

**ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN - ĐIỆN TỬ**



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**HỆ THỐNG QUẢN LÝ DỮ LIỆU ĐIỆN TIM
VÀ TƯƠNG TÁC GIỮA BỆNH NHÂN - BÁC SĨ**

Sinh viên thực hiện: **Cồ Huy Dũng**
Lớp Điện tử 02 - K66
Nguyễn Đức Dương
Lớp Điện tử 08 - K66
Giảng viên hướng dẫn: **TS. Hàn Huy Dũng**

Hà Nội 02/2025

LỜI NÓI ĐẦU

Trong thời đại công nghệ phát triển vượt bậc, Internet vạn vật (IoT) ngày càng khẳng định vai trò quan trọng, đặc biệt trong lĩnh vực y tế. Việc áp dụng IoT vào quản lý thiết bị y tế không chỉ giúp tối ưu hóa quy trình chăm sóc sức khỏe mà còn mở ra những cơ hội đổi mới, nâng cao chất lượng dịch vụ y tế. Tại Việt Nam, ứng dụng các công nghệ tiên tiến vào lĩnh vực y tế đóng vai trò quan trọng trong việc cải thiện sức khỏe cộng đồng, từ đó góp phần xây dựng nguồn nhân lực khỏe mạnh, đáp ứng yêu cầu phát triển đất nước.

Đề án này của chúng em xây dựng một hệ thống trang web quản lý các thiết bị y tế và dữ liệu điện tim, với mục tiêu tăng cường kết nối và hỗ trợ tương tác giữa bệnh nhân và bác sĩ. Hệ thống cho phép bệnh nhân dễ dàng theo dõi tình trạng sức khỏe tại nhà, đồng thời nhận được sự hỗ trợ chuyên môn nhanh chóng từ đội ngũ y tế. Với giao diện thân thiện và dễ sử dụng, hệ thống tích hợp các chức năng hữu ích như đặt lịch hẹn, theo dõi dữ liệu điện tim, và trao đổi thông tin trực tiếp giữa bệnh nhân và bác sĩ thông qua tính năng nhắn tin.

Trong quá trình thực hiện, chúng em đã nhận được sự hỗ trợ và hướng dẫn tận tình từ các thầy cô cùng sự giúp đỡ của các anh/chị/bạn tại các phòng thí nghiệm thuộc khoa Điện - Điện tử. Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến TS. Hàn Huy Dũng, người đã luôn đồng hành và đưa ra những góp ý quý báu trong việc hoàn thiện đề án. Đồng thời, sự phối hợp cùng nhóm firmware của SPARC Lab đã giúp chúng em tích lũy được nhiều kiến thức và kinh nghiệm thực tế.

Dẫu đã nỗ lực hoàn thiện, đề án của chúng em chắc chắn vẫn còn một số hạn chế. Chúng em rất mong nhận được những ý kiến đóng góp từ thầy cô và bạn đọc để tiếp tục phát triển và nâng cao chất lượng đề tài trong tương lai.

Chúng em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Chúng em gồm Cồ Huy Dũng, mã số sinh viên 20213834, thuộc lớp Điện tử 02, khóa K66, và Nguyễn Đức Dương, mã số sinh viên 20210259, thuộc lớp Điện tử 08, khóa K66.

Chúng em xin khẳng định rằng toàn bộ nội dung được trình bày trong đồ án "Hệ thống quản lý dữ liệu điện tim và tương tác giữa bệnh nhân - bác sĩ" là kết quả của quá trình tìm hiểu, nghiên cứu và thực hiện nghiêm túc của chính chúng em. Các thông tin, dữ liệu trong đồ án được đảm bảo tính trung thực và chính xác. Mọi tài liệu tham khảo đều được trích dẫn đầy đủ, tuân thủ các quy định về sở hữu trí tuệ.

Chúng em xin cam kết chịu trách nhiệm hoàn toàn về nội dung đã trình bày trong đồ án này.

Hà Nội, ngày 10 tháng 02 năm 2024

Người cam đoan

NGUYỄN ĐỨC DƯƠNG, CỒ HUY DŨNG

PHÂN CÔNG CÔNG VIỆC

STT	Nội dung	Thành viên
1	Nghiên cứu đề tài và đề xuất hệ thống	Dũng, Dương
2	Phân tích chức năng và yêu cầu hệ thống	Dũng, Dương
3	Thiết kế cơ sở dữ liệu	Dũng, Dương
4	Xây dựng các API cần thiết	Dũng, Dương
5	Phát triển hệ thống phía server	Dũng
6	Xây dựng giao diện website phù hợp với từng vai trò người dùng	Dương
7	Thực hiện kiểm thử và sửa lỗi	Dũng, Dương
8	Viết quyền đồ án	Dũng, Dương

MỤC LỤC

DANH MỤC KÝ HÌNH VẼ	10
DANH MỤC BẢNG BIỂU	15
TÓM TẮT ĐỒ ÁN	16
PHẦN MỞ ĐẦU	17
CHƯƠNG 1. THU THẬP YÊU CẦU	1
1.1 Phân tích yêu cầu hệ thống	1
1.1.1 Yêu cầu chức năng của hệ thống	1
1.1.2 Yêu cầu phi chức năng của hệ thống	2
1.1.3 Yêu cầu về người dùng hệ thống	3
1.2 Sơ đồ các trường hợp sử dụng	4
1.2.1 Sơ đồ tổng quát các trường hợp sử dụng	4
1.2.2 Sơ đồ trường hợp sử dụng chức năng tạo tài khoản người dùng .	4
1.2.3 Sơ đồ trường hợp sử dụng chức năng đăng nhập và đăng xuất khỏi hệ thống	5
1.2.4 Sơ đồ trường hợp sử dụng chức năng quản lý tài khoản người dùng	6
1.2.5 Sơ đồ trường hợp sử dụng chức năng quản lý người dùng hệ thống	8
1.2.6 Sơ đồ trường hợp sử dụng chức năng quản lý thiết bị y tế	11
1.2.7 Sơ đồ trường hợp sử dụng chức năng quản lý dữ liệu phiên đo .	14
1.2.8 Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ lịch khám	17
1.2.9 Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ nhắn tin .	21
CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG	23
2.1 Thẻ CRC (Class - Responsibility - Collaboration Card)	23
2.1.1 Thẻ CRC lớp Tài khoản người dùng	23
2.1.2 Thẻ CRC lớp Token	24

2.1.3	Thẻ CRC lớp Vai trò người dùng	24
2.1.4	Thẻ CRC lớp Trạng thái người dùng	25
2.1.5	Thẻ CRC lớp Người dùng	26
2.1.6	Thẻ CRC lớp Loại thiết bị	27
2.1.7	Thẻ CRC lớp Trạng thái thiết bị	27
2.1.8	Thẻ CRC lớp Thiết bị	28
2.1.9	Thẻ CRC lớp Thông số kỹ thuật	29
2.1.10	Thẻ CRC lớp Dữ liệu phiên đo	29
2.1.11	Thẻ CRC lớp Trạng thái lịch khám	30
2.1.12	Thẻ CRC lớp Kết quả lịch khám	31
2.1.13	Thẻ CRC lớp Lịch khám	32
2.1.14	Thẻ CRC lớp Thông báo liên quan đến lịch khám	33
2.1.15	Thẻ CRC lớp Chẩn đoán cho bệnh nhân	34
2.1.16	Thẻ CRC lớp Tin nhắn	34
2.1.17	Thẻ CRC lớp Nhóm trò chuyện	35
2.2	Sơ đồ lớp	36
2.3	Sơ đồ tuần tự	39
2.3.1	Sơ đồ tuần tự chức năng tạo tài khoản người dùng	39
2.3.2	Sơ đồ tuần tự chức năng đăng nhập và đăng xuất khỏi hệ thống	40
2.3.3	Sơ đồ tuần tự chức năng quản lý tài khoản người dùng	41
2.3.4	Sơ đồ tuần tự chức năng quản lý người dùng hệ thống	42
2.3.5	Sơ đồ tuần tự chức năng quản lý thiết bị y tế	46
2.3.6	Sơ đồ tuần tự chức năng quản lý dữ liệu phiên đo	49
2.3.7	Sơ đồ tuần tự chức năng quản lý dịch vụ lịch khám	55
2.3.8	Sơ đồ tuần tự chức năng quản lý dịch vụ nhắn tin	64
2.4	Xử lý và phân tích dữ liệu	67
2.5	Kết luận	69
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG		70
3.1	Sơ đồ tổng quan kiến trúc của hệ thống	70

3.2	Sơ đồ khối phần mềm	73
3.2.1	Website dành cho bệnh nhân	73
3.2.2	Website dành cho bác sĩ	74
3.2.3	Website cho quản trị viên	75
3.3	Thiết kế cơ sở dữ liệu	76
3.3.1	Chuyển đổi từ mô hình thực thể liên kết sang mô hình quan hệ .	76
3.3.2	Chuẩn hoá 3NF	77
3.3.3	Sơ đồ ERD	83
3.4	Thiết kế giao diện	83
3.5	Thiết kế các chức năng cho website và server	83
3.5.1	Thiết kế các API cần thiết	83
3.5.2	Sơ đồ tuần tự API	86
3.6	Kết luận chương	126
CHƯƠNG 4. TRIỂN KHAI VÀ KIỂM THỬ		127
4.1	Công nghệ sử dụng	127
4.1.1	Thiết kế giao diện website	127
4.1.2	Server	127
4.2	Triển khai ứng dụng	132
4.2.1	Quy trình CI/CD	132
4.2.2	Kiến trúc Microservices	133
4.3	Kiểm thử	134
4.3.1	Kiểm thử hoạt động của các API	134
4.3.2	Kiểm thử ứng dụng web	166
4.4	Kết luận chương	169
PHỤ LỤC		170

DANH MỤC HÌNH VẼ

Hình 1.1	Sơ đồ tổng quát các trường hợp sử dụng của hệ thống	4
Hình 1.2	Sơ đồ trường hợp sử dụng chức năng tạo tài khoản người dùng . . .	4
Hình 1.3	Sơ đồ trường hợp sử dụng chức năng đăng nhập và đăng xuất khỏi hệ thống	5
Hình 1.4	Sơ đồ trường hợp sử dụng chức năng quản lý tài khoản người dùng	6
Hình 1.5	Sơ đồ trường hợp sử dụng chức năng quản lý người dùng	8
Hình 1.6	Sơ đồ trường hợp sử dụng chức năng quản lý thiết bị y tế	11
Hình 1.7	Sơ đồ trường hợp sử dụng chức năng quản lý dữ liệu phiên đo . . .	14
Hình 1.8	Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ lịch khám . .	17
Hình 1.9	Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ nhắn tin . . .	21
Hình 2.1	Sơ đồ ERD	37
Hình 2.2	Sơ đồ tuần tự chức năng tạo tài khoản người dùng	39
Hình 2.3	Sơ đồ tuần tự chức năng đăng nhập vào hệ thống	40
Hình 2.4	Sơ đồ tuần tự chức năng đăng xuất khỏi hệ thống	41
Hình 2.5	Sơ đồ tuần tự chức năng quản lý tài khoản người dùng	41
Hình 2.6	Sơ đồ tuần tự chức năng tra cứu danh sách người dùng	42
Hình 2.7	Sơ đồ tuần tự chức năng tra cứu danh sách bác sĩ	43
Hình 2.8	Sơ đồ tuần tự chức năng tra cứu danh sách bệnh nhân đang theo dõi	43
Hình 2.9	Sơ đồ tuần tự chức năng sửa thông tin người dùng	44
Hình 2.10	Sơ đồ tuần tự chức năng xóa người dùng khỏi hệ thống	45
Hình 2.11	Sơ đồ tuần tự chức năng tra cứu danh sách thiết bị y tế	46
Hình 2.12	Sơ đồ tuần tự chức năng thêm mới thiết bị y tế	47
Hình 2.13	Sơ đồ tuần tự chức năng phân công thiết bị y tế cho người dùng . .	47
Hình 2.14	Sơ đồ tuần tự chức năng chỉnh sửa thông tin thiết bị y tế	48
Hình 2.15	Sơ đồ tuần tự chức năng xóa thiết bị y tế	49
Hình 2.16	Sơ đồ tuần tự chức năng tra cứu danh sách toàn bộ phiên đo	49
Hình 2.17	Sơ đồ tuần tự chức năng tra cứu danh sách các phiên đo cá nhân . .	50

Hình 2.18	Sơ đồ tuần tự chức năng tra cứu danh sách các phiên đo của bệnh nhân đang theo dõi	51
Hình 2.19	Sơ đồ tuần tự chức năng tra cứu thông tin chi tiết một phiên đo . .	52
Hình 2.20	Sơ đồ tuần tự chức năng xem đồ thị các tín hiệu phiên đo	52
Hình 2.21	Sơ đồ tuần tự chức năng thêm phiên đo mới	53
Hình 2.22	Sơ đồ tuần tự chức năng xóa một hoặc nhiều phiên đo	54
Hình 2.23	Sơ đồ tuần tự chức năng tìm kiếm lịch khám theo bác sĩ	55
Hình 2.24	Sơ đồ tuần tự chức năng tìm kiếm lịch khám theo thời gian rảnh . .	56
Hình 2.25	Sơ đồ tuần tự chức năng đặt lịch khám bởi bệnh nhân	57
Hình 2.26	Sơ đồ tuần tự chức năng đặt lịch tái khám bởi bác sĩ	58
Hình 2.27	Sơ đồ tuần tự chức năng tra cứu danh sách lịch khám trong hệ thống	58
Hình 2.28	Sơ đồ tuần tự chức năng tra cứu lịch khám cá nhân	59
Hình 2.29	Sơ đồ tuần tự chức năng tra cứu lịch khám của bệnh nhân đang được theo dõi	60
Hình 2.30	Sơ đồ tuần tự chức năng chấp nhận lịch khám	61
Hình 2.31	Sơ đồ tuần tự chức năng từ chối lịch khám	61
Hình 2.32	Sơ đồ tuần tự chức năng điền chẩn đoán	62
Hình 2.33	Sơ đồ tuần tự chức năng tra cứu thông tin chẩn đoán của lịch khám	62
Hình 2.34	Sơ đồ tuần tự chức năng chỉnh sửa thông tin chẩn đoán của lịch khám	63
Hình 2.35	Sơ đồ tuần tự chức năng xem lịch sử trò chuyện của các cuộc hội thoại	64
Hình 2.36	Sơ đồ tuần tự chức năng gửi tin nhắn tới các cuộc hội thoại	65
Hình 2.37	Sơ đồ tuần tự chức năng tạo nhóm trò chuyện	66
Hình 2.38	Sơ đồ tuần tự chức năng hiển thị tất cả các nhóm trò chuyện	66
Hình 2.39	Mô hình thực thể liên kết	69
Hình 3.1	Tổng quan kiến trúc hệ thống	70
Hình 3.2	Sơ đồ khối Website dành cho bệnh nhân	73
Hình 3.3	Sơ đồ khối Website dành cho bác sĩ	74
Hình 3.4	Sơ đồ khối Website dành cho quản trị viên	75
Hình 3.5	Sơ đồ ERD	83
Hình 3.6	Sơ đồ tuần tự API tạo mới tài khoản người dùng	87

Hình 3.7	Sơ đồ tuần tự API đăng nhập vào hệ thống	88
Hình 3.8	Sơ đồ tuần tự API đăng xuất khỏi hệ thống	89
Hình 3.9	Sơ đồ tuần tự API tra cứu danh sách tất cả người dùng trong hệ thống	90
Hình 3.10	Sơ đồ tuần tự API tra cứu danh sách toàn bộ bác sĩ trong hệ thống .	91
Hình 3.11	Sơ đồ tuần tự API tra cứu dữ liệu người dùng cụ thể dựa trên id . .	92
Hình 3.12	Sơ đồ tuần tự API tra cứu danh sách bệnh nhân đang được theo dõi	93
Hình 3.13	Sơ đồ tuần tự API chỉnh sửa thông tin người dùng	94
Hình 3.14	Sơ đồ tuần tự API xóa người dùng cụ thể dựa trên id	95
Hình 3.15	Sơ đồ tuần tự API tra cứu danh sách thiết bị y tế	96
Hình 3.16	Sơ đồ tuần tự API thêm mới thiết bị y tế	97
Hình 3.17	Sơ đồ tuần tự API đăng ký mượn thiết bị	98
Hình 3.18	Sơ đồ tuần tự API Cập nhật thiết bị và thông số kỹ thuật (nếu cần) .	99
Hình 3.19	Sơ đồ tuần tự API xóa thiết bị y tế	100
Hình 3.20	Sơ đồ tuần tự API tra cứu tất cả lịch hẹn	101
Hình 3.21	Sơ đồ tuần tự API bác sĩ tạo lịch tái khám cho bệnh nhân	102
Hình 3.22	Sơ đồ tuần tự API bệnh nhân chủ động đặt lịch hẹn	103
Hình 3.23	Sơ đồ tuần tự API tra cứu lịch hẹn của bác sĩ cụ thể	104
Hình 3.24	Sơ đồ tuần tự API tra cứu lịch hẹn của bệnh nhân cụ thể	105
Hình 3.25	Sơ đồ tuần tự API tra cứu lịch rảnh của bác sĩ cụ thể	106
Hình 3.26	Sơ đồ tuần tự API tra cứu bác sĩ phù hợp với thời gian đã chọn . . .	107
Hình 3.27	Sơ đồ tuần tự API bác sĩ xác nhận lịch hẹn	108
Hình 3.28	Sơ đồ tuần tự API bác sĩ từ chối lịch hẹn	109
Hình 3.29	Sơ đồ tuần tự API tra cứu danh sách các dữ liệu phiên đo	109
Hình 3.30	Sơ đồ tuần tự API tra cứu dữ liệu phiên đo theo id	110
Hình 3.31	Sơ đồ tuần tự API tra cứu dữ liệu phiên đo của bệnh nhân	111
Hình 3.32	Sơ đồ tuần tự API tra cứu dữ liệu phiên đo do bác sĩ phụ trách . . .	112
Hình 3.33	Sơ đồ tuần tự API tạo mới dữ liệu phiên đo	113
Hình 3.34	Sơ đồ tuần tự API cập nhật dữ liệu phiên đo	114
Hình 3.35	Sơ đồ tuần tự API xóa dữ liệu phiên đo	115

Hình 3.36	Sơ đồ tuần tự API tra cứu đồ thị các tín hiệu đã đo	116
Hình 3.37	Sơ đồ tuần tự API lấy chẩn đoán theo lịch hẹn	117
Hình 3.38	Sơ đồ tuần tự API tạo mới chẩn đoán	118
Hình 3.39	Sơ đồ tuần tự API cập nhật thông tin chẩn đoán	119
Hình 3.40	Sơ đồ tuần tự API tạo thông báo mới	120
Hình 3.41	Sơ đồ tuần tự API lấy danh sách thông báo của người dùng	121
Hình 3.42	Sơ đồ tuần tự API cập nhật trạng thái đã xem thông báo	122
Hình 3.43	Sơ đồ tuần tự API xóa thông báo	122
Hình 3.44	Sơ đồ tuần tự API lấy danh sách tin nhắn của người dùng	123
Hình 3.45	Sơ đồ tuần tự API gửi tin nhắn trong nhóm trò chuyện	124
Hình 3.46	Sơ đồ tuần tự API tạo nhóm trò chuyện mới	125
Hình 3.47	Sơ đồ tuần tự API lấy danh sách nhóm trò chuyện của người dùng .	126
Hình 4.1	Logo ReactJs	127
Hình 4.2	Logo TypeScript	128
Hình 4.3	Logo NodeJS	128
Hình 4.4	Logo MySQL	129
Hình 4.5	Logo MongoDB	130
Hình 4.6	Logo Postman	131
Hình 4.7	Logo Docker	132
Hình 4.8	Quy trình CI/CD	133
Hình 4.9	Kiến trúc microservices	133

DANH MỤC BẢNG BIỂU

Bảng 1.1	Phân tích trường hợp sử dụng chức năng tạo tài khoản người dùng	5
Bảng 1.2	Phân tích trường hợp sử dụng chức năng đăng nhập/đăng xuất . .	6
Bảng 1.3	Phân tích trường hợp sử dụng chức năng quản lý tài khoản người dùng	7
Bảng 1.4	Phân tích trường hợp sử dụng chức năng tra cứu danh sách người dùng	8
Bảng 1.5	Phân tích trường hợp sử dụng chức năng tra cứu danh sách bác sĩ .	9
Bảng 1.6	Phân tích trường hợp sử dụng chức năng tra cứu danh sách bệnh nhân	9
Bảng 1.7	Phân tích trường hợp sử dụng chức năng tra cứu danh sách người dùng	10
Bảng 1.8	Phân tích trường hợp sử dụng chức năng chỉnh sửa thông tin người dùng	10
Bảng 1.9	Phân tích trường hợp sử dụng chức năng xóa người dùng khỏi hệ thống	11
Bảng 1.10	Phân tích trường hợp sử dụng chức năng tra cứu danh sách thiết bị	12
Bảng 1.11	Phân tích trường hợp sử dụng chức năng thêm thiết bị	12
Bảng 1.12	Phân tích trường hợp sử dụng chức năng phân công thiết bị y tế . .	13
Bảng 1.13	Phân tích trường hợp sử dụng chức năng chỉnh sửa thông tin thiết bị	13
Bảng 1.14	Phân tích trường hợp sử dụng chức năng xóa thiết bị	14
Bảng 1.15	Phân tích trường hợp sử dụng chức năng tra cứu lịch sử các phiên đo	15
Bảng 1.16	Phân tích trường hợp sử dụng chức năng xem đồ thị các tín hiệu đã đo	15
Bảng 1.17	Phân tích trường hợp sử dụng chức năng tải bản ghi dữ liệu phiên đo	16
Bảng 1.18	Phân tích trường hợp sử dụng chức năng phiên đo	16
Bảng 1.19	Phân tích trường hợp sử dụng chức năng đặt lịch khám theo yêu cầu	18
Bảng 1.20	Phân tích trường hợp sử dụng chức năng tra cứu lịch khám trong hệ thống	18
Bảng 1.21	Phân tích trường hợp sử dụng chức năng tra cứu lịch khám cá nhân	19

Bảng 1.22	Phân tích trường hợp sử dụng chức năng tra cứu lịch khám của bệnh nhân đang được theo dõi	19
Bảng 1.23	Phân tích trường hợp sử dụng chức năng phê duyệt lịch khám . . .	20
Bảng 1.24	Phân tích trường hợp sử dụng chức năng điền chẩn đoán và đặt lịch tái khám	20
Bảng 1.25	Phân tích trường hợp sử dụng chức năng xem lịch sử trò chuyện . .	21
Bảng 1.26	Phân tích trường hợp sử dụng chức năng gửi tin nhắn	22
Bảng 1.27	Phân tích trường hợp sử dụng chức năng tạo nhóm trò chuyện . . .	22
Bảng 2.1	Thẻ CRC lớp Tài khoản người dùng	23
Bảng 2.2	Thẻ CRC lớp Token	24
Bảng 2.3	Thẻ CRC lớp Vai trò người dùng	24
Bảng 2.4	Thẻ CRC lớp Trạng thái người dùng	25
Bảng 2.5	Thẻ CRC lớp Người dùng	26
Bảng 2.6	Thẻ CRC lớp Loại thiết bị	27
Bảng 2.7	Thẻ CRC lớp Trạng thái thiết bị	27
Bảng 2.8	Thẻ CRC lớp Thiết bị	28
Bảng 2.9	Thẻ CRC lớp Thông số kỹ thuật	29
Bảng 2.10	Thẻ CRC lớp Dữ liệu phiên đo	29
Bảng 2.11	Thẻ CRC lớp Trạng thái lịch khám	30
Bảng 2.12	Thẻ CRC lớp Kết quả lịch khám	31
Bảng 2.13	Thẻ CRC lớp Lịch khám	32
Bảng 2.14	Thẻ CRC lớp Thông báo liên quan đến lịch khám	33
Bảng 2.15	Thẻ CRC lớp Chẩn đoán cho bệnh nhân	34
Bảng 2.16	Thẻ CRC lớp Tin nhắn	34
Bảng 2.17	Thẻ CRC lớp Nhóm trò chuyện	35
Bảng 3.1	Bảng chuẩn hoá bảng Tài khoản đăng nhập	77
Bảng 3.2	Bảng chuẩn hoá bảng Token đăng nhập	77
Bảng 3.3	Bảng chuẩn hoá bảng Vai trò người dùng	78
Bảng 3.4	Bảng chuẩn hoá bảng Trạng thái hoạt động	78
Bảng 3.5	Bảng chuẩn hoá bảng Người dùng	78

Bảng 3.6	Bảng chuẩn hoá bảng Loại thiết bị	79
Bảng 3.7	Bảng chuẩn hoá bảng Trạng thái thiết bị	79
Bảng 3.8	Bảng chuẩn hoá bảng Thiết bị	79
Bảng 3.9	Bảng chuẩn hoá bảng Thông số kỹ thuật	80
Bảng 3.10	Bảng chuẩn hoá bảng Dữ liệu phiên đo	80
Bảng 3.11	Bảng chuẩn hoá bảng Trạng thái lịch khám	80
Bảng 3.12	Bảng chuẩn hoá bảng Kết quả lịch khám	81
Bảng 3.13	Bảng chuẩn hoá bảng Lịch khám	81
Bảng 3.14	Bảng chuẩn hoá bảng Thông báo liên quan đến lịch khám	81
Bảng 3.15	Bảng chuẩn hoá bảng Chẩn đoán	82
Bảng 3.16	Bảng chuẩn hoá bảng Tin nhắn	82
Bảng 3.17	Bảng chuẩn hoá bảng Nhóm trò chuyện	82
Bảng 3.18	Bảng API xác minh tài khoản	83
Bảng 3.19	Bảng API quản lý người dùng trong hệ thống	84
Bảng 3.20	Bảng API quản lý thiết bị y tế	84
Bảng 3.21	Bảng API quản lý dữ liệu phiên đo	85
Bảng 3.22	Bảng API quản lý dịch vụ lịch khám	85
Bảng 3.23	Bảng API liên quan đến chẩn đoán cho bệnh nhân	86
Bảng 3.24	Bảng API liên quan đến thông báo về lịch khám	86
Bảng 3.25	Bảng API liên quan đến tin nhắn	86
Bảng 4.1	Bảng kiểm thử API đăng ký tài khoản	135
Bảng 4.2	Bảng kiểm thử API người dùng đăng nhập	135
Bảng 4.3	Bảng kiểm thử API đăng xuất người dùng	136
Bảng 4.4	Bảng kiểm thử API lấy danh sách người dùng	137
Bảng 4.5	Bảng kiểm thử API lấy danh sách bác sĩ	137
Bảng 4.6	Bảng kiểm thử API lấy dữ liệu người dùng bằng id	138
Bảng 4.7	Bảng kiểm thử API lấy danh sách bệnh nhân theo id của bác sĩ . . .	138
Bảng 4.8	Bảng kiểm thử API lấy danh sách bác sĩ theo id của bệnh nhân . . .	139
Bảng 4.9	Bảng kiểm thử API cập nhật thông tin người dùng	140

Bảng 4.10	Bảng kiểm thử API xóa thông tin người dùng	141
Bảng 4.11	Bảng kiểm thử API lấy dữ liệu thông kê	141
Bảng 4.12	Bảng kiểm thử API lấy danh sách thiết bị	142
Bảng 4.13	Bảng kiểm thử API lấy dữ liệu thiết bị theo id	142
Bảng 4.14	Bảng kiểm thử API thêm thiết bị	143
Bảng 4.15	Bảng kiểm thử API thêm thông số kỹ thuật thiết bị	144
Bảng 4.16	Bảng kiểm thử API xóa thiết bị theo id	145
Bảng 4.17	Bảng kiểm thử API cập nhật thông tin thiết bị	146
Bảng 4.18	Bảng kiểm thử API cập nhật thông số kỹ thuật thiết bị	147
Bảng 4.19	Bảng kiểm thử API xóa thông số kỹ thuật thiết bị	148
Bảng 4.20	Bảng kiểm thử API lấy tất cả dữ liệu phiên đo	149
Bảng 4.21	Bảng kiểm thử API lấy dữ liệu phiên đo theo id	149
Bảng 4.22	Bảng kiểm thử API lấy các dữ liệu phiên đo theo id của bác sĩ . . .	150
Bảng 4.23	Bảng kiểm thử API thêm dữ liệu phiên đo	151
Bảng 4.24	Bảng kiểm thử API cập nhật dữ liệu dữ liệu phiên đo	152
Bảng 4.25	Bảng kiểm thử API xóa dữ liệu dữ liệu phiên đo theo id	153
Bảng 4.26	Bảng kiểm thử API lấy tất cả lịch hẹn của các bác sĩ - các bệnh nhân	154
Bảng 4.27	Bảng kiểm thử API lấy danh sách lịch hẹn theo id bác sĩ	154
Bảng 4.28	Bảng kiểm thử API lấy danh sách lịch hẹn theo id bệnh nhân	154
Bảng 4.29	Bảng kiểm thử API tạo lịch hẹn bởi bác sĩ	155
Bảng 4.30	Bảng kiểm thử API tạo lịch hẹn bởi bệnh nhân	156
Bảng 4.31	Bảng kiểm thử API lấy danh sách bác sĩ có thể đặt lịch hẹn theo thời gian cụ thể	158
Bảng 4.32	Bảng kiểm thử API lấy danh sách thời gian có thể đặt lịch hẹn của bác sĩ cụ thể	158
Bảng 4.33	Bảng kiểm thử API bác sĩ chấp nhận lịch hẹn từ bệnh nhân	158
Bảng 4.34	Bảng kiểm thử API bác sĩ từ chối lịch hẹn từ bệnh nhân	159
Bảng 4.35	Bảng kiểm thử API tạo chẩn đoán mới	160
Bảng 4.36	Bảng kiểm thử API lấy thông tin chẩn đoán theo id lịch khám . . .	160
Bảng 4.37	Bảng kiểm thử API cập nhật chẩn đoán theo id	161

Bảng 4.38	Bảng kiểm thử API lấy thông báo theo id của người dùng	161
Bảng 4.39	Bảng kiểm thử API tạo thông báo	162
Bảng 4.40	Bảng kiểm thử API cập nhật trạng thái thông báo đã được xem . .	163
Bảng 4.41	Bảng kiểm thử API xóa thông báo theo id	163
Bảng 4.42	Bảng kiểm thử API tạo nhóm trò chuyện	164
Bảng 4.43	Bảng kiểm thử API tìm danh sách nhóm trò chuyện của người dùng	165
Bảng 4.44	Bảng kiểm thử API lấy tin nhắn theo id nhóm trò chuyện	165
Bảng 4.45	Bảng kiểm thử API gửi tin nhắn trong nhóm trò chuyện	166
Bảng 4.46	Bảng kiểm thử chức năng của website quản trị	166

TÓM TẮT ĐỒ ÁN

Đồ án "Hệ thống quản lý dữ liệu điện tim và tương tác giữa bệnh nhân - bác sĩ" phát triển một hệ thống tích hợp gồm Web/App/Server, hỗ trợ quản lý dữ liệu đo điện tim, đặt lịch hẹn và cung cấp kênh giao tiếp hiệu quả giữa bệnh nhân và bác sĩ. Hệ thống cho phép lưu trữ thông tin người dùng, dữ liệu lịch hẹn, các bản ghi điện tim và nội dung trao đổi giữa bệnh nhân và bác sĩ trên server, giúp người dùng dễ dàng tra cứu khi cần.

Một trong những điểm nổi bật của hệ thống là khả năng lưu trữ và hiển thị dữ liệu đo điện tim dưới dạng biểu đồ trực quan. Tính năng này giúp bệnh nhân theo dõi sức khỏe lâu dài và cung cấp dữ liệu giá trị hỗ trợ bác sĩ trong việc chẩn đoán và điều trị. Ngoài ra, hệ thống hỗ trợ đặt lịch hẹn trực tuyến, giúp bệnh nhân lựa chọn bác sĩ và thời gian thuận tiện. Bác sĩ có thể phê duyệt hoặc từ chối yêu cầu, với thông báo tự động gửi đến bệnh nhân. Sau khi lịch hẹn được xác nhận, bệnh nhân và bác sĩ có thể sử dụng tính năng nhắn tin để trao đổi thông tin sức khỏe.

Đồ án được thực hiện dựa trên quy trình phát triển phần mềm tiêu chuẩn, bao gồm các bước xác định yêu cầu, phân tích, thiết kế, triển khai và kiểm thử hệ thống. Phương pháp phân tích và thiết kế hướng đối tượng được áp dụng, với các luồng xử lý và hành động được biểu diễn thông qua sơ đồ UML. Ứng dụng Web sử dụng ReactJS, server được xây dựng trên nền NodeJS với framework NestJS, và cơ sở dữ liệu được triển khai bằng MySQL cùng MongoDB.

Nội dung quyển đồ án được trình bày theo quy trình phát triển phần mềm, với các chương lần lượt gồm: phân tích hệ thống, thiết kế hệ thống, triển khai và kiểm thử, kết thúc bằng phần kết luận. Các nội dung được diễn giải chi tiết kèm theo sơ đồ minh họa cụ thể.

PHẦN MỞ ĐẦU

Đặt vấn đề

Sau đại dịch Covid-19, sức khỏe đã trở thành mối quan tâm hàng đầu của người dân Việt Nam. Trong bối cảnh môi trường sống ngày càng ô nhiễm, thời gian tiếp xúc với các thiết bị điện tử tăng cao, và thói quen ít vận động thể thao trở nên phổ biến, ngày càng nhiều người bắt đầu chú ý hơn đến các tín hiệu quan trọng từ cơ thể như nhịp tim, huyết áp và nhiệt độ. Điều này càng nhấn mạnh tầm quan trọng của việc chăm sóc sức khỏe tại nhà.

Với những người muốn chủ động theo dõi tình trạng sức khỏe, các thiết bị đo lường nhỏ gọn kết hợp cùng ứng dụng di động hỗ trợ đã trở thành lựa chọn không thể thiếu. Tuy nhiên, đối với những người đang gặp vấn đề sức khỏe, việc phải đối mặt với cảnh đông đúc, chờ đợi lâu tại các bệnh viện lại là một thử thách lớn, gây ra không ít phiền toái và căng thẳng.

Vậy, liệu có một giải pháp hiệu quả nào cho phép người dùng theo dõi tình trạng sức khỏe ngay tại nhà mà vẫn nhận được sự tư vấn và hỗ trợ từ đội ngũ chuyên gia y tế một cách kịp thời và tiện lợi? Đây chính là vấn đề đặt ra, đòi hỏi một phương án tối ưu để cân bằng giữa nhu cầu chăm sóc sức khỏe cá nhân và sự hỗ trợ chuyên môn từ y tế.

Đề xuất hệ thống

Nhận thấy nhu cầu chăm sóc sức khỏe tại nhà không chỉ giới hạn ở những người gặp vấn đề về sức khỏe mà còn mở rộng đến cả những người bình thường mong muốn theo dõi tình trạng cơ thể, đề án của chúng em đề xuất một hệ thống IoT tiên tiến để quản lý và theo dõi dữ liệu điện tim. Hệ thống này bao gồm các thành phần chính: Website, Server, Ứng dụng di động, và Thiết bị đo điện tim. Trong phạm vi đề án, chúng em tập trung vào phát triển Server và Website.

Khi làm việc cùng nhóm phần cứng, chúng em đã được tiếp cận thiết bị đo điện tim không tiếp xúc, từ đó định hướng xây dựng một hệ thống quản lý các thiết bị đo, thu thập và lưu trữ dữ liệu theo thời gian thực, đồng thời đáp ứng nhu cầu phân tích chuyên sâu. Hệ thống cũng tích hợp các tính năng như lập lịch hẹn và nhắn tin giữa bệnh nhân và bác sĩ, giúp nâng cao hiệu quả chăm sóc sức khỏe và tối ưu hóa trải nghiệm người dùng.

Cụ thể, hệ thống sẽ bao gồm:

- Một website dành cho bệnh nhân, cung cấp khả năng xem thông tin cá nhân, theo dõi lịch sử đo, quản lý và đặt lịch hẹn, đồng thời trao đổi tin nhắn trực tiếp

với bác sĩ.

- Một website dành cho bác sĩ, hỗ trợ quản lý lịch hẹn, phản hồi yêu cầu, theo dõi kết quả đo của các bệnh nhân đã đặt lịch thành công, và trao đổi tin nhắn trực tiếp với bệnh nhân.
- Một website dành cho admin, cho phép quản lý toàn bộ hệ thống, bao gồm người dùng, thiết bị đo và các dữ liệu đo tương ứng.
- Một server để lưu cơ sở dữ liệu liên quan đến người dùng và dữ liệu đo của bệnh nhân, có thể phục vụ cho công tác nghiên cứu và phân tích dữ liệu sau này

Mục tiêu của đề tài

Sau khi đã trình bày đề xuất về một hệ thống theo dõi và quản lý dữ liệu điện tim, mục tiêu đặt ra khi thực hiện đề tài này đó là:

- Hiểu được cơ sở lý thuyết và phương pháp thiết kế một hệ thống phần mềm.
- Hoàn thiện Website và Server được đề ra trong mục Đề xuất hệ thống với các chức năng hoạt động ổn định.
- Đảm bảo hệ thống hoạt động đồng bộ với các thiết bị phần cứng đang được nghiên cứu và phát triển.
- Cung cấp tài liệu tham khảo đầy đủ và đảm bảo tính chính xác, trung thực.

Phương pháp nghiên cứu

Trong đồ án này, chúng em đã áp dụng kết hợp nhiều phương pháp nghiên cứu. Đầu tiên, chúng em tham khảo thông tin từ các bài báo khoa học và các sản phẩm liên quan đến thiết bị đo điện tim được phát triển trong các phòng nghiên cứu tại trường. Đồng thời, chúng em tìm hiểu cách các hệ thống phần cứng và phần mềm kết nối và hoạt động cùng nhau.

Sau khi nắm vững cơ sở lý thuyết, chúng em tiến hành thực nghiệm, thu thập và lưu trữ dữ liệu từ các bản ghi đo đặc, đồng thời xây dựng các biểu đồ để trực quan hóa các chỉ số sức khỏe. Để đảm bảo dữ liệu được truyền tải và xử lý chính xác, nhóm đã phối hợp với đội firmware kiểm tra tính toàn vẹn của hệ thống. Ngoài ra, chúng em cũng tham vấn các chuyên gia y tế, đặc biệt trong lĩnh vực tim mạch, để xác thực độ chính xác của đồ thị và các kết quả thu được, đảm bảo hệ thống đáp ứng yêu cầu chuyên môn.

Kết quả đạt được

Trong quá trình thực hiện đồ án, chúng em, Cồ Huy Dũng và Nguyễn Đức Dương, đã có cơ hội nghiên cứu chuyên sâu về phần cứng, hệ thống IoT và cách các thành phần trong hệ thống được kết nối và hoạt động cùng nhau. Những kết quả đạt được tính đến thời điểm hoàn thiện quyển đồ án bao gồm:

- Hoàn thiện quyền đồ án với nội dung chi tiết về các bước xây dựng và phát triển hệ thống.
- Hoàn thiện các sản phẩm ứng dụng theo kế hoạch đã đề ra trong phần Đề xuất hệ thống, với khả năng kết nối các thành phần một cách hiệu quả, tích hợp theo dõi, lưu trữ dữ liệu điện tim trên server, đáp ứng tốt nhu cầu phân tích, nghiên cứu và quản lý dữ liệu trong tương lai. Đồng thời tích hợp các tính năng tương tác giữa bệnh nhân và bác sĩ như đặt lịch hẹn, nhắn tin trực tiếp.
- Nâng cao kỹ năng làm việc nhóm, viết báo cáo đồ án, phối hợp chặt chẽ với nhóm firmware và các chuyên gia y tế, góp phần hoàn thiện và nâng cao chất lượng sản phẩm.

Cấu trúc đồ án

- Giới thiệu mục đích của đồ án, đề xuất hệ thống, phân tích tính khả thi và trình bày bố cục nội dung.
- Chương 1: Thu thập và xác định yêu cầu hệ thống, bao gồm kỹ thuật thu thập thông tin, xác định yêu cầu chi tiết và thiết kế các sơ đồ các trường hợp sử dụng của hệ thống.
- Chương 2: Phân tích hệ thống, tập trung vào mô tả thẻ CRC, thiết kế sơ đồ lớp và sơ đồ tuần tự.
- Chương 3: Thiết kế hệ thống, bao gồm sơ đồ kiến trúc tổng thể, sơ đồ khối phần mềm, thiết kế cơ sở dữ liệu, giao diện người dùng, sơ đồ lớp và các chức năng hệ thống.
- Chương 4: Triển khai và kiểm thử, trình bày quá trình cài đặt hệ thống và kiểm tra tính năng.
- Phần kết luận: Kết luận và đề xuất hướng phát triển trong tương lai.

CHƯƠNG 1. THU THẬP YÊU CẦU

Chương này sẽ tiến hành thu thập yêu cầu cho dự án đề tài "Hệ thống quản lý dữ liệu điện tim và tương tác giữa bệnh nhân - bác sĩ" dựa trên các mục tiêu đã nêu ra trong Mục Đề xuất hệ thống ở Phần mở đầu.

1.1 Phân tích yêu cầu hệ thống

1.1.1 Yêu cầu chức năng của hệ thống

Hệ thống được xây dựng để đáp ứng nhu cầu riêng của từng nhóm người dùng, mang đến các tính năng giúp đơn giản hóa quy trình thu thập, xử lý và lưu trữ dữ liệu y tế. Đồng thời, hệ thống còn đóng vai trò là cầu nối, tăng cường sự tương tác hiệu quả giữa bệnh nhân và bác sĩ, góp phần nâng cao trải nghiệm người dùng và chất lượng chăm sóc sức khỏe. Các chức năng chính bao gồm:

- Ghi dữ liệu đo từ thiết bị: Ứng dụng trên điện thoại thu nhận dữ liệu đo điện tim từ thiết bị thông qua kết nối Bluetooth. Dữ liệu này được lưu trữ dưới dạng file định dạng .csv và đồng bộ trực tiếp lên máy chủ hệ thống, đảm bảo an toàn và sẵn sàng cho việc truy cập, phân tích sau này.
- Hiển thị dữ liệu: Các số liệu được lưu trữ trên máy chủ sẽ được xử lý và tính toán theo các công thức chuyên môn, sau đó được hiển thị trực quan trên website và ứng dụng di động dưới dạng đồ thị đường, giúp bác sĩ dễ dàng theo dõi và phân tích các chỉ số sức khỏe.
- Lưu trữ dữ liệu: Hệ thống hỗ trợ lưu trữ dữ liệu đo điện tim từ thiết bị trên cả ứng dụng di động và máy chủ trung tâm. Dữ liệu được đồng bộ hóa tự động từ ứng dụng lên máy chủ, đảm bảo an toàn và bảo mật tuyệt đối. Việc lưu trữ song song này giúp bảo vệ dữ liệu quan trọng, đảm bảo tính toàn vẹn và khả năng truy cập cho mục đích phân tích hoặc sử dụng trong tương lai, mang lại sự tin cậy cao cho người dùng.
- Trao đổi và chia sẻ thông tin về dữ liệu y tế: Hệ thống giúp người dùng có thể trao đổi trực tiếp với bác sĩ, chia sẻ kết quả đo điện tim, hỏi đáp về các vấn đề sức khỏe và các vấn đề liên quan đến thiết bị. Ngoài ra, hệ thống còn hỗ trợ người dùng đặt lịch khám trực tiếp với bác sĩ, mang lại sự tiện lợi và hỗ trợ đáng kể cho người dùng trong việc xác định tình trạng sức khỏe hiện tại và nhận được sự tư vấn kịp thời.

Đối với bệnh nhân:

- Đăng nhập và đăng ký tài khoản bằng thông tin cá nhân, bao gồm tên, địa chỉ

email, ngày sinh, số điện thoại và mật khẩu.

- Cập nhật các thông tin cá nhân.
- Được theo dõi điện tim trực tiếp khi kết nối ứng dụng di động với thiết bị đo điện tim thông qua Bluetooth.
- Xem kết quả các phiên đo của mình, bao gồm biểu đồ và các thông số liên quan.
- Nhận thông báo và có thể trao đổi trực tiếp với bác sĩ về tình hình sức khỏe và các kết quả đo được từ thiết bị thông qua đặt lịch khám hoặc tin nhắn trực tiếp.

Đối với bác sĩ:

- Đăng nhập và đăng ký tài khoản bằng thông tin cá nhân, bao gồm tên, địa chỉ email, ngày sinh, số điện thoại và mật khẩu.
- Cập nhật thông tin cá nhân.
- Quản lý danh sách lịch hẹn, chấp nhận hoặc từ chối lịch hẹn đã đặt của bệnh nhân.
- Quản lý danh sách bệnh nhân đã được chấp nhận lịch hẹn.
- Quản lý danh sách các bản ghi dữ liệu đo của các bệnh nhân đã đặt lịch hẹn thành công.
- Nhận thông báo và có thể trao đổi trực tiếp với bệnh nhân về tình hình sức khỏe và các kết quả đo được từ thiết bị.

Đối với quản trị viên:

- Đăng nhập và đăng ký tài khoản bằng thông tin cá nhân, bao gồm tên, địa chỉ email, số điện thoại và mật khẩu.
- Cập nhật thông tin cá nhân.
- Quản lý danh sách người dùng trong hệ thống, bao gồm bệnh nhân và bác sĩ.
- Quản lý danh sách thiết bị, chịu trách nhiệm phân công thiết bị cho người dùng.
- Quản lý các bản ghi đã đo được.
- Quản lý lịch hẹn của toàn bộ người dùng.

1.1.2 Yêu cầu phi chức năng của hệ thống

- Hệ thống có thể tương thích với hầu hết các trình duyệt phổ biến hiện nay.
- Hệ thống đảm bảo tính bảo mật và quyền riêng tư thông tin của người dùng.
- Hệ thống phải có giao diện người dùng thân thiện, dễ sử dụng để có thể tương tác mà không gặp quá nhiều khó khăn.

- Thời gian phản hồi của hệ thống phải nhanh chóng và ổn định.
- Hệ thống cần được tối ưu hóa để hoạt động hiệu quả ngay cả khi có lưu lượng truy cập cao.
- Hệ thống cần được sao lưu dữ liệu định kỳ để đảm bảo tính an toàn và khả năng khôi phục dữ liệu khi cần thiết.

1.1.3 Yêu cầu về người dùng hệ thống

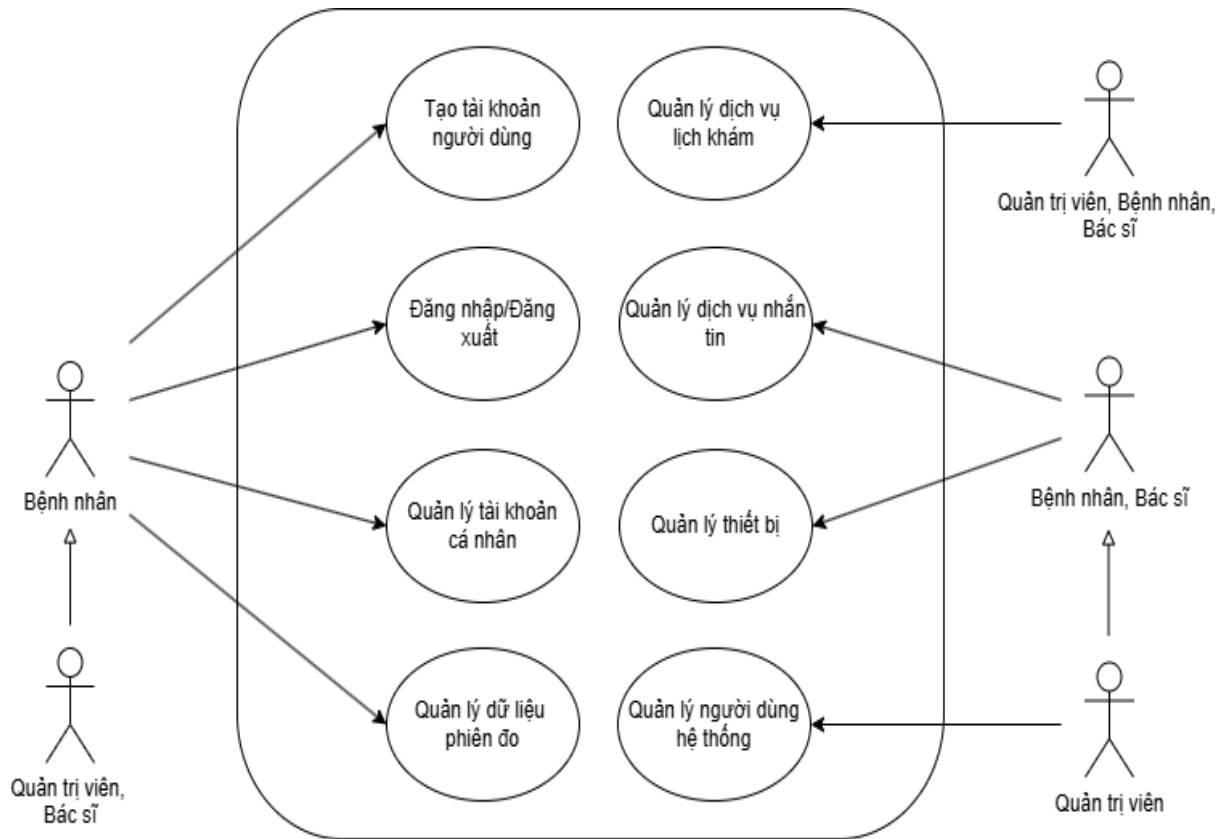
Hệ thống được thiết kế để phục vụ các đối tượng sau:

- Bệnh nhân: Sử dụng hệ thống để theo dõi dữ liệu điện tim của mình thông qua website. Bệnh nhân có thể đăng nhập vào tài khoản để truy cập thông tin cá nhân, xem danh sách bác sĩ, thông tin các bản ghi điện tim đã đo, đồng thời tìm kiếm và chọn lịch hẹn phù hợp để đăng ký. Ngoài ra, hệ thống còn cung cấp tính năng nhắn tin trực tiếp, giúp bệnh nhân dễ dàng trao đổi và nhận tư vấn từ bác sĩ.
- Bác sĩ: Hệ thống hỗ trợ bác sĩ quản lý và theo dõi tình trạng sức khỏe của các bệnh nhân đã đặt lịch hẹn thành công. Bác sĩ được phép truy cập kết quả các phiên đo, cung cấp tư vấn, trao đổi thông tin y tế, và chủ động đặt lịch tái khám khi cần. Ngoài ra, bác sĩ có thể tương tác với bệnh nhân qua tính năng nhắn tin hoặc nhóm chat để thảo luận và chia sẻ thông tin chuyên môn.
- Quản trị viên: Quản trị viên chịu trách nhiệm quản lý thông tin tài khoản của bệnh nhân, bác sĩ và các thiết bị y tế, đồng thời giám sát lịch khám và dữ liệu đo lường. Họ đảm bảo hệ thống vận hành chính xác, hiệu quả và đáp ứng nhu cầu của mọi người dùng.

Quá trình phân tích yêu cầu hệ thống đóng vai trò quan trọng trong việc xác định cụ thể các chức năng, yêu cầu phi chức năng, và các nhóm đối tượng người dùng mà hệ thống hướng tới phục vụ. Từ cơ sở này, việc thiết kế và phát triển hệ thống quản lý dữ liệu điện tim và tương tác giữa bệnh nhân và bác sĩ được triển khai, đảm bảo đáp ứng đầy đủ nhu cầu của người dùng, đồng thời tối ưu hóa về hiệu suất, bảo mật và khả năng sử dụng của hệ thống.

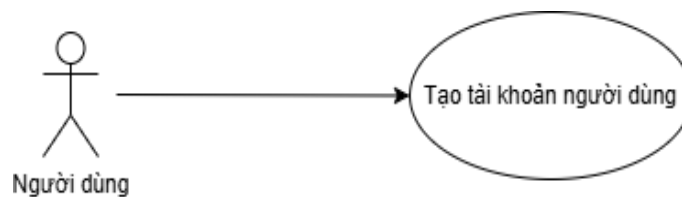
1.2 Sơ đồ các trường hợp sử dụng

1.2.1 Sơ đồ tổng quát các trường hợp sử dụng



Hình 1.1 Sơ đồ tổng quát các trường hợp sử dụng

1.2.2 Sơ đồ trường hợp sử dụng chức năng tạo tài khoản người dùng



Hình 1.2 Sơ đồ trường hợp sử dụng chức năng tạo tài khoản người dùng

Bảng 1.1 Phân tích trường hợp sử dụng chức năng tạo tài khoản người dùng

Tên chức năng	Tạo tài khoản người dùng
Tác nhân	Người sử dụng hệ thống (Bệnh nhân, Bác sĩ, Quản trị viên)
Mô tả	Hỗ trợ người dùng tạo tài khoản mới để truy cập các tính năng và dịch vụ của hệ thống
Điều kiện trước	Người dùng cần sử dụng thiết bị hỗ trợ kết nối Internet và chưa có tài khoản trong hệ thống
Dòng sự kiện chính	<ul style="list-style-type: none">- Người dùng truy cập màn hình tạo tài khoản mới- Nhập các thông tin cần thiết như email, mật khẩu và thông tin cá nhân- Hệ thống kiểm tra tính hợp lệ của thông tin được cung cấp- Nếu thông tin hợp lệ, tài khoản sẽ được tạo và thông báo thành công. Ngược lại, hệ thống hiển thị lỗi và yêu cầu người dùng nhập lại- Hoàn thành quá trình tạo tài khoản người dùng

1.2.3 Sơ đồ trường hợp sử dụng chức năng đăng nhập và đăng xuất khỏi hệ thống

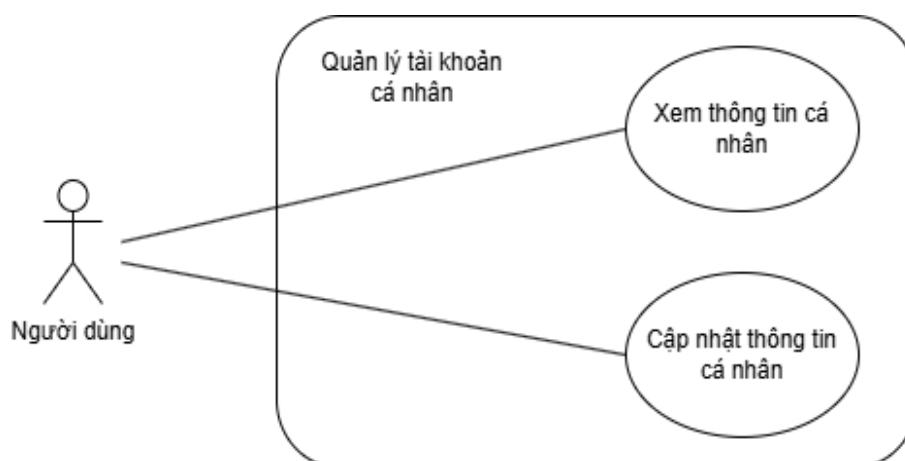


Hình 1.3 Sơ đồ trường hợp sử dụng chức năng đăng nhập và đăng xuất khỏi hệ thống

Bảng 1.2 Phân tích trường hợp sử dụng chức năng đăng nhập/đăng xuất

Tên chức năng	Đăng nhập và đăng xuất hệ thống
Tác nhân	Người sử dụng hệ thống
Mô tả	Cung cấp khả năng cho người dùng truy cập vào hệ thống bằng tài khoản đã đăng ký và rời khỏi hệ thống khi không cần sử dụng
Điều kiện trước	Người dùng phải có tài khoản hợp lệ
Dòng sự kiện chính	<p>Đăng nhập:</p> <ul style="list-style-type: none"> - Người dùng chọn tùy chọn đăng nhập - Nhập email và mật khẩu vào biểu mẫu - Hệ thống kiểm tra tính hợp lệ và xác thực tài khoản - Nếu thông tin hợp lệ, hệ thống cho phép truy cập và hiển thị trang chủ. Nếu không, hiển thị lỗi yêu cầu nhập lại tài khoản - Hoàn thành quá trình đăng nhập <p>Đăng xuất:</p> <ul style="list-style-type: none"> - Người dùng chọn tùy chọn đăng xuất - Hệ thống hủy phiên làm việc và đưa người dùng về giao diện đăng nhập - Hoàn thành quá trình đăng xuất

1.2.4 Sơ đồ trường hợp sử dụng chức năng quản lý tài khoản người dùng

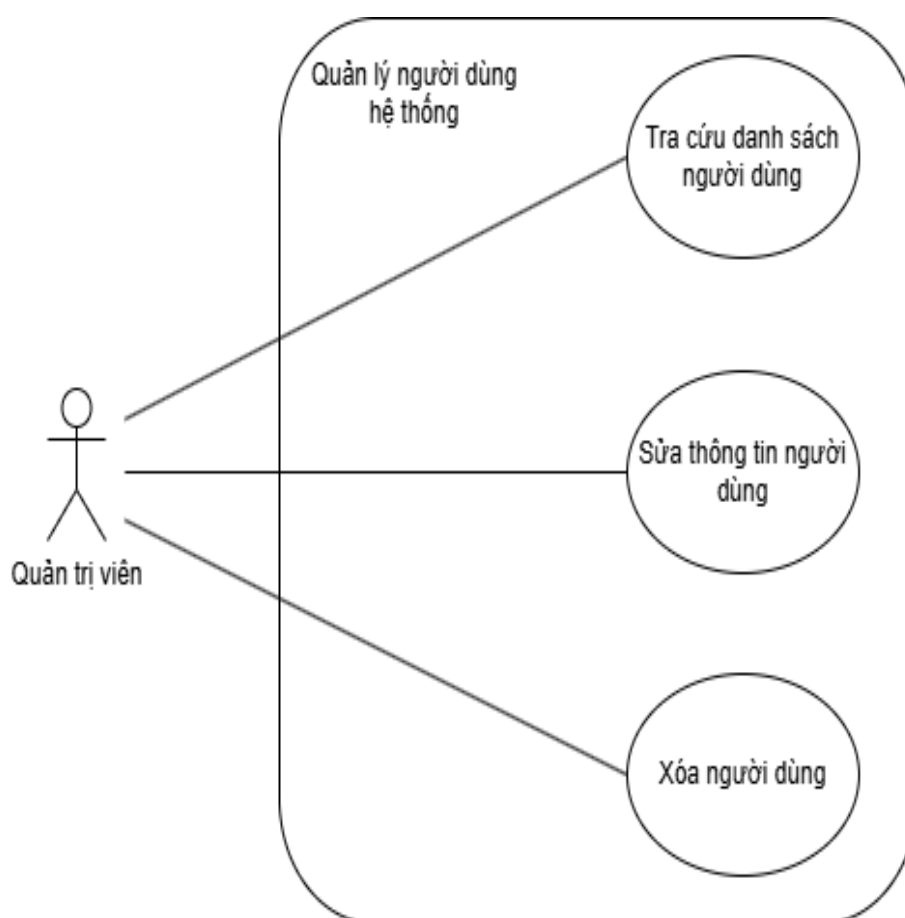


Hình 1.4 Sơ đồ trường hợp sử dụng chức năng quản lý tài khoản người dùng

Bảng 1.3 Phân tích trường hợp sử dụng chức năng quản lý tài khoản người dùng

Tên chức năng	Quản lý thông tin tài khoản người dùng
Tác nhân	Người sử dụng hệ thống
Mô tả	Hỗ trợ người dùng xem và chỉnh sửa các thông tin cá nhân được lưu trong hệ thống
Điều kiện trước	Người dùng cần đăng nhập vào hệ thống
Dòng sự kiện chính	<ul style="list-style-type: none">- Người dùng truy cập tính năng quản lý thông tin cá nhân- Hệ thống hiển thị các thông tin hiện tại của tài khoản- Người dùng thực hiện thay đổi thông tin cá nhân khi cần- Hệ thống kiểm tra tính chính xác và hợp lệ của thông tin đã cập nhật- Nếu thông tin hợp lệ, hệ thống tiến hành cập nhật và hiển thị thông báo. Nếu không, hiển thị lỗi và yêu cầu nhập lại- Hoàn thành quá trình quản lý thông tin tài khoản cá nhân

1.2.5 Sơ đồ trường hợp sử dụng chức năng quản lý người dùng hệ thống



Hình 1.5 Sơ đồ trường hợp sử dụng chức năng quản lý người dùng hệ thống

Bảng 1.4 Phân tích trường hợp sử dụng chức năng tra cứu danh sách người dùng

Tên chức năng	Tra cứu danh sách người dùng
Tác nhân	Quản trị viên
Mô tả	Hỗ trợ quản trị viên thực hiện các tác vụ như tra cứu và quản lý thông tin chi tiết của người dùng trong hệ thống
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none">- Quản trị viên truy cập tính năng tra cứu danh sách người dùng- Hệ thống hiển thị danh sách thông tin tất cả người dùng trong hệ thống- Quản trị viên thực hiện tìm kiếm và chọn một người dùng cụ thể- Hệ thống cung cấp thông tin chi tiết của người dùng được chọn- Hoàn thành quá trình tra cứu danh sách người dùng

Bảng 1.5 Phân tích trường hợp sử dụng chức năng tra cứu danh sách bác sĩ

Tên chức năng	Tra cứu danh sách bác sĩ
Tác nhân	Bệnh nhân
Mô tả	Hỗ trợ bệnh nhân tra cứu thông tin chi tiết của các bác sĩ trong hệ thống
Điều kiện trước	Bệnh nhân phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Bệnh nhân truy cập tính năng tra cứu danh sách bác sĩ - Hệ thống hiển thị danh sách các bác sĩ cùng thông tin cơ bản. - Bệnh nhân chọn bác sĩ cụ thể từ danh sách. - Hệ thống hiển thị thông tin chi tiết về bác sĩ được chọn. - Hoàn thành quá trình tra cứu danh sách bác sĩ trong hệ thống

Bảng 1.6 Phân tích trường hợp sử dụng chức năng tra cứu danh sách bệnh nhân

Tên chức năng	Tra cứu danh sách bệnh nhân đang theo dõi
Tác nhân	Bác sĩ
Mô tả	Hỗ trợ bác sĩ tìm kiếm và xem thông tin chi tiết của các bệnh nhân mà họ đang phụ trách trong hệ thống.
Điều kiện trước	Bác sĩ phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Bác sĩ truy cập tính năng tra cứu danh sách bệnh nhân đang phụ trách - Hệ thống hiển thị danh sách các bệnh nhân cùng thông tin cơ bản. - Bác sĩ chọn bệnh nhân cụ thể từ danh sách. - Hệ thống cung cấp thông tin chi tiết của bệnh nhân được chọn. - Hoàn thành quá trình tra cứu danh sách bệnh nhân

Bảng 1.7 Phân tích trường hợp sử dụng chức năng tra cứu danh sách người dùng

Tên chức năng	Tra cứu danh sách người dùng
Tác nhân	Quản trị viên
Mô tả	Hỗ trợ quản trị viên thực hiện các tác vụ như tra cứu và quản lý thông tin chi tiết của người dùng trong hệ thống
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên truy cập tính năng quản lý danh sách người dùng - Hệ thống hiển thị danh sách thông tin tất cả người dùng trong hệ thống - Quản trị viên thực hiện tìm kiếm và chọn một người dùng cụ thể - Hệ thống cung cấp thông tin chi tiết của người dùng được chọn - Hoàn thành quá trình tra cứu danh sách người dùng

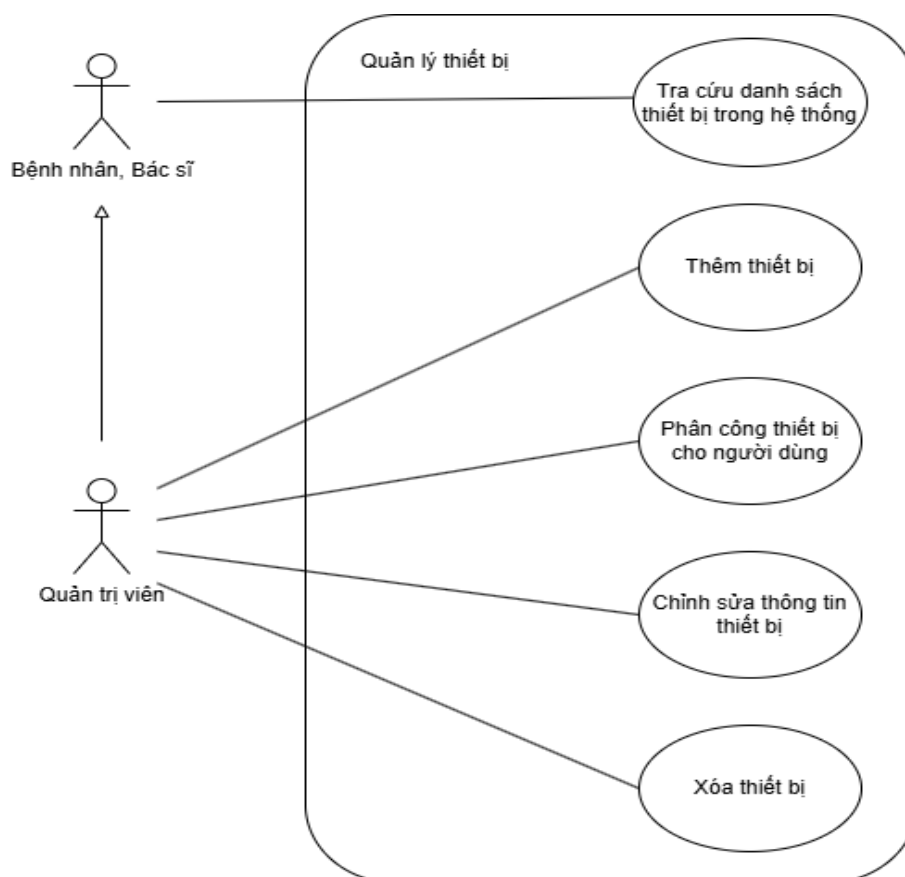
Bảng 1.8 Phân tích trường hợp sử dụng chức năng chỉnh sửa thông tin người dùng

Tên chức năng	Chỉnh sửa thông tin người dùng
Tác nhân	Quản trị viên
Mô tả	Hỗ trợ quản trị viên thực hiện việc chỉnh sửa thông tin chi tiết của người dùng trong hệ thống, đảm bảo thông tin được cập nhật chính xác và kịp thời
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên truy cập chức năng chỉnh sửa thông tin người dùng - Hệ thống hiển thị thông tin hiện tại của người dùng - Quản trị viên nhập thông tin cần chỉnh sửa - Hệ thống kiểm tra tính hợp lệ của thông tin mới - Nếu thông tin hợp lệ, hệ thống cập nhật thông tin và gửi thông báo thành công - Nếu thông tin không hợp lệ, hệ thống hiển thị lỗi và yêu cầu chỉnh sửa lại - Hoàn thành quá trình chỉnh sửa thông tin người dùng

Bảng 1.9 Phân tích trường hợp sử dụng chức năng xóa người dùng khỏi hệ thống

Tên chức năng	Xóa người dùng
Tác nhân	Quản trị viên
Mô tả	Hỗ trợ quản trị viên thực hiện việc xóa người dùng không còn hoạt động khỏi hệ thống
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Hệ thống hiển thị danh sách người dùng hiện có - Quản trị viên chọn các người dùng cần xóa khỏi hệ thống - Hệ thống hiển thị yêu cầu xác nhận xóa - Quản trị viên xác nhận hành động xóa - Hệ thống xóa người dùng được chọn và cập nhật dữ liệu đồng thời gửi thông báo xóa thành công - Hoàn thành quá trình xóa người dùng khỏi hệ thống

1.2.6 Sơ đồ trường hợp sử dụng chức năng quản lý thiết bị y tế



Hình 1.6 Sơ đồ trường hợp sử dụng chức năng quản lý thiết bị y tế

Bảng 1.10 Phân tích trường hợp sử dụng chức năng tra cứu danh sách thiết bị

Tên chức năng	Tra cứu danh sách thiết bị trong hệ thống
Tác nhân	Người sử dụng hệ thống
Mô tả	Hỗ trợ người dùng tra cứu danh sách thiết bị y tế trong hệ thống, bao gồm thông tin chi tiết về tình trạng và thời gian mượn của từng thiết bị.
Điều kiện trước	Người dùng cần đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Hệ thống hiển thị danh sách thiết bị theo quyền hạn của từng người dùng: + Đối với bệnh nhân và bác sĩ: Danh sách các thiết bị y tế đã mượn + Đối với quản trị viên: Danh sách toàn bộ thiết bị y tế trong hệ thống, bao gồm trạng thái mượn và người đang sử dụng Người dùng thực hiện tìm kiếm và lựa chọn thiết bị cần xem thông tin chi tiết - Hệ thống hiển thị đầy đủ thông tin chi tiết của thiết bị được chọn, kết thúc luồng sự kiện - Hoàn thành quá trình tra cứu danh sách thiết bị

Bảng 1.11 Phân tích trường hợp sử dụng chức năng thêm thiết bị

Tên chức năng	Thêm mới thiết bị vào hệ thống
Tác nhân	Quản trị viên
Mô tả	Hỗ trợ quản trị viên thêm thiết bị mới vào danh sách thiết bị của hệ thống, bao gồm thông tin chi tiết và trạng thái thiết bị
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên truy cập chức năng thêm thiết bị - Hệ thống hiển thị biểu mẫu nhập thông tin thiết bị - Quản trị viên nhập thông tin cần thiết (tên thiết bị, trạng thái, thời gian sử dụng,...) - Hệ thống kiểm tra tính hợp lệ của thông tin - Nếu thông tin hợp lệ, thiết bị được thêm vào hệ thống và hiển thị thông báo thành công. Ngược lại, hiển thị lỗi và yêu cầu chỉnh sửa thông tin - Hoàn thành quá trình thêm mới thiết bị

Bảng 1.12 Phân tích trường hợp sử dụng chức năng phân công thiết bị y tế

Tên chức năng	Phân công thiết bị y tế cho người dùng
Tác nhân	Quản trị viên
Mô tả	Cung cấp chức năng cho quản trị viên trong việc phân công thiết bị y tế từ hệ thống cho người dùng (bác sĩ hoặc bệnh nhân) để đảm bảo thiết bị được sử dụng hiệu quả
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên truy cập chức năng phân công thiết bị y tế - Hệ thống hiển thị danh sách thiết bị y tế hiện có và biểu mẫu phân công - Quản trị viên chọn thiết bị, chỉ định người dùng (bác sĩ hoặc bệnh nhân), và nhập thời gian mượn - Hệ thống xác minh tính hợp lệ của thông tin - Nếu thông tin hợp lệ, hệ thống cập nhật trạng thái thiết bị và thông báo kết quả thành công - Nếu thông tin không hợp lệ, hệ thống hiển thị lỗi và yêu cầu chỉnh sửa - Hoàn thành quá trình phân công thiết bị

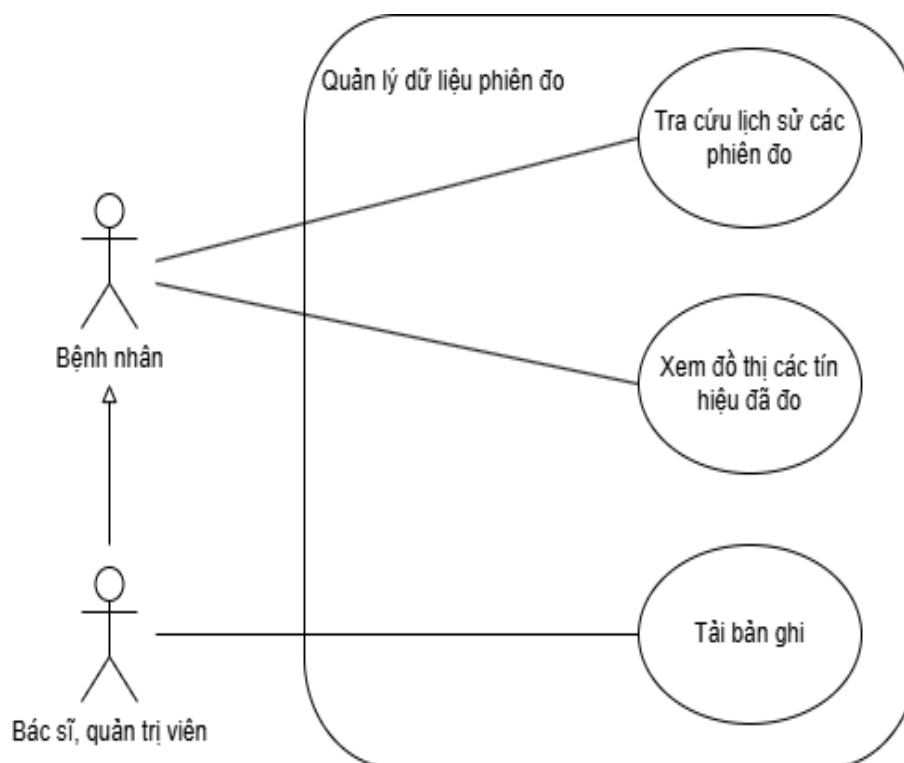
Bảng 1.13 Phân tích trường hợp sử dụng chức năng chỉnh sửa thông tin thiết bị

Tên chức năng	Chỉnh sửa thông tin thiết bị
Tác nhân	Quản trị viên
Mô tả	Hỗ trợ quản trị viên cập nhật thông tin chi tiết của thiết bị trong hệ thống, bao gồm các thông tin như tên thiết bị, trạng thái hoạt động và các thuộc tính khác
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên truy cập chức năng chỉnh sửa thông tin thiết bị - Hệ thống hiển thị thông tin hiện tại của thiết bị - Quản trị viên cập nhật các thông tin cần chỉnh sửa (tên thiết bị, trạng thái, thông số kỹ thuật,...) - Hệ thống kiểm tra tính hợp lệ của thông tin mới - Nếu thông tin hợp lệ, hệ thống cập nhật thành công và thông báo cho quản trị viên - Nếu thông tin không hợp lệ, hệ thống hiển thị lỗi và yêu cầu nhập lại - Hoàn thành quá trình chỉnh sửa thông tin thiết bị

Bảng 1.14 Phân tích trường hợp sử dụng chức năng xóa thiết bị

Tên chức năng	Xóa thiết bị khỏi hệ thống
Tác nhân	Quản trị viên
Mô tả	Hỗ trợ quản trị viên thực hiện việc xóa thiết bị không còn hoạt động hoặc không cần sử dụng khỏi danh sách thiết bị trong hệ thống
Điều kiện trước	Quản trị viên cần đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên truy cập chức năng xóa thiết bị - Hệ thống hiển thị danh sách thiết bị hiện có - Quản trị viên chọn một hoặc nhiều thiết bị cần xóa - Quản trị viên xác nhận hành động xóa - Hệ thống thực hiện xóa thiết bị được chọn, cập nhật danh sách thiết bị và gửi thông báo kết quả - Hoàn thành quá trình xóa thiết bị

1.2.7 Sơ đồ trường hợp sử dụng chức năng quản lý dữ liệu phiên đo



Hình 1.7 Sơ đồ trường hợp sử dụng chức năng quản lý dữ liệu phiên đo

Bảng 1.15 Phân tích trường hợp sử dụng chức năng tra cứu lịch sử các phiên đo

Tên chức năng	Tra cứu lịch sử các phiên đo
Tác nhân	Người sử dụng hệ thống
Mô tả	Hỗ trợ người sử dụng hệ thống truy cập và quản lý lịch sử các phiên đo theo quyền hạn
Điều kiện trước	Người dùng phải đăng nhập vào hệ thống
Dòng sự kiện chính	<ul style="list-style-type: none"> - Người dùng chọn tính năng xem lịch sử các phiên đo - Hệ thống hiển thị danh sách dữ liệu các phiên đo dựa trên quyền hạn của người dùng: <ul style="list-style-type: none"> + Đối với bệnh nhân: Danh sách các phiên đo cá nhân + Đối với bác sĩ: Danh sách các phiên đo của bệnh nhân mà bác sĩ đang theo dõi + Đối với quản trị viên: Danh sách toàn bộ dữ liệu phiên đo trong hệ thống - Hoàn thành quá trình tra cứu lịch sử các phiên đo

Bảng 1.16 Phân tích trường hợp sử dụng chức năng xem đồ thị các tín hiệu đã đo

Tên chức năng	Xem đồ thị các tín hiệu đã đo
Tác nhân	Người dùng hệ thống (Bệnh nhân, Bác sĩ)
Mô tả	Hỗ trợ bệnh nhân và bác sĩ truy cập và xem đồ thị dữ liệu từ các phiên đo, cung cấp cái nhìn trực quan về các thông số tim mạch.
Điều kiện trước	Người dùng cần đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Người dùng truy cập chức năng xem đồ thị dữ liệu phiên đo - Hệ thống hiển thị danh sách các phiên đo để người dùng lựa chọn <ul style="list-style-type: none"> + Đối với bệnh nhân: Danh sách các phiên đo cá nhân + Đối với bác sĩ: Danh sách các phiên đo của bệnh nhân mà bác sĩ đang theo dõi + Đối với quản trị viên: Danh sách toàn bộ dữ liệu phiên đo trong hệ thống - Người dùng chọn một phiên đo cụ thể để xem đồ thị - Hệ thống hiển thị đồ thị tương ứng dựa trên dữ liệu phiên đo đã chọn - Hoàn thành quá trình xem đồ thị các tín hiệu đã đo

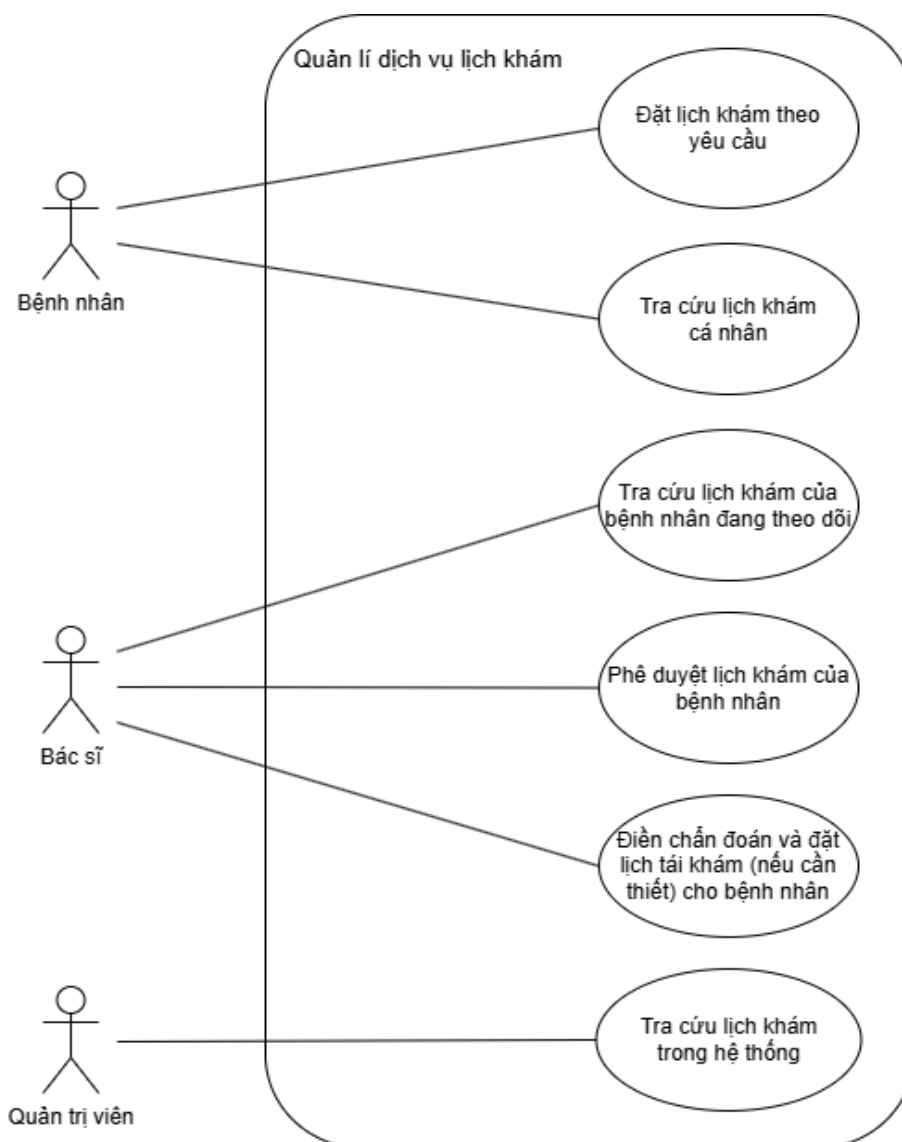
Bảng 1.17 Phân tích trường hợp sử dụng chức năng tải bản ghi dữ liệu phiên đo

Tên chức năng	Tải bản ghi dữ liệu phiên đo
Tác nhân	Bác sĩ, Quản trị viên
Mô tả	Cung cấp chức năng tải về bản ghi dữ liệu phiên đo. Bác sĩ có thể tải dữ liệu từ bệnh nhân mà mình quản lý, trong khi quản trị viên có quyền tải về bất kỳ bản ghi nào trong hệ thống
Điều kiện trước	Người dùng phải đăng nhập với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Người dùng truy cập chức năng tải bản ghi dữ liệu - Hệ thống hiển thị danh sách các phiên đo khả dụng theo từng vai trò người dùng - Người dùng chọn bản ghi cần tải xuống - Hệ thống xử lý và trả về tệp dữ liệu phiên đo để tải xuống - Tệp được tải thành công, hoàn thành quá trình tải bản ghi dữ liệu phiên đo

Bảng 1.18 Phân tích trường hợp sử dụng chức năng phiên đo

Tên chức năng	Xóa phiên đo
Tác nhân	Bác sĩ, Quản trị viên
Mô tả	Cung cấp chức năng xóa phiên đo.
Điều kiện trước	Người dùng phải đăng nhập với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Người dùng truy cập chức năng xóa dữ liệu phiên đo - Hệ thống hiển thị danh sách các phiên đo có thể xóa - Người dùng chọn phiên đo cần xóa dữ liệu - Hệ thống thực hiện xóa phiên đo được chọn, cập nhật danh sách phiên đo và gửi thông báo kết quả - Hoàn thành quá trình xóa dữ liệu phiên đo

1.2.8 Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ lịch khám



Hình 1.8 Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ lịch khám

Bảng 1.19 Phân tích trường hợp sử dụng chức năng đặt lịch khám theo yêu cầu

Tên chức năng	Đặt lịch khám theo yêu cầu
Tác nhân	Bệnh nhân
Mô tả	Cung cấp cho bệnh nhân khả năng linh hoạt trong việc đặt lịch khám, cho phép lựa chọn bác sĩ cụ thể hoặc thời gian rảnh phù hợp với nhu cầu cá nhân.
Điều kiện trước	Bệnh nhân phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Bệnh nhân truy cập chức năng đặt lịch khám theo yêu cầu - Hệ thống cung cấp hai lựa chọn: đặt lịch theo bác sĩ hoặc theo thời gian rảnh + Nếu đặt lịch theo bác sĩ: Hệ thống hiển thị danh sách bác sĩ và thời gian rảnh của từng bác sĩ để bệnh nhân lựa chọn + Nếu đặt lịch theo thời gian rảnh: Hệ thống hiển thị các khung thời gian trống và danh sách bác sĩ rảnh trong thời gian đó - Bệnh nhân chọn thời gian và bác sĩ phù hợp để đặt lịch - Hệ thống xử lý yêu cầu, lưu thông tin và thông báo đặt lịch thành công. Sau đó chờ lịch được xác nhận từ phía bác sĩ - Hoàn thành quá trình đặt lịch

Bảng 1.20 Phân tích trường hợp sử dụng chức năng tra cứu lịch khám trong hệ thống

Tên chức năng	Tra cứu lịch khám trong hệ thống
Tác nhân	Quản trị viên
Mô tả	Cho phép quản trị viên xem và tra cứu chi tiết các lịch khám đang có trong hệ thống
Điều kiện trước	Quản trị viên phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Quản trị viên truy cập chức năng tra cứu lịch khám - Hệ thống hiển thị danh sách các lịch khám trong hệ thống - Quản trị viên chọn một lịch khám cụ thể - Hệ thống hiển thị chi tiết lịch khám, bao gồm thời gian, bác sĩ phụ trách, bệnh nhân và thông tin chẩn đoán (nếu có) - Quá trình tra cứu lịch khám trong hệ thống hoàn tất

Bảng 1.21 Phân tích trường hợp sử dụng chức năng tra cứu lịch khám cá nhân

Tên chức năng	Tra cứu lịch khám cá nhân
Tác nhân	Bệnh nhân
Mô tả	Cung cấp chức năng cho bệnh nhân dễ dàng tra cứu và xem thông tin chi tiết các lịch khám đã được chấp nhận
Điều kiện trước	Bệnh nhân phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Bệnh nhân truy cập chức năng tra cứu lịch khám cá nhân - Hệ thống hiển thị danh sách các lịch khám của bệnh nhân đã được phê duyệt - Bệnh nhân chọn một lịch khám cụ thể để xem thông tin chi tiết - Hệ thống cung cấp thông tin chi tiết của lịch khám, bao gồm thời gian, bác sĩ phụ trách và chẩn đoán nếu có - Quá trình tra cứu lịch khám cá nhân hoàn tất

Bảng 1.22 Phân tích trường hợp sử dụng chức năng tra cứu lịch khám của bệnh nhân đang được theo dõi

Tên chức năng	Tra cứu lịch khám của bệnh nhân đang được theo dõi
Tác nhân	Bác sĩ
Mô tả	Cho phép bác sĩ tra cứu danh sách và xem thông tin chi tiết các lịch khám của bệnh nhân mà bác sĩ phụ trách
Điều kiện trước	Bác sĩ phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Bác sĩ truy cập chức năng tra cứu lịch khám của bệnh nhân đang được theo dõi - Hệ thống hiển thị danh sách các lịch khám đã được đặt (cả thành công và đang chờ xét duyệt) - Bác sĩ chọn một lịch khám cụ thể để xem thông tin chi tiết - Hệ thống cung cấp thông tin chi tiết của lịch khám, bao gồm thời gian, bệnh nhân đặt lịch và chẩn đoán nếu có - Quá trình tra cứu lịch khám của bệnh nhân đang được theo dõi hoàn tất

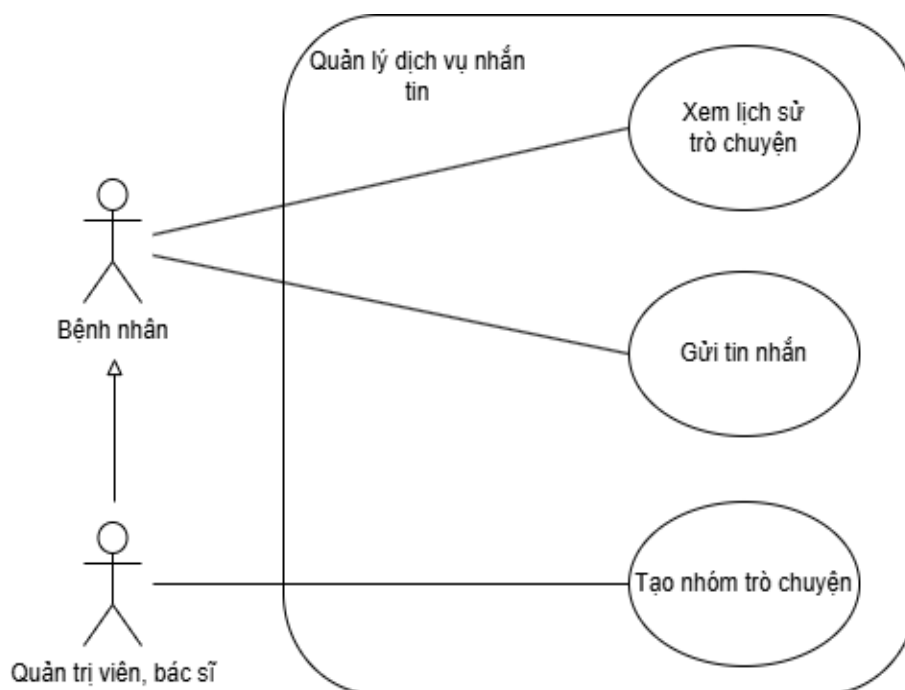
Bảng 1.23 Phân tích trường hợp sử dụng chức năng phê duyệt lịch khám

Tên chức năng	Phê duyệt lịch khám của bệnh nhân
Tác nhân	Bác sĩ
Mô tả	Cung cấp chức năng cho bác sĩ phê duyệt lịch khám của bệnh nhân, bao gồm chấp nhận hoặc từ chối các yêu cầu đặt lịch
Điều kiện trước	Bác sĩ phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Bác sĩ truy cập chức năng phê duyệt lịch khám của bệnh nhân đã đặt - Hệ thống hiển thị danh sách các lịch khám đang chờ phê duyệt - Bác sĩ chọn một lịch khám cụ thể để xem chi tiết thông tin - Bác sĩ thực hiện chấp nhận hoặc từ chối yêu cầu đặt lịch - Nếu chấp nhận: Hệ thống cập nhật trạng thái lịch khám thành "Đã phê duyệt" và thông báo cho bệnh nhân - Nếu từ chối: Hệ thống yêu cầu bác sĩ xác nhận và cung cấp lý do từ chối, sau đó thông báo cho bệnh nhân - Quá trình phê duyệt lịch khám hoàn tất

Bảng 1.24 Phân tích trường hợp sử dụng chức năng điền chẩn đoán và đặt lịch tái khám

Tên chức năng	Điền chẩn đoán và đặt lịch tái khám
Tác nhân	Bác sĩ
Mô tả	Hỗ trợ bác sĩ điền thông tin chẩn đoán cho từng lịch khám và đặt lịch tái khám nếu cần thiết
Điều kiện trước	Bác sĩ phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Bác sĩ truy cập chức năng điền chẩn đoán - Chọn lịch khám cần điền hoặc cập nhật chẩn đoán - Hệ thống hiển thị chi tiết lịch khám đã chọn - Bác sĩ điền hoặc cập nhật thông tin chẩn đoán - Nếu cần tái khám, bác sĩ nhấn nút "Đặt lịch tái khám" và nhập thời gian phù hợp - Hệ thống lưu thông tin và hoàn tất quy trình

1.2.9 Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ nhắn tin



Hình 1.9 Sơ đồ trường hợp sử dụng chức năng quản lý dịch vụ nhắn tin

Bảng 1.25 Phân tích trường hợp sử dụng chức năng xem lịch sử trò chuyện

Tên chức năng	Xem lịch sử trò chuyện
Tác nhân	Người sử dụng hệ thống
Mô tả	Cung cấp cho người dùng khả năng truy cập và xem lại lịch sử các đoạn hội thoại đã thực hiện
Điều kiện trước	Người dùng phải đăng nhập vào hệ thống với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Người dùng truy cập chức năng nhắn tin và chọn đoạn hội thoại mong muốn - Hệ thống hiển thị nội dung chi tiết của đoạn hội thoại đã chọn - Hoàn thành quá trình xem lịch sử trò chuyện

Bảng 1.26 Phân tích trường hợp sử dụng chức năng gửi tin nhắn

Tên chức năng	Gửi tin nhắn
Tác nhân	Người sử dụng hệ thống
Mô tả	Cung cấp chức năng cho phép người dùng gửi tin nhắn đến các đối tượng liên quan: bác sĩ chỉ gửi tin nhắn được với quản trị viên và những bệnh nhân mình theo dõi. Tương tự, bệnh nhân cũng chỉ gửi được tin nhắn cho bác sĩ đã chấp nhận lịch khám của mình
Điều kiện trước	Người dùng phải đăng nhập với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Người dùng truy cập chức năng gửi tin nhắn - Chọn người nhận hoặc nhóm nhận tin nhắn - Hệ thống hiển thị biểu mẫu để nhập nội dung tin nhắn - Người dùng nhập nội dung và nhấn nút gửi - Hệ thống gửi tin nhắn đến người nhận hoặc nhóm và thông báo thành công - Hoàn thành quá trình gửi tin nhắn

Bảng 1.27 Phân tích trường hợp sử dụng chức năng tạo nhóm trò chuyện

Tên chức năng	Tạo nhóm trò chuyện
Tác nhân	Quản trị viên, Bác sĩ
Mô tả	Cho phép người dùng tạo nhóm trò chuyện để trao đổi thông tin giữa các thành viên
Điều kiện trước	Người dùng phải đăng nhập với quyền truy cập hợp lệ
Dòng sự kiện chính	<ul style="list-style-type: none"> - Người dùng truy cập chức năng tạo nhóm trò chuyện - Hệ thống hiển thị biểu mẫu tạo nhóm - Người dùng nhập thông tin nhóm và thêm thành viên vào nhóm - Hệ thống tạo nhóm thành công và gửi thông báo đến các thành viên - Hoàn thành quá trình tạo nhóm trò chuyện

CHƯƠNG 2. TÌM HIỂU VÀ PHÂN TÍCH HỆ THỐNG

Chương này trình bày chi tiết quá trình phân tích hệ thống, dựa trên các yêu cầu đã được xác định trong Phần mở đầu và Chương 1. Nội dung bao gồm các bước sau:

- Thiết kế các thẻ CRC (Class - Responsibility - Collaboration Card) cùng các lớp dựa trên thông tin thu thập từ các sơ đồ use case mô tả chức năng của hệ thống.
- Xây dựng sơ đồ lớp với đầy đủ thuộc tính và phương thức, dựa trên các lớp đã được định nghĩa.
- Sau khi xác định thuộc tính, phương thức và chức năng của các lớp, tiếp tục thiết kế sơ đồ tuần tự để minh họa chi tiết sự tương tác giữa các lớp khi thực hiện các chức năng cụ thể.

2.1 Thẻ CRC (Class - Responsibility - Collaboration Card)

2.1.1 Thẻ CRC lớp Tài khoản người dùng

Bảng 2.1 Thẻ CRC lớp Tài khoản người dùng

Mặt trước thẻ

Tên lớp: Tài khoản người dùng (account)	ID: 0
Mô tả: Lớp quản lý thông tin tài khoản người dùng trong hệ thống	Use case liên quan: Đăng ký tài khoản, Quản lý tài khoản người dùng, Quản lý thông tin người dùng
Trách nhiệm (Responsibility): Xử lý đăng nhập và đăng xuất Đăng ký tài khoản mới cho người dùng	Các lớp cộng tác (Collaboration): Token đăng nhập Người dùng

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) user_email(string) user_password(string)	user_role(integer) createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Token đăng nhập, Người dùng	

2.1.2 Thẻ CRC lớp Token

Bảng 2.2 Thẻ CRC lớp Token

Mặt trước thẻ

Tên lớp: Token	ID: 1
Mô tả: Lớp quản lý thông tin token liên kết với mỗi tài khoản người dùng	Use case liên quan: Đăng nhập, đăng xuất
Trách nhiệm (Responsibility): Quản lý vòng đời của access_token và refresh_token (tạo mới, xóa khi cần) Xác minh token để kiểm tra tính hợp lệ trong các yêu cầu của người dùng Đảm bảo tính hợp lệ của phiên đăng nhập	Các lớp cộng tác (Collaboration): Tài khoản người dùng

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) account_id(string) refresh_token(string) expired_At(Datetime)	is_expired(integer) createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Tài khoản người dùng	

2.1.3 Thẻ CRC lớp Vai trò người dùng

Bảng 2.3 Thẻ CRC lớp Vai trò người dùng

Mặt trước thẻ

Tên lớp: Vai trò người dùng (user_role)	ID: 2
Mô tả: Lớp quản lý thông tin các vai trò của người dùng trong hệ thống	Use case liên quan: Quản lí thông tin người dùng
Trách nhiệm (Responsibility): Lưu trữ và quản lý các vai trò người dùng (như Bệnh nhân, Bác sĩ, Quản trị viên) Cung cấp thông tin vai trò để phân quyền truy cập trong hệ thống	Các lớp cộng tác (Collaboration): Người dùng

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) role_name(string)	createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Người dùng	

2.1.4 Thẻ CRC lớp Trạng thái người dùng

Bảng 2.4 Thẻ CRC lớp Trạng thái người dùng

Mặt trước thẻ

Tên lớp: Trạng thái người dùng (user_status)	ID: 3
Mô tả: Lớp quản lý thông tin các trạng thái của người dùng trong hệ thống	Use case liên quan: Quản lí thông tin người dùng
Trách nhiệm (Responsibility): Lưu trữ các trạng thái hoạt động của người dùng (ví dụ: hoạt động, bị khóa, tạm ngưng) Cung cấp thông tin trạng thái để hỗ trợ quản lý quyền truy cập hệ thống	Các lớp cộng tác (Collaboration): Người dùng

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) status_description(string)	createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Người dùng	

2.1.5 Thẻ CRC lớp Người dùng

Bảng 2.5 Thẻ CRC lớp Người dùng

Mặt trước thẻ

Tên lớp: Người dùng (User)	ID: 4
Mô tả: Lớp quản lý thông tin cá nhân và quyền hạn của người dùng trong hệ thống	Use case liên quan: Đăng ký tài khoản, Quản lý tài khoản người dùng, Quản lý lịch khám
Trách nhiệm (Responsibility): Quản lý thông tin người dùng, bao gồm thêm mới, chỉnh sửa và xóa dữ liệu Truy vấn danh sách người dùng hiện có trong hệ thống dựa trên các tiêu chí như ID, tài khoản, hoặc vai trò	Các lớp cộng tác (Collaboration): Tài khoản

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) username(string) user_role_id(integer) date_of_birth(timestamps) user_phone_number(string) user_gender(integer)	profile_image(string) additional_infor(string) user_status_id(integer) createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Tài khoản, Vai trò người dùng, Trạng thái người dùng, Lịch khám, Thiết bị, Dữ liệu phiên đo	

2.1.6 Thẻ CRC lớp Loại thiết bị

Bảng 2.6 Thẻ CRC lớp Loại thiết bị

Mặt trước thẻ

Tên lớp: Loại thiết bị (device_type)	ID: 5
Mô tả: Lớp quản lý thông tin các loại thiết bị được sử dụng trong hệ thống	Use case liên quan: Quản lí thiết bị
Trách nhiệm (Responsibility): Lưu trữ và quản lý danh sách các loại thiết bị	Các lớp cộng tác (Collaboration): Thiết bị

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) type_name(string)	createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Thiết bị	

2.1.7 Thẻ CRC lớp Trạng thái thiết bị

Bảng 2.7 Thẻ CRC lớp Trạng thái thiết bị

Mặt trước thẻ

Tên lớp: Trạng thái thiết bị (device_status)	ID: 6
Mô tả: Lớp quản lý thông tin các trạng thái của thiết bị được sử dụng trong hệ thống	Use case liên quan: Quản lí thiết bị
Trách nhiệm (Responsibility): Lưu trữ danh sách các trạng thái của thiết bị (ví dụ: Hoạt động, Không hoạt động, Bảo trì) Cung cấp thông tin trạng thái cho các chức năng quản lý và phân công thiết bị	Các lớp cộng tác (Collaboration): Thiết bị

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) status_description(string)	createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Thiết bị	

2.1.8 Thẻ CRC lớp Thiết bị

Bảng 2.8 Thẻ CRC lớp Thiết bị

Mặt trước thẻ

Tên lớp: Thiết bị (device)	ID: 7
Mô tả: Lớp quản lý thông tin chi tiết các thiết bị được sử dụng trong hệ thống	Use case liên quan: Quản lí thiết bị
Trách nhiệm (Responsibility): Quản lý thông tin thiết bị: thêm mới, chỉnh sửa, và xóa dữ liệu thiết bị Truy vấn danh sách các thiết bị hiện có trong hệ thống dựa trên các tiêu chí như ID bệnh nhân và bác sĩ	Các lớp cộng tác (Collaboration): Thiết bị

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) user_id(string) device_name(string) device_information(string) device_type_id(integer)	device_status_id(integer) device_start_time(timestamps) device_end_time(timestamps) createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Dữ liệu phiên đo, Người dùng, Loại thiết bị, Trạng thái thiết bị, Thông tin chi tiết thiết bị	

2.1.9 Thẻ CRC lớp Thông số kỹ thuật

Bảng 2.9 Thẻ CRC lớp Thông số kỹ thuật

Mặt trước thẻ

Tên lớp: Thông số kỹ thuật (device_details)	ID: 8
Mô tả: Lớp quản lý thông tin chi tiết thông số kỹ thuật của các thiết bị được sử dụng trong hệ thống	Use case liên quan: Quản lí thiết bị
Trách nhiệm (Responsibility): Lưu trữ và quản lý thông số kỹ thuật của thiết bị: thêm mới, chỉnh sửa, và xóa thông số theo ID thiết bị Truy vấn các thông số kỹ thuật dựa trên ID thiết bị cụ thể	Các lớp cộng tác (Collaboration): Thiết bị

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) associated_device_id(string) device_detail_name(string) device_detail_type(integer)	detail_value(string) detail_information(string) createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Thiết bị	

2.1.10 Thẻ CRC lớp Dữ liệu phiên đo

Bảng 2.10 Thẻ CRC lớp Dữ liệu phiên đo

Mặt trước thẻ

Tên lớp: Dữ liệu phiên đo (records)	ID: 9
Mô tả: Lớp quản lý thông tin các Dữ liệu phiên đo của mỗi phiên đo	Use case liên quan: Quản lý dữ liệu phiên đo
Trách nhiệm (Responsibility): Lưu trữ và quản lý danh sách các dữ liệu phiên đo: chỉnh sửa, và xóa dữ liệu Truy vấn danh sách dữ liệu phiên đo dựa trên ID bệnh nhân hoặc ID bác sĩ	Các lớp cộng tác (Collaboration): Người dùng Thiết bị

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) patient_id(string) device_id(string) record_file_url(string)	season_start_time(string) season_end_time(string) createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Thiết bị, Người dùng	

2.1.11 Thẻ CRC lớp Trạng thái lịch khám

Bảng 2.11 Thẻ CRC lớp Trạng thái lịch khám

Mặt trước thẻ

Tên lớp: Trạng thái lịch khám (schedule_status)	ID: 10
Mô tả: Lớp quản lý thông tin các trạng thái lịch khám hiện tại trong hệ thống	Use case liên quan: Quản lý lịch khám
Trách nhiệm (Responsibility): Lưu trữ danh sách các trạng thái lịch khám (ví dụ: Thành công (được bác sĩ chấp nhận), Thất bại (bị từ chối), Đang chờ xác nhận) Cung cấp thông tin các trạng thái lịch khám để hỗ trợ các chức năng liên quan đến quản lý và hiển thị lịch khám	Các lớp cộng tác (Collaboration): Lịch khám

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) status_description(string)	createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Lịch khám	

2.1.12 Thẻ CRC lớp Kết quả lịch khám

Bảng 2.12 Thẻ CRC lớp Kết quả lịch khám

Mặt trước thẻ

Tên lớp: Kết quả lịch khám (schedule_result)	ID: 11
Mô tả: Lớp quản lý thông tin các kết quả của lịch khám	Use case liên quan: Quản lý lịch khám
Trách nhiệm (Responsibility): Lưu trữ danh sách các kết quả lịch khám có thể có (ví dụ: Lịch khám đã diễn ra, Lịch khám không diễn ra (Bệnh nhân không tới, các lí do khác)) Cung cấp thông tin các kết quả lịch khám cho các chức năng liên quan đến lập lịch	Các lớp cộng tác (Collaboration): Lịch khám

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) result_description(string)	createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Lịch khám	

2.1.13 Thẻ CRC lớp Lịch khám

Bảng 2.13 Thẻ CRC lớp Lịch khám

Mặt trước thẻ

Tên lớp: Lịch khám (schedule)	ID: 12
Mô tả: Lớp quản lý thông tin các lịch khám hiện tại trong hệ thống	Use case liên quan: Quản lý lịch khám
Trách nhiệm (Responsibility): Lưu trữ danh sách các lịch khám Truy vấn danh sách lịch khám dựa trên ID bệnh nhân Lập lịch khám dựa theo các tiêu chí như bác sĩ hoặc thời gian rảnh	Các lớp cộng tác (Collaboration): Trạng thái lịch khám Kết quả lịch khám Người dùng

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) patient_id(string) doctor_id(string) schedule_start_time(timestamps) schedule_end_time(timestamps)	schedule_status_id(integer) schedule_result_id(integer) createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Trạng thái lịch khám, Kết quả lịch khám, Người dùng, Chẩn đoán cho bệnh nhân	

2.1.14 Thẻ CRC lớp Thông báo liên quan đến lịch khám

Bảng 2.14 Thẻ CRC lớp Thông báo liên quan đến lịch khám

Mặt trước thẻ

Tên lớp: Thông báo liên quan đến lịch khám (notifications_schedule)	ID: 14
Mô tả: Lớp quản lý thông tin các thông báo liên quan đến lịch khám trong hệ thống	Use case liên quan: Quản lý lịch khám
Trách nhiệm (Responsibility): Lưu trữ danh sách thông báo với thông tin lịch khám, bác sĩ, và bệnh nhân tương ứng Truy vấn danh sách thông báo theo ID bác sĩ hoặc ID bệnh nhân	Các lớp cộng tác (Collaboration): Người dùng

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) patient_id(string) doctor_id(string) schedule_start_time(timestamps) status(integer)	reject_reason(string) type(integer) is_seen(boolean) createdAt(Datetime) updatedAt(Datetime)
----------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------

Mối quan hệ (Relationships)

Tổng quát hóa (Generalize):

Toàn thể - Bộ phận (Aggregation):

Liên kết (Association): Người dùng

2.1.15 Thẻ CRC lớp Chẩn đoán cho bệnh nhân**Bảng 2.15 Thẻ CRC lớp Chẩn đoán cho bệnh nhân**

Mặt trước thẻ

Tên lớp: Chẩn đoán cho bệnh nhân (diagnosis)	ID: 15
Mô tả: Lớp quản lý thông tin chi tiết các chẩn đoán cho bệnh nhân (nếu có) của mỗi lịch khám trong hệ thống	Use case liên quan: Quản lý lịch khám
Trách nhiệm (Responsibility): Lưu trữ danh sách chẩn đoán cho bệnh nhân với lịch khám tương ứng Truy vấn danh sách chẩn đoán theo ID lịch khám	Các lớp cộng tác (Collaboration): Lịch khám

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) schedule_id(string) diagnosis_infor(string)	createdAt(Datetime) updatedAt(Datetime)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Lịch khám	

2.1.16 Thẻ CRC lớp Tin nhắn**Bảng 2.16 Thẻ CRC lớp Tin nhắn**

Mặt trước thẻ

Tên lớp: Tin nhắn (messages)	ID: 16
Mô tả: Lớp quản lý thông tin chi tiết các tin nhắn có trong hệ thống	Use case liên quan: Quản lý dịch vụ nhắn tin
Trách nhiệm (Responsibility): Lưu trữ danh sách tin nhắn được trao đổi giữa hai người hoặc trong các nhóm trò chuyện Truy vấn danh sách tin nhắn dựa trên ID nhóm hoặc ID cá nhân liên quan Cung cấp thông tin người gửi, nội dung tin nhắn và thời gian gửi	Các lớp cộng tác (Collaboration): Người dùng

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) senderId(string) groupChatId(string)	message(string) time(timestamps)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Người dùng	

2.1.17 Thẻ CRC lớp Nhóm trò chuyện

Bảng 2.17 Thẻ CRC lớp Nhóm trò chuyện

Mặt trước thẻ

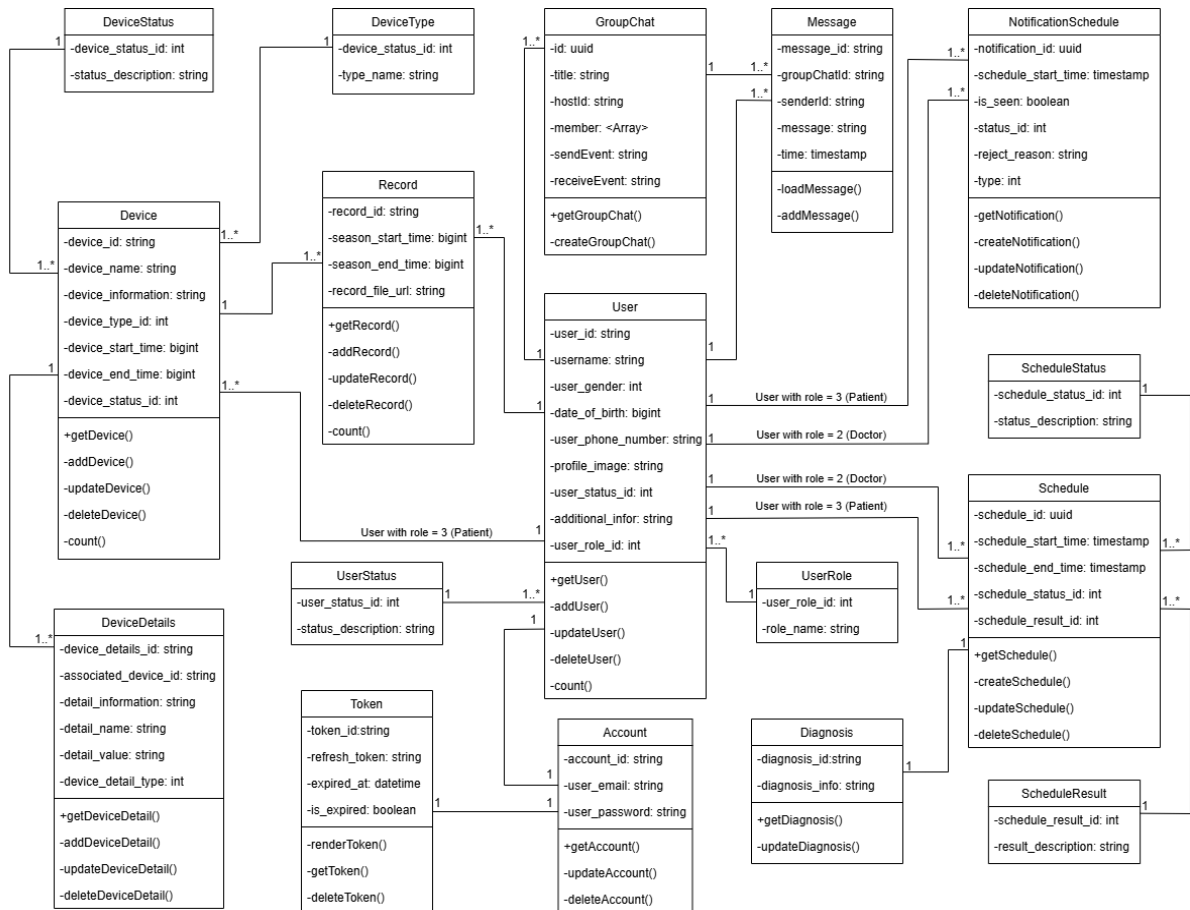
Tên lớp: Nhóm trò chuyện (group_chat)	ID: 17
Mô tả: Lớp quản lý thông tin chi tiết các nhóm trò chuyện có trong hệ thống	Use case liên quan: Quản lý dịch vụ nhắn tin
Trách nhiệm (Responsibility): Lưu trữ danh sách các nhóm trò chuyện trong hệ thống Truy vấn danh sách các nhóm trò chuyện dựa trên ID nhóm hoặc ID cá nhân liên quan Cung cấp thông tin người gửi, nội dung tin nhắn và thời gian gửi	Các lớp cộng tác (Collaboration): Người dùng, Tin nhắn

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) title(string) hostId(string)	member(array) sendEvent(string) receiveEvent(string)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Người dùng, Tin nhắn	

2.2 Sơ đồ lớp

Dựa trên các thẻ CRC đã được mô tả ở phần trước, chúng em xin trình bày sơ đồ lớp của hệ thống.



Hình 2.1 Sơ đồ UML

Hình 2.1 thể hiện các lớp trong hệ thống:

- Lớp Account (Tài khoản): Chịu trách nhiệm quản lý thông tin cơ bản của tài khoản trong hệ thống. Cung cấp các phương thức để thực hiện thao tác tạo mới, xác thực và quản lý dữ liệu liên quan đến tài khoản người dùng.
- Lớp Token: Đảm nhiệm việc lưu trữ và xử lý thông tin về các token sử dụng trong quá trình đăng nhập, đăng ký và xác thực người dùng, đảm bảo tính bảo mật và hiệu quả trong giao tiếp.
- Lớp UserRole (Vai trò người dùng): Mô tả và quản lý các vai trò mà người dùng có thể đảm nhiệm, hỗ trợ trong việc phân quyền và xác định hành vi của từng loại vai trò.
- Lớp UserStatus (Trạng thái người dùng): Theo dõi và duy trì trạng thái hiện tại của người dùng trong hệ thống, ví dụ như trạng thái hoạt động, bị khóa hoặc đang chờ xác minh.
- Lớp User (Người dùng): Lưu trữ thông tin chi tiết của người dùng sau khi đăng ký, cung cấp các phương thức xử lý liên quan đến thông tin cá nhân và các hoạt động

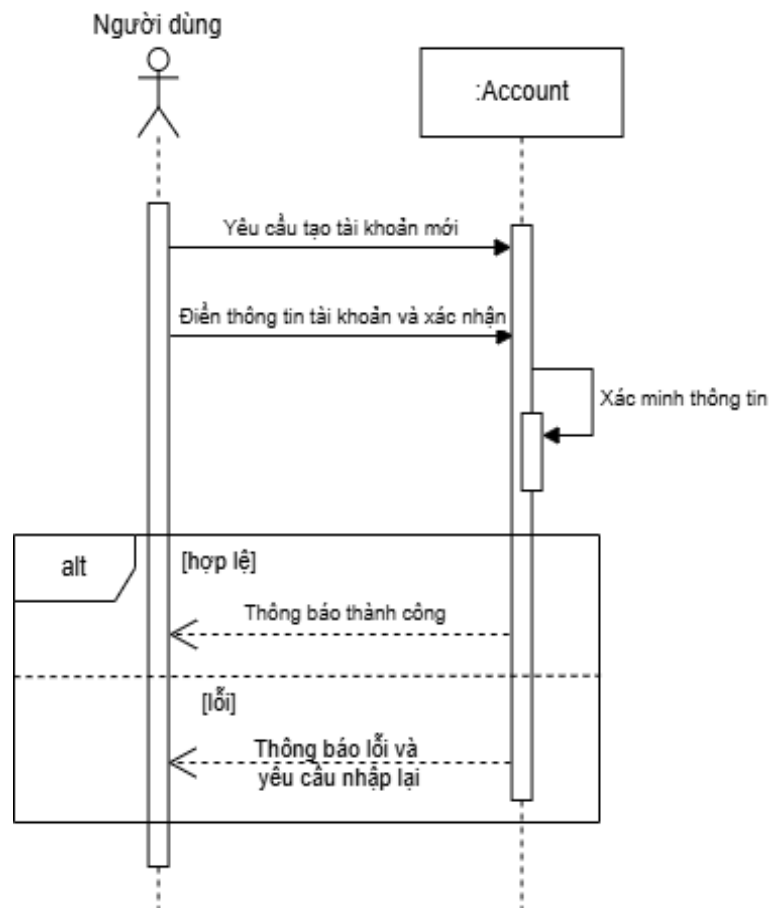
của người dùng trong hệ thống.

- Lớp DeviceType (Loại thiết bị): Định nghĩa các loại thiết bị khác nhau mà hệ thống hỗ trợ, giúp tổ chức và phân loại thông tin thiết bị một cách khoa học.
- Lớp DeviceStatus (Trạng thái thiết bị): Quản lý tình trạng hoạt động của thiết bị, cho phép theo dõi và kiểm soát thiết bị một cách hiệu quả trong hệ thống.
- Lớp Device (Thiết bị): Chứa các thông tin cơ bản về thiết bị và người dùng thiết bị, hỗ trợ việc quản lý, tương tác và thu thập dữ liệu từ thiết bị trong hệ thống.
- Lớp DeviceDetail (Thông số kỹ thuật): Cung cấp các thông tin kỹ thuật chi tiết về thiết bị, hỗ trợ trong việc cấu hình, kiểm tra và quản lý hiệu năng của thiết bị.
- Lớp Record (Dữ liệu phiên đo): Được thiết kế để lưu trữ thông tin về dữ liệu mỗi phiên đo của bệnh nhân, đảm bảo quản lý dữ liệu một cách hiệu quả, an toàn và hỗ trợ xử lý nhanh chóng, chính xác.
- Lớp ScheduleStatus (Trạng thái lịch khám): Quản lý trạng thái của lịch khám (bao gồm các trạng thái như đang chờ xác nhận, đã được bác sĩ chấp nhận, hoặc chưa được xác nhận), giúp theo dõi và cập nhật trạng thái trong suốt vòng đời của lịch khám.
- Lớp ScheduleResult (Kết quả lịch khám): Xác định các kết quả lịch khám trong hệ thống, phục vụ việc phân loại và quản lý lịch khám.
- Lớp Schedule (Lịch khám): Đại diện cho cấu trúc thông tin của một lịch khám, cung cấp các phương thức để lưu trữ, truy vấn và xử lý dữ liệu lịch khám.
- Lớp NotificationSchedule (Thông báo liên quan đến lịch khám): Chịu trách nhiệm tạo và quản lý các thông báo liên quan đến lịch khám, giúp người dùng và bác sĩ cập nhật tình trạng lịch khám kịp thời.
- Lớp Diagnosis (Chẩn đoán cho bệnh nhân): Quản lý thông tin chẩn đoán bệnh của bệnh nhân (nếu có).
- Lớp Message (Tin nhắn): Xử lý thông tin các tin nhắn được gửi trong hệ thống, cho phép người dùng trao đổi thông tin trong các cuộc trò chuyện cá nhân hoặc nhóm.
- Lớp GroupChat (Nhóm trò chuyện): Quản lý các nhóm trò chuyện trong hệ thống, cho phép tổ chức và thực hiện các hoạt động giao tiếp nhóm một cách hiệu quả.

2.3 Sơ đồ tuần tự

Để làm rõ từng luồng hoạt động, các sơ đồ tuần tự dưới đây được xây dựng dựa trên các sơ đồ trường hợp sử dụng và sơ đồ lớp, nhằm mô tả chi tiết quá trình tương tác giữa các thành phần trong toàn bộ quy trình.

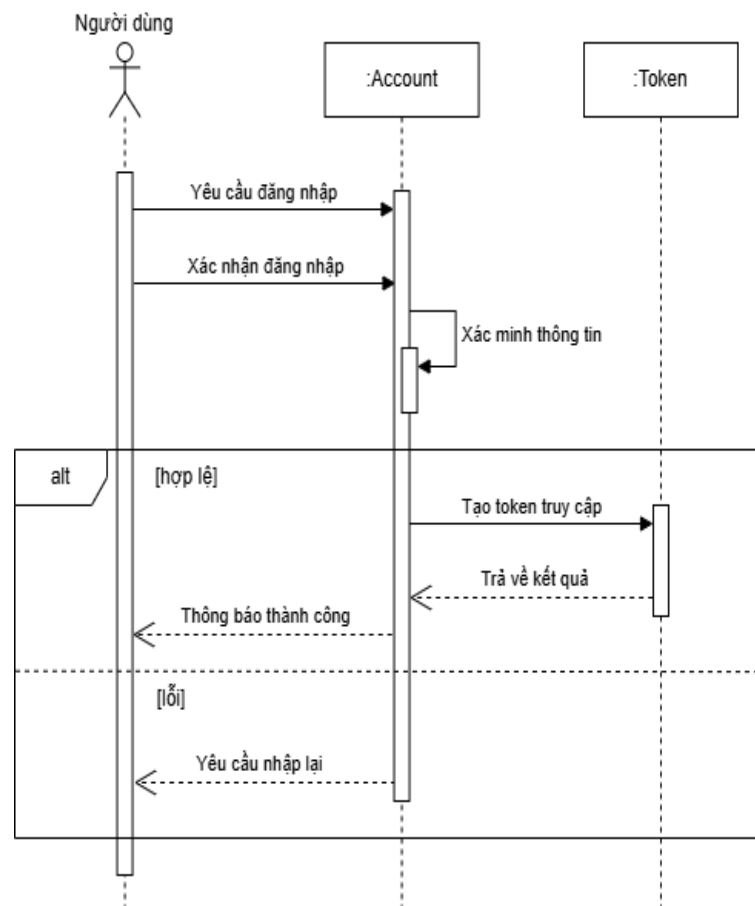
2.3.1 Sơ đồ tuần tự chức năng tạo tài khoản người dùng



Hình 2.2 Sơ đồ tuần tự chức năng tạo tài khoản người dùng

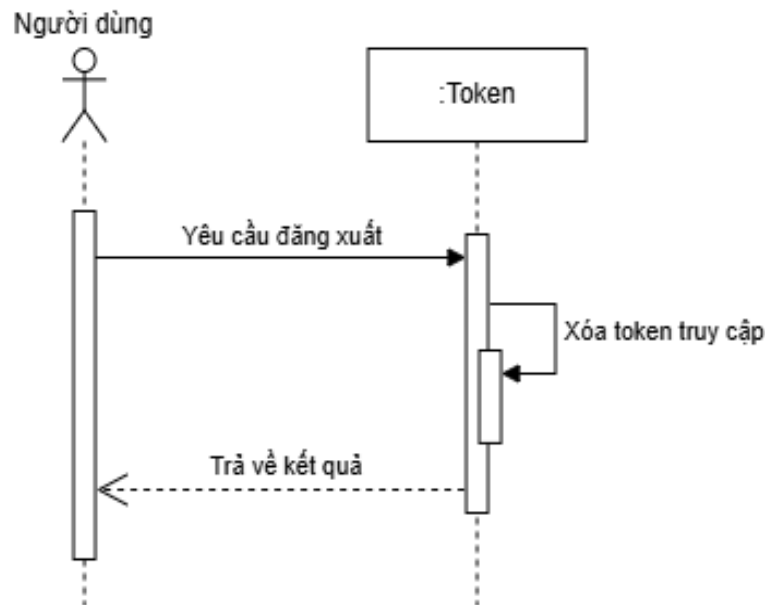
Sơ đồ tuần tự minh họa quá trình tạo tài khoản mới. Người dùng nhập các thông tin cần thiết vào biểu mẫu đăng ký và gửi yêu cầu, sau đó hệ thống chuyển yêu cầu đến lớp Account để xử lý. Trong trường hợp có lỗi, hệ thống sẽ phản hồi và hiển thị thông báo lỗi cho người dùng. Nếu quá trình đăng ký hoàn tất, lớp Account sẽ xác nhận và chuyển hướng người dùng đến trang đăng nhập.

2.3.2 Sơ đồ tuần tự chức năng đăng nhập và đăng xuất khỏi hệ thống



Hình 2.3 Sơ đồ tuần tự chức năng đăng nhập vào hệ thống

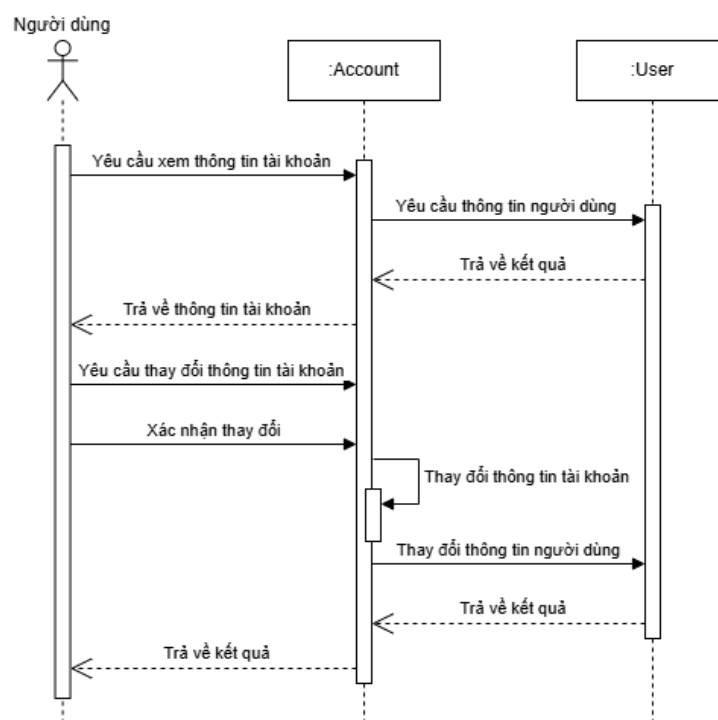
Sơ đồ tuần tự minh họa chi tiết quy trình đăng nhập vào hệ thống. Người dùng nhập các thông tin cần thiết vào biểu mẫu đăng nhập và gửi yêu cầu. Hệ thống sẽ chuyển yêu cầu đến lớp Account để xử lý. Nếu có lỗi xảy ra, hệ thống sẽ gửi thông báo lỗi để người dùng điều chỉnh. Ngược lại, khi đăng nhập thành công, lớp Account sẽ tạo token truy cập mới dựa trên lớp Token, gửi thông báo xác nhận và chuyển hướng người dùng tới trang chính của hệ thống.



Hình 2.4 Sơ đồ tuần tự chức năng đăng xuất khỏi hệ thống

Sơ đồ tuần tự minh họa chi tiết quy trình đăng xuất khỏi hệ thống. Người dùng gửi yêu cầu đăng xuất, hệ thống chuyển yêu cầu này đến lớp Token để xử lý. Lớp Token sẽ xóa thông tin token truy cập liên quan và gửi phản hồi xác nhận thành công, hoàn tất quá trình đăng xuất.

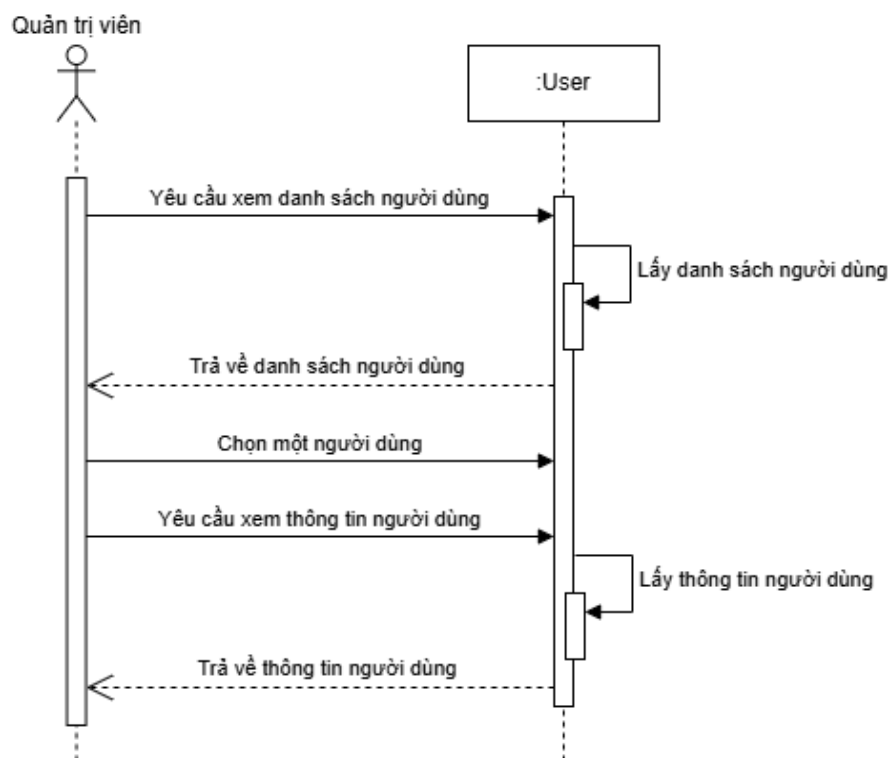
2.3.3 Sơ đồ tuần tự chức năng quản lý tài khoản người dùng



Hình 2.5 Sơ đồ tuần tự chức năng quản lý tài khoản người dùng

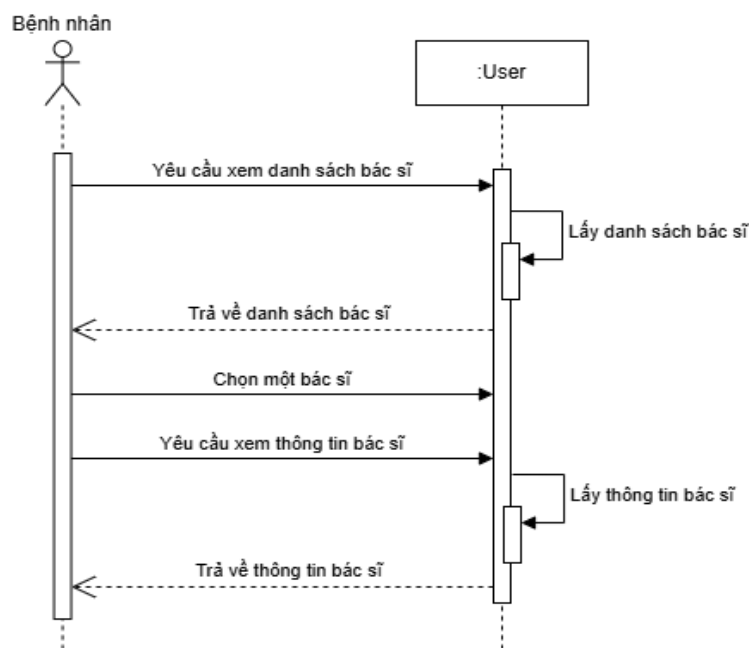
Sơ đồ tuần tự minh họa chi tiết quy trình quản lý tài khoản người dùng. Khi nhận yêu cầu xem thông tin, hệ thống chuyển yêu cầu đến lớp Account và lớp User để xử lý, sau đó trả về thông tin chi tiết tài khoản. Nếu người dùng yêu cầu thay đổi thông tin, hệ thống tiếp nhận và xử lý thông qua lớp Account. Trong trường hợp thông tin cần thay đổi liên quan đến lớp User, lớp Account sẽ gọi lớp User để thực hiện cập nhật. Khi quá trình thay đổi hoàn tất, hệ thống gửi thông báo xác nhận và kết quả đến người dùng, đảm bảo mọi thông tin được cập nhật chính xác.

2.3.4 Sơ đồ tuần tự chức năng quản lý người dùng hệ thống



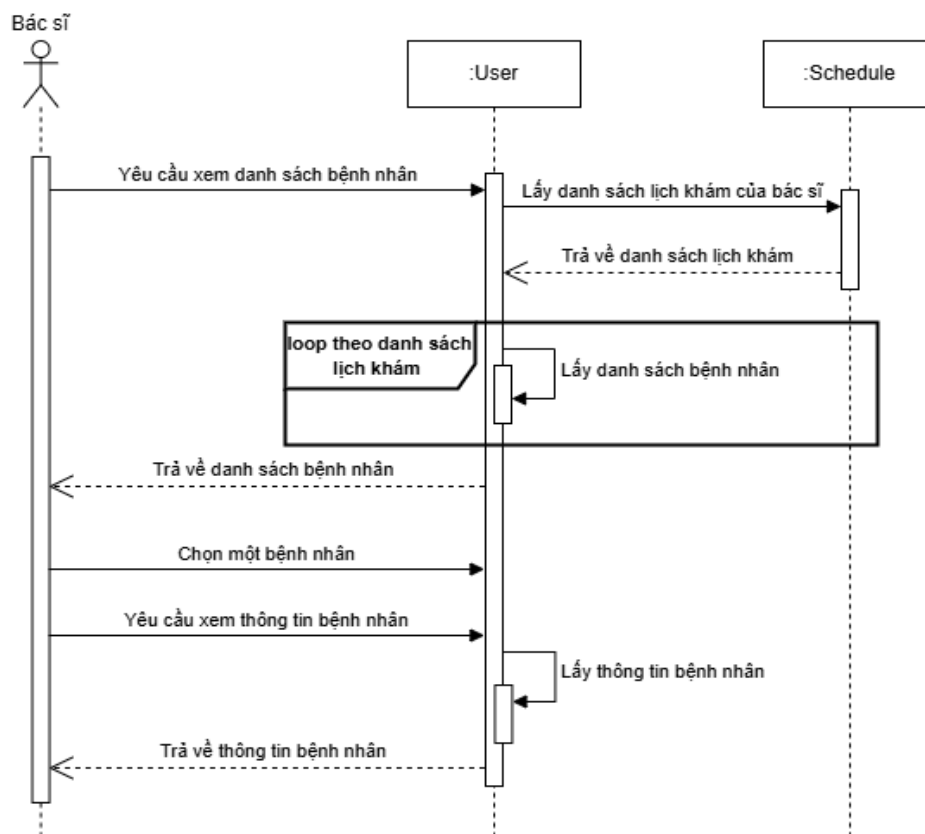
Hình 2.6 Sơ đồ tuần tự chức năng tra cứu danh sách người dùng

Sơ đồ tuần tự minh họa chi tiết quy trình tra cứu danh sách và thông tin người dùng. Khi quản trị viên gửi yêu cầu xem danh sách, hệ thống chuyển yêu cầu đến lớp User để xử lý và truy xuất danh sách từ cơ sở dữ liệu, sau đó trả về kết quả cho quản trị viên. Quản trị viên có thể tùy chọn xem thêm thông tin chi tiết của một người dùng cụ thể. Trong trường hợp này, hệ thống tiếp tục chuyển yêu cầu đến lớp User để lấy dữ liệu chi tiết từ cơ sở dữ liệu và trả kết quả lại. Quy trình đảm bảo thông tin được truy xuất chính xác, nhanh chóng, hỗ trợ quản trị viên quản lý hiệu quả hệ thống.



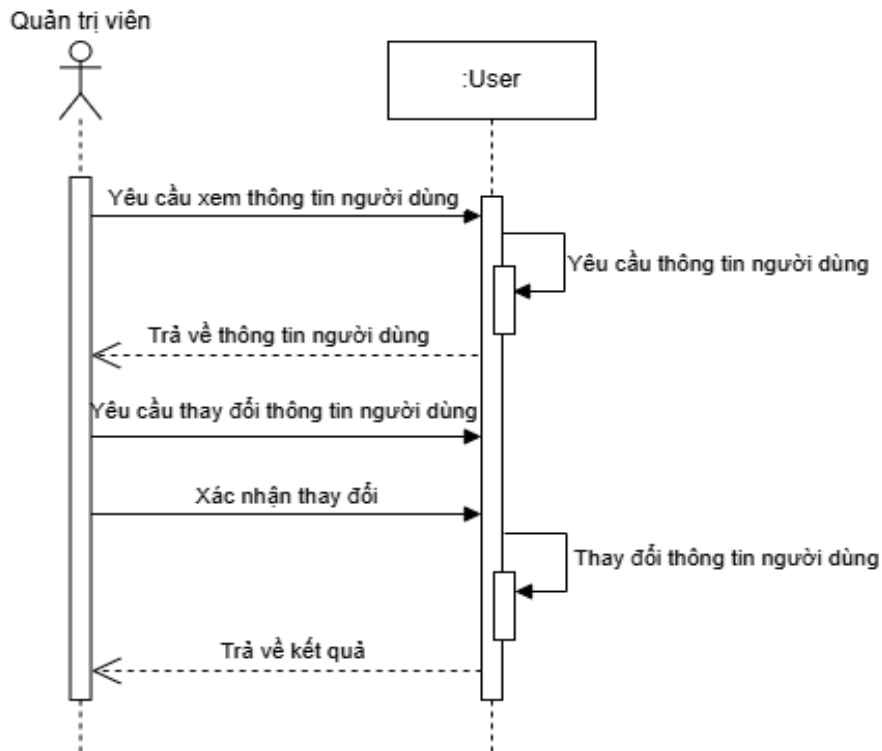
Hình 2.7 Sơ đồ tuần tự chức năng tra cứu danh sách bác sĩ

Tương tự quy trình tra cứu danh sách và thông tin người dùng đã đề cập ở trên, sơ đồ tuần tự này minh họa chi tiết quá trình truy xuất danh sách và thông tin chi tiết của các bác sĩ trong hệ thống, hỗ trợ bệnh nhân thuận tiện trong việc tìm kiếm và lựa chọn bác sĩ phù hợp.



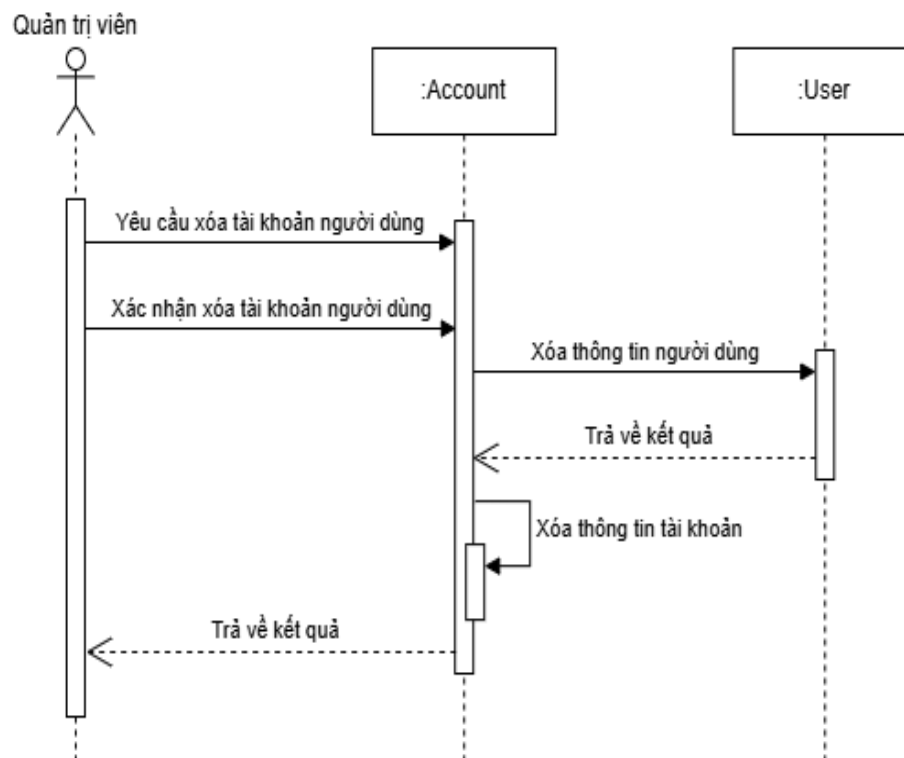
Hình 2.8 Sơ đồ tuần tự chức năng tra cứu danh sách bệnh nhân đang theo dõi

Sơ đồ tuần tự này tương tự quy trình tra cứu danh sách và thông tin người dùng, nhưng được áp dụng cho việc truy xuất danh sách bệnh nhân mà bác sĩ đang theo dõi. Hệ thống sẽ lấy danh sách lịch hẹn thông qua lớp Schedule, sau đó truy xuất danh sách bệnh nhân từ lớp User. Bác sĩ có thể chọn một bệnh nhân cụ thể để xem thông tin chi tiết, với dữ liệu được xử lý và cung cấp từ lớp User, đảm bảo tính chính xác và hỗ trợ hiệu quả trong việc quản lý, theo dõi và điều trị bệnh nhân.



Hình 2.9 Sơ đồ tuần tự chức năng sửa thông tin người dùng

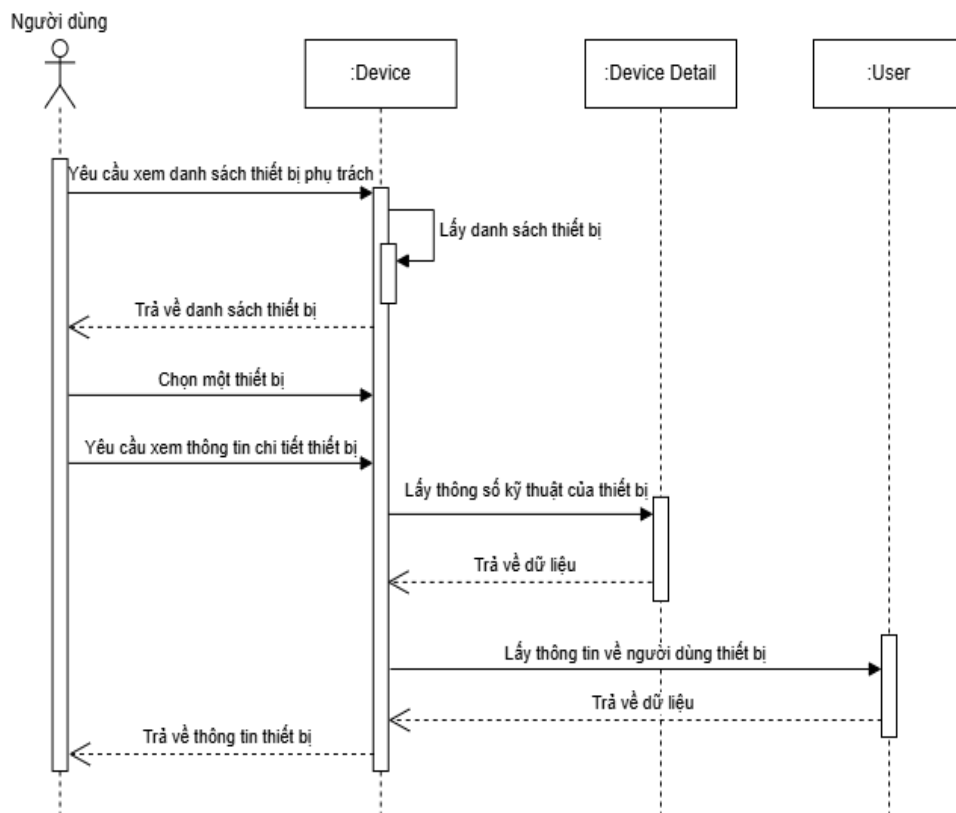
Sơ đồ tuần tự này minh họa chi tiết quy trình quản lý việc sửa thông tin người dùng bởi quản trị viên. Khi quản trị viên gửi yêu cầu xem thông tin người dùng, hệ thống chuyển yêu cầu đến lớp User để xử lý và trả về thông tin chi tiết. Nếu quản trị viên yêu cầu thay đổi thông tin, hệ thống tiếp nhận và gửi yêu cầu đến lớp User để cập nhật thông tin cần thiết. Sau khi hoàn tất quá trình thay đổi, hệ thống trả về kết quả xác nhận, đảm bảo mọi thông tin được cập nhật chính xác và nhanh chóng.



Hình 2.10 Sơ đồ tuần tự chức năng xóa người dùng khỏi hệ thống

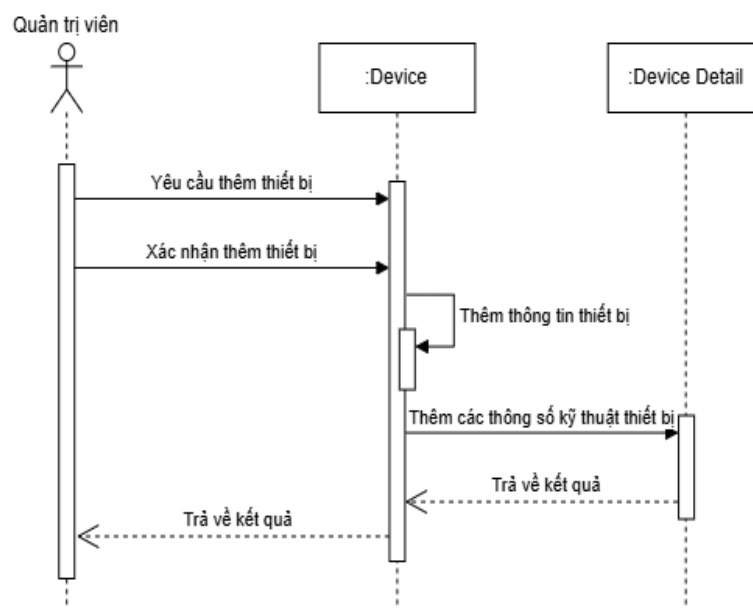
Sơ đồ tuần tự này minh họa quy trình xóa người dùng khỏi hệ thống. Đầu tiên, quản trị viên gửi yêu cầu xóa tài khoản người dùng. Hệ thống tiếp nhận và thực hiện xóa thông tin người dùng thông qua lớp User. Sau đó, hệ thống tiếp tục xóa thông tin tài khoản tương ứng thông qua lớp Account. Cuối cùng, hệ thống trả về kết quả xử lý để thông báo cho quản trị viên về trạng thái hoàn tất của quy trình.

2.3.5 Sơ đồ tuần tự chức năng quản lý thiết bị y tế



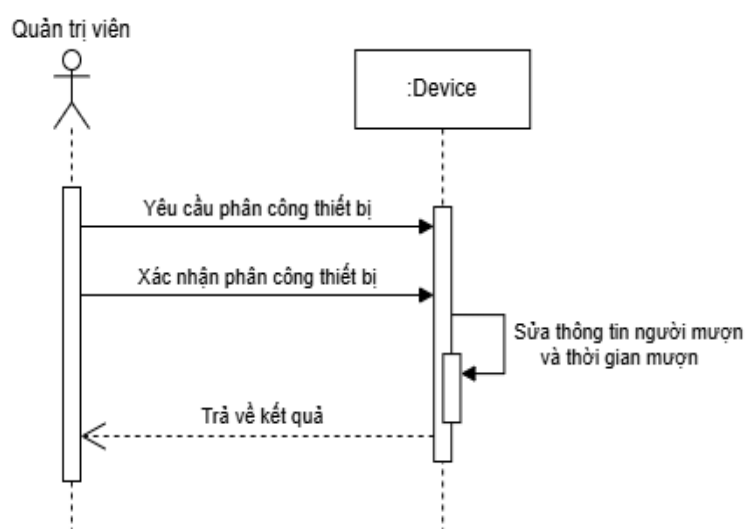
Hình 2.11 Sơ đồ tuần tự chức năng tra cứu danh sách thiết bị y tế

Sơ đồ tuần tự này minh họa quy trình quản lý thiết bị y tế. Người dùng gửi yêu cầu xem danh sách các thiết bị mà họ được phân công phụ trách. Hệ thống tiếp nhận và chuyển yêu cầu đến lớp Device để truy xuất danh sách thiết bị từ cơ sở dữ liệu, sau đó trả kết quả về cho người dùng. Khi người dùng chọn một thiết bị cụ thể và yêu cầu xem thông tin chi tiết, hệ thống sẽ lần lượt truy xuất thông số kỹ thuật từ lớp Device Detail và thông tin người sử dụng thiết bị từ lớp User. Toàn bộ dữ liệu thu thập được sẽ được hệ thống trả về, đảm bảo cung cấp đầy đủ thông tin hỗ trợ hiệu quả cho quá trình quản lý thiết bị y tế.



Hình 2.12 Sơ đồ tuần tự chức năng thêm mới thiết bị y tế

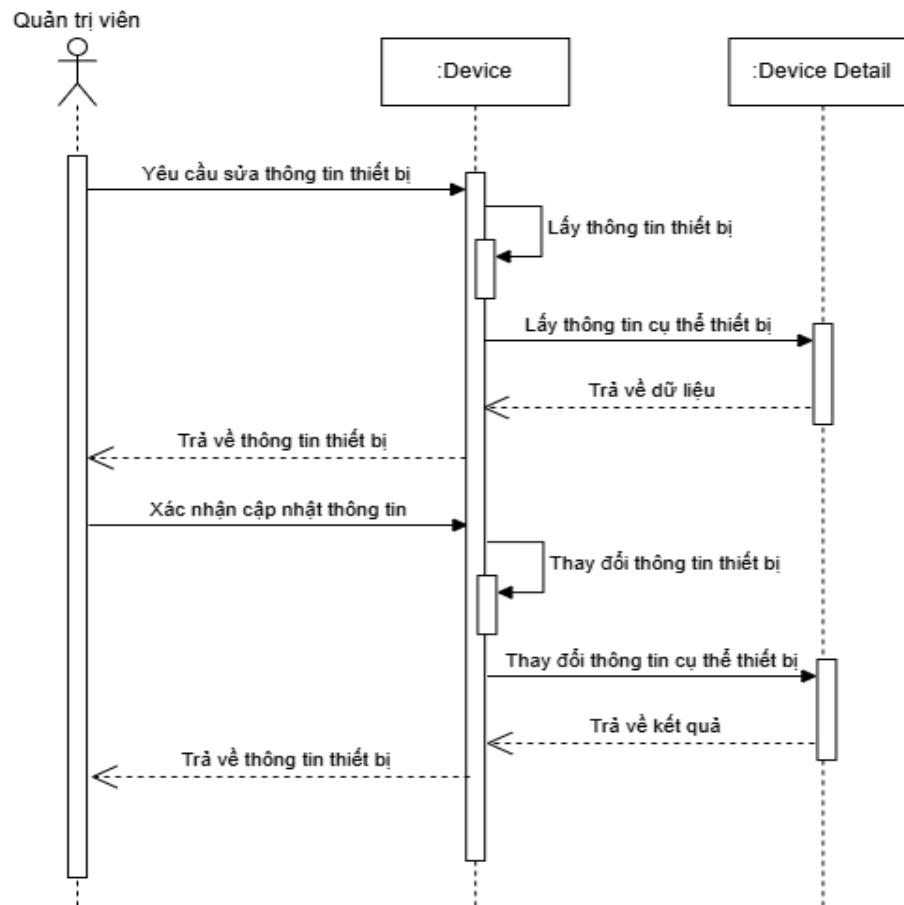
Sơ đồ tuần tự này minh họa quy trình thêm mới thiết bị y tế. Quản trị viên gửi yêu cầu thêm thiết bị, hệ thống tiếp nhận yêu cầu và chuyển đến lớp Device để xác nhận. Sau khi xác nhận, hệ thống thêm thông tin thiết bị vào cơ sở dữ liệu thông qua lớp Device. Đồng thời, nếu thiết bị có các thông số kỹ thuật kèm theo, lớp Device Detail sẽ được gọi để lưu trữ chi tiết thông số kỹ thuật của thiết bị. Cuối cùng, hệ thống trả về kết quả xác nhận việc thêm mới thiết bị.



Hình 2.13 Sơ đồ tuần tự chức năng phân công thiết bị y tế cho người dùng

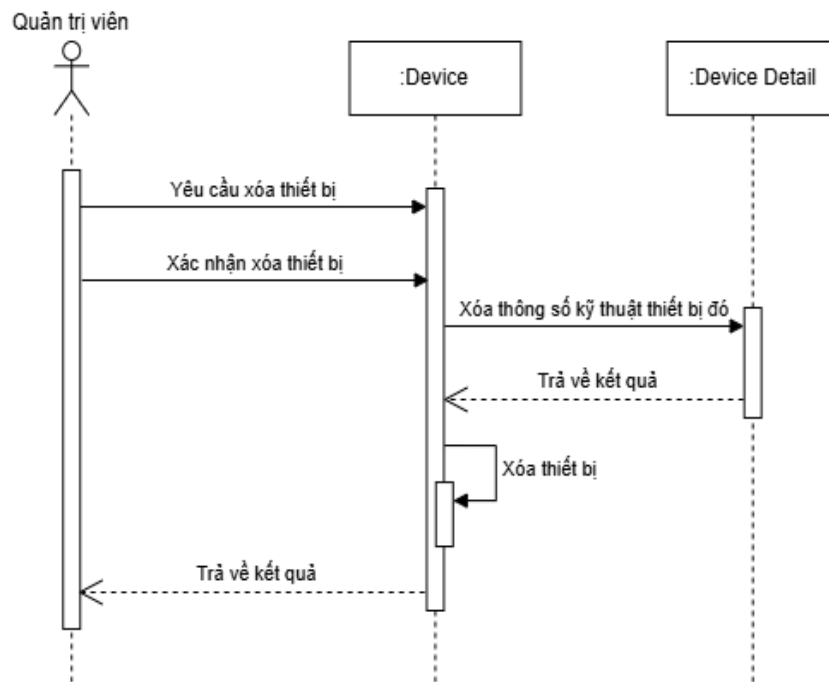
Sơ đồ tuần tự trên minh họa chi tiết quy trình phân công thiết bị y tế cho người dùng. Quản trị viên gửi yêu cầu phân công thiết bị, hệ thống tiếp nhận và chuyển đến lớp Device để xử lý. Lớp Device thực hiện việc cập nhật thông tin người dùng được mượn thiết bị và thời gian mượn vào cơ sở dữ liệu. Sau khi hoàn tất, hệ thống trả về kết

quả xác nhận việc phân công thiết bị, đảm bảo thiết bị được quản lý và phân bổ chính xác đến đúng người dùng.



Hình 2.14 Sơ đồ tuần tự chức năng chỉnh sửa thông tin thiết bị y tế

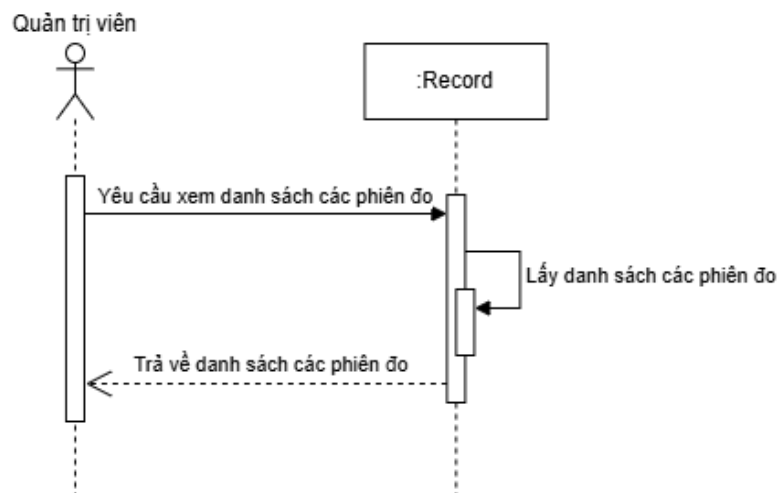
Sơ đồ tuần tự trên minh họa chi tiết quy trình sửa đổi thông tin thiết bị. Quản trị viên gửi yêu cầu sửa thông tin thiết bị, hệ thống chuyển yêu cầu đến lớp Device để truy xuất thông tin hiện tại từ lớp Device Detail. Sau khi nhận được dữ liệu, quản trị viên xác nhận thay đổi thông tin. Hệ thống xử lý việc cập nhật thông tin thiết bị trong lớp Device và thực hiện điều chỉnh thông số kỹ thuật trong lớp Device Detail nếu cần thiết. Khi hoàn tất, hệ thống trả về kết quả xác nhận, đảm bảo thông tin thiết bị được cập nhật đầy đủ và chính xác.



Hình 2.15 Sơ đồ tuần tự chức năng xóa thiết bị y tế

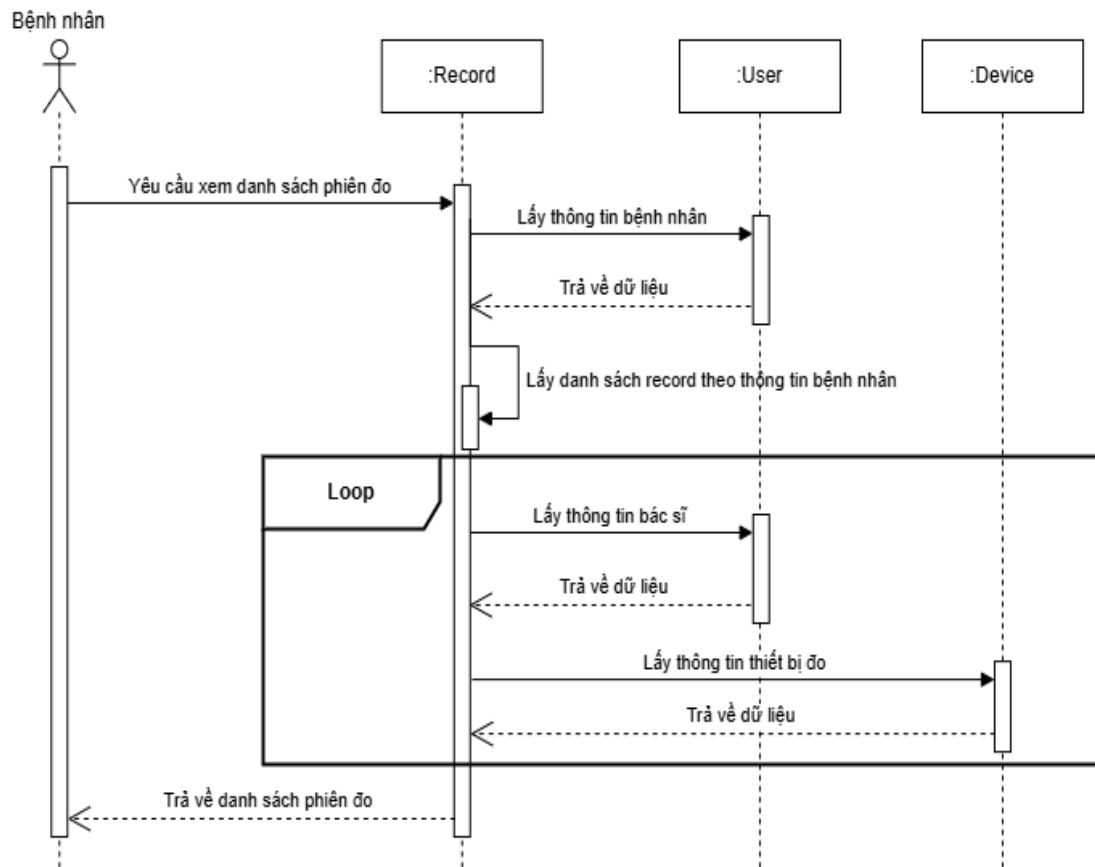
Sơ đồ tuần tự trên minh họa quy trình xóa thiết bị y tế khỏi hệ thống. Quản trị viên gửi yêu cầu xóa thiết bị, hệ thống chuyển yêu cầu đến lớp Device để xác nhận việc xóa. Sau khi xác nhận, hệ thống tiến hành xóa thông số kỹ thuật liên quan đến thiết bị trong lớp Device Detail. Sau đó, hệ thống thực hiện xóa thiết bị trong lớp Device. Khi quá trình xóa hoàn tất, hệ thống trả về kết quả, đảm bảo thiết bị và các thông tin liên quan được xóa bỏ hoàn toàn.

2.3.6 Sơ đồ tuần tự chức năng quản lý dữ liệu phiên đo



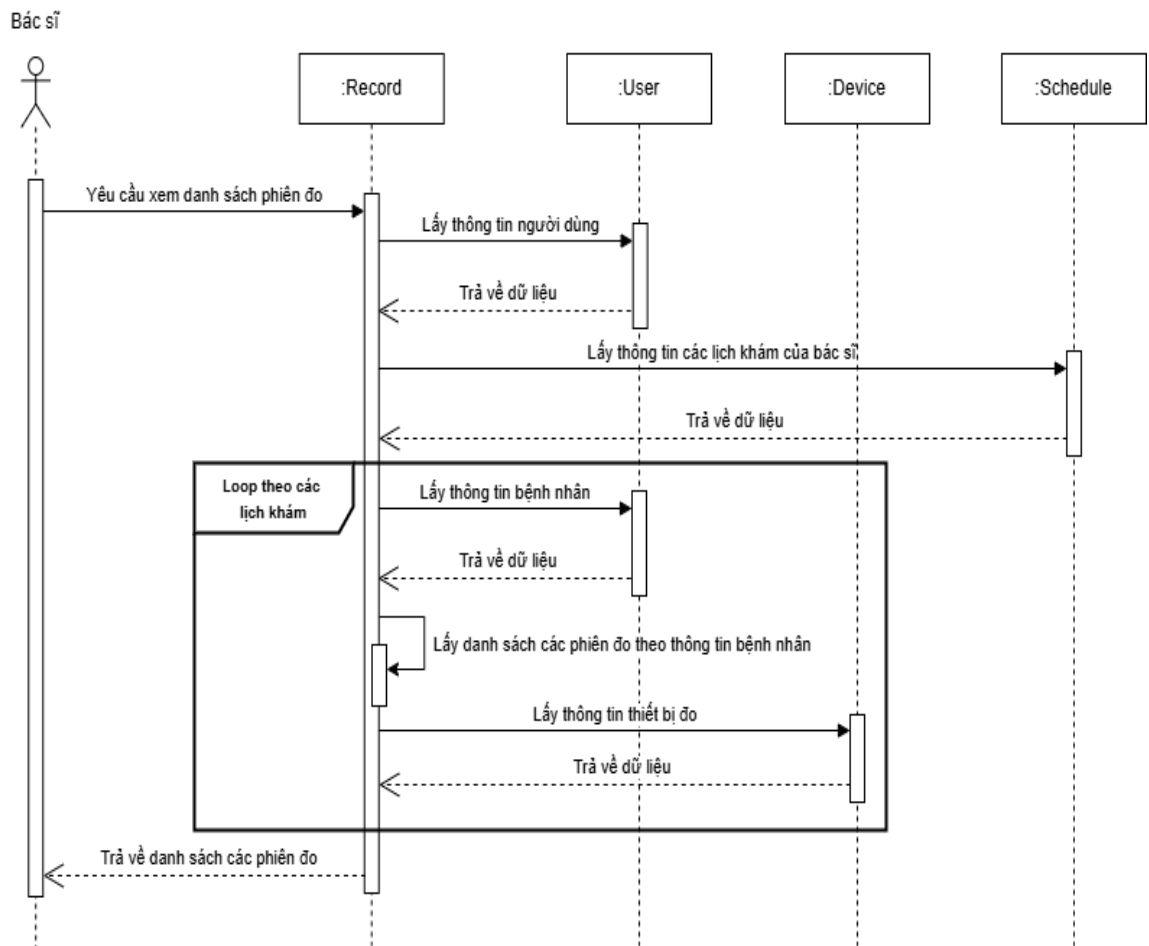
Hình 2.16 Sơ đồ tuần tự chức năng tra cứu danh sách toàn bộ phiên đo

Sơ đồ tuần tự trên minh họa quy trình tra cứu danh sách toàn bộ phiên đo trong hệ thống. Quản trị viên gửi yêu cầu xem danh sách các phiên đo, hệ thống chuyển yêu cầu đến lớp Record để truy xuất dữ liệu từ cơ sở dữ liệu. Sau khi hoàn tất, hệ thống trả về danh sách các phiên đo cho quản trị viên.



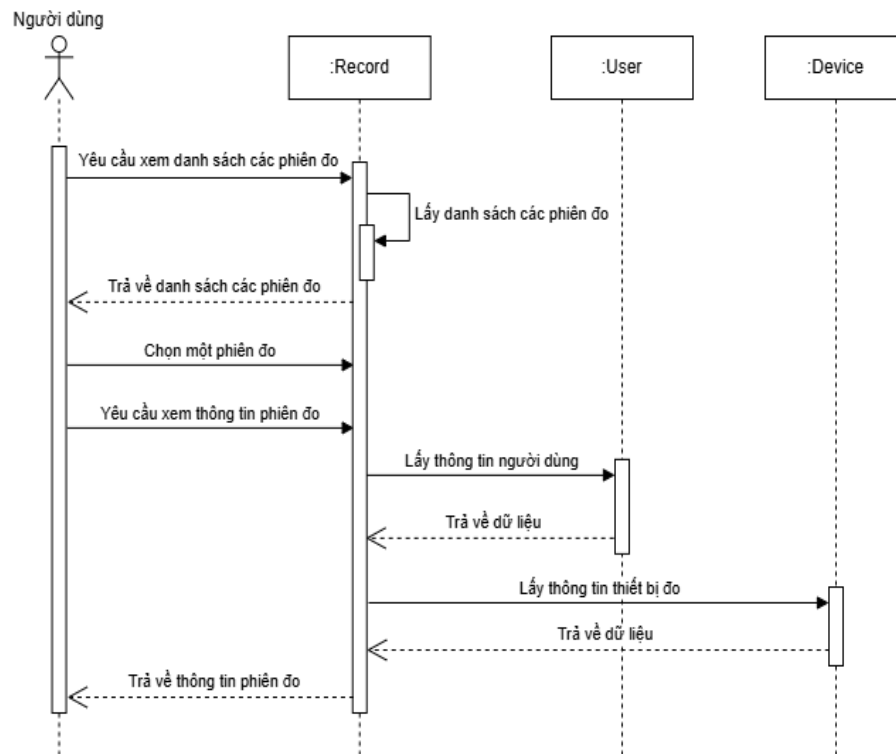
Hình 2.17 Sơ đồ tuần tự chức năng tra cứu danh sách các phiên đo cá nhân

Sơ đồ tuần tự trên minh họa quy trình tra cứu danh sách các phiên đo cá nhân của bệnh nhân. Khi bệnh nhân gửi yêu cầu, hệ thống sẽ lấy thông tin bệnh nhân từ lớp User và truy xuất danh sách các phiên đo liên quan từ lớp Record. Trong quá trình xử lý, hệ thống lặp qua danh sách phiên đo để lấy thông tin bác sĩ và thiết bị đo tương ứng từ lớp User và lớp Device. Sau khi tổng hợp dữ liệu, hệ thống trả về danh sách phiên đo chi tiết cho bệnh nhân.



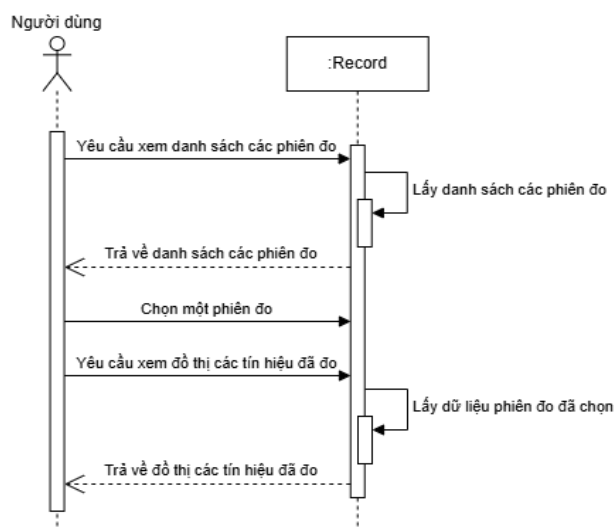
Hình 2.18 Sơ đồ tuần tự chức năng tra cứu danh sách các phiên đo của bệnh nhân đang theo dõi

Sơ đồ tuần tự này tương tự quy trình tra cứu danh sách các phiên đo cá nhân của bệnh nhân, nhưng áp dụng cho bác sĩ theo dõi các bệnh nhân mà mình phụ trách. Điểm khác biệt chính nằm ở việc hệ thống phải lấy danh sách lịch khám của bác sĩ từ lớp Schedule, sau đó lặp qua từng lịch khám để lấy thông tin bệnh nhân từ lớp User. Tiếp theo, hệ thống truy xuất danh sách phiên đo của từng bệnh nhân từ lớp Record và bổ sung thông tin thiết bị từ lớp Device, sau đó tổng hợp dữ liệu và trả về danh sách chi tiết các phiên đo cho bác sĩ.



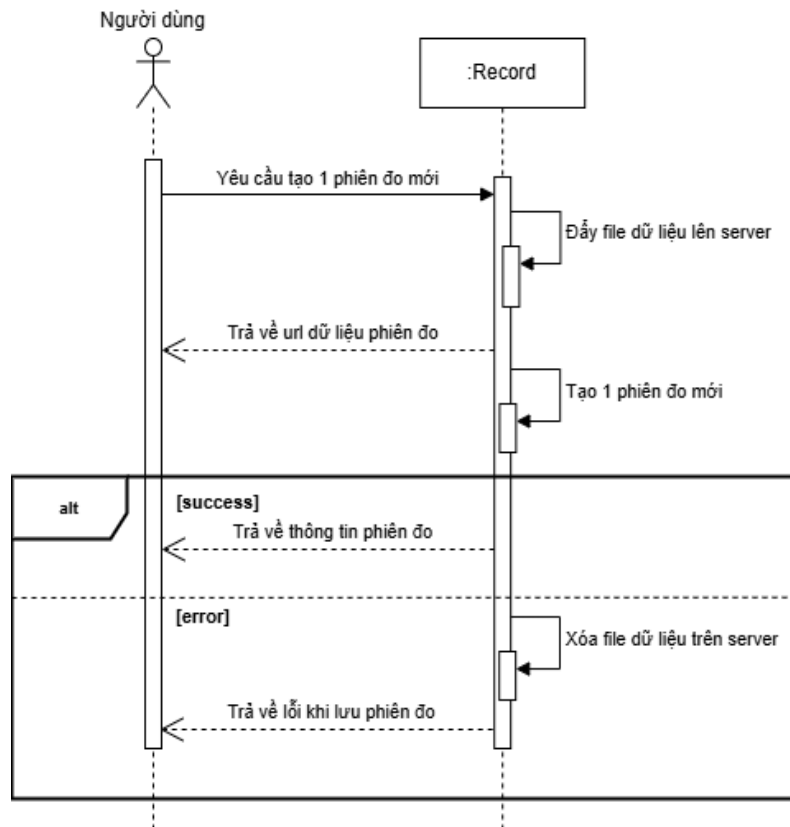
Hình 2.19 Sơ đồ tuần tự chức năng tra cứu thông tin chi tiết một phiên đo

Sơ đồ tuần tự này minh họa quy trình người dùng tra cứu thông tin chi tiết của một phiên đo cụ thể. Người dùng trước tiên yêu cầu xem danh sách các phiên đo, hệ thống truy xuất danh sách từ lớp Record và trả về kết quả. Sau đó, người dùng chọn một phiên đo cụ thể và gửi yêu cầu tra cứu thông tin chi tiết. Hệ thống tiếp tục truy vấn thông tin liên quan từ lớp User (để lấy thông tin người dùng) và lớp Device (để lấy thông tin thiết bị đo). Cuối cùng, toàn bộ thông tin chi tiết của phiên đo được trả về cho người dùng.



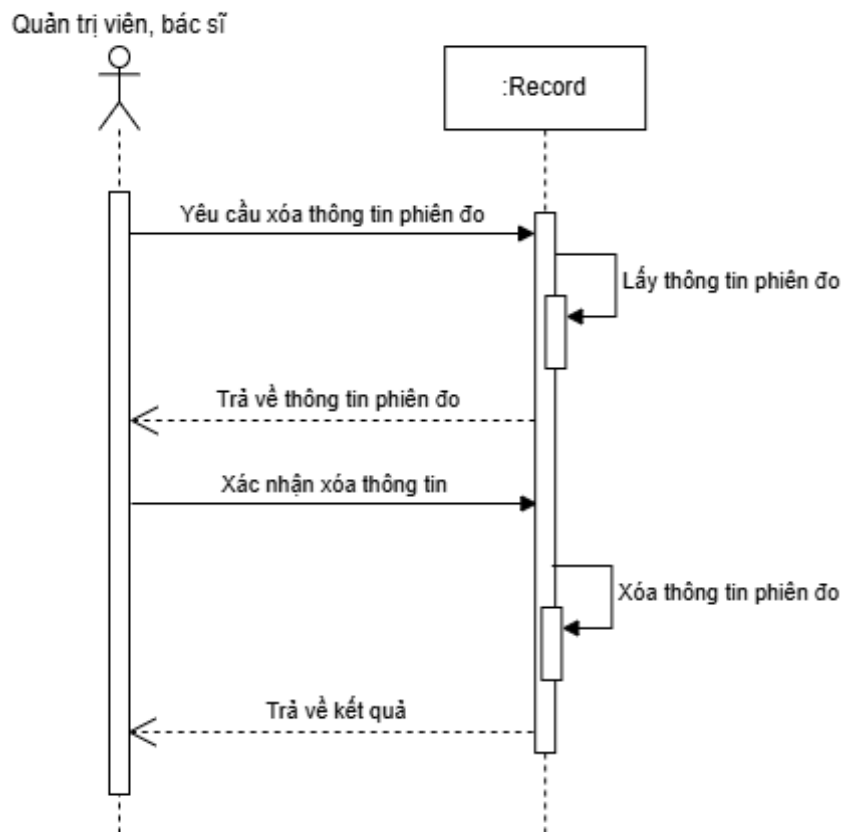
Hình 2.20 Sơ đồ tuần tự chức năng xem đồ thị các tín hiệu phiên đo

Sau khi danh sách phiên đo được truy xuất và người dùng đã lựa chọn một phiên đo cụ thể, họ có thể gửi yêu cầu để xem đồ thị tín hiệu. Hệ thống sẽ tiếp nhận yêu cầu, truy xuất dữ liệu các tín hiệu của phiên đo từ lớp Record, và trả về kết quả dưới dạng đồ thị trực quan, hỗ trợ người dùng theo dõi và phân tích một cách hiệu quả.



Hình 2.21 Sơ đồ tuần tự chức năng thêm phiên đo mới

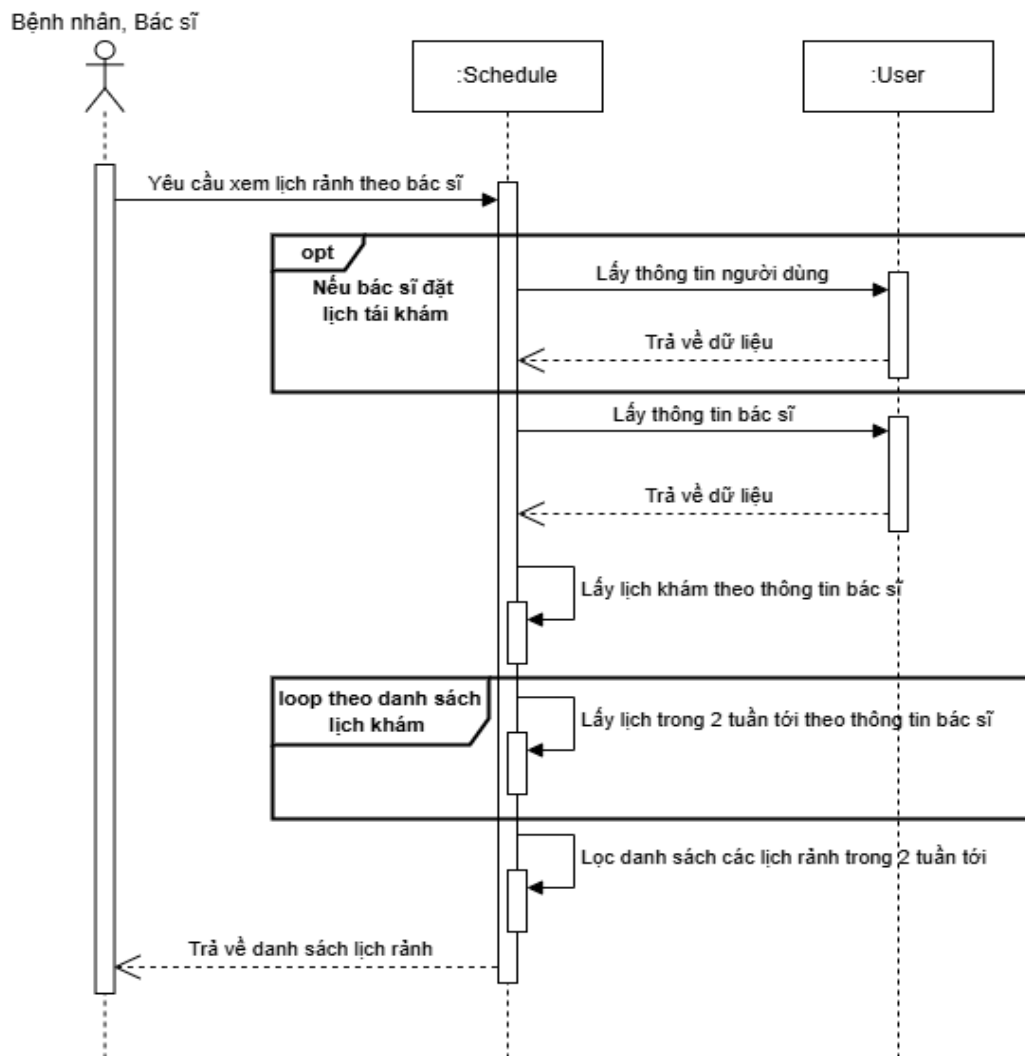
Sơ đồ tuần tự trên minh họa chi tiết quy trình thêm mới một phiên đo vào hệ thống. Đầu tiên, người dùng gửi yêu cầu tạo phiên đo mới, hệ thống xử lý bằng cách tải tệp dữ liệu phiên đo lên máy chủ thông qua lớp Record. Sau khi tệp được tải thành công, hệ thống trả về URL lưu trữ dữ liệu và tiếp tục tạo phiên đo mới, đồng thời gửi thông báo thành công cho người dùng. Trong trường hợp xảy ra lỗi trong quá trình xử lý, hệ thống sẽ tự động xóa tệp dữ liệu đã tải lên và trả về thông báo lỗi cho người dùng.



Hình 2.22 Sơ đồ tuần tự chức năng xóa một hoặc nhiều phiên đo

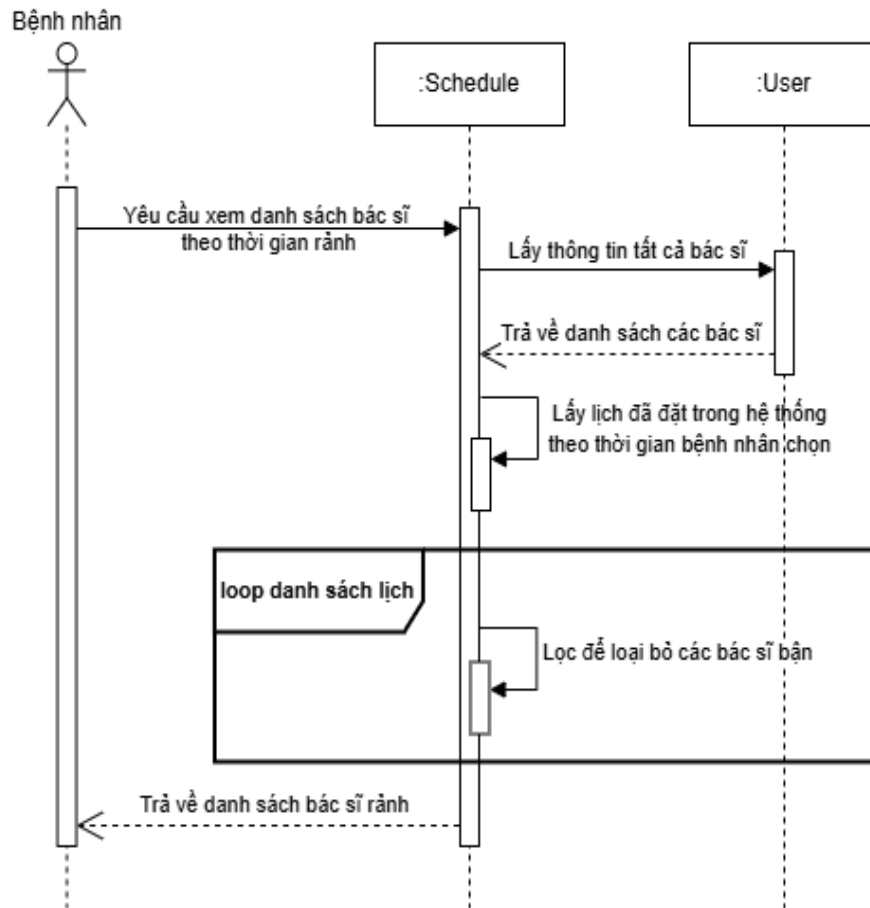
Sơ đồ tuần tự trên minh họa quy trình xóa một hoặc nhiều phiên đo trong hệ thống. Quản trị viên hoặc bác sĩ gửi yêu cầu xóa phiên đo, hệ thống tiếp nhận và lấy thông tin chi tiết phiên đo cần xóa từ lớp Record. Sau khi trả về thông tin cho người dùng và nhận được xác nhận xóa, hệ thống tiến hành xóa thông tin phiên đo trong cơ sở dữ liệu. Cuối cùng, hệ thống trả về kết quả xử lý, đảm bảo rằng thao tác xóa được thực hiện chính xác.

2.3.7 Sơ đồ tuần tự chức năng quản lý dịch vụ lịch khám



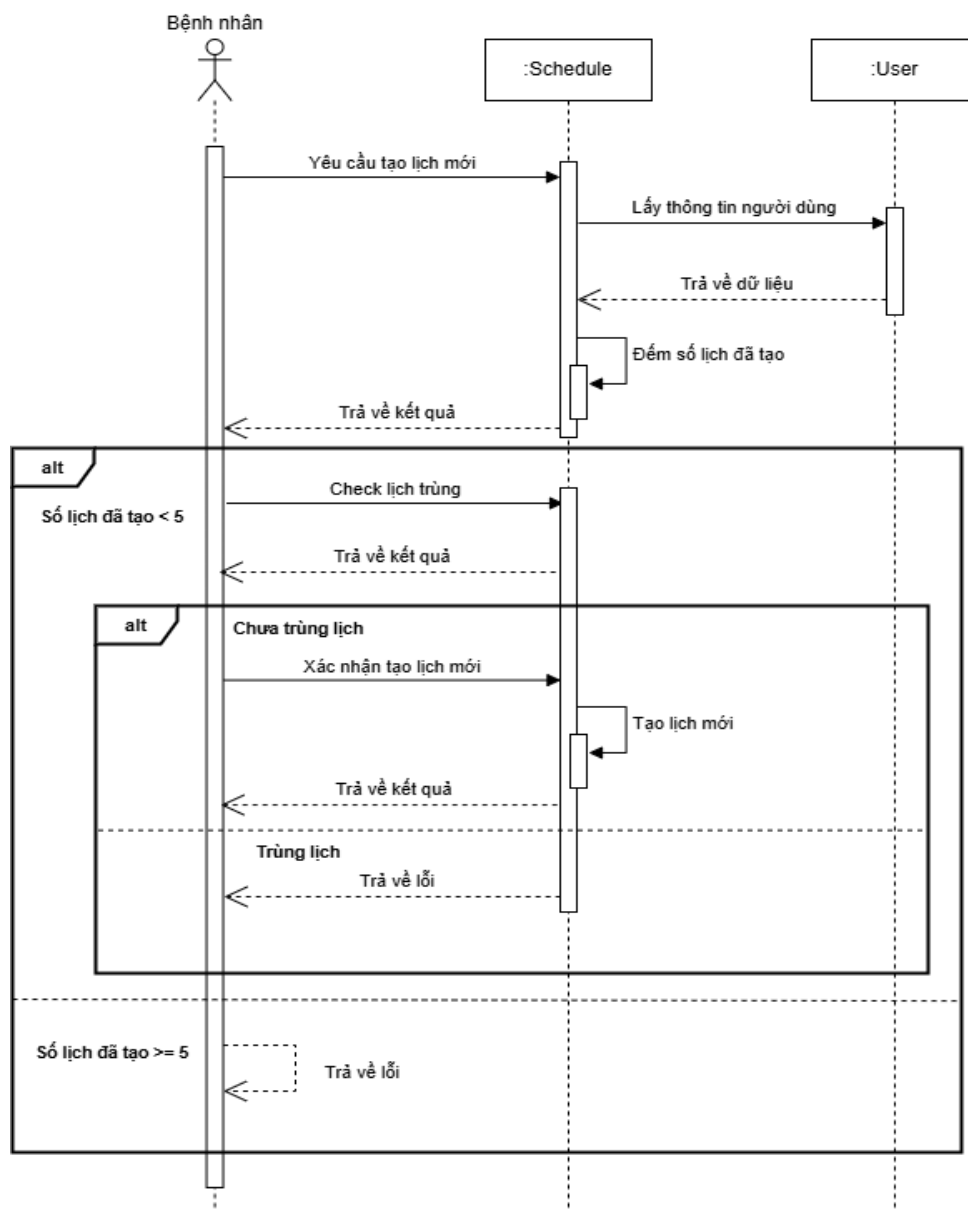
Hình 2.23 Sơ đồ tuần tự chức năng tìm kiếm lịch khám theo bác sĩ

Sơ đồ tuần tự trên minh họa quy trình tìm kiếm lịch rảnh của bác sĩ. Khi bệnh nhân hoặc bác sĩ gửi yêu cầu xem lịch rảnh, hệ thống chuyển yêu cầu đến lớp Schedule để xử lý. Nếu người yêu cầu là bác sĩ (xác định dựa trên tham số đầu vào), hệ thống sẽ truy xuất thông tin của chính bác sĩ từ lớp User. Ngược lại, nếu là bệnh nhân, hệ thống lấy thông tin của bác sĩ đã được chọn. Sau đó, danh sách lịch khám của bác sĩ trong hai tuần tới được truy xuất và lọc để xác định các khoảng thời gian rảnh. Kết quả cuối cùng là danh sách lịch rảnh, được hệ thống trả về để hỗ trợ người dùng trong việc lập kế hoạch.



Hình 2.24 Sơ đồ tuần tự chức năng tìm kiếm lịch khám theo thời gian rảnh

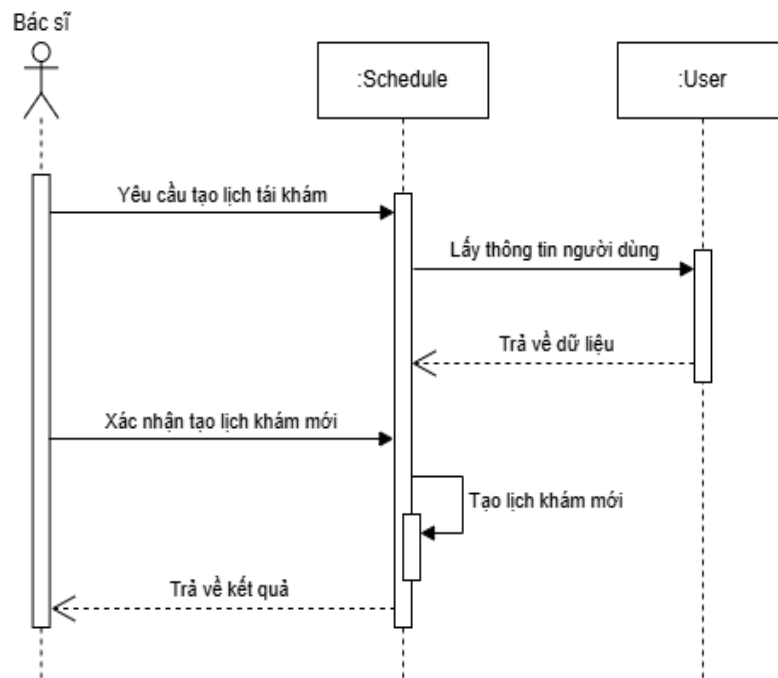
Sơ đồ tuần tự trên minh họa quy trình tìm kiếm lịch khám theo thời gian rảnh do bệnh nhân chọn. Khi bệnh nhân gửi yêu cầu xem danh sách bác sĩ theo thời gian rảnh, hệ thống sẽ chuyển yêu cầu đến lớp Schedule để xử lý. Đầu tiên, hệ thống truy xuất thông tin của tất cả các bác sĩ từ lớp User và trả về danh sách. Tiếp theo, dựa trên thời gian rảnh mà bệnh nhân đã chọn, hệ thống lấy danh sách lịch khám hiện có trong hệ thống. Danh sách này sau đó được lọc để loại bỏ các bác sĩ không rảnh trong khung thời gian đã chọn. Kết quả cuối cùng là danh sách các bác sĩ rảnh trong thời gian đó, được trả về cho bệnh nhân để hỗ trợ lựa chọn.



Hình 2.25 Sơ đồ tuần tự chức năng đặt lịch khám bởi bệnh nhân

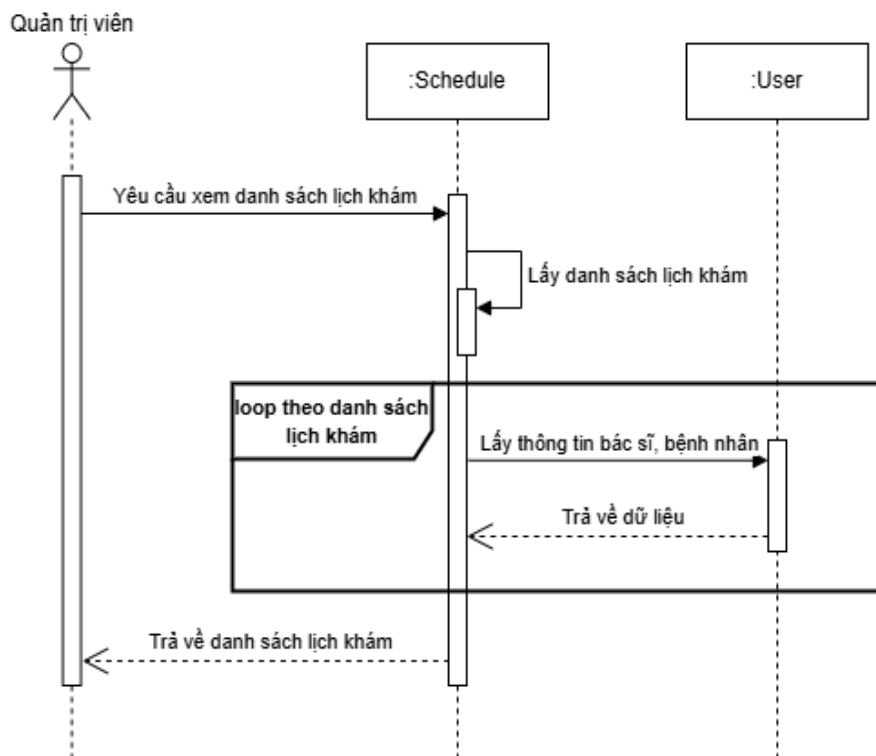
Sơ đồ tuần tự trên minh họa chi tiết quy trình đặt lịch khám mới của bệnh nhân. Đầu tiên, bệnh nhân gửi yêu cầu tạo lịch khám mới, hệ thống sẽ chuyển yêu cầu đến lớp Schedule để xử lý. Lớp Schedule sẽ lấy thông tin người dùng từ lớp User và kiểm tra số lượng lịch đã tạo của bệnh nhân.

- Trường hợp số lịch đã tạo dưới 5: Hệ thống kiểm tra xem lịch mới có trùng với các lịch hiện tại không. Nếu không trùng, hệ thống xác nhận và tạo lịch mới, sau đó trả về kết quả thành công. Nếu trùng lịch, hệ thống sẽ trả về thông báo lỗi cho bệnh nhân.
- Trường hợp số lịch đã tạo từ 5 trở lên: Hệ thống trả về thông báo lỗi, từ chối việc tạo thêm lịch.



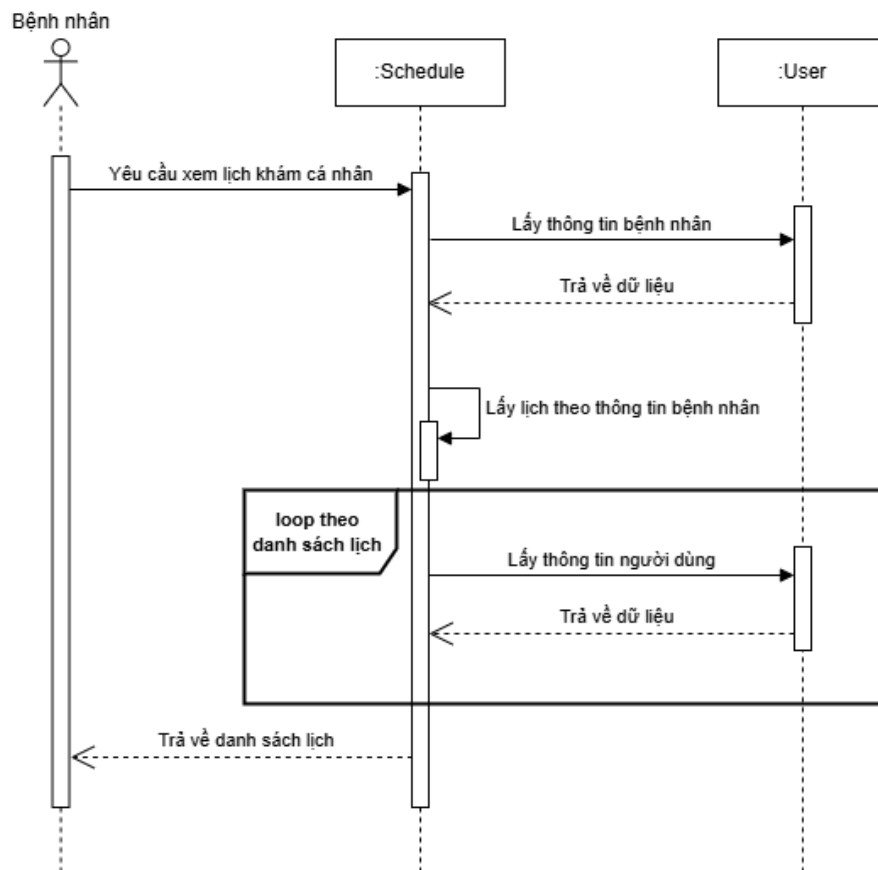
Hình 2.26 Sơ đồ tuần tự chức năng đặt lịch tái khám bởi bác sĩ

Sơ đồ tuần tự trên mô tả quy trình đặt lịch tái khám do bác sĩ thực hiện. Bác sĩ bắt đầu bằng cách gửi yêu cầu tạo lịch tái khám. Hệ thống tiếp nhận yêu cầu và lấy thông tin người dùng từ lớp User để xác thực. Sau khi hoàn tất xác minh, hệ thống tạo lịch tái khám mới thông qua lớp Schedule và gửi lại thông báo kết quả, cho biết quá trình tạo lịch đã thành công hay thất bại.



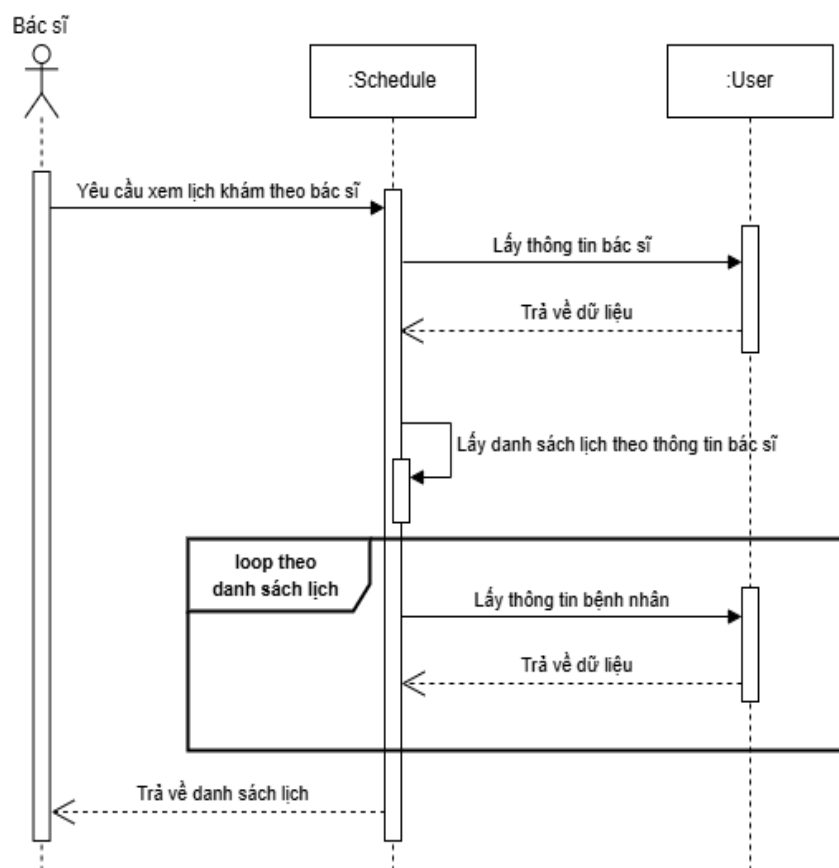
Hình 2.27 Sơ đồ tuần tự chức năng tra cứu danh sách lịch khám trong hệ thống

Sơ đồ tuần tự trên minh họa quy trình tra cứu danh sách lịch khám trong hệ thống do quản trị viên thực hiện. Quản trị viên gửi yêu cầu xem danh sách lịch khám, hệ thống tiếp nhận và truy xuất danh sách lịch khám từ lớp Schedule. Sau đó, hệ thống duyệt qua từng lịch khám, truy vấn thông tin chi tiết về bác sĩ và bệnh nhân liên quan từ lớp User. Cuối cùng, hệ thống trả về danh sách lịch khám kèm theo thông tin chi tiết, hỗ trợ quản trị viên dễ dàng theo dõi và quản lý.



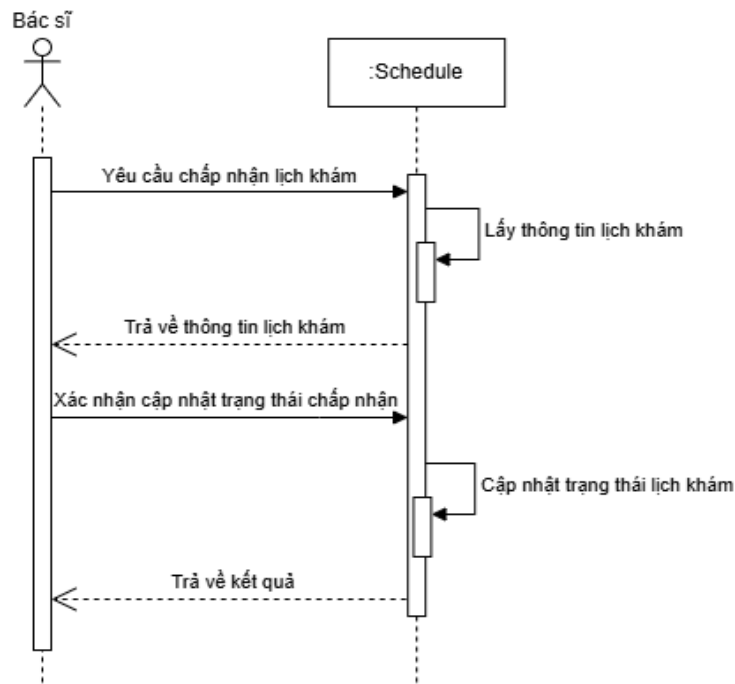
Hình 2.28 Sơ đồ tuần tự chức năng tra cứu lịch khám cá nhân

Sơ đồ tuần tự trên mô tả chi tiết quy trình tra cứu lịch khám cá nhân của bệnh nhân. Khi bệnh nhân gửi yêu cầu, hệ thống sẽ tiếp nhận và truy xuất thông tin bệnh nhân từ lớp User. Tiếp đó, hệ thống lấy danh sách lịch khám liên quan từ lớp Schedule. Để đảm bảo thông tin đầy đủ, hệ thống duyệt qua từng lịch khám và truy vấn thêm thông tin bác sĩ tương ứng với lịch khám. Sau khi hoàn tất, hệ thống trả về danh sách lịch khám chi tiết, hỗ trợ bệnh nhân trong việc theo dõi và quản lý lịch trình khám chữa bệnh.



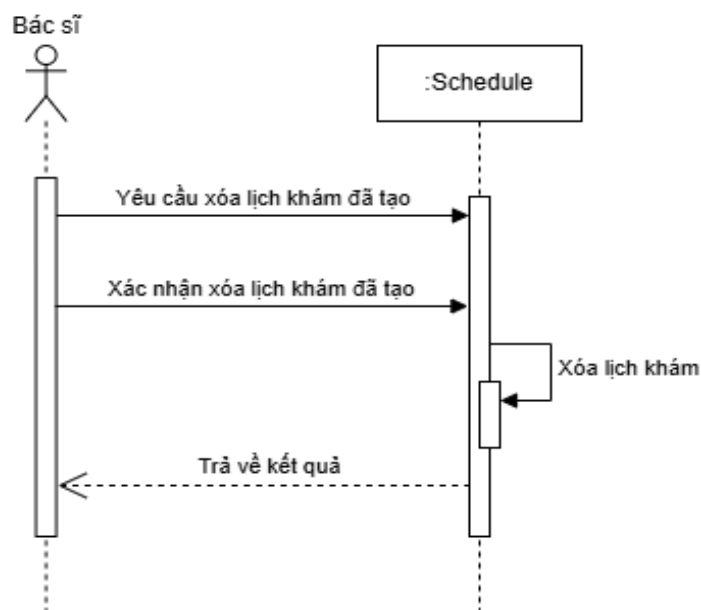
Hình 2.29 Sơ đồ tuần tự chức năng tra cứu lịch khám của bệnh nhân đang được theo dõi

Sơ đồ tuần tự trên minh họa quy trình tra cứu lịch khám của bệnh nhân đang được theo dõi bởi bác sĩ. Khi bác sĩ gửi yêu cầu, hệ thống sẽ truy xuất thông tin của bác sĩ từ lớp User để xác định danh sách lịch khám liên quan. Tiếp theo, hệ thống lấy danh sách lịch khám dựa trên thông tin bác sĩ từ lớp Schedule. Để cung cấp thông tin chi tiết, hệ thống duyệt qua từng lịch khám và truy vấn thông tin của bệnh nhân liên quan. Cuối cùng, hệ thống trả về danh sách lịch khám chi tiết cho bác sĩ, hỗ trợ trong việc quản lý và theo dõi bệnh nhân hiệu quả.



Hình 2.30 Sơ đồ tuần tự chức năng chấp nhận lịch khám

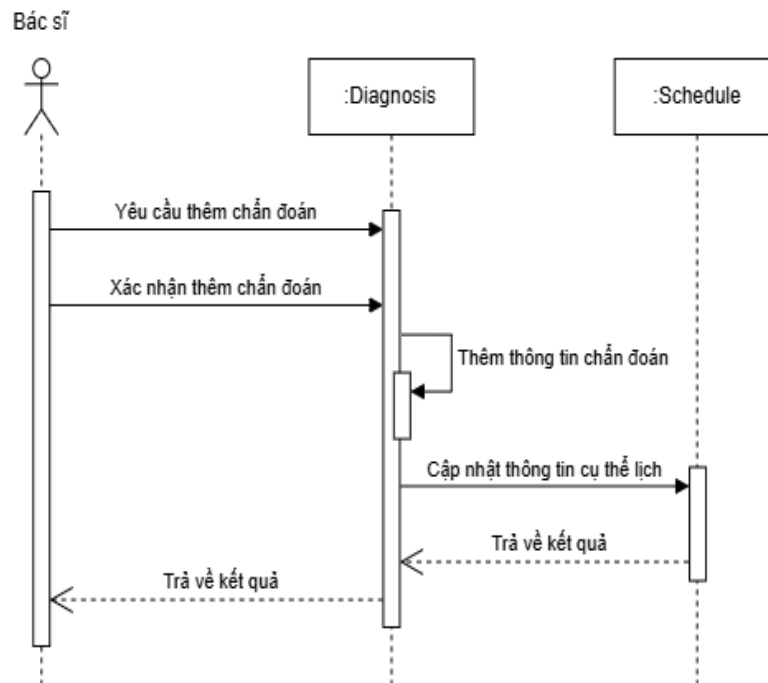
Sơ đồ tuần tự trên minh họa quy trình chấp nhận lịch khám của bác sĩ. Đầu tiên, bác sĩ gửi yêu cầu chấp nhận lịch khám. Hệ thống tiếp nhận yêu cầu và truy xuất thông tin lịch khám từ lớp Schedule để xác minh. Sau khi kiểm tra, bác sĩ xác nhận và hệ thống cập nhật trạng thái lịch khám thành đã chấp nhận. Cuối cùng, kết quả và các thông báo liên quan được trả về cho bác sĩ và bệnh nhân của lịch khám tương ứng, đảm bảo thông tin được cập nhật chính xác và kịp thời.



Hình 2.31 Sơ đồ tuần tự chức năng từ chối lịch khám

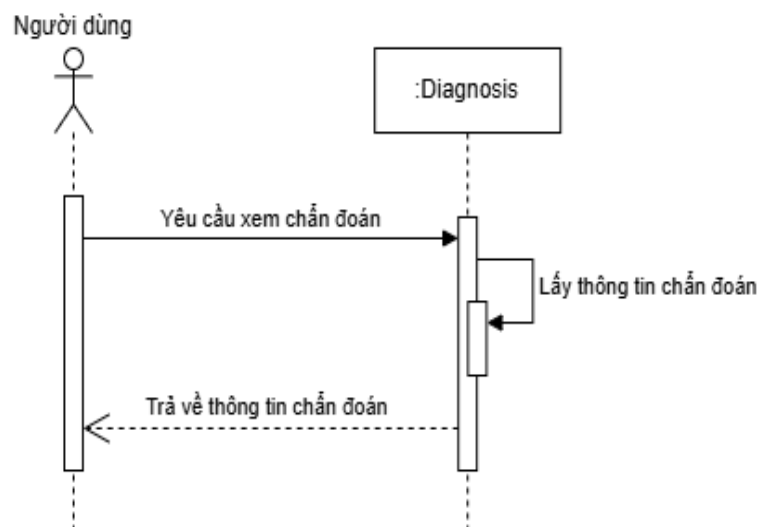
Sơ đồ tuần tự trên minh họa quy trình từ chối lịch khám của bác sĩ. Bác sĩ bắt đầu

bằng việc gửi yêu cầu hủy lịch khám bệnh nhân đã tạo. Hệ thống tiếp nhận yêu cầu, xác nhận hành động hủy và tiến hành xóa lịch khám tương ứng thông qua lớp Schedule. Sau khi quá trình hoàn tất, hệ thống trả về thông báo cho cả bác sĩ và bệnh nhân tương ứng về việc hủy lịch.



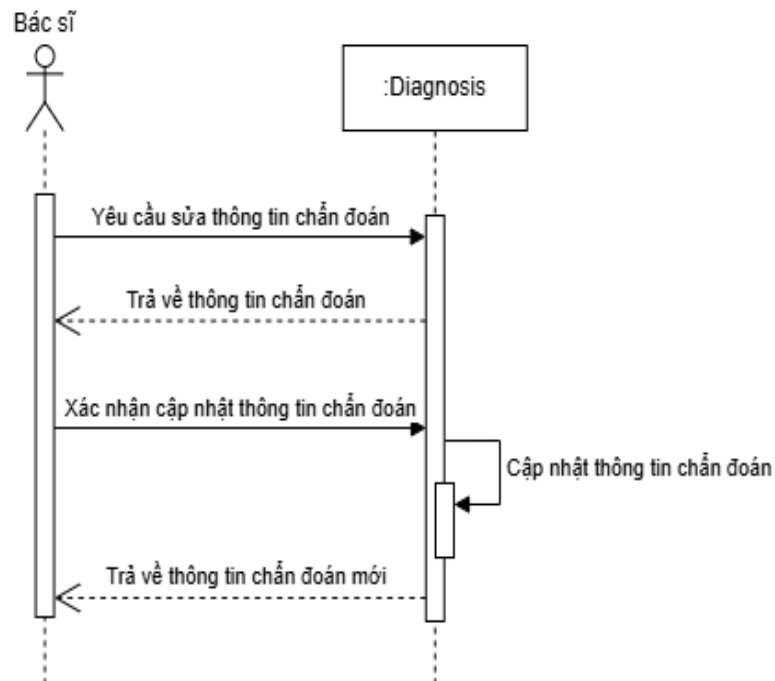
Hình 2.32 Sơ đồ tuần tự chức năng điền chẩn đoán

Sơ đồ tuần tự trên mô tả quy trình bác sĩ thực hiện điền chẩn đoán cho bệnh nhân. Ban đầu, bác sĩ gửi yêu cầu thêm thông tin chẩn đoán, hệ thống tiếp nhận và chờ xác nhận từ bác sĩ. Sau khi được xác nhận, hệ thống cập nhật thông tin chẩn đoán vào lớp Diagnosis, đồng thời điều chỉnh kết quả lịch khám thông qua lớp Schedule. Cuối cùng, hệ thống trả về kết quả, thông báo việc điền chẩn đoán đã được hoàn tất thành công.



Hình 2.33 Sơ đồ tuần tự chức năng tra cứu thông tin chẩn đoán của lịch khám

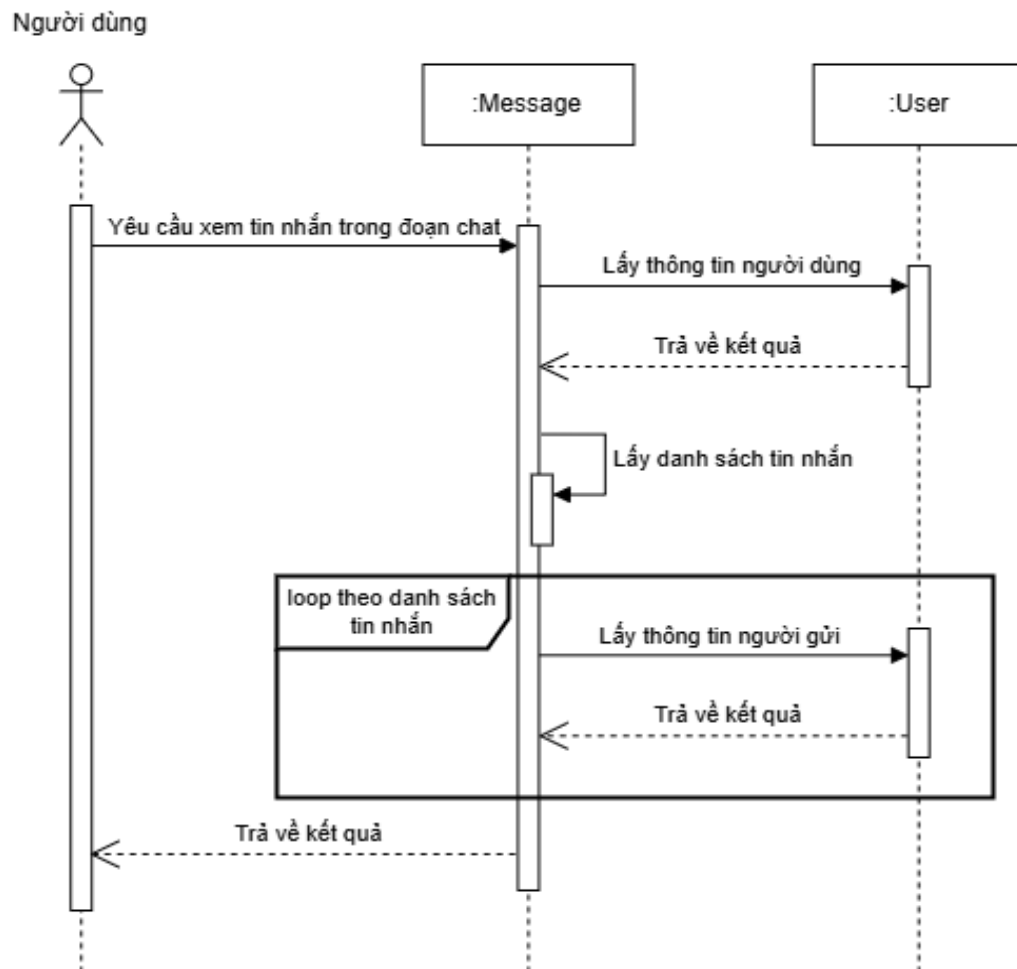
Sơ đồ tuần tự trên minh họa quy trình tra cứu thông tin chẩn đoán của lịch khám. Người dùng gửi yêu cầu xem thông tin chẩn đoán, hệ thống tiếp nhận và thực hiện truy xuất dữ liệu chẩn đoán từ lớp Diagnosis. Sau khi lấy được thông tin cần thiết, hệ thống trả về kết quả cho bệnh nhân, hỗ trợ họ nắm bắt thông tin chẩn đoán của lịch hẹn tương ứng một cách chi tiết và dễ dàng.



Hình 2.34 Sơ đồ tuần tự chức năng chỉnh sửa thông tin chẩn đoán của lịch khám

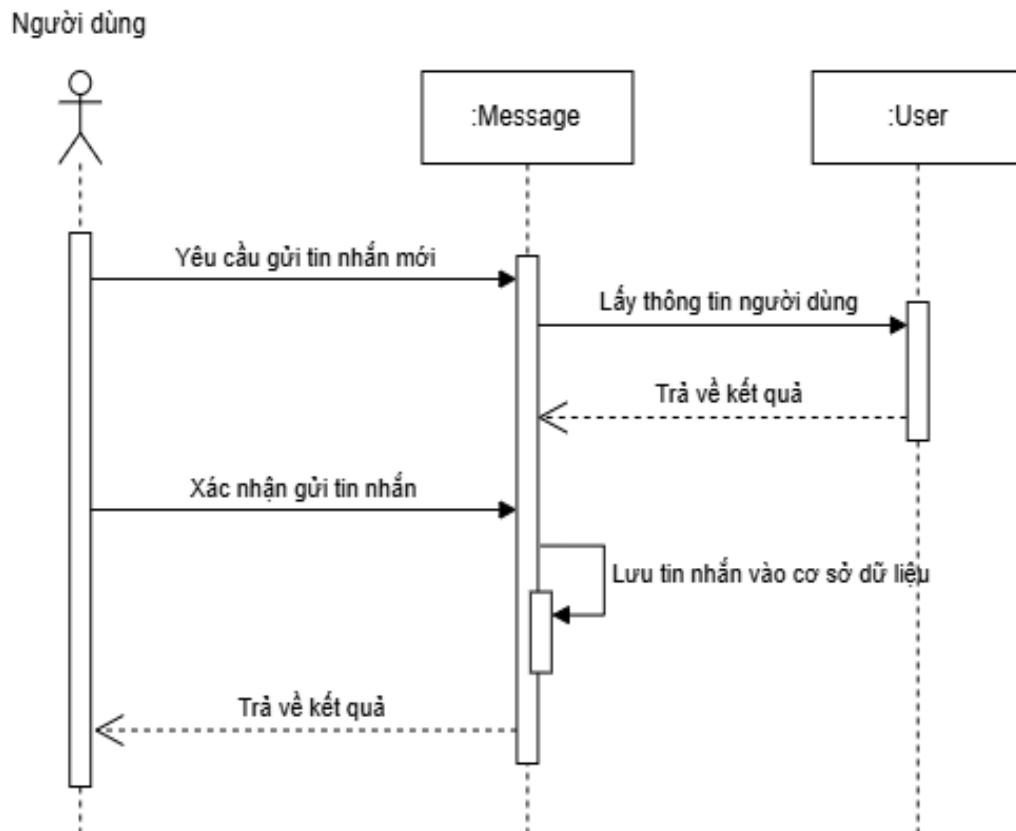
Sơ đồ tuần tự trên minh họa chi tiết quy trình chỉnh sửa thông tin chẩn đoán của lịch khám. Khi bác sĩ gửi yêu cầu chỉnh sửa, hệ thống sẽ truy xuất thông tin chẩn đoán hiện tại từ lớp Diagnosis và trả kết quả về cho bác sĩ. Sau khi bác sĩ cập nhật các thay đổi cần thiết và xác nhận, hệ thống sẽ tiến hành lưu thông tin mới vào cơ sở dữ liệu. Kết thúc quy trình, hệ thống phản hồi xác nhận việc chỉnh sửa thành công và hiển thị chẩn đoán đã được cập nhật, đảm bảo mọi thông tin được cập nhật đầy đủ và chính xác.

2.3.8 Sơ đồ tuần tự chức năng quản lý dịch vụ nhắn tin



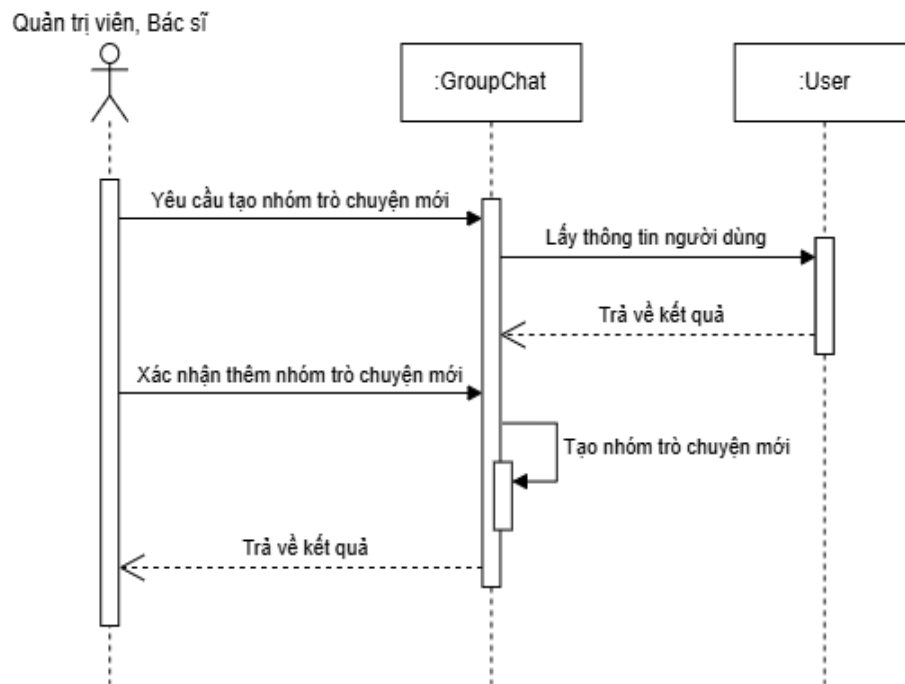
Hình 2.35 Sơ đồ tuần tự chức năng xem lịch sử trò chuyện của các cuộc hội thoại

Sơ đồ tuần tự trên mô tả quy trình xem lịch sử trò chuyện của các cuộc hội thoại. Người dùng gửi yêu cầu xem tin nhắn trong đoạn hội thoại, hệ thống tiếp nhận và truy xuất thông tin người dùng từ lớp User để xác thực. Sau đó, hệ thống truy xuất danh sách tin nhắn từ lớp Message. Đối với mỗi tin nhắn, hệ thống tiếp tục lấy thông tin của người gửi từ lớp User để bổ sung chi tiết. Cuối cùng, hệ thống trả về lịch sử cuộc hội thoại đầy đủ.



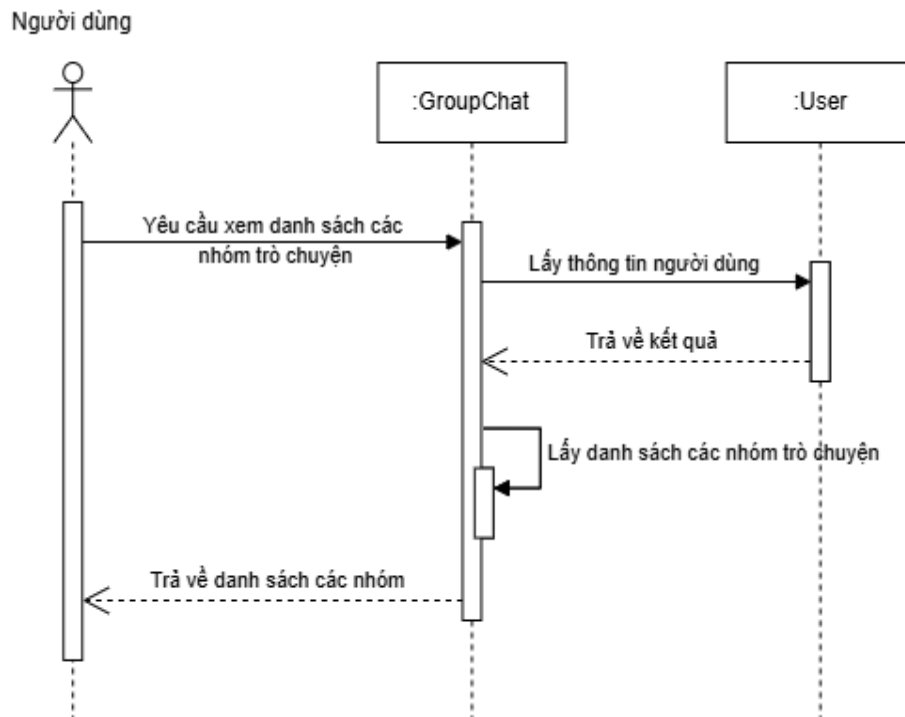
Hình 2.36 Sơ đồ tuần tự chức năng gửi tin nhắn tới các cuộc hội thoại

Sơ đồ tuần tự trên minh họa quy trình gửi tin nhắn mới trong một cuộc hội thoại. Khi người dùng gửi yêu cầu gửi tin nhắn, hệ thống sẽ xác thực thông tin người dùng thông qua lớp User. Sau khi thông tin được xác nhận, hệ thống lưu nội dung tin nhắn vào cơ sở dữ liệu thông qua lớp Message. Cuối cùng, hệ thống trả về kết quả xử lý, thông báo việc gửi tin nhắn thành công hoặc thất bại.



Hình 2.37 Sơ đồ tuần tự chức năng tạo nhóm trò chuyện

Sơ đồ tuần tự trên minh họa quy trình tạo nhóm trò chuyện mới. Đầu tiên, quản trị viên hoặc bác sĩ gửi yêu cầu tạo nhóm trò chuyện. Hệ thống sẽ truy xuất thông tin người dùng thông qua lớp User để xác thực và đảm bảo dữ liệu hợp lệ. Sau đó, hệ thống tiến hành tạo nhóm trò chuyện mới qua lớp GroupChat. Cuối cùng, hệ thống trả về kết quả xác nhận việc tạo nhóm thành công.



Hình 2.38 Sơ đồ tuần tự chức năng hiển thị tất cả các nhóm trò chuyện

Sơ đồ tuần tự trên minh họa quy trình hiển thị tất cả các nhóm trò chuyện. Người dùng gửi yêu cầu xem danh sách nhóm trò chuyện, hệ thống tiếp nhận và truy xuất thông tin người dùng thông qua lớp User để xác thực. Sau đó, hệ thống tiếp tục lấy danh sách các nhóm trò chuyện mà người dùng tham gia thông qua lớp GroupChat. Cuối cùng, hệ thống trả về danh sách nhóm trò chuyện đầy đủ tương ứng với người dùng.

Các sơ đồ tuần tự trên làm rõ cách thức hoạt động của từng chức năng trong hệ thống, bao gồm quản lý người dùng, thiết bị, dữ liệu phiên đo, lịch khám, chẩn đoán bệnh nhân, cũng như giao tiếp qua tin nhắn và nhóm trò chuyện. Thông qua các bước minh họa chi tiết, hệ thống đảm bảo sự tương tác mượt mà giữa người dùng và các lớp chức năng. Những sơ đồ này không chỉ cung cấp cái nhìn toàn diện về luồng xử lý mà còn hỗ trợ quá trình thiết kế, triển khai và kiểm tra tính đúng đắn của các tính năng, đảm bảo hệ thống vận hành hiệu quả và đáp ứng nhu cầu của người dùng.

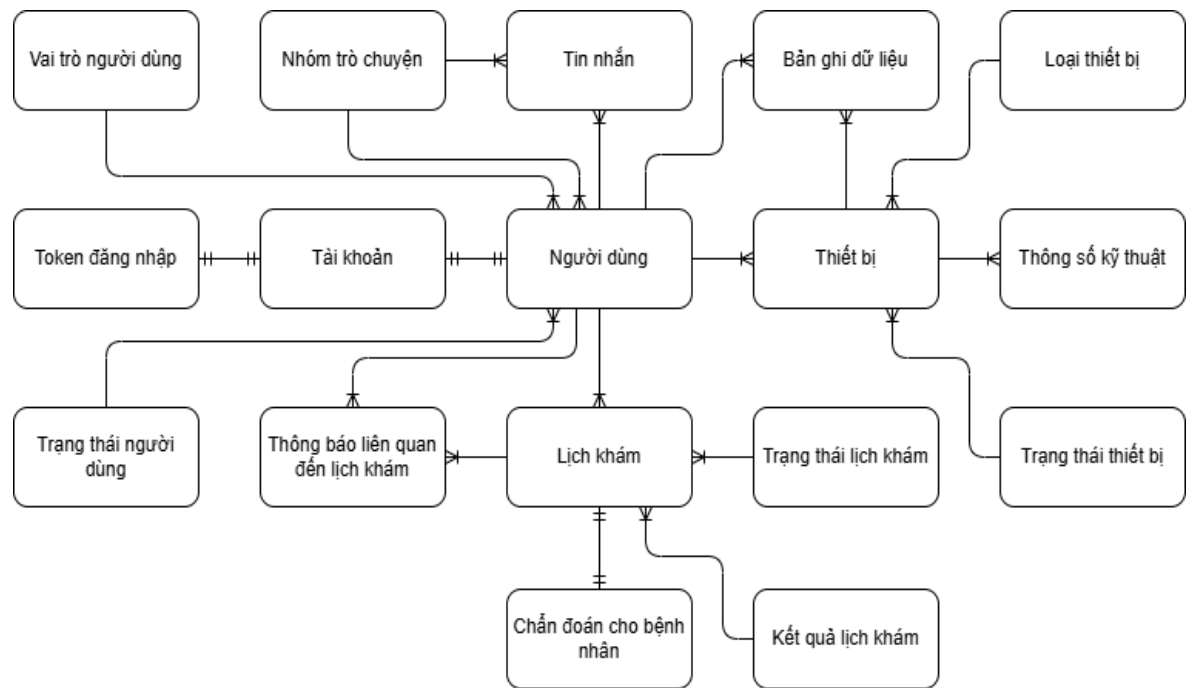
2.4 Xử lý và phân tích dữ liệu

Trong phần này, chúng em sẽ xác định và mô tả chi tiết các thực thể cùng các thuộc tính trong hệ thống. Quá trình này đóng vai trò quan trọng trong việc giúp chúng em hiểu rõ các thành phần cốt lõi, từ đó thiết kế và xây dựng cơ sở dữ liệu một cách tối ưu và hiệu quả.

Trước tiên, chúng em sẽ tiến hành xác định các thực thể trong hệ thống và mô tả chi tiết các thuộc tính của chúng thông qua bảng biểu và sơ đồ liên kết.

Thực thể	Thuộc tính
Tài khoản	ID tài khoản, Địa chỉ email đăng ký, Mật khẩu truy cập
Token đăng nhập	ID token, Token làm mới, Hạn sử dụng, Trạng thái token
Vai trò người dùng	ID vai trò, Tên vai trò
Trạng thái người dùng	ID trạng thái người dùng, Mô tả trạng thái
Người dùng	ID người dùng, Tên đầy đủ, Ngày tháng năm sinh, Giới tính, Số liên lạc, Vai trò người dùng, Trạng thái hoạt động, Đường dẫn ảnh đại diện, Thông tin bổ sung
Loại thiết bị	ID loại thiết bị, Tên loại thiết bị
Trạng thái thiết bị	ID trạng thái thiết bị, Mô tả trạng thái
Thiết bị	ID thiết bị, Tên thiết bị, Loại thiết bị, Thông tin chi tiết về thiết bị, Tình trạng hiện tại, Ngày bắt đầu thời gian mượn, Ngày kết thúc thời gian mượn
Thông số kỹ thuật	ID thông số, Tên thông số, Loại thông số, Giá trị thông số, Mô tả chi tiết thông số
Dữ liệu phiên đo	ID dữ liệu phiên đo, Đường dẫn lưu trữ phiên đo, Thời gian bắt đầu thu thập dữ liệu, Thời gian kết thúc thu thập dữ liệu
Trạng thái lịch khám	ID trạng thái, Mô tả trạng thái
Kết quả lịch khám	ID kết quả lịch khám, Mô tả kết quả lịch khám
Lịch khám	ID lịch khám, Thời gian bắt đầu lịch khám, Thời gian kết thúc lịch khám, Kết quả lịch khám
Chẩn đoán cho bệnh nhân	ID chẩn đoán, Thông tin chẩn đoán
Thông báo liên quan đến lịch khám	ID thông báo, Loại thông báo (gửi cho bác sĩ hoặc gửi cho bệnh nhân), Thời gian bắt đầu lịch khám, Nội dung thông báo, Trạng thái thông báo (nhắc khám, được bác sĩ chấp nhận, đang chờ xác nhận, từ chối, đặt lịch tái khám thành công, bị hủy tự động), Trạng thái đã xem, Lý do từ chối lịch (nếu có)
Tin nhắn	ID tin nhắn, Người gửi tin nhắn, Nhóm trò chuyện nhận tin nhắn, Nội dung tin nhắn, Thời điểm gửi
Nhóm trò chuyện	ID nhóm trò chuyện, Tên nhóm trò chuyện, Người tạo nhóm, Danh sách thành viên nhóm, Sự kiện gửi tin nhắn, Sự kiện nhận tin nhắn

Dựa trên bảng thực thể và thuộc tính đã hoàn thiện, mô hình thực thể liên kết của hệ thống được xác định như sau:



Hình 2.39 Mô hình thực thể liên kết

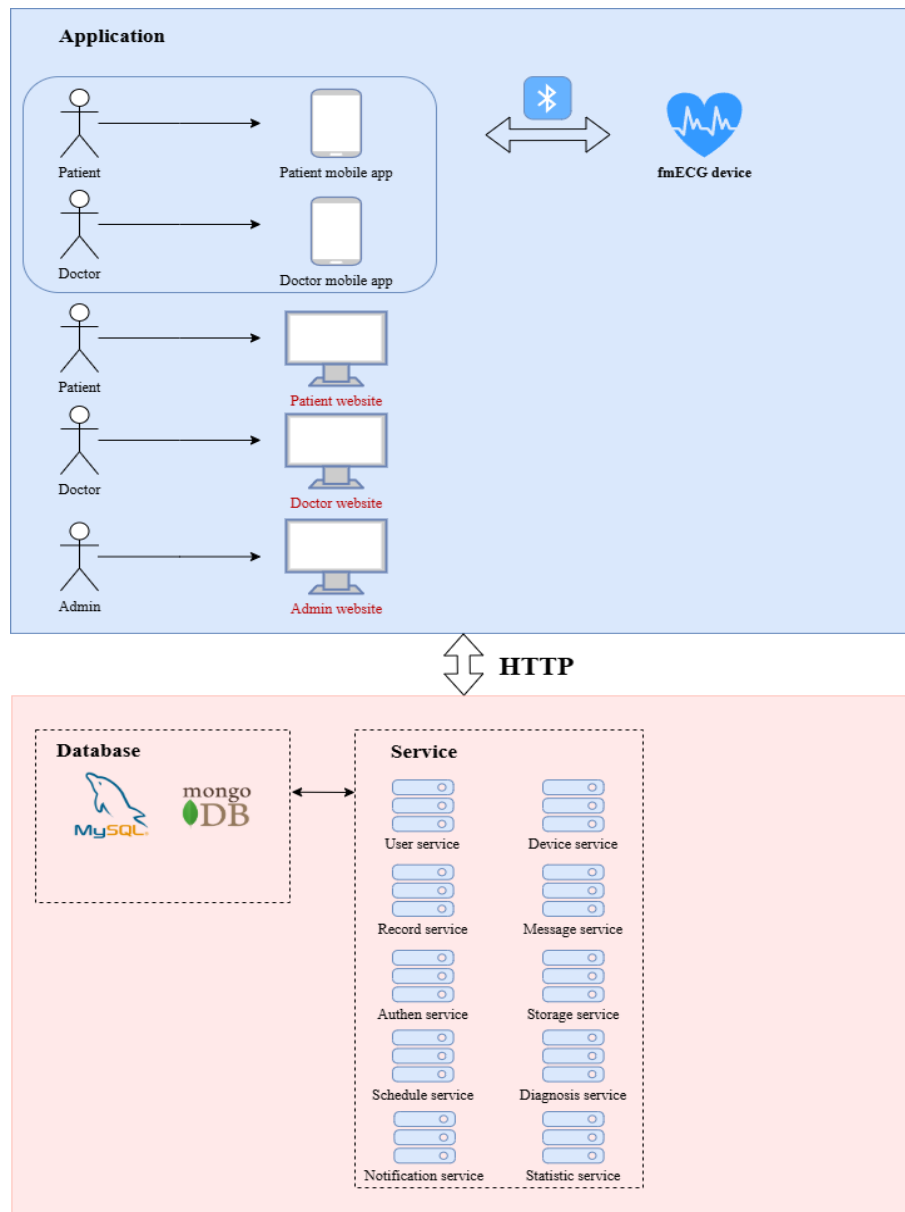
2.5 Kết luận

Chương này tập trung phân tích tổng quan về hệ thống, nhằm đảm bảo đáp ứng đầy đủ các yêu cầu và mục tiêu đã được đề cập ở các phần trước.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

Ở chương này, chúng em mô tả quá trình thiết kế hệ thống từ tổng quan đến chi tiết, dựa trên phân tích ở Chương 2. Mở đầu là xây dựng sơ đồ kiến trúc hệ thống. Tiếp theo, chương tập trung vào thiết kế giao diện người dùng và các chức năng chính cho website cùng server. Nội dung chính được thể hiện qua hình ảnh và sơ đồ minh họa, không chỉ mô tả chi tiết luồng hoạt động mà còn làm rõ cách các thành phần trong hệ thống phối hợp, hỗ trợ lẫn nhau.

3.1 Sơ đồ tổng quan kiến trúc của hệ thống



Hình 3.1 Tổng quan kiến trúc hệ thống

Hệ thống được chia thành ba phần chính: Thiết bị (Device), Máy chủ (Server) và Ứng dụng (bao gồm web và app). Mỗi thành phần đảm nhiệm một vai trò quan trọng, cùng phối hợp để đảm bảo hoạt động của toàn bộ hệ thống như được minh họa trong hình vẽ.

Hình 3.1 thể hiện ba phần:

- Device (Thiết bị): Gồm các thiết bị phần cứng đo điện tim, có khả năng kết nối với ứng dụng di động của bệnh nhân thông qua Bluetooth.
- Application (Ứng dụng): Bao gồm ứng dụng di động và website, phục vụ nhu cầu sử dụng của bệnh nhân, bác sĩ và quản trị viên.
- Server (Máy chủ): Chứa các dịch vụ (Services) xử lý yêu cầu từ ứng dụng và quản lý cơ sở dữ liệu.

Trong sơ đồ kiến trúc hệ thống, bệnh nhân sử dụng ứng dụng di động (Mobile App) để kết nối trực tiếp với thiết bị (Device). Ứng dụng di động thuộc Khối Ứng dụng (Application), chịu trách nhiệm giao tiếp với Khối Máy chủ (Server) thông qua các API và giao thức HTTP. Khi nhận được yêu cầu từ Application, Server sẽ thực hiện xử lý dữ liệu thông qua các dịch vụ (Services) được thiết kế riêng biệt. Tùy theo loại yêu cầu, các dịch vụ này sẽ truy xuất hoặc cập nhật dữ liệu trong cơ sở dữ liệu, sau đó gửi kết quả phản hồi cho người dùng, hoàn thiện quá trình tương tác giữa người dùng và hệ thống.

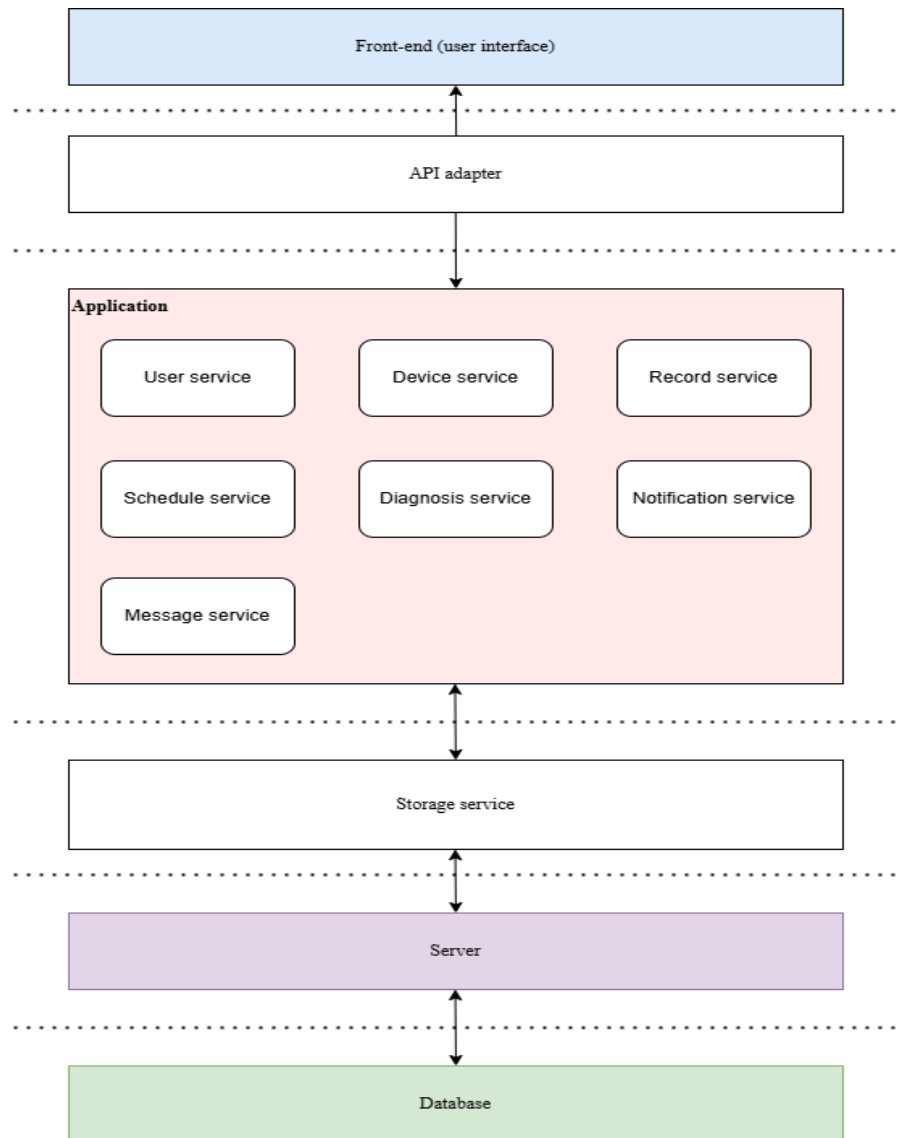
- Authen Service: Đảm nhận các nhiệm vụ liên quan đến bảo mật hệ thống, bao gồm mã hóa dữ liệu nhạy cảm, tạo và xác minh token để đảm bảo tính an toàn khi truy cập, quản lý phân quyền người dùng đối với API, và thực hiện mã hóa thông tin trước khi lưu trữ nhằm ngăn chặn rò rỉ dữ liệu.
- User Service: Xử lý toàn bộ các thao tác liên quan đến người dùng, như tạo tài khoản mới, xác minh thông tin đăng nhập, lấy thông tin cá nhân của người dùng, đồng thời hỗ trợ cập nhật và chỉnh sửa thông tin cá nhân khi cần thiết.
- Device Service: Chịu trách nhiệm quản lý thiết bị, bao gồm các chức năng như thêm mới, chỉnh sửa thông tin, xóa thiết bị, và cập nhật tình trạng thiết bị hoặc thông số liên quan để đảm bảo thiết bị hoạt động đúng trong hệ thống.
- Storage Service: Quản lý và vận hành hệ thống lưu trữ dữ liệu, bao gồm lưu trữ file, tài liệu, và các thông tin quan trọng của hệ thống. Đồng thời, đảm bảo tính nhất quán của dữ liệu thông qua cơ chế xử lý race condition, khóa truy cập và đồng bộ hóa dữ liệu trong các trường hợp truy cập đồng thời, cũng như tối ưu hóa hiệu suất lưu trữ và truy xuất dữ liệu.
- Record Service: Xử lý các dữ liệu liên quan đến phiên đo, bao gồm thêm mới, cập nhật, xóa dữ liệu, và xử lý các file đo được từ thiết bị trước khi lưu trữ hoặc gửi đến người dùng.

- **Message Service:** Quản lý toàn bộ các yêu cầu liên quan đến nhắn tin, bao gồm gửi, nhận, lưu trữ tin nhắn, hỗ trợ các nhóm trò chuyện giữa những người dùng trong hệ thống.
- **Schedule Service:** Đảm nhiệm việc quản lý lịch khám, bao gồm đặt lịch và xử lý các phản hồi liên quan, nhằm đảm bảo quy trình đặt lịch diễn ra mượt mà giữa bệnh nhân, bác sĩ và hệ thống.
- **Diagnosis Service:** Quản lý các tác vụ liên quan đến chẩn đoán, bao gồm thêm mới và chỉnh sửa thông tin chẩn đoán cho bệnh nhân, đảm bảo dữ liệu chính xác và hỗ trợ quá trình điều trị hiệu quả.
- **Notification Service:** Đảm bảo quản lý hiệu quả các thông báo cho người dùng, từ việc gửi thông báo sự kiện, cảnh báo, đến nhắc nhở, giúp người dùng cập nhật kịp thời các thông tin quan trọng từ hệ thống.
- **Statistic Service:** Chịu trách nhiệm tổng hợp số liệu thống kê trong hệ thống, bao gồm số lượng người dùng (bệnh nhân và bác sĩ), số thiết bị, và dữ liệu phiên đo theo từng tháng, nhằm cung cấp thông tin phục vụ quản lý hiệu quả.

Dưới đây là mô tả chi tiết về các phần nhỏ hơn trong kiến trúc hệ thống, được xây dựng dựa trên các đối tượng chính đã phân tích.

3.2 Sơ đồ khối phần mềm

3.2.1 Website dành cho bệnh nhân

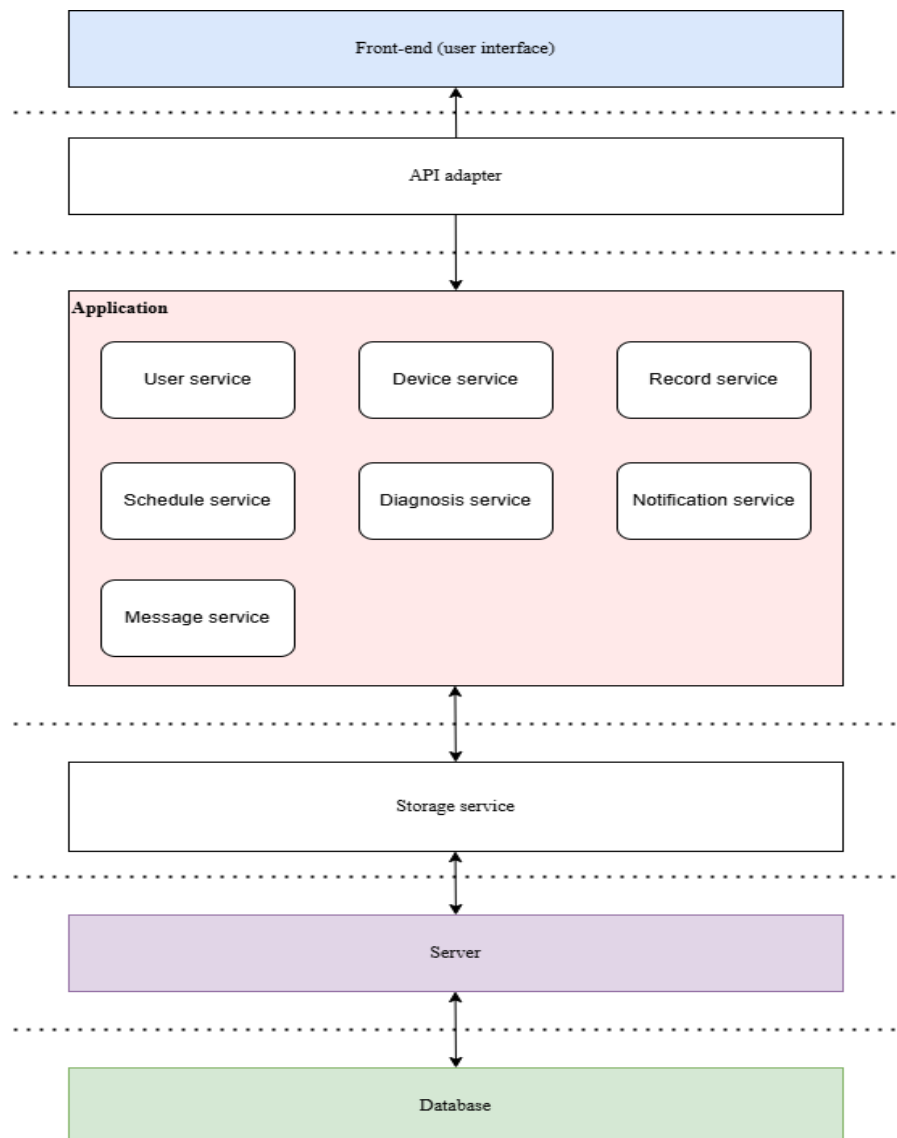


Hình 3.2 Sơ đồ khối Website dành cho bệnh nhân

Tầng trên cùng trong sơ đồ hình 3.2 là User Interface (Giao diện người dùng), nơi bệnh nhân trực tiếp tương tác với hệ thống thông qua API Adapter để gửi yêu cầu và nhận phản hồi. Các yêu cầu này được xử lý bởi các Services chính, bao gồm User Service, Device Service, Record Service, Schedule Service, Diagnosis Service, Notification Service và Storage Service.

Những Services này được thiết kế nhằm đáp ứng các nhu cầu của bệnh nhân, từ quản lý thông tin cá nhân, mượn và trả thiết bị, theo dõi lịch sử dữ liệu phiên đo, cho đến việc quản lý lịch khám, tra cứu thông tin chẩn đoán cho từng lịch khám. Ngoài ra, hệ thống còn đảm bảo việc gửi thông báo nhắc nhở kịp thời và hỗ trợ bệnh nhân trao đổi thông tin với bác sĩ một cách liền mạch và hiệu quả.

3.2.2 Website dành cho bác sĩ



Hình 3.3 Sơ đồ khối Website dành cho bác sĩ

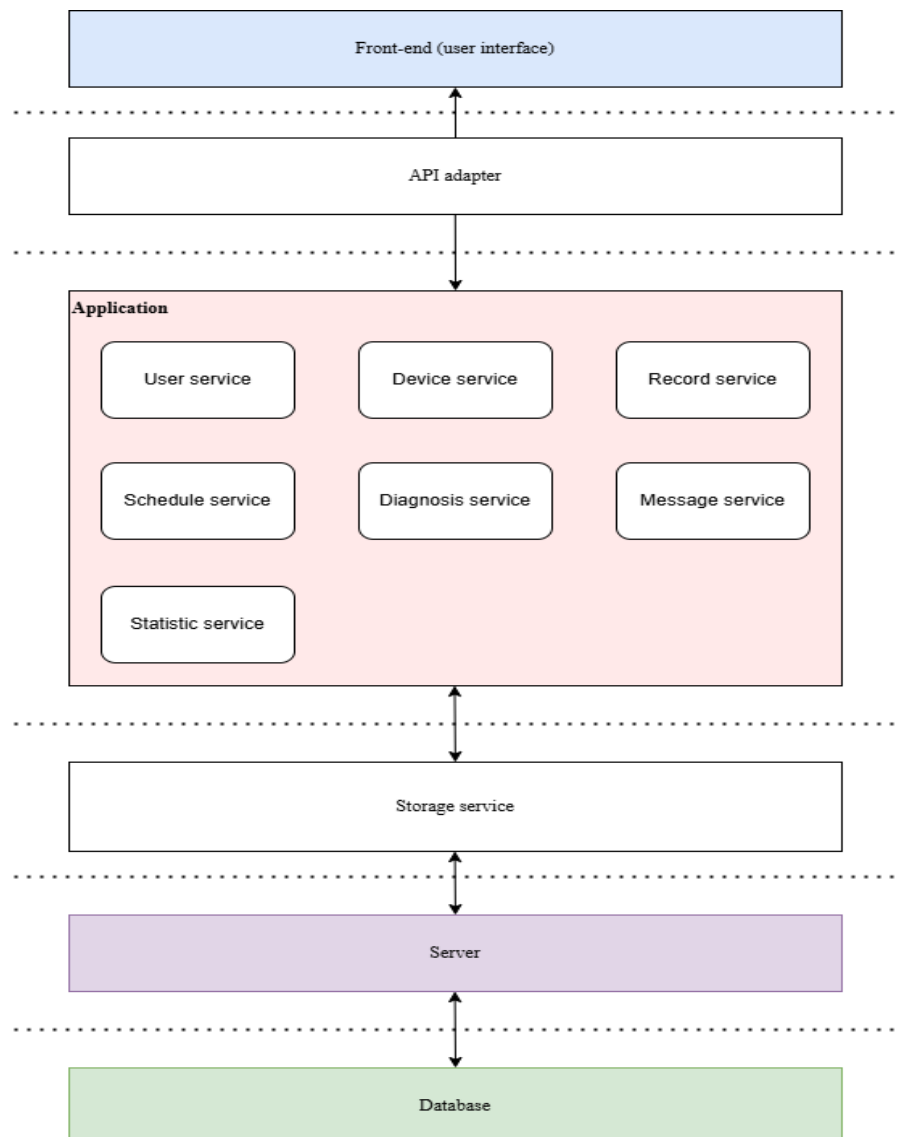
Tương tự với bệnh nhân, sơ đồ hình 3.3 được xây dựng để hỗ trợ bác sĩ thực hiện các nhiệm vụ chuyên môn thông qua giao diện người dùng và API Adapter, đảm bảo việc xử lý thông tin diễn ra nhanh chóng và hiệu quả.

Các Services chính, bao gồm User Service, Device Service, Record Service, Schedule Service, Diagnosis Service, Notification Service và Storage Service, đóng vai trò quan trọng trong việc hỗ trợ bác sĩ. Các nhiệm vụ được tập trung vào việc theo dõi và phân tích dữ liệu phiên đo từ bệnh nhân, quản lý lịch khám, chấp nhận hoặc từ chối yêu cầu đặt lịch, ghi nhận thông tin chẩn đoán, và trao đổi trực tiếp với bệnh nhân. Hơn nữa, hệ thống cung cấp khả năng tự động gửi thông báo về lịch khám và nhắc nhở các cuộc khám sắp tới, giúp bác sĩ quản lý thời gian và công việc hiệu quả hơn.

Cách tiếp cận này không chỉ hỗ trợ bác sĩ tổ chức công việc thuận lợi mà còn góp

phần nâng cao hiệu quả điều trị cho bệnh nhân.

3.2.3 Website cho quản trị viên



Hình 3.4 Sơ đồ khối Website dành cho quản trị viên

Về cơ bản, website dành cho admin được thiết kế với cấu trúc tương tự như website dành cho bác sĩ và bệnh nhân nhưng với quyền hạn mở rộng hơn. Admin có thể quản lý toàn bộ thông tin người dùng, dữ liệu phiên đo, theo dõi các thống kê tổng quan của hệ thống, kiểm soát thông tin và kết quả chẩn đoán từ các lịch khám, đồng thời điều hành việc mượn trả thiết bị một cách linh hoạt và hiệu quả.

3.3 *Thiết kế cơ sở dữ liệu*

3.3.1 Chuyển đổi từ mô hình thực thể liên kết sang mô hình quan hệ

Dựa trên bảng mô tả các thực thể và thuộc tính, chúng em tiến hành chuyển đổi từ mô hình thực thể liên kết thành mô hình quan hệ như sau.

- Tài khoản đăng nhập (**ID tài khoản đăng nhập**, Địa chỉ email đăng ký, Mật khẩu truy cập)
- Token đăng nhập (**ID token đăng nhập**, ID tài khoản đăng nhập, Token làm mới, Hạn sử dụng, Trạng thái token)
- Vai trò người dùng (**ID vai trò**, Tên vai trò)
- Trạng thái hoạt động (**ID trạng thái người dùng**, Mô tả trạng thái người dùng)
- Người dùng (**ID người dùng**, ID tài khoản đăng nhập, ID vai trò người dùng, ID trạng thái hoạt động, Tên đầy đủ, Ngày tháng năm sinh, Giới tính, Số liên lạc, Đường dẫn ảnh đại diện, Thông tin bổ sung)
- Loại thiết bị (**ID loại thiết bị**, Tên loại thiết bị)
- Trạng thái thiết bị (**ID trạng thái thiết bị**, Mô tả trạng thái thiết bị)
- Thiết bị (**ID thiết bị**, ID người dùng thiết bị, ID loại thiết bị, ID trạng thái thiết bị, Tên thiết bị, Thông tin chi tiết về thiết bị, Ngày bắt đầu thời gian mượn, Ngày kết thúc thời gian mượn)
- Thông số kỹ thuật (**ID thông số kỹ thuật**, ID thiết bị, Loại thông số, Tên thông số, Giá trị thông số, Mô tả chi tiết thông số)
- Dữ liệu phiên đo (**ID dữ liệu phiên đo**, ID bệnh nhân, ID thiết bị, Loại bản ghi, Đường dẫn lưu trữ dữ liệu phiên đo, Thời gian bắt đầu thu thập dữ liệu, Thời gian kết thúc thu thập dữ liệu)
- Trạng thái lịch khám (**ID trạng thái lịch khám**, Mô tả trạng thái lịch khám)
- Kết quả lịch khám (**ID kết quả lịch khám**, Mô tả kết quả lịch khám)
- Lịch khám (**ID lịch khám**, ID bệnh nhân, ID bác sĩ, ID trạng thái lịch khám, ID kết quả lịch khám, Thời gian bắt đầu lịch khám, Thời gian kết thúc lịch khám)
- Thông báo liên quan đến lịch khám (**ID thông báo**, ID lịch khám, Loại thông báo, Nội dung thông báo, Trạng thái thông báo, Trạng thái đã xem, Lý do từ chối lịch khám)

- Chẩn đoán (**ID chẩn đoán**, ID lịch khám, Thông tin chẩn đoán)
- Tin nhắn (**ID tin nhắn**, ID người gửi, ID nhóm trò chuyện nhận tin nhắn, Nội dung tin nhắn, Thời gian gửi tin nhắn)
- Nhóm trò chuyện (**ID nhóm trò chuyện**, Tên nhóm trò chuyện, Người tạo nhóm, Danh sách thành viên nhóm, Sự kiện gửi tin nhắn, Sự kiện nhận tin nhắn)

3.3.2 Chuẩn hoá 3NF

Các bảng đã được thiết kế theo nguyên tắc chuẩn hoá 3NF, vì không có thuộc tính lặp lại và các thuộc tính không phụ thuộc vào một tập hợp con của khóa chính.

3.3.2.1 Chuẩn hoá bảng Tài khoản

Bảng 3.1 Bảng chuẩn hoá bảng Tài khoản đăng nhập

Danh sách thuộc tính	ID tài khoản đăng nhập, Địa chỉ email đăng ký, Mật khẩu truy cập
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi tài khoản có một ID riêng, có duy nhất Địa chỉ email đăng ký, Mật khẩu truy cập	ID tài khoản → Địa chỉ email đăng ký, Mật khẩu truy cập
⇒ Khoá chính của bảng: ID tài khoản đăng nhập ⇒ Bảng Tài khoản đăng nhập đã ở 3NF	

3.3.2.2 Chuẩn hoá bảng Token đăng nhập

Bảng 3.2 Bảng chuẩn hoá bảng Token đăng nhập

Danh sách thuộc tính	ID token đăng nhập, ID tài khoản đăng nhập, Token làm mới, Hạn sử dụng, Trạng thái token
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi tài khoản có một ID token riêng, có duy nhất ID tài khoản, Token làm mới, Hạn sử dụng, Trạng thái token	ID token → ID tài khoản, Token làm mới, Hạn sử dụng, Trạng thái token
⇒ Khoá chính của bảng: ID token đăng nhập ⇒ Bảng Token đăng nhập đã ở 3NF	

3.3.2.3 Chuẩn hoá bảng Vai trò người dùng

Bảng 3.3 Bảng chuẩn hoá bảng Vai trò người dùng

Danh sách thuộc tính	ID vai trò, Tên vai trò
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi vai trò có một ID riêng, có duy nhất Tên vai trò	ID vai trò → Tên vai trò
⇒ Khoá chính của bảng: ID vai trò	
⇒ Bảng Vai trò người dùng đã ở 3NF	

3.3.2.4 Chuẩn hoá bảng Trạng thái hoạt động

Bảng 3.4 Bảng chuẩn hoá bảng Trạng thái hoạt động

Danh sách thuộc tính	ID trạng thái người dùng, Mô tả trạng thái người dùng
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi trạng thái người dùng có một ID riêng, có duy nhất Mô tả trạng thái người dùng	ID trạng thái người dùng → Mô tả trạng thái người dùng
⇒ Khoá chính của bảng: ID trạng thái người dùng	
⇒ Bảng Trạng thái hoạt động đã ở 3NF	

3.3.2.5 Chuẩn hoá bảng Người dùng

Bảng 3.5 Bảng chuẩn hoá bảng Người dùng

Danh sách thuộc tính	ID người dùng, ID tài khoản đăng nhập, ID vai trò người dùng, ID trạng thái hoạt động, Tên đầy đủ, Ngày tháng năm sinh, Giới tính, Số liên lạc, Đường dẫn ảnh đại diện, Thông tin bổ sung
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi người dùng có một ID riêng, có duy nhất ID tài khoản đăng nhập, ID vai trò người dùng, ID trạng thái hoạt động, Tên đầy đủ, Ngày tháng năm sinh, Giới tính, Số liên lạc, Đường dẫn ảnh đại diện, Thông tin bổ sung	ID người dùng → ID tài khoản đăng nhập, ID vai trò người dùng, ID trạng thái hoạt động, Tên đầy đủ, Ngày tháng năm sinh, Giới tính, Số liên lạc, Đường dẫn ảnh đại diện, Thông tin bổ sung
⇒ Khoá chính của bảng: ID người dùng	
⇒ Bảng Người dùng đã ở 3NF	

3.3.2.6 Chuẩn hoá bảng Loại thiết bị

Bảng 3.6 Bảng chuẩn hoá bảng Loại thiết bị

Danh sách thuộc tính	ID loại thiết bị, Tên loại thiết bị
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi loại thiết bị có một ID riêng, có duy nhất Tên loại thiết bị	ID loại thiết bị → Tên loại thiết bị
⇒ Khoá chính của bảng: ID loại thiết bị ⇒ Bảng loại thiết bị đã ở 3NF	

3.3.2.7 Chuẩn hoá bảng Trạng thái thiết bị

Bảng 3.7 Bảng chuẩn hoá bảng Trạng thái thiết bị

Danh sách thuộc tính	ID trạng thái thiết bị, Mô tả trạng thái thiết bị
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi trạng thái thiết bị có một ID riêng, có duy nhất Mô tả trạng thái thiết bị	ID trạng thái thiết bị → Mô tả trạng thái thiết bị
⇒ Khoá chính của bảng: ID trạng thái thiết bị ⇒ Bảng Trạng thái thiết bị đã ở 3NF	

3.3.2.8 Chuẩn hoá bảng Thiết bị

Bảng 3.8 Bảng chuẩn hoá bảng Thiết bị

Danh sách thuộc tính	ID thiết bị, ID người dùng thiết bị, Tên thiết bị, Thông tin chi tiết về thiết bị, Ngày bắt đầu thời gian mượn, Ngày kết thúc thời gian mượn
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi thiết bị có một ID thiết bị riêng, có duy nhất tên thiết bị, loại thiết bị, thông tin thiết bị, ID người dùng thiết bị, trạng thái thiết bị, ngày bắt đầu sử dụng, ngày kết thúc sử dụng	ID thiết bị → ID người dùng thiết bị, Tên thiết bị, Loại thiết bị, Thông tin thiết bị, Trạng thái thiết bị, Ngày bắt đầu sử dụng, Ngày kết thúc sử dụng
⇒ Khoá chính của bảng: ID thiết bị ⇒ Bảng Thiết bị đã ở 3NF	

3.3.2.9 Chuẩn hoá bảng Thông số kỹ thuật

Bảng 3.9 Bảng chuẩn hoá bảng Thông số kỹ thuật

Danh sách thuộc tính	ID thông số kỹ thuật, ID thiết bị, Loại thông số, Tên thông số, Giá trị thông số, Mô tả chi tiết thông số
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi thông số kỹ thuật sẽ có một ID riêng, có duy nhất ID thiết bị, Loại thông số, Tên thông số, Giá trị thông số, Mô tả chi tiết thông số	ID thông số kỹ thuật → ID thiết bị, Loại thông số, Tên thông số, Giá trị thông số, Mô tả chi tiết thông số
⇒ Khoá chính của bảng: ID thông số kỹ thuật ⇒ Bảng Thông số kỹ thuật đã ở 3NF	

3.3.2.10 Chuẩn hoá bảng Dữ liệu phiên đo

Bảng 3.10 Bảng chuẩn hoá bảng Dữ liệu phiên đo

Danh sách thuộc tính	ID dữ liệu phiên đo, ID bệnh nhân, ID thiết bị, Loại bản ghi, Đường dẫn lưu trữ dữ liệu phiên đo, Thời gian bắt đầu thu thập dữ liệu, Thời gian kết thúc thu thập dữ liệu
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi dữ liệu phiên đo có một ID riêng, có duy nhất ID bệnh nhân, ID thiết bị, Loại bản ghi, Đường dẫn lưu trữ dữ liệu phiên đo, Thời gian bắt đầu thu thập dữ liệu, Thời gian kết thúc thu thập dữ liệu	ID dữ liệu phiên đo → ID bệnh nhân, ID thiết bị, Loại bản ghi, Đường dẫn lưu trữ dữ liệu phiên đo, Thời gian bắt đầu thu thập dữ liệu, Thời gian kết thúc thu thập dữ liệu
⇒ Khoá chính của bảng: ID dữ liệu phiên đo ⇒ Bảng Dữ liệu phiên đo đã ở 3NF	

3.3.2.11 Chuẩn hoá bảng Trạng thái lịch khám

Bảng 3.11 Bảng chuẩn hoá bảng Trạng thái lịch khám

Danh sách thuộc tính	ID trạng thái lịch khám, Mô tả trạng thái lịch khám
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi trạng thái lịch khám có một ID riêng, có duy nhất Mô tả trạng thái lịch khám	ID trạng thái lịch khám → Mô tả trạng thái lịch khám
⇒ Khoá chính của bảng: ID trạng thái lịch khám ⇒ Bảng Trạng thái lịch khám đã ở 3NF	

3.3.2.12 Chuẩn hoá bảng Kết quả lịch khám

Bảng 3.12 Bảng chuẩn hoá bảng Kết quả lịch khám

Danh sách thuộc tính	ID kết quả lịch khám, Mô tả kết quả lịch khám
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi Kết quả lịch khám có một ID riêng, có duy nhất Mô tả kết quả lịch khám	ID kết quả lịch khám → Mô tả kết quả lịch khám
⇒ Khoá chính của bảng: ID kết quả lịch khám ⇒ Bảng Kết quả lịch khám đã ở 3NF	

3.3.2.13 Chuẩn hoá bảng Lịch khám

Bảng 3.13 Bảng chuẩn hoá bảng Lịch khám

Danh sách thuộc tính	ID lịch khám, ID bệnh nhân, ID bác sĩ, ID trạng thái lịch khám, ID kết quả lịch khám, Thời gian bắt đầu lịch khám, Thời gian kết thúc lịch khám
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi Lịch khám có một ID riêng, có duy nhất ID bệnh nhân, ID bác sĩ, ID trạng thái lịch khám, ID kết quả lịch khám, Thời gian bắt đầu lịch khám, Thời gian kết thúc lịch khám	ID lịch khám → ID bệnh nhân, ID bác sĩ, ID trạng thái lịch khám, ID kết quả lịch khám, Thời gian bắt đầu lịch khám, Thời gian kết thúc lịch khám
⇒ Khoá chính của bảng: ID lịch khám ⇒ Bảng Lịch khám đã ở 3NF	

3.3.2.14 Chuẩn hoá bảng Thông báo liên quan đến lịch khám

Bảng 3.14 Bảng chuẩn hoá bảng Thông báo liên quan đến lịch khám

Danh sách thuộc tính	ID thông báo, ID lịch khám, Loại thông báo, Nội dung thông báo, Trạng thái thông báo, Trạng thái đã xem, Lý do từ chối lịch khám
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi Thông báo liên quan đến lịch khám có một ID riêng, có duy nhất ID lịch khám, Loại thông báo, Nội dung thông báo, Trạng thái thông báo, Trạng thái đã xem, Lý do từ chối lịch khám	ID thông báo → ID lịch khám, Loại thông báo, Nội dung thông báo, Trạng thái thông báo, Trạng thái đã xem, Lý do từ chối lịch khám
⇒ Khoá chính của bảng: ID thông báo ⇒ Bảng Thông báo liên quan đến lịch khám đã ở 3NF	

3.3.2.15 Chuẩn hoá bảng Chẩn đoán

Bảng 3.15 Bảng chuẩn hoá bảng Chẩn đoán

Danh sách thuộc tính	ID chẩn đoán, ID lịch khám, Thông tin chẩn đoán
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi Chẩn đoán có một ID riêng, có duy nhất ID lịch khám, Thông tin chẩn đoán	ID chẩn đoán \rightarrow ID lịch khám, Thông tin chẩn đoán
\Rightarrow Khoá chính của bảng: ID chẩn đoán \Rightarrow Bảng Chẩn đoán đã ở 3NF	

3.3.2.16 Chuẩn hoá bảng Tin nhắn

Bảng 3.16 Bảng chuẩn hoá bảng Tin nhắn

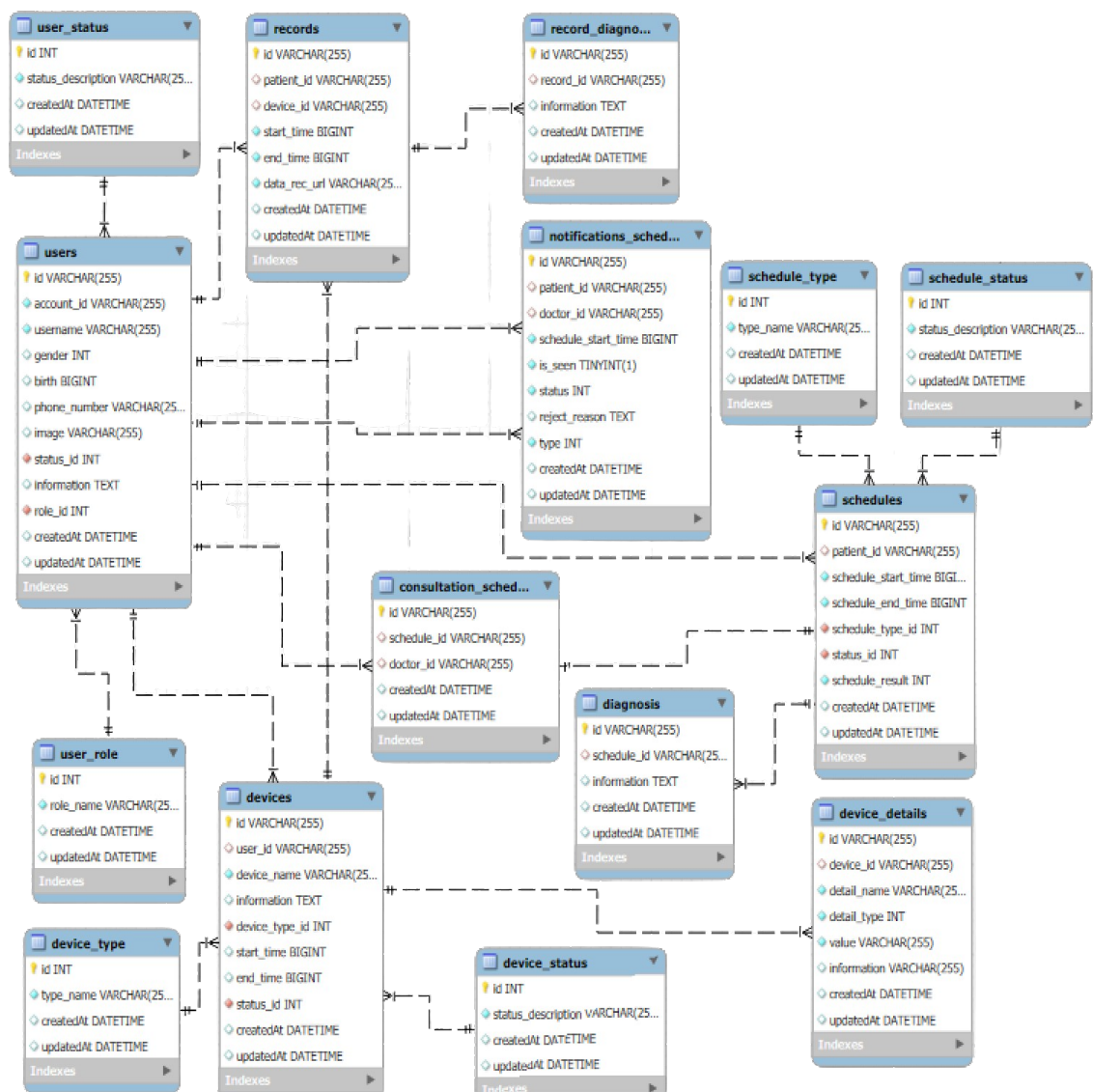
Danh sách thuộc tính	ID tin nhắn, ID người gửi, ID nhóm trò chuyện nhận tin nhắn, Nội dung tin nhắn, Thời gian gửi tin nhắn
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi Tin nhắn có một ID riêng, có duy nhất ID người gửi, ID nhóm trò chuyện nhận tin nhắn, Nội dung tin nhắn, Thời gian gửi tin nhắn	ID tin nhắn \rightarrow ID người gửi, ID nhóm trò chuyện nhận tin nhắn, Nội dung tin nhắn, Thời gian gửi tin nhắn
\Rightarrow Khoá chính của bảng: ID tin nhắn \Rightarrow Bảng Tin nhắn đã ở 3NF	

3.3.2.17 Chuẩn hoá bảng Nhóm trò chuyện

Bảng 3.17 Bảng chuẩn hoá bảng Nhóm trò chuyện

Danh sách thuộc tính	ID nhóm trò chuyện, Tên nhóm trò chuyện, Người tạo nhóm, Danh sách thành viên nhóm, Sự kiện gửi tin nhắn, Sự kiện nhận tin nhắn
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi Nhóm trò chuyện có một ID riêng, có duy nhất Tên nhóm trò chuyện, Người tạo nhóm, Danh sách thành viên nhóm, Sự kiện gửi tin nhắn, Sự kiện nhận tin nhắn	ID nhóm trò chuyện \rightarrow Tên nhóm trò chuyện, Người tạo nhóm, Danh sách thành viên nhóm, Sự kiện gửi tin nhắn, Sự kiện nhận tin nhắn
\Rightarrow Khoá chính của bảng: ID nhóm trò chuyện \Rightarrow Bảng Nhóm trò chuyện đã ở 3NF	

3.3.3 Sơ đồ ERD



Hình 3.5 Sơ đồ ERD

3.4 Thiết kế giao diện

3.5 Thiết kế các chức năng cho website và server

3.5.1 Thiết kế các API cần thiết

a) API xác minh tài khoản

Bảng 3.18 Bảng API xác minh tài khoản

Đường dẫn	Phương thức	Mô tả
api/ecg/auth/create-account	POST	Tạo mới tài khoản người dùng
api/ecg/auth/login	POST	Đăng nhập vào hệ thống
api/ecg/auth/logout	POST	Đăng xuất khỏi hệ thống

b) API quản lý người dùng trong hệ thống

Bảng 3.19 Bảng API quản lý người dùng trong hệ thống

Đường dẫn	Phương thức	Mô tả
api/ecg/users	GET	Tra cứu danh sách tất cả người dùng trong hệ thống
api/ecg/users/doctors	GET	Tra cứu danh sách toàn bộ bác sĩ trong hệ thống
api/ecg/users/:id	GET	Tra cứu dữ liệu người dùng cụ thể dựa trên id tương ứng
api/ecg/users/data/patient-data	GET	Tra cứu danh sách các bệnh nhân đang được theo dõi của bác sĩ cụ thể
api/ecg/users/	PUT	Chỉnh sửa thông tin người dùng
api/ecg/users/:userId	DELETE	Xóa người dùng cụ thể dựa trên id tương ứng

c) API quản lý thiết bị y tế

Bảng 3.20 Bảng API quản lý thiết bị y tế

Đường dẫn	Phương thức	Mô tả
api/ecg/device	GET	Tra cứu danh sách toàn bộ thiết bị y tế trong hệ thống
api/ecg/device/:id	GET	Tra cứu thông tin chi tiết của một thiết bị y tế dựa trên id tương ứng
api/ecg/device/add	POST	Thêm mới thiết bị y tế
api/ecg/device-detail	POST	Thêm thông số kỹ thuật cho một thiết bị y tế
api/ecg/device/:deviceId	PUT	Cập nhật thông tin chi tiết cho thiết bị y tế cụ thể
api/ecg/device/:deviceId	DELETE	Xóa thiết bị y tế dựa trên id tương ứng
api/ecg/device-detail	PUT	Cập nhật thông số kỹ thuật cho thiết bị y tế
api/ecg/device-detail/:detailId	PUT	Xoá thông số kỹ thuật của thiết bị y tế

d) API quản lý dữ liệu phiên đo

Bảng 3.21 Bảng API quản lý dữ liệu phiên đo

Đường dẫn	Phương thức	Mô tả
api/ecg/records	GET	Tra cứu dữ liệu tất cả các phiên đo
api/ecg/records/:id	GET	Tra cứu dữ liệu phiên đo cụ thể dựa theo id tương ứng
api/ecg/records/data/doctorId	GET	Tra cứu dữ liệu tất cả các phiên đo của bệnh nhân mà bác sĩ phụ trách
api/ecg/records/	POST	Tạo dữ liệu phiên đo mới
api/ecg/records/	PUT	Cập nhật dữ liệu phiên đo
api/ecg/records/:recordId	DELETE	Xóa dữ liệu phiên đo dựa theo id tương ứng

e) API quản lý dịch vụ lịch khám

Bảng 3.22 Bảng API quản lý dịch vụ lịch khám

Đường dẫn	Phương thức	Mô tả
api/ecg/schedules	GET	Tra cứu danh sách tất cả lịch khám trong hệ thống
api/ecg/schedules/doctorId	GET	Tra cứu danh sách lịch khám của bác sĩ cụ thể
api/ecg/schedules/patientId	GET	Tra cứu danh sách lịch khám của bệnh nhân cụ thể
api/ecg/schedules/create-by-doctor	POST	Cho phép bác sĩ đặt lịch tái khám cho bệnh nhân
api/ecg/schedules/create-by-patient	POST	Cho phép bệnh nhân đặt lịch khám với bác sĩ
api/ecg/schedules/time/available-doctor/:schedule-time	GET	Tra cứu danh sách các bác sĩ khả dụng theo thời gian đã chọn
api/ecg/schedules/available-schedule/:id	GET	Tra cứu các khung giờ trống có thể đặt lịch với bác sĩ cụ thể.
api/ecg/schedules/accept-schedule	PUT	Chấp nhận lịch khám cụ thể
api/ecg/schedules/reject-schedule/:id	DELETE	Từ chối lịch khám cụ thể

f) API liên quan đến chẩn đoán cho bệnh nhân

Bảng 3.23 Bảng API liên quan đến chẩn đoán cho bệnh nhân

Đường dẫn	Phương thức	Mô tả
api/ecg/diagnosis	POST	Tạo chẩn đoán mới cho bệnh nhân
api/ecg/diagnosis/ schedule/:scheduleId	POST	Tra cứu thông tin chẩn đoán dựa trên id lịch khám tương ứng
api/ecg/diagnosis/update	POST	Cập nhật thông tin chẩn đoán

g) API liên quan đến thông báo về lịch khám

Bảng 3.24 Bảng API liên quan đến thông báo về lịch khám

Đường dẫn	Phương thức	Mô tả
api/ecg/notification/get	GET	Tra cứu tất cả các thông báo của người dùng cụ thể
api/ecg/notification	POST	Tạo thông báo mới liên quan đến lịch khám
api/ecg/notification/ update-seen	POST	Cập nhật trạng thái thông báo đã được xem
api/ecg/notification/:id	DELETE	Xóa thông báo dựa trên id tương ứng

h) API liên quan đến tin nhắn

Bảng 3.25 Bảng API liên quan đến tin nhắn

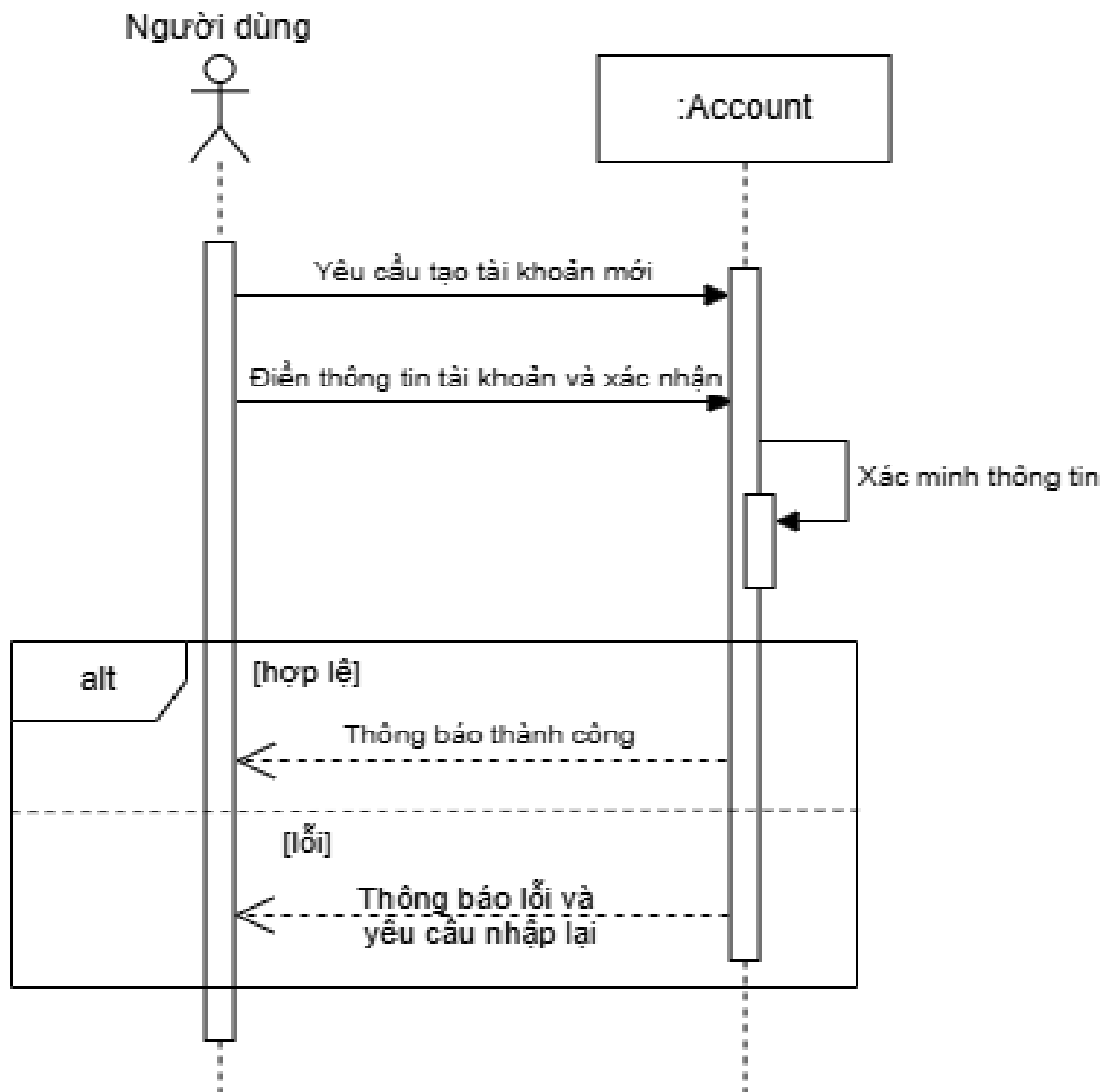
Đường dẫn	Phương thức	Mô tả
api/ecg/groupChat	POST	Tạo nhóm trò chuyện mới
api/ecg/groupChat	GET	Tra cứu danh sách các nhóm trò chuyện của người dùng
api/ecg/chat/messages/ groupChatId	GET	Tra cứu lịch sử trò chuyện của các đoạn hội thoại đã thực hiện
api/ecg/chat/send	POST	Cho phép người dùng gửi tin nhắn đến các đối tượng liên quan

3.5.2 Sơ đồ tuần tự API

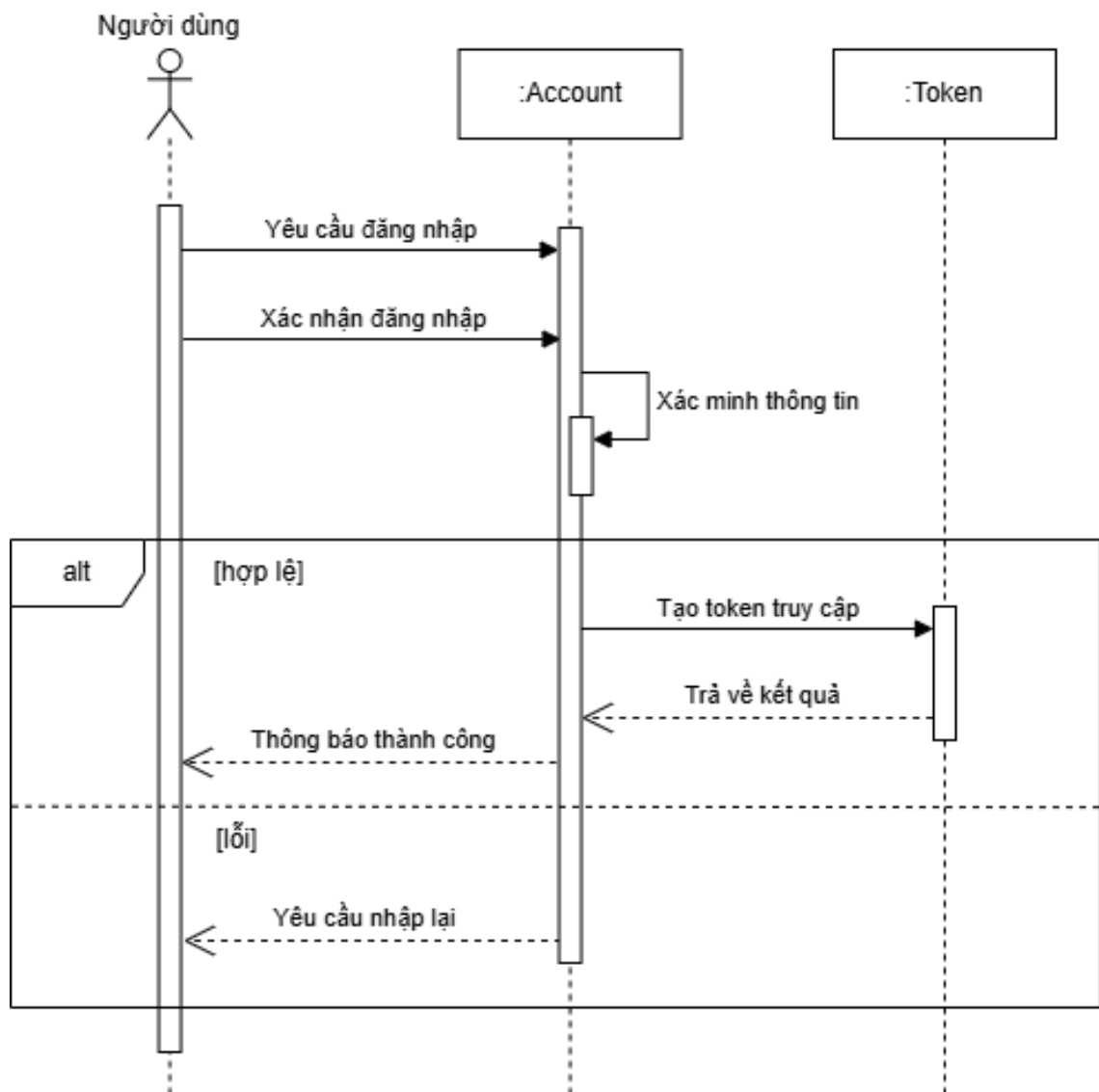
Phần này cung cấp các sơ đồ tuần tự minh họa chi tiết cách thức hoạt động của các API trong hệ thống. Dựa trên bảng API đã thiết kế, các sơ đồ tuần tự sẽ mô phỏng luồng

xử lý từ khi nhận yêu cầu đến khi trả về kết quả cho người dùng, cho thấy rõ sự tương tác giữa các thành phần và lớp chức năng trong hệ thống.

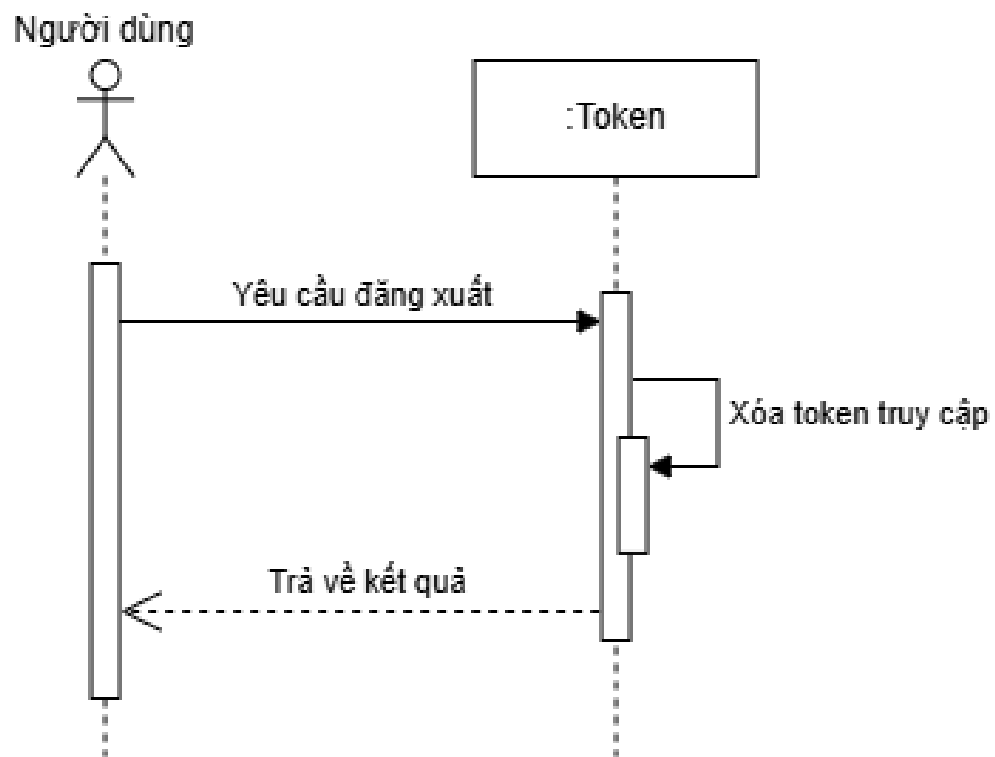
3.5.2.1 Các API phục vụ mục đích xác minh tài khoản



Hình 3.6 Sơ đồ tuần tự API tạo mới tài khoản người dùng

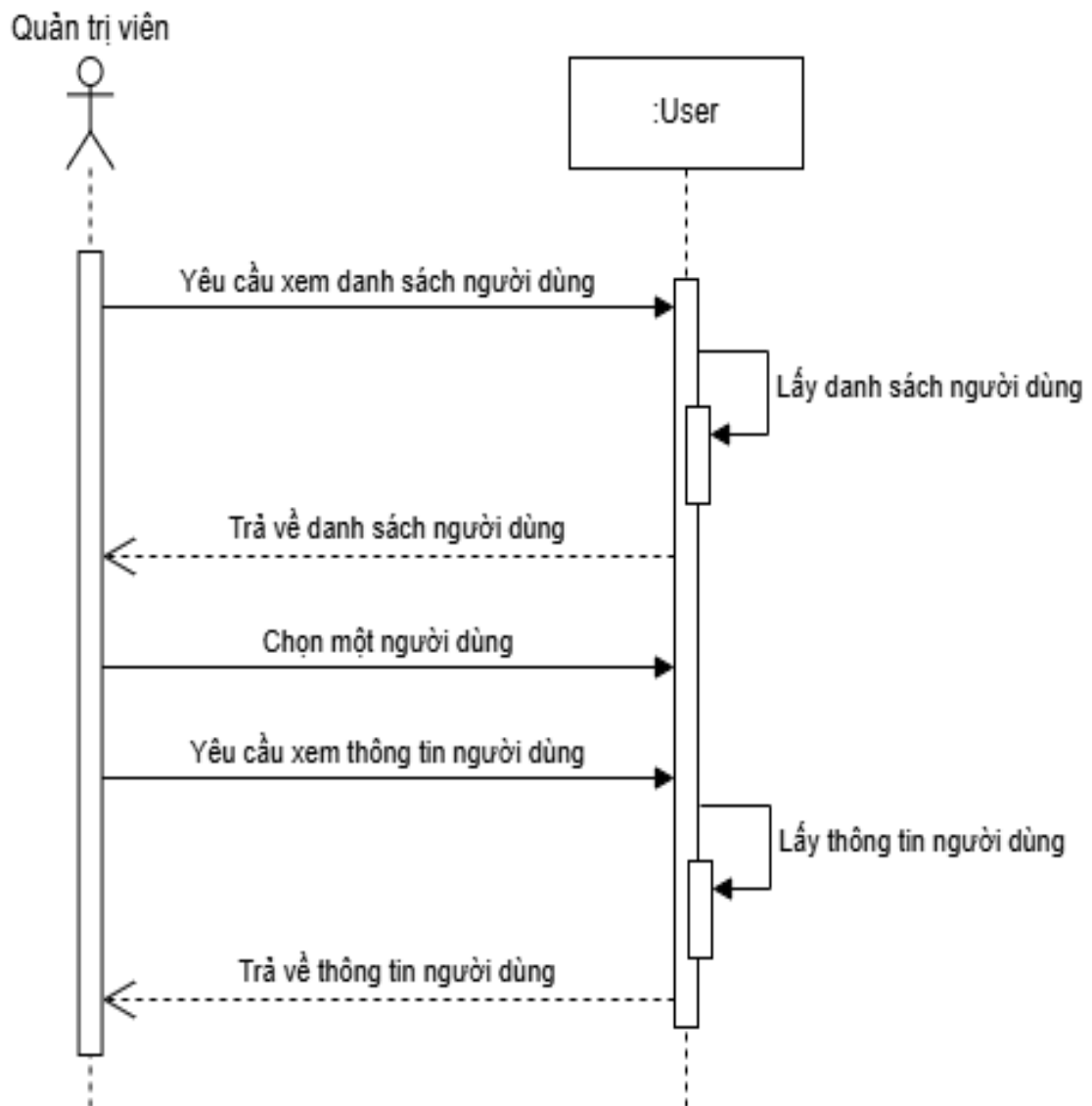


Hình 3.7 Sơ đồ tuần tự API đăng nhập vào hệ thống

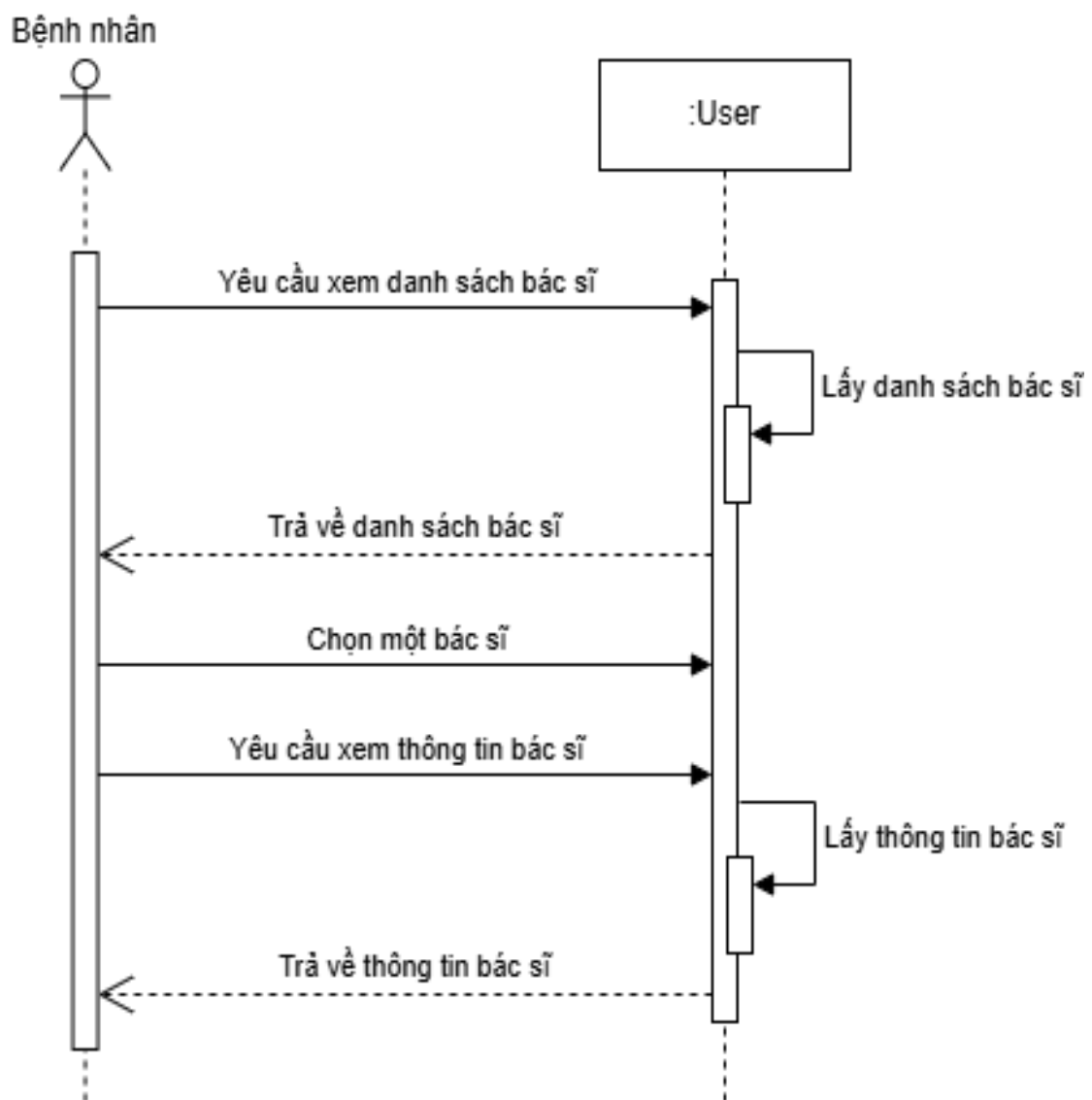


Hình 3.8 Sơ đồ tuần tự API đăng xuất khỏi hệ thống

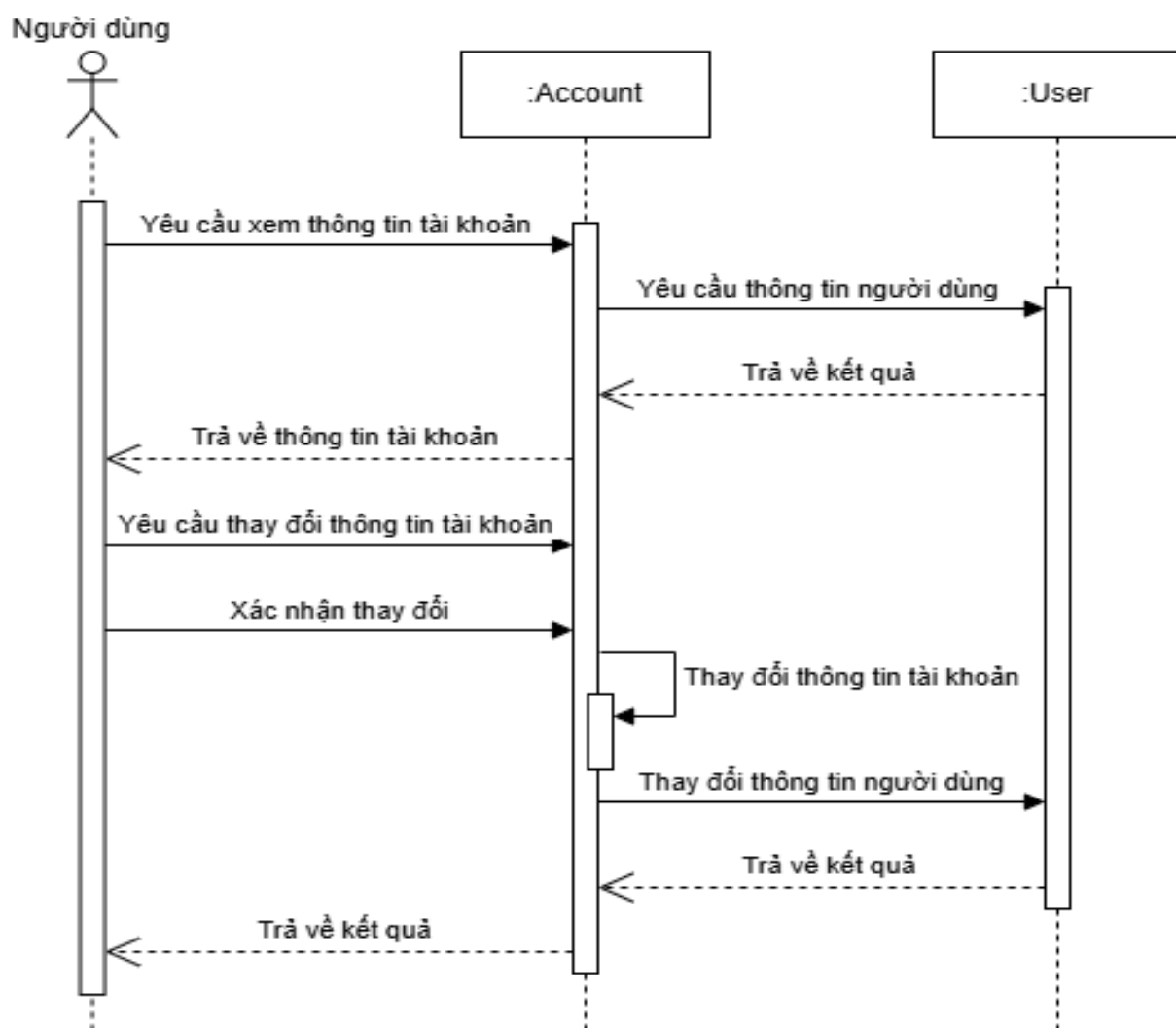
3.5.2.2 Các API phục vụ mục đích quản lý người dùng trong hệ thống



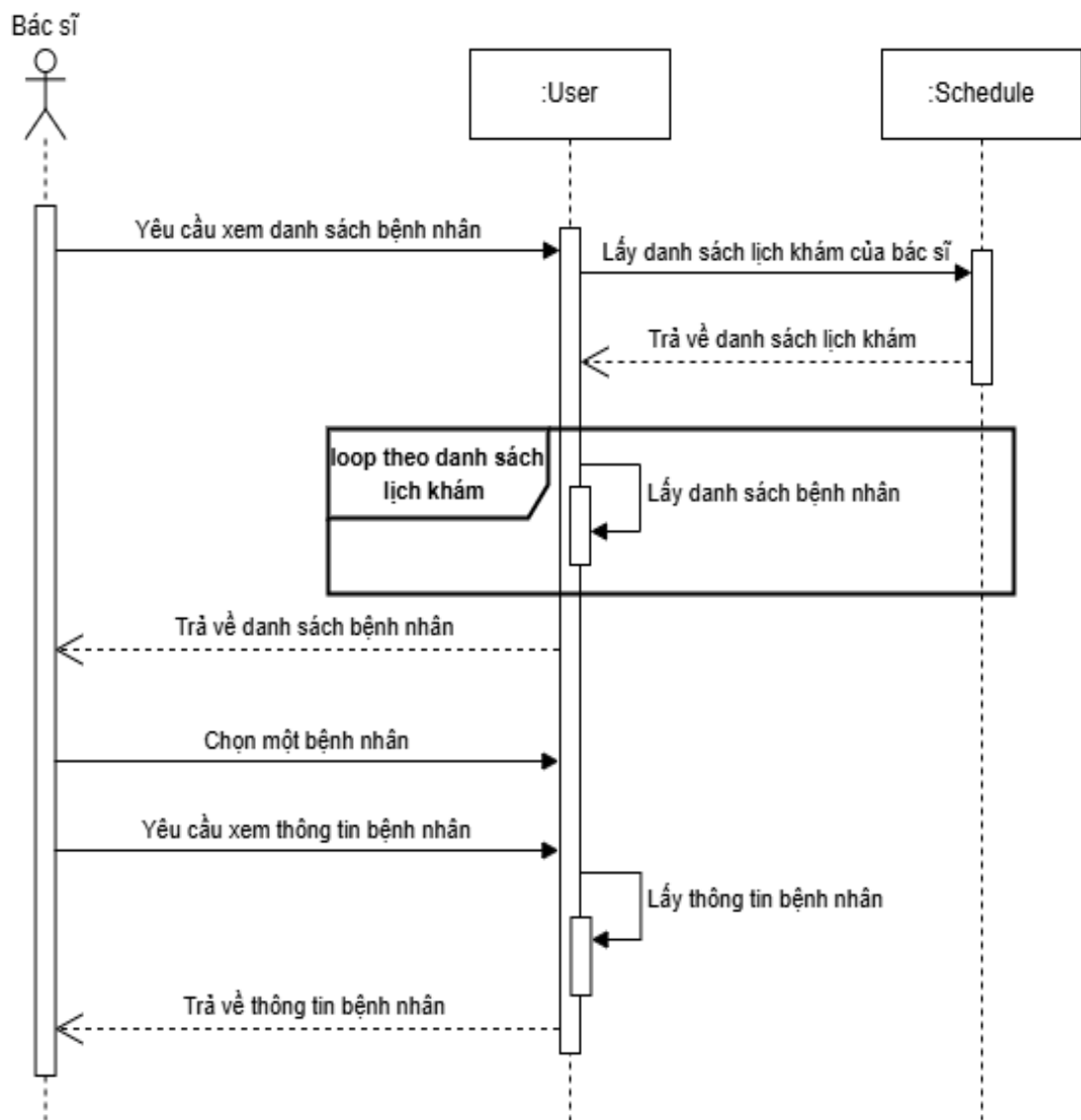
Hình 3.9 Sơ đồ tuần tự API tra cứu danh sách tất cả người dùng trong hệ thống



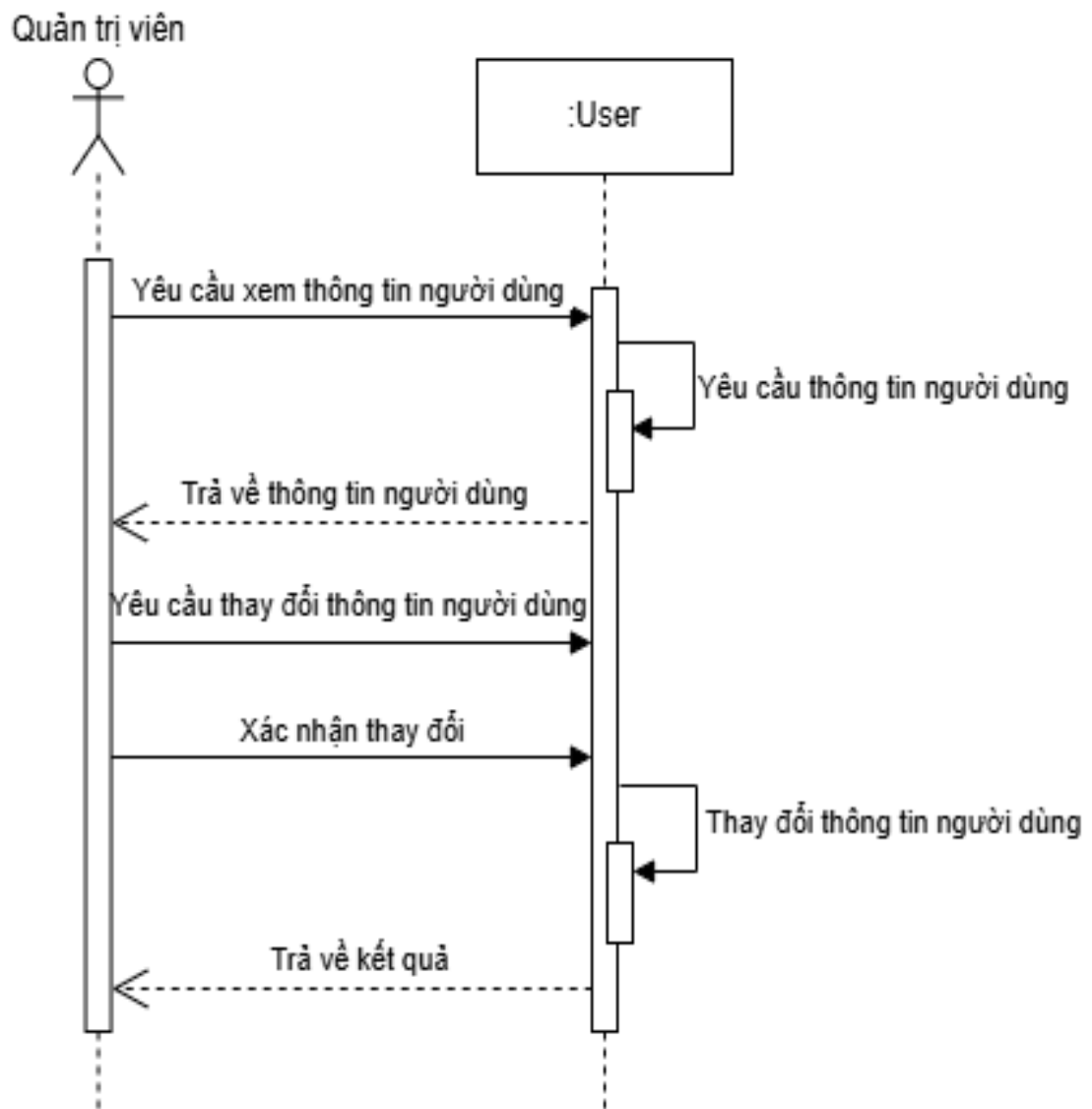
Hình 3.10 Sơ đồ tuần tự API tra cứu danh sách toàn bộ bác sĩ trong hệ thống



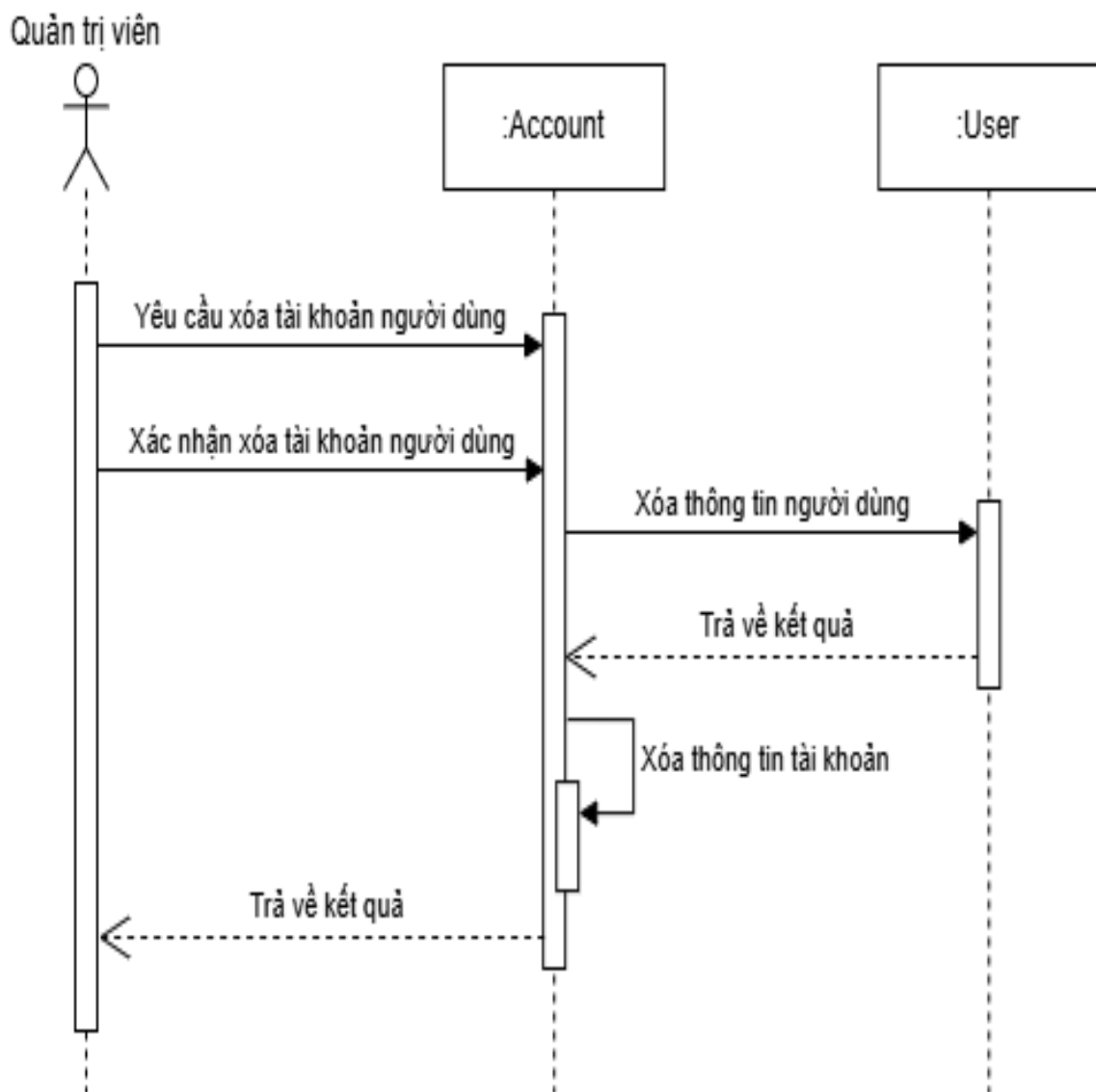
Hình 3.11 Sơ đồ tuần tự API tra cứu dữ liệu người dùng cụ thể dựa trên id



Hình 3.12 Sơ đồ tuần tự API tra cứu danh sách bệnh nhân đang được theo dõi

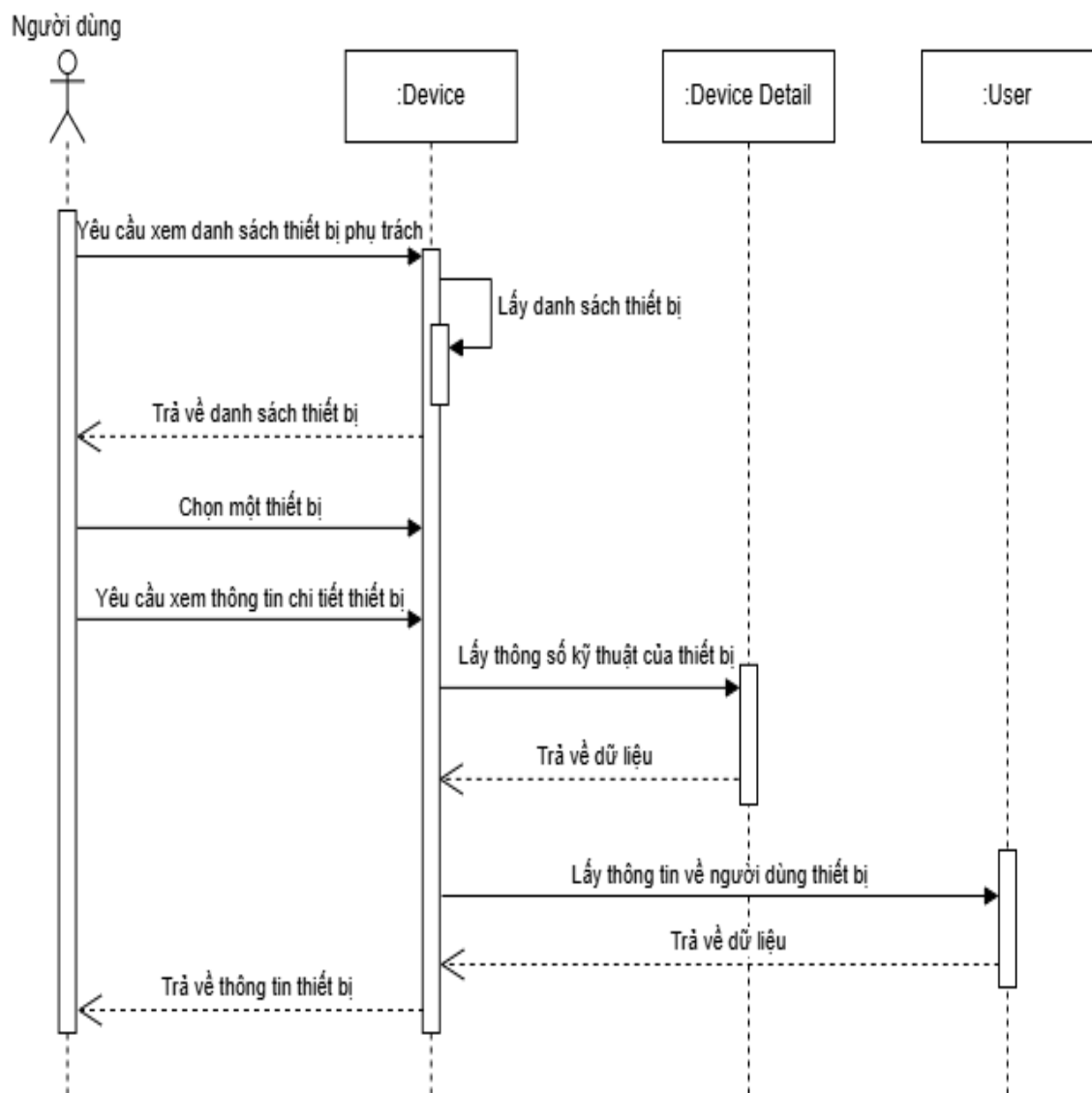


Hình 3.13 Sơ đồ tuần tự API chỉnh sửa thông tin người dùng

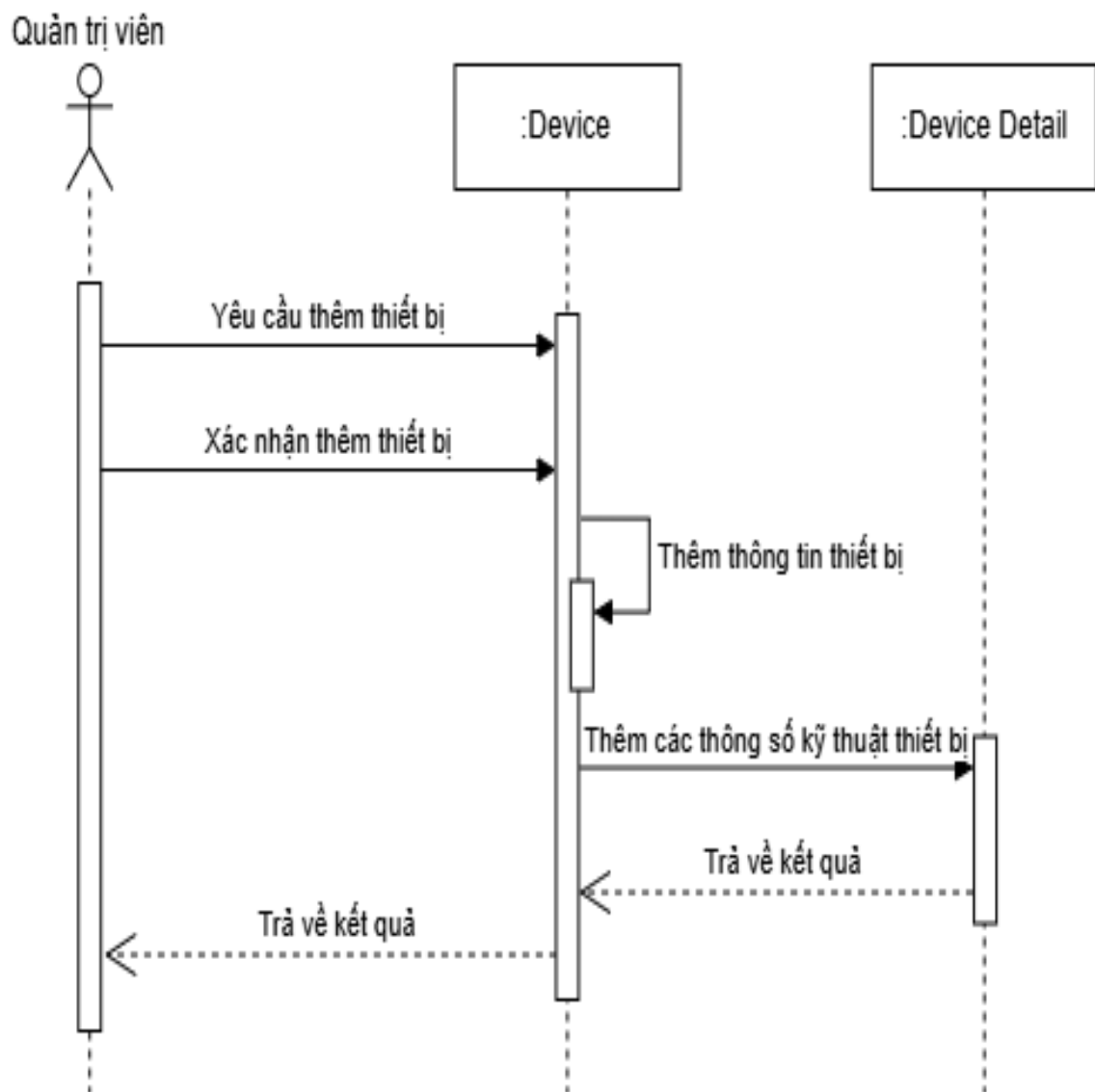


Hình 3.14 Sơ đồ tuần tự API xóa người dùng cụ thể dựa trên id

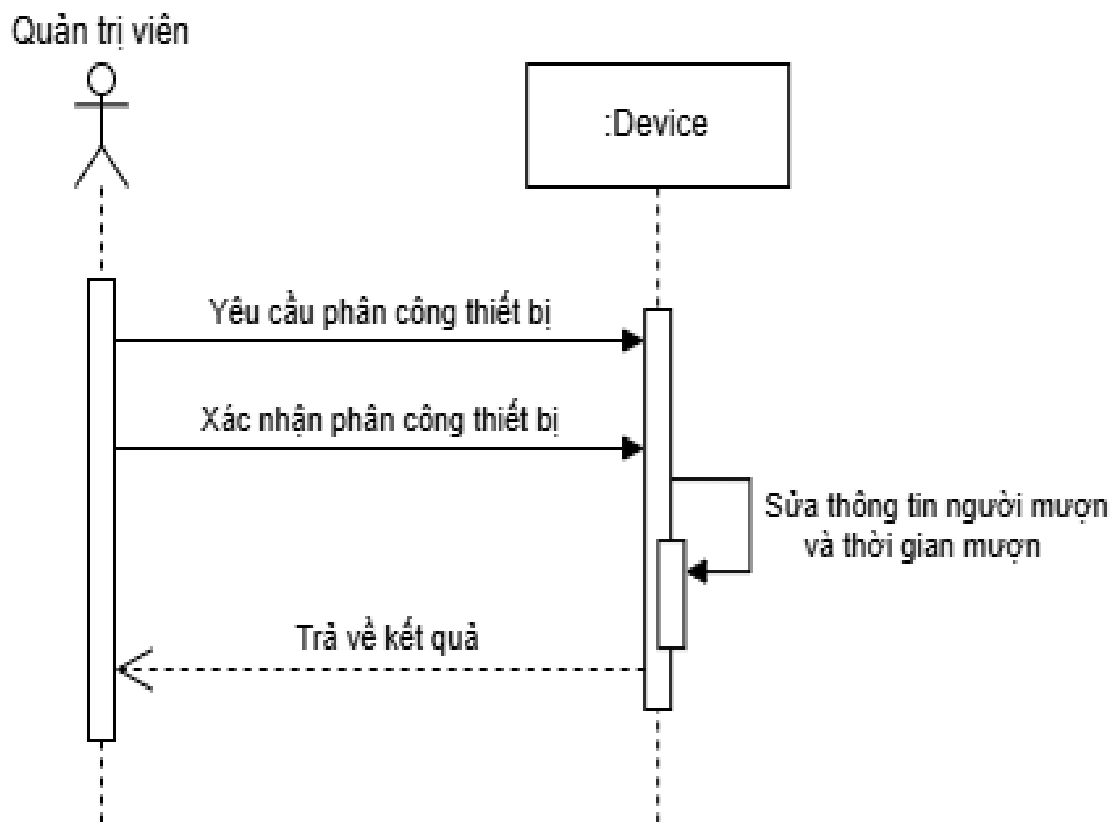
3.5.2.3 Các API phục vụ mục đích quản lý thiết bị y tế



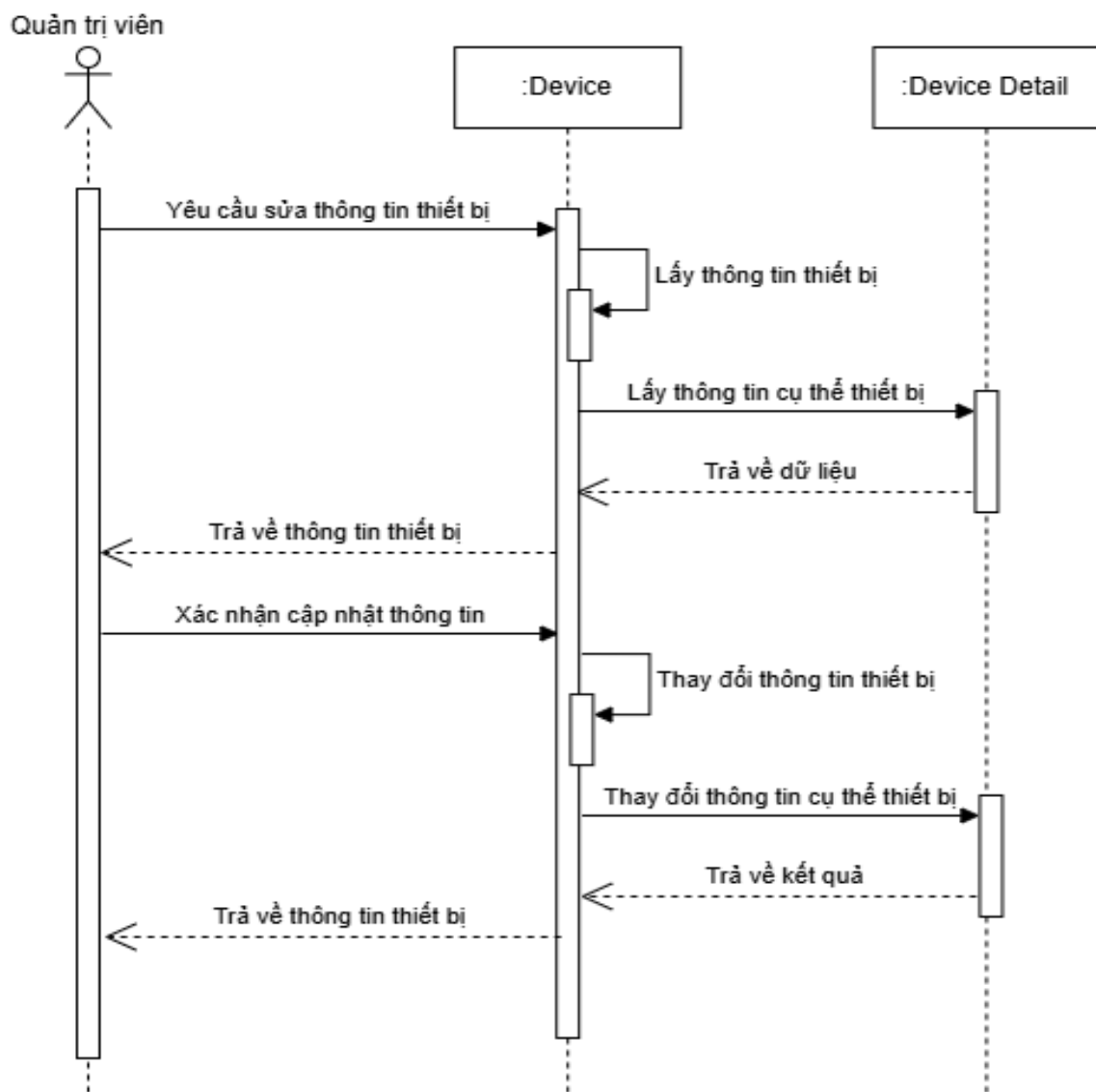
Hình 3.15 Sơ đồ tuần tự API tra cứu danh sách toàn bộ thiết bị y tế



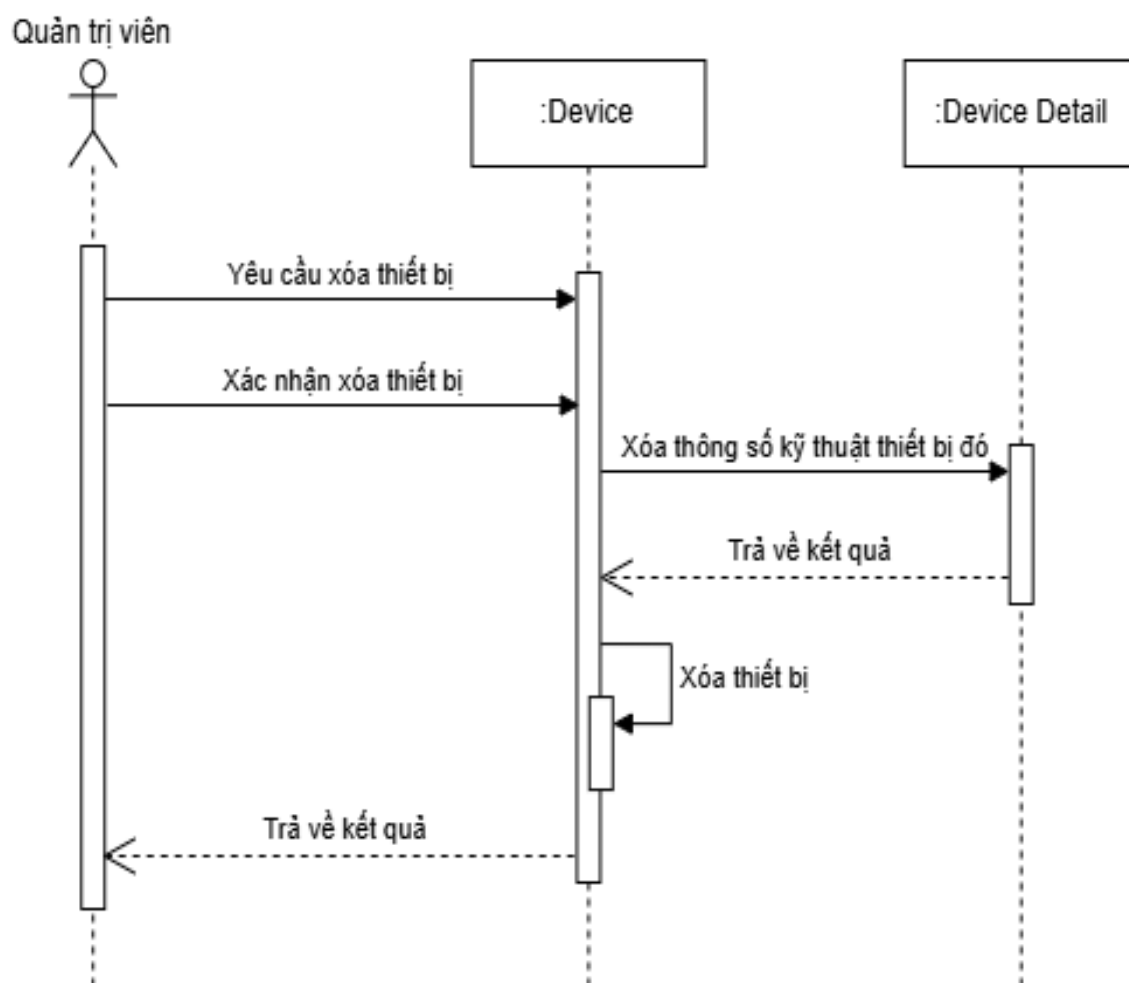
Hình 3.16 Sơ đồ tuần tự API thêm mới thiết bị y tế



Hình 3.17 Sơ đồ tuần tự API đăng ký mượn thiết bị

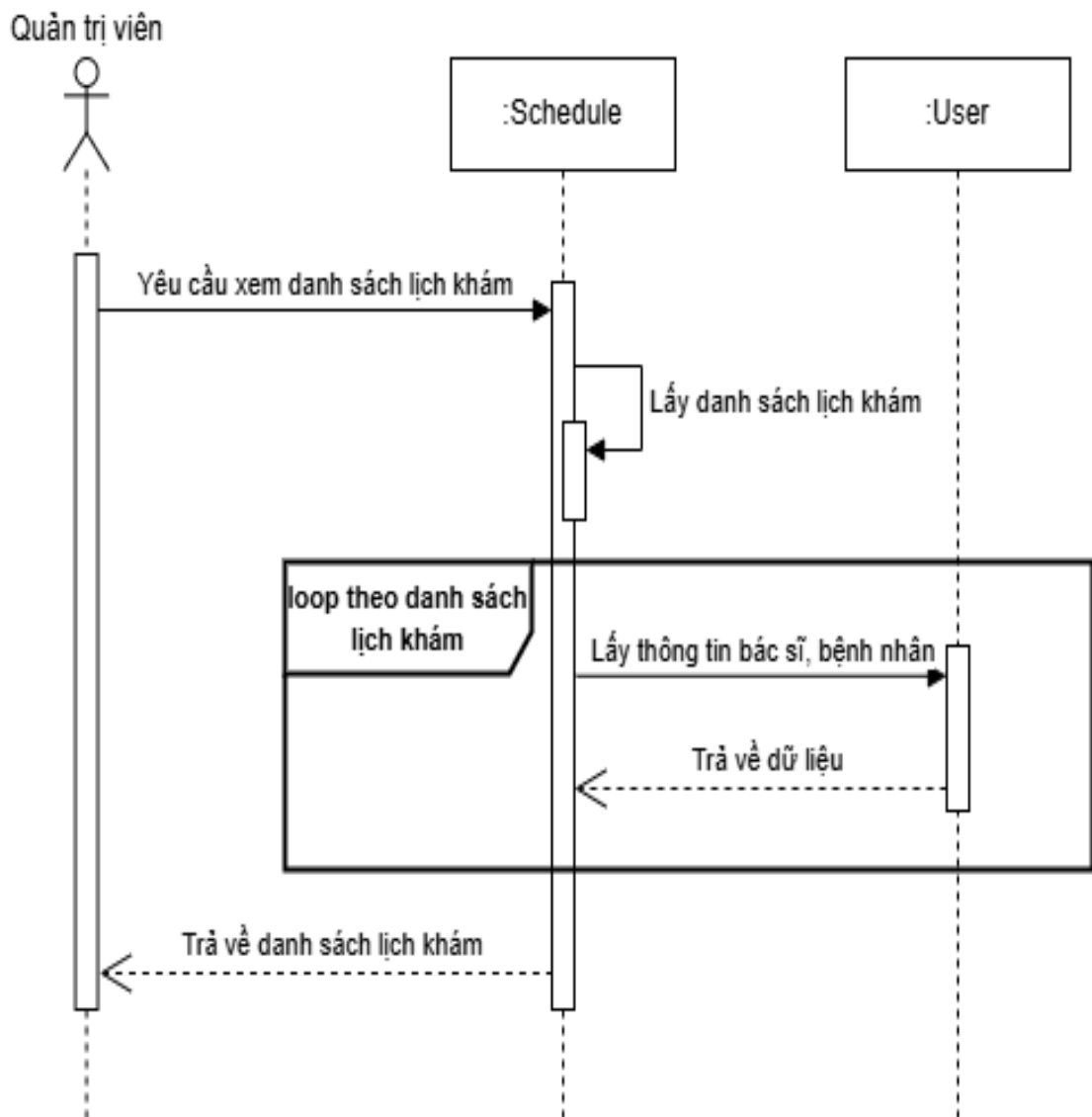


Hình 3.18 Sơ đồ tuần tự API Cập nhật thiết bị và thông số kỹ thuật (nếu cần)

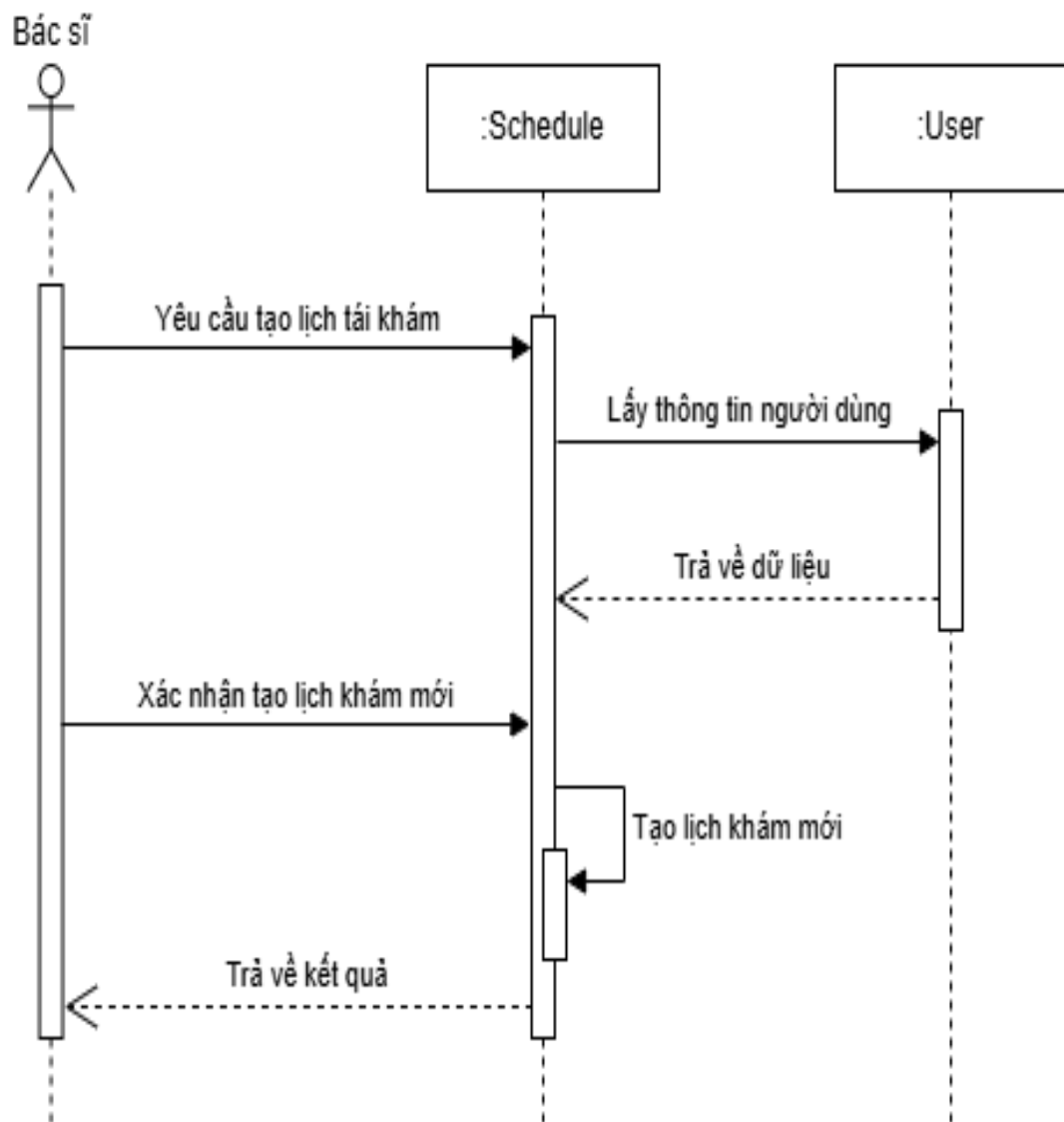


Hình 3.19 Sơ đồ tuần tự API xóa thiết bị y tế

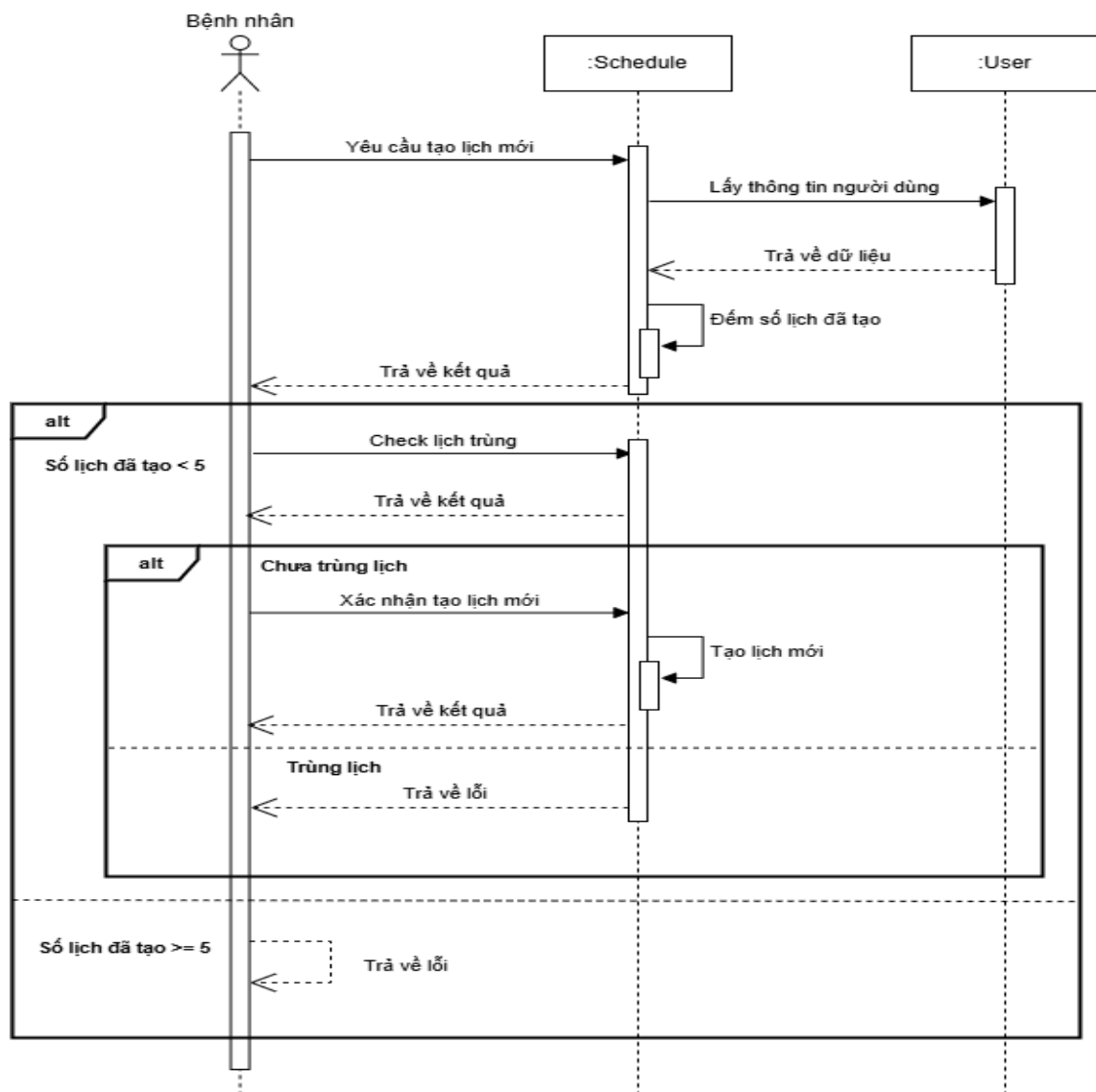
3.5.2.4 Các API phục vụ mục đích đặt lịch hẹn bác sĩ - bệnh nhân



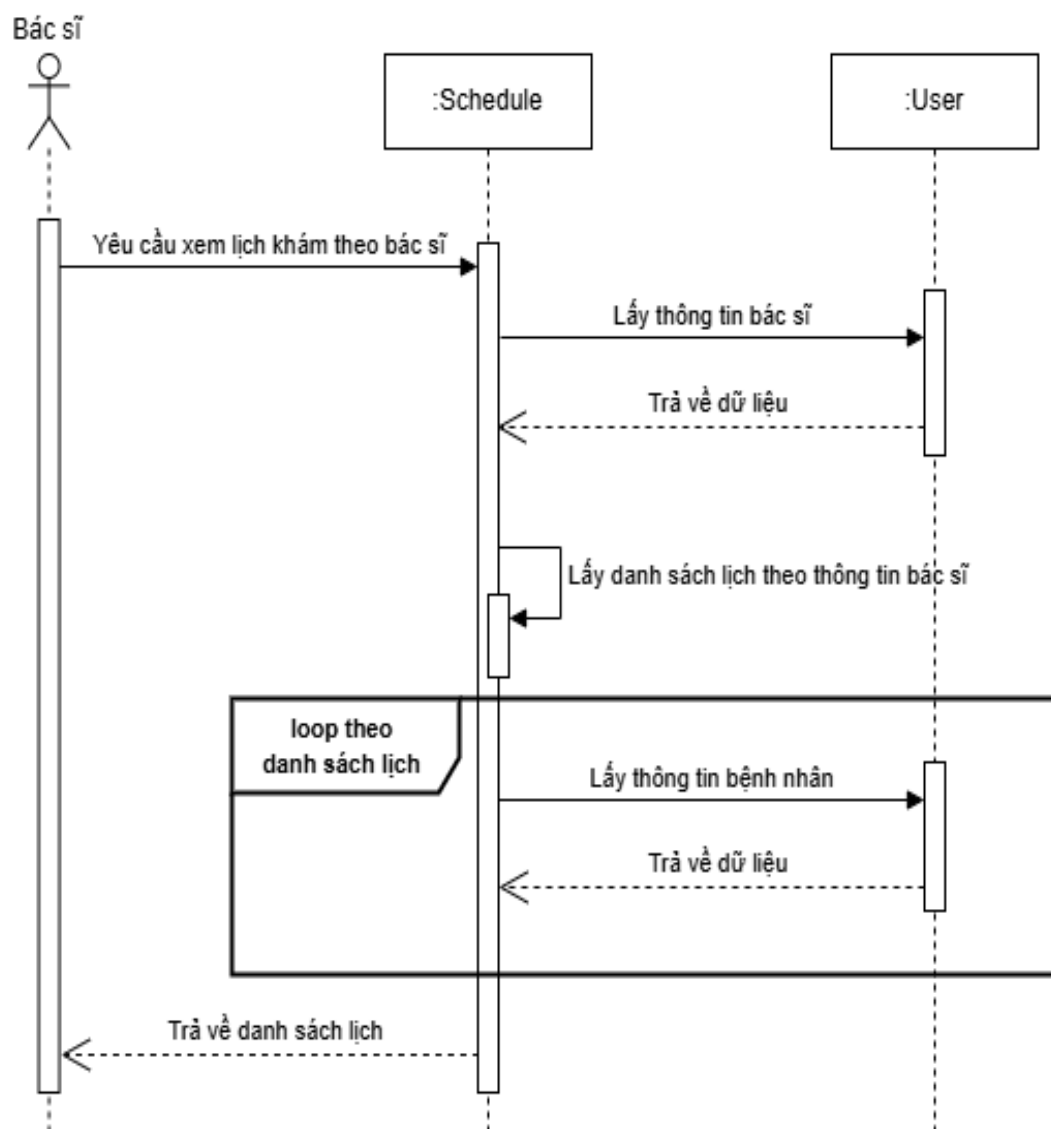
Hình 3.20 Sơ đồ tuần tự API tra cứu tất cả lịch hẹn



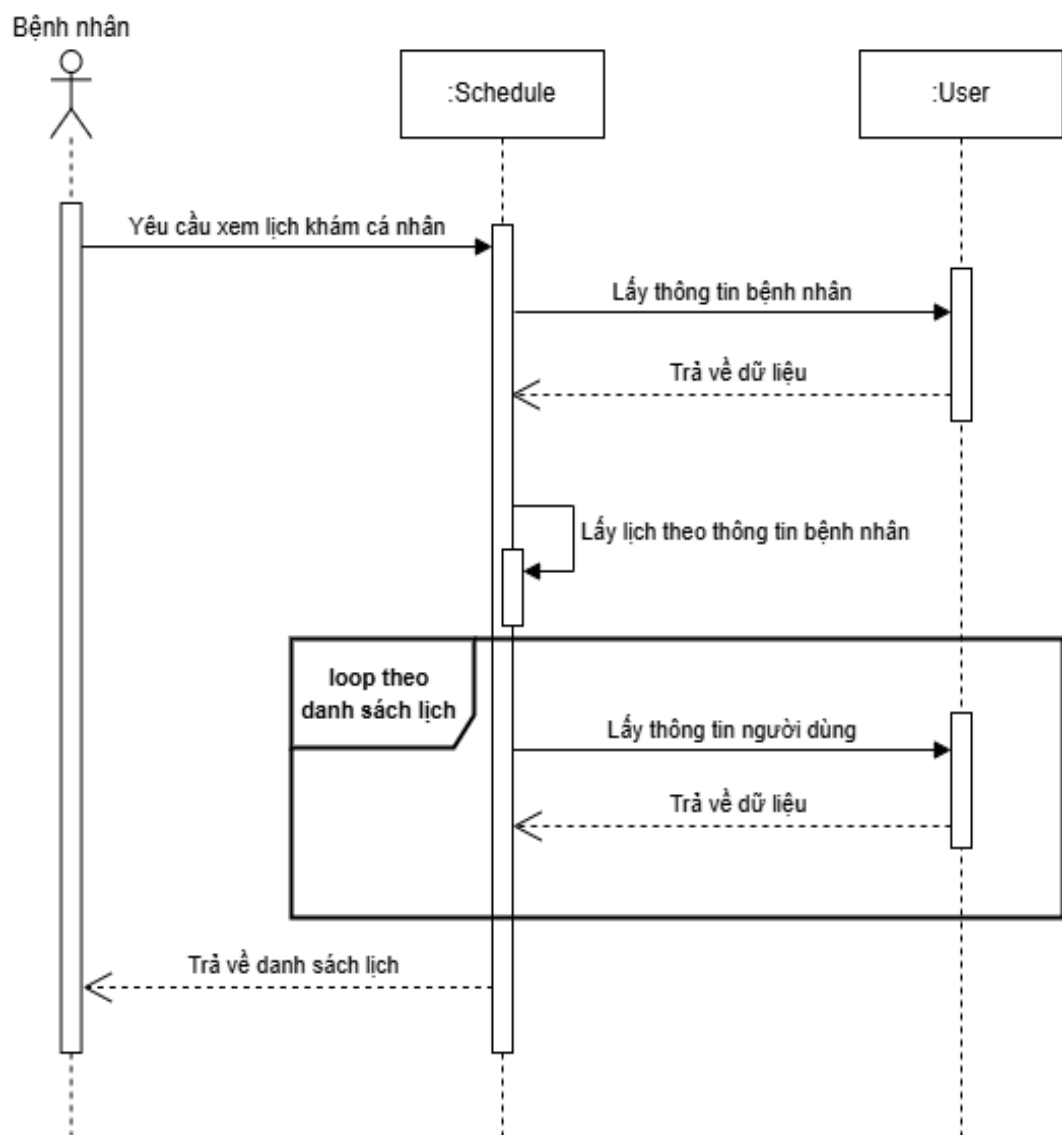
Hình 3.21 Sơ đồ tuần tự API bác sĩ tạo lịch tái khám cho bệnh nhân



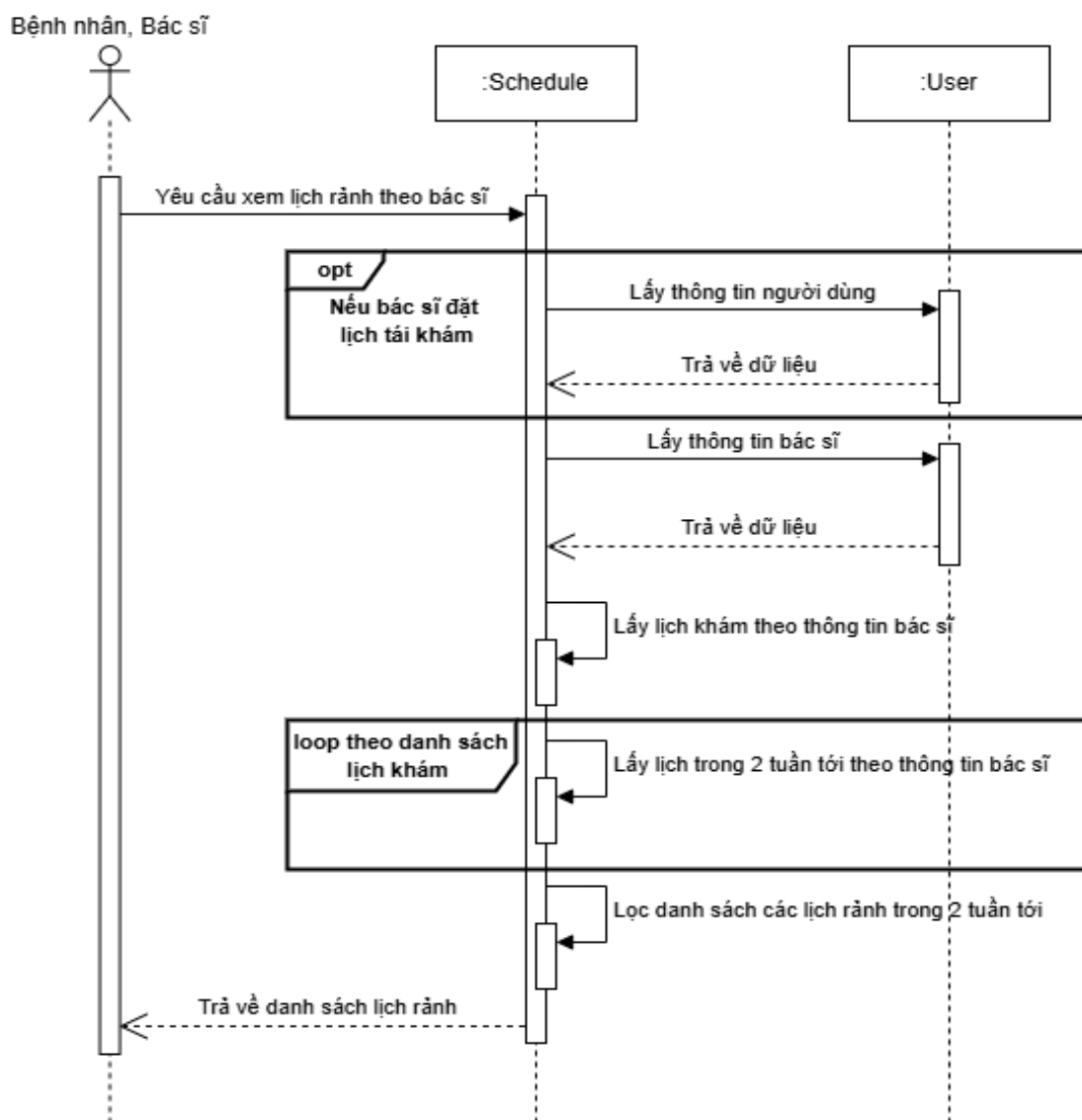
Hình 3.22 Sơ đồ tuần tự API bệnh nhân chủ động đặt lịch hẹn



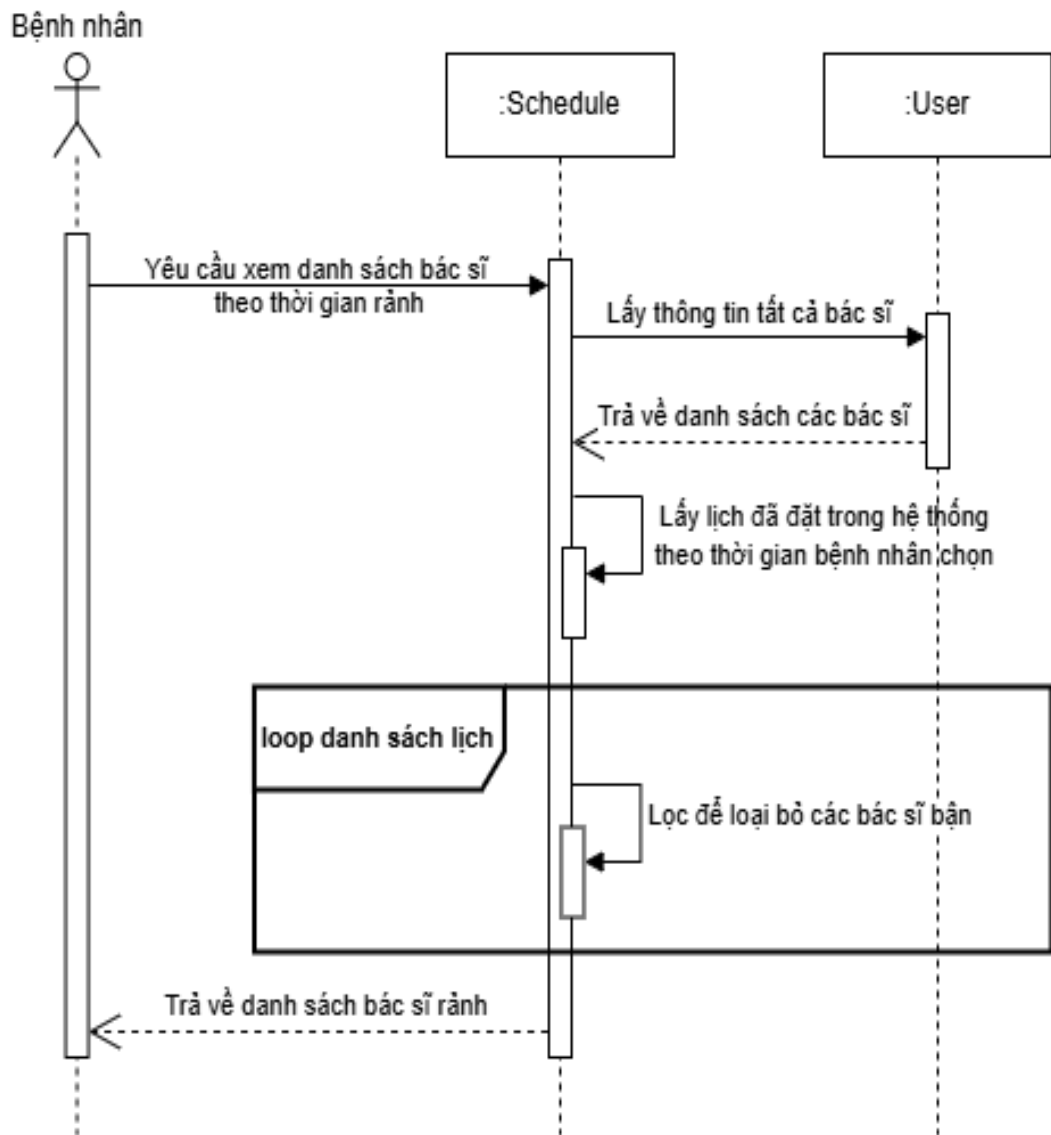
Hình 3.23 Sơ đồ tuần tự API tra cứu lịch hẹn của bác sĩ cụ thể



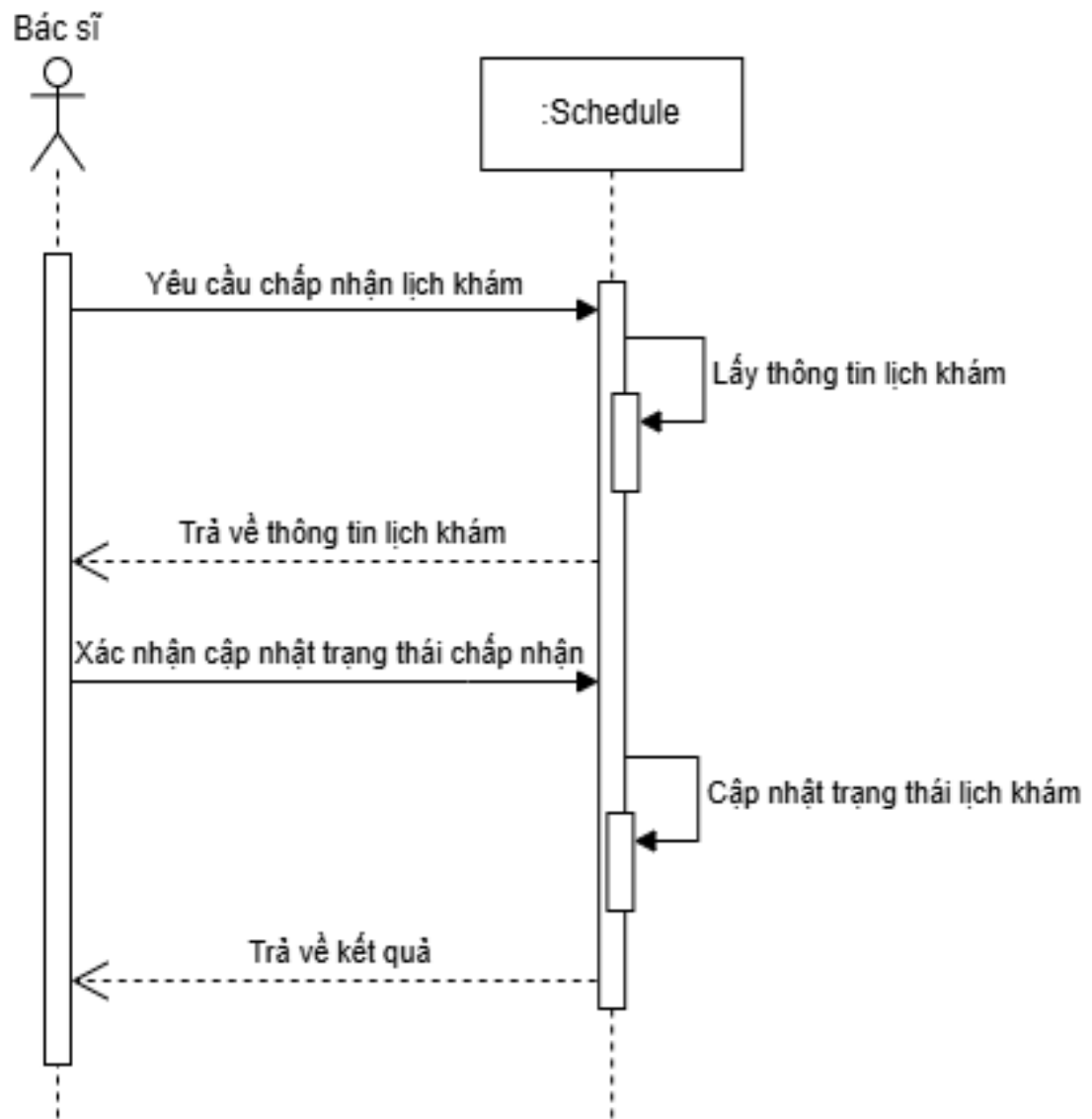
Hình 3.24 Sơ đồ tuần tự API tra cứu lịch hẹn của bệnh nhân cụ thể



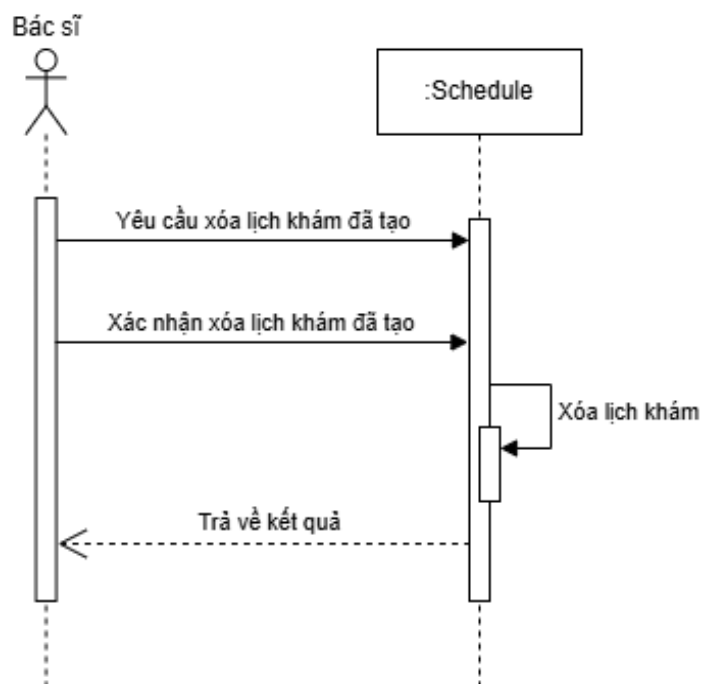
Hình 3.25 Sơ đồ tuần tự API tra cứu lịch rảnh của bác sĩ cụ thể



Hình 3.26 Sơ đồ tuần tự API tra cứu bác sĩ phù hợp với thời gian đã chọn

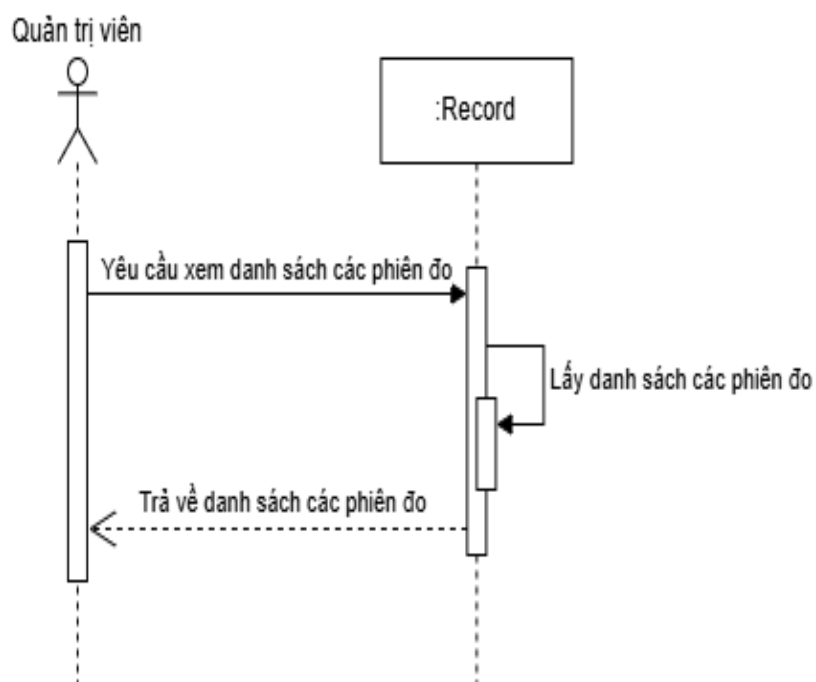


Hình 3.27 Sơ đồ tuần tự API bác sĩ xác nhận lịch hẹn

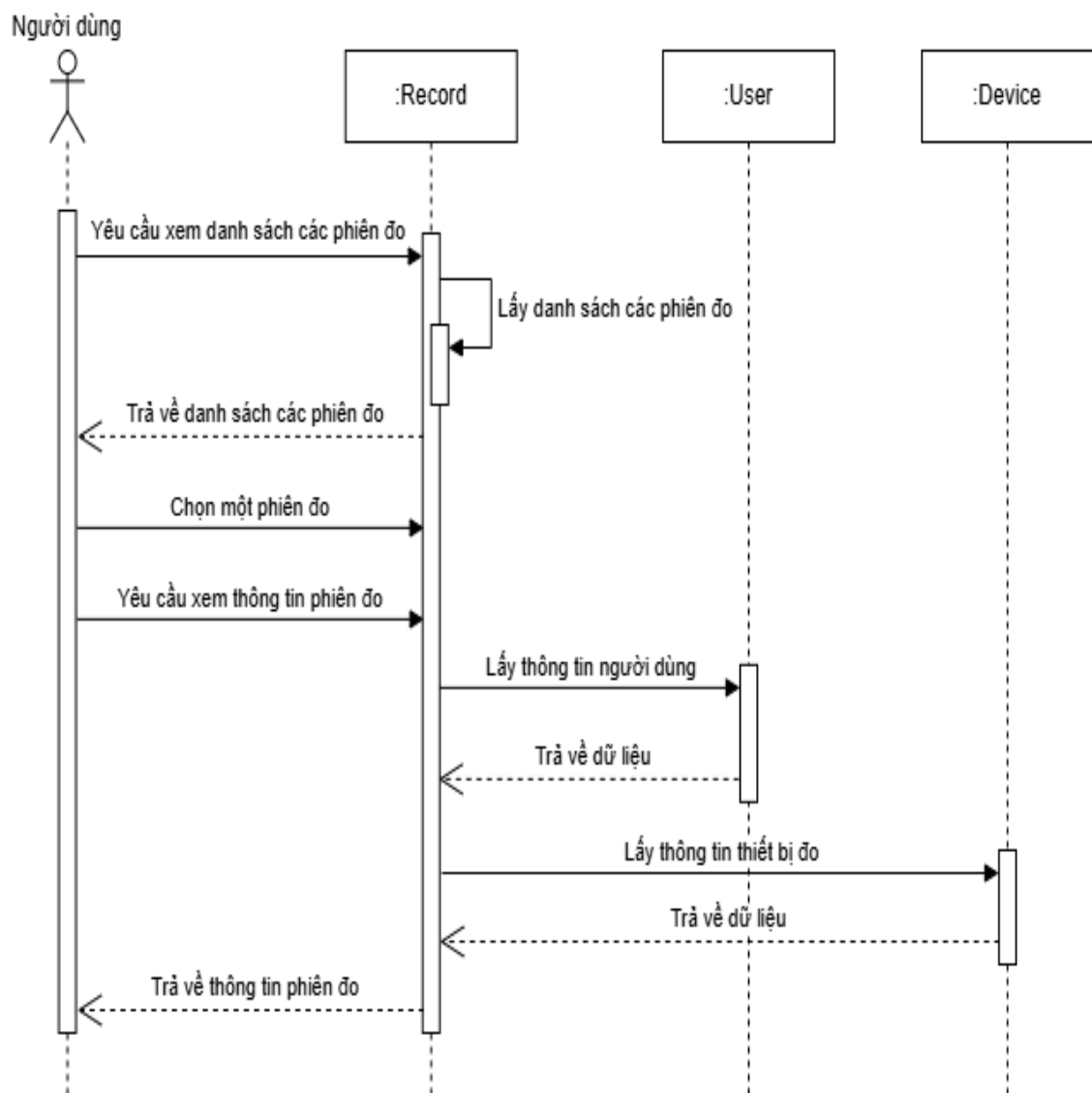


Hình 3.28 Sơ đồ tuần tự API bác sĩ từ chối lịch hẹn

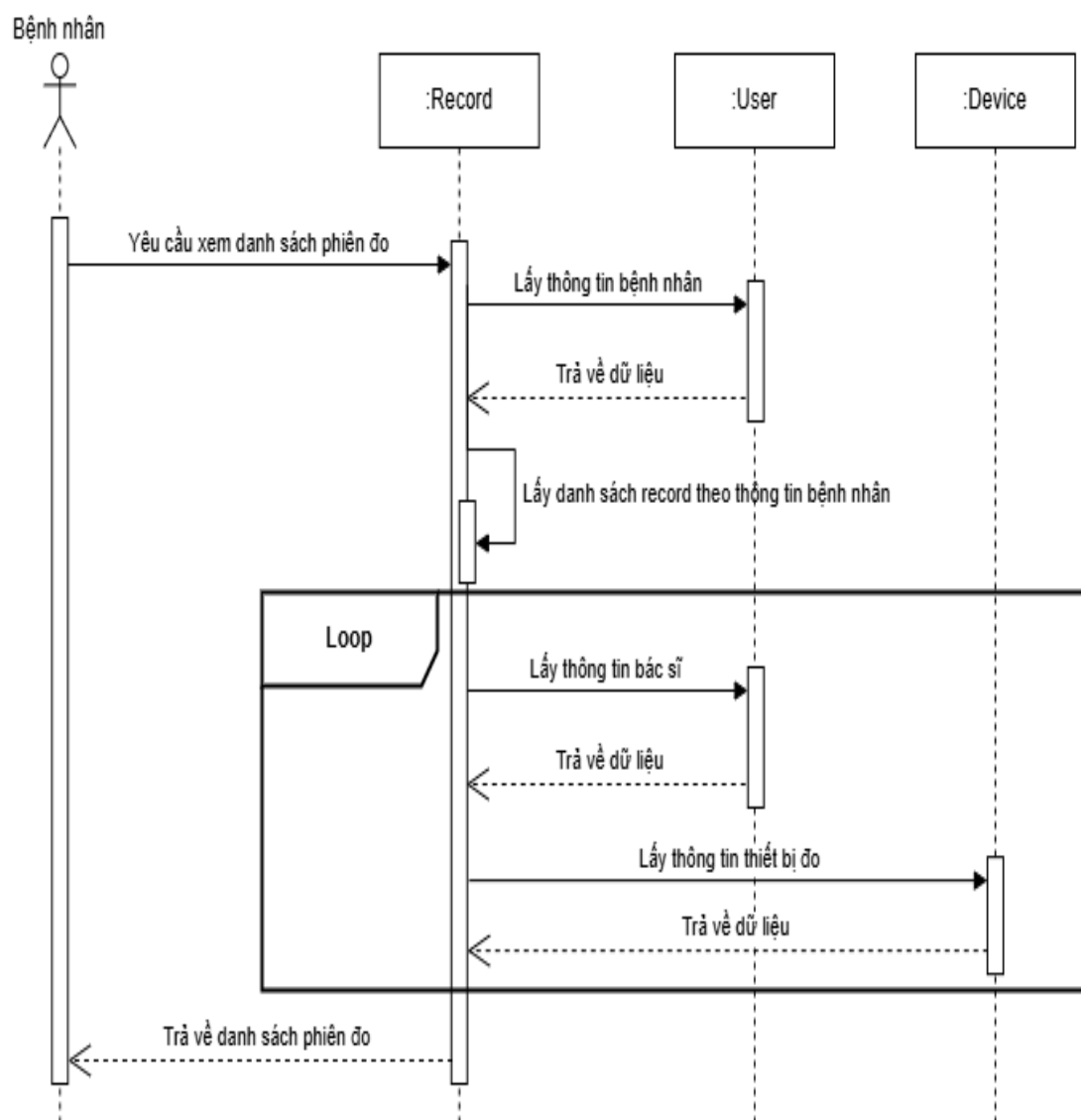
3.5.2.5 Các API phục vụ mục đích liên quan đến dữ liệu phiên đo



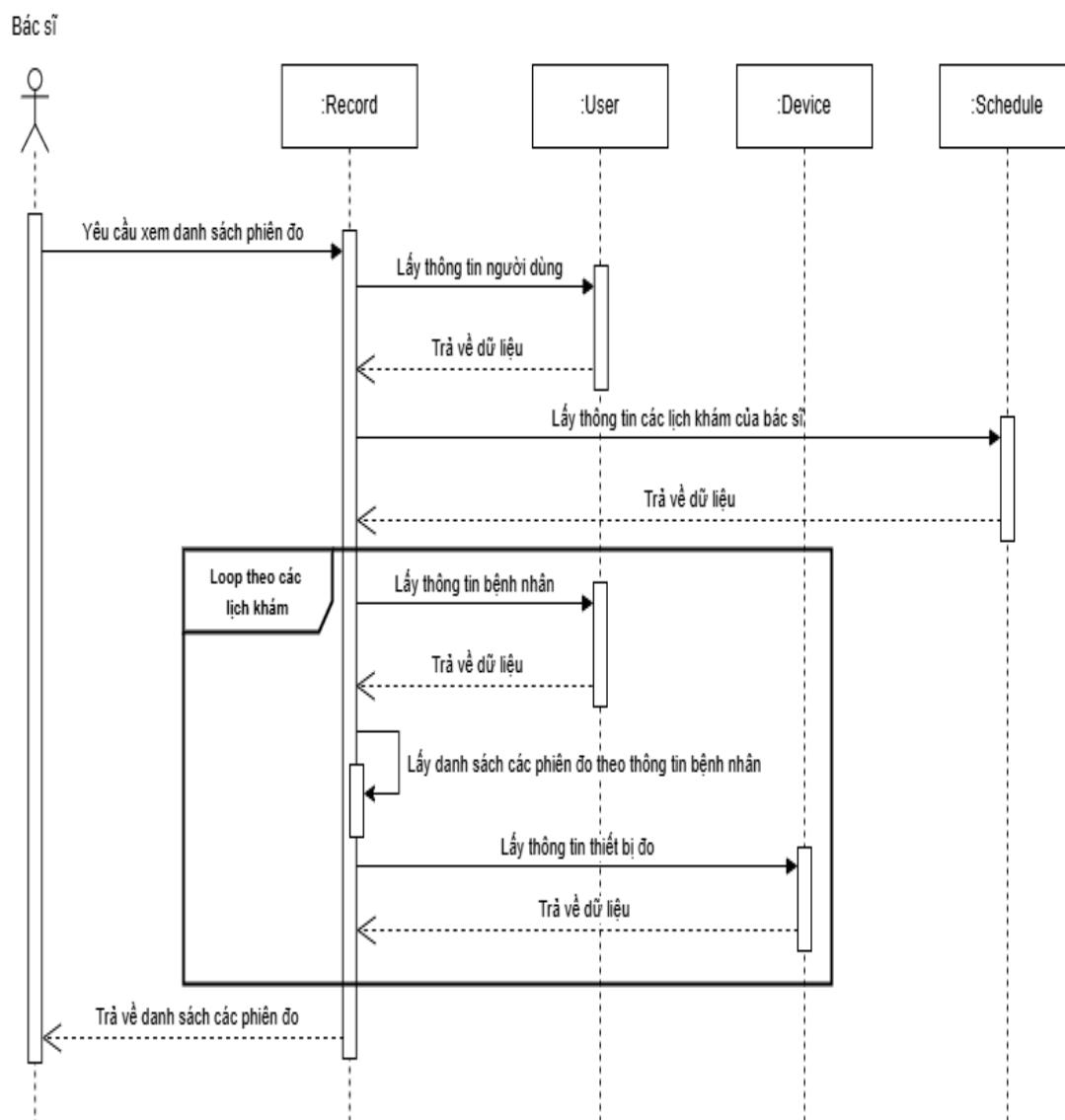
Hình 3.29 Sơ đồ tuần tự API tra cứu danh sách các dữ liệu phiên đo



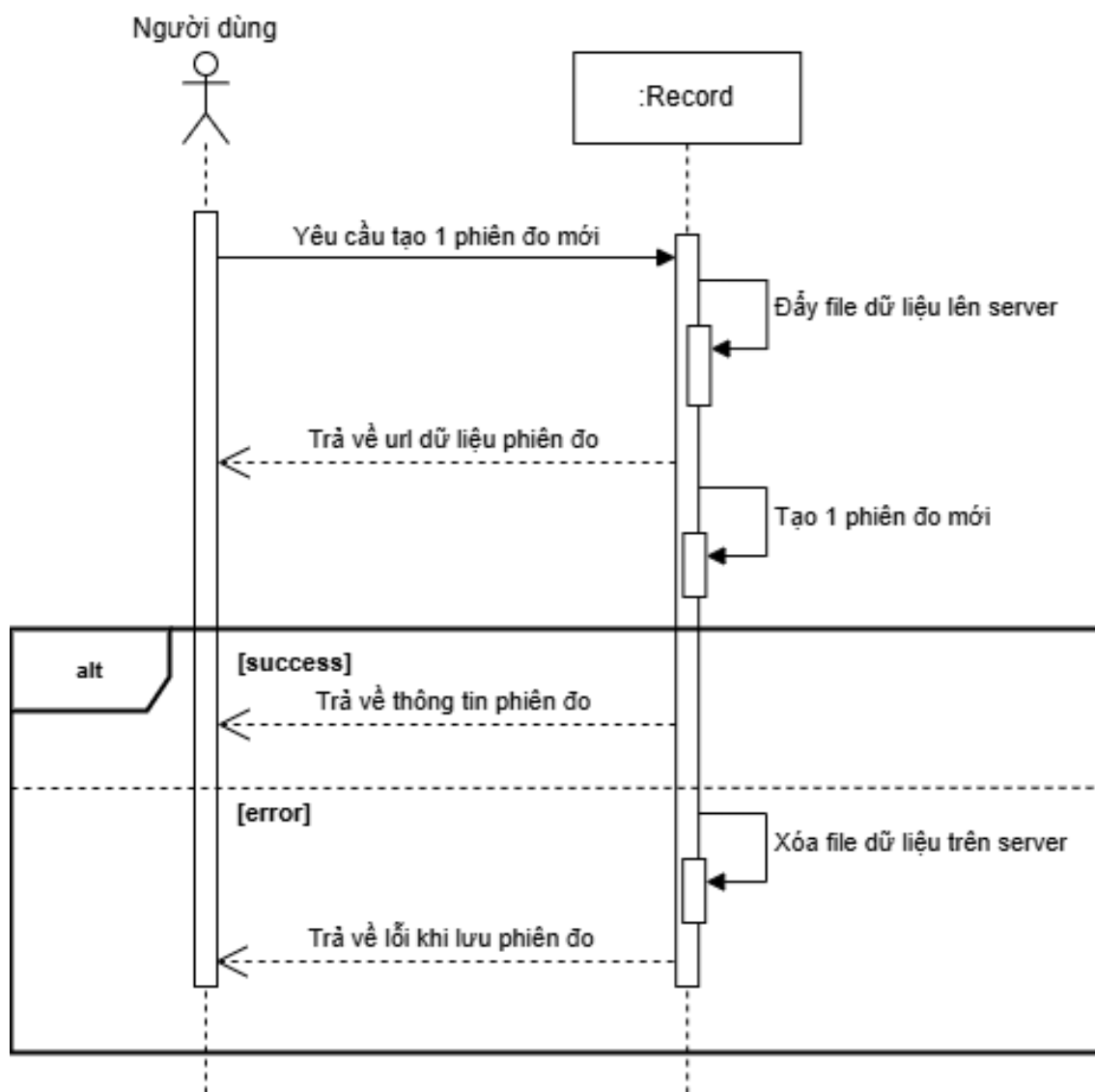
Hình 3.30 Sơ đồ tuần tự API tra cứu dữ liệu phiên đo theo id



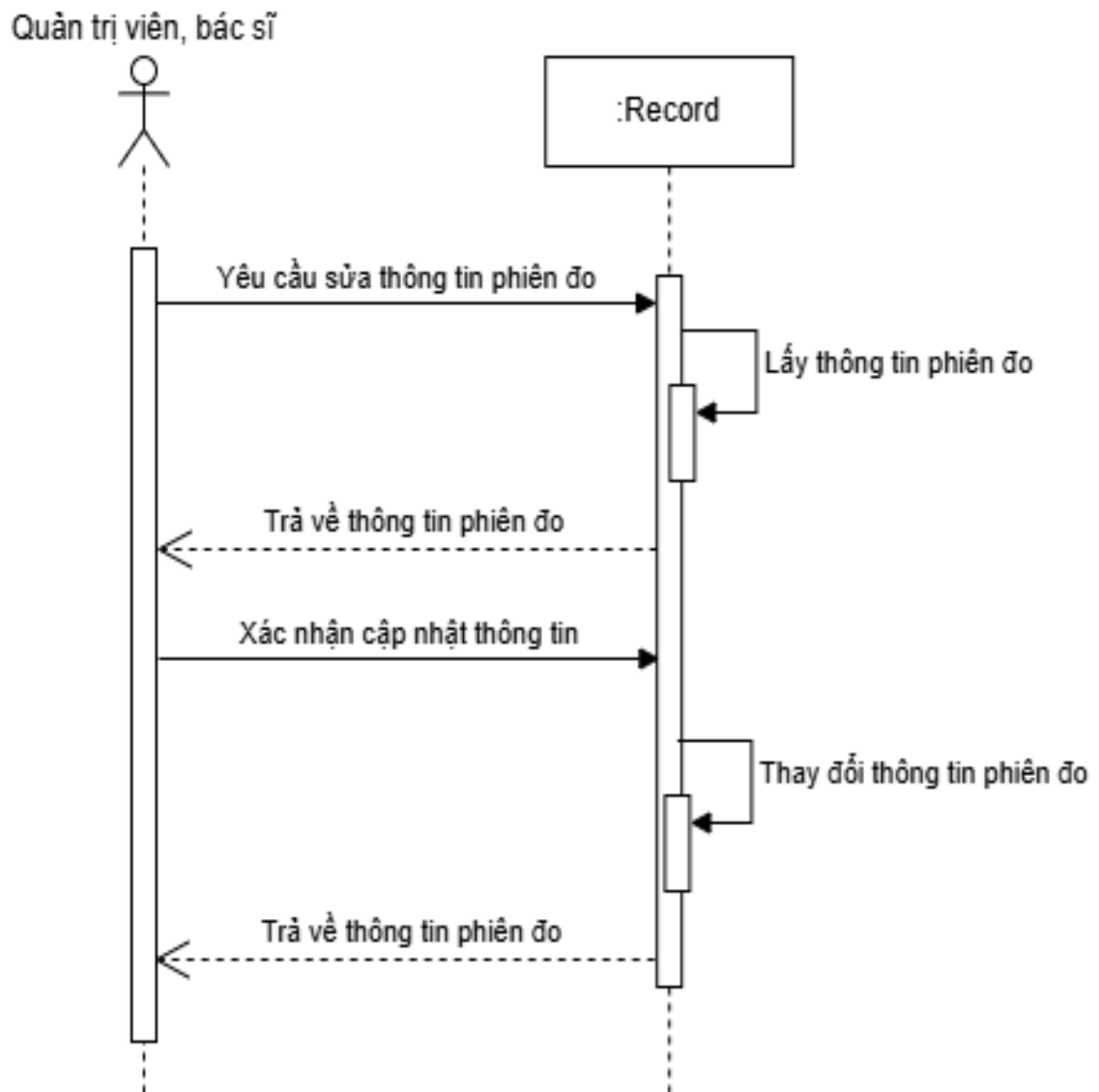
Hình 3.31 Sơ đồ tuần tự API tra cứu dữ liệu phiên đo của bệnh nhân



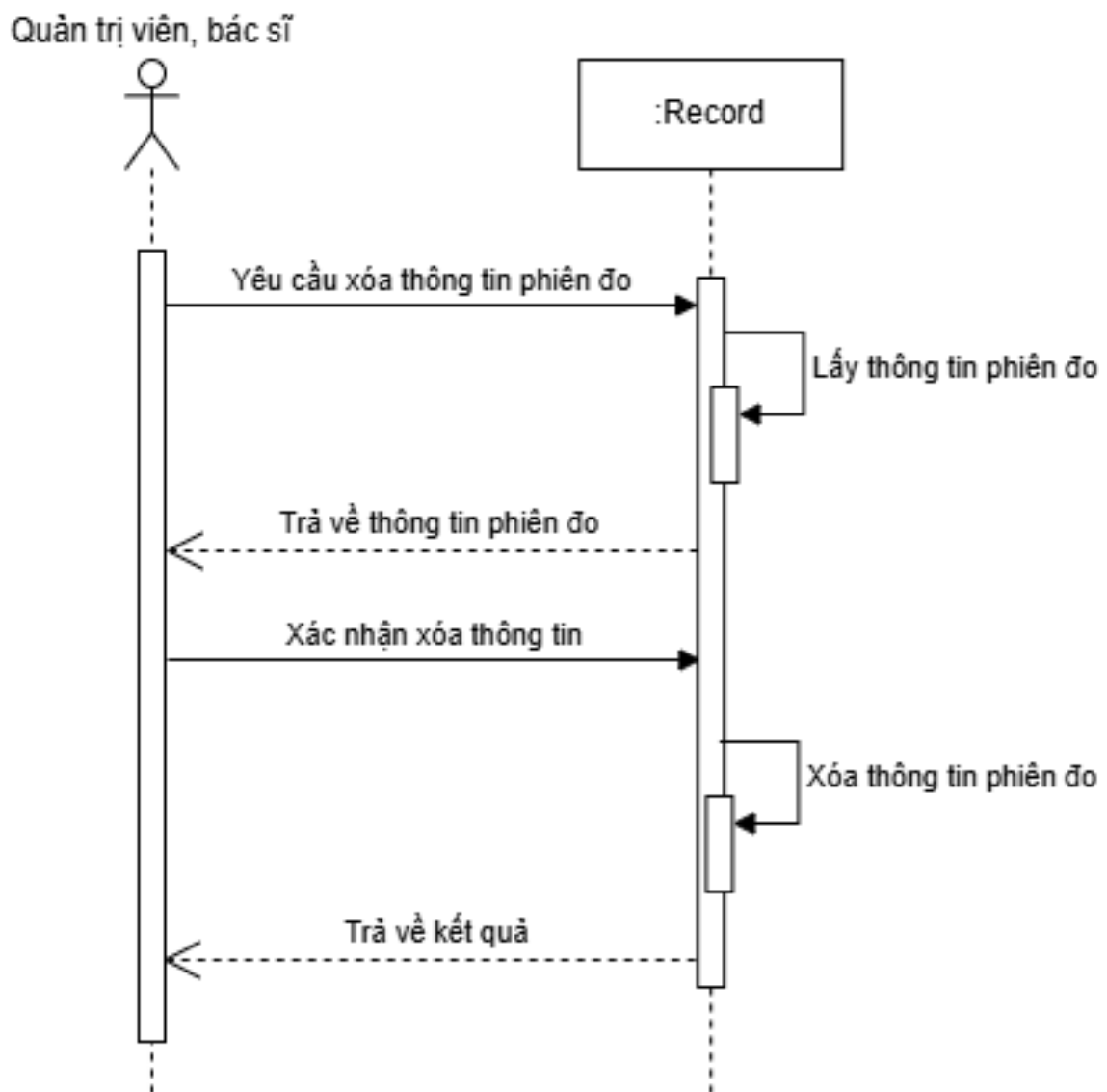
Hình 3.32 Sơ đồ tuần tự API tra cứu dữ liệu phiên đo do bác sĩ phụ trách



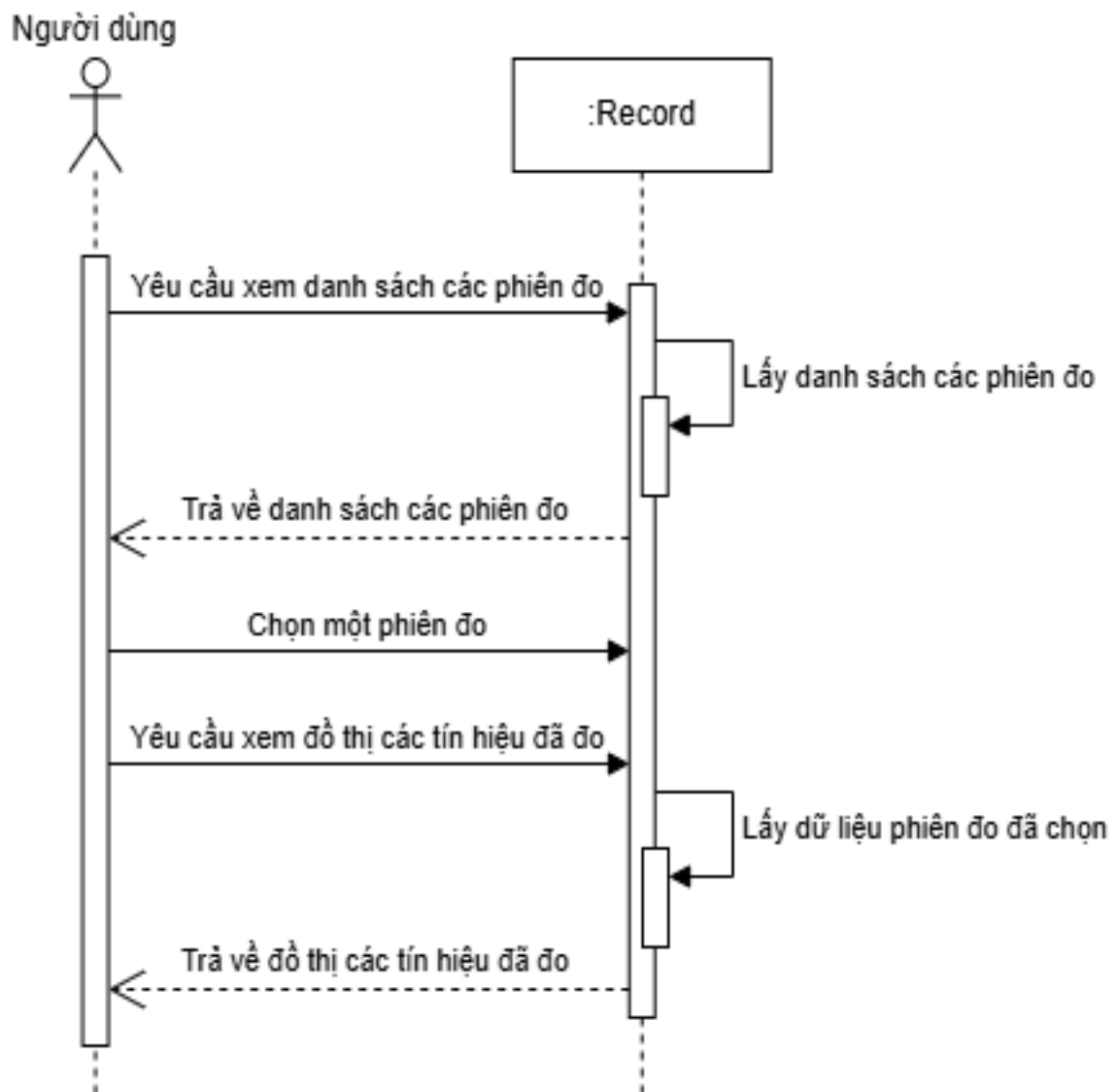
Hình 3.33 Sơ đồ tuần tự API tạo mới dữ liệu phiên đo



Hình 3.34 Sơ đồ tuần tự API cập nhật dữ liệu phiên đo

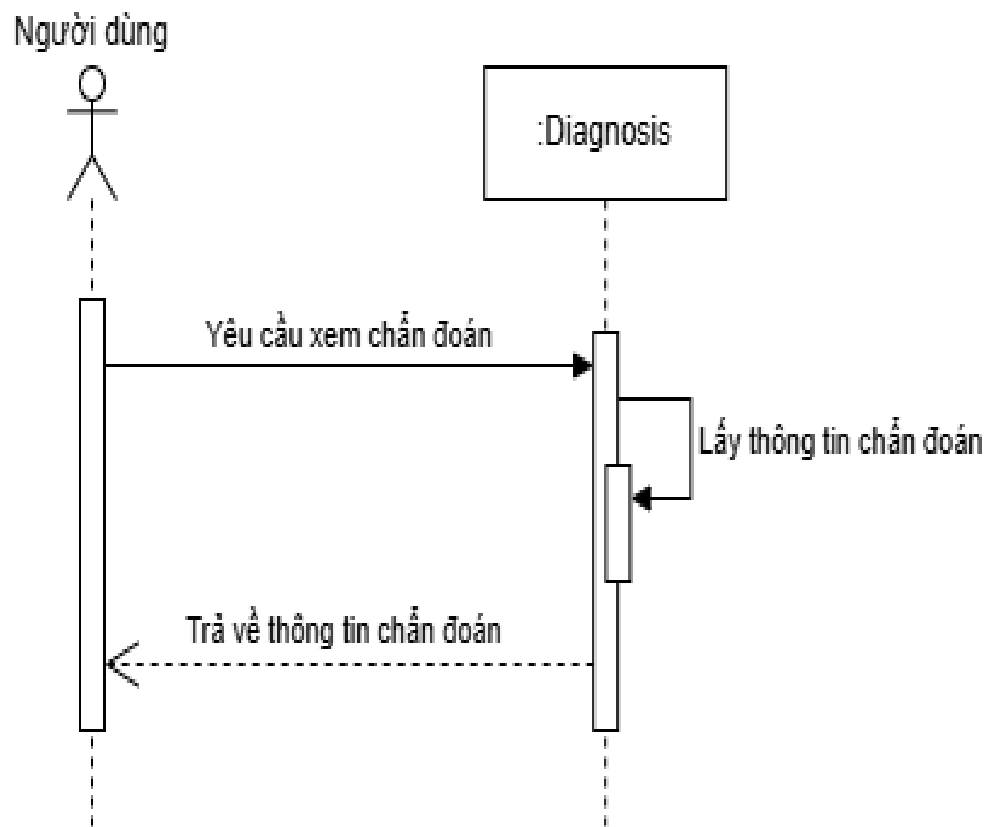


Hình 3.35 Sơ đồ tuần tự API xóa dữ liệu phiên đo

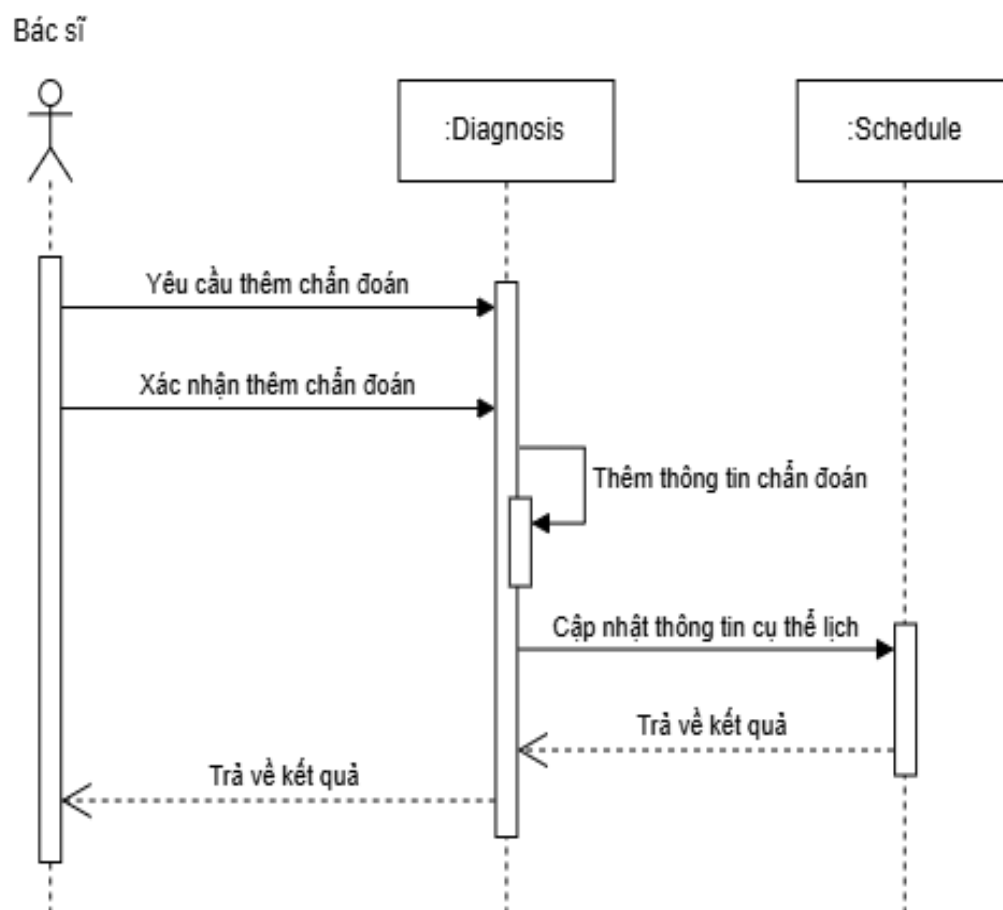


Hình 3.36 Sơ đồ tuần tự API tra cứu đồ thị các tín hiệu đã đo

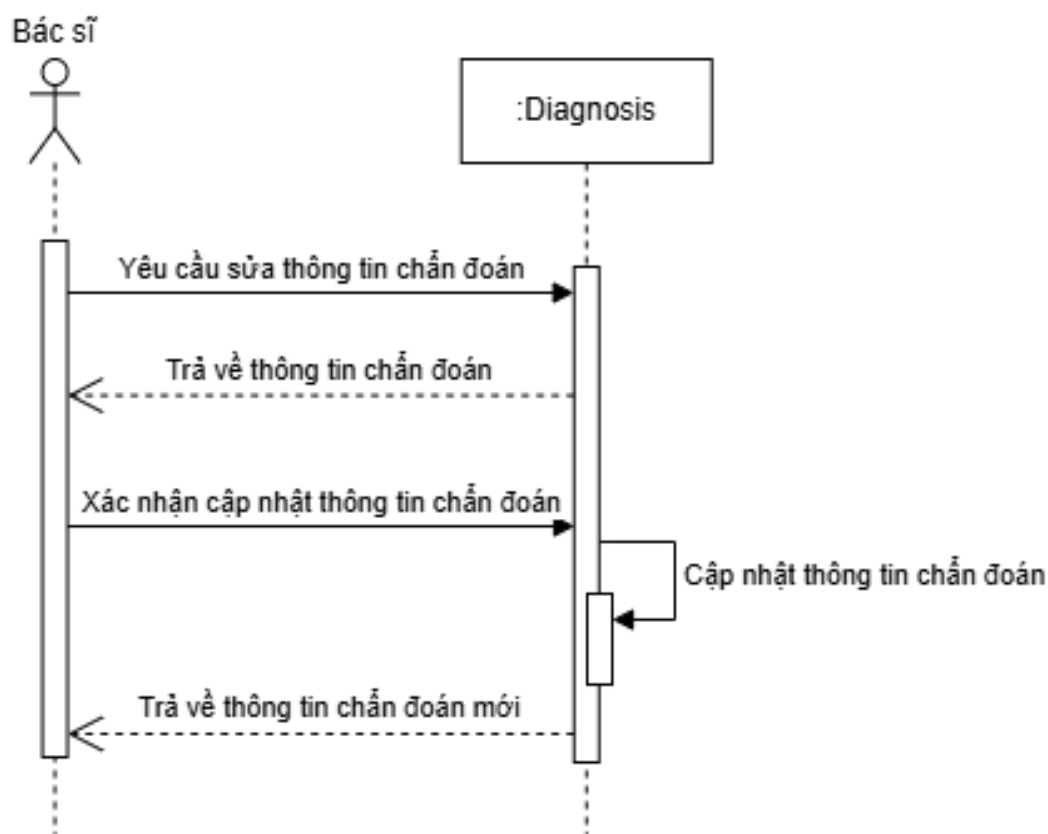
3.5.2.6 Các API phục vụ mục đích liên quan đến chẩn đoán



Hình 3.37 Sơ đồ tuần tự API lấy chẩn đoán theo lịch hẹn



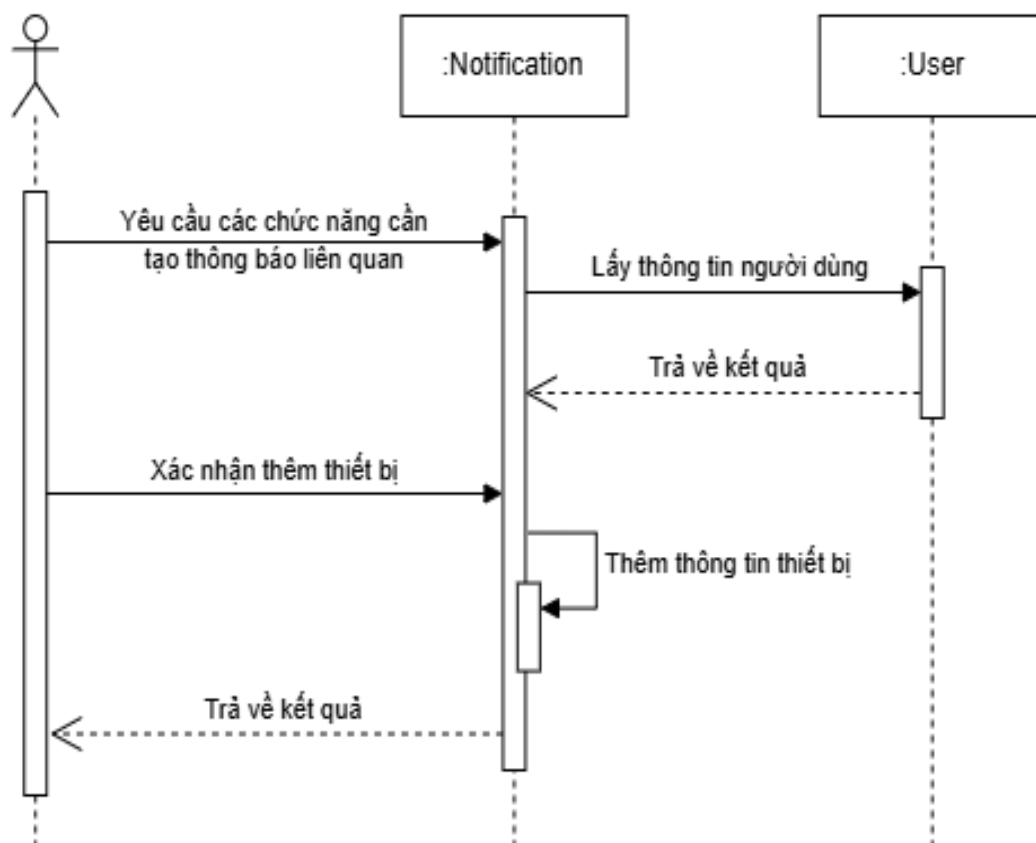
Hình 3.38 Sơ đồ tuần tự API tạo mới chẩn đoán



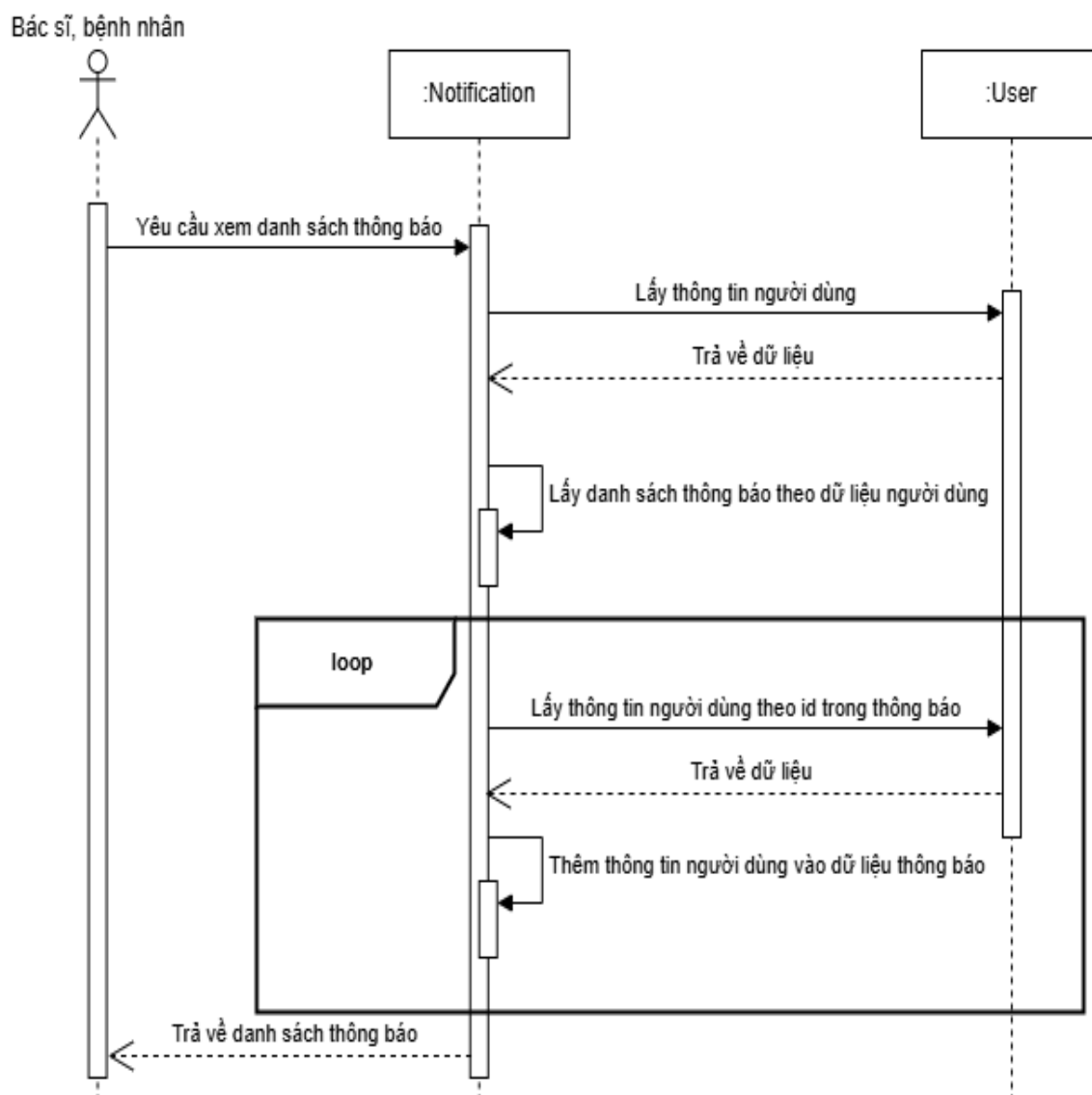
Hình 3.39 Sơ đồ tuần tự API cập nhật thông tin chẩn đoán

3.5.2.7 Các API phục vụ mục đích liên quan đến thông báo

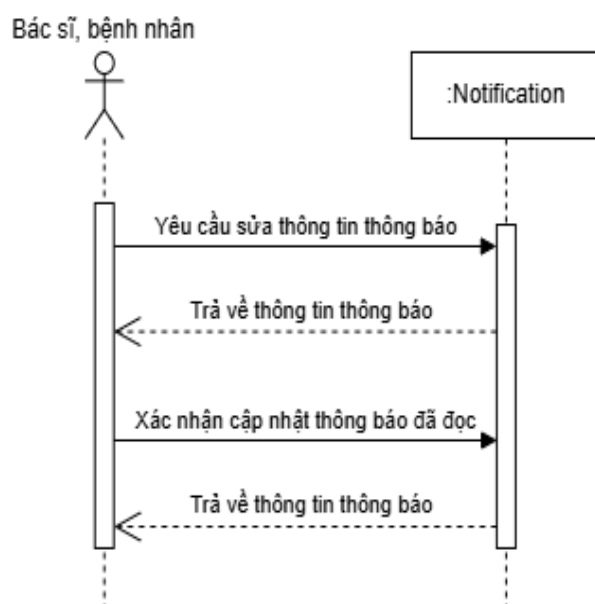
Bác sĩ, bệnh nhân



Hình 3.40 Sơ đồ tuần tự API tạo thông báo mới

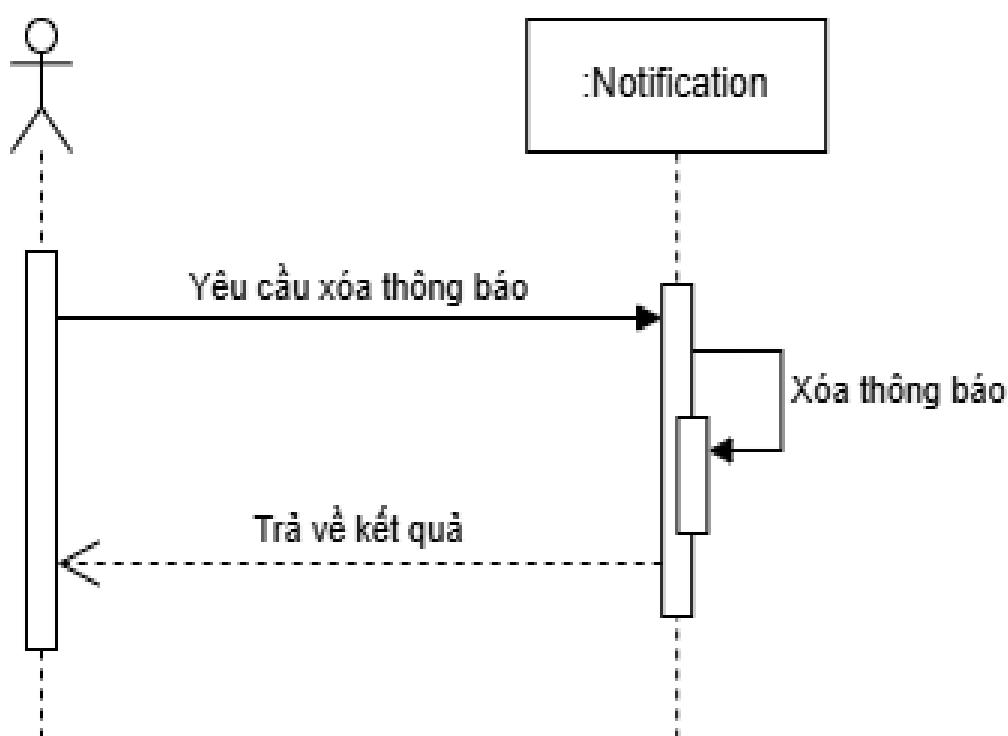


Hình 3.41 Sơ đồ tuần tự API lấy danh sách thông báo của người dùng



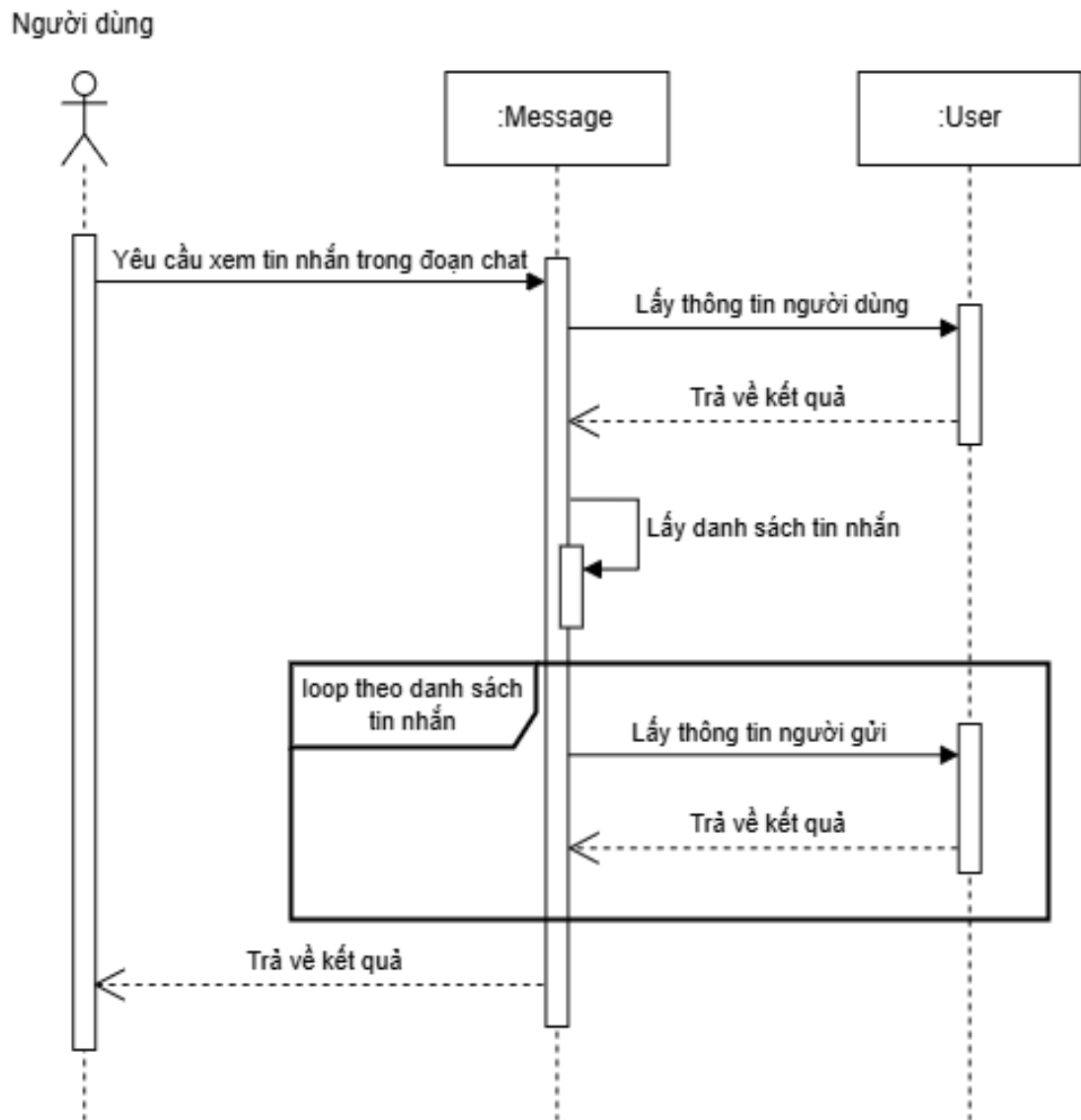
Hình 3.42 Sơ đồ tuần tự API cập nhật trạng thái đã xem thông báo

Bác sĩ, bệnh nhân

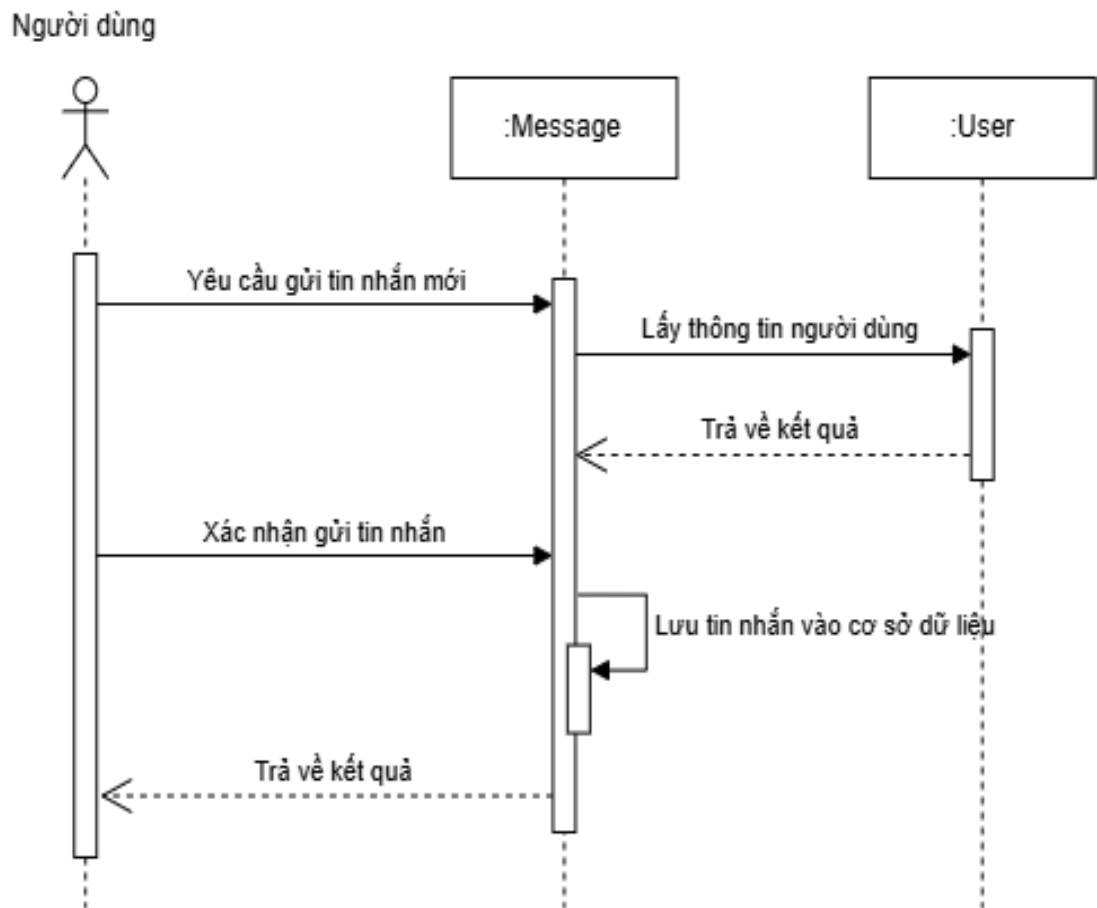


Hình 3.43 Sơ đồ tuần tự API xóa thông báo

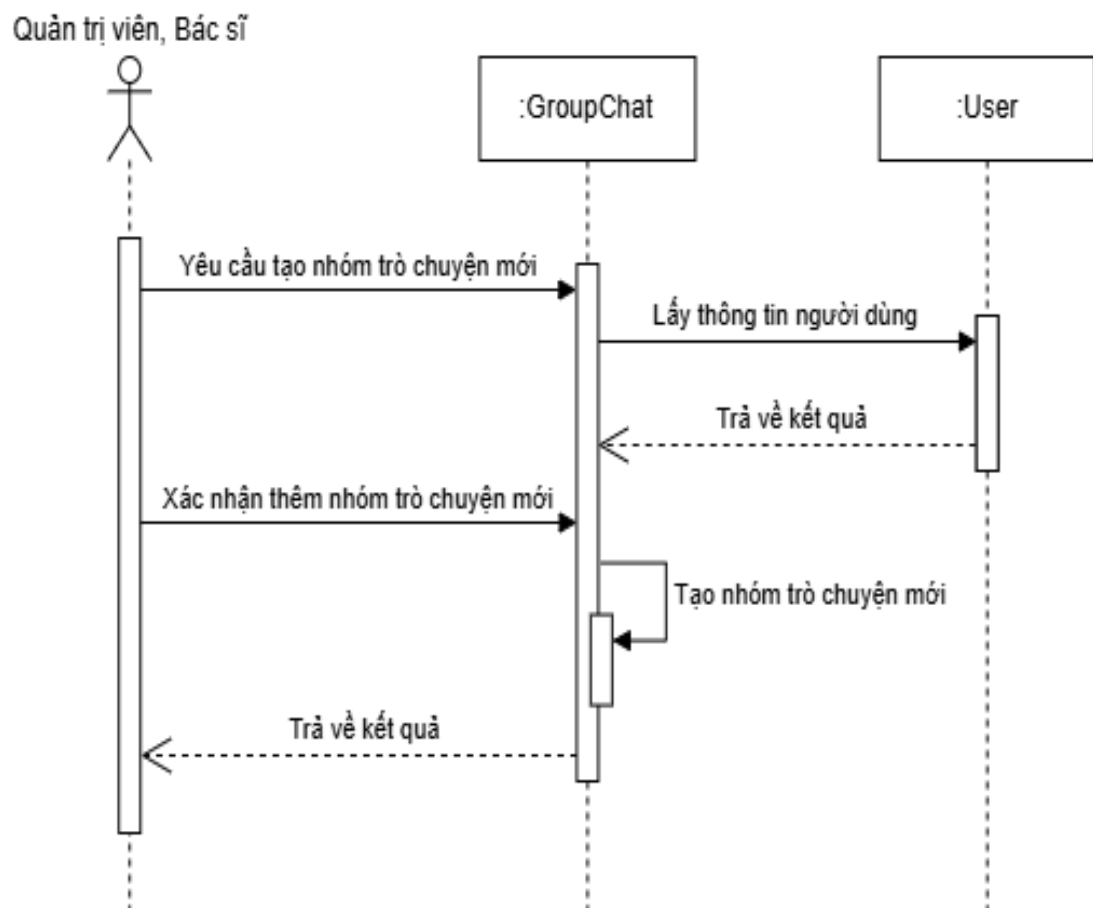
3.5.2.8 Các API phục vụ mục đích liên quan đến tin nhắn



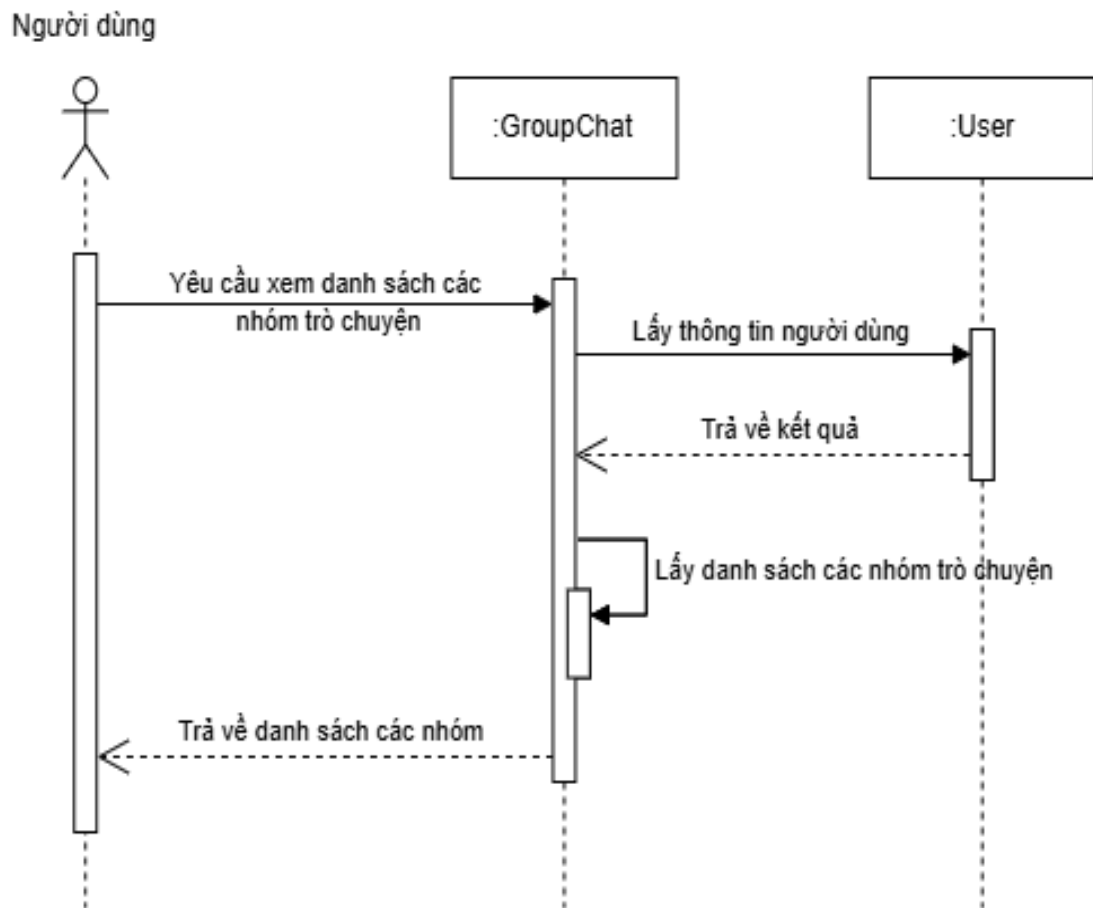
Hình 3.44 Sơ đồ tuần tự API lấy danh sách tin nhắn của người dùng



Hình 3.45 Sơ đồ tuần tự API gửi tin nhắn trong nhóm trò chuyện



Hình 3.46 Sơ đồ tuần tự API tạo nhóm trò chuyện mới



Hình 3.47 Sơ đồ tuần tự API lấy danh sách nhóm trò chuyện của người dùng

3.6 Kết luận chương

Chương 3 trình bày chi tiết về quá trình thiết kế hệ thống, bao gồm kiến trúc tổng thể và các thành phần cụ thể. Thiết kế hệ thống tập trung vào việc xây dựng kiến trúc vận hành hiệu quả và mượt mà, chú trọng vào tính bảo mật, hiệu suất và khả năng mở rộng tối ưu.

CHƯƠNG 4. TRIỂN KHAI VÀ KIỂM THỬ

4.1 Công nghệ sử dụng

4.1.1 Thiết kế giao diện website

4.1.1.1 *ReactJS*

ReactJS (thường được gọi tắt là React) là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook (nay là Meta). Chúng em ưu tiên sử dụng React để xây dựng giao diện người dùng (UI) bởi sự hiệu quả và linh hoạt của nó. React tập trung vào việc chia nhỏ giao diện thành các thành phần (components) độc lập, giúp quản lý và tái sử dụng code dễ dàng hơn.

Một số đặc điểm nổi bật của React chúng em đã sử dụng trong đồ án:

- Với các Component có thể lặp lại, chúng em có thể viết riêng ra để có thể tái sử dụng code.
- Hiệu suất cao, tối ưu việc cập nhật giao diện giúp phát triển giao diện nhanh chóng
- Dễ dàng debug: Facebook đã phát hành một Chrome extension hỗ trợ debug trong quá trình phát triển ứng dụng React.
- Tương thích với nhiều trình duyệt như Chrome, Edge, FireFox, Cốc Cốc, ...

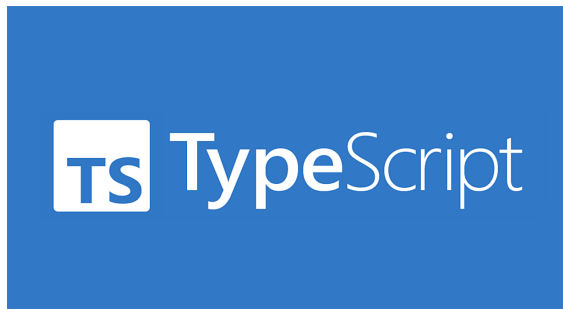


Hình 4.1 Logo ReactJs

4.1.2 Server

4.1.2.1 *TypeScript*

TypeScript là một ngôn ngữ lập trình mã nguồn mở được phát triển và duy trì bởi Microsoft. TypeScript là một công cụ mạnh mẽ giúp nâng cao chất lượng và hiệu suất của việc phát triển ứng dụng JavaScript.



Hình 4.2 Logo TypeScript

TypeScript mang lại nhiều lợi ích cho việc phát triển server của chúng em, bao gồm:

- Kiểu tĩnh giúp phát hiện lỗi ngay trong quá trình viết code, giúp tiết kiệm thời gian kiểm thử và sửa lỗi.
- Code dễ đọc và dễ bảo trì hơn với kiểu dữ liệu rõ ràng, giúp tăng hiệu suất và chất lượng code.
- Các công cụ và thư viện hỗ trợ mạnh mẽ từ TypeScript, giúp phát triển ứng dụng nhanh chóng và hiệu quả.
- Hỗ trợ OOP giúp chúng em xây dựng ứng dụng theo mô hình hướng đối tượng, dễ dàng mở rộng và bảo trì.

4.1.2.2 NodeJS

NodeJS (hay Node.js) là một môi trường chạy (runtime environment) cho JavaScript, cho phép bạn thực thi mã JavaScript bên ngoài trình duyệt web. Điều này giúp chúng em có thể sử dụng JavaScript để viết các ứng dụng phía máy chủ (back-end), các công cụ dòng lệnh, và nhiều thứ khác, thay vì chỉ giới hạn trong việc viết mã chạy trên trình duyệt (front-end).



Hình 4.3 Logo NodeJS

Một số đặc điểm chính của NodeJS:

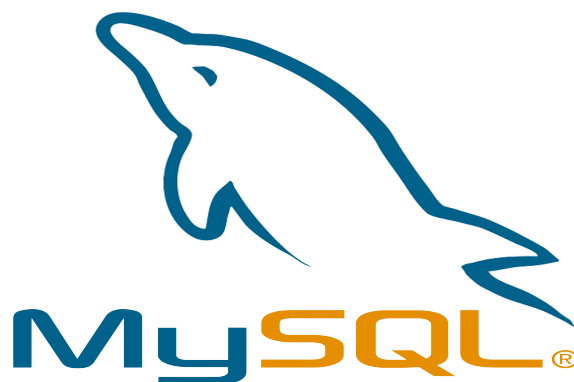
- Dựa trên V8 JavaScript Engine: NodeJS được xây dựng dựa trên V8, engine JavaScript cực kỳ mạnh mẽ của Google Chrome. Điều này giúp NodeJS thực thi mã JavaScript rất nhanh.
- Mã nguồn mở và đa nền tảng: NodeJS là dự án mã nguồn mở, được phát triển và duy trì bởi một cộng đồng lớn. Nó cũng chạy được trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux.
- Mô hình Non-blocking I/O: NodeJS sử dụng mô hình vào/ra không đồng bộ (non-blocking I/O), cho phép nó xử lý nhiều yêu cầu đồng thời một cách hiệu quả mà không bị tắc nghẽn.
- npm (Node Package Manager): NodeJS đi kèm với npm, một hệ thống quản lý gói mạnh mẽ, cho phép chúng em dễ dàng cài đặt, quản lý và chia sẻ các thư viện và module JavaScript hay TypeScript.

4.1.2.3 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến, được phát triển và duy trì bởi Oracle Corporation. Nó sử dụng Ngôn ngữ truy vấn cấu trúc (SQL) để quản lý và thao tác dữ liệu. MySQL lưu trữ dữ liệu trong các bảng, được tổ chức thành các cơ sở dữ liệu. Các bảng có thể liên kết với nhau thông qua các khóa, tạo thành mối quan hệ giữa các dữ liệu.

MySQL nổi tiếng với tốc độ, độ tin cậy và tính dễ sử dụng. Nó được sử dụng rộng rãi trong các ứng dụng web, ứng dụng doanh nghiệp và các hệ thống nhúng. MySQL hỗ trợ nhiều hệ điều hành, bao gồm Windows, Linux, macOS và Unix. Nó cũng tương thích với nhiều ngôn ngữ lập trình, như PHP, Java, Python và C++.

MySQL là một phần mềm mã nguồn mở, cho phép người dùng sử dụng, phân phối và sửa đổi mã nguồn một cách tự do theo các điều khoản của Giấy phép Công cộng GNU. Điều này đã góp phần vào sự phổ biến và phát triển của MySQL trong cộng đồng các nhà phát triển. Nó cung cấp nhiều tính năng mạnh mẽ, bao gồm bảo mật dữ liệu, sao lưu và phục hồi, và khả năng mở rộng.



Hình 4.4 Logo MySQL

4.1.2.4 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, được thiết kế để xử lý dữ liệu lớn, dữ liệu phi cấu trúc và bán cấu trúc một cách hiệu quả. Thay vì sử dụng các bảng và hàng như trong cơ sở dữ liệu quan hệ (SQL), MongoDB sử dụng các tài liệu (documents) theo định dạng JSON (JavaScript Object Notation) để lưu trữ dữ liệu. Điều này mang lại sự linh hoạt cao trong việc xử lý các loại dữ liệu khác nhau. MongoDB được phát triển và duy trì bởi MongoDB Inc.

Đặc điểm nổi bật của MongoDB là khả năng mở rộng linh hoạt theo chiều ngang (horizontal scaling) thông qua sharding, cho phép phân tán dữ liệu trên nhiều máy chủ, giúp xử lý lượng dữ liệu lớn và tăng hiệu suất. MongoDB thường được sử dụng trong các ứng dụng web hiện đại, ứng dụng di động, phân tích dữ liệu lớn và các ứng dụng yêu cầu tốc độ và khả năng mở rộng cao.



Hình 4.5 Logo MongoDB

4.1.2.5 Postman

Postman là một nền tảng mạnh mẽ giúp đơn giản hóa quá trình phát triển và kiểm thử API. Thay vì phải viết code phức tạp để tương tác với API, Postman cung cấp một giao diện trực quan, cho phép người dùng dễ dàng gửi các yêu cầu HTTP (GET, POST, PUT, DELETE,...) với đầy đủ các tùy chỉnh về headers, body, parameters. Không chỉ dừng lại ở việc gửi yêu cầu, Postman còn hỗ trợ tổ chức API thành các bộ sưu tập (Collections), tự động hóa kiểm thử bằng JavaScript, giả lập server (Mock Server) và theo dõi hiệu suất API. Với những tính năng này, Postman trở thành một "trợ thủ đắc lực" cho cả lập trình viên và kiểm thử viên.

Trong quá trình phát triển phần mềm hiện đại, API đóng vai trò trung tâm trong việc kết nối các ứng dụng. Việc kiểm thử API một cách hiệu quả là vô cùng quan trọng để đảm bảo chất lượng và tính ổn định của hệ thống. Postman ra đời để giải quyết bài toán này. Nó giúp người dùng dễ dàng tạo và kiểm thử các API, tiết kiệm thời gian và công sức so với việc viết code thủ công. Khả năng tự động hóa kiểm thử của Postman giúp phát hiện sớm các lỗi tiềm ẩn, trong khi tính năng mock server cho phép tiếp tục phát triển ngay cả khi API chưa hoàn thiện.



Hình 4.6 Logo Postman

4.1.2.6 Docker

Trước Docker, việc triển khai ứng dụng thường gặp nhiều khó khăn do sự khác biệt về môi trường giữa các máy tính. Ví dụ: một ứng dụng chạy tốt trên máy tính của nhà phát triển có thể gặp lỗi khi triển khai lên máy chủ do thiếu thư viện, phiên bản phần mềm khác nhau, hoặc cấu hình không tương thích. Docker giải quyết vấn đề này bằng cách đóng gói tất cả các thành phần cần thiết cho ứng dụng vào một container, đảm bảo ứng dụng luôn chạy như nhau ở mọi nơi. Các từ khóa chính cần quan tâm như: Image, container, DockerFile, Docker Compose, ...

Những lợi ích của Docker có thể kể đến như:

- Tính nhất quán: đảm bảo ứng dụng chạy trên mọi môi trường
- Tính di động: dễ dàng di chuyển ứng dụng giữa các máy tính, máy chủ, và đám mây.
- Tiết kiệm tài nguyên: Container nhẹ hơn nhiều so với máy ảo (Virtual Machine) vì chúng chia sẻ kernel của hệ điều hành host.
- Tăng tốc độ phát triển và triển khai: Docker giúp tự động hóa quá trình xây dựng, kiểm thử, và triển khai ứng dụng.
- Dễ dàng quản lý phiên bản: Docker giúp quản lý các phiên bản khác nhau của ứng dụng một cách dễ dàng.
- Cách ly ứng dụng: Container cung cấp môi trường cách ly, giúp ngăn chặn xung đột giữa các ứng dụng.



Hình 4.7 Logo Docker

4.2 Triển khai ứng dụng

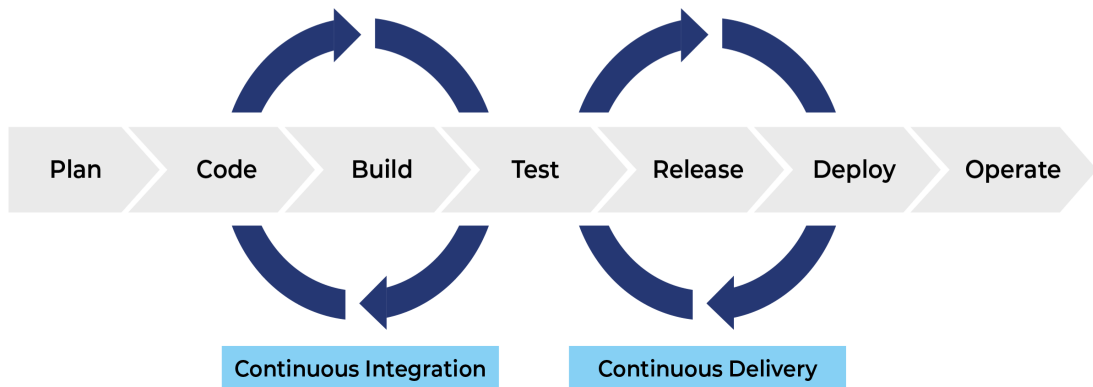
Trong quá trình triển khai ứng dụng, chúng em áp dụng quy trình phát triển ứng dụng CI/CD để liên tục cập nhật và deploy. Để triển khai hệ thống theo đúng quy trình CI/CD thì chúng em đã sử dụng các dịch vụ, VPS server để triển khai server API và website, Docker để tự động deploy ứng dụng, MySQL Server và MongoDB Compass để quản lý cơ sở dữ liệu, Github để quản lý code của dự án.

4.2.1 Quy trình CI/CD

CI/CD là một tập hợp các phương pháp và thực hành giúp tự động hóa quy trình phát triển, kiểm thử và triển khai phần mềm. CI (Continuous Integration - Tích hợp liên tục) tập trung vào việc thường xuyên tích hợp mã nguồn từ nhiều nhà phát triển vào một kho lưu trữ chung, sau đó tự động xây dựng và kiểm tra phần mềm. Việc này giúp phát hiện sớm các lỗi tích hợp và giảm thiểu xung đột giữa các thay đổi.

CD (Continuous Delivery/Continuous Deployment - Chuyển giao/Triển khai liên tục) mở rộng CI bằng cách tự động hóa quá trình chuyển giao phần mềm đã được kiểm tra đến môi trường thử nghiệm hoặc sản xuất. Continuous Delivery cho phép triển khai thủ công sau khi kiểm tra, trong khi Continuous Deployment tự động triển khai lên môi trường sản xuất. CI/CD giúp tăng tốc độ phát hành phần mềm, cải thiện chất lượng và giảm thiểu rủi ro.

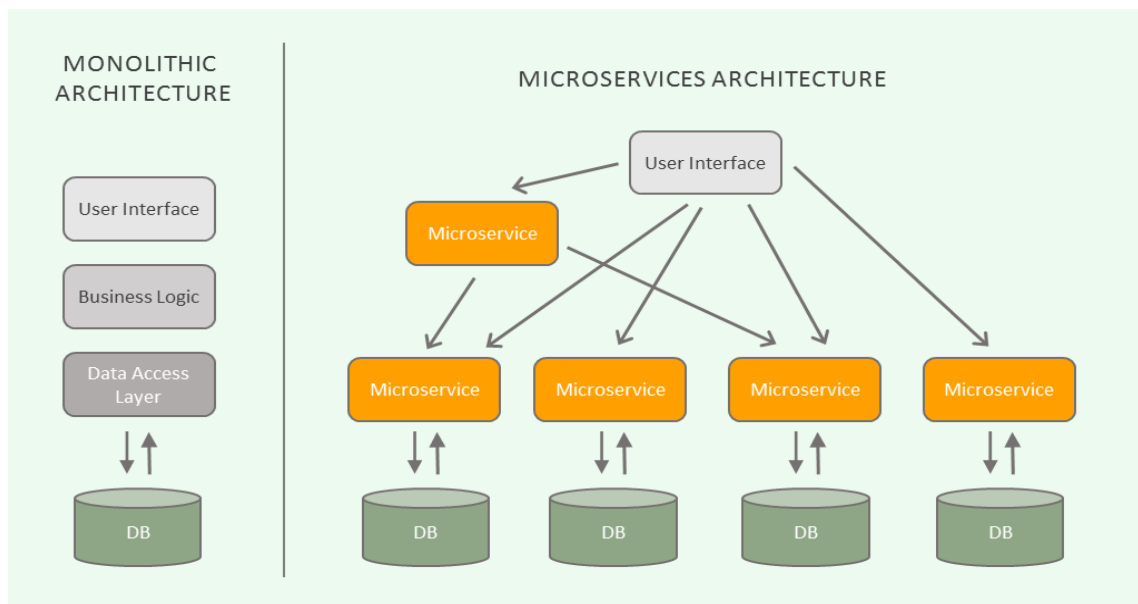
CI/CD



Hình 4.8 Quy trình CI/CD

4.2.2 Kiến trúc Microservices

Kiến trúc Microservices là một phương pháp phát triển phần mềm trong đó một ứng dụng lớn được chia thành nhiều dịch vụ nhỏ độc lập với nhau. Mỗi dịch vụ thực hiện một chức năng cụ thể và có thể được phát triển, triển khai, mở rộng và bảo trì một cách độc lập với các microservice khác. Chúng thường giao tiếp với nhau thông qua mạng, sử dụng các giao thức như HTTP/REST hoặc message queues.



Hình 4.9 Kiến trúc microservices

Một số lợi ích của microservice đã hỗ trợ chúng em:

- Tính độc lập: Mỗi microservice là một đơn vị triển khai độc lập. Việc thay đổi một

microservice không ảnh hưởng đến các microservice khác, giúp tăng tính linh hoạt và tốc độ phát triển.

- Tính chuyên biệt: Mỗi microservice tập trung vào một chức năng nghiệp vụ cụ thể, giúp đơn giản hóa việc phát triển và bảo trì.
- Khả năng mở rộng: Có thể mở rộng từng microservice một cách độc lập dựa trên nhu cầu, giúp tối ưu hóa việc sử dụng tài nguyên.
- Khả năng chịu lỗi: Nếu một microservice gặp sự cố, các microservice khác vẫn có thể hoạt động bình thường, giúp tăng tính ổn định của hệ thống.

4.3 Kiểm thử

4.3.1 Kiểm thử hoạt động của các API

Công cụ: Postman - Để xây dựng và thực hiện các yêu cầu API.

4.3.1.1 API liên quan đến việc xác thực người dùng

a) URL: POST auth/register

Bảng 4.1 Bảng kiểm thử API đăng ký tài khoản

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Người dùng mới	Dữ liệu đăng ký { "username": "Co Huy Dung", "password": "123456a@", "email": "co-dung2909@gmail.com", "birth": "29/09/2003", "gender": 1, "phone_number": "077433306x", "role": 0 }	Status code: 200 OK Response message: { "status": "success", "message": "register successfully" }	OK
TC-2	Tài khoản email đã tồn tại	Dữ liệu đăng ký { "username": "Nguyễn Đức B", "password": "xYnF1452", "email": "co-dung2909@gmail.com", "birth": "15/07/1995", "gender": 1, "phone_number": "091601736x", "role": 0 }	Status code: 400 Bad Request Response message: { "status": "error", "message": "email is existed" }	OK

b) URL: POST auth/login

Bảng 4.2 Bảng kiểm thử API người dùng đăng nhập

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Thông tin tài khoản và mật khẩu hợp lệ	Thông tin đăng nhập { "email": email đã đăng ký, "password": mật khẩu tương ứng }	Status code: 200 OK Response message: { "status": "success", data: Thông tin user sau khi đăng ký thành công }	OK
TC-2	Thông tin tài khoản và mật khẩu không hợp lệ	Thông tin đăng nhập { "email": email người dùng, "password": mật khẩu người dùng }	Status code: 401 Unauthorized Response message: { "status": "401 Unauthorized", "message": "Invalid email or password" }	OK

c) URL: POST auth/logout

Bảng 4.3 Bảng kiểm thử API đăng xuất người dùng

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Người dùng đã đăng nhập vào hệ thống	JWT Token tồn tại	Status code: 200 OK Response message: { "status": "success", "message": "Logged out successfully" }	OK
TC-2	Người dùng chưa đăng nhập vào hệ thống	JWT Token không tồn tại	Status code: 401 Unauthorized Response message: { "status": "error", "message": "No token found" }	OK

4.3.1.2 API liên quan đến thông tin người dùng

a) URL: GET users

Bảng 4.4 Bảng kiểm thử API lấy danh sách người dùng

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Admin của hệ thống	Access token tương ứng	Status code: 200 Response message: { "data": Danh sách người dùng }	OK
TC-2	Không phải admin	Access token tương ứng	Status code: 403 Response message: { "status": "error", "message": "403 Forbidden" }	OK
TC-3	Không có token	NULL	Status code: 401 Response message: { "status": "error", "message": "401 Unauthorized" }	OK

b) URL: GET users/doctors

Bảng 4.5 Bảng kiểm thử API lấy danh sách bác sĩ

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Admin, bác sĩ của hệ thống	Access token tương ứng	Status code: 200 Response message: { "data": Danh sách bác sĩ }	OK

TC-2	Không phải admin, bác sĩ	Access token tương ứng	Status code: 403 Response message: { "status": "error", "message": "403 Forbidden" }	OK
TC-3	Không có token	NULL	Status code: 401 Response message: { "status": "error", "message": "401 Unauthorized" }	OK

c) URL: GET users/:id

Bảng 4.6 Bảng kiểm thử API lấy dữ liệu người dùng bằng id

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Người dùng tồn tại trong hệ thống và có id tương ứng	id người dùng	Status code: 200 OK Response message: { "data": Thông tin người dùng }	OK
TC-2	Người dùng không tồn tại trong hệ thống với id tương ứng	id người dùng	Status code: 404 Not Found Response message: { "status": "error", "message": "No user found, please try again" }	OK

d) URL: GET users/data/patient-data

Bảng 4.7 Bảng kiểm thử API lấy danh sách bệnh nhân theo id của bác sĩ

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Admin, bác sĩ tương ứng với id	id của bác sĩ	Status code: 200 Response message: { "data": Danh sách bệnh nhân }	OK
TC-2	Không phải admin, bác sĩ	id của bác sĩ	Status code: 403 Response message: { "status": "error", "message": "403 Forbidden" }	OK
TC-3	Không có token		Status code: 401 Response message: { "status": "error", "message": "401 Unauthorized" }	OK

e) URL: GET users/data/doctor-id

Bảng 4.8 Bảng kiểm thử API lấy danh sách bác sĩ theo id của bệnh nhân

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Admin, bác sĩ hoặc bệnh nhân có id tương ứng	id của bệnh nhân	Status code: 200 Response message: { "data": Danh sách bác sĩ của bệnh nhân }	OK
TC-2	Bệnh nhân chưa có bác sĩ	id của bệnh nhân	Status code: 200 Response message: { "status": "error", "message": "No user found, please try again" }	OK

TC-3	Không phải admin, bác sĩ hoặc bệnh nhân	id của bệnh nhân	Status code: 403 Response message: { "status": "error", "message": "403 Forbidden" }	OK
TC-4	Không có token		Status code: 401 Response message: { "status": "error", "message": "401 Unauthorized" }	OK

f) URL: PUT users

Bảng 4.9 Bảng kiểm thử API cập nhật thông tin người dùng

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Người dùng đã tồn tại trong hệ thống với id cho trước	Dữ liệu cập nhật của người dùng	Status code: 200 OK Response message: { "data": Dữ liệu người dùng đã được cập nhật }	OK
TC-2	Người dùng không tồn tại trong hệ thống với id cho trước		Status code: 404 Not found Response message: { "message": "No user found to update, please try again" }	OK

TC-3	Người dùng không có quyền sửa thông tin		Status code: 403 Forbidden Response message: { "message": "403 Forbidden" }	OK
------	-----------------------------------------	--	-----------------------------------------------------------------------------------------------	----

g) URL: DELETE users/:userId

Bảng 4.10 Bảng kiểm thử API xóa thông tin người dùng

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Người dùng đã tồn tại trong hệ thống với id cho trước	id của người dùng	Status code: 200 OK Response message: { "message": "User has been deleted successfully" }	OK
TC-2	Người dùng không tồn tại trong hệ thống với id cho trước	id của người dùng	Status code: 404 Not found Response message: { "message": "No user found to delete, please try again" }	OK

h) URL: GET statistic

Bảng 4.11 Bảng kiểm thử API lấy dữ liệu thống kê

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Admin của hệ thống	Access token tương ứng	Status code: 200 OK Response message: { data: dữ liệu thống kê số lượng người dùng, thiết bị, dữ liệu phiên đo mỗi tháng }	OK
TC-2	Không phải admin của hệ thống	Access token tương ứng	Status code: 403 Forbidden Response message: { "message": "403 Forbidden" }	OK

4.3.1.3 API liên quan đến thiết bị

a) URL: GET device

Bảng 4.12 Bảng kiểm thử API lấy danh sách thiết bị

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Admin của hệ thống	Access token tương ứng	Status code: 200 OK Response message: { "data": Danh sách thiết bị }	OK
TC-2	Không phải admin của hệ thống	Access token tương ứng	Status code: 403 Forbidden Response message: { "message": "403 Forbidden" }	OK

b) URL: GET device/:id

Bảng 4.13 Bảng kiểm thử API lấy dữ liệu thiết bị theo id

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Thiết bị tồn tại với id cho trước	id thiết bị	Status code: 200 OK Response message: { data: Thông tin của thiết bị }	OK
TC-2	Thiết bị không tồn tại với id cho trước	id thiết bị	Status code: 404 Not Found Response message: { "message": "No device found, please try again" }	OK

c) URL: POST device/add

Bảng 4.14 Bảng kiểm thử API thêm thiết bị

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Bệnh nhân và bác sĩ tồn tại với ID tương ứng	Thông tin thiết bị { "user_id": id bệnh nhân, "device_name": Tên thiết bị, "infomation": Thông tin thiết bị, "device_type_id": id loại thiết bị, "status_id": id trạng thái thiết bị, "start_time": Giờ bắt đầu sử dụng "end_time": Giờ kết thúc sử dụng }	Status code: 200 OK Response message: { "message": "Device created successfully", data: Thông tin của thiết bị }	OK

TC-2	Người dùng không tồn tại với id đã cho	Thông tin thiết bị { "user_id": id bệnh nhân, "device_name": Tên thiết bị, "infomation": Thông tin thiết bị, "device_type_id": id loại thiết bị, "status_id": id trạng thái thiết bị, "start_time": Giờ bắt đầu sử dụng "end_time": Giờ kết thúc sử dụng }	Status code: 404 Not found Response message: { "message": "No user found, please try again" }	OK
------	----------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------	----

d) URL: POST device_detail

Bảng 4.15 Bảng kiểm thử API thêm thông số kỹ thuật thiết bị

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Thiết bị tồn tại với id cho trước	thông số kỹ thuật thiết bị { "device_id": id thiết bị, "detail_name": Tên chi tiết, "detail_type": loại chi tiết, "value": giá trị, "infomation": Thông tin ghi chú chi tiết, }	Status code: 200 OK Response message: { "message": "Device detail created successfully", data: thông số kỹ thuật của thiết bị }	OK

TC-2	Thiết bị không tồn tại với id cho trước	thông số kỹ thuật thiết bị { "device_id": id thiết bị, "detail_name": Tên chi tiết, "detail_type": loại chi tiết, "value": giá trị, "infomation": Thông tin ghi chú chi tiết, }	Status code: 404 Not found Response message: { "message": "Device not found" }	OK
TC-3	Người dùng không tồn tại với id tương ứng	thông số kỹ thuật thiết bị { "device_id": id thiết bị, "detail_name": Tên chi tiết, "detail_type": loại chi tiết, "value": giá trị, "infomation": Thông tin ghi chú chi tiết, }	Status code: 404 Not found Response message: { "message": "No user found, please try again" }	OK

e) URL: DELETE device/:device_id

Bảng 4.16 Bảng kiểm thử API xóa thiết bị theo id

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Thiết bị tồn tại với id cho trước	id thiết bị	Status code: 200 OK Response message: { "message": "Delete device successful" }	OK

TC-2	Không tồn tại thiết bị với id cho trước	id thiết bị	Status code: 404 Not found Response message: { "message": "Device not found" }	OK
------	-----------------------------------------	-------------	--------------------------------------------------------------------------------------------	----

f) URL: PUT device/:device_id

Bảng 4.17 Bảng kiểm thử API cập nhật thông tin thiết bị

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Thiết bị tồn tại với id cho trước	Thông tin thiết bị { "user_id": id bệnh nhân, "device_name": Tên thiết bị, "infomation": Thông tin thiết bị, "device_type_id": id loại thiết bị, "status_id": id trạng thái thiết bị, "start_time": Giờ bắt đầu sử dụng "end_time": Giờ kết thúc sử dụng }	Status code: 200 OK Response content: { data: Thông tin sau khi cập nhật của thiết bị }	OK

TC-2	Không tồn tại thiết bị với id cho trước	Thông tin thiết bị { "user_id": id bệnh nhân, "device_name": Tên thiết bị, "infomation": Thông tin thiết bị, "device_type_id": id loại thiết bị, "status_id": id trạng thái thiết bị, "start_time": Giờ bắt đầu sử dụng "end_time": Giờ kết thúc sử dụng }	Status code: 404 Not found Response content: { "message": "No device found to update, please try again" }	OK
TC-3	Người dùng không tồn tại với id cho trước	Thông tin thiết bị { "user_id": id bệnh nhân, "device_name": Tên thiết bị, "infomation": Thông tin thiết bị, "device_type_id": id loại thiết bị, "status_id": id trạng thái thiết bị, "start_time": Giờ bắt đầu sử dụng "end_time": Giờ kết thúc sử dụng }	Status code: 404 Not found Response content: { "message": "No user found, please try again" }	OK

g) URL: PUT device_detail/

Bảng 4.18 Bảng kiểm thử API cập nhật thông số kỹ thuật thiết bị

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Thông tin chi tiết mới phù hợp	thông số kỹ thuật thiết bị { "device_id": id thiết bị, "detail_name": Tên chi tiết, "detail_type": loại chi tiết, "value": giá trị, "infomation": Thông tin ghi chú chi tiết, }	Status code: 200 OK Response message: { "message": "Device detail updated successfully", data: thông số kỹ thuật của thiết bị sau khi cập nhật }	OK
TC-2	Thông tin chi tiết mới không phù hợp	thông số kỹ thuật thiết bị { "device_id": id thiết bị, "detail_name": Tên chi tiết, "detail_type": loại chi tiết, "value": giá trị, "infomation": Thông tin ghi chú chi tiết, }	Status code: 400 Bad Request Response message: { "message": "Error when update device" }	OK

h) URL: DELETE device_detail/:detail_id

Bảng 4.19 Bảng kiểm thử API xóa thông số kỹ thuật thiết bị

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Thông tin chi tiết mới phù hợp	id thông số kỹ thuật thiết bị	Status code: 200 OK Response message: { "message": "Device detail deleted successfully", data: thông số kỹ thuật của thiết bị sau khi cập nhật }	OK
------	--------------------------------	-------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------	----

4.3.1.4 API liên quan đến dữ liệu phiên đo

a) URL: GET records

Bảng 4.20 Bảng kiểm thử API lấy tất cả dữ liệu phiên đo

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Admin của hệ thống	Access Token tương ứng	Status code: 200 OK Response message: { data: Danh sách dữ liệu phiên đo }	OK
TC-2	Không phải admin của hệ thống	Access Token tương ứng	Status code: 403 Forbidden Response message: { "message": "Forbidden" }	OK

b) URL: GET records/:id

Bảng 4.21 Bảng kiểm thử API lấy dữ liệu phiên đo theo id

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Là admin, bác sĩ của hệ thống, dữ liệu phiên đo tồn tại	id của dữ liệu phiên đo	Status code: 200 OK Response message: { "data": Danh sách tất cả dữ liệu phiên đo trong hệ thống }	OK
TC-2	Không phải admin, bác sĩ	id của dữ liệu phiên đo	Status code: 403 Forbidden Response message: { "message": "Forbidden" }	OK
TC-3	Là admin, bác sĩ của hệ thống, dữ liệu phiên đo không tồn tại	id của dữ liệu phiên đo	Status code: 404 Not Found Response message: { "message": "No record found, please try again" }	OK

c) URL: GET records/doctor/:doctorId

Bảng 4.22 Bảng kiểm thử API lấy các dữ liệu phiên đo theo id của bác sĩ

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Là admin hoặc bác sĩ có id cho trước	id bác sĩ	Status code: 200 OK Response message: { data: Danh sách các dữ liệu phiên đo của bác sĩ }	OK
TC-2	Không phải là admin hoặc bác sĩ tương ứng	id bác sĩ	Status code: 403 Forbidden Response message: { "message": "Forbidden" }	OK

d) URL: POST records

Bảng 4.23 Bảng kiểm thử API thêm dữ liệu phiên đo

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Là admin hoặc bác sĩ, file hợp lệ,	Thông tin dữ liệu phiên đo <pre>{ "patient_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại dữ liệu phiên đo, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc, "data_rec_url": Đường dẫn file dữ liệu phiên đo }</pre>	Status code: 200 OK Response message: <pre>{ "message": "Record created successfully" }</pre>	OK
TC-2	Là admin hoặc bác sĩ, file không hợp lệ,	Thông tin dữ liệu phiên đo <pre>{ "patient_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại dữ liệu phiên đo, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc, "data_rec_url": Đường dẫn file dữ liệu phiên đo }</pre>	Status code: 400 Bad Request Response message: <pre>{ "message": "Invalid file type" }</pre>	OK

TC-3	Thông tin dữ liệu phiên đo không hợp lệ	Thông tin dữ liệu phiên đo { "patient_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại dữ liệu phiên đo, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc, "data_rec_url": Đường dẫn file dữ liệu phiên đo }	Status code: 400 Bad Request Response message: { "message": "Failed to create record" }	OK
------	-----------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------	----

e) URL: PUT records

Bảng 4.24 Bảng kiểm thử API cập nhật dữ liệu dữ liệu phiên đo

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Dữ liệu cập nhật phù hợp	Thông tin cập nhật dữ liệu phiên đo { "patient_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại dữ liệu phiên đo, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc, "data_rec_url": Đường dẫn file dữ liệu phiên đo }	Status code: 200 OK Response message: { "message": "Record updated successfully" }	OK

TC-2	Dữ liệu cập nhật không phù hợp	Thông tin cập nhật dữ liệu phiên đo { "patient_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại dữ liệu phiên đo, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc, "data_rec_url": Đường dẫn file dữ liệu phiên đo }	Status code: 404 Not Found Response message: { "message": "Error when update record" }	OK
------	--------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------	----

f) URL: DELETE records/:record_id

Bảng 4.25 Bảng kiểm thử API xóa dữ liệu dữ liệu phiên đo theo id

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Tồn tại dữ liệu phiên đo với id cho trước	id dữ liệu phiên đo	Status code: 200 OK Response message: { "message": "Record deleted successfully" }	OK
TC-2	Không tồn tại dữ liệu phiên đo với id cho trước	id dữ liệu phiên đo	Status code: 404 Not Found Response message: { "message": "No record found to delete, please try again" }	OK

4.3.1.5 API liên quan liên quan đến việc đặt lịch bác sĩ - bệnh nhân

a) URL: GET schedules

Bảng 4.26 Bảng kiểm thử API lấy tất cả lịch hẹn của các bác sĩ - các bệnh nhân

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Admin hệ thống	Access token tương ứng	Status code: 200 OK Response message: { data: Danh sách tất cả lịch hẹn }	OK
TC-2	Không phải admin hệ thống	NULL	Status code: 403 Forbidden Response message: { "message": "Forbidden" }	OK

b) URL: GET schedules/doctor-id

Bảng 4.27 Bảng kiểm thử API lấy danh sách lịch hẹn theo id bác sĩ

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Bác sĩ tồn tại trong hệ thống	id bác sĩ	Status code: 200 OK Response message: { data: Danh sách lịch hẹn của bác sĩ }	OK
TC-2	Bác sĩ không tồn tại trong hệ thống	id bác sĩ	Status code: 404 Not Found Response message: { "status": "error", "message": "No user found, please try again." }	OK

c) URL: GET schedules/patient-id

Bảng 4.28 Bảng kiểm thử API lấy danh sách lịch hẹn theo id bệnh nhân

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Bệnh nhân tồn tại trong hệ thống	id bác sĩ	Status code: 200 OK Response message: { data: Danh sách lịch hẹn của bệnh nhân }	OK
TC-2	Bệnh nhân không tồn tại trong hệ thống	id bác sĩ	Status code: 404 Not Found Response message: { "status": "error", "message": "No user found, please try again." }	OK

d) URL: POST schedules/create-by-doctor

Bảng 4.29 Bảng kiểm thử API tạo lịch hẹn bởi bác sĩ

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Các trường thông tin lịch hẹn hợp lệ	Thông tin lịch hẹn { "doctor_id": id bệnh nhân, "patient_id": id bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn "schedule_result": Kết quả lịch hẹn }	Status code: 200 OK Response message: { "message": "Schedule created successfully" }	OK

TC-2	Các trường thông tin lịch hẹn không hợp lệ	Thông tin lịch hẹn { "doctor_id": id bệnh nhân, "patient_id": id bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn "schedule_result": Kết quả lịch hẹn }	Status code: 400 Bad Request Response message: { "message": "Failed to create schedule" }	OK
------	--------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------	----

e) URL: POST schedules/create-by-patient

Bảng 4.30 Bảng kiểm thử API tạo lịch hẹn bởi bệnh nhân

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Các trường thông tin lịch hẹn hợp lệ	Thông tin lịch hẹn { "doctor_id": ID bệnh nhân, "patient_id": ID bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn }	Status code: 200 OK Response message: { "message": "Schedule created successfully" }	OK

TC-2	Lịch hẹn đang chờ phê duyệt của bệnh nhân vượt quá 5	Thông tin lịch hẹn { "doctor_id": ID bệnh nhân, "patient_id": ID bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn }	Status code: 400 Bad Request Response message: { "message": "Quá giới hạn lịch được đặt, vui lòng đợi các bác sĩ phê duyệt lịch đã đặt trước khi tiếp tục." }	OK
TC-3	Đặt lặp lịch hẹn (trùng thời gian và trùng bác sĩ)	Thông tin lịch hẹn { "doctor_id": ID bệnh nhân, "patient_id": ID bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn }	Status code: 400 Bad Request Response message: { "message": "Bạn đã đặt lịch vào thời điểm này trước đó, vui lòng đợi bác sĩ phê duyệt" }	OK
TC-4	Cùng 1 thời điểm, đặt lịch hẹn với các bác sĩ khác nhau	Thông tin lịch hẹn { "doctor_id": ID bệnh nhân, "patient_id": ID bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn }	Status code: 400 Bad Request Response message: { "message": "Bạn đã có lịch vào thời điểm này, vui lòng kiểm tra lại." }	OK

f) URL: GET schedules/time/available-doctor/:schedule_time

Bảng 4.31 Bảng kiểm thử API lấy danh sách bác sĩ có thể đặt lịch hẹn theo thời gian cụ thể

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Thời gian hợp lệ	Thời gian đã chọn trước	Status code: 200 OK Response message: { data: Danh sách các bác sĩ có thể đặt lịch hẹn }	OK

g) URL: GET schedules/available-schedule/:id

Bảng 4.32 Bảng kiểm thử API lấy danh sách thời gian có thể đặt lịch hẹn của bác sĩ cụ thể

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Bác sĩ tồn tại trong hệ thống	id của bác sĩ đã chọn	Status code: 200 OK Response message: { data: Danh sách các lịch hẹn có thể đặt với bác sĩ này }	OK

h) URL: PUT schedules/accept-schedule

Bảng 4.33 Bảng kiểm thử API bác sĩ chấp nhận lịch hẹn từ bệnh nhân

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Lịch hẹn của bệnh nhân đang ở trạng thái chờ phê duyệt	Thông tin lịch hẹn đang chờ phê duyệt { "doctor_id": id bệnh nhân, "patient_id": id bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn "schedule_result": Kết quả lịch hẹn }	Status code: 200 OK Response message: { "message": "Schedule accepted successfully" }	OK
------	--------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------	----

i) URL: DELETE schedules/reject-schedule/:id

Bảng 4.34 Bảng kiểm thử API bác sĩ từ chối lịch hẹn từ bệnh nhân

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Lịch hẹn của bệnh nhân đang ở trạng thái chờ phê duyệt	Thông tin lịch hẹn đang chờ phê duyệt { "doctor_id": id bệnh nhân, "patient_id": id bác sĩ, "schedule_start_time": Thời gian bắt đầu, "schedule_end_time": Thời gian kết thúc, "status_id": id trạng thái lịch hẹn "schedule_result": Kết quả lịch hẹn }	Status code: 200 OK Response message: { "message": "Schedule has been rejected successfully" }	OK

4.3.1.6 API liên quan liên quan đến chẩn đoán

1. URL: POST diagnosis/

Bảng 4.35 Bảng kiểm thử API tạo chẩn đoán mới

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Lịch hẹn của bệnh nhân đã được chấp nhận	Thông tin chẩn đoán của buổi khám { "schedule_id": ID lịch hẹn, "infomation": Thông tin chẩn đoán, }	Status code: 200 OK Response message: { "message": "Diagnosis created successfully" }	OK
TC-2	Lịch hẹn của bệnh nhân chưa được chấp nhận	Thông tin chẩn đoán của buổi khám { "schedule_id": ID lịch hẹn, "infomation": Thông tin chẩn đoán, }	Status code: 400 Bad Request Response message: { "message": "Failed to create diagnosis" }	OK

2. URL: GET diagnosis/schedule/:schedule_id

Bảng 4.36 Bảng kiểm thử API lấy thông tin chẩn đoán theo id lịch khám

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Lịch hẹn của bệnh nhân đã được chấp nhận	id lịch hẹn	Status code: 200 OK Response message: { data: dữ liệu chẩn đoán của buổi khám }	OK
TC-2	Lịch hẹn của bệnh nhân chưa được chấp nhận	id lịch hẹn	Status code: 400 Bad Request Response message: { "message": "No diagnosis found, please try again" }	OK

3. URL: POST diagnosis/update

Bảng 4.37 Bảng kiểm thử API cập nhật chẩn đoán theo id

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Thông tin cập nhật chẩn đoán hợp lệ	id chẩn đoán, thông tin chẩn đoán của buổi khám { "schedule_id": ID lịch hẹn, "infomation": Thông tin chẩn đoán, }	Status code: 200 OK Response message: { "message": "Diagnosis updated successfully" }	OK
TC-2	Thông tin cập nhật chẩn đoán không hợp lệ	id chẩn đoán, thông tin chẩn đoán của buổi khám { "schedule_id": ID lịch hẹn, "infomation": Thông tin chẩn đoán, }	Status code: 400 Bad Request Response message: { "message": "Error when update diagnosis by schedule id" }	OK

4.3.1.7 API liên quan liên quan đến thông báo

1. URL: GET notification/get

Bảng 4.38 Bảng kiểm thử API lấy thông báo theo id của người dùng

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Là bệnh nhân	id người dùng tương ứng	Status code: 200 OK Response message: { data: Danh sách thông báo của người dùng này }	OK

TC-2	Là bác sĩ của hệ thống	id người dùng tương ứng	Status code: 400 Bad Request Response message: { data: Danh sách thông báo của người dùng này }	OK
------	------------------------	-------------------------	-------------------------------------------------------------------------------------------------------------	----

2. URL: POST notification

Bảng 4.39 Bảng kiểm thử API tạo thông báo

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Là bác sĩ của hệ thống	Trạng thái của notification không phải là 1 hoặc 3	Status code: 200 OK Response message: { Trường doctor_id được gán giá trị là id của người dùng hiện tại "message": "Notification added successfully", }	OK
TC-2	Là bệnh nhân của hệ thống	Trạng thái của notification không phải là 1 hoặc 3	Status code: 200 OK Response message: { Trường patient_id được gán giá trị là id của người dùng hiện tại "message": "Notification added successfully", }	OK

TC-3	Là người dùng của hệ thống	Trạng thái của notification là 1 hoặc 3	Status code: 200 OK Response message: { Bỏ qua kiểm tra trùng lặp, tạo 2 thông báo và lưu. "message": "Notification added successfully", }	OK
------	----------------------------	-----------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------	----

3. URL: POST notification/update-seen

Bảng 4.40 Bảng kiểm thử API cập nhật trạng thái thông báo đã được xem

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Thông báo chưa được xem	id của thông báo	Status code: 200 OK Response message: { "message": "Seen status updated successfully", }	OK
TC-2	Thông báo đã được xem	id của thông báo	Status code: 400 Bad Request Response message: { "message": "400 Bad Request", }	OK

4. URL: DELETE notification/:id

Bảng 4.41 Bảng kiểm thử API xóa thông báo theo id

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Thông báo tồn tại trong hệ thống	id của thông báo	Status code: 200 OK Response message: { "message": "Seen status updated successfully", }	OK
TC-2	Thông báo không tồn tại trong hệ thống	id của thông báo	Status code: 400 Bad Request Response message: { "message": "400 Bad Request", }	OK

4.3.1.8 API liên quan liên quan đến tin nhắn

1. URL: POST groupChat

Bảng 4.42 Bảng kiểm thử API tạo nhóm trò chuyện

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Admin hoặc bác sĩ của hệ thống	Thông tin nhóm trò chuyện { "title": Tên nhóm trò chuyện, "hostId: id của người tạo nhóm, "member": Danh sách id của các thành viên trong nhóm trò chuyện, "sendEvent": Sự kiện gửi của nhóm trò chuyện "receiveEvent": Sự kiện nhận của nhóm trò chuyện }	Status code: 200 OK Response message: { data: thông tin của nhóm trò chuyện đã tạo, }	OK

TC-2	Là bệnh nhân của hệ thống (bệnh nhân chỉ có thể tạo đoạn chat 2 người với bác sĩ khám cho mình)	Thông tin nhóm trò chuyện { "title": Tên nhóm trò chuyện, "hostId: id của người tạo nhóm, "member": Danh sách id của các thành viên trong nhóm, "sendEvent": Sự kiện gửi của nhóm trò chuyện "receiveEvent": Sự kiện nhận của nhóm trò chuyện }	Status code: 200 OK Response message: { data: thông tin của đoạn chat đã tạo, }	OK
------	-------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------	----

2. URL: GET groupChat

Bảng 4.43 Bảng kiểm thử API tìm danh sách nhóm trò chuyện của người dùng

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC-1	Người dùng của hệ thống	id người dùng tương ứng	Status code: 200 OK Response message: { data: danh sách nhóm trò chuyện của người dùng này }	OK
TC-2	Không phải người dùng hệ thống	id của người dùng	Status code: 401 Unauthorized Response message: { "message": "401 Unauthorized", }	OK

3. URL: GET chat/messages/:groupChatId

Bảng 4.44 Bảng kiểm thử API lấy tin nhắn theo id nhóm trò chuyện

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC	Người dùng của hệ thống	id nhóm trò chuyện của người dùng	Status code: 200 OK Response message: { Tin nhắn của người dùng trong nhóm trò chuyện này được lấy đúng, Tin nhắn của các người dùng khác trong nhóm trò chuyện này được lấy đúng }	OK

4. URL: POST chat/send

Bảng 4.45 Bảng kiểm thử API gửi tin nhắn trong nhóm trò chuyện

Test case	Điều kiện	Đầu vào	Mong muốn đầu ra	Kết quả
TC	Người dùng của hệ thống	Thông tin tin nhắn { "senderId" : id người gửi, "groupId" : id nhóm trò chuyện, "message" : nội dung tin nhắn, "time": thời gian gửi }	Status code: 200 OK Tin nhắn được gửi đúng nhóm, Thời gian gửi được lưu vào cơ sở dữ liệu theo thời gian thực, Người nhận nhận được tin nhắn theo thời gian thực, đúng nhóm trò chuyện, đúng người gửi	OK

4.3.2 Kiểm thử ứng dụng web

Bảng 4.46 Bảng kiểm thử chức năng của website quản trị

Test case	Tái hiện	Kết quả mong muốn	Đánh giá
-----------	----------	-------------------	----------

Kiểm tra chức năng đăng nhập	1. Truy cập giao diện đăng nhập tài khoản → Nhập thông tin đăng nhập chưa có trong hệ thống → Nhấn nút đăng nhập 2. Truy cập giao diện đăng nhập tài khoản → Nhập thông tin đăng nhập hợp lệ → Nhấn nút đăng nhập	1. Hiện thị thông báo sai email hoặc mật khẩu 2. Đăng nhập thành công vào hệ thống	OK
Kiểm tra chức năng đăng ký	1. Truy cập giao diện đăng ký tài khoản mới → Nhập thông tin đăng ký tài khoản đã tồn tại → Nhấn đăng ký 2. Truy cập giao diện đăng ký tài khoản mới → Nhập các thông tin tài khoản đăng ký hợp lệ → Chọn nút đăng ký	1. Hiện thị thông báo email đã tồn tại trong hệ thống 2. Hiện thị thông báo tài khoản đã đăng ký thành công	OK
Kiểm tra chức năng quản lý người dùng	1. Truy cập màn hình quản lý người dùng 2. Xem thông tin người dùng 3. Sửa thông tin người dùng có sẵn → Lưu thay đổi 4. Xóa một người dùng khỏi hệ thống	1. Hiện thị danh sách người dùng 2. Hiện thị thông tin cụ thể của người dùng 3. Cập nhật thông tin người dùng thành công 4. Xóa tài khoản người dùng khỏi hệ thống thành công	OK
Kiểm tra chức năng xem thông tin bác sĩ	1. Truy cập màn hình thông tin bác sĩ 2. Xem thông tin cụ thể của bác sĩ	1. Hiện thị danh sách bác sĩ 2. Hiện thị thông tin cụ thể của bác sĩ	OK
Kiểm tra chức năng xem danh sách bệnh nhân	1. Truy cập màn hình quản lý bệnh nhân 2. Hiện thị thông tin cụ thể bệnh nhân	1. Hiện thị danh sách bệnh nhân 2. Hiện thị thông tin cụ thể của bệnh nhân	OK

Kiểm tra chức năng quản lý thiết bị	<ol style="list-style-type: none"> 1. Truy cập màn hình quản lý thiết bị 2. Xem thông tin chi tiết thiết bị 3. Thêm thiết bị mới với thông tin hợp lệ → Lưu thông tin 4. Sửa thông tin thiết bị có sẵn → Lưu thay đổi 5. Xóa thiết bị (Đối với Admin hệ thống) 	<ol style="list-style-type: none"> 1. Hiện thị danh sách thiết bị 2. Hiện thị thông tin chi tiết thiết bị 3. Thêm thiết bị thành công 4. Cập nhật thông tin thiết bị thành công 5. Xóa thiết bị khỏi hệ thống thành công 	OK
Kiểm tra chức năng quản lý đặt lịch hẹn bởi bệnh nhân	<ol style="list-style-type: none"> 1. Vào màn hình đặt lịch hẹn 2. Đặt lịch hẹn 3. Chọn bác sĩ hoặc giờ khám 4. Lưu 	<ol style="list-style-type: none"> 1. Hiện thị giao diện lịch, bao gồm những ngày trống, những ngày có lịch hẹn 2. Hiện thị giao diện đặt lịch 3. Giờ khám hoặc bác sĩ phù hợp sẽ hiển thị 4. Lịch hẹn được lưu thành công 	OK
Kiểm tra chức năng quản lý đặt lịch tái khám bởi bác sĩ	<ol style="list-style-type: none"> 1. Chọn 1 lịch khám đã thành công 2. Chọn đặt lịch tái khám 3. Chọn ngày giờ phù hợp 4. Lưu 	<ol style="list-style-type: none"> 1. Hiện thị giao diện thông tin chẩn đoán 2. Giao diện được mở rộng để đặt lịch tái khám 3. Thông tin lịch hẹn hiển thị trên màn hình 4. Lịch hẹn được lưu thành công 	OK
Kiểm tra chức năng quản lý thông tin cá nhân	<ol style="list-style-type: none"> 1. Truy cập màn hình quản lý thông tin cá nhân 2. Sửa thông tin cá nhân có sẵn → Lưu thông tin 	<ol style="list-style-type: none"> 1. Hiện thị thông tin tài khoản cá nhân 2. Cập nhật thông tin cá nhân thành công 	OK
Kiểm tra chức năng nhắn tin	<ol style="list-style-type: none"> 1. Truy cập màn hình nhắn tin 2. Chọn nhóm/ người dùng cần nhắn 3. Nhắn tin 	<ol style="list-style-type: none"> 1. Hiện thị danh sách các đoạn tin nhắn 2. Hiện thị tin nhắn lên màn hình 3. Tin nhắn được gửi thành công 	OK

Kiểm tra chức năng tạo nhóm trò chuyện	1. Truy cập màn hình nhắn tin 2. Chọn tạo nhóm trò chuyện 3. Nhập các thông tin, chọn thành viên phù hợp 4. Lưu	1. Hiển thị danh sách các đoạn tin nhắn, nút tạo nhóm 2. Hiển thị giao diện tạo nhóm trò chuyện 3. Thông tin thành viên phù hợp hiển thị lên màn hình 4. nhóm trò chuyện được tạo thành công	OK

4.4 Kết luận chương

Chương 4 đã trình bày chi tiết về quá trình triển khai hệ thống và các nền tảng, công cụ được sử dụng trong hệ thống. Đồng thời, chương cũng trình bày về quá trình kiểm thử hệ thống, bao gồm kiểm thử API và kiểm thử ứng dụng web. Nhờ đó, hệ thống đã được kiểm thử kỹ lưỡng trước khi triển khai, giúp đảm bảo tính ổn định và chất lượng của hệ thống.

PHỤ LỤC

Công cụ hỗ trợ trong đồ án Github

a) Giới thiệu chung

GitHub là một nền tảng lưu trữ mã nguồn và quản lý phiên bản sử dụng Git, được ra mắt vào năm 2008. Được thiết kế để hỗ trợ cộng đồng phát triển phần mềm, GitHub cho phép các nhà phát triển lưu trữ, quản lý, và theo dõi các dự án phần mềm một cách hiệu quả. Hiện tại, GitHub là một trong những nền tảng phổ biến nhất cho phát triển phần mềm mã nguồn mở và dự án doanh nghiệp.[13]

b) Cách sử dụng github

Tạo tài khoản GitHub: Truy cập GitHub theo đường dẫn <https://github.com/> và đăng ký một tài khoản miễn phí,

Tạo một kho lưu trữ: Người dùng cần phải tạo các kho lưu trữ để lưu trữ code trong dự án của mình.

Làm việc với dữ liệu trong kho lưu trữ: Người dùng có thể thêm, sửa đổi hoặc xóa các tệp tin trong kho lưu trữ của mình.

Tạo nhánh và quản lý nhánh: Khi có nhiều người dùng sử dụng một kho lưu trữ có thể tạo các nhánh riêng biệt để thay đổi code và ghép các nhánh đó vào nhánh chính sau khi công việc đã hoàn thành.

Đường dẫn mã nguồn

Link mã nguồn: <https://github.com/techcomrade/fmECG>

TÀI LIỆU THAM KHẢO

- [1] Javascript ecosystem overview, 2016.
- [2] Intro to bluetooth generic attribute profile (gatt), 2017.
- [3] About: Mysql, 2023.
- [4] Adminjs - adminjs, 2023.
- [5] Flutter performance profiling | flutter, 2023.
- [6] Introduction to bluetooth low energy | adafruit learning system, 2023.
- [7] Save simple data with sharedpreferences | android developers, 2023.
- [8] What is asynchronous javascript and xml (ajax)?, 2023.
- [9] What is ci/cd? | gitlab, 2023.
- [10] What is mysql ? | definition from techtarget, 2023.
- [11] What is postman (a tutorial for beginners), 2023.
- [12] Docker tutorial, 2024.
- [13] Github docs, 2024.
- [14] Reactjs docs: Getting started, 2024.
- [15] <https://cloud.z.com/>. Ci/cd là gì? ci/cd có liên hệ gì tới agile và devops?, 2023.
- [16] <https://giadinh.suckhoedoisong.vn>. Chuyện 1 bệnh nhân covid-19 không chịu “nằm không”, 2023.
- [17] Iryna Kurkina. What is node.js and why you should use it - academy smart, 2023.
- [18] Canh Toan Nguyen, Hoa Phung, Phi Tien Hoang, Tien Dat Nguyen, Hosang Jung, and Hyouk Ryeol Choi. Development of an insect-inspired hexapod robot actuated by soft actuators. *Journal of Mechanisms and Robotics*, 10(6), 2018.
- [19] Dang Thi Nhu Y, Nguyen Tien Hoang, Pham Khac Lieu, Hidenori Harada, Natacha Brion, Duong Van Hieu, Nguyen Van Hop, and Harry Olde Venterink. Effects of nutrient supply and nutrient ratio on diversity–productivity relationships of phytoplankton in the cau hai lagoon, vietnam. *Ecology and evolution*, 9(10):5950–5962, 2019.

[20] Benjamin Semah. What exactly is node.js? explained for beginners, 2022.

[21] Team TIS. The benefits of node.js for web application development?, 2022.