A Project Report

on

SMART HEALTH MONITORING USING IoT

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

N.DIHITHA SAI (15FE1A05A3)	**M.CHANDRIKA (15FE1A0586)**
N.RUPA SRI (15FE1A05A6)	**P.KAVYA (15FE1A05C0)**

Under the guidance of

Ms.K.SUSHMA M.Tech.,

Assistant Professor
Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VIGNAN'S LARA INSTITUTE OF TECHNOLOGY AND SCIENCE
(Affiliated to Jawaharlal Nehru Technological University Kakinada, Kakinada)
(An ISO 9001:2008 Certified Institution, Approved by AICTE)
Vadlamudi, Guntur Dist., Andhra Pradesh-522213.
April - 2019.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VIGNAN'S LARA INSTITUTE OF TECHNOLOGY AND SCIENCE
(Affiliated to Jawaharlal Nehru Technological University Kakinada, Kakinada)
(An ISO 9001:2008 Certified Institution, Approved by AICTE)
Vadlamudi-522213

CERTIFICATE

This is to certify that the project report entitled **"SMART HEALTH MONITORING USING IoT"** is a bonafide work done by **N.DIHITHA SAI (15FE1A05A3), M.CHANDRIKA (15FE1A0586), N.RUPA SRI (15FE1A05A6)** and **P.KAVYA (15FE1A05C0)** under my guidance and submitted in fulfillment of the requirements for the award of the degree of Bachelor of Technology in **COMPUTER SCIENCE AND ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA.** The work embodied in this project report is not submitted to any University or Institution for the award of any Degree or Diploma.

Project Guide	Head of the Department
Ms. K. SUSHMA M. Tech.,	**Mr. T. V. VAMSI KRISHNA** M.Tech,(Ph.D)
Assistant Professor	**Assistant Professor**

External Examiner

DECLARATION

We hereby declare that the project report entitled **"SMART HEALTH MONITORING USING IoT "** is a record of an original work done by us under the guidance of **Ms. K. SUSHMA,** Assistant Professor of Computer Science and Engineering and this project report is submitted in the fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering. The results embodied in this project report are not submitted to any other University or Institute for the award of any Degree or Diploma.

	Project Members	Signature
Place: Vadlamudi.	**N.Dihitha Sai** (15FE1A05A3)	________
Date:	**M. Chandrika** (15FE1A0586)	________
	N.Rupa Sri(15FE1A05A6)	________
	P.Kavya(15FE1A05C0)	________

ACKNOWLEDGEMENT

The satisfaction that accompanies with the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We are grateful to **Ms. K.SUSHMA**, Assistant professor, Department of Computer Science and Engineering for guiding through this project and for encouraging right from the beginning of the project till successful completion of the project . Every interaction with her was an inspiration.

We thank **Mr. T. V. VAMSI KRISHNA,** Assistant Professor & HOD**,** Department of Computer Science and Engineering for support and Valuable suggestions.

We also express our thanks to **Dr. K. PHANEENDRA KUMAR,** Principal, Vignan's Lara Institute of Technology & Science for providing the resources to carry out the project.

We also express our sincere thanks to our beloved **Chairman Dr. LAVU RATHAIAH** for providing support and stimulating environment for developing the project.

We also place our floral gratitude to all other teaching and lab technicians for their constant support and advice throughout the project.

Project Members

N.Dihitha Sai (15FE1A05A3)

M.Chandrika (15FE5A0586)

N.Rupa Sri (15FE1A05A6)

P.Kavya (15FE1A05C0)

ABSTRACT

Present day's IoT brings the gadgets together and assumes a fundamental part in different methodologies like smart home mechanization, savvy urban areas, vehicle parking, traffic control, brilliant industries, smart environment, agribusiness fields and patient health monitoring system and so on. One of the approaches is to monitor the health state of the patient and screen it to doctors or paramedical staff through the IoT, as it is hard to screen the patient for 24 hours. So here in our system old age patients and children with heart problems should be periodically monitored by their guardians or consulted doctor need to be informed about their health status from time to time while at work. Our proposed system uses sensors that allow to detect heart rate and temperature. The sensors are kept in a watch to keep track of the patient's health, even if the person is at home. The patients position is conveyed through mobile app using cloud. As soon as the patient heart beat goes above or below a certain limit, the system sends an alert to the controller which then transmits this over internet and alerts the doctor at hospital as well as guardian. Thus concerned ones may monitor heart rate as well as get an alert of heart attack to the patient immediately from anywhere and can be saved on time. It is very helpful for the people to connect and take good care of their loved one's when they are not around with that person.

LIST OF FIGURES

FIGURE NO		PAGE NO
1.1	Connection of things over internet	2
3.1	Block Diagram	14
3.2	Schematic Diagram	15
3.3	Flow chart	16
4.1	Arduino Uno	18
4.2	NodeMcu	18
4.3	Pulse Sensor	19
4.4	DHT11 sensor	20
4.5	LED	20
4.6	Creation of fields that are needed for our project	27
4.7	MATLAB visualization after creating fields	28
4.8	API keys generation	29
4.9	Front page design	30
4.10	Completion of front end design	31
4.11	Blocks section for front page	32
4.12	Location page	33
4.13	Location blocks	34
4.14	Patient details page	35
4.15	Patient details page	35
4.16	Guardian details page	36
5.1	Values of pulse and temperature	38
5.2	Location of the patient	39
5.3	Patient Details	40
5.4	Guardian Details	40

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
6.1	A test case for values updation	42
6.2	A test case for location updation	42
6.3	A test case for navigating to patient details page	42
6.4	A test case for navigating to guardian details page	43
6.5	A test case over the working of the temperature sensor	43

LIST OF ABBREVIATIONS

IoT	Internet of Things
DHT	Digital Humidity and Temperature
IDE	Integrated Development Environment
LED	Light Emitting Diode
API	Application Programming Interface
MIT	Massachusetts Institute of Technology
AI	Artificial Intelligence
NFC	Near Field Communication
RFID	Radio Frequency Communication
LTE	Long Term Evaluation
P2P	Peer to Peer
Wi-Fi	Wireless Fidelity
ICU	Intensive Care Unit
MCU	Micro Controller Unit
HDU	High Dependency Unit
PHMS	Patient Health Monitoring System
BWSN	Body Wireless Sensor Network
CP-ABE	Cipher text policy Attribute Based Encryption
ECG	Electrocardiography

TABLE OF CONTENTS

TITLE	PAGE NO
ABSTRACT	i
LIST OF FIGURES	ii
LIST OF TABLES	iii
LIST OF ABBREVIATIONS	iv
CHAPTER 1: INTRODUCTION	1-10
1.1 Internet of Things	1-7
1.2 Background study	8
1.3 Problem Statement	8
1.4 Purpose of health monitoring system	9
1.5 Objective of the study	9-10
CHAPTER 2: LITERATURE SURVEY	11-13
2.1 Literature Study	11
2.2 Existing Methods	11-13
2.2.1 Used Algorithm	11-12
2.2.2 Drawbacks in that Algorithm	12-13
CHAPTER 3: PROPOSED METHODOLOGY	14-16
3.1 Block Diagram	14
3.2 Schematic Diagram	14-15
3.3 Algorithm	15
3.4 Flow Chart	16
CHAPTER 4: REQUIREMENTS	17-37
4.1 Requirement Analysis	17
4.2 Hardware Requirements	17-20
4.2.1 Arduino Uno	17-18
4.2.2 NodeMcu	18
4.2.3 Pulse Sensor	19
4.2.4 DHT11 Sensor	20
4.2.5 LED	20
4.3 Software Requirements	20-37
4.3.1 Arduino IDE	20-26
4.3.2 ThingSpeak Cloud	26-29

4.3.3 MIT AppInventor	29-36
4.3.4 Arduino Programming	36-37
CHAPTER 5: RESULTS	38-40
5.1 Displaying pulse and temperature values in App	38
5.2 Location of the Patient	38-39
5.3 Patient Details Display in App	39-40
5.4 Guardian Details display in App	40
CHAPTER 6: TESTING	41-43
6.1 Testing Levels	41
6.2 System test cases	41-43
6.3 Performance Testing	43
CHAPTER 7: CONCLUSION & FUTURE WORK	44-45
7.1 Conclusion	44
7.2 Advantages of proposed system	44
7.3 Future work	45
REFERENCES	46
URL's	47
APPENDIX : Source Code	48-52

CHAPTER 1

INTRODUCTION

1.1 Internet Of Things

The **Internet of things** (IoT) is the network of physical devices, vehicles, with electronics, software, sensors, actuators, and connectivity which enable these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system existing Internet infrastructure. Experts estimate that the IoT will consist of about 30 billion objects by 2020.It is also estimated that the global market value of IoT will reach $7.1 trillion by 2020.

The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

"Things," in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, cameras streaming live feeds of wild animals in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist firefighters in search and rescue operations. Legal scholars suggest regarding "things" as an "inextricable mixture of hardware, software, data and service".

These devices collect useful data with the help of various existing technologies and then autonomously flow the data between other devices. The term "the Internet of things" was coined by Kevin Ashton of Procter Gamble, later MIT’s Auto-ID Center, in 1999.

1) IoT − Key Features

The most important features of IoT include artificial intelligence, connectivity, sensors, active engagement, and small device use. A brief review of these features is given below:

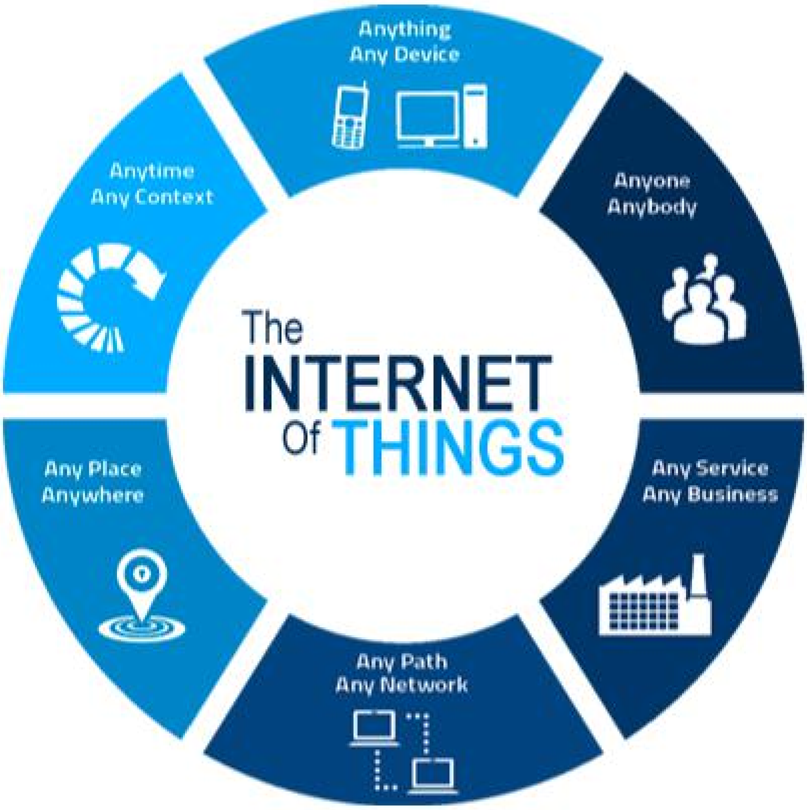

Fig 1.1: Connection of things over internet

i. AI – IoT essentially makes virtually anything "smart", meaning it enhances every aspect of life with the power of data collection, artificial intelligence algorithms, and networks. This can mean something as simple as enhancing your refrigerator and cabinets to detect when milk and your favorite cereal run low, and to then place an order with your preferred grocer.

ii. Connectivity – New enabling technologies for networking, and specifically IoT networking, mean networks are no longer exclusively tied to major providers. Networks can exist on a much smaller and cheaper scale while still being practical. IoT creates these small networks between its system devices.

iii. Sensors – IoT loses its distinction without sensors. They act as defining instruments which transform IoT from a standard passive network of devices into an active system capable of real-world integration.

iv. Active Engagement – Much of today's interaction with connected technology happens through passive engagement. IoT introduces a new paradigm for active content, product, or service engagement.

v. Small Devices – Devices, as predicted, have become smaller, cheaper, and more powerful over time. IoT exploits purpose-built small devices to deliver its precision, scalability, and versatility.

2) IoT – Advantages

The advantages of IoT span across every area of lifestyle and business. Here is a list of some of the advantages that IoT has to offer:

i. Improved Customer Engagement: Current analytics suffer from blind-spots and significant flaws in accuracy; and as noted, engagement remains passive. IoT completely transforms this to achieve richer and more effective engagement with audiences.

ii. Technology Optimization: The same technologies and data which improve the customer experience also improve device use, and aid in more potent improvements to technology. IoT unlocks a world of critical functional and field data.

iii. Reduced Waste: IoT makes areas of improvement clear. Current analytics give us superficial insight, but IoT provides real-world information leading to more effective management of resources.

iv. Enhanced Data Collection: Modern data collection suffers from its limitations and its design for passive use. IoT breaks it out of those spaces, and places it exactly where humans really want to go to analyze our world. It allows an accurate picture of everything. The new technology provides a wide range of enhanced data collection features where we can use advanced versions of cloud platforms etc.

3) IoT- Hardware

The hardware utilized in IoT systems includes devices for a remote dashboard, devices for control, servers, a routing or bridge device, and sensors. These devices manage key tasks and functions such as system activation, action specifications, security, communication, and detection to support-specific goals and actions.

i. IoT – Sensors: The most important hardware in IoT might be its sensors. These devices consist of energy modules, power management modules, RF modules, and sensing modules. RF modules manage communications through their signal processing, Wi-Fi, ZigBee, Bluetooth, radio transceiver, duplexer, and BAW. The sensing module manages sensing through assorted active and passive measurement devices. Here is a list of some of the measurement devices used in IoT: Devices accelerometers, temperature sensors, magnetometers, proximity sensors, gyroscopes, image sensors, acoustic sensors, light sensors, pressure sensors, gas RFID sensors, humidity sensors, micro flow sensors.

ii. Wearable Electronics: Wearable electronic devices are small devices worn on the head, neck, arms, torso, and feet. Smart watches not only help us stay connected, but as a part of an IoT system, they allow access needed for improved productivity. Current smart wearable devices include: Head – Helmets, glasses

4) IoT-Software

IoT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and process extension within the IoT network. They exploit integration with critical business systems (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

i. Data Collection: This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.

ii. Device Integration: Software supporting integration binds (dependent relationships all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because without them, it is not an IoT system. They manage the various applications, protocols, and limitations of each device to allow communication.

iii. Real-Time Analytics: These applications take data or input from various devices and convert it into viable actions or clear patterns for human analysis. They analyze information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry. Now-a-days real time analysis plays an important role where we have to acquire data from any place and at any time without any to get a smart environment.

iv. Application and Process Extension: These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

5) IoT − Technology and protocols

IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

i)NFC and RFID: RFID (radio-frequency identification) and NFC (near-field communication) provide simple, low energy, and versatile options for identity and access tokens, connection bootstrapping, and payments. RFID technology employs 2-way radio transmitter-receivers to identify and track tags associated with objects. NFC consists of communication protocols for electronic devices, typically a mobile device and a standard device.

ii)Low-Energy Bluetooth: This technology supports the low-power, long-use need of IoT function while exploiting a standard technology with native support across systems. iii. Low-Energy Wireless: This technology replaces the most power hungry aspect of an IoT system. Though sensors and other elements can power down over long periods, communication links (i.e., wireless) must remain in listening mode. Low-energy wireless not only reduces consumption, but also extends the life of the device through less use.

iii) Low-Energy Wireless: This technology replaces the most power hungry aspect of an IoT system. Though sensors and other elements can power down over long periods, communication links (i.e., wireless) must remain in listening mode. Low-energy wireless not only reduces consumption, but also extends the life of the device through less use.

iv) Radio Protocols: ZigBee, Z-Wave, and Thread are radio protocols for creating low-rate private area networks. These technologies are low-power, but offer high throughput unlike many similar options. This increases the power of small local device networks without the typical costs. This technology is not only used to retrieve the data from large distances but can also take information by using nearby radio protocols.

v) LTE-A: LTE-A, or LTE Advanced, delivers an important upgrade to LTE technology by increasing not only its coverage, but also reducing its latency and raising its throughput. It gives IoT a tremendous power through expanding its range, with its most significant applications being vehicle, UAV, and similar communication.

vi) Wi-Fi-Direct: Wi-Fi-Direct eliminates the need for an access point. It allows P2P (peer-to-peer) connections with the speed of WiFi, but with lower latency. WiFi-Direct eliminates an element of a network that often bogs it down, and it does not compromise on speed or throughput.

6) IoT − Common uses

IoT has applications across all industries and markets. It spans user groups from those who want to reduce energy use in their home to large organizations who want to streamline their operations. It proves not just useful, but nearly critical in many industries as technology advances and we move towards the advanced automation imagined in the distant future.

i)Engineering, Industry, and Infrastructure: Applications of IoT in these areas include improving production, marketing, service delivery, and safety. IoT provides a strong means of monitoring various processes; and real transparency creates greater visibility for improvement opportunities. The deep level of control afforded by IoT allows rapid and more action on those opportunities, which include events like obvious customer needs, nonconforming product, malfunctions in equipment, problems in the distribution network, and more.

ii)Government and Safety: IoT applied to government and safety allows improved law enforcement, defense, city planning, and economic management. The technology fills in the current gaps, corrects many current flaws, and expands the reach of these efforts. For example, IoT can help city planners have a clearer view of the impact of their design, and governments have a better idea of the local economy.

iii. Home and Office: In our daily lives, IoT provides a personalized experience from the home to the office to the organizations we frequently do business with. This improves our overall satisfaction, enhances productivity, and improves our health and safety. For example, IoT can help us customize our office space to optimize our work. Let us take another efficient example where our home get automated and get controlled itself without our help like when a person moves then to automatically on the fan or light etc.

iv. Health and Medicine: IoT pushes us towards our imagined future of medicine which exploits a highly integrated network of sophisticated medical devices. Today, IoT can dramatically enhance medical research, devices, care, and emergency care. The integration of all elements provides more accuracy, more attention to detail, faster reactions.

The diagnosis and treatment of patients involve several physiological parameters which need to be measured on a real-time basis like temperature, pulse rate, sugar level, Blood pressure, and most importantly the oxygen level. The sensor that communicates in mesh, collects and transmits some threshold parameter. This increases the efficiency and reliability of this field to a considerable level. It is not at all easy for the guardian to supervise the patient at each and every instant.

The Doctor or someone has to visit patient continuously in case of cardiac disease and in Intensive Care Unit (ICU). At such cases and also if the patient is not at home or at hospital the heart rate will be continuously monitored by the wearable device which has the pulse detector and send those values continuously to mobile application using NodeMcu. This will enhance the operational efficiency and also facilitate the early discharge of the patient. This will improve the normal life of a patient by reducing the risk of infection and severe condition when the doctor or nurse or guardian is not nearby.

This project describes the design of a heart rate and body temperature measuring device which displays the information on a Mobile Application. For each parameter, the threshold is decided. The audible warning sound will be arranged on the mobile application based on the limit over the heart rate. When the threshold value is exceeded. These signals are displayed on the remote system.

Apart from this when the patient is not nearby to guardian then location of that patient also shown in the mobile application. And with these the previous patient's record and guardian's details are also displayed in application for better treatment for the patient and intimate the guardian based on those details.

This mobile App will be present in the patient's device and the guardian's device. We can create several instances of the application.

This project consists of simple and low-cost components which are capable of processing real-time parameters like temperature, heart rate. We can advance it later for further improvements which were going to be discussed in further chapters in a detailed way to enhance the system in a better way.

1.1Background of the study

1)About remote health monitoring system

A remote health monitoring system is an extension of a hospital medical system where patient's vital body state can be monitored remotely. Traditionally the detection systems were only found in hospitals and were characterized by huge and complex circuitry which required high power consumption. Continuous advances in the semiconductor technology industry have led to sensors and micro controllers that are smaller in size, faster in operation, low in power consumption and affordable in cost. This has further seen development in the remote monitoring of vital life signs of patients especially the elderly. The remote health monitoring system can be applied in the following scenarios:

a) A patient is known to have a medical condition with unstable regulatory body system. This is in cases where a new drug is being introduced to a patient.

b) A patient is prone to heart attacks or may have suffered one before. The vitals may be monitored to predict and alert in advance any indication of the body status.

c) Critical body organ situation

d) The situation leading to the development of a risky life-threatening condition. This is for people at an advanced age and maybe having failing health conditions.

A simple patient monitoring system design can be approached by the number of parameters.

2) Multi-parameter monitoring system:

This has multiple parameters being monitored at the same time. An example of such a system can be found in High Dependency Units (HDU), Intensive Care Units (ICU), during the surgery at a hospital theatre or Post surgery recovery units in Hospitals. Several parameters that are monitored include the heartbeat, temperature. The Multi parameter monitoring health monitoring system can come in handy.

1.3 Statement of the problem

It is not possible to look after the one who is suffering from heart diseases. The guardian may not be available all the time with the patient. For example if the patient went outside and suddenly collapsed and there is no way to intimate the guardian about such condition. So we have to take care of the patient before hand and intimate the location of the patient to the guardian. And the patient's previous medical reports should also be there with patient in order to give better treatment.

1.4 Purpose Of Patient Health Monitoring System (PHMS)

A doctor or health specialist can use the patient health monitoring system(PHMS) to monitor remotely of all vital health parameters of the patient or person of interest. An attempt at designing a remote healthcare system made with locally available components.

A) Temperature, Heart Rate

B) Mobile Application to display every details like patient's previous record, and particular location where the patient is situated

C) A simple Thingspeak cloud to send data from sensor to the mobile application.

1.5 Objective of the study

Here the main objective is to design a Patient Health Monitoring System to diagnose the health condition of the patients. Giving care and health assistance to the bedridden patients at critical stages with advanced medical facilities have become one of the major problems in the modern hectic world. In hospitals where many patients whose physical conditions must be monitored frequently as a part of a diagnostic procedure, the need for a cost-effective and fast responding alert mechanism is inevitable. Proper implementation of such systems can provide timely warnings to the medical staffs and doctors and their service can be activated in case of medical emergencies.

The use of sensors detects the conditions of the patient and the data is collected and transferred using a microcontroller. The guardian need to be with the patient frequently to examine his/her current condition. So we have introduced a wireless system where the patient details like pulse rate and temperature are continuously monitored or read for the patient and displayed on the mobile application using the Thingspeak cloud where the arduino acts as the controller and NodeMcu will send the data to the cloud. Through that cloud the data is transferred to the mobile application.

In addition to these when the patient is collapsed at some other places then the patient's location has to be displayed on the mobile application. The patient's previous record's will help the doctor to treat the patient in a better way with that information. Apart form that the guardian's details are also kept along with all these details to intimate them and contact them if they have any queries about the patient's health condition. These all details will help the patient's treatment in a better way.Physiological monitoring hardware can be easily implemented using simple interfaces of the sensors with a Micro controller and can effectively be used for health care monitoring.

This will allow development of such low-cost devices based on natural human-computer interfaces.

The system we proposed here is efficient in monitoring the different physical parameters of many number of patients and then in alerting the concerned guardians if these parameters bounce above its predefined critical values. Thus, remote monitoring and control refer to a field of industrial automation that is entering a new era with the development of sensing devices.

The Internet of Things (IoT) platform offers a promising technology to achieve the health care services, and can further improve the medical service systems. IoT wearable platforms can be used to collect the needed information of the user and its ambient environment and communicate such information via wireless, where it is used for tracking the location of the user. Such a connectivity with external devices and services will allow for taking preventive measure (e.g., upon foreseeing an upcoming heart stroke) or provide immediate care (e.g., when a user falls and needs help). The patient previous records are helpful for better treatment of the patient.

CHAPTER 2
LITERATURE SURVEY

2.1 Literature Study :

One of the buzzwords in the Information Technology is Internet of Things (IoT). The future is Internet of Things, which will transform the real world objects into intelligent virtual objects. The IoT aims to unify everything in our world under a common infrastructure, giving us not only control of things around us, but also keeping us informed of the state of the things. In Light of this, present study addresses IoT concepts through systematic review of scholarly research papers, corporate white papers, professional discussions with experts and online databases. Moreover this research article focuses on definitions, geneses, basic requirements, characteristics and aliases of Internet of Things. The main objective of this paper is to provide an overview of Internet of Things, architectures, and vital technologies and their usages in our daily life. However, this manuscript will give good comprehension for the new researchers, who want to do research in this field of Internet of Things and facilitate knowledge accumulation in efficiently .

IoT based Smart health care with the help of smart devices and objects improves the health care monitoring system effectively, thus by reducing the inefficiencies of existing health care system. Smart devices with new and upgraded technologies enhances the data accuracy to be collected, real-time accessibility of patient’s condition, intelligent integration of data collected, maintaining the integrated data smartly through cloud service, etc.

2.2 Existing Methods :

There are several existing where they could not overcome certain problems. To discuss about the problems that they didn’t solved first let us know briefly about the previously existed systems.

2.2.1 Used Algorithm :

The specialized health care monitoring system for elderly people is a growing need in the aging population world. This system performs basic health checkups by measuring the body parameters regularly and report the data to the doctors. The result data are then displayed as statements in a web application where doctors and patients can interact with each other[1].

In Remote health monitoring system using IoT, Body wireless sensor Network (BWSN) is used to transmit the patients' health parameters collected through Raspberry Pi micro controller to the physicians and caretaker wirelessly [2]. Being long range wireless technology, emergency situation of the patient's health is quickly detected and timely intervention leads to save the life of the patient.

IoT along with smart devices reduce complexity and complications in the health care system. The penetration of mobile technologies and smart devices over health care system cause huge impact on the world. The full-fledged utilization of M-health and E-health applications in today's world is made aware to the people for improving and maintaining the good quality of life. Apart from regular monitoring of patients condition through M-health system, the main objective is to educate them through recommendations of healthy eating habits and effective workout routines for improving their quality of healthy life [3].

Owing to costlier health care and long waiting time in hospitals, the concept of in-home patient monitoring system have been emerging in the recent years. This system collects data of various body parameters through Biosensors, wearable devices and smart textiles and it transmits the data to central node server securely through Cipher text Policy Attribute Based Encryption (CP-ABE) method. In turn, the server shares the collected data to the hospitals for further treatment[4].

Internet of Things (IoT) and cloud computing plays a vital role in today's Tele-monitoring health system. This system keeps track of patient's physiological parameters through collection of body sensors' data using Raspberry Pi board. The patient's health card are developed by the doctors and displayed on a web page where doctors and patients can access and communicate each other without physical presence [5].

Existing methodologies in patient monitoring system focuses on providing better health care facilities to a number of patients with limited medicinal resources. These monitoring systems limit the patients to the bed and enable them to move around only a particular range from the bed side.

2.2.2 Drawbacks in that Algorithm :

As we have observed many of the existing systems before, then let's discuss about the disadvantages that we had observed and going to overcome in proposed system.

a) Displaying the data in the lcd screen will not effect any improvement because it is as same as ECG machine in the hospital. So, there we are not providing any solution by displaying in lcd screen.

b) Improper monitoring of health.

c) This is not cost efficient.

d) Increase in death rates.

e) Requires a specialist to operate.

f) Time-consuming

g)Time to time monitoring is not possible

2.3 Proposed Method :

Old age patients and children with heart problems should be periodically monitored by their guardians or consulted doctor need to be informed about their health status from time to time while at work. Our proposed system uses sensors that allow to detect heart rate and temperature. The sensors are kept in a watch to keep track of the patient's health, even if the person is at home.The patient's position is conveyed through mobile app using cloud. As soon as the patient heart beat goes above or below a certain limit, the system sends an alert to the controller which then transmits this over internet and alerts the doctor at hospital as well as guardian. Thus concerned ones may monitor heart rate as well as get an alert of heart attack to the patient immediately from anywhere and can be saved on time.

CHAPTER 3
PROPOSED METHODOLOGY

The prime goal is to develop a reliable patient monitoring system so that the guardians can monitor the patients, who are either hospitalized or executing their normal daily life activities. We propose an innovative system in which our system puts forward a smart patient health tracking that uses sensors to track patient health. The main objective is to design a Patient Monitoring System i.e. the patient's data will be sent to the guardian through App and location on emergencies. Patient, doctor or guardian can able to track patient's location at any point in time through Google Maps which would enable to send medical services in case of an emergency .The sensors are connected to the micro controller to track the status which is in turn interfaced to monitor display.Mobile Application is used send health information to the mobile application by staying in remote areas and also updates it. Patient's previous records will be there in the mobile application.

3.1 Block Diagram :

The block diagram is nothing but the general arrangement of the parts or the components of a complex system or process, such as an electronic circuit. Fig 3.1 shows the interconnection of circuits with each other in a simple way. The block diagram for smart health monitoring will be as follows:

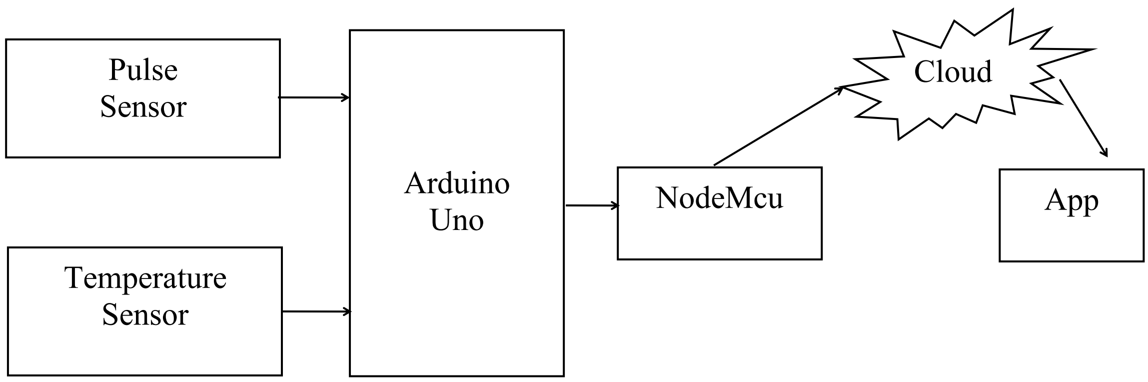

Fig 3.1 : Block Diagram

3.2 Schematic Diagram :

A schematic diagram is a representation of the elements of a system using abstract, graphic symbols rather than realistic pictures. Let us take a look at the schematic diagram of the pulse sensor without realistic picture of it and we will come to how the process of the sensor is done internally.

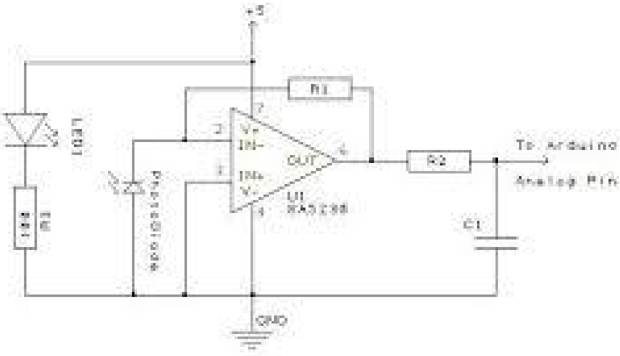

Fig 3.2 : Schematic Diagram

3.3 Algorithm :

An algorithm gives you a step by step process used for problem solving. The below are the steps for smart health monitoring using IoT.

Step 1: Know about the pins for every sensor.

Step 2: Initialize the code with the known pins. Be careful to notice the pins because based on the designated pins in code we have to connect the pins.

Step 3: After initialization is done make a call to setup function to set input and output data like pulse sensor will take input through analog pin and as well as the temperature sensor.

Step 4: Then write the code the that has to run continuously in the loop section of the program.

Step 5 : Thereafter establish connection between code and the ThingSpeak cloud using NodeMcu.

Step 6 : Generate API keys in he cloud by creating fields in the channel.

Step 7 : Save those keys to establish connection between cloud and the mobile application.

Step 8 : Develop an application by using the generated API keys.

Step 9 : Connect the circuit based on the pin numbers designated in the code.

Step 10: Run the code and check whether the data is updating in the mobile application.

3.4 Flow chart :

The work flow of the smart health monitoring is

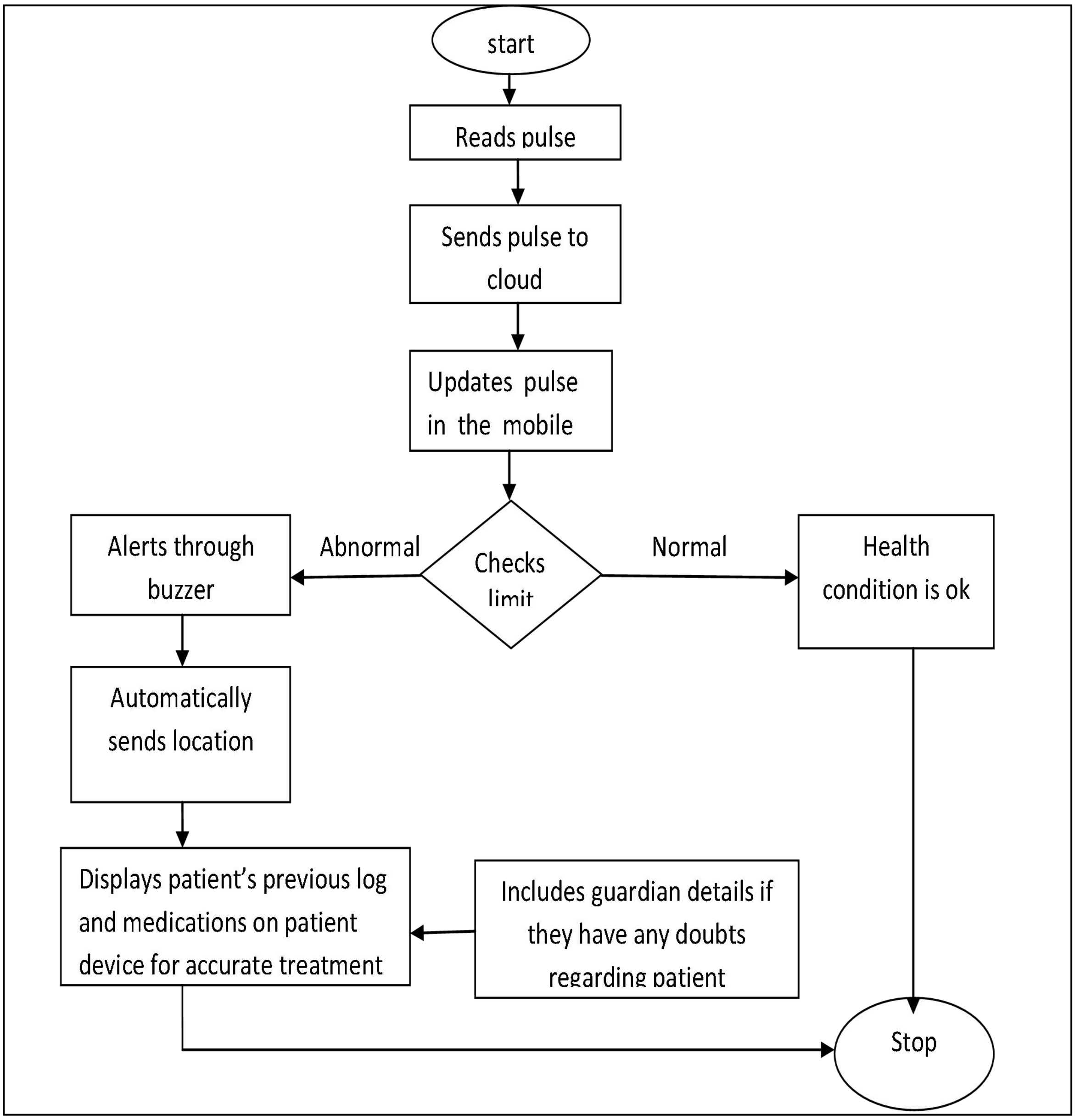

Fig 3.3: Flow chart of Proposed System

CHAPTER 4
REQUIREMENTS

4.1 Requirement analysis

Requirement Analysis is the first and important phase of the software developing the activity in developing any kind of project effectively. I started to list out all the functionalities that my application should provide. There have been some minor changes with respect to the functionalities over the course of development. Following are the requirements that have been implemented in this project.

1)Functional requirements:

a) The application must have a module to show patient's pulse rate and temperature.

b) Location Tracking: Application must have track location option with which doctor or guardian can track the location of the patient.

c) Based on threshold over pulse rate siren should play on application.

d) Guardian's previous details should be present in another module in application.

2) Non-functional requirements:

Non-functional requirements are not directly related to the functional behavior of the system.

a) Mobile application must be user-friendly, simple and interactive.

b) The user interface is designed in such way that novice users with little knowledge of web should be able to access this application.

c) Users are required to have some knowledge regarding lattitude and longitude in some cases.

4.2 Hardware Requirements:

4.2.1 Arduino Uno:

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 micro controller on the board. To do so you use the Arduino programming language(based on wiring), and the Arduino IDE, based on processing.

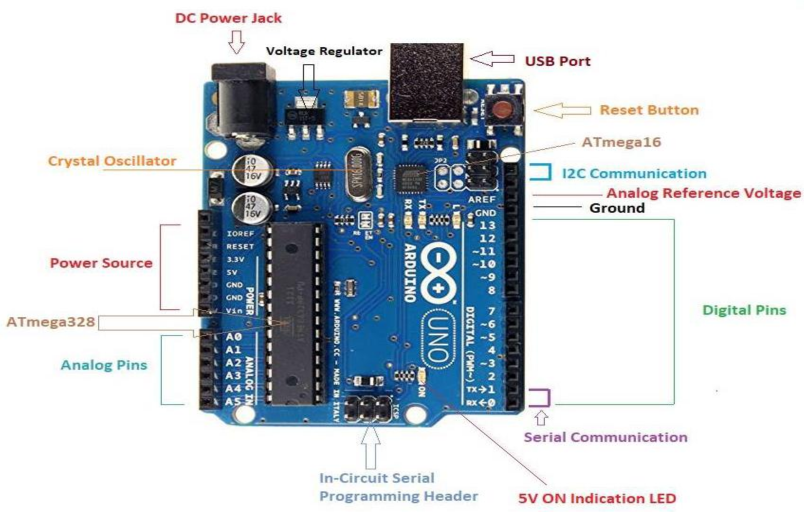

Fig 4.1: Arduino Uno

4.2.2 NodeMcu:

NodeMcu is an open source IoT platform. Which includes firmware which runs on the ESP8266 Wi-Fi Module from Espressif Systems,and hardware which is based on the ESP-12 module. The term “NodeMcu” by default refers to the firmware rather than the dev kits. NodeMcu firmware was developed so that AT commands can be replaced with Lua scripting making the life of developers easier. So it would be redundant to use AT commands again in NodeMcu. The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and microcontroller capability. It is a wi-fi module which is used to retrieve the data or modify the data from anywhere in the world. This wireless module once if connected to the cloud then it can be run nowhere in the earth with network connection.

Fig 4.2 : NodeMcu

4.2.3 Pulse Sensor:

The heartbeat sensor is based on the principle of photo phlethysmography. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of applications where heart pulse rate is to be monitored, the timing of the pulses is more important. The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.There are two types of photo phlethysmography:

Transmission: Light emitted from the light emitting device is transmitted through any vascular region of the body like earlobe and received by the detector.

Reflection: Light emitted from the light emitting device is reflected by the regions.

Fig 4.3: Pulse Sensor

Working of pulse sensor:

The basic heartbeat sensor consists of a light emitting diode and a detector like a light detecting resistor or a photodiode. The heart beat pulses causes a variation in the flow of blood to different regions of the body. When a tissue is illuminated with the light source, i.e. light emitted by the led, it either reflects (a finger tissue) or transmits the light (earlobe). Some of the light is absorbed by the blood and the transmitted or the reflected light is received by the light detector. The amount of light absorbed depends on the blood volume in that tissue. The detector output is in form of electrical signal and is proportional to the heart beat rate.

4.2.4 DHT11 Sensor:

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

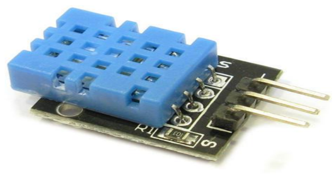

Fig 4.4: DHT11 Sensor

4.2.5 LED:

A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. This is used to show the output normally to the viewers offline.

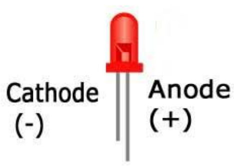

Fig 4.5 : LED

4.3 Software Requirements:

4.3.1 Arduino IDE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches:

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiringproject, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

The below small icons indicates how to compile and run our code in the IDE. Without knowing these terms it is very difficult to execute the program. There are various facilities provided by Arduino IDE like predefined examples where it will help us to understand the sensors without knowing them. And one more important thing is that COM port errors will come while embedding the code to the sensors. Those can be negotiated while reading the below content. And for ESPCOM errors will come where we have to install the versions to the software and everything is explained in a detail view to overcome every possibility of the error.

Now let us see the all option provided by arduino IDE to work in an efficient way. These will help you to do the tasks in the IDE without any interruptions because there is a high probability of occurring errors without knowing these basic things. If we know them properly then we can rectify the error easily to continue the task without any delay.

Verify

Checks your code for errors compiling it.

Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbookmenu instead.

Save

Saves your sketch.

Serial Monitor

Opens the serial monitor.

Additional commands are found within the five menus : File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

New

Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

Open

Allows to load a sketch file browsing through the computer drives and folders.

Sketchbook

Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

Examples

Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

Close

Closes the instance of the Arduino Software from which it is clicked.

Save

Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.

Save as...

Allows to save the current sketch with a different name.

Quit - Quits from the existing file.

Edit

Undo/Redo

Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

Cut

Removes the selected text from the editor and places it into the clipboard.

Copy

Duplicates the selected text in the editor and places it into the clipboard.

Copy for Forum

Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

Copy as HTML

Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

Paste

Puts the contents of the clipboard at the cursor position, in the editor.

Select All

Selects and highlights the whole content of the editor.

Comment/Uncomment

Puts or removes the // comment marker at the beginning of each selected line.

Sketch

Verify/Compile

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

Upload

Compiles and loads the binary file onto the configured board through the configured Port.

Upload Using Programmer

This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn

Boot loader command must be executed.

Export Compiled Binary

Saves a .hex file that may be kept as archive or sent to the board using other tools.

Show Sketch Folder - Opens the current sketch folder.

Include Library

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

Add File...

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

Uploading:

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

4.3.2 ThingSpeak Cloud:

ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyze 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.

Features of Thingspeak:

ThingSpeak allows you to aggregate, visualize and analyze 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.

a)Visualize your sensor data in real-time.

b)Aggregate data on-demand from third-party sources.

c)Run your IoT analytics automatically based on schedules or events.

d)Prototype and build IoT systems without setting up servers or developing web software.

To learn how you can collect, analyze and act on your IoT data with ThingSpeak, explore the topics below:

Collect : Sends sensor data privately to the cloud.

Analyze : Analyze and visualize the data with MATLAB.

Act : Trigger a reaction.

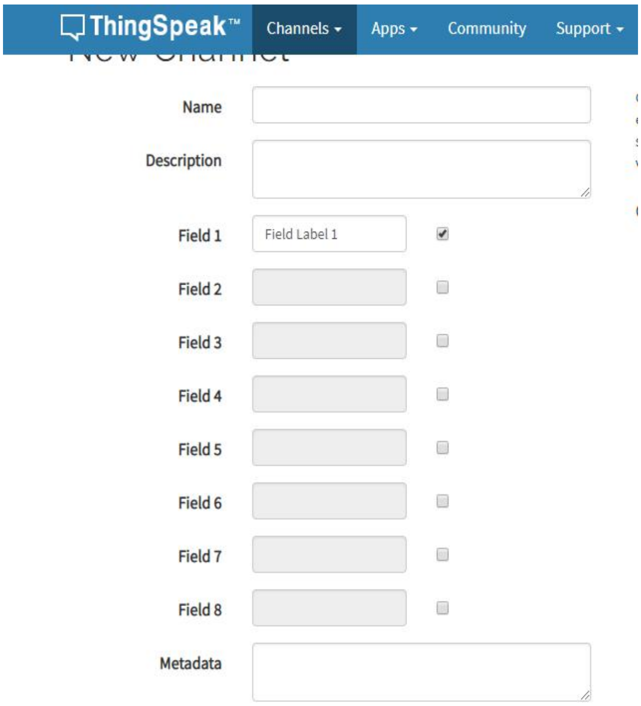

Fig 4.6: creation of fields needed for our project

Since we need to fields one for storing heart rate and the other for storing temperature in the fig 4.6 we will add two fields where name refers to name of the project.

ThingSpeak allows you to aggregate, visualize, and analyze live data streams in the **cloud**. **ThingSpeak** provides instant visualizations of data posted by your devices or equipment. Execute MATLAB code in **ThingSpeak**, and perform online analysis and processing of the data as it comes in.

ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. With ThingSpeak, your data is stored in channels. Each channel stores up to 8 fields of data. You can create as many channels as you need for your application.

ThingSpeak automatically charts the data that you send it, so you can remotely monitor your devices or equipment from anywhere. View your data from any web browser or mobile device. Share read-only views of your data with the clients and colleagues that you specify. Alternatively, you can use ThingSpeak to manage your data, and you can build your own front end for your clients and customers to log in to.

Let us see the visualization of data with MATLAB visualization after creating two fields. After connecting device the pulse rate and temperature data will be sent to cloud and the data is stored like in the fig 4.7

Fig 4.7: MATLAB Visualization after creation of fields

Apart from these we have to send the data to the mobile application.So for that purpose we used this Thingspeak cloud as a bridge between the working model and the mobile application. After creation of the fields Thingspeak will provide some features like API keys which are used to create a bond between the cloud and the mobile application. Those API keys are private and are know to the owner or the developer only. So it is highly secured to not undergo any attacks over the data. The fig 4.8 illustrates how API keys are generated after creation of the fields.

With a commercial ThingSpeak license, you can send data to ThingSpeak as fast as once every second. This not only enables near-real time monitoring of your devices, but it allows you to set up control loops from the cloud. For example, you could configure ThingSpeak to turn a light on when your motion sensor detects a person has walked into a room. For applications that require faster response times, the best practice is to have the control loop at the edge closer to the hardware.

The API keys in Fig 4.8 is used to get the data online from anywhere when it is connected to the network and retrieve the data from the sensors and send it remotely.

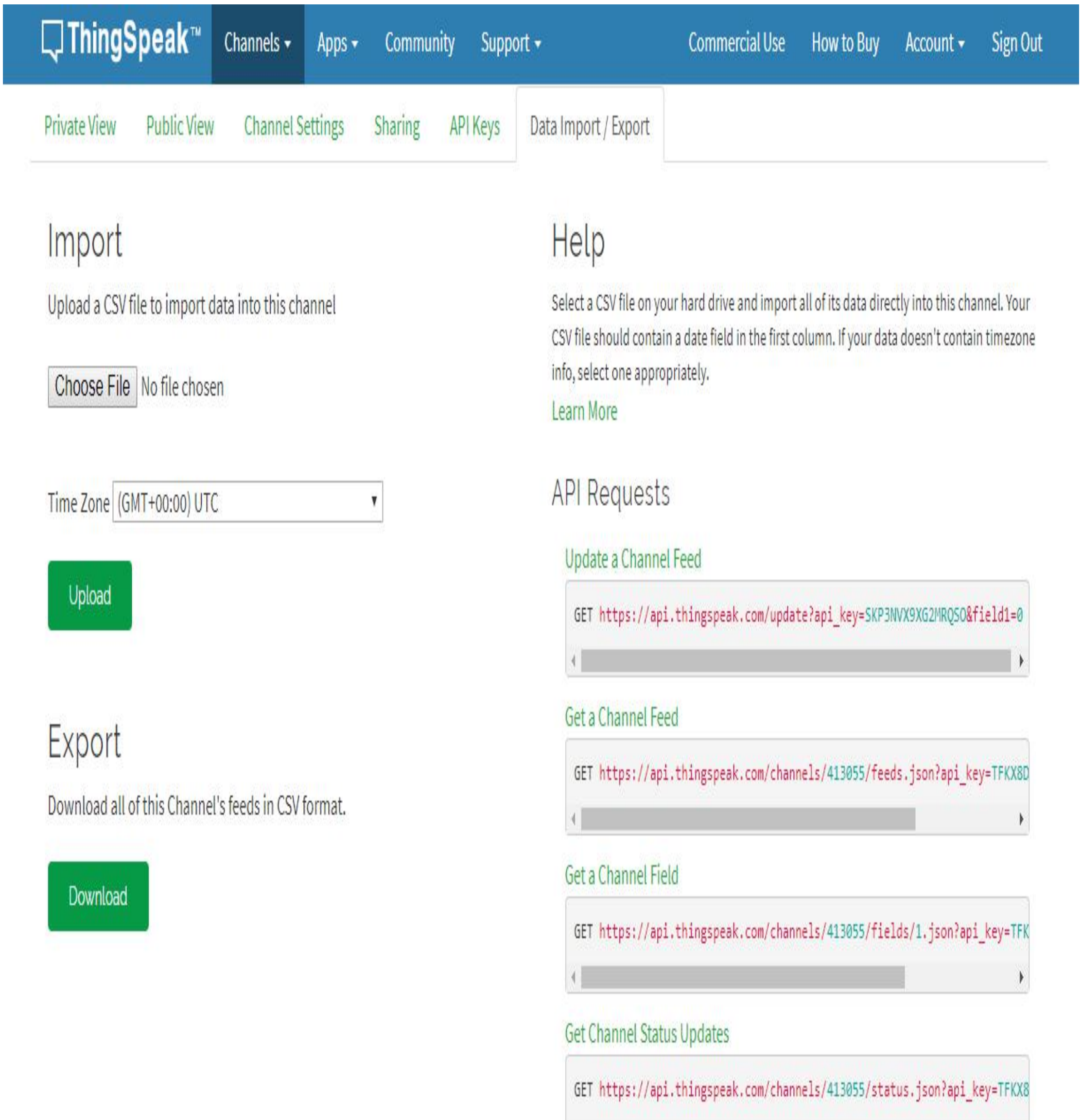

Fig 4.8: API keys Generation

4.3.3 MIT App Inventor:

App Inventor lets you develop applications for Android phones using a web browser and either a connected phone or emulator. The App Inventor servers store your work and help you keep track of your projects. This platform provides many tools that we will use regularly in a drag and drop manner. It consists of Designer and Blocks section. Designer section deals with front end which deals with the user interface whereas actual working will be done based on the Blocks section. Based on the utilities provided by the components the blocks section will be done and the API keys generated in the ThingSpeak cloud are used in blocks section to retrieve the data and show it on the user interface in the mobile application. MIT App Inventor consists of a palette where we will use them to develop a user interface. After completion of the application development we can build an apk file from the given options.

Designing of App:

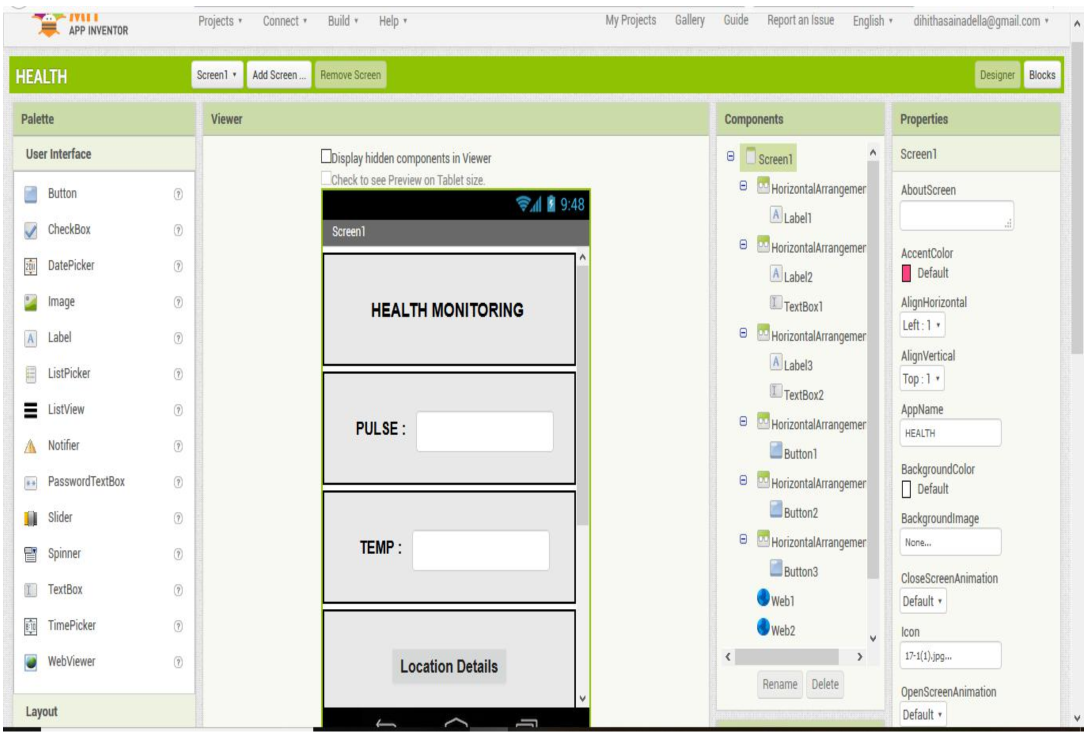

Fig 4.9: Front page design

The fig 4.9 and fig 4.10 illustrates how we designed the front end using the palette that was present at the left side of the figure. This palette consists of several layouts, sensors, media, maps, storage, social, connectivity, user interface etc and each of these has several things inside it which helps us to make the app easily. And in the right side we can modify the names, text, data etc. Fig 4.10 consists of web1, web2, clock1, sound1 at the bottom where these are things that we used to get connected to ThingSpeak cloud where the data is transferred remotely.

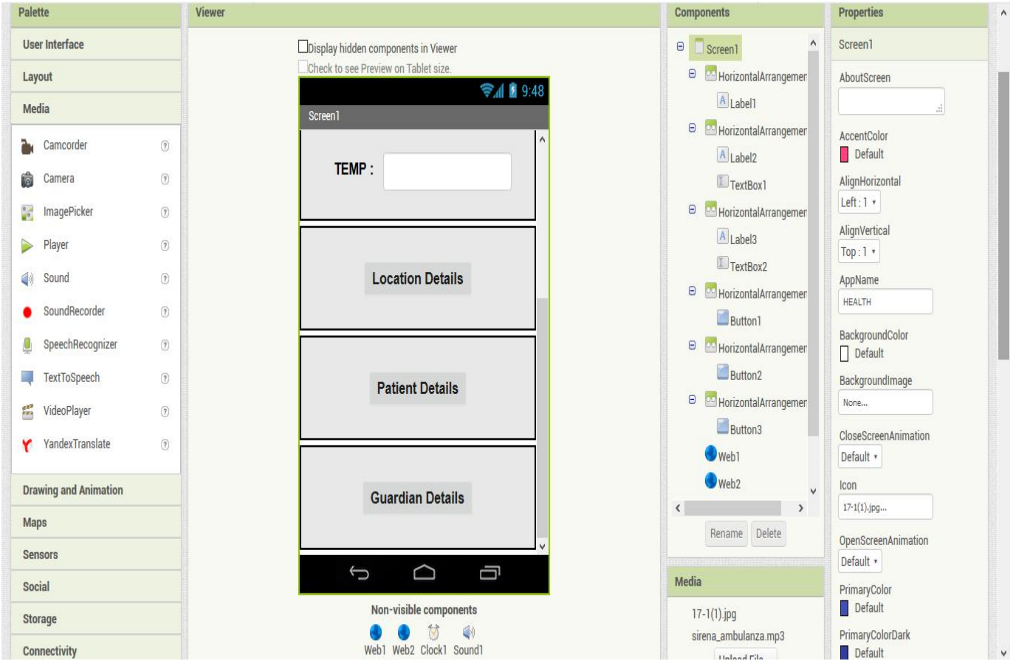

Fig 4.10: Completion of front end design

MIT App Inventor is an intuitive, visual programming environment that allows everyone – even children – to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes. And what's more, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments. The MIT App Inventor project seeks to democratize software development by empowering all people, especially young people, to move from technology consumption to technology creation.

Fig 4.11 illustrates the blocks mode of the front end design. In general when we click a button it will redirect to another page. So, to get that action we will do that in blocks mode where open another screen block will be given and in that we will pass the screen name that we want to open. For eg, if location button is clicked then ‘loc’ screen name is passed to open the ‘loc’ screen. Similarly with other buttons also we did the same. The sound, web1, web2, will be used here where web1 and web2 are used to get the data from ThingSpeak which provides blocks to pass the link. The sound is used to play a siren when the threshold exceeds.

Fig 4.11: Blocks section for front page

As the front page has to show the pulse value and temperature value as we said before the API keys generated in the ThingSpeak are used in blocks section in the above fig 4.11. As we that we had set some threshold to the pulse rate and if it's condition exceeds then a siren will automatically will play to intimate the guardian.

App Inventor is a cloud-based tool, which means you can build apps right in your web browser. **App Inventor** for Android **is** an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (**MIT**). It allows newcomers to computer programming to create software applications for the Android operating system . Because of it's in built blocks facility we can create many apps without any problem. Every single basic blocks that every app developer wanted are given in buit by them.

Now let's see the Location Details page. This page will open after clicking Location Details button. You can see that we named the screen as 'loc' which is used in the previous front page to open the screen when the button is clicked. In the fig 4.12 we dragged the location sensor into the screen. Then this sensor uses the mobile phone's location to track the patient. In the blocks page the location sensor provides several options like latitude and longitude and current address which we majorly used.

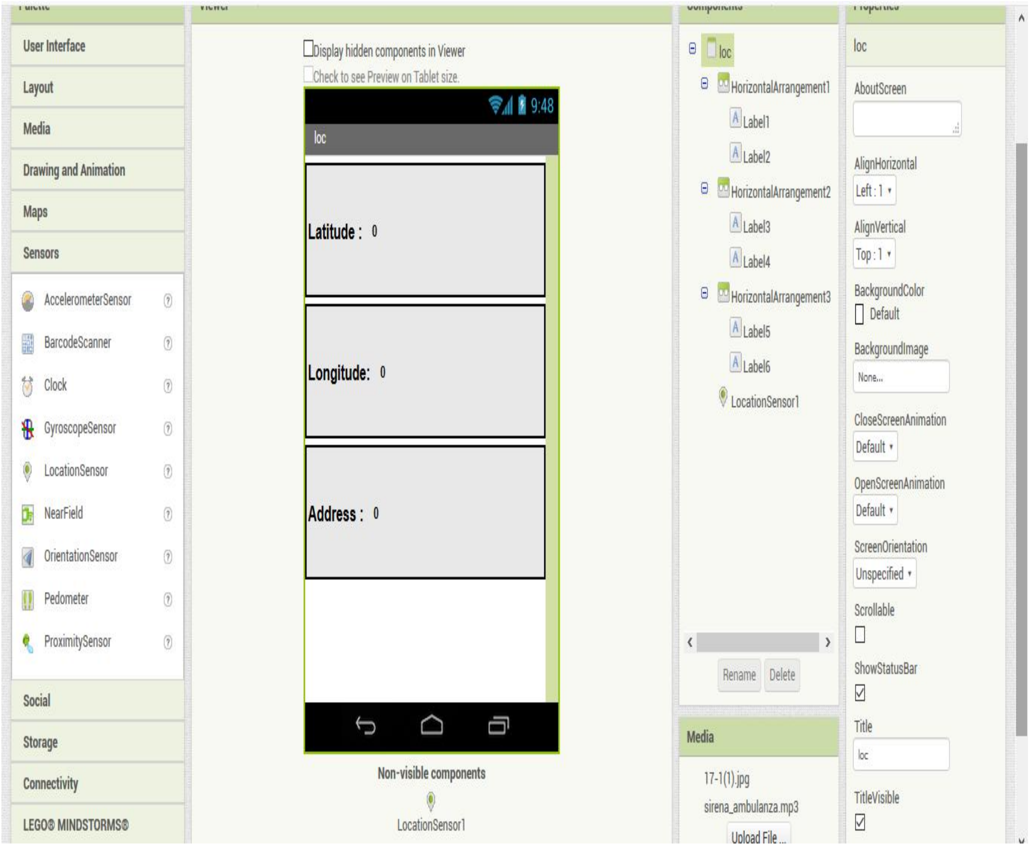

Fig 4.12: Location page

The LocationSensor component can determine the phone's latitude and longitude as well as a street address. You can use it to share your location with others, record "breadcrumbs" on a journey or treasure hunt, or as a way to take roll in class (as long as the students have Android devices!) by writing some conditions in the block section to take the attendance. These sensors provides us a flexible way to access the control over it.

As you see fig 4.13 there is set label text to block which is used to set the data or to update the data the obtained remotely and in fig 4.12 those labels are initialized to zero but later on the data will be updated based on the bocks section. Get latitude and get longitude will come by clicking latitude and longitude option in the location changed block. The location changed block is obtained after dragging the location sensor in the designer section. As you can see every designer has a block code to run, without that nothing will work properly.

The blocks section for the location page will be as follows:

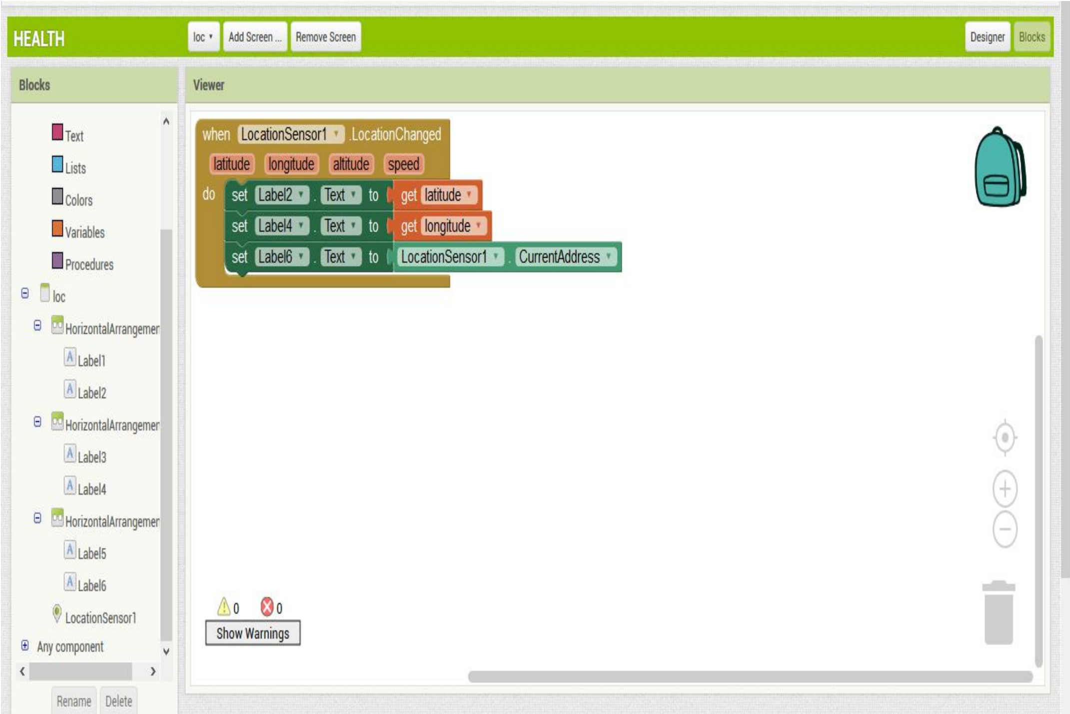

Fig 4.13: Location blocks

Get latitude, get longitude ,current address will be obtained from the location sensor component that was dragged in the location designer page.Similarly when patient details button was clicked then the fig 4.14 page will open.

Important thing is that the patient previous details and guardian details will change from person to person. In order to update these things the computer professional has to be there in the hospital. Fig 4.14 and 4.15 represents the patient details page in the designer page.

You can see that this page doesn't has any blocks section because the data has to be manually written since the patient details won't update without human interaction.

We can add any number of details in the patient details page. The more details we provide the most accurate treatment will be done to the patient and we can the blood group type of the person by adding a label and we said that all these details has to be taken care by the computer specialist to update for the first time while the app is going to be created.

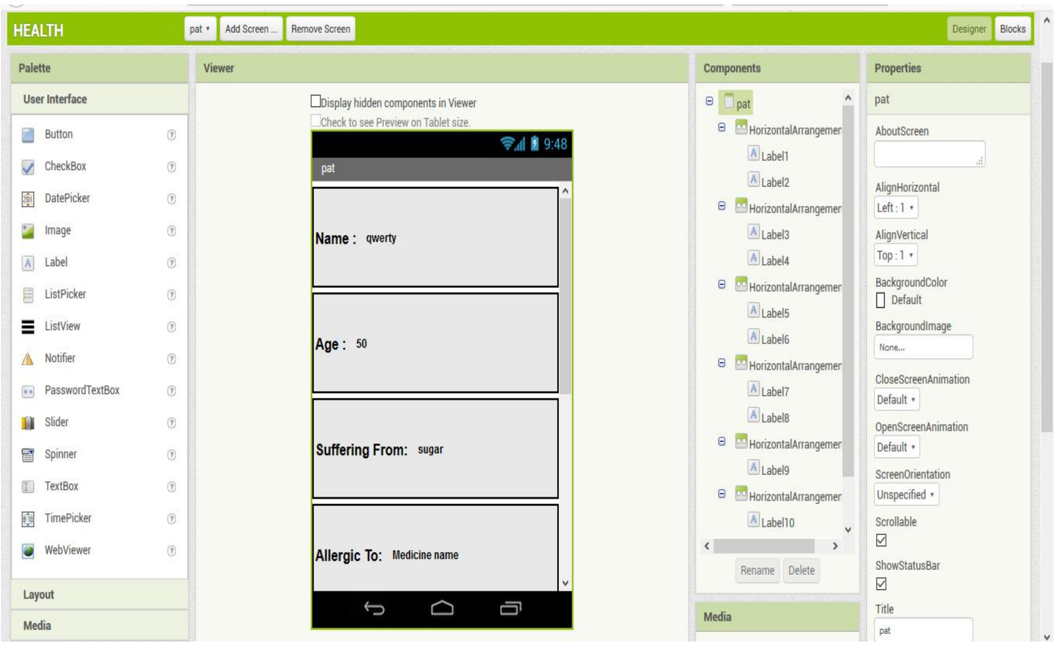

Fig 4.14: Patient details page

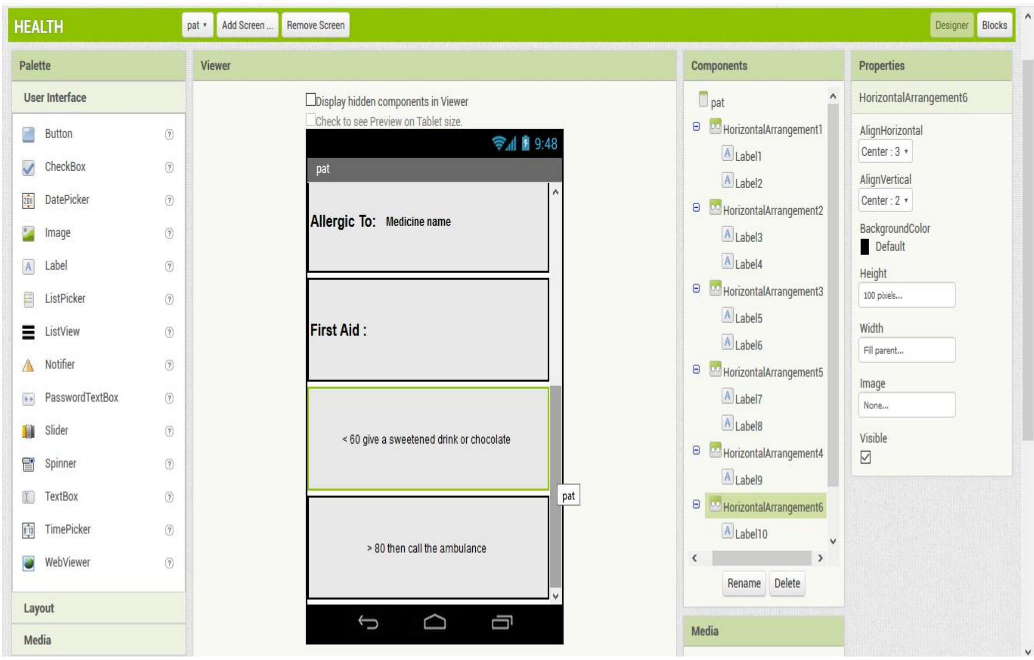

Fig 4.15: Patient details page

Now let us see the final page where it will show the guardians details after clicking the guardian details button. As you can see that there is a add screen button to add the screen and after clicking that it will ask to give the screen name which is used furthermore.

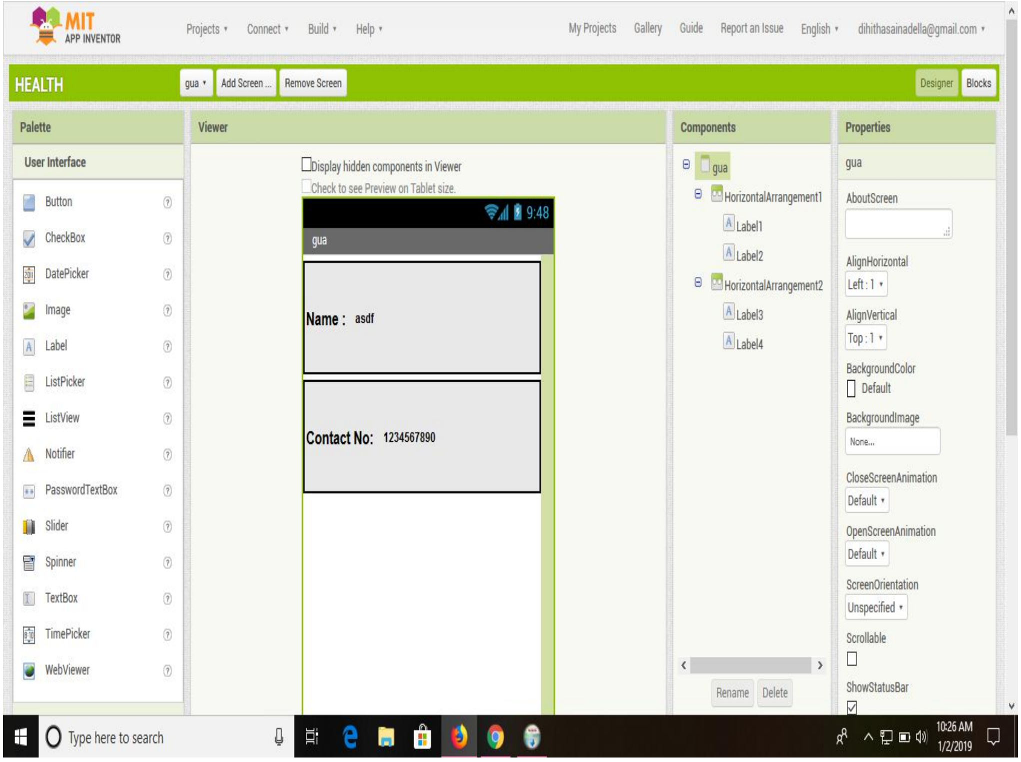

Fig 4.16: Guardian Details page

4.3.4 Arduino Programming:

Embedded C is an extension to C programming language that provides support for developing efficient programs for embedded devices.

Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows:

Embedded devices have resource constraints(limited ROM, limited RAM, limited stack space, less processing power)

Components used in embedded system and PCs are different; embedded systems typically uses smaller, less power consuming components. Embedded systems are more tied to the hardware.

Two salient **features of Embedded Programming** are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of languages:

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++, Java, Ada, etc.
- Application level language like Visual Basic, scripts, Access, etc.

CHAPTER 5

RESULTS

5.1 Displaying pulse and temperature values in App:

Fig 5.1 displays the pulse and temperature values in the application that was obtained from the ThingSpeak. Since in the implementation we showed how to connect ThingSpeak cloud to application. After establishing connection the values are getting updated time to time remotely. There is specific range because it will work over large distances also until it is connected to a network.

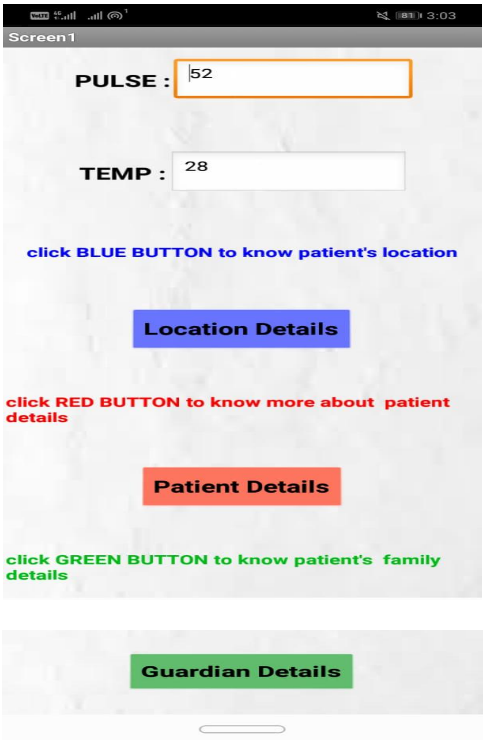

Fig 5.1: Values of pulse and temperature

5.2 Location of the patient:

When the alert sounds play then the person who is looking after the patient will click the location details button to know the patient’s location and the details will be displayed as shown in the fig 5.2

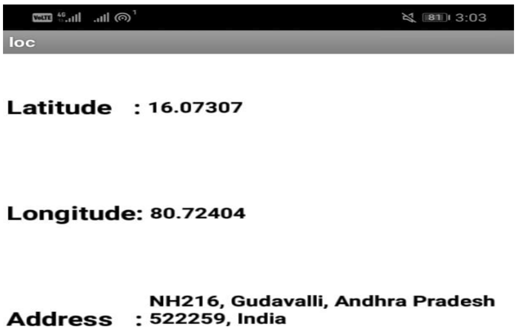

Figure 5.2: Location of the patient

5.3 Patient details display in App:

As we discussed the problem in the existed system that the patient's reports are stored in a hospital's database where the patient visits to check his/her health condition. The details are stored statically. It will be helpful if we make those details available remotely for better understanding of the patient to the doctor who maybe from any hospital. So we created a page where we can display the patient's previous records in the mobile application

When the patient went outside alone and collapsed then based on the alerting system the outsiders can join the patient to the nearest hospital and the patient's previous record will help the doctor to take care of the patient. This will help the doctor to treat him without any delay without taking some tests like blood group and the patient's allergic medicine and how was the condition of the patient previously.

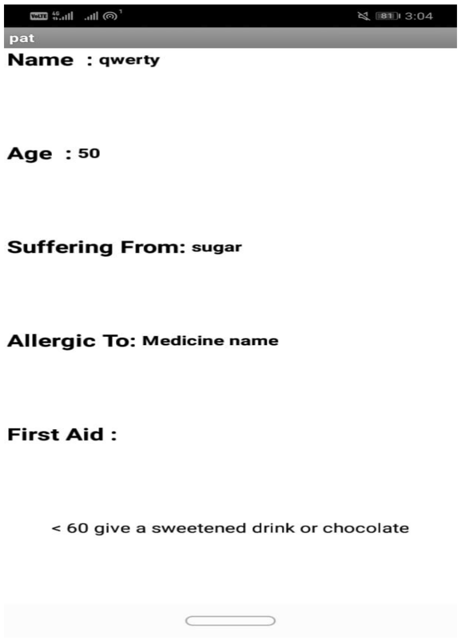

Fig 5.3: Patient Details

5.4 Guardian details display in App:

It displays the guardian name and phone number in case if the doctor has any queries over the patient can be contacted immediately by using this facility.

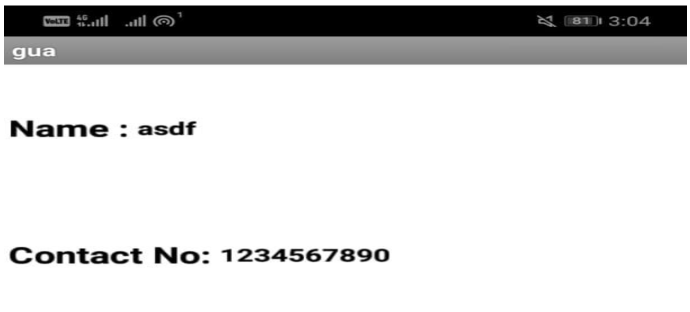

Fig 5.4 : Guardian’s Details

CHAPTER 6

TESTING

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application are working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of the application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum

6.1 Testing Levels:

There are various testing levels based on the specificity of the test.

a)Unit testing: Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.

b)Integration Testing: Integration testing is any type of software testing that seeks to verify the interfaces between components of a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

c)System Testing: System testing tests a completely integrated system to verify that the system meets its requirements.

d)Acceptance Testing: Acceptance testing tests the readiness of application, satisfying all requirements.

e)Performance testing: Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or devices such as response time or millions of instructions per second etc.

6.2 System Test cases:

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance with a specific requirement. I have designed and executed a few test cases to check if the project meets the functional requirements.

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
Does the values are updating in front page	-	Updating for every ten seconds	PASS
Does the values will update correctly without proper internet connection	-	Error 1101: Unable to get response from specified URL	FAIL
Will user get notified when sensors are failed	-	We can't get an alert at that time	FAIL

Table 6.1: A test case for values updation

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
Does the location is updating well	After clicking location details	Updating for every five seconds	PASS

Table 6.2:A test case for location updation

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
User is currently on patient details page	User clicks on patient details button	Directs to patient details page	PASS

Table 6.3 : A test case for navigating to patient details page

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
User is currently on guardian details page	User clicks on guardian details button	Directs to guardian details page	PASS
Will it pop any error message while entering wrong number of digits in mobile number	While entering mobile number other than 10 digits	It accepts it and displays no error	FAIL

Table 6.4 : A test case for navigating to guardian details

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
Does the temperature sensor will give accurate result when a person goes to extreme cold or hot places	-	Our used sensor will not provide certain accurate result at that situation	FAIL

Table 6.5 : A test case over the working of temperature sensor

6.3 Performance Testing:

Performance of the system can be determined based on the system/application responsiveness under all kinds of load. Performance testing in IoT framework is little different than traditional performance testing. IoT devices generate a lot of data which is saved in server and analyzed for immediate decisions. Hence IoT system must be built for high performance and scalability. And to measure these two key attributes, it is important to understand the business value for which it is build i.e. in our case patient health data. Hence it is necessary to simulate real world models, network conditions etc.

CHAPTER 7

CONCLUSION & FUTURE WORK

7.1 Conclusion:

In this project monitoring the patient especially in the rural region even the doctor is not available physically. This will be very useful to analyze the patient continuously using mobile application and location of the patient is updated dynamically to know where the patient has collapsed. The patients previous records, medicinal details and guardian details are also added as we mentioned which helps the doctor for fast treatment without testing basic tests like blood group, temperature, pulse rate etc. Moreover it is cost effective and reduce casualties is a biggest advantage by frequently monitoring the patient were predicted .

7.2 Advantages of proposed system

a)Cost efficient.

b)Time to time monitoring of health.

c)Number of visits for checkup decreases.

d)A portable remote health monitoring Internet of things in the medical field brings out the solution for effective patient monitoring at reduced cost and also reduces the trade-off between patient outcome and disease management.

e)Patients previous records helps doctor for better and accurate treatment.

f)It helps to track the patients location time to time.

7.3 Future work :

Our project is depending on the guardian to intimate the ambulance based on the obtained patient’s location in the app. This can be improved by directly intimating the nearby hospital and some other developments can be made in this and some of them are:

a)Brain Signal Monitoring

b)Tumour Detection

c)Accident detection

d) Obtaining nearby hospital details and directly intimating the hospital staff about the patient's condition based on which they will send ambulance to that place.

REFERENCES

[1] Ayush Bansal , Sunil Kumar, Anurag Bajpai, Vijay N. Tiwari, Mithun Nayak, Shankar Venkatesan, Rangavittal Narayanan, “Remote health monitoring system for detecting cardiac disorders”, IET Syst. Biol., 2015, Vol. 9, no. 6, pp. 309–314.

[2] Walinjkar, A., Woods, J., Personalized wearable systems for real-time ECG classification and healthcare interoperability: Real-time ECG classification and FHIR interoperability, In Internet Technologies and Applications, 2017, vol. 6, no. 8, pp. 9-14

[3]Abdullah, A., Ismael, A., Rashid, A., Abou-ElNour, A., Tarique, M., Real time wireless health monitoring application using mobile devices, International Journal of Computer Networks & Communications, 2015, vol. 7, no.6, pp. 13-30.

[4] Chaudhary, R., Jindal, A., Aujla, G. S., Kumar, N., Das, A. K., Saxena, N., LSCSH: Lattice-Based Secure Cryptosystem for Smart Healthcare in Smart Cities Environment, IEEE Communications Magazine, 2018, vol. 2, no. 6, pp. 24-32.

[5] S. Jaiswal, R. Katake, B. Kute, S. Ranjane, and P. D. Mehetre, “Survey of Health Monitoring Management Using Internet of Things (IOT),” Int. J.Sci. Res.,2017, vol. 5, no. 11, pp. 2243–2246.

URL's

i) https://www.elprocus.com/heartbeat-sensor-working-application/

ii) https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

iii) https://thingspeak.com/login

iv) http://ai2.appinventor.mit.edu/

v) https://www.codeproject.com/Articles/1073160/Programming-the-ESP-NodeMCU-with-the-Arduino-IDE

vi) https://www.arduino.cc/en/Tutorial/BuiltInExamples

APPENDIX

PulseSensor.ino:

```
#include <SoftwareSerial.h>
#include <DHT.h>
#define DHTPIN 6
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
SoftwareSerial cloud(9, 10);
int ledpin=7;
int pulsePin = A0;
volatile int BPM;
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;
String data;
int t;
void setup()
{
  pinMode(ledpin,OUTPUT);
  Serial.begin(9600);
  interruptSetup();
  cloud.begin(9600);
  dht.begin();
  delay(10);
}
```

```
void loop(){
 delay(1000);
 t = dht.readTemperature();
 if (QS == true)
 {
    digitalWrite(ledpin,HIGH);
    Serial.print("HeartBeat=");
    Serial.println(BPM);
    Serial.print("Temperature=");
     Serial.println(t);
    cloudPrint();
    QS = false;
  }
  else {
    digitalWrite(ledpin, LOW);
  }
 delay(20);}
void cloudPrint()
{
  data = "@" + String (BPM) + "+" + String (t);
  if (BPM>55 && BPM<85)
  {
  Serial.println(data);
  cloud.print(data);
  }
}
```

Interrupt.ino:

```
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P =512;
volatile int T = 512;
volatile int thresh = 525;
volatile int amp = 100;
volatile boolean firstBeat = true;
volatile boolean secondBeat = false;
void interruptSetup(){sei();}
ISR(TIMER2_COMPA_vect){
  cli();
  Signal = analogRead(pulsePin);
  sampleCounter += 2;
  int N = sampleCounter - lastBeatTime;
  if(Signal < thresh && N > (IBI/5)*3){
    if (Signal < T){
      T = Signal;
    }
 }
  if(Signal > thresh && Signal > P){
    P = Signal;
  }
  if (N > 250){
    if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
      Pulse = true;
          IBI = sampleCounter - lastBeatTime;
```

```
lastBeatTime = sampleCounter;
   if(secondBeat){
     secondBeat = false;
     for(int i=0; i<=9; i++){
       rate[i] = IBI;
     }
   }
   if(firstBeat){
     firstBeat = false;
     secondBeat = true;
     sei();
     return;
}
   word runningTotal = 0;
   for(int i=0; i<=8; i++){
     rate[i] = rate[i+1];
     runningTotal += rate[i];
   }

   rate[9] = IBI;
   runningTotal += rate[9];
   runningTotal /= 10;
   BPM = 60000/runningTotal;
   QS = true;
  } }
 if (Signal < thresh && Pulse == true){
  Pulse = false;
  amp = P - T;
```

```
  thresh = amp/2 + T;
    P = thresh;
    T = thresh;
  }
  if (N > 2500){
    thresh = 512;
    P = 512;
    T = 512;
    lastBeatTime = sampleCounter;
    firstBeat = true;
    secondBeat = false;
  }
  sei();
}
```