SMART TEMPERATURE DETECTION SYSTEM

A PROJECT REPORT

Submitted in fulfillment of requirement
for the degree of

Bachelor of Engineering
In
INFORMATION TECHNOLOGY

Submitted by

SAURABH RAUT(B120958555)
DARPAN SETHI(B120958518)
SUBHASH DEMUNDE(B120958520)
ROHITH PULLURI(B120958549)

Under the Guidance of

PROF. ANAND BHOSALE

DEPARTMENT OF INFORMATION TECHNOLOGY
INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,
SAVITRIBAI PHULE PUNE UNIVERSITY
HINJAWADI, PUNE(MH)-411057

CERTIFICATE

DEPARTMENT OF INFORMATION TECHNOLOGY

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJAWADI, PUNE-411057

This is to certify that

SAURABH RAUT(B120958555)

DARPAN SETHI(B120958518)

SUBHASH DEMUNDE(B120958520)

ROHITH PULLURI(B120958549)

Class: BE(IT) have satisfactorily completed Project titled, '**SMART TEMPERATURE DETECTION SYSTEM**' under my supervision as a part of Bachelor of Engineering in **INFORMATION TECHNOLOGY(2017-2018)** of Savitribai Phule Pune University.

Prof. ANAND BHOSALE
Project Guide

Prof. MANJUSHA AMRUTKAR
HOD(IT)

Principal

Place : Pune
Date :

External Examiner

Abstract

The system proposed in this report is an advanced solution for monitoring temperature at particular place which will raise an alert as soon as temperature goes beyond certain range. The technology behind this is cloud computing, Internet of Things which is an advanced and efficient solution for connecting the entire world of things in a network. The systems deals with monitoring and controlling environmental condition like temperature with sensors and send this information to cloud. The data updated from the current reading can be used to generate an e-alert if the temperature goes beyond certain limit. The proposed system may be used in the situation where fire can cause damage to the computers and may effect on users current work. This system may be useful in maintaining users integrity by automatically terminating the current transactions. Firebase real time database can be used to monitor the temperature conditions. Moreover, an alert message notification was implemented in Android application so that a user is notified whenever the temperature reading reaches the preset threshold. On the other hand, the smart chair system has brilliant commercial prospects, which can be helpful to build health care products with the help of wearable sensors, intelligent refrigerator/oven temperature tracking system and etc.

Acknowledgement

Here we have made a sincere attempt to acknowledge and expressing the deep sense gratitude to all those who have contribute to the successful completion of this academic project of such nature and magnitude. Working on the project Smart Temperature Detection System is lot of enjoyment and work. As interdependence is higher value than independence, we feel deep sense of gratitude to all those who have given us great deal of inspiration and guidance from start of our project work to end of our seminar/presentation. In this we are also thankful to Prof. Manjusha Amritkar H.O.D. of IT Department our Project. In his cordial support Prof. Vishal Chaudhary and entire Staff of IT Department for their unending heartier efforts during the project work. We would like to add special thanks to our project guide Prof. Anand Bhosale who have guided us right through the project work and seminar presentation. We are highly grateful to writers of those articles, books and magazines which we have referred during the literature survey and project work. We would also thanks to all the people who had directly or indirectly helped and guided us in our project work. Last but not least we would like to thank to all our friends, who stood by us throughout

Contents

Certificate i

Abstract ii

Contents iv

List of Figures vii

List of Tables 1

1 Introduction 1

2 Literature Survey 2

3 Proposed Methodology 5

3.1 Temperature sensor captures the current temperature reading and uploads to cloud server: 5

3.2 Real time data management using firebase: 5

3.3 Displaying current temperature on the portal: 5

3.4 Cloud sever will fetch the readings and match up with the boundary conditions: 6

3.5 Generation of an e-alert message: 6

3.6 Sending the location of the fire affected place to nearest fire station: . . 6

3.7 Respective action taken by Fire Station Admin: 6

4 PROPOSED SYSTEM OVERVIEW 7

4.1 EXTERNAL INTERFACE REQUIREMENT 8
4.1.1 User Interface 8
4.1.2 Hardware Interface 8
4.1.3 Software Interface 8
4.2 NON FUNCTIONAL REQUIREMENTS 8
4.2.1 Performance Requirement 8
4.2.2 Software Quality Attributes 8

5 ESP8266 9

6 FIREBASE REAL TIME DATA MANAGEMENT 11
6.1 Firebase Cloud Messaging 11
6.2 Real Time Database 11
6.3 Authentication 12
6.4 Hosting 12
6.5 Full featured App Platform 12

7 CONFIGURATION OF ESP8266 AND FIREBASE 13

8 SECURING USERS DATA 14
8.1 Cloudinary Management Console 14
8.2 Data Types With Cloudinary Management: 14
8.3 Cloudinary Management Dashboard: 15

9 UML DIAGRAM 16
9.1 Data Flow Diagram 16
9.2 System Architecture 18
9.3 System Architecture 19
9.4 Usecase 20

10 CONNECTION OF ESP 8266 WITH FIREBASE 21
10.1 Setting Up ESP 8266 21

10.2 FIREBASE MANAGEMENT CONSOLE 21
10.2.1 Creating A New Project 21
10.2.2 Setting Up Configuration 21
10.2.3 Using Firebase in Web 22
10.2.4 Updating Real Time Value On Firebase 22

11 TESTING **23**
11.1 SOFTWARE TESTING: 23
11.2 FUNCTIONAL TESTING 23
11.3 SYSTEM TESTING 24
11.3.1 White Box Testing: 25
11.3.2 Black Box Testing: 25

12 DESIGN AND IMPLEMENTATION CONSTRAINTS: **27**
12.1 Network Traffic 27
12.2 Limited Data Type Support 27
12.3 Network Failure 27

13 ASSUMPTION AND DEPENDENCIES **28**
13.1 Connectivity Providing Equipments Safety 28
13.2 Data Types Feasible To Cloud Server 28

14 Implementation Of Project **29**

15 Conclusion And Future Scope **35**

References **36**

List of Figures

4.1 System Overview. 7

5.1 ESP8266. 10

8.1 Cloudinary Dashboard. 15

9.1 Data Flow Diagram Level 1 16

9.2 Data Flow Diagram Level 2 17

9.3 System Architecture 18

9.4 System Architecture 19

9.5 Usecase 20

10.1 Temperature Update 22

11.1 Test Case 26

14.1 Login Page. 29

14.2 Main Page. 30

14.3 Location. 31

14.4 Cloudinary. 32

14.5 Dashboard. 33

14.6 Media Lib. 34

Chapter 1

Introduction

Cloud computing which mainly consists of using others resources for our use has become an enabling technology ecosystem with several application areas are Smart Home, weather forecasting, etc. Present innovations in technology mainly focus on controlling and monitoring of different activities. These are increasingly emerging to reach the human needs.

The data stored or processed by other can be accessed by cloud servers and can generate an e-alert message on entities like mobile, laptop, etc. connected to cloud using internet. Cloud service provider provides resources and services to upload, retrieve and visualize data received from various devices in cloud. Various applications are there to provide exciting features.

To add to these developing trends we are trying to improvise the existing cloud and IoT features in such a way that temperature can be monitored and an e-alarm can be generated. The monitoring of temperature is possible with temperature sensors and arduino. This details or readings can be fetched by cloud servers and matched with the data readings and can be sent to each users connected to cloud service. Each user can view the current temperature reports and can rely on the systems in case of fire.

Chapter 2

Literature Survey

We have referred following papers:

Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved

OVERVIEW: Cloud computing and Internet of Things (IoT), two very different technologies, are both already part of our life. Their massive adoption and use is expected to increase further, making them important components of the Future Internet. A novel paradigm where Cloud and IoT are merged together is foreseen as disruptive and an enabler of a large number of application scenarios. In this paper we focus our attention on the integration of Cloud and IoT, which we call the Cloud IoT paradigm. Many works in literature have surveyed Cloud and IoT separately: their main properties, features, underlying technologies, and open issues. However, to the best of our knowledge, these works lack a detailed analysis of the Cloud IoT paradigm. To bridge this gap, in this paper we review the literature about the integration of Cloud and IoT. We start analyzing and discussing the need for integrating them, the challenges deriving from such integration, and how these issues have been tackled in literature. We then describe application scenarios that have been presented in literature, as well as platforms both commercial and open source and projects implementing the Cloud IoT paradigm. Finally, we identify open issues, main challenges and future directions in this promising field.

Internet of Things-based Temperature Tracking System

The work described in this paper consists of a temperature tracking system that follows a Client-Server architecture. A Raspberry-Pi, a System-on-a-Chip (SoC) device, is responsible for sensing the temperature and streaming it to a server; the readings then are displayed in a mobile android application. For this system, a python application was developed to sense and stream the temperature, a servlet was developed to read and store the temperature in a SQLite database, and a mobile Android application was developed to read and display the temperature readings from the server. The initial versions of the project used the SoC device as a server (storing temperature readings into a local SQLite database), and both the SoC device and the mobile device needed to be connected in a local area network. However, the project was further developed to separate the server responsibilities from the SoC device. The system now supports user authentication, and both devices are connected through the Internet. This implementation allows the temperature readings to be viewed and displayed anytime from anywhere in the world since the database is hosted on a server which can be accessed over the internet. Also, this solution allows multiple SoC devices to stream temperatures to the server, to different mobile clients using the same database. The Android client application was also implemented to graphically show the temperature readings recorded by Raspberry-Pi using RESTful architecture. Moreover, an alert message notification was implemented in Android application so that a user is notified whenever the temperature reading reaches the preset threshold. On the other hand, the smart chair system has brilliant commercial prospects, which can be helpful to build health care products with the help of wearable sensors, intelligent refrigerator/oven temperature tracking system and etc.

Internet of Things (IoT) based Sensors to Cloud system using ESP8266 and Arduino Due The system proposed in this paper is an advanced solution for monitoring the weather conditions at a particular place and make the information vis-

ible anywhere in the world. The technology behind this is Internet of Things (IoT), which is an advanced and efficient solution for connecting the things to the internet and to connect the entire world of things in a network. Here things might be whatever like electronic gadgets, sensors and automotive electronic equipment. The system deals with monitoring and controlling the environmental conditions like temperature, relative humidity, light intensity and sound level with sensors and send this information to the cloud and then plot the sensor data as graphical statistics. The data updated from the implemented system can be accessible in the internet from anywhere in the world.

Web Services: Software-as-a-Service (SaaS), Communication, and Beyond Web services is a fast moving research field with a profound impact to many critical research areas, ranging from software to communication, from server platforms to mobile endpoints. It emerges as a disruptive technology to various enterprises to deliver services, computing, communication and information to their customers and partners. This talk intends to capture some recent advances of Web services and new Web services applications in softwareas- a-services, cloud computing, communication, message centric SOAP engine design, and mobile services computing endpoints. In addition to present the technical perspectives and extension potentials in these new developments, I would like to fill the gap between academic research and industry applications through real examples and use cases, with a goal to demonstrate how these technical advances of Web services are applied to industry products and applications, Standards, servers, clients, and new paradigms in software design and testing.

Chapter 3

Proposed Methodology

This project has been divided into several modules. These have been listed below.

3.1 Temperature sensor captures the current temperature reading and uploads to cloud server:

The temperature sensor will capture the current temperature readings. It will upload the current readings to cloud server after regular intervals. The cloud server will fetch the current readings of the temperature at the particular location.

3.2 Real time data management using firebase:

With the help of Amazon AWS a console can be build which will take the input readings check with the certain predefined readings and will be used to show the temperature readings to the user.

3.3 Displaying current temperature on the portal:

The current temperature readings will be made displayed on portal of the device connected to the cloud servers.

3.4 Cloud sever will fetch the readings and match up with the boundary conditions:

The cloud server will continuously check the updated readings with the boundary conditions set by system admin.

3.5 Generation of an e-alert message:

In case the temperature readings goes beyond certain range, there can be possibility of fire. In such case an e-alert message in the form of alert over portal and text message can be set in case device disconnected from the cloud server network.

3.6 Sending the location of the fire affected place to nearest fire station:

This stage mainly includes sending the coordinates of fire affected place to nearest fire station with the help of GPS of device.

3.7 Respective action taken by Fire Station Admin:

After getting an alert message from respective location fire station admin can take the subsequent action to control over the fire. The alert to fire station admin can be in any form such as an email or SMS message.

Chapter 4

PROPOSED SYSTEM OVERVIEW

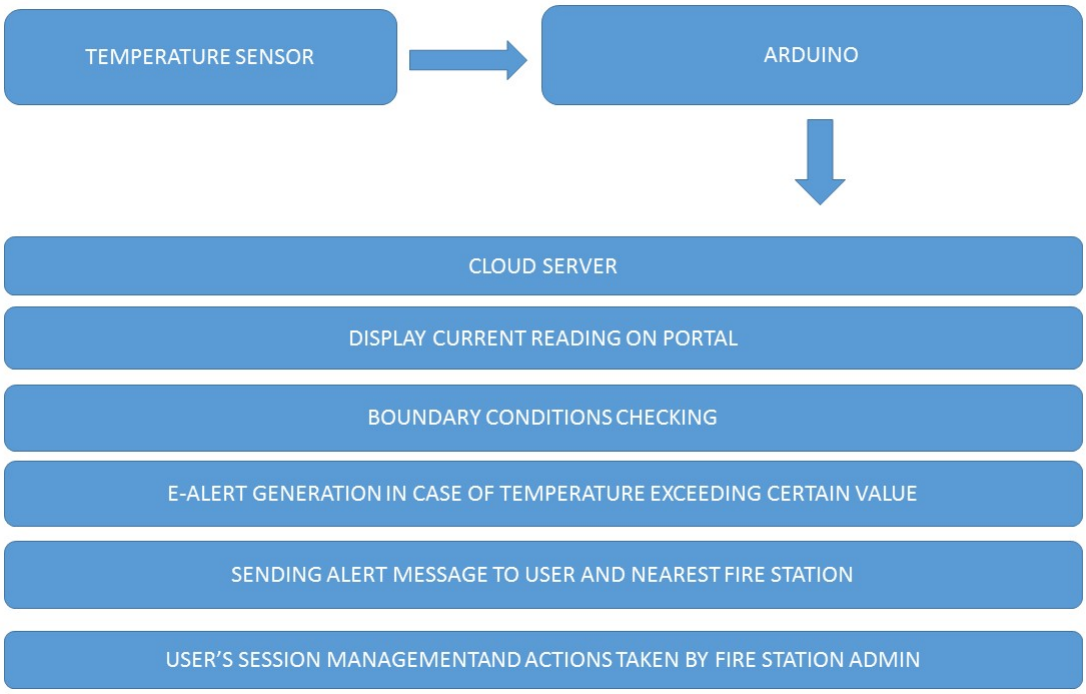

Figure 4.1: System Overview.

4.1 EXTERNAL INTERFACE REQUIREMENT

4.1.1 User Interface

1.Whenever Temperature crosses the threshold value then directly user data will be uploaded on cloud.

4.1.2 Hardware Interface

Application :

1.Computer/Laptop

2.ESP8266

3.Jumper Wires

4.1.3 Software Interface

1. Operating system : Windows, Ubuntu
2. Language : PHP,javascript,HTML

4.2 NON FUNCTIONAL REQUIREMENTS

4.2.1 Performance Requirement

System should give expected results Accuracy. Fast speed. .

4.2.2 Software Quality Attributes

Simple to use.

Easy User interface.

fast response.

Plenty functionalities.

Good Performance

Chapter 5

ESP8266

The ESP8266 is a low-cost WiFi microchip with full TCP-IP stack and microcontroller capability produced by Shanghai based Chinese manufacturer, Espressif Systems. The chip first came to the attention of western makers in August 2014 with the ESP-01 module, made by a third party manufacturer, Ai Thinker. This small module allows microcontrollers to connect to a WiFi network and make simple TCP-IP connections using Hayes style commands. However, at the time there was almost no English language documentation on the chip and the commands it accepted. The very low price and the fact that there were very few external components on the module which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, chip, and the software on it, as well as to translate the Chinese documentation. The ESP8285 is an ESP8266 with 1 MiB of built in flash, allowing for single chip devices capable of connecting to WiFi. The successor to these microcontroller chips is the ESP32.

PinOut of Version-01[edit]

esp v 1 pinout The Pinout is as follows for the 1st basic module, 1. VCC, Voltage (+ 3.3 V (upto 3.6 V it can handle)) 2. GND, Ground (0 V) 3. RX, Receive data bit X 4. TX, Transmit data bit X 5. CH PD, Chip Power Down 6. RST, Reset 7. GPIO 0, General Purpose Input Output No. 0 8. GPIO 2, General Purpose Input Output No. 2

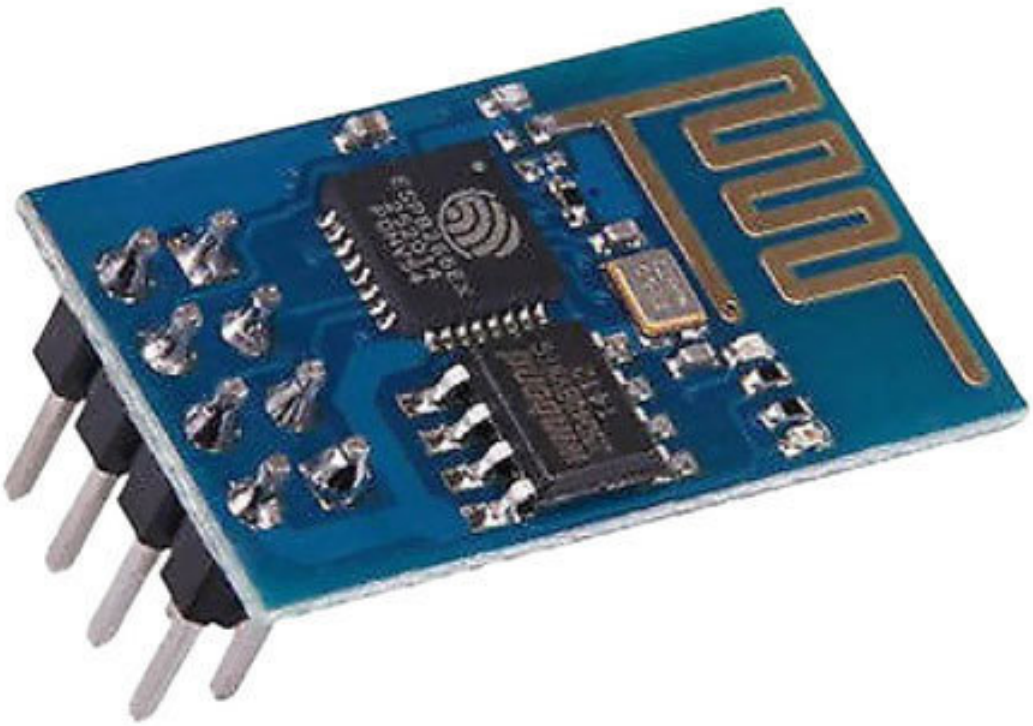

Figure 5.1: ESP8266.

Chapter 6

FIREBASE REAL TIME DATA MANAGEMENT

6.1 Firebase Cloud Messaging

Formerly known as Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) is a cross-platform solution for messages and notifications for Android, iOS, and web applications, which currently can be used at no cost.

6.2 Real Time Database

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The company provides client libraries that enable integration with android, iOS, JavaScript, Java, Objective-C, swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real- time database can secure their data by using the company's server-side-enforced security rules. Cloud Fire store which is Firebase's next generation of the Real-time

Database was released for beta use.

6.3 Authentication

Firebase auth has a built in email/password authentication system. It also supports OAuth2 for Google, Facebook, Twitter and GitHub. Well focus on email/password authentication for the most part. Firebases OAuth2 system is well-documented and mostly copy/paste.

6.4 Hosting

Firebase includes an easy-to-use hosting service for all of your static files. It serves them from a global CDN with HTTP/2.

6.5 Full featured App Platform

The Firebase team has integrated a bunch of new and existing Google products with Firebase. A bunch of these features apply to iOS and Android but not to web. They are: 1.Remote config

2.Test Lab

3.Crash

4.Notification

5.Dynamic Links

6.AdMob

Chapter 7

CONFIGURATION OF ESP8266 AND FIREBASE

Configuration 1. Start Arduino

2. Open File - Examples - FirebaseArduino - FirebaseRoom ESP8266

3. In FirebaseRoom ESP8266: Replace WIFI SSID and WIFI PASSWORD with WiFi credentials

4. Go to https://firebase.google.com/console/ and create a new Firebase Project

5. Go to Database

6. Copy the Database hostname (Database URL without https:// and trailing /)

7. In FirebaseRoom ESP8266: replace FIREBASE HOST with the Database Hostname

8. Go to - Project Settings - Database - Database secrets

9. Click Firebase Secrets - Show

10. Copy the Database Secret

11. In FirebaseRoom ESP8266: Replace FIREBASE AUTH with Database Secret

12. Select the board Board - ESP8266 Modules - NodeMCU 1.0

13. Select the serial port Port - /dev/tty...

14. Select the upload speed Upload Speed - 115200

15. Click Sketch - Upload

Chapter 8

SECURING USERS DATA

Most the times a condition may arise that the damage caused by the fire may cause threat to the significant data stored in users system. Securing user data to local machine is not the feasible option. Smart Temperature detection system provides an option to secure users data by automatically uploading it to the cloud server in case of emergency.

8.1 Cloudinary Management Console

Cloudinary's Management Console provides valuable information about your images, powerful image browsing capabilities, reports, and various account customization options. We have used cluoduinary as a cloud server .

8.2 Data Types With Cloudinary Management:

In the initial stage only image format like .png and .jpeg are uploaded. Since the cloudinary platform is more user-friendly with images we tried to upload only image data types. As the functionality of cloud server increases with the help of advance cloud servers such as Amazon AWS no of file types can be integrated.

8.3 Cloudinary Management Dashboard:

Cloudinary management console provide a smart way to manage your images and personal fiels.

Figure 8.1: Cloudinary Dashboard.

Chapter 9

UML DIAGRAM

9.1 Data Flow Diagram

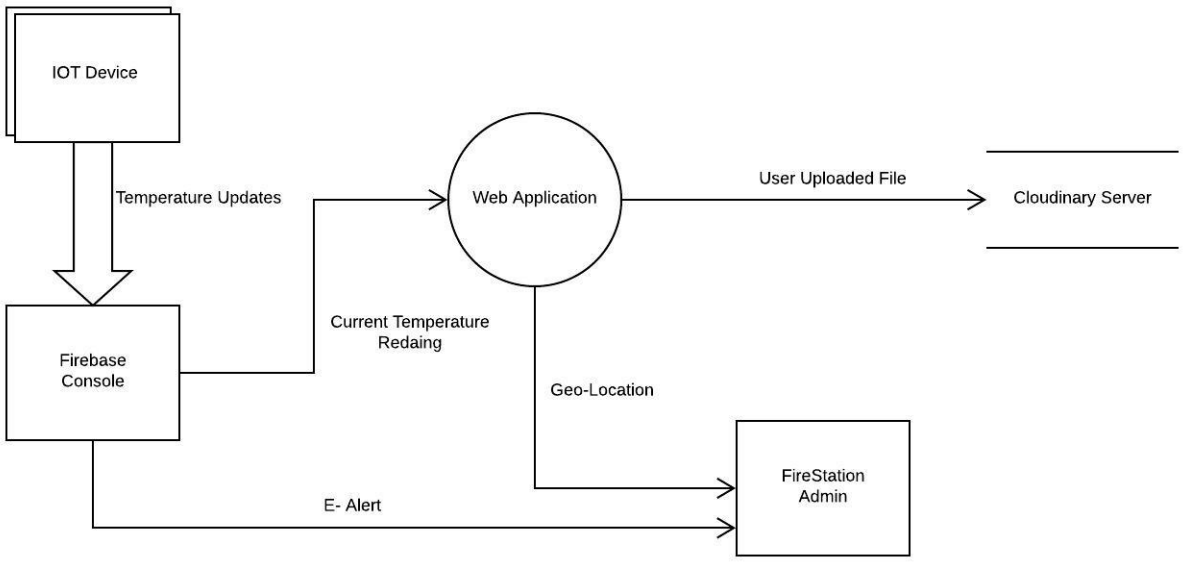

Figure 9.1: Data Flow Diagram Level 1

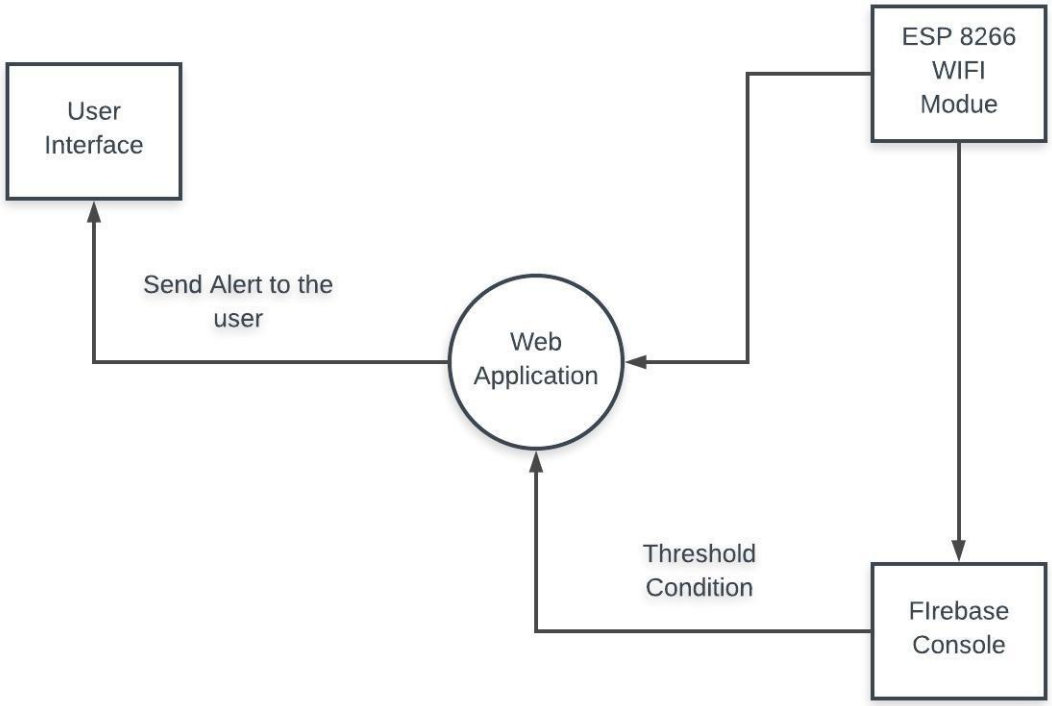

Figure 9.2: Data Flow Diagram Level 2

9.2 System Architecture

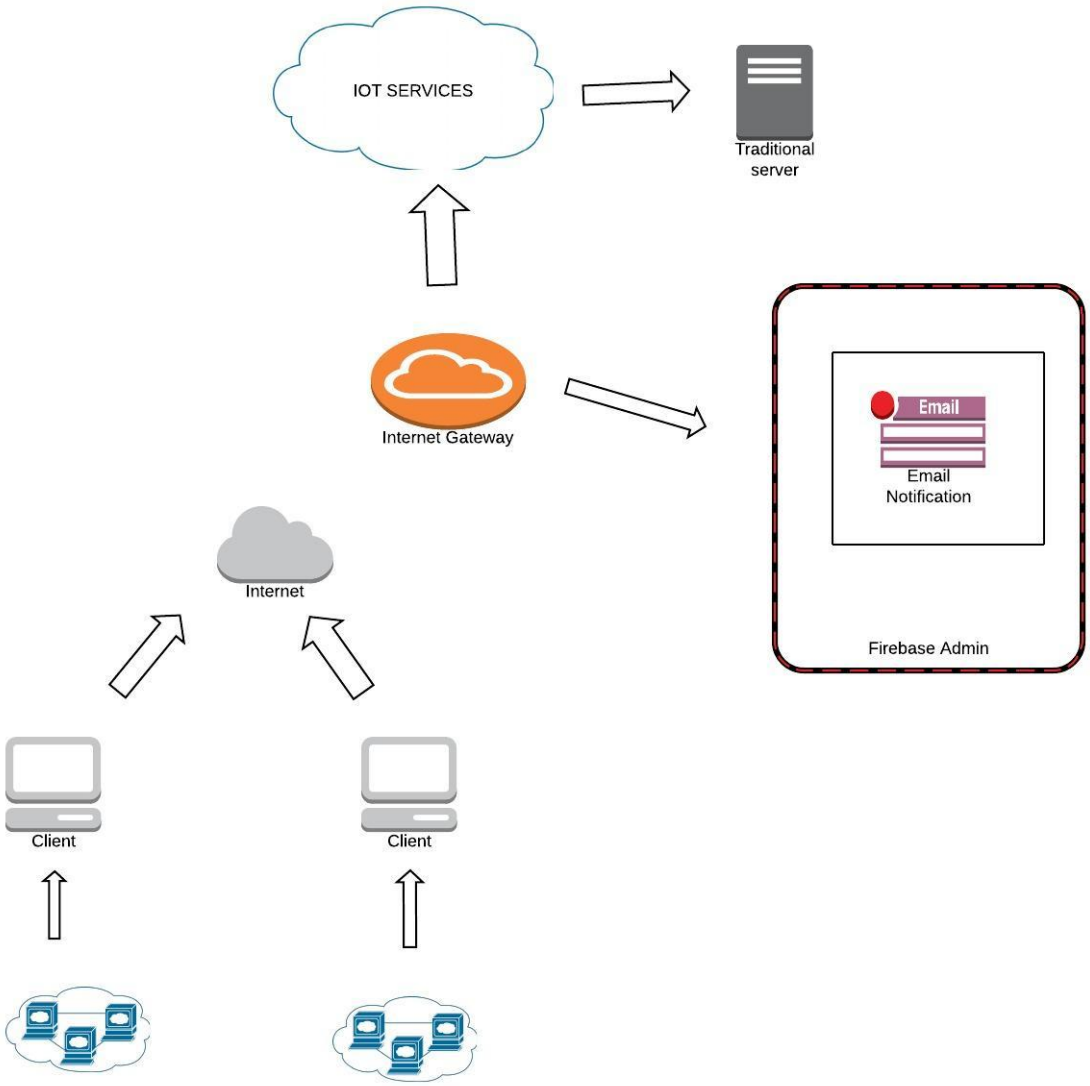

Figure 9.3: System Architecture

9.3 System Architecture

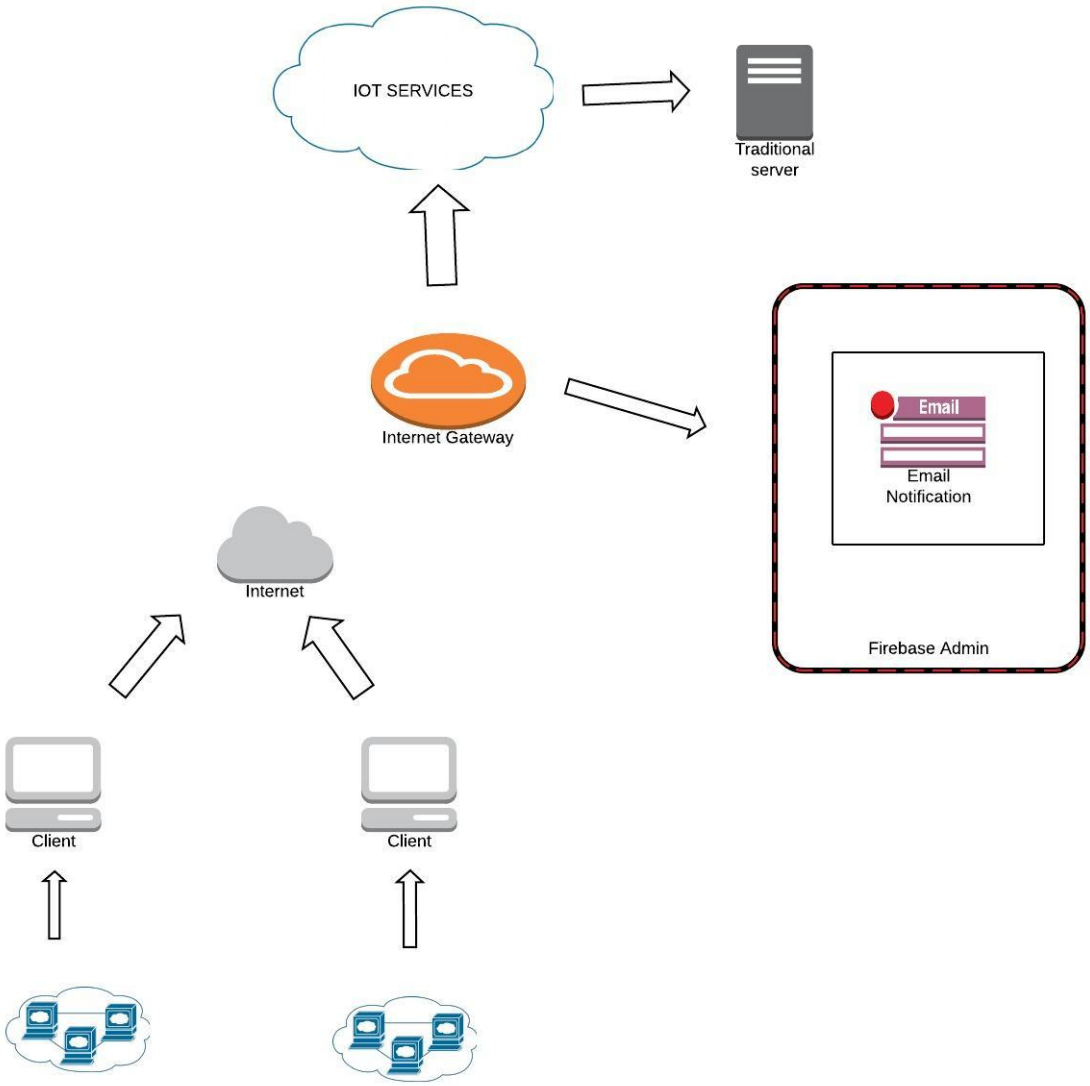

Figure 9.4: System Architecture

9.4 Usecase

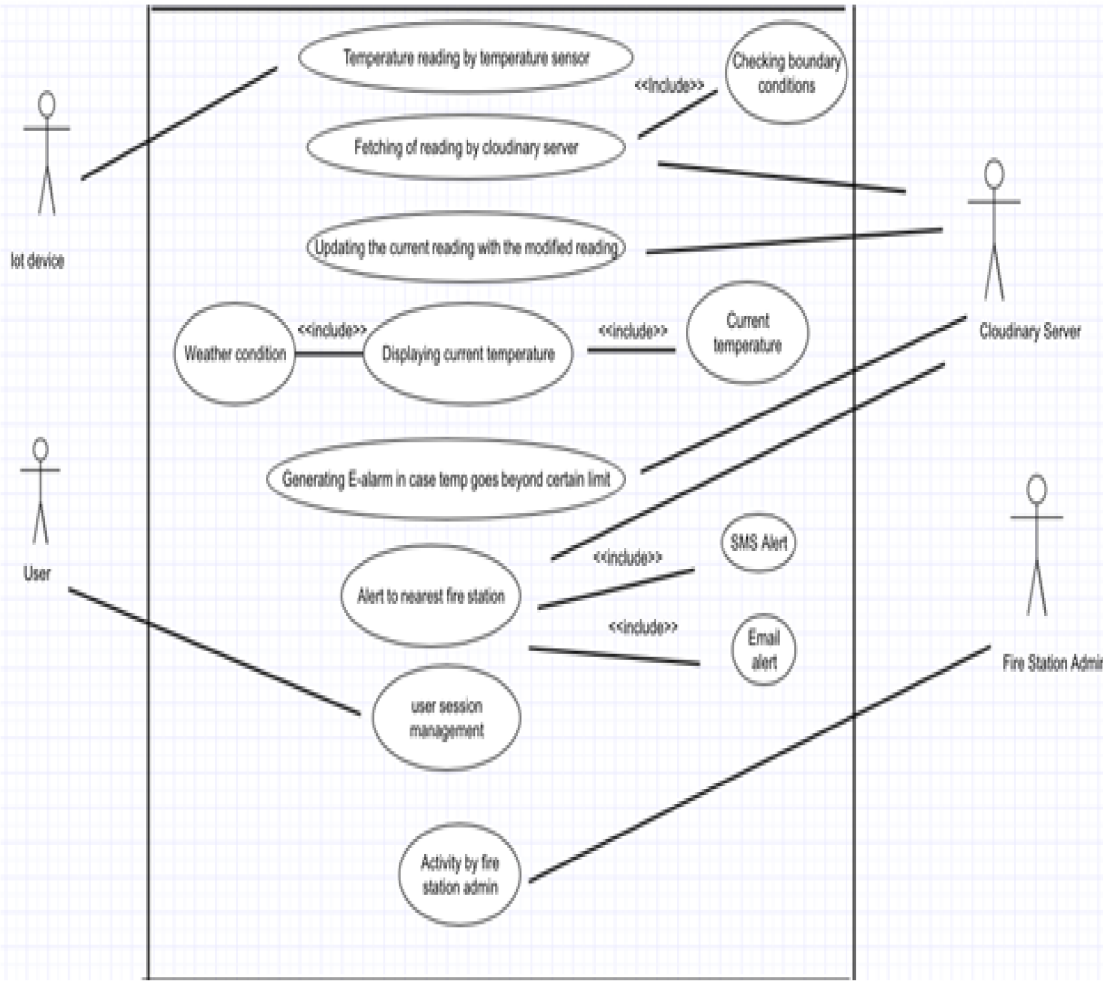

Figure 9.5: Usecase

Chapter 10

CONNECTION OF ESP 8266 WITH FIREBASE

10.1 Setting Up ESP 8266

This stage mainly involves connecting Node MCU ESP8266 wifi module with LM 35 sensor. After connecting hardware with sensors following will enable the sensor to send the temperature readings over the internet.

10.2 FIREBASE MANAGEMENT CONSOLE

10.2.1 Creating A New Project

A new project must be created in order to use firebase management console for our web app. After proper configurations use the toolbox to connect the firebase with our web app.

10.2.2 Setting Up Configuration

To connect our web app to firebase a api key and auth-Domain is required. This api-key is used as passkey to provide access real time database and authentication domain is mainly used for authentication purpose.

10.2.3 Using Firebase in Web

While integrating firebase with our web app can be done with your WiFi router name and password

10.2.4 Updating Real Time Value On Firebase

Thus the real time value fetched by sensors will be get updated to firebase in the form of consecutive readings obtained over the period. Timestamp is an important entity which will update the database value after a successful interval.

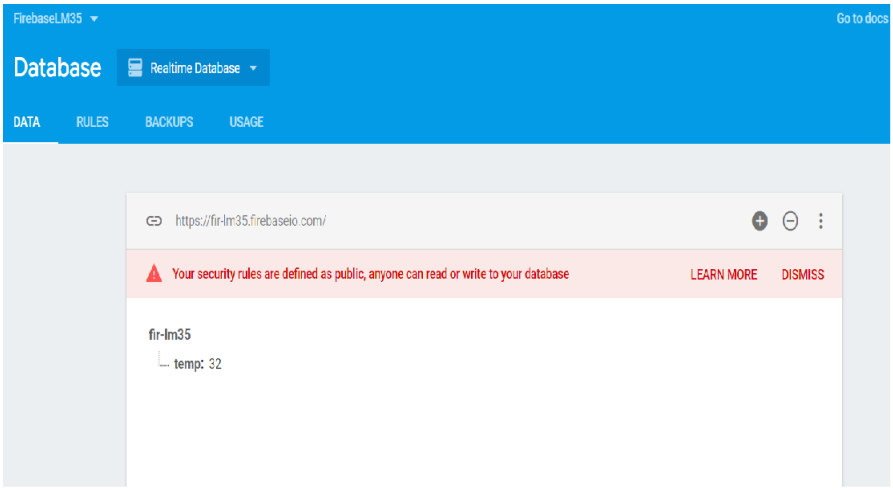

Figure 10.1: Temperature Update

Chapter 11

TESTING

11.1 SOFTWARE TESTING:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use. Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the com-ponent or system under test. meets the requirements that guided its design and development, responds correctly to all kinds of inputs, performs its functions within an acceptable time, is sufficiently usable, can be installed and run in its intended environments, and achieves the general result its stakeholders desire.

11.2 FUNCTIONAL TESTING

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test.

Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing).Functional testing usually describes what the system does. Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system. Functional testing differs from system testing in that functional testing "verifies a program by check-ing it against ... design document(s) or specification(s)", while system testing "validate[s] a program by checking it against the published user or system requirements"

Functional testing has many types:

Functional testing typically involves six steps[citation needed] 1. The identification of functions that the software is expected to perform 2. The creation of input data based on the function's specifications 3. The determination of output based on the function's specifications 4. The execution of the test case 5. The comparison of actual and expected outputs 6. To check whether the application works as per the customer need.

11.3 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hard-ware system(s). The purpose of integration testing is to detect any inconsistencies between the soft-ware units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

11.3.1 White Box Testing:

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

11.3.2 Black Box Testing:

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it.[15] Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing, and specification-based testing.

Test Case	Check Item	Text Case Objective	Steps to Execute	Text Data Input	Expected Result
1	Log-in Page	Leave all fields as blanks and click Log-In Button	Click Log-in		By leaving all fields as blank and on click Log-in button then mandatory symbol (*) should appear in front of Username and Password fields
2	Username	Enter Invalid name	NA	Username : Jacck	By entering invalid username then an error message should appear as "Please Enter Valid Username"
3	Username	Enter Valid Username	NA	Username: Jack	It should allow the user to proceed
4	Password		NA		The password field should display the encrypted format of the type as (****)
5	Password	Enter wrong password	NA	Password: ***	By entering invalid password then an error message should appear as "Please Enter Correct Passowrd"
6	Password	Enter correct password	NA	Password:********	It should allow the user to proceed
7	Log-in Button	Correct Inputs	Click Log-in		It should lead the user to respective application of web portal
8	Data Upload	Check if file accessible to the users or not	NA		Allow only Authorized users to access the uploaded data
9	Current Reading Temperature	Fetch the real time data from the arudino	NA		The real time temperature is shown to the user on the web portal
10	Data type	Check if the data	NA		Only real time data types like images is

Figure 11.1: Test Case

Chapter 12

DESIGN AND IMPLEMENTATION CONSTRAINTS:

12.1 Network Traffic

Number of users deployed by smart temperature detection system will make a major effect on speed of data transmission . Users data having high importance sometimes may be unable to upload on networking causing failure of maintaining users data.

12.2 Limited Data Type Support

A large number of data residing in users system is one of the major constraints since the feasibility of the data type supported by the cloud can unresolvable issue.

12.3 Network Failure

Failure of network during data uploading to the cloud server may cause failure of system since the architecture providing network connectivity may be damaged in the initial level of threat.

Chapter 13

ASSUMPTION AND DEPENDENCIES

13.1 Connectivity Providing Equipments Safety

The electronic components which are going to provide the network connectivity should be keep safe at its best level. Since damage cause to this system may result in alteration of system functionality.

13.2 Data Types Feasible To Cloud Server

Smart temperature system is able to upload data types that have support in cloud server. Unsupported data type may lead to prevent user from accessing data.

Chapter 14

Implementation Of Project

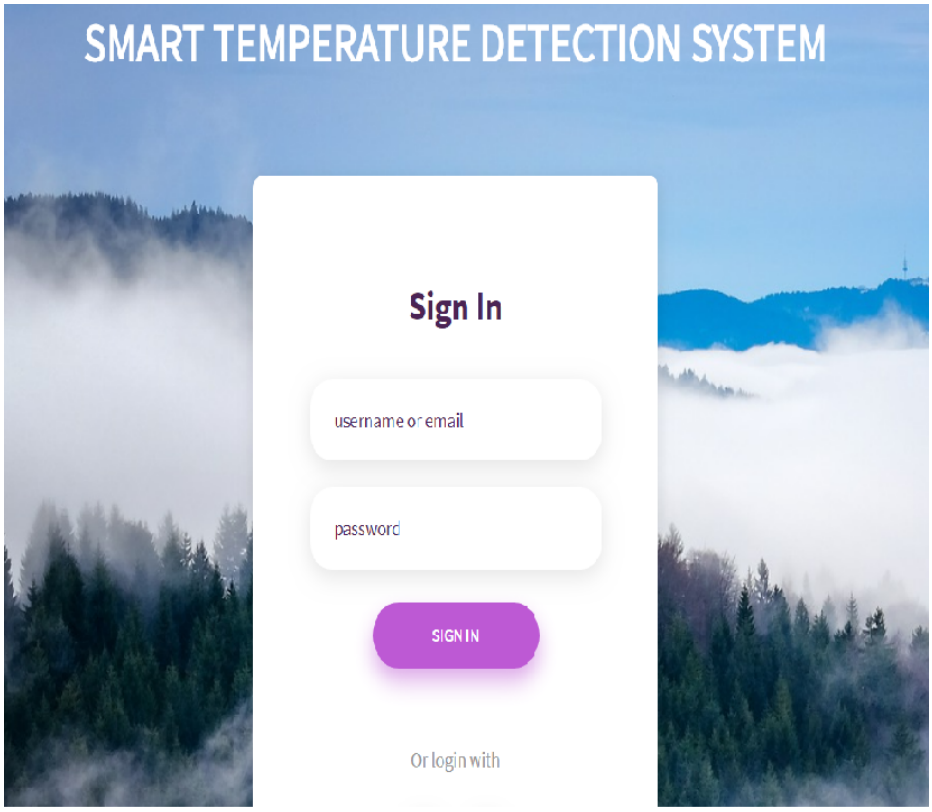

Figure 14.1: Login Page.

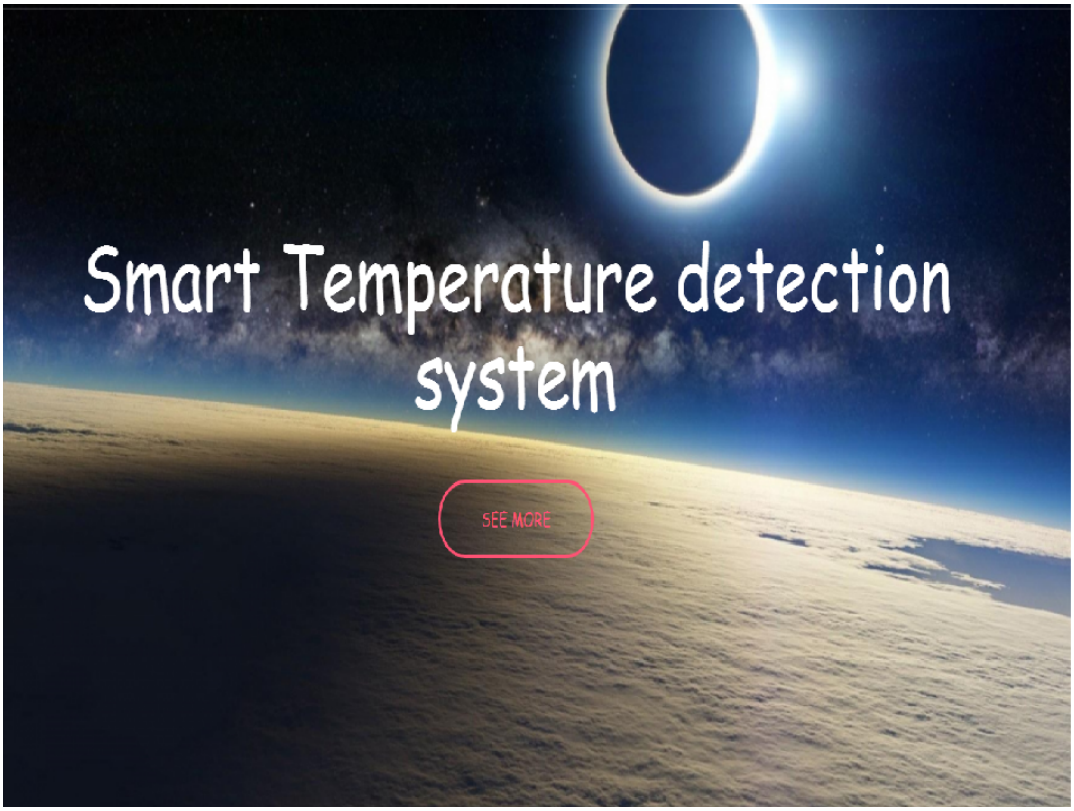

Figure 14.2: Main Page.

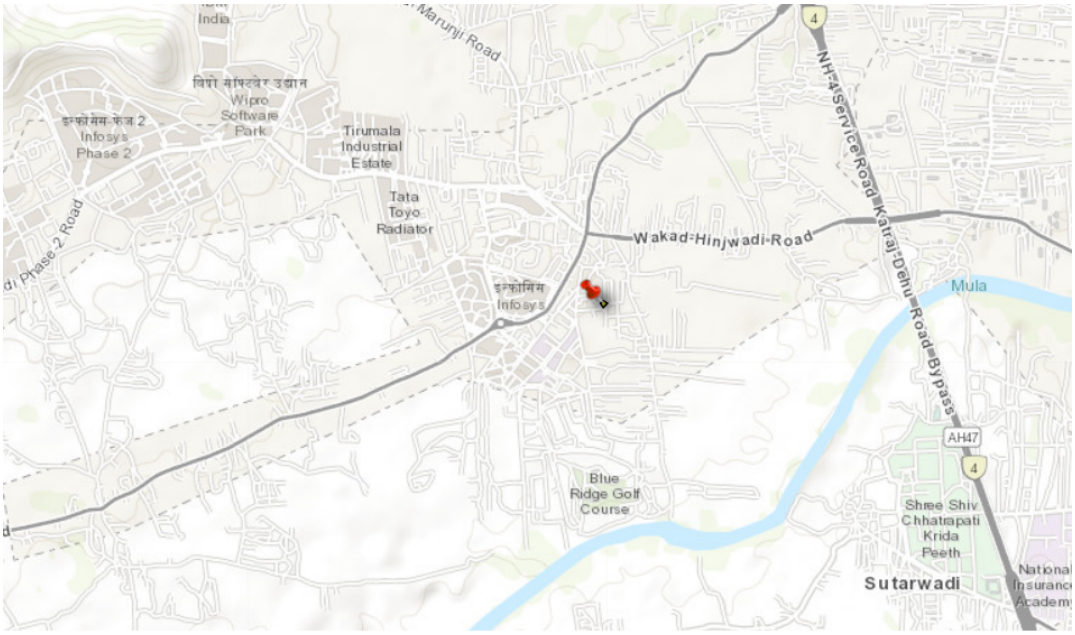

Figure 14.3: Location.

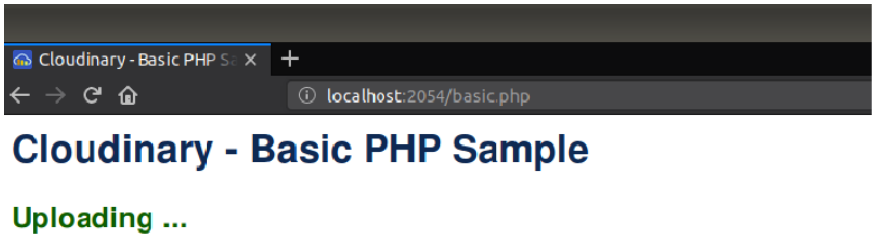

Figure 14.4: Cloudinary.

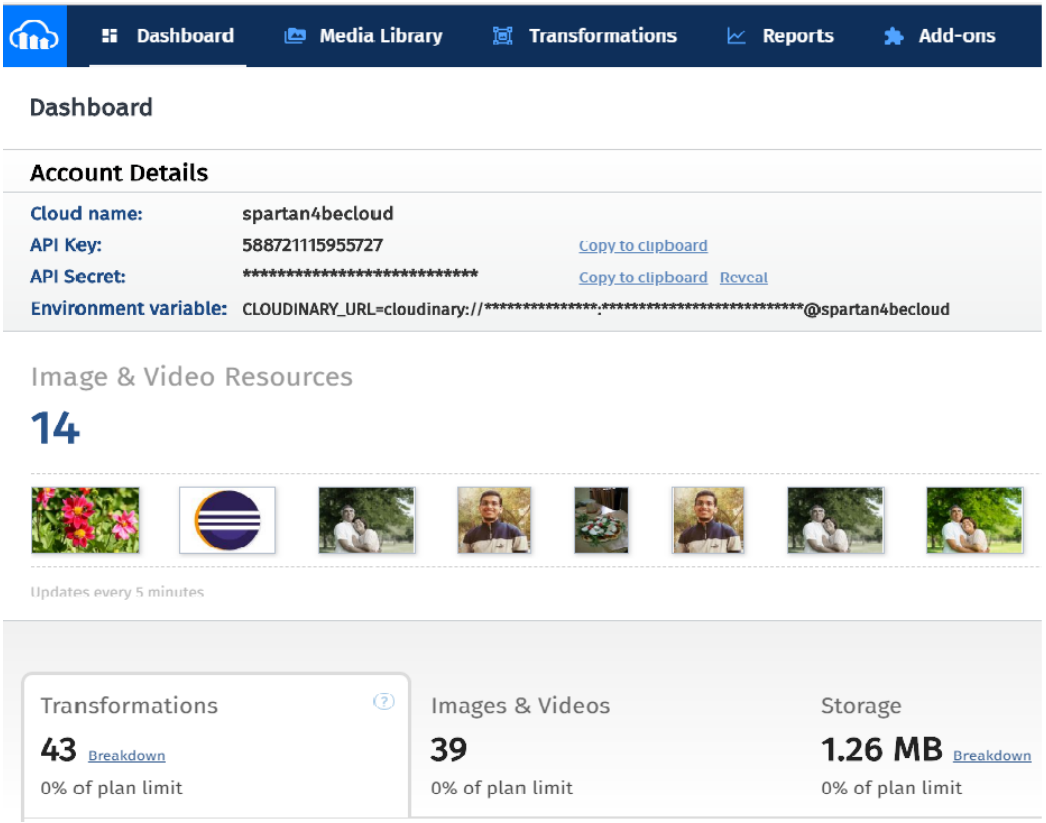

Figure 14.5: Dashboard.

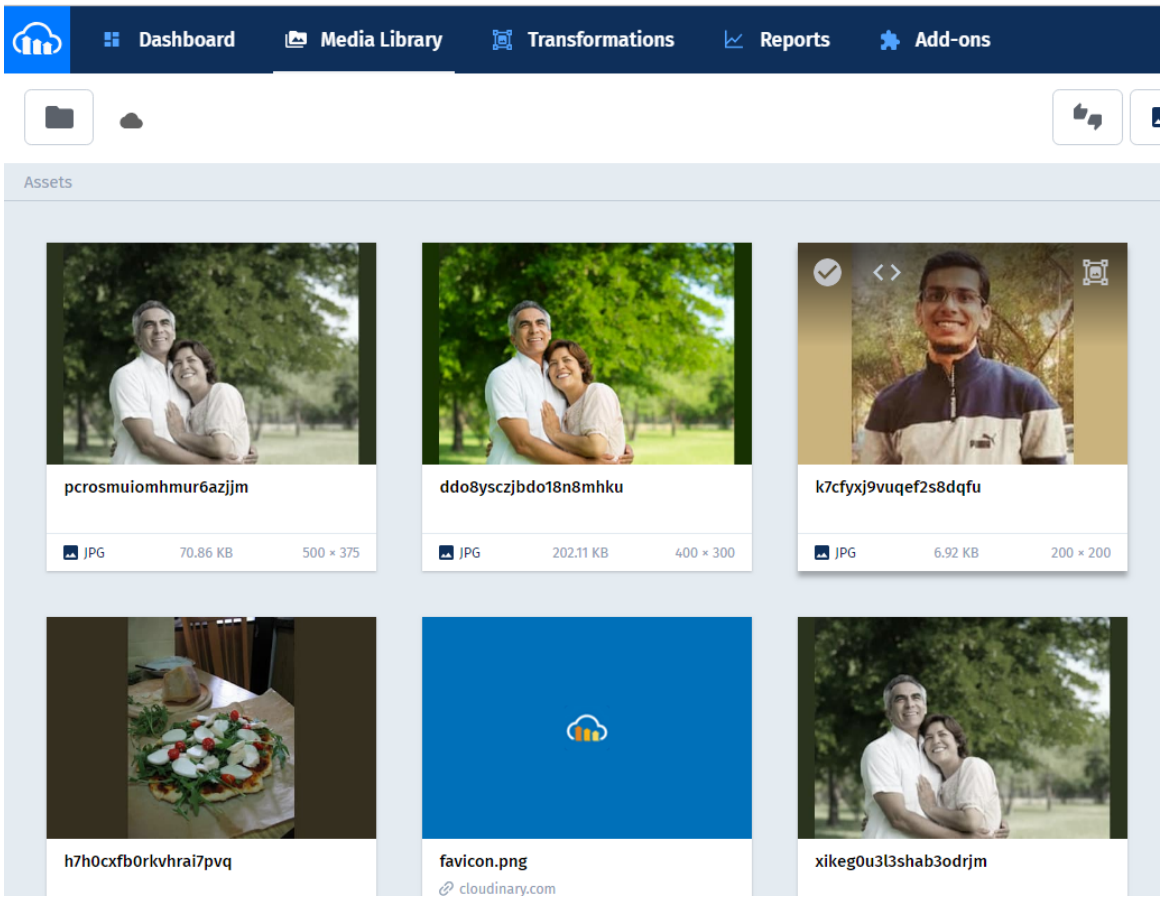

Figure 14.6: Media Lib.

Chapter 15

Conclusion And Future Scope

CONCLUSION

Thus developed system can efficiently use and monitor the temperature sensor. Besides certain constraint this system monitor real time temperature reading and sending E-alert to the nearest fire station in case of emergency. Thus damage cost bias can reduce to certain extend. System only focus on reducing time elapse between place where fire effected area by the fire station admin. All the data can be read by the smart device without interruption and delay because of the efficient use of communication algorithm in the control node. Employing embedded technology, based on Arduino, the Wireless Sensor Nodes are designed and implemented

FUTURE SCOPE

Thus the proposed system can be an integral part while monitoring weather conditions in industrial areas. Several other functionalities can also be integrated by using various IoT devices and sensors. Thus providing security to user data and reducing damage caused by natural disasters.

References

[1] R.V. Kranenburg, E. Anzelmo, A. Bassi, D. Caprio, S. Dodson, and M. Ratto, The internet of things. In Proceedings of 1st Berlin Symposium on Internet and Society, Berlin, Germany, October, 2011.

[2] D. Bandyopadhyay, and J. Sen, Internet of things: Applications and challenges in technology and standardization, Wireless Personal Communication, 58, 4969, 2011.

[3] Moghavvemi M. and Tan. S. A reliable and economically feasible remote sensing system for temperature and relative humidity measurement. Sensors and Actuators. 2005. 181-185.

[4] MadokaYuriyama, Takayuki Kushida, (2010), Sensor Cloud Infrastructure: Physical Sensor Management with Virtualized Sensors on Cloud Computing 13th International Conference on Network-Based Information Systems, IEEE, 2010

[5] N. Kurata, M. Suzuki, S. Saruwatari, and H. Morikawa, (2008), "Actual Application of Ubiquitous Structural Monitoring System using Wireless Sensor Networks", The 14th World Conference on Earthquake Engineering (14WCEE), 2008.