   
**Technological University Dublin**   
**Blanchardstown Campus**

Bachelor of Engineering in Computer Engineering

H3024 Project 2

***Project Title:*** *GP Virtual System*

***Author:*** *Muhammad Memon*

***Supervisor:*** *Mark Deegan*

***Date:*** *02/05/2024*

# 

# Declaration

The material contained in this assignment is the author’s original work, except where work quoted is duly acknowledged in the text. No aspect of this assignment has been previously submitted in any other unit or course.

Signed: Date: / / .

# Acknowledgments:

I want to thank all my mentors and Lecturers, Fergus Maughan, Ivan Smyth, Mark Deegan, and Dave Carroll for their advice and assistance in the research and completion of my project. They encouraged me to finish all my tasks successfully and supported my at every turn. I also want to thank all the technicians and students who gave me the tools I needed to finish the project as effectively without their assistance, I could have done it as well.

Furthermore, I would like to thank the University of Blanchardstown for accepting my project in the field of study. I would also want to thank Digi key and JLCPCB for deliver components in time for the project work successfully.

# Table of Contents

Declaration ii

Abstract iii

Acknowledgements iv

Table of Contents v

Abbreviations, Symbols, Acronyms vi

List of Figure Captions vii

List of Tables viii

Executive Summary 2

Chapter 1 Introduction 3

1.1 The Aim 3

1.2 Motivation 3

1.3 Layout of Report 3

Chapter 2 Background Research 4

Chapter 3 Project Design 5

3.1 Theory and reasoning for the choice. 5

3.2 Hardware design 5

3.3 Software Design 5

3.4 Software Flow Diagram 5

Chapter 4 Project Construction and Testing 6

4.1 Construction 6

4.2 Testing Procedure 6

4.3 Software Testing 6

4.4 Fault Tree 6

Chapter 6 Safety and Ethical Considerations of the Project. 7

Chapter 7 Conclusions and Recommendations 8

References 9

Appendix 10

Timetable 11

# Abbreviations, Symbols, Acronyms

Snap Electronic Design Automation (Snap EDA)

Printed Circuit Board (PCB)

VM – Virtual Machine

ISP – Integrated Serial Interface

SPI – Serial Peripheral Interface

PCB – Printed Circuit Board

YAML – Yet Markup Language

ESP82866 – Espresso Systems 82866

GND – Ground

VCC – Voltage at common collector

FTDI – Future Technology Devices International

UART – Universal Asynchronous Receiver Transmitter

12C – Inter-Integrated Circuit

ADC – Analogue to Digital Converter

USB – Universal Serial Bus

TCP/IP – Transmission Communication Protocol/Internet Protocol

SQL – Structured Query Language

PHP - Hypertext Preprocessor

GP – General practitioner

HTML – Hyper Text Markup Language

Node MCU - Node Microcontroller Unit

HTTP: Hypertext Transfer Protocol

FTP: File Transfer Protocol

POP3: Post Office Protocol version 3

SMTP: Simple Mail Transfer Protocol

SNMP: Simple Network Management Protocol

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

IP: Internet Protocol

ICMP: Internet Control Message Protocol

ARP: Address Resolution Protocol

MQTT - Message Queuing Telemetry Transport.

SDA - Serial Data Line

SCL - Serial Clock Line

# 

# List of Figure Captions

# Figure 1 – Block Diagram of the project

# Figure 2 – I2C Communication

# Figure 3 – SPI Breakdown

# Figure 4 – MQTT Breakdown

# Figure 5 – Breadboard, Tasmota and Ardunio connection with the DS18B20

# Figure 6 – Pinouts of the DS18B20 Temperature Sensor

# Figure 7 – Tasmota, Breadboard and Arduino connection with the SEN0203

# Figure 8 – Pinouts of the SEN0203 Heart rate Sensor

# Figure 9 – Adafruit Screen connections

# Figure 10 – Pinouts of the Adafruit screen

# Figure 11 – Working Website using Notepad ++

# Figure 12: Patient database made in SQL programming language

# Figure 13 – Working screen of Tasmota

# Figure 13 – Example of an SSID of tasmota -xxxxx-xxxxx.

# Figure 14 – Available networks Wi-Fi connection for tasmota

# Figure 15: Successful IP address connection to access Tasmota

# Figure 16: Layout of the Xamp

# Figure 17 – Database connection in PHP codes

# Figure 18 – Schematic of the project

# Figure 19 – Board/PCB of the project

# Figure 20 – Software Flow Diagram

# Figure 21 - Hardware Design Fault Tree

# Figure 22 – Software Design Fault Tree

# Figure 23 – TCP/IP Protocol

# Figure 23 – Timetable for Project 2

# List of Tables:

Table 1 – Pinouts of the Microcontroller ESP8266

Table 2 - Registered Patients

Table 3 - Registered Doctors

# Executive Summary

# The "GP Virtual Monitoring System" project is a health monitoring solution that tracks advanced technologies to enable remote tracking of vital signs. The health gadget is made up of sensors such as the SEN0203 Heart rate sensor, and DS18B20 Temperature Sensor, the system empowers individuals to monitor their heart rate, body temperature, and blood pressure from the comfort of their homes, mainly focused on individuals worldwide who cannot travel. Key features include real-time updates accessible through an HTML-based website, virtual appointments and prescriptions with healthcare professionals, Time to time accurate information on patient satisfaction.

# The ESP8266 microcontroller, SQL databases for data analysis and storage, HTML websites on Notepad++, Xampp technology for website local host and SQL database connectivity, Sublime Text for PHP server codes to interface with Xampp local host and SQL databases, and Microsoft Azure cloud services are some of the technologies that the project is built on. Overall, the GP Virtual Monitoring System represents a significant advancement in healthcare technology, offering a user-friendly interface, and reliable data transmission with databases and HTML websites. It promises to enhance continuity of care, promote early detection of health issues, and empower individuals to take control of their health effectively.

**The Block Diagram of the GP Virtual System**A screenshot of a graph

Description automatically generatedFigure 1 – Block Diagram of the project

# Chapter 1 Introduction

## The Aim

The project aims to implement a GP Virtual monitoring system utilizing a health gadget device monitoring with a SEN0203 Heart rate sensor, and DS18B20 Temperature Sensor. This device enables remote monitoring of an individual's health by detecting pulse/heart rate and body temperature. The collected health information is transmitted to a dedicated website using HTTP, generating a comprehensive health report accessible to both the patient and their General Practitioner (GP) or doctor. Additionally, the website also allows patients to schedule appointments and enables doctors or GPs to provide online prescriptions through the HTTP healthcare websites.

## Motivation

The motivation of this project is to reduce the number of premature deaths caused by a shortage of general practitioners, doctors, or even treatment options worldwide. Not everyone who needs medical attention for their disease sees a physician or general practitioner right away. Despite the large number of hospital employees, patients must still wait in lengthy lines to receive a routine check-up.

## Layout of Report

The layout of the project is organized into 6 chapters.

1st Chapter: This chapter provides a summary of the project, including information on its goals, the technologies and software employed, and the reasons for the project's motive for doing a GP Virtual project. and the technological tools employed in the GP Virtual System initiative. The technology and software from the final semester of the third year are covered in Chapter 2.

2nd Chapter: In this chapter, the software and technologies utilized throughout the last semester of the third year are discussed. Tasmota is one of the technologies I utilized last year, and it was intended to evaluate the sensors I also employed.

Third Chapter: This chapter describes the project design and explains why it was chosen. I chose to undertake the project, in part, to lighten the workload for the hospital and enable more patients to receive routine check-ups, which will cut down on travel time to the hospital or clinic and, of course, benefit those patients who are unable to travel. Additionally, it discusses the hardware and software architecture employed in this GP Virtual System Project. A piece of software utilized for this project is a SQL database, which connects to an HTML webpage to create a secure website where doctors and patients can be distributed to who is registered and who is not. The Heart is one of the pieces of hardware that operates with an Analog signal connection and connects to all the signal/data pins, to be utilized in this project.

4th Chapter: This chapter discusses the project's closure, and covers the software, testing process, and conclusion. The project's completion and whether it followed the plan are among the topics covered in the conclusion. The software and testing technique will discuss the tests conducted for this project. Among the tests was a successful verification of the temperature and heart rate sensor's compatibility with the Arduino and Tasmota.

5th Chapter: This chapter discusses the safety measures employed in the project. The security measures utilized in the project include a secured website that requires users to register with MYSQL to access it.

6th Chapter: The final chapter focuses on the project's overall future work which I believe should be implemented in the future to help individuals receive healthcare in the comfort of their homes. It also discusses what actions I could take to improve the project on the project capabilities if more time were given.

# Chapter 2 Background Research

The project will make use of research from both semesters, including some of the software and microcontrollers used in the last semester of the third year. Tasmota is one of the software tools utilized this semester for the GP Virtual System project and it will help with the temperature and heart rate sensors.

In the current semester, Tasmota and Arduino were used to retrieve data from it, which will then be stored in the HTML website, enabling the display of updated results on reports for patients and GPs/Doctors to view.

In the previous semester, two microcontrollers were used, The ESP32S3 is equipped with a dual-core X-Tensa LX7 MCU operating at 240MHz, while the ESP8266 is a WIFI chip featuring a Ten silica Diamond Standard CPU running at either 80 MHz or 160 MHz For the GP Virtual Monitoring System project, the ESP8266 (Node-MCU) will be used to connect all WIFI sensors, such as the DS18B20 temperature sensor, SEN0203 and connect them with Tasmota.

**Technologies used from last year in the this year project aswell:**

**TCP/IP Protocol**

* Defines data transmission standards over computer networks, including the internet.
* It breaks the data into packets, transmitting them to their destination.
* Enabling devices to communicate using a common language.

A blue and red table with text

Description automatically generated

**Application layer:**

* Facilitates networking communication, primarily through HTTP. [Hypertext transfer protocol].
* User applications include email services and messaging layers.
* An HTTP website is created in this project to display all medical reports accessible by doctors and patients.

**Transport layer:**

* Manages error free packet exchange between devices involves data packetization.
* Enables reliable communication between medical devices and the central monitoring system, optimizing patient care and safety.

**Networking layer:**

* Involves data transmission across the network, mainly using IP address.
* Connection of operating to tasmota is based on an IP address and access to medical reports on smartphones or laptops is also based on an IP address.

**Data Link layer or Physical layer:**

* Manages physical data transmission, primarily using ethernet.
* Implements protocols for the reliable transmission of data packets over Ethernet connections, ensuring data integrity.

# Figure 23 – TCP/IP Protocol

**I2C** **Communication**

* Utilizes two communication lines (SDA and SCL) for Adafruit screen connections to allow readings on Tasmota and health system to operate.

A diagram of a computer

Description automatically generated

## Figure 2 – I2C Communication

**Wireless Networking**

* Utilizes radio frequency (RF) connections between nodes and networks like telecommunication networks.
* Allows open-source platforms like tasmota or web servers like node red to operate using a wireless network such as Hotspot network connection or House network connection.

**SPI (Serial Peripheral Interface)**

* Synchronous serial communication interface used for short-distance communication in embedded systems.
* It operates as a master-slave protocol, facilitating straightforward interactions between microcontrollers.
* SPI includes clock and data lines, along with a select line, for efficient communication.

A diagram of a computer program

Description automatically generated with medium confidence

## Figure 3 – SPI Breakdown

**MQTT**

* Tasmota facilitates the transmission of sensor data to the cloud data service.
* MQTT parameters link Azure to Tasmota via muhammad03.westeurop.cloudapp.azure.com, ensuring smooth connectivity.
* Employing MQTT technology, the two websites developed for this project will utilize MQTT to enhance updates and ensure accuracy on the website. A blue circle with arrows pointing to a blue circle

  Description automatically generated

# Figure 4 – MQTT Breakdown

# Chapter 3: Project Design

## 3.1 Theory and reasoning for the choice.

**Reducing Hospital Burden:**

During the COVID-19 pandemic, hospitals had to deal with a wide variety of patients, which made it challenging to deliver timely medical care. Many patients suffered because of the strain on healthcare services, and receiving treatment grew increasingly difficult. By carrying out this project, patients will be able to obtain the medical care they want while remaining comfortable in their own homes, reducing the reduction of hospitals and removing the need for patients to wait in long queues. I know how painful it is because I lost a close family friend to COVID. After all, he didn't receive the treatment he needed.  
**Solutions for Remote Healthcare:**

The goal of the project is to solve the problem of shortage of general practitioners (GPs) and doctors. In this project about remote healthcare, Doctors/GPs may effectively diagnose and treat patients without needing to visit them in person. This provides continuity of treatment, enables doctors to work from home, and helps patients by providing access to current, yearly reports and prescriptions on the website run by HML. Additionally, doctors can work more quickly by using virtual meetings like Zoom.

**Healthcare Service Delivery:**  
The project aims to revolutionize healthcare delivery by enabling patients to receive treatment in the comfort of their own homes. Remote monitoring technologies allow patients to receive medical care without needing to visit hospitals or clinics.

This lowers the risk of illness and eases the burden on healthcare facilities. Additionally, by giving healthcare professionals the ability to effectively manage patient care from a distance, the project hopes to empower them and improve the general efficacy and accessibility of healthcare services. The yearly progress report program also facilitates comprehensive patient tracking, enabling both doctors and patients to evaluate patients' progress and make well-informed decisions for long-term healthcare management. Additionally, the section on internet prescriptions provides an easy way to use the remote health care system.

## 3.2 Hardware design

The hardware design of this project is based on Sensors, Microcontrollers (ESP8266), and a Bluetooth Module. The two WIFI sensors used in this project, are DS18B20(Temperature Sensor) and SEN0203(Heart Rate Sensor) which will show results on an HTML-operated website, real-time sensor updates will be shown on the current and yearly reports section of the GP Virtual Monitoring System Website. All updated values of the temperature and heart rate readings will also show on an Adafruit screen.

### Microcontroller – ESP8266

The ESP8266 will act as an access point facilitating communication between the health system and patients and will also ensure that accurate collection of patient information and provide precise health status updates.

|  |  |  |
| --- | --- | --- |
| Pin Category: | Name: | Description: |
| Power | **Micro-USB, 3.3V, GND, Vin** | **Micro-USB:** Node-MCU can be powered through the USB port  **3.3V:** Regulated 3.3V can be supplied to this pin to power the board  **GND:** Ground pins  **Vin:**External Power Supply |
| Control Pins | **EN, RST** | The pin and the button reset the microcontroller |
| Analog Pin | **A0** | Used to measure analog voltage in the range of 0-3.3V |
| GPIO Pins | **GPIO1 to GPIO16** | Node - MCU has 16 general-purpose input-output pins on its board |
| SPI Pins | **SD1, CMD, SD0, CLK** | Node-MCU has four pins available for SPI communication. |
| UART Pins | **TXD0, RXD0, TXD2, RXD2** | Node-MCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program. |
| I2C Pins |  | Node-MCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C. |

# Table 1 – Pinouts of the Microcontroller ESP8266

### 3.2.3 DS18B20 – Temperature Sensor

The DS18B20, temperature sensor operates from 3.0V to 5.5V and measures temperatures from -55C to +125C or -67F to +257F with a 0.5 accuracy. It operates with three pins (VCC, GND, Data pin) shown in Figure 1, breadboard, Tasmota, and Arduino connection with the DS18B20. The Pinout for the DS18B20 is also shown in Figure 2. The DS18B20 temperature sensor is used in this project as a body temperature so it works as a part of the health gadget where the patient can check their body temperature by holding the probe by the hand, the data will show regularly on the HTML-operated website.

A circuit board with wires and a pen

Description automatically generated A screenshot of a device

Description automatically generated A screenshot of a computer

Description automatically generated

# Figure 5 – Breadboard, Tasmota and Ardunio connection with the DS18B20

A wire with wires and text

Description automatically generated with medium confidence Figure 6 – Pinouts of the DS18B20 Temperature Sensor

### 3.2.2 SEN0203 – Heart rate sensor

The heart rate sensor operates from 3.3V to 5.5V, and the heart rate sensor is a thumb-sized heart rate monitor. It operates with 2 kinds of signal output modes, analog pulse mode and digital square wave mode. The pinout of the SEN0203 is shown in Figure 3, breadboard, Tasmota, and Arduino connection of the SEN0203 are shown in Figure 4. The SEN0203 heart rate sensor is used in this project as a heart rate so it works as a part of the health gadget where the patient can check their heart rate by wrapping the black belt around the wrist, earlobe, or other areas where it has contact with the skin, the data will regularly on the html operated website.

# A screenshot of a computer Description automatically generated A white background with black numbers Description automatically generatedFigure 7 – Tasmota, Breadboard and Arduino connection with the SEN0203

A diagram of a gravity sensor

Description automatically generated

Figure 8 – Pinouts of the SEN0203 Heart rate Sensor

### 3.2.4 SSD1306 – Adafruit screen

Adafruit Screen is a powerful processor, 10 NeoPixels, InfraRed receive and transmit, two buttons, a switch. In this project, it is used to show heart rate sensor reading at random and the temperature reads accurately accordingly to the control of the temperature.

A small black electronic device with a blue screen

Description automatically generatedA small screen with white text

Description automatically generated Figure 9 – Adafruit Screen connections

A diagram of a circuit board

Description automatically generated

Figure 10 – Pinouts of the Adafruit screen

## 3.3 Software Design

The software design of this project is based on Notepad ++, SQL databases, Tasmota, Microsoft Azure, Xammp, PHP servers, and Eagle.

### 3.3.1 Notepad ++

Notepad ++ is a source text editor that is designed to make notes regularly and also allows programming. In this project, Notepad ++ was a tool that was used to create the HTML website (Patients Medical Reports. Html (Reports page), Index.html (login page), GP Virtual System.html (Main page)

A screenshot of a computer

Description automatically generated Figure 11 – Working Website using Notepad ++

### 3.3.2 SQL databases

SQL is a Structured Query language that is used for relational databases with a collection of tables columns etc. It is set to be a declarative programming language.

The use of the SQL database in this project is to make a security-based database that is used to access the website, with two databases called Registered Patients and Registered Doctors. It will be used so that only registered patients can only/open the website and if the patients are not registered, it will show the details entered and say "Please, register yourself ".

## How to create the database

1. Download MYSQL by the link <https://www.mysql.com/downloads/>
2. After downloading, follow the steps and make sure to download MYSQL, MYSQL Workbench, and MYSQL Shell.
3. After the steps are followed and all details are memorized, open MYSQL Workbench
4. Open the server u created or make a new one.
5. Click on Schemas, right click on the white bar to access the option of creating a schema.
6. Click it, create a name, and click the down arrow to view tables, then create a new table and click apply.
7. After creating a table, select the rows and create a table starting off id where it always has a Primary Key and not null.
8. Click apply.

A table with numbers and names

Description automatically generated A screenshot of a computer

Description automatically generated Table 2 - Registered Patients Table 3 - Registered Doctors

A close-up of a computer screen

Description automatically generated

# Figure 12: Patient database made in SQL programming language

### 3.3.3 Tasmota

Tasmota is an open-source platform for ESP32-S3 and Node MCU-based devices which allows smart devices to communicate to wireless network protocol called MQTT. In this project, a cloud service called Azure is where the MQTT port was created in the networking of Azure Cloud Service. The Tasmota was successfully connected to Node MCU, Temperature (DS18B20), and Heart rate (SEN0203) sensors were used in Tasmota which can be seen in Figure 5 &7 . The pin out of the DS18B20 can be seen in Figure 6 and the SEN0203 Heart rate sensor is used in Figure 8

A screenshot of a computer

Description automatically generated

# Figure 13 – Working screen of Tasmota

Steps to config Node MCU to Tasmota:

1. Download the Esptool and tasmota.bin file
2. Open Command Prompt, find the directory or cmd the directory from the folder where everything is downloaded.
3. If used, erase the chip using the command “esptool.exe --port <COM port> erase\_flash”
4. After, upload the firmware using the command “esptool.exe --port COM5 write\_flash -fs 1MB -fm dout 0x0 tasmota.bin”.
5. Open network settings, a tasmota WIFI access point with an SSID of tasmota -xxxxx-xxxxx.

A screenshot of a phone

Description automatically generated Figure 13 – Example of an SSID of tasmota -xxxxx-xxxxx.

1. Click the SSID, once the screen of tasmota showing the available networks, seen in Figure 15.

A screenshot of a wifi password

Description automatically generated

# Figure 14 – Available networks Wi-Fi connection for tasmota

1. Wait for it, once an IP address shows, note it down as it will be the IP address to use tasmota each time with Node MCU(ESP8266). IP address used for tasmota: <http://192.168.43.21>0.

A screenshot of a computer

Description automatically generated Figure 15: Successful IP address connection to access Tasmota

1. Config the Sensors, the sensors I had to configure were the DS18B20 at D4 to set to be DS18 x 20 and to configure the SEN0203 at A0, which was set to be an ADC Input. The successful connection of both sensors operating on Tasmota is shown in Figure 4 & 2.
2. Click save and the sensors pops on the screen.

### 3.3.4 Microsoft Azure

Azure Cloud Services is a platform as a Service (PaaS) technology that allows to deployment of stable applications and is reliable to operate. It is designed to support an application to run a virtual machine (VM). This type of cloud service focuses on the application, not the underlying of the cloud infrastructure. It can be used for Microsoft's public cloud computing platform which provides a range of cloud services including storage and networking.

### 3.3.4 Xammp

XAMPP is an open-source collection of online solutions that comes with modules for MariaDB, PHP, Perl12, and Apache servers in addition to the Apache distribution for servers. It helps you create and test web applications before they are deployed by enabling you to set up a local web server environment for testing and development.

In this project, it is used to connect the MYSQL server database to the local host.

With the help of Xampp, the project was able to use a local host to access the website and connect to the MYSQL server.

-> Link for website on the web server of a laptop

localhost/project/index.html

-> Link for website on the web server of a smartphone

192.168.43.233/project

A screenshot of a computer

Description automatically generated Figure 16: Layout of the Xamp

# 3.3.5 PHP servers

### A PHP server is a web server that provides a platform for running PHP applications. It is used in receiving HTTP requests from clients, interpreting them, and returning responses.

### The use PHP servers in this project will be used in two ways for the GP Virtual System Project, The first use of the PHP server in this project is connecting the SQL Database to the HTTP-operated website accessed from the, it will be operated in such a way that each time, a user enters the details for login, each time, the database will check it, if the patient is registered, then the next page will be opened The connection between the database and the PHP server is displayed in the code snippet below; it is utilised to determine $con = new mysqli("localhost(Always the same)", "username", "Password", "Name of the table in the database");

A black and white text

Description automatically generatedFigure 17 – Database connection in PHP codes

### 3.3.6 Eagle

Eagle is an electronic design automation (EDA) that makes the Printed Circuit Board (PCB) that connects schematic diagrams routing and comprehensive library content.

A computer screen shot of a computer

Description automatically generated A computer screen shot of a circuit board

Description automatically generated Figure 18 – Schematic of the project Figure 19 – Board/PCB of the project

3.3.7 Snap EDA

SnapEDA is an electronics design data that provides PCB footprints and schematic symbols.

Libraries downloaded from SnapEDA are \_nodemcu\_esp8266.lbr, DS18B20\_lbr.

## 3.3.8 Netlify

Netlify is the platform of choice for creating dynamic, high-performing websites, and applications. It also increases team productivity by combining a wide range of services, technologies, and APIs into a single process, all while saving time and money.

The website could also be accessed without connecting to the database by using Netlify.   
To visit the Netlify-powered website, use this link: <https://gp-virtual-monitoring-system.netlify.app/>

This link will open a real website that connects to three other websites: Patients Medical Reports (Medical Reports page), GP Virtual System (Main page), and Index.html (Login page).

## Software Flow Diagram

 Figure 20 – Software Flow Diagram

### The software flow diagram for the GP Virtual System project illustrates how the Sensors are connected to Tasmota. Next, the SQL databases are sent to Xmapp, enabling HTML websites to be hosted locally. The index.html website logs in using the local host, and after entering the correct database details like the Name, Password, and Medical ID, it enables the display of the other website, which is displayed in the software flow diagram's websites section.

### 3.5 Alternate Design

**Overview:**

Vital sign tracking from home is made easier with the GP Virtual System simplified remote healthcare monitoring system. The system consists of a health device that has sensors for temperature, and heart rate. For real-time analysis, data was planned to proceed on an HTML-operated website but didn’t go well, make virtual appointments, and manage their medicines with an online prescriptions section in the Patients Medical Reports.html. Secured patient and doctor portals allow patients and providers to communicate while maintaining patient privacy using the SQL Database to allow registered patients and doctors to view and use the website.

**Important characteristics:**   
  
Small Health Device: Wirelessly connects to monitor vital signs such as DS18B20 – temperature sensor and SEN0203 – heart rate sensor.   
Secured Websites: Secured websites for patients and doctors with access to allow registered patients and doctors to view the website only.   
HTML Websites: Enables users to manage prescriptions, make appointments, and check health statistics.   
Secured Patients and Doctors Databases: Protect patient privacy while facilitating communication with the GP Virtual System.

Design Considerations:   
  
Use safe authentication methods and end-to-end encryption for data security.   
Scalability: Create a modular architecture that can grow and integrate in the future.   
User-Centric Design: User's experience first by using hardware that is ergonomic and has clear interfaces.   
Iterative updates and improvements can be made by incorporating user feedback for continuous improvement.

# Chapter 4 Project Construction and Testing

## 4.1 Construction

I began the project by understanding each component that would be used, and then I went on to create the websites using HTML, CSS, and Java. On the website, CSS was used to design the body fonts, text sizes, text layout, etc., and Java was used to create the search bar, which would allow users to look up local general practitioners in their area. In addition to making the websites, I conducted research for the upcoming classes so that I could review them with my supervisor for any additional instructions and once everything was sorted and all research was done, the components were ordered by the supervisor. While on the wait for the components, the proposal report was in the process of being typed up as it was a document for the supervisor to know the software, hardware, components, technologies, etc. used in the project, the weekly reports were also a document of grade that tried to be uploaded on Moodle and GitHub.

After finishing all the reports and websites, it was time for my proposal to be approved and for my components to arrive, so I began working on my software. Initially, I used the software to understand the components and connect them with Tasmota (connections of the two sensors, DS18B20, the temperature sensor, and SEN0203, the heart rate sensor in Figures 5 & 7). I then analysed Node-red to allow to work with Tasmota, but I had to change my mind due to error connections between tasmota and Node-Red so used a testing procedure to see if the two sensors in the project could connect in Tasmota and Arduino. The DS18B20 passed both stages of the test, but the SEN0203 was less accurate in Ardunio but is more accurate in Tasmota.

In addition to creating the Arduino connections, other software was supposed to create the SQL databases, as seen in Tables 1 and 2. The PHP code connection, as seen in Figure 17, enabled the creation of MYSQL and website connections. After the database and connection were created, I had to learn more about how to connect it to the website. After doing so, I learned that an open-source programme called Xampp was required to function; Figure 16 illustrates how it operated. Once the connection and all necessary resources were in place for the website to operate with the SQL database, the website was published on the local host and it was able to move forward with the Registered Patients and Registered doctors databases.

Another research was needed to do which was the hardware part of my project was how to connect the Adafruit screen to esp8266, after doing the research, found a sample code from my lecturer which allowed me to connect it to the esp32s3 but as I had a different chip which is the esp8266, the pin connections, and the code were needed to change around in order it to work with the esp8266 chip. Successful connections of the Adafruit screen are shown in Figure 9.

## 4.2 Testing Procedure

The concept of software and hardware forms the basis of the project's testing process.   
The DS18B20 temperature sensor and the SEN0203 heart rate sensor on Arduino and Tasmota were tested, and the results showed that while the SEN0203 heart rate sensor was somewhat accurate on Arduino if the values were set as random in the range of 0 to 600, the Arduino had accurate results using the DS18B20 temperature sensor and the same on Tasmota. This understanding underpins the hardware used for the testing. Slide 7 of the presentation on GitHub indicates that the heart rate sensor should have displayed an accurate value, but it didn't.

Another test was conducted to verify the connection between the Blood Pressure Monitor SBM70 and the HC05 Bluetooth Module. The HC05 should have worked with an LED, but due to Serial Bluetooth Terminal App connection with Arduino errors, it was not successfully connected to the HC05 Bluetooth. Further research was required after ordering components, as the SBM70 blood pressure monitor was blocking all other Bluetooth connections with the SBM70. As a result, blood pressure was not included in the GP Virtual System project.

Another test involved connecting PHP servers to Tasmota to update sensor values. This had a significant detrimental effect on the website because it was essential for real-time sensor updates to function on the HTML website. Due to a lack of knowledge of PHP, the PHP code that was supposed to connect the website with real-time updates was not completed for the project. However, because I was knowledgeable about databases, I was able to connect them to PHP successfully.

## 4.4 Fault Tree

## 4.4.1 Hardware Design Fault Tree

A diagram of a software system

Description automatically generated Figure 21 - Hardware Design Fault Tree

## 4.4.2 Hardware Design Fault Tree

A diagram of a computer system

Description automatically generated

Figure 22 – Software Design Fault Tree

# Chapter 5: Safety and Ethical Considerations of the Project.

Health and safety issues concerns are based on both software and hardware.

**Hardware issues:**

It relies on sensors like the DS18B20 temperature sensor and the SEN0203 heart rate sensor, which guarantee measurement accuracy and without meeting a risk to users.

For instance, making sure the sensors are calibrated to the extent of injury and discomfort for people wearing them. For instance, the SEN0203 is a wristband measurement that measures the heart rate around the wrist, fingers, or earlobe. This measurement can raise the risk of an incorrect connection, which could result in the heart rate sensor becoming hot and endangering patients.

**Software issues:**

The issues are based on software components like the firmware of the ESP8266 microcontroller, SQL databases for data analysis and storage, HTML websites created with Notepad++, Xampp technology for connecting websites to local hosts and SQL databases, and Sublime Text for PHP server codes that interface with Xampp local host and SQL databases that prioritize data security and privacy of the website. Patient data is protected from unauthorized access by secure authentication, encryption of the transferred data, and routine software updates. Another problem with the health and safety software is that if the real-time updates are inaccurate for any reason, the doctor and GP may think that the report's readings are accurate when, in fact, they may not be. In this case, the GP or doctor may suggest medication based on inaccurate health information, which could lead to a high risk to the patient's health. It also ensures the design of the PCB for the health gadget is a user-friendly and intuitive for the patients to analyse and used for the purpose it is made for.

**Electrical issues:**

The issues of electrical health and safety are based on electronic components and microcontrollers to prevent electrical risks including short circuits, electric shocks, and fires because the system includes electronic components and microcontrollers. To guarantee user safety, appropriate insulation, grounding, and ensuring all electrical safety laws are followed.

## Potential Ethical Issues Related to Data Privacy, Security, or Environmental Impact:

The GP Virtual Monitoring System project acknowledges and tackles several important concerns about the security and privacy of private health information. It emphasizes the necessity to protect patient confidentiality while acknowledging the inherent privacy problems related to the gathering and storage of personal health information. The project uses a PHP server to establish strong encryption and secure login procedures on the health report website to reduce the danger of data breaches or unauthorized access. By taking these precautions, we want to keep patient data safe and out of the hands of unauthorized people. In addition, the project respects ethical principles by not exploiting patient data for commercial or scientific reasons without express authorization.

# 

# Chapter 6 Conclusions and Recommendations

In summary, the GP Virtual System project had both successes and failures.   
  
The project was successful for several reasons:

Some of its components were finished, including creating a secure HTML website, making the website available to patients and doctors/GPs at any time via the link at 192.168.43.233/Project, successfully connecting PHP to a SQL database, connecting Registered Patients and Doctors databases, successfully connecting Tasmota and Arduino sensors, creating a PCB from JLC PCB, and successfully creating the report and presentation.

The project was not successful for a several of reasons that prevented it from being a respectable effort:

The PCB was delivered on schedule, but it wasn't soldered due to lack of time management. In addition, there were problems with the soldering iron being entirely burned, which meant that it shouldn't be used to solder the PCB's components, Configuring HTML websites with real-time sensor value updates so that physicians and patients can regularly access the patient's health information and the patients regularly stored yearly health data, it was planned to pair the HC05 with a blood pressure unit via Bluetooth. However, by the time the project was finished, there were issues with the code and the Bluetooth connection between the two modules that were used, which prevented the HC05 from connecting to the blood pressure monitor. Although setting up Node Red was another idea for this project, I was unable to use it to connect the real-time updates to the website because of a problem with the connection to Tasmota. The MQTT connection was established on Tasmota, and Node Red established a successful connection with Tasmota, although nothing appeared to be working. The idea was to read the values from Tasmota to Node-Red, which would then send them to MY SQL database, which would read them from PHP servers. Depending on how quickly everything worked, Node Red would then update the values on the website on a regular basis.

The following were some of the major issues that occurred during the project but were resolved on schedule:

• There were issues with the laptop's serial connection to the ESP8266 chip, which led to an error message saying "USB not recognised" on the laptop each time the ESP8266 chip was attached. I had to reset the chip by hitting the boot button in order to fix this problem. After that, I set up Tasmota and all of the MQTT connections on Tasmota.

• The schedule for placing component orders and designing a PCB presented another difficulty for me throughout the project. The failure to locate certain parts, such as the blood pressure unit that would connect to the ESP8266 and Arduino.

• Managing time during the semester and missing lessons due to holidays presented additional difficulties that delayed the project's progress. This problem became worse by the semester's brief length of time . In order to solve this issue, I made extra time for myself at home and at college in order to produce higher-quality work without reducing burden.

• Swapping out the microcontroller halfway through the semester created another major challenge, as the codes for Bluetooth and Wi-Fi connections on Arduino varied, and this had a big impact on the connections between the two networks. In order to get beyond this challenge, I carried out more investigation and ultimately discovered that all connections were controllable and just needed small tweaks to guarantee proper operation.

Future Recommendations:

As previously stated, there were issues and reasons why the project did not succeed. However, in my opinion, if the project's timeline had been extended a little bit, it would have made it possible to complete a better project and move forward with a better project at this time.   
  
The items that would have changed are that I would have designed the PHP code to handle real-time updates on the websites and solved any software issues in Node Red to enable Tasmota to communicate values to Node-red so that it could send to the SQL Database. Additionally, a PCB will be used for the project rather just a breadboard. Additionally, a fully functional medical system with a functioning USB or Bluetooth connection will be included in the project.

# References

[1] “Heart\_Rate\_Sensor\_SKU\_\_SEN0203-DFRobot,” *Dfrobot.com*, 2016. <https://wiki.dfrobot.com/Heart_Rate_Sensor_SKU__SEN0203>

[2]“MOODLE - TU Dublin Blanchardstown Campus: Log in to the site,” vle-bn.tudublin.ie. <https://vle-bn.tudublin.ie/login/index.php>

[3]“GP Patient Monitoring System,” Netlify.app, 2024. https://gp-virtual-monitoting-system.netlify.app/gp%20virtual%20system (accessed May 03, 2024)

[4]Components101, “NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet,” *components101.com*, Apr. 22, 2020. <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>

[5]“MySQL :: MySQL Downloads,” *Mysql.com*, 2019. <https://www.mysql.com/downloads/>

[6]“Download XAMPP,” *www.apachefriends.org*. <https://www.apachefriends.org/download.html>

[7]MMemon2003, “HealthProject2024/Hardware/SEN0203(Heart Rate) .md at main · MMemon2003/HealthProject2024,” *GitHub*, 2024. https://github.com/MMemon2003/HealthProject2024/blob/main/Hardware/SEN0203(Heart%20Rate)%20.md (accessed May 03, 2024).

[8]“SnapEDA,” *Snapeda.com*, 2010. <https://www.snapeda.com/>

[9]“Autodesk Eagle,” eagle.autodesk.com. http://eagle.autodesk.com/eagle/software-versions/50 (accessed May 03, 2024).

[10]“Microsoft Azure,” portal.azure.com. <https://portal.azure.com/#home>

[11]“ESP8266 0.96 inch OLED Display with Arduino IDE | Random Nerd Tutorials,” May 23, 2019. <https://randomnerdtutorials.com/esp8266-0-96-inch-oled-display-with-arduino-ide/>

[12]“Sublime Text,” www.sublimetext.com. <https://www.sublimetext.com/download_thanks?target=win-x64>

[14]“Netlify Create,” create.netlify.com. https://create.netlify.com/team/662d9527f44daf00bdc068a0 (accessed May 03, 2024).

[15]“Diagram Software and Flowchart Maker,” *www.drawio.com*. https://www.drawio.com/

[16]“Google,” *www.google.ie*. <https://www.google.ie/>

[17]M. Lab, “DS18B20 with ESP8266 NodeMCU (Single/Multiple): Display Readings on OLED,” *Microcontrollers Lab*, Jul. 24, 2021. https://microcontrollerslab.com/ds18b20-esp8266-nodemcu-single-multiple-display-readings-oled/ (accessed May 03, 2024).

[18]E. Fahad, “ESP8266 Bluetooth: HC05 Bluetooth interfacing with Nodemcu ESP8266,” *Electronic Clinic*, Nov. 01, 2020. https://www.electroniclinic.com/esp8266-bluetooth-hc05-bluetooth-interfacing-with-nodemcu-esp8266/ (accessed May 03, 2024).

[19]“HC-05 Bluetooth module,” 2010. Available: <https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf>

[20]“DS18B20 Temperature Sensor Pinout, Specifications, Equivalents & Datasheet,” *Components101.com*, 2018. <https://components101.com/sensors/ds18b20-temperature-sensor>

[21]

“ESP8266EX Datasheet.” Available: <https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>

[22]“DigiKey Electronics Home,” *Digi-Key Electronics*. <https://www.digikey.ie/>

[23]“RS | Electronic and Electrical Components,” ie.rs-online.com. https://ie.rs-online.com/web/

‌

‌

‌

‌

## Timetable:

A graph of blue and orange bars

Description automatically generated

# Figure 23 – Timetable for Project 2