

**TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG**  
**KHOA CÔNG NGHỆ THÔNG TIN, ROBOT**  
**VÀ TRÍ TUỆ NHÂN TẠO**

---oOo---



**BÁO CÁO TIỂU LUẬN**  
**CHUYÊN ĐỀ 1**

*Tên tiểu luận:*

**XÂY DỰNG WEB ĐẶT BÀN ĂN TẠI MỘT NHÀ**  
**HÀNG**

Giảng viên hướng dẫn:

**Nguyễn Thanh Sơn**

**Hồ Ngọc Giàu**

**Dương Anh Tuấn**

**Nguyễn Hữu Quyền**

Sinh viên thực hiện:

**Phạm Thanh Phong-22050001**

**Nguyễn Văn Khánh-22050079**

**Bình Dương, năm 2025**

[illegible]

# LỜI CẢM ƠN



Với tất cả lòng kính trọng và biết ơn sâu sắc, em xin gửi lời cảm ơn chân thành đến:

Ban Giám hiệu Trường Đại học Bình Dương cùng toàn thể quý thầy cô đã và đang công tác giảng dạy tại trường. Nhà trường đã tạo điều kiện thuận lợi để em có cơ hội tiếp cận với thực tiễn, áp dụng những kiến thức được học trên giảng đường vào thực tế. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến quý thầy cô Khoa Công nghệ Thông tin, những người đã tận tình giảng dạy, truyền đạt cho em những kiến thức quý báu trong suốt quá trình học tập.

Em xin bày tỏ lòng biết ơn chân thành đến **Nhóm giảng viên hướng dẫn** – người đã trực tiếp hướng dẫn em trong quá trình thực hiện báo cáo tiểu luận này. Thầy đã dành thời gian quý báu để tận tình góp ý, chỉnh sửa và định hướng giúp em hoàn thành đề tài một cách tốt nhất.

Những kiến thức đã tiếp thu được trong thời gian học tập tại trường và quá trình thực hiện đề tài sẽ là hành trang quý giá để em áp dụng trong công việc thực tế sau này. Em xin hứa sẽ luôn cố gắng phát huy tốt những kiến thức đã học nhằm đóng góp cho sự phát triển của bản thân và xã hội trong thời kỳ công nghiệp hóa – hiện đại hóa đất nước.

Do thời gian thực hiện có hạn, kiến thức và kinh nghiệm thực tế của bản thân còn nhiều hạn chế, nên bài tiểu luận không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, nhận xét của quý thầy cô để em có thể hoàn thiện hơn trong tương lai.

# MỤC LỤC

|   |    |
|---|----|
| CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI.....  | 1  |
| 1.1. GIỚI THIỆU ĐỀ TÀI.....   | 1  |
| 1.2. MỤC ĐÍCH CỦA ĐỀ TÀI.....   | 1  |
| 1.3. PHẠM VI ĐỀ TÀI.....  | 2  |
| CHƯƠNG 2: TÌM HIỂU HIỆN TRẠNG ĐỀ TÀI VÀ PHÂN TÍCH YÊU CẦU HỆ THỐNG..... | 4  |
| 2.1. HIỆN TRẠNG ĐỀ TÀI.....   | 4  |
| 2.2. YÊU CẦU CÁC CHỨC NĂNG CHÍNH CỦA ĐỀ TÀI.....                        | 4  |
| 2.2.1. Chức năng nghiệp vụ.....   | 4  |
| 2.2.2. Chức năng lưu trữ.....   | 5  |
| 2.2.3. Chức năng tra cứu.....   | 5  |
| 2.3. YÊU CẦU PHI CHỨC NĂNG.....   | 5  |
| 2.3.1. Tính tiến hóa.....   | 5  |
| 2.3.2. Tính tiện dụng.....  | 6  |
| 2.3.3. Tính hiệu quả.....   | 6  |
| 2.3.4. Tính tương thích.....  | 6  |
| 2.3.5. Tính tái sử dụng.....  | 7  |
| 2.3.6. Tính dễ bảo trì.....   | 7  |
| CHƯƠNG 3: PHÂN TÍCH YÊU CẦU.....  | 8  |
| 3.1. Mô hình hóa usecase.....   | 8  |
| 3.1.2. Danh sách Actor và Use Case.....                                 | 8  |
| 3.1.3.Đặc tả usecase.....   | 10 |
| 3.2. Mô hình hóa cấu trúc.....  | 23 |
| 3.3. Mô hình hóa hành vi.....   | 25 |
| 3.3.1. Lược đồ tuần tự.....   | 25 |
| CHƯƠNG 4: THIẾT KẾ HỆ THỐNG.....  | 37 |
| 4.1. Thiết kế cơ sở dữ liệu.....  | 37 |

|   |    |
|---|----|
| 4.1.1. Lược đồ quan hệ .....                                      | 37 |
| 4.1.2. Mô tả dữ liệu .....  | 37 |
| 4.2. Thiết kế giao diện .....                                     | 41 |
| 4.2.1. Giao diện cho nhân viên: .....                             | 41 |
| 4.2.2. Giao diện cho Admin: .....                                 | 42 |
| 4.2.3. Giao diện khách hàng .....                                 | 43 |
| 4.3. Kiến trúc hệ thống .....                                     | 45 |
| CHƯƠNG 5: TRIỂN KHAI TRÊN CLOUD .....                             | 48 |
| 5.1. Đóng gói và triển khai bằng Docker .....                     | 48 |
| 5.1.1. Đóng gói bằng Docker .....                                 | 48 |
| 5.2. Mô hình hệ thống trên AWS .....                              | 49 |
| 5.2.1. Giới thiệu .....   | 49 |
| 5.2.2. Kiến trúc tổng quan .....                                  | 50 |
| 5.2.3. EC2 Linux trong Auto Scaling Group .....                   | 50 |
| 5.2.4. Chi tiết triển khai dịch vụ .....                          | 51 |
| 5.2.5. Lý do chọn mở rộng theo chiều dọc (Vertical Scaling) ..... | 52 |
| 5.2.6. Lớp dữ liệu: Amazon RDS .....                              | 52 |
| 5.2.7. Giám sát và cảnh báo .....                                 | 53 |
| 5.2.8. Khả năng nâng cấp và mở rộng .....                         | 53 |
| 5.3. Kết luận .....   | 53 |
| CHƯƠNG 6: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN .....              | 55 |
| 6.1. KẾT QUẢ ĐẠT ĐƯỢC .....                                       | 55 |
| 6.1.1. Kết quả đạt được .....                                     | 55 |
| 6.1.2 HẠN CHẾ .....   | 55 |
| 6.2. Hình ảnh minh họa hệ thống .....                             | 56 |
| 6.3. Hướng phát triển trong tương lai .....                       | 58 |
| TÀI LIỆU THAM KHẢO .....  | 60 |

## MỤC LỤC HÌNH ẢNH

|   |    |
|---|----|
| Hình 3-1: Usecase tổng quát .....             | 8  |
| Hình 3-2: Usecase DatBan .....                | 10 |
| Hình 3-3: Usecase GoiThemMon .....            | 11 |
| Hình 3-4: Usecase DanhGia .....               | 12 |
| Hình 3-5: Usecase QLDanhGia .....             | 13 |
| Hình 3-6: Usecase QLBanAn .....               | 14 |
| Hình 3-7: Usecase QLHoaDon .....              | 15 |
| Hình 3-8: Usecase QLKhachHang .....           | 16 |
| Hình 3-9: Usecase QLNhanVien .....            | 17 |
| Hình 3-10: Usecase QLThucDon .....            | 18 |
| Hình 3-11: Usecase QLDonHang .....            | 19 |
| Hình 3-12: Usecase DangNhap .....             | 20 |
| Hình 3-13: Usecase DangKy .....               | 21 |
| Hình 3-14: Usecase DangXuat .....             | 21 |
| Hình 3-15: Sơ đồ mối quan hệ thực thể .....   | 23 |
| Hình 3-16: sequence đặt bàn .....             | 25 |
| Hình 3-17 :sequency Đăng Nhập .....           | 27 |
| Hình 3-18:sequency Đăng Xuất .....            | 28 |
| Hình 3-19:sequency Đăng Ký .....              | 29 |
| Hình 3-20: sequency Gọi Thêm Món .....        | 30 |
| Hình 3-21: sequency Đánh giá .....            | 30 |
| Hình 3-22: sequency Quản lý đơn đặt bàn ..... | 31 |
| Hình 3-23:sequency Quản lý Hóa Đơn .....      | 32 |
| Hình 3-24:sequency Quản lý Nhân viên .....    | 33 |

|  |    |
|--|----|
| Hình 3-25: sequency Quản lý Khách hàng.....        | 33 |
| Hình 3-26 :sequency Quản lý bàn ăn.....            | 34 |
| Hình 3-27: sequency Quản lý Thực đơn.....          | 35 |
| Hình 3-27: sequency Quản lý Đánh Giá.....          | 36 |
| Hình 4-1: lược đồ quan hệ.....                     | 37 |
| Hình 4-2: Giao diện quản lý bàn.....               | 41 |
| Hình 4-3: Giao diện quản lý danh sách đặt bàn..... | 42 |
| Hình 4-4: Giao diện quản lý người dùng.....        | 42 |
| Hình 4-5: Giao diện quản lý thực đơn.....          | 43 |
| Hình 4-6: Giao diện trang chủ.....                 | 43 |
| Hình 4-7: Giao diện đặt bàn.....                   | 44 |
| Hình 4-8: Giao diện chi tiết đặt bàn.....          | 44 |
| Hình 4-9: Giao diện chi tiết đặt món.....          | 45 |
| Hình 4-10: Kiến trúc hệ thống.....                 | 47 |
| Hình 5-1: Sơ đồ cloud.....                         | 48 |

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## 1.1. GIỚI THIỆU ĐỀ TÀI

Website đặt bàn ăn cho một nhà hàng là hệ thống quản lý đặt bàn trực tuyến giúp khách hàng có thể dễ dàng chọn bàn, đặt chỗ theo thời gian mong muốn, theo dõi các chương trình ưu đãi và cập nhật các món ăn mới từ nhà hàng.

Website quản lý nhiều bàn ăn tại các khu vực khác nhau trong nhà hàng. Mỗi bàn ăn có mã bàn, tên bàn, vị trí, số lượng chỗ ngồi, trạng thái (có sẵn, đã đặt, đang sử dụng), khu vực, thông tin khu vực bao gồm mã khu vực, tên khu vực.

Mỗi bàn có thể được đặt trong một khung giờ nhất định trong ngày. Thông tin về lịch đặt bàn gồm mã đặt bàn, mã khách hàng, mã bàn, ngày giờ đặt, ghi chú, trạng thái đặt bàn (chờ xác nhận, đã xác nhận, đã hủy).

Thông tin món ăn và thực đơn cũng được quản lý trong website, bao gồm mã món ăn, tên món ăn, giá, hình ảnh, mô tả, loại món ăn (khai vị, món chính, tráng miệng, nước uống...). Mỗi loại món ăn sẽ thuộc về một nhóm danh mục tương ứng.

Khách hàng muốn đặt bàn cần đăng ký tài khoản, cung cấp thông tin như: họ tên, số điện thoại, email, ngày sinh... Sau khi đăng nhập, khách hàng có thể đặt bàn, gọi món trước và theo dõi lịch sử đặt chỗ.

Mỗi tài khoản người dùng được phân quyền (quản trị, nhân viên, khách hàng), thông tin quyền bao gồm mã quyền, tên quyền.

Ngoài ra, hệ thống hỗ trợ tư vấn trực tuyến để khách hàng được giải đáp các thắc mắc nhanh chóng. Thông tin hỗ trợ bao gồm mã hỗ trợ, mã khách hàng, nội dung hỗ trợ, hình ảnh (nếu có).

## 1.2. MỤC ĐÍCH CỦA ĐỀ TÀI

- Xây dựng một website giúp khách hàng dễ dàng tìm kiếm và đặt bàn tại nhà hàng theo thời gian và vị trí mong muốn (vị trí bàn trong nhà hàng).
- Giúp nhà hàng quản lý hiệu quả các lịch đặt bàn, tránh tình trạng đặt trùng hoặc bỏ trống bàn không cần thiết.
- Giao diện thân thiện, dễ sử dụng cho cả người dùng phổ thông và quản trị



viên.

- Hỗ trợ hiển thị rõ ràng các món ăn, chương trình ưu đãi nhằm tăng sự thu hút với khách hàng.
- Hệ thống hoạt động ổn định, tốc độ tải nhanh và xử lý chính xác dữ liệu đặt chỗ.
- Ban quản trị có thể theo dõi và xử lý thông tin đặt chỗ, thông tin khách hàng, món ăn và các khu vực bàn ăn một cách hiệu quả.

### 1.3. PHẠM VI ĐỀ TÀI

Hệ thống được xây dựng theo kiến trúc tách biệt frontend và backend, sử dụng Django làm backend và React làm frontend. Cụ thể:

- Backend (Django + Django REST Framework): Xử lý logic nghiệp vụ và quản lý dữ liệu gồm người dùng, bàn ăn, khu vực, món ăn, đơn đặt bàn, đơn hàng. Dữ liệu được lưu trữ trong hệ quản trị cơ sở dữ liệu MySQL và được truy cập thông qua ORM của Django.
- Frontend (React + Vite + TailwindCSS): Hiển thị giao diện người dùng như đặt bàn, xem thực đơn, giỏ món, đăng ký/đăng nhập,... Giao diện được xây dựng theo hướng component hóa với hiệu năng cao nhờ cơ chế render phía client. TailwindCSS giúp tăng tốc quá trình xây dựng giao diện, dễ responsive.
- Giao tiếp frontend-backend thông qua RESTful API, sử dụng định dạng JSON.

Ưu điểm của kiến trúc Django + React (tách frontend-backend):

- Phân tách rõ frontend và backend, dễ bảo trì, mở rộng, và phân công theo vai trò lập trình viên.
- REST API giúp giao tiếp dữ liệu hiệu quả, có thể tái sử dụng cho mobile app hoặc các hệ thống khác.
- React + Vite mang lại tốc độ load nhanh, trải nghiệm mượt mà, thích hợp với SPA (Single Page Application).
- TailwindCSS giúp xây dựng UI nhanh chóng, gọn gàng, dễ kiểm soát.

Công nghệ sử dụng trong hệ thống:

- Ngôn ngữ lập trình: Python (Django), JavaScript ES6+ (React)

- Backend: Django + Django REST Framework (DRF)
- Frontend: ReactJS + Vite + TailwindCSS
- Cơ sở dữ liệu: MySQL
- Giao tiếp frontend-backend: RESTful API (dạng JSON)

## CHƯƠNG 2: TÌM HIỂU HIỆN TRẠNG ĐỀ TÀI VÀ PHÂN TÍCH YÊU CẦU HỆ THỐNG

### 2.1. HIỆN TRẠNG ĐỀ TÀI

Ngày nay, với sự phát triển nhanh chóng của công nghệ và nhu cầu ngày càng cao trong lĩnh vực ẩm thực, việc đặt bàn tại nhà hàng một cách thủ công đã không còn phù hợp với xu hướng hiện đại. Khách hàng mong muốn có thể đặt bàn, xem thực đơn, đặt món, đánh giá dịch vụ... chỉ bằng vài thao tác đơn giản qua điện thoại hoặc máy tính.

Tuy nhiên, phần lớn các nhà hàng nhỏ và vừa hiện nay vẫn đang quản lý việc đặt bàn theo phương pháp truyền thống như ghi tay hoặc qua cuộc gọi điện thoại, điều này dễ dẫn đến nhầm lẫn, đặt trùng bàn, hoặc bỏ sót thông tin khách hàng.

Hiểu được thực tế đó, nhóm chúng em quyết định chọn đề tài "Xây dựng Website đặt bàn ăn cho một nhà hàng" nhằm cung cấp một công cụ hỗ trợ người dùng và nhân viên nhà hàng trong việc quản lý đặt bàn, thực đơn, hóa đơn, đồng thời nâng cao trải nghiệm của khách hàng.

Website sẽ là một nền tảng trực tuyến cho phép người dùng:

Đăng ký/đăng nhập tài khoản.

Đặt bàn theo ngày giờ và khu vực mong muốn.

Gọi thêm món trực tuyến.

Đánh giá và xem tin tức khuyến mãi của nhà hàng.

Nhân viên và quản trị viên có thể dễ dàng xử lý đơn đặt, quản lý hóa đơn, thực đơn, bàn ăn và khách hàng một cách hiệu quả.

### 2.2. YÊU CẦU CÁC CHỨC NĂNG CHÍNH CỦA ĐỀ TÀI

#### 2.2.1. Chức năng nghiệp vụ

| STT | Công việc          | Chức năng |
|-----|--------------------|-----------|
| 1   | Quản lý nhân viên  | Lưu trữ   |
| 2   | Quản lý khách hàng | Lưu trữ   |

|    |                     |         |
|----|---------------------|---------|
| 3  | Quản lý bàn ăn      | Lưu trữ |
| 4  | Quản lý thực đơn    | Lưu trữ |
| 5  | Quản lý đánh giá    | Lưu trữ |
| 6  | Quản lý đơn đặt bàn | Lưu trữ |
| 7  | Quản lý hóa đơn     | Lưu trữ |
| 8  | Đăng nhập           | Lưu trữ |
| 9  | Đăng ký             | Lưu trữ |
| 10 | Đăng xuất           | Lưu trữ |
| 11 | Đặt bàn             | Lưu trữ |
| 12 | Gọi thêm món        | Lưu trữ |
| 13 | Đánh giá            | Lưu trữ |
| 14 | Tìm kiếm            | Tra cứu |

### **2.2.2. Chức năng lưu trữ**

Lưu trữ toàn bộ thông tin về người dùng, đặt bàn, hóa đơn, thực đơn.

Ghi nhận thay đổi (thêm, sửa, xóa) dữ liệu.

### **2.2.3. Chức năng tra cứu**

Người dùng có thể tìm kiếm bàn còn trống, món ăn.

Admin và nhân viên có thể tra cứu khách hàng, hóa đơn, lịch sử đặt bàn.

## **2.3. YÊU CẦU PHI CHỨC NĂNG**

### **2.3.1. Tính tiến hóa**

Hệ thống được thiết kế theo kiến trúc tách biệt frontend-backend và sử dụng RESTful API, cho phép dễ dàng mở rộng về mặt chức năng trong tương lai như:

- Tích hợp thanh toán trực tuyến qua VNPay, Momo,...
- Gợi ý món ăn theo lịch sử đơn hàng hoặc sở thích người dùng (dựa trên học máy).

- Triển khai mobile app sử dụng chung backend hiện có.

### ***2.3.2. Tính tiện dụng***

- Giao diện người dùng được thiết kế responsive, hỗ trợ tốt cả desktop và thiết bị di động.
- Phân tách rõ giữa giao diện khách hàng, nhân viên phục vụ và quản trị viên.
- Các thao tác chính (chọn món, đặt bàn, thanh toán) đều thực hiện không quá 3 bước.
- Font chữ, màu sắc và bố cục được tối ưu cho người lớn tuổi hoặc người ít tiếp xúc công nghệ.

### ***2.3.3. Tính hiệu quả***

- Hệ thống có thể phục vụ tối thiểu 100-200 người dùng đồng thời (concurrent users) trên server cấu hình trung bình (2 vCPU, 4 GB RAM) mà không bị nghẽn hoặc sập, nhờ cache hợp lý và API gọn nhẹ.
- Thời gian phản hồi API trung bình dưới 300ms cho các thao tác chính như đặt bàn, tìm kiếm món ăn.
- Cho phép lưu trữ tối thiểu 5.000 ảnh món ăn, với dung lượng mỗi ảnh không vượt quá 500KB (ảnh được nén trước khi upload), tương đương tổng dung lượng lưu trữ ảnh khoảng 2.5 GB.
- Ảnh được lưu bằng hệ thống file hoặc sử dụng dịch vụ lưu trữ đám mây (ví dụ Amazon S3, Cloudinary) để tối ưu hiệu năng khi cần mở rộng quy mô.
- Hệ thống hỗ trợ mở rộng theo chiều ngang nếu số lượng nhà hàng hoặc người dùng tăng, bằng cách deploy dạng container (Docker) và load balancing nếu cần thiết.

### ***2.3.4. Tính tương thích***

- Hệ thống frontend (React) hoạt động tốt trên các trình duyệt hiện đại gồm: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.
- Giao diện được thiết kế responsive bằng TailwindCSS, đảm bảo tương thích trên đa thiết bị: PC, laptop, máy tính bảng, smartphone.
- Backend xây dựng bằng Django (Python), có thể chạy ổn định trên nhiều hệ điều hành máy chủ như Ubuntu, CentOS.

- Cơ sở dữ liệu sử dụng: MySQL, có thể triển khai trên máy chủ vật lý hoặc dịch vụ đám mây (MySQL Server, RDS...).

### ***2.3.5. Tính tái sử dụng***

- Hệ thống được chia thành các lớp riêng biệt: frontend, backend, cơ sở dữ liệu – giúp dễ dàng bảo trì và tái sử dụng từng phần độc lập.
- Backend xây dựng theo kiến trúc RESTful API, cho phép các hệ thống khác (mobile app, hệ thống quản lý khác) tái sử dụng dữ liệu và chức năng thông qua HTTP.
- Frontend React được xây dựng theo mô hình component hóa, cho phép tái sử dụng các thành phần như nút, thẻ món ăn, form đặt bàn... trong nhiều giao diện khác nhau chỉ bằng cách truyền props.

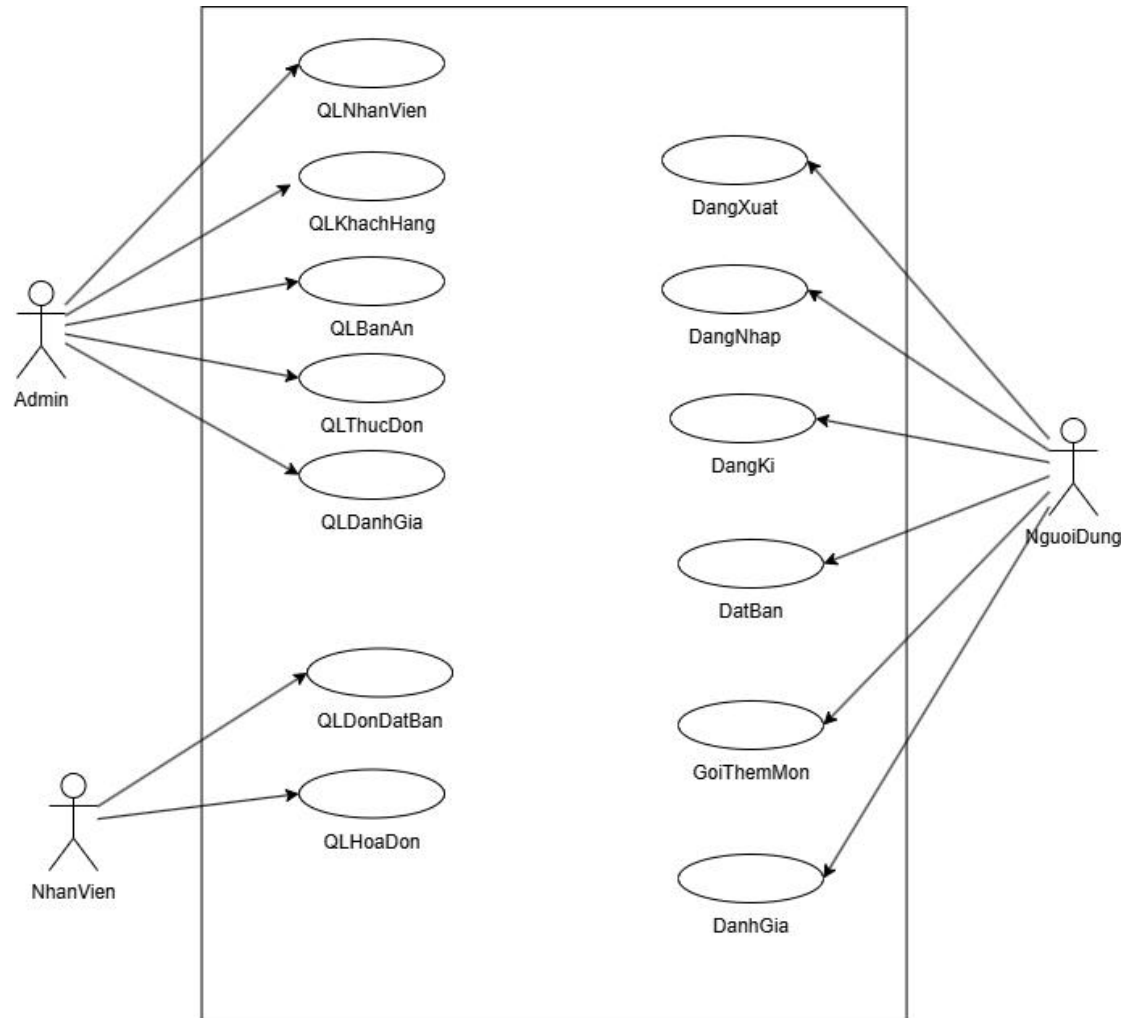
### ***2.3.6. Tính dễ bảo trì***

- Code được tổ chức theo chuẩn:
  - Frontend chia theo component, page, và service/API riêng biệt.
  - Backend tách rõ apps trong Django (vd: `booking`, `menu`, `users`...), dùng cấu trúc views – serializers – models rõ ràng.
- Tài liệu hướng dẫn sử dụng API (Swagger hoặc Postman collection) được xây dựng đầy đủ.
- Sử dụng migrations của Django để quản lý thay đổi cơ sở dữ liệu dễ dàng khi bảo trì hoặc nâng cấp.
- Ghi chú trong mã nguồn, cấu trúc repo rõ ràng giúp lập trình viên khác tiếp cận nhanh chóng.

## CHƯƠNG 3: PHÂN TÍCH YÊU CẦU

### 3.1. Mô hình hóa usecase

#### 3.1.1. Mô hình usecase tổng quát



Hình 3-1: Usecase tổng quát

#### 3.1.2. Danh sách Actor và Use Case

##### 3.1.2.1. Danh sách Actor

| Mã | Actor     | Ý nghĩa   |
|----|-----------|---|
| A1 | Admin     | Người quản trị hệ thống, có toàn quyền truy cập, quản lý nhân viên, khách hàng, bàn ăn, thực đơn, đánh giá. |
| A2 | Nhân viên | Quản lý hóa đơn, bàn, đơn đặt bàn,... dưới sự phân  |

|    |            |  |
|----|------------|--|
|    |            | công của Admin.  |
| A3 | Người dùng | Người dùng có chức năng đặt bàn, gọi thêm món, đánh giá,.. |

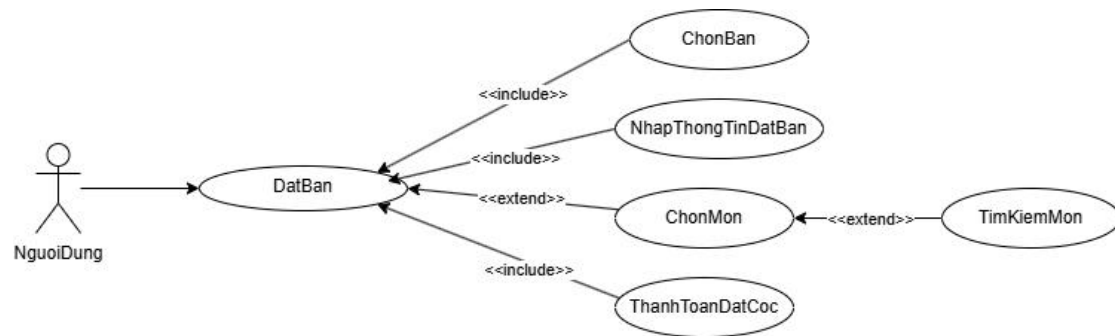
### 3.1.2.2. Danh sách Use Case

| STT | Tên Use-Case | Actor | Ý nghĩa  |
|-----|--------------|-------|--|
| 1   | DangNhap     | A3    | Đăng nhập hệ thống .   |
| 2   | DangKy       | A3    | Đăng ký tài khoản mới.   |
| 3   | DangXuat     | A3    | Đăng xuất khỏi hệ thống.   |
| 4   | DatBan       | A3    | Đặt bàn theo ngày giờ và số lượng người.   |
| 5   | GoiThemMon   | A3    | Gọi thêm món khi đang dùng tại nhà hàng  |
| 6   | DanhGia      | A3    | Gửi đánh giá về món ăn và dịch vụ.   |
| 7   | QLNhanVien   | A1    | Admin có thể thêm, sửa, xóa Nhân viên.   |
| 8   | QLKhachHang  | A1    | Admin có thể thêm, sửa, xóa Khách Hàng   |
| 9   | QLBanAn      | A1    | Admin có thể thêm, sửa, xóa Bàn ăn   |
| 10  | QLThucDon    | A1    | Admin có thể thêm, sửa, xóa các món ăn   |
| 11  | QLDanhGia    | A1    | Admin xem các đánh giá của khách   |
| 12  | QLDonDatBan  | A2    | Nhân viên xem danh sách và chi tiết các đơn đặt bàn, có quyền hủy đơn nếu chưa đến giờ đặt. Khi hủy, hệ thống cập nhật trạng thái và giải phóng bàn. |
| 13  | QLHoaDon     | A2    | Nhân viên xem danh sách hóa đơn, xác nhận thanh toán, in hóa đơn hoặc gửi lại cho khách. Không được phép chỉnh sửa nội dung hóa đơn.                 |



### 3.1.3.Đặc tả usecase

#### 3.1.3.1. Usecase "DatBan"



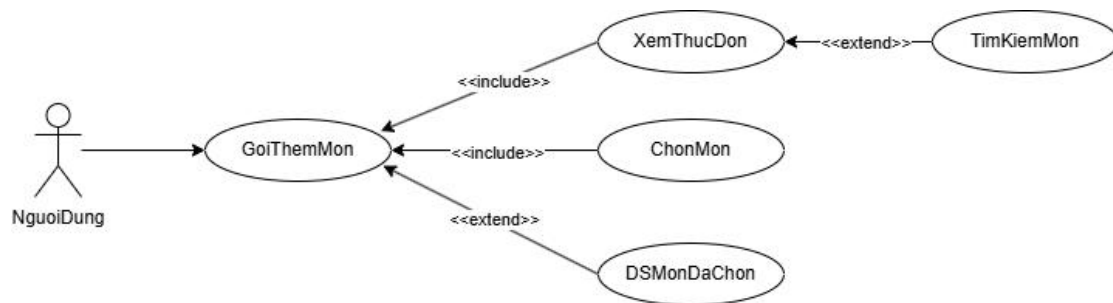
Hình 3-2: Usecase DatBan

- ❖ Tóm tắt:  
Use Case này mô tả cách người dùng (khách hàng) thực hiện thao tác đặt bàn tại nhà hàng thông qua hệ thống.
- ❖ Tác nhân:  
Người dùng (A3)
- ❖ Tiên điều kiện (Pre-condition):  
Người dùng đã đăng nhập vào hệ thống và có tài khoản hợp lệ.
- ❖ Hậu điều kiện (Post-condition):  
Một đơn đặt bàn mới được tạo và lưu trữ vào hệ thống. Bàn tương ứng được cập nhật trạng thái thành "Đã đặt".
- ❖ Luồng sự kiện chính (Main Flow):
  1. Người dùng đăng nhập thành công vào hệ thống.
  2. Chọn mục "Đặt bàn" từ giao diện chính.
  3. Hệ thống hiển thị danh sách bàn còn trống theo ngày và giờ cụ thể.
  4. Người dùng chọn bàn, nhập số người, ngày và giờ đặt, ghi chú nếu có.
  5. Nhấn "Xác nhận đặt bàn".
  6. Hệ thống kiểm tra tính hợp lệ (bàn chưa được đặt trong khoảng thời gian đó).
  7. Hệ thống lưu thông tin đặt bàn, cập nhật trạng thái bàn và hiển thị thông báo thành công.

❖ Luồng phụ (Alternative Flow):

- Nếu người dùng chọn giờ không hợp lệ (trong quá khứ), hệ thống sẽ thông báo lỗi.
- Nếu bàn đã được đặt vào khung giờ đó, hệ thống hiển thị thông báo và yêu cầu chọn bàn khác.

3.1.3.2. Usecase "GoiThemMon"



Hình 3-3: Usecase GoiThemMon

❖ Tóm tắt:

Cho phép người dùng gọi thêm món ăn trong khi đã có đặt bàn trước đó.

❖ Tác nhân:

Người dùng (A3)

❖ Tiền điều kiện:

Người dùng đã đăng nhập và có đơn đặt bàn đang hoạt động.

❖ Hậu điều kiện:

Món ăn được thêm vào đơn hàng liên kết với đặt bàn hiện tại.

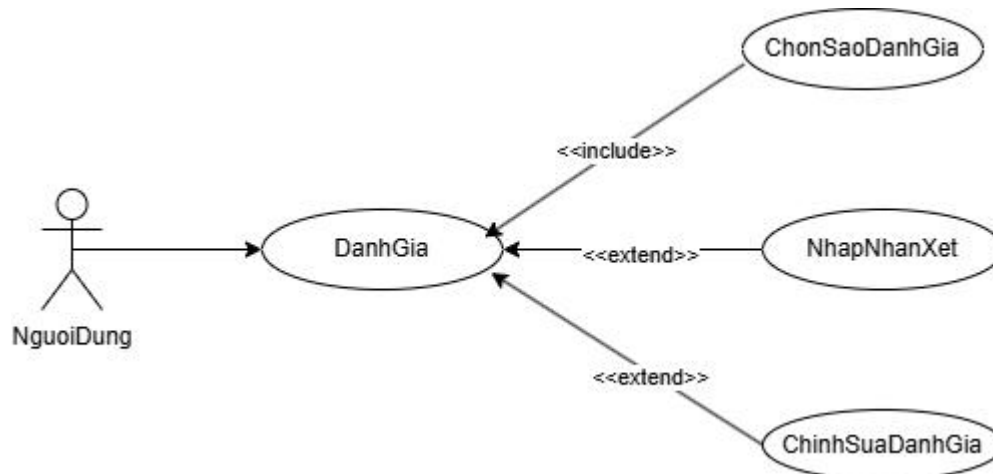
❖ Luồng sự kiện chính:

1. Người dùng đăng nhập.
2. Truy cập mục "Gọi thêm món".
3. Hệ thống hiển thị danh sách món ăn trong thực đơn.
4. Người dùng chọn món và số lượng.
5. Nhấn "Thêm vào đơn".
6. Hệ thống kiểm tra món hợp lệ và cập nhật vào chi tiết đơn hàng.
7. Hiển thị thông báo gọi món thành công.

❖ Luồng phụ:

- Nếu người dùng không có đơn đặt bàn, hệ thống báo lỗi và yêu cầu đặt bàn trước.
- Nếu món đã bị ngừng phục vụ, hệ thống thông báo không thể gọi món.

#### 3.1.3.3. Usecase "DanhGia"



Hình 3-4: Usecase DanhGia

❖ Tóm tắt:

Người dùng gửi đánh giá về món ăn và dịch vụ sau khi hoàn tất thanh toán.

❖ Tác nhân:

Người dùng (A3)

❖ Tiền điều kiện:

Người dùng đã thanh toán đơn hàng.

❖ Hậu điều kiện:

Đánh giá được lưu vào hệ thống và liên kết với đơn thanh toán tương ứng.

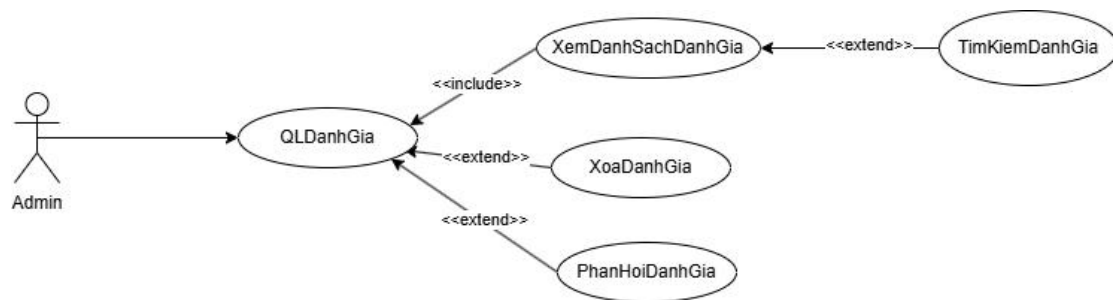
❖ Luồng sự kiện chính:

1. Người dùng đăng nhập.
2. Vào mục "Đánh giá".
3. Hệ thống hiển thị các đơn hàng đã hoàn tất.
4. Chọn đơn và nhập: điểm (1–5), nội dung đánh giá.
5. Nhấn "Gửi đánh giá".
6. Hệ thống lưu dữ liệu vào CSDL và thông báo gửi thành công.

❖ Luồng phụ:

- Nếu chưa có đơn thanh toán, hệ thống từ chối đánh giá.
- Nếu bỏ trống nội dung hoặc điểm, hệ thống báo lỗi.

#### 3.1.3.4. Usecase "QLDanhGia"



Hình 3-5: Usecase QLDanhGia

❖ Tóm tắt:

Admin xem danh sách các đánh giá của khách hàng để phân tích chất lượng dịch vụ.

❖ Tác nhân:

Admin (A1)

❖ Tiền điều kiện:

Admin đã đăng nhập hệ thống.

❖ Hậu điều kiện:

Danh sách đánh giá được hiển thị, không có thay đổi dữ liệu.

❖ Luồng sự kiện chính:

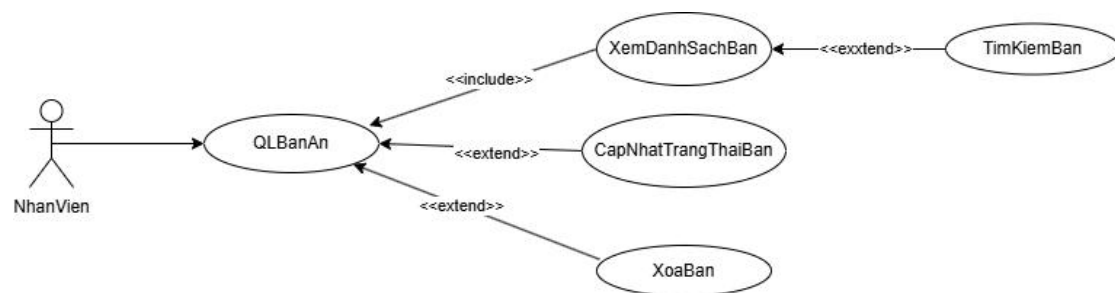
Admin đăng nhập.

1. Truy cập "Quản lý đánh giá".
2. Hệ thống hiển thị bảng đánh giá: người đánh giá, nội dung, điểm số, thời gian.
3. Admin có thể lọc theo điểm hoặc thời gian.

❖ Luồng phụ:

- Nếu không có đánh giá nào, hệ thống hiển thị thông báo rỗng.

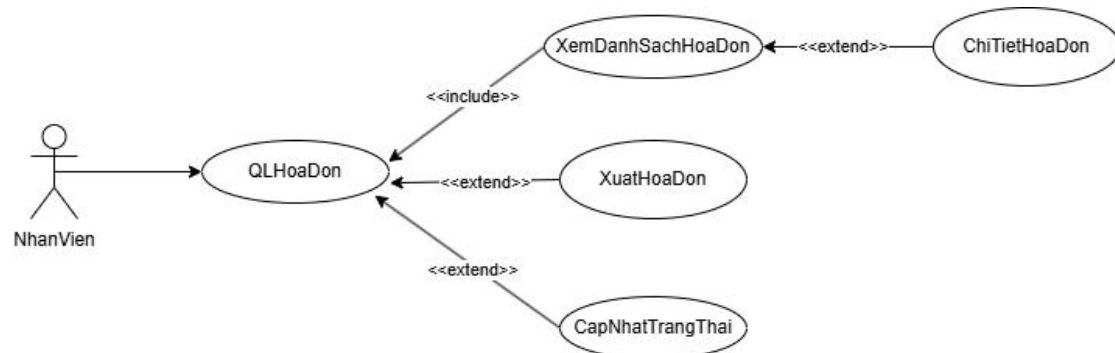
### 3.1.3.5. Usecase "QLBanAn"



Hình 3-6: Usecase QLBanAn

- ❖ Tóm tắt:  
Admin thực hiện thêm, sửa, xóa thông tin bàn ăn.
- ❖ Tác nhân:  
Admin (A1)
- ❖ Tiền điều kiện:  
Đã đăng nhập với quyền admin.
- ❖ Hậu điều kiện:  
Dữ liệu bàn ăn được cập nhật trong CSDL.
- ❖ Luồng sự kiện chính:
  1. Admin chọn mục "Bàn ăn".
  2. Thêm/sửa/xóa bàn ăn gồm: tên bàn, số chỗ, vị trí.
  3. Hệ thống kiểm tra hợp lệ và cập nhật CSDL.
  4. Hiển thị danh sách bàn mới nhất.
- ❖ Luồng phụ:
  - Nếu bàn đang được đặt, không thể xóa.
  - Nếu thông tin thiếu, hiển thị thông báo lỗi.

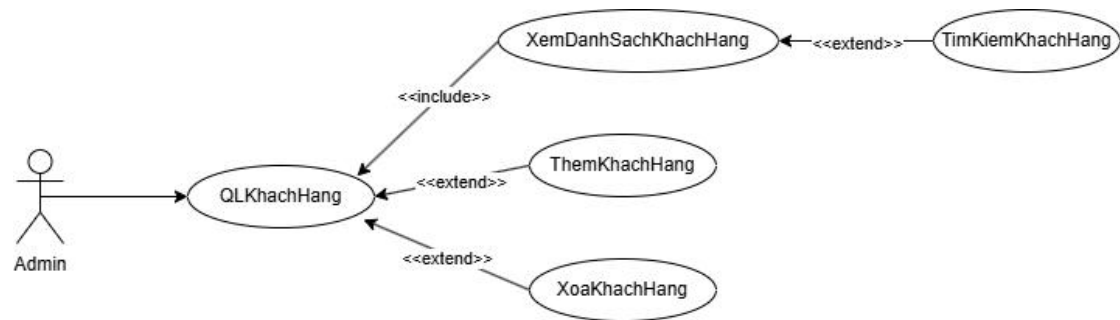
### 3.1.3.6. Usecase "QLHoaDon"



Hình 3-7: Usecase QLHoaDon

- ❖ Tóm tắt:  
Nhân viên xử lý hóa đơn sau khi khách dùng bữa, bao gồm xác nhận và in.
- ❖ Tác nhân:  
Nhân viên (A2)
- ❖ Tiên điều kiện:  
Nhân viên đã đăng nhập.
- ❖ Hậu điều kiện:  
Hóa đơn được xác nhận và đánh dấu là đã thanh toán.
- ❖ Luồng sự kiện chính:
  1. Nhân viên vào "Quản lý hóa đơn".
  2. Chọn hóa đơn chưa thanh toán.
  3. Kiểm tra chi tiết, xác nhận thanh toán.
  4. Hệ thống đánh dấu trạng thái "Đã thanh toán".
  5. Nhân viên có thể in hoặc gửi hóa đơn qua email.
- ❖ Luồng phụ:
  - Nếu hóa đơn đã thanh toán, không thể chỉnh sửa.
  - Nếu đơn liên kết chưa hoàn tất, từ chối thanh toán.

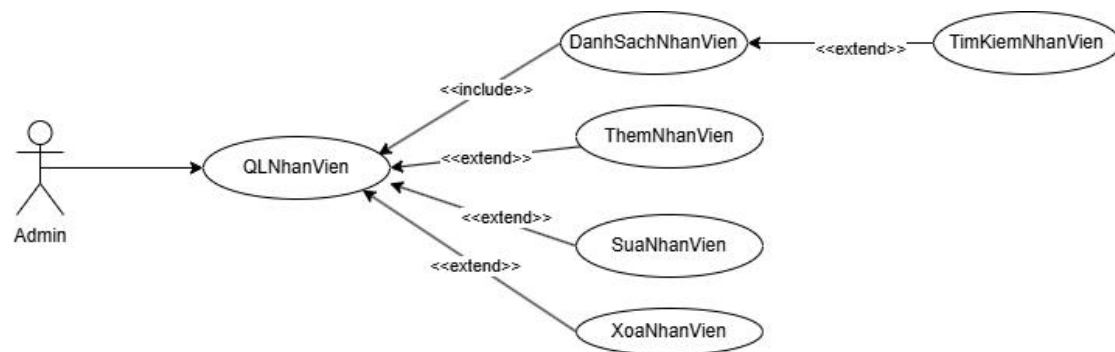
### 3.1.3.7. Usecase "QLKhachHang"



Hình 3-8: Usecase QLKhachHang

- ❖ Tóm tắt:  
Admin quản lý thông tin khách hàng: thêm, sửa, hoặc xóa nếu cần thiết.
- ❖ Tác nhân:  
Admin (A1)
- ❖ Tiền điều kiện:  
Đã đăng nhập hệ thống với quyền Admin.
- ❖ Hậu điều kiện:  
Danh sách khách hàng được cập nhật trong hệ thống.
- ❖ Luồng sự kiện chính:
  1. Admin vào mục "Quản lý khách hàng".
  2. Hệ thống hiển thị danh sách khách hàng.
  3. Admin thực hiện thao tác: Thêm/Sửa/Xóa khách hàng.
  4. Hệ thống kiểm tra hợp lệ và cập nhật vào CSDL.
  5. Hiển thị danh sách khách hàng mới nhất.
- ❖ Luồng phụ:
  - Không thể xóa nếu khách hàng còn đơn đặt bàn hoặc hóa đơn liên quan.
  - Nếu nhập thiếu thông tin, hệ thống sẽ cảnh báo.

### 3.1.3.8. Usecase "QLNhanVien"

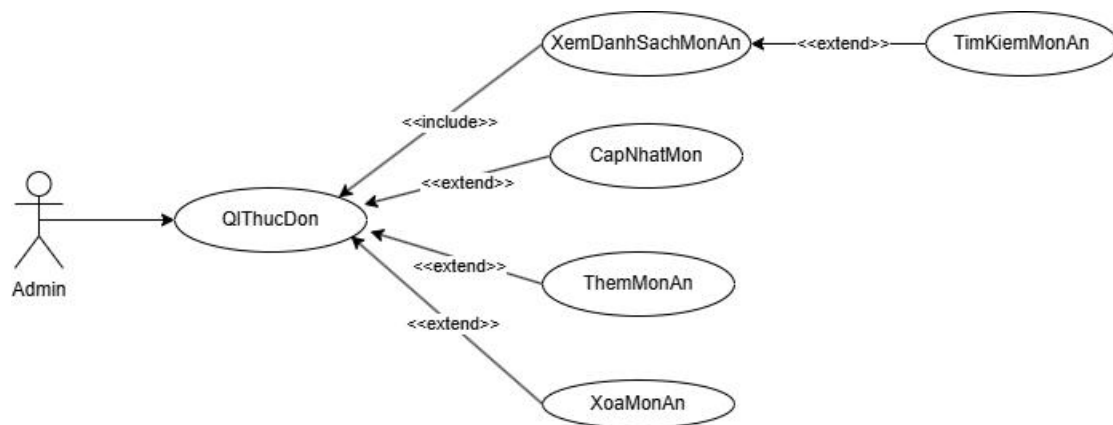


Hình 3-9: Usecase QLNhanVien

- ❖ Tóm tắt:  
Admin quản lý nhân viên: thêm mới, sửa hoặc xóa tài khoản nhân viên.
- ❖ Tác nhân:  
Admin (A1)
- ❖ Tiên điều kiện:  
Đã đăng nhập với quyền Admin.
- ❖ Hậu điều kiện:  
Danh sách nhân viên được cập nhật.
- ❖ Luồng sự kiện chính:
  1. Vào mục "Quản lý nhân viên".
  2. Thực hiện thêm mới/sửa/xóa nhân viên.
  3. Hệ thống kiểm tra hợp lệ, lưu vào CSDL.
  4. Hiện thị danh sách nhân viên mới nhất.
- ❖ Luồng phụ:
  - Không thể xóa nếu nhân viên đang xử lý hóa đơn hoặc đơn đặt bàn.
  - Cảnh báo nếu thiếu thông tin hoặc email bị trùng.

### 3.1.3.9. Usecase "QLThucDon"

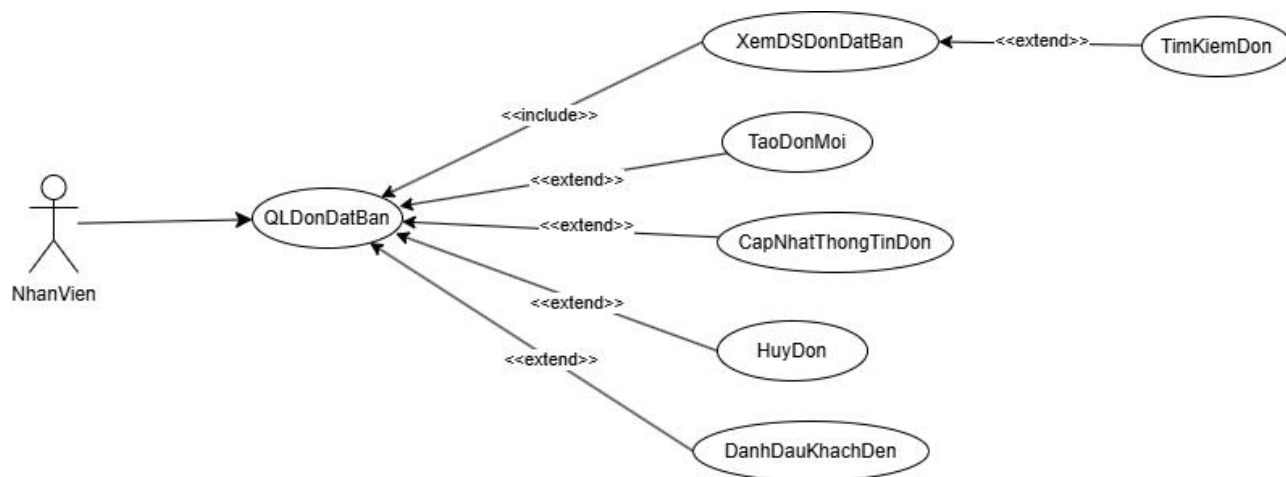




Hình 3-10: Usecase QLThucDon

- ❖ Tóm tắt:  
Admin thêm, sửa, xóa món ăn và phân loại món ăn trong thực đơn.
- ❖ Tác nhân:  
Admin (A1)
- ❖ Tiền điều kiện:  
Đã đăng nhập hệ thống với quyền Admin.
- ❖ Hậu điều kiện:  
Dữ liệu thực đơn được cập nhật.
- ❖ Luồng sự kiện chính:
  1. Truy cập mục "Thực đơn".
  2. Chọn Thêm/Sửa/Xóa món ăn.
  3. Nhập thông tin món: tên, giá, loại món, hình ảnh.
  4. Hệ thống lưu thay đổi vào CSDL.
- ❖ Luồng phụ:
  - Nếu món đang có trong đơn hàng thì không thể xóa.
  - Hình ảnh quá dung lượng quy định sẽ bị từ chối.

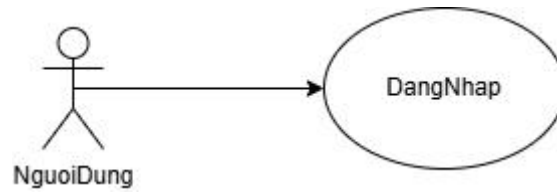
### 3.1.3.10. Usecase "QLDonDatBan"



*Hình 3-11: Usecase QLDonHang*

- ❖ Tóm tắt:  
Nhân viên quản lý đơn đặt bàn của khách hàng, có thể hủy hoặc xác nhận bàn.
- ❖ Tác nhân:  
Nhân viên (A2)
- ❖ Tiền điều kiện:  
Nhân viên đã đăng nhập.
- ❖ Hậu điều kiện:  
Đơn đặt bàn được cập nhật trạng thái và bàn được giải phóng nếu hủy.
- ❖ Luồng sự kiện chính:
  1. Nhân viên vào mục "Đơn đặt bàn".
  2. Xem danh sách các đơn đặt.
  3. Nhấn "Hủy" nếu khách báo hủy hoặc không đến.
  4. Hệ thống cập nhật trạng thái đơn và giải phóng bàn.
  5. Hiện thị danh sách cập nhật.
- ❖ Luồng phụ:
  - Không thể hủy nếu đơn đã quá giờ hoặc đã thanh toán.

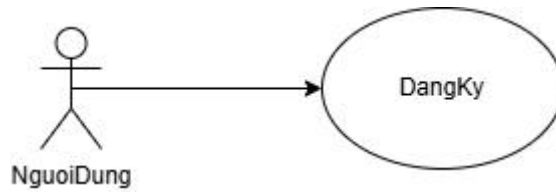
### 3.1.3.11. Usecase "DangNhap"



Hình 3-12: Usecase DangNhap

- ❖ Tóm tắt:  
Người dùng đăng nhập để sử dụng các chức năng đặt bàn, gọi món...
- ❖ Tác nhân:  
Người dùng (A3)
- ❖ Tiền điều kiện:  
Đã có tài khoản đăng ký trước đó.
- ❖ Hậu điều kiện:  
Người dùng được xác thực và truy cập giao diện tương ứng với quyền.
- ❖ Luồng sự kiện chính:
  1. Người dùng truy cập trang đăng nhập.
  2. Nhập email và mật khẩu.
  3. Nhấn "Đăng nhập".
  4. Hệ thống kiểm tra thông tin và đăng nhập nếu hợp lệ.
- ❖ Luồng phụ:
  - Nếu sai thông tin: thông báo lỗi.
  - Nếu tài khoản bị khóa: báo tài khoản không hợp lệ.

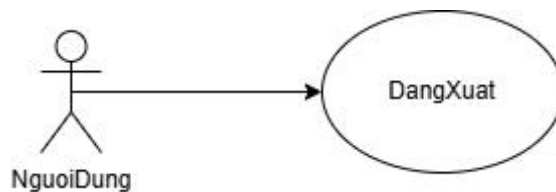
### 3.1.3.12. Usecase "DangKy"



*Hình 3-13: Usecase DangKy*

- ❖ Tóm tắt:  
Người dùng tạo tài khoản để sử dụng dịch vụ.
- ❖ Tác nhân:  
Người dùng (A3)
- ❖ Tiền điều kiện:  
Chưa có tài khoản trong hệ thống.
- ❖ Hậu điều kiện:  
Tài khoản người dùng mới được tạo và lưu vào CSDL.
- ❖ Luồng sự kiện chính:
  1. Người dùng vào trang đăng ký.
  2. Nhập thông tin: họ tên, email, SDT, mật khẩu.
  3. Nhấn "Đăng ký".
  4. Hệ thống kiểm tra hợp lệ và lưu tài khoản mới.
- ❖ Luồng phụ:
  - Email đã tồn tại: báo lỗi.
  - Thiếu thông tin: hiển thị thông báo yêu cầu bổ sung.

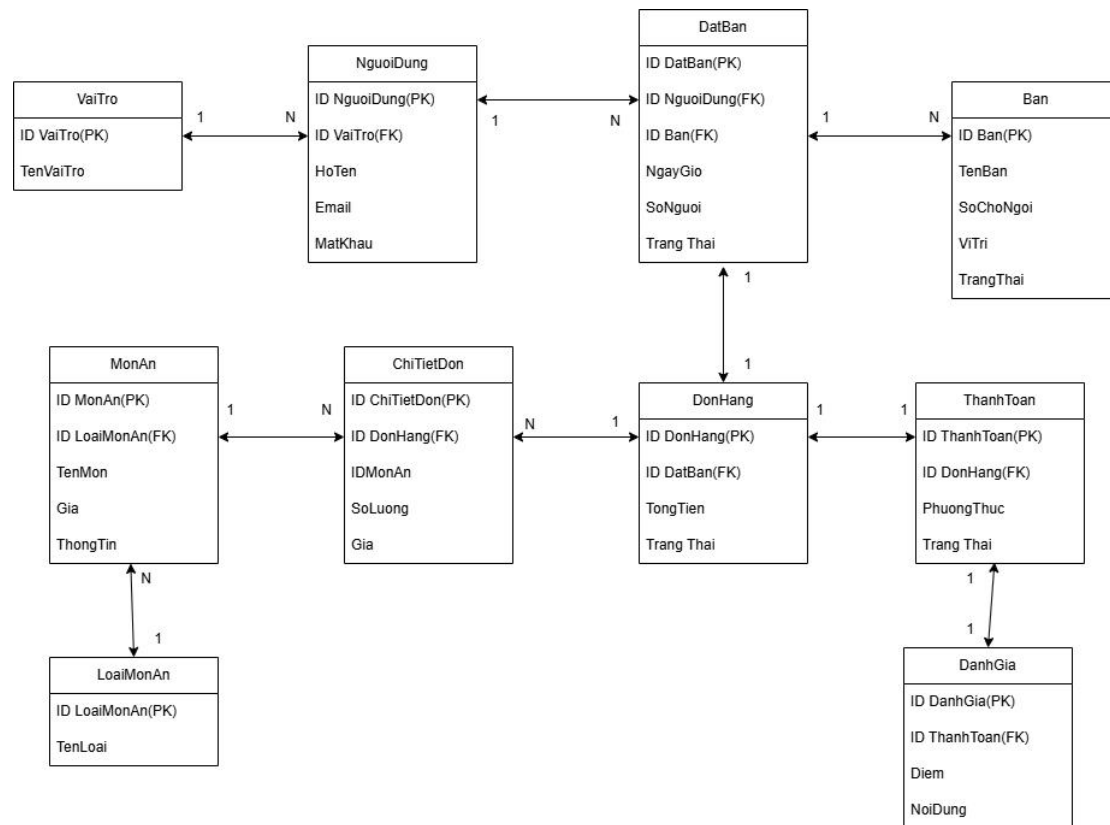
#### *3.1.3.13. Usecase "DangXuat"*



*Hình 3-14: Usecase DangXuat*

- ❖ Tóm tắt:  
Người dùng đăng xuất khỏi hệ thống.
- ❖ Tác nhân:  
Người dùng (A3)
- ❖ Tiền điều kiện:  
Đã đăng nhập hệ thống.
- ❖ Hậu điều kiện:  
Phiên đăng nhập bị hủy và trở về giao diện chính.
- ❖ Luồng sự kiện chính:
  1. Người dùng nhấn nút "Đăng xuất".
  2. Hệ thống xóa phiên làm việc.
  3. Quay về trang chủ hoặc đăng nhập.
- ❖ Luồng phụ:
  - Nếu không đăng nhập, không hiển thị nút “Đăng xuất”.

### 3.2. Mô hình hóa cấu trúc



Hình 3-15: Sơ đồ mối quan hệ thực thể

#### Mô tả sơ đồ mối quan hệ

1. Mỗi người dùng (NguoiDung) thuộc một vai trò (VaiTro), và một vai trò có thể gán cho nhiều người dùng → quan hệ 1-N.
2. Một người dùng có thể thực hiện nhiều lượt đặt bàn (DatBan), nhưng mỗi lượt đặt bàn chỉ thuộc về một người dùng → quan hệ 1-N.
3. Một bàn (Ban) có thể được đặt nhiều lần qua bảng DatBan, nhưng mỗi lượt đặt chỉ gán với một bàn → quan hệ 1-N.
4. Một lượt đặt bàn chỉ tạo ra một đơn hàng (DonHang), và mỗi đơn hàng tương ứng với một lượt đặt → quan hệ 1-1.
5. Một đơn hàng gồm nhiều món ăn thông qua bảng trung gian ChiTietDon, và một món ăn có thể xuất hiện trong nhiều đơn hàng → quan hệ N-N (thể hiện qua bảng ChiTietDon).

6. Mỗi món ăn thuộc về một loại món ăn (LoaiMonAn), và mỗi loại có thể chứa nhiều món ăn  $\rightarrow$  quan hệ 1-N.

7. Mỗi đơn hàng có thể có một thanh toán (ThanhToan), và mỗi thanh toán chỉ gắn với một đơn hàng  $\rightarrow$  quan hệ 1-1.

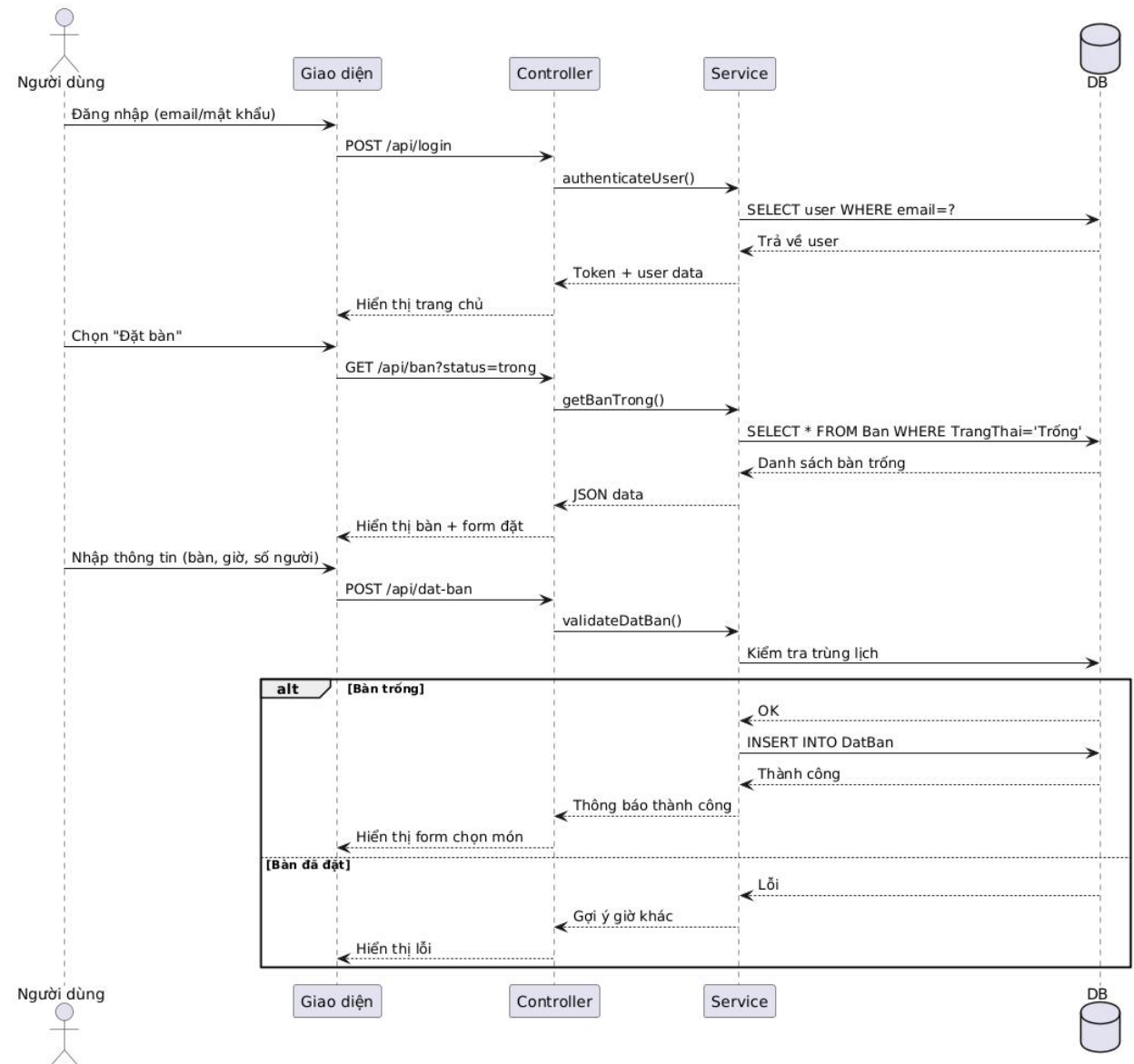
8. Một thanh toán có thể nhận được một đánh giá (DanhGia), và mỗi đánh giá chỉ liên quan đến một thanh toán  $\rightarrow$  quan hệ 1-1.

### 3.3. Mô hình hóa hành vi

#### 3.3.1. Lược đồ tuần tự

##### 3.3.1.1. Người dùng

- Usecase “DatBan”



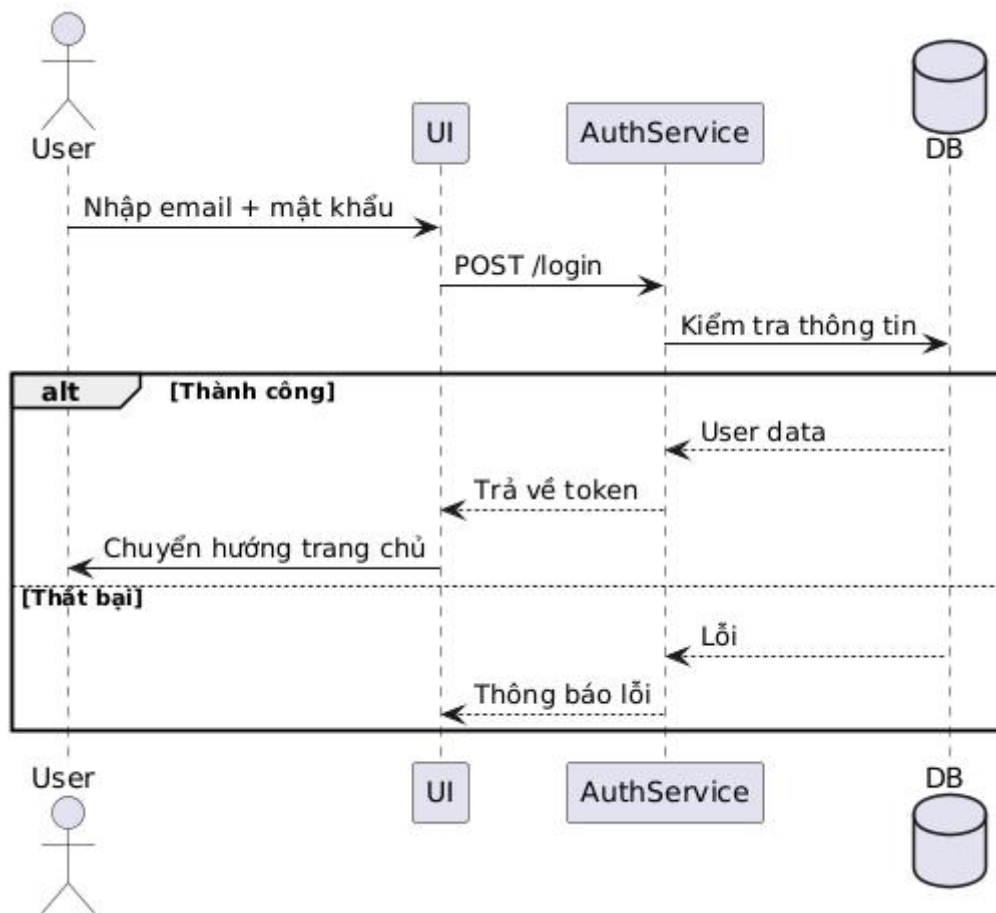
Hình 3-16: sequence đặt bàn



## **Dòng sự kiện chính**

1. Người dùng đăng nhập bằng email và mật khẩu. Hệ thống kiểm tra thông tin và cho phép truy cập nếu hợp lệ.
2. Sau khi đăng nhập, hệ thống hiển thị trang chủ cho người dùng.
3. Người dùng chọn chức năng "Đặt bàn", hệ thống tìm và hiển thị danh sách các bàn đang trống.
4. Người dùng nhập thông tin đặt bàn như số bàn, thời gian và số người.
5. Hệ thống kiểm tra bàn đã được đặt hay chưa:
  - + Nếu bàn còn trống, hệ thống lưu thông tin đặt bàn và chuyển sang bước chọn món.
  - + Nếu bàn đã được đặt, hệ thống thông báo lỗi và gợi ý người dùng chọn giờ khác.

- Usecase “ĐăngNhập”

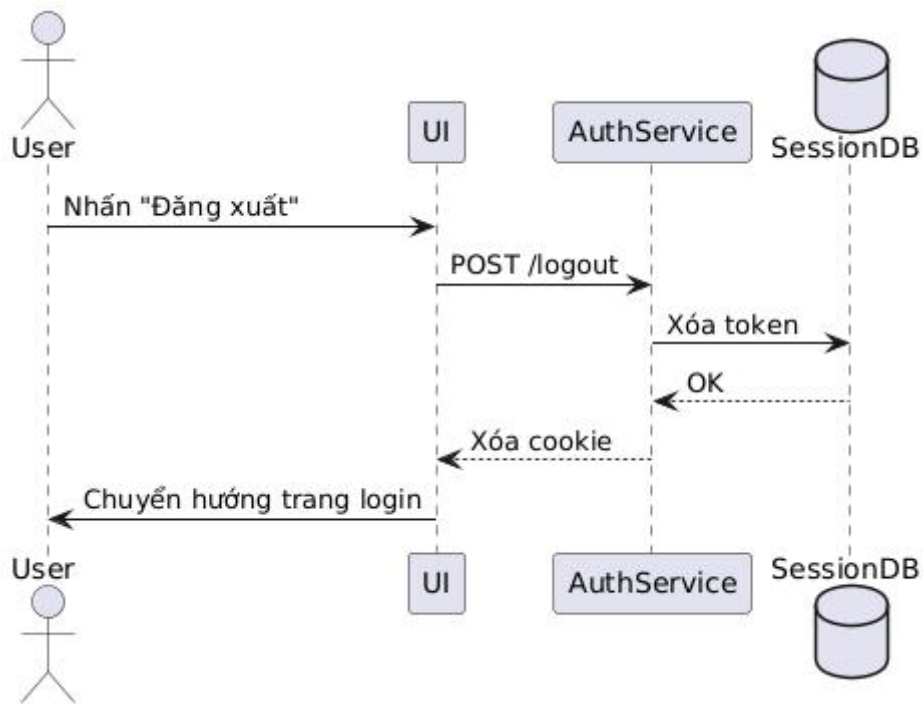


Hình 3-17 :sequency Đăng Nhập

### Dòng sự kiện chính

1. Người dùng nhập email và mật khẩu vào giao diện.
2. Hệ thống xác thực thông tin bằng cách so sánh với dữ liệu trong cơ sở dữ liệu.
  - + Thành công: Tạo token xác thực và chuyển hướng người dùng đến trang chủ.
  - + Thất bại: Hiện thị thông báo lỗi (ví dụ: "Sai mật khẩu" hoặc "Email không tồn tại").

- Usecase “ĐăngXuat”

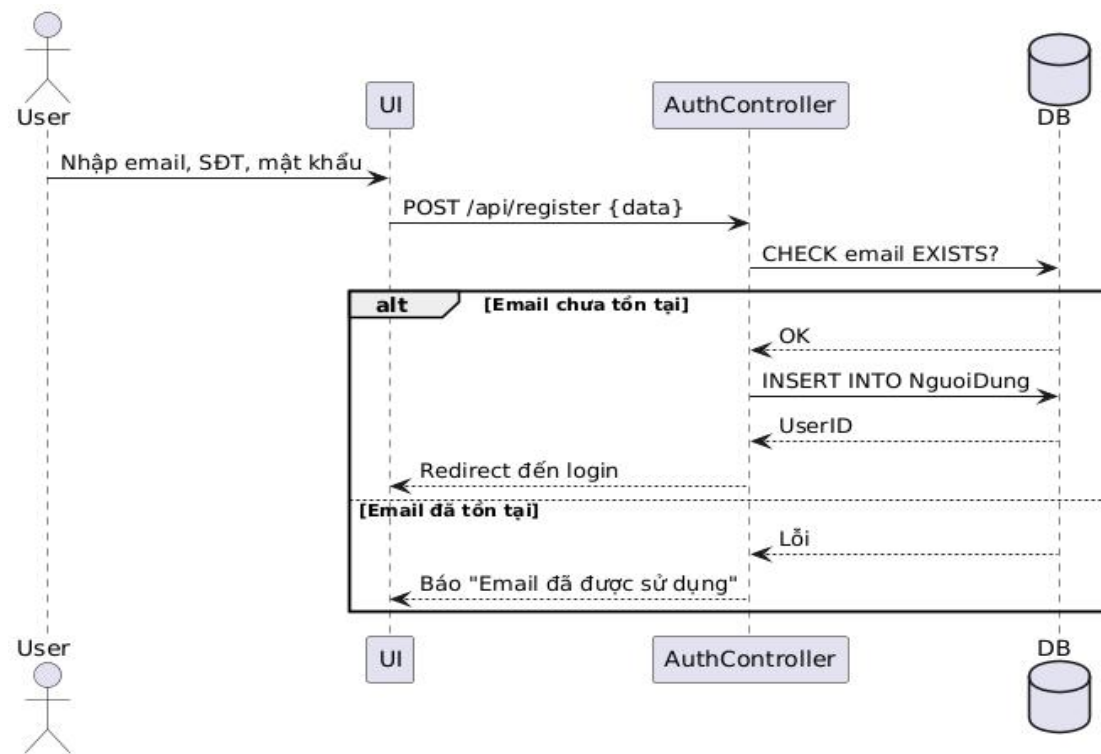


Hình 3-18:sequency Đăng Xuất

### Dòng sự kiện chính

1. Người dùng nhấn nút "Đăng xuất".
2. Hệ thống xóa thông tin phiên đăng nhập và chuyển hướng về trang đăng nhập.

- Usecase “ĐăngKý”

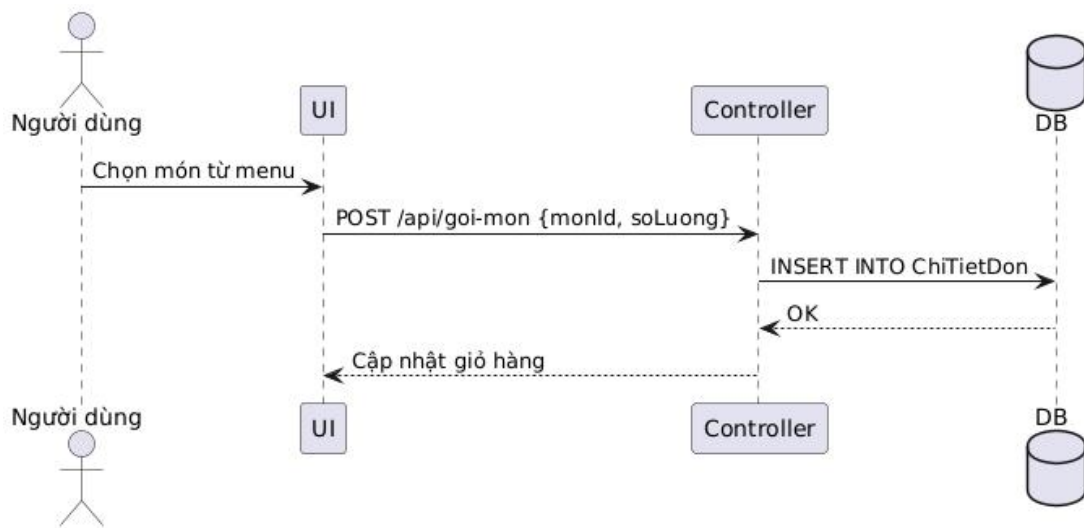


Hình 3-19:sequency Đăng Ký

### Dòng sự kiện chính

1. Người dùng nhập email, số điện thoại, mật khẩu vào form đăng ký.
2. Hệ thống kiểm tra trùng lặp email trong cơ sở dữ liệu:
  - + Email chưa tồn tại: Lưu thông tin người dùng vào cơ sở dữ liệu và chuyển hướng đến trang đăng nhập.
  - + Email đã tồn tại: Thông báo lỗi "Email đã được sử dụng" và yêu cầu nhập lại.

- Usecase “GoiThemMon”

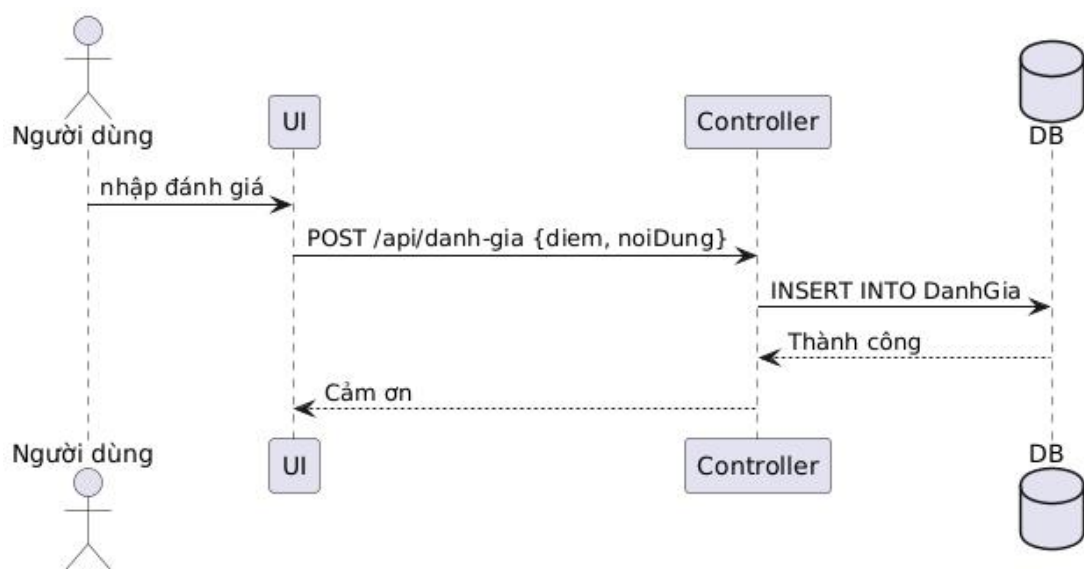


Hình 3-20: sequency Gọi Thêm Món

### Dòng sự kiện chính

1. Người dùng chọn món từ menu và nhập số lượng.
2. Hệ thống thêm món vào đơn hàng và cập nhật giỏ hàng.
3. Thông báo thành công khi hoàn tất.

- Usecase “DanhGia”



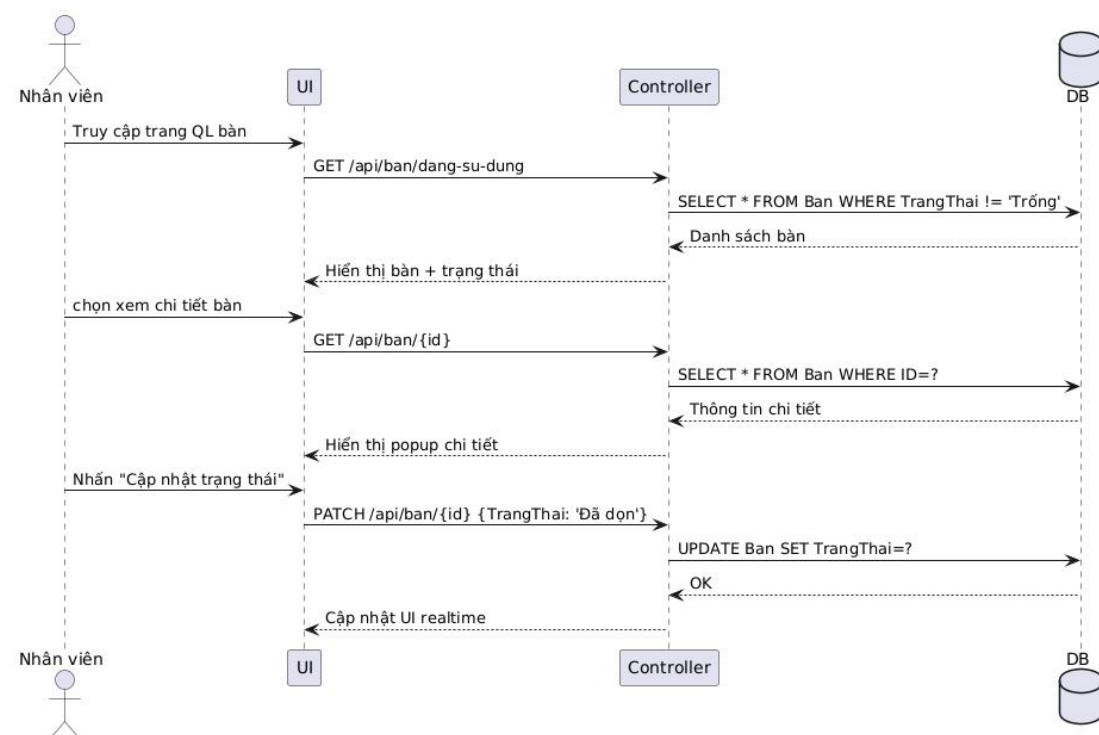
Hình 3-21: sequency Đánh giá

## Dòng sự kiện chính

1. Người dùng nhập điểm số và nội dung đánh giá qua giao diện.
2. Hệ thống lưu thông tin đánh giá vào cơ sở dữ liệu.
3. Hiện thị thông báo "Cảm ơn" khi hoàn tất.

### 3.3.1.2. Nhân Viên

#### ● Usecase “QLDonDatBan”

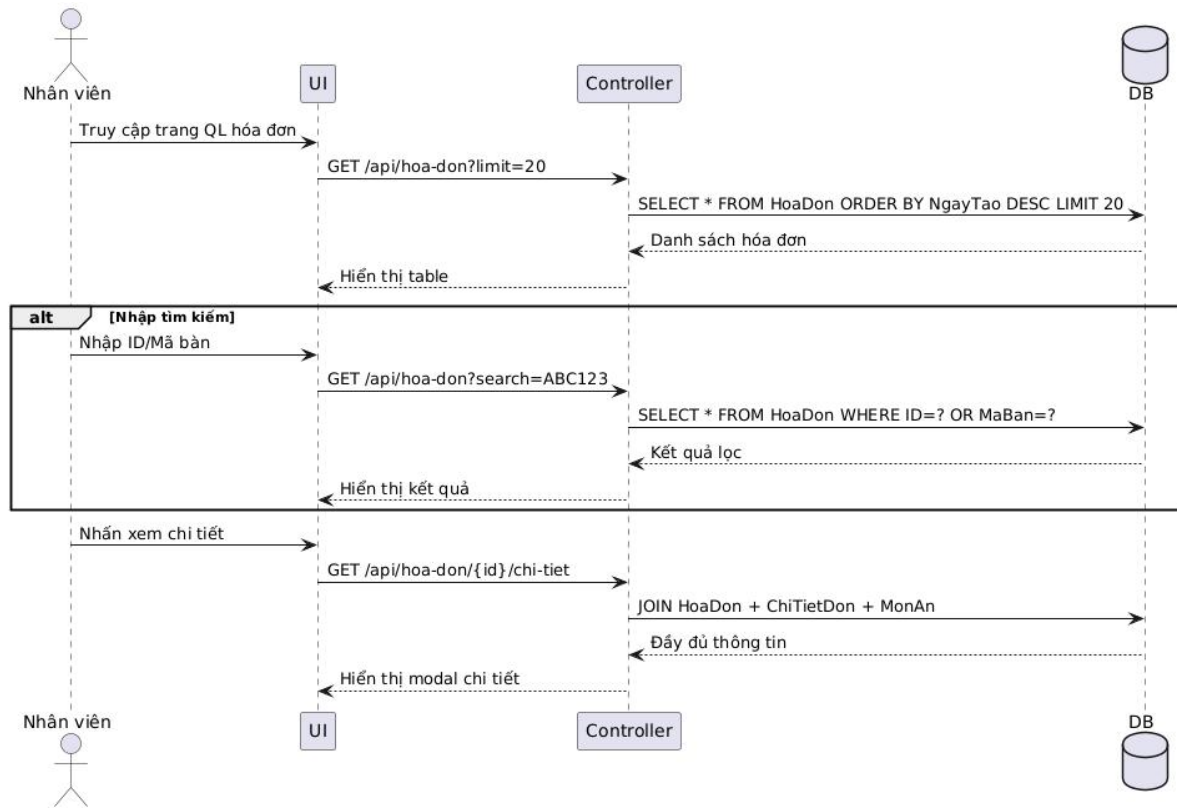


Hình 3-22: sequency Quản lý đơn đặt bàn

## Dòng sự kiện chính

1. Nhân viên truy cập trang quản lý bàn và chọn xem chi tiết một bàn.
2. Hệ thống hiển thị trạng thái hiện tại của bàn (ví dụ: "Đang sử dụng", "Trống").
3. Nhân viên nhấn "Cập nhật trạng thái" (ví dụ: chuyển từ "Trống" sang "Đã dọn").
4. Hệ thống cập nhật trạng thái mới vào cơ sở dữ liệu và hiển thị thay đổi ngay lập tức trên giao diện.

- Usecase “QLHoaDon”



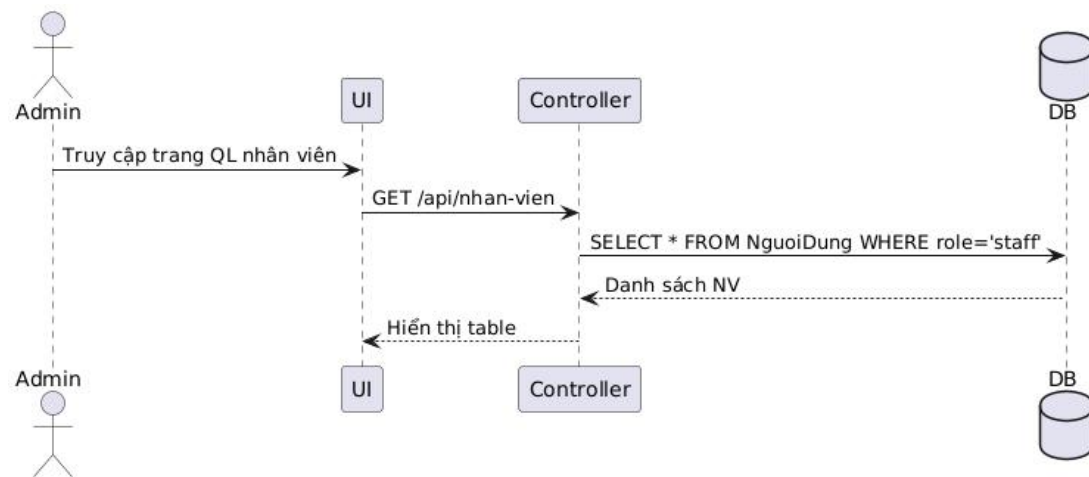
Hình 3-23:sequency Quản lý Hóa Đơn

### Dòng sự kiện chính

- 1.Nhân viên truy cập trang quản lý hóa đơn.
- 2.Hệ thống hiển thị 20 hóa đơn gần nhất theo thời gian tạo.
- 3.Nhân viên có thể:
  - + Tìm kiếm hóa đơn bằng ID hoặc mã bàn.
  - + Xem chi tiết hóa đơn (thông tin đơn hàng, món ăn, giá tiền).

### 3.3.1.3. Admin

- Usecase “QLNhanVien”

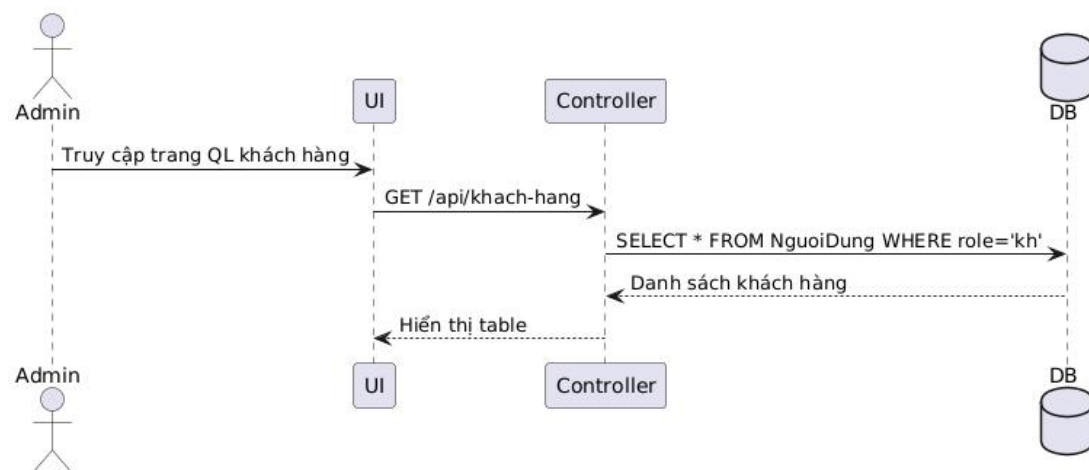


Hình 3-24:sequency Quản lý Nhân viên

### Dòng sự kiện chính

- 1.Admin truy cập trang quản lý nhân viên.
- 2.Hệ thống hiển thị danh sách nhân viên từ cơ sở dữ liệu, lọc theo vai trò "staff".

- Usecase “QLKhachHang”



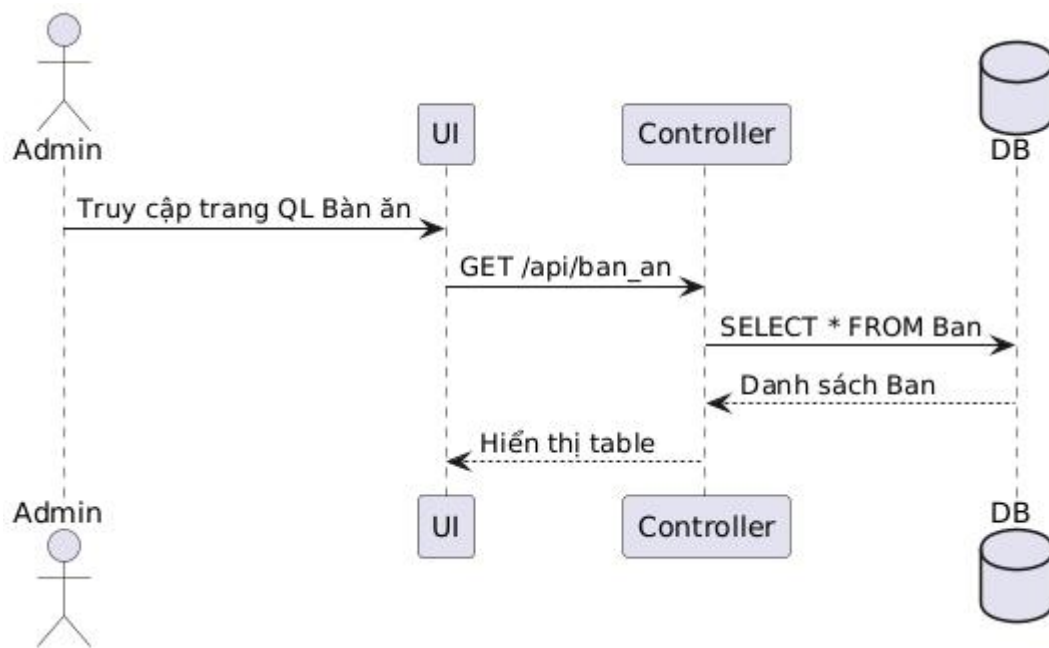
Hình 3-25: sequency Quản lý Khách hàng



## Dòng sự kiện chính

- 1.Admin truy cập trang quản lý khách hàng.
- 2.Hệ thống hiển thị danh sách khách hàng từ cơ sở dữ liệu, lọc theo vai trò "khách hàng".

### • Usecase “QLBanAn”

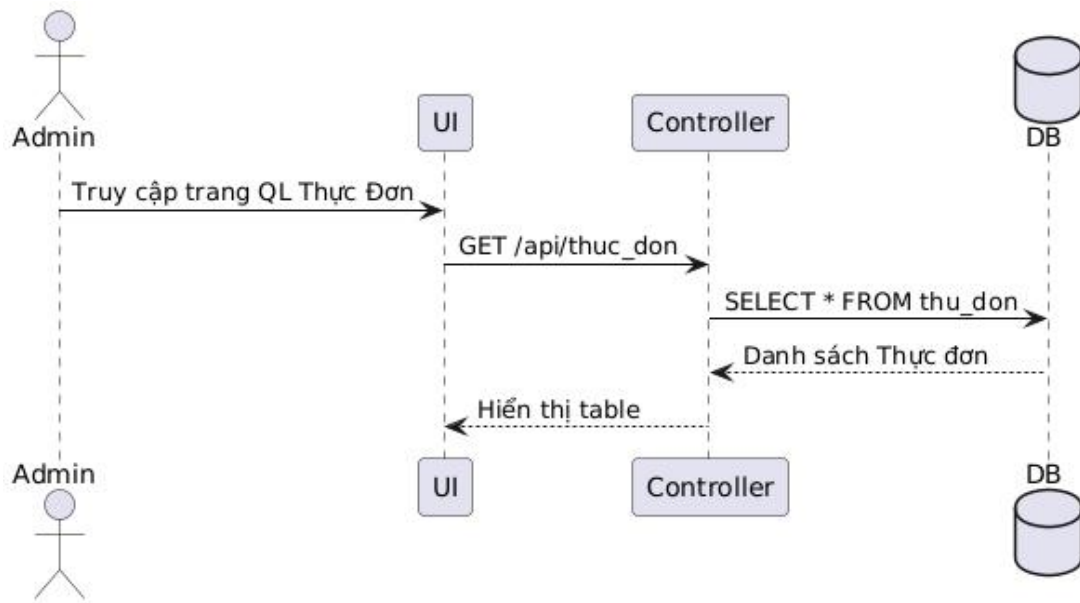


Hình 3-26 :sequency Quản lý bàn ăn

## Dòng sự kiện chính

- 1.Admin truy cập trang quản lý bàn ăn.
- 2.Hệ thống hiển thị toàn bộ thông tin bàn (trạng thái, vị trí, mã bàn).

- Usecase “QLThucDon”

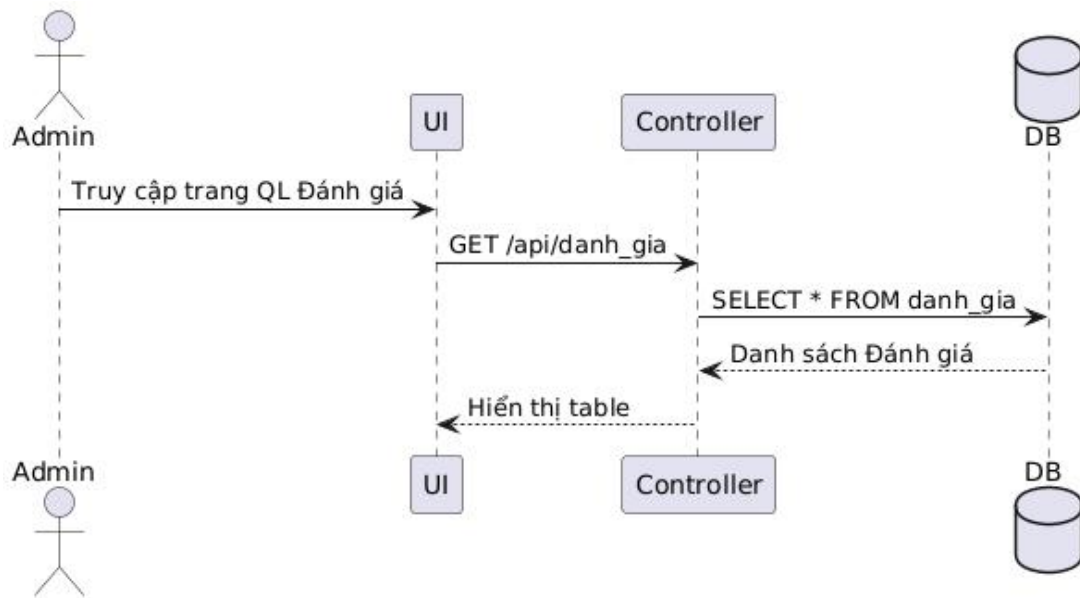


Hình 3-27: sequency Quản lý Thực đơn

### Dòng sự kiện chính

- 1.Admin truy cập trang quản lý thực đơn.
- 2.Hệ thống hiển thị toàn bộ danh sách món ăn từ cơ sở dữ liệu.
- 3.Thông tin được sắp xếp trong bảng, bao gồm các chi tiết như tên món, giá, loại món.

- Usecase “QLDanhGia”



Hình 3-27: sequency Quản lý Đánh Giá

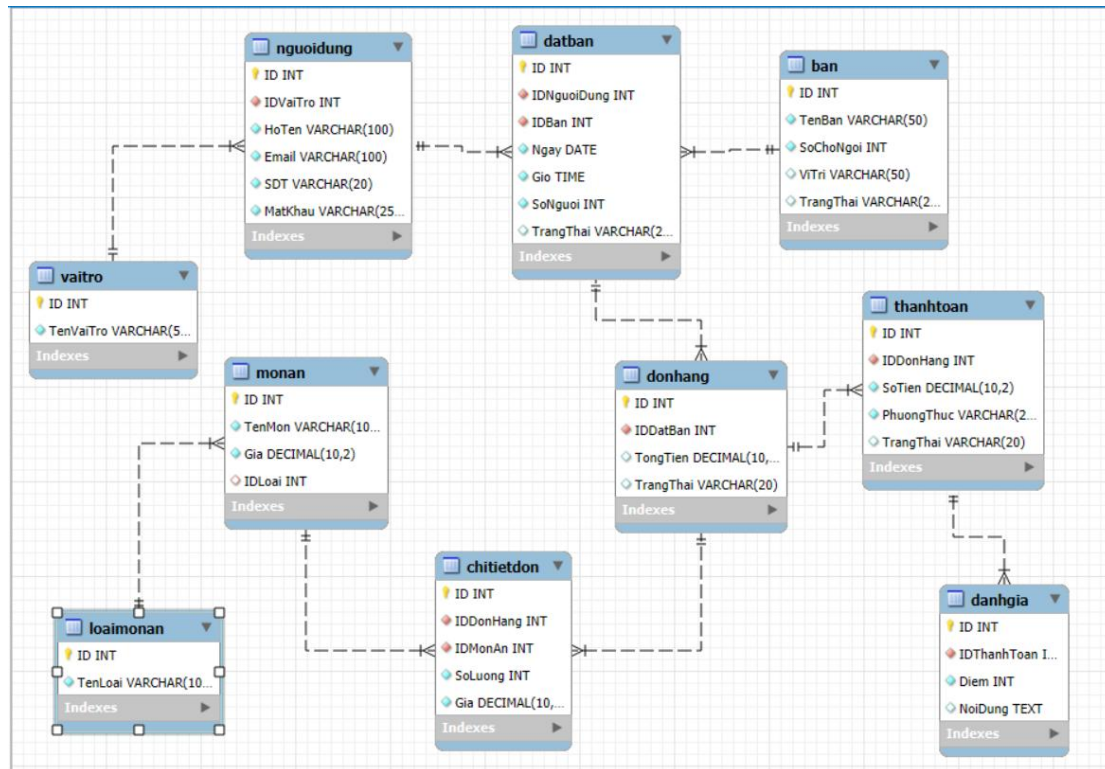
### Dòng sự kiện chính

- 1.Admin truy cập trang quản lý đánh giá.
- 2.Hệ thống hiển thị tất cả đánh giá từ người dùng, bao gồm điểm số, nội dung và thời gian đánh giá.
- 3.Dữ liệu được hiển thị dưới dạng bảng để dễ theo dõi và phân tích.

## CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

### 4.1. Thiết kế cơ sở dữ liệu

#### 4.1.1. Lược đồ quan hệ



Hình 4-1: lược đồ quan hệ

#### 4.1.2. Mô tả dữ liệu

##### 4.1.2.1. Thực thể VaiTro

| Thuộc tính | Kiểu dữ liệu | Mô tả                             |
|------------|--------------|-----------------------------------|
| ID         | Int          | Mã vai trò (khóa chính)           |
| TenVaiTro  | Varchar(50)  | Tên vai trò (Khách, Nhân viên...) |

#### 4.1.2.2. Thực thể *NguoIDung*

| Thuộc tính | Kiểu dữ liệu | Mô tả                             |
|------------|--------------|-----------------------------------|
| ID         | Int          | Mã người dùng (khóa chính)        |
| IDVaiTro   | Int          | Mã vai trò (liên kết bảng VaiTro) |
| HoTen      | Varchar(100) | Họ tên người dùng                 |
| Email      | Varchar(100) | Địa chỉ email, không trùng lặp    |
| SDT        | Varchar(20)  | Số điện thoại                     |
| MatKhau    | Varchar(255) | Mật khẩu đã mã hóa                |

#### 4.1.2.3. Thực thể *Ban*

| Thuộc tính | Kiểu dữ liệu | Mô tả                              |
|------------|--------------|------------------------------------|
| ID         | Int          | Mã bàn ăn (khóa chính)             |
| TenBan     | Varchar(50)  | Tên bàn                            |
| SoChoNgoni | Int          | Số chỗ ngồi                        |
| ViTri      | Varchar(50)  | Vị trí trong nhà hàng              |
| TrangThai  | Varchar(20)  | Trạng thái hiện tại (Trống/Đặt...) |

#### 4.1.2.4. Thực thể *DatBan*

| Thuộc tính  | Kiểu dữ liệu | Mô tả   |
|-------------|--------------|---|
| ID          | Int          | Mã đơn đặt bàn (khóa chính)                   |
| IDNguoIDung | Int          | Mã người đặt bàn (liên kết <i>NguoIDung</i> ) |
| IDBan       | Int          | Mã bàn được đặt (liên kết <i>Ban</i> )        |

|           |             |   |
|-----------|-------------|---|
| Ngày      | Date        | Ngày đặt bàn                              |
| Gio       | Time        | Giờ đặt bàn                               |
| SoNguoi   | Int         | Số người tham gia                         |
| TrangThai | Varchar(20) | Trạng thái đơn (Đang chờ, Đã xác nhận...) |

#### 4.1.2.5. Thực thể *LoaiMonAn*

| Thuộc tính | Kiểu dữ liệu | Mô tả                            |
|------------|--------------|----------------------------------|
| ID         | Int          | Mã loại món ăn                   |
| TenLoai    | Varchar(100) | Tên loại (Khai vị, Món chính...) |

#### 4.1.2.6. Thực thể *MonAn*

| Thuộc tính | Kiểu dữ liệu  | Mô tả                      |
|------------|---------------|----------------------------|
| ID         | Int           | Mã món ăn                  |
| TenMon     | Varchar(100)  | Tên món                    |
| Gia        | Decimal(10,2) | Giá món ăn                 |
| IDLoai     | Int           | Mã loại món ăn (LoaiMonAn) |

#### 4.1.2.7. Thực thể *DonHang*

| Thuộc tính | Kiểu dữ liệu | Mô tả                 |
|------------|--------------|-----------------------|
| ID         | Int          | Mã đơn hàng           |
| IDDatBan   | Int          | Gắn với 1 đơn đặt bàn |

|           |               |  |
|-----------|---------------|--|
| TongTien  | Decimal(10,2) | Tổng số tiền                             |
| TrangThai | Varchar(20)   | Trạng thái đơn (Đang xử lý, Hoàn tất...) |

#### 4.1.2.8. Thực thể ChiTietDon

| Thuộc tính | Kiểu dữ liệu  | Mô tả                 |
|------------|---------------|-----------------------|
| ID         | Int           | Mã chi tiết           |
| IDDonHang  | Int           | Mã đơn hàng (DonHang) |
| IDMonAn    | Int           | Mã món ăn (MonAn)     |
| SoLuong    | Int           | Số lượng món          |
| Gia        | Decimal(10,2) | Giá tại thời điểm đặt |

#### 4.1.2.9. Thực thể ThanhToan

| Thuộc tính | Kiểu dữ liệu  | Mô tả                                |
|------------|---------------|--------------------------------------|
| ID         | Int           | Mã thanh toán                        |
| IDDonHang  | Int           | Mã đơn hàng (DonHang)                |
| SoTien     | Decimal(10,2) | Số tiền thanh toán                   |
| PhuongThuc | Varchar(20)   | Phương thức (Tiền mặt, Momo...)      |
| TrangThai  | Varchar(20)   | Trạng thái (Thành công, Thất bại...) |

#### 4.1.2.10. Thực thể DanhGia

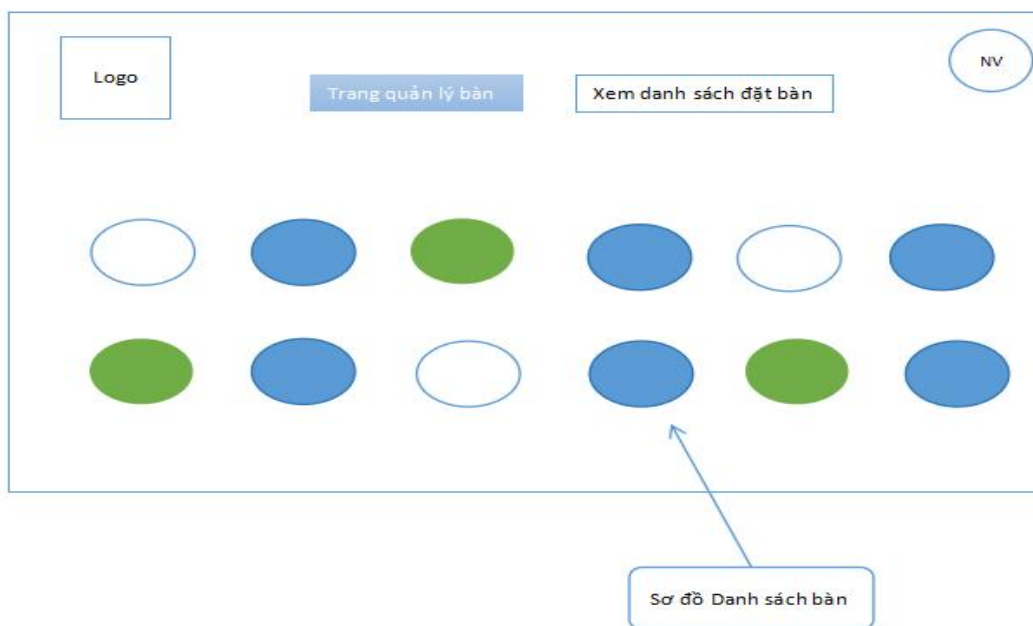
| Thuộc tính | Kiểu dữ liệu | Mô tả |
|------------|--------------|-------|
|------------|--------------|-------|

|             |      |                                  |
|-------------|------|----------------------------------|
| ID          | Int  | Mã đánh giá                      |
| IDThanhToan | Int  | Mã thanh toán đã hoàn tất        |
| Diem        | Int  | Điểm đánh giá (1 đến 5)          |
| NoiDung     | Text | Nội dung đánh giá của khách hàng |

## 4.2. Thiết kế giao diện

### 4.2.1. Giao diện cho nhân viên:

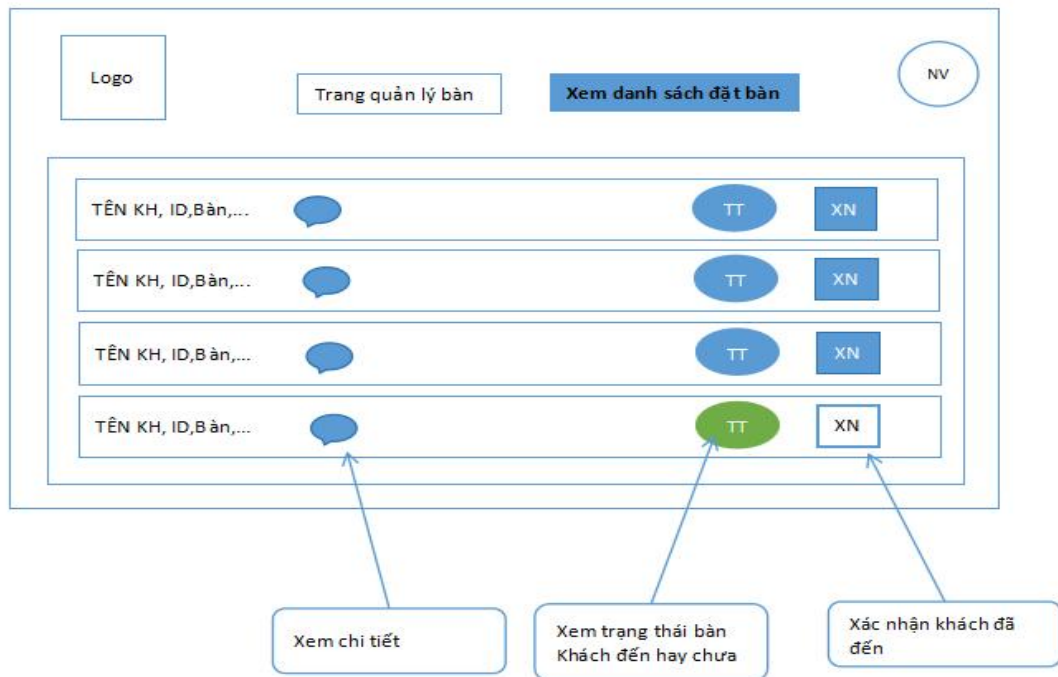
#### 4.2.1.1. Trang quản lý bàn



Hình 4-2: Giao diện quản lý bàn



#### 4.2.1.2. Trang quản lý danh sách đặt bàn



Hình 4-3: Giao diện quản lý danh sách đặt bàn

#### 4.2.2. Giao diện cho Admin:

##### 4.2.2.1. Trang quản lý người dùng



Hình 4-4: Giao diện quản lý người dùng

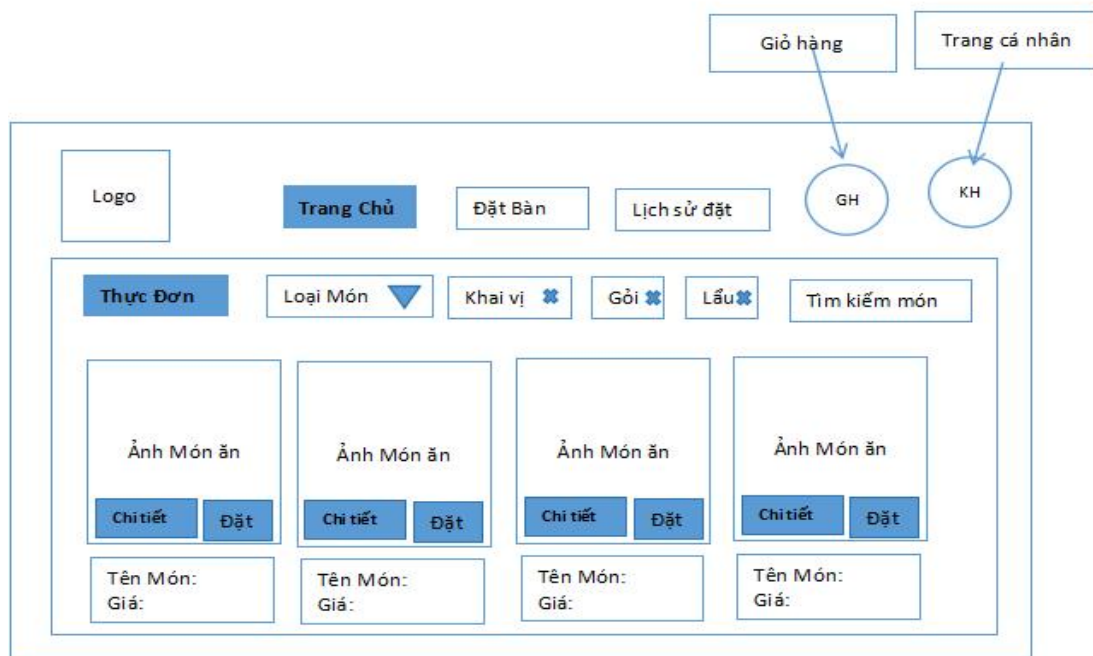
#### 4.2.2.2. Trang quản lý thực đơn



Hình 4-5: Giao diện quản lý thực đơn

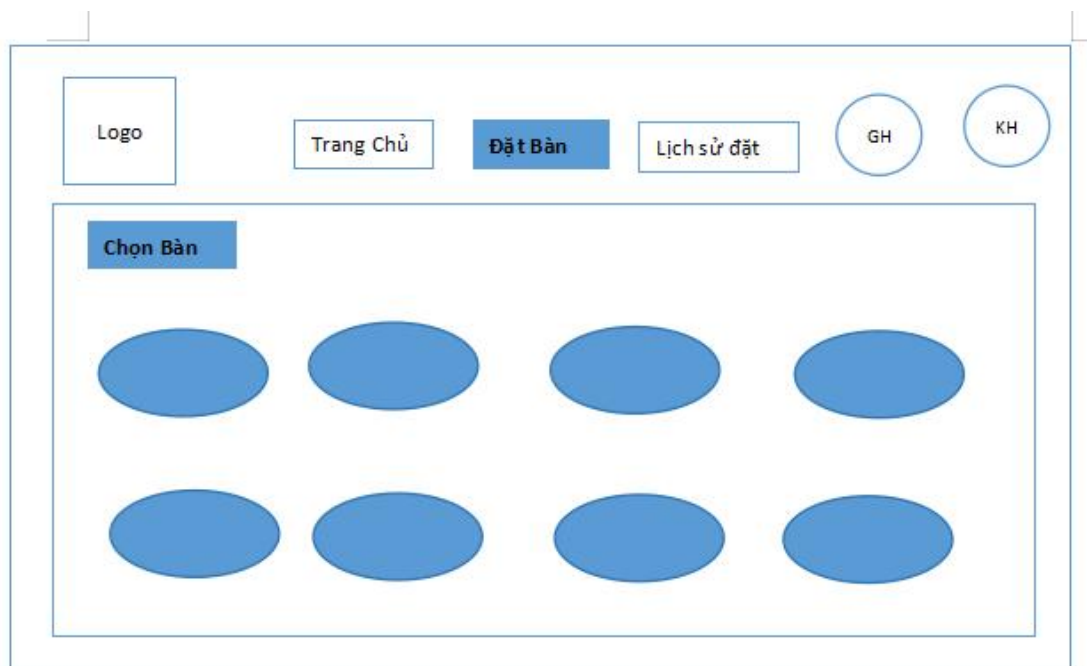
#### 4.2.3. Giao diện khách hàng

##### 4.2.3.1. Trang chủ



Hình 4-6: Giao diện trang chủ

#### 4.2.3.2. Trang đặt bàn



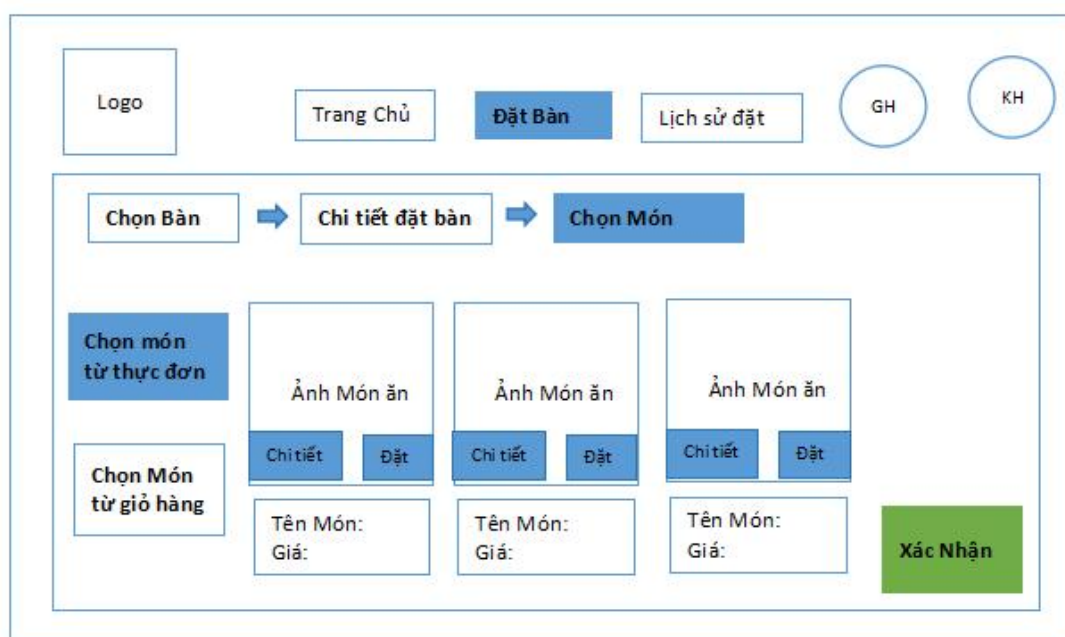
Hình 4-7: Giao diện đặt bàn

#### 4.2.3.3. Trang chi tiết đặt bàn

The screenshot shows a web interface for table booking details. At the top, there is a navigation bar with a 'Logo' box, a 'Trang Chủ' (Home) button, a highlighted 'Đặt Bàn' (Book Table) button, a 'Lịch sử đặt' (Booking History) button, and two circular buttons labeled 'GH' and 'KH'. Below the navigation bar, there is a large container. Inside this container, there is a 'Chọn Bàn' (Choose Table) button followed by a right-pointing arrow and a highlighted 'Chi tiết đặt bàn' (Book Table Details) button. Below these buttons, there are three input fields: 'Số người' (Number of people) with a dropdown arrow, 'Chọn, ngày, giờ' (Choose, date, time) with a dropdown arrow, and 'Ghi chú' (Note). At the bottom right of the container, there is a green button labeled 'Xác nhận đặt bàn' (Confirm booking).

Hình 4-8: Giao diện chi tiết đặt bàn

#### 4.2.3.4. Trang chi tiết đặt món



Hình 4-9: Giao diện chi tiết đặt món

### 4.3. Kiến trúc hệ thống

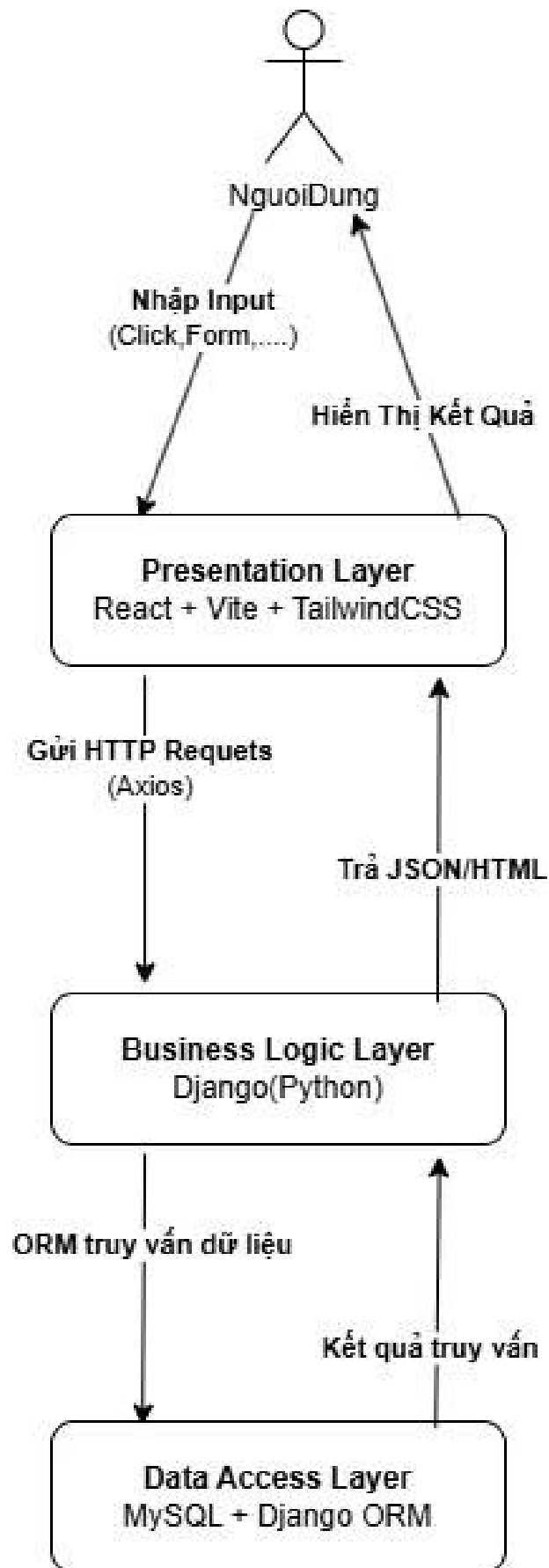
#### Kiến trúc hệ thống 3 lớp

Hệ thống được xây dựng theo mô hình kiến trúc 3 lớp (three-tier architecture), giúp phân chia rõ ràng các chức năng trong ứng dụng web, từ giao diện người dùng đến xử lý nghiệp vụ và truy xuất cơ sở dữ liệu. Mô hình này không chỉ giúp tổ chức mã nguồn một cách logic, dễ bảo trì, mà còn tạo điều kiện thuận lợi cho việc mở rộng hệ thống trong tương lai. Ba lớp chính bao gồm:

**1.Lớp trình bày (Presentation Layer):** Đây là lớp giao tiếp trực tiếp với người dùng, chịu trách nhiệm hiển thị giao diện, nhận và xử lý dữ liệu đầu vào. Trong hệ thống này, frontend được phát triển bằng React, sử dụng Vite để tối ưu hiệu suất build và hot reload trong quá trình phát triển, kết hợp với Tailwind CSS để tạo giao diện nhanh, đẹp và dễ tùy chỉnh. Các thành phần giao diện được chia nhỏ thành các component có thể tái sử dụng, kết hợp với React Router để định tuyến trang và axios/fetch API để giao tiếp với backend.

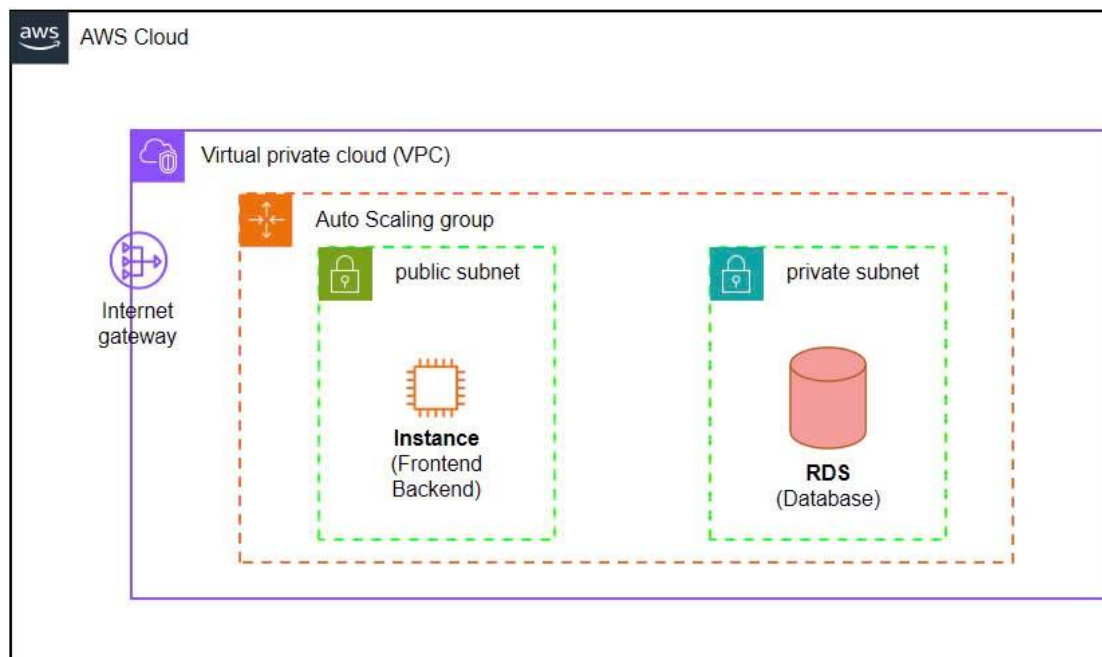
**2.Lớp nghiệp vụ (Business Logic Layer):** Đây là lớp xử lý các logic nghiệp vụ, xác thực người dùng, xử lý dữ liệu và đóng vai trò cầu nối giữa frontend và cơ sở dữ liệu. Hệ thống sử dụng Django – một framework web mạnh mẽ của Python – với kiến trúc MTV (Model–Template–View) để tổ chức mã nguồn rõ ràng. Lớp này tiếp nhận các yêu cầu từ frontend thông qua các URL và views, thực hiện xử lý cần thiết như đăng nhập, đặt bàn, tra cứu thông tin, rồi trả dữ liệu phù hợp về cho giao diện. Django cũng tích hợp sẵn cơ chế quản lý session để xử lý đăng nhập, đăng xuất an toàn và thuận tiện.

**3.Lớp truy xuất dữ liệu (Data Access Layer):** Lớp này đảm nhận vai trò giao tiếp với hệ quản trị cơ sở dữ liệu. Trong hệ thống, cơ sở dữ liệu MySQL được sử dụng để lưu trữ thông tin người dùng, thông tin bàn, lịch đặt chỗ, v.v. Django ORM giúp ánh xạ dữ liệu giữa các model Python và bảng trong cơ sở dữ liệu, cho phép truy vấn dữ liệu một cách trực quan, an toàn và dễ kiểm soát. Ngoài ra, cơ chế migration của Django giúp dễ dàng thay đổi cấu trúc bảng theo từng phiên bản hệ thống mà không ảnh hưởng đến dữ liệu hiện tại.



Hình 4-10: Kiến trúc hệ thống

## CHƯƠNG 5: TRIỂN KHAI TRÊN CLOUD



Hình 5-1: Sơ đồ cloud

### 5.1. Đóng gói và triển khai bằng Docker

#### 5.1.1. Đóng gói bằng Docker

Hệ thống được đóng gói bằng Docker để đảm bảo tính di động, dễ triển khai và dễ quản lý. Cụ thể:

- + Frontend (giao diện người dùng) được đóng gói trong một container riêng, sử dụng React (hoặc Vite nếu có).
- + Backend (xử lý nghiệp vụ) cũng được đóng gói trong một container riêng, sử dụng Django (hoặc Node.js tùy công nghệ).
- + Hai container này được khởi động và quản lý cùng lúc bằng docker-compose trên cùng một máy chủ EC2.

Việc sử dụng Docker giúp chuẩn hóa môi trường phát triển và triển khai, giảm thiểu lỗi do khác biệt môi trường giữa local và production.

#### 5.1.2. Triển khai trên AWS

Toàn bộ hệ thống được triển khai trên nền tảng AWS, sử dụng một EC2 Linux duy nhất nằm trong Auto Scaling Group. Dù có cấu hình Auto Scaling Group, hệ thống không mở rộng theo chiều ngang (nhiều EC2) mà mở rộng theo chiều dọc (nâng cấp cấu hình máy chủ EC2 khi tải tăng).

Cơ sở dữ liệu được tách riêng và triển khai bằng Amazon RDS sử dụng MySQL, đặt trong private subnet để đảm bảo an toàn, chỉ cho phép truy cập từ EC2 nội bộ thông qua Security Group.

Mô hình triển khai này giúp hệ thống:

- + Vận hành đơn giản với chi phí thấp.
- + Dễ nâng cấp khi cần (nâng cấp cấu hình EC2 hoặc mở rộng về sau).
- + Vẫn đảm bảo tính sẵn sàng nhờ Auto Scaling Group có khả năng tự động khởi động lại EC2 nếu gặp lỗi.

Giữ 1 EC2 với cấu hình mạnh, đơn giản hóa hệ thống.

Vẫn đảm bảo khả năng khởi động lại khi lỗi.

## **5.2. Mô hình hệ thống trên AWS**

### **5.2.1. Giới thiệu**

Trong bối cảnh hiện nay, việc triển khai hệ thống web trên nền tảng điện toán đám mây như Amazon Web Services (AWS) giúp đảm bảo tính linh hoạt, khả năng mở rộng và bảo mật. Tài liệu này mô tả chi tiết kiến trúc hệ thống được triển khai trên AWS, trong đó hệ thống sử dụng Auto Scaling Group để quản lý một máy chủ EC2 duy nhất chạy Linux với cấu hình cao.

Tuy có Auto Scaling Group, hệ thống không mở rộng bằng cách tăng số lượng EC2 mà lựa chọn mở rộng theo chiều dọc (vertical scaling): nâng cấp cấu hình máy chủ EC2 (RAM, CPU) khi tải tăng, nhằm giữ đơn giản kiến trúc, giảm chi



phí quản lý nhưng vẫn duy trì tính sẵn sàng và khả năng nâng cấp trong tương lai.

### **5.2.2. Kiến trúc tổng quan**

#### **5.2.2.1 Mô hình ba lớp**

Hệ thống được chia thành ba lớp chính:

- 1. Lớp trình bày (Frontend):** Giao diện người dùng, hiển thị nội dung và tiếp nhận tương tác.
- 2. Lớp nghiệp vụ (Backend):** Xử lý logic ứng dụng, giao tiếp với cơ sở dữ liệu.
- 3. Lớp dữ liệu (Database):** Lưu trữ và truy vấn dữ liệu.

#### **5.2.2.2 Thành phần chính trong sơ đồ**

VPC (Virtual Private Cloud): Môi trường mạng ảo riêng, nơi triển khai toàn bộ hạ tầng.

Public Subnet: Chứa EC2 có thể truy cập từ internet.

Private Subnet: Chứa RDS, không thể truy cập trực tiếp từ internet.

Internet Gateway: Cho phép EC2 trong Public Subnet truy cập internet.

Auto Scaling Group: Bao gồm 1 EC2, cho phép nâng cấp cấu hình EC2 khi cần.

Amazon RDS: Hệ quản trị cơ sở dữ liệu quan hệ (PostgreSQL/MySQL...).

### **5.2.3. EC2 Linux trong Auto Scaling Group**

#### **5.2.3.1 Lý do sử dụng Auto Scaling Group**

Tự động tái khởi động EC2 nếu gặp sự cố.

Đảm bảo tính sẵn sàng (availability): nếu EC2 bị tắt đột ngột, ASG có thể khởi động lại máy khác theo cùng cấu hình.

Tích hợp dễ dàng với Load Balancer (dù hiện tại chỉ có 1 EC2).

Sẵn sàng cho tương lai: có thể mở rộng lên nhiều EC2 khi cần bằng cách thay đổi Min/Max của ASG.

### 5.2.3.2 Cách triển khai EC2

Hệ điều hành: Ubuntu Server 22.04 LTS hoặc Amazon Linux 2.

Ứng dụng: Cài đặt frontend (React/Vite hoặc HTML) và backend (Node.js, Django...).

Triển khai bằng Docker: mỗi thành phần được đóng gói để tách biệt và dễ quản lý.

Reverse proxy: NGINX điều hướng truy cập frontend/backend.

### 5.2.3.3 Cấu hình phần cứng EC2 đề xuất

Loại EC2: `m6i.4xlarge` hoặc `c6a.4xlarge`

vCPU: 16

RAM: 64 GB

Ổ đĩa: EBS SSD gp3 200–500 GB

## 5.2.4. Chi tiết triển khai dịch vụ

### 5.2.4.1 Lưu lượng người dùng

Người dùng truy cập qua trình duyệt → gửi yêu cầu đến Load Balancer.

Load Balancer chuyển tiếp đến EC2 trong Auto Scaling Group.

Backend xử lý và truy vấn cơ sở dữ liệu từ RDS trong Private Subnet.

### 5.2.4.2 Quản lý bảo mật

Security Group:

EC2: Cho phép cổng 80, 443 (HTTP/HTTPS) từ internet; SSH từ IP quản trị.

RDS: Chỉ nhận kết nối từ EC2 Security Group.

IAM Role: Gán quyền cho EC2 truy cập S3, CloudWatch, hoặc các dịch vụ khác.

Network ACLs & VPC flow logs: Giám sát lưu lượng và chặn truy cập trái phép.

### **5.2.5. Lý do chọn mở rộng theo chiều dọc (Vertical Scaling)**

#### **5.2.5.1 Đơn giản và tối ưu**

Không cần xử lý logic phân phối tải (load balancing logic phức tạp).

Không lo vấn đề đồng bộ phiên (session state, cache đồng nhất).

Dễ phát triển và bảo trì trong giai đoạn đầu.

#### **5.2.5.2 Tối ưu hiệu suất**

Với cấu hình mạnh, EC2 có thể xử lý hàng ngàn kết nối đồng thời.

Có thể giám sát và nâng cấp cấu hình dễ dàng qua bảng điều khiển AWS hoặc CLI.

#### **5.2.5.3 Chi phí linh hoạt**

Tính toán và dự trù chi phí dễ hơn.

Chỉ trả tiền cho 1 EC2 hoạt động liên tục.

Vẫn đảm bảo mức độ sẵn sàng nhờ Auto Scaling tự động khởi động lại khi lỗi.

### **5.2.6. Lớp dữ liệu: Amazon RDS**

#### **5.2.6.1 Triển khai trong private subnet**

Tăng cường bảo mật, tránh bị tấn công trực tiếp từ internet.

Giao tiếp chỉ thông qua EC2 (qua security group nội bộ).

#### **5.2.6.2 Thông số RDS đề xuất**

Engine: PostgreSQL hoặc MySQL

Loại instance: `db.t3.medium`

Tính năng: Sao lưu tự động, Multi-AZ (tùy chọn), mã hóa dữ liệu

### **5.2.7. Giám sát và cảnh báo**

Amazon CloudWatch:

Giám sát CPU, Memory (qua agent), Disk, Network của EC2.

Tạo Alarm: cảnh báo nếu CPU > 80% → xem xét nâng cấp EC2.

Auto Recovery: Nếu EC2 lỗi vật lý, ASG tự thay thế instance khác cùng cấu hình.

### **5.2.8. Khả năng nâng cấp và mở rộng**

#### **5.2.8.1 Trong giai đoạn đầu**

Giữ 1 EC2 với cấu hình mạnh, đơn giản hóa hệ thống.

Vẫn đảm bảo khả năng khởi động lại khi lỗi.

#### **5.2.8.2 Trong tương lai**

Khi lưu lượng tăng cao không thể mở rộng thêm theo chiều dọc:

Tăng Max EC2 trong Auto Scaling Group.

Thêm Load Balancer phân phối tải nhiều EC2.

Tách frontend/backend sang các instance khác nhau hoặc container service (ECS, EKS).

### **5.3. Kết luận**

Việc triển khai hệ thống bằng cách sử dụng Auto Scaling Group với một EC2 Linux duy nhất, có cấu hình cao, là một giải pháp tối ưu cho các hệ thống quy mô nhỏ đến trung bình, cần:

Đảm bảo sẵn sàng khi gặp sự cố (ASG tự thay thế).

Giữ chi phí và vận hành đơn giản.

Dễ dàng nâng cấp theo cả chiều dọc và ngang trong tương lai.

# CHƯƠNG 6: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

## 6.1. KẾT QUẢ ĐẠT ĐƯỢC

### 6.1.1. Kết quả đạt được

Sau thời gian nghiên cứu và phát triển, đề tài "Xây dựng website đặt bàn ăn tại một nhà hàng" đã đạt được các kết quả như sau:

- ❖ Xây dựng hoàn chỉnh hệ thống website đặt bàn ăn với kiến trúc tách biệt frontend và backend, cho phép khách hàng dễ dàng đặt bàn theo thời gian, vị trí và số người mong muốn.
- ❖ Phát triển đầy đủ các chức năng quản lý:
- ❖ Quản lý bàn ăn, khu vực, thực đơn, đơn đặt bàn, đơn hàng, hóa đơn và đánh giá.
- ❖ Các vai trò (admin, nhân viên, khách hàng) được phân quyền rõ ràng, tương ứng với chức năng cụ thể.
- ❖ Giao diện người dùng thân thiện, thiết kế responsive sử dụng TailwindCSS, tối ưu cho cả desktop và thiết bị di động.
- ❖ Tích hợp RESTful API giữa frontend (React) và backend (Django), đảm bảo tốc độ xử lý nhanh, hiệu quả và dễ mở rộng.
- ❖ Cơ sở dữ liệu được thiết kế chặt chẽ, mô hình hóa hợp lý, dễ bảo trì và nâng cấp.
- ❖ Xử lý nghiệp vụ đặt bàn hiệu quả: tránh trùng bàn, kiểm tra thời gian đặt, giải phóng bàn nếu hủy.
- ❖ Thực đơn hiển thị rõ ràng, dễ lựa chọn, hỗ trợ gọi món trước hoặc thêm món trong quá trình sử dụng bàn.
- ❖ Đảm bảo an toàn thông tin và xác thực người dùng qua chức năng đăng ký, đăng nhập, đăng xuất có kiểm tra hợp lệ.

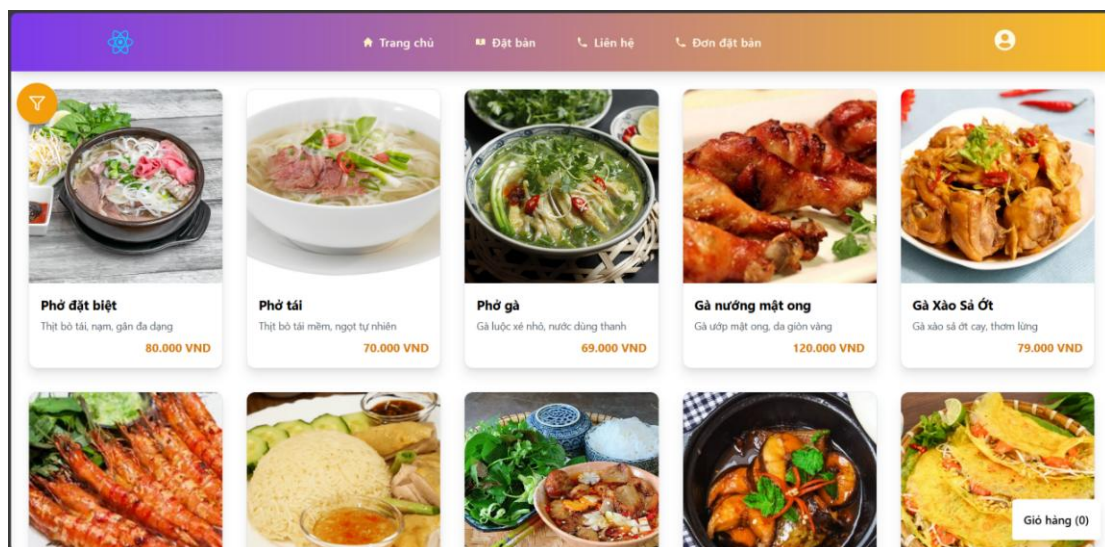
### 6.1.2 HẠN CHẾ

Mặc dù hệ thống đã hoàn thiện các chức năng cơ bản, nhưng do giới hạn về thời gian và kinh nghiệm thực tiễn, đề tài vẫn còn một số điểm hạn chế:

- ❖ Chưa tích hợp thanh toán trực tuyến, người dùng hiện chỉ có thể đặt bàn và gọi món, chưa thực hiện thanh toán qua VNPay, Momo, v.v.
- ❖ Chưa có chức năng gợi ý món ăn theo sở thích người dùng hoặc lịch sử đơn hàng (thiếu tính năng cá nhân hóa).
- ❖ Chưa triển khai hệ thống thông báo thời gian thực (real-time) để cập nhật ngay trạng thái đặt bàn hoặc gọi món (ví dụ dùng WebSocket).
- ❖ Chưa hỗ trợ trang admin đa ngôn ngữ, hiện tại hệ thống chỉ có giao diện tiếng Việt.
- ❖ Thiếu các công cụ thống kê báo cáo như số lượt đặt bàn theo ngày, doanh thu từng khu vực, hiệu suất phục vụ theo nhân viên,...
- ❖ Hệ thống mới dừng ở mức thử nghiệm với dữ liệu mẫu, chưa được triển khai thực tế tại nhà hàng.

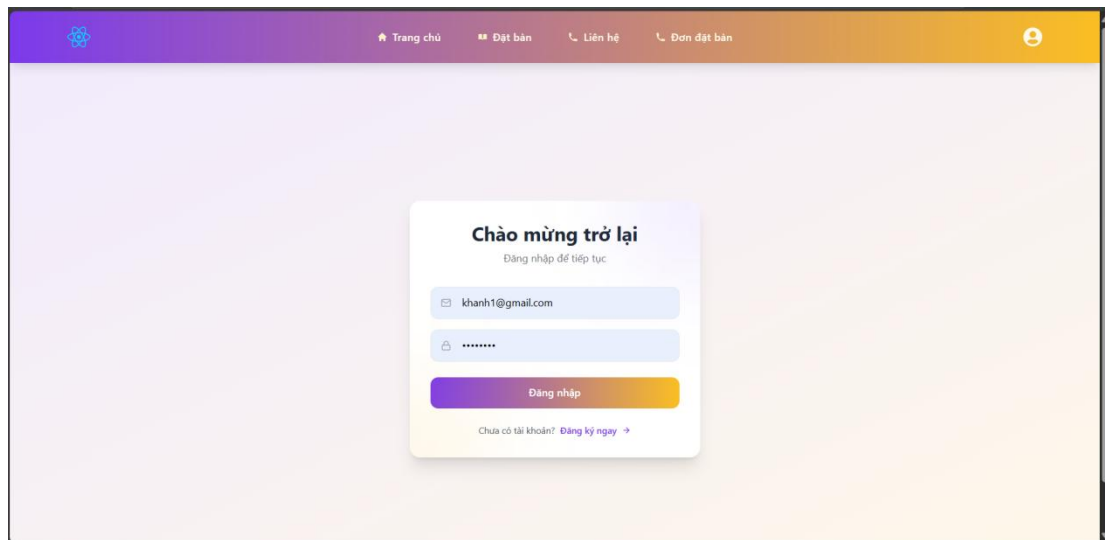
## 6.2. Hình ảnh minh họa hệ thống

- ❖ Giao diện chính của website, nơi người dùng có thể xem, tìm kiếm thông tin chi tiết các món ăn và giá cả

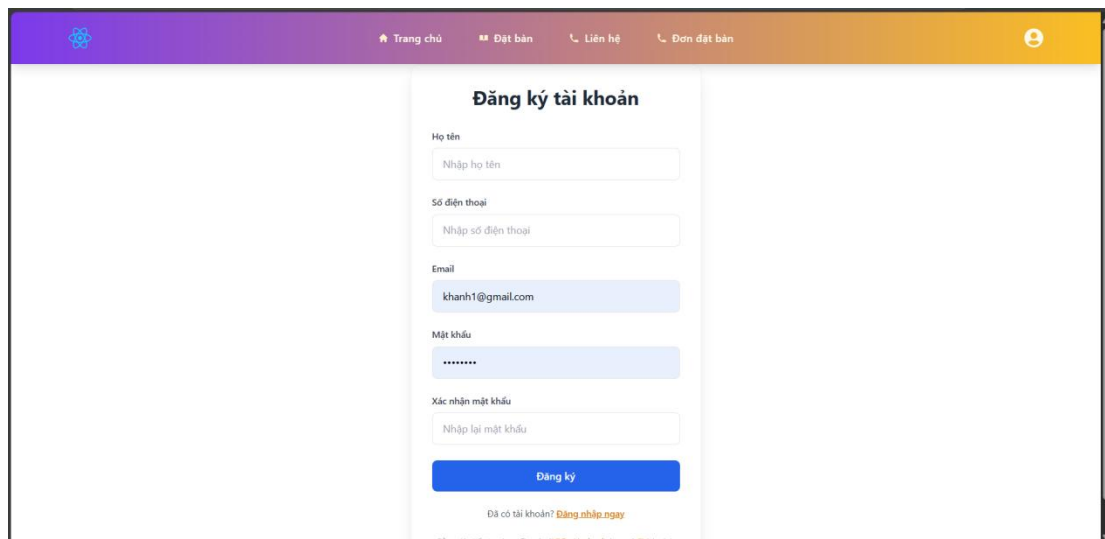


Hình 6-1: Giao diện trang chủ

- ❖ Cho phép khách hàng tạo tài khoản hoặc đăng nhập để sử dụng các chức năng như đặt bàn, xem lịch sử



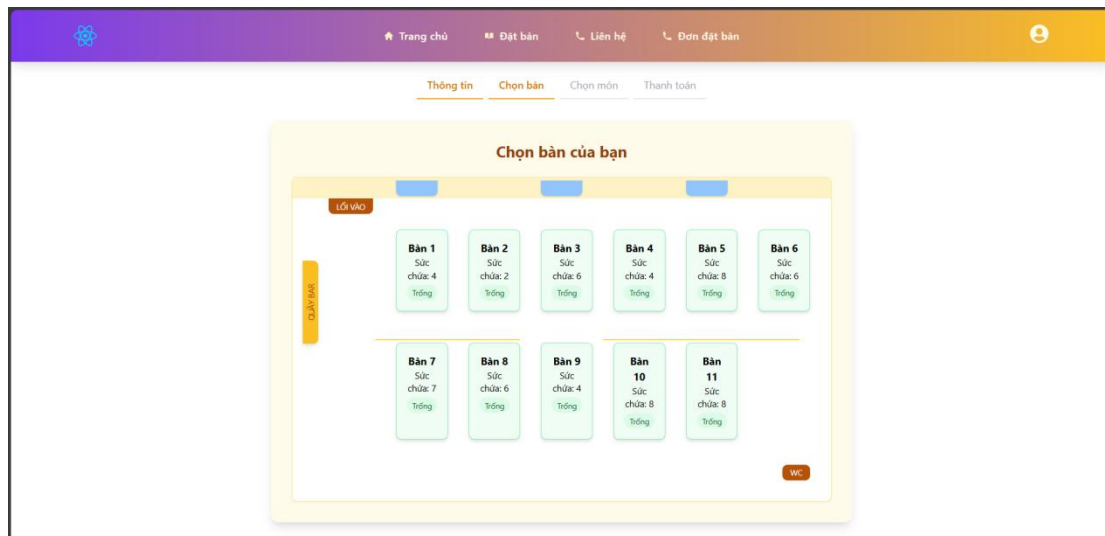
*Hình 6-2: Giao diện đăng nhập*



*Hình 6-3: Giao diện đăng ký*

- ❖ Cho phép người dùng chọn ngày, giờ và các bàn còn trống để đặt





Hình 6-4: Giao diện chọn bàn

❖ Khách hàng có thể xem lại lịch sử đặt bàn

| Lịch sử đơn hàng |            |          |       |             |             |                     |
|------------------|------------|----------|-------|-------------|-------------|---------------------|
| Mã đơn           | Ngày đặt   | Giờ đặt  | Bàn   | Trạng thái  | Tổng tiền   | Hành động           |
| 17               | 2025-05-12 | 18:00:00 | Bàn 1 | ĐÃ XÁC NHẬN | 210.000 VND | <a href="#">Xem</a> |
| 18               | 2025-05-12 | 18:00:00 | Bàn 3 | ĐÃ XÁC NHẬN | 490.000 VND | <a href="#">Xem</a> |
| 19               | 2025-05-12 | 19:00:00 | Bàn 8 | ĐÃ XÁC NHẬN | 230.000 VND | <a href="#">Xem</a> |
| 20               | 2025-05-12 | 19:00:00 | Bàn 2 | ĐÃ XÁC NHẬN | 375.000 VND | <a href="#">Xem</a> |
| 21               | 2025-05-13 | 18:00:00 | Bàn 3 | ĐÃ XÁC NHẬN | 220.000 VND | <a href="#">Xem</a> |
| 22               | 2025-05-13 | 18:00:00 | Bàn 8 | ĐÃ XÁC NHẬN | 650.000 VND | <a href="#">Xem</a> |
| 23               | 2025-05-13 | 18:00:00 | Bàn 5 | ĐÃ XÁC NHẬN | 265.000 VND | <a href="#">Xem</a> |
| 24               | 2025-05-13 | 19:00:00 | Bàn 5 | ĐÃ XÁC NHẬN | 145.000 VND | <a href="#">Xem</a> |
| 25               | 2025-05-13 | 13:00:00 | Bàn 3 | ĐÃ XÁC NHẬN | 315.000 VND | <a href="#">Xem</a> |
| 26               | 2025-05-13 | 12:00:00 | Bàn 4 | ĐÃ XÁC NHẬN | 515.000 VND | <a href="#">Xem</a> |
| 27               | 2025-05-13 | 19:00:00 | Bàn 3 | ĐÃ XÁC NHẬN | 170.000 VND | <a href="#">Xem</a> |
| 28               | 2025-05-14 | 13:00:00 | Bàn 3 | ĐÃ XÁC NHẬN | 610.000 VND | <a href="#">Xem</a> |

Hình 6-5: Giao diện lịch sử đặt bàn

### 6.3. Hướng phát triển trong tương lai

Để hệ thống trở nên hoàn thiện hơn và đáp ứng tốt hơn nhu cầu thực tế tại các nhà hàng, trong tương lai nhóm đề tài định hướng mở rộng và phát triển thêm các chức năng sau:

❖ Tích hợp thanh toán trực tuyến:

- Kết nối với các cổng thanh toán như Momo, VNPay, ZaloPay,... giúp khách hàng thanh toán dễ dàng khi đặt bàn và gọi món trực tuyến.

- ❖ Phát triển ứng dụng mobile (iOS/Android):
  - Tái sử dụng backend hiện có, xây dựng ứng dụng di động giúp người dùng đặt bàn, gọi món tiện lợi hơn khi di chuyển.
- ❖ Thêm hệ thống báo cáo và thống kê trực quan:
  - Cung cấp biểu đồ doanh thu theo ngày, tháng, phân tích món ăn được ưa chuộng, thống kê hiệu suất làm việc của nhân viên.
- ❖ Tích hợp chatbot hỗ trợ tư vấn khách hàng 24/7:
  - Hướng đến trải nghiệm tương tác tự động, giải đáp nhanh các thắc mắc của khách về thực đơn, giờ hoạt động, chương trình khuyến mãi.
- ❖ Ứng dụng trí tuệ nhân tạo (AI) để:

Gợi ý món ăn phù hợp dựa trên lịch sử đặt món hoặc khẩu vị.

Dự đoán lượng khách để tối ưu hóa số lượng bàn ăn và nhân sự phục vụ.

- ❖ Hỗ trợ đa ngôn ngữ:
  - Phát triển giao diện với nhiều ngôn ngữ (Anh, Nhật, Hàn...) để phục vụ cả khách quốc tế.
- ❖ Nâng cao bảo mật hệ thống:
  - Tích hợp xác thực 2 lớp (Two-factor Authentication) cho người dùng và quản trị.
  - Mã hóa dữ liệu nhạy cảm và kiểm tra an toàn định kỳ.
- ❖ Tích hợp chức năng đặt tiệc, tổ chức sự kiện:
  - Cho phép khách hàng đặt nhiều bàn/lầu/khu vực và lên thực đơn trước cho tiệc sinh nhật, lễ kỷ niệm,...

## TÀI LIỆU THAM KHẢO

Django Documentation. (n.d.). The web framework for perfectionists with deadlines. Truy cập từ: <https://docs.djangoproject.com/>

React Documentation. (n.d.). A JavaScript library for building user interfaces. Truy cập từ: <https://react.dev/>

TailwindCSS Documentation. (n.d.). Utility-first CSS framework. Truy cập từ: <https://tailwindcss.com/>

Vite Documentation. (n.d.). Next Generation Frontend Tooling. Truy cập từ: <https://vitejs.dev/>

MySQL Documentation. (n.d.). MySQL 8.0 Reference Manual. Truy cập từ: <https://dev.mysql.com/doc/>

Amazon Web Services (AWS). (n.d.). Deploying a web application using EC2 and RDS. Truy cập từ: <https://aws.amazon.com/>

REST API Design Guidelines – Best Practices. (n.d.). Truy cập từ: <https://restfulapi.net/>