



**Palestine Polytechnic University**  
**College of IT and Computer Engineering**  
**Computer System Engineering Department**

**Project Title:**

Remote follow up healthcare system

**Group Members:**

Basil Saleem Samouh

Raed Nseer Shawamreh

Nadim Mahmoud Atawneh

**Supervisor:**

Dr. Mohammed Aldasht

**Date:**

2 Jun, 2021

## الإهداء

إلى المعلم الاول وشمعة هذا الكون على مر الزمان سيدنا وحبينا "محمد عليه الصلاة والسلام"

يا من أحمل اسمك بكل فخر

يا من يرتعش قلبي لذكرك

يا من أودعتني الله أهديك هذا البحث أبي

إلى حكمتي ..... وعلمي

إلى أدبي ..... وحملي

إلى ينبوع الصبر والتفاؤل والأمل

إلى كل من في الوجود بعد الله ورسوله أمي الغالية

إلى سندي وقوتي وملادي بعد الله

إلى من آثروني على أنفسهم

إلى من علموني علم الحياة

إلى من أظهروا لي ما هو أجمل من الحياة إخوتي

إلى من سأفتقدهم ..... وأتمنى أن يفتقدوني

إلى من جعلهم الله إخوتي بالله ..... ومن أحببتهم بالله طلاب هندسة انظمة الحاسوب

## الشكر والعرفان

بسم الله الرحمن الرحيم، والحمد لله رب العالمين الذي وفقنا وأعاننا على إنهاء هذا البحث والخروج به بهذه الصورة المتكاملة، فبالأمس القريب بدأنا مسيرتنا التعليمية ونحن نتحسس الطريق برهبة وارتباك، فرأينا أن هندسة أنظمة الحاسوب هدفاً سامياً وحباً وغاية تستحق السير لأجلها، وإن بحثنا يحمل في طياته طموح شباب يحلمون أن تكون أمتهم العربية كالشامة بين الأمم.

نتقدم بجزيل الشكر والامتنان إلى جامعتنا " جامعة بوليتكنك فلسطين " التي احتضنتنا طوال فترة دراستنا في كلية هندسة الحاسوب وتكنولوجيا المعلومات. ولابد لنا ونحن نخطو خطواتنا الأخيرة في الحياة الجامعية من وقفة نعود إلى أعوام قضيناها في رحاب الجامعة مع أساتذتنا الكرام الذين قدموا لنا الكثير باذلين بذلك جهوداً كبيرة في بناء جيل فننتقدم بجزيل الشكر والتقدير لهم جميعاً.

وانطلاقاً من مبدأ أنه من لا يشكر الناس لا يشكر الله ، فإننا نتوجه بالشكر الجزيل للأستاذ الدكتور محمد الدشت الذي رافقنا في مسيرتنا لإنجاز هذا البحث وكانت له بصمات واضحة من خلال توجيهاته وانتقاداته البناءة والدعم الأكاديمي، كما نشكر عائلاتنا التي صبرت وتحملت معنا ورفدتنا بالكثير من الدعم على جميع الأصعدة، ونشكر الأصدقاء والأحباب وكل من قدم لنا الدعم المادي أو المعنوي.

وفي النهاية ننتقدم بجزيل الشكر والعرفان إلى كل من ساهم في إنجاز هذا العمل المتواضع.

# Abstract

With an improvement in technology and miniaturization of sensors, there have been attempts to utilize the new technology in various areas to improve the quality of human life. One main area of research that has seen an adoption of the technology is the healthcare sector.

As a result, this project is an attempt to solve a healthcare problem currently society is facing. The main objective of the project was to design a remote healthcare system. It's comprised of three main parts. The first part being, detection of patient's vitals using sensors, second for sending data to a server and the last part was providing the detected data for remote viewing on a mobile application. Remote viewing of the data enables a doctor or guardian to monitor a patient's health progress away from hospital premises. Mobile application can get external measurements like Blood Pressure, Body temperature and Blood Sugar Level.

This project used some mid-range cost components to build this system, which is available in laboratories, electronics shops and pharmacies, which these components are ESP8266, MAX30100, and AD8232 ECG.

The aim of the project was to come up with a Remote Health Monitoring System that can be made with locally available sensors with a view to making it affordable if it were to be mass produced.

Hence the proposed architecture collects the sensor data through ESP8266 microcontroller and relays it to the server where it is processed and analyzed for remote viewing.

After the project is completed, we expect all of its objectives to be achieved by designing a system of doing the necessary checks for heart patients to separate and reduce the gathering in health centers, using a mobile application interface between the doctor and the patient to collect all the measurements and examinations in one platform.

# Table of Contents

الإهداء.....	I
الشكر والعرفان .....	II
Abstract.....	III
Table of Contents .....	IV
List of Figures.....	V
Chapter 1 Introduction .....	1
1.1 Overview.....	1
1.2 Motivation .....	1
1.3 Importance .....	1
1.4 System Objectives .....	2
1.5 Short Description of the System .....	2
1.6 Problem Statement.....	3
Chapter 2 Background.....	4
2.1 Overview.....	4
2.2 Theoretical Background .....	4
2.3 Short description of the project hardware components.....	5
Chapter 3 System Design.....	12
3.1 Overview.....	12
3.2 Context diagram of the system .....	12
3.2.1 Main function of the system:.....	12
3.3 Detailed design.....	13
3.3.1 ESP8266 .....	13
3.3.2 MAX30100 .....	13
3.3.3 AD8232 ECG .....	14
3.4 Flowchart of the mobile application .....	16
3.5 Schematic diagrams .....	17
Chapter 4 Software .....	20
4.1 Overview.....	20
4.2 Software Implementation tools .....	20
4.3.2 Arduino IDE (Arduino Software) .....	20
4.3.3 Visual Studio Code.....	20

4.3.4 MongoDB .....	28
4.3.5 Heroku .....	28
4.3.5 Postman .....	29
4.4 Users.....	31
Chapter 5 Validation and Discussion.....	32
5.1 Overview.....	32
5.2 Units testing.....	32
5.2.1 Hardware Testing .....	32
5.2.2 Software Testing .....	36
5.2.3 How does it works .....	36
5.3 Implementation issues and challenges .....	36
Chapter 6 Conclusion.....	38
6.1 Overview.....	38
6.2 Summary .....	38
6.2 Future work .....	38
References.....	39

## List of Figures

<b>FIGURE 1: SHORT DESCRIPTION OF THE SYSTEM.....</b>	<b>2</b>
<b>FIGURE 2: SOFTWARE USE-CASE DIAGRAM .....</b>	<b>5</b>
<b>FIGURE 3: GENERAL DESCRIPTION OF PROJECT HARDWARE COMPONENTS .....</b>	<b>6</b>
<b>FIGURE 4: ESP8266, SOURCE [1].....</b>	<b>6</b>
<b>FIGURE 5: MAX30100 SENSOR, SOURCE [3] .....</b>	<b>7</b>
<b>FIGURE 6: MAX30100 FUNCTIONAL DIAGRAM, SOURCE [2].....</b>	<b>8</b>
<b>FIGURE 7: INTERFACING AD8232 ECG SENSOR WITH ESP8266, SOURCE [4] .....</b>	<b>9</b>
<b>FIGURE 8: AD8232 ECG SENSOR CONNECTION TABLE .....</b>	<b>10</b>
<b>FIGURE 9: AD8232 ECG SENSOR PLACEMENT ON BODY, SOURCE [4].....</b>	<b>10</b>
<b>FIGURE 10: CONTEXT DIAGRAM OF THE SYSTEM .....</b>	<b>12</b>
<b>FIGURE 11: BLOCK DIAGRAM OF THE SYSTEM.....</b>	<b>13</b>
<b>FIGURE 12: BH1790GLC HEART RATE MONITOR, SOURCE [7] .....</b>	<b>14</b>
<b>FIGURE 13: SIMPLIFIED DESIGN FOR AD8232 AND 3 PADS .....</b>	<b>15</b>
<b>FIGURE 14: FLOWCHART OF SMARTPHONE APPLICATION.....</b>	<b>16</b>
<b>FIGURE 15: SCHEMATIC DIAGRAM OF THE SYSTEM.....</b>	<b>17</b>
<b>FIGURE 16: SCHEMATIC DIAGRAM OF MAX30100 .....</b>	<b>18</b>
<b>FIGURE 17: SCHEMATIC DIAGRAM OF AD8232 ECG.....</b>	<b>19</b>

<b>FIGURE 20: PATIENT SCREEN EXTERNAL READING.....</b>	<b>23</b>
<b>FIGURE 21: DOCTOR MAIN SCREEN .....</b>	<b>24</b>
<b>FIGURE 22: DOCTOR SCREEN AFTER CLICK ON FIRST PATIENT NORMAL CASE.....</b>	<b>25</b>
<b>FIGURE 23: DOCTOR SCREEN AFTER CLICK ON FIRST PATIENT ABNORMAL CASE .....</b>	<b>26</b>
<b>FIGURE 24:EXAMPLE READING BLOOD PRESSURE.....</b>	<b>27</b>
<b>FIGURE 25: MODEL-VIEW-CONTROLLER (MVC) PATTERN.....</b>	<b>29</b>
<b>FIGURE 26: USER DATAFLOW DIAGRAM.....</b>	<b>31</b>
<b>FIGURE 27: SPO2 VALUE FOR HUMAN.....</b>	<b>32</b>
<b>FIGURE 28: HEART-RATE VALUE FOR HUMAN .....</b>	<b>33</b>
<b>FIGURE 29: ECG DIAGRAM IF IT CONNECTED.....</b>	<b>33</b>
<b>FIGURE 30: ALL CIRCUITS IN BREADBOARDS 1 .....</b>	<b>34</b>
<b>FIGURE 31: ALL CIRCUITS IN BREADBOARDS 2.....</b>	<b>34</b>
<b>FIGURE 33: FINAL RESULT 2.....</b>	<b>35</b>

# Chapter 1

## Introduction

### 1.1 Overview

It is a simplified system-based project that do some regular medical examinations for heart patients, and this system is called a remote follow up healthcare system, and the tests are: Blood Oxygen Level, ECG and Heart Rate Measurement.

These readings are uploaded and displayed on application in the smartphone in real time in front of the specialist doctor, and the time between the patient and the doctor is organized and also called-in in the event or when something is occurred to the patient.

### 1.2 Motivation

According what happen of the Covid-19 that Faced the whole world, which is spreading widely in human gatherings, and with the World Health Organization's warnings that states spacing and reduce overcrowding, it's supposed from us to be creative in creating new ideas to face this crisis, and help the WHO in terms of Creative Crisis Management, and taking advantage of the current experience that we have to complete this project.

### 1.3 Importance

The importance in the medical system is mainly focused on reducing human gatherings and waiting times in clinics and hospitals, the continuity of doctor-patient communication, and the ease of health care service.

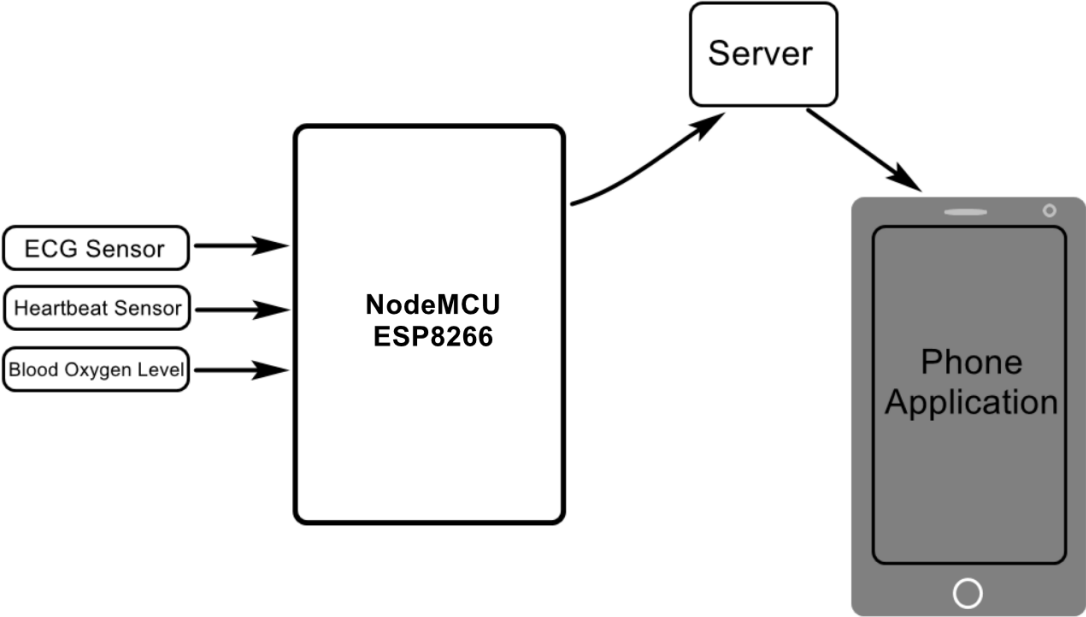


## 1.4 System Objectives

We will design a system of doing the necessary checks for heart patients, so that the system environment is similar to the environment that found in hospitals and specialized clinics, and the objectives can be summarized in the following points:

- To separate and reduce the gathering in health centers, thus reducing the rate of infection with the virus.
- Long-term follow-up of the patient's condition remotely.
- Direct communication between the patient and the doctor.

## 1.5 Short Description of the System



**Figure 1: Short description of the system**

As the shown in the figure above, the system is divided into three parts:

- 1) Inputs: which are represented in the four sensors, which are:
  - a) ECG (Electrocardiogram) sensor: records the pathway of electrical impulses through the heart muscle.

- b) Heartbeat Sensor: is an electronic device that is used to measure the heart rate.
  - c) Blood Oxygen Level sensor: is a medical device that indirectly monitors the oxygen saturation of a patient's blood.
- 
- 2) Process: The inputs are taken to the processing section, and then processed the data that is sent from the ESP32 for transfer to the dedicated server.
  - 3) Output: Receive information from the server after linking patient data to the dedicated application on the phone.

## 1.6 Problem Statement

The daily presence of patients in hospitals and clinics and their intense gathering, waiting for reviews and examinations, which causes a state of fatigue and severe exhaustion for patients.

And with the presence of the Covid-19 pandemic and the rapid increase of people infected with this virus, which directly threatens the lives of people with chronic diseases, especially heart patients, which led to the danger of their presence in cramped places.

# Chapter 2

## Background

### 2.1 Overview

This chapter describes briefly the theoretical background of the project, we will introduce a short description of the hardware and software parts that are used in the system.

### 2.2 Theoretical Background

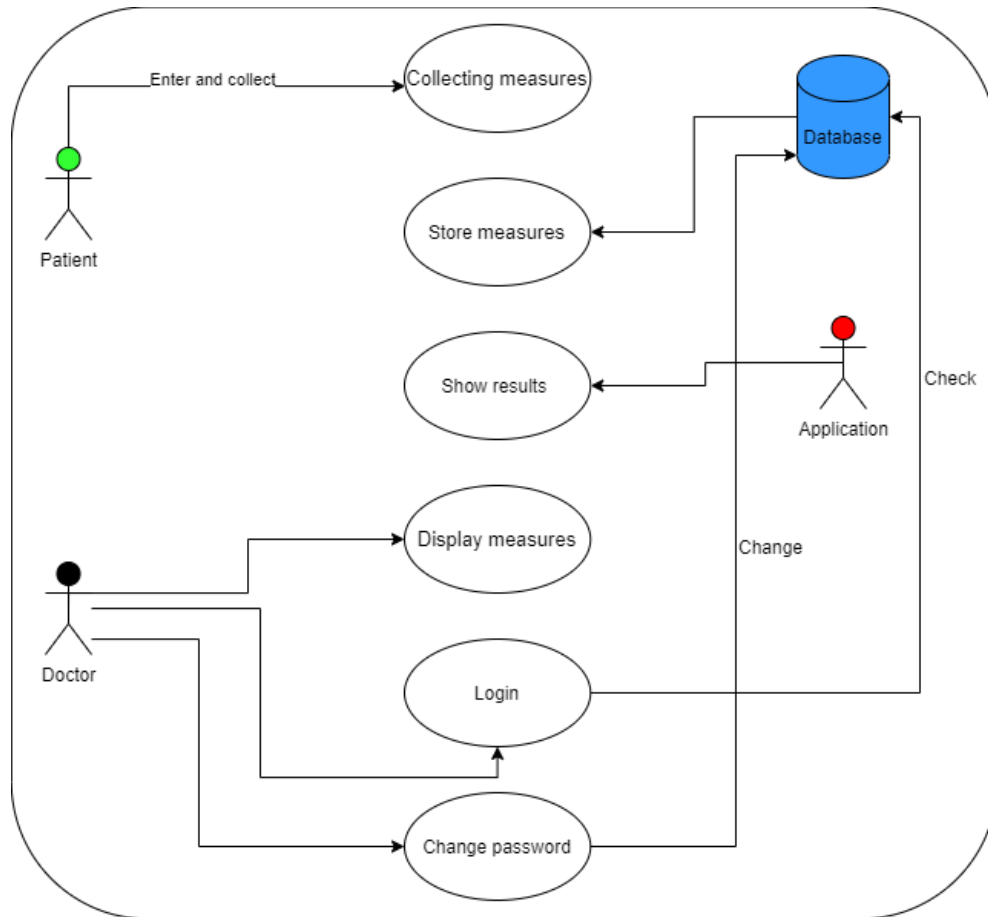
This project is designed to do in following-up patients through collecting vital signs and some important measures, remotely, for cardiovascular patients, during the normal activity of the patient. Of course, this will include some minimal training for the patient how to properly use the system, as these measures are taken by several sensors.

The system is separated into two parts, the first one is the hardware, which consists of 2 sensors connected to the NodeMCU ESP8266 and server. These components are connected with a voltage source.

The second part is the software, which contains a simple and easy-to-use mobile application, which will require a login process with a unique username and password, after that, tests are presented to the patient and the physician in the app. Moreover, it will be presented with dates and times for each result.

In the figure below, we present the software in a use-case diagram, as we see the system is collecting measures from the patient, then transfer these data to a database, which is built-in in application. To present these measurements it is necessary for the doctor to login with ID and password, and request display measures from the application,

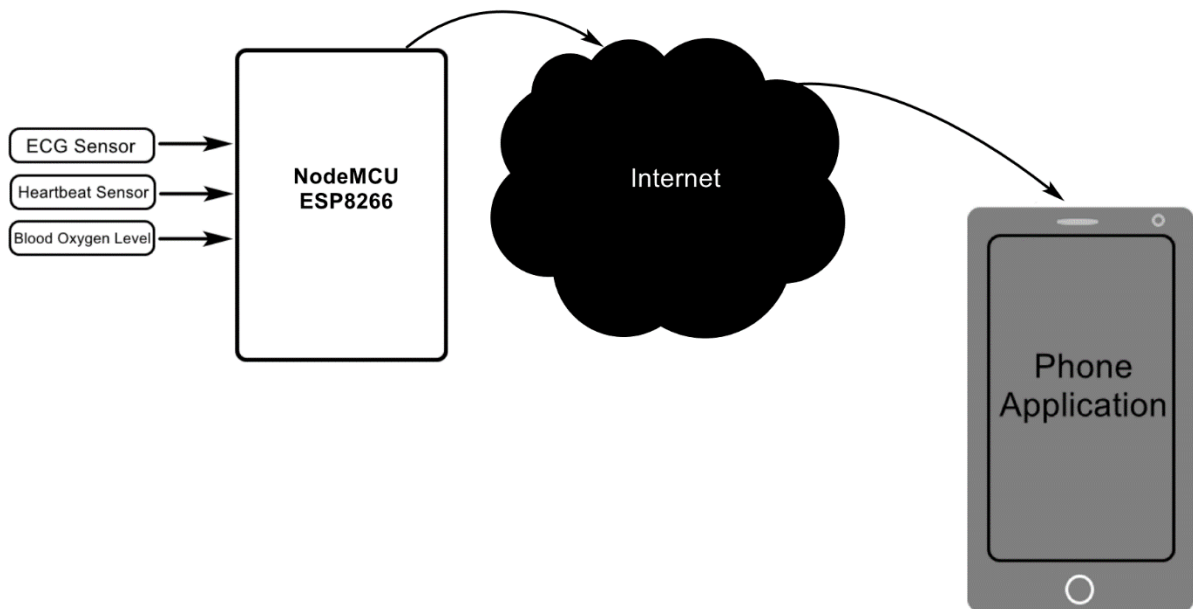
if there any problem with login, the application will reset the password, and continue the process.



**Figure 2: Software use-case diagram**

### 2.3 Short description of the project hardware components

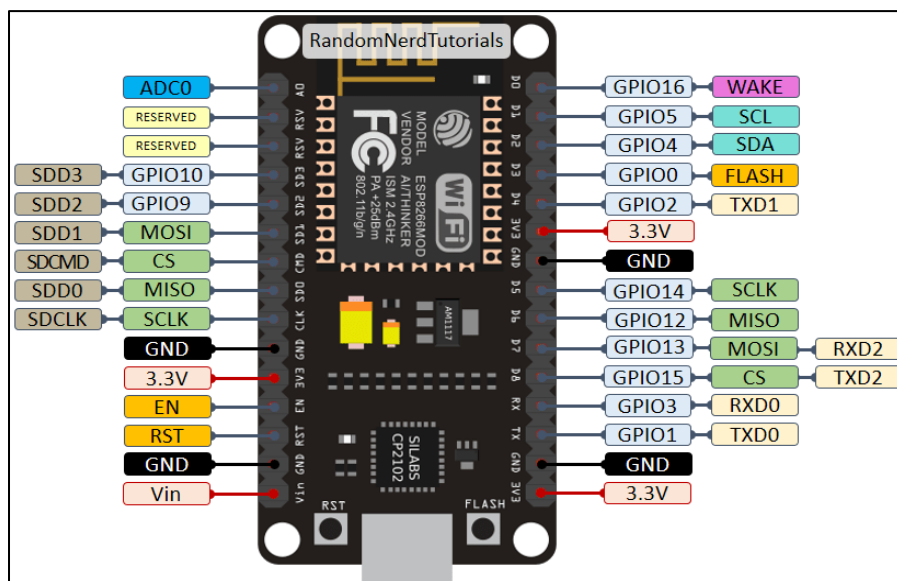
In this part we will illustrate the hardware components used in the system, in addition, we will introduce a short description of each component used in the system and why they are chosen.



**Figure 3: General description of project hardware components**

### 2.3.1 NodeMCU ESP8266

Is a series of low-cost, and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor and comes with 17 GPIO pins. Not all GPIOs are exposed in all ESP8266 development boards, some GPIOs are not recommended to use, and others have very specific functions [1].



**Figure 4: ESP8266, source [1]**

The ESP8266 peripherals include:

- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated PLL, regulators, and power management units
- Integrated temperature sensor
- Supports antenna diversity
- Power down leakage current of  $< 10\mu\text{A}$
- Integrated low power 32-bit CPU could be used as application processor

We chose it because it has the required pins for the project, in addition it is used for networking, using the module's Wi-Fi Antenna, with 32-bit RISC microprocessor core (single core processor) that able to connect to routers and transmit data that we got it from sensors to server and its low cost and much higher in processing speed if we compared to the Arduino Mega.

### 2.3.2 The MAX30100

The MAX30100 is an integrated pulse oximetry and heart-rate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals [2].

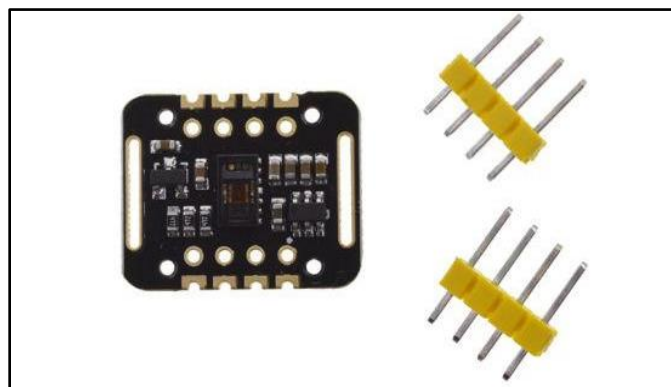
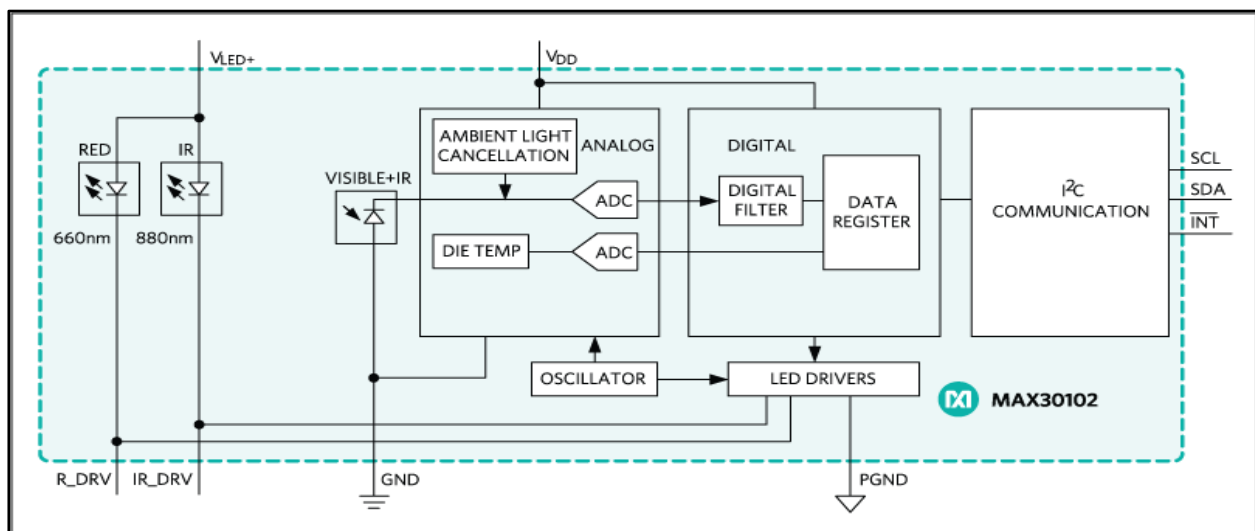


Figure 5: MAX30100 sensor, source [3]

### Key Features:

- ❖ Complete Pulse Oximeter and Heart-Rate Sensor.
- ❖ Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically.
- ❖ Ultra-Low-Power Operation Increases Battery Life for Wearable Devices.
- ❖ -40°C to +85°C Operating Temperature Range.

The MAX30100 operates on a single 1.8V power supply and a separate 3.3V power supply for the internal LEDs. Communication is through a standard I2C-compatible interface [2].



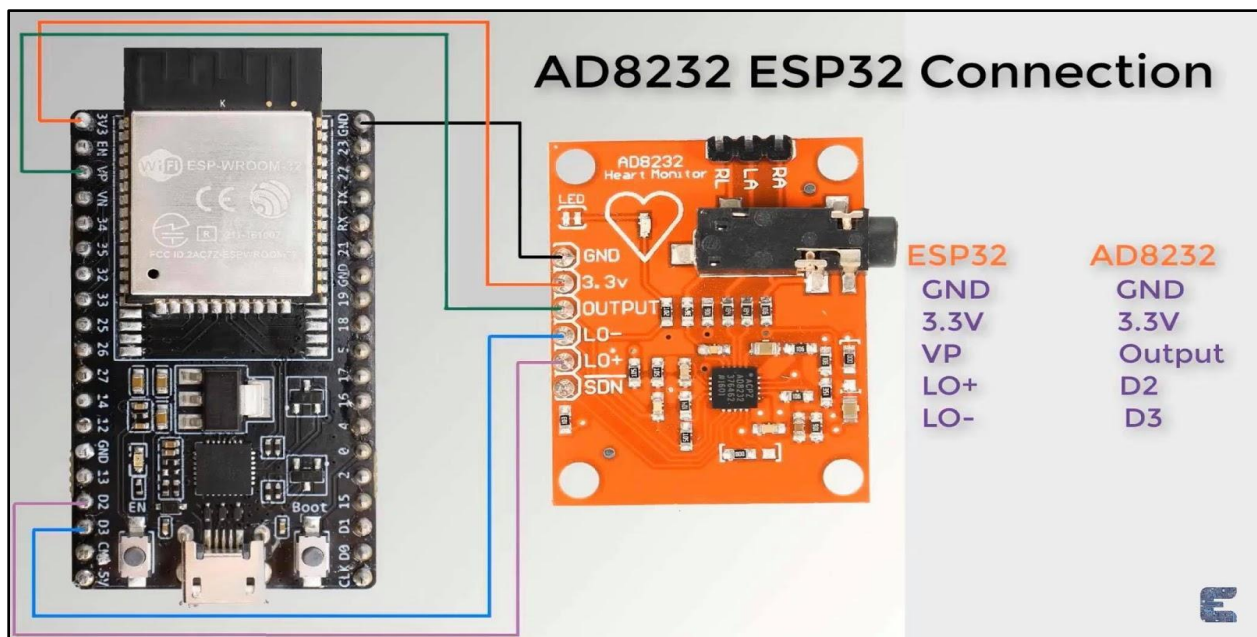
**Figure 6: MAX30100 Functional Diagram, source [2]**

This sensor was chosen for its ability to make two readings (Heart-Rate and Oximeter Biosensor) in one sensor instead of two. Also, to save wire connection and save energy consumption.

### 2.3.3 AD8232 ECG Sensor

An ECG is a digital recording of the electrical signals in the heart. It is also called an electrocardiogram or an EKG. The ECG is used to determine heart rate, heart rhythm, and other information regarding the heart's condition. ECGs are used to help diagnose heart arrhythmias, heart attacks, pacemaker function, and heart failure [4].

In addition, the plug-in for the cable is a male sound plug which will make the cable easily removed or inserted into the board. The sensor assembled on an arm pulse and a leg pulse [4].



**Figure 7: Interfacing AD8232 ECG Sensor with ESP8266, source [4]**

An AD8232 ECG Sensor has 3 pins of disposable electrodes attached directly to the chest or arms to detect every heartbeat, in addition the electrodes of the ECG sensor will convert heartbeat to electric signal so it makes the ECG sensor easy to connect with the ESP8266 [4].

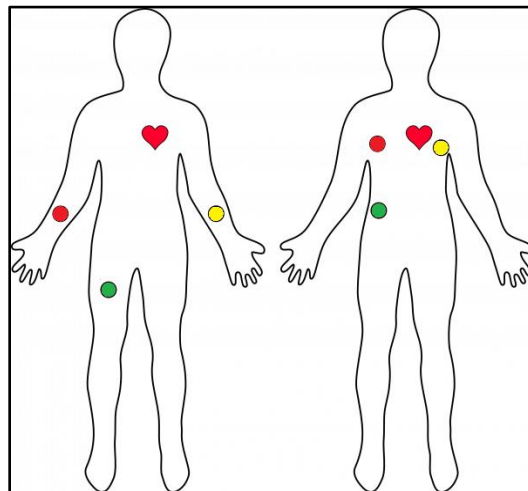


The AD8232 module breaks out nine connections from the IC:

Essential 6 pins: SDN, LO+, LO-, OUTPUT, 3.3V, GND

Board Label	Pin Function	Arduino Communication
GND	Ground	GND
3.3v	3.3v Power Supply	3.3v
OUTPUT	Output Signal	Any Analog pin
LO-	Leads-off-Detect -	Any Digital pin
LO+	Leads-off-Detect +	Any Digital pin
SDN	Shutdown	Not used

**Figure 8: AD8232 ECG Sensor connection table**



**Figure 9: AD8232 ECG Sensor Placement on Body, source [4]**

Also provided 3 pins on this board known as pads (red, yellow, green):

Red: RA (Right Arm),

Yellow: LA (Left Arm),

Green: RL (Right Leg).

All of every sensor electrode has methods to assemble in the body. So, training and tutorials are needed for patients.

It is recommended to snap the sensor pads on the leads before application to the body. The closer to the heart the pads are, the better the measurement.

We chose the AD8232 ECG Sensor in our project because it's very lightweight, slim, cheap and accurately measures continuous heartbeat and gives rate data of heartbeat.

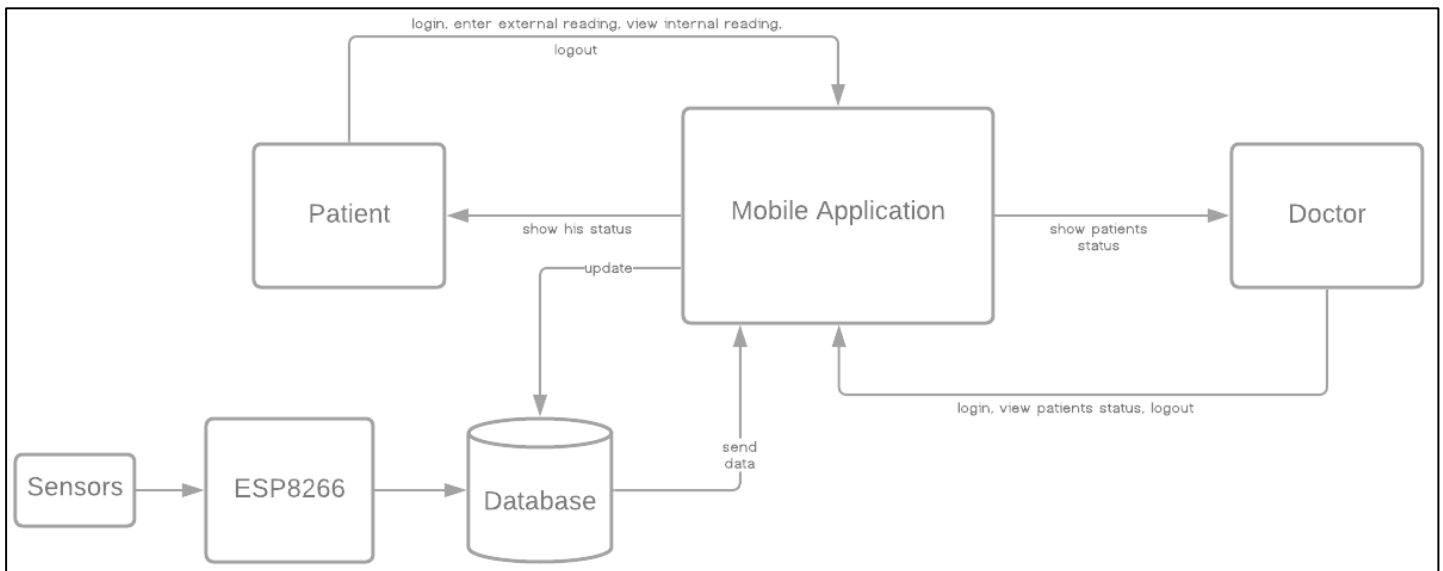
# Chapter 3

## System Design

### 3.1 Overview

In this chapter we will discuss the conceptual design of the system, it shows the system requirement analysis, a block diagram of the system, flow charts, schematic diagram and interface layout for healthcare application.

### 3.2 Context diagram of the system

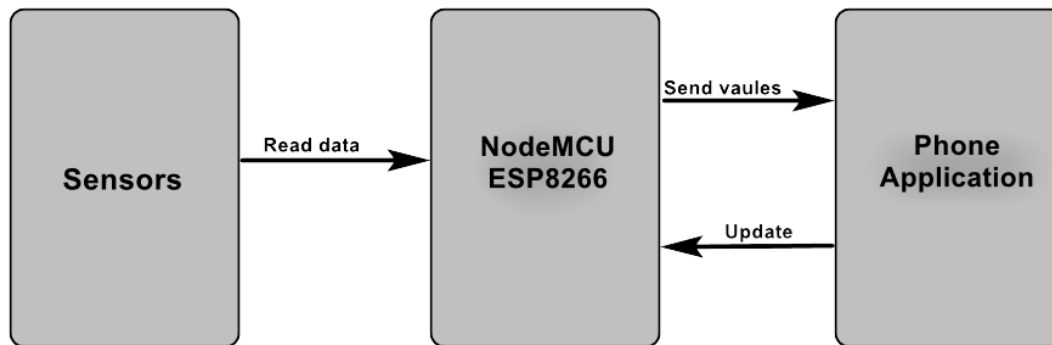


**Figure 10: Context diagram of the system**

#### 3.2.1 Main function of the system:

- Show the patient status.
- Make Communication between doctor and patient.
- Book the doctor (for the patient), and remote his measurements.

### 3.3 Detailed design



**Figure 11: Block diagram of the system**

In general, in the figure above, ESP8266 is going to read the data from the sensors as the inputs, to do some process and organize the data, then transmit these values to the phone application, to be available to show to the doctor in real time.

#### 3.3.1 ESP8266

We need a microcontroller in our project, the ESP microcontrollers were chosen and we chose the NodeMCU ESP8266 instead of the ESP32, even if ESP32 it comes with 48 pins compared to ESP8266 that have only 17 pins and higher processing speed (160/240 MHz to 80MHz), but it has the required pins and the project doesn't need higher processing speed, also both of them are cheap but Nodemcu8266 is cheaper than ESP32 (almost half).

For Microcomputer like Raspberry pi, ESP8266 is much cheaper than Raspberry pi and also Leads to the desired outcome.

#### 3.3.2 MAX30100

We need a sensor that calculates the blood oxygen level and sensor to heart-rate monitor, so we search and we discover a sensor for heart-rate called BH1790GLC, and its optical sensor for heart rate monitors is optimized for wearable devices such as smart

watches and sports bands, but we didn't find for blood oxygen level as individual sensor, so we search more and we find MAX30100 sensor that do these measurements as one sensor.



**Figure 12: BH1790GLC Heart Rate Monitor, source [7]**

### 3.3.3 AD8232 ECG

ECG (Electrocardiogram) sensor records the pathway of electrical impulses through the heart muscle, so we have chosen AD8232 from among many sensors such as MAX-ECG-Monitor and shield-EKG-EMG, because it's cheap and has many features such as:

- High signal gain ( $G = 100$ ) with dc blocking capabilities.
- Fast restore feature improves filter settling.
- Qualified for automotive applications like Fitness and activity heart rate monitors.

AD8232 design allows for an ultralow power analog-to-digital converter (ADC) or an embedded microcontroller to acquire the output signal easily. So, this sensor has 6 essential pins: SDN, LO+, LO-, OUTPUT, 3.3V, GND connected to ESP8266 and 3 electrodes connected to the body [8].

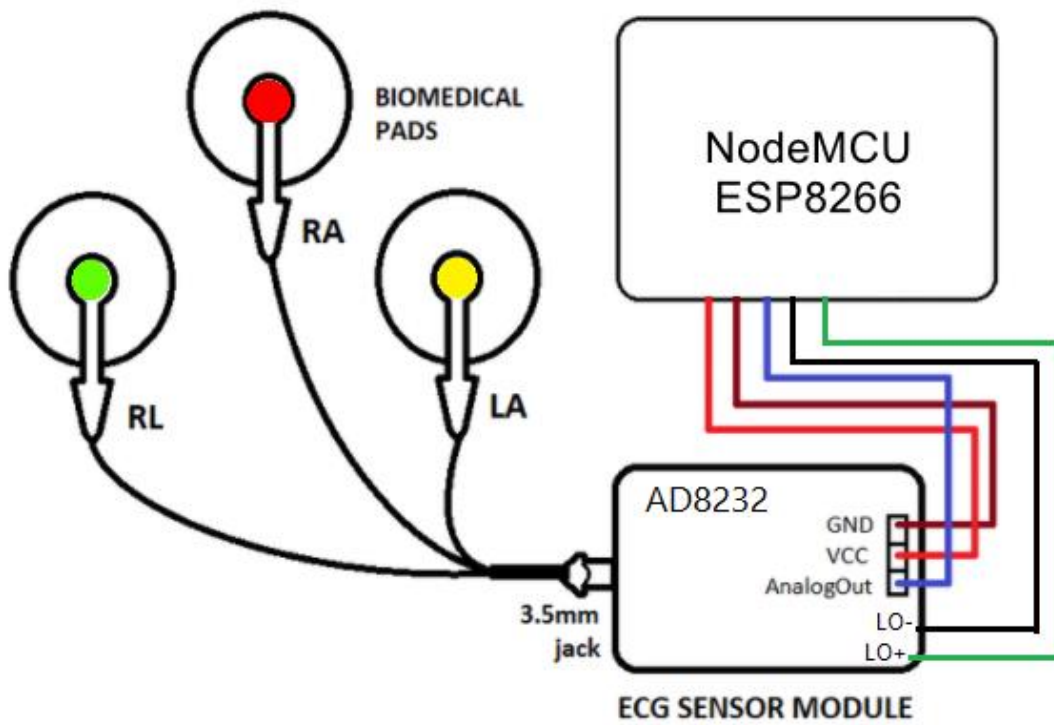
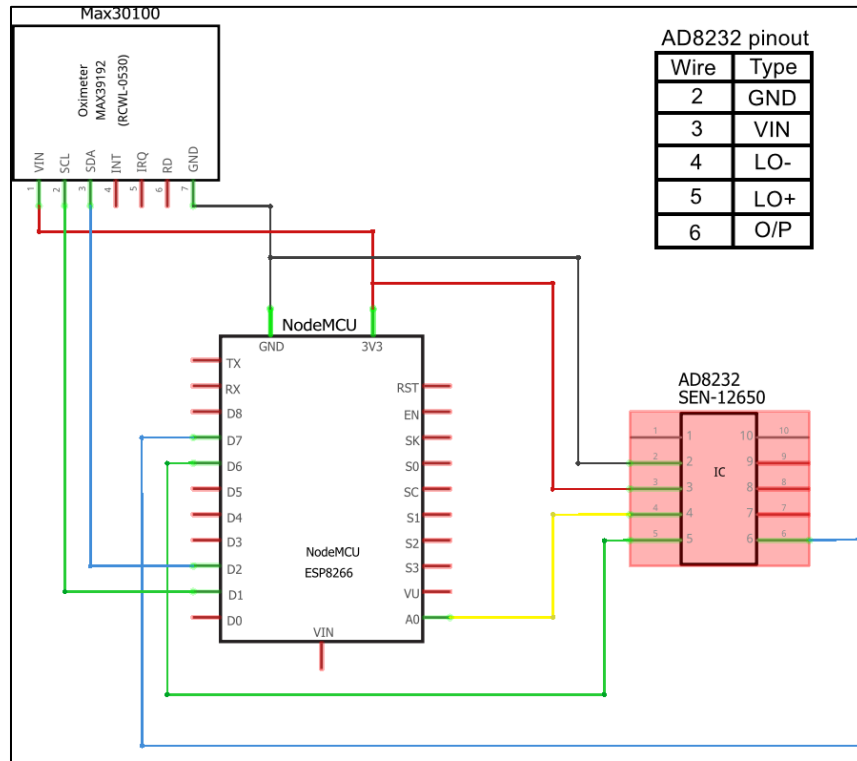


Figure 13: simplified design for AD8232 and 3 pads



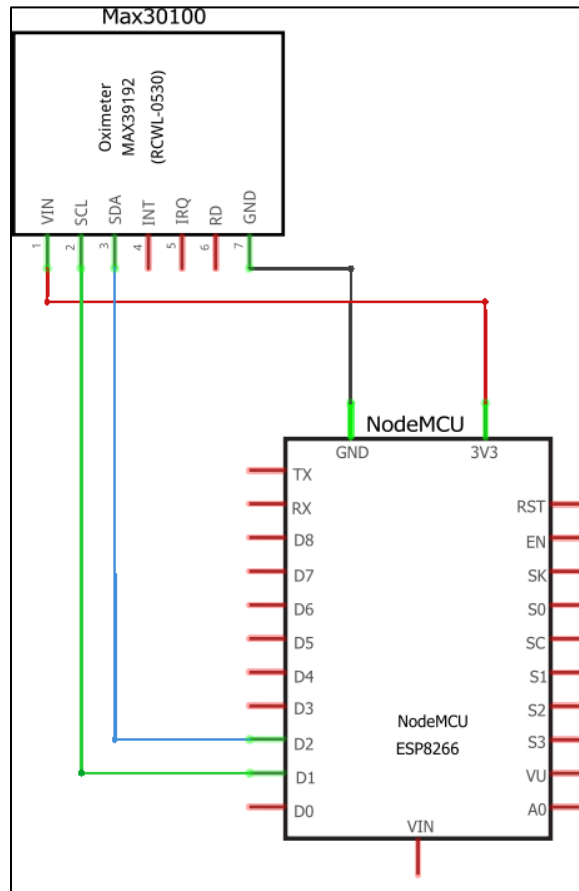
### 3.5 Schematic diagrams



**Figure 15: Schematic diagram of the system**

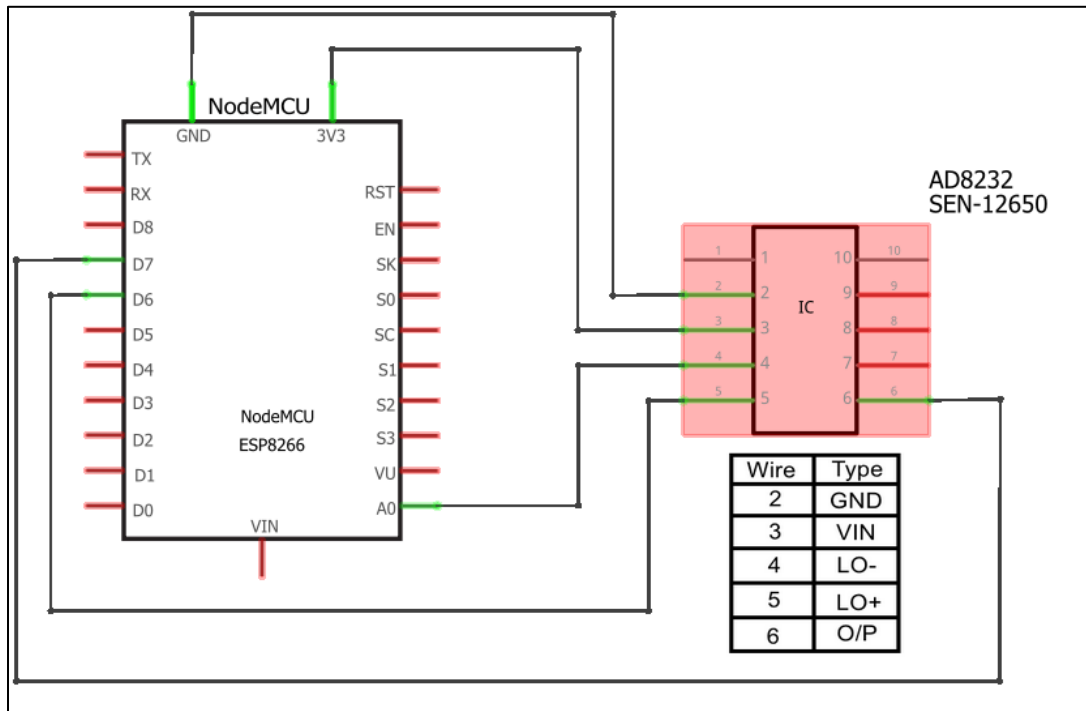
The figure above shows the hardware part of the project that includes the ECG sensor and MAX30100 sensor connected directly to the Nodemcu ESP8266, as well as connecting the wires to the dedicated pins.





**Figure 16: Schematic diagram of MAX30100**

Here is the schematic diagram of MAX30100 sensor connected to the Nodemcu ESP8266 separately from the rest of the other components of the system.



**Figure 17: Schematic diagram of AD8232 ECG**

Here is the schematic diagram of the AD8232 sensor connected to the Nodemcu ESP8266 separately from the rest of the other components of the system.

# Chapter 4

## Software

### 4.1 Overview

This chapter describes the implementations of the software that we used in this project, such as the IDEs used to build the project codes, and the user interface.

### 4.2 Software Implementation tools

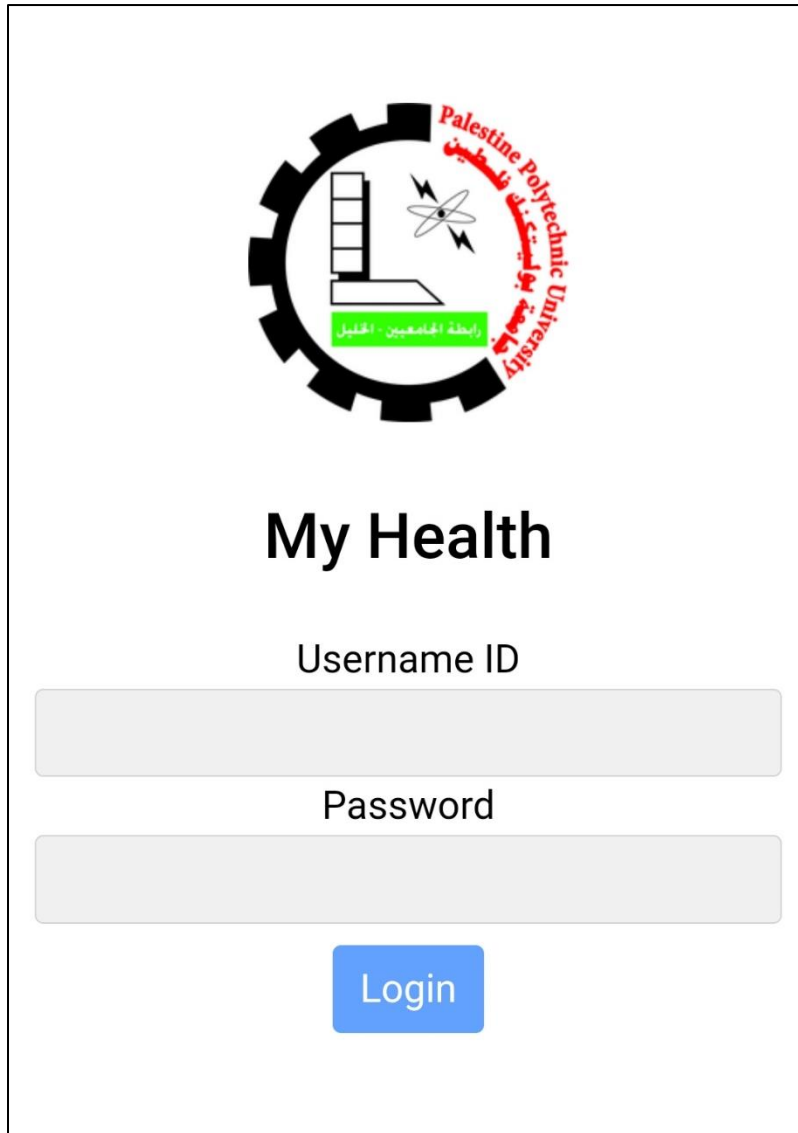
This section will provide some information about the main programs and software technologies used in our project:

#### 4.3.2 Arduino IDE (Arduino Software)

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board and ESP microcontrollers [9]. In our project, we used Arduino IDE to writing code (each component owns part from code to get data and send to server) then make sure there are no errors to upload the program into ESP8266.

#### 4.3.3 Visual Studio Code

Is a lightweight but powerful source code editor which runs on computer and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js [10]. We used VSC for build front-end and back-end web page that will be converted to phone application (APK extension).



**Figure 18: Main screen of Smartphone application**



**Figure 19: Patient main screen**

In general, this is the main screen after the system discovers the ID for a patient, and the first option is for the patient to enter external measurements, which include optional medical examinations like: Body temperature, Blood Pressure and Blood Sugar Level, and the second one is for viewing all his measurements.

تسجيل الخروج» رجوع

## القراءات الخارجية

ضغط الدم

نسبة السكر في الدم

درجة حرارة الجسم

حفظ القراءات

**Figure 18: Patient screen external reading**

After click on first circle button in patient main screen, it will direct the user to external reading screen that allow patient to enter his external measurements.

## My Health

تسجيل الخروج

basil

الحالة	الاسم	الرقم
مستقر	nadim	1
غير مستقر	نديم محمود عطاونة	2

## My Health

تسجيل الخروج

basil

الحالة	الاسم	الرقم
غير مستقر	nadim	1
غير مستقر	نديم محمود عطاونة	2

**Figure 19: Doctor main screen**

This is the main screen after the system discovers the ID for a doctor, which the last column in left side is described if measurements are abnormal or normal for his patients, in this case we have two cases, abnormal and normal for first patient.

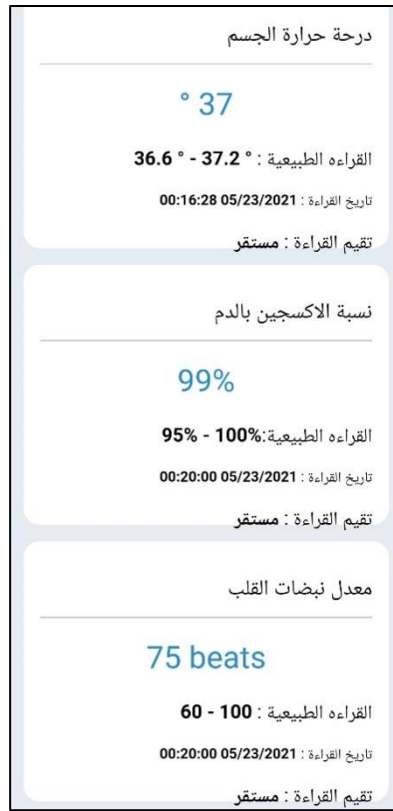
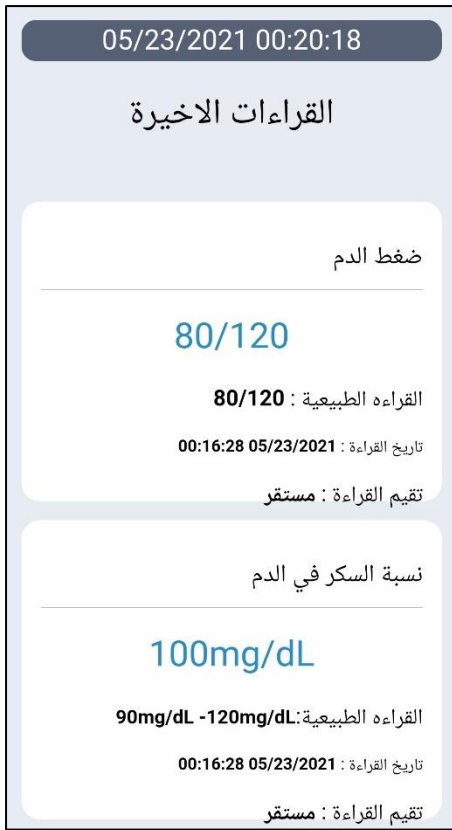


Figure 20: Doctor screen after click on first patient normal case

In this term we see that the statue of first patient is normal and all his measurements also normal, and doctor can check last four measurements for this patient.



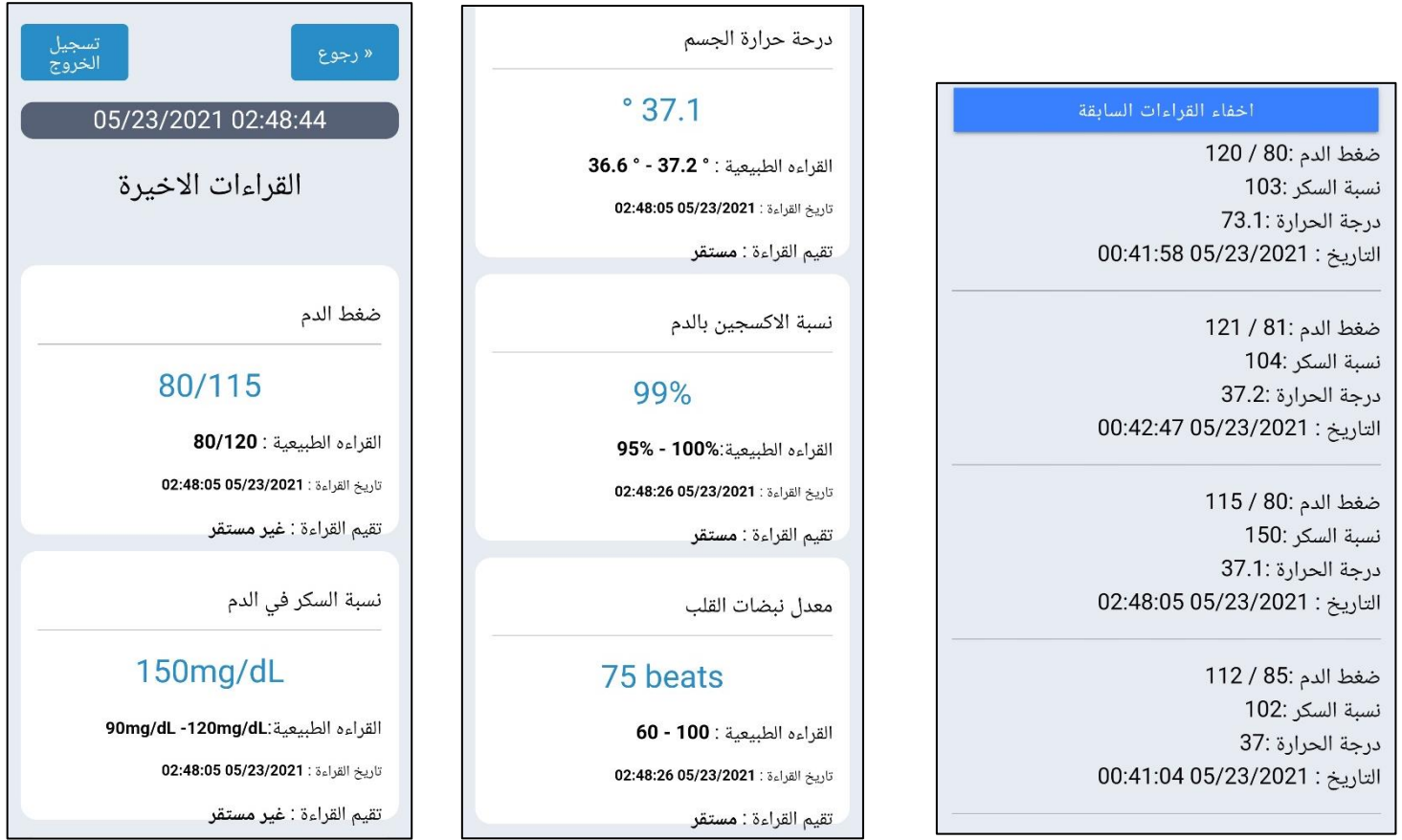


Figure 21: Doctor screen after click on first patient abnormal case

In this term we see that the statue of first patient is abnormal and all his measurements also abnormal, and doctor can check last four measurements for this patient.



**Figure 22:Example Reading Blood pressure**

The figure above explains that the patient or his assistants must use this device and take its reading, then enter it in the place designated for it in the application (self-use).

#### 4.3.4 MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas [11]. We used it for build non-relational database (NoSQL), because the application will have a many user's information and details to deal with.

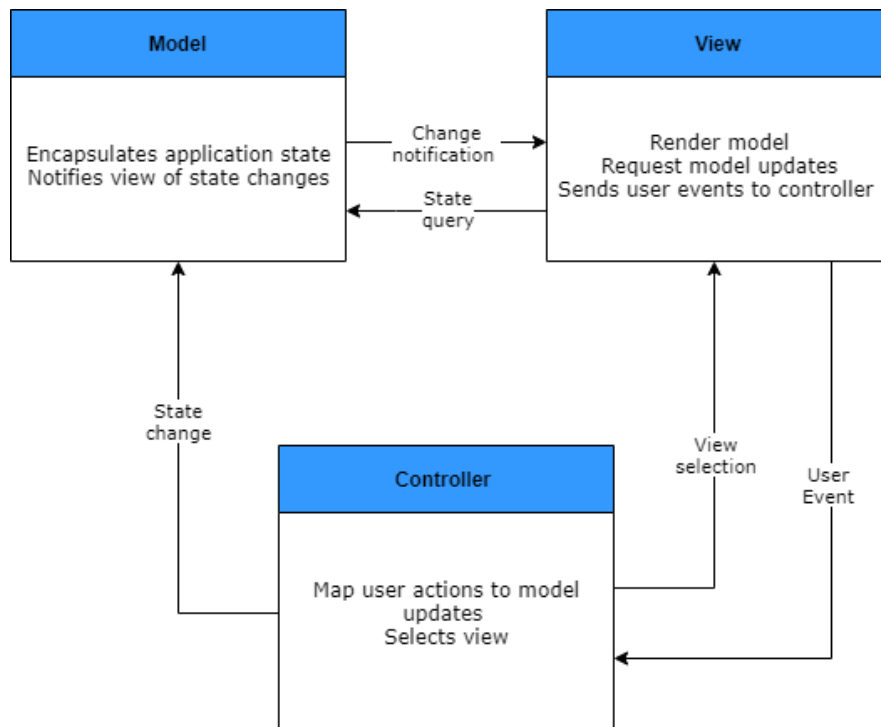
We have in our non-Relation database three collections:

- Patient info externals: is representing as last 4 and least external measurements, and it have 5 documents each document contains: document ID, patient ID (inside DB), sugar, pressure1 (diastolic), pressure2 (systolic), temp and date.
- Patient infos: is representing as internal measurements, and it have 2 documents (2 patients) each document contains: document ID, patient ID (inside DB), ecg, o2 and pumb.
- users: is representing as all users in the application, and it have 3 documents (1 doctor and 2 patients) each document contains: user ID (inside DB), role, username and password.

#### 4.3.5 Heroku

Is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. This platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market [12]. We used it for receiving a request form phone application and response action as we need.

Heroku use MVC design pattern and separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., enter presses keypad, touch, etc.)



**Figure 23: Model-View-Controller (MVC) pattern**

Advantages of MVC pattern:

- Allows the data to change independently of its representation and vice versa.
- Supports presentation of the same data in different ways with changes made in one representation shown in all of them.

#### 4.3.5 Postman

Is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster [13]. we used it for control the server that we build, like add or delete patient/doctor.

In general, this API is giving you ability to send requests in Postman to connect to APIs you are working with. Your requests can retrieve, add, delete, and update data. Whether you are building or testing your own API, or integrating with a third-party API,

you can try out your requests in Postman. Your requests can send parameters, authorization details, and anybody data you require.

In the main we have 2 folders:

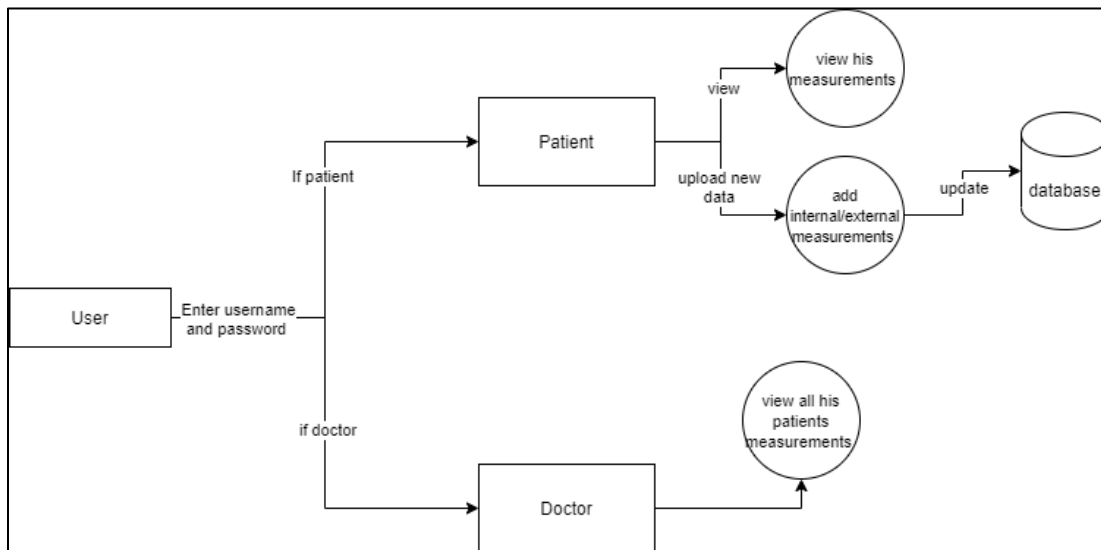
- User: have 1 get and 2 post functions, which get is representing I send request to server and get data and post is representing I send data to the server, functions details:
  1. Get all user: is representing as get all users information in our application (doctors/patients), and present as get function.
  2. Reg user: is representing as I create a new user with his details (username, password and role), and present as post function.
  3. User login: is representing as I send doctor username to take his token that will be used in Patient Info, and present as post function.
  
- Patient Info: have 2 get and 2 post functions, functions details:
  1. Create data: is representing as create data for internal measurements from nodemcu 8266 and these data are heart rate, spo2 and ecg values, and present as post function.
  2. Get all patient: is representing as that get all patient's information in this token. after we take doctor token, to use it to send his patient data to him, in other words we will take this token and place it in specific place in his patient API, to let his doctor to see his measurements, and present as get function.
  3. Create external data: is representing as create data for external measurements from patient and these data are temperature, blood pressure and blood sugar level values, and present as post function.
  4. Get data: is representing as that get all patient information like internal/external measurements from his ID inside database, and present as get function.

## 4.4 Users

The application has two user roles:

- Patient: he can view his status and enter his external measurements and his role present as 1.
- Doctor: he can view his patient's status and his role present as 2.

In figure below we add dataflow diagram for user roles and the data is updated on the database:



**Figure 24: user dataflow diagram**

# Chapter 5

## Validation and Discussion

### 5.1 Overview

In this chapter, we will describe and discuss the steps, results of the implementation, and testing of the hardware and software parts.

### 5.2 Units testing

We start by testing all the parts to ensure that all of the functions work perfectly and without errors.

#### 5.2.1 Hardware Testing

After completing the phase of connect of nodemcu pins with MAX30100 and AD8232, and make sure that microcontroller is working after we test that it can connect to Wi-Fi network, also be sure that all the components connected correctly with ESP nodemcu8266 pins and after we upload the arduino code to nodemcu, the values (spo2, heart-rate and ECG) should be correct as follows:

- Max30100: This sensor is working correctly and gives 2 reading for spo2 and heart-rate, as the thresholds and test cases below:
  1. Spo2

Condition	SpO2 Range
Normal	95% to 100%
Brain Gets Affected	80% to 85%
Cyanosis	Below 67%

Figure 25: Spo2 value for human

Test	Value	Status
1	97%	normal
2	83%	abnormal
3	99%	normal

**Table 1: Test case spo2**

2. Heart-rate

Condition	Heart rate range
<b>Normal</b>	<b>60 to 100</b>
<b>Abnormal</b>	<b>&gt; 100 or &lt;60</b>

**Figure 26: Heart-rate value for human**

Test	Value	Status
1	120	abnormal
2	117	abnormal
3	73	normal

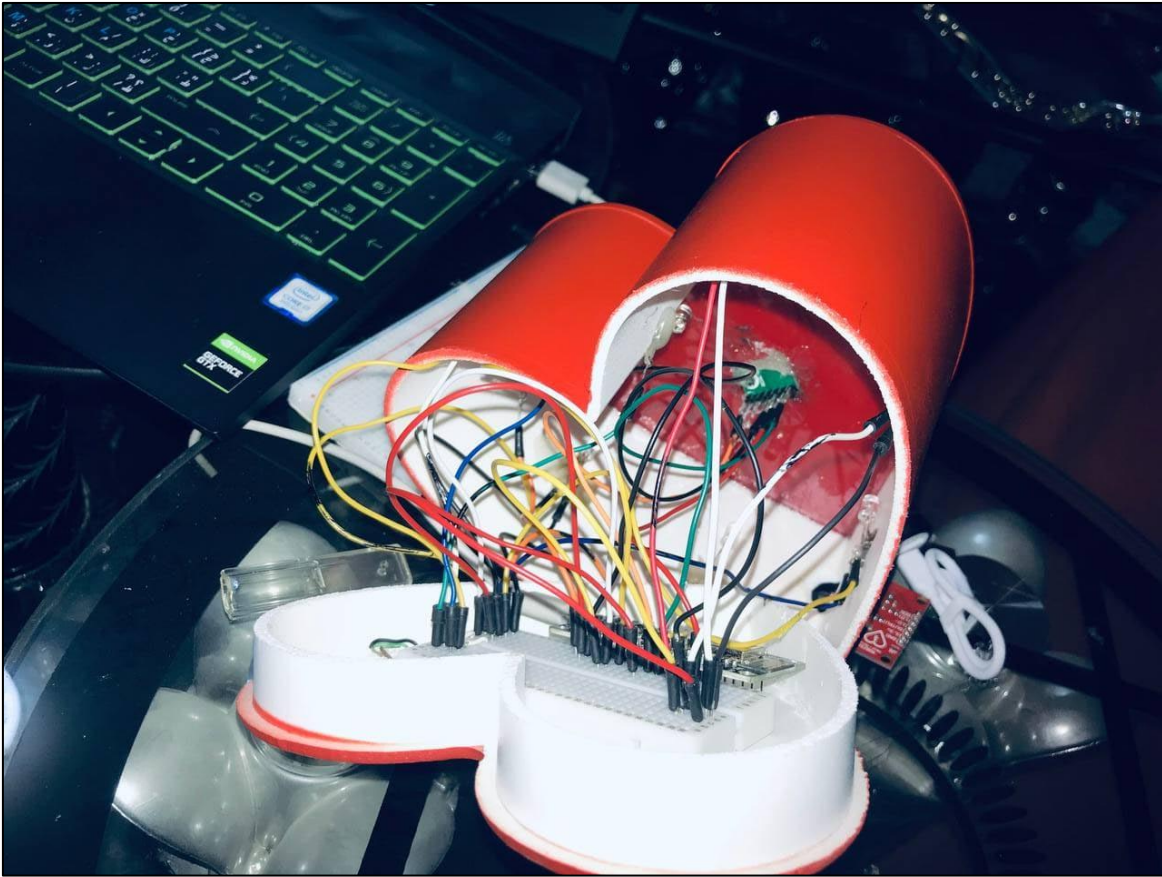
**Table 2: Test case heart-rate**

- ECG: This sensor is requiring some specific conditions to working correctly, so the area on which the electrodes were placed should be clean from hair and the patient should not move, otherwise lack of these conditions leads the signal be noised and meaningless reading. This diagram is shown when the ECG is connected. Otherwise, if not connected this diagram won't show.

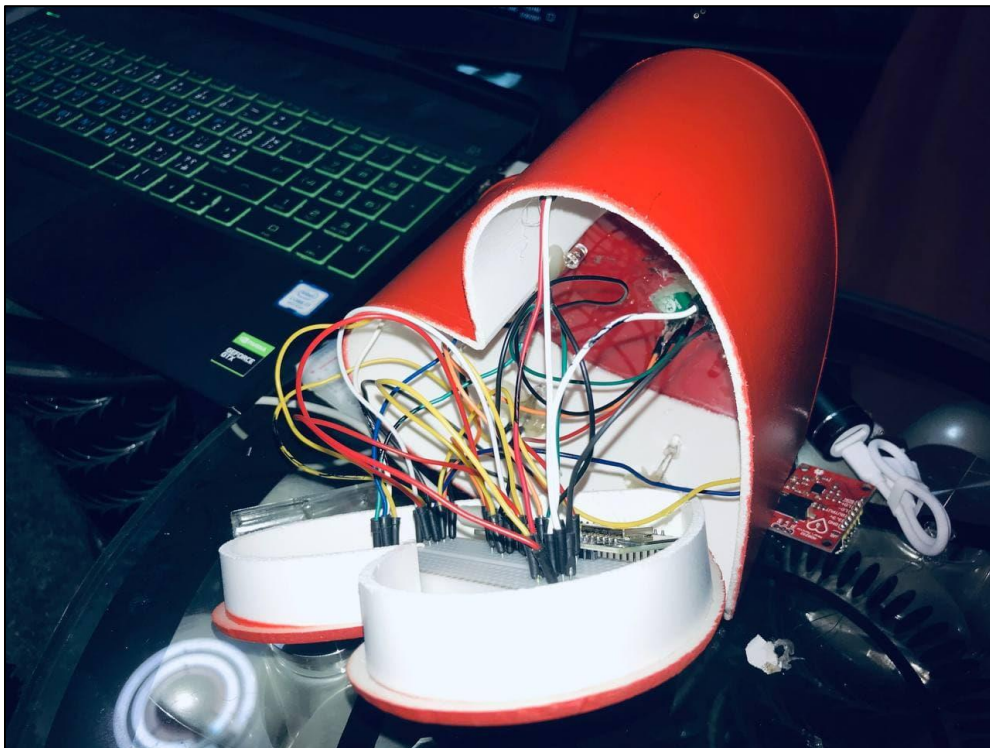


**Figure 27: ECG diagram if it connected**

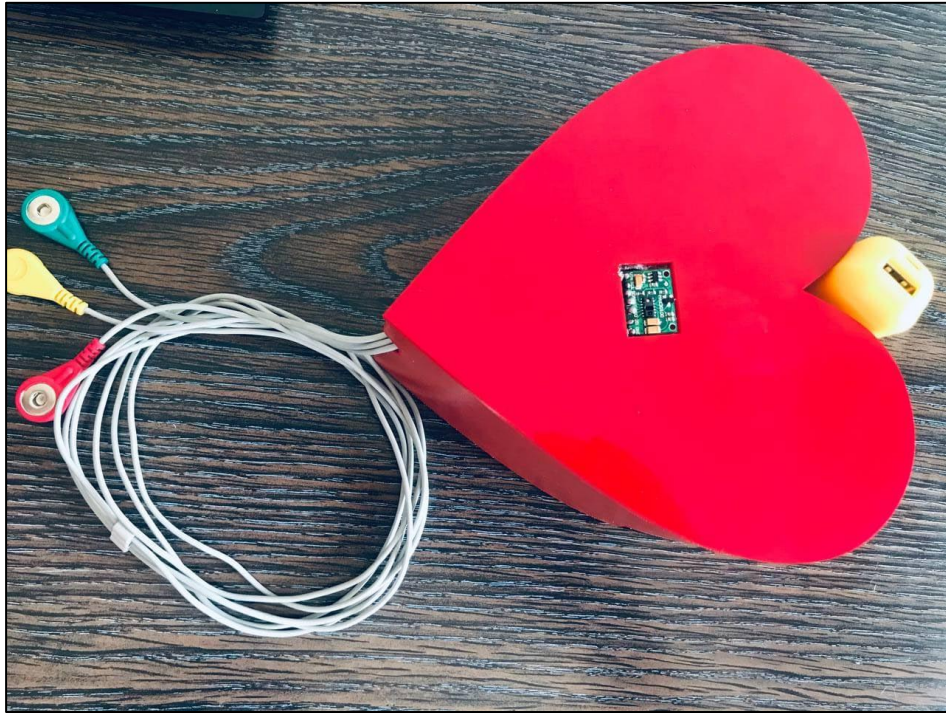




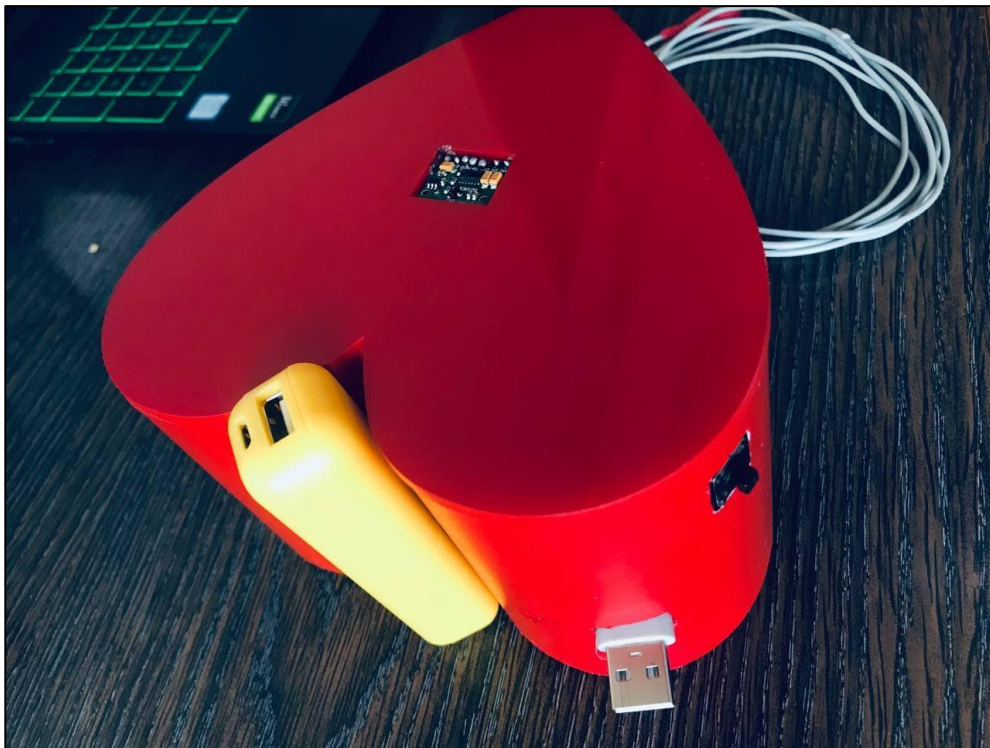
**Figure 28: All circuits in breadboards 1**



**Figure 29: All circuits in breadboards 2**



**Figure 32: Final result 1**



**Figure 30: Final result 2**

### 5.2.2 Software Testing

In this section, we will describe the ways to test the software that we used to implement our system.

The project needs from the user to login with his username and password, then if username detected as patient, the application shows him the patient page, if detected as doctor show him the doctor page.

If the patient enters external measurements to the application, these values transmit to the server and stored in database. If the patient or doctor want to view the latest and the 4 previous measurements, the application will show it to the user with their exact date and time, and if it normal or not for latest measurements.

### 5.2.3 How does it works

When the nodemcu is powered up, it connects to the Network via Wi-Fi, and the system does nothing until the patient pushes the pushbutton, at which point the system activates, the LEDs light up, and the sensors begin taking 15 readings, the last reading is sent to the server and stored in the databases. The device then comes to a halt until the patient presses the button to start taking new readings.

## 5.3 Implementation issues and challenges

We faced several problems since the beginning of the project, one of these is the hard of connecting the hardware and software together and combine them in one system. And the electronic component is defected during hardware testing, like what happened with us while we testing MAX30100 two times. There are other problems related to poor internet, when the WiFi is weak signal, the connections between microcontroller and server be not responsive or not real time response.

In addition to the challenges that we faced the insufficient time and the circumstances that made it difficult for us to meet in a face-to-face because of Corona

and the difficulty of arranging suitable dates for everyone, but we did the project as required and obtained excellent results.

On the other hand, the electronic parts (hardware) problems were not available in abundance and needed time to be treated and delivered. The programming is another challenge, such as learning programming languages, application environments, and creating web pages using open-source front end development framework.

# Chapter 6

## Conclusion

### 6.1 Overview

In this chapter, we will conclude the informative summary and future work of our project.

### 6.2 Summary

The main objective of the experiment was successfully achieved. All the individual modules like Max30100 module, ECG module and remote viewing module gave out the intended results.

The project Remote follow up healthcare system is a telemedicine application which allows the doctor to view the patient's vital signs and parameters remotely time. We propose a mobile phone based on patient monitoring system with good extensibility and transmitted remotely in a real time sense.

The presented monitoring system will be a cost effective, flexible and robust solution supporting a unique mobile based computational platform.

### 6.2 Future work

The future scope for this project is that the notification will be send only to the doctor whom the patient consults specially who have the abnormal status. That is, the alert send can be made specific, along with the patient's name and ID.

Other medical sensors will be placed in the heart box, such as measuring temperature, blood pressure and sugar blood level.

## References

[1] NodeMCU ESP8266 at: NodeMCU ESP8266 at: <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/> [Accessed:2021/4/23]

[2] MAX30100 at:

<https://www.maximintegrated.com/en/products/sensors/MAX30100.html>

[Accessed:2021/4/23]

[3] MAX30100 at: <https://www.electrodragon.com/product/max30102-pulse-oximeter-heart-rate-sensor-module/> [Accessed:2020/11/15]

[4] AD8232 ECG Sensor at: <https://how2electronics.com/iot-ecg-monitoring-ad8232-sensor-esp32/> [Accessed:2020/11/15]

[5] BH1790GLC at: <https://www.digikey.ie/product-detail/en/rohm-semiconductor/BH1790GLC-EVK-001/BH1790GLC-EVK-001-ND/6677095>

[Accessed:2021/1/2]

[6] AD8232 ECG at: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf> [Accessed:2021/1/4]

[7] Arduino IDE at: <https://www.arduino.cc/en/software> [Accessed:2021/4/3]

[8] Visual Studio Code at: <https://code.visualstudio.com/docs> [Accessed:2021/4/3]

[9] MongoDB at: <https://www.mongodb.com/what-is-mongodb> [Accessed:2021/4/18]

[10] Heroku at:

<https://www.heroku.com/about#:~:text=Heroku%20is%20a%20container%2Dbased,getting%20their%20apps%20to%20market> [Accessed:2021/4/18]

[11] Postman at: <https://www.postman.com> [Accessed:2021/5/08]

[12] Heart-rate value for human at: <https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings> [Accessed:2021/5/28]