

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



**ĐỒ ÁN**  
**TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG  
TÍCH HỢP TRONG HỆ THỐNG QUẢN LÝ DỮ LIỆU ĐIỆN  
TIM, HỖ TRỢ KẾT NỐI GIỮA BÁC SĨ - BỆNH NHÂN**

Sinh viên thực hiện: **Trần Xuân Quyền**  
Lớp Điện tử 06 - K66

Giảng viên hướng dẫn: **TS. Hàn Huy Dũng**

Hà Nội 02/2025

# LỜI NÓI ĐẦU

Trong thời đại công nghệ phát triển vượt bậc, Internet vạn vật (IoT) đã trở thành một phần không thể thiếu trong nhiều lĩnh vực, đặc biệt là y tế. Việc ứng dụng IoT vào quản lý thiết bị y tế đã mở ra những cơ hội mới, góp phần nâng cao chất lượng chăm sóc sức khỏe và tối ưu hóa các quy trình y khoa. Tại Việt Nam, trong bối cảnh công nghiệp hóa, hiện đại hóa đang diễn ra mạnh mẽ, việc áp dụng các công nghệ tiên tiến vào y tế không chỉ giúp cải thiện hệ thống chăm sóc sức khỏe mà còn xây dựng nền tảng vững chắc để phát triển nguồn nhân lực khỏe mạnh, sẵn sàng đóng góp cho sự phát triển của đất nước.

Đồ án này trình bày một hệ thống ứng dụng di động tích hợp trong quản lý dữ liệu tim mạch, hỗ trợ kết nối trực tiếp giữa bệnh nhân và bác sĩ. Thông qua hệ thống này, bệnh nhân có thể dễ dàng theo dõi tình trạng sức khỏe tại nhà, kết nối với các bác sĩ để nhận được sự tư vấn kịp thời và chuyên nghiệp. Giao diện thân thiện của ứng dụng cho phép người dùng đặt lịch hẹn, xem lịch sử bệnh án, theo dõi dữ liệu điện tim, và tương tác trực tiếp với đội ngũ y bác sĩ một cách hiệu quả.

Trong quá trình thực hiện đồ án, em đã có cơ hội làm việc cùng team web của SPARC Laboratory và nhận được sự hướng dẫn tận tình từ các thầy cô và anh/chị/bạn trong các phòng thí nghiệm thuộc khoa Điện - Điện tử. Đặc biệt, em xin gửi lời cảm ơn chân thành đến TS. Hàn Huy Dũng và SPARC Laboratory – những người đã trực tiếp hướng dẫn, chỉ ra các điểm cần cải thiện trong quá trình thực hiện đồ án cũng như thiết kế hệ thống. Đồng thời, em cũng rất trân trọng sự hỗ trợ và hợp tác từ nhóm firmware SPARC Lab, qua đó em đã học hỏi và tích lũy được nhiều kiến thức quý báu.

Mặc dù đã nỗ lực hết mình trong quá trình hoàn thiện đồ án, nhưng khó có thể tránh khỏi những thiếu sót. Em rất mong nhận được ý kiến đóng góp từ quý thầy cô và bạn đọc để có thể cải thiện và phát triển đề tài này tốt hơn trong tương lai.

Em xin chân thành cảm ơn!

# LỜI CAM ĐOAN

Em là Trần Xuân Quyên, mã số sinh viên 20214068, sinh viên lớp Điện tử 06, khóa K66.

Em xin cam đoan toàn bộ nội dung được trình bày trong đồ án "Phát triển ứng dụng di động tích hợp trong hệ thống quản lý dữ liệu tim mạch, hỗ trợ kết nối giữa bệnh nhân - bác sĩ" là kết quả quá trình tìm hiểu, nghiên cứu và làm việc của 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. 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**

**TRẦN XUÂN QUYÊN**

# PHÂN CÔNG CÔNG VIỆC

STT	Nội dung	Thành viên
1	Tìm hiểu đề tài, đề xuất hệ thống	Quyển
2	Phân tích hệ thống	Quyển
3	Thiết kế cơ sở dữ liệu	Quyển
4	Thiết kế API	Quyển
5	Xây dựng phía back-end	Quyển
6	Thiết kế ứng dụng	Quyển
7	Kiểm thử	Quyển
8	Viết quyển đồ án	Quyển

# MỤC LỤC

<b>DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT</b>	<b>i</b>
<b>DANH MỤC KÝ HÌNH VẼ</b>	<b>ii</b>
<b>DANH MỤC BẢNG BIỂU</b>	<b>v</b>
<b>TÓM TẮT ĐỒ ÁN</b>	<b>vi</b>
<b>PHẦN MỞ ĐẦU</b>	<b>viii</b>
<b>CHƯƠNG 1. THU THẬP YÊU CẦU</b>	<b>1</b>
1.1 Phân tích yêu cầu hệ thống . . . . .	1
1.1.1 Yêu cầu về người dùng hệ thống . . . . .	1
1.1.2 Yêu cầu chức năng của hệ thống . . . . .	1
1.1.3 Yêu cầu phi chức năng của hệ thống . . . . .	2
<b>CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG</b>	<b>4</b>
2.1 Thẻ CRC (Class - Responsibility - Collaboration Card) . . . . .	4
2.1.1 Thẻ CRC lớp Tài khoản người dùng . . . . .	4
2.1.2 Thẻ CRC lớp Token . . . . .	5
2.1.3 Thẻ CRC lớp Vai trò người dùng . . . . .	5
2.1.4 Thẻ CRC lớp Trạng thái người dùng . . . . .	6
2.1.5 Thẻ CRC lớp Người dùng . . . . .	7
2.1.6 Thẻ CRC lớp Loại thiết bị . . . . .	8
2.1.7 Thẻ CRC lớp Trạng thái thiết bị . . . . .	8
2.1.8 Thẻ CRC lớp Thiết bị . . . . .	9
2.1.9 Thẻ CRC lớp Thông số kỹ thuật . . . . .	10
2.1.10 Thẻ CRC lớp Dữ liệu phiên đo . . . . .	10
2.1.11 Thẻ CRC lớp Loại lịch hẹn . . . . .	11

2.1.12	Thẻ CRC lớp Trạng thái lịch hẹn . . . . .	12
2.1.13	Thẻ CRC lớp Lịch hẹn . . . . .	12
2.1.14	Thẻ CRC lớp Lịch hẹn của bác sĩ . . . . .	13
2.1.15	Thẻ CRC lớp Thông báo liên quan đến lịch hẹn . . . . .	14
2.1.16	Thẻ CRC lớp Chẩn đoán cho bệnh nhân . . . . .	15
2.1.17	Thẻ CRC lớp Tin nhắn . . . . .	16
2.1.18	Thẻ CRC lớp Nhóm trò chuyện . . . . .	16
2.2	Sơ đồ lớp . . . . .	17
2.3	Sơ đồ tuần tự . . . . .	20
2.4	Xử lý và phân tích dữ liệu . . . . .	20
2.5	Kết luận . . . . .	22

### **CHƯƠNG 3. THIẾT KẾ HỆ THỐNG** **23**

3.1	Sơ đồ kiến trúc tổng quan của hệ thống . . . . .	23
3.2	Sơ đồ khối phần mềm . . . . .	25
3.2.1	Website dành cho bệnh nhân . . . . .	25
3.2.2	Website dành cho bác sĩ . . . . .	26
3.2.3	Website cho quản trị viên . . . . .	27
3.3	Thiết kế cơ sở dữ liệu . . . . .	27
3.3.1	Chuẩn hoá 3NF . . . . .	27
3.3.2	Từ điển dữ liệu . . . . .	30
3.3.3	Sơ đồ ERD . . . . .	31
3.4	Thiết kế giao diện . . . . .	32
3.5	Thiết kế các chức năng cho website và server . . . . .	32
3.5.1	Thiết kế API . . . . .	32
3.5.2	Sơ đồ tuần tự API . . . . .	32
3.6	Kết luận chương . . . . .	32

### **CHƯƠNG 4. TRIỂN KHAI VÀ KIỂM THỬ** **33**

4.1	Công nghệ sử dụng . . . . .	33
-----	-----------------------------	----

4.1.1	Thiết kế giao diện website . . . . .	33
4.1.2	Server . . . . .	33
4.2	Triển khai ứng dụng . . . . .	35
4.2.1	Kiến trúc Microservices . . . . .	35
4.2.2	Triển khai Server và ứng dụng web trên máy chủ VPS . . . . .	35
4.3	Kiểm thử . . . . .	35
4.3.1	Kiểm thử hoạt động của các API . . . . .	35
4.3.2	Kiểm thử ứng dụng web . . . . .	60
4.4	Kết luận chương . . . . .	62
	<b>PHỤ LỤC</b>	<b>63</b>

## DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

<b>Từ viết tắt</b>	<b>Thay cho</b>
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

<b>Hình 2.1</b>	<b>Sơ đồ ERD . . . . .</b>	<b>18</b>
<b>Hình 2.2</b>	<b>Mô hình thực thể liên kết . . . . .</b>	<b>22</b>
<b>Hình 3.1</b>	<b>Kiến trúc tổng quan hệ thống . . . . .</b>	<b>23</b>
<b>Hình 3.2</b>	<b>Sơ đồ khối Website dành cho bệnh nhân . . . . .</b>	<b>25</b>
<b>Hình 3.3</b>	<b>Sơ đồ khối Website dành cho bác sĩ . . . . .</b>	<b>26</b>
<b>Hình 3.4</b>	<b>Sơ đồ khối Website dành cho quản trị viên . . . . .</b>	<b>27</b>
<b>Hình 3.5</b>	<b>Sơ đồ ERD . . . . .</b>	<b>31</b>
<b>Hình 4.1</b>	<b>Kiến trúc của NodeJS . . . . .</b>	<b>33</b>
<b>Hình 4.2</b>	<b>Docker . . . . .</b>	<b>35</b>

## DANH MỤC BẢNG BIỂU

<b>Bảng 2.1</b>	<b>Thẻ CRC lớp Tài khoản người dùng . . . . .</b>	<b>4</b>
<b>Bảng 2.2</b>	<b>Thẻ CRC lớp Token . . . . .</b>	<b>5</b>
<b>Bảng 2.3</b>	<b>Thẻ CRC lớp Vai trò người dùng . . . . .</b>	<b>5</b>
<b>Bảng 2.4</b>	<b>Thẻ CRC lớp Trạng thái người dùng . . . . .</b>	<b>6</b>
<b>Bảng 2.5</b>	<b>Thẻ CRC lớp Người dùng . . . . .</b>	<b>7</b>
<b>Bảng 2.6</b>	<b>Thẻ CRC lớp Loại thiết bị . . . . .</b>	<b>8</b>
<b>Bảng 2.7</b>	<b>Thẻ CRC lớp Trạng thái thiết bị . . . . .</b>	<b>8</b>
<b>Bảng 2.8</b>	<b>Thẻ CRC lớp Thiết bị . . . . .</b>	<b>9</b>
<b>Bảng 2.9</b>	<b>Thẻ CRC lớp Thông số kỹ thuật . . . . .</b>	<b>10</b>
<b>Bảng 2.10</b>	<b>Thẻ CRC lớp Dữ liệu phiên đo . . . . .</b>	<b>10</b>
<b>Bảng 2.11</b>	<b>Thẻ CRC lớp Loại lịch hẹn . . . . .</b>	<b>11</b>
<b>Bảng 2.12</b>	<b>Thẻ CRC lớp Trạng thái lịch hẹn . . . . .</b>	<b>12</b>
<b>Bảng 2.13</b>	<b>Thẻ CRC lớp Lịch hẹn . . . . .</b>	<b>13</b>
<b>Bảng 2.14</b>	<b>Thẻ CRC lớp Lịch hẹn của bác sĩ . . . . .</b>	<b>13</b>
<b>Bảng 2.15</b>	<b>Thẻ CRC lớp Thông báo liên quan đến lịch hẹn . . . . .</b>	<b>14</b>
<b>Bảng 2.16</b>	<b>Thẻ CRC lớp Chẩn đoán cho bệnh nhân . . . . .</b>	<b>15</b>
<b>Bảng 2.17</b>	<b>Thẻ CRC lớp Tin nhắn . . . . .</b>	<b>16</b>
<b>Bảng 2.18</b>	<b>Thẻ CRC lớp Nhóm trò chuyện . . . . .</b>	<b>16</b>
<b>Bảng 3.1</b>	<b>Bảng chuẩn hoá bảng Người dùng . . . . .</b>	<b>28</b>
<b>Bảng 3.2</b>	<b>Bảng chuẩn hoá bảng Tài khoản . . . . .</b>	<b>28</b>
<b>Bảng 3.3</b>	<b>Bảng chuẩn hoá bảng Thiết bị . . . . .</b>	<b>28</b>
<b>Bảng 3.4</b>	<b>Bảng chuẩn hoá bảng Thông số thiết bị . . . . .</b>	<b>29</b>
<b>Bảng 3.5</b>	<b>Bảng chuẩn hoá bảng Bản ghi dữ liệu . . . . .</b>	<b>29</b>
<b>Bảng 3.6</b>	<b>Bảng user . . . . .</b>	<b>30</b>
<b>Bảng 3.7</b>	<b>Bảng account . . . . .</b>	<b>30</b>
<b>Bảng 3.8</b>	<b>Bảng device . . . . .</b>	<b>30</b>
<b>Bảng 3.9</b>	<b>Bảng device_detail . . . . .</b>	<b>31</b>

<b>Bảng 4.1</b>	<b>Bảng kiểm thử API đăng ký tài khoản . . . . .</b>	<b>37</b>
<b>Bảng 4.2</b>	<b>Bảng kiểm thử API đăng nhập . . . . .</b>	<b>38</b>
<b>Bảng 4.3</b>	<b>Bảng kiểm thử API đăng xuất . . . . .</b>	<b>38</b>
<b>Bảng 4.4</b>	<b>Bảng kiểm thử API gửi token đặt lại mật khẩu . . . . .</b>	<b>39</b>
<b>Bảng 4.5</b>	<b>Bảng kiểm thử API đặt lại mật khẩu . . . . .</b>	<b>39</b>
<b>Bảng 4.6</b>	<b>Bảng kiểm thử API đăng ký tài khoản . . . . .</b>	<b>41</b>
<b>Bảng 4.7</b>	<b>Bảng kiểm thử API chấp nhận tài khoản . . . . .</b>	<b>41</b>
<b>Bảng 4.8</b>	<b>Bảng kiểm thử API từ chối phê duyệt tài khoản . . . . .</b>	<b>42</b>
<b>Bảng 4.9</b>	<b>Bảng kiểm thử API lấy danh sách người dùng . . . . .</b>	<b>43</b>
<b>Bảng 4.10</b>	<b>Bảng kiểm thử API lấy thông tin của người dùng thông qua ID . . .</b>	<b>43</b>
<b>Bảng 4.11</b>	<b>Bảng kiểm thử API lấy thông tin của người dùng thông qua chức vụ</b>	<b>44</b>
<b>Bảng 4.12</b>	<b>Bảng kiểm thử API cập nhật thông tin người dùng . . . . .</b>	<b>44</b>
<b>Bảng 4.13</b>	<b>Bảng kiểm thử API xóa thông tin người dùng . . . . .</b>	<b>45</b>
<b>Bảng 4.14</b>	<b>Bảng kiểm thử API lấy danh sách thiết bị . . . . .</b>	<b>46</b>
<b>Bảng 4.15</b>	<b>Bảng kiểm thử API lấy thông tin thiết bị theo ID . . . . .</b>	<b>46</b>
<b>Bảng 4.16</b>	<b>Bảng kiểm thử API thêm thiết bị . . . . .</b>	<b>47</b>
<b>Bảng 4.17</b>	<b>Bảng kiểm thử API cập nhật thông tin thiết bị . . . . .</b>	<b>48</b>
<b>Bảng 4.18</b>	<b>Bảng kiểm thử API xóa thiết bị . . . . .</b>	<b>49</b>
<b>Bảng 4.19</b>	<b>Bảng kiểm thử API lấy danh sách bản ghi . . . . .</b>	<b>50</b>
<b>Bảng 4.20</b>	<b>Bảng kiểm thử API lấy danh sách phiên đo ECG của bệnh nhân . .</b>	<b>51</b>
<b>Bảng 4.21</b>	<b>Bảng kiểm thử API lấy danh sách phiên đo ECG mà bác sĩ phụ trách</b>	<b>51</b>
<b>Bảng 4.22</b>	<b>Bảng kiểm thử API lấy thông tin một phiên đo ECG . . . . .</b>	<b>52</b>
<b>Bảng 4.23</b>	<b>Bảng kiểm thử API lấy dữ liệu bản ghi một phiên đo ECG . . . . .</b>	<b>52</b>
<b>Bảng 4.24</b>	<b>Bảng kiểm thử API tải dữ liệu đo ECG . . . . .</b>	<b>53</b>
<b>Bảng 4.25</b>	<b>Bảng kiểm thử API cập nhật thông tin phiên đo . . . . .</b>	<b>54</b>
<b>Bảng 4.26</b>	<b>Bảng kiểm thử API xóa thông tin một phiên đo ECG . . . . .</b>	<b>55</b>
<b>Bảng 4.27</b>	<b>Bảng kiểm thử API lấy danh sách phân công bác sĩ - bệnh nhân . .</b>	<b>56</b>
<b>Bảng 4.28</b>	<b>Bảng kiểm thử API tạo một phân công bác sĩ - bệnh nhân . . . . .</b>	<b>57</b>
<b>Bảng 4.29</b>	<b>Bảng kiểm thử API cập nhật thông tin phân công . . . . .</b>	<b>57</b>

<b>Bảng 4.30</b>	<b>Bảng kiểm thử API xóa thông tin phân công . . . . .</b>	<b>58</b>
<b>Bảng 4.31</b>	<b>Bảng kiểm thử API lấy danh sách bệnh nhân mà bác sĩ đang quản lý theo ID bác sĩ . . . . .</b>	<b>59</b>
<b>Bảng 4.32</b>	<b>Bảng kiểm thử API lấy thông tin bác sĩ đang quản lý bệnh nhân theo ID bệnh nhân . . . . .</b>	<b>59</b>
<b>Bảng 4.33</b>	<b>Bảng kiểm thử chức năng của website quản trị . . . . .</b>	<b>60</b>

# TÓM TẮT ĐỒ ÁN

Đồ án "Phát triển ứng dụng di động tích hợp trong hệ thống quản lý dữ liệu tim mạch, hỗ trợ kết nối giữa bệnh nhân - bác sĩ" là một hệ thống tích hợp bao gồm Web/App/Server, sử dụng công nghệ Bluetooth Low Energy để hỗ trợ việc theo dõi và quản lý dữ liệu điện tim trên ứng dụng di động một cách hiệu quả và tiện lợi. Dữ liệu sau mỗi phiên đo được lưu trữ trên server, cho phép người dùng xem lại bất kỳ lúc nào và phục vụ cho các mục đích nghiên cứu trong tương lai. Bên cạnh đó, hệ thống còn cung cấp các tính năng hỗ trợ kết nối trực tiếp giữa bệnh nhân và bác sĩ, như đặt lịch hẹn, nhắn tin trao đổi thông tin và theo dõi dữ liệu sức khỏe, mang lại sự tiện ích và chuyên nghiệp trong quản lý sức khỏe.

Trong đồ án, nhóm tập trung hoàn thiện hệ thống theo quy trình phát triển phần mềm, từ việc xác định các yêu cầu hệ thống đến phân tích, thiết kế và triển khai. Quy trình này áp dụng phương pháp phân tích và thiết kế hướng đối tượng, sử dụng ngôn ngữ UML để biểu diễn các luồng thực hiện hành động. Về mặt kỹ thuật, ứng dụng di động được phát triển trên hệ điều hành Android và iOS sử dụng Flutter Framework, trong khi ứng dụng Web được xây dựng với ReactJS và AdminJS. Server của hệ thống sử dụng NodeJS, framework NestJS, và được triển khai trên Amazon EC2. Hệ thống sử dụng cơ sở dữ liệu MySQL kết hợp với NoSQL (MongoDB) để đảm bảo hiệu năng và khả năng lưu trữ linh hoạt.

Quyển đồ án được trình bày theo quy trình phát triển phần mềm, với nội dung được triển khai chi tiết trong từng chương, bao gồm: phân tích hệ thống, thiết kế hệ thống, triển khai và kiểm thử, và cuối cùng là phần kết luận. Tất cả các nội dung được minh họa bằng sơ đồ và các diễn giải cụ thể, đảm bảo tính rõ ràng và khoa học.

## ABSTRACT

The project "Development of a Mobile Application Integrated into a Cardiovascular Data Management System, Supporting Patient-Doctor Connectivity" is an integrated system comprising Web/App/Server, leveraging Bluetooth Low Energy (BLE) technology to facilitate effective and convenient monitoring of electrocardiogram (ECG) data through a mobile application. The data collected after each measurement session is stored on the server, enabling users to review it at any time and supporting future research purposes. Furthermore, the system provides features that enhance direct connectivity between patients and doctors, including appointment scheduling, messaging for information exchange, and health data monitoring, delivering convenience and professionalism in healthcare management.

This project focuses on completing the system following a software development lifecycle (SDLC), starting from identifying system requirements to analyzing, designing, and implementing the solution. The development process applies object-oriented analysis and design (OOAD) methodologies, utilizing Unified Modeling Language (UML) to represent action flows. From a technical perspective, the mobile application is developed for both Android and iOS platforms using the Flutter framework. The web application is built with ReactJS and AdminJS, while the server utilizes NodeJS and the NestJS framework, deployed on Amazon EC2. The system employs MySQL in combination with NoSQL (MongoDB) databases to ensure both performance and flexible data storage capabilities.

The project documentation is structured according to the software development lifecycle, with detailed content presented in individual chapters, including system analysis, system design, implementation, testing, and finally, conclusions. All content is illustrated with diagrams and comprehensive explanations, ensuring clarity and scientific rigor.

# 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 ứng dụng di động cho bệnh nhân có thể xem được dữ liệu điện tim của bản thân, đặt lịch và nhắn tin với bác sĩ của mình
- Một ứng dụng di động cho bác sĩ để có thể kiểm soát, phản hồi lịch hẹn, xem được kết quả đo của các bệnh nhân được quản lý, nhắn tin với bệnh nhân, nhóm chat
- 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 tim 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 tim 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 kĩ 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 thẻ 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 Phân tích 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 ứng dụng di động. Bệnh nhân có thể đăng nhập vào tài khoản để truy cập thông tin cá nhân, xem danh sách bác sĩ, thông tin các bản ghi điện tim đã đo, đồng thời tìm kiếm và chọn lịch hẹn phù hợp để đăng ký. Ngoài ra, hệ thống còn cung cấp tính năng nhắn tin trực tiếp, giúp bệnh nhân dễ dàng trao đổi và nhận tư vấn từ bác sĩ.
- Bác sĩ: Hệ thống hỗ trợ bác sĩ quản lý và theo dõi tình trạng sức khỏe của các bệnh nhân đã đặt lịch hẹn thành công. Bác sĩ được phép truy cập kết quả các phiên đo, cung cấp tư vấn, trao đổi thông tin y tế, và chủ động đặt lịch tái khám khi cần. Ngoài ra, bác sĩ có thể tương tác với bệnh nhân qua tính năng nhắn tin hoặc nhóm chat để thảo luận và chia sẻ thông tin chuyên môn.

### 1.1.2 Yêu cầu chức năng của hệ thống

Các chức năng chính của hệ thống bao gồm:

- Ghi dữ liệu đo từ thiết bị: Ứng dụng trên điện thoại thu nhận dữ liệu đo điện tim từ thiết bị thông qua kết nối Bluetooth. Dữ liệu này được lưu trữ dưới dạng file định dạng .csv và đồng bộ trực tiếp lên máy chủ hệ thống, đảm bảo an toàn và sẵn sàng cho việc truy cập, phân tích sau này.
- Hiển thị dữ liệu: Các số liệu được lưu trữ trên máy chủ sẽ được xử lý và tính toán theo các công thức chuyên môn, sau đó được hiển thị trực quan trên website và ứng dụng di động dưới dạng đồ thị đường, giúp bác sĩ dễ dàng theo dõi và phân tích các chỉ số sức khỏe.
- Lưu trữ dữ liệu: Hệ thống hỗ trợ lưu trữ dữ liệu đo điện tim từ thiết bị trên cả ứng dụng di động và máy chủ trung tâm. Dữ liệu được đồng bộ hóa tự động từ ứng dụng lên máy chủ, đảm bảo an toàn và bảo mật tuyệt đối. Việc lưu trữ song

song này giúp bảo vệ dữ liệu quan trọng, đảm bảo tính toàn vẹn và khả năng truy cập cho mục đích phân tích hoặc sử dụng trong tương lai, mang lại sự tin cậy cao cho người dùng.

- Trao đổi và chia sẻ thông tin về dữ liệu y tế: Hệ thống giúp người dùng có thể trao đổi trực tiếp với bác sĩ, chia sẻ kết quả đo điện tim, hỏi đáp về các vấn đề sức khỏe, và các vấn đề liên quan đến thiết bị. Điều này mang lại sự tiện lợi và hỗ trợ đáng kể cho người dùng trong việc xác định 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 các phiên đo của mình, bao gồm biểu đồ và các thông số liên quan.
- Nhận thông báo và có thể trao đổi trực tiếp với bác sĩ về tình hình sức khỏe và các kết quả đo được từ thiết bị.

#### **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 đã đặt lịch hẹn thành công.
- Nhận thông báo và có thể trao đổi trực tiếp với bệnh nhân về tình hình sức khỏe và các kết quả đo được từ thiết bị.

### **1.1.3 Yêu cầu phi chức năng của hệ thống**

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

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

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

## CHƯƠNG 2. TÌM HIỂU VÀ PHÂN TÍCH HỆ THỐNG

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

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

### 2.1 Thẻ CRC (Class - Responsibility - Collaboration Card)

#### 2.1.1 Thẻ CRC lớp Tài khoản người dùng

**Bảng 2.1 Thẻ CRC lớp Tài khoản người dùng**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.2 Thẻ CRC lớp Token

**Bảng 2.2 Thẻ CRC lớp Token**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.3 Thẻ CRC lớp Vai trò người dùng

**Bảng 2.3 Thẻ CRC lớp Vai trò người dùng**

Mặt trước thẻ

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

Mặt sau thẻ

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

#### 2.1.4 Thẻ CRC lớp Trạng thái người dùng

**Bảng 2.4 Thẻ CRC lớp Trạng thái người dùng**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.5 Thẻ CRC lớp Người dùng

**Bảng 2.5 Thẻ CRC lớp Người dùng**

Mặt trước thẻ

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

Mặt sau thẻ

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



### 2.1.6 Thẻ CRC lớp Loại thiết bị

**Bảng 2.6 Thẻ CRC lớp Loại thiết bị**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.7 Thẻ CRC lớp Trạng thái thiết bị

**Bảng 2.7 Thẻ CRC lớp Trạng thái thiết bị**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.8 Thẻ CRC lớp Thiết bị

**Bảng 2.8 Thẻ CRC lớp Thiết bị**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.9 Thẻ CRC lớp Thông số kỹ thuật

**Bảng 2.9 Thẻ CRC lớp Thông số kỹ thuật**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.10 Thẻ CRC lớp Dữ liệu phiên đo

**Bảng 2.10 Thẻ CRC lớp Dữ liệu phiên đo**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.11 Thẻ CRC lớp Loại lịch hẹn

**Bảng 2.11 Thẻ CRC lớp Loại lịch hẹn**

Mặt trước thẻ

<b>Tên lớp:</b> Loại lịch hẹn (schedule_type)	<b>ID:</b> 10
<b>Mô tả:</b> Lớp quản lý thông tin các loại lịch hẹn có thể đặt	<b>Use case liên quan:</b> Quản lý lịch hẹn
<b>Trách nhiệm (Responsibility):</b> Lưu trữ danh sách các loại lịch hẹn khả dụng (ví dụ: Lịch khám, Lịch tư vấn thiết bị) Cung cấp thông tin các loại lịch hẹn cho các chức năng liên quan đến lập lịch	<b>Các lớp cộng tác (Collaboration):</b> Lịch hẹn

Mặt sau thẻ

<b>Thuộc tính (Attributes):</b> id(uuid) type_name(string)	createdAt(Datetime) updatedAt(Datetime)
<b>Mối quan hệ (Relationships)</b> Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Lịch hẹn	

### 2.1.12 Thẻ CRC lớp Trạng thái lịch hẹn

**Bảng 2.12 Thẻ CRC lớp Trạng thái lịch hẹn**

Mặt trước thẻ

<b>Tên lớp:</b> Trạng thái lịch hẹn (schedule_status)	<b>ID:</b> 11
<b>Mô tả:</b> Lớp quản lý thông tin các trạng thái lịch hẹn hiện tại trong hệ thống	<b>Use case liên quan:</b> Quản lý lịch hẹn
<b>Trách nhiệm (Responsibility):</b> Lưu trữ danh sách các trạng thái lịch hẹn (ví dụ: Thành công (được bác sĩ chấp nhận), Thất bại (bị từ chối), Đang chờ xác nhận) Cung cấp thông tin các trạng thái lịch hẹn để hỗ trợ các chức năng liên quan đến quản lý và hiển thị lịch hẹn	<b>Các lớp cộng tác (Collaboration):</b> Lịch hẹn

Mặt sau thẻ

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

### 2.1.13 Thẻ CRC lớp Lịch hẹn

**Bảng 2.13 Thẻ CRC lớp Lịch hẹn**

Mặt trước thẻ

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

Mặt sau thẻ

<b>Thuộc tính (Attributes):</b> id(uuid) patient_id(string) schedule_start_time(timestamps) schedule_end_time(timestamps) schedule_type_id(integer)	status_id(integer) schedule_result(integer) createdAt(Datetime) updatedAt(Datetime)
<b>Mối quan hệ (Relationships)</b> Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Loại lịch hẹn, Trạng thái lịch hẹn, Lịch hẹn của bác sĩ, Người dùng, Chẩn đoán cho bệnh nhân	

#### 2.1.14 Thẻ CRC lớp Lịch hẹn của bác sĩ

**Bảng 2.14 Thẻ CRC lớp Lịch hẹn của bác sĩ**

Mặt trước thẻ

<b>Tên lớp:</b> Lịch hẹn của bác sĩ (consultation_schedule)	<b>ID:</b> 13
<b>Mô tả:</b> Lớp quản lý thông tin các lịch hẹn của bác sĩ trong hệ thống	<b>Use case liên quan:</b> Quản lý lịch hẹn
<b>Trách nhiệm (Responsibility):</b> Lưu trữ danh sách các lịch hẹn với ID bác sĩ tương ứng Truy vấn danh sách lịch hẹn theo ID bác sĩ	<b>Các lớp cộng tác (Collaboration):</b> Lịch hẹn Người dùng

Mặt sau thẻ

<b>Thuộc tính (Attributes):</b> id(uuid) schedule_id(string) doctor_id(string)	createdAt(Datetime) updatedAt(Datetime)
<b>Mối quan hệ (Relationships)</b> Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Người dùng, Lịch hẹn	

### 2.1.15 Thẻ CRC lớp Thông báo liên quan đến lịch hẹn

**Bảng 2.15 Thẻ CRC lớp Thông báo liên quan đến lịch hẹn**

Mặt trước thẻ

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

Mặt sau thẻ

<b>Thuộc tính (Attributes):</b> id(uuid) patient_id(string) doctor_id(string) schedule_start_time(timestamps) status(integer)	reject_reason(string) type(integer) is_seen(boolean) createdAt(Datetime) updatedAt(Datetime)
<b>Mối quan hệ (Relationships)</b> Tổng quát hóa (Generalize): Toàn thể - Bộ phận (Aggregation): Liên kết (Association): Người dùng	

## 2.1.16 Thẻ CRC lớp Chẩn đoán cho bệnh nhân

**Bảng 2.16 Thẻ CRC lớp Chẩn đoán cho bệnh nhân**

Mặt trước thẻ

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

Mặt sau thẻ

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



### 2.1.17 Thẻ CRC lớp Tin nhắn

**Bảng 2.17 Thẻ CRC lớp Tin nhắn**

Mặt trước thẻ

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

Mặt sau thẻ

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

### 2.1.18 Thẻ CRC lớp Nhóm trò chuyện

**Bảng 2.18 Thẻ CRC lớp Nhóm trò chuyện**

Mặt trước thẻ

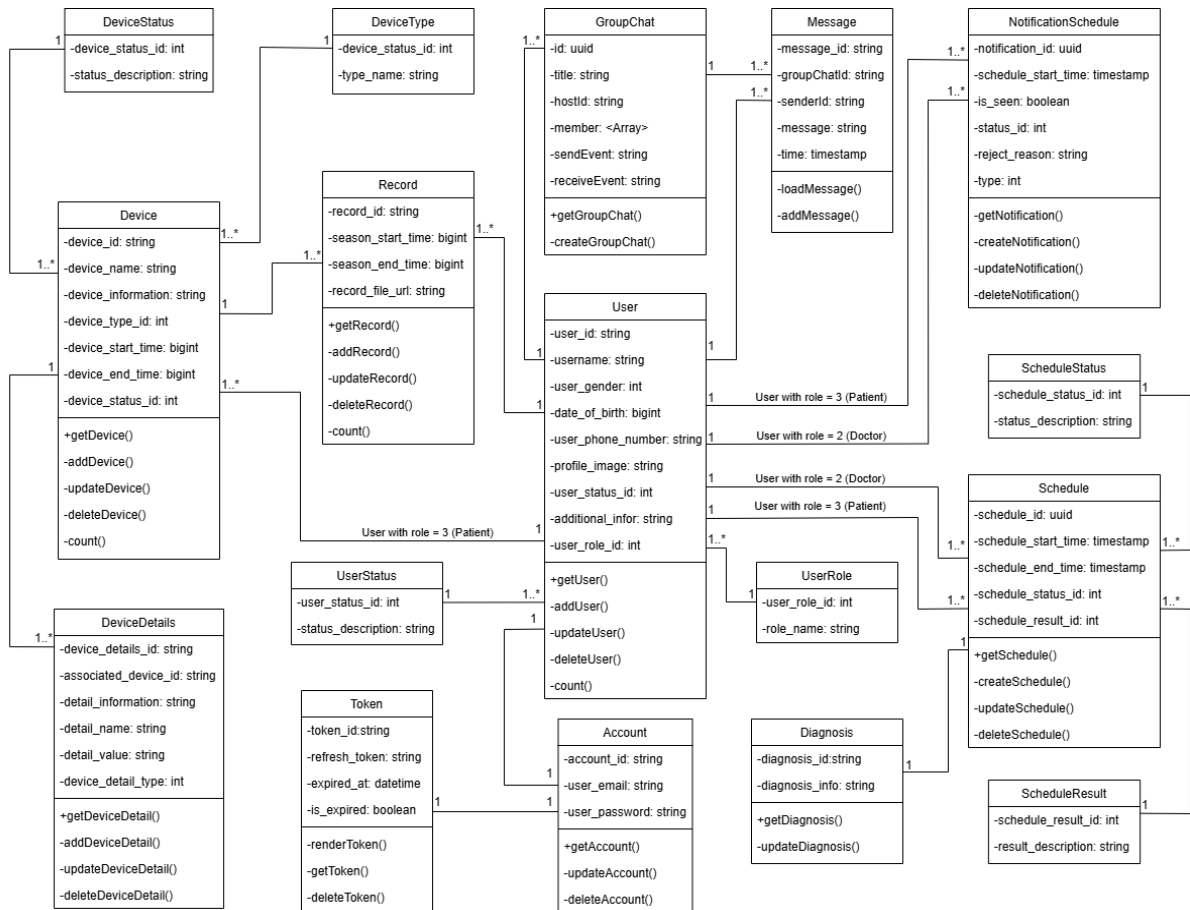
<b>Tên lớp:</b> Nhóm trò chuyện (group_chat)	<b>ID:</b> 17
<b>Mô tả:</b> Lớp quản lý thông tin chi tiết các nhóm trò chuyện có trong hệ thống	<b>Use case liên quan:</b> Quản lý dịch vụ nhắn tin
<b>Trách nhiệm (Responsibility):</b> Lưu trữ danh sách các nhóm trò chuyện trong hệ thống Truy vấn danh sách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

Mặt sau thẻ

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

## 2.2 Sơ đồ lớp

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



**Hình 2.1 Sơ đồ UML**

Hình 2.1 thể hiện các lớp trong hệ thống:

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

của người dùng trong hệ thống.

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

### ***2.3 Sơ đồ tuần tự***

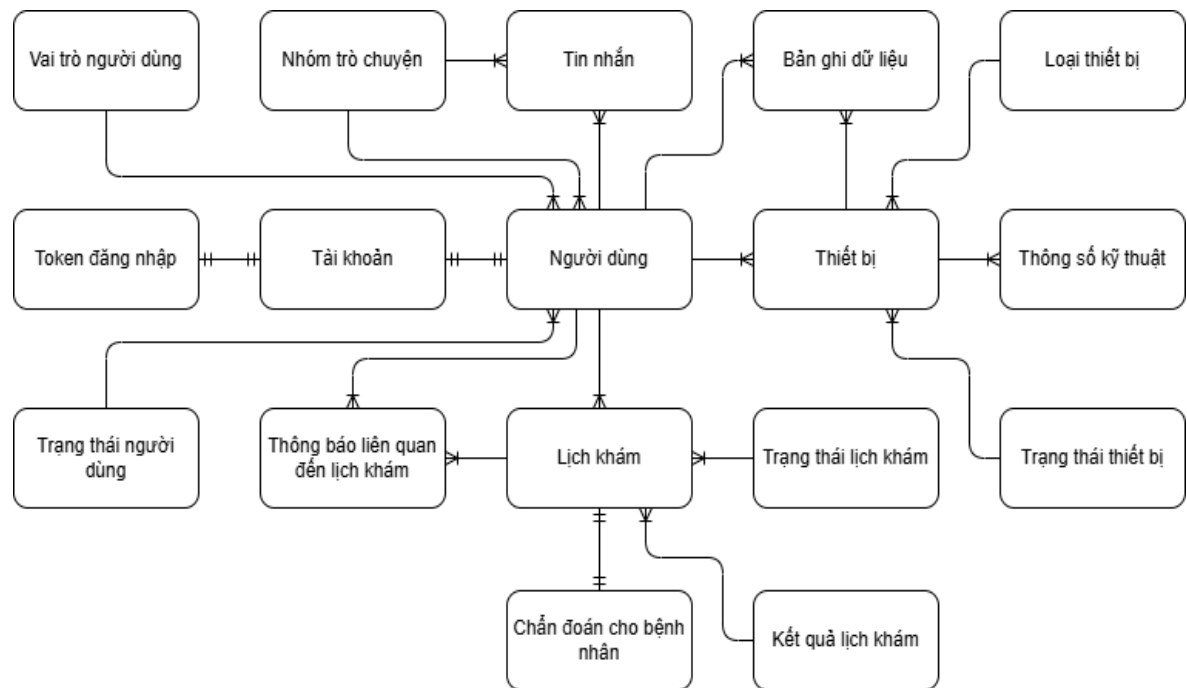
### ***2.4 Xử lý và phân tích dữ liệu***

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

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

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

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



**Hình 2.2** Mô hình thực thể liên kết

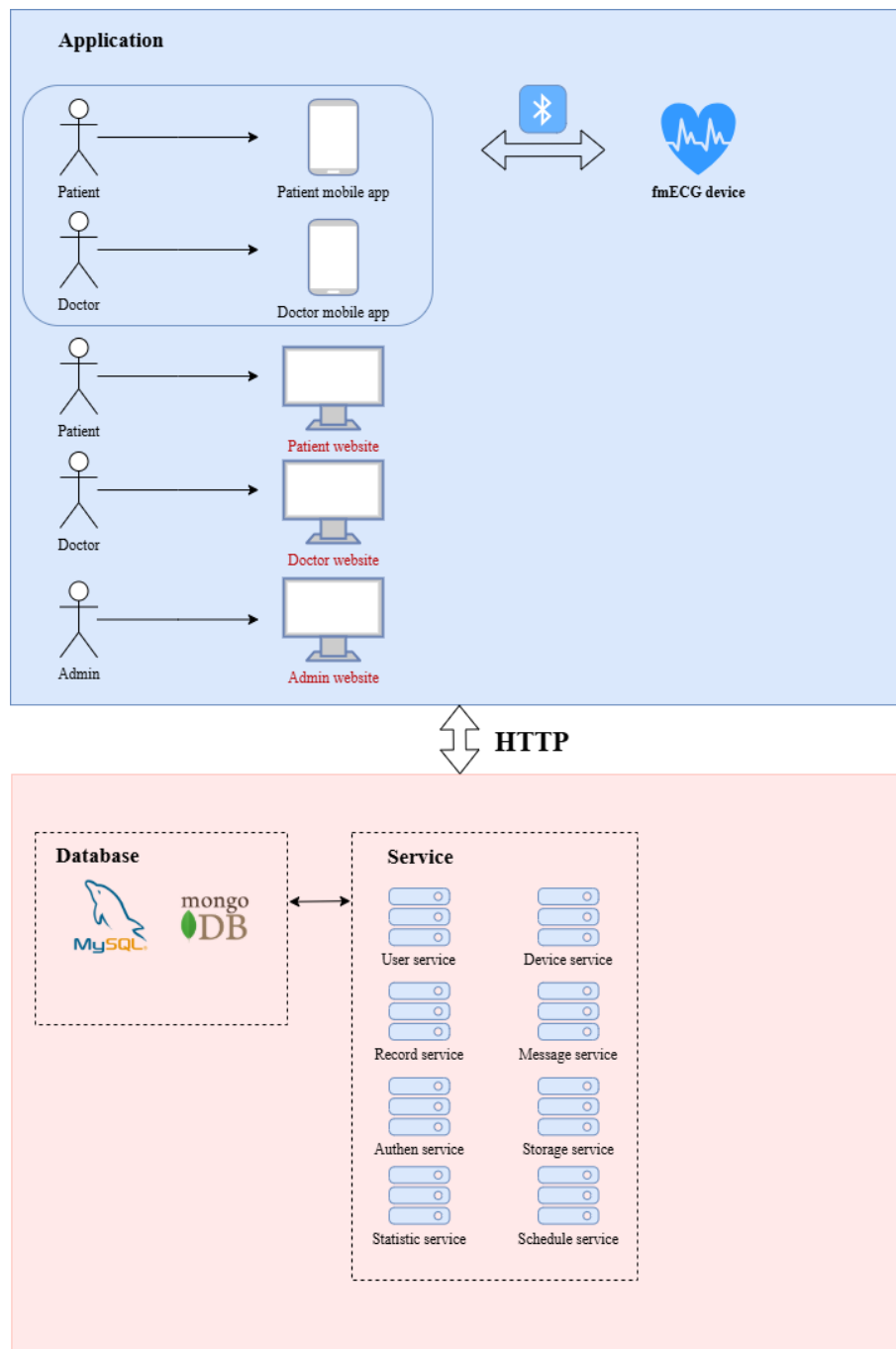
## 2.5 Kết luận

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

## CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

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 và website cho bệnh nhân, bác sĩ, quản trị viên.
- Server (Máy chủ): Bao gồm các Services để xử lý các yêu cầu gửi từ Application, cơ sở dữ liệu.

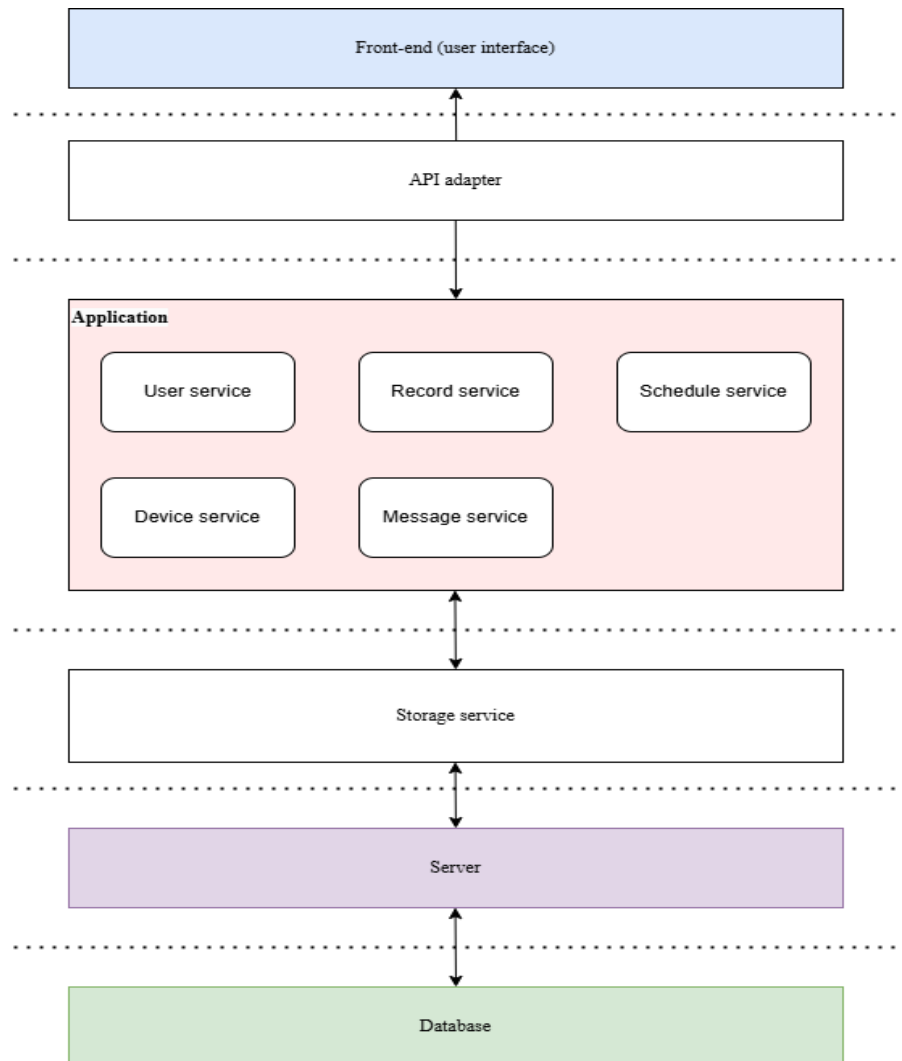
Hệ thống được thiết kế với sự tương tác trực tiếp giữa bệnh nhân và thiết bị thông qua ứng dụng di động. Ứng dụng đóng vai trò trung gian, giao tiếp với máy chủ bằng API và giao thức HTTP. Máy chủ chịu trách nhiệm xử lý dữ liệu thông qua các dịch vụ chuyên biệt. Mỗi dịch vụ được thiết kế để xử lý một 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 thêm, sửa, xóa, xử lý dữ liệu file đo.
- 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**

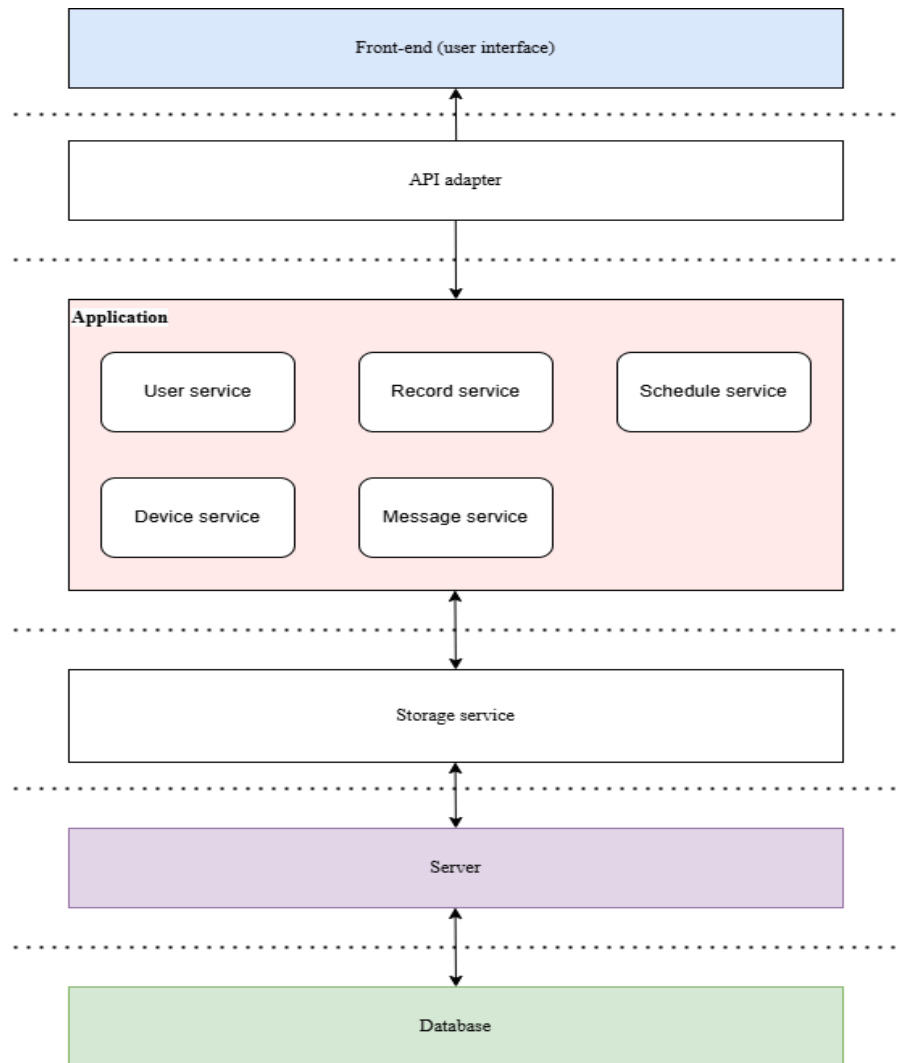
Lớp trên cùng trong sơ đồ hình 3.2 là lớp giao diện người dùng, nơi người dùng trực tiếp tương tác với hệ thống, thực hiện các request thông qua API Adapter.

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

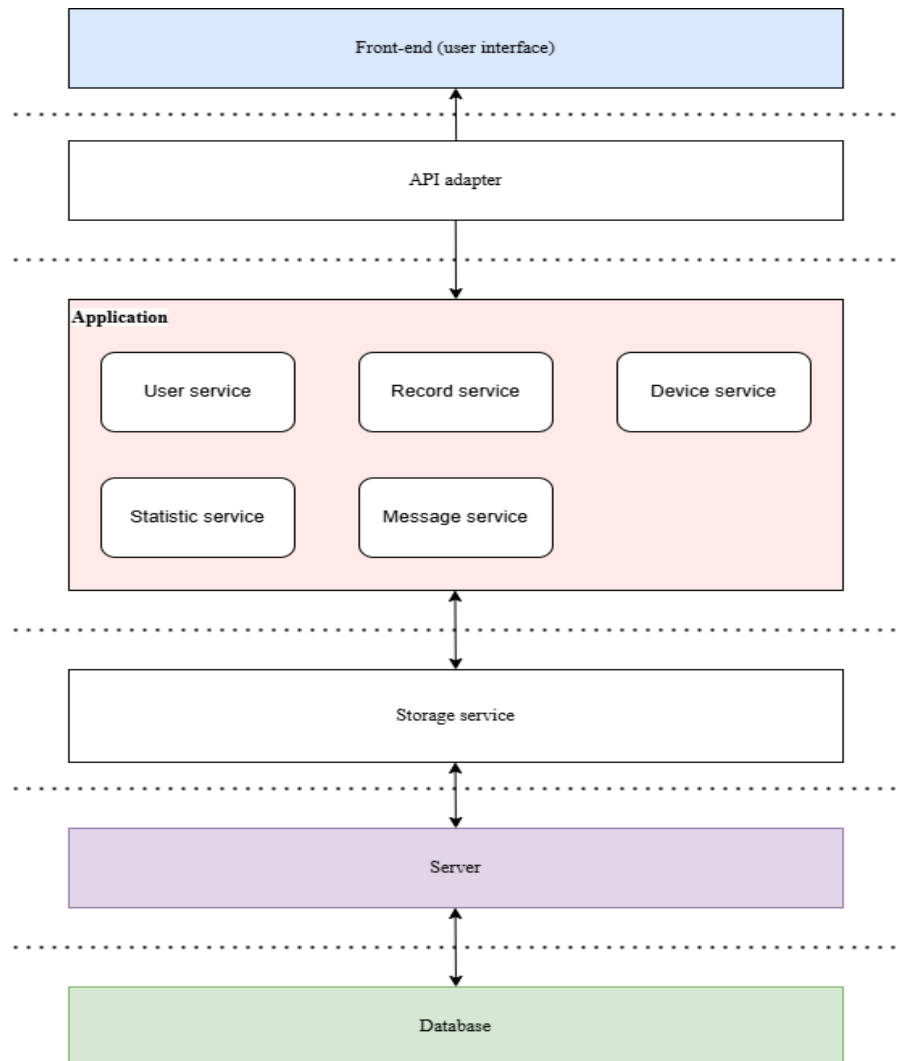


**Hình 3.3 Sơ đồ khối Website dành cho bác sĩ**

- 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.

## 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**

<b>Danh sách thuộc tính</b>	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
<b>Quy tắc nghiệp vụ</b>	<b>Phụ thuộc hàm</b>
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**

<b>Danh sách thuộc tính</b>	ID tài khoản, Email, Mật khẩu
<b>Quy tắc nghiệp vụ</b>	<b>Phụ thuộc hàm</b>
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 Thiết bị

**Bảng 3.3 Bảng chuẩn hoá bảng Thiết bị**

<b>Danh sách thuộc tính</b>	ID thiết bị, ID người dùng thiết bị, Tên thiết bị, Loại thiết bị, Thông tin thiết bị, Trạng thái thiết bị, Ngày bắt đầu sử dụng, Ngày kết thúc sử dụng
<b>Quy tắc nghiệp vụ</b>	<b>Phụ thuộc hàm</b>
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, trạng thái thiết bị, ngày bắt đầu sử dụng, ngày kết thúc sử dụng	ID thiết bị → ID người dùng thiết bị, Tên thiết bị, Loại thiết bị, Thông tin thiết bị, Trạng thái thiết bị, Ngày bắt đầu sử dụng, Ngày kết thúc sử dụng
⇒ Khoá chính của bảng: ID thiết bị ⇒ Bảng Thiết bị đã ở 3NF	

#### 3.3.1.4 Chuẩn hoá bảng Thông số thiết bị

**Bảng 3.4 Bảng chuẩn hoá bảng Thông số thiết bị**

<b>Danh sách thuộc tính</b>	ID thông số, ID thiết bị, Tên thông số, Giá trị, Thông tin thông số, Loại thông số
<b>Quy tắc nghiệp vụ</b>	<b>Phụ thuộc hàm</b>
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	

#### 3.3.1.5 Chuẩn hoá bảng Bản ghi dữ liệu

**Bảng 3.5 Bảng chuẩn hoá bảng Bản ghi dữ liệu**

<b>Danh sách thuộc tính</b>	ID bản ghi dữ liệu, ID người dùng, ID thiết bị, Đường dẫn lưu trữ dữ liệu, Thời gian bắt đầu đo, Thời gian kết thúc đo
<b>Quy tắc nghiệp vụ</b>	<b>Phụ thuộc hàm</b>
Mỗi bản ghi có một ID bản ghi dữ liệu riêng, có duy nhất ID người dùng, ID thiết bị, 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, 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	

### 3.3.2 Từ điển dữ liệu

**Bảng 3.6 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 3.7 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.

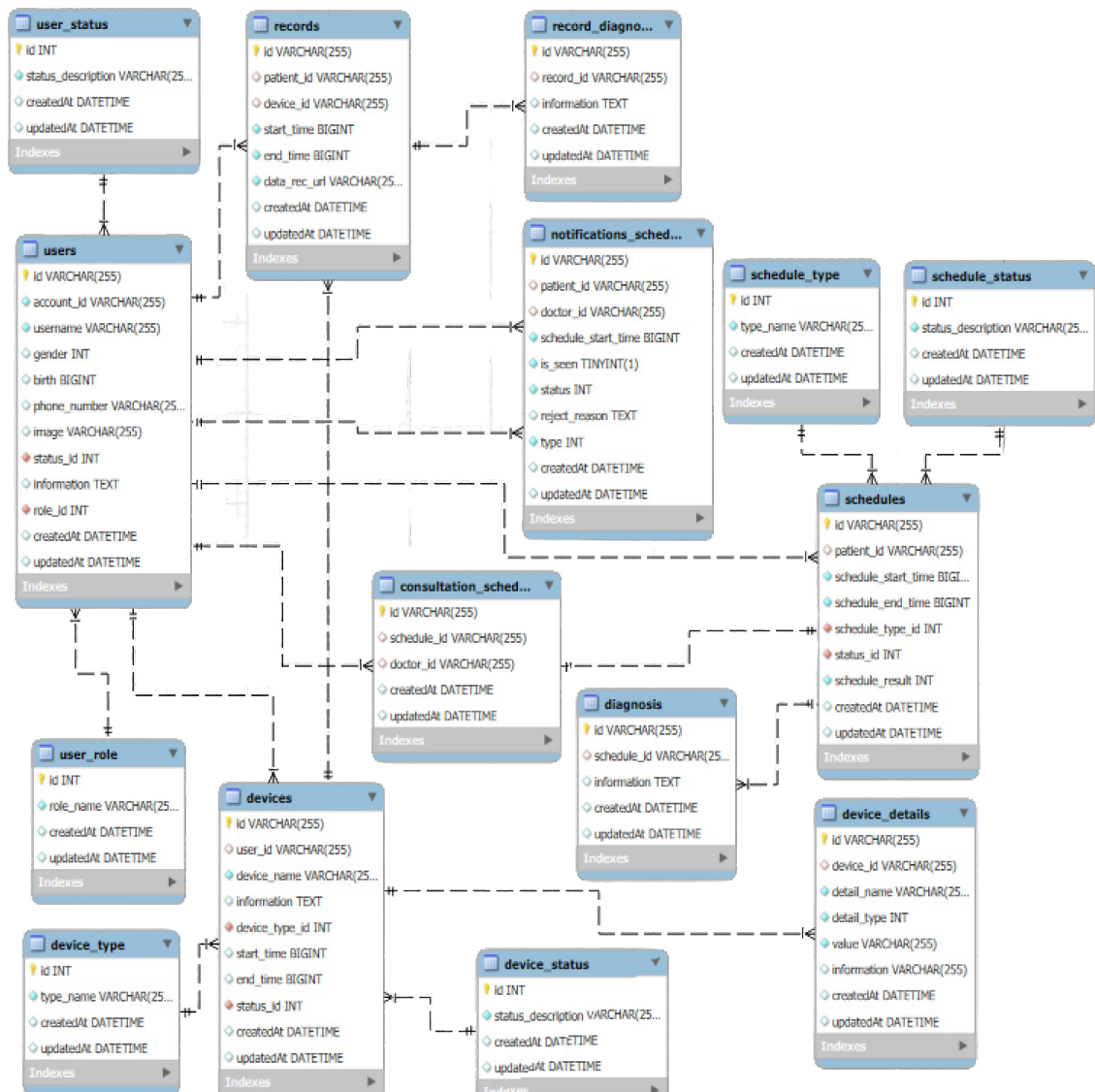
**Bảng 3.8 Bảng device**

Thuộc tính	Kiểu dữ liệu	Mô tả
id	STRING	ID thiết bị
user_id	STRING	ID người dùng
device_name	STRING	Tên thiết bị.
information	STRING	Thông tin thêm về thiết bị.
device_type_id	INTEGER	Loại thiết bị (1 - thiết bị đo điện tim, 2 - thiết bị đo nhịp tim, âm thanh tim).
start_time	DATE	Thời gian lúc bắt đầu sử dụng thiết bị.
end_time	DATE	Thời gian lúc kết thúc 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 3.9 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.

### 3.3.3 Sơ đồ ERD



**Hình 3.5 Sơ đồ ERD**



### **3.4 *Thiết kế giao diện***

### **3.5 *Thiết kế các chức năng cho website và server***

#### **3.5.1 *Thiết kế 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.[20]

#### 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.[?]

#### 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	<ol style="list-style-type: none"> <li>1. Vào màn hình quản lý người dùng</li> <li>2. Xem thông tin người dùng</li> <li>3. Sửa thông tin người dùng có sẵn → Lưu thay đổi</li> <li>4. Xóa một người dùng khỏi hệ thống</li> </ol>	<ol style="list-style-type: none"> <li>1. Hiện thị danh sách người dùng</li> <li>2. Hiện thị thông tin cụ thể của người dùng</li> <li>3. Cập nhật thông tin người dùng thành công</li> <li>4. Xóa tài khoản người dùng khỏi hệ thống thành công</li> </ol>	OK
Kiểm tra chức năng xem danh sách bác sĩ	<ol style="list-style-type: none"> <li>1. Vào màn hình danh sách bác sĩ phụ trách</li> <li>2. Xem thông tin cụ thể của bác sĩ</li> </ol>	<ol style="list-style-type: none"> <li>1. Hiện thị danh sách bác sĩ</li> <li>2. Hiện thị thông tin cụ thể bác sĩ</li> </ol>	OK
Kiểm tra chức năng xem danh sách bệnh nhân	<ol style="list-style-type: none"> <li>1. Vào màn hình quản lý bệnh nhân</li> <li>2. Hiện thị thông tin cụ thể bệnh nhân</li> </ol>	<ol style="list-style-type: none"> <li>1. Hiện thị danh sách bệnh nhân</li> <li>2. Hiện thị thông tin cụ thể bệnh nhân</li> </ol>	OK
Kiểm tra chức năng quản lý thiết bị	<ol style="list-style-type: none"> <li>1. Vào màn hình quản lý thiết bị</li> <li>2. Xem thông tin chi tiết thiết bị</li> <li>3. Thêm thiết bị mới với thông tin hợp lệ → Lưu thông tin</li> <li>4. Sửa thông tin thiết bị có sẵn → Lưu thay đổi</li> <li>5. Xóa thiết bị</li> </ol>	<ol style="list-style-type: none"> <li>1. Hiện thị danh sách thiết bị</li> <li>2. Hiện thị thông tin chi tiết thiết bị</li> <li>3. Thêm thiết bị thành công</li> <li>4. Cập nhật thông tin thiết bị thành công</li> <li>5. Xóa thiết bị khỏi hệ thống thành công</li> </ol>	OK
Kiểm tra chức năng quản lý phiên đo	<ol style="list-style-type: none"> <li>1. Vào màn hình quản lý phiên đo</li> <li>2. Sửa thông tin danh mục có sẵn → Lưu thay đổi</li> <li>3. Xem đồ thị một phiên đo</li> <li>4. Tải bản ghi phiên đo → Lưu tệp</li> </ol>	<ol style="list-style-type: none"> <li>1. Hiện thị danh sách phiên đo</li> <li>2. Cập nhật thông tin phiên đo</li> <li>3. Hiện thị đồ thị dữ liệu phiên đo</li> <li>4. Tải file dữ liệu phiên đo thành công</li> </ol>	OK

Kiểm tra chức năng quản lý phân công bệnh nhân - bác sĩ	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	1. Hiển 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.

## PHỤ LỤC

### ***Công cụ hỗ trợ trong đồ án***

#### **Github**

##### a) Giới thiệu chung

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

##### b) Cách sử dụng github

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

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

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

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

### ***Đường dẫn mã nguồn***

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



## TÀI LIỆU THAM KHẢO

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

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

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