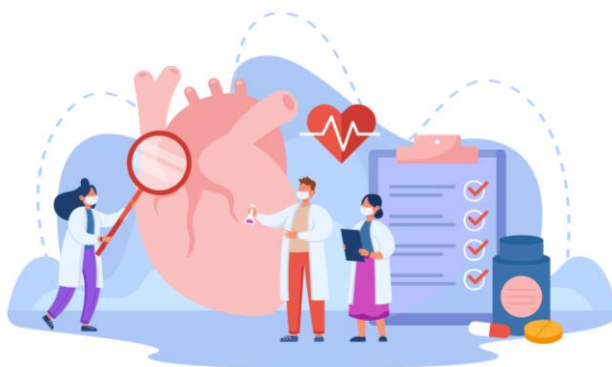




**BÁO CÁO**  
**PBL5 - ĐỒ ÁN KỸ THUẬT MÁY TÍNH**



**HỆ THỐNG GIÁM SÁT NHỊP TIM VÀ CẢNH BÁO**  
**CHO NGƯỜI DÙNG TÍN HIỆU BẤT THƯỜNG**

CBDN hướng dẫn: KS. Trần Phú Quy (ENCLAVE Software)

Giảng viên đồng hướng dẫn: TS. Ninh Khánh Duy

<b>NHÓM 3 (CLC)</b> <b>HỌ VÀ TÊN SINH VIÊN</b>	<b>LỚP HỌC PHẦN ĐỒ ÁN</b>
Cao Kiều Văn Mạnh	20.15A
Trần Đình Minh Khoa	20.15A
Lương Thiện	20.15A
Nguyễn Quốc Cường	20.15A

## TÓM TẮT ĐỒ ÁN

Hiện nay, các vấn đề về bệnh tim mạch dần trở nên phổ biến trên khắp thế giới. Tuy nhiên, thời điểm hiện tại, để nhận biết được dấu hiệu bệnh, người dân cần phải đến các phòng khám, bệnh viện để kiểm tra. Quá trình này tuy đảm bảo, nhưng lại tốn thời gian và tiền bạc dẫn đến nhiều người không có khả năng tiếp cận.

Cùng với sự phát triển của xã hội, hầu hết các gia đình đều sở hữu ít nhất một thiết bị smartphone, từ đó nhóm chúng em đã có ý tưởng về xây dựng dự án giám sát nhịp tim và cảnh báo cho người dùng tín hiệu nhịp tim bất thường, đưa ra các thông tin hỗ trợ cho người dùng thông qua smartphone.

Hiện tại, nhóm đã lắp đặt được thiết bị phần cứng và xây dựng thành công ứng dụng di động để hỗ trợ nhận dạng được rối loạn nhịp tim với kết quả tương đối chính xác, hoàn thành những chức năng chính, đưa vào hoạt động và kiểm tra trong thực tế. Tuy nhiên, kết quả vẫn còn nhiều thiếu sót và hạn chế. Nhóm chúng em sẽ tiếp tục phát triển và hoàn thiện hơn trong tương lai.

## BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Đánh giá
Cao Kiều Văn Mạnh	Phân công công việc, đảm bảo tiến độ thực hiện đồ án	Hoàn thành
	Tìm hiểu, triển khai mô hình học máy	
	Xây dựng server hệ thống	
	Ghép nối và thử nghiệm sản phẩm	
	Lắp đặt và lập trình thiết bị phần cứng	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	
Trần Đình Minh Khoa	Tìm hiểu, triển khai mô hình học máy	Hoàn thành
	Hỗ trợ xây dựng server hệ thống	
	Ghép nối và thử nghiệm sản phẩm	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	
Lương Thiện	Xây dựng ứng dụng di động	Hoàn thành
	Xây dựng cơ sở dữ liệu	
	Ghép nối và thử nghiệm sản phẩm	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	
Nguyễn Quốc Cường	Lắp đặt và lập trình thiết bị phần cứng	Hoàn thành
	Hỗ trợ xây dựng cơ sở dữ liệu	
	Ghép nối và thử nghiệm sản phẩm	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	

## MỤC LỤC

<b>MỤC LỤC .....</b>	<b>iii</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>v</b>
<b>DANH MỤC BẢNG .....</b>	<b>vi</b>
<b>1. Giới thiệu .....</b>	<b>1</b>
1.1. Thực trạng.....	1
1.2. Các vấn đề cần giải quyết.....	1
1.3. Đề xuất giải pháp tổng quan .....	2
<b>2. Giải pháp.....</b>	<b>3</b>
2.1. Nguyên lý và sơ đồ hoạt động của hệ thống.....	3
2.2. Giải pháp về phần cứng .....	4
2.2.1. Linh kiện sử dụng .....	4
2.2.2. Thông số kỹ thuật và nguyên lý hoạt động .....	5
2.2.3. Sơ đồ mạch và lắp đặt.....	7
2.3. Giải pháp về truyền thông .....	7
2.3.1. Server FastAPI.....	7
2.3.2. Firebase Realtime Database .....	7
2.4. Giải pháp ứng dụng di động .....	8
2.4.1. Công nghệ sử dụng .....	8
2.4.2. Sơ đồ usecase hệ thống.....	8
2.4.3. Cấu trúc hệ thống của ứng dụng di động.....	9
2.5. Giải pháp về nhận diện tín hiệu ECG bất thường.....	9
2.5.1. Thông tin về tín hiệu ECG.....	9
2.5.2. Thu thập dữ liệu.....	10
2.5.3. Thuật toán tìm đỉnh trong tín hiệu ECG .....	12
2.5.4. Tiền xử lý dữ liệu .....	12
2.5.5. Huấn luyện mô hình .....	14
<b>3. Kết quả.....</b>	<b>19</b>
3.1. Công cụ và framework sử dụng.....	19

3.2. Kết quả thu thập dữ liệu .....	19
3.3. Kết quả lắp đặt phần cứng .....	20
3.4. Kết quả xây dựng phần mềm.....	20
3.4.1. Kết quả xây dựng server.....	20
3.4.2. Kết quả xây dựng ứng dụng di động .....	20
3.5. Kết quả mô hình phát hiện bất thường trong nhịp tim.....	24
3.5.1. Kết quả trên tập huấn luyện.....	24
3.5.2. Kết quả kiểm thử mô hình .....	26
3.5.3. Kết quả sử dụng thực tế.....	27
4. Kết luận.....	29
4.1. Đánh giá sản phẩm .....	29
4.2. Hướng phát triển.....	29
5. Danh mục tài liệu tham khảo.....	30

## DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ tổng thể của hệ thống .....	3
Hình 2: Cảm biến nhịp tim DFRobot .....	4
Hình 3: Board Uno Wifi Wemos D1 .....	5
Hình 4: Vị trí để dán miếng đệm cảm biến .....	6
Hình 5: Sơ đồ lắp đặt mạch .....	7
Hình 6: Sơ đồ usecase của ứng dụng di động .....	8
Hình 7: Cấu trúc hệ thống của mobile app .....	9
Hình 8: QRS Complex trong tín hiệu điện tim .....	10
Hình 9: Mô tả ECG của các lớp .....	12
Hình 10: Cấu trúc mô hình CNN nhóm tham khảo .....	15
Hình 11: Kết quả phần cứng sau khi lắp đặt .....	20
Hình 12: Giao diện chức năng đăng nhập, đăng ký, quên mật khẩu .....	21
Hình 13: Giao diện trang chủ .....	21
Hình 14: Giao diện thông tin cá nhân .....	22
Hình 15: Giao diện chức năng đo và xem kết quả .....	23
Hình 16: Giao diện chức năng xem lịch sử đo .....	24
Hình 17: Kết quả huấn luyện trên model 1 .....	24
Hình 18: Kết quả huấn luyện trên model 2 .....	25
Hình 19: Kết quả huấn luyện trên model 3 .....	25

## DANH MỤC BẢNG

<i>Bảng 1: Đề xuất giải pháp tổng quan .....</i>	<i>2</i>
<i>Bảng 2. Các linh kiện được sử dụng.....</i>	<i>4</i>
<i>Bảng 3: Bảng mô tả dữ liệu ECG thu thập được .....</i>	<i>11</i>
<i>Bảng 4: Bảng mô tả phương pháp xử lý dữ liệu.....</i>	<i>13</i>
<i>Bảng 5: Bảng mô tả về dữ liệu kiểm thử. ....</i>	<i>13</i>
<i>Bảng 6: Bảng thông tin mô hình CNN nhóm tự phát triển.....</i>	<i>16</i>
<i>Bảng 7: Các API được sử dụng. ....</i>	<i>16</i>
<i>Bảng 8: Các khoảng nhịp tim bình thường. ....</i>	<i>18</i>
<i>Bảng 9: Bảng kết quả dữ liệu dataset 1. ....</i>	<i>19</i>
<i>Bảng 10: Bảng kết quả dữ liệu dataset 2. ....</i>	<i>19</i>
<i>Bảng 11: Bảng kết quả dữ liệu kiểm thử .....</i>	<i>19</i>
<i>Bảng 12: Bảng danh sách endpoint của API.....</i>	<i>20</i>
<i>Bảng 13: Bảng kết quả kiểm thử mô hình (%) .....</i>	<i>27</i>
<i>Bảng 14. Bảng kết quả sử dụng thực tế.....</i>	<i>28</i>

## **1. Giới thiệu**

### **1.1. Thực trạng**

Theo báo cáo của Tổ chức Y tế Thế giới (WHO), các bệnh về tim mạch là nguyên nhân gây tử vong số một hiện nay. Ước tính có đến 17,9 triệu người chết vì các bệnh tim mạch vào năm 2019, chiếm 32% tổng số ca tử vong toàn cầu và trong đó hơn 75% số ca tử vong đến từ các quốc gia có thu nhập thấp và trung bình.

Hiện nay có rất nhiều loại sản phẩm nhận dạng và chẩn đoán các bệnh lý liên quan đến tim mạch. Tuy nhiên các hệ thống đó thường rất lớn, tiêu tốn rất nhiều dữ liệu, kèm theo đó là chi phí cao. Do đó, nhóm chúng em tiến hành thử nghiệm với đề tài này, nhằm tìm ra giải pháp tốt, chi phí thấp và tiện lợi nhất cho người dùng.

Giải pháp của chúng em cung cấp một phương pháp đơn giản và hiệu quả để chẩn đoán bệnh tim mạch mà không cần sử dụng những trang thiết bị phức tạp. Thay vào đó, người dùng có thể sử dụng một ứng dụng di động đơn giản và nhận được kết quả chẩn đoán tức thì.

Với giải pháp này, nhóm chúng em hy vọng sẽ giúp tiết kiệm thời gian, công sức và tiền bạc cho người dùng, đồng thời mang lại khả năng phát hiện sớm các bệnh tim mạch và tăng cường chất lượng chăm sóc sức khỏe trong cộng đồng.

### **1.2. Các vấn đề cần giải quyết**

Các vấn đề mà nhóm cần phải giải quyết trong quá trình xây dựng dự án này như sau:

- Cần có thiết bị thu tín hiệu điện tâm đồ ECG.
- Cần có dữ liệu tín hiệu điện tâm đồ được lấy từ các cơ sở dữ liệu của các bệnh viện để đảm bảo các vấn đề về mặt y học.
- Xây dựng được mô hình, thuật toán nhận dạng được rối loạn nhịp tim.
- Cần có ứng dụng để việc thao tác sử dụng thiết bị phần cứng và theo dõi kết quả trực quan, dễ dàng.
- Hệ thống cần được xây dựng để hoạt động theo thời gian thực.



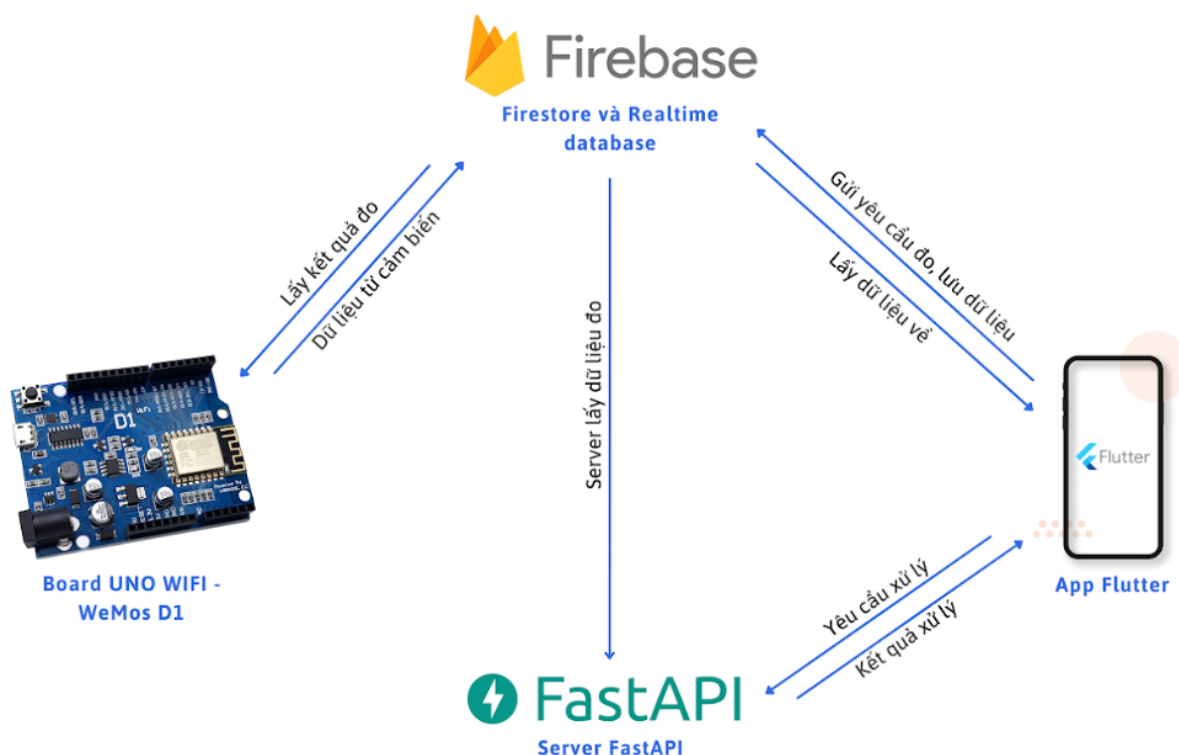
### 1.3. Đề xuất giải pháp tổng quan

Vấn đề	Giải pháp đề xuất
Phần cứng	Sử dụng Board Uno Wifi Wemos D1 làm board mạch chính. Sử dụng cảm biến điện tim DF Robot để thu tín hiệu ECG.
Nhận dạng nhịp tim	Thu thập và xử lý dữ liệu từ PhysioNet service. Xây dựng và huấn luyện model nhận dạng rối loạn nhịp tim với mô hình CNN.
Ứng dụng di động	Xây dựng ứng dụng di động bằng Flutter với các chức năng điều khiển thiết bị và theo dõi kết quả.
Server	Xây dựng server, viết API sử dụng FastAPI.

*Bảng 1: Đề xuất giải pháp tổng quan*

## 2. Giải pháp

### 2.1. Nguyên lý và sơ đồ hoạt động của hệ thống



Hình 1: Sơ đồ tổng thể của hệ thống

Nguyên lý hoạt động của hệ thống như sau:

- Người dùng sẽ truy cập ứng dụng và gửi yêu cầu đến cảm biến để tiến hành đo nhịp tim.
- Sau khi đo, dữ liệu từ cảm biến sẽ được gửi lên Firebase để lưu trữ.
- Server sẽ đọc dữ liệu từ Firebase, khi đủ dữ liệu sẽ tiến hành xử lý và đưa ra kết quả dự đoán.
- Sau khi có kết quả, server sẽ tiến hành trả dữ liệu về ứng dụng di động và lưu dữ liệu vào Firestore. Dựa vào mỗi request của mobile app sẽ nhận được API tương ứng.
- Khi đã có kết quả trả về từ server, người dùng có thể vào ứng dụng để xem kết quả ở màn hình.

## 2.2. Giải pháp về phần cứng

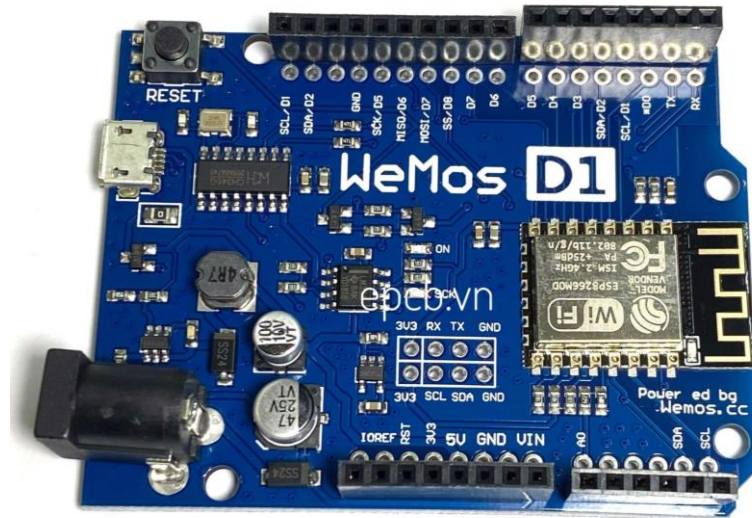
### 2.2.1. Linh kiện sử dụng

Linh kiện	Chức năng	Giá thành (VNĐ)
Board Uno Wifi Wemos D1	Điều khiển các thành phần. Đọc dữ liệu từ cảm biến và gửi lên Firebase.	79.000
Cảm biến điện tim DFRobot Gravity	Đo tín hiệu điện tim ECG từ người dùng.	500.000
Màn hình LCD	Hiển thị thông tin đến người dùng.	20.000
Loa buzzer	Thông báo trong quá trình đo.	8.000
Sạc dự phòng	Cung cấp nguồn điện cho hệ thống	200.000
<b>Tổng cộng</b>		<b>807.000</b>

Bảng 2. Các linh kiện được sử dụng.



Hình 2: Cảm biến nhịp tim DFRobot



Hình 3: Board Uno Wifi Wemos D1

### 2.2.2. Thông số kỹ thuật và nguyên lý hoạt động

#### ❖ Cảm biến điện tim DFRobot Gravity: Analog ECG Heart Rate Sensor

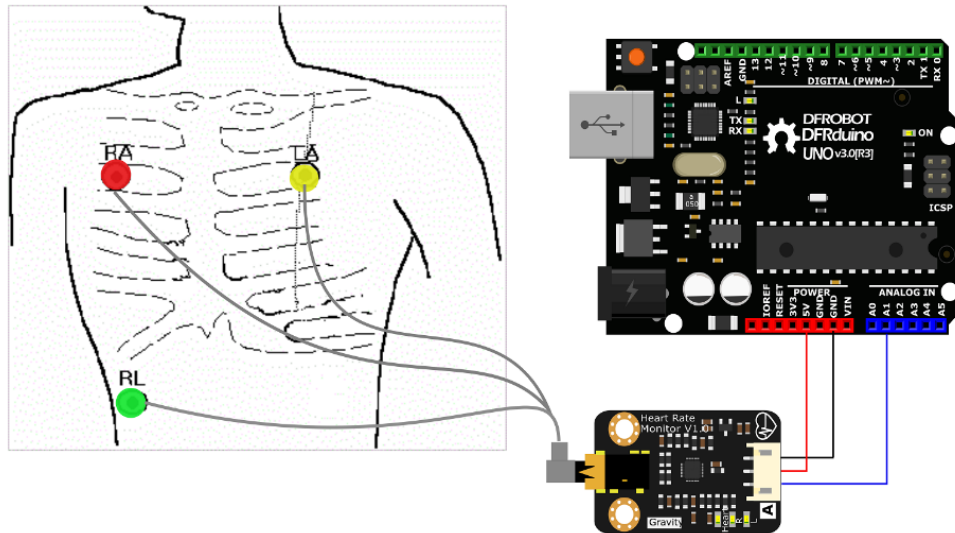
##### - Thông số kỹ thuật:

- Model: DFRobot Gravity: Analog ECG Heart Rate Sensor.
- Điện áp đầu vào: 3.3 - 6V (ổn định nhất ở 5V).
- Điện áp đầu ra: 0 - 3.3V.
- Dòng hoạt động: nhỏ hơn 10mA.

##### - Các lưu ý khi sử dụng:

- Tín hiệu điện tâm đồ có thể khá nhiễu do hoạt động cơ bắp xung quanh.
- Để cải thiện chất lượng tín hiệu cần giữ miếng đệm càng gần tim càng tốt và ở đúng vị trí, không di chuyển quá nhiều trong khi đo, làm sạch khu vực dán cũng như sử dụng miếng đệm mới cho mỗi lần đo.

- **Nguyên lý hoạt động:** cảm biến thông qua các miếng dán trên ngực người dùng sẽ đo được sự thay đổi dòng điện của tim người. Dòng điện này có thể được vẽ thành biểu đồ dưới dạng ECG.



Hình 4: Vị trí để dán miếng đệm cảm biến

#### ❖ Board Uno Wifi Wemos D1

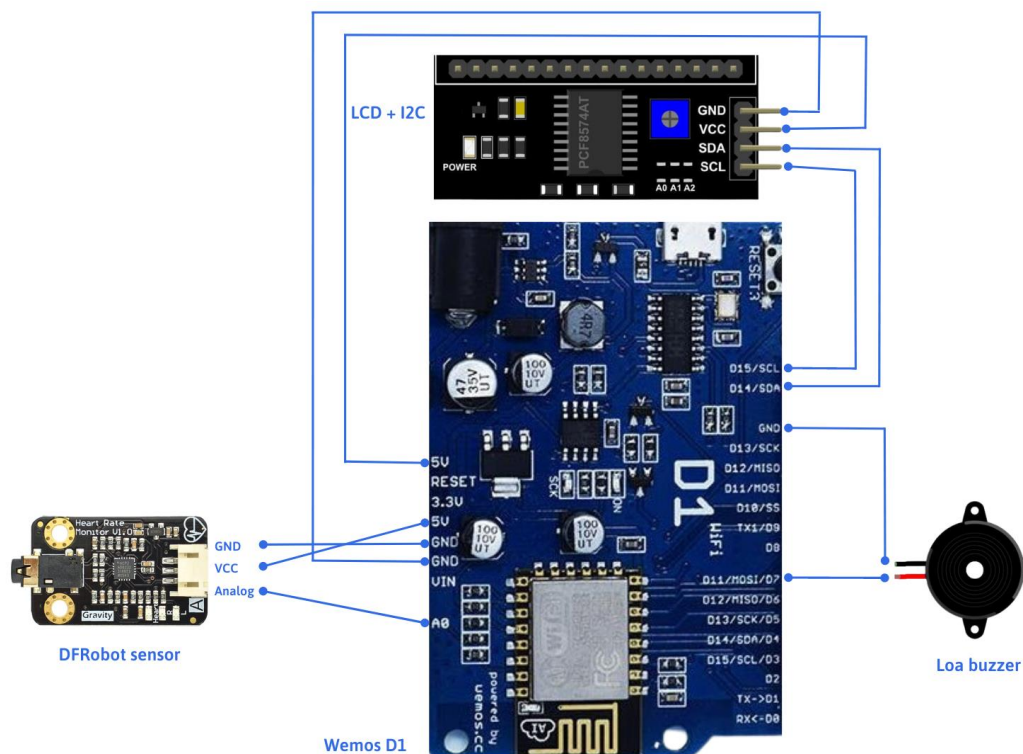
##### - Thông số kỹ thuật:

- Board mạch được thiết kế với hình dáng tương tự Arduino Uno nhưng trung tâm lại là module wifi Soc ESP8266 được build lại firmware NodeMCU Lua để có thể chạy với chương trình Arduino.
- 11 chân đầu vào và đầu ra digital.
- 1 đầu vào analog (điện áp tối đa 3.2V).
- Kết nối thông qua USB.
- Điện áp đầu vào: 6 – 9V.

- **Nguyên lý hoạt động:** board Uno Wifi Wemos D1 sẽ là board mạch chính, đọc dữ liệu từ cảm biến và gửi dữ liệu này lên Firebase Realtime database.

### 2.2.3. Sơ đồ mạch và lắp đặt

Các thiết bị phần cứng được kết nối với nhau theo sơ đồ sau.



Hình 5: Sơ đồ lắp đặt mạch

## 2.3. Giải pháp về truyền thông

### 2.3.1. Server FastAPI

- FastAPI là một framework web cho phép xây dựng và cấu hình các API nhanh chóng và dễ dàng bằng Python.
- Nhóm chúng em tiến hành xây dựng server bằng FastAPI và cung cấp các API chính cho toàn bộ hệ thống, các API này sẽ được gọi ở phía client là ứng dụng di động.

### 2.3.2. Firebase Realtime Database

- Firebase Realtime Database là một dịch vụ cơ sở dữ liệu thời gian thực (realtime) của Firebase, cung cấp cho người dùng một cơ sở dữ liệu NoSQL lưu trữ dữ liệu dưới dạng JSON và cho phép truy cập và đồng bộ hóa dữ liệu giữa các ứng dụng di động, web và các thiết bị khác nhau trong thời gian thực.

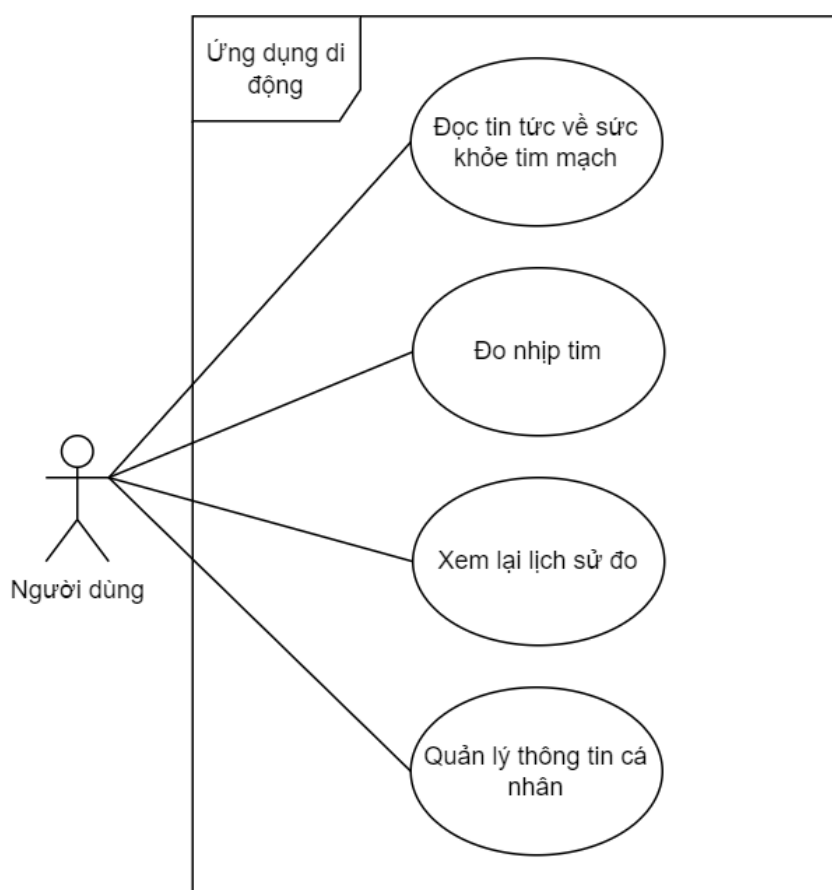
- Nhóm chúng em sử dụng Firebase Realtime Database để điều khiển thiết bị phần cứng thông qua ứng dụng di động cũng như để lưu trữ dữ liệu đo được từ cảm biến.

## 2.4. Giải pháp ứng dụng di động

### 2.4.1. Công nghệ sử dụng

- Ngôn ngữ lập trình: sử dụng Flutter, Dart để lập trình mobile app trên nền tảng android.
- Một số thư viện hỗ trợ trong quá trình phát triển bao gồm: firebase\_auth, cloud\_firestore, firebase\_database, firebase\_storage,...
- Ứng dụng và công cụ hỗ trợ phát triển: Genymotion và Android Studio.

### 2.4.2. Sơ đồ usecase hệ thống



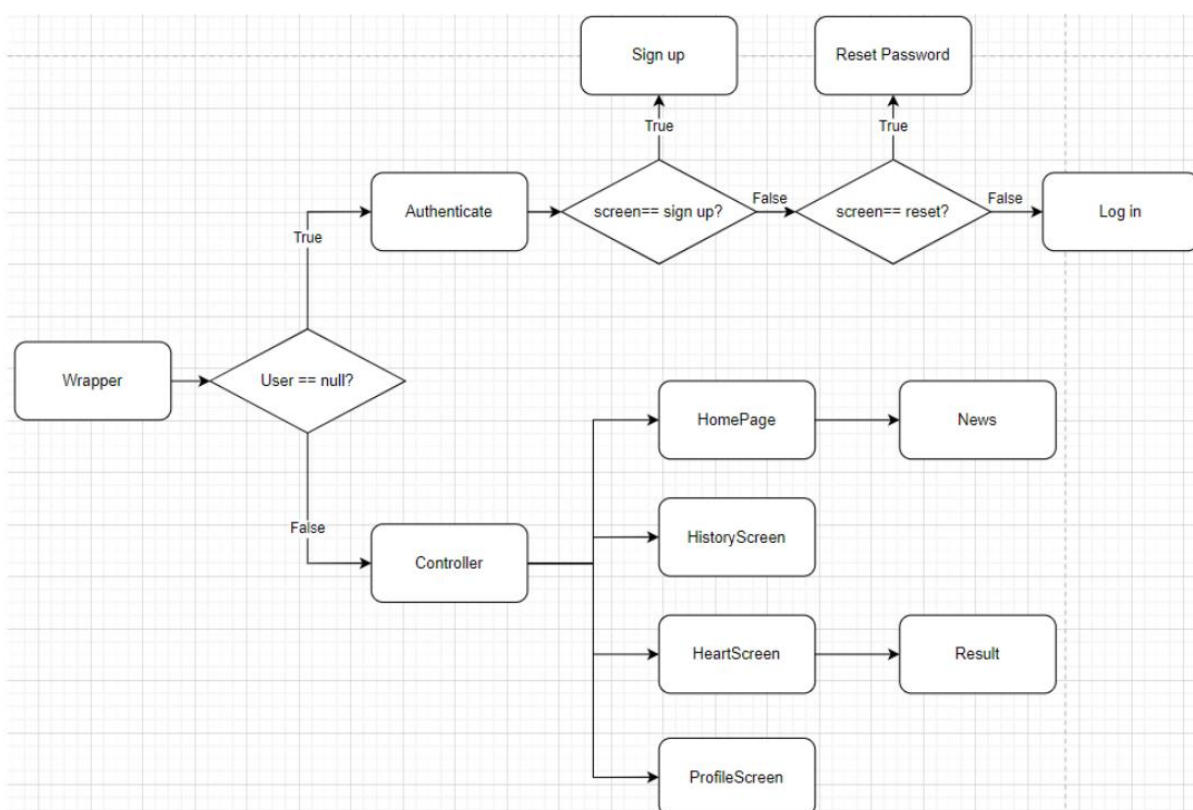
Hình 6: Sơ đồ usecase của ứng dụng di động

- Người dùng có thể đọc các bài báo, thông tin về bảo vệ sức khỏe tim mạch, chế độ ăn uống và tập thể dục.



- Chức năng đo nhịp tim và xem kết quả đo: khi người dùng bấm nút yêu cầu đo thì server sẽ gửi yêu cầu tới cảm biến để tiến hành đo nhịp tim và trả lại kết quả hiển thị trên giao diện ứng dụng.
- Chức năng xem lại lịch sử: Hiển thị ra lịch sử của các lần đo và các cảnh báo về bệnh và sức khỏe.
- Chức năng quản lý thông tin cá nhân: người dùng có thể quản lý và chỉnh sửa thông tin cá nhân của bản thân.

### 2.4.3. Cấu trúc hệ thống của ứng dụng di động



Hình 7: Cấu trúc hệ thống của mobile app

## 2.5. Giải pháp về nhận diện tín hiệu ECG bất thường

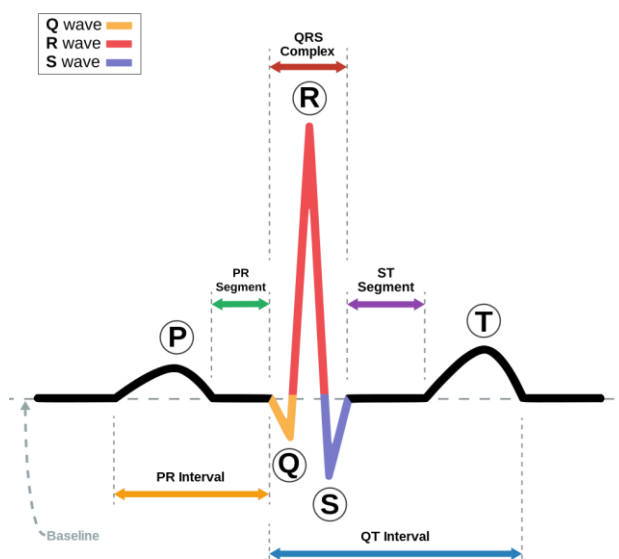
### 2.5.1. Thông tin về tín hiệu ECG

Điện tâm đồ - Electrocardiogram (viết tắt ECG) là một xét nghiệm ghi lại hoạt động điện học của tim dưới dạng đồ thị. Các xung điện tự nhiên điều phối sự co bóp của tim để giữ máu tuần hoàn. Điện tâm đồ ghi lại những xung điện này. Sự thay đổi của xung điện được phát hiện qua điện tâm đồ có thể là dấu hiệu của nhiều bệnh lý liên quan đến tim.



Trong tín hiệu ECG, một thành phần quan trọng là QRS Complex. Nó thể hiện sự tương ứng giữa các điện thế trên bề mặt của cơ thể và các sự kiện điện tâm thu và tâm trương trong tim. QRS complex được hình thành bởi ba sóng điện: Sóng Q, sóng R và sóng S. Sóng R là sóng có biên độ lớn nhất và thường là sóng đầu tiên trong QRS complex. Sóng Q là sóng âm đầu tiên trong QRS complex và có biên độ âm. Sóng S là sóng âm thứ hai và có biên độ âm.

Thời gian trung bình để QRS complex xuất hiện là từ 0,06 đến 0,1 giây, và biên độ trung bình của nó là từ 5 đến 30 mm. Bất kỳ thay đổi nào trong hình dạng, biên độ hoặc độ dài của QRS complex cũng có thể cho thấy các vấn đề về tim mạch như rối loạn nhịp tim, suy tim, hẹp van tim, đau tim và các bệnh lý khác.



Hình 8: QRS Complex trong tín hiệu điện tim

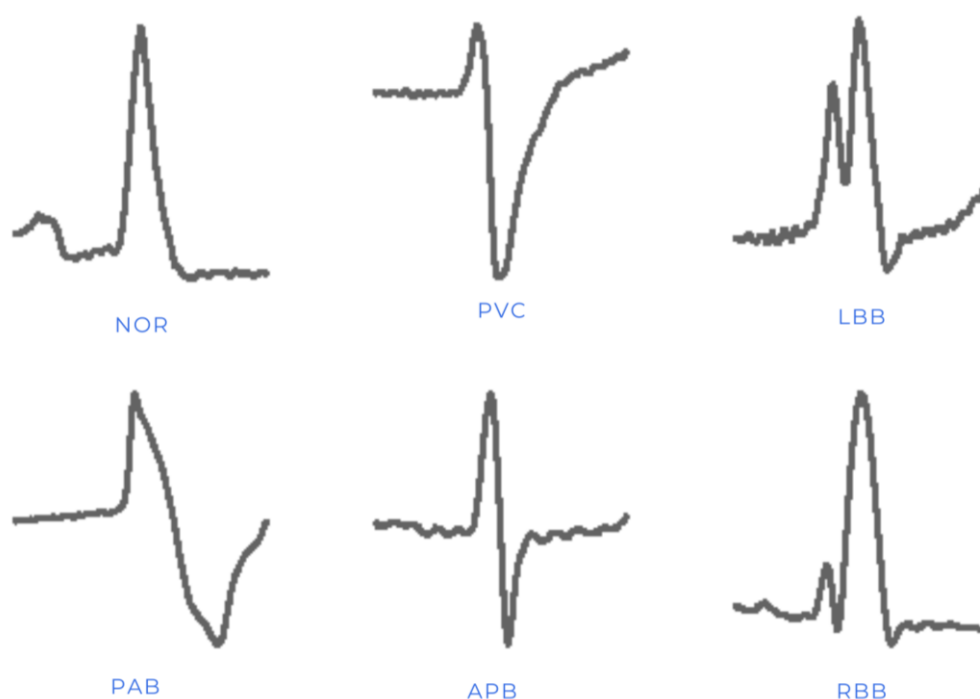
### 2.5.2. Thu thập dữ liệu

- Nguồn dữ liệu tín hiệu ECG được lấy từ cơ sở dữ liệu MIT-BIH Arrhythmia Database, được đo từ 45 bệnh nhân, bao gồm 19 nữ (23 - 89 tuổi) và 26 nam (32 - 89 tuổi) ở tần số 360Hz.
- Liên kết đến bộ dữ liệu: <https://www.physionet.org/content/mitdb/1.0.0/>
- Dữ liệu bao gồm 48 records, mỗi record dài khoảng 30 phút.
- Dataset có tổng cộng 19 lớp. Nhóm tiến hành lấy ra 6 lớp để sử dụng, cụ thể như sau:

- NOR: Normal - Nhịp xoang bình thường
  - PVC: Premature Ventricular Contraction - Nhịp tâm thất bất thường xuất phát từ tâm thất.
  - APB: Atrial Premature Beat - Nhịp xoang bất thường xuất phát từ tâm nhĩ.
  - LBB: Left Bundle Branch Block - Rối loạn dẫn truyền do khối nhánh trái bị ngăn cản.
  - RBB: Right Bundle Branch Block - Rối loạn dẫn truyền do khối nhánh phải bị ngăn cản.
  - PAB: Paced Beat - Nhịp tim được tạo bởi máy phát nhịp điện cấy trong cơ thể.
- Trong đó, lấy ra các record '100', '101', '103', '106', '107', '118', '109', '209' để làm dữ liệu kiểm thử mô hình.

Lớp	Records
<b>NOR</b>	100, 101, 103, 105, 108, 112, 113, 114, 115, 117, 121, 122, 123, 202, 205, 219, 230, 234
<b>PVC</b>	106, 116, 119, 200, 201, 203, 208, 210, 213, 215, 221, 228, 233
<b>PAB</b>	102, 104, 107, 217
<b>RBB</b>	118, 124, 212, 231
<b>LBB</b>	109, 111, 207, 213
<b>APB</b>	209, 220, 222, 223, 232

Bảng 3: Bảng mô tả dữ liệu ECG thu thập được



Hình 9: Mô tả ECG của các lớp.

### 2.5.3. Thuật toán tìm đỉnh trong tín hiệu ECG

Ta tiến hành tìm đỉnh R của tín hiệu ECG bằng hàm hỗ trợ trong thư viện neurokit2 dựa trên bài báo *“An Open-Source Feature Extraction Tool for the Analysis of Peripheral Physiological Data”* (Tài liệu tham khảo mục [10]). Thuật toán sau để tìm đỉnh R của tín hiệu ECG như sau:

- Tạo một cửa sổ có độ dài 800ms, trượt cửa sổ đó lần lượt trên từng mẫu của tín hiệu, với mỗi lần trượt đó ta thu được một đoạn tín hiệu có độ dài 800ms.
- Sau đó, tìm giá trị lớn nhất trên đoạn tín hiệu đó, nếu vị trí của mẫu lớn nhất đó nằm giữa đoạn tín hiệu thì đó là một đỉnh R.
- Cuối cùng xử lý toàn bộ các đoạn tín hiệu xuất ra từ cửa sổ trượt, ta thu được danh sách các đỉnh R của tín hiệu đó.

### 2.5.4. Tiền xử lý dữ liệu

#### ❖ Dữ liệu huấn luyện

- Từ các đỉnh ECG được đánh dấu sẵn trong bộ dữ liệu, ta tiến hành cắt dữ liệu thành các hình ảnh có kích thước 128 x 128 để tạo thành 2 bộ dữ liệu, cụ thể như sau.

Bộ dữ liệu	Số mẫu trong một ảnh	Tần số lấy mẫu (Hz)	Thời gian của một ảnh (s)
Dataset 1	128	360	~ 0.36
Dataset 2	24	33	~ 0.72

Bảng 4: Bảng mô tả phương pháp xử lý dữ liệu.

#### ❖ Dữ liệu kiểm thử

- Từ các record đã được lấy ra cho việc kiểm thử, ta tiến hành cắt dữ liệu thành các hình ảnh có kích thước 128 x 128 để tạo thành các tập dữ liệu kiểm thử.
- Trong đó, một tập dữ liệu kiểm thử sẽ được cắt theo đỉnh đã được đánh dấu và một tập dữ liệu kiểm thử sẽ được cắt theo đỉnh tìm được bằng thuật toán tìm đỉnh đã được đề cập trong phần 2.5.3.

Bộ dữ liệu kiểm thử	Cách cắt	Số mẫu	Tần số lấy mẫu (Hz)	Thời gian của một ảnh (s)
Test set 1.1	Dùng đỉnh được đánh dấu	128	360	~ 0.36
Test set 1.2	Dùng thuật toán tìm đỉnh	128	360	~ 0.36
Test set 2.1	Dùng đỉnh được đánh dấu	24	33	~ 0.72
Test set 2.2	Dùng thuật toán tìm đỉnh	24	33	~ 0.72

Bảng 5: Bảng mô tả về dữ liệu kiểm thử.

#### ❖ Xử lý dữ liệu thu được từ cảm biến

##### Về tần số lấy mẫu:

- Với tần số 20Hz, cảm biến cho kết quả dữ liệu khá thừa vì bị mất đi nhiều mẫu quan trọng do tần số thấp.
- Với tần số 33Hz, cảm biến cho kết quả đồ thị khá ổn.
- Với tần số 50Hz, cảm biến cho ra đồ thị nhiều nhiễu vì tốc độ lấy mẫu quá nhanh.
- Kết luận: Sử dụng mức tần số 33Hz cho kết quả tốt nhất. Do đó nhóm sẽ sử dụng mức tần số này trên cảm biến.

### **Về số mẫu trong một ảnh:**

- Với dữ liệu đo được từ cảm biến, việc tăng sample lấy mẫu giúp có nhiều thông tin hơn, tuy nhiên khi tăng lên 30 samples thì một số ảnh cắt ra chứa thông tin của đỉnh khác, dẫn đến có thể gây sai lệch kết quả.
- Vì cảm biến của nhóm có mặt hạn chế về khả năng đo so với thiết bị được sử dụng để thu thập dataset gốc. Do đó, sau quá trình khảo sát nhóm chọn cắt ảnh với 12 samples hoặc 24 samples từ dữ liệu đọc từ cảm biến tùy theo mô hình sử dụng.
- Để đảm bảo sự đồng nhất về thời gian của tín hiệu ECG trong một ảnh. Đối với các mô hình được huấn luyện với dataset 1, dữ liệu từ cảm biến sẽ được cắt với 12 sample (tương đương độ dài  $\sim 0.36s$ ). Đối với các mô hình được huấn luyện với dataset 2, dữ liệu từ cảm biến sẽ được cắt với 24 sample (tương đương độ dài  $\sim 0.72s$ ).

### **2.5.5. Huấn luyện mô hình**

#### **❖ Giới thiệu về mô hình CNN**

CNN là một loại mạng nơ-ron tích chập được sử dụng phổ biến trong xử lý ảnh và nhận dạng hình ảnh. Nó được truyền cảm hứng từ cách mà não nhận thức hình ảnh ở người. Mô hình CNN có cấu trúc dựa trên các lớp:

- Lớp Convolution: chứa các bộ lọc convolutional để phát hiện các đặc trưng. Chúng hoạt động như các detector đặc trưng ở nhiều vị trí và kích thước trong hình ảnh.
- Lớp Pooling: rút gọn kích thước dữ liệu bằng cách trích xuất đặc trưng quan trọng. Nó giúp khả năng tổng quát hóa và giảm sự phụ thuộc vào vị trí.
- Lớp Fully Connected: kết nối tất cả nơ-ron ở lớp trước với mỗi nơ-ron ở lớp này, tương tự như các mạng nơ-ron tích chập thông thường.
- Lớp Output: dự đoán xác suất thuộc về lớp nào dựa trên đặc trưng được trích xuất.

#### **❖ Lý do lựa chọn mô hình CNN**

Nhóm quyết định sử dụng mô hình CNN để giải quyết bài toán phân loại tín hiệu điện tim bất thường vì các lý do chính sau đây.

- Mô hình CNN giúp giảm bớt các giá trị nhiễu trong quá trình huấn luyện mô hình bằng các lớp tích chập. Do đó giúp cải thiện độ chính xác của mô hình khi đưa vào sử dụng thực tế với các thiết bị giá rẻ, có độ nhiễu cao.
- Bằng các sử dụng mô hình CNN, chúng ta có thể sử dụng với nhiều loại thiết bị đo tín hiệu ECG với tần số và khả năng đo đặc khác nhau.

#### ❖ Huấn luyện mô hình nhận diện tín hiệu ECG bất thường với mô hình CNN

Nhóm tiến hành phân loại 6 loại tín hiệu ECG đã được đề cập ở trên sử dụng mô hình CNN. Trong đó, nhóm tiến hành huấn luyện 3 mô hình, cụ thể như sau:

- **Mô hình 1:** Xây dựng dựa trên cấu trúc CNN tham khảo từ bài báo “*ECG arrhythmia classification using a 2-D convolutional neural network*” (Tài liệu tham khảo mục [1]) bao gồm các lớp như hình 11. Mô hình được huấn luyện trên bộ dữ liệu Dataset 1.

	Type	Kernel size	Stride	# Kernel	Input size
Layer1	Conv2D	3 x 3	1	64	128 x 128 x 1
Layer2	Conv2D	3 x 3	1	64	128 x 128 x 64
Layer3	Pool	2 x 2	2		128 x 128 x 64
Layer4	Conv2D	3 x 3	1	128	64 x 64 x 64
Layer5	Conv2D	3 x 3	1	128	64 x 64 x 128
Layer6	Pool	2 x 2	2		64 x 64 x 128
Layer7	Conv2D	3 x 3	1	256	32 x 32 x 128
Layer8	Conv2D	3 x 3	1	256	32 x 32 x 256
Layer9	Pool	2 x 2	2		32 x 32 x 256
Layer10	Full			2048	16 x 16 x 256
Layer11	Out			8	2048

Hình 10: Cấu trúc mô hình CNN nhóm tham khảo.

- **Mô hình 2:** Nhóm tiến hành xây dựng lại mô hình CNN với 9 lớp được mô tả tại bảng 5 và tiến hành huấn luyện trên bộ dữ liệu Dataset 1.
- **Mô hình 3:** Sử dụng mô hình CNN có cấu trúc như mô hình 2, tuy nhiên mô hình 3 sẽ được huấn luyện trên bộ dữ liệu Dataset 2.

Lớp	Kernel size	Kernel	Hàm kích hoạt	Kích thước đầu ra
Conv2D	3x3	32	ReLU	126 x 126 x 32
MaxPooling2D	2x2	-	-	63 x 63 x 32
Conv2D	3x3	64	ReLU	61 x 61 x 64
MaxPooling2D	2x2	-	-	30 x 30 x 64
Conv2D	3x3	128	ReLU	28 x 28 x 128
MaxPooling2D	2x2	-	-	14 x 14 x 128
Flatten	-	-	-	25,088
Dense	-	64	ReLU	64
Dense (Output)	-	6	Softmax	6

Bảng 6: Bảng thông tin mô hình CNN nhóm tự phát triển.

Tên API	Công thức	Mục đích
Batch Normalization	$y_i = \frac{x_i - \text{mean}}{\sqrt{\text{variance} - \epsilon}}$	Chuẩn hóa các giá trị đầu vào về trung bình bằng 0 và độ lệch chuẩn bằng 1, giúp ổn định quá trình huấn luyện.
Activation Function: ReLU	$\text{ReLU}(x) = \max(0; x)$	ReLU giúp lọc các giá trị nhỏ hơn 0 và giúp tốc độ tính toán nhanh hơn, giảm thời gian huấn luyện.
Loss Function	Cross entropy	Tính toán tổn thất chéo giữa các nhãn thực tế và dự đoán.
Optimizer Function	Adam	Giúp cải thiện hiệu quả của giải thuật gradient descent bằng cách sử dụng các tỷ lệ học thích hợp, giúp cho quá trình tối ưu hội tụ nhanh hơn và ổn định hơn.

Bảng 7: Các API được sử dụng.

### ❖ Tinh chỉnh mô hình

Cả ba mô hình đều được huấn luyện với các bộ dữ liệu với số epoch = 30, batch size = 128. Trong dự án này, nhóm sử dụng các kỹ thuật Drop-out, Regularization và Early stopping để tinh chỉnh mô hình giúp giảm overfitting.

- **Drop-out:** Drop-out là một layer trong mạng nơ ron, cho phép ngắt ngẫu nhiên các kết nối theo tỉ lệ nhất định, Ví dụ: nn.Dropout(0.5) tức là ngắt ngẫu nhiên 50% kết nối. Điều này ngăn model quá phụ thuộc vào một số nơ ron nhất định, yêu cầu model phải phân tán sự phụ thuộc vào nhiều nơ ron. Layer này sẽ được vô hiệu hóa khi kết thúc quá trình training để kết quả dự đoán luôn xác định.
- **Regularization:** Regularization dùng để phạt model khi các hệ số của model trở nên quá lớn, nếu cho phép các hệ số của model trở nên quá lớn sẽ khiến model phức tạp hơn và có thể khiến model 'học thuộc lòng' tập train, thay vì cố gắng ghi nhớ các đặc trưng chung của dữ liệu.
- **Early stopping:** Early stopping là một kỹ thuật giúp ngăn chặn việc huấn luyện quá mức (overfitting) mô hình. Trong quá trình huấn luyện, sau mỗi epochs, mô hình được đánh giá trên tập validation. Nếu hiệu suất trên tập validation không cải thiện trong một số epochs nhất định (thường gọi là patience), việc huấn luyện sẽ dừng lại. Mô hình sẽ lấy lại trạng thái tốt nhất trên tập validation để dừng quá trình huấn luyện. Điều này giúp ngăn ngừa việc mô hình quá phụ thuộc vào tập train và tối ưu quá mức trên đó, dẫn đến hiệu suất kém trên dữ liệu thực tế.



### 2.5.6. Ngưỡng xác định nhịp tim BPM bình thường hay bất thường

Ngưỡng xác định nhịp tim BPM của một người dựa trên tuổi là bình thường hay không được xác định như bảng bên dưới.

<b>Độ tuổi</b>	<b>Khoảng BPM bình thường</b>
0 – 1	100 – 160
1 – 3	90 – 150
3 – 5	80 – 140
5 – 12	70 – 120
12 – 18	60 – 100
18 – 100	60 – 100

*Bảng 8: Các khoảng nhịp tim bình thường.*

### 3. Kết quả

#### 3.1. Công cụ và framework sử dụng

- Công cụ lập trình: Arduino IDE, Visual studio code, Pycharm, Android Studio, Google Colab.
- Cơ sở dữ liệu: Firebase với Firestore và Firebase realtime database.
- Ngôn ngữ sử dụng: Python, C++, Dart.
- Framework sử dụng: FastAPI, Flutter.

#### 3.2. Kết quả thu thập dữ liệu

##### ❖ Dataset 1

Set\Class	NOR	PVC	PAB	RBB	LBB	APB
Train set	51578	5238	2896	4074	4466	1623
Validation set	12894	1309	724	1018	1116	406

Bảng 9: Bảng kết quả dữ liệu dataset 1.

##### ❖ Dataset 2

Set\Class	NOR	PVC	PAB	RBB	LBB	APB
Train set	51537	5238	2893	4071	4463	1620
Validation set	12884	1309	723	1018	1116	405

Bảng 10: Bảng kết quả dữ liệu dataset 2.

##### ❖ Tập dữ liệu kiểm thử

Set\Class	NOR	PVC	PAB	RBB	LBB	APB
Test 1.1	10309	635	2078	2166	2492	517
Test 1.2	863	91	1381	118	1941	20
Test 2.1	10300	634	2076	2165	2490	516
Test 2.2	2194	52	405	499	551	61

Bảng 11: Bảng kết quả dữ liệu kiểm thử

### 3.3. Kết quả lắp đặt phần cứng



Hình 11: Kết quả phần cứng sau khi lắp đặt.

### 3.4. Kết quả xây dựng phần mềm

#### 3.4.1. Kết quả xây dựng server

- Xây dựng thành công server sử dụng FastAPI framework.
- Server cung cấp các API sau cho ứng dụng di động.

STT	API	Method	Mô tả
1	/predict	GET	Lấy kết quả dự đoán từ server.
2	/getbpm	GET	Lấy kết quả đo BPM từ server.
3	/isrunning	GET	Kiểm tra thiết bị phần cứng có đang hoạt động hay không.

Bảng 12: Bảng danh sách endpoint của API.

#### 3.4.2. Kết quả xây dựng ứng dụng di động

##### ❖ Chức năng đăng ký, đăng nhập, quên mật khẩu:

- Người dùng phải đăng ký tài khoản để đăng nhập vào ứng dụng, nếu chưa có tài khoản thì chuyển qua đăng ký.
- Chức năng: hệ thống sẽ gửi link qua gmail để người dùng có thể đổi mật khẩu.

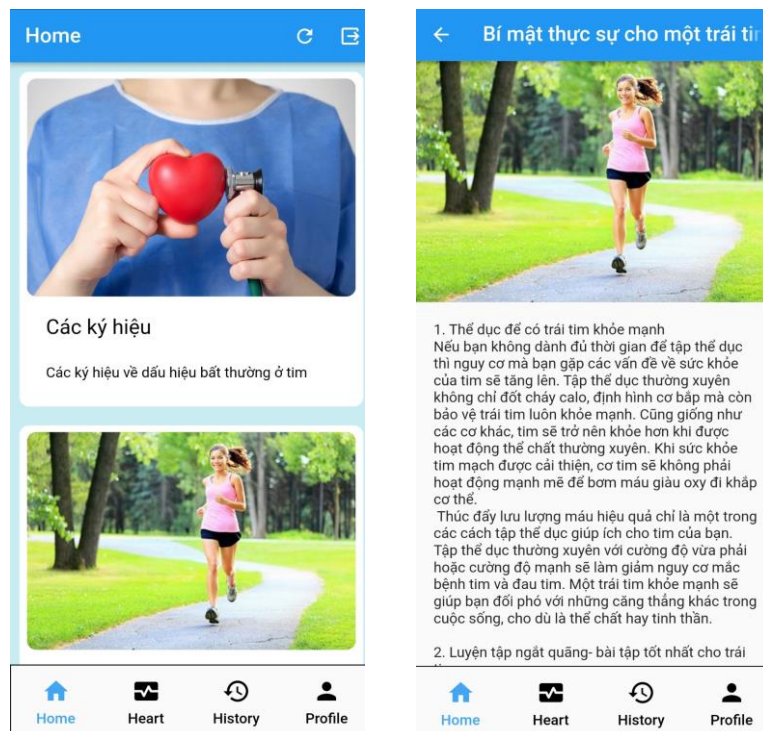
The image displays three mobile application screens for user authentication:

- Log In:** Features input fields for 'Email' and 'Password', a blue 'Sign in' button, and links for 'Sign up' and 'Forget password'.
- Sign Up:** Includes input fields for 'Email', 'Password', 'Name', and 'Birth', a 'Gender' dropdown menu (set to 'Male'), a blue 'Sign up' button, and a 'Sign in' link.
- Forget password:** Features an input field for 'Email' with the placeholder 'Type email here' and a blue 'Reset password' button.

Hình 12: Giao diện chức năng đăng nhập, đăng ký, quên mật khẩu.

## ❖ Trang chủ

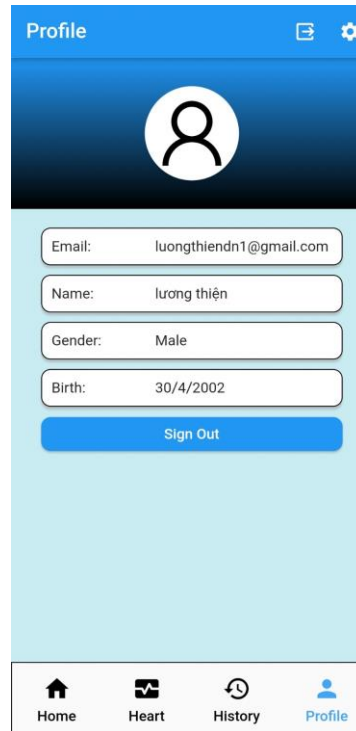
- Người dùng có thể đọc các bài báo, các cách để giúp có một sức khỏe tốt, một trái tim khỏe mạnh.



Hình 13: Giao diện trang chủ.

### ❖ Trang thông tin cá nhân

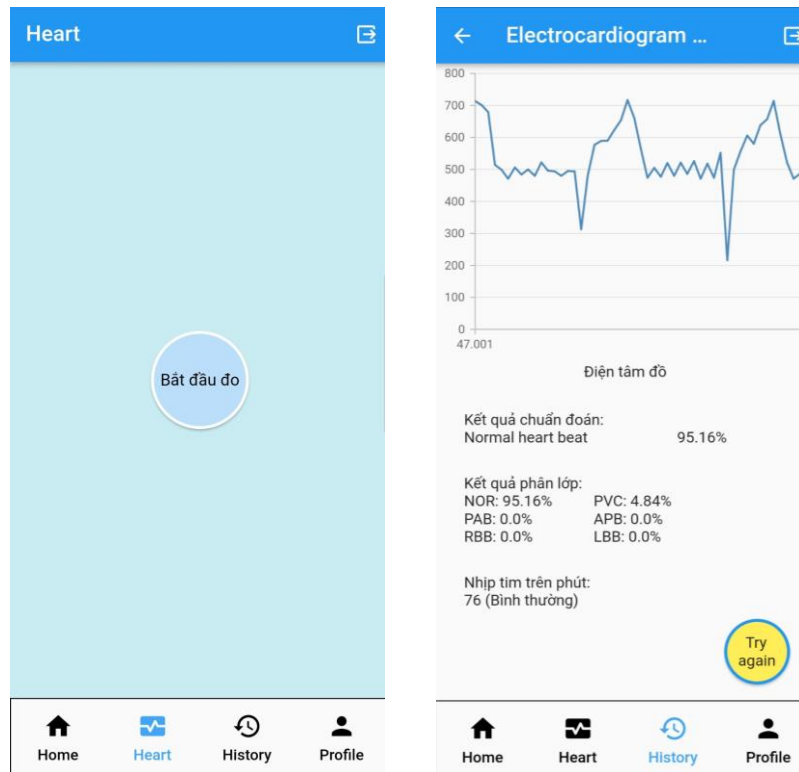
- Người dùng có thể xem và quản lý thông tin cá nhân của chính mình.



Hình 14: Giao diện thông tin cá nhân.

### ❖ Chức năng đo và xem kết quả

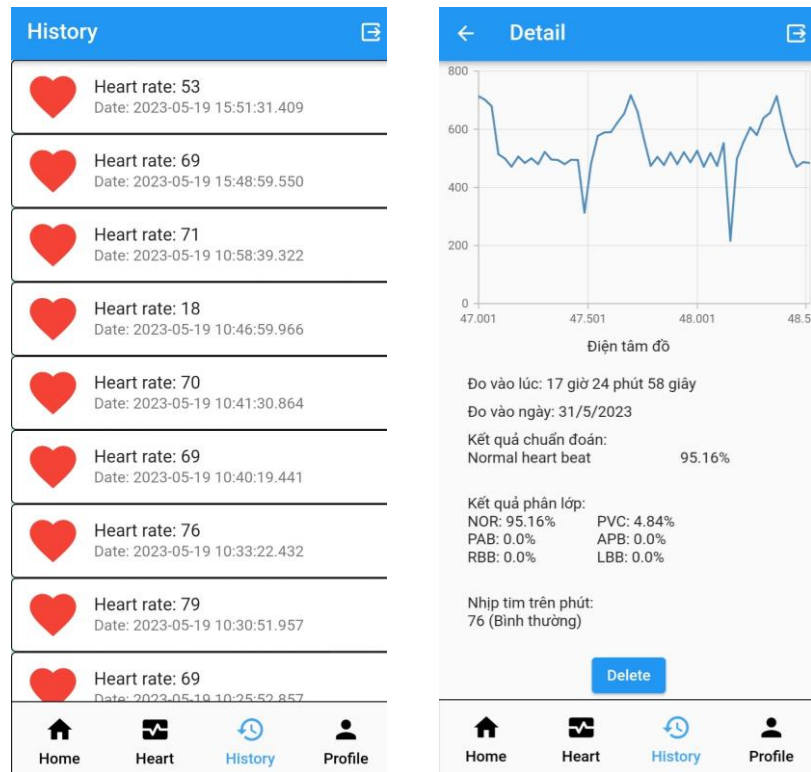
- Người dùng ấn nút đo, thì bắt đầu sẽ chạy ra điện tâm đồ và server sẽ trả về kết quả đo ở trang kết quả đo.
- Người dùng có thể sau khi ấn đo, kết quả sẽ trả về trang này, người dùng có thể lưu kết quả lần đo này, app sẽ chuyển dữ liệu về trang lịch sử đo.



Hình 15: Giao diện chức năng đo và xem kết quả.

❖ Chức năng xem lại lịch sử đo

- Người dùng có thể xem được các kết quả đo của mình trước đây.
- Ở trang lịch sử đo, người dùng có thể xem được thời gian mình đo là từ lúc nào ngày bao nhiêu, và xem được nhịp tim lúc đó.

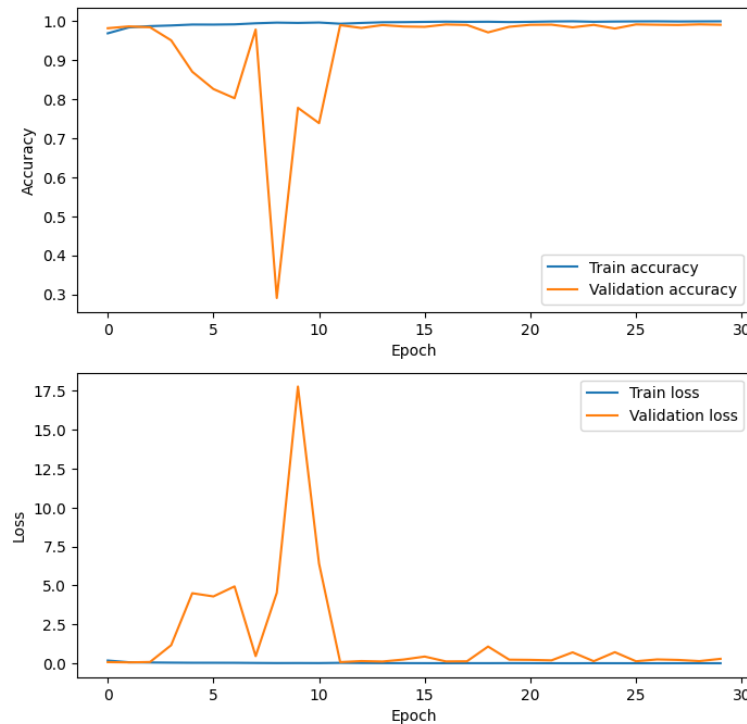


Hình 16: Giao diện chức năng xem lịch sử đo.

### 3.5. Kết quả mô hình phát hiện bất thường trong nhịp tim

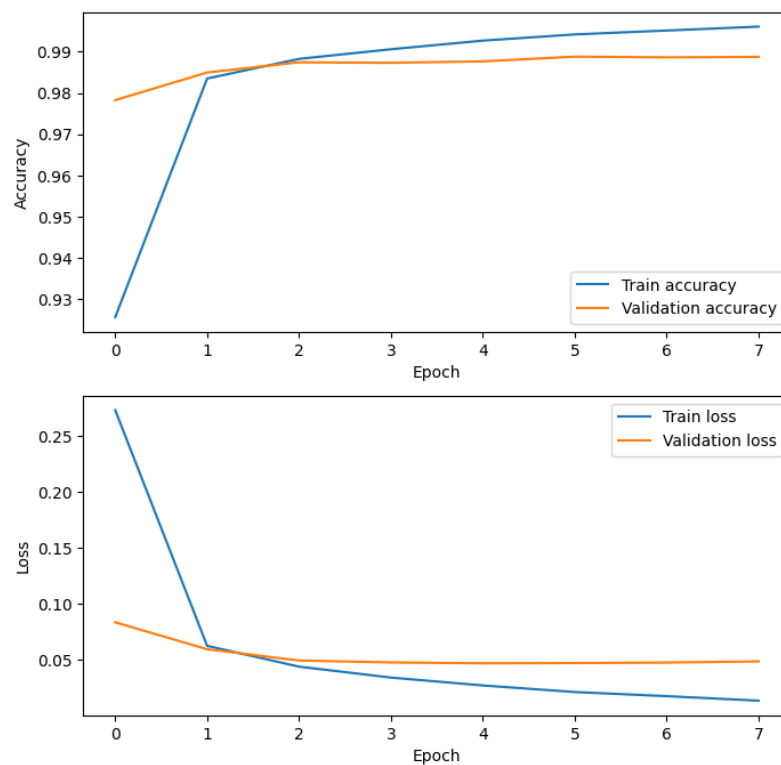
#### 3.5.1. Kết quả trên tập huấn luyện

##### ❖ Model 1



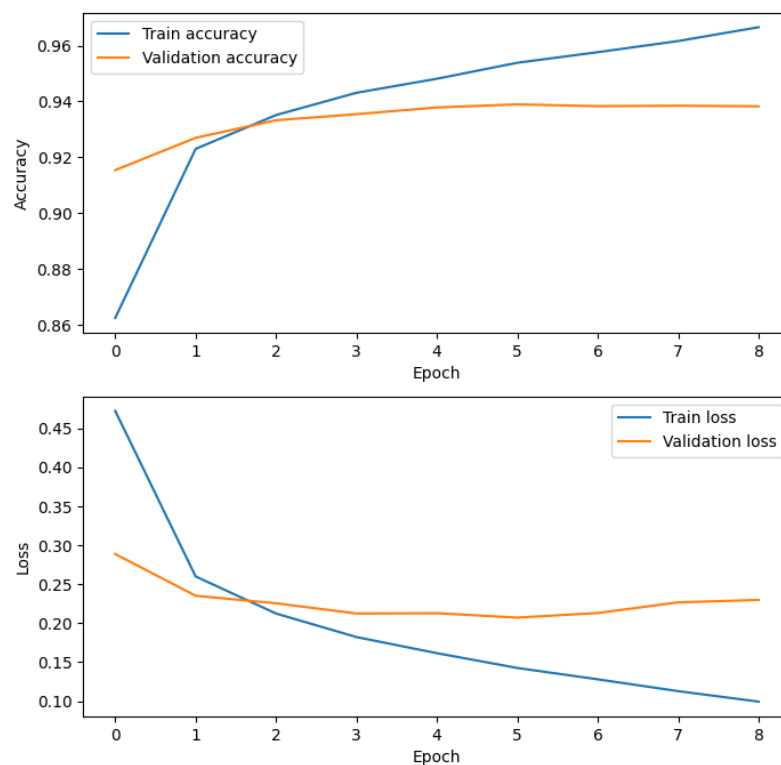
Hình 17: Kết quả huấn luyện trên model 1

## ❖ Model 2



Hình 18: Kết quả huấn luyện trên model 2.

## ❖ Model 3



Hình 19. Kết quả huấn luyện trên model 3.



### 3.5.2. Kết quả kiểm thử mô hình

#### ❖ Metrics kiểm thử mô hình

- **Accuracy:** là độ chính xác, được sử dụng để đánh giá mức độ hoạt động chính xác của một mô hình.

$$\text{Accuracy} = \frac{\text{Số lượng dự đoán đúng}}{\text{Tổng số lượng dự đoán}}$$

- **Precision:** là độ chính xác, được sử dụng để đánh giá độ chính xác của phép phân lớp positive trong mô hình. Với mức Precision cao hơn cho thấy độ chính xác của phép phân lớp positive của mô hình cao hơn.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

- **Recall:** là một thước đo hiệu quả của phân lớp dương tính, cho biết tỷ lệ các mẫu positive thực sự được phân lớp đúng là positive.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

- **F1:** là một chỉ số được sử dụng để đánh giá hiệu quả của một mô hình phân lớp. Nó kết hợp cả độ chính xác (precision) và độ toàn diện (recall) thông qua hàm trung bình hài hòa.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

#### ❖ Kết quả kiểm thử

Tiến hành kiểm thử 3 mô hình trên với các bộ dữ liệu kiểm thử tương ứng. Trong đó, các metrics precision, recall và f1-score được tính toán trên lớp Normal. Ta có bảng kết quả như sau.

Model	Tập test	precision	recall	f1-score	Accuracy
Model 1	1.1	0.99	0.99	0.99	0.99
	1.2	0.93	1.00	0.96	0.53
Model 2	1.1	0.99	1.00	0.99	0.99
	1.2	0.95	1.00	0.97	0.53
Model 3	2.1	0.97	0.99	0.98	0.96
	2.2	0.77	0.90	0.83	0.64

Bảng 13: Bảng kết quả kiểm thử mô hình (%)

#### ❖ Nhận xét

- Trên tập dữ liệu kiểm thử được xử lý dựa trên đỉnh đã được đánh dấu trong bộ dữ liệu gốc. Các metrics kiểm tra mô hình đều cho kết quả tốt (trên 90%).
- Trên tập dữ liệu kiểm thử được xử lý dựa trên thuật toán tìm đỉnh. Kết quả kiểm thử giảm xuống ở mức trung bình – khá. Nguyên nhân có thể vì nhiều loại bệnh sẽ phụ thuộc vào một số đỉnh khác nhau như Q và S. Do đó khi tìm bằng đỉnh R có thể dẫn đến nhiều dữ liệu bị sai sót làm cho độ chính xác giảm xuống.

#### 3.5.3. Kết quả sử dụng thực tế

- Nhóm tiến hành sử dụng cả ba mô hình trên để sử dụng trong thực tế. Sau đây là bảng kết quả tổng hợp khi đo đạc trên 4 người bình thường. Mỗi người được đo tổng cộng 5 lần liên tiếp.

Người dùng	Mô hình	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5
Cao Kiều Văn Mạnh	Model 1	32.14%	36.51%	23.81%	40.32%	33.9%
	Model 2	16.07%	7.94%	1.59%	16.13%	6.78%
	Model 3	92.86%	90.48%	95.24%	98.39%	96.61%
Trần Đình Minh Khoa	Model 1	25%	26.67%	33.93%	18.75%	33.9%
	Model 2	14.29%	15%	19.64%	10.94%	16.97%
	Model 3	89.29%	90%	92.73%	87.5%	82.76%
Lương Thiện	Model 1	1.49%	0%	1.59%	0%	4.48%
	Model 2	0%	0%	0%	0%	0%
	Model 3	77,61%	84.85%	79.37%	76.92%	88.06%
Nguyễn Quốc Cường	Model 1	0%	2.39%	2.65%	1.93%	0%
	Model 2	0%	0%	0%	0%	0%
	Model 3	93.02%	87.75%	86.62%	88.23%	90.31%

Bảng 14. Bảng kết quả sử dụng thực tế.

#### ❖ Đánh giá

- Với các tập kiểm thử được tạo dựa trên đỉnh đã được đánh dấu từ trước (1.1, 2.1), cả ba mô hình đều cho kết quả tốt với độ chính xác và các metrics khác như precision, recall, f1 trên 90%
- Với các tập kiểm thử được tạo bằng thuật toán xác định đỉnh mà nhóm sử dụng (1.2, 2.2), cả ba mô hình đều cho độ chính xác khá thấp. Nguyên nhân có thể vì thuật toán xác định đỉnh chưa quá chính xác cũng như một số lớp bệnh không phụ thuộc vào đỉnh R dẫn đến sai sót.
- Dựa trên bảng kết quả trên, để đảm bảo độ chính xác của mô hình là tốt nhất, nhóm lựa chọn model 3 để sử dụng trên hệ thống.

## **4. Kết luận**

### **4.1. Đánh giá sản phẩm**

#### **❖ Sản phẩm phần cứng**

- Cảm biến: hoạt động tương đối ổn định, tuy nhiên dễ bị ảnh hưởng bởi các yếu tố ngoại cảnh dẫn đến kết quả dữ liệu đo được bị nhiễu khá nhiều.
- Sản phẩm: sản phẩm đã được lắp đặt hoàn chỉnh, gọn gàng và tương đối dễ sử dụng.

#### **❖ Ứng dụng di động**

- Ứng dụng di động xây dựng được hoạt động tốt và ổn định.
- Tuy nhiên giao diện và trải nghiệm người dùng trên ứng dụng chưa thực sự được tối ưu.

#### **❖ Mô hình phát hiện tín hiệu ECG bất thường**

- Mô hình đưa ra dự đoán khá tốt về tín hiệu ECG bất thường của người dùng.
- Tuy nhiên vẫn còn nhiều hạn chế về dữ liệu và mô hình huấn luyện.
- Chưa có dữ liệu y khoa phù hợp để kiểm thử chính xác nhất hiệu năng của mô hình.

### **4.2. Hướng phát triển**

- Phát triển thuật toán nhận dạng được nhiều lớp bệnh hơn và đưa ra kết quả chính xác hơn. Phát triển thuật toán để tìm đỉnh của tín hiệu ECG chính xác hơn.
- Mở rộng các chức năng trên ứng dụng di động như các bài báo về sức khỏe, thông báo cho người dùng,...
- Đưa ra thêm các đánh giá y tế chi tiết khi áp dụng hệ thống vào thực tế.

## 5. Danh mục tài liệu tham khảo

- [1] Tae Joon Jun, Hoang Minh Nguyen, Daeyoun Kang, Dohyeun Kim, Daeyoung Kim, Young-Hak Kim, *ECG arrhythmia classification using a 2-D convolutional neural network*, <https://arxiv.org/abs/1804.06812>
- [2] Random Nerd Tutorials, *ESP8266 NodeMCU: Getting Started with Firebase (Realtime Database)*, <https://randomnerdtutorials.com/esp8266-nodemcu-firebase-realtime-database/>
- [3] *Firebase Documentation*, <https://firebase.google.com/docs>
- [4] *Flutter documentation*, <https://docs.flutter.dev/>
- [5] Chung Pham Van, *[Deep Learning] Tìm hiểu về mạng tích chập (CNN)*, <https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>
- [6] Tuan Nguyen, *Bài 6: Convolutional neural network*, <https://nttuan8.com/bai-6-convolutional-neural-network/>
- [7] George Moody, Roger Mark, *MIT-BIH Arrhythmia Database*, <https://www.physionet.org/content/mitdb/1.0.0/>
- [8] DFRobot, *Product Wiki - DFRobot Heart Rate Monitor Sensor*, [https://wiki.dfrobot.com/Heart\\_Rate\\_Monitor\\_Sensor\\_SKU\\_SEN0213](https://wiki.dfrobot.com/Heart_Rate_Monitor_Sensor_SKU_SEN0213)
- [9] Khiem H. Le, Hieu H. Pham, Thao BT. Nguyen, Tu A. Nguyen, Tien N. Thanh, Cuong D. Do, *LightX3ECG: A Lightweight and eXplainable Deep Learning System for 3-lead Electrocardiogram Classification*, <https://arxiv.org/abs/2207.12381v1>
- [10] Mohsen Nabian, Yu Yi , Jolie Wormwood, Karen S Quigley, Lisa F Barrett, Sarah Ostadabbas, *An Open-Source Feature Extraction Tool for the Analysis of Peripheral Physiological Data*, <https://pubmed.ncbi.nlm.nih.gov/30443441/>