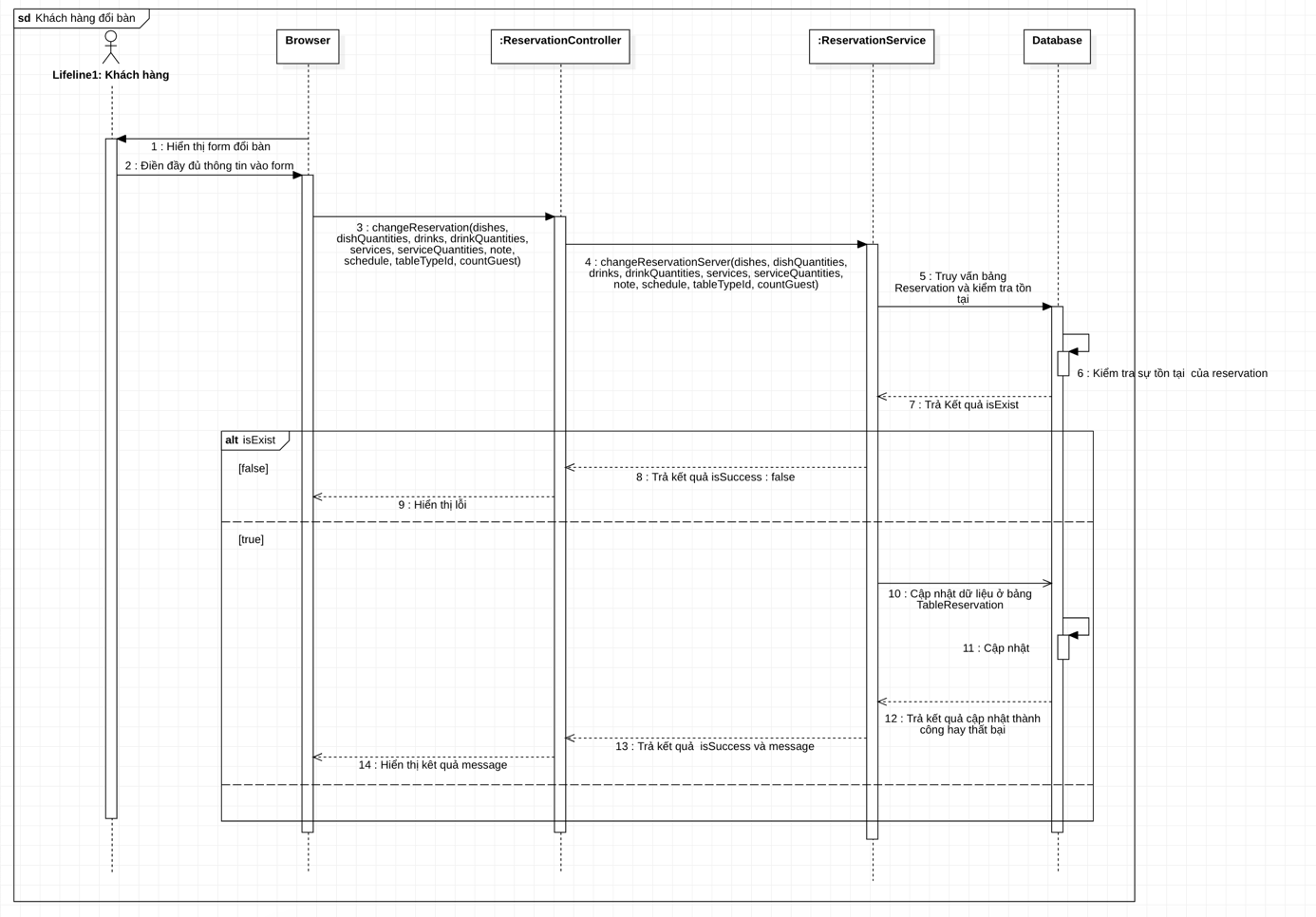


Hình 4.4 Mô tả tương tác Use-case khách hàng thanh toán cọc

### 4.1.3. Use-case khách hàng đổi bàn

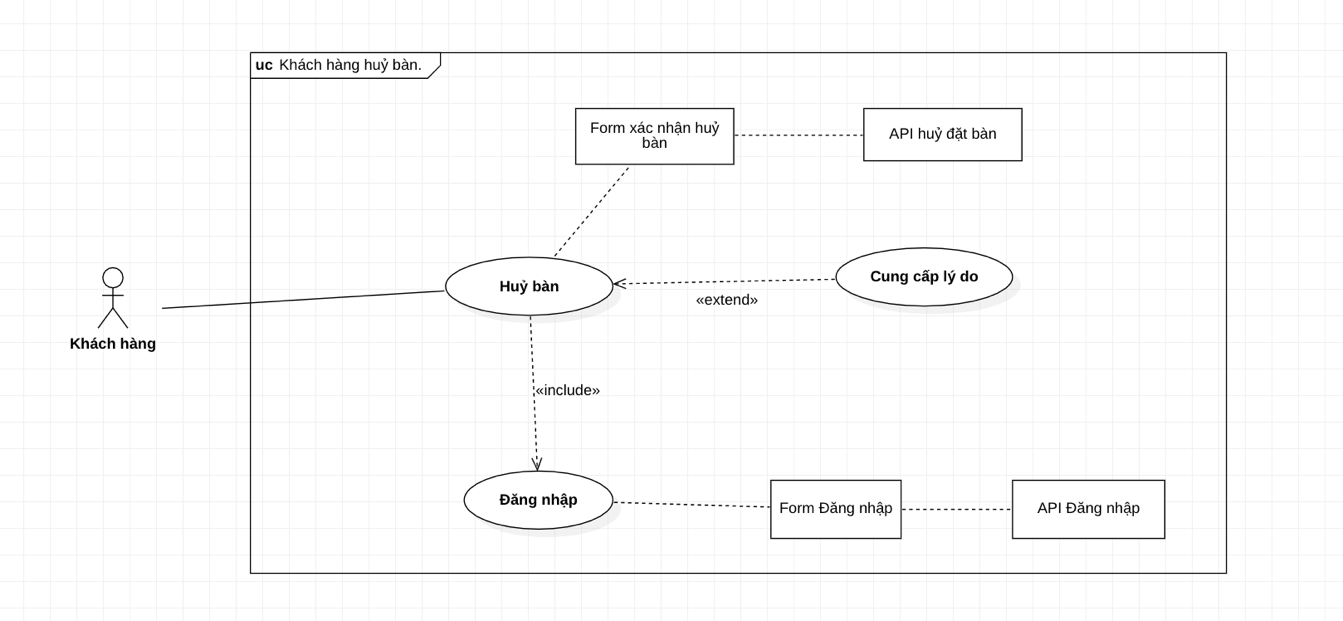


Hình 4.5 Use-case khách hàng đổi bàn

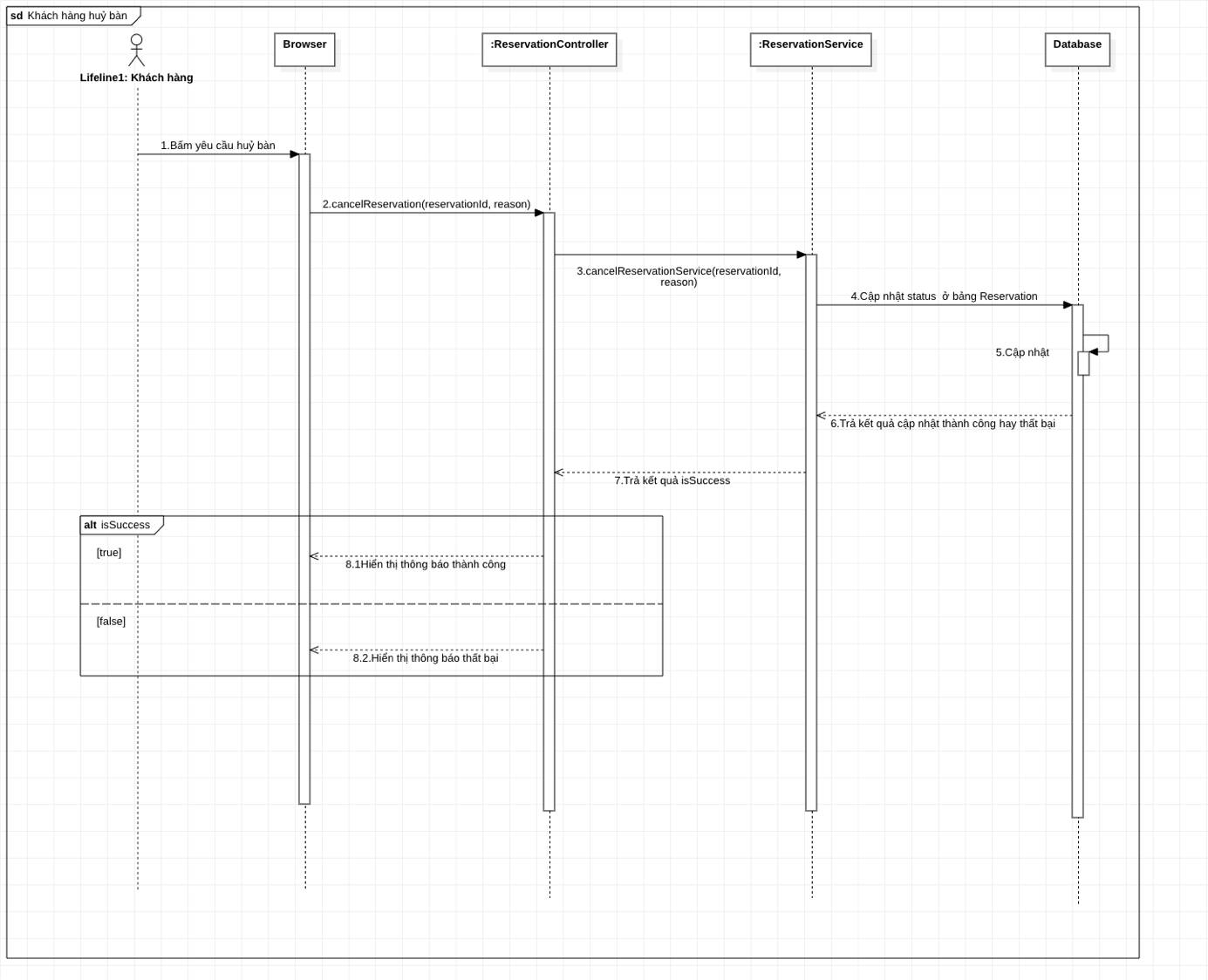


Hình 4.6 Mô tả tương tác Use-case khách hàng đổi bàn

### 4.1.4. Use-case khách hàng huỷ bàn

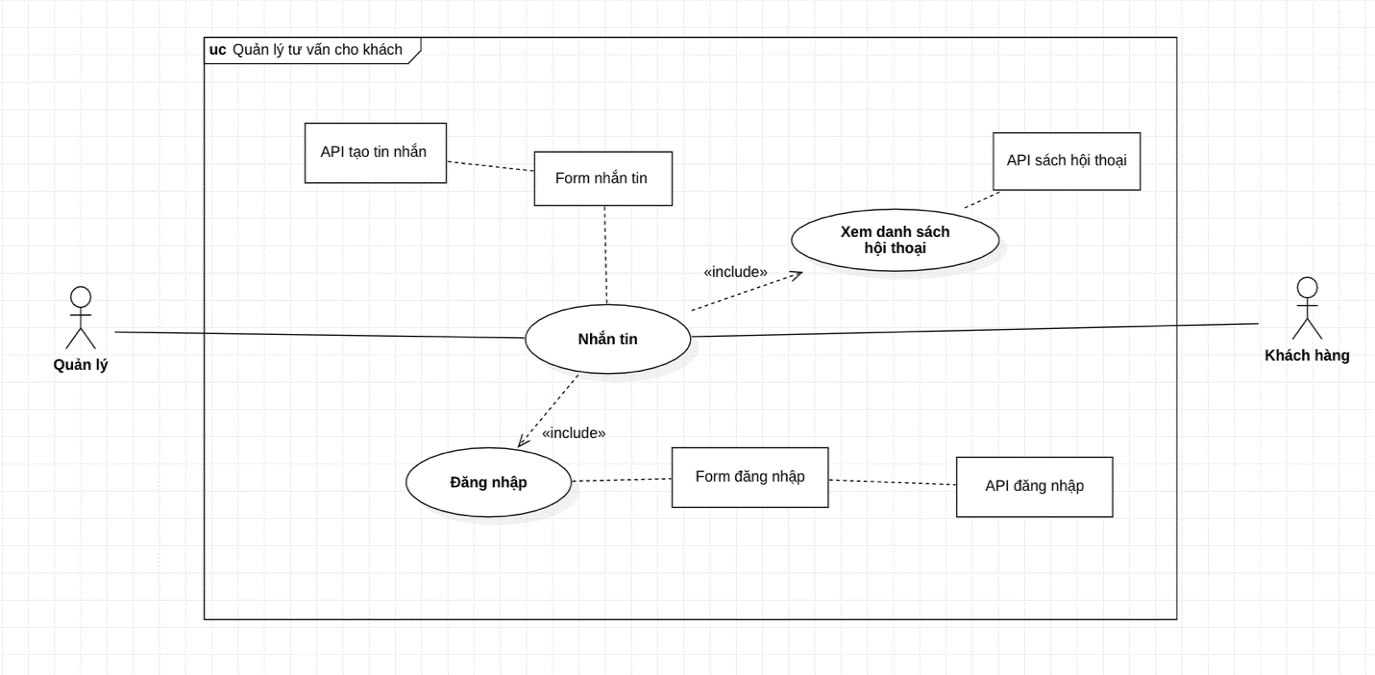


Hình 4.7 Use-case khách hàng huỷ bàn

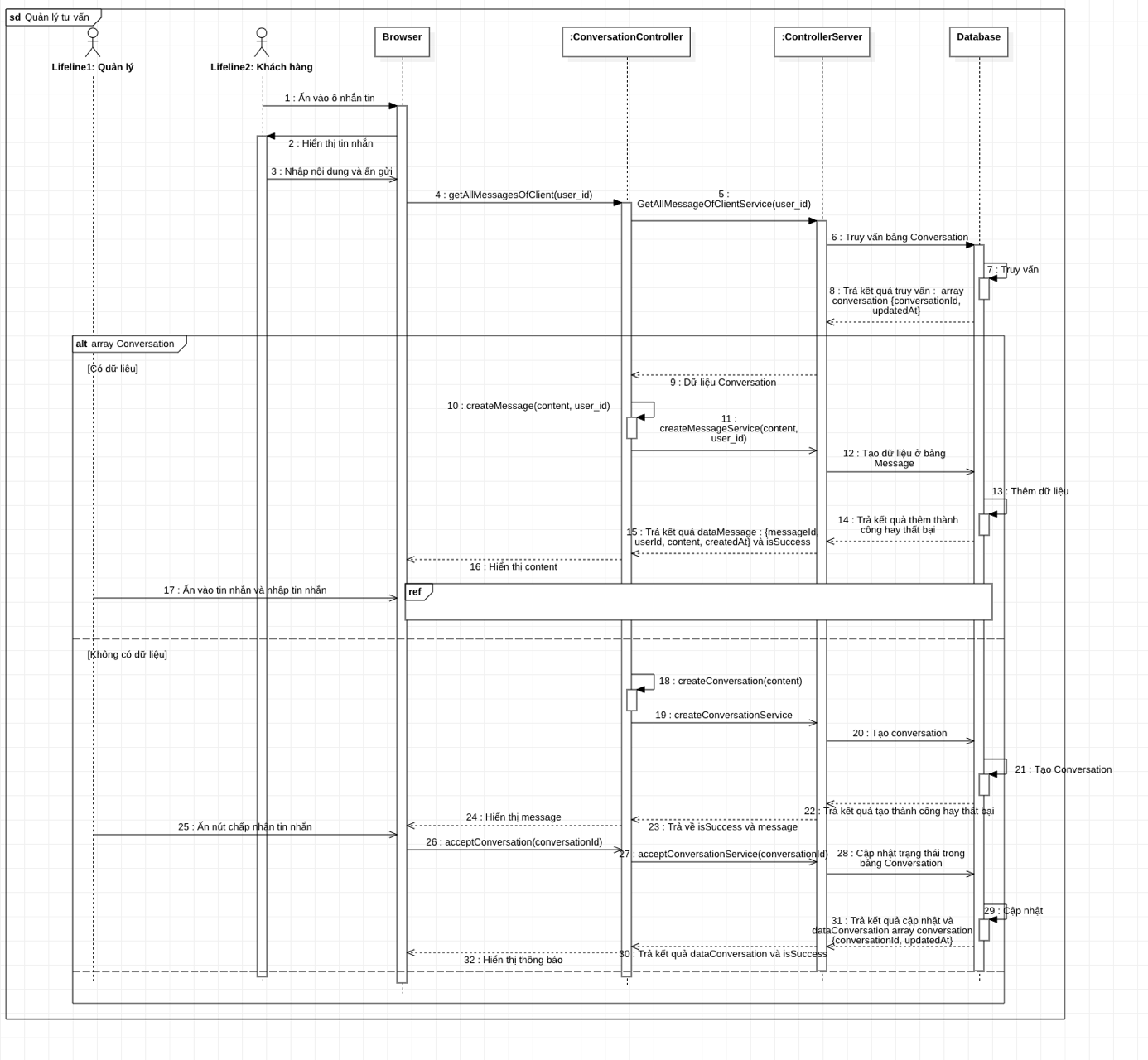


Hình 4.8 Mô tả tương tác Use-case khách hàng huỷ bàn

### 4.1.5. Use-case quản lý tư vấn cho khách hàng

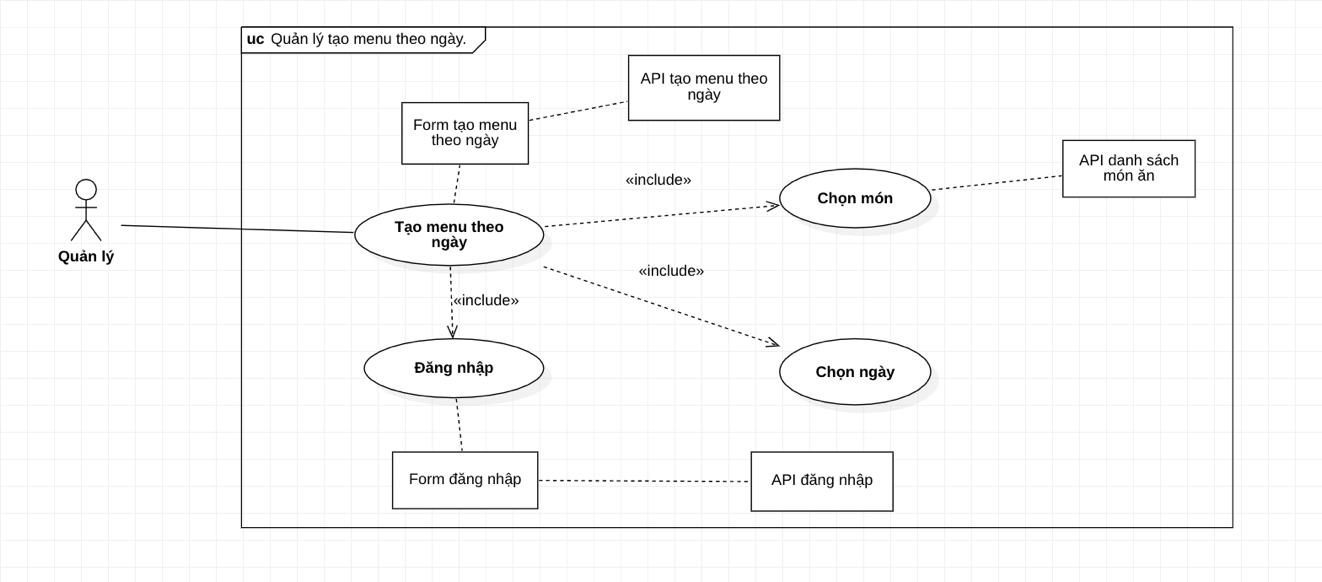


Hình 4.9 Use-case quản lý tư vấn cho khách hàng

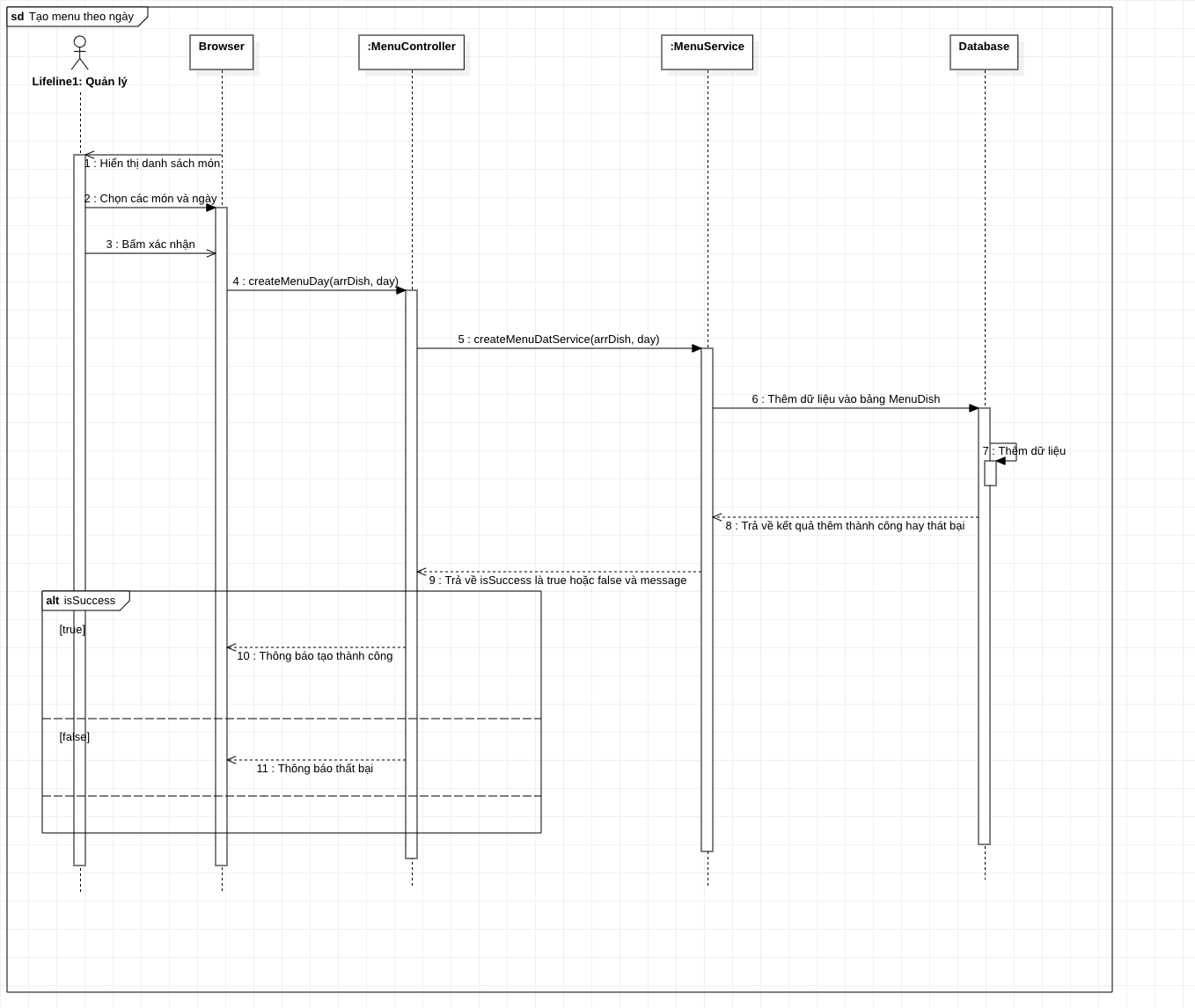


Hình 4.10 Mô tả tương tác Use-case quản lý tư vấn cho khách hàng

### 4.1.6. Use-case quản lý tạo menu theo ngày



Hình 4.11 Use-case quản lý tạo menu theo ngày



Hình 4.12 Mô tả tương tác Use-case quản lý tạo menu theo ngày

## 4.2. Thiết kế phần mềm để xử lý use-case

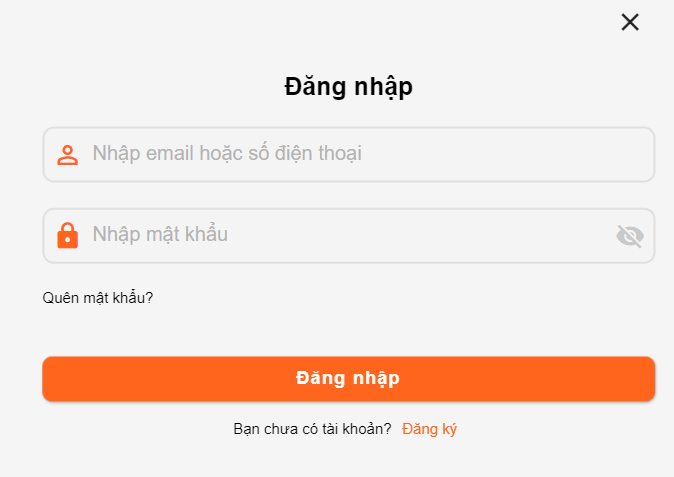
### 4.2.1. Form

#### 4.2.1.1. Form đăng nhập

- Ảnh & tên Form:

+ Tên Form: Login Form.

+ Ảnh Form:



Hình 4.13 Form đăng nhập

- Users: Khách hàng, quản lý nhà hàng.

- Mỗi control chính trên form:

+ Control: Text-Field số điện thoại hoặc email.

* Inputs: nhập số điện thoại hoặc email từ bàn phím.
* Outputs: dữ liệu số điện thoại hoặc email được lấy từ inputs của control.

+ Control: Text-Field mật khẩu.

* Inputs: nhập mật khẩu từ bàn phím.
* Outputs: dữ liệu mật khẩu được lấy từ inputs của control.

+ Control: Button Đăng nhập.

* Inputs: sự kiện click chuột.
* Outputs:

++ Thông báo thành công, thông tin user và chuyển tới form trước đó.

++ Thông báo thất bại.

- Xử lý:

+ Gọi API: Sau khi người dùng nhập số điện thoại hoặc email và mật khẩu và nhấn button Đăng nhập, form có thể gọi [API đăng nhập] để xác thực thông tin đăng nhập của khách hàng.

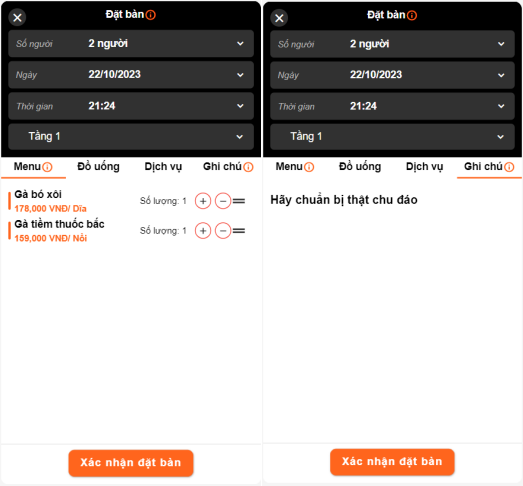
+ Kiểm tra dữ liệu: Form kiểm tra tính hợp lệ của số điện thoại hoặc email và mật khẩu trước khi thực hiện gọi API. Nếu dữ liệu không hợp lệ, thông báo lỗi sẽ được hiển thị trên form.

#### 4.2.1.2. Form đặt bàn

- Ảnh & tên Form:

+ Tên Form: Reservation Form.

+ Ảnh Form:



Hình 4.14. Form đặt bàn

- Users: Khách hàng.

- Mỗi control chính trên form:

+ Control: Text-Field (Nhập số lượng khách).

* Inputs: số lượng khách nhập từ bàn phím.
* Outputs: dữ liệu khách từ control.

+ Control: Date Picker (Chọn ngày).

* Inputs: ngày được chọn từ Date Picker.
* Outputs: dữ liệu ngày (dd/MM/yyyy) từ control.

+ Control: Time Picker (Chọn giờ).

* Inputs: giờ được chọn từ Time Picker.
* Outputs: dữ liệu giờ (hh:mm) từ control.

+ Control: Dropdown-Button (Chọn loại bàn).

* Inputs: loại bàn được chọn từ danh sách lấy từ database.
* Outputs: dữ liệu loại bàn từ control.

+ Control: Button Menu.

* Inputs: sự kiện click chuột.
* Outputs: danh sách món ăn đã được user chọn từ màn hình Món ăn.

+ Control: Button Đồ uống.

* Inputs: sự kiện click chuột.
* Outputs: danh sách đồ uống đã được user chọn từ màn hình Đồ uống.

+ Control: Button Dịch vụ.

* Inputs: sự kiện click chuột.
* Outputs: danh sách dịch vụ đã được user chọn từ màn hình Dịch vụ.

+ Control: Button Ghi chú.

* Inputs: sự kiện click chuột.
* Outputs: control Text-field Ghi chú.

+ Control: Text-Field (Ghi chú).

* Inputs: nội dung ghi chú nhập từ bàn phím.
* Outputs: dữ liệu ghi chú lấy từ inputs của control.

+ Control: Button Xác nhận đặt bàn.

* Inputs: sự kiện click chuột.
* Outputs: kết quả đặt bàn.

- Xử lý:

+ Lấy danh sách loại bàn từ database: Form sẽ gọi [API danh sách loại bàn] để lấy danh sách loại bàn từ database và hiển thị lên Dropdown-Button (Chọn loại bàn) để người dùng có thể chọn.

+ Kiểm tra dữ liệu: Form sẽ kiểm tra tính hợp lệ của các thông tin nhập vào (ví dụ: ngày không được trống, số lượng khách mời phải là số dương, v.v.) và hiển thị thông báo lỗi nếu có.

+ Xử lý đặt bàn:

* Khi người dùng nhấn vào nút "Xác nhận đặt bàn" trên Form, kiểm tra xem người dùng đã đăng nhập chưa.
* Nếu người dùng chưa đăng nhập, hiển thị một Form đăng nhập.
* Nếu người dùng đã đăng nhập, tiến hành gọi [API xác nhận đặt bàn].

+ Xử lý kết quả đặt bàn:

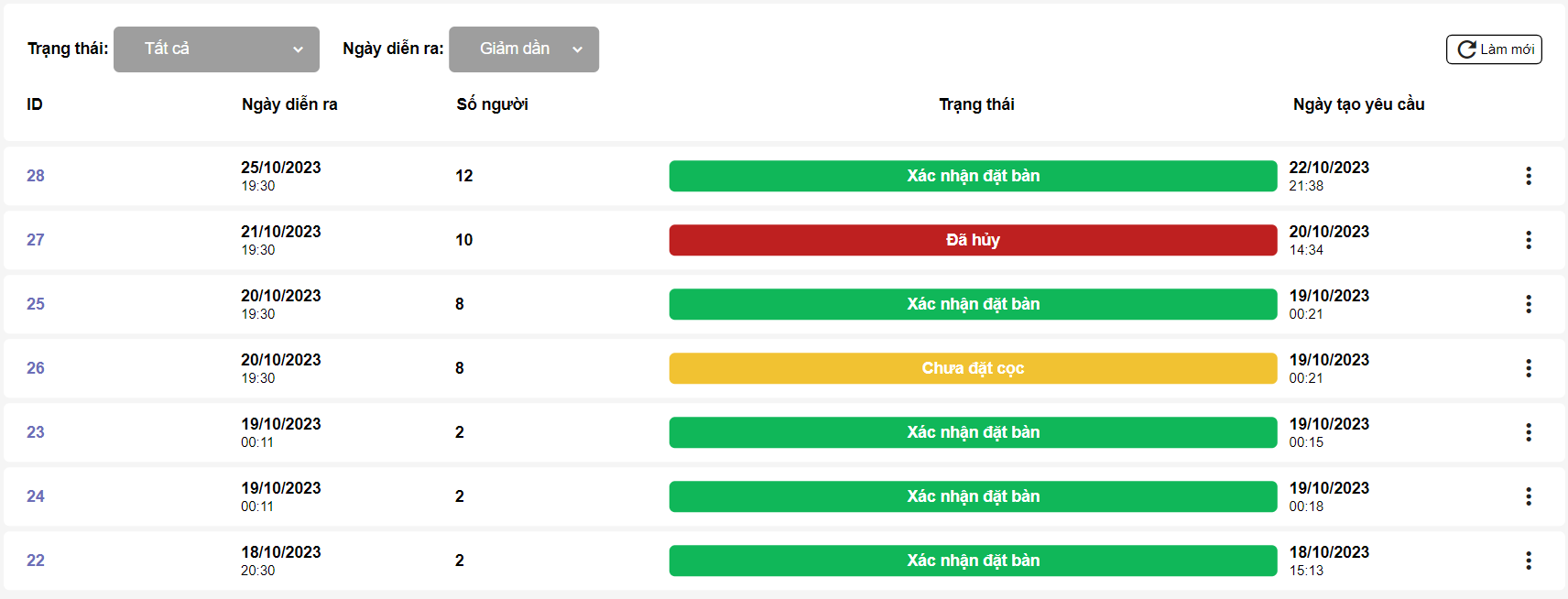
* Sau khi gọi [API xác nhận đặt bàn], nhận kết quả trả về từ API.
* Nếu kết quả đặt bàn thành công, hiển thị thông báo đặt bàn thành công và kiểm tra xem có phí trả trước hay không?.
  + Nếu có phí trả trước, hiển thị Form thanh toán để người dùng nhập thông tin thanh toán và hoàn tất đặt bàn.
  + Nếu không có phí trả trước, kết thúc quá trình đặt bàn.
* Nếu kết quả đặt bàn không thành công, hiển thị thông báo lỗi cho người dùng.

#### 4.2.1.3. Form danh sách yêu cầu đặt bàn

- Ảnh & tên Form:

+ Tên Form: Reservation List Form.

+ Ảnh Form:



Hình 4.15. Form danh sách yêu cầu đặt bàn

- Users: Khách hàng, quản lý.

- Mỗi control chính trên Form:

+ Control: Dropdown-Button (Lọc theo trạng thái).

* Inputs: trạng thái được chọn từ danh sách trạng thái lấy từ database.
* Outputs: dữ liệu trạng thái lấy từ Inputs của Control.

+ Control: Dropdown-Button (Lọc theo ngày diễn ra).

* Inputs: thứ tự được chọn: tăng dần hay giảm dần.
* Outputs: dữ liệu thứ tự từ control.

+ Control: Button (Làm mới).

* Inputs: sự kiện click chuột.
* Outputs: danh sách yêu cầu đặt bàn lấy từ database.

+ Control: List (Danh sách yêu cầu đặt bàn).

* Inputs: danh sách yêu cầu đặt bàn lấy từ database.
* Outputs: dữ liệu lấy từ inputs của control.

- Xử lý:

+ Đầu tiên, gọi [API danh sách yêu cầu đặt bàn] để hiển thị danh sách đặt bàn lên control danh sách yêu cầu đặt bàn.

+ Xử lý khi ấn button làm mới: gọi [API danh sách yêu cầu đặt bàn] để làm mới hiển thị danh sách đặt bàn lên control danh sách yêu cầu đặt bàn.

+ Xử lý khi chọn dropdown-button lọc theo trạng thái và lọc theo ngày diễn ra: gọi [API danh sách yêu cầu đặt bàn] để hiển thị danh sách đặt bàn theo tiêu chí lọc lên control danh sách yêu cầu đặt bàn.

#### 4.2.1.4. Form chi tiết yêu cầu đặt bàn

- Ảnh & tên Form:

+ Tên Form: Reservation Detail Form.

+ Ảnh Form:

- Users: Khách hàng, quản lý.

- Mỗi control chính trên Form:

+ Control: Text (Mã đặt bàn).

* Inputs: id đặt bàn lấy từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Text (Số lượng khách).

* Inputs: số lượng khách lấy từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Text (Thời gian diễn ra).

* Inputs: thời gian diễn ra lấy từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Text (Trạng thái yêu cầu đặt bàn).

* Inputs: trạng thấy lấy từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: List (Bàn).

* Inputs: danh sách bàn lấy từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: List (Món ăn).

* Inputs: danh sách món ăn từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: List (Đồ uống).

* Inputs: danh sách đồ uống từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: List (Dịch vụ).

* Inputs: danh sách dịch vụ từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Button (Đặt cọc).

* Inputs: sự kiện click chuột.
* Outputs: mở [Payment Confirmation Form].

- Xử lý:

+ Đầu tiên, Form gọi [API chi tiết yêu cầu đặt bàn] để hiển thị dữ liệu lên các control.

+ Xử lý khi ấn button Đặt cọc: mở [Payment Confirmation Form]; truyền mã đặt bàn và tiền cọc sang [Payment Confirmation Form].

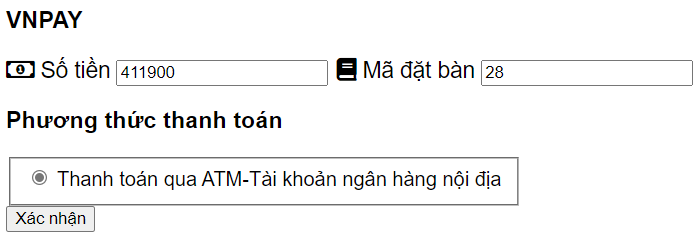
+ Returns outputs: Kết quả xử lý được trả về từ API, bao gồm thông tin chi tiết về yêu cầu đặt bàn hoặc thông báo lỗi nếu không tìm thấy yêu cầu.

#### 4.2.1.5. Form xác nhận thanh toán

- Ảnh & tên Form:

+ Tên Form: Payment Confirmation Form.

+ Ảnh Form:



Hình 4.16. Form xác nhận thanh toán

- Users: Khách hàng.

- Mỗi control chính trên form:

+ Control: Text (Số tiền).

* Inputs: phí trả trước của yêu cầu đặt bàn lấy từ database.
* Outputs: dữ liệu phí trả trước lấy từ inputs của control.

+ Control: Text (Mã đặt bàn).

* Inputs: mã yêu cầu đặt bàn lấy từ database.
* Outputs: dữ liệu mã yêu cầu đặt bàn lấy từ inputs của control.

+ Control: Button (Xác nhận).

* Inputs: sự kiệc click.
* Outputs: chuyển đến trang thanh toán VN-Pay.

- Xử lý:

+ Lấy mã đặt bàn và phí trả trước từ Reservation Detail Form để hiển thị lên text số tiền và text mã đặt bàn.

#### 4.2.1.6. Form đổi bàn

- Ảnh & tên Form:

+ Tên Form: Change Reservation Form.

+ Ảnh Form:

- Users: Khách hàng.

- Mỗi control chính trên form:

+ Control: Text (Mã đặt bàn).

* Inputs: id đặt bàn lấy từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Text-Field (Số lượng khách).

* Inputs: nhập số lượng khách bằng bàn phím.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Date-Picker (Ngày diễn ra).

* Inputs: ngày được chọn từ Date Picker.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Time-Picker (Thời gian diễn ra).

* Inputs: thời gian được chọn từ Date Picker.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Dropdown-Button (Chọn loại bàn).

* Inputs: loại bàn được chọn từ danh sách lấy từ database.
* Outputs: dữ liệu loại bàn từ control.

+ Control: Button (Thêm món).

* Inputs: sự kiện click.
* Outputs: món mới được chọn.

+ Control: List (Món ăn).

* Inputs: danh sách món ăn từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Button (Thêm đồ uống).

* Inputs: sự kiện click.
* Outputs: đồ uống mới được chọn.

+ Control: List (Đồ uống).

* Inputs: danh sách đồ uống từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Button (Thêm dịch vụ).

* Inputs: sự kiện click.
* Outputs: dịch vụ mới được chọn.

+ Control: List (Dịch vụ).

* Inputs: danh sách dịch vụ từ database.
* Outputs: dữ liệu lấy từ inputs của control.

+ Control: Button (Xác nhận).

* Inputs: sự kiện click chuột.
* Outputs: thông báo đổi bàn thành công hoặc thất bại.

- Xử lý:

+ Khi nhấn button Xác nhận: Gọi [API đổi bàn].

* Thành công: Thông báo thành công, cập nhật thông tin yêu cầu đặt bàn.
* Thất bại: Thông báo lí do thất bại.

#### 4.2.1.7. Form xác nhận huỷ bàn

- Ảnh & tên Form:

+ Tên Form: Cancel Reservation Form.

+ Ảnh Form:

- Users: Khách hàng.

- Mỗi control chính trên form:

+ Control: Text-Field (lý do huỷ).

* Inputs: nhập lý do huỷ từ bàn phím.
* Outputs: dữ liệu lý do huỷ được lấy từ inputs của control.

+ Control: Button (Xác nhận).

* Inputs: sự kiệc click.
* Outputs: thông báo huỷ bàn thành công hoặc thất bại.

+ Control: Button (Thoát).

* Inputs: sự kiệc click.
* Outputs: thoát [Cancel Reservation Form].

- Xử lý:

+ Khi nhấn button Xác nhận: Gọi [API huỷ đặt bàn].

* Thành công: Thông báo thành công, cập nhật trạng thái yêu cầu đặt bàn.
* Thất bại: Thông báo lí do thất bại.

#### 4.2.1.8. Form tạo menu theo ngày

- Ảnh & tên Form:

+ Tên Form: Create Menu by Date Form.

+ Ảnh Form:

- Users: Quản lý.

- Mỗi control chính trên form:

+ Control: Date Picker (chọn ngày).

* Inputs: ngày được chọn từ Date Picker.
* Outputs: dữ liệu ngày (dd/MM/yyyy) từ Inputs của control.

+ Control: Button (Thêm món).

* Inputs: sự kiệc click.
* Outputs: Form chọn món ăn.

+ Control: List (Món ăn).

* Inputs: danh sách món ăn từ Button (Thêm món).
* Outputs: thoát [Cancel Reservation Form].

+ Control: Button (Xác nhận).

* Inputs: sự kiện click.
* Outputs: thông báo thêm Menu thành công hay thất bại.

- Xử lý:

+ Xử lý dữ liệu: ngày đã chọn từ Control chọn ngày không được ở trong quá khứ.

+ Khi nhấn button Xác nhận: Gọi [API tạo menu theo ngày].

* Thành công: Thông báo thành công.
* Thất bại: Thông báo lí do thất bại.

#### 4.2.1.9. Form chấp nhận tin nhắn

- Ảnh & tên Form:

+ Tên Form: Accept Message Form.

+ Ảnh Form:

- Users: Quản lý.

- Mỗi control chính trên form:

+ Control: Button (Chấp nhận).

* Inputs: sự kiện click.
* Outputs: thông báo chấp nhận thành công hay thất bại.

- Xử lý:

+ Đầu tiên, đối với khách hàng, Form sẽ gọi [API danh sách tin nhắn].

* + Thành công: Hiển thị danh sách tin nhắn của hội thoại.
  + Thất bại: Thông báo lí do thất bại.

+ Xử lý dữ liệu: Kiểm tra nội dung tin nhắn hợp lệ: không rỗng.

+ Khi nhấn button Gửi: Gọi [API chấp nhận tin nhắn].

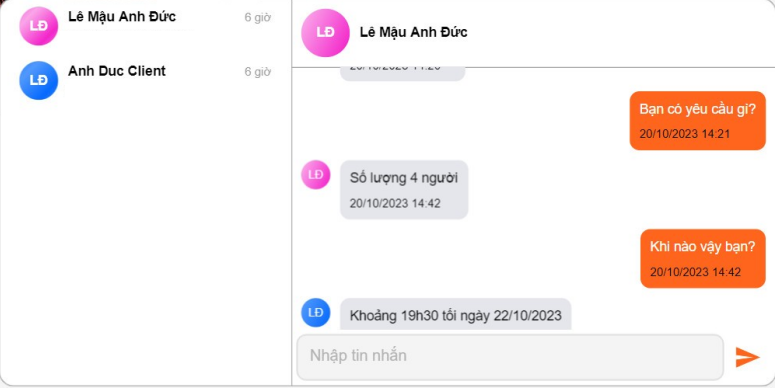
* Thành công: Thông báo chấp nhận thành công.
* Thất bại: Thông báo lí do thất bại.

#### 4.2.1.10. Form nhắn tin

- Ảnh & tên Form:

+ Tên Form: Message Form.

+ Ảnh Form:



Hình 4.17. Form nhắn tin

- Users: Khách hàng, quản lý.

- Mỗi control chính trên form:

+ Riêng quản lý có thêm Control: List (danh sách hội thoại)

* Inputs: danh sách hội thoại lấy từ database.
* Outputs: dữ liệu danh sách hội thoại từ Inputs trong control.

+ Control: List (danh sách tin nhắn).

* Inputs: danh sách tin nhắn lấy từ database.
* Outputs: dữ liệu danh sách tin nhắn từ Inputs trong control.

+ Control: Text-Field (Nhập tin nhắn).

* Inputs: nhập nội dung tin nhắn từ bàn phím.
* Outputs: dữ liệu nội dung tin nhắn từ Inputs của control.

+ Control: Button (Gửi).

* Inputs: sự kiện click.
* Outputs: tin nhắn mới được hiển thị theo real-time.

- Xử lý:

+ Đầu tiên, đối với quản lý, Form sẽ gọi [API danh sách hội thoại].

* Thành công: Hiển thị danh sách hội thoại, quản lý chọn hội thoại để gọi [API danh sách tin nhắn].
  + Thành công: Hiển thị danh sách tin nhắn của hội thoại.
  + Thất bại: Thông báo lí do thất bại.
* Thất bại: Thông báo lí do thất bại.

+ Đầu tiên, đối với khách hàng, Form sẽ gọi [API danh sách tin nhắn].

* + Thành công: Hiển thị danh sách tin nhắn của hội thoại.
  + Thất bại: Thông báo lí do thất bại.

+ Xử lý dữ liệu: Kiểm tra nội dung tin nhắn hợp lệ: không rỗng.

+ Khi nhấn button Gửi: Gọi API tạo tin nhắn.

* Thành công: Hiển thị tin nhắn theo real-time.
* Thất bại: Thông báo lí do thất bại.

### 4.2.2. API

#### 4.2.2.1. API đăng nhập

|  |  |
| --- | --- |
| Tên | API đăng nhập |
| Ý nghĩa | Người dùng đăng nhập vào website. |
| Method | Post |
| Inputs | * Login: Đây là số điện thoại hoặc email được cung cấp để đăng nhập. * Password: Đây là mật khẩu tương ứng với số điện thoại hoặc email để xác thực đăng nhập. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết liệu đăng nhập đã thành công hay không. * msg: Thông điệp trả về. * user: Thông tin người dùng, như tên, địa chỉ email, v.v., có thể được trả về nếu đăng nhập thành công. |
| Xử lý | * Tiền truy cập: API kiểm tra định dạng: * Login: số điện thoại gồm 10 chữ số hoặc email đúng định dạng email. * Mật khẩu: Mật khẩu tối thiểu 6 kí tự, có chứa chữ hoa, chữ thường và kí tự số. * Truy cập CSDL: API thực hiện truy vấn đến bảng Account để kiểm tra xem login và mật khẩu có khớp với dữ liệu đã lưu hay không. Nếu có, truy vấn đến bảng User để lấy dữ liệu trả về outputs. Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định xem đăng nhập thành công hay không. Nếu thông tin đăng nhập hợp lệ, API sẽ trả về trạng thái xác thực (isSuccess) thành công và các thông tin người dùng (user) tương ứng. Nếu không, API sẽ trả về trạng thái xác thực không thành công. Thông điệp (msg) cũng được trả về dựa vào trạng thái xác thực. |

Bảng 4.1. API đăng nhập

#### 4.2.2.2. API danh sách loại bàn

|  |  |
| --- | --- |
| Tên | API danh sách loại bàn |
| Ý nghĩa | Lấy danh sách loại bàn. |
| Method | Get |
| Inputs |  |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết liệu lấy danh sách thành công hay không. * msg: Thông điệp trả về. * tableTypes: Danh sách loại bàn(tableTypeId: id loại bàn, name: tên loại bàn). |
| Xử lý | * Truy cập CSDL: API thực hiện truy vấn đến bảng TableType để lấy danh sách loại bàn. Điều này có thể thực hiện thông qua hàm “Model.findAll” của “Sequelize”. * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định xem lấy danh sách thành công hay không. Nếu lấy thành công, API sẽ trả về trạng thái xác thực (isSuccess) thành công và danh sách loại bàn (tableType). Nếu không, API sẽ trả về trạng thái xác thực không thành công. Thông điệp (msg) cũng được trả về dựa vào trạng thái xác thực. |

Bảng 4.2. API danh sách loại bàn

#### 4.2.2.3. API xác nhận đặt bàn

|  |  |
| --- | --- |
| Tên | API xác nhận đặt bàn |
| Ý nghĩa | Khách hàng xác nhận đặt bàn. |
| Method | Post |
| Inputs | * countGuest: số lượng khách. * schedule: ngày diễn ra. * note: ghi chú của khách hàng. * tableTypeId: mã loại bàn. * dishes: danh sách mã món ăn. * dishQuantities: danh sách số lượng món ăn. * drinks: danh sách mã đồ uống. * drinkQuantities: danh sách số lượng đồ uống. * services: danh sách mã dịch vụ. * serviceQuantities: danh sách số lượng dịch vụ. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết đặt bàn thành công hay không. * msg: Thông điệp trả về. * data: * reservationId: mã đặt bàn. * preFee: phí đặt trước. * deadline: thời hạn thanh toán phí đặt trước. |
| Xử lý | * Truy cập CSDL: Bắt đầu transaction: * API thực hiện truy vấn đến bảng Reservation để tạo mới 1 yêu cầu đặt bàn với thông tin từ Inputs * API thực hiện truy vấn đến bảng Table, TableReservation để lấy danh sách bàn theo loại bàn từ Inputs. Điều này có thể thực hiện thông qua hàm “Model.findAll” của “Sequelize”. Tiếp theo kiểm tra số lượng bàn trống có phù hợp với số lượng khách tham dự và thời gian diễn ra hay không? * Đúng: thực hiện truy vấn đến table TableReservation để tạo mới dữ liệu thể hiện mối quan hệ n-n giữa bàn với yêu cầu đặt bàn. Cộng phí trả trước theo bàn vào phí trả trước tổng. * Sai: Trả về lí do thất bại (msg). * API thực hiện truy vấn đến table Dish, Drink để kiểm tra từng mã món ăn, mã đồ uống có khớp với dữ liệu hay không? Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * Đúng: thực hiện truy vấn đến table MenuReservation để tạo mới dữ liệu thể hiện mối quan hệ n-n giữa món ăn hoặc đồ uống với yêu cầu đặt bàn. Cộng phí trả trước theo món vào phí trả trước tổng. * Sai: Trả về lí do thất bại (msg). * API thực hiện truy vấn đến bảng Service để kiểm tra từng mã dịch vụ có khớp với dữ liệu hay không? Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * Đúng: thực hiện truy vấn đến table ServiceReservation để tạo mới dữ liệu thể hiện mối quan hệ n-n giữa dịch vụ với yêu cầu đặt bàn. Cộng phí trả trước theo dịch vụ vào phí trả trước tổng. * Sai: Trả về lí do thất bại (msg). * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định xem xác nhận đặt bàn thành công hay không. * Nếu thành công: Kết thúc transaction. API sẽ trả về trạng thái xác thực (isSuccess) thành công, mã đặt bàn (reservationId), phí trả trước (preFee), thời hạn thanh toán (deadline), thông điệp thành công (msg). * Nếu không: Rollback transaction. API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.3. API xác nhận đặt bàn.

#### 4.2.2.4. API danh sách yêu cầu đặt bàn

|  |  |
| --- | --- |
| Tên | API danh sách yêu cầu đặt bàn |
| Ý nghĩa | Trả về danh sách đặt bàn theo tiêu chí lọc, phân trang, trạng thái. |
| Method | Get |
| Inputs | * status: trạng thái đặt bàn. * page: phân trang hiện tại. * limit: số lượng mỗi trang. * order: thứ tự ngày diễn ra. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết lấy danh sách thành công hay không. * msg: Thông điệp trả về. * data: * maxPage: tổng số trang. * total: số lượng yêu cầu đặt bàn. * currentPage: trang hiện tại. * row: danh sách yêu cầu đặt bàn. * reservationId: mã yêu cầu đặt bàn. * schedule: thời gian diễn ra. * status: trạng thái đặt bàn. * countGuest: số lượng khách. * createAt: thời gian tạo yêu cầu đặt bàn. |
| Xử lý | * Tiền truy cập CSDL: * API tính toán vị trí đầu và cuối cần lấy danh sách yêu cầu đặt bàn dựa vào Inputs: limit và page, nhờ vào công thức: [bắt đầu, kết thúc] = [limit \* (page - 1), limit \* page]. * Truy cập CSDL: * API thực hiện truy vấn đến bảng Reservation để lấy danh sách yêu cầu đặt bàn dựa vào Inputs. Điều này có thể thực hiện thông qua hàm “Model.findAndCountAll” của “Sequelize”. * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định xem xác nhận lấy danh sách thành công hay không. * Đúng: Tính toán số lượng tất cả (total) yêu cầu đặt bàn, số trang lớn nhất (maxPage), trang hiện tại (currentPage). API sẽ trả về trạng thái xác thực (isSuccess) thành công, dữ liệu (data). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.4. API danh sách yêu cầu đặt bàn

#### 4.2.2.5. API chi tiết yêu cầu đặt bàn

|  |  |
| --- | --- |
| Tên | API chi tiết yêu cầu đặt bàn |
| Ý nghĩa | Trả về chi tiết đặt bàn. |
| Method | Get |
| Inputs | * reservationId: mã yêu cầu đặt bàn. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết lấy danh sách thành công hay không. * msg: Thông điệp trả về. * data: * reservation: * reservationId: mã yêu cầu đặt bàn. * schedule: thời gian diễn ra. * status: trạng thái đặt bàn. * countGuest: số lượng khách. * preFee: phí đặt cọc. * note: ghi chú của khách. * managerNote: ghi chú của quản lý. * createAt: thời gian tạo yêu cầu đặt bàn. * menus: Danh sách món ăn, đồ uống liên quan. * tables: Danh sách bàn liên quan. * services: Danh sách dịch vụ liên quan. |
| Xử lý | * Tiền truy cập CSDL: * API xác thực liệu rằng người dùng có quyền xem chi tiết yêu cầu đặt bàn này không? * Đúng: là khách hàng tạo yêu cầu đặt bàn hoặc là quản lý. * Sai: là khách hàng không tạo yêu cầu đặt bàn. * Truy cập CSDL: * API thực hiện truy vấn đến bảng Table, TableReservation để lấy danh sách bàn theo loại bàn. Điều này có thể thực hiện thông qua hàm “Model.findAll” của “Sequelize”. * API thực hiện truy vấn đến table Dish để kiểm tra từng mã món ăn, mã đồ uống có khớp với dữ liệu hay không? Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * API thực hiện truy vấn đến bảng Service để kiểm tra từng mã dịch vụ có khớp với dữ liệu hay không? Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định xem xác nhận lấy chi tiết yêu cầu đặt bàn thành công hay không. * Đúng: API sẽ trả về trạng thái xác thực (isSuccess) thành công, dữ liệu (data). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.5. API chi tiết yêu cầu đặt bàn

#### 4.2.2.6. API xác nhận thanh toán

|  |  |
| --- | --- |
| Tên | API xác nhận thanh toán. |
| Ý nghĩa | Khách hàng xác nhận thanh toán và điều hướng qua trang thanh toán của VNPAY. |
| Method | Post |
| Inputs | * reservationId: mã yêu cầu đặt bàn. * amount: phí trả trước của yêu cầu đặt bàn. |
| Outputs | Mở trang thanh toán VNPAY. |
| Xử lý | * Truy cập CSDL: * API thực hiện truy vấn đến bảng Reservation để lấy thông tin yêu cầu đặt bàn. Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định xem xác nhận thông tin trùng với Inputs không, và đã thanh toán chưa?. * Đúng: API sẽ mở trang VNPAY để thanh toán. * Sai: API sẽ mở trang VNPAY thông báo thất bại. * Dựa trên kết quả thanh toán, API dựa vào kết quả thanh toán từ VNPAY trả về? * Đúng: API cập nhật yêu cầu đặt bàn trạng thái “Đã thanh toán”. |

Bảng 4.6. API xác nhận thanh toán

#### 4.2.2.7. API huỷ đặt bàn

|  |  |
| --- | --- |
| Tên | API huỷ đặt bàn. |
| Ý nghĩa | Khách hàng huỷ đặt bàn. |
| Method | Post |
| Inputs | * reservationId: mã yêu cầu đặt bàn. * reason: lí do hủy. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết huỷ bàn thành công hay không. * msg: Thông điệp trả về. |
| Xử lý | * Truy cập CSDL: * API thực hiện truy vấn đến bảng Reservation để lấy thông tin yêu cầu đặt bàn. Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. Sau đó, cập nhật dữ liệu: trạng thái đặt bàn thành “hủy” và lí do hủy. * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định huỷ yêu cầu đặt bàn thành công hay không?. * Đúng: API sẽ trả về trạng thái xác thực (isSuccess) thành công, dữ liệu (data). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.7. API huỷ đặt bàn

#### 4.2.2.8. API đổi bàn

|  |  |
| --- | --- |
| Tên | API đổi bàn |
| Ý nghĩa | Khách hàng đổi thông tin đặt bàn. |
| Method | Post |
| Inputs | * reservationId: mã đặt bàn. * countGuest: số lượng khách. * schedule: ngày diễn ra. * note: ghi chú của khách hàng. * tableTypeId: mã loại bàn. * dishes: danh sách mã món ăn. * dishQuantities: danh sách số lượng món ăn. * drinks: danh sách mã đồ uống. * drinkQuantities: danh sách số lượng đồ uống. * services: danh sách mã dịch vụ. * serviceQuantities: danh sách số lượng dịch vụ. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết đặt bàn thành công hay không. * msg: Thông điệp trả về. * data: * reservationId: mã đặt bàn. * preFee: phí đặt trước. * deadline: thời hạn thanh toán phí đặt trước. |
| Xử lý | * Truy cập CSDL: Bắt đầu transaction. * API thực hiện truy vấn đến bảng Reservation để lấy thông tin yêu cầu đặt bàn dựa vào Inputs reservationId. Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. Tiếp theo kiểm tra yêu cầu đặt bàn có tồn tại không? * Đúng: tiếp tục. * Sai: Trả về lí do thất bại (msg). * API thực hiện truy vấn đến bảng Table, TableReservation để lấy danh sách bàn theo loại bàn từ Inputs. Điều này có thể thực hiện thông qua hàm “Model.findAll” của “Sequelize”. Tiếp theo kiểm tra số lượng bàn trống có phù hợp với số lượng khách tham dự và thời gian diễn ra hay không? * Đúng: Cập nhật dữ liệu ở bảng TableReservation. Cộng phí trả trước theo bàn vào phí trả trước tổng. * Sai: Trả về lí do thất bại (msg). * API thực hiện truy vấn đến table Dish để kiểm tra từng mã món ăn, mã đồ uống có khớp với dữ liệu hay không? Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * Đúng: thực hiện truy vấn đến table MenuReservation để cập nhật dữ liệu thể hiện mối quan hệ n-n giữa món ăn hoặc đồ uống với yêu cầu đặt bàn. Cộng phí trả trước theo món vào phí trả trước tổng. * Sai: Trả về lí do thất bại (msg). * API thực hiện truy vấn đến bảng Service để kiểm tra từng mã dịch vụ có khớp với dữ liệu hay không? Điều này có thể thực hiện thông qua hàm “Model.findOne” của “Sequelize”. * Đúng: thực hiện truy vấn đến table ServiceReservation để cập nhật dữ liệu thể hiện mối quan hệ n-n giữa dịch vụ với yêu cầu đặt bàn. Cộng phí trả trước theo dịch vụ vào phí trả trước tổng. * Sai: Trả về lí do thất bại (msg). * Xác thực: Dựa trên kết quả truy cập CSDL, API sẽ xác định xem đổi bàn thành công hay không? * Nếu thành công: kết thúc transaction. API sẽ trả về trạng thái xác thực (isSuccess) thành công, mã đặt bàn (reservationId), phí trả trước (preFee), thời hạn thanh toán (deadline), thông điệp thành công (msg). * Nếu không: rollback transaction. API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.8. API đổi bàn

#### 4.2.2.9. API chấp nhận tin nhắn

|  |  |
| --- | --- |
| Tên | API chấp nhận tin nhắn. |
| Ý nghĩa | Để tránh xung đột việc nhắn tin giữa khách hàng với quản lý. Website chỉ cho phép một quản lý nhắn tin trực tiếp với một khách hàng. Một quản lý cần phải chấp nhận tin nhắn để có thể bắt đầu nhắn tin với khách hàng. |
| Method | Patch |
| Inputs | * conversationId: mã hội thoại |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết tạo menu thành công hay không. * msg: Thông điệp trả về. |
| Xử lý | * Truy cập CSDL: * API thực hiện truy vấn đến bảng Conversation để tìm conversation có mã hội thoại trùng với Inputs và cập nhật trạng thái chấp nhận. Điều này có thể thực hiện thông qua hàm “Model.update” của “Sequelize”. Sau đó kiểm tra, liệu cập nhật thành công? * Đúng: API thực hiện truy vấn đến bảng UserConversation để tạo mới dữ liệu thể hiện việc quản lý đã tham gia vào đoạn hội thoại. * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). * Xác thực: * Dựa trên kết quả truy cập CSDL, API sẽ xác định chấp nhận thành công hay không? * Đúng: API sẽ trả về trạng thái xác thực (isSuccess) thành công, thông điệp thành công (msg). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.9. API chấp nhận tin nhắn

#### 4.2.2.10. API danh sách hội thoại

|  |  |
| --- | --- |
| Tên | API danh sách hội thoại. |
| Ý nghĩa | Hiển thị danh sách hội thoại cho quản lý. |
| Method | Get |
| Inputs |  |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết tạo menu thành công hay không. * msg: Thông điệp trả về. * conversations: danh sách hội thoại. |
| Xử lý | * Truy cập CSDL: * API thực hiện truy vấn đến bảng UserConversation để tìm những conversation có mã quản lý và những conversation chưa được chấp nhận. Điều này có thể thực hiện thông qua hàm “Model.findAll” của “Sequelize”. * Xác thực: * Dựa trên kết quả truy cập CSDL, API sẽ xác định lấy danh sách thành công hay không? * Đúng: API sẽ trả về trạng thái xác thực (isSuccess) thành công, thông điệp thành công (msg), danh sách hội thoại (conversations). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.10. API danh sách hội thoại

#### 4.2.2.11. API danh sách tin nhắn

|  |  |
| --- | --- |
| Tên | API danh sách tin nhắn. |
| Ý nghĩa | Hiển thị danh sách tin nhắn theo hội thoại. |
| Method | Get |
| Inputs | converstationId: mã hội thoại |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết tạo menu thành công hay không. * msg: Thông điệp trả về. * messages: danh sách hội thoại. |
| Xử lý | * Truy cập CSDL: * API thực hiện truy vấn đến bảng Message để tìm những tin nhắn thuộc đoạn hội thoại. Điều này có thể thực hiện thông qua hàm “Model.findAll” của “Sequelize”. * Xác thực: * Dựa trên kết quả truy cập CSDL, API sẽ xác định lấy danh sách thành công hay không? * Đúng: API sẽ trả về trạng thái xác thực (isSuccess) thành công, thông điệp thành công (msg), danh sách tin nhắn (conversations). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.11. API danh sách tin nhắn

#### 4.2.2.12. API nhắn tin

|  |  |
| --- | --- |
| Tên | API nhắn tin. |
| Ý nghĩa | Hiển thị danh sách tin nhắn theo hội thoại. |
| Method | Post |
| Inputs | converstationId: mã hội thoại.  content: nội dung tin nhắn. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết tạo menu thành công hay không. * msg: Thông điệp trả về. |
| Xử lý | * Truy cập CSDL: * API thực hiện truy vấn đến bảng Message để tạo mới tin nhắn thuộc đoạn hội thoại dực vào Inputs. Điều này có thể thực hiện thông qua hàm “Model.create” của “Sequelize”. * Xác thực: * Dựa trên kết quả truy cập CSDL, API sẽ xác định tạo tin nhắn thành công hay không? * Đúng: API sẽ trả về trạng thái xác thực (isSuccess) thành công, thông điệp thành công (msg), danh sách tin nhắn (conversations). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

Bảng 4.12. API nhắn tin

#### 4.2.2.13. API tạo menu theo ngày

|  |  |
| --- | --- |
| Tên | API tạo menu theo ngày. |
| Ý nghĩa | Quản lý tạo menu theo ngày. |
| Method | Post |
| Inputs | * date: ngày. * dishes: danh sách mã món ăn. |
| Outputs | * isSuccess: Trạng thái xác thực, cho biết tạo menu thành công hay không. * msg: Thông điệp trả về. |
| Xử lý | * Tiền xử lý: Nếu date trong quá khứ, API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). * Truy cập CSDL: * API thực hiện truy vấn đến bảng Menu để lấy danh sách menu. Điều này có thể thực hiện thông qua hàm “Model.findAll” của “Sequelize”. Sau đó kiểm tra, liệu ngày đó đã có Menu chưa? * Đúng: tiếp tục. * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). * Xác thực: * Dựa trên kết quả truy cập CSDL, API sẽ xác định huỷ yêu cầu đặt bàn thành công hay không? * Đúng: API sẽ trả về trạng thái xác thực (isSuccess) thành công, thông điệp thành công (msg). * Sai: API sẽ trả về trạng thái xác thực (isSuccess) không thành công, thông điệp thất bại (msg). |

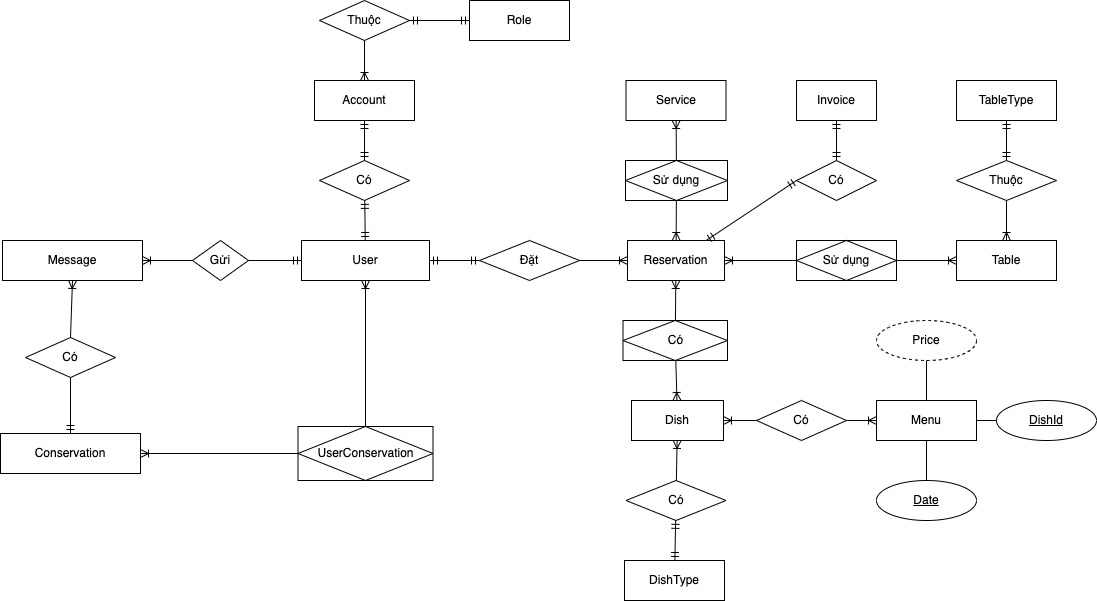
Bảng 4.13. API tạo menu theo ngày

## 4.3. Thiết kế cơ sở dữ liệu cho phần mềm

### 4.3.1. Mô hình thực thể ERD

* + Role(roleId,name)
  + Account(accountId, phone, email, password, otp, verified)
  + User(userId, fullName, address, birthday, gender)
  + Reservation(reservationId, countGuest, schedule, note, status, preFee, createAt, managerNote)
  + Dishes(dishId, name, description, price, image, isDel, isDrink, unit)
  + DishesType(dishTypeId, type, isDrinkType)
  + Menu(date, price)
  + Service(serviceId, name, price, image, unit)
  + Table(tableId, name, isDel)
  + TableType(tableTypeId, name, description, fee)
  + Invoice(invoiceId, price, createAt)
  + Conversation(conversationId, createAt, updateAt)
  + Message(messageId, content, createAt, updateAt)

4.3.2. Mô hình thực thể kết hợp ERD



Hình 4.18. Mô hình thực thể kết hợp ERD

### 4.3.3. Thiết kế chi tiết thực thể:

* + Role(roleId,name)
  + Account(accountId, phone, email, password, otp, verified, roleId)
  + User(userId, fullName, address, birthday, gender, accountId)
  + Reservation(reservationId, countGuest, schedule, note, status, preFee, refundFee, createAt, managerNote, userId)
  + Service(serviceId, name, price, image, unit)
  + Service\_Reservation(serviceReservationId, price, quantity, reservationId, serviceId)
  + Dishes(dishId, name, description, price, image, isDel, isDrink, unit, dishesTypeId)
  + DishesType(dishTypeId, type, isDrinkType)
  + Menu(date, price, dishId)
  + Menu\_Reservation(menuReservationId, order, price, quantity, dishId, reservationId)
  + Table(tableId, name, isDel, tableTypeId)
  + TableType(tableTypeId, name, description, fee)
  + Invoice(invoiceId, price, createAt, reservationId)
  + Conversation(conversationId, createAt, updateAt)
  + UserConservation(userConservationId, createdAt, updateAt, conversationId, userId)
  + Message(messageId, content, createAt, updateAt, userId, conversationId)
  + Table\_Reservation(tableReservationId, reservationId, tableId)

**4.3.4. Chi tiết bảng :**

* **ROLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| roleId | INT | Khoá chính | NOT NULL |
| name | NVARCHAR(100) |  | NOT NULL |

* **ACCOUNTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| accountId | INT | Khoá chính | NOT NULL |
| roleId | INT | Khoá ngoại | NOT NULL |
| phone | VARCHAR(255) | Unique | NOT NULL |
| email | VARCHAR(255) | Unique | NOT NULL |
| password | VARCHAR(255) |  | NOT NULL |
| otp | INT |  | NOT NULL, Default 0 |
| verified | BOOLEAN |  | NOT NULL,  Default 0 |

* **USERS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| userId | INT | Khoá chính | NOT NULL |
| accoundId | INT | Khoá ngoại | NOT NULL |
| userName | NVARCHAR(255) | Unique | NOT NULL |
| address | NVARCHAR(255) |  |  |
| birthday | DATE |  | NOT NULL |
| gender | BOOLEAN |  | NOT NULL,  Defaul 0 |

* **TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| tableId | INT | Khoá chính | NOT NULL |
| name | NVARCHAR(255) | unique | NOT NULL |
| tableTypeId | INT | Khoá ngoại | NOT NULL |
| isDel | TINY(2) |  | Default 0 |

* **TABLETYPES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| tableTypeId | INT | Khoá chính | NOT NULL |
| name | NVARCHAR(255) | unique | NOT NULL |
| description | NVARCHAR(255) |  |  |
| fee | INT |  | NOTNULL |

* **DISHTYPES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| dishTypeId | INT | Khoá chính | NOT NULL |
| type | NVARCHAR(255) | Unique | NOT NULL |
| isDrinkType | BOOLEAN |  | NOT NULL,  Default ‘0’ |

* **DISHES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| dishId | INT | Khoá chính | NOT NULL |
| dishTypeId | INT | Khoá ngoại | NOT NULL |
| name | NVARCHAR(255) | unique | NOT NULL |
| description | NVARCHAR(255) |  |  |
| price | INT |  | NOT NULL |
| image | VARCHAR(255) |  |  |
| isDel | BOOLEAN |  | NOT NULL,  Default 0 |
| isDrink | BOOLEAN |  | NOT NULL,  Default 0 |
| unit | NVARCHAR(255) |  | NOT NULL |

* **MENUS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| date | DATETIME | Khoá chính | NOT NULL |
| dishId | INT | Khoá chính, khoá ngoại | NOT NULL |
| price | DOUBLE |  | NOT NULL |

* **INVOICES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| invoiceId | INT | Khoá chính | NOT NULL |
| reservationId | INT | Khoá ngoại | NOT NULL |
| price | INT |  | NOT NULL |
| createAt | DATETIME |  | NOT NULL |

* **RESERVATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| reservationId | INT | Khoá chính | NOT NULL |
| userId | INT | Khoá ngoại | NOT NULL |
| schedule | DATETIME |  | NOT NULL |
| countGuest | INT |  | NOT NULL |
| note | NVARCHAR(255) |  |  |
| status | INT |  | NOT NULL,  Default ‘-1’ |
| preFee | DOUBLE |  | NOT NULL,  Default 0 |
| createAt | DATETIME |  | NOT NULL |
| managerNote | NVARCHAR(255) |  |  |
| refundFee | DOUBLE |  | NOT NULL, Default 0 |

* **TABLE\_RESERVATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| reservationId | INT | Khoá chính, khoá ngoại | NOT NULL |
| tableId | INT | Khoá chính, khoá ngoại | NOT NULL |

Bảng 4.44. Bảng TABLE\_RESERVATIONS

* **MENU\_RESERVATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| dishId | INT | Khoá chính, khoá ngoại | NOT NULL |
| reservationId | INT | Khoá chính, khoá ngoại | NOT NULL |
| order | INT |  | NOT NULL |
| price | DOUBLE |  | NOT NULL |

* **SERVICES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| serviceId | INT | Khoá chính | NOT NULL |
| name | NVARCHAR(255) |  | NOT NULL |
| price | INT |  | NOT NULL |
| unit | NVARCHAR(255) |  | NOT NULL |

* **SERVICE\_RESERVATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| reservationId | INT | Khoá chính, khoá ngoại | NOT NULL |
| serviceId | INT | Khoá chính, khoá ngoại | NOT NULL |

* **MESSAGES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| messageId | INT | Khoá chính | NOT NULL |
| conversationID | INT | Khoá ngoại | NOT NULL |
| userId | INT | Khoá ngoại | NOT NULL |
| content | NVARCHAR(255) |  | NOT NULL |
| createdAt | DATETIME |  | NOT NULL |
| updatedAt | DATETIME |  | NOT NULL |

* **CONVERSATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| conversationId | INT | Khoá chính | NOT NULL |
| Accept\_manager | TINYINT(1) | Khoá ngoại | NOT NULL |
| createdAt | DATETIME |  | NOT NULL |
| updatedAt | DATETIME |  | NOT NULL |

* **USERCONVERSATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu** | **Khoá** | **Ghi chú** |
| userConversationId | INT | Khoá chính | NOT NULL |
| conversationID | INT | Khoá ngoại | NOT NULL |
| userId | INT | Khoá ngoại | NOT NULL |
| createdAt | DATETIME |  | NOT NULL |
| updatedAt | DATETIME |  | NOT NULL |

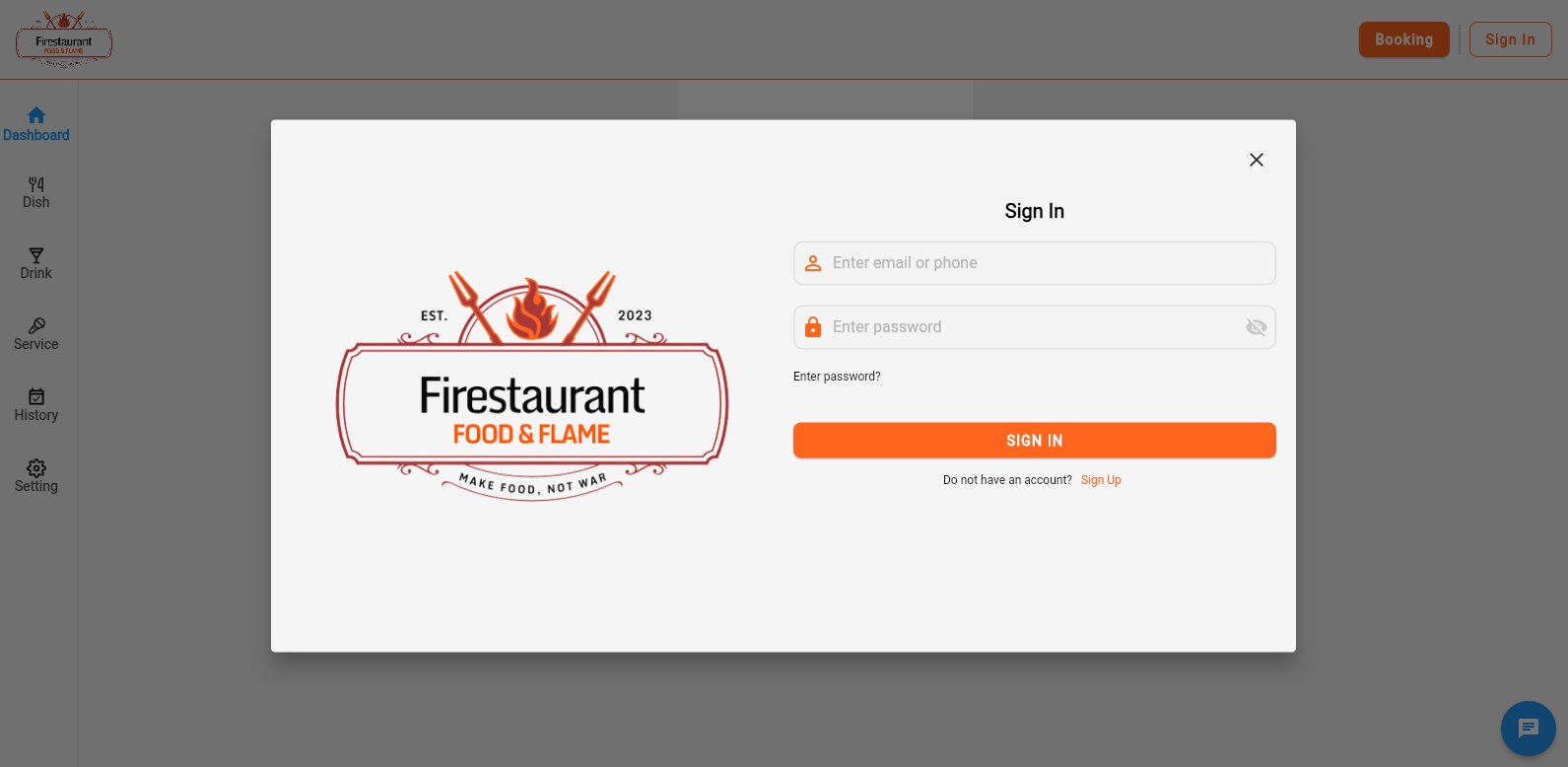
## 4.4. Bảng tham chiếu

|  |  |  |  |
| --- | --- | --- | --- |
| Usecase | Form | API | Table |
| UC-01/ Khách hàng đặt bàn. | - Form đăng nhập.  - Form đặt bàn. | - API đặt bàn.  - API đăng nhập. | User, Account, Role Reservation, Table, TableType, Dish, Service, TableReservation, MenuReservation, ServiceReservation. |
| UC-02/ Khách hàng xác nhận thanh toán. | - Form đăng nhập.  - Form danh sách yêu cầu đặt bàn.  - Form chi tiết yêu cầu đặt bàn.  - Form xác nhận thanh toán. | - API đăng nhập.  - API danh sách yêu cầu đặt bàn.  - API chi tiết yêu cầu đặt bàn.  - API xác nhận thanh toán. | User, Account, Role, Reservation. |
| UC-03/ Khách hàng đổi bàn. | - Form đăng nhập.  - Form danh sách yêu cầu đặt bàn.  - Form chi tiết yêu cầu đặt bàn.  - Form đổi bàn. | - API đăng nhập.  - API danh sách yêu cầu đặt bàn.  - API chi tiết yêu cầu đặt bàn.  - API đổi bàn. | User, Account, Role Reservation, Table, TableType, Dish, Service,TableReservation, MenuReservation, ServiceReservation. |
| UC-04/ Khách hàng huỷ bàn. | - Form đăng nhập.  - Form danh sách yêu cầu đặt bàn.  - Form chi tiết yêu cầu đặt bàn.  - Form huỷ bàn. | - API đăng nhập.  - API danh sách yêu cầu đặt bàn.  - API chi tiết yêu cầu đặt bàn.  - API huỷ bàn. | User, Account, Role Reservation. |
| UC-05/ Quản lý tư vấn cho khách hàng. | - Form đăng nhập.  - Form danh sách hội thoại.  - Form xác nhận tin nhắn.  - Form danh sách tin nhắn.  - Form nhắn tin. | - API đăng nhập.  - API danh sách hội thoại.  - API xác nhận tin nhắn.  - API danh sách tin nhắn.  - API nhắn tin. | User, Account, Role Reservation. |
| UC-06/ Quản lý tạo menu theo ngày. | - Form đăng nhập.  - Form tạo menu theo ngày. | - API đăng nhập.  - API tạo menu theo ngày. | User, Account, Dish, Menu. |

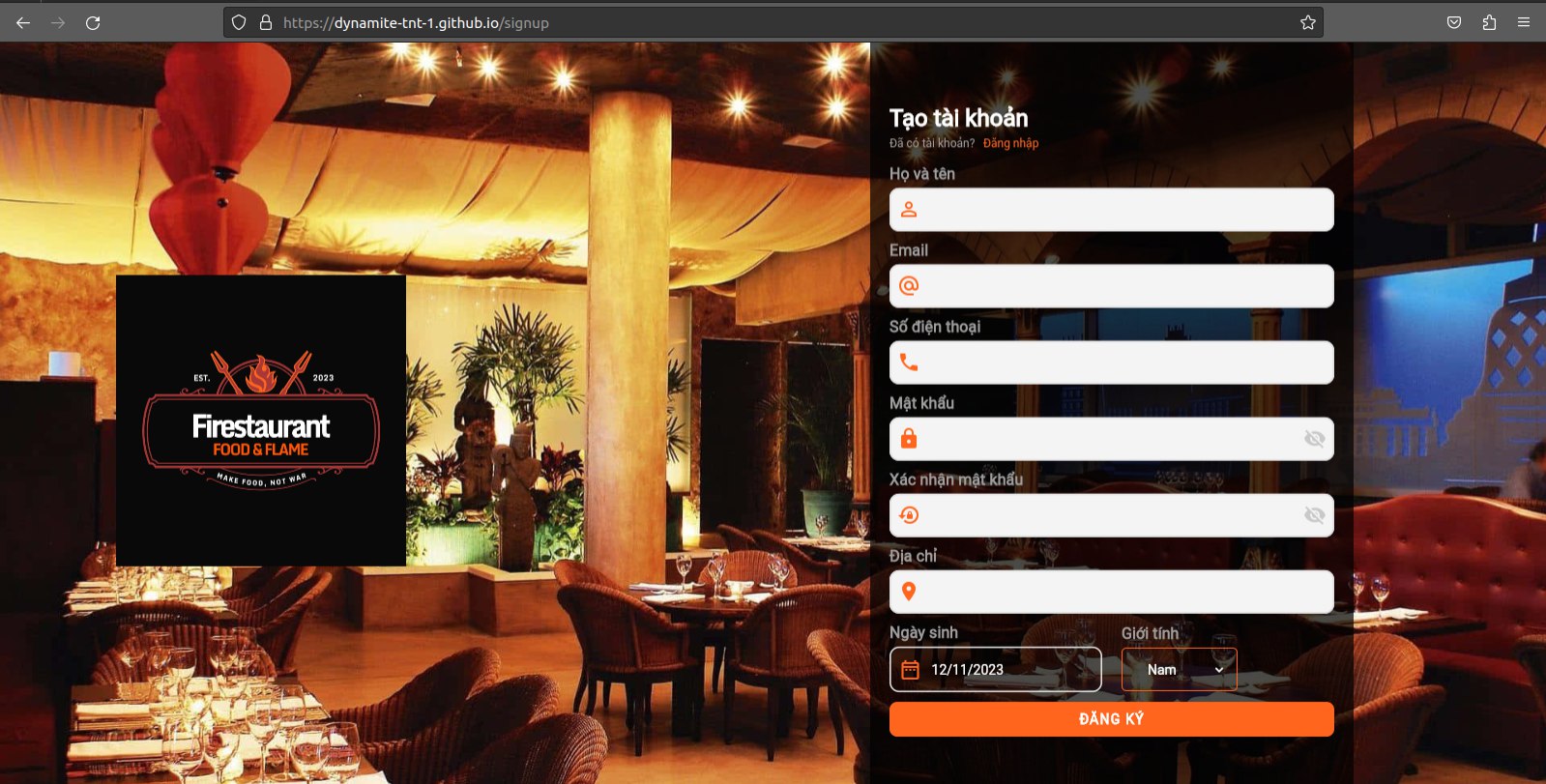
Bảng 4.14. Bảng tham chiếu

# Demo chương trình

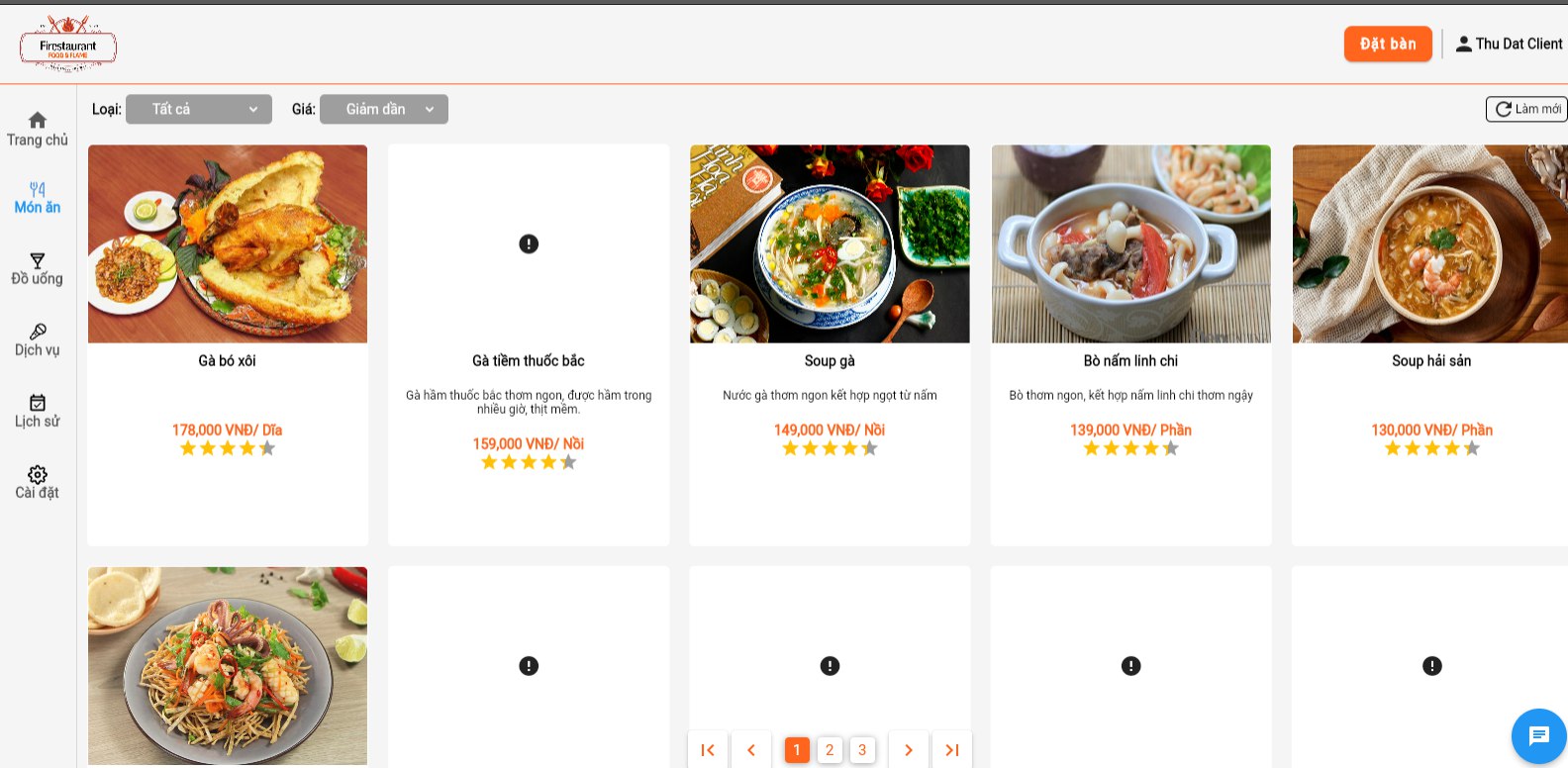
## 5.1 . Đăng ký :



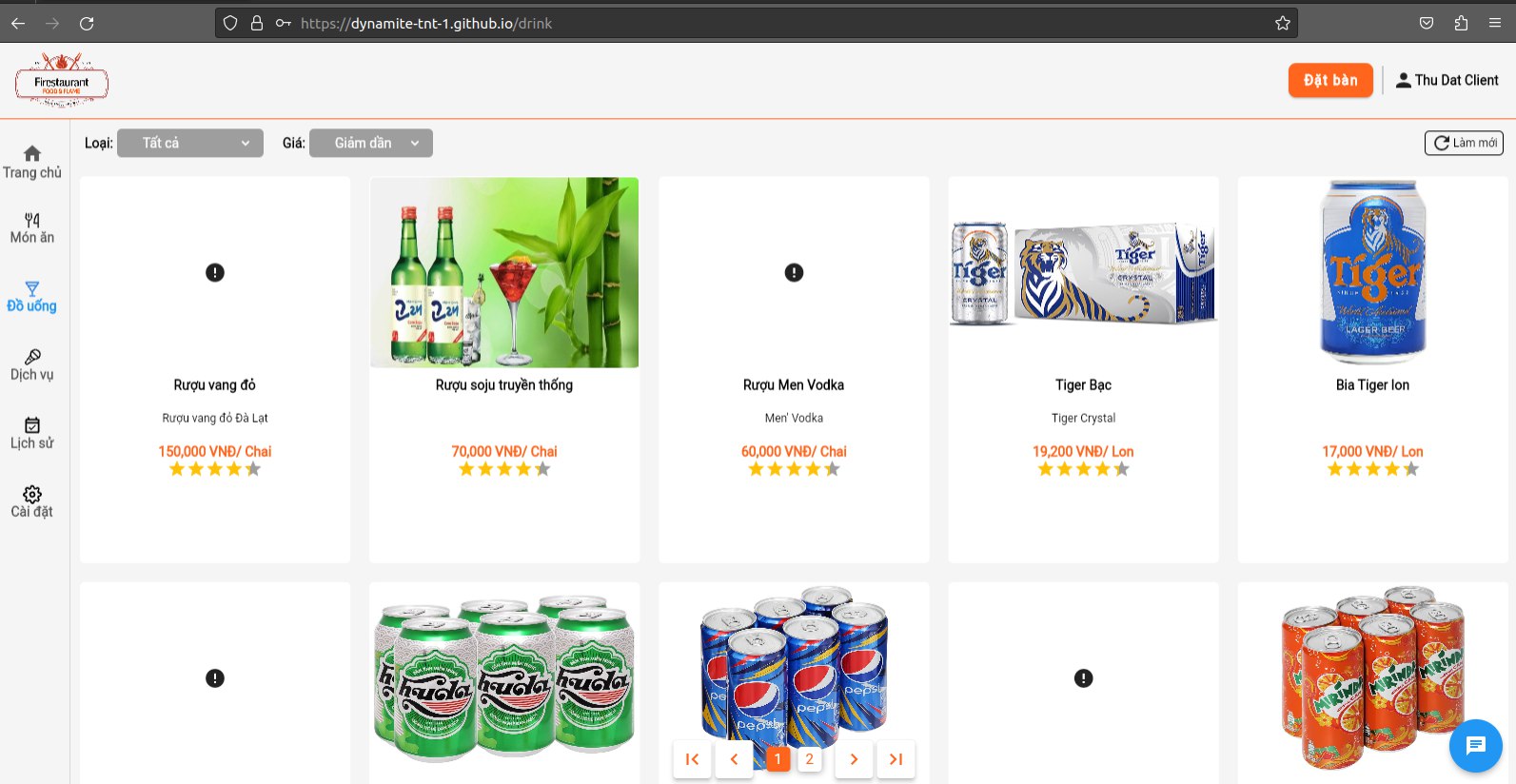
## 5.2. Đăng nhập :



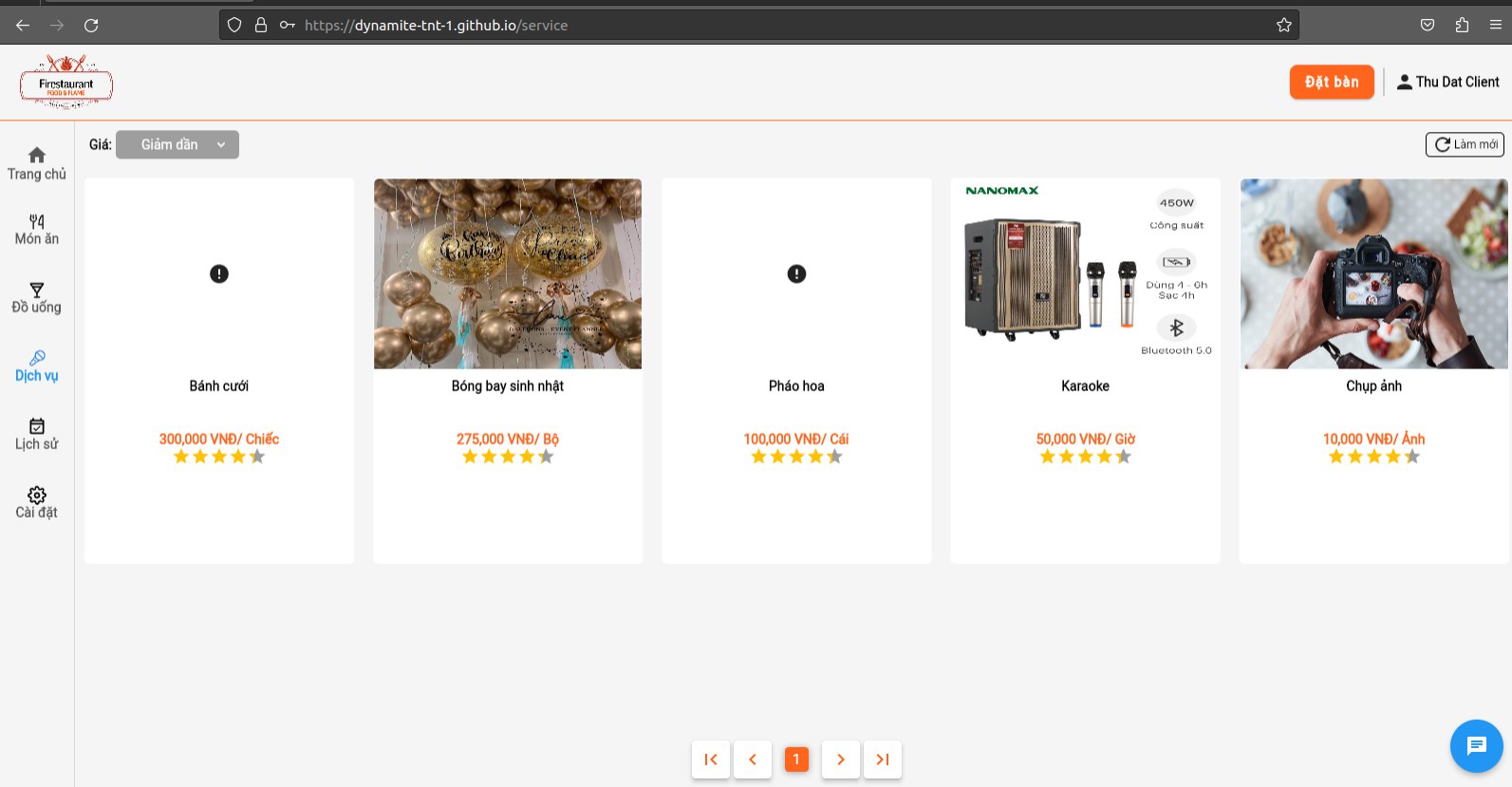
## 5.3. Danh sách món



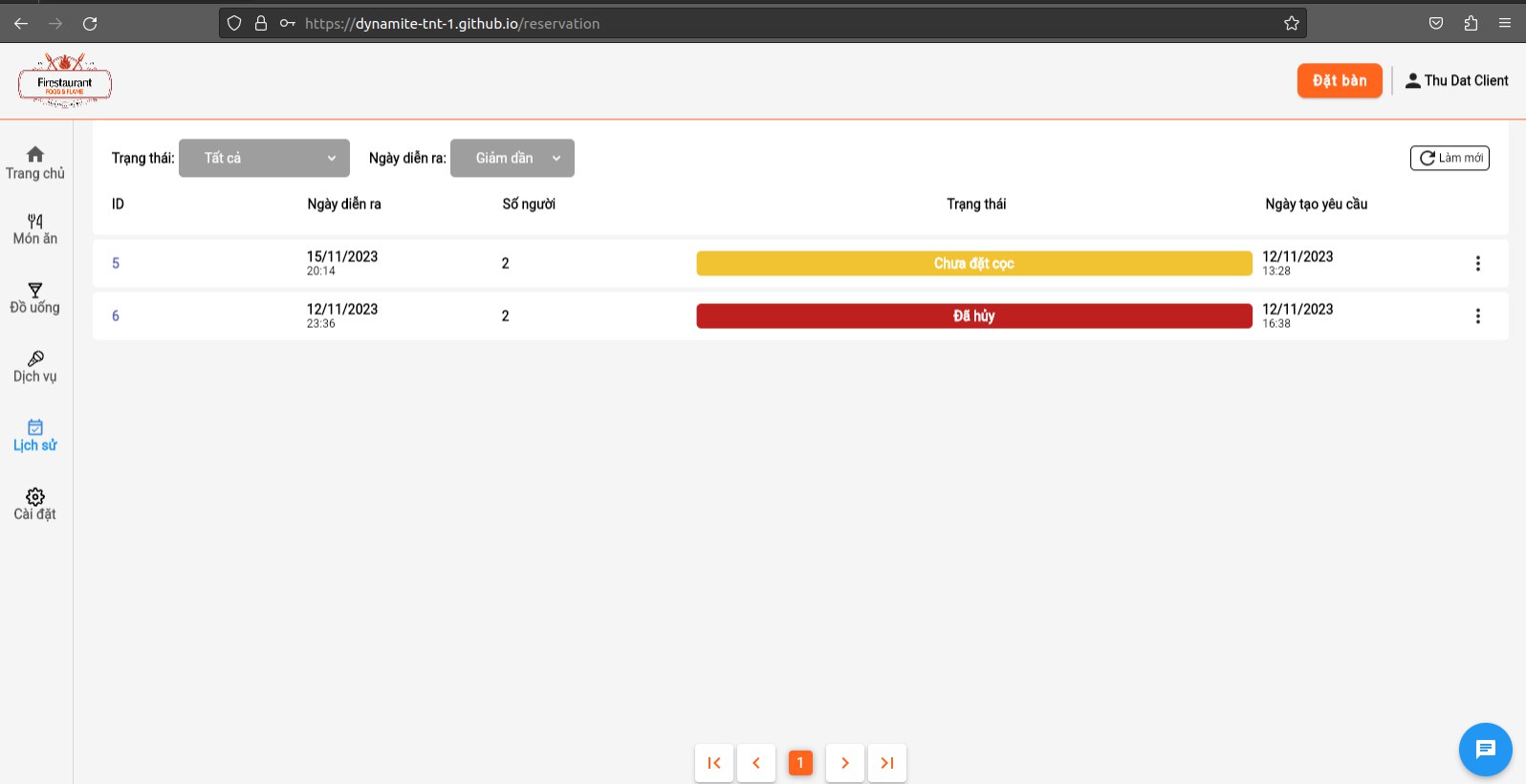
## 5.4. Danh sách đồ uống



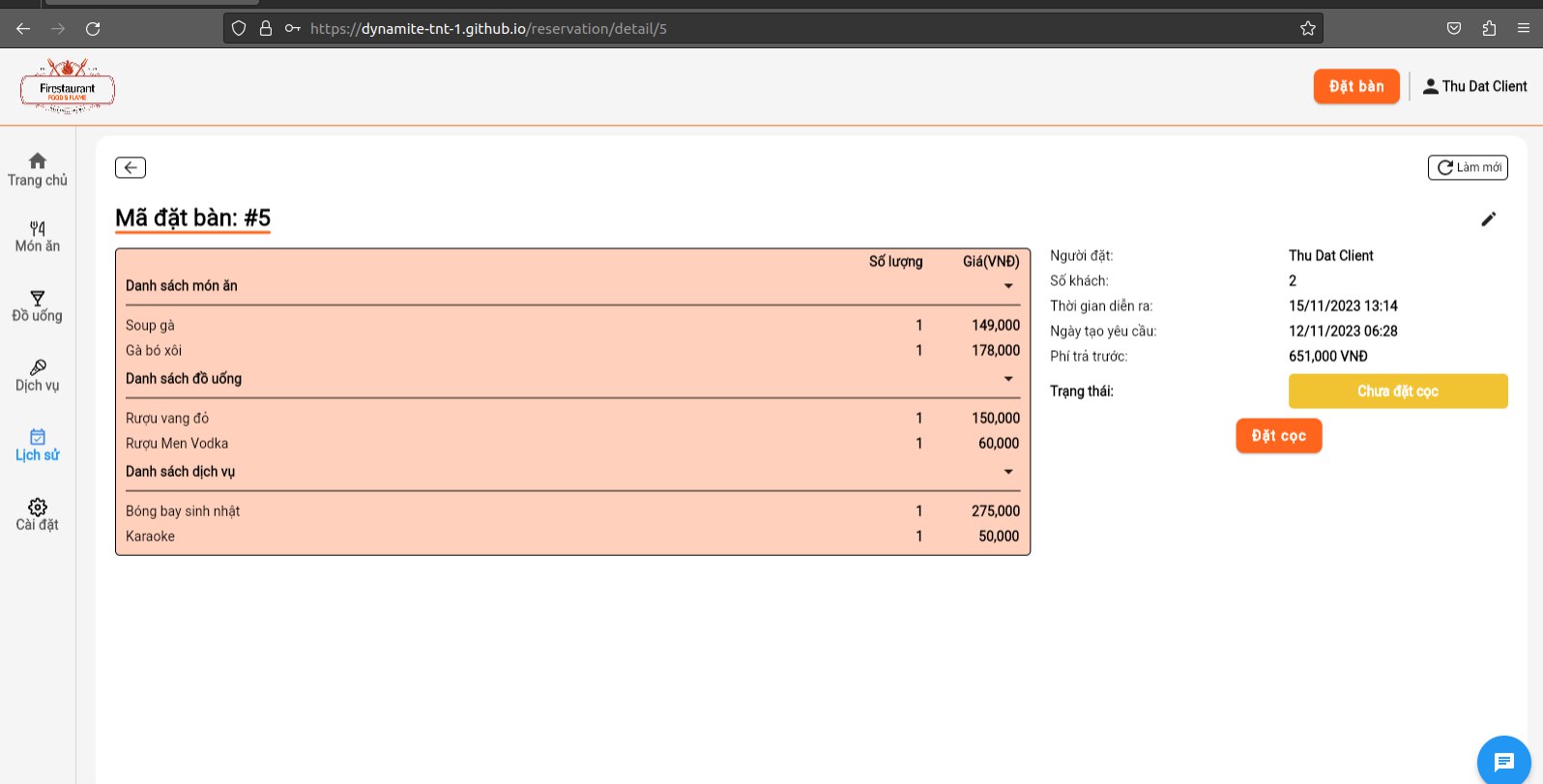
## 5.5. Danh sách dịch vụ



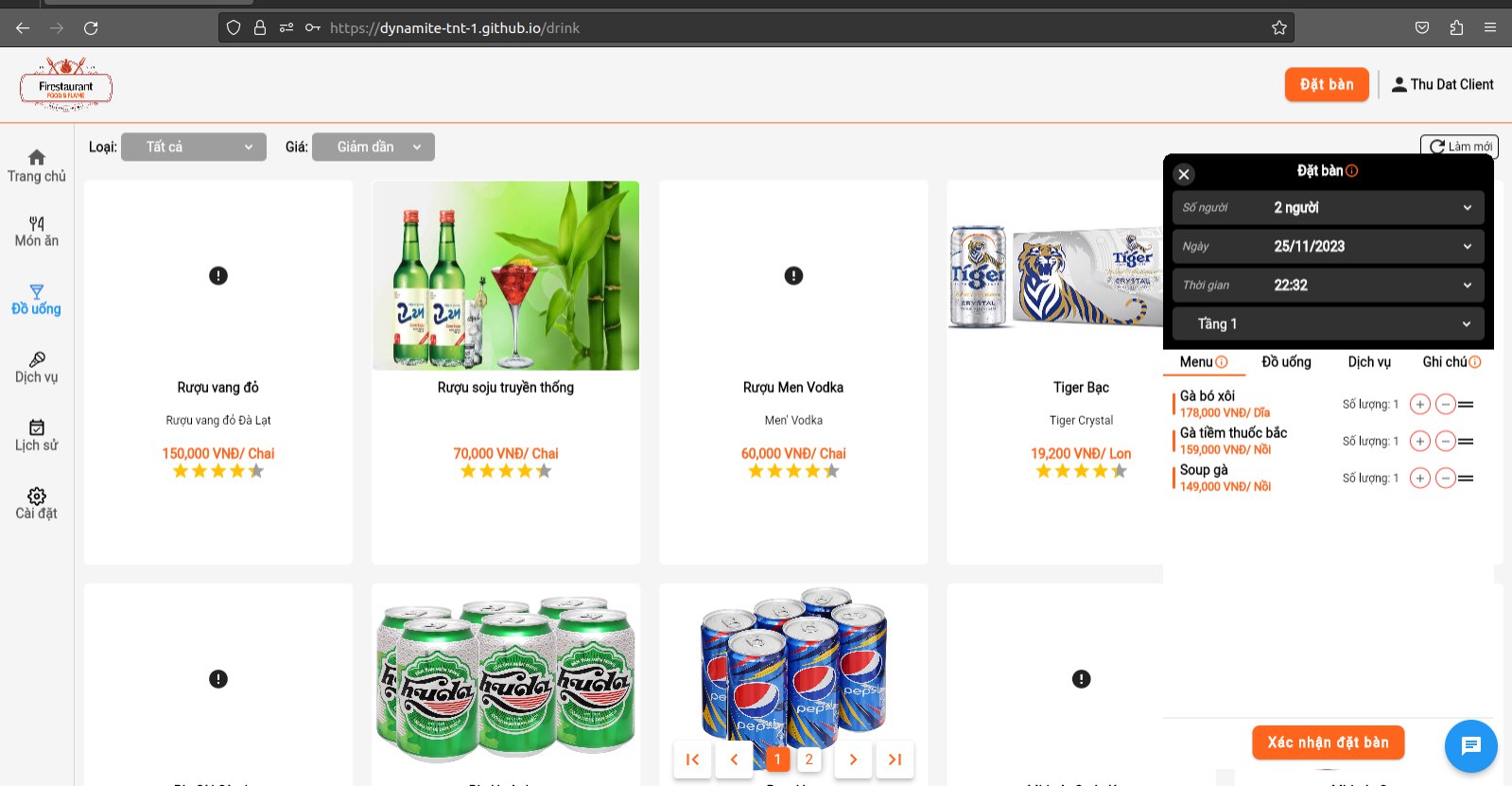
## 5.6 Danh sách lịch sử đặt bàn



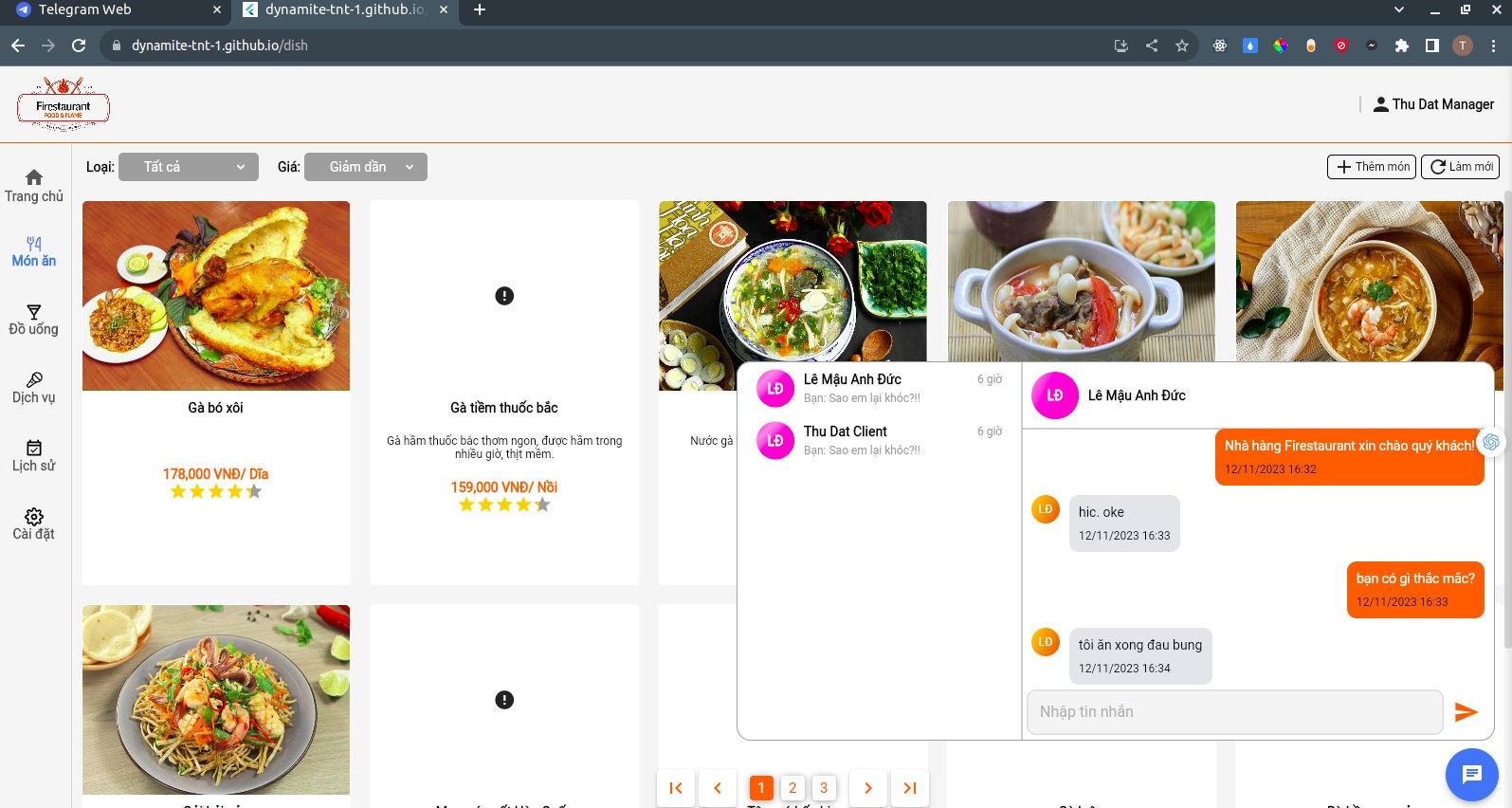
## 5.7. Chi tiết đặt bàn.



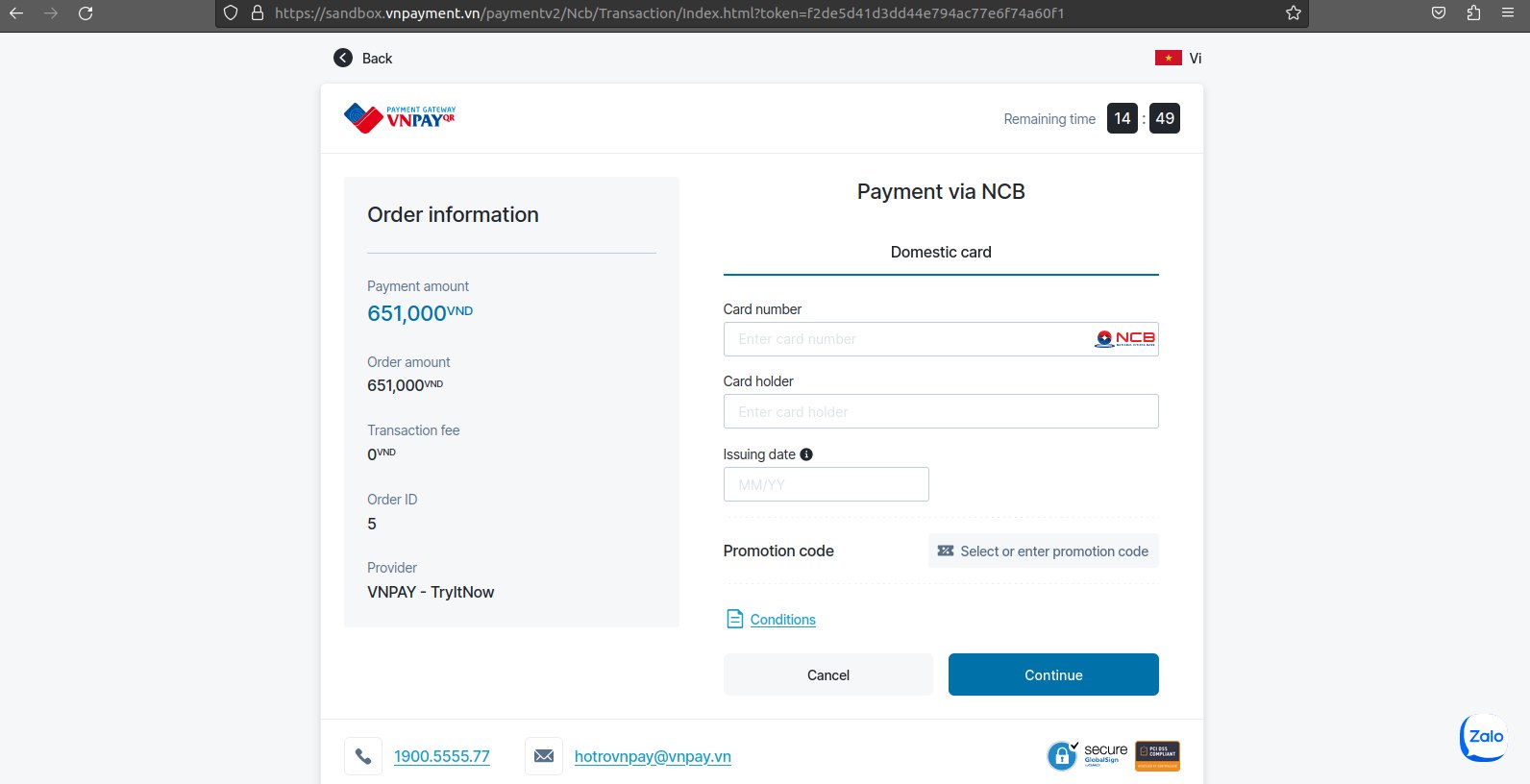
## 5.8. Đặt bàn



## 5.9 Nhắn tin



## 5.10. Thanh toán tiền đặt cọc :



# KẾT LUẬN

## 6.1. Kết quả đạt được

**Lý thuyết**

Sau thời gian thực hiện đề tài em đã đạt được nhiều tiến bộ cả về mặt tìm hiểu, nghiên cứu lý thuyết lẫn kỹ năng lập trình. Có thể nói, thông qua đề tài đã thực hiện, em đã đạt được:

* Hiểu biết nhiều hơn về các kỹ thuật lập trình back-end (Nodejs) và front-end (Flutter Web).
* Phân tích thiết kế theo hướng đối tượng theo hướng chuyên nghiệp hơn.
* Hiểu hơn về nghiệp vụ đặt bàn tại nhà hàng.
* Nâng cao tinh thần tự học, tự tìm hiểu, nghiên cứu.

**Thực hành**

Xây dựng được ứng dụng web đặt bàn tại nhà hàng gồm các chức năng:

* Đăng ký và đăng nhập tài khoản.
* Quản lý món ăn, menu, dịch vụ,…
* Quản lý yêu cầu đặt bàn tại nhà hàng.

## 6.2. Hạn chế

Tuy ứng dụng web đã hoàn thành tuy nhiên vẫn còn những hạn chế trong việc phân tích chuẩn xác và thực hiện hóa ý tưởng.

Trong quá trình thực hiện và phát triển gặp một số khó khăn làm ảnh hướng đến tiến độ thực hiện như sau:

* Thời gian thực hiện dự án gấp nên phân phối thời gian chưa hợp lý.
* Một số nội dung, kiến thức liên quan đến thiết kế giao diện bằng Flutter Web còn hạn chế và xây dựng back-end bằng Nodejs còn thiếu kinh nghiệm nên giao diện sử dụng và tổ chức server chưa được tối ưu.

## 6.3. Hướng phát triển

Hệ thống còn nhiều thiếu xót nhưng trong tương lai có thể phát triển thêm nhằm đáp ứng nhu cầu của người dùng như:

* Xử lý các lỗi chặt chẽ hơn trước khi đưa vào sử dụng.
* Bổ sung nhiều hơn các báo cáo, thống kê.