PARALYSED PATIENT MONITORING KIT (PARAKIT)

AHMAD SYAHMI ADLI BIN SHAMSUDDIN

Bachelor of Computer Science (Computer Systems & Networking) with Honours

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : AHMAD SYAHMI ADLI BIN SHAMSUDDIN

Date of Birth :

Title : PARALYSED PATIENT MONITORING KIT

Academic Session : SEMESTER I (2022/2023)

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997) *
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done) *
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang.
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

AHMAD SYAHMI ADLI BIN SHAMSUDDIN

Date: 30 January 2023

(Supervisor's Signature)

TS. DR. MOHD IZHAM MOHD JAYA

Date: 30 January 2023

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name	AHMAD SYAHMI ADLI BIN SHAMSUDDIN
Thesis Title	PARALYSED PATIENT MONITORING KIT (PARAKIT)
Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

TS. DR. MOHD IZHAM BIN MOHD JAYA
HEAD OF PROGRAM (SOFTWARE ENGINEERING)
FACULTY OF COMPUTING
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG DARUL MAKMUR
TEL : 09-431 5580

____
(Supervisor's Signature)

Date: 30 JANUARY 2023

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis, and, in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Computer Systems & Networking).

TS. DR. MOHD IZHAM BIN MOHD JAYA
HEAD OF PROGRAM (SOFTWARE ENGINEERING)
FACULTY OF COMPUTING
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG DARUL MAKMUR
TEL : 09-431 5580

(Supervisor's Signature)

Full Name : TS. DR. MOHD IZHAM BIN MOHD JAYA

Position : PROJECT SUPERVISOR

Date : 30 JANUARY 2023

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : AHMAD SYAHMI ADLI BIN SHAMSUDDIN
ID Number : CA19146
Date : 30 JANUARY 2023

PARALYSED PATIENT MONITORING KIT (PARAKIT)

AHMAD SYAHMI ADLI BIN SHAMSUDDIN

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Computer Systems & Networking)

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2023

ACKNOWLEDGEMENTS

First and foremost, I want to express my gratitude to the Almighty Allah, Subhanahu Wa Ta'ala, for giving me the foresight and courage to complete this project. I want to express my gratitude to my project supervisor, Ts. Dr. Mohd Izham Bin Mohd Jaya, as well as my academic supervisor, Dr. Zahian Binti Ismail, for their ongoing support of my academic endeavours.

In addition, I would like to thank my family for always being supportive of me regardless of whatever challenges I may face. Without them all, it's possible that the project wouldn't have been finished on schedule. I owe a debt of gratitude to every professor and lecturer at the Faculty of Computing in Universiti Malaysia Pahang (UMP) who has contributed to the project in any way, either directly or indirectly.

Finally, I'd like to express my gratitude to my acquaintances and classmates who have supported me through this whole endeavour whenever I've encountered difficulties.

ABSTRAK

Projek ini memaparkan hasil projek tahun akhir yang bertujuan untuk membantu pesakit lumpuh yang tinggal di rumah dalam memantau kesihatan mereka dengan mudah dan tepat. Kit ini dikenali sebagai "Paralyse Patient Monitoring Kit (PARAKIT)" dan menggunakan tiga buah sensor, iaitu Pulse Rate Sensor, Non-Contact Infrared Sensor, dan MQ135 Air Quality Sensor. Kaedah Fuzzy Logic digunakan untuk memproses data dari ketiga-tiga sensor tersebut dan menentukan keadaan semasa pesakit. Hasil kajian menunjukkan bahawa PARAKIT mampu memproses data dengan tepat dan memberikan analisis yang akurat mengenai keadaan kesihatan pesakit. Kit ini mudah dan praktikal digunakan kerana doktor atau jururawat yang terlatih hanya perlu dibawa ke rumah pesakit dan tidak memerlukan pemasangan yang rumit. Ini membuat kit ini sangat sesuai untuk pesakit yang lumpuh satu badan yang tinggal di rumah dan tidak berupaya datang ke hospital dan klinik kesihatan untuk melakukan pemeriksaan kesihatan secara rutin. Secara keseluruhan, PARAKIT membuka peluang bagi pembangunan sistem memantauan pesakit yang lebih canggih dan berkesan bagi memantau kesihatan pesakit yang lumpuh satu badan. Projek ini merupakan langkah penting dalam memastikan bahawa pesakit yang lumpuh satu badan menerima perawatan yang berkualiti dan membantu mereka untuk membuat tindakan pencegahan yang berguna untuk kesihatan mereka.

ABSTRACT

This report presents the results of a final year project aimed at helping paralyzed patients who live at home to monitor their health easily and accurately. The kit is known as the "Paralyzed Patient Monitoring Kit (PARAKIT)" and uses three sensors, namely the Pulse Rate Sensor, the Non-Contact Infrared Sensor, and the MQ135 Air Quality Sensor. Fuzzy Logic method is used to process data from the three sensors and determine the patient's current condition. The results show that PARAKIT is capable of processing data accurately and providing a precise analysis of the patient's health status. The kit is easy and practical to use because trained doctors or nurses only need to bring it to the patient's home and does not require complicated installations. This makes the kit very suitable for paralyzed patients who live at home and are unable to come to hospitals and health clinics for routine health checks. Overall, PARAKIT opens up opportunities for the development of more advanced and effective patient monitoring systems to monitor the health of paralyzed patients. This project is a crucial step in ensuring that paralyzed patients receive quality care and helping them to make useful preventive actions for their health

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
PHP	Hypertext Preprocessor
UNO	Arduino’s version
IoT	Internet of Things
PARAKIT	Paralysed Patient Monitroing Kit
ECG	Electrocardiogram
IV Fluids	Intravenous Fluids

CHAPTER 1

INTRODUCTION

1.1 Background Study

A person who is fully paralyzed is unable to move any part of their body due to a complete loss of function in the nerves, muscles, or spinal cord. This type of paralysis can be caused by a variety of conditions, including spinal cord injuries, stroke, and neurological disorders. Individuals who are fully paralyzed are usually unable to move independently and may require assistance with all aspects of daily life, including dressing, bathing, and eating. Despite these challenges, many fully paralyzed patients can lead fulfilling and meaningful lives with the help of supportive friends, family, and medical doctors or healthcare professional.

The main objective of this project is to develop a Paralyzed Patient Monitoring Kit (PARAKIT) where this kit will help the healthcare professional to monitor patients remotely from clinic or hospital. The monitoring kit will have three different sensors which are air quality sensor, temperature sensor, and pulse rate sensor. The monitoring kit also will be included with dashboard monitoring that designed to continuously monitor the vital signs and environmental conditions of a paralyzed patient in their home and provide real-time alerts to healthcare professionals if any abnormalities or potential health risks are detected.

By combining the data from these three sensors, the monitoring system will be able to provide a comprehensive overview of the health and well-being of the paralyzed patient and allow healthcare professionals to take timely action if necessary. The use of IoT technology will enable the system to be remotely accessed and controlled, making it possible for healthcare professional to monitor the patient from any location.

1.2 Problem Statement

Each paralyzed patient has their own disease apart from not being able to move. Diseases that may happen to them are fever, asthma, heart disease and high blood pressure. The problem that might arise is when the health professional needs to go to the patient's house to do basic monitoring. Some of Clinic have an extra charge to go to patient house.

Furthermore, the paralyzed patient will have difficulty moving from one part to another part where it will take a lot of time to complete the movement. Most of them also will feel uncomfortable as their body cannot move strongly and it is possible that a lot of movement will cause them to feel pain in their muscles and cause shortness of breath.

1.3 Objectives

There are several objectives that this Paralysed Patient Monitoring System propose which as follows:

1. To study the new method for monitoring paralysed patient at their home.

2. to develop an IoT based system that capable to monitor and determine paralysed patient's condition using fuzzy logic.

3. to evaluate the effectiveness of paralysed patient monitoring system in monitoring paralysed patient's condition.

1.4 Scope of Project

The scope of the development "Paralyzed Patient Monitoring Kit "are as follows:

1. Three sensors will be used in this project which are pulse rate sensor, non-contact infrared temperature sensor and MQ135 air quality sensor.
2. The monitoring kit is suitable to be used for paralyzed patient that completely unable to move and medical doctor or healthcare professional that responsible to monitor paralyzed patient condition.
3. The kit will be able to do daily important monitoring such as body temperature, heart rate and home air quality monitoring.

1.5 Report Organization

This chapter contain of five subchapters. The first subchapter is an introduction to introduce the Paralysed Patient Monitoring Kit (PARAKIT) in general and elaborates why the system should implement and how the system will help both of paralyzed patient and healthcare professional. The second subchapter is problem statement or motivation of the project which flesh out the problem faced by paralyzed patient to go to hospital to monitor their basic medical check-up while the third subchapter are the objectives which is the target to improve the problem faced by both user, paralysed patient and healthcare professional. The last subchapter is the scope of the project that require to determine who are the user that will be use the system, sensor used to collect data, and database use for the system.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The comparison of the three existing system is compulsory to ensure the proposed system can do better. The three existing system are Patient Monitoring System for IoT-Based Healthcare System Using Deep Learning, Patient Health Monitoring using Arduino through IoT and An Intelligent IoT Based Healthcare System Using Fuzzy Neural Networks and the comparison includes their Monitoring sensor used, Communication Protocols, functionality, medium, internet connection and function. The five indicators mentioned are very useful in the evaluation to ease define the strength and weakness of the system and simultaneously reduce the weakness of the proposed system.

2.2 Existing Systems

2.2.1 PATIENT MONITORING SYSTEM FOR IOT-BASED HEALTHCARE SYSTEM USING DEEP LEARNING

The author of this project are Abhishek Lakshman Singh, Ashok Yadav and Prof Satish Ranbhise where they focusing to develop a wireless health monitoring and disease prediction using data mining that can monitor patient for 24 hours [1]. There are three sensors had been used for this project which are ECG, Pulse Rate and body temperature sensor. All the sensor used is possible to measure all required value for wireless basic monitoring. Figure 1 below show the block diagram of this proposed project for better understanding on how the system work and what components used to the project successful.

The figure 2.1 above show all the important component that used for this project where the first things that need to focus on is type of sensor used. There are focus on is type of sensor used. There are 4 sensor use shown in the block diagram which are Temperature sensor or LM35, Pulse sensor, Electrocardiography sensor to measure patient heartbeat and Humidity sensor for in house air quality measurement. All the sensors are connected to Arduino through any of the six analogue pins available. The ESP 8266, a Wi-Fi module is used to collect sensor data from Arduino serial monitor and deliver the data to IoT platform through internet by using GSM module communication protocol. The IoT platform will visualize the collected data in graph form including sensor historical and current data. The alert function also available and able to deliver the alert message and the author do not mention what type of device or system used to deliver the alert message. Finally, the system can deliver disease prediction by using intelligent data mining to detect the illness that might face to the patient.

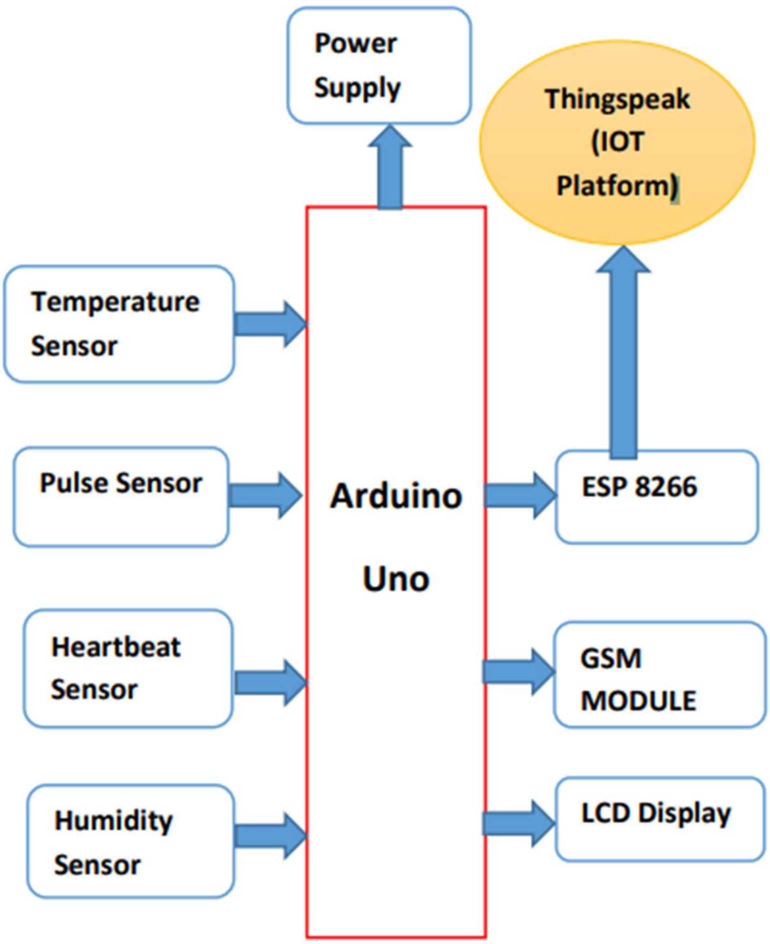


Figure 2.1 Patient Health Monitoring and Disease Prediction Block Diagram

2.2.2 AN INTELLIGENT IOT BASED HEALTHCARE SYSTEM USING FUZZY NEURAL NETWORKS

The research paper or project is focusing to change the current implementation where doctor and nurse need to monitor patient in the given schedule [2]. Therefore, the researchers of this project want to improve the current implementation into 24/7 remotely monitoring method. There are 4 sensors used in this project which are pulse rate sensor, blood pressure sensor and temperature sensor. Figure 2.2 below is block diagram which will explain in detail on how the this project work on each block diagram.

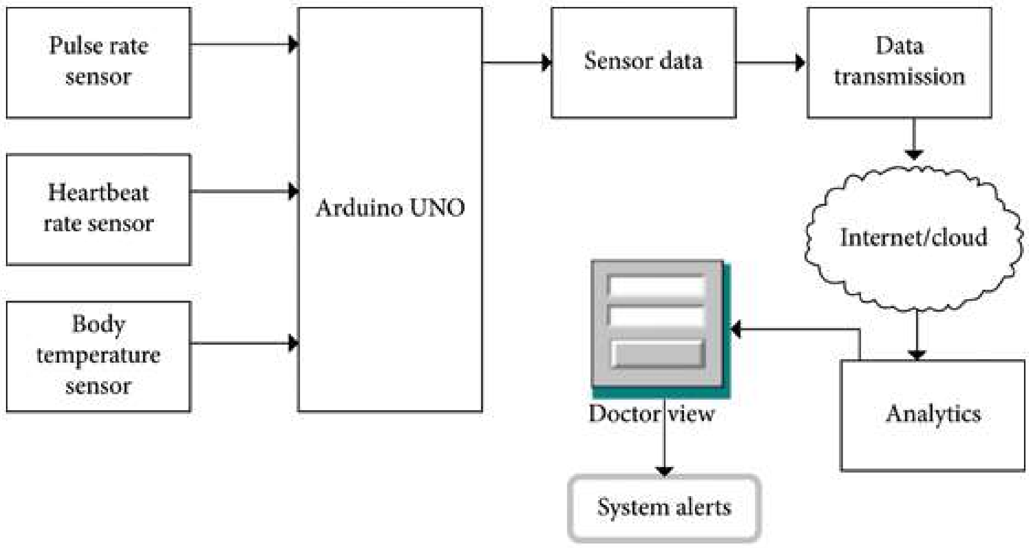


Figure 2.2 Smart Healthcare Block Diagram

The physical view above shows the important component of the proposed system where the most important part is three sensor used in this project directly connected to Arduino Uno to combine the collected data from those sensors. The analytics block diagram is used to construct artificial intelligent technique where in this project the author decided to use Fuzzy Logic method to provide decision making. The last block diagram is a doctor view where it used to show the data visualisation from the sensor data.

2.2.3 PATIENT HEALTH MONITORING USING ARDUINO THROUGH IOT

In this proposed project, the author is focusing to provide a solution to manage healthcare especially for elderly people for their medical assistance[3]. This is where they introduce this kind of telemedicine where utilise the IoT monitoring. The second factor for author in building this ecosystem is basically to develop an affordable solution and can be used for any type of economical family or individual. This project only required one microcontroller to collect data from the sensor and visualize on the website. In this project, there are three sensors used to measure patient parameter which are blood pressure, temperature and heart rate sensor. In figure 2.3 below show the system block diagram for better understanding on required website.

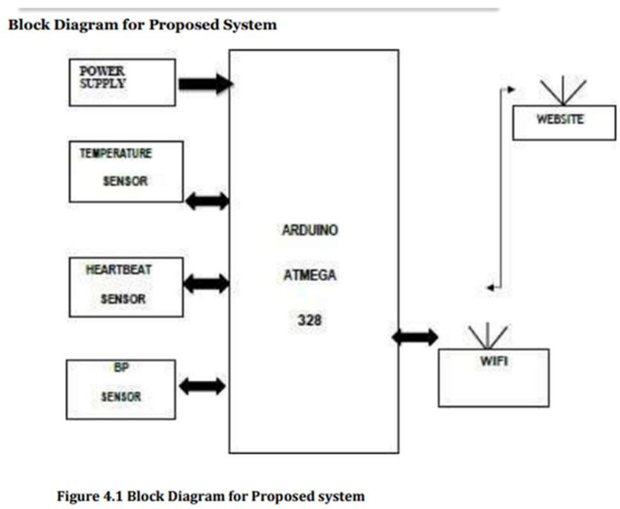


Figure 2.3 Smart Healthcare Block Diagram

The figure above shows all required components needed and how to use the project structure from collecting data and data visualisation. The three sensors use in the system is directly connected to the Arduino which act as microcontroller for data collection. Then, the Arduino have will transfer all three collected data Wi-Fi router before going to directly visualize at the system website. For this project, there is no artificial intelligent method used either Fuzzy Logic or prediction method.

2.3 Analysis/Comparison of Existing Systems

Table 2.3.1 Comparison of three existing system

	An Intelligent IoT Based Healthcare System Using Fuzzy Neural Networks	**Patient Health Monitoring using Arduino through IoT**	**Patient Health Monitoring using IoT and Disease Prediction using Data Mining**
Monitoring Sensor Used	• Temperature Sensor • HeartBeat Sensor • BP Sensor • HC05 Bluetooth module	• Heart Rate Sensor • Pulse Rate Sensor • Body Temperature Sensor	• LM35 Temperature Sensor • Pulse Sensor • Heartbeat Sesnsor • Humidity Sensor
Communication Protocol	WI-FI, Bluetooth	WI-FI	TCP/IP Protocol
Processing Device	Arduino UNO	Arduino ATMEGA	Arduino UNO
Alert	Alert implement in the system	No alert features implemented	Alert implement in the system
Server / Database Use	Cloud Database	Cloud Database	SQLite local Database
Artificial Intelligent Method	Fuzzy Neural network	No Artificial Intelligence use in this project	Prediction using Data Mining

Table 2.3.2 Advantage and Disadvantage of three existing system

	ADVANTAGE	DISADVANTAGE
An Intelligent IoT Based Healthcare System Using Fuzzy Neural Networks	• Implementation of Artificial Intelligent method using Fuzzy Neural • Low-Cost project implementation • Cloud Database implementation • Able to monitor patient for 24 hours	• Need to add analogue Bluetooth module • No Display or output show in the project report • Using LM35 temperature sensor that is not suitable to read body temperature
Patient Health Monitoring using Arduino through IoT	• Low-Cost project implementation • Cloud Database implementation • Able to monitor patient for 24 hours • Using Arduino Atmega with embedded WI-FI	• No Artificial Intelligent method implemented
Patient Monitoring System for IoT-Based Healthcare System Using Deep Learning	• Cloud Database implementation • Able to monitor patient for 24 hours • Implementation of Artificial Intelligent method using Deep Learning Method • Using Thing Speak to display historical sensor data	• High-Cost project implementation • Using LM35 temperature sensor that is not suitable to read body temperature • Using SQLite local database to store data

2.4 Summary

The three existing system chose are very competent. They have chosen of monitoring sensor, relevant communication protocol, processing device and type of database used. However, every system has their strong and weakness that we can identify.

The relevant comparison of those three existing projects is where most of the monitoring Sensor used are relevant and similar to the proposed project, Paralyzed Monitoring Kit (PARAKIT). Both of the three-review projects use an affordable processing device which are Arduino Uno, Arduino ATMEGA and ESP 8266. PARAKIT project will use Arduino UNO for analog connection and Raspberry Pi B+ in order to deliver sensor data to database. Furthermore, all those three projects use cloud databases to store data. The cloud database is useful to prevent any data loss if the physical or local server is corrupted or broken. The cloud database is also able to deliver the data to dashboard 24/7.

Finally, Artificial intelligent technique use is different on those three proposed projects where the first project uses Fuzzy Neural Network and third project use prediction using data mining while PARAKIT will use Fuzzy Logic method where doctor can view the output whether the patient is in a good condition or not. Fuzzy Neural Network method also has similar method with Fuzzy Logic, however the setup rule might be different because of the different sensor data. Overall, the three current systems are quite like PARAKIT monitoring systems and most of the features in those systems will be implemented in PARAKIT monitoring system.

CHAPTER 3

METHODOLOGY

3.1 Project Overview

Figure 3.1 shows the project overview for the Paralyzed Patient Monitoring Kit (PARAKIT). It starts with the microcontroller collecting all the data using the sensors and pre-defined data involved in the system. Then, the microcontroller sends all the data collected to the MQTT broker in a two-way communication. The broker then receives and transports the data to the local database to store all the data stated.

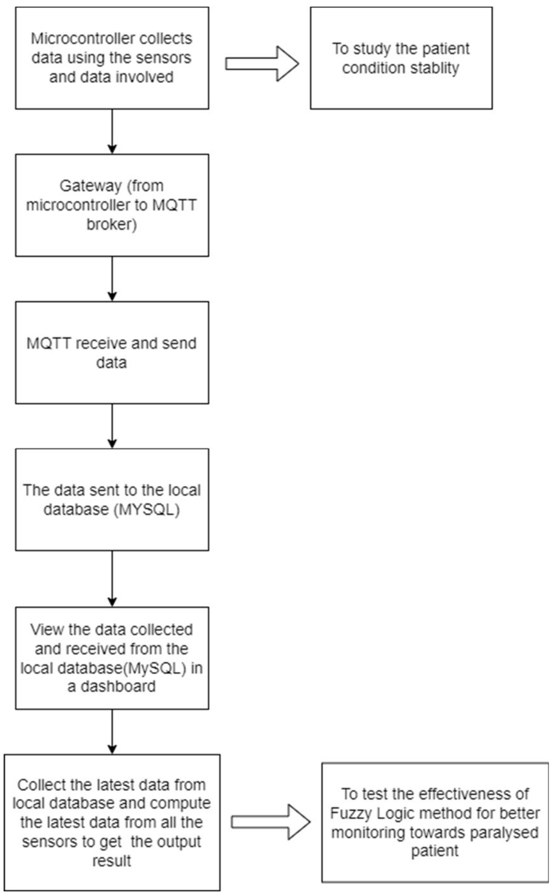


Figure 3.1: Proposed Project Architecture.

After that, the user able to view all the data collected in their appropriate interface such as graphs, tables, etc. inside a dashboard. For the project analysis section, AI technique is being presented and calculated its algorithm by computing fuzzy logic method in order to get the best current condition of paralyzed patient. Thus, certified nurse or Doctor can standby for any consequence that might face by the patient.

3.2 Project Flowchart

Figure 3.2 shows the first section of the project flowchart of Paralysed Patient Monitoring Kit to deliver basic remote monitoring for paralysed patient. The procedure begins with the collection of data of human heart rate through ECG sensor, the paralysed patient body temperature through non-touch infrared sesor and the patient surrounding air quality through MQ135 Air Quality sensor. All gathered data from those 3 sensor is analyzed and delivered to the Arduino UNO. The data measurements will then be compared and examined.

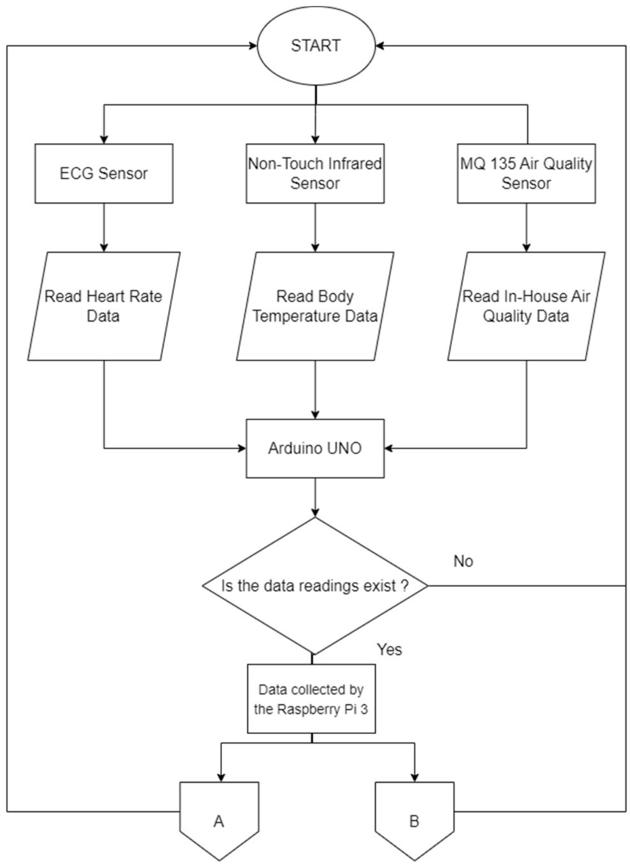


Figure 3.2: Project Flowchart (first section)

Figure 3.3 shows the second section of the project flowchart of Paralysed Patient Monitoring Kit where it started with the chosen Relay button. There are only two relay buttons present in the dashboard where the first button is “Relay Oxygen Button” and the other button is “Relay IV Infusion Button”. Certified nurses and doctors have the ability to control the button and the condition of the button will transmit to the raspberry pi via WebSocket. The rasbpberry pi will listen to WebSocket data and apply the request whether to turn “ON” or “OFF” the switch and this functionality is in the close loop implementation.

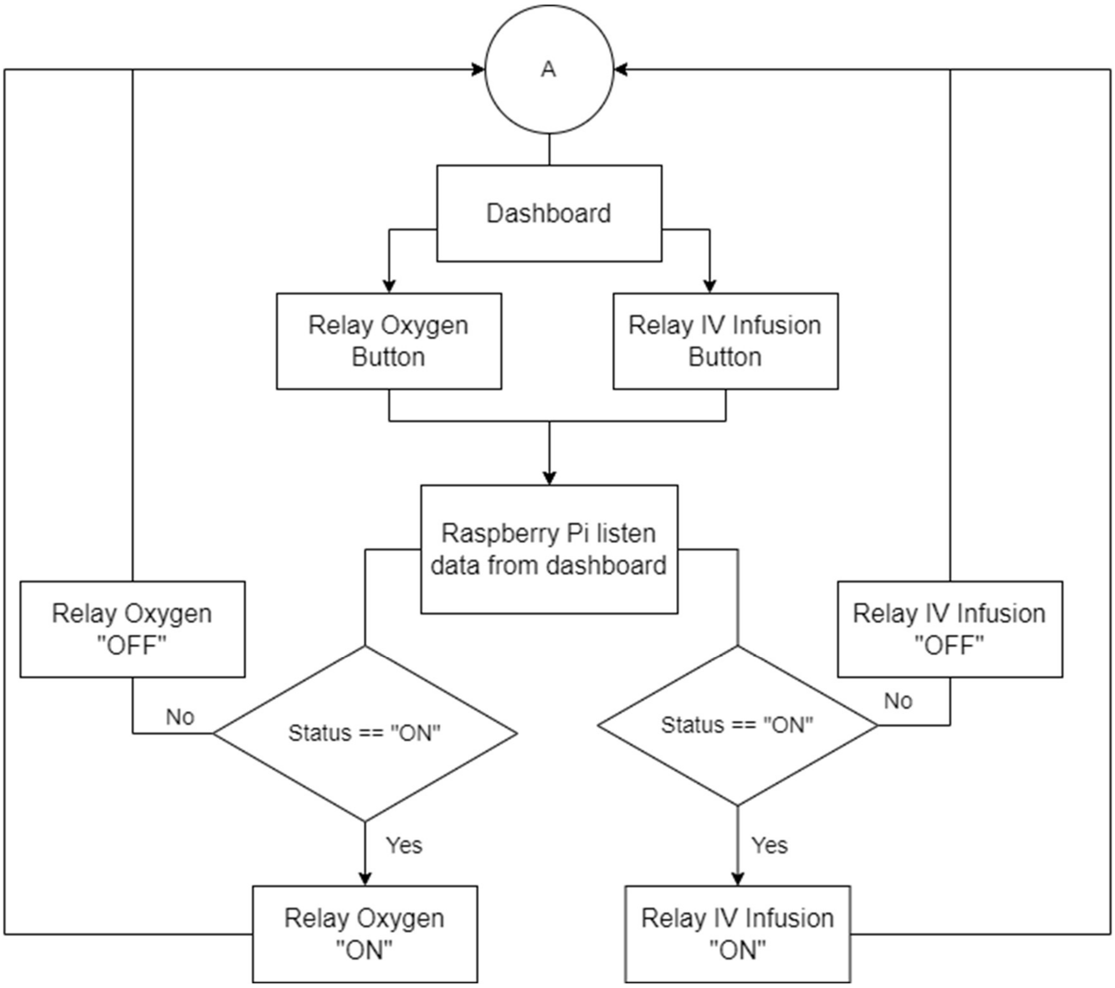


Figure 3.3: Project Flowchart (second section)

Figure 3.4 shows the third section of the project flowchart of Paralysed Patient Monitoring Kit. If heart rate, air quality and body temperature exceed the limit as stated in figure 8 below, the Telegram app will generate an alert message and notify the relevant authorities. In addition to that, the dashboard will serve its function of warning nearby people by triggering the buzzer module. Thus, the dashboard will display the data. Other than that, if the data readings are below the limit, the flow continues to display the data in the dashboard. The entire data will be sent to the local database.

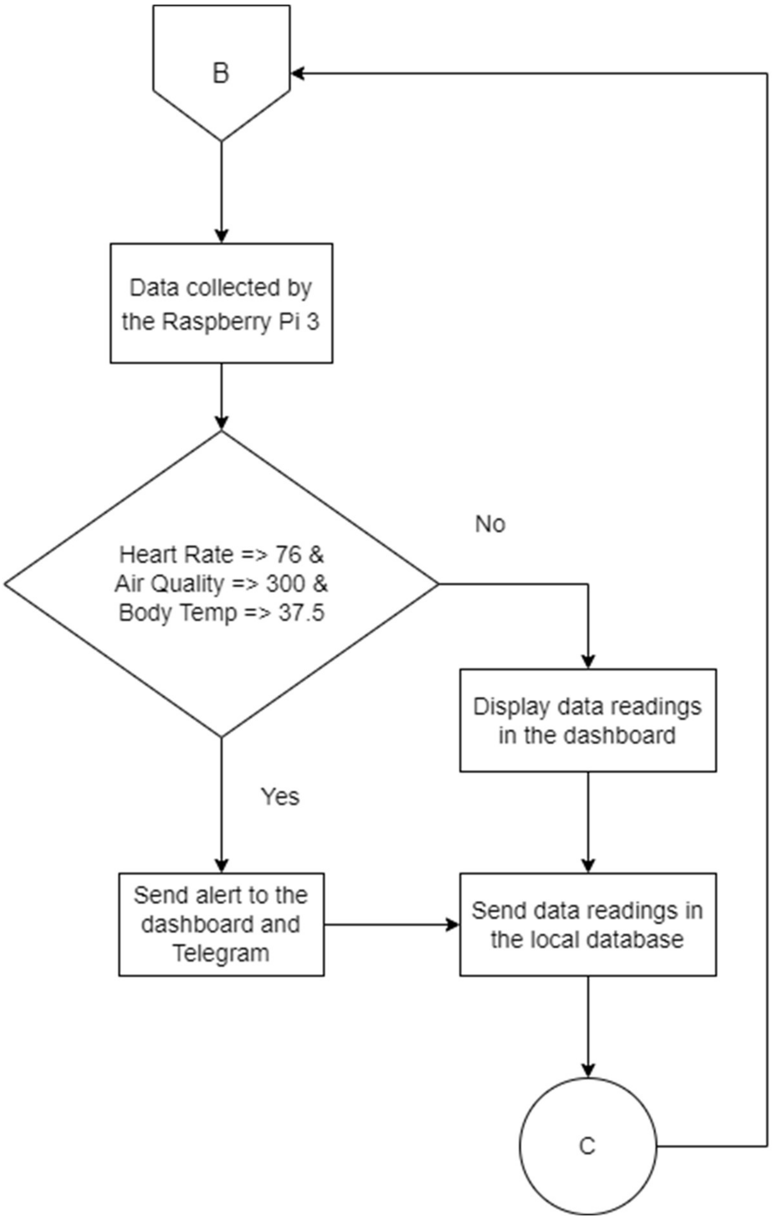


Figure 3.4: Project Flowchart (third section)

Figure 3.5 shows the fourth section of the project flowchart. The data collected from the database will be used to do fuzzy logic technique to detect the patient's condition based on the input of latest data from those three proposed sensors. If the readings of is fuzzy logic output are unstable, it will send the alert message of patient condition is currently unstable and store the data into the local database and to the Telegram to alert the certified nurse or doctor and continues the flow to display the fuzzy logic result readings in the dashboard. If the readings are below the limit, it will also continue to display the fuzzy logic readings result in the dashboard.

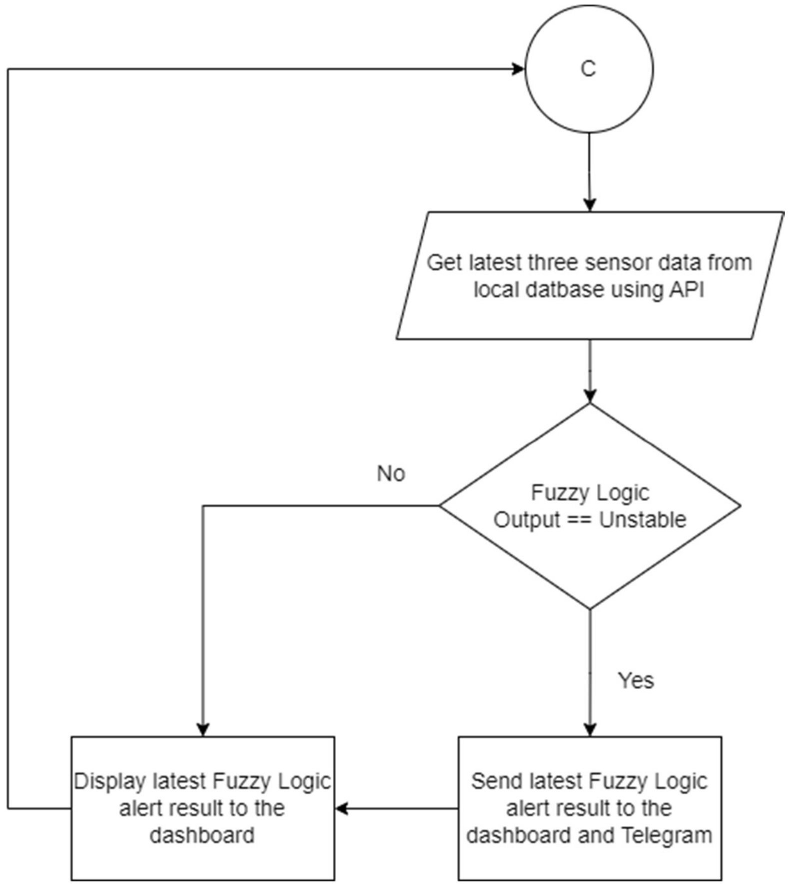


Figure 3.5: Project Flowchart (forth section)

Based on figure 3.3, figure 3.4 and figure 3.5 there are close loop implementation. There is no stop functionality in every section making the system will continue running to display the data for 24 hours. This is imporatnt to make sure the system can run and monitor continuesly without failure.

3.3 Project Design

3.3.1 IoT Architecture

The figure 3.6 below shows the proposed architecture where there are four layers present to indicate specific function of the IOT environment.

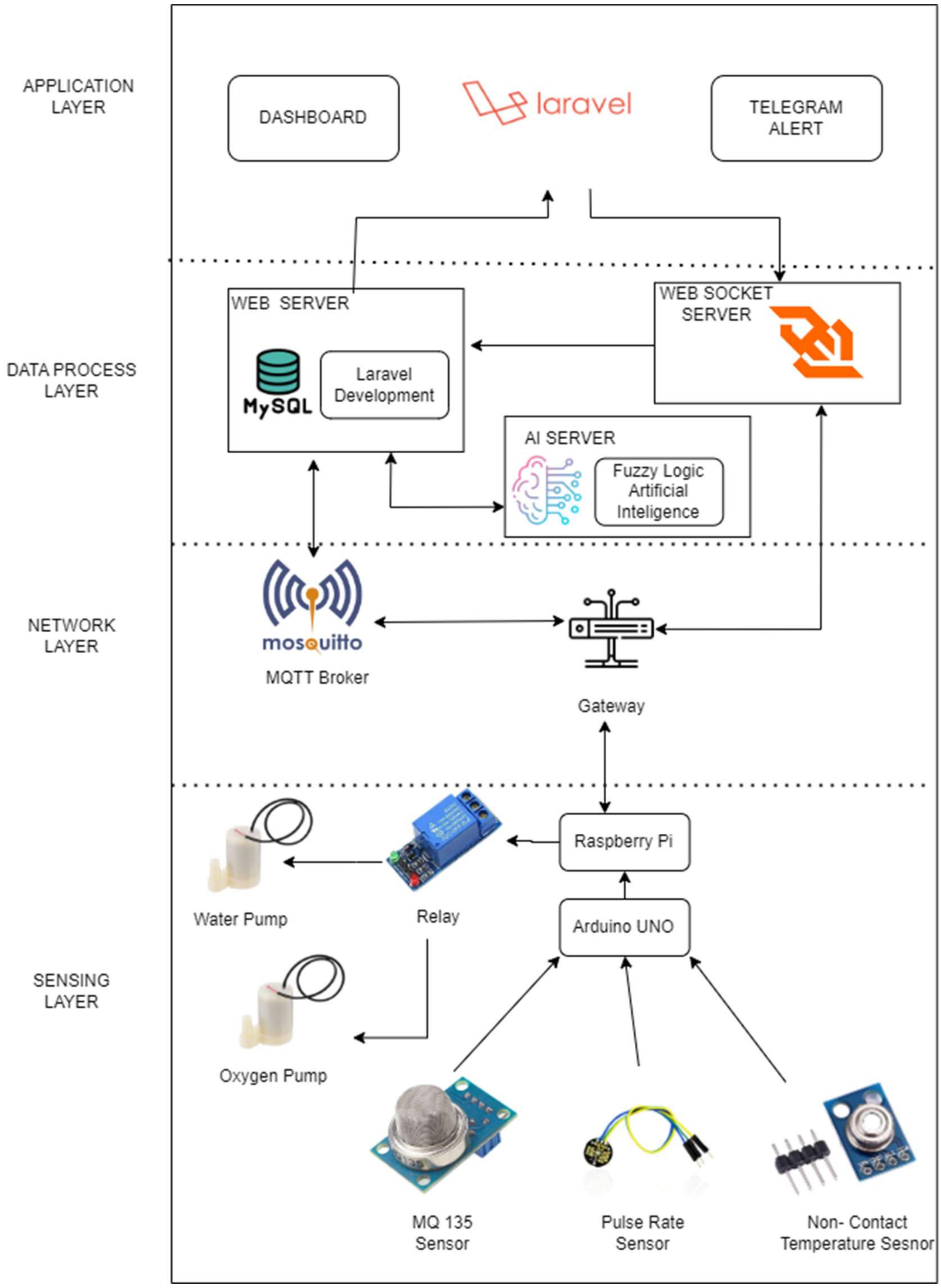


Figure 3.6: IoT Architecture for Paralysed Patient Monitoring Kit

The sensing layer as shown in figure 3.7 consists of three different types of sensors designed to measure basic patient monitoring parameters such as indoor air quality, heart rate, and body temperature of the paralyzed patient. The first sensor, MQ135 Gas Sensor, also known as the Air Quality Sensor, can detect ammonia, sulfur, and carbon dioxide that are commonly present in our environment. The second sensor is an pulse rate sensor, which measures the electrical activity of the heart by detecting signals from multiple electrodes placed on the chest, arms, and legs. The third sensor, a non-contact temperature sensor, measures body temperature without making physical contact with the patient. It works by using infrared technology to detect the amount of infrared radiation emitted by the person's skin. The MQ135 sensor, pulse rate sensor, and non-contact temperature sensor are connected to an Arduino Uno microcontroller and then transmit the sensor data to the Raspberry Pi via serial communication for further processing and analysis. A relay switch has been integrated into the system and is connected to the Raspberry Pi to provide the functionality of "On" and "Off" control of the nose oxygen tube and IV infusion through oxygen pump and water pump, ensuring patient receive automated first aid when needed. The connection of the relay switch to the Raspberry Pi allows the system to control the operation of the nose oxygen tube and IV infusion based on the patient's condition.

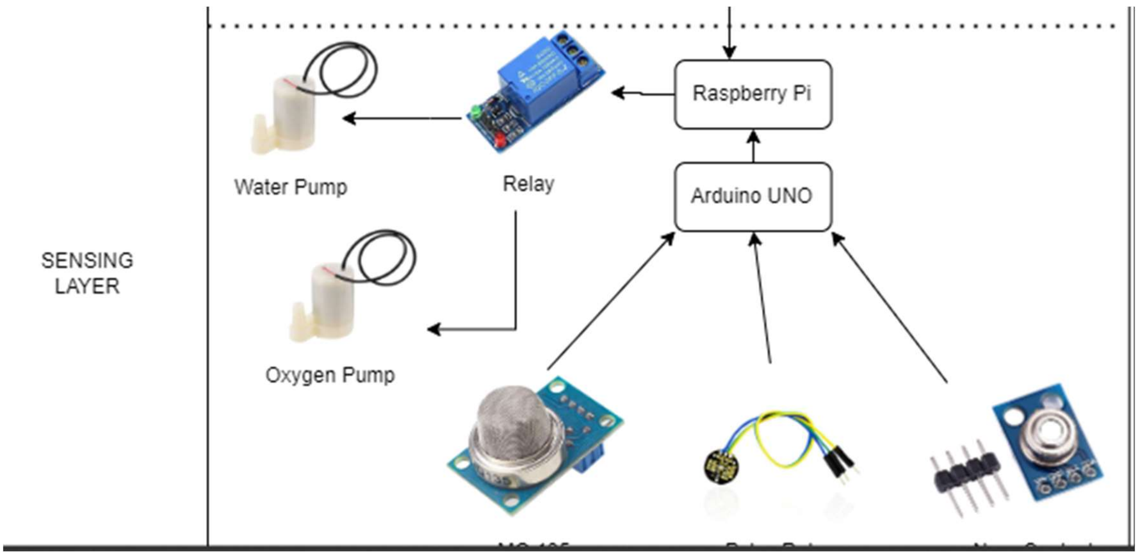


Figure 3.7: Sensing Layer of IoT Architecture

The network layer in figure 3.8 below is accountable for the transfer of information from the sensing layer to the data processing layer. The gateway is the primary element in this layer, serving as a connection between the Raspberry Pi and the MQTT broker. The Raspberry Pi gathers sensor readings from the Arduino Uno and transfers them to the gateway, which then sends the data to the MQTT broker. The MQTT broker functions as a mediator, facilitating the transmission of data to the data processing layer. The publish and subscribe pattern, where the Raspberry Pi is the publisher and the data processing layer is the subscriber, allows for seamless communication between the layers and real-time updates to the data processing layer. The MQTT broker ensures secure and efficient data transmission, crucial for the proper functioning of the system. The network layer also features a WebSocket server, which monitors changes made to the dashboard in the application layer. The WebSocket server communicates with the gateway, which then updates the dashboard, ensuring real-time updates for the user.

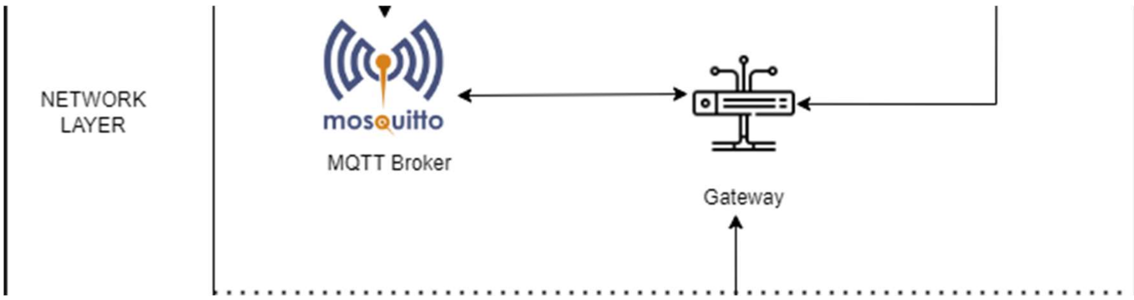


Figure 3.8: Data Process Layer of IoT Architecture

The data processing layer of the Paralysed Patient Monitoring Kit based on figure 3.9 is a vital component that manages all incoming data from the sensing layer and processes it for display on the application layer. The web server is responsible for storing the data in the database located on the server and receiving data from the MQTT broker to update the database. Additionally, the web server communicates with the AI server, which utilizes fuzzy logic methods, to analyze and make decisions based on the incoming data. Fuzzy logic is a type of mathematical logic that deals with uncertainty and imprecision. In the AI server, it is used to make decisions based on heart rate data, body temperature data and air quality data inputs by representing the information as a set of fuzzy sets. The output of the fuzzy logic method can then be used to make decisions based on the best possible conclusion derived from the input data.

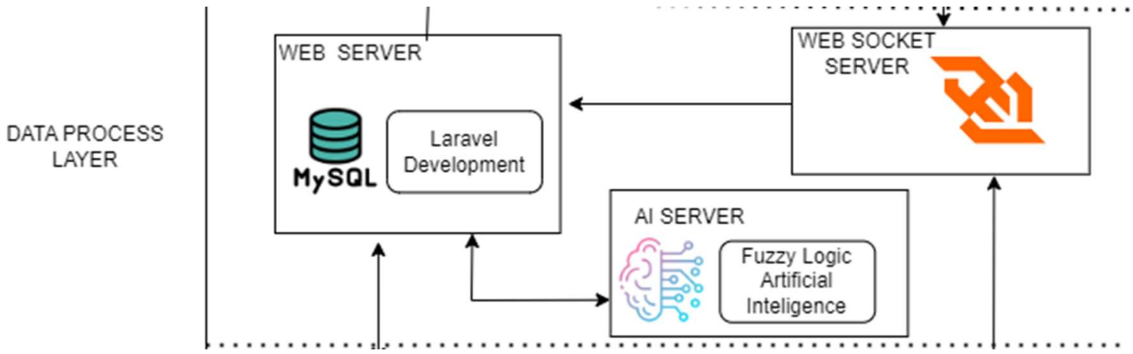


Figure 3.9: Application Layer of IoT Architecture

Next, the application layer architecture as shown in Figure 3.10 below where the user is able to see the result from the sensor. The application layer of this project is built using the Laravel framework which provides a user-friendly dashboard for monitoring and controlling the system. The dashboard allows for easy reporting, analytics based on the data collected from the sensors and processed by the fuzzy logic. It also provides visualizations of the sensor data, making it easy to understand and interpret. The dashboard also allows the user to configure the fuzzy logic parameters. With the dashboard, it is possible to monitor the system in real-time, view historical data, and make adjustments as necessary. It also allows the user to control "ON and "OFF" of the oxygen pump and water pump in real-time by sending and receiving data between the client and the server using WebSocket.

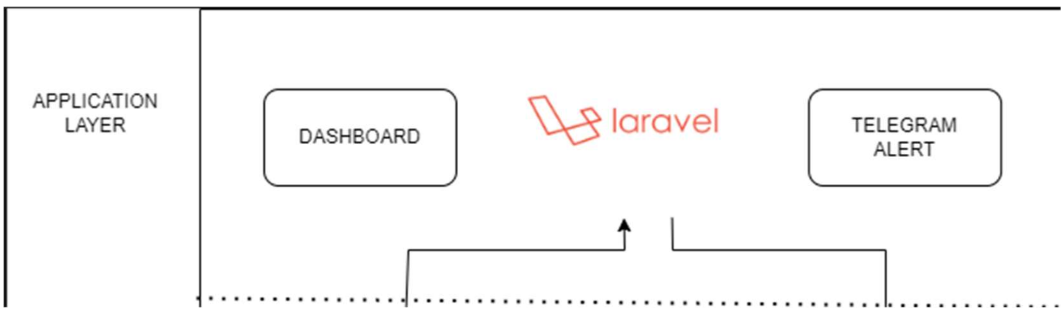


Figure 3.10: Application Layer of IoT Architecture

3.3.2 Circuit Diagram

The circuit diagram of the Paralysed Patient Monitoring System is shown in Figure 3.10. The Air Quality sensor is connected to the A1 analog pin and the AO pin, with its VCC and GND pins connected to the 5V and GND pins on the First Arduino Uno, respectively. The pulse rate sensor has 3 output pin which are VCC connected to 5V, GND connected to ground pin and Analog pin connected to A0 on the digital output pins on the Second Arduino, respectively. The infrared temperature sensor has its SCL and SDA pins connected to the SCL and SDA digital pins on the Arduino and its VSS and VDD connected to the 5V and GND power output pins, respectively. The Arduino transmits sensor data to the Raspberry Pi via a serial connection. The IV Infusion relay has its source, represented by an orange wire, connected to GPIO18 on the Raspberry Pi and its GND connected to a ground pin. The negative relay output is connected to the water pump and a pair of the negative output is connected to the 5V on the Raspberry Pi. The oxygen tank relay has the same circuit setup as the IV Infusion relay.

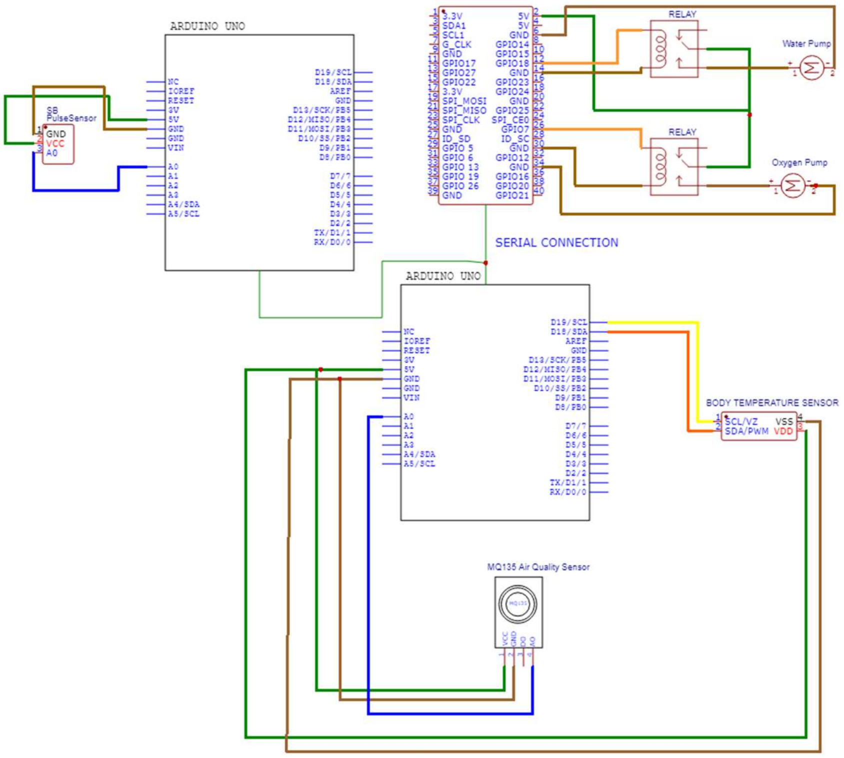


Figure 3.11: Circuit Diagram

3.3.3 Project Sketch

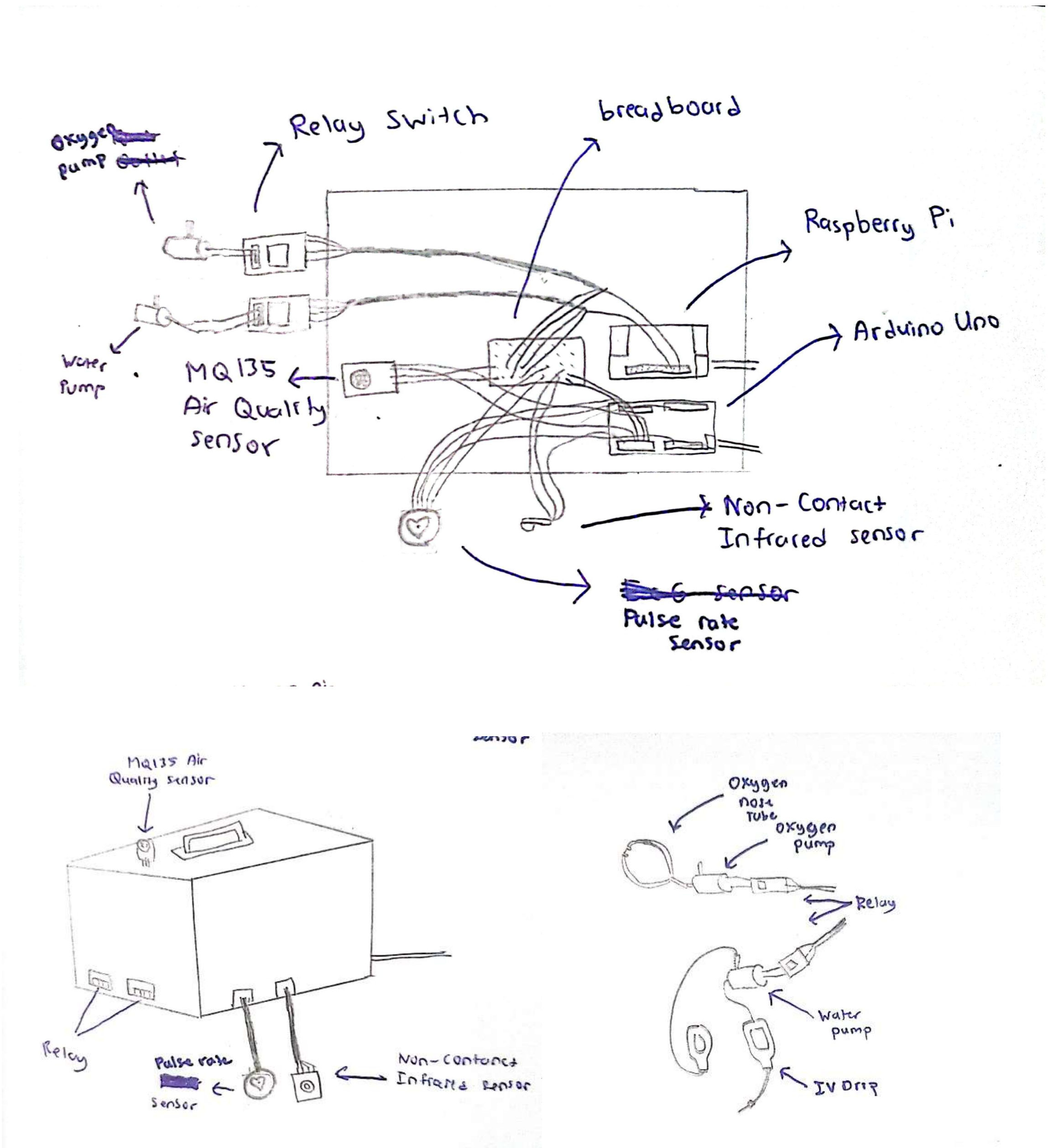


Figure 3.12: Project Sketch

3.4 Software and Hardware Requirements

There are multiple software requirements that will be used to develop the Paralysed Patient Monitoring Kit (PARAKIT) which are Arduino IDE is used to upload analogue sensor into Arduino UNO, Google Chrome is used to display the dashboard written in JSON and PHP files, MQTT is used to act as a broker that provides two-way communication for sending data between a client and a server, MySQL used as a local database storage to store all sensor data, and XAMPP Control Panel is used to run PHP 3.0 and MySQL.

Moreover, Visual Studio Code is used to run and execute python and PHP files, and VNC Viewer to run a virtual software for Raspberry Pi 3. Other than that, there are also multiple hardware requirements that will be used which consists of Arduino UNO, pulse rate sensor, Jumper wires, non-contact temperature sensor, MQ135 Air Quality Sensor, Relay Switch, Water Pump, Extension Plug Socket, and Raspberry Pi 3.

Figure 3.12 shows an image of Arduino UNO. Arduino UNO is an open-source microcontroller board that is low-cost, versatile, and user-friendly that can be used in a variety of electrical applications. In this project, it is used as the first microcontroller to read all the 3 data sensors which are pulse rate sensor, air quality sensor and non-contact infrared sensor since all sensor type used analogue pin on the microcontroller to take sensors data.

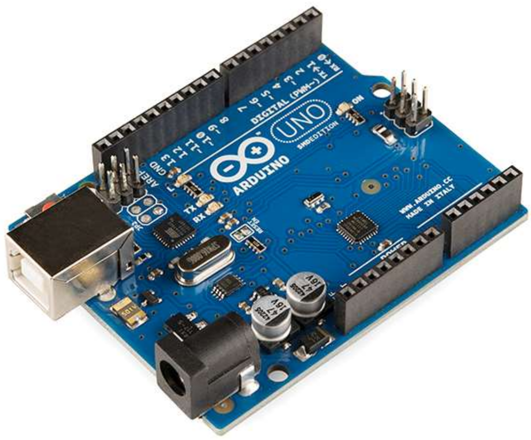


Figure 3.13: Arduino UNO

Figure 3.14 show the pulse rate sensor is a small electronic device that can also be useful in monitoring the heart rate of paralyzed patients. When a person becomes paralyzed, they may have limited or no ability to move their body. As a result, it can be challenging to monitor their heart rate and detect any changes that may indicate a medical emergency. However, by using a pulse rate sensor like the XD-58c, medical professionals can monitor the heart rate of paralyzed patients continuously and detect any changes that may require medical attention. For paralyzed patients, the pulse rate sensor can be attached to a limb or other part of the body where the skin is easily accessible. The sensor works by emitting light that penetrates the skin and is partially absorbed by the blood in the capillaries. The sensor then detects the amount of light reflected back, which changes with each heartbeat.

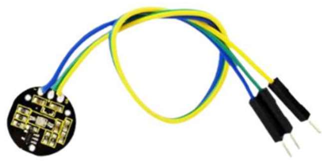


Figure 3.14: Pulse Rate Sensor

Figure 3.15 below shows a non-contact infrared sensor is a device specifically designed to collect body temperature data of paralyzed patients. It uses infrared radiation to detect the presence of an object or measure its temperature. These sensors work by emitting a beam of infrared radiation, which is then reflected back by the object being detected. The sensor then measures the intensity of the reflected radiation to determine the temperature of the object. Compared to other temperature sensors such as thermocouples and RTDs, non-contact infrared sensors offer several advantages for paralyzed patients, including faster response times, ease of use, and the ability to measure temperature from a distance without physical contact. They are particularly useful in clinical settings, as they can measure body temperature quickly and accurately without the need for invasive methods.

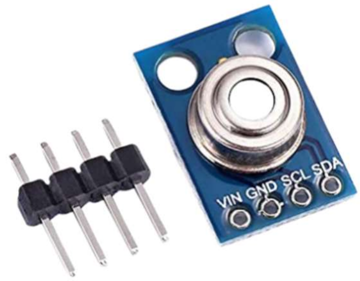


Figure 3.15: Non-Contact Temperature Sensor

Figure 3.16 show the MQ135 air quality sensor that detects and measures the concentration of various gases, including ammonia, sulfur dioxide, and nitrogen oxides. This sensor is commonly used in air quality monitoring applications and can be used to detect pollutants in the air. In the context of paralyzed patients, the MQ135 sensor could be used to monitor the air quality in the patient's room, ensuring that it is safe for them to breathe. This sensor could also be used to detect any potential respiratory irritants that may negatively impact the patient's health. Additionally, the sensor can be integrated with other monitoring systems to provide real-time data and alert caregivers if the air quality in the patient's room becomes unsafe.

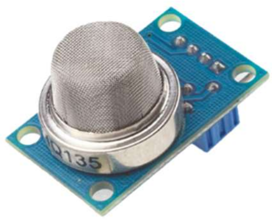


Figure 3.16: MQ135 Air Quality Sensor

Figure 3.17 show relay switches that used in the Paralysed Patient Monitoring System to control the "ON" and "OFF" status of the IV fluid and oxygen tank. The IV fluid and oxygen tank relays are activated or deactivated based on the readings from the sensors. In the event that the readings exceed the set threshold, the nurse or doctor will take appropriate action to adjust the IV fluid or oxygen tank accordingly." The use of relay switches ensures that the IV fluid and oxygen supply is properly managed and regulated, and the system operates effectively.

Figure 3.17: Relay Switch

Figure 3.18 show the water and oxygen pump hardware component that use fo IV fluid delivery system into patient body. It is used to pump water from an IV fluid bottle into another bottle that is used to measure the exact amount of fluid that will be dripped into the patient's body. The water pump ensures that the correct amount of fluid is delivered to the patient, helping to maintain their fluid balance and support their overall health. The accurate control of the water pump is crucial in ensuring the patient receives the right amount of IV fluid, and it is typically managed by certified nurses and doctors who monitor the data from the sensors and adjust the flow rate as needed.

Figure 3.18: Water and Oxygen Pump

Figure 3.19 shows jumper wires. A jumper wire is an electrical cable used to link distant electrical circuits on printed circuit boards. In this system, it is used to create connection to the microcontroller for all the sensors used.

Figure 3.19: Jumper Wires

Raspberry Pi 3 B+ as shown in figure 3.20 below is a small single-board computer that can be used as the foundation of a wide range of IoT-based projects, including paralyzed patient monitoring systems. The device is equipped with powerful processing capabilities, as well as built-in wireless connectivity options such as Wi-Fi and Bluetooth, making it an ideal choice for remote monitoring applications. Additionally, the Raspberry Pi 3 B+ can be easily integrated with a variety of sensors and actuators, such as temperature and humidity sensors, ECG sensors and more, making it possible to collect, process and transmit real-time data from paralyzed patients. With the aid of a web-based dashboard, the data can be easily accessed and analysed from anywhere, enabling certified nurses or doctors to monitor the health and well-being of their patients remotely. Overall, the Raspberry Pi 3 B+ is a powerful and versatile platform for building IoT-based paralyzed patient monitoring systems that can improve the quality of care for patients while reducing costs for healthcare providers.

Figure 3.20: Jumper Wires

Table 3.1 and 3.2 below show software and hardware requirements for Paralysed Patient Monitoring Kit (PARAKIT) respectively.

Table 3.1 Software requirements

Software	Description
Arduino IDE	To upload analog sensor into Arduino UNO.
Google Chrome	To display the dashboard written in JSON and PHP files.
MQTT	To act as a broker that provides two-way communication for sending data between a client and a server.
MySQL	A local database storage.
XAMPP Control Panel	To run PHP 3.0 and MySQL.
Visual Studio Code	To run and execute python and PHP files.
VNC Viewer	A virtual software for Raspberry Pi 3.
Laravel	To develop a analytics and real time dashboard

Table 3.2 Hardware requirements

Hardware	Description
Arduino UNO	To read Soil Moisture sensor.
Pulse Rate Sensor	To collect paralysed patient pulse rate data
Non-Contact infrared sensor data	To collect paralysed patient body temperature data
MQ135 Air Quality	To collect air quality in paralysed patient home
Jumper wires	To connect all sensors used to the microcontroller.
Relay	To control "ON" and "OFF" of water pump and oxygen pump
Raspberry Pi 3	A microcontroller/microprocessor to receive sensor nodes and execute python and PHP files.

3.5 Data Collection

In a paralyzed patient monitoring system using IoT, the data collected is crucial in providing real-time information about the patient's health status whether the patient condition is stable or unstable based on per hour monitoring. The data collected by the system will be from three different sensors, the MQ135 Air Quality sensor, the ECG sensor, and the non-contact pulse rate sensor.

The MQ135 Air Quality sensor can be used to detect various gases such as ammonia, sulphur dioxide, benzene, smoke, and CO2, which are important indicators of air quality. This information is crucial in monitoring air quality around patients, especially those with respiratory problems, and ensuring that the air is safe for them to breathe. To analyse the Air Quality in the patient's house, the Air Pollution Index (API) can be used to assess air quality and its impact on the patient's health. The index is divided into 5 categories, each corresponding to a specific range of index values and associated with potential health concerns. In Malaysia, the API follows the United States Air Quality Index (AQI) categories, with the exception that the "Unhealthy for Sensitive Group" and "Unhealthy AQI" categories are grouped together as "Unhealthy" in Malaysia [4].

The graph in figure 3.21 below provides a visual representation of the API levels represented by different shapes, each indicating a different level of air pollution. For example, a triangle shape represents good air quality with an API level of 0 to 50, while a square shape indicates moderate air quality with an API level of 51 to 100. The graph includes a key that explains the different shapes and corresponding API levels, making it easy for users to interpret the data. Thus, MQ135 will collect all the data from sensor and the data will value based on the Air Pollution index (API) as stated in the table below.

Air Pollution Index (API) Values	Levels of Health Concern	Colour	Shape
`	Good	Blue	
51 to 100	Moderate	Green	
101 to 200	Unhealthy	Yellow	
201 to 300	Very Unhealthy	Orange	
>300	Hazardous	Red	

Figure 3.21 Air Pollution Index Table

A pulse rate sensor is a device that detects and measures the heart rate or pulse of an individual. It works by detecting the blood flow through the arteries, which increases and decreases as the heart beats.. Resting heart rate, which is the heart rate when a person is at rest, is an important indicator of overall health. A high resting heart rate is often associated with various health conditions such as heart disease, hypertension, and stress. Monitoring resting heart rate using a pulse rate sensor can be a useful tool in identifying changes in heart rate and potentially detecting health issues early. Figure 3.20 below shows the resting heart rate based on gender that will be used to monitor paralyzed patient heart rate condition in this project [5].

RESTING HEART RATE
BY AGE AND GENDER

AGE	ATHLETE	VERY GOOD	ABOVE AVERAGE	AVERAGE	BELOW AVERAGE	POOR
20-39	47-54	55-60	61-68	69-75	76-83	84-94
40-59	46-54	55-60	61-67	68-76	77-84	85-94
60-79	45-53	54-59	60-66	67-74	75-83	84-97

AGE	ATHLETE	VERY GOOD	ABOVE AVERAGE	AVERAGE	BELOW AVERAGE	POOR
20-39	52-59	60-65	66-73	74-81	82-88	89-98
40-59	51-58	59-63	64-70	71-78	79-85	86-96
60-79	52-58	59-63	64-69	70-77	78-85	86-95

Figure 3.22 Resting Heart Rate Table

The non-contact temperature sensor utilizes infrared technology to measure the patient's body temperature without the need for physical contact. This makes it an ideal option for monitoring patients who may have mobility limitations or are unable to physically move. The data collected from the sensor provides real-time information about the patient's body temperature, which is crucial in monitoring their overall health and detecting any potential fever or temperature-related issues. The collected data will be referred to figure 3.23 below to monitor patient body temperature in real time [6].

Standard	Celsius	Fahrenheit
Hypothermia	<35°C	95.0°F
Normal	36.5-37.5°C	97.7-99.5°F
Fever/Hyperthermia	>37.5or 38.3°C	99.5-100.9°F
Hyperpyrexia	>40.0 or 41.5°C	104.0-106.7°F

Figure 3.23 Standard body temperature range table

Overall, the data collected from these sensors will provide comprehensive information about the patient's health status, helping healthcare providers make informed decisions about their care.

3.6 Data Classification using Fuzzy Logic

Fuzzy logic is a mathematical approach to handle uncertainty in decision making. In order to apply this method to classify the condition of paralyzed patients, a set of fuzzy rules must be established [7]. The fuzzy rules consist of two main components: antecedents and consequences. Antecedents represent the input variables that the system receives, in this case, the Heart Rate data from the ECG sensor, the Air quality data from the MQ135 Air Quality sensor, and the body temperature data from the non-contact pulse rate sensor. The consequences, on the other hand, represent the output of the system, which is the condition of the paralyzed patient whether stable or unstable. These fuzzy rules are designed to capture the relationship between the input variables and the output and help the system to determine the condition of the paralyzed patient based on the collected data. Table 3.3 below shows the data classification for fuzzy logic rules to be implemented in the system.

Table 3.3 Heart Rate Membership Function

Pulse Rate Data(BPM)	
45 – 55	Low
56 – 75	Normal
76 – 95	High

Table 3.4 Heart Rate Membership Function

Air Pollution Index Data (API)	
0 – 100	Good
101 - 200	Moderate
300 -500	Unhealthy

Table 3.5 Air Quality Membership Function

Body Temperature Data (BPM)	
35 – 36.4	LowTemp
36.5 – 37.5	NormalTemp
37.5 – 38.3	HighTemp

Table 3.6 Data classification table for fuzzy logic rules # Normal, Caution, Emergency

Heart Rate data (BPM)	Air Pollution Index Data (Percent)	Body Temperature Data (Celsius)	Condition
Low	Moderate	Low	Caution
Normal	Good	Normal	Normal
High	Unhealthy	High	Emergency

3.7 Evaluation

3.7.1 AI Techniques

The proposed system utilizes the Artificial Intelligence (AI) technique of Fuzzy Logic. Fuzzy Logic is a mathematical logic that operates with degrees of truth rather than strict true or false values. It allows variables to have values ranging between 0 and 1, indicating the degree to which they are true or false. In this system, Fuzzy Logic is used to determine the condition of the paralyzed patient. The output of the Fuzzy Logic will be either stable or unstable, depending on the patient's condition over a period of one hour. If the patient's condition is found to be unstable, the doctor can take the necessary steps such as contacting the patient's family or visiting their home. The fuzzy logic system is tested using fuzzy rules, which consist of antecedents and consequences. The antecedents in this case are the Heart Rate data, Air Quality data and Body Temperature data, while the consequence is the condition of the paralyzed patient.

3.7.2 IoT Dashboard Mock-up

An IoT dashboard is a visual representation of data collected from Internet of Things (IoT) devices. It allows users to monitor and control connected devices in real-time, providing valuable insights into the performance and behavior of the devices. The dashboard typically displays data in the form of graphs, charts, and other visual aids, making it easy to understand and interpret the information.

The IoT dashboard, as shown in Figure 3.25, is divided into four crucial pieces of information. Firstly, the sensor status section displays whether the sensors are currently running or not. If the sensors are functioning properly, the status will be updated to "Online". The next section is for notifications, which is used to visualize any updates or alerts from the sensors. Following that, there is a historical data section, where certified nurses or doctors can analyze up to four recent data points, with new data being added every 5 minutes. Two sensors, temperature and pulse rate are displayed using graph visualizations, while the air quality sensor uses a gauge visualization.

Finally, the last part of the dashboard is the fuzzy logic output, which displays the patient's condition computed by fuzzy logic, based on the latest sensor data from MySql. Then, two “ON” and “OFF” button for controlling oxygen tank and IV fluid is based on 30 minutes patient condition data. If the 30 minutes data show unstable condition and received alerts message more than 20 minutes, the certified nurse can turn on the button to stabilise the patient.

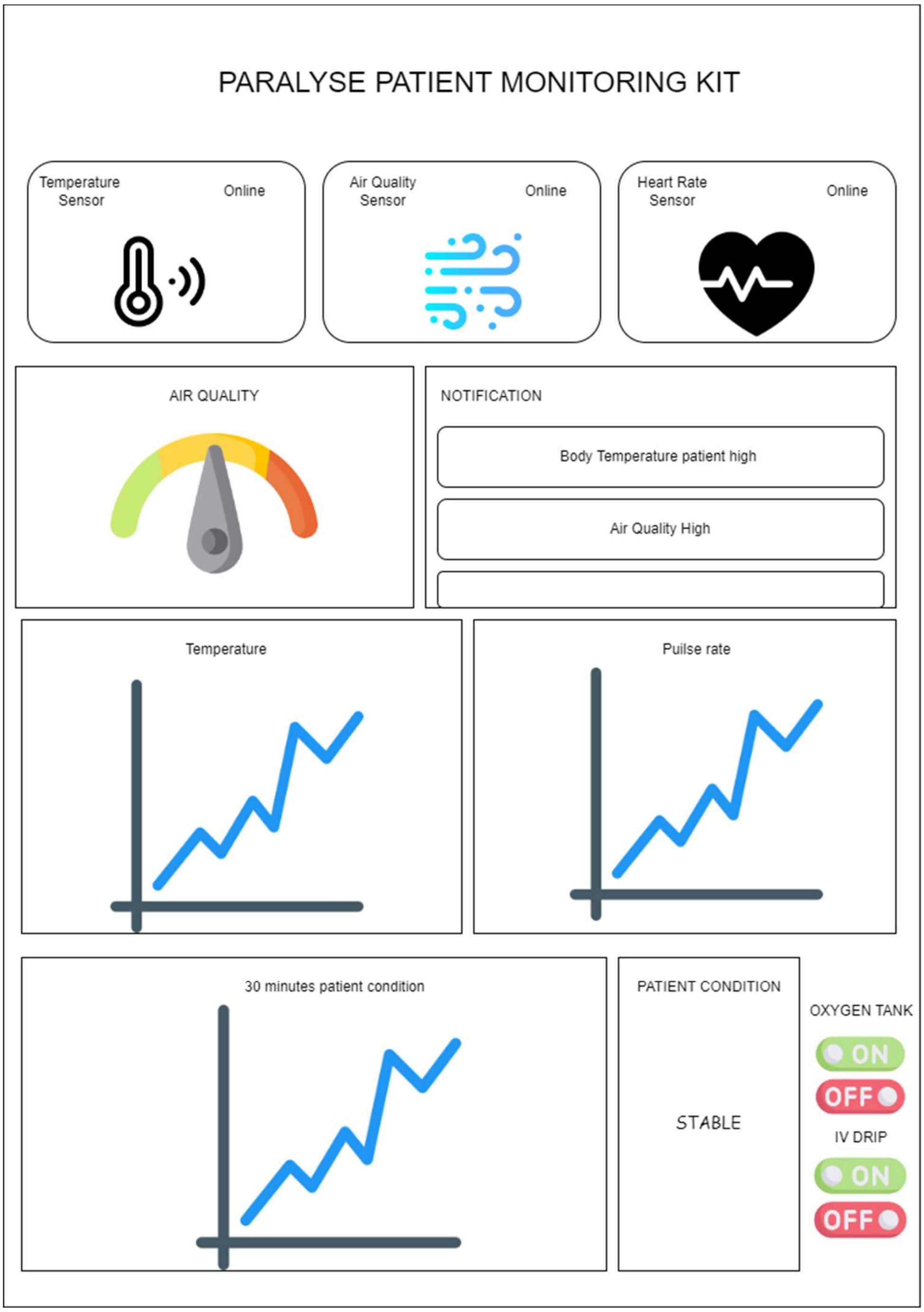


Figure 3.24 Standard body temperature range table

The usability and user-friendliness of the dashboard will be evaluated through a User Acceptance Test (UAT). A group of respondents will be selected to participate in the test and provide feedback on their experience with the dashboard. TABLE 3.7 outlines the plan for conducting the UAT and evaluating the performance of the dashboard.

Table 3.7 Data classification table for fuzzy logic rules # Normal, Caution, Emergency

No	Acceptance Requirement	Test Result		Comments
		Success	Fail	
1	All sensor data represented by using graphs			
2	All sensor data obtained in real time			
3	The user interface of the dashboard is easy to read and eye-catching			
4	All the buttons are functioning well			
5	User register, login and logout			
6	Fuzzy results are displayed			
7	Alarm can be triggered			

3.8 Summary

Hence, in Chapter 3 we can conclude that the project design, project flowchart, project architecture, software and hardware requirements, data collection and data classification using fuzzy logic were explained in detail. All the requirements involved in making this Paralysed Patient Monitoring Kit have been planned in detail in terms of suitable hardware and software necessities and Artificial Intelligence technique use to depict paralysed patient condition, and simplicity to achieve a sustainable, dependable and trouble-free system.

CHAPTER 4

IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 Introduction

This chapter discusses the research's implementation, result, and findings. This chapter presents the findings from the experiment or testing that was performed. It also contains an explanation of the conversation, demonstrating that the study objectives were met.

4.2 Implementation Process

Implementation is a process to record all the steps in developing the Paralyse Patient Monitoring Kit (PARAKIT). This section, student should discuss in detail on the implementation process that involve in completing this project from the beginning - development process, coding involved and interface and result of research gathering.

4.2.1 Development of Paralysed Patient Monitoring System (PARAKIT)

4.2.1.1 HARWARE SETUP

In the hardware development of the Paralysed Patient Monitoring Kit that refer to figure 4.1 below, a container acts as a kit that contains all the necessary components. The kit includes two Arduino boards, one Raspberry Pi, one breadboard, and two relays. Each of these components serves a critical purpose in the functionality of the system. The first relay controls the IV drip, while the second relay is responsible for managing the oxygen tank. The inclusion of both relays enables the system to control these two essential components, ensuring that the patient's oxygen and medication needs are met. One of the Arduinos is connected to a non-contact infrared sensor and an air quality sensor, which helps to monitor the patient's environment and detect potential issues with air quality. The other Arduino is connected to a pulse rate sensor, which requires its own reading timer to ensure accurate readings. By including all of these components in the kit, the

PARAKIT is able to provide comprehensive monitoring and control, helping to improve the safety and comfort of paralysed patients in their homes.

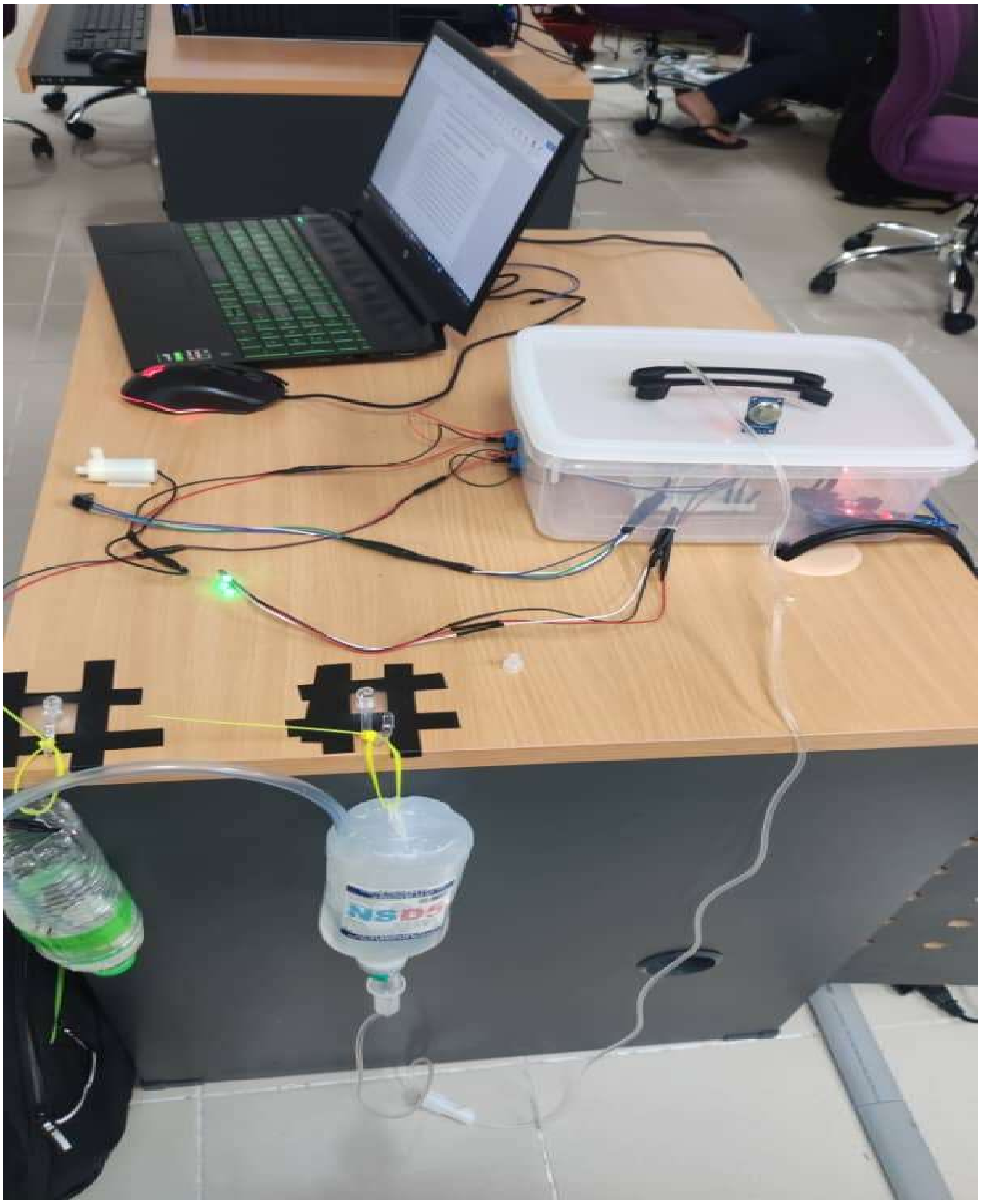


Figure 4.1: Hardware Implementation

4.2.1.2 MQTT AND WEBSOCKET SETUP

Now that the setup is complete, the next step is to implement the functionality to send data from the sensing layer to the data processing layer, where it will be stored in a database using the MQTT network layer protocol [8]. Additionally, the system will also be able to control its actuators through the WebSocket network layer protocol, which enables the application layer to send triggers back to the sensing layer. This is explained in Chapter 3.4.3 to 3.4.5 of the documentation.

To accomplish this, coding development is necessary, and four Python programs will be created, publishAirQua_Temp.py and publishPulRate.py for publishing data, and controlOutput.py and checkond.py for listening to WebSocket events that will enable the control and condition of actuator whether it "ON" or "OFF" condition. These programs will be continuously executed within the RaspberryPi environment. The results of the code executions are provided in the figures that follow.

4.2.1.3 WEB APPLICATION DEVELOPMENT SETUP

The web application development process cannot begin until the hardware setup is fully completed. The following subchapter will cover various configurations in the Laravel framework, including the creation of database and API controllers, routing, event channels, and the development process for each page. These discussions will provide insight into the implementation of the Laravel framework and help in the development of the web application.

i. CONTROLLER SETUP

In the development of PARAKIT, the DBController plays a critical role in managing the application's data, specifically related to patients' health conditions. The DBController is responsible for handling interactions with the database and provides methods for retrieving and storing data in the temperature, pulse_rate, air_quality, fuzzylogic, and alaramstatusos tables based on figure 4.2 and 4.3 [9]. These tables contain critical information related to a patient's health, including their body temperature, pulse rate, and air quality, which are essential for tracking their health status.

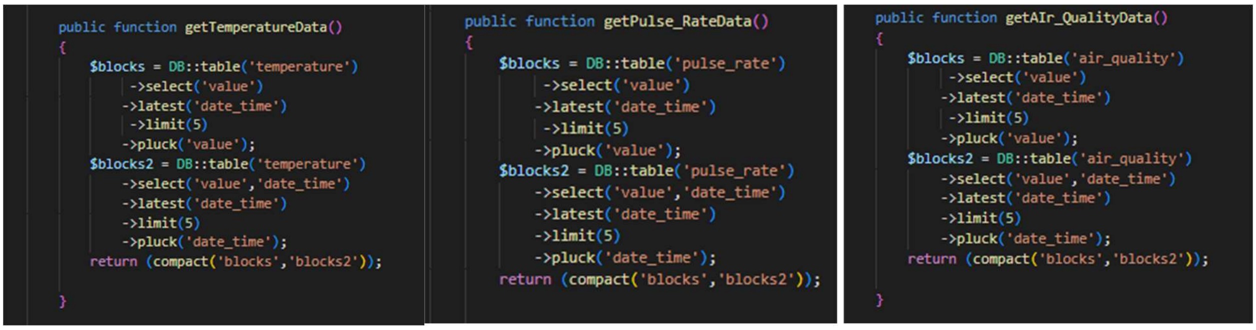
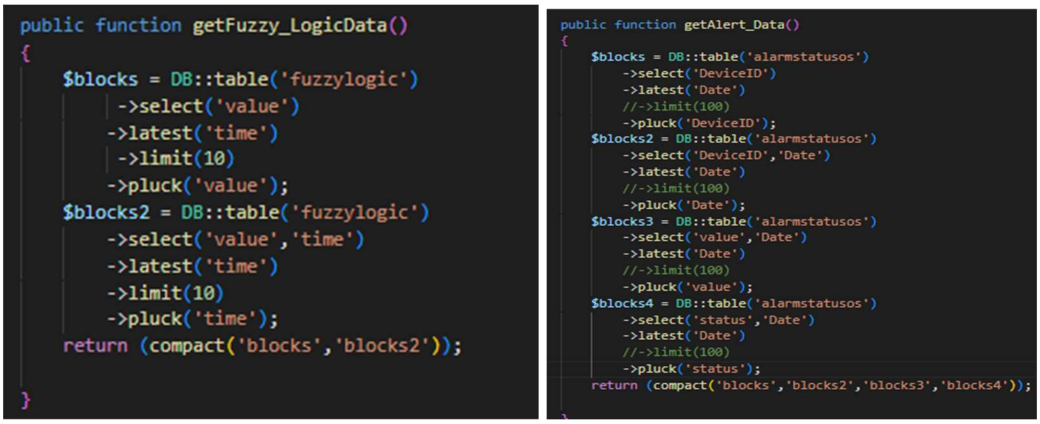


Figure 4.2: Temperature and pulse Rate and air quality Table (dbController)

The fuzzylogic table stores data related to a patient's condition computed through fuzzy logic, which helps to provide more accurate insights into their health status. The alaramstatusos table is particularly important as it keeps track of caution and emergency patient conditions and triggers alarms as necessary. The alarmstatusos table is different as there are 4 value data need to be subcribe from the alarmstatusos table to display in dashboard. By storing this information in a structured way, the DBController allows for efficient querying and manipulation of the database, ensuring that the data is correctly processed and stored.

The constructor method initializes the temperature, pulse_rate, and air_quality models, allowing the controller to interact with these models and their associated database tables.

The three methods, getTemperatureApi, getPulseRateApi, and getAirQualityApi, each correspond to an API endpoint that returns the latest value for the temperature, pulse rate, and air quality, respectively. These methods use the "DB::table" method to query the database and retrieve the most recent value for each data type. The "latest" method is used to get the most recent data based on the "date_time" column, and the "limit" method is used to limit the result to only the latest value. Finally, the "pluck" method is used to return only the "value" column from the database query.



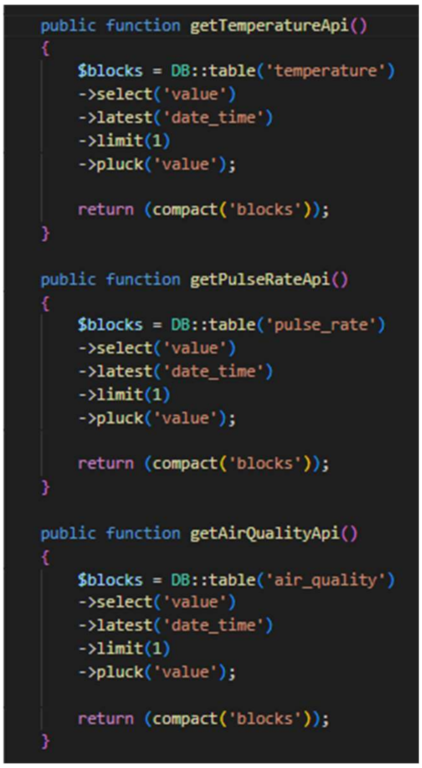
```
public function getFuzzy_LogicData()
{
    $blocks = DB::table('fuzzylogic')
        ->select('value')
        ->latest('time')
        ->limit(10)
        ->pluck('value');
    $blocks2 = DB::table('fuzzylogic')
        ->select('value','time')
        ->latest('time')
        ->limit(10)
        ->pluck('time');
    return (compact('blocks','blocks2'));

}
```

```
public function getAlert_Data()
{
    $blocks = DB::table('alarmstatusos')
        ->select('DeviceID')
        ->latest('Date')
        //->limit(100)
        ->pluck('DeviceID');
    $blocks2 = DB::table('alarmstatusos')
        ->select('DeviceID','Date')
        ->latest('Date')
        //->limit(100)
        ->pluck('Date');
    $blocks3 = DB::table('alarmstatusos')
        ->select('value','Date')
        ->latest('Date')
        //->limit(100)
        ->pluck('value');
    $blocks4 = DB::table('alarmstatusos')
        ->select('status','Date')
        ->latest('Date')
        //->limit(100)
        ->pluck('status');
    return (compact('blocks','blocks2','blocks3','blocks4'));
```

Figure 4.3: fuzzylogic and alarmstatusos Table (dbController)

The fugure 4.4 below is an example of an API controller in Laravel. The purpose of an API controller is to provide a mechanism for external systems to communicate with the web application and exchange data. In this case, the controller provides APIs for getting the latest temperature, pulse rate, and air quality data. The controller uses the DB class, which is part of the Laravel database abstraction layer, to interact with the database and retrieve the necessary data.



```
public function getTemperatureApi()
{
    $blocks = DB::table('temperature')
    ->select('value')
    ->latest('date_time')
    ->limit(1)
    ->pluck('value');

    return (compact('blocks'));
}

public function getPulseRateApi()
{
    $blocks = DB::table('pulse_rate')
    ->select('value')
    ->latest('date_time')
    ->limit(1)
    ->pluck('value');

    return (compact('blocks'));
}

public function getAirQualityApi()
{
    $blocks = DB::table('air_quality')
    ->select('value')
    ->latest('date_time')
    ->limit(1)
    ->pluck('value');

    return (compact('blocks'));
}
```

Figure 4.4: fuzzylogic and alarmstatusos Table (dbController)

ii. ROUTES SETUP

In this Laravel code below on figure 4.5 , the routes are defined to determine how the server should respond to the client's request. All the routes are defined in the web.php file. It first registers a middleware called auth, which requires the user to be authenticated to access the routes. After this, several routes are defined for different pages and functionalities of the application.

For instance,"Route::get('/route/temperatureroute','\App\Http\Controllers\dbcontroller@getTemperatureData')" defines a route for getting the temperature data from the database, and it calls the getTemperatureData function from the dbcontroller. Similarly, other routes are defined for getting pulse rate, air quality, fuzzy logic, and alert data.

Apart from data retrieval, there are also routes for controlling some features of the application, like turning "DRIP-ON", "DRIP-OFF", "OXYGEN-ON", and "OXYGEN-OFF". These routes are defined using the "Route::post" method, which listens for a POST request. When the request is made, it triggers an event that notifies the Raspberry Pi that the control the actuator.

```php
Route::group(['middleware' => 'auth'], function () {

    Route::get('/route/temperatureroute', '\App\Http\Controllers\dbcontroller@getTemperatureData');
    Route::get('/route/pulse_rateroute', '\App\Http\Controllers\dbcontroller@getPulse_RateData');
    Route::get('/route/air_qualityroute', '\App\Http\Controllers\dbcontroller@getAIr_QualityData');
    Route::get('/route/fuzzylogicroute', '\App\Http\Controllers\dbcontroller@getFuzzy_LogicData');
    Route::get('/route/alertroute', '\App\Http\Controllers\dbcontroller@getAlert_Data');


    Route::post('/controlpagetempon1', function () {
        $message = "DRIP-ON";
        event(new App\Events\patientmonitor($message));
        return redirect('dashboard2');
    });

    Route::post('/controlpagetempoff1', function () {
        $message = "DRIP-OFF";
        event(new App\Events\patientmonitor($message));
        return redirect('dashboard2');
    });

    Route::post('/controlpagetempon2', function () {
        $message = "OXYGEN-ON";
        event(new App\Events\patientmonitor($message));
        return redirect('dashboard2');
    });

    Route::post('/controlpagetempoff2', function () {
        $message = "OXYGEN-OFF";
        event(new App\Events\patientmonitor($message));
        return redirect('dashboard2');
    });

    Route::get('/', function () {
        return view('home');
    });
```

Figure 4.5: Routes setup for Web Application

Next is API routes that act as a public routes that call the corresponding methods in the “patientApiController”. They are defined with the get HTTP method and their respective URIs: “tempApiRoute”, “pulseApiRoute”, and “airQualityApiRoute”. Each of these routes is associated with a specific method in the “patientApiController”, which retrieves the latest data from the corresponding table in the database and returns it in JSON format. These API routes can be used by AI server to retrieve latest sesors data from the database to compute fuzzy logic.

```php
Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('tempApiRoute', '\App\Http\Controllers\patientApiController@getTemperatureApi');
Route::get('pulseApiRoute', '\App\Http\Controllers\patientApiController@getPulseRateApi');
Route::get('airQualityApiRoute', '\App\Http\Controllers\patientApiController@getAirQualityApi');
```

Figure 4.6: Routes setup for API latest sensors data

4.2.1.4 DASHBOARD

The first phase of the dashboard is a comprehensive display of the latest data collected from three different sensors. These sensors measure three critical aspects of a patient's condition: body temperature, air quality, and pulse rate. Each data type is clearly categorized with its own unit of measurement, making it easier for healthcare professionals to quickly interpret the information. The body temperature data is displayed in degrees Celsius (°C), the air quality data is displayed using the Air Pollution Index (API), and the pulse rate data is displayed in beats per minute (BPM).

Additionally, a digital clock has also been implemented on the dashboard to help healthcare professionals monitor the current condition of the patient more effectively. This clock allows them to keep track of the time and ensure that they are giving the patient the proper attention and care that they need. The clock is accurate and synchronized with the current time, ensuring that healthcare professionals have the most up-to-date information at their disposal.

The first phase of the dashboard also visualizes an intelligent artificial display. This phase is focused on displaying the results from analyzing the data collected from the three sensors to determine the patient's condition. The "FUZZY 30 minutes of data history" is a display that shows a history of the patient's condition based on 30 minutes of results from fuzzy logic calculations. This display is crucial in helping healthcare professionals make informed decisions about the patient's care, as it provides a clear and concise overview of the patient's condition over time. For example, if a patient's condition changes from normal to an "EMERGENCY" condition within a 20-minute period, this display makes it easier for the doctor to quickly identify the change and take appropriate action, such as calling an ambulance or taking the patient to the hospital for treatment. Finally, there are "ON" and "OFF" buttons for the IV DRIP and OXYGEN TANK. These buttons are used to control the relay switch that is connected to the water pump for the IV DRIP and the oxygen pump for the OXYGEN TANK. The buttons can only be controlled by certified nurses and doctors based on the results of the latest data from one or all three sensors. This allows healthcare professionals to make informed decisions about the patient's care and provide them with the necessary treatment and support.

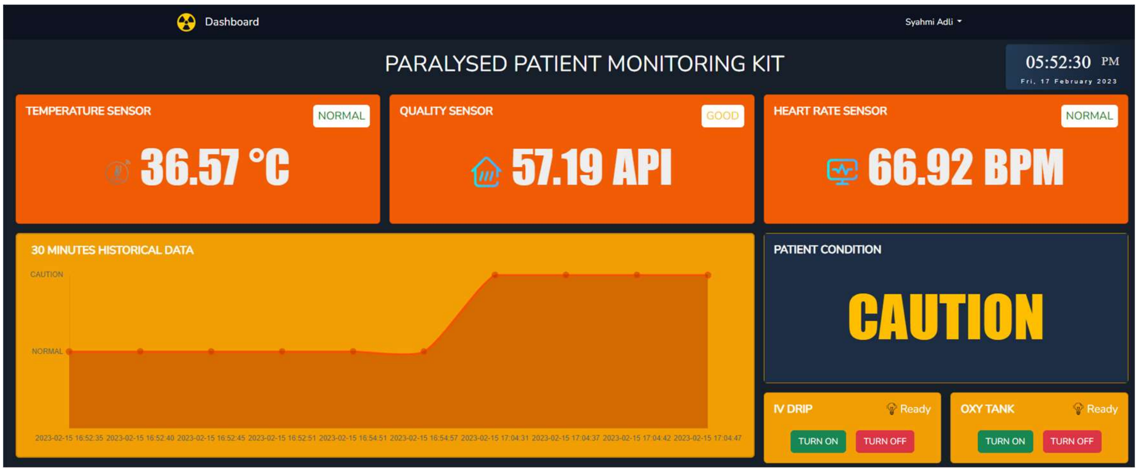


Figure 4.7: Dashboard (First Phase)

The second part of the system based on figure 4.8 below is focused on the alert phase and historical data visualization. This phase has several key features that help doctor and certified nurse monitor the patient's condition and respond quickly in case of any unusual changes. First, the map visualization feature is implemented in the system to display the location of the patient and PARAKIT. Next, we have the alert historical table data. This table is designed to display the alert historical data generated from the Fuzzy Logic implementation in the AI server. All the alert results generated in the AI server will be stored in the "alarmstatusos" table on the webserver. Then, all the data from the "alarmstatusos" table will be displayed in the notification table on the dashboard. This can be helpful for certified nurses and doctors to analyze the historical condition of the paralysed patient.

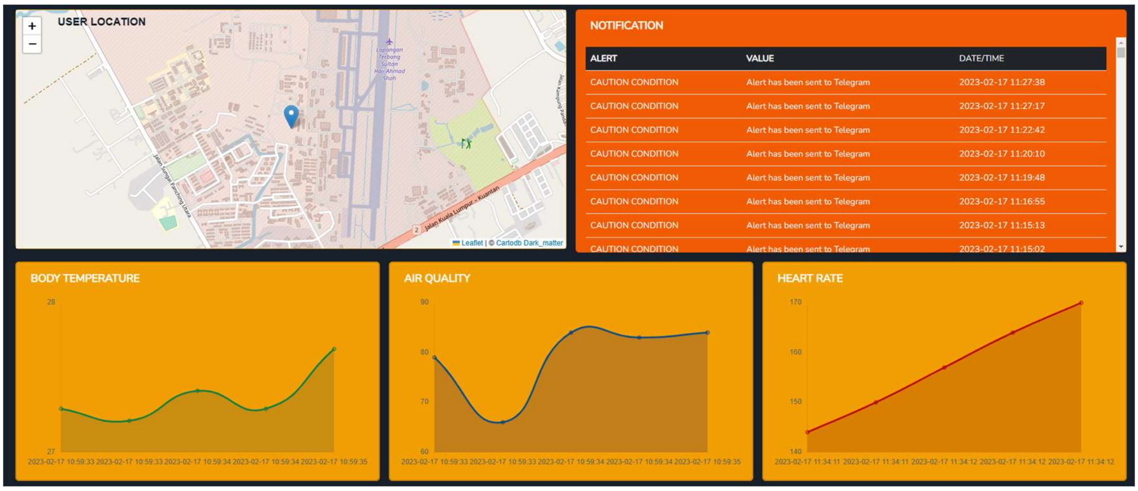


Figure 4.8: Dashboard (Second Part)

The alert function, in addition to being stored in the database, will also be transmitted to the mobile phone of the certified nurses and doctors through the Telegram application based on figure 4.9. This provides an added layer of convenience and accessibility for the medical professionals to monitor the condition of paralyzed patients. The real-time notification feature of Telegram enables the healthcare providers to receive alerts instantly, even when they are not physically present at the location of the patient. This is crucial in ensuring that timely and effective action can be taken in case of any emergencies. Furthermore, the use of Telegram for transmitting the alert messages adds a level of security and privacy, as the messages are encrypted and can only be accessed by authorized individuals. The integration of the alert function with the Telegram application enhances the overall efficiency of the system, enabling the medical professionals to monitor the health of paralyzed patients more effectively.

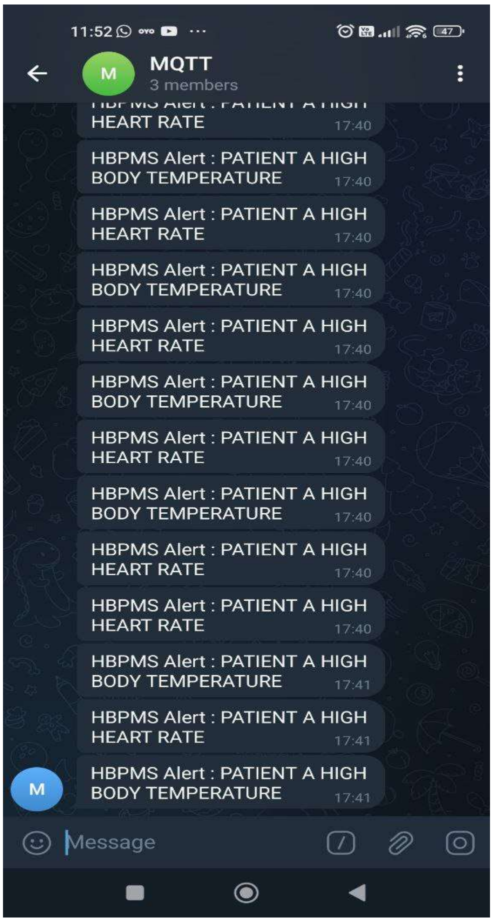


Figure 4.9: Telegram alert function

4.2.1.5 AI IMPLEMENTATION

In AI implementation, the first aspect to consider is antecedent control. This project involves three input datasets: pulserate ranging between 40 to 100, temperature ranging between 35 to 39, and air quality data ranging between 0 to 300. The antecedent for the output is patient condition, which ranges from 0 to 100 percent.

Next is the membership function based on figure 4.5, figure 4.6 and figure 4.7 below, which is a curve that maps the input data points to their membership values [10]. The primary set used in this fuzzy logic method is Pulse Rate data, where the range between 45 to 60 is categorized as "Low", 55 to 65 as "Normal", and 70 to 95 as "High". The temperature range between 35 to 37 is labelled as "Low", 36 to 37.5 as "Normal", and 37 to 39 as "High". For the air quality data, the range between 0 to 100 is categorized as "Low", 80 to 220 as "Moderate", and 200 to 300 as "Unhealthy". The ranges for pulse rate, body temperature, and water quality are based on the data collection described in subchapter 3.6, specifically in figures 3.20, 3.22, and 3.19.

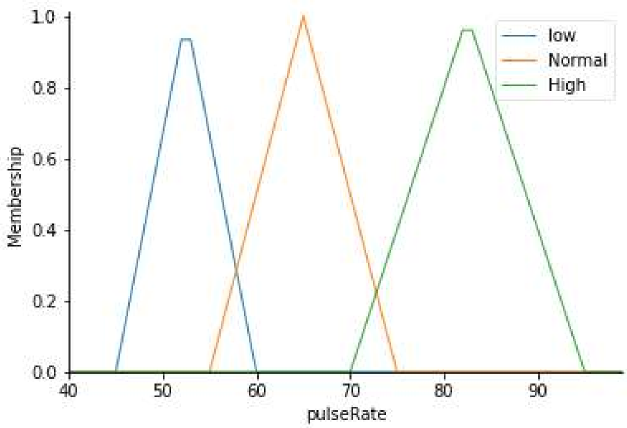


Figure 4.10: Pulse Rate Membership

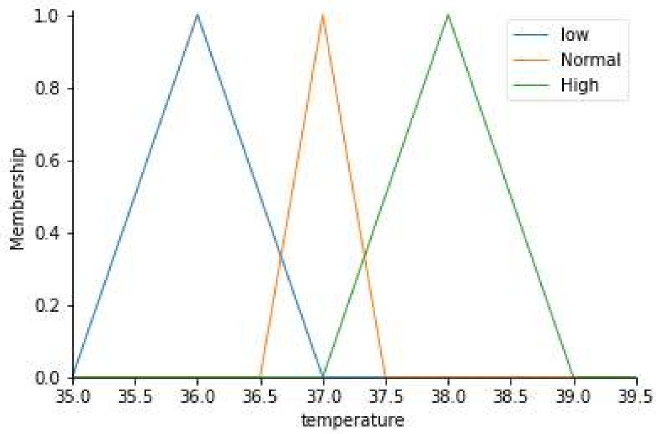


Figure 4.11: Temperature Membership

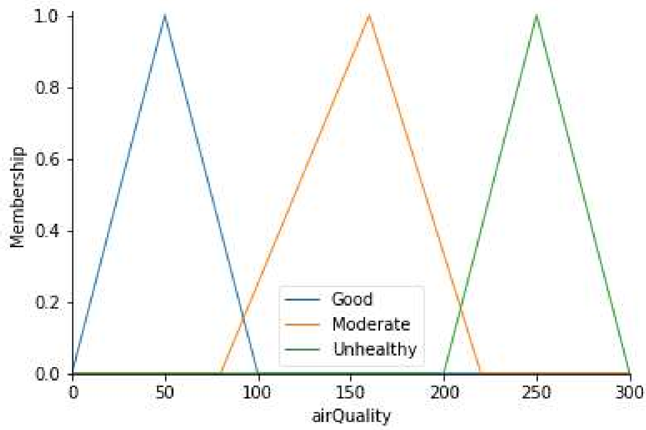


Figure 4.12: Air Quality Membership

The patient condition membership data is in percentage units. The range between 0 to 40 is categorized as "Normal", 30 to 70 as "Caution", and 60 to 100 as "Emergency".

Finally, to ensure proper operation of Fuzzy Logic, rules must be added to the membership sets. For this fuzzy method, there are only three rules. The first rule states that if pulseRate is "High", temperature is "High", and airQuality is "High", then the patient condition will be "Emergency". The second rule states that if pulseRate is "Low", temperature is "Low", and airQuality is "Moderate", then the patient condition will be "Caution". The third and final rule states that if pulseRate is "Normal", temperature is "Normal", and airQuality is "Good", then the patient condition will be "Normal". The code below shows the whole process and step for fuzzy logic to compute all the antecedents and measure based on rules given.

```
# Define the fuzzy sets for the inputs
pulseRate = ctrl.Antecedent(np.arange(40, 100, 1), 'pulseRate') #range = 400
temperature = ctrl.Antecedent(np.arange(35, 39, 0.5), 'temperature') #range = 200
airQuality = ctrl.Antecedent(np.arange(0, 300, 10), 'airQuality')

# Define the fuzzy sets for the output
patientCondition = ctrl.Consequent(np.arange(0, 100, 1), 'patientCondition')

# Define the membership functions
pulseRate['Low'] = fuzz.trimf(pulseRate.universe, [45, 52.5, 60])
pulseRate['Normal'] = fuzz.trimf(pulseRate.universe, [55, 65, 75])
pulseRate['High'] = fuzz.trimf(pulseRate.universe, [70, 82.5, 95])

temperature['Low'] = fuzz.trimf(temperature.universe, [35, 36, 37])
temperature['Normal'] = fuzz.trimf(temperature.universe, [36.5, 37, 37.5])
temperature['High'] = fuzz.trimf(temperature.universe, [37, 38, 39])

#create membership function for organic carbon
airQuality['Good'] = fuzz.trimf(airQuality.universe, [0, 50, 100])
airQuality['Moderate'] = fuzz.trimf(airQuality.universe, [80, 160, 220])
airQuality['Unhealthy'] = fuzz.trimf(airQuality.universe, [200, 250, 300])

pulseRate.view()
temperature.view()
airQuality.view()

patientCondition['Normal'] = fuzz.trimf(patientCondition.universe, [0, 20, 40])
patientCondition['Caution'] = fuzz.trimf(patientCondition.universe, [30, 50, 70])
patientCondition['Emergency'] = fuzz.trimf(patientCondition.universe, [60, 80, 100])

patientCondition.view()

# Define the rule base using the 'rule' method

rule1 = ctrl.Rule(pulseRate['High'] | temperature['High'] | airQuality['Unhealthy'], patientCondition['Emergency'])
rule2 = ctrl.Rule(pulseRate['Low'] | temperature['Low'] | airQuality['Moderate'], patientCondition['Caution'])
rule3 = ctrl.Rule(pulseRate['Normal'] | temperature['Normal'] | airQuality['Good'], patientCondition['Normal'])
```

Figure 4.13: Fuzzy Logic Code (First Section)

After the membership and rules have been set, the rules that have been established will be processed. The three input data sets will be retrieved from the MySql database through the Application Programming Interface (API) function, as shown in the coding diagram on lines 61 to 97. The code on lines 107 to 117 serves to send the result of the fuzzy logic calculation to the dashboard and to store the data in the database by pushing it to the MQTT broker.

```python
# Create the control system using the rules
patCond_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])

# Create a control system simulation
patCond= ctrl.ControlSystemSimulation(patCond_ctrl)

file = open('important.pkl', 'wb')
pickle.dump(patCond,file)
file.close()

while True:

    pusher_client = pusher.Pusher(app_id='1', key=u'umpfkpusher', secret=u'u%M15z2h%3A', cluster=u'mt1', ssl=False, host=u'10.26.30.32', port=6001)

    URLdustdensity = "http://10.66.38.41:8000/api/tempApiRoute"
    rdustdensity = requests.get(URLdustdensity)
    datar = rdustdensity.json()
    myresult1 = datar['blocks'][0]
    myresult1 = np.float32(myresult1)

    #thi use for status alarm
    URLcarbonmonoxide = "http://10.66.38.41:8000/api/pulseApiRoute" #tukar ikut route api dengan ip
    rcarbonmonoxide = requests.get(URLcarbonmonoxide)
    datar = rcarbonmonoxide.json()
    myresult2 = datar['blocks'][0]
    myresult2 = np.float32(myresult2)

    URLcarbonmonoxide = "http://10.66.38.41:8000/api/airQualityApiRoute" #tukar ikut route api dengan ip
    rcarbonmonoxide = requests.get(URLcarbonmonoxide)
    datar = rcarbonmonoxide.json()
    myresult3 = datar['blocks'][0]
    myresult3 = np.float32(myresult3)

    patCond.input['temperature'] = myresult1
    patCond.input['airQuality'] = myresult3
    patCond.input['pulseRate'] = myresult2


# In[19]:



    patCond.compute()



    pusher_client.trigger(u'CA19146patient', u'App\Events\patientmonitor', {u'Pred1': str(patCond.output['patientCondition'])})
    client.publish("CA19146/PATIENTCONDDATA",str(patCond.output['patientCondition']),0)
    print('publish to socket')

    # In[20]:


    print(patCond.output['patientCondition'])
    patientCondition.view(sim=patCond)

    time.sleep(5)
```

Figure 4.14: Fuzzy Logic Code (Second Section)

4.3 Result and Discussion

The conclusion of the data collected from the three sensors discussed in Subchapter 4.1 and Subchapter 4.2 can be effectively evaluated using Fuzzy Logic. Table 4.1 presents some of the results of the testing performed on the system, which can assist doctors and certified nurses in taking prompt actions in response to the needs and requirements of the patients.

By analysing the data and applying the fuzzy logic rules, the system is able to provide clear and concise notifications to the doctors and certified nurses, allowing them to respond quickly and effectively in emergency situations. The use of Fuzzy Logic ensures that the conclusion drawn from the data is accurate and reliable, helping to improve the overall health and well-being of the paralyzed patients being monitored. The ability to quickly evaluate and respond to the data is crucial in ensuring that the patients receive the proper care and attention they need in a timely manner.

Table 4.1 Fuzzy Logic Result based on tested data

Pulse Rate (BPM)	Temperature (CELCIUS)	Air Quality (API)	condition	Result (%)
56	36.8	60	NORMAL	33.41
60	37	40		20
65	36.5	70		32.7
60	37.8	100	CAUTION	53.09
75	38	70		52.58
54	35.6	190		49.9
80	39.6	180	EMERGENCY	65.57
85	37.5	220		79.97
74	38	270		70.39

The user acceptance test was conducted to 5 users to test the efficiency of the system condition and all the acceptance requirements are listed in the table 4.2 below. The User Acceptance Test Below resulted from average of 5 respondent and most of the respondent choose success as the system comply all the requirement in the table below.

Table 4.2 User Acceptance Test Result

No	Acceptance Requirement	Test Result		Comments
		Success	Fail	
1	All sensor data represented by using graphs	/		
2	All sensor data obtained in real time	/		
3	The user interface of the dashboard is easy to read and eye-catching	/		
4	All the buttons are functioning well	/		
5	User register, login and logout	/		
6	Fuzzy results are displayed	/		
7	Alarm can be triggered	/		

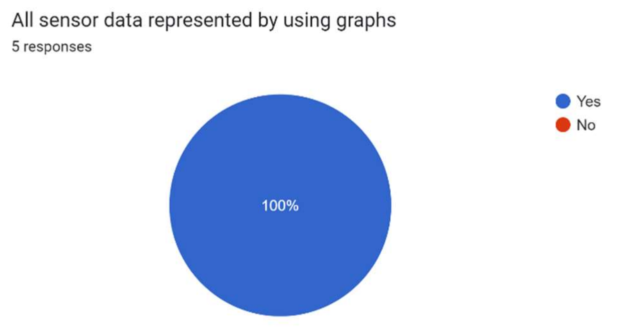


Figure 4.15: User Acceptance Test NO. 1

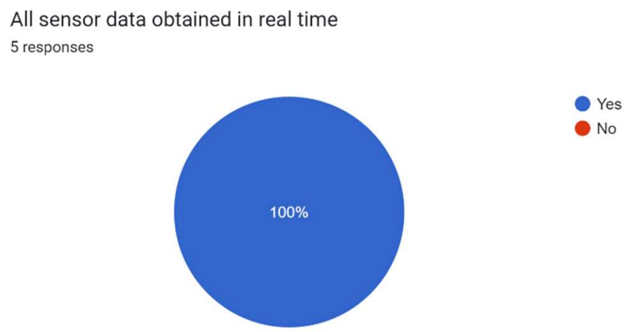


Figure 4.16: User Acceptance Test NO. 2

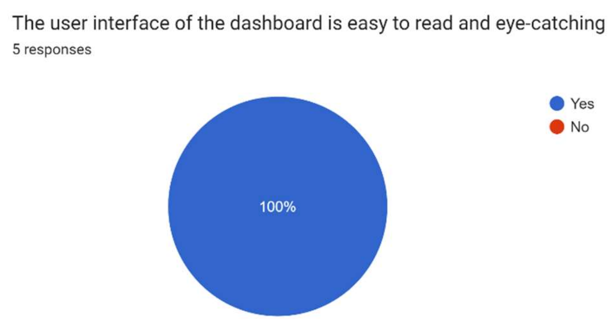


Figure 4.17: User Acceptance Test NO. 3

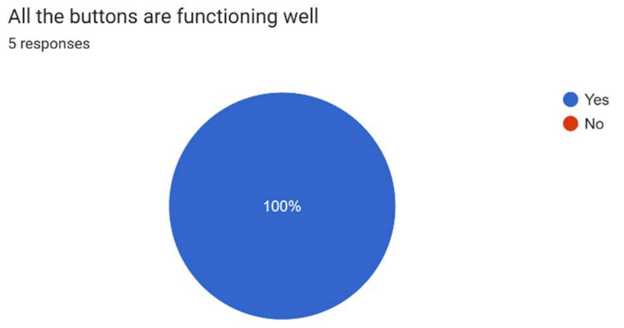


Figure 4.18: User Acceptance Test NO. 4

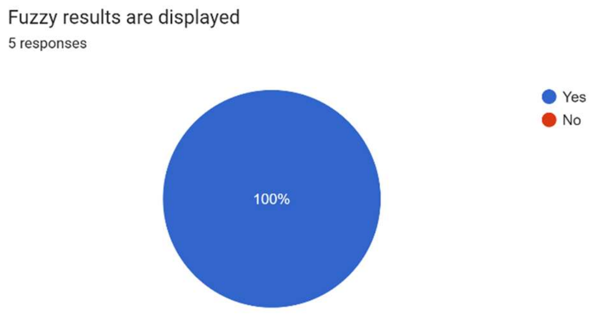


Figure 4.19: User Acceptance Test NO. 5

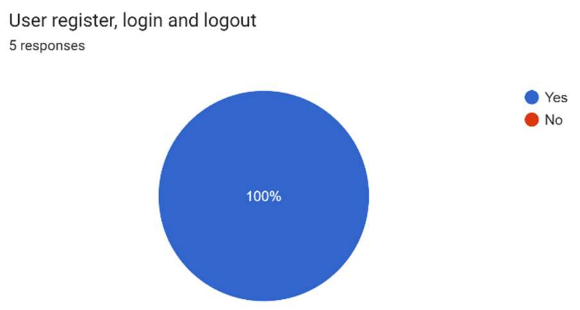


Figure 4.20: User Acceptance Test NO. 6

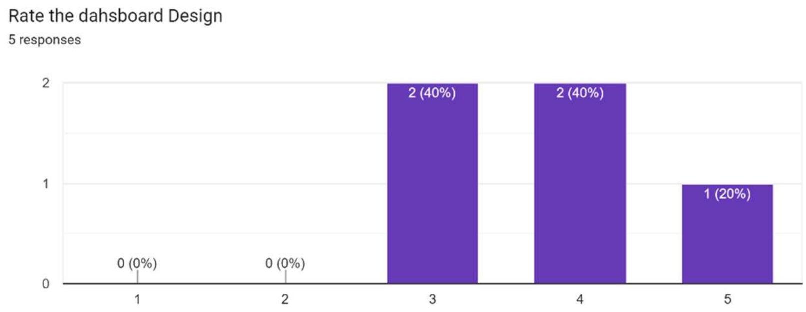


Figure 4.21: User Rating for Dashboard Design

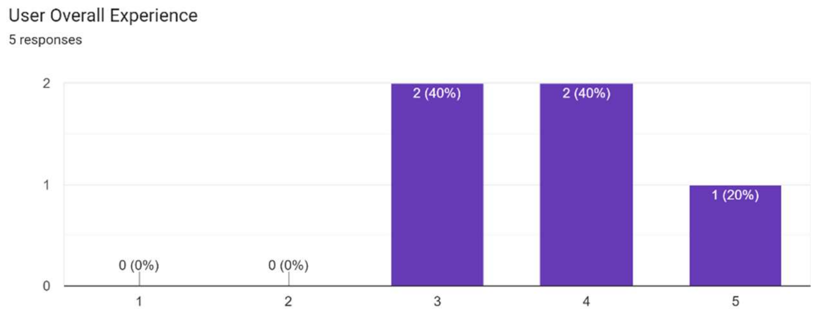


Figure 4.22: User Rating for User Overall Experience

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 5 will provide a summary of the project called the Paralysed Patient Monitoring Kit, which is designed to monitor bedridden patients in their homes. The main objective of this project is to enhance the regular checking method for paralysed patients, which previously required them to go to a hospital or clinic. This previous method made it difficult for paralysed patients to breathe and took a lot of time, as they were unable to move easily due to their condition. The process of taking a paralysed patient to a hospital is also very difficult for the family, as they need to pick up the patient from their bed, put them in a car, and bring them to the hospital. Some families even call doctors to come to their home to do the basic checking, which can be expensive. By implementing and developing the Paralysed Patient Monitoring Kit, patients can do basic monitoring of their body temperature and pulse rate. This data can then be analyzed by doctors and certified nurses, along with air quality data, to assess the patient's current condition. This way, doctors can determine whether the patient is in a normal, cautionary, or emergency condition. Overall, the Paralysed Patient Monitoring Kit is a revolutionary project that can significantly improve the quality of life for bedridden patients, especially those who are paralysed. By allowing them to monitor their own health at home, patients can avoid the hassle and expense of going to hospitals or clinics for regular check-ups. Furthermore, doctors can provide better and more timely care to patients, which can ultimately lead to better health outcomes.

5.2 Project Constraint

During the development process of this project, there were several constraints that needed to be addressed. Despite these constraints, the project was completed successfully with minimal impact. Nevertheless, there is room for improvement in dealing with these constraints in future projects.

Several factors can limit the development process of a project. These may include budgetary constraints, time constraints, technical limitations, or unforeseen obstacles. In the case of this project, several constraints need to be applied to for better get better project result in the future.

- **Limited Budget:** Budgetary constraints can be a significant challenge for any project, and this project was no exception. In particular, the development of this project required a significant budget due to the testing of several sensor failures. This resulted in increased costs for equipment, materials, and personnel. Unfortunately, the budget for this project was not provided by the university, which added an additional layer of difficulty.

- **Time Constraint:** Time constraints can be a significant challenge for any project, and this project was no exception. In particular, the project was developed by an individual and they had only 6 weeks to complete the development of the Paralysed Patient Monitoring Kit, which included the development of the sensor, dashboard, and the report from chapter 1 to chapter 5. This required the individual to work diligently and efficiently to ensure that all components of the project were completed on time.

- **Server Problem:** One of the major challenges faced during the development of the Paralysed Patient Monitoring Kit project was the issues with the server. The project used an MQTT and Web Socket server, but the server was often down and unable to subscribe to data from MQTT or control output using the Web Socket server. This created significant issues during the development and testing phases of the project and resulted in delays and frustration for development and testing process .

However, it is important to acknowledge these constraints and make efforts to address the in future projects. For example, budgetary constraints may be mitigated by better planning and resource allocation by Faculty, while time constraints may be addressed by more efficient project management and long timeline given. Technical limitations can be overcome by exploring alternative solutions or investing in research and development. Unforeseen obstacles may be dealt with through contingency planning and adaptability.

5.3 Future Work

Moving forward, there are several potential upgrades and improvements that could be made to the Paralysed Patient Monitoring Kit (PARAKIT). One potential upgrade is the addition of an OPENCV gauge monitoring system to monitor the quantity of oxygen in the tank. This could help to ensure that patients have a reliable supply of oxygen at all times, and could provide an early warning system for when the tank needs to be refilled.

In addition to the OPENCV gauge monitoring, there are several other potential improvements that could be made to the PARAKIT dashboard. For example, the visualization chart design could be improved to make it more intuitive and user-friendly. The dashboard could also be made more customizable, allowing users to select which data they want to display and how it is presented. Other potential improvements could include adding more advanced analytics and data processing capabilities to help doctors and nurses more easily identify trends and patterns in the patient's condition.

To further enhance the PARAKIT's monitoring capabilities, additional sensors could be added to the kit. One potential addition is an ECG sensor, which could help doctors and nurses to monitor the patient's heart rate and identify any potential cardiovascular issues. Other potential sensors could include a blood pressure sensor, a blood oxygen level sensor, or a humidity and temperature sensor to monitor the patient's environment and ensure that they are comfortable and safe.

Finally, the oxygen tank control has already been implemented, but there are still other potential improvements that could be made to further enhance the system's capabilities. For example, an automatic oxygen tank refill system could be added to help ensure that the tank is always full, without requiring constant monitoring and refilling by a caregiver or nurse. This could help to improve the patient's overall safety and comfort, while also reducing the burden on caregivers and healthcare providers. Additionally, other features such as voice-activated controls or remote monitoring capabilities could be explored, to make the system even more convenient and user-friendly. By continuing to explore and implement these types of upgrades and improvements, the PARAKIT could become an even more valuable tool for monitoring and caring for paralysed patients in their homes.

REFERENCES

[1] Rajan Jeyaraj, P., & Nadar, E. R. S. (2022). Smart-monitor: patient monitoring system for IoT-based healthcare system using deep learning. IETE Journal of Research, 68(2), 1435-1442.

[2] Hameed, K., Bajwa, I. S., Ramzan, S., Anwar, W., & Khan, A. (2020). An intelligent IoT based healthcare system using fuzzy neural networks. Scientific programming, 2020, 1-15 [3] B. Beaglehole, R. T. Mulder, C. M. Frampton, J. M. Boden, N.-H. Giles, and C. J. Bell, "Psychological distress and psychiatric disorder after natural disasters: systematic review and meta-analysis," 2018, doi: 10.1192/bjp.2018.210.

[3] Srikanth, B., Divya, P., Nandini, P., Sabira, S., & Bharathi, T. (2020). Patient Health Monitoring using Arduino through IoT. EPRA International Journal of Research & Development (IJRD), 619-633

[4] Abd Rahman, N. H., Lee, M. H., & Suhartono & Latif, M. T. (2016). Evaluation performance of time series approach for forecasting air pollution index in Johor, Malaysia. *Sains Malaysiana*, *45*(11), 1625-1633.

[5] Labs, D. (2022, July 19). What is a good resting heart rate by age & gender? WHOOP. https://www.whoop.com/thelocker/resting-heart-rate-by-age-and-gender/

[6] Kushimoto, S., Gando, S., Saitoh, D., Mayumi, T., Ogura, H., Fujishima, S., ... & Aikawa, N. (2013). The impact of body temperature abnormalities on the disease severity and outcome in patients with severe sepsis: an analysis from a multicenter, prospective survey of severe sepsis. *Critical care*, *17*, 1-9.

[7] Becker, K., Thull, B., Käsmacher-Leidinger, H., Stemmer, J., Rau, G., Kalff, G., & Zimmermann, H. J. (1997). Design and validation of an intelligent patient monitoring and alarm system based on a fuzzy logic process model. Artificial intelligence in medicine, 11(1), 33-53.

[8] M. A. Prada, P. Reguera, S. Alonso, A. Morán, J. J. Fuertes, and M. Domínguez, "Communication with resource-constrained devices through MQTT for control education," *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 150–155, Jan. 2016, doi: 10.1016/J.IFACOL.2016.07.169.

[9] Stauffer, M. (2019). *Laravel: Up & running: A framework for building modern php apps*. O'Reilly Media.

[10] Vyas, S., Gupta, S., Bhargava, D., & Boddu, R. (2022). Fuzzy Logic System Implementation on the Performance Parameters of Health Data Management Frameworks. *Journal of Healthcare Engineering*, *2022*.