**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

A picture containing text

Description automatically generated

**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:**

**XÂY DỰNG PHÒNG KHÁM DỰA TRÊN HỢP ĐỒNG THỎA THUẬN**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | : TS. Tôn Long Phước |
| Sinh viên thực hiện | :20115331 – Nguyễn Văn Ngọ |
|  | :20118191 – Nguyễn Thanh Sang |
| Lớp | : DHKTPM16GTT |

**TP. HỒ CHÍ MINH – NĂM 2024**

**MINISTRY OF INDUSTRY AND TRADE INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY**

A picture containing text

Description automatically generated

**GRADUATION THESIS**

**TOPIC:**

**BUILDING A CLINIC BASED ON AGREEMENT CONTRACT**

|  |  |
| --- | --- |
| Instructors | : TS. Tôn Long Phước |
| Students perform | :20115331 – Nguyễn Văn Ngọ |
|  | :20118191 – Nguyễn Thanh Sang |
| Class | : DHKTPM16GTT |

**HO CHI MINH CITY – YEAR 2023**

**ABSTRACT**

**Abstract.** Growing demands for modern health services in urban areas have increased the demand for a secure, transparent, and efficient clinic management system. In this regard, we propose a blockchain-based healthcare management application for the improvement of overall experiences of the patients and facilitation in operations of medical professionals. We integrate advanced technologies like blockchain for managing data securely; smart contracts-automatically handle transactions; AI-based chatbots-can support the patients. The research proposes blockchain-based healthcare management for improving patients and smoothening clinic operations. The proposed system leverages blockchain for securing data, smart contracts for automation, and AI chatbots for support to patients. Some of the key features of the system include online appointment management, management of medical records, and doctor-patient communication. Next.js, NestJS, and Solidity are used for the system, and it shall be deployed on cloud infrastructure, which guarantees performance and security. Preliminary results include increased efficiency and more transparency, with scaling up and integration of advanced analytics to be continued in the future.

**Keywords.** Health care, RAG chatbot, NextJs, NestJs, MongoDB

**MỤC LỤC**

[DANH MỤC CÁC HÌNH ẢNH 1](#_Toc135061223)

[DANH MỤC CÁC BẢNG 3](#_Toc135061224)

[CHƯƠNG 1: NHẬT KÝ THỰC HIỆN ĐỀ TÀI 4](#_Toc135061225)

[CHƯƠNG 2: GIỚI THIỆU 9](#_Toc135061226)

[2.1 Tổng quan: 9](#_Toc135061227)

[2.2 Mục tiêu đề tài: 10](#_Toc135061228)

[2.3 Phạm vi đề tài: 10](#_Toc135061229)

[2.4 Mô tả yêu cầu chức năng: 11](#_Toc135061230)

[2.5 Mô tả yêu cầu phi chức năng: 12](#_Toc135061231)

[CHƯƠNG 3: CƠ SỞ LÝ THUYẾT 13](#_Toc135061232)

[3.1 Kiến trúc của hệ thống: 13](#_Toc135061233)

[3.1.1 RESTful API: 13](#_Toc135061234)

[3.1.2 Kiến trúc ứng dụng: 14](#_Toc135061235)

[3.2 Điện toán đám mây: 14](#_Toc135061236)

[3.2.1 Firebase Authentication: 15](#_Toc135061237)

[3.2.2 Cloudinary: 15](#_Toc135061238)

[3.2.3 ZegoClound: 16](#_Toc135061239)

[3.2.4 MongoDB: 17](#_Toc135061240)

[3.3 Ngôn ngữ lập trình, Framework và công nghệ hỗ trợ: 17](#_Toc135061241)

[3.3.1 Ngôn ngữ lập trình Javascript: 17](#_Toc135061242)

[3.3.2 NodeJS: 18](#_Toc135061243)

[3.3.3 Framework React native: 19](#_Toc135061244)

[3.3.4 Framework ReactJS: 20](#_Toc135061245)

[3.3.5 Framework ExpressJS: 21](#_Toc135061246)

[3.3.6 SocketIO: 22](#_Toc135061247)

[3.3.7 JWT: 23](#_Toc135061248)

[3.3.8 ChartJS: 24](#_Toc135061249)

[3.3.9 ChatRAG: 25](#_Toc135061250)

[CHƯƠNG 4: PHÂN TÍCH 26](#_Toc135061251)

[4.1 Phân tích yêu cầu bằng UML (Unified Modeling Language): 26](#_Toc135061252)

[4.1.1 Sơ đồ usecase tổng quát: 26](#_Toc135061253)

[4.1.2 Danh sách các tác nhân và mô tả: 27](#_Toc135061254)

[4.1.3 Danh sách các usecase và mô tả 29](#_Toc135061255)

[4.1.4 Đặc tả usecase, sơ đồ activity, sơ đồ sequence 39](#_Toc135061256)

[4.2 Sơ đồ lớp: 71](#_Toc135061257)

[4.3 Sơ đồ deployment: 72](#_Toc135061258)

[4.4 Sơ đồ luồng màn hình: 72](#_Toc135061259)

[4.4.1 Sơ luồng màn hình mobile: 72](#_Toc135061260)

[4.4.2 Sơ đồ luồng màn hình web: 73](#_Toc135061261)

[CHƯƠNG 5: THIẾT KẾ VÀ HIỆN THỰC 74](#_Toc135061262)

[5.1 Giao diện thiết kế: 74](#_Toc135061263)

[5.1.1 Giao diện mobile: 74](#_Toc135061264)

[5.1.2 Giao diện web: 76](#_Toc135061265)

[5.2 Giao diện hiện thực: 84](#_Toc135061266)

[5.2.1 Giao diện mobile: 84](#_Toc135061267)

[5.2.2 Giao diện web: 92](#_Toc135061268)

[CHƯƠNG 6: KẾT LUẬN 115](#_Toc135061269)

[6.1 Kết quả đạt được: 115](#_Toc135061270)

[6.2 Hạn chế của đề tài: 115](#_Toc135061271)

[6.3 Hướng phát triển: 115](#_Toc135061272)

[TÀI LIỆU THAM KHẢO 116](#_Toc135061273)

# DANH MỤC CÁC HÌNH ẢNH

[3.1. 1 Mô hình RESTful API [1] 13](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005021)

[3.1. 2 Mô hình kiến trúc ứng dụng 14](#_Toc135005022)

[3.2. 1 Firebase Authentication là gì?[2]……………………………………………………. 15](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005081)

[3.2. 2 Cloudinary là gì? [3] 16](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005082)

[3.2. 3 ZegoCloud là gì? [4] 16](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005083)

[3.2. 4 MongoDB là gì? [5] 17](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005084)

[3.3.1 Javascript là gì?[6]……………………………………………………………..... ……..18](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005164)

[3.3. 2 NodeJS là gì? [7] 19](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005165)

[3.3. 3 React native là gì? [8] 20](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005166)

[3.3. 4 ReactJS là gì? [9] 21](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005167)

[3.3. 5 ExpressJS là gì? [10] 22](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005168)

[3.3. 6 SocketIO là gì? [11] 23](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005169)

[3.3. 7 Các thành phần của JWT [12] 23](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005170)

[3.3. 8 Chart.js là gì? [13] 24](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135005171)

[3.3. 9 Chat RAG là gì? [14] 25](#_Toc135005172)

[4.1. 1 Sơ đồ usecase tổng quát ……………………………………………………………. 26](#_Toc135005305)

[4.1.4.1. 1 Sơ đồ activity Đăng ký tài khoản cho bác sĩ ………………………………………41](#_Toc135006579)

[4.1.4.1. 2 Sơ đồ sequence Đăng ký tài khoản cho bác sĩ 42](#_Toc135006580)

[4.1.4.2. 1 Sơ đồ activity Đặt lịch khám bởi bệnh nhân…………………………………. …..44](#_Toc135006592)

[4.1.4.2. 2 Sơ đồ sequence Đặt lịch khám bởi bệnh nhân 45](#_Toc135006593)

[4.1.4.3. 1 Sơ đồ activity Hủy lịch khám bởi bệnh nhân………………………………………47](#_Toc135006603)

[4.1.4.3. 2 Sơ đồ sequence Hủy lịch khám bởi bệnh nhân 48](#_Toc135006604)

[4.1.4.4. 1 Sơ đồ activity Khám bệnh qua video call của bệnh nhân……………………………51](#_Toc135006613)

[4.1.4.4. 2 Sơ đồ sequence Khám bệnh qua video call 52](#_Toc135006614)

[4.1.4.5. 1 Sơ đồ activity Xác nhận lịch khám bởi bác sĩ……………………………………..53](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135006622)

[4.1.4.5. 2 Sơ đồ sequence Xác nhận lịch khám bởi bác sĩ 54](#_Toc135006623)

[4.1.4.6. 1 Sơ đồ activity Hủy lịch khám bởi bác sĩ…………………………………………..56](#_Toc135006632)

[4.1.4.6. 2 Đặc tả usecase Hủy lịch khám bởi bác sĩ 57](#_Toc135006633)

[4.1.4.7. 1 Sơ đồ activity Phản hồi bệnh nhân sau khi khám………………………………… 59](#_Toc135006640)

[4.1.4.7. 2 Sơ đồ sequence Phản hồi bệnh nhân sau khi khám 60](#_Toc135006641)

[4.1.4.8. 1 Sơ đồ activity Gửi tin nhắn…………………………………………………………61](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135006649)

[4.1.4.8. 2 Sơ đồ sequence Gửi tin nhắn 62](#_Toc135006650)

[4.1.4.9. 1 Sơ đồ activity Chatbot tự động………………………………………….…............63](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135006656)

[4.1.4.9. 2 Sơ đồ sequence Chatbot tự động 64](#_Toc135006657)

[4.1.4.10. 1 Sơ đồ activity Đánh giá bác sĩ sau khi khám …………………………………….65](#_Toc135006666)

[4.1.4.10. 2 Sơ đồ sequence Đánh giá bác sĩ sau khi khám 66](#_Toc135006667)

[4.1.4.11. 1 Sơ đồ activity Nhắc nhở sức khỏe bệnh nhân……………………………………67](#_Toc135006676)

[4.1.4.11. 2 Sơ đồ sequence Nhắc nhở sức khỏe bệnh nhân 68](#_Toc135006677)

[4.1.4.12. 1 Sơ đồ activity Xem đánh giá tổng quan bệnh nhân………………………………70](#_Toc135006684)

[4.1.4.12. 2 Sơ đồ sequence Xem đánh giá tổng quan bệnh nhân 70](#_Toc135006685)

[4.4. 1 Sơ đồ luồng màn hình mobile ………………………………………………………….72](#_Toc135007111)

[4.4. 2 Sơ đồ luồng màn hình web 73](file:///C:\Users\Bao%20Tran\Downloads\N44_LTuan_BTran_BAO_CAO_DO_AN_T&T_HEATHCARE_V1.docx#_Toc135007112)

# DANH MỤC CÁC BẢNG

[Bảng 4.1.2 Danh sách các tác nhân và mô tả 29](#_Toc133776171)

[Bảng 4.1.3 Danh sách các usecase và mô tả 38](#_Toc133776172)

[Bảng 4.1.4.1 Đặc tả usecase Đăng ký tài khoản cho bác sĩ 40](#_Toc133776173)

[Bảng 4.1.4.2 Đặc tả usecase Đặt lịch khám bởi bệnh nhân 44](#_Toc133776174)

[Bảng 4.1.4.3 Đặc tả usecase Hủy lịch khám bởi bệnh nhân 47](#_Toc133776175)

[Bảng 4.1.4.4 Đặc tả usecase Khám bệnh qua video call 50](#_Toc133776176)

[Bảng 4.1.4.5 Đặc tả usecase Xác nhận lịch khám bởi bác sĩ 53](#_Toc133776177)

[Bảng 4.1.4.6 Đặc tả usecase Hủy lịch khám bởi bác sĩ 55](#_Toc133776178)

[Bảng 4.1.4.7 Đặc tả usecase Nhắc nhở bệnh nhân sau khi khám bệnh 58](#_Toc133776179)

[Bảng 4.1.4.8 Đặc tả usecase Gửi tin nhắn 61](#_Toc133776180)

[Bảng 4.1.4.9 Đặc tả usecase Chatbot tự động 63](#_Toc133776181)

[Bảng 4.1.4.10 Đặc tả usecase Đánh giá bác sĩ 65](#_Toc133776182)

[Bảng 4.1.4.11 Đặc tả usecase Nhắc nhở sức khỏe bệnh nhân 67](#_Toc133776183)

[Bảng 4.1.4.12 Đặc tả usecase Xem đánh giá tổng quan bệnh nhân 69](#_Toc133776184)

# CHƯƠNG 1: NHẬT KÝ THỰC HIỆN ĐỀ TÀI

|  |  |  |  |
| --- | --- | --- | --- |
|  | Công việc | Thành viên thực hiện | Ghi chú |
| Tuần 1  Từ: 16/08/2024  Đến: 22/08/2024 | Tiếp nhận đề tài:  - Liên hệ giáo viên hướng dẫn thầy Tôn Long Phước.  - Phân chia công việc nhóm (chi tiết các công việc cần làm cụ thể từng công việc).  - Tìm hiểu các hệ thống và giải pháp tương tự có thể tham khảo.  - Tạo Trello quản lý công việc hàng tuần | Nguyễn Văn Ngọ  Nguyễn Thanh Sang | Tổ chức họp nhóm bàn bạc kế hoạch thực hiện đề tài |
| Tuần 2  Từ: 23/08/2023  Đến: 29/08/2023 | - Phân tích yêu cầu của đề tài  - Giới hạn mục tiêu đề tài  - Thu thập ý kiến thực tế các bệnh nhân  - Tìm tài liệu từ các nguồn như: Bộ Y tế Việt Nam, …  - Tìm hiểu các nghiệp vụ cần thiết.  - Xác định các quy tắc nghiệp vụ  - Xác định yêu cầu chức năng, phi chức năng.  - Phân tích kiến trúc của hệ thống | Nguyễn Văn Ngọ  Nguyễn Thanh Sang | Tổ chức họp nhóm bàn bạc phân tích thực hiện đề tài. |
| Tuần 3  Từ: 30/08/2024  Đến: 06/09/2023 | Vẽ đặc tả các sơ đồ:  - Sơ đồ usecase  - Sơ đồ activity  - Sơ đồ sequence  - Sơ đồ lớp  - Sơ đồ quan hệ | Nguyễn Văn Ngọ  Nguyễn Thanh Sang | Tổ chức họp nhóm |
| Tuần 4  Từ: 07/09/2024  Đến: 13/09/2024 | Thiết kế giao diện ứng dụng | Nguyễn Thanh Sang | Tổ chức họp nhóm nêu rõ các vấn đề để cho phía backend và front-end thống nhất |
| Thiết kế hệ thống:  - Thiết kế cơ sở dữ liệu  - Thiết kế API | Nguyễn Thanh Sang |
| Tuần 5  Từ: 14/09/2024  Đến: 20/09/2024 | Hiện thực giao diện web phân quyền bác sĩ, admin:  - Giao diện đăng ký  - Giao diện tạo thông tin bác sĩ: Thông tin cá nhân, hồ sơ làm việc, …  - Giao diện admin: Hiển thị danh sách bác sĩ, tạo lịch, tạo ca làm, … | Nguyễn Văn Ngọ | Tổ chức họp nhóm đánh giá kết quả trong tuần |
| Hiện thực các API:  - Đăng ký  - Đăng nhập  - Tạo thông tin người dùng  - Duyệt bác sĩ bởi admin | Nguyễn Thanh Sang |
| Tuần 6  Từ: 21/09/2024  Đến: 27/09/2024 | Hiện thực các giao diện web phân quyền bác sĩ:  - Giao diện quản lý ca làm  - Giao diện quản lý lịch khám  - Tích hợp API của backend đã xây dựng ở tuần trước | Nguyễn Văn Ngọ | Tổ chức họp nhóm đánh giá kết quả trong tuần |
| Hiện thực các API liên quan tới lịch khám & ca làm của bác sĩ:  - Đăng ký ca làm  - Trả về danh sách ca làm đã đăng ký  - Đăng ký lịch khám bệnh đối với bệnh nhân  - Hủy lịch khám đối với bệnh nhân & bác sĩ  - Xác nhận lịch khám đối với bác sĩ | Nguyễn Thanh Sang |  |
| Tuần 7  Từ: 28/10/2024  Đến: 05/10/2024 | Tích hợp các API liên quan đến lịch khám và ca làm  Hiện thực giao diện xem và quản lý thông báo | Nguyễn Văn Ngọ | Tổ chức họp nhóm đánh giá kết quả trong tuần, phản hồi các thắc mắc |
| Hiện thực giao diện đăng ký, đăng nhập, tạo thông tin.  Tích hợp các API của tuần 5 | Nguyễn Thanh Sang |
| Tuần 8  Từ: 06/10/2024  Đến: 13/10/2024 | Hiện thực các API trò chuyện, nhắc nhở bệnh nhân, quản lý thông báo, quản lý bệnh nhân  Hiện thực giao diện về lịch khám, đăng ký lịch khám  Tích hợp API của tuần 6 | Nguyễn Thanh Sang | Tổ chức họp nhóm đánh giá kết quả trong tuần, phản hồi các thắc mắc |
| Tích hợp các API quản lý thông báo, trò chuyện và nhắc nhở bệnh nhân  Xây dựng giao diện quản lý bệnh nhân đối với bác sĩ | Nguyễn Văn Ngọ |
| Tuần 9  Từ: 14/10/2024  Đến: 21/10/2024 | Hiện thực các API về quản lý chỉ số sức khỏe: BMI, đường huyết, huyết áp.  Hiện thực giao diện nhập và hiển thị các chỉ số sức khỏe, quản lý thông báo đối với bệnh nhân.  Tích hợp API về quản lý chỉ số sức khỏe. | Nguyễn Thanh Sang | Tổ chức họp nhóm đánh giá kết quả trong tuần, phản hồi các thắc mắc |
| Tích hợp các API về quản lý bệnh nhân.  Hiện thực giao diện và chức năng gọi điện để khám | Nguyễn Văn Ngọ |
| Tuần 10  Từ: 28/10/2024  Đến: 05/11/2024 | Hiện thực giao diện và chức năng nhận cuộc gọi từ bác sĩ  Tích hợp ChatRAG để làm chatbot  Hiện thực các API quản lý bài viết của bác sĩ | Nguyễn Thanh Sang | Tổ chức họp nhóm đánh giá kết quả trong tuần, phản hồi các thắc mắc |
| Hiện thực giao diện quản lý bài viết đối với bác sĩ  Tích hợp API quản lý bài viết | Nguyễn Văn Ngọ |
| Tuần 11  Từ: 06/11/2024  Đến: 13/11/2024 | Hiện thực giao diện xem bài viết của bác sĩ, bình luận, thả cảm xúc.  Tích hợp ChatRAG để làm chatbot  Tích hợp API | Nguyễn Thanh Sang | Tổ chức họp nhóm đánh giá kết quả trong tuần, phản hồi các thắc mắc |
| Hiện thực giao diện web với phân quyền bệnh nhân  Tích hợp các API đã có sẵn | Nguyễn Văn Ngọ |  |
| Tuần 12  Từ: 14/11/2024  Đến: 21/11/2024 | Hiện thực giao diện app với phân quyền bác sĩ với các chức năng nhằm hỗ trợ web như:  - Nhận thông báo  - Xem danh sách bệnh nhân  - Tạo nhắc nhở  - Quản lý lịch khám  - Gọi điện khám  - Trò chuyện, tư vấn | Nguyễn Thanh Sang | Tổ chức họp nhóm đánh giá kết quả trong tuần, phản hồi các thắc mắc |
| Tuần 13  Từ: 22/11/2024  Đến: 29/11/2024 | Kiểm tra lại các chức năng và tiến hành sửa lỗi  Chuẩn bị tài liệu | Nguyễn Văn Ngọ  Nguyễn Thanh Sang | Đầu tuần chia nhau kiểm thử, đánh giá chức năng của nhau. |
| Tuần 14  Từ: 06/11/2024  Đến: 13/11/2024 | Hoàn thiện tất cả công việc còn lại:  - Hoàn tất đề tài, nộp theo yêu cầu của Giảng viên.  - Chuẩn bị PPT để báo cáo  - Tiến hành triển khai web trên host, app xuất file cài android | Nguyễn Văn Ngọ  Nguyễn Thanh Sang | Tiến hành đóng gói và triển khai ứng dụng |
| Tuần 15  20/11/2024 -27/11/2024 | Báo cáo đề tài trước Giảng viên hướng dẫn và các nhóm khác | Nguyễn Văn Ngọ  Nguyễn Thanh Sang | Tiến hành báo cáo |

# CHƯƠNG 2: GIỚI THIỆU

## 2.1 Tổng quan:

Trong bối cảnh nhu cầu chăm sóc sức khỏe ngày càng tăng tại các thành phố lớn, việc xây dựng và quản lý các phòng khám hiện đại không chỉ đơn thuần là cung cấp các dịch vụ y tế truyền thống, mà còn phải đảm bảo trải nghiệm tiện lợi, minh bạch và bảo mật cho cả bệnh nhân và đội ngũ y tế. Đặc biệt, với số lượng lớn bệnh nhân cần theo dõi định kỳ và các yêu cầu ngày càng cao về chất lượng dịch vụ, một hệ thống quản lý y tế thông minh là điều cần thiết để đáp ứng các thách thức này [6].

Hệ thống phòng khám chăm sóc sức khỏe của chúng tôi được thiết kế nhằm mang lại trải nghiệm tối ưu cho tất cả người dùng. Đối với bệnh nhân, hệ thống cung cấp các tính năng như đặt lịch hẹn trực tuyến, lưu trữ và truy xuất hồ sơ bệnh án, theo dõi các chỉ số sức khỏe. Đối với bác sĩ, hệ thống hỗ trợ quản lý hồ sơ y tế của bệnh nhân, tổ chức lịch làm việc, và đưa ra các gợi ý chuyên môn dựa trên dữ liệu y khoa. Tất cả những điều này nhằm tăng hiệu quả làm việc, giảm thời gian chờ đợi và nâng cao chất lượng chăm sóc sức khỏe.

Ngoài ra, việc quản lý dữ liệu y tế cần đảm bảo tính minh bạch và bảo mật cao. Những rủi ro liên quan đến việc lạm dụng thông tin cá nhân hay sai sót trong quá trình xử lý dữ liệu là những vấn đề lớn mà ngành y tế hiện đại phải đối mặt. Chính vì vậy, chúng tôi đã tích hợp các công nghệ tiên tiến để giải quyết những thách thức này.

Blockchain được ứng dụng để lưu trữ và quản lý dữ liệu y tế trong hệ thống. Với các tính năng như tính phi tập trung, không thể thay đổi và minh bạch, công nghệ này giúp đảm bảo rằng mọi thông tin y tế đều được ưu trữ an toàn và chỉ có thể được truy cập bởi những người được cấp quyền. Điều này không chỉ bảo vệ quyền riêng tư của bệnh nhân mà còn tăng cường sự tin cậy giữa các bên liên quan.

Để nâng cao trải nghiệm người dùng, hệ thống tích hợp AI chatbot dựa trên công nghệ RAG (Retrieval Augmented Generation). Chatbot này không chỉ hỗ trợ bệnh nhân trong việc đặt lịch hẹn hoặc tra cứu thông tin mà còn có khả năng cung cấp tư vấn y tế ban đầu dựa trên dữ liệu và các tài liệu y khoa.

## 2.2 Mục tiêu đề tài:

Ứng dụng phát triển nhằm giúp cho:

* **Bệnh nhân:** sau khi đăng ký tài khoản trên Web hoặc App NS Healthcare – Chăm sóc sức khỏe thông minh, họ có thể thực hiện nhiều chức năng trong hệ thống. Bệnh nhân có thể đăng ký lịch tư vấn khám chữa bệnh, xem lại hồ sơ bệnh án và toa thuốc trước đây, nhận tư vấn tự động qua chatbot RAG, cập nhật tin tức mới và các bài blog từ bác sĩ, cũng như quản lý thông tin cá nhân.
* **Bác sĩ:** sau khi đăng ký tài khoản trên Web và được Admin xác thực, họ có thể truy cập hệ thống để sử dụng các tính năng hỗ trợ công việc. Bác sĩ có thể xem các chỉ số thống kê tổng quan, quản lý lịch hẹn và danh sách bệnh nhân, thực hiện khám chữa bệnh, tạo hồ sơ bệnh án, kê toa thuốc, và yêu cầu các dịch vụ như siêu âm, xét nghiệm. Ngoài ra, họ có thể đăng bài trên blog, quản lý danh sách bệnh nhân cùng các biểu đồ thống kê, xem thông báo đặt lịch và quản lý thông tin cá nhân.
* **Admin:** việc đăng nhập hệ thống giúp quản lý toàn bộ hoạt động của NS Healthcare. Admin có thể kiểm duyệt và kích hoạt tài khoản bác sĩ, xem thống kê tổng quan về bác sĩ, bệnh nhân, doanh thu, thuốc và dịch vụ phổ biến. Ngoài ra, họ đảm nhiệm thanh toán hóa đơn cho bệnh nhân, kiểm tra bệnh nhân đã đặt lịch hẹn, và đảm bảo vận hành suôn sẻ của hệ thống.

## 2.3 Phạm vi đề tài:

* Phạm vi sử dụng:
* Các bệnh nhân ở TP.HCM.
* Các bác sĩ chuyên khoa ở TP.HCM.

## 2.4 Mô tả yêu cầu chức năng:

* **Bệnh nhân:** Tiến hành đăng ký tài khoản trên Web hoặc App (NS Healthcare – Chăm sóc sức khỏe thông minh). Sau khi đăng ký thành công tài khoản, bệnh nhân có thể thực hiện các chức năng của hệ thống:
* Đăng ký lịch tư vấn khám chữa bệnh.
* Xem lại các hồ sơ bệnh án, toa thuốc đã được kê trước đó.
* Được tư vấn tự động thông qua RAG chatbot.
* Xem được các tin tức mới, các bài Blog của các bác sĩ.
* Quản lý các thông tin cá nhân.
* **Bác sĩ:** Tiến hành đăng ký tài khoản trên Web (NS Healthcare). Sau khi đăng ký thành công tài khoản (đợi xác thực từ bên Admin), sau khi được xác thực tài khoản thành công thì bác sĩ có thể đăng nhập được vào hệ thống và có thể thực hiện được các chức năng bên trong hệ thống:
* Xem được các chỉ số thống kê (Tất cả các thông tin tổng quan ở phần Bảng điều khiển).
* Quản lý được lịch hẹn tư vấn khám.
* Quản lý được danh sách đợi tư vấn khám.
* Khám chữa bệnh cho bệnh nhân, tạo hồ sơ bệnh án, kết quả khám của mỗi cuộc hẹn,  
  tạo được các yêu cầu dịch vụ (siêu âm, xét nghiệm,…), kê toa thuốc cho bệnh nhân.
* Tạo các kết quả xét nghiệm, siêu âm, đưa ra chẩn đoán bệnh.
* Quản lý bệnh nhân thông qua danh sách bệnh nhân được hiển thị trên giao diện, các biểu đồ thống kê hằng ngày.
* Xem được tất cả các thông báo đặt lịch.
* Quản lý trang blog (đăng bài, thả cảm xúc, bình luận, …).
* Quản lý các thông tin cá nhân.
* **Admin:** Đăng nhập vào để thực hiện các chức năng trong hệ thống:
* Xem thống kê tổng số bác sĩ, bệnh nhân.
* Kiểm duyệt, kích hoạt tài khoản cho bác sĩ.
* Xem thống kê doanh thu, thuốc, các dịch vụ được sử dụng nhiều.
* Thanh toán các hóa đơn cho bệnh nhân.
* Checkin cho các bệnh nhân có lịch hẹn.
* Quản lý thuốc.

## 2.5 Mô tả yêu cầu phi chức năng:

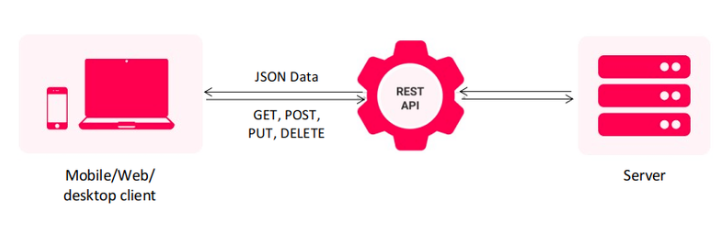
* Giao diện dễ nhìn, thân thiện và dễ sử dụng.
* Khả năng mở rộng.
* Khả năng bảo trì.
* Khả năng tương tác.
* Tính bảo mật.
* Tính đúng đắn.
* Độ tin cậy cao.

# CHƯƠNG 3: CƠ SỞ LÝ THUYẾT

## 3.1 Kiến trúc của hệ thống:

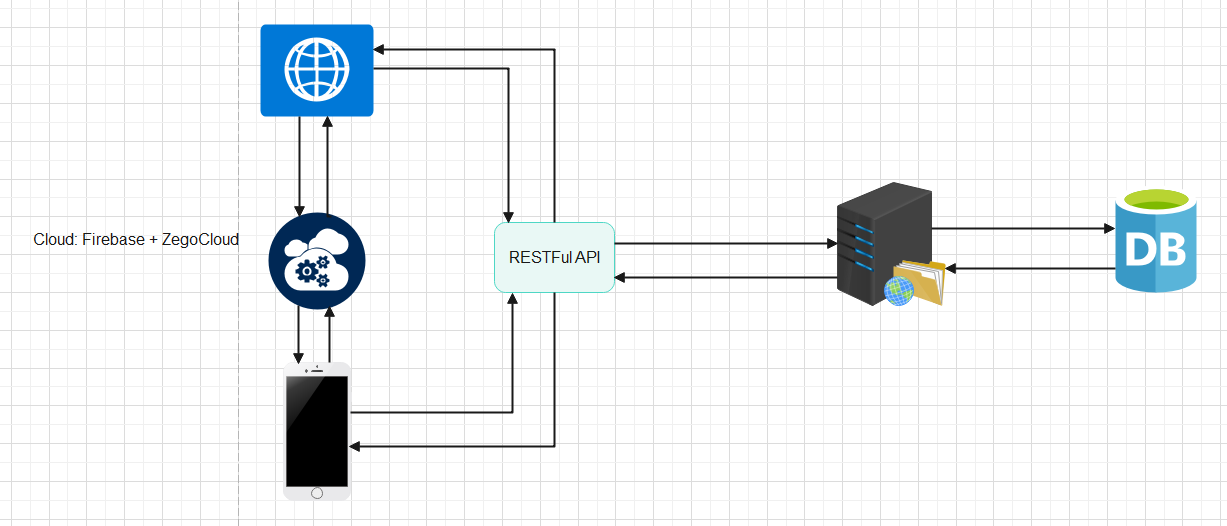
### 3.1.1 RESTful API:

* Mô hình RESTful API (Representational State Transfer): là một kiến trúc phần mềm được sử dụng để thiết kế các API (Application Programming Interface) cho các ứng dụng web và mobile. RESTful API sử dụng các phương thức HTTP (GET, POST, PUT, DELETE) để quản lý các tài nguyên (resource) và các trạng thái của chúng.
* Mô hình RESTful API bao gồm các thành phần chính sau:
* Resource: Là các đối tượng được quản lý và truy xuất bằng API. Mỗi tài nguyên sẽ có một URI (Uniform Resource Identifier) để xác định địa chỉ của nó trên mạng.
* HTTP Method: Sử dụng các phương thức HTTP như GET, POST, PUT, DELETE để thực hiện các hành động trên các tài nguyên.
* Representation: Các tài nguyên được trả về dưới dạng các định dạng như JSON, XML hoặc HTML.
* Stateless: Các yêu cầu đến server không chứa bất kỳ thông tin trạng thái nào về trạng thái của ứng dụng. Server sẽ lưu trữ tất cả thông tin trạng thái của ứng dụng trên client.
* Mô hình RESTful API giúp cho việc truy xuất và quản lý các tài nguyên trở nên đơn giản hơn. Nó được sử dụng rộng rãi trong các dự án phát triển ứng dụng web và mobile.



3.1. Mô hình RESTful API [1]

### 3.1.2 Kiến trúc ứng dụng:



3.1. Mô hình kiến trúc ứng dụng

## 3.2 Điện toán đám mây:

Điện toán đám mây (Cloud Computing): là một mô hình cho phép truy cập vào các tài nguyên máy tính (bao gồm phần cứng, phần mềm, dữ liệu) thông qua Internet, mà không cần phải cài đặt hoặc quản lý các tài nguyên này trên máy tính cá nhân hoặc các thiết bị cục bộ. Thay vào đó, người dùng có thể thuê hoặc sử dụng các tài nguyên máy tính được cung cấp bởi các nhà cung cấp dịch vụ đám mây để thực hiện các tác vụ, lưu trữ dữ liệu hoặc triển khai các ứng dụng trên Internet.

### 3.2.1. Progressive Web App:

Progressive Web App (PWA) là một dạng ứng dụng web hiện đại kết hợp các ưu điểm của trang web truyền thống với trải nghiệm mượt mà như ứng dụng gốc. Với khả năng hoạt động offline nhờ công nghệ Service Workers, PWA đảm bảo truy cập nhanh chóng và ổn định ngay cả trên mạng yếu. Người dùng có thể dễ dàng cài đặt PWA trực tiếp từ trình duyệt mà không cần qua các cửa hàng ứng dụng, đồng thời tận hưởng các tính năng như thông báo đẩy, giao diện toàn màn hình, và khả năng tương thích trên nhiều nền tảng. Nhờ hiệu suất cao, chi phí phát triển thấp và tính tiện lợi, PWA ngày càng được ưa chuộng trong việc mang lại trải nghiệm ứng dụng linh hoạt và tối ưu.

|  |
| --- |
| A diagram of a computer network  Description automatically generated with medium confidence |
| Hình 3.2.1. Các đặc điểm nổi bật của Progressive Web App. |

Bức hình minh họa các đặc điểm nổi bật của Progressive Web App (PWA), giúp làm rõ lý do tại sao loại ứng dụng này ngày càng phổ biến. Các đặc điểm chính bao gồm:

* Responsive: PWA tương thích với nhiều kích thước màn hình và thiết bị, bao gồm cả máy tính để bàn, máy tính bảng và điện thoại thông minh.
* Connectivity Independent: PWA có thể hoạt động ngay cả khi không có kết nối Internet nhờ sử dụng công nghệ Service Workers.
* App-like: PWA mang lại trải nghiệm giống như ứng dụng gốc (native app) với giao diện và cách sử dụng thân thiện, mượt mà.
* Fresh: Luôn đảm bảo cập nhật nội dung và các tính năng mới nhất nhờ cơ chế cập nhật tự động.
* Safe: Được phát triển trên giao thức HTTPS, PWA đảm bảo an toàn khi truyền tải dữ liệu và bảo vệ thông tin người dùng.
* Discoverable: Dễ dàng được tìm thấy trên các công cụ tìm kiếm, giúp người dùng tiếp cận dễ dàng hơn.
* Re-engageable: PWA hỗ trợ tính năng như thông báo đẩy (push notifications), giúp tương tác và kết nối lại với người dùng hiệu quả hơn.
* Installable: Người dùng có thể cài đặt PWA vào màn hình chính mà không cần qua các cửa hàng ứng dụng, mang lại sự tiện lợi và nhanh chóng.
* Linkable: PWA có thể được chia sẻ dễ dàng qua URL, không yêu cầu quy trình cài đặt phức tạp.

Tổng thể, các đặc điểm này giúp PWA trở thành lựa chọn linh hoạt, hiệu quả và phù hợp với nhu cầu hiện đại trong phát triển ứng dụng web.

### 3.2.2 MongoDB:

* MongoDB: là một hệ thống cơ sở dữ liệu phi quan hệ (NoSQL) được phát triển bởi MongoDB Inc. Nó được thiết kế để xử lý các ứng dụng với dữ liệu khối lượng lớn, dữ liệu phân tán và tải trọng cao. MongoDB sử dụng mô hình lưu trữ tài liệu (document-oriented) để lưu trữ dữ liệu.
* MongoDB cũng được tích hợp tốt với các ngôn ngữ lập trình khác nhau, bao gồm các ngôn ngữ phổ biến như JavaScript, Python, Java và Ruby. Nó cũng có khả năng tích hợp với các nền tảng phổ biến như Node.js và các hệ thống điện toán đám mây như AWS và Azure.

A screenshot of a video game

Description automatically generated

**3.3.1 Sơ dồ kiến trúc Mongodb**

## 3.3 Ngôn ngữ lập trình, Framework và công nghệ hỗ trợ:

* Node.js được phát triển và duy trì bởi OpenJS Foundation, với mục tiêu tạo ra một nền tảng phát triển được xây dựng trên cơ sở mã nguồn mở, linh hoạt và hiệu quả cao cho các ứng dụng web hiện đại.
* Một số ưu điểm của Node.js bao gồm khả năng xử lý đồng thời, hiệu suất cao và khả năng mở rộng dễ dàng. Node.js cũng được hỗ trợ bởi một số thư viện và framework như Express, Nest.js và Socket.IO, giúp cho việc phát triển ứng dụng web trở nên dễ dàng hơn.

A diagram of a function

Description automatically generated

### 3.3.3 Framework Nestjs:

**Nest hay NestJS** là một framework để xây dựng các ứng dụng server-side mạnh mẽ dựa trên nền tảng Nodejs. NestJS được xây dựng và hỗ trợ ngôn ngữ TypeScript (vẫn cho phép lập trình viên code bằng JavaScript thuần), và kết hợp các tính năng của OOP (Object Oriented Programming), FP (Functional Programming), and FRP (Functional Reactive Programming). Ưu điểm của React Native bao gồm:

* Ngôn ngữ mạnh về kiểu dữ liệu: NestJS được xây dựng trên nền TypeScript, một phiên bản hỗ trợ kiểu dữ liệu của JavaScript. Sử dụng Typescript khi viết code giúp code sạch hơn và dễ xử lý khi có lỗi.

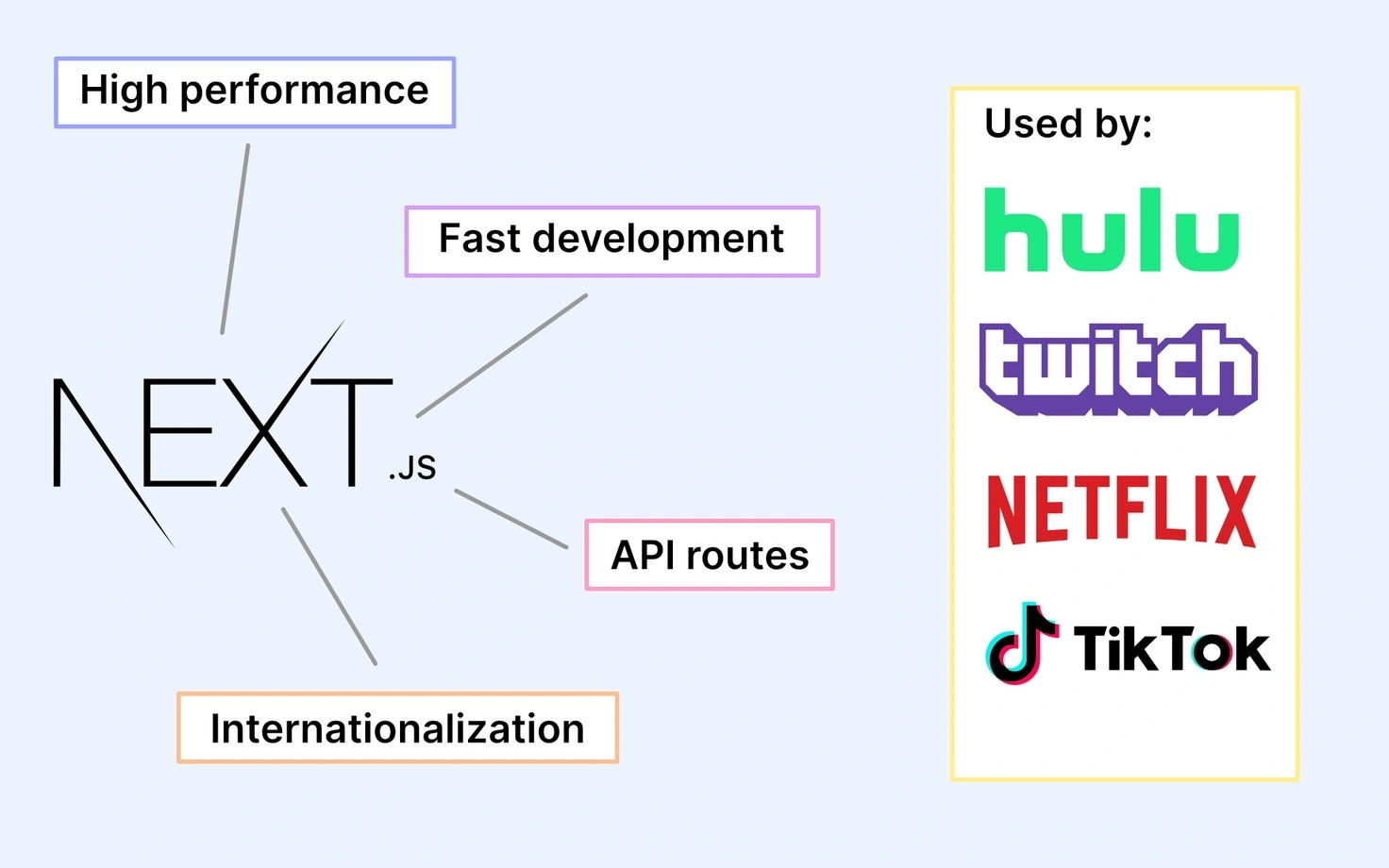
|  |
| --- |
|  |
| Hình 3.3.3. Các tính năng của nestjs [11]. |

* Kiến trúc modular: NestJS được xây dựng trên kiến trúc modular giúp chúng ta dễ tổ chức và phát triển mã nguồn trên quy mô lớn hơn. Kiến trúc này bao gồm các khối tách biệt nhau như controller, provider, và module, giúp cho việc quản lý dễ dàng hơn [11].
* Dependency Injection: NestJS sử dụng Dependency Injection (DI) để quản lý luồng của các phụ thuộc giữ các module và component. Nó cho phép ứng dụng dễ kiểm thử và bảo trì hơn, do sự độc lập giữa các component với nhau [11].
* Built-In Validation: NestJS được hỗ trợ sẵn tính năng validate dữ liệu đầu vào - thứ đặc biệt quan trong khi xây dựng các API. Framework này giúp định nghĩa và thực thi các validation rule một cách dễ dàng hơn, giảm đi các sai sót và lỗi không đáng có.

### 3.3.4 Framework NextJs:

NextJS là framework mã nguồn mở được xây dựng trên nền tảng của React, cho phép chúng ta xây dựng các trang web tĩnh có tốc độ siêu nhanh và thân thiện. Ưu điểm của ReactJS bao gồm:

* Component-based: Nextjs sử dụng cấu trúc component, giúp tách biệt các thành phần trong ứng dụng và dễ dàng tái sử dụng code.
* Virtual DOM: Nextjssử dụng Virtual DOM, giúp cải thiện hiệu suất bằng cách chỉ cập nhật các phần thay đổi trong DOM thay vì toàn bộ.
* Codebase chia sẻ: Nextjscũng cho phép chia sẻ codebase giữa các ứng dụng web và mobile, giảm thời gian và chi phí phát triển.
* Cộng đồng lớn: Nextjs có cộng đồng lớn, hỗ trợ và cập nhật liên tục.
* Tuy nhiên, Nextjs cũng có một số nhược điểm:
* Khóa học: vì Nextjs sử dụng JSX để tạo giao diện, việc học Nextjs đòi hỏi sự hiểu biết về cả HTML và Javascript.
* Khó khăn trong việc quản lý state: với các ứng dụng lớn, việc quản lý state trong Nextjs có thể trở nên khó khăn và phức tạp.



3.3. Nextjs là gì? [9]

### 3.3.7 JWT:

Diagram

Description automatically generatedJWT: là viết tắt của JSON Web Token. Đây là một tiêu chuẩn mã hóa dữ liệu dựa trên JSON được sử dụng để xác thực và truyền dữ liệu an toàn giữa các bên. JWT thường được sử dụng trong các ứng dụng web và di động, trong đó các người dùng cần đăng nhập và được phân quyền truy cập vào các tài nguyên.

3.3. Các thành phần của JWT [12]

### 3.3.8 ChartJS:

* Chart.js**:** là một thư viện JavaScript mã nguồn mở được sử dụng để tạo các biểu đồ động trên trình duyệt. Thư viện này hỗ trợ nhiều loại biểu đồ như biểu đồ đường, biểu đồ cột, biểu đồ tròn, biểu đồ vùng, biểu đồ bánh xe và biểu đồ scatter. Chart.js dễ sử dụng và tùy biến, cung cấp nhiều tùy chọn để điều chỉnh các thuộc tính của biểu đồ, bao gồm cả màu sắc, kiểu dữ liệu và kiểu hiển thị.
* Chart.js được phát triển và duy trì bởi nhóm phát triển tại Github, và là một trong những thư viện biểu đồ phổ biến nhất trên thị trường. Chart.js có thể được sử dụng trong nhiều loại ứng dụng web, bao gồm các báo cáo tài chính, dashboard quản lý dữ liệu và các ứng dụng phân tích dữ liệu.

### 3.3.9 RAG chatbot:

* Retrieval-Augmented Generation (RAG) là một kỹ thuật giúp nâng cao khả năng của mô hình sinh (language model generation) kết hợp với tri thức bên ngoài (external knowledge)
* Phương pháp này thực hiện bằng cách truy xuất thông tin liên quan từ kho tài liệu (tri thức) và sử dụng chúng cho quá trình sinh câu trả lời dựa trên LLMs.

A diagram of a computer process

Description automatically generated

3.3. RAG là gì? [14]

# CHƯƠNG 4: PHÂN TÍCH

## 4.1 Phân tích yêu cầu bằng UML (Unified Modeling Language):

### 4.1.1 Sơ đồ usecase tổng quát:

A diagram of a system

Description automatically generated

4.1. Sơ đồ usecase tổng quát4.1.2 Danh sách các tác nhân và mô tả:

|  |  |  |
| --- | --- | --- |
| Tác nhân | Mô tả tác nhân | Ghi chú |
| Bệnh nhân | Sau khi đăng ký thành công tài khoản, bệnh nhân có thể thực hiện các chức năng của hệ thống:  - Đăng ký lịch tư vấn khám.  - Được tư vấn tự động thông qua chatbot RAG.  - Quản lý được các hồ sơ bệnh án, toa thuốc đã được kê trước đó.  - Quản lý các thông tin cá nhân. |  |
| Bác sĩ | Sau khi đăng ký thành công tài khoản (đợi xác thực từ bên Admin), sau khi được xác thực tài khoản thành công thì bác sĩ có thể đăng nhập được vào hệ thống và có thể thực hiện được các chức năng bên trong hệ thống:  - Xem được các chỉ số thống kê (Tất cả các thông tin tổng quan ở phần Bảng điều khiển).  - Quản lý được lịch hẹn tư vấn khám.  - Quản lý bệnh nhân thông qua Danh sách bệnh nhân được render trên bảng, các biểu đồ thống kê hằng ngày.  - Xem được tất cả các thông báo đặt lịch.  - Chẩn đoán tình trạng của bệnh nhân.  - Tạo hồ sơ bệnh án của bệnh nhân.  - Tạo các dịch vụ thăm khám cho bệnh nhân.  - Tạo các kết quả khám dịch vụ (xét nghiệm, siêu âm,…).  - Tạo hóa đơn thuốc cho bệnh nhân.  - Quản lý trang Blog.  - Quản lý các thông tin cá nhân. |  |
| Admin | Sau khi đăng nhập vào để thực hiện các chức năng trong hệ thống:  - Xem thống kê tổng số bác sĩ, bệnh nhân.  - Kiểm duyệt, kích hoạt tài khoản cho bác sĩ.  - Quản lí, mua bán thuốc chữa bệnh.  - Quản lí các dịch vụ thăm khám.  - Checkin khám chữa bệnh cho bệnh nhân.  - Thanh toán các hóa đơn cho bệnh nhân. |  |

Bảng 4.1.2 Danh sách các tác nhân và mô tả

### 4.1.3 Danh sách các usecase và mô tả

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Tên Use case | Mô tả ngắn gọn Use case | Chức năng |
| UC01 | Đăng ký | Bác sĩ/ bệnh nhân truy cập vào Web/ App tiến hành tạo tài khoản của mình. | Đăng ký tài khoản trên hệ thống của NS Healthcare. |
| UC02 | Đăng nhập | Bác sĩ/ bệnh nhân/ admin vào Web/ App tiến hành đăng nhập tài khoản của mình đã tạo trước đó. | Đăng nhập vào hệ thống của NS Healthcare. |
| UC03 | Quản lý thống kê | Admin có thể thống kê về doanh thu, số lượng bệnh nhân hoàn thành khám trong ngày, các loại thuốc bán chạy và các dịch vụ được sử dụng nhiều nhất. | Thống kê |
| UC04 | Quản lý nhân viên | Admin có thể quản lý nhân viên trong hệ thống bao gồm, y tá, bác sĩ. | Quản lý nhân sự trong hệ thống NS Healthcare |
| UC05 | Quản lý bệnh nhân | Quản lý danh sách bệnh nhân, thông tin bệnh nhân, tạo bệnh nhân mới, chỉnh sửa thông tin bệnh nhân trong hệ thống phòng khám. | Quản lý thông tin bệnh nhân |
| UC06 | Quản lý thuốc | Quản lý các loại thuốc, nhập thuốc, các lô thuốc. Xử lý các lô thuốc hết hạn sử dụng. | Quản lý thuốc thuốc, lô thuốc. |
| UC07 | Quản lý thanh toán | Thanh toán các hóa đơn dịch vụ, hóa đơn thuốc. Xuất hóa đơn. Xử lý thanh toán chuyển khoản. | Thanh toán hóa đơn. |
| UC08 | Quản lý dịch vụ | Tạo, sửa thông tin, đơn giá các dịch vụ thăm khám chữa bệnh. | Tạo, chỉnh sửa các dịch vụ. |
| UC09 | Quản lý lịch hẹn | Xem danh sách các lịch hẹn theo ngày, checkin cho bệnh nhân vào phòng chờ khám. | Xem, thống kê các lịch hẹn theo ngày. |
| UC10 | Tư vấn | Tư vấn về quy trình khám chữa bệnh, cách phòng tránh bệnh. Giá thành của các dịch vụ trong phòng khám bằng chatbot | Tư vấn khám chữa bệnh. |
| UC11 | Quản lý khám chữa bệnh | Xem danh sách các bệnh nhân chờ khám, tạo các yêu cầu dịch vụ cho bệnh nhân, tạo kết quả khám, kê toa thuốc. | Khám bệnh |
| UC12 | Quản lý blog | Tạo, xem chi tiết các blog về sức khỏe, các bệnh thường gặp, cách phòng tránh. Các phương pháp cải thiện sức khỏe. | Tạo, chỉnh sửa các blog |
| UC13 | Quản lý thông tin cá nhân | Mỗi người dùng đều có thể xem và chỉnh sửa thông tin cá nhân của mình. | Xem, chỉnh sửa thông tin cá nhân. |
| UC14 | Xem các bài viết | Xem chi tiết các blog được bác sĩ, admin viết. | Đọc blog |
| UC15 | Xem hồ sơ bệnh án | Bệnh nhân có thể xem lại các hồ sơ bệnh án, lịch sử khám bệnh, toa thuốc trước đó tại phòng khám NS Healthcare của chính mình. | Xem lại hồ sơ bệnh án. |
| UC16 | Xem lịch hẹn | Bệnh nhân có thể xem chi tiết danh sách các lịch hẹn khám của mình bao gồm bác sĩ khám, ngày giờ. | Xem chi tiết lịch hẹn |
| UC17 | Đặt lịch khám | Bệnh nhân chọn chuyên khoa muốn khám, bác sĩ khám, ngày giờ chi tiết để đặt lịch khám. | Đặt lịch khám chữa bệnh. |
| UC18 | Thanh toán dịch vụ | Admin chọn hóa đơn dịch vụ muốn thanh toán, tiến hành thanh toán và xuất hóa đơn. | Thanh toán hóa đơn dịch vụ. |
| UC19 | Thanh toán đơn thuốc | Admin chọn hóa đơn thuốc cần thanh toán, xuất hóa đơn cho bệnh nhân. | Thanh toán hóa đơn thuốc. |
| UC20 | Checkin lịch hẹn | Khi bệnh nhân đến phòng khám sẽ được checkin và vào phòng chờ khám. | Checkin lịch hẹn cho bệnh nhân. |
| UC21 | Tạo các dịch vụ yêu cầu | Từ những chẩn đoán ban đầu bác sĩ tạo các yêu cầu dịch vụ chi tiết như siêu âm, xét nghiệm máu,… để có được kết quả cụ thể hơn | Tạo yêu cầu dịch vụ cho bệnh nhân. |
| UC22 | Tạo lịch tái khám | Tạo lịch tái khám cho bệnh nhân vào một ngày giờ cụ thể sau khi đã khám xong. | Tạo lịch tái khám cụ thể cho bệnh nhân. |
| UC23 | Tạo hồ sơ bệnh án | Tạo kết quả khám sau khi khám, đưa ra chẩn đoán, các lưu ý cho bệnh nhân. | Tạo kết quả khám |
| UC24 | Xem các hồ sơ bệnh án | Bác sĩ sẽ có quyền xem các hồ sơ bệnh án trước kia của bệnh nhân hiện đang khám để hiểu rõ hơn về tình trạng bệnh của bệnh nhân. | Xem chi tiết các hồ sơ bệnh án của bệnh nhân. |
| UC25 | Kê toa thuốc | Trong quá trình tạo hồ sơ bệnh án bác sĩ có thể kê toa thuốc cho bệnh nhân bao gồm số ngày sử dụng, liều lượng sử dụng và lưu ý khi sử dụng | Kê toa thuốc cho bệnh nhân |
| UC26 | Quản lý thống kê dành cho bác sĩ | Bác sĩ có thể thống kê các lịch sử khám bệnh đã được hoàn thành của mình theo khoảng thời gian. | Thống kê số lượng cuộc khám đã hoàn thành. |
| UC27 | Chỉnh sửa thông tin cá nhân | Mỗi người dùng có thể sửa thông tin cá nhân của mình trong hệ thống. | Cập nhật thông tin cá nhân. |
| UC28 | Xem thông tin cá nhân | Mỗi người dùng khi đăng nhập vào hệ thống có thể xem chi tiết thông tin cá nhân của mình như họ tên, giới tính, ngày sinh,…. | Xem chi tiết thông tin cá nhân. |
| UC29 | Đăng xuất | Đăng xuất khỏi hệ thống, quay trở lại trang đăng nhập. | Đăng xuất khỏi hệ thống. |
| UC30 | Xem toa thuốc | Bệnh nhân xem lại toa thuốc đã được bác sĩ kê ở những lần khám trước đó. | Xem chi tiết toa thuốc. |
| UC31 | Thay đổi lịch khám | Bệnh nhân có thể thay đổi lại ngày giờ cuộc hẹn của mình. | Thay đổi thời gian cuộc hẹn. |
| UC32 | Hủy lịch khám | Bệnh nhân có thể hủy lịch khám đã đăng ký trước đó. | Hủy lịch hẹn trước đó. |

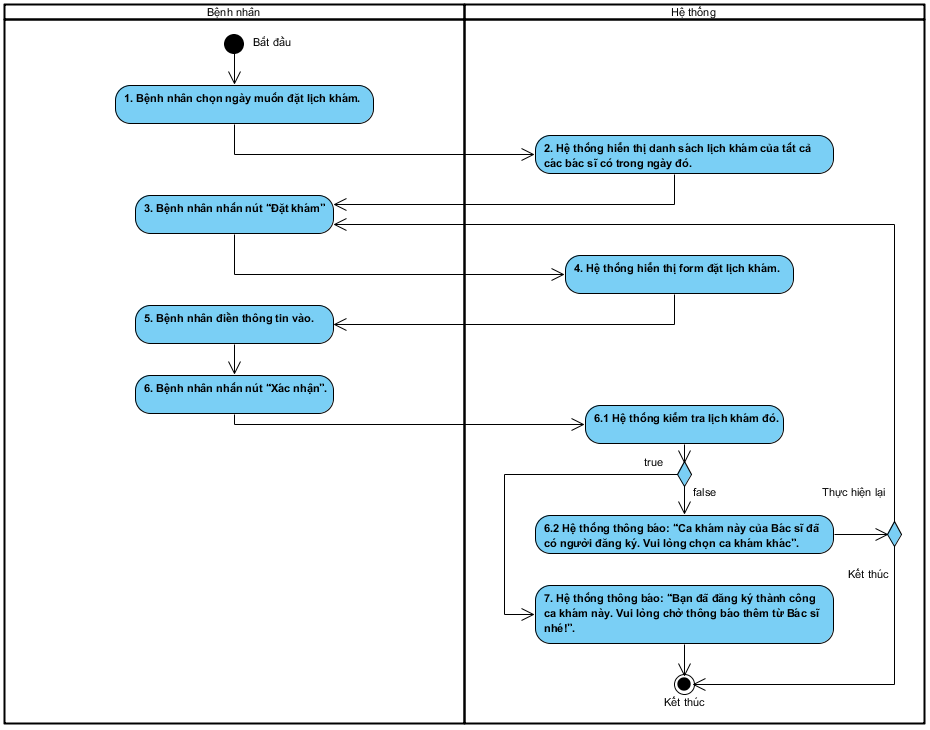
Bảng 4.1.3 Danh sách các usecase và mô tả

#### 4.1.4.1 Usecase “Đặt lịch khám bởi bệnh nhân”:

|  |  |
| --- | --- |
| **Tên use case**: Đặt lịch khám bởi bệnh nhân | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bệnh nhân có thể đăng ký lịch khám với bác sĩ mà mình chọn. | |
| **Actor chính**: Bệnh nhân | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện (Post-condition):** Đăng ký thành công lịch khám, hệ thống sẽ tự động cập nhật lại danh sách lịch khám của bệnh nhân. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bệnh nhân chọn ngày muốn đặt lịch khám. |  |
|  | 2. Hệ thống hiển thị danh sách lịch khám của tất cả các bác sĩ có trong ngày đó. |
| 3. Bệnh nhân nhấn nút “Đặt khám” |  |
|  | 4. Hệ thống hiển thị form đặt lịch khám. |
| 5. Bệnh nhân điền thông tin vào. |  |
| 6. Bệnh nhân nhấn nút “Xác nhận”. |  |
|  | 7. Hệ thống thông báo: “Bạn đã đăng ký thành công ca khám này. Vui lòng chờ thông báo thêm từ Bác sĩ nhé!”. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1 Hệ thống kiểm tra lịch khám đó. |
|  | 6.2 Hệ thống thông báo: “Ca khám này của Bác sĩ đã có người đăng ký. Vui lòng chọn ca khám khác”. |

Bảng 4.1.4.2 Đặc tả usecase Đặt lịch khám bởi bệnh nhân

##### 4.1.4.1.1. Sơ đồ Activity “Đặt lịch khám bởi bệnh nhân”:



4.1.4.2. 1 Sơ đồ activity Đặt lịch khám bởi bệnh nhân

##### 4.1.4.2.2 Sequence “Đặt lịch khám bởi bệnh nhân”:

A diagram with text and numbers

Description automatically generated with medium confidence

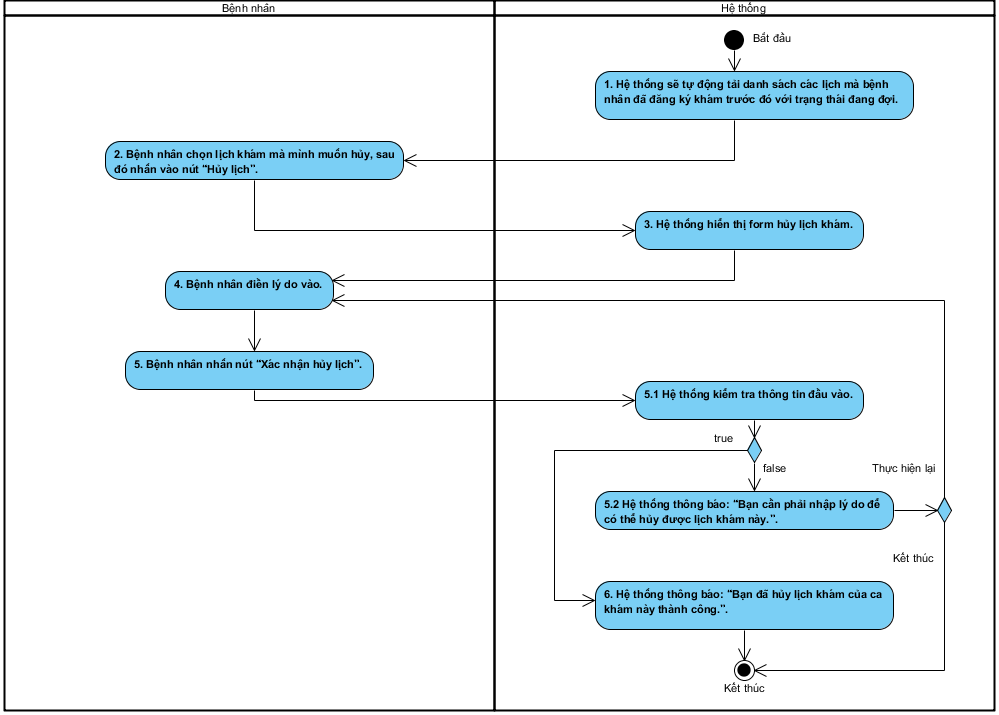
4.1.4.2. 2 Sơ đồ sequence Đặt lịch khám bởi bệnh nhân

#### 4.1.4.3 Usecase “Hủy lịch khám bởi bệnh nhân”:

|  |  |
| --- | --- |
| **Tên use case**: Hủy lịch khám bởi bệnh nhân | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bệnh nhân có thể hủy lịch khám (nếu trước đó đã đăng ký lịch khám). | |
| **Actor chính**: Bệnh nhân | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện (Post-condition):** Hủy thành công lịch khám đã đăng ký trước đó, hệ thống sẽ tự động cập nhật lại danh sách lịch khám của bệnh nhân. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
|  | 1. Hệ thống sẽ tự động tải danh sách các lịch mà bệnh nhân đã đăng ký khám trước đó với trạng thái đang đợi. |
| 2. Bệnh nhân chọn lịch khám mà mình muốn hủy, sau đó nhấn vào nút “Hủy lịch”. |  |
|  | 3. Hệ thống hiển thị form hủy lịch khám. |
| 4. Bệnh nhân điền lý do vào. |  |
| 5. Bệnh nhân nhấn nút “Xác nhận hủy lịch”. |  |
|  | 6. Hệ thống thông báo: “Bạn đã hủy lịch khám của ca khám này thành công.”. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1 Hệ thống kiểm tra thông tin đầu vào. |
|  | 5.2 Hệ thống thông báo: “Bạn cần phải nhập lý do để có thể hủy được lịch khám này.”. |

Bảng 4.1.4.3 Đặc tả usecase Hủy lịch khám bởi bệnh nhân

##### 4.1.4.3.1 Activity “Hủy lịch khám bởi bệnh nhân”:



4.1.4.3. 1 Sơ đồ activity Hủy lịch khám bởi bệnh nhân

##### 4.1.4.3.2 Sequence “Hủy lịch khám bởi bệnh nhân”:

### 4.1.4 Đặc tả usecase, sơ đồ activity, sơ đồ sequence các chức năng chính trong hệ thống.

#### 4.1.4.1 Usecase “Checkin lịch hẹn”:

|  |  |
| --- | --- |
| **Tên use case**: Checkin lịch hẹn | |
| **Mô tả sơ lược**:Admin là người checkin cho bệnh nhân khi bệnh nhân tới khám bệnh, sau khi checkin bệnh nhân sẽ được vào khu vực chờ khám. | |
| **Actor chính**: Admin | |
| **Actor phụ**: không | |
| **Tiền điều kiện (Pre-condition):** không | |
| **Hậu điều kiện (Post-condition):** Trạng thái của cuộc hẹn được chuyển thành “waiting”. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
|  | 1. Hệ thống load lên các lịch hẹn theo ngày đặt và có trạng thái là “booked”. |
| 2.Admin chọn nút checkin |  |
|  | 3. Hệ thống cập nhật trạng thái của cuộc hẹn lại thành “waiting”. Hiển thị thông báo checkin thành công. |
| **Luồng sự kiện thay thế (alternate flow):** | |

Bảng 4.1.4.1. Đặc tả usecase Đăng ký tài khoản cho bác sĩ

##### 4.1.4.1.1. Sơ đồ Activity “Checkin lịch hẹn”:

|  |
| --- |
| A screenshot of a computer  Description automatically generated |
| Hình 4.1.4.1.1. Activity checkin lịch hẹn |

##### 4.1.4.1.2. Sơ đồ Sequence “Checkin lịch hẹn”:

|  |
| --- |
| A diagram of a diagram  Description automatically generated |
| Hình 4.1.4.1.2. Sơ đồ sequence “Checkin lịch hẹn”. |

#### 4.1.4.2 Usecase “Tạo các dịch vụ yêu cầu”:

|  |  |
| --- | --- |
| **Tên use case**: Tạo các dịch vụ yêu cầu. | |
| **Mô tả sơ lược**: Từ những chẩn đoán ban đầu bác sĩ tạo các yêu cầu dịch vụ chi tiết như siêu âm, xét nghiệm máu,… để có được kết quả cụ thể hơn. | |
| **Actor chính**: Bác sĩ | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Bệnh nhân hiện đang được khám. | |
| **Hậu điều kiện (Post-condition):** Tạo thành công yêu cầu dịch vụ với trạng thái “awaiting payment” chờ người dùng thanh toán. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bác sĩ chọn nút tạo yêu cầu dịch vụ. |  |
|  | 2. Hệ thống hiển thị modal tạo yêu cầu |
| 3. Chọn dịch vụ, nhập thông tin chẩn đoán ban đầu, ghi chú. |  |
| 4. Nhấn nút yêu cầu dịch vụ |  |
|  | 5. Kiểm tra thông tin đầu vào |
|  | 6. Tạo yêu cầu dịch vụ |
|  | 7. Thông báo tạo yêu cầu dịch vụ thành công |
|  | 8. Đóng modal, hiển thị yêu cầu dịch vụ vừa tạo tại màn hình thông tin khám. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1. Yêu cầu nhập đầy đủ thông tin cần thiết. |

Bảng 4.1.4.2 Đặc tả usecase tạo các dịch vụ yêu cầu.

##### 4.1.4.2.1. Sơ đồ Activity “Tạo các dịch vụ yêu cầu”:

|  |
| --- |
| A diagram of a diagram  Description automatically generated |
| Hình 4.1.4.2.1. Sơ đồ Activity “Tạo các dịch vụ yêu cầu”. |

##### 4.1.4.2.2. Sơ đồ Sequence “Tạo các dịch vụ yêu cầu”:

|  |
| --- |
| A diagram of a medical procedure  Description automatically generated with medium confidence |
| Hình 4.1.4.2.2. Sơ đồ Sequence “Tạo các dịch vụ yêu cầu”. |

#### 4.1.4.3. Usecase “Tạo hồ sơ bệnh án”:

|  |  |
| --- | --- |
| **Tên use case**: Tạo hồ sơ bệnh án. | |
| **Mô tả sơ lược**: Tạo kết quả khám sau khi khám và xem qua các kết quả từ những dịch vụ khác, đưa ra chẩn đoán, các lưu ý cho bệnh nhân. | |
| **Actor chính**: Bác sĩ | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Bệnh nhân hiện đang được khám. | |
| **Hậu điều kiện (Post-condition):** Tạo thành công kết quả khám cho cuộc hẹn và lưu xuống database, bác sĩ có thể xem lại hồ sơ bệnh án vừa tạo và hoàn thành quy trình khám | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bác sĩ chọn nút tạo kết quả khám |  |
|  | 2. Hệ thống chuyển hướng đến trang tạo hồ sơ bệnh án |
| 3. Bác sĩ nhập đầy đủ các form bắt buộc. |  |
| 4. Nhấn nút tạo hồ sơ bệnh án |  |
|  | 5. Hệ thống kiểm tra thông tin đầu vào |
|  | 6. Tiến hành lưu hồ sơ bệnh án. |
|  | 7. Thông báo lưu hồ sơ bệnh án thành công |
|  | 8. Điểu hướng về lại màn hình chi tiết cuộc hẹn và hiển thị hồ sơ bệnh án vừa lưu. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1. Yêu cầu nhập đầy đủ thông tin cần thiết |

Bảng 4.1.4.2 Đặc tả usecase Đặt lịch khám bởi bệnh nhân4.1.4.2.1 Activity “Đặt lịch khám bởi bệnh nhân”:

##### 4.1.4.3.1. Sơ đồ Activity “Tạo hồ sơ bệnh án”:

|  |
| --- |
| A diagram of a diagram  Description automatically generated |
| Hình 4.1.4.3.1. Sơ đồ activity “tạo hồ sơ bệnh án”. |

##### 4.1.4.3.2. Sơ đồ Sequence “Tạo hồ sơ bệnh án”:

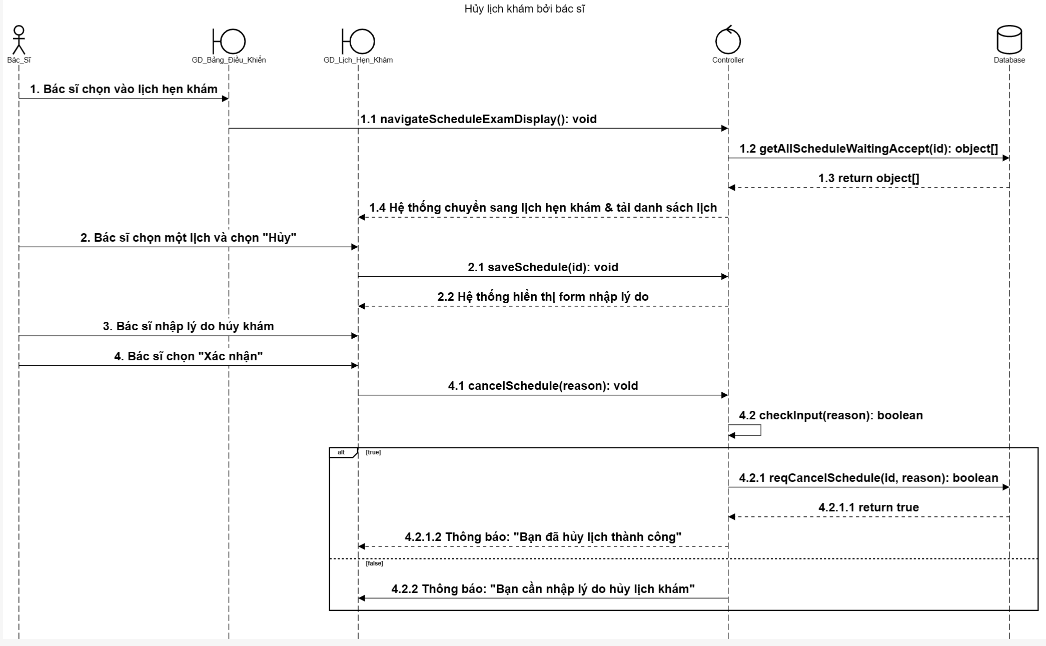
|  |
| --- |
| A diagram of a diagram  Description automatically generated with medium confidence |
| Hình 4.1.4.3.2. Sơ đồ sequence “tạo hồ sơ bệnh án”. |

#### 4.1.4.3 Usecase “Hủy lịch khám bởi bệnh nhân”:

|  |  |
| --- | --- |
| **Tên use case**: Hủy lịch khám bởi bệnh nhân | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bệnh nhân có thể hủy lịch khám (nếu trước đó đã đăng ký lịch khám). | |
| **Actor chính**: Bệnh nhân | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện (Post-condition):** Hủy thành công lịch khám đã đăng ký trước đó, hệ thống sẽ tự động cập nhật lại danh sách lịch khám của bệnh nhân. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
|  | 1. Hệ thống sẽ tự động tải danh sách các lịch mà bệnh nhân đã đăng ký khám trước đó với trạng thái đang đợi. |
| 2. Bệnh nhân chọn lịch khám mà mình muốn hủy, sau đó nhấn vào nút “Hủy lịch”. |  |
|  | 3. Hệ thống hiển thị form hủy lịch khám. |
| 4. Bệnh nhân điền lý do vào. |  |
| 5. Bệnh nhân nhấn nút “Xác nhận hủy lịch”. |  |
|  | 6. Hệ thống thông báo: “Bạn đã hủy lịch khám của ca khám này thành công.”. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1 Hệ thống kiểm tra thông tin đầu vào. |
|  | 5.2 Hệ thống thông báo: “Bạn cần phải nhập lý do để có thể hủy được lịch khám này.”. |

Bảng 4.1.4.3 Đặc tả usecase Hủy lịch khám bởi bệnh nhân

##### 4.1.4.3.2 Sequence “Hủy lịch khám bởi bệnh nhân”:



4.1.4.3. Sơ đồ sequence Hủy lịch khám bởi bệnh nhân

#### 4.1.4.5 Usecase “Xác nhận lịch khám bởi bác sĩ”:

|  |  |
| --- | --- |
| **Tên use case**: Xác nhận lịch khám bởi bác sĩ | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bác sĩ có thể xác nhận lịch khám mà bệnh nhân đã đăng ký khám. | |
| **Actor chính**: Bác sĩ | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống, vào phần “Lịch hẹn khám” | |
| **Hậu điều kiện (Post-condition):** Lịch khám của bệnh nhân được xác nhận thành công, sau đó hệ thống sẽ tự động cập nhật lại danh sách cuộc hẹn khám. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
|  | 1. Hệ thống sẽ tự động tải danh sách tất cả các lịch mà bệnh nhân đã đăng ký và được bác sĩ xác nhận khám. |
| 2. Bác sĩ nhấn vào nút “Xác nhận”. |  |
|  | 3. Hệ thống thông báo: “Bạn đã xác nhận lịch khám cho bệnh nhân”. |
| - **Luồng sự kiện thay thế (alternate flow):** | |

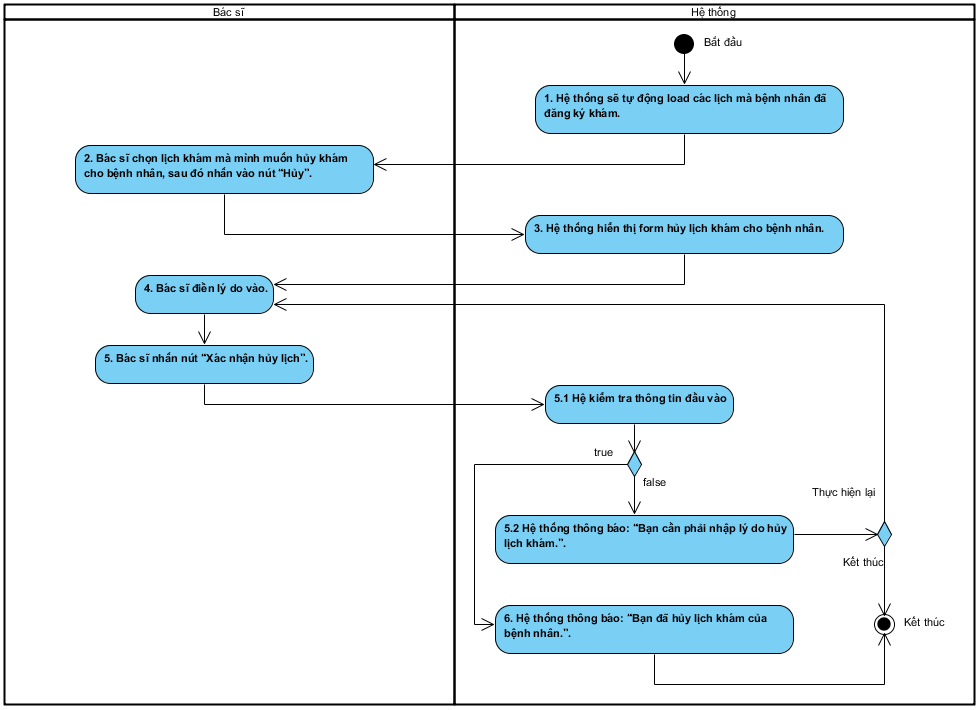
Bảng 4.1.4.5 Đặc tả usecase Xác nhận lịch khám bởi bác sĩ

#### 4.1.4.6 Usecase “Hủy lịch khám bởi bác sĩ”:

|  |  |
| --- | --- |
| **Tên use case**: Hủy lịch khám bởi bác sĩ | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bác sĩ có thể hủy lịch khám mà bệnh nhân đã đăng ký khám. | |
| **Actor chính**: Bác sĩ | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống, vào phần “Lịch hẹn khám” | |
| **Hậu điều kiện (Post-condition):** Hủy thành công lịch khám của bệnh nhân đã đăng ký khám, hệ thống sẽ tự động cập nhật lại danh sách lịch khám của bác sĩ. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
|  | 1. Hệ thống sẽ tự động tải danh sách lịch mà bệnh nhân đã đăng ký khám. |
| 2. Bác sĩ chọn lịch khám mà mình muốn hủy khám cho bệnh nhân, sau đó nhấn vào nút “Hủy”. |  |
|  | 3. Hệ thống hiển thị form hủy lịch khám cho bệnh nhân. |
| 4. Bác sĩ điền lý do vào. |  |
| 5. Bác sĩ nhấn nút “Xác nhận hủy lịch”. |  |
|  | 6. Hệ thống thông báo: “Bạn đã hủy lịch khám của bệnh nhân.”. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1 Hệ thống kiểm tra thông tin đầu vào. |
|  | 5.2 Hệ thống thông báo: “Bạn cần phải nhập lý do hủy lịch khám.”. |

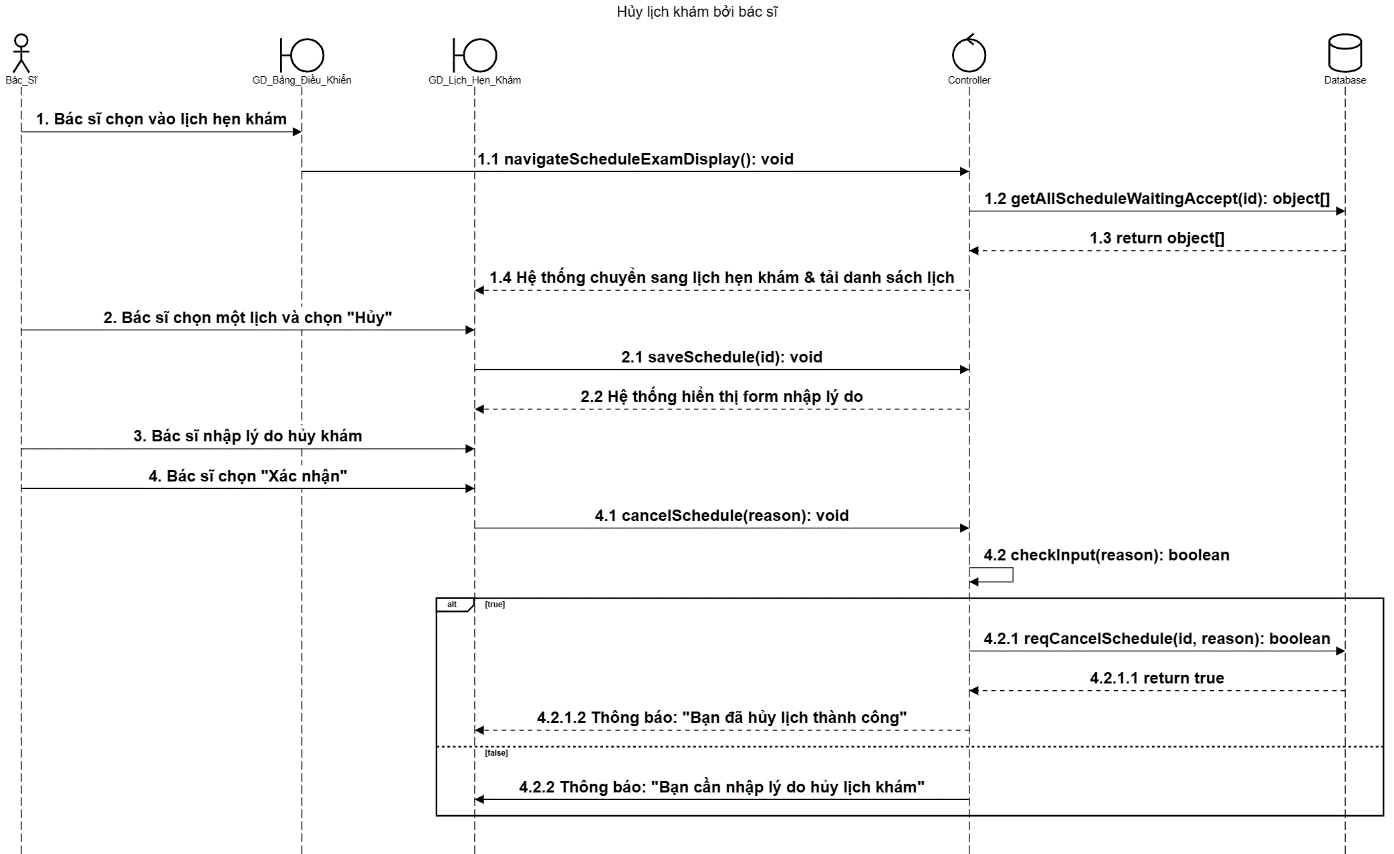
Bảng 4.1.4.6 Đặc tả usecase Hủy lịch khám bởi bác sĩ

##### 4.1.4.6.1 Activity “Hủy lịch khám bởi bác sĩ”:



4.1.4.6. Sơ đồ activity Hủy lịch khám bởi bác sĩ

##### 4.1.4.6.2 Sequence “Hủy lịch khám bởi bác sĩ”:



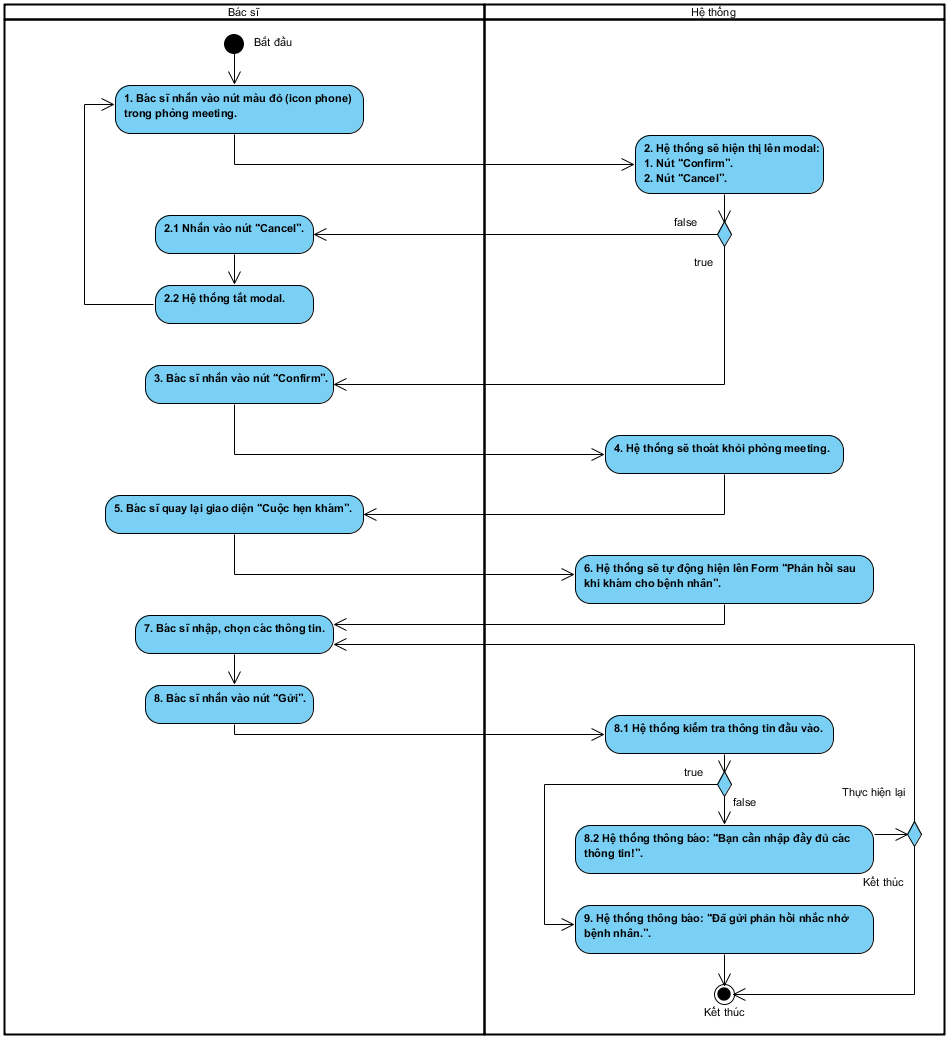
4.1.4.6. Đặc tả usecase Hủy lịch khám bởi bác sĩ

#### 4.1.4.7 Usecase “Phản hồi bệnh nhân sau khi khám”:

|  |  |
| --- | --- |
| **Tên use case**: Phản hồi bệnh nhân sau khi khám | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống và sau khi khám cho bệnh nhân xong, bác sĩ có thể nhắc nhở thêm cho bệnh nhân chú ý sức khỏe. | |
| **Actor chính**: Bác sĩ | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống và sau khi khám cho bệnh nhân xong. | |
| **Hậu điều kiện (Post-condition):** Nhắc nhở bệnh nhân thành công, hệ thống sẽ tự động cập nhật lại danh sách kết quả khám của bệnh nhân. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bác sĩ nhấn vào nút màu đỏ (icon phone) trong phòng khám. |  |
|  | 2. Hệ thống sẽ hiển thị lên modal:  1. Nút “Confirm”.  2. Nút “Cancel”. |
| 3. Bác sĩ nhấn vào nút “Confirm”. |  |
|  | 4. Hệ thống sẽ thoát khỏi phòng meeting |
| 5. Bác sĩ quay lại giao diện “Cuộc hẹn khám”. |  |
|  | 6. Hệ thống sẽ tự động hiện lên Form “Phản hồi sau khi khám cho bệnh nhân”. |
| 7. Bác sĩ nhập, chọn các thông tin. |  |
| 8. Bác sĩ nhấn vào nút “Gửi”. |  |
|  | 9. Hệ thống thông báo: “Đã gửi phản hồi cho bệnh nhân.”. |
| **Luồng sự kiện thay thế (alternate flow):** | |
| 2.1 Nhấn vào nút “Cancel”. |  |
|  | 2.2 Hệ thống tắt modal. |
|  | 8.1 Hệ thống kiểm tra thông tin đầu vào. |
|  | 8.2 Hệ thống thông báo: “Bạn cần nhập đầy đủ các thông tin!”. |

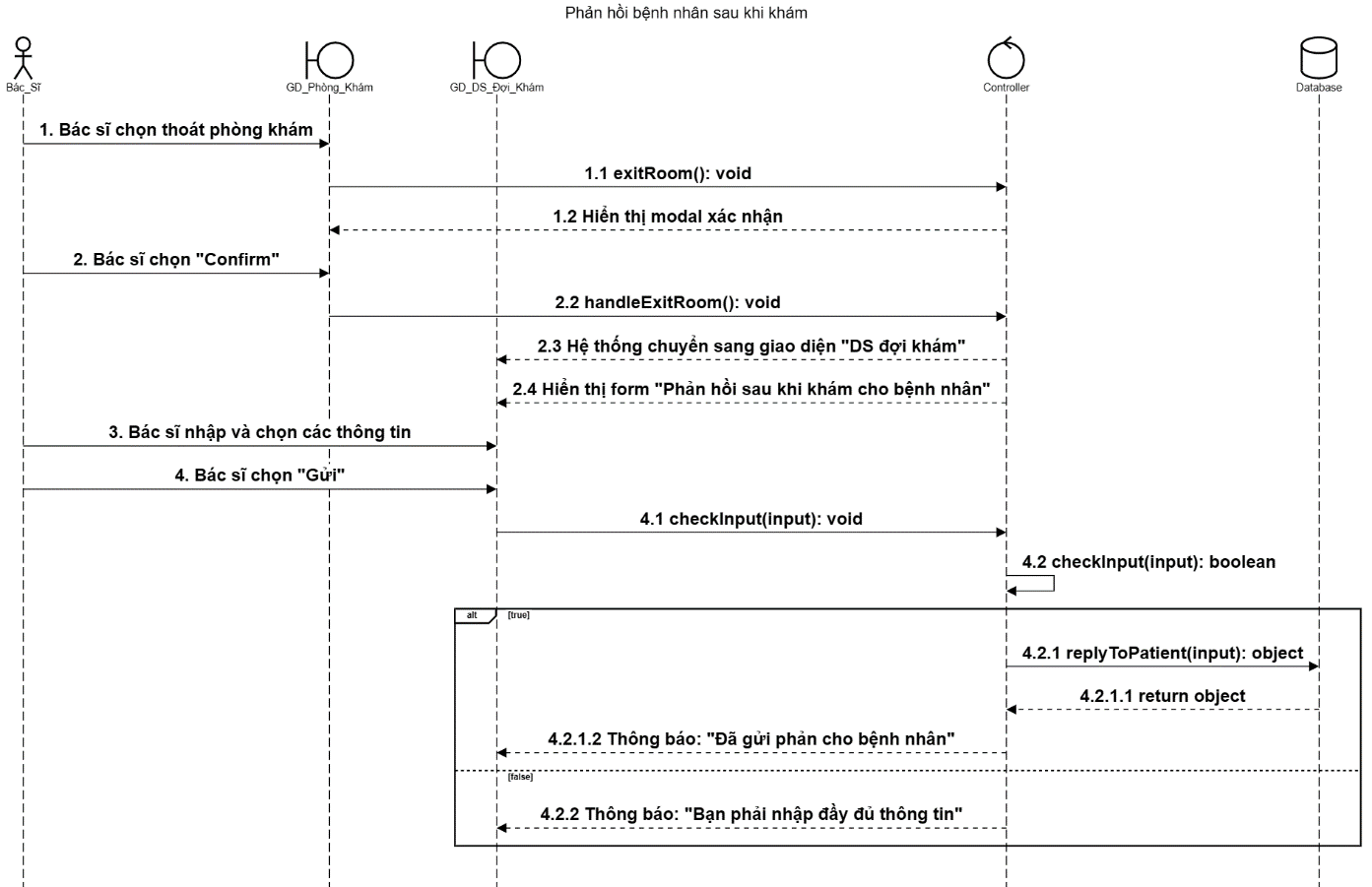
Bảng 4.1.4.7 Đặc tả usecase Nhắc nhở bệnh nhân sau khi khám bệnh

##### 4.1.4.7.1 Activity “Phản hồi bệnh nhân sau khi khám”:



4.1.4.7. Sơ đồ activity Phản hồi bệnh nhân sau khi khám

##### 4.1.4.7.2 Sequence “Phản hồi bệnh nhân sau khi khám”:



4.1.4.7. Sơ đồ sequence Phản hồi bệnh nhân sau khi khám

#### 4.1.4.8 Usecase “Gửi tin nhắn”:

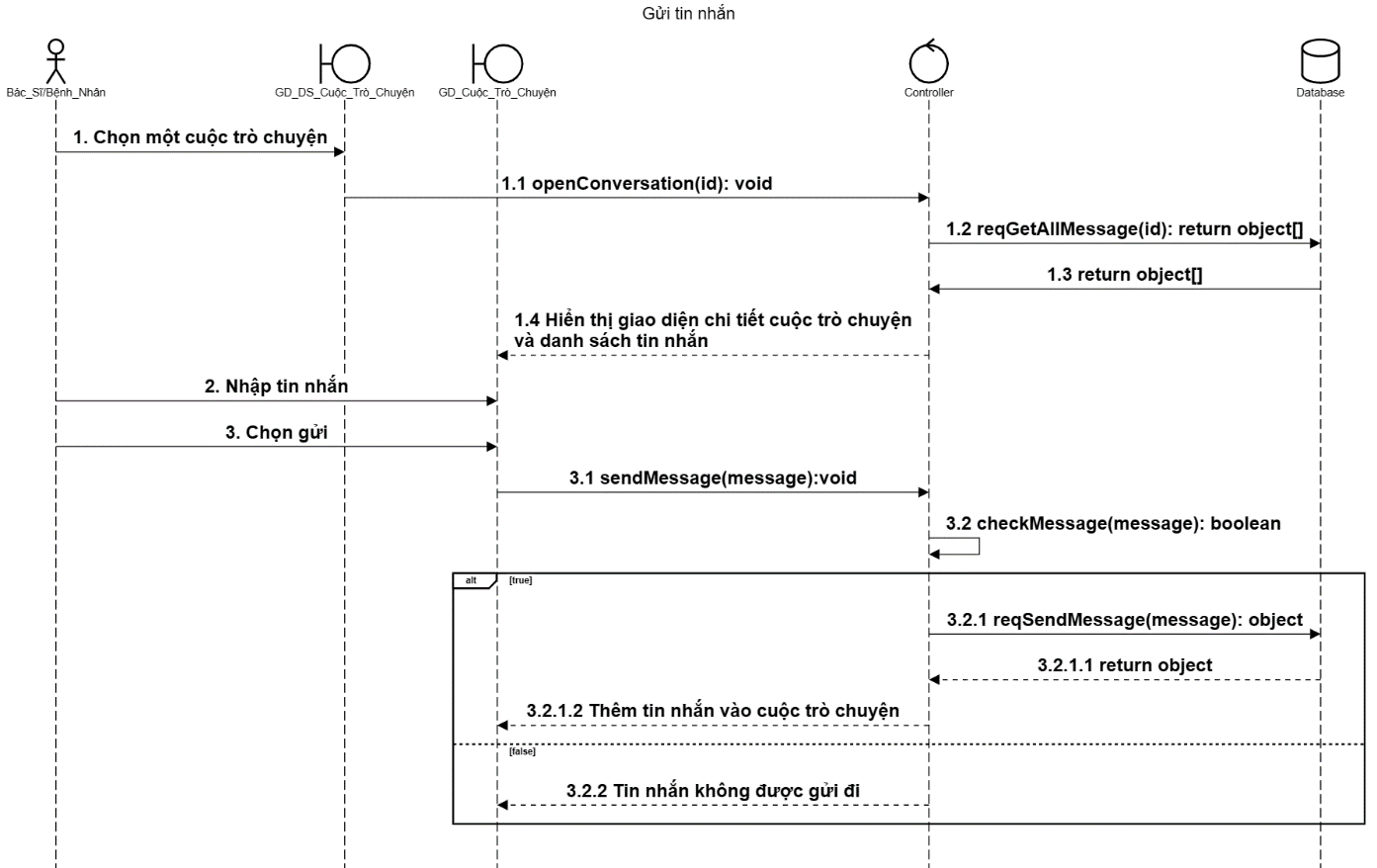
|  |  |
| --- | --- |
| **Tên use case**: Gửi tin nhắn | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bác sĩ/ bệnh nhân có thể thực hiện chức năng nhắn tin. | |
| **Actor chính**: Bác sĩ/ Bệnh nhân | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống và có cuộc trò chuyện. | |
| **Hậu điều kiện (Post-condition):** Gửi tin nhắn đi thành công, hệ thống tự động cập nhật lại các tin nhắn. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bác sĩ/ Bệnh nhân chọn vào một cuộc trò chuyện. |  |
|  | 2. Hệ thống sẽ hiển thị các tin nhắn. |
| 3. Bác sĩ/ Bệnh nhân nhập tin nhắn (text/ hình ảnh/ icon, …). |  |
| 4. Bác sĩ/ Bệnh nhân nhấn nút “Gửi” (hoặc Enter). |  |
|  | 5. Hệ thống xử lý và render ra giao diện. |
| **Luồng sự kiện thay thế (alternate flow):** | |

Bảng 4.1.4.8 Đặc tả usecase Gửi tin nhắn

##### Image17.png4.1.4.8.1 Activity “Gửi tin nhắn”:

4.1.4.8. Sơ đồ activity Gửi tin nhắn

##### 4.1.4.8.2 Sequence “Gửi tin nhắn”:



4.1.4.8. Sơ đồ sequence Gửi tin nhắn

#### 4.1.4.9 Usecase “Chatbot tự động”:

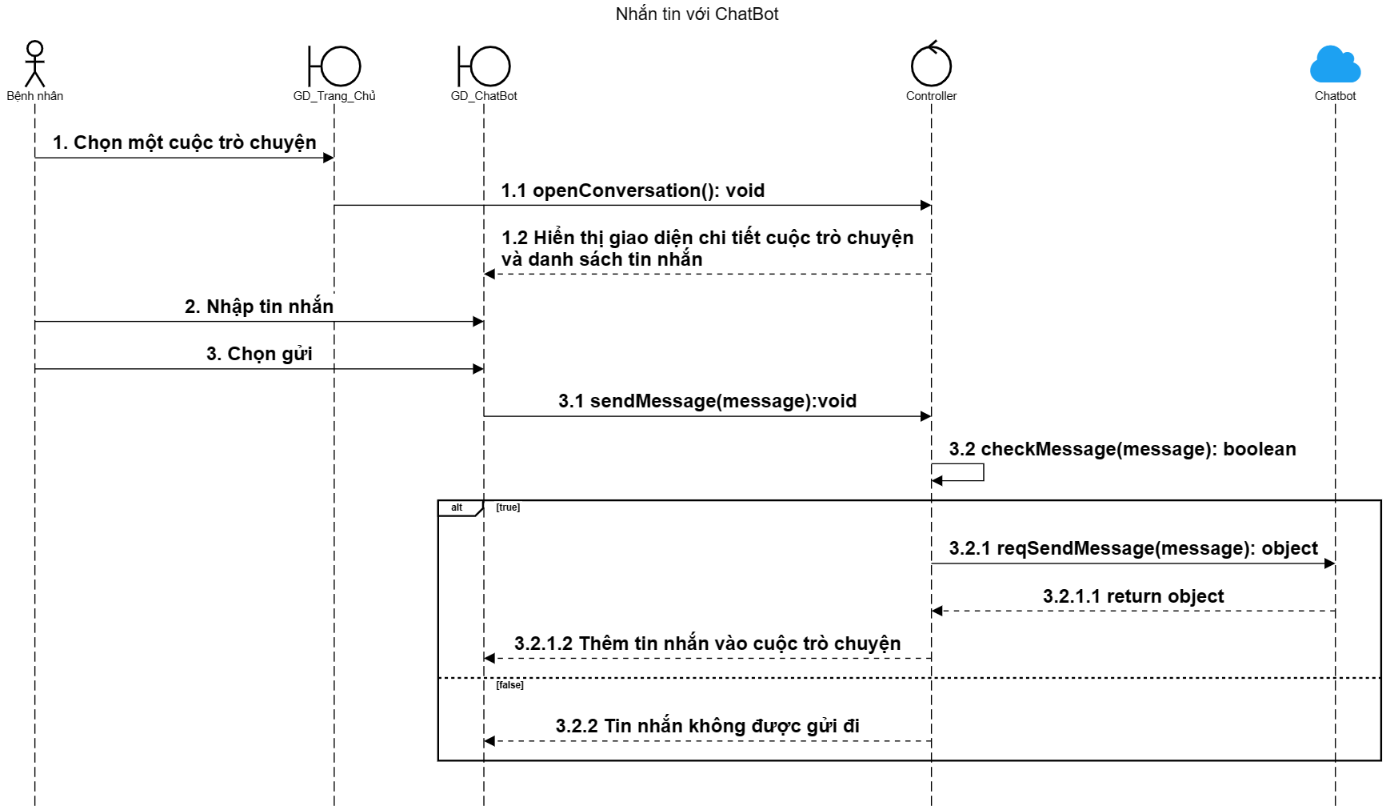
|  |  |
| --- | --- |
| **Tên use case**: Chatbot tự động | |
| **Mô tả sơ lược**: Tư vấn tự động nhằm giải đáp các câu hỏi, thắc mắc cho bệnh nhân. | |
| **Actor chính**: Bệnh nhân | |
| **Actor phụ**: ChatRAG | |
| **Tiền điều kiện (Pre-condition):** Không | |
| **Hậu điều kiện (Post-condition):** Hiển thị ra các câu trả lời để phản hồi các câu hỏi, thắc mắc cho bệnh nhân. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bệnh nhân nhấn vào icon chat (góc dưới bên phải). |  |
|  | 2. Hệ thống sẽ hiển thị giao diện chat. |
| 3. Bệnh nhân nhập các câu hỏi thắc mắc. |  |
| 4. Bệnh nhân nhấn nút “Gửi” (hoặc Enter). |  |
|  | 5. Hệ thống xử lý và tự động render ra câu trả lời thích hợp nhất để giải đáp cho câu hỏi của bệnh nhân. |
| **Luồng sự kiện thay thế (alternate flow):** | |

Bảng 4.1.4.9 Đặc tả usecase Chatbot tự động

##### Image16.png4.1.4.9.1 Activity “Chatbot tự động”:

4.1.4.9. Sơ đồ activity Chatbot tự động

##### 4.1.4.9.2 Sequence “Chatbot tự động”:



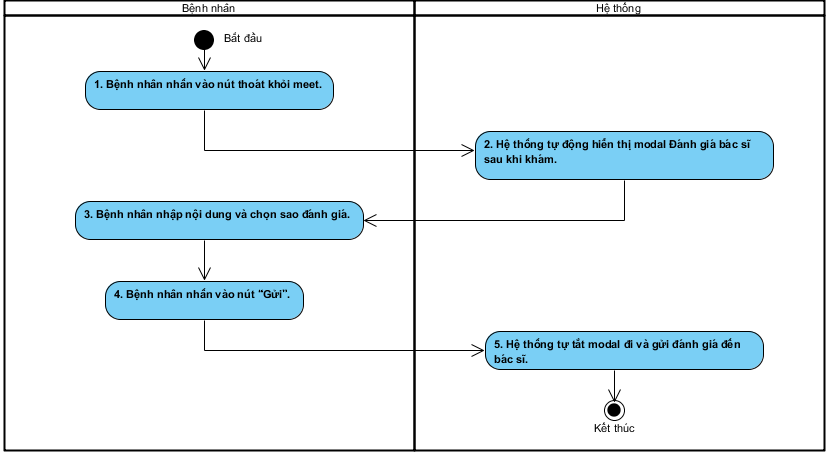
4.1.4.9. Sơ đồ sequence Chatbot tự động

#### 4.1.4.10 Usecase “Đánh giá bác sĩ sau khi khám”:

|  |  |
| --- | --- |
| **Tên use case**: Đánh giá bác sĩ | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống và kết thúc việc khám, bệnh nhân có thể thực hiện chức năng đánh giá bác sĩ sau khi khám. | |
| **Actor chính**: Bệnh nhân | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống và tham gia khám bệnh với bác sĩ | |
| **Hậu điều kiện (Post-condition):** Hệ thống gửi đánh giá đến bác sĩ, danh sách đánh giá của bệnh nhân được cập nhật lại. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bệnh nhân nhấn vào nút thoát khỏi meet. |  |
|  | 2. Hệ thống tự động hiển thị modal Đánh giá bác sĩ sau khi khám. |
| 3. Bệnh nhân nhập nội dung và chọn sao đánh giá. |  |
| 4. Bệnh nhân nhấn vào nút “Gửi”. |  |
|  | 5. Hệ thống tự tắt modal đi và gửi đánh giá đến bác sĩ. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

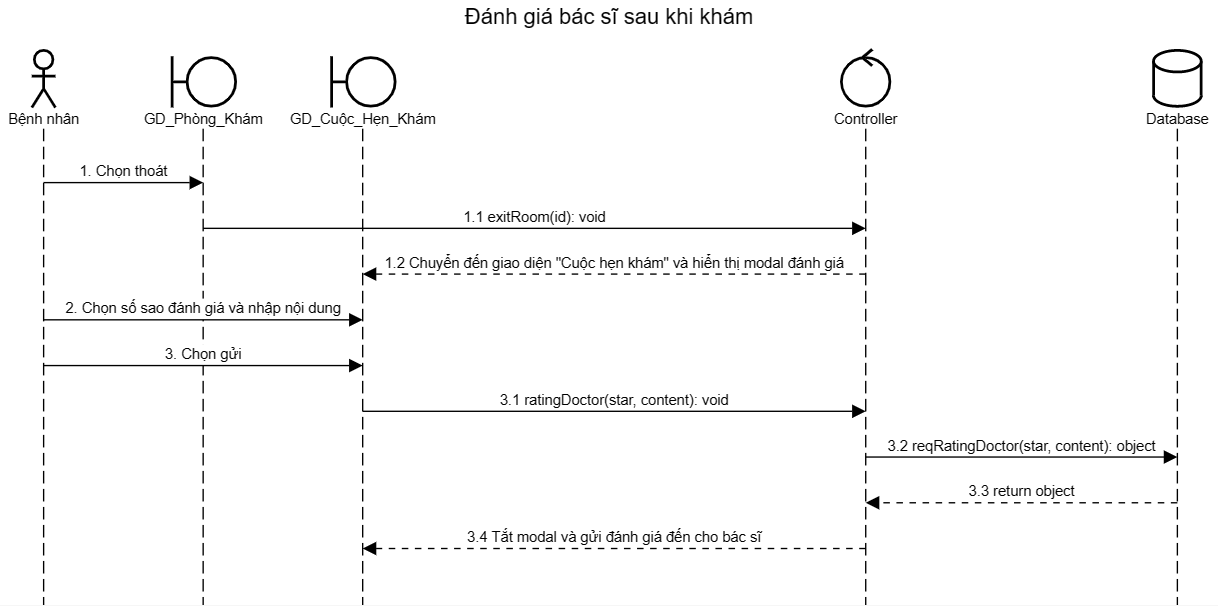
Bảng 4.1.4.10 Đặc tả usecase Đánh giá bác sĩ

##### 4.1.4.10.1 Activity “Đánh giá bác sĩ sau khi khám”:



4.1.4.10. Sơ đồ activity Đánh giá bác sĩ sau khi khám

##### 4.1.4.10.2 Sequence “Đánh giá bác sĩ sau khi khám”:

****

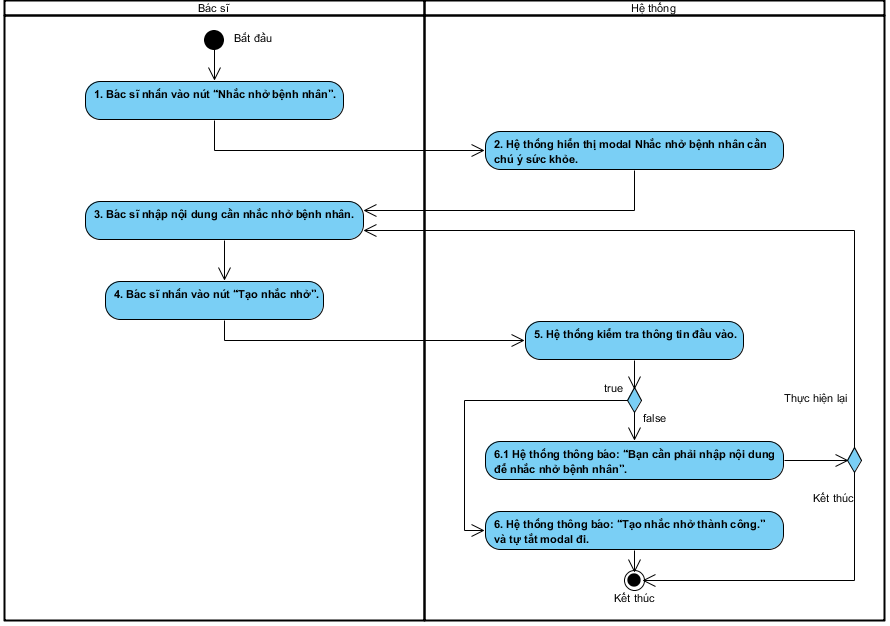
4.1.4.10. Sơ đồ sequence Đánh giá bác sĩ sau khi khám

#### 4.1.4.11 Usecase “Nhắc nhở sức khỏe bệnh nhân”:

|  |  |
| --- | --- |
| **Tên use case**: Nhắc nhở sức khỏe bệnh nhân | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bác sĩ có thể thực hiện chức năng nhắc nhở sức khỏe để gửi đến bệnh nhân. | |
| **Actor chính**: Bác sĩ | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện (Post-condition):** Hệ thống sẽ gửi thông báo đến bệnh nhân. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bác sĩ nhấn vào nút “Nhắc nhở bệnh nhân”. |  |
|  | 2. Hệ thống hiển thị modal Nhắc nhở bệnh nhân cần chú ý sức khỏe. |
| 3. Bác sĩ nhập nội dung cần nhắc nhở bệnh nhân. |  |
| 4. Bác sĩ nhấn vào nút “Tạo nhắc nhở”. |  |
|  | 5. Hệ thống kiểm tra thông tin đầu vào. |
|  | 6. Hệ thống thông báo: “Tạo nhắc nhở thành công.” và tự tắt modal đi. |
| - **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1 Hệ thống thông báo: “Bạn cần phải nhập nội dung để nhắc nhở bệnh nhân”. |

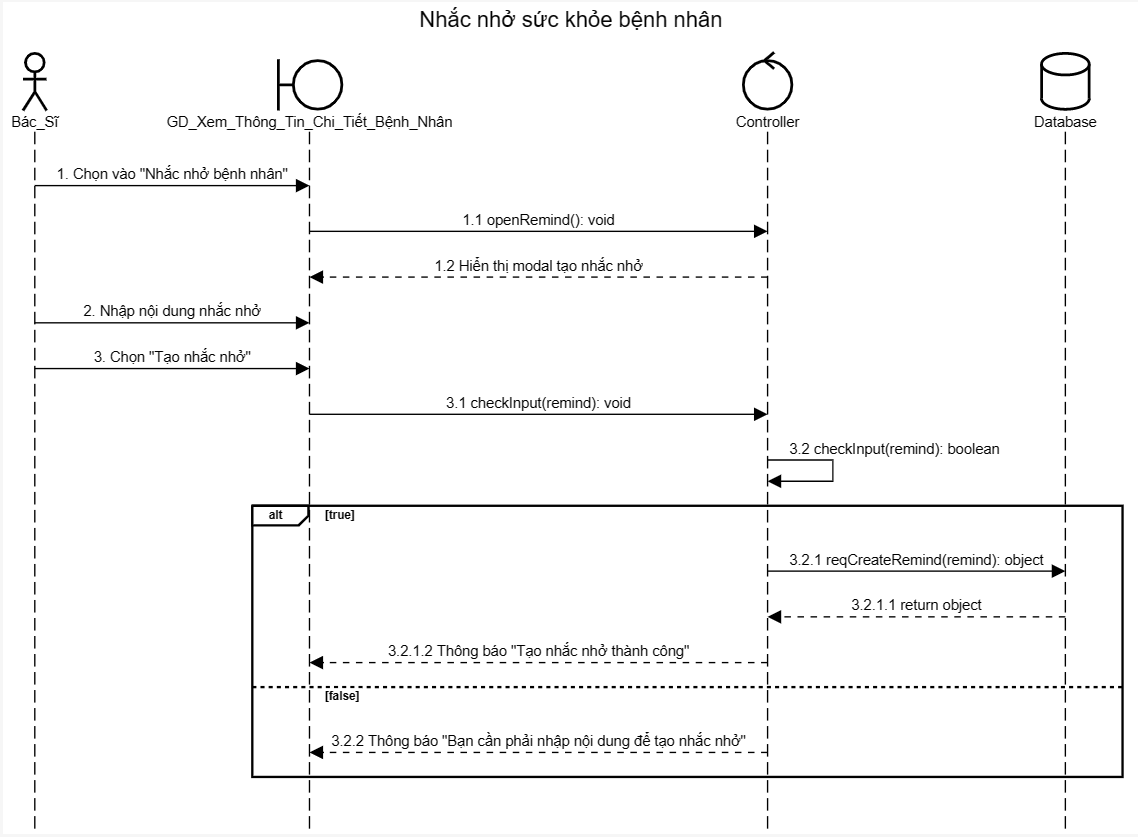
Bảng 4.1.4.11 Đặc tả usecase Nhắc nhở sức khỏe bệnh nhân

##### 4.1.4.11.1 Activity “Nhắc nhở sức khỏe bệnh nhân”:



4.1.4.11. Sơ đồ activity Nhắc nhở sức khỏe bệnh nhân

##### 4.1.4.11.2 Sequence “Nhắc nhở sức khỏe bệnh nhân”:



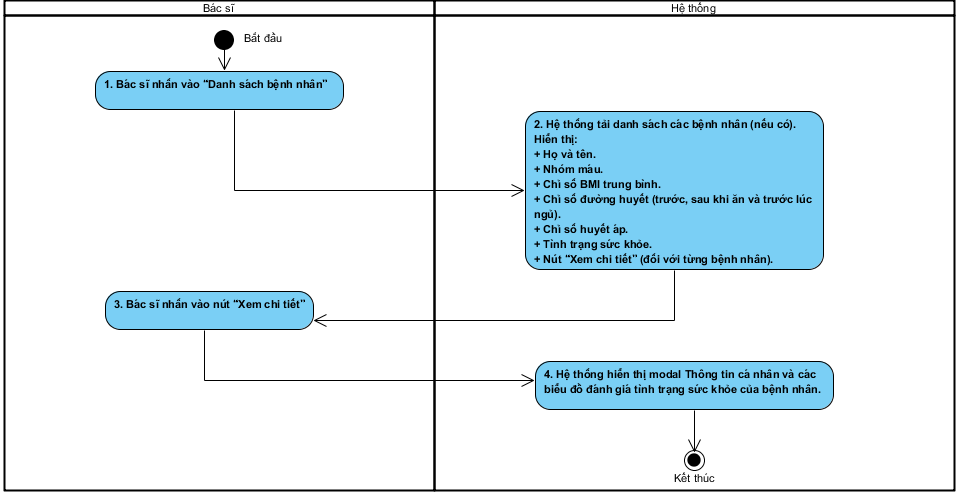
4.1.4.11. Sơ đồ sequence Nhắc nhở sức khỏe bệnh nhân

#### 4.1.4.12 Usecase “Xem đánh giá tổng quan sức khỏe”:

|  |  |
| --- | --- |
| **Tên use case**: Xem đánh giá tổng quan sức khỏe bệnh nhân | |
| **Mô tả sơ lược**: Sau khi đăng nhập thành công vào hệ thống, bác sĩ có thể xem được danh sách đánh giá tổng quan sức khỏe bệnh nhân. | |
| **Actor chính**: Bác sĩ | |
| **Actor phụ**: Không | |
| **Tiền điều kiện (Pre-condition):** Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện (Post-condition):** Thông tin bệnh nhân sẽ được tải và hiển thị. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Bác sĩ nhấn vào “Danh sách bệnh nhân” |  |
|  | 2. Hệ thống tải danh sách các bệnh nhân (nếu có). Hiển thị:  + Họ và tên.  + Nhóm máu.  + Chỉ số BMI trung bình.  + Chỉ số đường huyết (trước, sau khi ăn và trước lúc ngủ).  + Chỉ số huyết áp.  + Tình trạng sức khỏe.  + Nút “Xem chi tiết” (đối với từng bệnh nhân). |
| 3. Bác sĩ nhấn vào nút “Xem chi tiết” |  |
|  | 4. Hệ thống hiển thị modal Thông tin cá nhân và các biểu đồ đánh giá tình trạng sức khỏe của bệnh nhân. |
| **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |

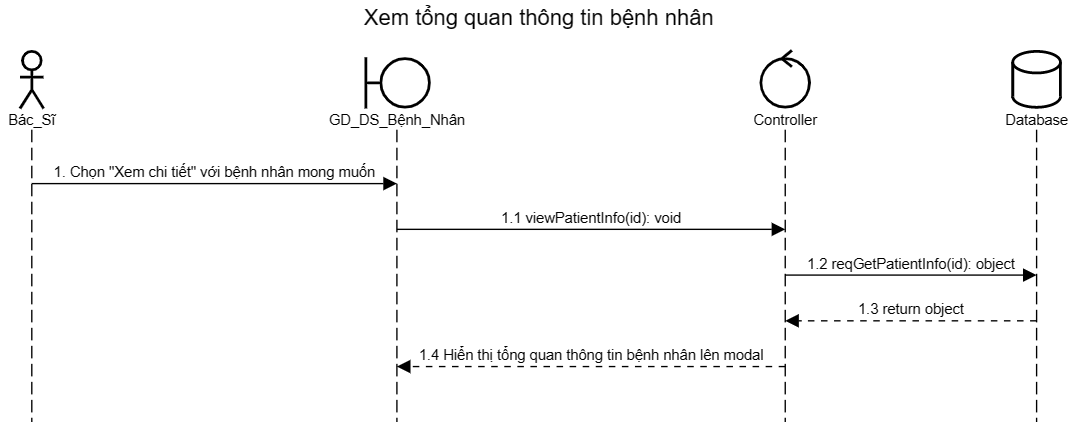
Bảng 4.1.4.12 Đặc tả usecase Xem đánh giá tổng quan bệnh nhân

##### 4.1.4.12.1 Activity “Xem đánh giá tổng quan bệnh nhân”:



4.1.4.12. Sơ đồ activity Xem đánh giá tổng quan bệnh nhân

##### 4.1.4.12.2 Sequence “Xem đánh giá tổng quan bệnh nhân”:

****

4.1.4.12. Sơ đồ sequence Xem đánh giá tổng quan bệnh nhân

#### 4.1.4.13. Usecase “Thanh toán dịch vụ”:

|  |  |
| --- | --- |
| **Tên use case**: Thanh toán dịch vụ | |
| **Mô tả sơ lược**: Bệnh nhân cần phải thanh toán các dịch vụ được yêu cầu trước khi thực hiện các dịch vụ đó. | |
| **Actor chính**: Admin | |
| **Actor phụ**: VietQr | |
| **Tiền điều kiện (Pre-condition):** Có yêu cầu dịch vụ với trạng thái là chờ thanh toán. | |
| **Hậu điều kiện (Post-condition):** Hệ thống sẽ thay đổi trạng thái của yêu cầu thànhđã thanh toán. | |
| **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Chọn yêu cầu dịch vụ cần thanh toán |  |
|  | 2. Hiển thị modal thanh toán chi tiết |
| 3. Chọn phương thức thanh toán |  |
| 4. Nhấn nút thanh toán |  |
|  | 5. Kiểm tra phương thức thanh toán |
|  | 6. Nếu thanh toán bằng tiền mặt hệ thống đóng modal thông báo, hiển thị thông báo chọn có xuất hóa đơn hay không. |
|  | 7. Hiển thị thùy chọn xuất. |
| - **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1. Hiển thị modal chứa QR Code nếu thanh toán chuyển khoản |
|  | 6.2. Hệ thống liên tục kiểm tra các giao dịch. |

Bảng 4.1.4.11 Đặc tả usecase thanh toán dịch vụ

##### 4.1.4.13.1 Activity “Thanh toán dịch vụ”:

|  |
| --- |
| **A diagram of a computer  Description automatically generated** |
| Hình 4.1.4.13.1. Sơ đồ activity “thanh toán dịch vụ” 4.1.4.13.2 Sequence “Thanh toán dịch vụ”:  |  | | --- | | A diagram of a project  Description automatically generated | | Hình 4.1.4.13.2. Sequence “Thanh toán dịch vụ”. | |

## 4.2 Sơ đồ lớp:

|  |
| --- |
| A diagram of a computer flowchart  Description automatically generated |
| Hình 4.2. Sơ đồ lớp của hệ thống NSHealthCare |

Sơ đồ lớp này mô tả chi tiết cấu trúc dữ liệu và mối quan hệ trong hệ thống quản lý phòng khám y tế. Hệ thống có một lớp cơ bản là Person, lưu giữ thông tin chung như họ tên, địa chỉ, email, và số điện thoại, được kế thừa bởi hai lớp Patient và Doctor để quản lý thông tin cụ thể cho bệnh nhân và bác sĩ. Bệnh nhân (Patient) có thể có nhiều cuộc hẹn (Appointment), hồ sơ y tế (MedicalRecord), và xét nghiệm y tế (MedicalTest). Bác sĩ (Doctor) được liên kết với các loại khám (ExaminationType) mà họ chuyên trách và có thể phụ trách nhiều cuộc hẹn với bệnh nhân.

Mỗi cuộc hẹn (Appointment) được quản lý chi tiết với thông tin về thời gian, bệnh nhân, và bác sĩ liên quan. Cuộc hẹn có thể sinh ra các hồ sơ y tế, trong đó chứa thông tin về chẩn đoán, dấu hiệu sinh tồn, phương pháp điều trị, và đơn thuốc (Prescription). Đơn thuốc được liên kết với các loại thuốc (Medicine), mỗi loại thuốc được quản lý chi tiết theo lô hàng (Batch) để theo dõi tồn kho và hạn sử dụng. Xét nghiệm y tế (MedicalTest) được lưu trữ với trạng thái, chẩn đoán ban đầu, và kết quả, gắn với từng cuộc hẹn và bệnh nhân cụ thể.

Hệ thống còn hỗ trợ quản lý các dịch vụ y tế (Service) như khám tổng quát hay xét nghiệm máu, được liên kết với các loại khám (ExaminationType). Thông báo (Notification) gửi đến bệnh nhân hoặc bác sĩ để cung cấp thông tin quan trọng. Ngoài ra, phần nội dung như bài viết (Blog) và danh mục bài viết (Category) cũng được quản lý để cung cấp thông tin hữu ích cho bệnh nhân. Với thiết kế này, hệ thống cung cấp một cách tiếp cận toàn diện, hiệu quả để quản lý mọi khía cạnh của phòng khám y tế, từ lịch hẹn, hồ sơ y tế, đến thuốc và dịch vụ.

## 4.3 Kiến trúc phần mềm:

Kiến trúc phần mềm này kết hợp giữa Next.js ở phía client và Nest.js ở phía server để xây dựng một hệ thống hiệu quả, bảo mật và có khả năng mở rộng. Next.js đại diện cho phần giao diện người dùng (frontend), nơi tiếp nhận các yêu cầu từ người dùng thông qua các HTTP Request và chuyển tiếp chúng đến backend. Tại phía server, Nest.js đảm nhiệm vai trò xử lý logic với cấu trúc module hóa. Các yêu cầu được định tuyến qua các route và lần lượt đi qua các thành phần như CORS Middleware để kiểm tra và xử lý chính sách chia sẻ tài nguyên, sau đó đến JWT Authentication Middleware để xác thực người dùng thông qua token JWT. Khi yêu cầu được xác nhận hợp lệ, nó được chuyển đến Controller, nơi chịu trách nhiệm định tuyến và gọi đến Service Layer để thực hiện logic nghiệp vụ cụ thể. Service Layer tiếp tục giao tiếp với Data Access Layer để lấy dữ liệu từ cơ sở dữ liệu MongoDB hoặc tương tác với các dịch vụ bên ngoài như AWS thông qua AWS SDK.

|  |
| --- |
| A diagram of a software process  Description automatically generated |
| Hình 4.2. Kiến trúc phần mềm hệ thống NSHealthCare |

Hệ thống sử dụng Mongoose, một thư viện ODM mạnh mẽ, để quản lý và truy vấn dữ liệu từ cơ sở dữ liệu MongoDB. Ngoài ra, tích hợp với AWS SDK cung cấp khả năng sử dụng các dịch vụ đám mây như lưu trữ tệp (S3) hoặc xử lý dữ liệu nâng cao. Sau khi xử lý xong yêu cầu, Nest.js trả phản hồi về cho Next.js, nơi kết quả được render và hiển thị trên giao diện người dùng. Với cách phân tầng rõ ràng và sử dụng các công nghệ tiên tiến, kiến trúc này đảm bảo tính bảo mật, hiệu suất và khả năng mở rộng cho hệ thống.

## 4.4 Sơ đồ deployment:

## A diagram of a cloud computing system Description automatically generated

Hình 4.3. Sơ đồ deloyment của hệ thống NSHeathcare.

Sơ đồ triển khai này minh họa kiến trúc hệ thống của một ứng dụng web, với các thành phần chính được triển khai trên Google Cloud VM Instances. Frontend được xây dựng bằng Next.js, được triển khai trên Nginx (port 80) để phục vụ các file tĩnh và xử lý giao tiếp từ người dùng. Backend, được xây dựng bằng NestJS + Node.js, chạy trên cùng một Google Cloud VM Instance (port 3002) và đảm nhiệm logic nghiệp vụ chính của hệ thống.

Backend giao tiếp trực tiếp với:

* MongoDB Atlas Database: Lưu trữ dữ liệu như thông tin người dùng, hồ sơ y tế, và lịch hẹn. Backend sử dụng MongoDB Connection String để kết nối với cơ sở dữ liệu.
* AWS S3 Bucket: Lưu trữ các tệp tĩnh, như hình ảnh và tài liệu đính kèm. Backend sử dụng S3 API để tải lên hoặc truy xuất tệp.
* Infura (Ethereum Blockchain): Tương tác với các smart contract được triển khai trên Ethereum blockchain, chẳng hạn để xử lý giao dịch hoặc xác minh dữ liệu blockchain. Kết nối này được thực hiện thông qua Infura API.

Ngoài ra, frontend (Nginx) có thể trực tiếp truy cập AWS S3 Bucket để tải các file như hình ảnh hoặc tài liệu thông qua URL công khai hoặc presigned URL. Hệ thống được thiết kế để các thành phần giao tiếp với nhau qua giao thức HTTPS, đảm bảo tính bảo mật và hiệu quả. Tổng thể, kiến trúc này cung cấp một hệ thống linh hoạt, mạnh mẽ, và dễ dàng mở rộng cho các ứng dụng web phức tạp.

# CHƯƠNG 5: THIẾT KẾ VÀ HIỆN THỰC

## 5.1 Giao diện thiết kế:

### 5.1.1 Giao diện web:

A close-up of a doctor

Description automatically generated

#### Hình 5.1.2.1 Trang chủ – Bệnh nhân

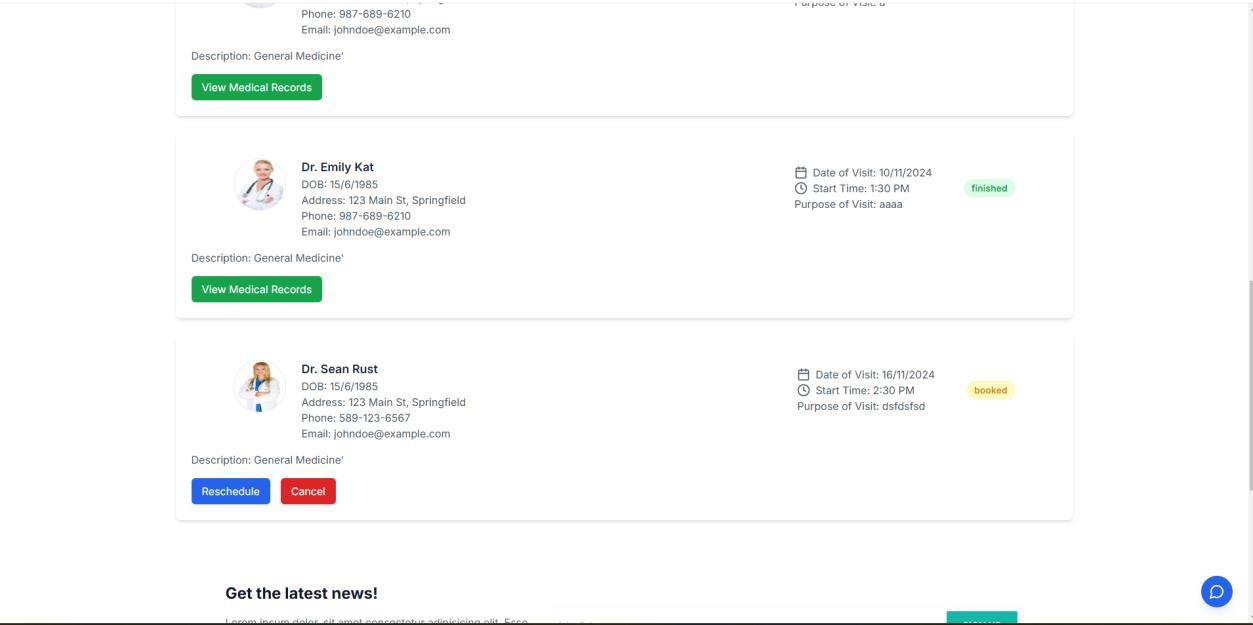
A screenshot of a computer

Description automatically generated

#### Hình 5.1.2.1 Trang đặt lịch khám – Bệnh nhân

Giao diện web cung cấp chức năng đặt lịch hẹn với bác sĩ, cho phép người dùng chọn ngày từ lịch, lựa chọn khung giờ phù hợp trong danh sách thời gian khả dụng, và nhập mục đích hẹn.

Người dùng có thể nhấn "Submit" để xác nhận hoặc "Close" để hủy thao tác. Ngoài ra, giao diện còn hiển thị thông tin chi tiết của bác sĩ, bao gồm hình ảnh, tên, chuyên môn và địa chỉ,cùng với danh sách gợi ý các bác sĩ khác để người dùng tham khảo.



A screenshot of a computer

Description automatically generated

#### Hình 5.1.2.1 Trang hiển thị lịch sử khám bệnh – Bệnh nhân

Giao diện web hiển thị danh sách các cuộc hẹn với bác sĩ, bao gồm thông tin chi tiết như tên bác sĩ, ngày sinh, địa chỉ, số điện thoại, email, và mô tả chuyên môn.

Mỗi cuộc hẹn đi kèm thông tin về ngày giờ, mục đích khám, và trạng thái cuộc hẹn (đã hoàn thành, đã đặt). Người dùng có thể xem hồ sơ y tế qua nút "View Medical Records" hoặc thực hiện các thao tác khác như "Reschedule" để thay đổi lịch hẹn và "Cancel" để hủy lịch hẹn.

A screenshot of a graph

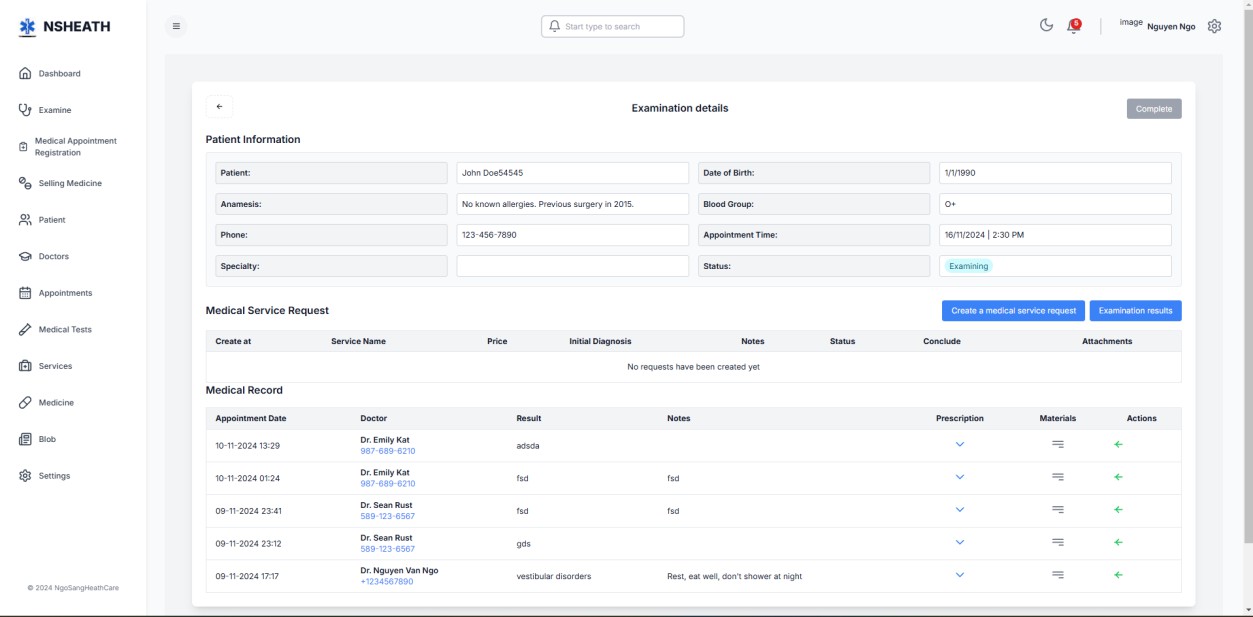
Description automatically generated

#### Hình 5.1.2.1 Trang Dashboard – Bác sĩ

A screenshot of a graph

Description automatically generated

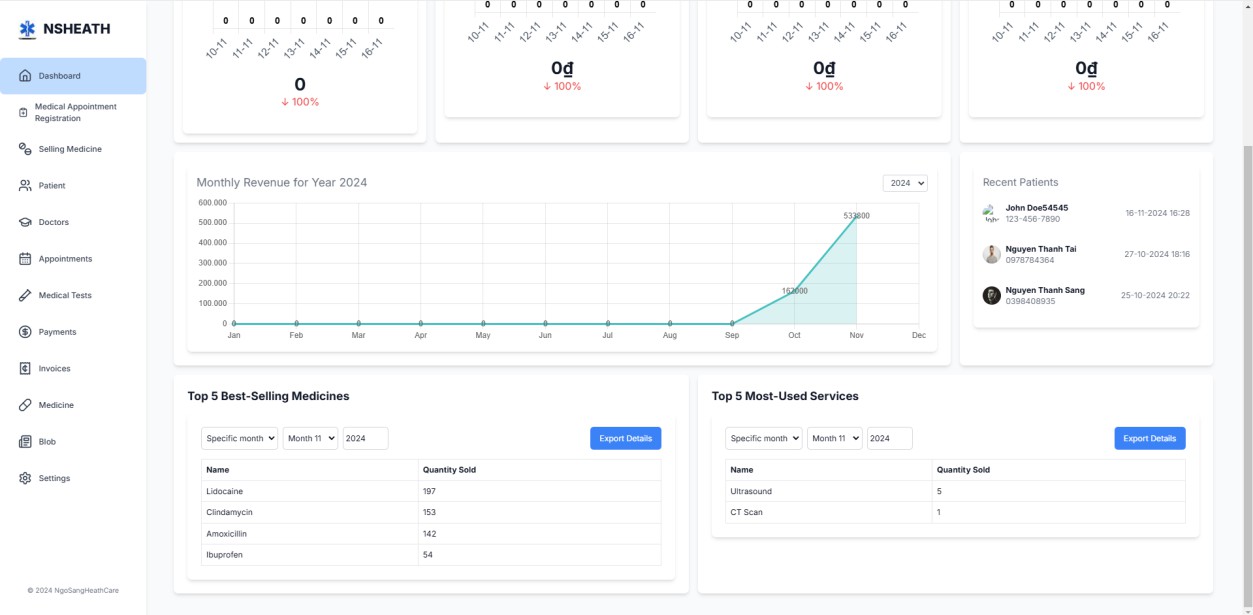
#### Hình 5.1.2.2 Giao diện khi có thông báo – Bác sĩ



#### Hình 5.1.2.6 Giao diện khám bệnh– Bác sĩ

Giao diện khám bệnh cung cấp thông tin chi tiết về bệnh nhân và tình trạng hiện tại, đồthời hỗ trợ các chức năng quan trọng trong quá trình khám bệnh (hình 3.6). Người dùng có thể sử dụng nút "Tạo yêu cầu dịch vụ" để tạo các yêu cầu chẩn đoán như siêu âm, chụp CT, xét nghiệm máu,...

Ngoài ra, nút "Tạo kết quả khám bệnh" cho phép nhập và lưu trữ kết quả khám bệnh sau khi hoàn tất quá trình thăm khám. Các chức năng này được thiết kế để hỗ trợ bác sĩ quản lý quá trình khám chữa bệnh một cách nhanh chóng và hiệu quả.



#### Hình 5.1.2.8 Giao diện Dashboard - Admin

Giao diện thống kê cung cấp thông tin tổng hợp về hoạt động của hệ thống y tế, bao gồm các

biểu đồ và bảng số liệu chi tiết (Hình 3.9). Người dùng có thể xem thống kê doanh thu, số

lượng lịch hẹn khám theo thời gian linh hoạt (tuần, tháng, năm) thông qua bộ chọn thời gian

ở góc trên. Giao diện có nút "Xuất file Excel" cho phép tải xuống dữ liệu thống kê chi tiết để

phục vụ công việc quản lý hoặc báo cáo.

Phần doanh thu được hiển thị bằng biểu đồ cột hoặc đường, giúp so sánh doanh thu theo từng

giai đoạn hoặc theo từng bác sĩ, dịch vụ. Phần thống kê lịch hẹn hiển thị tổng số lịch hẹn, số

lịch hoàn thành, đã hủy, và đang chờ xử lý, với biểu đồ tròn hoặc dạng bảng chi tiết. Ngoài

ra, giao diện cũng bao gồm các chỉ số tăng trưởng hoặc giảm sút qua từng giai đoạn, hỗ trợ

phân tích xu hướng.

### 5.1.2 Yêu cầu về phần cứng:

|  |  |  |  |
| --- | --- | --- | --- |
| CPU | RAM | HDD | Architecture |
| Intel Core i3 trở lên | 4GB | 256GB | 64 bit |

**CHƯƠNG 6. KIỂM THỬ HỆ THỐNG.**

### 6.1. Danh sách các tình huống kiểm thử các chức năng chính ứng dụng:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TestID** | **Ngày thực hiện** | **Chức năng** | **Mô tả** | **Quy trình kiểm tra** | **Kết quả mong muốn** | **Kết quả thực tế** | **Kết quả** |
| T01 | 25/11/2024 | Đặt lịch khám | Kiểm tra hệ thống có tạo ra lịch khám mới không. | 1. Chọn bác sĩ.  2. Nhập tình trạng, lí do đặt lịch khám. | Tạo lịch hẹn khám thành công với trạng thái “booked”. | Tạo lịch hẹn khám thành công với trạng thái “booked”. | Pass |
| T02 | 25/11/2024 | Tạo yêu cầu dịch vụ | Kiểm tra hệ thống có tạo thành công yêu cầu dịch vụ không | 1. Bác sĩ chọn dịch vụ muốn tạo yêu cầu.  2. Nhập chẩn đoán ban đầu, ghi chú (nếu có).  3. Nhấn nút “request”. | Tạo yêu cầu dịch vụ thành công với trạng thái “awaiting payment”. | Tạo yêu cầu dịch vụ thành công với trạng thái “paid”. | Fail |
| T03 | 25/11/2024 | Tạo hồ sơ bệnh án | Kiểm tra hệ thống có lưu thành công hồ sơ bệnh án không | 1. Bác sĩ chọn nút tạo kết quả khám.  2. Nhập chẩn đoán, kê đơn thuốc.  3. Nhấn nút save changes. | Tạo hồ sơ bệnh án thành công. | Tạo hồ sơ bệnh án thành công. | Pass |
| T04 | 25/11/2024 | Kê đơn thuốc | Kiểm tra hệ thống có tạo thành công đơn thuốc không | 1. Bác sĩ tìm kiếm thuốc theo tên.  2. Chọn loại thuốc, nhập số ngày sử dụng, số lượng mỗi lần sử dụng, cách sư dụng.  3. Chọn nút thêm. | Thuốc và cách sử dụng được thêm vào toa thuốc. | Thuốc và cách sử dụng được thêm vào toa thuốc. | Pass |
| T05 | 25/11/2024 | Checkin lịch hẹn | Kiểm tra hệ thống có cập nhật thành công trạng thái không | 1. Đăng nhập vào hệ thống.  2. Thực hiện check-in cho lịch hẹn đã đặt. | Trạng thái lịch hẹn được cập nhật thành "waiting". | Trạng thái lịch hẹn được cập nhật thành "waiting". | Pass |
| T06 | 25/11/2024 | Xem hồ sơ bệnh án | Kiểm tra hệ thống có hiển thị đúng thông tin hồ sơ bệnh án không | 1. Đăng nhập vào hệ thống.  2. Tìm và mở hồ sơ bệnh án của bệnh nhân cụ thể. | Thông tin hồ sơ bệnh án được hiển thị chính xác. | Thông tin hồ sơ bệnh án được hiển thị chính xác. | Pass |
| T07 | 25/11/2024 | Hủy lịch khám | Kiểm tra hệ thống có hủy thành công lịch khám không. | 1. Đăng nhập vào hệ thống.  2. Chọn lịch khám cần hủy.  3. Nhấn nút "Hủy". | Lịch khám được hủy thành công và trạng thái chuyển sang “cancelled”. | Lịch khám được hủy thành công và trạng thái chuyển sang “cancelled”. | Pass |
| T08 | 25/11/2024 | Thanh toán dịch vụ bằng hình thức chuyển khoản. | Kiểm tra việc thanh toán dịch vụ khám bệnh qua VietQR | 1. Chọn dịch vụ cần thanh toán.  2. Quét mã QR của VietQR và hoàn tất thanh toán. | Thanh toán thành công và trạng thái dịch vụ chuyển sang "paid". | Thanh toán thành công và trạng thái dịch vụ chuyển sang "paid". | Pass |
| T09 | 25/11/2024 | Thống kê doanh thu | Kiểm tra chức năng thống kê doanh thu dựa trên các dịch vụ khám và đơn thuốc | 1. Truy cập mục thống kê.  2. Chọn thời gian cần thống kê.  3. Xem và xuất dữ liệu thống kê ra file Excel. | Dữ liệu doanh thu được hiển thị chính xác và file Excel được xuất thành công. | Dữ liệu doanh thu được hiển thị chính xác. Không xuất được file excel. | Fail |
| T10 | 25/11/2024 | Chatbot tư vấn | Kiểm tra chatbot phản hồi đúng với câu hỏi cơ bản của người dùng. | 1. Mở chatbot AI.  2. Nhập câu hỏi như "Triệu chứng ho có nguy hiểm không?" | Chatbot phản hồi thông tin cơ bản liên quan đến câu hỏi. | Chatbot phản hồi thông tin cơ bản liên quan đến câu hỏi. | Pass |
| T11 | 25/11/2024 | Bán thuốc theo đơn đã kê. | Kiểm tra hệ thống có xử lý bán thuốc theo đơn thuốc đã kê hay không | 1. Đăng nhập vào hệ thống.  2. Chọn bệnh nhân hoặc tìm kiếm đơn thuốc đã kê.  3. Xác nhận số lượng thuốc và nhấn "Bán thuốc".  4. In hóa đơn bán thuốc. | Giao dịch bán thuốc được xử lý thành công và thông tin được lưu lại trong hệ thống. | Giao dịch bán thuốc được xử lý thành công và thông tin được lưu lại trong hệ thống. | Pass |

**6.2**. **Danh sách một số các test case còn lại :**

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Chức năng** | **Kết quả** |
| T12 | Đăng nhập | Pass |
| T13 | Thêm dịch vụ mới | Pass |
| T14 | Đổi mật khẩu | Pass |
| T15 | Phục hồi mật khẩu | Pass |
| T16 | Đăng xuất | Pass |
| T17 | Lọc danh sách lịch khám | Pass |
| T18 | Thanh toán đơn thuốc | Pass |
| T19 | Kiểm tra thông tin dịch vụ | Pass |
| T20 | Xem lại đơn thuốc | Pass |
| T21 | Thêm thuốc mới | Pass |
| T22 | Xem lại thông tin khám | Pass |
| T23 | Cập nhật thông tin dịch vụ | Fail |
| T24 | Xem thông tin hóa đơn | Pass |
| T25 | In hóa đơn thanh toán | Pass |
| T26 | Cập nhật thông tin bệnh nhân | Pass |
| T27 | Thêm dịch vụ mới | Pass |
| T28 | Cập nhật dịch vụ | Pass |
| T29 | Xem thông tin cá nhân | Pass |
| T30 | Cập nhật thông tin cá nhân | Fail |
| T31 | Nhập lô thuốc | Pass |
| T32 | Cập nhật thông tin thuốc | Pass |
| T33 | Thêm tài khoản bác sĩ mới | Chưa thực hiện |
| T34 | Tạo kết quả dịch vụ yêu cầu | Pass |
| T35 | Tìm kiếm lịch hẹn theo bác sĩ | Chưa thực hiện |

**6.3. Thống kê kết quả kiểm thử.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tổng số Test Case** | **Số Test Case Pass** | **Số Test Case Fail** | **Số Test Case Chưa Thực Hiện** |
| 35 | 29 | 4 | 2 |

# CHƯƠNG 7: KẾT LUẬN

## 7.1 Kết quả đạt được:

Nhóm đã xây dựng và triển khai thành công một hệ thống quản lý y tế thông minh trên nền tảng web, đáp ứng các nhu cầu cơ bản và cần thiết trong hoạt động y tế. Hệ thống bao gồm các chức năng chính như đăng ký lịch hẹn trực tuyến, quản lý quy trình khám chữa bệnh, tạo và quản lý hồ sơ bệnh án, cũng như hỗ trợ tạo các yêu cầu dịch vụ khám trong quá trình khám. Ngoài ra, hệ thống cung cấp khả năng quản lý các loại thuốc, lô thuốc, hỗ trợ kê đơn và bán thuốc theo đơn đã được kê.

Hệ thống tích hợp VietQR trong việc xử lý tự động các thanh toán chuyển khoản, giúp xử lý nhanh chóng và thuận tiện các giao dịch chuyển khoản. Bên cạnh đó, chatbot AI được tích hợp nhằm hỗ trợ tư vấn tự động, mang lại sự tiện lợi cho người dùng. Hệ thống cũng thực hiện tốt việc quản lý thông tin người dùng.

Hệ thống hỗ trợ các chức năng thống kê cần thiết, bao gồm thống kê doanh thu theo dịch vụ, đơn thuốc, số lượng dịch vụ và thuốc đã bán. Ngoài ra, người dùng có thể xuất các báo cáo chi tiết dưới dạng Excel, cung cấp thông tin cụ thể về số lần dịch vụ được sử dụng và các loại thuốc bán chạy. Nhờ đó, hệ thống giúp các cơ sở y tế dễ dàng quản lý và tối ưu hóa hoạt động. Với các tính năng cơ bản và hữu ích, hệ thống đã hoàn thành mục tiêu hỗ trợ quản lý hiệu quả và nâng cao trải nghiệm cho cả bệnh nhân lẫn đội ngũ y tế.

Nhóm đã tìm hiểu và triển khai thành công một số công nghệ trong hệ thống, cụ thể như sau:

* Triển khai thành công công nghệ Progressive Web App (PWA), giúp hệ thống hoạt động hiệu quả trên mọi thiết bị, từ máy tính đến điện thoại di động.
* Ứng dụng công nghệ blockchain để lưu trữ và quản lý dữ liệu y tế, đảm bảo tính an toàn và minh bạch.
* Tích hợp smart contracts vào hệ thống nhằm lưu trữ hồ sơ bệnh án, đảm bảo tính minh bạch, không thể thay đổi, và chỉ truy cập bởi những bên được cấp quyền.
* Tìm hiểu và triển khai hệ thống trên nền tảng Google Cloud Platform (GCP), đảm bảo tính ổn định, hiệu suất cao và khả năng mở rộng linh hoạt.
* Sử dụng các dịch vụ đám mây như cơ sở dữ liệu, lưu trữ tệp và triển khai ứng dụng.
* Tích hợp AI chatbot dựa trên công nghệ RAG (Retrieval Augmented Generation) để hỗ trợ bệnh nhân tra cứu thông tin, nhận tư vấn ban đầu dựa trên dữ liệu y khoa.

Thông qua đồ án quá trình thực hiện đồ án, nhóm đã tích lũy được nhiều kiến thức và làm quen với các quy trình triển khai ứng dụng thực tế, từ phát triển, thử nghiệm đến triển khai trên môi trường thực tế. Đầu tiên, nhóm hiểu sâu hơn về vai trò của smart contracts trong việc quản lý dữ liệu y tế, giúp đảm bảo tính minh bạch và bảo mật cho hồ sơ bệnh án. Việc ứng dụng công nghệ blockchain không chỉ tăng cường độ tin cậy mà còn đảm bảo an toàn tuyệt đối cho người dùng, đặc biệt trong ngành y tế, nơi dữ liệu nhạy cảm cần được bảo vệ chặt chẽ. Ngoài ra, nhóm đã làm quen và triển khai thành công hệ thống trên Google Cloud, đảm bảo hiệu năng ổn định, khả năng mở rộng và đáp ứng nhu cầu sử dụng đa dạng của các bên liên quan.

Đồng thời, nhóm đã nắm bắt được phương pháp tổ chức, quản lý và phân phối dữ liệu y tế trong một hệ thống phức tạp với nhiều vai trò khác nhau, bao gồm bệnh nhân, bác sĩ và quản trị viên. Những kiến thức và kỹ năng này không chỉ giúp nhóm hoàn thành đồ án mà còn là nền tảng cho các dự án công nghệ trong tương lai.

## 7.2 Hạn chế của đồ án:

* Hệ thống chưa được kiểm tra kỹ lưỡng trong nhiều tình huống thực tế phức tạp, đặc biệt là các trường hợp ngoại lệ như lỗi mạng, lỗi xác thực tài khoản, hoặc xung đột dữ liệu trong quá trình xử lý. Điều này có thể ảnh hưởng đến độ tin cậy và tính ổn định khi triển khai hệ thống ở môi trường thực tế.
* Hệ thống hiện tại chỉ hỗ trợ các chức năng thống kê cơ bản như biểu đồ và số liệu tổng quan. Các tính năng phân tích nâng cao, chẳng hạn như dự đoán xu hướng bệnh tật, phân tích hành vi bệnh nhân, hoặc hỗ trợ ra quyết định lâm sàng dựa trên dữ liệu y tế, vẫn chưa được triển khai.
* Chatbot sử dụng công nghệ RAG hiện chỉ hỗ trợ các câu trả lời cơ bản và chưa tận dụng đầy đủ khả năng xử lý ngôn ngữ tự nhiên (NLP). Điều này dẫn đến việc chatbot chưa thể xử lý hoặc trả lời các câu hỏi phức tạp của bệnh nhân một cách chính xác và hiệu quả.
* Việc tích hợp blockchain và PWA tuy mang lại nhiều lợi ích nhưng cũng làm tăng độ phức tạp và yêu cầu tài nguyên cao khi vận hành, đặc biệt là với các hệ thống có số lượng lớn người dùng hoặc dữ liệu y tế cần xử lý.

## 7.3 Hướng phát triển:

Sau khi hoàn thành quá trình thực hiện và báo cáo đồ án, nhóm đã đặt ra mục tiêu tiếp tục mở rộng và phát triển thêm nhiều tính năng mới nhằm nâng cao chất lượng cũng như khả năng ứng dụng thực tế của NS Healthcare. Mục tiêu chính là đáp ứng tốt hơn nhu cầu sử dụng của người dùng, đồng thời mang lại trải nghiệm tối ưu và tiện lợi hơn khi sử dụng hệ thống.

Nhóm dự định tập trung phát triển thêm các tính năng quản lý nhiều loại bệnh chi tiết hơn, từ bệnh mãn tính đến các loại bệnh cần điều trị đặc biệt. Việc này không chỉ giúp hệ thống hỗ trợ tốt hơn cho bệnh nhân mà còn cung cấp các công cụ mạnh mẽ cho bác sĩ trong việc theo dõi và điều trị bệnh tật. Điều này sẽ làm cho hệ thống trở nên toàn diện hơn, thu hút thêm nhiều người sử dụng và mở rộng phạm vi ứng dụng ra cộng đồng lớn hơn.

Bên cạnh đó, nhóm dự kiến triển khai một cơ chế tính phí và quản lý tài chính minh bạch, nhằm không chỉ hỗ trợ bệnh nhân dễ dàng thanh toán các dịch vụ y tế. Cơ chế này sẽ tích hợp các công nghệ thanh toán hiện đại, hỗ trợ đa dạng hình thức thanh toán và giúp việc quản lý chi phí trở nên đơn giản, hiệu quả hơn.

Ngoài ra, nhóm cũng có kế hoạch cải tiến giao diện và tính năng của ứng dụng để phù hợp với nhiều đối tượng người dùng, đặc biệt là người lớn tuổi và những người không quen thuộc với công nghệ. Việc này không chỉ tăng trải nghiệm người dùng mà còn mở rộng hệ thống đến với nhiều đối tượng hơn, góp phần xây dựng một nền tảng chăm sóc sức khỏe thông minh, phổ biến và đáng tin cậy.

Nhìn chung, nhóm sẽ không ngừng cải tiến và phát triển hệ thống NS Healthcare để không chỉ đáp ứng các yêu cầu hiện tại mà còn hướng đến việc xây dựng một ứng dụng chăm sóc sức khỏe toàn diện, phù hợp với xu hướng công nghệ và nhu cầu y tế trong tương lai.

**TÀI LIỆU THAM KHẢO**

[1]. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. IEEE access, 4, 2292-2303.

[2]. Gao, L., Liu, P., & Zhang, Y. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. Prompt Engineering Guide. Retrieved from https://www.promptingguide.ai/research/rag (Accessed on November 16, 2024).

[3] NestJS Documentation. (n.d.). Retrieved from https://docs.nestjs.com/ (Accessed on November 16, 2024).

[4] Adetunji, O., Ajaegbu, C., Otuneme, N., & Omotosho, O.J. (2020). Dawning of progressive web applications (PWA): Edging out the pitfalls of traditional mobile development.

[5]. Modi, R. (2018). Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain. Packt Publishing Ltd.

[6]. Chen, Y., Ding, S., Xu, Z., Zheng, H., & Yang, S. (2019). Blockchain-based medical records secure storage and medical service framework. Journal of medical systems, 43, 1-9.

[7]. Baehaqi, A., Basit, M. S., Indrajit, R. E., & Kurniawan, R. D. (2023). Front End Learning Management System Development Using the Nextjs Framework. Jurnal Teknik Informatika (Jutif), 4(4), 899-911.

[8]. Koinly. (n.d.). How to use Etherscan? Retrieved from https://koinly.io/fr/blog/how-to-use-etherscan/ (Accessed on November 16, 2024).

[9]. Hegedűs, P. (2018, May). Towards analyzing the complexity landscape of solidity based ethereum smart contracts. In Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (pp. 35-39).

[10]. Bisong, E., & Bisong, E. (2019). An overview of google cloud platform services. Building Machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners.

[11]. Brilworks. (2023, December 27). NestJS vs NextJS: When to Choose Each Framework. Brilworks. Retrieved from <https://www.brilworks.com/blog/nestjs-vs-nextjs/> (Accessed on November 16, 2024).