

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN



BÀI TẬP LỚN
XÂY DỰNG WEBSITE QUẢN LÝ KHÁCH SẠN

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM

SINH VIÊN: **NGUYỄN VĂN HIẾU**
MÃ LỚP: **12522W.2KS**
HƯỚNG DẪN: **VŨ XUÂN THẮNG**

HƯNG YÊN – 2025

NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan bài tập lớn “Xây dựng website quản lý khách sạn” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của Thầy Vũ Xuân Thành

Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong bài tập lớn và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

Hưng Yên, ngày ... tháng ... năm

SINH VIÊN

LỜI CẢM ƠN

Để có thể hoàn thành bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện bài tập lớn môn học này.

Đặc biệt em xin chân thành cảm ơn Thầy Vũ Xuân Thắng đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn!

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ	7
DANH MỤC CÁC BẢNG.....	8
DANH MỤC CÁC HÌNH VẼ.....	9
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	11
1.1. Lý do chọn đề tài	11
1.2. Mục tiêu của đề tài.....	11
1.2.1 Mục tiêu tổng quát	11
1.2.2 Mục tiêu cụ thể	11
1.3. Giới hạn và phạm vi của đề tài	12
1.3.1 Đối tượng nghiên cứu	12
1.3.2 Phạm vi nghiên cứu	12
1.4. Nội dung thực hiện	12
1.5. Phương pháp tiếp cận.....	12
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	14
2.1. Quy trình phát triển phần mềm.....	14
2.2. Phương pháp phân tích thiết kế hướng đối tượng	16
2.3. Thiết kế và lập trình font-end	22
2.4. Tổng quan về lập trình back end	23
2.5. Công nghệ thao tác dữ liệu	24
2.6. Mô hình thao tác dữ liệu 3 layer.....	25
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	27
3.1 Đặc tả yêu cầu phần mềm	27
3.1.1 Các yêu cầu chức năng	27
3.1.2 Biểu đồ lớp thực thể.....	47

3.1.3 Các yêu cầu phi chức năng	58
3.2 Thiết kế hệ thống	60
3.2.1 Thiết kế cơ sở dữ liệu	60
3.2.2 Thiết kế lớp đối tượng	68
3.2.3 Thiết kế giao diện	76
CHƯƠNG 4: TRIỂN KHAI WEBSITE.....	84
4.1 Triển khai các chức năng cho phân hệ người dùng	84
4.1.1 Trang chủ	84
4.1.2 Trang xem phòng	88
4.1.3 Trang xem dịch vụ	91
4.2 Triển khai các chức năng cho phân hệ quản trị nội dung	95
4.2.1 Trang quản lý nhân viên	95
4.3 Kiểm thử và triển khai ứng dụng	99
4.3.1 Kiểm thử	99
4.3.2 Đóng gói ứng dụng	100
4.3.3 Triển khai ứng dụng	101
KẾT LUẬN	102
TÀI LIỆU THAM KHẢO	103

DANH MỤC CÁC THUẬT NGỮ

STT	Từ viết tắt	Cụm từ tiếng anh	Điễn giải
1	MVC	Model View Control	Mẫu thiết kế phần mềm
.....

DANH MỤC CÁC BẢNG

Bảng 3.1 Bảng danh sách các lớp đối tượng	48
Bảng 3.2 Bảng LoaiPhong	60
Bảng 3.4 Bảng Phong.....	61
Bảng 3.5 Bảng LoaiThietBi	61
Bảng 3.6 Bảng ThietBi.....	61
Bảng 3.7 Bảng LoaiDichVu.....	62
Bảng 3.8 Bảng DichVu	62
Bảng 3.9 Bảng PhieuDichVu	62
Bảng 3.10 Bảng TaiKhoan	63
Bảng 3.11 Bảng KhachHang.....	63
Bảng 3.12 Bảng LoaiSuKien.....	64
Bảng 3.13 Bảng SuKien	64
Bảng 3.14 Bảng PhieuDatPhong.....	65
Bảng 3.15 Bảng ChiTietPhieuDatPhong	65
Bảng 3.16 Bảng PhieuThuePhong	66
Bảng 3.17 Bảng ChiTietPhieuThuePhong	66
Bảng 3.18 Bảng NhanVien	67
Bảng 3.19 Bảng HoaDon	68

DANH MỤC CÁC HÌNH VẼ

Hình 2.1 Biểu đồ Use case	17
Hình 2.2 Biểu đồ lớp	18
Hình 2.3 Biểu đồ tuần tự	18
Hình 2.4 Biểu đồ trạng thái	19
Hình 2.5 Biểu đồ trạng thái	20
Hình 2.6 Biểu đồ thành phần.....	21
Hình 2.7 Biểu đồ triển khai	21
Hình 2.8 Mô hình 3-layer.....	25
Hình 3.1 Biểu đồ use case tổng quát.....	33
Hình 3.2 Biểu đồ lớp thực thể	47
Hình 3.3 Lược đồ ER Diagram	60
Hình 3.4 Biểu đồ VOPC của use case quản lý phòng.....	69
Hình 3.5 Biểu đồ VOPC của use case quản lý phiếu dịch vụ.....	70
Hình 3.6 Biểu đồ VOPC của use case quản lý khách hàng	70
Hình 3.7 Biểu đồ VOPC của use case quản lý nhân viên	71
Hình 3.8 Biểu đồ tuần tự đăng nhập	71
Hình 3.9 Biểu đồ tuần tự đăng ký	72
Hình 3.10 Trang chủ	76
Hình 3.11 Trang danh sách phòng	77
Hình 3.12 Trang dịch vụ	77
Hình 3.13 Trang đặt phòng	78
Hình 3.14 Trang thông tin khách sạn	78
Hình 3.15 Trang liên hệ	79
Hình 3.16 Trang thông tin cá nhân.....	79
Hình 3.17 Trang thanh toán	80
Hình 3.18 Trang đánh giá.....	80
Hình 3.19 Trang thống kê	81
Hình 3.20 Trang quản lý phòng	81

Hình 3.21 Trang quản lý dịch vụ	82
Hình 3.22 Trang đăng kí	82
Hình 3.23 Trang đăng nhập.....	83
Hình 4.1 Source code GitHub	101

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Nước ta đang bước vào thời kì công nghiệp hóa – hiện đại hóa nhằm đưa Việt Nam trở thành nước công nghiệp văn minh, hiện đại. Ngành du lịch là một trong những ngành có đóng góp to lớn trong công cuộc cách mạng đó và được xem là ngành “kinh tế mũi nhọn” trong thời đại hiện nay. Nhờ được sự quan tâm của nhà nước, có được nguồn vốn khổng lồ từ nước ngoài, nhiều khu du lịch được hình thành và phát triển ở Việt Nam, từ đó khách sạn mọc lên cũng ngày càng nhiều.

Khách sạn đóng vai trò quan trọng trong ngành du lịch và dịch vụ lưu trú. Đây không chỉ là nơi cung cấp chỗ ở cho du khách mà còn là điểm đến quan trọng để có thể trải nghiệm văn hóa địa phương, thư giãn, nghỉ ngơi sau hành trình dài.

Với sự phát triển mạnh mẽ và sự cạnh tranh gay gắt của ngành du lịch, việc quản lý khách sạn cũng trở nên phức tạp và đòi hỏi sự chuyên nghiệp cao. Việc nâng cao chất lượng dịch vụ, tối ưu hóa hoạt động và quản lý hiệu quả trong khách sạn là vô cùng cần thiết. Từ đó giúp tăng sự hài lòng của khách hàng, giúp khách sạn tăng doanh thu, lợi nhuận.

Hệ thống quản lý khách sạn hiện tại thường gặp phải nhiều hạn chế như: thiếu tính linh hoạt trong quản lý, sự phụ thuộc quá nhiều vào con người, thiếu thông tin và dữ liệu chính xác để ra quyết định, và khó khăn trong việc tối ưu hóa hoạt động và tài nguyên. Điều này đặt ra yêu cầu cấp thiết cho việc nghiên cứu và áp dụng các giải pháp công nghệ thông tin để cải thiện hiệu suất và hiệu quả của quản lý khách sạn.

1.2. Mục tiêu của đề tài

1.2.1 Mục tiêu tổng quát

Mục tiêu chung của đề tài "Xây dựng website quản lý khách sạn" là tạo ra một hệ thống quản lý hoàn chỉnh và hiệu quả nhằm giải quyết các vấn đề trong lĩnh vực quản lý và vận hành khách sạn.

1.2.2 Mục tiêu cụ thể

- Phát triển một hệ thống thông minh, linh hoạt có khả năng điều chỉnh linh hoạt theo yêu cầu của khách hàng

- Tối ưu hóa việc sử dụng tài nguyên như phòng, nhân viên, dịch vụ,... nhằm giảm thiểu lãng phí, từ đó tăng cường hiệu suất hoạt động của khách sạn

- Nâng cao chất lượng dịch vụ và trải nghiệm của khách hàng thông qua việc cung cấp thông tin chi tiết, đáng tin cậy, dễ tiếp cận, đồng thời tối ưu hóa quy trình đặt phòng, nhận phòng và thanh toán

- Tăng cường quản lý nhân sự, phát triển các giải pháp quản lý nhân sự hiệu quả

- Xây dựng một hệ thống phần mềm quản lý hiện đại dễ dàng sử dụng, giúp việc quản lý và vận hành khách sạn hiệu quả và linh hoạt hơn.

1.3. Giới hạn và phạm vi của đề tài

1.3.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu: Hệ thống quản lý và vận hành của khách sạn.

Khách thể nghiên cứu: Khách hàng đã đến khách sạn, các nhà cung cấp dịch vụ và các cán bộ quản lý khách sạn.

1.3.2 Phạm vi nghiên cứu

Phạm vi nghiên cứu trong vòng 1 tháng thu thập số liệu thứ cấp và trong thời gian 2 tháng thực hiện nghiên cứu. Đề tài "Quản lý Khách sạn" không chỉ mang lại giá trị nghiên cứu khoa học mà còn cung cấp các giải pháp thực tiễn để cải thiện hoạt động kinh doanh trong ngành dịch vụ lưu trú và du lịch.

1.4. Nội dung thực hiện

Áp dụng quy trình phát triển phần mềm và vận dụng các kiến thức, kỹ năng tổng hợp từ các học phần đã học để xây dựng nên đề tài “Xây dựng website quản lý khách sạn” sẽ thực hiện những nội dung sau:

- Khảo sát, phân tích hệ thống.
- Thiết kế đặc tả các chức năng của hệ thống.
- Xây dựng cơ sở dữ liệu.
- Thiết kế và đặc tả giao diện.
- Lập trình hệ thống.
- Triển khai thực nghiệm hệ thống

1.5. Phương pháp tiếp cận

Trong quá trình thực hiện đề tài, em đã áp dụng các phương pháp sau:

- Phương pháp quan sát thực tiễn: quan sát qua các nhân viên đang làm việc, thông qua việc trực tiếp thuê phòng tại khách sạn. Khảo sát, phỏng vấn quản lí, nhân viên về quy trình làm việc.
- Phương pháp thu thập thông tin: thu thập thông tin từ việc quan sát các website liên quan đến khách sạn, app trên điện thoại.
- Phương pháp phân tích, so sánh: phân tích kết quả thu được cùng những thông tin thu thập được, từ đó đưa ra ý tưởng xây dựng, thiết kế hệ thống phần mềm sau này.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Quy trình phát triển phần mềm

Quy trình phát triển phần mềm là một tập hợp các bước được thực hiện để tạo ra phần mềm. Quy trình phát triển phần mềm giúp đảm bảo rằng phần mềm được phát triển một cách hiệu quả và đáp ứng được nhu cầu của người dùng.

Chúng ta nên áp dụng quy trình phát triển phần mềm là vì:

- Quy trình phát triển phần mềm giúp xác định rõ ràng các mục tiêu, phạm vi, nguồn lực, thời gian và chi phí của dự án, giảm thiểu rủi ro và tăng khả năng thành công.
- Quy trình phát triển phần mềm giúp cải thiện giao tiếp và hợp tác giữa các bên liên quan, như khách hàng, nhà phát triển, người dùng và nhà quản lý.
- Quy trình phát triển phần mềm giúp áp dụng các tiêu chuẩn, kỹ thuật và công cụ chuyên nghiệp để tạo ra các sản phẩm phần mềm chất lượng cao, an toàn và bảo mật.
- Quy trình phát triển phần mềm giúp theo dõi và kiểm soát tiến độ, chất lượng và chi phí của dự án, đánh giá hiệu suất và đưa ra các biện pháp khắc phục khi cần.
- Quy trình phát triển phần mềm giúp duy trì và nâng cấp phần mềm theo thời gian, đáp ứng nhu cầu thay đổi của khách hàng và thị trường.

Một quy trình phát triển phần mềm thông thường sẽ gồm 6 giai đoạn chính sau:

- Lên kế hoạch và phân tích yêu cầu - Analysis.
- Thiết kế phần mềm - Design.
- Tiến hành lập trình - Development.
- Kiểm thử - Testing.
- Triển khai - Deployment.
- Bảo trì - Maintenance.

Bước 1: Lên kế hoạch và phân tích yêu cầu

Đây là giai đoạn đầu tiên và quan trọng nhất trong quy trình phát triển phần mềm. Ở giai đoạn này, nhà phát triển sẽ tiến hành nghiên cứu về nhu cầu, mong muốn, vấn đề của khách hàng và người dùng. Nhà phát triển sẽ xác định được mục tiêu, phạm vi, ngân sách, thời gian và các ràng buộc của dự án. Nhà phát triển cũng sẽ xây dựng được các tài liệu yêu cầu (requirement documents) để làm cơ sở cho các giai đoạn tiếp theo.

Bước 2: Thiết kế phần mềm

Ở giai đoạn này, nhà phát triển sẽ dựa vào các tài liệu yêu cầu để thiết kế ra kiến trúc, giao diện và các thành phần của phần mềm. Thiết kế phần mềm sẽ giúp nhà phát triển hiểu được cách thức hoạt động, tính năng và chức năng của sản phẩm. Thiết kế phần mềm cũng sẽ giúp nhà phát triển ước lượng được nguồn lực, công cụ và công nghệ cần thiết cho việc lập trình. Kết quả của giai đoạn này là một tài liệu thiết kế phần mềm (SDD) hoặc một tài liệu đặc tả thiết kế (SDD).

Bước 3: Tiến hành lập trình

Giai đoạn này, nhà phát triển sẽ bắt tay vào viết mã nguồn cho các thành phần của phần mềm theo thiết kế đã có. Lập trình là giai đoạn chiếm nhiều thời gian và công sức nhất trong quy trình phát triển phần mềm. Lập trình cũng là giai đoạn có thể gặp nhiều lỗi và sai sót nhất. Do đó, nhà phát triển cần tuân thủ các chuẩn mực lập trình (coding standards), áp dụng các kỹ thuật lập trình tốt (best practices) và sử dụng các công cụ hỗ trợ (tools) để giảm thiểu lỗi và tăng hiệu quả.

Bước 4: Kiểm thử

Giai đoạn kiểm thử, nhà phát triển sẽ kiểm tra chất lượng của phần mềm bằng cách thực hiện các loại kiểm thử khác nhau, như kiểm thử đơn vị (unit testing), kiểm thử tích hợp (integration testing), kiểm thử hệ thống (system testing), kiểm thử chấp nhận (acceptance testing), kiểm thử hiệu năng (performance testing), kiểm thử bảo mật (security testing) và kiểm thử khả năng sử dụng (usability testing).

Mục tiêu của giai đoạn này là phát hiện, sửa chữa các lỗi, sai sót, thiếu sót và không phù hợp của sản phẩm so với yêu cầu và thiết kế. Kiểm thử sẽ đảm bảo cho sản phẩm hoạt động ổn định, an toàn và hiệu quả. Kết quả của giai đoạn này là các báo cáo kiểm thử và các phiên bản sửa lỗi (bug fixes) của phần mềm.

Bước 5: Triển khai

Ở giai đoạn này, nhà phát triển sẽ chuyển giao sản phẩm cho khách hàng và người dùng. Sản phẩm sẽ được cài đặt, cấu hình và vận hành trên môi trường thực tế. Nhà phát triển sẽ cung cấp các hướng dẫn, tài liệu và hỗ trợ kỹ thuật cho khách hàng và người dùng. Nhà phát triển cũng sẽ thu thập phản hồi và đánh giá của khách hàng và người dùng về sản phẩm.

Bước 6: Bảo trì

Đây là giai đoạn cuối cùng và kéo dài nhất trong quy trình phát triển phần mềm. Ở giai đoạn này, nhà phát triển sẽ tiếp tục theo dõi, duy trì và cập nhật sản phẩm để đáp ứng nhu cầu thay đổi của khách hàng và người dùng. Nhà phát triển sẽ sửa chữa các lỗi, nâng cấp các tính năng, tăng cường bảo mật và hiệu suất của sản phẩm.

Các mô hình sử dụng cho quy trình phát triển phần mềm:

- Mô hình chữ V - V Model
- Mô hình thác nước - Waterfall Model
- Mô hình Agile - Agile Model
- Mô hình xoắn ốc - Boehm Spiral Model
- Mô hình Scrum - Scrum Model

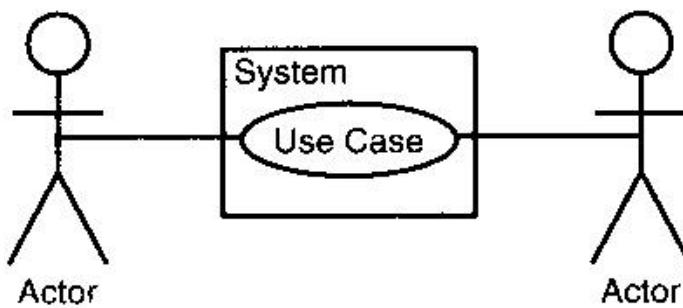
2.2. Phương pháp phân tích thiết kế hướng đối tượng

Ngôn ngữ mô hình hoá thống nhất (Unified Modeling Language – UML) là một ngôn ngữ để biểu diễn mô hình theo hướng đối tượng với chủ đích là:

- Mô hình hoá các hệ thống sử dụng các khái niệm hướng đối tượng.

- Thiết lập một kết nối từ nhận thức của con người đến các sự kiện cần mô hình hóa.
- Giải quyết vấn đề về mức độ thừa kế trong các hệ thống phức tạp, có nhiều ràng buộc khác nhau.
- Tạo ngôn ngữ mô hình hóa có thể sử dụng được bởi người và máy.

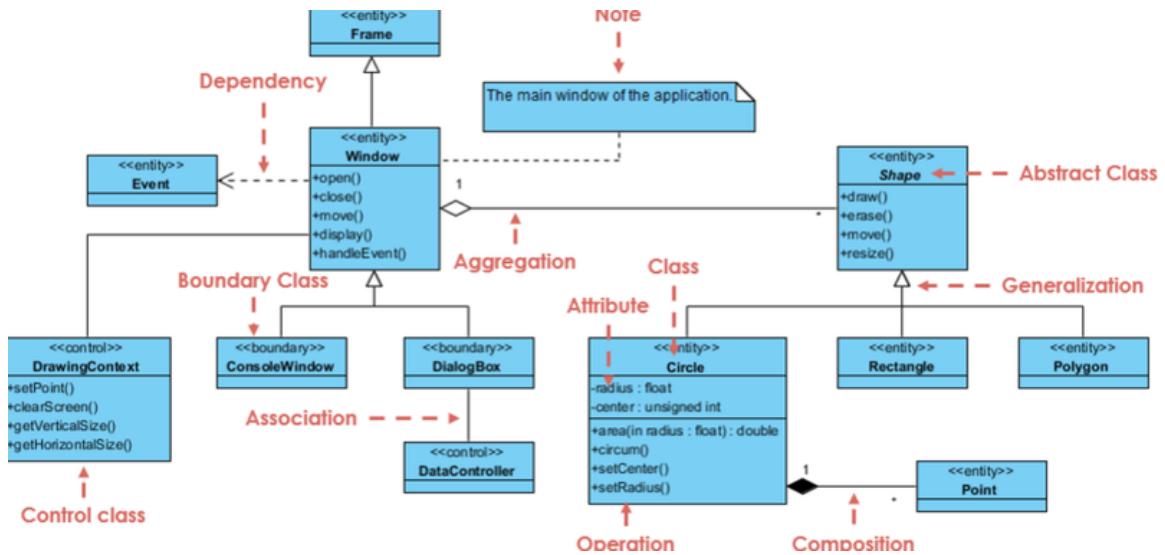
➤ **Biểu đồ Use Case (Use Case Diagram)**



Hình 2.1 Biểu đồ Use case

Biểu đồ Use case biểu diễn sơ đồ chức năng của hệ thống. Từ tập yêu cầu của hệ thống, biểu đồ use case sẽ phải chỉ ra hệ thống cần thực hiện điều gì để thỏa mãn các yêu cầu của người dùng hệ thống đó. Đi kèm với biểu đồ use case là các kịch bản.

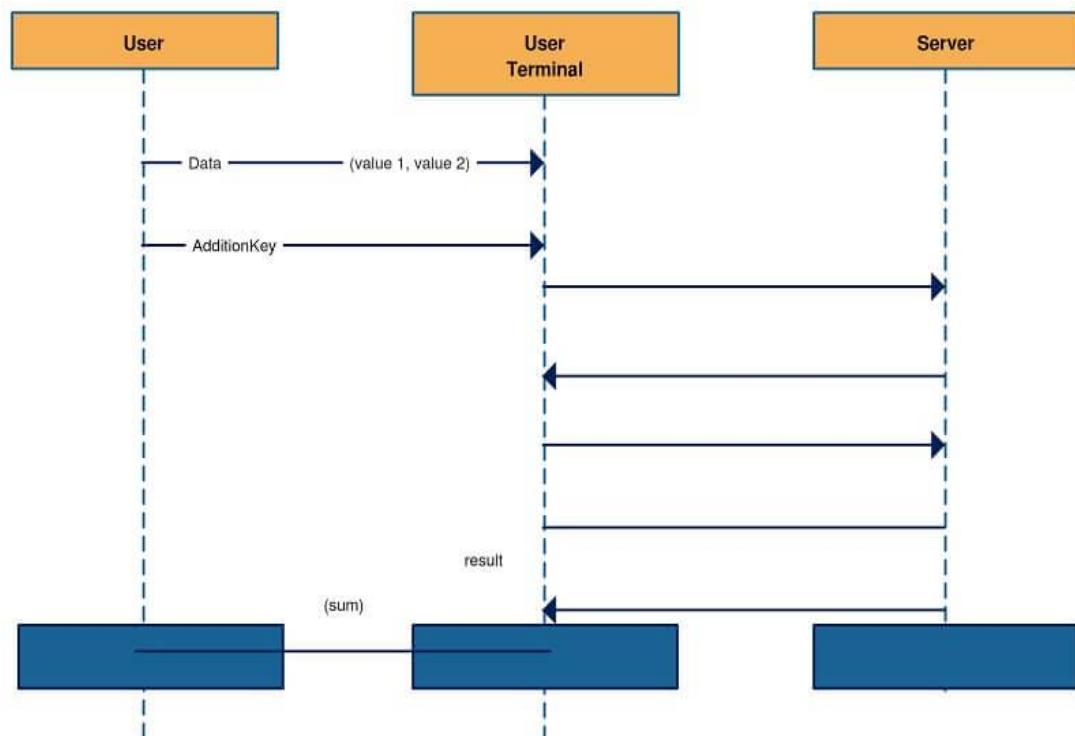
➤ **Biểu đồ lớp (Class Diagram)**



Hình 2.2 Biểu đồ lớp

Biểu đồ lớp chỉ ra các lớp đối tượng trong hệ thống, các thuộc tính và phương thức của từng lớp và các mối quan hệ giữa những lớp đó.

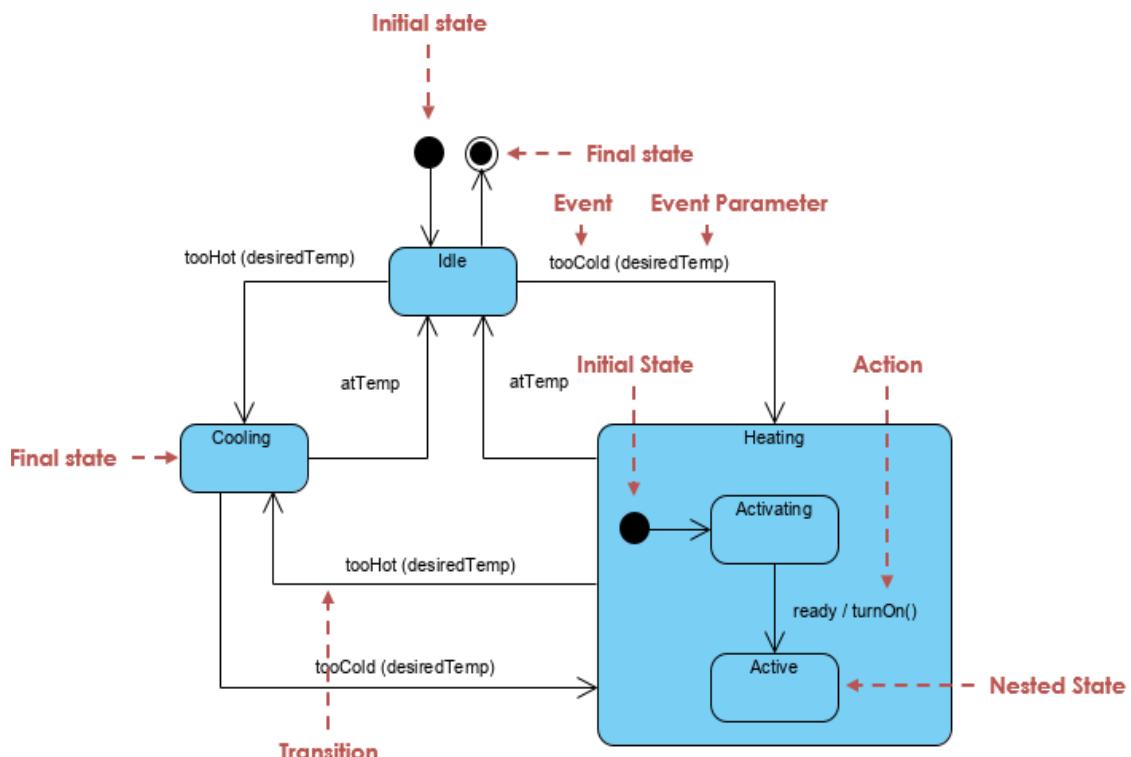
➤ *Biểu đồ tuần tự (Sequence Diagram)*



Hình 2.3 Biểu đồ tuần tự

Biểu đồ tuần tự là biểu đồ dùng để xác định các trình tự diễn ra sự kiện của một nhóm đối tượng nào đó. Nó miêu tả chi tiết các thông điệp được gửi và nhận giữa các đối tượng đồng thời cũng chú trọng đến việc trình tự về mặt thời gian gửi và nhận các thông điệp đó.

➤ *Biểu đồ trạng thái (State Machine Diagram)*

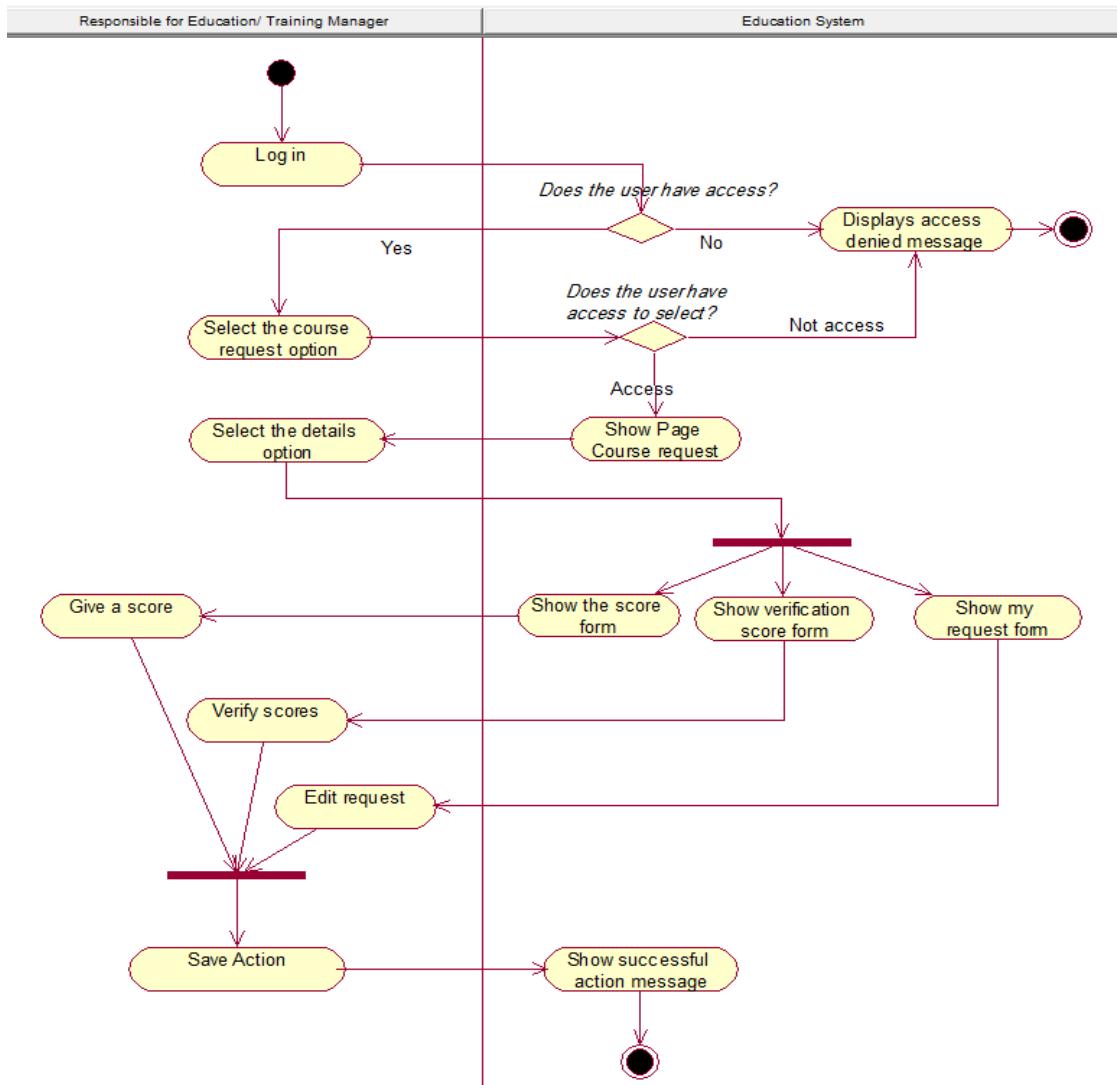


Hình 2.4 Biểu đồ trạng thái

Biểu đồ trạng thái là dạng biểu đồ mô tả các trạng thái có thể có và sự chuyển đổi giữa các trạng thái đó khi có các sự kiện tác động của một đối tượng.

Đối với các đối tượng có nhiều trạng thái thì biểu đồ trạng thái là sự lựa chọn tốt nhất giúp chúng ta có thể hiểu rõ hơn về hệ thống.

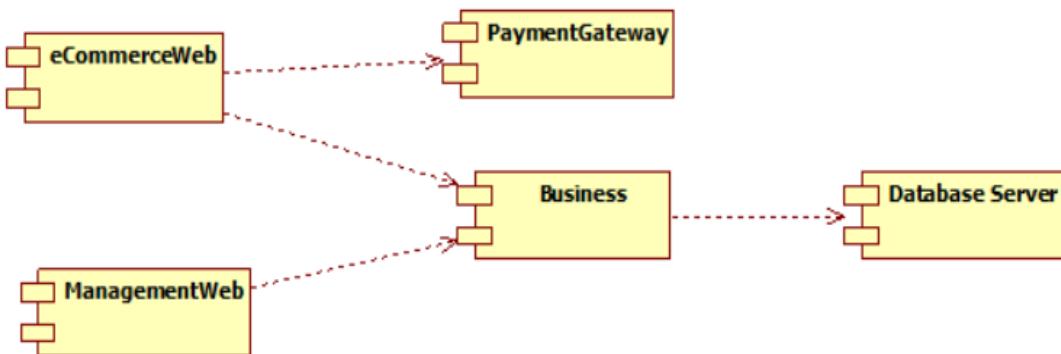
➤ *Biểu đồ hoạt động (Activity Diagram)*



Hình 2.5 Biểu đồ trạng thái

Biểu đồ hoạt động biểu diễn các hoạt động và sự đồng bộ, chuyển tiếp các hoạt động, thường được sử dụng để biểu diễn các phương thức phức tạp của các lớp.

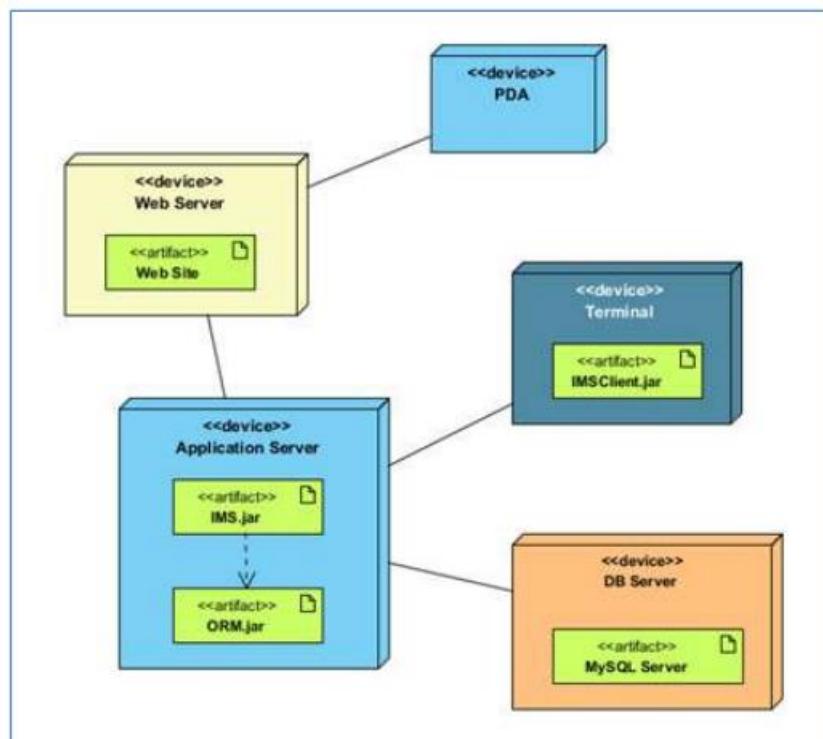
➤ **Biểu đồ thành phần (Component Diagram)**



Hình 2.6 Biểu đồ thành phần

Component Diagram là bản vẽ cho biết cấu trúc của hệ thống theo thành phần phần mềm.

➤ **Biểu đồ triển khai (Deployment Diagram)**



Hình 2.7 Biểu đồ triển khai

Deployment Diagram là bản vẽ giúp chúng ta xác định sẽ triển khai hệ thống phần mềm như thế nào. Đồng thời, xác định chúng ta sẽ đặt các thành phần phần mềm (component) lên hệ thống ra sao.

Deployment Diagram thể hiện rõ kiến trúc triển khai nên nó sẽ ảnh hưởng đến sự thiết kế, phát triển, hiệu năng, khả năng mở rộng của hệ thống v.v...

2.3. Thiết kế và lập trình font-end

Front-end, còn được gọi là phía client-side, là phần của một ứng dụng hoặc trang web mà người dùng cuối có thể thấy và tương tác trực tiếp. Nó bao gồm tất cả các yếu tố liên quan đến giao diện người dùng, thiết kế, và các ngôn ngữ như HTML và CSS. Front-end cũng liên quan đến lập trình phía người dùng, nơi các chức năng và tương tác người dùng được xây dựng.

Để trở thành một lập trình viên Front-end, bạn cần học và làm quen với các yếu tố quan trọng sau:

- HTML (Hypertext Markup Language): Đây là ngôn ngữ đánh dấu cơ bản cho việc tạo cấu trúc và nội dung của trang web. HTML xác định cách các phần tử trên trang web sẽ được định dạng và hiển thị.
- CSS (Cascading Style Sheets): CSS được sử dụng để kiểm soát giao diện và bố cục của trang web. Nó cho phép bạn tạo ra thiết kế, màu sắc, kích thước và các hiệu ứng trên trang.
- JavaScript: JavaScript là ngôn ngữ lập trình phía người dùng và giúp tạo ra các tương tác động trên trang web. Với JavaScript, bạn có thể thêm tính năng như kiểm tra hợp lệ của biểu mẫu, hiệu ứng động, và tương tác người dùng.

Ngoài ba yếu tố cơ bản trên, một lập trình viên Front-end thường cần hiểu về các công nghệ và khái niệm khác như:

- React(ReactJS): React(ReactJS) là một thư viện JavaScript mã nguồn mở, được dùng để xây dựng giao diện người dùng (frontend) cho web. React chỉ

tập trung vào phần hiển thị giao diện (view), chứ không can thiệp vào cách sắp xếp logic nghiệp vụ hoặc cấu trúc ứng dụng.

- Bootstrap: Bootstrap là một framework CSS và JavaScript phổ biến giúp xây dựng trang web nhanh chóng và có tích hợp sẵn các thành phần và giao diện thường được sử dụng.
- Tailwind: Tailwind là framework CSS hỗ trợ phát triển xây dựng nhanh chóng giao diện người dùng, nó cũng có điểm chung giống như Bootstrap & điểm làm nó nổi bật hơn cả đó là chúng ta có thể tùy biến phát triển CSS theo cách mà chúng ta định nghĩa ra. Tailwind CSS cung cấp các class với những nhiệm vụ khác nhau, sử dụng class để thêm vào thẻ HTML.
- JQuery: jQuery là một thư viện JavaScript đơn giản hóa việc tương tác và làm việc với DOM (Document Object Model) trên các trình duyệt web. Nó giúp giảm bớt việc viết mã JavaScript dài dòng và phức tạp.
- UX/UI (User Experience/User Interface): Hiểu về thiết kế giao diện người dùng và tối ưu hóa trải nghiệm người dùng là quan trọng. Điều này bao gồm việc thiết kế sao cho trang web dễ sử dụng và hấp dẫn cho người dùng.

2.4. Tổng quan về lập trình back end

Node.js, hay NodeJS, đang trở thành xu hướng thế giới trong xây dựng và phát triển ứng dụng web. Có khoảng 2% (khoảng 22 triệu) ứng dụng web trên thế giới đang sử dụng Node.js cho backend.

NodeJS là một runtime environment JavaScript mã nguồn mở, được xây dựng trên nền tảng V8 engine của Google Chrome. Nó cho phép các lập trình viên xây dựng các ứng dụng web một cách nhanh chóng và có khả năng mở rộng cao, nhờ vào mô hình lập trình không đồng bộ và tức thời. Nói một cách đơn giản, NodeJS cho phép bạn viết các ứng dụng máy chủ (backend) bằng JavaScript, cùng một ngôn ngữ mà bạn đã sử dụng để tạo giao diện người dùng (front end).

NodeJS Backend là phần phía máy chủ của một ứng dụng web được xây dựng bằng Node.js. Quá trình này bao gồm xử lý logic, dữ liệu và tương tác với cơ sở dữ liệu bên phía máy chủ. Cụ thể NodeJS Backend thực hiện các nhiệm vụ:

- Xử lý yêu cầu từ phía người dùng: Khi bạn truy cập một trang web, trình duyệt của bạn sẽ gửi một yêu cầu đến máy chủ. NodeJS Backend sẽ nhận yêu cầu này, phân tích và xử lý nó.
- Truy xuất và xử lý dữ liệu: Backend sẽ kết nối với cơ sở dữ liệu để lấy hoặc lưu trữ dữ liệu theo yêu cầu. Ví dụ: Khi bạn đăng nhập vào một trang web, backend sẽ kiểm tra thông tin đăng nhập của bạn trong cơ sở dữ liệu để xác thực.
- Thực hiện các phép tính: Backend có thể thực hiện các phép tính phức tạp, chẳng hạn như tính toán giá trị đơn hàng, tạo báo cáo, ...
- Tạo ra các API: Backend có thể cung cấp các API để các ứng dụng khác có thể tương tác với nó. Ví dụ: Một ứng dụng di động có thể sử dụng API của backend để lấy dữ liệu từ máy chủ.
- Quản lý phiên: Backend theo dõi trạng thái của mỗi người dùng, chẳng hạn các sản phẩm đã thêm vào giỏ hàng, thông tin đăng nhập, ...

2.5. Công nghệ thao tác dữ liệu

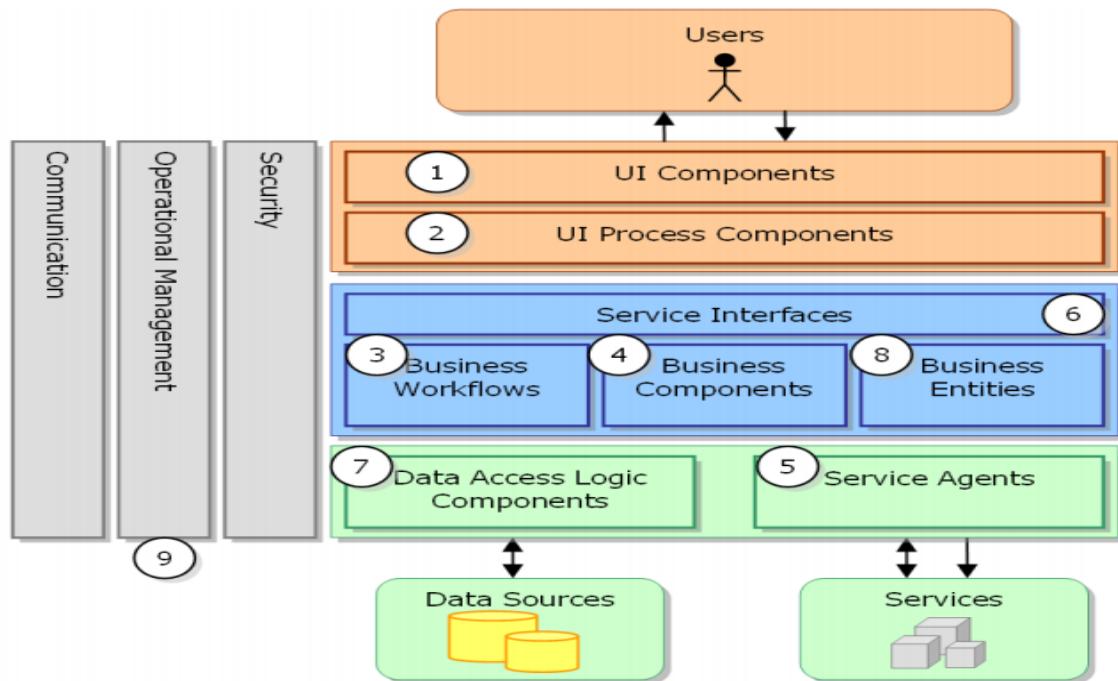
Node.js được sử dụng làm nền tảng phát triển backend, kết hợp với MySQL làm hệ quản trị cơ sở dữ liệu. Để kết nối và thao tác với MySQL, ứng dụng sử dụng thư viện mysql2 – một thư viện phổ biến giúp kết nối, truy vấn và xử lý dữ liệu một cách hiệu quả trong môi trường Node.js.

Ưu điểm:

- Hiệu suất cao: Node.js sử dụng mô hình bất đồng bộ, giúp xử lý nhiều truy vấn đồng thời mà không bị chặn.
- Dễ mở rộng: Hỗ trợ nhiều phương pháp kết nối khác nhau như callback, promise hoặc sử dụng ORM như Sequelize.

- Tương thích tốt: Node.js có thể kết hợp với nhiều hệ thống khác để xây dựng các API RESTful kết nối với MySQL.

2.6. Mô hình thao tác dữ liệu 3 layer



Hình 2.8 Mô hình 3-layer

Mô hình 3 lớp hay còn được gọi là mô hình Three Layer(3-Layer), mô hình này ra đời nhằm phân chia các thành phần trong hệ thống, các thành phần cùng chức năng sẽ được nhóm lại với nhau và phân chia công việc cho từng nhóm để dữ liệu không bị chồng chéo và chạy lộn xộn.

Mô hình này phát huy hiệu quả nhất khi bạn xây dựng một hệ thống lớn, việc quản lý code và xử lý dữ liệu lỗi dễ dàng hơn.

Mô hình 3-layer gồm có 3 phần chính:

- Presentation Layer (GUI): Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện (win form, web form,...) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

- Business Logic Layer (BLL): Layer này phân ra 2 thành nhiệm vụ :

- Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.
- Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.
 - Data Access Layer (DAL): Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa,...).

➤ **Ưu điểm**

- Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.
- Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cài đặt trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.
- Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại khi có sự thay đổi giữa hai môi trường (Winform sang Webfrom) thì chỉ việc thay đổi lại lớp GUI.
- Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.
- Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Đặc tả yêu cầu phần mềm

3.1.1 Các yêu cầu chức năng

a) Chức năng của phân hệ quản trị nội dung

Mục	Tên chức năng	Mô tả
I	Quản lý nhân viên	
1	Thêm thông tin nhân viên	Chức năng này cho phép thêm thông tin của nhân viên vào hệ thống.
2	Cập nhật thông tin nhân viên	Chức năng này cho phép cập nhật lại thông tin của nhân viên trong hệ thống.
3	Xóa thông tin nhân viên	Chức năng này cho phép xoá đi thông tin nhân viên nếu nhân viên đó không còn làm việc nữa.
4	Tìm kiếm nhân viên	Chức năng cho phép tìm kiếm thông tin của nhân viên trong hệ thống.
5	Hiển thị thông tin nhân viên	Chức năng này cho phép hiển thị thông tin nhân viên trong hệ thống.
II	Quản lý phòng	
1	Thêm thông tin phòng	Chức năng này cho phép thêm thông tin của phòng: Mã phòng, tên phòng, số lượng phòng, giá phòng, mã loại phòng, ... vào hệ thống.
2	Cập nhật thông tin phòng	Chức năng này cho phép cập nhật lại thông tin của phòng.
3	Tìm kiếm phòng	Chức năng này cho phép tìm kiếm thông tin phòng theo: mã phòng, tên phòng,...
4	Hiển thị thông tin phòng	Chức năng này cho phép hiển thị thông tin của phòng.

5	Xóa thông tin phòng	Chức năng này cho phép xóa thông tin phòng.
III	Đăng nhập	
1	Đăng nhập của người quản trị	Xác thực danh tính người quản trị. Người quản trị thực hiện đăng nhập vào hệ thống (điền email, mật khẩu). Nếu email, mật khẩu khớp với dữ liệu được lưu thì thông báo đăng nhập thành công, ngược lại đưa ra thông báo yêu cầu đăng nhập lại.
IV	Quản lý loại phòng	
1	Thêm loại phòng	Chức năng này cho phép nhân viên thêm loại phòng mới.
2	Sửa loại phòng	Chức năng này cho phép nhân viên sửa loại phòng.
3	Xóa loại phòng	Chức năng này cho phép nhân viên xóa loại phòng.
4	Tìm kiếm loại phòng	Chức năng này cho phép nhân viên tìm kiếm loại phòng.
V	Quản lý dịch vụ	
1	Thêm dịch vụ	Nhân viên hoặc admin có thể thêm thông tin về dịch vụ mới vào hệ thống.
2	Sửa dịch vụ	Cập nhật hoặc chỉnh sửa thông tin của dịch vụ.
3	Xóa dịch vụ	Loại bỏ thông tin về dịch vụ không còn sử dụng.
4	Tìm kiếm dịch vụ	Tìm kiếm thông tin dịch vụ theo: Mã dịch vụ, tên dịch vụ.

VI	Quản lý thiết bị	
1	Thêm thiết bị	Nhân viên hoặc admin có thể thêm thông tin về thiết bị mới vào hệ thống.
2	Sửa thiết bị	Cập nhật hoặc chỉnh sửa thông tin của thiết bị.
3	Xóa thiết bị	Loại bỏ thông tin về thiết bị không còn sử dụng.
4	Tìm kiếm thiết bị	Tìm kiếm thông tin thiết bị theo: Mã thiết bị, tên thiết bị.
VII	Quản lý khách hàng	
1	Thêm khách hàng	Nhân viên hoặc admin có thể thêm thông tin về khách hàng mới vào hệ thống.
2	Sửa khách hàng	Cập nhật hoặc chỉnh sửa thông tin của khách hàng.
3	Xóa khách hàng	Loại bỏ thông tin về khách hàng không còn sử dụng.
4	Tìm kiếm khách hàng	Tìm kiếm thông tin khách hàng theo: Mã khách hàng, tên khách hàng.
VIII	Quản lý tài khoản	
1	Thêm tài khoản	Nhân viên hoặc admin có thể thêm thông tin về tài khoản mới vào hệ thống.
2	Sửa tài khoản	Cập nhật hoặc chỉnh sửa thông tin của tài khoản.
3	Xóa tài khoản	Loại bỏ thông tin về tài khoản không còn sử dụng.
4	Tìm kiếm tài khoản	Tìm kiếm thông tin tài khoản theo: Mã tài khoản, tên tài khoản, email.

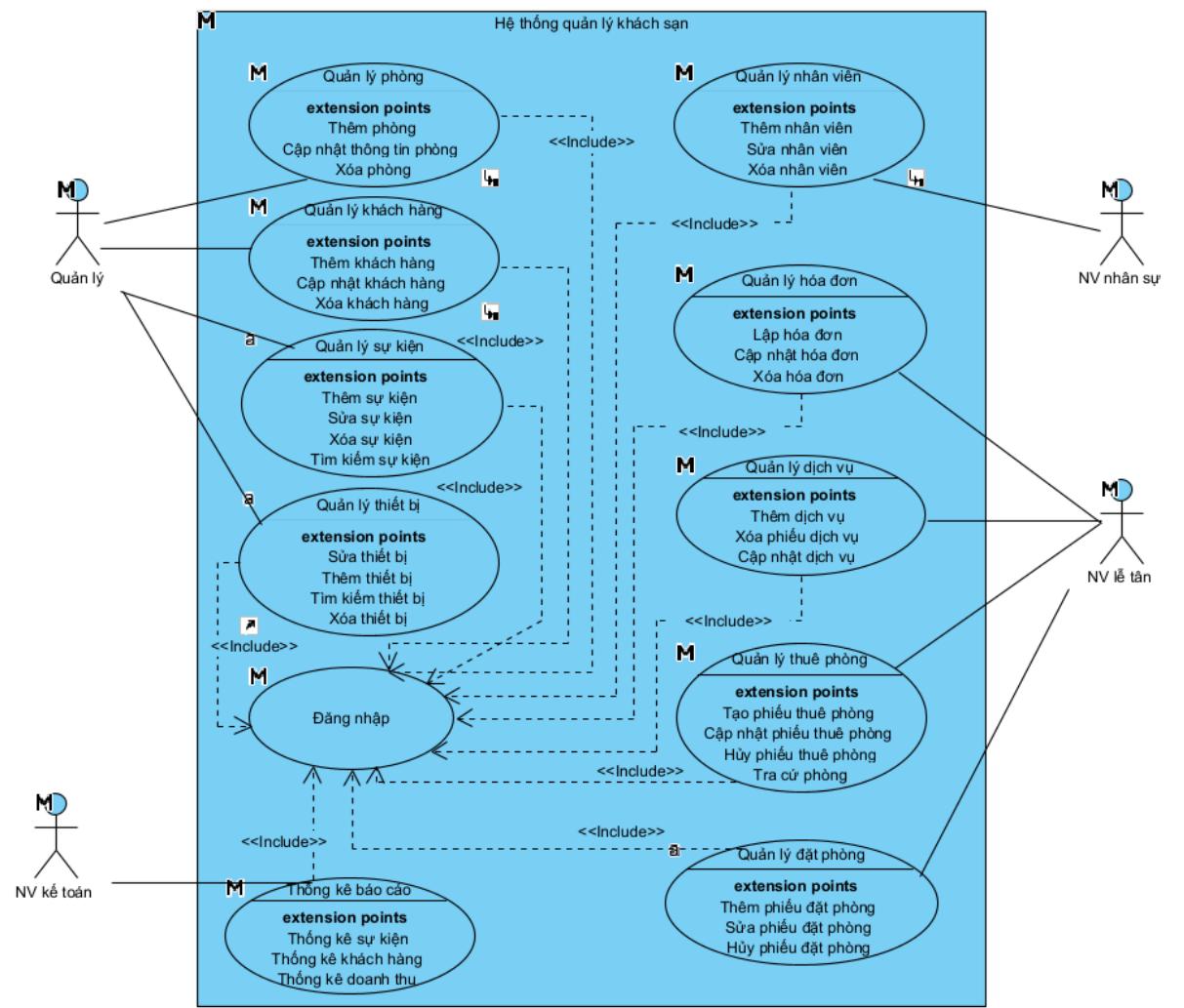
IX	Quản lý hóa đơn	
1	Thêm hóa đơn	Nhân viên hoặc admin có thể thêm thông tin về hóa đơn mới vào hệ thống.
2	Sửa hóa đơn	Cập nhật hoặc chỉnh sửa thông tin của hóa đơn.
3	Xóa hóa đơn	Loại bỏ thông tin về hóa đơn.
4	Tìm kiếm hóa đơn	Tìm kiếm thông tin hóa đơn theo: Mã hóa đơn, tên hóa đơn, tên khách hàng.
X	Quản lý sự kiện	
1	Thêm sự kiện	Nhân viên hoặc admin có thể thêm thông tin về sự kiện mới vào hệ thống.
2	Sửa sự kiện	Cập nhật hoặc chỉnh sửa thông tin của sự kiện.
3	Xóa sự kiện	Loại bỏ thông tin về sự kiện.
4	Tìm kiếm sự kiện	Tìm kiếm thông tin sự kiện theo: Mã sự kiện, tên sự kiện.
XI	Quản lý thống kê	
1	Thống kê doanh thu	Nhân viên hoặc admin có thể thống kê doanh thu theo thời gian.
2	Thống kê đặt phòng	Thống kê số lượng phòng được đặt.
3	Thống kê hủy phòng	Thống kê số lượng phòng bị hủy.
4	Thống kê khách hàng	Thống kê số lượng khách hàng.

b) Chức năng của phân hệ người dùng

Mục	Tên chức năng	Mô tả
1	Xem phòng	Khách hàng có thể xem thông tin về phòng, bao gồm mô tả về diện tích phòng, tiện nghi, giá cả, hình ảnh, ...
2	Xem chi tiết phòng	Người dùng có thể xem chi tiết phòng bằng cách click vào hình ảnh của phòng đó. Tại đây, khách hàng có thể xem chi tiết đánh giá, mô tả phòng, ... và có thể đặt phòng ngay.
3	Đọc bài viết, tin tức, blog	Khách hàng có thể đọc các bài viết, tin tức, blog và thông tin mới nhất về phòng, xu hướng, và sự kiện.
4	Đặt phòng	Khách hàng có thể chọn và đặt phòng mà họ muốn. Khách hàng phải cung cấp thông tin ngày nhận phòng và ngày trả phòng.
5	Quản lý thông tin cá nhân	Khách hàng cập nhật thông tin của bản thân vào trang user từ đó những thông tin của khách hàng sẽ được quản lý và lưu thông tin. Ngoài ra khách hàng có thể nhìn thấy ảnh đại diện của bản thân ở mọi trang khi khách hàng đã đăng nhập và cập nhật thông tin của bản thân.
6	Thanh toán	Khách hàng có thể thanh toán qua nhiều hình thức thanh toán (COD, Banking, ...) giúp khách hàng thanh toán một cách an toàn và nhanh chóng.
7	Quản lý đặt phòng	Khách hàng có thể theo dõi lịch sử đặt phòng, kiểm tra trạng thái đặt phòng hiện tại, và thực hiện các thao tác như chỉnh sửa hoặc hủy đặt

		phòng.
8	Đánh giá	Khách hàng có thể để lại đánh giá và nhận xét về trải nghiệm của mình tại khách sạn.
9	Liên hệ và hỗ trợ	Khách hàng có thể đặt câu hỏi, yêu cầu trợ giúp hoặc tư vấn về các dịch vụ, chính sách của khách sạn.
10	Xem ưu đãi và khuyến mãi	Khách hàng có thể theo dõi các chương trình ưu đãi, khuyến mãi, mã giảm giá hoặc các gói combo đặc biệt do khách sạn cung cấp theo mùa hoặc dịp lễ đặc biệt.
I	Đăng ký, Đăng nhập	
1	Đăng ký của người dùng	Người dùng thực hiện đăng ký vào hệ thống (điền email, mật khẩu). Nếu đăng ký thành công email, mật khẩu sẽ được lưu thì thông báo đăng ký thành công, thông tin được lưu ở cơ sở dữ liệu. Chuyển qua bước đăng nhập.
2	Đăng nhập của người dùng	Xác thực danh tính người dùng. Người dùng thực hiện đăng nhập vào hệ thống (điền email, mật khẩu). Nếu email, mật khẩu khớp với dữ liệu được lưu thì thông báo đăng nhập thành công, ngược lại đưa ra thông báo yêu cầu đăng nhập lại. Nếu không tồn tại thì yêu cầu đăng ký tài khoản.

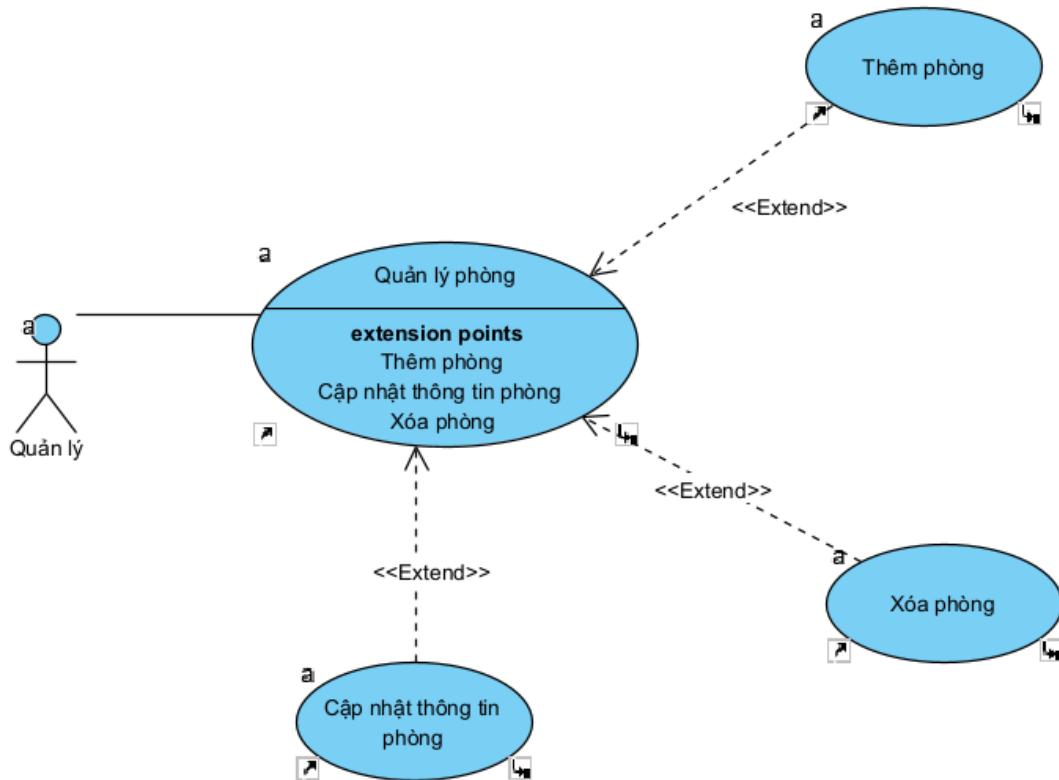
c) Biểu đồ ca sử dụng



Hình 3.1 Biểu đồ use case tổng quát

Biểu đồ tổng quát gồm các chức năng chính như sau: Quản lý loại phòng, quản lý phòng, quản lý dịch vụ, quản lý thiết bị, quản lý thiết bị phòng, quản lý khách hàng, quản lý nhân viên, quản lý đặt phòng, quản lý hóa đơn, báo cáo thống kê,... Tất cả các chức năng này đều yêu cầu phải đăng nhập.

➤ Biểu đồ Usecase quản lý phòng



- Dòng sự kiện chính

1. Đăng nhập vào hệ thống
2. **SYSTEM** Hiển thị giao diện cho người dùng đăng nhập
3. Người dùng nhập thông tin tài khoản.
4. Người dùng nhấn nút "Đăng Nhập".
5. **SYSTEM** Hệ thống kiểm tra thông tin tài khoản:
 - 5.1. **if** Nếu thông tin hợp lệ:
 - 5.1.1. **SYSTEM** Hệ thống xác nhận và chuyển người dùng đến giao diện chính.
end if
 - 5.2. **if** Nếu thông tin không hợp lệ:
 - 5.2.1. **SYSTEM** Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.
end if
6. **SYSTEM** Hiển thị giao diện trang chủ
7. Chọn chức năng quản lý phòng
8. **SYSTEM** Giao diện quản lý phòng hiển thị

9. **SYSTEM** Hệ thống lấy về danh sách phòng hiển thị lên giao diện
10. **if** Nếu người dùng yêu cầu chức năng thêm phòng
 - 10.1. **SYSTEM** Hệ thống thực hiện chức năng thêm thông tin phòng
 - end if**
11. **if** Nếu người dùng yêu cầu chức năng xóa thông tin phòng
 - 11.1. **SYSTEM** Hệ thống thực hiện chức năng xóa thông tin phòng
 - end if**
12. **if** Nếu người dùng yêu cầu chức năng sửa thông tin phòng
 - 12.1. **SYSTEM** Hệ thống thực hiện chức năng sửa thông tin phòng
 - end if**
13. **if** Nếu người dùng yêu cầu chức năng tìm kiếm thông tin phòng
 - 13.1. **SYSTEM** Hệ thống thực hiện chức năng tìm kiếm thông tin phòng
 - end if**

- **Dòng sự kiện phụ**

- a. Thêm phòng
 1. Chọn chức năng thêm phòng
 2. **SYSTEM** Hiển thị khung nhập thông tin phòng
 3. Nhập thông tin phòng
 - 3.1. Nhập số phòng
 - 3.2. Chọn loại phòng
 - 3.3. Nhập giá phòng
 4. Nhấn nút thêm phòng
 5. **SYSTEM** Hệ thống kiểm tra tính hợp lệ của dữ liệu:
 - 5.1. Kiểm tra nếu phòng đã tồn tại.
 6. **if** Nếu thông tin hợp lệ:
 - 6.1. **SYSTEM** Hệ thống lưu thông tin phòng vào cơ sở dữ liệu và hiển thị thông báo "Thêm phòng thành công".

7. **else if** Nếu có lỗi:

 7.1. **SYSTEM** Hệ thống hiển thị thông báo lỗi.

end if

b. Sửa phòng

1. Chọn chức năng sửa phòng

2. **SYSTEM** Hiển thị danh sách phòng

3. Chọn vào phòng cần sửa

4. **SYSTEM** Hiển thị thông tin lên các ô nhập liệu

5. Sửa thông tin

 5.1. Sửa số phòng

 5.2. Sửa loại phòng

 5.3. Sửa giá phòng

6. Nhấn nút sửa

7. **SYSTEM** Hệ thống kiểm tra tính hợp lệ của dữ liệu mới:

 7.1. Kiểm tra xem thông tin có trùng không.

8. **if** Nếu thông tin hợp lệ:

 8.1. **SYSTEM** Hệ thống cập nhật thông tin phòng vào cơ sở dữ liệu và hiển thị thông báo "Sửa thành công".

9. **else if** Nếu có lỗi:

 9.1. **SYSTEM** Hệ thống hiển thị thông báo lỗi.

end if

c. Xóa phòng

1. Chọn chức năng xóa phòng

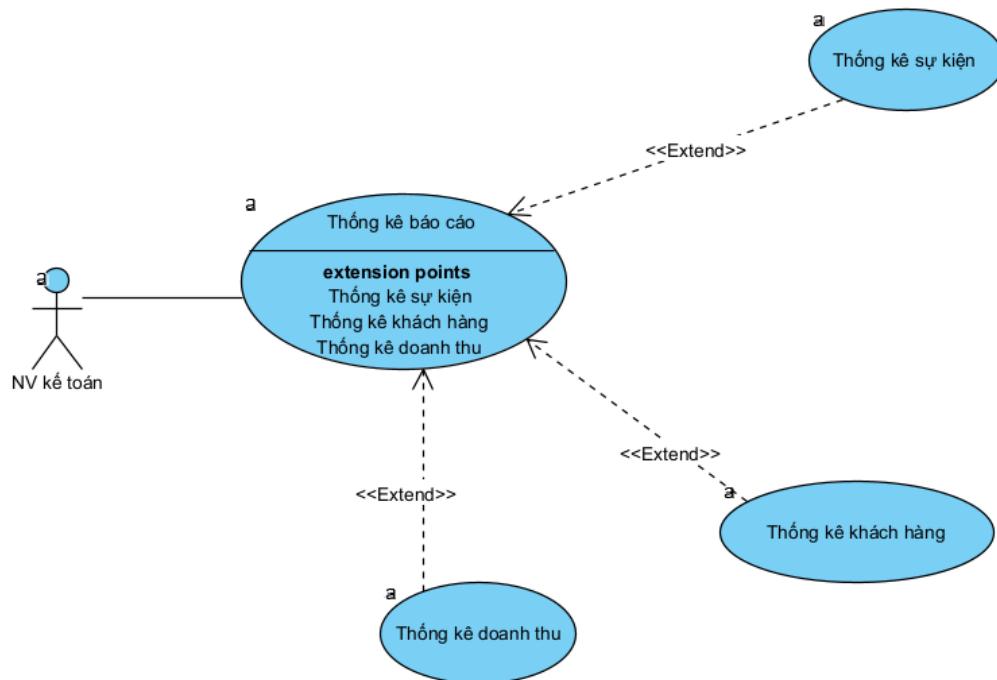
2. **SYSTEM** Hiển thị danh sách phòng

3. Chọn phòng cần xóa

4. Nhấn nút xóa phòng

5. **SYSTEM** Hệ thống xóa phòng khỏi cơ sở dữ liệu và hiển thị thông báo "Xóa phòng thành công".

➤ **Biểu đồ Usecase thống kê**



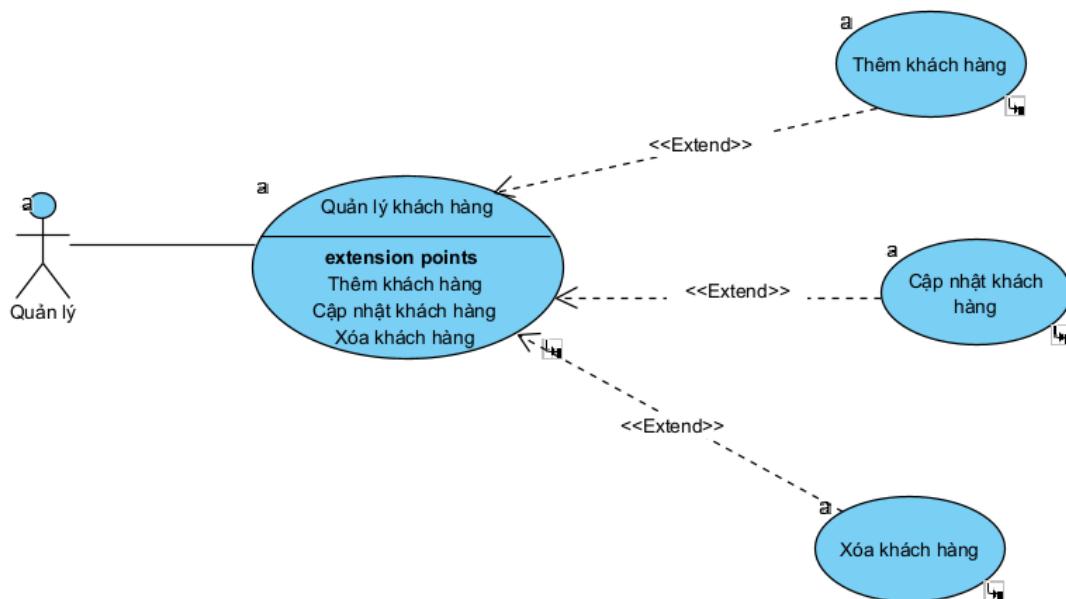
1. Đăng nhập vào hệ thống
2. **SYSTEM** Hiển thị giao diện cho người dùng đăng nhập
3. Người dùng nhập thông tin tài khoản.
4. Người dùng nhấn nút "Đăng Nhập".
5. **SYSTEM** Hệ thống kiểm tra thông tin tài khoản:
 - 5.1. **if** Nếu thông tin hợp lệ:
 - 5.1.1. **SYSTEM** Hệ thống xác nhận và chuyển người dùng đến giao diện chính.

end if
 - 5.2. **if** Nếu thông tin không hợp lệ:
 - 5.2.1. **SYSTEM** Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại thông tin.

end if

6. **SYSTEM** Hiển thị giao diện trang chủ
 7. Chọn chức năng thống kê báo cáo
 8. **SYSTEM** Giao diện thống kê báo cáo hiển thị
 9. **SYSTEM** Hệ thống lấy về danh sách phòng hiển thị lên giao diện
 10. **if** Nếu người dùng yêu cầu chức năng thống kê doanh thu
 - 10.1. **SYSTEM** Hệ thống thực hiện chức năng thống kê doanh thu
- end if
11. **if** Nếu người dùng yêu cầu chức năng thống kê khách hàng
 - 11.1. **SYSTEM** Hệ thống thực hiện chức năng thống kê khách hàng
- end if
12. **if** Nếu người dùng yêu cầu chức năng thống kê sự kiện
 - 12.1. **SYSTEM** Hệ thống thực hiện chức năng thống kê sự kiện
- end if

➤ Biểu đồ Usecase quản lý khách hàng



- Dòng sự kiện chính

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn quản lý khách hàng từ giao diện chính sau khi đã đăng nhập	2. Hiển thị giao diện
3. Tùy chọn chức năng con có trong quản lý khách hàng	

- Dòng sự kiện phụ

a. Thêm khách hàng

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng thêm khách hàng	2. Hiển thị giao diện
3. Nhập thông tin:	
<ul style="list-style-type: none"> – Tên khách hàng: khác rỗng – Địa chỉ – SDT: bắt đầu là số 0, không chứa chữ – Email 	
4. Click nút để lưu	

b. Sửa khách hàng

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng sửa khách hàng	2. Hiển thị giao diện
3. Nhập thông tin cần sửa:	
<ul style="list-style-type: none"> – Tên khách hàng: khác rỗng – Địa chỉ 	

<ul style="list-style-type: none"> - SDT: bắt đầu là số 0, không chứa chữ - Email 	
4. Click nút để lưu	

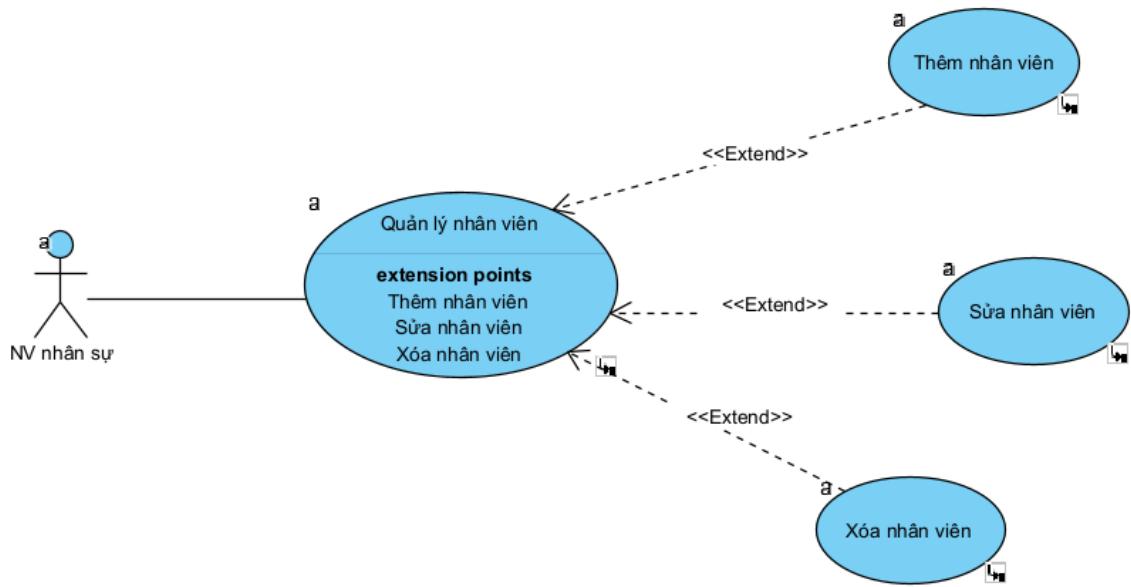
c. Xóa khách hàng

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng xóa khách hàng	
3. Chọn thông tin khách hàng muốn xóa trong danh sách	2. Hiển thị giao diện
4. Click xóa để xóa	

d. Tìm kiếm khách hàng

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng tìm kiếm	2. Hiển thị giao diện
3. Nhập tên cần tìm	
4. Click tìm kiếm	

➤ Biểu đồ Usecase quản lý nhân viên



- Dòng sự kiện chính thêm nhân viên

Hành động của tác nhân	Phản ứng của hệ thống
1. Người quản lí chọn chức năng Quản lý nhân viên.	
	2. Hiển thị danh sách các chức năng quản lí nhân viên, cho phép quản lí lựa chọn.
3. Chọn chức năng: thêm thông tin nhân viên, quản lí tiến hành việc thêm nhân viên vào hệ thống.	
	4. Hiển thị trang nhập thông tin nhân viên.
5. Quản lí nhập các thông tin nhân viên mà hệ thống yêu cầu.	
6. Gửi thông tin đã nhập tới hệ thống.	

	7. Kiểm tra thông tin đã nhập, nếu chính xác thì lưu thông tin này vào cơ sở dữ liệu và thông báo thành công, trường hợp sai so với định dạng đã thống nhất thì thông báo thêm mới không thành công.
--	--

- Dòng sự kiện phụ:

Hành động của tác nhân	Phản ứng của hệ thống
1. Người dùng không muốn thêm thông tin nhân viên, ấn chọn nút “Huỷ”.	
	2. Thoát khỏi giao diện nhập thông tin nhân viên.
3. Thông tin nhân viên không hợp lệ.	
	4. Hệ thống thông báo thông tin không hợp lệ, yêu cầu người dùng nhập lại thông tin quay về bước 1.

– Dòng sự kiện chính sửa nhân viên

Hành động của tác nhân	Phản ứng của hệ thống
1. Người quản lí chọn chức năng Quản lý nhân viên.	
	2. Hiển thị danh sách các chức năng quản lí nhân viên, cho phép quản lí lựa chọn.

3. Chọn chức năng: Sửa thông tin nhân viên, quản lý tiến hành việc sửa thông tin nhân viên vào hệ thống.	
	4. Hiển thị trang sửa thông tin nhân viên.
5. Quản lý nhập các thông tin nhân viên mà hệ thống yêu cầu sửa.	
6. Gửi thông tin đã sửa tới hệ thống.	
	7. Kiểm tra thông tin đã nhập, nếu chính xác thì lưu thông tin này vào cơ sở dữ liệu và thông báo thành công, trường hợp sai so với định dạng đã thống nhất thì thông báo sửa không thành công.

– **Dòng sự kiện phụ:**

Hành động của tác nhân	Phản ứng của hệ thống
1. Người dùng không muốn sửa thông tin nhân viên, ấn chọn nút “Huỷ”.	
	2. Thoát khỏi Giao diện sửa thông tin nhân viên.
3. Thông tin nhân viên không hợp lệ.	
	4. Hệ thống thông báo thông tin không hợp lệ, yêu cầu người dùng nhập lại thông tin quay về bước 1.

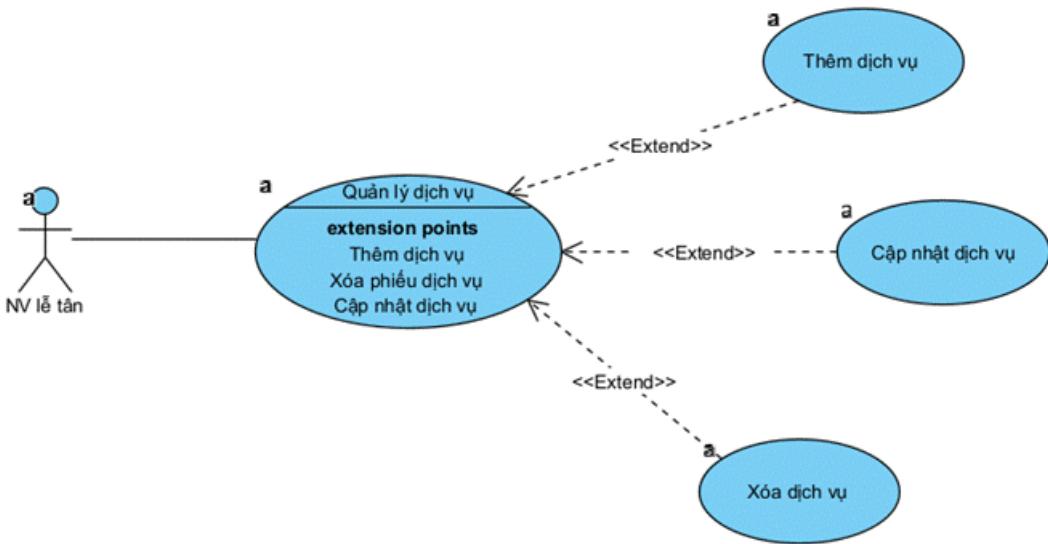
– **Dòng sự kiện chính:**

Hành động của tác nhân	Phản ứng của hệ thống
1. Người quản lí chọn chức năng Quản lý nhân viên.	
	2. Hiển thị danh sách các chức năng quản lí thông tin nhân viên, cho phép quản lí lựa chọn.
3. Chọn chức năng: Xoá thông tin nhân viên, quản lí tiến hành việc xoá thông tin nhân viên khỏi hệ thống.	
	4. Hệ thống thông báo thông tin nhân viên được xoá.

– **Dòng sự kiện phụ:**

Hành động của tác nhân	Phản ứng của hệ thống
1. Người dùng không muốn xoá thông tin nhân viên, ấn chọn nút “Huỷ”.	
	2. Thoát khỏi giao diện xoá thông tin nhân viên.

➤ **Biểu đồ Usecase quản lý dịch vụ**



- Dòng sự kiện chính

Hành động của tác nhân	Phản ứng của hệ thống
1. Đăng nhập vào hệ thống	2. Hiển thị giao diện cho người dùng đăng nhập
3. Sau khi đăng nhập thành công thì hiển thị giao diện trang chủ	4. Hiển thị giao diện trang chủ
5. Chọn chức năng quản lý dịch vụ	6. Giao diện quản lý dịch vụ hiển thị
7. Tùy chọn các chức năng con có trong chức năng quản lý dịch vụ	

- Dòng sự kiện phụ

a. Thêm dịch vụ

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng thêm dịch vụ	2. Hiển thị giao diện

<p>3. Nhập thông tin dịch vụ:</p> <ul style="list-style-type: none"> + Tên dịch vụ: khác rỗng + Đơn giá: > 0 	
<p>4. Click nút Thêm để lưu vào hệ thống</p>	<p>5. Thông báo</p>

b. *Sửa dịch vụ*

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng sửa dịch vụ	2. Hiển thị giao diện
3. Click vào dòng chứa dữ liệu dịch vụ cần sửa	4. Hiển thị dữ liệu phòng cần sửa lên các ô dữ liệu
5. Sửa các thông tin dịch vụ cần sửa + Tên dịch vụ: khác rỗng + Đơn giá: > 0	
6. Click Sửa để lưu vào hệ thống	7. Thông báo

c. *Xóa dịch vụ*

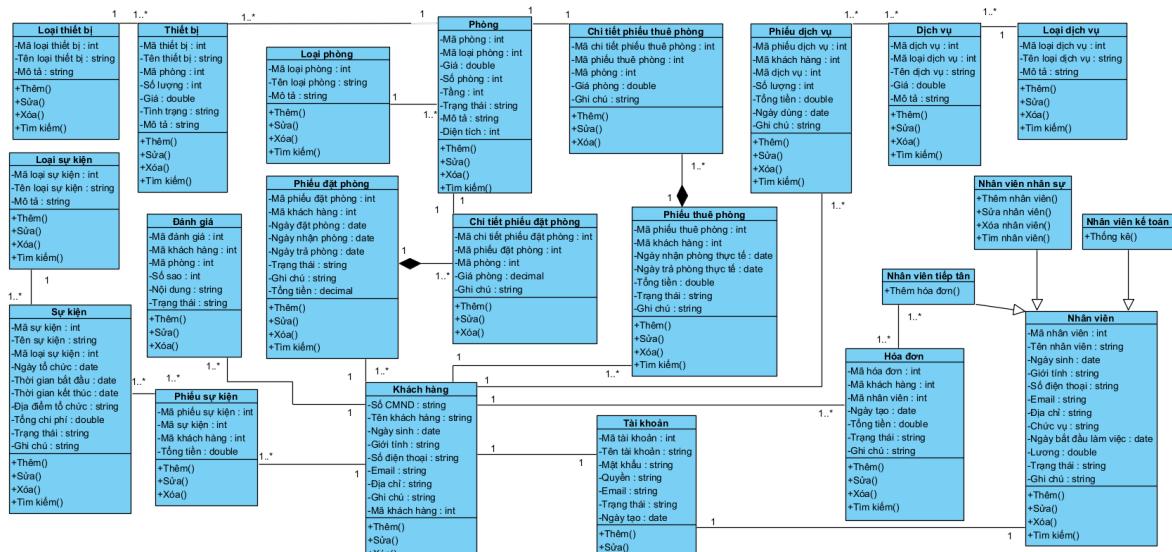
Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng xoá dịch vụ	2. Hiển thị giao diện

3. Chọn dịch vụ cần xóa ở danh sách dịch vụ	
4. Click nút “Xoá” để xoá	5. Thông báo

d. Tìm kiếm dịch vụ

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn chức năng tìm kiếm dịch vụ	2. Hiển thị giao diện
3. Nhập từ khóa tìm kiếm	
4. Click nút “Tìm kiếm” để tìm kiếm	

3.1.2 Biểu đồ lớp thực thể



Hình 3.2 Biểu đồ lớp thực thể

a) Danh sách các lớp đối tượng

Bảng 3.1 Bảng danh sách các lớp đối tượng

STT	Tên lớp	Mô tả
1	Tài khoản	Lớp tài khoản cho phép: thêm, sửa, xoá, thông tin tài khoản.
2	Nhân viên	Lớp này định nghĩa các thông tin của 1 nhân viên trong hệ thống như tên, tuổi, ngày tháng năm sinh và có thể sử dụng được các phương thức thêm, sửa, xóa, tìm kiếm nhân viên.
3	Khách hàng	Lớp này định nghĩa các thông tin của 1 người dùng trong hệ thống như tên, tuổi, ngày tháng năm sinh và có thể sử dụng được các phương thức thêm, sửa, xóa, tìm kiếm khách hàng.
4	Phòng	Lớp phòng định nghĩa các thông tin cơ bản của 1 phòng bao gồm các thông tin như mã phòng, mã loại phòng, số tầng. Và các phương thức như thêm, sửa, xóa, tìm kiếm phòng.
5	Loại phòng	Lớp loại phòng định nghĩa các thông tin liên quan đến các loại phòng, bao gồm các thông tin loại phòng và các phương thức như thêm, sửa, xóa, tìm kiếm loại phòng.
6	Thiết bị	Lớp thiết bị định nghĩa các thông tin cơ bản liên quan đến 1 thiết bị bao gồm các phương thức thêm thiết bị, xóa thiết bị, cập nhật thiết bị.
7	Loại thiết bị	Lớp loại thiết bị định nghĩa các thông tin liên quan đến loại thiết bị gồm các thông tin thiết bị và các phương thức thêm, sửa, xóa loại thiết bị.

8	Loại dịch vụ	Lớp loại dịch vụ định nghĩa các thông tin cơ bản của 1 loại dịch vụ như mã loại dịch vụ, tên loại dịch vụ, mô tả và các phương thức như thêm loại dịch vụ, sửa loại dịch vụ, xóa loại dịch vụ.
9	Dịch vụ	Lớp dịch vụ định nghĩa các thông tin cơ bản liên quan đến dịch vụ như tên, giá và các phương thức thêm, sửa, xóa.
10	Phiếu dịch vụ	Lớp này định nghĩa các thông tin của 1 phiếu dịch vụ trong hệ thống như mã khách hàng, mã dịch vụ, số lượng và có thể sử dụng được các phương thức thêm, sửa, xóa, tìm kiếm phiếu dịch vụ.
11	Phiếu đặt phòng	Lớp này định nghĩa các thông tin của 1 phiếu đặt phòng trong hệ thống và có thể sử dụng được các phương thức thêm, sửa, xóa, tìm kiếm phiếu đặt phòng.
12	Chi tiết phiếu đặt phòng	Lớp này định nghĩa các thông tin của 1 chi tiết phiếu đặt phòng trong hệ thống và có thể sử dụng được các phương thức thêm, sửa, xóa, tìm kiếm chi tiết phiếu đặt phòng.
13	Hóa đơn	Lớp này định nghĩa các thông tin của 1 hóa đơn trong hệ thống và có thể sử dụng các phương thức thêm, sửa, xoá và tìm kiếm hóa đơn.
14	Phiếu thuê phòng	Lớp này định nghĩa các thông tin của 1 phiếu thuê phòng trong hệ thống và có thể sử dụng các phương thức thêm, sửa, xoá và tìm kiếm phiếu thuê phòng.
15	Chi tiết phiếu thuê phòng	Lớp này định nghĩa các thông tin chi tiết của 1 phiếu thuê phòng trong hệ thống và có thể sử dụng các phương thức thêm, sửa, xoá và tìm kiếm

		phiếu thuê phòng.
16	Đánh giá	Lớp này định nghĩa các thông tin của 1 đánh giá trong hệ thống và có thể sử dụng các phương thức thêm, sửa, xoá và tìm kiếm đánh giá.

b) Chi tiết hóa các lớp đối tượng

1. Lớp tài khoản

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaTaiKhoan	int	Khóa chính	Mã tài khoản
2	TenTaiKhoan	Nvarchar(30)	Khác rỗng	Tên tài khoản
3	MatKhau	Nvarchar(30)	Khác rỗng	Mật khẩu
4	Quyen	Nvarchar(10)	Khác rỗng	Quyền
5	NgayTao	Datetime	Khác rỗng	Ngày tạo
6	Email	Nvarchar(30)	Khác rỗng	Email
7	TrangThai	Nvarchar(10)	Khác rỗng	Trạng thái

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemTaiKhoan()	Void	Public	Thêm tài khoản
2	SuaTaiKhoan()	Void	Public	Sửa tài khoản
3	XoaTaiKhoan()	Void	Public	Xóa tài khoản

4	TimKiemTaiKhoan()	Void	Public	Tìm kiếm tài khoản
5	CapNhatTrangThai()	Void	Public	Cập nhật trạng thái

2. Lớp khách hàng

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaKhachHang	int	Khóa chính	Mã khách hàng
2	TenKhachHang	Nvarchar(30)	Khác rỗng	Tên khách hàng
3	GioiTinh	Nvarchar(10)	Khác rỗng	Giới tính
4	NgaySinh	Date	Khác rỗng	Ngày sinh
5	Anh	Nvarchar(max)	Khác rỗng	Ảnh
6	SoDienThoai	Bit	Khác rỗng	Số điện thoại
7	DiaChi	Nvarchar(30)	Khác rỗng	Địa chỉ

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemKhachHang()	Void	Public	Thêm khách hàng
2	SuaKhachHang()	Void	Public	Sửa khách hàng
3	XoaKhachHang()	Void	Public	Xóa khách hàng

4	TimKiemKhachHang()	Void	Public	Tìm kiếm khách hàng
---	--------------------	------	--------	---------------------

3. Lớp nhân viên

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaNhanVien	int	Khóa chính	Mã nhân viên
2	TenNhanVien	Nvarchar(30)	Khác rỗng	Tên nhân viên
3	GioiTinh	Nvarchar(10)	Khác rỗng	Giới tính
4	NgaySinh	Date	Khác rỗng	Ngày sinh
5	Anh	Nvarchar(max)	Khác rỗng	Ảnh
6	SoDienThoai	Bit	Khác rỗng	Số điện thoại
7	DiaChi	Nvarchar(30)	Khác rỗng	Địa chỉ
8	DaXoa	Bit	Khác rỗng	Đã xóa
9	ChucVu	Nvarchar(30)	Khác rỗng	Chức vụ

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemNhanVien()	Void	Public	Thêm nhân viên
2	SuaNhanVien()	Void	Public	Sửa nhân viên
3	XoaNhanVien()	Void	Public	Xóa nhân viên
4	TimKiemNhanVien()	Void	Public	Tìm kiếm nhân

				viên
--	--	--	--	------

4. Lớp phòng

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaPhong	int	Khóa chính	Mã phòng
2	MaLoaiPhong	int	Khóa ngoại	Mã loại phòng
3	SoTang	int	Khác rỗng	Số tầng
4	Gia	Nvarchar(50)	Khác rỗng	Giá
5	Anh	Nvarchar(max)	Khác rỗng	Ảnh
6	SoPhong	int	Khác rỗng	Số phòng
7	TrangThai	Nvarchar(30)	Khác rỗng	Trạng thái

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemPhong()	Void	Public	Thêm phòng
2	SuaPhong()	Void	Public	Sửa phòng
3	XoaPhong()	Void	Public	Xóa phòng
4	TimKiemPhong()	Void	Public	Tìm kiếm phòng

5. Lớp loại phòng

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaLoaiPhong	int	Khóa chính	Mã loại phòng

2	TenLoaiPhong	Nvarchar(50)	Khác rỗng	Tên loại phòng
3	TrangThai	Nvarchar(30)	Khác rỗng	Trạng thái

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemLoaiPhong()	Void	Public	Thêm loại phòng
2	SuaLoaiPhong()	Void	Public	Sửa loại phòng
3	XoaLoaiPhong()	Void	Public	Xóa loại phòng
4	TimKiemLoaiPhong()	Void	Public	Tìm kiếm loại phòng

6. Lớp loại thiết bị

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaLoaiThietBi	int	Khóa chính	Mã loại thiết bị
2	TenLoaiThietBi	Nvarchar(50)	Khác rỗng	Tên loại thiết bị
4	Mô tả	Nvarchar(255)	Khác rỗng	Mô tả

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemLoaiThietBi()	Void	Public	Thêm loại thiết bị

2	SuaLoaiThietBi()	Void	Public	Sửa loại thiết bị
3	XoaLoaiThietBi()	Void	Public	Xóa loại thiết bị
4	TimKiemLoaiThietBi()	Void	Public	Tìm kiếm loại thiết bị

7. Lớp thiết bị

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaThietBi	int	Khóa chính	Mã thiết bị
2	TenThietBi	Nvarchar(50)	Khác rỗng	Tên thiết bị
3	MaPhong	int	Khóa ngoại	Mã phòng
4	TrangThai	Nvarchar(30)	Khác rỗng	Trạng thái

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemThietBi()	Void	Public	Thêm thiết bị
2	SuaThietBi()	Void	Public	Sửa thiết bị
3	XoaThietBi()	Void	Public	Xóa thiết bị
4	TimKiemThietBi()	Void	Public	Tìm kiếm thiết bị

8. Lớp dịch vụ

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả

1	MaDichVu	int	Khóa chính	Mã dịch vụ
2	TenDichVu	Nvarchar(50)	Khác rỗng	Tên dịch vụ
3	MaLoaiDichVu	int	Khóa ngoại	Mã loại dịch vụ
4	MoTa	Nvarchar(30)	Khác rỗng	Mô tả
5	Gia	decimal	Khác rỗng	Giá

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemDichVu()	Void	Public	Thêm dịch vụ
2	SuaDichVu()	Void	Public	Sửa dịch vụ
3	XoaDichVu()	Void	Public	Xóa dịch vụ
4	TimKiemDichVu()	Void	Public	Tìm kiếm dịch vụ

9. Lớp loại dịch vụ

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaLoaiDichVu	int	Khóa chính	Mã loại dịch vụ
2	TenLoaiDichVu	Nvarchar(50)	Khác rỗng	Tên dịch vụ
3	MoTa	Nvarchar(30)	Khác rỗng	Mô tả

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemLoaiDichVu()	Void	Public	Thêm loại dịch vụ
2	SuaLoaiDichVu()	Void	Public	Sửa loại dịch vụ
3	XoaLoaiDichVu()	Void	Public	Xóa loại dịch vụ
4	TimKiemLoaiDichVu()	Void	Public	Tìm kiếm loại dịch vụ

10. Lớp phiếu dịch vụ

- Danh sách thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	MaPhieuDichVu	int	Khóa chính	Mã phiếu dịch vụ
2	MaDichVu	int	Khóa ngoại	Mã dịch vụ
3	MaKhachHang	int	Khóa ngoại	Mã khách hàng
4	TongTien	decimal	Khác rỗng	Tổng tiền
5	Ghi chú	Nvarchar(30)	Khác rỗng	Ghi chú

- Danh sách phương thức

STT	Tên phương thức	Kiểu dữ liệu	Mức truy cập	Mô tả
1	ThemPhieuDichVu()	Void	Public	Thêm phiếu dịch vụ

2	SuaPhieuDichVu()	Void	Public	Sửa phiếu dịch vụ
3	XoaPhieuDichVu()	Void	Public	Xóa phiếu dịch vụ
4	TimKiemPhieuDichVu()	Void	Public	Tìm kiếm phiếu dịch vụ

3.1.3 Các yêu cầu phi chức năng

a) Yêu cầu về hiệu năng

- Tốc độ phản hồi: Website cần có tốc độ tải trang nhanh để cung cấp trải nghiệm người dùng mượt mà. Tốc độ tải trang cần được duy trì ở mức chấp nhận được dưới các điều kiện tải khác nhau.
- Độ tương thích: Website phải tương thích hoàn toàn với các trình duyệt web phổ biến như Chrome, Firefox, Safari.

b) Yêu cầu về giao diện người dùng

- Thiết kế giao diện hấp dẫn: Giao diện người dùng cần phải được thiết kế với một giao diện hấp dẫn, sáng sủa và thú vị. Sử dụng màu sắc, hình ảnh và phông chữ hợp lý để tạo ấn tượng mạnh mẽ.
- Thân thiện với thiết bị di động: Giao diện cần phải linh hoạt để thích nghi với các thiết bị di động và tablet. Đảm bảo rằng trang web hiển thị đẹp và hoạt động mượt trên các kích thước màn hình khác nhau.
- Chức năng tìm kiếm: Cần có một hộp tìm kiếm để người dùng có thể tìm kiếm một cách dễ dàng.

c) Tương tác người dùng

- Xem phòng: Người dùng cần có khả năng xem chi tiết các phòng có sẵn trên trang web. Sử dụng thư viện hình ảnh và hiển thị giá cả để họ có thể tìm hiểu phòng.
- Đăng nhập và đăng ký: Cần cung cấp tính năng đăng nhập cho người dùng hiện có và khả năng đăng ký tài khoản mới.

d) Hiệu ứng trực quan

- Hiệu ứng mượt mà: Sử dụng JavaScript và CSS để thêm hiệu ứng mượt mà như slide, hover effects, và transition animations để làm cho trang web sống động hơn.
- Giao diện người dùng tương tác: Đảm bảo rằng các phần tử giao diện như nút, form, và menu có hiệu ứng tương tác khi người dùng tương tác với chúng.

e) Tương tác với Backend

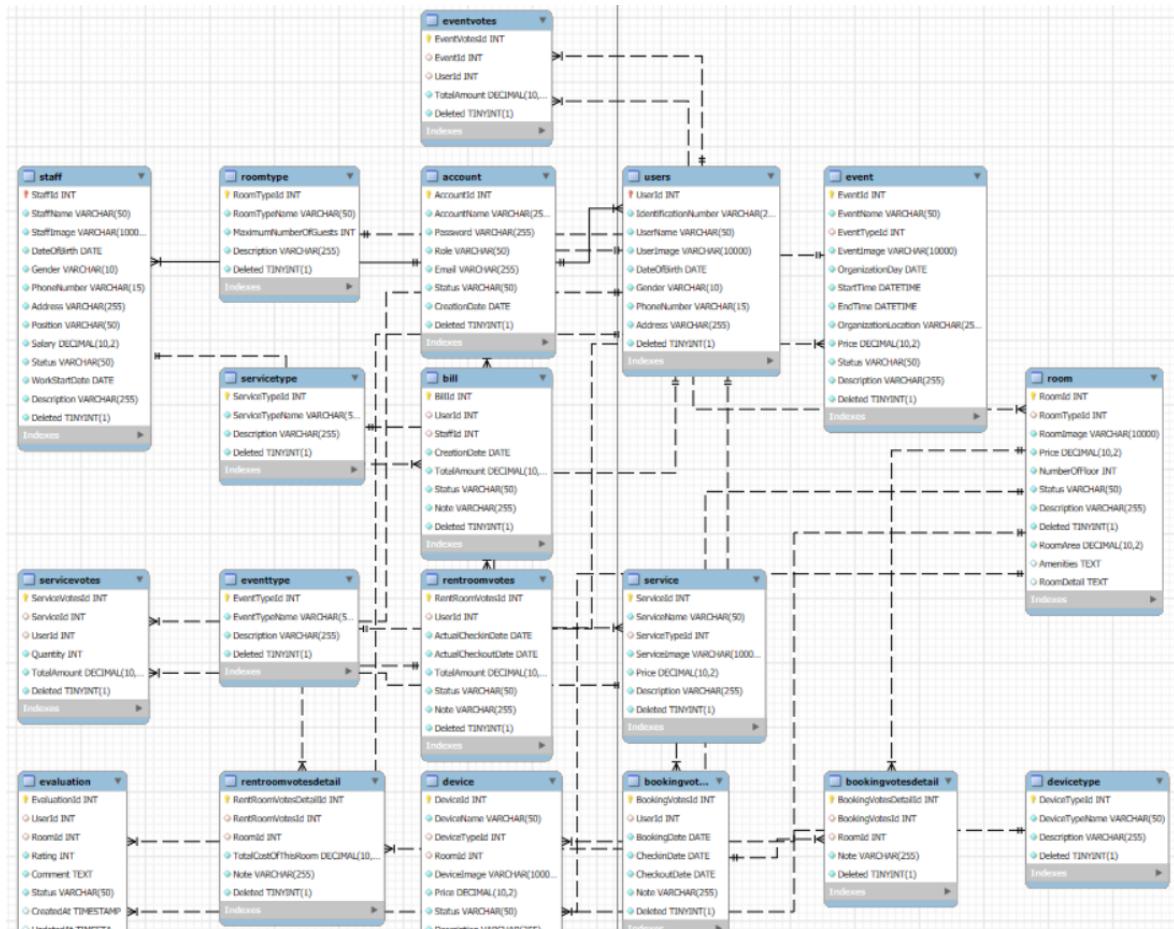
- Xác thực và quản lý người dùng: Đảm bảo rằng tương tác với backend cho phép xác thực người dùng và quản lý đơn hàng và tài khoản.

f) Bảo mật

- Bảo mật dữ liệu người dùng: Đảm bảo rằng thông tin cá nhân của người dùng được bảo mật và mã hóa an toàn.
- Chứng nhận an toàn: Sử dụng HTTPS để đảm bảo an toàn trong việc truyền dữ liệu giữa trình duyệt và máy chủ.
- Quản lý phiên đăng nhập: Đảm bảo rằng phiên đăng nhập của người dùng được quản lý một cách an toàn và đúng cách.

3.2 Thiết kế hệ thống

3.2.1 Thiết kế cơ sở dữ liệu



Hình 3.3 Lược đồ ER Diagram

- Danh sách các bảng

Bảng 3.2 Bảng Loại Phòng

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaLoaiPhong	int	Primary key	Mã loại phòng
2	TenLoaiPhong	Nvarchar(50)	Not null	Tên loại phòng
3	MoTa	Nvarchar(max)	Not null	Mô tả

Bảng 3.3 Bảng Phong

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaPhong	Int	Primary key	Mã Phòng
2	MaLoaiPhong	Int	Not null	Mã loại phòng
3	Gia	decimal	Not null	Giá
4	Tang	int	Not null	Tầng
5	TrangThai	Nvarchar(20)	Not null	Trạng thái

Bảng 3.4 Bảng LoaiThietBi

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaLoaiThietBi	Int	Primary key	Mã loại thiết bị
2	TenLoaiThietBi	Nvarchar(50)	Not null	Tên loại thiết bị
3	MoTa	Nvarchar(max)	Not null	Mô tả

Bảng 3.5 Bảng ThietBi

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaThietBi	Int	Primary key	Mã thiết bị
2	TenThietBi	Nvarchar(50)	Not null	Tên thiết bị
3	MaLoaiThietBi	Int	Not null	Mã loại thiết bị
4	MaPhong	Int	Not null	Mã phòng

5	Gia	Decimal	Not null	Giá
6	TinhTrang	Nvarchar(50)	Not null	Tình trạng
7	MoTa	Nvarchar(max)	Not null	Mô tả

Bảng 3.6 Bảng LoaiDichVu

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaLoaiDichVu	Int	Primary key	Mã loại dịch vụ
2	TenLoaiDichVu	Nvarchar(50)	Not null	Tên loại dịch vụ
3	MoTa	Int	Not null	Mô tả

Bảng 3.7 Bảng DichVu

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaDichVu	Int	Primary key	Mã dịch vụ
2	TenDichVu	nvarchar	Not null	Tên dịch vụ
3	Gia	Float	Not null	Giá dịch vụ
4	MaLoaiDichVu	Int	Not null	Mã loại dịch vụ

Bảng 3.8 Bảng PhieuDichVu

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaPhieuDichVu	Int	Primary key	Mã phiếu dịch vụ
2	MaDichVu	Int	Not null	Mã dịch vụ

3	SoLuong	Int	Not null	Số lượng
4	ThanhTien	Decimal	Not null	Thành tiền

Bảng 3.9 Bảng TaiKhoan

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaTaiKhoan	Int	Primary key	Mã tài khoản
2	TenTaiKhoan	Nvarchar(50)	Not null	Tên tài khoản
3	MatKhau	Nvarchar(50)	Not null	Mật khẩu
4	Quyen	Nvarchar(10)	Not null	Quyền
5	Email	Nvarchar(50)	Not null	Email
6	TrangThai	Nvarchar(20)	Not null	Trạng thái
7	NgayTao	Date	Not null	Ngày tạo
8	Deleted	Bit	Not null	Đã xóa

Bảng 3.10 Bảng KhachHang

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaKhachHang	Int	Primary key	Mã Khách hàng
2	SoCMND	Nvarchar(20)	Not null	Số CMND
3	TenKhachHang	Nvarchar(50)	Not null	Tên khách hàng
4	NgaySinh	Date	Not null	Ngày sinh
5	GioiTinh	Nvarchar(10)	Not null	Giới tính

6	SoDienThoai	Nvarchar(20)	Not null	Số điện thoại
7	DiaChi	Nvarchar(50)	Not null	Địa chỉ
8	Deleted	Bit	Not null	Đã xóa

Bảng 3.11 Bảng LoaiSuKien

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaLoaiSuKien	Int	Primary key	Mã loại sự kiện
2	TenLoaiSuKien	Nvarchar(50)	Not null	Tên loại sự kiện
3	MoTa	Nvarchar(max)	Not null	Mô tả

Bảng 3.12 Bảng SuKien

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaSuKien	Int	Primary key	Mã sự kiện
2	TenSuKien	Nvarchar(50)	Not null	Tên sự kiện
3	MaLoaiSuKien	Int	Not null	Mã loại sự kiện
4	MaKhachHang	Int	Not null	Mã khách hàng
5	NgayToChuc	Date	Not null	Ngày tổ chức
6	ThoiGianBatDau	Date	Not null	Thời gian bắt đầu
7	ThoiGianKetThuc	Date	Not null	Thời gian kết thúc
8	DiaDiemToChuc	Nvarchar(50)	Not null	Địa điểm tổ chức
9	TongChiPhi	Decimal	Not null	Tổng chi phí

10	TrangThai	Nvarchar(50)	Not null	Trạng thái
11	GhiChu	Nvarchar(max)	Not null	Ghi chú

Bảng 3.13 Bảng PhieuDatPhong

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaPhieuDatPhong	Int	Primary key	Mã phiếu đặt phòng
2	MaKhachHang	Int	Not null	Mã khách hàng
3	NgayDatPhong	Date	Not null	Ngày đặt phòng
4	NgayNhanPhong	Date	Not null	Ngày nhận phòng
5	NgayTraPhong	Date	Not null	Ngày trả phòng
6	GhiChu	Nvarchar(max)	Not null	Ghi chú

Bảng 3.14 Bảng ChiTietPhieuDatPhong

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaChiTietPhieuDatPhong	Int	Primary key	Mã chi tiết phiếu đặt phòng
2	MaPhieuDatPhong	Int	Not null	Mã phiếu đặt phòng
3	MaPhong	Int	Not null	Mã phòng

4	GhiChu	Nvarchar(max)	Not null	Ghi chú
---	--------	---------------	----------	---------

Bảng 3.15 Bảng PhieuThuePhong

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaPhieuThuePhong	Int	Primary key	Mã phiếu thuê phòng
2	MaKhachHang	Int	Not null	Mã khách hàng
3	NgayNhanPhongThucTe	Date	Not null	Ngày nhận phòng thực tế
4	NgayTraPhongThucTe	Date	Not null	Ngày trả phòng thực tế
5	TongTien	Decimal	Not null	Tổng tiền
6	TrangThai	Nvarchar(50)	Not null	Trạng thái
7	GhiChu	Nvarchar(max)	Not null	Ghi chú

Bảng 3.16 Bảng ChiTietPhieuThuePhong

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaChiTietPhieuThuePhong	Int	Primary key	Mã chi tiết phiếu thuê phòng
2	MaPhieuThuePhong	Int	Not null	Mã phiếu thuê phòng

3	MaPhong	Int	Not null	Mã phòng
4	MaPhieuDichVu	Int	Not null	Mã phiếu dịch vụ
5	TongTienChoPhongNay	Decimal	Not null	Tổng tiền cho phòng này
6	GhiChu	Nvarchar(max)	Not null	Ghi chú

Bảng 3.17 Bảng NhanVien

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaNhanVien	Int	Primary key	Mã nhân viên
2	TenNhanVien	Int	Not null	Tên nhân viên
3	NgaySinh	Date	Not null	Ngày sinh
4	GioiTinh	Nvarchar(10)	Not null	Giới tính
5	SoDienThoai	Nvarchar(20)	Not null	Số điện thoại
6	Email	Nvarchar(30)	Not null	Email
7	DiaChi	Nvarchar(50)	Not null	Địa chỉ
8	ChucVu	Nvarchar(20)	Not null	Chức vụ
9	Luong	Decimal	Not null	Lương
10	TrangThai	Nvarchar(30)	Not null	Trạng thái
11	NgayBatDauLamViec	Date	Not null	Ngày bắt đầu làm việc
12	GhiChu	Nvarchar(max)	Not null	Ghi chú

Bảng 3.18 Bảng HoaDon

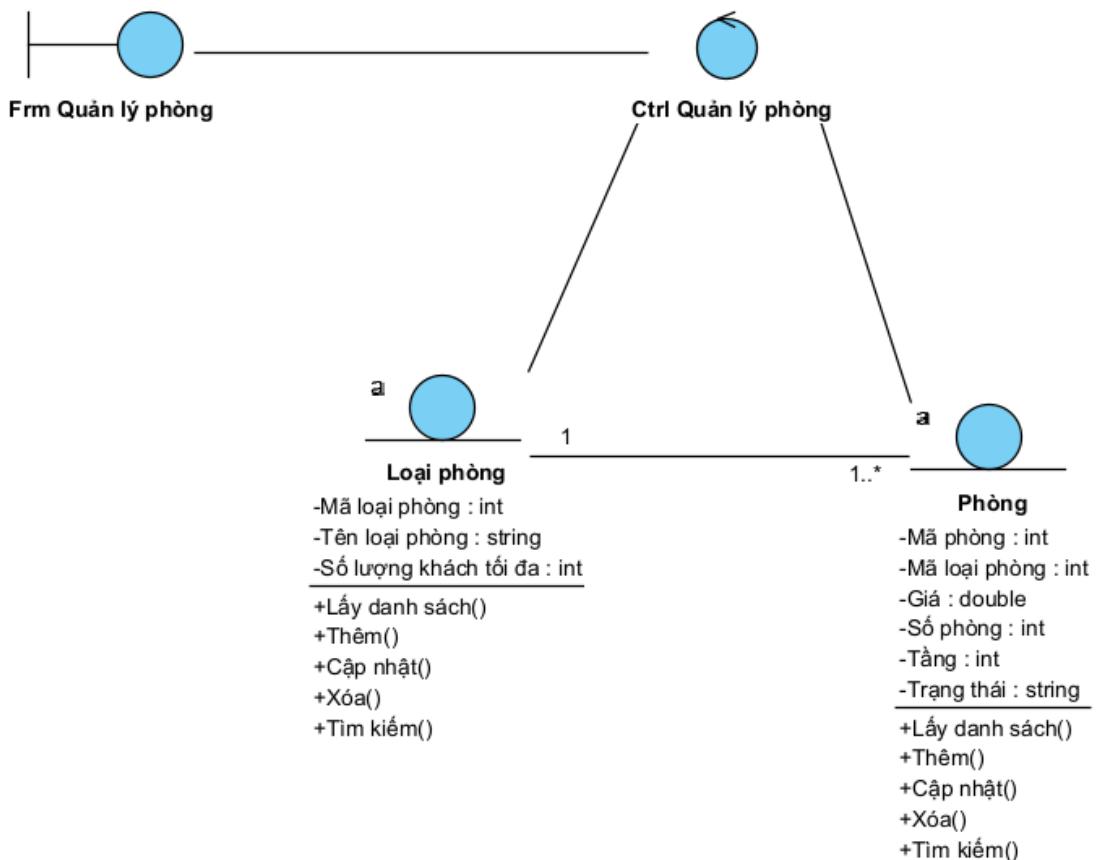
STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
1	MaHoaDon	Int	Primary key	Mã hóa đơn
2	MaPhieuThuePhong	Int	Not null	Mã phiếu đặt phòng
3	MaKhachHang	Int	Not null	Mã khách hàng
4	MaNhanVien	Int	Not null	Mã nhân viên
5	NgayTao	Date	Notnull	Ngày tạo hóa đơn
6	TongTien	Decimal	Notnull	Tổng tiền
7	TrangThai	Nvarchar(50)	Not null	Trạng thái
8	GhiChu	Nvarchar(max)	Not null	Ghi chú

3.2.2 Thiết kế lớp đối tượng

a) Biểu đồ lớp VOPC của các ca sử dụng

➤ **Biểu đồ VOPC của use case quản lý phòng**

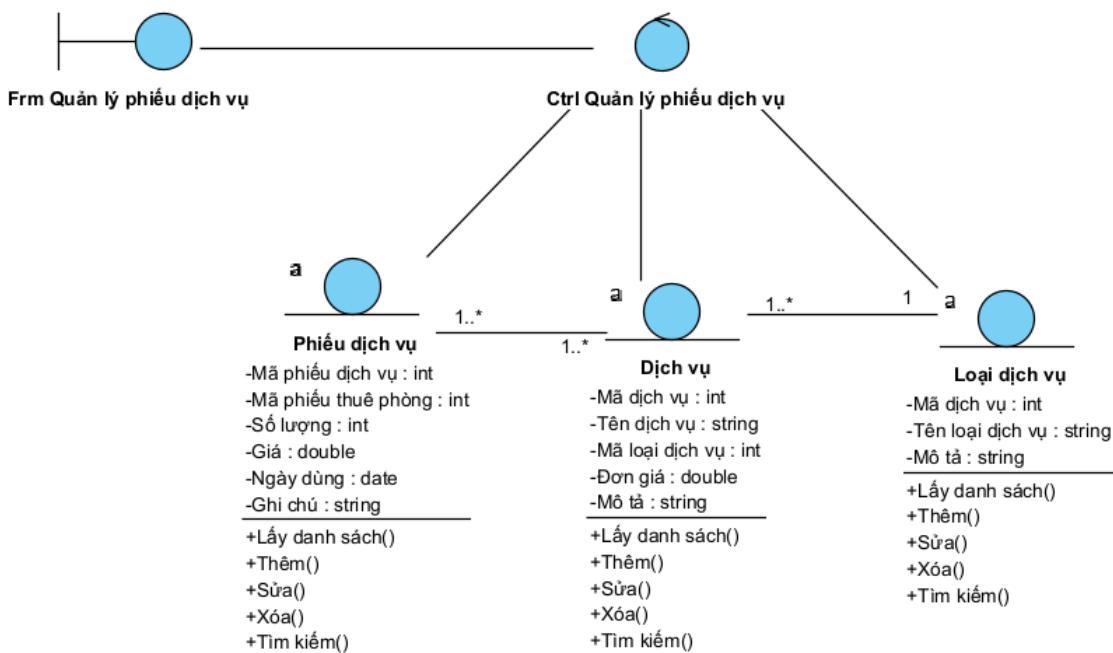
- Boundary: Frm Quản lý phòng
- Control: Ctrl Quản lý phòng
- Entity: **Loại phòng, Phòng**



Hình 3.4 Biểu đồ VOPC của use case quản lý phòng

➤ **Biểu đồ VOPC của use case quản lý phiếu dịch vụ**

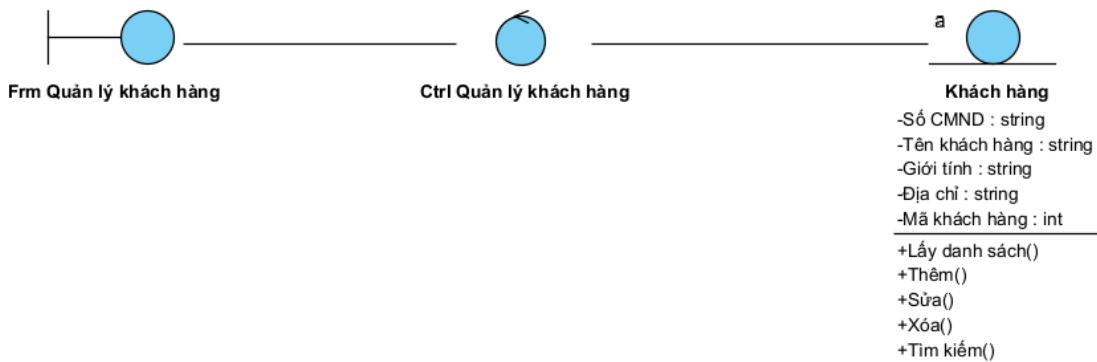
- Boundary: Frm Quản lý phiếu dịch vụ
- Control: Ctrl Quản lý phiếu dịch vụ
- Entity: **Phiếu dịch vụ, Dịch vụ, Loại dịch vụ**



Hình 3.5 Biểu đồ VOPC của use case quản lý phiếu dịch vụ

➤ **Biểu đồ VOPC của use case quản lý khách hàng**

- Boundary: Frm Quản lý khách hàng
- Control: Ctrl Quản lý khách hàng
- Entity: Khách hàng

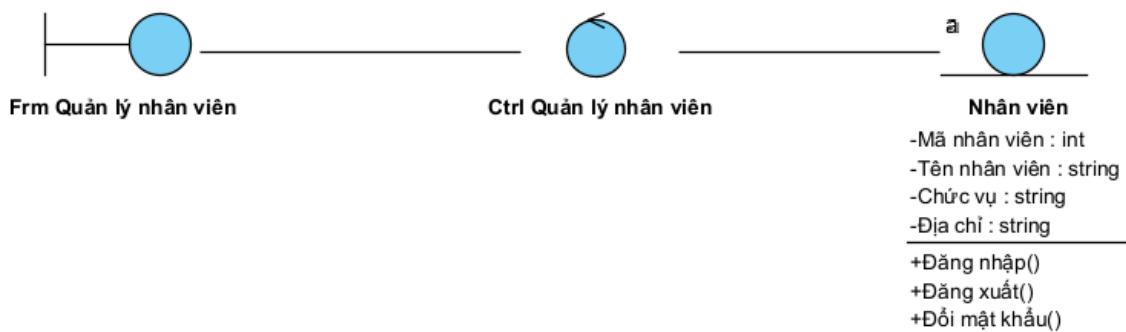


Hình 3.6 Biểu đồ VOPC của use case quản lý khách hàng

➤ **Biểu đồ VOPC của use case quản lý nhân viên**

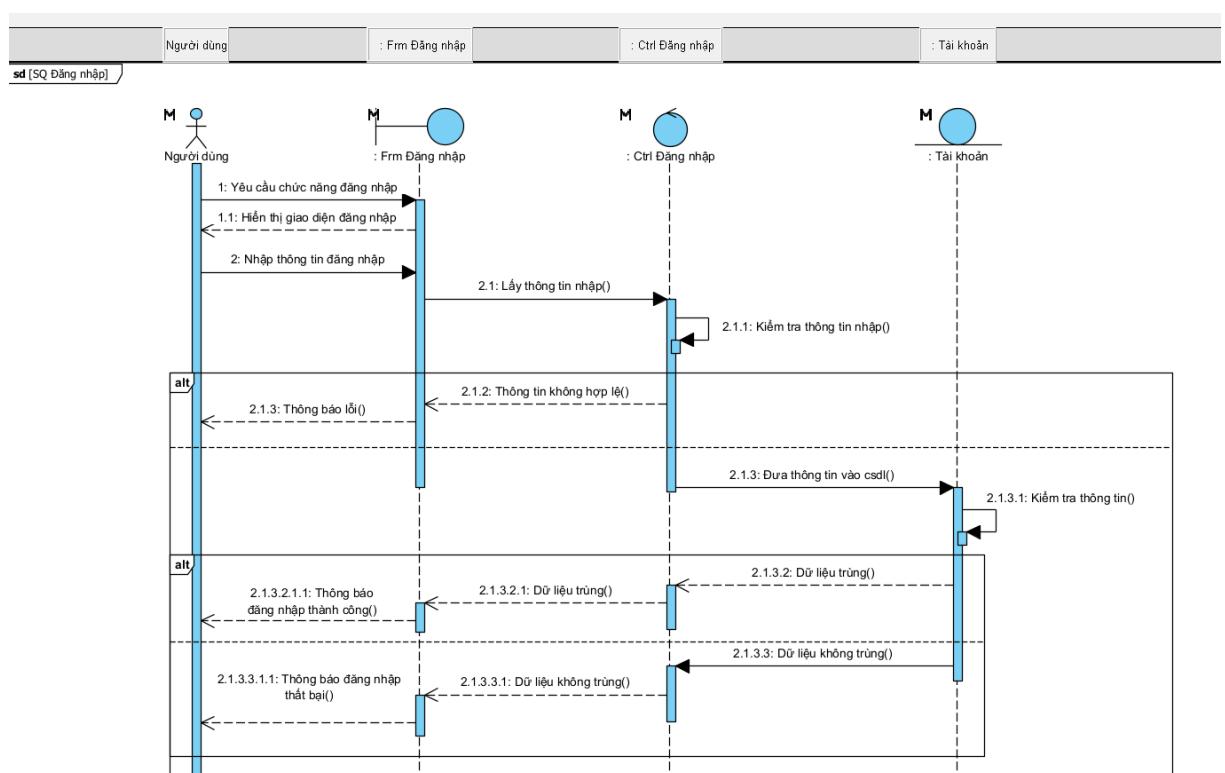
- Boundary: Frm Quản lý nhân viên

- Control: Ctrl Quản lý nhân viên
- Entity: **Nhân viên**



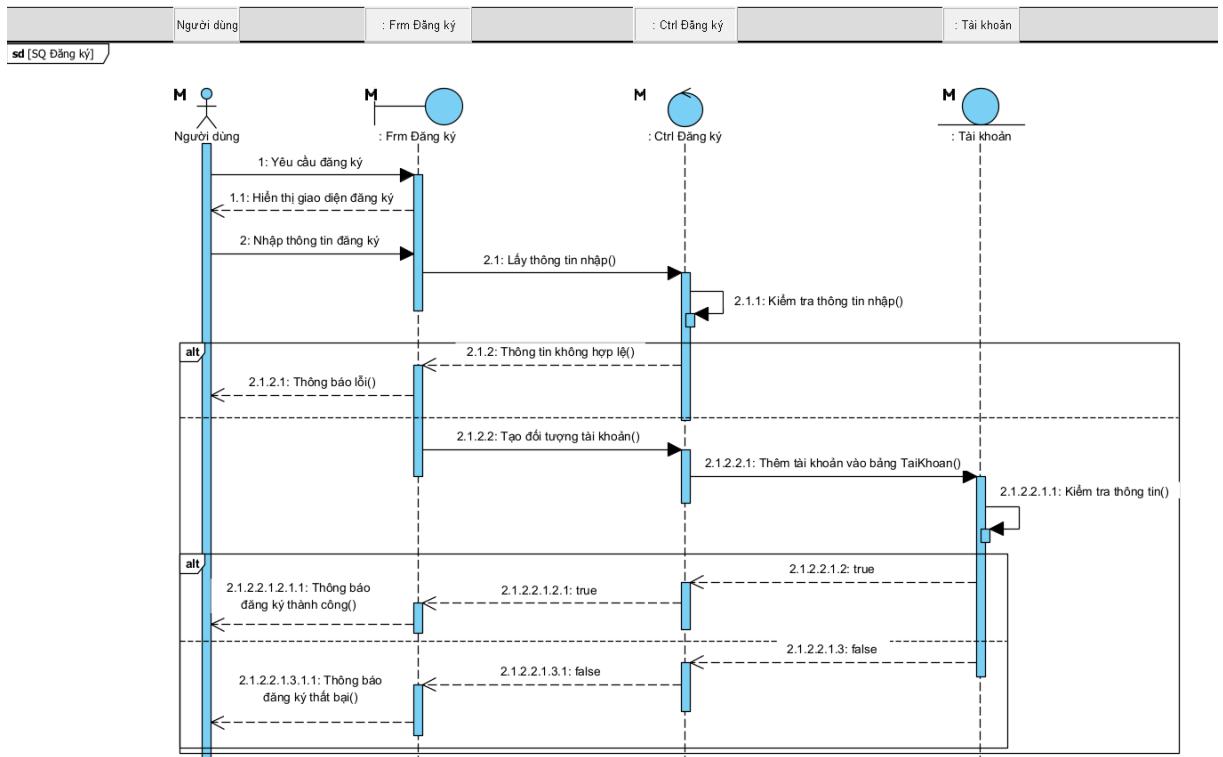
Hình 3.7 Biểu đồ VOPC của use case quản lý nhân viên

b) Biểu đồ tuần tự

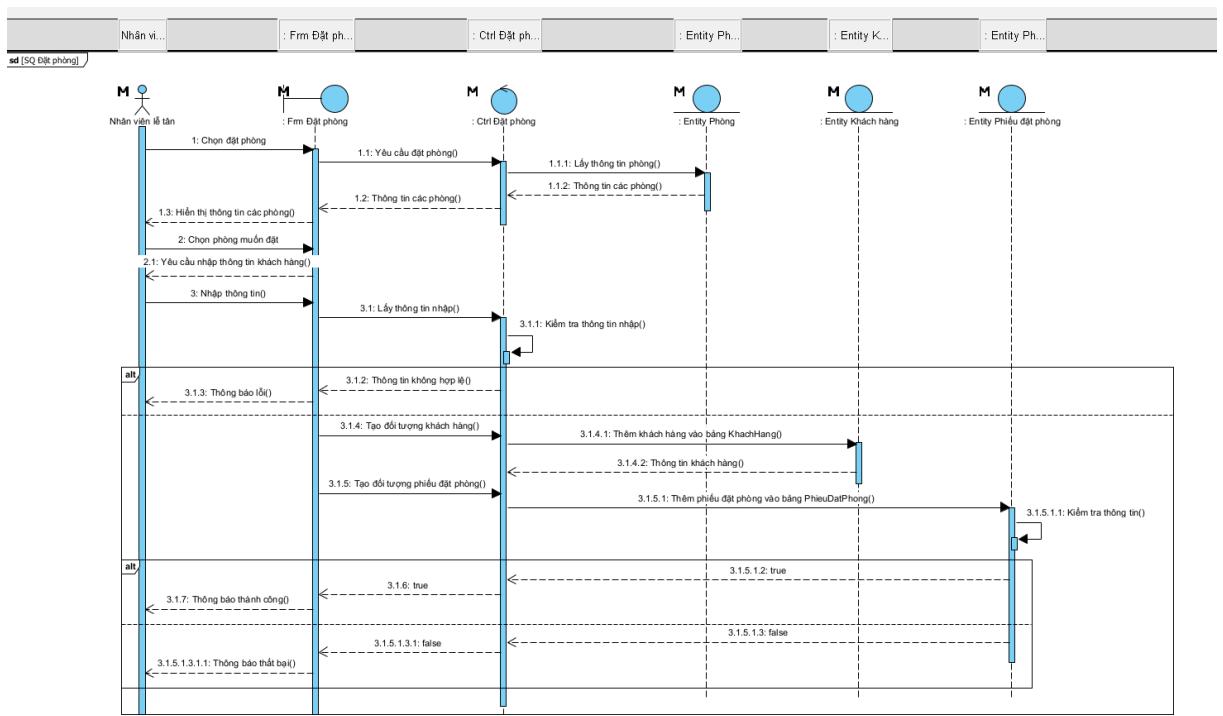


Hình 3.8 Biểu đồ tuần tự đăng nhập

Bài tập lớn: Xây dựng website quản lý khách sạn

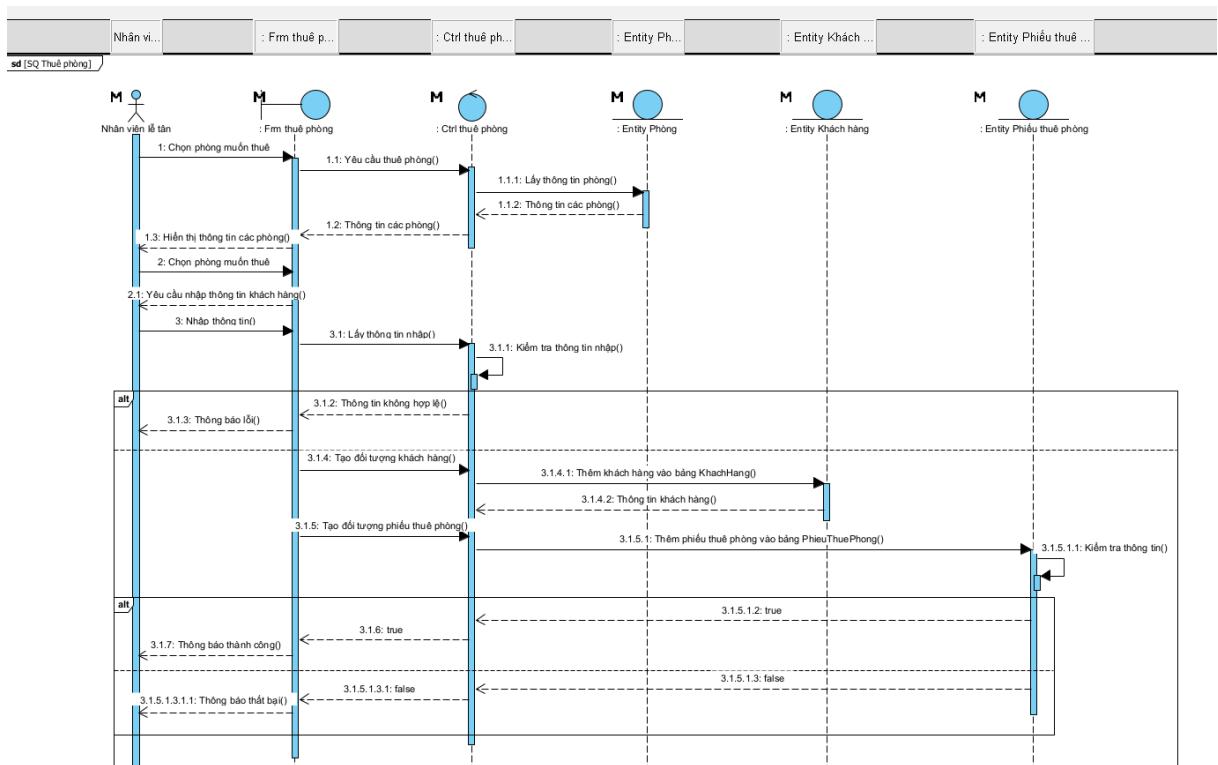


Hình 3.9 Biểu đồ tuần tự đăng ký

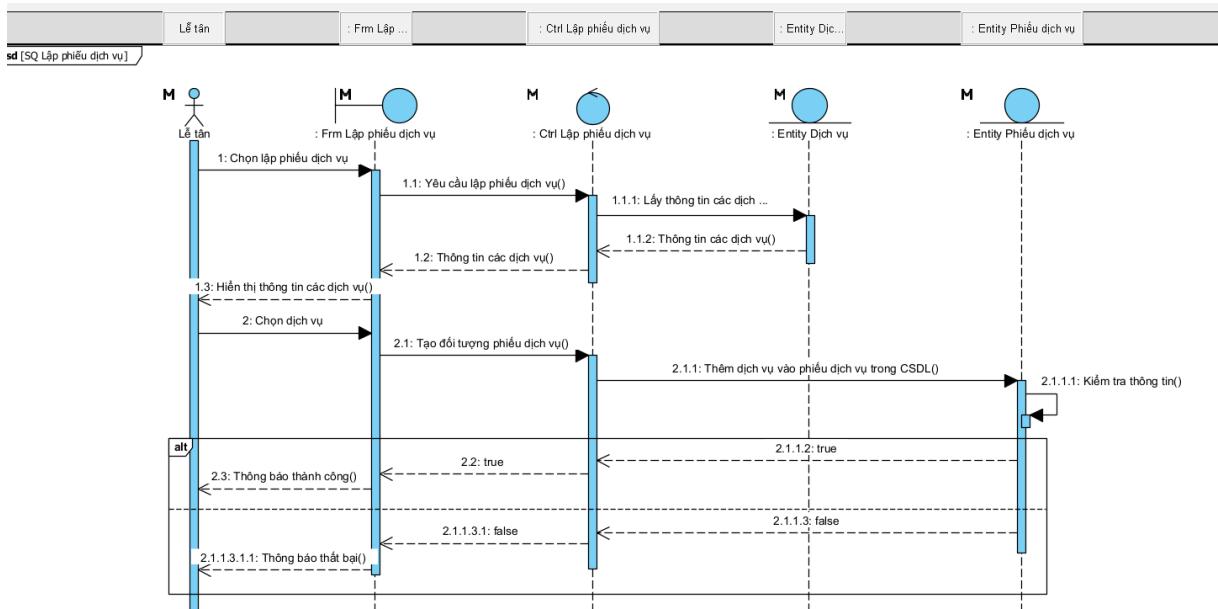


Hình 3.10 Biểu đồ tuần tự đặt phòng

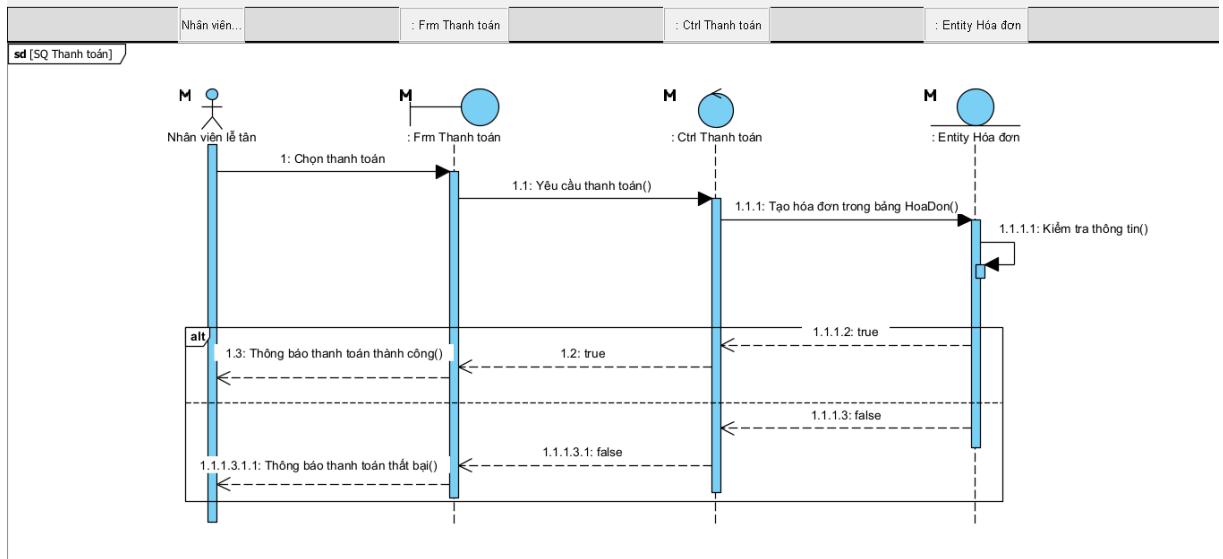
Bài tập lớn: Xây dựng website quản lý khách sạn



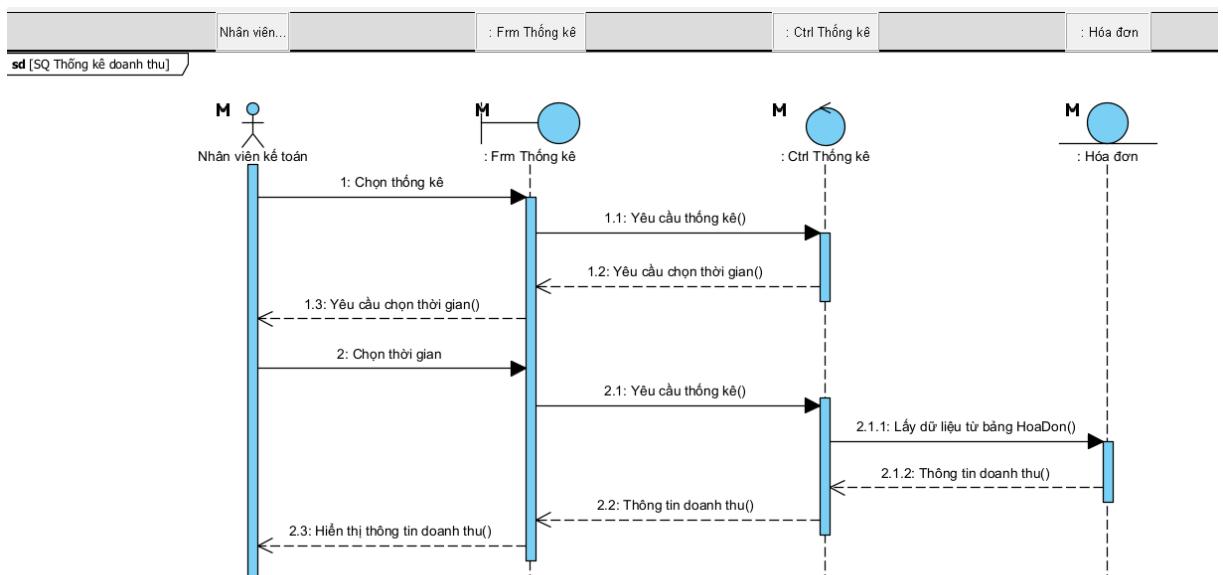
Hình 3.11 Biểu đồ tuần tự thuê phòng



Hình 3.12 Biểu đồ tuần tự lập phiếu dịch vụ

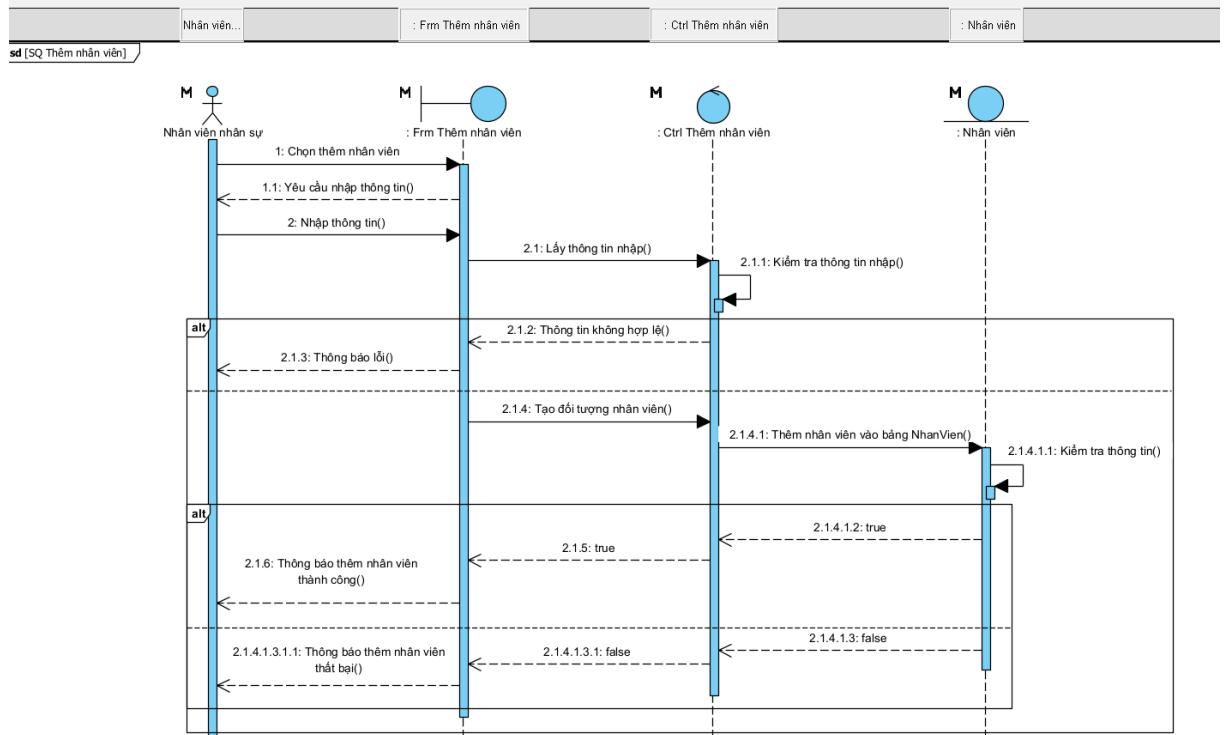


Hình 3.13 Biểu đồ tuần tự thanh toán

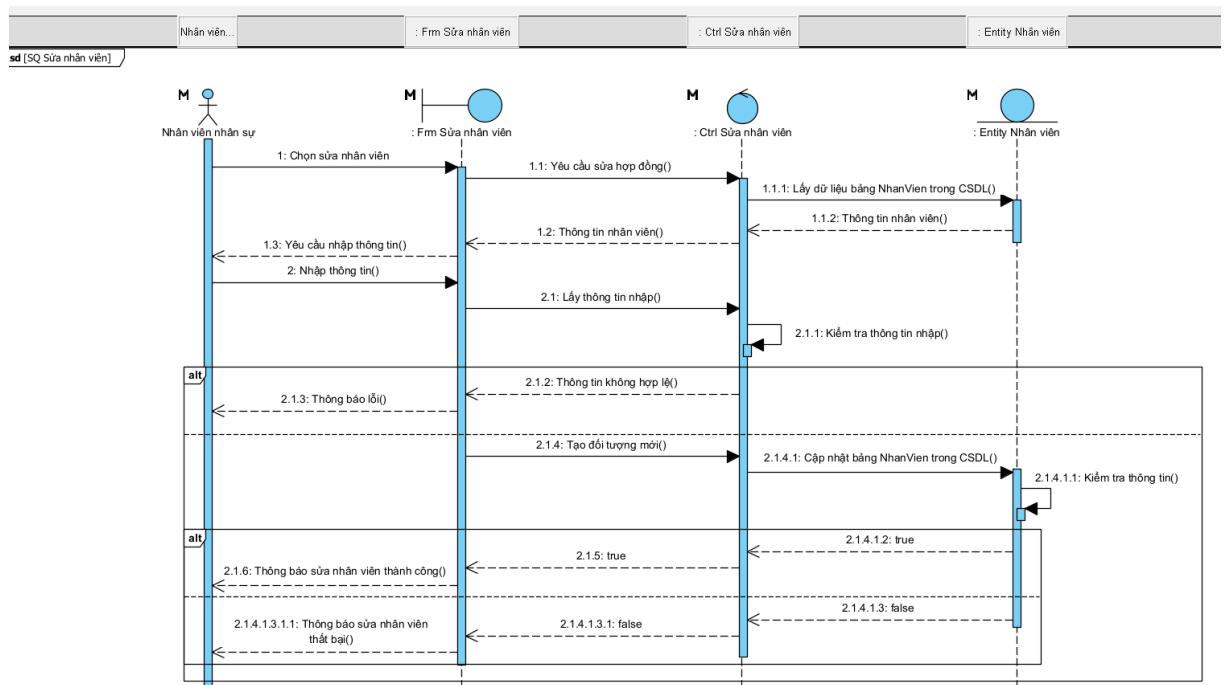


Hình 3.14 Biểu đồ tuần tự thống kê

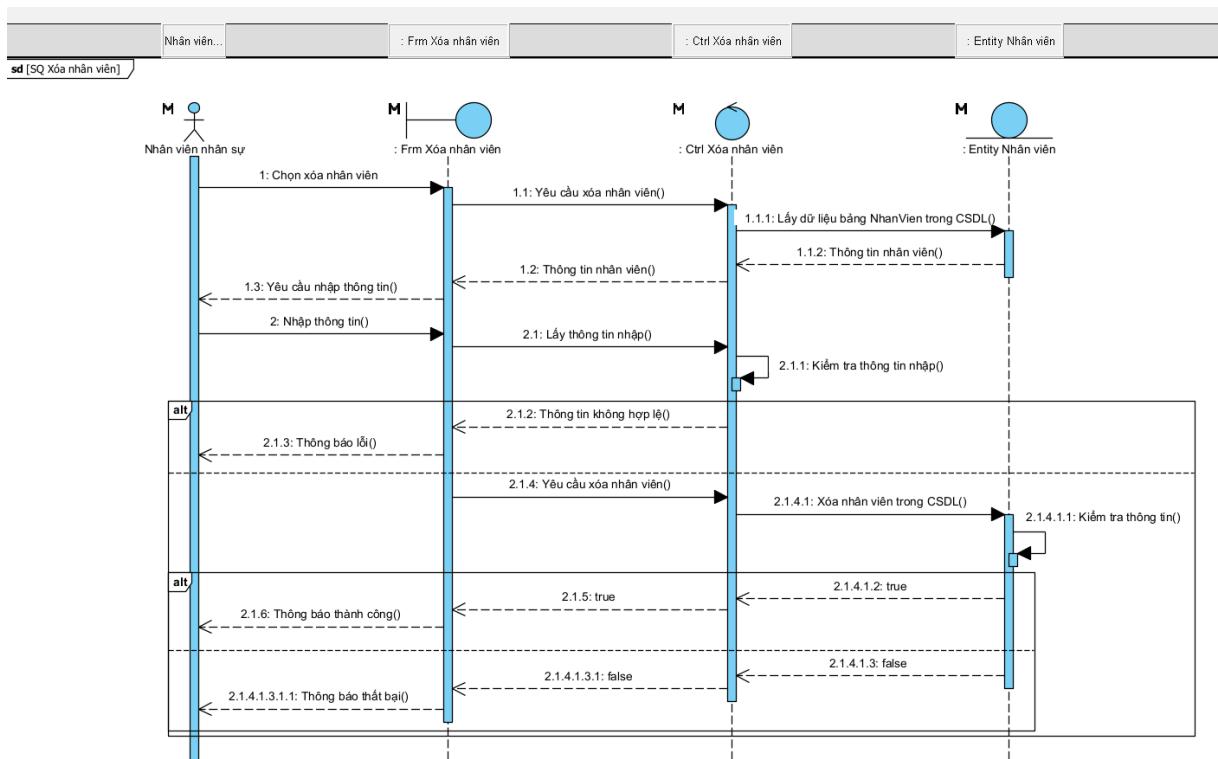
Bài tập lớn: Xây dựng website quản lý khách sạn



Hình 3.15 Biểu đồ tuần tự thêm nhân viên

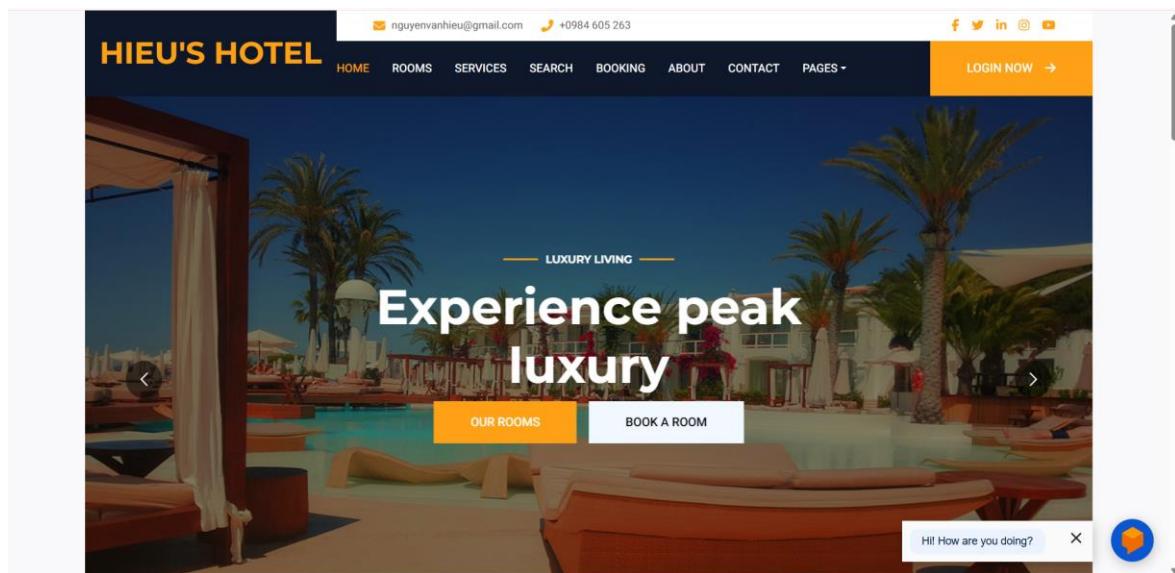


Hình 3.16 Biểu đồ tuần tự sửa nhân viên

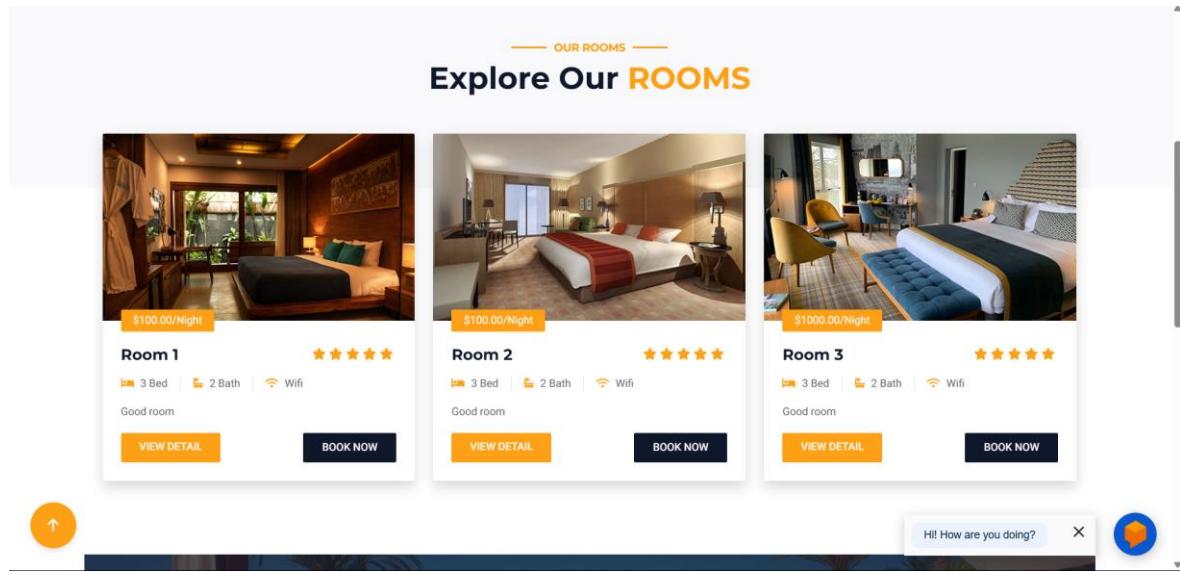


Hình 3.17 Biểu đồ tuần tự xóa nhân viên

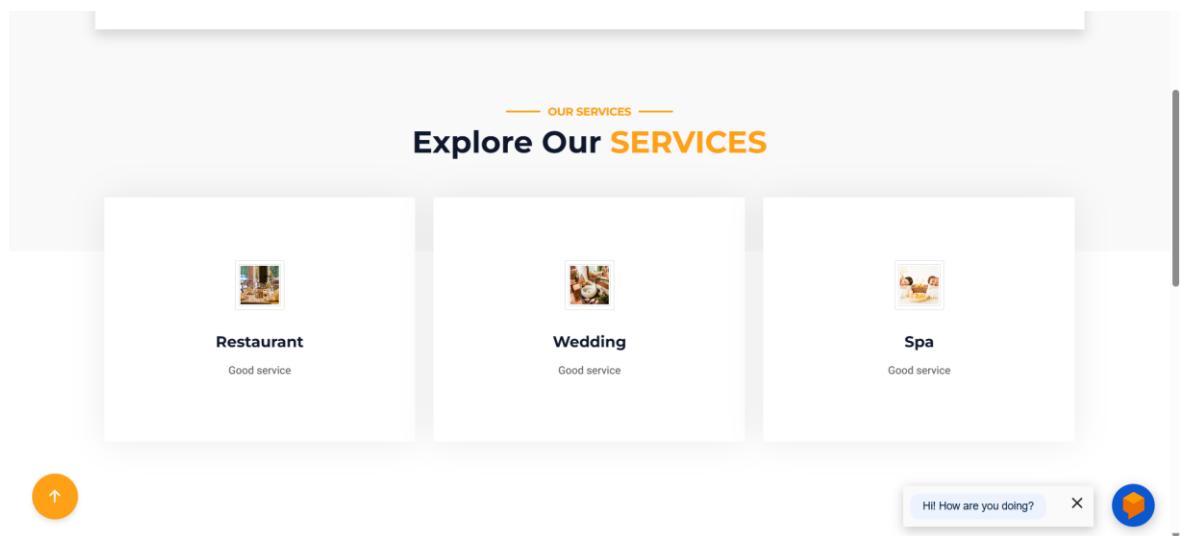
3.2.3 Thiết kế giao diện



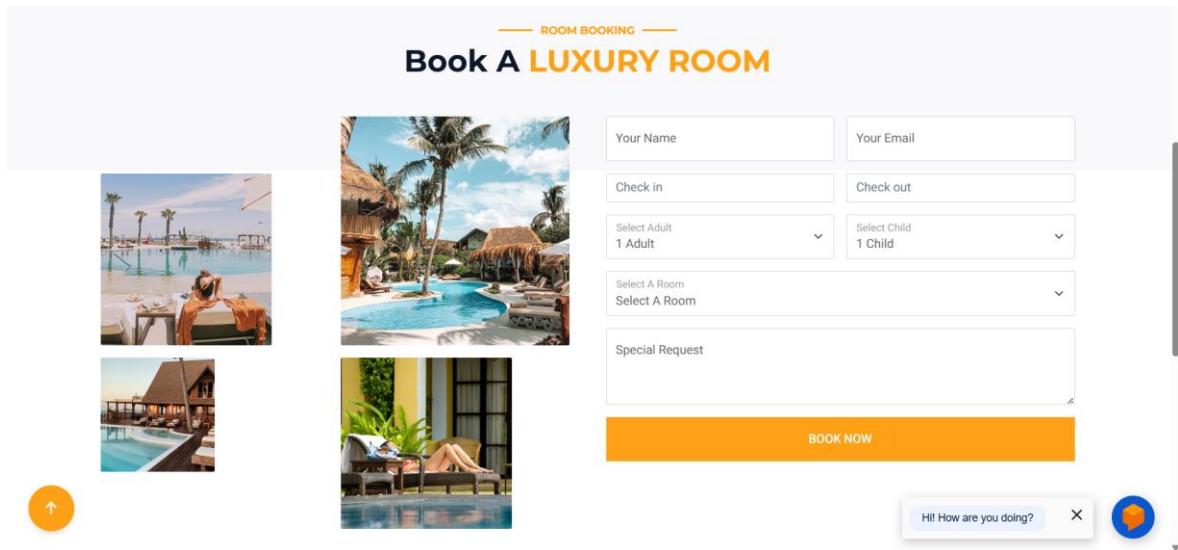
Hình 3.18 Trang chủ



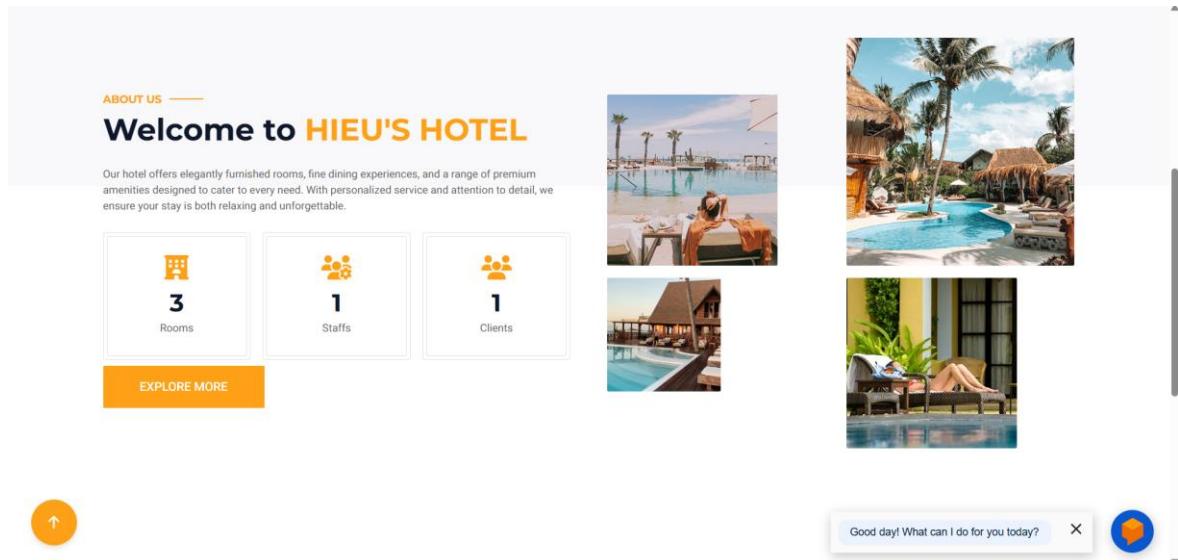
Hình 3.19 Trang danh sách phòng



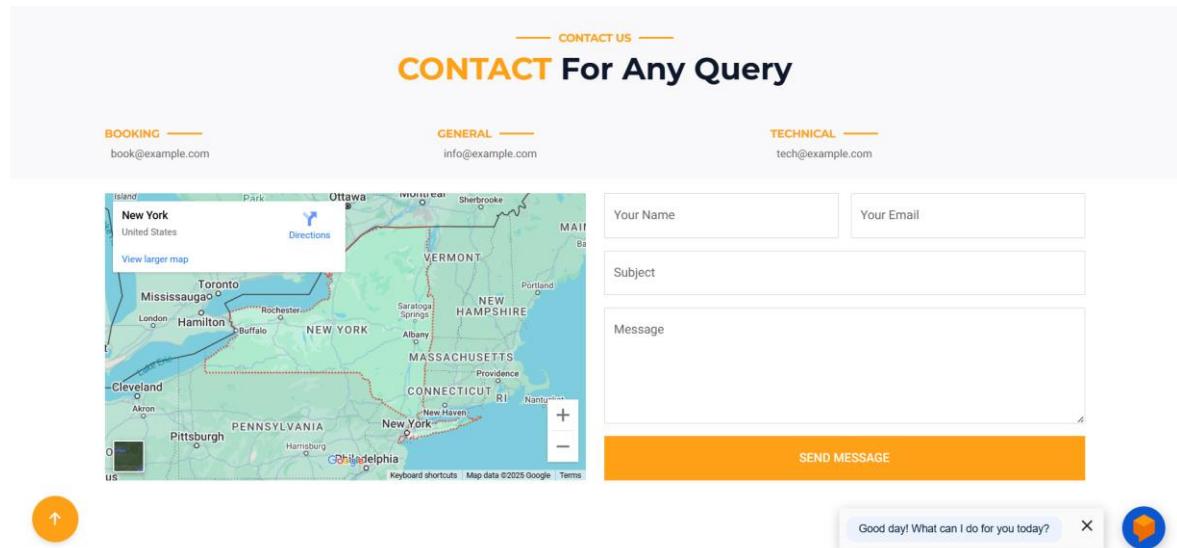
Hình 3.20 Trang dịch vụ



Hình 3.21 Trang đặt phòng



Hình 3.22 Trang thông tin khách sạn



Hình 3.23 Trang liên hệ

The screenshot shows a profile page for a user named "hieu". The page has a red header with the user's name and a profile picture. Below the header, the user's identification number is listed as "Identification Number: 012345". On the right, there is a sidebar titled "Information" containing fields for "Email" (hieu@gmail.com), "Phone" (0123456789), "Gender" (Male), and "Address" (No). At the bottom, there are social media icons for Facebook, Twitter, and Instagram. A blue speech bubble icon with a white location pin and the text "Good day! What can I do for you today?" is located in the bottom right corner.

Hình 3.24 Trang thông tin cá nhân

The screenshot shows a checkout interface for a hotel booking. At the top, it displays "Checkout" with navigation links "HOME / PAGES / CHECKOUT". Below this, a section titled "Unpaid Bookings" lists a single booking entry:

<input type="checkbox"/> Select
Booking ID: 1
Check-in: 2025-03-16T17:00:00.000Z
Check-out: 2025-03-17T17:00:00.000Z
Total: \$10.00
Status: UnPaid

Below the bookings, there is a "Payment Options" section with two radio buttons: "ZaloPay" and "COD". A prominent orange button labeled "Proceed to Payment" is located at the bottom left of this section. In the top right corner of the page, there is a live chat window with the message "Hello! How can I help you?" and a close button.

Hình 3.25 Trang thanh toán

The screenshot shows a "Hotel Review" form. It includes sections for rating and commenting. The ratings are displayed as star icons:

Overall Rating:	★★★★★
Room Quality:	★★★★★
Customer Service:	★★★★★
Hotel Amenities:	★★★★★
Location:	★★★★★

Below the ratings, there is a "Comments/Review:" text area with a rich text editor toolbar above it. An orange "Submit Review" button is located at the bottom left of the form. A file upload section "Upload Images/Videos (Optional):" with a "Choose Files" button and a "No file chosen" message is also present. In the top right corner, there is a live chat window with the message "Hello! How can I help you?" and a close button.

Hình 3.26 Trang đánh giá

Bài tập lớn: Xây dựng website quản lý khách sạn

The screenshot shows the Hotel Management system's dashboard. On the left, a sidebar lists various pages: Dashboard, Account, Staff, User, Room, Room Type, Device, Device Type, Event Type, Event, Event Votes, Service, Service Type, and Service Votes. The 'Dashboard' page is currently selected. The main area displays four cards: 'Booking' (152 total, 24 new since last visit), 'Revenue' (\$2.100, \$52+ since last week), 'Customers' (28441 total, 520 newly registered), and 'Comments' (152 Unread, 85 responded). Below these cards is a section titled 'Recent Booking' with three entries:

Customer	Name	Room Type	Check-in Date	Check-out Date	Status
John Smith	John Smith	Deluxe	2025-05-10	2025-05-15	Cor
Jane Doe	Jane Doe	Standard	2025-05-12	2025-05-14	Per Pay
Peter Jones	Peter Jones	Suite	2025-05-15	2025-05-20	Car

Below the booking table is a chart titled 'Booking Overview' comparing two datasets from January to July. The chart shows two lines: 'First Dataset' (blue) and 'Second Dataset' (green). The second dataset shows significant fluctuations, peaking in May and July.

At the bottom right, there is a 'Notifications' section with a 'TODAY' button.

Hình 3.27 Trang thống kê

The screenshot shows the 'Manage Rooms' page. The sidebar on the left is identical to the one in the previous dashboard screenshot, with 'Room' selected. At the top, there are buttons for '+ New' and 'Delete'. The main area has a search bar labeled 'Search...'. A table titled 'Manage Rooms' lists three room entries:

Room ID	Room Type Id	Image	Price	Floor	MaximumNumberOfGuests	Status	Description	RoomArea
1	1		\$100.00	10	5	Available	Good room	100.00
2	1		\$100.00	3	3	Available	Good room	10.00
3	1		\$1000.00	2	2	Available	Good room	100.00

Hình 3.28 Trang quản lý phòng

Bài tập lớn: Xây dựng website quản lý khách sạn

The screenshot shows a web-based hotel management system. On the left, there is a sidebar menu with the following items:

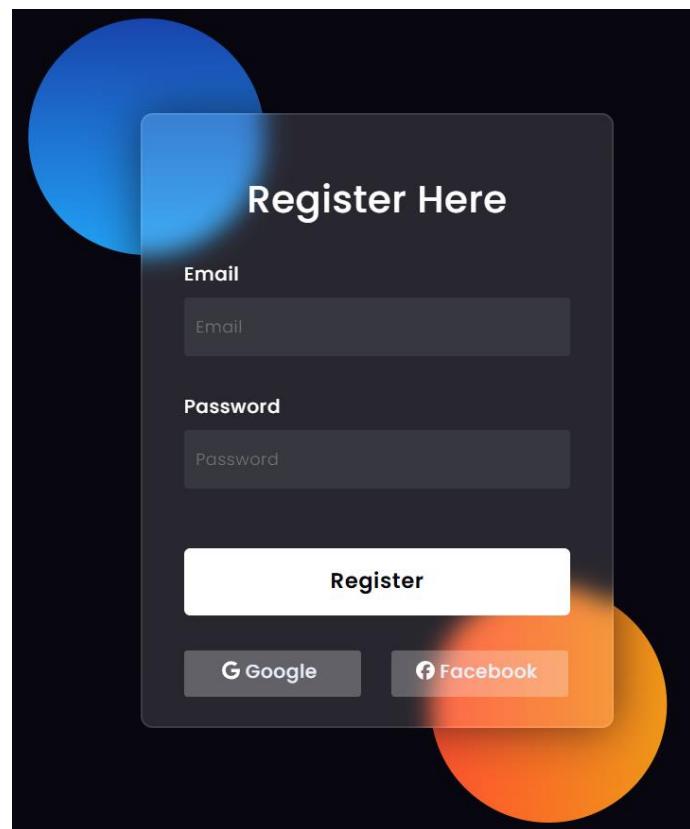
- HOME
- Dashboard
- PAGES
 - Account
 - Staff
 - User
 - Room
 - Room Type
 - Device
 - Device Type
 - Service** (highlighted with a blue border)
 - Service Type
 - Service Votes
 - Bill
 - Booking Votes
 - Rent Room Votes

The main content area is titled "Service Management". It displays a table with three rows of service data:

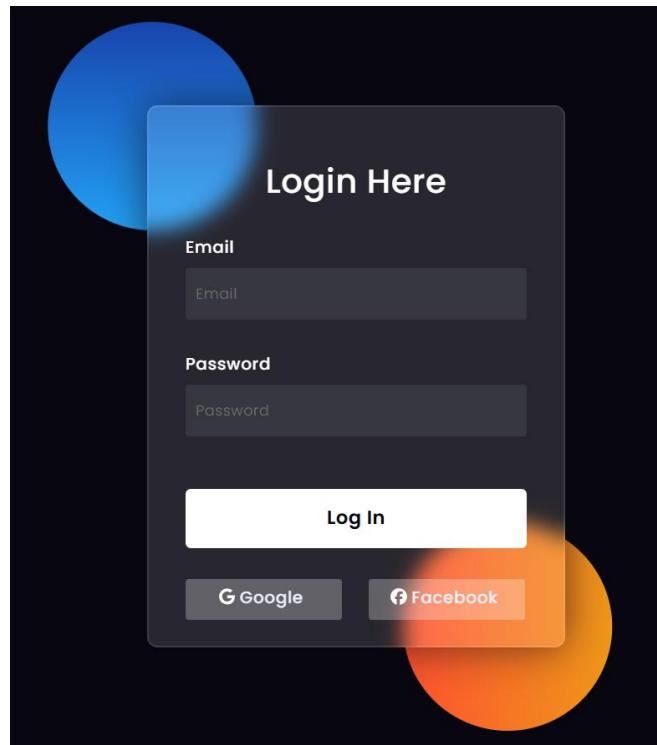
Service ID	Service Name	Service Type Id	Image	Description	Price	Deleted	Actions
1	Restaurant	1		Good service	100 USD	Active	
2	Wedding	2		Good service	1,000 USD	Active	
3	Spa	3		Good service	1,000 USD	Active	

At the bottom right of the main content area, it says "by Nguyen Van Hieu". The URL "localhost:3002/service" is visible at the bottom left.

Hình 3.29 Trang quản lý dịch vụ



Hình 3.30 Trang đăng ký



Hình 3.31 Trang đăng nhập

CHƯƠNG 4: TRIỂN KHAI WEBSITE

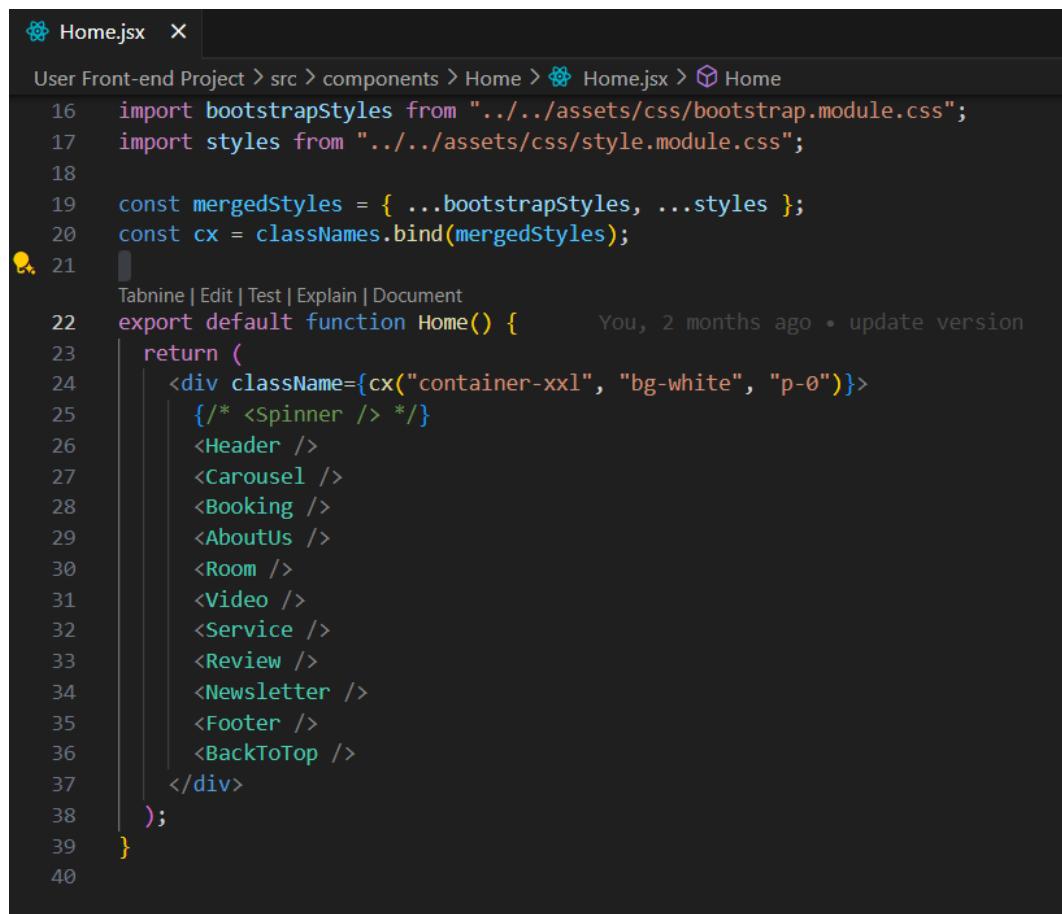
4.1 Triển khai các chức năng cho phân hệ người dùng

Để xây dựng được các chức năng của trang người dùng theo thiết kế đã được trình bày ở chương 3, bài tập lớn đã sử dụng HTML, CSS, và JavaScript, Jquery, ReactJS để thiết kế giao diện và thao tác dữ liệu các trang theo yêu cầu. Tiếp theo, bài tập lớn sẽ trình bày các kỹ thuật được sử dụng để xây dựng các trang.

4.1.1 Trang chủ

a) Phía font end

- Xây dựng bộ cục trang Home bằng các thẻ HTML



```


  Home.jsx  X

User Front-end Project > src > components > Home > Home.jsx > Home
  16 import bootstrapStyles from "../../assets/css/bootstrap.module.css";
  17 import styles from "../../assets/css/style.module.css";
  18
  19 const mergedStyles = { ...bootstrapStyles, ...styles };
  20 const cx = classNames.bind(mergedStyles);
  21
  22 Tabnine | Edit | Test | Explain | Document
  23 export default function Home() {
  24   return (
  25     <div className={cx("container-xxl", "bg-white", "p-0")}>
  26       {/* <Spinner /> */}
  27       <Header />
  28       <Carousel />
  29       <Booking />
  30       <AboutUs />
  31       <Room />
  32       <Video />
  33       <Service />
  34       <Review />
  35       <Newsletter />
  36       <Footer />
  37       <BackToTop />
  38     </div>
  39   );
  40 }


```

Về mặt cấu tạo html, trang chủ gồm 2 phần chính là head và body. Phần phần head dùng cặp thẻ Phân head là phần cấu hình của một trang web, nơi khai báo các thư viện, link, các thẻ SEO(<meta>). Phần body gồm các thẻ để tạo nên cấu trúc của trang web dùng cặp thẻ <body></body>. Thân trang web (body) lại được chia thành 3 phần header, main, footer, các thẻ này đều được hỗ trợ trong phiên bản HTML5.

Header sử dụng cặp thẻ <header></header>. Phần này chứa navigation bar (<nav>) và tools bar. Navigation bar là cặp thẻ <nav></nav> bên trong có chứa thẻ và tập các thẻ để định dạng các phần tử của menu. Tuy nhiên mặc định các thẻ sẽ mặc định sắp xếp theo chiều dọc và các phần tử trong sẽ có định dạng như margin, padding, và list-style. Như vậy để menu hiển thị theo chiều ngang cần sử dụng display: flex và list-style: none bỏ đi các định dạng của .

Bên trong các thẻ còn có các menu con khác các menu con này cũng được định dạng giống như trên. Tuy nhiên chúng sẽ được bổ sung thêm các hậu tố (pseudo class) “hover” để tạo hiệu ứng xuất hiện các menu con khi trỏ vào từng item của menu cha. Khi click chuột vào các item, trang web sẽ chuyển hướng, như vậy bên trong nội dung các thẻ còn chứa các cặp thẻ <a> và thuộc tính href chứa các đường dẫn chuyển hướng.

➤ Kỹ thuật định dạng bằng CSS

```
102
103  /*** Header ***/
104  .carousel-caption {
105    top: 0;
106    left: 0;
107    right: 0;
108    bottom: 0;
109    background: □rgba(15, 23, 43, 0.7);
110    z-index: 1;
111  }
112
113  .carousel-control-prev,
114  .carousel-control-next {
115    width: 10%;
116  }
117
118  .carousel-control-prev-icon,
119  .carousel-control-next-icon {
120    width: 3rem;
121    height: 3rem;
122  }
123
124  @media (max-width: 768px) {
125    #header-carousel .carousel-item {
126      position: relative;
127      min-height: 450px;
128    }
129  }
```

- Sử dụng Javascript, Jquery, ReactJS để lập trình các chức năng

```

25      Tabnine | Edit | Test | Explain | Document
26  export default function Header() {
27      const [isHovered, setIsHovered] = useState(false);
28      const [isDesktop, setIsDesktop] = useState(window.innerWidth >= 992);
29
30      useEffect(() => {
31          const handleResize = () => {
32              setIsDesktop(window.innerWidth >= 992);
33          };
34          window.addEventListener("resize", handleResize);
35          return () => window.removeEventListener("resize", handleResize);
36      }, []);
37
38      const handleMouseEnter = () => {
39          if (isDesktop) setIsHovered(true);
40      };
41
42      const handleMouseLeave = () => {
43          if (isDesktop) setIsHovered(false);
44      };
45
46      const handleClick = () => {
47          setIsHovered((prev) => !prev);
48      };
49

```

b) Phía backend

- Triển khai các lớp tầng DataAccess

```

1  import { executeMysqlQuery } from "../config/db";
2  import Service from "../models/service";
3  import { serviceSchema } from "../../schemas/service";
4
5  > export const getService = async (req, res) => { ... }
19  };
20
21 > export const getServiceById = async (req, res) => { ... }
37  };
38
39 > export const createService = async (req, res) => { ... }
64  };
65
66 > export const updateService = async (req, res) => { ... }
92  };
93
94 > export const deleteService = async (req, res) => { ... }
106  };
107

```

- Triển khai lớp tầng Bussiness

```
1 import { executeMysqlQuery } from "../config/db";
2 import Service from "../models/service";
3 import { serviceSchema } from "../../schemas/service";
4
5 > export const getService = async (req, res) => { ...
19   };
20
21 > export const getServiceById = async (req, res) => { ...
37   };
38
39 > export const createService = async (req, res) => { ...
64   };
65
66 > export const updateService = async (req, res) => { ...
92   };
93
94 > export const deleteService = async (req, res) => { ...
106   };
107
```

➤ Triển khai lớp tầng Controller

```
1 import express from "express";          You, 2 months ago • upda...
2
3 import {
4   getService,
5   getServiceById,
6   createService,
7   updateService,
8   deleteService,
9 } from "../controllers/service";
10
11 const router = express.Router();
12
13 router.get("/service/get-all", getService);
14 router.get("/service/get-data-by-id/:id", getServiceById);
15 router.post("/service/create", createService);
16 router.put("/service/update", updateService);
17 router.delete("/service/delete/:id", deleteService);
18
19 export default router;
20
```

4.1.2 Trang xem phòng

a) Phía frontend

- Xây dựng bộ cục trang xem phòng bằng các thẻ HTML

```
1 import Header from "../Header/Header";           You, la
2 import PageHeader from "../PageHeader/PageHeader";
3 import Booking from "../Booking/Booking";
4 import Room from "../Room/Room";
5 import Review from "../Review/Review";
6 import Newsletter from "../Newsletter/Newsletter";
7 import Footer from "../Footer/Footer";
8 import BackToTop from "../BackToTop/BackToTop";
9
10 Tabnine | Edit | Test | Explain | Document
11 export default function RoomPage() {
12   const pageHeaderProps = {
13     title: "Rooms",
14   };
15
16   return (
17     <div>
18       <Header />
19       <PageHeader {...pageHeaderProps} />
20       <Booking />
21       <Room />
22       <Review />
23       <Newsletter />
24       <Footer />
25       <BackToTop />
26     </div>
27   );
28 }
```

- Kỹ thuật định dạng bằng CSS

```

163   .section-title::before {
164     position: absolute;
165     content: "";
166     width: 45px;
167     height: 2px;
168     top: 50%;
169     left: -55px;
170     margin-top: -1px;
171     background: var(--primary);
172   }
173
174   .section-title::after {
175     position: absolute;
176     content: "";
177     width: 45px;
178     height: 2px;
179     top: 50%;
180     right: -55px;
181     margin-top: -1px;
182     background: var(--primary);
183   }
184
185
186   .section-title.text-start::before,
187   .section-title.text-end::after {
188     display: none;
189   }

```

- Sử dụng Javascript, Jquery, ReactJS để lập trình các chức năng

```

24  export default function Room({ filters }) {
32    const fetchRooms = async () => {
33      try {
34        const response = await axios.get(
35          "http://localhost:3000/api/rooms/get-all"
36        );
37        let data = response.data;
38        data = filterRooms(data, filters);
39        setRooms(data);
40      } catch (error) {
41        console.error("Error fetching rooms:", error);
42      }
43    };
44
45 >  const filterRooms = (rooms, filters) => { ...
120
121
122  const BookNowButton = ({ roomProps }) => {
123    BookNowButton.propTypes = {
124      roomProps: PropTypes.object.isRequired,
125    };
126    const user = JSON.parse(localStorage.getItem("user"));
127    // console.log("Account Id: ", user.account.AccountId);
128
129 >  if (!user) { ...
137
138
139 >  const handleBookNow = async () => { ...
180

```

b) Phía backend

- Triển khai các lớp tầng DataAccess

```
you, 2 weeks ago | 1 author (you)
1 import { executeMySQLQuery } from "../config/db";
2 import Room from "../models/room";
3 import { roomSchema } from "../schemas/room";
4
5 > export const getAllRooms = async (req, res) => { ...
14   };
15
16 > export const getRoomById = async (req, res) => { ...
27   };
28
29 > export const createRoom = async (req, res) => {
58   };
59
60 > export const updateRoom = async (req, res) => { ...
90   };
91
92 > export const deleteRoom = async (req, res) => { ...
102   };
103
```

- Triển khai lớp tầng Bussiness

```
you, 2 weeks ago | 1 author (you)
1 import { executeMySQLQuery } from "../config/db";
2 import Room from "../models/room";
3 import { roomSchema } from "../schemas/room";
4
5 > export const getAllRooms = async (req, res) => { ...
14   };
15
16 > export const getRoomById = async (req, res) => { ...
27   };
28
29 > export const createRoom = async (req, res) => {
58   };
59
60 > export const updateRoom = async (req, res) => { ...
90   };
91
92 > export const deleteRoom = async (req, res) => { ...
102   };
103
```

- Triển khai lớp tầng Controller

```
You, last month | 1 author (You)
1 import express from "express";
2
3 import {
4   getAllRooms,
5   getRoomById,
6   createRoom,
7   updateRoom,
8   deleteRoom,
9 } from "../controllers/room.js";
10
11 const router = express.Router();      You, last month
12
13 router.get("/rooms/get-all", getAllRooms);
14 router.get("/rooms/get-data-by-id/:id", getRoomById);
15 router.post("/rooms/create", createRoom);
16 router.put("/rooms/update", updateRoom);
17 router.delete("/rooms/delete/:id", deleteRoom);
18
19 export default router;
20
```

4.1.3 Trang xem dịch vụ

a) Phía frontend

- Xây dựng bộ cục trang xem dịch vụ bằng các thẻ HTML

```
You, 4 days ago | Author (rou) You, last
1 import Header from "../Header/Header";
2 import PageHeader from "../PageHeader/PageHeader";
3 import Booking from "../Booking/Booking";
4 import Service from "../Service/Service";
5 import Review from "../Review/Review";
6 import Newsletter from "../Newsletter/Newsletter";
7 import Footer from "../Footer/Footer";
8 import BackToTop from "../BackToTop/BackToTop";
9
10 Tabnine | Edit | Test | Explain | Document
11 export default function ServicePage() {
12   const pageHeaderProps = {
13     title: "Services",
14   };
15
16   return (
17     <div>
18       <Header />
19       <PageHeader {...pageHeaderProps} />
20       <Booking />
21       <Service />
22       <Review />
23       <Newsletter />
24       <Footer />
25       <BackToTop />
26     </div>
27   );
28 }
```

- Kỹ thuật định dạng bằng CSS

```

190
191  /*** Service ***/
192  .service-item {
193    height: 320px;
194    padding: 30px;
195    display: flex;
196    flex-direction: column;
197    justify-content: center;
198    text-align: center;
199    background: #ffffff;
200    box-shadow: 0 0 45px rgba(0, 0, 0, 0.08);
201    transition: 0.5s;
202  }
203
204  .service-item:hover {
205    background: var(--primary);
206  }
207
208  .service-item .service-icon {
209    margin: 0 auto 30px auto;
210    width: 65px;
211    height: 65px;
212    transition: 0.5s;
213  }
214
215  .service-item i,
216  .service-item h5,
217  .service-item p {
218    transition: 0.5s;
219  }

```

- Sử dụng Javascript, Jquery, ReactJS để lập trình các chức năng

```

18
19  const cx = classNames.bind({ ...bootstrapstyles, ...styles, ...serviceStyles });
20
21  Tabnine | Edit | Test | Explain | Document
22  export default function Service() {
23    const [services, setServices] = useState([]);
24
25    useEffect(() => {
26      AOS.init({ duration: 3000 });
27      fetchServices();
28    }, []);
29
30    const fetchServices = () => {
31      axios
32        .get("http://localhost:3000/api/service/get-all")
33        .then((response) => setServices(response.data))
34        .catch((error) => console.error(error));
35    };

```

b) Phía backend

- Triển khai các lớp tầng DataAccess

```
You, 3 weeks ago | 1 author (You)
1 import { executeMysqlQuery } from "../config/db";
2 import Service from "../models/service";
3 import { serviceSchema } from "../../schemas/service";
4
5 > export const getService = async (req, res) => {
19   };
20
21 > export const getServiceById = async (req, res) => {
37   };
38
39 > export const createService = async (req, res) => {
64   };
65
66 > export const updateService = async (req, res) => {
92   };
93
94 > export const deleteService = async (req, res) => {
106   };
107
```

- Triển khai lớp tầng Bussiness

```
You, 3 weeks ago | 1 author (You)
1 import { executeMysqlQuery } from "../config/db";
2 import Service from "../models/service";
3 import { serviceSchema } from "../../schemas/service";
4
5 > export const getService = async (req, res) => {
19   };
20
21 > export const getServiceById = async (req, res) => {
37   };
38
39 > export const createService = async (req, res) => {
64   };
65
66 > export const updateService = async (req, res) => {
92   };
93
94 > export const deleteService = async (req, res) => {
106   };
107
```

- Triển khai lớp tầng Controller

```
1 import express from "express";          You, 2 months ago • upda
2
3 import {
4   getService,
5   getServiceById,
6   createService,
7   updateService,
8   deleteService,
9 } from "../controllers/service";
10
11 const router = express.Router();
12
13 router.get("/service/get-all", getService);
14 router.get("/service/get-data-by-id/:id", getServiceById);
15 router.post("/service/create", createService);
16 router.put("/service/update", updateService);
17 router.delete("/service/delete/:id", deleteService);
18
19 export default router;
20
```

4.2 Triển khai các chức năng cho phân hệ quản trị nội dung

4.2.1 Trang quản lý nhân viên

a) Phía frontend

- Xây dựng bộ cục trang quản lý nhân viên bằng các thẻ HTML

```
16  export default function Staff() {  
366    );  
367  
368    return (  
369      <div> You, last month • update version ...  
370      <Toast ref={toast} />  
371      <Toolbar  
372        className="mb-4"  
373        left={leftToolbarTemplate}  
374        right={rightToolbarTemplate}  
375      />  
376  
377    > <DataTable ...  
440      </DataTable>  
441  
442      /* Dialog Add/Edit */  
443    > <Dialog ...  
590      </Dialog>  
591  
592      /* Dialog Confirm Delete */  
593    > <Dialog ...  
612      </Dialog>  
613      </div>  
614    );  
615  }  
616
```

➤ Kỹ thuật định dạng bằng CSS

```
1  import React, { useState, useEffect, useRef } from "react";  6.9k (gzipped: 2.7k)  
2  import { DataTable } from "primereact/datatable";  236.6k (gzipped: 57.6k)  
3  import { Column } from "primereact/column";  291 (gzipped: 213)  
4  import { Button } from "primereact/button";  55.9k (gzipped: 15.8k)  
5  import { InputText } from "primereact/inputtext";  53k (gzipped: 15.1k)  
6  import { InputNumber } from "primereact/inputnumber";  72.8k (gzipped: 19.7k)  
7  import { InputTextarea } from "primereact/inputtextarea";  53.6k (gzipped: 15.3k)  
8  import { Calendar } from "primereact/calendar";  124.3k (gzipped: 33.5k)  
9  import { Dropdown } from "primereact/dropdown";  105.6k (gzipped: 27.8k)  
10 import { FileUpload } from "primereact/fileupload";  90.7k (gzipped: 24.7k)  
11 import { Dialog } from "primereact/dialog";  64.4k (gzipped: 18.7k)  
12 import { Toolbar } from "primereact/toolbar";  28.9k (gzipped: 8.3k)  
13 import { Toast } from "primereact/toast";  63.1k (gzipped: 18k)  
14 import axios from "axios";  62.6k (gzipped: 23.1k)
```

➤ Sử dụng Javascript, Jquery, ReactJS để lập trình các chức năng

```

  |> inline | Edit | Test | Explain | Document
  export default function Staff() {
    const [staffs, setStaffs] = useState([]);
    > const [staff, setStaff] = useState({ ...
      });
    const [staffDialog, setStaffDialog] = useState(false);
    const [deleteStaffDialog, setDeleteStaffDialog] = useState(false);
    const [selectedStaffs, setSelectedStaffs] = useState(null);
    const [globalFilter, setGlobalFilter] = useState("");
    const toast = useRef(null);
    const token = "";

    useEffect(() => {
      | fetchStaffs();
    }, [ ]);

    > const fetchStaffs = () => { ... };
    > const openNew = () => { ... };
    > const hideDialog = () => setStaffDialog(false);
    const hideDeleteDialog = () => setDeleteStaffDialog(false);

    > const validateStaff = () => { ... };
  }

```

b) Phía backend

- Triển khai các lớp tầng DataAccess

```

  |> 1 sec ago | author (100)
  1 < import bcryptjs from "bcryptjs";  22.1k (gzipped: 9.9k)
  2
  3   import { executeMysqlQuery } from "../config/db.js";
  4   import Staff from "../models/staff";
  5   import { staffSchema } from "../schemas/staff";
  6
  7 > export const getAllStaff = async (req, res) => { ...
  18   };
  19
  20 > export const getStaffById = async (req, res) => { ...
  32   };
  33
  34 > export const createStaff = async (req, res) => { ...
  97   };
  98
  99 > export const updateStaff = async (req, res) => { ...
L37   };
L38 |
L39 > export const deleteStaff = async (req, res) => { ...
L50   };
L51 |

```

➤ Triển khai lớp tầng Bussiness

```
1 ✓ import bcryptjs from "bcryptjs"; 22.1k (gzipped: 9.9k)
2
3   import { executeMysqlQuery } from "../config/db.js";
4   import Staff from "../models/staff";
5   import { staffSchema } from "../schemas/staff";
6
7 > export const getAllStaff = async (req, res) => {
18   };
19
20 > export const getStaffById = async (req, res) => {
32   };
33
34 > export const createStaff = async (req, res) => {
97   };
98
99 > export const updateStaff = async (req, res) => {
137   };
138 |
139 > export const deleteStaff = async (req, res) => {
150   };
151 |
```

➤ Triển khai lớp tầng Controller

```
You, 2 months ago | Author (You) You, 2 months ago •
```

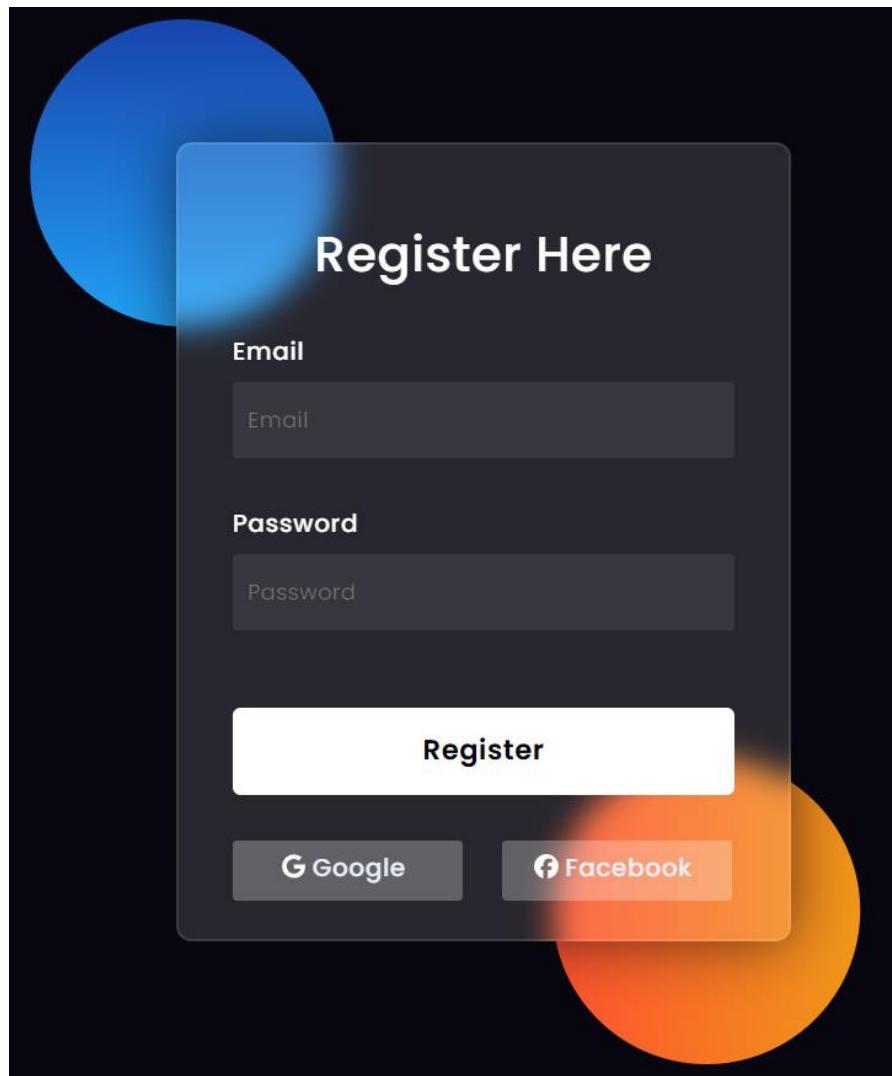
```
1 import express from "express";  
2  
3 import {  
4   getAllStaff,  
5   getStaffById,  
6   createStaff,  
7   updateStaff,  
8   deleteStaff,  
9 } from "../controllers/staff.js";  
10  
11 const router = express.Router();  
12  
13 router.get("/staff/get-all", getAllStaff);  
14 router.get("/staff/get-data-by-id/:id", getStaffById);  
15 router.post("/staff/create", createStaff);  
16 router.put("/staff/update", updateStaff);  
17 router.delete("/staff/delete/:id", deleteStaff);  
18  
19 export default router;  
20
```

4.3 Kiểm thử và triển khai ứng dụng

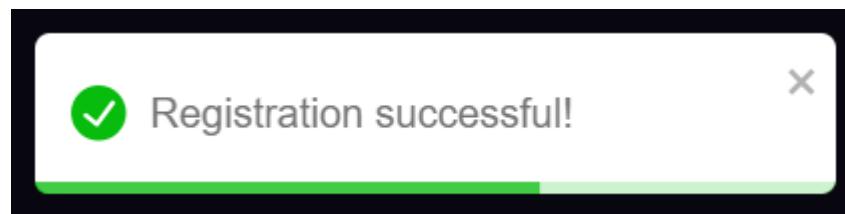
4.3.1 Kiểm thử

- Chức năng: Đăng ký tài khoản

- Input

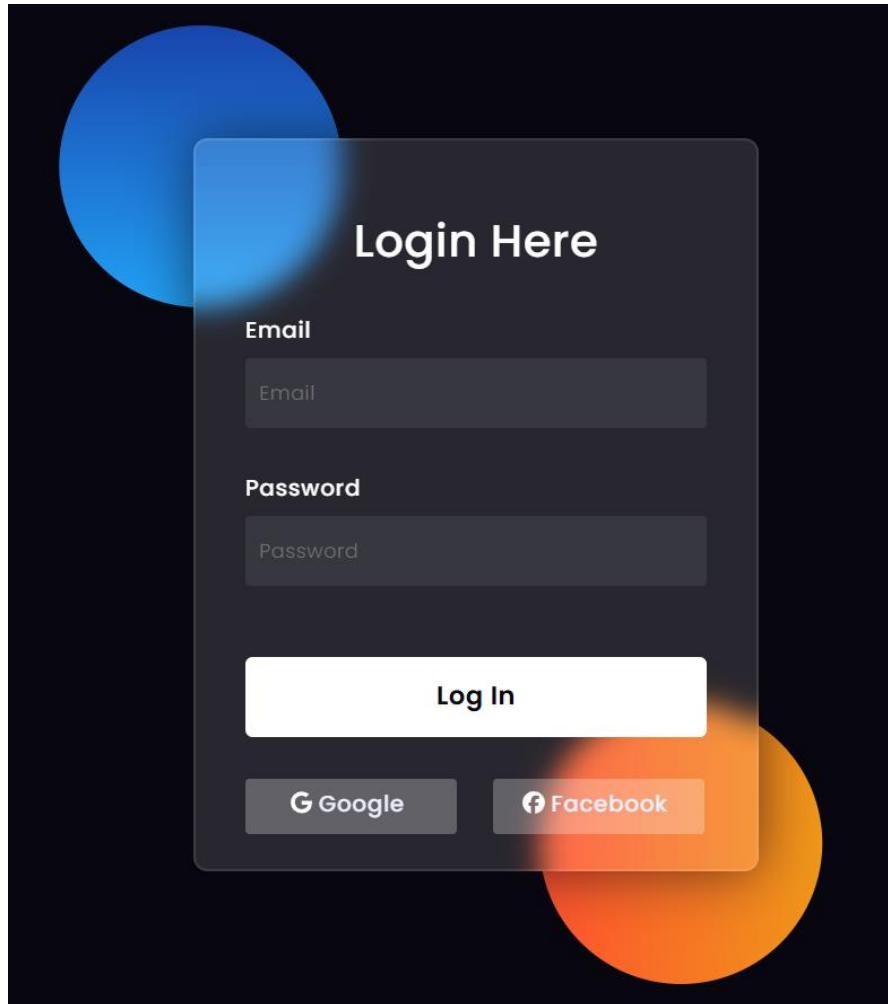


- Output

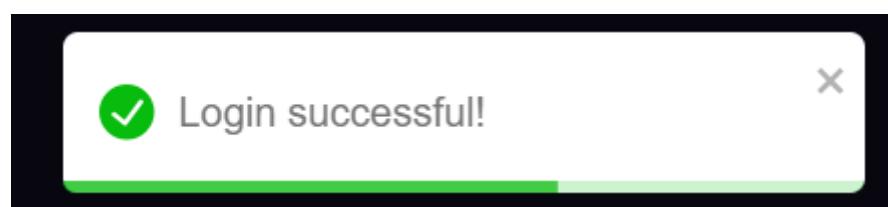


- Chức năng đăng nhập

- Input

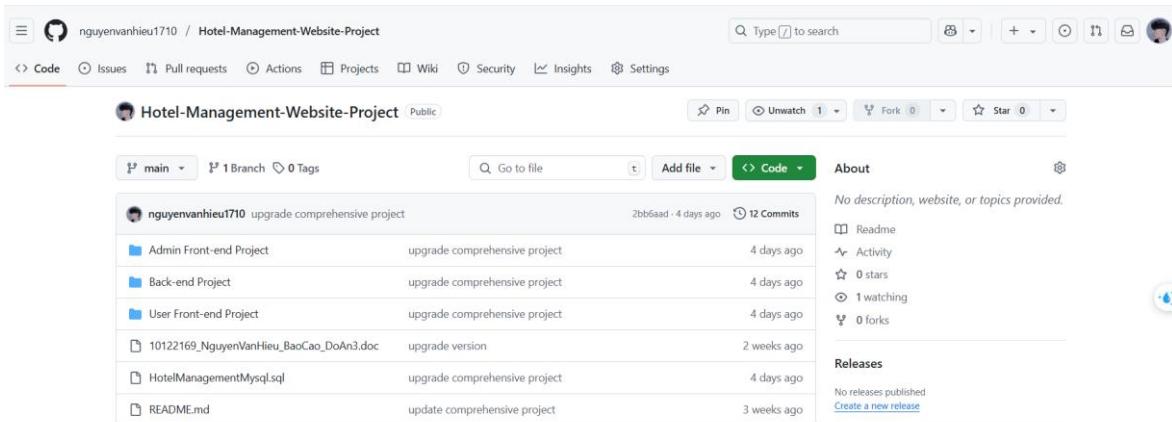


- Output



4.3.2 Đóng gói ứng dụng

Link GitHub: <https://github.com/nguyenvanhieu1710/Hotel-Management-Website-Project>



Hình 4.1 Source code GitHub

4.3.3 Triển khai ứng dụng

- Phần cứng:
 - o Một máy chủ cơ bản có thẻ sử dụng CPU với xung nhịp trung bình, RAM từ 8GB đến 16GB, và ổ cứng để lưu trữ mã nguồn và tài nguyên tĩnh (hình ảnh, CSS, JavaScript).
 - o Băng thông mạng phải đủ để xử lý số lượng người dùng dự kiến.
 - o Kết nối internet ổn định với băng thông đủ để phục vụ số lượng truy cập dự kiến.
- Phần mềm:
 - o Trình duyệt phổ biến như: Google Chrome, Edge, FireFox...
 - o Visual Studio Code 2020 trở lên

KẾT LUẬN

➤ Kết quả đạt được

- Hoàn thành được chi tiết các chức năng phân hệ quản trị nội dung.
- Giao diện thân thiện, bắt mắt, bố cục cân đối, màu sắc hài hòa.
- Xử lý được nhiều dữ liệu đầu vào, đầu ra.
- Bố trí được file tệp khoa học, dễ dàng phát hiện cũng như sửa lỗi khi hệ thống gặp vấn đề.
- Tương tác tốt giữa người dùng và hệ thống.
- Xử lý tốt các điều khiển thao tác đến dữ liệu trong cơ sở dữ liệu, đảm bảo tính logic.

➤ Hạn chế của đế tài

Bài tập lớn không tránh khỏi những hạn chế. Bài tập lớn vẫn còn một số chức năng chưa được triển khai hoặc chưa hoàn thiện đầy đủ. Điều này có thể bao gồm các tính năng mở rộng, sự tương tác phía người dùng, hoặc các phần mềm hỗ trợ. Ngoài ra, em nhận thức được những kỹ năng còn thiếu sót, đặc biệt là trong việc phân tích và thiết kế hệ thống. Điều này đã tạo ra một số thách thức trong quá trình triển khai và duy trì sản phẩm.

➤ Hướng phát triển của đế tài

Để giải quyết những hạn chế này, hướng phát triển của đế tài có thể tập trung vào nâng cấp sản phẩm bằng cách triển khai các chức năng chưa hoàn thiện, tối ưu hóa trải nghiệm người dùng, và tích hợp các công nghệ mới. Ngoài ra, việc nâng cao kỹ năng phân tích và thiết kế hệ thống thông qua việc tham gia vào các dự án thực tế hoặc theo học các khóa đào tạo chuyên sâu sẽ giúp em cải thiện khả năng quản lý và triển khai dự án website một cách hiệu quả hơn trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] Giáo trình thiết kế và lập trình Web cơ bản, Khoa Công nghệ thông tin, Trường Đại học Sư phạm Kỹ thuật Hưng Yên.
- [2] Chu Thị Minh Huệ, Nguyễn Minh Tiên, Nguyễn Minh Quý(Khoa CNTT – Đại học SPKT Hưng Yên), “Giáo trình phân tích và thiết kế hướng đối tượng với UML”, 2022.