LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CT5053NI - Cloud Computing & IoT

Assessment Type

Main Report

Semester

2022 Autumn/Spring

Group Members:

London Met ID	Student Name
21040628	Shribisha Gartaula
21040602	Arjay Bikram Khand
21040622	Samyug Raj Pandeya
21040618	Raj Sagar Thakur
21040614	Nirdesh Budhathoki

Assignment Due Date: 4th May 2023

Assignment Submission Date: 4th May 2023

Word Count: 2925 words

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgment

We would like to convey our sincere gratitude to our Cloud Computing and IoT Module Leader and Tutor Mr. Shishir Subedi Sir and Mr. Sugat Man Shakya Sir for giving us the chance to complete this Group Project and for their continued support for the project. Their suggestions and support were extremely helpful in completing our group project and report.

We would like to express our gratitude to our seniors, teammates, and everyone who helped us and are helping us to edit and correct the errors that kept us dwelling on the Testing part and on the logical part of this group project also We would also like to express our appreciation to them for encouraging us to carry out research related IoT prototypes and also for suggesting us about **Heart-B** for research and report development. We got to learn a lot of new and unknown things.

Similarly, we would also like to thank all of our classmates who helped us complete the project by exchanging interesting ideas and sharing their experiences. We want to thank our friends who displayed appreciation for our work and motivated us to continue our work. Last, but not the least, we are grateful to our parents for their support and encouragement, without which We could not have completed this project in the limited timeframe.

Abstract

This report presents the development of an Internet of Things (IoT)-enabled heart rate monitoring system that aims to address the growing concern of heart disease in Nepal. The prototype is designed to provide real-time and remote monitoring of heart rate data, enabling healthcare providers to identify and respond to heart-related issues quickly. The system is also equipped with data analytics tools that can collect and analyse large amounts of heart rate data, providing valuable insights into heart health trends, risk factors, and potential interventions.

The heart rate monitoring system is cost-effective and can be integrated with mobile and web-based applications, making it accessible and easy to use for a wider range of people in Nepal, particularly those in remote or underserved areas. The system's remote consultation feature reduces the need for in-person visits, further increasing its accessibility. By improving heart rate monitoring, the system has the potential to improve heart health outcomes in Nepal, reduce healthcare costs, and increase the reach of healthcare services. Overall, this report highlights the potential of IoT-enabled heart rate monitoring systems in addressing healthcare challenges, particularly in resource-limited settings.

Table Of Contents

List of Figures

List of Tables

1. Introduction

The term "Internet of Things" (IoT) refers to a network of physical things that are wired together and outfitted with sensors, software, and other technological components to connect to other network devices or the cloud and exchange data. (Oracle, 2022).

Our project, named “Heart-B”, is also based on the principle of IoT with the primary objective of making notes and reports of our daily heart rate easier and highly available whenever needed. The main components used in this project are an Arduino Uno, a Heartbeat rate sensor, an ESP-8266 Wi-Fi module, a Resistor, and a LED.

1.1 Current Scenario

In Nepal's current situation, the absence of IoT-enabled heart rate monitoring devices may have negative impacts on both the country's individual and general health. Devices with IoT heart rate monitoring capabilities can be helpful for patients, medical professionals, and public health officials. Real-time information about a person's heart rate can be obtained from these gadgets, which can be used to monitor heart health while exercising and track the efficacy of treatment.

Without access to such devices, individuals may not be able to monitor their heart health as effectively, which could lead to a delay in the detection and treatment of heart disease. Additionally, healthcare providers may not have access to real-time heart rate data for their patients, which could impact their ability to make informed decisions about their care.

Regarding the broader impact on public health, public health professionals may find it more challenging to track trends in heart disease at the population level if they lack access to IoT-enabled heart rate monitoring devices. This might hinder their capacity to create efficient heart disease prevention and treatment plans for Nepal.

1.2 Problem Statement and Project as A Solution

Problem Statement:

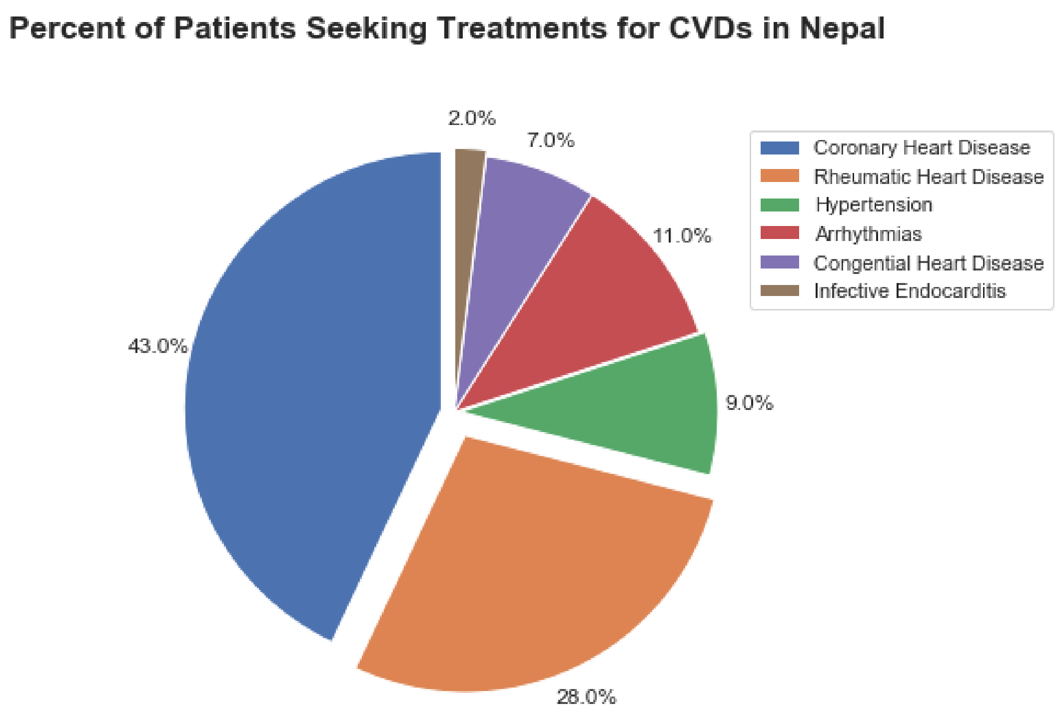

Figure 1: heart diseases patient *(Clinic One, 2023)*

Heart disease is a major health concern in Nepal, and the lack of access to medical facilities in remote areas makes it difficult to diagnose and manage heart-related conditions. Traditional heart rate monitoring methods are expensive and require regular in-person visits to a healthcare provider, making it challenging for people to monitor their heart health regularly.

Solution:

An IoT-enabled heart rate monitoring system can be developed as a solution to these challenges. The system can be designed to provide real-time and remote monitoring of heart rate data, enabling healthcare providers to identify and respond to heart-related issues quickly. The system can also collect and analyse large amounts of heart rate data, providing valuable insights into heart health trends and risk factors. By making heart rate monitoring accessible and affordable, the system can help improve heart health outcomes in Nepal, particularly in remote and underserved areas.

2. Aims and Objectives

2.1 Aim

The main aim of our project Heart-B is to make a portable Heartbeat Sensor that anyone can carry around wherever and wherever they go and keep track/record of their heart rate and help the user to keep note of it and maintain it to their normal range.

2.2 Objectives

The objectives of our group project are:

- To explain the heart rate monitoring system's operation
- To describe the system's technology and methods
- To evaluate the heart rate monitoring system's accuracy and precision
- To offer suggestions for efficiently utilizing the system
- To draw attention to any hazards or negative effects that may be experienced when using the system.
- To educate people on the advantages of heart rate monitoring systems for promoting cardiovascular health and wellness, including medical professionals, researchers, and the public.

3. Background

3.1 System Overview

This study created a useful technique for lowering the incidence of heart attacks or irregular heart rates brought on by any heart disease condition. In this project, we address this issue by developing a system that monitors and records a patient's or user's heartbeat rate. This heartbeat rate detector lets the user or patient know whether they have an abnormal or problematic cardiac condition.

Various devices were used in this project to assist in its completion. To connect the breadboard, pulse sensor, and wi-fi module, Arduino was applied. Similarly, a breadboard was connected to the Arduino, and the necessary resistors and LEDs were added to the code. Similarly, a pulse sensor was attached to the breadboard and Arduino to measure the heart rate shown by the breadboard's LED.

Similarly, a wi-fi module was used to connect Arduino and code it as needed. Similarly, LEDs were used to detect heartbeats, which were recorded using a pulse sensor. To light the LEDs, resistors were used so that any volt given could light the LED. Finally, jump wires were used to connect the Arduino to the breadboard, the pulse sensors, and the wireless network module.

3.2 Design Diagram

3.2.1 Hardware Architecture

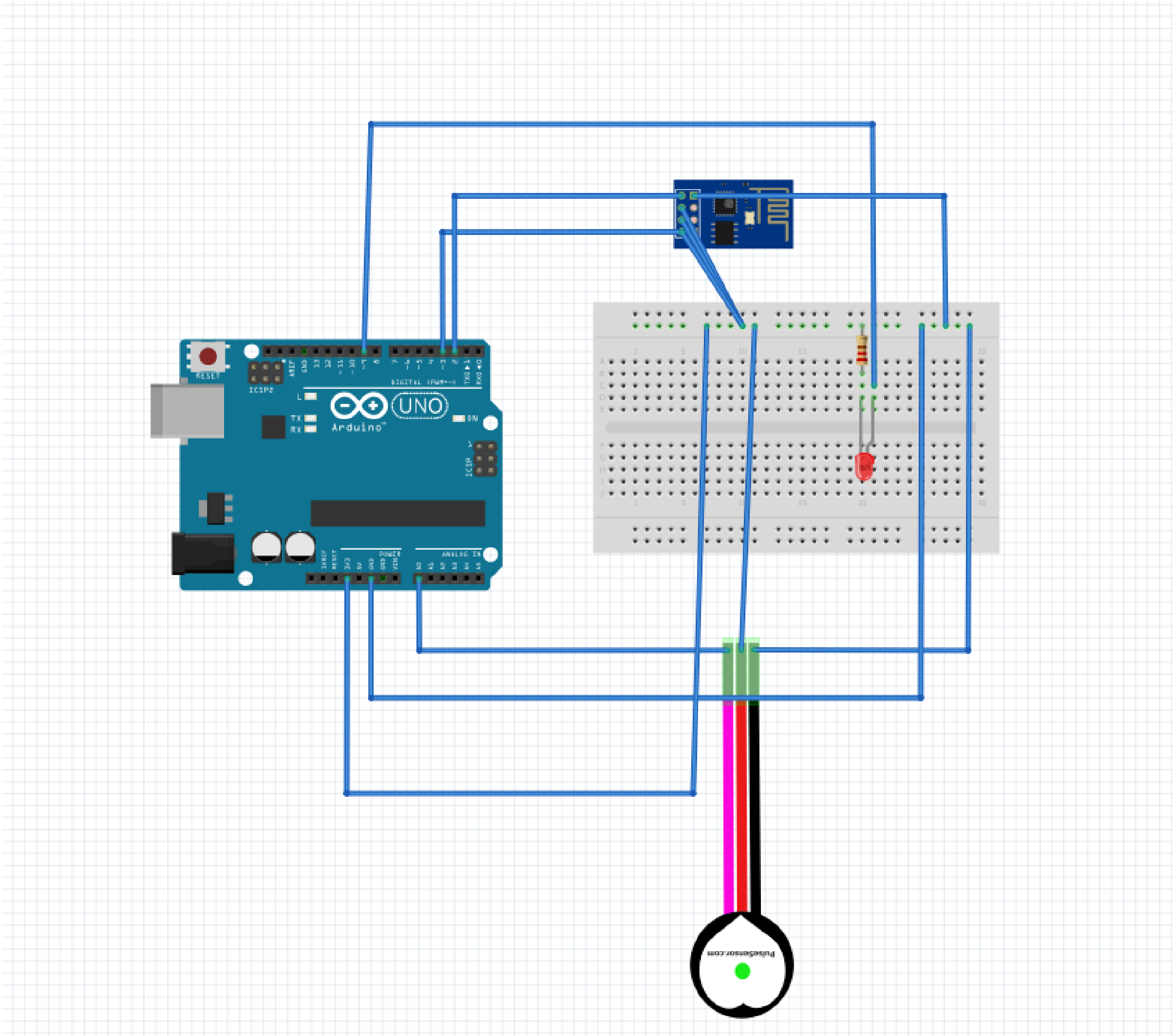

Figure 2: Hardware Architecture

In this project, an ESP8266 Wi-Fi module-based online platform called ThingSpeak will receive data from an Arduino Uno-based Heart Rate Monitoring System that uses Pulse Sensor to determine heart rate. Our sensing data is displayed using ThingSpeak over the internet at any moment and from any location. Being able to present real-time data without any latency gives it an advantage over other IoT platforms.

Pin Name	Arduino Pin
Wi-Fi Module Vin	3.3 V
Wi-Fi Module RST	3.3 V
Wi-Fi Module CH-PD	3.3 V
Wi-Fi Module RX	TX
Wi-Fi Module TX	RX
Wi-Fi Module GND	GND
Pulse Sensor Vin	3.3 V
Pulse Sensor GND	GND
LED +ve	9
LED -ve	GND

Table 1: Connecting Device integration.

3.2.2 Circuit Diagram

A circuit diagram, also called an electrical diagram, elementary diagram, or electronic schematic, is a simplified graphical representation of an electrical circuit. Circuit diagrams are used for the design, construction, and maintenance of electrical and electronic equipment (BYJU's, 2023).

The diagram shown in below is the circuit diagram of IOT project:

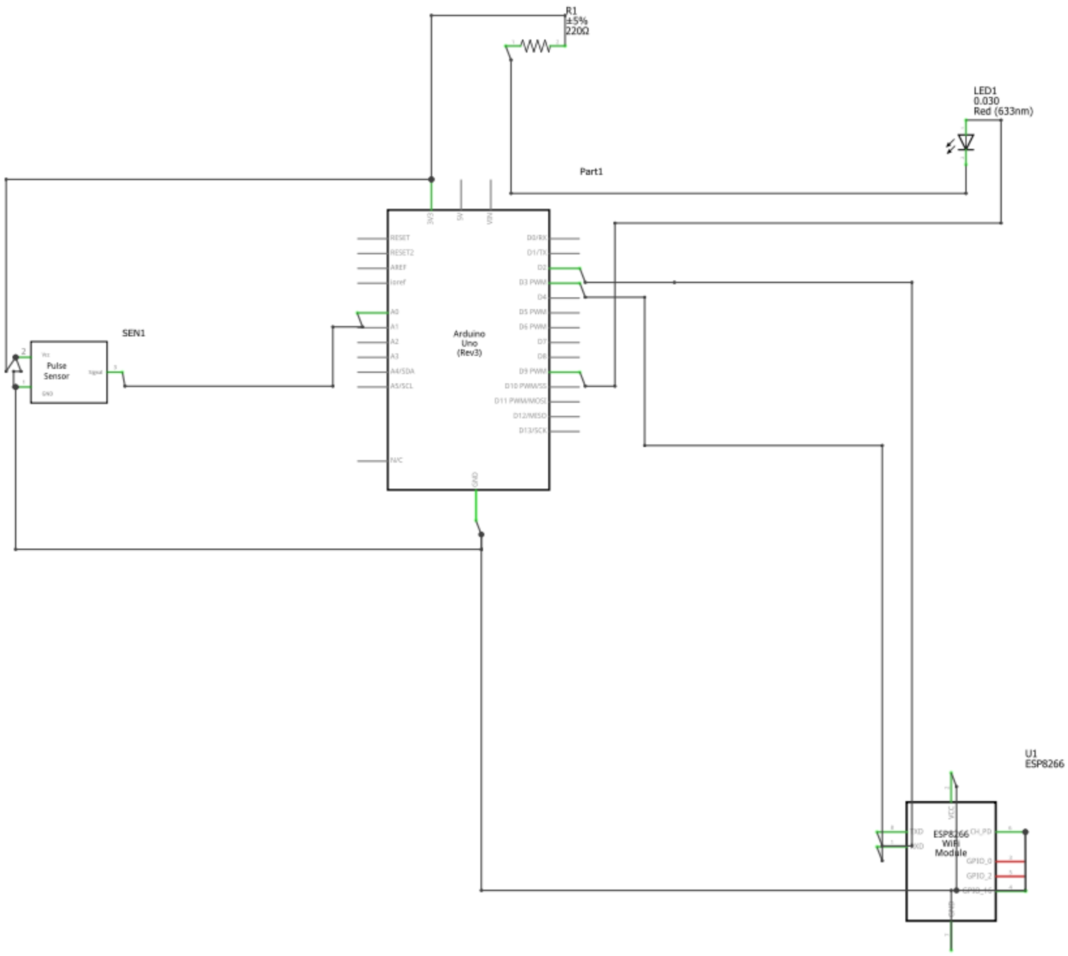

Figure 3: Circuit Diagram

3.2.3 Flowchart

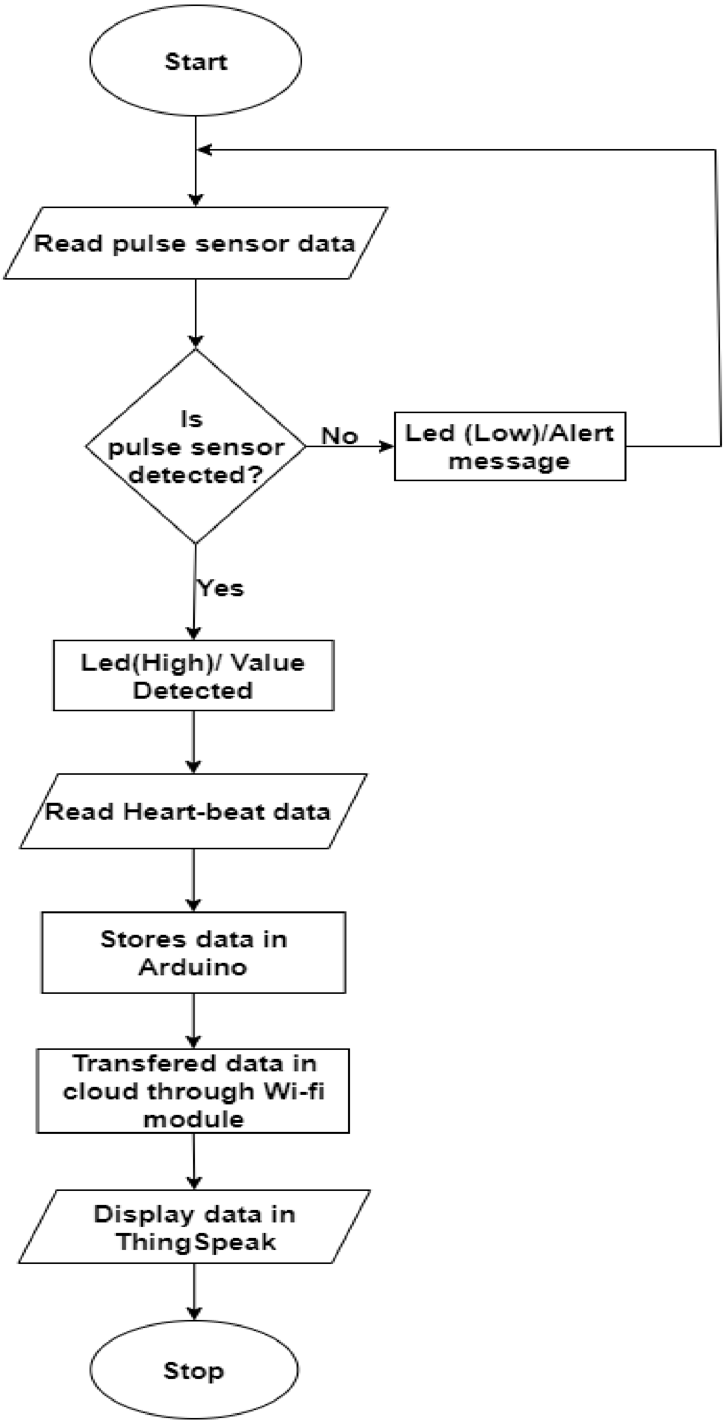

Figure 4: Flowchart

3.3 Requirement Analysis

3.3.1 Hardware

- **Arduino Uno**:

  The system's brain, Arduino, processes the data from the sensor. To creating projects, Arduino is an open-source hardware platform that is easily accessible to enthusiasts & amateurs throughout the world. It includes an ATMEGA microcontroller, which processes data and supports the IoT system's efficient operation. And Arduino can be programmed an infinite number of times, allowing us to create a variety of Internet of Things projects by simply altering a single line of code (Skyfi Labs). It uses the C++ simplified version. In addition to 6 analog pin inputs, it has 14 digital pins, a USB connector, a power jack, and an ICSP header.

  Arduino will send the heartbeat rate data to the cloud computing platform which is ThingSpeak using NodeMCU.

Figure 5: Arduino UNO (AzureFlim, 2022)

- **ESP-8266 Wi-Fi module**:

We need to use the ESP-8266 Wi-Fi module to establish the Wi-Fi communication between the Arduino and the cloud platform.

Figure 6: ESP-8266 Wi-Fi module (ElectronicWings, 2022)

- **Jump Wires**:

Jumper wires are electrical cables with connector pins on both ends. Without the use of solder, they are utilized to link two circuit locations. Jumpers are tiny metal connections that are used to open and close circuit components. A circuit board for an electrical system is controlled by two or more connecting points on them. The motherboard and other computer hardware must be configured by them (Wiltronics).

Figure 7: Jump Wires (Adruino, 2022)

- **Bread Board**:

A breadboard is a thin plastic board used to contain wired-together electrical components such as transistors, resistors, chips, etc. It is a solderless instrument used to make temporary electrical prototypes and test circuit designs.

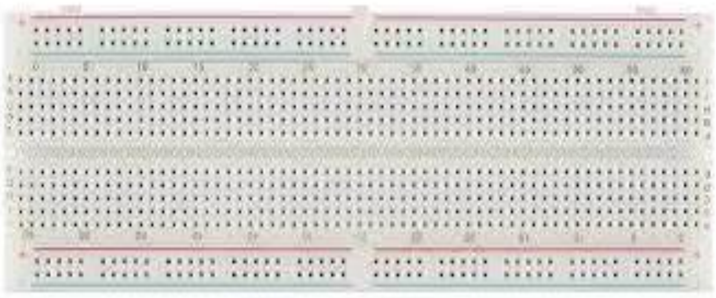

Figure 8: Bread Board (Adafruit, 2023)

- **LED (Light Emitting Diode)**:

A Light Emitting Diode (LED) is a semiconductor device, which can emit light when an electric current passes through it. We used it to test the circuit if it is working properly or not.

Figure 9: Light Emitting Diode (Walmart, 2022)

- **Heart Rate Sensor**:

A well-designed, low-power, plug-and-play heart-rate sensor for the Arduino is available as the Pulse Sensor. Students, artists, athletes, makers, game developers, and others who want to use real-time heart-rate data in their work can all gain from it. Additionally, this sensor is simple to clip onto a fingertip or earlobe and directly plugs into Arduino.

Figure 10: Heart Rate Sensor (Whadda, 2022)

- **220-ohm Resistor**:

A 220-ohm resistor is an electronic component that is used to resist the flow of electricity in a circuit.

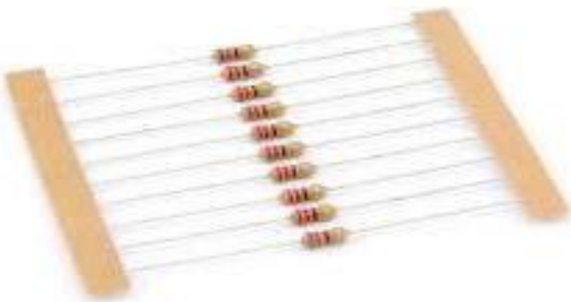

Figure 11: Resistor (Chip, 2023)

3.2.2 Software

- **Arduino IDE**:

  Arduino is an open-source platform that uses a C++-based programming language, making it simple to use for both beginners and experts.

  We used it for coding the program for our project.

Figure 12: Arduino IDE (Github, 2022)

- **Tinker CAD**:

  It is an online platform where we can create the design and simulate the circuits for our project. We used to make 3D circuit diagrams.

Figure 13: Tinker CAD (CAD, 2022)

- **Draw.io**:

  Draw.io is proprietary software for making diagrams and charts. The software lets you choose from an automatic layout function or create a custom layout.

  We used Draw.io to distribute the task and make a detailed plan.

Figure 14: Draw.io (Diagrams.net, 2022)

- **Google Docs**:

  Google Docs is the most used software that we used to complete all our documentation and reporting using suitable headings and indentations.

Figure 15: Google Docs (Google, 2022)

- **ThingSpeak**:

  ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyse live data streams in the cloud.

  We used Arduino UNO to send data from our devices, create instant visualizations of live data, and send alerts to ThingSpeak which shows real-time data without lagging.

Figure 16: ThingSpeak (ThingSpeak, 2022)

4. Development

4.1 Planning and Design

The initial phase in developing a heart rate monitoring system for an IoT project is to establish the system's requirements. This includes determining the target people, test rate, desired level of accuracy, and acceptable amount of error. Following the identification of the requirements, the next step is to select the appropriate sensors for measuring the heart rate. The sensors chosen must be precise, dependable, and should not give the user any discomfort.

In an IoT project, the design of a heart rate monitoring system involves integrating the sensors with the IoT platform. This is possible with wireless communication protocols such as Wi-Fi. The sensors' data is delivered to the IoT platform, where it is processed and analysed. Healthcare professionals and individuals can access the processed data via a web-based interface or a mobile application.

It is critical to test the sensors on a regular basis to guarantee the reliability and precision of the heart rate monitoring system. Standardisation is comparing sensor data to a known measurement standard and modifying the sensors accordingly. Regular maintenance of the system is also necessary to maintain its long-term reliability and functionality.

4.2 Resource Collection

A variety of devices and supplies were required to completely show the development of this project. Those devices were to be managed from various locations. Moreover, half of the devices were operated by Islington College's resource department. Each member of the group contributed in to help identify materials for this project. The equipment purchased from Islington College's resource department include:

1. Arduino Uno
2. Breadboard
3. Wi-Fi Module
4. LEDs
5. Jump Wires
6. Resistors

Only the pulse sensor, which was one of the main devices in our IOT project, was obtained from an external source.

4.3 System Development

This IOT project system development follows the same steps as seen in the 3-D graphic. First, we connected the Arduino Uno to the breadboard by connecting the ground and 5 voltage pins of the Arduino to the positive and negative terminals of the breadboard. Following that, the Wi-Fi module is connected to the Arduino Uno and Breadboard, and electricity is passed through the microcontroller to the Wi-Fi module. And LED and Resistors is being Connected to check the Detection of sensor and maintain the ground level of voltage in breadboard.

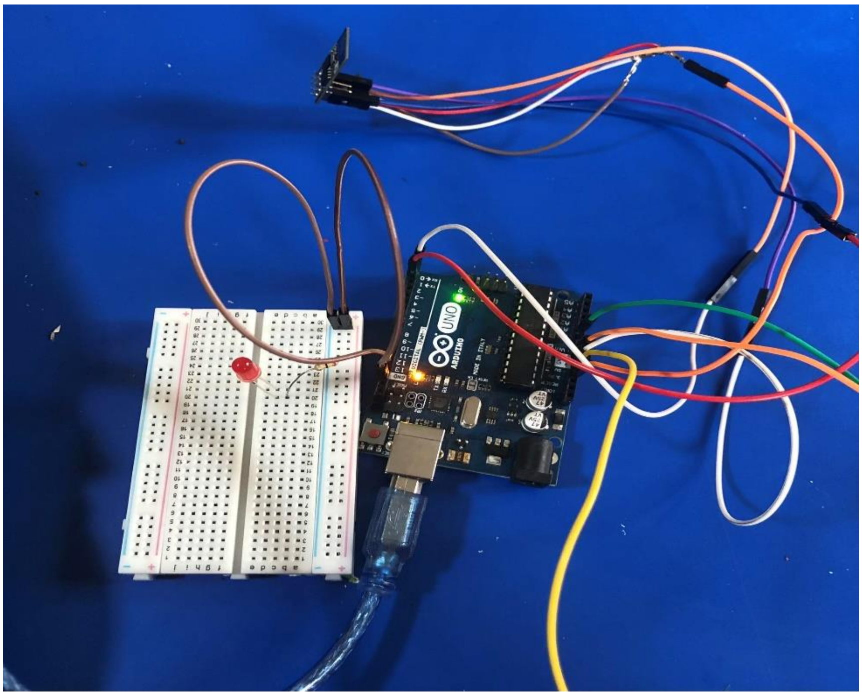

Figure 17: Assemble of Arduino, Wi-fi module and breadboard.

The Arduino and breadboard are then linked to the pulse sensor. The pulse sensor's ground pin is linked to the breadboard's ground pin, and the pulse sensor's vin pin is connected to the breadboard's 5-volt pin. These pins were wired together to power the pulse sensor. Meanwhile, the pulse sensor's A0 pin is linked to Arduino's analogue pin A0 to give input signals beat.

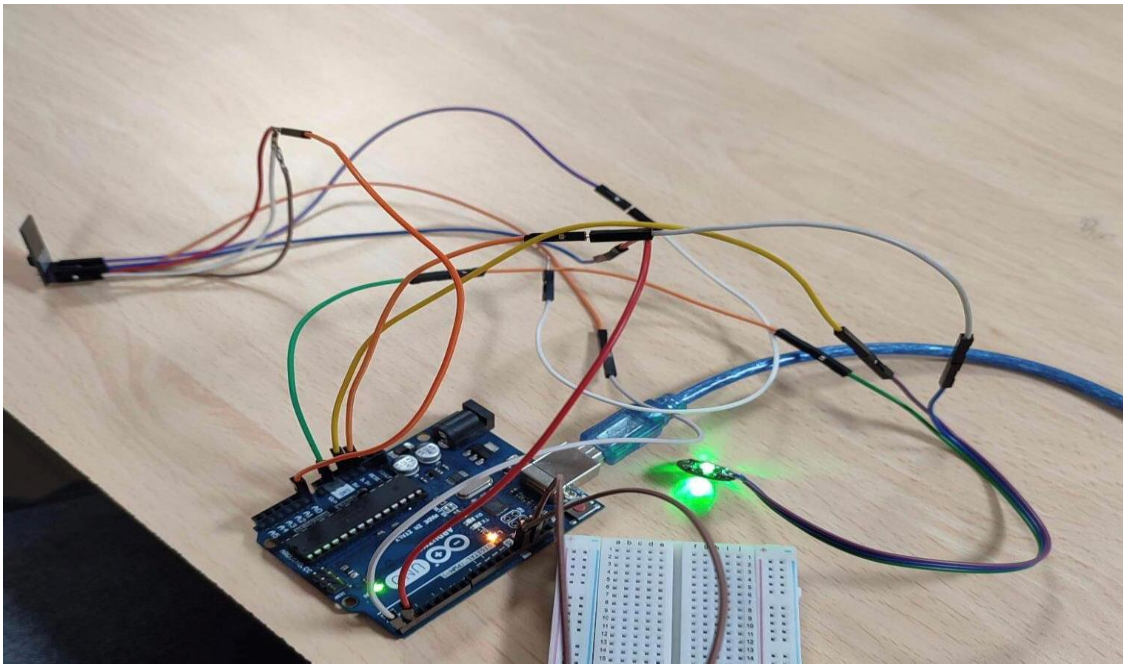

Figure 18: Assemble of Arduino, pulse sensor, breadboard and wi-fi module.

The other components are being joined together. To increase the system's reliability, jump wires are connected to the Wi-Fi module and sensor. Finally, a programme code is executed in the Arduino IDE software to properly run the system.

4.4 Types of Sensors and Actuator used in project.

The pulse sensor, commonly known as the heart rate sensor, was used in the project. A pulse sensor is an analogue sensor that measures pulse and heart rate. To evaluate blood flow, a pulse sensor is placed in the fingertip and analyses variations in light absorption and reflection onto the skin.

Light Emitting Diode (LED) actuators have been used in the project. LED is a digital actuator that emits light when current runs through it. This digital actuator has two states: on and off. A value of 1 activates the LED, whereas a value of 0 turns it off.

5. Result and Findings

We initially considered the idea of a cloud-based heart rate observation system when we started the project. The project's final output was a practical framework that made advantage of Internet of Things (IoT) innovation. When the pulse rate is detected, the framework causes a Driven to squint, indicating the position of a heartbeat.

The final design included locations for a Led that glows when a pulse rate is detected. To ensure the system's accuracy and use, extensive testing was done, yielding a successful outcome.

The heart rate sensor isn't operating or not taking measurements, as shown by a blinking Lens. However, if the Powered stops blinking, it means the sensor is functioning normally and gathering heart rate data. The chosen data is transmitted to the cloud through the Wi-Fi module and shown on the ThingSpeak stage.

5.2 Testing

5.2.1 Test 1

Test	1
Objective	To check the working of LED light
Activity	LED blink
Expected Result	Led will blink if the data is being detected
Actual result	Led was blinked
Conclusion	The test was successful.

Table 2: Test-1

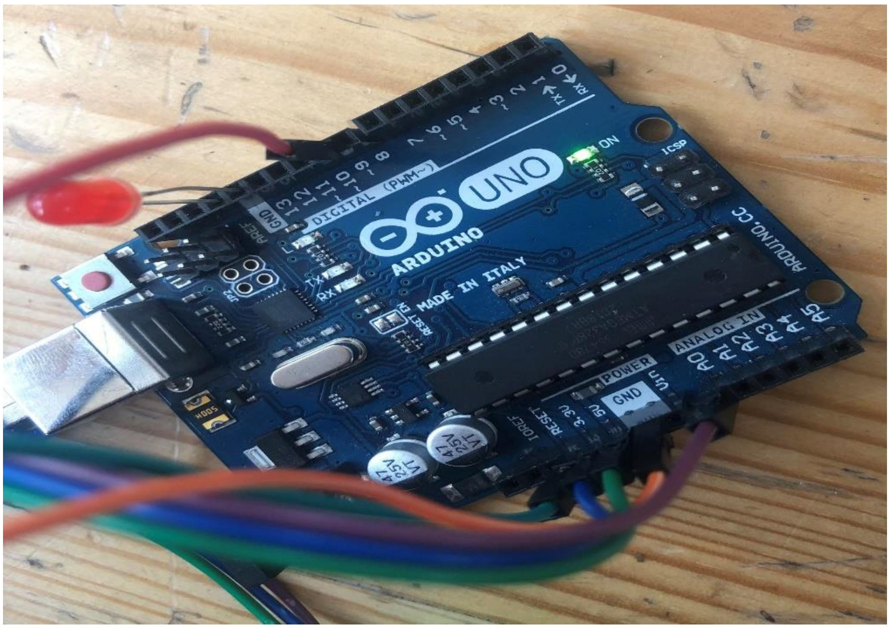

Figure 19: Test-1

5.2.2 Test 2

Test	2
Objective	To check working of pulse sensor
Activity	Green light will be turn on
Expected Result	Green light will show if the sensor is linked properly
Actual result	Green light was shown in the sensor
Conclusion	The test was successful.

Table 3: Test-2

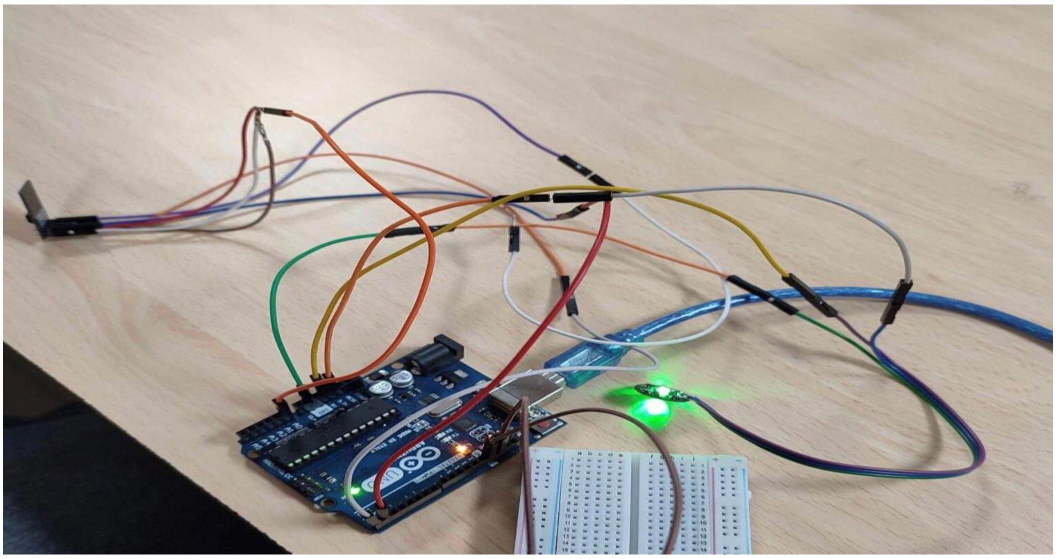

Figure 20: Test-2

5.2.3 Test 3

Test	3
Objective	To show the execution of cloud information
Activity	Heart rate data will be shown in cloud
Expected Result	Pulse rate will be display in ThingSpeak
Actual result	The pulse rate was displayed
Conclusion	The test was successful.

Table 4: Test-3

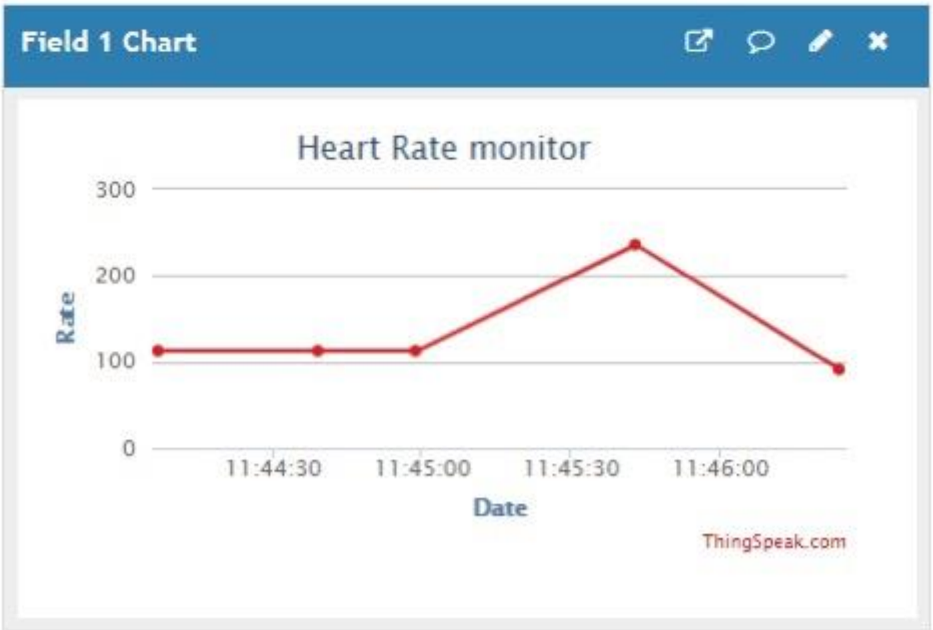

Figure 21: Test-3

6. Future works

Although there are many areas that can be improved to make things simpler, more portable, and more accurate, our Heart-B system can be considered fully functional. Users won't need a larger screen or monitor to view their data in the future thanks to the addition of the display, which will simplify things for them. The Bluetooth system and alerting software are additional components that can be added to the system to send messages and heartbeat rate data to the user or the patient's close relative to monitor their heart health. The alerting software can also send alerting messages to the patient's guardian if there is something abnormally wrong with the patient's heart rate. This device, which may be integrated into a smartwatch or Fitbit band, can determine whether a patient's heart rate is typical for their age, height, and weight. The sensor will only be able to detect the patient's heartbeat rate and can record it constantly at predetermined intervals of time since it will be in contact with the patient's wrist (in the case of a smartwatch).

Not only that, but we also can use GUI Interface in the future to add more users and connect them or their recorded data through Bluetooth on their own devices. We could upload data to AWS for backup on a daily to weekly basis. We can also encrypt data on the user’s side to maintain their privacy and keep records away from unauthorized users.

7. Conclusion

In this coursework, we were assigned to do an IoT Prototype Project. As this was group coursework, we were divided into random groups. So, in the beginning, we faced several challenges such as team coordination and cooperation between the members. We faced other problems like a shortage of several components required in the project. But eventually, our teamwork was good enough to make this project successful.

We learned about using software such as Arduino IDE, Tinker CAD, and Draw.io and hardware components such as Arduino Uno, LED, Heart Rate Sensor, Resistor, Wi-Fi module (ESP8266), Jumper wires, and Bread Board, etc. Overall, this coursework helped us gain knowledge about many useful things which will help us in the future. Working as a team, we also gained more knowledge about teamwork and work distribution. All the members greatly contributed to solving the problems and making a proper report including all the required documentation. In conclusion, we gained a lot of experience and had a lot of fun while completing the project with full coordination from everyone.

8. References

Adafruit, 2023. *Adafruit.* [Online]
Available at: learn.adafruit.com
[Accessed 2 April 2023].

Adruino, E., 2022. *Exploring Arduino.* [Online]
Available at: www.exploringarduino.com
[Accessed 2 April 2023].

AzureFlim, 2022. *AzureFlim.* [Online]
Available at: www.azureflim.com
[Accessed 2 April 2023].

CAD, T., 2022. *Tinker CAD.* [Online]
Available at: www.tinkercad.com
[Accessed 2 April 2023].

Chip, T., 2023. *Twins Chip.* [Online]
Available at: www.twinschip.com
[Accessed 2 April 2023].

Diagrams.net, 2022. *Diagrams.net.* [Online]
Available at: www.diagrams.net
[Accessed 2 April 2023].

ElectronicWings, 2022. *ElectronicWings.* [Online]
Available at: www.electronicwings.com
[Accessed 2 April 2023].

Github, 2022. *Github.* [Online]
Available at: www.github.com
[Accessed 2 April 2023].

Google, 2022. *GoogleDocs.* [Online]
Available at: www.google.com
[Accessed 2 April 2023].

Labs, S., 2022. *Skyfy Labs.* [Online]
Available at: www,skyfilabs.com
[Accessed 2 April 2023].

Oracle, 2022. *Oracle.* [Online]

Available at: www.oracle.com

[Accessed 2 April 2023].

Skyfi Labs, 2019. *What is Arduino in IoT?.* [Online]

Available at: https://www.skyfilabs.com/resources/what-is-arduino-in-iot

[Accessed 20 March 2023].

ThingSpeak, 2022. *ThingSpeak.* [Online]

Available at: www.thingspeak.com

[Accessed 2nd April 2023].

Walmart, 2022. *Walmart.* [Online]

Available at: www.walmart.ca

[Accessed 2 April 2023].

Whadda, 2022. *Whadda.* [Online]

Available at: www.whadda.com

[Accessed 2 April 2023].

Wiltronics, 2022. *What Are Jumper Wires: Colour, Types and Uses.* [Online]

Available at: https://www.wiltronics.com.au/wiltronics-knowledge-base/what-are-jumper-wires/

[Accessed 20 March 2023].

BYJU's, 2023. *BYJU's.* [Online]

Available at: www.byjus.com

[Accessed 2 May 2022].

Clinic One, 2023. *Clinic One.* [Online]

Available at: www.clinicone.com.np

[Accessed 2 May 2023].

9. Appendix

9.1 Source Code

```
#include <SoftwareSerial.h>
#include <stdlib.h>
#define DEBUG true
SoftwareSerial ser(2,13);
#define SSID "Islington-College"     // Enter Your WiFi Name Here
#define PASS "I$LiNGT0N2023"       // Enter Your WiFi Password Here
#define IP "184.106.153.149"// thingspeak.com ip

String msg = "GET https://api.thingspeak.com/update?api_key=R8AUP1KHA41L5K0J&field1=0"; //Enter your API key

//Variables
int error;
int sensorPin = 0;         // Connect Pulse Sensor Signal Pin to Analog Pin A0
int ledpin = 9;        //  Connect Led Positive Pin to Arduino Pin 9

volatile int BPM;           // int that holds raw Analog in 0. updated every 2mS
volatile int Signal;              // holds the incoming raw data
volatile int IBI = 600;     // int that holds the time interval between beats
volatile boolean Pulse = false;     // "True" when heartbeat is detected. "False" when not a "live beat".
volatile boolean QS = false;       // becomes true when Arduino finds a beat.

static boolean serialVisual = true;   // Set to 'false' by Default.  Re-set to 'true' to see Arduino Serial Monitor ASCII Visual Pulse
volatile int rate[10];                  // array to hold last ten IBI values
volatile unsigned long sampleCounter = 0;            // used to determine pulse timing
```

volatile unsigned long lastBeatTime = 0; // used to find IBI

volatile int P =512; // used to find peak in pulse wave, seeded

volatile int T = 512; // used to find trough in pulse wave, seeded

volatile int thresh = 525; // used to find instant moment of heart beat, seeded

volatile int amp = 100; // used to hold amplitude of pulse waveform, seeded

volatile boolean firstBeat = true; // used to seed rate array so we startup with reasonable BPM

volatile boolean secondBeat = false; // used to seed rate array so we startup with reasonable BPM

```
void setup()  {
  Serial.begin(115200); //or use default 115200.
  ser.begin(115200);
  Serial.println("AT");
  ser.println("AT");
  delay(3000);
  if(ser.find("OK"))  {
    connectWiFi();
  }
  interruptSetup();
}

void loop() {
  start: //label
  error=0;
  updatebeat();
  //Resend if transmission is not completed
  if (error==1) {
    goto start;
  }

  delay(1000);
}
```

```
void updatebeat() {
  String cmd = "AT+CIPSTART=\"TCP\",\"";
  cmd += IP;
  cmd += "\",80";
  Serial.println(cmd);
  ser.println(cmd);
  delay(2000);
  if(ser.find("Error")) {
    return;
  }
  cmd = msg ;
  cmd += "&field1=";
  cmd += BPM;
  cmd += "\r\n";
  Serial.print("AT+CIPSEND=");
  ser.print("AT+CIPSEND=");
  Serial.println(cmd.length());
  ser.println(cmd.length());
  if(ser.find(">")) {
    Serial.print(cmd);
    ser.print(cmd);
  }
  else{
   Serial.println("AT+CIPCLOSE");
   ser.println("AT+CIPCLOSE");
    //Resend...
    error=1;
  }
}

boolean connectWiFi() {
  Serial.println("AT+CWMODE=1");
```

```
    ser.println("AT+CWMODE=1");
    delay(2000);
    String cmd="AT+CWJAP=\"";
    cmd+=SSID;
    cmd+="\",\"";
    cmd+=PASS;
    cmd+="\"";
    Serial.println(cmd);
    ser.println(cmd);
    delay(5000);
    if(ser.find("OK"))  {
      Serial.println("OK");
      return true;
    }else {
      return false;
    }
  }

  void interruptSetup() {
     // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
  }
  ISR (TIMER2_COMPA_vect){                    // triggered when Timer2 counts to
124
    cli();                                   // disable interrupts while we do this
    Signal = analogRead(sensorPin);             // read the Pulse Sensor
    sampleCounter += 2;                         // keep track of the time in mS
    int N = sampleCounter - lastBeatTime;        // monitor the time since the last
beat to avoid noise
      //  find the peak and trough of the pulse wave
    if(Signal < thresh && N > (IBI/5)*3){       // avoid dichrotic noise by waiting 3/5
of last IBI
      if (Signal < T){                        // T is the trough
        T = Signal;                           // keep track of lowest point in pulse wave
```

```
    }
  }
  if(Signal > thresh && Signal > P){        // thresh condition helps avoid noise
    P = Signal;                             // P is the peak
  }                                         // keep track of highest point in pulse wave



  if (N > 250){                             // avoid high frequency noise
    if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
      Pulse = true;                         // set the Pulse flag when there is a pulse
      digitalWrite(ledpin,HIGH);            // turn on pin 13 LED
      IBI = sampleCounter - lastBeatTime;   // time between beats in mS
      lastBeatTime = sampleCounter;         // keep track of time for next pulse
      if(secondBeat){                       // if this is the second beat
        secondBeat = false;                 // clear secondBeat flag
        for(int i=0; i<=9; i++){            // seed the running total to get a realistic BPM
at startup
          rate[i] = IBI;
        }
      }
      if(firstBeat){                        // if it's the first time beat is found
        firstBeat = false;                  // clear firstBeat flag
        secondBeat = true;                  // set the second beat flag
        sei();                              // enable interrupts again
        return;                             // IBI value is unreliable so discard it
      }
      word runningTotal = 0;                // clear the runningTotal variable
      for(int i=0; i<=8; i++){              // shift data in the rate array
        rate[i] = rate[i+1];                // and drop the oldest IBI value
        runningTotal += rate[i];            // add up the 9 oldest IBI values
      }
      rate[9] = IBI;                        // add the latest IBI to the rate array
      runningTotal += rate[9];              // add the latest IBI to runningTotal
```

```
            runningTotal /= 10;                    // average the last 10 IBI values
            BPM = 60000/runningTotal;              // how many beats can fit into a minute?
that's BPM!
            QS = true;                             // set Quantified Self flag

          }
        }
        if (Signal < thresh && Pulse == true){   // when the values are going down, the
beat is over
          digitalWrite(ledpin,LOW);            // turn off pin 13 LED
          Pulse = false;                       // reset the Pulse flag so we can do it again
          amp = P - T;                         // get amplitude of the pulse wave
          thresh = amp/2 + T;                  // set thresh at 50% of the amplitude
          P = thresh;                          // reset these for next time
          T = thresh;
        }
        if (N > 2500){                         // if 2.5 seconds go by without a beat
          thresh = 512;                        // set thresh default
          P = 512;                             // set P default
          T = 512;                             // set T default
          lastBeatTime = sampleCounter;        // bring the lastBeatTime up to date
          firstBeat = true;                    // set these to avoid noise
          secondBeat = false;                  // when we get the heartbeat back
        }
        sei();

      }
```

9.2 Block Diagram

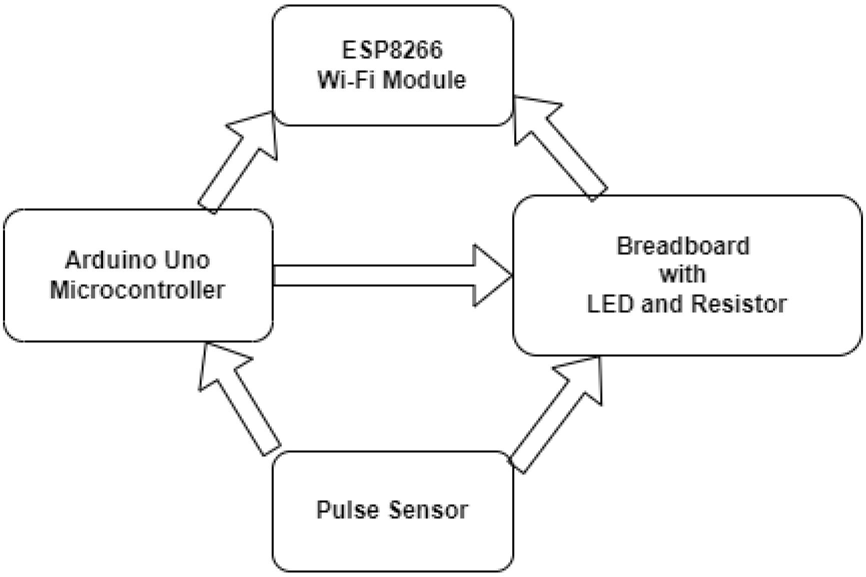

Figure 22: Block Diagram

9.3 Individual Contribution Plan

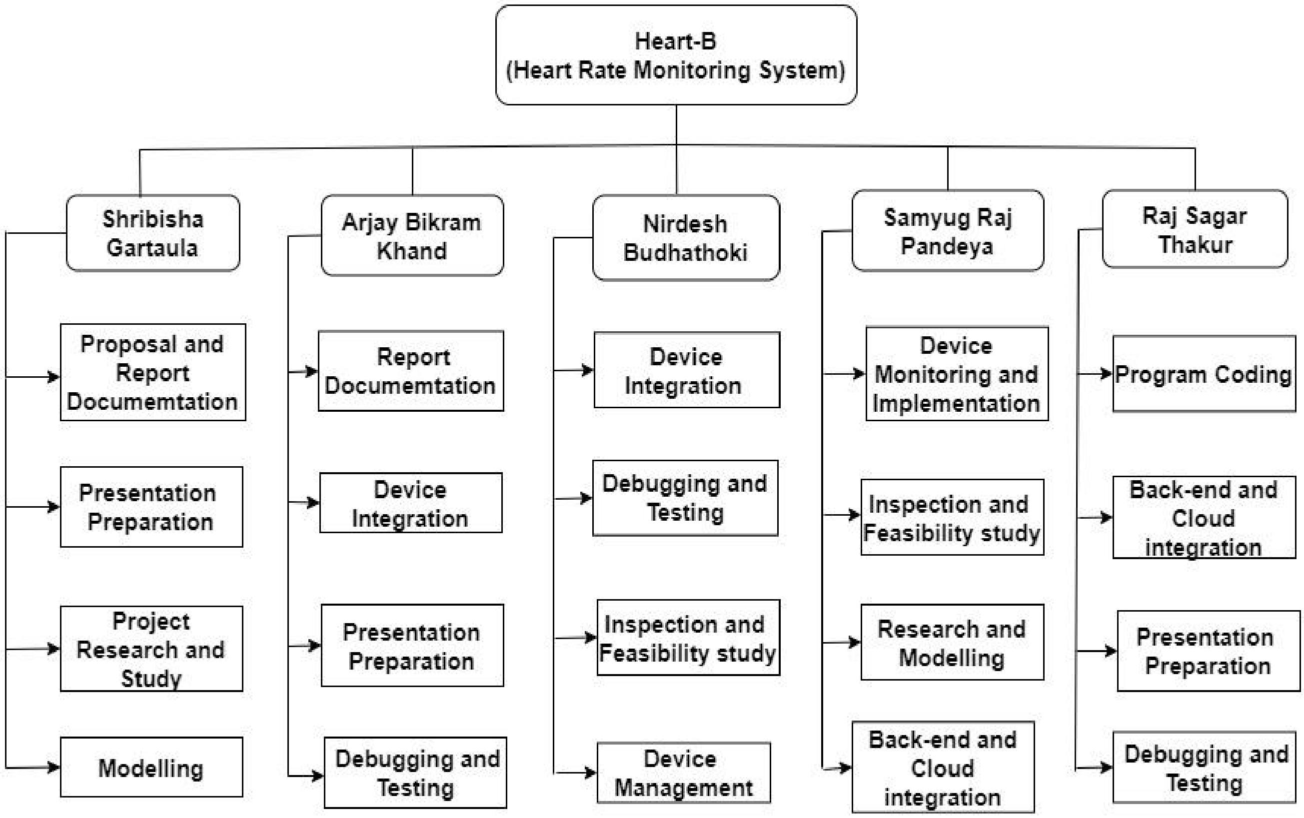

Figure 23: Individual contribution plan

We five members of Team 5 distributed different tasks to every member of our team. We all contributed 20% each from our side to make this project, cooperated, and made 100% effort for the project, from our side.