REAL-TIME SUPPLY CHAIN MONITORING SYSTEM

minor project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

By

AMIT KUMAR YADAV	**(17UECS0048)**
BITTU KUMAR PRASAD	**(17UECS0116)**
B. SHIV KUMAR	**(17UECS0844)**

Under the guidance of
K. KISHORE KUMAR., M.E.,Ph.D.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
VEL TECH RANGARAJAN Dr.SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University Estd u/s 3 of UGC Act, 1956)
CHENNAI 600 062, TAMILNADU, INDIA

May, 2020

CERTIFICATE

It is certified that the work contained in the project report titled "REAL-TIME SUPPLY CHAIN MONITORING SYSTEM " by "AMIT KUMAR YADAV (17UECS0048) BITTU KUMAR PRASAD (17UECS0116) B. SHIV KUMAR (17UECS0844)" has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor

K. Kishore Kumar

Assistant Professor

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr.Sagunthala R&D

Institute of Science and Technology

May, 2020

Signature of Head of the Department

Dr. V. Srinivasa Rao

Professor & Head

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr.Sagunthala R&D

Institute of Science and Technology

May,2020

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

AMIT KUMAR YADAV

Date: / /

(Signature)

BITTU KUMAR PRASAD

Date: / /

(Signature)

B. SHIV KUMAR

Date: / /

APPROVAL SHEET

This project report entitled REAL-TIME SUPPLY CHAIN MONITORING SYSTEM by AMIT KUMAR YADAV (17UECS0048), BITTU KUMAR PRASAD (17UECS0116), B. SHIV KUMAR(17UECS0844) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

K. Kishore Kumar., M.E.,(Ph.D.)

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO). DSc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. V. S. S. Kumar, Ph.D.,** for providing us with an environment to complete our project successfully.

We obligated to our beloved **Registrar Prof. Dr. E. KANNAN,** for providing immense support in all our endeavors.

We thankful to our esteemed **Director Academics Dr. ANNE KOTESWARA RAO, Ph.D.,** providing a wonderful academic environment to complete our project successfully.

We record indebtedness to our **Dean & Head, Department of Computer Science and Engineering Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. K. RESHMA, M.E., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We also take this opportunity to express a deep sense of gratitude to Our Internal Supervisor **K. KISHORE KUMAR., M.E.,PH.D.,** for her cordial support, valuable information and guidance, he helped us in completing this project through various stages.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

AMIT KUMAR YADAV (17UECS0048)
BITTU KUMAR PRASAD (17UECS0116)
B. SHIV KUMAR (17UECS0844)

ABSTRACT

In this Project we are developing a real-time supply chain monitoring device which can provide supply managers and clients with live data about temperature, humidity and location of the product being supplied, mitigating any problems along the chain before they arise. This project enables remote accessing and monitoring of the status of the product in between supply chains and can be checked for any faults in the supply chain by the live data provided. Internet of Things (IoT) concept is being used to design and develop the project and We are using an IoT-cloud database service to receive and organize all the sensory live data sent from the edge layer after, as this data from the edge layer is raw, we used some data extraction algorithms to extract necessary data for sake of less transmission cost and less latency. This real time data from the cloud database would be further interfaced virtually through a telegram chatbot which make uses of data extraction algorithms from json files from the cloud database server APIs. This project assumes to full fills the need of a low cost, compact, real time monitoring system. In a supply chain, the status of the product remains hidden and mostly human dependent, where this project comes in handy. This can be made use in industries like pharmaceutical, chemical and food industries etc. This project would further help in decision making, validating the stock, delivering delicates etc

Keywords: supply chain, json, chatbot, APIs, cloud server

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

UML	unified modeling Language
HT	humidity temperature
CDB	cloud data base

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

A sensetive supply chain denotes a series of action or equipment applied to maintain product at a specific temperature, pressure and humidity. It is used to preserve and extend the self life of a product. From the time it is manufactured till the time it is utilized a specific constant temperature is maintained, Unlike goods or merchandise supply chain are perishable when exposed to air. but sensetive supply chains like cold chains are common in food and pharmaceutical companies and some of the chemical industries. A cold chain can be managed by the quality of management system. It should be analyzed, controlled, documented and validated.

There are numerous problems related to supply chain which are costly and severe and are irrefutable. In supply chain of pharmaceutical products like vaccine or sensitive medicines, the slight variations of temperature and humidity degrades the whole product, which might in turn result in a poisonous degradation, this degradation is usually not detectable without clinical testing, so in order to keep track of this sudden hikes in temperature and humidity, we need a real time monitoring system. The same applies for the chemical industries, which involves transporting of sensitive chemicals all along the supply chain.

Sensitive supply chain planning and control is the act of facilitating an efficient and effective use of resources and assets, while producing and delivering products according to market demand and the customers' requirements . The aim is to decide what and how much to produce and deliver when and where; and also how products and information should flow throughout the supply chain. From a supply chain planning and control view this includes activities such as overall resource planning, sales and operations management, demand and supplier management, performance 3 measurement and management, and event management, in addition to more operative activities for capacity and material planning and shop floor scheduling systems.

1.2 Aim of the project

To develop a product based solution for supply chain problems which can record, monitor report and ensure the temperature and humidity conditions that the supply chain is phasing through. This project aims for efficient supply of vaccines, blood samples, foods, and sensitive chemicals. Which could be very fatal and critical if handled improperly.

1.3 Project Domain

Our project falls under the domain of IOT as it consists of sensors at the end points, which are further connected with a cloud server, whose data will be retrieved, analysed, and is reported via UI.

1.4 Scope of the Project

Management of supply chains can be a complex operation, especially in supply chain of sensitive chemical and pharmaceutical and food based supplies. Our project tries to involve an oversight of a temperature and humidity controlled supply chain, typically used to monitor and ensure the appropriate conditions all along the supply chain.

1.5 Methodology

This project make uses of totally three modules, two of them being the end points modules i.e the edge module which consists of condition sensors like temperature and humidity sensor, and the other end point consists an UI which is responsible for interacting with the owner or anyone who is concerned of.IN our project, we have made used of telegram chat bot as an UI for the sake of convenience and simplicity. the other module is the server-cloud-server module, which involves a server through which the edge module interacts with the cloud database service through a specific API, and another server which retrieves this data and send to the processing module through APIs in form of JSON files. this module also consists of the data extracting algorithms responsible for data extraction from the JSONs files. After appropriate data extraction this data is represented to the user through the UI.

Chapter 2

LITERATURE REVIEW

Features of supply chain management are in existence from over last few eras whether it's purchasing or any of logistics activity however, the term supply chain management entered the public domain when Keith Oliver, a consultant at Booz Allen Hamilton [1] used it in an interview for the Financial Times in 1982. It gained admiration and acceptance in the mid-1990s, when a number of articles and books were published on the subject. Supply chain research Group at the University of Tennessee defines supply chain Management as the systematic strategic co-ordination of multiple business functions and strategies within a particular company across all business functions in and across an organisation, with the objectives of improving the long-term performance of the individual companies and the supply chain as a whole. [2].

A supply chain as the network of organisations that are involved, through the upstream and downstream linkages in different processes and activities that produces value in the form of products and services in the hands of ultimate customers.

A customer-focused definition is given by Hines "Supply chain strategies require a total systems view of the links in the chain that work together efficiently to create customer satisfaction at the end point of

delivery to the consumer[2].
As a consequence costs must be lowered throughout the chain by driving out unnecessary expenses, movements, and handling. The main focus is turned to efficiency and added value, or the end-user's perception of value." [3]

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

In India, the sensitive supply chains like the one of pharmaceutical and food industries are still handled on human basis. Where one person checks the supply product through a machinery at regular intervals, which can monitor HT hikes up to only a particular extent and aren't mostly precised, which results in failure of the traditional human based monitoring system.

3.2 Proposed System

in order to overcome this problems of existing systems, related to precision and accuracy, we have proposed an IoT based real time supply chain monitoring system, which monitors the entire supply chain through all the phases of its transmission and with real time system interaction it is able to detect even slightest of temperature and humidity hikes in the system, and is able to store the status of the system at periodic intervals in a cloud serviced data base through unique application

program interface(API), this data stored in the cloud server(in our case we are using an open source cloud server provided by Thingspeak), this stored data is sent to the processing module via application program interface(API) in form of java script object notion files(JSON), this data is retrieved via using data extraction functions and is then analysed for threshold frequency, the moment a threshold frequency is predicted to reach, the processing modules make uses of the telegram chat bot object and initiates a reporting message through telegram's API which is unique to each chat interface. The quality and latency time will be core features to depict the efficiency of the proposed supply chain monitoring system.

Advantages

the proposed system is highly sensitive to the smallest hikes in temperature and humidity changes and it comprises of minimal latency in reporting a threshold, and saves the cost of human monitoring.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The real time supply chain monitoring system is very much feasible in terms of economy, It is developed in consideration to keep the cost as minmum as possible. The hardware used is the system is of low cost and easily available, and the software uses cloud to store data which reduces the cost of memory storage. It is developed with the goal of reducing the cost so that cold chain can be utilised by the middle class family for disinfecting their food and for preserving foods and other perishable item at a very low cost . Smart cold chain at low cost

can reduce the pharmaceutical charges for distributing vaccines, blood ,organs and tissues.

3.3.2 Technical Feasibility

The main challenge of the REAL TIME CHAIN MONITORING SYSTEM is to maintain the desirable temperature as per requirement, this is however solved by using a bot which would send an immediate alert message to the manager if it recognizes any glitch within the system. The sensor used within the system can resist high tempetaure and pressure and furthermore it does not alter the cold chain as in case of chemicals which can be highly reactive.

3.3.3 Social Feasibility

Smart Cold Chain is widely excepted by the managers , as it reduces the risk of contamination and it preserves the cold chain at specified temperature and pressure. It has reduces the work load of checkng the cold chain on regular basis as an alert message is sent to manager which is quite simpler and easier as compared to the traditional way of monitoring cold chain.

3.4 System Specification

3.4.1 Hardware Specification

- temperature and humidity sensor - DHT11 embedded module(3.5-5.5V, 20 90H, 0 50*C)
- micro-controller - Arduino UNO(6-20V, 40mA)

- bread board/ PCB - 6 x 6 (mini)
- nodeMCU - for connectivity

3.4.2 Software Specification

- Thingspeak cloud DB - up to 4 channels
- service interval - 1 per 15 seconds
- JSON file limit - 5 MB/file
- chat processing limit - 1/15

3.4.3 Standards and Policies

In terms of standardisation organisation ISO, the longest establishef founded in 1996 by joint commitee with IEC at standardisation for DHT11 technology.
The ISO DTH11 makes variety of categories with aspect of DTH11 Which is addressed. It takes reading of Temperature and Humidity and other associated portocols, data content formatting, conformance testing and varios other small fields.
In addition to the DHT11 . In 1999 variety of commercial companies has set a consotium to research and standardise DHT11 technology.In 2003 organisation was split with abundant standardization technique under the brand new entity called EPC global.
The node MCUis a support for ESP32 32-bit MCU. It was Developed by ESP8266 Open Source community, it costs approximately 5 usd storage can range upto 4M bytes. Its firmware for open Source prototyping board designs.

Chapter 4

MODULE DESCRIPTION

4.1 General Architecture

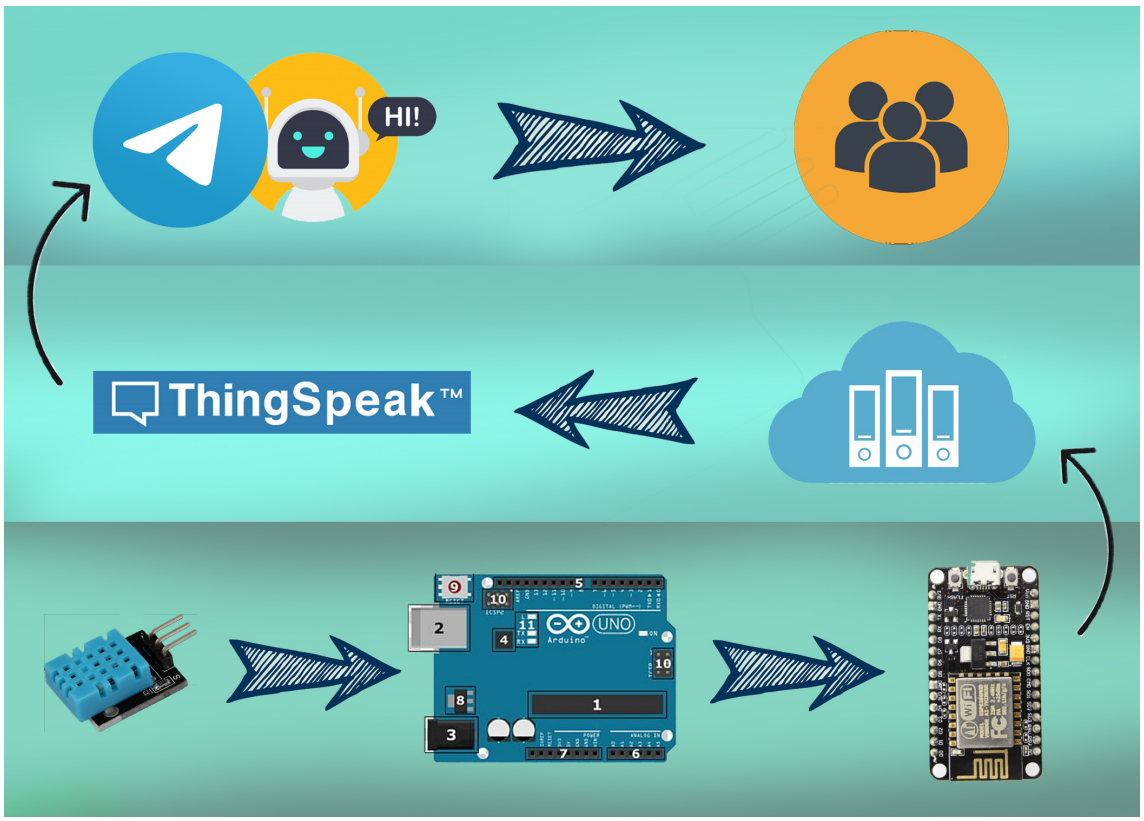

Figure 4.1: **Architecture Diagram**

abstract architecture of the multiple modules interacting together, where each layer demonstrates each modules and each segment implies the integration of one modules with another.

4.2 Design Phase

4.2.1 Data Flow Diagram

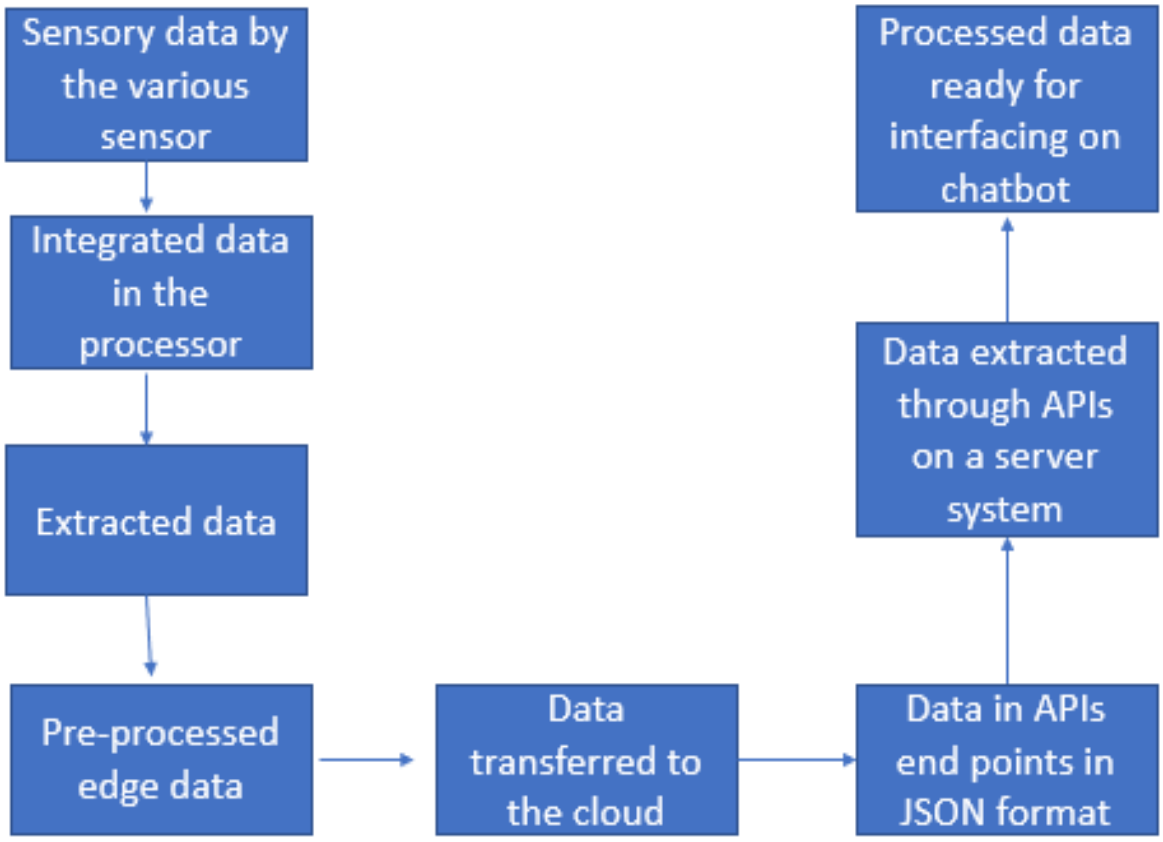

Figure 4.2: **Data Flow Diagram**

The above data flow diagram of the proposed system demonstrates the data flow of sensory data, data formats, the type of processing and the way data are fetched. As it is clear the sensory data which is the raw data is processed at the edge layer and is sent to the cloud DB via API and the data stored in the cloud DB is fetched through API in the form of JSON data sheet, which in turns is parsed in the processing module which also consists of the chat bot code interacts with the UI through telegram get/post requests and present the information from the data fetched.

4.2.2 UML Diagram

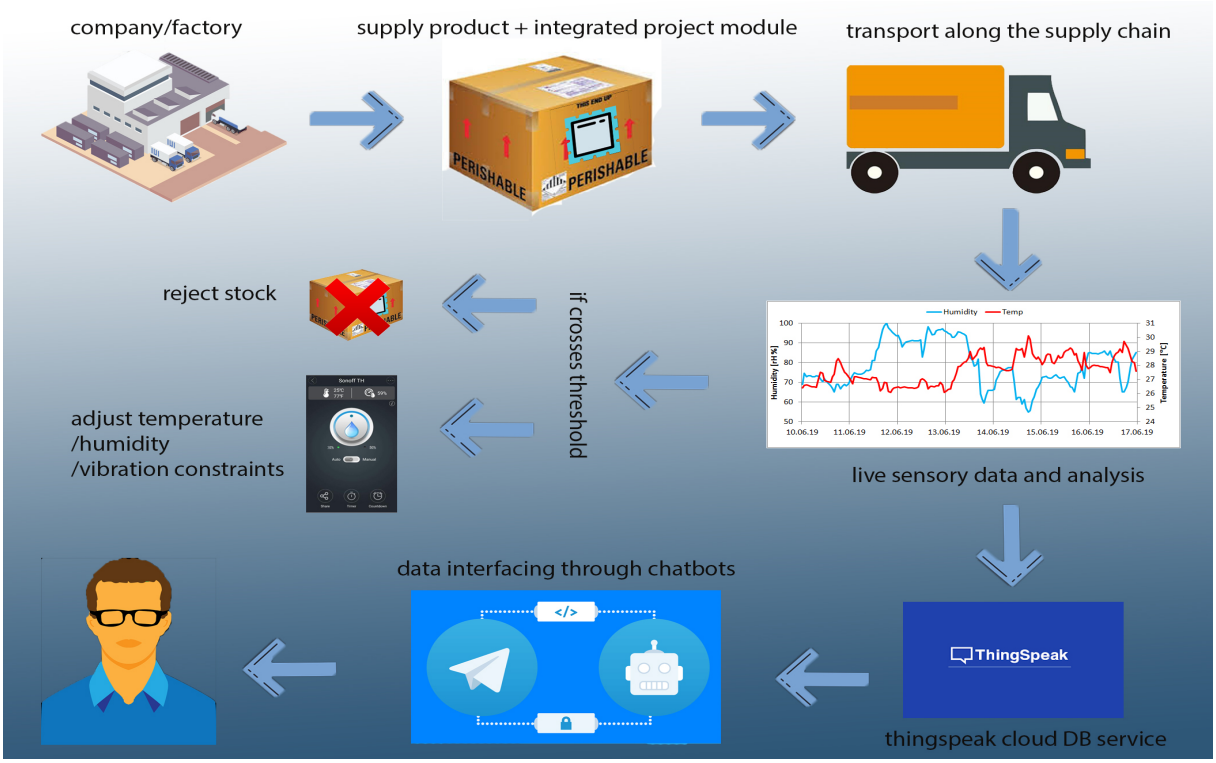

Figure 4.3: **UML Diagram**

UML diagram i.e unified modelling language diagram represent the whole proposed system with involved agents and factors inclusive of the owner, supply chain storage environment and other social and environmental entities involved in the whole process of the supply chain. In this UML diagram, we can clearly see the initialisation of supply chain by the company and the storage box which consists of the product to be concerned about, which is embedded by a small system, which is the proposed module and the way it interacts with the supply chain and the pathway, by which it interacts/ reports the owner/ user.

4.2.3 Use Case Diagram

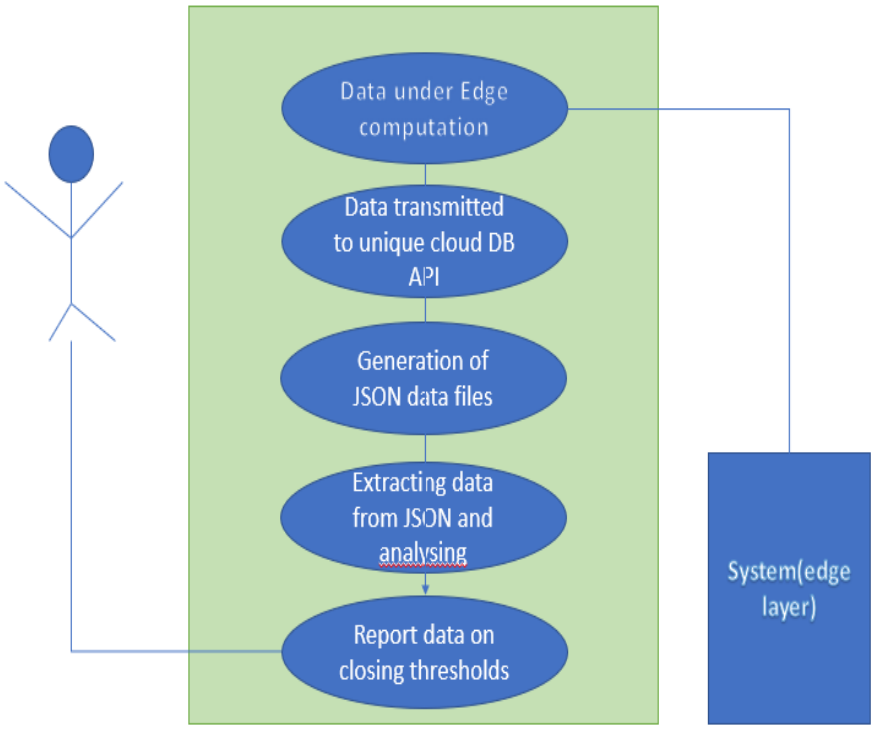

Figure 4.4: **Use Case Diagram**

A use case diagram is a simplest representation of a user's interaction with the whole system that presents the relationship between the user or the one who is concerned and the different use cases in which the user is involved. And in our proposed system the user is only involved at the end point, as the proposed system is a single flowed model, in which modules function in series one after another, therefore user interacts with the whole system on a result basis, where he/she is reported, when a threshold is approaching.

4.2.4 Class Diagram

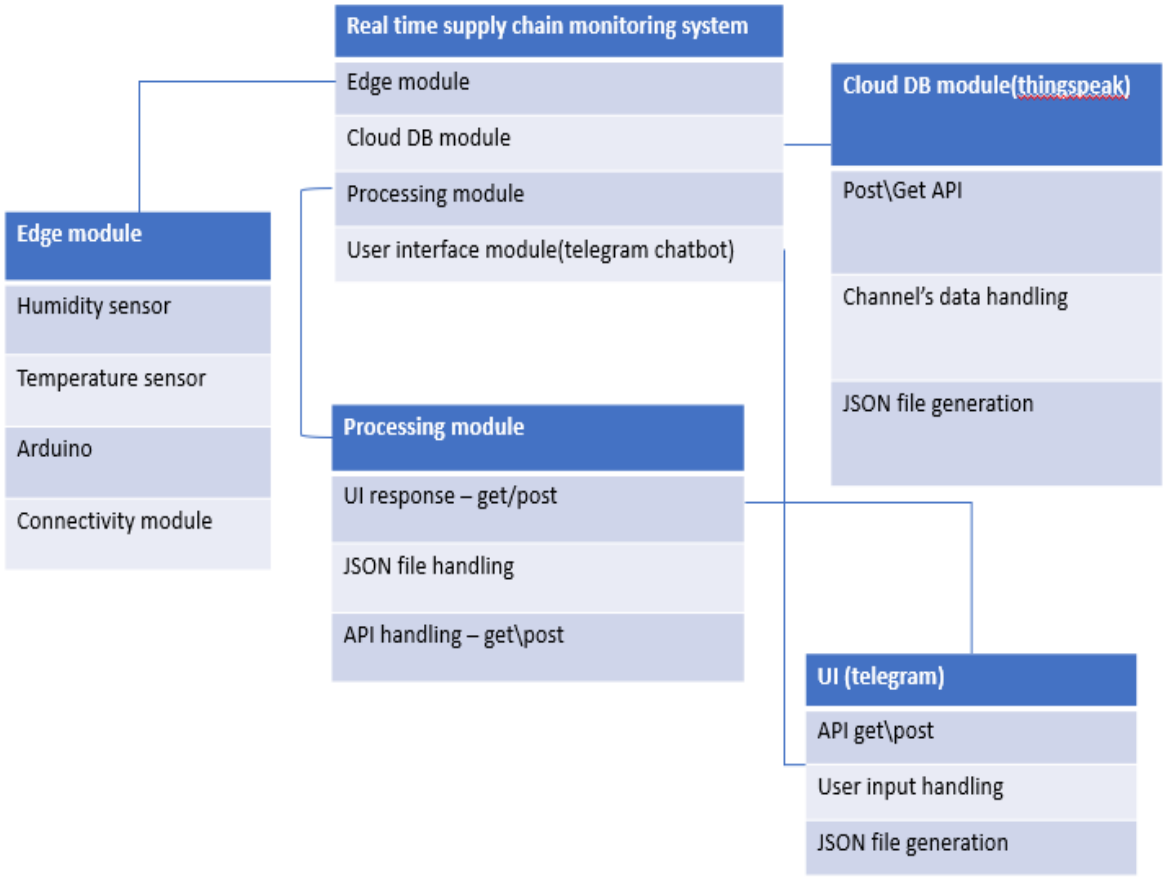

Figure 4.5: **Class Diagram**

The class diagram of the proposed system is the main building block of its object-oriented modeling. which has been used for general prospective and conceptual modeling of the structure of the use cases, and thereby translating the whole model to a program code. In this report, we have used the class diagram for data modelling and code construction.

4.2.5 Sequence Diagram

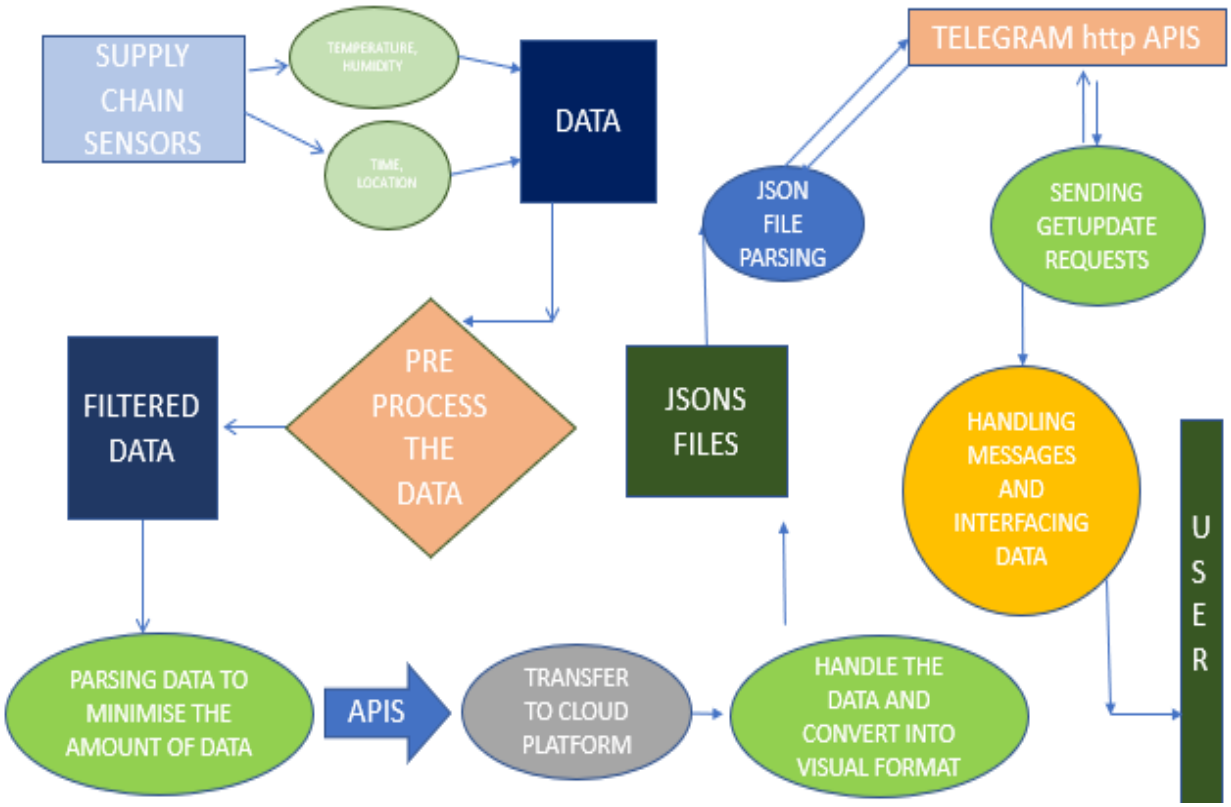

Figure 4.6: **Sequence Diagram**

The sequence diagram of the proposed system shows the object interactions by a sequencing model. The diagram depicts the objects and classes involved in the whole system and the scenarios involving it and the sequence of response and data exchanged between the objects needed to carry out the functionality of the scenario.

4.3 Module Description

.

Module 1: Real time sensory data collection and edge computation
Module 2: Real Time data gathering and prognostics via JSON files
Module 3: Real Timestatus reporting via a Telegram Chat-Bot

4.3.1 Module 1: Real time sensory data collection and edge computation

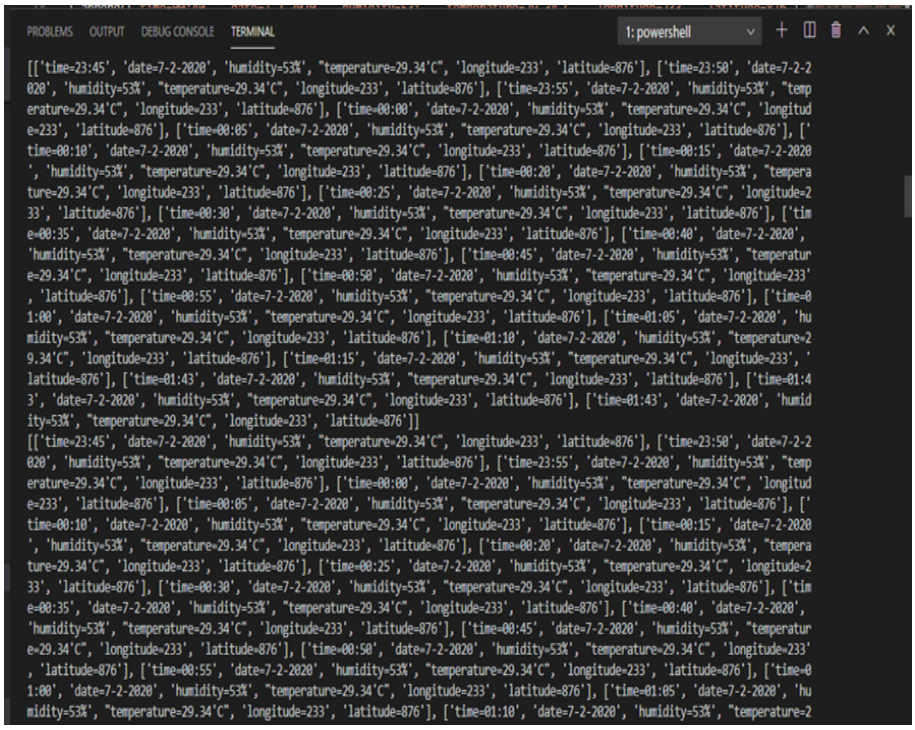

Figure 4.7: **sensory raw data**

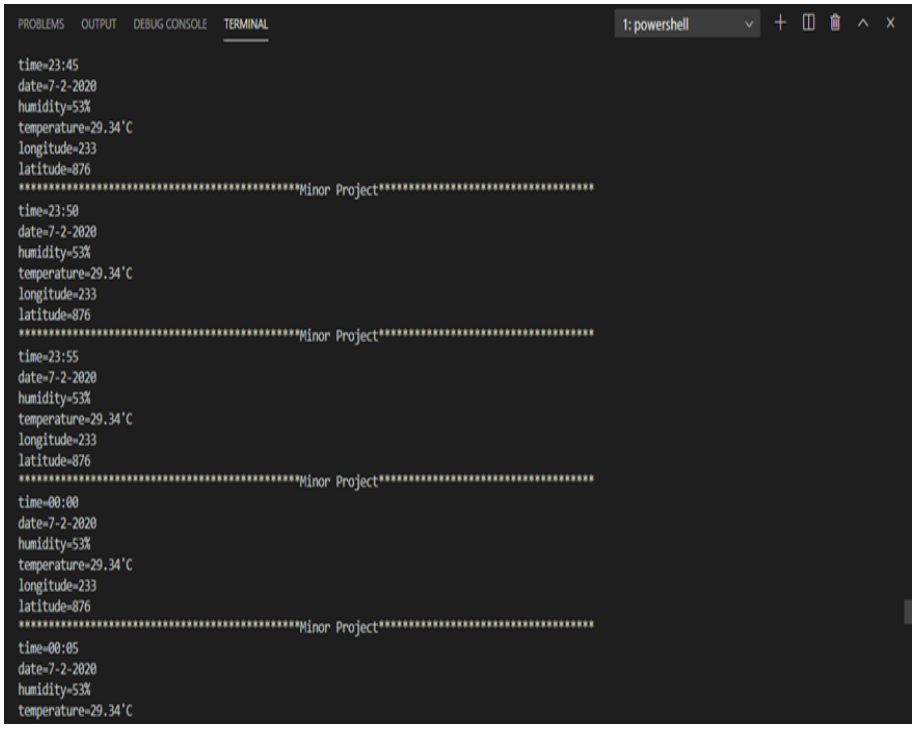

Figure 4.8: **edge processing of real time local data**

4.3.2 Module 2: Cloud DB service

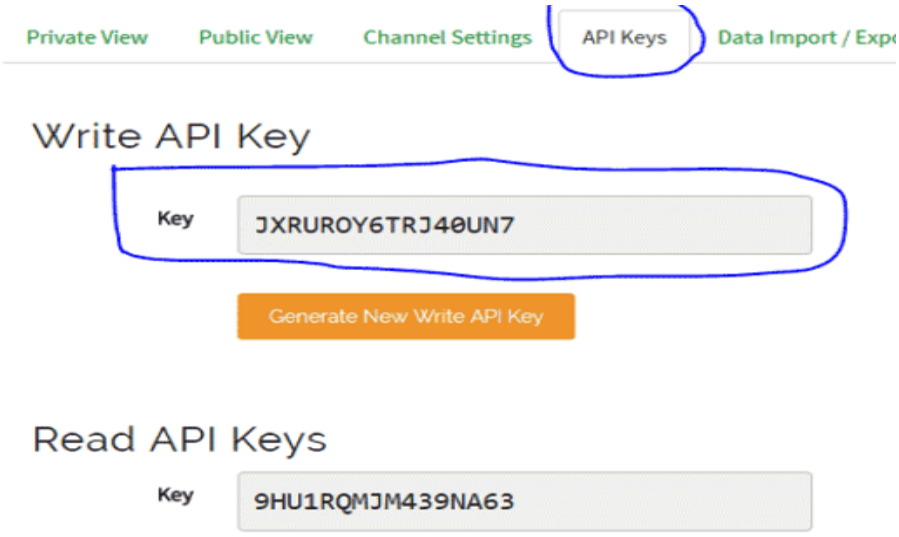

Figure 4.9: **registering channel and API key generation**

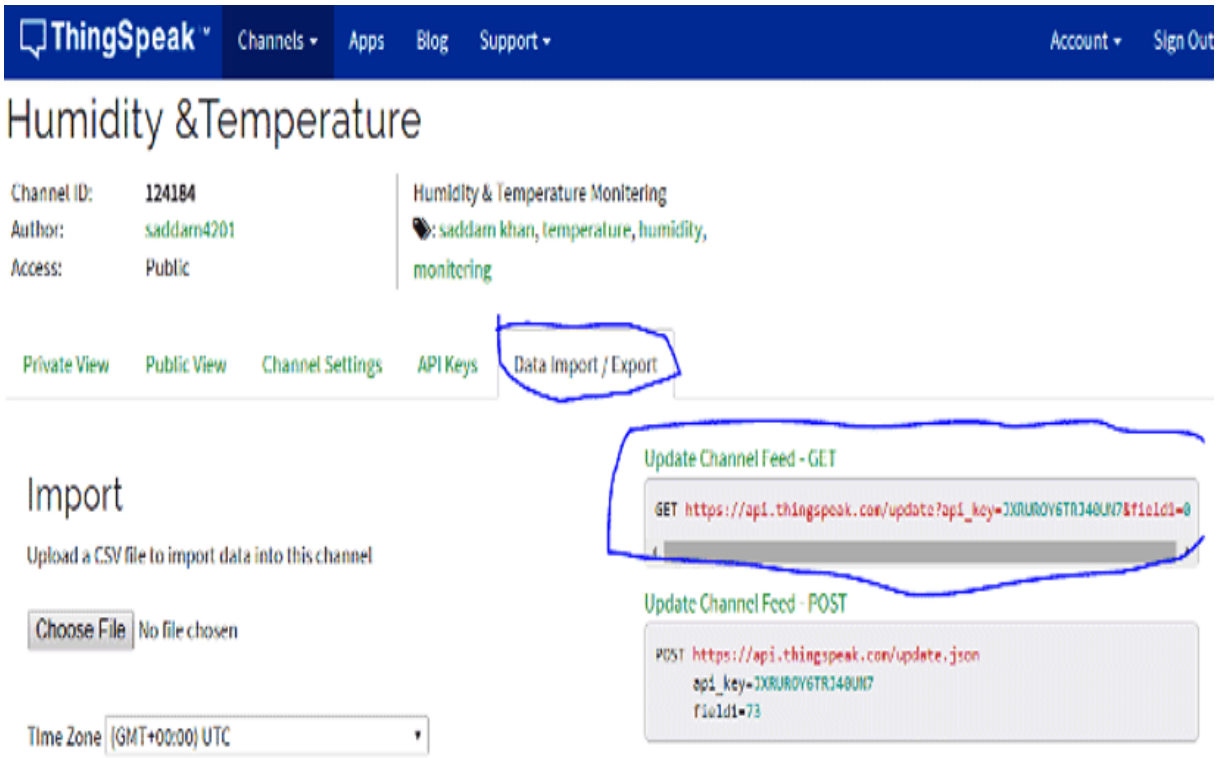

Figure 4.10: **API generation for data file retrieval in JSON format**

4.3.3 Module 3: Telegram chat bot

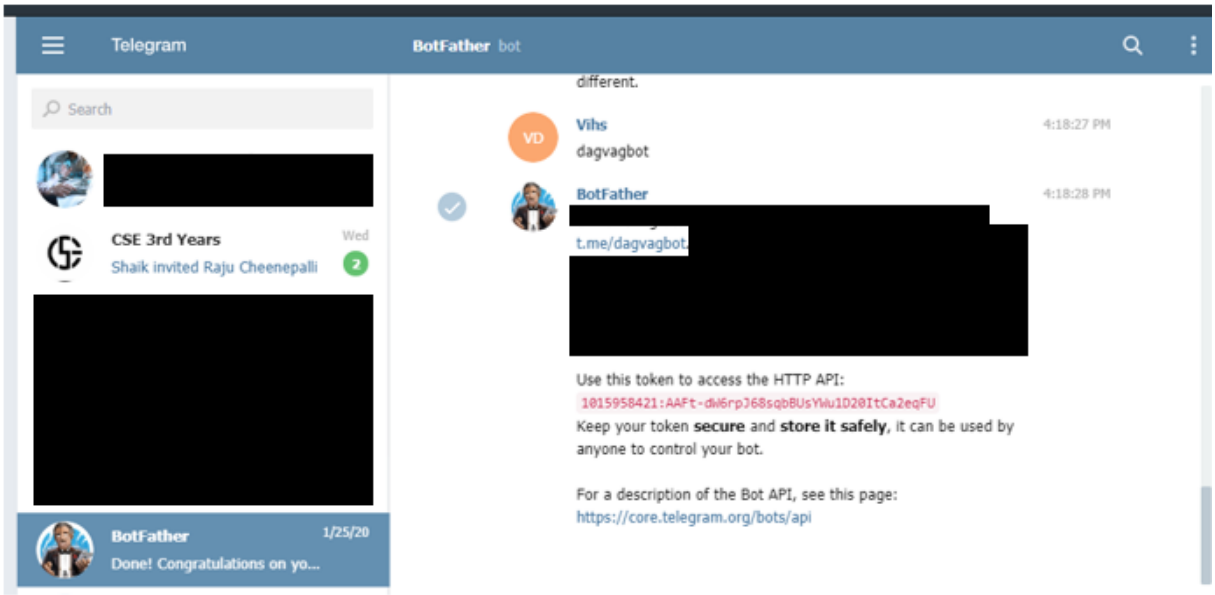

Figure 4.11: **generating unique API and building telegram chat bot**

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

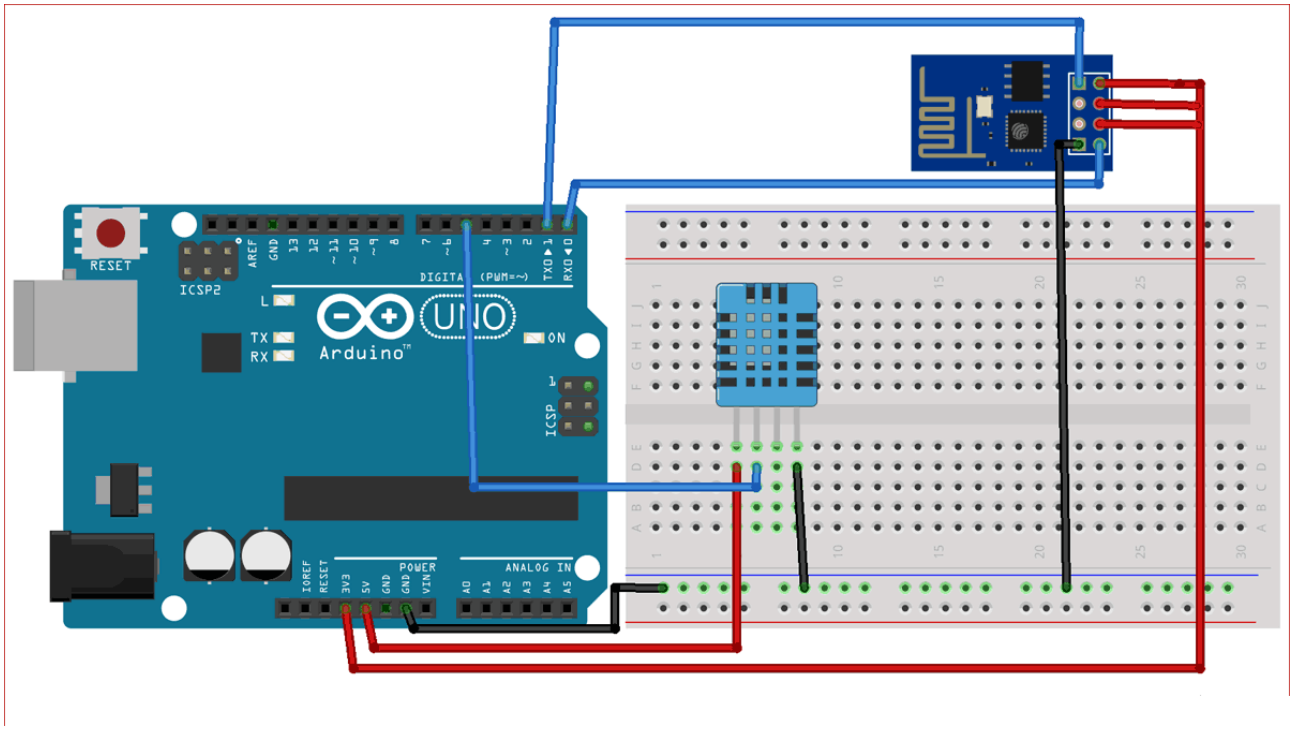

Figure 5.1:

The proposed system involves a circuit consisting of the hardware module, where the arduino UNO plays as the core center which process all the sensory data coming from the DHT11 module once in every 15 seconds as it is the limit of requests which the cloud DB can handle. It consiste of a connectivity module i.e Node MCU which enales the arduino UNO to interact with the cloud by sending data packages every 15 seconds to a linked chennel through an unique API, specific to the channel.

5.1.2 Output Design

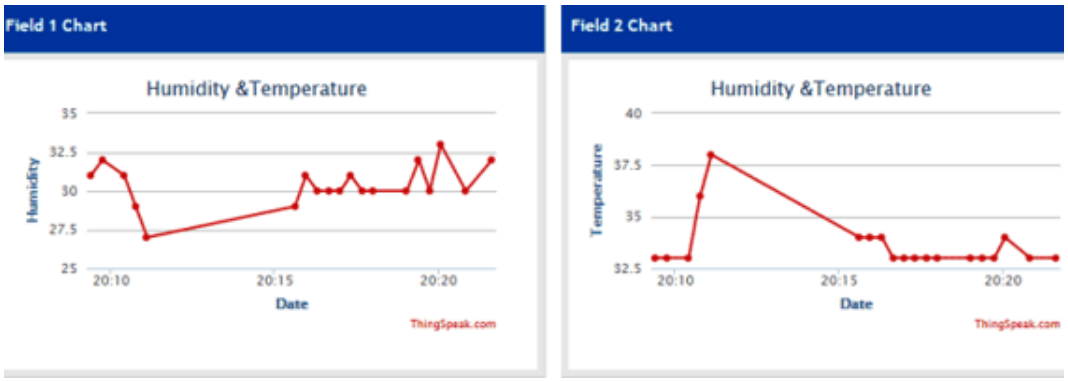

Figure 5.2: **Test Image**

.

The system make uses of a IOT-Cloud service which provides a channel where the edge module is able to interact and we can visualise the data through a specific API unique to the channel, as shown in Fig 5.2, the visual representation of the sensory data which is received through the edge module, this same data is fetched in form of JSON files and is reported via chatbot.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Our project consiste of total of 3 modules so unit testing would be done on each of the modules:

1. The edge layer and its components:
In this testing we tested the accuracy of edge layer sensors and we tested the efficiency of the edge layer algorithms for data extraction.

2. The cloud database (a thing-speak service):
In this testing, we assured that the cloud server restful API are accessible, updated and works efficiently.

3. The telegram chatbot:
In this testing, we assured that the clients messages are accessible through the API end points of the telegram chatbot.

Input

Table 1: edge testing on boundary values

CONSTRAINTS	KNOWN TEMEPERATURE	TEMEPRATURE AS PER THE EDGE MODULE	KNOWN HUMIDITY	HUMIDITY AS PER EDGE MODULE
SAMPLE 1:	36.66*C	35*C	61 RH	64 RH
SAMPLE 2:	36.66*C	35.56*C	61.56 RH	64.95 RH
SAMPLE 3:	36.05*C	35.01*C	62.33 RH	64.87 RH
SAMPLE 4:	35*C	34*C	67 RH	69.93 RH
SAMPLE 5:	29*C	30*C	58 RH	58.45 RH
SAMPLE 6:	38*C	37.89*C	59.67 RH	61 RH
SAMPLE 7:	28.55*C	29.05*C	58 RH	59.03 RH

in the edge module we tested the accuracy at end points at values at which it is expected

Test result

The calculated accuracy of humidity sensor = +0.52 percent RH
The calculated accuracy of temperature sensor = +0.1 *C
The following accuracy degree falls under the standard of operation, and it gave accurate reults even at the boundary values.

5.3.2 Integration testing

In the integration testing section, we will be performing testing by using automated tools, in this case we have made used of Sudan testing tool, which compromises of the stage flow hence, reporting every status of each stage, and hence examining the status we can bring conclusions.

Input

```
1
2 Integration test for edge-cloud-server module integration:
3
4 SUDAN testing tool Version: 1.4.10
5 MQTT Connection Code: 0
6 TESTING:
7  $SYS/broker/version
8 $SYS/broker/timestamp
9 $SYS/broker/uptime
10 $SYS/broker/clients/total
11 $SYS/broker/clients/maximum
12 $SYS/broker/clients/inactive
13 $SYS/broker/clients/disconnected
14 $SYS/broker/clients/active
15 $SYS/broker/clients/connected
16 $SYS/broker/clients/sensory_processing
17 $SYS/broker/messages/data_extraction
18 $SYS/broker/messages/refining_data
19 $SYS/broker/messages/data_received
20 $SYS/broker/messages/data_stored
21 $SYS/broker/messages/data_updated
22 $SYS/broker/messages/data_updated_api_json
23 $SYS/broker/client/inactive
24 $SYS/broker/client/disconnected
25 $SYS/broker/timestamp
26 $SYS/broker/uptime
27 $SYS/broker/clients/total
28 $SYS/broker/clients/maximum
29
30
31 Integration test for cloud-processing module integration:
32
33 SUDAN testing tool Version: 1.4.10
34
35 $SYS/broker/version
36 $SYS/broker/timestamp
37 $SYS/broker/connection/wirenboard-AOLRRIF3.wb221/state
38 $SYS/broker/uptime
39 $SYS/broker/clients/total
40 $SYS/broker/clients/maximum
41 $SYS/broker/clients/inactive
42 $SYS/broker/clients/disconnected
43 $SYS/broker/clients/active
44 $SYS/broker/clients/connected
45 $SYS/broker/clients/expired
46 $SYS/broker/messages/stored
47 $SYS/broker/messages/received
```

```
$SYS/broker/messages/sent
$SYS/broker/subscriptions/count
$SYS/broker/retained messages/count
$SYS/broker/heap/current
$SYS/broker/heap/maximum
$SYS/broker/publish/messages/dropped
$SYS/broker/publish/messages/received
$SYS/broker/load/messages/sent/15min
$SYS/broker/load/publish/dropped/1min
$SYS/broker/load/publish/dropped/5min
$SYS/broker/load/publish/dropped/15min
$SYS/broker/load/publish/sent/1min
$SYS/broker/load/publish/sent/5min
$SYS/broker/load/publish/sent/15min
$SYS/broker/load/publish/received/1min
$SYS/broker/load/publish/received/5min
$SYS/broker/load/publish/received/15min
$SYS/broker/load/bytes/received/inactive
$SYS/broker/disconnected
```

Test result

As we can see,on close examining the stages flow is as per the data flow model, and hence can be concluded that the integration testing between different modules is as per required constraints.

5.3.3 Functional testing

Input

```
JSON file:

{"channel":{"id":84008,"name":"Temperature \u0026 Humidity","description":"Temperature \u0
```

Test Result

Expected output:
end points where the temperature and humidity constraints crossed the threshold . i.e (50RH and 40*C)

Actual output:

1st humidity end points : 54.06.00 RH

2nd humidity endpoint : 55.00 RH

Temperature end point : NILL

as we can examine the on the basis of data set the reporting is done very accurately and hence would be considered a success as a functional testing module.

5.3.4 White Box Testing

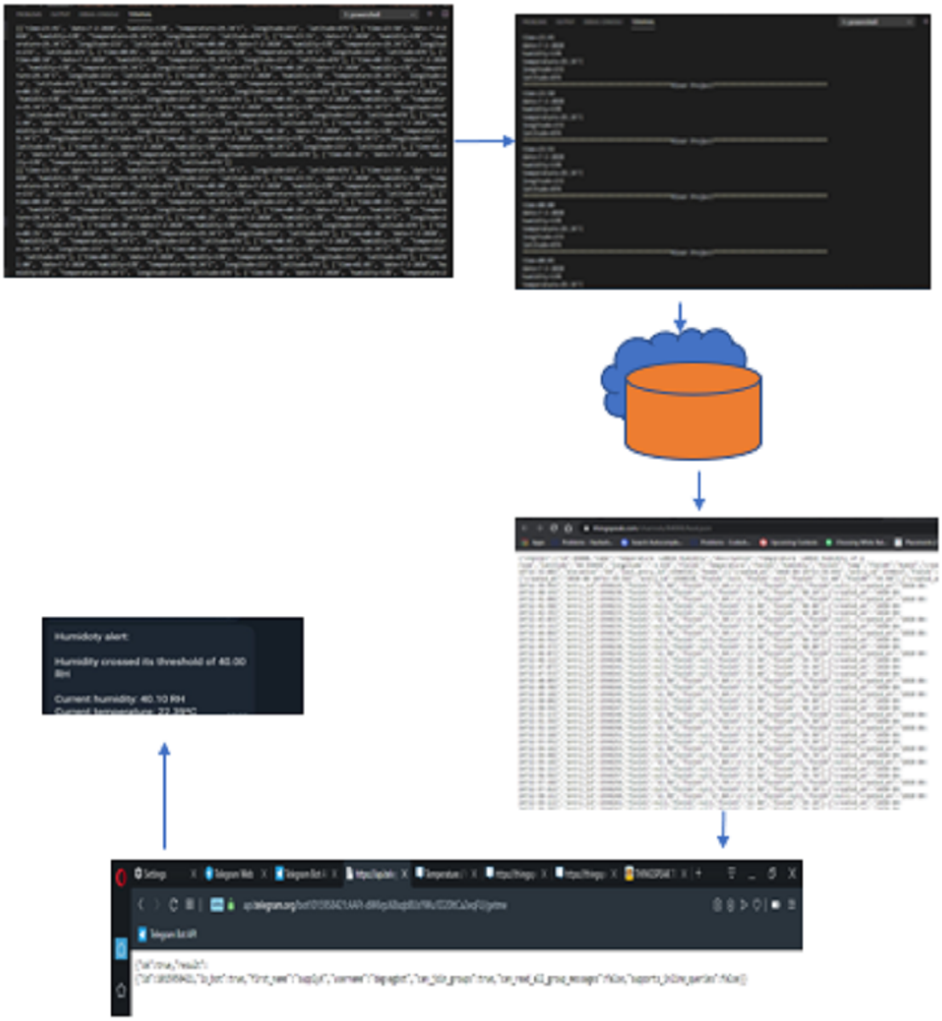

Figure 5.3: **flow of each and every stage based on the previous stages**

.

Here each phase of output is examined on the basis of the previous output which act as an input module for the output examination. Like the generation of JSON file data set is examined on the basis of the data set generated at the edge layer, likewise every stage is dependent on one or more privous stages and hence is examined wholly on the basis of all the previous stages.

5.3.5 Black Box Testing

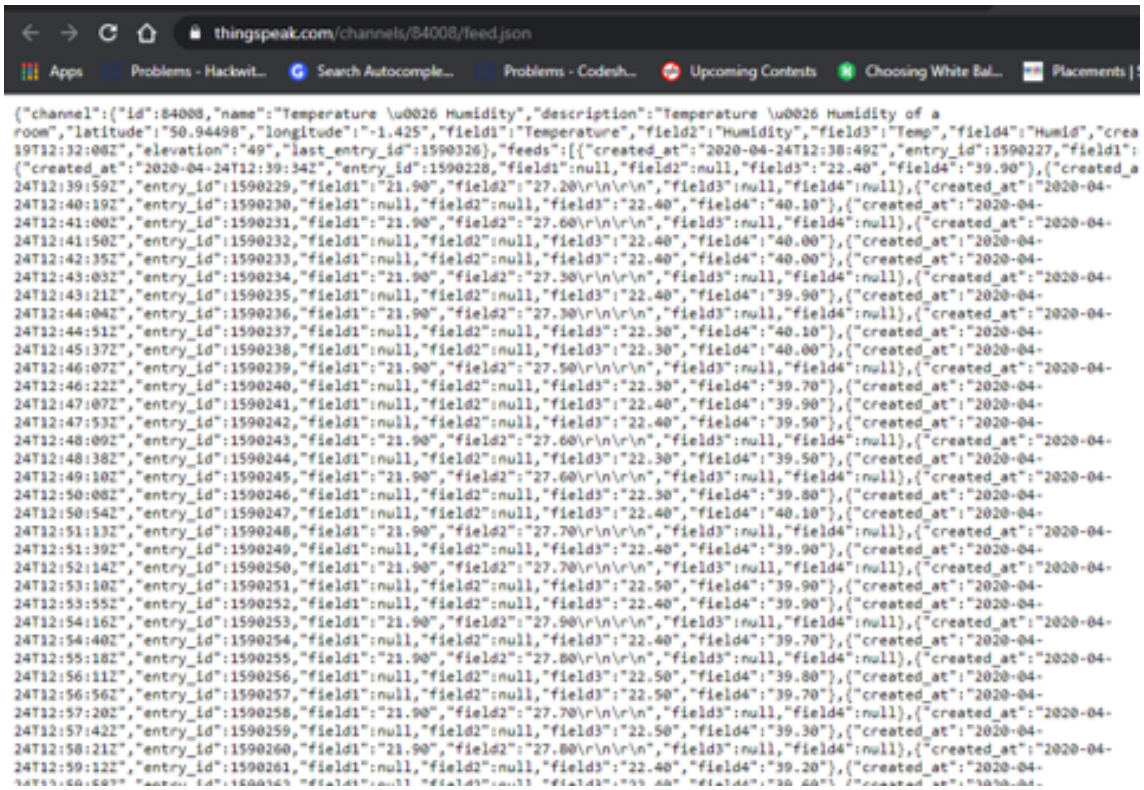

Figure 5.4: **input data set including boundary cases**

.

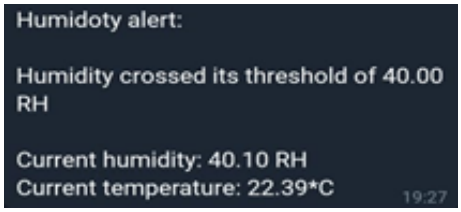

Figure 5.5: **output yielded on input basis**

.

5.3.6 Test Result

Test Scenarios	Result
Unit testing of the edge layer	Pass
Integration testing of the IoT-Cloud interaction	pass
Integration testing of the Cloud-process model interaction	pass
Functional testing of the edge computing and chat computing	Pass
White box testing	Examined and validated
Black box testing	Examined and validated

Figure 5.6: **Test Results**

5.4 Testing Strategy

We have performed unit testing on each of the 3 modules based on the input/output of the modules, and on the basis of response of the get/ post commands to retrieve API, and JSON files.

In case of Integration testing, we have made used of some automated testing tool like Sudan, which is able to present the status of each stage in every second, by analysing this continuous row of status presentation, we can easily evaluate the flow of the functioning system. By examining the status reading, we can assure the updation of data in every stage.

The Functionality testing involves the determining of functional units, which in this case are the chat retrieval algorithm and edge data processing algorithm. whose testing is purely based on the performance of the algorithm on boundary values.

In case of Black Box testing, the output is avaluated on the basis of feeded sample input and is examined for correctivity.

wWhile in White Box testing each stage is evaluated on the basis of previous

stage and is examined accurately by comparing expected output stage with the actual output stage.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of the proposed system is majorly decided by two components i.e the latency time between each module interaction and the degree of accuracy and precision of the sensor. In order to overcome to make it efficient we have made used of minimal edge processing and minimal computing to detect the approaching threshold, by this way the latency between the threshold sensing by the sensor and reporting the same is minimised to a greater extent. The proposed system efficiency can be improved by including the standard sensor with sensor of greater degree of accuracy and precision, by this we can drastically increase the whole efficiency of the proposed system. And my using a grater band width we would be able to increase the efficiency of reporting to much reliable extent.

6.2 Comparison of Existing and Proposed System

As the existing system make uses of man monitoring the system with technology, there could be constraint hikes which can go undetectable, as one can't keep continuing checking the conditions every short interval, therefore there is a high chance of failing to recognise the temperature and humidity hikes and may be unable to yield accurate results in time. In the proposed system we have overarmed this unreliability by making it semi-automatic, the proposed system is able to check the constrains once in every 15 seconds and would transmit if the constraints are approaching the pre-determined threshold value. And moreover the existing system is much costlier in comparison of the proposed system, as the proposed one is an one time installment.

6.3 Advantages of the Proposed System

The main advantage of the proposed system is that, it would be a reliable source of monitoring the supply chain unlike the human-monitoring system, which was unreliable in many extents, our proposed system is able to manage cold chains and other sensitive supply chains, ensuring absolute precision and accuracy. With a low latency, our proposed system is able to report the approaching threshold conditions in almost no time.

6.4 Sample Code

code for edge computing and cloud DB servcoie storage

```
1
2 #include <ESP8266WiFi.h>
3
4 String apiKey = "H38TEGNC0XKW43BB";     //  Enter your Write API key from ThingSpeak
5
6 const char *ssid =  "how2electronics";     // replace with your wifi ssid and wpa2 key
7 const char *pass =  "alhabibi";
8 const char* server = "api.thingspeak.com";
9
10 #define DHTPIN 0          //pin where the dht11 is connected
11
12 DHT dht(DHTPIN, DHT11);
13
14 WiFiClient client;
15
16 void setup()
17 {
18       Serial.begin(115200);
19       delay(10);
20       dht.begin();
21
22       Serial.println("Connecting to ");
23       Serial.println(ssid);
24       WiFi.begin(ssid, pass);
25      while (WiFi.status() != WL_CONNECTED)
26     {
```

```
27              delay(500);
28              Serial.print(".");
29      }
30       Serial.println("");
31       Serial.println("WiFi connected");
32
33 }
34
35 void loop()
36 {
37
38       float h = dht.readHumidity();
39       float t = dht.readTemperature();
40
41              if (isnan(h) || isnan(t))
42                 {
43                      Serial.println("Failed to read from DHT sensor!");
44                       return;
45                 }
46
47                          if (client.connect(server,80))   //   "184.106.153.149" or api.th
48                          {
49
50                                  String postStr = apiKey;
51                                  postStr +="&field1=";
52                                  postStr += String(t);
53                                  postStr +="&field2=";
54                                  postStr += String(h);
55                                  postStr += "\r\n\r\n";
56
57                                  client.print("POST /update HTTP/1.1\n");
58                                  client.print("Host: api.thingspeak.com\n");
59                                  client.print("Connection: close\n");    client.print("X-THINGS
60                                  client.print("Content-Type: application/x-www-form-urlencoded
61                                  client.print("Content-Length: ");
62                                  client.print(postStr.length());
63                                  client.print("\n\n");
64                                  client.print(postStr);
65                                  Serial.print("Temperature: ");
```

```
                                Serial.print(t);
                                Serial.print(" degrees Celcius, Humidity: ");
                                Serial.print(h);
                                Serial.println("%. Send to Thingspeak.");
                            }
            client.stop();
            Serial.println("Waiting...");
  // thingspeak needs minimum 15 sec delay between updates
  delay(1000);
}
```

.

code for Telegram chat bot(python)

```
from telegram.ext import Updater, CommandHandler, Filters, MessageHandler

updater = Updater(token="1015958421:AAFt-dW6rpJ68sqbBUsYWu1D20ItCa2eqFU")
dispatcher = updater.dispatcher

def start(bot, update):
    bot.send_message(chat_id=update.message.chat_id, parse_mode="HTML", text="hey there!!!

start_handler = CommandHandler("start", start)
dispatcher.add_handler(start_handler)

def func(text):
    l = ["hii", "Hii", "Hello", "hello", "r u there", "sup"]
    k = ["show me the status","Show me the status"]
    funcs = ["how are you", "how are u doin", "wassup", "u should be doing something", "su

    if text in l:
        return 'hey {name}!!!, track your supply chain products real-time'
    elif text in funcs:
        return 'hey i am here to help you {name}'
    else:
        return 'sorry!! i didnt undersatnd {name} what u just said'

def echo(bot, update):
    bot.send_message(chat_id=update.message.chat_id, text=func(update.message.text).upper(

echo_handler = MessageHandler(Filters.text, echo)
```

```
28
29 dispatcher.add_handler(echo_handler)
30 updater.start_polling()
31
32 refugeAPI=Updater(url(https//thingspeak.com/channel=id=23242312SJDBS))
33 object1=Filters.retrive(refugeAPI)
34 json_1=Filters.convertJSON(refugeAPI)
35 x=Filters.retrive_sonstraints(json_1)
36 fix_treshold=34.7
37 fix_humidity=30
38 echo(Updater(fix_humidity,fix_treshold, x))
```

Output

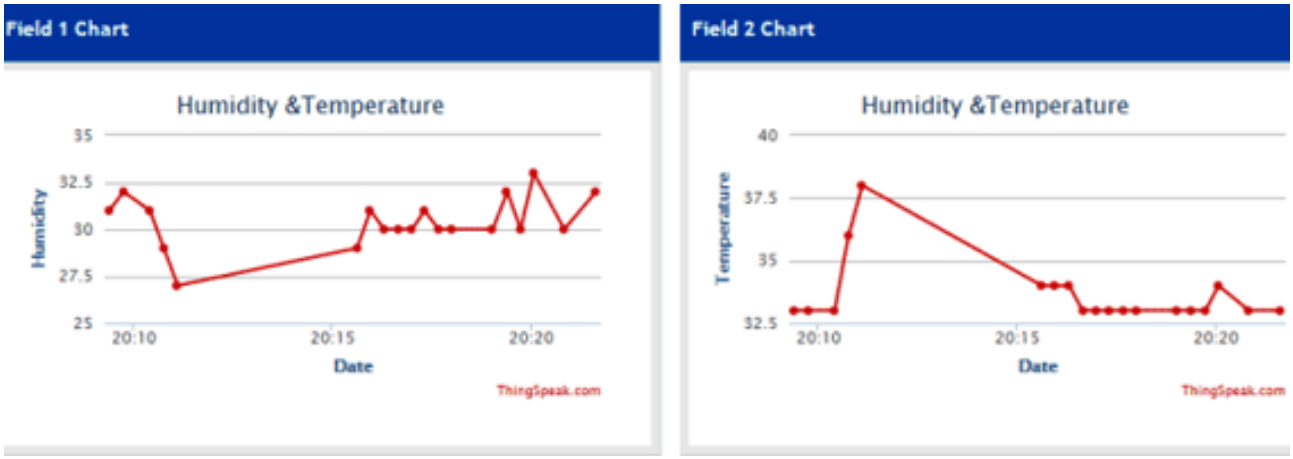

Figure 6.1: **Output 1**

Figure 6.2: **Output 2**

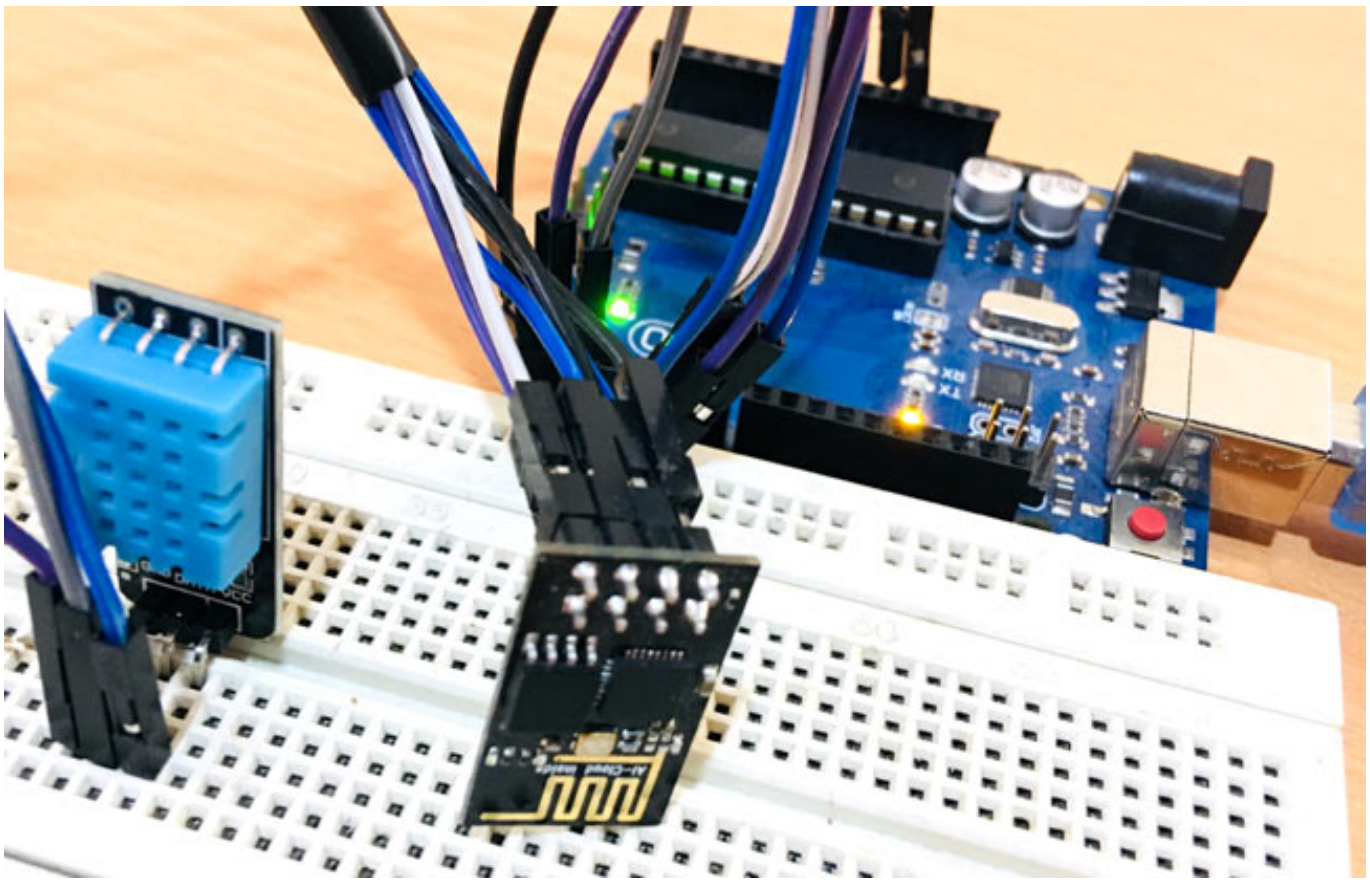

Figure 6.3: **Output 3**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In this report, we have discussed the existing system, and how inefficient and unreliable it was, and we proposed our solution to overcome that unreliability and to provide accurate and precised feedback to the owner and others who is concerned. Our project aimed to build a compact product, which acts as reliable source for monitoring supply chain systems, resolving many problems resulting in severe consequences. We explained how our project interacts with the system environment and the efficient ways of handling the huge amount of real-time sensory data.

7.2 Future Enhancements

The proposed cold chain monitoring system functions accurately according to conditions, but the whole process of supply chain monitoring can be enhanced into working automatically without any user reporting, and making its own optimal decision with high degree of considerations, this can be achieved via integrating further decision making modules powered by artificial intelligence. This way we can make a fully automated supply chain monitoring as well as decision making system, which requires almost no human feedback.

Reference

[1] D. Seifert, ”Efficient Consumer Response: Supply Chain Management (SCM)” in Category Management (CM) and Collaborative Planning Forecasting and Replenishment (CPFR) als neue Strategieansätze, Mering:Verlag Hampp, vol 2, pg. 2,3,2004

[2] Beijing Laboratory of pharmaceutical and Safety, College of Engineering, China Agricultural and medical University, Beijing 100083, vol 1, issue id: Linz86055, pg. 4,5,6,7,2007

[3] College of Food Science, Tibet Agricultural and Animal Husbandry College, china national journal, vol 1, issue id. Linzi860000, page. 1,2,5,8, 2012