

Monitoring of Peripheral Vascular Disease Using Neuro-fuzzy Algorithm and Wireless Body Sensor Network

by

**Jhon Paul E. Jaspe
River Arliss M. Lontok**

A Thesis Report Submitted to the School of Electrical, Electronics, and
Computer Engineering in Partial Fulfillment of the Requirements for the Degree

Bachelor of Science in Computer Engineering

Mapúa University
April 2019

TABLE OF CONTENTS

TITLE PAGE	i
TABLE OF CONTENTS.....	ii
LIST OF TABLES	iii
LIST OF FIGURES	iv
Chapter I: INTRODUCTION	1
Chapter II: REVIEW OF RELATED LITERATURE.....	4
Peripheral Vascular Disease	4
Blood Pressure.....	7
Wireless Body Sensors.....	8
Neuro-fuzzy Systems	10
Chapter III: METHODOLOGY	12
Methodology Process	12
Conceptual Framework	15
System Flowchart.....	17
Block Diagrams.....	21
Hardware Development.....	25
Linear Regression Model	39
Squared Error Cost Function.....	40
Gradient Descent	41
Min-Max Normalization	41
Mean Arterial Pressure.....	44
Ankle-Brachial Index	45
Neuro Fuzzy System	45

ANFIS Architecture	45
MAP Linguistic Labels	47
Rule Base	48
ANFIS Learning Algorithm	49
Error Measure.....	49
Least Square Estimate.....	51
Software Development.....	52
Statistical Treatment.....	78
REFERENCES	81

LIST OF TABLES

Table 2.1 Characteristics of Patients Involved in Study of Pleo et. al.....	4
Table 3.1 Sample training dataset for oxygen saturation regression model	29
Table 3.2 Comparison of estimated oxygen saturation with commercially calibrated.....	29
Table 3.3 Sample dataset for the arm blood pressure regression model.....	36
Table 3.4 Sample dataset for the leg blood pressure regression model.....	37
Table 3.5 Estimated and actual arm blood pressure readings.....	38
Table 3.6 Estimated and actual leg blood pressure readings	38
Table 3.7 Comparison of learning rates based from number of iterations and cost function ..	43
Table 3.8 Simplified Blood Pressure range interpretation.....	45
Table 3.9 Mean Arterial Pressure Linguistic Label.....	48
Table 3.10 Recorded System Readings of the Device with Timestamps	78
Table 3.11 Single-sample t-test Evaluation Table	79
Table 3.12 Table for Pearson R test of PTT and Systolic Blood Pressure	80
Table 3.13 Table for Pearson R test of RPDPT and Diastolic Blood Pressure	80

LIST OF FIGURES

Figure 2.1 Construction of the wireless impedance implant used to monitor PVD	5
Figure 2.2 Measurement Scheme for Impedance Plethysmography and Spectroscopy	6
Figure 2.3 Analysis of Microstimulator Used for Possible PVD Treatment Method	6
Figure 2.4 Graphical Representation of the Ankle-Brachial Index Procedure	7
Figure 2.5 Sample Wireless Body Area Network System for Health Monitoring	8
Figure 2.6 Fuzzy Inference System Architecture	9
Figure 2.7 Diagram of Simple Neuro-fuzzy System	10
Figure 3.1 Methodology Process	12
Figure 3.2 Conceptual Framework	15
Figure 3.3.1 System Flowchart	17
Figure 3.3.2 HR and BP Calculation and Oxygen Saturation Calculation Flowcharts	18
Figure 3.3.3 Determine Health Status Flowchart.....	19
Figure 3.3.4 Determine PVD Status Flowchart	20
Figure 3.4.1 Sensor Node Block Diagram.....	21
Figure 3.4.2 System Block Diagram.....	22
Figure 3.5 Schematic of ECG sensor node	25
Figure 3.6 Gain to R_G resistance values	26
Figure 3.7 ECG electrode placement on the human body	27
Figure 3.8 LED driver circuit for red and IR LEDs	29
Figure 3.9 OP905 photodiode relative response vs. wavelength graph	30
Figure 3.10 OP905 photodiode receiver circuit	30
Figure 3.11 Schematic of PPG sensor node circuit.....	31
Figure 3.12 Full Schematic Diagram of the System	33

Figure 3.13 (McCarthy, Mathewson, & O'Flynn, 2011) Derivation of PTT from EGC and PPG signals	34
Figure 3.14 (Elgendi, 2012) (a) Original PPG signal (b) First derivative of a PPG	35
Figure 3.15 Regression Model Generation Flowchart.....	42
Figure 3.16 ANFIS structure with two inputs, one output, and eight rules	46
Figure 3.17 Heart Rate and Blood Pressure Pages for GUI Mock-up	52
Figure 3.18 Oxygen Saturation and PVD Status Pages for GUI Mock-up.....	53
Figure 3.19 Entity Relationship Diagram	54
Figure 3.20 Admin Use Case Diagram	55
Figure 3.21 Doctor Use Case Diagram	56
Figure 3.22 Patient Use Case Diagram	50
Figure 3.23 Mobile Application Login Page Flowchart	58
Figure 3.24 User Logout Flowchart	59
Figure 3.25 Admin Login Flowchart	60
Figure 3.26 Admin Homepage Flowchart	61
Figure 3.27 Add Doctor Flowchart	62
Figure 3.28 Remove Doctor Flowchart	63
Figure 3.29 View Doctors Flowchart	64
Figure 3.30 Doctor Homepage Flowchart	65
Figure 3.31 Update Account Login Flowchart	66
Figure 3.32 Register Patient Flowchart	67
Figure 3.33 View Patients List Flowchart	68
Figure 3.34 Add Patient Flowchart	69
Figure 3.35 Remove Patient Flowchart	70
Figure 3.36 View Patient Flowchart	71

Figure 3.37 View Patient Profile Flowchart	72
Figure 3.38 Patient Registration Completion Flowchart	73
Figure 3.39 Patient Homepage Flowchart	74
Figure 3.40 View Doctor Flowchart	75
Figure 3.41 Monitor Patient Flowchart	76

Chapter I

INTRODUCTION

Peripheral vascular disease (PWD), also known as peripheral artery disease, is a cardiovascular disorder that affects arteries outside of the brain or heart. If left untreated, PVD can cause constant leg pain, gangrene, and increased risk from a stroke. PVD commonly results from complications in diabetes or atherosclerosis, although other factors such as smoking, obesity, and advanced age can result in increased risk of developing the disease. In the Philippines, PVD affects around 5% of Filipinos aged 40 years and older, and most patients diagnosed with PVD were found to suffer from other medical complications such as hypertension (American Heart Association, 2018).

In recent years, research on PVD has primarily focused on disease diagnosis and classification. For instance, a study conducted in India's Karunya University used the toe brachial index to determine whether the patient is suffering from a specific type of peripheral cardiovascular disease (Manimegalai, Augustine, & Thanushkodi, 2013). Another study focused on using endoscopic techniques to monitor a patient's peripheral vascular condition (Hirano, et al., 2013). While there are studies that focused on real-time monitoring of cardiovascular conditions, these systems focused on monitoring the heart itself (Wannenburg & Malekian, 2015).

Previous studies showed that while there have been researches regarding detection of PVD, monitoring of the disease after diagnosis has been lacking. Furthermore, recent studies regarding PVD make use of invasive or non-portable methods (e.g. endoscopy, arteriography), making them impractical for monitoring purposes. Parameters such as electrocardiogram

(ECG) signals and real-time brachial index were also not considered in previous studies on Peripheral Vascular Disease. The lack of real-time monitoring for PVD makes it difficult to check if the disease is being mitigated, especially considering that advanced PVD can lead to further health complications like heart failure.

This study aims to develop a real-time monitoring system that can be used to monitor the status of a person who is suffering from Peripheral Vascular Disease. This study specifically aims to (1) develop a system to monitor a patient's blood pressure, heart rate, and internal cardiovascular activity using blood pressure, photoplethysmography (PPG), and ECG sensors; (2) train a neuro-fuzzy algorithm to determine the overall health status of the patient and integrate it into the developed system; (3) test and verify the accuracy of the developed system; and (4) design and integrate a mobile application that can be used by the patient and his assigned doctor to view the gathered data.

The benefit of the study is that people suffering from PVD will be able to regularly check on the progress of their disease, even without hospital appointments. Furthermore, since PVD, if left unchecked, can cause heart failure or stroke, the study can prove beneficial to providing warning signs of further complications that can arise from PVD.

It should be noted that this study will primarily focus on the monitoring of PVD. While other cardiovascular diseases can be monitored using the same parameters that will be used for the proposed system, application of the system for other heart diseases may result in less accurate results than expected. Furthermore, the study assumes that all participants in the testing are suffering from PVD. The system is also designed to transmit data to the server every 30 seconds.

Chapter II

Review of Related Literature

Peripheral Vascular Disease

Peripheral Vascular Disease (PWD), also known as Peripheral Arterial Disease (PAD), is a disease that results in the narrowing of the arteries that supply blood to the peripheral regions of the body (e.g. arms, legs). It is caused by cholesterol build-up in the affected regions of the cardiovascular system, a condition also known as atherosclerosis. PVD usually results as a complication from diabetes, although smokers and the obese are also at high risk of developing this disease. Left untreated, PVD can cause constant leg pain, erectile dysfunction, gangrene, and increased risk of suffering a stroke (American Heart Association, 2018).

According to a 2012 study, PVD is a serious and prevalent disease that affects 3-10% of the global population. In the Philippines, it is estimated that 5% of Filipinos aged 40 years and older suffer from PVD. It was also determined that factors such as smoking, diabetes, hypertension, and renal insufficiency contributed to an increased risk of contracting the disease (Pleo, Cresencio, & Alajar, 2012).

	N (70)	%
Age in Years	60 + 12.2	
Males	18	26
Females	52	74
Hypertension	44	63
Average BP	Systolic (SD) 140+18.32	Diastolic (SD) 82+12.39
(+) PAD	Resting 16	6MWT 15
Smoking	10	(36)(34)
(+) PAD	Resting 5	6MWT 4
		14 (50)(40)

Table 2.1 Characteristics of Patients Involved in Study of Pleo et. al.

Due to a focus on more well-known cardiovascular disorders such as stroke and heart failure, there are not many studies that cover techniques on how to diagnose and monitor PVD. A study that did focus on PVD made use of the patient's peripheral blood flow as a way to monitor the progress of the disease (Shabanivaraki, Breen, & Gargiulo, 2015). Another study monitored PVD using wireless impedances recorded by implanted microdevices that were powered by ultrasound energy (Celinskis & Towe, 2014).

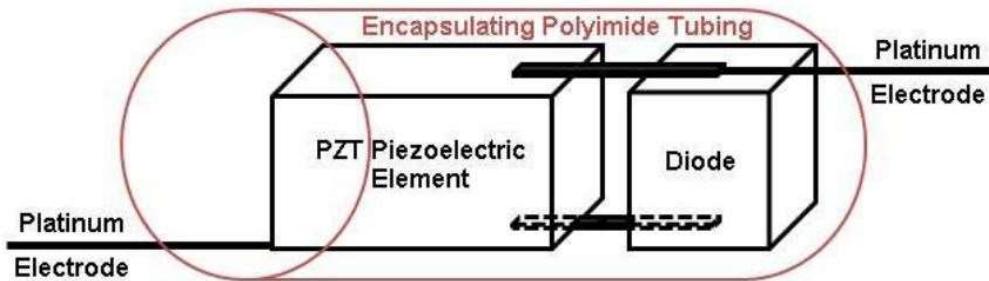


Figure 2.1 Construction of the wireless impedance implant used to monitor PVD

However, in recent years, novel methods of diagnosing and monitoring the progress of PVD in a patient's body have been devised as awareness of the severity and prevalence of the disease increased. For instance, photoplethysmography has been suggested as a way to monitor PVD in conjunction with a toe brachial index (Manimegalai, Augustine, & Thanushkodi, 2013). Combined impedance plethysmography and spectroscopy has also been used to diagnose various peripheral cardiovascular diseases, especially those affecting the lower extremities of the body (Pittella, Pisa, Piuzzi, Rizzuto, & Del Prete, 2017).

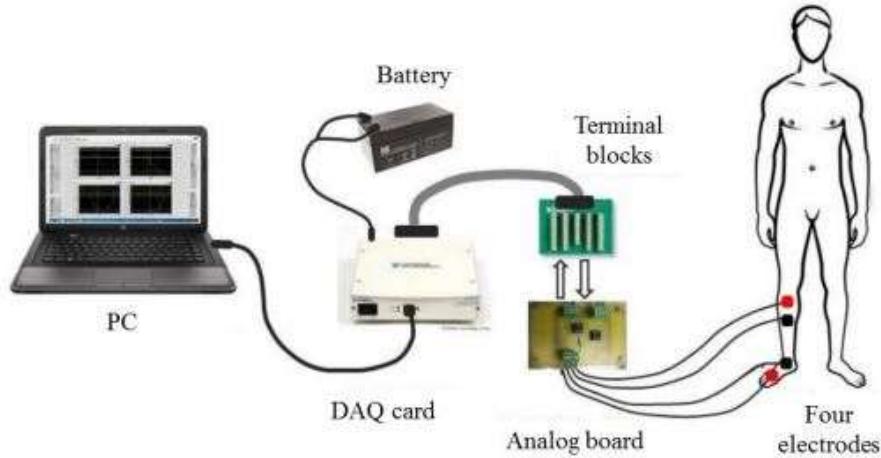


Figure 2.2 Measurement Scheme for Impedance Plethysmography and Spectroscopy

While there have been some studies that focused on treating PVD, it should be noted that the techniques used in these researches are as of writing untested on humans. A 2013 study went into detail on the use of microstimulators to stimulate the sciatic nerve, which can serve as an alternate method of treating PVD. However, the study tested the procedure on rats, and as a result, its actual effectiveness on humans is still unknown (Towe, Gruber, Gulick, & Herman, 2013).

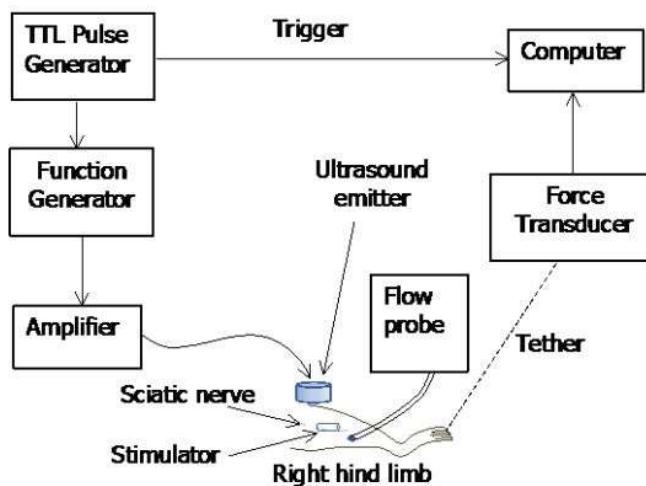


Figure 2.3 Analysis of Microstimulator Used for Possible PVD Treatment Method

Blood Pressure

One of the leading risk factors when it comes to cardiovascular diseases is blood pressure abnormality. According to a 2014 study, the burden caused by persistent high blood pressure, more commonly known as hypertension, is still substantial despite advancements in modern treatments (Rapsomaniki, et al., 2014). As such, blood pressure reading is one of the most common ways used to diagnose possible cardiovascular diseases.

One of the most commonly-used non-invasive methods of determining blood pressure is photoplethysmography (PPG). PPG makes use of optical sensors to measure changes in blood volume, which can be used to determine a patient's Pulse Transmit Time (PTT) and heart rate. Due to the strong correlation between PPT and blood pressure, the former value can be used as a basis in determining the latter (Noche, Villaverde, & Lazaro, 2017).

For Peripheral Vascular Disease, the leading method of diagnosis is the ankle brachial index (ABI). ABI works by getting the ratio of the systolic blood pressure readings of the patient's ankle and upper arm. The resulting ratio is then compared with a scale to determine if the patient is suffering from PVD or not. The main reason ABI is the preferred diagnostic method when it comes to PVD is because apart from being non-invasive, this method is more cost-effective compared to other diagnostic tools, and thus could be more routinely determined in a large variety of patients (Rac-Albu, Iliuta, Guberna, & Sinescu, 2014).

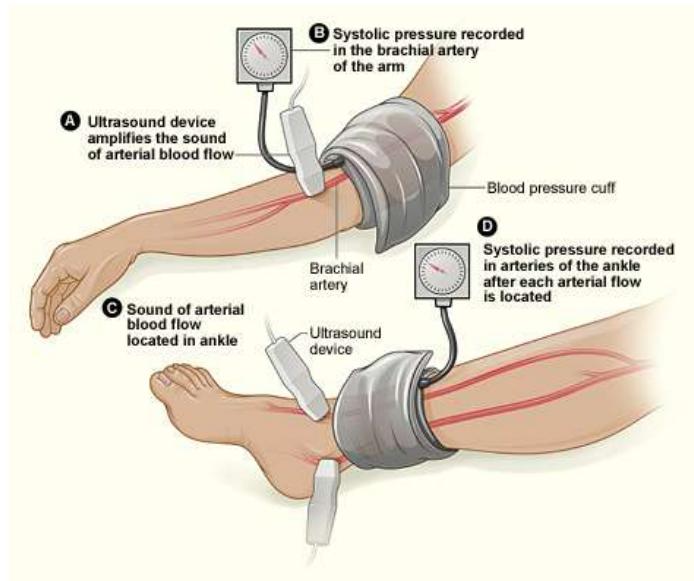


Figure 2.4 Graphical Representation of the Ankle-Brachial Index Procedure

Normal ABI ratios range from 0.9 to 1.30. If the patient's calculated ratio goes above 1.30, then it is likely that the patient is suffering from arterial calcification. On the other hand, if the ratio falls below 0.9, then the patient is likely suffering from Peripheral Vascular Disease (Infectious Diseases Society of America, 2012).

ABI ^a	Interpretation
>1.30	Poorly compressible vessels, arterial calcification
0.90–1.30	Normal
0.60–0.89	Mild arterial obstruction
0.40–0.59	Moderate obstruction
<0.40	Severe obstruction

Table 2.2 Interpretation Table for Ankle Brachial Index Ratios

Wireless Body Sensors

In recent years, wireless sensor networks have become commonplace in various scientific and industrial applications, usually in the form of real-time detection or early warning systems. For instance, wireless sensor networks can be used to monitor pest outbreaks in certain types of crops using a combination of strategically-placed sensors and image processing equipment that make use of Haar feature-based cascade classifiers for pest detection (Mistal, Villaverde, & Hadi, 2018). However, wireless sensor networks can also be placed on the body to conduct real-time monitoring at a more personal level.

Wireless body sensor networks (WBSN) are networks of wearable devices that have attached sensors which can monitor a human's physiological state. With modern technology, the creation of tiny biomedical devices which can be placed inside the body or on its surface have become a reality. Common physiological signals obtained from the human body are blood pressure, blood flow, ECG, and EEG (Yuce, 2013).

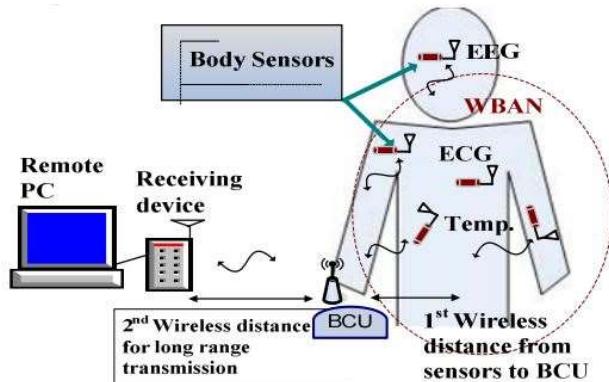


Figure 2.5 Sample Wireless Body Area Network System for Health Monitoring

Nowadays, WBSN are commonly used in medical and healthcare applications to facilitate real-time monitoring of various physiological conditions. As a result, WBSN has found commonplace use in patient supervision and disease monitoring systems (Fourati, 2014). The data gathered by WBSNs can also be sent to patients and doctors in real-time through IoT

integration – for instance, a smartphone app that can transmit and verify patient information to doctors will serve to benefit medical professionals by allowing them to view necessary data on the go (Samonte, Mullen, Banaga, Cortes, & Dela Calzada, 2018).

WSBN can be used to perform real-time monitoring of factors such as heart rate, blood pressure, and body temperature. Coupled with a fuzzy inference system, these variables can be used to determine the overall health status of a patient. This is useful in cases wherein healthcare personnel are monitoring the progress of non-communicable diseases, such as cardiovascular and pulmonary diseases (Billones, Vicmudo, & Dadios, 2015).

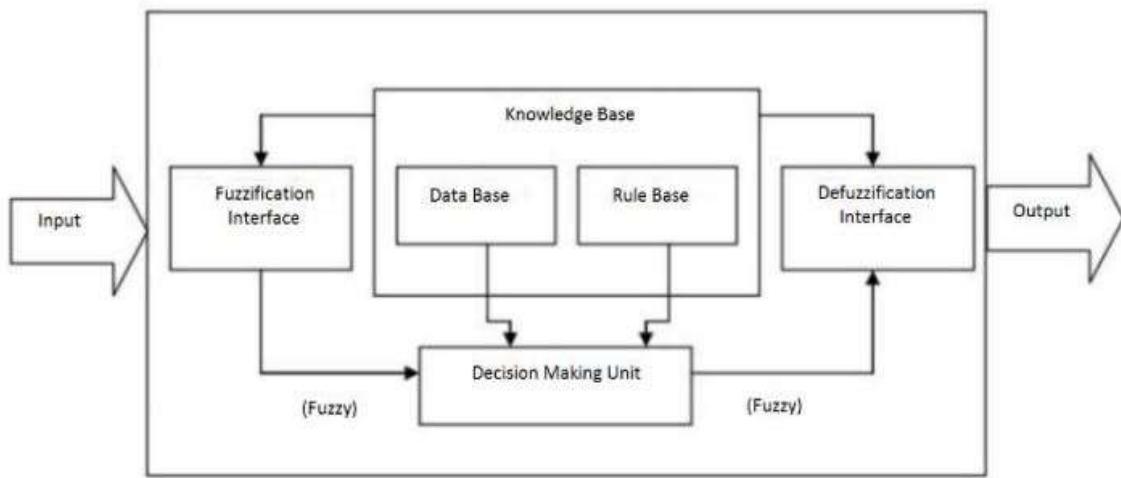


Figure 2.6 Fuzzy Inference System Architecture

For peripheral vascular disease, only one study made use of wireless body sensor networks in the past five years, and this study involved the use of wireless impedances to monitor the disease's progression (Celinskis & Towe, 2014). However, other related cardiovascular diseases have made extensive use of WSBN for monitoring purposes. For instance, concepts related to Internet of Things (IoT) were used to create wearable graphene oxide-nylon sensors that functioned as electrocardiograms. By having people suffering from cardiovascular diseases wear these devices, the mortality rate of such diseases can potentially

decrease. This is because medical personnel are now able to gain real-time information concerning the status of the patient's heart and related organs for an extended period of time, giving them the information they need to devise better treatments (Hallfors, Jaoude, Ismail, & Isakovic, 2017).

Another application of WBSN related to the cardiovascular system involved the creation of a wireless monitoring platform used for early detection and treatment of hypertension. This device, which is worn on the chest area, made use of electrocardiogram (ECG), photoplethysmography (PPG), and ballistocardiogram (BCG) signals to determine if the patient is suffering from hypertension (Janjua, Guldenring, Finlay, & McLaughlin, 2017).

Neuro-fuzzy Systems

Neuro-fuzzy systems (NFS) are systems that combine fuzzy logic with artificial neural networks. These systems are widely used in the field of artificial intelligence with various applications in medical systems, economic systems, traffic control, and image processing (Kar, Das, & Ghosh, 2014). Fuzzy logic and neural networks are core components in devices that make use of neuro-fuzzy systems. The learning capabilities of an artificial neural network, together with the human-like reasoning of fuzzy logic systems improve the inferences made by a machine. The combination of these techniques serves to create robust systems that can efficiently solve practical problems (Shihabudheen & Pillai, 2018).

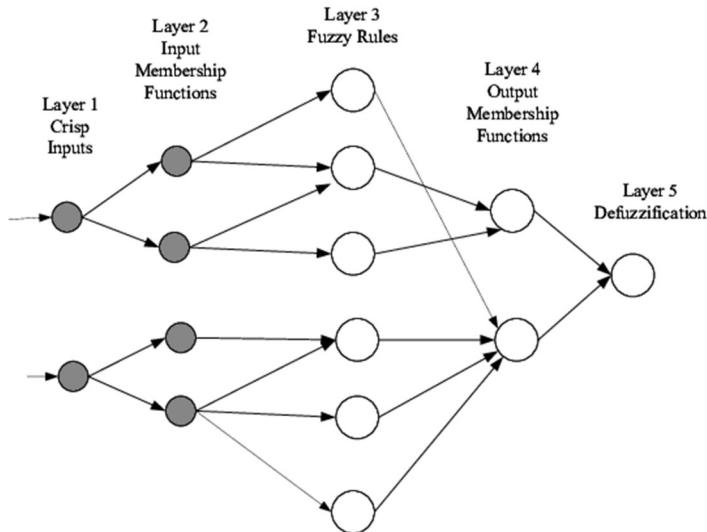


Figure 2.7 Diagram of Simple Neuro-fuzzy System

In conventional Boolean logic, a value can only be either 0 (false) or 1 (true). Fuzzy logic is an extension of Boolean logic which allow handling of values in between 0 (false) and 1 (true). This makes it possible for a value to be partially true or false. It does so by providing flexible approximations which allow interpretations of if-then rules in its decision-making. Fuzzy logic is widely used in medical applications where decision-making is uncertain (Morsi & El Gawad, 2013).

In recent years, neural networks have become more widely-used for classification purposes. For instance, a medicinal application of this concept involves the use of a convolutional neural network to accurately count and classify white blood cells found in the body. This counting and classification process is then used to identify the type of treatment best-suited for the patient, making such a method useful for those suffering from diseases like anemia, leukemia, and Acquired Immunodeficiency Syndrome (AIDS) (Macawile, Quiñones, Ballado Jr., Cruz, & Caya, 2018).

Chapter III

Research Methodology

Methodology Process

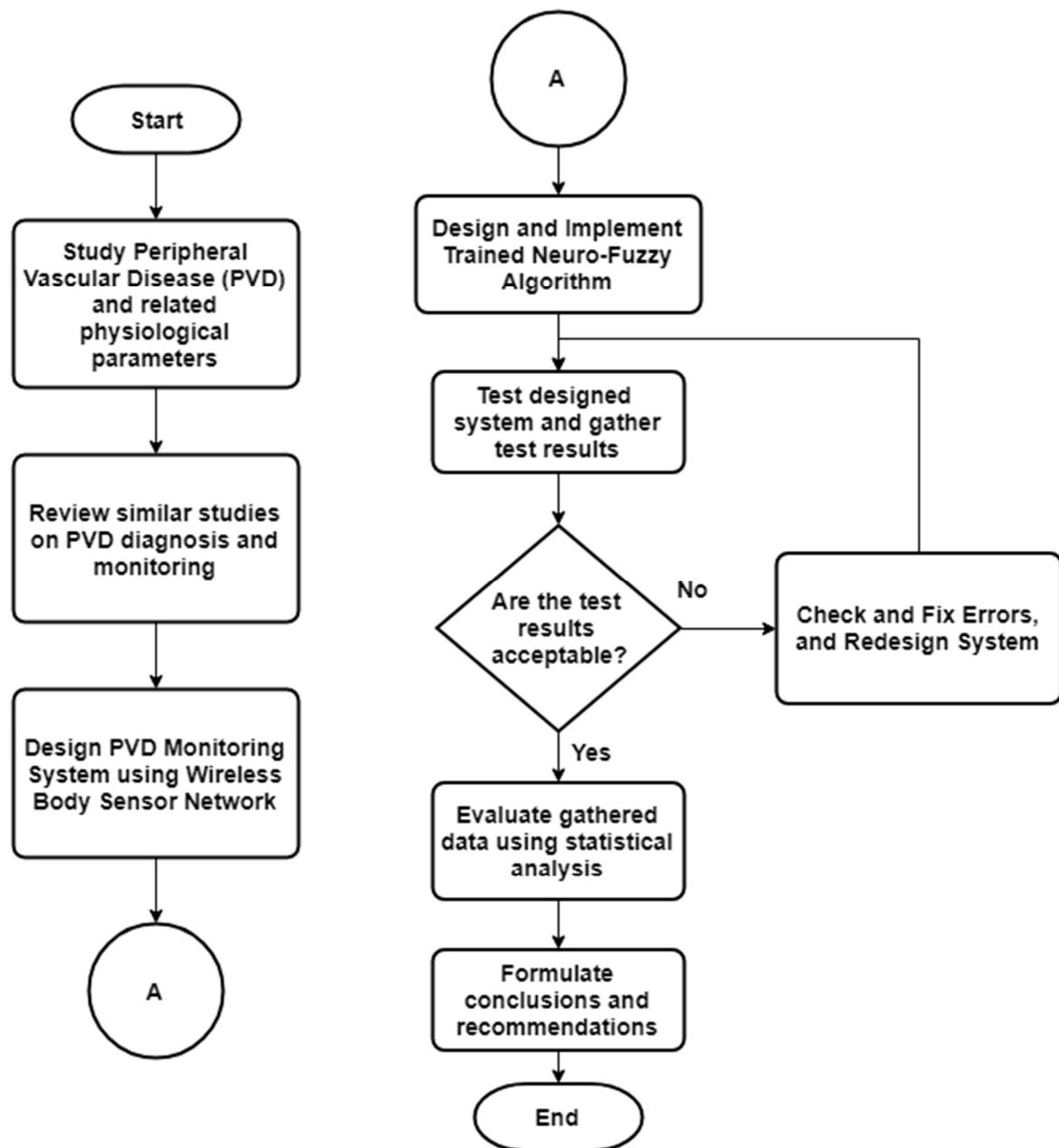


Figure 3.1 Methodology Process

Figure 3.1 depicts the processes and actions needed in conducting the research. The initial procedure to be done must be research on the causes, symptoms, and related factors concerning Peripheral Vascular Disease. The causes, risks, symptoms, diagnosis, and treatments for PVD must be identified so that the proponents can propose a solution relevant to the problem at hand. Previous studies regarding PVD shall also be used as resources in formulating the study. Ethical values which may affect the research will also be kept in consideration. The proposed monitoring system will implement a wireless body sensor network in measuring and transmitting a patient's physiological state, thus previous studies which used this technology for medical monitoring applications must be analyzed.

In monitoring a patient's physiological state, measured crisp values are to be fuzzified to make a more detailed interpretation of a patient's status. For the determination of a subject's risk of acquiring PVD, as well as producing early warning signals, a neuro-fuzzy system is to be utilized. As such, it is necessary to analyze similar systems that were implemented in the healthcare industry.

Physiological parameters such as heart rate, blood pressure, blood volume pulse, blood oxygen saturation, and cholesterol level are measurable quantities that are to be used in monitoring a patient's health and determining their PVD risk status. The system will incorporate wireless sensors that will provide all the measurements required by the application. Electrocardiography (ECG) is a process that allows for monitoring of the heart's electrical activity with electrodes placed on the skin will be used to determine the heart rate of a patient. Photoplethysmography (PPG), which is a process that uses optical measurements in determining the volume of blood within the arteries, will be used to measure blood pressure as well as oxygen saturation level.

Hardware development of the system shall be divided into the following: development of sensor interfacing circuits, implementation of a wireless network protocol for data transmission, microcontroller interfacing, and power supply for each independent circuit. Each sensor must be tested to determine the validity of its produced values. All necessary computations in acquiring the required parameters shall be done by the microcontroller, after which it will be transmitted wirelessly to a centralized web server where the collection of data occurs. A portion of the software must be developed to allow communication with the server. A back-end database system shall be used for persistent storage of all acquired data. Testing of the hardware components and its communication with the server shall be done to detect any errors and resolve them early on during development.

A significant portion of this study is the data, since it will be the basis for all interpretations as well as the training set to be used for the neuro-fuzzy network. As such, data collection from subjects will be done after completing the system hardware. A variety of subjects will be used for this study and consists of: PVD-diagnosed patients, smokers, diabetics, obese patients, and those that have none of the aforementioned conditions. This is so that the neuro-fuzzy network will have a better generalization for determining the physiological state and overall health status of the patient.

In constructing the fuzzy inference system for this study, the initial step to be taken is the creation of the knowledge base, which is a set of IF-THEN rules that is formulated using expert knowledge. To determine the fuzzy set, membership functions are used to associate its elements to values ranging from 0 to 1. One of the most commonly-used membership functions (MF) is the trapezoidal MF, due to its simplicity and computational efficiency for near real-time applications. The fuzzy inference system will implement the Mamdani model, along with

a fuzzification method using the trapezoidal membership function and a centroid defuzzification method. Testing on the fuzzy inference engine is to be done once developed, to determine any errors and provide possible solutions.

The fuzzy inference engine is to be used along with an artificial neural network, to create the neuro-fuzzy system. The collected sensor data stored in the database is to be used as a training set for the neuro-fuzzy model. A computer will first be used to train the model, after which prototype testing will occur. If the results produced are acceptable, it will be converted to a mobile version for deployment in the mobile monitoring application.

A mobile application will be used for patient monitoring. The trained neuro-fuzzy model will be included in the application to provide the outputs needed for the monitoring of the patient's PVD health status. Other measured physiological parameters will also be displayed in the mobile application to allow the patient and healthcare personnel to view relevant health information about the patient. Testing of the mobile application would include comparison between neuro-fuzzy outputs from the computer version and the mobile version to ensure no errors occurred during the conversion process.

Conceptual Framework

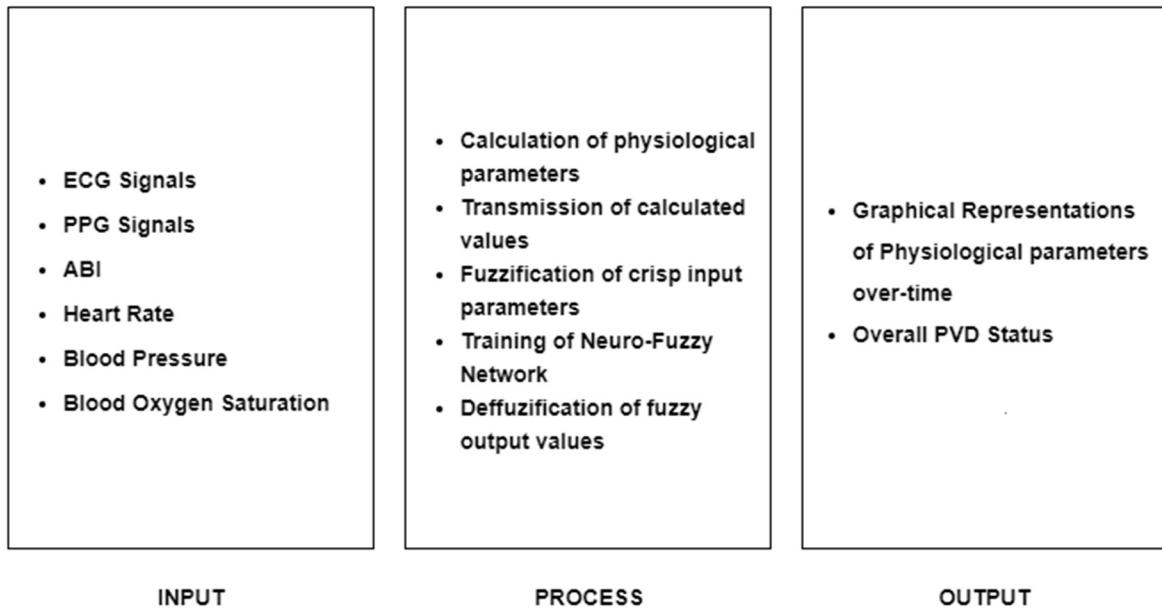


Figure 3.2 Conceptual Framework

Figure 3.2 depicts the overall conceptual framework of the proposed system. The proposed monitoring system generally will make use of sensor readings as its primary inputs. These signals will be used to calculate the physiological parameters relevant to the study.

All obtained values are to be wirelessly transmitted to a central server where the data will be stored. The neuro-fuzzy system will have access to this server to get the data required for training. Before the neuro-fuzzy system is trained, the values must be fuzzified to produce a fuzzy set using the fuzzy inference system. The fuzzy set will be fed to the neuro-fuzzy model for training, and once trained it will be tested to determine if the output is acceptable. The output shall undergo defuzzification which will be used to generate interpretation for the given inputs.

The monitoring system will have a mobile application where the patient or user may view the output generated by the neuro-fuzzy system. The patient can also use the app to

monitor different physiological values such as the heart rate, blood pressure, and blood oxygen saturation of a patient. User-friendly graphical representations of these parameters shall also be displayed for the user to track their overall health status.

System Flowchart

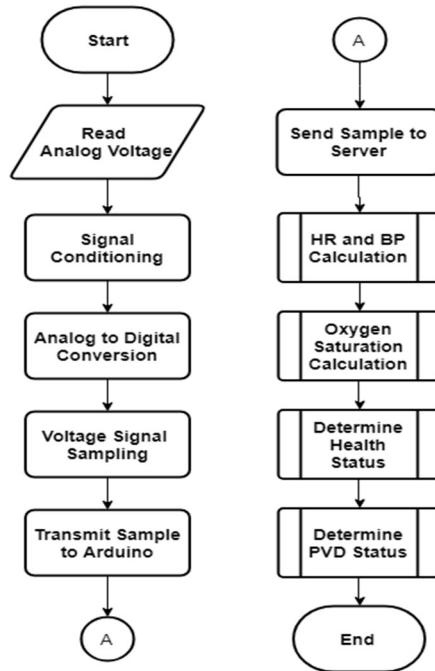


Figure 3.3.1 System Flowchart

The proposed system will rely on the inputs received from the ECG and PPG sensors. These inputs are in terms of voltage and are to be conditioned using a signal conditioning circuit. The signal will be amplified enough to be read by the ADC and its noise will be reduced. Sampling of the signal will be done by setting a time in the master Arduino microcontroller that tells the sensor nodes when to start, stop, and send the voltage readings to the Arduino. The Arduino will be responsible for collecting data from the sensor nodes and sending it to the server. Calculation of the required parameters will be done in the server side. For monitoring these parameters, a mobile application will be utilized. The data calculated from the server will be shown through the mobile application.

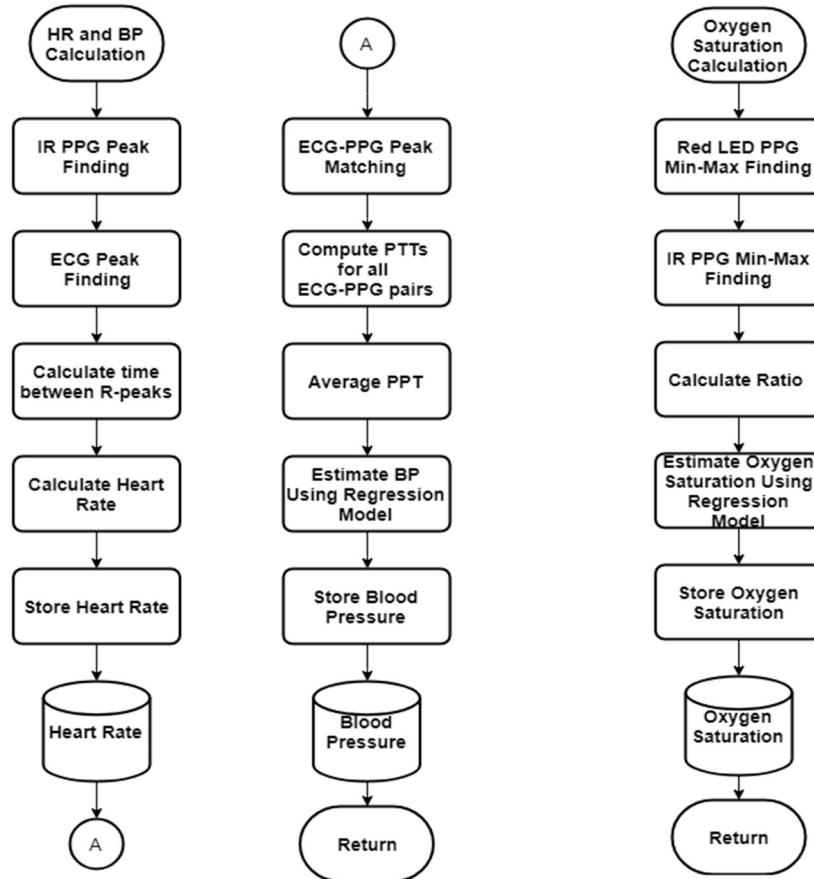


Figure 3.3.2 HR and BP Calculation and Oxygen Saturation Calculation Flowcharts

The figures above show the procedures for calculating the heart rate, blood pressure, and oxygen saturation. The peaks of the ECG signal will be determined, and it is corresponding to the R-peaks for which the time difference between the R-peaks can be used to estimate the heart rate of a person. For the blood pressure, the IR PPG signal and the ECG signal are needed. Their peaks must be determined, and each ECG signal must be matched with a PPG signal, given the sampling period set from the microcontroller. The time difference between the PPG peak and ECG peak is the PTT, all the pairs' PTTs can be computed, and the PTTs are to be averaged. Using the generated regression model, and the average PTT as the input to the model, the blood pressure can be estimated.

For the oxygen saturation, the IR PPG and the red LED PPG are needed. The minimum and maximum peaks must be determined, and the ratio can be calculated using the ratio of the AC and DC components of the IR and red LED PPG signals. A regression model is to be used for estimating the oxygen saturation. All calculated parameters are to be saved to the database for later use.

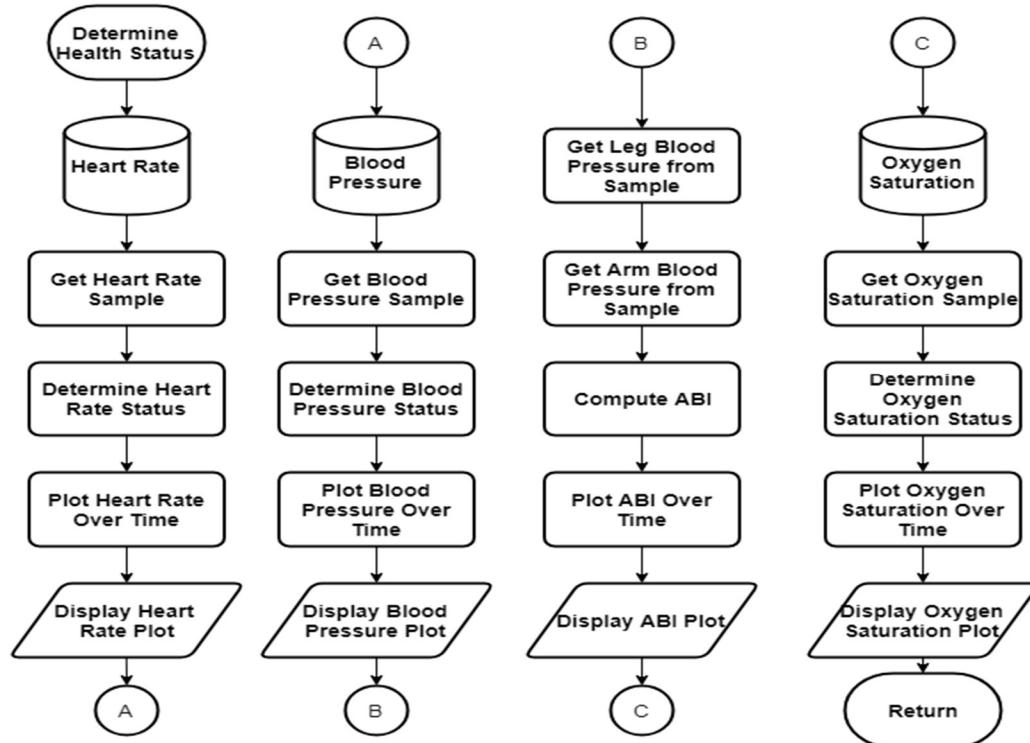


Figure 3.3.3 Determine Health Status Flowchart

For determining the health status, the sample of the calculated parameters stored in the database is needed. These samples are to be fed to the fuzzy part of the neuro-fuzzy system, to determine the statuses of these parameters. Graphical representations of these parameters are also to be displayed in the mobile application.

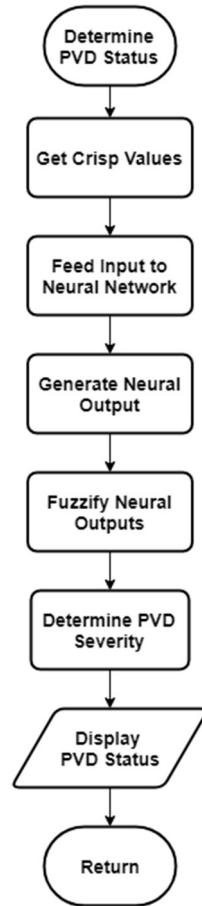


Figure 3.3.4 Determine PVD Status Flowchart

For determining the PVD status of the patient, the crisp values which are the calculated parameters stored in the database are needed. The neural network will be responsible for determining the value to be fed in the fuzzy system which would output the severity of the PVD status. The PVD severity will be based from the interpretation of the ABI ratios wherein the obstruction present within the arteries are ranged from normal to severe, but instead of using only the ABI, the system will utilize the heart rate and oxygen saturation for determining the severity.

Block Diagrams

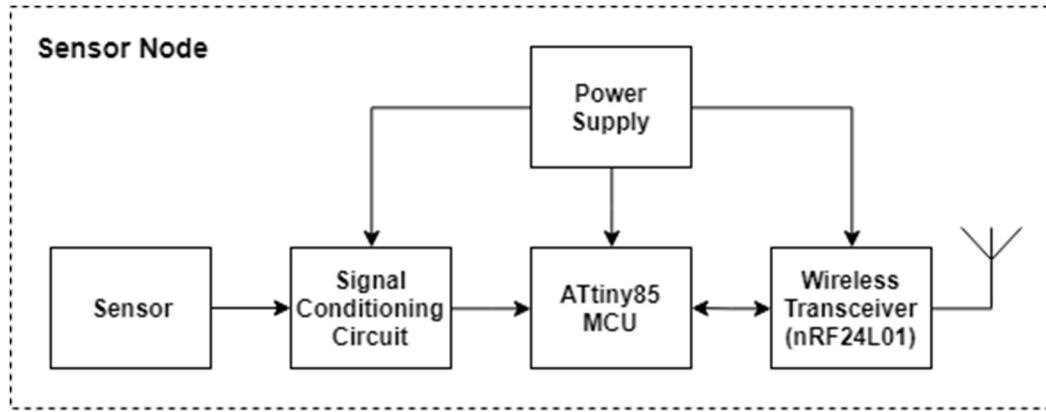


Figure 3.4.1 Sensor Node Block Diagram

The proposed system will implement a wireless body sensor network. Each physiological sensor will be buffered to a signal conditioning circuit that will amplify and reduce the noise of the sensor's analog signal. Depending on the sensor, the signal conditioning circuit may be different to address specific needs. The conditioned signal will be converted to its digital form using the microcontroller's ADC converter. The proponents chose the ATtiny85 as the microcontroller primarily because of its portability, as well as providing enough computing power for the application. For each sensor, the required parameter values will be computed and formatted by the microcontroller to send wirelessly to a master microcontroller where all sensor values are collected. The nRF24L01 was chosen because of its low cost and wireless capabilities that is sufficient for the application. It can also handle up to six addressable pipes, which means that six sensor nodes may be connected to the nRF24L01 wireless transceiver on the master-side. As for the power supply, a coin cell battery is to be used due to its small size and is enough to power a sensor node.

The sensor nodes are responsible for obtaining the ECG and PPG signals, they must communicate with the Arduino to determine when to start and stop acquiring samples. The sensor nodes will receive a ready signal which they must respond with an ok, before they can begin sampling. When all nodes are ready, a begin signal is to be received wherein they will now get the readings from the ADC channel of the microcontroller. The nodes will continue to take samples until the Arduino sends the end signal which they must respond with an ok. They will stop sampling and will wait for a submit signal before the acquired samples are sent to the Arduino. To indicate the end of sample submission, they will add an ok signal so that the Arduino may know that the sample the nodes sent is complete.

The measurement from the ADC channel of the microcontroller is obtained which is an integer and to be converted to a floating point which is the voltage. This voltage will be stored in memory together with the other readings until they are submitted to the Arduino. When it is time to submit the samples, it will be sent to the Arduino until the sample is empty before which it will wait again to start a new sample acquisition.

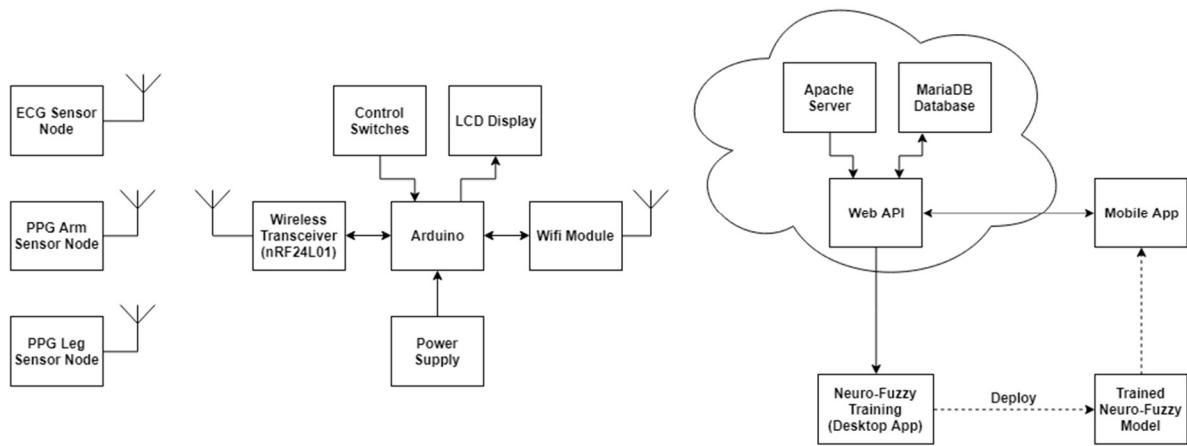


Figure 3.4.2 System Block Diagram

Figure 3.4 shows the block diagram of the system. For the hardware, there are multiple sensor nodes connected to a master microcontroller (Arduino) through an nRF24L01 wireless transceiver. The values sent by the sensor nodes to the Arduino will be displayed to an LCD display attached within the device. Control switches will be used to turn the device on/off as well as traverse through what is displayed. The Arduino will be responsible for determining which data came from what sensor node and will create HTTP requests to the web API from where the data will be stored. A Wi-Fi module will be used to allow the Arduino to communicate with the web API.

For the software part of the system, the web API will serve as the bridge for communication between the hardware and the software. All collected data from the sensors shall be saved to the database, and a desktop app developed in Python shall be responsible for collecting this data and feeding it to the neuro-fuzzy model for training. Once training is done, testing within the desktop app shall be done, and if it produces acceptable results, it will be deployed to the mobile application. The mobile application will be for Android devices only and will be developed in Android Studio using Java/Kotlin.

The Arduino will be responsible for determining when the sensors nodes will acquire the ECG and PPG signals as well as receiving them and sending it to the server. The nodes will be needed to get ready before they begin sampling. Over a specified amount of time, the sampling will be ended, and they will be sent to the Arduino. After receiving all samples from all sensor nodes, the samples will be sent to the server where they are processed.

The Arduino will send a ready signal to all sensor nodes and will wait for their confirmation with an ok signal. The state will be saved in the Arduino and when all the sensor nodes are ready, sampling of ECG and PPG signals may begin. A begin signal is sent to the

sensor nodes, and they will start acquiring data from their respective sensors. A timer is set in the Arduino which indicates period of sampling for the sensor nodes. When the time is up, sampling is to be ended by the Arduino.

To end the sensor nodes acquisition of samples, an end signal is sent to all sensor nodes. An ok signal indicating confirmation that the sensor nodes ended sampling is to be awaited. When all nodes have confirmed that they ended sampling, the Arduino will now receive the samples obtained by the sensor nodes. In doing so, samples are to be received one at a time, which is the reason for selecting a sensor node whose samples will be received. A submit signal is sent to the selected node indicating it to start submitting the samples. An ok signal from the selected node is to be checked which indicates that the node is finished submitting its stored samples. This will be done for all the sensor nodes until all have submitted their samples.

The Arduino will send all samples acquired from the sensor nodes to the server. From the stored samples, each sample will be retrieved and will be sent to the server. It is removed from the stored samples when it is done sending it to the server. This will be done until all samples are sent to the server.

Hardware Development

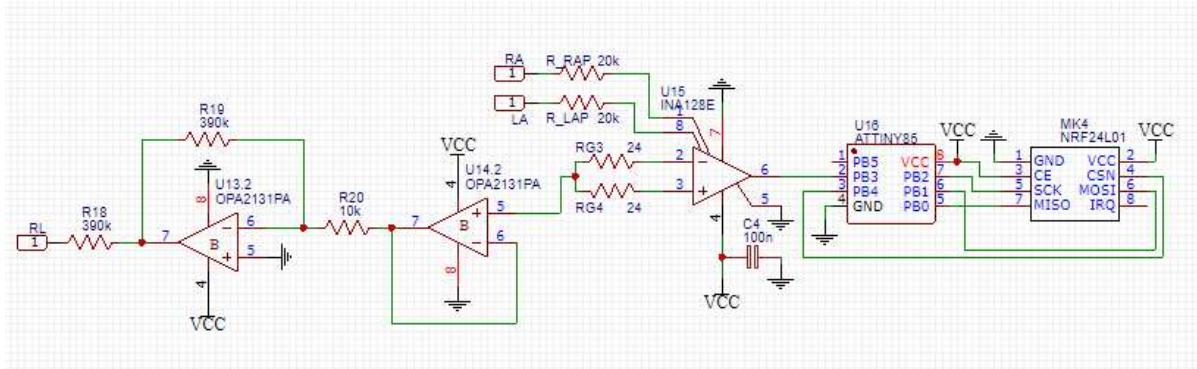


Figure 3.5 Schematic of ECG sensor node

Figure 3.5 depicts the signal conditioning circuit to be used for the ECG sensor node. Three electrodes would be placed on the subject, which are labelled RA, LA, and RL which stands for Right Arm, Left Arm, and Right Leg respectively. For the arm electrodes, resistors are used for input protection before it goes to the instrumentation amplifier. The OPA2131 general-purpose operational amplifier, is used for the Right Led Drive (RLD) amplifier circuit which aims to reduce the common-mode interference by eliminating interference noise. The human body has the tendency to pick up electromagnetic interference (EMI), mostly from the noise generated by electrical power lines. The circuit is adapted from the datasheet of the INA128 instrumentation amplifier where it functions in low voltage operation. ECG signals are relatively weak and requires amplification with high gains. The RG resistors will dictate the gain for the amplifier given the formula.

$$G = 1 + \frac{50k\Omega}{R_G} \quad (3.1)$$

Typical voltage levels of ECG signals range between 1mV~5mV. For a voltage output of up to 5V needed to be buffered in the ADC channel of the microcontroller, the gain needed can be computed using the ratio of the voltage output and the voltage input.

$$G = \frac{V_o}{V_i} \quad (3.2)$$

$$G \approx \frac{5V}{5mV} \approx 1000 \quad (3.3)$$

DESIRED GAIN (V/V)	INA128	
	R _G (Ω)	NEAREST 1% R _G (Ω)
1	NC	NC
2	50.00k	49.9k
5	12.50k	12.4k
10	5.556k	5.62k
20	2.632k	2.61k
50	1.02k	1.02k
100	505.1	511
200	251.3	249
500	100.2	100
1000	50.05	49.9
2000	25.01	24.9
5000	10.00	10
10000	5.001	4.99

Figure 3.6 Gain to R_G resistance values

The table taken from the datasheet of the INA128 instrumentation amplifier shows a list of desired gain with its corresponding R_G resistance value and the nearest available resistance value for 1% precision resistors. For a gain of 1000, a resistance value of 50.05 Ω is needed and is to be divided by two as labeled RG/2 in the schematic diagram.

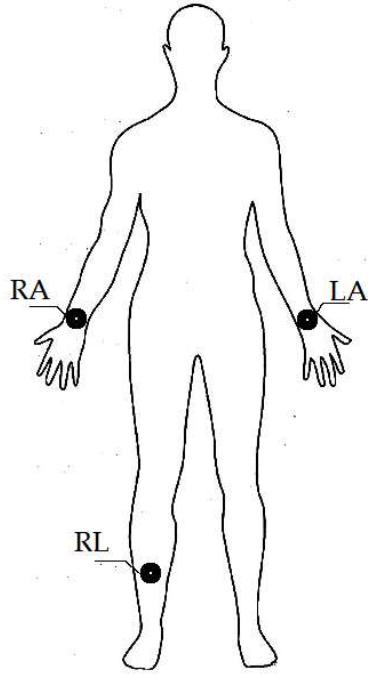


Figure 3.7 ECG electrode placement on the human body

The three electrodes, RA, LA, and RL placements on the human body are shown in the figure. The arm electrodes RA, and LA, are placed near the wrist area of the subject while the RL is placed on the lower right leg.

The output of the signal conditioning circuit will be buffered into the ADC channel of the ATtiny85 microcontroller from where it will be processed, and eventually be sent to the master Arduino microcontroller using an nRF24L01 wireless transceiver.

In estimating the heart rate of the patient, the time difference between ECG R-peaks is needed to be determined. These R-peaks will provide a simple calculation for the heart rate of the test subject using the following equation.

$$\text{Heart Rate (HR)} = \frac{60}{R - R \text{ duration}} \quad (3.4)$$

Systems which measure blood oxygen saturation SpO₂ uses two lights combined which are the red light (660nm) and near infra-red (NIR) light (940nm) to emitted by an LED. A photo detector is to be used to detect the transmitted light by the LEDs, and will be buffered to an amplifier to obtain the desired signal. To measure the oxygen saturation the normalized AC and DC components of the red and IR LEDs are divided with each other.

$$\frac{R}{IR} = \frac{\frac{AC_R}{DC_R}}{\frac{AC_{IR}}{DC_{IR}}} \quad (3.5)$$

The pulse oximeter will need calibration given the data that is obtained through experimentation. For the calibration of the pulse oximeter, a calibrated commercially available pulse oximeter will be used to determine the oxygen saturation of the test subjects. While for the proposed device, the ratio of the AC-DC components ratio will be measured from the PPG signals. This will be combined to create the dataset consisting of the AC-DC components as the inputs and the calibrated pulse oximeter's oxygen saturation readings as the expected outputs. Using the linear regression model with gradient descent algorithm and squared error cost function, the model for the proposed system's pulse oximeter will be estimated and compared to the calibrated pulse oximeter's calibration curve. A graph is to be generated showcasing the calibration curve of the commercially calibrated pulse oximeter, and the proposed system's estimation model will be shown. The test subjects will be divided into two categories, one is for the training dataset, and the other is for the testing dataset. The training dataset will be used for generating the regression model for estimating the oxygen saturation and the testing dataset will be used to determine the accuracy of the device.

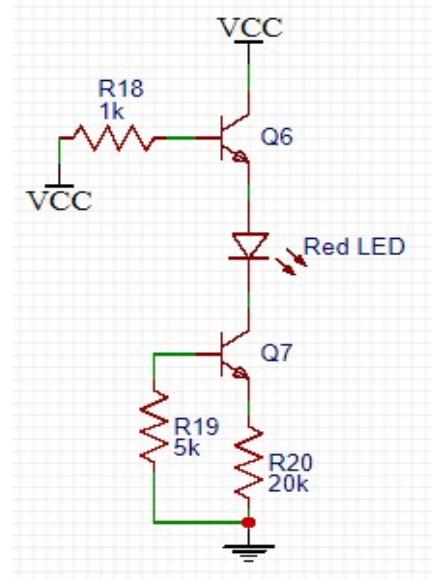
Table 3.1 Sample training dataset for oxygen saturation regression model

Patient #	$\frac{R}{IR}$	Commercially Calibrated SpO2 %

For the testing the accuracy of the estimation model, comparison between the estimated oxygen saturation and the commercially calibrated pulse oximeter's oxygen saturation reading shall be done.

Table 3.2 Comparison of estimated oxygen saturation with commercially calibrated

Patient #	Estimated SpO2 %	Commercially Calibrated SpO2 %

**Figure 3.8** LED driver circuit for red and IR LEDs

A driver circuit for the red and IR LEDs will be used, to provide sufficient power to the LEDs to work. Enable (EN) pins will be connected to the driver circuit as well as a select (S) pin because only one detector circuit will be used, the transmitting LEDs would need to be multiplexed using the microcontroller. For this study, the 696-SML-LX0603SRW red LED which has a wavelength of 660nm and the APTD3216F3C-P22 IR LED which has a wavelength of 940nm will be used. The photodiode to be used in the detector side will be the OP905 which has a wavelength response range of 500nm to 110nm as shown.

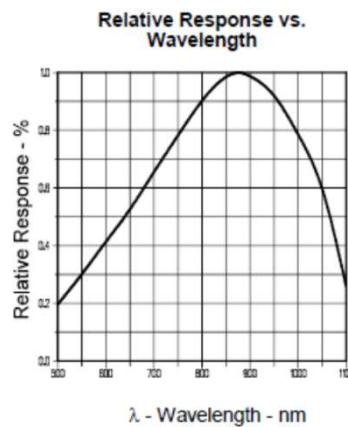


Figure 3.9 OP905 photodiode relative response vs. wavelength graph

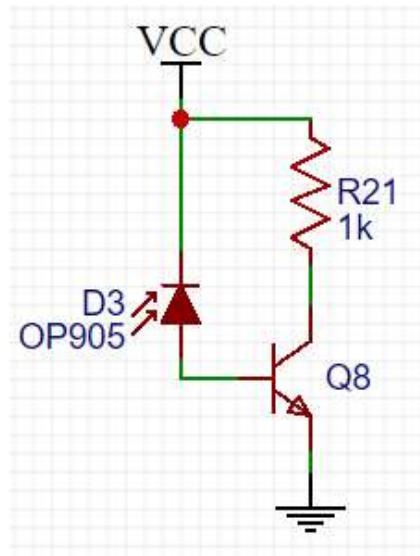


Figure 3.10 OP905 photodiode receiver circuit

The photodiode circuit will be used to detect the red LED and IR LED signals generated from the LED driver circuit. Since the red LED (660nm) and the IR LED's (940nm) wavelengths are within the wavelength response of the OP905 photodiode, it is possible to use this as the detector for the transmitted light in obtaining the PPG signal. The output V_o will be buffered to the signal conditioning circuit for the PPG sensor node.

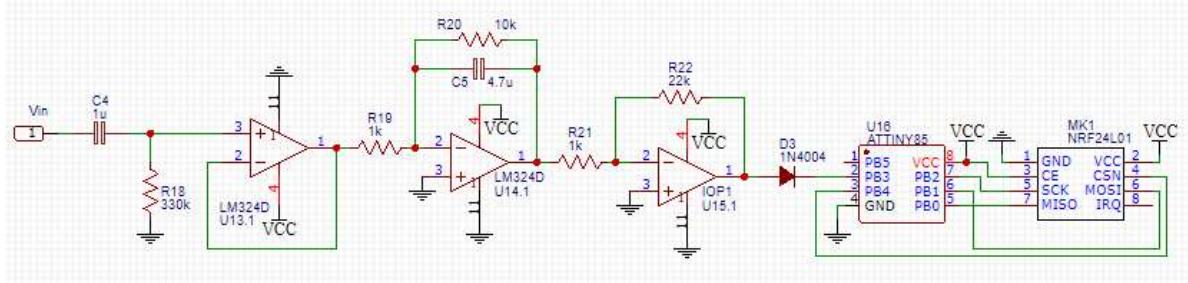


Figure 3.11 Schematic of PPG sensor node circuit

The PPG signal obtained from the LED and photodiode circuit will be buffer to a signal conditioning circuit through the jumper J1. The desired frequency range for this application would be 0.5Hz to 2Hz which is equivalent to 30 to 120 bpm. The heart rate is factored into this due to PPG signal would be affected by the rate of blood flow pumped by the heart. There are three stages for this signal conditioning circuit all of which uses the LM324N operational amplifiers. The first stage is a passive high pass filter and buffer stage wherein signals higher than the cutoff frequency will pass. The cutoff frequency for the high pass filter is computed.

$$F_H = \frac{1}{2\pi \cdot 330k\Omega \cdot 1uF} = 0.483 \text{ Hz} \quad (3.6)$$

$$A_1 = \frac{R_5}{R_4} = \frac{10k\Omega}{1k\Omega} = 10 \quad (3.7)$$

The signal is amplified before going into the next stage, by a factor of ten. The next stage which is an active low pass filter where unwanted high frequency noise is to be removed. The cutoff frequency for the low pass filter is computed.

$$F_H = \frac{1}{2\pi \cdot 10k\Omega \cdot 4.7\mu F} = 3.386 \text{ Hz} \quad (3.8)$$

$$A_2 = \frac{R_7}{R_6} = \frac{22k\Omega}{1k\Omega} = 22 \quad (3.9)$$

In the final stage, higher gain is necessary for the signal to be buffer to the microcontroller's ADC channel. A gain of 22 is used for the final stage making the overall gain equal to 220 computed using the formula.

$$A_T = A_1 \cdot A_2 \quad (3.10)$$

The output is rectified to remove unwanted negative voltages from the signal that will be going in the microcontroller's ADC channel. Similar to the other sensor nodes used within the system's wireless body network, the microcontroller that will process the PPG signal will be an ATtiny85, and has an nRF24L01 wireless transceiver to send the desired parameters obtained from the PPG signal to the master Arduino microcontroller where it will be sent to the server for storage in the database.

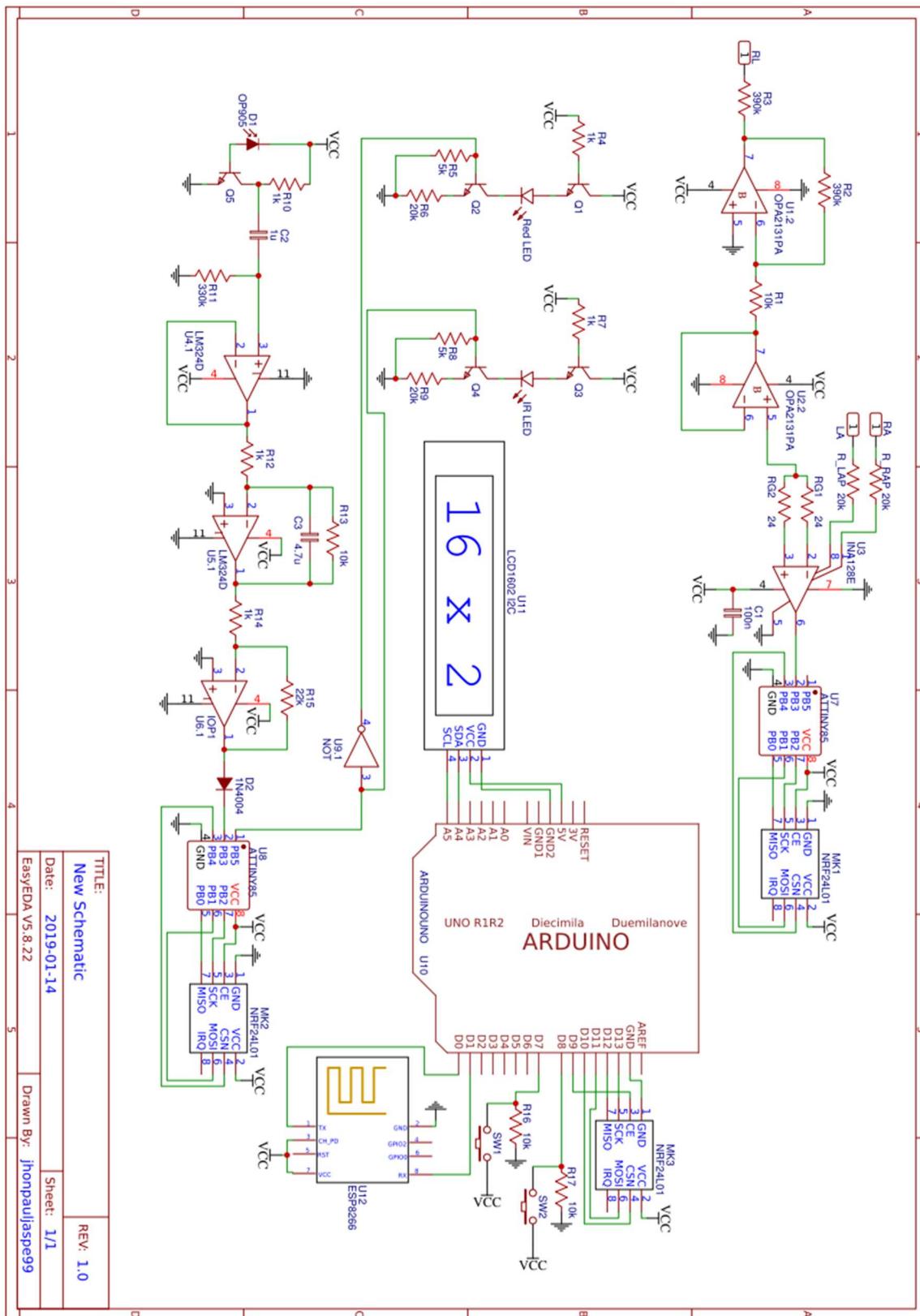


Figure 3.12 Full Schematic Diagram of the System

The schematic diagram of the master Arduino microcontroller consists of an nRF24L01 wireless transceiver module that will receive data from the sensor nodes transmitting through their own nRF24L01 module. It also has an LCD display which is connected to an I2C interface and control switches wherein data sent to the Arduino may be displayed and the position of what is display in the LCD can be controlled using the switches. An ESP8266 Wi-Fi module will be used to transfer data from the hardware to the server. This will be done by using HTTP requests containing data received from the sensors.

Sensor nodes in the system contain an ATtiny85 microcontroller which is attached to an nRF24L01 wireless transceiver. The microcontroller's analog input channel AN2 is where the signal conditioning outputs will be connected. Conditioned sensor output signals will be processed by the ATtiny microcontroller and will be sent to the master Arduino microcontroller.

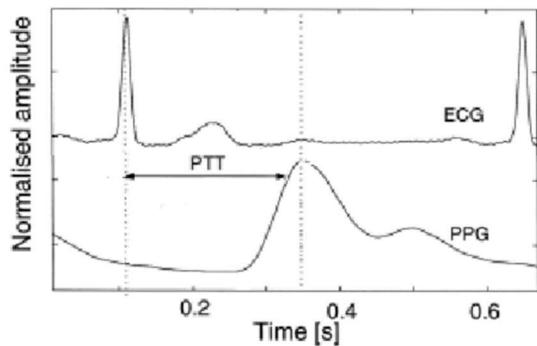


Figure 3.13 (McCarthy, Mathewson, & O'Flynn, 2011) Derivation of PTT from ECG and PPG signals

For each test subject, the ECG and PPG signals are to be obtained along with a measurement their blood pressure using the conventional method of measuring blood pressure. The PTT will be used as the feature in the blood pressure estimation models. The PPG signal from both the arm and the leg are to be measured, which would result in the calculation of two

PTT values, one for the arm, and another for the leg. The Pulse Transit Time (PTT) can be computed by getting the difference between the time of the PPG's maximum/systolic peak (t_S) and the time of the ECG's R peak (t_R) in seconds.

$$PTT = t_S - t_R \quad (3.11)$$

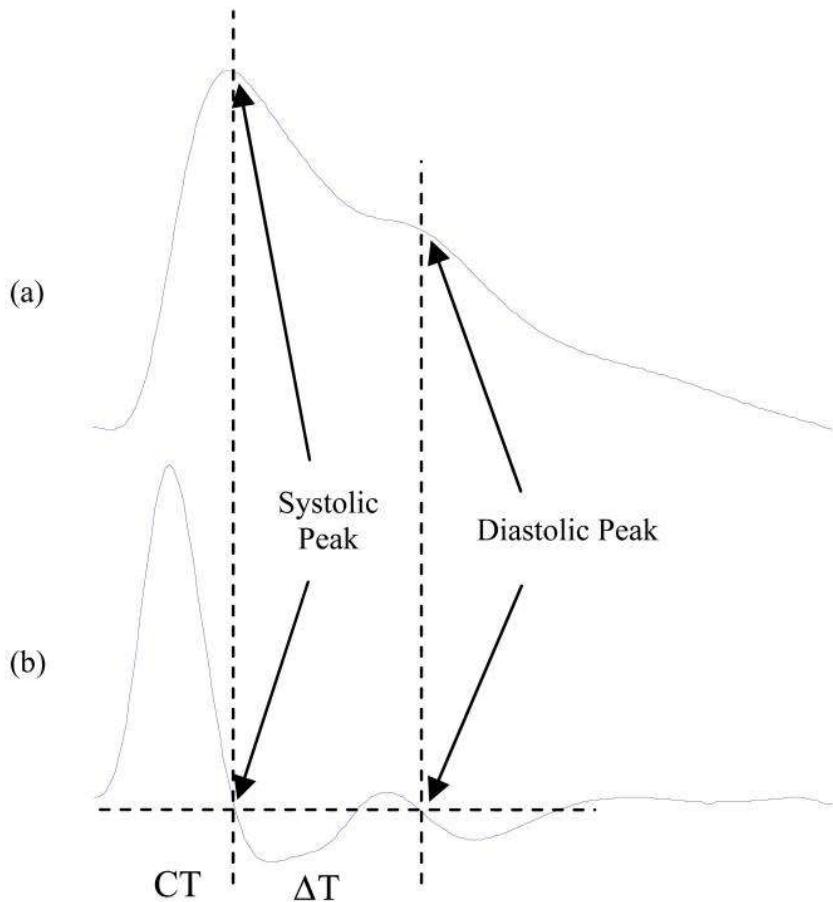


Figure 3.14 (Elgendi, 2012) (a) Original PPG signal (b) First derivative of a PPG

The PPG signal has a systolic and diastolic peak, and the peak-to-peak time ($SDPPT$) is related to the time taken for the pressure wave to propagate from the heart to periphery and back. The systolic peak can be determined by identifying the global maximum of the PPG signal. The dicrotic notch is determined to be the inflection point, and the diastolic peak is the local maximum after the inflection point. For determining the systolic and diastolic peak times,

the first derivative of the PPG signal is to be taken. After which, the time of the two positive to negative zero-crossings are to be measured which is the time of the systolic and diastolic peak times. After determining the systolic and diastolic peaks of the PPG signal, the peak-to-peak time (*SDPPT*) can be obtained by getting the difference between the time of the diastolic peak (t_D) and the time of the systolic peak (t_S) in seconds.

$$SDPPT = t_D - t_S \quad (3.12)$$

Since for the estimation of the systolic blood pressure, the ECG's R-peak time and the PPG's Systolic peak time is used in calculating for the PTT which is its linear regression input. The same concept will be used in estimating for the diastolic blood pressure, but instead of the PPG's systolic peak, its diastolic peak will be used. The R-Peak Diastolic-Peak Time (RPDPT) which is the time difference between the PPG's diastolic peak and the ECG's R-peak will be the input for the diastolic blood pressure linear regression model. To obtain this, the PTT acquired is added to the SDPTT.

$$RPDPT = PTT + SDPPT \quad (3.13)$$

Table 3.3 Sample dataset for the arm blood pressure regression model

Patient #	Arm			
	Normalized PTT	Systolic Blood Pressure (mmHg)	Normalized RPDPT	Diastolic Blood Pressure (mmHg)

Table 3.4 Sample dataset for the leg blood pressure regression model

Patient #	Leg			
	Normalized <i>PTT</i>	Systolic Blood Pressure (mmHg)	Normalized <i>RPDPT</i>	Diastolic Blood Pressure (mmHg)

The given dataset shall be used to determine the systolic and diastolic blood pressure estimation models for both the arm and the leg. It consists of systolic and diastolic blood pressure measurements from the arm and the leg. The systolic and diastolic blood pressure values will be measured using the conventional method of measurement using blood pressure cuffs. Using the proposed system's method, the ECG and PPG signals are to be obtained, given these signals and through a signal processing methodology such as peak finding, the *PTT* and the *RPDPT* values can be calculated. This will be done for both the arm and the leg as well. The calculated *PTT* and the *RPDPT* values are to be normalized using the min-max normalization equation before recorded into the dataset. Both the arm and leg will have their corresponding systolic and diastolic blood pressure estimation models. The dataset will be divided into two categories, one for training and another for testing.

Table 3.5 Estimated and actual arm blood pressure readings

Patient #	Arm			
	Systolic Blood Pressure (mmHg)		Diastolic Blood Pressure (mmHg)	
	Estimated	Actual	Estimated	Actual

Table 3.6 Estimated and actual leg blood pressure readings

Patient #	Leg			
	Systolic Blood Pressure (mmHg)		Diastolic Blood Pressure (mmHg)	
	Estimated	Actual	Estimated	Actual

Once training is completed, the optimal model shall be used for the testing data set where it will estimate the blood pressure of unseen data. Both the arm and leg systolic and diastolic blood pressure estimation models are to be evaluated. The testing dataset will be used to determine whether the obtained blood pressure estimation models are accurate. The estimated and actual systolic and diastolic blood pressure values for both the arm and the leg are needed to evaluate the blood pressure estimation models.

Linear Regression Model

A univariate linear regression model will be utilized for the estimation of the oxygen saturation percentage and the blood pressures. For the oxygen saturation, the input variable is the AC-DC components ratio of the red and IR LED from the PPG signal. The R/IR ratio is used to generate a model that estimated the SpO2%. The parameters θ_i are to be obtained using the gradient descent iterative algorithm.

$$SpO_2\% = \theta_0 + \theta_1 \cdot \frac{R}{IR} \quad (3.14)$$

Where:

$SpO_2\%$ – Oxygen Saturation % (Hypothesis)

$\frac{R}{IR}$ – AC-DC Components Ratio (Input)

θ_i – Parameters

As for the blood pressure estimation model, models for the systolic and diastolic blood pressure will be generated. The blood pressure estimation models will also be univariate linear regression models. The parameters are also to be obtained using the gradient descent iterative algorithm. The calculated Pulse Transit Time (PTT) will be normalized using the Min-Max normalization and will be the input variable for estimating the systolic blood pressure in mmHg.

$$SBP = \theta_0 + \theta_1 \cdot PTT' \quad (3.15)$$

Where:

SBP – Systolic Blood Pressure in mmHg (Hypothesis)

PTT' – Normalized Pulse Transit Time (Input)

θ_i – Parameters

Similarly, the diastolic blood pressure will be estimated using a linear regression model, but the input variable will be the normalized R-Peak Diastolic-Peak Time (RPDPT). The RPDPT will be normalized using the Min-Max normalization as well and will be used to estimate the diastolic blood pressure in mmHg.

$$DBP = \theta_0 + \theta_1 \cdot RPDPT' \quad (3.16)$$

Where:

DBP – Diastolic Blood Pressure in mmHg (Hypothesis)

$RPDPT'$ – Normalized R-Peak Diastolic-Peak Time (Input)

θ_i – Parameters

Squared Error Cost Function

To determine the accuracy of the hypothesis, the squared error cost function is to be computed. For every iteration done using the gradient descent computation of the parameters, the cost function is to be plotted to determine whether it converges. The goal is to minimize the cost function which is indicative of the error between the predicted values and the actual values.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h(x_i) - y_i)^2 \quad (3.16)$$

Where:

$h(x_i)$ – Predicted value of the i^{th} training set

y_i – Actual value of the i^{th} training set

Gradient Descent

Given the blood pressure estimation model, and the input feature PTT, the parameters of the estimation model is to be estimated using gradient descent. Gradient descent is an iterative algorithm for finding the minimum of a function, in this case the cost function. Each of the parameters in the hypothesis function is subjected to gradient descent. A learning rate is to be chosen to determine how fast or how slow the function will move towards the optimal value.

$$\theta_j = \theta_j - \alpha \cdot \frac{1}{m} \cdot \sum_{i=1}^m ((h(x_i) - y_i) \cdot x_i) \quad (3.17)$$

Where:

m – Number of training examples

α – Learning rate

x_i – Value of the input in the i^{th} training set

Min-Max Normalization

One way of speeding up the gradient descent iterative algorithm, the min-max normalization of the training examples may be done. This would allow the input values to have a standardized range of values between 0 and 1. Normalization is done during the preprocessing stage on every training example before it is used in training.

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (3.18)$$

Where:

x_i – Value of the input in the i^{th} training set

x_i' – Normalized value of the input in the i^{th} training set

x – Training set

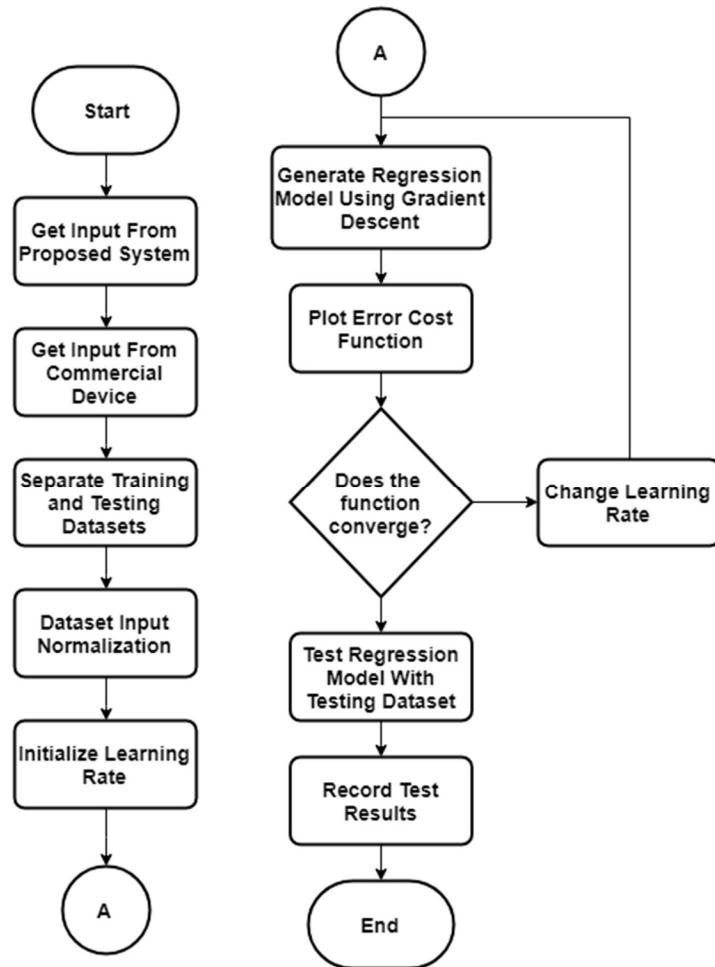


Figure 3.15 Regression Model Generation Flowchart

In generating the regression models to be used for the estimation of the blood pressure and the oxygen saturation, the procedures are to be done. The parameters needed from the proposed device, pulse transit time (PTT) and AC-DC component ratio are to be obtained together with the commercially available devices' blood pressure and oxygen saturation readings to create the training and testing datasets. The inputs for the blood pressure estimation model will be the PTT and the expected output is the blood pressure. While for the oxygen saturation estimation model, the AC-DC component ratio are the inputs and the expected outputs will be the oxygen saturation. The inputs are needed to be normalized since it will

help the gradient descent algorithm find the optimal solution with fewer iterations. Using the gradient descent algorithm and the squared error cost function, the estimation model will be generated. The cost function is to be plotted to determine whether the function converges, if it does, then the model with the more optimal converging method will be chosen. Different learning rates shall be tested to determine which model will be a better estimation model than the other.

In computing for the ABI of a patient, the ratio between systolic blood pressure on the arm and the leg is needed. Using the conventional method of blood pressure measurement, the blood pressure on the arm and leg will be measured. The ECG signal and the PPG signals from both the arm and the leg of the patient are measured. Given these signals, the PTT from the arm and the leg may be computed. The records for the PTT and systolic and diastolic blood pressures for the arm, and the leg, shall be separate training sets for the blood pressure estimation model. The given table is for the training set that will be used in the blood pressure estimation model. Both the arm and leg will have their corresponding systolic and diastolic blood pressure estimation models. The dataset will be divided into two categories, one for training and another for testing.

Table 3.7 Comparison of learning rates based from number of iterations and cost function

Learning Rate	Number of Iterations	Squared Error Cost Function Value

Multiple learning rates shall be considered for the gradient descent to determine the optimal model for estimating the blood pressure. In determining the optimal model, the

learning rate with the least number of iterations, given a converging cost function shall be chosen.

Mean Arterial Pressure

Given the systolic (SBP) and diastolic (DBP) blood pressures (mmHg) the mean arterial pressure (MAP) is approximated using the given formula.

$$MAP = \frac{SBP + 2DBP}{3} \quad (3.19)$$

It must be noted that this approximation is only valid when the subject has a normal resting heart rate. At the beginning of each data acquisition for each subject, a short period will be given to the subjects for them to rest and acquire a normal resting heart rate. Once a normal resting heart rate has been achieved, using the proposed system, the ECG and PPG will first be measured. After which, using the commercially available blood pressure and pulse oximeter, the blood pressure and blood oxygen saturation of the subject will be measured respectively. The recording of the ECG and PPG signals are done first before using the blood pressure cuffs to ensure that any effects of using the cuffed blood pressure measurement device to the ECG and PPG signals are not recorded.

In generating the membership function for the neuro-fuzzy system, a simplified version of a blood pressure chart is utilized. It basically considers the multiple hypertension stages as a single interpretation of high blood pressure.

Table 3.8 Simplified Blood Pressure range interpretation

Interpretation	Systolic Blood Pressure (mmHg)	Diastolic Blood Pressure (mmHg)
Low	Less than 90	Less than 60
Normal	90-120	60-80
Elevated	120-139	80-89
High	Greater than 140	Greater than 90

Ankle-Brachial Index

The most common way of diagnosing a person for peripheral vascular disease only involves the use of the systolic blood pressure measurements. It is done by computing for the ankle brachial index (ABI) of a person which is given by the formula.

$$ABI = \frac{SBP_{Leg}}{SBP_{Arm}} \quad (3.19)$$

Neuro Fuzzy System

A Neuro-Fuzzy System (NFS) is a combination of Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS). For the estimation of the Ankle-Brachial Index (ABI), an Adaptive Neuro-Fuzzy Inference System (ANFIS) is to be used with the Mean Arterial Pressure (MAP) as the input.

ANFIS Architecture

The ANFIS structure will consist of five layers: Fuzzification, Inference, Implication, Aggregation, and Defuzzification layers. It has two inputs which are the Mean Arterial Pressure (MAP) of the arm (x) and the MAP of the leg (y). The output (f) will be the ABI estimated value. In the given structure, the circular nodes are fixed nodes while the square nodes represent the adaptive nodes that contain the adaptive parameters.

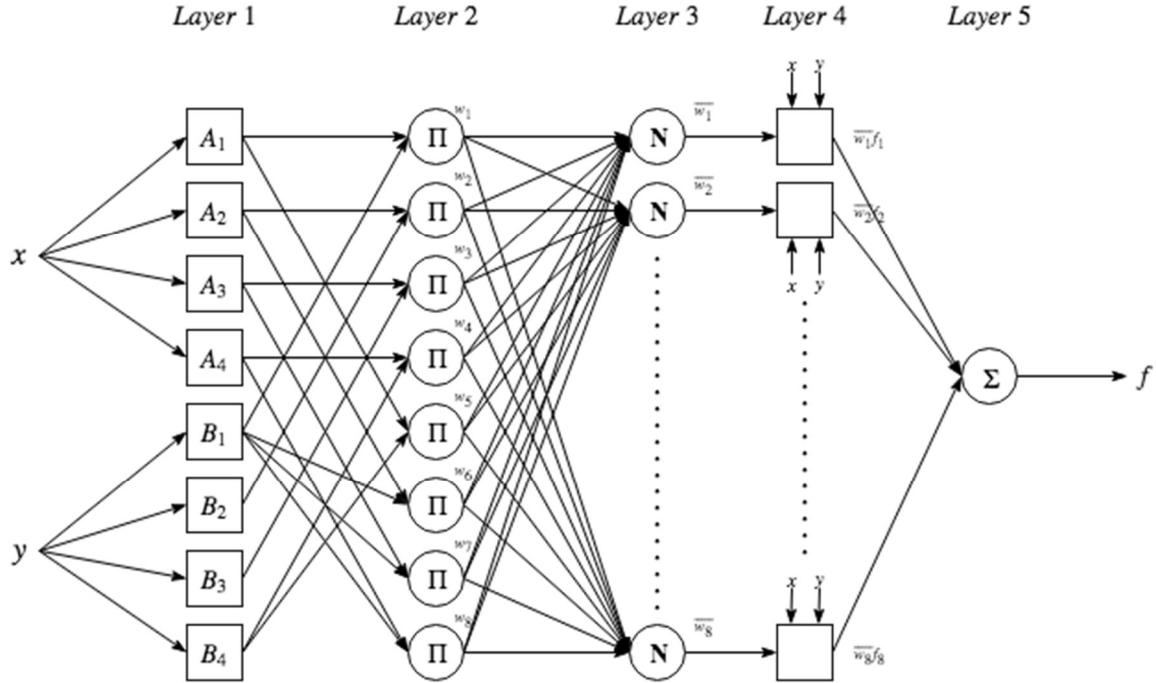


Figure 3.16 ANFIS structure with two inputs, one output, and eight rules.

To further explain the output of each layer, the notation O_i^j where i represents the i^{th} node and j represents the j^{th} layer will be used.

Layer 1: In this layer, the membership function value of the MAP will be generated.

$$O_i^1 = \mu_{A_i}(x), \quad i = 1 \dots 4; \quad (3.20)$$

$$O_i^1 = \mu_{B_{i-4}}(x), \quad i = 5 \dots 8; \quad (3.21)$$

Where x is the MAP value which will be the input for the nodes and A_i is the linguistic label (low, normal, elevated, high). For generating the membership function values, the generalized bell function is utilized which is defined as:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{d_i} \right)^2 \right]^{b_i}} \quad (3.22)$$

Where $\{b_i, c_i, d_i\}$ are referred to as premise parameters, whose values will be updated during training of the neuro-fuzzy system.

Layer 2: In this layer, each node calculates the firing strength of a rule by multiplication for all 8 nodes in this layer:

$$O_i^2 = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(x), \quad i = 1 \dots 8 \quad (3.23)$$

Layer 3: In this layer, the firing strengths calculated from the previous layer is normalized.

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum_{k=1}^n w_k} \quad (3.24)$$

Layer 4: In this layer, each node calculates the overall contribution of the i^{th} rule to the overall output.

$$O_i^4 = \bar{w}_i \cdot f_i = \bar{w}_i \cdot (p_i x + q_i y + r_i) \quad (3.25)$$

Layer 5: In this layer, the single node is a fixed node that calculates the overall output which is the summation of all signals from the previous layer.

$$O_i^5 = \sum_i \bar{w}_i \cdot f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (3.26)$$

MAP Linguistic Labels

Using the simplified blood pressure range chart, the linguistic labels for the mean arterial pressure is defined. These are used to label the premise parameters from the 1st layer of the adaptive network.

Table 3.9 Mean Arterial Pressure Linguistic Label

Low
Normal
Elevated
High

Rule Base

Given the relationship of the leg's blood pressure parameter in the numerator and the arm's blood pressure parameter in the denominator of the equation for the ABI, and the four types of interpretation for the mean arterial pressure. A rule base with eight rules is constructed with idea of what the interpretation of the MAP would be for the arm and the leg to generate an ABI value that is related to the ratio of the leg and the arm blood pressure. The rules are defined as follows:

- 1) If Arm MAP is **LOW** and Leg MAP is **LOW**, then $f_1 = p_1x + q_1y + r_1$.
- 2) If Arm MAP is **NORMAL** and Leg MAP is **NORMAL**, then $f_2 = p_2x + q_2y + r_2$.
- 3) If Arm MAP is **ELEVATED** and Leg MAP is **ELEVATED**,
then $f_3 = p_3x + q_3y + r_3$.
- 4) If Arm MAP is **HIGH** and Leg MAP is **HIGH**, then $f_4 = p_4x + q_4y + r_4$.
- 5) If Arm MAP is **LOW** and Leg MAP is **HIGH**, then $f_5 = p_5x + q_5y + r_5$.
- 6) If Arm MAP is **NORMAL** and Leg MAP is **LOW**, then $f_6 = p_6x + q_6y + r_6$.
- 7) If Arm MAP is **ELEVATED** and Leg MAP is **LOW**, then $f_7 = p_7x + q_7y + r_7$.
- 8) If Arm MAP is **HIGH** and Leg MAP is **LOW**, then $f_8 = p_8x + q_8y + r_8$.

ANFIS Learning Algorithm

The ANFIS is trained using a hybrid learning algorithm in determining the premise parameters in the 1st layer (b, c, d) and the consequent parameters (p, q, r) in the 4th layer. In the forward pass, the linear consequent parameters are determined using the least square estimate. For the backward pass, the non-linear premise parameters are determined using backpropagation by gradient descent.

Error Measure

Given a training dataset with P entries, the error for the p th entry in the dataset is defined as the sum of the squared error.

$$E_p = \sum_{i=1}^{\#(L)} (T_{i,p} - O_{i,p}^L)^2 \quad (3.27)$$

Where

$T_{i,p}$ is the i th component of the p th target entry.

$O_{i,p}^L$ is the i th output at layer L of the p th entry.

L is the number of layers

$\#(k)$ is the number of nodes at the k th layer

For the beginning of the learning process, the error rate at the output i th node at layer L is computed as follows:

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2(T_{i,p} - O_{i,p}^L) \quad (3.28)$$

Generalized derivative formula of the error of the p th entry with respect to the output of a node at the k th layer, and i th node of the p th training dataset entry.

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k} \quad (3.29)$$

For each premise parameter, its error rate is calculated through back propagation which observes the chain rule. O^* is output of each layer within the set S until it reaches the parameter in question. For the generalization of the equation α denotes the parameters, wherein each premise parameter's (b, c, d) error rate must be determined.

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha} \quad (3.30)$$

The derivative of the overall error with respect to α is:

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha} \quad (3.31)$$

Update formula for a parameter α , using the gradient descent method.

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \quad (3.32)$$

The learning rate η , with k as the step size that indicates the convergence rate of the adaptive network is defined as follows:

$$\eta = \frac{k}{\sqrt{\sum \alpha \left(\frac{\partial E}{\partial \alpha} \right)^2}} \quad (3.33)$$

Least Square Estimate

To obtain the consequent parameter in the 4th layer of the adaptive network, the least square estimate is used. From the output which is the summation of all the normalized weight \bar{w}_i and the output f_i at each node i , wherein f_i contains the consequent parameters (p, q, r) , the expanded form of the equation is obtained. Since the neuro fuzzy system has 8 rules in its rule base, there are 8 consequent parameters (p, q, r) to be obtained.

$$\begin{aligned}
 f &= \sum_{i=1}^8 \bar{w}_i \cdot f_i \\
 &= \bar{w}_1 \cdot f_1 + \dots + \bar{w}_8 \cdot f_8 \\
 &= \bar{w}_1 \cdot (p_1x + q_1y + r_1) + \dots + \bar{w}_8 \cdot (p_8x + q_8y + r_8) \\
 &= (\bar{w}_1x)p_1 + (\bar{w}_1y)q_1 + (\bar{w}_1)r_1 + \dots + (\bar{w}_8x)p_8 + (\bar{w}_8y)q_8 + (\bar{w}_8)r_8
 \end{aligned} \tag{3.34}$$

For the whole training dataset where n is the number of entries, the equation obtained is as follows with y as the output for each entry

$$\begin{aligned}
 (\bar{w}_1x)_1p_1 + (\bar{w}_1y)_1q_1 + (\bar{w}_1)_1r_1 + \dots + (\bar{w}_8x)_1p_8 + (\bar{w}_8y)_1q_8 + (\bar{w}_8)_1r_8 &= y_1 \\
 &\vdots \\
 &\vdots \\
 (\bar{w}_1x)_np_1 + (\bar{w}_1y)_nq_1 + (\bar{w}_1)_nr_1 + \dots + (\bar{w}_8x)_np_8 + (\bar{w}_8y)_nq_8 + (\bar{w}_8)_nr_8 &= y_n
 \end{aligned} \tag{3.35}$$

To simplify the equation, it is represented in matrix form with θ as the unknown consequent parameter vector that is needed to be obtained.

$$A\theta = y \tag{3.36}$$

The consequent parameters (p, q, r) are obtained as a matrix using the given formula

$$\theta = (A^T A)^{-1} A^T y \tag{3.37}$$

Software Development

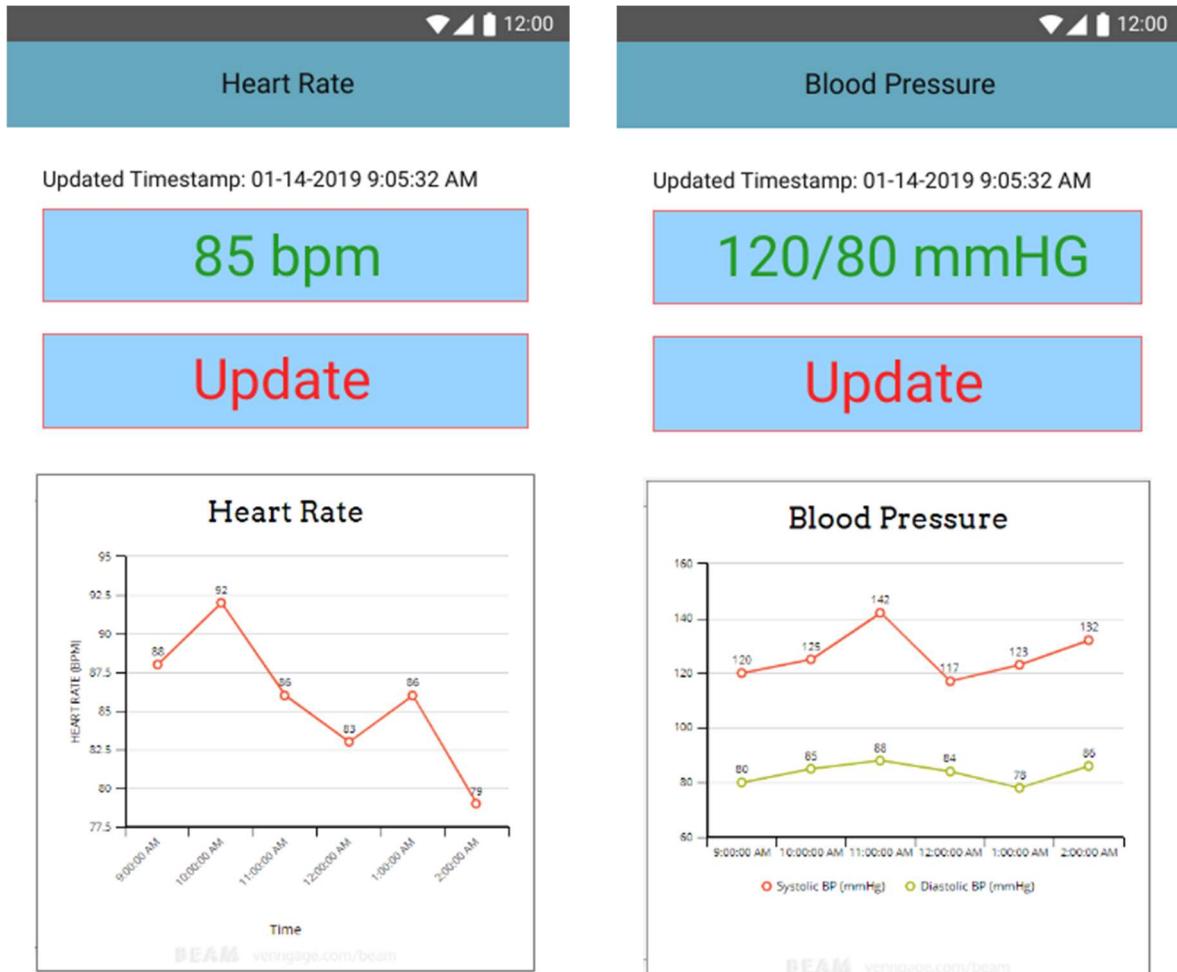


Figure 3.17 Heart Rate and Blood Pressure Pages for GUI Mock-up

The figures show a sample of the GUI mock-up for the monitoring mobile application.

The updated values are shown together with the time of last update from the server. Graphical representation of the parameters over time are shown for easier visualization. As long as the mobile application is connected to the internet, it will update with the latest readings stored in the server. The heart rate is shown in beats per minute (bpm), while the blood pressure graph shown two different lines indicating the systolic and diastolic blood pressures over time. An

Update button will allow the user to retrieve the data from the server at will, if there exist an updated measurement.

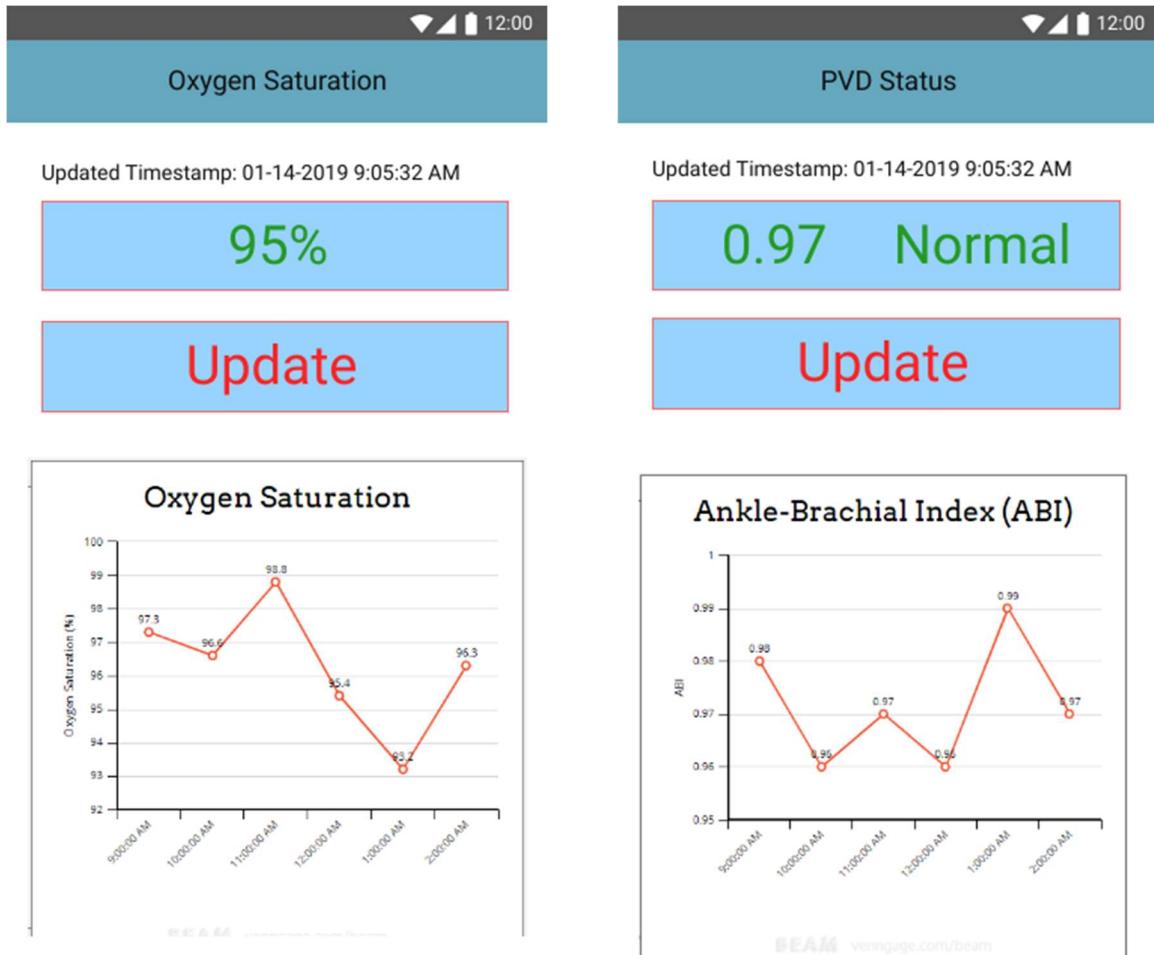


Figure 3.18 Oxygen Saturation and PVD Status Pages for GUI Mock-up

The figures show a sample for the pages of the oxygen saturation and the PVD status of the user. The oxygen saturation is shown in terms of percentage, from the last updated value retrieved from the server and the values represented as a graph over time. For the PVD status, the ABI value is shown together with the interpretation of its severity. To monitor the PVD status, the graph of the ABI over a period is shown. It is updated from as along as it is connected to the internet and the device is being used.

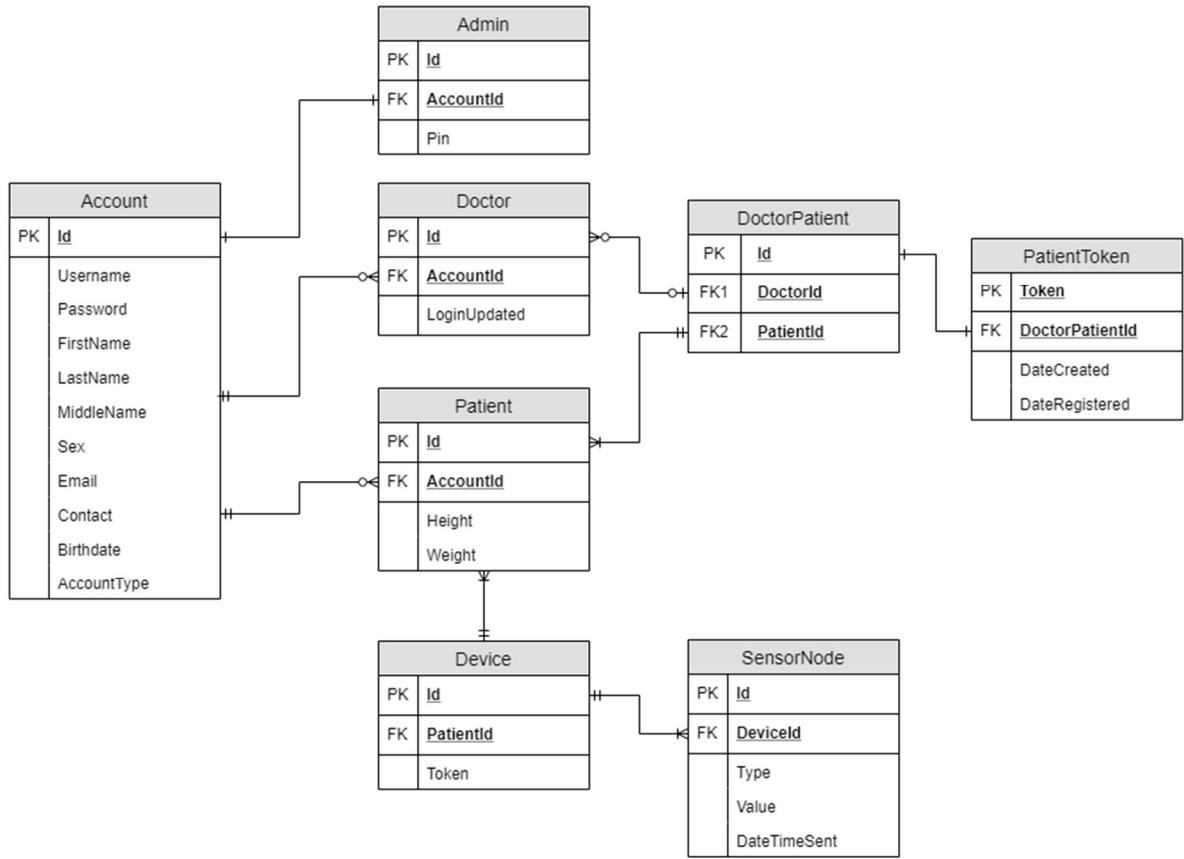


Figure 3.19 Entity Relationship Diagram

There will be three types of accounts that is built within the system, the admin, doctor, and patient which is reflected in the database. For every account, basic information about the user is necessary and will be stored in the database. Doctors may have a list of patients and in order to store this, the DoctorPatient table is necessary. It is collection of doctor and patient ids which allow identification whose doctor a certain patient belongs to. Tokens will be used for patient's registration. When a doctor registers a patient to the application, the token is generated and stored in the database. The doctor must provide the patient with the token generated. Upon a patient's attempt for registration, the token provided is checked whether it is valid and is already in the registered for the registration to proceed. Patients have the device

that allows their health to be monitored. Each device is associated to a patient and has a unique token that is hardcoded in the device. This allows identification whose device is sending data to the server. A device may have multiple sensor nodes connected which allows the physiological parameter values to be identified from which device it came from, what type of value it is sending as well as the actual value that it saves in the database.

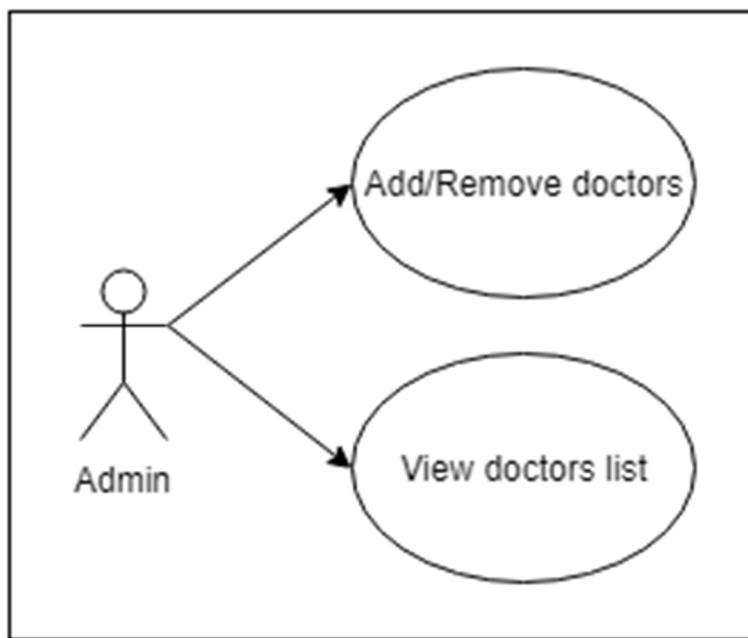


Figure 3.20 Admin Use Case Diagram

For the mobile application, an admin will be responsible for maintaining the list of doctors that are participating in the application. In order to make the application more exclusive and avoid the case of fake accounts, an admin may be assigned by the hospital or facility using the application. The admin is may only be the one who can add or remove doctors, as well as view all the doctors that are registered.

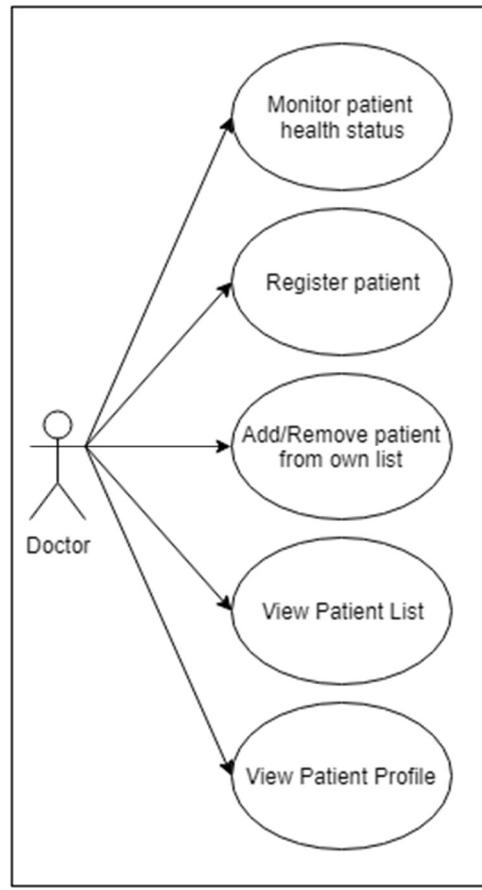


Figure 3.21 Doctor Use Case Diagram

Doctors which are added by the admin may be able to use the mobile application to monitor their patient's health status. This would include all measured physiological parameters as well as an overall PVD status of the patient. Graphical representation and data history may also be viewed through here. Patients are registered by the doctors; necessary information are provided by the doctors. Upon registration, a "Patient Token" is provided and the doctor must give it to the patient or a permitted person to monitor the patient's health status. This token is then used to complete the patient's account, by providing a username and password for the patient's login information. After which, the patient may be able to use the mobile application to monitor their health status. The doctor may also add the patient to their list, for easier access to the patient's information.

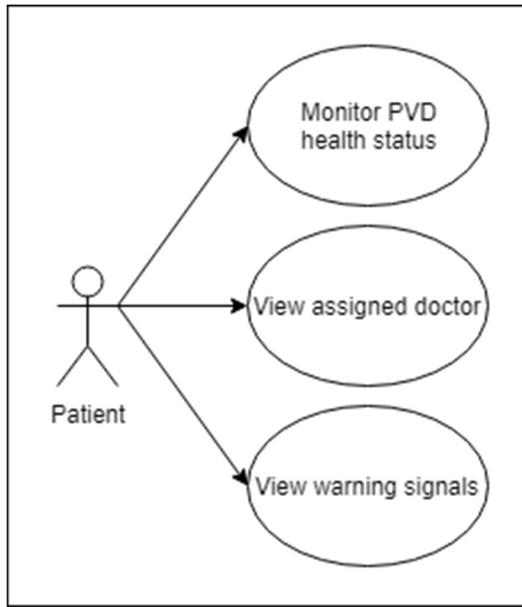


Figure 3.22 Patient Use Case Diagram

Patients may access the mobile application to monitor their health status, which is all the physiological parameters that were measured from them by the sensors as well as the overall status of their peripheral vascular disease as produced by the neuro-fuzzy network. Graphical representations of the data over-time will also be provided as a more user-friendly way of monitoring their health status. The patient may view the doctor assigned to them which is the one who also provided their registration. Early warning signals will also be provided to warn the patients when a health parameter has a value that is not normal for example, when the blood pressure is high.

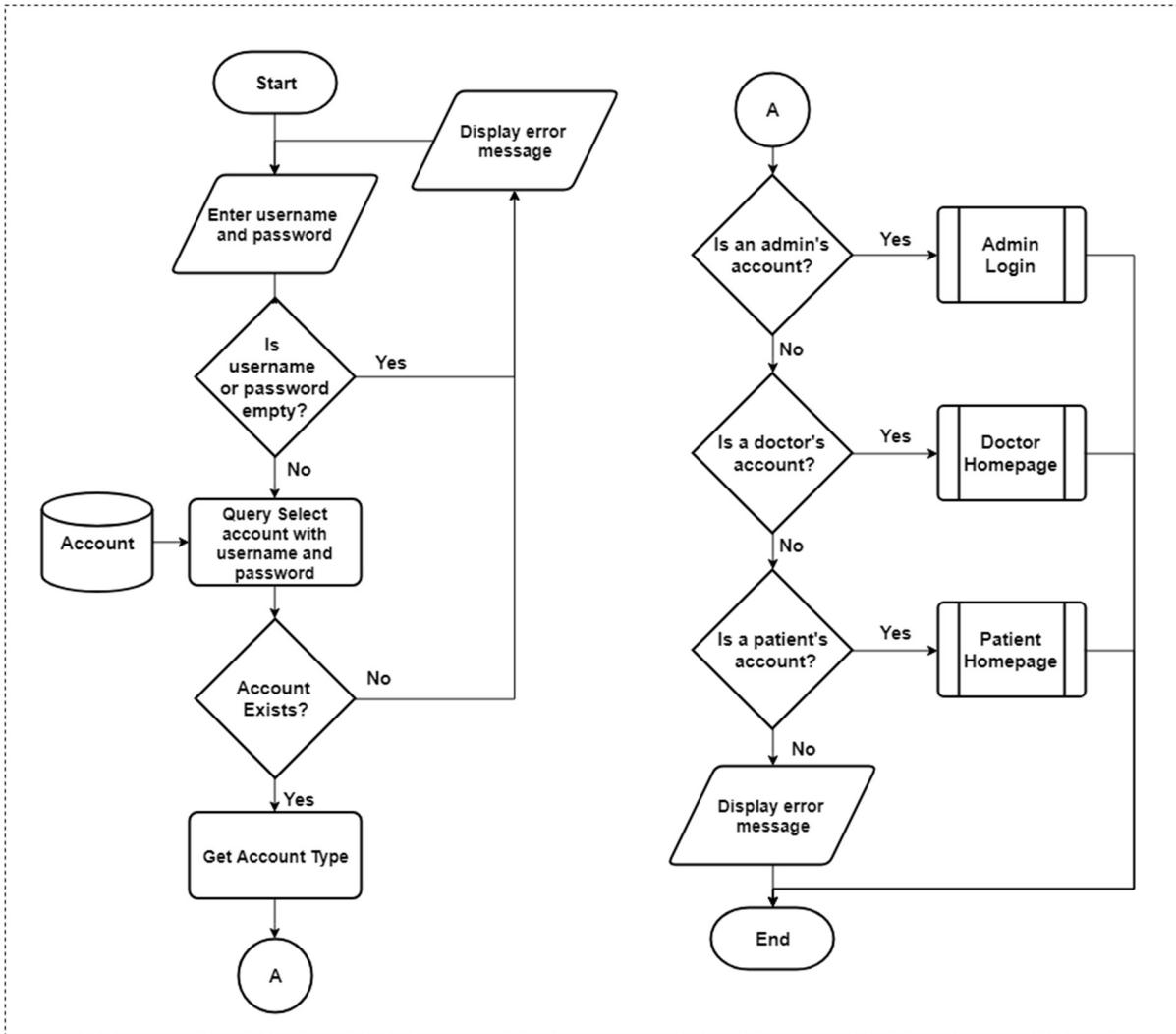


Figure 3.23 Mobile Application Login Page Flowchart

Figure 3.23 depicts the login page for the mobile monitoring application. The user is presented with text boxes wherein the username and password will be entered. When the user tries to log in without any input, an error message is displayed. For the case when input is present, the account whose username and password belongs to is obtained from the database. If the account doesn't exist, an error message will be displayed, if it does exist, the account type is used to determine the next page the user will be presented. There are three types of accounts, the admin, the doctor, and the patient, each of which has different pages presented in the application as well as functionalities for these pages.

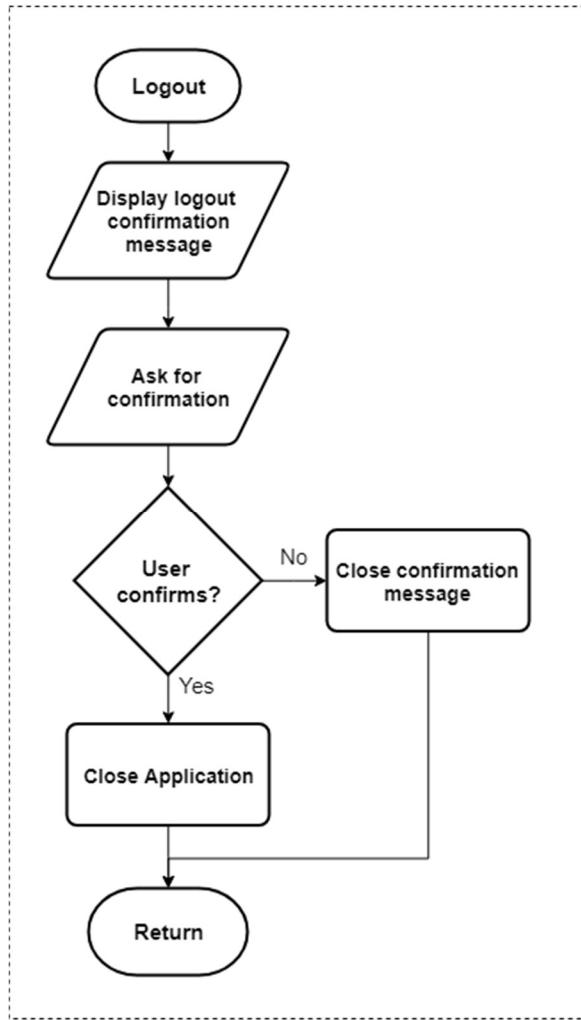


Figure 3.24 User Logout Flowchart

Figure 3.24 depicts the flowchart for when a user tries to logout of the application. A confirmation is first displayed notifying the user of the action to be done and asks for confirmation, to avoid accidental presses. If the user doesn't confirm logging out, the message is closed and the application return to its previous state and operations. The application is closed when the user confirms logging out.

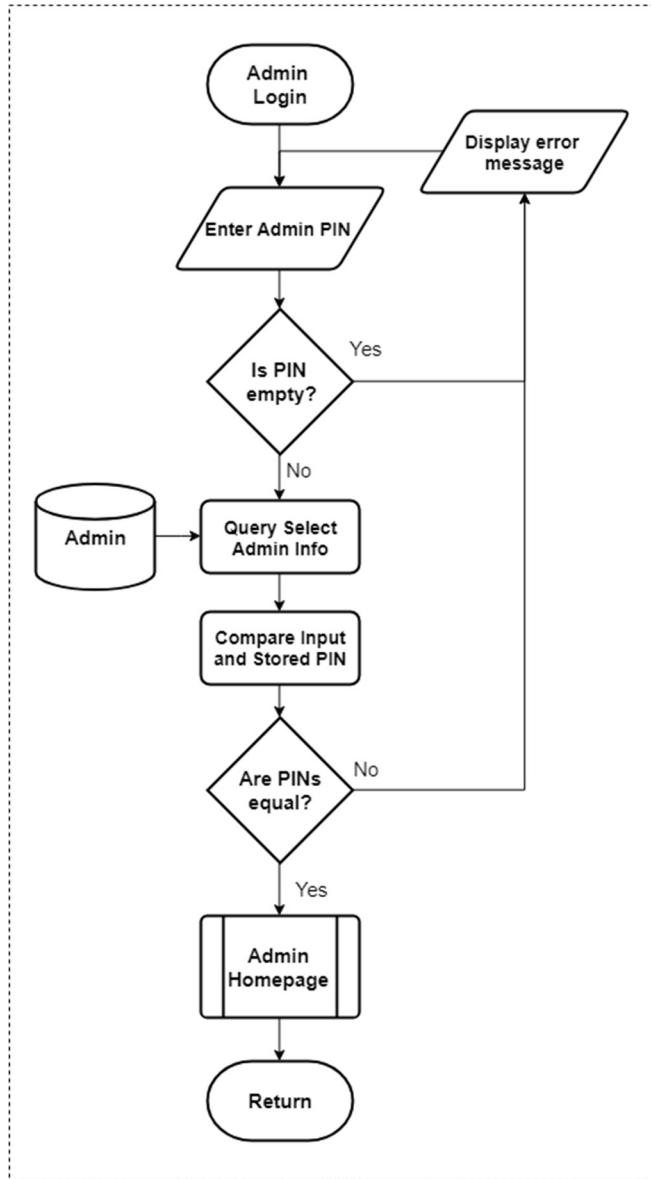


Figure 3.25 Admin Login Flowchart

Figure 3.25 depicts the admin login flowchart. The admin is asked for the PIN number that is associated with their account. Empty input is checked with corresponding error messages. The admin's information is obtained from the database and the stored PIN is compared with the input PIN to determine its validity. An error message is displayed if they are not equal and if they are, the admin is presented to their homepage.

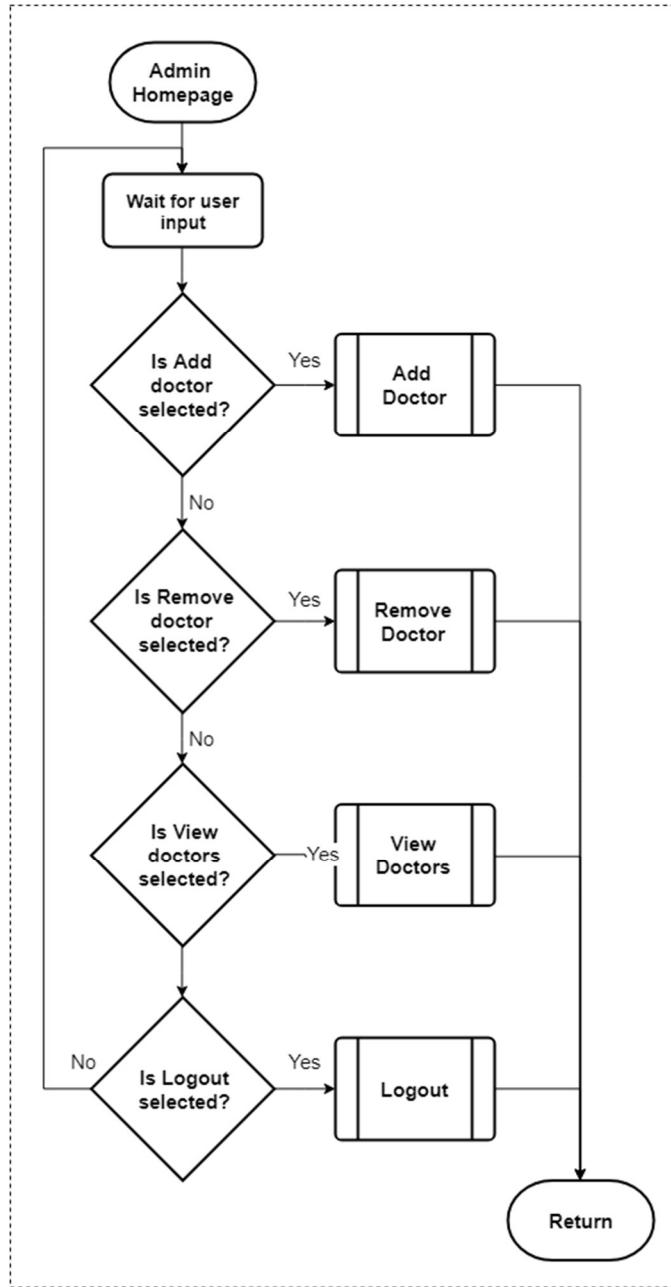


Figure 3.26 Admin Homepage Flowchart

Figure 3.26 depicts the flowchart for the admin's homepage. The admin's homepage can be thought of as just a gateway to the admin's main functionalities. In this page, there are buttons that allow the user to navigate to other pages, and the user's input is being waited

for. The admin can add and remove doctors from the application, as well as view the list of doctors registered in the app. Finally, the admin may logout of the application.

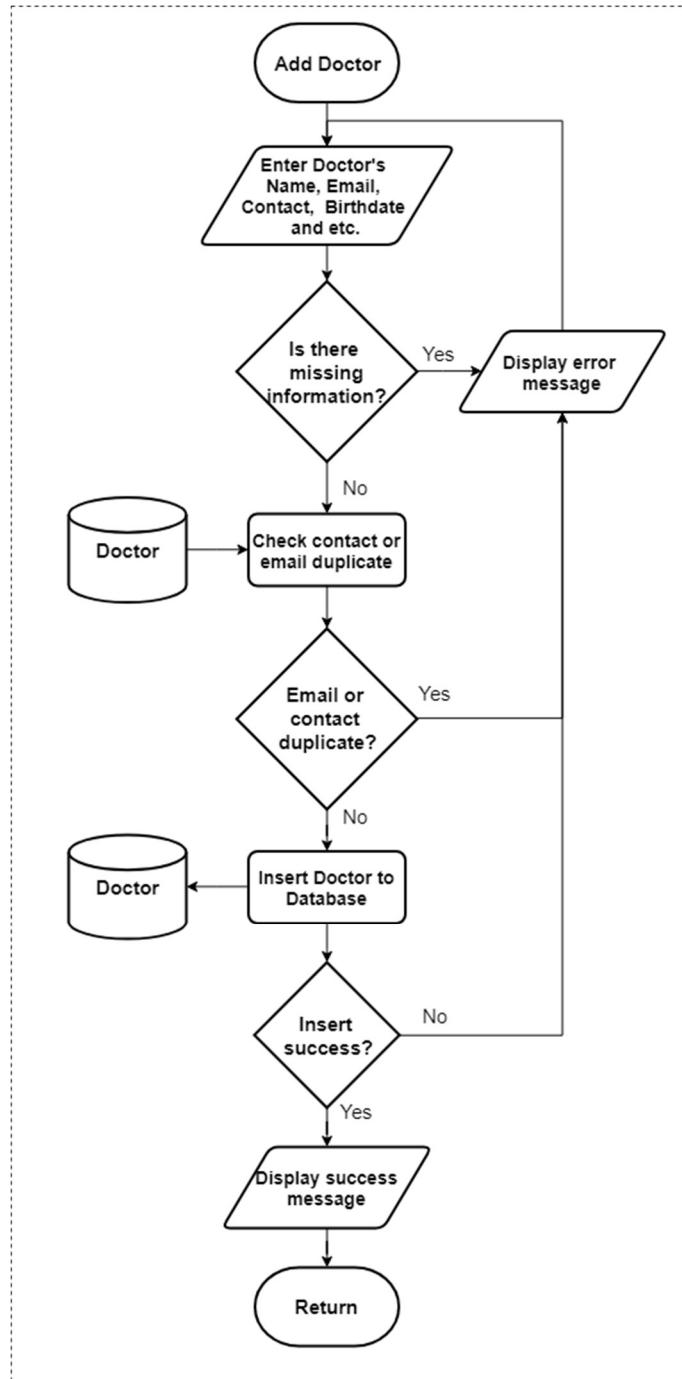


Figure 3.27 Add Doctor Flowchart

Figure 3.27 depicts the flowchart for adding doctors in the application. The admin is the only one capable of adding and removing doctors. General information of the doctor is entered for registration, and is checked to make sure that there isn't any blank information left by the admin. The contact number and email address must be unique for every doctor, so it is checked for any duplicates in the database, an error message is displayed if there is a duplicate. If it is unique, then it is inserted to the database, and database insertion is also checked if it is successful. It is to be noted that by default, the doctor's username and password is their email and contact number respectively, because it is confidential data and the doctor may not want the admin to know their credentials. The username and password of the doctor is required to be updated by the themselves upon first login in the app.

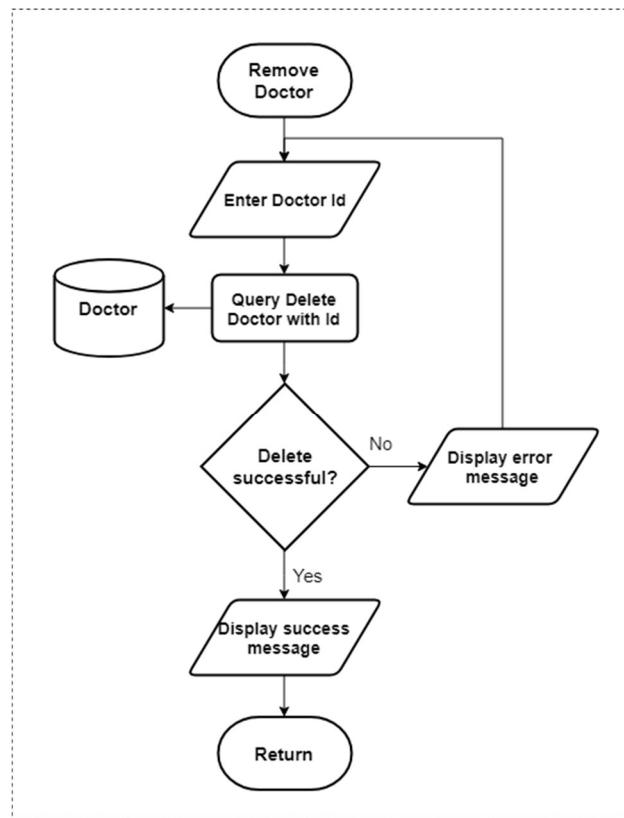


Figure 3.28 Remove Doctor Flowchart

Figure 3.28 depicts the flowchart for the removal of doctors in the app. The admin must enter the id of the doctor for the application. A delete query is issued to the database containing the doctor id, and successful deletion is checked. If it is not successful, it may be the case of non-existent or invalid doctor id entered by the admin, thus an error message is displayed, and execution of the program continues to ask for input again. Upon successful deletion of the doctor's account, a success message is displayed.

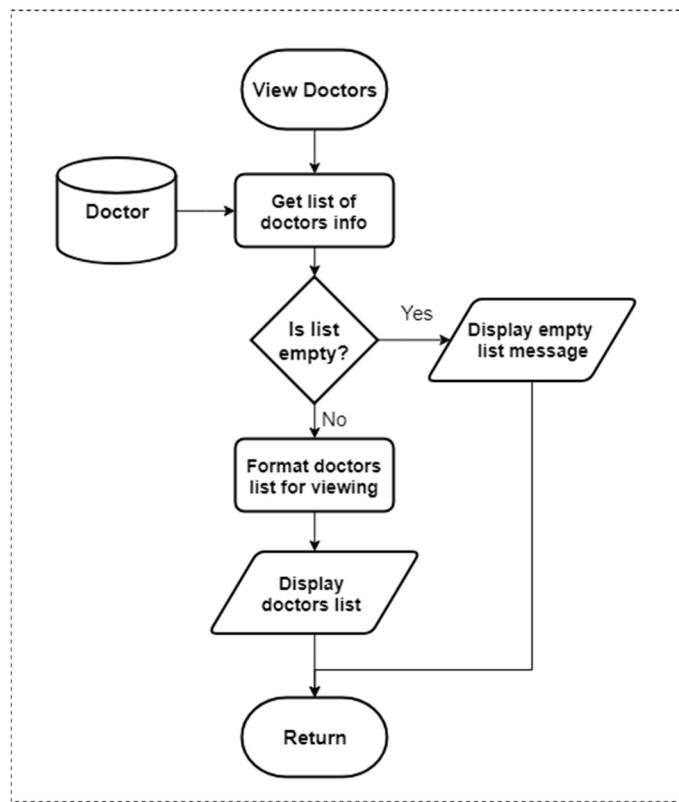


Figure 3.29 View Doctors Flowchart

Figure 3.29 depicts the flowchart for viewing the list of doctors in the app. All acquirable doctor information stored in the database is obtained and a list is returned from the query. The list is checked if it is empty, to display a message indicating that there are no doctors registered in the app. The non-empty list of doctor information is formatted for displaying and finally it is displayed for the admin to view.

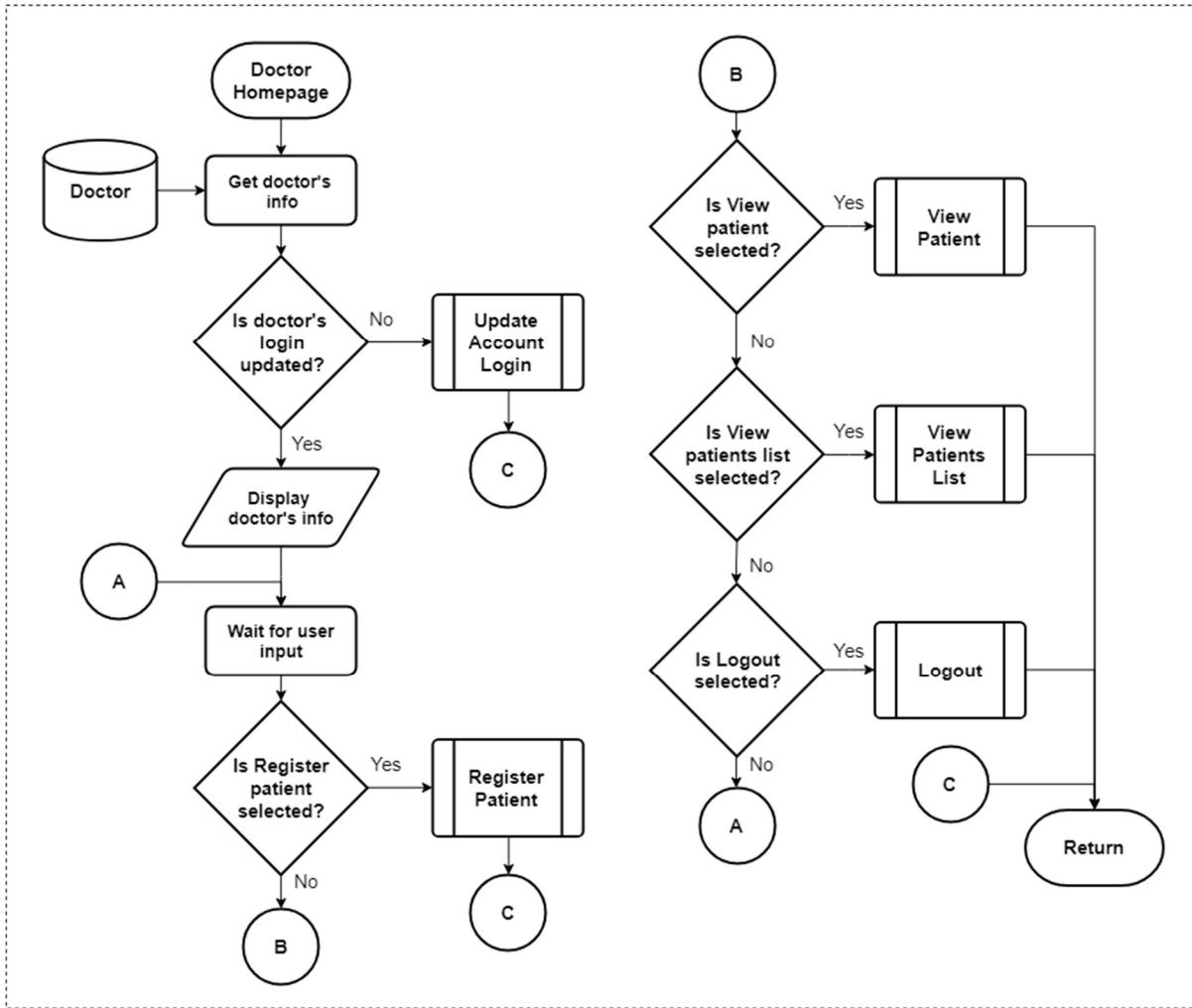


Figure 3.30 Doctor Homepage Flowchart

Figure 3.30 depicts the homepage of the doctor in the application. The doctor's info is obtained from the database and checked whether his account login credentials are already updated. If it is not, a new page wherein they will be able to update their account login info is to be displayed. The doctor's functionalities within their homepage include registering a patient, viewing a patient, and viewing their list of patients. Like all users of the app, the doctor may also log out.

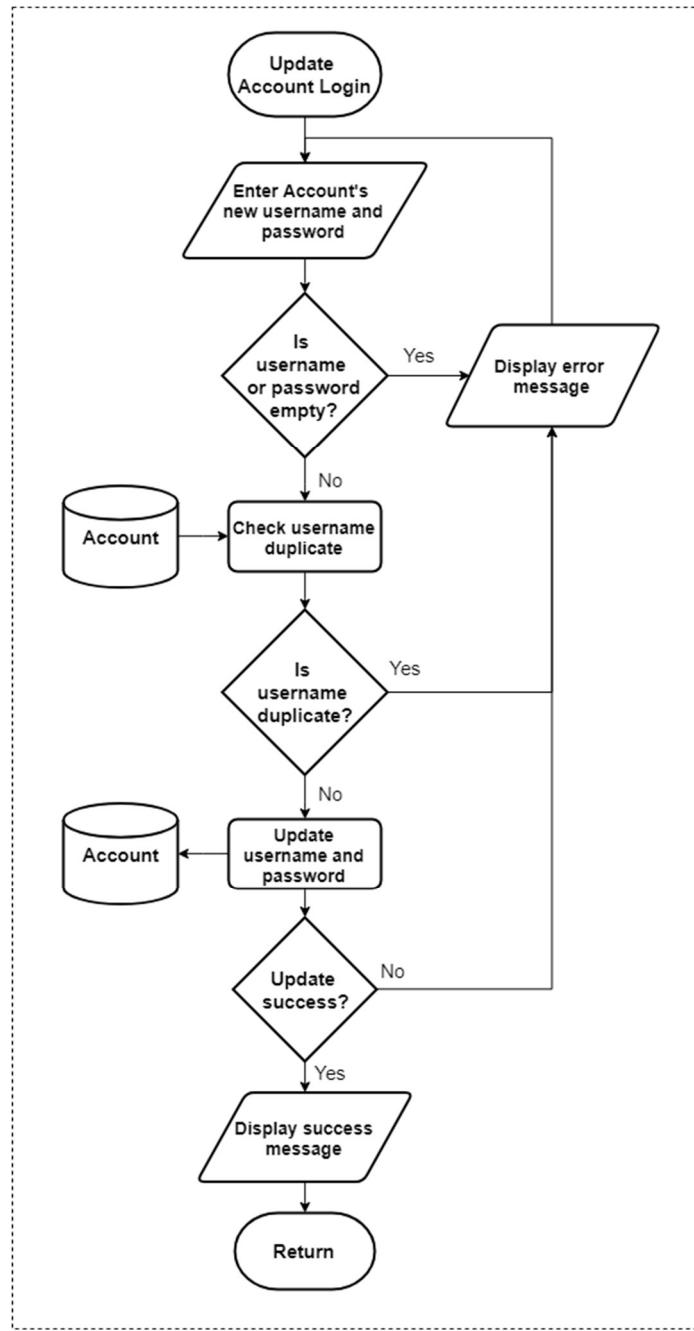


Figure 3.31 Update Account Login Flowchart

Figure 3.31 depicts the update account login flowchart. The doctor and the patient accounts may be able to update their login information, since they themselves will not be the ones registering their information in the application. For the doctor, the admin registers them,

and for the patient, the doctor registers them. This process allows them to change their username and password that is used when logging in the app.

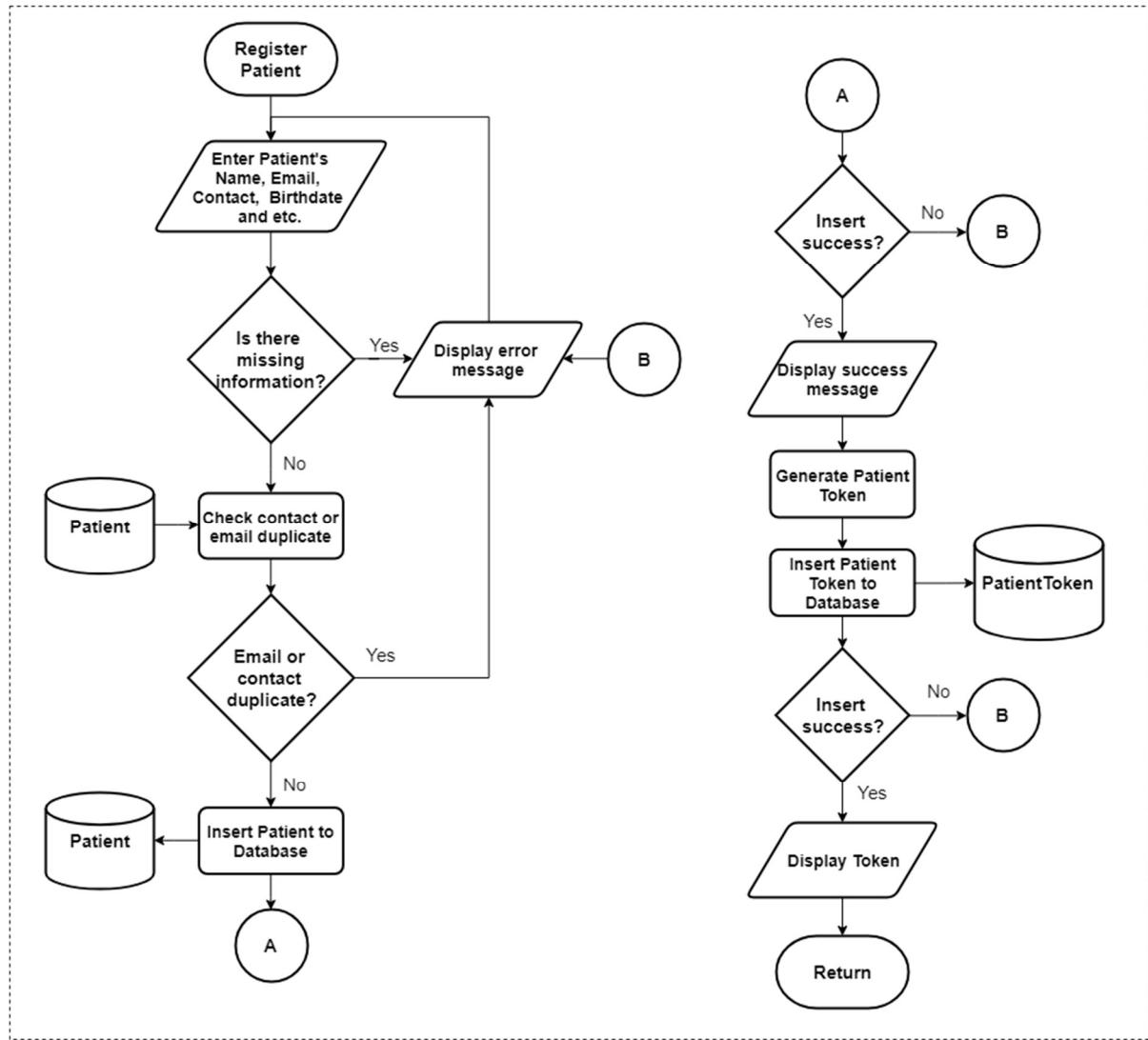


Figure 3.32 Register Patient Flowchart

Figure 3.32 depicts the flowchart for a patient's registration process. The doctor is the only one capable of registering a patient to the application. The patient's general account information is needed to be entered and there must be no empty fields. Similar to the registration of the doctor, the contact and email must be unique from the other registered accounts. The patient's information is inserted to the database when it is valid, and error

messages are displayed in case of any insertion errors or invalid information. A patient token will be generated and stored in the database. The doctor must give this token to the patient to continue their registration.

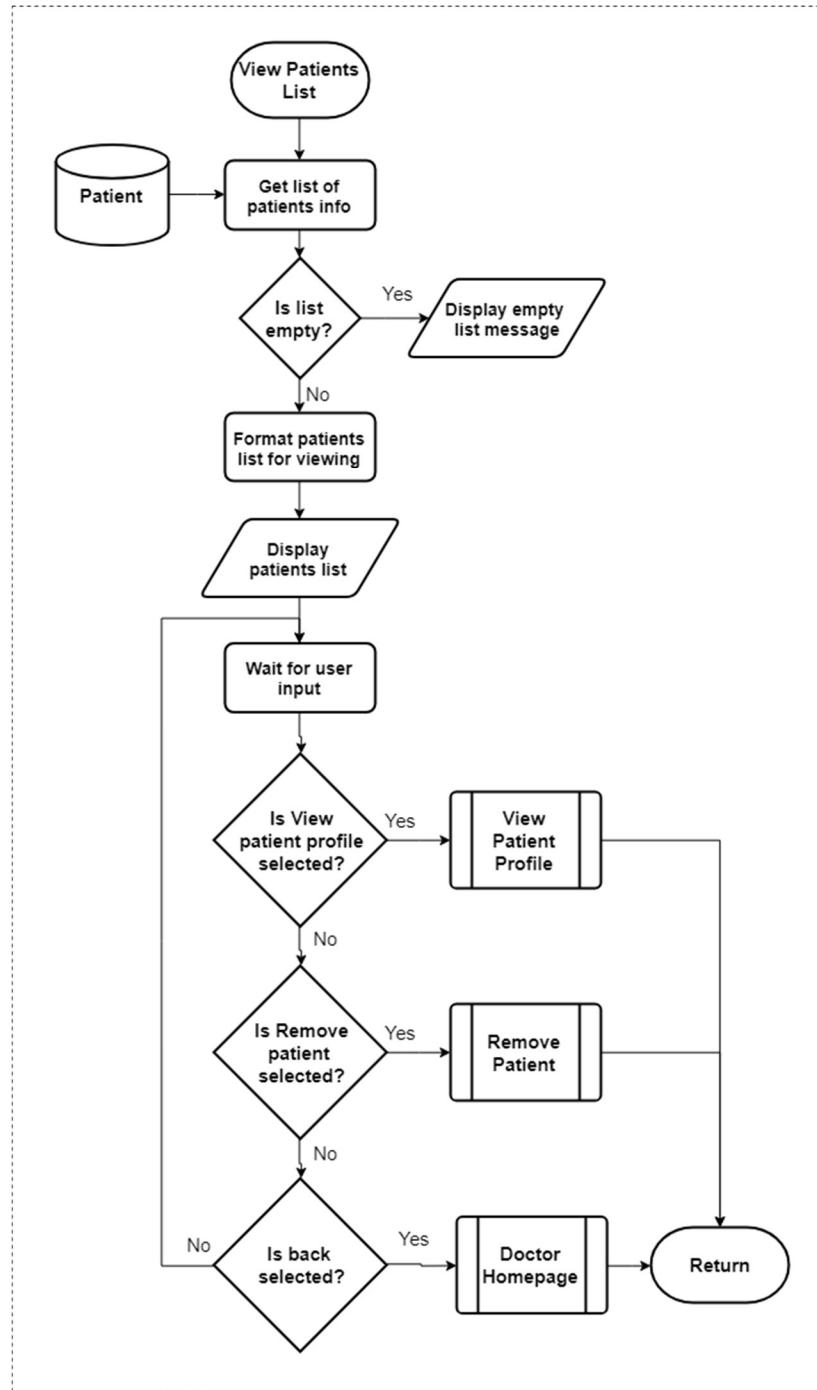


Figure 3.33 View Patients List Flowchart

Figure 3.33 depicts the view patients list flowchart. Doctors have a list of patients stored in the database. To view this, the list of patient information is obtained from the database and is formatted for being displayed as a list. The doctor may also perform operation to patients selected from the list. They may view their profile or remove the patient from their list. Pressing the back button will navigate back to the homepage of the doctor.

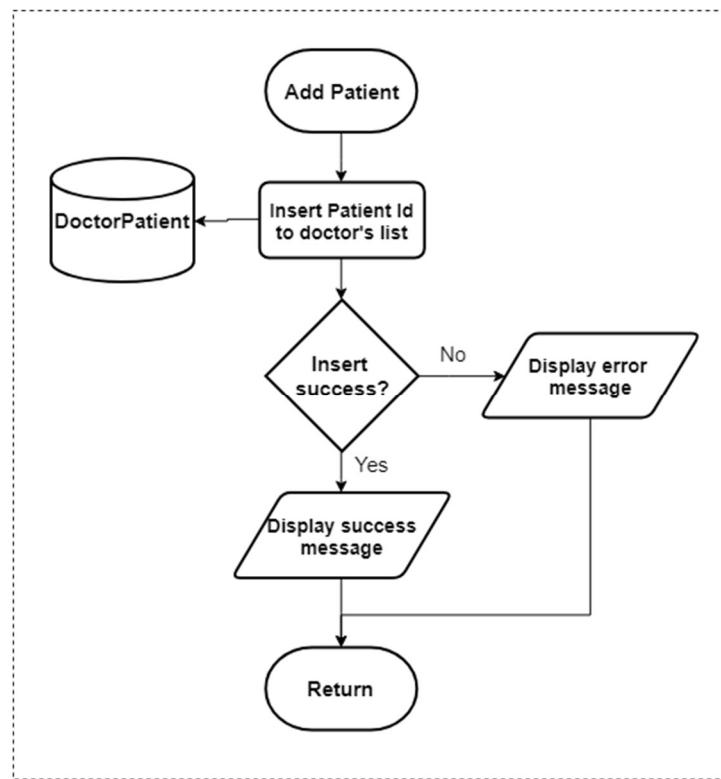


Figure 3.34 Add Patient Flowchart

Figure 3.34 depicts the add patient flowchart. Not to be confused with register patient, this process only adds the patient to the doctor's list of patients. The patient id is needed for this process, and a successful insertion is checked with corresponding messages if it is an error or not.

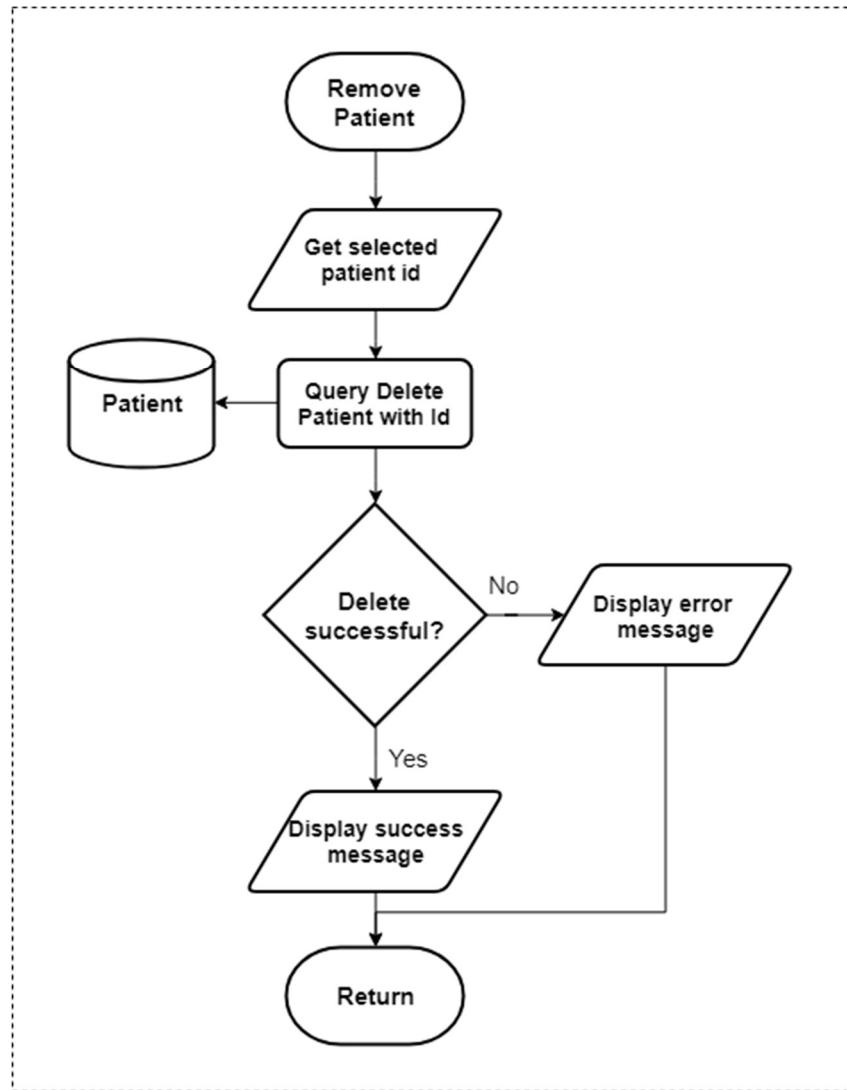


Figure 3.35 Remove Patient Flowchart

Figure 3.35 depicts the flowchart for the removal of a patient in the application. From the doctor's patient's list, a patient can be selected for removal, their patient id is used to issue a query for deletion. In the case of an unsuccessful deletion, an error message is displayed, else a success message is displayed notifying the user of the process done.

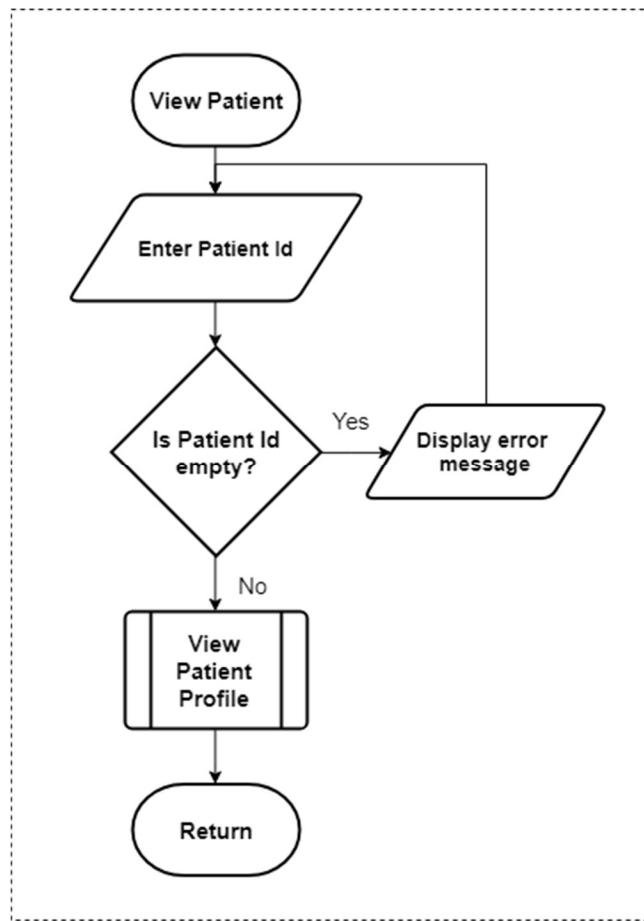


Figure 3.36 View Patient Flowchart

Figure 3.36 depicts the view patient flowchart. The patient's id is entered by the doctor to view the patient, which when valid, redirects them to view the patient's profile. This process is to allow doctors that may want to view a patient's profile but is not in their patient's list because they were not the ones who registered them in the first place. Doctors may share patient ids and allow other doctors to view their patients profile and monitor their status, without the need for sharing mobile equipment.

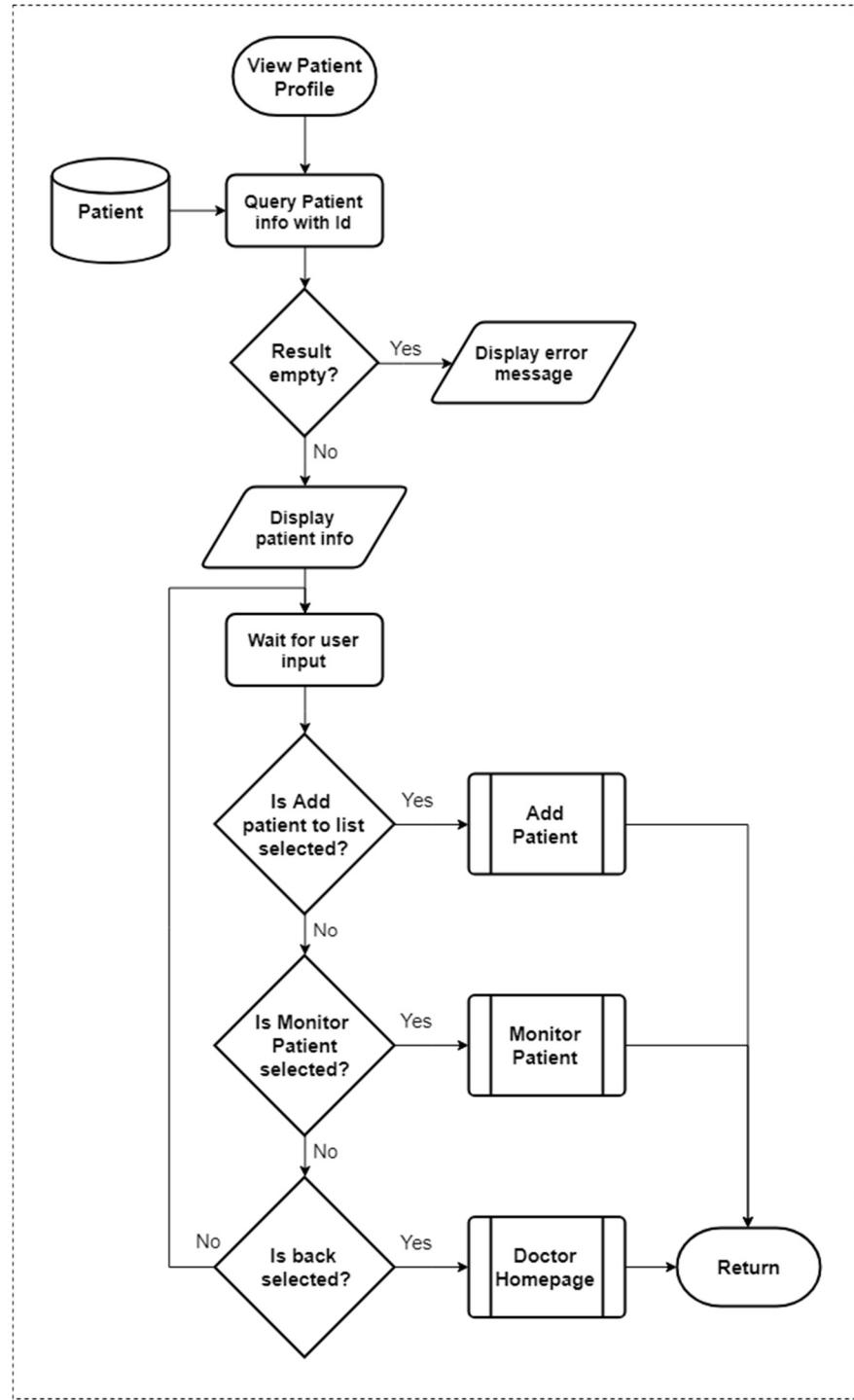


Figure 3.37 View Patient Profile Flowchart

Figure 3.37 depicts the view patient profile. Upon entry to the page, patient information is first obtained from the database to be displayed. Doctors may view a patient's profile, from where they may choose to add the patient to their patient's list. To monitor the patient's status,

they may select monitor patient, and another page will be displayed where they may be able to view all the data acquired by the sensors from the patients as well as results obtained from the neuro-fuzzy model. Selecting the back button will redirect the doctor to their homepage.

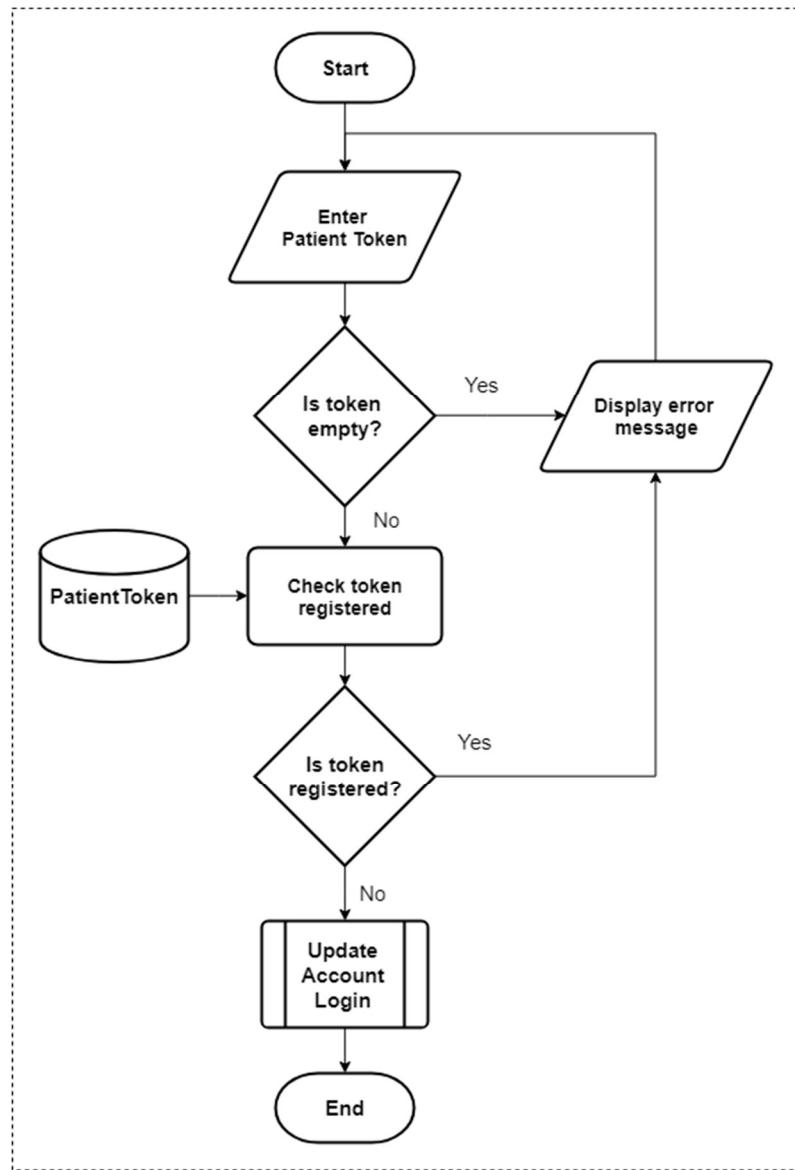


Figure 3.38 Patient Registration Completion Flowchart

Figure 3.38 depicts the completion for a patient's registration. When a doctor registers a patient to the application, a token will be provided to the patient and that token will be used to complete the patient's registration through this process. The patient must enter the token

and it must be checked in the database whether the token is still valid and not yet registered. Error messages will be shown in the case that it is not valid and already registered. Upon identifying that the token is valid, the patient may now update their account login credentials where they will enter the desired username and password. If the update is successful, they may now log in to the application and use it.

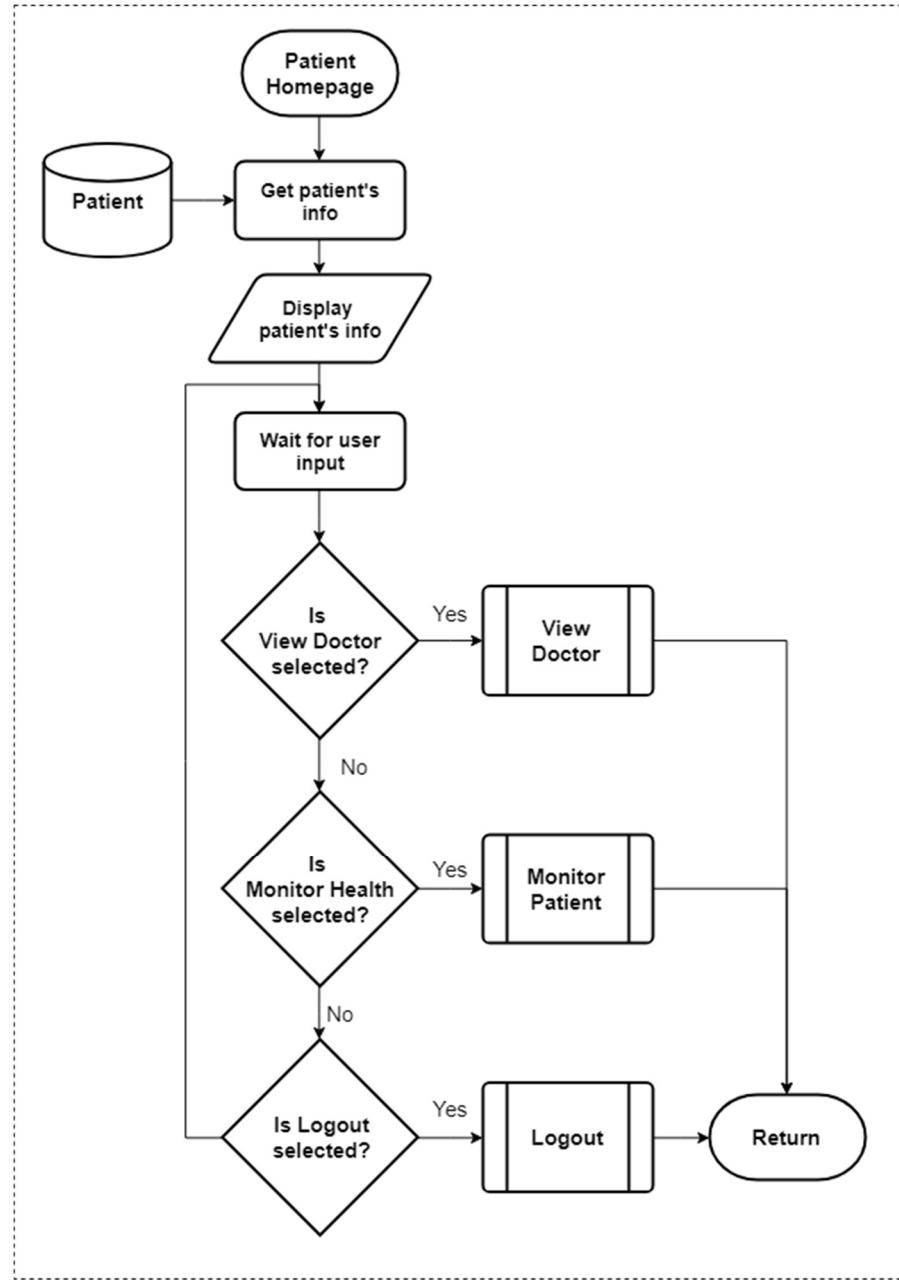


Figure 3.39 Patient Homepage Flowchart

Figure 3.39 depicts the homepage of the patient. When the patient logs in successfully to the application, they will be redirected to their homepage, and their information is first obtained from the database and displayed in their homepage. The patient's capabilities in their homepage is to view their doctor, monitor their status, and logout. Choosing one of the following will redirect them to another page.

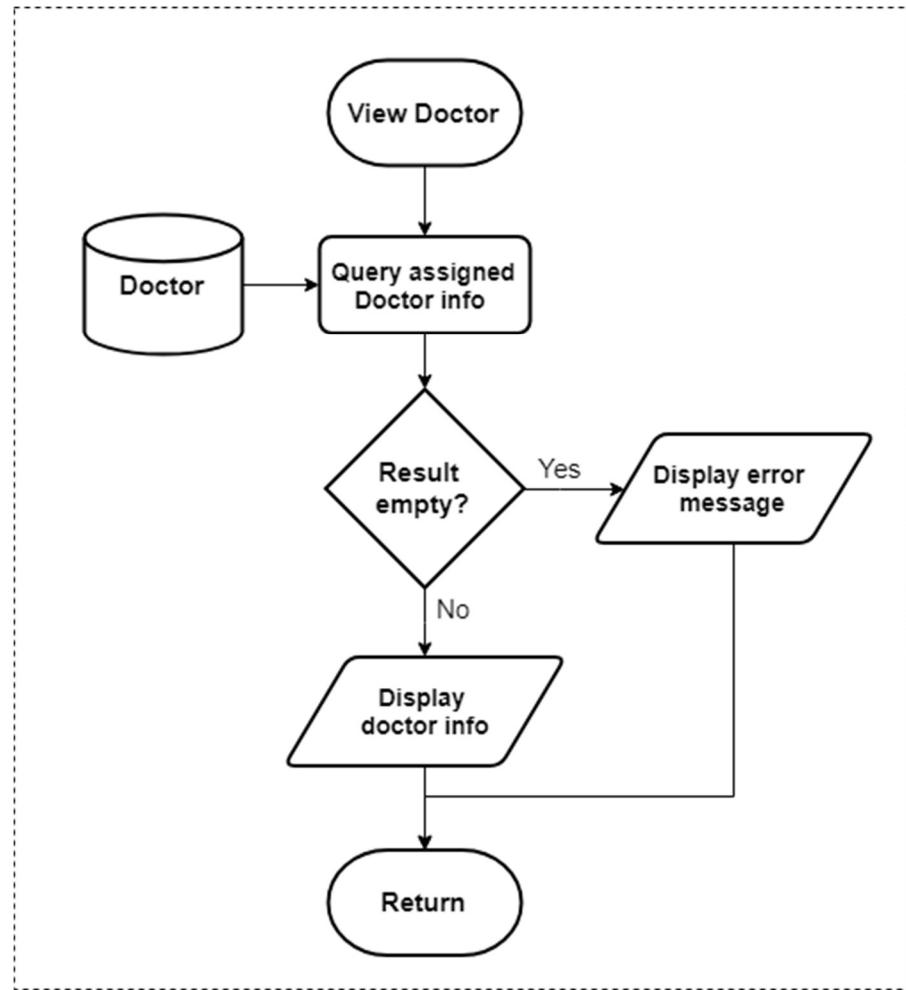


Figure 3.40 View Doctor Flowchart

Figure 3.40 depicts the view doctor flowchart. Patients are registered by the doctors, and the doctor who performed the registration for the patient is assigned to the patient. The patient may view the information of their assigned doctor in this page.

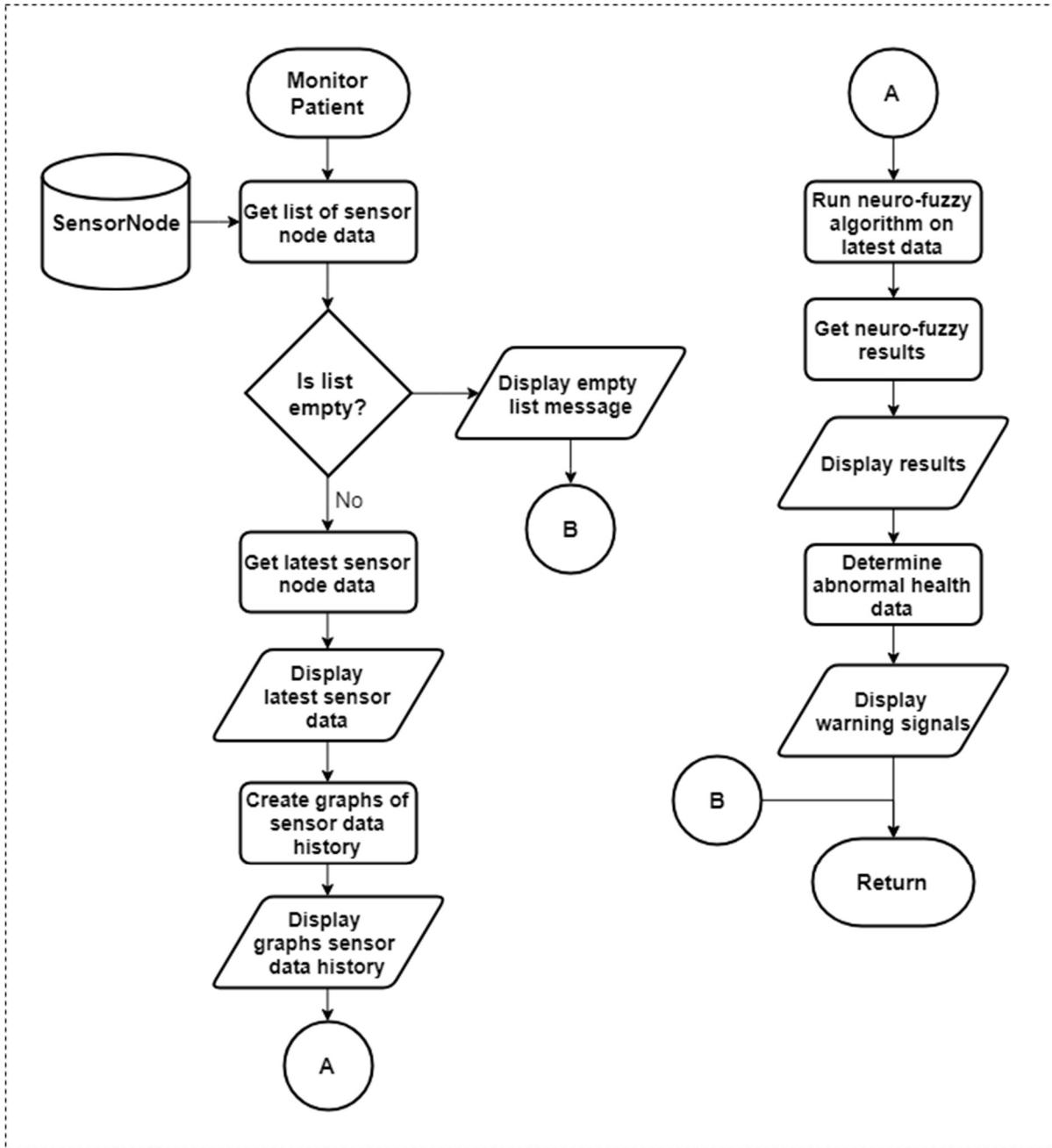


Figure 3.41 Monitor Patient Flowchart

Figure 3.41 depicts the monitor patient flowchart. In this page, all of the information about the physiological parameters measured by the sensors are displayed for the user to view. The doctor and the patient may access this page. A query to obtain a list of sensor data is done to the database and if the list is empty, it is possible that the patient has not yet used the device

resulting in an empty list of sensor data. Each sensor data stored in the database also has a date and time stored which indicates when that certain data was sent to the database for storage. The latest of which is selected from the list of sensor data is displayed. The history of sensor data that is stored in the database is transformed and displayed into a graphical representation to track the changes in the measured physiological parameters of the patient. The neuro-fuzzy algorithm will be used to monitor the PVD status of the patient, and its inputs are the data from the sensors. The latest data obtained will be fed to the neuro-fuzzy model and display the result which is their PVD status. For cases when a patient's physiological parameters have reached values which are unusually large or small relative to the normal values as defined by expert knowledge, warning signals are to be displayed to notify the patient and make them aware that there are possible risks may come due to this abnormality.

Statistical Treatment

To check the accuracy and regularity of the developed system's transmission times, the system readings will be subjected to a single-sample t -test with 80 trials on multiple test subjects. This test will determine if the mean time difference of the actual system readings is close in value to the expected 30-second transmission time of the real-time monitoring system. The null hypothesis of this test is that the difference between the expected and actual mean of the system readings' time difference is small or negligible. Should the null hypothesis be proven correct, this would mean that the system is able to transmit and update the relevant patient data close to or around the expected timeframe.

Table 3.10 Recorded System Readings of the Device with Timestamps

Table 3.11 Single-sample *t*-test Evaluation Table

	Test Value					
	t	df	Significance	Mean Difference	Confidence Interval	
					Lower	Upper
Values						

The correlation of the Pulse Transit Time (PTT) to systolic blood pressure, as well as the correlation of R-Peak Diastolic-Peak Time (RPDPT) to diastolic blood pressure, will be determined. The statistical test that will be used to accomplish this is the Pearson R test, otherwise known as the Pearson correlation coefficient. In this test, the proponents' null hypothesis is that there is strong correlation between the sets of variables that are being tested. Should the null hypothesis be proven correct, this would mean that the variables being used to compute for the systolic and diastolic blood pressures – PTT and RPDPT, respectively – are valid and accurate in consideration of the calibration already performed on the system via linear regression. For greater accuracy of the test, the proponents will conduct at least 100 trials on multiple test subjects.

$$r = \frac{N \sum xy - (\sum x)(\sum y)}{\sqrt{(N \sum x^2 - (\sum x)^2)(N \sum y^2 - (\sum y)^2)}} \quad (3.38)$$

Table 3.12 Table for Pearson R test of PTT and Systolic Blood Pressure

Table 3.13 Table for Pearson R test of RPDPT and Diastolic Blood Pressure

REFERENCES

- American Heart Association. (2018, November 2018). *What is Peripheral Vascular Disease?* Retrieved from American Heart Association Website: https://www.heart.org/-/media/data-import/downloadables/pe-abh-what-is-peripheral-vascular-disease-ucm_300323.pdf
- Billones, R. K., Vicmudo, M., & Dadios, E. (2015). Fuzzy Inference System Wireless Body Area Network Architecture Simulation for Health Monitoring. *8th IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management*. Cebu City, Philippines: IEEE.
- Celinskis, D., & Towe, B. (2014). Wireless Impedance Measurements for Monitoring Peripheral Vascular Disease. *36th Annual Conference of the IEEE Engineering in Medicine and Biology Society*. Chicago, Illinois, USA: IEEE.
- Elgendi, M. (2012). On the Analysis of Fingertip Photoplethysmogram Signals. *Current Cardiology Reviews*, 14-25.
- Fourati, L. C. (2014). Wireless Body Area Network and Healthcare Monitoring System. *IEEE International Conference on Healthcare Informatics*. Verona, Italy: IEEE.
- Hallfors, N., Jaoude, M. A., Ismail, M., & Isakovic, A. (2017). Graphene oxide-nylon Sensors for Wearable IoT Healthcare. *2017 Sensors Networks Smart and Emerging Technologies*. Beirut, Lebanon: IEEE.
- Hirano, H., Horiuchi, T., Hirano, H., Kurita, Y., Ukawa, T., Nakamura, R., . . . Tsuji, T. (2013). Monitoring of Peripheral Vascular Condition Using a Log-Linearized Arterial Viscoelastic Index During Endoscopic Thoracic Sympathectomy. *35th Annual Conference of the IEEE EMBS*. Osaka, Japan: IEEE.
- Infectious Diseases Society of America. (2012). *Infectious Diseases Society of America Clinical Practice Guideline for the Diagnosis and Treatment of Diabetic Foot Infections*. Oxford University Press.
- Janjua, G., Guldenring, D., Finlay, D., & McLaughlin, J. (2017). Wireless Chest Wearable Vital Sign Monitoring Platform for Hypertension. *39th Annual Conference of the IEEE Engineering in Medicine and Biology Society*. Seogwipo, South Korea: IEEE.
- Kar, S., Das, S., & Ghosh, P. (2014). Applications of Neuro-fuzzy Systems: A Brief Review and Future Outline. *Applied Soft Computing*, 243-259.

- Macawile, M., Quiñones, V., Ballado Jr., A., Cruz, J., & Caya, M. (2018). White blood cell classification and counting using convolutional neural network. *3rd International Conference on Control and Robotics Engineering* (pp. 259-263). Nagoya, Japan: IEEE.
- Manimegalai, P., Augustine, S., & Thanushkodi, K. (2013). Determination of Peripheral Arterial Diseases from Toe Brachial Index using Photoplethysmography. *39th Annual Northeast Bioengineering Conference*. IEEE.
- McCarthy, B. M., Mathewson, A., & O'Flynn, B. (2011). An Investigation of Pulse Transit Time as a Non-Invasive Blood Pressure Measurement Method. *Journal of Physics: Conference Series*. 307. 012060.
- Mistal, M., Villaverde, J., & Hadi, B. (2018). Early Warning Device for Brown Plant Hopper Detection in Palay Using Wireless Sensor Networks. *IEEE Region 10 Conference (TENCON)* (pp. 1607-1611). Jeju, South Korea: IEEE.
- Morsi, I., & El Gawad, Y. (2013). Fuzzy Logic in Heart Rate and Blood Pressure. *IEEE Sensors Applications Symposium Proceedings*, 113-117.
- Noche, K., Villaverde, J., & Lazaro, J. (2017). Portable non-invasive blood pressure measurement using pulse transmit time. *9th IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management* (pp. 1-4). Manila, Philippines : IEEE.
- Pittella, E., Pisa, S., Piuzzi, E., Rizzuto, E., & Del Prete, Z. (2017). Combined Impedance Plethysmography and Spectroscopy for the Diagnosis of Diseases of Peripheral Vascular System. *IEEE International Symposium on Medical Measurements and Applications*. Rochester, Minnesota, USA: IEEE.
- Pleo, S. C., Cresencio, J. V., & Alajar, E. (2012). Frequency of Peripheral Arterial Disease Diagnosed by Measurement of the Ankle Brachial Index Among Diabetics with No Symptoms of Intermittent Claudication. *Journal of the ASEAN Federation of Endocrine Studies*, Vol. 27, No. 2.
- Rac-Albu, M., Iliuta, L., Guberna, S. M., & Sinescu, C. (2014). The Role of Ankle Brachial Index for Predicting Peripheral Arterial Disease. *Maedica (Buchar)*, 295-302.
- Rapsomaniki, E., Timmis, A., George, J., Pujades-Rodriguez, M., Shah, A., Denaxas, S., . . . Hemingway, H. (2014). Blood pressure and incidence of twelve cardiovascular diseases: lifetime risks, healthy life-years lost, and age-specific associations in 1.25 million people. *Lancet*, 1899-1911.

- Samonte, M., Mullen, R., Banaga, J., Cortes, W., & Dela Calzada, J. (2018). In-patient medication delivery in mobile app and outpatient online lab results for hospitals. *2nd International Conference on E-Society, E-Education and E-Technology*, (pp. 155-159). Taipei, Taiwan.
- Shabanivaraki, E., Breen, P., & Gargiulo, G. (2015). HeMo: Towards an Inexpensive Wearable Peripheral Blood Flow Monitoring Device. *IEEE Biomedical Circuits and Systems Conference*. Atlanta, Georgia, USA: IEEE.
- Shihabudheen, K., & Pillai, G. (2018). Recent Advances in Neuro-fuzzy Systems: A Survey. *Knowledge-Based Systems*, 136-162.
- Towe, B., Gruber, T., Gulick, D., & Herman, R. (2013). Wireless Microstimulators for Treatment of Peripheral Vascular Disease. *6th Annual International IEEE EMBS Conference on Neural Engineering*. San Diego, California, USA: IEEE.
- Wannenburg, J., & Malekian, R. (2015). Body Sensor Network for Mobile Health Monitoring, a Diagnosis and Anticipating System. *IEEE Sensors Journal*, Vol. 50 No. 12, 6839-6852.
- Yuce, M. R. (2013). Recent Wireless Body Sensors: Design and Implementation. *IEEE MTT-S International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications*. Singapore, Singapore: IEEE.