

A SECURE IoT BASED MODERN HEALTHCARE SYSTEM USING BODY SENSOR NETWORK

A PROJECT REPORT

Submitted by:

LEKSHMI S.R

LLMC17MCA020

to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



Department of Computer Applications

**LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY
KUTTICAL, THIRUVANANTHAPURAM 695574**

JULY 2020

A SECURE IoT BASED MODERN HEALTHCARE SYSTEM USING BODY SENSOR NETWORK

A PROJECT REPORT

Submitted by:

LEKSHMI S.R

LLMC17MCA020

to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



Department of Computer Applications

**LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY
KUTTICAL, THIRUVANANTHAPURAM 695574**

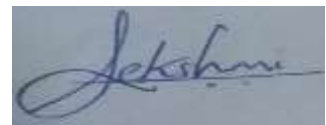
JULY 2020

DECLARATION

I undersigned hereby declare that the project report “**A SECURE IoT BASED MODERN HEALTHCARE SYSTEM USING BODY SENSOR NETWORK**”, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Ms. Anjana J. This submission represents my ideas in my own words and, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University.

Thiruvananthapuram

06/7/2020

A handwritten signature in blue ink, appearing to read 'Lekshmi', is shown within a rectangular box.

LEKSHMI S.R

LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY

(Managed By Archdiocese Of Changanacherry)

(Affiliated To APJ Abdul Kalam Technological University, Kerala)

KUTTICAL, THIRUVANANTHAPURAM-695574

DEPARTMENT OF COMPUTER APPLICATIONS



CERTIFICATE

This is to certify that the project work entitled “A SECURE IoT BASED MODERN HEALTHCARE SYSTEM USING BODY SENSOR NETWORK” is a bonafide record of the work done by Ms LEKSHMI S R, Reg No LLMC17MCA020, student of Department of Computer Applications, Lourdes Matha College of Science & Technology, Kuttichal, Thiruvananthapuram, affiliated to APJ Abdul Kalam Technological University, Kerala during the academic year 2019-2020 from January 2020 to July 2020 in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications from APJ Abdul Kalam Technological University, Kerala.

**Prof. Anjana J
(Internal Supervisor)**

**Prof. Justin G Russel
(Project Co-Ordinator)**

**Prof. Selma Joseph
(Head of the Dept.)**

TABLE OF CONTENT

TITLE	PAGE NO:
ACKNOWLEDGEMENT	I
ABSTRACT	II
Chapter 1 INTRODUCTION	1
1.1 General Background	1
1.2 About the Project	3
1.3 Objectives and Scope	3
1.4 Limitations	4
Chapter 2 LITERATURE SURVEY	5
2.1 Study of Similar Work	5
2.1.1 Existing System	5
2.1.2 Drawbacks of Existing System	5
2.1.3 Survey	6
Chapter 3 OVERALL DESCRIPTION	7
3.1 Proposed System	7
3.2 Features of Proposed System	8
3.3 Functions of Proposed System	9
3.4 Requirement Specification	9
3.5 Feasibility Analysis	10
3.5.1 Technical Feasibility	11
3.5.2 Economical Feasibility	11
3.5.3 Behavioral Feasibility	12
3.5.4 Operational Feasibility	12
Chapter 4 OPERATING ENVIRONMENT	13
4.1 Hardware Requirements	13
4.2 Software Requirements	13
4.3 Tools and Platforms	14
4.3.1 Arduino IDE	14
4.3.2 Embedded C	15
4.3.3 Thingspeak	16
4.3.4 IoT	16
4.3.5 Thingspeak Architecture	18
4.3.6 MQTT Protocols	18
4.3.7 HTTP	19
4.3.8 Thingspeak Features	20
4.3.9 Blynk Server	20
4.3.10 Windows 10	21
4.3.11 Android Language	22
4.3.12 NodeMCU	22
4.3.13 DHT 11 Temperature sensor	25
4.3.14 Heart Rate pulse sensor Amped	27
Chapter 5 DESIGN	30
5.1 System Design	30

5.1.1 Activity Diagram and Flow Chart	31
5.1.1.2 Block Diagram	31
5.1.1.2 Flow Chart	34
5.1.1.3 Activity Diagram	36
5.2 Database Design	37
5.3 Input Design	38
5.4 Output Design	39
5.5 Program Design	42
Chapter 6 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	44
6.1 Functional Requirements	44
6.2 Non-functional Requirements	45
Chapter 7 TESTING	46
7.1 Testing Strategies	46
7.2 Unit Testing	47
7.3 Integration Testing	48
7.4 System Testing	48
7.5 User Acceptance Testing	48
7.6 Data Validation Testing	48
7.7 Output Testing	49
7.8 Testing Result	49
Chapter 8 RESULTS AND DISCUSSION	50
8.1 Screen Shots	51
Chapter 9 CONCLUSION	56
9.1 System Implementation	56
9.2 Conclusion	57
9.3 Future Enhancement	57
REFERENCES/BIBLIOGRAPHY	58
1.Books	58
2.Websites	58
3.Journals	58
APPENDICES	59
1.SCRUM Model	59
2.List of Tables	62
3.List of Figures	62
4.Abbreviations and Notations	63
5.Coding	64

ACKNOWLEDGEMENT

At the outset I like to mention that a project report of this magnitude could not have been possible to make without the support, assistance and guidance of some distinguished personalities.

I am greatly indebted to all those who helped and guided me to make this report without which it would not have been possible for me to do this work.

First of all, I like to express my sincere thanks to **Rev.Dr. Tomy Joseph Padinjareveetil**, Director of Lourdes Matha College of Science and Technology and our principal **Prof. Dr. Mohanlal PP** for granting permission to do this project and giving necessary guidance and assistance.

Prof. Anjana J who was my internal supervisor for the project has always been at her best to help and guide me to the right path which was a great factor for the successful completion of this project.

I like to extend my sincere gratitude to **Prof. Selma Joseph, Head** Of the department of Computer Application. she was always approachable and gave correct advice to go ahead with the project.

Let me also take this opportunity to extend gratitude to our esteemed institution Lourdes Matha College of Science and Technology.

My parents who held my hand throughout in my endeavour to do this project need a special mention.

I also like to thank my friends who helped me abundantly in the successful completion of this project report.

Any omission in acknowledgement may be pardoned as it is not intentional.

ABSTRACT

In the present-day scenario, we find a large number of elderly people staying alone in flats or at isolated places. The present system of a patient physically moving to a healthcare provider has yielded no efficient result. Therefore, an attempt was made to make use of modern developed technologies like IOT and Body Sensor Network (BSN). In this project, therefore it was thought to make use of the already developed technologies like IOT and BSN. However, development of this technology demands careful consideration of security of patient. An attempt is therefore made to cover major security requirements. Subsequently a secure IOT based healthcare system using various body sensors already available in the market is proposed. The proposed system envisages use of various body sensors which can be worn by the patient and can be easily managed by the veteran patients. The output of the sensors used are consolidated by a local processing unit which further forward it to the healthcare server through the mobile network. The project uses NodeMCU (ESP8266) worn by the patient on his wrist and a few body sensors which are commercially available in the market. The front end of the system consists of Arduino IDE and Blynk and Thingspeak an online analytical IoT tool act as back end of the system. The proposed system enables a patient's medical status to be monitored by family member or friend on real time and thus a patient getting timely and quality healthcare. This system when further developed will be definitely a blessing particularly to isolated elders suffering from prolonged life style diseases. About 75 % of elders of today is suffering from one or two of the life style diseases namely diabetes, hypertension etc. The proposed BSN healthcare system come to the timely rescue of these elderly patients who are otherwise not able to make it to hospital due to poor health condition. Depending on a patient's requirement, he is made to wear various body sensors. The proposed body sensor network with the help of IOT make it possible for the designated healthcare team to get the patient health condition instantly. Components used in the proposed system are commercially available and are affordable. BSN healthcare system with the help of IOT is a very valuable instrument for healthcare and is definitely going to revolutionize the healthcare system of the world.

CHAPTER 1

INTRODUCTION

1.1 GENERAL BACKGROUND

Health is a dynamic process which needs to be continuously monitored. Health sectors have been facing various hospital admission problems due to higher rate of patient admission to hospital. To this aim, a system is proposed for human health care. The system provides regular monitoring of patient's metabolic parameters and disease detection using the parametric values obtained. Due to increase in number of sudden deaths caused by chronic heart failure or high blood pressure, it is necessary to provide continuous health monitoring service at home. The prime goal was to develop a reliable patient monitoring system so that the healthcare professionals can monitor the patients, who are either hospitalized or executing their normal daily life activities. Recently, the patient monitoring systems is one of the major advancements because of its improved technology. In our system we are measuring patient's parameters (ECG, temperature, heart rate, pulse, etc) different available sensors. This sensor collected data i.e. biometric information is given to NodeMCU and then it is transferred to server.

Today, the IoT has become one of the most promising communication paradigms, and one in which all the smart objects in our daily life become part of the Internet owing to their communication and computing capabilities. This opportunity brings with its new security challenges for IoT applications. Every smart object (or sensor) in the IoT represents a potential risk in terms of system vulnerability. That is, each intelligent object may become a vulnerable entry point for any malicious attack. Two security issues, i.e. (1) physical protection for smart objects, and (2) how to maintain data confidentiality, integrity and privacy during data collection among smart objects, have thus emerged. Given the novelty and innovative nature of IoT technologies, there seems to be a general expectation for a new and revolutionary security solution tailored specifically to IoT-based objects. This is because traditional security protection mechanisms may not be suitable for smart objects. For example, firewalls containing network management control protocols are able to manage high-level traffic through the Internet. However, this application-level solution is not suitable for endpoint devices in IoT applications because these devices usually possess a specific, defined mission with limited

resources available to accomplish it. Therefore, the refinement of traditional security solutions to fit the specific security requirements of IoT-based smart objects is one of the most promising ways of securing IoT-based application systems.

In the present-day scenario, we find a large no of elderly people staying alone in flats or at isolated places. Recent research indicates that about 80% of aged people above the age of 65 are suffering from at least one life style or chronic disease. This causes many elderly people difficulties in taking care of themselves. Hence it is apparent to provide a decent quality of timely healthcare services to the elderly who are affected. The rapid advancement of IT and communication is making it possible to provide innovative modern healthcare solution instantly. Now IOT enables to extend the concept to the internet and make it more efficient and feasible. IOT allows seamless interactions around different types of devices such as medical sensors, monitoring cameras, home appliances and so on. In healthcare system IOT involves many kinds of cheap sensors that enables aged people to enjoy modern healthcare services anywhere any time and thus improving quality of the life of aged people. To give a brief about the proposed project, it consists of wrist band worn by patient on wrist with temperature sensor and NODE MCU. Temperature sensor DHT 11 is embedded in the wrist band itself. Whereas temperature, oxygen level, and heart rate are measured by sensors clipped on to a fingertip and connected to the NodeMCU by a wire. C++ embedded code will then be uploaded to NodeMCU using android application. The medical data base of the patient is then transmitted through the mobile network to the designated family/member or friend on real time and thus the patient getting timely medical care.

In the future development of technology IoT has a profound influence. In addition, with the development low power embedded technology, sensor technology is widely used. System provides real time health monitoring as well as disease prediction over the internet. It can work base on synthetic as well as real time training data. Accuracy of prediction is good than other learning approaches. System also having a capability to provide the alert when any criticalness 24*7. For Future studies to implement a such systems with parallel processing with high dimensional data using Hadoop or cloud environment.

1.2 ABOUT THE PROJECT

In the modern health care environment, the usage of IoT technologies brings convenience of physicians and patients, since they are applied to various medical areas. The body sensor network (BSN) technology is one of the core technologies of IoT developments in healthcare system, where a patient can be monitored using a collection of tiny-powered and lightweight wireless sensor nodes. However, the development of this new technology in healthcare applications without considering security makes patient privacy vulnerable. At first, we highlight the major security requirements in BSN-based modern healthcare system. Subsequently, we propose a secure IoT-based healthcare system using BSN, called BSN-Care, which can efficiently accomplish those requirements. The body sensor network (BSN) technology is one of the most imperative technologies used in IoT-based modern healthcare system. It is basically a collection of low-power and lightweight wireless sensor nodes that are used to monitor the human body functions and surrounding environment. Since BSN nodes are used to collect sensitive (life-critical) information and may operate in hostile environments, accordingly, they require strict security mechanisms to prevent malicious interaction with the system. It basically employs low power, lightweight sensor nodes which monitor temperature, heart rate, pulse and oxygen level. Since sensors are required to collect sensitive information of the patient, proper care is taken to ensure proper security measures. Blynk server ensures timely passing of patient correct medical status to the concerned family and friends on real time.

1.3 OBJECTIVES AND SCOPE

IOT based modern healthcare system is primarily intended to take care of the health aspects of older people staying alone. The system is envisaged to real time monitoring of the health parameters of dependent patients and provide timely and quality healthcare to them. This will help them abundantly in ageing gracefully with all possible and timely healthcare which is affordable and fully secured. When fully implemented IOT based BSN-CARE will revolutionize the healthcare all over the world.

Objective of the project is to make affordable, fully secure and timely healthcare to the elderly. The system make use of the body sensor networks which with the help of various body sensors measures the health parameters of the patient on real time. NodeMCU worn by patient consolidates the readings and pass it on to Blynk server an android application and then through mobile network. The information is passed on to designated family member/friend on real time by Blynk/cloud server through mobile network who then take appropriate action to save the patient.

1.4 LIMITATIONS

- ❖ Due to lockdown and closing of shops and traffic, greater effort had to be put in to obtain components.
- ❖ Due to lockdown, we lost lot of valuable classes and guidance by the experienced lectures at the institution.
- ❖ On line classes cannot fully compensate for class room instructions and demonstration by experts.
- ❖ Wearable IOT devices have greater demand in the market these days due to availability of affordable internet everywhere.

CHAPTER 2

LITERATURE SURVEY

2.1 STUDY OF SIMILAR WORK

2.1.1 EXISTING SYSTEM

At present a patient has to travel to the nearest hospital for treatment. To see a doctor, he has to wait for his turn. Doctor on examination of basic medical parameters prescribe medicines or ask for further investigation at the lab. After that patient if required will be admitted for inpatient treatment or will be disposed of as an outpatient with necessary medicines. This process is time and effort consuming and is expensive. Moreover, in case of a medical emergency it will take time to reach a doctor/hospital. In many such emergencies a patient by the time he reaches the hospital is dead. There are many such case of “brought dead” in our present scenario. Modern healthcare system with the advent of IOT is able to provide BSN CARE and thus saving many lives. It is boon to old people staying alone and at isolated places.

2.1.2 DRAWBACKS OF EXISTING SYSTEM

- ❖ Existing System is inefficient.
- ❖ It is very tedious and time consuming.
- ❖ Lack of safe and security.
- ❖ Complexity.
- ❖ More human efforts.
- ❖ Implementation Issues.
- ❖ High Expense

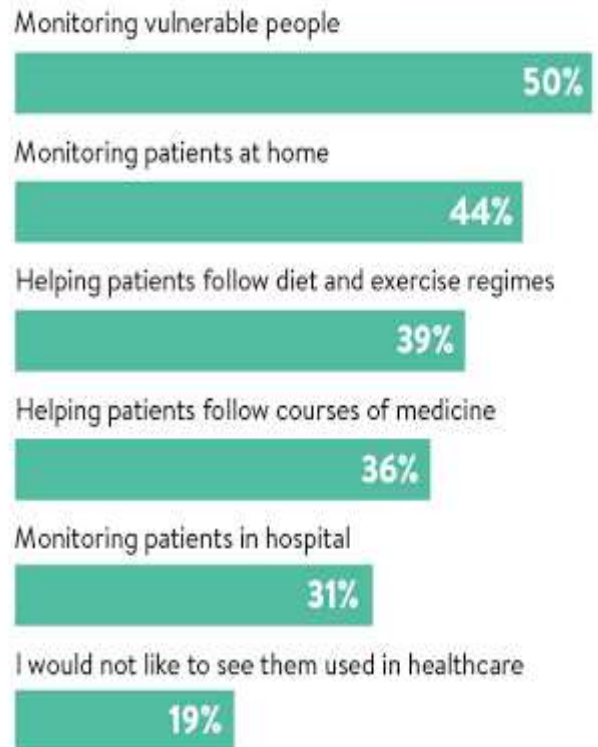
2.1.3 SURVEY

CONSUMER ATTITUDES TOWARDS CONNECTED AND WEARABLE HEALTHCARE DEVICES

WOULD YOU LIKE TO SEE DEVICES USED IN HEALTHCARE?



HOW UK CONSUMERS WANT DEVICES TO BE USED



Source: YouGov/Trustmarque 2015

CHAPTER 3

OVERALL DESCRIPTION

3.1 PROPOSED SYSTEM

❖ IOT BASED BSN-CARE

The proposed system is intended to assist old people staying alone with quality and timely healthcare. The patient health status can be seen by the designated family member/friend on real time which enable them to take timely action.

I have used the following components in my project:

- Wrist band
- NodeMCU
- Temperature sensor
- Heart beat pulse sensor

- ❖ **Wrist Band:** Is worn by patient on the wrist. NodeMCU and temperature sensors are embedded on to it.
- ❖ **NodeMCU:** Is a nano chip on the wrist band to which other sensors are also connected. Sensors feed their output to the nano chip which convert them to digital signal and is fed to the cloud server application. ESP 8266 NodeMCU is used in the project.
- ❖ **Temperature sensor:** I have used DHT 11 temperature sensor. It measures the temperature of the patient on any given time and feed it in to the NodeMCU. In case of up normal temperature, it gives an alarm to the patient to be careful.
- ❖ **Heart Rate Pulse Sensor Amped:** This sensor is clipped on to a finger of the patient and connected to the NodeMCU with the help of wire. This sensor measured the heart beat pulse rate and is fed to the NodeMCU as electrical signals (digital) through the connected wire.
- ❖ **Open Source Android App (Blynk):-** Blynk is an open source android app which is designed and developed in order to control the hardware via internet of things (IOT). This digitally displays sensor data; it can accumulate and visualize the data. Plus, it can also do other parameters such as:

- ❖ **Blynk App:** This app gives us to create amazing interfaces for a project using multiple widgets which is an in-build app.
- ❖ **Blynk server:** It acts as an interface between the smartphone and hardware which is responsible for the communication. We can also use Blynk cloud or compile our private Blynk server. It's an open source that can control any number of devices plus can also be launched on Raspberry pi.
- ❖ **Blynk Libraries:** For all the standard hardware platforms, supports communication with the sensor and the complete progression of incoming and outgoing instructions.
- ❖ **Embedded C:** It is mainly used for the purpose of real time response. RTS (real time response) is designed and developed as a device which corrects based on the time of response. The advanced version of RTS follows the concept of responding with delay is fine. For instance, this includes railway platform which displays schedule system.
- ❖ **Arduino IDE:** Arduino IDE where IDE (Integrated Development Environment). This is basically an open source app where one can code, compile, and upload a file in an Arduino device. In fact, any Arduino modules are adapted by this software, which has in build features by default. It is available for operating systems for instance MAC, Windows, Linux, and runs on the java software. A range of Arduino modules, consist of Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro etc. Every module contains a microcontroller on the board which is in build by default.
- ❖ **Thingspeak:** Thingspeak is an IoT analytics platform service that allows you to aggregate, visualize and analyse live data streams in the cloud. Thingspeak provides instant visualizations of data posted by your devices to Thingspeak. With the ability to execute MATLAB code in Thingspeak you can perform online analysis and processing of the data as it comes in. Thingspeak is often used for prototyping and proof of concept IoT systems that require analytics.

3.2 FEATURES OF PROPOSED SYSTEM

- ❖ Cost Effective
- ❖ Efficient
- ❖ Can Implement Easily
- ❖ Safe and Secure
- ❖ Easy to maintain
- ❖ Can easily add advanced technologies

- ❖ User Friendly
- ❖ Light Weight
- ❖ Ease of Use

3.3 FUNCTIONS OF PROPOSED SYSTEM

- ❖ The proposed IOT based BSN CARE health system is cost effective as its components are available at affordable prices.
- ❖ The system is very efficient as the information on human health is obtained on real time at the health care provider.
- ❖ A solar power system is provided for powering all the sensors.
- ❖ An ESP8266 WIFI module is used for connecting to the internet
- ❖ In this project, a system for 24*7 human health monitoring is designed and implemented.
- ❖ The system can be implemented easily due to availability of wearable IOT devices and mobile network.
- ❖ It is safe and secure as adequate security measures are incorporated.
- ❖ It is easy to maintain, user friendly and light weight.
- ❖ Analysis and prediction of chronic disorders in primary stage through the data mining techniques for better decision by doctors.
- ❖ Future expansion is possible as advanced technologies can easily be added.

3.4 REQUIREMENT SPECIFICATION

System analyst tasks to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. These can be done through questionnaires, details investigation, observation, collection of samples etc. As the details are collected, the analyst studies the requirements data to identify the features the new system should have, including both the information the system produces and operational features such as processing controls, response times, and input output methods. Requirement specification simply means, “Figuring out what to make before you make it”. It determines what people need before you start developing a product for them. Requirement definition is the activity of translating the information gathered in to a document that defines a set of requirements. These should accurately reflect what consumer wants. It is an abstract

description of the services that the system should provide and the constraints under the system must operate. This document must be written for that the end user and the stake holder can understand it.

The notations used for requirements definition should be based on natural languages, forms and simple intuitive diagrams. The requirements fall into two categories: functional requirements and non-functional requirements.

The requirements of specification of the proposed system are as follows:

- ❖ NODE MCU (ESP8266 1.0 wifi mod)
- ❖ Embedded C
- ❖ Open Source Android App (Blynk)
- ❖ Thingspeak
- ❖ Temperature sensor
- ❖ Heart beat pulse sensor Amped
- ❖ Wrist band
- ❖ Arduino IDE

3.5 FEASIBILITY ANALYSIS

The initial investigation points to be question whether the project is feasible. The feasibility study concerns with the considerations made to verify whether the system fit to be developed in all terms. Once the idea to develop software is put forward, the question that rises first will pertain to be the feasibility aspects. Feasibility study is a test of proposed system regarding its efficiency, impact on the organization, ability to meet the need of users and effective use of resources.

Thus, when a new project is proposed, it normally goes through a feasibility study before it is approved for development. A feasibility study is conducted to select the best system that meets the system performance requirements. This entitles an identification description, an evaluation of candidate system and the selection of the best system for the job.

During system analysis, a feasibility study of the proposed system was carried out to see whether it was beneficial to the organization.

Three key considerations that are involved in the feasibility study. They are,

- ❖ Technical Feasibility
- ❖ Economic Feasibility
- ❖ Behaviour Feasibility
- ❖ Operational Feasibility

3.5.1 TECHNICAL FEASIBILITY

Technical Feasibility centres on the existing computer system hardware, software, etc. and to some extent how it can support the proposed addition. This involves financial considerations to accommodate technical enhancements. Technical support is also a reason for the success of the project. The techniques needed for the system should be available and it must be reasonable to use. Technical Feasibility is mainly concerned with the study of function, performance, and constraints that may affect the ability to achieve the system. By conducting an efficient technical feasibility, we need to ensure that the project works to solve the existing problem area.

Since the project is designed using Embedded C as programming language. It is very efficient and user friendly. Here we are using NODE MCU to feed the program and the readings from various sensors are send to the mobile phones using Bluetooth data monitoring app which easy to use and maintain.

3.5.2 ECONOMIC FEASIBILITY

The role of interface design is to reconcile the differences that prevail among the software engineer's design model, the designed system meets the end user requirement with economical way at minimal cost within the affordable price by encouraging more of proposed system. Economic feasibility is concerned with comparing the development cost with the income/benefit derived from the developed system. In this we need to derive how this project will help the management to take effective decisions.

Economic Feasibility is mainly concerned with the cost incurred in the implementation of the project. The hardware consists of Temperature sensors, humidity sensors, Heart rate pulse sensor amped, a led, buzzer. The overall cost for making this prototype is considered as Rs 1900/- Thingspeak is a free software so no cost is needed to buy the

backend. My system is economically feasible because the project completed in few months. so, less resources are used.

The system once developed must be used efficiently. Otherwise there is no meaning for developing the system. For this a careful study of the existing system and its drawbacks are needed. The user should be able to distinguish the existing one and Proposed one, so that one must be able to appreciate the characteristics of the proposed System, the manual one is not highly reliable and also is considerably fast. The proposed system is efficient, reliable and also quickly responding.

3.5.3 BEHAVIOURL FEASIBILITY

Proposed projects are beneficial only if they can be changed in to information system that will meet operation requirement of the organization. People are inherently resistant to change and computers have been known to facilitate changes. An estimate should be made of how strong reaction the user staff is likely to have towards the development of a computerized system. The behavioral feasibility depends upon whether the system performed in the expected way or not. Behavioral Feasibility study is a test of system proposal according to it workability, impact on organization, ability to meet user's need and effective use of resources. However, a feasibility study provides a useful starting point for full analysis. Our system is behaviorally feasible because of the effective use of the resources and also the system satisfies user needs and is user friendly. The different sensors used will increase the systems performance and makes is more effective. The user will be able to identify the malicious websites quickly and effectively.

3.5.4 OPERATIONAL FEASIBILITY

There is not much difficulty in implementing the system. The proposed system is effective, user friendly and functionally efficient. The user of the system must be aware of the internal working of the system so that the user will not face any problems running the system. In our system we are using NODE MCU microcontroller and various sensors. We can extend or add any features to the system easily

The sensors will give accurate value according to the programs that are fed in to the micro controller. The user can easily use the system and app. There is no need to worry about the internal procedures of the system.

CHAPTER 4

OPERATING ENVIRONMENT

4.1 HARDWARE REQUIREMENTS

Processor	:	Intel Pentium5, or above
RAM	:	8GB or above
Hard Disk	:	256GB or above
Display Size	:	15” Colour Monitor or above
Screen Resolution	:	1920*1080(Full HD)
Keyboard	:	PC/AT Enhanced Type
Mouse	:	Logitech PS/2 Port Mouse
Micro Controller	:	NODE MCU
Temperature Sensor	:	DHT 11 sensor
Pulse sensor	:	Heart Rate Pulse Sensor Amped

4.2 SOFTWARE REQUIREMENTS

Operating System	:	Windows 10
Front End	:	Android application (Blynk), Arduino IDE
Back End	:	Thingspeak
Language	:	C++,Embedded C

4.3 TOOLS AND PLATFORMS

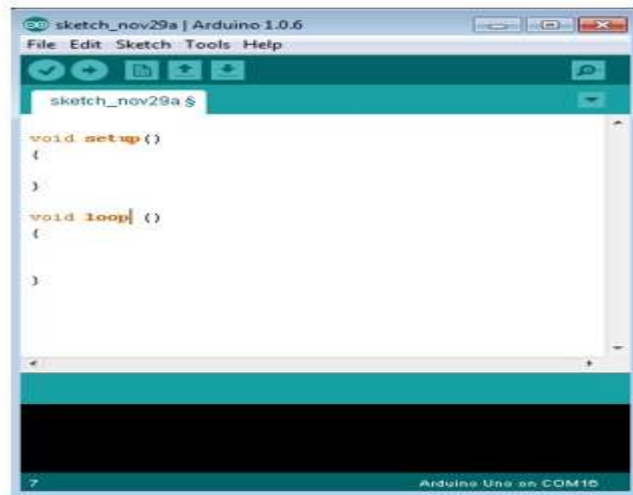
4.3.1 Arduino IDE

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

The key features are –

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board.
- You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.



4.3.2 Embedded C

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems. Embedded C programming typically requires nonstandard extensions to the C language in order to support enhanced microprocessor features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. In 2008, the C Standards Committee extended the C language to address such capabilities by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as fixed-point arithmetic, named address spaces and basic I/O hardware addressing. Embedded C uses most of the syntax and semantics of standard C, e.g., `main()` function, variable definition, datatype declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array). All these devices have one thing in common: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to define how the device actually works. Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt

Handling, and I/O Ports.

An embedded system has three components –

- It has hardware.
- It has application software.
- It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small-scale embedded system may not have RTOS.

So, we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

Advantages

- ❖ Easily Customizable
- ❖ Low power consumption
- ❖ Low cost
- ❖ Enhanced performance

Disadvantages

- ❖ High development effort.
- ❖ Larger time to market.

4.3.3 Thingspeak

Thingspeak is an IoT analytics platform service that allows you to aggregate, visualize and analyse live data streams in the cloud. Thingspeak provides instant visualizations of data posted by your devices to Thingspeak. With the ability to execute MATLAB code in Thingspeak you can perform online analysis and processing of the data as it comes in. Thingspeak is often used for prototyping and proof of concept IoT systems that require analytics.

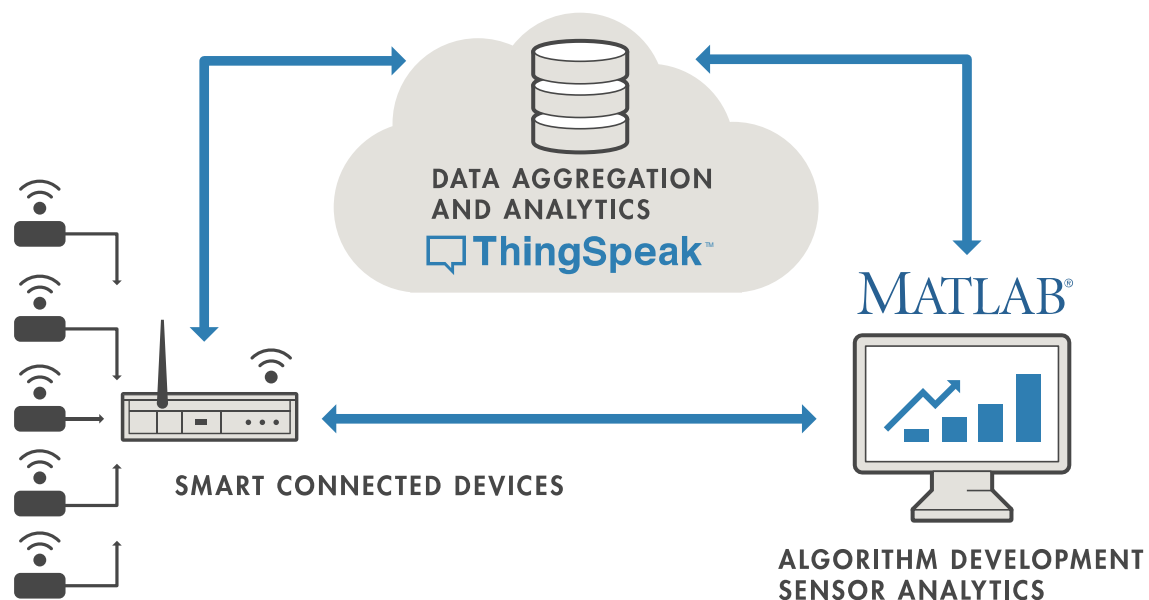
4.3.4 What is IOT?

Internet of Things (IoT) describes an emerging trend where a large number of embedded

devices (things) are connected to the Internet. These connected devices communicate with people and other things and often provide sensor data to cloud storage and cloud computing resources where the data is processed and analyzed to gain important insights. Cheap cloud computing power and increased device connectivity is enabling this trend.

IoT solutions are built for many vertical applications such as environmental monitoring and control, health monitoring, vehicle fleet monitoring, industrial monitoring and control, and home automation.

At a high level, many IoT systems can be described using the diagram below:



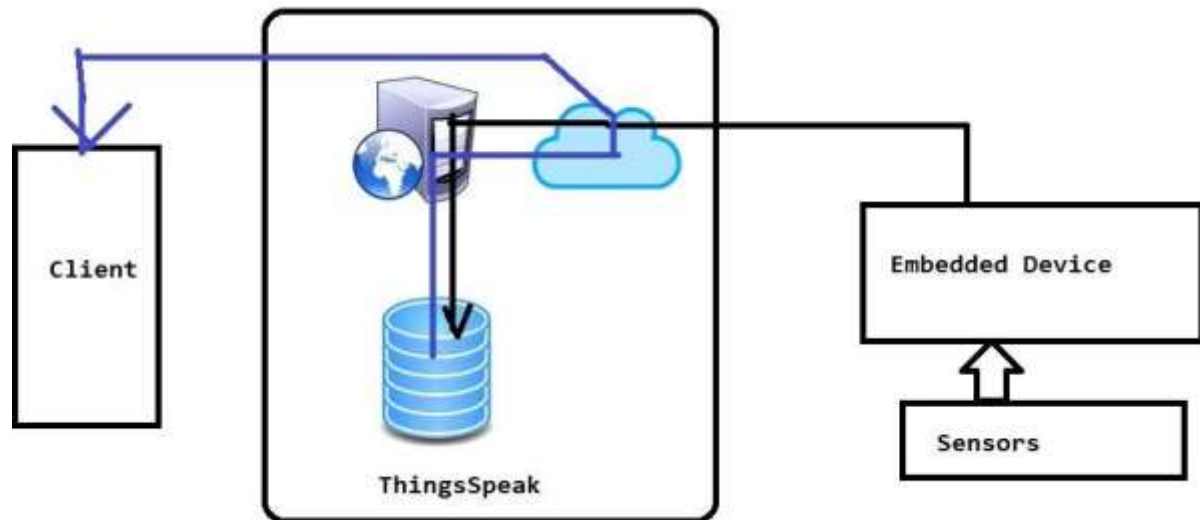
On the left, we have the smart devices (the “things” in IoT) that live at the edge of the network. These devices collect data and include things like wearable devices, wireless temperatures sensors, heart rate monitors, and hydraulic pressure sensors, and machines on the factory floor. In the middle, we have the cloud where data from many sources is aggregated and analyzed in real time, often by an IoT analytics platform designed for this purpose.

The right side of the diagram depicts the algorithm development associated with the IoT application. Here an engineer or data scientist tries to gain insight into the collected data by performing historical analysis on the data. In this case, the data is pulled from the IoT platform into a desktop software environment to enable the engineer or scientist to prototype algorithms

that may eventually execute in the cloud or on the smart device itself.

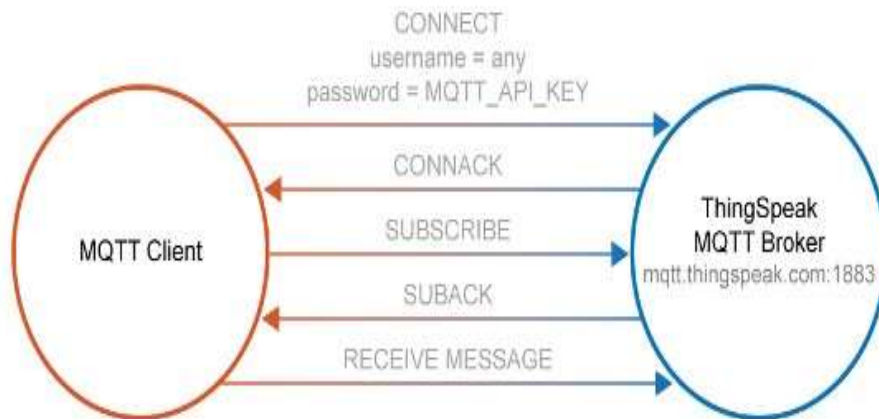
An IoT system includes all these elements. Thingspeak fits in the cloud part of the diagram and provides a platform to quickly collect and analyse data from internet connected sensors.

4.3.5 THINGSPEAK ARCHITECTURE



4.3.6 MQTT PROTOCOLS

MQTT is a common protocol used in IoT systems to connect low-level devices and sensors. MQTT is used to pass short messages to and from a broker. Thingspeak has recently added an MQTT broker so devices can send messages to Thingspeak. A message might contain the current temperature in an office collected by a sensor. Thingspeak takes the message and stores its content in a Thingspeak channel. Once the data is in a channel, you can easily visualize and analyze the data with MATLAB code.



1. Subscribe to a channel feed

`channels/<channelID>/subscribe/<format>/<api_key>`

2. Subscribe to a private channel feed

`channels/<channelID>/subscribe/fields/field<fieldNumber>/<apiKey>`

3. Subscribe to all fields of a channel

`channels/<channelID>/subscribe/fields/+/<apiKey>`

<api_key> is not required to subscribe to public channels

4.3.7 HTTP (Hyper Text Transfer Protocol)

The Thing HTTP App allow a microcontroller or low-level device to connect to any web service using HTTP over internet. We can create an HTTP object using Thing HTTP app and then control the object using simple API commands.

Thingspeak supports GET, POST, PUT, DELETE method. We have a device interface with many web services and API without having to implementing on the device level.



4.3.8 THINGSPEAK KEY FEATURES

Thingspeak allows you to aggregate, visualize and analyse live data streams in the cloud. Some of the key capabilities of Thingspeak include the ability to:

- ❖ Easily configure devices to send data to Thingspeak using popular IoT protocols.
- ❖ Visualize your sensor data in real-time.
- ❖ Aggregate data on-demand from third-party sources.
- ❖ Use the power of MATLAB to make sense of your IoT data.
- ❖ Run your IoT analytics automatically based on schedules or events.
- ❖ Prototype and build IoT systems without setting up servers or developing web software.
- ❖ Automatically act on your data and communicate using third-party services like Twilio or Twitter.

4.3.9 BLYNK SERVER

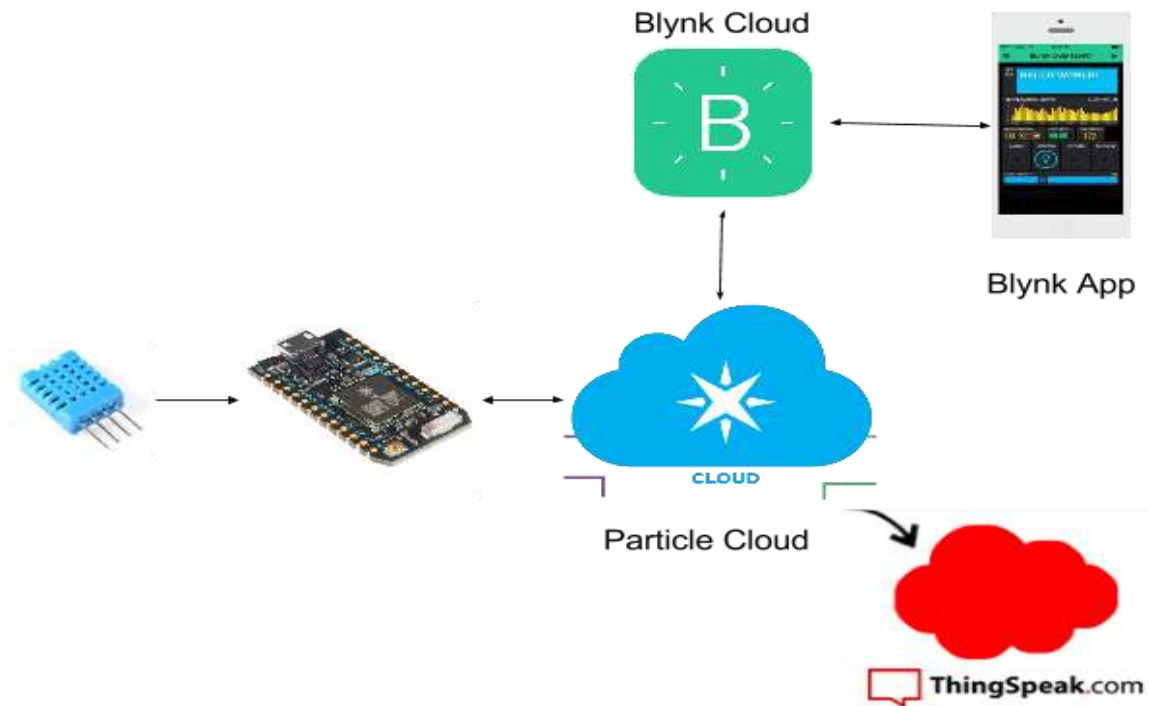
Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. Blynk is a new platform that allows you to quickly build interfaces for controlling and monitoring your hardware projects from your iOS and Android device. After downloading the Blynk app, you can create a project dashboard and arrange buttons, sliders, graphs, and other widgets onto the screen.

Open Source Android App (Blynk)-: Blynk is an open source android app which is designed and developed in order to control the hardware via internet of things (IOT). This digitally displays sensor data; it can accumulate and visualize the data. Plus, it can also do other parameters such as:

Blynk App: This app gives us to create amazing interfaces for a project using multiple widgets which is an in-build app.

Blynk server: It acts as an interface between the smartphone and hardware which is responsible for the communication. We can also use blynk cloud or compile our private blynk server. It's an open source that can control any number of devices plus can also be launched on Raspberry pi.

Blynk Libraries: For all the standard hardware platforms, supports communication with the sensor and the complete progression of incoming and outgoing instructions.



4.3.10 WINDOWS 10

Windows 10 is a major version of the Microsoft Windows operating system that was released on July 29, 2015. It is built on the Windows NT kernel and follows Windows 8. Part of the reason Microsoft decided to name the 2015 release "Windows 10" (and skipped "Windows 9") is because the operating system is designed to be a new direction for Microsoft. One of the primary aims of Windows 10 is to unify the Windows experience across multiple devices, such as desktop computers, tablets, and smartphones. As part of this effort, Microsoft developed Windows 10 Mobile alongside Windows 10 to replace Windows Phone – Microsoft's previous mobile OS. Windows 10 also integrates other Microsoft services, such as Xbox Live and the Cortana voice recognition assistant.

While Windows 10 includes many new features, it also brings back the Start Menu, which was dropped in Windows 8. The new and improved Start Menu provides quick access to settings, folders, and programs and also includes tiles from the Windows 8 interface. The bottom of the Windows 10 Start Menu includes a search bar that allows you to search both your local PC and the web.

4.3.11 ANDROID LANGUAGE

Android is an open source and Linux-based Operating System for mobile devices such as smart phones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need to develop only for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007, whereas the first commercial version, Android 1.0, was released in September 2008. On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance. The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

4.3.12 NODE MCU

NodeMCU is a nano chip on the wrist band to which other sensors are also connected sensors feed their output to the nano chip which convert them to digital signal and is fed to the cloud server application. ESP 8266 NodeMCU is used in the project.

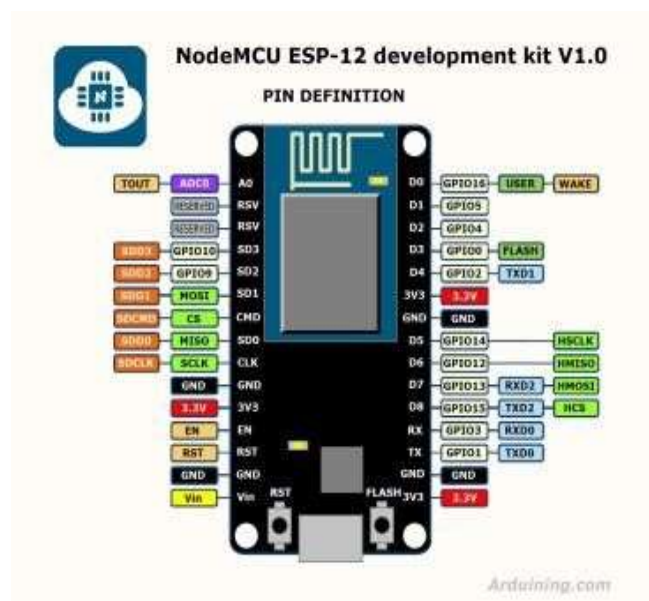
NodeMCU is an open source development board and firmware based in the widely used ESP8266 -12E WiFi module. It allows you to program the ESP8266 WiFi module with the simple and powerful LUA programming language or Arduino IDE.

With just a few lines of code you can establish a WiFi connection and define input/output pins according to your needs exactly like Arduino, turning your ESP8266 into a web server and a lot more. It is the WiFi equivalent of ethernet module. Now you have internet of things (iot) real tool.

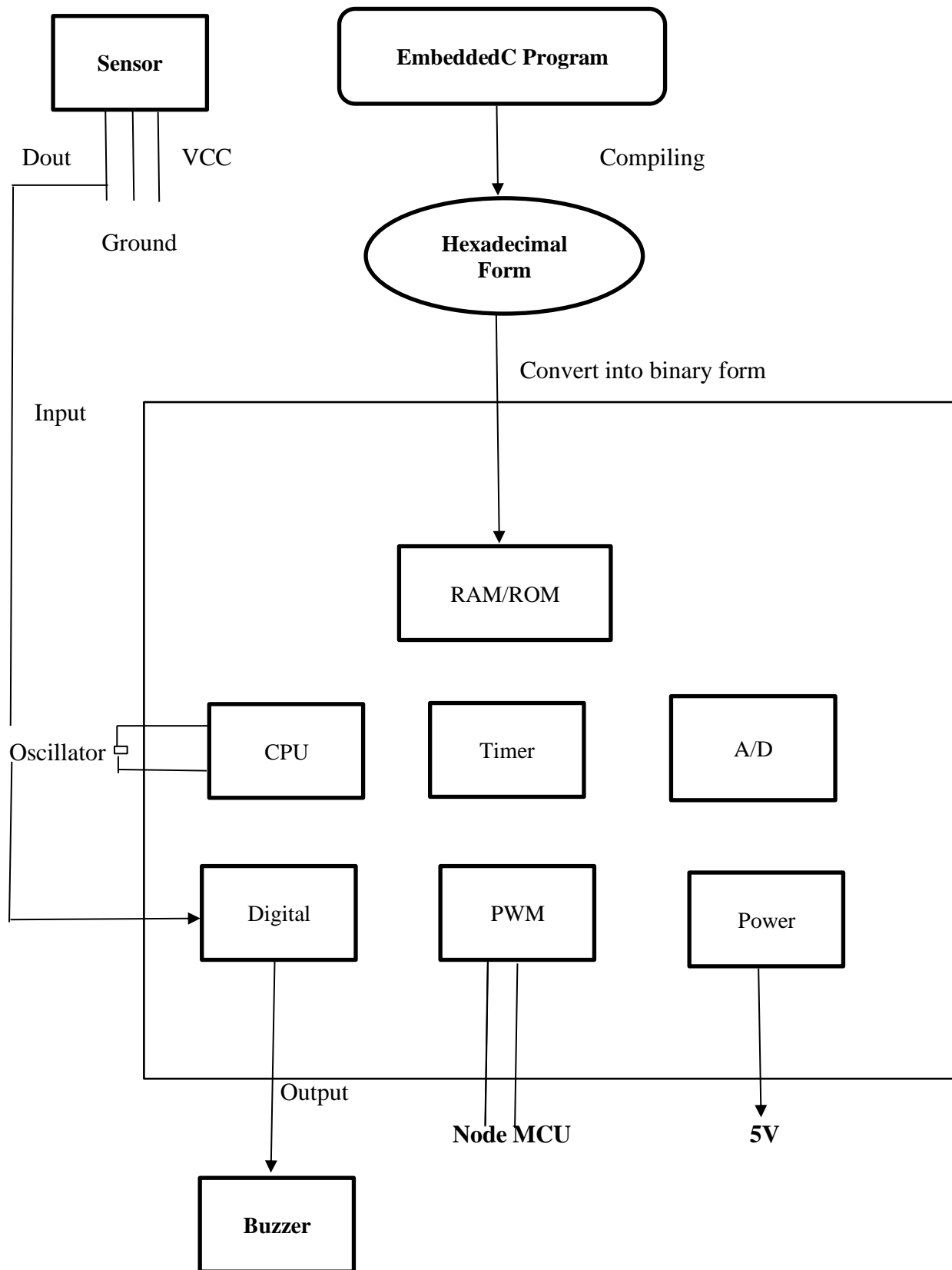
With its USB-TTL, the NodeMCU Dev board supports directly flashing from USB port. It combines features of WIFI access point and station + microcontroller. These features make the NodeMCU extremely powerful tool for Wifi networking. It can be used as access point and/or station, host a webserver or connect to internet to fetch or upload data.

Features

- ❖ Finally, programmable WiFi module.
- ❖ Arduino-like (software defined) hardware IO.
- ❖ Can be programmed with the simple and powerful Lua programming language or Arduino IDE.
- ❖ USB-TTL included, plug & play.
- ❖ 10 GPIOs D0-D10, PWM functionality, IIC and SPI communication, 1-Wire and ADC A0 etc. all in one board.
- ❖ Wifi networking (can be used as access point and/or station, host a web server), connect to internet to fetch or upload data.
- ❖ Event-driven API for network applications.
- ❖ PCB antenna.



System Architecture



4.3.13 DHT11 TEMPERATURE SENSOR

The DHT11 is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc... to measure humidity and temperature instantaneously. DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

Working Principle of DHT11 Sensor

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz.i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controllers.

Applications

This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to predict weather conditions. The humidity sensor is used as a preventive measure in homes

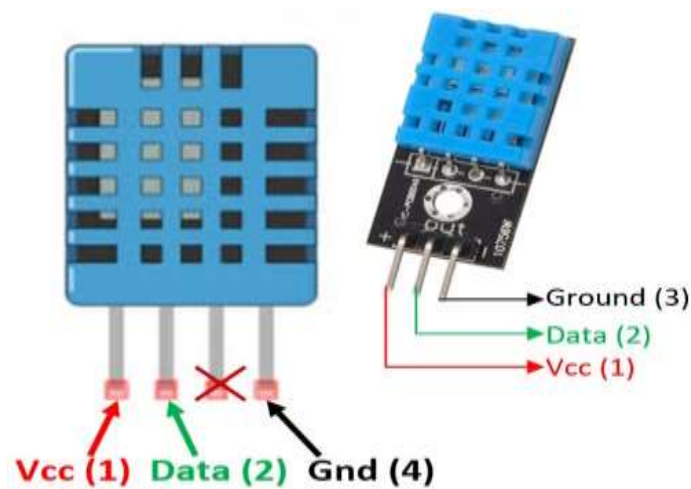
where people are affected by humidity. Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure.

Pin Identification and Configuration:

No:	Pin Name	Description
For DHT11 Sensor		
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	NC	No Connection and hence not used
4	Ground	Connected to the ground of the circuit
For DHT11 Sensor module		
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

DHT11 Specifications

- ❖ Operating Voltage: 3.5V to 5.5V
- ❖ Operating current: 0.3mA (measuring) 60uA (standby)
- ❖ Output: Serial data
- ❖ Temperature Range: 0°C to 50°C
- ❖ Humidity Range: 20% to 90%
- ❖ Resolution: Temperature and Humidity both are 16-bit
- ❖ Accuracy: $\pm 1^{\circ}\text{C}$ and $\pm 1\%$

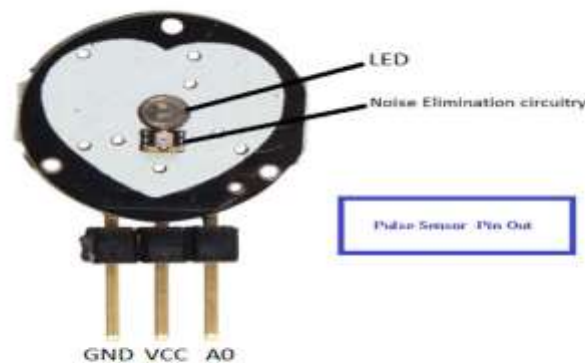


4.3.14 HEART RATE PULSE SENSOR AMPED

Heart rate pulse sensor amp is a such type of sensor which is mainly used for sensing heartbeat rate. Normally it is very difficult task to measure the exact heartbeat rate, but this have become so much easy with the help of this pulse sensor amp. If we talk about heartbeat, then heart beat is a periodic signal that is produced by any software or hardware system for giving intimation to normal of working of any system. For measuring this periodic intimation signal, so many sensors have been using currently in market but here we shell only talk about pulse sensor amp. This is basically plug and play heartbeat sensor and have been using by makers, athletes, game developers and students in their hardware projects. It is easily available in market or online shop.

Pin Configuration of Heart Rate Pulse Sensor

Every heart rate sensor consists of three pins first one is ground pin which is used for supplying ground to this sensor and it is connected to source ground pin. Second one is VCC pin which is used for power on this heart rate sensor and it is connected to source VCC pin. This sensor is powered on at almost 3.3V to 5V dc voltages. Similarly, the last one is A0 pin which is an analogue pin and it is used for receiving analogue signal.



Working Principle of Heart Rate Pulse Sensor

The working principle of this heartbeat rate sensor is very simple. If we talk about heartbeat rate, then heartbeat rate is the ratio of time between two consecutive heartbeats. Similarly, when the human blood is circulated in human body then this blood is squeezed in capillary tissues. As a result, the volume of capillary tissues is increased but this volume is decreased after each heartbeat. This change in volume of capillary tissues, effects on the LED light of heart rate pulse sensor, which transmits light after each heartbeat.

This change in light is very small but this can be measured by connecting any controller with this pulse sensor. Means, the LED light which have every pulse sensor helps for measuring pulse rate. The working of this sensor could be checked by placing human finger in front of this pulse sensor. When finger is placed in front of this pulse sensor then the reflection of LED light is changed based on the volume of blood change inside capillary vessels. Means during heartbeat the volume of blood in capillary vessels will be high and then will be low after each

heartbeat. So, by changing this volume the LED light is changed. This change in of LED light measures the heartbeat rate of finger.



Applications

- ❖ Wireless supervision of people during hazardous operations.
- ❖ In an overcrowded emergency department.
- ❖ Chronic surveillance of abnormal heart failure.
- ❖ In cardio-vascular disease for monitoring the hyper tension.

Advantages

- ❖ Continuous monitoring.
- ❖ Reducing hospitalization fee.
- ❖ It directly calibrated the temperature in Celsius (centigrade).
- ❖ It basically detects the present expression of patient.
- ❖ In this the heart beat sensor gives the digital output to the microcontroller & output from microcontroller gives the rate in beats per minute (BPM).

CHAPTER 5

DESIGN

5.1 SYSTEM DESIGN

System design is a reduction of an entire system by studying the various operations performed and their relationships within the system and the requirements of its success. One aspect of design is defining the boundaries of the system and determining whether or not the candidate system should consider other related system.

System can be defined, as an orderly grouping of interdependent components can be simple or complex. The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing the candidate system. It also includes the construction of programs and program testing.

The first step in the system design is to determine how the output is to be produced and in what format. Samples of the output and the inputs are also presented. In the second step, input data and master files are to be designed to meet requirement of the proposed output. The processing phase's system's objectives and complete documentation.

System design has two phases:

- Logical
- Physical

The logical design reviews the present physical system, prepares the input and output and also prepares a logical design walk- through. We have to deal with how to take entries required and whether and how to process the user data. Also, we have to deal with how to present the data in an informative and appealing format. This design also involves the methodology to store, modify and retrieve data from the data base as per the requirement.

Physical design maps out the details of the physical system, plans the system implementation, devices a test and implementation plan and new hardware and software. We have to decide how and where to store the input data and how to process it so as to present it to the user in an easy, informative and attractive manner.

5.1.1 Activity Diagram and Flow Chart

i. Block Diagram

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams. Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.

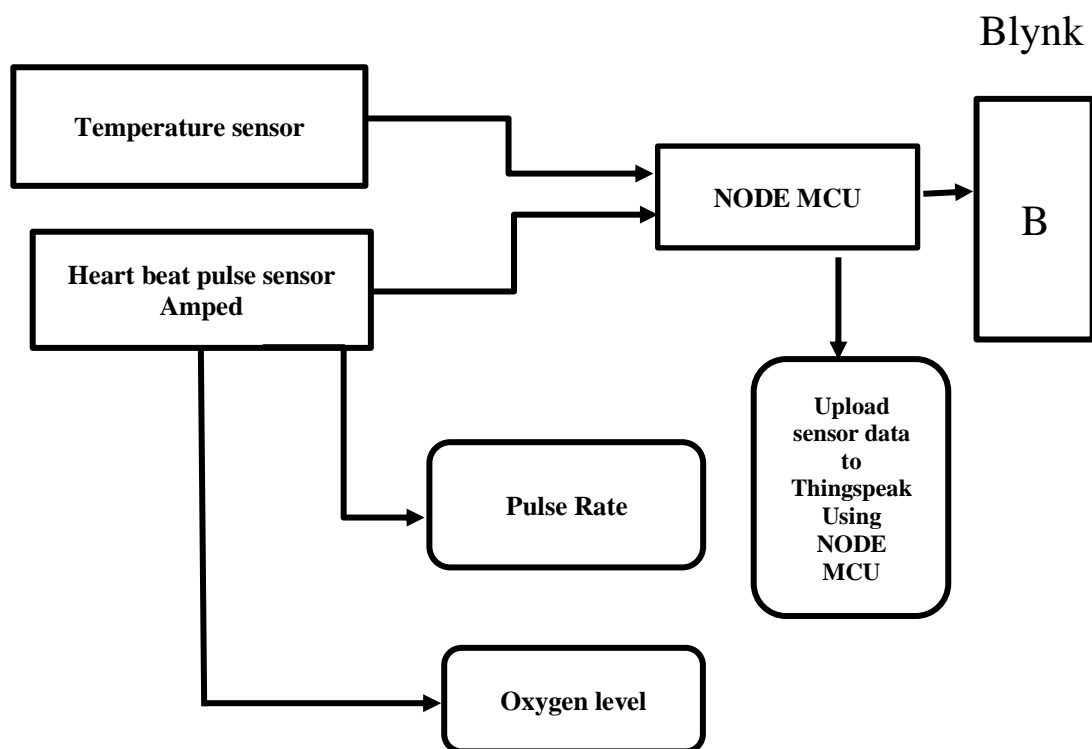


Figure 5.1: Block Diagram

Hardware Module

In this module, we use a microcontroller Node MCU, Heart Rate Pulse Sensor Amped, DHT11 Sensor as hardware module.

NODE MCU stands for Node Microcontroller Unit is an Open source software and hardware development environment.

DHT11 is used for Temperature monitoring the value became higher than the specified degree it automatically takes corresponding action.

Power failure detection can be calculated from the circuit using Thingspeak it fails message send through Blynk and Email.

Heart Rate Pulse Sensor Amped: This sensor is clipped on to a finger of the patient and connected to the NodeMCU with the help of wire. This sensor measured the heart beat pulse rate and is fed to the NodeMCU as electrical signals (digital) through the connected wire.

Software Module

Thingspeak which is an open source IOT and API to store sensor value, perform action and display output in graphical form at web level.

Blynk which is a new platform for making Quick interface for our hardware project

BLOCK DIAGRAM OF PROPOSED BSN-CARE

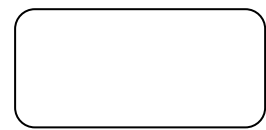
Sensors output of the sensors worn by the patient namely Temperature sensor, Heart beat pulse sensor, oxygen level and pulse rate sensor are fed to the NODE MCU. These sensor data are uploaded to Thingspeak using NODE MCU. Blynk server then forward the coded data to family or friend and to healthcare server when integrated. The data so received is analyzed and interpreted by the healthcare server when fully developed who takes timely action. In this case the family member or friend who is connected to sensor output should be capable of taking necessary action on receipt of sensor data or alert notification on the mobile network or Email.

ii. Flowchart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Components of Flowchart

Start/End



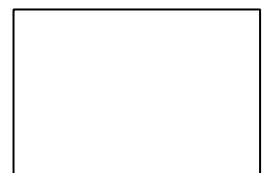
Connector



Input/output



Process



Decision

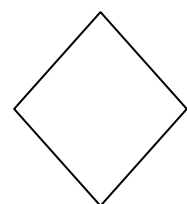


Figure 5.1.2 Components of flow chart

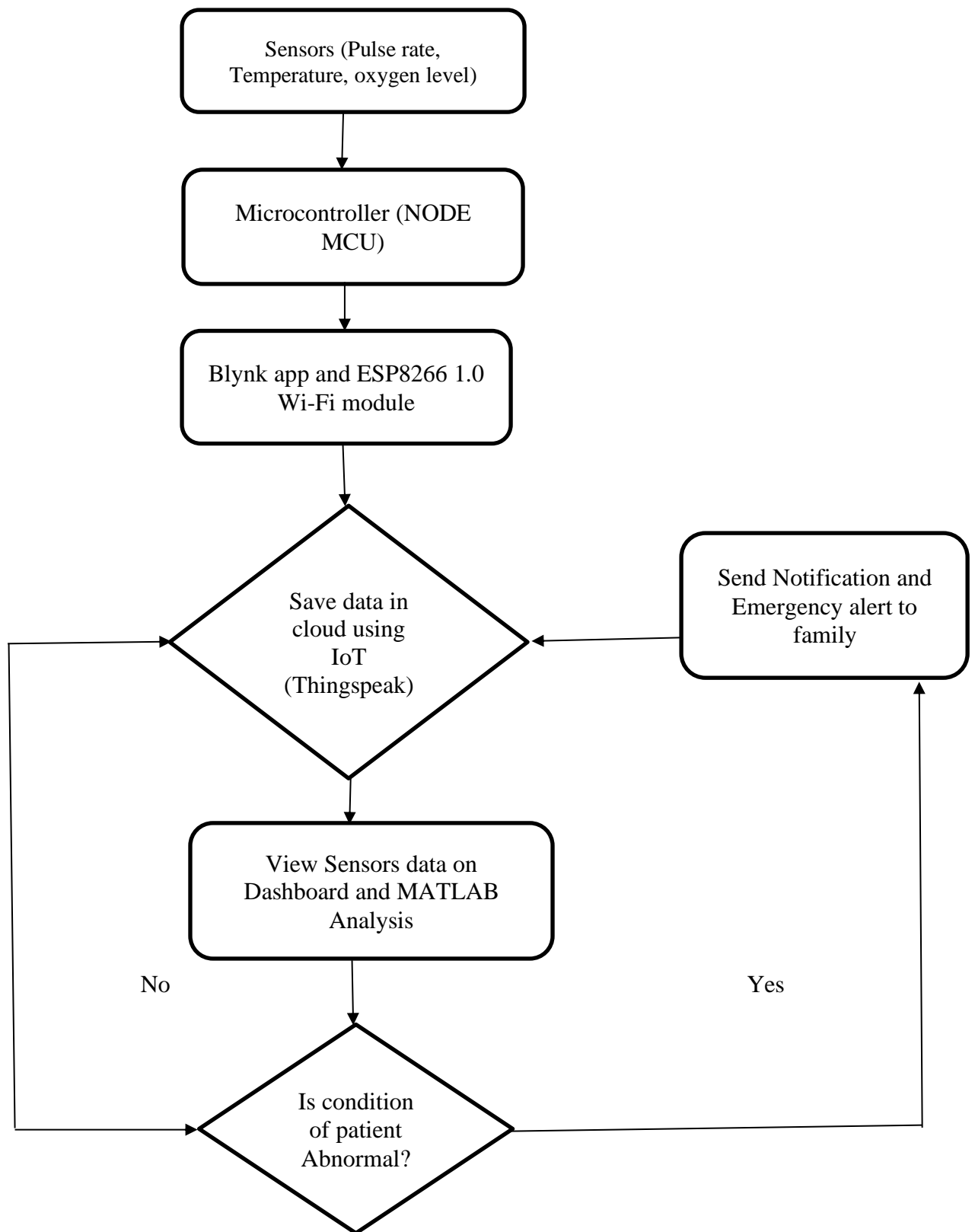


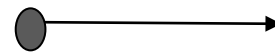
Figure 5.1.3 System Flow Chart of the proposed system (Level-1)

i. Activity Diagram

An activity diagram is a UML behavior diagram that represents the workflow of stepwise activities of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behavior diagrams. An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram is used by developers to understand the flow of programs on a high level.

Components of Activity Diagram

Start Point/initial state



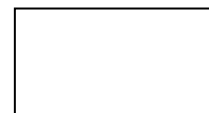
Activity



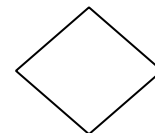
Action flow



Class/object



Decision/branching



Merge

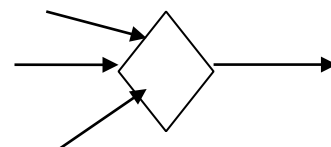


Figure 5.1.4 Components of activity diagram

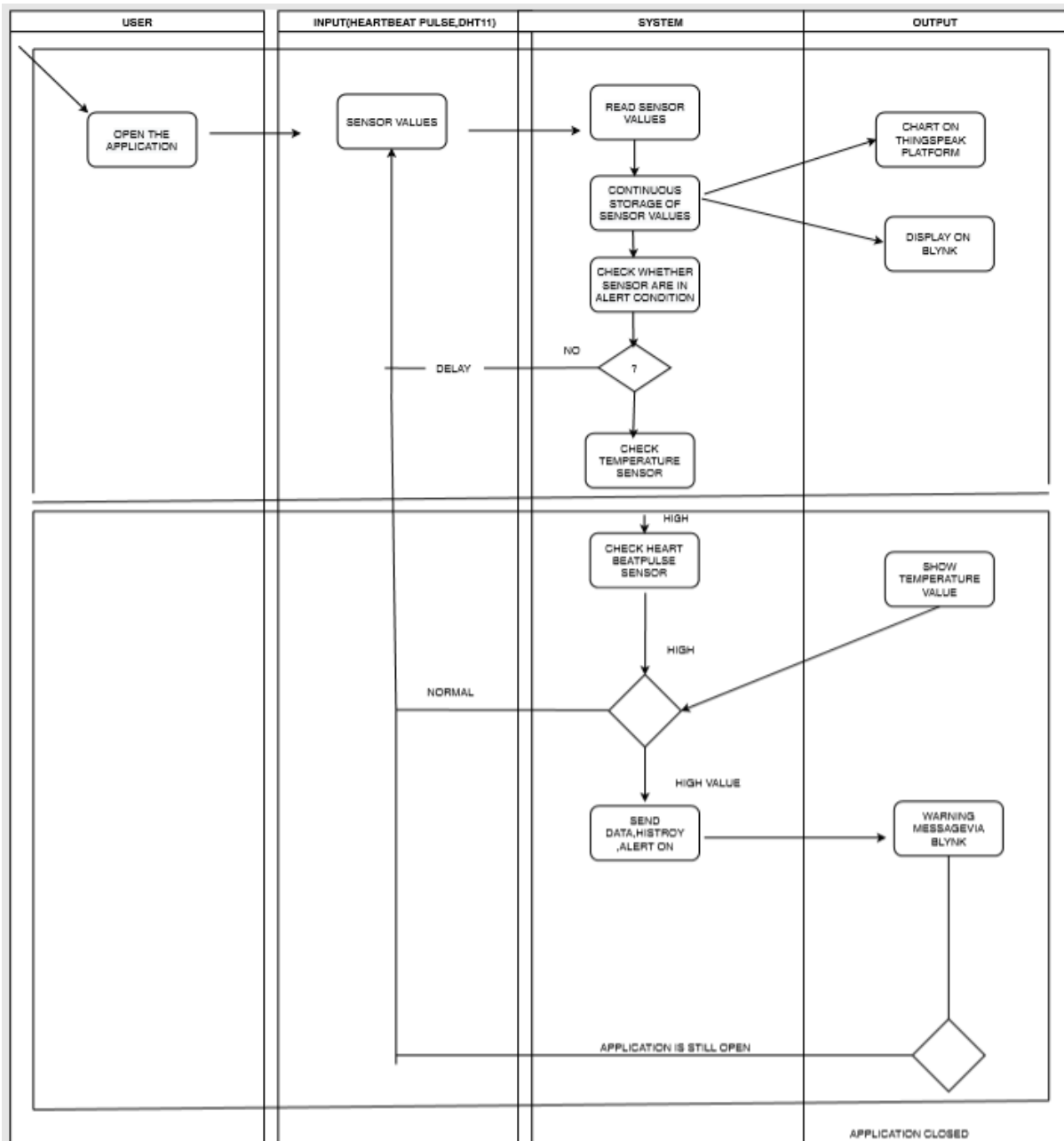


Figure 5.1.5 Project Activity Diagram

ii. UML Diagram

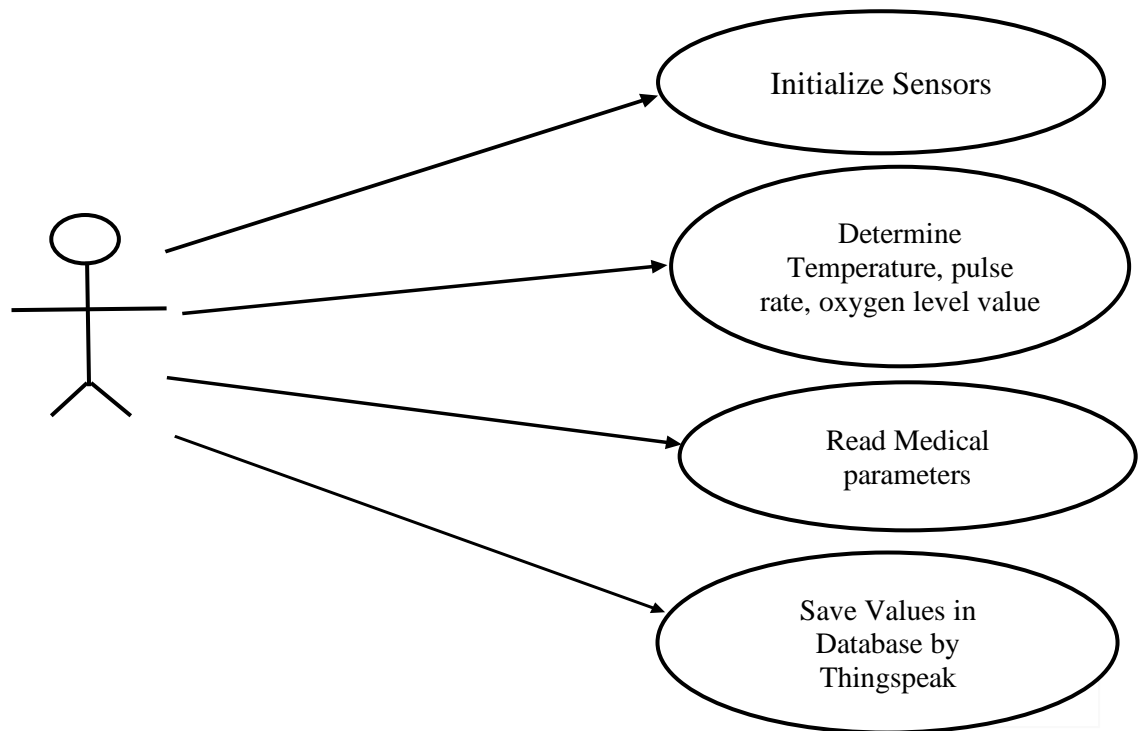


Figure 5.1.6 Use case Diagram

5.2 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve users more quickly and efficiently. The general objective of a database is to make information access easy, quick, inexpensive, integrated and shared by different applications and users. Database design is an important yet sometimes overlooked part of the application development lifecycle. An accurate and up-to-date data model can serve as an important reference tool for Database Administrators, developers, and other members of joint application development team. The process of creating a data model helps the team uncover additional questions to ask of end users. Effective database design also allows the team to develop applications that perform well from the beginning. By building quality into the project, the team reduces the overall time it takes to complete the project, which in turn reduces project development costs. The central theme behind database design is to "measure twice, cut once". Effective database designers will keep in mind the principles of normalization while they design a database.

Thingspeak is an IoT analytics platform service that allows you to aggregate, visualize and analyse live data streams in the cloud. Thingspeak provides instant visualizations of data posted by your devices to Thingspeak. With the ability to execute MATLAB code in Thingspeak you can perform online analysis and processing of the data as it comes in. Thingspeak is often used for prototyping and proof of concept IoT systems that require analytics.

5.2.1 THING SPEAK KEY FEATURES

Thingspeak allows you to aggregate, visualize and analyse live data streams in the cloud. Some of the key capabilities of Thingspeak include the ability to:

- Easily configure devices to send data to Thingspeak using popular IoT protocols.
- Visualize your sensor data in real-time.
- Aggregate data on-demand from third-party sources.
- Use the power of MATLAB to make sense of your IoT data.
- Run your IoT analytics automatically based on schedules or events.
- Prototype and build IoT systems without setting up servers or developing web software.
- Automatically act on your data and communicate using third-party services like Twilio® or Twitter

5.3 INPUT DESIGN

Input designing is the basic theory to be considered during system study. The input media used in the system is the keyboard. Details are entered in the system through different data entry screens. The system is designed in a user-friendly manner. Appropriate error messages are displayed when a false data is entered. Design of the system is web-oriented and is highly interactive to the users. The user interface design is very important for any application. The interface design defines how the software communicates within itself, to system that interpreted with it and with human who use it. The interface design is very good. The user will fall into an interactive software application.

Hardware inputs are:

- ✦ Heart Rate Pulse Sensor Amped
- ✦ DHT11 Sensor
- ✦ Power supply

Input form:

The screenshot shows the ThingSpeak web interface for a channel named "BSN-CARE". The channel ID is 3367134, and the API key is a268000000000000000000000000000000. The "Channel Settings" tab is active, showing a "Percentage complete" of 50%. The "Channel ID" is 3367134, and the "Name" is BSN-CARE. The "Description" field is empty. There are seven "Field" inputs: Field 1 (Temperature), Field 2 (Heart Rate), Field 3 (Respiratory Rate), Field 4, Field 5, Field 6, and Field 7. Each field has a "Public" checkbox. To the right, there is a "Help" section with instructions on how to use the channel settings, including a list of required fields (Name, Description, Location, API Key, and Tags) and a "Show Channel Location" section with latitude and longitude inputs.

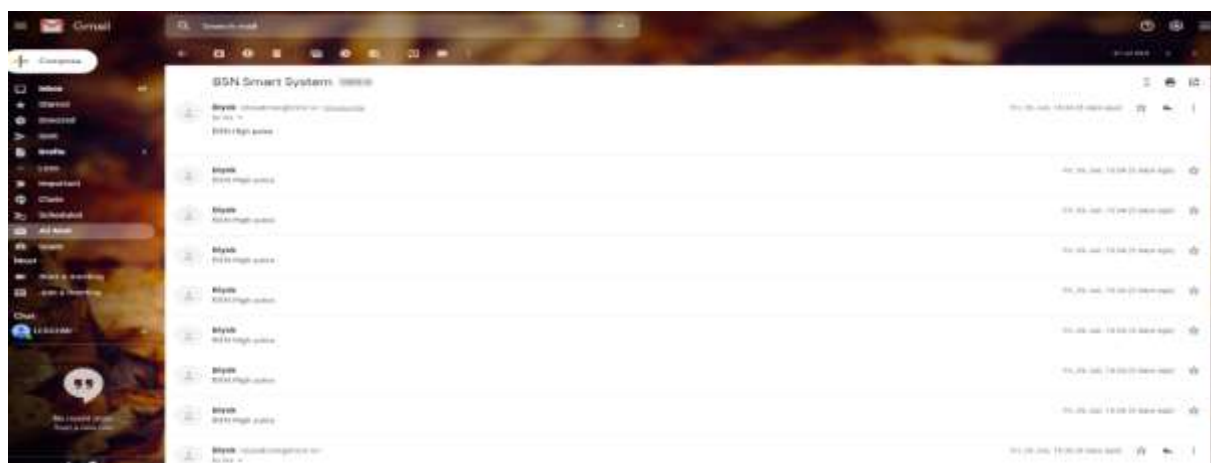
5.4 OUTPUT DESIGN

Computer output is the most important one to the user. A major form of the output is the display of the information gathered by the system and the servicing the user requests to the system. Output generally refers to the results or information that is generated by the system. It can be in the form of operational documents and reports. Since some of the users of the system may not operate the system, but merely use the output from the system to aid them in decision-making, much importance is given to the output design. Output generation hence serves two main purposes, providing proper communication of information to the users and providing data in a form suited for permanent storage to be used later on. The output design phase consists of two stages, output definition and output specification. Output definition takes into account the type of outputs, its contents, formats, its frequency and its volume. The output specification describes each type of output in detail.

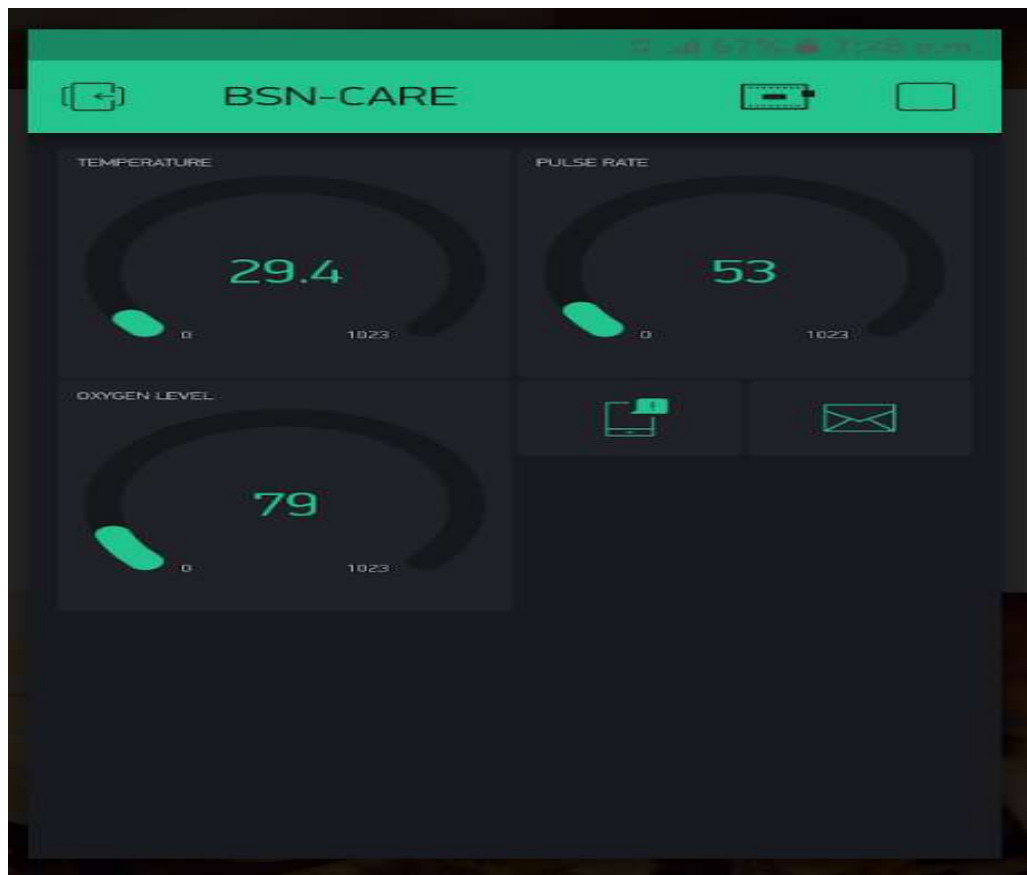
SENSOR REAL TIME READING VIA THINGSPEAK



MAIL ALERT



BLYNK SCREEN



PATIENT'S HISTORY DATA

Location	Altitude	Latitude	Longitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude	Altitude
----------	----------	----------	-----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

process	Input Design	Output Design
Measuring temperature of the patient.	Temperature sensor worn by the patient measures temperature and sends it.	Temperature on receipt stored in database.
Finding pulse rate, oxygen level of patient.	Respective sensors worn by patient senses pulse rate, oxygen level and sends them.	Pulse rate, oxygen level, received are stored in database.
Processing of data from NodeMCU to patient's mobile application (Blynk app).	Blynk app activated.	Medical parameters are displayed on mobile application of patient, and sent to Family. Abnormalities notified to all concerned.
Adding new patient.	Register with Email id and password.	New patient successfully added.
View patient details.	Open List of patients.	View list of patients.
History of patient.	Open particular patients page.	Read history of particular patient.

Table: 5.4.1 input output design

5.5 PROGRAM DESIGN

Temperature sensor

Step 1: A sensor meant to measure temperature of patient when active will detect temperature and pass it on to NodeMCU.

Step 2: The measured value is then sent to the database.

Step 3: It again go to step1 for reading the temperature continuously.

Heart beat pulse sensor Amped

Step 1: When active, this sensor will measure the pulse rate and oxygen level of the patient involved.

Step 2: The value so obtained is sent to database of the concerned patient.

Step 3: The sensor again goes to step1 if active to measure again.

Blynk Application

Step 1: Blynk is an open source android application which is designed and developed in order to control the hardware via IoT. This digitally displays server data; it can accumulate and visualize the data.

Step 2: If the application is active, processing of data from NodeMCU to patient mobile application (Blynk).

Step 3: Medical parameters are displayed on mobile application of patient and sent to healthcare team. Abnormalities noticed are notified to all concerned.

Thingspeak & ESP8266(NodeMCU)

Thingspeak provides instant visualization of data posted by ESP8266(NODE MCU) using cloud concept. NodeMCU ESP8266 upload sensor data using internet. This is a three-step process.

- Connect to WIFI hotspot having internet access.
- Read sensor data.
- Upload data to Thingspeak.

Step 1: Now sign up Thingspeak by entering your Email ID using your account.

Step 2: Configure Thingspeak.

Step 2.1: Create new channel. Click new channel enter name and field. You may have multiple fields depending on number of sensors. Create multiple field such as temperature, oxygen level and pulse rate etc.

Step 2.2: Click on API key, tab and write API key and update channel feed line. This line is required for data upload to cloud server.

Step 3: Programming ESP8266 to upload data to Thingspeak cloud server.

Step 4: Check data on Thingspeak server.

Open Thingspeak account and click on private view of channel.

CHAPTER 6

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

6.1 FUNCTIONAL REQUIREMENTS

Functional requirements represent the intended behaviour of the system. This behaviour may be expressed as services, tasks or functions that the specified system is required to perform.

Functional requirements are the main roles for which the system is designed. IOT based BSN healthcare system is intended to provide timely and quality healthcare to veterans. The proposed system consists of body sensors worn by the patients and wrist band with NodeMCU which regulates the output of the body sensors. output of the NodeMCU which are actually the health parameters of the patient is fed to the Blynk server through mobile network and thus the family is alerted who take appropriate action. Functional requirements for each of the uses cases described below:

- Descriptions of data to be entered into the system.
- Descriptions of operations performed by each screen.
- Descriptions of work-flows performed by the system.
- Descriptions of system reports or other outputs.
- Who can enter the data into the system?
- How the system meets applicable regulatory requirements.

6.2 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements are “system shall be requirement”. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", “quality goals”, "quality of service requirements" and "non-behavioural requirements. Some of the non-functional requirements are mentioned below

- Usability: The system shall have a clean interface with only needed features, clear terminology and tool tips wherever necessary. Warnings or alerts shall be specified in clear way.
- Efficiency: The system shall respond to different searches being conducted like searching particular product, search quantity, etc. in a very fast way.
- Interoperability: The system shall be able to interact with other systems. The system should able to be supported at least one software which has a relationship with payment process
- Portability: The system shall be independent of the specific technological platform used to implement it.
- Reliability: Reliability defined as a measure of the time between failures occurring in a system (measure show frequently the system fails), so that the system shall operate without any failure for a particular period of time
- Availability: Availability measures the percentage of time the system is in its operational state so that the system shall be available for use 24 hours per day and 365days per year.

CHAPTER 7

TESTING

Software testing is critical element of software quality assurance and represents the ultimate review of specifications, design and code generation. System testing is the stage of implementation, it is aimed for ensuring that the system works accurately and efficiently before live operations commences.

Testing is a purpose of executing a programmed with intend of finding errors.

- Preparing a test case that has high probability of finding undiscovered errors.
- Testing to erase out all kinds of bucks from the program.

Before going for testing, first we have to decide the type of test. For this impact system, unit testing is carried out. And the following things are taken to consideration.

- i. To ensure that information properly places in and out of the program.
- ii. To ensure that the module operates properly at boundaries established to limit or restrict processing.
- iii. To find out whether all statements in module have been executed at least once.
- iv. To find out whether error handling paths are working correctly.

7.1 TESTING STRATEGIES

A strategy for software testing integrates software test case design methods in to a well-planned series of steps that results in the successful construction of the software. The strategy provides a road map that describes the step to be conducted as part of testing, when these steps are planned and undertaken, and how much effort, time and resources will be required. Therefore, any testing strategy must incorporate test planning, test case, design, test execution and resultant data collection and evaluation. A software testing strategy should be flexible enough to promote customized testing approach. At the same time, it must be rigid enough to promote

reasonable planning and management tracking as the project processes. The project manager, software engineer and testing specialists develop a strategy for software testing. The general characteristics of software testing strategy are:

- Testing begins at the component level and works “outward” toward the integration of the entire computer system.
- Different testing techniques are appropriate at different point in time.

A strategy for software testing must accommodate low-level testis that are necessary to verify a small source code segment has been correctly implemented as well as high level testing that validate major system function against customer requirements.

These testing strategy checks the correctness of the newly engineered system. Errors found through the testing is rectified at the initial stages itself for better final product. Following are the series tests carried out: -

Unit Testing: In this each unit or the component is tested to see that the desired output is obtained.

Integration Testing: The objective here is to take the unit tested modules and see their performance where integrated with other modules of the system.

System Testing: In system testing, the software and the other system elements are tested as a whole. System testing is actually a series of different tests whose primary purpose is to fully exercise the whole system for its final desired result.

7.2 UNIT TESTING

Unit test comprises of a set test performed by an individual programmer prior to the integration of the unit into large system. Program unit is usually small enough that the programmers who developed and can it in great detail and certainly in greater than will possible when the unit is integrated into evolving software project. Unit testing should be an exhaustive as possible. In this system, each module was tested individually to ensure that every representation in the module meets the requirements.

Prior to integration of the unit in to a large system, it is necessary to perform a set of tests to each unit of the system. Accordingly, functional efficiency of all sensors used, alert by Blynk

application and correct storing of patient data etc. by the android app were tested repeatedly.

7.3 INTEGRATION TESTING

It is the systematic technique for constructing the program structure to uncover errors associated with the interface. The objective is to take unit tested module and built the program structure that has been dictated by design. Body sensors are connected to the NodeMCU and their output is fed to the mobile app. The entire thing is tested as a whole to see the functioning of integrated system.

Bottom-up integration consists of a unit test followed by testing of the entire system. Subsystem consists of several modules that communicated with other defined interface. The errors were isolated and corrected to produce a fully functional system. Top-down integration method is an incremental approach to the construction of the program structure. The project was tested to ensure that every representation meets the requirements.

7.4 SYSTEM TESTING

In system testing, the software and other system elements are tested as a whole. System testing is actually a series of different tests whose primary purpose is to fully exercise the IOT based BSN healthcare system.

7.5 USER ACCEPTANCE TESTING

This testing is generally performed when the project is nearing its end. This test mainly qualifies the project and decides if it will be accepted by the users of the system. The users or the customers of the project are responsible for the test.

7.6 DATA VALIDATION TESTING

Data validation is the process of testing the accuracy of data; a set of rule you can apply to a control to specify the type and range of data that can enter. It can be used to display error alert when users enter incorrect values into a form. In this project data validation testing carried out on all input from pages to test the accuracy.

7.7 OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered into two ways; one is on screen and another is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as the specified requirements by the user. Hence output testing does not result in any correction in the system.

7.8 TESTING RESULT

SL.NO	Procedures	Input	Expected Output	Pass or Fail
1	Temperature determination of patient.	Temperature input of patient.	Actual temperature of patient.	Is reading temperature correctly.
2	Determination of pulse rate and oxygen level.	Patient pulse rate and oxygen level.	Actual measure of patient's pulses rate & oxygen level.	Reading correctly
3	Accuracy of processing of data from NodeMCU to patients Thingspeak channel.	Medical parameters of patient.	Display of medical data accurately on real time.	Pass
4	Testing of Thingspeak operation accuracy.	Log in with username and password.	Real time medical data. Medical history sheet for a year & can be downloaded.	Pass
5	Blynk operation accurately.	Patients medical parameters from sensors.	Correct interpretation of patient's readings and detection of abnormalities and notifying the same.	Pass
6	Adding new patient.	New patient's data.	Registered successfully.	Fail
7	View patient data.	Patients data	Shows the list of user patient.	Pass
8	View history of any patient.	Patient's history data.	Shows the history of any user patient as desired.	Pass

Table 7.1 Testing and results

CHAPTER 8

RESULTS AND DISCUSSION

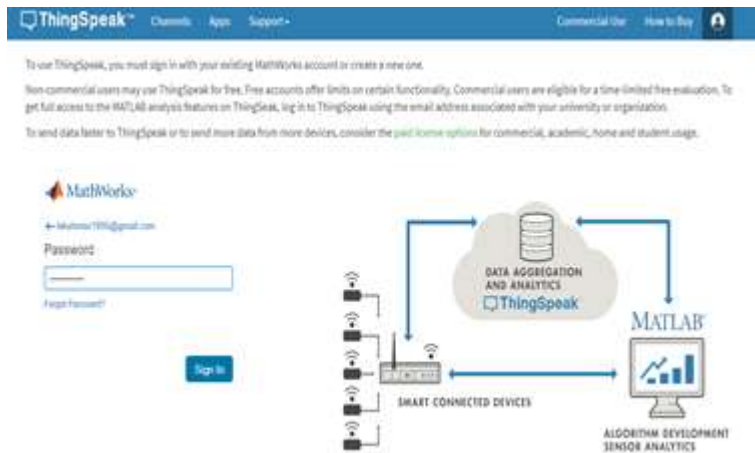
The existing healthcare system is age old and modern IOT based technology is not made use of resulting in poor end results. The proposed system make use of the modern communication technology to obtain better clientele satisfaction. The proposed IOT based BSN healthcare has the following advantages over the old system.

- Timely and appropriate medical care at the door step.
- Reduced human effort.
- Quick and appropriate action can be taken easily.
- Improved efficiency.
- Security enhanced for human life.
- Give proper information to the user.
- Economical.

Result of the project is undoubtedly great. An old patient who is physically unable to reach a doctor with the help of this system will be able to get quality and timely healthcare at this home. The medical data base of the patient is then transmitted through the mobile network to the designated family/member or friend on real time and thus the patient getting timely medical care. The information is passed on to designated family member/friend on real time by Blynk/cloud server through mobile network who then take appropriate action to save the patient.

8.1 SCREENSHOTS

1. Thingspeak Login Page



Patients data base can be seen here by the patient, friend or family/physician by logging in using mail ID & password. Patient's medical data can be seen in a graphical form with all details including date & time and the data is saved here for a year.

2. Thingspeak Real time medical data chart



This is real time medical data chart by Thingspeak. Patient's medical data is displayed here on real time on which patient//family/doctor can take action.

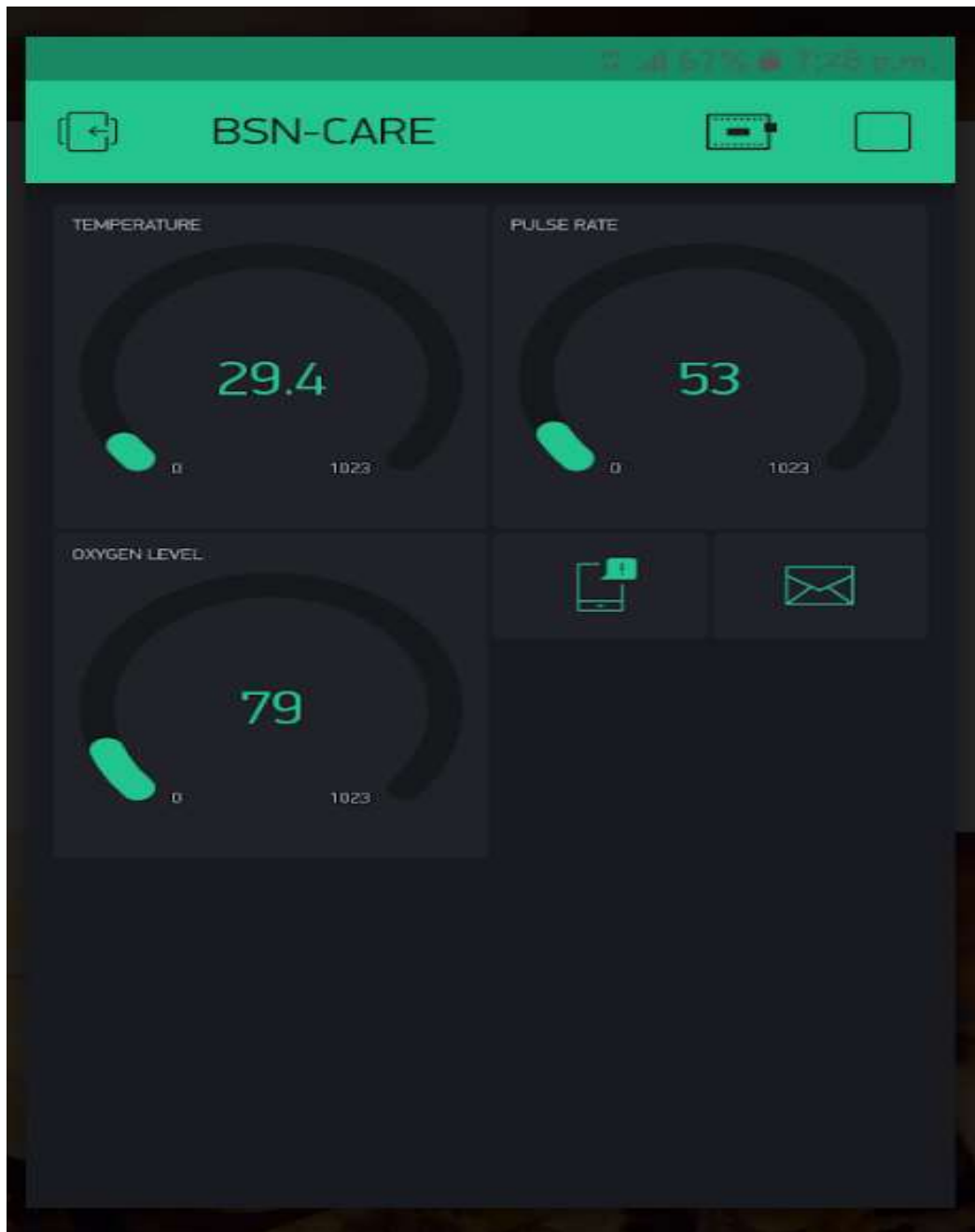
3. Patient's history data

created at	entry_id	field1	field2	field3	latitude	longitude	elevation	status
2020-06-26	1	29.5	108					
2020-06-26	97	29.5		48	72			
2020-06-26	98	29.5		63	94			
2020-06-26	99	29.5		52	78			
2020-06-26	54	29.4		48	72			
2020-06-26	55	29.4		54	81			
2020-06-26	56	29.5		54	81			
2020-06-26	57	29.5		54	81			
2020-06-26	58	29.5		53	79			
2020-06-26	59	29.5		54	81			

The screenshot shows a Thingspeak web interface with a data feed titled 'Total Patient Activation Failure'. The feed contains 59 data points, each with a timestamp (e.g., 2020-06-26 12:00:00), a latitude value (e.g., 29.5), a longitude value (e.g., 108), and an elevation value (e.g., 108). The interface includes a navigation bar with tabs for Home, Data, Alerts, and Settings. The data points are displayed in a table format with columns for Time, Latitude, Longitude, and Elevation.

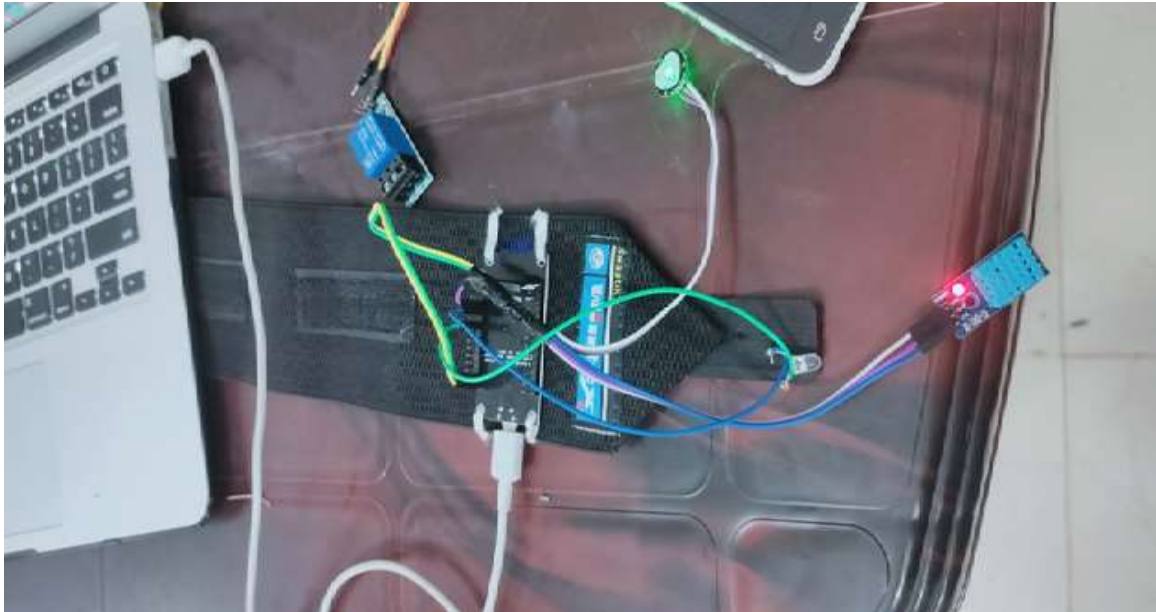
This is patient's medical history data saved by Thingspeak for a year. This can be seen on the screen by family/doctor/patient or friend can be downloaded as per our choice.

4.Blynk Screen



This is Blynk display screen. Medical parameters namely, Temperature, Pulse Rate and Oxygen Level etc. are displayed here. In case of up normal readings of any of these, Blynk will give a notification in its screen and will sent an Email to the family/doctor or friend.

5.wrist band assembly



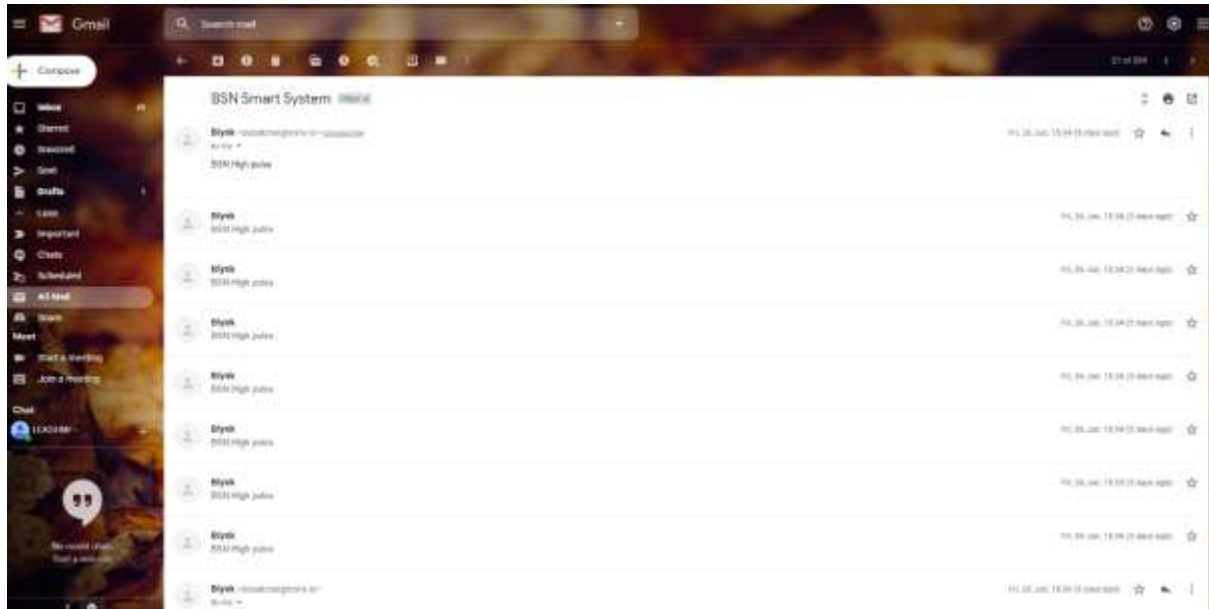
This screenshot depicts wrist band worn by patient. It consists of NodeMCU microcontroller with in built latest Wi-Fi module ESP8266 1.0. In addition, wrist band also has a DHT11 Temperature sensor, Heart rate pulse sensor, an LED and Buzzer. Temperature, pulse rate & oxygen level are known here.

6.Final Product



This screenshot shows the final product. Wrist band with all its components are fixed on to a suitable box for ease of operation. In addition, output of sensors fixed on the two fingers (Temperature sensor, Heart beat pulse sensor) are also integrated with this. LED and Buzzer is also used to indicate up normal readings.

7.Notification through Email



This is Email screen for notification. In case of up normal readings of the medical parameters of the patient, a notification in the form of Email is sent to friend/doctor/family.

CHAPTER 9

CONCLUSION

9.1 SYSTEM IMPLEMENTATION

System implementation is the most important phase of a project where theoretical design is turned in to a working system. In this stage the conceived system is installed and operated to see its efficiency and modification or changes if any are incorporated. In this phase, user training and trials are important to minimize errors or snags if any. In the proposed BSN-CARE project, selection of various body sensors and NODE MCU to suit the particular requirement of a patient is of paramount importance. Patients particularly elders are to be trained to use the body sensors. They also should be educated to make use of the mobile network. In case of a elderly patient who is not able to handle the sensors and cellphone due to his acute chronic medical condition, a friend or relative of the patient is also to be trained to take care of. Due care has to be given for successful implementation of the project. The project when fully developed and implemented will greatly change the old healthcare system.

9.2 CONCLUSION

IOT based BSN-CARE healthcare system is undoubtedly going to revolutionize the existing conventional healthcare practices of the world. Formation of long queues in hospitals and in front of renowned doctors for specialist consultation etc. will soon be a thing of the past. The conventional old healthcare system could not cater for medical emergencies and thus many breathing last due to not getting timely medical care. Modern IOT based BSN healthcare system envisages provision of quality and timely healthcare at your door step based on data received on real time at the exclusive healthcare network. IOT makes it possible to extend the network on to the internet and the data will be therefore available to the designated health team on real time who can react immediately. This system when fully developed will be a boon to all particularly older generation particularly those staying alone as timely and quality medical aid will be available to them even before they realize it's seriousness. More over global consultation of expert renowned specialist doctors will be just a click away which could not be even dreamt before. Therefore, BSN healthcare IOT based is a blessing to our generation as good health is an important part of modern life.

9.3 FUTURE ENHANCEMENT

- ❖ The system can be enhanced to cover in remote villages where IOT can be gainfully utilized.
- ❖ Body sensors in due course of time will be employing most modern technologies to minimize or make the radiations negligible.
- ❖ Since internet and mobile network has conquered the whole world making the whole world a global village IOT based BSN healthcare can be extended to benefit the whole global population.
- ❖ IOT based BSN healthcare will revolutise the field of medical care in due course of time benefitting the whole world.
- ❖ Once fully developed and fully integrated consultation of an expert specialist anywhere in the world will be just a click away from you.

REFERENCE/ BIBLIOGRAPHY

1.BOOKS

- ❖ Arduino: The complete guide to Arduino for beginners, Including Projects, Tips, Tricks and Programming, James Arthur, kindle edition published 2017 Nov 26, ISBN”1925989704”
- ❖ Programming Arduino; Getting started with sketches, Second Edition, by Simon Monk.Mc Graw hills. ISBN”1259641643”2011 Nov 8.
- ❖ Internet of Things (IOT): Systems and Applications: Jamil Y. Khan, Meh Met R. Yuce, publisher CRC Press,2019 Sep 17 ISBN”0429678053”

2.WEBSITES

<https://www.tutorialspoint.com/arduino/index.htm>

<https://www.javatpoint.com/iot-tutorial>

<https://www.wikipedia.com>

<https://stackoverflow.com>

<https://www.w3schools.com>

<https://slideshare.com>

https://www.mdpi.com/journal/sensors/special_issues/body_sensor_networks

3.JOURNALS AND PUBLICATIONS

- Kuo-Hui Yeh, A Secure IoT-Based Healthcare System with Body Sensor Networks IEEE. Volume 4, vol 16, no.6, December 9,2016
- P. Gope and T. Hwang, “BSN-care: A secure IoT-based modern healthcare system using body sensor network,” IEEE Sensors J., vol. 16, no. 5, pp. 1368–1376, Mar. 2016.
- Mosenia A, Sur-Kolay S, Raghunathan A, Jha N. Wearable Medical Sensor-based System Design IEEE Transactions on MultiScale Computing Systems. 2017 May 20.

APPENDIX

1.SCRUM MODEL

Git

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

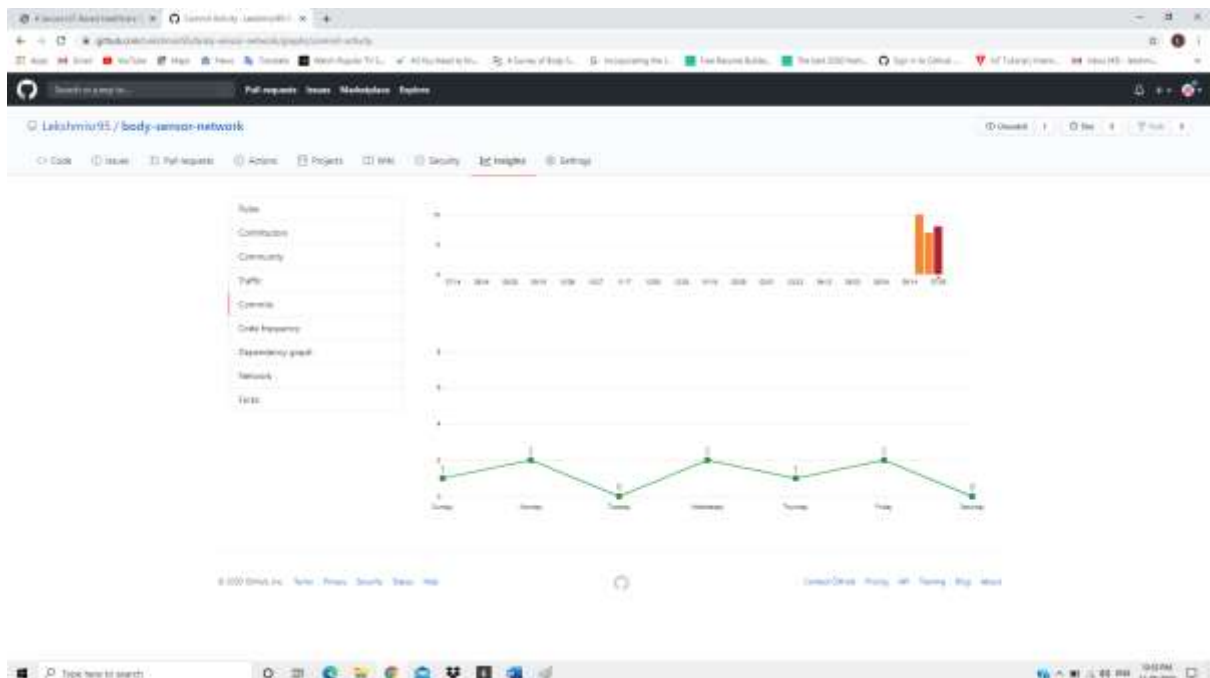
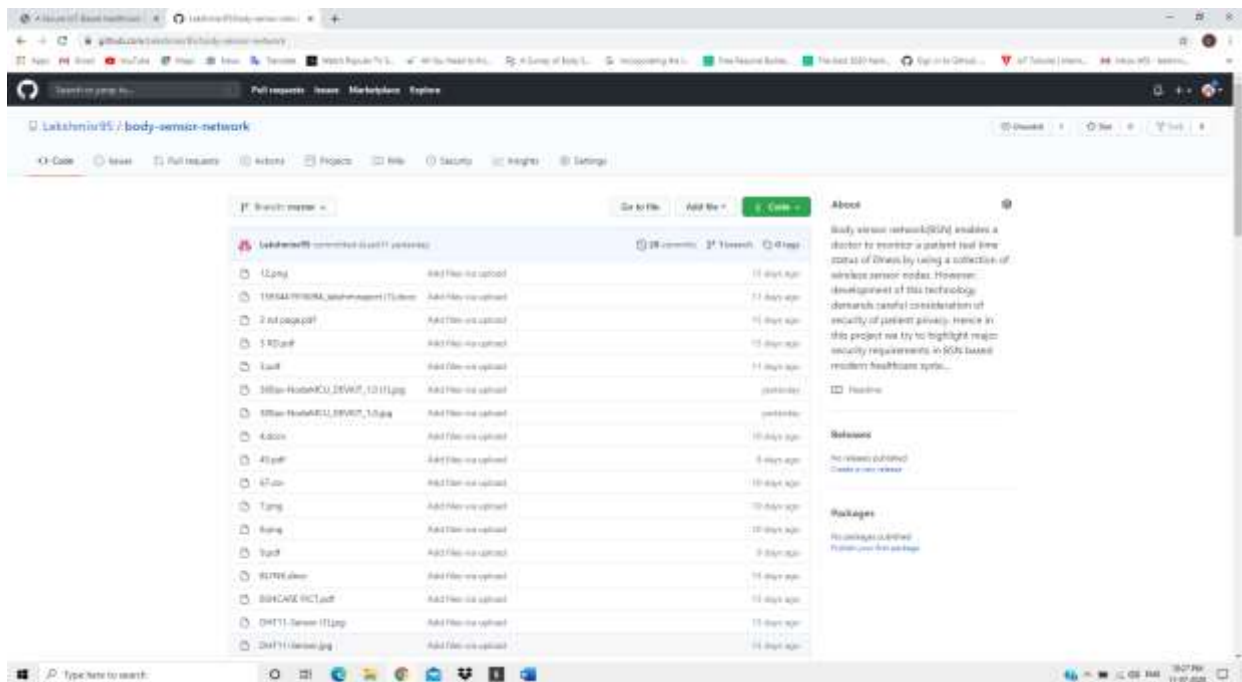
Git Repositories

A Git repository contains the history of a collection of files starting from a certain directory. The process of copying an existing Git repository via the Git tooling is called cloning. After cloning a repository, the user has the complete repository with its history on his local machine. Of course, Git also supports the creation of new repositories. If you want to delete a Git repository, you can simply delete the folder which contains the repository. If you clone a Git repository, by default, Git assumes that you want to work in this repository as a user.

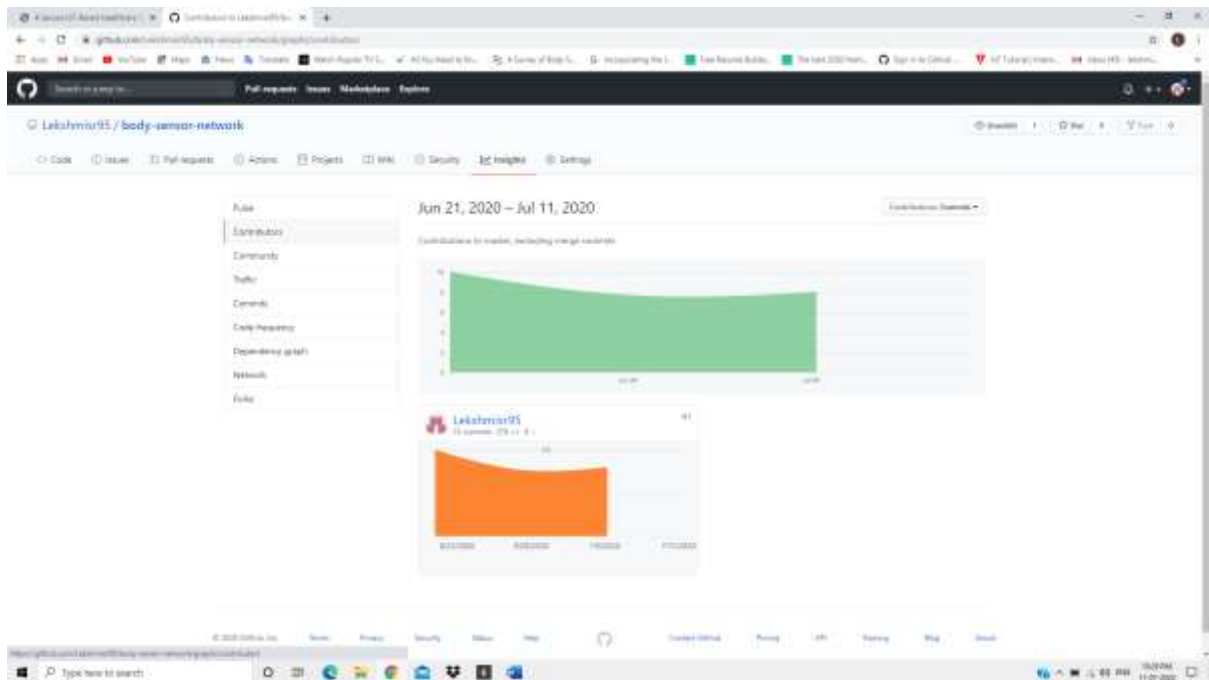
Scrum

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level. The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

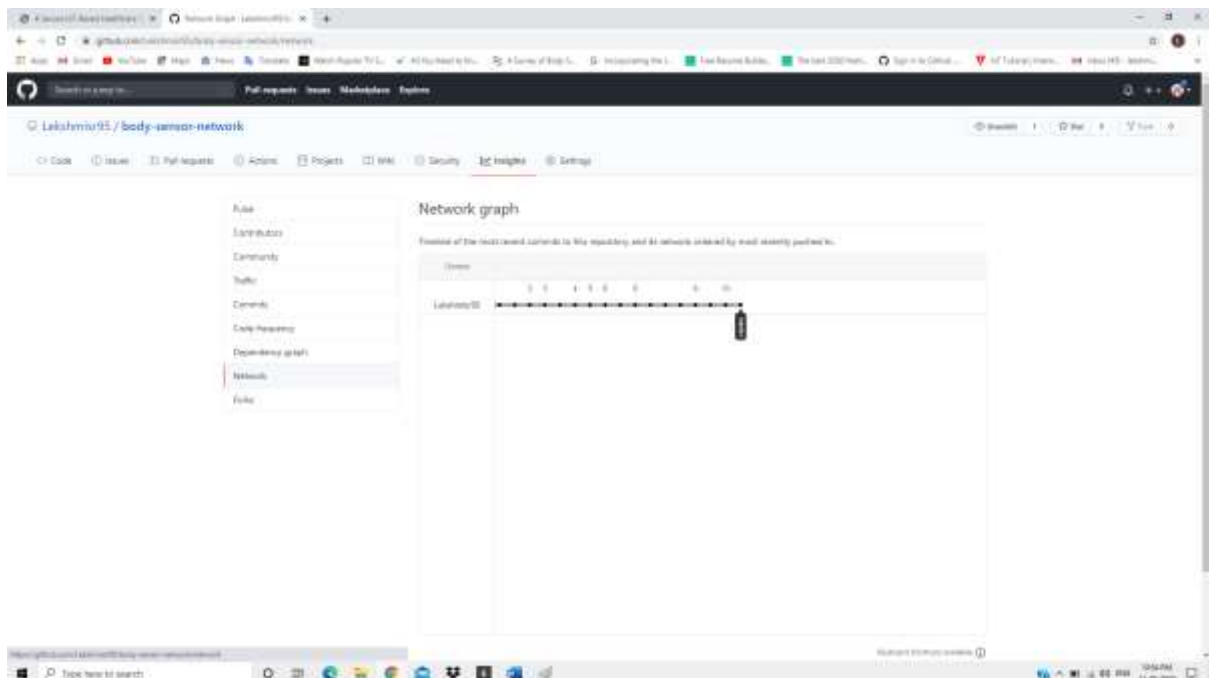
1.commits



2.Contributors



3.Network



2 LIST OF TABLES

Table No	Table Name	Page No
5.4.1	Input output table	42
7.1	Testing and results	51

3 LIST OF FIGURES

Figure No	Name	Page No
5.1	Block Diagram	31
5.1.3	System Flow Chart of proposed system	34
5.1.5	Project Activity Diagram	36
5.1.6	Use Case Diagram	37

4. ABBREVIATIONS AND NOTATIONS

IoT

Internet of Things (IoT) describes an emerging trend where a large number of embedded devices (things) are connected to the Internet. These connected devices communicate with people and other things and often provide sensor data to cloud storage and cloud computing resources where the data is processed and analyzed to gain important insights. Cheap cloud computing power and increased device connectivity is enabling this trend.

IoT solutions are built for many vertical applications such as environmental monitoring and control, health monitoring, vehicle fleet monitoring, industrial monitoring and control, and home automation.

GIT

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

UML

UML (Unified Modelling Language) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

5. CODING

Blynk

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
BlynkTimer timer;
char auth[] = "ZZUN3trrZey7dn57AQC_rtMGeDksuken";
char ssid[] = "AndroidAP";
char pass[] = "jwuv0451";

void sendSensor()
{
  int Sen=analogRead(A0);
  Blynk.virtualWrite(V0,Sen);
  if(Sen>580)
  {
    // Blynk.emaillekshmisr1995@gmail.com", "");
    Blynk.notify("Alert");
  }
}

void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

Thingspeak

```
#include <ESP8266WiFi.h> // ESP8266WiFi.h library
const char* ssid = "AndroidAP"; // replace subscribe with your WiFi SSID(Name)
const char* password = "jwuv0451"; //replace with Your Wifi Password name
const char* host = "api.thingspeak.com";
const char* writeAPIKey = "70ULX909YKM9XT7T"; //copy yout ThingSpeak channel
API Key.
void setup() {
  Serial.begin(115200);
```



```

delay(1000);
Serial.println("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
    Serial.println("");
    Serial.println("WiFi connected");
}
void loop() {
int Sen=analogRead(A0);
    WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
return;
}
    String url = "/update?key=";
    url+=writeAPIKey;
    url+="&field1=";
    url+=String(Sen);

    url+="\r\n";
// Request to the server
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
"Host: " + host + "\r\n" +
"Connection: close\r\n\r\n");
    Serial.print("sensor:");
    Serial.print(Sen);
    Serial.print("\n");
    Serial.println("Send to ThingSpeak.\n");
client.stop();
    Serial.println("Wait for 15 sec to update next datapack in thingSpeak");
    delay(1000);
}

```