**Project Report**

*on*

**“Smart Health Monitoring System Using Pules and Temperature Sensor”**

**Submitted by**

**Soham Kokate**

# Karan Pagar

# Pranav Chaudhari

# Ram Gitte

**Under the guidance of**

**Prof. Mrs. Mrunal P. Fatangare**

***In partial fulfilment of***

# Diploma in Information Technology

[2022-2023]

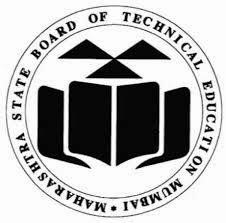
At



DEPARTMENT OF INFORMATION TECHNOLOGY

# MIT POLYTECHNIC

**PUNE-411038**

Affiliated to

MSBTE

## Acknowledgement

I take immense pleasure in thanking **Dr. R.S kale**, our principle for having permitted me to carry out Smart Health Monitoring project work.

I would like to thank my Head of the IT Department **Prof. M. P. Fatangare** for her constructive criticism throughout my project.

I would like to thank Department project coordinator **Prof. Mrs. Pallavi Nehete** for their support and advices to get and complete my project.

I wish to express my deep sense of gratitude to our Internal Guide, **Prof. Mrs. Mrunal Fatangare** , Lecturer, and Department of Information Technology for their valuable guidance and useful suggestions, which helped me in completing of the project work, in time.

Finally, yet important, I would like to express my heartfelt thanks to my beloved parents for their blessings and my friends for their help and wishes for successful completion of this project

## ABSTRACT

for those who require ongoing monitoring that cannot be offered outside of hospitals, smart healthcare is crucial. Additionally, it is crucial in rural areas where adjacent clinics may communicate with metropolitan hospitals on the health of their patients. This study proposes a smart health monitoring system that notifies the concerned party through SMS while monitoring the patient's state using biomedical sensors. Here, the biomedical sensors are interfaced to a Wi-Fi module controller, which reads the data, and outputs it to an LCD display or serial monitor. For a smartphone to show data, SMS must be used. a message sent by the patient's family members to make it simple to access the patient's information. Monitoring the patient at home and regularly checking on their health is a particularly difficult chore in the routinely busy work. It is important to periodically check on older people's health in order to prevent serious medical emergencies.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | **CONTENTS** | **PAGE NO.** |
| **1** | **INTRODUCTION**   * 1. Motivation   2. Background   3. Need Of the Project | **7** |
| **2** | **LITERATURE SURVEY** | **8-11** |
| **3** | **SCOPE OF PROJECT**  3.1 Project objectives  3.2 Project Features  3.3 Project Cost (Overall Cost) | **12-13** |
| **4** | **PROPOSED METHODOLOGY**  4.1 System Architecture  4.2 Introduction  4.2.1 Working  4.2.2 Description of certain components | **15-19** |
| **5** | **DETAILS DESIGN AND WORKING PROCESSES**  5.1 Hardware Software Requirement  5.2 DFD’s (0, 1, ) level  5.3 Use Case Diagrams  5.4 ER Diagram  5**.**5 Timeline Chart  5.6 Implementation  5.6.1 Sample Code  5.6.2 Types of testing  5.6.3 Test case | **20-52** |
| **6** | **RESULTS AND APPLICATION**  6.1 Screen Shot  6.2 Results  6.3 Application | **53-59** |
| **7** | **Conclusion and Future Scope**……………………………………….. | **60** |
| **8** | **References and Bibliography**………………………………………... | **61** |
|  | **Paper Published**……………………………………………………… | **62-68** |

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PAGE NO.** |
| 1. | Cost Estimation Table | **13** |
| 2. | Hardware Requirement | **20** |
| 3. | Software Requirement. | **20** |
| 5.6.3.1 | Testing of User Side | **52** |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1. | System Architecture of Software | **14** |
| 2. | System Architecture of Hardware | **14** |
| 3. | Wi-Fi Module | **17** |
| 4. | Temperature sensor | **18** |
| 5. | Oxygen Sensor | **18** |
| 6. | LCD display | **19** |
| 7. | DFD Level 0 | **21** |
| 8. | DFD Level 1 | **21** |
| 9. | Use case diagram | **22** |
| 10. | ER Diagram | **23** |
| 11. | Timeline Chart | **24** |
| 6.1 | Smart Health Monitoring System | **53** |
| 6.2 | System in Ready state to work | **53** |
| 6.3 | Patient is checking Blood Pressure and Oxygen Level | **54** |
| 6.4 | Patient is Measuring the Temperature of Body | **54** |
| 6.5 | All Sensors are working and displaying the Readings | **55** |
| 6.5.1 | Graph and Readings of Blood Pressure and Oxygen Level | **56** |
| 6.6 | Pulse rate and Temperature reading | **56** |
| 6.7 | The web application use to store the Data | **57** |
| 6.7.1 | Graph of Temperature Sensor | **57** |
| 6.7.2 | Data is Stored in Excel Sheet | **58** |

---------------------------------------------------------------------------------------------------------------------------

**CHAPTER 1 INTRODUCTION**

**1.1 MOTIVATION**

Health monitoring is the major problem in today’s world. Due to lack of proper health monitoring, patient suffer from serious health issues. Without monitoring our health, it is not possible to state whether you are a healthy or sick person. This problem leads.

**1.2 BACKGROUND**

Before the all-in-one smart health care monitoring system made hospital where the only source of checking the pulse, temperature and the oxygen in blood which used three different devices. So the patient need to visit hospital regularly. This is why the All in one Smart Health Care Monitoring system is made for the patient’s convince with using SMS system.

**1.3 NEED OF SMART TROLLEY**

With the help of Smart Health Care Monitoring System, we can skip the process of going to hospital or a doctor for the routine checkups, and we can do this checkup at home. By giving it a SMS system, we can send the data to patients mobile directly, so they can be cautious about their health.

**CHAPTER 2 LITERATURE SURVEY**

### 2.1 Mrunal Fatangare and Hemlata Ohal “An Empirical Survey on Early Health Condition Prediction based on Clinical Parameters” Journal of Engineering and Technology.

This paper discusses the development of a crossover parametric model for the cross-analysis of diseases such as cancer, cardiovascular disease, and diabetes. The model aims to improve the precision with which these diseases are recognized by analysing the influence of fluctuations in bodily parameters. The paper compares several machine learning and deep learning algorithms for disease detection using cross-body parameters. Results show that the system increases the accuracy of disease detection compared to non-cross parametric techniques. The paper examines several related studies and discusses methods, styles, actions, and procedures used in developing the crossover parametric model.

**2.2 M. Fatangare, P. U. Nehete, L. R. Mahajan, A. J. Nair and S. Madhurkar, “Identification And Expenditure Tracking Using RFID Based Intelligent Card,” 2020 4thInternational Conference on trends in Electronics and Informatics (ICOEI)(48184), 2020,pp. 573-579,doi:10.1109/ICOEI48184.2020.9143033**

### The abstract describes a system that uses RFID technology to track student attendance and financial transactions at schools and universities. Each student is provided with a card containing an RFID chip that serves as an ID, security, and payment method. The system aims to simplify the attendance-taking process and reduce errors that may occur with manual methods. It also captures classroom data for the distribution of appropriate attendance scores, which can be used for administrative decisions. Additionally, the system uses the RFID tag as a key for the student's authorized bank account to manage account-related transactions. The abstract highlights the versatility of RFID technology in various industries, such as transportation, medical services, farming, and hospitality

### 2.3 M. Fatangare, A. Nimbalkar, G. Chite, A. Narkhede and A. Khilnani, “An Efficient Temperature Monitoring using Raspberry Pi,” 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 1-5, doi:10.1109/ICICT48043.2020.9112376

**CHAPTER 2 LITERATURE SURVEY**

The paper discusses an IoT-based smart health monitoring system that has the capability to monitor the health status of patients remotely and provide real-time feedback to healthcare providers. This system comprises various IoT devices, such as wearables, sensors, and mobile applications, which collect and transmit the health data of patients to a centralized platform. The platform analyzes this data using various machine learning algorithms and provides insights and recommendations to healthcare providers for timely intervention.

**2.4 IoT BASED SMART HEALTH MONITORING SYSTEM, U. B. Mahadevaswamy, Durgad Zaibabuktiyar**

The proposed system is a Temperature Monitoring system that uses various components such as a Temperature Sensor, Jumper Wires, Programmable Logic Controller, and Analog Card & Raspberry Pie 3 to track temperature in various places. One of the significant improvements over the old system is that the data is stored in a Database & can be accessed from all over the World. The system is divided primarily into two sections: equipment and programming. Raspberry pi takes data and saves it in the MySQL database, which is connected to a user-friendly Webpage that displays the extracted data. The system can be used in industrial furnaces with equal efficiency, making it suitable for various temperature monitoring applications

**2.5 IoT-Based Smart Health Unit Syed Muhammad Waqas Shah, Maruf Pasha 01 Jan 2017-Journal of Software-Vol.12**

The paper focuses on addressing the healthcare challenges faced by remote and underdeveloped areas, where access to healthcare services is limited due to the lack of proper medical infrastructure. The proposed solution is an IoT-based smart health monitoring system that can provide basic health services and access to medical staff remotely. The proposed system consists of three layers: the sensor layer, the network access layer, and the service layer. The sensor layer includes various sensors, such as temperature, blood pressure, and heart rate sensors, that collect patient health data. The network access layer provides connectivity between the sensors and the service layer, allowing for efficient communication. The service layer includes medical professionals who analyze the data and provide medical advice and treatment remotely.

**2.6 IoT BASED CARE SYSTEM BASED ON IoT Jeon Seong Ho**

**CHAPTER 2 LITERATURE SURVEY**

Overall, the research paper highlights the potential of IoT-based healthcare systems to transform the way healthcare is delivered and managed. With the increasing prevalence of connected devices and the growing demand for more personalized and efficient healthcare services, it is likely that IoT-based smart care systems will play an increasingly important role in the future of healthcare.

**2.7 Development of Smart Health Monitoring System D. S. Dayanal G. Kalpana2 D. Godwin Immanuel**

The proposed Android application is designed to be user-friendly and easily accessible to people of all ages and fitness levels. It can be downloaded from the Google Play Store and used on any Android device, making it convenient for users to access their fitness information and track their progress on the go. In addition to providing essential features such as BMI and BMR calculators, workout plans, and meal plans, the application also has a calorie counter that can help users monitor their calorie intake and ensure they are staying within their daily limits.

**2.8 A Smart Healthcare Monitoring System Using Smartphone Interface Aksht1, Gaurav1 , Zahid1, Bhupendra1, Aditi1, Sachin Kumar1, Maneesha1, P. K. Pandey**

The proposed smart healthcare monitoring framework using sensors and the IoT in the paper is designed to address the challenges faced by the healthcare sector in rural areas of India. The shortage of healthcare personnel and infrastructure in these areas makes it difficult for patients to access quality healthcare services. The proposed system aims to bridge this gap by enabling remote monitoring of medical parameters and tracking of medical equipment, leading to the provision of smart hospital services. The system consists of a network of sensors and devices that are deployed in rural hospitals and clinics. The sensors are used to monitor the vital signs of patients, such as blood pressure, heart rate, and temperature. The data collected by the sensors is transmitted to a central system using the IoT, where it is analyzed by machine learning algorithms to detect patterns and anomalies in the patient's health status.

**2.9 Android based health care monitoring system Maradugu Anil Kumar1, Y. Ravi Sekhar1**

**CHAPTER 2 LITERATURE SURVEY**

The paper suggests the use of a system for continuous monitoring of a patient's health parameters, which includes heart rate, oxygen saturation level, and temperature. The system utilizes a web server and an Android application, enabling doctors to remotely monitor the patient's condition. The application allows doctors to access the patient's medical history stored on the server, providing a comprehensive overview of their health. This system aims to improve patient care by allowing for timely intervention and management of potential health risks. The paper further discusses the technical details of the system, including the hardware and software components, as well as the security measures implemented to protect patient data.

**2.10 A health remote monitoring application based on wireless body area network Elhoussaine Baba1, Abdelillah Jibab1, Ahmed Hammouch1 01 Apr 2018**

The development of a wearable wireless body area network (WBAN) application for remote health monitoring is a significant advancement in healthcare technology. The application is designed to monitor physiological signals and provide real-time transmission of data. The application employs four biomedical sensor nodes that collect data such as ECG, body temperature, blood pressure, and pulse rate. The sensors are placed on the body and are connected to a central node that wirelessly transmits the data to a monitoring PC. The data collected by the sensors can be analyzed by healthcare professionals to track patient health and detect any abnormalities. The use of wearable WBAN applications has great potential in remote health monitoring and can significantly improve patient outcomes by allowing for continuous and real-time monitoring of vital signs.

**3.1 Project objectives:**

**CHAPTER 3 SCOPE OF THE PROJECT**

The primary objectives of Smart Health Monitoring are to:

1. Monitor and track the health status of patients using advanced technologies to improve medical outcomes.
2. Provide real-time alerts and notifications to healthcare providers when potential health problems are detected, allowing for timely intervention and treatment.
3. Improve patient engagement and participation in their own health by providing access to personalized health data and actionable insights.
4. Enable remote monitoring and telemedicine services to expand access to quality care and reduce healthcare costs.
5. Enhance the efficiency of healthcare delivery by automating routine tasks and streamlining data collection and analysis.
6. Ensure data security and privacy to protect sensitive personal health information.

Overall, the aim of Smart Health Monitoring is to improve the quality of care for patients, increase efficiency in healthcare delivery, and ultimately improve health outcomes while reducing costs.

**3.2 Project features:**

Smart Health Monitoring contains following features:

1. Each Smart Trolley is fitted with security tracking device to stop people taking it out of the premises.
2. A mobile application that allows users to view their health data in real-time, set goals, track progress and receive alerts related to their health.
3. All the collected data is stored securely in the cloud, providing easy access to healthcare professionals for better analysis and diagnosis.
4. Advanced machine learning algorithms can be used to analyze the collected data and provide insights on patterns and trends, which can help doctors make better diagnoses.

**3.3 Project Cost**

**CHAPTER 3 SCOPE OF THE PROJECT**

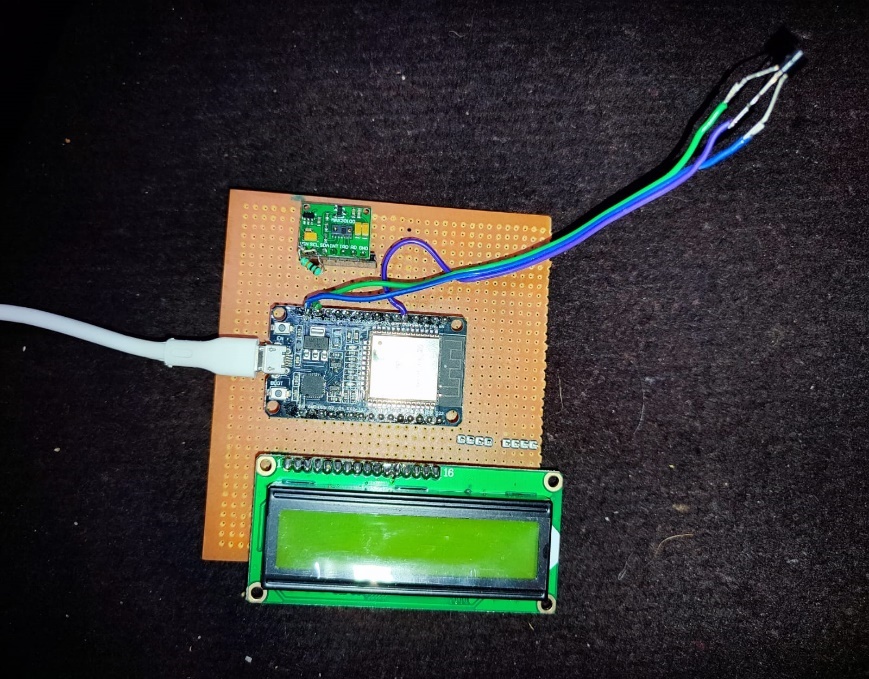
|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Components** | **Quantity** | **Price** |
| **01** | **Oxygen Sensor** | **1** | **250 Rs** |
| **02** | **Temperature sensor** | **1** | **100 Rs** |
| **03** | **Wi-Fi Module** | **1** | **350 Rs** |
| **04** | **LCD** | **1** | **350 Rs** |
| **05** | **Wi-Fi Module code** | **1** | **2000 Rs** |
| **06** | **Circuit Board** | **1** | **100 Rs** |
| **07** | **Labour per Hour(100rs)** | **50 hours** | **5000 Rs** |

**Table 1: Cost Estimation Table**

### 4.1 System Architecture

**CHAPTER 4 PROPOSED METHODOLOGY**

**Figure** **1: System Architecture of Software**



**Figure** **2: System Architecture of Hardware**

**CHAPTER 4 PROPOSED METHODOLOGY**

**4.2 Introduction:**

The development of sensor networks, particularly in the last years, has extended their applicability in various domains, such as heritage preservation, environmental motoring and human activity recognition. Sensor based systems are known as being highly application dependent. This also includes the top layer, which should handle the data in an efficient and useful manner. Furthermore, the size of collected data is rapidly increasing with the number and scale of deployed sensor networks and specialized methods able to deal with such scale and still satisfy application requirements are needed.

we show how we can apply machine learning (ML) algorithms on automatically gathered sensor data combined with manually collected data in order to predict different events. Our demonstration is based on the data collected from a sensor node deployed in our lab, which measured temperature, humidity, light and pressure over 15 days. These parameters are affected by human presence. In parallel, we manually collected data related to human presence and events in the lab. These two sets of data are then aligned and used for training ML algorithms which are then able to predict the number of people in the lab.

This work is focused on ML for analysis of sensor data as a part of a complete vertical system integration, spanning from hardware at the bottom level to data-driven ML algorithms at the topmost. To the best of our knowledge, this is the first sensor system with such a deep vertical integration. Moreover, we consider our system as an example of applying machine learning methods on sensor data, which can provide high-level guidelines for similar applications involving prediction from sensor data.

#### 4.2.1 Working

IOT patient monitoring has 3 sensors. They are Temperature sensor, Heartbeat sensor, Oxygen sensor. This project is very useful since the doctor can monitor patient health parameters just by visiting website or URL. And nowadays many IOT apps are also being developed. So now the doctor or family members can monitor or track the patient health through the Android application. To operate IOT based health monitoring system project, you need a Wi-Fi connection.

With the help of Smart Health Care Monitoring System, we can skip the process of going to hospital or a doctor for the routine checkups , and we can do this checkup at home. By giving it a SMS system, we can send the data to patients mobile directly, so they can be cautious about their health. Health monitoring is the major problem in today’s world. Due to lack of proper health monitoring, patient suffer from serious health issues. Without monitoring our health, it is not possible to state whether you are a healthy or sick person. This problem leads.

## 4.2.2Description of certain components

**CHAPTER 4 PROPOSED METHODOLOGY**

Smart Health Monitoring systems typically consist of several components that work together to monitor a patient's health and provide relevant data to healthcare professionals. Here are some common components:

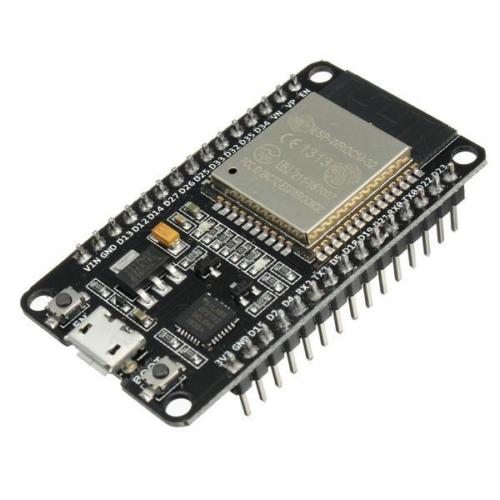
1. Wearable devices: These are small electronic devices that can be worn on the body, such as smartwatches or fitness trackers. They can track various health metrics like heart rate, blood pressure, and sleep patterns.
2. Sensors: These are devices that can detect physical changes in the environment or the body. For example, a sensor could track a patient's movement, temperature, or glucose levels.
3. Connectivity: Smart Health Monitoring systems use wireless connectivity to send data from wearable devices and sensors to a central repository, such as a cloud-based platform.
4. Data analytics: The data collected by the wearable devices and sensors is analyzed using machine learning algorithms to identify patterns and trends in the patient's health data. This information can be used to help healthcare professionals make more informed decisions about the patient's care.
5. Mobile applications: Many Smart Health Monitoring systems have mobile applications that can be used by patients to view their health data and receive alerts about any concerning changes. These apps may also provide resources for managing chronic conditions or accessing medical advice.
6. Cloud-based platforms: The data collected by the wearable devices and sensors is often stored on secure cloud-based platforms. These platforms allow multiple healthcare professionals to access the same data, which can lead to better collaboration and more coordinated care.

* **Wi-Fi Module**: The ESP32 is a low-cost, low-power system-on-chip (SoC) with integrated Wi-Fi and Bluetooth capabilities. It is widely used in Internet of Things (IoT) applications, such as smart home devices, wearables, and industrial automation.

**CHAPTER 4 PROPOSED METHODOLOGY**

* Here are some key features of the ESP32:

1. Dual-core 32-bit CPU running at up to 240 MHz
2. Wi-Fi 802.11 b/g/n/e/i with support for WPA/WPA2 and WEP encryption
3. Bluetooth 4.2 and Bluetooth Low Energy (BLE) support
4. GPIO pins, UART, I2C, SPI, ADC, DAC interfaces
5. 4 MB flash memory and 520 KB SRAM
6. Support for various operating systems, including Free RTOS and ESP-IDF Overall, the ESP32 is a powerful and flexible platform that enables developers to create a wide range of IoT applications that require wireless connectivity.

****

## Figure 3: WI-FI module

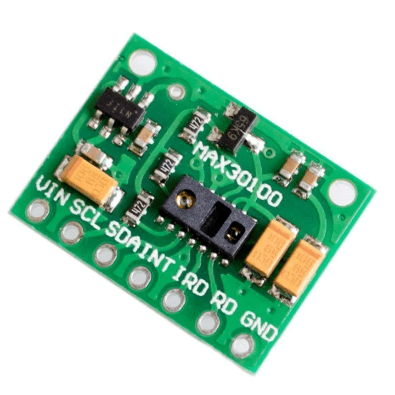
* **Temperature sensor:** A health temperature sensor is a device that is used to measure body temperature. Health temperature sensors are important for monitoring fever, which is a common symptom of many illnesses. They are also used in medical settings to monitor patients' temperatures during surgical procedures and hospital stays. Some health temperature sensors can be connected to smartphones or other devices via Bluetooth or Wi-Fi, allowing for easy monitoring and tracking of temperature readings over time.

**CHAPTER 4 PROPOSED METHODOLOGY**

## 

**Figure 4: Temperature sensor LM350**

* **Oxygen Sensor:**  Oxygen sensors are also used in healthcare monitoring to measure the oxygen saturation levels in a person's blood. This type of sensor is commonly known as a pulse oximeter. Pulse oximeters work by emitting a light through a person's skin and measuring the amount of light that is absorbed by the blood. The oxygen saturation level is then calculated based on the amount of light that is absorbed. These sensors are commonly used in hospitals and clinics to monitor patients with respiratory problems, such as asthma or COPD, or those undergoing anesthesia during surgery. They can also be used by individuals at home to monitor their oxygen saturation levels and detect any potential health issues. Oxygen sensors used in healthcare monitoring are highly accurate and can provide real-time information about a person's oxygen levels. This information can be used to identify potential health issues early and help prevent complications.

****

**Figure 5: Oxygen sensor MAX 30100**

* **LCD display:** LCD stands for Liquid Crystal Display, and it is a type of display technology commonly used in electronic projects. In electronic projects, LCDs can be used to display information, such as text or graphics. When using an LCD in a project, it is important to choose the right type of LCD for your needs and to make sure that your microcontroller or other device can interface with the LCD correctly.

**CHAPTER 4 PROPOSED METHODOLOGY**

****

**Figure 6: LCD Display**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

**5.1 Hardware Software Requirements**

## Hardware resources

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Sensor Name** | **Specification** |
| 1 | Oxygen sensor | Oxygen sensor MAX 30100 |
| 2 | Wi-Fi Module | Wi-Fi Module ESP32 |
| 3 | LCD Display | 16\*2 |
| 4 | Temperature sensor | Temperature sensor LM350 |

**Table 2: Hardware Requirements**

## Software resources

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Software language** | **Specification** |
| 1 | Think Speak | To Store Data on Cloud |

## 

## Table 3:Software Requirements

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

### 5.2 DFD’s (0, 1, ) level

Sensors

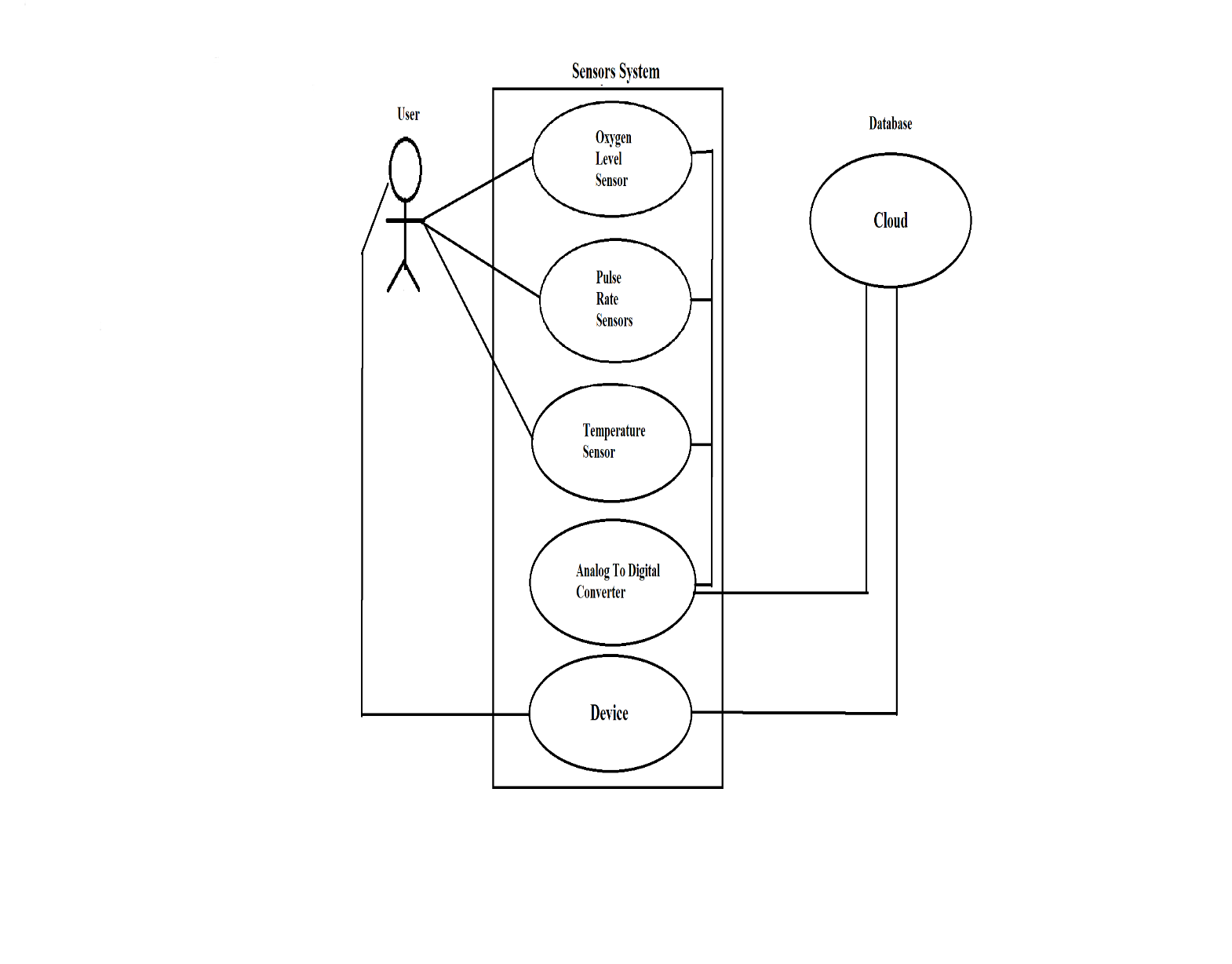
Analysing code of all sensors and collect data on cloud

Paitent

**Figure 7: DFD Level 0 of software application**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  | | --- | | Receiver receives |     Press  the Sensor    At receiver,  Data display the  Output of Paitent |
| Patient | t  Sensor connect to Paitent      Choose the selected path |
|  |

**Figure 8: DFD Level 1 of software application**

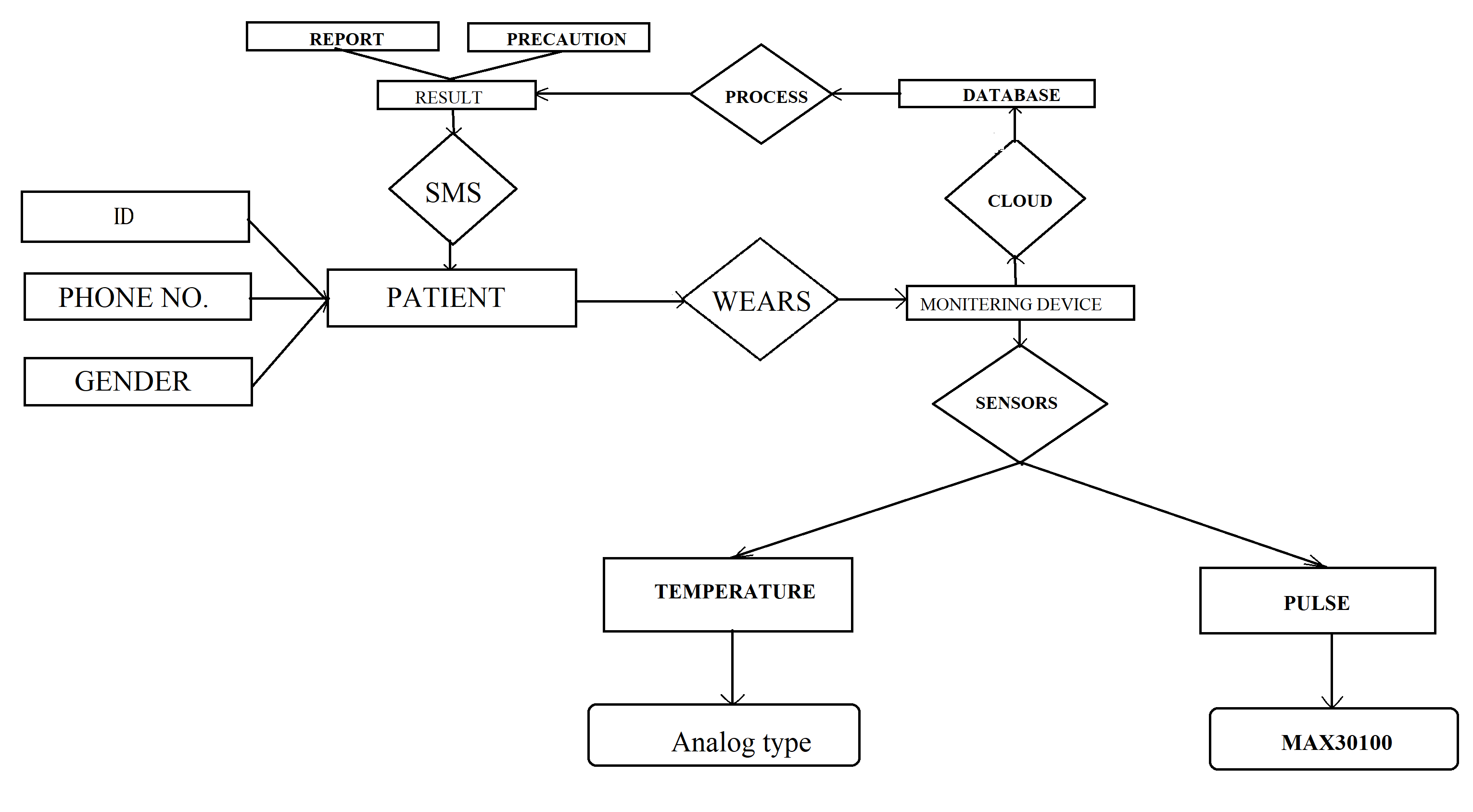
**5.3 Use Case Diagrams**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

**Figure 9: Use Case Diagram of Smart Health Monitoring**

**5.4 ER Diagram**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

****

**Figure 10: ER Diagram**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

### 5.5 Timeline Chart



**Fig 11: Timeline Chart**

### 5.6 IMPLEMENTATION

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

The following steps are involved in the implementation of the project:

Smart health monitoring can be implemented using various technologies and techniques, but here is one example of how it can be done:

1. Wearable devices: Smart health monitoring can be achieved by using wearable devices that are capable of tracking various vital signs such as heart rate, blood pressure, body temperature, and oxygen saturation. These devices can be connected to a smartphone or tablet via Bluetooth or Wi-Fi to monitor the user's health status in real-time.
2. Internet of Things (IoT) sensors: IoT sensors can be installed in various locations such as hospitals, clinics, and homes to monitor the health status of patients remotely. These sensors can be used to track vital signs such as blood pressure, heart rate, respiratory rate, and body temperature.
3. Artificial Intelligence (AI): AI can be used to analyze the data collected from wearable devices and IoT sensors. The AI algorithms can identify patterns and trends in the data, which can be used to predict potential health issues and provide early intervention.
4. Cloud computing: Cloud computing can be used to store and analyze the data collected from wearable devices and IoT sensors. The data can be securely stored in the cloud, and the AI algorithms can be run on the cloud servers to provide real-time health monitoring.
5. Mobile applications: Mobile applications can be developed to provide users with access to their health data and real-time health monitoring. These applications can also provide users with alerts and notifications if there are any abnormalities detected in their health status.
6. Electronic Health Records (EHRs): EHRs can be integrated with smart health monitoring systems to provide healthcare providers with access to real-time health data. This can help healthcare providers to make informed decisions about the patient's treatment and care.

Overall, smart health monitoring can improve the quality of care and reduce healthcare costs by

providing real-time health monitoring and early intervention.

**5.6. 1 Sample Code**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#include <SoftwareSerial.h>

float pulse = 0;

float temp = 0;

SoftwareSerial ser(9, 10);

String apiKey = "JW2AUM9WBCSIHNVJ";

// Variables

int pulsePin = A0; // Pulse Sensor purple wire connected to analog pin 0

int blinkPin = 7 ; // pin to blink led at each beat

int fadePin = 13; // pin to do fancy classy fading blink at each beat

int fadeRate = 0; // used to fade LED on with PWM on fadePin

// Volatile Variables, used in the interrupt service routine!

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! Must be seeded!

volatile boolean Pulse = false; // "True" when User's live heartbeat is detected. "False" when nota "live beat".

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

volatile boolean QS = false; // becomes true when Arduoino finds a beat.

// Regards Serial OutPut -- Set This Up to your needs

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()

{

lcd.begin(16, 2);

pinMode(blinkPin, OUTPUT); // pin that will blink to your heartbeat!

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

pinMode(fadePin, OUTPUT); // pin that will fade to your heartbeat!

Serial.begin(115200); // we agree to talk fast!

interruptSetup(); // sets up to read Pulse Sensor signal every 2mS

// IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS THAN THE BOARD VOLTAGE,

// UN-COMMENT THE NEXT LINE AND APPLY THAT VOLTAGE TO THE A-REF PIN

// analogReference(EXTERNAL);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" Patient Health");

lcd.setCursor(0, 1);

lcd.print(" Monitoring ");

delay(4000);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Initializing....");

delay(5000);

lcd.clear();

lcd.setCursor(0, 0);

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

lcd.print("Getting Data....");

ser.begin(9600);

ser.println("AT");

delay(1000);

ser.println("AT+GMR");

delay(1000);

ser.println("AT+CWMODE=3");

delay(1000);

ser.println("AT+RST");

delay(5000);

ser.println("AT+CIPMUX=1");

delay(1000);

String cmd = "AT+CWJAP=\"Alexahome\",\"98765432\"";

ser.println(cmd);

delay(1000);

ser.println("AT+CIFSR");

delay(1000);

}

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

// Where the Magic Happens

void loop()

{

serialOutput();

if (QS == true) // A Heartbeat Was Found

{

// BPM and IBI have been Determined

// Quantified Self "QS" true when arduino finds a heartbeat

fadeRate = 255; // Makes the LED Fade Effect Happen, Set 'fadeRate' Variable to 255 to fade LED with pulse

serialOutputWhenBeatHappens(); // A Beat Happened, Output that to serial.

QS = false; // reset the Quantified Self flag for next time

}

ledFadeToBeat(); // Makes the LED Fade Effect Happen

delay(20); // take a break

read\_temp();

esp\_8266();

}

void ledFadeToBeat()

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

{

fadeRate -= 15; // set LED fade value

fadeRate = constrain(fadeRate, 0, 255); // keep LED fade value from going into negative numbers!

analogWrite(fadePin, fadeRate); // fade LED

}

void interruptSetup()

{

// Initializes Timer2 to throw an interrupt every 2mS.

TCCR2A = 0x02; // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC MODE

TCCR2B = 0x06; // DON'T FORCE COMPARE, 256 PRESCALER

OCR2A = 0X7C; // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE

TIMSK2 = 0x02; // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A

sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED

}

void serialOutput()

{ // Decide How To Output Serial.

if (serialVisual == true)

{

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

arduinoSerialMonitorVisual('-', Signal); // goes to function that makes Serial Monitor Visualizer

}

else

{

sendDataToSerial('S', Signal); // goes to sendDataToSerial function

}

}

void serialOutputWhenBeatHappens()

{

if (serialVisual == true) // Code to Make the Serial Monitor Visualizer Work

{

Serial.print("\*\*\* Heart-Beat Happened \*\*\* "); //ASCII Art Madness

Serial.print("BPM: ");

Serial.println(BPM);

}

else

{

sendDataToSerial('B', BPM); // send heart rate with a 'B' prefix

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

sendDataToSerial('Q', IBI); // send time between beats with a 'Q' prefix

}

}

void arduinoSerialMonitorVisual(char symbol, int data )

{

const int sensorMin = 0; // sensor minimum, discovered through experiment

const int sensorMax = 1024; // sensor maximum, discovered through experiment

int sensorReading = data; // map the sensor range to a range of 12 options:

int range = map(sensorReading, sensorMin, sensorMax, 0, 11);

// do something different depending on the

// range value:

switch (range)

{

case 0:

Serial.println(""); /////ASCII Art Madness

break;

case 1:

Serial.println("---");

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

break;

case 2:

Serial.println("------");

break;

case 3:

Serial.println("---------");

break;

case 4:

Serial.println("------------");

break;

case 5:

Serial.println("--------------|-");

break;

case 6:

Serial.println("--------------|---");

break;

case 7:

Serial.println("--------------|-------");

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

break;

case 8:

Serial.println("--------------|----------");

break;

case 9:

Serial.println("--------------|----------------");

break;

case 10:

Serial.println("--------------|-------------------");

break;

case 11:

Serial.println("--------------|-----------------------");

break;

}

}

void sendDataToSerial(char symbol, int data )

{

Serial.print(symbol);

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

Serial.println(data);

}

ISR(TIMER2\_COMPA\_vect) //triggered when Timer2 counts to 124

{

cli(); // disable interrupts while we do this

Signal = analogRead(pulsePin); // read the Pulse Sensor

sampleCounter += 2; // keep track of the time in mS with this variable

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)

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

{ // thresh condition helps avoid noise

P = Signal; // P is the peak

} // keep track of highest point in pulse wave

// NOW IT'S TIME TO LOOK FOR THE HEART BEAT

// signal surges up in value every time there is a pulse

if (N > 250)

{ // avoid high frequency noise

if ( (Signal > thresh) && (Pulse == false) && (N > (IBI / 5) \* 3) )

{

Pulse = true; // set the Pulse flag when we think there is a pulse

digitalWrite(blinkPin, HIGH); // turn on pin 13 LED

IBI = sampleCounter - lastBeatTime; // measure time between beats in mS

lastBeatTime = sampleCounter; // keep track of time for next pulse

if (secondBeat)

{ // if this is the second beat, if secondBeat == TRUE

secondBeat = false; // clear secondBeat flag

for (int i = 0; i <= 9; i++) // seed the running total to get a realisitic BPM at startup

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

{

rate[i] = IBI;

}

}

if (firstBeat) // if it's the first time we found a beat, if firstBeat == TRUE

{

firstBeat = false; // clear firstBeat flag

secondBeat = true; // set the second beat flag

sei(); // enable interrupts again

return; // IBI value is unreliable so discard it

}

// keep a running total of the last 10 IBI values

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

}

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

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

// QS FLAG IS NOT CLEARED INSIDE THIS ISR

pulse = BPM;

}

}

if (Signal < thresh && Pulse == true)

{ // when the values are going down, the beat is over

digitalWrite(blinkPin, 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;

}

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

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(); // enable interrupts when youre done!

}// end isr

void esp\_8266()

{

// TCP connection AT+CIPSTART=4,"TCP","184.106.153.149",80

String cmd = "AT+CIPSTART=4,\"TCP\",\"";

cmd += "184.106.153.149"; // api.thingspeak.com

cmd += "\",80";

ser.println(cmd);

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

Serial.println(cmd);

if (ser.find("Error"))

{

Serial.println("AT+CIPSTART error");

return;

}

String getStr = "GET /update?api\_key=";

getStr += apiKey;

getStr += "&field1=";

getStr += String(temp);

getStr += "&field2=";

getStr += String(pulse);

getStr += "\r\n\r\n";

// send data length

cmd = "AT+CIPSEND=4,";

cmd += String(getStr.length());

ser.println(cmd);

Serial.println(cmd);

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

delay(1000);

ser.print(getStr);

Serial.println(getStr); //thingspeak needs 15 sec delay between updates

delay(3000);

}

void read\_temp(){

int temp\_val = analogRead(A1);

float mv = (temp\_val / 1024.0) \* 5000;

float cel = mv / 10;

temp = (cel \* 9) / 5 + 32;

Serial.print("Temperature:");

Serial.println(temp);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("BPM :");

lcd.setCursor(7, 0);

lcd.print(BPM);

lcd.setCursor(0, 1);

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

lcd.print("Temp.:");

lcd.setCursor(7, 1);

lcd.print(temp);

lcd.setCursor(13, 1);

lcd.print("F");}

**5.6.2 Types of Testing**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

* **Software testing:**

Smart Health Monitoring typically involves several steps to ensure that the system is functioning as intended and that it meets the requirements of its users. Here are a few common steps involved in testing such systems

1. Functional Testing: This step is used to verify whether the system functions according to the design specifications mentioned in the requirement documents
2. Usability Testing: In this phase, the user interface of the system is tested to ensure that it is user-friendly and easy to use.
3. Performance Testing: The performance of the system is tested under various conditions to ensure that it can handle high loads and operate efficiently.
4. Security Testing: Security testing is performed to identify vulnerabilities in the system and to prevent unauthorized access or data breaches.
5. Integration Testing: This step involves testing the integration of the various modules of the system to ensure that they work together seamlessly.

Acceptance Testing: In this final stage, the system is tested by stakeholders to ensure that it meets their specific requirements. It is important to conduct thorough testing of Smart Health Monitoring systems to ensure that they are reliable, secure, and meet the needs of their users.

•**Unit Testing**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

Unit Testing is an essential part of the software development process and is particularly important for Smart Health Monitoring systems. Unit testing involves testing individual components of the system, such as functions, modules or classes, to ensure that they work correctly. Here are some key aspects to consider when performing unit testing for Smart Health Monitoring systems:

1. Test Coverage: It's important to ensure that all functionality of the system is covered by unit tests. This includes testing different inputs, edge cases, and error conditions.
2. Test Data: It's important to use realistic test data when performing unit tests. This can help identify issues with data validation or handling.
3. Automation: Automating unit tests can save time and effort, allowing developers to run tests frequently and catch issues early.
4. Mocking: In Smart Health Monitoring systems, there may be external dependencies, such as sensors or APIs. Using a mocking framework can help simulate these dependencies and make testing more efficient.
5. Regression Testing: After fixing defects found in unit testing, it's important to re-run relevant tests to ensure that other parts of the system were not inadvertently affected.

In summary, unit testing is crucial for ensuring the quality and reliability of Smart Health Monitoring systems. It helps catch issues early in the development process, decreases the likelihood of bugs and errors, and ultimately leads to a better user experience.

## • Integration Testing

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

## Integration testing for smart health monitoring involves testing the integration of various components that make up the system. Some examples of components that may be tested during integration testing include:

## Sensors: These are the devices that collect health data, such as heart rate, blood pressure, and temperature.

## Data storage: The data collected by the sensors needs to be stored somewhere, whether that's in a local database or in the cloud.

## Analytics engine: This component processes the data collected by the sensors and generates insights and alerts based on the data

## User interface: This is the part of the system that the user interacts with, whether that's a mobile app or a web-based dashboard.

## Integration testing is typically done after unit testing, where each individual component is tested in isolation, and before system testing, where the entire system is tested as a whole. It helps to catch any issues that may arise when different components are brought together, ensuring that the final product is reliable and functional.

## • Functional Testing

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

Functional testing for smart health monitoring involves testing the features and functionality of the system to ensure that it is performing as intended. Some key areas that should be tested include:

1. Data collection: The system should be able to collect accurate data from various sensors and other devices. This data should be transmitted securely and stored appropriately.
2. Data analysis: The system should be able to analyze the collected data in real-time, detecting any anomalies or abnormalities and alerting users and healthcare providers when necessary.
3. User interface: The user interface should be intuitive and easy to use, allowing users to interact with the system effectively
4. Integration: The system should be able to integrate with other devices and systems, such as electronic health records (EHRs), to provide a comprehensive view of a patient's health
5. Security: The system should have robust security measures in place to protect sensitive patient data.

## • System Testing

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

System Testing for Smart Health Monitoring involves a series of tests to ensure that the system is working as intended. Here are some common types of system testing for smart health monitoring:

1. Functional Testing: This type of testing ensures that the system functions properly and meets all the requirements specified in the design. It includes testing the features of the system, such as data collection, analysis, and alerts.
2. Performance Testing: This type of testing evaluates the system's ability to perform under various conditions. It includes stress testing, load testing, and endurance testing to ensure that the system can handle large volumes of data and traffic.
3. Usability Testing: This type of testing evaluates the user experience of the system. It includes testing the ease of use and navigation of the system, as well as the clarity and relevance of the data displayed.
4. Security Testing: This type of testing ensures that the system is secure and protected from unauthorized access. It includes testing the system's authentication, encryption, and access control mechanisms.
5. Compatibility Testing: This type of testing ensures that the system works as intended with different devices, platforms, and operating systems. It includes testing the system's compatibility with different browsers, devices, and networks.

By conducting these tests, you can ensure that your smart health monitoring system is reliable, efficient, and secure.

## • Usability Testing

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

Usability testing for smart health monitoring is an essential part of the system testing process. It helps to ensure that the system is easy to use, intuitive, and efficient for users. Here are some common steps involved in conducting usability testing for smart health monitoring:

1. Define User Profiles: The first step in usability testing is to identify the different types of users who will be interacting with the system. This could include healthcare professionals, patients, or caregivers. By defining user profiles, you can better understand the needs and expectations of each group.
2. Identify Key Tasks: Once you have defined the user profiles, the next step is to identify the key tasks or activities that users will perform using the system. This could include setting up the system, entering data, reviewing results, or setting notifications.
3. Create Test Scenarios: Based on the identified tasks, create test scenarios that simulate real-world usage of the system. These scenarios should be designed to evaluate how well the system meets the needs of different user groups.
4. Recruit Participants: Find participants who match the defined user profiles to participate in the usability testing. You can recruit participants from clinics, hospitals, or through online platforms.
5. Conduct Tests: Provide participants with the test scenarios and observe them as they perform the tasks. Ask them to provide feedback on their experience, such as ease of use, clarity of instructions, and overall satisfaction with the system.

## • GUI Testing

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

## GUI testing for smart health monitoring involves evaluating the graphical user interface (GUI) of the system to ensure that it is functioning as intended. Here are some common steps involved in conducting GUI testing for smart health monitoring:

## Identify All GUI Elements: The first step in GUI testing is to identify all the GUI elements used in the system, such as buttons, text boxes, and drop-down menus. This will help you to create a comprehensive checklist of all elements to test.

## Verify Functionality: Once you have identified all the GUI elements, verify that they are functioning properly. For instance, check that clicking on buttons or entering data into text boxes leads to the expected results

## Validate Layout and Formatting: Check that the layout and formatting of the GUI elements are consistent across different screens and devices. This includes verifying that fonts, colors, and images are displayed correctly and that the overall design is visually appealing.

## Evaluate Navigation and Interactions: Evaluate how users move through the system and interact with the GUI elements. Test the navigation between screens, the ability to search for information, and the ease of performing tasks.

## Verify Accessibility: Ensure that the system is accessible to users with disabilities. This includes checking that the GUI elements can be navigated using assistive technology such as screen readers and that the overall design meets accessibility guidelines.

## Test Error Handling: Test how the system handles errors, such as incorrect input, missing information, or system crashes. Verify that error messages are displayed clearly and that users are guided through the process of correcting the errors.

## • Performance Testing

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

Performance testing for smart health monitoring involves evaluating the system's ability to handle user traffic and respond quickly to requests, while also maintaining system stability and reliability. Here are some key areas that should be covered in performance testing:

1. Load Testing: This type of testing involves simulating a large number of users accessing the smart health monitoring system simultaneously, to determine its maximum capacity and identify any bottlenecks or issues that may arise under heavy traffic.
2. Stress Testing: Stress testing is used to evaluate how well the system can handle sudden spikes in traffic or increased load beyond its normal capacity.
3. Endurance Testing: Endurance testing is performed to evaluate the system's ability to sustain prolonged usage over time, such as for 24 hours or more.
4. Scalability Testing: Scalability testing is used to determine whether the system can handle increased demand by adding additional resources or expanding infrastructure.
5. Security Testing: Security testing is crucial for ensuring that the smart health monitoring system is secure from external threats and can protect sensitive user data.
6. Usability Testing: Usability testing ensures that the smart health monitoring system is easy to use and navigate, with clear instructions and intuitive design elements that make it accessible to all users.

**5.6.3.Test Case**

**CHAPTER 5 DETAILS DESIGN AND WORKING PROSESSES**

**Test Case ID: Test\_1**  **Test Designed by:** Karan,Pranav,Soham,Ram

**Test Priority (Low/Medium/High):** Medium  **Test Designed date:** 30th Jan 2022

**Module Name:** Software (User Side)  **Executed by:** Karan,Pranav,Soham,Ram  **Test Title:** Verify the User side of the system  **Test Execution date:** 30th Jan 2022

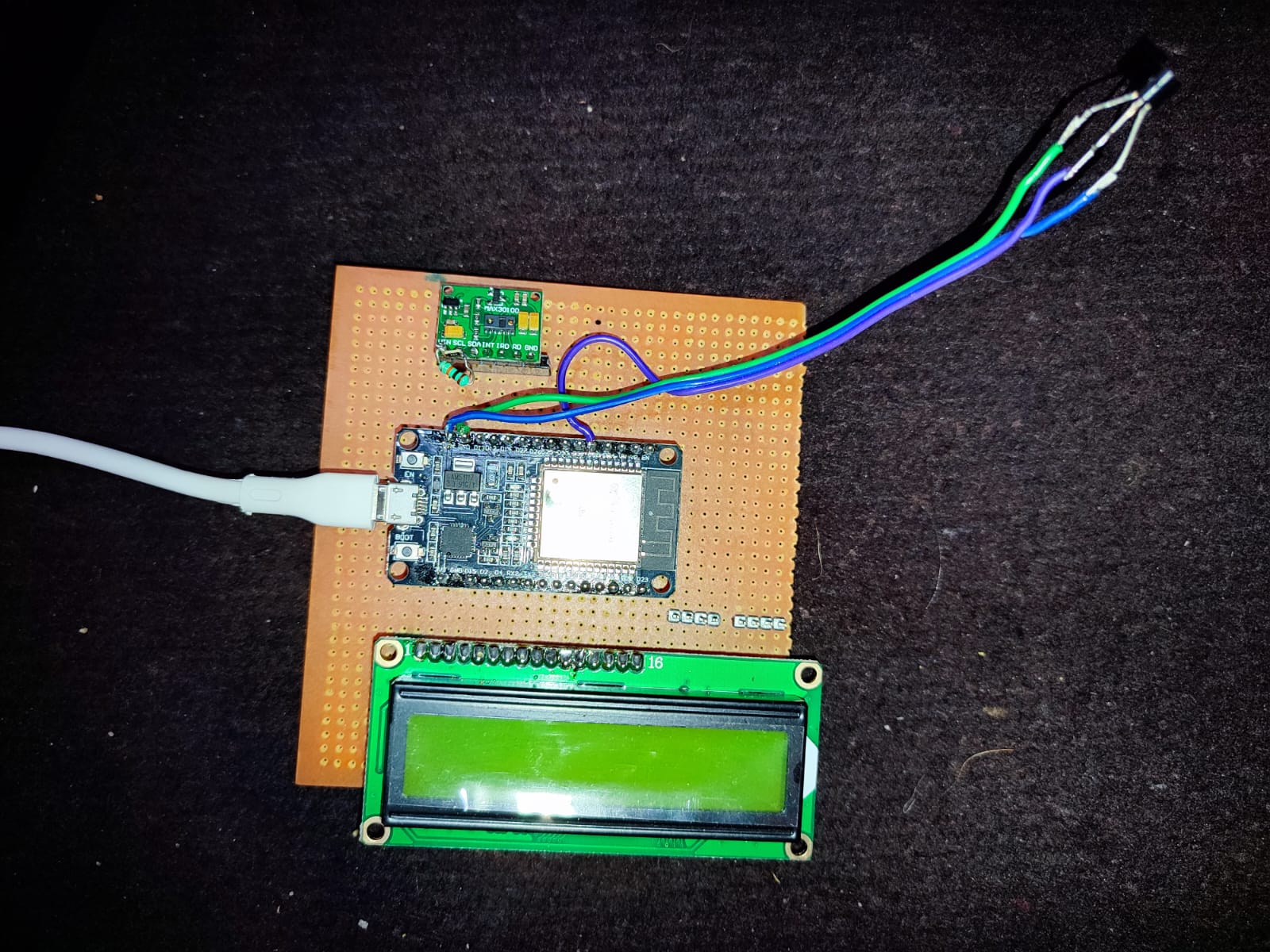
**Description:** Testing the User side of system

**Pre-conditions: buttons should work and can open other modules**

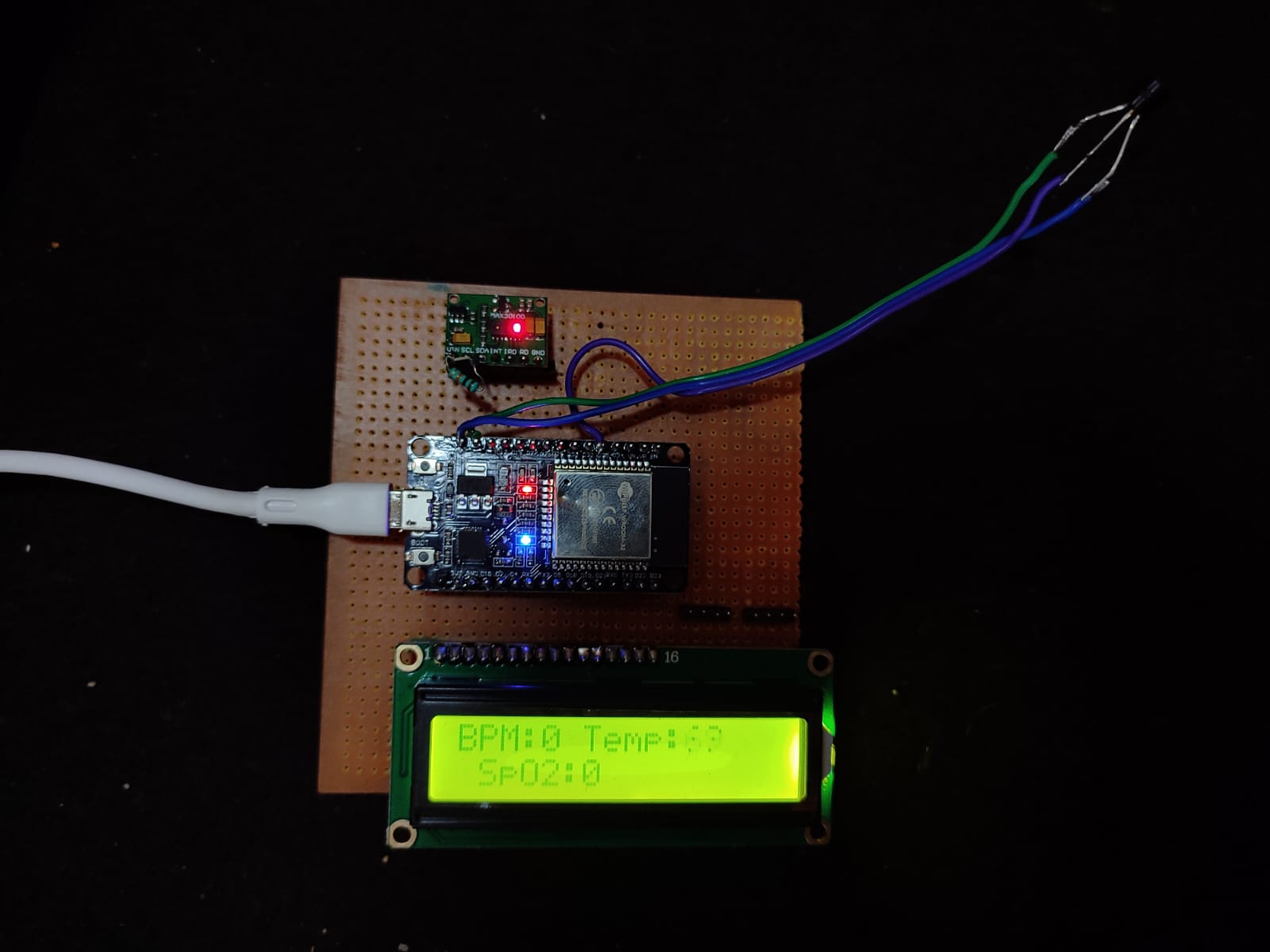
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. no. | Test case ID | Description | Input Data | Expected Result | Actual Result | Status |
| 1. | TC1 | Karan Pagar  Age:19 | Temperature,  Pulse,  Oxygen Level | 97.4F-99.6F  60-100bpm  95-100 sp02 | 98.3F  78bpm  99sp02 | Pass |
| 2. | TC2 | Rohan Kulkarni  Age:26 | Temperature,  Pulse,  Oxygen Level | 97.4F-99.6F  60-100bpm  95-100 sp02 | 98.5F  83bpm  100sp02 | Pass |
| 3. | TC3 | Pranav Chaudhari  Age:21 | Temperature,  Pulse,  Oxygen Level | 97.4F-99.6F  60-100bpm  95-100 sp02 | 97.9F  86bpm  100sp02 | Pass |
| 4. | TC4 | Ram Gitte  Age:20 | Temperature,  Pulse,  Oxygen Level | 97.4F-99.6F  60-100bpm  95-100 sp02 | 98.7F  88bpm  98sp02 | Pass |
| 5. | TC5 | Manohar Samvase  Age:38 | Temperature,  Pulse,  Oxygen Level | 97.4F-99.6F  60-100bpm  95-100 sp02 | 98.2F  92bpm  99sp02 | Pass |

### 6.1 Screenshots

**CHAPTER 6 RESULTS AND APPLICATION**



**Figure 6.1: Smart Health Monitoring System**

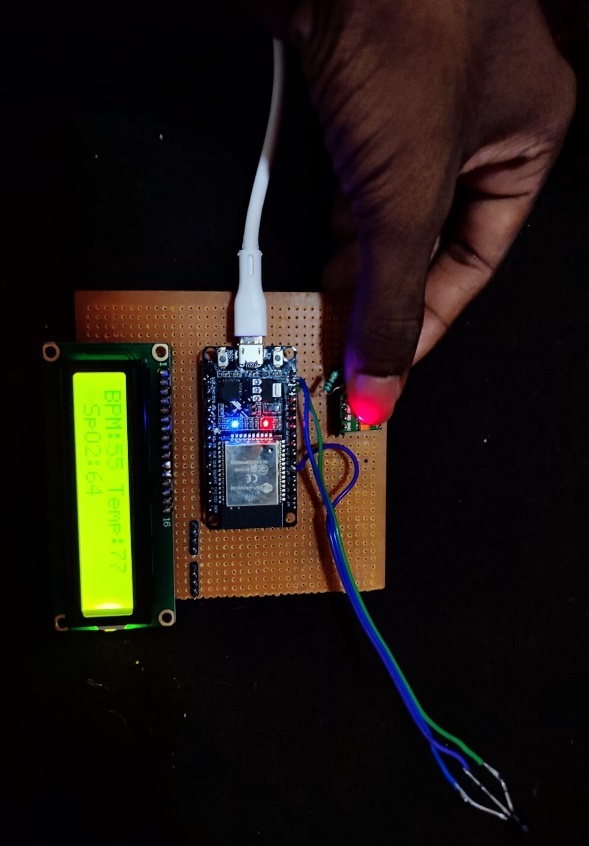


## Figure 6.2: System in Ready state to work

When the System is connected to the Laptop/ Computer then it get turn on and displays the Blood pressure ,

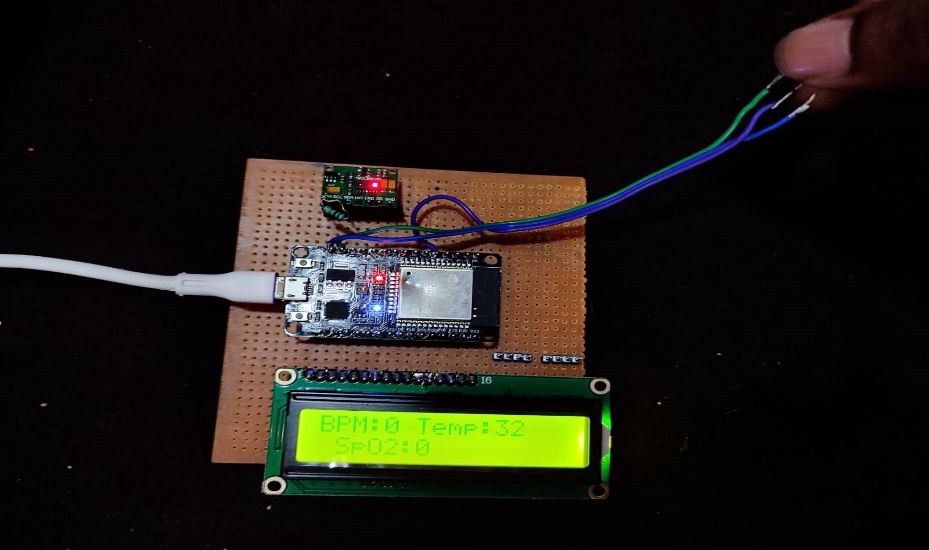
**CHAPTER 6 RESULTS AND APPLICATION**

Temperature meter and Oximeter reading . when the Patient will put their Finger on the Sensor then it will

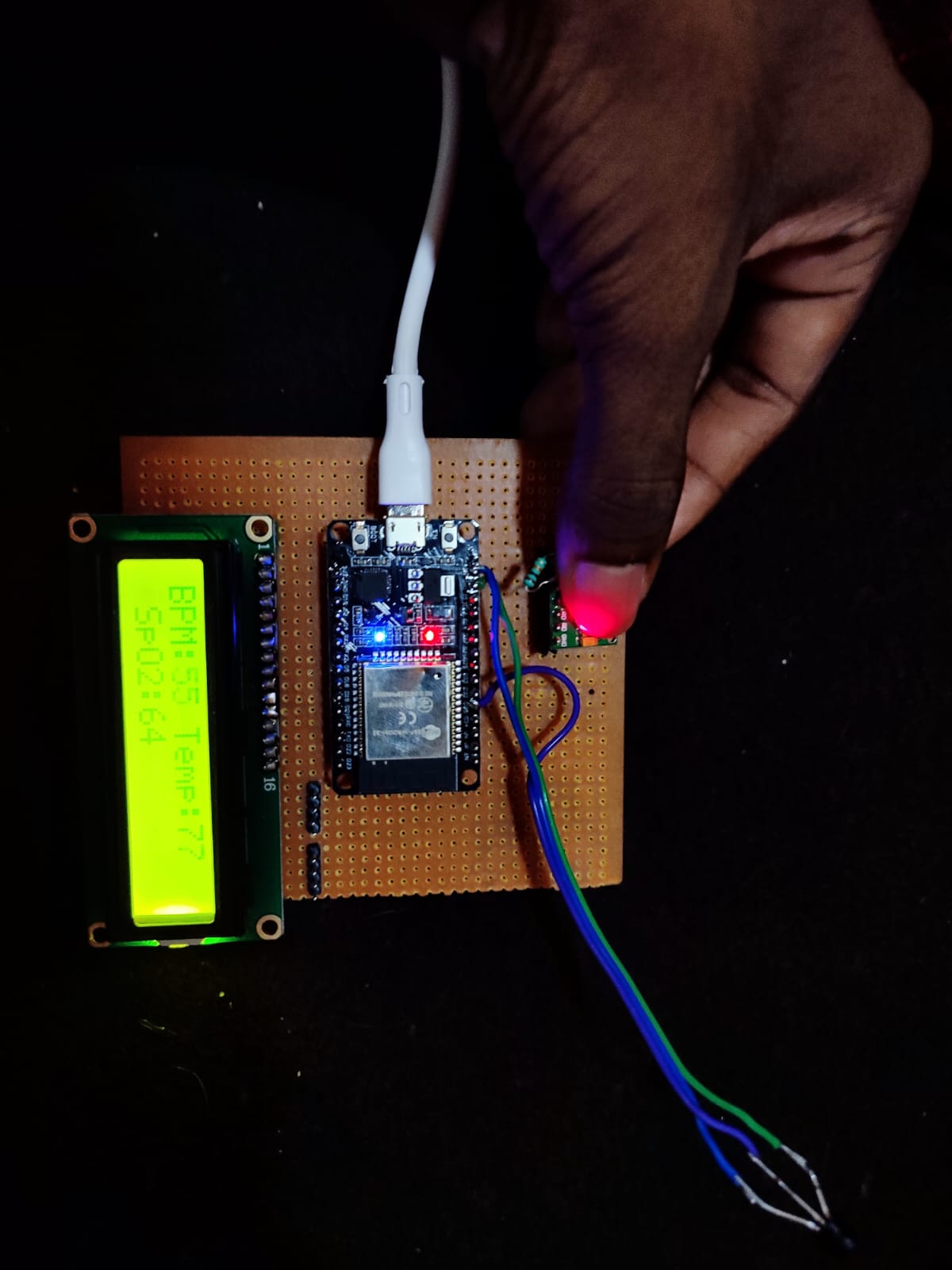
Measure the Blood Pressure and Oxygen Level. And after pressing the Temperature Sensor it measure

the temperature of body and gives the reading to patients.

## 

 **Figure 6.3 Patient is checking Blood Pressure and Oxygen Level**

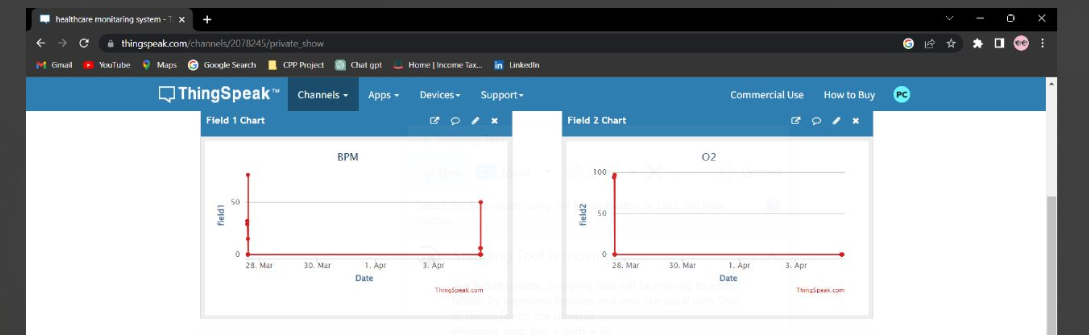
**Figure 6.4 Patient is Measuring the Temperature of Body**

 The figure show that when the Patient will press the Sensor first it will display it’s reading first

**CHAPTER 6 RESULTS AND APPLICATION**

like Temperature Sensor.

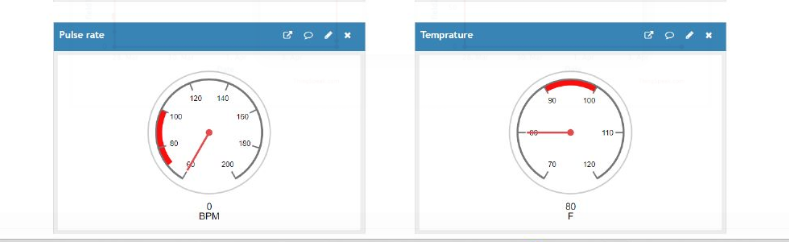
**Figure 6.5: All Sensors are working and displaying the Readings**



**CHAPTER 6 RESULTS AND APPLICATION**

## Figure 6.5.1 Graph and Readings of Blood Pressure and Oxygen Level

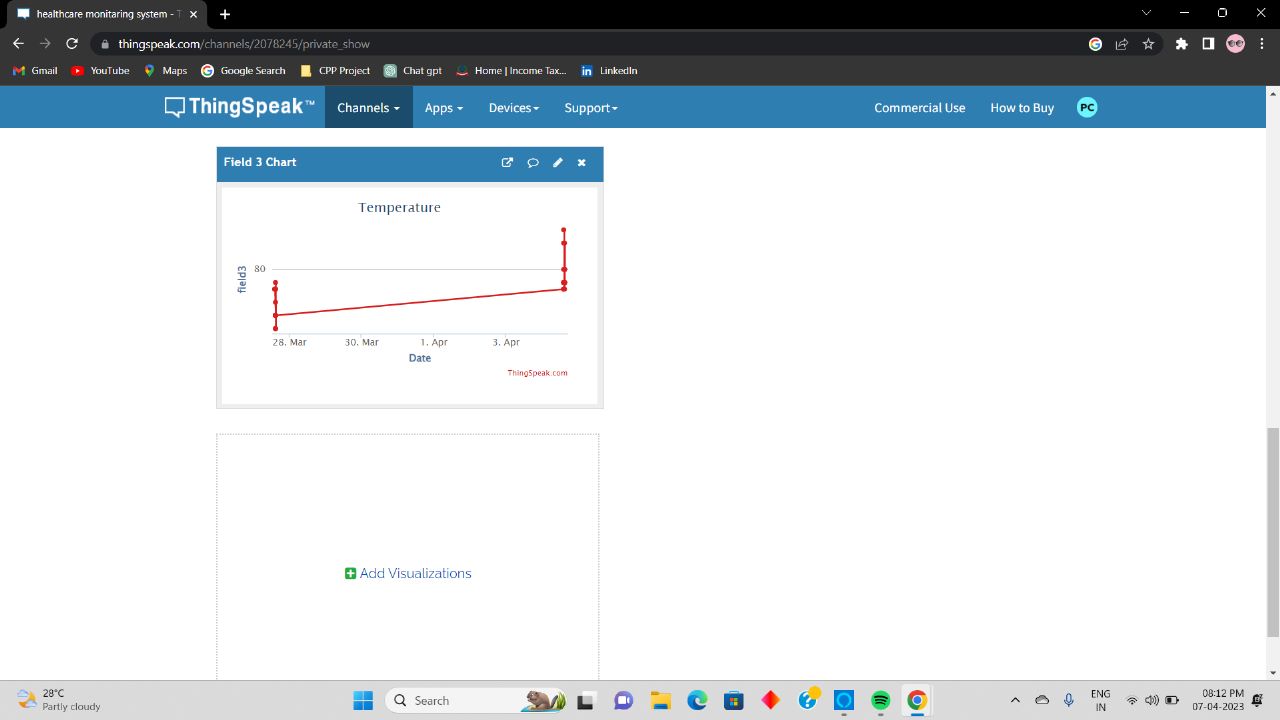
## 



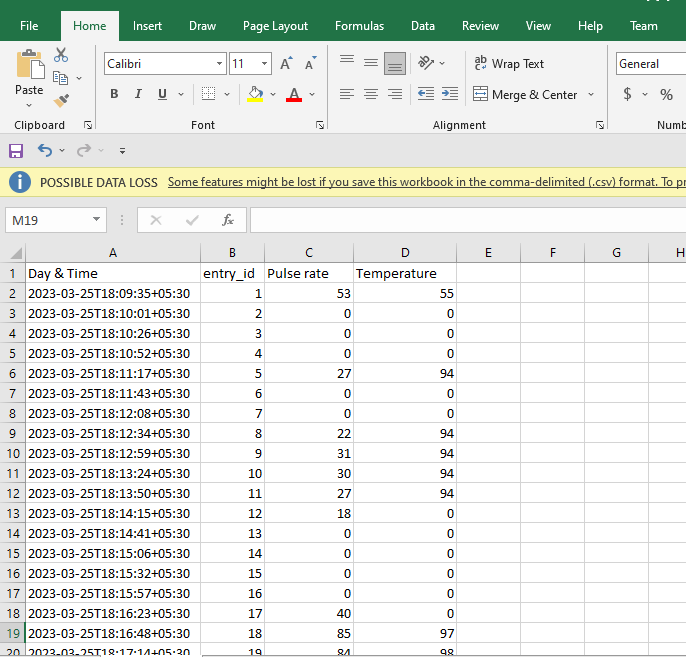
## Figure 6.6: Pulse rate and Temperature reading

## 

**CHAPTER 6 RESULTS AND APPLICATION**

 **Figure6.7: The web application use to store the Data**

## Figure 6.7.1: Graph of Temperature Sensor



**CHAPTER 6 RESULTS AND APPLICATION**

### Figure 6.7.2: Data is Stored in Excel Sheet

### 6.2 Applications

**CHAPTER 6 RESULTS AND APPLICATION**

**Applications of Smart Health Monitoring are as follows:**

* Remote patient monitoring
* Fitness tracking
* Clinical trials
* Medication adherence

**CHAPTER 7 CONCLUSION AND FUTURE SCOPE**

### 7.1 Conclusion

The use of smart health monitoring systems has the potential to improve health outcomes, and reduce healthcare costs by preventing hospitalizations and enabling earlier treatment. However, it is important to ensure that these systems are designed with user privacy and security in mind, and that they are accessible and affordable to all patients, regardless of socioeconomic status.

### 7.2 Future Scope

Smart health monitoring projects have tremendous potential for the future. With the increasing prevalence of chronic diseases and the aging population, there is a growing need for remote health monitoring systems that can provide timely and accurate information about a patient's health status. Smart health monitoring projects have been gaining significant attention in recent years due to their potential to revolutionize the healthcare industry. These projects involve the use of advanced technologies such as sensors, wearables, and artificial intelligence to monitor various health parameters of individuals in real-time.

**CHAPTER 8 REFERENCES , BIBLIOGRAPHY,**

[1] Mrunal Fatangare and Hemlata Ohal “An Empirical Survey on Early Health Condition Prediction based on Clinical Parameters” Journal of Engineering and Technology.

[2]. M. Fatangare, P. U. Nehete, L. R. Mahajan, A. J. Nair and S. Madhurkar, “Identification And Expenditure Tracking Using RFID Based Intelligent Card,” 2020 4thInternational Conference on trends in Electronics and Informatics (ICOEI)(48184), 2020,pp. 573-579,doi:10.1109/ICOEI48184.2020.9143033

### [3]. M. Fatangare, A. Nimbalkar, G. Chite, A. Narkhede and A. Khilnani, “An Efficient Temperature Monitoring using Raspberry Pi,” 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 1-5, doi:10.1109/ICICT48043.2020.9112376

[4]. IoT BASED SMART HEALTH MONITORING SYSTEM, U. B. Mahadevaswamy, Durgad Zaibabuktiyar

[5]. IoT-Based Smart Health Unit Syed Muhammad Waqas Shah, Maruf Pasha 01 Jan 2017-Journal of Software-Vol.12

[6]. IoT BASED CARE SYSTEM BASED ON IoT Jeon Seong Ho

[7]. Development of Smart Health Monitoring System D. S. Dayanal G. Kalpana2 D. Godwin Immanuel

[8].A Smart Healthcare Monitoring System Using Smartphone Interface Aksht1, Gaurav1 , Zahid1, Bhupendra1, Aditi1, Sachin Kumar1, Maneesha1, P. K. Pandey

[9].Android based health care monitoring system Maradugu Anil Kumar1, Y. Ravi Sekhar1

[10].A health remote monitoring application based on wireless body area network Elhoussaine Baba1, Abdelillah Jibab1, Ahmed Hammouch1 01 Apr 201

*Smart Health Monitoring System Using Pulse and Temperature Sensor*

Prof.Mrunal Fatangare   
 *Departement of Information Technology*  
MAEER’S MIT PolytechnicPune,India  
mrunal.fatangare@mitwpu.edu.in

Soham kokate  
*Department of Information Technology*  
MAEER’S MIT Polytechnic, Pune, India  
kokatesoham824@gmail.com  
  
 Karan Pagar  
*Department of Information Technology*

MAEER’S MIT Polytechnic, Pune, India pagarkaran5502@gmail.com  
 Ram Gitte  
*Department of Information Technology*  
 MAEER’S MIT Polytechnic, Pune, India  
 ramgitte42@gmail.com

Pranav Chaudhari  
 *Department of Information Technology*  
 *MAEER’s MIT Polytechnic*Pune, India  
pranav7545@gmail.com

*Abstract*—for those who require ongoing monitoring that cannot be offered outside of hospitals, smart healthcare is crucial. Additionally, it is crucial in rural areas where adjacent clinics may communicate with metropolitan hospitals on the health of their patients. This study proposes a smart health monitoring system that notifies the concerned party through SMS while monitoring the patient's state using biomedical sensors. Here, the biomedical sensors are interfaced to a Wi-Fi module controller, which reads the data, and outputs it to an LCD display or serial monitor. For a smartphone to show data, SMS must be used. a message sent by the patient's family members to make it simple to access the patient's information. Monitoring the patient at home and regularly checking on their health is a particularly difficult chore in the routinely busy work. It is important to periodically check on older people's health in order to prevent serious medical emergencies.

# Introduction (*Heading 1*)

A smart health monitoring system is a type of technology that provides continuous monitoring of a person's health and wellness using a range of sensors, wearables, and other connected devices. The system may measure vital signs including heart rate, blood pressure, and temperature as well as physical activity, sleep habits, and other health-related data.

The collected data is transmitted to a central platform, such as a mobile app or web-based dashboard, where it can be analysed and interpreted by healthcare professionals, caregivers, or the individual themselves. This can help to identify potential health problems early on and allow for proactive interventions.

Smart health monitoring systems can be used in a variety of settings, including hospitals, clinics, and homes. They can be especially useful for individuals with chronic health conditions, older adults, and those who require close monitoring due to illness or injury.

Overall, smart health monitoring systems have the potential to improve healthcare outcomes, reduce healthcare costs, and enhance the quality of life for individuals by enabling more personalized and proactive care. A smart health monitoring system project is a system designed to monitor an individual's health and wellness, which can include tracking vital signs, physical activity, and other health-related data. The project aims to improve the quality of life for individuals by helping them manage their health and wellness more effectively. The system consists of various sensors, wearable devices, and other IoT (Internet of Things) technologies that collect and analyze data on the individual's health status. The data is then transmitted to a central monitoring system that processes and interprets the information to provide personalized health insights. One of the key benefits of a smart health monitoring system is its ability to detect and alert individuals or healthcare professionals about potential health problems before they become serious.

In addition to improving health outcomes, a smart health monitoring system can also reduce healthcare costs by minimizing hospitalizations and emergency room visits.Overall, a smart health monitoring system project is a promising area of research and development that has the potential to revolutionize the way individuals manage their health and wellness. which can include tracking vital signs, physical activity, and other health-related data. The project aims to improve the quality of life for individuals by helping them manage their health and wellness more effectively. The system consists of various sensors, wearable devices, and other IoT (Internet of Things) technologies that collect and analyze data on the individual's health status. The data is then transmitted to a central monitoring system that processes and interprets the information to provide personalized health insights.One of the key benefits of a smart health monitoring system is its ability to detect and alert individuals or healthcare professionals about potential health problems before they become serious. In addition to improving health outcomes, a smart health monitoring system can also reduce healthcare costs by minimizing hospitalizations and emergency. Smart health monitoring systems are becoming increasingly popular due to their ability to provide real-time health monitoring and analysis of patients. They are designed to improve the accuracy of medical diagnosis, reduce the time required for diagnosis, and improve patient outcomes. Smart health monitoring systems have gained increasing popularity over the past few years due to the growth in the use of connected devices and Internet of Things (IoT) technology. These systems allow for the collection and analysis of health data in real-time, enabling healthcare professionals to monitor and track patient health remotely.

# Literature review

This paper discusses the development of a crossover parametric model for the cross-analysis of diseases such as cancer, cardiovascular disease, and diabetes. The model aims to improve the precision with which these diseases are recognized by analysing the influence of fluctuations in bodily parameters. The paper compares several machine learning and deep learning algorithms for disease detection using cross-body parameters. Results show that the system increases the accuracy of disease detection compared to non-cross parametric techniques. The paper examines several related studies and discusses methods, styles, actions, and procedures used in developing the crossover parametric model. [1]

The abstract describes a system that uses RFID technology to track student attendance and financial transactions at schools and universities. Each student is provided with a card containing an RFID chip that serves as an ID, security, and payment method. The system aims to simplify the attendance-taking process and reduce errors that may occur with manual methods. It also captures classroom data for the distribution of appropriate attendance scores, which can be used for administrative decisions. Additionally, the system uses the RFID tag as a key for the student's authorized bank account to manage account-related transactions. The abstract highlights the versatility of RFID technology in various industries, such as transportation, medical services, farming, and hospitality.[2]

The paper discusses an IoT-based smart health monitoring system that has the capability to monitor the health status of patients remotely and provide real-time feedback to healthcare providers. This system comprises various IoT devices, such as wearables, sensors, and mobile applications, which collect and transmit the health data of patients to a centralized platform. The platform analyzes this data using various machine learning algorithms and provides insights and recommendations to healthcare providers for timely intervention. [3]

The proposed system is a Temperature Monitoring system that uses various components such as a Temperature Sensor, Jumper Wires, Programmable Logic Controller, and Analog Card & Raspberry Pie 3 to track temperature in various places. One of the significant improvements over the old system is that the data is stored in a Database & can be accessed from all over the World. The system is divided primarily into two sections: equipment and programming. Raspberry pi takes data and saves it in the MySQL database, which is connected to a user-friendly Webpage that displays the extracted data. The system can be used in industrial furnaces with equal efficiency, making it suitable for various temperature monitoring applications.[4]

The paper focuses on addressing the healthcare challenges faced by remote and underdeveloped areas, where access to healthcare services is limited due to the lack of proper medical infrastructure. The proposed solution is an IoT-based smart health monitoring system that can provide basic health services and access to medical staff remotely. The proposed system consists of three layers: the sensor layer, the network access layer, and the service layer. The sensor layer includes various sensors, such as temperature, blood pressure, and heart rate sensors, that collect patient health data. The network access layer provides connectivity between the sensors and the service layer, allowing for efficient communication. The service layer includes medical professionals who analyze the data and provide medical advice and treatment remotely.[5]

Overall, the research paper highlights the potential of IoT-based healthcare systems to transform the way healthcare is delivered and managed. With the increasing prevalence of connected devices and the growing demand for more personalized and efficient healthcare services, it is likely that IoT-based smart care systems will play an increasingly important role in the future of healthcare. [6]

The proposed Android application is designed to be user-friendly and easily accessible to people of all ages and fitness levels. It can be downloaded from the Google Play Store and used on any Android device, making it convenient for users to access their fitness information and track their progress on the go. In addition to providing essential features such as BMI and BMR calculators, workout plans, and meal plans, the application also has a calorie counter that can help users monitor their calorie intake and ensure they are staying within their daily limits.[7]

The proposed smart healthcare monitoring framework using sensors and the IoT in the paper is designed to address the challenges faced by the healthcare sector in rural areas of India. The shortage of healthcare personnel and infrastructure in these areas makes it difficult for patients to access quality healthcare services. The proposed system aims to bridge this gap by enabling remote monitoring of medical parameters and tracking of medical equipment, leading to the provision of smart hospital services. The system consists of a network of sensors and devices that are deployed in rural hospitals and clinics. The sensors are used to monitor the vital signs of patients, such as blood pressure, heart rate, and temperature. The data collected by the sensors is transmitted to a central system using the IoT, where it is analyzed by machine learning algorithms to detect patterns and anomalies in the patient's health status. [8]

The paper suggests the use of a system for continuous monitoring of a patient's health parameters, which includes heart rate, oxygen saturation level, and temperature. The system utilizes a web server and an Android application, enabling doctors to remotely monitor the patient's condition. The application allows doctors to access the patient's medical history stored on the server, providing a comprehensive overview of their health. This system aims to improve patient care by allowing for timely intervention and management of potential health risks. The paper further discusses the technical details of the system, including the hardware and software components, as well as the security measures implemented to protect patient data.[9]

The development of a wearable wireless body area network (WBAN) application for remote health monitoring is a significant advancement in healthcare technology. The application is designed to monitor physiological signals and provide real-time transmission of data. The application employs four biomedical sensor nodes that collect data such as ECG, body temperature, blood pressure, and pulse rate. The sensors are placed on the body and are connected to a central node that wirelessly transmits the data to a monitoring PC. The data collected by the sensors can be analyzed by healthcare professionals to track patient health and detect any abnormalities. The use of wearable WBAN applications has great potential in remote health monitoring and can significantly improve patient outcomes by allowing for continuous and real-time monitoring of vital signs.[10]

III. ARCHITECTURE

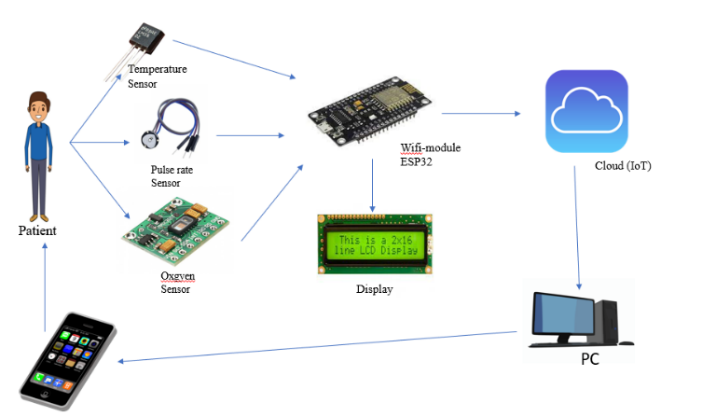


Fig 1. System Architecture

A smart health monitoring system is a complex project that typically involves the use of various hardware and software components. The architecture of such a system can vary depending on its specific requirements, but here are some key components that are typically involved:

**Sensors**:

Smart health monitoring systems typically rely on sensors to gather data about a patient's health. These sensors can include devices that measure vital signs such as heart rate, blood pressure, and oxygen saturation, as well as more specialized sensors that monitor specific health conditions.

**Data storage**:

The data collected by the sensors needs to be stored securely and reliably. This can involve the use of databases or cloud-based storage solutions.

**Data processing**:

The raw data collected by the sensors needs to be processed in order to extract meaningful insights about the patient's health. This can involve the use

**Communication:**

Smart health monitoring systems typically involve the use of wireless communication technologies such as Bluetooth or Wi-Fi to transmit data from the sensors to the data storage and processing components.

**User interface:**

A user interface is needed to allow healthcare professionals and patients to interact with the system. This can involve the use of a web-based interface, a mobile app, or other types of graphical interfaces.

**Security:**

Since health data is sensitive, it is important to ensure that the system is secure and protected from unauthorized access. This can involve the use of encryption, user authentication, and other security measures.

Overall, the architecture of a smart health monitoring system is complex and involves many different components. However, by carefully designing and integrating these components, it is possible to create a system that can provide valuable insights into a patient's health and help healthcare professionals provide better care.

Temperature sensor reedings:

Table 1: Temperature sensor readings

|  |  |  |
| --- | --- | --- |
| User 1: 98.20 F | User 11: 99.30F | User 21: 96.10F |
| User 2: 99.10F | User 12: 98.40F | User 22: 98.00F |
| User 3: 97.90F | User 13: 97.60F | User 23: 98.60F |
| User 4: 98.60F | User 14: 98.80F | User 24: 98.20F |
| User 5: 99.40F | User 15: 99.20F | User 25: 97.50F |
| User 6: 97.80F | User 16: 98.50F | User 26: 98.50F |
| User 7: 98.90F | User 17: 98.90F | User 27: 97.40F |
| User 8: 99.50F | User 18: 99.00F | User 28: 92.20F |
| User 9: 98.10F | User 19: 97.70F | User 29: 91.10F |
| User 10: 98.70F | User 20: 98.30F | User 30: 90.00F |

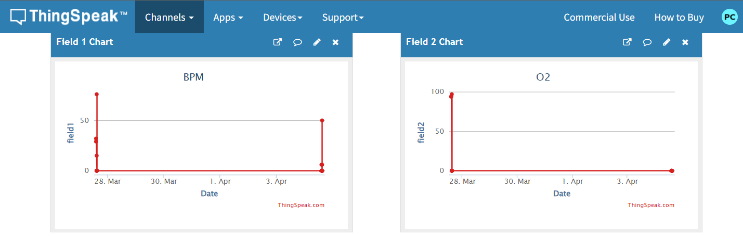
Let's say we have a smart health monitoring system that consists of a pulse sensor or an oxygen sensor. The system is being used by 5 users, and the readings are being recorded and analyzed by the system.

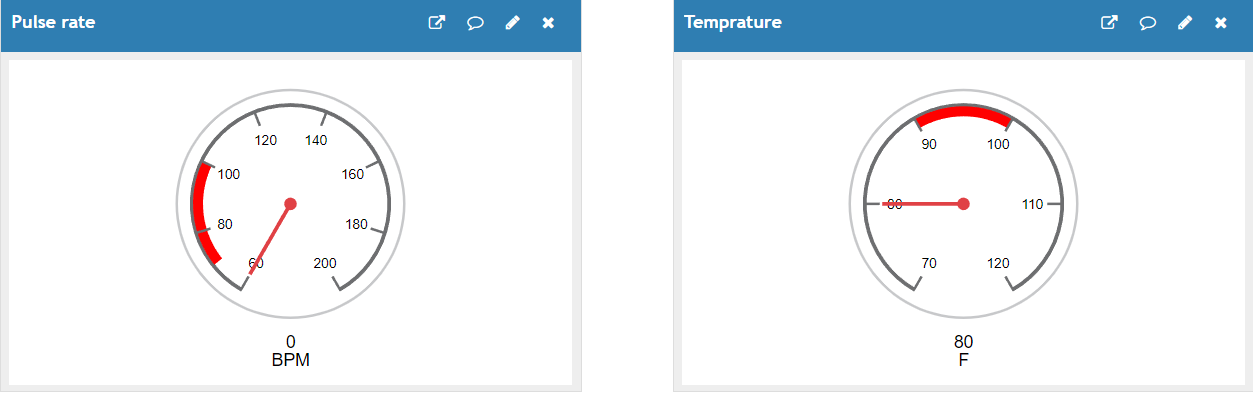
Oxygen sensor readings:

Here are some hypothetical readings for a single user over a period of time:

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Time | Pulse rate | Oxygen level |
| 04/08/2023 | 09:00:00 | 76 bpm | 97 % |
| 04/08/2023 | 10:00:00 | 83 bpm | 94 % |
| 04/08/2023 | 11:00:00 | 89 bpm | 91 % |
| 04/08/2023 | 12:00:00 | 96 bpm | 88 % |
| 04/08/2023 | 13:00:00 | 101 bpm | 85 % |

Table no.2 Oxygen and BPM sensor Readings

1. GRAPHS AND REEDINGS OF SENSORS

Fig no.2 Blood pressure and oxygen level sensor reedings.

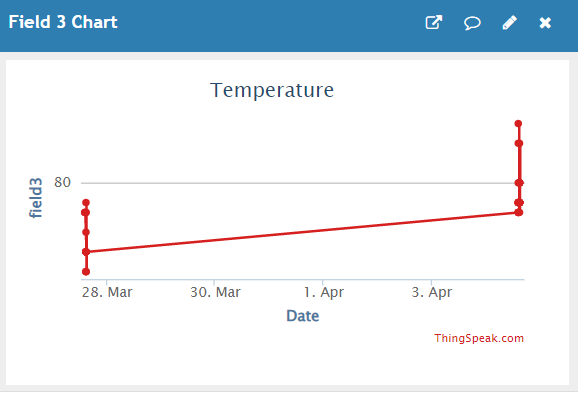
 Fig no.3 Pulse rate and temperature reeding.

Fig no.4 Temperature Graph

* There are some devices information:

1. Temperature sensor
2. Oxygen sensor
3. Wi-Fi Module
4. LCD
5. **Temperature sensor:**

A health temperature sensor is a device that is used to measure body temperature. Health temperature sensors are important for monitoring fever, which is a common symptom of many illnesses. They are also used in medical settings to monitor patients' temperatures during surgical procedures and hospital stays. Some health temperature sensors can be connected to smartphones or other devices via Bluetooth or Wi-Fi, allowing for easy monitoring and tracking of temperature readings over time.

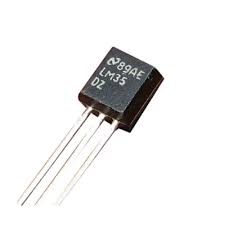
****

Fig no.5 Temperature sensor LM350

1. **Oxygen Sensor:**

Oxygen sensors are also used in healthcare monitoring to measure the oxygen saturation levels in a person's blood. This type of sensor is commonly known as a pulse oximeter. Pulse oximeters work by emitting a light through a person's skin and measuring the amount of light that is absorbed by the blood. The oxygen saturation level is then calculated based on the amount of light that is absorbed. These sensors are commonly used in hospitals and clinics to monitor patients with respiratory problems, such as asthma or COPD, or those undergoing anesthesia during surgery. They can also be used by individuals at home to monitor their oxygen saturation levels and detect any potential health issues. Oxygen sensors used in healthcare monitoring are highly accurate and can provide real-time information about a person's oxygen levels. This information can be used to identify potential health issues early and help prevent complications.

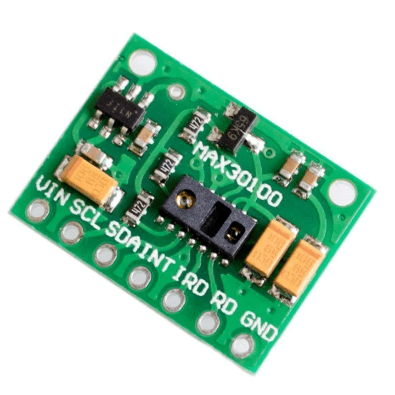
****

Fig no.6 Oxygen sensor MAX 30100

1. **Wi-Fi Module:**

The ESP32 is a low-cost, low-power system-on-chip (SoC) with integrated Wi-Fi and Bluetooth capabilities. It is widely used in Internet of Things (IoT) applications, such as smart home devices, wearables, and industrial automation.

* Here are some key features of the ESP32:

1. Dual-core 32-bit CPU running at up to 240 MHz
2. Wi-Fi 802.11 b/g/n/e/i with support for WPA/WPA2 and WEP encryption
3. Bluetooth 4.2 and Bluetooth Low Energy (BLE) support
4. GPIO pins, UART, I2C, SPI, ADC, DAC interfaces
5. 4 MB flash memory and 520 KB SRAM
6. Support for various operating systems, including Free RTOS and ESP-IDF Overall, the ESP32 is a powerful and flexible platform that enables developers to create a wide range of IoT applications that require wireless connectivity.

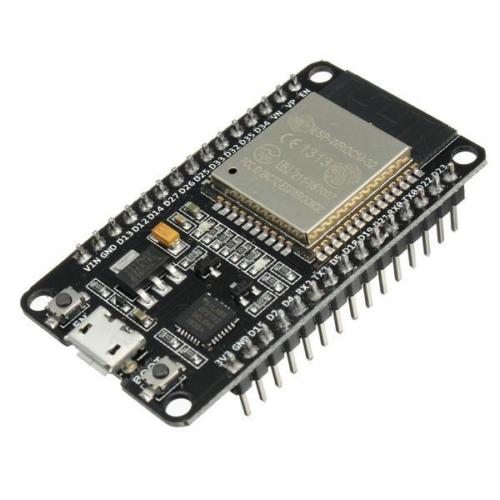
****

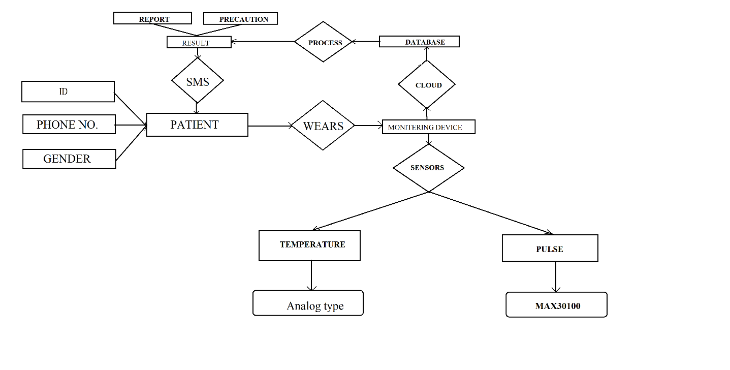
Fig no.7 WI-FI module

1. **LCD:**

LCD stands for Liquid Crystal Display, and it is a type of display technology commonly used in electronic projects. In electronic projects, LCDs can be used to display information, such as text or graphics. When using an LCD in a project, it is important to choose the right type of LCD for your needs and to make sure that your microcontroller or other device can interface with the LCD correctly.

Fig no.8 LCD Display

1. **Flow Chart**

****

1. **Future scope:**

Smart health monitoring projects have tremendous potential for the future. With the increasing prevalence of chronic diseases and the aging population, there is a growing need for remote health monitoring systems that can provide timely and accurate information about a patient's health status.

Smart health monitoring projects have been gaining significant attention in recent years due to their potential to revolutionize the healthcare industry. These projects involve the use of advanced technologies such as sensors, wearables, and artificial intelligence to monitor various health parameters of individuals in real-time.

1. **Challenges:**

Smart health monitoring systems have become increasingly popular due to their ability to monitor various health parameters such as heart rate, blood pressure, glucose level, and more in real-time. However, as with any technology, there are new challenges that arise with the development and implementation of these systems. Here are some of the new challenges that smart health monitoring system projects face:

Data privacy and security: With the increasing amount of personal health data being collected and transmitted by smart health monitoring systems, ensuring the privacy and security of this data has become a critical challenge. Developers need to ensure that appropriate measures are in place to protect the confidentiality, integrity, and availability of health data

1. Accuracy and reliability: Smart health monitoring systems must be accurate and reliable to be useful. Developers must ensure that the systems can detect and report health data accurately and in real-time.
2. Cost-effectiveness: Smart health monitoring systems can be expensive to develop and implement, making cost-effectiveness a critical challenge. Developers must balance the cost of developing and maintaining the system with the potential benefits and impact on the users' health outcomes.
3. User acceptance and adoption: Smart health monitoring systems can be complex and require users to learn new skills and adapt to new technologies. User acceptance and adoption are critical to the success of these systems, and they need to be designed with user needs and preferences in mind.

**Conclusion:**

The use of smart health monitoring systems has the potential to improve health outcomes, and reduce healthcare costs by preventing hospitalizations and enabling earlier treatment. However, it is important to ensure that these systems are designed with user privacy and security in mind, and that they are accessible and affordable to all patients, regardless of socioeconomic status.

##### References

1. Mrunal Fatangare and Hemlata Ohal “An Empirical Survey on Early Health Condition Prediction based on Clinical Parameters” Journal of Engineering and Technology.
2. M. Fatangare, P. U. Nehete, L. R. Mahajan, A. J. Nair and S. Madhurkar, "Identification And Expenditure Tracking Using RFID Based Intelligent Card," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 573-579, doi: 10.1109/ICOEI48184.2020.9143033.
3. M. Fatangare, A. Nimbalkar, G. Chite, A. Narkhede and A. Khilnani, "An Efficient Temperature Monitoring using Raspberry Pi," 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 1-5, doi: 10.1109/ICICT48043.2020.9112376
4. IoT BASED SMART HEALTH MONITORING SYSTEM,[U. B. Mahadevaswamy](https://typeset.io/authors/u-b-mahadevaswamy-yvdcomk6ll), [Durgad Zaibabuktiyar](https://typeset.io/authors/durgad-zaibabuktiyar-1raklf9ou9).
5. IoT-Based Smart Health Unit [Syed Muhammad Waqas Shah](https://typeset.io/authors/syed-muhammad-waqas-shah-1ra1dwstyd), [Maruf Pasha](https://typeset.io/authors/maruf-pasha-ms64zi44ke) 01 Jan 2017**-**[Journal of Software](https://typeset.io/journals/journal-of-software-3ithatqi)**-**Vol. 12
6. IoT SMART CARE SYSTEM BASED ON IoT [Jeon Seong Ho](https://typeset.io/authors/jeon-seong-ho-15p9q7zebv).
7. Development of Smart Health Monitoring System D.S.Dayana1 G.Kalpana2 D. Godwin Immanuel.
8. A Smart Healthcare Monitoring System Using Smartphone Interface [Akshat](https://typeset.io/authors/akshat-11q6roi8pr)1, [Gaurav](https://typeset.io/authors/gaurav-49evd0t8wg)1, [Zahid](https://typeset.io/authors/zahid-2wr0663s01)1, [Bhupendra](https://typeset.io/authors/bhupendra-4lgn3pnmi9)1, [Aditi](https://typeset.io/authors/aditi-21twuf1gi9)1, [Sachin Kumar](https://typeset.io/authors/sachin-kumar-4zp0e9j52r)1, [Maneesha](https://typeset.io/authors/maneesha-4zshnwxn6w)1, [P. K. Pandey](https://typeset.io/authors/p-k-pandey-4oqudsu5wr) .
9. Android based health care monitoring system[Maradugu Anil Kumar](https://typeset.io/authors/maradugu-anil-kumar-4466h78j5a)1, [Y.Ravi Sekhar](https://typeset.io/authors/y-ravi-sekhar-29iflgesd0)1
10. A health remote monitoring application based on wireless body area networks [Elhoussaine Baba](https://typeset.io/authors/elhoussaine-baba-9khfhytjc9)1, [Abdelillah Jilbab](https://typeset.io/authors/abdelillah-jilbab-3c7qiolwak)1, [Ahmed Hammouch](https://typeset.io/authors/ahmed-hammouch-jgh655ahd7)1 01 Apr 2018