

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 bối cảnh phát triển không ngừng của công nghệ, Internet vạn vật (IoT) đã và đang trở thành một yếu tố không thể thiếu trong nhiều lĩnh vực, đặc biệt là y tế. Việc ứng dụng IoT trong quản lý thiết bị y tế đã mở ra những hướng đi mới, giúp cải thiện chất lượng chăm sóc sức khỏe và tối ưu hóa quy trình điều trị. Đối với Việt Nam, trong quá trình công nghiệp hóa và hiện đại hóa, việc áp dụng những công nghệ tiên tiến vào các lĩnh vực thiết yếu, đặc biệt là y tế, trở thành nền tảng quan trọng để chăm sóc sức khỏe con người, tạo ra những công dân khỏe mạnh, sẵn sàng đóng góp vào sự phát triển của đất nước.

Đồ án này của chúng em trình bày 1 hệ thống trang web quản lý dữ liệu điện tim, hỗ trợ tương tác giữa bệnh nhân và bác sĩ, giúp bệnh nhân dễ dàng kiểm soát tình trạng sức khỏe tại nhà và kết nối trực tiếp với đội ngũ y bác sĩ, mang lại sự tư vấn chuyên nghiệp và kịp thời. Với giao diện thân thiện, bệnh nhân, bác sĩ có thể đặt lịch hẹn, xem lịch sử bệnh án, theo dõi dữ liệu điện tim, tương tác trực tiếp với nhau thông qua hệ thống.

Trong quá trình thực hiện đồ án, chúng em đã nhận được sự hướng dẫn và hỗ trợ tận tình từ các thầy cô và các anh/chị/bạn trong các phòng thí nghiệm thuộc trường Điện - Điện tử. Trước tiên, chúng em xin gửi lời cảm ơn chân thành đến TS. Hàn Huy Dũng, những người đã trực tiếp hướng dẫn và chỉ ra những điểm cần khắc phục trong quy trình thực hiện đồ án cũng như thiết kế hệ thống của chúng em. Ngoài ra, chúng em cũng rất may mắn khi được kết hợp và làm việc cùng các anh/chị/bạn ở nhóm firmware SPARC Lab, nhờ đó mà chúng em đã học hỏi được rất nhiều.

Mặc dù đã cố gắng kiểm tra kỹ lưỡng, nhưng đồ án của chúng em chắc chắn không thể tránh khỏi những thiếu sót và hạn chế. Chúng em rất mong nhận được ý kiến đóng góp từ các thầy cô và bạn đọc để hoàn thiện và phát triển đề tài hơn nữa.

Chúng em xin chân thành cảm ơn!

LỜI CAM ĐOAN

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

Chúng em xin cam đoan 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 quá trình tìm hiểu, nghiên cứu và làm việc của chúng em. Các dữ liệu được nêu trong đồ án là hoàn toàn trung thực. Mọi thông tin trích dẫn đều tuân thủ các quy định về sở hữu trí tuệ; các tài liệu tham khảo được liệt kê rõ ràng. Chúng em xin chịu hoàn toàn trách nhiệm với những nội dung được viết 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	Tìm hiểu đề tài, đề xuất hệ thống	Dũng, Dương
2	Phân tích hệ thống	Dũng, Dương
3	Thiết kế cơ sở dữ liệu	Dũng, Dương
4	Thiết kế API	Dũng, Dương
5	Xây dựng phía back-end	Dũng, Dương
6	Thiết kế website quản trị	Dũng, Dương
7	Kiểm thử	Dũng, Dương
8	Viết quyển đồ án	Dũng, Dương

MỤC LỤC

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT	i
DANH MỤC KÝ HÌNH VẼ	ii
DANH MỤC BẢNG BIỂU	v
TÓM TẮT ĐỒ ÁN	vi
PHẦN MỞ ĐẦU	viii
CHƯƠNG 1. THU THẬP YÊU CẦU	1
1.1 Yêu cầu hệ thống	1
1.1.1 Yêu cầu về người dùng hệ thống	1
1.1.2 Yêu cầu chức năng	1
1.1.3 Yêu cầu phi chức năng	3
CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG	4
2.1 Thẻ CRC (Class - Responsibility - Collaboration Card)	5
2.1.1 Thẻ CRC lớp account	5
2.1.2 Thẻ CRC lớp user role	5
2.1.3 Thẻ CRC lớp user status	5
2.1.4 Thẻ CRC lớp users	5
2.1.5 Thẻ CRC lớp device type	5
2.1.6 Thẻ CRC lớp device status	5
2.1.7 Thẻ CRC lớp devices	5
2.1.8 Thẻ CRC lớp device details	5
2.1.9 Thẻ CRC lớp records	5
2.1.10 Thẻ CRC lớp record diagnosis	5
2.1.11 Thẻ CRC lớp notification schedule	5

2.1.12	Thẻ CRC lớp schedule type	5
2.1.13	Thẻ CRC lớp schedule status	5
2.1.14	Thẻ CRC lớp schedules	5
2.1.15	Thẻ CRC lớp diagnosis	5
2.1.16	Thẻ CRC lớp device type	5
2.1.17	Thẻ CRC lớp consultation schedule	5
2.2	Sơ đồ lớp	7
2.3	Sơ đồ tuần tự	7
2.3.1	Sơ đồ tuần tự đăng ký tài khoản	7
2.3.2	Sơ đồ tuần tự đăng nhập tài khoản	7
2.3.3	Sơ đồ tuần tự lấy tất cả người dùng	7
2.3.4	Sơ đồ tuần tự lấy người dùng theo id	7
2.3.5	Sơ đồ tuần tự cập nhật người dùng theo id	7
2.3.6	Sơ đồ tuần tự lấy bệnh nhân theo id bác sĩ	7
2.3.7	Sơ đồ tuần tự lấy bác sĩ theo id bệnh nhân	7
2.3.8	Sơ đồ tuần tự lấy tất cả bác sĩ	7
2.3.9	Sơ đồ tuần tự xóa người dùng theo id	7
2.4	Phân tích dữ liệu	7
2.5	Kết luận	8
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG		9
3.1	Sơ đồ kiến trúc tổng quan của hệ thống	9
3.2	Sơ đồ khối phần mềm	11
3.2.1	Website dành cho bệnh nhân	11
3.2.2	Website dành cho bác sĩ	12
3.2.3	Website cho quản trị viên	13
3.3	Thiết kế cơ sở dữ liệu	13
3.3.1	Chuẩn hoá 3NF	13
3.3.2	Từ điển dữ liệu	19
3.3.3	Sơ đồ ERD	23

3.4	Thiết kế giao diện	23
3.5	Thiết kế các chức năng cho website và server	23
3.5.1	Thiết kế API	23
3.5.2	Sơ đồ tuần tự API	23
3.6	Kết luận chương	23
CHƯƠNG 4. TRIỂN KHAI VÀ KIỂM THỬ		24
4.1	Công nghệ sử dụng	24
4.1.1	Thiết kế giao diện website	24
4.1.2	Server	24
4.2	Triển khai ứng dụng	26
4.2.1	Kiến trúc Microservices	26
4.2.2	Triển khai Server và ứng dụng web trên máy chủ VPS	26
4.3	Kiểm thử	26
4.3.1	Kiểm thử hoạt động của các API	26
4.3.2	Kiểm thử ứng dụng web	50
4.4	Kết luận chương	52
KẾT LUẬN		53
Kết luận chung		53
Hướng phát triển		53
Kiến nghị và đề xuất		53
PHỤ LỤC		55

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt	Thay cho
API	Application Programming Interface
JWT	JSON Web Token
ECG	Electrocardiogram
HTML	Hypertext Markup Language
ID	Identification
3NF	Third Normal Form
ERD	Entity-Relationship Diagram
AWS	Amazon Web Services
IOT	Internet Of Things
UML	Unified Modeling Language
PDA	Patient Doctor Assignment
CRC	Class - Responsibility - Collaboration
AI	Artificial Intelligence
GPT	Generative Pre-trained Transformer

DANH MỤC HÌNH VẼ

Hình 3.1	Kiến trúc tổng quan hệ thống	9
Hình 3.2	Sơ đồ khối Website dành cho bệnh nhân	11
Hình 3.3	Sơ đồ khối Website dành cho bác sĩ	12
Hình 3.4	Sơ đồ khối Website dành cho quản trị viên	13
Hình 4.1	Kiến trúc của NodeJS	24
Hình 4.2	Docker	26

DANH MỤC BẢNG BIỂU

Bảng 2.1	Thẻ CRC lớp Người dùng	5
Bảng 2.2	Thẻ CRC lớp Tập đính kèm	6
Bảng 3.1	Bảng chuẩn hoá bảng Người dùng	14
Bảng 3.2	Bảng chuẩn hoá bảng Tài khoản	14
Bảng 3.3	Bảng chuẩn hoá bảng Token đăng nhập	14
Bảng 3.4	Bảng chuẩn hoá bảng Thiết bị	15
Bảng 3.5	Bảng chuẩn hoá bảng Thông số thiết bị	15
Bảng 3.6	Bảng chuẩn hoá bảng Bản ghi dữ liệu	16
Bảng 3.7	Bảng chuẩn hoá bảng Thông tin hội thoại	16
Bảng 3.8	Bảng chuẩn hoá bảng Thành viên tham gia hội thoại	16
Bảng 3.9	Bảng chuẩn hoá bảng Tin nhắn	17
Bảng 3.10	Bảng chuẩn hoá bảng Tập đính kèm	17
Bảng 3.11	Bảng chuẩn hoá bảng Phân công bệnh nhân - bác sĩ	18
Bảng 3.12	Bảng chuẩn hoá bảng Tài khoản phê duyệt	18
Bảng 3.13	Bảng user	19
Bảng 3.14	Bảng account	19
Bảng 3.15	Bảng register	19
Bảng 3.16	Bảng record	20
Bảng 3.17	Bảng patient_doctor_assignment	20
Bảng 3.18	Bảng token	20
Bảng 3.19	Bảng device	21
Bảng 3.20	Bảng device_detail	21
Bảng 3.21	Bảng conversation	21
Bảng 3.22	Bảng message	22
Bảng 3.23	Bảng conversation_member	22
Bảng 3.24	Bảng conversation_attachment	22
Bảng 4.1	Bảng kiểm thử API đăng ký tài khoản	28

Bảng 4.2	Bảng kiểm thử API đăng nhập	28
Bảng 4.3	Bảng kiểm thử API đăng xuất	29
Bảng 4.4	Bảng kiểm thử API gửi token đặt lại mật khẩu	30
Bảng 4.5	Bảng kiểm thử API đặt lại mật khẩu	30
Bảng 4.6	Bảng kiểm thử API đăng ký tài khoản	32
Bảng 4.7	Bảng kiểm thử API chấp nhận tài khoản	32
Bảng 4.8	Bảng kiểm thử API từ chối phê duyệt tài khoản	33
Bảng 4.9	Bảng kiểm thử API lấy danh sách người dùng	33
Bảng 4.10	Bảng kiểm thử API lấy thông tin của người dùng thông qua ID . . .	34
Bảng 4.11	Bảng kiểm thử API lấy thông tin của người dùng thông qua chức vụ	35
Bảng 4.12	Bảng kiểm thử API cập nhật thông tin người dùng	35
Bảng 4.13	Bảng kiểm thử API xóa thông tin người dùng	36
Bảng 4.14	Bảng kiểm thử API lấy danh sách thiết bị	36
Bảng 4.15	Bảng kiểm thử API lấy thông tin thiết bị theo ID	37
Bảng 4.16	Bảng kiểm thử API thêm thiết bị	37
Bảng 4.17	Bảng kiểm thử API cập nhật thông tin thiết bị	38
Bảng 4.18	Bảng kiểm thử API xóa thiết bị	40
Bảng 4.19	Bảng kiểm thử API lấy danh sách bản ghi	41
Bảng 4.20	Bảng kiểm thử API lấy danh sách phiên đo ECG của bệnh nhân . .	41
Bảng 4.21	Bảng kiểm thử API lấy danh sách phiên đo ECG mà bác sĩ phụ trách	42
Bảng 4.22	Bảng kiểm thử API lấy thông tin một phiên đo ECG	42
Bảng 4.23	Bảng kiểm thử API lấy dữ liệu bản ghi một phiên đo ECG	43
Bảng 4.24	Bảng kiểm thử API tải dữ liệu đo ECG	43
Bảng 4.25	Bảng kiểm thử API cập nhật thông tin phiên đo	44
Bảng 4.26	Bảng kiểm thử API xóa thông tin một phiên đo ECG	45
Bảng 4.27	Bảng kiểm thử API lấy danh sách phân công bác sĩ - bệnh nhân . .	46
Bảng 4.28	Bảng kiểm thử API tạo một phân công bác sĩ - bệnh nhân	47
Bảng 4.29	Bảng kiểm thử API cập nhật thông tin phân công	47
Bảng 4.30	Bảng kiểm thử API xóa thông tin phân công	48

Bảng 4.31	Bảng kiểm thử API lấy danh sách bệnh nhân mà bác sĩ đang quản lý theo ID bác sĩ	49
Bảng 4.32	Bảng kiểm thử API lấy thông tin bác sĩ đang quản lý bệnh nhân theo ID bệnh nhân	49
Bảng 4.33	Bảng kiểm thử chức năng của website quản trị	50

TÓM TẮT ĐỒ ÁN

Đồ án "Hệ thống quản lý dữ liệu điện tim và tương tác giữa bệnh nhân - bác sĩ" là hệ thống bao gồm Web/App/Server giúp việc đặt lịch và tương tác giữa các bác sĩ, bệnh nhân trên website và ứng dụng di động một cách dễ dàng và hiệu quả. Tại đây thông tin người dùng, dữ liệu đặt lịch, dữ liệu đo hay các đoạn hội thoại sẽ được lưu trữ trên server để người dùng có thể xem lại bất cứ lúc nào. Bên cạnh đó hệ thống cung cấp tính năng đặt lịch hẹn giúp bệnh nhân thoải mái chọn lịch phù hợp với thời gian của mình. Bác sĩ sẽ đồng ý hoặc từ chối lịch, sau đó thông báo sẽ được gửi đến bệnh nhân. Nếu bác sĩ quên, sẽ có mail thông báo đến bác sĩ. Sau khi bác sĩ đồng ý lịch hẹn, bệnh nhân và bác sĩ có thể nhắn tin để trao đổi thông tin.

Đồ án tập trung hoàn thiện theo quy trình phát triển phần mềm, sau khi xác định được các yêu cầu của hệ thống sẽ tiến hành phân tích, thiết kế và triển khai hệ thống. Quá trình này áp dụng phương pháp phân tích và thiết kế hướng đối tượng, đồng thời sử dụng ngôn ngữ UML để biểu diễn các luồng thực hiện hành động. Ứng dụng Web sử dụng ReactJS, và server là NodeJS, framework NestJS. Cơ sở dữ liệu sử dụng là MySQL và MongoDB.

Quyển đồ án được trình bày theo quy trình phát triển phần mềm, nội dung mỗi chương được triển khai và trình bày thông qua sơ đồ và diễn giải chi tiết. Các chương theo thứ tự lần lượt là phân tích hệ thống, thiết kế hệ thống, triển khai và kiểm thử, cuối cùng là kết luận.

ABSTRACT

The project "Electrocardiogram Data Management System and Patient-Doctor Interaction" is a system comprising Web/App/Server components that facilitate scheduling and interaction between doctors and patients through websites and mobile applications in an easy and efficient manner.

Here, user information, appointment schedules, measurement data, and conversations are stored on the server, allowing users to review them at any time. Additionally, the system offers a scheduling feature that enables patients to conveniently select a suitable appointment time. Doctors can approve or decline the schedule, and a notification will then be sent to the patient. If the doctor forgets, a reminder email will be sent to the doctor. Once the doctor approves the appointment, patients and doctors can chat to exchange information.

The project focuses on completing the software development process. After identifying the system requirements, the analysis, design, and implementation phases are carried out. This process applies object-oriented analysis and design methods, while utilizing UML language to represent action execution flows. The web application uses ReactJS, the server is built with NodeJS and the NestJS framework, and the databases used are MySQL and MongoDB.

The project report is presented according to the software development process, with each chapter developed and detailed through diagrams and comprehensive explanations. The chapters are ordered as follows: system analysis, system design, implementation and testing, and finally, conclusion.

PHẦN MỞ ĐẦU

Đặt vấn đề

Kể từ sau đại dịch Covid-19 khiến sức khỏe trở thành mối quan tâm hàng đầu của người dân Việt Nam. Kèm theo đó, con người đang tiếp xúc với môi trường sống ngày càng ô nhiễm, tiếp xúc với các thiết bị điện tử ngày càng nhiều, không vận động thể thao; một cơ sở người đã bắt đầu để ý đến những tín hiệu từ cơ thể như tim mạch, huyết áp, nhiệt độ cơ thể. Đặc biệt, việc chăm sóc sức khỏe tại nhà ngày càng trở nên quan trọng và cần thiết. Đối với những người có nhu cầu tự theo dõi sức khỏe, các thiết bị đo lường nhỏ gọn và ứng dụng di động hỗ trợ đã trở thành công cụ thiết yếu. Đối với những người đang gặp vấn đề về sức khỏe, việc phải đến các bệnh viện đông đúc để kiểm tra và khám chữa bệnh là một thách thức không nhỏ. Do đó, câu hỏi đặt ra là liệu có phương án khả thi nào giúp người dùng có thể theo dõi tình trạng sức khỏe ngay tại nhà, nhưng vẫn nhận được sự tư vấn và hỗ trợ từ các chuyên gia y tế hay không?

Đề xuất hệ thống

Để ý nhu cầu chăm sóc sức khỏe tại nhà của không chỉ những người đang gặp vấn đề mà kể cả những người bình thường, đồ án của chúng em đề xuất một hệ thống IOT theo dõi và quản lý dữ liệu điện tim bao gồm: Website, Server, Mobile App, Thiết bị đo. Trong đồ án này chúng em tập trung vào Server và website.

Trong thời gian làm việc cùng các anh chị trong nhóm phần cứng, chúng em đã được tiếp cận với thiết bị đo điện tim bằng điện cực không tiếp xúc, với mong muốn xây dựng được một hệ thống IOT có thể kết nối các thiết bị đo điện tim, thu thập dữ liệu điện tim theo thời gian thực, lưu trữ dữ liệu hợp lý để phục vụ cho mục đích phân tích dữ liệu sau này, đồng thời tối ưu hóa quá trình lấy dữ liệu từ người dùng. Cụ thể, hệ thống sẽ bao gồm:

- Một website cho bệnh nhân có thể đặt lịch và nhắn tin với bác sĩ của mình
- Một website cho bác sĩ để có thể phản hồi lịch hẹn, xem được kết quả đo của các bệnh nhân được quản lý, nhắn tin với bệnh nhân, nhóm chat
- Một website cho admin để quản lý hệ thống, quản lý người dùng, quản lý dữ liệu
- 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:

- Nắm được cơ sở lý thuyết và cách thiết kế một hệ thống phần mềm.
- Thực hiện hoàn chỉnh Website và Server được đề ra trong mục Đề xuất hệ thống, các ứng dụng hoạt động ổn định
- Có thể kết hợp tốt với các thiết bị phần cứng đang được hợp tác nghiên cứu
- Cung cấp tài liệu tham khảo một cách đầy đủ, trung thực

Phương pháp nghiên cứu

Trong đồ án lần này, chúng em đã thực hiện kết hợp các phương pháp nghiên cứu, đầu tiên là tham khảo thông tin các bài báo, sản phẩm về thiết bị điện tử trong các phòng nghiên cứu tại trường, cùng với đó, tìm hiểu cách các hệ thống phần cứng và phần mềm hoạt động, kết nối với nhau. Sau khi đã nắm được cơ sở lý thuyết, chúng em tiến hành các bài thực nghiệm, lưu trữ dữ liệu các bản ghi đo đạc, thử nghiệm vẽ các biểu đồ thể hiện chỉ số được lấy từ các bản ghi kết hợp với nhóm firmware để phân tích số liệu nhằm chắc chắn dữ liệu được truyền đúng, kết nối với những chuyên gia có chuyên môn về lĩnh vực y tế, đặc biệt là lĩnh vực tim mạch để chắc chắn đồ thị đã biểu diễn được đúng dữ liệu đo được.

Kết quả đạt được

Trong suốt quá trình thực hiện đồ án, hai chúng em Trần Minh Tuấn, Phạm Quang Huy đã được tìm hiểu và nghiên cứu sâu hơn về cả phần cứng, hệ thống IOT, cách kết nối các hệ thống với nhau. Các kết quả đạt được cho đến thời điểm hoàn thiện quyển đồ án bao gồm:

- Hoàn thành quyển đồ án với nội dung chi tiết về quá trình xây dựng và phát triển hệ thống
- Hoàn thành các sản phẩm ứng dụng đã đề ra trong mục Đề xuất hệ thống, các sản phẩm đã có sự kết nối, dữ liệu điện tử được theo dõi và lưu trên server, dữ liệu có thể được phân tích và nghiên cứu sau này
- Được phát triển các kỹ năng làm việc nhóm, viết đồ án, kết hợp với nhóm phần cứng, nhóm firmware, các chuyên gia trong lĩnh vực y tế để các sản phẩm được hoàn thiện hơn

Cấu trúc đồ án

- Phần mở đầu: Trình bày về mục đích của đồ án, đề xuất hệ thống, phân tích tính khả thi và bố cục đồ án
- Chương 1: Trình bày chi tiết các khâu thu thập yêu cầu. Bao gồm kỹ thuật thu thập, xác định yêu cầu hệ thống, thiết kế sơ đồ use case

- Chương 2: Trình bày chi tiết các khâu trong phân tích hệ thống. Bao gồm mô tả thể CRC, thiết kế sơ đồ lớp, sơ đồ tuần tự
- Chương 3: Trình bày chi tiết khâu thiết kế cho hệ thống. Bao gồm thiết kế sơ đồ kiến trúc hệ thống, sơ đồ khối phần mềm, thiết kế cơ sở dữ liệu, thiết kế giao diện, sơ đồ lớp và thiết kế chức năng cho hệ thống
- Chương 4: Trình bày khâu triển khai và kiểm thử
- Phần kết luận: Kết luận và đưa ra hướng phát triển

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 Yêu cầu hệ thống

1.1.1 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 truy cập vào tài khoản của mình, có thể tìm kiếm và chọn lịch, chọn bác sĩ phù hợp để đăng ký khám hoặc tư vấn thiết bị, nhắn tin với bác sĩ của mình.
- Bác sĩ: sử dụng hệ thống để thực hiện theo dõi các bệnh nhân có trong hệ thống. Bác sĩ có quyền truy cập vào kết quả ECG của bệnh nhân, có thể trao đổi, tư vấn với bệnh nhân về các thông tin liên quan, đặt lịch tái khám cho bệnh nhân. Nhắn tin với bệnh nhân của mình và các nhóm chat để trao đổi thông tin.
- Quản trị viên: sử dụng hệ thống để quản lý các tài khoản người dùng, thiết bị, quản lý lịch của bác sĩ và bệnh nhân, quản lý các bản dữ liệu đo.

1.1.2 Yêu cầu chức năng

Các chức năng chính của hệ thống bao gồm:

- Ghi dữ liệu đo của thiết bị: App trên điện thoại ghi dữ liệu đo được từ thiết bị đo điện tim thông qua Bluetooth. Dữ liệu đo được ghi lại dưới dạng bảng biểu theo các tham số và được lưu trữ trên máy chủ của hệ thống.
- Hiển thị dữ liệu: Các số liệu được lưu trữ trên máy chủ hệ thống được tính toán theo công thức để hiển thị lên trên web, app dưới dạng đồ thị đường.
- Lưu trữ: Hệ thống hỗ trợ lưu dữ liệu mà người dùng đo được từ thiết bị trên cả ứng dụng và trên server của hệ thống. Dữ liệu điện tim cũng được đồng bộ hóa và lưu trữ trên máy chủ của hệ thống. Qua quá trình đồng bộ hóa, dữ liệu từ ứng dụng được truyền đến máy chủ và được lưu trữ an toàn và bảo mật trên hệ thống. Việc lưu trữ dữ liệu trên cả ứng dụng và máy chủ giúp đảm bảo rằng dữ liệu quan trọng này được lưu trữ một cách đáng tin cậy và có sẵn cho phân tích hoặc sử dụng tương lai.

- 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 và trợ lý ảo, chia sẻ kết quả đo điện tim, hỏi đáp về các vấn đề sức khỏe, các vấn đề liên quan đến hệ thống và thiết bị hoặc thảo luận về các quyết định. Điều này mang lại sự tiện lợi và hỗ trợ đáng kể cho người dùng trong việc xác định về tình trạng sức khỏe hiện tại của bản thân.

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 ECG 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
- Nhận tư vấn từ trợ lý ảo về hệ thống và các thông tin liên quan về sức khỏe.

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 bệnh nhân
- Quản lý danh sách các bản ghi dữ liệu đo của các bệnh nhân trong list quản lý của mình
- 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
- Tìm thêm thông tin về y tế và hệ thống này thông qua tương tác với trợ lý ảo

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, bản ghi của các dữ liệu
- Quản lý phân công bác sĩ - bệnh nhân
- Quản lý phê duyệt tài khoản khi người dùng đăng ký trên hệ thống

1.1.3 Yêu cầu phi chức năng

- 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

Quá trình phân tích yêu cầu hệ thống giúp xác định rõ ràng các chức năng, yêu cầu phi chức năng và đối tượng người dùng mà hệ thống cần phục vụ. Dựa trên nền tảng này, việc thiết kế và phát triển hệ thống quản lý dữ liệu điện tim sẽ được tiến hành, đảm bảo đáp ứng đầy đủ mọi yêu cầu của người dùng, đồng thời tối ưu hóa hiệu suất, bảo mật và tính khả dụng của hệ thống.

CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG

Chương này sẽ trình bày chi tiết về quá trình phân tích hệ thống, dựa trên các yêu cầu đã được nêu trong Phần mở đầu và Chương 1. Quá trình này bao gồm các bước sau:

- Thiết kế các thẻ CRC (Class - Responsibility - Collaboration Card) cùng các lớp dựa trên thông tin từ các sơ đồ use case chức năng của hệ thống
- Xây dựng sơ đồ lớp dựa trên các lớp đã được định nghĩa với thuộc tính và phương thức
- Sau khi đã xác định về phương thức cũng như thuộc tính các lớp cùng chức năng, bước tiếp theo sẽ thiết kế các sơ đồ tuần tự minh họa sự tương tác giữa các lớp khi thực hiện chức năng cụ thể

2.1 Thẻ CRC (Class - Responsibility - Collaboration Card)

2.1.1 Thẻ CRC lớp account

2.1.2 Thẻ CRC lớp user role

2.1.3 Thẻ CRC lớp user status

2.1.4 Thẻ CRC lớp users

2.1.5 Thẻ CRC lớp device type

2.1.6 Thẻ CRC lớp device status

2.1.7 Thẻ CRC lớp devices

2.1.8 Thẻ CRC lớp device details

2.1.9 Thẻ CRC lớp records

2.1.10 Thẻ CRC lớp record diagnosis

2.1.11 Thẻ CRC lớp notification schedule

2.1.12 Thẻ CRC lớp schedule type

2.1.13 Thẻ CRC lớp schedule status

2.1.14 Thẻ CRC lớp schedules

2.1.15 Thẻ CRC lớp diagnosis

2.1.16 Thẻ CRC lớp device type

2.1.17 Thẻ CRC lớp consultation schedule

Bảng 2.1 Thẻ CRC lớp Người dùng

Mặt trước thẻ

Tên lớp: Người dùng (User)	ID: 1
Mô tả: Đối tượng mô tả thông tin người dùng	Use case liên quan: Đăng ký tài khoản, Quản lý tài khoản cá nhân, Quản lý phân công bác sĩ - bệnh nhân
Trách nhiệm (Responsibility): Thêm, sửa, xóa thông tin người dùng Lấy danh sách người dùng Lấy thông tin người dùng qua ID, tài khoản, chức vụ	Các lớp cộng tác (Collaboration): Tài khoản

Mặt sau thẻ

Thuộc tính (Attributes): id(uuid) username(String) birth(Timestamps) phone_number(String)	image(String) role(Integer) created_at(Timestamps) updated_at(Timestamps)
Mối quan hệ (Relationships) Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Tài khoản, Phân công bác sĩ - bệnh nhân	

Bảng 2.2 Thẻ CRC lớp Tập đính kèm

Mặt trước thẻ

Tên lớp: Tập đính kèm	ID: 14
Mô tả: Đối tượng mô tả thông tin Tập đính kèm tin nhắn	Use case liên quan: Quản lý dịch vụ nhắn tin
Trách nhiệm (Responsibility): Mô tả thông tin Tập đính kèm Lấy thông tin tập đính kèm theo tin nhắn, cuộc hội thoại	Các lớp cộng tác (Collaboration): Tin nhắn, Cuộc hội thoại

Mặt sau thẻ

Thuộc tính (Attributes): id(String) content_url(String) file_name(String)	size(Integer) thumbnail_url(String) type(Integer)
---	---

Mối quan hệ (Relationships):

Tổng quát hóa (Generalize):

Toàn thể - Bộ phận (Aggregation):

Liên kết (Association): Tin nhắn, Cuộc hội thoại

2.2 Sơ đồ lớp

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

2.3 Sơ đồ tuần tự

Để phân tích cụ thể hơn về từng luồng trong hệ thống thông qua sơ đồ use case và sơ đồ hoạt động, dưới đây là phần thiết kế về các sơ đồ tuần tự.

2.3.1 Sơ đồ tuần tự đăng ký tài khoản

Sơ đồ tuần tự mô tả chi tiết quá trình người dùng đăng ký một tài khoản mới trên hệ thống. Người dùng sẽ phải gửi yêu cầu đăng ký, hệ thống sẽ được xử lý bởi lớp Register, nếu có lỗi phát sinh hệ thống sẽ thông báo lỗi cho người dùng. Nếu việc đăng ký thành công, lớp Register sẽ gửi thông báo chuyển sang phê duyệt tài khoản và trả về kết quả.

2.3.2 Sơ đồ tuần tự đăng nhập tài khoản

2.3.3 Sơ đồ tuần tự lấy tất cả người dùng

2.3.4 Sơ đồ tuần tự lấy người dùng theo id

2.3.5 Sơ đồ tuần tự cập nhật người dùng theo id

2.3.6 Sơ đồ tuần tự lấy bệnh nhân theo id bác sĩ

2.3.7 Sơ đồ tuần tự lấy bác sĩ theo id bệnh nhân

2.3.8 Sơ đồ tuần tự lấy tất cả bác sĩ

2.3.9 Sơ đồ tuần tự xóa người dùng theo id

2.4 Phân tích dữ liệu

Tại phần này, chúng em sẽ tiến hành xác định và mô tả các thực thể cũng như thuộc tính trong hệ thống. Việc này giúp chúng em có thể nắm được các phần chính trong việc thiết kế nên cơ sở dữ liệu.

Đầu tiên, chúng em sẽ xác định các thực thể và mô tả thuộc tính của nó trong hệ thống dưới dạng bảng và sơ đồ mô hình liên kết.

Sau khi hoàn thành được bảng thực thể và thuộc tính, chúng em xác định được mô hình thực thể liên kết như sau:

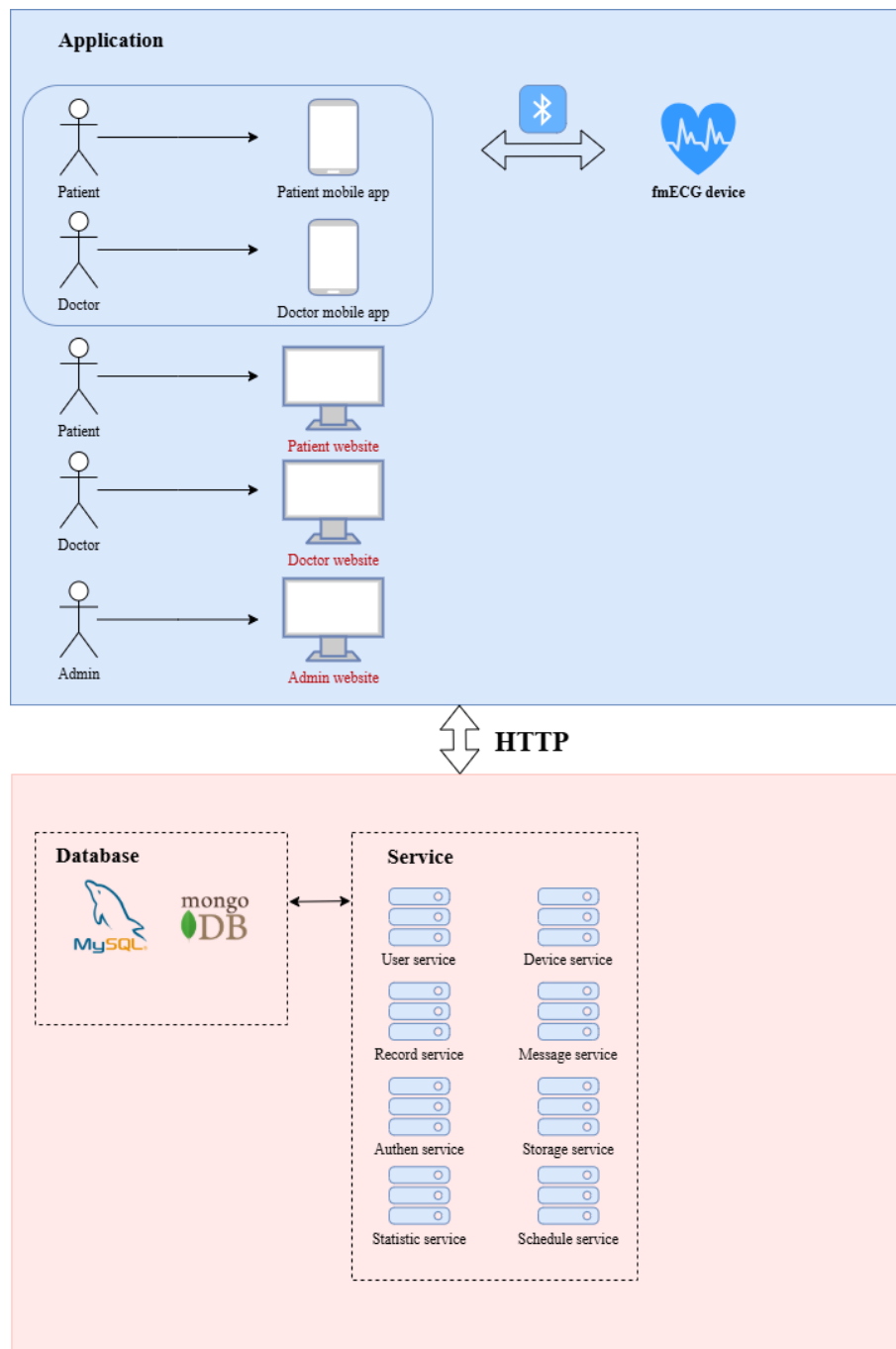
2.5 Kết luận

Chương này thực hiện phân tích khái quát về hệ thống, nhằm đáp ứng các yêu cầu và mục tiêu đã được đề ra trong các phần trên.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

Trong chương này, chúng em sẽ trình bày về thiết kế hệ thống từ tổng quan đến chi tiết dựa trên phân tích từ Chương 2. Đầu tiên là xây dựng sơ đồ kiến trúc hệ thống. Sau đó thiết kế giao diện, chức năng cho website và server. Nội dung chính của chương tập trung vào các hình ảnh và sơ đồ mô tả từng luồng hoạt động của hệ thống.

3.1 Sơ đồ kiến trúc tổng quan của hệ thống



Hình 3.1 Kiến trúc tổng quan hệ thống

Hệ thống được chia làm ba phần chính bao gồm Device (Thiết bị), Server (Máy chủ) và Application (Ứng dụng bao gồm: web và app). Mỗi thành phần đóng vai trò quan trọng trong việc vận hành tổng thể hệ thống trong hình vẽ

Hình 3.1 thể hiện ba phần:

- Device (Thiết bị): Bao gồm thiết bị phần cứng đo điện tim, có thể kết nối với ứng dụng di động của bệnh nhân thông qua Bluetooth
- Application (Ứng dụng): Bao gồm ứng dụng di động cho bệnh nhân, bác sĩ; website của quản trị viên, website của bác sĩ, website của bệnh nhân
- Server (Máy chủ): Bao gồm các Services để xử lý các yêu cầu gửi từ Application, cơ sở dữ liệu.

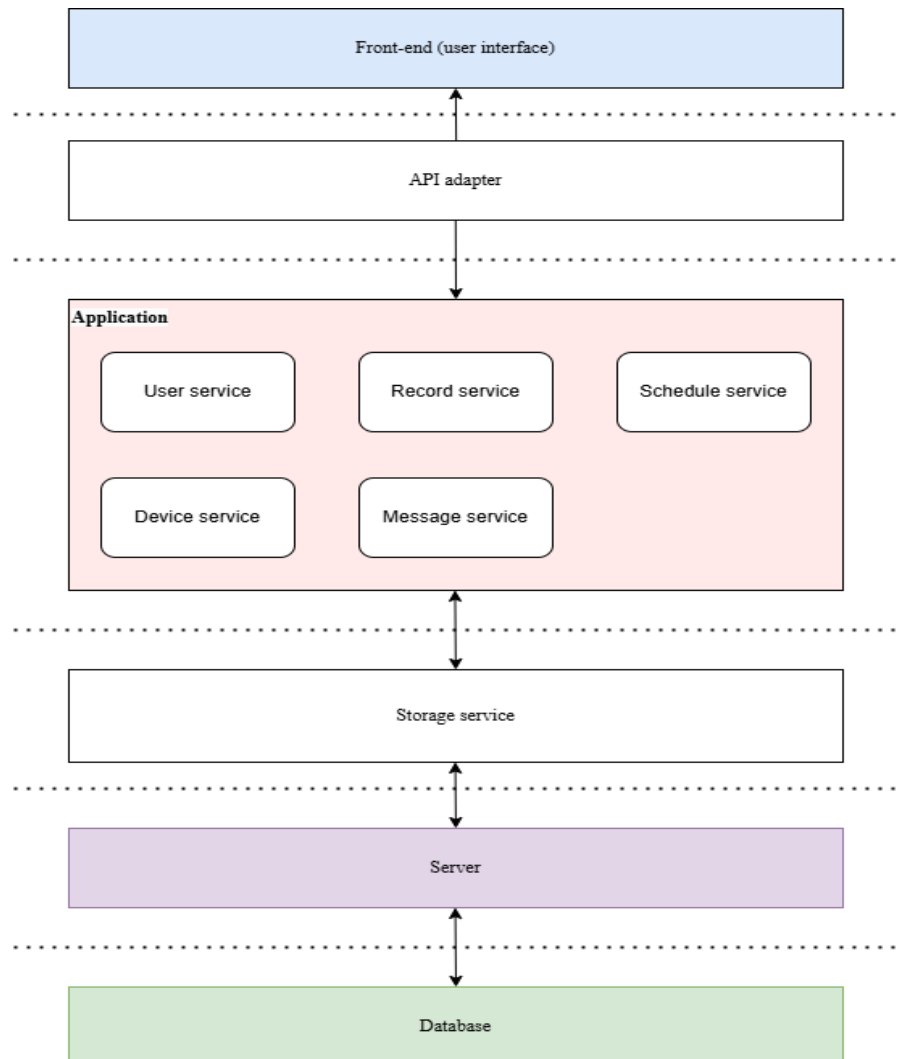
Hệ thống được thiết kế với sự tương tác trực tiếp giữa bệnh nhân và thiết bị thông qua ứng dụng di động. Ứng dụng đóng vai trò trung gian, giao tiếp với máy chủ bằng API và giao thức HTTP. Máy chủ chịu trách nhiệm xử lý dữ liệu thông qua các dịch vụ chuyên biệt. Mỗi dịch vụ được thiết kế để xử lý một loại yêu cầu cụ thể, bao gồm cả việc truy vấn và cập nhật cơ sở dữ liệu, trước khi trả kết quả về cho người dùng.

- User service: Xử lý các yêu cầu liên quan đến người dùng như đăng ký, đăng nhập, lấy thông tin cá nhân
- Device service: Xử lý các tác vụ liên quan tới thiết bị như thêm, sửa, xóa thiết bị, cập nhật thông tin.
- Storage service: Xử lý các tác vụ liên quan tới lưu trữ dữ liệu của hệ thống.
- Record service: Xử lý các tác vụ liên quan tới bản ghi như là thêm sửa xóa, xử lý dữ liệu file đo, vẽ đồ thị
- Message service: Xử lý các yêu cầu liên quan tới tin nhắn.
- Authen service: Xử lý các tác vụ liên quan tới bảo mật như: mã hoá và cung cấp token, kiểm tra token, phân quyền truy cập api, mã hoá các dữ liệu nhạy cảm trước khi lưu trữ.
- Statistic service: Xử lý các tác vụ liên quan tới thống kê như: thống kê người dùng, thiết bị, bản ghi.
- Schedule service: Xử lý các tác vụ liên quan đến đặt lịch hẹn, phản hồi lịch.

Sau đây là mô tả chi tiết về từng khối nhỏ hơn trong kiến trúc hệ thống, dựa trên các đối tượng đã được xác định trong hệ thống.

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

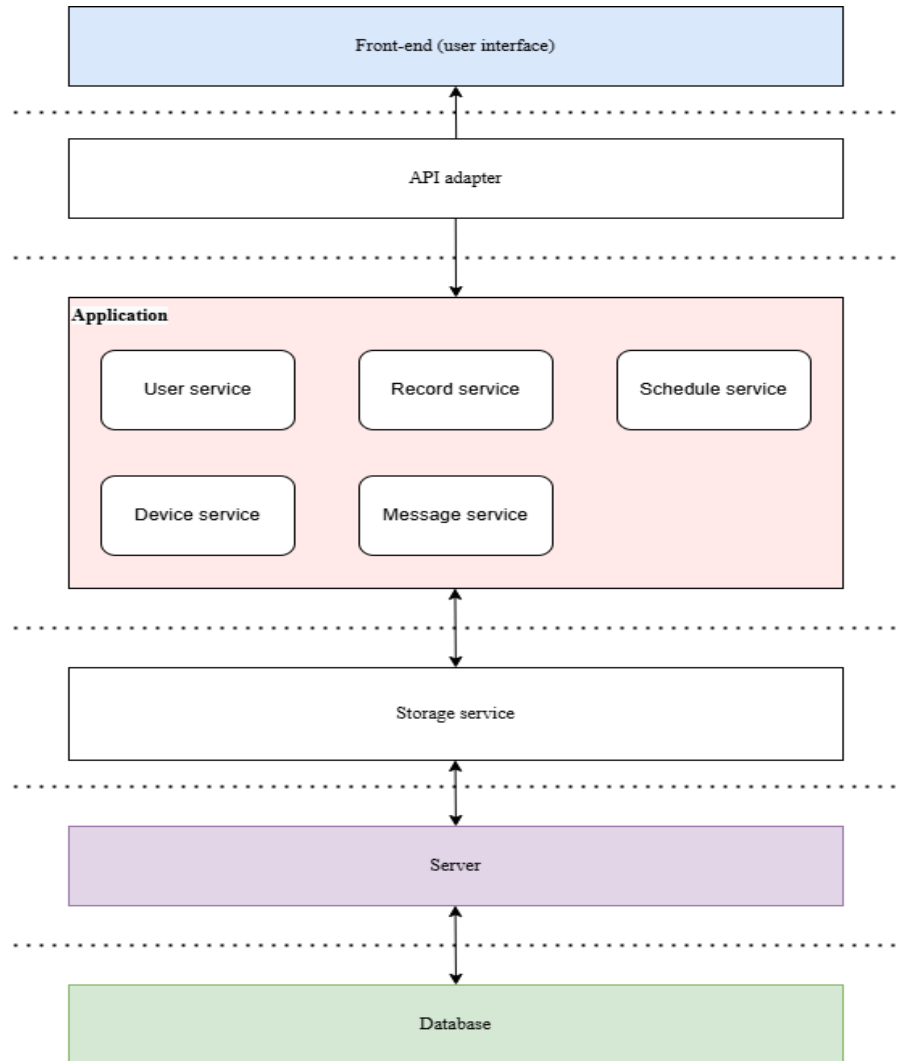
Lớp trên cùng trong sơ đồ hình 3.2 là lớp giao diện người dùng, nơi người dùng trực tiếp tương tác với hệ thống, thực hiện các request thông qua API Adapter.

Sau đó các request được gửi đến lớp Application nơi chứa các Services để xử lý các request từ người dùng. Kết quả được trả về giao diện. Ở đây các services gồm có:

- User service: Khối có nhiệm vụ xử lý thông tin cá nhân
- Device service: Khối có nhiệm vụ xử lý các vấn đề liên quan đến mượn trả thiết bị.
- Schedule service: Khối có nhiệm vụ quản lý việc đặt lịch của bệnh nhân
- Record service: Khối có nhiệm vụ xử lý thông tin bản ghi

- Message service: Khối có nhiệm vụ quản lý việc chat, trao đổi thông tin với bác sĩ
- Storage service: Khối service đặc biệt có nhiệm vụ lưu thông tin vào bộ nhớ

3.2.2 Website dành cho bác sĩ

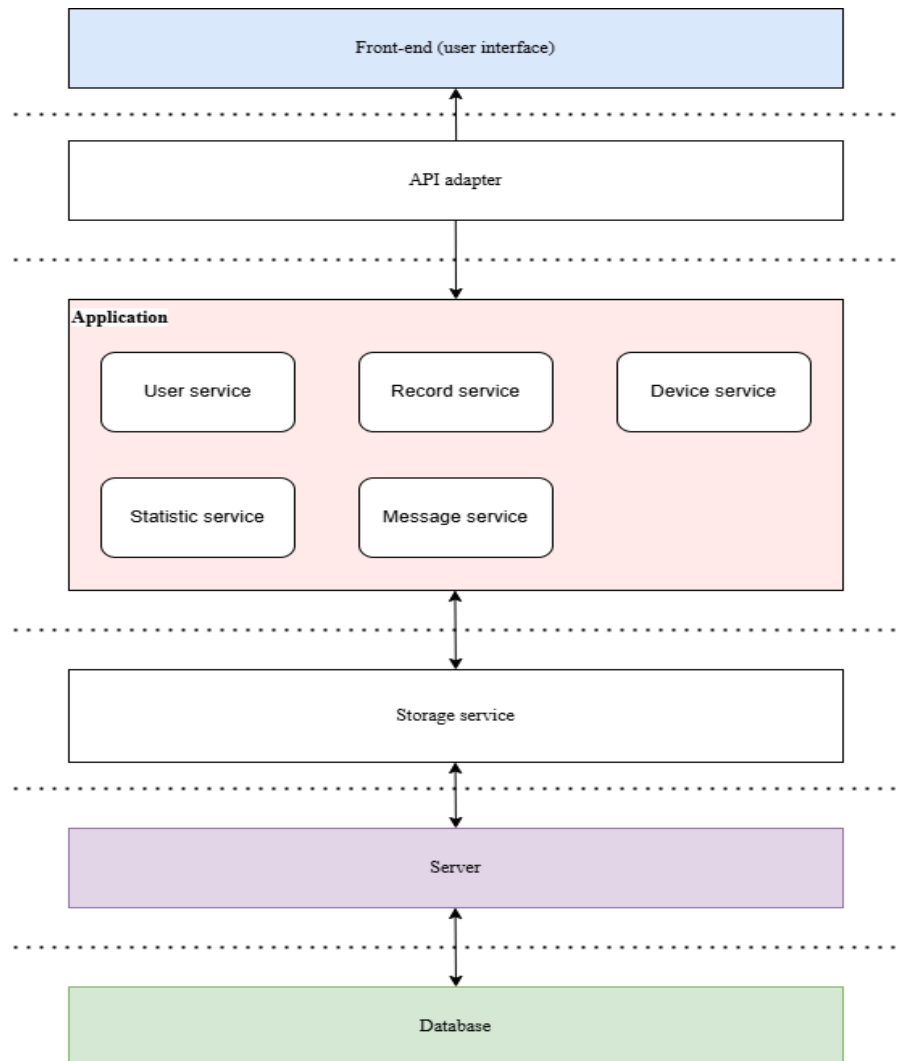


Hình 3.3 Sơ đồ khối Website dành cho bác sĩ

- User service: Khối có nhiệm vụ xử lý thông tin cá nhân
- Device service: Khối có nhiệm vụ xử lý các vấn đề liên quan đến mượn trả thiết bị.
- Schedule service: Khối có nhiệm vụ quản lý việc đặt lịch của bệnh nhân
- Record service: Khối có nhiệm vụ xử lý thông tin bản ghi
- Message service: Khối có nhiệm vụ quản lý việc chat, trao đổi thông tin với bác sĩ, trao đổi trong nhóm.

- Storage service: Khối service đặc biệt có nhiệm vụ lưu thông tin vào bộ nhớ

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 có cấu trúc tương tự như website dành cho bác sĩ và bệnh nhân. Admin sẽ có quyền quản lý toàn bộ thông tin người dùng, thiết bị, bản ghi. Bên cạnh đó, admin có quyền thống kê dữ liệu từ các services thông qua Statistic service.

3.3 Thiết kế cơ sở dữ liệu

3.3.1 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.

3.3.1.1 Chuẩn hoá bảng Người dùng

Bảng 3.1 Bảng chuẩn hoá bảng Người dùng

Danh sách thuộc tính	ID người dùng, ID tài khoản, Tên người dùng, Ngày sinh, Giới tính, Số điện thoại, Quyền, Trạng thái, Đường dẫn lưu trữ ảnh, Thông tin người dùng
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi người dùng có một ID riêng, có duy nhất ID tài khoản, tên, ngày sinh, giới tính, số điện thoại, quyền, trạng thái, đường dẫn lưu trữ ảnh, thông tin	ID người dùng → ID tài khoản, Tên người dùng, Ngày sinh, Giới tính, Số điện thoại, Quyền, Trạng thái, Đường dẫn lưu trữ ảnh, Thông tin
⇒ Khoá chính của bảng: ID người dùng ⇒ Bảng Người dùng đã ở 3NF	

3.3.1.2 Chuẩn hoá bảng Tài khoản

Bảng 3.2 Bảng chuẩn hoá bảng Tài khoản

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

3.3.1.3 Chuẩn hoá bảng Token đăng nhập

Bảng 3.3 Bảng chuẩn hoá bảng Token đăng nhập

Danh sách thuộc tính	ID token, ID tài khoản, Token truy cập, Token làm mới
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi người dùng có một ID token riêng, có duy nhất ID tài khoản, token truy cập và token làm mới	ID token → ID tài khoản, Token truy cập, Token làm mới
⇒ Khoá chính của bảng: ID token đăng nhập ⇒ Bảng Token đã ở 3NF	

3.3.1.4 Chuẩn hoá bảng Thiết bị

Bảng 3.4 Bảng chuẩn hoá bảng Thiết bị

Danh sách thuộc tính	ID thiết bị, ID người dùng thiết bị, ID bác sĩ theo dõi, Tên thiết bị, Loại thiết bị, Thông tin thiết bị, Trạng thái thiết bị, Ngày bắt đầu sử dụng
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi thiết bị khi được sử dụng sẽ có một ID thiết bị riêng, có duy nhất tên thiết bị, loại thiết bị, thông tin thiết bị. ID người dùng thiết bị, ID bác sĩ theo dõi, trạng thái thiết bị, ngày bắt đầu sử dụng	ID thiết bị → ID người dùng thiết bị, ID bác sĩ theo dõi, Tên thiết bị, Loại thiết bị, Thông tin thiết bị, Trạng thái thiết bị, Ngày bắt đầu sử dụng
⇒ Khoá chính của bảng: ID thiết bị ⇒ Bảng Thiết bị đã ở 3NF	

3.3.1.5 Chuẩn hoá bảng Thông số thiết bị

Bảng 3.5 Bảng chuẩn hoá bảng Thông số thiết bị

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

3.3.1.6 Chuẩn hoá bảng Bản ghi dữ liệu

Bảng 3.6 Bảng chuẩn hoá bảng Bản ghi dữ liệu

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

3.3.1.7 Chuẩn hoá bảng Thông tin hội thoại

Bảng 3.7 Bảng chuẩn hoá bảng Thông tin hội thoại

Danh sách thuộc tính	ID hội thoại, Tên hội thoại, Loại hội thoại, Đường dẫn avatar hội thoại
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi hội thoại có một ID hội thoại riêng, có duy nhất tên hội thoại, loại hội thoại, đường dẫn avatar hội thoại	ID hội thoại → Tên hội thoại, Loại hội thoại, Đường dẫn avatar hội thoại
⇒ Khoá chính của bảng: ID hội thoại	
⇒ Bảng Thông tin hội thoại đã ở 3NF	

3.3.1.8 Chuẩn hoá bảng Thành viên tham gia hội thoại

Bảng 3.8 Bảng chuẩn hoá bảng Thành viên tham gia hội thoại

Danh sách thuộc tính	ID, ID hội thoại, ID người dùng tham gia, Trạng thái thông báo, Tác vụ, Trạng thái đã xem
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi hội thoại có nhiều ID người dùng tham gia, mỗi người dùng có duy nhất trạng thái thông báo, tác vụ, trạng thái đã xem trong một hội thoại	ID → ID hội thoại, ID người dùng tham gia, Trạng thái thông báo, Tác vụ, Trạng thái đã xem
⇒ Khoá chính của bảng: ID	
⇒ Bảng Thành viên tham gia hội thoại đã ở 3NF	

3.3.1.9 Chuẩn hoá bảng Tin nhắn

Bảng 3.9 Bảng chuẩn hoá bảng Tin nhắn

Danh sách thuộc tính	ID tin nhắn, ID hội thoại, ID người gửi, Các tệp đính kèm, Tin nhắn hệ thống, Ghim, Thời gian ghim tin nhắn, Các lượt thả cảm xúc
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi tin nhắn có một ID tin nhắn riêng, thuộc một hội thoại, có duy nhất ID người gửi, các tệp đính kèm, tin nhắn hệ thống, ghim, thời gian ghim tin nhắn, các lượt thả cảm xúc	ID tin nhắn → ID hội thoại, ID người gửi, các tệp đính kèm, tin nhắn hệ thống, ghim, thời gian ghim tin nhắn, các lượt thả cảm xúc
⇒ Khoá chính của bảng: ID tin nhắn ⇒ Bảng Tin nhắn đã ở 3NF	

3.3.1.10 Chuẩn hoá bảng Tập đính kèm

Bảng 3.10 Bảng chuẩn hoá bảng Tập đính kèm

Danh sách thuộc tính	ID tệp đính kèm, ID tin nhắn, ID hội thoại, Đường dẫn nội dung, Tên tệp, Kích thước tệp, Đường dẫn thumbnail, Loại đính kèm
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi tệp đính kèm có một ID tệp đính kèm riêng, thuộc một tin nhắn trong một hội thoại, có duy nhất đường dẫn nội dung, tên tệp, kích thước tệp, đường dẫn thumbnail, loại đính kèm	ID tệp đính kèm → ID tin nhắn, ID hội thoại, Đường dẫn nội dung, Tên tệp, Kích thước tệp, Đường dẫn thumbnail, Loại đính kèm
⇒ Khoá chính của bảng: ID tệp đính kèm ⇒ Bảng Tệp đính kèm đã ở 3NF	

3.3.1.11 Chuẩn hoá bảng Phân công bệnh nhân - bác sĩ

Bảng 3.11 Bảng chuẩn hoá bảng Phân công bệnh nhân - bác sĩ

Danh sách thuộc tính	ID phân công, ID bệnh nhân, ID bác sĩ, Ngày bắt đầu
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi phân công bệnh nhân - bác sĩ có một ID phân công riêng, có duy nhất ID bệnh nhân, ID bác sĩ, ngày bắt đầu. Mỗi bệnh nhân thuộc một phân công duy nhất, còn mỗi bác sĩ có thể được giao nhiều phân công.	ID phân công → ID bệnh nhân, ID bác sĩ, Ngày bắt đầu
⇒ Khoá chính của bảng: ID phân công	
⇒ Bảng Phân công bệnh nhân - bác sĩ đã ở 3NF	

3.3.1.12 Chuẩn hoá bảng Tài khoản phê duyệt

Bảng 3.12 Bảng chuẩn hoá bảng Tài khoản phê duyệt

Danh sách thuộc tính	ID tài khoản phê duyệt, Email, Mật khẩu, Tên người dùng, Ngày sinh, Giới tính, Số điện thoại, Quyền, Trạng thái, Đường dẫn lưu trữ ảnh, Thông tin người dùng
Quy tắc nghiệp vụ	Phụ thuộc hàm
Mỗi tài khoản chờ phê duyệt có một ID riêng, có duy nhất Email, Mật khẩu, Tên người dùng, Ngày sinh, Giới tính, Số điện thoại, Quyền, Trạng thái, Đường dẫn lưu trữ ảnh, Thông tin người dùng	ID tài khoản phê duyệt → Email, Mật khẩu, Tên người dùng, Ngày sinh, Giới tính, Số điện thoại, Quyền, Trạng thái, Đường dẫn lưu trữ ảnh, Thông tin người dùng
⇒ Khoá chính của bảng: ID tài khoản phê duyệt	
⇒ Bảng Người dùng đã ở 3NF	

3.3.2 Từ điển dữ liệu

Bảng 3.13 Bảng user

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID người dùng.
account_id	STRING	ID tài khoản người dùng.
username	STRING	Họ và tên người dùng.
gender	INT	Giới tính của người dùng.
birth	DATE	Ngày tháng năm sinh của người dùng.
phone_number	STRING	Số điện thoại của người dùng.
image	STRING	Ảnh đại diện của người dùng.
status	INTEGER	Trạng thái sử dụng của người dùng (0 - còn đang sử dụng, 1 - không còn đang sử dụng).
information	STRING	Thông tin thêm về người dùng.
role	INTEGER	Chức vụ của người dùng (0 - bệnh nhân, 1 - bác sĩ, 2 - quản trị viên).
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu.
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu.

Bảng 3.14 Bảng account

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID tài khoản
email	STRING	Địa chỉ email của tài khoản.
password	STRING	Mật khẩu của tài khoản.

Bảng 3.15 Bảng register

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID tài khoản đăng ký
email	STRING	Địa chỉ email của tài khoản đăng ký.
password	STRING	Mật khẩu của tài khoản đăng ký.
username	STRING	Họ và tên của tài khoản đăng ký.
gender	INT	Giới tính của tài khoản đăng ký.
birth	DATE	Ngày sinh của tài khoản đăng ký.
phone_number	STRING	Số điện thoại của tài khoản đăng ký.
image	STRING	Ảnh đại diện của tài khoản đăng ký.
status	INTEGER	Trạng thái của tài khoản đăng ký (0 - đang chờ xử lý, 1 - được admin chấp nhận, 2 - bị admin từ chối).
information	STRING	Thông tin thêm về người dùng.
role	INTEGER	Chức vụ của tài khoản đăng ký (0 - bệnh nhân, 1 - bác sĩ, 2 - quản trị viên).

Bảng 3.16 Bảng record

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID bản ghi record
user_id	STRING	ID bệnh nhân
device_id	STRING	ID thiết bị theo dõi
record_type	INTEGER	Loại bản ghi (0 - PPG, 1 - PCG, 2 - HeartRate, 3 - ECG).
data_rec_url	STRING	Đường dẫn lưu trữ bản ghi.
start_time	DATE	Thời gian bắt đầu phiên đo.
end_time	DATE	Thời gian kết thúc phiên đo.
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu.
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu.

Bảng 3.17 Bảng patient_doctor_assignment

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID phân công bác sĩ - bệnh nhân
patient_id	STRING	ID bệnh nhân
doctor_id	INTEGER	ID bác sĩ
start_date	DATE	Ngày bắt đầu phân công.
end_date	DATE	Ngày kết thúc phân công.
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu.
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu.

Bảng 3.18 Bảng token

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID của mã đại diện xác thực
account_id	STRING	ID tài khoản
access_token	STRING	Mã đại diện cho người dùng thao tác với hệ thống.
refresh_token	STRING	Mã đại diện cho người dùng thao tác với hệ thống sau khi access_token hết hạn.
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu..
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu..

Bảng 3.19 Bảng device

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID thiết bị
user_id	STRING	ID bệnh nhân
doctor_id	STRING	ID bác sĩ
device_name	STRING	Tên thiết bị.
information	STRING	Thông tin thêm về thiết bị.
device_type	INTEGER	Loại thiết bị (1 - thiết bị đo điện tim, 2 - thiết bị đo nhịp tim, âm thanh tim).
start_date	DATE	Thời gian lúc bắt đầu sử dụng thiết bị.
status	INTEGER	Trạng thái của thiết bị (0 - đang được sử dụng, 1 - đang không được sử dụng).
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu.
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu.

Bảng 3.20 Bảng device_detail

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID thông số thiết bị
device_id	STRING	ID thiết bị
detail_name	STRING	Tên trường thông số thiết bị dựa vào detail_type.
infomation	STRING	Thông tin về trường detail_name của thiết bị.
value	STRING	Giá trị thông số của thiết bị dựa vào detail_type (1 - tần số, 2 - dữ liệu kết nối, 3 - dung lượng lưu trữ).
detail_type	INTEGER	Loại thông tin về thiết bị (1 - tín hiệu đo, 2 - loại kết nối, 3 - kiểu lưu trữ).
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu.
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu.

Bảng 3.21 Bảng conversation

Thuộc tính	Kiểu dữ liệu	Mô tả
conversation_id	STRING	ID cuộc hội thoại
name	STRING	Tên cuộc hội thoại
type	INTEGER	Loại cuộc hội thoại (0 - hội thoại 2 người, 1 - hội thoại nhóm, 2 - chatbot)
avatar	STRING	Đường dẫn ảnh đại diện của cuộc hội thoại
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu

Bảng 3.22 Bảng message

Thuộc tính	Kiểu dữ liệu	Mô tả
message_id	STRING	ID tin nhắn
conversation_id	STRING	ID hội thoại
sender_id	STRING	ID người gửi tin nhắn
attachments	LIST <Map>	Danh sách tệp tin đính kèm khi gửi tin nhắn
system_message	BOOLEAN	Tin nhắn trả về từ hệ thống
pin	BOOLEAN	Đánh dấu tin nhắn có được ghim không
pin_time	DATE	Thời gian ghim tin nhắn
reactions	LIST <Map>	Danh sách các phản ứng lại với tin nhắn này.
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu

Bảng 3.23 Bảng conversation_member

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID của thành viên hội thoại
conversation_id	STRING	ID hội thoại
user_id	STRING	ID người dùng
status_notify	INTEGER	Trạng thái thông báo tin nhắn (0 - có thông báo tin nhắn, 1 - không nhận thông báo tin nhắn).
seen	BOOLEAN	Tin nhắn đã được xem chưa.
role	INTEGER	Vai trò của người dùng trong hội thoại (tương ứng trong bảng user).
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu

Bảng 3.24 Bảng conversation_attachment

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID tệp đính kèm tin nhắn
conversation_id	STRING	ID cuộc hội thoại
message_id	STRING	ID tin nhắn
content_url	STRING	Đường dẫn lưu trữ tệp đính kèm
file_name	STRING	Tên của tệp tin.
size	INTEGER	Dung lượng của tệp tin.
thumbnail_url	STRING	Đoạn mã chứa hình ảnh tĩnh của tệp tin.
type	INTEGER	Loại của tệp tin (0 - ảnh, 1 - video, 2 - tệp).
created_at	DATE	Thời gian lúc tạo mới dữ liệu trong cơ sở dữ liệu
updated_at	DATE	Thời gian lúc thay đổi dữ liệu trong cơ sở dữ liệu

3.3.3 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 API

3.5.2 Sơ đồ tuần tự API

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

4.1.2 Server

4.1.2.1 Javascript

JavaScript là một ngôn ngữ lập trình động, được Brendan Eich phát triển vào năm 1995 và chuẩn hóa bởi tổ chức ECMA dưới tên gọi ECMAScript. Ban đầu được gọi là Mocha, sau đó đổi thành LiveScript và cuối cùng là JavaScript. Nó được sử dụng rộng rãi trong phát triển web để tạo ra các trang web tương tác và ứng dụng web động.

JavaScript chạy trên trình duyệt của người dùng, giúp tạo ra các trang web phản hồi nhanh. Nó có cú pháp đơn giản, dễ học, và hỗ trợ nhiều tính năng như lập trình hướng đối tượng dựa trên nguyên mẫu, lập trình không đồng bộ thông qua callback, promises, và async/await, cũng như khả năng tương tác và thay đổi Document Object Model (DOM) của trang web.[1]

4.1.2.2 NodeJs

Node.js là một môi trường chạy JavaScript phía máy chủ, được phát triển bởi Ryan Dahl vào năm 2009, dựa trên V8 engine của Google. Node.js cho phép các nhà phát triển sử dụng JavaScript để viết mã server-side, giúp tạo ra các ứng dụng web hiệu suất cao và có khả năng mở rộng. Node.js nổi bật với mô hình I/O không đồng bộ và dựa trên sự kiện, cho phép xử lý nhiều yêu cầu đồng thời mà không bị chặn, rất hữu ích cho các ứng dụng thời gian thực và hệ thống yêu cầu xử lý lượng lớn dữ liệu.[22]

Hình 4.1 Kiến trúc của NodeJS

Một trong những lợi thế lớn của Node.js là npm (Node Package Manager), hệ thống quản lý gói lớn nhất thế giới, cung cấp hàng ngàn thư viện và module sẵn có để mở rộng chức năng của ứng dụng. Node.js phù hợp cho việc xây dựng các API RESTful, ứng dụng web thời gian thực, ứng dụng đơn trang (Single Page Applications - SPAs), và các ứng dụng Internet of Things (IoT). Với cộng đồng phát triển mạnh mẽ và năng động, Node.js đã trở thành một công cụ quan trọng trong lĩnh vực phát triển web hiện đại.

4.1.2.3 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được phát triển bởi công ty MySQL AB và sau này được Oracle Corporation mua lại. MySQL sử dụng ngôn ngữ truy vấn cấu trúc (SQL) để quản lý và truy vấn dữ liệu, giúp các nhà phát triển dễ dàng tạo, quản lý và thao tác với cơ sở dữ liệu.[3]

MySQL nổi bật với hiệu suất cao, độ tin cậy và khả năng mở rộng, làm cho nó trở thành lựa chọn phổ biến cho các ứng dụng web và doanh nghiệp. Nó hỗ trợ nhiều tính năng như giao dịch, khóa hàng (row-level locking), và khả năng chịu tải cao, giúp đảm bảo tính toàn vẹn và nhất quán của dữ liệu.

Một trong những ưu điểm lớn của MySQL là khả năng tích hợp tốt với nhiều ngôn ngữ lập trình như PHP, Java, và Python, cùng với việc hỗ trợ nhiều hệ điều hành như Windows, Linux và macOS. MySQL cũng có một cộng đồng lớn và tài liệu phong phú, giúp các nhà phát triển dễ dàng tìm kiếm giải pháp và hỗ trợ.

Với các đặc điểm như hiệu suất mạnh mẽ, tính năng phong phú và khả năng mở rộng, MySQL được sử dụng rộng rãi trong nhiều ứng dụng từ các trang web nhỏ đến các hệ thống lớn, phức tạp của các doanh nghiệp và tổ chức lớn.

4.1.2.4 PostgreSQL

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS) mã nguồn mở, miễn phí, được phát triển bởi cộng đồng toàn cầu. PostgreSQL bắt nguồn từ dự án POSTGRES của Đại học California tại Berkeley vào những năm 1980. Nó chính thức được phát hành lần đầu tiên vào năm 1996 dưới tên PostgreSQL.

PostgreSQL hỗ trợ một loạt các kiểu dữ liệu đa dạng và phong phú, giúp nó trở nên linh hoạt và mạnh mẽ trong việc quản lý dữ liệu. Do đó trong hệ thống đề xuất chúng em đã sử dụng PostgreSQL để lưu trữ dữ liệu tin nhắn, giúp cải thiện khả năng quản lý dữ liệu và tăng tốc độ truy vấn tin nhắn.[16]

4.1.2.5 Postman

Postman là một công cụ phổ biến dành cho việc phát triển và thử nghiệm API, giúp các nhà phát triển tạo, quản lý và kiểm thử các API một cách hiệu quả. Ban đầu ra mắt dưới dạng một tiện ích mở rộng cho trình duyệt, Postman đã phát triển thành một ứng dụng độc lập với giao diện người dùng thân thiện và trực quan, hỗ trợ nhiều loại yêu cầu HTTP như GET, POST, PUT, DELETE.[11]

Postman cung cấp các tính năng mạnh mẽ như tạo và quản lý bộ sưu tập yêu cầu API, viết kiểm thử tự động bằng JavaScript, và tạo tài liệu API tự động. Ngoài ra, Postman còn hỗ trợ mock server để giả lập API và monitoring để theo dõi hiệu suất và tính khả dụng của API theo thời gian thực. Với những tính năng này, Postman đã trở

thành công cụ không thể thiếu đối với các nhà phát triển và nhóm phát triển API, giúp nâng cao hiệu quả và chất lượng trong quá trình phát triển phần mềm.

4.1.2.6 Docker

Docker là một nền tảng mã nguồn mở được thiết kế để tự động hóa việc triển khai, mở rộng và quản lý các ứng dụng trong các container. Container là một đơn vị phần mềm nhẹ, có thể chạy độc lập và bao gồm tất cả các thành phần cần thiết để chạy ứng dụng, bao gồm mã nguồn, thư viện, và cấu hình hệ thống.

Docker được phát triển bởi Docker, Inc. và lần đầu tiên được ra mắt vào năm 2013. Từ đó, Docker đã nhanh chóng trở thành một công cụ quan trọng trong quá trình phát triển phần mềm, đặc biệt là trong các môi trường DevOps.[12]

Docker không chỉ hỗ trợ các lập trình viên và các chuyên gia IT trong việc phát triển và triển khai ứng dụng, mà còn đóng vai trò quan trọng trong việc thúc đẩy các phương pháp làm việc hiện đại như Continuous Integration/Continuous Deployment (CI/CD) và microservices. Với Docker, các doanh nghiệp có thể đẩy nhanh quá trình phát triển phần mềm, cải thiện chất lượng sản phẩm, và giảm thiểu chi phí vận hành.

Hình 4.2 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 phát triển và xây dựng hệ thống. 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 để quản lý cơ sở dữ liệu, Github để quản lý code.

4.2.1 Kiến trúc Microservices

4.2.2 Triển khai Server và ứng dụng web trên máy chủ VPS

4.3 Kiểm thử

4.3.1 Kiểm thử hoạt động của các API

Môi trường:

- Base URL: <http://103.200.20.59/> hoặc <http://localhost:3000/>

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 api/auth/register

Bảng 4.1 Bảng kiểm thử API đăng ký tài khoản

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng chưa có tài khoản trên hệ thống	Thông tin đăng ký tài khoản <pre>{ "username": "Nguyen Van A", "password": "123456789", "email": "test@gmail.com", "birth": "", "gender": 1, "phone_number": "0123344562", "role": 0 }</pre>	Status code: 200 OK Response content: <pre>{ "status": "success", "message": "Your account is pending." }</pre>	OK
TC-2	Người dùng đã tồn tại tài khoản	Thông tin đăng ký <pre>{ "username": "Nguyen Van A", "password": "123456789", "email": "test@gmail.com", "birth": "", "gender": 1, "phone_number": "0123344562", "role": 0 }</pre>	Status code: 400 Bad Request Response content: <pre>{ "status": "error", "message": "Email has already been in use" }</pre>	OK

b) URL: POST api/auth/login

Bảng 4.2 Bảng kiểm thử API đăng nhập

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Thông tin tài khoản và mật khẩu hợp lệ	Thông tin đăng nhập { "email": email người dùng, "password": mật khẩu người dùng }	Status code: 200 OK Response content: { "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 content: { "status": "error", "message": "Invalid email or password" }	OK

c) URL: POST api/auth/logout

Bảng 4.3 Bảng kiểm thử API đăng xuất

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	User đã đăng nhập vào hệ thống	JWT Token tồn tại	Status code: 200 OK Response content: { "status": "success", "message": "Logged out successfully" }	OK

TC-2	User chưa đăng nhập vào hệ thống	JWT Token không tồn tại	Status code: 401 Unauthorized Response content: { "status": "error", "message": "No token found" }	OK
------	----------------------------------	-------------------------	--	----

d) URL: POST api/auth/reset-password

Bảng 4.4 Bảng kiểm thử API gửi token đặt lại mật khẩu

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng đã đăng ký tài khoản	Email người dùng { "email": email người dùng }	Status code: 200 OK Response content: { "status": "success", "message": "Reset token sent to email" "resetToken": token }	OK
TC-2	Người dùng chưa đăng ký tài khoản	Email người dùng { "email": email người dùng }	Status code: 404 Not Found Response content: { "status": "error", "message": "User not found" }	OK

e) URL: POST api/auth/reset-password/reset

Bảng 4.5 Bảng kiểm thử API đặt lại mật khẩu

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	User đã đăng ký tài khoản, reset token và mật khẩu hợp lệ	Thông tin reset mật khẩu { "resetToken": "816e8d", "password": "123456", "email": "test@gmail.com" }	Status code: 200 OK Response content: { "status": "success", "message": "Password reset successful" }	OK
TC-2	Reset token không hợp lệ	Thông tin reset mật khẩu { "resetToken": "816e8d", "password": "123456", "email": "test@gmail.com" }	Status code: 400 Bad Request Response content: { "status": "error", "msg": "Invalid reset token" }	OK
TC-3	Reset token hết hạn	Thông tin reset mật khẩu { "resetToken": "816e8d", "password": "123456", "email": "test@gmail.com" }	Status code: 400 Bad Request Response content: { "status": "error", "message": "Reset token has expired" }	OK

4.3.1.2 API liên quan đến việc xét duyệt đăng ký tài khoản

a) URL: GET api/register/list

Bảng 4.6 Bảng kiểm thử API đăng ký tài khoản

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng là admin của hệ thống kèm theo token	Access token của người dùng trong Bearer Token	Status code: 200 OK Response content: { "status": "success", "data": Danh sách các tài khoản chờ phê duyệt }	OK
TC-2	Người dùng không phải là admin của hệ thống kèm theo token	Access token của người dùng trong Bearer Token	Status code: 403 Forbidden Response content: { "status": "error", "message": "You don't have permission to access" }	OK
TC-3	Yêu cầu không kèm theo token	NULL	Status code: 401 Unauthorized Response content: { "status": "error", "message": "No token found" }	OK

b) URL: POST api/register/accepted

Bảng 4.7 Bảng kiểm thử API chấp nhận tài khoản

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Tài khoản phê duyệt tồn tại trên hệ thống	ID tài khoản phê duyệt	Status code: 200 OK Response content: { "status": "success", "message": "Accept account successfully" }	OK

TC-2	Tài khoản phê duyệt không tồn tại trên hệ thống	ID tài khoản phê duyệt	Status code: 404 Not Found Response content: { "status": "error", "message": "Register account not found" }	OK
------	---	------------------------	--	----

c) URL: POST api/register/rejected

Bảng 4.8 Bảng kiểm thử API từ chối phê duyệt tài khoản

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Tài khoản phê duyệt tồn tại trên hệ thống	ID tài khoản phê duyệt	Status code: 200 OK Response content: { "status": "success", "message": "Reject account successfully" }	OK
TC-2	Tài khoản phê duyệt tồn tại trên hệ thống	ID tài khoản phê duyệt	Status code: 404 Not Found Response content: { "status": "error", "message": "Register account not found" }	OK

4.3.1.3 API liên quan đến thông tin người dùng

a) URL: GET api/user

Bảng 4.9 Bảng kiểm thử API lấy danh sách người dùng

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Người dùng là admin của hệ thống kèm theo token	Access token của người dùng trong Bearer Token	Status code: 200 OK Response content: { "status": "success", "data": Danh sách người dùng }	OK
TC-2	Người dùng không phải là admin của hệ thống kèm theo token	Access token của người dùng trong Bearer Token	Status code: 403 Forbidden Response content: { "status": "error", "message": "You don't have permission to access" }	OK
TC-3	Yêu cầu không kèm theo token	NULL	Status code: 401 Unauthorized Response content: { "status": "error", "message": "No token found" }	OK

b) URL: GET api/user/id/:userId

Bảng 4.10 Bảng kiểm thử API lấy thông tin của người dùng thông qua ID

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng tồn tại với ID tương ứng	ID người dùng	Status code: 200 OK Response content: { "status": "success", data: Thông tin người dùng }	OK
TC-2	Người dùng không tồn tại với ID tương ứng	ID người dùng	Status code: 404 Not Found Response content: { "status": "error", "message": "User not found" }	OK

c) URL: GET api/user/:role

Bảng 4.11 Bảng kiểm thử API lấy thông tin của người dùng thông qua chức vụ

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Role tồn tại trong CSDL	Role	Status code: 200 OK Response content: { "status": "success", data: Danh sách người dùng ứng với role }	OK
TC-2	Role không tồn tại trong CSDL	Role	Status code: 404 Not Found Response content: { "status": "error", "message": "Role does not exist" }	OK

d) URL: POST api/user/update

Bảng 4.12 Bảng kiểm thử API cập nhật thông tin người dùng

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng tồn tại với ID tương ứng	Thông tin cập nhật của người dùng { "username": Họ và tên, "birth": Ngày sinh, "phone_number": Số điện thoại, "gender": Giới tính, "status": Trạng thái }	Status code: 200 OK Response content: { "status": "success", "data": Thông tin sau khi cập nhật của người dùng }	OK

TC-2	Người dùng không tồn tại với ID tương ứng	Thông tin cập nhật của người dùng { "username": Họ và tên, "birth": Ngày sinh, "phone_number": Số điện thoại, "gender": Giới tính, "status": Trạng thái }	Status code: 404 Not found Response content: { "status": "error", "message": "User not found" }	OK
------	---	--	--	----

e) URL: DELETE api/user/delete/:userId

Bảng 4.13 Bảng kiểm thử API xóa thông tin người dùng

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng tồn tại với ID tương ứng	ID người dùng	Status code: 200 OK Response content: { "status": "success", "message": "Delete user successfully" }	OK
TC-2	Người dùng không tồn tại với ID tương ứng	ID người dùng	Status code: 404 Not found Response content: { "status": "error", "message": "User not found" }	OK

4.3.1.4 API liên quan đến thiết bị

a) URL: GET api/device

Bảng 4.14 Bảng kiểm thử API lấy danh sách thiết bị

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Người dùng đã đăng nhập vào hệ thống	Access token của người dùng trong Bearer Token	Status code: 200 OK Response content: { "status": "success", "data": Danh sách thiết bị }	OK
TC-2	Người dùng chưa đăng nhập vào hệ thống	Access token của người dùng trong Bearer Token	Status code: 401 Unauthorized Response content: { "status": "error", "message": "No token found" }	OK

b) URL: GET api/device/:deviceId

Bảng 4.15 Bảng kiểm thử API lấy thông tin thiết bị theo ID

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Thiết bị tồn tại với ID tương ứng	ID thiết bị	Status code: 200 OK Response content: { "status": "success", data: Thông tin của thiết bị }	OK
TC-2	Thiết bị không tồn tại với ID tương ứng	ID thiết bị	Status code: 404 Not Found Response content: { "status": "error", "message": "Device not found" }	OK

c) URL: POST api/device/add

Bảng 4.16 Bảng kiểm thử API thêm thiết bị

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
-----------	-----------	---------	------------------	---------

TC-1	Bệnh nhân và bác sĩ tồn tại với ID tương ứng	Thông tin thiết bị { "device_name": Tên thiết bị, "device_type": Loại thiết bị, "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "status": Trạng thái, "infomation": Thông tin thiết bị, "start_date": Ngày bắt đầu sử dụng }	Status code: 200 OK Response content: { "status": "success", data: Thông tin của thiết bị }	OK
TC-2	Bệnh nhân hoặc bác sĩ không tồn tại với ID tương ứng	Thông tin thiết bị { "device_name": Tên thiết bị, "device_type": Loại thiết bị, "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "status": Trạng thái, "infomation": Thông tin thiết bị, "start_date": Ngày bắt đầu sử dụng }	Status code: 404 Not found Response content: { "status": "error", "message": "User not found" }	OK

d) URL: POST api/device/update

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 tương ứng	Thông tin thiết bị { "device_name": Tên thiết bị, "device_type": Loại thiết bị, "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "status": Trạng thái, "infomation": Thông tin thiết bị, "start_date": Ngày bắt đầu sử dụng }	Status code: 200 OK Response content: { "status": "success", data: Thông tin sau khi cập nhật của thiết bị }	OK
TC-2	Thiết bị không tồn tại với ID tương ứng	Thông tin thiết bị { "device_name": Tên thiết bị, "device_type": Loại thiết bị, "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "status": Trạng thái, "infomation": Thông tin thiết bị, "start_date": Ngày bắt đầu sử dụng }	Status code: 404 Not found Response content: { "status": "error", "message": "Device not found" }	OK

TC-3	Bệnh nhân hoặc bác sĩ không tồn tại với ID tương ứng	Thông tin thiết bị { "device_name": Tên thiết bị, "device_type": Loại thiết bị, "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "status": Trạng thái, "infomation": Thông tin thiết bị, "start_date": Ngày bắt đầu sử dụng }	Status code: 404 Not found Response content: { "status": "error", "message": "User not found" }	OK
------	--	--	--	----

e) URL: DELETE api/device/delete/:deviceId

Bảng 4.18 Bảng kiểm thử API xóa thiết bị

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Thiết bị tồn tại với ID tương ứng	ID thiết bị	Status code: 200 OK Response content: { "status": "success", "message": "Delete device successful" }	OK
TC-2	Thiết bị không tồn tại với ID tương ứng	ID thiết bị	Status code: 404 Not found Response content: { "status": "error", "message": "Device not found" }	OK

4.3.1.5 API liên quan đến bản ghi ECG

Tham khảo bảng ?? để xem thông tin của các api liên quan

a) URL: GET api/record

Bảng 4.19 Bảng kiểm thử API lấy danh sách bản ghi

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng đăng nhập vào hệ thống	Access token của người dùng trong Bearer Token	Status code: 200 OK Response content: { "status": "success", data: Danh sách bản ghi }	OK
TC-2	Người dùng chưa đăng nhập vào hệ thống	NULL	Status code: 401 Unauthorized Response content: { "status": "error", "message": "No token found" }	OK

b) URL: GET api/record/user/:userId

Bảng 4.20 Bảng kiểm thử API lấy danh sách phiên đo ECG của bệnh nhân

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	NULL	ID bệnh nhân	Status code: 200 OK Response content: { "status": "success", "data": Danh sách thông tin của tất cả phiên đo của bệnh nhân }	OK
TC-2	Lỗi đường truyền server	ID bệnh nhân	Status code: 500 Internal Server Error Response content: { "status": "error", "message": "An error occurred while retrieving the records" }	OK

c) URL: GET api/record/doctor/:doctorId

Bảng 4.21 Bảng kiểm thử API lấy danh sách phiên đo ECG mà bác sĩ phụ trách

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	NULL	ID bác sĩ	Status code: 200 OK Response content: { "status": "success", "data": Danh sách thông tin của tất cả phiên đo mà bác sĩ phụ trách }	OK
TC-2	Lỗi đường truyền server	ID bác sĩ	Status code: 500 Internal Server Error Response content: { "status": "error", "message": "An error occurred while retrieving the records" }	OK

d) URL: GET api/record/id/:recordId

Bảng 4.22 Bảng kiểm thử API lấy thông tin một phiên đo ECG

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Phiên đo tồn tại với ID tương ứng	ID phiên đo	Status code: 200 OK Response content: { "status": "success", data: Thông tin phiên đo ECG }	OK
TC-2	Phiên đo không tồn tại với ID tương ứng	ID phiên đo	Status code: 404 Not Found Response content: { "status": "error", "message": "Record not found" }	OK

e) URL: GET api/record/getData/:recordId

Bảng 4.23 Bảng kiểm thử API lấy dữ liệu bản ghi một phiên đo ECG

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Phiên đo tồn tại với ID tương ứng	ID phiên đo	Status code: 200 OK Response content: { "status": "success", data: Dữ liệu bản ghi phiên đo ECG }	OK
TC-2	Phiên đo không tồn tại với ID tương ứng	ID phiên đo	Status code: 404 Not Found Response content: { "status": "error", "message": "Record not found" }	OK
TC-3	Lỗi đường truyền server	ID phiên đo	Status code: 500 Internal Server Error Response content: { "status": "error", "message": "An error occurred while retrieving the records" }	OK

f) URL: GET api/record/download/:recordId

Bảng 4.24 Bảng kiểm thử API tải dữ liệu đo ECG

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Bản ghi tồn tại với ID phiên đo tương ứng	ID phiên đo	Status code: 200 OK Response content: { "status": "success", data: File bản ghi dữ liệu ECG }	OK

TC-2	Phiên đo không tồn tại với ID tương ứng	ID phiên đo	Status code: 404 Not Found Response content: { "status": "error", "message": "Record not found" }	OK
TC-3	Lỗi đường truyền server	ID phiên đo	Status code: 500 Internal Server Error Response content: { "status": "error", "message": "An error occurred while downloading the record file" }	OK

g) URL: POST api/record/update

Bảng 4.25 Bảng kiểm thử API cập nhật thông tin phiên đo

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Phiên đo tồn tại với ID tương ứng	Thông tin phiên đo { "user_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại bản ghi, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc }	Status code: 200 OK Response content: { "status": "success", "message": "Update record successful" }	OK

TC-2	Phiên đo không tồn tại với ID tương ứng	Thông tin phiên đo { "user_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại bản ghi, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc }	Status code: 404 Not Found Response content: { "status": "error", "message": "Record not found" }	OK
TC-3	Thông tin người dùng hoặc thiết bị không tồn tại với ID tương ứng	Thông tin phiên đo { "user_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại bản ghi, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc }	Status code: 404 Not Found Response content: { "status": "error", "message": "Update info error" }	OK

h) URL: DELETE api/record/delete/:recordId

Bảng 4.26 Bảng kiểm thử API xóa thông tin một phiên đo ECG

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Phiên đo tồn tại với ID tương ứng	ID phiên đo	Status code: 200 OK Response content: { "status": "success", "message": "Delete record successfull" }	OK

TC-2	Phiên đo không tồn tại với ID tương ứng	ID phiên đo	Status code: 404 Not Found Response content: { "status": "error", "message": "Record not found" }	OK
------	---	-------------	--	----

4.3.1.6 API liên quan liên quan đến việc phân công bác sĩ - bệnh nhân

Tham khảo bảng ?? để xem thông tin của các api liên quan

a) URL: GET api/pda

Bảng 4.27 Bảng kiểm thử API lấy danh sách phân công bác sĩ - bệnh nhân

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Người dùng truy cập với quyền admin	Access token của người dùng trong Bearer Token	Status code: 200 OK Response content: { "status": "success", data: Danh sách phân công bác sĩ bệnh nhân }	OK
TC-2	Người dùng truy cập không phải với tư cách admin	NULL	Status code: 403 Forbidden Response content: { "status": "error", "message": "You don't have permission to access" }	OK
TC-3	Người dùng chưa đăng nhập	NULL	Status code: 401 Unauthorized Response content: { "status": "error", "message": "No token found" }	OK

b) URL: POST api/pda/create

Bảng 4.28 Bảng kiểm thử API tạo một phân công bác sĩ - bệnh nhân

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	NULL	Thông tin phân công bác sĩ - bệnh nhân { "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "start_date": Ngày bắt đầu, "end_date": Ngày kết thúc }	Status code: 200 OK Response content: { "status": "success", data: Thông tin phân công }	OK
TC-2	Thông tin bệnh nhân hoặc bác sĩ không tồn tại với ID tương ứng	Thông tin phiên đo { "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "start_date": Ngày bắt đầu, "end_date": Ngày kết thúc }	Status code: 404 Not Found Response content: { "status": "error", "message": "User not found" }	OK

c) URL: POST api/pda/update

Bảng 4.29 Bảng kiểm thử API cập nhật thông tin phân công

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Phân công tồn tại với ID tương ứng	Thông tin phân công { "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "start_date": Ngày bắt đầu, "end_date": Ngày kết thúc }	Status code: 200 OK Response content: { "status": "success", "message": "Update pda successful" }	OK

TC-2	Phân công không tồn tại với ID tương ứng	Thông tin phân công { "user_id": ID bệnh nhân, "device_id": ID thiết bị, "record_type": Loại bản ghi, "start_time": Thời gian bắt đầu, "end_time": Thời gian kết thúc }	Status code: 404 Not Found Response content: { "status": "error", "message": "PDA not found" }	OK
TC-3	Thông tin bác sĩ hoặc bệnh nhân không tồn tại với ID tương ứng	Thông tin phân công { "user_id": ID bệnh nhân, "doctor_id": ID bác sĩ, "start_date": Ngày bắt đầu, "end_date": Ngày kết thúc }	Status code: 404 Not Found Response content: { "status": "error", "message": "User not found" }	OK

d) URL: DELETE api/pda/delete/:pdaId

Bảng 4.30 Bảng kiểm thử API xóa thông tin phân công

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	Phân công tồn tại với ID tương ứng	ID phân công	Status code: 200 OK Response content: { "status": "success", "message": "Delete PDA successfull" }	OK
TC-2	Phân công không tồn tại với ID tương ứng	ID phân công	Status code: 404 Not Found Response content: { "status": "error", "message": "PDA not found" }	OK

e) URL: GET api/pda/patient/{:doctorId}

Bảng 4.31 Bảng kiểm thử API lấy danh sách bệnh nhân mà bác sĩ đang quản lý theo ID bác sĩ

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	NULL	ID bác sĩ	Status code: 200 OK Response content: { "status": "success", data: Thông tin của các bệnh nhân mà bác sĩ đang được phân công }	OK
TC-2	Bác sĩ không tồn tại với ID tương ứng	ID bác sĩ	Status code: 404 Not Found Response content: { "status": "error", "message": "Doctor not found" }	OK

f) URL: GET api/pda/doctor/{:patientId}

Bảng 4.32 Bảng kiểm thử API lấy thông tin bác sĩ đang quản lý bệnh nhân theo ID bệnh nhân

Test case	Điều kiện	Đầu vào	Đầu ra mong muốn	Kết quả
TC-1	NULL	ID bệnh nhân	Status code: 200 OK Response content: { "status": "success", data: Thông tin của bác sĩ được phân công cho bệnh nhân }	OK

TC-2	Bệnh nhân không tồn tại với ID tương ứng	ID bệnh nhân	Status code: 404 Not Found Response content: { "status": "error", "message": "Patient not found" }	OK
------	--	--------------	---	----

4.3.2 Kiểm thử ứng dụng web

Bảng 4.33 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. Vào màn hình đăng nhập → Nhập thông tin tài khoản hoặc mật khẩu chưa có trên hệ thống → Nhấn nút đăng nhập 2. Vào màn hình đăng nhập → Nhập thông tin tài khoản hoặc mật khẩu hợp lệ → Nhấn nút đăng nhập	1. Hiển thị thông báo sai email hoặc mật khẩu 2. Đăng nhập thành công vào hệ thống	OK
Kiểm tra chức năng đăng ký	1. Chọn màn hình đăng ký tài khoản → Nhập thông tin đăng ký tài khoản đã tồn tại → Nhấn đăng ký 2. Chọn chức năng đăng ký tài khoản → Nhập các thông tin tài khoản đăng ký hợp lệ → Chọn nút đăng ký	1. Hiển thị thông báo email đã tồn tại trong hệ thống 2. Hiển thị thông báo tài khoản đang chờ phê duyệt	OK
Kiểm tra chức năng quản lý người dùng	1. Vào 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. Hiển thị danh sách người dùng 2. Hiển 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 danh sách bác sĩ	<ol style="list-style-type: none"> 1. Vào màn hình danh sách bác sĩ phụ trách 2. Xem thông tin cụ thể của bác sĩ 	<ol style="list-style-type: none"> 1. Hiện thị danh sách bác sĩ 2. Hiện thị thông tin cụ thể bác sĩ 	OK
Kiểm tra chức năng xem danh sách bệnh nhân	<ol style="list-style-type: none"> 1. Vào màn hình quản lý bệnh nhân 2. Hiện thị thông tin cụ thể bệnh nhân 	<ol style="list-style-type: none"> 1. Hiện thị danh sách bệnh nhân 2. Hiện thị thông tin cụ thể bệnh nhân 	OK
Kiểm tra chức năng quản lý thiết bị	<ol style="list-style-type: none"> 1. Vào 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 	<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 phiên đo	<ol style="list-style-type: none"> 1. Vào màn hình quản lý phiên đo 2. Sửa thông tin danh mục có sẵn → Lưu thay đổi 3. Xem đồ thị một phiên đo 4. Tải bản ghi phiên đo → Lưu tệp 	<ol style="list-style-type: none"> 1. Hiện thị danh sách phiên đo 2. Cập nhật thông tin phiên đo 3. Hiện thị đồ thị dữ liệu phiên đo 4. Tải file dữ liệu phiên đo thành công 	OK
Kiểm tra chức năng quản lý phân công bệnh nhân - bác sĩ	<ol style="list-style-type: none"> 1. Vào màn hình quản lý phân công 2. Phân công bệnh nhân cho bác sĩ → Lưu thông tin 3. Sửa thông tin phân công có sẵn → Lưu thay đổi 4. Hủy phân công bệnh nhân cho bác sĩ 	<ol style="list-style-type: none"> 1. Hiện thị danh sách phân công 2. Phân công thành công 3. Cập nhật thông tin phân công thành công 4. Hủy phân công thành công 	OK

Kiểm tra chức năng quản lý phê duyệt tài khoản	1. Vào màn hình quản lý phê duyệt tài khoản 2. Phê duyệt/Từ chối tài khoản	1. Hiển thị danh sách tài khoản phê duyệt 2. Phê duyệt/Từ chối tài khoản thành công	OK
Kiểm tra chức năng quản lý thông tin cá nhân	1. Vào màn hình quản lý thông tin cá nhân 2. Sửa thông tin cá nhân có sẵn → Lưu thông tin	1. Hiển thị thông tin tài khoản cá nhân 2. Cập nhật thông tin cá nhân thành công	OK
Kiểm tra chức năng nhắn tin	1. Vào màn hình nhắn tin bệnh nhân/bác sĩ 2. Nhập và gửi tin nhắn	1. Hiển thị danh sách các cuộc hội thoại 2. Hiển thị tin nhắn lên màn hình	OK
Kiểm tra chức năng hỏi, nhận tư vấn từ trợ lý ảo	1. Vào màn hình nhắn tin với trợ lý ảo 2. Nhập và gửi tin nhắn	1. Hiển thị màn hình nhắn tin với trợ lý ảo 2. Hiển thị tin nhắn lên màn hình và phản hồi từ trợ lý ảo	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. kèm theo đó là quá trình kiểm thử với mục tiêu và quy trình rõ ràng. Việc kiểm thử đóng một vai trò to lớn trong quá trình phát triển hệ thống, giúp cho hệ thống được hoàn thiện một cách tối đa.

KẾT LUẬN

Kết luận chung

Trong quá trình thực hiện đồ án, chúng em đã thực hiện các nghiên cứu và phát triển để hiện thực một hệ thống chất lượng và hữu ích cho việc quản lý dữ liệu điện tâm đồ. Trải qua các phần của đồ án, chúng em đã tiến hành nghiên cứu về cơ sở lý thuyết về tín hiệu ECG, thiết kế cơ sở dữ liệu phù hợp, xây dựng giao diện người dùng tương tác và triển khai ứng dụng trên môi trường máy chủ thực tế.

Qua quá trình thực hiện, chúng em đã đạt được những kết quả đáng kể, bao gồm xây dựng giao diện người dùng trên web, tích hợp các chức năng quản lý bệnh nhân, bác sĩ, tin tức và dữ liệu. Xây dựng thành công tính năng tự động triển khai ứng dụng khi có thay đổi mới.

Tuy nhiên, trong quá trình triển khai, chúng em cũng đã gặp một số thách thức và vấn đề chưa giải quyết hoàn toàn. Ví dụ, việc tích hợp thêm các thuật toán phân tích tiên tiến có thể là một hướng phát triển tiềm năng. Đồng thời, cần tối ưu hóa hệ thống để đảm bảo tính sẵn sàng và khả năng mở rộng trong tương lai.

Hướng phát triển

Để nâng cao hiệu suất và tính năng của ứng dụng, chúng em đề xuất một số hướng phát triển tiềm năng sau này:

- Phát triển thêm các thuật toán phân tích sâu hơn để cung cấp độ chính xác cao hơn trong việc chẩn đoán các vấn đề về tim.
- Tích hợp khả năng dự báo và cảnh báo sớm về các tình trạng nguy hiểm hoặc không bình thường trong dữ liệu ECG.
- Mở rộng ứng dụng để hỗ trợ nhiều loại thiết bị cảm biến và chuẩn giao tiếp khác nhau.
- Phát triển giao diện web thân thiện hơn để người dùng có thể truy cập và theo dõi dữ liệu ECG mọi lúc, mọi nơi.

Kiến nghị và đề xuất

Dựa trên kinh nghiệm từ quá trình phát triển đồ án, chúng em có một vài đề xuất như sau:

- Đảm bảo tích hợp các phản hồi từ người dùng, chuyên gia y tế và các nhóm nghiên cứu để liên tục cải thiện và hoàn thiện ứng dụng.

- Xây dựng thêm các buổi hội thảo để giới thiệu về hệ thống, hướng dẫn cách dùng cho các chuyên gia y tế hoặc các thành viên trong nhóm nghiên cứu muốn tích.
- Tìm kiếm cơ hội để đưa hệ thống vào thực tiễn trong môi trường bệnh viện hoặc phòng nghiên cứu.

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] Openai document, 2024.
- [15] Reactjs docs: Getting started, 2024.
- [16] What is postgresql?, 2024.
- [17] <https://cloud.z.com/>. Ci/cd là gì? ci/cd có liên hệ gì tới agile và devops?, 2023.
- [18] <https://giadinh.suckhoedoisong.vn>. Chuyện 1 bệnh nhân covid-19 không chịu “nằm không”, 2023.
- [19] Iryna Kurkina. What is node.js and why you should use it - academy smart, 2023.
- [20] 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.

- [21] 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.
- [22] Benjamin Semah. What exactly is node.js? explained for beginners, 2022.
- [23] Team TIS. The benefits of node.js for web application development?, 2022.