INTERNET OF THINGS (IOT) BASED PATIENT'S HEALTH MONITORING DEVICE USING RASPBERRY PI

SALINI PRADHAN (1102181006)

THIS PROJECT REPORT IS SUBMITTED TO FULFILL
THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF
BACHELOR OF COMPUTER ENGINEERING WITH HONOURS

SCHOOL OF ENGINEERING

FEBRUARY 2022

DEDICATION

This thesis is dedicated entirely to my adored parents and brother. They have been my strength and courage over my last four years of academic life at MIU. As a result, I'm dedicating this thesis to them for their continuous love and support in helping me accomplish my four-year bachelor's in Computer Engineering degree in Malaysia.

ACKNOWLEDGEMENT

I want to use this time to express my gratitude to everyone who has helped me complete my project. Firstly, my supervisor, DR. Rohilah Sahak, who was extremely helpful and persistent in encouraging me to complete the demonstration flawlessly. I would like to express my gratitude to my coordinator, DR. Fatimah Audah Md. Zaki, and TS. DR. Ahmad Anwar Zainuddin, for their unwavering guidance. I would want to express my gratitude to my parents and brother for their assistance and immense support. Finally, I would thank all the lecturers of Computer Engineering Department and the Head of Department for creating this opportunity and guiding me through these four years for this day.

DECLARATION AND COPYRIGHT PAGE

Name: SALINI PRADHAN

Matrics Number: 1102181006

I hereby declare that this research is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references and a bibliography is appended.

Signature …… …….. Date 28/01/2022

APPROVAL PAGE

INTERNET OF THINGS (IOT) BASED PATIENT MONITERING SYSTEM USING RASPBERRY PI

BY

SALINI PRADHAN

Approved by:

"I hereby declare that I have read this Project and in my opinion this Project is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering"

Supervisor

.... Date:

Co-Supervisor

.... Date: 28/01/2022

TABLE OF CONTENTS

ABSTRACT

Health monitoring is a process of continuously checking the progress of a medical study. Such that, it must be done in conformity with strategy, superior healthcare practices, regulation norms, and standard operating procedures. In some health-care settings, monitoring is crucial. This creates an environment where the doctor and patient can interact. The doctor can also be up to date with the patient's current health status. The sensor initialization is the first part of this system. This project consists of different types of sensors like temperature sensor, and pulse rate sensor. The Arduino Uno is used for the pulse sensor to covert the analogue signals in to digital signals. The sensor is then connected to the Raspberry Pi 3 Model B+ that sends a signal through the internet which allows the doctor interact with the data of the patient. All the data form the sensor is stored in a cloud database. An alert message gets provided to the doctor if indeed the sensor output exceeds the specified sensor readings. The doctor and patient can interact through text message. Due to this procedure, many patients with critical heart conditions can be monitored and time can be saved. This project, provides a low-cost monitoring from one's home itself. This is a safe and reliable project that helps different patients around the world with underlying heart diseases as it is very patient friendly. This system saves so much time for the patient which can save lives. The patient can be monitored completely without the physical presence of a doctor which is a very important factor due to the ongoing COVID-19 situation.

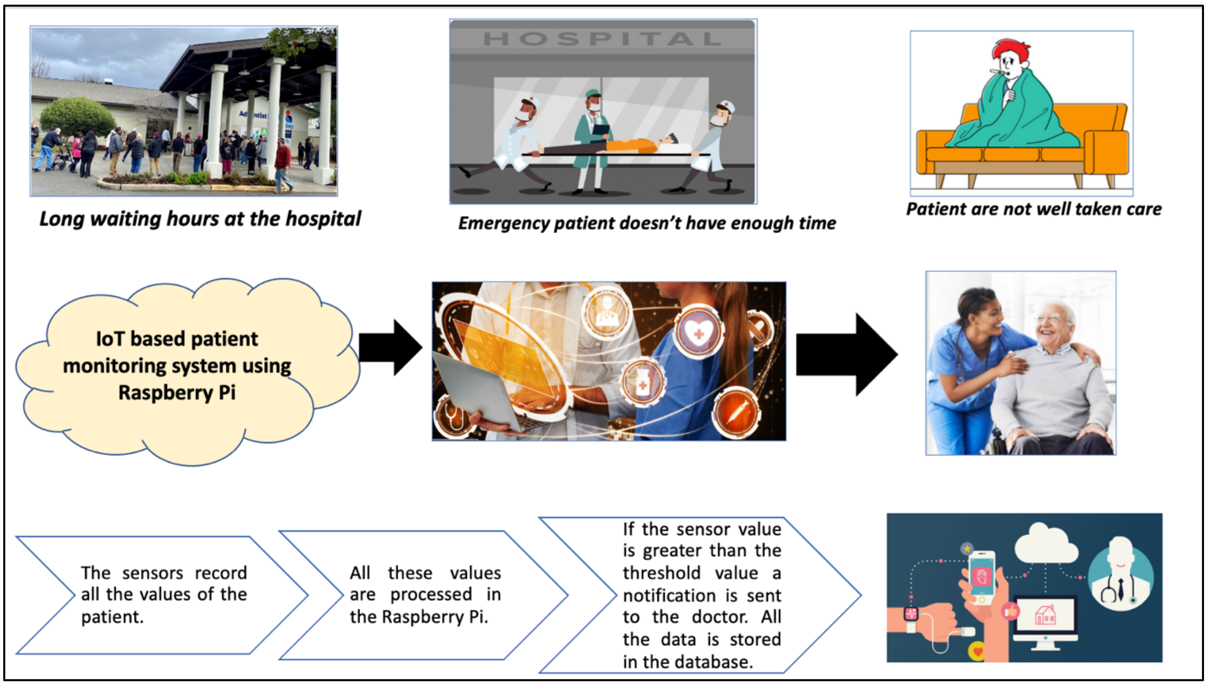

Figure 1: The Graphical Abstract of the IoT based Smart Healthcare device monitoring system.

CHAPTER 1

A. Introduction

The desire for a better existence is inextricably tied to one's physical well-being. Unfortunately, the global medical problem has created a problem as a result of a number more factors, notably poor health care services, significant disparities among cities and suburbs, especially physician and nurse scarcity at key moments(Islam et al., 2020).All machines are interconnected as Internet technologies improve. We can make many things more effective and simpler for human life by advancing technology. The most up-to-date medical technology and communication techniques can assist in lowering healthcare costs(Mathivanan et al., 2018).

IoT has become increasingly important in daily life of people, and IOT technologies have advanced dramatically over the years. IoT is used to connect technology in a variety of industries, such as health monitoring, sensor networks, artificial intelligence, smart homes, and so on, and to transmit the results. The idea behind IoT is to collect reliable data from things and communicate it effectively using existing infrastructure(Vithiya et al., 2019). Real-time analyzation is a vital of an IoT, thus the main motive of this proposed system is to establish a smart health device depending on the robust IoT tools, with the help of a microcontroller known as the Raspberry Pi Model B+ (Hamim et al., 2019).

The sensor nodes on the patient's side are to monitor the patient's symptoms(Jaiswal et al., 2017). The sensors are mostly heartbeat and temperature. The above detectors capture data, which is then transferred to a server. After then, the servers will analyze the monitoring data as well as make it accessible to certain other subscribers, like the cloud and physician orders, in able to locate and check medical problems(Jaiswal et al., 2017).

1.1. Problem Statement

When a patient visits a doctor there is always a long waiting line even in the emergency room to get checked. The developing technology has enhanced the way of the patient being treated. Most of the patients might be in a severe heart condition due to the amount of increasing pollution, eating habits etc. Each and every second are important for the heart patients. To solve this problem, health-care management technology lets clinicians take the proper measurements at the right moment on a patient's health and safeguards them from future problems(Naik & Sudarshan, 2019).

The patient uses the different sensor to measure the heart rate, blood oxygen level, blood pressure and temperature. The data collected from these sensors are stored in the cloud database. If security is not provided to go through the data by the doctor; anyone can prescribe any sort of medicine to the patient. By including security rules such as authentication by ID and password, network systems must confirm if the data created by IoT tools is only accessible by identified users(Kaur & Jasuja, 2017).So that, the patient's life won't be at risk. In the previous system, patients had to go see the doctor every two days for a checkup so that the doctor could properly treat them. However, a real-time doctor is unable to assess the patient's health(Saip & Mohamed, 2018). This health monitoring system helps the patient from saving time to getting updated about his/her health status staying at home.

1.2. Research Objectives

The main objectives of this project are:

- To implement an IoT based patient monitoring device that monitors the temperature of the body and pulse rate and send SMS alert to the doctor if it is higher than the threshold value
- To measure the patient's pulse rate, and body temperature using Raspberry Pi.
- To store patients reading of the sensors in the cloud database using Raspberry Pi.

1.3. Scope of Study

IoT based patient's smart healthcare device, helps enhance the health of the patient by staying at home. Temperature and pulse are the sensors used. The sensor is then linked with the Raspberry Pi Model B+. The Raspberry Pi connects the reading of the patient with the database. This proposed system is detailed with its methodology and key components listed; along with its block diagram, circuit diagram and cloud computing. The flowchart of the research activity along with the project is elaborated. Then, the preliminary data is added with the budget of this project fully described. This project is then concluded with a Gantt chart for FYP 1 and FYP2.

1.4. Hypothesis

More parameters can be sensed and tracked as sensors become available or as biomedical trends evolve, dramatically improving the ability of the wireless detecting devices in biomedical field. In a graphical LCD display, the rate of change of time can be shown with the different health variables. The healthcare device can be developed and enhanced into a small wristwatch. This device can be portable and easy to use. There can be additions of different sensor like ECG sensor, spo2 sensor, accelerometer sensor.

1.5. Research Questions

1. Explain requirements in getting an accurate reading.
2. Evaluate the possible challenges and limitations observed during the use of different kinds of sensor like heart sound, pulse rate and blood pressure.
3. Compute the message to the doctor get sent in the exact time.
4. Calculate the possible ways to have accuracy with the different measurement.

5. Explain if sensor collects the data and sends it to the cloud database.

1.6. Limitations of the Study

In this project, the limitations are that the threshold value of the patient cannot be shown in real time. The patient being sick cannot be shown as the probability of them getting sick is very low and one can never know when a person can get sick. So, this cannot be shown in reality but one can increase the temperature by using a boiling water or very warm cloth. The pulse rate of the patient can go up even while running or jumping here and there. So, this cannot be applicable for people if the they brisk walk of run around. For temperature sensor, it is for patient who have different kinds of diseases that bound fall them sick. So, the doctor can monitor their day-to-day progress like a condition like covid, influenza, SAR etc. For the pulse sensor, it is for patients who have underlying heart disease who have been kept into supervision. This can be used as when the alert notification is sent to the doctor the doctor can immediately send a message or come and see the patient in person.

CHAPTER 2

a. Smart Healthcare Monitoring System

In hospitals, where a patient's condition must be monitored on a frequent basis, this is normally done by doctor or other paramedical personnel who constantly monitors critical metrics including body temperature, heartbeat, and blood pressure. As a result, after some time, this task becomes tedious. Many studies have attempted to communicate patient data from a sender device to a receiver device using SMS (shot message service) with GSM (global system for mobile communications) or RF (radio-frequency) module. Furthermore, the patient's history is not provided in these circumstances; only current data is provided. So, the goal of this project is to continually monitor the data of the patient and, if the sensor value is greater than the threshold value an alert message is sent to the doctor. The Raspberry Pi links the senor with the cloud database. All the sensor data will be able to get stored in the cloud database and make it accessible from anywhere in the globe with the help of the Raspberry Pi. As a result, clinicians will be able to view the patient's medical history at any time and from anywhere. Designers will be able to monitor patients remotely via mobile or PC (Muqeet & Quadri, 2019).

As a result of our fascination with the use of IoT and the work of other researchers, we will construct a prototype to monitor the patient's health condition locally and also from a webpage that can be managed from a mobile phone or PC. In this proposed system, Raspberry Pi Model B+ was used. We can verify these values from anywhere on our

Smartphone by viewing this webpage from either the Smartphone or the PC. Raspberry Pi contains Wi-Fi from within those helps to communicate with the internet and the microprocessor. The patient's data is sent to a webpage via the cloud using a Raspberry Pi 3 in this project. This program displays the current status of the patient's health whenever we open it. They kept the gear with inbuilt Wi-Fi coupled to a sensor i.e., temperature, heartbeat on patient's side. The main processor utilized in this project is the Raspberry Pi 3, and communication between the internet and our hardware is established utilizing inbuilt Wi-Fi(Muqeet & Quadri, 2019).

b. Emerging of a Smart Healthcare Monitoring System in IoT Environment

The Internet of Things enables devices to be sensed and controlled remotely through network infrastructure, resulting in Direct Interaction with sensors over network architecture, which improves system accuracy and reliability while reducing human interaction(Saip & Mohamed, 2018). As the physical world is incorporated into computer-based systems, the Internet of Things presents enormous opportunity. As a result of this integration, human engagement is decreased, efficiency is increased, and accuracy is improved. Keeping a patient under close observation and monitoring vital indicators such as pulse rate, blood pressure, and core temperature is a critical occurrence in healthcare monitoring systems (Ahmed et al., 2018).

IoT sector is progressing to cut the price and patient experience while also ensuring that patients receive appropriate medications and live a long life. Undiagnosed health risks can be handled in traditional health care with IoT, that assures medical care through keeping a unique identification for every person, lowering the risk of issues. Through connectivity in between medical sensing element and your desktop or cellphone, that can interface using your host via configuration, decreases overall total price also minimizes time consumption (Ganesh, 2019).

c. Opportunities and Difficulties of a Smart Healthcare Monitoring System

The volume, velocity, variety, veracity, and value are frequently used for the explanation of big data. The quantity of data produced is referred as volume, while rate at which it is produced is referred to as velocity. Variety is known as the wide range of the types of data available, whereas veracity is known as the unpredictability of what types of data that may be introduced in the future. Likewise, value is known as the quantity of data that can be obtained from a huge set of data. The cloud database is beneficial for preserving enormous quantities of data in a way that allows value to be extracted. In this approach, both clinical symptoms as well as the regularity of doctor appointments were tracked in the patient's medical records. The information can subsequently be used by algorithms to evaluate the health diagnosis(Baker et al., 2017). Cloud technologies can handle a variety of data processing tasks, but computational offloading and machine learning are the most important. Complex data processing is sent to the cloud via computational offloading, which is beyond the capabilities of reduced wearable technology. By sending raw or partially processed sensor data to the cloud, the computing capacity of multiple devices can be utilized for processing. Using such a heavy computer network rather than loading on a singular smartphone has several benefits, including the ability to run more advanced analytics, produce results much quicker, as well as lengthen mobile device power consumption due to less embedding. Complex sensor nodes that quantify Emg signal, sugar levels, or accelerometers for object tracking would benefit greatly from parallel computing unloading. ECG signal, for example, have a consistent structure, and deviations from it can indicate a variety of heart problems, such as heart failure, cardiac infection, and even heart attack.

The challenges with the Smart Healthcare Monitoring System are just that IoT offers for more mobility; for example, if a patient requires constant care, he or she can stay home instead of in a clinic and also be watched on something like a regular schedule using IoT technology. Certain smartwatches, such as sensors, irritate the patient's skin(Selvaraj & Sundaravaradhan, 2020). Sound will impact the data sent to the sensor towards the control device as well as ultimately to the monitoring center, reducing the data accuracy. A improved structure enables data transport without jeopardizing overall authenticity. The adoption using noise-reduction equipment can also help enhance information transmission quality. The bulk of existing ECG monitoring devices involve a guided signal analysis. This increases the expense and increases the risk of a detection error. Signal analysis can be done with machine learning, resulting in better productivity as well as decreased costs. As even the amount of sensing devices grows, the amount of energy needed to operate these grows, resulting in increased power consumption and energy leakage. An optimization approach could be used to reducing the power consumption.

2.1 Literature Overview

i. *A Personal Healthcare IoT System model using Raspberry Pi 3*(Yattinahalli & Savithramma, 2018)

   This project is a system that updates the cloud data and gets the location information of the area that one wants to visit. This location-based system supports patients in identifying their location and protects pollutant-sensitive clients by offering and leading them to medical centers and specialist credentials. Then, the doctor monitors the patients' health likewise patient enter his/her location. The data is updated in the cloud server every time a new patient comes.

ii. *An IoT based patient monitoring system using raspberry Pi* (Kumar & Rajasekaran, 2016)

   In this project, the system measures the temperature, movement, heart beat and respiration of the patient using all the required sensor. The data is sent to the Raspberry Pi through a signal. The Microcontroller Board is programmed via the internet to monitor the health of patients. After that, all the data is collected in the web database. The doctor or any caretaker can retrieve the information from the web database.

iii. *Automatic health monitoring system using raspberry pi.* (Prabha, n.d.)

   This system consists of a heart beat sensor and temperature sensor which measures the patient. This system contains Raspberry Pi B+ to store the acquired data from the patient. The retrieved data is shown in the LCD display. If the data is not in the normal range the alarm gets triggered. GSM technology is used to send the stored values to the server. The most recent data is shown in the webpage. The doctor logins with his username and password to see the patient’s data. Doctors have access to all of a patient's previous medical records and can recommend medications and prescription adjustments. To access personal health history, patients are also provided a unique login credentials.

iv. *IoT Based Health Monitoring System Using Raspberry Pi* (Gutte & Vadali, 2018)

In this system, it provides technical support to improve healthcare systems in an easier and faster way. This system collects a precise physical parameter of a patient and likewise makes required data available on the internet. All the data that is taken from the sensor is stored int the database through the raspberry. The outputs are also displayed in LCD. This speeds up the diagnosis and treatment of the patient. If the patients reding is greater than the sensor threshold value. Then, an SMS alert is sent to the doctor and a precaution notification is sent to the patient.

v. *Iot Based Health Monitoring System Using Raspberry PI – Review* (Rohit & Tank, 2018)

Different sensors, including as a heartbeat, body temperature , an ECG , a blood pressure , and a patient position, are mounted to the patients in this design, and the output signal is sequentially sent to the Raspberry Pi. On the Raspberry Pi, various signals are received consecutively or concurrently. Readings are displayed in the Raspberry Pi program window, and the readings of all these sensors can be checked by a specialist via the Ports connected of a Raspberry Pi connected to the internet. The Raspberry Pi is a payment card-sized device powered by an Embedded controller. The Raspberry Pi is a little portable laptop that could be carried with you wherever you go.

vi. *Healthcare Monitoring System and transforming Monitored data into Real time Clinical Feedback based on IoT using Raspberry Pi* (khan et al., 2019)

Personalized healthcare systems must be portable, hence compact, light, and low-power equipment should be used. This structure consists of 1 Raspberry Pi minicomputer that is inexpensive and easy to use, as well as a small Arduino microcontroller board. ECG and pulse sensors detect analogue data, while temperature sensors detect digital data. The Arduino Microcontroller receives the sensing data via its input and output interfaces, which is then sent to the Raspberry Pi via the USB cable which connects two systems. Recording is done with a 5MP Raspberry Pi webcam that is securely linked to the Raspberry Pi.

vii. *Smart Healthcare Monitoring System Using Raspberry Pi on Iot Platform* (Naik & Sudarshan, 2019)

In this project, the following system structure discusses the relationship between the various elements. This system is divided into two components. The hardware unit is made up of transmitters and receivers, whereas the software unit is made up of programming languages like Python and MATLAB, as well as their interfaces. In this article, we discussed some relevant IoT applications for health monitoring. The simple functioning steps of an IoT application Collecting data, processing data, storing data, and transferring data are all steps in the data collection process. Although each program may handle the initial and end steps, storage does not apply to all or even some apps.

viii. *Health monitoring systems using IoT and Raspberry Pi—A review* (Pardeshi et al., 2017)

The fundamental functioning phases of an IoT system include data collecting, data interpretation, data storage, and data transfer. Every application has the initial and last stages, although some applications may or may not have the processing and storage stages. Real-time direct data communication, raw file transfer, and real-time on-board operations are all part of the information process of collecting. MEMS technology can lower the amount of energy used for data acquisition. Several Iot systems have the imbalanced data attribute, attempting to take use of the compressed sensing paradigm. Compacted sensing has been extensively examined and researched in health systems and wearable monitoring linkages.

ix. *IoT-based Health Monitoring System with Medicine Remainder using Raspberry Pi* (Amru et al., 2020)

The pulse and temperature sensors such components are used to monitor the patient's pulse and temperature independently. This device will send a pre-programmed IoT notification to the doctor whenever any of these variables surpasses its predefined threshold. He is constantly analyzing the patient at the end of the day. The Internet of Things (IoT) concept is used in this project. The user is accommodated by the crisis transition associated with the fundamental module. If somebody believes they needs professional help right now, they can push this switch, and the Raspberry Pi will receive the signal and send the predefined message to the specialist.

x. *Monitoring System Heartbeat and Body Temperature Using Raspberry Pi* (Sollu et al., 2018)

In this project, the patient's BPM is first read. To the Raspberry Pi, there is Serial Communication. All of the information is kept in a database and shown on an LCD panel. The Raspberry Pi retrieves the sequential information from Arduino. The Microcontroller Board was also used to capture body temperature data. Acquire a unique identifier from a transportable or desktop pc. The Raspberry Pi module holds the results of patient’s information processing as well as object tracking reports. This module is responsible for storing and transmitting patient information to mobile or smartphone devices.

Table 1 describes the literature review of the different paper. These paper all give background research of the developed paper itself. The author's name, the title of the publication, the research design, the equipment utilized, the key findings, the study's limitations, and future research are all covered. The major discoveries, study limitations, and research gaps are all interconnected. All the papers are analyzed critically to get the results. Table 1 emphasizes of the main points of the all the reach paper.

2.2 Literature Review

S. No.	Author(s) & Year	Title of Research Paper	Variables Studied/ Research Design	Equipment/ Instruments/ Apparatus used for Experiments/Analysis/ Characterization, etc.	Important Findings	Limitations of Study	Research Gap/ Novelty of Research Study
1.	(Yattinahalli & Savithramma, 2018)	A Personal Healthcare IoT System model using Raspberry Pi 3	Availability of physician Availability of hospitals nearby	Raspberry Pi 3, Cloud computing, Internet of things	The system concept includes sensors, databases, gateway devices, medical accessibility and hospital data to help the health-related IoT network patrician.	There are no notification services available.	GSM module can be installed.
2.	(Kumar & Rajasekaran, 2016)	An IoT based patient monitoring system using raspberry Pi.	Rate of respiration of the patient Temperature of the patient Rate of heartbeat of the patient, Rate of acceleration of the patient	Raspberry Pi board Respiration Sensor Accelerometer Sensor Temperature Sensor Internet of things	Patients can record their health status in their own mobile phone and then store the information using this technology advancement.	There are no provisions for a checkup.	Check of provisions can be developed.
3.	(Naik & Sudarshan, 2019)	Smart Healthcare Monitoring System Using Raspberry Pi on Iot Platform	patient's body temperature, ECG, heart rate parameter, blood pressure, Electrical activity of the heart, Rate of movement	temperature sensor, BP sensor, heart rate sensor, an ECG sensor, acceleration sensor, raspberry Pi with GSM	A doctor can review a patient's medical history and recommend medicine and prescription adjustments. Patients with special IDs and passwords have access to their records.	Data transmission is contingent on the presence of a smartphone.	Build up a website for the data transmission.

4.	(Prabha, n.d.)	Automatic health monitoring system using raspberry pi.	Rate of respiration of the patient, Temperature of the patient, Rate of heartbeat of the patient, Rate of acceleration of the patient	Raspberry Pi board, Respiration sensor, Accelerometer sensor, Temperature sensor,	The alarm system, which comprises of a buzzer and LED, warns the doctors when the threshold value is reached.	There are no notification services available.	GSM module can be installed.
5.	(Kirankumar & Prabhakaran, 2017)	Design and implementation of low-cost web based human health monitoring system using Raspberry Pi 2	Blood pressure measurement of the patient, Heart beat rate of the patient, Alcohol detection, Electrical activity of the heart	Raspberry Pi 2, Blood pressure machine, Heart beat sensor, Alcohol sensor, EMG sensor, Sound sensor, ECG sensor, Video camera	The system has two modes of operation: adult patient monitoring and infant monitoring. The acquired data can be seen locally as well as globally over the internet.	There are no notification services available.	GSM module can be installed.
6.	(Gutte & Vadali, 2018)	IoT Based Health Monitoring System Using Raspberry Pi	Heart beat rate of the patient, Temperature of the patient, Electrical activity of the heart	Temperature sensor, Pulse rate sensor, ECG sensor Raspberry Pi Module,	The suggested system uses IoT devices to monitor the health of older individuals and store the acquired data on an IoT server.	Built only for elderly people. There are no provisions for a checkup.	This can be focused on all types of patients.
7.	(Gupta et al., 2015)	Healthcare based on IoT using Raspberry Pi	Electrical activity of the heart, Heart beat rate	ECG sensor Raspberry Pi Module, GSM Module, Buzzer	If the heart rate is out of the usual range, send an SMS to the designated person via the GSM module, and notify the hospital via a buzzer sound.	Only a few parameters are being tracked.	More parameter can be added such as temperature sensor, blood pressure sensor etc.
8.	(Mehta et al., 2018)	IoT Based Patient Health Monitoring System	Temperature, ECG, Pulse Rate, Spo2 level	Heart pulse sensor, temperature sensor, ECG	Any changes in the patient's health are promptly identified, and the doctor is notified	Only a few parameters are being tracked.	More parameter can be added such as accelerometer sensor , blood pressure sensor etc.

				sensor, spo2 sensor, Raspberry Pi	through SMS using the GSM module.		
9.	(Rohit & Tank, 2018)	Iot Based Health Monitoring System Using Raspberry PI - Review	Electrical activity of the heart, Heart beat rate, Temperature, blood pressure, Position of the patient	Pulse/Heart beat sensor, ECG sensor, Body temperature sensor, Blood pressure sensor, Patient position sensor, Raspberry Pi	This entire health monitoring system fits into a small compact gadget the size of a cell phone or wrist watch.	Monitoring isn't personalized for each patient.	The monitoring can be personalized to each patient.
10.	(Sankaran et al., 2020)	Design of IoT based Health Care Monitoring Systems using Raspberry Pi: A Review of the Latest Technologies and Limitations	Body Temperature, Fetus movement, blood pressure, Patient movement	temperature sensor, blood pressure sensor, Accelerometer sensor, fetus movement, Raspberry Pi	Sensor signals that are sent the raspberry pi can accurately check where the fetus is present at the current moment.	Only a few parameters are being tracked.	More parameter can be added such as temperature sensor, blood pressure sensor etc.
11.	(khan et al., 2019)	Healthcare Monitoring System and transforming Monitored data into Real time Clinical Feedback based on IoT using Raspberry Pi	Heart beat rate, Body Temperature, Electrical activity of the heart	Raspberry pi 2, heartbeat sensor, ECG sensor, temperature sensor, pi camera, Arduino Nano (used as ADC), Power supply, LCD, Key board, Wi-Fi Dongle	The method can be made more patient-friendly by having a video interaction between the patient and the doctor. When patient sensor data approaches abnormalities, the GSM/GPRS module can be used to generate an SMS alarm.	Monitoring isn't personalized for each patient.	The monitoring can be personalized to each patient.
12.	(Amru et al., 2020)	IoT-based Health Monitoring System with Medicine	Heart beat rate, Body Temperature,	Heart beat sensor, temperature sensor, raspberry pi, RTC (DS1307), a buzzer and	In regard to genius urban inhabitants, the key components of urban shrewd regions are coordinated with knowledgeable home human	Only a few parameters are being tracked.	More parameter can be added.

		Remainder using Raspberry Pi		LCD monitor, APR9600, speaker and IR receptor	services, such as high-prescription, clever homeland, intelligent living, and safety.		
13.	(Pardeshi et al., 2017)	Health monitoring systems using IoT and Raspberry Pi—A review	Heart beat rate, Body Temperature, blood pressure, Electrical activity of the heart	temperature sensor LM-35, blood pressure sensor, heartbeat sensor, ECG sensor, raspberry pi and GSM module.	Any irregularities in the health conditions can be directly detected and communicated to the individual using GSM technology or the internet.	Data transmission is contingent on the presence of a smartphone. Notification that only goes one direction (patient to doctor).	Build up a website for the data transmission
14.	(Sollu et al., 2018)	Monitoring System Heartbeat and Body Temperature Using Raspberry Pi	Heart beat rate, Body Temperature	Heart beat sensor, Body temperature sensor, LCD display, Raspberry pi, Arduino UNO, PC	Any patient data contained in the database can be retrieved using the id number or the name of the patient.	There are no notification services available.	GSM module can be installed.
15.	(Neyja et al., 2017)	An IoT-Based E-Health Monitoring System Using ECG Signal	The path patient has travelled, Result of the ECG of the patient, Alert signal of the hospital, Electrical activity of the heart	Electrocardiogram (ECG) sensors Hidden Markov Model (HMM) Predictor Hospital alert management Patient Table Management	A system for the installation of the IoT ecosystem to allow an ECG signal to be monitored in instantaneously and foreseen in the following path for the patient with CVD.	Only a few parameters are being tracked.	More parameter can be added such as temperature sensor, blood pressure sensor etc.
16.	(Rahman et al., 2019)	IoT Based Patient Monitoring System Using ECG Sensor	Heart beat rate, Body Temperature	Raspberry Pi 3 Model B, Raspberry Pi Camera Module, Arduino Uno, 20x4 LCD display,	Unless the ECG signals or temperature measurements exceed or fall below target value, the system will send an	Hardly a few metrics are monitored.	More parameter can be added such as ECG sensor, blood pressure sensor etc.

				temperature sensor, pulse rate sensor	electronic text message to the physicians or families.		
17.	(Vineetha et al., 2020)	A real time IoT based patient Health monitoring system Using machine learning algorithms	Heart beat rate, Body Temperature, humidity detector, bomb detector, rate of acceleration	temperature sensor, humidity sensor, pulse sensor ECG module GPS module LoRa WAN module bomb detector, accelerometer, Zig Bee module, raspberry pi	It is highly beneficial for armed troops during conflicts and rescue operations because it may be used without a network.	Limited checkups available.	This system can be modified into full body monitoring system.
18.	(Ganesh, 2019)	Health Monitoring System using Raspberry Pi and IOT	Blood pressure, heart sound, pulse rate	Heart beat sensor, Heart sound sensor, blood pressure sensor, raspberry pi 3 Model B, Pi camera, Monitor	The system provides efficient medical care for patients, and the information gathered is networked globally via the internet, which are then linked to cloud storage, so that doctors may use this data to provide a prompt and efficient solution.	Only a few parameters are being tracked.	More parameter can be added such as temperature sensor, ECG sensor etc.
19.	(Hamim et al., 2019)	IoT Based Remote Health Monitoring System for Patients and Elderly People	Heart beat rate of the patient, Temperature of the patient, Galvanic Skin Response (GSR)	Heart beat sensor, Body temperature sensor, Galvanic Skin Response (GSR) sensor, Raspberry Pi, Arduino Uno	The goal of this project was to create a long-term solution. Patients can use a continuous patient monitoring system.	Hardly a few metrics are monitored.	A larger number sensors, such as respiration rate, might be integrated monitor, blood sugar detector, and heart rate detector
20.	(Jaiswal et al., 2017)	IoT-cloud based framework for patient's data collection in smart healthcare	Temperature of the patient, Electrical activity of the heart, Blood pressure	Raspberry Pi, Electrocardiogram (ECG) sensors, Body temperature sensor	Our present study is focused on how information is connected with an IoT-based public health system that used a	There are no notification services available.	GSM module can be installed.

		system using raspberry-pi			Raspberry Pi and a Docker container.		
21.	(Kaur & Jasuja, 2017)	Health monitoring based on IoT using Raspberry PI	Temperature of the patient, Heart beat rate of the patient	Body temperature sensor, Heart beat sensors, Raspberry Pi, Arduino UNO, Node-RED, MQTT protocol	This platform employs a single-board minicomputer Raspberry Pi with IBM Bluemix cloud, as well as the MQTT protocol for dependable applications.	Limited checkups available.	This system can be modified into full body monitoring system.
22.	(Mathivanan et al., 2018)	IoT based continuous monitoring of cardiac patients using Raspberry Pi	Heart beat rate of the patient, Electrical activity of the heart	ECG sensor, Heart beat sensors, Raspberry Pi, Arduino UNO	By utilizing data analytics, it creates potential and improves the quality of care provided by providing modern technology.	Hardly a few metrics are monitored.	A larger number sensors, such as respiration rate, might be integrated monitor, blood sugar detector, and EEG detector
23.	(Saip & Mohamed, 2018)	Smart Health Monitoring and Controlling using Raspberry Pi	Heart beat rate of the patient, Temperature of the patient, Blood pressure	Body temperature sensor, Heart beat sensors, Blood pressure sensor, Arduino Uno, Bluetooth module, Motor driver, Raspberry Pi,	The system receives information from the patient such as temperature, blood pressure, and pulse rate and displays it on the GUI.	There are no notification services available.	GSM module can be installed.
24.	(Muqeet & Quadri, 2019)	IoT based Patient Monitoring System Using Raspberry Pi	Heart beat rate of the patient, Temperature of the patient	Raspberry Pi 3 Microprocessor, PIC Microcontroller (PIC16F72), Heartbeat sensor, Temperature sensor (LM35), LCD with the driver board, Power supply adapter	This system is to constantly monitor patient data and, if necessary, to issue an emergency button, utilizing various technology such as the Internet of Things (IoT)	The programming language is not that advanced programming interface. Hardly a few metrics are monitored.	This intelligent operation can be carried out with the help of the Embedded LINUX programming language. In the future, we may be able to integrate a variety of many other health-related sensors, such as ECG and EMG sensors.

25.	(Yadavalli et al., 2020)	Secured IoT Based Health Monitoring System	Respiratory tracker, Electrical activity of the heart, Heart beat rate of the patient, fall detection tracker	Respiratory sensor, ECG Sensor, Heartbeat sensor, fall detection sensor, Raspberry Pi, GSM kit	Data security is offered, and this technology can be installed in clinics, allowing for large amounts of data to also be acquired and safeguarded in a cloud-based system.	Data transmission is contingent on the presence of a smartphone. Notification that only goes one direction (patient to doctor).	Build up a website for the data transmission
26.	(Vithiya et al., 2019)	Detection, Monitoring and Tracking of Survivors under Critical Condition Using Raspberry-Pi	Pulse rate, Temperature of the patient, Gas detection, tracking the GPS	Heartbeat sensor, GP sensor, Gas sensor, Arduino controller, Bluetooth module, Body temperature sensor, Raspberry Pi 3	This system is sensing the temperature, heartbeat, and neighboring dangerous chemicals in the nearby region, as well as the soldiers' precise address, and presenting the situation on either a web page	Limited checkups available.	This system can be modified into full body monitoring system.
27.	(Singh et al., n.d.)	IoT Based Health Monitoring System for Persons with Intellectual Disabilities	Temperature of the patient, Pulse rate, uses microscopic metal discs to monitor electrical activity in the brain	Arduino Uno, Temperature sensor, LM35 Pulse sensor, EEG sensor, Bluetooth module HC 05, Raspberry Pi 3	The mentally handicapped patients should be monitored on a regular basis, and their loved ones should be kept up to date on their health state while they are at work.	There are no notification services available.	GSM module can be installed.
28.	(Chakravorty et al., 2018)	IoT Based Patient Guidance System using Raspberry Pi	Temperature of the patient, Electrical activity of the heart, Blood pressure, pulse rate	Raspberry Pi 3, Electrocardiogram (ECG) sensors, Body temperature sensor, one touch switch, LCD display,	It is to create a close to zero, reduced, dependable, non-intrusive, as well as non-invasive heart rhythm monitoring that receives and analyzes sensor results to discover if it is within such a "normal" range and transmits such data to the user's cell phone utilizing.	There are no notification services available.	GSM module can be installed.

29.	(Pandey & Chinnamuthu, 2018)	IOT Based Patient Monitoring System utilizing Raspberry Pi and Web-Page	Temperature of the patient, Electrical activity of the heart, Blood pressure	Raspberry Pi 3, Electrocardiogram (ECG) sensors, Body temperature sensor, Temperature sensor, GPRS Module, LCD display, Buzzer alarm,	This is an auto-running mode architecture, we should not be using Linux OS to run the application. All we should do is connect the Raspberry Pi to electricity, and it will restart at a certain time. Insert the SIM card further into GSM module then activate the component's capability.	Hardly a few metrics are monitored.	Include additional factors for screening not just the patient's health but also the Satisfaction with Life Index, such as EEG (Electro-encephalogram), stickiness sensors, and richness checking sensors.
30.	(Mathew & Abubeker, 2017)	IoT based Real Time Patient Monitoring and Analysis using Raspberry Pi 3	Temperature of the patient, Electrical activity of the heart, Blood pressure	Raspberry Pi 3, Electrocardiogram (ECG) sensors, Body temperature sensor, Temperature sensor, GSM Module-SIM 800C	Those parameters can be accessed and diagnosed by a health professional everywhere in the world. Whether there is any modification in all these parameters, a notification should be sent to the pre-defined number as well.	Data transmission is contingent on the presence of a smartphone. Notification that only goes one direction (patient to doctor).	Build up a website for the data transmission

Table 1: Literature Review

CHAPTER 3

3.1. Description of Methodology

This section explains the research methodology that has been used for this project. The initial task of an IoT based smart healthcare monitoring system, according to the workflow, is to collect data from patients via sensors. Temperature and heartbeat rate are examples of health parameters. Low-power sensors are placed in such systems. The Arduino Uno converts the analogue signal from the sensors to digital signal. The Arduino prints to the serial port taking the analogue signal from the both sensors The digital signal is then processed into the Raspberry Pi. All the data that has been read and was stored in the firebase database. The Raspberry Pi is programmed to store all the readings of the sensor in the firebase Realtime database

They gather information from patients on a regular basis. The health state of a particular patient is observed using this often-collected data, and necessary medicines are given. This data is displayed on a Raspberry Pi firebase Realtime database, and if the reading is done then and there, it is processed and displayed in real time firebase database. The information is then moved to a Firebase database for storage before even being made available to the system's targeted clients. As a result, the client-server model of computer networking is used in this system. All the data collected just on Firebase database is stored and given to all peers in the system's reference as required. (Gutte & Vadali, 2018). Figure 2 shows how the

Smart Healtcare system works with the basic steps.These steps are implemented to develop this Smart Helathcare Device using Raspberry Pi.

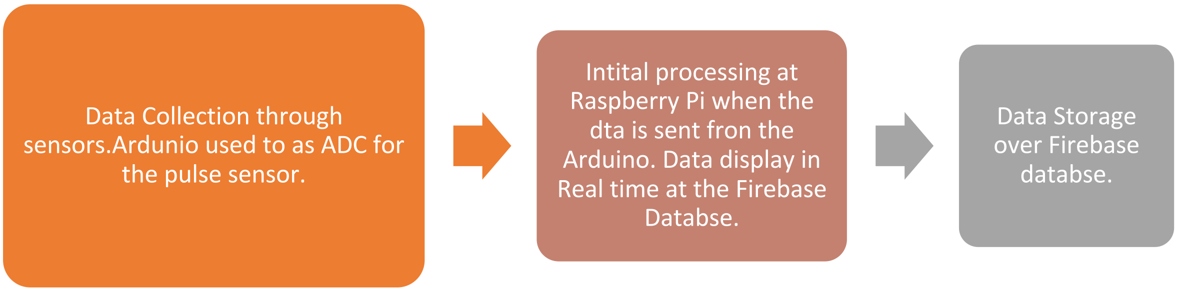

Figure2: The basic outline of the proposed system

The proposed system is made up of three major blocks, as shown in Figure 2. Patient data is collected using instruments such as a temperature meter and a pulse rate detector. Wireless connectivity is built-in to one block of the Raspberry Pi model; thus, no separate Wi-Fi module is necessary. Data is collected and sent to the Firebase Realtime database by this side. The Firebase database stores all of the system's data. Doctors have access to this information, which they can utilize to collect patient information and write suitable medications.

The design process for this system has been depicted in Figure 2. This indicates that if the value of the sensor is higher than the threshold value, a notification will be sent via the user application to the physician and the caregiver and the physician and caregiver is even sending an alert message. The value is entered in the database using the Raspberry pi if the sensor result is below the predefined threshold; the data can be retrieved through the user application by logging in with the user authentication by the doctor or the patient. The doctor will be able to write the prescription by going through the data and make changes if necessary.

The software tools that will be used are RASPBIAN OS, Python language, Arduino IDE and C language. This will help us in interacting with the Raspberry Pi, getting the data stored in the database and sending alert messages to the doctor as well as the care taker if needed.

Figure 3 shows the flow chart of the smart health care device using Raspberry Pi.Firstly ,all the sensors are intialialized. The read sensors are then transferred to by the help of an ADC which is the arduino Uno and then to the Raspberry Pi. Whereas if sensor value is higher, then perhaps the threshold value; patient and caregiver will receive an alert notification. All the associated sensor value is then stored in the firebase databse. An application which is the firebase Realtime databse can collect all the data from both the tempearture and pulse senosr.The doctor can write a prescription for the patient; if the sensor value is greater than the threshold vlaue and aret message is sent to the doctor .The patient can follow up with this information.

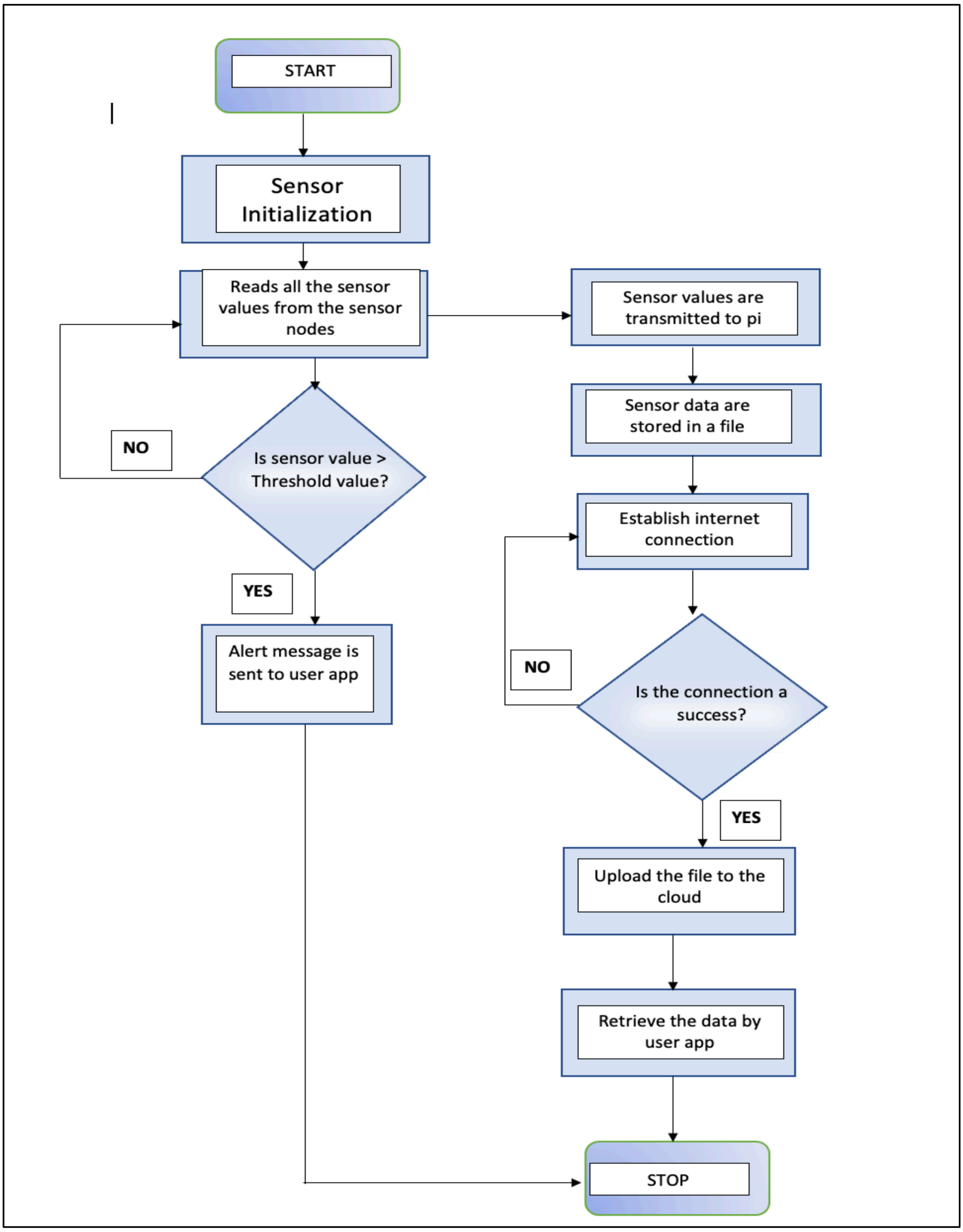

Figure 3: Flowchart of The Smart Healthcare Monitoring Device

Figure 4 depicts the system's overall layout, including the temperature as well as heartbeat sensors' connections. The Arduino Uno is then connected to the sensors. The Arduino serves as an ADC for the pulse sensor, as well as the signals are transferred to the Raspberry Pi. Because the temperature sensor has an ADC built in, the digital signal is sent to the Raspberry Pi. The Raspberry Pi is connected to the Arduino through serial COM. The data is sent in the form of a JSON object. Because the Raspberry Pi 3B+'s WIFI has already been turned on, all sensor data is sent to the Firebase database. The data is stored in the Firebase Real-time database using Python programming language and the Pyrebase module, just like on the Raspberry Pi.

a. Block Diagram

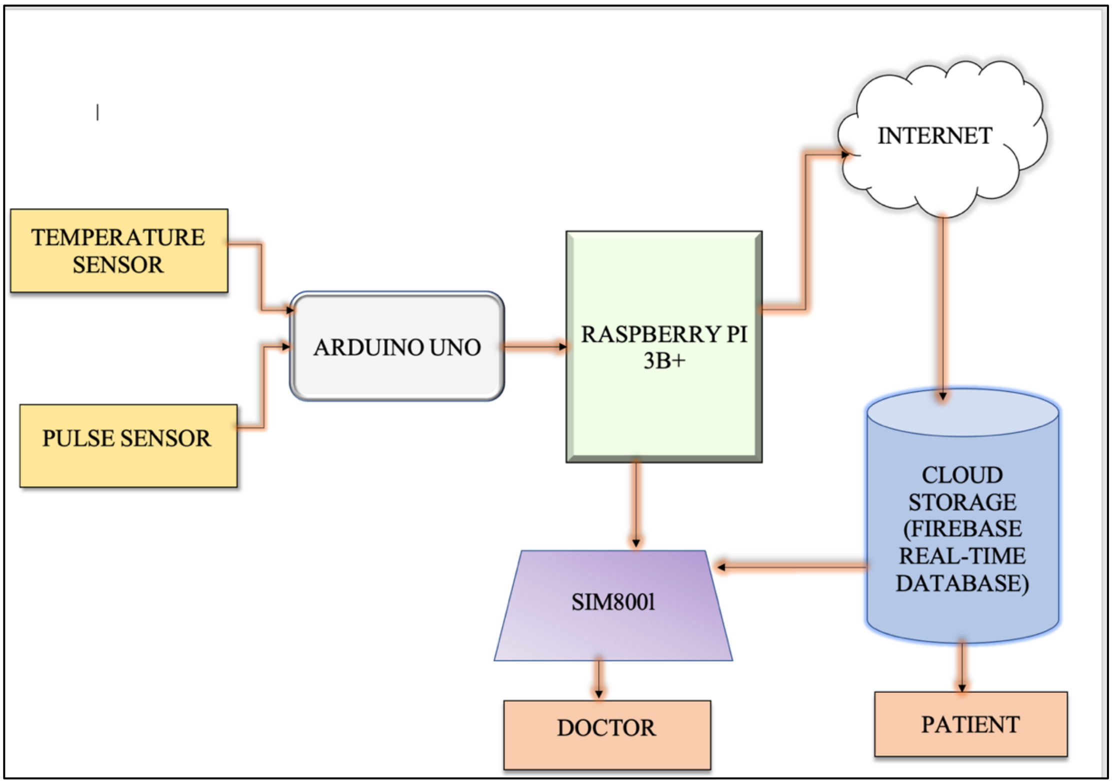

Figure 4: Smart Healthcare Device Block Diagram

The circuit schematic for the Raspberry Pi-based Smart Healthcare Device is shown in Figure 5. The Arduino Uno is used to power the breadboard. All of the sensors are then linked to the Arduino Uno, with the pulse sensor attached to the analogue A0 port and the temperature sensor to the D2 port. The Raspberry P is then linked to the Arduino. The SIM 800l is powered by the Arduino at 3.3V and is linked to the RX and TX GPIO pins. Similarly, the circuit is finished.

b. Circuit diagram

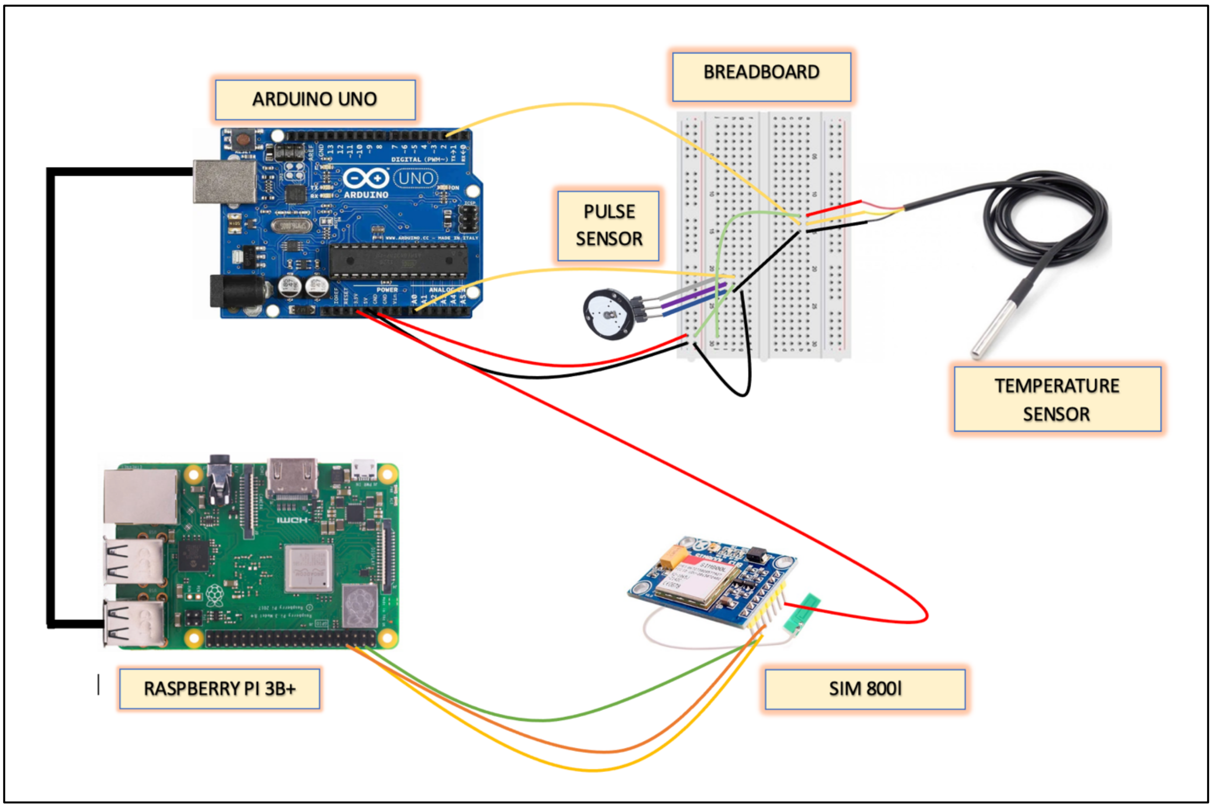

Figure 5 : Smart Healthcare Device Circuit Diagram

Figure 6 shows how th cloud storage works with microcontroller.Firstly all the data is collected from the sensor values then proccesed in to the rapberry pi microcontroller.Then ,tis microcontroller sends signal via wriresless transcievrs into the computer.The raspberry pi can be connected to the internet directly.All the data is stored in the computer.The patient and the doctor can access in information through the GSM (Global System for Mobile Communications) signal. The GSM MODEM(Modulator-Demodulator) sends the GSM signal to the doctor and the patient.

c. Cloud Computing

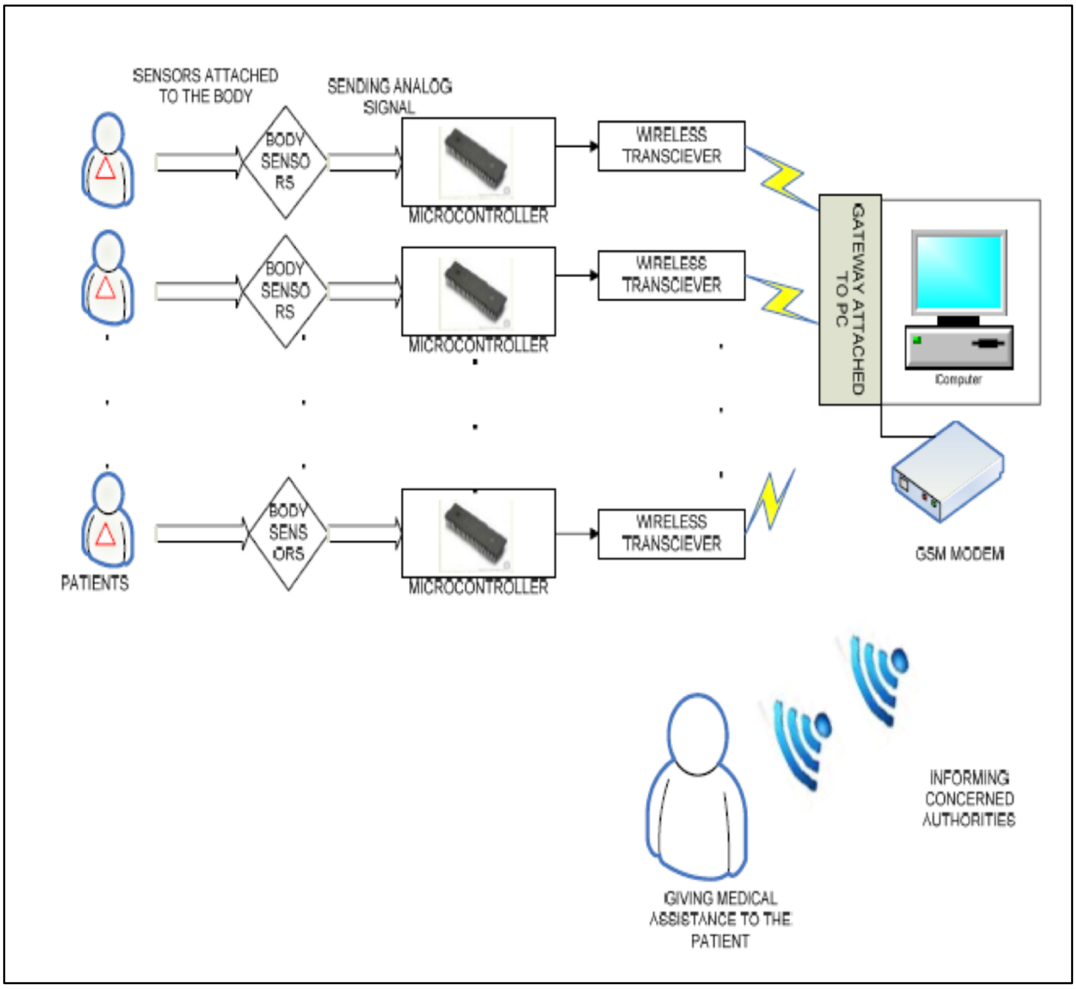

Figure 6: Cloud Computing in Smart Health care Device

3.2. Flowchart of Research Activities

Figure 7 shows the descriptive analysis on how the research activity was conducted with the research, desgining and modeling.This project was briefly evaluated with any errors.It was checked if it fulfills the objective.If it doesn't again the research was re-evaluated.Likewise,the final report was obtained.

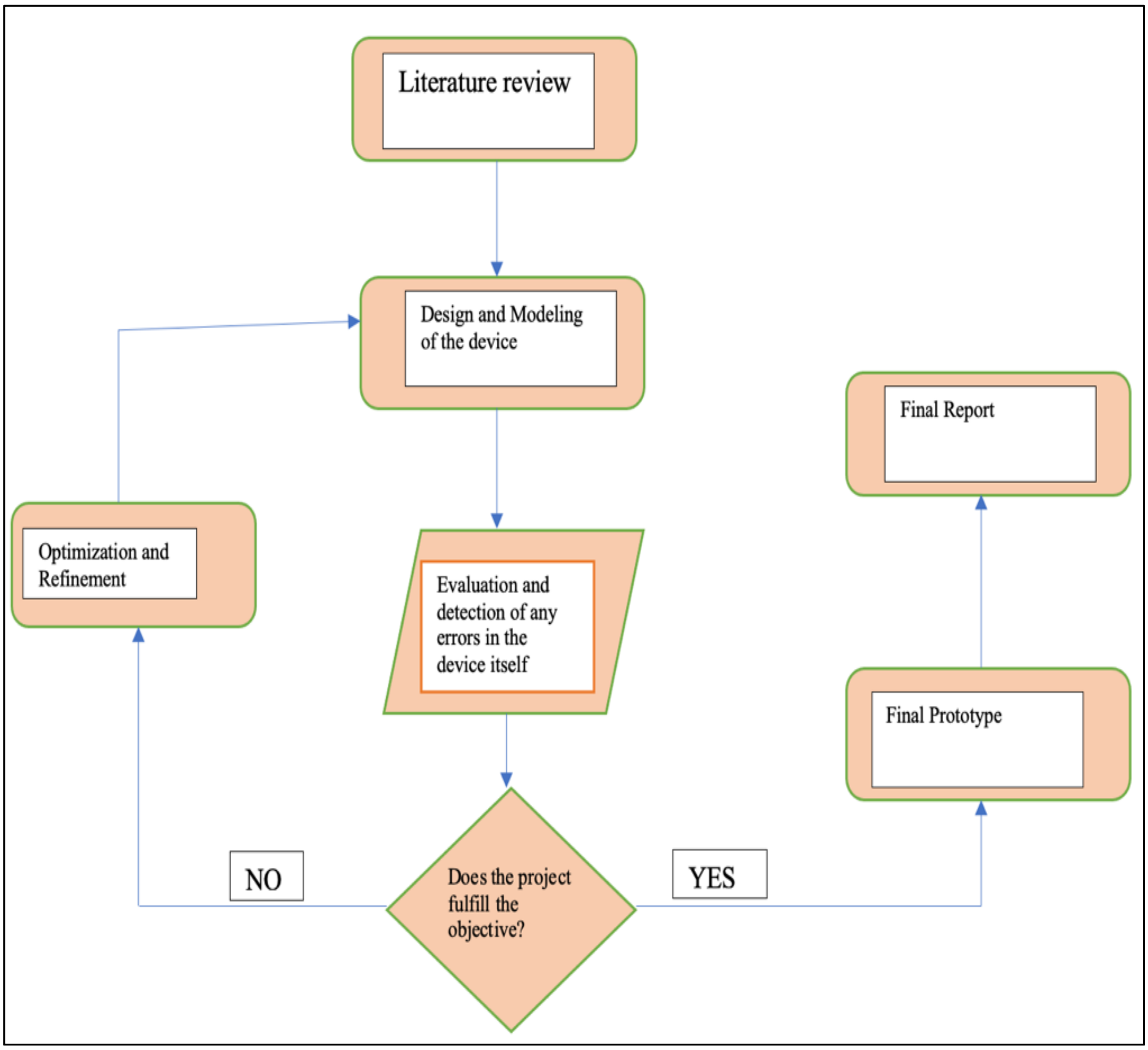

Figure 7: Smart Healthcare Device Research Activity Flowchart

3.2. Suitability of Hardware

The suitability of hardware components that will be employed in the Iot-based patient monitoring system has been thoroughly researched. Several techniques are examined since they employ classification methods that necessitate the utilization of various hardware components. To establish which strategy will work best for this application, these hardware devices are analyzed in terms of price and performance. This research is also being performed to satisfy all of the project's objectives, which also include examining and ensuring that the hardware components of the system. This corresponds to the required objectives, such as reading the patient's temperature and pulse, storing the sensors' data in the cloud server, and sending an SMS alert if the sensor values exceed the threshold value.

3.3. Connectivity of Hardware Components

In table 2, this shows the connections between the Raspberry Pi and Arduino, SIM800l and Arduino, and Raspberry Pi 3B+ and SIM800l, Raspberry Pi and Arduino.

Raspberry Pi 3B+	SIM800l	Arduino Uno
RX	RX	-
TX	TX	-
-	Vcc	3.3V
GND	GND	GND

Table 2: Connections of the Raspberry Pi, SIM800l and Arduino Uno

In table 3, this shows the connections between the Pulse sensor and Temperature sensor, Pulse sensor and Arduino, and Temperature sensor and Arduino Uno.

Pulse Sensor	Temperature Sensor	Arduino Uno
Vcc	Vcc	Vcc
GND	GND	GND
Signal	-	A0
-	Signal	D2

Table 3: Connections of the Pulse sensor, Temperature sensor and Arduino Uno.

3.4. Contribution of the Main Hardware components

1) ***Raspberry Pi Model B+***

Raspberry Pi is a single microcontroller board with a CPU, a GPU and a storage device. The gadget was established in the UK approximately the dimensions of a bank card. For all those who did not even want computers or any other system software to use this program, this board was developed. Excluding the microcontroller board that needs a supply of power, the system design can be run with just this board. The Raspberry Pi features a BCM2837 chip-based Broadband systems, including an ARM Core A53 1.2 GHz processors, Video Core IV GPU, and 1GB RAM. There is no integrated hard drive, rather it launches and retains information through an SD card. The OS, applications and data needed to run the Raspberry Pi all have to be saved mostly on SD card. The microcontroller works with the operating system. The operating system is responsible for systems integration and input management.

This model offers characteristics including streaming video in HD. It also gives the possibility of playing 3D games with high-definition audio. This gadget uses the ARM CPU. In the healthcare sector, the type Raspberry Pi B+, that may be connected to a variety of metered equipment. The microcontroller is a digital information system.

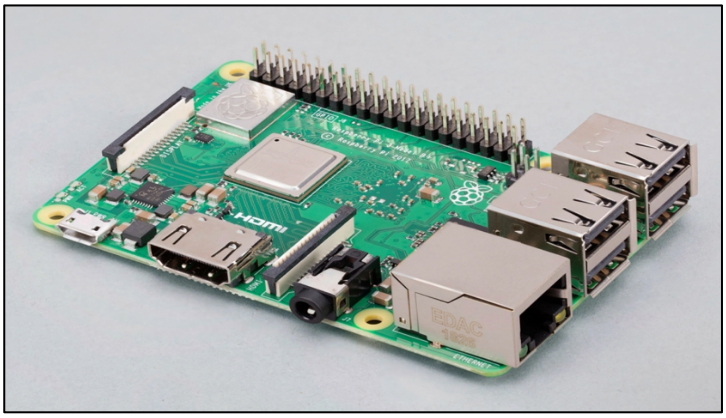

Figure 8: Raspberry Pi

2) XD58C- Heart beat Sensor

If users create or analyze a fitness regime, the pulse rate information is somewhat valuable. The problem is that it could be hard to physically measure the pulse rate. Fortunately, this can be helped also with pulse detector. The software may be used by sportsmen, fitness professionals and cellphone designers that require a real-time cardiac beater sensor. The optical sensor with amplifier is easy to filter out street sounds, which allows precise pulse recordings to be produced quickly and easily. The energy consumption can be reduced, consuming just 4 mA at 3.3V, which makes it even better overall portable use.

Just click on the pulse rate sensor and attach the Heart beat sensor to the ears or the fingertips and the heart rate may be measured around 3 or 5.5

volts. For most individuals, the cardiac cycle is between 60 and 80 beats per minute. A normal heart rate for well-trained athletes is 40 to 60 beats per minute.

Age	Pulse rate
< 1 month	120 to 160
1to 12 months	80 to 140
12 months to 2 years	80 to 130
2to 6 years	75 to 120
6 to 12 years	75 to 110
> 12 years	60 to 100

Table 4: Normal Pulse Rate of Different Ages

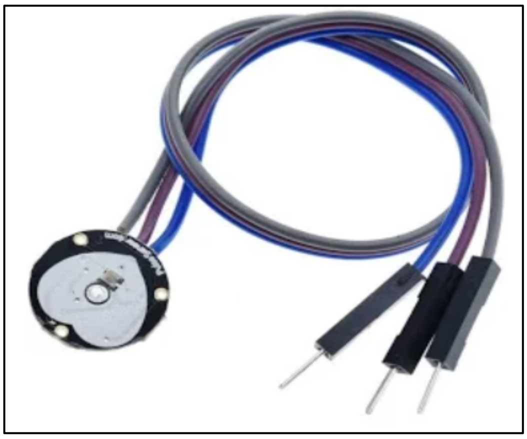

Figure 9: Heart beat Sensor

3) *Temperature Sensor DS18B20*

The DS18B20 sensor from Maxim integrated is a 1-wire programmed sensor. The sensor is built to last, plus it can even be obtained inside a waterproof version, allowing a easy setup. It has a 5°C precision and therefore can measure a wide temperature range between -55°C and

+125°C. Because each sensor has its own position, data is transported using only one MCU pin, making it an ideal choice when monitoring heat over numerous places while compromising several of one’s microcontroller's input pin.

Normal	37°C = normal temperature.
Abnormal	Fever: 38°C - 39.9°C High fever: 40°C >
	Armpit(auxiliary) temperature Fever: 37.4°C - 39.4°C High fever:39.5°C >
	<36.1°C means a low body

Table 5: Body Temperature

No:	**Pin Name**	**Description**
1	Ground	Connect to the ground of the circuit
2	Vcc	Powers the Sensor, can be 3.3V or 5V
3	Data	This pin gives output the temperature value which can be read using 1-wire method

Table 6: Pin Configuration

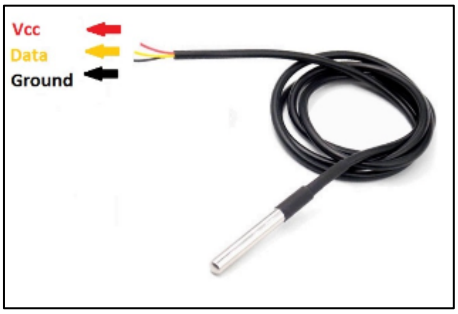

Figure 10: Temperature Sensor

4) SIM 800l GPRS GSM module

The SIM800L is a tiny device for sending or receiving GPRS, transmitting but also receiving text messages, and making as well as receiving voice calls. The reduced price, small size and four spectrum bandwidth characteristics of this Module makes it perfect for any long-range communication projects.

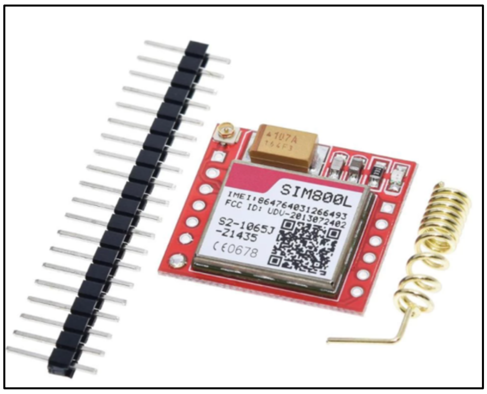

Figure 11: SIM800L

5) Arduino Uno R3

The Uno board has 14 digital input and output pins, with six of them dedicated to PWM generation. As illustrated in Figure 13, it is founded on the Atmega 328P and is an open-source design with a 16Mhz clock frequency. It could be connected to a USB flash memory with such a 32kb capacity for data storage. ICSP header is supplied to directly interface it as a serial device, and saved data could be readily erased by simply pressing the reset button. With specially assigned pins (A0 to A5) capable of reading analogue data, the ADC (analogue to digital converter) becomes even more impressive. Arduino UNO can differentiate 1024 analogue signals thanks to the 10-bit integrated ADC. This can convert analogue voltages between 0 and 5 volts into different 1024 integer values (khan et al., 2019).

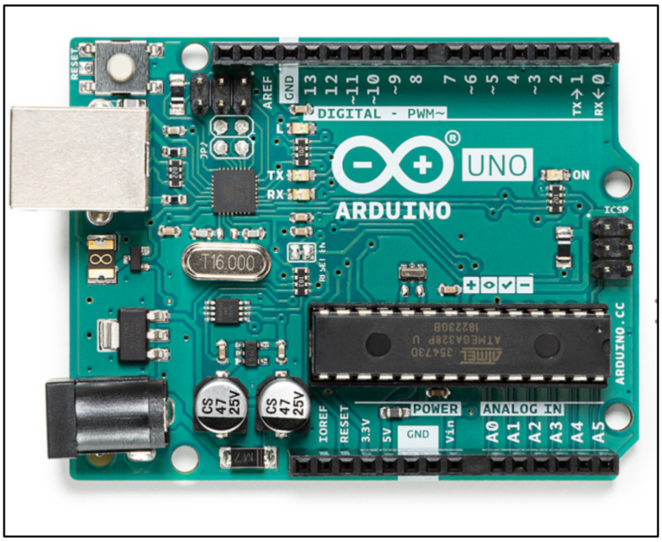

Figure 12: Arduino Uno R3

3.4. Arduino IDE Installation

The Arduino IDE (Integrated Development Environment) is a software development environment for the Arduino microcontroller. To send and communicate with programmes, it communicates to Arduino plus hardware. The codes are stored using the Arduino IDE editor. It's an Arduino programming environment that's open - source software as well as simple to use (Ramesh Saha et al., 2021).

3.5. Raspbian OS Installation

Raspbian is a simple operating system to set up on the Raspberry Pi. First step would be to install Raspbian then copy the disc image to just a microSD card, then reboot the Raspberry Pi from the microSD card. The above method needs a microSD card (at least 8 GB), a desktop with such a slot for it, as well as, of class, a Raspberry Pi and basic items (a mouse, keyboard, screen, and power source). This isn't the only way to install Raspbian (more on that later), but it's a useful starting point since it can be used to install a range of other operating systems on the Raspberry Pi.

Preliminary Data

Budget of Proposed System

HARDWARE			
NUM.	**ITEM**	**QUANTITY**	**PRICE (RM)**
1.	Raspberry Pi 3 Model B+	1	220.00
2.	Jumper Cables	1	2.00
3.	XD58C- Heart beat Sensor	1	12.90
4.	DS18B20- Temperature Sensor	1	5.50
5.	Breadboard	1	4.80
6.	SIM 800l GPRS GSM module	1	24.50
7.	Arduino Uno R3	1	36.19
8.	PCB Board	3	6.00
TOTAL			79.30
SOFTWARE			
1.	Python language		Free
2.	RASPBIAN OS		Free
3.	Arduino IDE		Free
4.	Firebase		Free

Table 7: Preliminary data

CHAPTER 4

RESULT AND DISCUSSION

4.1 The Result of the IoT Based Patient Monitoring System

The IoT based patient monitoring system's Arduino code was applies using three predefines libraries. The libraries are PulseSensorPlayground.h, OneWire.h and DallasTemperature.h. PulseSensorPlayground.h includes the PulseSensorPlayground Library. DallasTemperature.h which Supports Arduino Library for Dallas Temperature ICs. OneWire allows to connect to Maxim/Dallas 1-wire devices, like temperature sensors. The code below in Table is for the Arduino Uno which acts as an analogue to digital signal converter (ADC) in-between the sensors and the Raspberry Pi. The Arduino takes the analogue signals from the pulse sensors which is pulse and converters it to the digital signal. For the temperature sensor the signal is already in the digital port D2 where as for the pulse sensor it is in the port A0 in the Arduino Uno.All the codes can be found in Appendix 1A.

This is the code for the Raspberry Pi using the python programming language, for the Raspberry Pi to store all the data in the firebase database. The response from the urllib is an internal library for JSON data type. Raspberry Pi sends all the data to the firebase database using the library for Firebase Realtime database which is pyrebase. For the serial communication of the Raspberry Pi to the Arduino using the library which is serial. Time function is also used. SIM800l library function is used to send all the alert messages. All the codes can be found in Appendix 1B.

All the codes for the dashboard are written in JavaScript, HTML and CSS. This is to display the data of pulse sensor in a graphical format where as for the temperature sensor it shows as it increases. All the codes can be found in Appendix 1C.

4.1.1 Testing the IoT Based Patient Monitoring System

The best results and performs as expected. The three-unit testing that was conducted was the Raspberry Pi and Arduino Uno was powered up, the reading from the temperature and pulse sensor was stored in the firebase database and if the value of the sensor is greater than the threshold value; and alert email was sent to the doctor and the care giver.

4.1.1.1 Unit Testing: Powering the Raspberry Pi and Arduino Uno

As the SSH was enabled in the Raspberry Pi, the raspberry pi was powered with the help of an 5V adapter with 3A current. as shown in the Figure 14. As soon as the Raspberry Pi is connected to power, a red LED light on the board illuminates. The green LED light also glows in the Raspberry Pi.

Figure 13: The Raspberry Pi 3B+ was powered

The Arduino is also connected to the Raspberry pi serial COM. In the Arduino, the board's green LED power indication glows. The power supply in the Raspberry pi was a 5.1V adapter with 3A current.

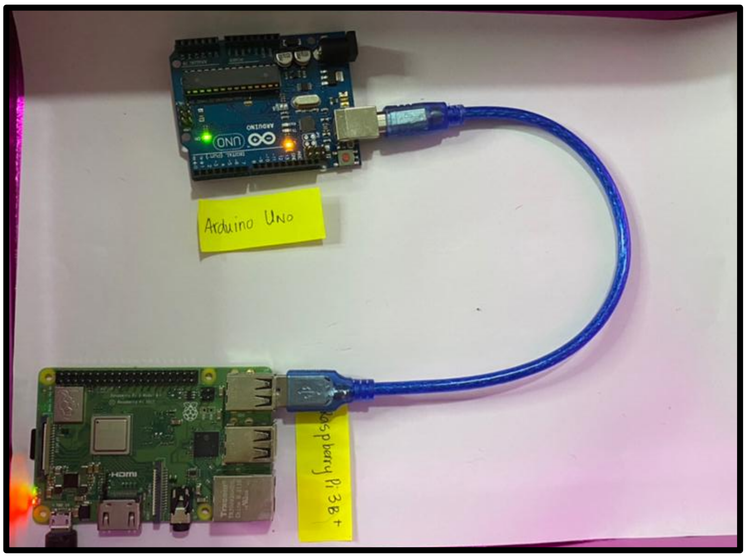

Figure 14: The Arduino Uno was powered

4.1.1.2 Unit Testing: Getting the Reading from the Pulse and Temperature Sensors

In this second unit testing, first the patient gets oneself checked with the pulse and temperature sensor. Then the Arduino which is the ADC sends the digital signal to the Raspberry Pi. The data from the two sensors were obtained through the firebase database. Raspberry Pi was coded into sending the data into the firebase database. Figure 13 below depicts the data of both the patients “0001” and “0002”. All of the readings have been processed into the Raspberry Pi and displayed via the IoT server firebase database. Figure 14 shows the dashboard of the dataset of the reading of the pulse sensor of the patient.

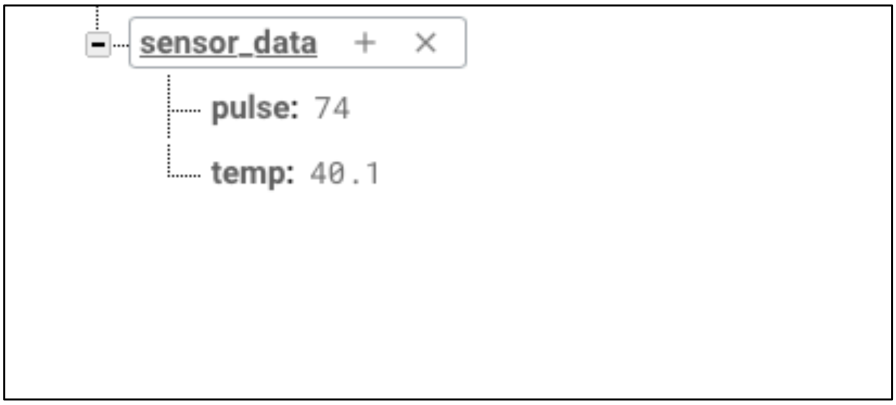

Figure 15: Reading of the Temperature and Pulse sensor in the firebase Realtime Database

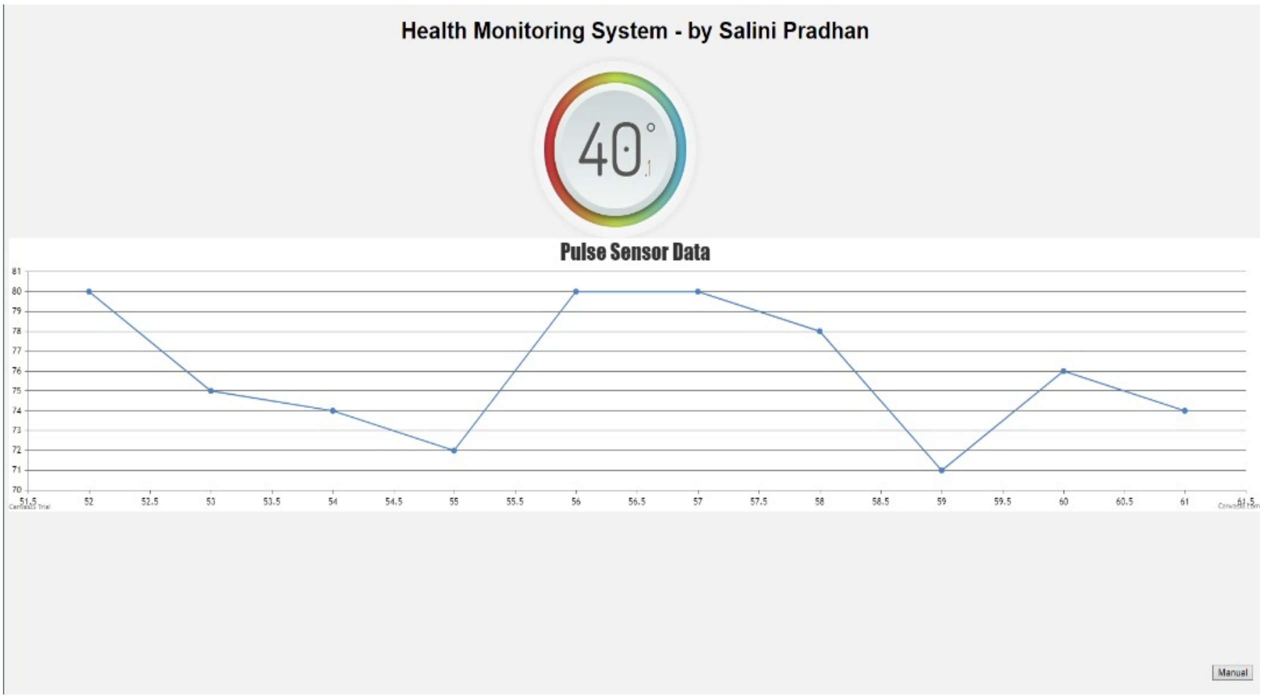

Figure 16: Dashboard of the pulse and temperature sensor data

4.1.1.3 System Testing

The third stage of testing was system testing, which involved testing the complete Iot based patient monitoring system using Raspberry Pi. This system testing included the testing of both the sensors which is temperature sensor and pulse read different kinds of data. Then, Arduino was used as an ADC and then the data was transferred from the raspberry pi to the firebase database. All the reading were displayed in the dashboard of both the temperature and pulse sensor.

If the reading exceeds the threshold, the message is delivered to the doctor via the GSM module. For the temperature sensor, if the temperature is 37°C then message that is sent to the doctor “Patient is having a fever of:” the value that is recorded by the patient at that time i.e. In figure 16 the alert message is sent to the doctor “Patient is having fever of: 37.6”. In the figure 16 below for the pulse sensor, if the beat is greater than 100 bpm in the average reading then an alert message was sent to the doctor “Patient has high pulse rate:102” and if the beat is lower that 60 bpm then an alert message was sent to the doctor “Patient has low pulse rate:58”.

After the doctor receives the notification, he can immediately prescribe medicine through the message to the patient.

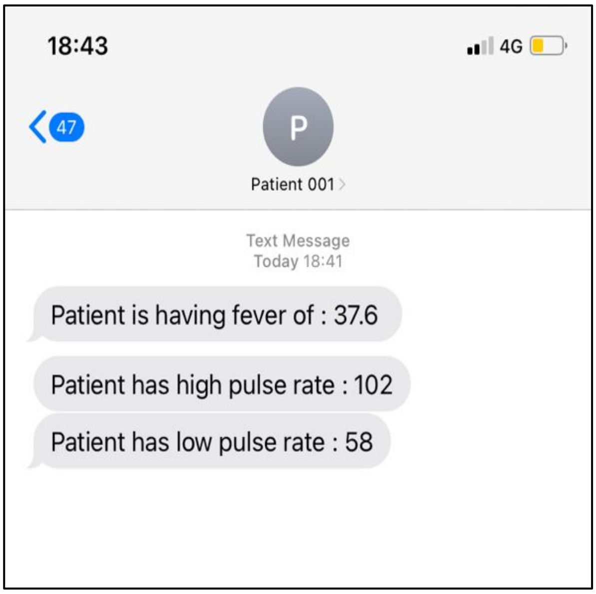

Figure 17: Alert message sent to the doctor as the temperature and pulse was greater than the threshold and pulse was lower than the threshold.

4.1.2 The Final Result of the IoT Based Patient Monitoring System

The final results are shown in the figures below.

In Figure 20 all the wires that are connected to the GND and Vcc is soldered on the PCB board by soldering it. This is the initial process when both the sensors were connected to the Arduino Uno.

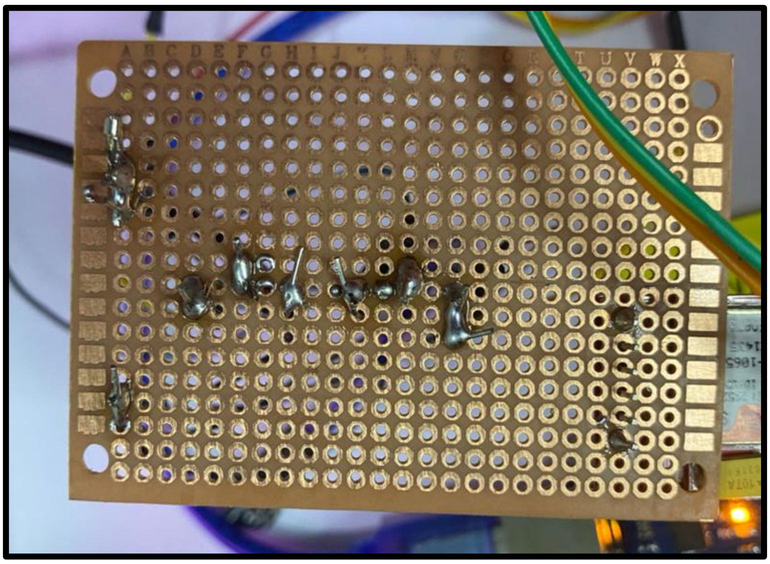

Figure 18: All the wire connected from the sensors to the Arduino Uno was soldered to the PCB board

In Figure 21 all connected wire to the PCB board which has been soldered and will be connected to the Arduino Uno.

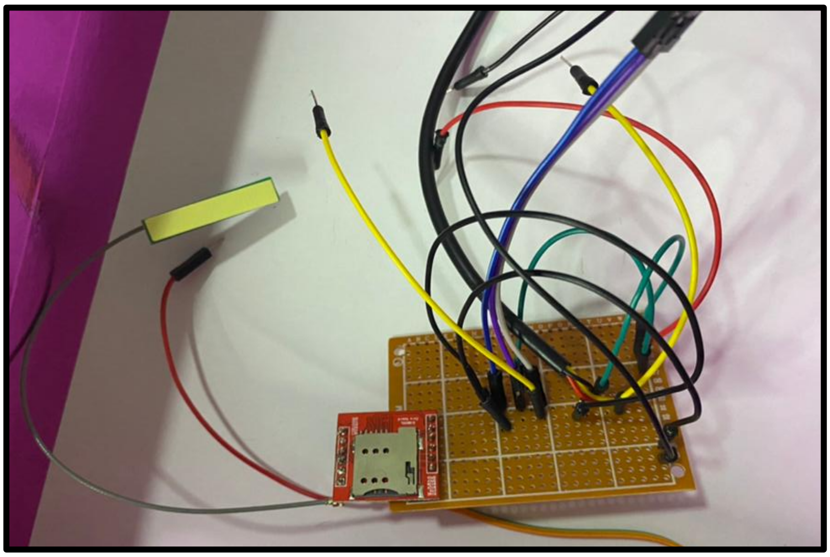

Figure 19: All the wires soldered to the PCB board. Connections of the wire.

The circuit has been completed in Figure 22 with the Raspberry Pi supplying electricity to the Arduino Uno. Temperature sensor, pulse sensor, and SIM800l are all powered by the Arduino Uno. The pulse sensor was functional if the green LED light on the pulse sensor is glowing. The circuit has become complete, with the analogue signal from the pulse sensor being delivered to the Arduino Uno. On the PCB board, the pulse sensor is connected to GND and VCC. The Arduino has been the source of electricity for the board. Whenever the patient obtains a pulse measurement, the L LED on the Arduino blinks every time. The temperature sensor is attached to D2 upon that Arduino's digital pin. The information from the temperature is transferred to the Arduino in this manner. The temperature sensor is connected to the Arduino because both sensors output signals in the very same direction. As a result, while this initiative is for ICU patients with pre - existing health issues, it cannot be proved in the real world with patients with normal health circumstances.

As soon as the Arduino receives all of the signals, it sends them towards the Raspberry Pi, in which all of the temperature and pulse sensor data is sent through serial COM. The Raspberry Pi 3b+ that is being used is WIFI compatible. So, when developing on the Raspberry Pi, all of the data from the Raspberry Pi is sent to the Firebase real-time database that use the Pyrebase library.

Figure 20: All the wires soldered to the PCB board. Connections of the wire.

Figure 23 shows how many of the data was saved in the Firebase Real - time manner, including the real-time readings of the patient's temperature as well as pulse rate, which were "pulse:72" and "temp: 30 ." An alert message is provided to the doctor whether any of the sensor's readings exceed the threshold. Every one of the alarm messages were saved in the firebase database, as seen below: "Alerts: Patient has a fever of 40" and "Alert: Patient has a pulse rate of 110".

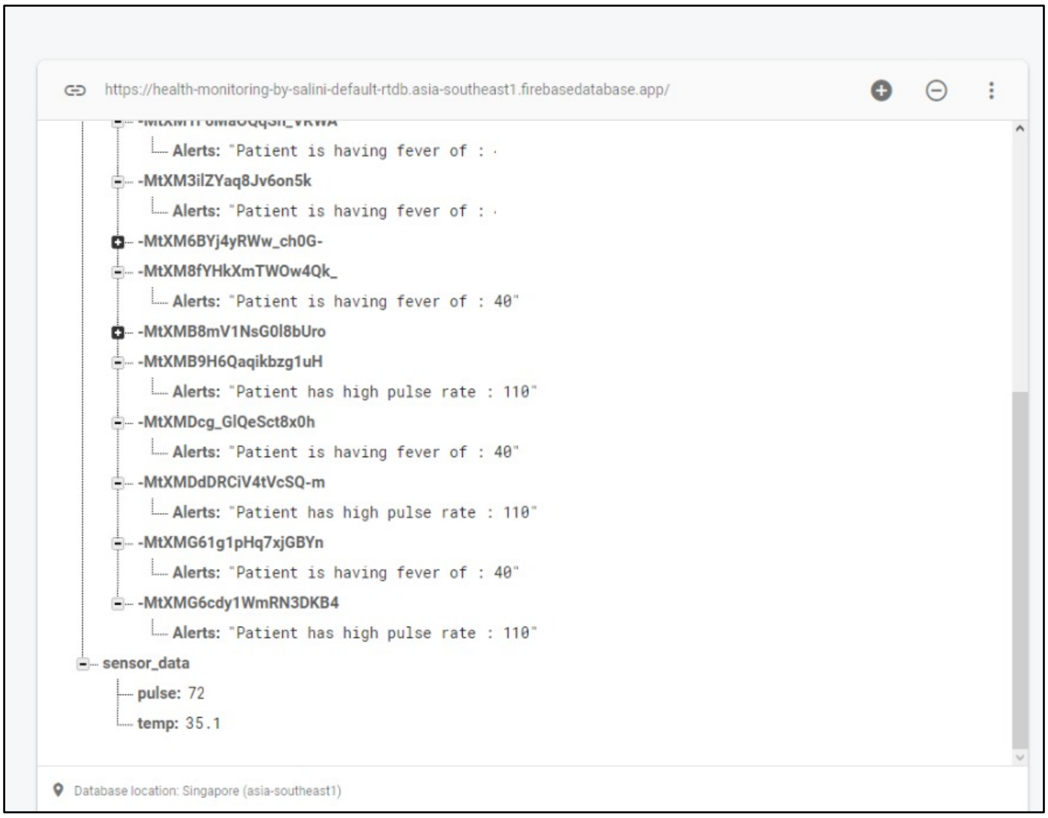

Figure 21: All the wires soldered to the PCB board. Connections of the wire.

In figure 24, all the dashboard been shown in the form of a dashboard as shown below the pulse rate of the patient that varies as the time changes and the temperature of the patient.

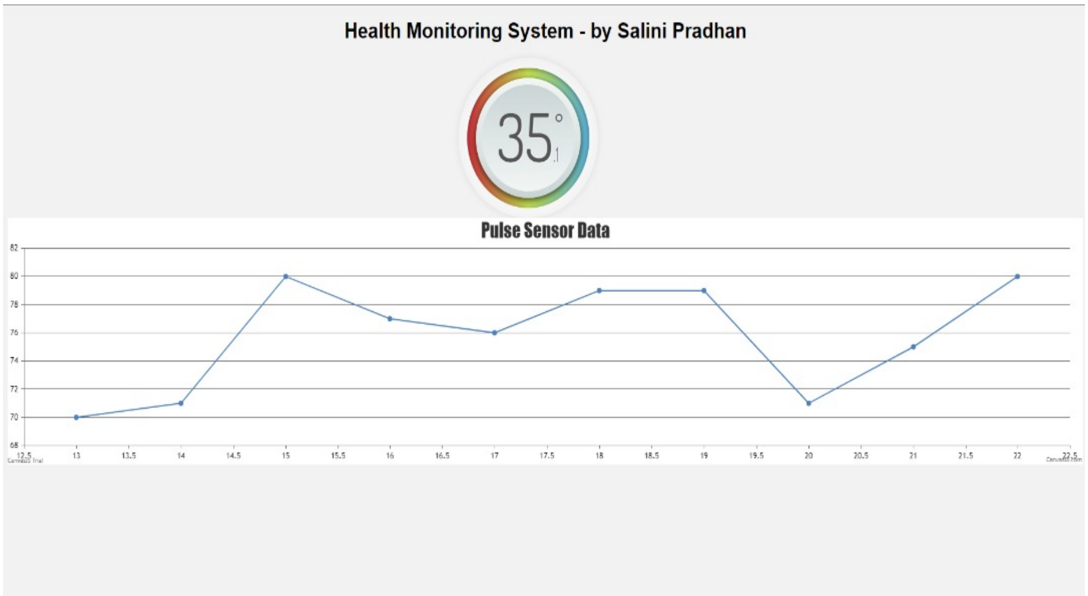

Figure 22: Dashboard of the pulse and temperature sensor data

In Figure 25 and Figure26 shows the alert messages that was sent to the doctor from the SIM 800l.If the sensor values is greater than the threshold.

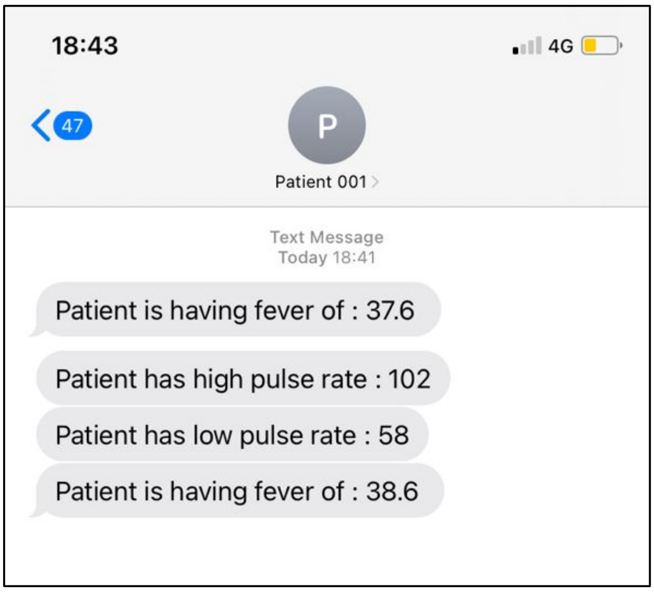

Figure 23: All the wires soldered to the PCB board. Connections of the wire.

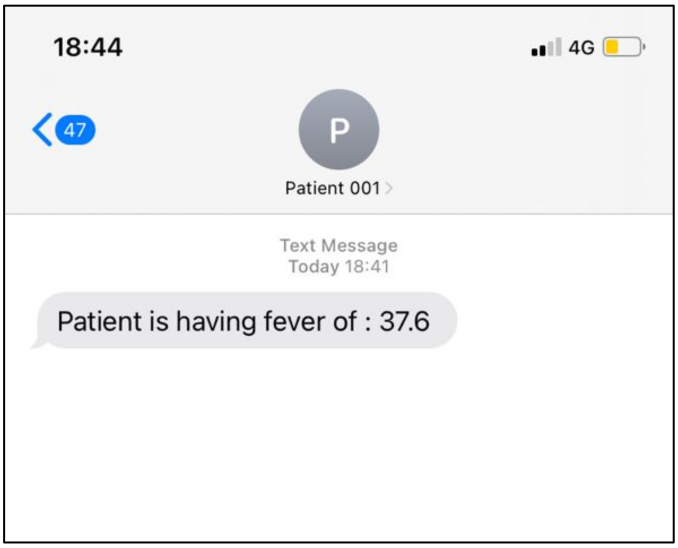

Figure 24: All the wires soldered to the PCB board. Connections of the wire.

4.2 Meeting Research Objectives

As a result, the study objectives are assessed and measured in order to assure the project's success. In order to accomplish the project's objectives, extensive study was conducted in terms of literature review, research methodology, and other studies related to the project. The following is an explanation of how to accomplish each of the project's main objectives. It has been sectioned into objective 1, objective 2 and objective 3. All these objectives have been briefly described of how it was accomplished.

4.2.1 Meeting Research Objective 1

The first objective of this project is *to measure the patient's pulse rate, and body temperature using Raspberry Pi*. This first objective is the beginning on how the project starts by recording the temperature and pulse sensor of the patients. The explanation of this objective can be found on pages 18-24 of Chapter 2's Literature

Review section. In a similar project, the same components were utilized to use a Raspberry Pi to measure the patient's reading. It has been thoroughly investigated, with descriptive ways for obtaining the results from various types of sensors using the Raspberry Pi. In Chapter 3 page 22-25, this objective is examined in depth and described in terms of the methodology. In this segment, it gives a brief description on how the both temperature and pulse sensor works in a flowchart. It also summarizes on what software's were used for the Raspberry pi to take the reading of the two sensors, where Arduino is use as an analogue to digital signal converter. In this section, it explains what programming languages were in used in both the Arduino and Raspberry Pi and the software's that were installed for their operation. In the Chapter 3 there is a section called Contribution of the Main Hardware Components, it describes what sensors were used and description of the functioning of the sensors. The workings of both the temperature and pulse sensors can be viewed in the result section. This means that the Arduino was configured as an analogue to digital converter and connected to the Raspberry Pi, which was programmed to read and save the data sent by the Arduino. The temperature of the patient was 74 degree celsius and Pulse rate was 40.1bpmHere, both the temperature and pulse sensors read the patient's temperature and pulse. As a result, the first objective was accomplished.

4.2.2 Meeting Research Objective 2

The second objective of this project is *to store patients reading of the sensors in the cloud database using Raspberry P*i. This objective is clearly described on pages 22-

27 of Chapter 3, Research Methodology. In this section, Figure 6 depicts how cloud computing works in this section, whereas Figure 4 depicts a block diagram of how the sensors are connected to the Raspberry Pi and stored in the cloud storage, which in this case is Firebase. The block diagram is explained in page 24, states that the data can be retrieved by the patient and the doctor. The part, Result and Discussion, on pages 37-38, contains all of the codes needed to programme the Raspberry Pi to store the reading of both the temperature and pulse sensor in the firebase database. The data from both the sensors can be seen in Realtime in the database. In the result section, it can be seen in the second unit testing that both the temperature and pulse senor reading has been configure in the Realtime database. Figure 13 shows the readings of both the temperature and pulse sensor which has been stored in the real-time firebase database. Likewise, the connection between the sensors and raspberry pi was built where the data was store successfully in the cloud storage. As a result, the objective was fulfilled.

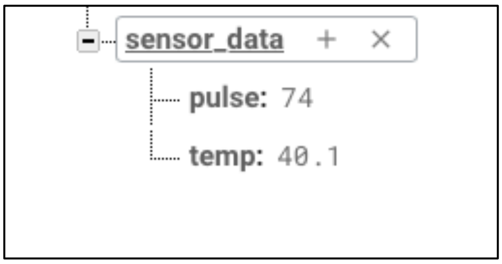

Figure 25: Reading of the Temperature and Pulse sensor in the firebase Realtime Database

4.2.3 Meeting Research Objective 3

The third objective of this project is *to implement an IoT based patient monitoring device that monitors and sends SMS alert to the doctor if the sensor value is higher than the threshold value*. This objective is explained on pages 18-24 Literature Review section of Chapter 2. The sensors have indeed been attached to an ADC or directly to the Raspberry Pi, and when the sensor value exceeds a certain threshold, the Raspberry's GSM network notifies to the doctor. Where the doctor can obtain information and recommend various medications. The purpose is outlined descriptively on pages 22-27 of Chapter 3, Research Methodology. The flowchart in Figure 4 shows how such an alarm message is issued to the doctor whenever the pulse and temperature sensors exceed the threshold level as well as the procedure comes to a halt. Similarly, in figure 6 in this section when the sensor value is higher than the predefined threshold, it shows how the signal gets sent through the GSM modem the doctor. Likewise, in the same chapter there is a segment called Contribution of Main Hardware Components where Sim 800l has been described briefly of the work that it does to send the signal via alert message to the doctor. In the result section, Page 42-49, the alert message was sent to the doctor if the value of both the temperature and pulse sensor was greater than the threshold. As shown in figure 28, the objective was obtained.

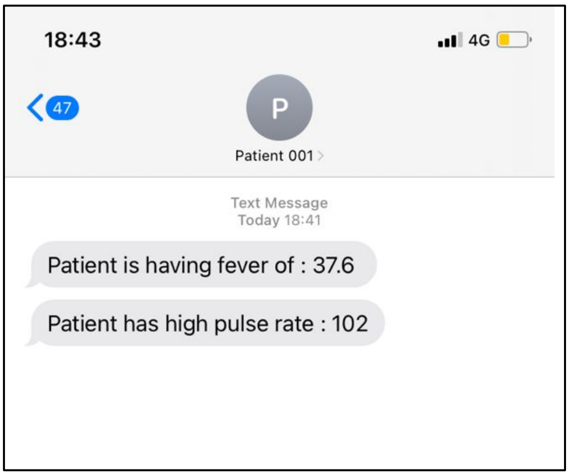

Figure 26: All the wires soldered to the PCB board. Connections of the wire

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

All the research for the systems development has been thoroughly done. The IoT-based Patient Monitoring has been developed to address the concerns identified in the problem statement section, namely, the problem of scheduling a doctor's appointment and waiting to be checked for patients with underlying cardiac diseases. As a result, this project is being worked on in order to save the patient's time, monitor the patient without the doctor's physical presence, and save the patient's life.

Significant technical achievements were made in relation to the project statement and objectives. The projects' software and hardware technicalities were investigated, the suitability of the hardware was investigated, the Arduino IDE software was installed and Arduino codes were studied and written, the RASPBIAN OS software was installed and Python codes were studied and written, the integration of the hardware components was analyzed, and the hardware component was assembled. All of this achievement has aided to the project's final outcome by satisfying the project's objectives.

In future, the research will be improved further by expanding the research to the hardware aspects of the project. The addition of different types of sensors like oxygen sensor, fall detection sensor, air flow sensor, respiration sensor etc. This can be added to this IoT based health monitoring system. The algorithm will be improved and detailed written as codes or instructions for all the types of sensors in the Arduino Uno. This will aid in comprehending the complication of the IoT-based patient monitoring system's operation, which uses a microcontroller, microprocessor, and several types of sensors to send an alert message through SIM800l if the sensor values exceed the threshold values. It will also be achieved thorough study of the Arduino programming language and Python programming language for the Raspberry Pi 3B+ for the different types of sensors to monitor the patient. This helps in providing an accurate result for what is happening to the patient.

The Raspberry Pi-based IoT-based patient monitoring system will be upgraded in the future by the addition of an emergency button. In that moment, the emergency button can assist the patient in contacting the caregiver or nurse. For patients with underlying heart problems, action can be performed right away. This will assist the system in becoming more advanced. This will assist the patient in a variety of ways, including getting medicine as soon as possible, getting blood drawn as soon as possible, and keeping the patient in ventilation if necessary. This system will bring about a transformation in the healthcare field that will result in a next-level change, which will be even more significant in the case of a pandemic.

Another feature that could be introduced is the use of Python's programming language to detect data anomalies. With the help of several libraries in Python, 5000 data points of the patient's heart beat can be accumulated and studied over a longer period of time. One of the libraries is the scikit-learn Python library, which is a free machine learning library for Python. Python supports NumPy and SciPy Python numerical and scientific libraries, along with methods like as supported vector machines, random forests, and k-neighbors. Using the predict function, one may predict the labels of the data values based just on learned model. The choice to convert a projected probability or scoring into the class label is governed by a parameter known as the "decision threshold," "discrimination threshold," or simply the "threshold." The default value for the threshold is 0.5 for normalized prediction or scores within range of 0 to 1. Examine the cutoff graphs of the individual datasets in the same way.

REFERENCES

Ahmed, Z. U., Mortuza, M. G., Uddin, M. J., Kabir, Md. H., Mahiuddin, Md., & Hoque, MD. J. (2018). Internet of Things Based Patient Health Monitoring System Using Wearable Biomedical Device. *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, 1–5. https://doi.org/10.1109/CIET.2018.8660846

Amru, M., Mahesh, A. V. N., & Ramesh, P. (2020). IoT-based Health Monitoring System with Medicine Remainder using Raspberry Pi. *IOP Conference Series: Materials Science and Engineering*, *981*, 042081. https://doi.org/10.1088/1757-899X/981/4/042081

Baker, S. B., Xiang, W., & Atkinson, I. (2017). Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities. *IEEE Access*, *5*, 26521–26544. https://doi.org/10.1109/ACCESS.2017.2775180

Chakravorty, D., Islam, S., & Rana, T. K. (2018). IoT Based Patient Guidance System using Raspberrypi. *2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, 1–4. https://doi.org/10.1109/IEMENTECH.2018.8465190

Ganesh, E. N. (2019). Health Monitoring System using Raspberry Pi and IOT. *Oriental Journal of Computer Science and Technology*, *12*(1), 08–13. https://doi.org/10.13005/ojcst12.01.03

Gupta, M. S. D., Patchava, V., & Menezes, V. (2015). Healthcare based on IoT using Raspberry Pi. *2015 International Conference on Green Computing* *and Internet of Things (ICGCIoT)*, 796–799. https://doi.org/10.1109/ICGCIoT.2015.7380571

Gutte, A., & Vadali, R. (2018). IoT Based Health Monitoring System Using Raspberry Pi. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1–5. https://doi.org/10.1109/ICCUBEA.2018.8697681

Hamim, Mohd., Paul, S., Hoque, S. I., Rahman, Md. N., & Baqee, I.-A. (2019). IoT Based Remote Health Monitoring System for Patients and Elderly People. *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, 533–538. https://doi.org/10.1109/ICREST.2019.8644514

Islam, Md. M., Rahaman, A., & Islam, Md. R. (2020). Development of Smart Healthcare Monitoring System in IoT Environment. *SN Computer Science*, *1*(3), 185. https://doi.org/10.1007/s42979-020-00195-y

Jaiswal, K., Sobhanayak, S., Mohanta, B. K., & Jena, D. (2017). IoT-cloud based framework for patient's data collection in smart healthcare system using raspberry-pi. *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 1–4. https://doi.org/10.1109/ICECTA.2017.8251967

Kaur, A., & Jasuja, A. (2017). Health monitoring based on IoT using Raspberry PI. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 1335–1340. https://doi.org/10.1109/CCAA.2017.8230004 khan, I., Zeb, K., Mahmood, A., Uddin, W., Khan, M. A., Saif-ul-Islam, & Kim, H. J. (2019). Healthcare Monitoring System and transforming Monitored data into Real time Clinical Feedback based on IoT using Raspberry Pi. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (ICoMET)*, 1–6. https://doi.org/10.1109/ICOMET.2019.8673393

Kirankumar, C. K. R., & Prabhakaran, M. (2017). Design and implementation of low cost web based human health monitoring system using Raspberry Pi 2. *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, 1–5. https://doi.org/10.1109/ICEICE.2017.8191881

Kumar, R., & Rajasekaran, M. P. (2016). An IoT based patient monitoring system using raspberry Pi. *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, 1–4. https://doi.org/10.1109/ICCTIDE.2016.7725378

Mathew, N. A., & Abubeker, K. M. (2017). IoT based real time patient monitoring and analysis using Raspberry Pi 3. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2638–2640. https://doi.org/10.1109/ICECDS.2017.8389932

Mathivanan, M., Balamurugan, M., L., H., Nandini, & Reddy, M. (2018). *IoT based continuous monitoring of cardiac patients using Raspberry Pi*. 020025. https://doi.org/10.1063/1.5078984

Mehta, A., Mehta, B., Joshi, M., Gohil, S., & Mahadik, S. (2018). *IoT Based Patient Health Monitoring System*. *3*(10), 3.

Muqeet, M. A., & Quadri, M. U. (2019). *IoT based Patient Monitoring System Using Raspberry Pi*. *7*(2), 6.

Naik, K. S., & Sudarshan, E. (2019). *SMART HEALTHCARE MONITORING SYSTEM USING RASPBERRY Pi ON IoT PLATFORM*. *14*(4), 5.

Neyja, M., Mumtaz, S., Huq, K. M. S., Busari, S. A., Rodriguez, J., & Zhou, Z. (2017). An IoT-Based E-Health Monitoring System Using ECG Signal. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 1–6. https://doi.org/10.1109/GLOCOM.2017.8255023

Pandey, G. K., & Chinnamuthu, P. (2018). IOT Based Patient Monitoring System utilizing Raspberry Pi and Web-Page. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2140–2144. https://doi.org/10.1109/RTEICT42901.2018.9012153

Pardeshi, V., Sagar, S., Murmurwar, S., & Hage, P. (2017). Health monitoring systems using IoT and Raspberry Pi—A review. *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 134–137. https://doi.org/10.1109/ICIMIA.2017.7975587

Prabha, M. G. M. (n.d.). *AUTOMATIC HEALTH MONITORING SYSTEM USING RASPBERRY PI*. 8.

Rahman, A., Rahman, T., Ghani, N. H., Hossain, S., & Uddin, J. (2019). IoT Based Patient Monitoring System Using ECG Sensor. *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, 378–382. https://doi.org/10.1109/ICREST.2019.8644065

Ramesh Saha, Biswas, S., Sarmah, S., Karmakar, S., & Das, P. (2021). A Working Prototype Using DS18B20 Temperature Sensor and Arduino for Health Monitoring. *SN Computer Science*, *2*(1), 33. https://doi.org/10.1007/s42979-020-00434-2

Rohit, S. L., & Tank, B. V. (2018). Iot Based Health Monitoring System Using Raspberry PI - Review. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 997–1002. https://doi.org/10.1109/ICICCT.2018.8472957

Saip, M. A., & Mohamed, A. S. (2018). Smart Health Monitoring and Controlling using Raspberry Pi3. *International Innovative Research Journal of Engineering and Technology*, *4*(1), 24–28. https://doi.org/10.32595/iirjet.org/v4i1.2018.70

Sankaran, S., Murugan, P. R., Chandrasekaran, D., Murugan, V., Alaguramesh, K., Britto, P. I., & Govindaraj, V. (2020). Design of IoT based Health Care Monitoring Systems using Raspberry Pi: A Review of the Latest Technologies and Limitations. *2020 International Conference on Communication and Signal Processing (ICCSP)*, 0028–0032. https://doi.org/10.1109/ICCSP48568.2020.9182325

Yattinahalli, S., & Savithramma, R. M. (2018). A Personal Healthcare IoT System model using Raspberry Pi 3. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 569–573. https://doi.org/10.1109/ICICCT.2018.8473184

APPENDICES

APPENDIX IA -The written in the Arduino IDE using C++ programming language for the Pulse and Temperature Sensor to work.

```
#define USE_ARDUINO_INTERRUPTS true
#include <PulseSensorPlayground.h>
// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

DeviceAddress insideThermometer;

// Variables
const int PulseWire = 0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.
int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.


PulseSensorPlayground pulseSensor;


void setup()
{

   Serial.begin(9600);         // For Serial Monitor

   // Configure the PulseSensor object, by assigning our variables to it.
   pulseSensor.analogInput(PulseWire); //pointing the A0 variable from the arduino
    pulseSensor.blinkOnPulse(LED13); //auto-magically blink Arduino's LED with heartbeat.
    pulseSensor.setThreshold(Threshold); //assignening the 550 to thr threshold

   // Double-check the "pulseSensor" object was created and "began" seeing a signal.
   if (pulseSensor.begin())
         {
                   Serial.println("We created a pulseSensor Object !");  //This prints one time at
                 Arduino power-up,  or on Arduino reset.
          }

   sensors.begin(); //temperature sensor
   Serial.print(sensors.getDeviceCount(), DEC); //it checkshow many temperature sensor
connect
   if (sensors.isParasitePowerMode()) Serial.println("ON"); //to check if its working ,library
defined function if the temperature sensor is connected
   printAddress(insideThermometer); //function name
   sensors.setResolution(insideThermometer, 9); //internal function
}
```

```
void printTemperature(DeviceAddress deviceAddress)
{
            // method 2 - faster
            float tempC = sensors.getTempC(deviceAddress);
             if (tempC == DEVICE_DISCONNECTED_C)
            {
                        Serial.println("Error: Could not read temperature data");
                        return;
            }
   Serial.print("Temp C: ");
   Serial.print(tempC);
   Serial.print(" Temp F: "); // temperature in celcius
   Serial.println(DallasTemperature::toFahrenheit(tempC)); // Converts tempC to Fahrenheit
}


void loop()
{
             int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor
            object that returns BPM as an "int".
            // "myBPM" hold this BPM value now.

             if (pulseSensor.sawStartOfBeat())
            {           // Constantly test to see if "a beat happened".
                          Serial.println(" A HeartBeat Happened ! "); // If test is "true", print a message
                          "a heartbeat happened".
                        Serial.print("BPM: ");
                         Serial.println(myBPM);
             }

    delay(20);                  // Delaying time for20 ms
    sensors.requestTemperatures(); //to re quest next temperature value
    printTemperature(insideThermometer);
}

//identify the temperature sensor
void printAddress(DeviceAddress deviceAddress)
{
            for (uint8_t i = 0; i < 8; i++)
            {
                        if (deviceAddress[i] < 16) Serial.print("0");
                         Serial.print(deviceAddress[i], HEX);
            }
}
```

APPENDIX IB – The Code for the Raspberry Pi 3B+ to store all the sensor’s data in the Firebase Real-Time database.

```
from urllib import response //internal library for JSON data type
import pyrebase    //library for firebase
import serial         //library for serial communication between Arduino
import time    //time functions
from sim800l import SIM800L       //library function Sim800l
sim800l = SIM800L('/dev/serial0')  //serial port (GPIO pin) for simm800l

//firebase Realtime database configuration
config = {
   "apiKey": "AIzaSyAAawGHw2l1Q6krVGTlAcxUVBaszZyNRlc",
   "authDomain": "health-monitoring-by-salini.firebaseapp.com",
   "databaseURL":"https://health-monitoring-by-salini-default-rtdb.asia
southeast1.firebasedatabase.app",
   "projectId": "health-monitoring-by-salini",
   "storageBucket": "health-monitoring-by-salini.appspot.com",
   "messagingSenderId": "1097375343656",
   "appId": "1:1097375343656:web:8a23d9a4f639f4c65f811d"
}

firebase = pyrebase.initialize_app(config)     //creating firebase object using config file
db = firebase.database()     //creating database object using firebase.database function
sensorData = db.child("sensor_data").get()               //accessing sensorDatabse from the
                                                          firebase database
serialport = serial.Serial("/dev/ttyS0", 9600, timeout=0.5)  //serial port connection for Arduino

response = serialport.readlines().  //reading sensors data from Arduino through serial port

//calling the loop function
def loop():

   pulse = response[0]       //array value 0
   temp = response[1]            //array value 1

   //updating the values in the db
   db.child("sensor_data").update({"pulse": pulse})
   db.child("sensor_data").update({"temp": temp})

   pulse = db.child("sensor_data").child("pulse").get().val()
   temp = db.child("sensor_data").child("temp").get().val()

   //save the alert messages in the db
   def saveAlert(sms):
      data = {"Alerts": sms}
      db.child("Alerts").push(data)

   sms = ""   //sms variable
   if(temp > 37.8):
      sms = "Patient is having fever of : " + str(temp)
      sim800l.send_sms('+601164271745', sms)  //send the message to the doctor
      saveAlert(sms)          //save the database to the firebase db

   if(pulse > 100):
      sms = "Patient has high pulse rate : " + str(pulse)
      sim800l.send_sms('+601164271745', sms)
      saveAlert(sms)
```

```
    if(pulse < 60):
        sms = "Patient has low pulse rate : " + str(pulse)
        sim800l.send_sms('+601164271745', sms)
        saveAlert(sms)


while 1:
    loop()
    time.sleep(10)
```

APPENDIX IC – The Code for the Dashboard of the Pulse and Temperature Sensors

firebase.js

```
// Initialize Firebase
var firebaseConfig = {
    apiKey: "AIzaSyAAawGHw2l1Q6krVGTlAcxUVBaszZyNRlc",
    authDomain: "health-monitoring-by-salini.firebaseapp.com",
    databaseURL: "https://health-monitoring-by-salini-default-rtdb.asia-
southeast1.firebasedatabase.app",
    projectId: "health-monitoring-by-salini",
    storageBucket: "health-monitoring-by-salini.appspot.com",
    messagingSenderId: "1097375343656",
    appId: "1:1097375343656:web:8a23d9a4f639f4c65f811d"
};
firebase.initializeApp(firebaseConfig);
```

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="./temp_gauge/style.css">
</head>


<body>

    <h1 style="text-align: center; font-family: Arial, Helvetica, sans-serif;">Health Monitoring
System - by Salini Pradhan</h1>
    <div class="container">
        <div class="de">
            <div class="den">
                <div class="dene">
                    <div class="denem">
                        <div class="deneme" id="temp">
                            0<span>.0</span><strong>&deg;</strong>
```

```
                </div>
              </div>
            </div>
          </div>
        </div>
    </div>
    <div id="chartContainer" style="height: 370px; width:100%;"></div>
    <script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.4.3/firebase.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.4.3/firebase-app.js"></script>
    <!-- If you enabled Analytics in your project, add the Firebase SDK for Analytics -->
    <script src="https://www.gstatic.com/firebasejs/8.4.3/firebase-analytics.js"></script>
    <!-- Add Firebase products that you want to use -->
    <script src="https://www.gstatic.com/firebasejs/8.4.3/firebase-database.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.4.3/firebase-firestore.js"></script>
    <script src="./firebase.js"></script>

    <script>
      var dps = []; // dataPoints
      var chart = new CanvasJS.Chart("chartContainer", {
        title: {
          text: "Pulse Sensor Data"
        },
        data: [{
          type: "line",
          dataPoints: dps
        }]
      });

      var xVal = 0;
      var yVal = 100;
      var updateInterval = 1000;
      var dataLength = 10;


      function updateChart(count, yVal) {

        count = count || 1;

        for (var j = 0; j < count; j++) {
          yVal = yVal;
          dps.push({
            x: xVal,
            y: yVal
          });
          xVal++;
        }

        if (dps.length > dataLength) {
          dps.shift();
        }

        chart.render();
      };
      updateChart(dataLength, 0);
    </script>
    <script>
```

```
        firebase.database().ref('sensor_data/').on('value', (snapshot) => {
            var temp = snapshot.child('temp').val();
            var tempVal = temp.toString();
            document.getElementById("temp").innerHTML = tempVal.split(".")[0] + "<span>." +
tempVal.split(".")[1] + "</span><strong>&deg;</strong>";
            updateChart(1, snapshot.child('pulse').val());
        });
    </script>
</body>

</html>
```

style.css (for the pulse sensor data)

```
@charset "UTF-8";
body {
  background-color: #1c2039;
}

.gauge {
  position: relative;
  width: 7em;
  height: 5.8em;
  margin: 0.5em auto;
  font-size: 5em;
  direction: ltr;
}

.gauge-inner {
  position: relative;
  top: -4.85em;
  opacity: 0;
  transition: opacity 0.5s;
}
.gauge.load .gauge-inner {
  opacity: 1;
}
.gauge-inner .bar {
  left: 50%;
  position: absolute;
  transform-origin: 0 2.5em;
  width: 0.03em;
  height: 0.2em;
  background: rgba(255, 255, 255, 0.2);
}
.gauge-inner .bar.peak {
  height: 0.35em;
}

.gauge-outer {
  position: relative;
  height: 100%;
  margin-top: 0.3em;
}
.gauge-outer .bar {
  position: absolute;
  width: 0.05em;
```

```
  height: 0.7em;
  left: 50%;
  transform-origin: 0 3.43em;
  background-color: rgba(0, 0, 0, 0.2);
  opacity: 0;
  transition: opacity 0.5s;
}
.gauge.load .gauge-outer .bar {
  opacity: 1;
}

.gauge-digits {
  position: absolute;
  height: 81%;
  width: 70%;
  top: 1.3em;
  left: 50%;
  transform: translateX(-50%);
  color: rgba(255, 255, 255, 0.4);
  opacity: 0;
  transition: opacity 0.5s 0.5s;
}
.gauge.load .gauge-digits {
  opacity: 1;
}
.gauge-digits.scale {
  transform: translateX(-50%) scale(0.8);
}
.gauge-digits .digit {
  position: absolute;
  font-size: 0.4em;
  line-height: 0.2;
}
.gauge-digits .current-digit {
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
  font-size: 1.5em;
  color: rgba(255, 255, 255, 0.9);
}
.gauge-digits .current-digit:after {
  content: "°";
}
```

style.css (for temperature sensor real-time data)

```
@import url(https://fonts.googleapis.com/css?family=Dosis:200,400,500,600);
html, body { height: 100%; }
body { background: #f2f2f2; }

.container { width: 300px; margin: 10px auto 0; }
.de .den, .de .dene, .de .denem, .de .deneme { position: absolute;  left: 50%; top: 50%; }
.de {
    position: relative;
    width: 240px;
    height: 240px;
```

```
    border-radius: 100%;
    box-shadow: 0 0 10px rgba(0, 0, 0, .1);
    background-color: transparent;
}
.den {
    position: relative;
    width: 210px;
    height: 210px;
    margin: -105px 0 0 -105px;
    border-radius: 100%;
    box-shadow: inset 0 2px 10px rgba(0, 0, 0, .5), 0 2px 20px rgba(255, 255, 255, 1);
    background: #df3341;
    background: -moz-linear-gradient(left, #df3341 0%, #d4f355 50%, #61c0ec 100%);
    background: -webkit-gradient(linear, left top, right top, color-stop(0%,#df3341), color-stop(50%,#d4f355), color-stop(100%,#61c0ec));
    background: -webkit-linear-gradient(left, #df3341 0%,#d4f355 50%,#61c0ec 100%);
    background: linear-gradient(to right, #df3341 0%,#d4f355 50%,#61c0ec 100%);
    position:relative;
}
.dene {
    width: 180px;
    height: 180px;
    margin: -90px 0 0 -90px;
    border-radius: 100%;
    box-shadow: inset 0 2px 2px rgba(255, 255, 255, .4), 0 3px 13px rgba(0, 0, 0, .85);
    background: #f2f6f5;
    background: -moz-linear-gradient(top, #f2f6f5 0%, #cbd5d6 100%);
    background: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #f2f6f5), color-stop(100%, #cbd5d6));
    background: -webkit-linear-gradient(top, #f2f6f5 0%, #cbd5d6 100%);
    background: -o-linear-gradient(top, #f2f6f5 0%, #cbd5d6 100%);
}
.denem {
    width: 160px;
    height: 160px;
    margin: -80px 0 0 -80px;
    border-radius: 100%;
    background: #cbd5d6;
    background: -moz-linear-gradient(top, #cbd5d6 0%, #f2f6f5 100%);
    background: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #cbd5d6), color-stop(100%, #f2f6f5));
    background: -webkit-linear-gradient(top, #cbd5d6 0%, #f2f6f5 100%);
}

.deneme {
    padding: 3px 10px 0 10px;
    width: 120px;
    height: 137px;
    display: inline-block;
    margin: -70px 0 0 -70px;
    color: #555;
    text-shadow: 1px 1px 1px white;
    font-family: 'Dosis';
    font-size: 100px;
    font-weight: 400;
    text-align: center;
}
```

```
.deneme span { font-size: 30px; font-weight: 200; }
.deneme strong { position: absolute; right: 10px; top: 25px; font-size: 34px; }
```

APPENDIX II – Gantt Chart

ACTIVITIES	TIMELINE										
	FYP 1 (2021)					ACTIVITIES	FYP 2 (2021 – 2022)				
	APRIL	MAY	JUNE	JULY	AUGUST		OCTOBER	NOVEMBER	DECEMBER	JANUARY	FEBRUARY
Literature Review.						Implementation and Development.					
Research Proposal.						System Testing.					
Research Proposal Presentation.						Submission of Progress Report and Presentation.					
Research Proposal Report Submission.						Modification.					
Research on the Existing Systems and Required Components.						Testing.					
Project Progress Presentation.						Documentation.					
FYP 1 Presentation.						Final Presentation.					
Final version report submission.						Final Report submission.					

Table 8: Gantt Chart for FYP1 and FYP2