

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP
NGÀNH HỆ THỐNG THÔNG TIN

XÂY DỰNG ỨNG DỤNG ĐẶT LỊCH VÀ HỖ TRỢ KHÁM
BỆNH TỪ XA

GVHD : TS. Nguyễn Ngọc Quang

Sinh viên thực hiện : Lê Ngọc Trường

Mã sinh viên : 2020601391

Lớp : 2020DHHTTT01 – K15

Hà Nội – Năm 2024

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP
NGÀNH HỆ THỐNG THÔNG TIN

XÂY DỰNG ỨNG DỤNG ĐẶT LỊCH VÀ HỖ TRỢ KHÁM
BỆNH TỪ XA

GVHD : TS. Nguyễn Ngọc Quang

Sinh viên thực hiện : Lê Ngọc Trường

Mã sinh viên : 2020601391

Lớp : 2020DHHTTT01 – K15

Hà Nội – Năm 2024

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn đến TS. Nguyễn Ngọc Quang đã nhiệt tình hướng dẫn và giúp đỡ em trong thời gian thực tập tốt nghiệp hoàn thành đề tài “Xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa”. Cảm ơn thầy đã hướng dẫn, góp ý cho em về ý tưởng đề tài, các tài liệu liên quan, kiến thức phân tích thiết kế hệ thống, những kiến thức thực tế và ứng dụng cao.

Đặc biệt trong quá trình thực hiện đề tài thực tập tốt nghiệp, thầy đã cho em có cách phân tích và xây dựng cơ sở dữ liệu, hệ thống đáp ứng nhu cầu thiết yếu cho bài toán của đề tài. Với những buổi thảo luận đầy tâm huyết, em vừa được học kiến thức, vừa sửa được những lỗi mắc phải trong khi thực hiện đề tài. Vì vậy em tăng thêm khả năng phân tích, thiết kế và tư duy để xây dựng hệ thống đạt chuẩn.

Cuốn đồ án này được hoàn thành theo đúng thời gian quy định của nhà trường cũng như của khoa không chỉ là sự nỗ lực của em mà còn sự giúp đỡ, chỉ bảo của thầy hướng dẫn.

Em rất mong nhận được ý kiến đóng góp thầy, cô để em học thêm được nhiều kinh nghiệm và rút ra nhiều bài học sau đồ án này.

Em xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN	i
DANH MỤC BẢNG BIỂU	v
DANH MỤC HÌNH ẢNH	vi
CHƯƠNG 1 TỔNG QUAN VỀ BÀI TOÁN	1
1.1. Tên đề tài.....	1
1.2. Lý do chọn đề tài	1
1.3. Phương pháp nghiên cứu	2
1.4. Mục tiêu nghiên cứu	2
1.5. Nội dung nghiên cứu.....	3
CHƯƠNG 2 KHẢO SÁT THỰC TẾ VÀ CƠ SỞ LÝ THUYẾT	4
2. 1. Khảo sát hệ thống thực tế	4
2.1.1. Tình trạng y tế hiện nay và sự cần thiết của áp dụng công nghệ thông tin vào y tế:.....	4
2.1.2. Ưu nhược điểm của các website, ứng dụng đặt lịch khám bệnh và hỗ trợ khám bệnh từ xa.....	4
2.1.3. Xác định và phân tích các giá trị nghiệp vụ	5
2. 2. Cơ sở lý thuyết.....	7
2.2.1. Giới thiệu về NodeJS	7
2.2.2. Giới thiệu về MySQL	10
2.2.3. Giới thiệu về GetStream	13
2.2.4. Giới thiệu về React Native.....	14
2.2.5. Giới thiệu về Expo	18
2. 3. Kiến thức áp dụng.....	21
CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	22

3.1. Phân tích hệ thống.....	22
3.1.1. Vai trò của người dùng.....	22
3.1.2. Mô tả nghiệp vụ hệ thống	22
3.1.3. Yêu cầu chức năng.....	26
3.1.4. Yêu cầu phi chức năng.....	27
3.2. Thiết kế bảng.....	28
3.2.1. Bảng Users – Accounts	28
3.2.2. Bảng Patients	30
3.2.3. Bảng Doctors	31
3.2.4. Bảng Appointments.....	32
3.2.5. Bảng Package Appointments	33
3.2.6. Bảng Specializations.....	34
3.2.7. Bảng Appointment Contacts	34
3.2.8. Bảng Feedbacks	35
3.2.9. Bảng Patients_Doctors (Doctor Favorites).....	35
3.3. Đặc tả Use Case	36
3.3.1. Use Case Đăng nhập	37
3.3.2. Use case đăng nhập bằng Google	40
3.3.3. Use case Đăng ký.....	42
3.3.4. Use case xem danh sách bác sĩ yêu thích	45
3.3.5. Use Case Xem Top bác sĩ	49
3.3.6. Use case Xem bác sĩ theo chuyên ngành.....	51
3.3.7. Use case yêu thích bác sĩ	55
3.3.8. Use case Tìm kiếm bác sĩ	57
3.3.9. Use case Đặt lịch khám bệnh.....	60

3.3.10. Use case Nhắn tin với bác sĩ.....	64
3.3.11. Use case Đánh giá dịch vụ.....	67
3.3.12. Use case Call với bác sĩ	69
3.3.13. Use case Quản lý danh sách lịch hẹn (patient)	72
3.3.14. Use case Xem lịch sử nhắn tin khám bệnh (patient)	76
3.3.15. Use case Quản lý thông tin cá nhân (patient)	79
3.3.16. Use case Quản lý thông tin cá nhân (doctor).....	82
3.3.17. Use case Quản lý danh sách lịch hẹn (doctor).....	85
3.3.18. Use case Nhắn tin với bệnh nhân.....	88
3.3.19. Use case Xem phản hồi đánh giá.....	91
3.3.20. Use case Call với bệnh nhân.....	93
3.3.21. Use case Quản lý bệnh nhân	97
3.3.22. Use case Quản lý bác sĩ	100
3.3.23. Use case Quản lý lịch hẹn khám bệnh.....	103
3.3.24. Use case Thống kê báo cáo.....	106
3.3.25. Use case Quản lý chuyên ngành	109
CHƯƠNG 4 GIAO DIỆN CHƯƠNG TRÌNH	113
4.1. Giao diện chung	113
4.2. Giao diện Admin	133
KẾT LUẬN.....	136
TÀI LIỆU THAM KHẢO	138

DANH MỤC BẢNG BIỂU

Bảng 3. 1 Bảng Users - Accounts	28
Bảng 3. 2 Bảng Patients	30
Bảng 3. 3 Bảng Doctors	31
Bảng 3. 4 Bảng Appointments	32
Bảng 3. 5 Bảng Package Appointment	33
Bảng 3. 6 Bảng Specializations	34
Bảng 3. 7 Bảng Appointment Contacts.....	34
Bảng 3. 8 Bảng Feedbacks.....	35
Bảng 3. 9 Bảng Doctor Favorites.....	35

DANH MỤC HÌNH ẢNH

Hình 2.1 Hoạt động của NodeJS.....	9
Hình 2.2 Logo MySQL	10
Hình 2.3 Logo GetStream	13
Hình 2.4 Logo React Native	15
Hình 2.5 Logo Expo.....	19
Hình 3.1 Thiết kế bảng.....	28
Hình 3.2 Biểu đồ usecase tổng quát.....	36
Hình 3.3 Use case chức năng đăng nhập	37
Hình 3.4 Biểu đồ trình tự chức năng Đăng nhập	38
Hình 3.5 Biểu đồ VOPC chức năng Đăng nhập	39
Hình 3.6 Use case chức năng đăng nhập bằng Google.....	40
Hình 3.7 Biểu đồ trình tự chức năng đăng nhập bằng Google	41
Hình 3.8 Biểu đồ VOPC chức năng Đăng nhập bằng Google.....	42
Hình 3.9 Biểu đồ use case chức năng đăng ký	42
Hình 3.10 Biểu đồ trình tự chức năng Đăng ký	44
Hình 3.11 Biểu đồ VOPC chức năng Đăng ký	45
Hình 3.12 Biểu đồ use case xem danh sách bác sĩ yêu thích.....	45
Hình 3.13 Biểu đồ trình tự chức năng Xem danh sách bác sĩ yêu thích...47	47
Hình 3. 14 Biểu đồ VOPC chức năng Xem danh sách bác sĩ yêu thích...48	48
Hình 3. 15 Biểu đồ use case chức năng Xem Top bác sĩ.....	49
Hình 3. 16 Biểu đồ trình tự chức năng Xem top bác sĩ	50
Hình 3. 17 Biểu đồ VOPC chức năng Xem top bác sĩ	51
Hình 3. 18 Biểu đồ use case chức năng xem bác sĩ theo chuyên ngành ..51	51
Hình 3. 19 Biểu đồ trình tự chức năng xem bác sĩ theo chuyên ngành....53	53
Hình 3. 20 Biểu đồ VOPC chức năng xem bác sĩ theo chuyên ngành ..54	54
Hình 3. 21 Biểu đồ usecase chức năng yêu thích bác sĩ.....	55
Hình 3. 22 Biểu đồ trình tự chức năng Yêu thích bác sĩ.....	56
Hình 3. 23 Biểu đồ VOPC chức năng Yêu thích bác sĩ.....	57
Hình 3. 24 Biểu đồ usecase chức năng Tìm kiếm bác sĩ	57

Hình 3. 25 Biểu đồ trình tự chức năng Tìm kiếm bác sĩ	58
Hình 3. 26 Biểu đồ VOPC chức năng Tìm kiếm bác sĩ.....	59
Hình 3. 27 Biểu đồ use case chức năng Đặt lịch khám bệnh	60
Hình 3. 28 Biểu đồ trình tự chức năng Đặt lịch khám bệnh.....	63
Hình 3. 29 Biểu đồ VOPC chức năng Đặt lịch khám bệnh	64
Hình 3. 30 Biểu đồ usecase chức năng Nhắn tin với bác sĩ.....	64
Hình 3. 31 Biểu đồ trình tự chức năng Nhắn tin với bác sĩ	66
Hình 3. 32 Biểu đồ VOPC chức năng Nhắn tin với bác sĩ	67
Hình 3. 33 Biểu đồ usecase chức năng Đánh giá dịch vụ.....	67
Hình 3. 34 Biểu đồ trình tự chức năng Đánh giá dịch vụ.....	69
Hình 3. 35 Biểu đồ VOPC chức năng Đánh giá dịch vụ	69
Hình 3. 36 Biểu đồ use case chức năng Call với bác sĩ.....	70
Hình 3. 37 Biểu đồ trình tự chức năng Call với bác sĩ	71
Hình 3. 38 Biểu đồ VOPC chức năng Call với bác sĩ	72
Hình 3. 39 Biểu đồ use chức năng Quản lý danh sách lịch hẹn (patient).	73
Hình 3. 40 Biểu đồ trình tự chức năng Quản lý danh sách lịch hẹn (pati)	75
Hình 3. 41 Biểu đồ VOPC chức năng Quản lý danh sách lịch hẹn (pati) 76	
Hình 3. 42 Biểu đồ usecase chức năng Xem lịch sử nhắn tin (patient)....	77
Hình 3. 43 Biểu đồ trình tự chức năng Xem lịch sử nhắn tin (patient)	78
Hình 3. 44 Biểu đồ VOPC chức năng Xem lịch sử nhắn tin (patient)	79
Hình 3. 45 Biểu đồ use case chức năng Quản lý thông tin cá nhân (pati) 79	
Hình 3. 46 Biểu đồ trình tự chức năng Quản lý thông tin cá nhân (pati) . 81	
Hình 3. 47 Biểu đồ VOPC chức năng Quản lý thông tin cá nhân (pati) .. 82	
Hình 3. 48 Biểu đồ use case chức năng Quản lý thông tin cá nhân (doc) 82	
Hình 3. 49 Biểu đồ trình tự chức năng Quản lý thông tin cá nhân (doc) . 84	
Hình 3. 50 Biểu đồ VOPC chức năng Quản lý thông tin cá nhân (doc)... 85	
Hình 3. 51 Biểu đồ use case chức năng Quản lý danh sách lịch hẹn	85
Hình 3. 52 Biểu đồ trình tự chức năng Quản lý danh sách lịch hẹn (doc)87	
Hình 3. 53 Biểu đồ VOPC chức năng Quản lý danh sách lịch hẹn (doc). 88	
Hình 3. 54 Biểu đồ use case chức năng Nhắn tin với bệnh nhân	88

Hình 3. 55 Biểu đồ trình tự chức năng Nhắn tin với bệnh nhân.....	90
Hình 3. 56 Biểu đồ VOPC chức năng nhắn tin với bệnh nhân.....	91
Hình 3. 57 Biểu đồ use case Xem phản hồi đánh giá	91
Hình 3. 58 Biểu đồ trình tự chức năng Xem phản hồi đánh giá	93
Hình 3. 59 Biểu đồ VOPC chức năng Xem phản hồi đánh giá	93
Hình 3. 60 Biểu đồ use case chức năng Call với bệnh nhân	93
Hình 3. 61 Biểu đồ trình tự chức năng Call với bệnh nhân	96
Hình 3. 62 Biểu đồ VOPC chức năng Call với bệnh nhân	97
Hình 3. 63 Biểu đồ use case chức năng Quản lý bệnh nhân.....	97
Hình 3. 64 Biểu đồ trình tự chức năng Quản lý bệnh nhân	100
Hình 3. 65 Biểu đồ VOPC chức năng Quản lý bệnh nhân	100
Hình 3. 66 Biểu đồ use case chức năng Quản lý bác sĩ	100
Hình 3. 67 Biểu đồ trình tự chức năng Quản lý bác sĩ	102
Hình 3. 68 Biểu đồ VOPC chức năng Quản lý bác sĩ.....	103
Hình 3. 69 Biểu đồ use case Quản lý lịch hẹn khám bệnh	104
Hình 3. 70 Biểu đồ trình tự chức năng Quản lý lịch hẹn khám bệnh	105
Hình 3. 71 Biểu đồ VOPC chức năng Quản lý lịch hẹn khám bệnh	106
Hình 3. 72 Biểu đồ use case chức năng Thông kê báo cáo	106
Hình 3. 73 Biểu đồ trình tự chức năng Thông kê báo cáo	108
Hình 3. 74 Biểu đồ VOPC chức năng Thông kê báo cáo	109
Hình 3. 75 Biểu đồ use case chức năng Quản lý chuyên ngành.....	109
Hình 3. 76 Biểu đồ trình tự chức năng Quản lý chuyên ngành	111
Hình 3. 77 Biểu đồ VOPC chức năng Quản lý chuyên ngành	112
Hình 4. 1 Giao diện login.....	113
Hình 4. 2 Giao diện chọn tài khoản	113
Hình 4. 3 Giao diện màn Home	114
Hình 4. 4 Giao diện màn danh sách bác sĩ theo chuyên ngành	115
Hình 4. 5 Giao diện màn chi tiết bác sĩ.....	115
Hình 4. 6 Giao diện màn chọn ngày giờ đặt lịch	116
Hình 4. 7 Giao diện màn chọn gói dịch vụ và thời lượng	117

Hình 4. 8 Giao diện màn thông tin bệnh nhân và nhập biểu hiện vấn đề	117
Hình 4. 9 Giao diện màn Review Summary	118
Hình 4. 10 Giao diện thông báo đặt lịch thành công và mail thông báo	119
Hình 4. 11 Mail thông báo cuộc hẹn.....	119
Hình 4. 12 Giao diện màn danh sách lịch hẹn theo status	120
Hình 4. 13 Giao diện lịch hẹn completed	120
Hình 4. 14 Giao diện màn feedback	121
Hình 4. 15 Giao diện thông báo feedback thành công.....	121
Hình 4. 16 Giao diện thông báo call và call	122
Hình 4. 17 Giao diện nhắn tin.....	123
Hình 4. 18 Giao diện xem lại lịch sử trò chuyện	124
Hình 4. 19 Giao diện profile	125
Hình 4. 20 Giao diện chỉnh sửa profile.....	125
Hình 4. 21 Giao diện tìm kiếm bác sĩ theo tên	126
Hình 4. 22 Giao diện xem danh sách bác sĩ yêu thích	127
Hình 4. 23 Giao diện thêm bác sĩ vào danh sách yêu thích.....	128
Hình 4. 24 Giao diện upload avatar	129
Hình 4. 25 Giao diện đăng nhập bằng email / password	129
Hình 4. 26 Giao diện màn tạo tài khoản với role bệnh nhân	130
Hình 4. 27 Giao diện đăng ký tài khoản bác sĩ.....	131
Hình 4. 28 Giao diện màn xác thực mã xác thực.....	131
Hình 4. 29 Giao diện quản lý lịch hẹn của bác sĩ	131
Hình 4. 30 Giao diện nhắn tin và quản lý thông tin cá nhân của bác sĩ .	132
Hình 4. 31 Mail thông báo có lịch hẹn của bác sĩ.....	132
Hình 4. 32 Giao diện dashboard	133
Hình 4. 33 Giao diện quản lý bác sĩ.....	133
Hình 4. 34 Giao diện quản lý bệnh nhân	134
Hình 4. 35 Giao diện quản lý lịch hẹn	134
Hình 4. 36 Giao diện tìm kiếm lịch hẹn theo ngày	134
Hình 4. 37 Giao diện filter lịch theo status	135

Hình 4. 38 Giao diện quản lý chuyên ngành	135
Hình 4. 39 Giao diện thêm chuyên ngành	135

CHƯƠNG 1 TỔNG QUAN VỀ BÀI TOÁN

1.1. Tên đề tài

Xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa

1.2. Lý do chọn đề tài

Trong bối cảnh công nghệ số bùng nổ và dịch bệnh toàn cầu, nhu cầu tiếp cận dịch vụ y tế một cách nhanh chóng, tiện lợi và an toàn ngày càng trở nên cấp thiết.

Xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa là một đề tài hết sức cấp thiết trong bối cảnh hiện nay khi hệ thống y tế Việt Nam đang phải đối mặt với nhiều thách thức. Theo thống kê của Bộ Y tế, mỗi năm có khoảng 800 triệu lượt khám chữa bệnh, trong đó 60% là khám chữa bệnh ngoại trú, dẫn đến tình trạng quá tải bệnh nhân, thời gian chờ đợi kéo dài, ảnh hưởng đến chất lượng dịch vụ và sự hài lòng của người bệnh. Bên cạnh đó, chi phí khám chữa bệnh cao, tiếp cận dịch vụ y tế khó khăn ở vùng sâu vùng xa và dịch bệnh toàn cầu như Covid-19 đã đặt ra nhiều thách thức cho ngành y tế, khiến việc tiếp cận dịch vụ y tế an toàn trở nên khó khăn hơn bao giờ hết.

Trong bối cảnh đó, ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa ra đời như một giải pháp tối ưu, góp phần giải quyết những vấn đề nan giải của ngành y tế hiện nay. Với sự phát triển mạnh mẽ của công nghệ thông tin và sự phổ biến của thiết bị di động, việc xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa trở nên khả thi hơn bao giờ hết. Ứng dụng này giúp người bệnh chủ động hơn trong việc quản lý sức khỏe, tiết kiệm thời gian, chi phí và tiếp cận dịch vụ y tế dễ dàng hơn.

Đồng thời, ứng dụng cũng hỗ trợ cơ sở y tế quản lý bệnh nhân hiệu quả, phân bổ lịch khám hợp lý, giảm tải áp lực cho bác sĩ, nâng cao năng suất làm việc và mở rộng phạm vi tiếp cận bệnh nhân. Với những lợi ích to lớn và tiềm năng phát triển, việc xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa là một giải

pháp thiết thực, góp phần nâng cao chất lượng dịch vụ y tế, mang lại nhiều lợi ích cho người dân và cơ sở y tế.

1.3. Phương pháp nghiên cứu

Khảo sát tình hình thực trạng trong nước hiện nay, thông qua việc nghiên cứu các ứng dụng, website liên quan đến y tế, đặt lịch, ... như Alo Doctor, Doctor Home, VNPT eHealth,... từ đó áp dụng những điểm tốt, phát triển những phần có thể cải tiến, phù hợp với người dùng.

Kết hợp với việc nghiên cứu lý thuyết về thiết kế ứng dụng, bảo mật thông tin, để đưa ra giải pháp cải tiến phù hợp, nâng cao tính năng, hiệu quả và bảo mật cho ứng dụng.

1.4. Mục tiêu nghiên cứu

Ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa được phát triển với mục tiêu hướng đến người dùng, mang lại trải nghiệm thuận tiện, dễ sử dụng và cung cấp đầy đủ thông tin y tế cần thiết. Ứng dụng được thiết kế với những mục tiêu cụ thể:

- Hỗ trợ đặt lịch khám hiệu quả:
 - Cho phép người dùng đặt lịch khám trực tuyến với các bác sĩ chuyên khoa tại các cơ sở y tế liên kết.
 - Hệ thống tự động xác định thời gian khám phù hợp với lịch làm việc của bác sĩ và nhu cầu của người dùng.
 - Quản lý thông tin lịch hẹn, thông báo lịch khám kịp thời cho người dùng.
- Tư vấn y tế trực tuyến:
 - Kết nối người dùng với bác sĩ chuyên khoa qua video call hoặc chat trực tuyến để nhận tư vấn y tế.
 - Hệ thống lưu trữ lịch sử các cuộc tin nhắn tư vấn, giúp người dùng theo dõi quá trình điều trị.

Bên cạnh việc cung cấp đầy đủ thông tin y tế, ứng dụng còn hướng đến việc nâng cao trải nghiệm người dùng bằng cách tối ưu hóa giao diện, tính năng, đảm

bảo ứng dụng thân thiện, dễ sử dụng và phù hợp với nhu cầu của đa số người dùng.

1.5. Nội dung nghiên cứu

- Khảo sát hệ thống:

Khảo sát tình trạng y tế Việt Nam hiện nay, sự cấp thiết của các ứng dụng liên quan đến y tế: đặt lịch khám bệnh, hỗ trợ khám y tế online. Khảo sát các ứng dụng và trang web liên quan đến y tế, đặt lịch nổi bật như: BookCarer, Alo Doctor, Doctor Home, VNPT eHealth từ đó áp dụng để xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa - Online Doctor Appointment.

- Phân tích chức năng hệ thống:

Vẽ và phân tích biểu đồ User Case, biểu đồ tuần tự, biểu đồ lớp.

- Phân tích thiết kế cơ sở dữ liệu:

Phân tích thiết kế các bảng, kiểu dữ liệu lưu trữ, quan hệ giữa các bảng.

- Phân tích thiết kế chương trình.

CHƯƠNG 2 KHẢO SÁT THỰC TẾ VÀ CƠ SỞ LÝ THUYẾT

2. 1. Khảo sát hệ thống thực tế

2.1.1. Tình trạng y tế hiện nay và sự cần thiết của áp dụng công nghệ thông tin vào y tế:

Tình trạng y tế Việt Nam hiện nay đang đối mặt với nhiều thách thức, bao gồm sự chênh lệch lớn trong tiếp cận dịch vụ y tế giữa các vùng miền. Theo thống kê về nhân lực của ngành Y tế, hiện nay trên cả nước có khoảng hơn 35.000 nhân viên y tế, trong số đó lực lượng bác sĩ chiếm trên 96.200 người. Bên cạnh đó, số lượng y tá và hộ lý chiếm hơn 105.000 người, tương ứng 13 y tá, hộ lý/1 vạn dân, chi phí khám chữa bệnh cao và sự gia tăng của bệnh mãn tính không lây nhiễm (1/3 người trưởng thành bị bệnh tim mạch, 5,5% dân số mắc bệnh tiểu đường). Trong bối cảnh đó, sự xuất hiện của các ứng dụng y tế như đặt lịch khám bệnh, hỗ trợ khám y tế online trở nên cấp thiết, giúp người dân dễ dàng tiếp cận dịch vụ y tế chất lượng cao, giảm thiểu chi phí và thời gian.

2.1.2. Ưu nhược điểm của các website, ứng dụng đặt lịch khám bệnh và hỗ trợ khám bệnh từ xa

Các ứng dụng như BookCare, Alo Doctor, Doctor Home và VNPT eHealth đã khẳng định vai trò quan trọng của mình trong việc kết nối người bệnh với bác sĩ, giúp đặt lịch hẹn khám bệnh thuận tiện, nhanh chóng. BookCare nổi bật với hệ thống quản lý lịch hẹn thông minh, Alo Doctor cung cấp dịch vụ tư vấn y tế trực tuyến 24/7, Doctor Home tập trung vào dịch vụ khám chữa bệnh tại nhà, VNPT eHealth chú trọng đến việc ứng dụng công nghệ thông tin vào quản lý y tế.

Tuy nhiên, các ứng dụng này cũng tồn tại những hạn chế như khả năng tích hợp với hệ thống y tế công còn hạn chế, bảo mật thông tin chưa được đảm bảo tuyệt đối, thiếu hụt chuyên gia y tế có chuyên môn cao. Dựa trên những ưu điểm và nhược điểm của các ứng dụng hiện có, ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa - Online Doctor Appointment cần được phát triển theo hướng tích hợp với hệ thống y tế công, bảo mật thông tin, kết nối với các chuyên gia y tế có chuyên

môn cao, nhằm mang đến cho người dân dịch vụ y tế chất lượng, thuận tiện, hiệu quả.

2.1.3. Xác định và phân tích các giá trị nghiệp vụ

Với ứng dụng Online Doctor Appointment sẽ mang lại nhiều giá trị và lợi ích hơn các ứng dụng trước đây.

a. Mang lại giá trị nghiệp vụ:

- Tăng cường hiệu quả hoạt động của cơ sở y tế:
 - Quản lý lịch hẹn hiệu quả: Ứng dụng giúp các cơ sở y tế quản lý lịch hẹn khám bệnh một cách khoa học, giảm thiểu tình trạng quá tải, giúp bệnh nhân đặt lịch dễ dàng và tránh phải chờ đợi lâu.
 - Giảm tải cho bác sĩ: Bác sĩ có thể tập trung vào công việc khám chữa bệnh thay vì dành thời gian cho việc tiếp nhận bệnh nhân, sắp xếp lịch hẹn.
 - Nâng cao hiệu quả tiếp nhận bệnh nhân: Hệ thống quản lý thông tin bệnh nhân, lịch hẹn, hồ sơ y tế được đồng bộ và quản lý tập trung, giúp cơ sở y tế tiếp nhận bệnh nhân hiệu quả hơn.
 - Hỗ trợ quản lý nguồn lực y tế: Cơ sở y tế có thể theo dõi và phân bổ nguồn lực hiệu quả hơn dựa trên dữ liệu về số lượng bệnh nhân, loại bệnh, nhu cầu dịch vụ.
- Nâng cao chất lượng dịch vụ y tế:
 - Cung cấp dịch vụ khám chữa bệnh từ xa: Giúp người bệnh ở vùng sâu vùng xa, người khuyết tật, người bận rộn tiếp cận với bác sĩ chuyên khoa, thuận tiện và tiết kiệm chi phí.
 - Hỗ trợ tư vấn trực tuyến: Giúp người bệnh giải đáp nhanh chóng các thắc mắc về sức khỏe, nâng cao kiến thức về sức khỏe và phòng bệnh.
- Nâng cao sự hài lòng của người bệnh:

- Tiếp cận dịch vụ thuận tiện: Giúp người bệnh đặt lịch khám bệnh, khám bệnh từ xa dễ dàng, thuận tiện, tiết kiệm thời gian và chi phí.
- Cải thiện trải nghiệm bệnh nhân: Cung cấp dịch vụ chăm sóc khách hàng chuyên nghiệp, giải đáp thắc mắc nhanh chóng, theo dõi tiến trình điều trị.
- Tạo môi trường y tế hiện đại: Cung cấp dịch vụ y tế thông minh, ứng dụng công nghệ, mang đến sự chuyên nghiệp và hiện đại cho ngành y tế.
- Thúc đẩy phát triển ngành y tế:
 - Khuyến khích ứng dụng công nghệ thông tin: Hỗ trợ các cơ sở y tế ứng dụng công nghệ thông tin vào hoạt động khám chữa bệnh, nâng cao hiệu quả và chất lượng dịch vụ.
 - Thúc đẩy sự phát triển của các dịch vụ y tế trực tuyến: Tạo động lực cho sự phát triển của các dịch vụ y tế trực tuyến, mở ra nhiều cơ hội kinh doanh mới cho các doanh nghiệp trong lĩnh vực y tế.

b. Mang lại giá trị kinh tế:

- Tăng doanh thu cho cơ sở y tế: Ứng dụng giúp cơ sở y tế thu hút thêm bệnh nhân, tăng doanh thu từ dịch vụ khám chữa bệnh, tư vấn trực tuyến.
- Tăng cường hiệu quả sử dụng nguồn lực: Giúp cơ sở y tế tối ưu hóa việc sử dụng nhân lực, cơ sở vật chất, giảm chi phí hoạt động.
- Mở rộng thị trường: Ứng dụng giúp cơ sở y tế tiếp cận thêm khách hàng ở vùng sâu vùng xa, khách hàng bận rộn, mở rộng thị trường và tăng cường khả năng cạnh tranh.
- Tạo ra cơ hội kinh doanh mới: Ứng dụng có thể kết hợp với các dịch vụ y tế khác như đặt lịch khám chuyên khoa, mua thuốc online, đặt lịch xét nghiệm, tạo ra cơ hội kinh doanh mới.

c. Mang lại giá trị sử dụng:

- Tiết kiệm thời gian và chi phí: Người dùng tiết kiệm thời gian và chi phí đi lại, chờ đợi khám bệnh, tư vấn sức khỏe.
- Nâng cao quyền tự chủ: Người dùng chủ động lựa chọn bác sĩ, lựa chọn thời gian khám bệnh phù hợp với lịch trình cá nhân.
- Cải thiện chất lượng cuộc sống: Giúp người dùng quản lý sức khỏe chủ động, phát hiện sớm các vấn đề sức khỏe, giảm thiểu nguy cơ mắc bệnh.
- Thúc đẩy lối sống tích cực: Khuyến khích người dùng chăm sóc sức khỏe chủ động, nâng cao ý thức về sức khỏe và phòng bệnh.

2. 2. Cơ sở lý thuyết

2.2.1. Giới thiệu về NodeJS

a. NodeJS là gì?

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế.

Đặc điểm chính của NodeJS:

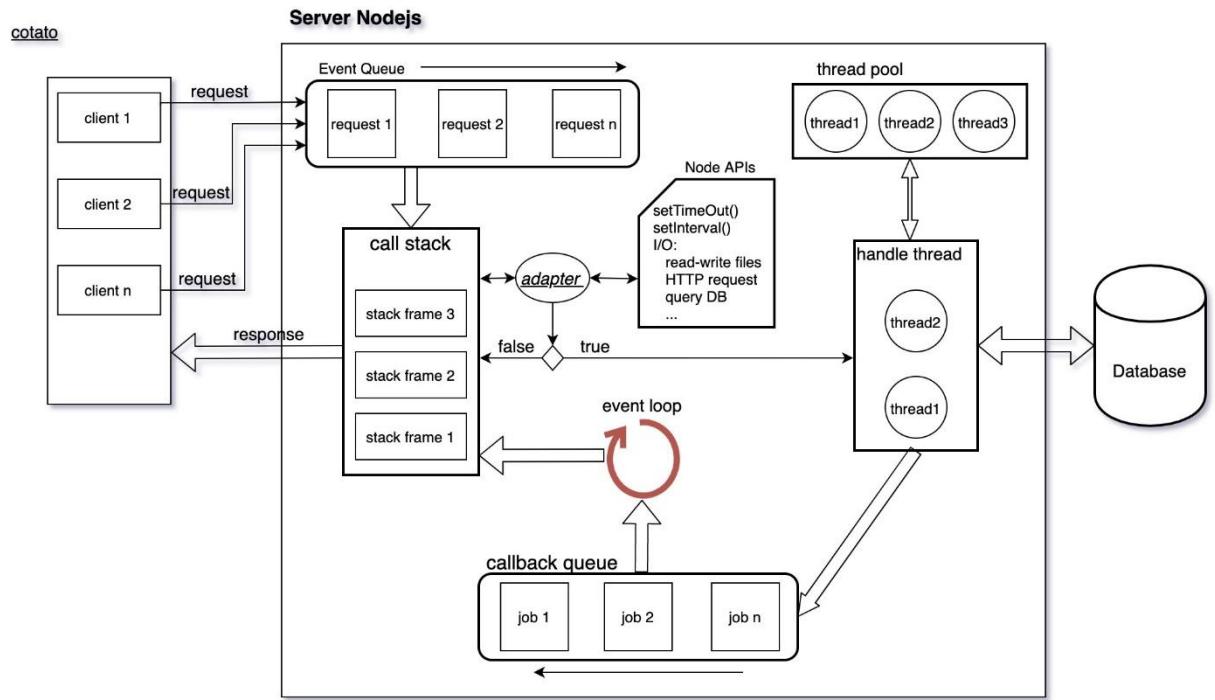
- + Không đồng bộ (Non-blocking): Điều này đề cập đến cách NodeJS xử lý các thao tác I/O mà không chặn luồng thực thi. Thay vì đợi cho mọi thao tác hoàn thành, NodeJS tiếp tục thực hiện các công việc khác và sẽ nhận được thông báo khi thao tác I/O kết thúc. Điều này làm cho ứng dụng có khả năng xử lý nhiều yêu cầu cùng một lúc mà không làm giảm hiệu suất.
- + Chạy rất nhanh: Sự nhanh chóng của NodeJS được đề cập đến việc sử dụng V8 JavaScript Engine, được phát triển bởi Google. V8 biên dịch mã JavaScript thành mã máy nhanh chóng, giúp NodeJS có hiệu suất cao.

- + Đơn luồng nhưng khả năng mở rộng cao: NodeJS sử dụng mô hình đơn luồng với sự kiện lặp, giúp tối ưu hóa việc xử lý đồng thời. Sự kiện lặp cho phép xử lý không chặn của các sự kiện và làm cho NodeJS có khả năng mở rộng tốt, có thể xử lý nhiều yêu cầu đồng thời. Điều này khác biệt so với mô hình đa luồng của một số máy chủ truyền thống.
- + Không đệm: NodeJS không sử dụng bộ đệm cho dữ liệu, điều này có thể giúp giảm độ trễ và làm cho ứng dụng phản ứng nhanh hơn.
- + Có giấy phép MIT: Điều này có nghĩa là NodeJS được phân phối dưới giấy phép MIT License. Giấy phép này cho phép người sử dụng tự do sử dụng, sao chép, sửa đổi và phân phối mã nguồn mở của NodeJS mà không gặp các ràng buộc lớn.

Tóm lại, NodeJS là một nền tảng hiệu suất cao, không đồng bộ, sử dụng đơn luồng, và có khả năng mở rộng tốt, làm cho nó trở thành lựa chọn phổ biến cho xây dựng các ứng dụng web phản ứng nhanh và hiệu quả.

b. NodeJS làm việc như thế nào?

Ý tưởng chính của Node.js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node.js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node.js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.



Hình 2.1 Hoạt động của NodeJS

c. Ứng dụng NodeJS trong Backend:

NodeJS đóng vai trò là server quản lý chính, tương tác với database MySQL để truy vấn dữ liệu:

- + Ứng dụng web phía máy chủ: Node.js cung cấp một cách tiếp cận không đồng bộ và sự kiểm soát sự kiện, làm cho nó phù hợp để xây dựng các ứng dụng web phía máy chủ hiệu suất cao. Bằng cách sử dụng Node.js làm server chính, bạn có thể xử lý các yêu cầu đồng thời mà không phải tạo ra nhiều luồng hoặc tiến trình.
- + API (Application Programming Interface): Với Node.js, bạn có thể xây dựng các API để cung cấp dữ liệu cho các ứng dụng di động, ứng dụng web phía người dùng hoặc bất kỳ ứng dụng nào khác cần truy xuất dữ liệu từ server.
- + WebSocket Servers: Node.js cung cấp hỗ trợ tích hợp cho giao thức WebSocket, cho phép user xây dựng các ứng dụng thời gian thực như trò chuyện trực tiếp, trò chơi trực tuyến và ứng dụng đa phương tiện.
- + Các ứng dụng real-time: Với sự hỗ trợ của WebSocket và thư viện như Socket.IO, Node.js có thể được sử dụng để xây dựng các ứng dụng real-

time như bảng điều khiển real-time, cập nhật dữ liệu real-time và thông báo real-time.

2.2.2. Giới thiệu về MySQL

a. MySQL là gì?

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,..

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc hỗ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...



Hình 2.2 Logo MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ tương đối phổ biến và được sử dụng nhiều. Có thể nhận thấy ở công cụ này rất nhiều ưu điểm, tuy nhiên song song với đó, nó cũng tồn tại một vài nhược điểm nhỏ:

- ***Ưu điểm của MySQL:***

- Sử dụng dễ dàng: MySQL là cơ sở dữ liệu tốc độ cao và ổn định, công cụ này dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp hệ thống lớn các hàm tiện ích.
- Tính bảo mật cao: MySQL phù hợp với các ứng dụng có truy cập cơ sở dữ liệu trên internet vì nó sở hữu nhiều tính năng bảo mật, thậm chí là bảo mật cấp cao.
- Đa tính năng: MySQL có thể hỗ trợ hàng loạt các chức năng SQL từ hệ quản trị cơ sở dữ liệu quan hệ trực tiếp và cả gián tiếp.
- Khả năng mở rộng và mạnh mẽ: Công cụ MySQL có khả năng xử lý khối dữ liệu lớn và có thể mở rộng khi cần thiết.
- Tương thích trên nhiều hệ điều hành: MySQL tương thích để chạy trên nhiều hệ điều hành, như Novell NetWare, Windows * Linux *, nhiều loại UNIX * (như Sun * Solaris *, AIX và DEC * UNIX), OS / 2, FreeBSD *,... MySQL cũng cung cấp phương tiện mà các máy khách có thể chạy trên cùng một máy tính với máy chủ hoặc trên một máy tính khác (giao tiếp qua mạng cục bộ hoặc Internet).
- Cho phép khôi phục: MySQL cho phép các transaction được khôi phục, cam kết và phục hồi sự cố.

- ***Nhược điểm của MySQL:***

- MySQL bị hạn chế dung lượng, cụ thể, khi số bản ghi của người dùng lớn dần, sẽ gây khó khăn cho việc truy xuất dữ liệu, khiến người dùng cần áp dụng nhiều biện pháp để tăng tốc độ chia sẻ dữ liệu như chia tải database ra nhiều server, hoặc tạo cache MySQL.
- So với Microsoft SQL Server hay Oracle, độ bảo mật của MySQL chưa cao bằng. Và quá trình Restore cũng có phần chậm hơn. Tuy nhiên, những nhược điểm này không đáng kể với những hệ quản trị nội dung cỡ trung, bởi chỉ những hệ thống lớn hoặc rất lớn mới phát sinh các yêu cầu phức tạp hơn.

b. Ứng dụng của MySQL trong Hệ thống Đặt Lịch & Khám Bệnh Từ Xa:

- *Lưu trữ thông tin bệnh nhân: MySQL có thể lưu trữ một lượng lớn dữ liệu bệnh nhân, bao gồm thông tin cá nhân, lịch hẹn khám bệnh,...*
- *Quản lý lịch hẹn: MySQL hỗ trợ việc tạo, quản lý và theo dõi các lịch hẹn khám bệnh, bao gồm thông tin về bác sĩ, chuyên khoa, ngày giờ, tình trạng đặt lịch,...*
- *Phân tích dữ liệu: MySQL có thể hỗ trợ phân tích dữ liệu để theo dõi hiệu quả hoạt động của hệ thống, xác định các xu hướng khám bệnh, cải thiện dịch vụ chăm sóc sức khỏe,...*

c. Các điểm mạnh của MySQL khi ứng dụng trong hệ thống

MySQL là một hệ quản trị cơ sở dữ liệu (DBMS) mạnh mẽ và phổ biến, và nó có thể được sử dụng làm cơ sở dữ liệu chính để lưu trữ các thông tin quan trọng của hệ thống với một số điểm mạnh của MySQL khi sử dụng nó như một cơ sở dữ liệu chính:

- + Độ tin cậy cao: được thiết kế để đảm bảo tính toàn vẹn, bảo mật và sẵn sàng của dữ liệu. Nó hỗ trợ các tính năng như ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo rằng các giao dịch được thực hiện một cách đáng tin cậy và an toàn.
- + Phạm vi linh hoạt: MySQL hỗ trợ các loại dữ liệu phong phú và đa dạng, bao gồm chuỗi, số, ngày tháng, mảng, JSON, hình học và nhiều loại dữ liệu khác. Điều này cho phép bạn lưu trữ và truy vấn dữ liệu theo các định dạng phù hợp với yêu cầu của bạn.
- + Hiệu suất và tối ưu hóa: MySQL cung cấp các công cụ và kỹ thuật tối ưu hóa để cải thiện hiệu suất của cơ sở dữ liệu. Nó hỗ trợ các chỉ mục, truy vấn song song, thống kê và các kỹ thuật tối ưu hóa khác để đảm bảo rằng truy vấn và thao tác với dữ liệu diễn ra một cách hiệu quả.
- + Mở rộng và khả năng mở rộng: MySQL hỗ trợ khả năng mở rộng lên đến hàng trăm ngàn kết nối đồng thời và hàng tỷ bản ghi. Nó hỗ trợ phân vùng

dữ liệu, sao lưu và phục hồi, và có thể được cấu hình để xử lý khối lượng dữ liệu lớn và tải công việc cao.

- + Hỗ trợ đa nền tảng: MySQL có sẵn cho nhiều nền tảng, bao gồm Linux, Windows và macOS. Điều này cho phép bạn triển khai MySQL trên các môi trường hệ thống khác nhau một cách dễ dàng.
- + Cộng đồng mạnh mẽ: MySQL có một cộng đồng lớn và nhiều người dùng trên toàn cầu. Điều này đảm bảo rằng bạn có thể tìm thấy tài liệu phong phú, hỗ trợ từ cộng đồng và các công cụ bên thứ ba để làm việc với MySQL.

2.2.3. Giới thiệu về GetStream

a. GetStream là gì?

GetStream là một nền tảng PaaS (Platform as a Service - Nền tảng Dịch vụ) cung cấp các API và SDK cho phép các nhà phát triển xây dựng và tích hợp tính năng chat trực tiếp vào các ứng dụng web và di động của họ. Stream giúp bạn tạo ra những trải nghiệm chat với nhiều tính năng phong phú.



Hình 2.3 Logo GetStream

Các tính năng quan trọng của GetStream:

- + API Feed và SDK:
 - API Feed mạnh mẽ: Cho phép bạn tạo, quản lý và phân phối nội dung luồng thông tin một cách dễ dàng. Bạn có thể tùy chỉnh luồng thông tin theo nhu cầu của mình, chẳng hạn như newsfeed, activity feed, notifications, timelines, messaging, v.v.
 - SDK đa ngôn ngữ: Stream cung cấp SDK cho nhiều ngôn ngữ phổ biến như Python, JavaScript, Ruby, PHP, .NET, Go,... giúp bạn tích hợp API vào ứng dụng của mình một cách dễ dàng.
- + Chat trực tiếp:

- Tính năng chat đa dạng: Hỗ trợ các tính năng chat trực tiếp phong phú như:
 - Hiển thị trạng thái đang gõ
 - Trạng thái trực tuyến của người dùng
 - Hỗ trợ các cuộc hội thoại có chủ đề (threaded conversations)
 - Thông báo đầy
 - Kiểm duyệt tin nhắn và người dùng
 - Phản hồi (reactions)
 - Đánh dấu người dùng và kênh (mentions)
- Dễ dàng tích hợp: API và SDK của Stream được thiết kế để bạn có thể dễ dàng tích hợp tính năng chat trực tiếp vào ứng dụng web và di động của mình.
- + Audio & Video call:
 - Tính năng phong phú: Hỗ trợ các tính năng như:
 - Gọi điện thoại video và âm thanh
 - Chia sẻ màn hình
 - Chuyển đổi giữa âm thanh và video
 - Trạng thái trực tuyến của người dùng
 - Hỗ trợ nhiều người tham gia cuộc gọi
 - Tích hợp với các dịch vụ khác: Có thể tích hợp với các dịch vụ gọi điện thoại khác như Twilio và Vonage.

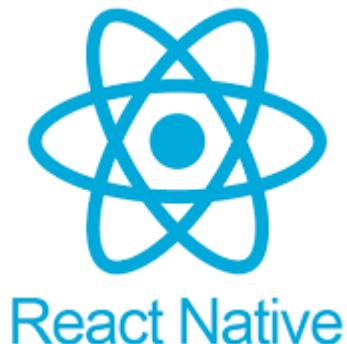
2.2.4. Giới thiệu về React Native

a. React Native là gì?

React Native là một framework phát triển ứng dụng di động sử dụng ngôn ngữ JavaScript. Nó được tạo ra bởi Facebook để xây dựng các ứng dụng di động đa nền tảng với giao diện người dùng tương tự như ứng dụng native.

React Native cho phép lập trình viên sử dụng cùng một codebase để phát triển ứng dụng cho cả hệ điều hành iOS và Android, giúp tiết kiệm thời gian và công

sức. Nó sử dụng các thành phần UI được xây dựng lại từ React, một thư viện JavaScript phổ biến để xây dựng giao diện người dùng web..



Hình 2.4 Logo React Native

Nguyên tắc hoạt động của React Native là gì? Về nguyên tắc hoạt động thì nó gần như giống với React trừ việc React Native hiện tại không thao tác cùng với DOM qua DOM ảo. Nó đang chạy 1 quá trình xử lý nền (Được phiên dịch JavaScript và viết bởi những nhà phát triển) trực tiếp ngay trên thiết bị đầu cuối cũng như giao tiếp cùng với nền tảng góp phần qua một cầu trung gian, theo đợt và bất đồng bộ.

Những thành phần React sẽ bao bọc mã gốc cũng như tương tác cùng với API gốc thông qua mô hình Javascript và UI khai báo của React. Chính điều này đã giúp cho quá trình phát triển ứng dụng dành cho nhiều nền tảng trở nên nhanh hơn. Hiện tại React Native sẽ không dùng HTML mà thay vào đó sẽ sử dụng những thành phần khác nhau vào luồng Javascript.

React Native có nhiều tính năng hữu ích để xây dựng các mobile application và tạo các giao diện người dùng (UI) đẹp và dễ sử dụng:

- + Tạo các thành phần (component) tái sử dụng và dễ sử dụng: React Native cho phép tạo các thành phần (component) tái sử dụng, giúp tách biệt các phần khác nhau của ứng dụng di động. Điều này không chỉ làm cho mã nguồn dễ đọc và bảo trì hơn mà còn tăng tính linh hoạt trong phát triển ứng dụng.

- + Quản lý trạng thái ứng dụng (application state) một cách hiệu quả: React Native cung cấp Redux, một thư viện quản lý trạng thái mạnh mẽ, giúp đơn giản hóa quá trình quản lý và theo dõi các trạng thái khác nhau của ứng dụng. Redux giúp duy trì trạng thái nhất quán và dễ dàng quản lý trong các ứng dụng phức tạp.
- + Tích hợp vào các ứng dụng web hiện có hoặc độc lập: React Native có thể tích hợp vào các ứng dụng di động hiện có hoặc xây dựng các ứng dụng di động mới từ đầu. Với khả năng linh hoạt này, các nhà phát triển có thể chọn cách tiếp cận phù hợp nhất với dự án của mình.
- + Hiệu suất và khả năng tương thích với các thư viện và framework khác: React Native có hiệu suất tốt và tương thích với nhiều thư viện và framework khác, như Redux và MobX. Điều này cho phép các nhà phát triển tận dụng các công cụ và thư viện yêu thích để xây dựng ứng dụng di động mạnh mẽ và hiệu quả.
- + Hỗ trợ điều hướng (routing): React Native hỗ trợ điều hướng cho các ứng dụng di động thông qua React Navigation, một thư viện điều hướng chuyên dụng. Điều này giúp xây dựng các ứng dụng đơn trang (Single-Page Applications) và đa trang một cách dễ dàng và linh hoạt.
- + Cú pháp đơn giản và dễ hiểu: React Native sử dụng cú pháp đơn giản và dễ hiểu của JavaScript và JSX, giúp các nhà phát triển dễ dàng học và sử dụng. Điều này đặc biệt hữu ích cho những người mới bắt đầu hoặc đã quen thuộc với JavaScript và React.

Mặc dù React Native cung cấp cho chúng ta khá nhiều hooks trong toàn bộ vòng đời của 1 component, nhưng thực tế thì chúng ta hầu như chỉ sử dụng một số hooks như sau:

- + useEffect: cho phép thực hiện các hiệu ứng phụ trong functional component. Bạn có thể sử dụng nó để thực hiện các tác vụ như gọi API, đăng ký hoặc huỷ đăng ký sự kiện, hoặc cập nhật DOM.
- + useState: là hook được sử dụng để quản lý trạng thái trong một functional component. Nó trả về một cặp giá trị: giá trị hiện tại của trạng thái và một hàm để cập nhật trạng thái đó
- + useContext: cho phép bạn sử dụng giá trị từ context trong một functional component. Đây là cách để chia sẻ dữ liệu giữa các component mà không cần truyền props qua từng cấp component
- + useReducer: là hook được sử dụng khi bạn cần quản lý trạng thái phức tạp hơn, tương tự như cách hoạt động của Redux. Nó nhận vào một reducer và một trạng thái ban đầu, sau đó trả về trạng thái hiện tại cùng với hàm dispatch để cập nhật trạng thái
- + useRef: tạo ra một tham chiếu có thể giữ một giá trị thay đổi mà không cần render lại component khi giá trị này thay đổi. Thường được sử dụng để truy cập trực tiếp vào các phần tử DOM hoặc giữ lại giá trị giữa các lần render

b. Ưu điểm của React Native và ứng dụng:

React Native là một framework JavaScript mạnh mẽ được phát triển bởi Facebook, cho phép xây dựng các ứng dụng di động hiệu quả cho cả iOS và Android. Đặc biệt trong bối cảnh phát triển các ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa, React Native mang lại nhiều lợi ích vượt trội:

- + **Dễ dàng học và sử dụng:** React Native sử dụng JavaScript và React, là các công nghệ phổ biến với cú pháp đơn giản và dễ hiểu. Điều này giúp các nhà phát triển nhanh chóng học và triển khai ứng dụng. Ngoài ra, tài liệu phong phú và cộng đồng hỗ trợ mạnh mẽ cung cấp nhiều tài nguyên hữu ích cho việc bắt đầu và phát triển dự án.

- + Tốc độ và hiệu suất: Ứng dụng phát triển bằng React Native có hiệu suất cao nhờ việc sử dụng các thành phần UI native. Điều này đảm bảo trải nghiệm người dùng mượt mà và nhanh chóng, một yếu tố quan trọng trong các ứng dụng y tế đòi hỏi phản hồi nhanh và đáng tin cậy.
- + Tính linh hoạt: React Native cho phép sử dụng và tích hợp các thư viện và plug-in khác nhau để mở rộng chức năng của ứng dụng. Bạn có thể dễ dàng kết hợp React Native với các thư viện quản lý trạng thái như Redux, hoặc các công cụ theo dõi sự kiện và quản lý dữ liệu như Firebase.
- + Hỗ trợ cộng đồng tốt: Cộng đồng React Native rất lớn và nhiệt huyết, cung cấp nhiều tài nguyên và hỗ trợ cho người dùng. Từ các diễn đàn, nhóm cộng đồng, đến kho thư viện phong phú trên GitHub, bạn luôn có thể tìm thấy sự trợ giúp khi gặp vấn đề.
- + Khả năng tái sử dụng: React Native cho phép tái sử dụng các thành phần của ứng dụng trên cả iOS và Android, giúp tiết kiệm thời gian và công sức trong phát triển. Điều này đặc biệt hữu ích trong việc triển khai nhanh chóng và duy trì ứng dụng đa nền tảng.
- + Tích hợp tốt với các công cụ khác: React Native tích hợp tốt với các công cụ phổ biến như Redux, Webpack, Babel, và ESLint. Điều này giúp tối ưu hóa quá trình phát triển và đảm bảo chất lượng mã nguồn.

React Native được sử dụng trong *ứng dụng đặt lịch* và hỗ trợ khám bệnh từ xa để xây dựng giao diện người dùng tương tác, đồng thời cập nhật dữ liệu real-time: đặt lịch khám bệnh, Chat Real-time, tương tác với các doctor như like hoặc comment, audio & video call, quản lý thông tin cá nhân, đồ họa và hiệu ứng tương tác, tích hợp với các thư viện và framework frontend khác, ...

2.2.5. Giới thiệu về Expo

- a) *Expo là gì ?*

Expo là một bộ công cụ mã nguồn mở và một nền tảng mạnh mẽ được xây dựng xung quanh React Native, giúp các nhà phát triển dễ dàng tạo và triển khai

các ứng dụng di động cho cả iOS và Android. Expo cung cấp một loạt các tính năng và dịch vụ giúp tối ưu hóa quy trình phát triển ứng dụng di động.



Hình 2.5 Logo Expo

b) *Ưu điểm của Expo*

- Dễ dàng thiết lập và bắt đầu: Expo giúp bạn thiết lập một dự án React Native mới chỉ trong vài phút với một bộ công cụ CLI (Command Line Interface) mạnh mẽ. Bạn không cần phải cài đặt các công cụ và SDK phức tạp để bắt đầu.
- Hot Reloading: Tính năng Hot Reloading cho phép bạn xem các thay đổi trong mã nguồn ngay lập tức mà không cần phải rebuild lại ứng dụng. Điều này giúp tăng tốc quá trình phát triển và thử nghiệm.
- Quản lý tài sản và cấu hình dễ dàng: Expo cung cấp các công cụ quản lý tài sản (hình ảnh, video, font chữ) và cấu hình ứng dụng một cách dễ dàng. Bạn có thể cấu hình các thông tin cần thiết trong file app.json.
- Bộ thư viện phong phú: Expo SDK đi kèm với nhiều API và thư viện tích hợp sẵn, giúp bạn dễ dàng triển khai các tính năng như camera, định vị, thông báo, chia sẻ, và nhiều hơn nữa mà không cần phải cài đặt thêm các thư viện bên ngoài.
- Dịch vụ Expo:
 - Expo Go: Một ứng dụng di động cho phép bạn chạy và kiểm tra ứng dụng của mình ngay trên thiết bị di động mà không cần phải build lại ứng dụng.
 - Expo Build: Dịch vụ build và phân phối ứng dụng của Expo giúp bạn dễ dàng xây dựng các bản phát hành cho cả iOS và Android mà không cần phải cấu hình phức tạp.

- Expo EAS (Expo Application Services): Bao gồm EAS Build, EAS Submit, và EAS Update giúp quản lý quá trình build, submit lên các app store, và cập nhật ứng dụng một cách linh hoạt.
- Tích hợp dễ dàng với các dịch vụ bên ngoài: Expo dễ dàng tích hợp với các dịch vụ bên ngoài như Firebase, Sentry, và các dịch vụ phân tích khác để quản lý và theo dõi ứng dụng của bạn.
- Cộng đồng và tài liệu phong phú: Expo có một cộng đồng lớn và tài liệu chi tiết, cung cấp nhiều hướng dẫn, ví dụ và hỗ trợ từ cộng đồng. Điều này giúp bạn dễ dàng tìm kiếm giải pháp cho các vấn đề gặp phải trong quá trình phát triển.

c) *Hạn chế của Expo*

Mặc dù Expo có nhiều ưu điểm, nhưng cũng có một số hạn chế mà chúng ta cần lưu ý:

- Hạn chế về tùy chỉnh native code: Expo quản lý nhiều phần của dự án, do đó, bạn có thể gặp khó khăn khi cần tích hợp các native module không được hỗ trợ bởi Expo. Tuy nhiên, với tính năng "Eject" (chuyển sang Managed Workflow), bạn có thể chuyển dự án Expo sang một dự án React Native tiêu chuẩn để tùy chỉnh native code.
- Kích thước ứng dụng lớn hơn: Ứng dụng Expo có thể có kích thước lớn hơn do bao gồm nhiều tính năng và thư viện tích hợp sẵn, ngay cả khi bạn không sử dụng tất cả chúng.

d) *Ứng dụng của Expo trong phát triển ứng dụng y tế*

Expo cung cấp nhiều tính năng và dịch vụ hữu ích cho việc phát triển các ứng dụng y tế, đặc biệt là các ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa:

- Quản lý và thông báo lịch hẹn: Sử dụng Expo Notifications để gửi thông báo nhắc nhở người dùng về lịch hẹn khám bệnh.

- Tích hợp video call và chat: Sử dụng Expo để tích hợp với các dịch vụ video call và chat như GetStream, Sendbird, Firebase, Twilio giúp thực hiện các cuộc gọi video và chat trong thời gian thực giữa bác sĩ và bệnh nhân.
- Tạo dự án nhanh chóng.

2. 3. Kiến thức áp dụng

– Để thiết kế được hệ thống em đã sử dụng những kiến thức đã học vào project này:

- + Ngôn ngữ lập trình: Javascript, Typescript.
- + Hệ quản trị cơ sở dữ liệu: MySQL, Firebase.
- + Công cụ: Visual Studio Code, Postman, MySQL Workbench 8.0 CE, Android Studio, Ngrok, Figma,...
- + Framework: NodeJS – NestJS, React Native.
- + Một số công nghệ khác: GetStream, Firebase, AWS S3

– Những công việc chính khi thực hiện đề tài:

- + Khảo sát và nghiên cứu đề tài
- + Tìm kiếm và phân tích dữ liệu
- + Khảo sát sơ bộ
- + Phân tích và thiết lập dự án
- + Xây dựng hệ thống
- + Hoàn thành hệ thống và báo cáo

CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Phân tích hệ thống

3.1.1. Vai trò của người dùng

- Hệ thống chia làm 3 nhóm chính: bệnh nhân (patient), bác sĩ (doctor), người quản trị (admin)
- Mô tả nhóm người dùng:
 - + Bệnh nhân (patient): truy cập ứng dụng, quản lý thông tin cá nhân, xem danh sách bác sĩ yêu thích, tìm kiếm bác sĩ, đặt lịch khám bệnh, nhắn tin với bác sĩ, đánh giá dịch vụ, call với bác sĩ để nhận tư vấn / khám bệnh trực tiếp, xem danh sách lịch hẹn. Khách hàng khi đăng ký tài khoản mới hoặc dùng tài khoản google đồng thời các hoạt động như đặt lịch, lịch sử nhắn tin, đánh giá nhận xét thì sẽ được lưu lại trong cơ sở dữ liệu.
 - + Bác sĩ (doctor): quản lý thông tin cá nhân, nhắn tin với bệnh nhân, xem lịch sử khám của bác sĩ, xem phản hồi đánh giá từ bệnh nhân, call với bệnh nhân.
 - + Người quản trị (admin): quản lý bệnh nhân, quản lý bác sĩ, quản lý lịch hẹn thăm khám bệnh, thống kê và báo cáo

3.1.2. Mô tả nghiệp vụ hệ thống

Hệ thống xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa cần lưu trữ các thông tin sau:

- + Tài khoản (User - Account):
 - id: Mã tài khoản.
 - email: Địa chỉ email.
 - password: Mật khẩu.
 - role: Vai trò (patient, doctor, admin).
 - verified: Xác nhận tài khoản.

- verification code: Mã xác nhận.
- verification expiry: Hạn sử dụng mã xác nhận.
- created at: Thời gian tạo tài khoản.
- updated at: Thời gian cập nhật tài khoản.

Một tài khoản thuộc về một bệnh nhân hoặc một bác sĩ.

+ Bệnh nhân (Patient):

- id: Định danh bệnh nhân.
- name: Tên bệnh nhân.
- date of birth: Ngày sinh.
- gender: Giới tính.
- phone number: Số điện thoại.
- avatar: Ảnh đại diện.
- address: Địa chỉ.
- created at: Thời gian tạo.
- updated at: Thời gian cập nhật.

Mỗi bệnh nhân có 1 và chỉ 1 tài khoản riêng. Mỗi bệnh nhân có thể có 0, 1 hoặc nhiều lịch hẹn. Mỗi bệnh nhân có thể đánh giá 0, 1 hoặc nhiều bác sĩ và có thể thích 0, 1 hoặc nhiều bác sĩ.

+ Bác sĩ (Doctor):

- id: Mã bác sĩ.
- name: Tên bác sĩ.
- description: Mô tả.
- date of birth: Ngày sinh.
- gender: Giới tính.

- phone number: Số điện thoại.
- avatar: Ảnh đại diện.
- address: Địa chỉ.
- hospital: Bệnh viện.
- years experience: Số năm kinh nghiệm.
- start day of week: Ngày bắt đầu làm việc trong tuần.
- time start: Thời gian bắt đầu làm việc trong ngày.
- end day of week: Ngày kết thúc làm việc trong tuần.
- time end: Thời gian kết thúc làm việc trong ngày.
- created at: Thời gian tạo.
- updated at: Thời gian cập nhật.

Mỗi bác sĩ có 1 và chỉ 1 tài khoản và 1 chuyên ngành. Mỗi bác sĩ có thể có 0, 1 hoặc nhiều lịch hẹn và có thể nhận được 0, 1 hoặc nhiều đánh giá từ bệnh nhân.

+ Lịch hẹn (Appointment):

- id: Mã lịch hẹn.
- date: Ngày hẹn.
- status: Trạng thái (upcoming, completed, cancelled).
- description: Mô tả.
- duration: Thời lượng.
- created at: Thời gian tạo.
- updated at: Thời gian cập nhật.

Mỗi lịch hẹn có một bệnh nhân và một bác sĩ, một gói dịch vụ hẹn lịch (package appointment), một danh bạ đặt lịch hẹn (appointment contact) và mỗi lịch hẹn có thể có 0 hoặc 1 đánh giá (feedback).

+ Gói dịch vụ hẹn lịch (Package Appointment):

- id: Định danh gói dịch vụ.
- name: Tên gói dịch vụ.
- price: Giá.
- icon: Biểu tượng.
- description: Mô tả.

Mỗi gói dịch vụ hẹn lịch có thể thuộc về 0, 1 hoặc nhiều lịch hẹn.

+ Chuyên ngành (Specialization):

- id: Mã chuyên ngành.
- name: Tên chuyên ngành.
- description: Mô tả.
- icon: Biểu tượng.
- created at: Thời gian tạo.
- updated at: Thời gian cập nhật.

Mỗi chuyên ngành có thể có 0, 1 hoặc nhiều bác sĩ.

+ Danh bạ đặt lịch hẹn (Appointment Contact):

- id: Mã danh bạ.
- name: Tên cuộc hẹn
- doctor id: Mã bác sĩ.
- patient id: Mã bệnh nhân.

Danh bạ đặt lịch hẹn thuộc về một lịch hẹn.

+ Nhận xét đánh giá (Feedback):

- id: Mã đánh giá.
- rating: Đánh giá.

- comment: Nhận xét.
- date: Ngày đánh giá.

Mỗi đánh giá do 1 bệnh nhân dành cho 1 bác sĩ. Mỗi đánh giá thuộc về 1 lịch hẹn.

3.1.3. Yêu cầu chức năng

- Bệnh nhân (Patient):
 - Xác thực: đăng ký, đăng nhập, đăng nhập bằng google
 - Quản lý thông tin cá nhân
 - Yêu thích bác sĩ
 - Xem danh sách bác sĩ yêu thích
 - Tìm kiếm bác sĩ
 - Đặt lịch khám bệnh
 - Nhắn tin với bác sĩ
 - Đánh giá dịch vụ - bác sĩ
 - Call với bác sĩ để nhận tư vấn, khám bệnh trực tiếp
 - Quản lý danh sách lịch hẹn
- Bác sĩ (Doctor):
 - Đăng nhập, đăng ký
 - Quản lý thông tin cá nhân
 - Nhắn tin với bệnh nhân, xem lịch sử khám của bác sĩ
 - Xem phản hồi đánh giá từ bệnh nhân
 - Call với bệnh nhân
 - Quản lý danh sách lịch hẹn

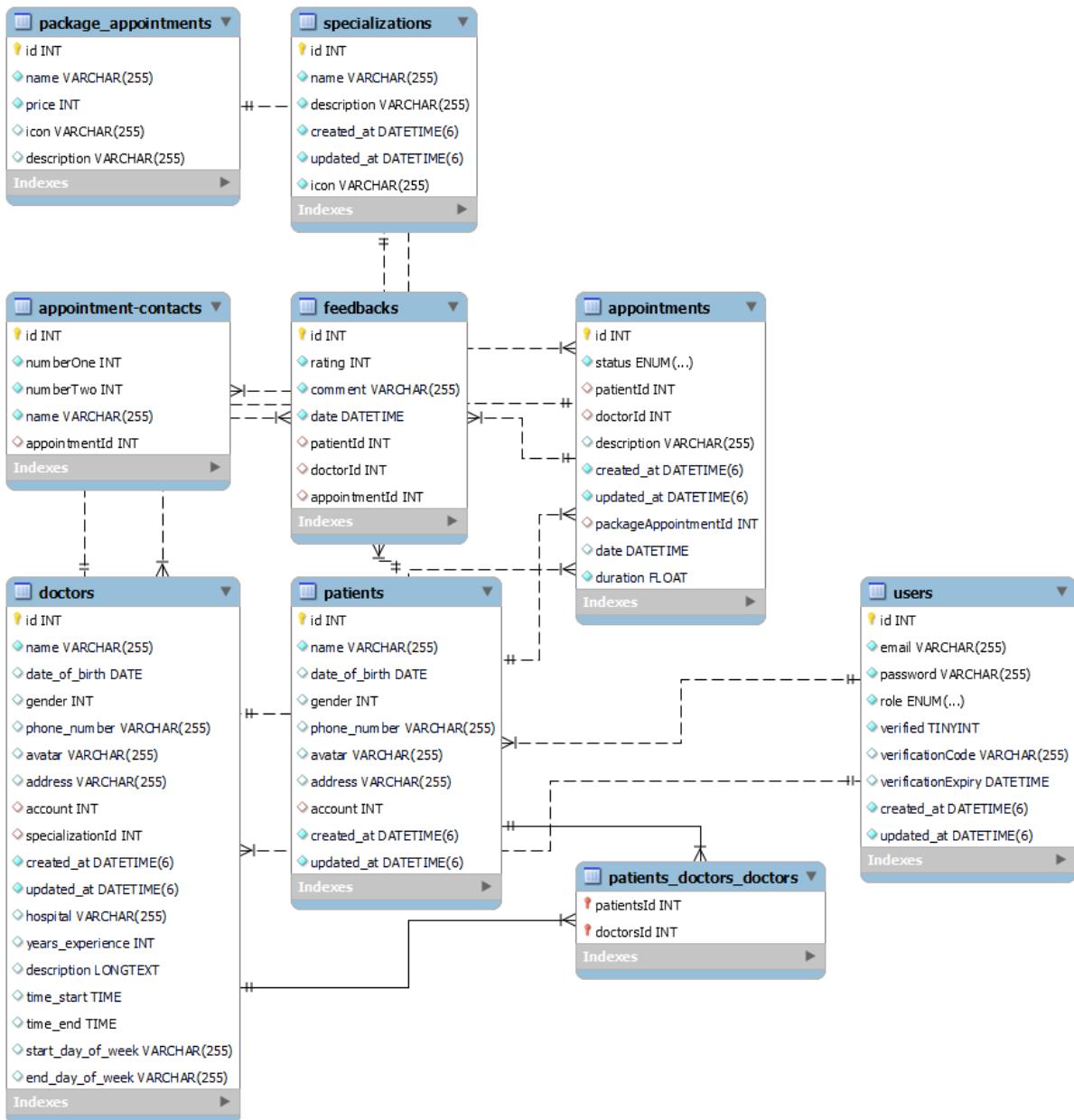
- Quản trị viên (Admin):
 - Quản lý bệnh nhân
 - Đăng nhập
 - Quản lý bác sĩ
 - Quản lý lịch hẹn thăm khám bệnh
 - Thông kê báo cáo

3.1.4. Yêu cầu phi chức năng

- Giao diện ứng dụng được thiết kế dễ sử dụng, có tính thẩm mỹ cao.
- Ứng dụng chạy được trên các thiết bị phổ biến, đa nền tảng: IOS, Android
- Hiệu năng: tốc độ, khả năng và độ tin cậy của hệ thống.

Hệ thống được bảo mật phân quyền để thực hiện các chức năng của hệ thống.

3.2. Thiết kế bảng



Hình 3.1 Thiết kế bảng

3.2.1. Bảng Users – Accounts

Bảng 3. 1 Bảng Users - Accounts

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
-----	----------------	-------	--------------	---------

1	<code>id</code>	Mã tài khoản	<code>int(11)</code>	PK
2	<code>email</code>	Địa chỉ email	<code>varchar(255)</code>	
3	<code>password</code>	Mật khẩu	<code>varchar (255)</code>	
4	<code>role</code>	Vai trò	<code>enum('admin', 'patient', 'doctor')</code>	
5	<code>verified</code>	Xác nhận tài khoản.	<code>tinyint(4)</code>	
6	<code>verification code</code>	Mã xác nhận.	<code>varchar(255)</code>	
8	<code>verification expiry</code>	Hạn sử dụng mã xác nhận.	<code>datetime</code>	
9	<code>created at</code>	Thời gian tạo tài khoản.	<code>timestamp</code>	
10	<code>updated at:</code>	Thời gian cập nhật tài khoản.	<code>timestamp</code>	

3.2.2. *Bảng Patients*

Bảng 3. 2 Bảng Patients

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	Id	Mã bệnh nhân.	int(11)	PK
2	Name	Tên bệnh nhân.	varchar(255)	
3	date of birth	Ngày sinh.	date	
4	Gender	Giới tính.	int(11)	
5	phone number	Số điện thoại.	varchar(255)	
6	avatar.	Ảnh đại diện.	varchar(255)	
7	address	Địa chỉ.	varchar(255)	
8	created at	Thời gian tạo.	timestamp	
9	updated at	Thời gian cập nhật.	timestamp	
10	account	Id tài khoản	int(11)	FK

3.2.3. *Bảng Doctors*

Bảng 3. 3 Bảng Doctors

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	id	Mã bác sĩ.	int(11)	PK
2	name	Tên bác sĩ.	varchar(255)	
3	description	Mô tả.	longtext	
4	date of birth	Ngày sinh.	date	
5	gender	Giới tính.	int(11)	
6	phone number	Số điện thoại.	varchar(255)	
7	avatar	Ảnh đại diện.	varchar(255)	
8	address	Địa chỉ.	varchar(255)	
9	hospital	Bệnh viện.	timestamp	
10	years experience	Số năm kinh nghiệm.	int(11)	
11	start day of week	Ngày bắt đầu làm việc trong tuần.	varchar(255)	
12	time start	Thời gian bắt đầu làm việc trong ngày.	time	

13	end day of week	Ngày kết thúc làm việc trong tuần.	varchar(255)	
14	time end	Thời gian kết thúc làm việc trong ngày.	time	
15	created at	Thời gian tạo.	timestamp	
16	updated at	Thời gian cập nhật.	timestamp	
17	account	Id tài khoản	int(11)	FK
18	specializationId	Id chuyên ngành	int(11)	FK

3.2.4. *Bảng Appointments*

Bảng 3. 4 Bảng Appointments

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	id	Mã lịch hẹn.	int(11)	PK
2	date	Ngày hẹn.	datetime	
3	status	Trạng thái (upcoming, completed, cancelled).	Enum(upcoming, completed, cancelled)	
4	description	Mô tả.	varchar(255)	
5	duration	Thời lượng.	float	

6	created at	Thời gian tạo.	timestamp	
7	updated at	Thời gian cập nhật.	timestamp	
8	patientId	Mã bệnh nhân	int(11)	FK
9	doctorId	Mã bác sĩ	int(11)	FK
10	packageAppointmentId	Mã gói dịch vụ	int(11)	FK

3.2.5. *Bảng Package Appointments*

Bảng 3. 5 Bảng Package Appointment

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	id	Mã gói dịch vụ.	int(11)	PK
2	name	Tên gói dịch vụ.	varchar(255)	
3	price	Giá.	int(11)	
4	icon	Biểu tượng.	varchar(255)	
5	description	Mô tả	varchar(255)	

3.2.6. *Bảng Specializations*

Bảng 3. 6 Bảng Specializations

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	id	Mã chuyên ngành.	int(11)	PK
2	name	Tên chuyên ngành.	varchar(255)	
3	description	Mô tả.	varchar(255)	
4	icon	Biểu tượng.	varchar(255)	
5	created at	Thời gian tạo.	timestamp	
6	updated at	Thời gian cập nhật.	timestamp	

3.2.7. *Bảng Appointment Contacts*

Bảng 3. 7 Bảng Appointment Contacts

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	id	Mã danh bạ.	int(11)	PK
2	name	Tên cuộc hẹn	varchar(255)	
3	doctor id	mã bác sĩ.	int(11)	FK
4	patient id	Mã bệnh nhân.	int(11)	FK

5	appointmentId	Mã lịch hẹn	int(11)	FK
---	---------------	-------------	---------	----

3.2.8. *Bảng Feedbacks*

Bảng 3. 8 Bảng Feedbacks

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	id	Mã đánh giá.	int(11)	PK
2	rating	Đánh giá.	int(11)	
3	comment	Nhận xét.	varchar(255)	
4	date	Ngày đánh giá.	datetime	
5	patientId	Mã bệnh nhân	int(11)	FK
6	doctorId	Mã bác sĩ	int(11)	FK
7	appointmentId	Mã lịch hẹn	int(11)	FK

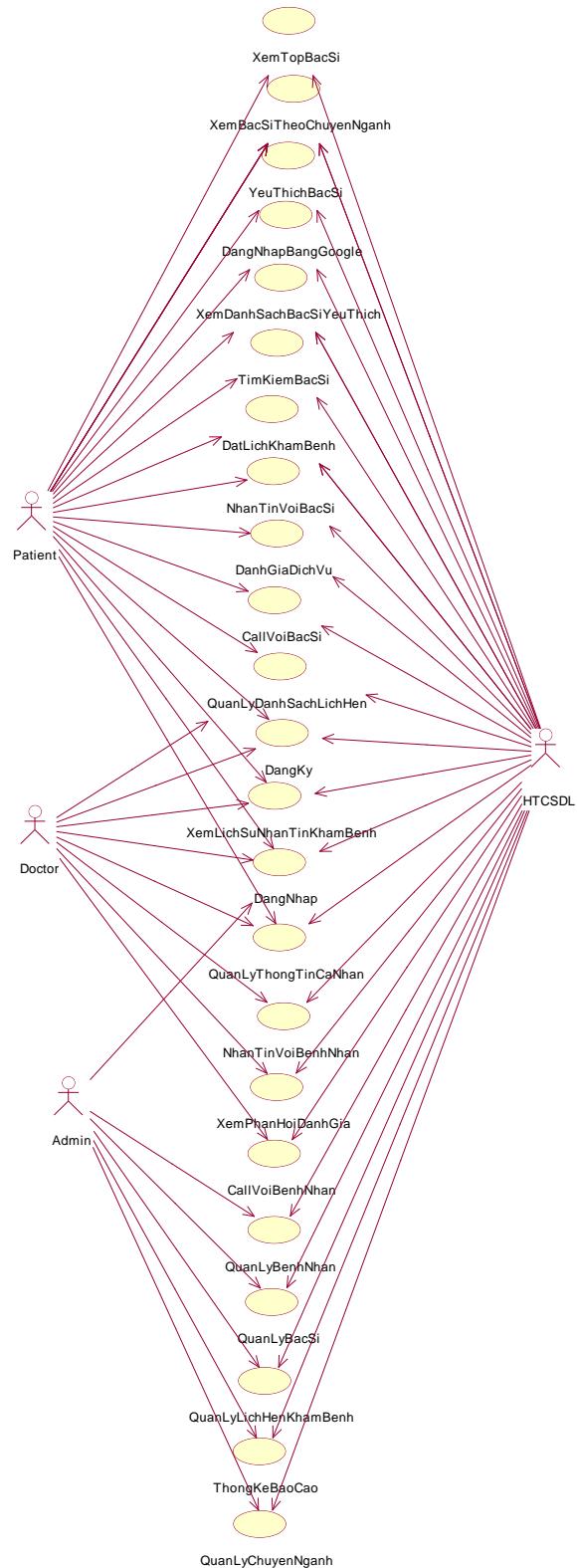
3.2.9. *Bảng Patients_Doctors (Doctor Favorites)*

Bảng 3. 9 Bảng Doctor Favorites

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu	Ghi chú
1	patientId	Mã bệnh nhân	int(11)	PK/FK
2	doctorId	Mã bác sĩ	int(11)	PK/FK

3.3. Đặc tả Use Case

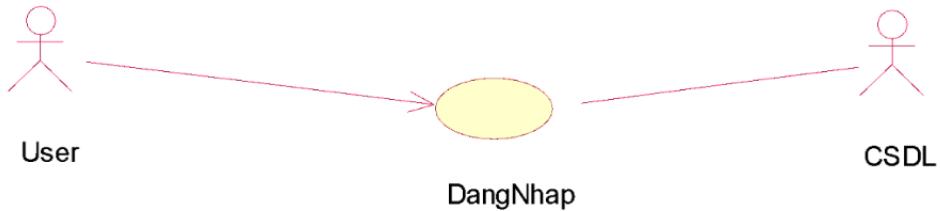
Biểu đồ use case tổng quát :



Hình 3.2 Biểu đồ usecase tổng quát

3.3.1. Use Case Đăng nhập

a. Biểu đồ Use Case:



Hình 3.3 Use case chức năng đăng nhập

b. Đặc tả use case:

1. Tên Use Case

Use Case Đăng nhập.

2. Mô tả văn tắt

Use case này cho phép user tham gia vào hệ thống để sử dụng các chức năng yêu cầu quyền người dùng.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case này bắt đầu khi người dùng chọn “Sign in with password” trên giao diện. Hệ thống sẽ hiển thị màn hình gồm các trường gồm: email, password lên màn hình.

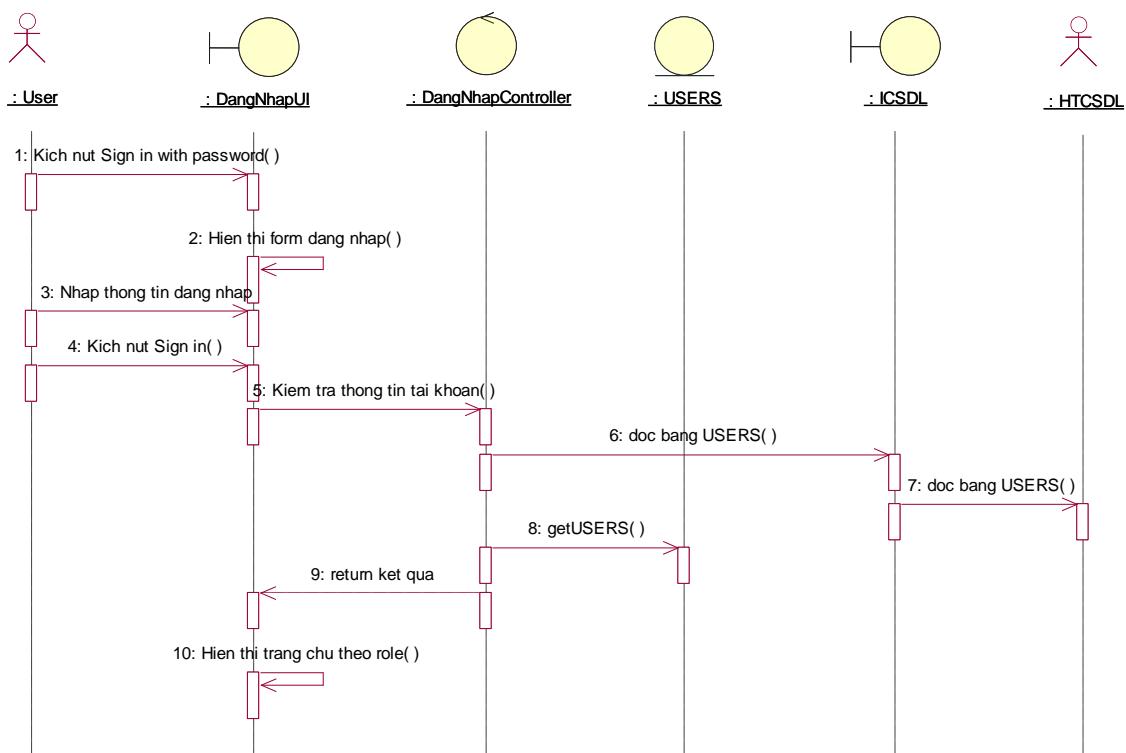
2. Người dùng nhập thông tin và kích nút “Đăng nhập” trong màn hình đăng nhập. Hệ thống truy xuất vào bảng USERS xác thực tài khoản. Nếu thông tin trùng khớp thì hệ thống sẽ lấy thông tin tài khoản và chuyển hướng đến Home theo role của tài khoản.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

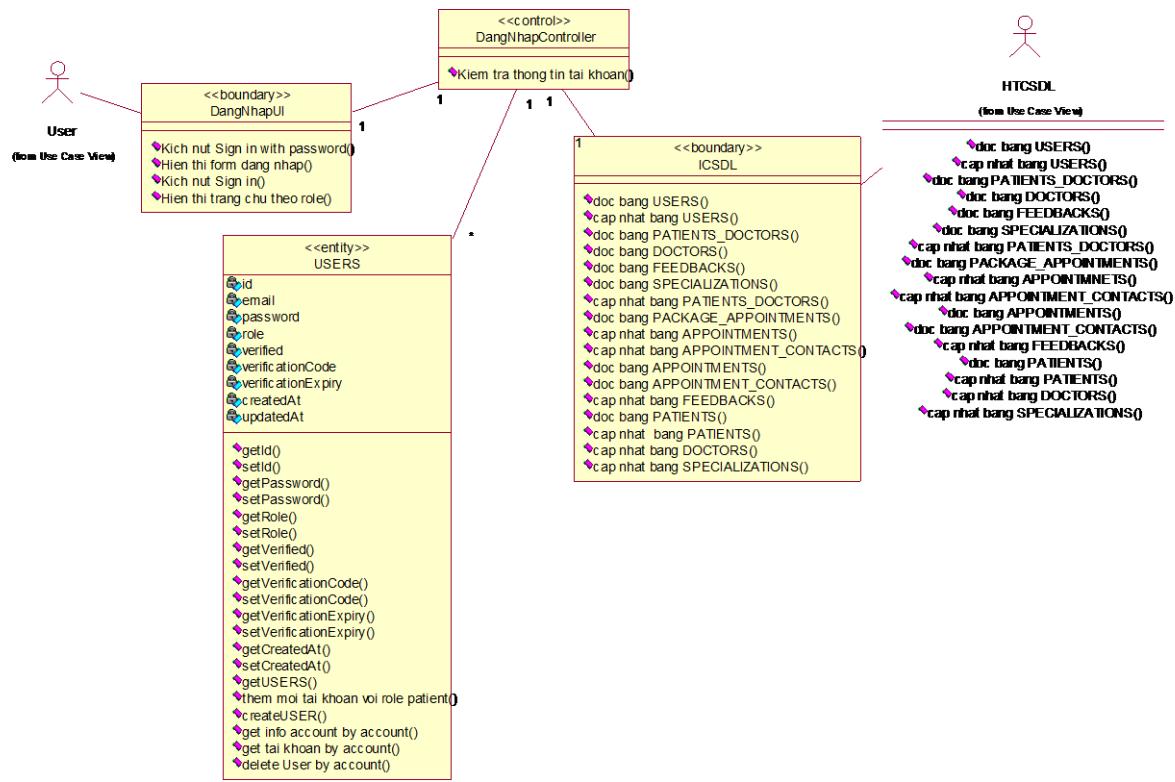
1. Tại bước 3.1.1 trong luồng cơ bản, nếu người dùng nhập tên đăng nhập hoặc mật khẩu sai, hệ thống hiển thị thông báo lỗi. Người dùng có thể nhập lại hoặc bỏ qua thao tác. Use case kết thúc.
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
4. Các yêu cầu đặc biệt: Không có.
5. Tiền điều kiện: User có tài khoản trên hệ thống
6. Hậu điều kiện: Nếu use case thành công, người dùng sẽ đăng nhập được vào hệ thống. Nếu không trạng thái của hệ thống không thay đổi.

c. Biểu đồ trình tự:



Hình 3.4 Biểu đồ trình tự chức năng Đăng nhập

d. Biểu đồ VOPC:



Hình 3.5 Biểu đồ VOPC chức năng Đăng nhập

3.3.2. Use case đăng nhập bằng Google

a. Biểu đồ use case:



Hình 3.6 Use case chức năng đăng nhập bằng Google

b. Đặc tả use case:

1. Tên Use Case: Use Case đăng nhập bằng Google

2. Mô tả văn tắt

Use case này cho phép Bệnh nhân (Patient) dùng tài khoản google để truy cập vào hệ thống và được quyền sử dụng các chức năng yêu cầu quyền bệnh nhân.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case này bắt đầu khi bệnh nhân click vào biểu tượng “Login With Google” trên màn hình đăng nhập. Hệ thống sẽ hiển thị danh sách tài khoản google hiện có.

2. Bệnh nhân chọn tài khoản muốn đăng nhập. Hệ thống sẽ xác thực tài khoản sau đó lấy thông tin tài khoản và hiển thị màn hình Home theo role của tài khoản.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

1. Tại bước 2 trong luồng cơ bản, nếu email không tồn tại trong CSDL thì hệ thống sẽ thêm tài khoản vào bảng USERS sau đó lấy thông tin tài khoản và hiển thị màn hình Home theo role của tài khoản. Use case kết thúc

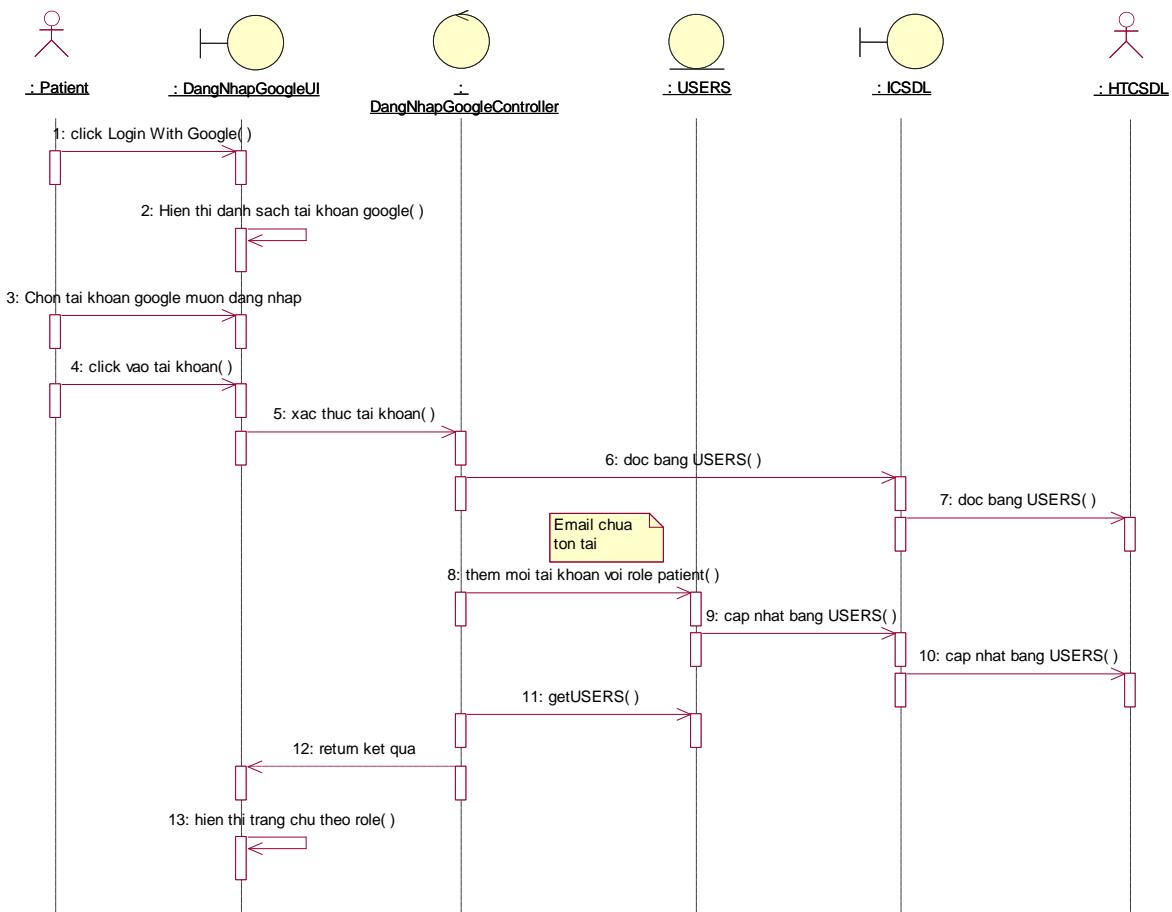
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Không

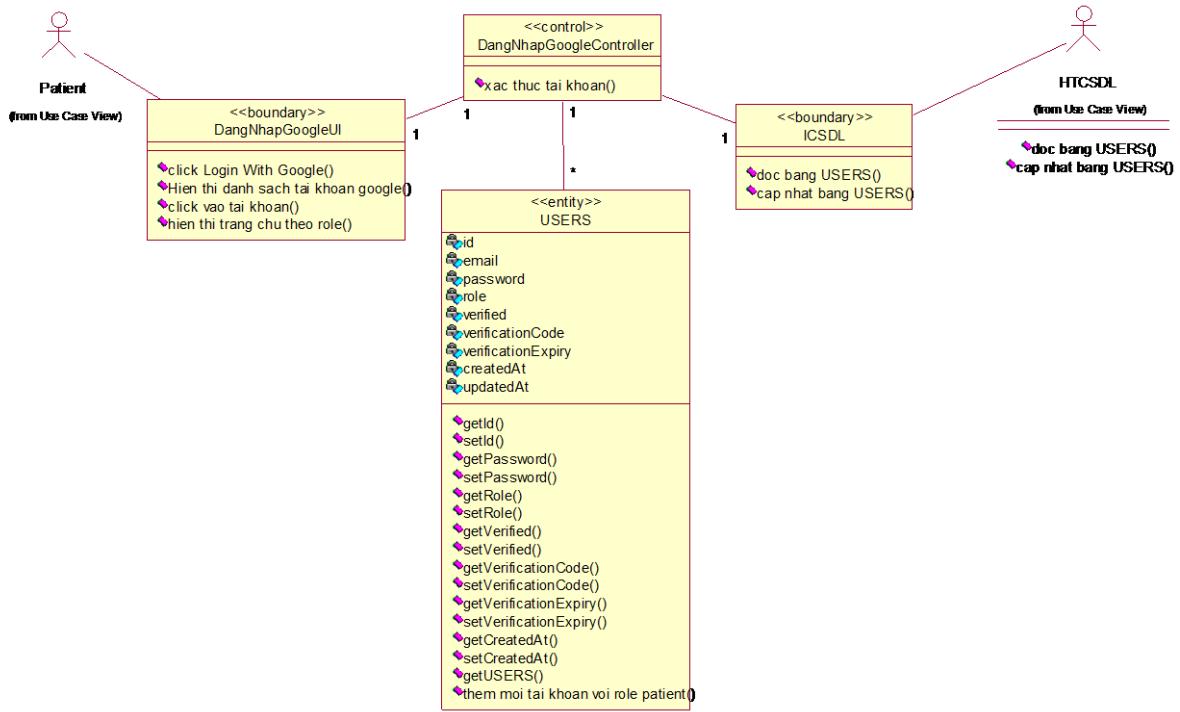
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3.7 Biểu đồ trình tự chức năng đăng nhập bằng Google

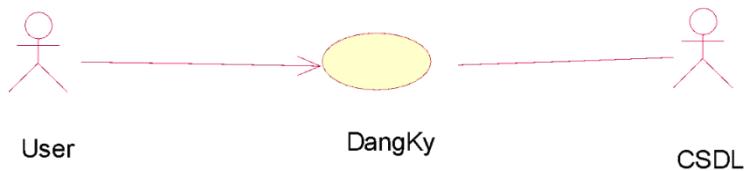
d. Biểu đồ VOPC:



Hình 3.8 Biểu đồ VOPC chức năng Đăng nhập bằng Google

3.3.3. Use case Đăng ký

a. Biểu đồ use case:



Hình 3.9 Biểu đồ use case chức năng đăng ký

b. Đặc tả use case:

1. Tên Use Case: Use Case Đăng ký

2. Mô tả vắn tắt

Use case này cho phép User đăng ký tài khoản để truy cập vào hệ thống và được quyền sử dụng các chức năng yêu cầu quyền người dùng.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case này bắt đầu khi người dùng click vào nút “Sign in with password” ở màn hình Login hệ thống sẽ hiển thị form đăng nhập password, người dùng chọn vào “Sign up”. Hệ thống sẽ hiển thị màn hình đăng ký.

2. Người dùng điền họ tên, email, mật khẩu, role và chọn chuyên ngành nếu là role doctor, sau đó kích hoạt nút “Sign up”. Hệ thống thêm một bản ghi mới vào bảng USERS trong CSDL và hiển thị màn hình Login.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

1. Tại luồng cơ bản nếu user nhập thiếu thông tin thì hệ thống sẽ hiện ra một dòng alert “Bạn chưa điền đủ thông tin. Xin hãy kiểm tra lại.”. Use case kết thúc.

2. Tại luồng cơ bản nếu user nhập Email không đúng định dạng thì hệ thống sẽ hiện ra một dòng alert “Not invalid Email”. Use case kết thúc.

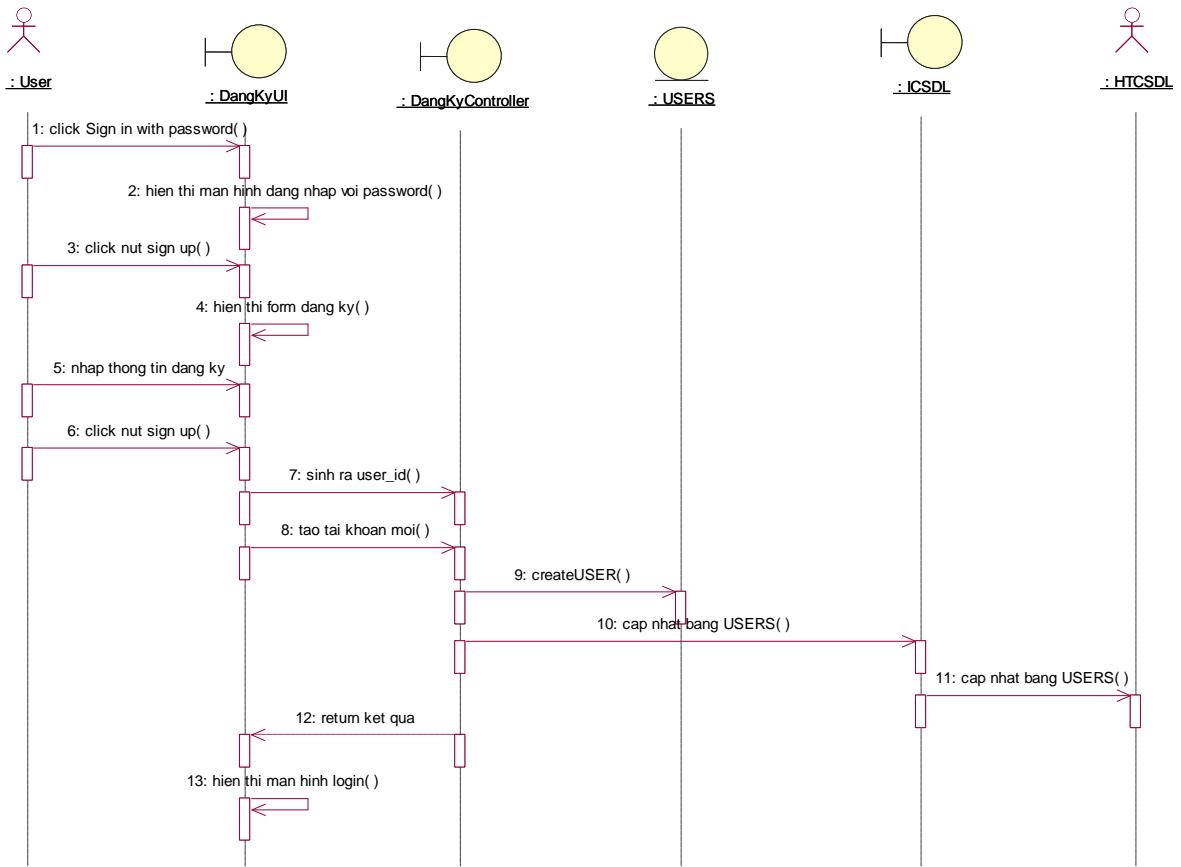
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Không

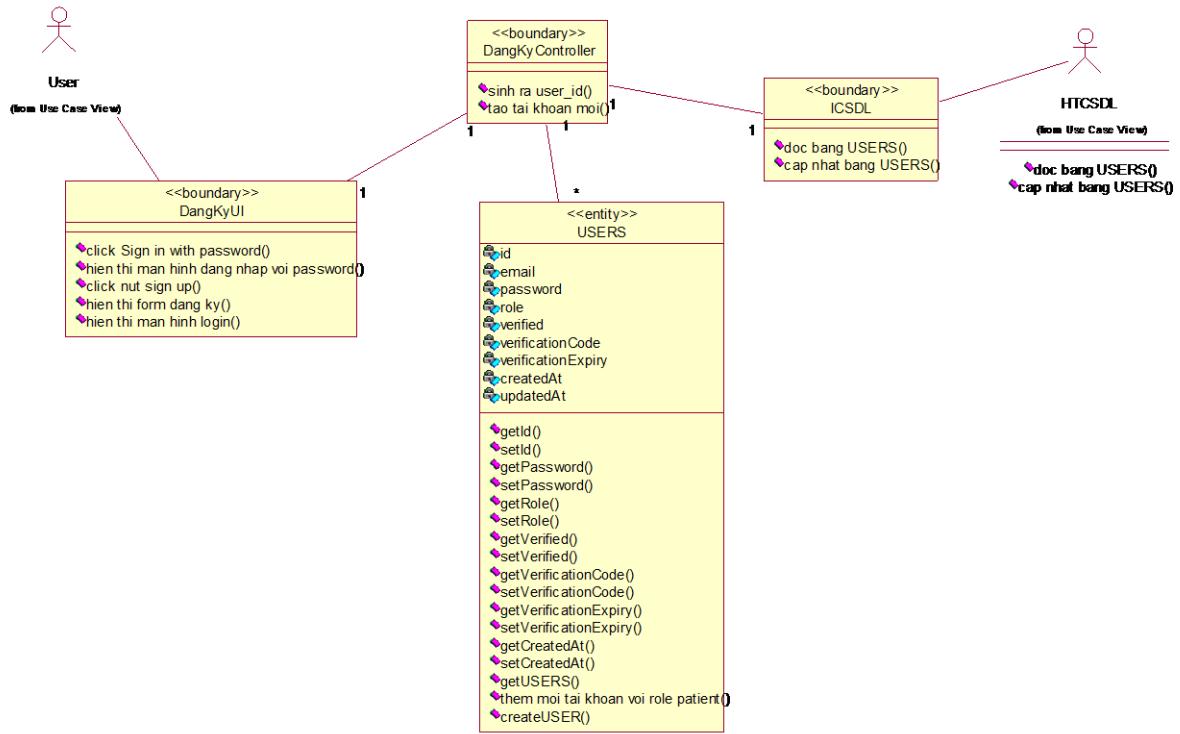
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3.10 Biểu đồ trình tự chức năng Đăng ký

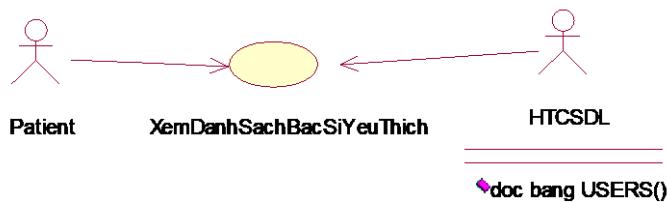
d. Biểu đồ VOPC:



Hình 3.11 Biểu đồ VOPC chức năng Đăng ký

3.3.4. Use case xem danh sách bác sĩ yêu thích

a. Biểu đồ use case:



Hình 3.12 Biểu đồ use case xem danh sách bác sĩ yêu thích

b. Đặc tả use case:

1. Tên Use Case: Xem danh sách bác sĩ yêu thích

2. Mô tả văn tắt

Use case này cho phép bệnh nhân xem danh sách bác sĩ đã được yêu thích của mình trước đó.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case này bắt đầu khi người dùng nhấn vào icon trái tim ở màn hình Home. Hệ thống sẽ lấy thông tin về danh sách bác sĩ yêu thích của bệnh nhân từ bảng DOCTORS và PATIENTS_DOCTORS sau đó chuyển tới màn hình danh sách bác sĩ mà bệnh nhân đã thích trước đó.

2. Người dùng click vào 1 trong số các bác sĩ trong danh sách. Hệ thống sẽ chuyển tới màn hình hiển thị thông tin chi tiết về bác sĩ đó.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

1. Tại bước 1 ở luồng cơ bản, nếu khách chưa có yêu thích bác sĩ nào. Hệ thống sẽ hiển thị màn hình với chữ “Favorite doctors list is empty”. Use case kết thúc.

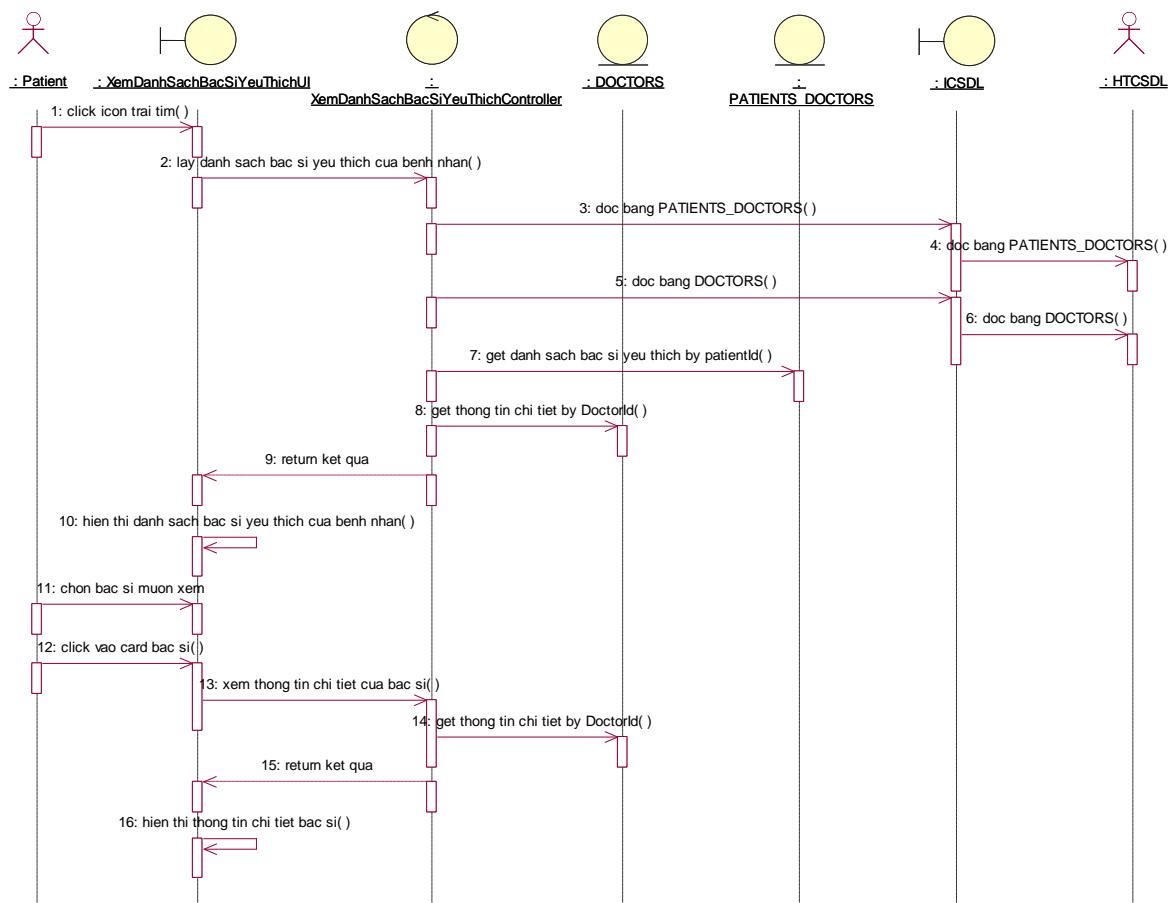
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Người dùng đăng nhập thành công.

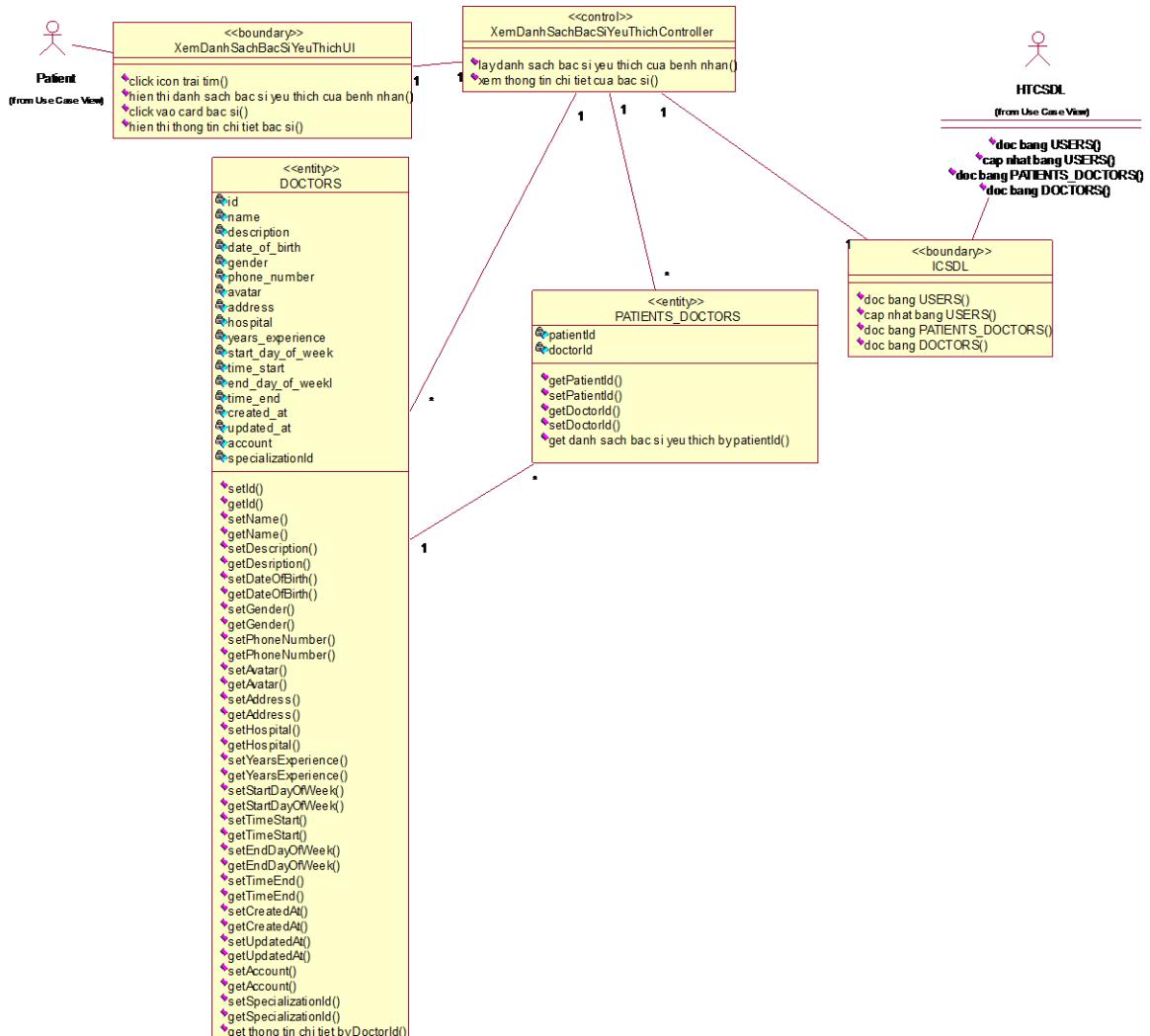
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3.13 Biểu đồ trình tự chức năng Xem danh sách bác sĩ yêu thích

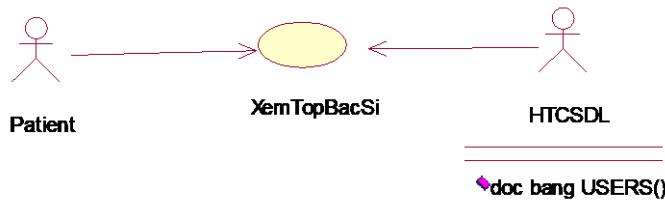
d. Biểu đồ VOPC:



Hình 3. 14 Biểu đồ VOPC chúc năng Xem danh sách bác sĩ yêu thích

3.3.5. Use Case Xem Top bác sĩ

a. Biểu đồ use case:



Hình 3. 15 Biểu đồ use case chức năng Xem Top bác sĩ

b. Đặc tả use case:

1. Tên Use Case: Xem Top bác sĩ

2. Mô tả văn tắt

Use case này cho phép bệnh nhân xem danh sách top bác sĩ theo trung bình rating.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use Case này bắt đầu khi người dùng click vào chữ “See All” tại phần “Top Doctors” ở màn Home. Hệ thống sẽ lấy thông tin danh sách bác sĩ từ bảng DOCTORS và FEEDBACKS sắp xếp từ cao xuống thấp theo trung bình rating và chuyển tới màn hình danh sách top bác sĩ.

2. Người dùng click vào 1 trong số các bác sĩ trong danh sách. Hệ thống sẽ chuyển tới màn hình hiển thị thông tin chi tiết về bác sĩ đó.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

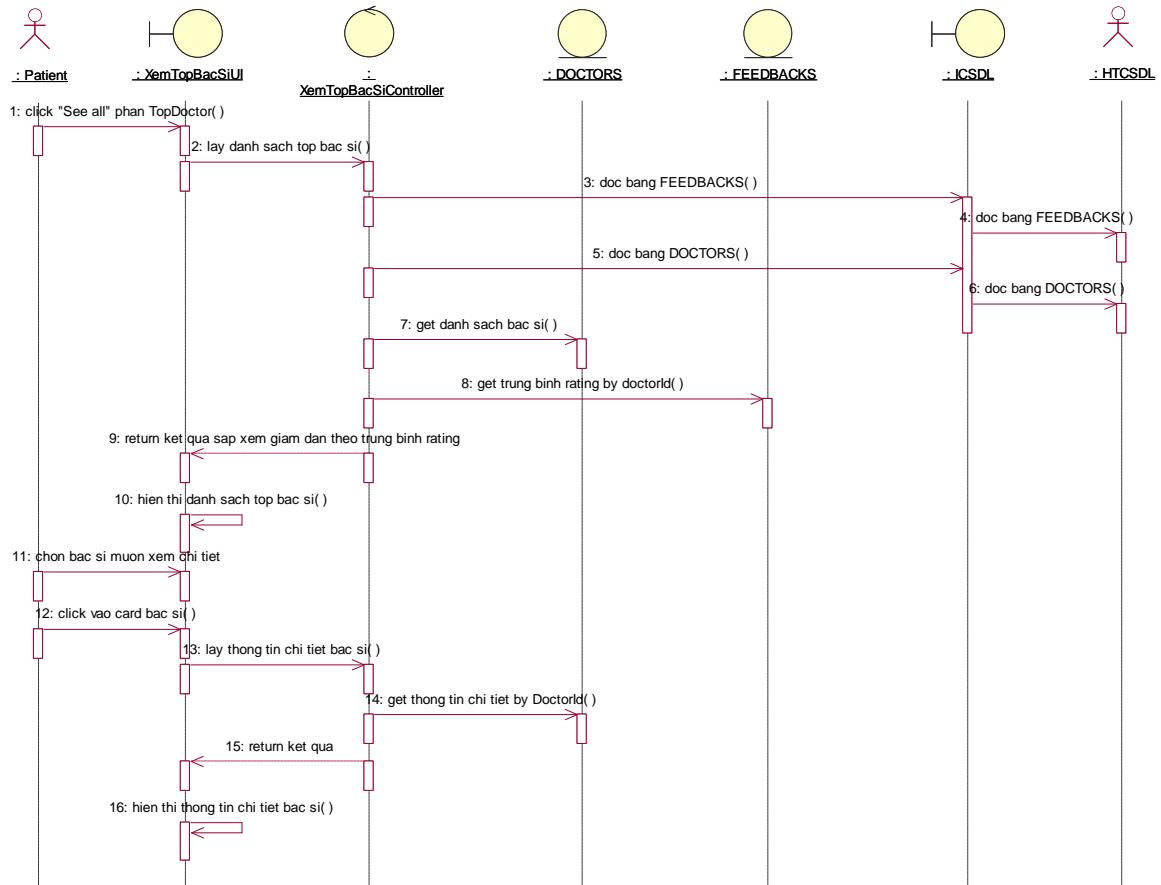
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người dùng đăng nhập thành công.

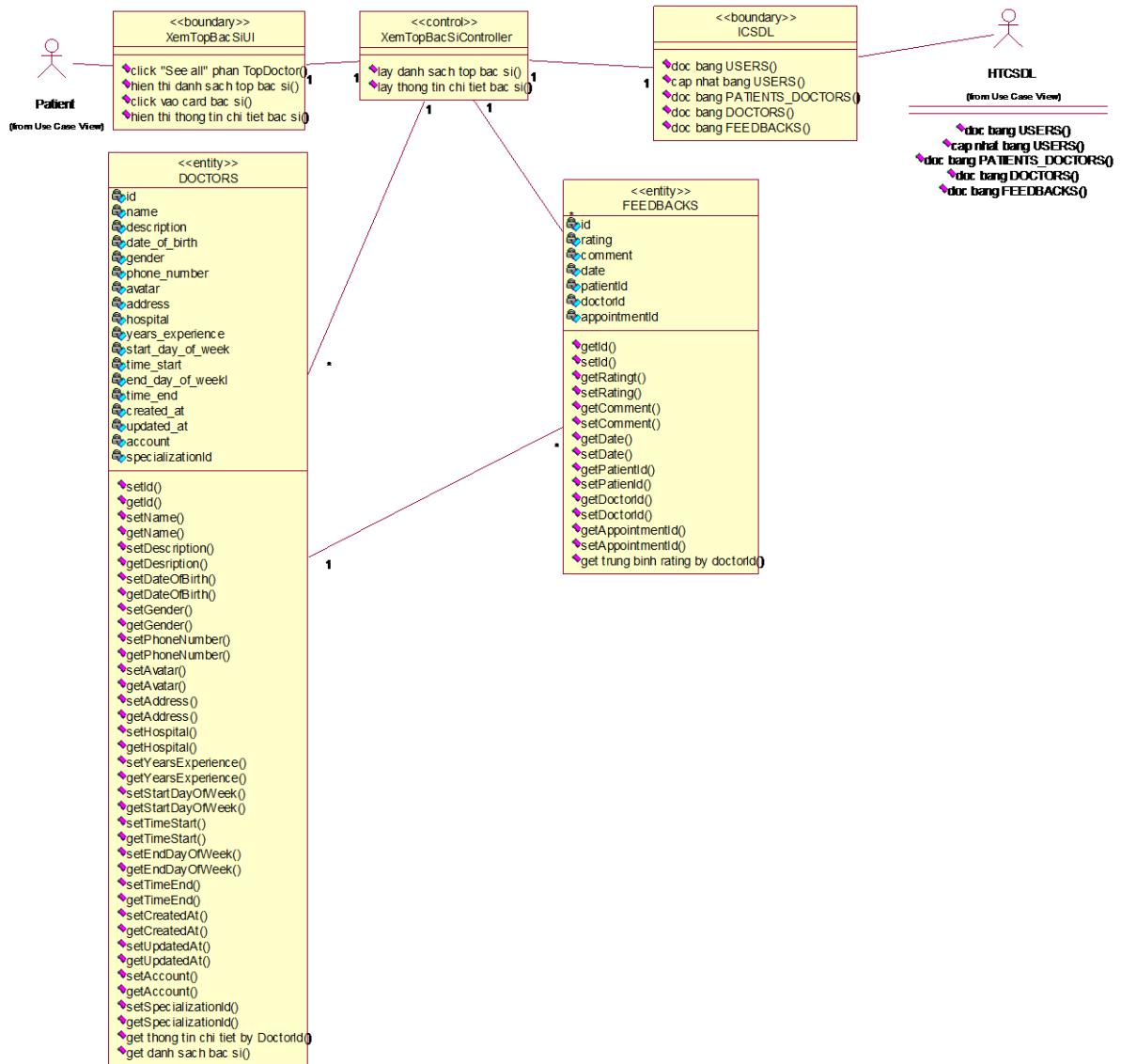
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3. 16 Biểu đồ trình tự chức năng Xem top bác sĩ

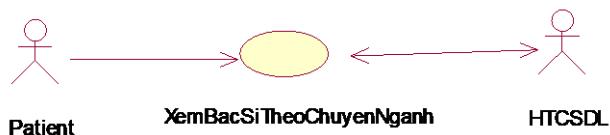
d. Biểu đồ VOPC:



Hình 3. 17 Biểu đồ VOPC chúc năng Xem top bác sĩ

3.3.6. Use case Xem bác sĩ theo chuyên ngành

a. Biểu đồ use case:



Hình 3. 18 Biểu đồ use case chúc năng xem bác sĩ theo chuyên ngành

b. Đặc tả use case:

1. Tên Use Case: Xem bác sĩ theo chuyên ngành

2. Mô tả văn tắt: Use case này cho phép bệnh nhân xem bác sĩ theo chuyên ngành.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case này bắt đầu khi bệnh nhân bấm 1 trong số các icon chuyên ngành tại màn Home. Hệ thống lấy danh sách các bác sĩ từ bảng DOCTORS và SPECIALIZATIONS sau đó chuyển tới màn hình danh sách bác sĩ của chuyên ngành đó.

2. Bệnh nhân click vào 1 trong số các bác sĩ trong danh sách. Hệ thống sẽ chuyển tới màn hình hiển thị thông tin chi tiết về bác sĩ đó.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

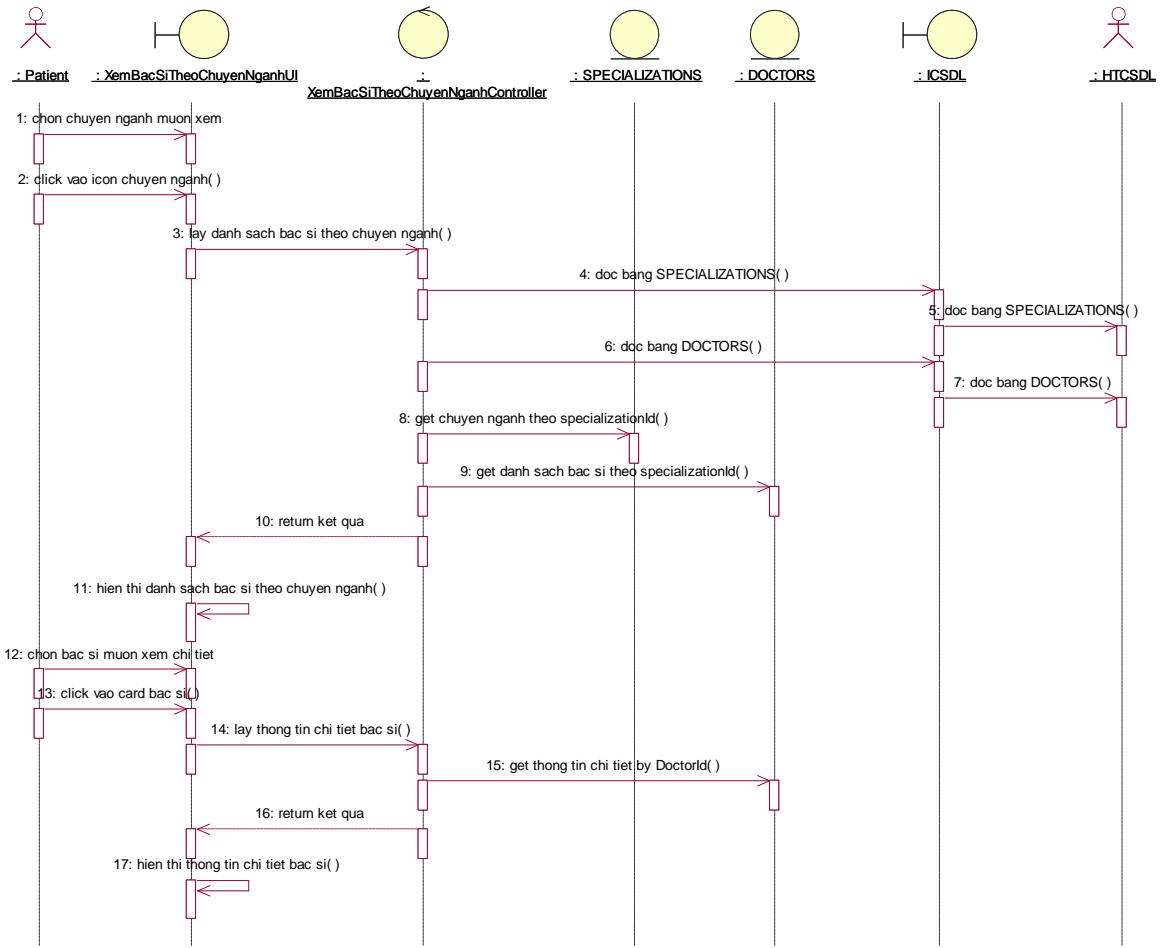
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người dùng đã đăng nhập thành công.

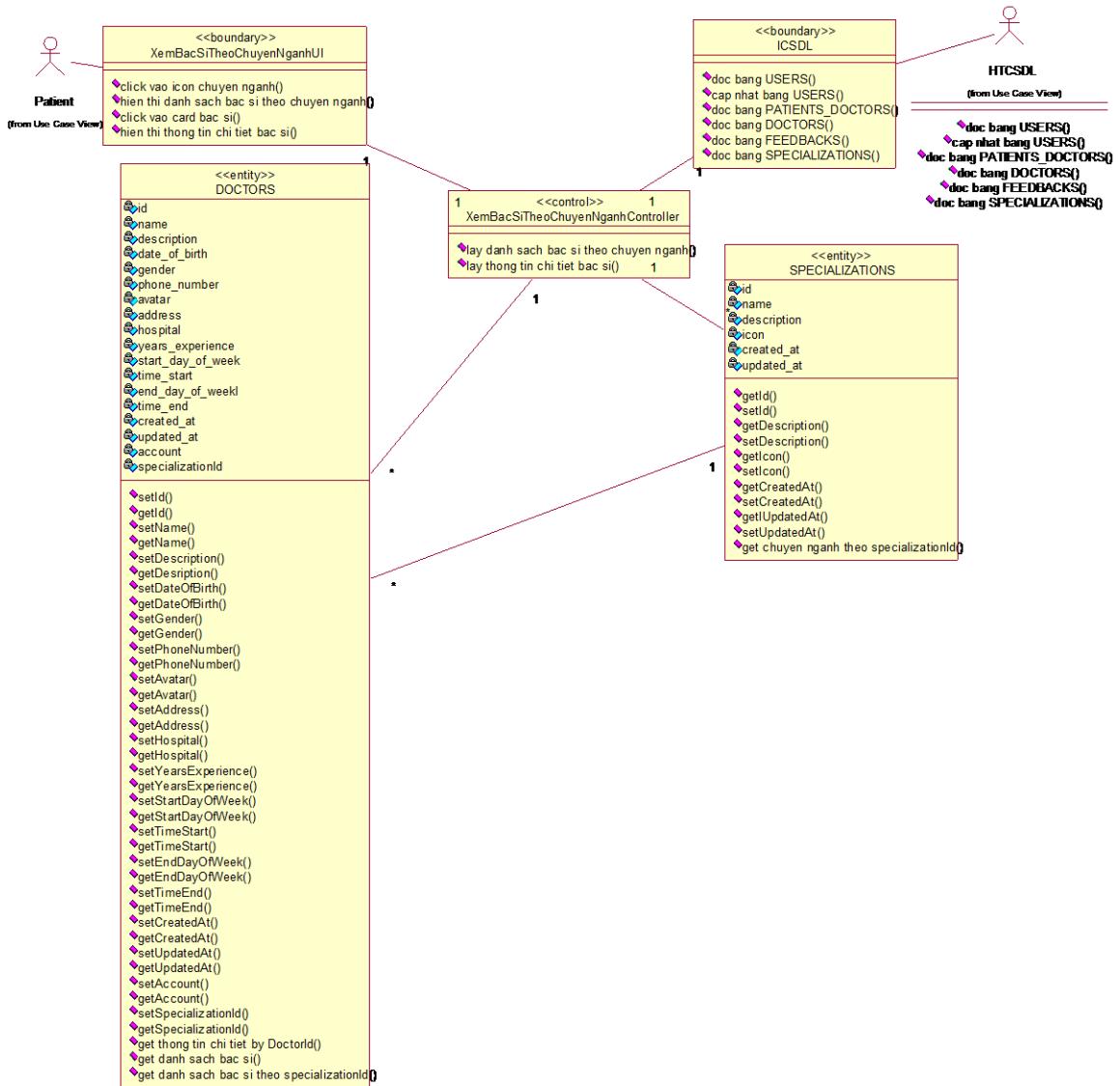
6. Hậu điều kiện: Không có

c. Biểu đồ trình tự:



Hình 3. 19 Biểu đồ trình tự chức năng xem bác sĩ theo chuyên ngành

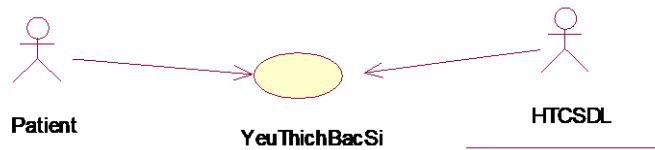
d. Biểu đồ VOPC:



Hình 3. 20 Biểu đồ VOPC chức năng xem bác sĩ theo chuyên ngành

3.3.7. Use case yêu thích bác sĩ

a. Biểu đồ use case:



Hình 3. 21 Biểu đồ usecase chức năng yêu thích bác sĩ

b. Đặc tả use case:

1. Tên Use Case: Yêu thích bác sĩ

2. Mô tả vắn tắt

Use case này cho phép bệnh nhân thêm bác sĩ vào danh sách bác sĩ yêu thích.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case bắt đầu khi bệnh nhân click vào icon trái tim của góc phải trên của mỗi card thông tin bác sĩ tại màn thông tin chi tiết bác sĩ. Hệ thống lấy patientId từ bảng USERS, lấy doctorId từ bảng DOCTORS và thêm dữ liệu patientId, doctorId, date now vào bảng PATIENTS_DOCTORS.

2. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

1. Nếu cặp patientId – doctorId đã tồn tại trong bảng PATIENTS_DOCTORS. Hệ thống sẽ xóa cặp này khỏi bảng

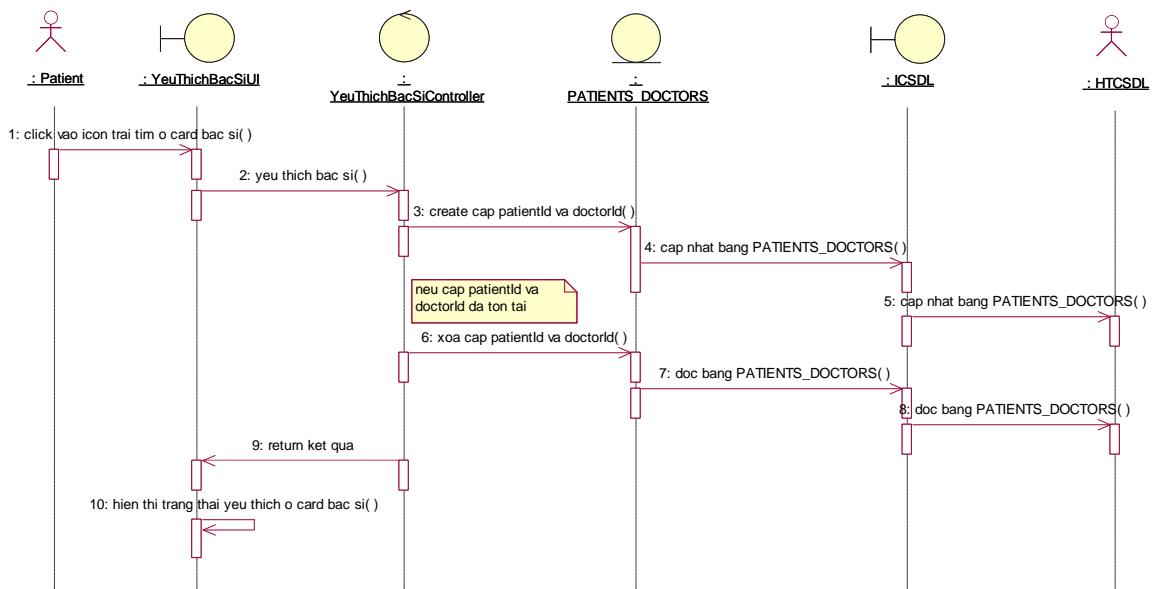
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người dùng đăng nhập thành công.

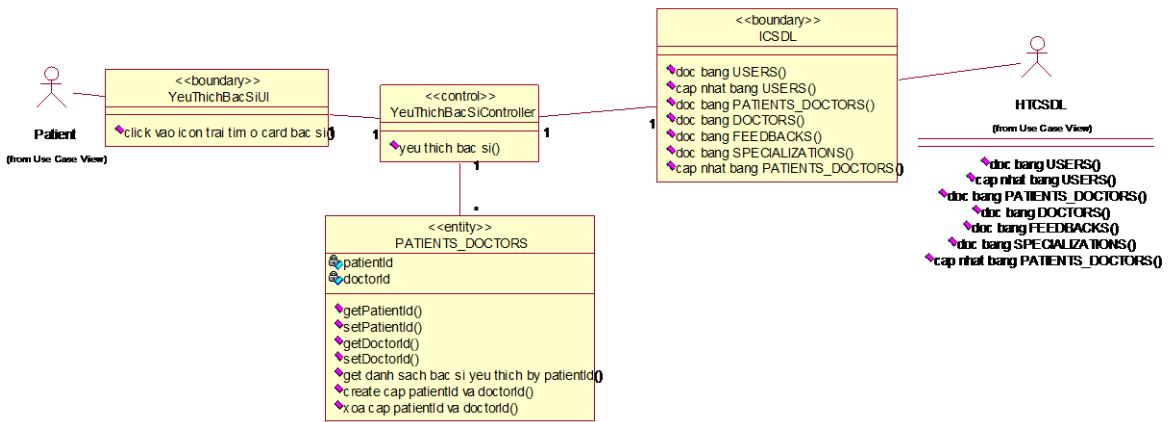
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3. 22 Biểu đồ trình tự chức năng Yêu thích bác sĩ

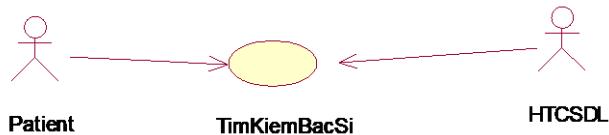
d. Biểu đồ VOPC:



Hình 3. 23 Biểu đồ VOPC chức năng Yêu thích bác sĩ

3.3.8. Use case Tìm kiếm bác sĩ

a. Biểu đồ use case:



Hình 3. 24 Biểu đồ usecase chúc năng Tìm kiếm bác sĩ

b. Đặc tả use case:

1. Tên Use Case: Tìm kiếm bác sĩ

2. Mô tả văn tắt

Use case này cho phép bệnh nhân tìm kiếm bác sĩ theo tên.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case bắt đầu khi bệnh nhân click vào ô tìm kiếm ở màn Home. Hệ thống sẽ chuyển tới màn tìm kiếm bác sĩ.

2. Bệnh nhân nhập tên bác sĩ muốn tìm kiếm và ấn enter hoặc button tìm kiếm. Hệ thống sẽ tìm kiếm bác sĩ theo tên từ bảng DOCTORS và hiển thị danh sách bác sĩ có cùng tên hoặc gần giống với tên bệnh nhân muốn tìm kiếm.

3. Use case kết thúc

3.2. Các luồng rẽ nhánh:

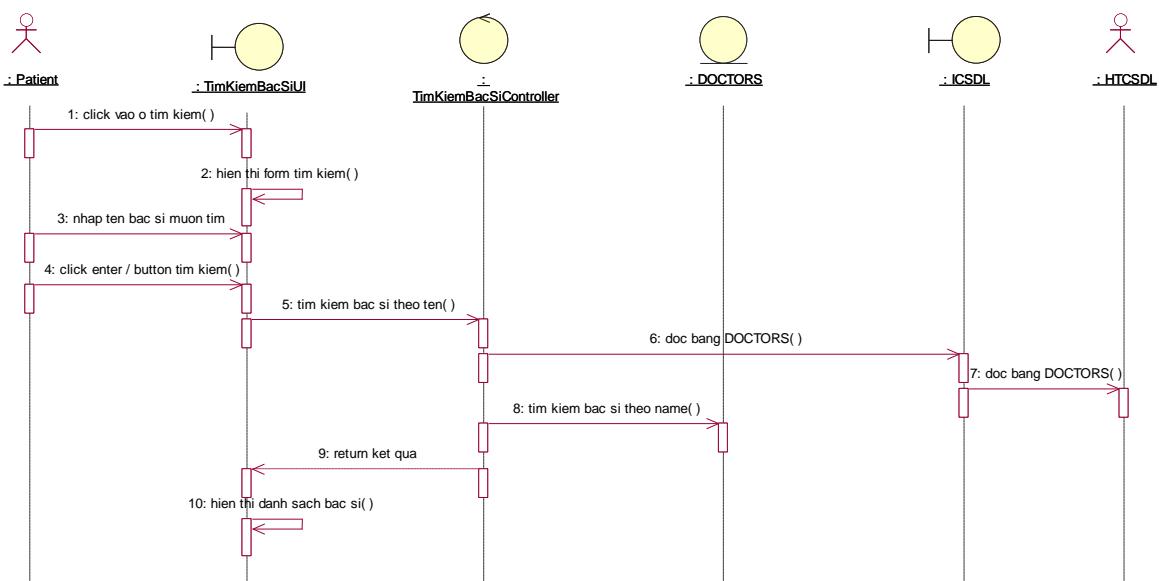
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Bệnh nhân đăng nhập thành công.

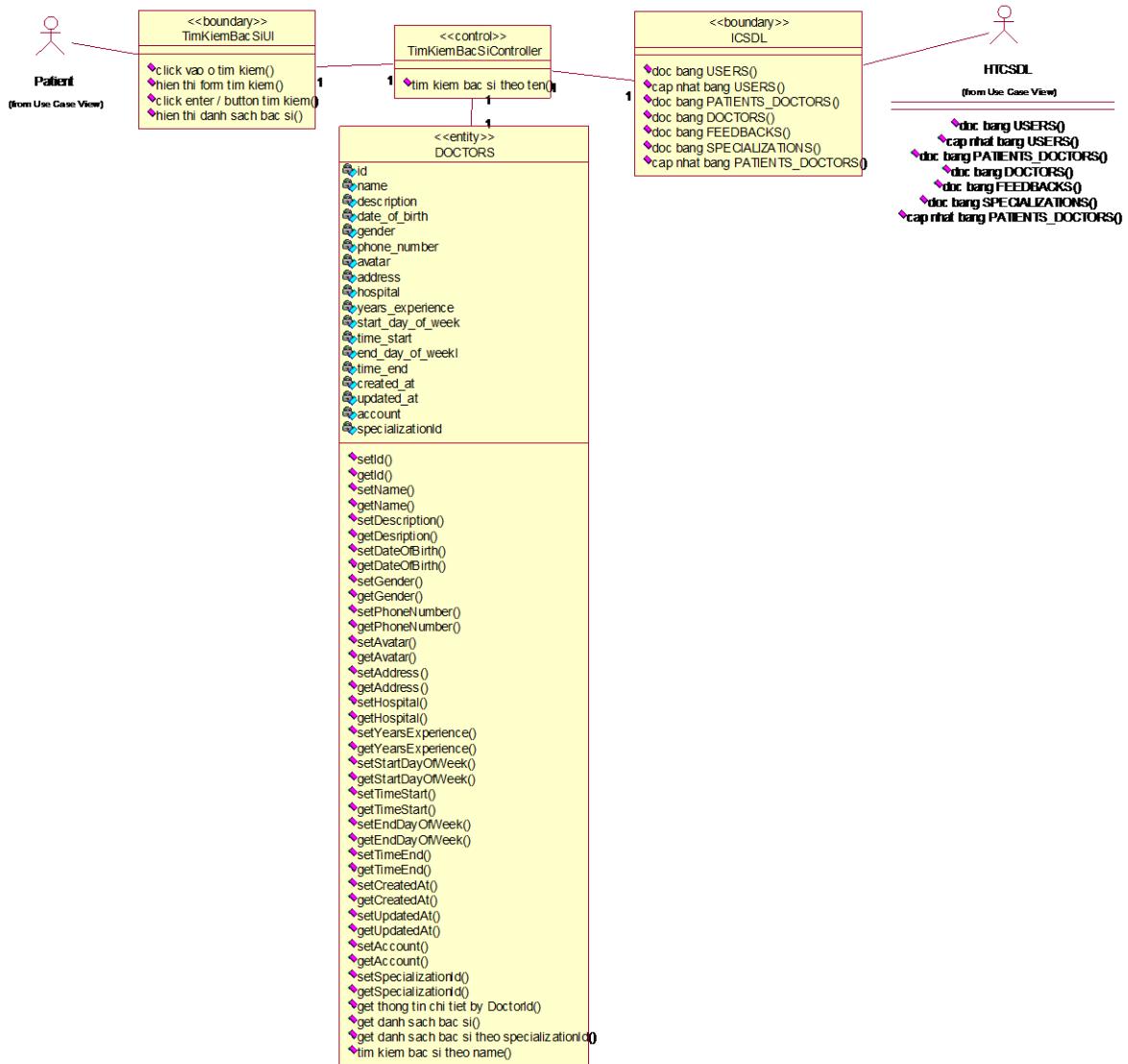
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3. 25 Biểu đồ trình tự chức năng Tìm kiếm bác sĩ

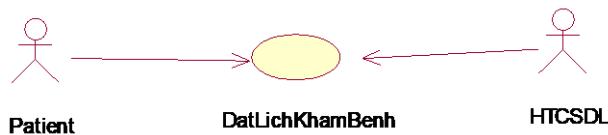
d. Biểu đồ VOPC:



Hình 3. 26 Biểu đồ VOPC chức năng Tìm kiếm bác sĩ

3.3.9. Use case **Đặt lịch khám bệnh**

a. Biểu đồ use case:



Hình 3. 27 Biểu đồ use case chức năng Đặt lịch khám bệnh

b. Đặc tả use case:

1. Tên Use Case: Đặt lịch khám bệnh

2. Mô tả vắn tắt

Use case này cho phép bệnh nhân đặt lịch khám bệnh.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case bắt đầu khi bệnh nhân click vào button “Book Appointment” tại màn hình chi tiết bác sĩ. Hệ thống sẽ lấy start day of week, time start, end day of week, time end của doctor từ bảng DOCTORS và chuyển đến màn chọn ngày và giờ cho lịch hẹn.

2. Bệnh nhân chọn ngày và giờ theo nhu cầu sau đó ấn vào button “Next”. Hệ thống sẽ lấy các gói dịch vụ từ: id, name, icon, price, description từ bảng PACKAGE_APPOINTMENTS và chuyển đến màn chọn gói dịch vụ.

3. Bệnh nhân chọn gói dịch vụ thích hợp và thời lượng khám sau đó ấn vào button “Next”. Hệ thống sẽ chuyển qua màn hình chi tiết về bệnh nhân.

4. Bệnh nhân xem lại thông tin cá nhân và nhập vấn đề đang mắc phải. Sau đó ấn vào button “Next”. Hệ thống sẽ chuyển qua màn hình Review Summary hiển thị thông tin bác sĩ, gói dịch vụ đang dùng, thời gian bắt đầu, trong bao lâu, ...

5. Bệnh nhân ấn vào button “Next”. Hệ thống sẽ lấy các thông tin cần thiết: patientId, doctorId, date, time, selected package id, ... lưu vào bảng Appointments và Appointment-contacts, đồng thời gửi mail thông báo về

cuộc hẹn cho 2 mail của doctor và patient và thông báo cho bệnh nhân biết đặt lịch thành công. Sau đó hiển thị thông báo đặt lịch thành công.

6. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

1. Tại bước 5 trong luồng cơ bản. Nếu quá trình tạo lịch hẹn xảy ra sai sót. Hệ thống sẽ thông báo “Appointment failed. Please check your internet connection then try again.”. Use case kết thúc.

2. tại các bước 2 trong luồng cơ bản. Nếu người dùng ấn nút quay lại. Use case kết thúc.

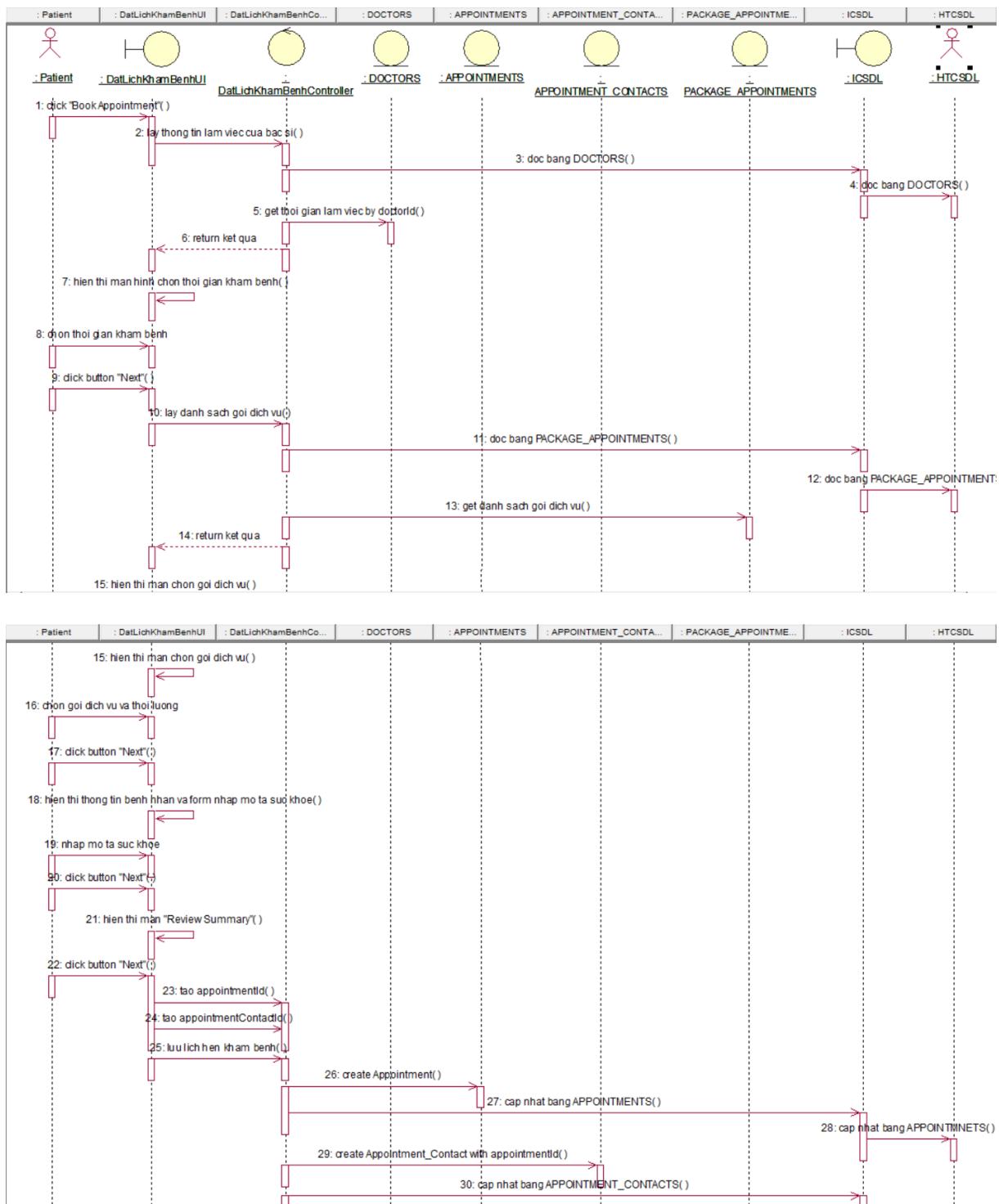
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

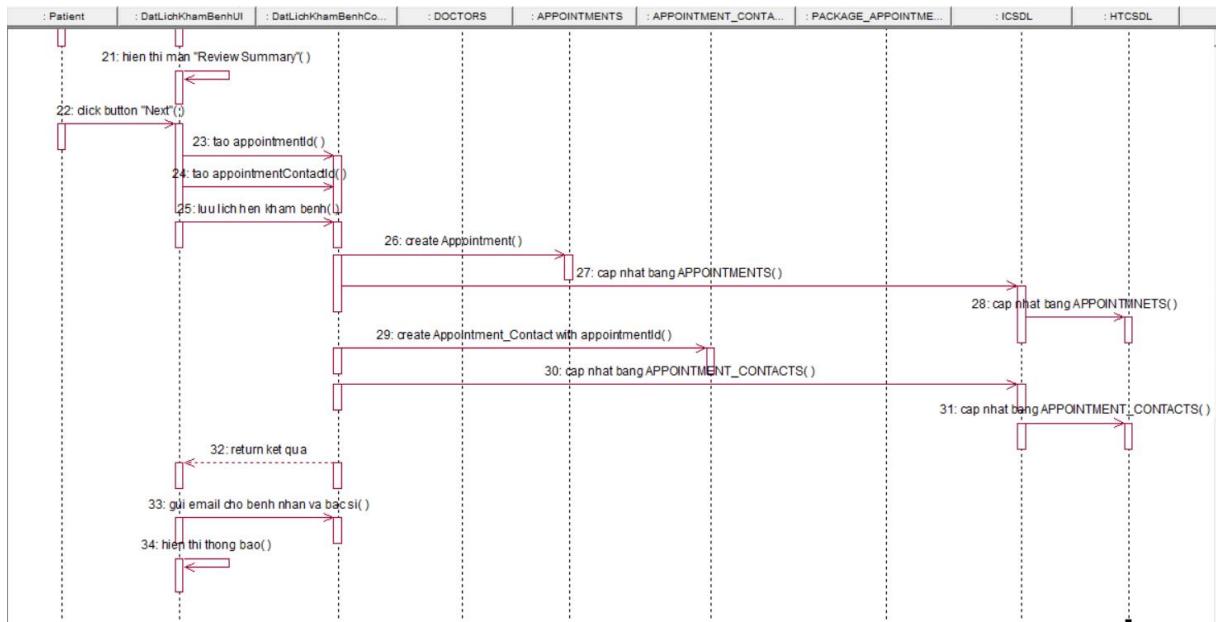
4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Bệnh nhân đã đăng nhập thành công.

6. Hậu điều kiện: Không có.

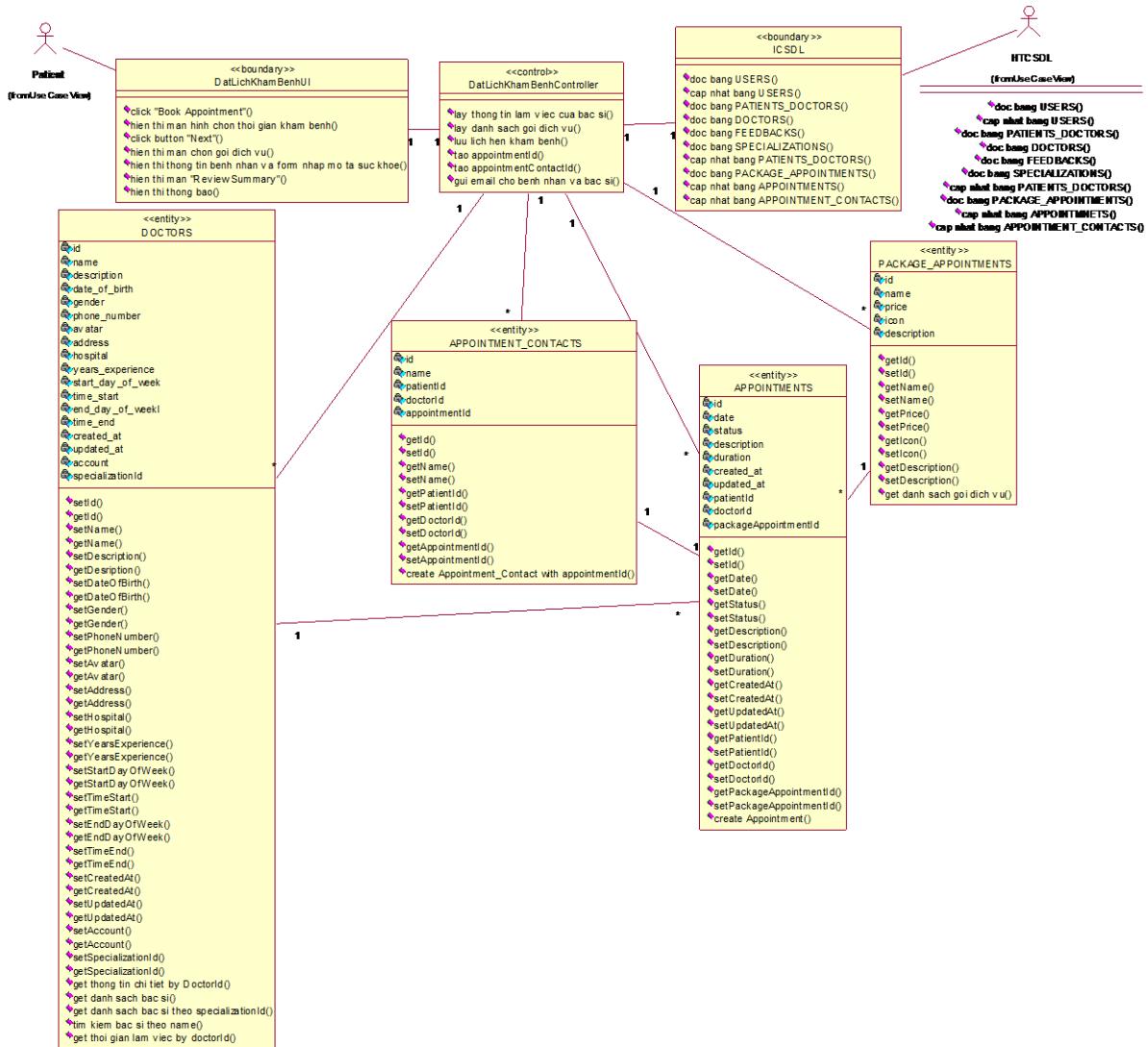
c. Biểu đồ trình tự:





Hình 3. 28 Biểu đồ trình tự chức năng Đặt lịch khám bệnh

d. Biểu đồ VOPC:



Hình 3. 29 Biểu đồ VOPC chức năng Đặt lịch khám bệnh

3.3.10. Use case Nhắn tin với bác sĩ

a. Biểu đồ use case:



Hình 3. 30 Biểu đồ usecase chức năng Nhắn tin với bác sĩ

b. Đặc tả use case:

1. Tên Use Case: Use Case Nhắn tin với bác sĩ

2. Mô tả văn tắt

Use case này cho phép bệnh nhân nhắn tin, trao đổi qua tin nhắn với bác sĩ.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case bắt đầu khi bệnh nhân nhấn vào lịch hẹn có gói dịch vụ là nhắn tin tại màn hình danh sách lịch hẹn upcoming. Hệ thống lấy thông tin chi tiết lịch hẹn của bệnh nhân đó: id, date, status, description, doctor, patient, package appointment, duration từ bảng APPOINTMENTS, DOCTORS và PACKAGE_APPOINTMENTS trong cơ sở dữ liệu và hiển thị lên màn hình.

2. Bệnh nhân nhấn button “Message” tại màn chi tiết lịch hẹn. Hệ thống sẽ lấy patientId và doctorId của lịch hẹn đó từ bảng APPOINTMENT_CONTACT sau đó tạo 1 channel trên cloud với patientId và doctorId sau đó giữ kết nối giữa patientId và doctorId. Chuyển tới màn hình nhắn tin.

3. Bệnh nhân và bác sĩ trao đổi với nhau cho đến khi bệnh nhân click nút “Complete” ở góc trên phải. Hệ thống thay đổi status của lịch hẹn từ “upcoming” thành “completed” đồng thời thông báo kết quả.

4. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

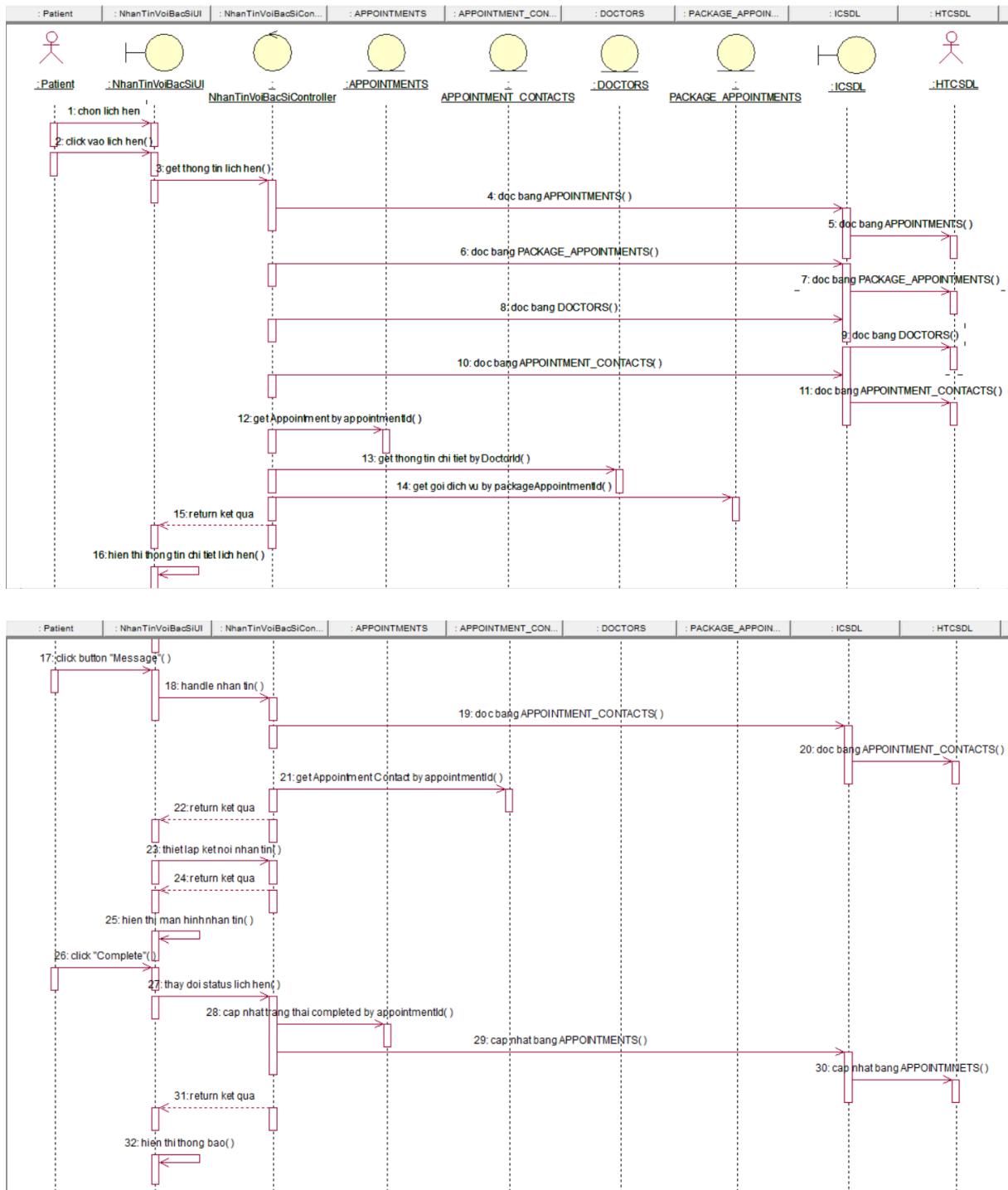
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Bệnh nhân đã đăng nhập thành công.

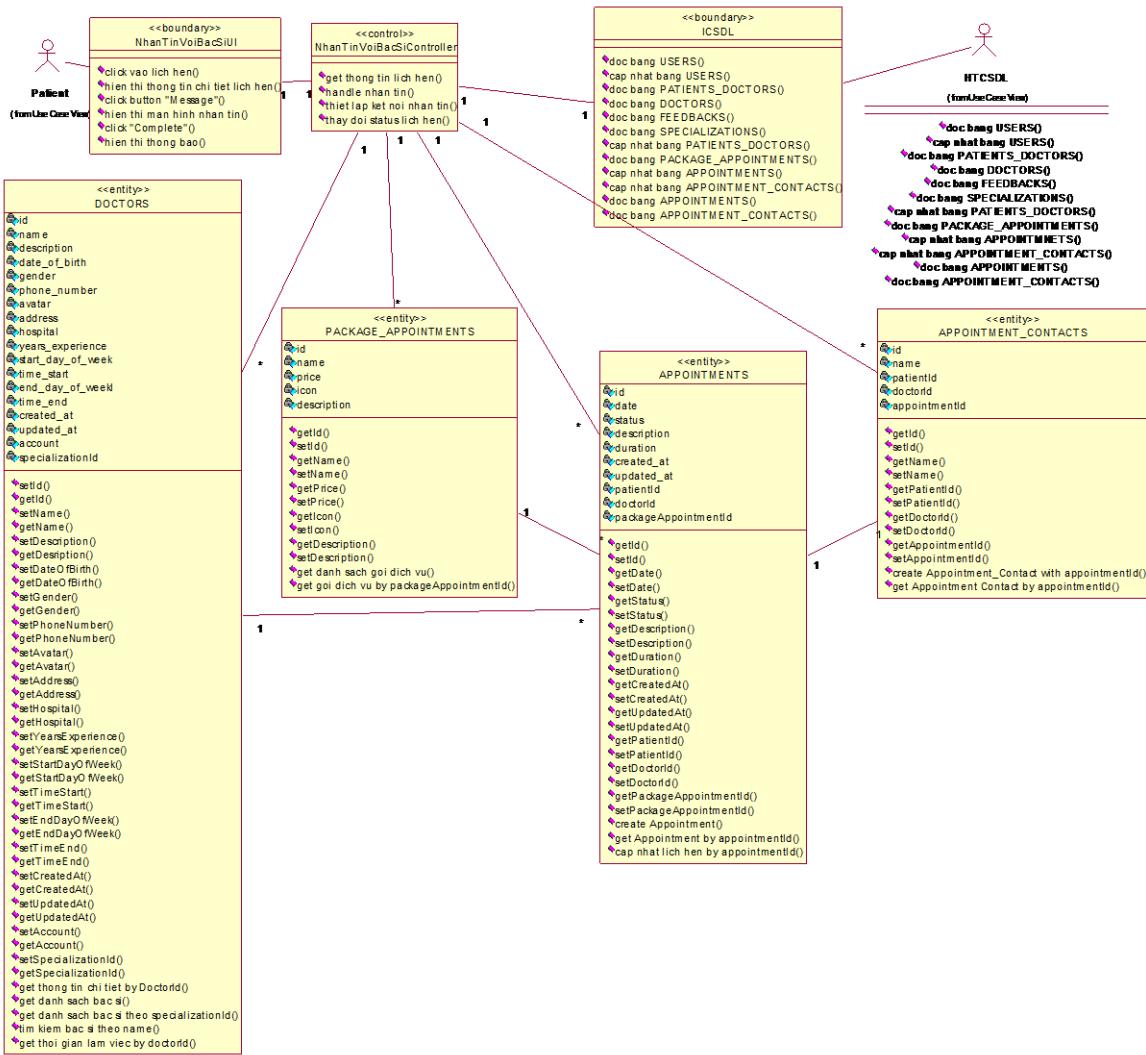
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3. 31 Biểu đồ trình tự chức năng Nhắn tin với bác sĩ

d. Biểu đồ VOPC:



Hình 3. 32 Biểu đồ VOPC chúc năng Nhắn tin với bác sĩ

3.3.11. Use case Đánh giá dịch vụ

a. Biểu đồ use case:



Hình 3. 33 Biểu đồ usecase chúc năng Đánh giá dịch vụ

b. Đặc tả use case:

1. Tên Use Case: Use Case Đánh giá dịch vụ

2. Mô tả văn tắt

Use case này cho phép bệnh nhân nhận xét, đánh giá dịch vụ và bác sĩ.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case bắt đầu khi bệnh nhân chọn “Leave a Review” bác sĩ muốn nhận xét tại màn hình danh sách lịch hẹn ở tab “Completed”. Hệ thống sẽ chuyển tới màn hình Review.

2. Bệnh nhân chọn số sao mình muốn đánh giá và nhập review về bác sĩ. Sau đó án “Submit” để gửi đánh giá. Hệ thống sẽ patientId, doctorId và appointmentId của lịch hẹn được review. Sau đó cùng với rating và comment mà bệnh nhân đã nhập trước đó thêm vào bảng Feedbacks. Sau đó thông báo submit thành công.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh:

1. Tại bước 2 của luồng cơ bản. Nếu lưu thất bại hệ thống sẽ thông báo lỗi và yêu cầu thử lại. Use case kết thúc.

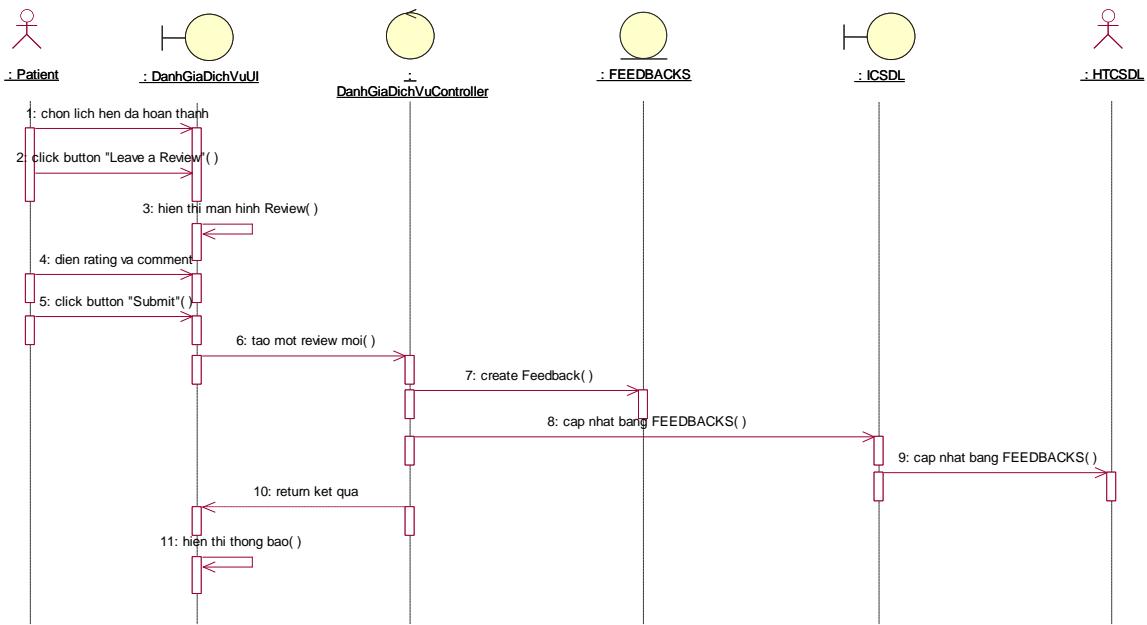
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Không có.

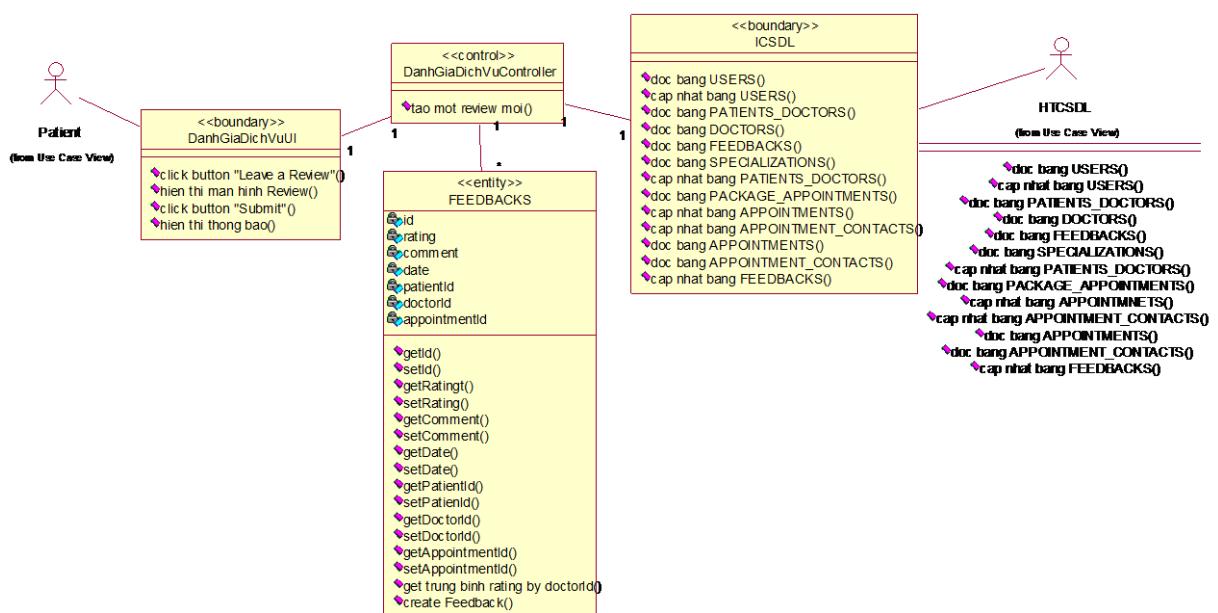
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3. 34 Biểu đồ trình tự chức năng Dánh giá dịch vụ

d. Biểu đồ VOPC:



Hình 3. 35 Biểu đồ VOPC chức năng Dánh giá dịch vụ

3.3.12. Use case Call với bác sĩ

a. Biểu đồ use case:



Hình 3. 36 Biểu đồ use case chức năng Call với bác sĩ

b. Đặc tả use case:

1. Tên Use Case: Use case Call với bác sĩ
2. Mô tả vắn tắt: Use case này cho bệnh nhân call trực tiếp với bác sĩ.
3. Luồng các sự kiện:
 - 3.1. Luồng cơ bản:
 1. Use case bắt đầu khi bệnh nhân chọn lịch hẹn có gói dịch vụ là call tại màn hình danh sách lịch hẹn. Hệ thống lấy thông tin chi tiết lịch hẹn của bệnh nhân đó: id, date, status, description, doctor, patient, package appointment, duration từ bảng APPOINTMENTS, DOCTORS và PACKAGE_APPOINTMENTS trong cơ sở dữ liệu và hiển thị lên màn hình.
 2. Bệnh nhân nhấn nút “Voice Call” hoặc “Video Call” để tiến hành cuộc gọi. Hệ thống sẽ lấy patientId và doctorId của lịch hẹn đó từ bảng APPOINTMENT_CONTACT để gửi thông báo cuộc gọi tới thiết bị của bác sĩ.
 3. Bác sĩ ấn vào nghe cuộc gọi để tiến hành call với bệnh nhân. Hệ thống sẽ kết nối bệnh nhân và bác sĩ với nhau sau đó chuyển qua màn hình call.
 4. Sau khi quá trình call hoàn tất, bệnh nhân tiến hành ấn nút end call ở thanh điều khiển cuộc gọi. Hệ thống ngắt kết nối người dùng đó khỏi cuộc gọi. Thay đổi status của lịch hẹn thành “completed” và hiển thị thông báo.
 5. Use case kết thúc.

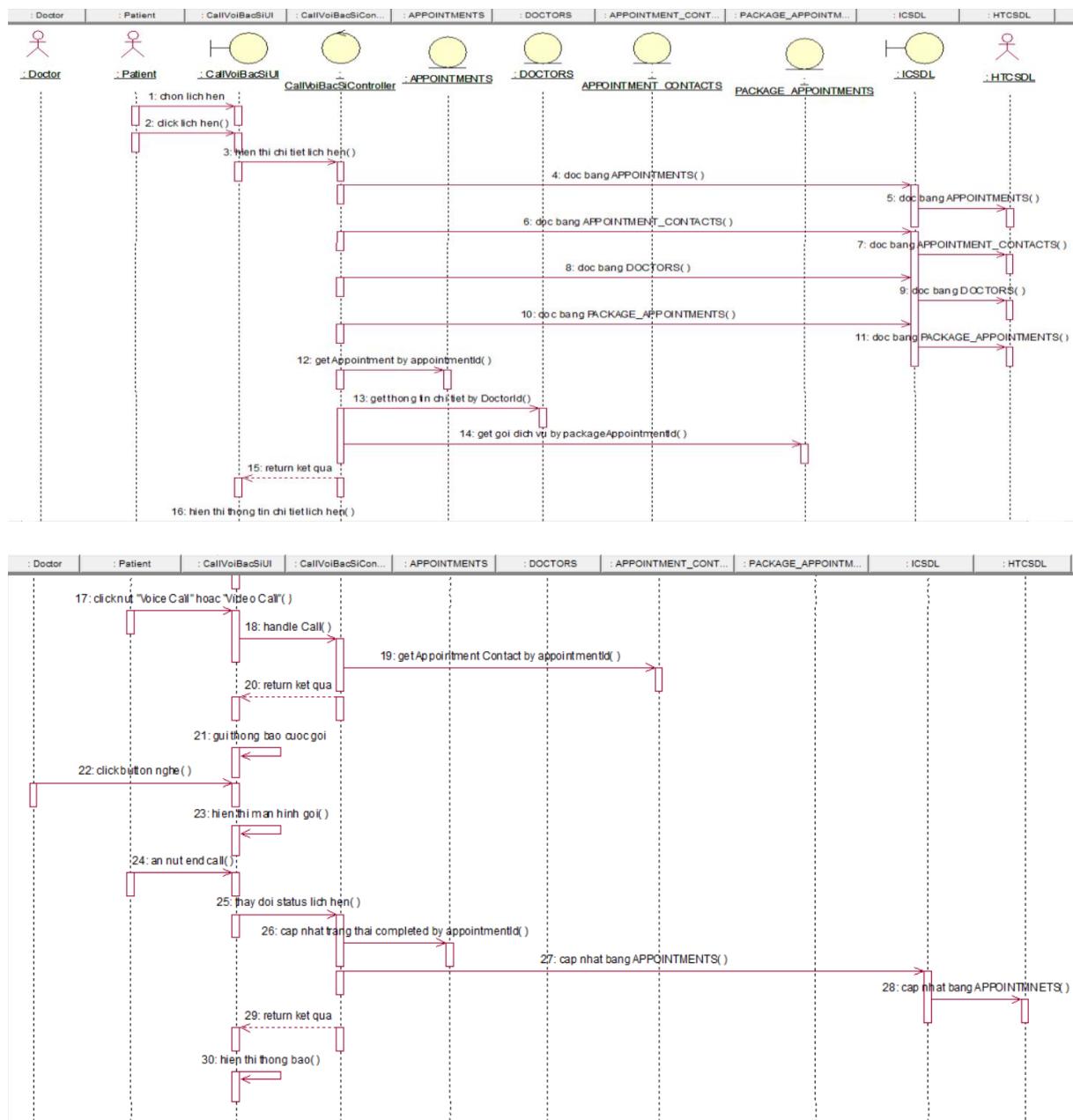
3.2. Các luồng rẽ nhánh:

1. Tại bước 3 của luồng cơ bản, nếu bác sĩ ấn không nghe cuộc gọi hoặc thời gian chờ nghe quá 1 phút thì cuộc gọi kết thúc. Use case kết thúc.

2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

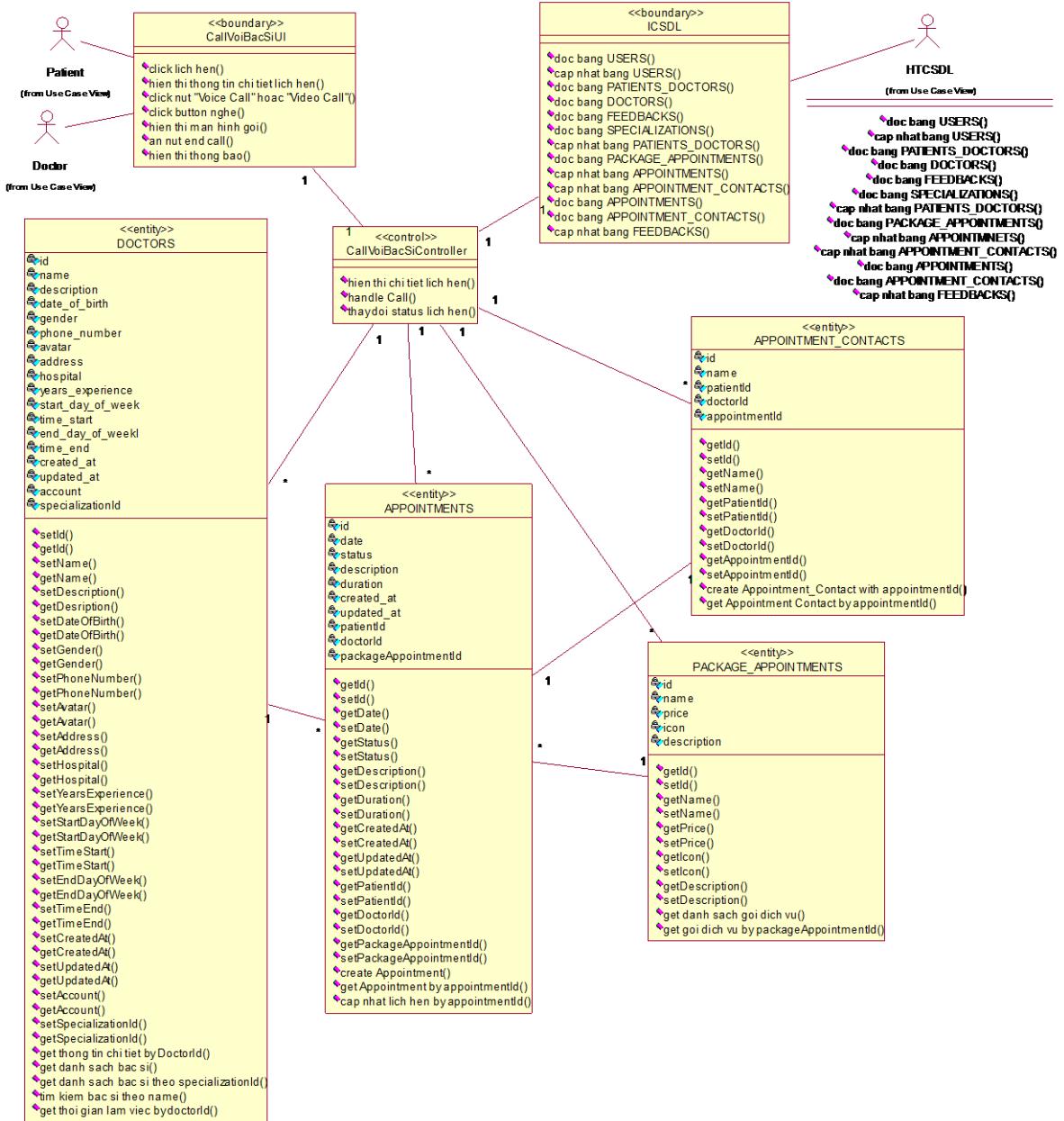
4. Các yêu cầu đặc biệt: Không có
5. Tiền điều kiện: Bệnh nhân đã đăng nhập vào hệ thống.
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3. 37 Biểu đồ trình tự chức năng Call với bác sĩ

d. Biểu đồ VOPC:



Hình 3. 38 Biểu đồ VOPC chúc năng Call với bác sĩ

3.3.13. Use case Quản lý danh sách lịch hẹn (patient)

a. Biểu đồ use case:



Hình 3. 39 Biểu đồ use case chức năng Quản lý danh sách lịch hẹn (patient)

b. Đặc tả use case:

1. Tên Use Case: Use Case Quản lý danh sách lịch hẹn (patient)

2. Mô tả văn tắt:

Use case này cho phép bệnh nhân xem danh sách lịch hẹn (upcoming, completed, cancelled) và hủy lịch hẹn đang upcoming.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1) Use case này bắt đầu khi bệnh nhân click vào tab “Appointments”. Hệ thống lấy thông tin chi tiết của các lịch hẹn của bệnh nhân đó: id, date, status, description, doctor, package appointment, duration từ bảng APPOINTMENTS, DOCTORS và PACKAGE_APPOINTMENTS trong cơ sở dữ liệu và hiển thị lên màn hình.

2) Xem danh sách lịch hẹn theo status:

a) Bệnh nhân click vào các cột trạng thái ở bên trên màn hình. Hệ thống sẽ hiển thị các danh sách lịch hẹn của bệnh nhân theo status tương ứng.

3) Dừng lịch hẹn:

a) Tại tab lịch hẹn upcoming, bệnh nhân click vào nút “Cancel Appointment” ở dưới mỗi lịch hẹn. Hệ thống sẽ thay đổi status lịch hẹn thành “cancelled” đồng thời thông báo hủy thành công.

4) Đặt lịch lại:

- a) Tại tab lịch hẹn completed, bệnh nhân click vào nút “Book again” ở dưới mỗi lịch hẹn. Hệ thống sẽ chuyển tới màn hình thông tin chi tiết bác sĩ.

5) Use case kết thúc

3.2. Các luồng rẽ nhánh:

1. Tại bước 1 của luồng cơ bản. Nếu danh sách lịch hẹn của bệnh nhân là trống thì sẽ hiển thị màn hình “You don't have an appointment yet” thông báo bệnh nhân chưa có lịch hẹn n

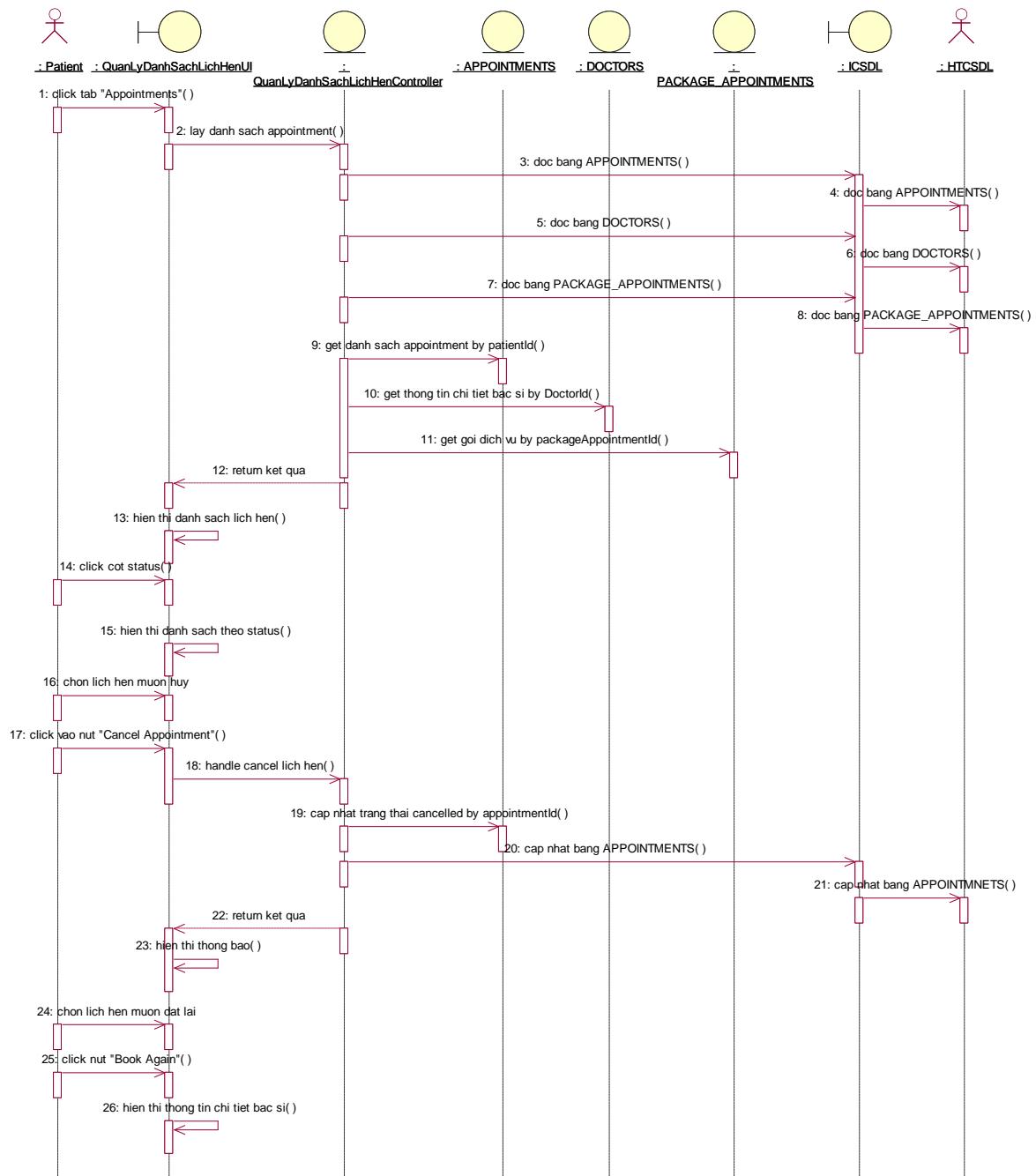
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: bệnh nhân đã đăng nhập vào hệ thống.

6. Hậu điều kiện: Không có.

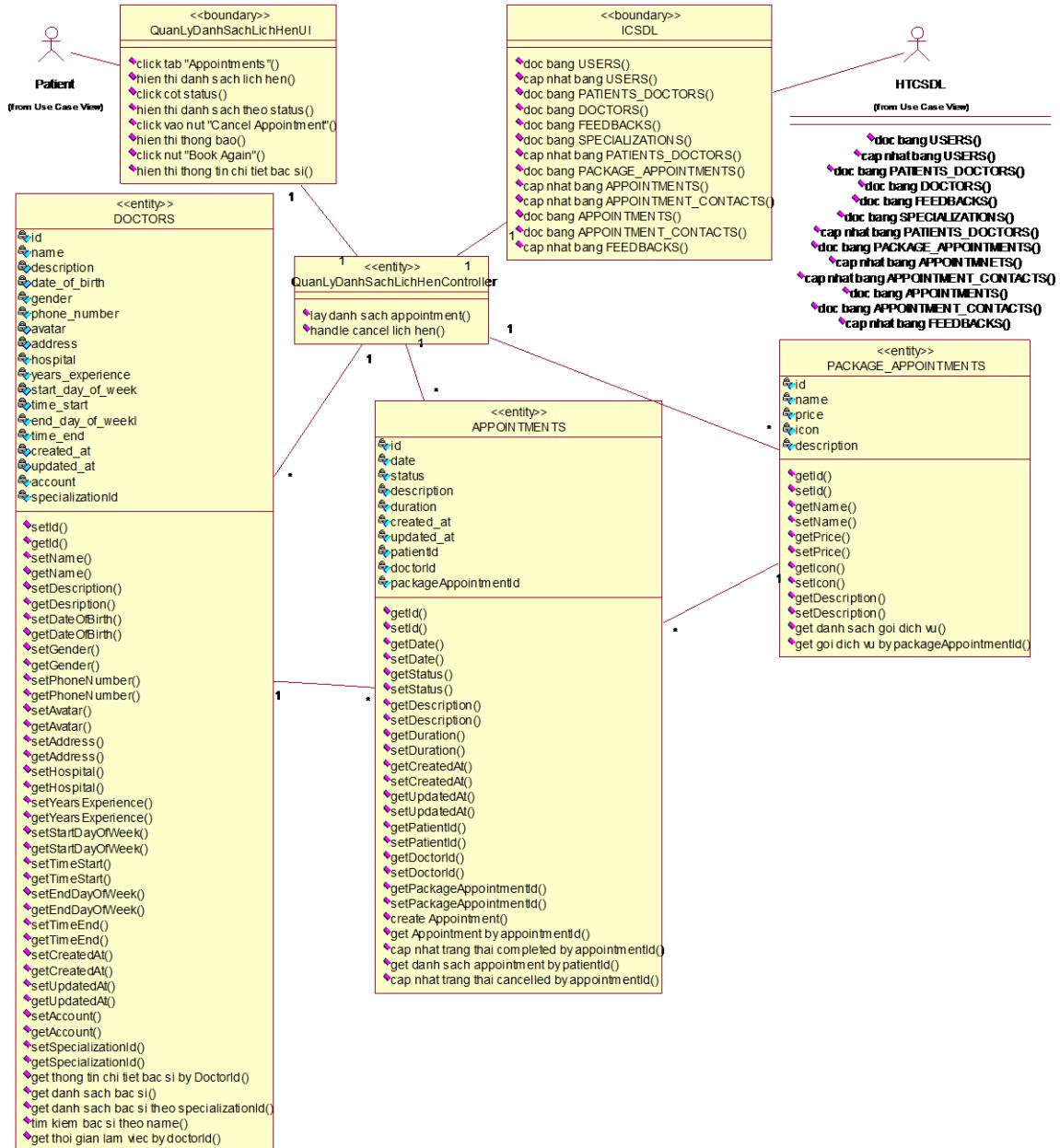
c. Biểu đồ trình tự:



Hình 3. 40 Biểu đồ trình tự chức năng (patient)

Quản lý danh sách lịch hẹn

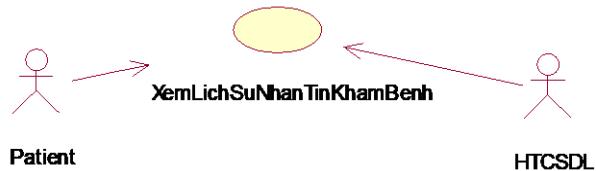
d. Biểu đồ VOPC:



Hình 3. 41 Biểu đồ VOPC chirc năng Quản lý danh sách lịch hẹn (patient)

3.3.14. Use case Xem lịch sử khám bệnh (patient)

a) Biểu đồ use case:



Hình 3. 42 Biểu đồ usecase chức năng Xem lịch sử nhắn tin khám bệnh (patient)

b) Đặc tả use case:

1. Tên Use Case: Use Case Xem lịch sử nhắn tin khám bệnh (patient)

2. Mô tả văn tắt

Use case này cho phép bệnh nhân xem lại lịch sử trò chuyện với bác sĩ.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1. Use case bắt đầu khi bệnh nhân ấn vào tab “History”. Hệ thống sẽ lấy danh sách các tin nhắn trên cloud của bệnh nhân (patientId) thông qua bảng Appointment-contacts và hiển thị lên màn hình danh sách bác sĩ đã trò chuyện.

2. Bệnh nhân chọn vào 1 bác sĩ ở màn hình. Hệ thống sẽ hiển thị lịch sử trò chuyện giữa bệnh nhân và bác sĩ.

3. Use case kết thúc

3.2. Các luồng rẽ nhánh:

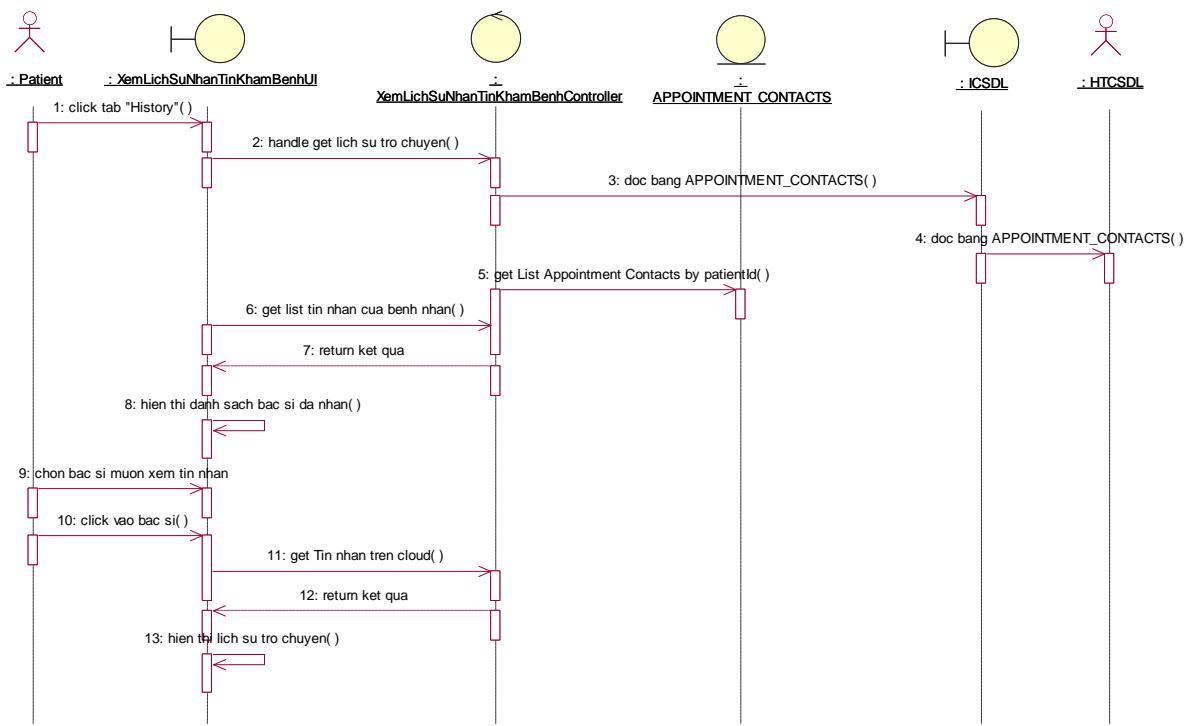
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Bệnh nhân đã đăng nhập vào hệ thống.

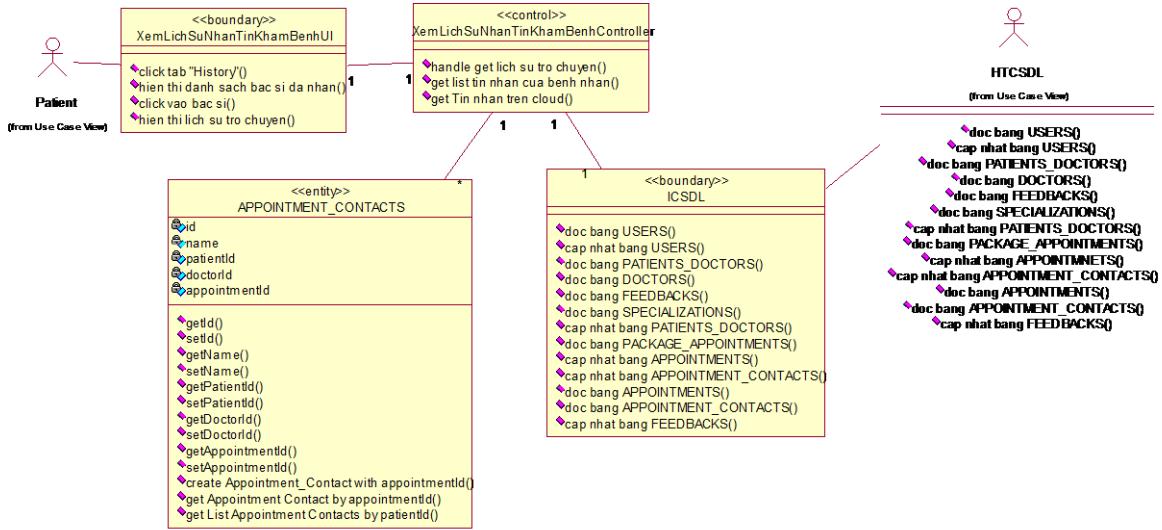
6. Hậu điều kiện: Không có.

c) Biểu đồ trình tự:



Hình 3. 43 Biểu đồ trình tự chức năng Xem lịch sử nhắn tin khám bệnh (patient)

d) Biểu đồ VOPC:



Hình 3. 44 Biểu đồ VOPC chức năng Xem lịch sử nhắn tin khám bệnh (patient)

3.3.15. Use case Quản lý thông tin cá nhân (patient)

a. Biểu đồ use case:



Hình 3. 45 Biểu đồ use case chức năng Quản lý thông tin cá nhân (patient)

b. Đặc tả use case:

1. Tên Use Case: Use Case Quản lý thông tin cá nhân (patient)

2. Mô tả văn tắt

Use case này cho phép bệnh nhân quản lý thông tin cá nhân.

3. Luồng các sự kiện:

3.1. Luồng cơ bản:

1) Use case này bắt đầu khi bệnh nhân nhấp vào tab “Profile” trong màn hình.

Hệ thống lấy thông tin chi tiết của bệnh nhân: id, name, avatar, email, address,

birth of date, gender từ bảng USERS và PATIENTS trong cơ sở dữ liệu và hiển thị lên màn hình.

2) Edit profile:

- a) Bệnh nhân ấn vào nút “Edit Profile” tại màn hình Profile. Hệ thống sẽ chuyển tới màn hình form Edit Profile với các thông tin cá nhân cũ của bệnh nhân hoặc mặc định.
- b) Bệnh nhân thay đổi thông tin cá nhân, sau đó ấn “Update”. Hệ thống sẽ tiến hành cập nhật thông tin cá nhân mới vào bảng PATIENTS và hiển thị lại thông tin cá nhân mới của bệnh nhân.

3) Upload Avatart:

- a) Bệnh nhân nhấn vào icon Cây bút ở góc phải avatar. Hệ thống hiển thị danh sách file ảnh có trong thiết bị.
- b) Người dùng chọn ảnh muốn upload và căn chỉnh ảnh. Sau đó ấn nút upload. Hệ thống sẽ lưu ảnh lên storage S3 và cập nhật link ảnh mới vào cột avatar của bảng PATIENTS của bệnh nhân.

4) Use case kết thúc.

3.2. Các luồng rẽ nhánh:

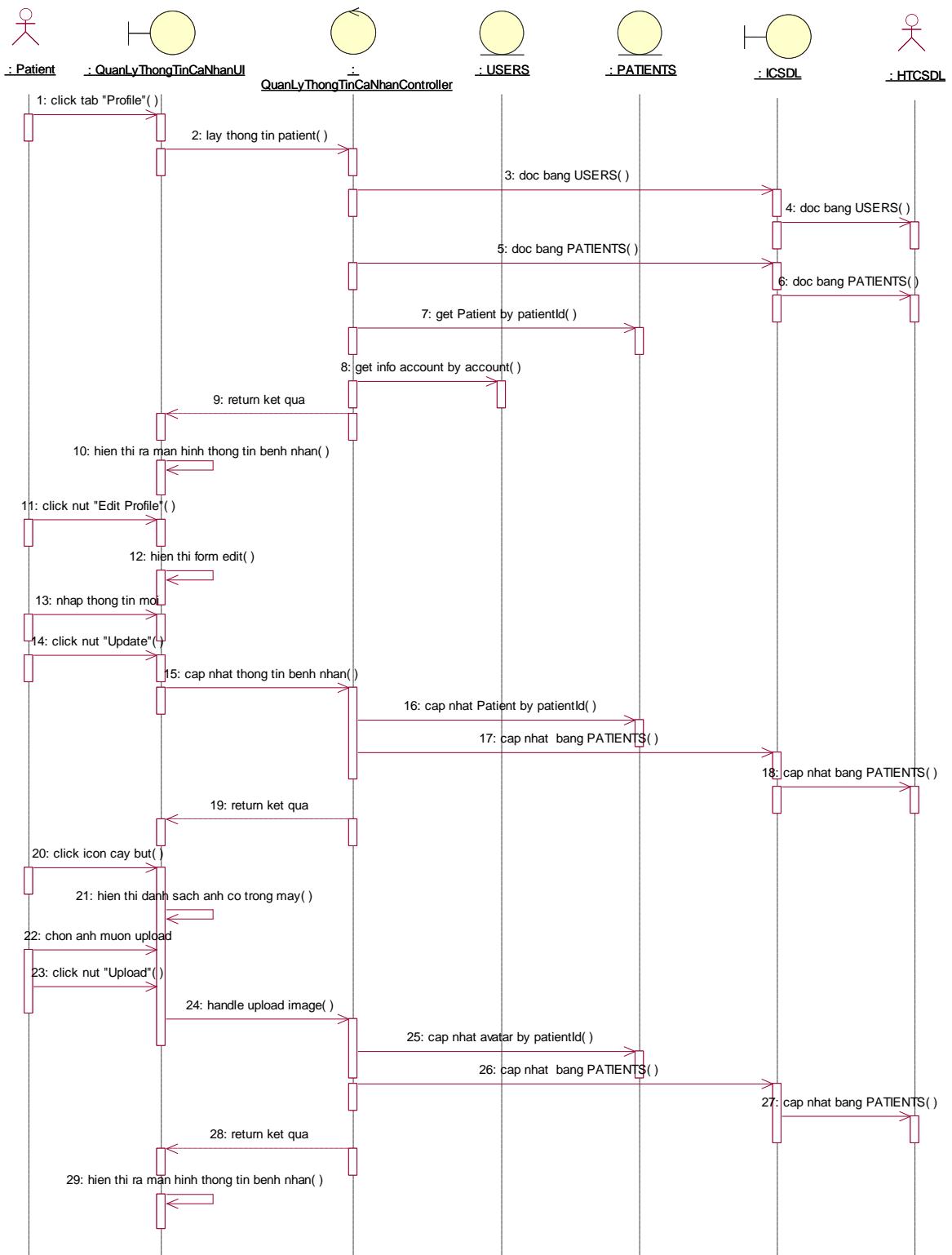
- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
- 2) Tại bước 2a, 3a bệnh nhân click vào nút “Quay lại”. Use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Bệnh nhân đã đăng nhập vào hệ thống.

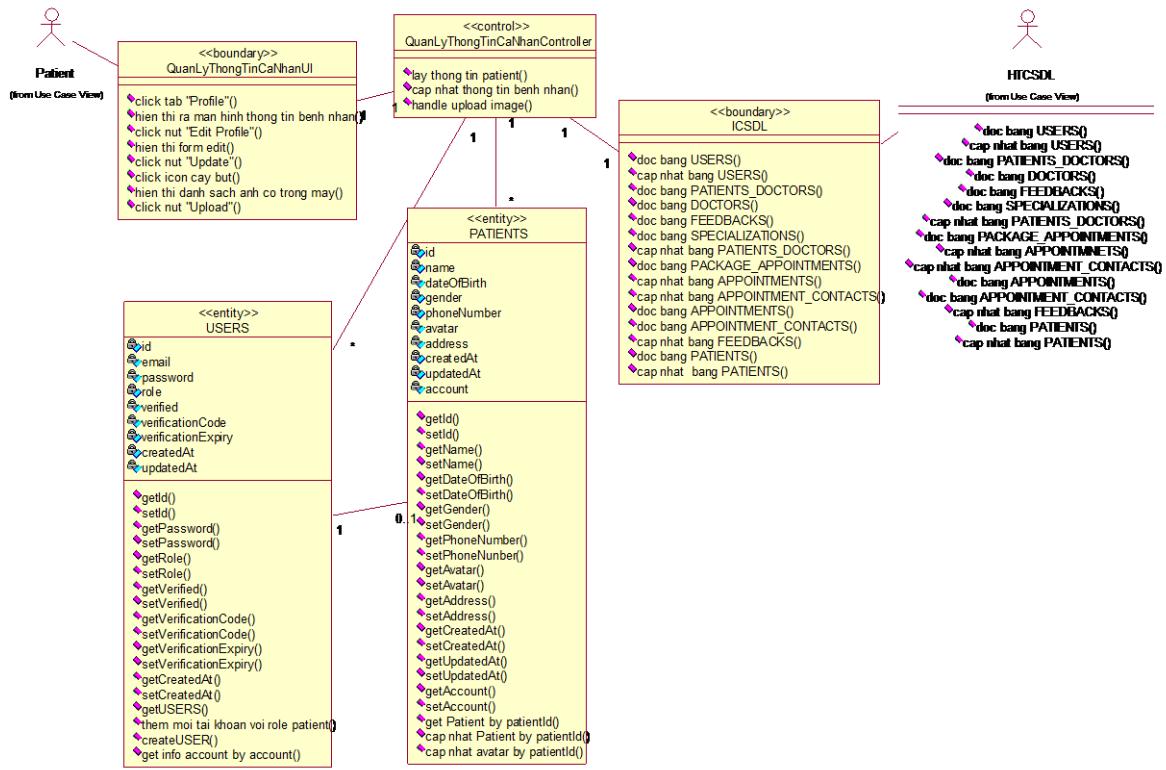
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự:



Hình 3. 46 Biểu đồ trình tự chức năng Quản lý thông tin cá nhân (patient)

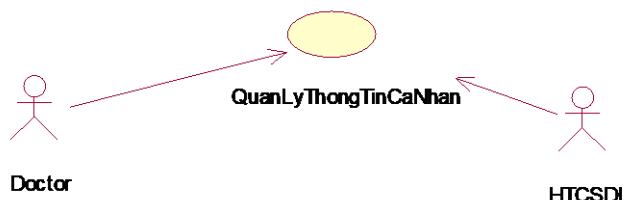
d. Biểu đồ VOPC:



Hình 3. 47 Biểu đồ VOPC chức năng Quản lý thông tin cá nhân (patient)

3.3.16. Use case Quản lý thông tin cá nhân (doctor)

a. Biểu đồ use case



Hình 3. 48 Biểu đồ use case chức năng Quản lý thông tin cá nhân (doctor)

b. Đặc tả use case:

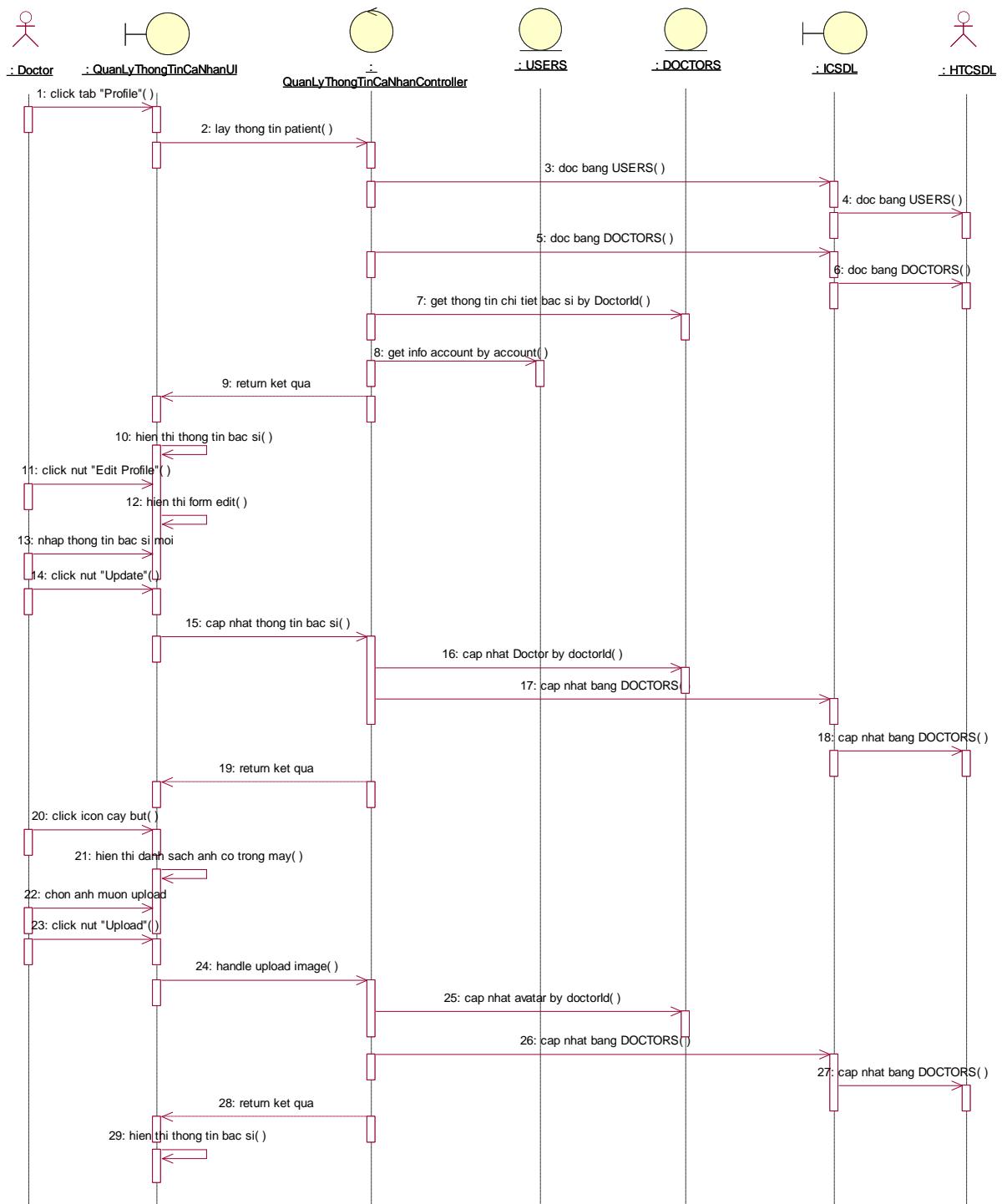
- Tên use case: Use case Quản lý thông tin cá nhân (doctor)
- Mô tả văn tắt

Use case này cho phép bác sĩ quản lý thông tin cá nhân.

3. Luồng sự kiện

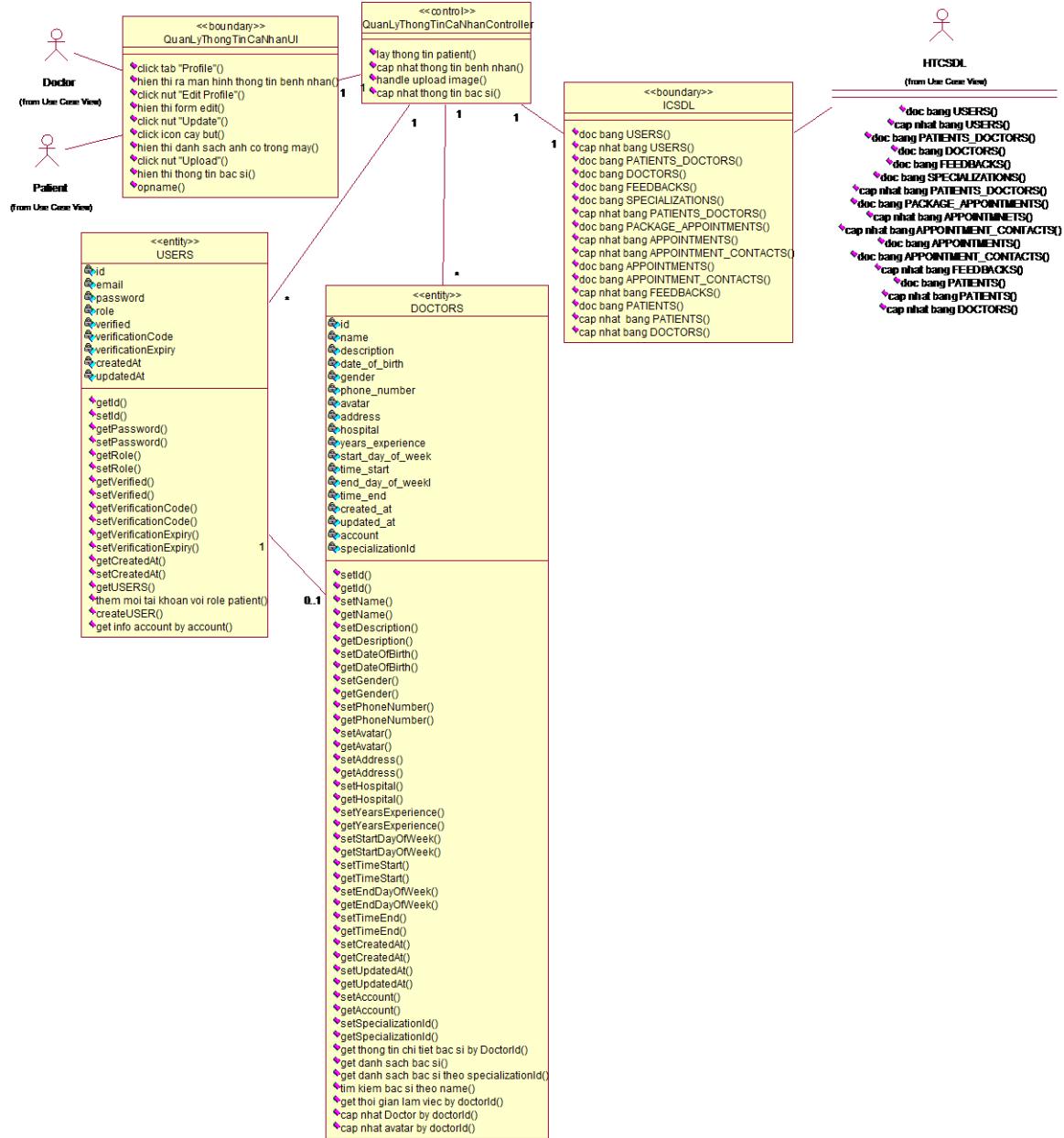
3.1. Luồng cơ bản

1. Use case này bắt đầu khi bác sĩ nhấn vào tab “Profile” trong màn hình. Hệ thống lấy thông tin chi tiết của bác sĩ: id, name, avatar, email, specialization, hospital, description, address, birth of date, gender từ bảng USERS, DOCTORS và SPECIALIZATIONS trong cơ sở dữ liệu và hiển thị lên màn hình.
 2. Edit profile:
 - a) Bác sĩ ấn vào nút “Edit Profile” tại màn hình Profile. Hệ thống sẽ chuyển tới màn hình form Edit Profile với các thông tin cá nhân cũ của bác sĩ hoặc mặc định.
 - b) Bác sĩ thay đổi thông tin cá nhân, sau đó ấn “Update”. Hệ thống sẽ tiến hành cập nhật thông tin cá nhân mới vào bảng DOCTORS và hiển thị lại thông tin cá nhân mới của bác sĩ.
 3. Upload Avatar:
 - a) Bác sĩ nhấn vào icon Cây bút ở góc phải avatar. Hệ thống hiển thị danh sách file ảnh có trong thiết bị.
 - b) Người dùng chọn ảnh muốn upload và căn chỉnh ảnh. Sau đó ấn nút upload. Hệ thống sẽ lưu ảnh lên storage S3 và cập nhật link ảnh mới vào cột avatar của bảng DOCTORS của bác sĩ.
 4. Use case kết thúc.
- ### 3.2. Các luồng rẽ nhánh:
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
 2. Tại bước 2a, 3a bác sĩ click vào nút “Quay lại”. Use case kết thúc.
 4. Các yêu cầu đặc biệt: Không có
 5. Tiền điều kiện: Bác sĩ đã đăng nhập vào hệ thống.
 6. Hậu điều kiện: Không có.
 - c. Biểu đồ trình tự



Hình 3. 49 Biểu đồ trình tự chức năng Quản lý thông tin cá nhân (doctor)

d. Biểu đồ VOPC



Hình 3. 50 Biểu đồ VOPC chức năng Quản lý thông tin cá nhân (doctor)

3.3.17. Use case Quản lý danh sách lịch hẹn (doctor)

a. Biểu đồ use case



Hình 3. 51 Biểu đồ use case chức năng Quản lý danh sách lịch hẹn

b. Đặc tả use case:

1. Tên use case: Use case Quản lý danh sách lịch hẹn (doctor)
2. Mô tả vắn tắt

Use case này cho phép bác sĩ xem danh sách lịch hẹn (upcoming, completed, cancelled) và hủy lịch hẹn đang upcoming.

3. Luồng sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi bác sĩ click vào tab “Appointments”. Hệ thống lấy thông tin chi tiết của các lịch hẹn của bác sĩ đó: id, date, status, description, patient, package appointment, duration từ bảng APPOINTMENTS, PATIENTS và PACKAGE_APPOINTMENTS trong cơ sở dữ liệu và hiển thị lên màn hình.

2. Xem danh sách lịch hẹn theo status:

a) Bác sĩ click vào các cột trạng thái ở bên trên màn hình. Hệ thống sẽ hiển thị các danh sách lịch hẹn của bác sĩ theo status tương ứng.

3. Dừng lịch hẹn:

a) Tại tab lịch hẹn upcoming, bác sĩ click vào nút “Cancel Appointment” ở dưới mỗi lịch hẹn. Hệ thống sẽ thay đổi status lịch hẹn thành “cancelled” đồng thời tải mới lại danh sách lịch hẹn.

4. Use case kết thúc

3.2. Các luồng rẽ nhánh

1. Tại bước 1 của luồng cơ bản. Nếu danh sách lịch hẹn của bệnh nhân là trống thì sẽ hiển thị màn hình “You don't have an appointment yet” thông báo bệnh nhân chưa có lịch hẹn.

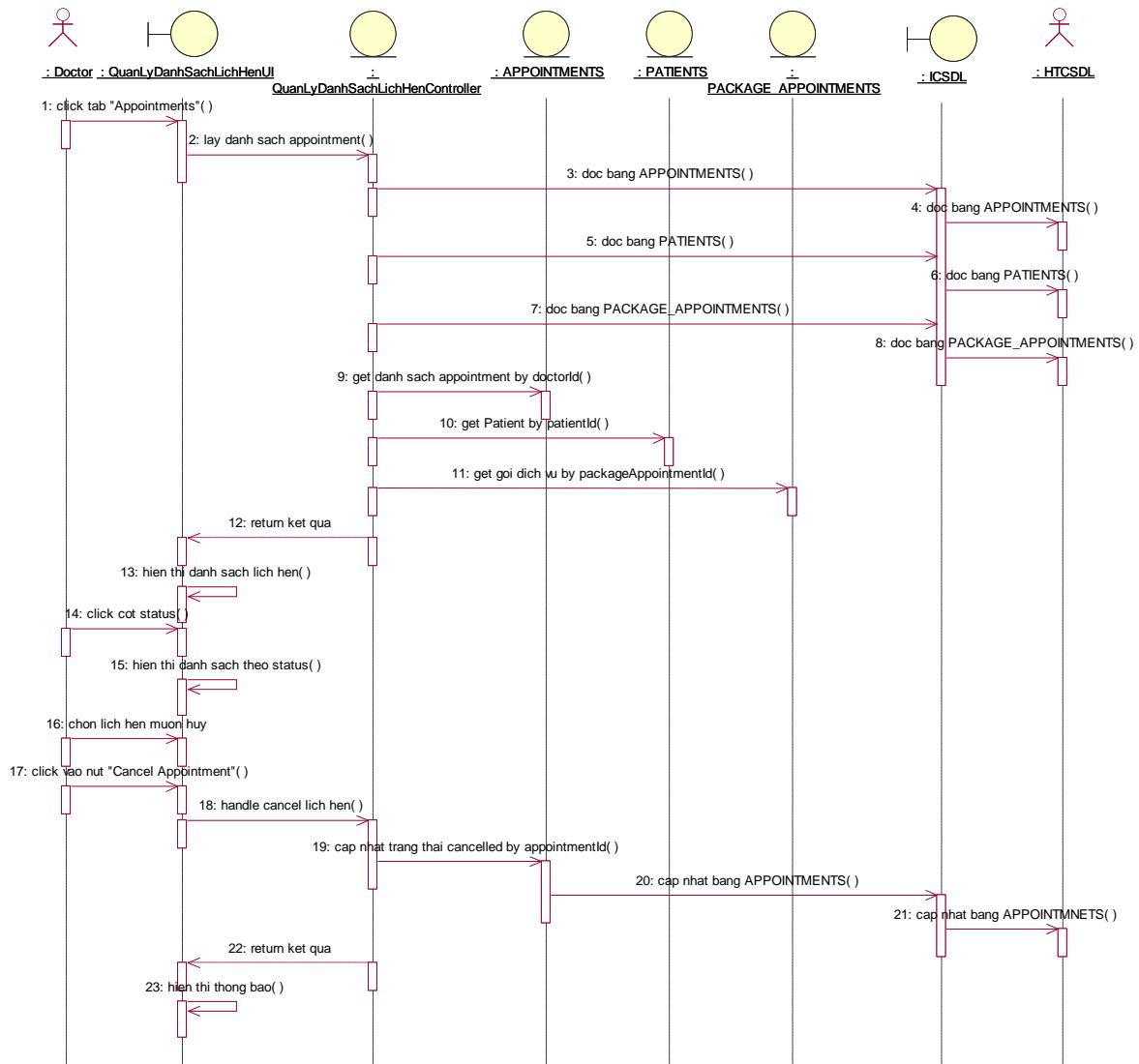
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Bác sĩ đã đăng nhập vào hệ thống.

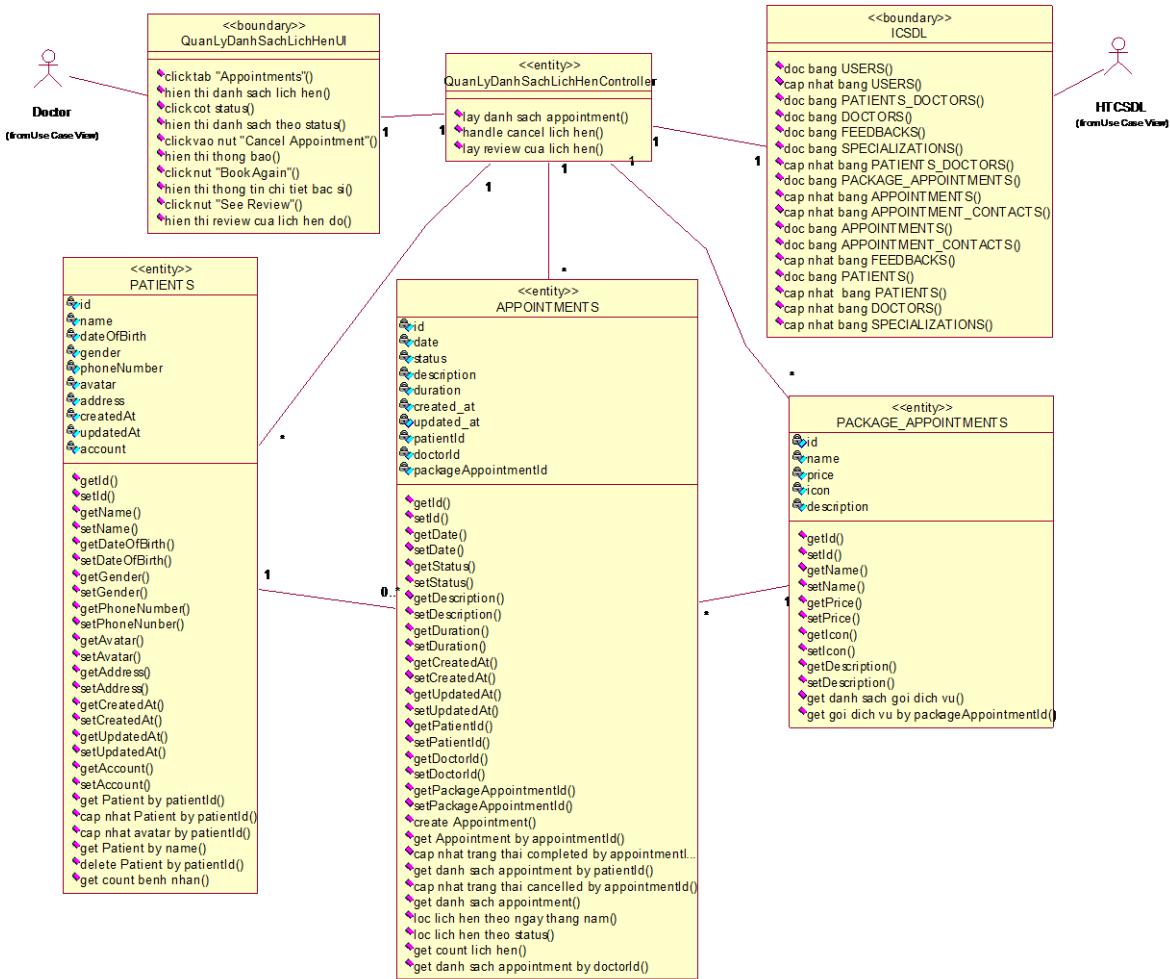
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự



Hình 3. 52 Biểu đồ trình tự chirc năng Quản lý danh sách lịch hẹn (doctor)

d. Biểu đồ VOPC



Hình 3. 53 Biểu đồ VOPC chúc năng Quản lý danh sách lịch hẹn (doctor)

3.3.18. Use case Nhắn tin với bệnh nhân

a. Biểu đồ use case



Hình 3. 54 Biểu đồ use case chúc năng Nhắn tin với bệnh nhân

b. Đặc tả use case:

- Tên use case: Use case nhắn tin với bệnh nhân
- Mô tả văn tắt

Use case này cho phép bác sĩ nhắn tin tư vấn, trao đổi trực tiếp với bệnh nhân.

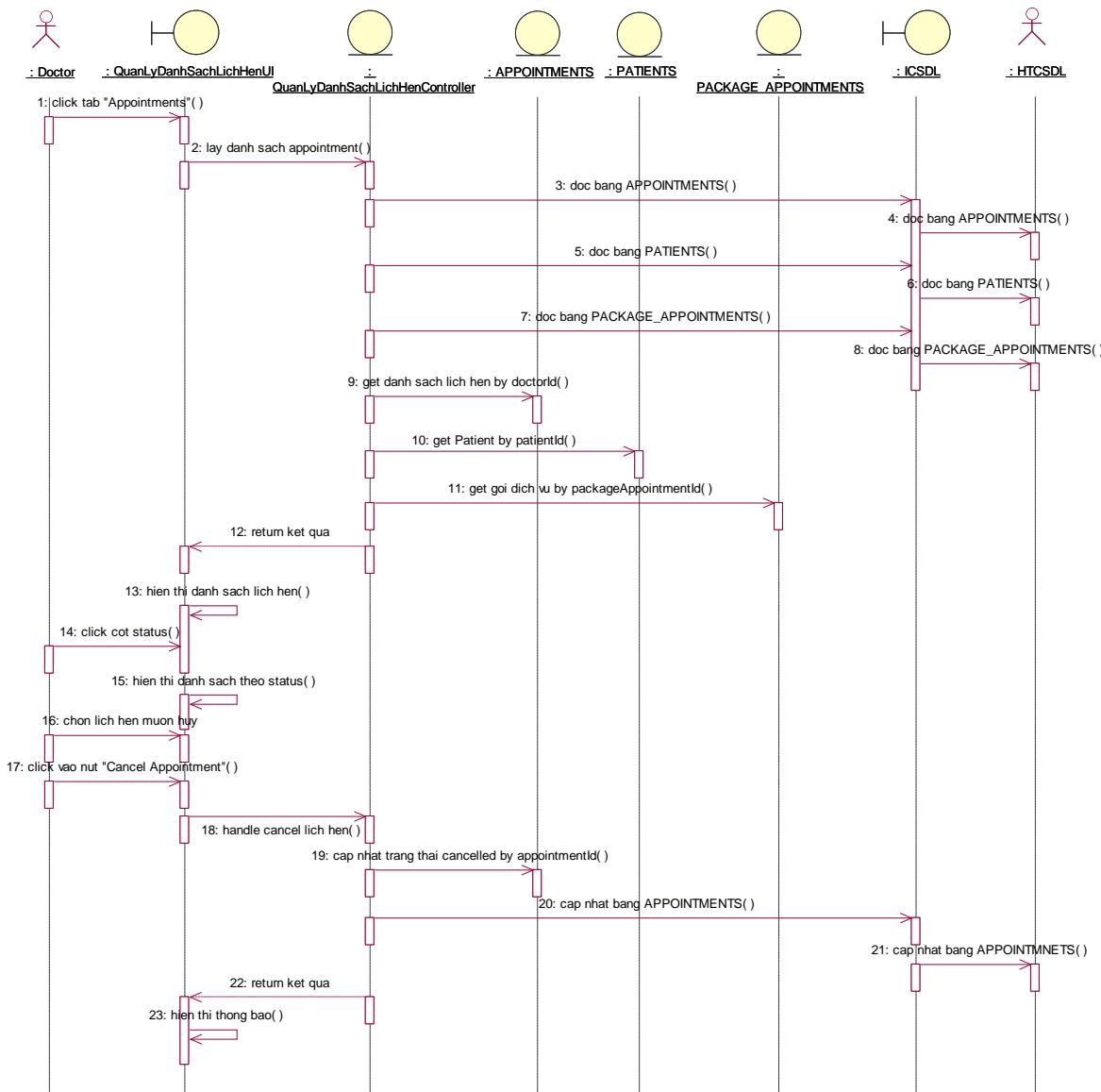
3. Luồng sự kiện

3.1. Luồng cơ bản

1. Use case bắt đầu khi bác sĩ nhấn vào lịch hẹn có gói dịch vụ là nhắn tin tại màn hình danh sách lịch hẹn. Hệ thống lấy thông tin chi tiết của các lịch hẹn của bệnh nhân đó: id, date, status, description, doctor, patient, package appointment, duration từ bảng APPOINTMENTS, PATIENTS và PACKAGE_APPOINTMENTS trong cơ sở dữ liệu và hiển thị lên màn hình.
2. Bác sĩ nhấn button “Message” tại màn chi tiết lịch hẹn. Hệ thống sẽ lấy patientId và doctorId sau đó tạo 1 channel với patientId và doctorId sau đó giữ kết nối giữa patientId và doctorId. Chuyển tới màn hình nhắn tin.
3. Bệnh nhân và bác sĩ trao đổi với nhau cho đến khi 1 trong 2 bên click nút “Complete” ở góc trên phải. Hệ thống thay đổi status của lịch hẹn từ “upcoming” thành “completed” đồng thời chuyển màn hình về danh sách lịch hẹn.
4. Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
4. Các yêu cầu đặc biệt: Không có
5. Tiền điều kiện: Bác sĩ đã đăng nhập vào hệ thống.
6. Hậu điều kiện: Không có.
- c. Biểu đồ trình tự



Hình 3. 55 Biểu đồ trình tự chức năng Nhắn tin với bệnh nhân
d. Biểu đồ VOPC



Hình 3. 56 Biểu đồ VOPC chức năng nhắn tin với bệnh nhân

3.3.19. Use case Xem phản hồi đánh giá

a. Biểu đồ use case



Hình 3. 57 Biểu đồ use case Xem phản hồi đánh giá

b. Đặc tả use case:

1. Tên use case: Use case xem phản hồi đánh giá
2. Mô tả văn tắt

Use case này cho phép bác sĩ xem phản hồi đánh giá của bệnh nhân.

3. Luồng sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi bác sĩ ấn chọn vào 1 lịch hẹn đã hoàn thành ở màn hình danh sách lịch hẹn. Hệ thống sẽ lấy dữ liệu của lịch hẹn đó: id, date, rating, comment,... ở bảng APPOINTMENTS và FEEDBACKS và hiển thị lên màn hình.

2. Use case kết thúc.

3.2. Các luồng rẽ nhánh

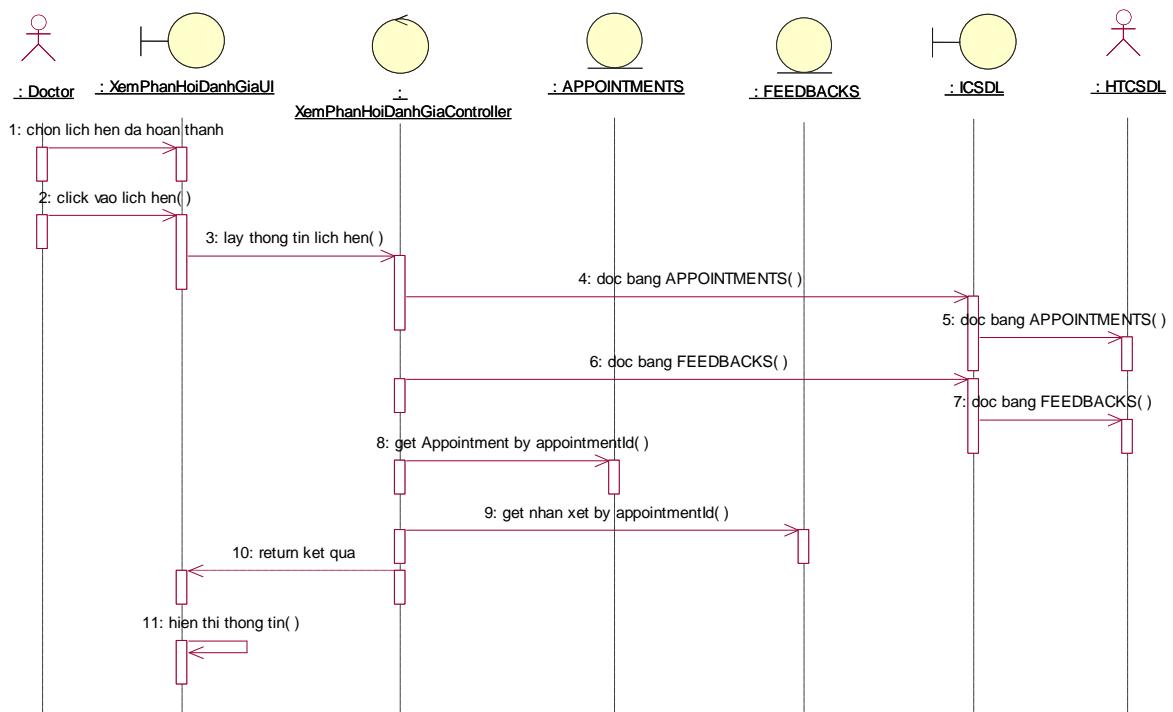
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Bác sĩ đã đăng nhập vào hệ thống.

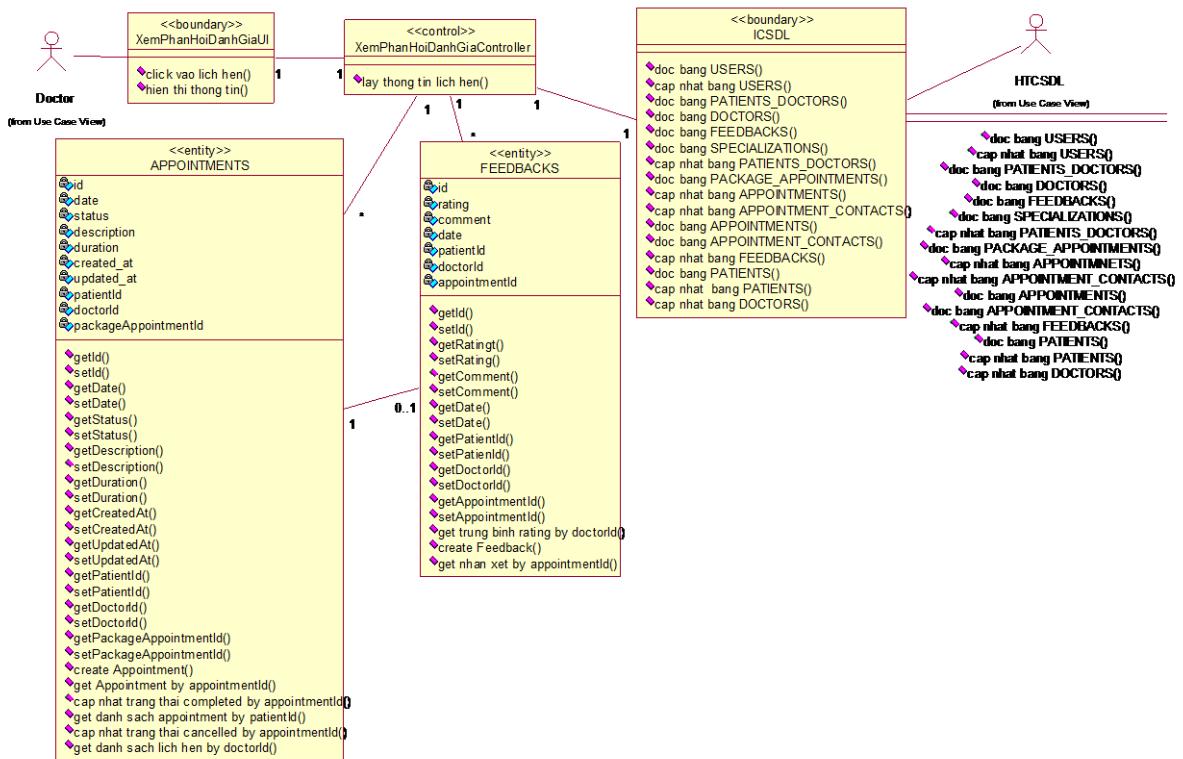
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự



Hình 3. 58 Biểu đồ trình tự chức năng Xem phản hồi đánh giá

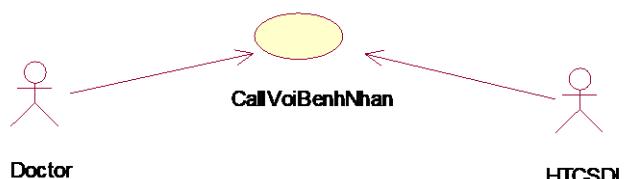
d. Biểu đồ VOPC



Hình 3. 59 Biểu đồ VOPC chức năng Xem phản hồi đánh giá

3.3.20. Use case Call với bệnh nhân

a. Biểu đồ use case



Hình 3. 60 Biểu đồ use case chức năng Call với bệnh nhân

b. Đặc tả use case:

- Tên use case: Use case Call với bệnh nhân

- Mô tả văn tắt

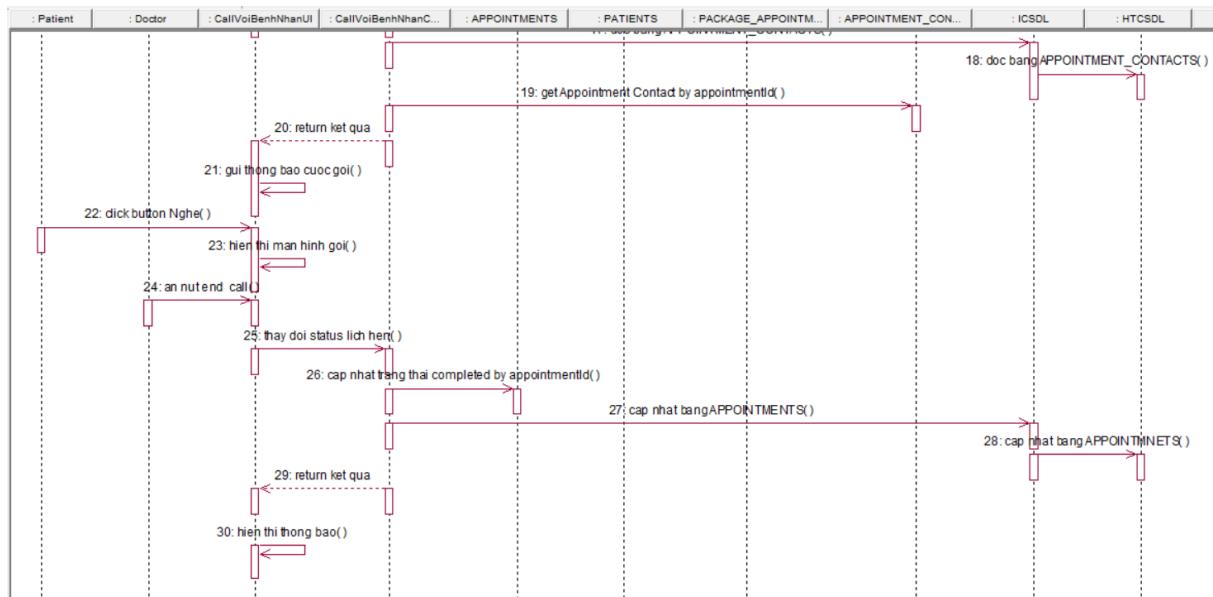
Use case này cho phép bác sĩ call trực tiếp với bệnh nhân

- Luồng sự kiện

- Luồng cơ bản

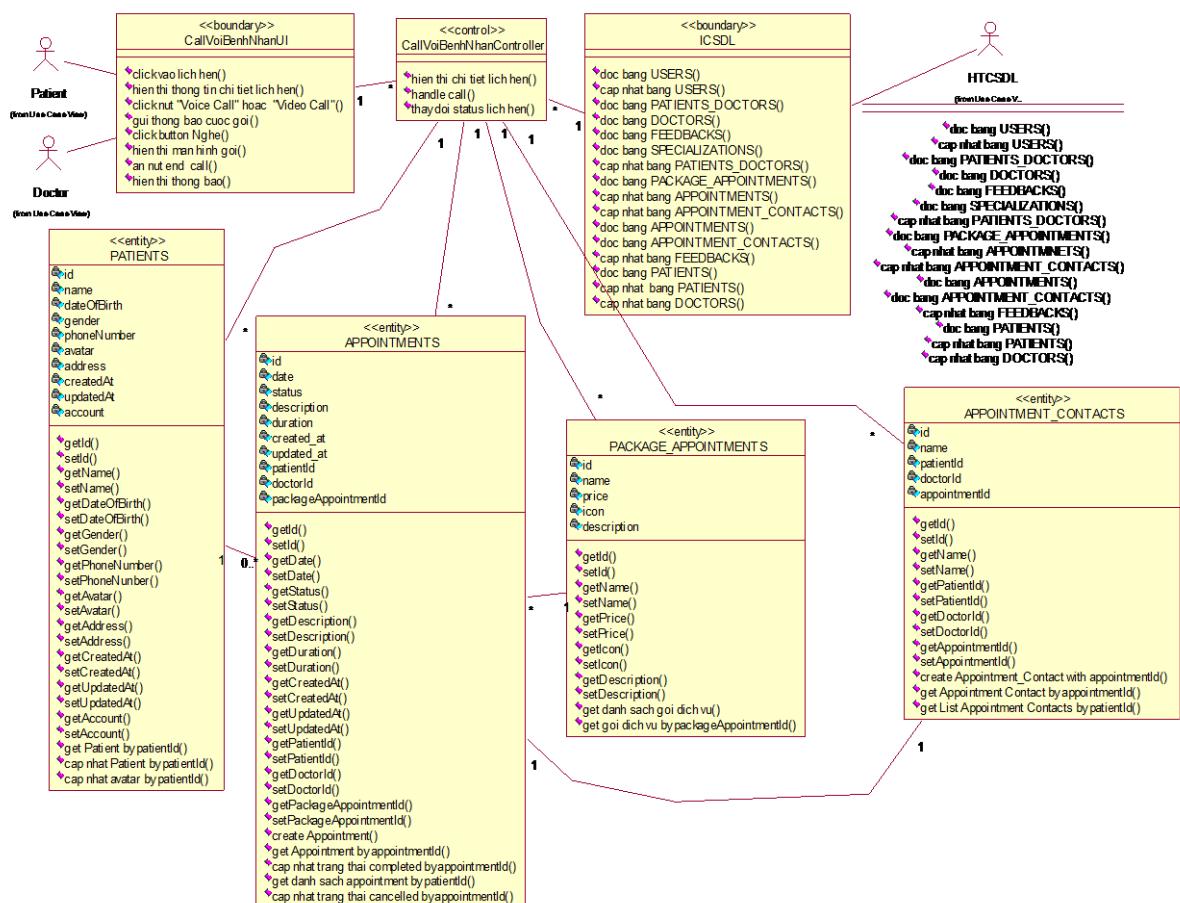
1. Use case bắt đầu khi bác sĩ chọn lịch hẹn có gói dịch vụ là call tại màn hình danh sách lịch hẹn. Hệ thống lấy thông tin chi tiết lịch hẹn của bác sĩ đó: id, date, status, description, patient, package appointment, duration từ bảng APPOINTMENTS, PATIENTS và PACKAGE_APPOINTMENTS trong cơ sở dữ liệu và hiển thị lên màn hình.
 2. Bác sĩ nhấn nút “Voice Call” hoặc “Video Call” để tiến hành cuộc gọi. Hệ thống sẽ lấy patientId và doctorId của lịch hẹn đó từ bảng APPOINTMENT_CONTACT để gửi thông báo cuộc gọi tới thiết bị của bác sĩ.
 3. Bệnh nhân ấn vào nghe cuộc gọi để tiến hành call với bác sĩ. Hệ thống sẽ kết nối bệnh nhân và bác sĩ với nhau sau đó chuyển qua màn hình call.
 4. Sau khi quá trình call hoàn tất, bác sĩ tiến hành ấn nút end call ở thanh điều khiển cuộc gọi. Hệ thống ngắt kết nối người dùng đó khỏi cuộc gọi. Thay đổi status của lịch hẹn thành “completed” và hiển thị thông báo.
 5. Use case kết thúc.
- ### 3.2. Các luồng rẽ nhánh
1. Tại bước 3 của luồng cơ bản, nếu bệnh nhân ấn không nghe cuộc gọi hoặc thời gian chờ nghe quá 1 phút thì cuộc gọi kết thúc. Use case kết thúc.
 2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
 4. Các yêu cầu đặc biệt: Không có
 5. Tiền điều kiện: Bác sĩ đã đăng nhập vào hệ thống.
 6. Hậu điều kiện: Không có.
- c. Biểu đồ trình tự





Hình 3. 61 Biểu đồ trình tự chức năng Call với bệnh nhân

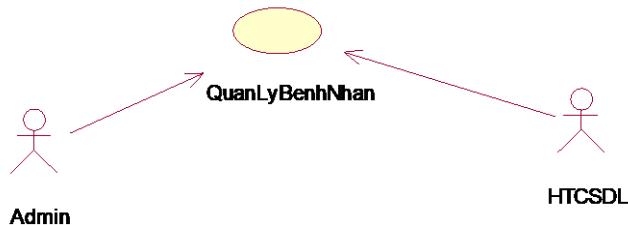
d. Biểu đồ VOPC



Hình 3. 62 Biểu đồ VOPC chức năng Call với bệnh nhân

3.3.21. Use case Quản lý bệnh nhân

a. Biểu đồ use case



Hình 3. 63 Biểu đồ use case chức năng Quản lý bệnh nhân

b. Đặc tả use case:

1. Tên use case: Use case Quản lý bệnh nhân
2. Mô tả văn tắt

Use case này cho phép quản trị viên xem danh sách, tìm kiếm bệnh nhân và xóa bệnh nhân.

3. Luồng sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên click vào nút “Manage Patient”. Hệ thống sẽ lấy thông tin chi tiết của các bệnh nhân: id, name, email, address, birthdate, gender, avatar từ bảng USERS và PATIENTS trong CSDL và hiển thị lên màn hình.

2. Tìm kiếm bệnh nhân:

- a) Quản trị viên nhập từ khóa ở cột nhập tìm kiếm tên bệnh nhân muốn tìm kiếm và ấn nút “Enter” trên màn hình.

- b) Hệ thống sẽ truy cập và kiểm tra thông tin trong bảng PATIENTS và hiển thị thông tin bệnh nhân cần tìm ra màn hình.

3. Xóa bệnh nhân:

- a) Quản trị viên chọn bệnh nhân muốn xóa, và click vào button “Delete” tại bệnh nhân đó. Hệ thống sẽ thông báo có chắc chắn muốn xóa bệnh nhân này.

b) Quản trị viên chọn “OK” để xóa bệnh nhân này ở bảng PATIENTS và USERS, sau đó cập nhật lại danh sách bệnh nhân.

4. Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bước 3b, nếu quản trị viên chọn “Cancelled” thì use case kết thúc.

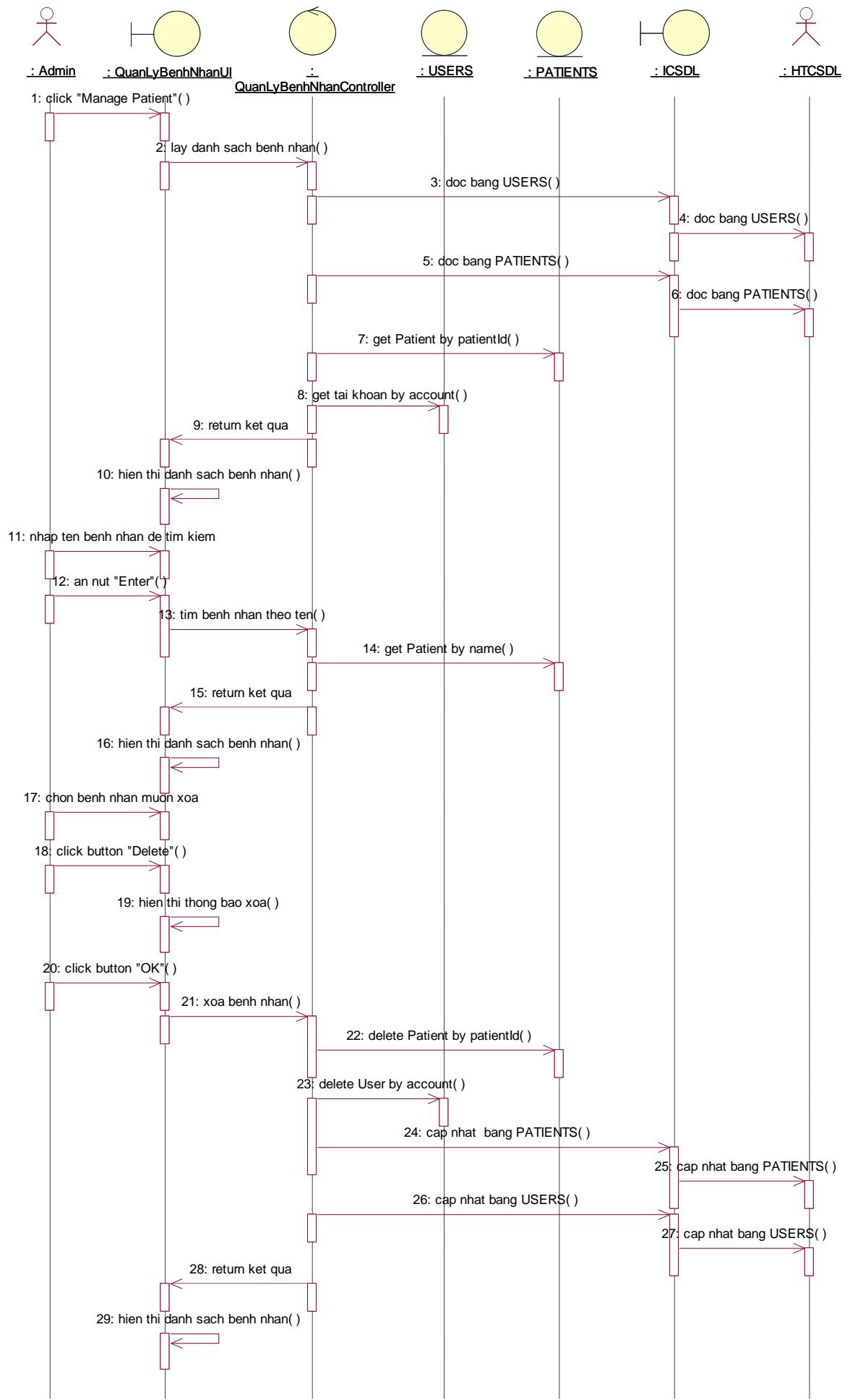
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Quản trị viên đã đăng nhập vào hệ thống.

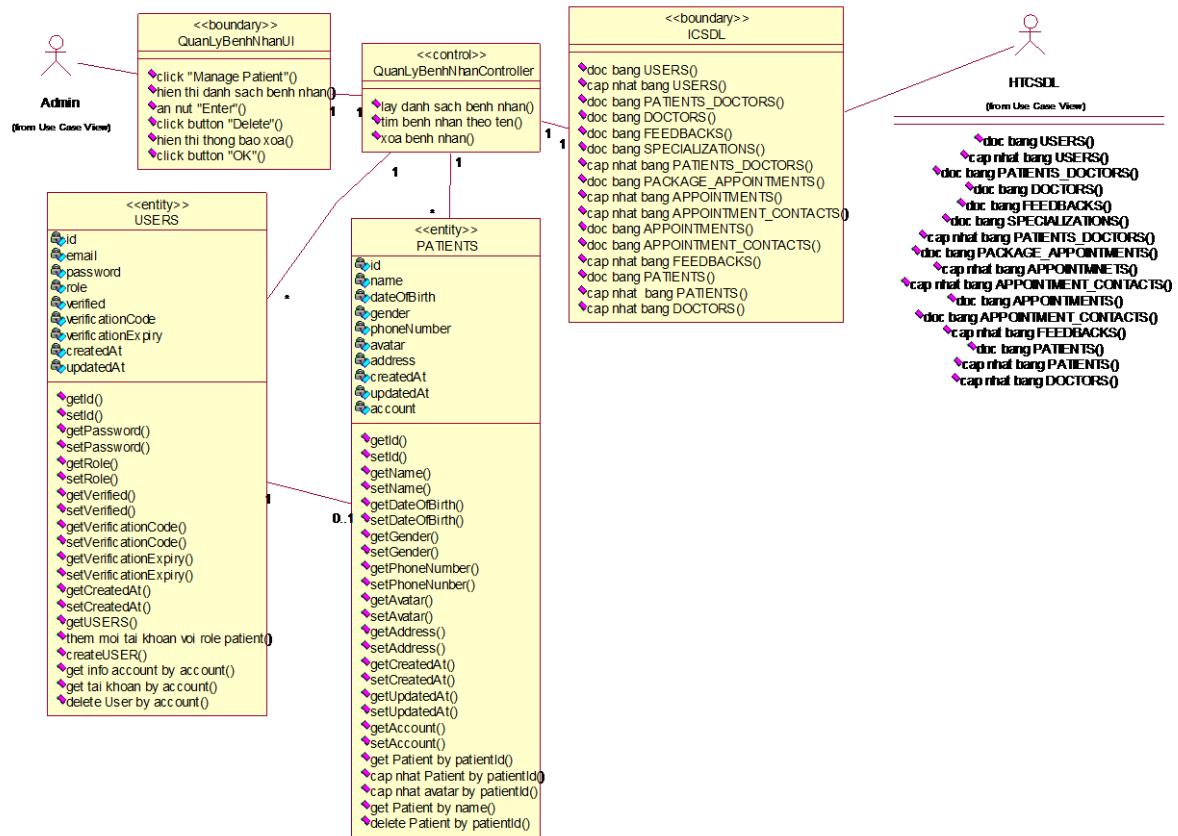
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự



Hình 3. 64 Biểu đồ trình tự chức năng Quản lý bệnh nhân

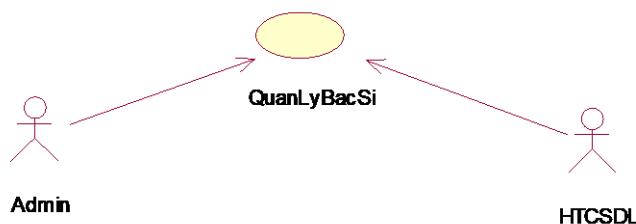
d. Biểu đồ VOPC



Hình 3. 65 Biểu đồ VOPC chức năng Quản lý bệnh nhân

3.3.22. Use case Quản lý bác sĩ

a. Biểu đồ use case



Hình 3. 66 Biểu đồ use case chức năng Quản lý bác sĩ

b. Đặc tả use case:

- Tên use case: Use case Quản lý bác sĩ
- Mô tả văn tắt

Use case này cho phép quản trị viên xem danh sách, tìm kiếm bác sĩ và xóa bác sĩ.

3. Luồng sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên click vào nút “Manage Doctor”. Hệ thống sẽ lấy thông tin chi tiết của các bác sĩ: id, name, email, address, birthdate, gender, avatar, specialization, hospital từ bảng USERS và DOCTORS, SPECIALIZATIONS trong CSDL và hiển thị lên màn hình.

2. Tìm kiếm bác sĩ:

- a) Quản trị viên nhập từ khóa ở cột nhập tìm kiếm tên bác sĩ muốn tìm kiếm và ấn nút “Search” trên màn hình.

- b) Hệ thống sẽ truy cập và kiểm tra thông tin trong bảng DOCTORS và hiển thị thông tin bác sĩ cần tìm ra màn hình.

3. Use case kết thúc.

3.2. Các luồng rẽ nhánh

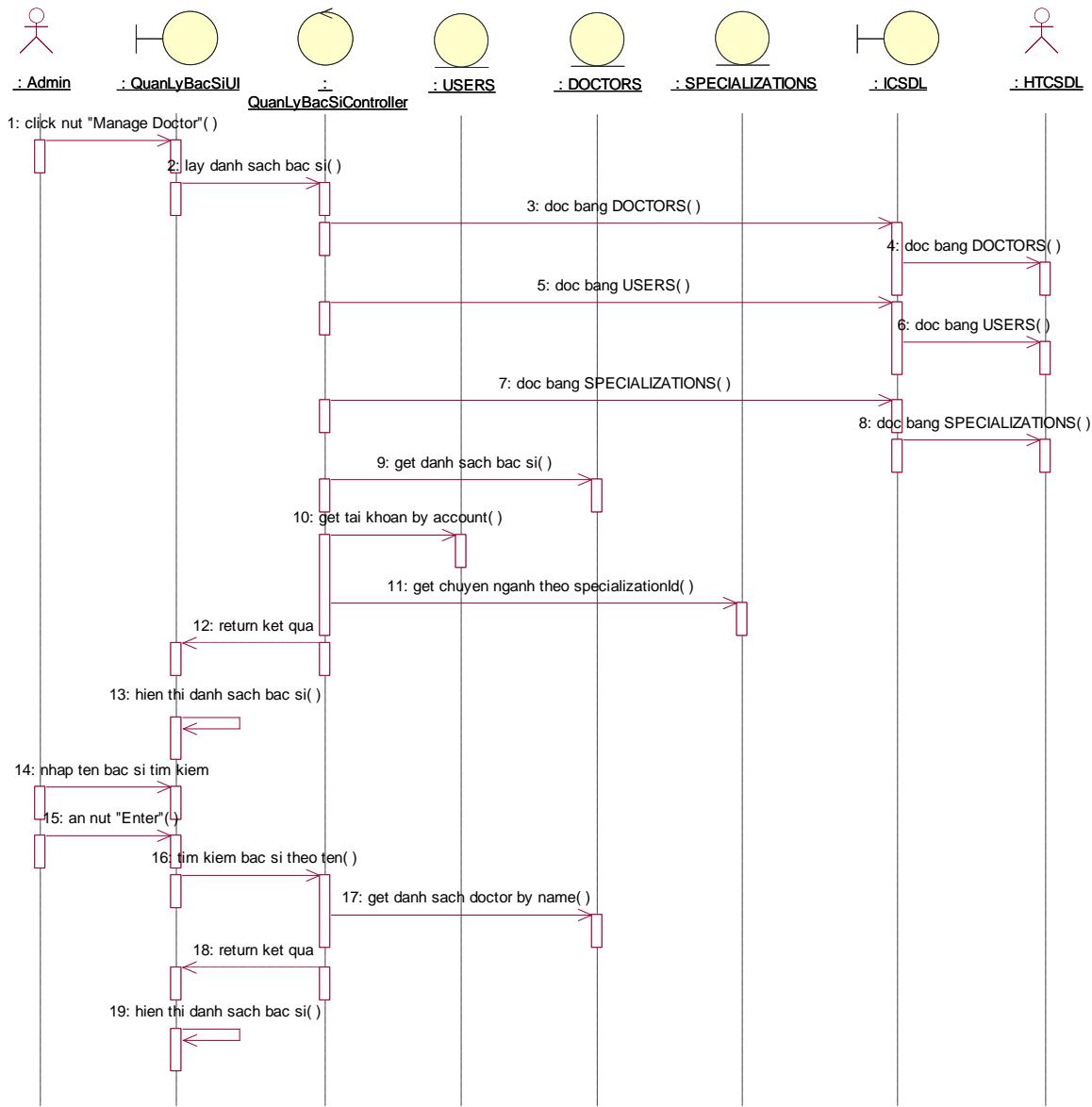
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Quản trị viên đã đăng nhập vào hệ thống.

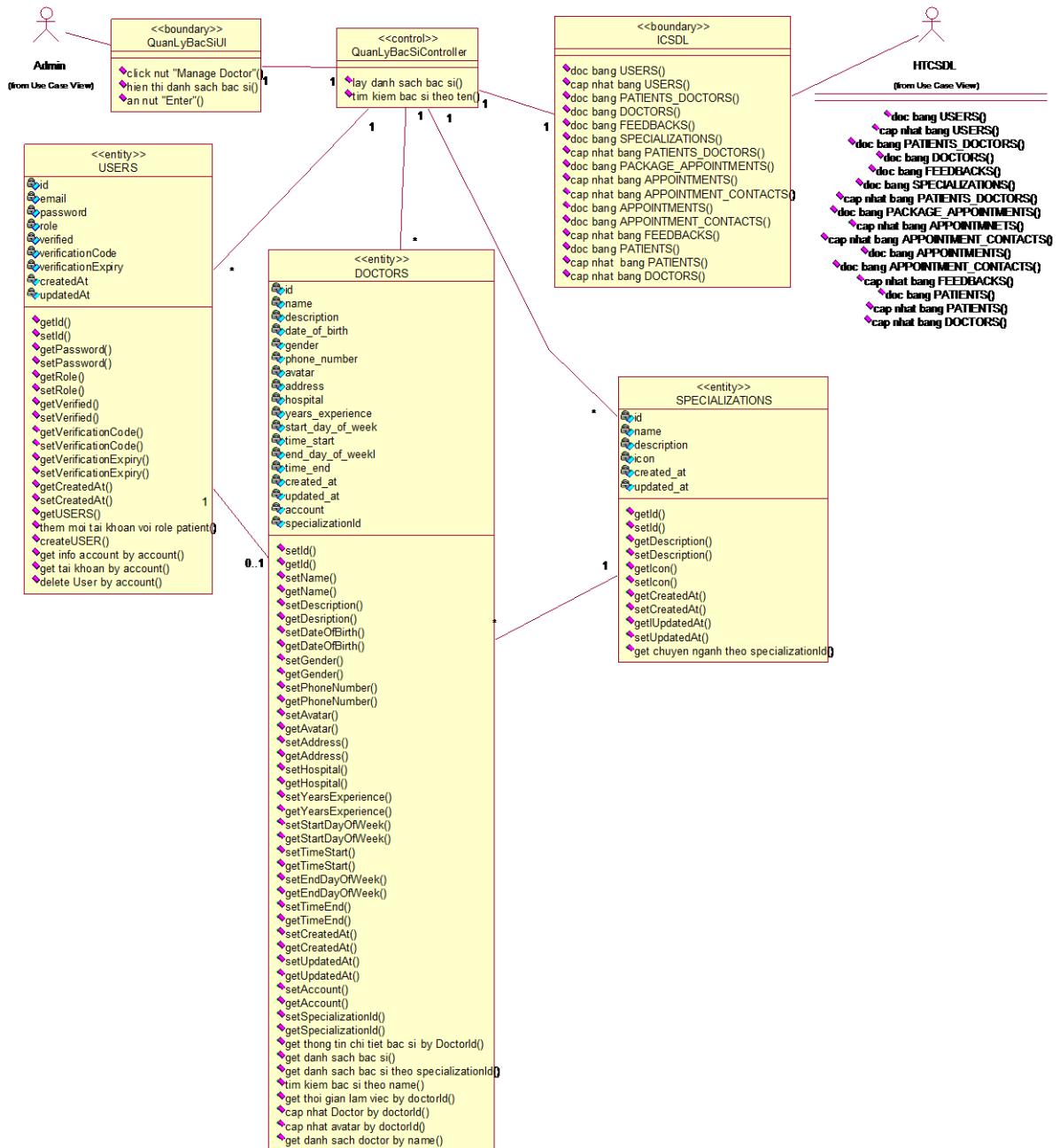
6. Hậu điều kiện: Không có.

- c. Biểu đồ trình tự



Hình 3. 67 Biểu đồ trình tự chức năng Quản lý bác sĩ

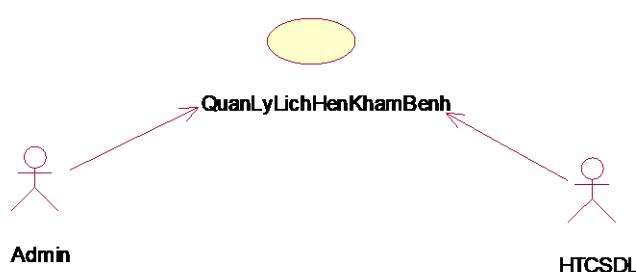
d. Biểu đồ VOPC



Hình 3. 68 Biểu đồ VOPC chúc năng Quản lý bác sĩ

3.3.23. Use case Quản lý lịch hẹn khám bệnh

a. Biểu đồ use case



Hình 3. 69 Biểu đồ use case Quản lý lịch hẹn khám bệnh

b. Đặc tả use case:

1. Tên use case: Use case Quản lý lịch hẹn khám bệnh
2. Mô tả văn tắt

Use case này cho phép quản trị viên xem danh sách lịch khám, tìm kiếm lịch khám bệnh theo ngày, filter lịch hẹn khám bệnh theo status.

3. Luồng sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên click vào nút “Manage Appointment”. Hệ thống sẽ lấy thông tin: id, date, doctor, patient, status, duration, package appointment từ bảng APPOINTMENTS, DOCTORS, PATIENTS, PACKAGE_APPOINTMENTS và hiển thị lên màn hình.

2. Tìm kiếm lịch khám bệnh theo ngày:

- a) Quản trị viên chọn ngày tháng ở cột nhập ngày tháng, sau đó ấn “Enter”.

- b) Hệ thống sẽ lọc ra những lịch hẹn thuộc ngày đó từ bảng APPOINTMENTS và hiển thị lên màn hình.

3. Filter lịch hẹn khám bệnh theo status:

- a) Quản trị viên click vào ô filter lịch hẹn khám bệnh theo status và chọn status mình muốn (upcoming, completed, cancelled).

- b) Hệ thống sẽ lấy danh sách những lịch hẹn theo status đó từ bảng APPOINTMENTS và hiển thị lên màn hình.

4. Use case kết thúc

3.2. Các luồng rẽ nhánh

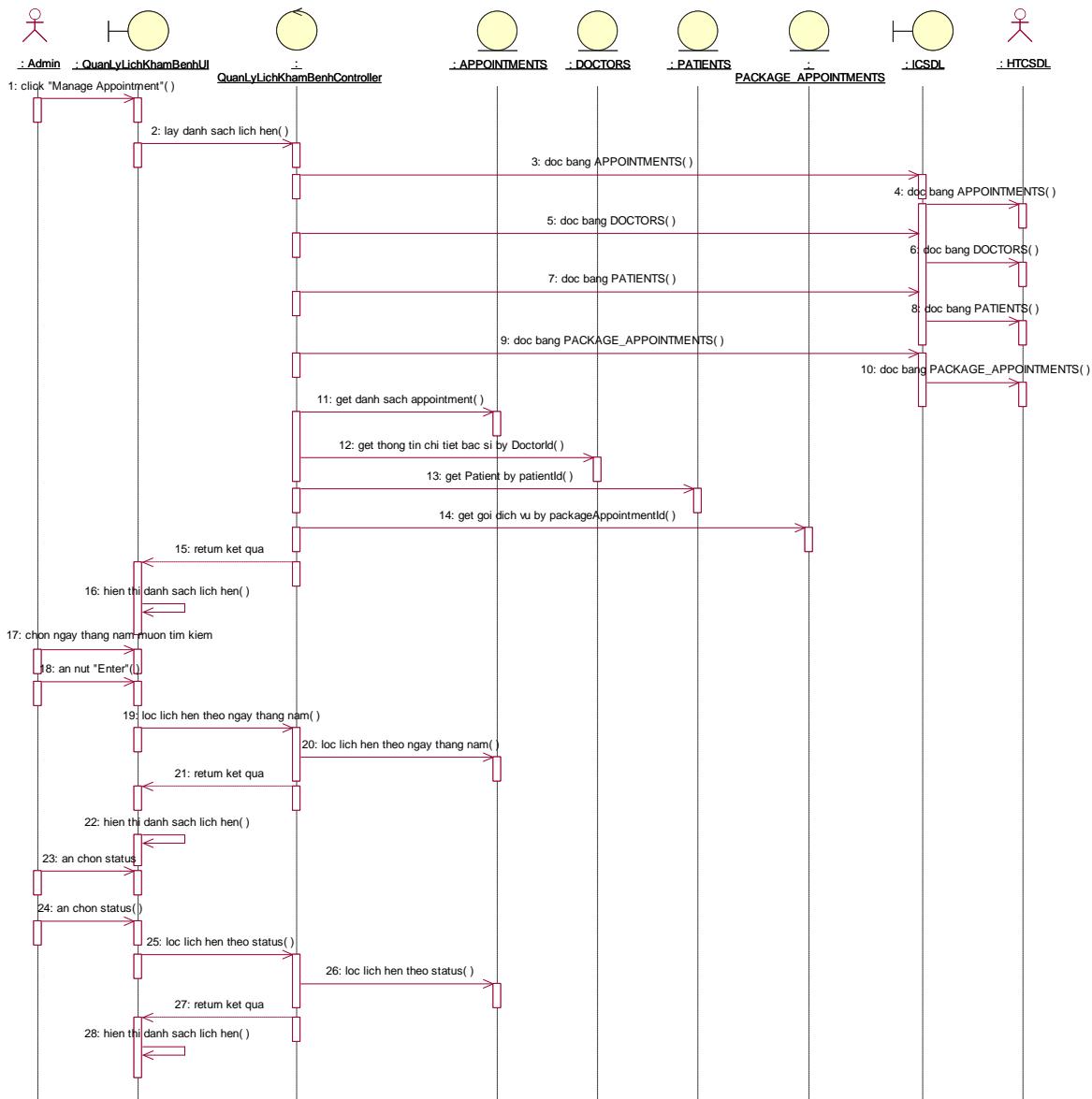
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Quản trị viên đã đăng nhập vào hệ thống.

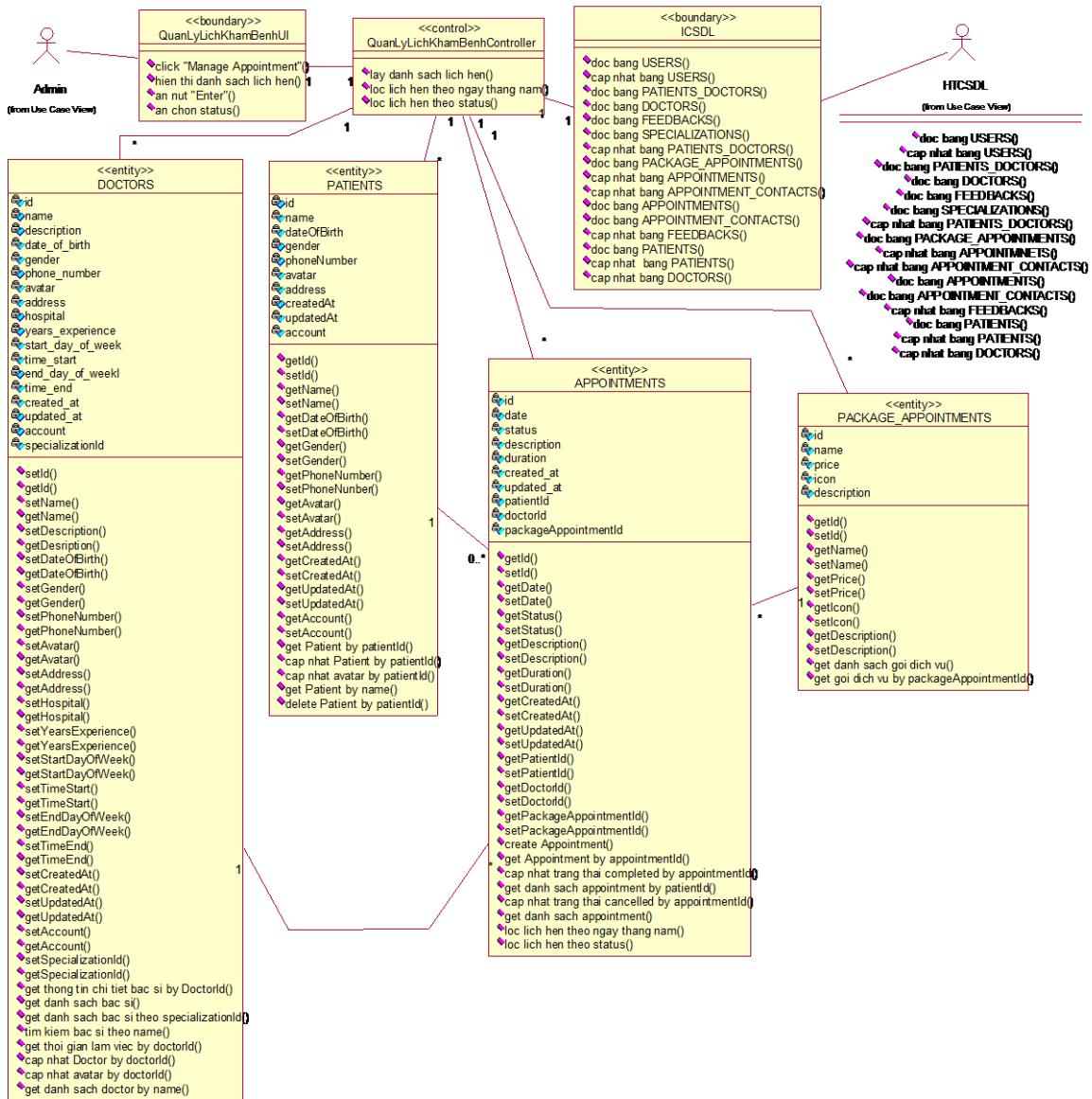
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự



Hình 3. 70 Biểu đồ trình tự chức năng Quản lý lịch hẹn khám bệnh

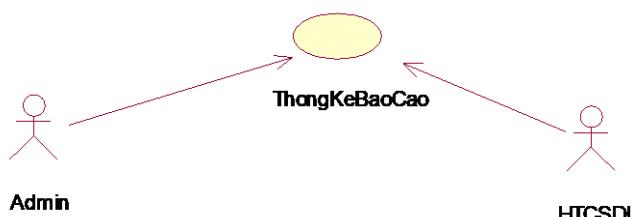
d. Biểu đồ VOPC



Hình 3. 71 Biểu đồ VOPC chúc năng Quản lý lịch hẹn khám bệnh

3.3.24. Use case Thống kê báo cáo

a. Biểu đồ use case



Hình 3. 72 Biểu đồ use case chúc năng Thống kê báo cáo

b. Đặc tả use case:

1. Tên use case: Use case Thống kê báo cáo

2. Mô tả văn tắt

Use case này cho phép quản trị viên xem thông kê báo cáo.

3. Luồng sự kiện

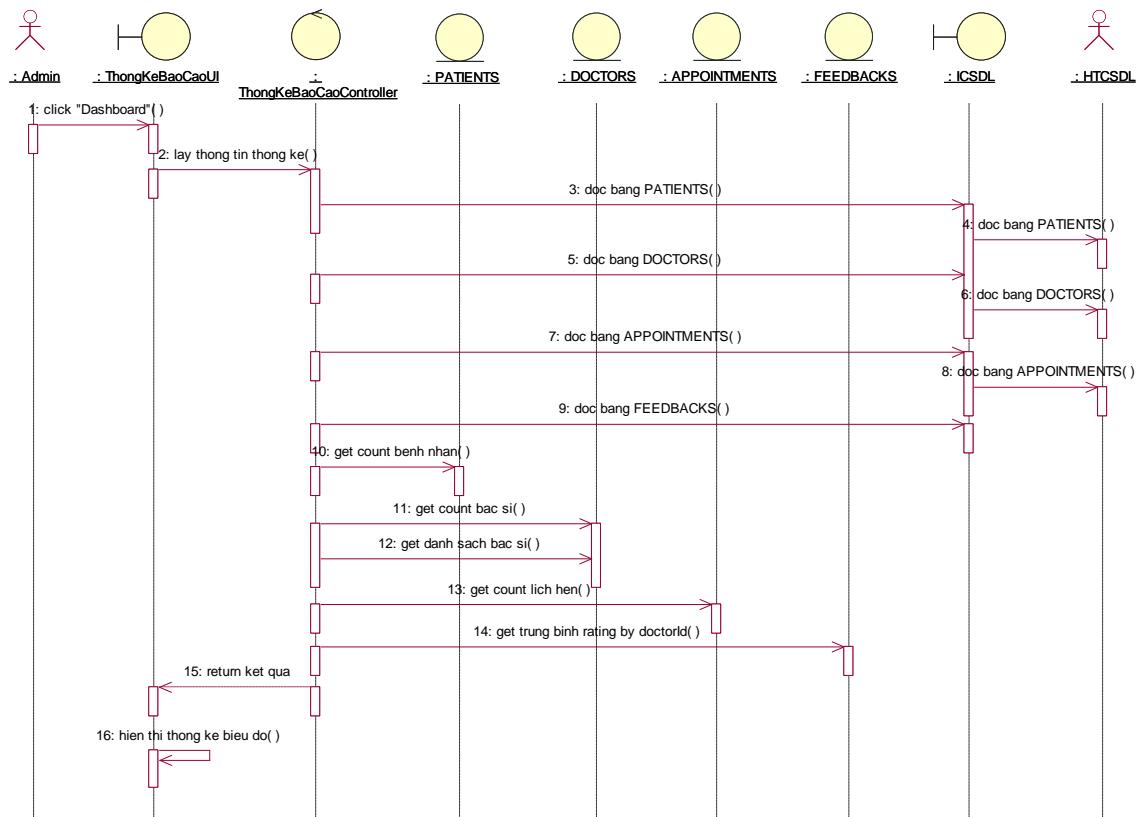
3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên click vào “Dashboard” ở màn hình. Hệ thống sẽ lấy ra tổng bệnh nhân, tổng bác sĩ, tổng số lượng lịch hẹn khám, tổng số lượng lịch hẹn khám mới, tổng số lượng lịch khám upcoming, tổng số lượng lịch khám completed, tổng số lượng lịch khám cancelled, top bác sĩ, ... từ các bảng PATIENTS, DOCTORS, APPOINTMENTS, FEEDBACKS... sau đó hiển thị ra màn hình dưới dạng các biểu đồ trực quan.

3.2. Các luồng rẽ nhánh

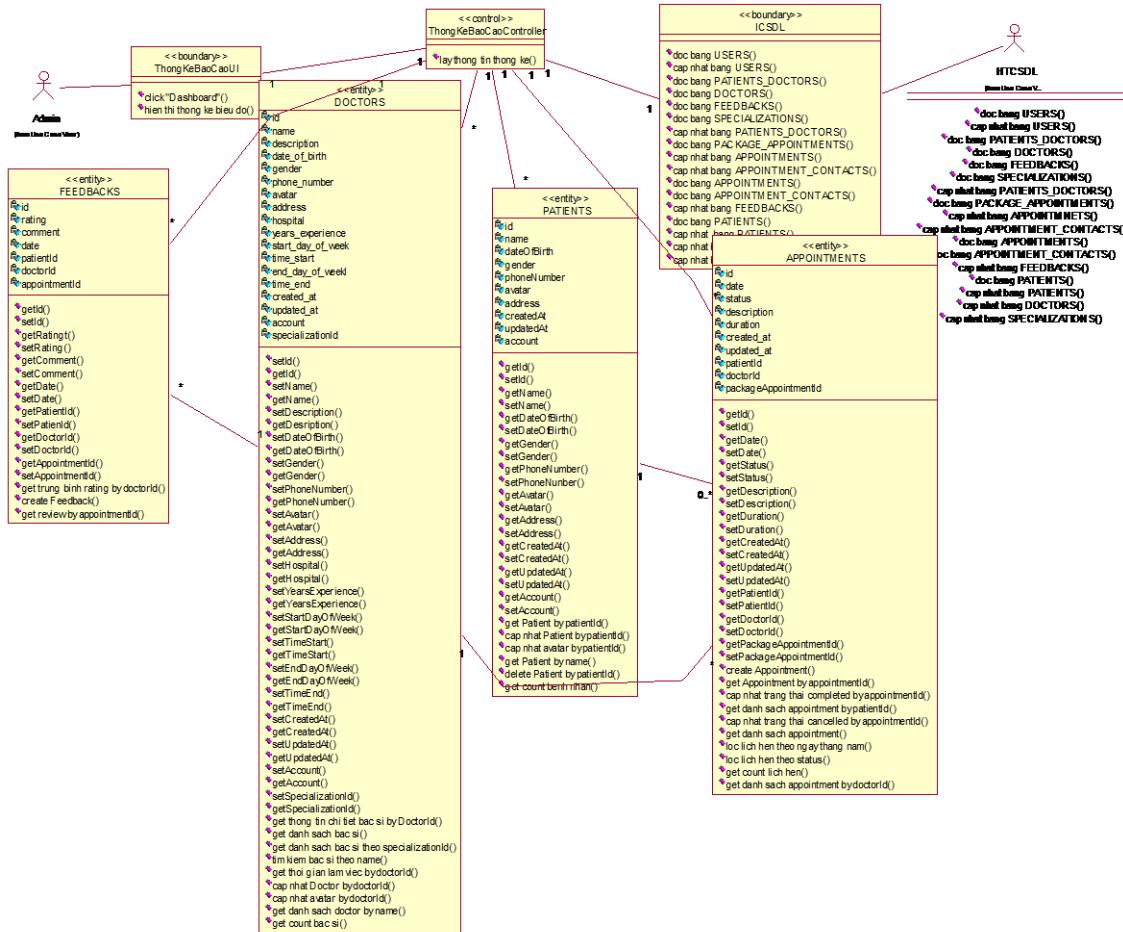
1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có
 5. Tiền điều kiện: Quản trị viên đã đăng nhập vào hệ thống.
 6. Hậu điều kiện: Không có.
- c. Biểu đồ trình tự



Hình 3. 73 Biểu đồ trình tự chức năng Thông kê báo cáo

d. Biểu đồ VOPC



Hình 3. 74 Biểu đồ VOPC chức năng Thông kê báo cáo

3.3.25. Use case Quản lý chuyên ngành

a. Biểu đồ use case



Hình 3. 75 Biểu đồ use case chức năng Quản lý chuyên ngành

b. Đặc tả use case:

- Tên use case: Use case Quản lý chuyên ngành
- Mô tả vắn tắt

Use case này cho phép quản trị viên quản lý chuyên ngành của bác sĩ.

3. Luồng sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên click vào “Manage Specialization”. Hệ thống sẽ lấy danh sách các chuyên ngành: id, name, icon, description từ bảng SPECIALIZATIONS và hiển thị lên màn hình.

2. Thêm chuyên ngành mới:

a) Admin click vào button “New” trên màn hình. Hệ thống sẽ hiển thị form nhập thêm chuyên ngành mới.

b) Admin nhập thông tin: tên, ảnh, description và ấn “Save”. Hệ thống sẽ thêm mới chuyên ngành vào CSDL. Sau đó hiển thị lại danh sách được cập nhật.

3. Use case kết thúc

3.2. Các luồng rẽ nhánh

1. Tại bước 2b nếu admin click “Close modal” thì use case kết thúc.

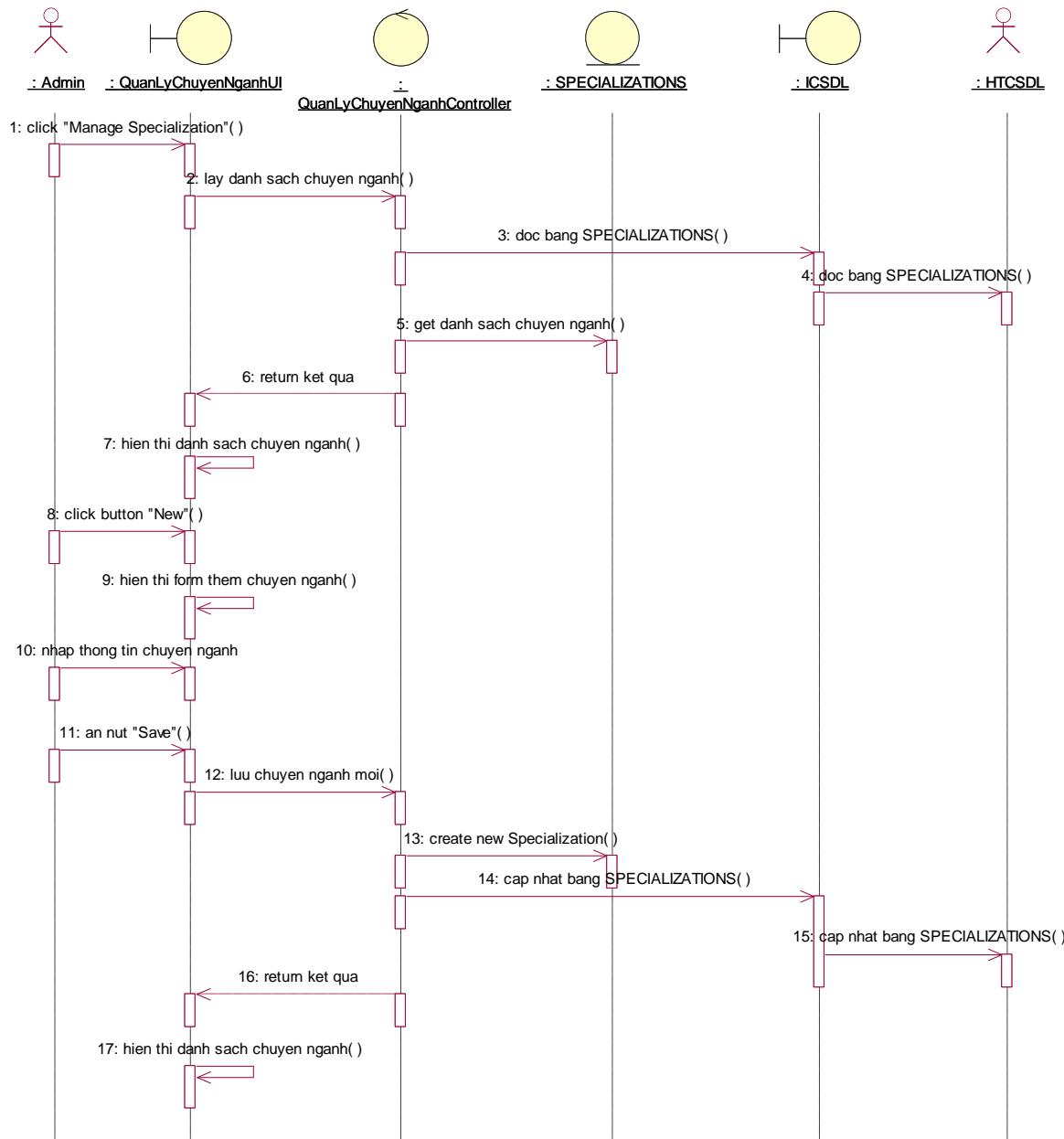
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Quản trị viên đã đăng nhập vào hệ thống.

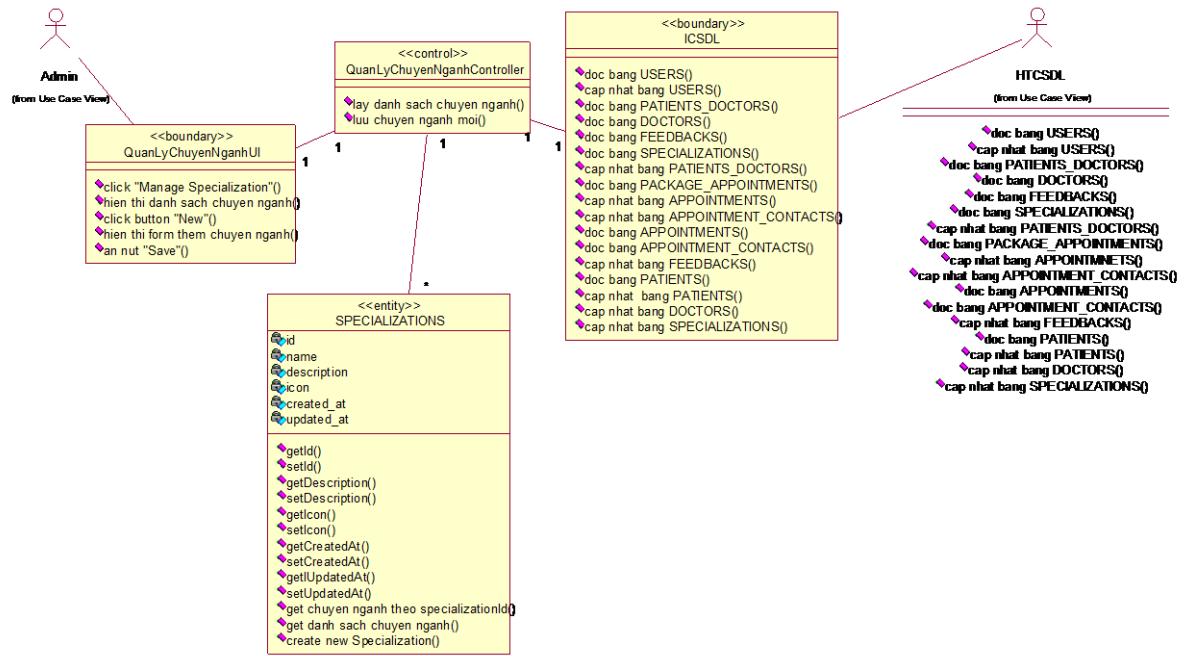
6. Hậu điều kiện: Không có.

c. Biểu đồ trình tự



Hình 3. 76 Biểu đồ trình tự chức năng Quản lý chuyên ngành

d. Biểu đồ VOPC



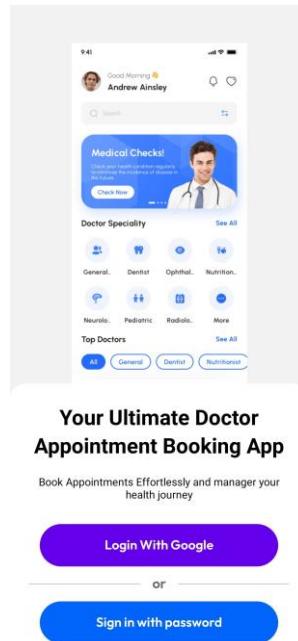
Hình 3. 77 Biểu đồ VOPC chirc năng Quản lý chuyên ngành

CHƯƠNG 4 KẾT QUẢ ĐẠT ĐƯỢC

4.1. Giao diện chung

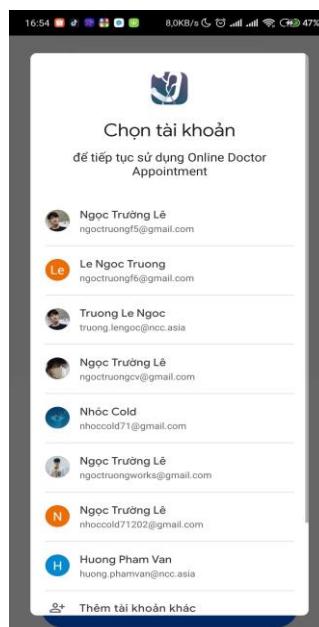
Giao diện ứng dụng của bệnh nhân và hướng dẫn sử dụng

- Click vào Login With Google để dùng app với tư cách là bệnh nhân



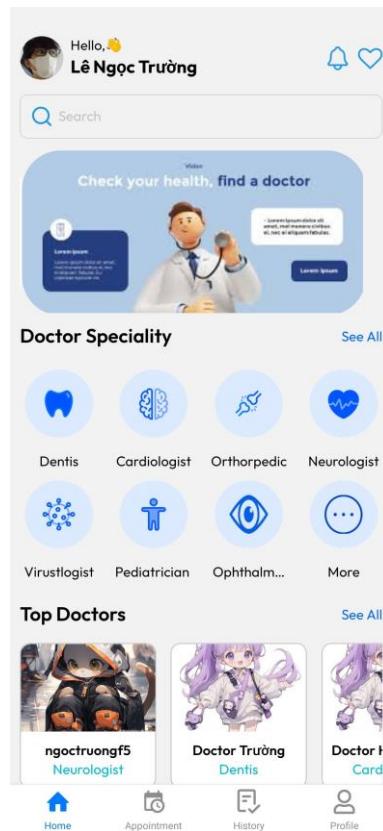
Hình 4. 1 Giao diện login

- Chọn tài khoản google muốn đăng nhập



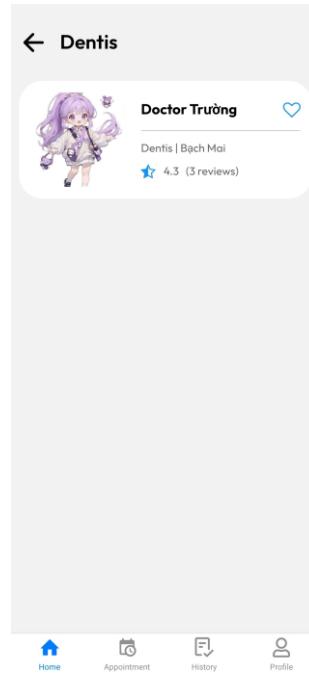
Hình 4. 2 Giao diện chọn tài khoản

- Tại màn hình Home, bệnh nhân có thể ấn vào danh sách top bác sĩ hoặc tìm bác sĩ theo chuyên ngành hoặc tìm kiếm theo tên.



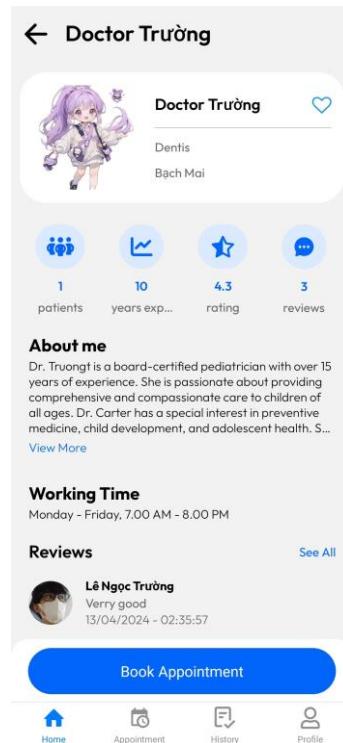
Hình 4. 3 Giao diện màn Home

- Bệnh nhân chọn ấn vào icon chuyên ngành để tìm các sĩ theo chuyên ngành
- Tại màn hình danh sách chuyên ngành bệnh nhân click vào 1 bác sĩ



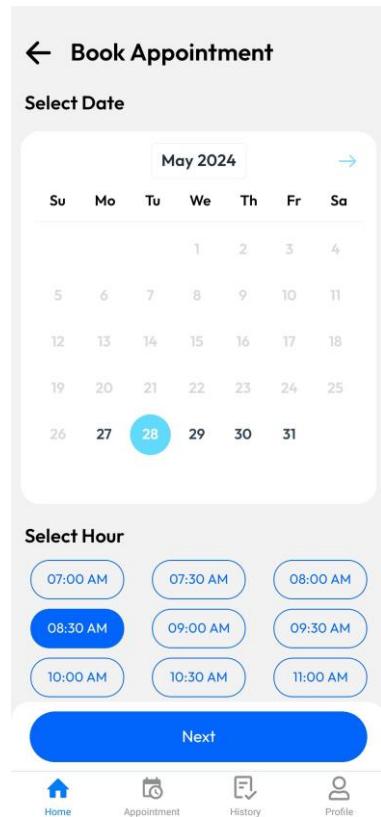
Hình 4. 4 Giao diện màn danh sách bác sĩ theo chuyên ngành

- Tại màn hình thông tin chi tiết về bác sĩ. Bệnh nhân nhấn nút “Book Appointment”



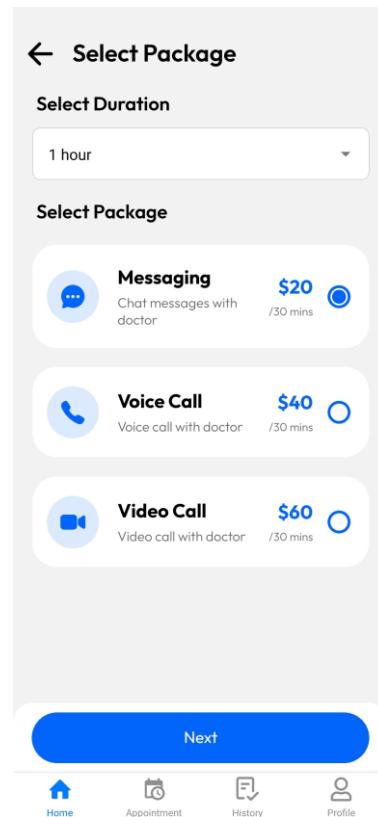
Hình 4. 5 Giao diện màn chi tiết bác sĩ

- Bệnh nhân chọn ngày và giờ hẹn lịch sau đó ấn “Next”



Hình 4. 6 Giao diện màn chọn ngày giờ đặt lịch

- Bệnh nhân chọn thời lượng khám và gói dịch vụ khám sau đó ấn “Next”



Hình 4. 7 Giao diện màn chọn gói dịch vụ và thời lượng

- Bệnh nhân xem thông tin cá nhân và nhập vấn đề đang gặp phải sau đó ấn “Next”

Patient Details

Full Name
Lê Ngọc Trường

Gender
Male

Your Age
4 years

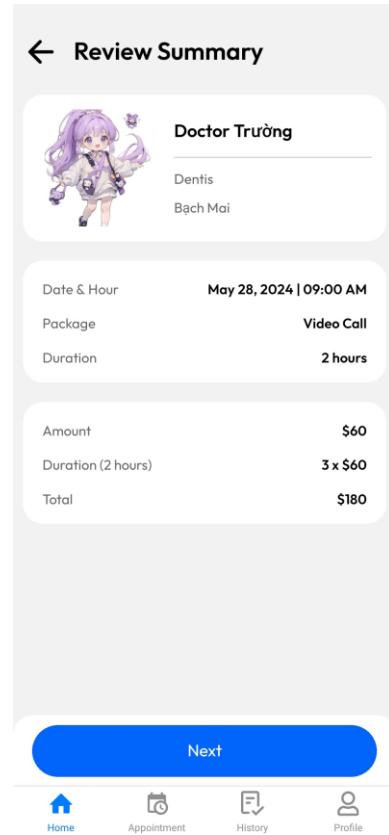
Write Your Problem
Tôi bị đau bụng, sốt cao 39 độ. Đau toàn thân

Next

Home Appointment History Profile

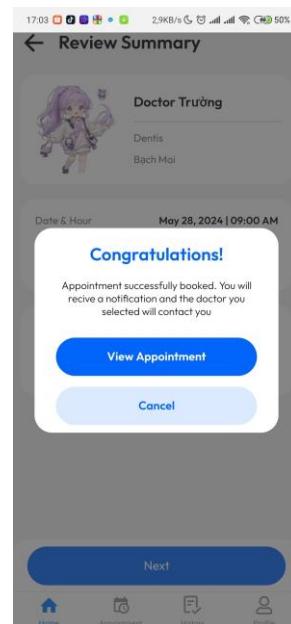
Hình 4. 8 Giao diện màn thông tin bệnh nhân và nhập biểu hiện vấn đề

- Bệnh nhân xem lại nội dung cuộc hẹn 1 lần nữa sau đó ấn “Next”

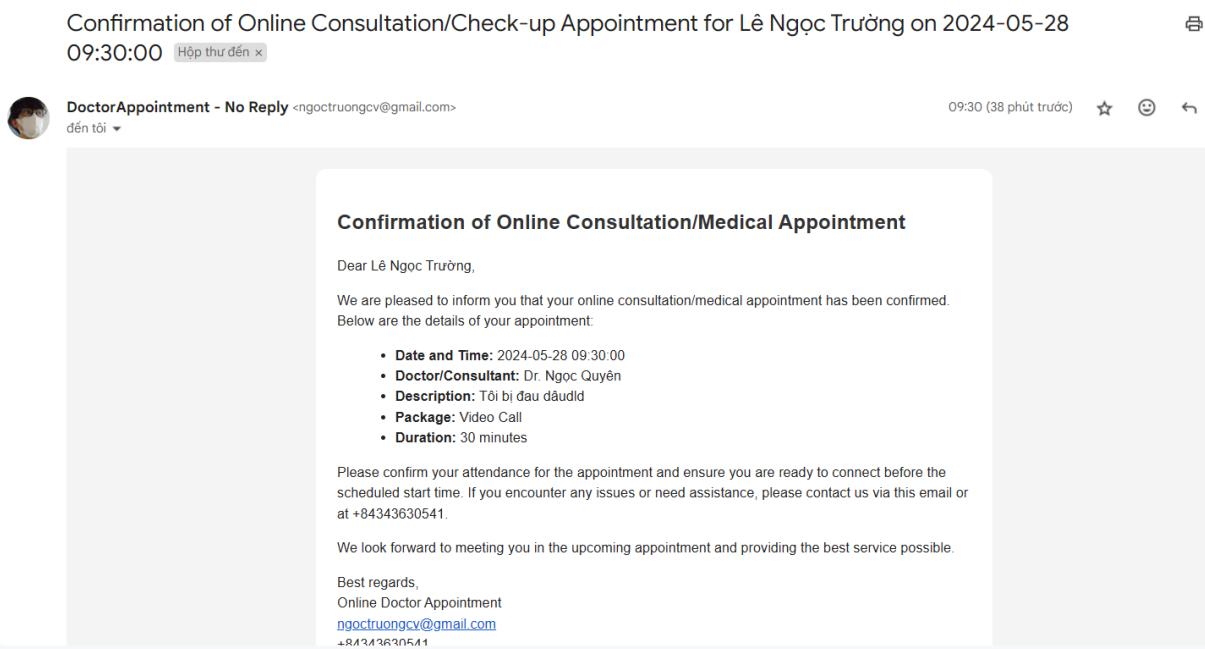


Hình 4. 9 Giao diện màn Review Summary

- Sau khi có thông báo tạo thành công cuộc hẹn. Bệnh nhân ấn vào “View Appointment” để xem lịch hẹn.



Hình 4. 10 Giao diện thông báo đặt lịch thành công và mail thông báo



Hình 4. 11 Mail thông báo cuộc hẹn

- Tại danh sách lịch hẹn, bệnh nhân chọn các tab status để xem các lịch hẹn theo status

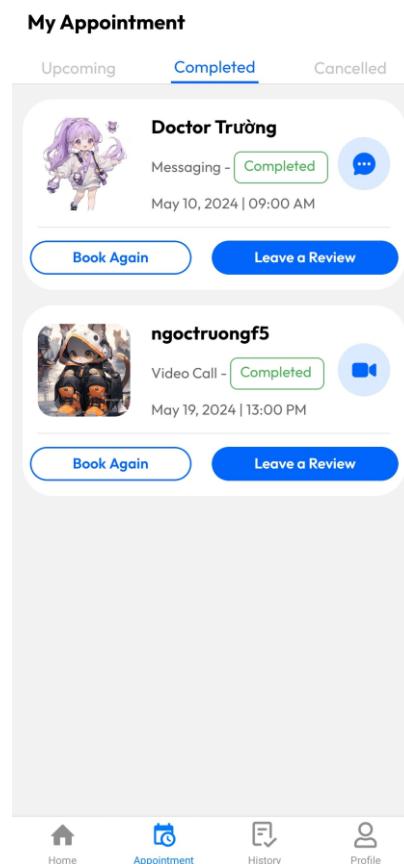
The image displays three separate mobile application screens, each titled 'My Appointment'. Each screen has tabs at the top for 'Upcoming', 'Completed', and 'Cancelled'. The 'Completed' tab is highlighted in the first two screens, while the 'Cancelled' tab is highlighted in the third.

- Upcoming Tab:** Shows three upcoming appointments:
 - Doctor Trưởng: Voice Call - Upcoming on May 13, 2024 | 20:00 PM. Includes a 'Cancel Appointment' button.
 - ngoctruongf5: Messaging - Upcoming on May 15, 2024 | 10:00 AM. Includes a 'Cancel Appointment' button.
 - Doctor Trưởng: Video Call - Upcoming on May 28, 2024 | 09:00 AM. Includes a 'Cancel Appointment' button.
- Completed Tab:** Shows three completed appointments:
 - Doctor Trưởng: Messaging - Completed on May 10, 2024 | 09:00 AM. Includes 'Book Again' and 'Leave a Review' buttons.
 - ngoctruongf5: Video Call - Completed on May 19, 2024 | 13:00 PM. Includes 'Book Again' and 'Leave a Review' buttons.
- Cancelled Tab:** Shows one cancelled appointment:
 - Doctor Trưởng: Messaging - Cancelled on May 14, 2024 | 09:00 AM. Includes 'Book Again' and 'Leave a Review' buttons.

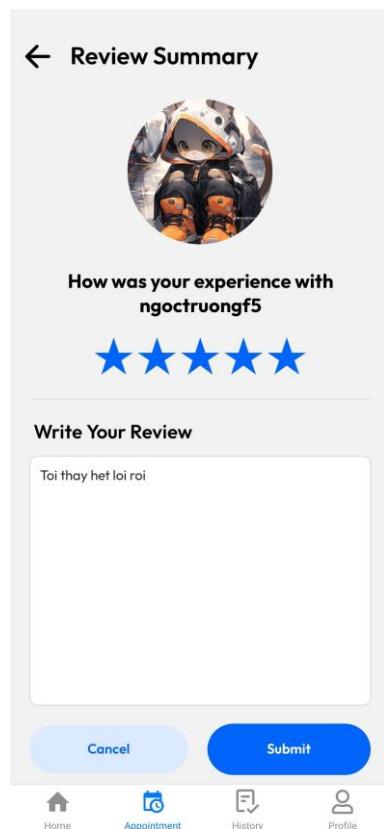
At the bottom of each screen, there is a navigation bar with icons for Home, Appointment, History, and Profile.

Hình 4. 12 Giao diện màn danh sách lịch hẹn theo status

- Bệnh nhân có thể review về bác sĩ đó khi ấn “Leave a Review”

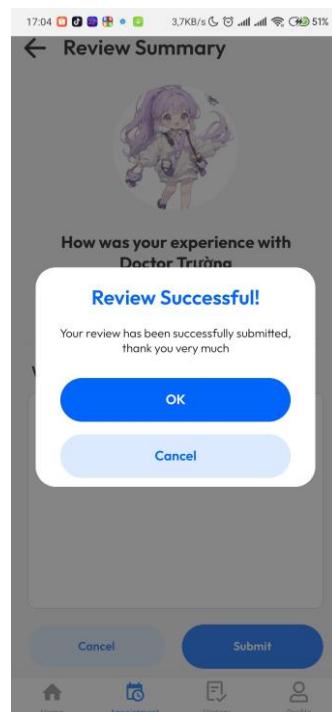
*Hình 4. 13 Giao diện lịch hẹn completed*

- Tại màn Review, bệnh nhân cho số điểm và comment sau đó ấn “Submit”



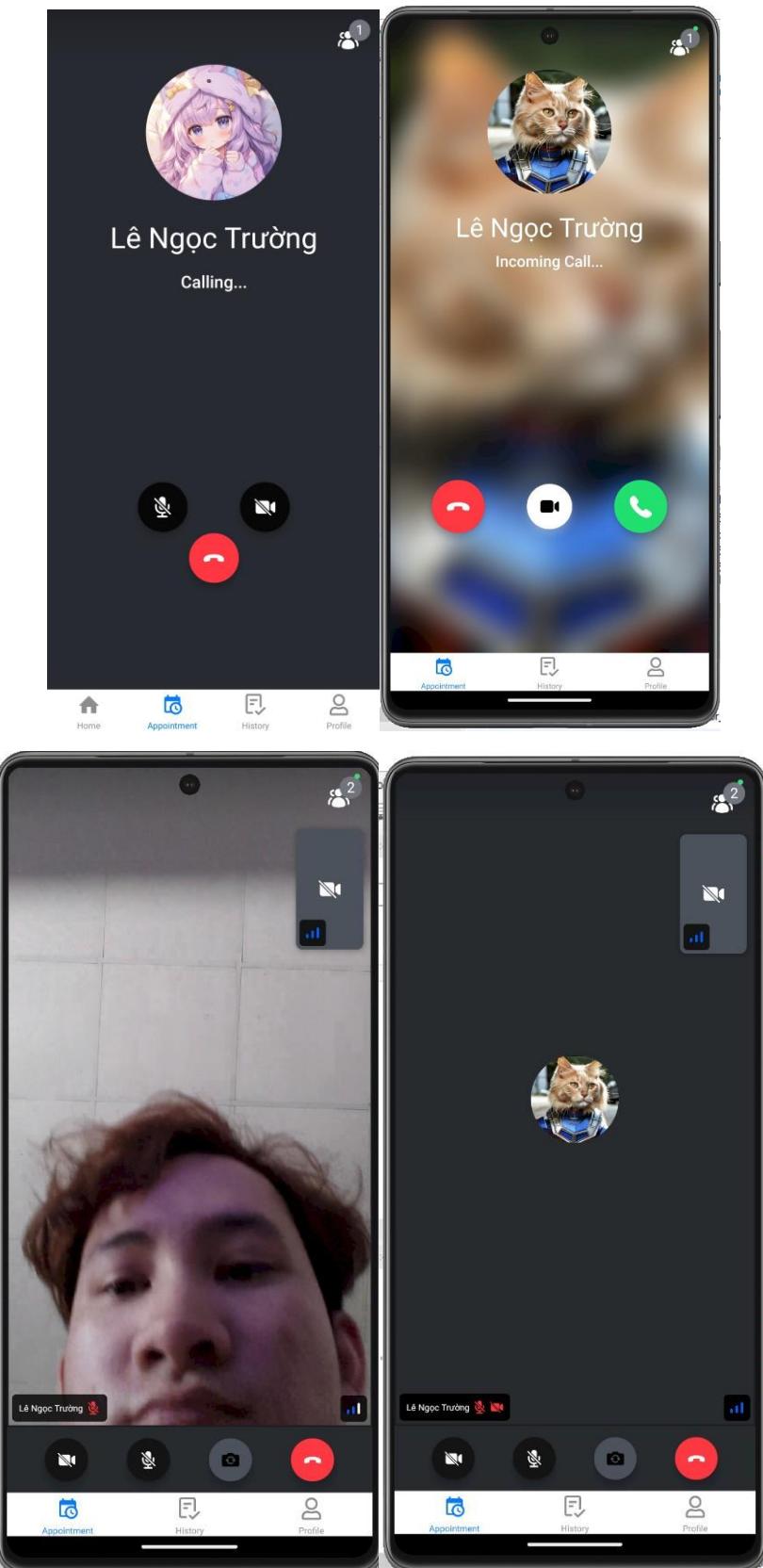
Hình 4. 14 Giao diện màn feedback

- Sau khi submit bệnh nhân sẽ nhận được thông báo review thành công. Bệnh nhân ấn “OK” để quay trở lại màn danh sách lịch hẹn.



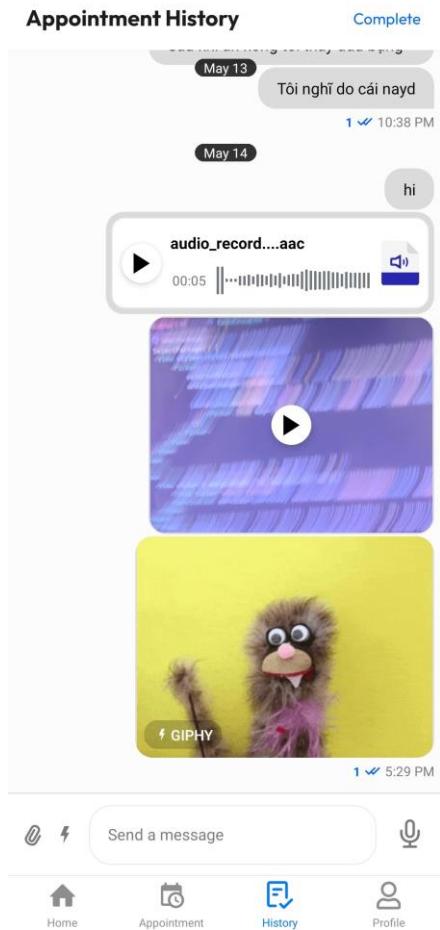
Hình 4. 15 Giao diện thông báo feedback thành công

- Bệnh nhân có thể video hoặc audio call với bác sĩ



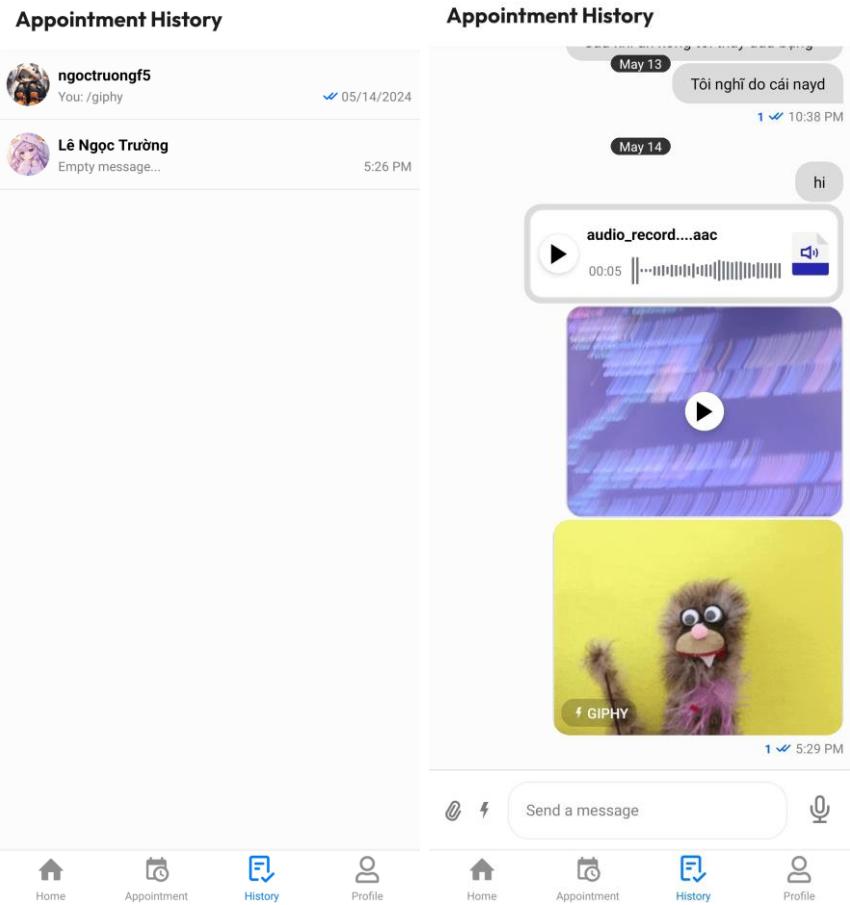
Hình 4. 16 Giao diện thông báo call và call

- Hoặc bệnh nhân có thể nhắn tin với bác sĩ khi ấn vào các lịch hẹn với các gói dịch vụ khác nhau



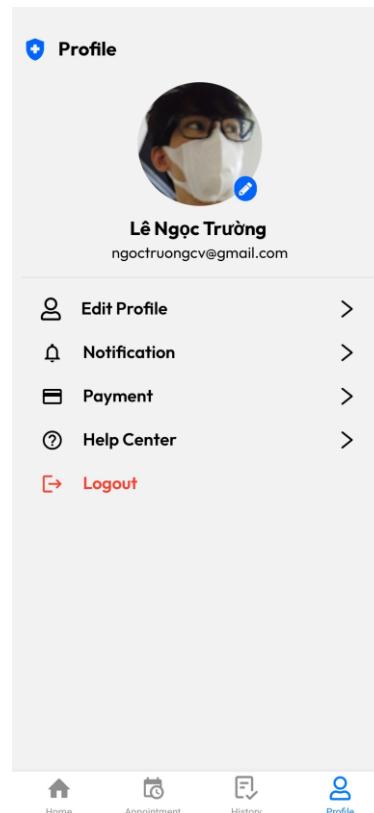
Hình 4. 17 Giao diện nhắn tin

- Bệnh nhân ấn tab “History” để xem lịch sử nhắn tin khám bệnh



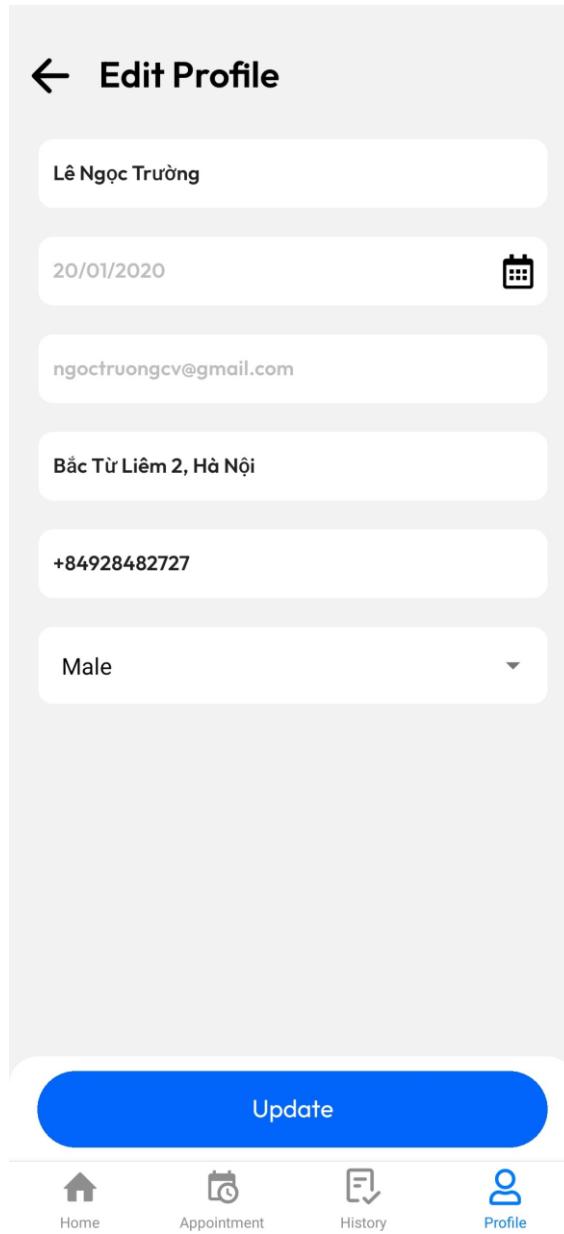
Hình 4. 18 Giao diện xem lại lịch sử trò chuyện

- Bệnh nhân ấn tab “Profile” để xem thông tin về tài khoản

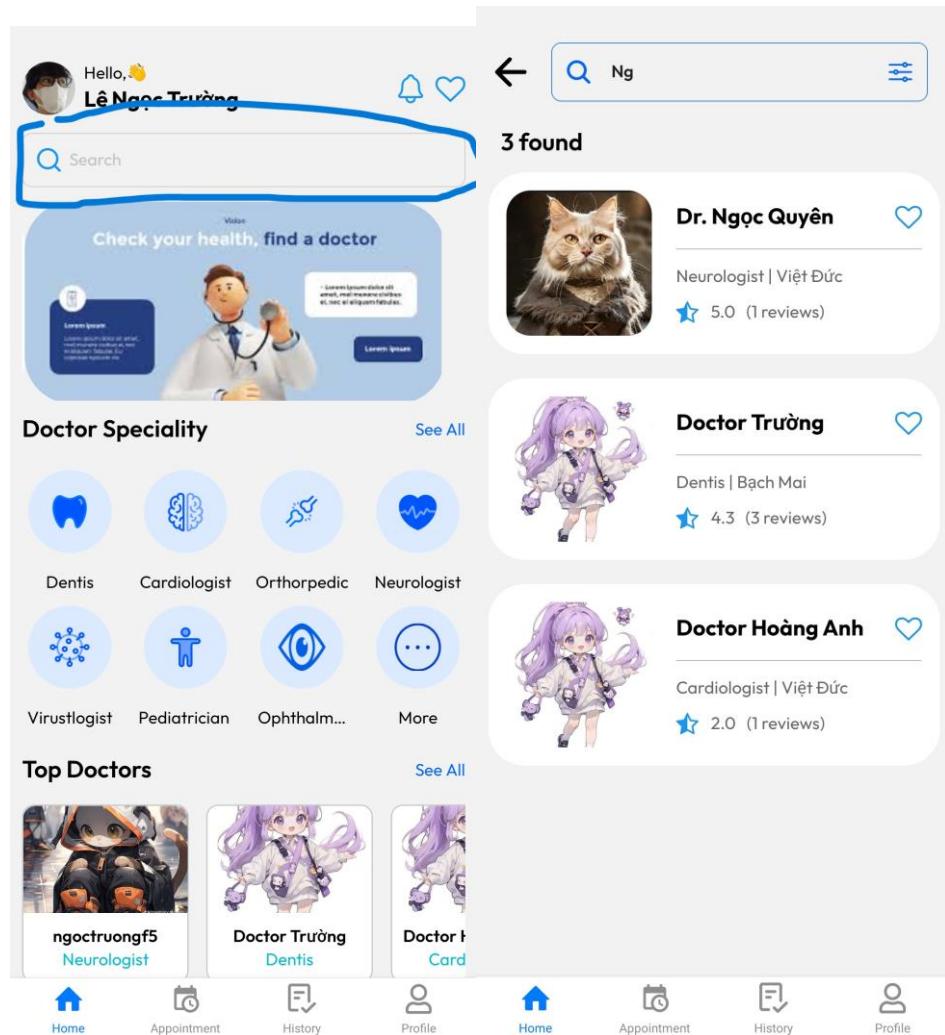


Hình 4. 19 Giao diện profile

- Án Edit Profile để cập nhật tài khoản

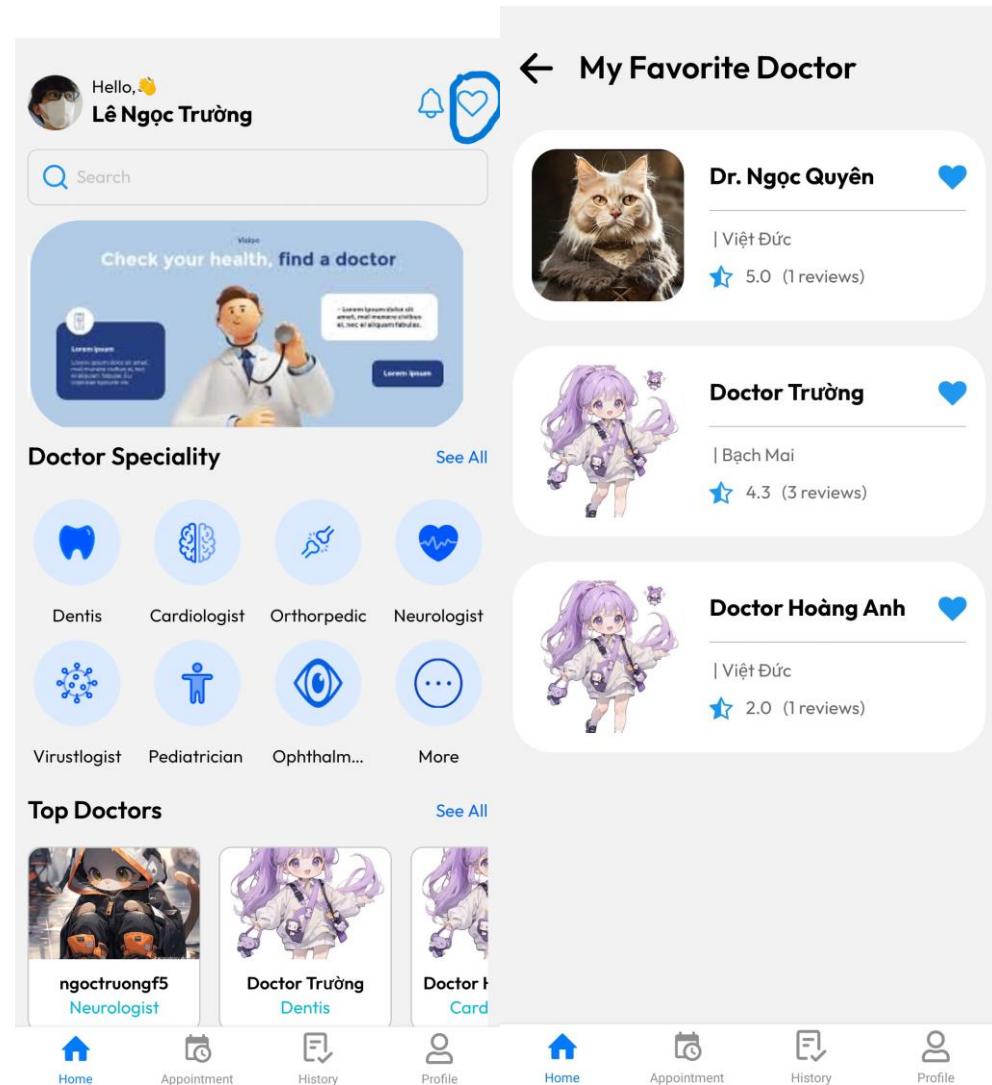
*Hình 4. 20 Giao diện chỉnh sửa profile*

- Bệnh nhân có thể ánh vào ô tìm kiếm ở màn Home để tìm kiếm bác sĩ theo tên



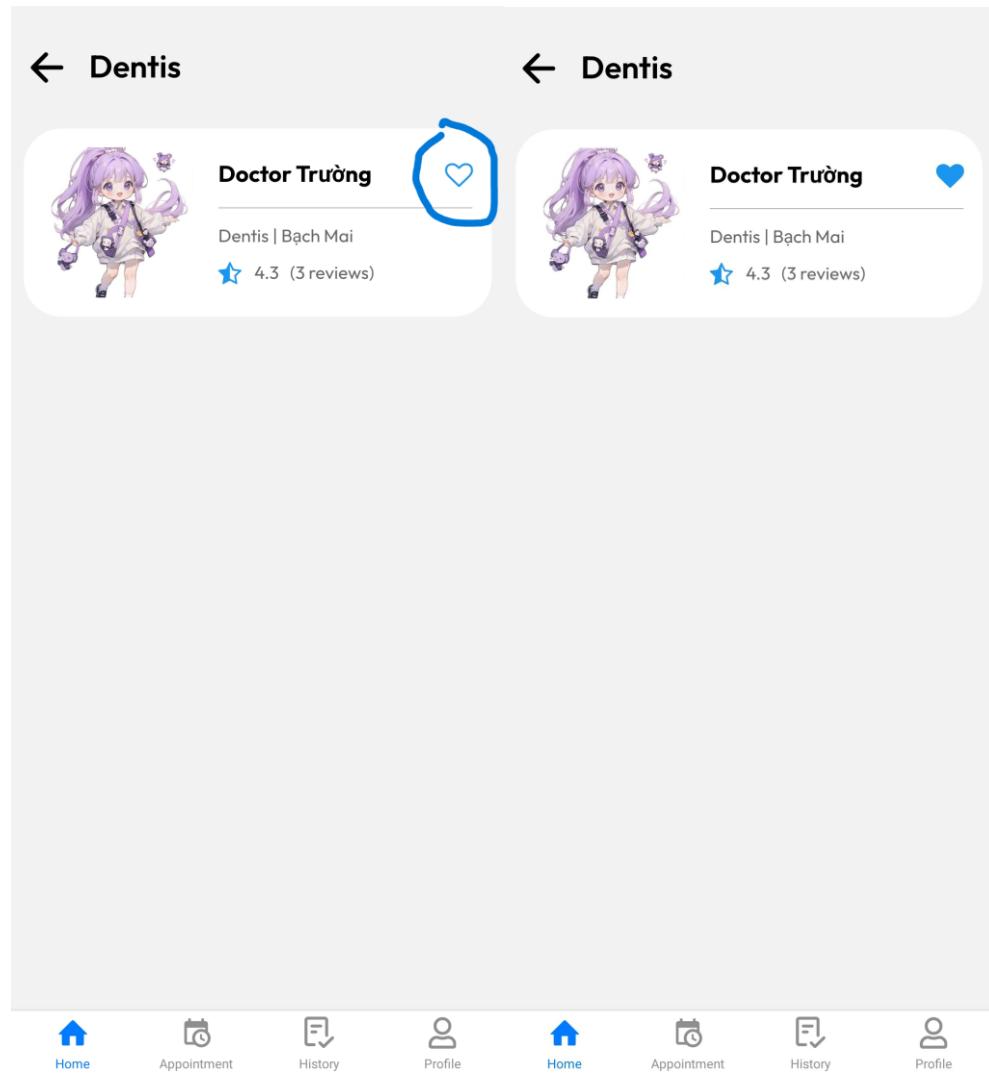
Hình 4. 21 Giao diện tìm kiếm bác sĩ theo tên

- Bệnh nhân có thể xem danh sách bác sĩ yêu thích khi ấn vào Icon trái tim ở màn Home



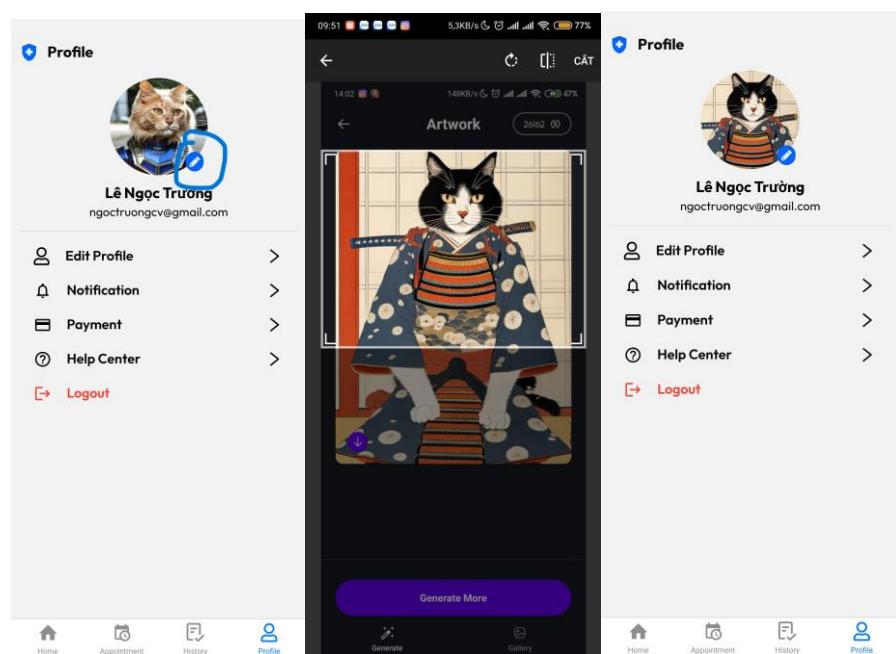
Hình 4. 22 Giao diện xem danh sách bác sĩ yêu thích

- Bệnh nhân có thể thêm bác sĩ vào danh sách yêu thích bằng cách ấn vào icon trái tim của mỗi góc phải card bác sĩ.



Hình 4. 23 Giao diện thêm bác sĩ vào danh sách yêu thích

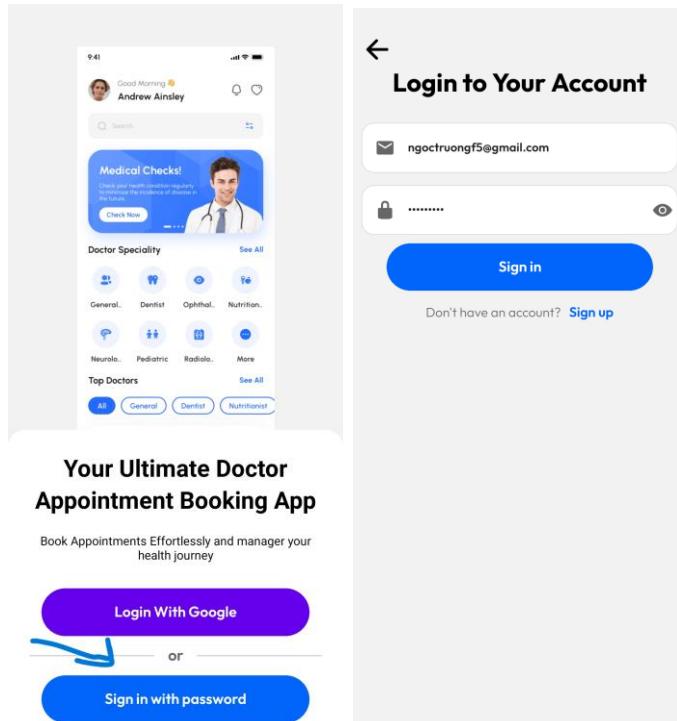
- Bệnh nhân có thể upload avatar mới tại màn Profile



Hình 4. 24 Giao diện upload avatar

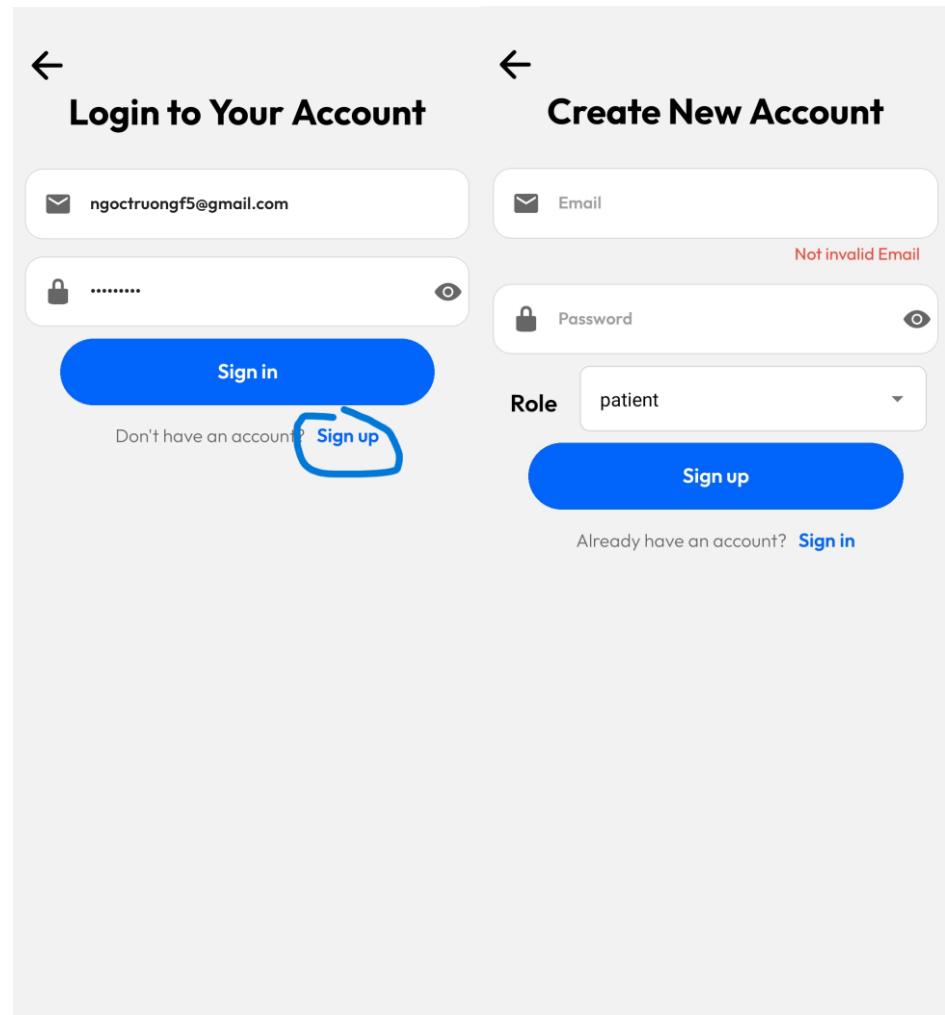
Giao diện ứng dụng của bác sĩ và hướng dẫn sử dụng

- Bác sĩ login tài khoản bằng email / password:

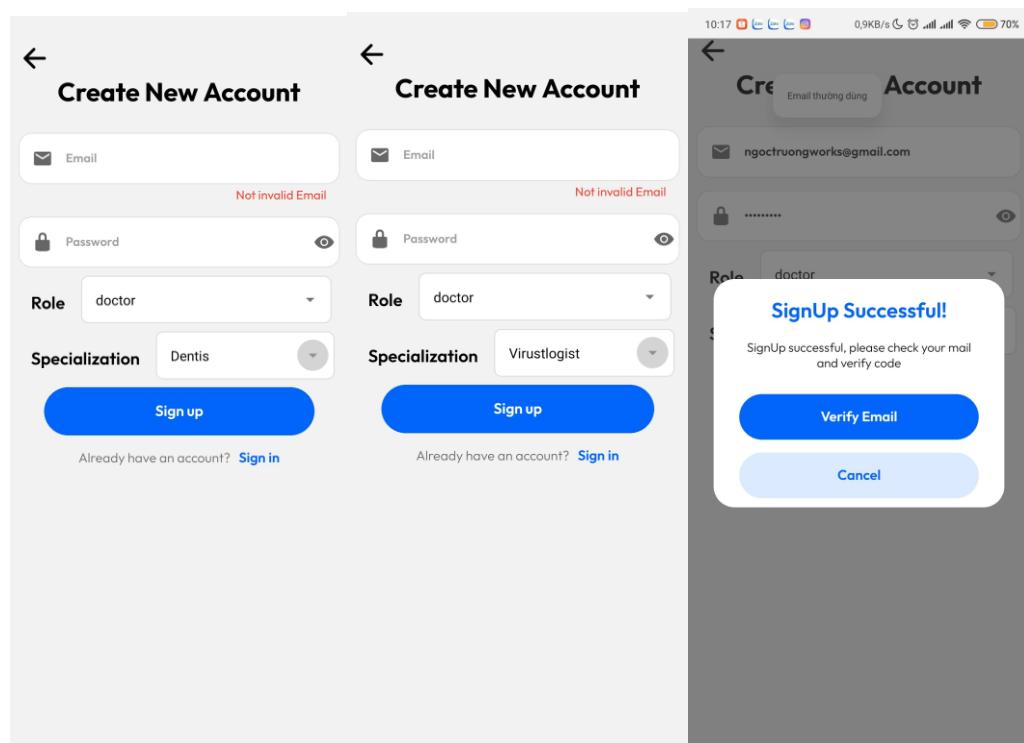


Hình 4. 25 Giao diện đăng nhập bằng email / password

- Người dùng có thể đăng ký khi ấn chữ “Sign up”, tuy nhiên ưu tiên đăng ký tài khoản này với role bác sĩ vì role bệnh nhân thì tài khoản google tiện hơn

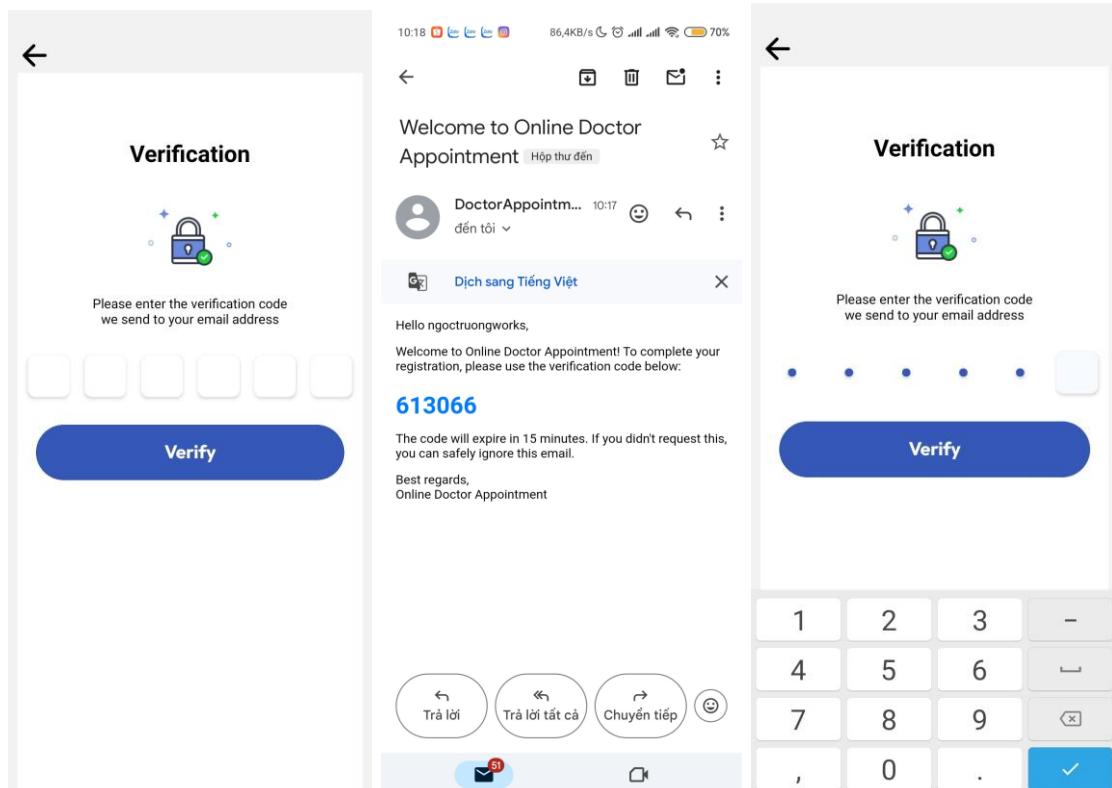


Hình 4. 26 Giao diện màn tạo tài khoản với role bệnh nhân



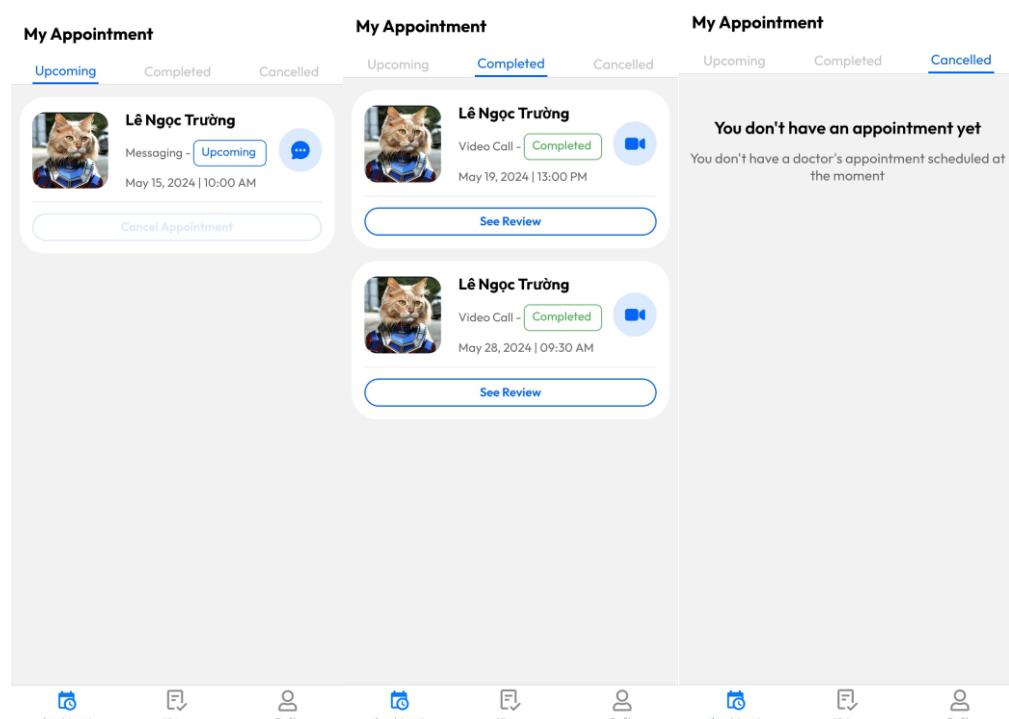
Hình 4. 27 Giao diện đăng ký tài khoản bác sĩ

- Click “Verify Email” và check email của bạn để lấy mã xác thực



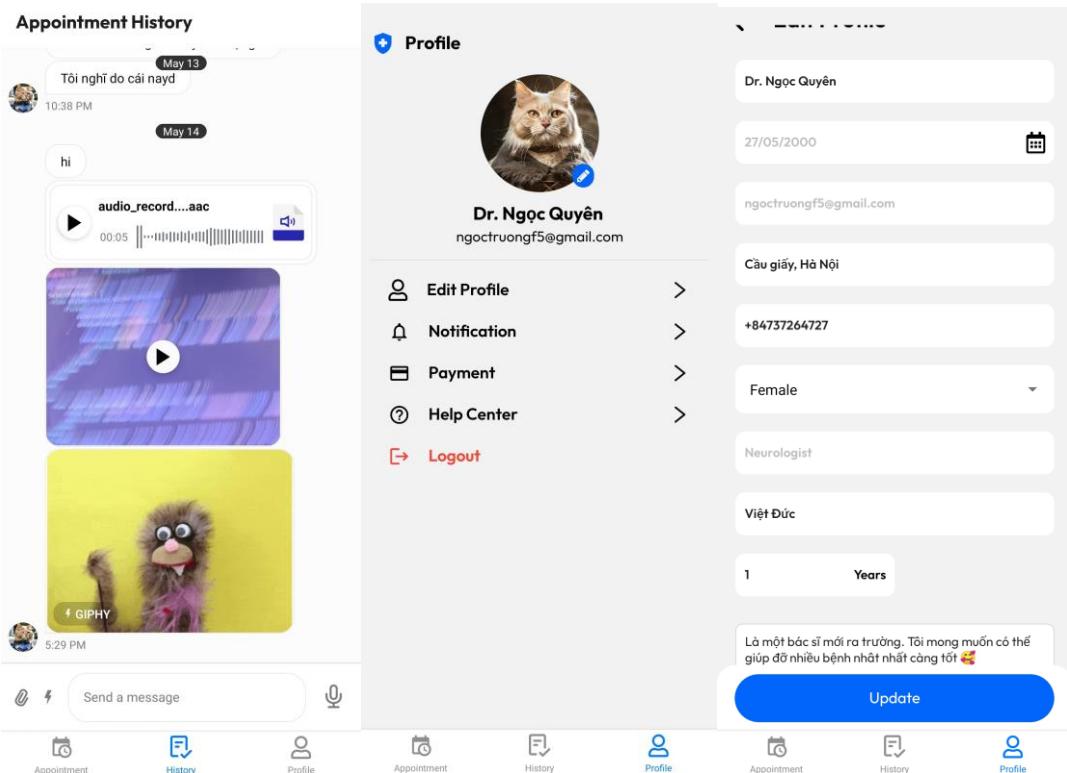
Hình 4. 28 Giao diện màn hình xác thực mã xác thực

- Giao diện chính của vai trò bác sĩ là Quản lý lịch hẹn



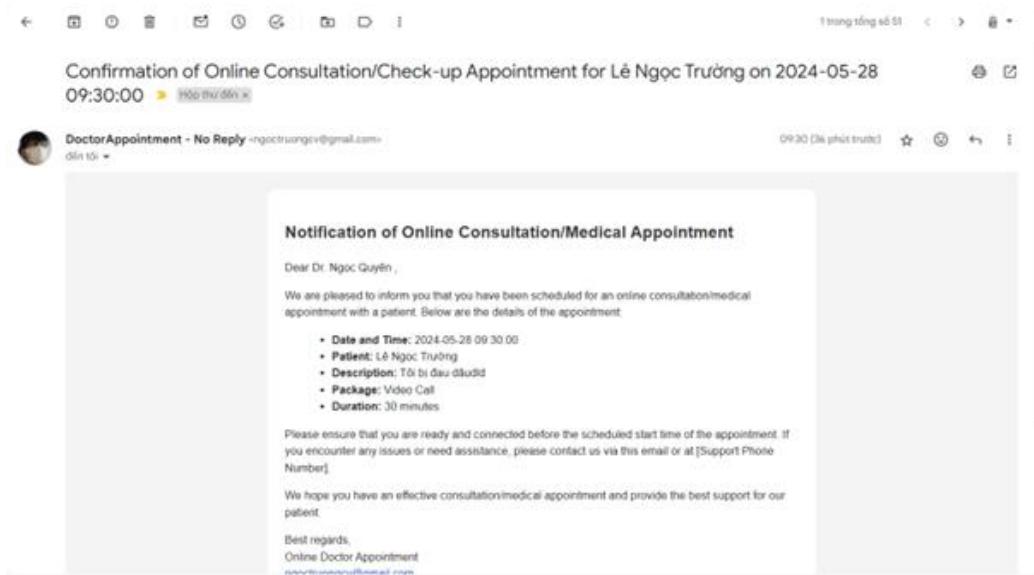
Hình 4. 29 Giao diện quản lý lịch hẹn của bác sĩ

- Có thể call và nhắn tin với bệnh nhân tương tự với giao diện bên phía bệnh nhân. Ngoài ra các chức năng upload avatar và cập nhật thông tin cá nhân cũng tương tự



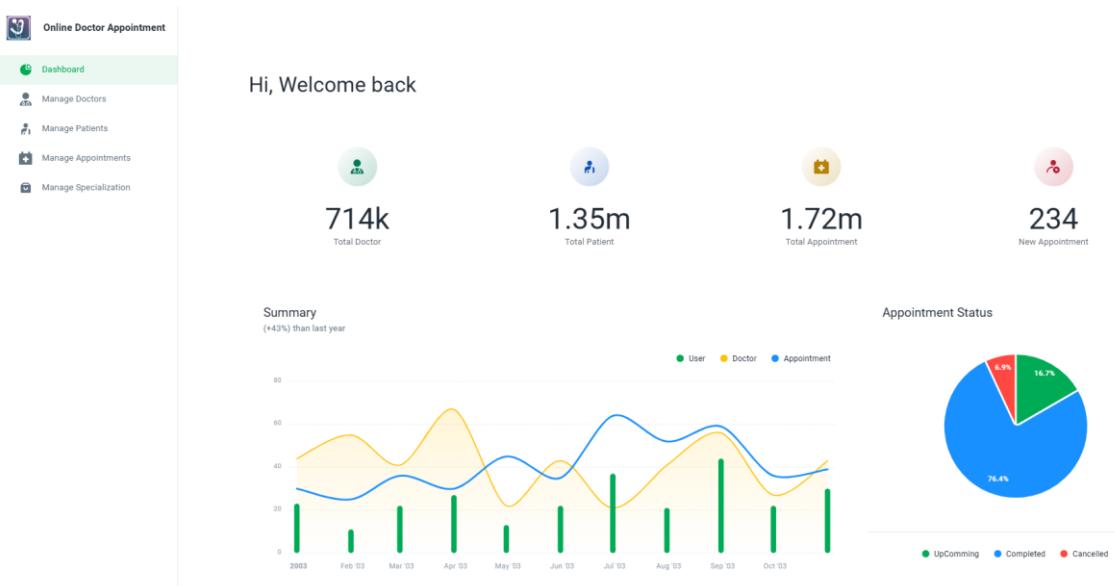
Hình 4. 30 Giao diện nhắn tin và quản lý thông tin cá nhân của bác sĩ

- Mail thông báo khi có lịch hẹn của bệnh nhân

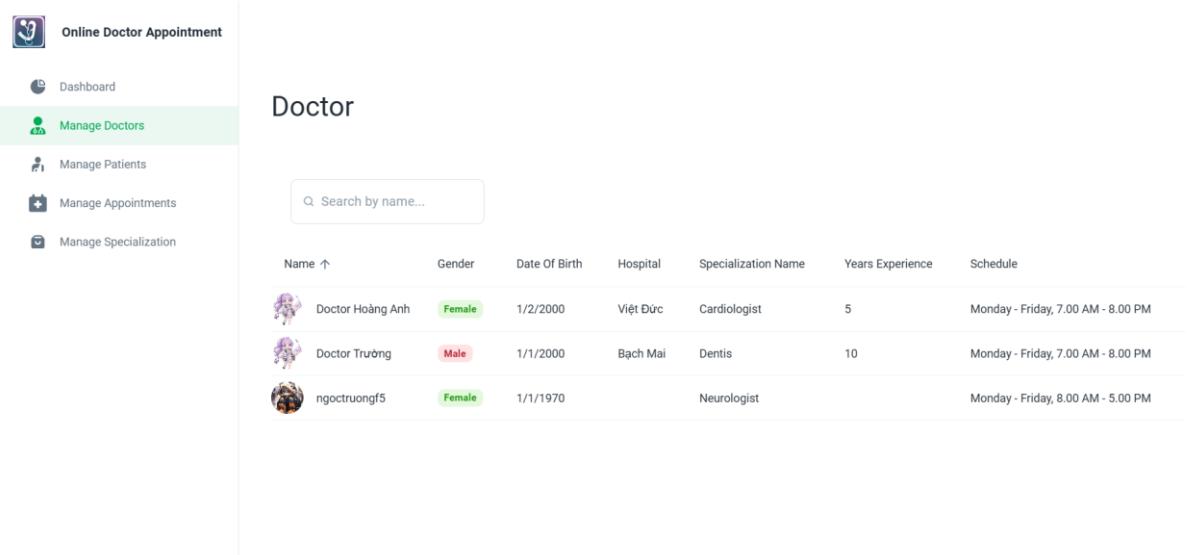


Hình 4. 31 Mail thông báo có lịch hẹn của bác sĩ

4.2. Giao diện Admin



Hình 4. 32 Giao diện dashboard



Hình 4. 33 Giao diện quản lý bác sĩ

Name ↑	Email	Gender	Date Of Birth	Address	Action
Le Ngoc Truong	ngoctruongf6@gmail.com	Female			Delete
Lê Ngọc Trường	ngoctruongcv@gmail.com	Male	2020-01-20	Bắc Từ Liêm 2, Hà Nội	Delete
abc	abc11@gmail.com	Female			Delete
ngoctruongf10	ngoctruongf10@gmail.com	Male	2002-01-07	Hoàng Mai, Hà Nội	Delete

Hình 4. 34 Giao diện quản lý bệnh nhân

Doctor	Patient	Package	Date ↑	Description	Duration	Status
Doctor Trưởng	Lê Ngọc Trường	Messaging	5/10/2024	tôi bị ốm, đau đầu, đau bụng, đau lưng, đau chân, đau cổ tay, đau ngực, đau chân. Nói chung chỗ nào cũng đau hết	0	Completed
Doctor Trưởng	Lê Ngọc Trường	Voice Call	5/13/2024	tôi bị ốm, đau đầu, đau bụng, đau lưng, đau chân, đau cổ tay, đau ngực, đau chân. Nói chung chỗ nào cũng đau hết	0	Upcoming
Doctor Trưởng	Lê Ngọc Trường	Messaging	5/14/2024	tôi bị ốm, đau đầu, đau bụng, đau lưng, đau chân, đau cổ tay, đau ngực, đau chân. Nói chung chỗ nào cũng đau hết	0	Cancelled
ngoctruongf5	Lê Ngọc Trường	Messaging	5/15/2024	Tôi bị ốm	0	Upcoming
ngoctruongf5	Lê Ngọc Trường	Video Call	5/19/2024	Toi thay chong mat, hoi met moi 1 chut, can tu van	0.5	Completed

Hình 4. 35 Giao diện quản lý lịch hẹn

Doctor	Patient	Package	Date ↑	Description	Duration	Status
Doctor Trưởng	Lê Ngọc Trường	Messaging	5/10/2024	tôi bị ốm, đau đầu, đau bụng, đau lưng, đau chân, đau cổ tay, đau ngực, đau chân. Nói chung chỗ nào cũng đau hết	0	Completed

Hình 4. 36 Giao diện tìm kiếm lịch hẹn theo ngày

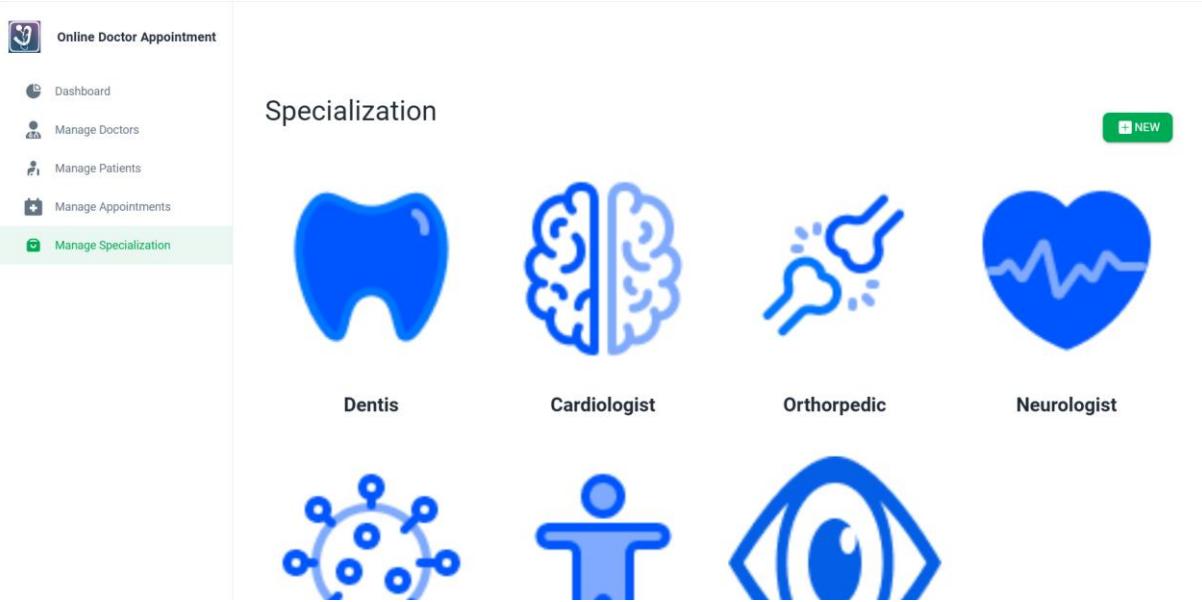
Appointment

Select Date

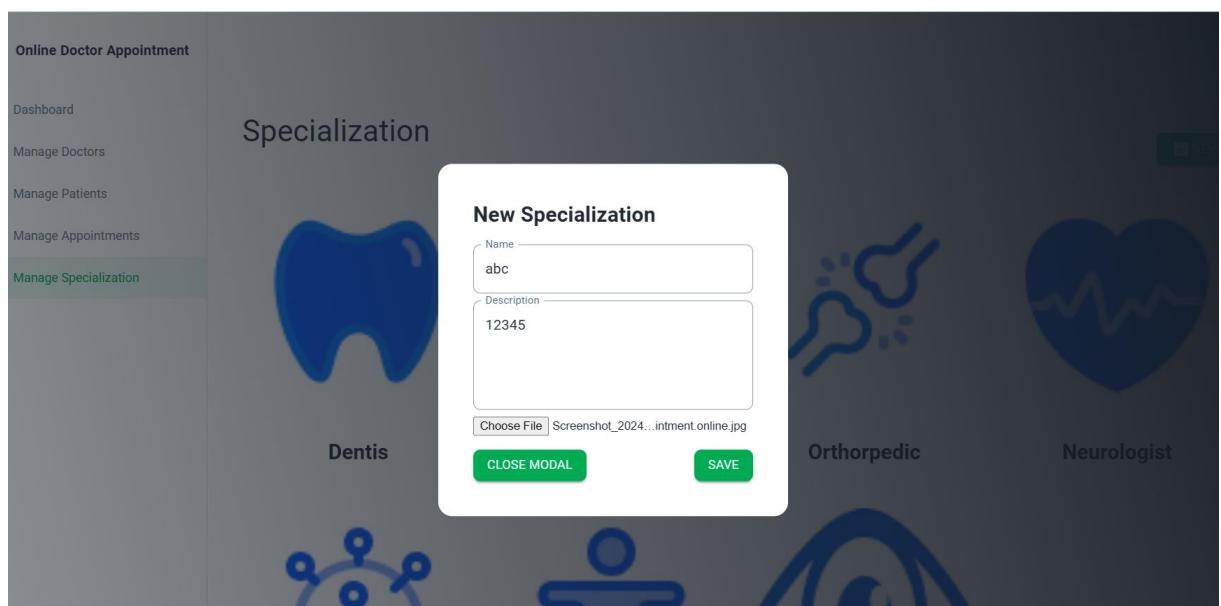
Filter by status

Doctor	Patient	Package	Date ↑	Description	Duration	Status
Doctor Trường	Lê Ngọc Trường	Voice Call	5/13/2024	tôi bị ốm, đau đầu, đau bụng, đau lưng, đau chân, đau cổ tay, đau ngực, đau chân. Nói chung chỗ nào cũng đau hết	0	Upcoming
ngotrueongf5	Lê Ngọc Trường	Messaging	5/15/2024	Tôi bị ốm	0	Upcoming

Hình 4. 37 Giao diện filter lịch theo status



Hình 4. 38 Giao diện quản lý chuyên ngành



Hình 4. 39 Giao diện thêm chuyên ngành

KẾT LUẬN

Đề tài “Xây dựng ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa” với mục đích mang đến một giải pháp tiện lợi và hiệu quả cho người dùng trong việc tiếp cận dịch vụ y tế. Ứng dụng được thiết kế với mong muốn giảm thiểu thời gian chờ đợi, tạo điều kiện thuận lợi cho người bệnh đặt lịch hẹn khám bệnh, trao đổi với bác sĩ từ xa, theo dõi lịch sử khám chữa bệnh, và tiếp cận thông tin y tế một cách nhanh chóng và dễ dàng.

Ứng dụng được xây dựng dựa trên nền tảng công nghệ hiện đại, kết hợp với giao diện thân thiện và dễ sử dụng, phù hợp với nhu cầu của đa dạng đối tượng người dùng. Trong quá trình thực hiện đề tài, em đã học hỏi và tích lũy thêm nhiều kiến thức, kinh nghiệm về lập trình, thiết kế ứng dụng, quản lý dữ liệu, cũng như kiến thức về ngành y tế, giúp em hiểu rõ hơn về nhu cầu và mong muốn của người dùng.

Tuy nhiên, do hạn chế về kinh nghiệm và thời gian, ứng dụng vẫn còn một số hạn chế về tính năng và khả năng mở rộng. Ví dụ, ứng dụng chưa tích hợp đầy đủ các chức năng hỗ trợ khám bệnh từ xa, chưa hỗ trợ đa ngôn ngữ, và chưa được tối ưu hóa cho các thiết bị di động khác nhau.

Em rất mong nhận được những lời nhận xét, góp ý quý báu từ thầy cô để có thể hoàn thiện ứng dụng hơn nữa, mang lại nhiều lợi ích thiết thực cho người dùng.

Qua quá trình thực hiện đề tài, em đã có thêm nhiều kiến thức và kỹ năng, góp phần nâng cao năng lực chuyên môn và khả năng ứng dụng công nghệ vào thực tiễn. Em tin rằng, với sự phát triển của công nghệ thông tin, ứng dụng đặt lịch và hỗ trợ khám bệnh từ xa sẽ ngày càng phát triển và trở thành một phần không thể thiếu trong hệ thống y tế hiện đại, góp phần nâng cao chất lượng dịch vụ y tế và mang lại cuộc sống khỏe mạnh cho cộng đồng.

- **Những kết quả đạt được:**

Về công nghệ:

- + Hiểu được quá trình thiết kế một hệ thống thực tế đi từ bước cơ bản: khảo sát dự án, phân tích hệ thống, thiết kế, thực hiện, kiểm thử, triển khai, bảo trì.
- + Xây dựng thành công ứng dụng và website đáp ứng nhu cầu đặt lịch và tư vấn trực tuyến.
- + Nắm được các kiến thức xây dựng giao diện website: HTML, CSS, JavaScript, ReactJS.
- + Nắm vững về cách xây dựng ứng dụng thông qua thư viện, framework: NodeJS, Firebase, GetStream, React Native trong lập trình.
- + Sử dụng thành thạo các công cụ hỗ trợ lập trình: Visual Studio Code, MySQL, Git và Github,

Về chương trình:

- + Cho phép bệnh nhân đăng nhập để thực hiện các chức năng như đặt lịch, nhắn tin, call với bác sĩ 1 cách trực tuyến nhanh chóng chỉ vài click, nhận xét đánh giá dịch vụ, tìm kiếm các bác sĩ theo chuyên ngành hoặc theo tên, xem lại lịch sử khám bệnh với bác sĩ bằng tin nhắn,...
- + Bác sĩ có thể xem review của bệnh nhân về cuộc hẹn đó, nhắn tin và call với bệnh nhân 1 cách nhanh chóng chỉ vài click. Xem lại lịch sử nhắn tin với người bệnh,...
- + Trang quản trị đầy đủ chức năng để quản lý một hệ thống: quản lý người bệnh, quản lý bác sĩ, quản lý lịch hẹn, quản lý chuyên ngành.

- Hướng phát triển

- + Tiếp tục hoàn thiện các tính năng mở rộng phần mềm nhằm cải tiến và nâng cấp chương trình.
- + Hoàn thiện chương trình, sửa chữa các lỗi, cải thiện tính bảo mật.
- + Xây dựng ứng dụng với quy mô lớn hơn. Thêm nhiều chức năng hơn
- + Đăng nhập bằng tài khoản Apple, Facebook.

TÀI LIỆU THAM KHẢO

- [1]. *Giáo trình nhập môn công nghệ phần mềm*, Phùng Đức Hòa, NXB Thông kê, 2019.
- [2]. *Giáo trình phân tích thiết kế hệ thống*, Nguyễn Thị Thanh Huyền, NXB Giáo dục Việt Nam, 2011.
- [3]. *Giáo trình thiết kế Web*, Nguyễn Trung Phú(ch.b), NXB Thông kê, 2019.
- [4]. *Head First JavaScript Programming*, Elisabeth Robson, 2014.
- [5]. *Giáo trình Hệ quản trị cơ sở dữ liệu*, Đỗ Ngọc Sơn, Phan Văn Viên, Nguyễn Phương Nga - Trường Đại học Công nghiệp Hà Nội – NXB Khoa học và Kỹ thuật.
- [6]. *Giới thiệu về ReactJS*, Đỗ Khánh Toàn, Viblo, 2017.
- [7]. *What Is MySQL?*, Oracle, 2024