

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

---oOo---



**TIỂU LUẬN MÔN XÂY DỰNG HTTT TRÊN**  
**CÁC FRAMEWORK**

*Tên tiểu luận:*

**HỆ THỐNG ĐẶT BÀN ĂN TRỰC TUYẾN**

Người hướng dẫn: **Dương Quang Sinh**

Sinh viên thực hiện:

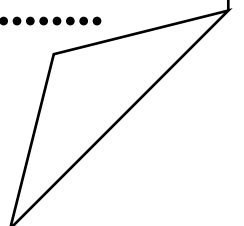
**Phạm Thanh Phong - 22050001**

**Nguyễn Văn Khánh - 22050079**

**Võ Anh Hiệp - 22050104**

**Thành phố Hồ Chí Minh, tháng 12 năm 2025**

## This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings present.



# LỜI MỞ ĐẦU

Trong bối cảnh công nghệ thông tin ngày càng phát triển mạnh mẽ, việc ứng dụng các hệ thống web vào lĩnh vực dịch vụ, đặc biệt là lĩnh vực nhà hàng – ẩm thực, đã trở thành xu hướng tất yếu. Nhu cầu của khách hàng không chỉ dừng lại ở việc tìm kiếm thông tin nhà hàng mà còn mong muốn có thể đặt bàn nhanh chóng, tiện lợi và chính xác thông qua các nền tảng trực tuyến. Đồng thời, các nhà hàng cũng cần một công cụ hiệu quả để quản lý thông tin, tiếp nhận và xử lý yêu cầu đặt bàn một cách khoa học.

Xuất phát từ thực tế đó, đề tài “Xây dựng hệ thống Web đặt bàn ăn” được lựa chọn nhằm nghiên cứu và triển khai một hệ thống trung gian kết nối giữa khách hàng, chủ nhà hàng (partner) và quản trị viên (admin). Hệ thống cho phép khách hàng tìm kiếm nhà hàng, xem thông tin chi tiết, lựa chọn khung giờ và thực hiện đặt bàn; đồng thời hỗ trợ nhà hàng quản lý thông tin, xác nhận hoặc từ chối các đơn đặt bàn, cũng như giúp admin quản lý và giám sát toàn bộ hoạt động của hệ thống.

Trong tiểu luận này, nhóm tập trung phân tích yêu cầu nghiệp vụ, thiết kế hệ thống, xây dựng cơ sở dữ liệu, mô hình hóa các luồng hoạt động và triển khai các chức năng chính của website đặt bàn ăn dựa trên công nghệ Django và Django REST Framework. Thông qua quá trình thực hiện, đề tài nhằm làm rõ cách một hệ thống web có thể giải quyết bài toán đặt bàn trong thực tế, nâng cao trải nghiệm người dùng và góp phần tối ưu hóa quy trình quản lý cho các nhà hàng trong môi trường số hóa hiện nay.

## LỜI CẢM ƠN

Trong quá trình thực hiện đề tài và hoàn thành báo cáo này, em đã nhận được nhiều sự hỗ trợ và động viên quý báu từ thầy cô, gia đình và bạn bè. Em xin bày tỏ lòng biết ơn sâu sắc đến thầy **Dương Quang Sinh** — người đã tận tình hướng dẫn, góp ý chi tiết và định hướng cho em trong suốt quá trình triển khai đề tài. Nhờ sự chỉ dẫn của thầy, em không chỉ hiểu rõ hơn về cách tiếp cận bài toán, cách tổ chức nội dung báo cáo mà còn học được thêm nhiều kinh nghiệm quan trọng trong việc nghiên cứu và xây dựng một hệ thống theo đúng quy trình.

Em cũng xin chân thành cảm ơn gia đình đã luôn quan tâm, tạo điều kiện về thời gian, tinh thần và môi trường học tập để em có thể tập trung thực hiện đề tài. Đồng thời, em xin cảm ơn bạn bè và các anh/chị đã sẵn sàng chia sẻ tài liệu, trao đổi ý tưởng, góp ý trong những lúc em gặp khó khăn, giúp em có thêm động lực để hoàn thành công việc đúng tiến độ.

Mặc dù đã cố gắng tìm hiểu và thực hiện đề tài một cách nghiêm túc, do hạn chế về thời gian, kinh nghiệm thực tế và kiến thức chuyên môn, báo cáo này chắc chắn vẫn còn những thiếu sót nhất định về nội dung cũng như cách trình bày. Em rất mong nhận được các ý kiến góp ý và nhận xét từ thầy để em có thể rút kinh nghiệm, cải thiện và hoàn thiện hơn trong các nghiên cứu và dự án sau này.

Em xin chân thành cảm ơn.

# MỤC LỤC

LỜI CẢM ƠN .....	2
GIỚI THIỆU HỆ THỐNG .....	6
Lý do chọn đề tài .....	6
Mô tả tổng quát về hệ thống .....	6
Hướng giải quyết dự kiến .....	7
CHƯƠNG 1: KẾ HOẠCH THỰC HIỆN .....	9
1.1 Mô hình phát triển .....	9
1.2 Giai đoạn Phân tích và Thu thập yêu cầu .....	9
1.3 Giai đoạn Thiết kế hệ thống .....	10
1.4 Giai đoạn Triển khai và Lập trình .....	12
1.5 Giai đoạn kiểm thử và tích hợp .....	14
1.6 Giai đoạn hoàn thiện và đóng gói đồ án .....	14
CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG .....	15
2.1. Khảo sát hiện trạng .....	15
2.2. Yêu cầu chức năng và phi chức năng .....	15
2.2.1. Yêu cầu chức năng .....	15
2.2.2. Yêu cầu phi chức năng .....	16
2.3. Các đối tượng và chức năng hệ thống .....	17
2.3.1. Khách hàng .....	17
2.3.2. Đối tác nhà hàng (Partner) .....	17
2.3.3. Quản trị viên (Admin) .....	17
Chương 3: THIẾT KẾ HỆ THỐNG .....	18
3.1. Sơ đồ BFD .....	18
3.2. Use case .....	20
3.3. Mô hình UML .....	23
Hình 3: Sơ đồ UML .....	23
Mô tả .....	23

1. Phân tích chi tiết các nhóm lớp .....	23
2. Các mối quan hệ quan trọng (Relationships) .....	26
3.4. Mô hình động .....	27
3.4.1. Sơ đồ Sequence .....	27
3.4.2. Sơ đồ Activity .....	28
Chương 4: THIẾT KẾ DỮ LIỆU VÀ DÒNG DỮ LIỆU .....	31
4.1 Thiết kế ERD .....	31
4.2 Chuyển sang mô hình quan hệ .....	33
4.3 Thiết kế CSDL .....	34
4.4 Sơ đồ DFD .....	37
Hình 7: Sơ đồ DFD level 0 .....	37
1. Tổng quan .....	37
2. Các thành phần chính .....	37
3. Chi tiết các luồng dữ liệu (Data Flows) .....	38
CHƯƠNG 5: TRIỂN KHAI VÀ KIỂM THỬ .....	39
5.1 Kiến trúc ứng dụng .....	39
5.2 Công nghệ sử dụng .....	39
5.3 Xử lý sự kiện và tương tác .....	40
5.4 Giao diện người dùng .....	40
5.4.1. Phân hệ Khách hàng (Customer Side) .....	40
Hình 9: giao diện trang chủ .....	41
Hình 10: giao diện quy trình đặt bàn .....	41
Hình 11: Quản lý Đặt bàn cá nhân .....	42
5.4.2. Phân hệ Đối tác (Partner Dashboard) .....	42
Hình 12: Quản lý Nhà hàng và Thông báo .....	43
Hình 13: Quản lý Danh sách Booking .....	43
5.4.3. Phân hệ Quản trị viên (Admin Portal) .....	44
Hình 14: Trang quản trị hệ thống .....	44
5.5 Kiểm thử và đánh giá .....	44

5.5.1. Kết quả kiểm thử chức năng (Functional Testing) .....	44
5.5.2. Đánh giá Giao diện và Trải nghiệm (UI/UX Evaluation) .....	45
5.5.3. Kết luận chung .....	45
CHƯƠNG 6: TỔNG KẾT .....	46
6.1 Kết quả đạt được .....	46
6.2 Hạn chế và hướng phát triển .....	47
TÀI LIỆU THAM KHẢO .....	49

# GIỚI THIỆU HỆ THỐNG

## Lý do chọn đề tài

- ❖ Trong bối cảnh ngành dịch vụ ăn uống phát triển mạnh mẽ tại Việt Nam, việc quản lý đặt bàn vẫn còn nhiều bất cập. Phần lớn các nhà hàng vừa và nhỏ vẫn tiếp nhận đặt bàn qua điện thoại hoặc trực tiếp, dẫn đến tình trạng ghi nhận sai sót, quên lịch hẹn, hoặc xung đột thời gian giữa các đơn đặt bàn. Đối với khách hàng, việc tìm kiếm nhà hàng phù hợp và đặt bàn trước vẫn gặp nhiều khó khăn khi phải gọi điện từng nơi để hỏi thông tin và xác nhận chỗ trống.
- ❖ Từ thực tế đó, việc xây dựng một nền tảng trung gian kết nối khách hàng và nhà hàng trở nên cấp thiết. Hệ thống không chỉ giúp khách hàng dễ dàng tìm kiếm, so sánh và đặt bàn trực tuyến, mà còn hỗ trợ các chủ nhà hàng quản lý đơn đặt bàn một cách khoa học, giảm thiểu sai sót và tối ưu hóa công suất phục vụ. Đây chính là động lực chính thúc đẩy việc lựa chọn đề tài "Hệ thống đặt bàn ẩm thực trực tuyến" cho đề án tốt nghiệp này.

## Mô tả tổng quát về hệ thống

- ❖ Hệ thống đặt bàn ẩm thực trực tuyến là một nền tảng web hoạt động theo mô hình ba bên gồm Khách hàng (Customer), Đối tác nhà hàng (Partner) và Quản trị viên (Admin). Mỗi vai trò trong hệ thống có những chức năng và quyền hạn riêng biệt, phục vụ cho mục đích sử dụng cụ thể.
- ❖ Về phía Khách hàng, hệ thống cung cấp giao diện thân thiện để tìm kiếm nhà hàng theo vị trí địa lý, xem thông tin chi tiết bao gồm thực đơn, hình ảnh, giờ mở cửa và đánh giá. Khách hàng có thể chọn ngày, khung giờ phù hợp và số lượng người để đặt bàn trực tuyến. Sau khi đặt bàn thành công, hệ thống sẽ gửi thông báo về trạng thái đơn đặt, từ chờ xác nhận đến được chấp nhận hoặc từ chối, giúp khách hàng chủ động trong việc sắp xếp lịch trình.
- ❖ Về phía Đối tác nhà hàng, sau khi đăng ký và được quản trị viên phê duyệt, chủ nhà hàng có thể tạo và quản lý thông tin nhà hàng của mình trên nền tảng. Họ được phép thiết lập thực đơn với đầy đủ mô tả và giá cả, upload hình ảnh nhà hàng, cũng như cấu hình các khung giờ đặt bàn với số lượng chỗ tối đa cho mỗi khung giờ. Khi có đơn đặt bàn mới, đối tác sẽ nhận được thông báo và có thể



xác nhận hoặc từ chối dựa trên tình hình thực tế. Sau khi khách đến ăn, đối tác có thể cập nhật trạng thái đơn là hoàn thành hoặc không đến, giúp hệ thống theo dõi được hiệu quả phục vụ.

- ❖ Về phía Quản trị viên, đây là vai trò then chốt đảm bảo chất lượng nền tảng. Admin chịu trách nhiệm phê duyệt các đối tác mới đăng ký cũng như các nhà hàng mới được tạo, đảm bảo tính xác thực và chất lượng của thông tin. Ngoài ra, admin có thể tạm ngừng hoặc khóa tài khoản đối tác vi phạm, đồng thời theo dõi các số liệu thống kê tổng quan về hoạt động của hệ thống như số lượng người dùng, số đơn đặt bàn theo thời gian, tỷ lệ hủy và hoàn thành, từ đó có cơ sở đánh giá hiệu quả vận hành.
- ❖ Về mặt kỹ thuật, hệ thống được xây dựng trên nền tảng Django và Django REST Framework cho phía backend, sử dụng MySQL làm cơ sở dữ liệu quan hệ, và áp dụng JWT (JSON Web Token) cho việc xác thực người dùng. Kiến trúc hệ thống tuân theo mô hình RESTful API, tách bạch giữa backend xử lý logic nghiệp vụ và frontend hiển thị giao diện, tạo điều kiện thuận lợi cho việc mở rộng và bảo trì sau này.

## **Hướng giải quyết dự kiến**

Để xây dựng hệ thống đặt bàn ẩm thực một cách bài bản và khoa học, đồ án sẽ tuân theo quy trình phát triển phần mềm chuẩn, bao gồm các giai đoạn phân tích, thiết kế, triển khai và kiểm thử.

- ❖ Giai đoạn đầu tiên tập trung vào phân tích yêu cầu và xác định phạm vi hệ thống. Dựa trên nghiên cứu thực tế về quy trình đặt bàn tại các nhà hàng và nhu cầu của người dùng, đồ án sẽ xác định rõ các chức năng cốt lõi cần có, các ràng buộc nghiệp vụ, cũng như phân định vai trò và quyền hạn của từng nhóm người dùng. Kết quả của giai đoạn này là tài liệu đặc tả yêu cầu chi tiết, bao gồm các use case diagram mô tả tương tác giữa người dùng và hệ thống.
- ❖ Giai đoạn thiết kế được thực hiện ở hai mức độ: thiết kế hệ thống và thiết kế chi tiết. Ở mức hệ thống, đồ án sẽ thiết kế kiến trúc tổng thể theo mô hình client-server với RESTful API, xác định các thành phần chính như authentication service, restaurant service, booking service và notification service. Ở mức chi tiết, đồ án sẽ thiết kế cơ sở dữ liệu dựa trên mô hình quan hệ, xây dựng ERD (Entity Relationship Diagram) để thể hiện các thực thể và

mối quan hệ giữa chúng. Đồng thời, các class diagram, sequence diagram và activity diagram sẽ được xây dựng để làm rõ cấu trúc hệ thống và luồng xử lý nghiệp vụ.

- ❖ Giai đoạn triển khai sẽ được thực hiện theo phương pháp mô-đun hóa, phát triển từng chức năng độc lập và tích hợp dần. Hệ thống sẽ được chia thành các app Django riêng biệt theo chức năng: app accounts xử lý quản lý người dùng và xác thực, app restaurants quản lý thông tin nhà hàng và thực đơn, app bookings xử lý logic đặt bàn, và app notifications gửi thông báo cho người dùng. Mỗi app sẽ được phát triển độc lập với các model, view và serializer riêng, sau đó tích hợp lại thành hệ thống hoàn chỉnh thông qua các API endpoint.
- ❖ Đặc biệt, trong quá trình triển khai, đồ án sẽ chú trọng vào việc xử lý các nghiệp vụ phức tạp như kiểm tra tính khả dụng của khung giờ đặt bàn, đảm bảo không vượt quá số lượng chỗ tối đa, và tự động gửi thông báo khi có thay đổi trạng thái đơn đặt. Các ràng buộc nghiệp vụ như không cho phép đặt bàn quá khứ, không cho phép hủy đơn đã hoàn thành, và yêu cầu phê duyệt từ admin trước khi nhà hàng xuất hiện công khai sẽ được implement cẩn thận để đảm bảo tính toàn vẹn dữ liệu.
- ❖ Giai đoạn kiểm thử sẽ được thực hiện song song với quá trình phát triển, bao gồm unit test cho từng function, integration test cho các module tích hợp, và system test cho toàn bộ hệ thống. Các test case sẽ được thiết kế dựa trên use case và nghiệp vụ thực tế, đảm bảo hệ thống hoạt động đúng trong mọi tình huống. Cuối cùng, một số người dùng thử nghiệm sẽ được mời để sử dụng và đánh giá hệ thống, từ đó điều chỉnh và hoàn thiện trước khi triển khai chính thức.

# CHƯƠNG 1: KẾ HOẠCH THỰC HIỆN

## 1.1 Mô hình phát triển

- ❖ Đồ án áp dụng mô hình phát triển thác đổ có điều chỉnh (Modified Waterfall), trong đó các giai đoạn được thực hiện tuần tự nhưng cho phép quay lại giai đoạn trước nếu phát hiện vấn đề. Mô hình này phù hợp với quy mô và tính chất của đồ án, khi yêu cầu hệ thống đã được xác định tương đối rõ ràng từ đầu và cần có tài liệu đầy đủ cho từng giai đoạn phát triển.
- ❖ Quá trình phát triển được chia thành năm giai đoạn chính: Phân tích và Thu thập yêu cầu, Thiết kế hệ thống, Triển khai và Lập trình, Kiểm thử và Tích hợp, và cuối cùng là Triển khai và Bảo trì. Mỗi giai đoạn có mục tiêu và sản phẩm đầu ra cụ thể, đồng thời phục vụ làm nền tảng cho giai đoạn tiếp theo. Việc hoàn thành đầy đủ một giai đoạn trước khi chuyển sang giai đoạn kế tiếp giúp kiểm soát chất lượng tốt hơn và giảm thiểu rủi ro phát sinh trong quá trình phát triển.

## 1.2 Giai đoạn Phân tích và Thu thập yêu cầu

- ❖ Giai đoạn đầu tiên tập trung vào việc nghiên cứu và phân tích đầy đủ các yêu cầu chức năng và phi chức năng của hệ thống. Quá trình này bắt đầu bằng việc nghiên cứu thực tế về quy trình đặt bàn tại các nhà hàng, phỏng vấn chủ nhà hàng để hiểu các vấn đề họ gặp phải trong quản lý đặt chỗ, đồng thời khảo sát khách hàng để nắm bắt nhu cầu và mong muốn khi sử dụng dịch vụ đặt bàn trực tuyến.
- ❖ Từ dữ liệu thu thập được, đồ án sẽ xác định ba nhóm người dùng chính của hệ thống là Khách hàng, Đối tác nhà hàng và Quản trị viên, sau đó phân tích chi tiết các chức năng cần thiết cho từng nhóm. Đối với khách hàng, các chức năng cơ bản bao gồm đăng ký tài khoản, tìm kiếm nhà hàng theo địa điểm, xem thông tin chi tiết nhà hàng, đặt bàn trực tuyến, xem lịch sử đặt bàn và hủy đơn nếu cần. Đối với đối tác nhà hàng, hệ thống cần hỗ trợ đăng ký tài khoản đối tác, tạo và quản lý thông tin nhà hàng, thiết lập thực đơn, cấu hình khung giờ đặt bàn, xem và xử lý các đơn đặt bàn. Đối với quản trị viên, các chức năng quản trị bao gồm phê duyệt đối tác và nhà hàng mới, tạm ngừng hoặc khóa tài khoản vi phạm, và theo dõi thống kê hoạt động hệ thống.

- ❖ Bên cạnh yêu cầu chức năng, giai đoạn này cũng xác định các yêu cầu phi chức năng quan trọng như hiệu năng hệ thống phải đáp ứng được đồng thời nhiều người dùng, bảo mật thông tin cá nhân và dữ liệu đặt bàn, giao diện thân thiện và dễ sử dụng, khả năng mở rộng để thêm tính năng trong tương lai. Đặc biệt, các ràng buộc nghiệp vụ cũng được làm rõ trong giai đoạn này, chẳng hạn như đối tác phải được admin phê duyệt mới có thể tạo nhà hàng, nhà hàng phải được duyệt mới hiển thị công khai, không cho phép đặt bàn quá khứ, phải kiểm tra số chỗ trống trước khi chấp nhận đơn đặt, và khách hàng chỉ có thể hủy đơn khi còn ở trạng thái chờ xác nhận hoặc đã xác nhận.
- ❖ Sản phẩm đầu ra của giai đoạn này là tài liệu Đặc tả yêu cầu phần mềm (SRS - Software Requirements Specification) chi tiết, bao gồm mô tả tổng quan hệ thống, danh sách các chức năng theo từng vai trò người dùng, các use case diagram với mô tả chi tiết từng use case, các ràng buộc nghiệp vụ và yêu cầu phi chức năng. Tài liệu này sẽ là cơ sở để thực hiện giai đoạn thiết kế tiếp theo.

### 1.3 Giai đoạn Thiết kế hệ thống

- ❖ Sau khi hoàn tất phân tích yêu cầu, giai đoạn thiết kế sẽ chuyển các yêu cầu thành các mô hình và kiến trúc hệ thống cụ thể. Giai đoạn này được chia thành hai phần chính là thiết kế kiến trúc tổng thể và thiết kế chi tiết.
- ❖ Trong thiết kế kiến trúc tổng thể, đồ án sẽ xác định cấu trúc hệ thống theo mô hình client-server với kiến trúc RESTful API. Hệ thống được chia thành ba tầng chính: tầng presentation là các giao diện người dùng cho khách hàng, đối tác và admin; tầng application là các API endpoint xử lý logic nghiệp vụ; và tầng data là cơ sở dữ liệu MySQL lưu trữ thông tin. Việc áp dụng kiến trúc RESTful giúp tách biệt rõ ràng giữa frontend và backend, tạo điều kiện cho việc phát triển song song và dễ dàng mở rộng trong tương lai.
- ❖ Hệ thống backend sẽ được tổ chức thành các module chức năng độc lập theo pattern của Django apps. Module accounts chịu trách nhiệm quản lý người dùng, xác thực và phân quyền, sử dụng JWT token để duy trì phiên làm việc. Module restaurants quản lý thông tin nhà hàng, thực đơn, hình ảnh và khung giờ đặt bàn. Module bookings xử lý toàn bộ logic đặt bàn, từ kiểm tra tính khả dụng, tạo đơn mới, đến xác nhận và cập nhật trạng thái. Module notifications đảm nhiệm việc gửi thông báo tự động cho người dùng khi có sự kiện quan

trọng xảy ra. Cuối cùng, module `admin_panel` cung cấp các chức năng quản trị và thống kê cho quản trị viên.

- ❖ Trong thiết kế chi tiết, phần quan trọng nhất là thiết kế cơ sở dữ liệu. Dựa trên phân tích yêu cầu, hệ thống cần quản lý mười một thực thể chính bao gồm người dùng, khách hàng, đối tác, địa điểm, nhà hàng, khung giờ, hình ảnh nhà hàng, món ăn, đơn đặt bàn và thông báo. Các thực thể này được thiết kế thành các bảng trong cơ sở dữ liệu quan hệ với các mối quan hệ rõ ràng: mỗi người dùng có thể là một khách hàng hoặc một đối tác thông qua quan hệ một-một, mỗi đối tác có thể sở hữu nhiều nhà hàng, mỗi nhà hàng thuộc về một địa điểm và có nhiều khung giờ đặt bàn, nhiều hình ảnh, nhiều món ăn. Một đơn đặt bàn liên kết với một khách hàng, một nhà hàng và một khung giờ cụ thể.
- ❖ Đối với mỗi bảng, đồ án sẽ thiết kế chi tiết các trường dữ liệu với kiểu dữ liệu phù hợp, xác định khóa chính và khóa ngoại, thiết lập các ràng buộc toàn vẹn dữ liệu như NOT NULL, UNIQUE, CHECK constraints, và thiết kế các index để tối ưu hóa truy vấn. Ví dụ, bảng `bookings` sẽ có ràng buộc kiểm tra ngày đặt phải lớn hơn hoặc bằng ngày hiện tại, số lượng khách phải lớn hơn 0, và các index trên `customer_id`, `restaurant_id` và `booking_date` để tăng tốc độ truy vấn.
- ❖ Ngoài thiết kế dữ liệu, giai đoạn này còn xây dựng các sequence diagram để mô tả chi tiết luồng tương tác giữa các thành phần trong những nghiệp vụ phức tạp như quy trình đặt bàn từ lúc khách hàng gửi yêu cầu, hệ thống kiểm tra slot khả dụng, tạo đơn, đến gửi thông báo cho đối tác. Activity diagram cũng được xây dựng để làm rõ các bước xử lý và điểm quyết định trong từng chức năng. Class diagram thể hiện cấu trúc các lớp đối tượng trong hệ thống, các thuộc tính và phương thức của chúng, cũng như mối quan hệ kế thừa và kết hợp giữa các lớp.
- ❖ Sản phẩm của giai đoạn thiết kế bao gồm tài liệu Thiết kế hệ thống chi tiết với các sơ đồ kiến trúc tổng thể, ERD (Entity Relationship Diagram), class diagram, sequence diagram cho các use case quan trọng, activity diagram cho các nghiệp vụ phức tạp, đặc tả API endpoints với format request/response, và schema cơ sở dữ liệu hoàn chỉnh với tất cả các bảng, trường, ràng buộc và index.

## 1.4 Giai đoạn Triển khai và Lập trình

- ❖ Giai đoạn triển khai là quá trình chuyển các thiết kế thành mã nguồn thực tế. Để đảm bảo chất lượng và khả năng bảo trì, đồ án áp dụng chiến lược phát triển từng module độc lập theo thứ tự phụ thuộc, sau đó tích hợp dần thành hệ thống hoàn chỉnh.
- ❖ Module đầu tiên được triển khai là accounts, vì đây là module nền tảng mà các module khác phụ thuộc vào. Trong module này, trước tiên sẽ xây dựng các model User, Customer và Partner với đầy đủ các trường thông tin theo thiết kế database. Tiếp theo là implement hệ thống xác thực JWT bao gồm các chức năng tạo token khi đăng nhập, xác thực token trong các request sau đó, và làm mới token khi hết hạn. Các API endpoint cho đăng ký khách hàng và đối tác được phát triển với đầy đủ validation để đảm bảo dữ liệu đầu vào hợp lệ, ví dụ kiểm tra số điện thoại chưa tồn tại, mật khẩu đủ mạnh. API đăng nhập xác thực thông tin và trả về JWT token. Các API quản lý profile cho phép người dùng xem và cập nhật thông tin cá nhân. Cuối cùng là implement hệ thống phân quyền dựa trên role, đảm bảo mỗi endpoint chỉ có thể được truy cập bởi đúng nhóm người dùng có quyền.
- ❖ Sau khi hoàn thành module accounts, tiếp theo là module restaurants vì nó cung cấp dữ liệu cơ bản về nhà hàng cho các module khác. Module này bắt đầu với việc xây dựng model Location để quản lý thông tin địa điểm theo cấp thành phố, quận huyện và phường xã. Model Restaurant được tạo với đầy đủ thông tin như tên, địa chỉ, mô tả, giờ mở cửa, thời lượng một slot đặt bàn, trạng thái và đánh giá. Các model liên quan như RestaurantImage cho hình ảnh nhà hàng, MenuItem cho món ăn trong thực đơn, và TimeSlot cho các khung giờ đặt bàn cũng được implement trong module này.
- ❖ Các API endpoint cho nhà hàng được phát triển theo nguyên tắc RESTful: API liệt kê nhà hàng có thể truy cập công khai với các bộ lọc theo địa điểm và đánh giá, API xem chi tiết nhà hàng cũng public và trả về đầy đủ thông tin bao gồm hình ảnh và thực đơn. Các API tạo, sửa, xóa nhà hàng chỉ cho phép đối tác thao tác với nhà hàng của mình. Đặc biệt, business logic được implement để đảm bảo nhà hàng mới tạo có trạng thái PENDING và chỉ hiển thị công khai sau khi admin duyệt chuyển sang APPROVED. Tương tự, các API quản lý menu và time slots cũng được phát triển với đầy đủ validation và permission checking.

- ❖ Module bookings là module quan trọng nhất chứa logic nghiệp vụ cốt lõi của hệ thống. Model Booking được thiết kế với các trường thông tin về khách hàng, nhà hàng, khung giờ đã chọn, ngày đặt, số người, yêu cầu đặc biệt và trạng thái đơn. Trạng thái đơn đặt bàn được quản lý thông qua một state machine với các trạng thái PENDING, CONFIRMED, REJECTED, CANCELLED, COMPLETED và NO\_SHOW, cùng với các quy tắc chuyển đổi trạng thái hợp lệ.
- ❖ API tạo booking là API phức tạp nhất, yêu cầu nhiều bước validation. Đầu tiên kiểm tra ngày đặt không được là quá khứ, sau đó validate nhà hàng có trạng thái APPROVED, time slot tồn tại và đang active. Bước quan trọng nhất là kiểm tra tính khả dụng của slot bằng cách đếm số lượng booking hiện có cho cùng nhà hàng, ngày và khung giờ với trạng thái không phải CANCELLED hay REJECTED, rồi so sánh với max\_bookings của time slot. Chỉ khi còn chỗ trống, đơn mới được tạo với trạng thái PENDING và một notification được gửi tự động cho đối tác nhà hàng.
- ❖ Các API khác trong module bookings bao gồm API xem danh sách booking của khách hàng, API xem booking của nhà hàng dành cho đối tác, API xác nhận hoặc từ chối booking chỉ cho phép khi trạng thái là PENDING, API hủy booking cho khách hàng khi trạng thái còn là PENDING hoặc CONFIRMED, và API đánh dấu hoàn thành hoặc không đến cho đối tác sau khi đã đến ngày đặt. Mỗi thay đổi trạng thái đều kích hoạt gửi notification cho bên liên quan.
- ❖ Module notifications được phát triển song song với bookings để hỗ trợ việc gửi thông báo tự động. Model Notification lưu trữ thông tin về người nhận, tiêu đề, nội dung, loại thông báo và trạng thái đã đọc. Các hàm helper được viết để tạo notification dễ dàng từ các module khác, ví dụ hàm notify\_partner\_new\_booking khi có đơn đặt mới, notify\_customer\_booking\_confirmed khi đơn được xác nhận. API endpoints cho phép người dùng xem danh sách thông báo của mình, đánh dấu đã đọc, và xóa thông báo không cần thiết.
- ❖ Cuối cùng, app admin\_panel được hoàn thiện để phục vụ duyệt đối tác/nhà hàng và thống kê vận hành; ở giai đoạn đầu có thể tận dụng cơ chế quản trị sẵn có để đảm bảo tiến độ, sau đó hoàn thiện dashboard/statistics theo yêu cầu đề án. Quá trình triển khai được thực hiện theo hướng tích hợp dần: mỗi app hoàn

thành sẽ được kết nối vào luồng tổng thể để kiểm tra tương thích dữ liệu – phân quyền – trạng thái, tránh dồn lỗi tích hợp về cuối.

### **1.5 Giai đoạn kiểm thử và tích hợp**

Kiểm thử được tổ chức theo hướng bám sát nghiệp vụ và use case đã mô hình hóa, đảm bảo mỗi chức năng hoạt động đúng và phối hợp trơn tru trong luồng tổng thể. Ở mức đơn vị, các kiểm tra tập trung vào validation dữ liệu, quy tắc phân quyền theo role, và logic trạng thái của booking để tránh sai lệch chuyển trạng thái hoặc cho phép thao tác trái điều kiện. Ở mức tích hợp, trọng tâm là các luồng xuyên app như: tạo booking → gửi thông báo → partner xử lý → hệ thống cập nhật trạng thái → khách nhận thông báo; đồng thời kiểm tra tính nhất quán dữ liệu khi lọc nhà hàng công khai (APPROVED) và khi kiểm tra khả dụng theo khung giờ. Ở mức hệ thống, kiểm thử tập trung vào các kịch bản thực tế (đặt quá khứ, slot đầy, hủy sai trạng thái, partner xử lý đơn đã được xử lý...) để đảm bảo phản hồi đúng và dữ liệu không bị “bẩn”. Kết quả của giai đoạn này là bộ test case theo use case và báo cáo lỗi/điều chỉnh, làm cơ sở cho bước hoàn thiện.

### **1.6 Giai đoạn hoàn thiện và đóng gói đồ án**

Sau khi các luồng nghiệp vụ chạy ổn định, hệ thống được hoàn thiện về trải nghiệm sử dụng, chuẩn hóa dữ liệu mẫu phục vụ demo và rà soát các yêu cầu phi chức năng quan trọng như bảo mật (JWT, phân quyền), tính nhất quán dữ liệu và hiệu năng truy vấn ở các màn hình hay dùng (danh sách nhà hàng, danh sách booking, thống kê). Song song, tài liệu đồ án được hoàn tất theo cấu trúc chương mục: mô tả hệ thống, kế hoạch thực hiện, thiết kế (BFD/Use Case/UML/DFD/ERD), mô tả luồng xử lý và kịch bản kiểm thử; đảm bảo thống nhất giữa mô hình thiết kế và hệ thống đã triển khai. Cuối cùng là đóng gói sản phẩm demo (môi trường chạy, dữ liệu mẫu, hướng dẫn sử dụng) để phục vụ trình bày và đánh giá, đồng thời nêu rõ các giới hạn phạm vi và hướng mở rộng sau này để thể hiện khả năng phát triển tiếp.



## CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG

### 2.1. Khảo sát hiện trạng

Hiện nay, tại nhiều nhà hàng vừa và nhỏ, việc đặt bàn vẫn chủ yếu được thực hiện thông qua các hình thức truyền thống như gọi điện thoại trực tiếp hoặc nhắn tin qua mạng xã hội. Cách làm này tuy đơn giản nhưng tồn tại nhiều hạn chế, đặc biệt khi số lượng khách tăng cao vào các khung giờ cao điểm. Nhân viên dễ ghi nhận sai thông tin, trùng lịch đặt bàn hoặc không kiểm soát chính xác số lượng khách trong từng khung giờ.

Đối với khách hàng, quá trình đặt bàn thủ công gây mất thời gian, phụ thuộc vào thời gian làm việc của nhà hàng và khó theo dõi trạng thái đặt bàn. Khách hàng không thể biết chính xác khung giờ nào còn trống hoặc yêu cầu của mình đã được xác nhận hay chưa.

Bên cạnh đó, một số nền tảng đặt bàn hiện nay chỉ tập trung vào các chuỗi nhà hàng lớn, chưa phù hợp với các nhà hàng nhỏ lẻ. Ngoài ra, nhiều hệ thống chưa cho phép nhà hàng chủ động quản lý khung giờ hoặc xử lý linh hoạt các trường hợp như hủy bàn, từ chối yêu cầu hoặc khách không đến.

Từ những bất cập trên, nhu cầu xây dựng một hệ thống web đặt bàn ăn có tính tập trung, dễ sử dụng và phù hợp với quy mô nhà hàng vừa và nhỏ là hoàn toàn cần thiết.

### 2.2. Yêu cầu chức năng và phi chức năng

#### 2.2.1. Yêu cầu chức năng

Hệ thống cần đáp ứng các chức năng cơ bản cho từng nhóm người dùng như sau:

**Đối với khách hàng:**

- ❖ Đăng ký và đăng nhập tài khoản.
- ❖ Tìm kiếm và xem danh sách nhà hàng.
- ❖ Xem thông tin chi tiết nhà hàng, bao gồm địa chỉ, hình ảnh, menu và khung giờ.
- ❖ Thực hiện đặt bàn theo ngày, khung giờ và số lượng khách.

- ❖ Xem lịch sử đặt bàn và hủy đặt bàn trong các trường hợp cho phép.

#### ***Đối với đối tác nhà hàng (Partner):***

- ❖ Đăng ký tham gia hệ thống và chờ quản trị viên phê duyệt.
- ❖ Quản lý thông tin nhà hàng, hình ảnh và menu.
- ❖ Tạo và quản lý các khung giờ đặt bàn.
- ❖ Xác nhận, từ chối hoặc cập nhật trạng thái các đơn đặt bàn.

#### ***Đối với quản trị viên (Admin):***

- ❖ Quản lý tài khoản người dùng và đối tác.
- ❖ Duyệt nhà hàng trước khi hiển thị công khai.
- ❖ Giám sát hoạt động chung của hệ thống.
- ❖ Các chức năng trên được thiết kế nhằm đáp ứng đầy đủ quy trình nghiệp vụ đặt bàn trong thực tế, từ lúc khách hàng gửi yêu cầu đến khi nhà hàng hoàn tất phục vụ.

#### ***2.2.2. Yêu cầu phi chức năng***

Bên cạnh các chức năng nghiệp vụ, hệ thống cần đáp ứng các yêu cầu phi chức năng nhằm đảm bảo khả năng vận hành ổn định và an toàn:

- Tính dễ sử dụng: Giao diện và luồng thao tác đơn giản, dễ hiểu đối với người dùng không am hiểu kỹ thuật.
- Tính bảo mật: Thông tin người dùng phải được bảo vệ, mật khẩu được lưu dưới dạng mã hóa, phân quyền rõ ràng theo vai trò.
- Hiệu năng: Hệ thống cần phản hồi nhanh các yêu cầu đặt bàn và truy vấn dữ liệu, đặc biệt trong các khung giờ cao điểm.
- Tính mở rộng: Kiến trúc hệ thống cho phép dễ dàng bổ sung các chức năng mới trong tương lai như thanh toán trực tuyến hoặc đánh giá nhà hàng.
- Độ tin cậy: Dữ liệu đặt bàn phải được lưu trữ chính xác, tránh mất mát hoặc trùng lặp thông tin.

## **2.3. Các đối tượng và chức năng hệ thống**

### **2.3.1. Khách hàng**

Khách hàng là người sử dụng hệ thống để tìm kiếm và đặt bàn ăn. Các chức năng chính bao gồm xem thông tin nhà hàng, lựa chọn khung giờ phù hợp, gửi yêu cầu đặt bàn và theo dõi trạng thái đơn đặt bàn. Khách hàng cũng có thể hủy đặt bàn khi kế hoạch thay đổi, giúp nhà hàng chủ động điều chỉnh lịch phục vụ.

### **2.3.2. Đối tác nhà hàng (Partner)**

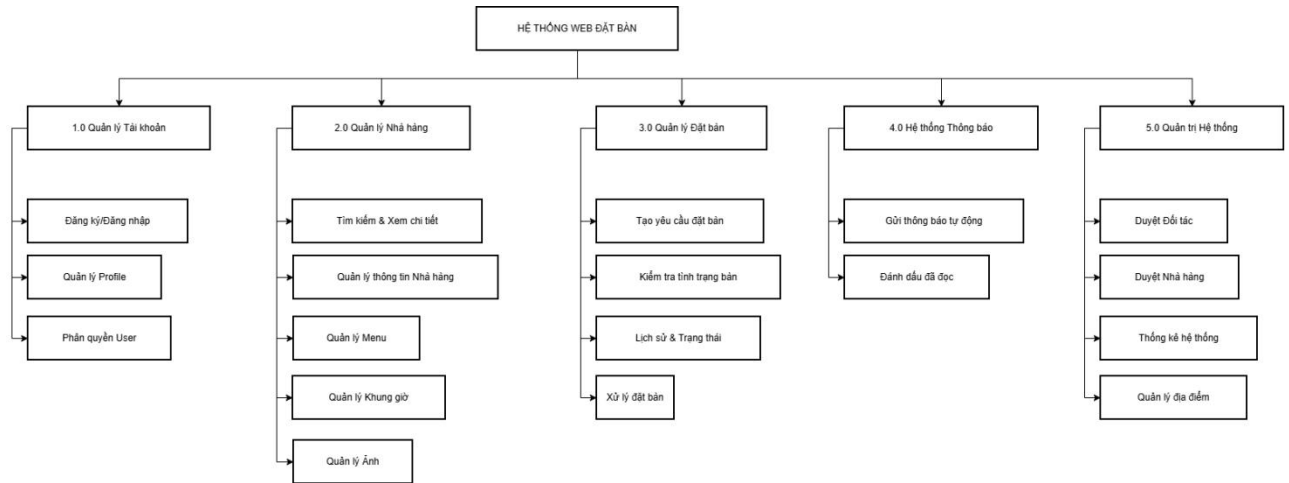
Đối tác nhà hàng là người quản lý trực tiếp các hoạt động của nhà hàng trên hệ thống. Partner chịu trách nhiệm cập nhật thông tin nhà hàng, tạo khung giờ đặt bàn và xử lý các yêu cầu từ khách hàng. Việc cho phép partner chủ động xác nhận hoặc từ chối đơn đặt bàn giúp phản ánh đúng khả năng phục vụ thực tế của nhà hàng.

### **2.3.3. Quản trị viên (Admin)**

Quản trị viên đóng vai trò giám sát và điều phối hệ thống. Admin có quyền kiểm duyệt các nhà hàng trước khi công khai, đảm bảo thông tin hiển thị trên hệ thống là hợp lệ và đáng tin cậy. Ngoài ra, admin còn theo dõi hoạt động tổng thể nhằm đảm bảo hệ thống vận hành ổn định.

# CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

## 3.1. Sơ đồ BFD



Hình 1: Sơ đồ BFD kiến trúc hệ thống

Mô tả các nhóm chức năng

### 1. Quản lý Tài khoản (Account Management)

Nhóm chức năng chịu trách nhiệm về định danh và thông tin người dùng trong hệ thống.

- **Đăng ký/Đăng nhập:** Xử lý việc tạo tài khoản mới và xác thực người dùng truy cập.
- **Quản lý Profile:** Cho phép người dùng xem và cập nhật thông tin cá nhân (họ tên, SĐT, email).
- **Phân quyền User:** Hệ thống phân chia vai trò người dùng (Khách hàng, Chủ nhà hàng, Admin) để cấp quyền truy cập tương ứng.

### 2. Quản lý Nhà hàng (Restaurant Management)

Nhóm chức năng cốt lõi liên quan đến dữ liệu của các đối tác nhà hàng.

- **Tìm kiếm & Xem chi tiết:** Chức năng dành cho khách hàng để tìm nhà hàng theo bộ lọc và xem thông tin chi tiết.
- **Quản lý thông tin Nhà hàng:** Cho phép đối tác cập nhật địa chỉ, giới thiệu, hotline.

- **Quản lý Menu:** Thêm, sửa, xóa món ăn và cập nhật giá cả.
- **Quản lý Khung giờ (Time Slots):** Thiết lập các khung giờ mở cửa và giới hạn số lượng bàn.
- **Quản lý Ảnh:** Tải lên và sắp xếp hình ảnh không gian quán, món ăn.

### 3. Quản lý Đặt bàn (Booking Management)

Nhóm chức năng xử lý nghiệp vụ chính của hệ thống: quy trình đặt bàn ăn.

- **Tạo yêu cầu đặt bàn:** Khách hàng chọn ngày, giờ và số lượng người để gửi yêu cầu.
- **Kiểm tra tình trạng bàn:** Hệ thống tự động kiểm tra xem khung giờ đó còn chỗ trống hay không (Available Slots).
- **Lịch sử & Trạng thái:** Người dùng xem lại các đơn đặt bàn cũ và trạng thái hiện tại của đơn.
- **Xử lý (Booking Processing):** Đây là quy trình nghiệp vụ chi tiết bao gồm các hành động:
  - **Xác nhận/Từ chối:** Đối tác phản hồi yêu cầu của khách.
  - **Hoàn thành/No-show:** Cập nhật trạng thái sau khi thời gian đặt bàn kết thúc (Khách đã đến ăn hoặc không đến).
  - **Hủy đặt bàn:** Cho phép hủy đơn khi đang ở trạng thái chờ.

### 4. Hệ thống Thông báo

Nhóm chức năng đảm bảo sự liên lạc giữa hệ thống và người dùng.

- **Gửi thông báo tự động:** Hệ thống tự động gửi tin khi có sự kiện (Có đơn mới, Đơn được duyệt, Đơn bị hủy).
- **Đánh dấu đã đọc:** Quản lý trạng thái xem thông báo của người dùng.

### 5. Quản trị Hệ thống

Nhóm chức năng dành riêng cho Quản trị viên (Admin) để vận hành nền tảng.

- **Duyệt Đối tác:** Xác minh thông tin đăng ký của chủ nhà hàng.

- **Duyệt Nhà hàng:** Kiểm duyệt nội dung nhà hàng trước khi cho phép hiển thị công khai (Public).
- **Thống kê hệ thống:** Xem báo cáo tổng quan về lượng người dùng, đơn đặt bàn.
- **Quản lý địa điểm:** Quản lý danh mục Tỉnh/Thành phố, Quận/Huyện để phục vụ chức năng tìm kiếm.

### 3.2. Use case



Hình 2: Use case tổng quát

## Mô tả tổng quan

### 1. Các tác nhân (Actors)

- ❖ **Khách hàng (Customer):** người dùng cuối sử dụng hệ thống để tra cứu thông tin nhà hàng và thực hiện đặt bàn, đồng thời theo dõi và thao tác trên các lượt đặt của mình.
- ❖ **Đối tác (Partner):** chủ/đại diện nhà hàng tham gia nền tảng để đăng ký trở thành đối tác, quản trị thông tin nhà hàng, cấu hình hoạt động và xử lý các đơn đặt bàn từ khách.
- ❖ **Admin (Quản trị viên):** người vận hành hệ thống, chịu trách nhiệm quản lý đối tác, quản lý người dùng, thiết lập cấu hình chung và phân quyền để hệ thống hoạt động đúng chính sách.

### 2. Chi tiết chức năng theo tác nhân

#### A. Nhóm chức năng dành cho Khách hàng

Trong sơ đồ, Khách hàng có thể Đăng ký để tạo tài khoản mới và Đăng nhập để truy cập hệ thống. Khi sử dụng dịch vụ, Khách hàng thực hiện Tìm kiếm nhà hàng và Xem chi tiết nhà hàng để xem thông tin cần thiết trước khi ra quyết định. Từ màn hình chi tiết, Khách hàng có thể lựa chọn Xem thực đơn như một thao tác mở rộng (không bắt buộc). Sau khi chọn được nhà hàng phù hợp, Khách hàng thực hiện Đặt bàn để gửi yêu cầu đặt chỗ. Ngoài ra, Khách hàng có chức năng Quản lý lịch sử đặt bàn để theo dõi các lượt đặt đã tạo; trong quá trình quản lý lịch sử này, Khách hàng có thể Hủy/Sửa đặt bàn như một thao tác mở rộng khi cần thay đổi kế hoạch.

#### B. Nhóm chức năng dành cho Đối tác (Nhà hàng)

Đối tác bắt đầu bằng Đăng ký đối tác để gửi yêu cầu tham gia nền tảng, sau đó sử dụng Đăng nhập để thao tác các chức năng quản trị. Khi đã có quyền truy cập, Đối tác thực hiện Quản lý hồ sơ nhà hàng nhằm cập nhật thông tin nhà hàng (phục vụ hiển thị cho khách), Quản lý thực đơn để cập nhật các món ăn, và Cấu hình nhà hàng để thiết lập các thông số vận hành (phục vụ hoạt động đặt bàn). Về nghiệp vụ đặt bàn, Đối tác có chức năng Quản lý đơn đặt bàn để xem và xử lý các yêu cầu từ khách; trong chức năng này, thao tác Xác nhận / từ chối đặt bàn được thể hiện là phần mở rộng khi Đối tác quyết định xử lý từng đơn cụ thể.

### **C. Nhóm chức năng dành cho Admin**

Admin sử dụng Đăng nhập để vào hệ thống quản trị. Trong sơ đồ, Admin có nhóm chức năng Quản lý đối tác (theo dõi/kiểm soát đối tác trên nền tảng), Quản lý người dùng (quản trị tài khoản người dùng), Cấu hình hệ thống (thiết lập tham số chung), và Phân quyền (quản lý quyền truy cập theo vai trò). Riêng nghiệp vụ kiểm duyệt đối tác được thể hiện qua use case Duyệt đăng ký đối tác như một thao tác mở rộng gắn với quá trình Quản lý đối tác.

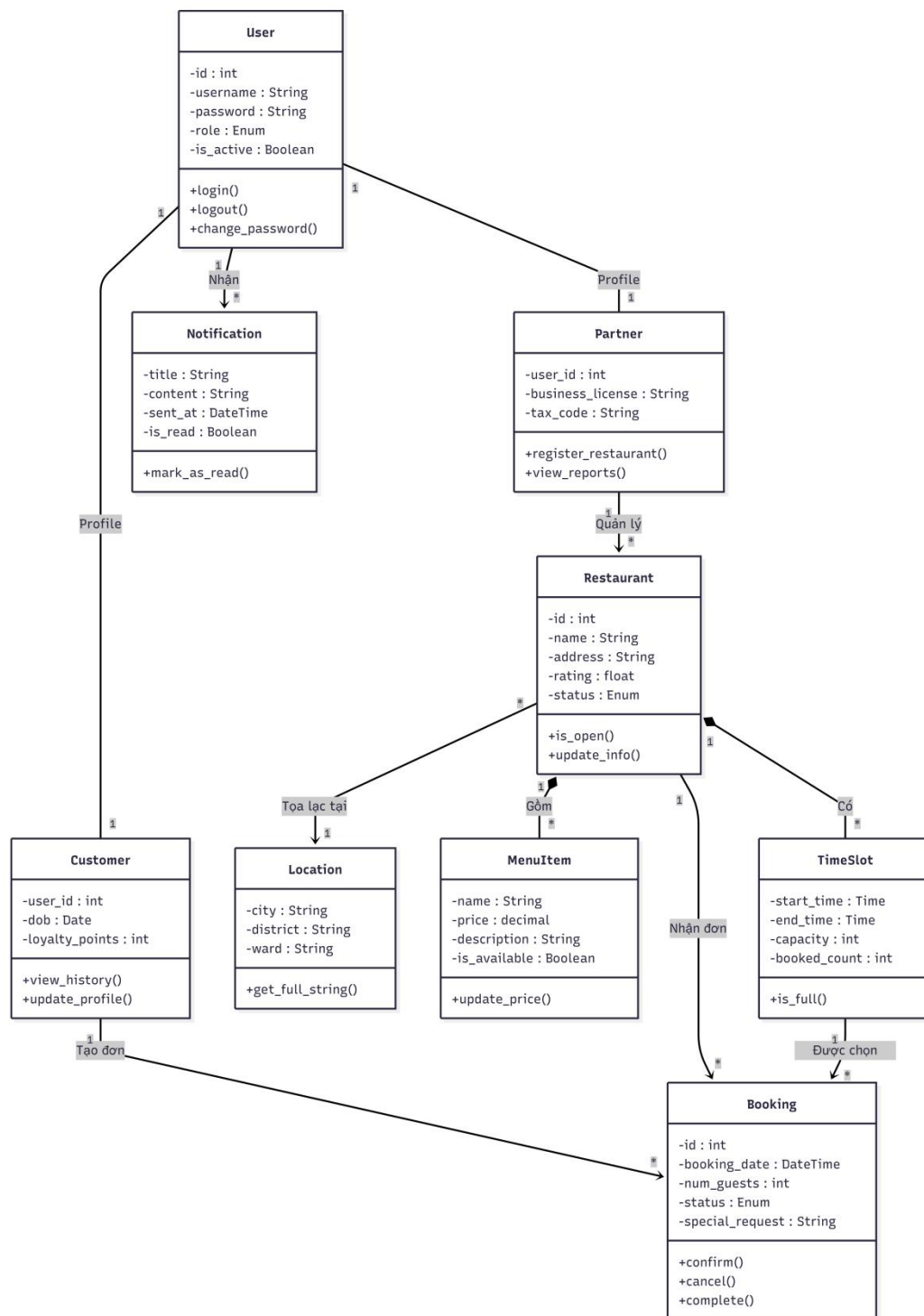
### **3. Các mối quan hệ đặc biệt trong sơ đồ**

**Association (nét liền):** thể hiện tác nhân trực tiếp sử dụng chức năng (ví dụ Khách hàng ↔ Tìm kiếm nhà hàng; Đối tác ↔ Quản lý đơn đặt bàn; Admin ↔ Phân quyền).

**<extend> (nét đứt):** thể hiện chức năng mở rộng, chỉ xảy ra khi có nhu cầu, không phải bước bắt buộc của chức năng chính. Trong sơ đồ có các quan hệ: Xem thực đơn mở rộng Xem chi tiết nhà hàng; Hủy/Sửa đặt bàn mở rộng Quản lý lịch sử đặt bàn; Xác nhận / từ chối đặt bàn mở rộng Quản lý đơn đặt bàn; và Duyệt đăng ký đối tác mở rộng Quản lý đối tác.



### 3.3. Mô hình UML



Hình 3: Sơ đồ UML

#### Mô tả

#### 1. Phân tích chi tiết các nhóm lớp

##### A. Nhóm Quản lý Người dùng (User + Profile)

Hệ thống quản lý tài khoản theo mô hình User kết hợp hồ sơ (Profile) thay vì kế thừa.

❖ **Lớp User (tài khoản đăng nhập):**

- ◆ Đóng gói thông tin xác thực và trạng thái: id, username, password, role (Enum), is\_active (Boolean).
- ◆ Cung cấp các phương thức dùng chung: login(), logout(), change\_password().
- ◆ Quan hệ: User có thể gắn 01 hồ sơ Customer hoặc 01 hồ sơ Partner (liên kết “Profile” kiểu 1-1).

❖ **Lớp Customer (hồ sơ Khách hàng):**

- ◆ Không kế thừa từ User; là hồ sơ liên kết qua user\_id.
- ◆ Thuộc tính: dob, loyalty\_points.
- ◆ Phương thức: view\_history() (xem lịch sử đặt/hoạt động), update\_profile().
- ◆ Lớp Partner (hồ sơ Đối tác):
- ◆ Không kế thừa từ User; là hồ sơ liên kết qua user\_id.
- ◆ Thuộc tính pháp lý: business\_license, tax\_code.
- ◆ Phương thức: register\_restaurant() (đăng ký/quản lý đăng ký nhà hàng), view\_reports().

***B. Nhóm Quản lý Nhà hàng (Restaurant Management)***

Nhóm này thể hiện rõ thành phần (Composition – hình thoi đen) giữa Restaurant và các thành phần con.

❖ **Lớp Restaurant:**

- ◆ Là trung tâm nghiệp vụ: id, name, address, rating, status (Enum).
- ◆ Phương thức: is\_open(), update\_info().
- ◆ Quan hệ quản lý: do Partner quản lý (một Partner có thể quản lý nhiều Restaurant).

❖ **Lớp MenuItem (Thực đơn):**

- ◆ Quan hệ Composition với Restaurant (món ăn phụ thuộc vòng đời nhà hàng).
- ◆ **Thuộc tính:** name, price, description, is\_available.

- ◆ **Phương thức:** `update_price()`.

❖ **Lớp TimeSlot (Khung giờ):**

- ◆ Quan hệ Composition với Restaurant.
- ◆ **Thuộc tính:** `start_time`, `end_time`, `capacity`, `booked_count`.
- ◆ Phương thức: `is_full()` kiểm tra khung giờ đã đầy hay chưa.

❖ **Lớp Location:**

- ◆ Thông tin định vị: `city`, `district`, `ward`.
- ◆ Phương thức: `get_full_string()`.
- ◆ Quan hệ: Restaurant gắn với Location để biểu diễn vị trí (một Location có thể được nhiều Restaurant dùng).

**C. Nhóm Xử lý Đặt bàn (*Booking Transaction*)**

❖ **Lớp Booking:**

- ◆ Là bản ghi nghiệp vụ đặt bàn, liên kết các thực thể chính.
- ◆ Thuộc tính: `id`, `booking_date` (`DateTime`), `num_guests`, `status` (`Enum`), `special_request`.
- ◆ Phương thức: `confirm()`, `cancel()`, `complete()`.
- ◆ Quan hệ đúng theo hình: Booking được Customer tạo, thuộc về Restaurant nhận đơn, và chọn 1 TimeSlot để đặt.

**D. Nhóm Phụ trợ**

❖ **Lớp Notification:**

- ◆ Gửi thông báo cho User.
- ◆ Thuộc tính: `title`, `content`, `sent_at`, `is_read`.
- ◆ Phương thức: `mark_as_read()`.
- ◆ Quan hệ: 1 User nhận nhiều Notification.

## 2. Các mối quan hệ quan trọng (Relationships)

### ❖ Association (Profile 1-1):

User —(Profile)→ Customer (1-1, thông qua user\_id)

User —(Profile)→ Partner (1-1, thông qua user\_id)

Không dùng Generalization/kế thừa trong sơ đồ này.

### ❖ Composition (hình thoi đen):

Restaurant ◆— MenuItem (1 nhà hàng có nhiều món; món phụ thuộc nhà hàng)

Restaurant ◆— TimeSlot (1 nhà hàng có nhiều khung giờ; khung giờ phụ thuộc nhà hàng)

### ❖ Association nghiệp vụ chính:

Partner —(Quản lý)→ Restaurant

Restaurant —(Tọa lạc tại)→ Location

Customer —(Tạo đơn)→ Booking

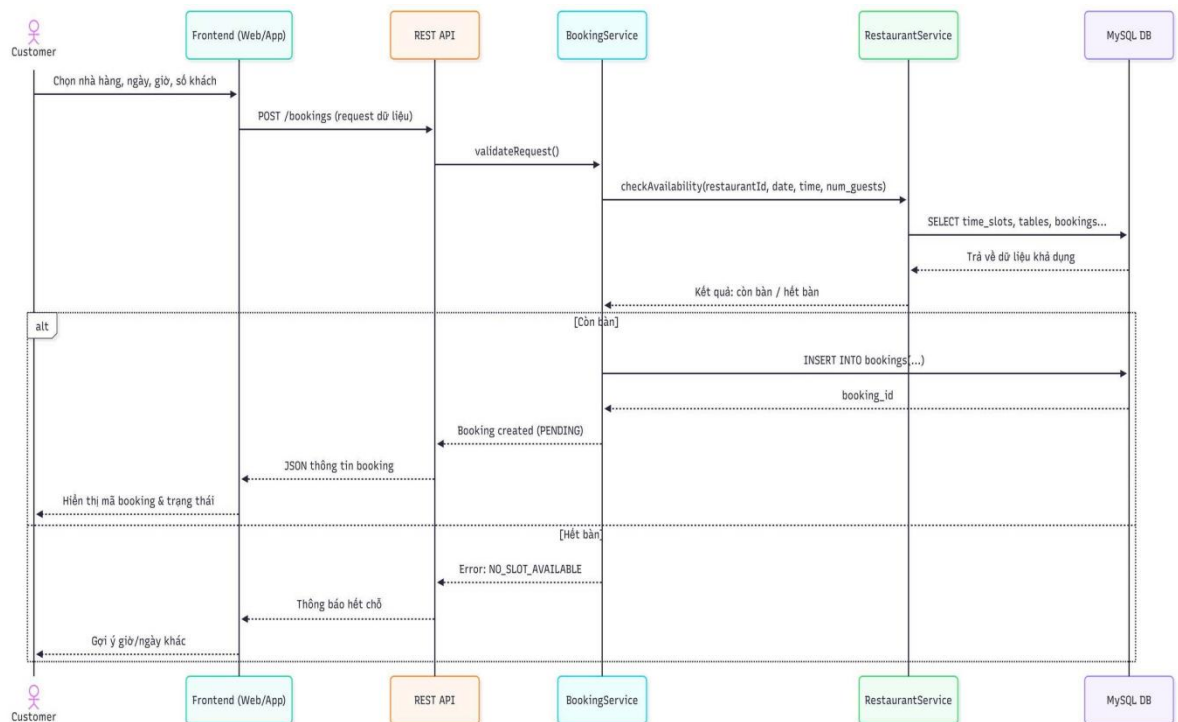
Booking —(Được chọn)→ TimeSlot

Restaurant —(Nhận đơn)→ Booking

User —(Nhận)→ Notification

### 3.4. Mô hình động

#### 3.4.1. Sơ đồ Sequence



Hình 4: Sơ đồ tuần tự

Sơ đồ Sequence mô tả trình tự tương tác giữa các đối tượng trong quá trình khách hàng đặt bàn ăn, bao gồm các bước chính:

##### 1. Các thành phần chính

- **Actor (Tác nhân):** Customer (Khách hàng) - người bắt đầu tương tác.
- **Lifelines (Đường đời):** Các hình chữ nhật ở trên cùng đại diện cho các hệ thống hoặc thành phần phần mềm tham gia vào quy trình:
  - Frontend (Web/App): Giao diện người dùng.
  - REST API: Cổng tiếp nhận yêu cầu.
  - BookingService: Dịch vụ xử lý logic đặt bàn.
  - RestaurantService: Dịch vụ quản lý thông tin nhà hàng.
  - MySQL DB: Cơ sở dữ liệu lưu trữ.

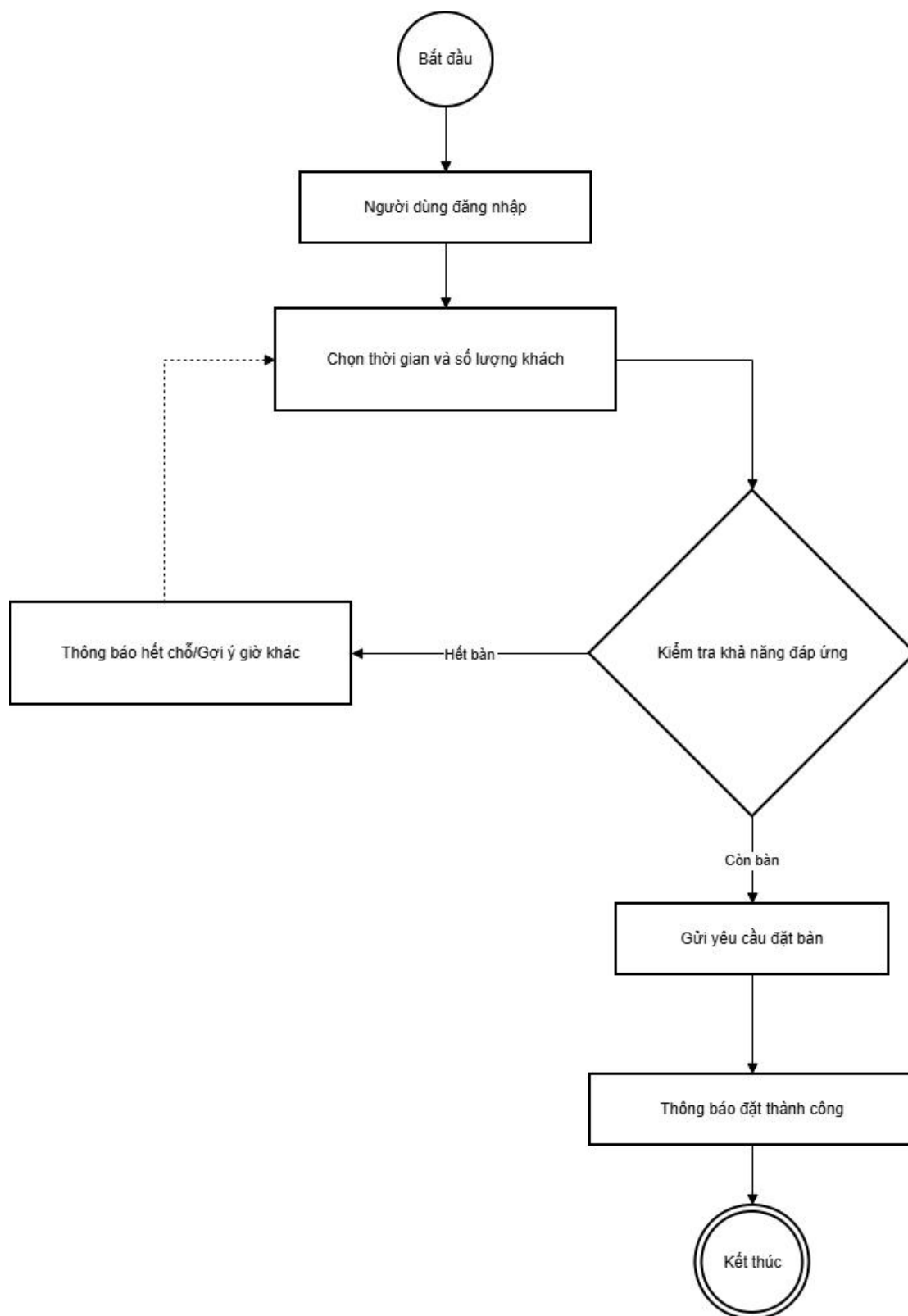
- **Messages (Thông điệp):** Các mũi tên (nét liền và nét đứt) thể hiện việc gọi hàm, gửi yêu cầu (request) hoặc trả về kết quả (response) theo trình tự thời gian từ trên xuống dưới.
- **Fragment (Khung cấu trúc):** Khung chữ nhật có nhãn alt (alternative). Đây là cấu trúc rẽ nhánh (tương tự như if/else trong lập trình).

## ***2. Luồng hoạt động (Workflow) trong hình:***

Biểu đồ mô tả luồng đi như sau:

- **Khởi tạo:** Khách hàng chọn thông tin và gửi yêu cầu đặt bàn (POST /bookings).
- **Xử lý:**
  - REST API xác thực yêu cầu (validateRequest).
  - BookingService gọi sang RestaurantService để kiểm tra bàn trống (checkAvailability).
  - RestaurantService truy vấn MySQL DB để xem dữ liệu bàn/giờ.
- **Rẽ nhánh (Khung alt):**
  - Trường hợp 1 (Còn bàn): Hệ thống chèn đơn đặt bàn vào Database (INSERT INTO bookings), tạo trạng thái PENDING và trả về thông tin vé đặt thành công cho khách hàng.
  - Trường hợp 2 (Hết bàn): Hệ thống trả về lỗi NO\_SLOT\_AVAILABLE và Frontend sẽ hiển thị thông báo hết chỗ, gợi ý giờ khác cho khách.

### ***3.4.2. Sơ đồ Activity***



*Hình 5: Sơ đồ hoạt động đặt bàn*

Mô tả chi tiết Sơ đồ Hoạt động (Activity Diagram) - Chức năng Đặt bàn

**1. Mục đích:** Sơ đồ mô tả trình tự các hành động và các quyết định diễn ra khi người dùng thực hiện quy trình đặt bàn trên hệ thống.

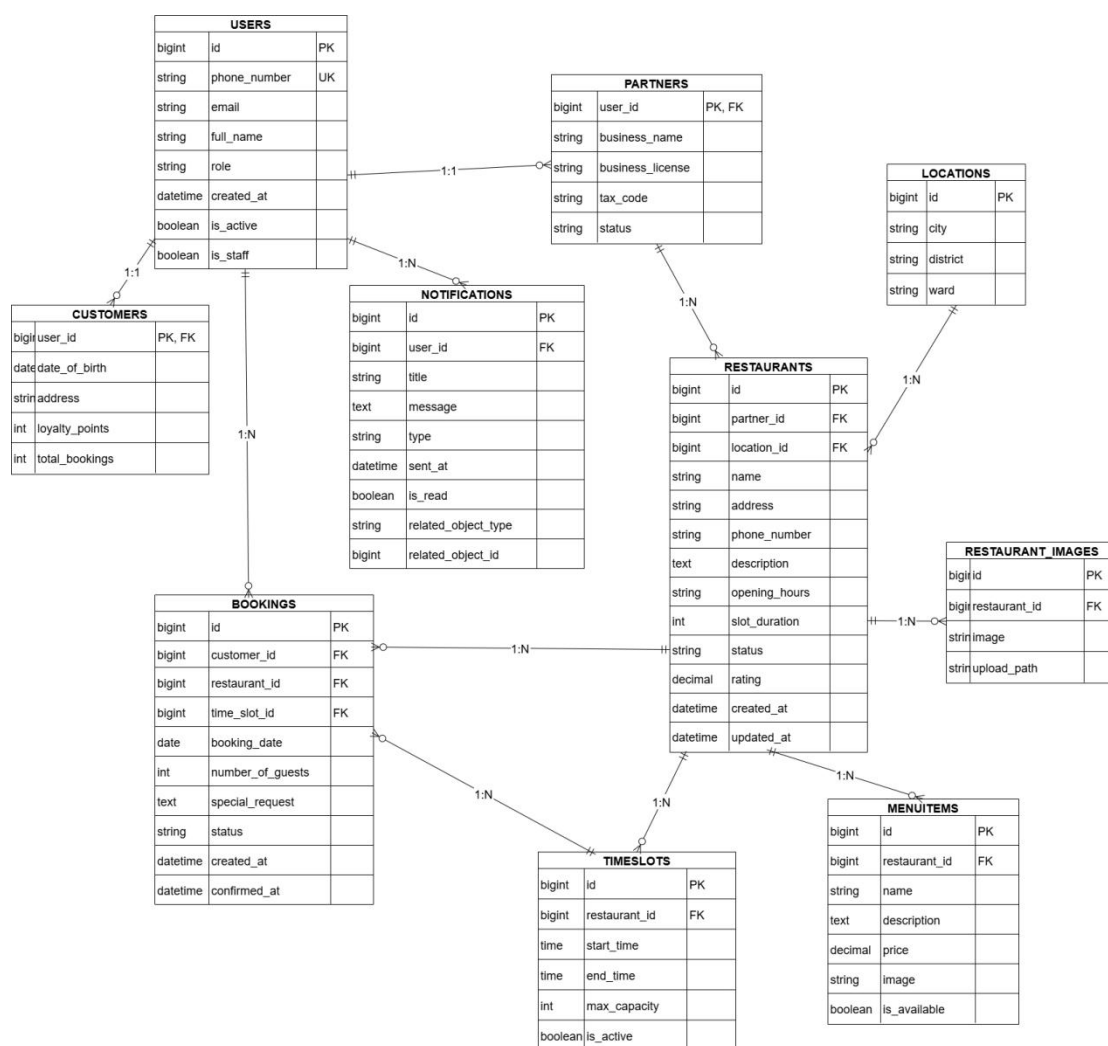
## 2. Các thành phần và luồng hoạt động:

- **Điểm bắt đầu (Initial Node):** Quy trình được kích hoạt khi người dùng truy cập vào giao diện đặt bàn.
- **Người dùng đăng nhập:** Hệ thống yêu cầu người dùng xác thực danh tính để gắn thông tin đặt bàn với tài khoản cụ thể.
- **Chọn thời gian và số lượng khách:** Người dùng nhập các thông tin cần thiết cho buổi hẹn (ngày, giờ, số người).
- **Kiểm tra khả năng đáp ứng (Decision Node):** Đây là điểm rẽ nhánh quan trọng của quy trình. Hệ thống sẽ kiểm tra dữ liệu trong kho dữ liệu:
  - **Trường hợp 1 - Hết bàn (Luồng phụ):** Nếu thời gian hoặc số lượng khách không thể đáp ứng, hệ thống sẽ hiển thị thông báo lỗi (hoặc gợi ý giờ khác) và điều hướng người dùng quay lại bước nhập thông tin.
  - **Trường hợp 2 - Còn bàn (Luồng chính):** Nếu thỏa mãn điều kiện, quy trình đi tiếp xuống bước gửi yêu cầu.
- **Gửi yêu cầu đặt bàn:** Hệ thống ghi nhận thông tin và xử lý giao dịch.
- **Thông báo thành công:** Hệ thống phản hồi lại cho người dùng xác nhận việc đặt bàn đã hoàn tất.
- **Điểm kết thúc (Final Node):** Quy trình đặt bàn kết thúc.



# CHƯƠNG 4: THIẾT KẾ DỮ LIỆU VÀ DÒNG DỮ LIỆU

## 4.1 Thiết kế ERD



Hình 6: Sơ đồ ERD

### Mô tả sơ đồ ERD

#### 1. Phân tích chi tiết các thực thể (Tables)

**A. Cụm Quản lý Người dùng (User Management)** Thiết kế sử dụng kỹ thuật Table-per-Type (Mỗi loại một bảng) để xử lý quan hệ kế thừa:

**USERS (Bảng cha):**

- Lưu trữ thông tin xác thực dùng chung: phone\_number (UK - Unique Key, đảm bảo số điện thoại là duy nhất), email, role, password (ẩn trong thực tế), is\_active.
- Đóng vai trò trung tâm kiểm soát đăng nhập.

#### ***CUSTOMERS (Bảng con):***

- Liên kết 1-1 với bảng USERS thông qua khóa chính user\_id (vừa là PK vừa là FK).
- Lưu thông tin riêng của khách: loyalty\_points (điểm thưởng), date\_of\_birth.

#### ***PARTNERS (Bảng con):***

- Liên kết 1-1 với bảng USERS qua user\_id.
- Lưu thông tin doanh nghiệp: business\_name, tax\_code (mã số thuế).

### **B. Cụm Quản lý Nhà hàng (Restaurant Management)**

#### ***RESTAURANTS:***

- Là bảng trung tâm của cụm này.
- Liên kết N-1 với PARTNERS (Một đối tác sở hữu nhiều nhà hàng).
- Liên kết N-1 với LOCATIONS (Một địa điểm/quận có nhiều nhà hàng).

#### ***LOCATIONS:***

- Được tách riêng để chuẩn hóa dữ liệu địa lý (city, district, ward). Việc này giúp hệ thống dễ dàng lọc nhà hàng theo khu vực (ví dụ: "Tìm tất cả quán ở Quận 1").

#### ***Các bảng chi tiết (Detail Tables):***

- MENUITEMS: Lưu thực đơn, liên kết N-1 với Nhà hàng.
- RESTAURANT\_IMAGES: Lưu thư viện ảnh, liên kết N-1 với Nhà hàng.
- TIMESLOTS: Lưu các khung giờ phục vụ (start\_time, end\_time, max\_capacity), liên kết N-1 với Nhà hàng.

### **C. Cụm Giao dịch Đặt bàn (Booking Transactions)**

#### ***BOOKINGS:***

- Là bảng quan trọng nhất, lưu trữ lịch sử giao dịch.

**Khóa ngoại (FK):**

- customer\_id: Ai đặt?
- restaurant\_id: Đặt quán nào?
- time\_slot\_id: Đặt khung giờ nào?
- Thuộc tính: status (trạng thái đơn), number\_of\_guests, booking\_date.

**D. Phụ trợ**

**NOTIFICATIONS:** Lưu thông báo hệ thống gửi tới USERS. Quan hệ 1-N (Một người dùng nhận nhiều thông báo).

**3. Giải thích các ký hiệu quan hệ (Relationships)**

Trong sơ đồ sử dụng ký hiệu Crow's Foot:

- Đường thẳng có 2 gạch dọc (||): Quan hệ 1-1 (One-to-One).
  - Ví dụ: USERS - CUSTOMERS. Một User chỉ ứng với một hồ sơ Customer duy nhất.
- Đường thẳng có chân chim (|< hoặc O<): Quan hệ 1-N (One-to-Many).
  - Ví dụ: PARTNERS - RESTAURANTS. Một Partner có thể có nhiều Restaurant (chân chim), nhưng một Restaurant chỉ thuộc về một Partner (gạch dọc).

**4.2 Chuyển sang mô hình quan hệ**

Quan hệ	Mô tả	Kiểu
USERS ↔ CUSTOMERS	1 người dùng = 1 khách hàng	1:1
USERS ↔ PARTNERS	1 người dùng = 1 đối tác	1:1
USERS → BOOKINGS	1 khách hàng có nhiều đặt bàn	1:N
USERS → NOTIFICATIONS	1 người dùng nhận nhiều thông báo	1:N
PARTNERS → RESTAURANTS	1 đối tác quản lý nhiều nhà hàng	1:N
LOCATIONS → RESTAURANTS	1 địa điểm có nhiều nhà hàng	1:N
RESTAURANTS → TIMESLOTS	1 nhà hàng có nhiều khung giờ	1:N
RESTAURANTS → MENUITEMS	1 nhà hàng có nhiều món ăn	1:N

RESTAURANTS → BOOKINGS	1 nhà hàng có nhiều đặt bàn	1:N
TIMESLOTS → BOOKINGS	1 khung giờ có nhiều đặt bàn	1:N

### 4.3 Thiết kế CSDL

#### ***Bảng dữ liệu: USERS (Người dùng)***

Tên trường (Field)	Kiểu dữ liệu (Data Type)	Ràng buộc (Key)	Mô tả
id	bigint	PK (Primary Key)	Mã định danh duy nhất của người dùng.
phone_number	string	UK (Unique Key)	Số điện thoại (Là thông tin duy nhất để đăng nhập/định danh).
email	string		Địa chỉ Email.
full_name	string		Họ và tên đầy đủ.
role	string		Vai trò người dùng (ví dụ: CUSTOMER, PARTNER, ADMIN).
created_at	datetime		Ngày giờ tạo tài khoản.
is_active	boolean		Trạng thái hoạt động (True: Đang hoạt động, False: Bị khóa).
is_staff	boolean		Đánh dấu có phải là nhân viên hệ thống/quản trị viên hay không.

#### ***Bảng: CUSTOMERS (Khách hàng)***

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	bigint	PK, FK	Khóa chính, đồng thời là khóa ngoại trỏ về USERS.id.
date_of_birth	date		Ngày sinh.
address	string		Địa chỉ khách hàng.
loyalty_points	int		Điểm tích lũy thành viên.
total_bookings	int		Tổng số lần đã đặt bàn.

**Bảng: PARTNERS (Đối tác/Chủ nhà hàng)**

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	bigint	PK, FK	Khóa chính, trỏ về USERS.id.
business_name	string		Tên doanh nghiệp/hộ kinh doanh.
business_license	string		Giấy phép kinh doanh.
tax_code	string		Mã số thuế.
status	string		Trạng thái duyệt đối tác (Pending, Approved...).

**Bảng: RESTAURANTS (Nhà hàng)**

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	PK	Mã nhà hàng.
partner_id	bigint	FK	Thuộc về đối tác nào (PARTNERS.user_id).
location_id	bigint	FK	Địa chỉ thuộc khu vực nào (LOCATIONS.id).
name	string		Tên nhà hàng.
address	string		Địa chỉ chi tiết (số nhà, đường).
phone_number	string		Số điện thoại liên hệ của quán.
description	text		Mô tả giới thiệu quán.
opening_hours	string		Giờ mở cửa hiển thị (text).
slot_duration	int		Thời lượng trung bình mỗi lượt ăn (phút).
status	string		Trạng thái quán (Open, Closed, Maintenance).
rating	decimal		Điểm đánh giá trung bình.
created_at	datetime		Ngày tạo.
updated_at	datetime		Ngày cập nhật thông tin gần nhất.

**Bảng: MENUITEMS (Thực đơn)**

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	PK	Mã món ăn.
restaurant_id	bigint	FK	Thuộc nhà hàng nào.
name	string		Tên món.
description	text		Mô tả món ăn.
price	decimal		Giá tiền.

image	string		Đường dẫn ảnh món ăn.
is_available	boolean		Còn món hay không.

**Bảng: TIMESLOTS (Khung giờ)**

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	PK	Mã khung giờ.
restaurant_id	bigint	FK	Thuộc nhà hàng nào.
start_time	time		Giờ bắt đầu nhận khách.
end_time	time		Giờ kết thúc.
max_capacity	int		Số bàn tối đa trong khung giờ này.
is_active	boolean		Kích hoạt khung giờ này không.

**Bảng: RESTAURANT\_IMAGES (Ảnh nhà hàng)**

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	PK	Mã hình ảnh.
restaurant_id	bigint	FK	Thuộc nhà hàng nào.
image	string		Tên file ảnh/URL.
upload_path	string		Đường dẫn lưu trữ.

**Bảng: BOOKINGS (Đơn đặt bàn)**

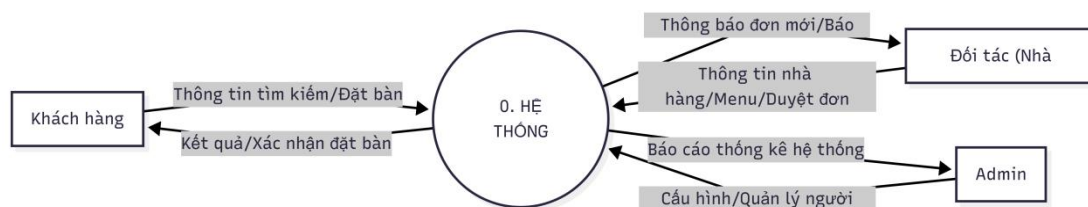
Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	PK	Mã đơn đặt bàn.
customer_id	bigint	FK	Khách hàng đặt (CUSTOMERS.user_id).
restaurant_id	bigint	FK	Đặt tại quán nào.
time_slot_id	bigint	FK	Đặt khung giờ nào.
booking_date	date		Ngày đặt bàn.
number_of_guests	int		Số lượng khách.
special_request	text		Yêu cầu đặc biệt (ghế trẻ em, view...).
status	string		Trạng thái (Pending, Confirmed, Cancelled).
created_at	datetime		Ngày tạo đơn.

confirmed_at	datetime		Ngày giờ đơn được xác nhận.
--------------	----------	--	-----------------------------

**Bảng: NOTIFICATIONS (Thông báo)**

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	PK	Mã thông báo.
user_id	bigint	FK	Người nhận thông báo (USERS.id).
title	string		Tiêu đề thông báo.
message	text		Nội dung thông báo.
type	string		Loại thông báo (System, Promo, Booking).
sent_at	datetime		Thời gian gửi.
is_read	boolean		Đã đọc hay chưa.
related_object_type	string		Loại đối tượng liên quan (ví dụ: 'booking').
related_object_id	bigint		ID của đối tượng liên quan (ví dụ: booking_id).

## 4.4 Sơ đồ DFD



Hình 7: Sơ đồ DFD level 0

### 1. Tổng quan

Sơ đồ mô tả hệ thống "Đặt bàn nhà hàng" như một hộp đen duy nhất (Process 0), thể hiện các luồng thông tin vào và ra giữa hệ thống với ba tác nhân chính: Khách hàng, Đối tác (Nhà hàng) và Quản trị viên (Admin).

### 2. Các thành phần chính

**Trung tâm (Process):**

**HỆ THỐNG:** Đại diện cho toàn bộ phần mềm/ứng dụng đặt bàn. Mọi quy trình xử lý chi tiết đều nằm ẩn bên trong vòng tròn này.

### **Các tác nhân ngoài (External Entities):**

- **Khách hàng:** Người dùng cuối sử dụng dịch vụ tìm kiếm và đặt bàn.
- **Đối tác (Nhà hàng):** Chủ nhà hàng hoặc quản lý, người cung cấp dịch vụ ăn uống.
- **Admin:** Người quản trị vận hành hệ thống.

### **3. Chi tiết các luồng dữ liệu (Data Flows)**

#### **Tương tác với Khách hàng:**

- **Vào (Input):** Thông tin tìm kiếm/Đặt bàn (Khách gửi yêu cầu tìm quán, chọn giờ, số người).
- **Ra (Output):** Kết quả/Xác nhận đặt bàn (Hệ thống trả về danh sách quán, hoặc vé xác nhận đặt chỗ thành công).

#### **Tương tác với Đối tác (Nhà hàng):**

- **Vào (Input):** Thông tin nhà hàng/Menu/Duyệt đơn (Đối tác cập nhật món ăn, giờ mở cửa và xác nhận chấp nhận/từ chối đơn của khách).
- **Ra (Output):** Thông báo đơn mới/Báo cáo (Hệ thống báo ting-ting khi có khách đặt, gửi báo cáo doanh thu).

#### **Tương tác với Admin:**

- **Vào (Input):** Cấu hình/Quản lý người dùng (Admin thiết lập tham số hệ thống, khóa/mở tài khoản).
- **Ra (Output):** Báo cáo thống kê hệ thống (Hệ thống tổng hợp số liệu toàn sàn để Admin theo dõi).



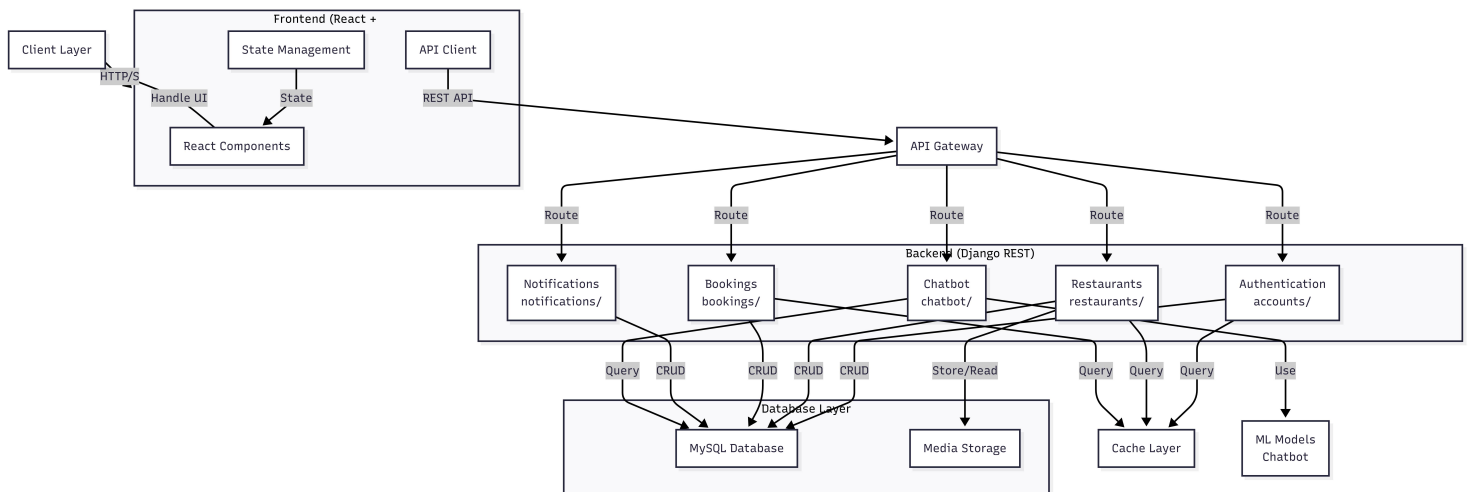
# CHƯƠNG 5: TRIỂN KHAI VÀ KIỂM THỬ

## 5.1 Kiến trúc ứng dụng

"Hệ thống website đặt bàn ăn được triển khai theo mô hình Client – Server với kiến trúc tách biệt hoàn toàn (Decoupled Architecture) giữa Frontend và Backend, giao tiếp thông qua chuẩn RESTful API.

Thay vì mô hình MVC truyền thống, hệ thống được tổ chức thành các tầng rõ rệt:

- **Frontend (Client Layer):** Sử dụng React để xây dựng giao diện, chịu trách nhiệm quản lý trạng thái (State Management) và gửi yêu cầu thông qua API Client.
- **Backend (Server Layer):** Sử dụng Django REST Framework, tiếp nhận yêu cầu từ API Gateway và điều hướng đến các module nghiệp vụ cụ thể (Authentication, Bookings, Restaurants, Chatbot, Notifications).
- **Database Layer:** Quản lý dữ liệu bền vững với MySQL, kết hợp với Cache Layer và Media Storage để tối ưu hóa hiệu suất và lưu trữ đa phương tiện."



Hình 8: Sơ đồ kiến trúc ứng dụng

## 5.2 Công nghệ sử dụng

Trong quá trình triển khai hệ thống, các công nghệ sau được sử dụng:

- ❖ Ngôn ngữ lập trình: Python 3.10+
- ❖ Framework Backend: Django 4.x, Django REST Framework
- ❖ Hệ quản trị cơ sở dữ liệu: PostgreSQL 13+
- ❖ Cơ chế xác thực: JWT (JSON Web Token)
- ❖ Frontend: ReactJS 18.x, TypeScript, HTML, CSS
- ❖ Thư viện giao tiếp HTTP: Axios
- ❖ Công cụ build frontend: Vite
- ❖ Công cụ phát triển: Visual Studio Code

### 5.3 Xử lý sự kiện và tương tác

Hệ thống xử lý các sự kiện chính sau:

- **Sự kiện đăng nhập:** Kiểm tra thông tin người dùng, xác thực và phân quyền.
- **Sự kiện đặt bàn:** Khi người dùng gửi yêu cầu đặt bàn, hệ thống kiểm tra tình trạng bàn trống và lưu thông tin đặt bàn.
- **Sự kiện quản lý đặt bàn:** Quản trị viên xác nhận hoặc từ chối yêu cầu đặt bàn.
- **Sự kiện hủy đặt bàn:** Người dùng có thể hủy đặt bàn trước thời gian quy định.

### 5.4 Giao diện người dùng

Hệ thống được thiết kế với giao diện hiện đại, thân thiện, phân chia rõ ràng cho ba nhóm đối tượng người dùng: Khách hàng (Customer), Đối tác nhà hàng (Partner) và Quản trị viên (Admin). Giao diện sử dụng tông màu chủ đạo là Cam (#f97316) kết hợp với nền trắng (cho Customer) và nền tối (Dark Mode cho Partner) để tạo điểm nhấn và trải nghiệm thị giác tốt nhất.

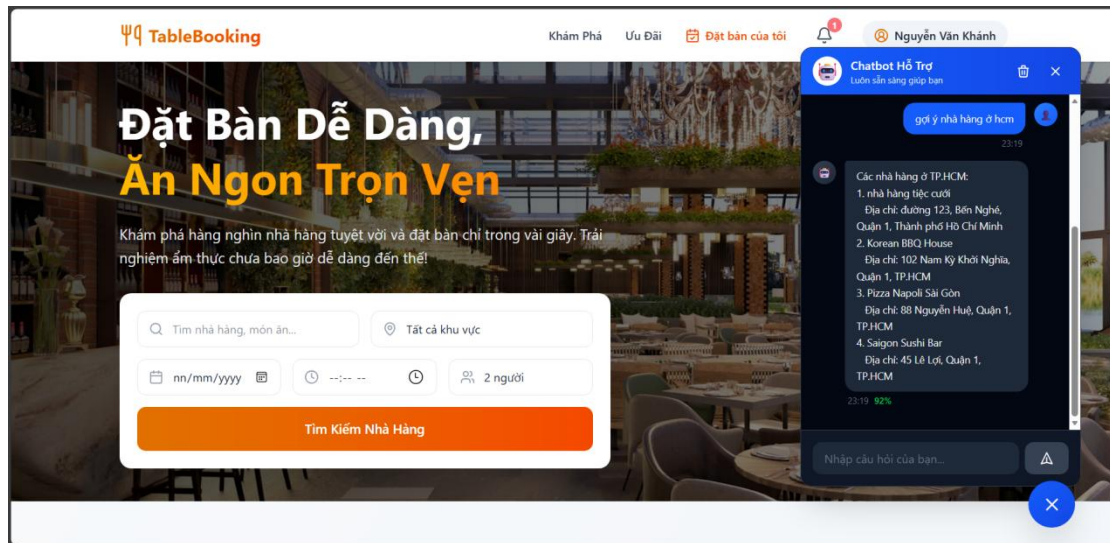
#### 5.4.1. Phân hệ Khách hàng (Customer Side)

Giao diện dành cho khách hàng tập trung vào sự đơn giản, giúp người dùng tìm kiếm và đặt bàn nhanh chóng.

##### *Trang chủ và Chatbot hỗ trợ:*

Giao diện trang chủ hiển thị thanh tìm kiếm nổi bật, cho phép lọc theo tên nhà hàng, món ăn hoặc khu vực.

Hệ thống tích hợp Chatbot thông minh (NLP) ở góc màn hình. Chatbot có khả năng hiểu ngôn ngữ tự nhiên tiếng Việt, giúp gợi ý nhà hàng theo nhu cầu (ví dụ: "gợi ý nhà hàng ở hcm") và hỗ trợ giải đáp thắc mắc tức thì cho người dùng.



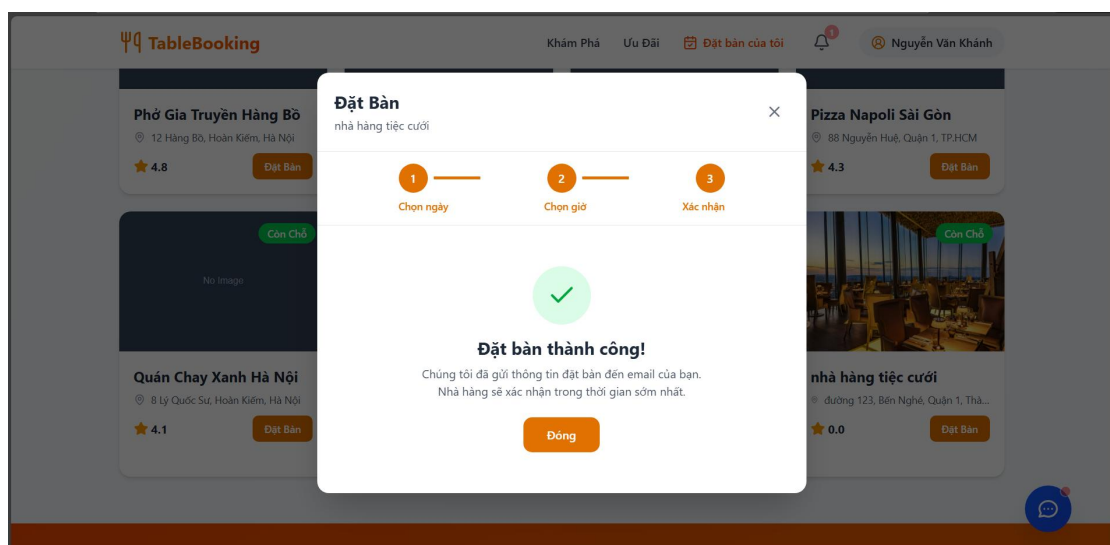
Hình 9: giao diện trang chủ

### Quy trình Đặt bàn:

Luồng đặt bàn được tối ưu hóa qua các bước rõ ràng:

Chọn ngày -> Chọn giờ -> Xác nhận.

Sau khi hoàn tất, hệ thống hiển thị thông báo (Modal) xác nhận đặt bàn thành công trực quan, đồng thời thông báo việc gửi email xác nhận đến người dùng.



Hình 10: giao diện quy trình đặt bàn

### ***Quản lý Đặt bàn cá nhân:***

Người dùng có thể theo dõi lịch sử đặt bàn tại trang "Đặt bàn của tôi".

Danh sách hiển thị đầy đủ thông tin: Tên nhà hàng, thời gian, số người và Trạng thái đơn (Chờ xác nhận, Đã hủy, Đã xác nhận) giúp người dùng chủ động nắm bắt thông tin.

*Hình 11: Quản lý Đặt bàn cá nhân*

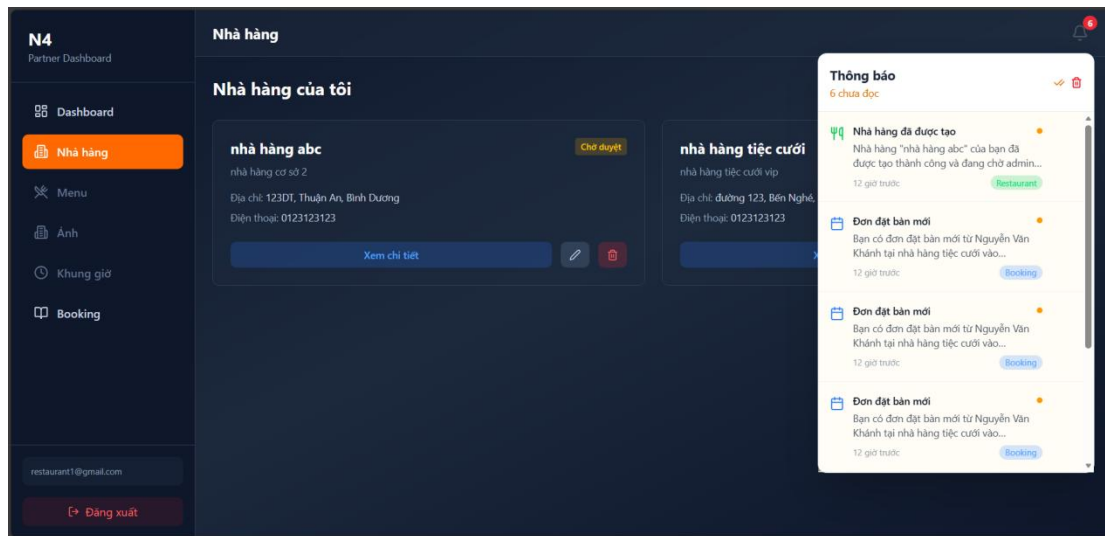
### ***5.4.2. Phân hệ Đối tác (Partner Dashboard)***

Giao diện dành cho Đối tác sử dụng chế độ Dark Mode chuyên nghiệp, giúp giảm mỏi mắt khi thao tác quản lý trong thời gian dài.

### ***Quản lý Nhà hàng và Thông báo:***

Dashboard cung cấp cái nhìn tổng quan về thông tin nhà hàng.

Hệ thống thông báo (Notification) cập nhật theo thời gian thực (Real-time) mỗi khi có đơn đặt bàn mới hoặc trạng thái nhà hàng được Admin phê duyệt, đảm bảo đối tác không bỏ lỡ thông tin quan trọng.



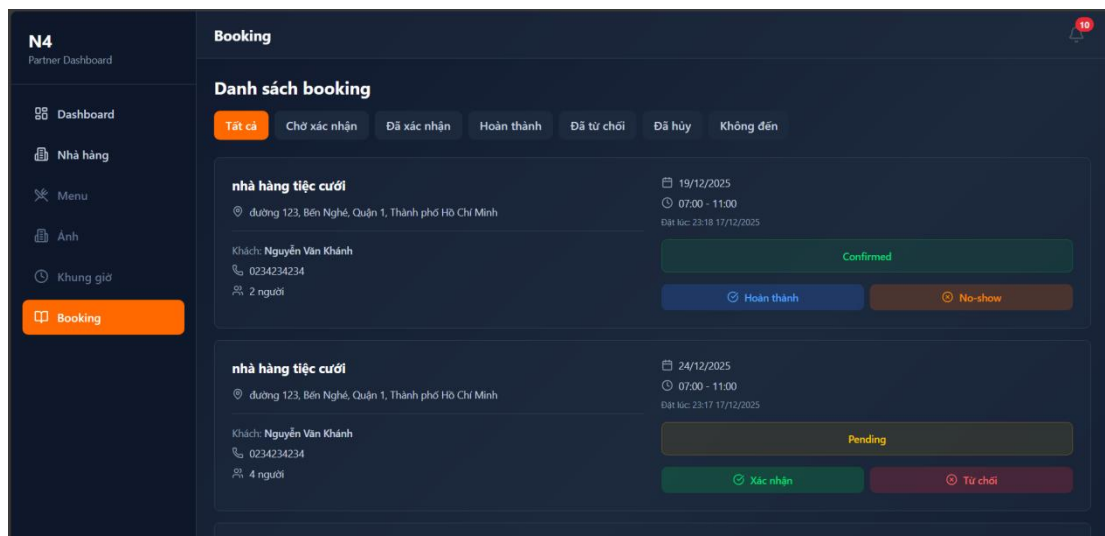
Hình 12: Quản lý Nhà hàng và Thông báo

### ***Quản lý Danh sách Booking:***

Đây là chức năng trọng tâm, cho phép đối tác xem danh sách các đơn đặt bàn sắp tới.

Giao diện cung cấp các bộ lọc trạng thái (Tất cả, Chờ xác nhận, Đã xác nhận...).

Các nút hành động (Action buttons) như: Xác nhận, Từ chối, Hoàn thành, No-show được bố trí trực quan, giúp quy trình xử lý đơn diễn ra nhanh chóng.



Hình 13: Quản lý Danh sách Booking

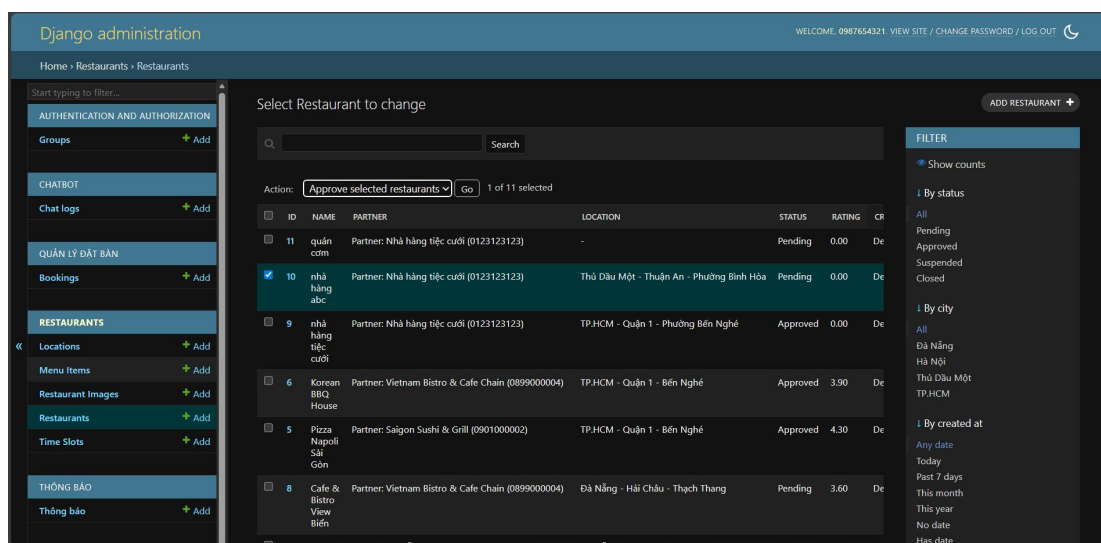
### 5.4.3. Phân hệ Quản trị viên (Admin Portal)

#### Trang quản trị hệ thống:

Sử dụng giao diện quản trị mạnh mẽ (nền tảng Django Administration).

Admin có toàn quyền kiểm soát dữ liệu hệ thống bao gồm: Quản lý người dùng (Users), Phê duyệt nhà hàng (Restaurants), Quản lý danh mục menu, và Cấu hình các tham số hệ thống.

Giao diện dạng bảng giúp dễ dàng tìm kiếm, lọc và chỉnh sửa dữ liệu hàng loạt.



Hình 14: Trang quản trị hệ thống

## 5.5 Kiểm thử và đánh giá

Quá trình kiểm thử được thực hiện theo phương pháp kiểm thử hộp đen (Black-box testing), tập trung vào việc xác minh các chức năng nghiệp vụ thông qua giao diện người dùng và đánh giá trải nghiệm người dùng (UI/UX).

### 5.5.1. Kết quả kiểm thử chức năng (Functional Testing)

Dựa trên các kịch bản sử dụng thực tế, hệ thống đã đáp ứng tốt các yêu cầu nghiệp vụ đề ra:

Chức năng	Kịch bản kiểm thử	Kết quả thực tế	Đánh giá
Tìm kiếm & Chatbot	Khách hàng nhập từ khóa tìm kiếm hoặc hỏi Chatbot gợi ý nhà hàng.	Chatbot phản hồi danh sách nhà hàng chính xác theo địa điểm (như hình minh họa gợi ý tại TP.HCM).	Đạt

<b>Quy trình Đặt bàn</b>	Khách hàng chọn khung giờ và xác nhận đặt bàn.	Hệ thống hiển thị Modal thông báo thành công và lưu đơn vào CSDL.	<b>Đạt</b>
<b>Xử lý Đặt bàn</b>	Đối tác nhận đơn mới và thao tác chuyển trạng thái (Xác nhận/Từ chối).	Trạng thái đơn hàng cập nhật tức thì bên phía Khách hàng (từ "Chờ xác nhận" sang "Đã xác nhận").	<b>Đạt</b>
<b>Thông báo</b>	Có đơn hàng mới được tạo.	Hệ thống gửi thông báo đẩy (Notification) đến Dashboard của Partner ngay lập tức.	<b>Đạt</b>
<b>Quản trị Admin</b>	Admin phê duyệt một nhà hàng mới đăng ký.	Nhà hàng chuyển sang trạng thái hoạt động và xuất hiện trên trang tìm kiếm của khách.	<b>Đạt</b>

### 5.5.2. Đánh giá Giao diện và Trải nghiệm (UI/UX Evaluation)

- ❖ **Tính thẩm mỹ:** Giao diện được thiết kế đồng bộ. Việc sử dụng màu cam làm điểm nhấn trên nền trắng/đen giúp hướng dẫn thị giác người dùng tốt, làm nổi bật các nút hành động quan trọng (như nút "Đặt bàn", "Booking").
- ❖ **Tính phản hồi:** Hệ thống cung cấp phản hồi rõ ràng cho người dùng sau mỗi tác vụ. Ví dụ: Khi đặt bàn thành công, một cửa sổ popup xuất hiện xác nhận; khi có tin nhắn mới, biểu tượng chuông thông báo hiển thị chấm đỏ.
- ❖ **Tính dễ sử dụng:** Bố cục thông tin rõ ràng, phân cấp hợp lý.
- ❖ **Với Khách hàng:** Các bước đặt bàn được chia nhỏ (Step-by-step), tránh gây rối mắt.
- ❖ **Với Đối tác:** Các thông tin quan trọng (Ngày giờ, Số khách, Số điện thoại) được hiển thị ngay trên thẻ booking, giúp ra quyết định xử lý nhanh.

### 5.5.3. Kết luận chung

Sau quá trình kiểm thử và đánh giá, hệ thống Website Đặt bàn nhà hàng hoạt động ổn định, các chức năng chính giữa ba phân hệ (Khách hàng - Đối tác - Admin) có sự liên kết dữ liệu chính xác và mượt mà. Giao diện người dùng đạt yêu cầu về tính thẩm mỹ và tiện dụng, sẵn sàng cho việc triển khai thực tế.

## CHƯƠNG 6: TỔNG KẾT

### 6.1 Kết quả đạt được

#### ❖ *Hệ thống Quản lý Nhà hàng*

Hệ thống đã hoàn thiện đầy đủ các chức năng CRUD cho nhà hàng, bao gồm quản lý thông tin cơ bản, menu items, upload và quản lý hình ảnh nhà hàng. Ngoài ra, hệ thống còn hỗ trợ quản lý khung giờ làm việc linh hoạt và quy trình phê duyệt nhà hàng với các trạng thái rõ ràng từ PENDING đến APPROVED và ACTIVE, đảm bảo kiểm soát chất lượng đối tác.

#### ❖ *Hệ thống Đặt bàn*

Chức năng đặt bàn được triển khai hoàn chỉnh với validation dữ liệu đầu vào, cho phép tạo booking, xác nhận hoặc từ chối đặt chỗ từ phía nhà hàng. Hệ thống hỗ trợ khách hàng hủy booking, theo dõi trạng thái đặt chỗ realtime, hiển thị các khung giờ còn trống và tự động tính toán tỷ lệ "no-show" để hỗ trợ quyết định quản lý của đối tác.

#### ❖ *Dashboard Quản lý*

Dashboard cung cấp cái nhìn tổng quan về hoạt động của nền tảng với các thống kê quan trọng: tổng số nhà hàng trong hệ thống, số lượng booking trong ngày và trong tuần, tổng số menu items. Dữ liệu được cập nhật realtime giúp quản trị viên và đối tác nắm bắt tình hình kinh doanh kịp thời.

#### ❖ *Hệ thống Thông báo*

Hệ thống thông báo đa kênh đã được tích hợp hoàn chỉnh, bao gồm email notifications cho các sự kiện quan trọng, push notifications cho thiết bị di động, in-app notifications với giao diện bell icon trực quan, và sử dụng Django Signals để trigger thông báo tự động dựa trên các sự kiện nghiệp vụ.

#### ❖ *Chatbot AI*

Chatbot được phát triển với khả năng nhận diện intent đa dạng (FAQ, tìm kiếm nhà hàng, đặt bàn, v.v.), trích xuất thông tin từ câu hỏi người dùng (địa điểm, thời gian, loại ẩm thực), nhận diện lời chào và lọc các câu hỏi không có nghĩa.



Hệ thống sử dụng Vietnamese SBERT cho semantic search và mô hình ViT5 để sinh câu trả lời tự nhiên bằng tiếng Việt.

### ❖ *Authentication & Authorization*

Hệ thống bảo mật được xây dựng dựa trên JWT token với cơ chế phân quyền rõ ràng theo vai trò (ADMIN, PARTNER, CUSTOMER). Các permission classes tùy chỉnh như IsPartnerOwner và IsActivePartner đảm bảo chỉ những người dùng có quyền mới có thể truy cập và thao tác với dữ liệu. Mật khẩu được mã hóa an toàn sử dụng các thuật toán industry-standard.

## 6.2 Hạn chế và hướng phát triển

### ✧ *Hạn chế hiện tại*

- ❖ **Chatbot:** Hệ thống chatbot hiện tại chỉ hỗ trợ tiếng Việt, hạn chế khả năng phục vụ khách hàng quốc tế. Knowledge base cần được cập nhật thường xuyên để đảm bảo tính chính xác của thông tin. Một số intent phức tạp chưa được xử lý hoàn toàn, có thể dẫn đến trải nghiệm người dùng chưa mượt mà trong một số tình huống.
- ❖ **Performance:** Chatbot đang chạy đồng bộ (synchronously), có thể trở thành điểm nghẽn khi lượng request tăng cao. Hệ thống chưa có caching cho các FAQ queries phổ biến, dẫn đến việc xử lý lặp lại không cần thiết. Quá trình xử lý hình ảnh chưa được tối ưu hóa về mặt hiệu năng.
- ❖ **UI/UX:** Dashboard hiện tại thiếu các biểu đồ và đồ thị chi tiết để visualize dữ liệu một cách trực quan. Giao diện mobile responsive chưa được hoàn thiện ở tất cả các màn hình. Animation và transitions còn cơ bản, chưa tạo được trải nghiệm người dùng mượt mà và hiện đại.
- ❖ **Infrastructure:** Hệ thống chưa có load balancing để phân phối tải đều giữa các server. Chiến lược backup database còn cơ bản, chưa có automated backup schedule và disaster recovery plan. Quy trình deployment vẫn thực hiện manual, dễ gây ra lỗi human error và tốn thời gian.

### ✧ *Hướng phát triển tương lai*

- ❖ Trong tương lai, hệ thống sẽ được cải tiến và mở rộng theo nhiều hướng nhằm nâng cao chất lượng dịch vụ và trải nghiệm người dùng. Về mặt hiệu năng, chatbot sẽ được chuyển sang xử lý bất đồng bộ sử dụng Celery, tích hợp Redis caching cho các truy vấn phổ biến, và tối ưu hóa xử lý hình ảnh với CDN. Giao

diện người dùng sẽ được nâng cấp với analytics dashboard đầy đủ biểu đồ, phát triển mobile app đa nền tảng, và cải thiện hệ thống push notification.

- ❖ Khả năng AI của chatbot sẽ được mở rộng để hỗ trợ đa ngôn ngữ (tiếng Anh, tiếng Trung), nâng cao độ chính xác trong việc trích xuất thông tin, và tích hợp sentiment analysis để phân tích đánh giá của khách hàng. Các tính năng mới sẽ được bổ sung như chương trình khách hàng thân thiết, tùy chỉnh cấu hình bàn linh hoạt, phân tích doanh thu chi tiết, và tích hợp với các cổng thanh toán phổ biến như VNPay và Stripe.
- ❖ Về mặt hạ tầng kỹ thuật, hệ thống sẽ được containerize với Docker, triển khai trên Kubernetes để dễ dàng scale, và xây dựng CI/CD pipeline tự động với GitHub Actions. Trong dài hạn, kiến trúc hệ thống có thể chuyển sang microservices để dễ dàng bảo trì và mở rộng, áp dụng event-driven architecture với Kafka, và phát triển các tính năng advanced như recommendation engine, social features, và business intelligence tools phục vụ ra quyết định chiến lược.

## TÀI LIỆU THAM KHẢO

- [1] Django Software Foundation, "Django Documentation," version 4.2, 2023. [Online]. Available: <https://docs.djangoproject.com/en/4.2/>
- [2] Python Software Foundation, "Python Documentation," version 3.11, 2023. [Online]. Available: <https://docs.python.org/3.11/>
- [3] React, "React Documentation," version 18, Meta Platforms, Inc., 2023. [Online]. Available: <https://react.dev/>
- [4] PostgreSQL Global Development Group, "PostgreSQL Documentation," version 15, 2023. [Online]. Available: <https://www.postgresql.org/docs/15/>
- [5] Django REST Framework, "Django REST Framework Documentation," version 3.14, 2023. [Online]. Available: <https://www.django-rest-framework.org/>
- [6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019, pp. 3982-3992.
- [7] H. V. Nguyen, H. T. Nguyen, and V. D. Lai, "ViT5: Pretrained Text-to-Text Transformer for Vietnamese Language Generation," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Online, 2021, pp. 136-142.
- [8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, Minnesota, 2019, pp. 4171-4186.