VŨ ĐỨC DŨNG

KỸ THUẬT PHẦN MỀM

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐATN ĐẠI HỌC NGÀNH KỸ THUẬT PHẦN MỀM

XÂY DỰNG WEBSITE BOOKING HOTELS STAYBUDDY

CBHD: **TS. Nguyễn Thị Mỹ Bình**

Sinh viên: **Vũ Đức Dũng**

Mã số sinh viên: ***2021603749***

Hà Nội – Năm 2024

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐATN ĐẠI HỌC NGÀNH KỸ THUẬT PHẦN MỀM

XÂY DỰNG WEBSITE BOOKING HOTELS STAYBUDDY

CBHD:	**TS. Nguyễn Thị Mỹ Bình**
Sinh viên:	**Vũ Đức Dũng**
Mã số sinh viên:	***2021603749***

Hà Nội – Năm 2024

LỜI NÓI ĐẦU

Là một sinh viên trường Đại học Công nghiệp Hà Nội, em xin gửi lời cảm ơn chân thành đến trường và cô Nguyễn Thị Mỹ Bình. Trước hết, em xin gửi lời cảm ơn đến Trường Đại học Công nghiệp Hà Nội đã cung cấp cho em môi trường học tập và nghiên cứu tốt nhất. Những cơ sở vật chất và dịch vụ hỗ trợ tại trường đã giúp chúng em tiếp cận được những tài nguyên quan trọng và hoàn thành tốt các học phần trước đó.

Em cũng muốn gửi lời cảm ơn đến cô Nguyễn Thị Mỹ Bình – giáo viên hướng dẫn của em, cô đã cung cấp cho chúng em sự hỗ trợ và giúp đỡ trong quá trình thực hiện đồ án tốt nghiệp. Những lời khuyên, định hướng và sự hỗ trợ của cô đã giúp chúng em hoàn thành nhiệm vụ của mình một cách tốt nhất.

Học phần Đồ án tốt nghiệp là một học phần vô cùng quan trọng, đó là minh chứng cho những kiến thức đã có sau bốn năm học tập tại trường. Học phần này đã tổng hợp lại những kiến thức, kỹ năng quan trọng nhất trong năm bốn năm học tập với chuyên ngành mã em đã chọn, giúp cho em có được những nền tảng vững chắc trước khi tham gia vào môi trường lao động thực tế tại các doanh nghiệp. Một lần nữa em xin cảm TS. Nguyễn Thị Mỹ Bình đã giúp đỡ, hướng dẫn, chỉ bảo tận tình để em hoàn thành tốt Đồ án tốt nghiệp của mình. Em đã cố gắng hoàn thành Đồ án tốt nghiệp nhưng vẫn mong nhận được sự đóng góp của thầy, cô và các bạn để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

DANH MỤC CÁC TỪ VIẾT TẮT

HT CSDL	Hệ thống cơ sở dữ liệu
DOM	Document Object Model
HTTP	HyperText Transfer Protocol
SSR	Server-Side Rendering
SSG	Static Site Generation
SEO	Search Engine Optimization
API	Application Programming Interface

DANH MỤC CÁC BẢNG

Bảng 3.1. Chi tiết bảng auth_provider 40
Bảng 3.2. Chi tiết bảng user 40
Bảng 3.3. Chi tiết bảng role 41
Bảng 3.4. Chi tiết bảng category 41
Bảng 3.5. Chi tiết bảng favorite 41
Bảng 3.6. Chi tiết bảng hotel 42
Bảng 3.7. Chi tiết bảng review 42
Bảng 3.8. Chi tiết bảng review_reply 42
Bảng 3.9. Chi tiết bảng room 42
Bảng 3.10. Chi tiết bảng ticket 43
Bảng 3.11. Chi tiết bảng transactions 43
Bảng 3.12. Chi tiết bảng type_room 44
Bảng 3.13. Chi tiết bảng type_utility 44
Bảng 3.14. Chi tiết bảng utility 44
Bảng 6.1. Bảng lập kế hoạch kiểm thử 96
Bảng 6.2. Bảng yêu cầu phần cứng 96
Bảng 6.3. Bảng yêu cầu phần mềm 97
Bảng 6.4. Bảng yêu cầu về công cụ kiểm thử 97
Bảng 6.5. Bảng yêu cầu kiểm thử chức năng 98
Bảng 6.6. Bảng yêu cầu kiểm thử phi chức năng 99
Bảng 6.7. Bảng phân loại lỗi 99
Bảng 6.8. Bảng test case chức năng Đăng nhập 100
Bảng 6.9. Bảng test case chức năng Đăng ký 101
Bảng 6.10. Bảng test case chức năng Xem thông tin khách sạn 102
Bảng 6.11. Bảng test case chức năng Xem khách sạn yêu thích 102
Bảng 6.12. Bảng test case chức năng Cập nhật thông tin 102
Bảng 6.13. Bảng test case chức năng Xem đánh giá khách sạn 104
Bảng 6.14. Bảng test case chức năng Xem đơn đặt phòng 104

Bảng 6.15. Bảng test case chức năng Tìm kiếm khách sạn 105
Bảng 6.16. Bảng test case chức năng Cập nhật khách sạn yêu thích ... 105
Bảng 6.17. Bảng test case chức năng Đặt phòng 106
Bảng 6.18. Bảng test case chức năng Đổi mật khẩu 106
Bảng 6.19. Bảng test case chức năng Khôi phục mật khẩu 107
Bảng 6.20. Bảng test case chức năng Nạp tiền vào hệ thống 108
Bảng 6.21. Bảng test case chức năng Xem lịch sử nạp tiền 109
Bảng 6.22. Bảng test case chức năng Đăng ký tài khoản đối tác 109
Bảng 6.23. Bảng test case chức năng Xem thống kê đặt phòng và doanh thu 110
Bảng 6.24. Bảng test case chức năng Quản lý khách sạn 110
Bảng 6.25. Bảng test case chức năng Quản lý loại phòng 111
Bảng 6.26. Bảng test case chức năng Quản lý phòng 112
Bảng 6.27. Bảng test case chức năng Cập nhật đơn đặt phòng 113
Bảng 6.28. Bảng test case chức năng Quản lý đánh giá khách sạn 113
Bảng 6.29. Bảng báo cáo kết quả kiểm thử 115

DANH MỤC CÁC HÌNH ẢNH

Hình 2.1. Biểu đồ use case chính 4
Hình 2.2. Biểu đồ use case Đăng nhập 5
Hình 2.3. Biểu đồ use case Quản lý khách sạn 6
Hình 2.4. Biểu đồ use case Quản lý loại phòng 6
Hình 2.5. Biểu đồ use case Quản lý phòng 7
Hình 2.6. Biểu đồ use case Cập nhật thông tin 7
Hình 2.7. Biểu đồ use case Cập nhật đơn đặt phòng 8
Hình 3.1 Các bảng trong CSDL 37
Hình 3.2. Quan hệ các bảng 45
Hình 4.1. Biểu đồ trình tự use case Đăng nhập 46
Hình 4.2. Biểu đồ lớp use case Đăng nhập 47
Hình 4.3. Biểu đồ trình tự use case Xem thông tin khách sạn 47
Hình 4.4. Biểu đồ lớp use case Xem thông tin khách sạn 48
Hình 4.5. Biểu đồ trình tự use case Xem khách sạn yêu thích 49
Hình 4.6. Biểu đồ lớp use case Xem khách sạn yêu thích 49
Hình 4.7. Biểu đồ trình tự use case Cập nhật thông tin 50
Hình 4.8. Biểu đồ lớp use case Cập nhật thông tin 51
Hình 4.9. Biểu đồ trình tự use case Xem đánh giá khách sạn 52
Hình 4.10. Biểu đồ lớp use case Xem đánh giá khách sạn 52
Hình 4.11. Biểu đồ trình tự use case Xem đơn đặt phòng 52
Hình 4.12. Biểu đồ lớp use case Xem đơn đặt phòng 53
Hình 4.13. Biểu đồ trình tự Tìm kiếm khách sạn 54
Hình 4.14. Biểu đồ lớp use case Tìm kiếm khách sạn 54
Hình 4.15. Biểu đồ trình tự use case Cập nhật khách sạn yêu thích 55
Hình 4.16. Biểu đồ lớp use case Cập nhật khách sạn yêu thích 55
Hình 4.17. Biểu đồ trình tự use case Đặt phòng 56
Hình 4.18. Biểu đồ lớp use case Đặt phòng 56
Hình 4.19. Biểu đồ trình tự use case Thanh toán 57

Hình 4.20. Biểu đồ lớp use case Thanh toán 58
Hình 4.21. Biểu đồ trình tự use case Đăng ký 58
Hình 4.22. Biểu đồ lớp use case Đăng ký 59
Hình 4.23. Biểu đồ trình tự use case Đổi mật khẩu 60
Hình 4.24. Biểu đồ lớp use case Đổi mật khẩu 61
Hình 4.25. Biểu đồ trình tự use case Khôi phục mật khẩu 62
Hình 4.26. Biểu đồ lớp use case Khôi phục mật khẩu 62
Hình 4.27. Biểu đồ trình tự use case Nạp tiền vào hệ thống 63
Hình 4.28. Biểu đồ lớp use case Nạp tiền vào hệ thống 64
Hình 4.29. Biểu đồ trình tự use case Xem lịch sử nạp tiền 64
Hình 4.30. Biểu đồ lớp use case Xem lịch sử nạp tiền 65
Hình 4.31. Biểu đồ trình tự use case Đăng ký tài khoản đối tác 66
Hình 4.32. Biểu đồ lớp use case Đăng ký tài khoản đối tác 66
Hình 4.33. Biểu đồ trình tự use case Xem thống kê đặt phòng và doanh thu 67
Hình 4.34. Biểu đồ lớp use case Xem thống kê đặt phòng và doanh thu67
Hình 4.35. Biểu đồ trình tự Quản lý khách sạn 69
Hình 4.36. Biểu đồ lớp use case Quản lý khách sạn 69
Hình 4.35. Biểu đồ trình tự Quản lý loại phòng 71
Hình 4.36. Biểu đồ lớp use case Quản lý loại phòng 71
Hình 4.35. Biểu đồ trình tự Quản lý phòng 73
Hình 4.36. Biểu đồ lớp use case Quản lý phòng 73
Hình 4.35. Biểu đồ trình tự Cập nhật đơn đặt phòng 74
Hình 4.36. Biểu đồ lớp use case Cập nhật đơn đặt phòng 75
Hình 4.35. Biểu đồ trình tự Cập nhật mã giảm giá 77
Hình 4.36. Biểu đồ lớp use case Quản lý đánh giá khách sạn 77
Hình 5.1. Sơ đồ điều hướng giữa các màn hình 85
Hình 5.2. Màn hình trang đăng nhập 86
Hình 5.3. Màn hình đăng ký 87

Hình 5.4. Màn hình trang xem thông tin khách sạn..... 89
Hình 5.5. Màn hình trang xem khách sạn yêu thích 89
Hình 5.6. Màn hình trang cập nhật thông tin 90
Hình 5.7. Màn hình xem đơn đặt phòng 90
Hình 5.8. Màn hình trang đổi mật khẩu 91
Hình 5.9. Màn hình trang khôi phục mật khẩu 91
Hình 5.10. Màn hình trang nạp tiền vào hệ thống 92
Hình 5.11. Màn hình trang quản lý xem lịch sử nạp tiền 92
Hình 5.12. Màn hình trang đăng ký tài khoản đối tác 93
Hình 5.13. Màn hình trang xem thống kê đặt phòng và doanh thu 93
Hình 5.14. Màn hình trang quản lý khách sạn 94
Hình 5.15. Màn hình trang quản lý loại phòng..... 94
Hình 5.16. Màn hình trang quản lý phòng..... 95
Hình 5.17. Màn hình trang cập nhật đơn đặt phòng 95
Hình 5.18. Màn hình trang quản lý đánh giá khách sạn 95

MỤC LỤC

LỜI NÓI ĐẦU v
DANH MỤC CÁC TỪ VIẾT TẮT vi
DANH MỤC CÁC BẢNG vii
DANH MỤC CÁC HÌNH ẢNH ix
CHƯƠNG 1. MÔ TẢ CÁC YÊU CẦU CỦA HỆ THỐNG 1
1.1 Khảo sát hệ thống 1
1.1.1 Tổng quan về hệ thống 1
1.2 Các hoạt động của hệ thống 2
1.2.1 Hoạt động đặt khách sạn 2
1.2.2 Cập nhật thông tin hệ thống 2
1.3 Các yêu cầu của hệ thống 2
1.3.1 Yêu cầu chức năng 2
1.3.2 Yêu cầu phi chức năng 3
CHƯƠNG 2. PHÂN TÍCH CÁC YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG 4
2.1 Biểu đồ use case 4
2.1.1 Các use case chính 4
2.1.2 Quan hệ giữa các use case 4
2.2 Mô tả chi tiết các use case 9
2.2.1 Đăng nhập 9
2.2.2 Xem thông tin khách sạn 10
2.2.3 Xem khách sạn yêu thích 11
2.2.4 Cập nhật thông tin 12
2.2.5 Xem đánh giá khách sạn 13
2.2.6 Xem đơn đặt phòng 14
2.2.7 Tìm kiếm khách sạn 15
2.2.8 Cập nhật khách sạn yêu thích 16
2.2.9 Đặt phòng 17

2.2.10 Thanh toán 18
2.2.11 Đăng ký 19
2.2.12 Đổi mật khẩu 20
2.2.13 Khôi phục mật khẩu 22
2.2.14 Nạp tiền vào hệ thống 23
2.2.15 Xem lịch sử nạp tiền 24
2.2.16 Đăng ký tài khoản đối tác 25
2.2.17 Xem thống kê đặt phòng và doanh thu 26
2.2.18 Quản lý khách sạn 27
2.2.19 Quản lý loại phòng 29
2.2.20 Quản lý phòng 31
2.2.21 Cập nhật đơn đặt phòng 32
2.2.22 Quản lý đánh giá khách sạn 34
CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU 37
3.1 Mô hình hóa dữ liệu 37
3.2 Thiết kế bảng 40
3.3 Quan hệ các bảng 45
CHƯƠNG 4. THIẾT KẾ CÁC CHỨC NĂNG CỦA HỆ THỐNG 46
4.1 Use case Đăng nhập 46
4.1.1 Biểu đồ trình tự: 46
4.1.2 Biểu đồ lớp phân tích: 47
4.2 Use case Xem thông tin khách sạn 47
4.2.1 Biểu đồ trình tự 47
4.2.2 Biểu đồ lớp 48
4.3 Use case Xem khách sạn yêu thích 48
4.3.1 Biểu đồ trình tự 48
4.3.2 Biểu đồ lớp 49
4.4 Use case Cập nhật thông tin 50
4.4.1 Biểu đồ trình tự 50

4.4.2 Biểu đồ lớp 51
4.5 Use case Xem đánh giá khách sạn 51
4.5.1 Biểu đồ trình tự 51
4.5.2 Biểu đồ lớp 52
4.6 Use case Xem đơn đặt phòng 52
4.6.1 Biểu đồ trình tự 52
4.6.2 Biểu đồ lớp 53
4.7 Use case Tìm kiếm khách sạn 54
4.7.1 Biểu đồ trình tự 54
4.7.2 Biểu đồ lớp 54
4.8 Use case Cập nhật khách sạn yêu thích 55
4.8.1 Biểu đồ trình tự 55
4.8.2 Biểu đồ lớp 55
4.9 Use case Đặt phòng 56
4.9.1 Biểu đồ trình tự 56
4.9.2 Biểu đồ lớp 56
4.10 Use case Thanh toán 57
4.10.1 Biểu đồ trình tự 57
4.10.2 Biểu đồ lớp 58
4.11 Use case Đăng ký 58
4.11.1 Biểu đồ trình tự 58
4.11.2 Biểu đồ lớp 59
4.12 Use case Đổi mật khẩu 60
4.12.1 Biểu đồ trình tự 60
4.12.2 Biểu đồ lớp 61
4.13 Use case Khôi phục mật khẩu 61
4.13.1 Biểu đồ trình tự 61
4.13.2 Biểu đồ lớp 62
4.14 Use case Nạp tiền vào hệ thống 63

4.14.1 Biểu đồ trình tự 63
4.14.2 Biểu đồ lớp 64
4.15 Use case Xem lịch sử nạp tiền 64
4.15.1 Biểu đồ trình tự 64
4.15.2 Biểu đồ lớp 65
4.16 Use case Đăng ký tài khoản đối tác 65
4.16.1 Biểu đồ trình tự 65
4.16.2 Biểu đồ lớp 66
4.17 Use case Xem thống kê đặt phòng và doanh thu 67
4.17.1 Biểu đồ trình tự 67
4.17.2 Biểu đồ lớp 67
4.18 Use case Quản lý khách sạn 68
4.18.1 Biểu đồ trình tự 68
4.18.2 Biểu đồ lớp 69
4.19 Use case Quản lý loại phòng 70
4.19.1 Biểu đồ trình tự 70
4.19.2 Biểu đồ lớp 71
4.20 Use case Quản lý phòng 72
4.20.1 Biểu đồ trình tự 72
4.20.2 Biểu đồ lớp 73
4.21 Use case Cập nhật đơn đặt phòng 74
4.21.1 Biểu đồ trình tự 74
4.21.2 Biểu đồ lớp 75
4.22 Use case Quản lý đánh giá khách sạn 76
4.22.1 Biểu đồ trình tự 76
4.22.2 Biểu đồ lớp 77
CHƯƠNG 5. CÀI ĐẶT VÀ KẾT QUẢ 78
5.1 Giới thiệu ngôn ngữ và công nghệ sử dụng 78
5.1.1 JavaScript 78

5.1.2 TypeScript 78
5.1.3 NestJS 80
5.1.4 Next.js 81
5.1.5 PostgreSql 82
5.1.6 RabbitMQ 83
5.1.7 Hệ thống Middleware Logging và Monitoring 83
5.2 Sơ đồ điều hướng giữa các màn hình 85
5.3 Kết quả chương trình 86
5.3.1 Màn hình trang đăng nhập 86
5.3.2 Màn hình đăng ký 87
5.3.3 Màn hình trang xem thông tin khách sạn 88
5.3.4 Màn hình trang xem khách sạn yêu thích 89
5.3.5 Màn hình trang cập nhập thông tin 90
5.3.6 Màn hình trang xem đơn đặt phòng 90
5.3.7 Màn hình trang đổi mật khẩu 91
5.3.8 Màn hình trang khôi phục mật khẩu 91
5.3.9 Màn hình trang nạp tiền vào hệ thống 92
5.3.10 Màn hình trang xem lịch sử nạp tiền 92
5.3.11 Màn hình trang đăng ký tài khoản đối tác 93
5.3.12 Màn hình trang xem thống kê đặt phòng và doanh thu 93
5.3.13 Màn hình trang quản lý khách sạn 94
5.3.14 Màn hình trang quản lý loại phòng 94
5.3.15 Màn hình trang quản lý phòng 94
5.3.16 Màn hình trang cập nhật đơn đặt phòng 95
5.3.17 Màn hình trang quản lý đánh giá khách sạn 95
CHƯƠNG 6. KIỂM THỬ HỆ THỐNG 96
6.1 Kế hoạch kiểm thử 96
6.1.1 Lịch trình công việc 96
6.1.2 Những yêu cầu về tài nguyên 96

6.1.3 Phạm vi kiểm thử 97
6.1.4 Chiến lược kiểm thử 98
6.1.5 Quản lý lỗi 99
6.2 Test case 100
6.2.1 Thiết kế test case 100
6.2.2 Báo cáo kiểm thử 115
CHƯƠNG 7. TỔNG KẾT VÀ ĐÁNH GIÁ 116
7.1 Đánh giá kết quả 116
7.2 Hướng phát triển 116
TÀI LIỆU THAM KHẢO 117

CHƯƠNG 1. MÔ TẢ CÁC YÊU CẦU CỦA HỆ THỐNG

1.1 Khảo sát hệ thống

1.1.1 Tổng quan về hệ thống

Một website booking hotels "StayBuddy" là một nền tảng đặt phòng khách sạn trực tuyến, cung cấp cho người dùng một trải nghiệm đặt phòng linh hoạt và thuận tiện. Hệ thống này phục vụ nhiều đối tượng khác nhau, bao gồm khách du lịch, đối tác chủ khách sạn.

Với giao diện thân thiện và dễ sử dụng, khách hàng có thể dễ dàng tìm kiếm và đặt phòng tại các khách sạn đa dạng. Họ có thể lọc tìm kiếm theo tiêu chí như địa điểm, ngày check-in/out, giá cả, tiện ích và xếp hạng sao. Sau khi chọn khách sạn, khách hàng có thể xem thông tin chi tiết về phòng ốc, giá cả và đánh giá từ khách trước, hỗ trợ họ đưa ra quyết định đặt phòng phù hợp.

Điểm nổi bật của website là hỗ trợ đa ngôn ngữ giúp tiếp cận nhiều khách hàng quốc tế. Bên cạnh đó, hệ thống còn cung cấp nhiều tính năng hỗ trợ khách hàng như xem thông tin khách sạn, xem khách sạn yêu thích, cập nhật thông tin, xem đánh giá khách sạn, xem đơn đặt phòng, cập nhật khách sạn yêu thích, đặt phòng, thanh toán, đổi mật khẩu, khôi phục mật khẩu, nạp tiền vào hệ thống, xem lịch sử nạp tiền, đăng ký tài khoản đối tác.

Ngoài ra, website booking hotels "StayBuddy" cũng cung cấp các tính năng quản lý cho đối tác chủ khách sạn. Các đối tác có thể xem thông kê đặt phòng và doanh thu, quản lý thông tin khách sạn, loại phòng, phòng, đặt phòng, đánh giá.

Với các tính năng nổi bật và mô hình kinh doanh hiệu quả, website booking hotels "StayBuddy" hướng tới mục tiêu trở thành nền tảng đặt phòng khách sạn hàng đầu, mang lại trải nghiệm tuyệt vời cho cả khách hàng và đối tác.

1.2 Các hoạt động của hệ thống

1.2.1 Hoạt động đặt khách sạn

Hoạt động nghiệp vụ

- Khi người dùng chọn 1 khách sạn, hệ thống sẽ hiện thị thông tin chi tiết của khách sạn.
- Người dùng lựa chọn ngày check-in/check-out và số lượng khách, hệ thống sẽ hiện thị các phòng khác sạn có sẵn trong thời gian đó.
- Khi người dùng chọn phòng và tiến hành đặt phòng khi người dùng kích nút **Đặt ngay**, hệ thống sẽ tạo đơn đặt phòng và yêu cầu thanh toán, người dùng nhập thông tin thanh toán. Sau đó hệ thống sẽ tạo mới đơn đặt phòng cho người dùng, người dùng có thể theo dõi trạng thái đơn hàng tại mục **Đặt chỗ**.

1.2.2 Cập nhật thông tin hệ thống

Người quản lí có thể thêm, sửa, xóa thông tin:

- ❖ Danh mục khách sạn.
- ❖ Người dùng.

1.3 Các yêu cầu của hệ thống

1.3.1 Yêu cầu chức năng

Các chức năng đối tác:

- ✓ Đăng nhập.
- ✓ Xem thống kê đặt phòng và doanh thu.
- ✓ Quản lý khách sạn.
- ✓ Quản lý loại phòng.
- ✓ Quản lý phòng.
- ✓ Cập nhật đơn đặt phòng.
- ✓ Quản lý đánh giá khách sạn.

Các chức năng phía người dùng:

- ✓ Đăng nhập, đăng ký, đăng ký tài khoản đối tác.

- ✓ Khôi phục mật khẩu.
- ✓ Xem thông tin khách sạn.
- ✓ Xem đánh giá khách sạn.
- ✓ Tìm kiếm khách sạn.
- ✓ Đặt phòng.
- ✓ Thanh toán.
- ✓ Nạp tiền vào hệ thống
- ✓ Xem lịch sử nạp tiền.
- ✓ Xem đơn đặt phòng.
- ✓ Cập nhật thông tin.
- ✓ Đổi mật khẩu.
- ✓ Xem khách sạn yêu thích
- ✓ Cập nhật khách sạn yêu thích.

1.3.2 Yêu cầu phi chức năng

Giao diện:

- ✓ Giao diện hỗ trợ tiếng Việt và tiếng Anh.
- ✓ Cung cấp giao diện đơn giản, gần gũi, trực quan về dễ sử dụng đối với người dùng.

Bảo mật:

- ✓ Thực hiện bảo mật bằng các cách mã hóa các thông tin nhạy cảm của người dùng.
- ✓ Phân quyền truy cập, chỉ cho phép người dùng truy cập những chức năng được cho phép.
- ✓ Kiểm định những dữ liệu người dùng nhập vào.

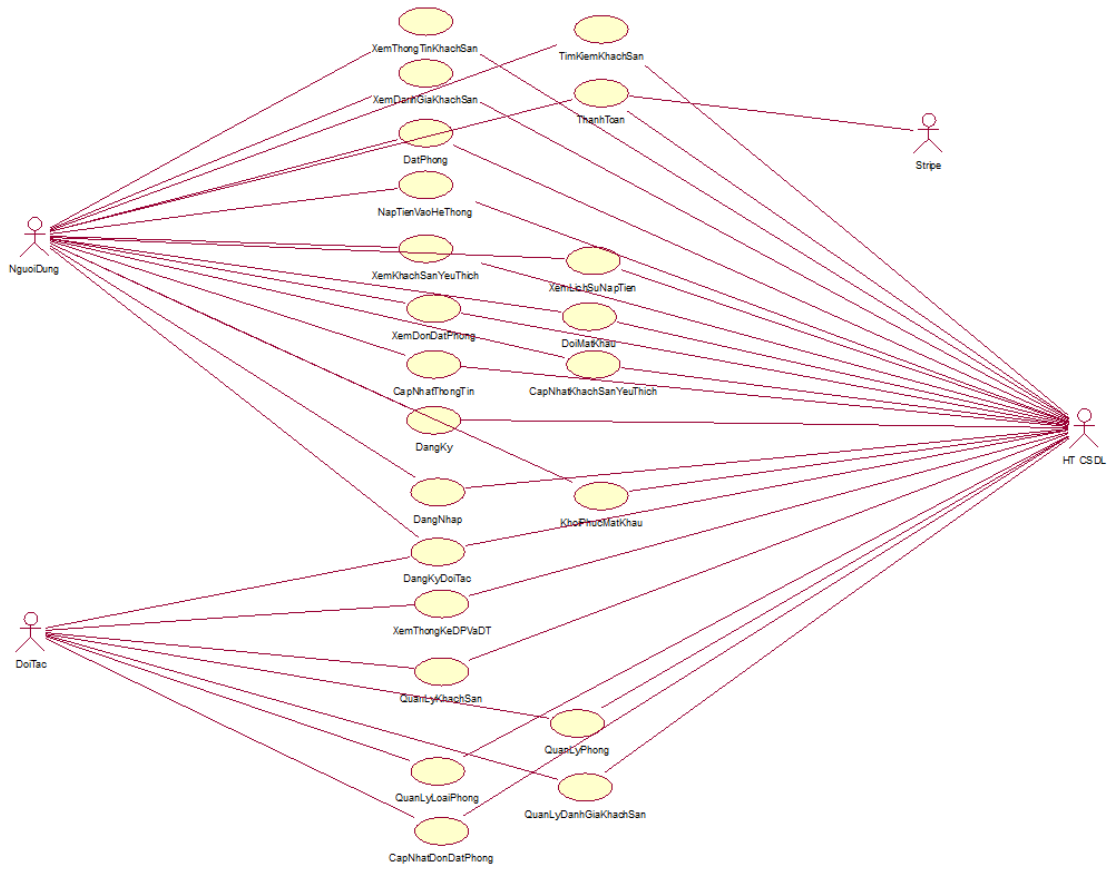
Tốc độ xử lý & thời gian hoạt động:

- ✓ Tốc độ xử lý nhanh chóng, thực hiện loading khi lấy hoặc ghi dữ liệu.
- ✓ Có thể hoạt động hiệu quả 24/24.

CHƯƠNG 2. PHÂN TÍCH CÁC YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG

2.1 Biểu đồ use case

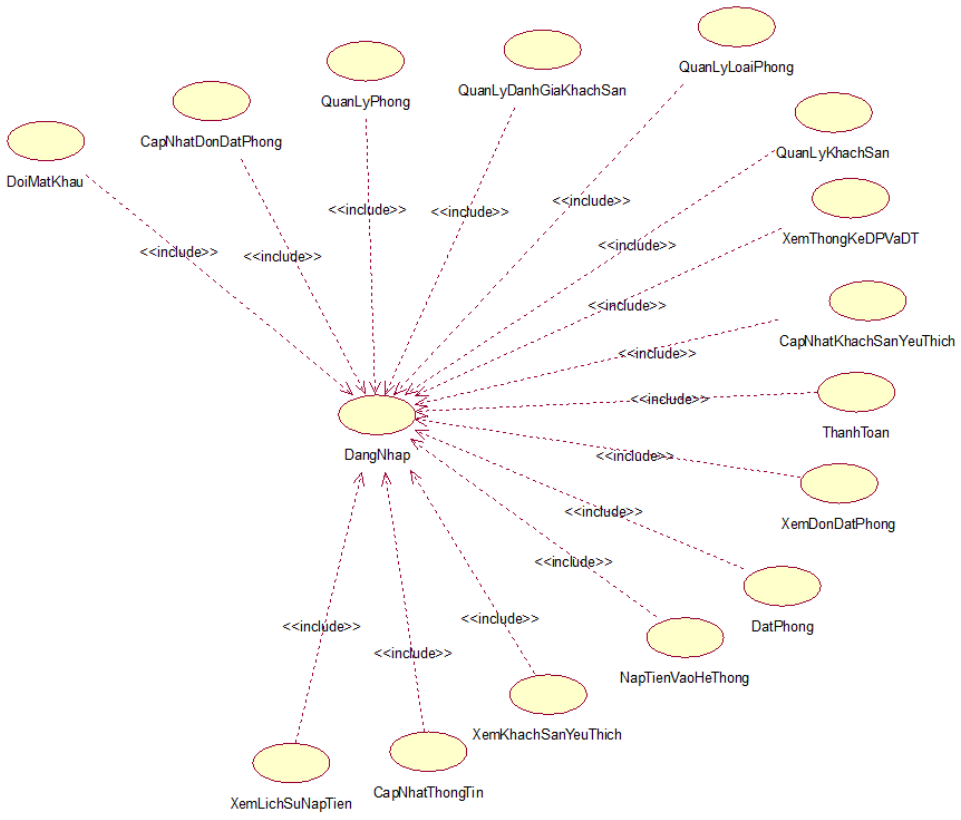
2.1.1 Các use case chính



Hình 2.1. Biểu đồ use case chính

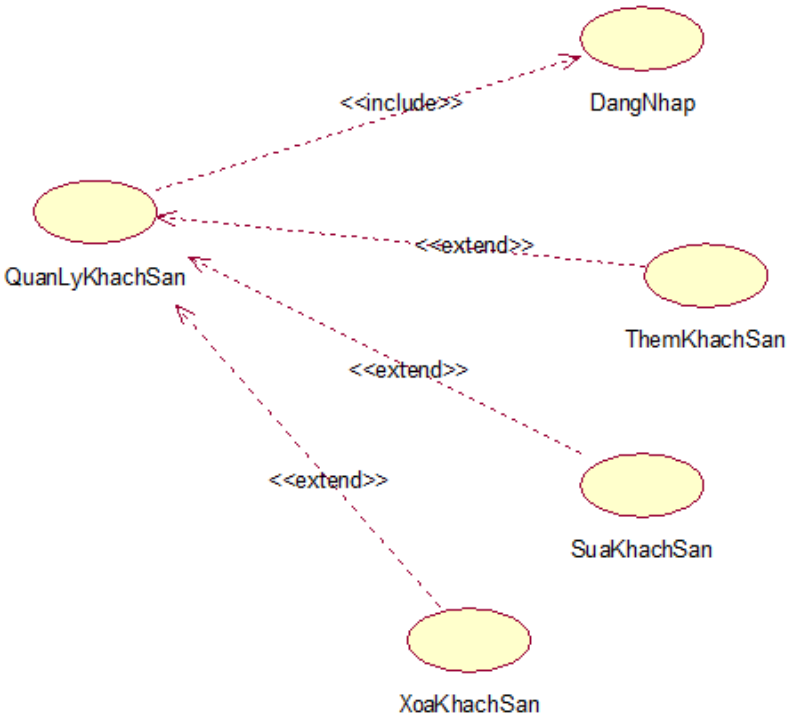
2.1.2 Quan hệ giữa các use case

Đăng nhập



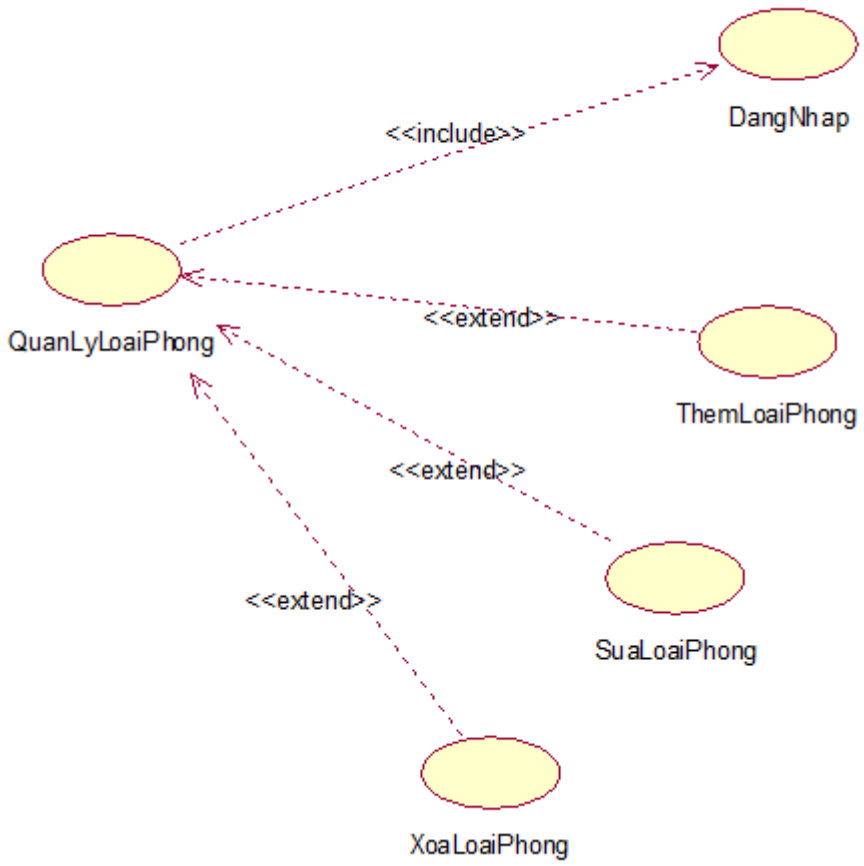
Hình 2.2. Biểu đồ use case Đăng nhập

Quản lý khách sạn



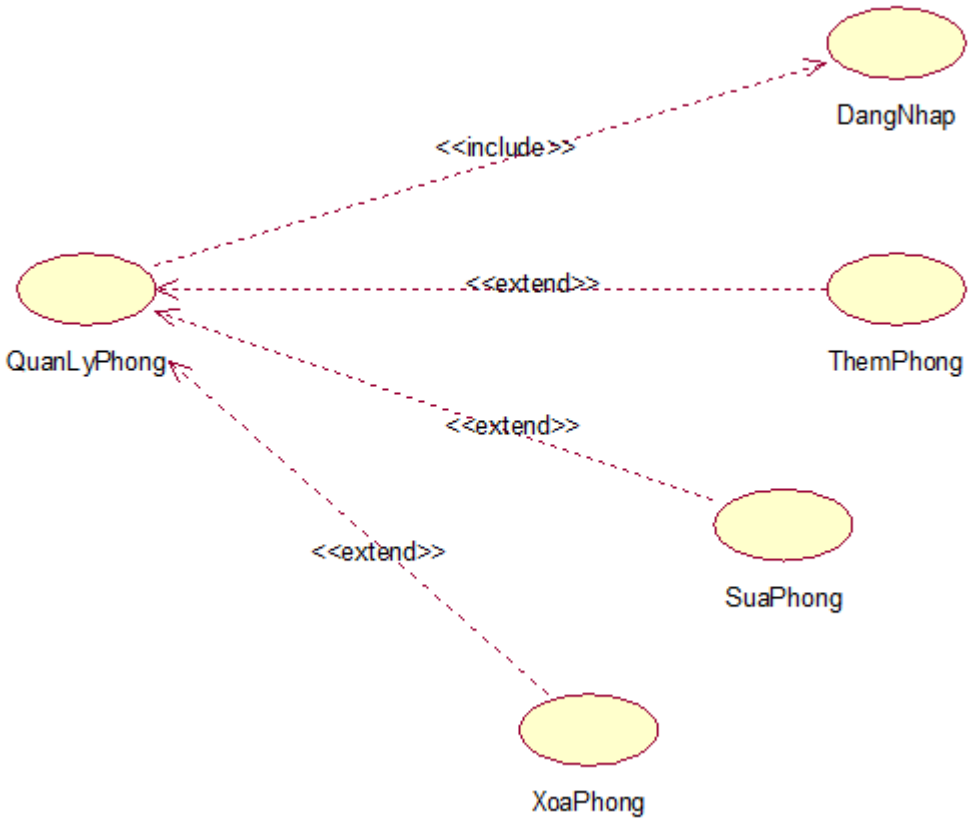
Hình 2.3. Biểu đồ use case Quản lý khách sạn

Quản lý loại phòng



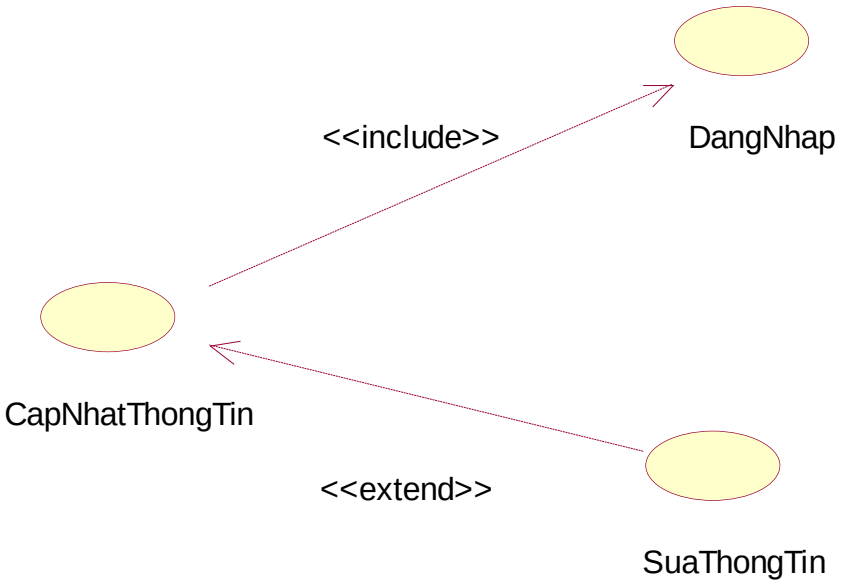
Hình 2.4. Biểu đồ use case Quản lý loại phòng

Quản lý phòng



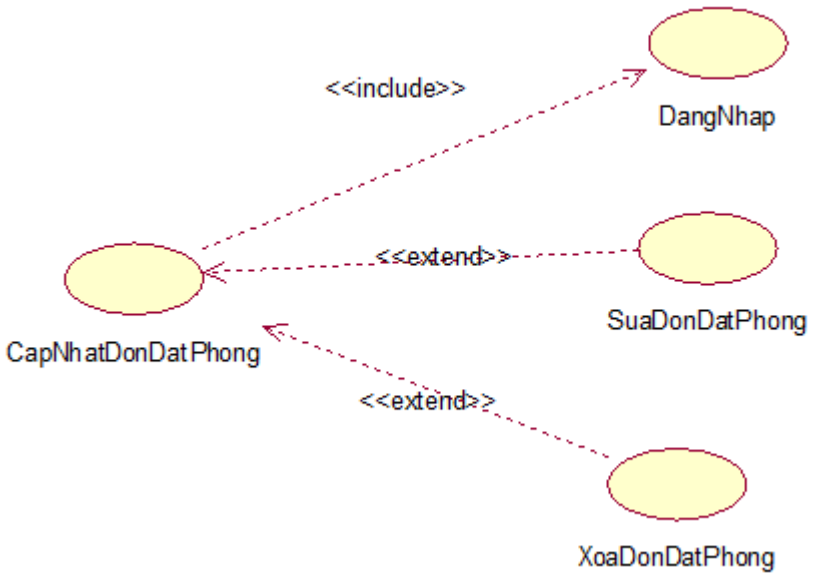
Hình 2.5. Biểu đồ use case Quản lý phòng

Cập nhật thông tin



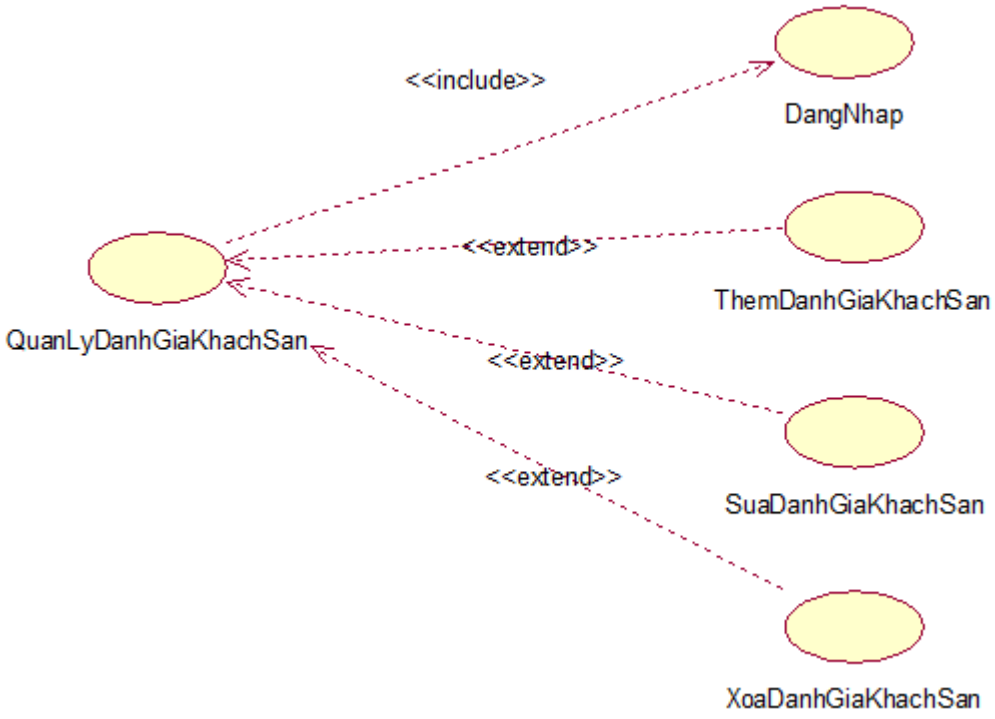
Hình 2.6. Biểu đồ use case Cập nhật thông tin

Cập nhật đơn đặt phòng



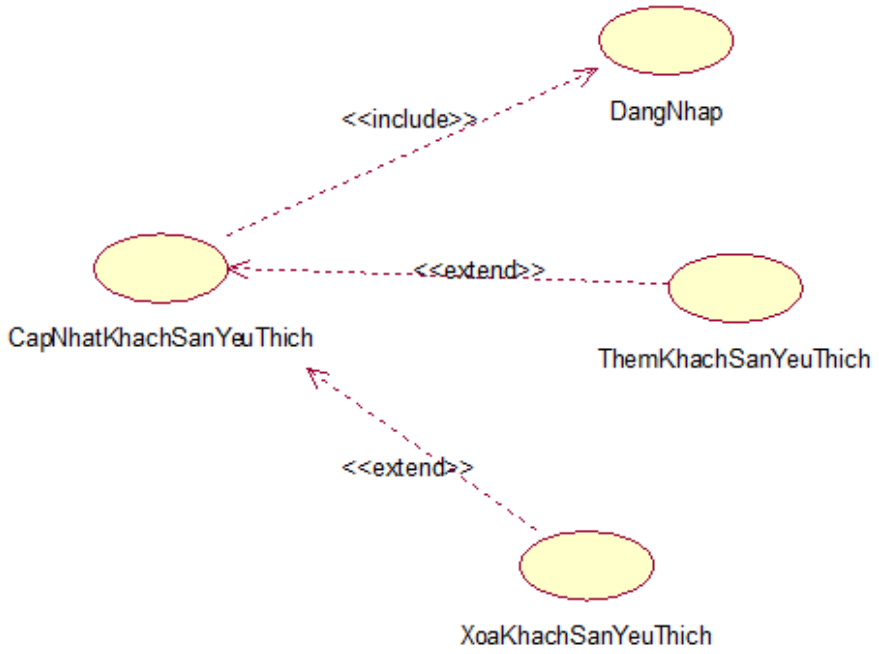
Hình 2.7. Biểu đồ use case Cập nhật đơn đặt phòng

Quản lý đánh giá khách sạn

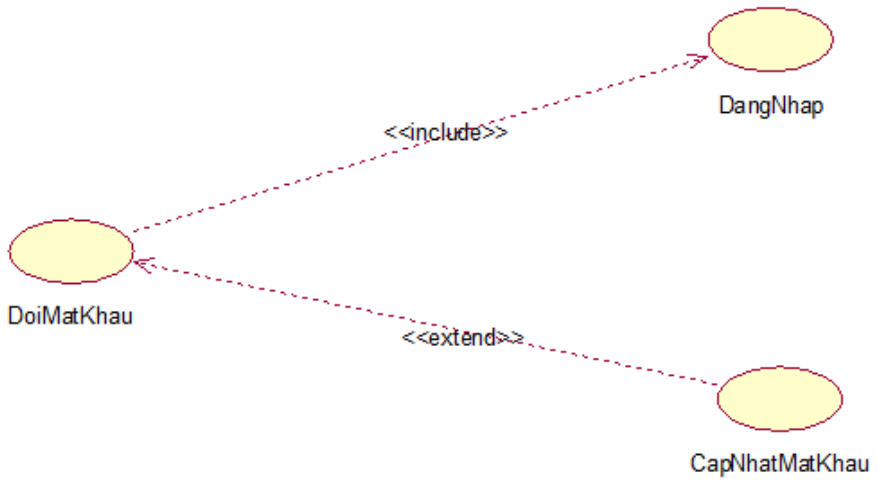


Hình 2.8. Biểu đồ use case Quản lý đánh giá khách sạn

Cập nhật khách sạn yêu thích



Hình 2.9. Biểu đồ use case Cập nhật khách sạn yêu thích



Hình 2.10 Biểu đồ use case Đổi mật khẩu

2.2 Mô tả chi tiết các use case

2.2.1 Đăng nhập

Mô tả vắn tắt:

Use case này cho phép người dùng (người dùng, đối tác & người quản trị) đăng nhập vào hệ thống.

Luồng sự kiện:

- **Luồng cơ bản:**
    1. Use case này bắt đầu khi người dùng muốn đăng nhập vào hệ thống. Hệ thống sẽ hiển thị màn hình yêu cầu nhập tên đăng nhập và mật khẩu.
    2. Người dùng nhập tên và mật khẩu sau đó click vào nút **Đăng nhập.** Hệ thống kiểm tra tên đăng nhập, mật khẩu và quyền truy nhập trong bảng **user** và hiển thị màn hình trang chủ hệ thống. Use case kết thúc.

- **Luồng rẽ nhánh:**
    1. Tại bước 2 trong luồng cơ bản nếu người dùng nhập sai tên hoặc/và mật khẩu thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng nhập lại để tiếp tục và use case kết thúc.
    2. Tại bước 2 trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Không có.

Hậu điều kiện:

Không có.

Điểm mở rộng:

Không có.

2.2.2 Xem thông tin khách sạn

Mô tả vắn tắt:

Use case này cho phép người dùng xem thông tin chi tiết của khách sạn.

Luồng sự kiện:

- **Luồng cơ bản:**
    1. Use case này bắt đầu khi người dùng click vào một khách sạn trên trang chủ. Hệ thống sẽ chuyển hướng sang màn hình thông tin chi tiết của khách sạn đó bao gồm tên khách sạn, hình ảnh, loại phòng, phòng, đánh giá, địa chỉ, số điện thoại liên hệ, mô tả. Use case kết thúc.

- **Luồng rẽ nhánh:**
    1. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Không có.

Hậu điều kiện:

Không có.

Điểm mở rộng:

Không có.

2.2.3 Xem khách sạn yêu thích

Mô tả vắn tắt:

Use case này cho phép người dùng xem những khách sạn mà người dùng yêu thích.

Luồng sự kiện:

- **Luồng cơ bản:**
    1. Use case này bắt đầu khi người dùng click vào **Biểu tượng trái tim** trên thanh tiêu đề của hệ thống. Hệ thống sẽ hiển thị các khách sạn yêu thích của người dùng đó bao gồm hình ảnh, tên khách sạn, địa chỉ, đánh giá. Use case kết thúc

- **Luồng rẽ nhánh:**
    1. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập tài khoản.

Hậu điều kiện:

Không có.

Điểm mở rộng:

Không có.

2.2.4 Cập nhật thông tin

Mô tả vắn tắt:

Use case này cho phép người dùng xem cập nhật thông tin cá nhân.

Luồng sự kiện:

- **Luồng cơ bản:**
    1. Use case này bắt đầu khi người dùng click vào nút **Trang cá nhân** trên thanh tiêu đề của hệ thống. Hệ thống sẽ chuyển sang màn hình thông tin tài khoản và hiển thị lên những thông tin gồm ảnh đại diện, họ tên, email, số điện thoại, ngày sinh tại bảng user.
    2. Khi người dùng nhập mới các thông tin tài khoản và click vào nút **Lưu thông tin**, hệ thống sẽ cập nhật thông tin thay đổi của người dùng lên cơ sở dữ liệu. Và use case kết thúc.

- **Luồng rẽ nhánh:**
    1. Tại bước 2 luồng cơ bản, nếu người dùng nhập email và số điện thoại trùng với 1 người dùng khác trên hệ thống thì hệ thống sẽ báo lỗi và yêu cầu nhập lại.

2. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị **thông báo lỗi** và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập.

Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin người dùng sẽ được cập nhật vào bảng **user**.

Điểm mở rộng:

Không có.

2.2.5 Xem đánh giá khách sạn

Mô tả vắn tắt:

Use case này cho phép người dùng xem các đánh giá về khách sạn từ những người dùng đã đặt và sử dụng dịch vụ tại khách sạn đó. Người dùng có thể sắp xếp đánh giá theo nhiều tiêu chí khác nhau, bao gồm thời gian và trạng thái có ảnh/không có ảnh.

Luồng sự kiện:

- **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng click vào tab **Đánh giá** trong trang chi tiết khách sạn. Hệ thống sẽ hiện thị các đánh giá bao gồm tên người dùng, ảnh đại diện, điểm đánh giá trung bình, nội dung đánh giá, ngày đánh giá, ảnh kèm theo (nếu có). Use case kết thúc.

- **Luồng rẽ nhánh:**

1. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải truy cập vào trang chi tiết của một khách sạn

Hậu điều kiện:

Không có.

Điểm mở rộng:

Không có.

2.2.6 Xem đơn đặt phòng

Mô tả vắn tắt:

Use case này cho phép người dùng xem danh sách đơn đặt phòng, bao gồm thông tin cơ bản về phòng và trạng thái thanh toán.

Luồng sự kiện:

- **Luồng cơ bản:**
  1. Use case này bắt đầu khi người dùng click vào nút **Đơn đặt phòng** tại menu tài khoản của người dùng phần trên cùng của màn hình hệ thống, hệ thống sẽ lấy ra thông tin của các phòng đã đặt gồm tên phòng, ảnh phòng, mô tả đánh giá, địa chỉ, trạng thái thanh toán tại bảng **ticket**.
  2. Khi người dùng click vào một đơn đặt phòng, hệ thống chuyển sang màn hình thông tin chi tiết của khách sạn chứa phòng đã đặt đó và use case kết thúc.

- **Luồng rẽ nhánh:**
  1. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập.

Hậu điều kiện:

Không có.

Điểm mở rộng:

Không có.

2.2.7 Tìm kiếm khách sạn

Mô tả vắn tắt:

Use case này cho phép người dùng tìm kiếm khách sạn dựa trên các thông tin như địa điểm, ngày đặt, số lượng phòng và sức chứa.

Luồng sự kiện:

- **Luồng cơ bản:**
  1. Use case này bắt đầu khi người dùng nhập đầy đủ thông tin gồm địa điểm, ngày nhận và trả phòng và số lượng phòng, sức chưa và click vào nút **Tìm kiếm**. Hệ thống sẽ kiểm tra các thông tin đầu vào. Nếu thông tin hợp lệ, hệ thống thực hiện tìm kiếm trong bảng **hotel** và hiển thị danh sách khách sạn phù hợp gồm tên khách sạn, địa chỉ, hình ảnh, đánh giá.
  2. Khi người dùng chọn một khách sạn từ danh sách. Hệ thống sẽ chuyển sang màn hình thông tin chi tiết khách sạn tương ứng. Và use case kết thúc

- **Luồng rẽ nhánh:**
  1. Tại bước 1 luồng cơ bản, nếu người dùng không nhập đầy đủ thông tin hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập đầy đủ thông tin.
  2. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Không có.

Hậu điều kiện:

Nếu use case kết thúc thành công, danh sách khách sạn phù hợp được hiển thị cho người dùng

Điểm mở rộng:

Không có.

2.2.8 Cập nhật khách sạn yêu thích

Mô tả vắn tắt:

Use case này cho phép người dùng thêm hoặc xóa khách sạn khỏi danh sách yêu thích.

Luồng sự kiện:

- **Luồng cơ bản:**
  1. Use case này bắt đầu khi người dùng click vào **Biểu tượng trái tim** trên thẻ khách sạn.
  2. Hệ thống sẽ kiểm tra trạng thái yêu thích của khách sạn. Nếu khách sạn chưa được yêu thích, hệ thống sẽ thêm thông tin khách sạn vào bảng **favorite** và hiển thị thông báo thêm thành công. Nếu khách sạn đã được yêu thích, hệ thống sẽ xóa thông tin khách sạn khỏi bảng **favorite** và hiện thị thông báo xóa thành công. Và use case kết thúc

- **Luồng rẽ nhánh:**
  1. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập.

Hậu điều kiện:

Trạng thái yêu thích của khác sạn được cập nhập thành công trong hệ thống (thêm vào hoặc xóa khỏi danh sách yêu thích).

Điểm mở rộng:

Không có.

2.2.9 Đặt phòng

Mô tả vắn tắt:

Use case này cho phép người dùng đặt phòng thông qua giao diện chi tiết khách sạn với hai phương thức thanh toán: thanh toán qua tài khoản hệ thống hoặc thanh toán trực tiếp.

Luồng sự kiện:

- **Luồng cơ bản:**
  1. Use case này bắt đầu khi người dùng click vào nút **Đặt phòng** tại giao diện chi tiết của một khác sạn. Hệ thống sẽ chuyển sang màn hình nhập thông tin cho đơn đặt phòng gồm tên người đặt, email, số điện thoại, các lựa chọn bổ sung (nếu có), thời gian nhận phòng, thời gian trả phòng và click vào nút **Tiếp tục thanh toán**.
  2. Hệ thống sẽ tạo mới đơn đặt phòng với trạng thái chờ thanh toán trong bảng **ticket** và chuyển sang màn hình lựa chọn phương thức thanh toán gồm thanh toán qua tài khoản hệ thống hoặc thanh toán trực tiếp. Và use case kết thúc.

- **Luồng rẽ nhánh:**
  1. Tại bước 2 của luồng cơ bản, nếu người dùng không thể thực hiện thành công với phương thức thanh toán đã chọn hệ thống sẽ báo lỗi và yêu cầu người dùng chọn lại.
  2. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập.

Hậu điều kiện:

Nếu use case kết thúc thành công hệ thống sẽ tạo mới một đơn đặt phòng trong bảng **ticket**.

Điểm mở rộng:

Không có.

2.2.10 Thanh toán

Mô tả vắn tắt:

Use case này cho phép người dùng thanh toán đơn đặt phòng trên hệ thống thông qua hai phương thức: thanh toán bằng tài khoản trên hệ thống hoặc thanh toán trực tiếp.

Luồng sự kiện:

- **Luồng cơ bản:**
    1. Use case này bắt đầu khi khách hàng chọn phương thức thanh toán là thanh toán bằng tài khoản hệ thống hoặc thanh toán trực tiếp với Stripe và click vào nút **Thanh toán** trong trang lựa chọn phương thức thanh toán.
    2. Nếu người dùng chọn thanh toán bằng tài khoản hệ thống, hệ thống sẽ hiện thị số dư khả dụng hiện có trên hệ thống, nếu số dư khả dụng đủ để thực hiện giao dịch thanh toán. Hệ thống sẽ hiện thị thống báo thành công.
    3. Nếu người dùng chọn thanh toán trực tiếp với Stripe. Hệ thống sẽ hiển thị một màn hình yêu cầu khách hàng cần nhập thông tin tài khoản thanh toán của mình. Người dùng nhập thông tin tài khoản để tiến hành thanh toán..

4. Khi thanh toán thành công, hệ thống cập nhật trạng thái thanh toán đơn đặt phòng trong bảng **ticket** và hiển thị màn hình thanh toán thành công. Use case kết thúc

- **Luồng rẽ nhánh:**
    1. Tại bước 2, 3, 4 trong luồng cơ bản, nếu người dùng thanh toán không thành công, hệ thống sẽ báo lỗi và yêu cầu người dùng thực hiện lại giao dịch hoặc chọn phương thức thanh toán khác.
    2. Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập.

Hậu điều kiện:

Nếu use case kết thúc thành công hệ thống sẽ cập nhập trạng thái thanh toán của đơn đặt phòng trong bảng **ticket**.

Điểm mở rộng:

Không có.

2.2.11 Đăng ký

Mô tả vắn tắt:

Use case này cho phép người dùng đăng ký tài khoản.

Luồng sự kiện:

- **Luồng cơ bản:**
    1. Use case này bắt đầu khi người dùng click vào nút **Đăng ký** tại màn hình đăng nhập. Hệ thống sẽ chuyển người dùng sang màn hình đăng ký, yêu cầu người dùng nhập họ và tên, mật khẩu, email.

2. Khi người dùng nhập thông tin và click vào nút **Đăng ký**. Hệ thống sẽ lưu dữ liệu tài khoản của người dùng vào bảng **user**.
3. Hệ thống sẽ gửi một email xác minh địa chỉ email về email mà người dùng đã đăng ký, người dùng click vào xác nhận email. Tài khoản của người dùng trên hệ thống sẽ được xác minh. Và use case kết thúc.

- **Luồng rẽ nhánh:**
  1. Tại bước 2 luồng cơ bản, nếu người dùng nhập email đã có trong cơ sở dữ liệu tại bảng **user** hoặc bỏ bất kỳ trường thông tin nào thì hệ thống sẽ báo lỗi và yêu cầu người dùng nhập lại.
  2. Tại bước 3 trong luồng cơ bản, nếu người dùng không xác minh địa chỉ email, tài khoản của người dùng sẽ không được xác minh trong hệ thống, người dùng sẽ không thể đăng nhập tài khoản này và use case kết thúc.
  3. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Không có.

Hậu điều kiện:

Nếu use case này kết thúc thành công hệ thống sẽ tạo người dùng mới trong bảng **user**.

Điểm mở rộng:

Không có.

2.2.12 Đổi mật khẩu

Mô tả vắn tắt:

Use case này cho phép người dùng đổi mật khẩu.

Luồng sự kiện:

- **Luồng cơ bản:**
    1. Use case này bắt đầu khi người dùng click nút **Đổi mật khẩu** tại giao diện trang cá nhân, hệ thống hiển thị giao diện yêu người dùng nhập mật khẩu hiện tại, mật khẩu mới, xác nhận mật khẩu mới.
    2. Người dùng nhập đầy đủ thông tin và click nút **Lưu thông tin**. Hệ thống sẽ kiểm tra mật khẩu hiện tại. Nếu mật khẩu hiện tại chính xác, hệ thống kiểm tra tính hợp lệ của mật khẩu mới. Nếu mật khẩu mới hợp lệ và khớp với mật khẩu xác nhận, hệ thống cập nhập mật khẩu mới của người dùng trong bảng **user** và hiển thị thông báo thành công. Use case kết thúc

- **Luồng rẽ nhánh:**
    1. Tại bước 2 luồng cơ bản, nếu mật khẩu hiện tại không chính xác, mật khẩu mới không hợp lệ hoặc mật khẩu xác nhận không khớp với mật khẩu mới, hệ thống sẽ hiển thị thông báo lỗi.
    2. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập

Hậu điều kiện:

Nếu use case này kết thúc thành công hệ thống sẽ cập nhập mật khẩu mới của người dùng trong bảng **user**.

Điểm mở rộng:

Không có.

2.2.13 Khôi phục mật khẩu

Mô tả vắn tắt:

Use case này cho phép người dùng đặt lại mật khẩu trong trường hợp quên mật khẩu thông qua email xác thực.

Luồng sự kiện:

- **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng chọn **Khôi phục mật khẩu** trên màn hình đăng nhập. Hệ thống sẽ chuyển sang màn hình nhập yêu cầu nhập địa chỉ email cần khôi phục mật khẩu.
2. Người dùng nhập email và chọn **Gửi mã xác thực**. Hệ thống kiểm tra email hợp lện, nếu hợp gửi một mã xác thực về **email** và chuyển hướng người dùng đến màn hình nhập mã xác thực.
3. Người dùng nhập mã xác thực và click vào nút **Xác thực**. Nếu nhập mã xác thực đúng, hệ thống sẽ chuyển đến màn hình đặt lại mật khẩu mới.
4. Người dùng nhập lại mật khẩu mới và xác nhận mật mật khẩu và click nút **Đặt lại**. Hệ thống kiểm tra hai mật khẩu trùng khớp. Nếu trùng khớp, cập nhật mật khẩu mới trong bảng **user** và hiện thị thông báo thành công. Hệ thống chuyển hướng người dùng về màn hinh trang đăng nhập. Use case kết thúc

- **Luồng rẽ nhánh:**

1. Tại bước 3, nếu mã xác thực không chính xác hệ thống sẽ hiện thị **thông báo lỗi** và người dùng được phép nhập lại mã xác thực.
2. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị **thông báo lỗi** và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải có tài khoản trên hệ thống

Hậu điều kiện:

Nếu use case kết thúc thành công thì mật khẩu sẽ được cập nhập lại trong bảng **users**.

Điểm mở rộng:

Không có.

2.2.14 Nạp tiền vào hệ thống

Mô tả vắn tắt:

Use case này cho phép người dùng nạp tiền vào tài khoản trên hệ thống thông qua cổng thanh toán trực tuyến.

Luồng sự kiện:

- **Luồng cơ bản:**
  1. Use case này bắt đầu khi người chọn **Nạp tiền** trên thanh menu giao diện trang cá nhân. Hệ thống hiện thị giao diện nạp tiên, người dùng nhập số tiền muốn nạp và click **Nạp tiền**. Hệ thống chuyển sang màn hình nhập thông tin thanh toán bao gồm email, số thẻ, ngày hết hạn, CVC, họ tên trên thẻ, quốc gia.
  2. Người dùng nhập đầy đủ thông tin và click **Pay**, hệ thống thực hiện kiểm tra giao dịch qua cổng thanh toán trực tuyến. Nếu thành công, cộng tiền vào số dư khả dụng của người dùng trong bảng **user**, tạo một bản ghi giao dịch mới trong bảng **transactions** và hiển thị màn hình nạp tiền thành công thành công. Use case kết thúc

- **Luồng rẽ nhánh:**
  1. Tại bước 2 luồng cơ bản, nếu nạp tiền không thành công hệ thống sẽ chuyển người dùng đến màn hình **Nạp tiền thất bại**.

2. Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị **thông báo lỗi** và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập.

Hậu điều kiện:

Nếu use case kết thúc thành công thì tiền sẽ được cộng vào số dư khả dụng của người dùng trong bảng **user**.

Một giao dịch mới được lưu vào bảng **transactions**.

Điểm mở rộng:

Không có

2.2.15 Xem lịch sử nạp tiền

Mô tả vắn tắt:

Use case này cho phép người dùng xem danh sách các giao dịch nạp tiền của mình trên hệ thống.

Luồng sự kiện:

- **Luồng cơ bản:**

  1. Use case này bắt đầu khi người dùng click vào **Nạp tiền** trên thanh menu trang các nhân. Hệ thống sẽ truy vấn bảng **transactions**, hiển thị danh sách lịch sử nạp tiền lên màn hình bao gồm ngày thực hiện giao dịch, loại giao dịch, số tiền nạp, trạng thái giao dịch. Use case kết thúc.

- **Luồng rẽ nhánh:**

  1. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người dùng phải đăng nhập.

Hậu điều kiện:

Không có.

Điểm mở rộng:

Không có.

2.2.16 Đăng ký tài khoản đối tác

Mô tả vắn tắt:

Use case này cho phép người dùng đăng ký tài khoản đối tác.

Luồng sự kiện:

- **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng click vào nút **Bắt đầu ngay** trên màn hình đăng ký đối tác. Hệ thống chuyển hướng sang màn hình nhập địa chỉ email.
2. Người dùng nhập địa chỉ email và click nút **Tiếp tục**. Hệ thống kiểm tra tính hợp lệ của email. Nếu email hợp lệ hệ thống chuyển hướng sang màn hình nhập thông tin cá nhân.
3. Người dùng nhập đầy đủ thông tin cá nhân bao gồm họ tên, số điện thoại và click **Tiếp theo**. Nếu hợp lệ hệ, thống chuyển sang màn hình nhập mật khẩu.
4. Người dùng nhập mật khẩu và xác nhận mật khẩu và click **Tạo tài khoản**. Hệ thống tạo môt tài khoản mới trong bảng **user** và chuyển đến màn hình yêu cầu xác thực địa chỉ email đồng thời gửi cho người dùng một email xác thực tài khoản.
5. Người dùng xác thực địa chỉ email qua đường link trong email. Hệ thống kiểm tra mã xác thực, nếu hợp lệ tài khoản đối tác sẽ được xác minh trong bảng **user** và chuyển người dùng về trang Đăng nhập. Use case kết thúc.

- **Luồng rẽ nhánh:**

1. Tại bước 2, 3, 4 trong luồng cơ bản, nếu người dùng nhập dữ liệu không hợp lệ, hệ thống sẽ hiện thị thông báo lỗi và yêu cầu nhập lại.
2. Tại bước 5 trong luồng cơ bản, nếu người dùng không xác minh địa chỉ email, tài khoản của người dùng sẽ không được xác minh trong hệ thống, người dùng sẽ không thể đăng nhập tài khoản này và use case kết thúc.
3. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Không có.

Hậu điều kiện:

Nếu use case kết thúc thành công một tài khoản đối tác được tạo trong bảng user.

Điểm mở rộng:

Không có.

2.2.17 Xem thống kê đặt phòng và doanh thu

Mô tả vắn tắt:

Use case này cho phép đối tác xem thống kê đơn đặt phòng và doanh thu của mình trên hệ thống.

Luồng sự kiện:

- **Luồng cơ bản:**

1. Use case này bắt đầu khi đối tác click vào nút **Bảng điều khiển** trên thanh menu. Hệ thống sẽ chuyển hướng sang màn hình **Bảng điều khiển** hiển thị tổng đơn đặt phòng, tổng doanh thu,

danh sách đơn đặt phòng mới, thống kê số đơn đặt phòng và doanh thu theo từng tháng trong bảng **ticket**. Use case kết thúc.

- **Luồng rẽ nhánh:**

1. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người quản trị phải đăng nhập với quyền đối tác trước khi thực hiện use case.

Hậu điều kiện:

Không có.

Điểm mở rộng:

Không có.

2.2.18 Quản lý khách sạn

Mô tả vắn tắt:

Use case này cho phép đối tác có thể xem, thêm, sửa, xóa các khách sạn của mình trên hệ thống quản trị.

Luồng sự kiện:

- **Luồng cơ bản:**

1. Use case này bắt đầu khi đối tác click vào nút **Khách sạn** trên thanh menu hệ thống quản trị. Hệ thống sẽ truy vấn bảng **hotel**, hiển thị danh sách các khách sạn gồm tên khách sạn, đánh giá, địa chỉ, số điện thoại, mô tả lên màn hình.
2. **Thêm khách sạn:**
   a. khi đối tác chọn **Thêm khách sạn,** hệ thống hiển thị form nhập thông tin chi tiết về khách sạn gồm: tên khách sạn, địa chỉ, mô tả, ảnh, số điện thoại.

b. Đối tác nhập thông tin chi tiết về khách sạn và click vào nút **Thêm khách sạn**. Hệ thống sẽ kiểm tra dữ liệu nhập vào và tạo một bản ghi mới trong bảng **hotel** đồng thời cập nhật khách sạn này lên danh sách.

3. Sửa khách sạn:

a. Khi đối tác click vào nút **Sửa** trên một thẻ thông tin về khách sạn. Hệ thống sẽ hiển thị các trường cho phép đối tác sửa thông tin của khách sạn.

b. Đối tác thay đổi thông tin khách sạn gồm các thông tin được chỉ ra trong **Thêm khách sạn** và click vào nút **Lưu thay đổi**. Hệ thống kiểm tra thông tin thay đổi, cập nhật bảng **hotel** và cập nhật danh khách sạn.

4. Xóa khách sạn:

a. khi đối tác click vào nút **Xóa** trên một thẻ thông tin về khách sạn, hệ thống sẽ hiển thị form xác nhận xóa.

b. Đối tác click nút **Xác nhận,** hệ thống sẽ xóa bản ghi khách sạn khỏi bảng **hotel** và cập nhật danh sách khách sạn.

5. Use case kết thúc khi đối tác đóng cửa sổ.

- **Luồng rẽ nhánh:**

1. Tại bước 2b hoặc bước 3b trong luồng cơ bản. Khi thông tin về khách sạn, nhập không đủ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập bổ sung.
2. Tại bước 4b trong luồng cơ bản. Khi đối tác click nút **Hủy,** hệ thống sẽ không thực hiện xóa khách sạn và use case kết thúc.
3. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người quản trị phải đăng nhập với quyền đối tác trước khi thực hiện use case.

Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin khách sạn sẽ được cập nhật vào bảng hotel.

Điểm mở rộng:

Không có.

2.2.19 Quản lý loại phòng

Mô tả vắn tắt:

Use case này cho phép đối tác có thể xem, thêm, sửa, xóa các loại phòng trong khách sạn của mình trên hệ thống quản trị.

Luồng sự kiện:

- **Luồng cơ bản:**
  1. Use case này bắt đầu khi đối tác chủ khác sạn click vào nút **Loại phòng** trên thanh menu hệ thống quản trị. Hệ thống sẽ truy vấn bảng **type_room**, hiển thị danh sách các loại phòng gồm tên khách sạn, tên loại phòng, mô tả, ngày tạo lên màn hình.
  2. **Thêm loại phòng:**
     a. khi đối tác chọn **Thêm loại phòng,** hệ thống hiển thị form nhập thông tin chi tiết về loại phòng gồm: tên loại phòng, mô tả và chọn khách sạn.
     b. Đối tác nhập thông tin chi tiết về loại phòng và click vào nút **Thêm loại phòng**. Hệ thống sẽ kiểm tra dữ liệu nhập vào và tạo một bản ghi mới trong bảng **type_room** đồng thời cập nhật loại phòng này lên danh sách.
  3. **Sửa loại phòng:**

a. Khi đối tác click vào nút **Sửa** trên một thẻ thông tin về loại phòng. Hệ thống sẽ hiển thị các trường cho phép đối tác sửa thông tin của loại phòng

b. Đối tác thay đổi thông tin loại phòng gồm các thông tin được chỉ ra trong **Thêm loại phòng** và click vào nút **Cập nhật**. Hệ thống kiểm tra thông tin thay đổi, cập nhật bảng **type_room** và cập nhật danh loại phòng.

4. **Xóa loại phòng:**

a. khi đối tác click vào nút **Xóa** trên một thẻ thông tin về loại phòng, hệ thống sẽ hiện thị form xác nhận xóa.

b. Đối tác click nút **Xác nhận,** hệ thống sẽ xóa bản ghi loại phòng khỏi bảng **type_room** và cập nhật danh sách loại phòng.

5. Use case kết thúc khi đối tác đóng cửa sổ.

➢ **Luồng rẽ nhánh:**

1. Tại bước 2b hoặc bước 3b trong luồng cơ bản. Khi thông tin về khách sạn, nhập không đủ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập bổ sung.
2. Tại bước 4b trong luồng cơ bản. Khi đối tác click nút **Hủy,** hệ thống sẽ không thực hiện xóa loại phòng và use case kết thúc.
3. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người quản trị phải đăng nhập với quyền đối tác trước khi thực hiện use case.

Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin loại phòng sẽ được cập nhật vào bảng type_room.

Điểm mở rộng:

Không có.

2.2.20 Quản lý phòng

Mô tả vắn tắt:

Use case này cho phép đối tác có thể xem, thêm, sửa, xóa các phòng trong loại phòng của mình trên hệ thống quản trị.

Luồng sự kiện:

- **Luồng cơ bản:**
  1. Use case này bắt đầu khi đối tác click vào nút **Phòng** trên thanh menu hệ thống quản trị. Hệ thống sẽ truy vấn bảng **room**, hiển thị danh sách các phòng gồm tên phòng, ảnh phòng, mô tả, tên loại phòng, tên khách sạn, sức chứa, giá, tiện ích, trạng thái đặt phòng lên màn hình.
  2. **Thêm phòng:**
     a. khi đối tác chọn **Thêm phòng**, hệ thống hiện thị form nhập thông tin chi tiết về phòng gồm: tên phòng, mô tả, sức chứa, giá, ảnh phòng và chọn loại phòng.
     b. Đối tác nhập thông tin chi tiết về phòng click vào nút **Thêm phòng**. Hệ thống sẽ kiểm tra dữ liệu nhập vào và tạo một bản ghi mới trong bảng **room** đồng thời cập nhật phòng này lên danh sách.
  3. **Sửa phòng:**
     a. Khi đối tác click vào nút **Sửa** trên một thẻ thông tin về phòng. Hệ thống sẽ hiển thị các trường cho phép đối tác sửa thông tin của phòng.
     b. Đối tác thay đổi thông tin phòng gồm các thông tin được chỉ ra trong **Thêm phòng** và click vào nút **Cập nhật**

phòng. Hệ thống kiểm tra thông tin thay đổi, cập nhật bảng **room** và cập nhật danh phòng.

4. **Xóa phòng:**
   a. khi đối tác click vào nút **Xóa** trên một thẻ thông tin về phòng, hệ thống sẽ hiện thị form xác nhận xoa.
   b. Đối tác click nút **Xác nhận**, hệ thống sẽ xóa bản ghi phòng khỏi bảng **room** và cập nhật danh sách phòng.
5. Use case kết thúc khi đối tác đóng cửa sổ.

➢ **Luồng rẽ nhánh:**

1. Tại bước 2b hoặc bước 3b trong luồng cơ bản. Khi thông tin về khách sạn, nhập không đủ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập bổ sung.
2. Tại bước 4b trong luồng cơ bản. Khi đối tác click nút **Hủy,** hệ thống sẽ không thực hiện xóa phòng và use case kết thúc.
3. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người quản trị phải đăng nhập với quyền đối tác trước khi thực hiện use case.

Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin phòng sẽ được cập nhật vào bảng room.

Điểm mở rộng:

Không có.

2.2.21 Cập nhật đơn đặt phòng

Mô tả vắn tắt:

Use case này cho phép đối tác có thể xem, sửa, xóa đơn đặt phòng trên hệ thống quản trị.

Luồng sự kiện:

- **Luồng cơ bản:**

1. Use case này bắt đầu khi đối tác click vào nút **Đặt chỗ** trên thanh menu hệ thống quản trị. Hệ thống sẽ truy vấn bảng **ticket**, hiển thị danh sách các đơn đặt phòng gồm mã phòng, tên khách hàng, tên phòng, sức chưa, tiện ích, ngày đặt phòng, ngày nhận phòng, ngày trả phòng, giá, trạng thái thanh toán, phương thức thanh toán lên màn hình.
2. **Sửa đơn đặt phòng:**
   a. Khi đối tác click vào **Biểu tượng sửa** trên một dòng thông tin về đơn đặt phòng. Hệ thống sẽ hiển thị các trường cho phép đối tác sửa thông tin của đơn đặt phòng.
   b. Đối tác thay đổi thông tin khách sạn gồm các thông tin tên khách hàng, tiện ích, ngày nhận phòng, ngày trả phòng, giờ nhận phòng, giờ trả phòng, giá phòng, phương thức thanh toán, trạng thái thanh toán và click vào nút **Lưu**. Hệ thống kiểm tra thông tin thay đổi, cập nhật bảng **ticket** và cập nhật danh đơn đặt phòng.
3. **Xóa đơn đặt phòng:**
   a. khi đối tác click vào nút **Xóa** trên một dòng thông tin về đơn đặt phòng, Hệ thống sẽ hiện thị form xác nhận xóa.
   b. Đối tác click nút **Xác nhận**, hệ thống sẽ xóa bản ghi đơn đặt phòng khỏi bảng **ticket** và cập nhật danh sách đơn đặt phòng.
4. Use case kết thúc khi đối tác đóng cửa sổ.

- **Luồng rẽ nhánh:**

1. Tại bước 2b hoặc bước 3b trong luồng cơ bản. Khi thông tin về khách sạn, nhập không đủ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập bổ sung.
2. Tại bước 3b trong luồng cơ bản. Khi đối tác click nút **Hủy,** hệ thống sẽ không thực hiện xóa đơn đặt phòng và use case kết thúc.
3. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người quản trị phải đăng nhập với quyền đối tác trước khi thực hiện use case.

Hậu điều kiện:

Nếu use case kết thúc thành công thì thông tin đơn đặt phòng sẽ được cập nhật vào bảng ticket.

Điểm mở rộng:

Không có.

2.2.22 Quản lý đánh giá khách sạn

Mô tả vắn tắt:

Use case này cho phép đối tác có thể xem, thêm, sửa, xóa các đánh giá trong khách sạn của mình trên hệ thống quản trị.

Luồng sự kiện:

- **Luồng cơ bản:**

1. Use case này bắt đầu khi đối tác click vào nút **Đánh giá** trên thanh menu hệ thống quản trị. Hệ thống sẽ truy vấn bảng **review**, hiển thị danh sách các đánh giá gồm tên khách sạn, tên khách hàng, đánh giá, nội dung bình luận lên màn hình.
2. **Thêm đánh giá:**

a. khi đối tác click vào **Thêm đánh giá,** hệ thống hiện thị form nhập thông tin chi tiết về đánh giá gồm: tên khách hàng, đánh giá, nội dung bình luận, hình ảnh và chọn khách sạn.

b. Đối tác nhập thông tin chi tiết về đánh giá click vào nút **Thêm đánh giá**. Hệ thống sẽ kiểm tra dữ liệu nhập vào và tạo một bản ghi mới trong bảng **review** đồng thời cập nhật đánh giá này lên danh sách.

3. Sửa đánh giá:

a. Khi đối tác click vào nút **Sửa** trên một thẻ thông tin về đánh giá. Hệ thống sẽ hiển thị các trường cho phép đối tác sửa thông tin của đánh giá.

b. Đối tác thay đổi thông tin khách sạn gồm các thông tin được chỉ ra trong **Thêm đánh giá** và click vào nút **Lưu**. Hệ thống kiểm tra thông tin thay đổi, cập nhật bảng **review** và cập nhật danh đánh giá.

4. Xóa đánh giá:

a. khi đối tác click vào nút **Xóa** trên một thẻ thông tin về đánh giá, hệ thống sẽ hiện thị form xác nhận xóa.

b. Đối tác click nút **Xác nhận**, hệ thống sẽ xóa bản ghi đánh giá khỏi bảng **review** và cập nhật danh sách đánh giá.

5. Use case kết thúc khi đối tác đóng cửa sổ.

- **Luồng rẽ nhánh:**

1. Tại bước 2b hoặc bước 3b trong luồng cơ bản. Khi thông tin về khách sạn, nhập không đủ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập bổ sung.
2. Tại bước 4b trong luồng cơ bản. Khi đối tác click nút **Hủy,** hệ thống sẽ không thực hiện xóa đánh giá và use case kết thúc.

3. Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi và use case kết thúc.

Các yêu cầu đặc biệt:

Không có.

Tiền điều kiện:

Người quản trị phải đăng nhập với quyền đối tác trước khi thực hiện use case.

Hậu điều kiện:

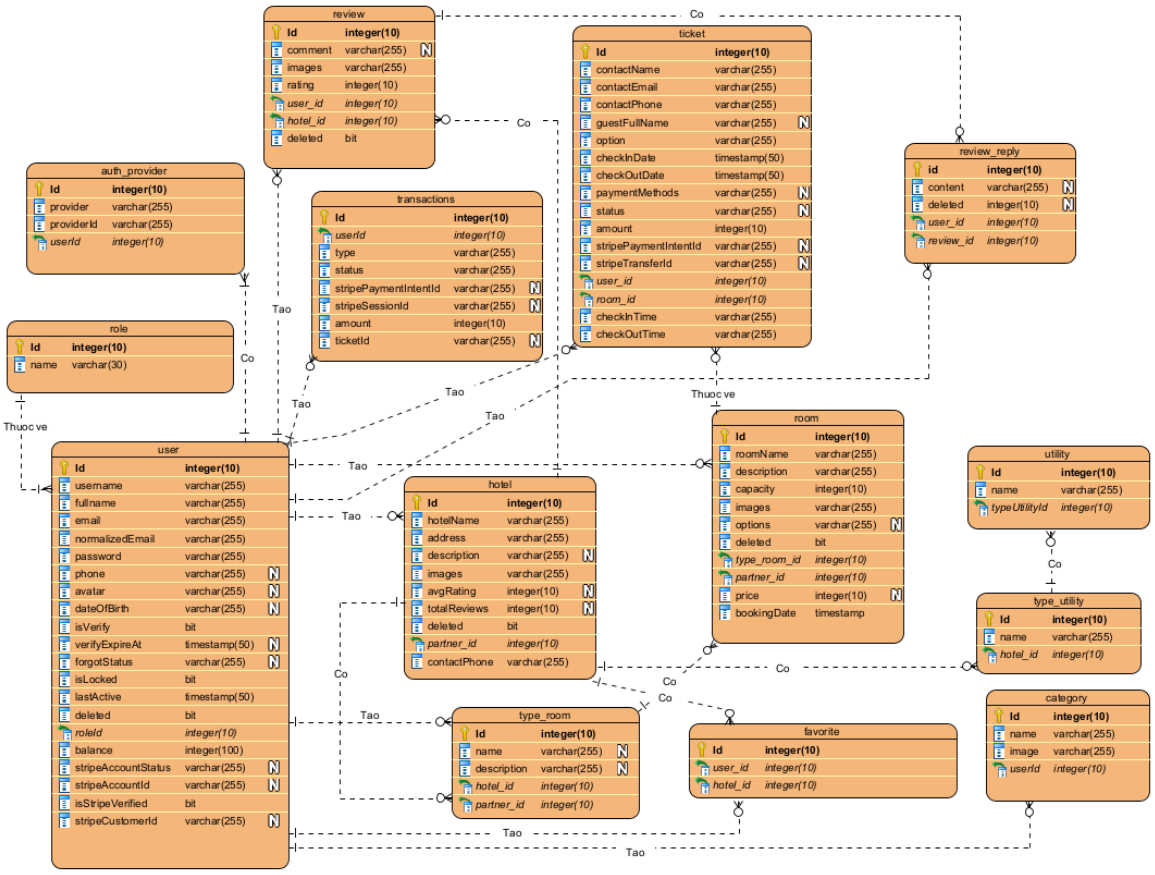
Nếu use case kết thúc thành công thì thông tin đánh giá sẽ được cập nhật vào bảng review.

Điểm mở rộng:

Không có.

CHƯƠNG 3. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.1 Mô hình hóa dữ liệu



Hình 3.1 Các bảng trong CSDL

auth_provider: Lưu thông tin liên kết giữa người dùng và các phương thức xác thực bên thứ ba. Chi tiết về bảng auth_provider gồm id, provider, providerId, user.

user: Lưu thông tin về người dùng. Chi tiết về bảng user bao gồm id, username, fullname, email, normalizedEmail, password, phone, avatar, dateOfBirth, isVerify, verifyExpireAt, forgotStatus, isLocked, lastActive, stripeAccountId, stripeCustomerId, stripeAccountStatus, isStripeVerified, balance, role, reivews, reviewReplies, favorites, hotels, rooms, type_rooms, authProviders, transactions, deleted.

role: Lưu thông tin về vai trò của người dùng. Chi tiết về bảng role bao gồm id, name, users.

category: Lưu thông tin về danh mục. Chi tiết về bảng category bao gồm id, name, image, userId.

favorite: Lưu thông tin về các khách sạn yêu thích. Chi tiết về bảng favorite bao gồm id, user_id, hotel_id.

hotel: Lưu thông tin về các khách sạn. Chi tiết về bảng hotel bao gồm id, hotelName, address, description, images, partner, contactPhone, reviews, typeRooms, favorites, avgRating, totalReviews, typeUtility, deleted.

review: Lưu thông tin về các đánh giá. Chi tiết về bảng review bao gồm id, user_id, hotel_id, customerName, comment, images, rating, replies, deleted.

review_reply: Lưu thông tin về chi tiết các phản hồi đánh giá. Chi tiết về bảng review_reply bao gồm id, review_id, user_id, content, deleted.

room: Lưu thông tin về các phòng. Chi tiết về bảng room gồm id, roomName, description, capacity, images, options, price, type_room_id, partner_id, tickets, bookingDate, deleted.

ticket: Lưu thông tin về các đơn đặt phòng của người dùng. Chi tiết về bảng ticket gồm id, user_id, contactName, contactEmail, contactPhone, guestFullName, option, room_id, checkInDate, checkOutDate, checkInTime, checkOutTime, paymentMethods, status, amount, stripePaymentIntentId, stripeTransferId.

transactions: Lưu thông tin về các giao dịch của người dùng. Chi tiết về bảng transactions gồm id, amount, status, type, stripePaymentIntentId, stripeSessionId, userId, ticketId.

type_room: Lưu thông tin về loại phòng của khách sạn. Chi tiết bảng type_room gồm id, name, description, rooms, hotel_id, partner_id.

type_utility: Lưu thông tin về loại tiện ích của khách sạn. Chi tiết bảng type_utility gồm id, name, hotel_id, utilities.

utility: Lưu thông tin về các tiện ích của khách sạn. Chi tiết bảng utility gồm id, name, typeUtility.

Mỗi vai trò có thể thuộc về 1 hoặc nhiều người dùng. Mỗi người dùng chỉ có 1 vai trò.

Mỗi người dùng có thể có 1 hoặc nhiều phương thức liên kết tài khoản. Mỗi phương thức liên kết tài khoản chỉ thuộc về 1 người dùng.

Mỗi người dùng có thể đánh giá 0, 1 hoặc nhiều khách sạn. Mỗi đánh giá thuộc về 1 người dùng.

Mỗi người dùng có thể tạo 0, 1 hoặc nhiều phản hồi đánh giá. Mỗi đánh giá thuộc về 1 người dùng.

Mỗi người dùng có thể tạo 0, 1 hoặc nhiều khách sạn yêu thích. Mỗi khách sạn yêu thích thuộc về 1 người dùng.

Mỗi người dùng có thể tạo 0, 1 hoặc nhiều đơn đặt phòng. Mỗi đơn đặt phòng thuộc về 1 người dùng.

Mỗi người dùng có thể tạo 0, 1 hoặc nhiều giao dịch. Mỗi giao dịch thuộc về 1 người dùng.

Mỗi người dùng với vai trò là đối tác có thể tạo 0, 1 hoặc nhiều khách sạn. Mỗi khách sạn chỉ thuộc về 1 người dùng.

Mỗi người dùng với vai trò là đối tác có thể tạo 0, 1 hoặc nhiều loại phòng. Mỗi loại phòng thuộc về 1 người dùng.

Mỗi người dùng với vai trò là đối tác có thể tạo 0, 1, hoặc nhiều phòng. Mỗi phòng thuộc về 1 người dùng.

Mỗi người dùng với vai trò là quản trị có thể tạo 0, 1, hoặc nhiều danh mục. Mỗi danh mục thuộc về 1 người dùng.

Mỗi khách sạn có thể có 0, 1 hoặc nhiều loại phòng. Mỗi loại phòng chỉ thuộc về 1 khách sạn.

Mỗi khách sạn có thể có 0, 1 hoặc nhiều đánh giá. Mỗi đánh giá chỉ thuộc về 1 khách sạn.

Mỗi khách sạn có thể có 0, 1 hoặc nhiều loại tiện nghi. Mỗi loại tiện nghi chỉ thuộc về 1 khách sạn.

Mỗi khách sạn có thể có 0, 1 hoặc nhiều khách sạn yêu thích. Mỗi khách sạn yêu thích chỉ thuộc về 1 khách sạn.

Mỗi loại phòng có thể có 0, 1 hoặc nhiều phòng. Mỗi phòng chỉ thuộc về 1 loại phòng.

Mỗi loại tiện nghi có thể có 0, 1 hoặc nhiều tiện nghi. Mỗi tiện nghi thuộc về 1 loại tiện nghi.

Mỗi đánh giá có thể có 0, 1 hoặc nhiều phản hồi đánh giá. Mỗi phản hồi đánh giá chỉ thuộc về 1 đánh giá.

Mỗi phòng có thể có 0, 1 hoặc nhiều đơn đặt phòng. Mỗi đặt phòng chỉ thuộc về 1 phòng.

3.2 Thiết kế bảng

Bảng 3.1. Chi tiết bảng auth_provider

auth_provider

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	uuid			✓	✓	uuid_generate_v4
provider	character varying			✓	✗	
providerId	character varying			✓	✗	
created_at	timestamp without time zone			✓	✗	now()
updated_at	timestamp without time zone			✓	✗	now()
userId	integer			✗	✗	

Bảng 3.2. Chi tiết bảng user

user

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('user_id_seq':
username	character varying			✓	✗	
fullname	character varying			✗	✗	
email	character varying			✓	✗	
normalizedEmail	character varying			✓	✗	
password	character varying			✓	✗	
phone	character varying			✗	✗	
avatar	character varying			✗	✗	'https://p21-ad-sg.iby
dateOfBirth	timestamp without time zone			✗	✗	
isVerify	boolean			✓	✗	false
verifyExpireAt	timestamp without time zone			✗	✗	
forgotStatus	user_forgotstatus_enum			✗	✗	

isLocked	boolean					false
lastActive	timestamp without time zone					'2024-12-09 14:56:45
created_at	timestamp without time zone					now()
updated_at	timestamp without time zone					now()
deleted	boolean					false
roleId	integer					
balance	integer					0
stripeAccountId	character varying					
stripeAccountStatus	jsonb					
isStripeVerified	boolean					false
stripeCustomerId	character varying					

Bảng 3.3. Chi tiết bảng role

role

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer					nextval('role_
name	character varying	30				
permissionIds	integer[]					'{}'::integer[]
created_at	timestamp without time zone					now()
updated_at	timestamp without time zone					now()

Bảng 3.4. Chi tiết bảng category

category

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer					nextval('category_
name	character varying					
image	character varying					
userId	integer					

Bảng 3.5. Chi tiết bảng favorite

favorite

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer					nextval('favorite_id
user_id	integer					
hotel_id	integer					

Bảng 3.6. Chi tiết bảng hotel

hotel

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('hotel_id_se
hotelName	character varying			✓	✗	
address	character varying			✓	✗	
description	character varying			✗	✗	
images	text[]			✓	✗	
avgRating	integer			✗	✗	
totalReviews	integer			✗	✗	
created_at	timestamp without time zone			✓	✗	now()
updated_at	timestamp without time zone			✓	✗	now()
deleted	boolean			✓	✗	false
partner_id	integer			✓	✗	
contactPhone	character varying			✓	✗	

Bảng 3.7. Chi tiết bảng review

review

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('review_id_s
comment	character varying	255		✓	✗	
images	text[]			✗	✗	
rating	integer			✗	✗	
created_at	timestamp without time zone			✓	✗	now()
updated_at	timestamp without time zone			✓	✗	now()
deleted	boolean			✓	✗	false
user_id	integer			✗	✗	
hotel_id	integer			✗	✗	

Bảng 3.8. Chi tiết bảng review_reply

review_reply

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('review_repl
content	text			✓	✗	
created_at	timestamp without time zone			✓	✗	now()
updated_at	timestamp without time zone			✓	✗	now()
deleted	boolean			✓	✗	false
review_id	integer			✗	✗	
user_id	integer			✗	✗	

Bảng 3.9. Chi tiết bảng room

room

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('room_id_s
roomName	character varying			✓		'default room'::char
description	character varying					
capacity	integer			✓		1
images	text[]			✓		
options	json					
price	integer			✓		0
created_at	timestamp without time zone			✓		now()
updated_at	timestamp without time zone			✓		now()
deleted	boolean			✓		false
type_room_id	integer					
partner_id	integer			✓		
bookingDate	timestamp without time zone					

Bảng 3.10. Chi tiết bảng ticket

ticket

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('ticket_id_se
contactName	character varying			✓		
contactEmail	character varying			✓		
contactPhone	character varying			✓		
guestFullName	character varying					
option	text[]			✓		
checkInDate	timestamp without time zone			✓		
checkOutDate	timestamp without time zone			✓		
checkInTime	character varying			✓		
checkOutTime	character varying			✓		
paymentMethods	ticket_paymentmethods_enum					
status	ticket_status_enum					'pending'::ticket_sta
amount	integer			✓		0
stripePaymentIntentId	character varying					
stripeTransferId	character varying					
created_at	timestamp without time zone			✓		now()
updated_at	timestamp without time zone			✓		now()
user_id	integer					
room_id	integer					

Bảng 3.11. Chi tiết bảng transactions

transactions

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
userId	integer			✓		
type	transactions_type_enum			✓		
status	transactions_status_enum			✓		'PENDING'::transactio
createdAt	timestamp without time zone			✓		now()
updatedAt	timestamp without time zone			✓		now()
stripePaymentIntentId	character varying					
stripeSessionId	character varying					
id	integer			✓	✓	nextval('transactions_
amount	integer			✓		
ticketId	character varying					

Bảng 3.12. Chi tiết bảng type_room

type_room

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('type_room_i
name	character varying					
description	character varying					
created_at	timestamp without time zone			✓		now()
updated_at	timestamp without time zone			✓		now()
hotel_id	integer					
partner_id	integer			✓		

Bảng 3.13. Chi tiết bảng type_utility

type_utility

General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('type_utility_id_s
name	character varying			✓		
created_at	timestamp without time zone			✓		now()
updated_at	timestamp without time zone			✓		now()
hotel_id	integer					

Bảng 3.14. Chi tiết bảng utility

utility

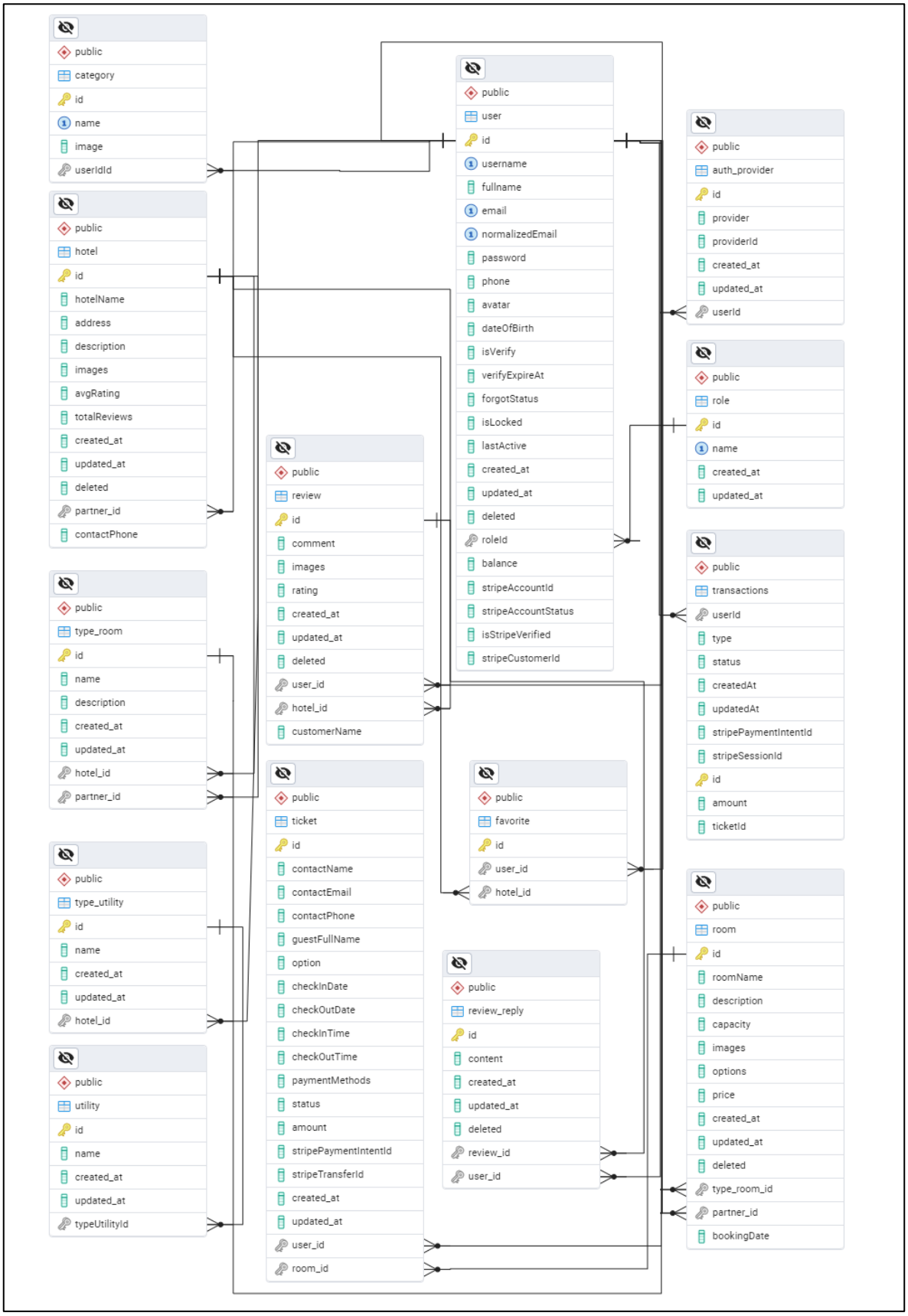
General | Columns | Advanced | Constraints | Parameters | Security | SQL

Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
id	integer			✓	✓	nextval('utility_id_
name	character varying			✓		
created_at	timestamp without time zone			✓		now()
updated_at	timestamp without time zone			✓		now()
typeUtilityId	integer			✓		

3.3 Quan hệ các bảng

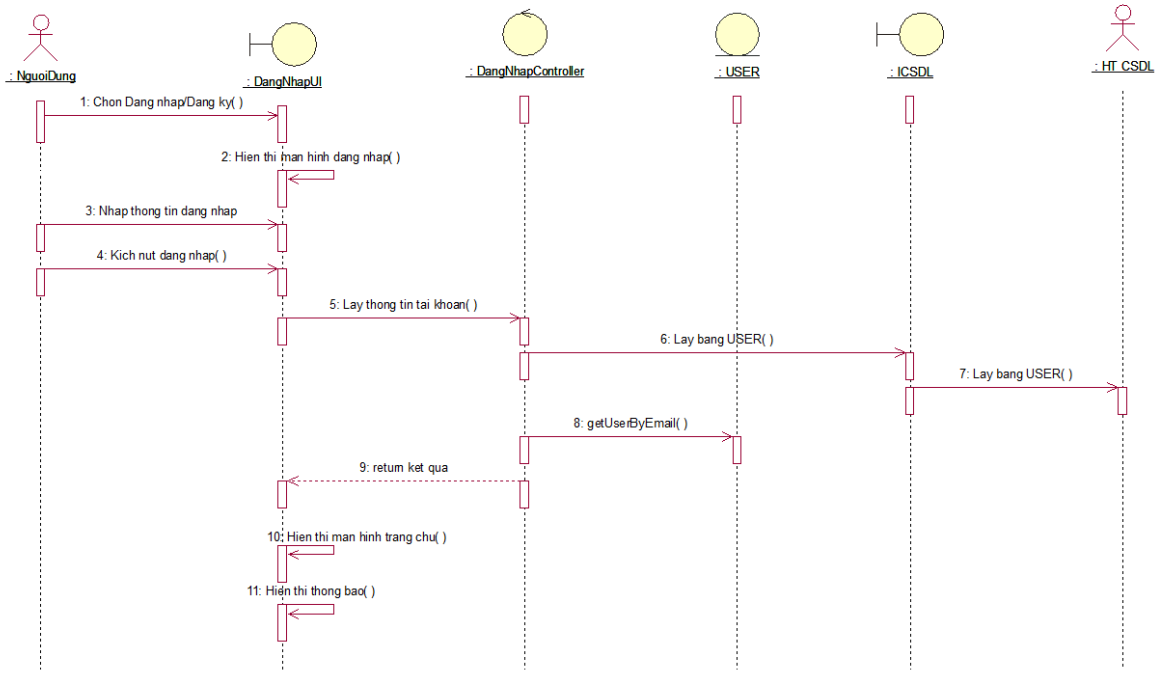


Hình 3.2. Quan hệ các bảng

CHƯƠNG 4. THIẾT KẾ CÁC CHỨC NĂNG CỦA HỆ THỐNG

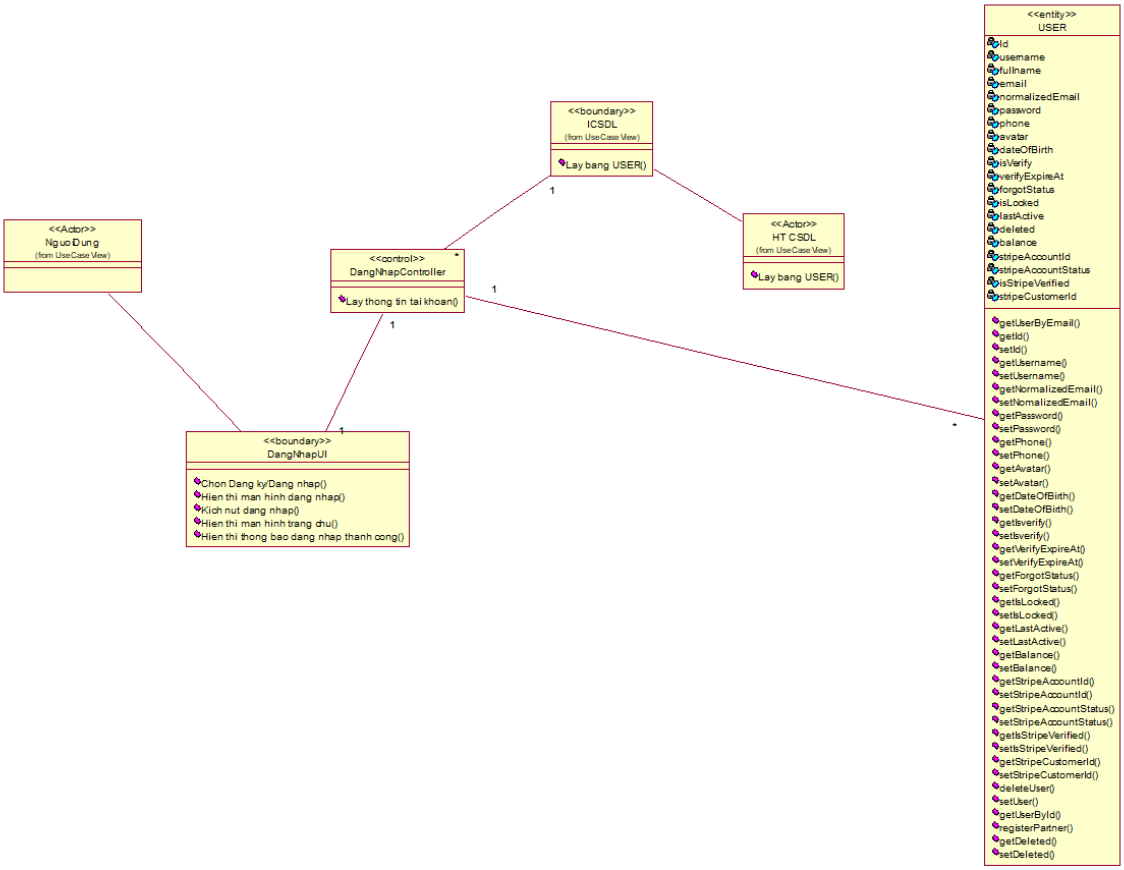
4.1 Use case Đăng nhập

4.1.1 Biểu đồ trình tự:



Hình 4.1. Biểu đồ trình tự use case Đăng nhập

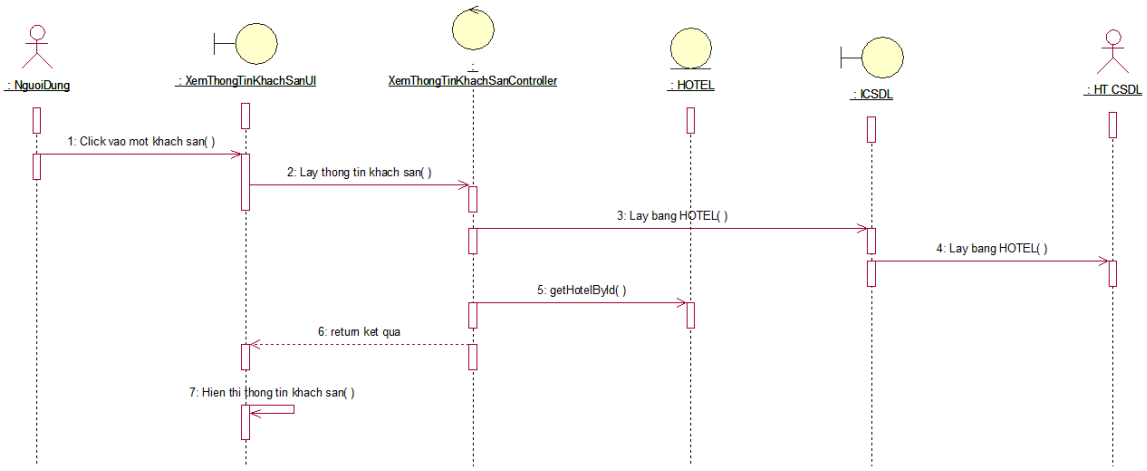
4.1.2 Biểu đồ lớp phân tích:



Hình 4.2. Biểu đồ lớp use case Đăng nhập

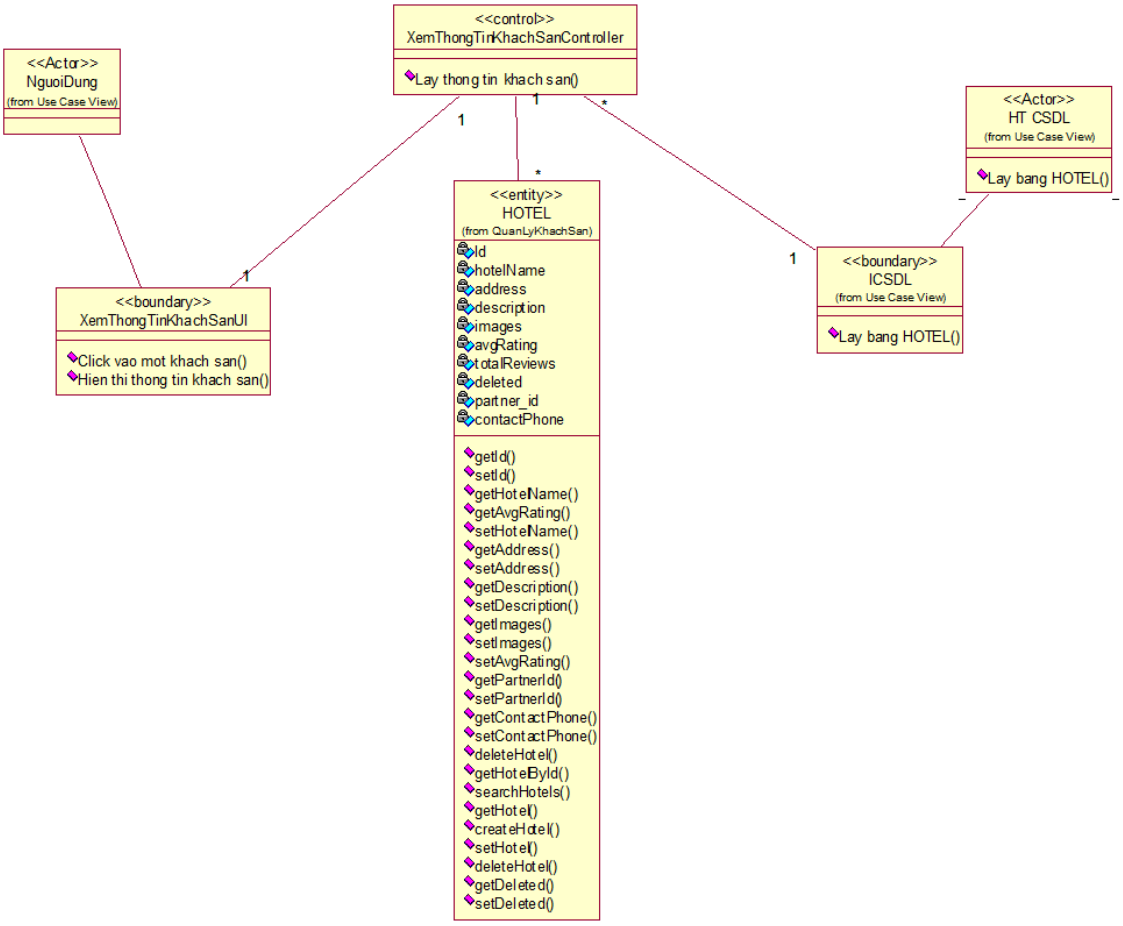
4.2 Use case Xem thông tin khách sạn

4.2.1 Biểu đồ trình tự



Hình 4.3. Biểu đồ trình tự use case Xem thông tin khách sạn

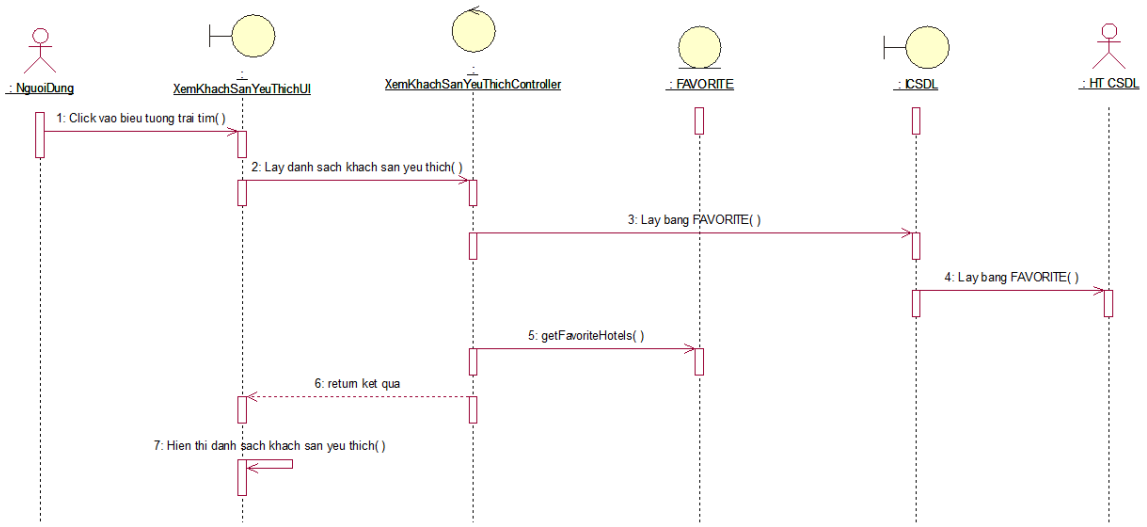
4.2.2 Biểu đồ lớp



Hình 4.4. Biểu đồ lớp use case Xem thông tin khách sạn

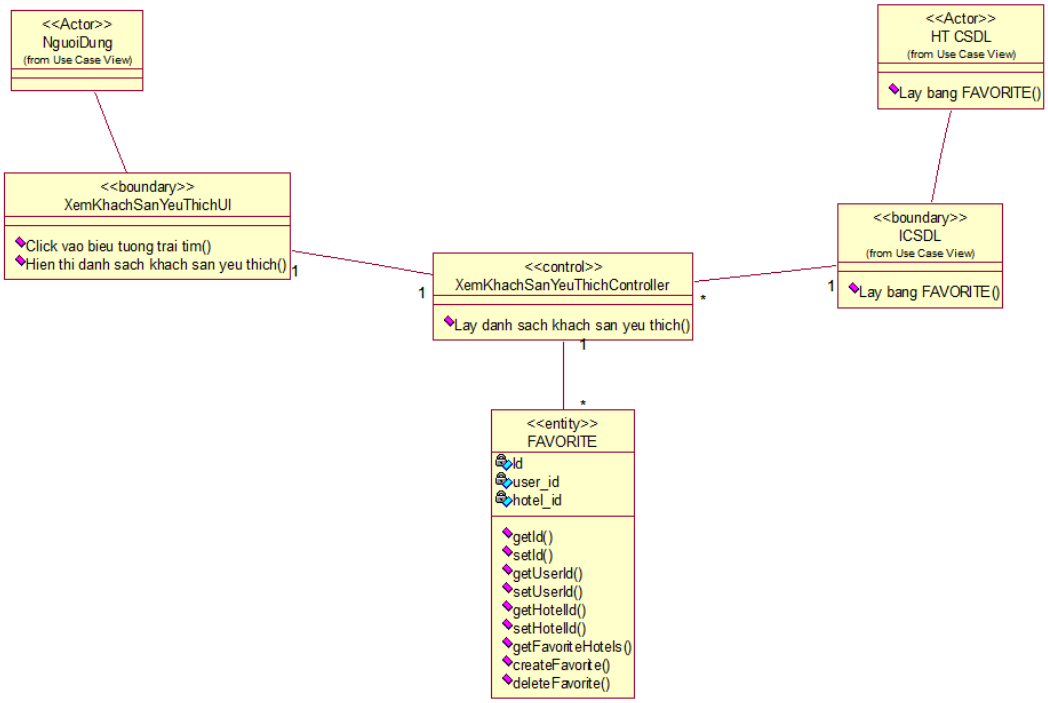
4.3 Use case Xem khách sạn yêu thích

4.3.1 Biểu đồ trình tự



Hình 4.5. Biểu đồ trình tự use case Xem khách sạn yêu thích

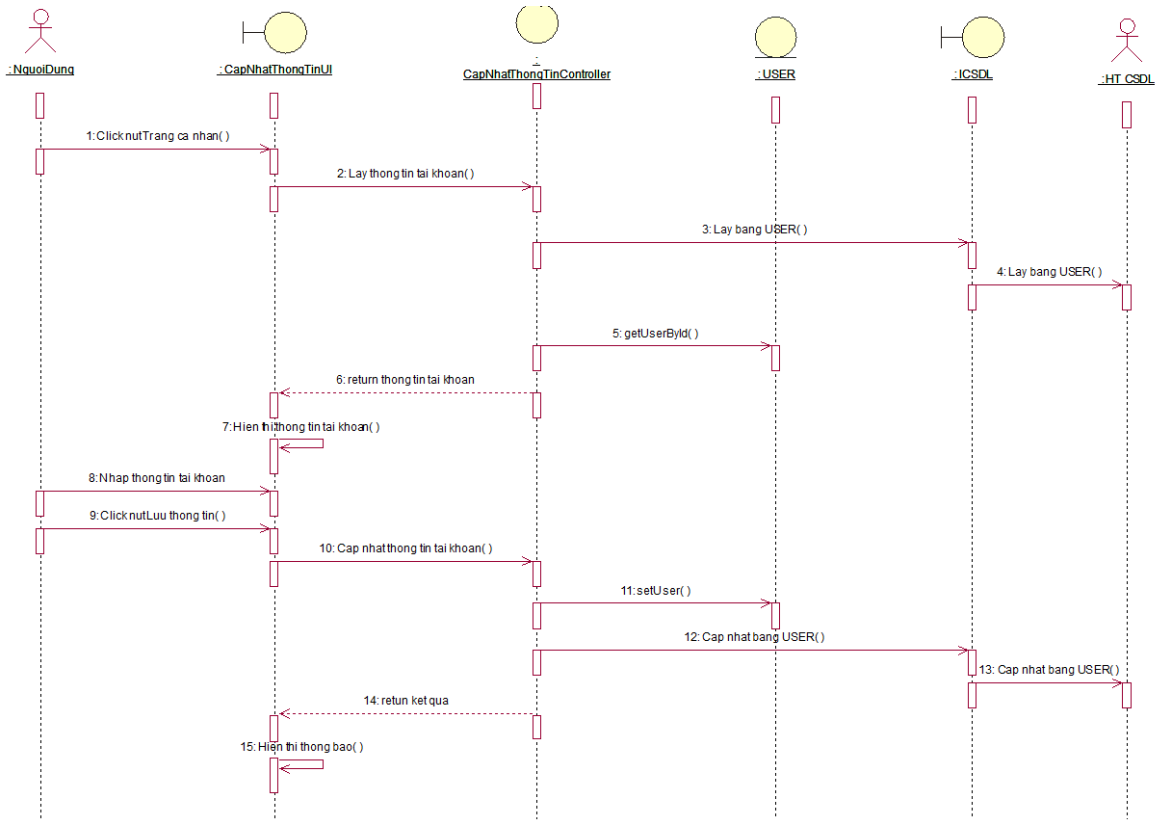
4.3.2 Biểu đồ lớp



Hình 4.6. Biểu đồ lớp use case Xem khách sạn yêu thích

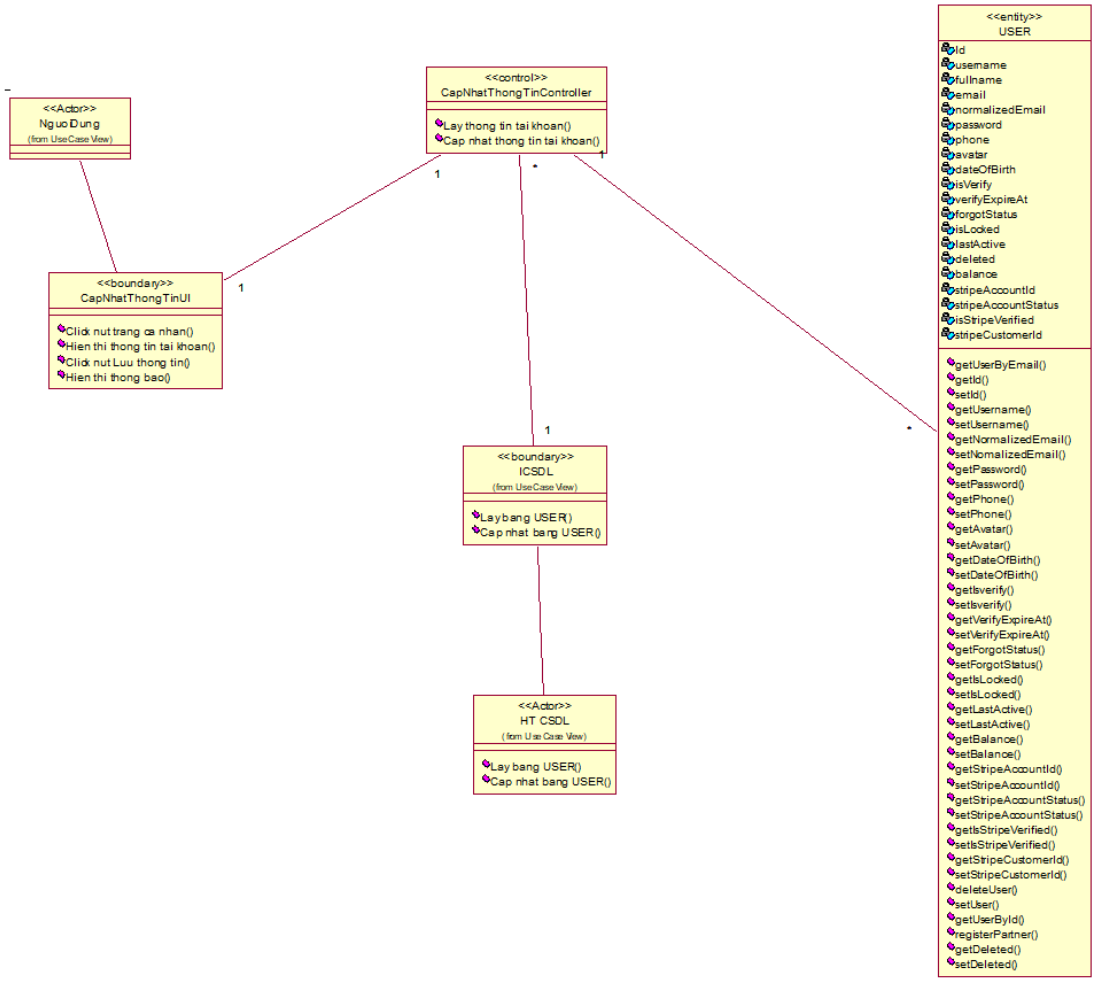
4.4 Use case Cập nhật thông tin

4.4.1 Biểu đồ trình tự



Hình 4.7. Biểu đồ trình tự use case Cập nhật thông tin

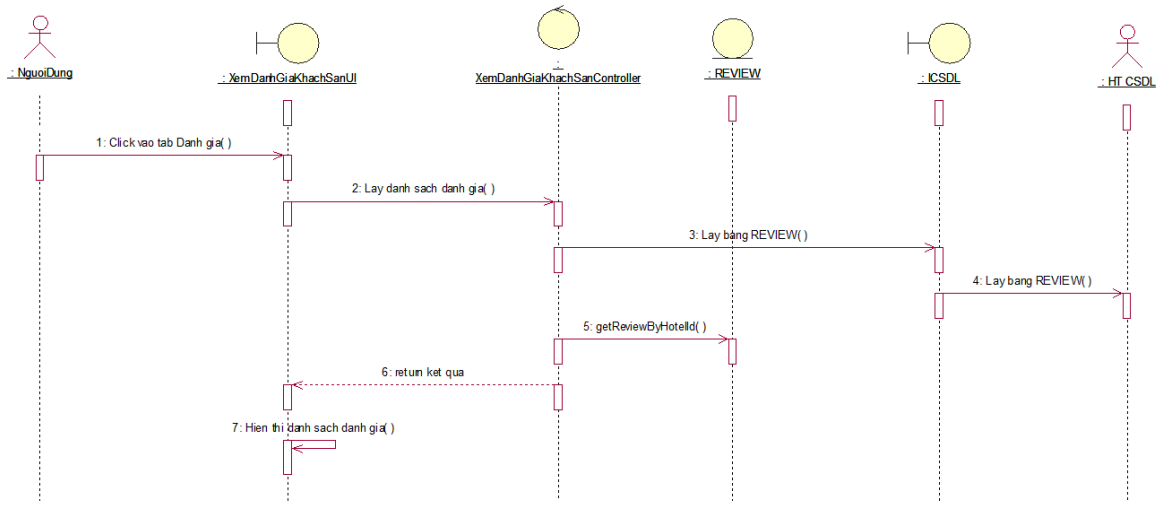
4.4.2 Biểu đồ lớp



Hình 4.8. Biểu đồ lớp use case Cập nhật thông tin

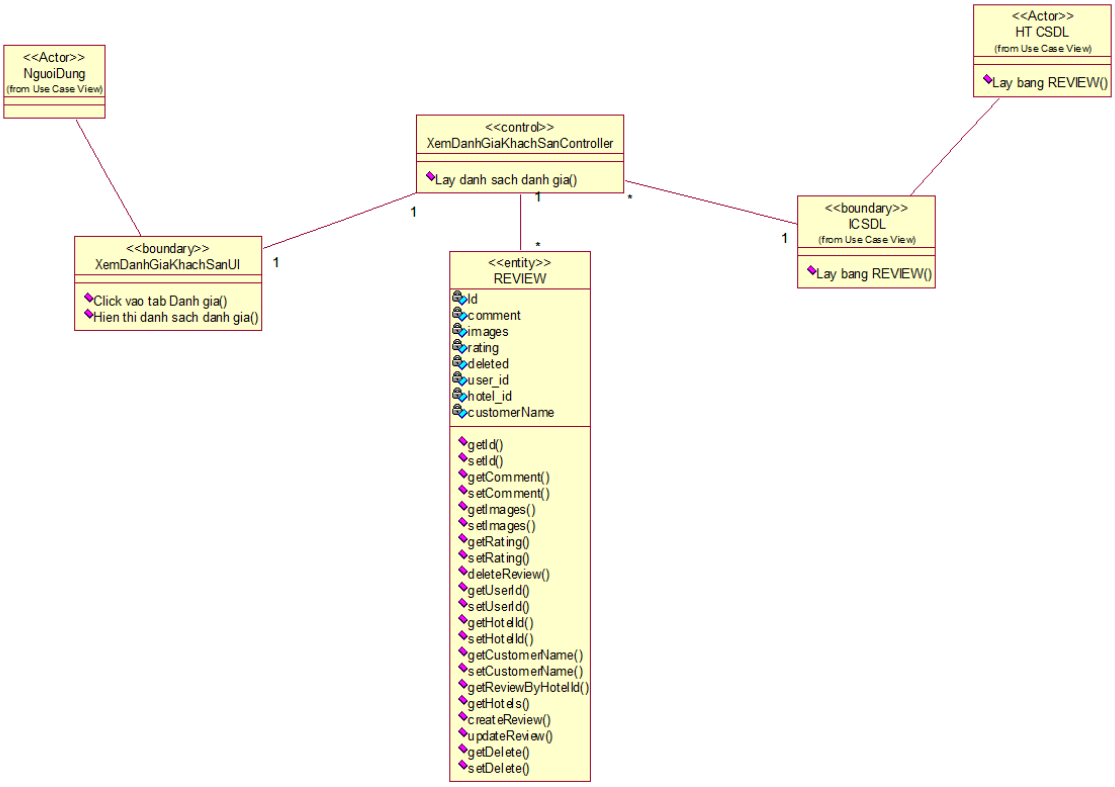
4.5 Use case Xem đánh giá khách sạn

4.5.1 Biểu đồ trình tự



Hình 4.9. Biểu đồ trình tự use case Xem đánh giá khách sạn

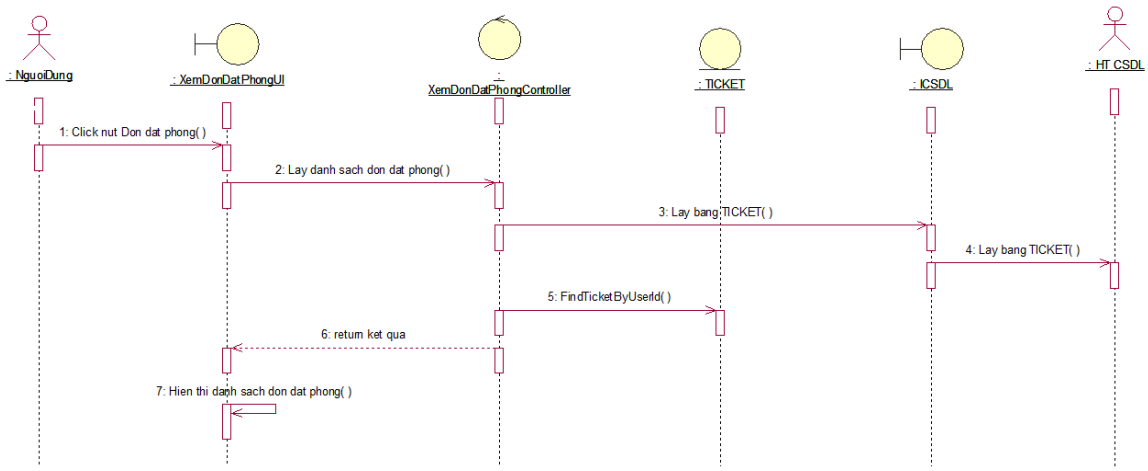
4.5.2 Biểu đồ lớp



Hình 4.10. Biểu đồ lớp use case Xem đánh giá khách sạn

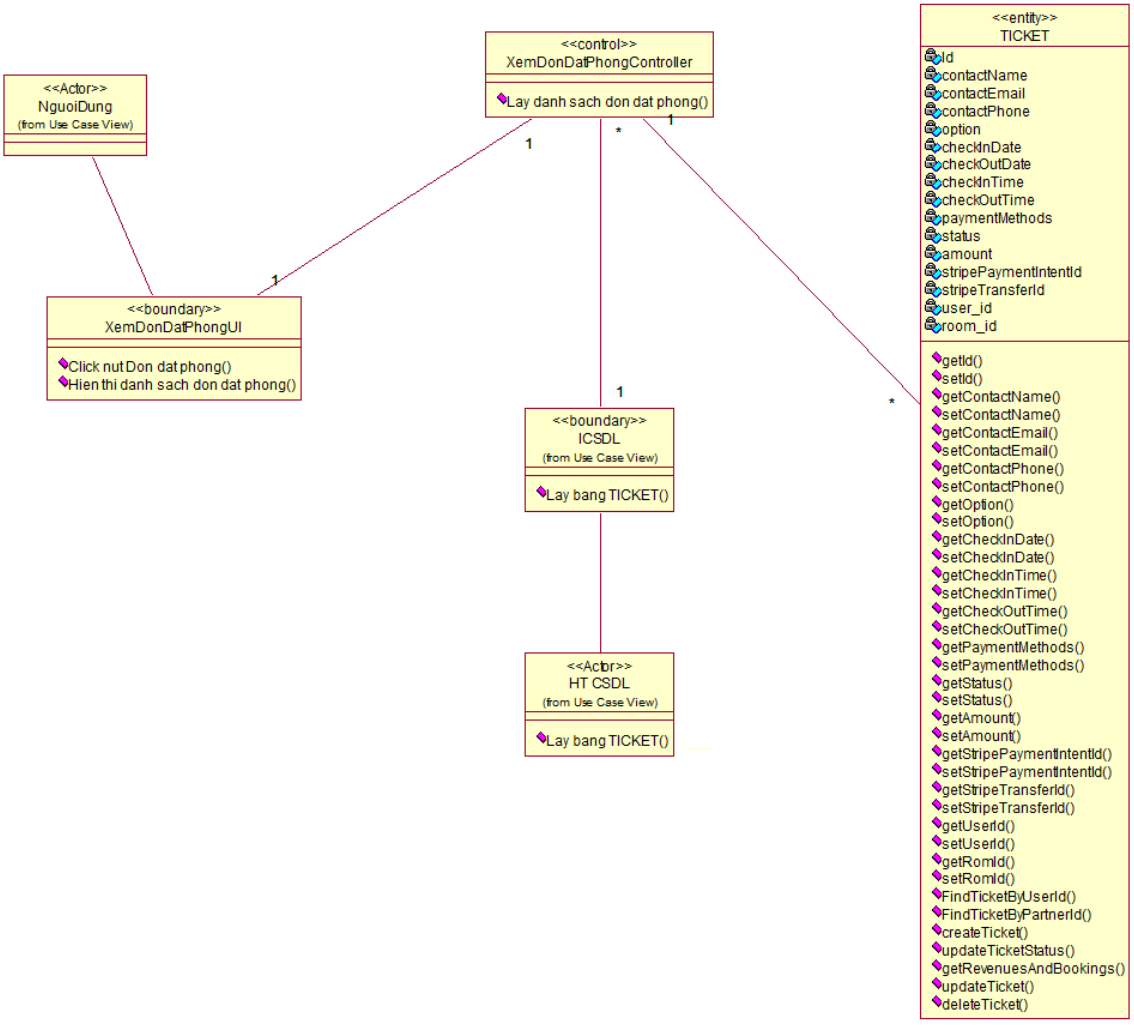
4.6 Use case Xem đơn đặt phòng

4.6.1 Biểu đồ trình tự



Hình 4.11. Biểu đồ trình tự use case Xem đơn đặt phòng

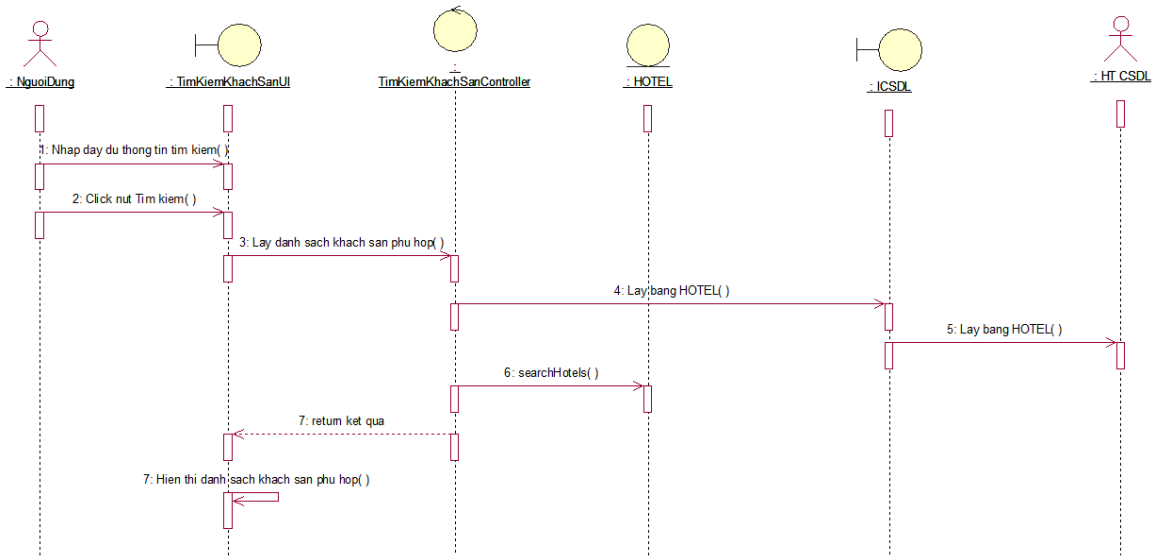
4.6.2 Biểu đồ lớp



Hình 4.12. Biểu đồ lớp use case Xem đơn đặt phòng

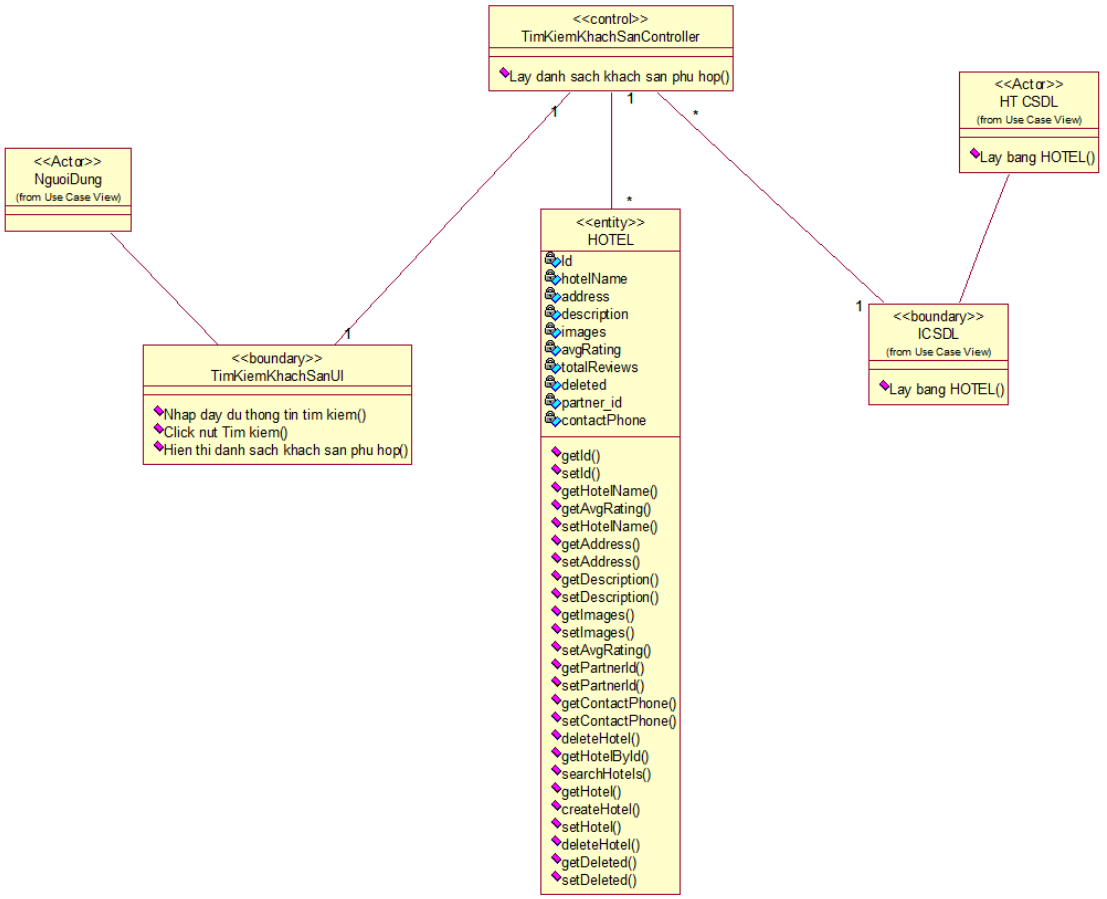
4.7 Use case Tìm kiếm khách sạn

4.7.1 Biểu đồ trình tự



Hình 4.13. Biểu đồ trình tự Tìm kiếm khách sạn

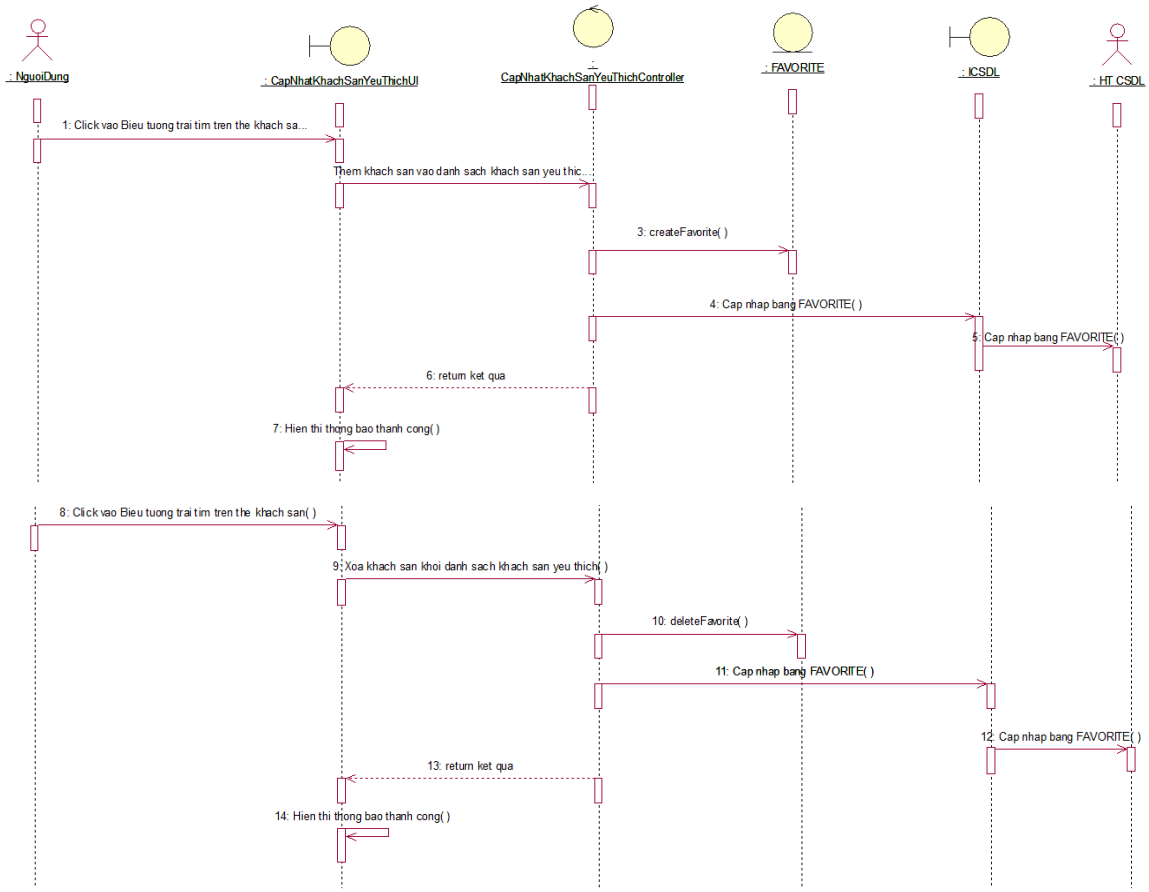
4.7.2 Biểu đồ lớp



Hình 4.14. Biểu đồ lớp use case Tìm kiếm khách sạn

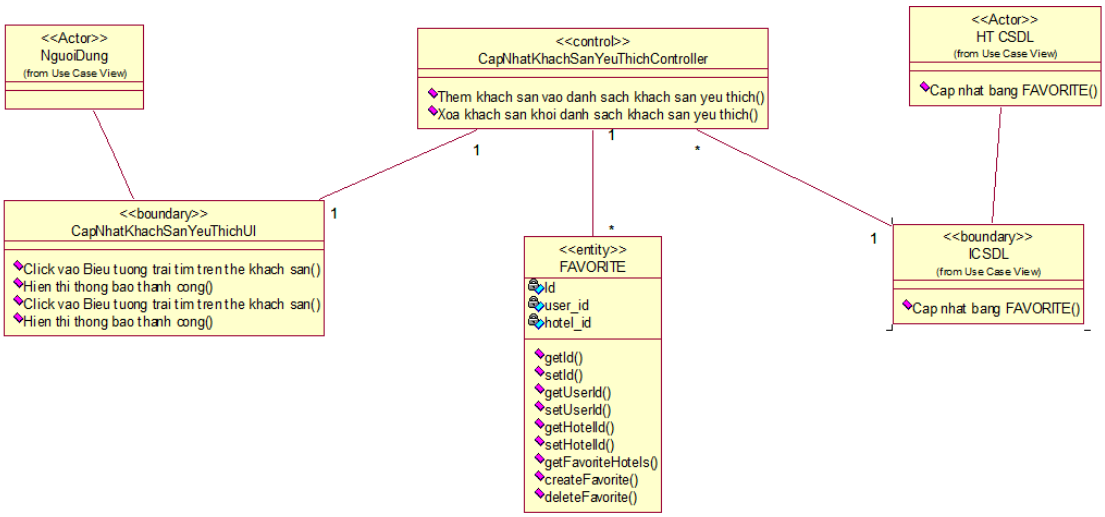
4.8 Use case Cập nhật khách sạn yêu thích

4.8.1 Biểu đồ trình tự



Hình 4.15. Biểu đồ trình tự use case Cập nhật khách sạn yêu thích

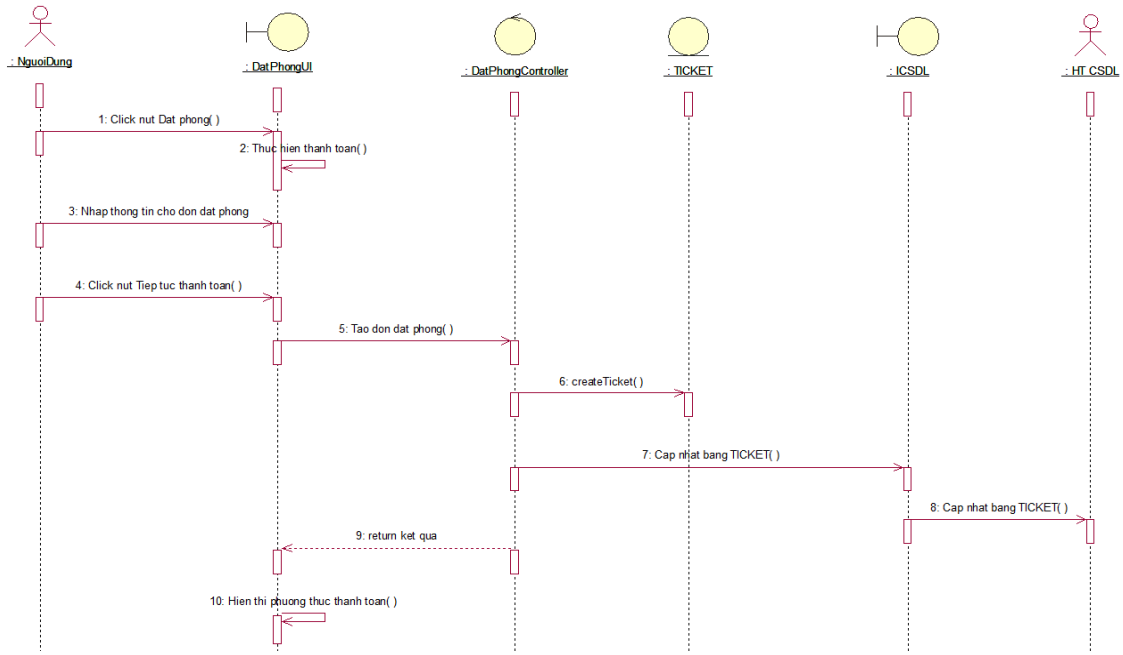
4.8.2 Biểu đồ lớp



Hình 4.16. Biểu đồ lớp use case Cập nhật khách sạn yêu thích

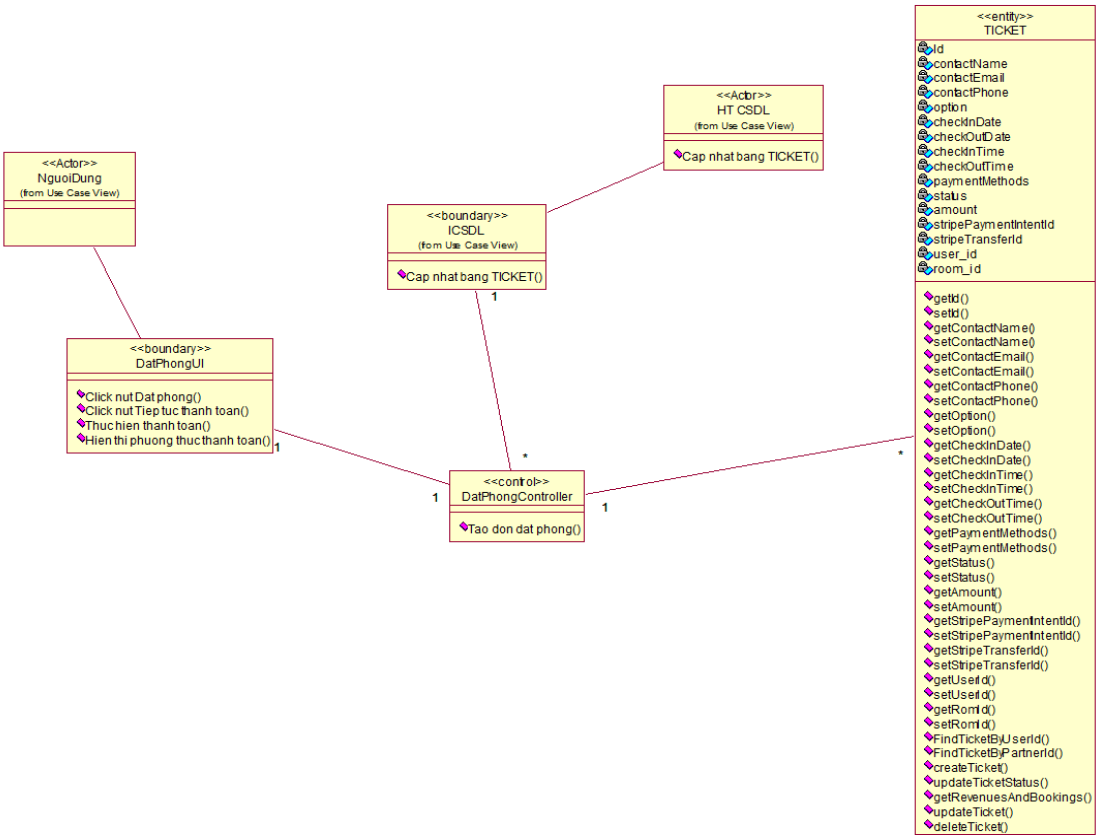
4.9 Use case Đặt phòng

4.9.1 Biểu đồ trình tự



Hình 4.17. Biểu đồ trình tự use case Đặt phòng

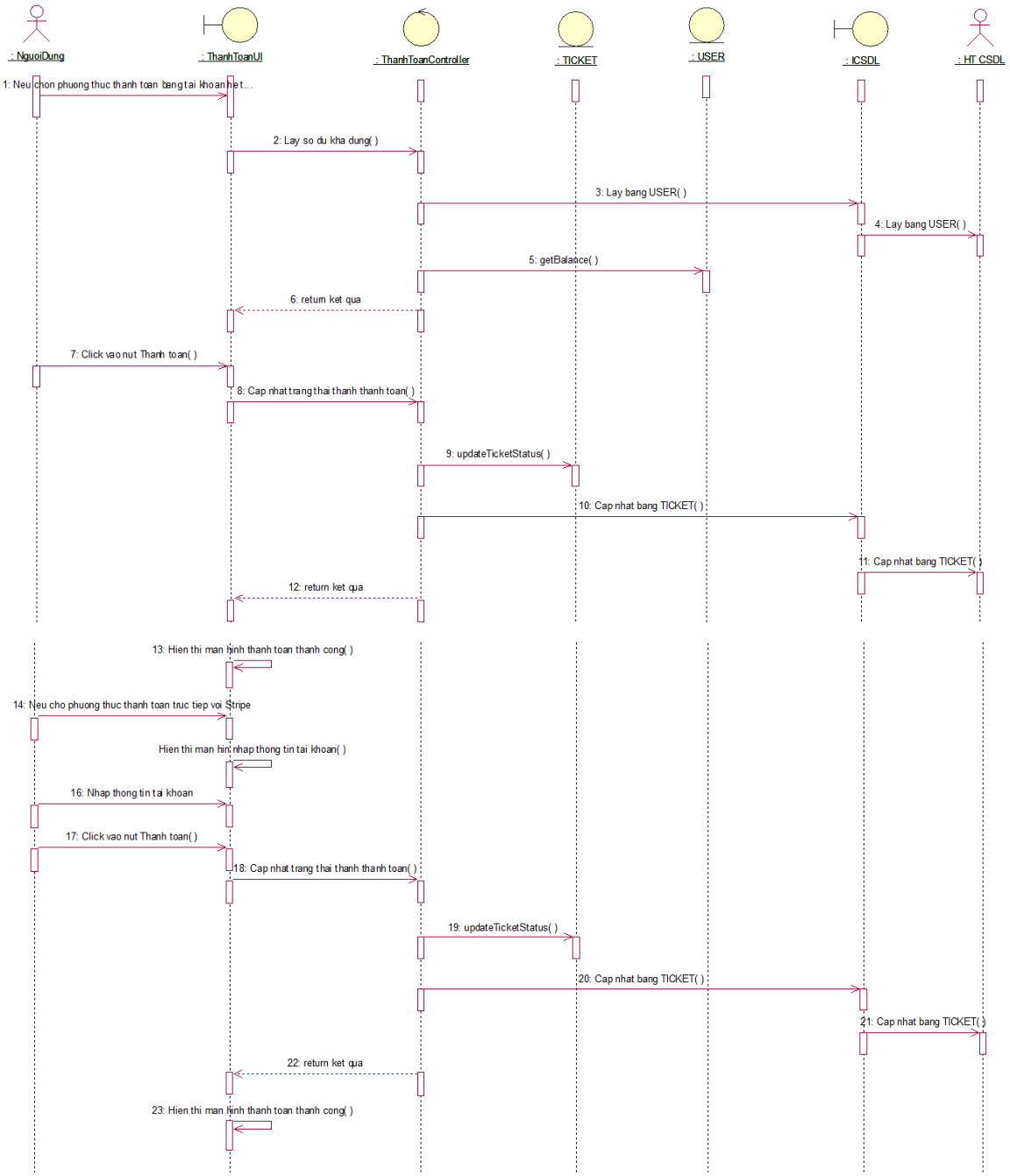
4.9.2 Biểu đồ lớp



Hình 4.18. Biểu đồ lớp use case Đặt phòng

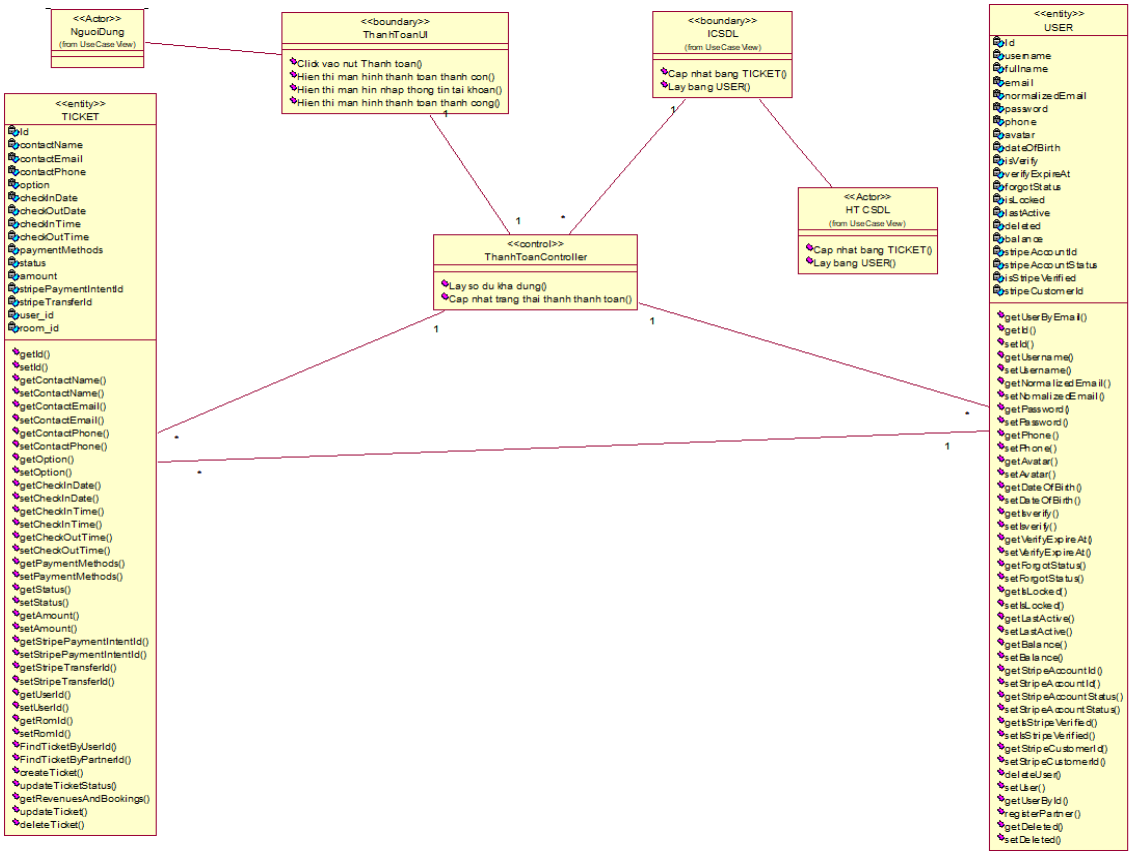
4.10 Use case Thanh toán

4.10.1 Biểu đồ trình tự



Hình 4.19. Biểu đồ trình tự use case Thanh toán

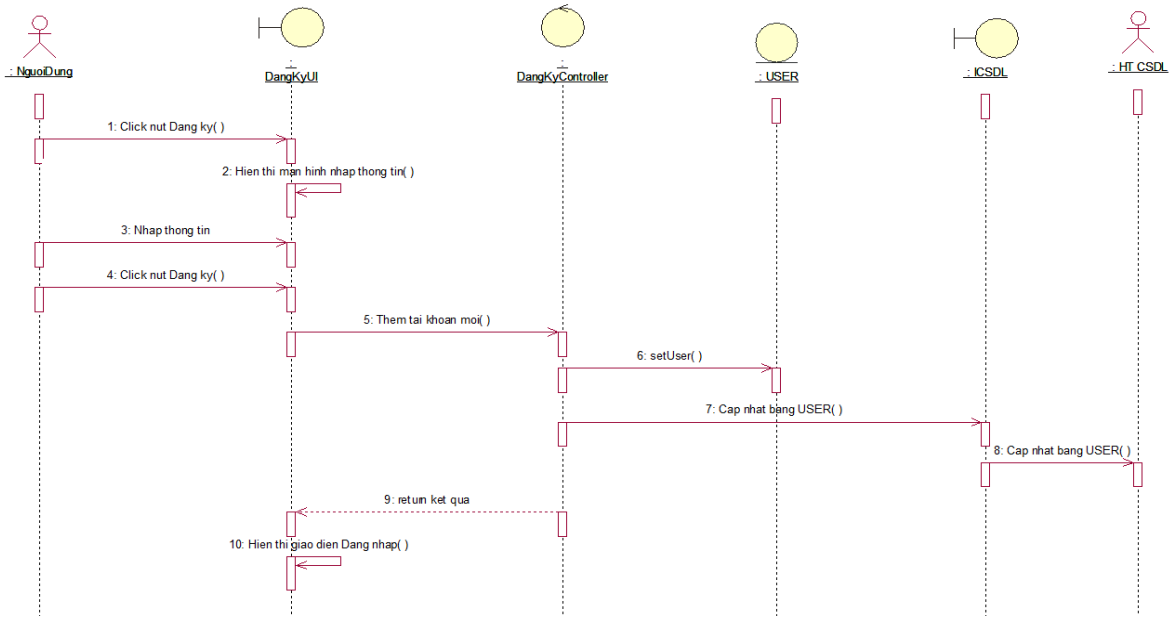
4.10.2 Biểu đồ lớp



Hình 4.20. Biểu đồ lớp use case Thanh toán

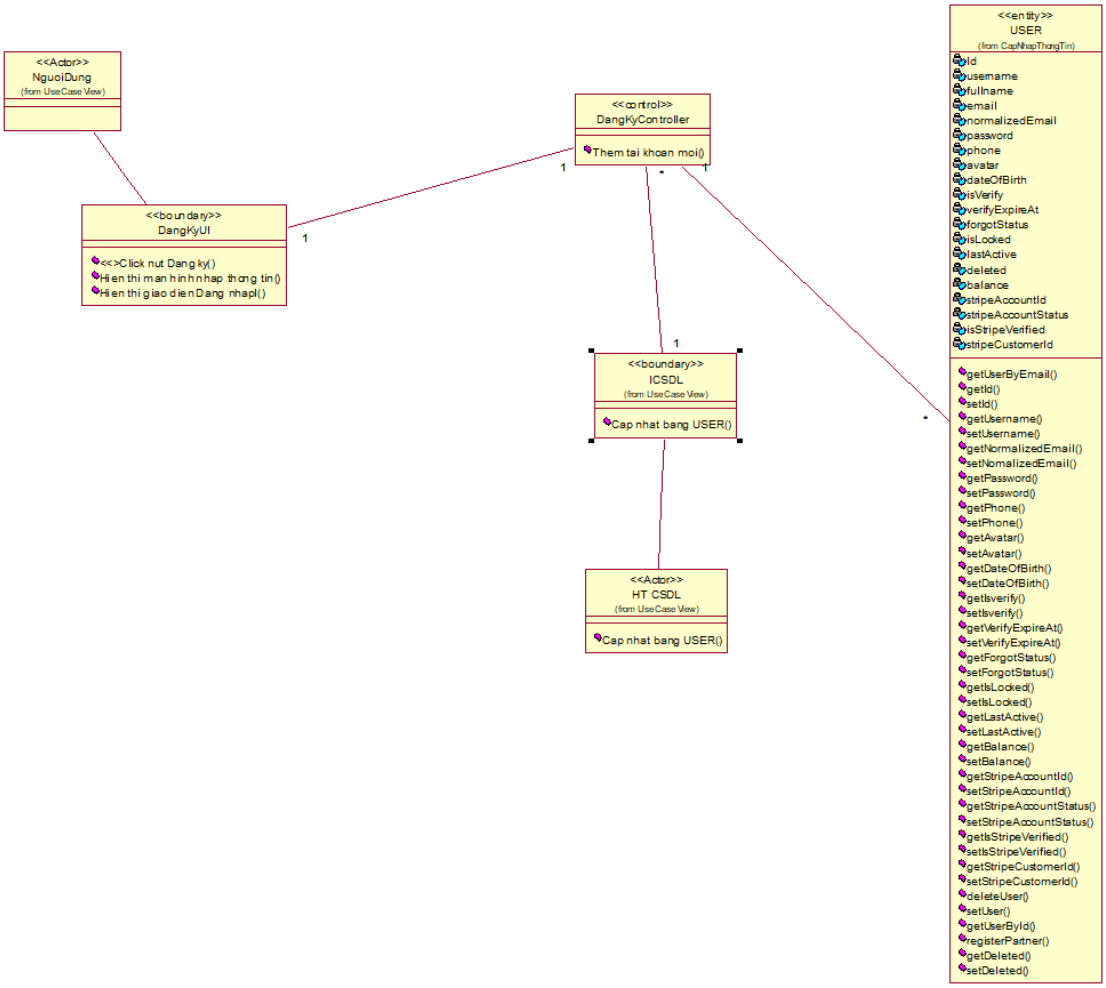
4.11 Use case Đăng ký

4.11.1 Biểu đồ trình tự



Hình 4.21. Biểu đồ trình tự use case Đăng ký

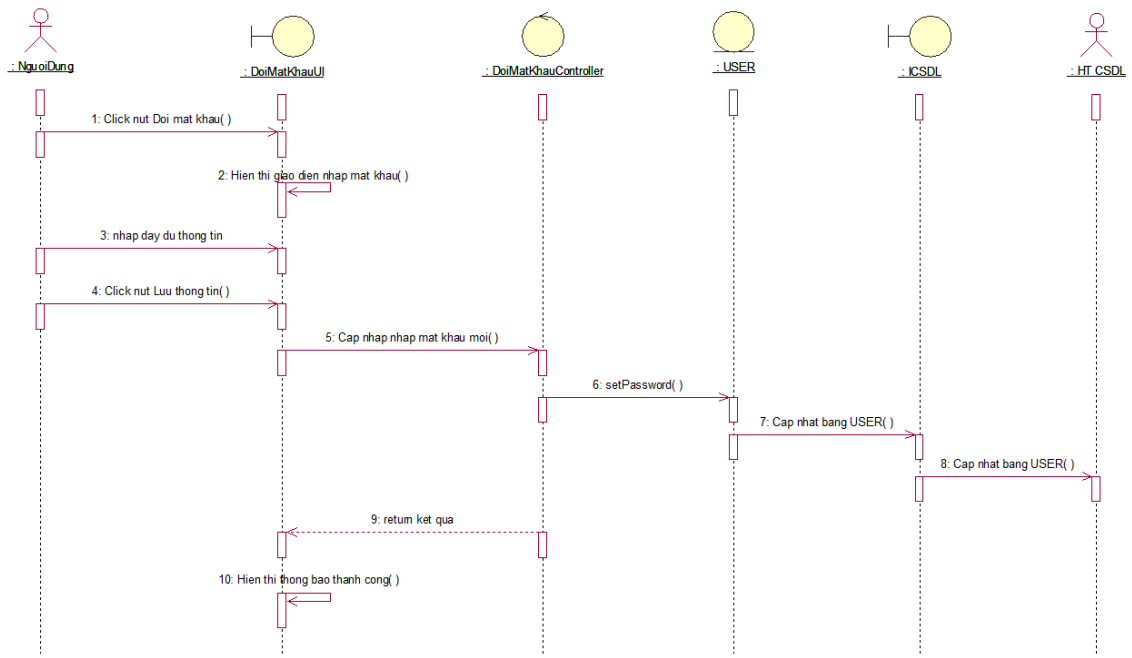
4.11.2 Biểu đồ lớp



Hình 4.22. Biểu đồ lớp use case Đăng ký

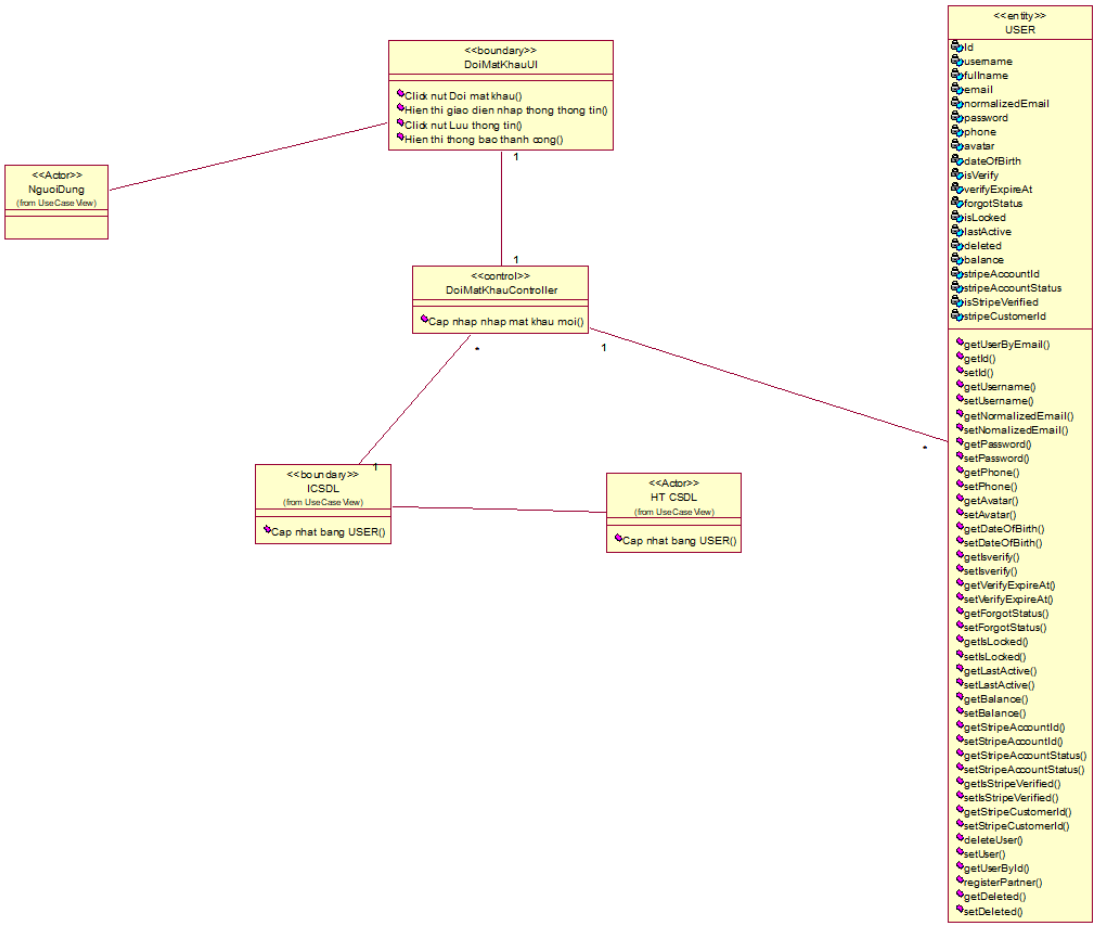
4.12 Use case Đổi mật khẩu

4.12.1 Biểu đồ trình tự



Hình 4.23. Biểu đồ trình tự use case Đổi mật khẩu

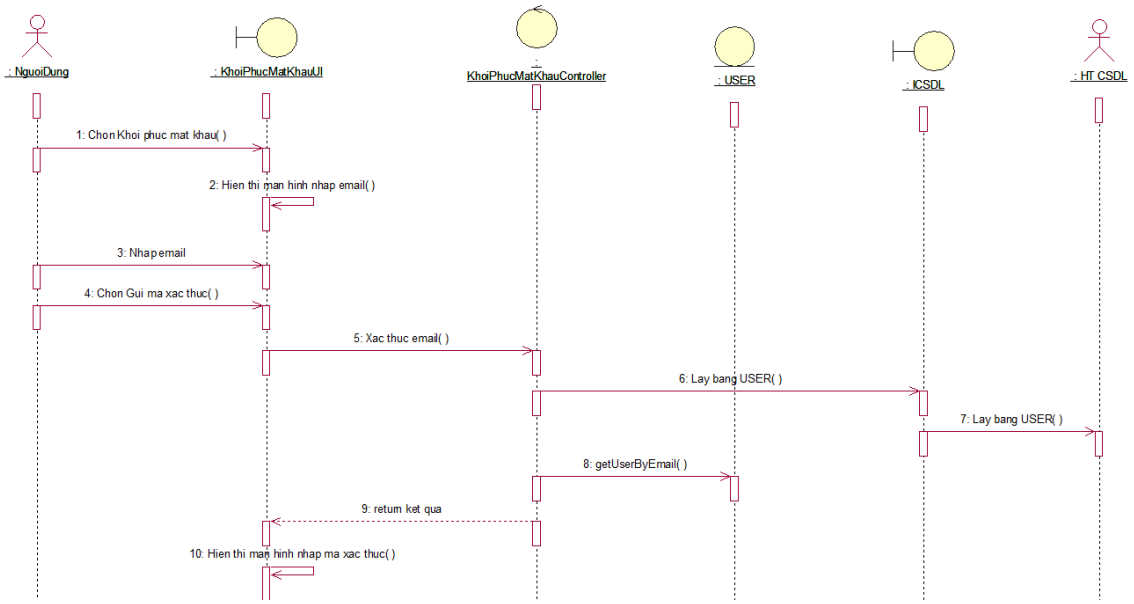
4.12.2 Biểu đồ lớp

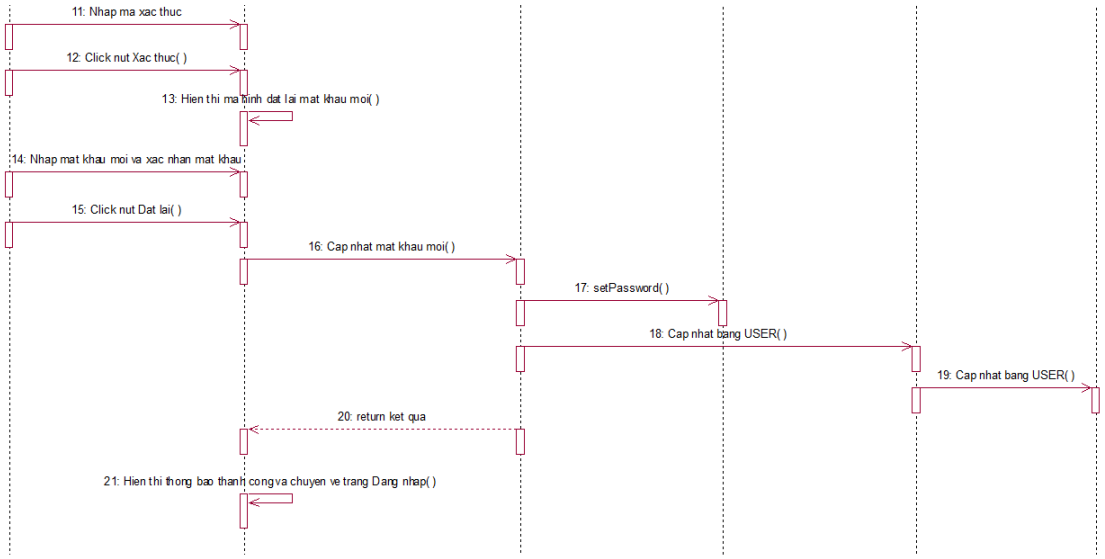


Hình 4.24. Biểu đồ lớp use case Đổi mật khẩu

4.13 Use case Khôi phục mật khẩu

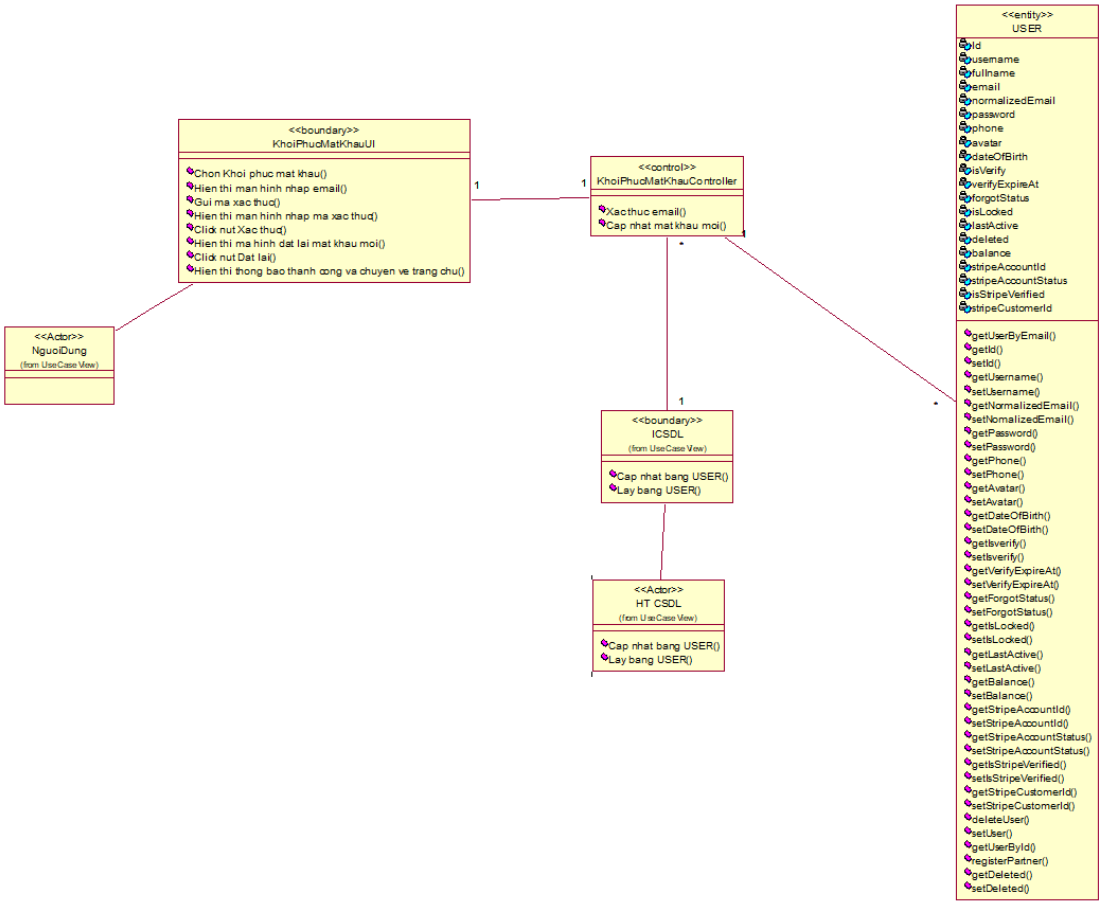
4.13.1 Biểu đồ trình tự





Hình 4.25. Biểu đồ trình tự use case Khôi phục mật khẩu

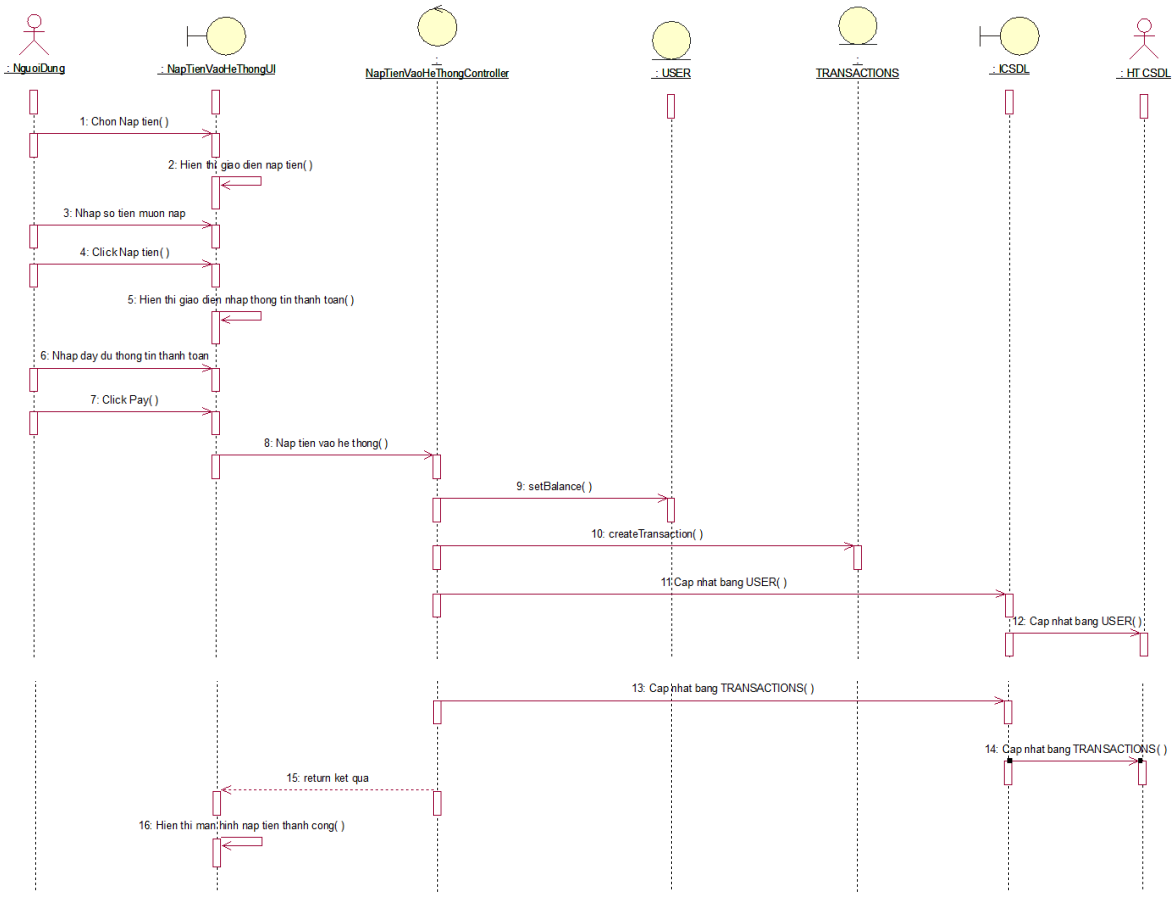
4.13.2 Biểu đồ lớp



Hình 4.26. Biểu đồ lớp use case Khôi phục mật khẩu

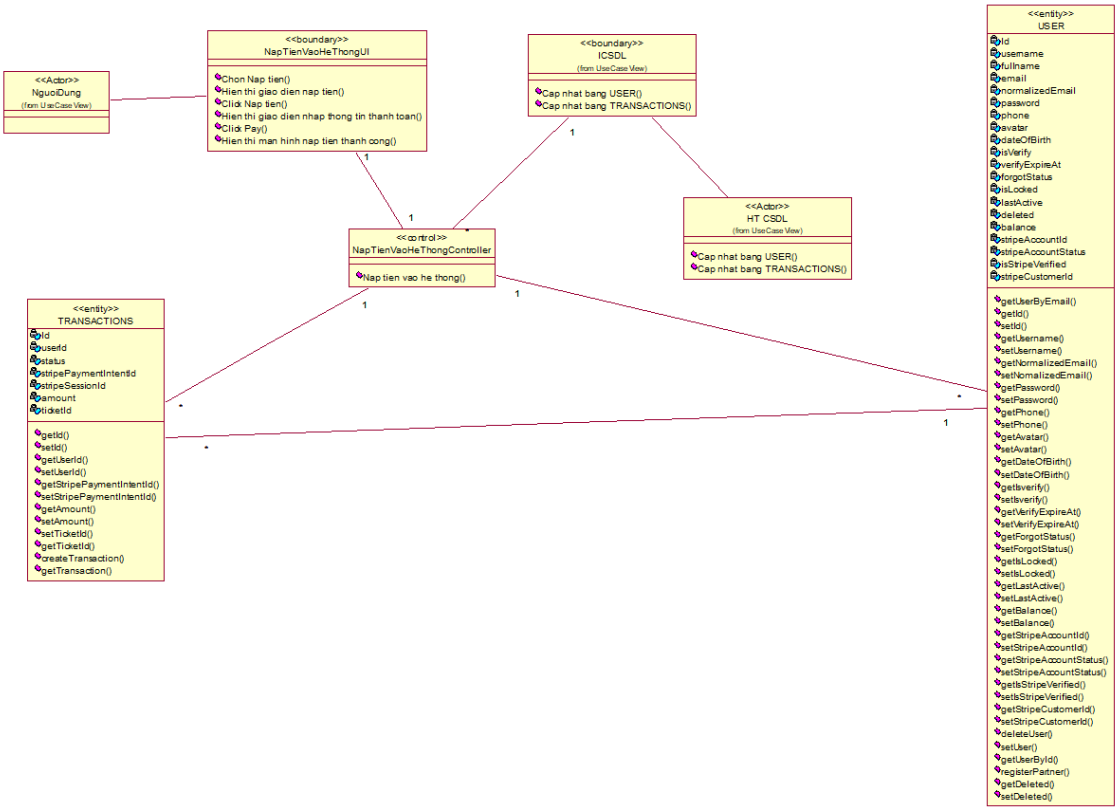
4.14 Use case Nạp tiền vào hệ thống

4.14.1 Biểu đồ trình tự



Hình 4.27. Biểu đồ trình tự use case Nạp tiền vào hệ thống

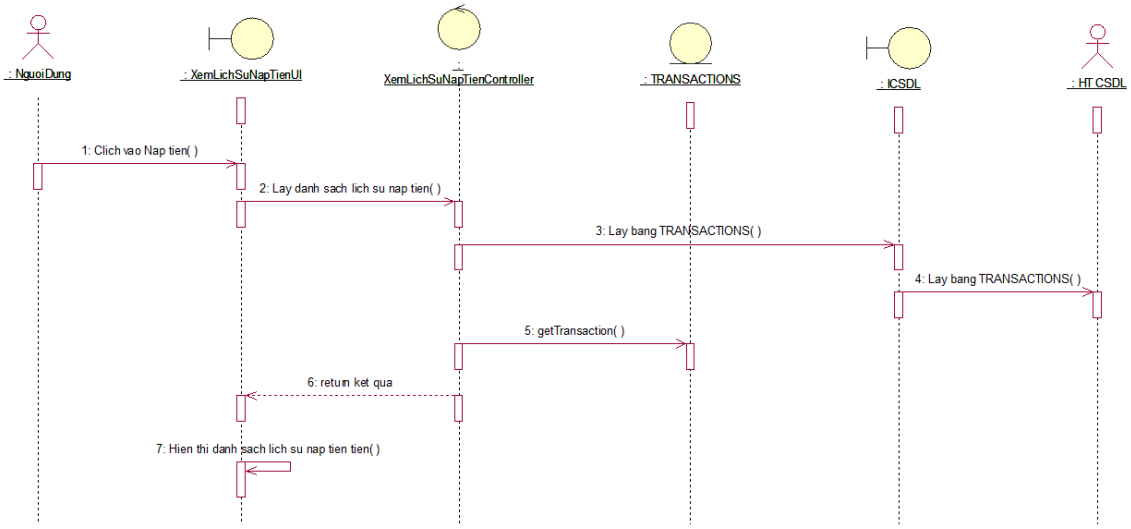
4.14.2 Biểu đồ lớp



Hình 4.28. Biểu đồ lớp use case Nạp tiền vào hệ thống

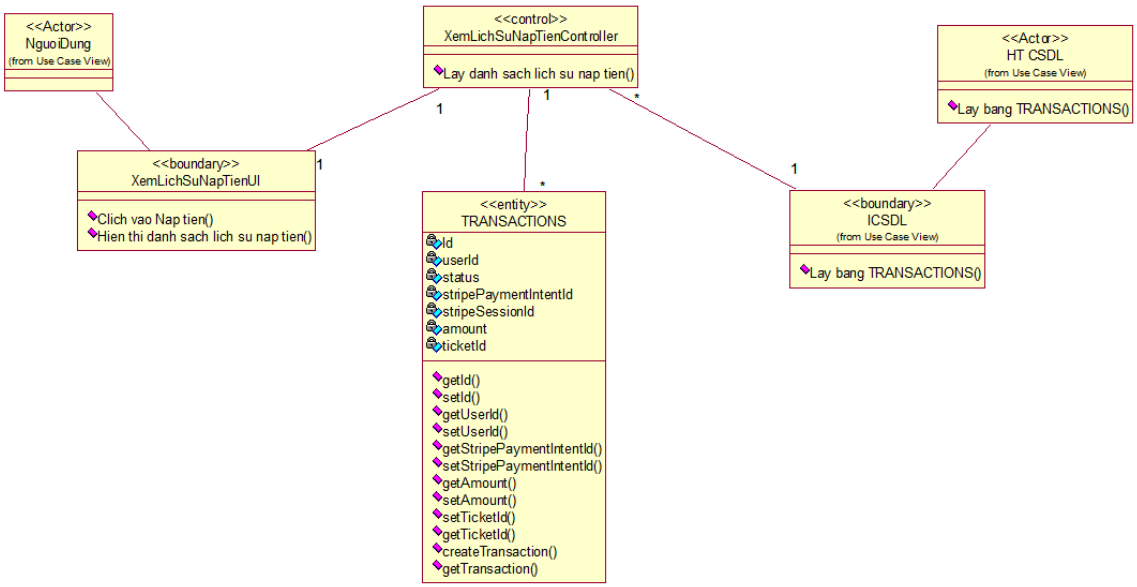
4.15 Use case Xem lịch sử nạp tiền

4.15.1 Biểu đồ trình tự



Hình 4.29. Biểu đồ trình tự use case Xem lịch sử nạp tiền

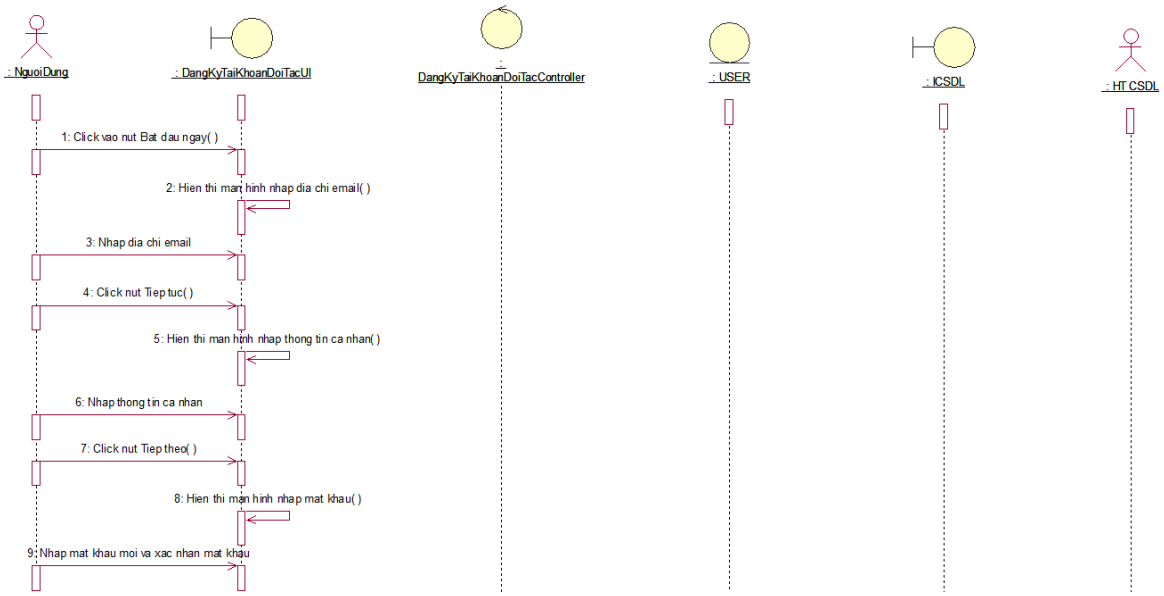
4.15.2 Biểu đồ lớp

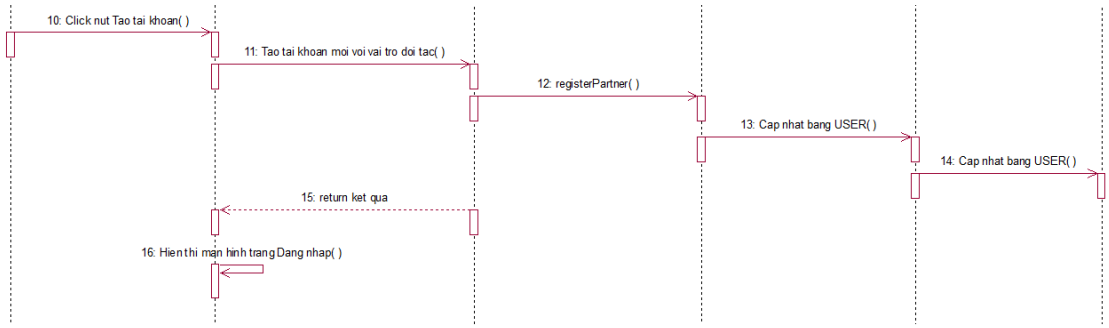


Hình 4.30. Biểu đồ lớp use case Xem lịch sử nạp tiền

4.16 Use case Đăng ký tài khoản đối tác

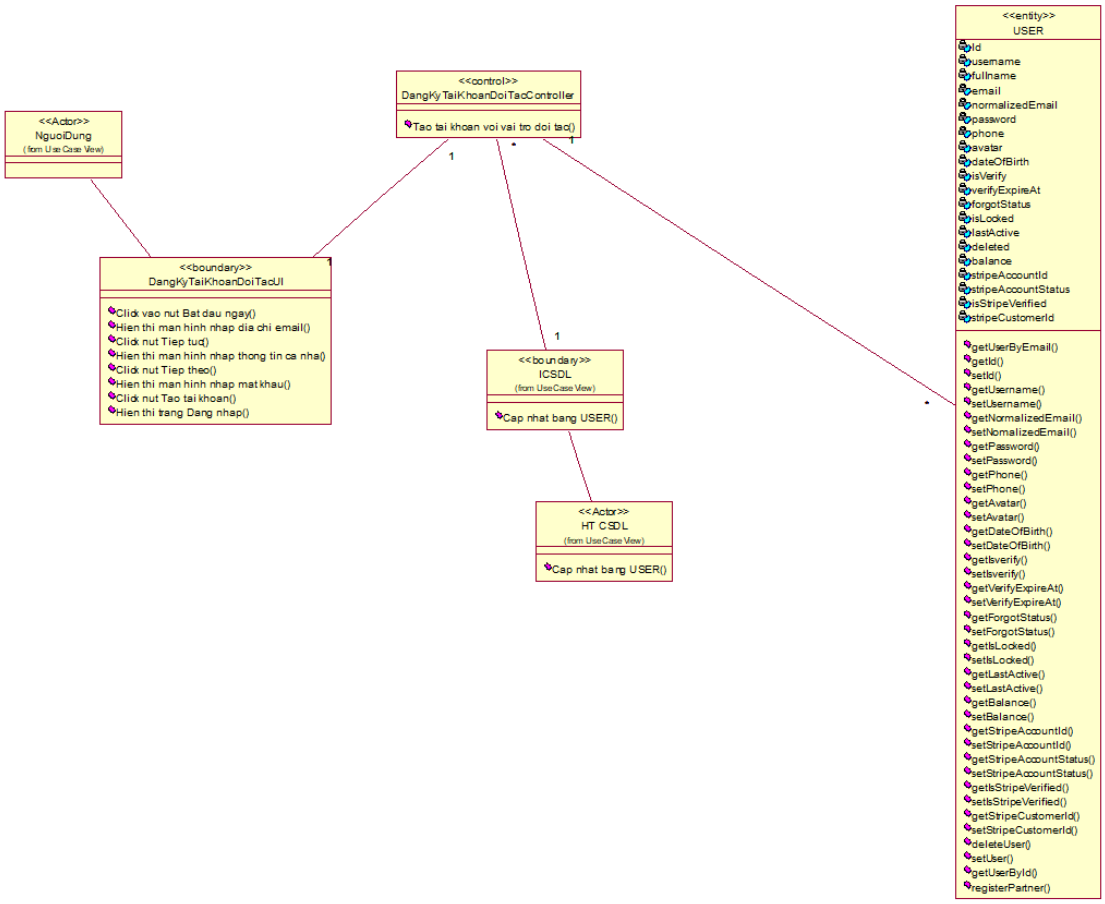
4.16.1 Biểu đồ trình tự





Hình 4.31. Biểu đồ trình tự use case Đăng ký tài khoản đối tác

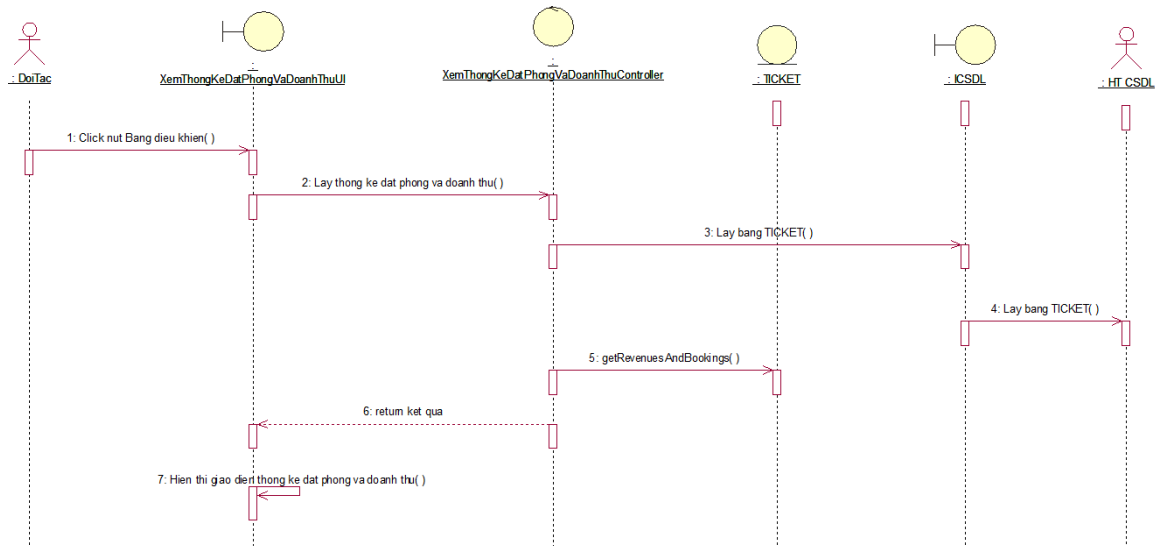
4.16.2 Biểu đồ lớp



Hình 4.32. Biểu đồ lớp use case Đăng ký tài khoản đối tác

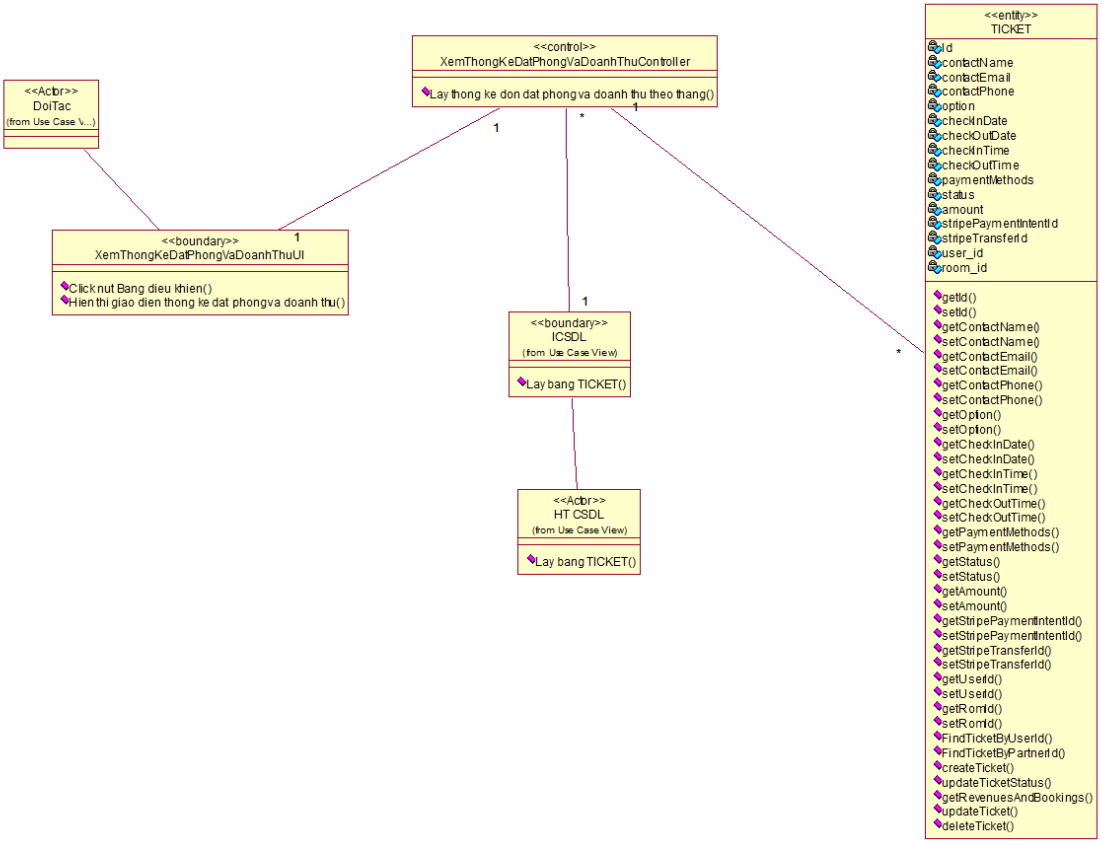
4.17 Use case Xem thống kê đặt phòng và doanh thu

4.17.1 Biểu đồ trình tự



Hình 4.33. Biểu đồ trình tự use case Xem thống kê đặt phòng và doanh thu

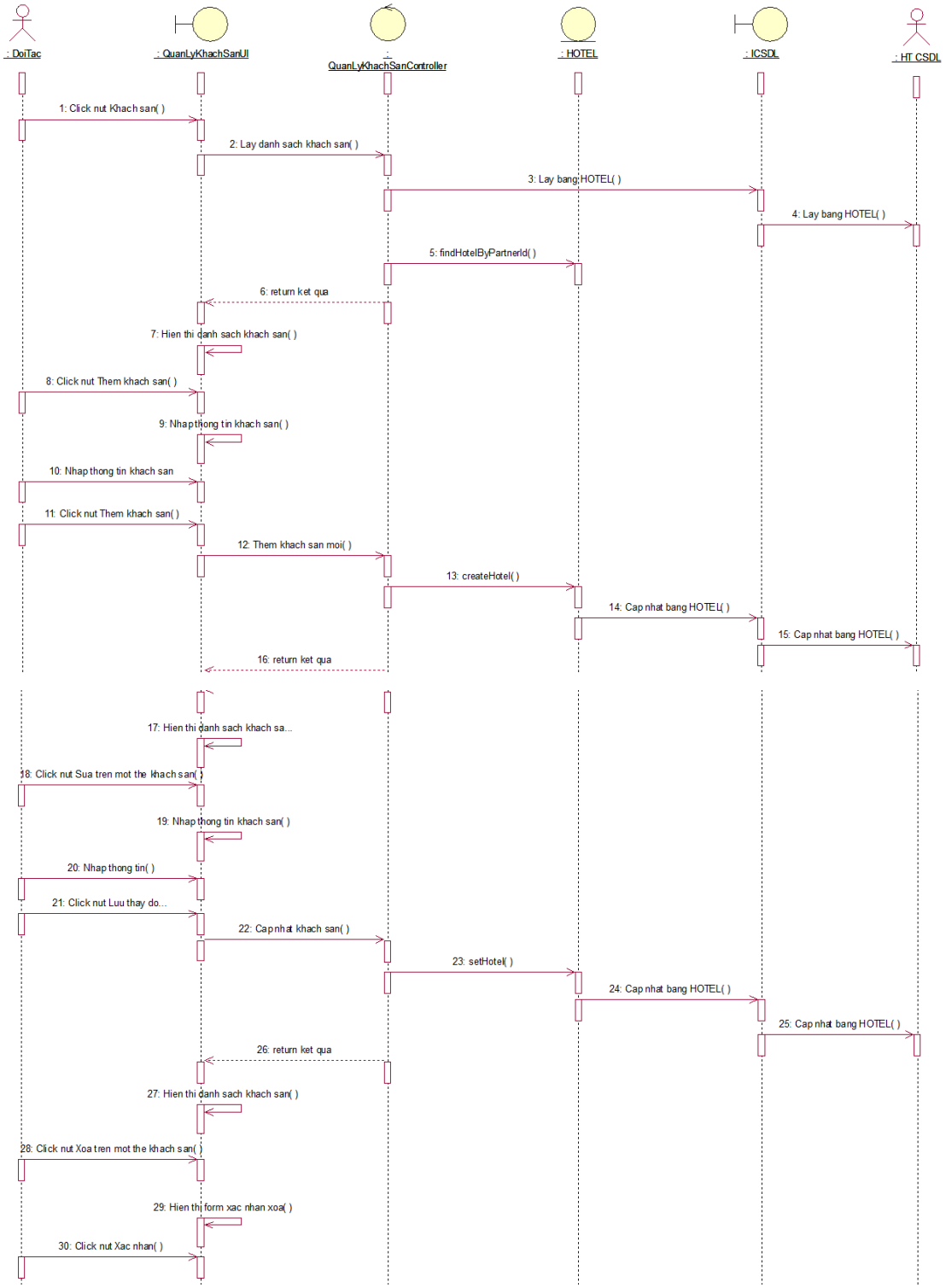
4.17.2 Biểu đồ lớp

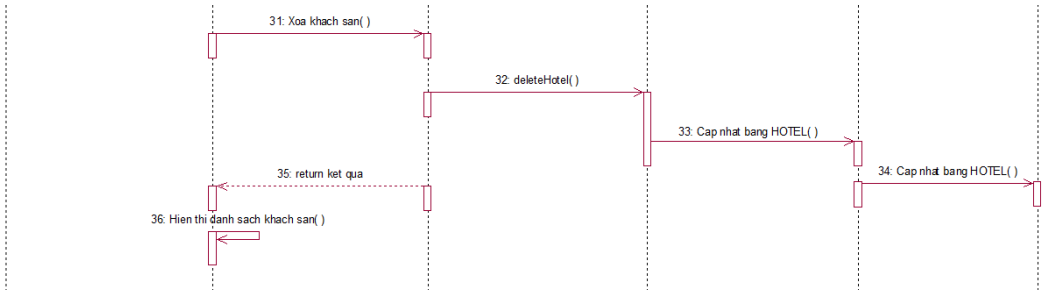


Hình 4.34. Biểu đồ lớp use case Xem thống kê đặt phòng và doanh thu

4.18 Use case Quản lý khách sạn

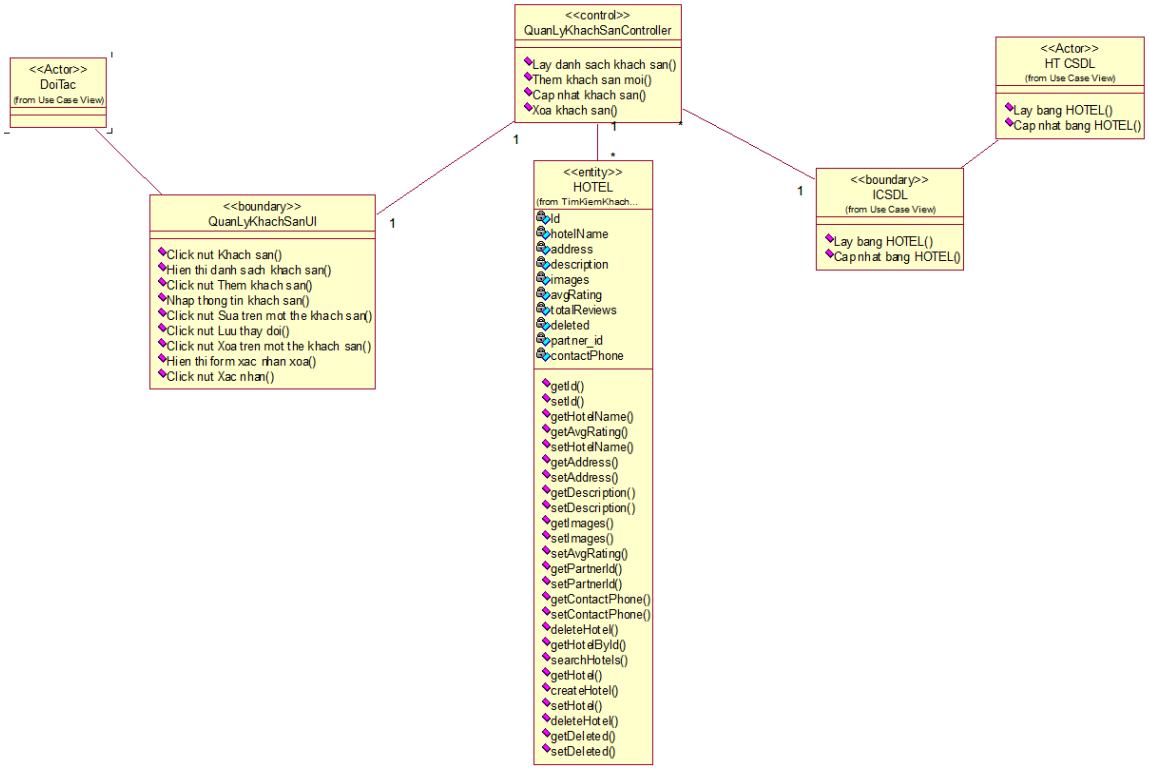
4.18.1 Biểu đồ trình tự





Hình 4.35. Biểu đồ trình tự Quản lý khách sạn

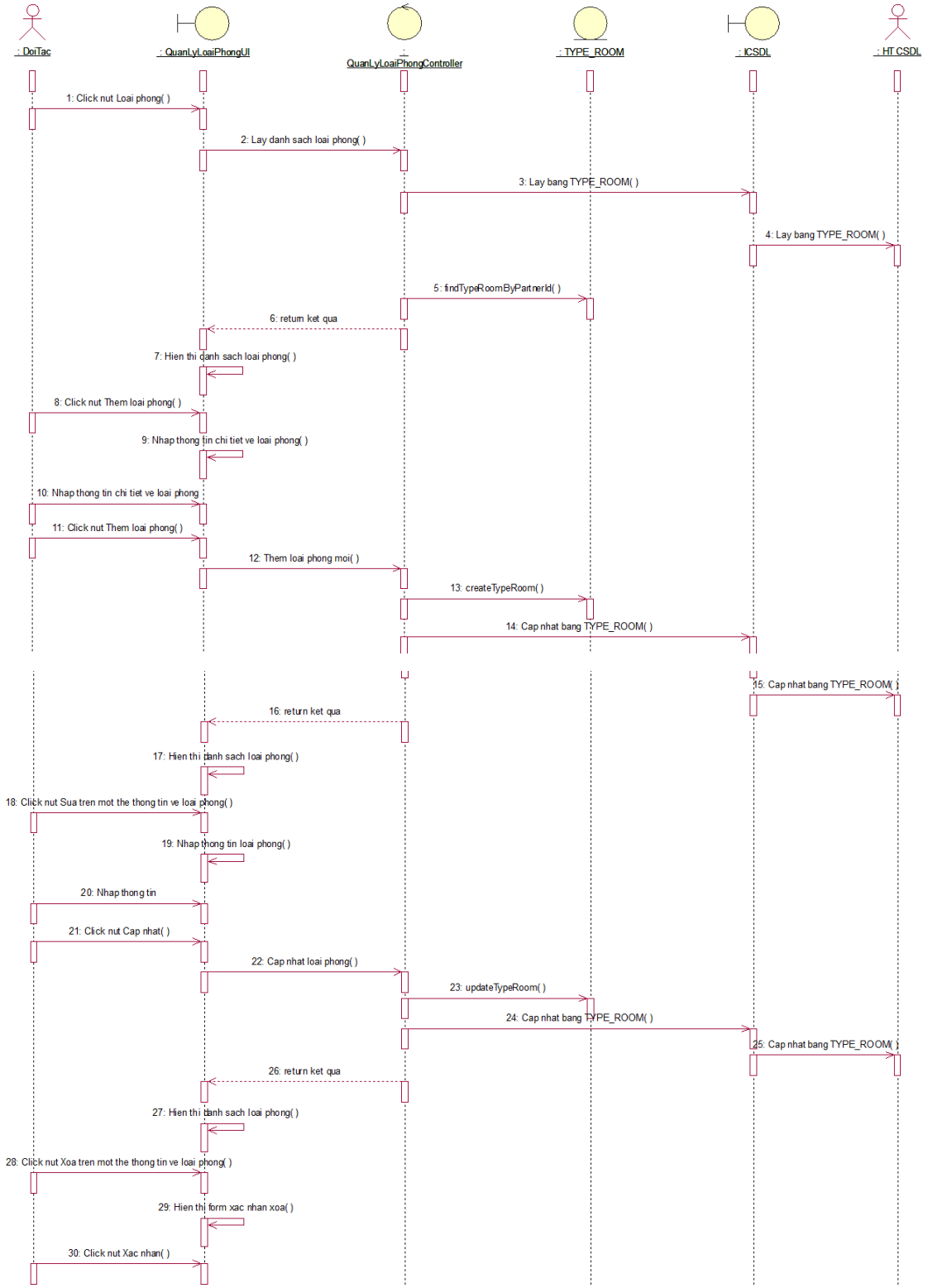
4.18.2 Biểu đồ lớp

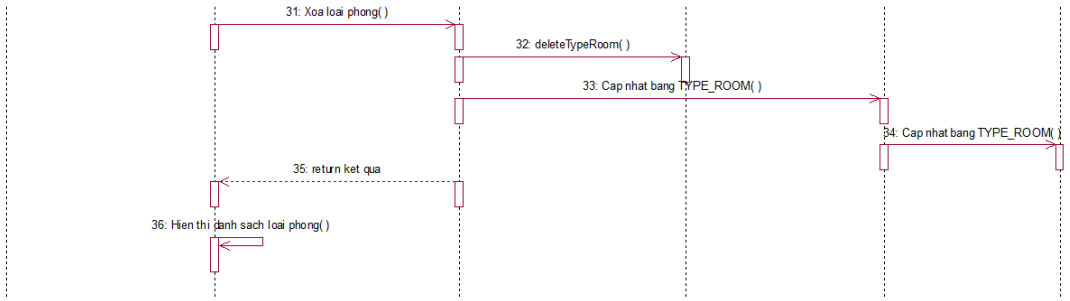


Hình 4.36. Biểu đồ lớp use case Quản lý khách sạn

4.19 Use case Quản lý loại phòng

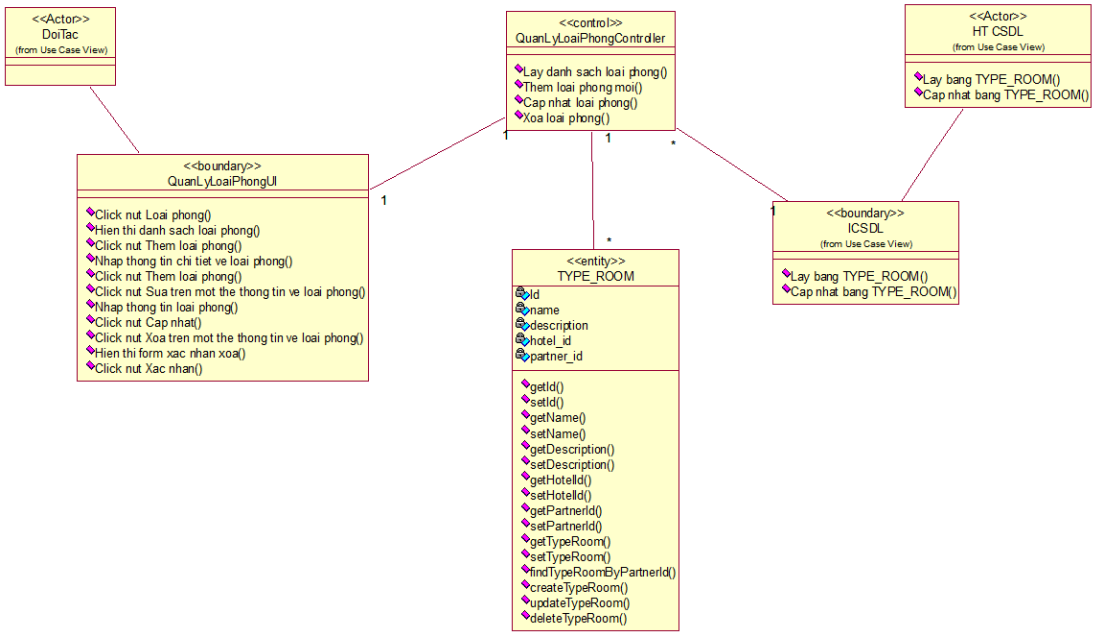
4.19.1 Biểu đồ trình tự





Hình 4.37. Biểu đồ trình tự Quản lý loại phòng

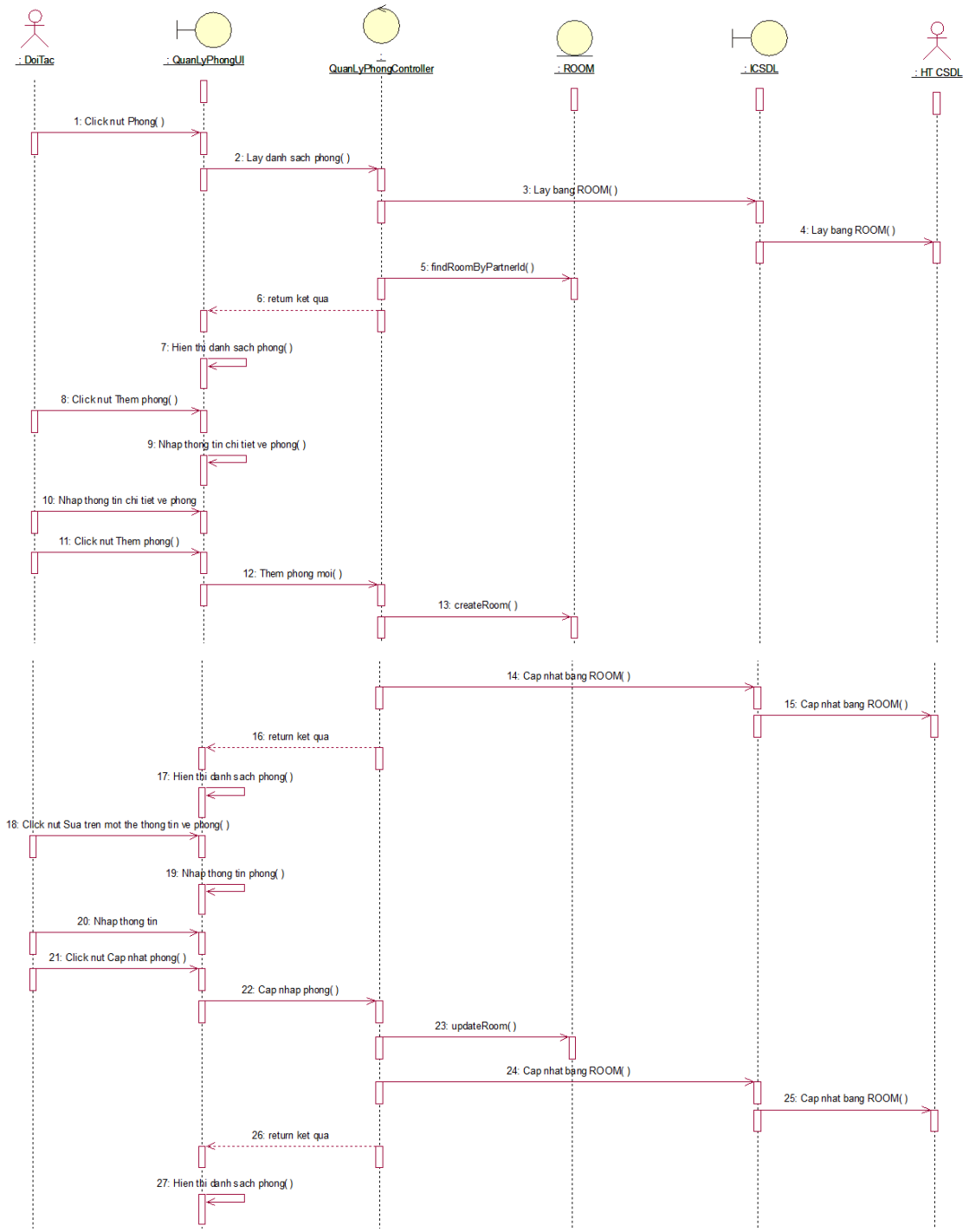
4.19.2 Biểu đồ lớp

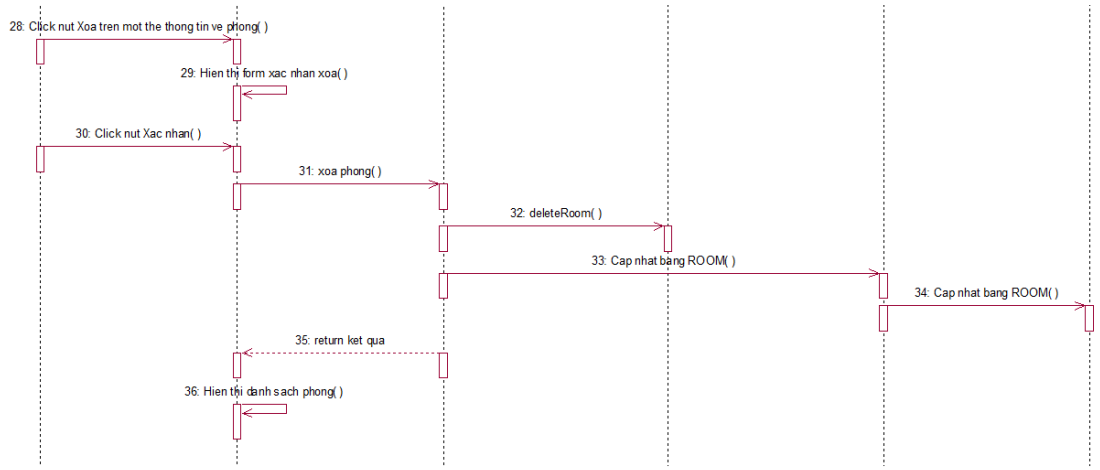


Hình 4.38. Biểu đồ lớp use case Quản lý loại phòng

4.20 Use case Quản lý phòng

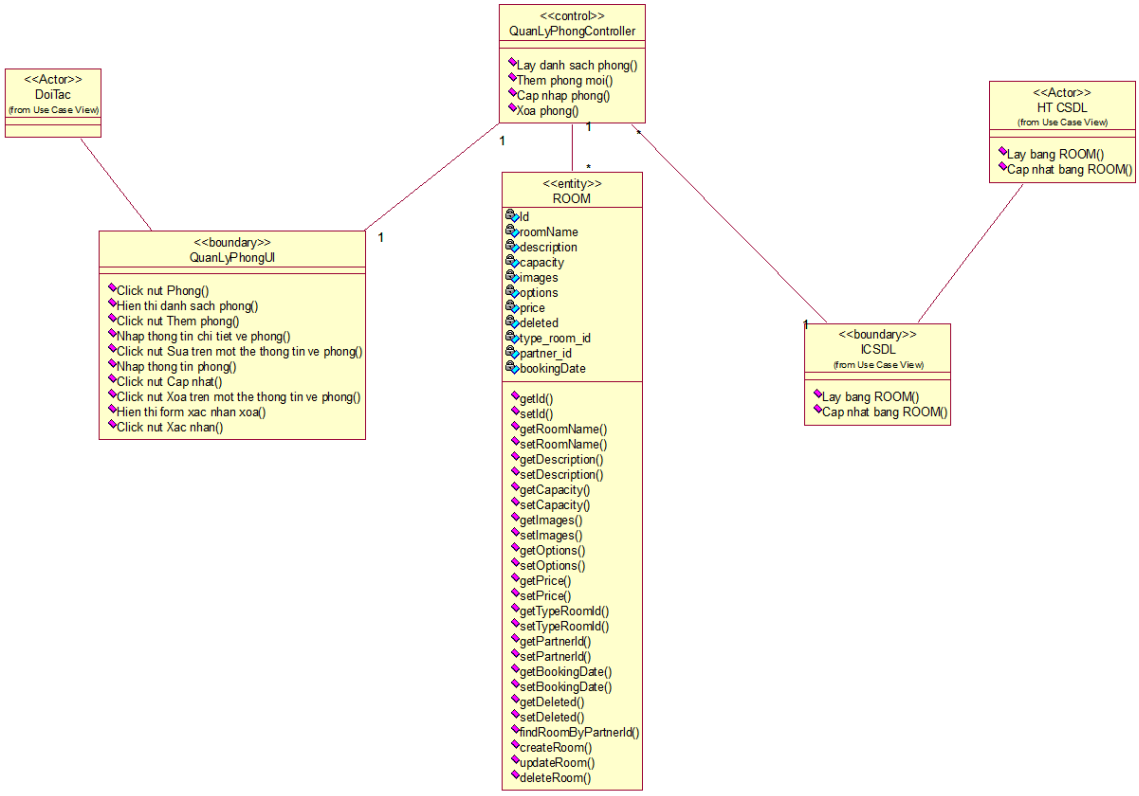
4.20.1 Biểu đồ trình tự





Hình 4.39. Biểu đồ trình tự Quản lý phòng

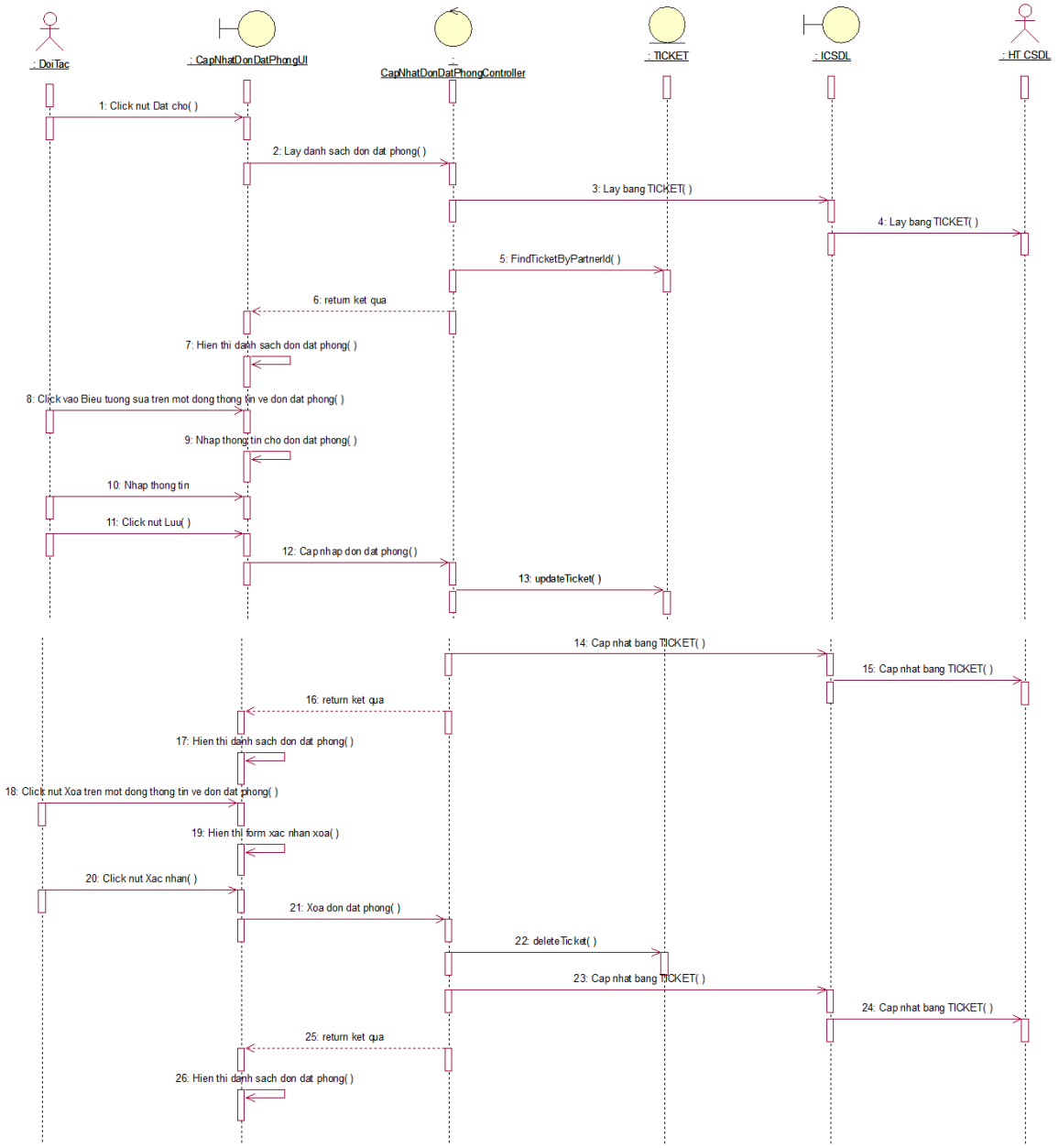
4.20.2 Biểu đồ lớp



Hình 4.40. Biểu đồ lớp use case Quản lý phòng

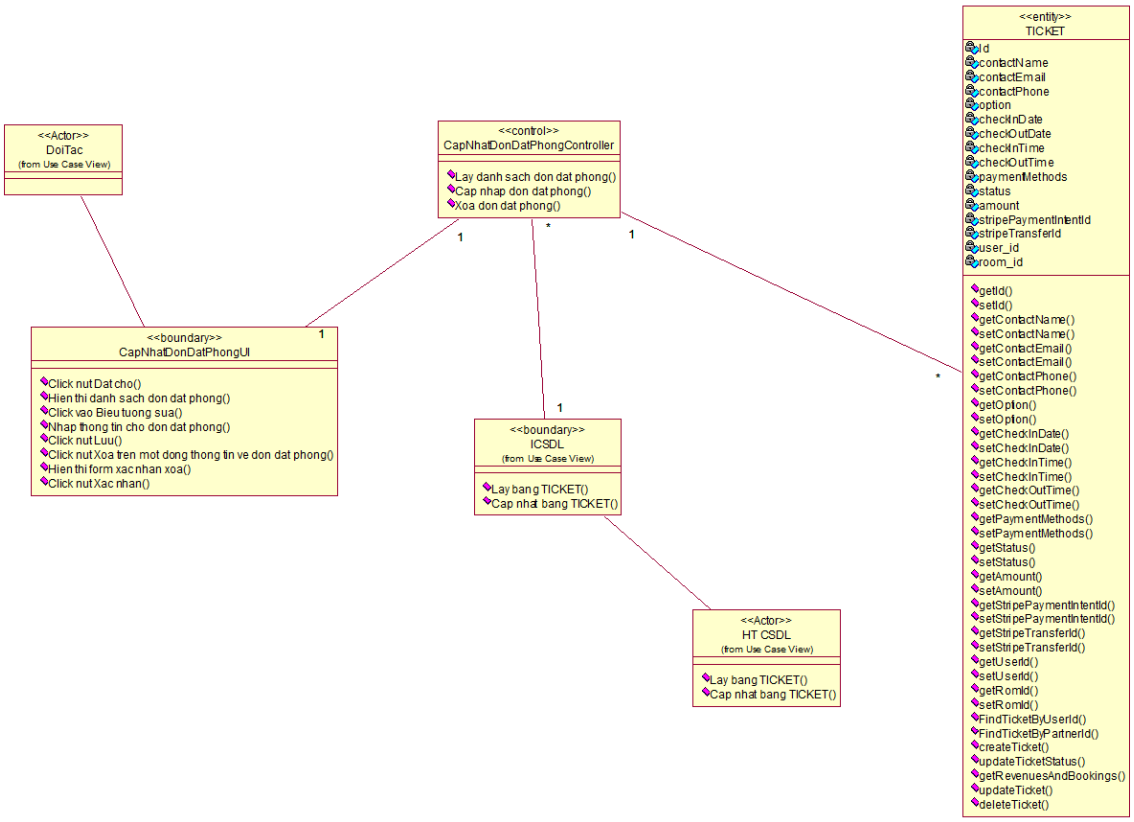
4.21 Use case Cập nhật đơn đặt phòng

4.21.1 Biểu đồ trình tự



Hình 4.41. Biểu đồ trình tự Cập nhật đơn đặt phòng

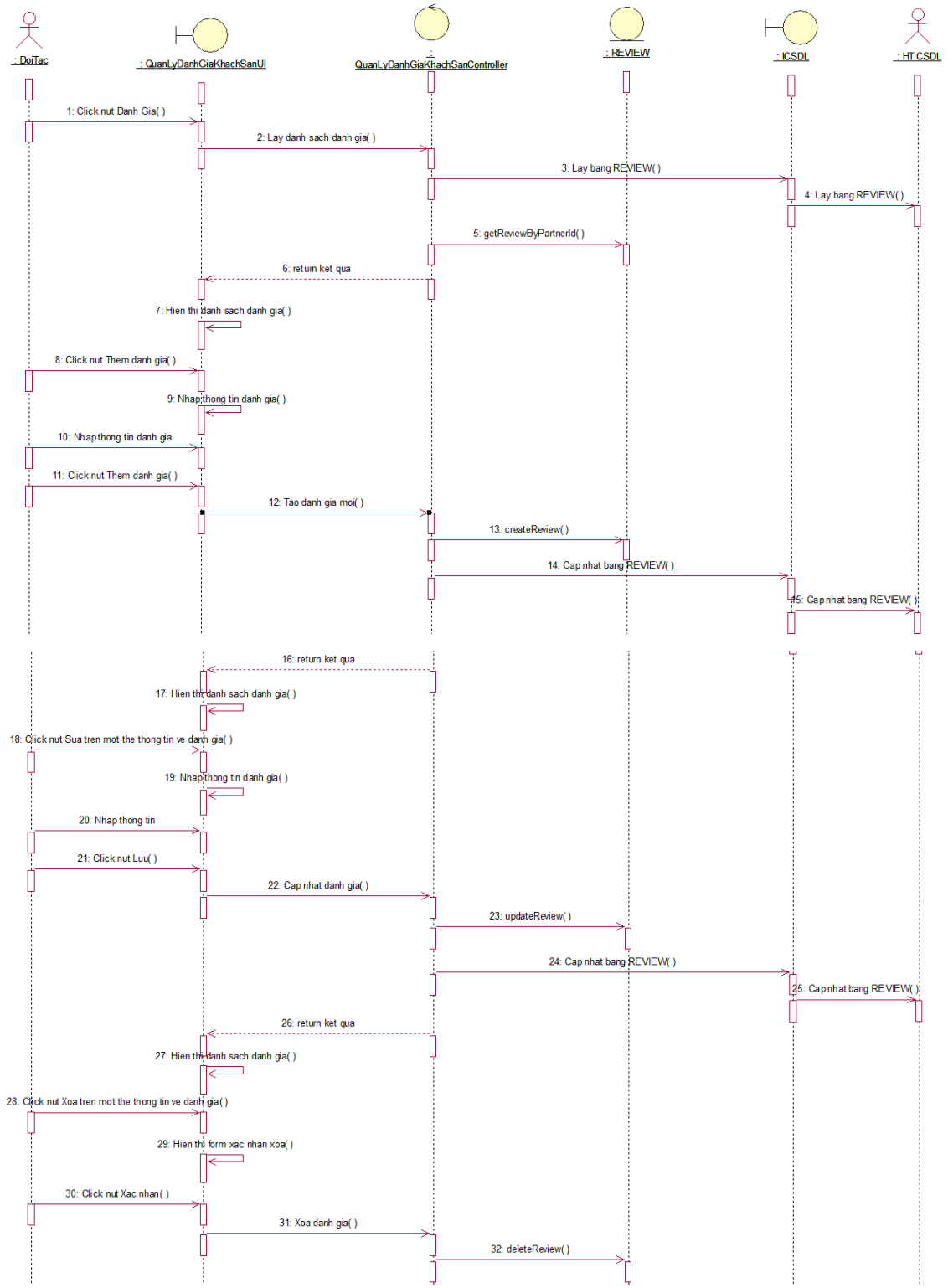
4.21.2 Biểu đồ lớp

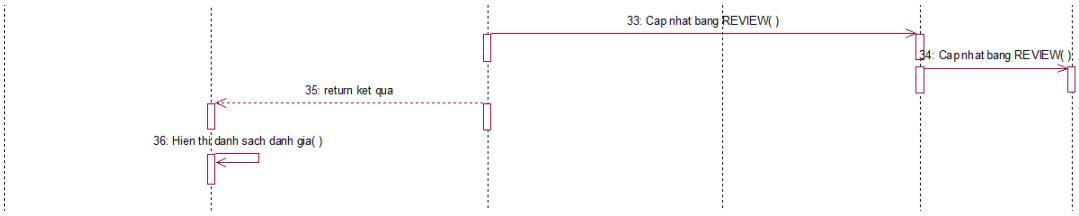


Hình 4.42. Biểu đồ lớp use case Cập nhật đơn đặt phòng

4.22 Use case Quản lý đánh giá khách sạn

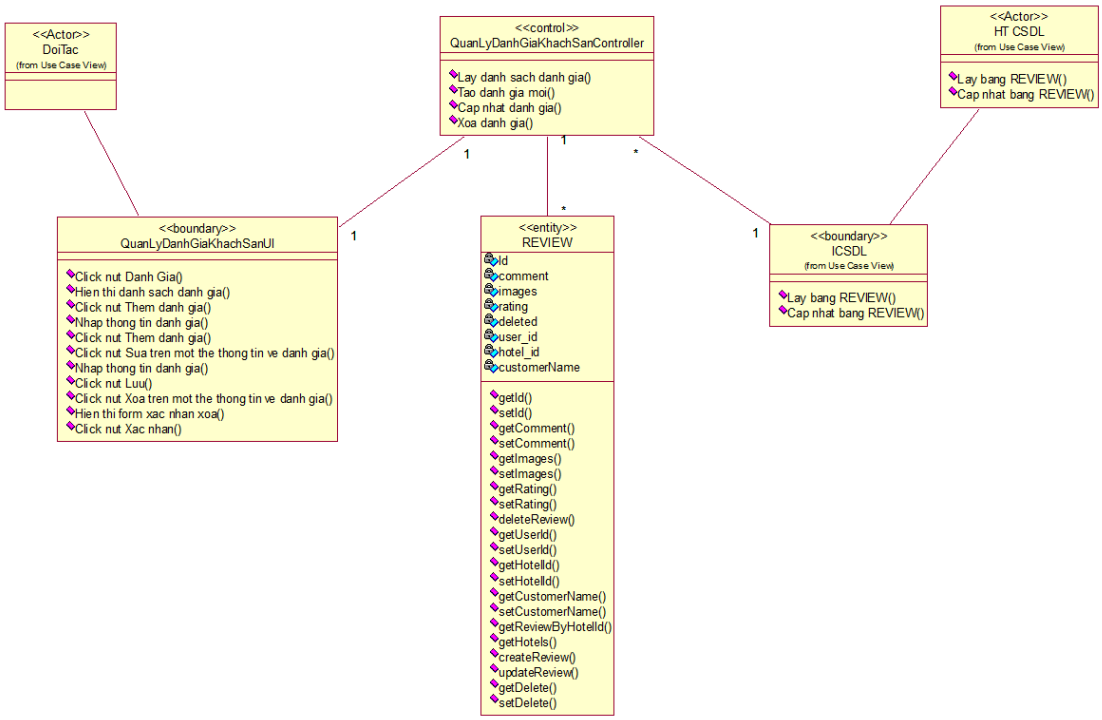
4.22.1 Biểu đồ trình tự





Hình 4.43. Biểu đồ trình tự Cập nhật mã giảm giá

4.22.2 Biểu đồ lớp



Hình 4.44. Biểu đồ lớp use case Quản lý đánh giá khách sạn

CHƯƠNG 5. CÀI ĐẶT VÀ KẾT QUẢ

5.1 Giới thiệu ngôn ngữ và công nghệ sử dụng

5.1.1 JavaScript

JavaScript là một ngôn ngữ lập trình phổ biến trong lĩnh vực phát triển web. Ban đầu, nó được tạo ra để thêm các chức năng tương tác và động vào các trang web. Tuy nhiên, với sự phát triển của cộng đồng phát triển phần mềm, JavaScript đã trở thành một ngôn ngữ lập trình đa năng được sử dụng cho nhiều mục đích khác nhau, bao gồm phát triển ứng dụng web, ứng dụng di động, game, và thậm chí là phần mềm server-side.

Dưới đây là một số điểm nổi bật của JavaScript:

Động (Dynamic): JavaScript là một ngôn ngữ lập trình động, cho phép thay đổi và tương tác với các thành phần của trang web một cách linh hoạt.

Client-Side Scripting: JavaScript thường được sử dụng để thêm các chức năng tương tác và động vào các trang web, bao gồm xử lý sự kiện người dùng, thao tác với DOM (Document Object Model), và gửi và nhận dữ liệu từ máy chủ.

Cú pháp dễ hiểu: Cú pháp của JavaScript tương đối đơn giản và dễ hiểu, cho phép người mới bắt đầu dễ dàng học và sử dụng.

Cộng đồng lớn và hỗ trợ mạnh mẽ: JavaScript có một cộng đồng phát triển lớn mạnh, với rất nhiều thư viện và framework được phát triển để hỗ trợ các mục đích khác nhau.

JavaScript là một phần không thể thiếu trong việc phát triển các ứng dụng web hiện đại, và với sự phát triển của Node.js, nó cũng đã mở ra khả năng phát triển ứng dụng server-side và ứng dụng di động.

5.1.2 TypeScript

TypeScript là một ngôn ngữ lập trình mạnh mẽ và mã nguồn mở được phát triển và duy trì bởi Microsoft. TypeScript xây dựng trên nền tảng JavaScript bằng cách bổ sung khả năng định nghĩa kiểu tĩnh (Static type definitions), biến nó trở thành một phần mở rộng (Superset) của JavaScript. TypeScript cải thiện

trải nghiệm phát triển bằng cách giúp lập trình viên phát hiện lỗi sớm thông qua kiểm tra kiểu dữ liệu, đồng thời hỗ trợ phát triển các ứng dụng quy mô lớn với chất lượng và khả năng bảo trì mã nguồn tốt hơn. Đối với các dự án quy mô lớn, việc áp dụng TypeScript có thể dẫn đến phần mềm mạnh mẽ hơn, trong khi vẫn triển khai được ở bất kỳ môi trường nào mà ứng dụng JavaScript thông thường có thể chạy. Dù không thể làm cho phần mềm của bạn hoàn toàn không có lỗi, TypeScript có thể ngăn chặn nhiều lỗi liên quan đến kiểu dữ liệu, cùng với Clever IntelliSense giúp tăng tốc độ phát triển.

Dưới đây là một số điểm nổi bật của TypeScript:

Kiểu dữ liệu tĩnh (Static Typing): Hệ thống kiểu của TypeScript giúp phát hiện lỗi ngay tại thời điểm biên dịch (compile time), giảm lỗi trong quá trình chạy (runtime errors) và nâng cao độ tin cậy của mã nguồn.

Hỗ trợ IDE nâng cao: TypeScript tích hợp xuất sắc với các môi trường phát triển tích hợp (IDEs) phổ biến như Visual Studio Code, cung cấp tính năng như tự động hoàn thành (autocompletion), tái cấu trúc mã nguồn (refactoring) và nhiều hơn thế.

Cải thiện khả năng đọc và bảo trì mã nguồn: Các định nghĩa kiểu (Type definitions) và giao diện (Interfaces) giúp mã nguồn dễ hiểu hơn và dễ bảo trì hơn.

Kiểu tổng quát (Generics): Cho phép tạo các thành phần có thể tái sử dụng và hoạt động với nhiều kiểu dữ liệu khác nhau.

Mô-đun (Modules): Tổ chức mã nguồn thành các đơn vị có thể tái sử dụng, giúp bảo trì mã nguồn tốt hơn.

Trình trang trí (Decorators): Thêm siêu dữ liệu (Metadata) vào các lớp và hàm để tùy chỉnh nâng cao.

Classes và Interfaces: Hỗ trợ lập trình hướng đối tượng với khả năng định nghĩa và mở rộng dữ liệu và chức năng rõ ràng.

TypeScript ngày càng trở nên phổ biến nhờ khả năng cải thiện chất lượng mã nguồn và hỗ trợ các dự án lớn, trở thành lựa chọn hàng đầu cho các nhà phát triển web hiện đại.

5.1.3 NestJS

NestJS là một framework dùng để xây dựng các ứng dụng phía server Node.js một cách hiệu quả và có khả năng mở rộng. Được xây dựng với TypeScript và hỗ trợ đầy đủ TypeScript, NestJS cho phép lập trình viên viết mã bằng TypeScript hoặc JavaScript thuần. Framework này kết hợp các yếu tố của lập trình hướng đối tượng (OOP), lập trình hàm (FP) và lập trình phản ứng hàm (FRP), đồng thời sử dụng các framework HTTP mạnh mẽ như Express (mặc định) và Fastify. NestJS mang đến một kiến trúc ứng dụng hoàn chỉnh, giúp các lập trình viên xây dựng các ứng dụng có khả năng kiểm thử cao, mở rộng tốt, kết nối lỏng lẻo và dễ dàng bảo trì. Với kiến trúc lấy cảm hứng từ Angular, NestJS giúp giải quyết các vấn đề cốt lõi về kiến trúc ứng dụng trên Node.js, cung cấp sự cân bằng giữa tính linh hoạt và tổ chức mã nguồn.

Dưới đây là một số điểm nổi bật của NestJS:

Hỗ trợ TypeScript mạnh mẽ: NestJS được xây dựng trên TypeScript, mang lại lợi ích của kiểu dữ liệu tĩnh và nâng cao khả năng bảo trì mã nguồn.

Kiến trúc module hoá: Cho phép tổ chức mã nguồn thành các module độc lập, giúp dễ dàng mở rộng và bảo trì ứng dụng.

Tích hợp với các framework HTTP phổ biến: NestJS hỗ trợ Express và Fastify, mang đến hiệu năng cao và khả năng tùy chỉnh linh hoạt.

Dependency Injection (DI): Tính năng tiêm phụ thuộc giúp quản lý các dịch vụ và giảm bớt sự phụ thuộc chặt chẽ giữa các thành phần trong ứng dụng.

Hỗ trợ Middleware và Pipes: NestJS cung cấp cơ chế middleware và pipes mạnh mẽ để xử lý yêu cầu và phản hồi trong ứng dụng.

Hệ thống Guard và Interceptor: Bảo mật và kiểm soát luồng dữ liệu hiệu quả thông qua Guard và Interceptor, giúp ứng dụng an toàn và linh hoạt hơn.

Tích hợp dễ dàng với thư viện bên thứ ba: NestJS cho phép sử dụng hàng ngàn module của bên thứ ba thông qua NPM, mang lại sự linh hoạt trong phát triển ứng dụng.

Khả năng mở rộng và kiểm thử cao: Với cấu trúc rõ ràng và module hoá, NestJS hỗ trợ kiểm thử đơn vị và mở rộng ứng dụng một cách dễ dàng.

Nhờ vào khả năng kết hợp TypeScript và kiến trúc module hoá, NestJS đang trở thành lựa chọn hàng đầu cho việc phát triển các ứng dụng server-side Node.js hiện đại.

5.1.4 Next.js

Next.js là một framework phát triển ứng dụng React mạnh mẽ được xây dựng trên nền tảng Node.js. Nó cho phép các lập trình viên phát triển các ứng dụng React với tính năng kết xuất phía server (SSR) và tạo trang tĩnh (SSG) một cách dễ dàng và hiệu quả. Next.js được tối ưu hóa để phát triển các ứng dụng web nhanh, linh hoạt và có khả năng mở rộng cao. Next.js cung cấp môi trường phát triển mạnh mẽ với khả năng tối ưu hoá hiệu suất, giúp cải thiện trải nghiệm người dùng thông qua các tính năng như tải trang nhanh, hỗ trợ SEO tốt và chia sẻ tài nguyên hiệu quả.

Dưới đây là một số điểm nổi bật của Vue.js:

Kết xuất phía server (SSR): Next.js hỗ trợ kết xuất trang phía server, giúp cải thiện SEO và tăng tốc độ tải trang ban đầu.

Tạo trang tĩnh (SSG): Cho phép tạo các trang tĩnh với nội dung được dựng sẵn khi build, giúp tối ưu tốc độ và tài nguyên.

Routing tự động: Hệ thống routing dựa trên cấu trúc thư mục, không cần phải cấu hình thủ công.

Tích hợp API Routes: Next.js cho phép tạo các API backend ngay trong ứng dụng bằng cách sử dụng thư mục pages/api.

Hỗ trợ Pre-fetching: Tự động prefetch các liên kết nội bộ khi chúng xuất hiện trong viewport, giúp cải thiện tốc độ điều hướng trang.

Tối ưu hoá hình ảnh: Next.js cung cấp thành phần next/image để tối ưu hóa hình ảnh, hỗ trợ lazy loading và định dạng hiện đại.

Hỗ trợ TypeScript: Next.js có sẵn cấu hình TypeScript, giúp phát triển ứng dụng với độ ổn định và bảo trì cao hơn.

Tích hợp CSS và Sass: Hỗ trợ viết CSS Module, SCSS, và tích hợp các công cụ styling hiện đại.

Hiệu năng cao: Tối ưu hóa việc tải tài nguyên, render nhanh chóng và hỗ trợ các công cụ phân tích hiệu suất như Lighthouse.

Next.js là một framework hàng đầu để phát triển các ứng dụng React hiện đại, cung cấp giải pháp tối ưu cho SEO, tốc độ và trải nghiệm người dùng, đồng thời hỗ trợ mạnh mẽ các tính năng như SSR, SSG và API Routes.

5.1.5 PostgreSql

PostgreSQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở mạnh mẽ và linh hoạt. Nó được thiết kế để xử lý các ứng dụng có yêu cầu cao về tính toàn vẹn dữ liệu, hiệu suất và mở rộng. Dưới đây là một số điểm nổi bật của PostgreSQL:

Mã nguồn mở: PostgreSQL là một dự án mã nguồn mở, điều này có nghĩa là bạn có thể sử dụng và tùy chỉnh mã nguồn theo nhu cầu của mình mà không cần phải trả bất kỳ chi phí nào.

Hỗ trợ đa nền tảng: PostgreSQL có sẵn cho nhiều hệ điều hành khác nhau như Linux, Windows và macOS, cho phép bạn chạy cơ sở dữ liệu trên nhiều môi trường.

Tính toàn vẹn dữ liệu cao: PostgreSQL cung cấp các tính năng để đảm bảo tính toàn vẹn dữ liệu, bao gồm ràng buộc khóa ngoại, kiểm soát truy cập và giao dịch ACID (Atomicity, Consistency, Isolation, Durability).

Hiệu suất cao: PostgreSQL được tối ưu hóa để xử lý các tải công việc lớn và cung cấp các công cụ để tinh chỉnh hiệu suất, bao gồm các chỉ mục và truy vấn song song.

PostgreSQL là một lựa chọn phổ biến cho các ứng dụng web và doanh nghiệp, đặc biệt là khi cần một hệ quản trị cơ sở dữ liệu mạnh mẽ, linh hoạt và ổn định.

5.1.6 RabbitMQ

RabbitMQ là một message broker mã nguồn mở, phổ biến trong việc xử lý các nhiệm vụ nền (background jobs), định tuyến tin nhắn (routing messages), và đóng vai trò là trung gian giao tiếp giữa các microservices. Dưới đây là một số điểm nổi bật của RabbitMQ:

Hàng đợi tin nhắn (Message Queuing): Hỗ trợ lưu trữ và quản lý hàng đợi tin nhắn, đảm bảo các tin nhắn được xử lý tuần tự và không bị mất.

Mô hình xuất bản/đăng ký (Publish/Subscribe): RabbitMQ cho phép mô hình publish/subscribe, nơi một producer có thể gửi tin nhắn đến nhiều consumer cùng lúc.

Độ tin cậy cao: Với persistent messaging và acknowledgments, RabbitMQ đảm bảo tin nhắn không bị mất ngay cả khi xảy ra sự cố hệ thống.

Cụm máy chủ (Clustering): RabbitMQ hỗ trợ clustering để tăng khả năng sẵn sàng và mở rộng, giúp hệ thống ổn định hơn khi có nhiều yêu cầu.

Quản lý trực quan: RabbitMQ cung cấp giao diện web trực quan để quản lý, giám sát và phân tích các hoạt động trong hệ thống.

RabbitMQ là một giải pháp mạnh mẽ và linh hoạt trong việc xử lý giao tiếp giữa các thành phần của hệ thống. Với các tính năng nổi bật như quản lý hàng đợi tin nhắn, hỗ trợ publish/subscribe, và độ tin cậy cao, RabbitMQ là lựa chọn hàng đầu cho các ứng dụng cần xử lý khối lượng lớn tin nhắn hoặc giao tiếp giữa các microservices.

5.1.7 Hệ thống Middleware Logging và Monitoring

Hệ thống middleware logging và monitoring được thiết kế để theo dõi và ghi nhận các hoạt động của người dùng cũng như các lỗi phát sinh trong hệ thống. Việc giám sát và phân tích các hoạt động này giúp đội ngũ phát triển có thể phát hiện sự cố nhanh chóng và tối ưu hóa hiệu suất hệ thống.

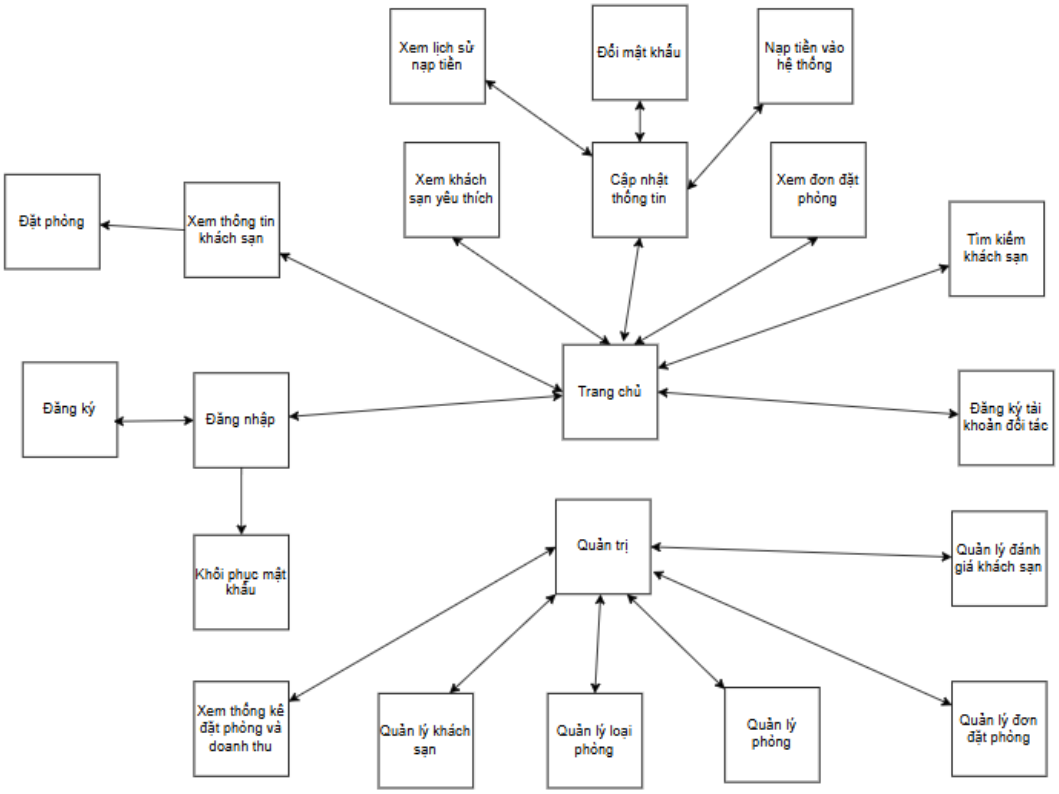
Dưới đây là một số điểm nổi bật của hệ thống Middleware Logging và Monitoring:

Activity Logging Middleware: Middleware này có chức năng ghi lại các hoạt động của người dùng và gửi các thông tin vào bot Telegram. Mỗi khi người dùng thực hiện các thao tác như đăng nhập, thay đổi dữ liệu, hay truy cập API, các thông tin này sẽ được ghi lại và gửi đi. Thông tin cần ghi nhận gồm: thời gian thực hiện, email (nếu đã đăng nhập), địa chỉ IP, Endpoint được truy cập, phương thức thực hiện.

Error Tracking và Notification Sytem: Hệ thống Error Tracking tự động phát hiện lỗi và gửi thông báo chi tiết đến bot Telegram. Điều này giúp đội ngũ phát triển nắm bắt ngay lập tức các sự cố và giải quyết chúng một cách nhanh chóng.

Cronjob Tự Động Gửi Báo Cáo: Cronjob giúp tự động hóa việc gửi báo cáo theo lịch trình định kỳ. Điều này giúp đội ngũ phát triển không cần phải làm thủ công mỗi khi cần tổng hợp thông tin. Tổng hợp log hoạt động và lỗi hệ thống định kỳ gửi chúng qua bot Telegram. Giúp đội ngũ phát triển dễ dàng nắm bắt tình trạng hệ thống mà không phải kiểm tra thường xuyên.

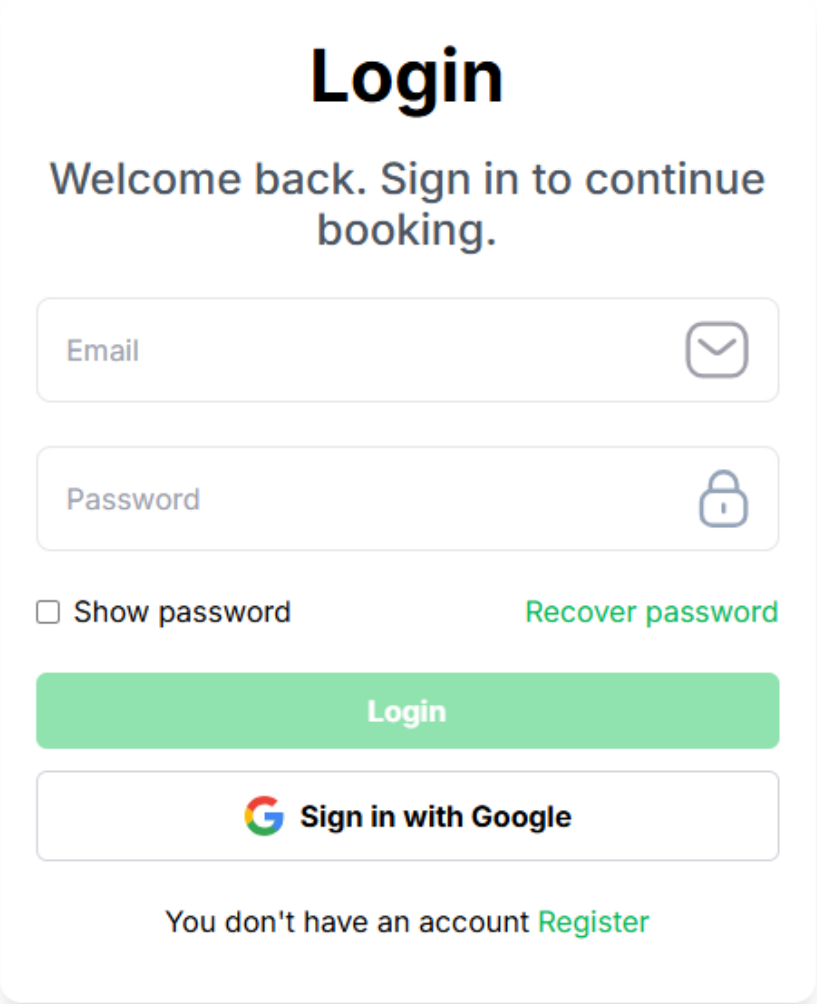
5.2 Sơ đồ điều hướng giữa các màn hình



Hình 5.1. Sơ đồ điều hướng giữa các màn hình

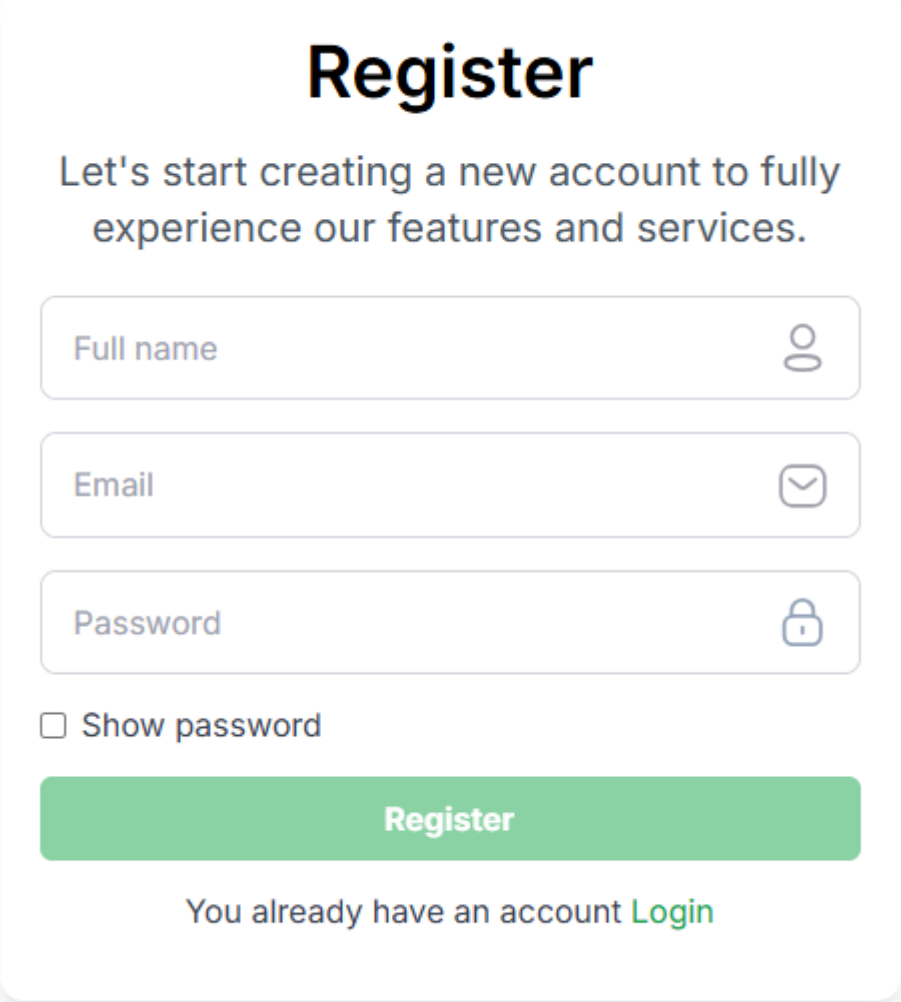
5.3 Kết quả chương trình

5.3.1 Màn hình trang đăng nhập



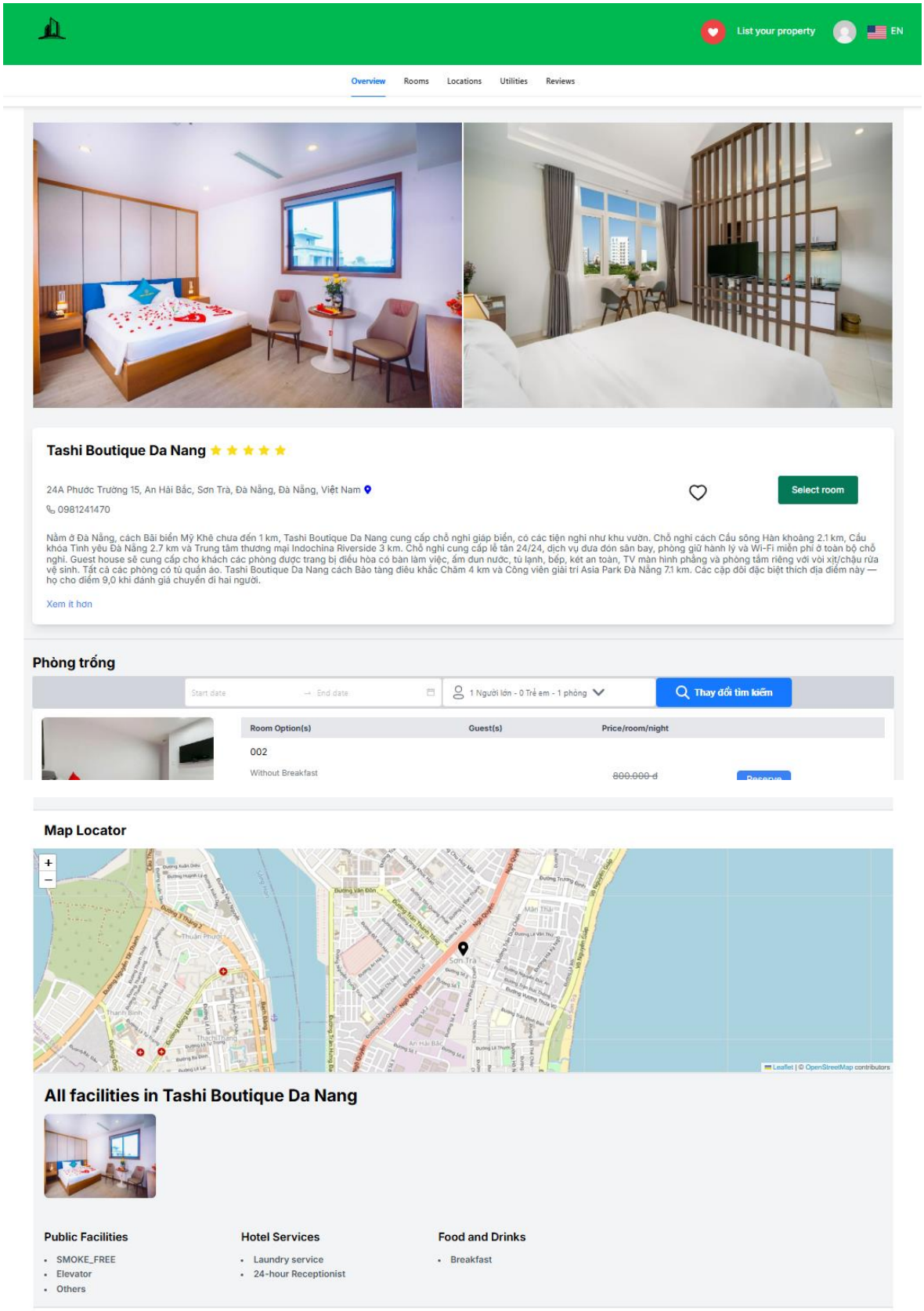
Hình 5.2. Màn hình trang đăng nhập

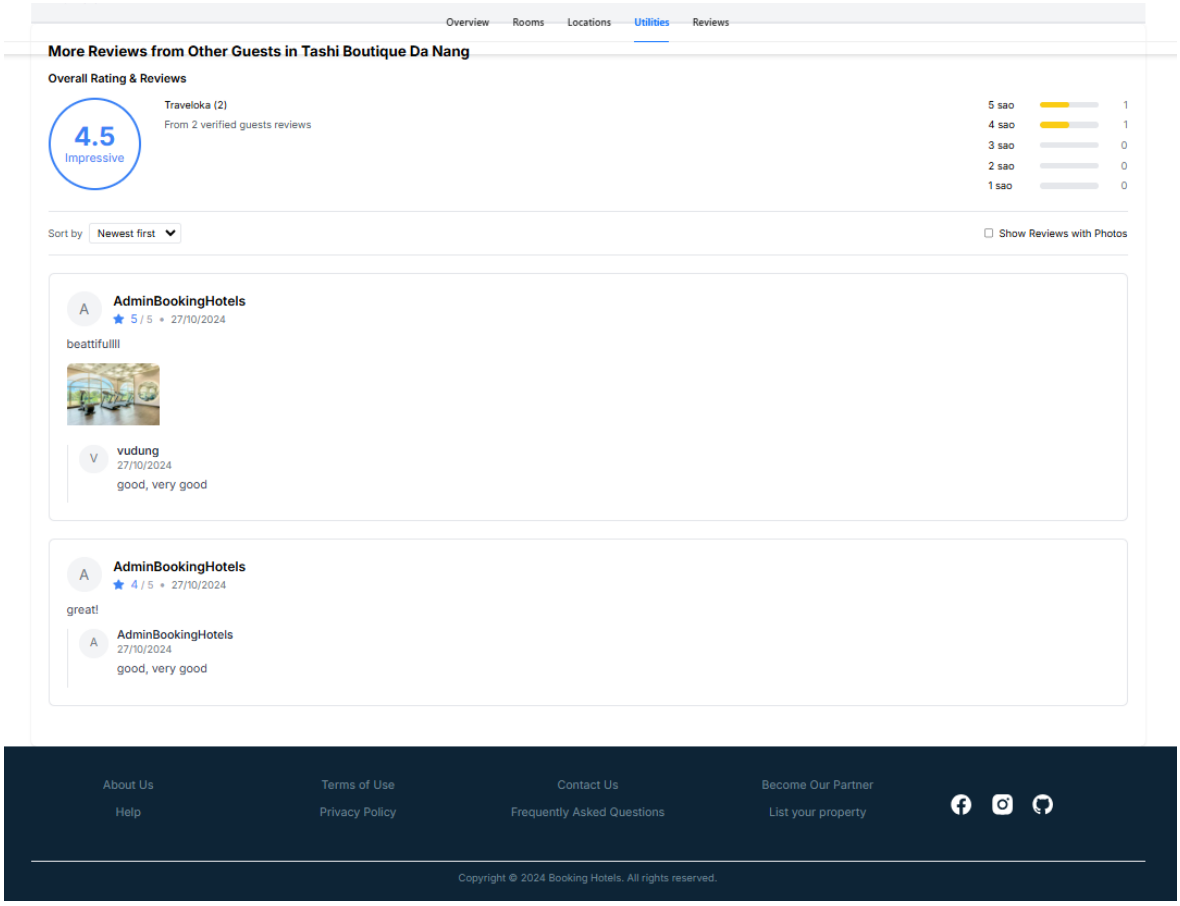
5.3.2 Màn hình đăng ký



Hình 5.3. Màn hình đăng ký

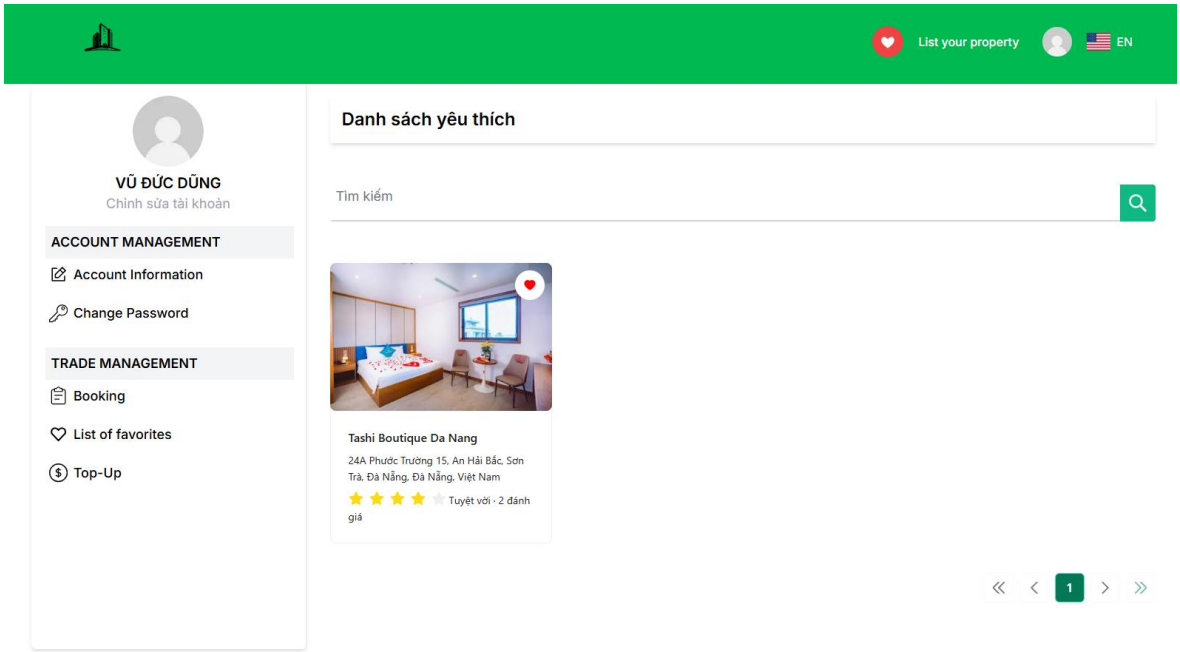
5.3.3 Màn hình trang xem thông tin khách sạn





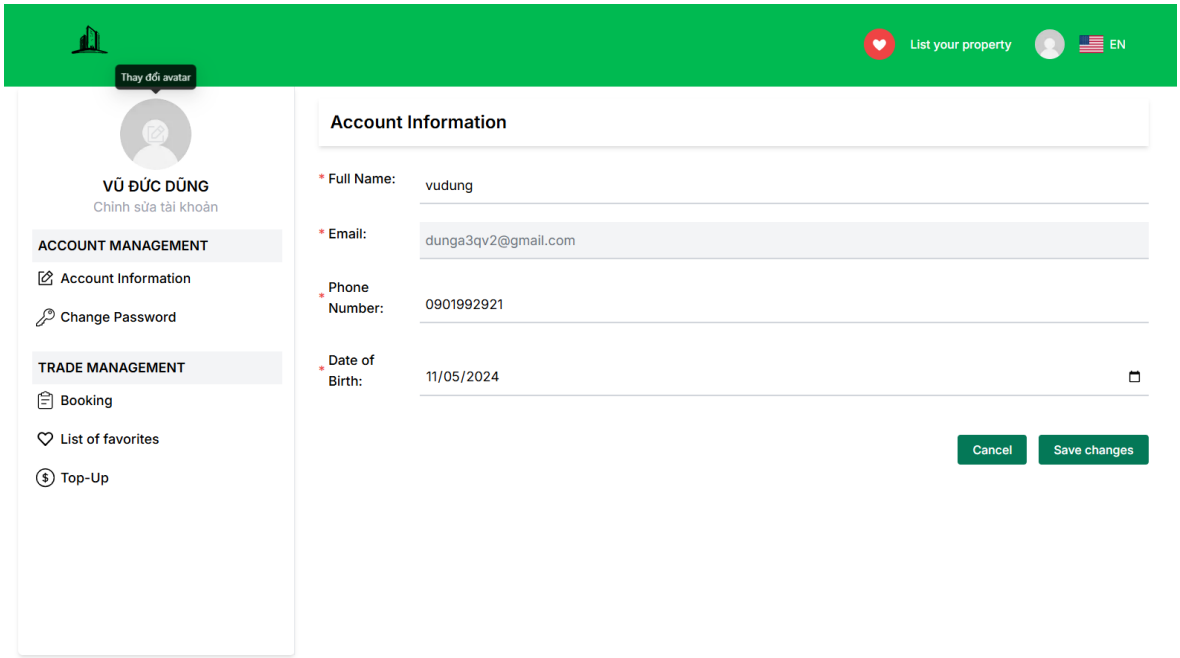
Hình 5.4. Màn hình trang xem thông tin khách sạn

5.3.4 Màn hình trang xem khách sạn yêu thích



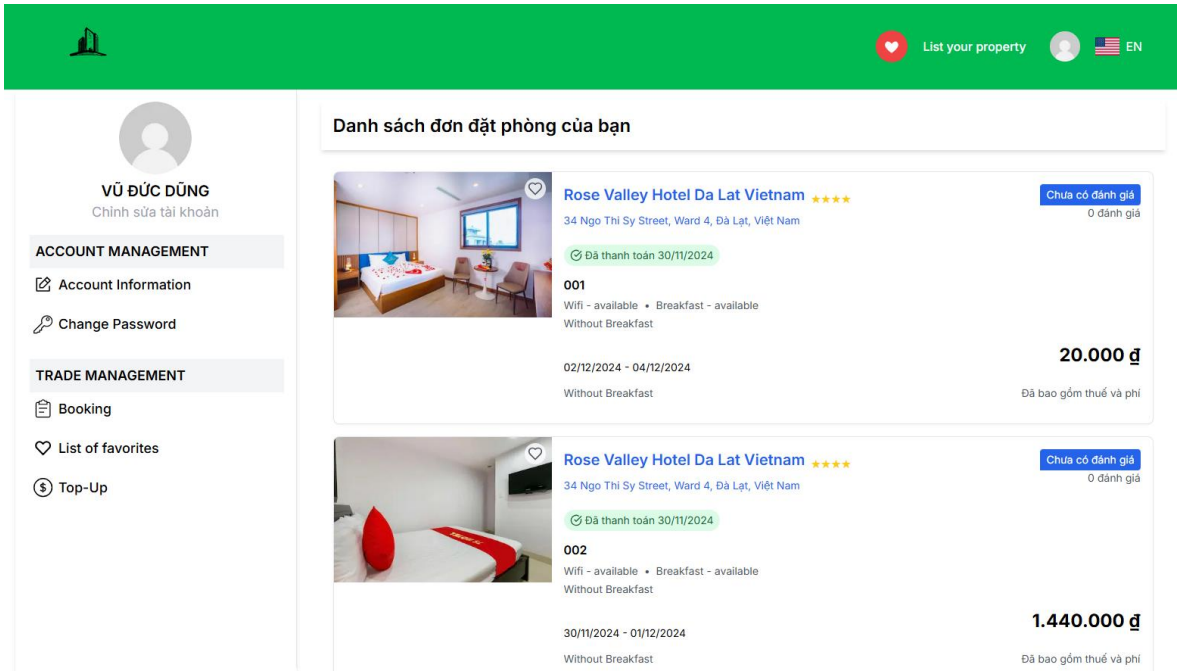
Hình 5.5. Màn hình trang xem khách sạn yêu thích

5.3.5 Màn hình trang cập nhập thông tin



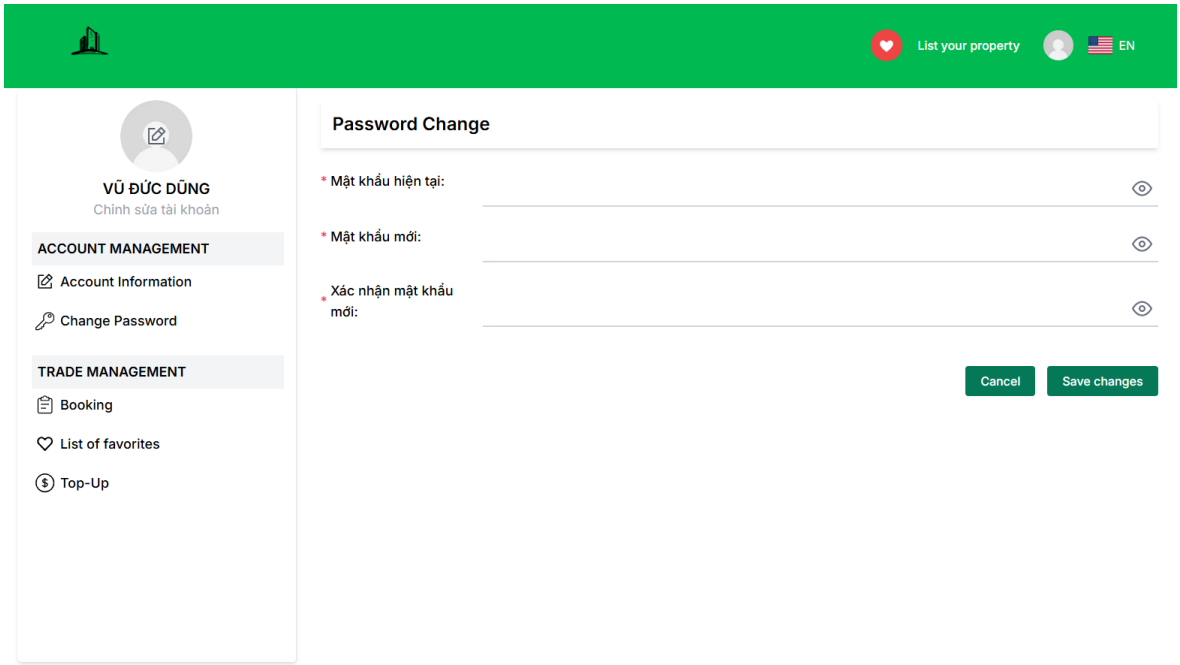
Hình 5.6. Màn hình trang cập nhật thông tin

5.3.6 Màn hình trang xem đơn đặt phòng



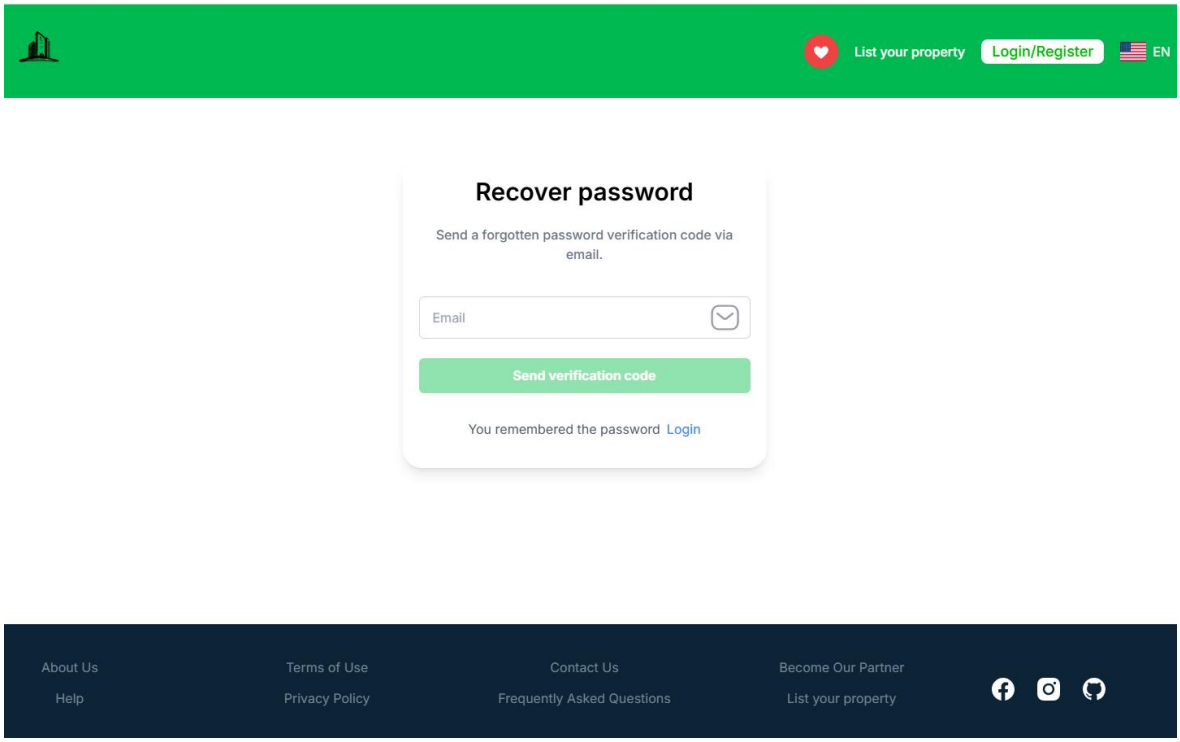
Hình 5.7. Màn hình xem đơn đặt phòng

5.3.7 Màn hình trang đổi mật khẩu



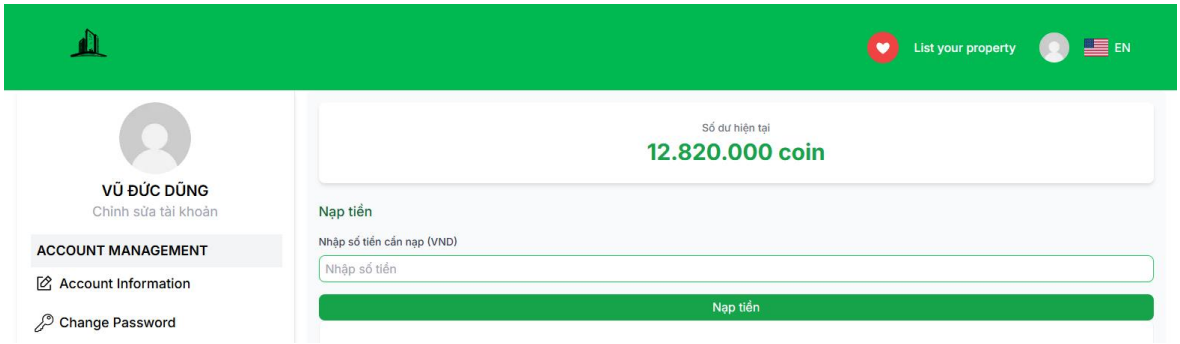
Hình 5.8. Màn hình trang đổi mật khẩu

5.3.8 Màn hình trang khôi phục mật khẩu



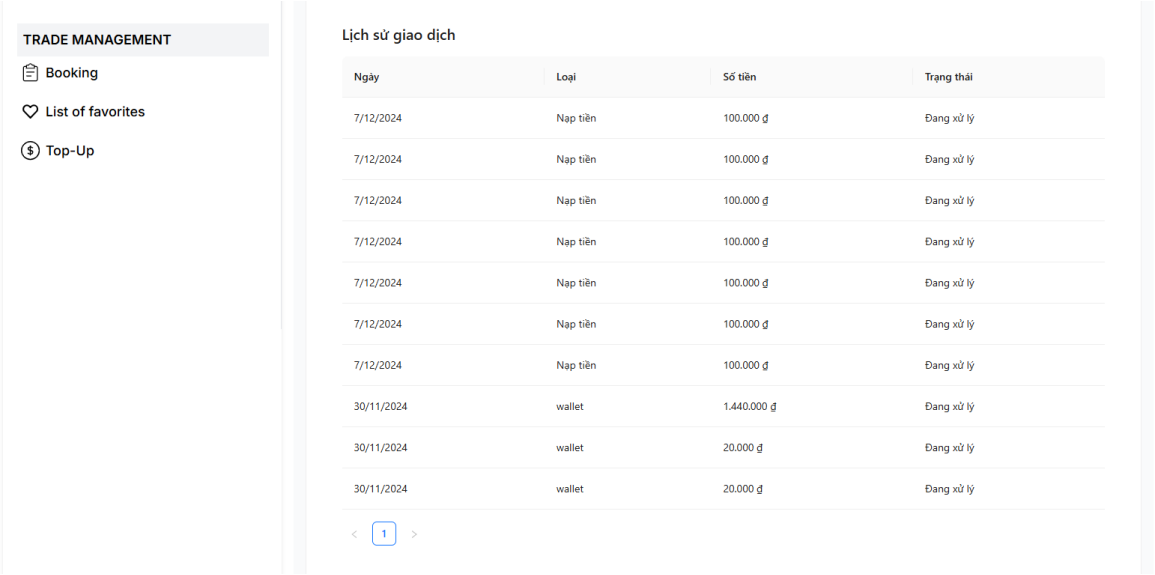
Hình 5.9. Màn hình trang khôi phục mật khẩu

5.3.9 Màn hình trang nạp tiền vào hệ thống



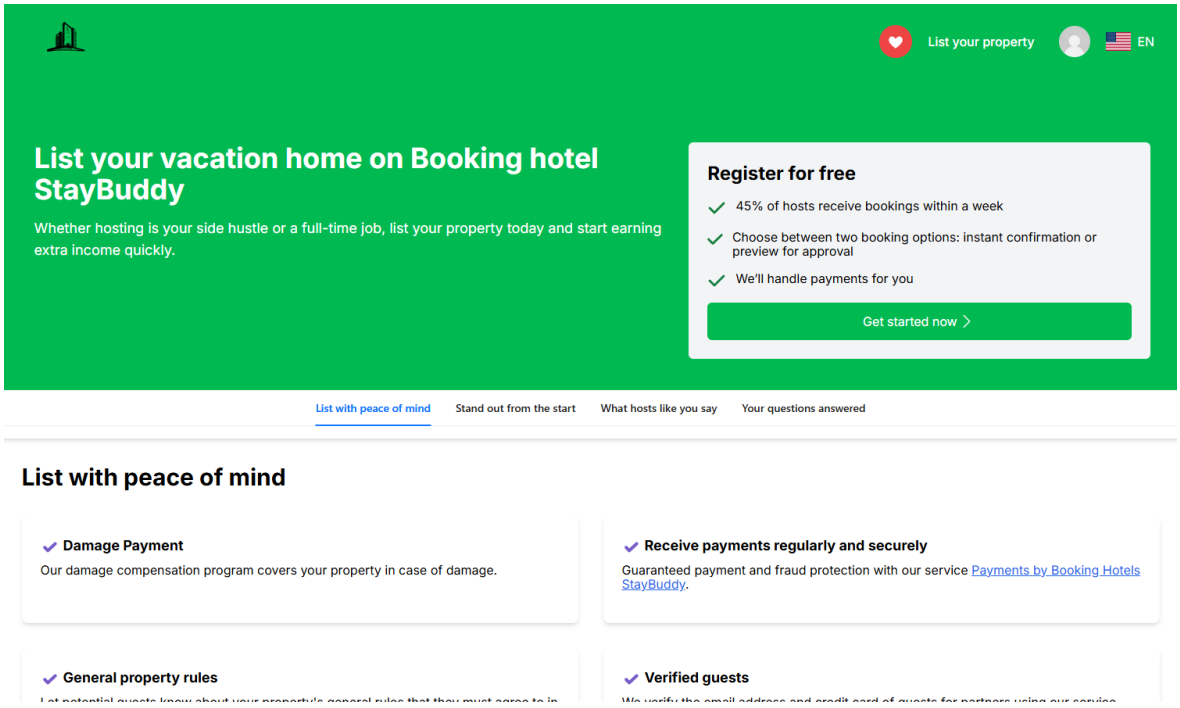
Hình 5.10. Màn hình trang nạp tiền vào hệ thống

5.3.10 Màn hình trang xem lịch sử nạp tiền



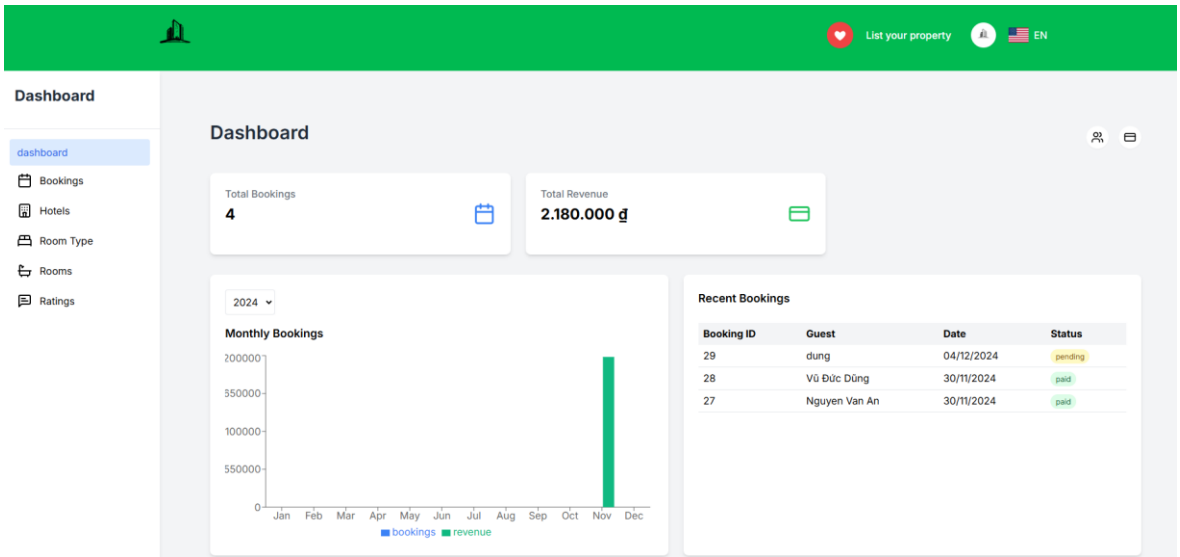
Hình 5.11. Màn hình trang quản lý xem lịch sử nạp tiền

5.3.11 Màn hình trang đăng ký tài khoản đối tác



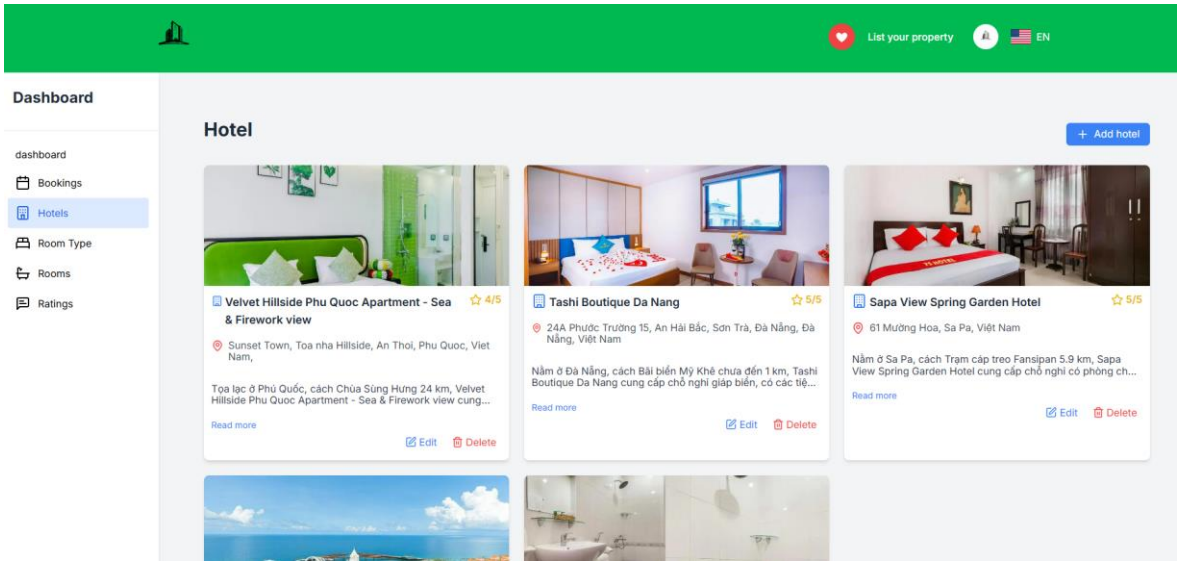
Hình 5.12. Màn hình trang đăng ký tài khoản đối tác

5.3.12 Màn hình trang xem thống kê đặt phòng và doanh thu



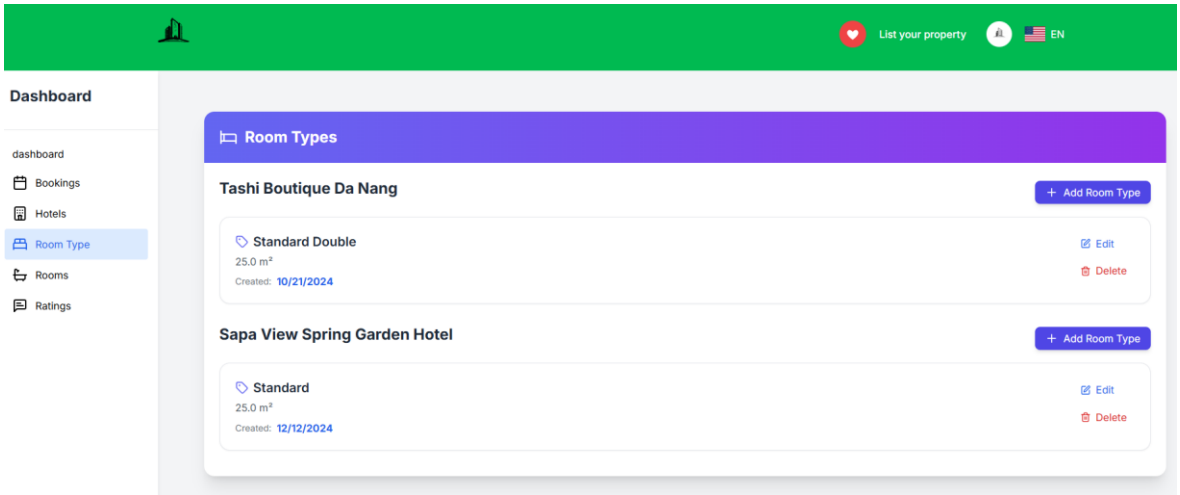
Hình 5.13. Màn hình trang xem thống kê đặt phòng và doanh thu

5.3.13 Màn hình trang quản lý khách sạn



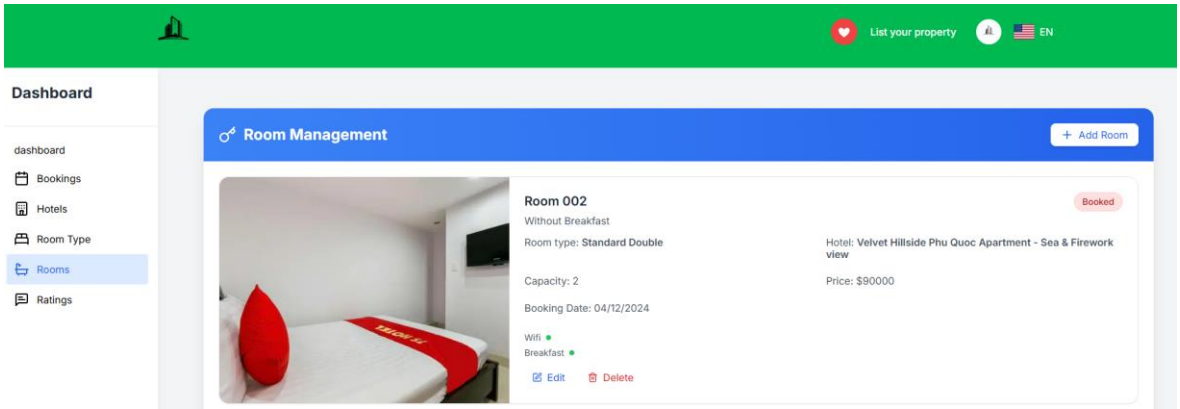
Hình 5.14. Màn hình trang quản lý khách sạn

5.3.14 Màn hình trang quản lý loại phòng



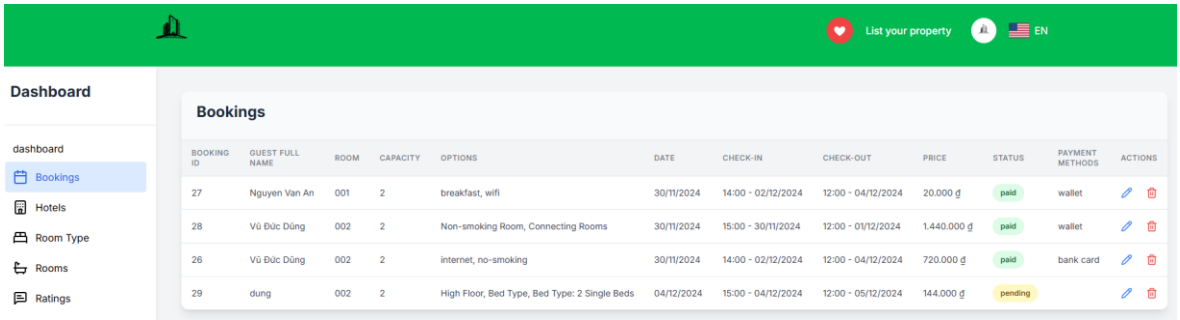
Hình 5.15. Màn hình trang quản lý loại phòng

5.3.15 Màn hình trang quản lý phòng



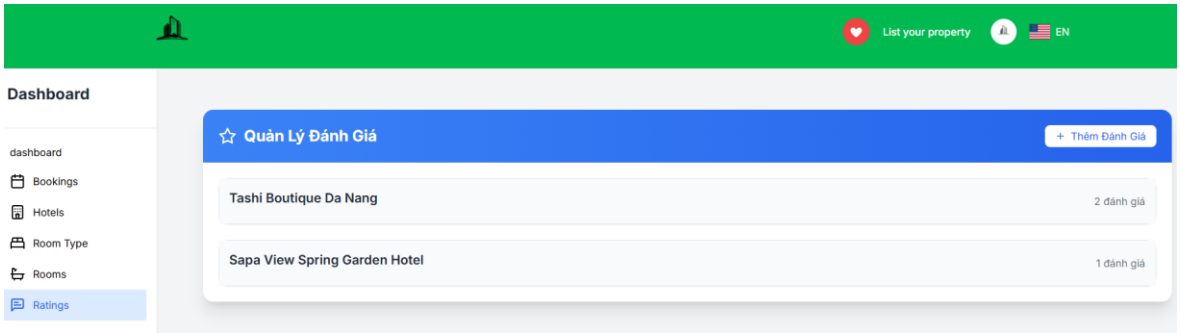
Hình 5.16. Màn hình trang quản lý phòng

5.3.16 Màn hình trang cập nhật đơn đặt phòng



Hình 5.17. Màn hình trang cập nhật đơn đặt phòng

5.3.17 Màn hình trang quản lý đánh giá khách sạn



Hình 5.18. Màn hình trang quản lý đánh giá khách sạn

CHƯƠNG 6. KIỂM THỬ HỆ THỐNG

6.1 Kế hoạch kiểm thử

6.1.1 Lịch trình công việc

Bảng 6.1. Bảng lập kế hoạch kiểm thử

Cột mốc	Sản phẩm bàn giao	Thời gian làm	Ngày bắt đầu	Ngày kết thúc
Lập kế hoạch kiểm thử	Tài liệu Test Plan	2 ngày	10/12/2024	11/12/2024
Xem lại các tài liệu	Tài liệu Test Plan	1 ngày	15/12/2024	15/12/2024
Thiết kế các testcase	Tài liệu Testcase	1 ngày	16/12/2024	16/12/2024
Viết các testcase và xem lại các test case	Tài liệu Testcase	2 ngày	17/12/2024	18/12/2024
Thực thi các testcase	Tài liệu Testcase	1 ngày	18/12/2024	18/12/2024

6.1.2 Những yêu cầu về tài nguyên

- **Phần cứng**

Bảng 6.2. Bảng yêu cầu phần cứng

CPU	RAM	HDD	Architecture
Intel Core i5, 2.6 GHz	4 GB	500GB	64 bit

- **Phần mềm**

Bảng 6.3. Bảng yêu cầu phần mềm

Tên phần mềm	Phiên bản	Loại
IE8	8.0.6001.18702IC	Trình duyệt web
Google Chrome	19.0.1084.56	Trình duyệt web
Microsoft Windown 10 professional	10	Hệ điều hành

- **Công cụ kiểm thử**

Bảng 6.4. Bảng yêu cầu về công cụ kiểm thử

Hoạt động	Công cụ	Nhà cung cấp	Phiên bản
Quản lý Testcase	Microsoft Office Excel	Microsoft	2018
Quản lý Configuration	Microsoft Office Word	Microsoft	2018
Defect Tracking	Microsoft Office Excel	Microsoft	2018
Quản lý tiến độ kiểm thử	Microsoft Office Excel	Microsoft	2018

- **Môi trường kiểm thử**
  + Máy tính cá nhân truy cập vào project bằng trình duyệt
  + Các chức năng của project được kiểm tra trên các trình duyệt IE8, Google Chrome.
  + Hệ điều hành sử dụng là Microsoft Windows 11

6.1.3 Phạm vi kiểm thử

- Đăng nhập
- Đăng ký
- Xem thông tin khách sạn
- Xem khách sạn yêu thích
- Cập nhật thông tin

- Xem đánh giá khách sạn
- Xem đơn đặt phòng
- Tìm kiếm khách sạn
- Cập nhật khách sạn yêu thích
- Đặt phòng
- Đổi mật khẩu
- Khôi phục mật khẩu
- Nạp tiền vào hệ thống
- Xem lịch sử nạp tiền
- Đăng ký tài khoản đối tác
- Xem thống kê đặt phòng và doanh thu
- Quan lý khách sạn
- Quản lý loại phòng
- Quản lý phòng
- Cập nhật đơn đặt phòng
- Quản lý đánh giá khách sạn

6.1.4 Chiến lược kiểm thử

- **Giai đoạn kiểm thử:** Kiểm thử trong khi code và kiểm thử sau khi hoàn thành code chương trình
- **Các loại kiểm thử:**
  + Kiểm thử chức năng

Bảng 6.5. Bảng yêu cầu kiểm thử chức năng

Mục đích	+Đảm bảo các chức năng được kiểm tra hoạt động chính xác theo yêu cầu đặc tả
Kỹ thuật	+Thực thi tất cả các trường hợp có thể có cho mỗi nhóm chức năng sử dụng dữ liệu hợp lệ và không hợp lệ để xác định: - Kết quả mong đợi khi dữ liệu hợp lệ được sử dụng

	- Cảnh báo phù hợp hiện ra khi dữ liệu không hợp lệ được sử dụng
Tiêu chuẩn dừng	+Tất cả các test case đã được thiết kế đều được thực thi +Tất cả các lỗi tìm thấy đều được ghi nhận lý do rõ ràng để có thể giúp cho developer khắc phục
Phụ trách	Test Designer/Tester
Cách kiểm thử	Kiểm thử thủ công tuân theo các bước định nghĩa trong test case
Xử lý ngoại lệ	Liện kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử

+ Kiểm thử phi chức năng

Bảng 6.6. Bảng yêu cầu kiểm thử phi chức năng

Mục đích	Đảm bảo các hệ thống hoạt động tốt, đảm bảo các yêu cầu phi chức năng đã phân tích
Kỹ thuật	Kiểm tra tất cả các màn hình trên tất cả các trình duyệt được hỗ trợ
Tiêu chuẩn dừng	Hệ thống đáp ứng tất cả các yêu cầu phi chức năng đã được định nghĩa
Phụ trách	Test Designer/Tester
Cách kiểm thử	Kiểm thử thủ công tuân theo các bước định nghĩa trong test case trền nhiều trình duyệt và màn hình khác nhau
Xử lý ngoại lệ	Liện kê tất cả các vấn đề liên quan phát sinh trong quá trình thực thi kiểm thử

6.1.5 Quản lý lỗi

- **Phân loại lỗi**

Bảng 6.7. Bảng phân loại lỗi

Mức độ nghiêm trọng	Đặc tả lỗi
Cao	-Không thể tạo tài khoản mới -Không thể đăng nhập -Không thể xem chi tiết khách sạn -Không thể tìm kiếm khách sạn -Không thể liên hệ được với hệ thống -Không khôi phục được mật khẩu -Không đặt phòng được -Không thống kê được hệ thống -Không hiển thị thông tin ở các chức năng quản lý
Trung bình	-Tìm kiếm sai khách sạn -Hiển thị sai thông tin hoạt động của người dùng -Hiển thị sai thông tin đơn đặt phòng -Hiển thị sai thông tin ở các chức năng quản lý -Hiển thị sai thông tin thống kê
Thấp	-Hiện thị kết quả tìm kiếm không đúng mẫu -Các chức năng chạy không ổn định

- **Quy trình xử lý lỗi**
  + Ghi nhận lại các lỗi được tìm thấy trong quá trình kiểm thử trên bảng thống kê lỗi
  + Gán bugs cho người phụ trách
- **Tài liệu đầu ra:** Test case

6.2 Test case

6.2.1 Thiết kế test case

- **Chức năng Đăng nhập**

Bảng 6.8. Bảng test case chức năng Đăng nhập

	Trường hợp kiểm thử	Các bước thực hiện	Kết quả mong muốn
1	Bỏ trống các mục bắt buộc	Bỏ trống các trường bắt buộc click Đăng nhập	Thông báo nhập đầy đủ thông tin
2	Sai tên đăng nhập hoặc mật khẩu	Nhập sai tên đăng nhập hoặc mật khẩu và click Đăng nhập	Thông báo sai tên đăng nhập hoặc mật khẩu, đăng nhập thất bại
3	Nhập đúng tên đăng nhập và mật khẩu	Nhập đúng tên đăng nhập và mật khẩu click "Đăng nhập"	Đăng nhập thành công, chuyển tới màn hình chức năng tương ứng
4	Không nhập gì	Không nhập mục nào cả và click vào "Đăng nhập"	Thông báo phải nhập đầu đủ các mục, đăng nhập thất bại

- **Chức năng Đăng ký**

Bảng 6.9. Bảng test case chức năng Đăng ký

	Trường hợp kiểm thử	Các bước thực hiện	Kết quả mong muốn
1	Bỏ trống tất cả các trường	Bỏ trống các trường bắt buộc click Đăng ký	Hiển thị thông báo nhập đầy đủ thông tin
2	Nhập email không đúng định dạng abcd@gmail.com	Nhập email sai định dạng click Đăng ký	Cảnh báo nhập email không đúng định dạng
3	Mật khẩu dưới độ dài tối thiểu	Nhập mật khẩu ít hơn 6 ký tự click Đăng ký	Cảnh báo mật khẩu phải có ít nhất 6 kí tự
4	Đăng ký bằng email đã tồn tại	Đăng ký bằng email đã tồn tại click Đăng ký	Thông báo email đã tồn tại, đăng ký thất bại.

- **Chức năng Xem thông tin khách sạn**

Bảng 6.10. Bảng test case chức năng Xem thông tin khách sạn

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Xem khách sạn thành công	Click vào 1 khách sạn trong trang chủ	Hiển thị chi tiết thông tin khách sạn gồm tên khách sạn, hình ảnh, loại phòng, phòng, đánh giá, địa chỉ, số điện thoại liên hệ, mô tả.

- **Chức năng Xem khách sạn yêu thích**

Bảng 6.11. Bảng test case chức năng Xem khách sạn yêu thích

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Xem khách sạn yêu thích thành công	Click vào biểu tượng trái tim trên thanh	Hiển thị danh sách khách sạn yêu thích gồm hình ảnh, tên khách sạn, địa chỉ, đánh giá.

- **Chức năng Cập nhật thông tin**

Bảng 6.12. Bảng test case chức năng Cập nhật thông tin

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**

1	Kiểm tra cập nhật Ảnh đại diện	Click avatar của người dùng	Cập nhật thành công. Hiển thị thông tin sau khi cập nhật.
2	Kiểm tra cập nhật Họ tên	Click texbox Họ và nhập thông tin, click button "Lưu thông tin"	Cập nhật thành công. Hiển thị thông tin sau khi cập nhật.
3	Kiểm tra cập nhật Ngày sinh	Click texbox Ngày sinh và nhập thông tin đúng format Date hoặc chọn ngày tại Datepicker, click button "Lưu thông tin"	Cập nhật thành công. Hiển thị thông tin sau khi cập nhật.
4	Kiểm tra cập nhật Số điện thoại	Click texbox Số điện thoại và nhập thông tin đúng format số điện thoại, click button "Lưu thông tin"	Cập nhật thành công. Hiển thị thông tin sau khi cập nhật.

- **Chức năng Xem đánh giá khách sạn**

Bảng 6.13. Bảng test case chức năng Xem đánh giá khách sạn

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Xem đánh giá thành công	Click tab đánh giá trong trang chi tiết khách sạn	Hiển thị các đánh giá gồm tên người dùng, ảnh đại diện, điểm đánh giá trung bình, nội dung đánh giá, ngày đánh giá, ảnh kèm theo (nếu có).

- **Chức năng Xem đơn đặt phòng**

Bảng 6.14. Bảng test case chức năng Xem đơn đặt phòng

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Xem đơn đặt phòng	Click Đơn đặt phòng tai menu tài khoản	Hiển thị các đánh giá gồm tên phòng, ảnh phòng, mô tả đánh giá, địa chỉ, trạng thái thanh toán.

- **Chức năng Tìm kiếm khách sạn**

Bảng 6.15. Bảng test case chức năng Tìm kiếm khách sạn

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Không nhập thông tin tìm kiếm	Không nhập thông tin tìm kiếm, click vào button "Tìm kiếm"	Thông báo lỗi và yêu cầu nhập thông tin tìm kiếm
2	Nhập thiếu thông tin tìm kiếm	Nhập từ khóa tìm kiếm, không nhập ngày, số lượng, click vào button "Tìm kiếm"	Thông báo lỗi và yêu cầu nhập đầy đủ thông tin tìm kiếm
3	Nhập đầy đủ thông tin tìm kiếm	Nhập đầy đủ thông tin, click vào button "Tìm kiếm"	Trả về kết quả đúng với thống tin tìm kiếm

- **Chức năng Cập nhật khách sạn yêu thích**

Bảng 6.16. Bảng test case chức năng Cập nhật khách sạn yêu thích

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Thêm khách sạn yêu thích thành công	Click vào biểu tượng trái tim trên thẻ khách sạn	Thông báo thêm thành công
2	Xóa khách sạn yêu thích	Click vào biểu tượng trái tim trên thẻ khách sạn đã yêu thích	Thông báo xóa thành công

- **Chức năng Đặt phòng**

Bảng 6.17. Bảng test case chức năng Đặt phòng

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Nhập thiếu thông tin cho đơn đặt phòng	Click vào textbox email và nhập thông tin, click vào button "Tiếp tục thanh toán"	Hiển thị thông báo yêu cầu nhập thống tin còn thiếu
2	Nhập đầy đủ thông tin cho đơn đặt phòng	Nhập đầy đủ thông tin, click vào button "Tiếp tục thanh toán"	Tạo mới đơn đặt phòng

- **Chức năng Đổi mật khẩu**

Bảng 6.18. Bảng test case chức năng Đổi mật khẩu

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Nhập mật khẩu mới không đủ độ dài	Nhập mật khẩu hiện tại, mật khẩu mới với độ dài dưới 6 kí tự, xác nhận mật khẩu, click vào Lưu thông tin	Hiển thị cảnh báo mật khẩu không đủ độ dài
2	Nhập mật khẩu hiện tại không đúng	Nhập mật khẩu hiện tại sai, mật khẩu mới, xác nhận mật khẩu,	Hiển thị thông báo mật khẩu không đúng

		Click vào button "Lưu thông tin"	
3	Nhập xác nhận mật khẩu không khớp với mật khẩu mới	Nhập mật khẩu hiện tại, mật khẩu mới, xác nhận mật khẩu không khớp với mật khẩu mới, click vào button "Lưu thông tin"	Hiển thị cảnh báo xác nhận mật khẩu không khớp

- **Chức năng Khôi phục mật khẩu**

Bảng 6.19. Bảng test case chức năng Khôi phục mật khẩu

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Không nhập email	Click vào "Gửi mã xác nhận"	Hiển thị thông báo yêu cầu nhập email
2	Nhập sai mã xác thực	Click vào textbox mã xác thực và nhập mã xác thực sai, click vào button "Xác thực"	Hiển thị thông báo sai mã xác thực
3	Mật khẩu mới không đủ độ dài	Click vào textbox mật khẩu mới và nhập mật khẩu dưới 6 kí tự, click vào button "Đặt lại"	Hiển thị cảnh báo mật khẩu không đủ độ dài
4	Nhập xác nhận mật khẩu không khớp với mật khẩu mới	Click textbox xác nhận mật khẩu và nhập mật khẩu không khớp với	Hiển thị cảnh báo xác nhận mật khẩu sai

		mật khẩu mới, click vào button "Đặt lại"	
5	Nhập đúng định dạng mật khẩu mới và xác nhận mật khẩu trùng khớp với mật khẩu mới	Click vào textbox mật khẩu mới nhập mật khẩu đúng định dạnh, click vào textbox xác nhận mật khẩu và nhập trùng mật khẩu mới, click vào "Đặt lại"	Cập nhật mật khẩu mới

- **Chức năng Nạp tiền vào hệ thống**

Bảng 6.20. Bảng test case chức năng Nạp tiền vào hệ thống

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Bỏ trống các trường bắt buộc	Click vào textbox email và nhập thông tin, click vào button "Pay"	Hiển thị cảnh báo yêu cầu nhập đầy đủ thông tin
2	Bỏ trống tất cả các trường	Click vào button "Pay"	Hiển thị cảnh báo yêu cầu nhập thông tin
3	Nhập đầy đủ thông tin	Nhập đầy đủ thông tin, click vào button "Pay"	Cộng tiền vào số dư khả dụng của người dùng và tạo một bản ghi giao dịch mới

- **Chức năng Xem lịch sử nạp tiền**

Bảng 6.21. Bảng test case chức năng Xem lịch sử nạp tiền

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Xem lịch sử nạp tiền thành công	Click vào button "Nạp tiền" trên thanh menu trang cá nhân	Hiển thị danh sách lịch sử nạp tiền gồm ngày thực hiện giao dịch, loại giao dịch, số tiền nạp, trạng thái giao dịch.

- **Chức năng Đăng ký tài khoản đối tác**

Bảng 6.22. Bảng test case chức năng Đăng ký tài khoản đối tác

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Nhập email không đúng định dạng abcd@gmail.com	Nhập email sai định dạng, click vào "Tiếp tục"	Cảnh báo nhập email không đúng định dạng
2	Không nhập gì	Không nhập email và click vào "Tiếp tục"	Cảnh báo yêu cầu nhập email để tục
3	Nhập số điện thoại không đúng định dạng bao gồm 10 chữ số	Nhập số điện thoại sai định dạng, click vào "Tiếp theo"	Cảnh báo nhập số điện thoại không đúng định dạng
4	Nhập mật khẩu không đủ độ dài	Nhập mật khẩu dưới 6 kí tự, click "Tạo tài khoản"	Hiện cảnh báo mật khẩu không đủ độ dài

5	Nhập xác nhận mật khẩu không khớp với mật khẩu	Nhập xác nhận mật khẩu không khớp với mật khẩu, click "Tạo tài khoản"	Hiển thị cảnh báo yêu cầu nhập lại xác nhận mật khẩu

- **Chức năng Xem thống kê đặt phòng và doanh thu**

Bảng 6.23. Bảng test case chức năng Xem thống kê đặt phòng và doanh thu

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Xem thống kê đặt phòng và doanh thu thành công	Click vào "Bảng điều khiển" trên thanh menu	Hiển thị tổng đơn đặt phòng, tổng doanh thu, danh sách đơn đặt phòng mới, thống kê số đơn đặt phòng và doanh thu theo tháng

- **Chức năng Quản lý khách sạn**

Bảng 6.24. Bảng test case chức năng Quản lý khách sạn

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Kiểm tra xem danh sách khách sạn	Click nút "Khách sạn" tại trang quản lý	Hiển thị các khách sạn và hành động Sửa, Xóa
2	Kiểm tra Thêm khách sạn	Click "Thêm khách sạn" Nhập các thông tin	Thêm khách sạn thành công

		Click "Thêm khách sạn"	
3	Kiểm tra Sửa khách sạn	Click Sửa tại 1 khách sạn Sửa các thông tin khách sạn Click "Lưu thay đổi"	Sửa thông tin khách sạn thành công
4	Kiểm tra Xóa khách sạn	Click Xóa tại 1 khách sạn Click "Xác nhận" trên thanh cảnh báo	Hiện cảnh báo chắc chắn muốn xóa Xóa khách sạn thành công

- **Chức năng Quản lý loại phòng**

Bảng 6.25. Bảng test case chức năng Quản lý loại phòng

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Kiểm tra xem danh sách loại phòng	Click "Loại phòng" tại trang quản lý	Hiển thị danh sách loại phòng và hành động Sửa, Xóa
2	Kiểm tra Thêm loại phòng	Click "Thêm loại phòng" Nhập các thông tin Click "Thêm loại phòng"	Thêm loại phòng thành công

3	Kiểm tra Sửa loại phòng	Click Sửa tại 1 loại phòng Sửa các thông tin loại phòng Click "Cập nhật"	Sửa thông tin loại phòng thành công
4	Kiểm tra Xóa loại phòng	Click Xóa tại 1 loại phòng Click "Xác nhận" trên thanh cảnh báo	Hiện cảnh báo chắc chắn muốn xóa Xóa loại phòng thành công

- **Chức năng Quản lý phòng**

Bảng 6.26. Bảng test case chức năng Quản lý phòng

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Kiểm tra xem danh sách phòng	Click "Phòng" tại trang quản lý	Hiển thị danh sách phòng và hành động Sửa, Xóa
2	Kiểm tra Thêm phòng	Click "Thêm phòng" Nhập các thông tin Click "Thêm phòng"	Thêm phòng thành công
3	Kiểm tra sửa phòng	Click Sửa tại 1 phòng Sửa các thông tin phòng	Sửa thông tin phòng thành công

		Click "Cập nhật phòng"	
4	Kiểm tra Xóa phòng	Click Xóa tại 1 phòng Click "Xác nhận" trên thanh cảnh báo	Hiện cảnh báo chắc chắn muốn xóa Xóa phòng thành công

- **Chức năng Cập nhật đơn đặt phòng**

Bảng 6.27. Bảng test case chức năng Cập nhật đơn đặt phòng

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Kiểm tra xem danh sách đơn đặt phòng	Click "Đặt chỗ" tại trang quản lý	Hiển thị danh sách đơn đặt phòng hành động Sửa, Xóa
2	Kiểm tra sửa đơn đặt phòng	Click vào biểu tượng sửa tại 1 đơn đặt phòng Sửa các thông tin đơn đặt phòng Click "Lưu"	Sửa thông tin đơn đặt phòng thành công
3	Kiểm tra Xóa đơn đặt phòng	Click Xóa tại 1 đơn đặt phòng Click "Xác nhậ" trên thanh cảnh báo	Hiện cảnh báo chắc chắn muốn xóa Xóa đơn đặt phòng thành công

- **Chức năng Quản lý đánh giá khách sạn**

Bảng 6.28. Bảng test case chức năng Quản lý đánh giá khách sạn

	Trường hợp kiểm thử	**Các bước thực hiện**	**Kết quả mong muốn**
1	Kiểm tra xem danh sách đánh giá khách sạn	Click "Đánh giá" tại trang quản lý	Hiển thị danh sách đánh giá khách sạn và hành động Sửa, Xóa
2	Kiểm tra Thêm đánh giá khách sạn	Click "Thêm đánh giá" Nhập các thông tin Click "Thêm đánh giá"	Thêm đánh giá khách sạn thành công
3	Kiểm tra Sửa đánh giá khách sạn	Click Sửa tại 1 đánh giá khách sạn Sửa các thông tin đánh giá khách sạn Click "Lưu"	Sửa thông tin đánh giá khách sạn thành công
4	Kiểm tra Xóa đánh giá khách sạn	Click Xóa tại 1 đánh giá khách sạn Click "Xác nhận" trên thanh cảnh báo	Hiện cảnh báo chắc chắn muốn xóa Xóa đánh giá khách sạn thành công

6.2.2 Báo cáo kiểm thử

Kết quả kiểm thử sau khi thực thi các test case

Bảng 6.29. Bảng báo cáo kết quả kiểm thử

Số lượng testcase đã chạy	Pass	Faile	Untest	Ghi chú
64	64	0	0	

CHƯƠNG 7. TỔNG KẾT VÀ ĐÁNH GIÁ

7.1 Đánh giá kết quả

Áp dụng được các kiến thức về phân tích thiết kế hệ thống, thiết kế cơ sở dữ liệu vào việc xây dựng website thương mại điện tử.

Tìm hiểu và áp dụng những kỹ năng, kiến thức về NestJS, Next.js, PostgreSql để xây dựng website booking hotels “StayBuddy”.

Website hiện tại có một số các tính năng cơ bản như: Đăng ký, Đăng nhập, Đặt phòng, Xem đơn đặt phòng, Quản lý khách sạn, Quản lý phòng,…

Hệ thống được thiết kế tương đối rõ ràng và chi tiết. Giao diện dễ sử dụng, thân thiện với người dùng. Các usecase và chức năng được giới thiệu, mô tả chi tiết và dễ hình dung.

7.2 Hướng phát triển

Trong thời gian tới, em dự định mở rộng và phát triển các tính năng nâng cao như đề xuất sản phẩm phù hợp với nhu cầu tìm kiếm của người dùng, nhằm cá nhân hóa trải nghiệm khách hàng. Đồng thời, em sẽ tích hợp các công nghệ AI vào hệ thống, hỗ trợ hiệu quả cho việc quản lý khách sạn và nâng cao trải nghiệm người dùng. Ngoài ra, em sẽ nghiên cứu sâu hơn về xu hướng phát triển của các website thương mại điện tử để cải tiến và đáp ứng tốt hơn nhu cầu thị trường.

TÀI LIỆU THAM KHẢO

[1] Giáo trình thiết kế web – Trường đại học Công Nghiệp Hà Nội, 2017.

[2] Nguyễn Thị Thanh Huyền, Giáo trình cơ sở dữ liệu, NXB Giáo Dục, 2010.

[3] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Giáo trình cơ sở dữ liệu, NXB Giáo dục Việt Nam, 2011.

[4] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, Giáo trình phân tích thiết kế hệ thống, NXB Giáo dục VN, 2011.

[5] Thạch Bình Cường, Nguyễn Đức Mận (2008), Kiểm thử và đảm bảo chất lượng phần mềm, NXB Bách Khoa Hà Nội.

[6] Tài liệu JavaScript: https://www.w3schools.com/js/

[7] Tài liệu Typescript: https://www.w3schools.com/typescript/index.php

[8] Tài liệu NestJS : https://docs.nestjs.com/

[9] Tài liệu Next.js : https://nextjs.org/docs

[10] Tài liệu PostgreSql: https://www.w3schools.com/postgresql/index.php