IoT BASED REMOTE WEATHER CONTROLLING

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Computer Science and Engineering

Project Mentor:
Name: Mr. Sushant Kumar
Designation : Assistant Prof.

Submitted By :
Keshav Bhandari
[17ESKCS079]

Department of Computer Science and Engineering
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2020-2021

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Computer Science and Engineering

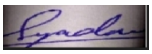
CERTIFICATE

This is to certify that Mr. Keshav Bhandari, a student of B.Tech(Computer Science & Engineering) VII semester has submitted his/her Project Report entitled "IOT BASED REMOTE WEATHER CONTROLLING" under my guidance.

Mentor

Mr. Sushant Kumar

(Assistant Prof.)

Signature: 

Coordinator

Dr. Mukesh Kumar Gupta

(Head of Department)

.

Ms. Anjana Sangwan

(Assistant Prof.)

DECLARATION

We hereby declare that the report of the project entitled ”IOT BASED REMOTE WEATHER CONTROLLING” is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of ”Mr. Sushant Kumar” (Dept. of Computer Science and Technology) and coordination of ”Dr. Mukesh Kumar Gupta”(HOD) and ”Ms. Anjana Sangwan” (Dept.of Computer Science and Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Computer Science and Technology. It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members **Signature**

Keshav Bhandari 

[17ESKCS079]

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Mr. Sushant Kumar. He has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Ms. Anjana Sangwan for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Prof. Dr. Mukesh Gupta, HOD, Department of Computer Science and Engineering, for facilitating, motivating and supporting us during each phase of development of the project.Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:

Keshav Bhandari (17ESKCS079)

Contents

List of Figures

List of Tables

Chapter 1

Project Chapter

1.1 Problem Statement and Objective

IoT based Remote Weather Controlling System is a project whose central task is to remotely monitor and control the weather conditions in order to maintain a specific required range of temperature and humidity. The core idea of the project is to implement an intelligent and smart monitoring and controlling with less user interaction as possible.

1.2 Investigation and Analysis

Investigation and Ananlysis shows that this project can handle real-time issues and provide efficient solutions for them.

•**Automation and Digitalization:** Automation is done which reduces required human resource with an ideal solution providing remote monitoring and controlling.

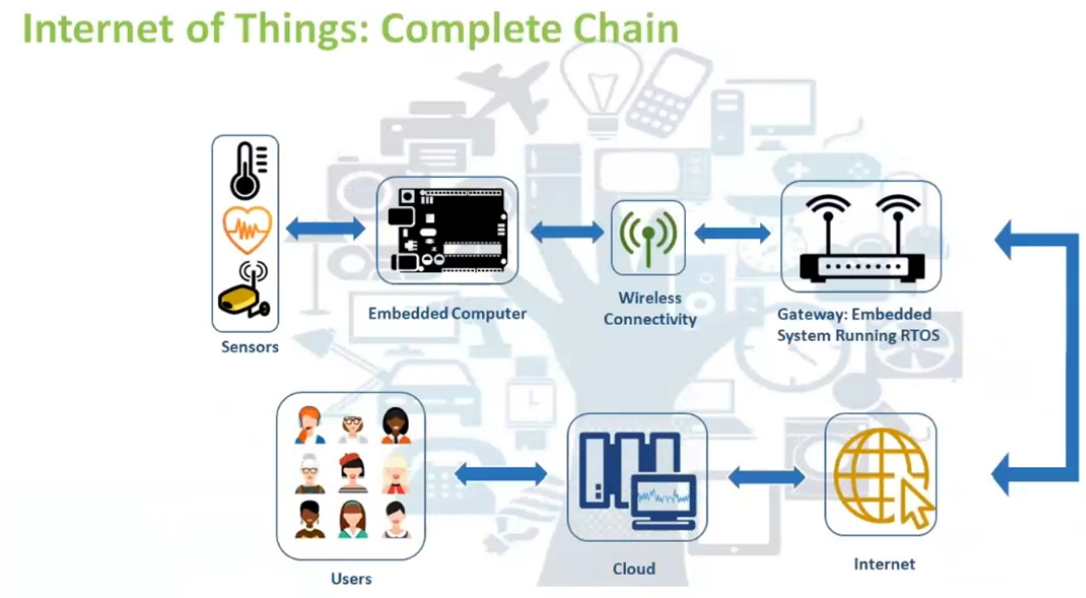
•**Risk Minimization:** It eradicates the risk of accidents that may occur in case the temperature and humidity conditions are not met.

1.3 Introduction to Project

IoT based Remote Weather Controlling System remotely monitors and control the weather conditions in order to maintain a specific required range of temperature and humidity. It implements an intelligent and smart monitoring and controlling with less user interaction as possible.

Its extra-ordinary feature is to maintain your industry temperature and humidity requirements within the entire industry area range through IoT to save time, human resource and to get good results in reduced efforts. It also minimises the risk of accidents that may occur in case the conditions are not met and thus, it is a system that holds a great significance.

1.4 Proposed Solution and Device



This diagram represents the entire IoT chain that has been involved in the project that depicts the perspective of the product. So, the data is read through sensors by the use of embedded system and is sent to the gateway through wireless connectivity.

From the gateway, the data is sent to the network server through internet and then to the application server to the users.

1.5 Scope of the Project

IoT based Remote Weather Controlling System will use Arduino sensors to detect temperature and humidity. It updates the sensor values at regular intervals and based on that turn on or off the relay devices (air conditioners, heaters, etc.) in order to maintain the condition range with least interaction with user. The project covers Remote Sensing, Remote Monitoring, Automation, event based Action Triggering and Large Area Covering through LoRa. This project has a wide scope of application. Applicable in Industries, Hospitals, Storage-houses, Home, Swimming pools, Oven, etc.

Chapter 2

Software Requirement Specification

2.1 Overall Description

IoT based Remote Weather Controlling System is a project whose central task is to remotely monitor and control the weather conditions in order to maintain a specific required range of temperature and humidity. The core idea of the project is to implement an intelligent and smart monitoring and controlling with less user interaction as possible.

Its extra-ordinary feature is to maintain your industry temperature and humidity requirements within the entire industry area range through IoT to save time, human resource and to get good results in reduced efforts. It also minimises the risk of accidents that may occur in case the conditions are not met and thus, it is a system that holds a great significance.

2.1.1 Product Perspective

2.1.1.1 System Interfaces

1) Read Sensor data:

-Temperature

- Humidity

2) Data uplink to application server (Ubidots)

3) Dashboard for monitoring temperature and humidity values

4) Relay Control through downlinks based on Trigger Actions and Events

5) Automation and Remote Monitoring

2.1.1.2 User Interfaces

User interface is the platform through which user can easily interact with the system. In our application, we have a dashboard that is user friendly and can be accessed through login id and password through Ubidots.

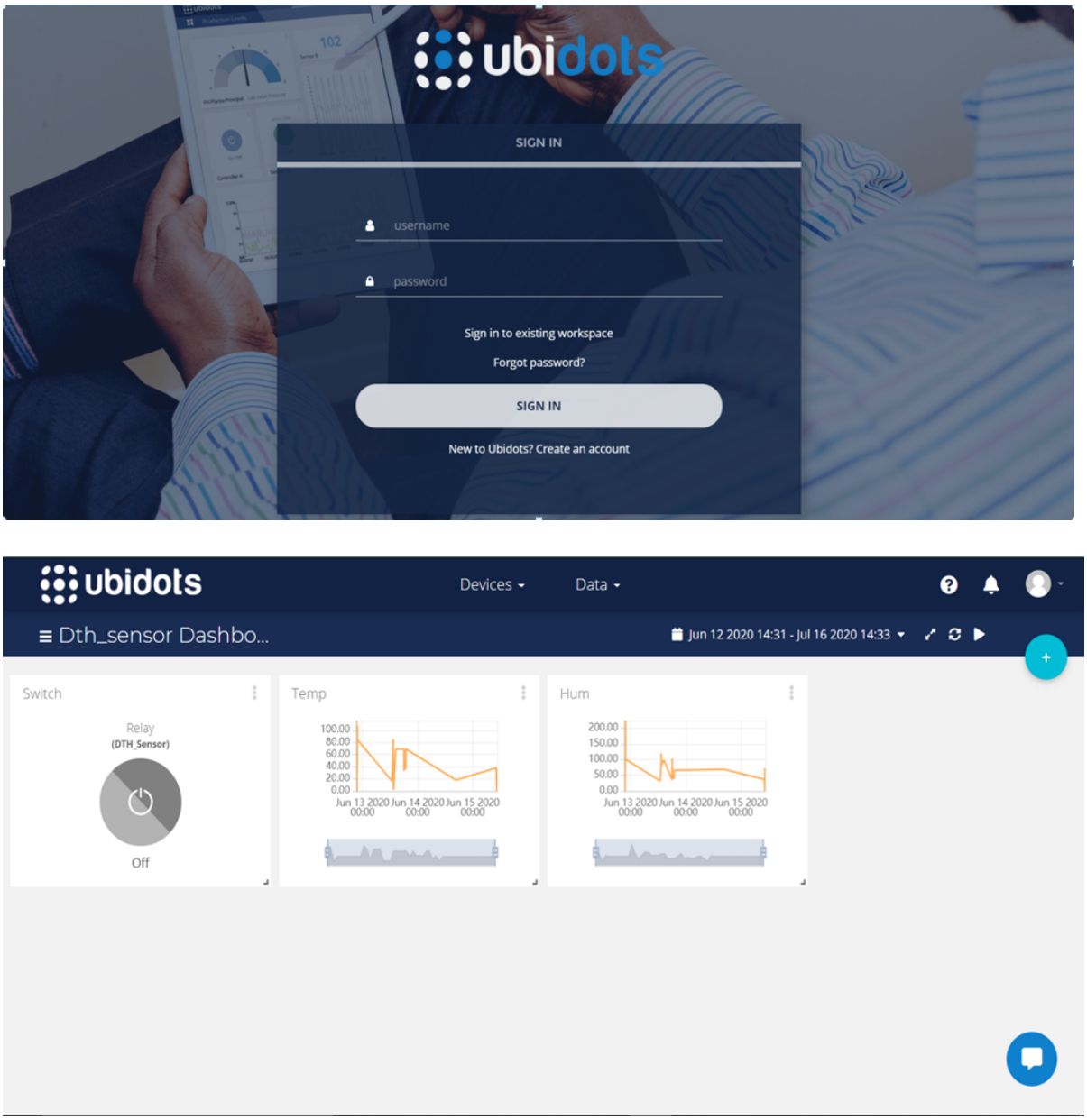


Figure 2.1: Snaps of User Interfaces

2.1.1.3 Hardware Interfaces

• **LG-01N Single Channel LoRa Gateway:** It is the hardware interface required for sending data from Arduino to TTN.

• **LoRa End Node:** It is our (Arduino UNO + LoRa Shield) on which the sensors are mounted.

• **DHT11 Sensor:** It is a temperature and humidity sensor.

• **Antenna:** It is attached with the gateway to increase the range.

• **Relay:** It is an actuator that helps us turn (Air Conditioners, heater, etc.) on or off.

• **Connecting Wires:** The connecting wires are required for connecting sensors to the Arduino board.

• **USB Cable:** To connect Arduino to laptop in order to upload our program to it.

• **Power Supply:** It is required to power our end node.

2.1.1.4 Software Interfaces

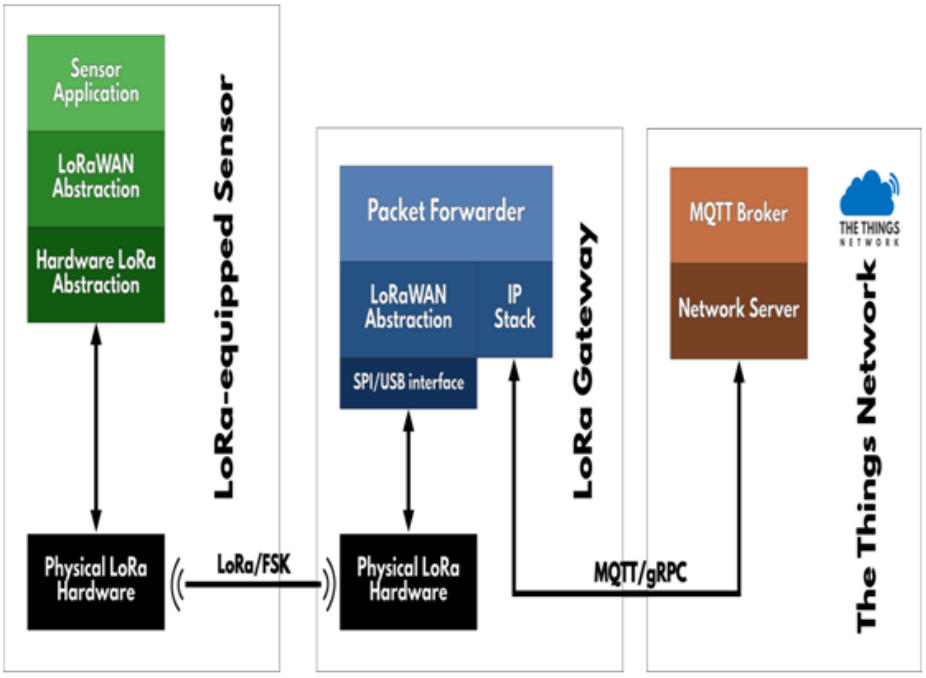
• **Sensor Interfacing:** Arduino Software IDE

• **Network Server:** The Things Network (TTN)

• **Application Server:** Ubidots Application

• **Web Browser:** web browser for running Ubidots (Chrome, Firefox, etc.)

2.1.1.5 Communications Interfaces



2.1.1.6 Memory Constraints

The number of uplinks that Ubidots can handle is limited to 10000 and since, it is a prototype, that amount of uplinks and downlinks are sufficient in order to serve our purpose and since everything is based on cloud, the memory constraints differ from cloud to cloud.

2.1.1.7 Operations

Hardware components of environment of our application include Arduino and sensors for calculating temperature and humidity. At extreme hot or cold conditions, Arduino will get weathered off so one should avoid extreme conditions.

Software components of our application include Arduino Software IDE for Arduino and The Things Network (TTN) and Ubidots are network server and application server respectively. For them, good high speed internet stability is required or else loss of packets can occur.

2.1.1.8 Project Functions

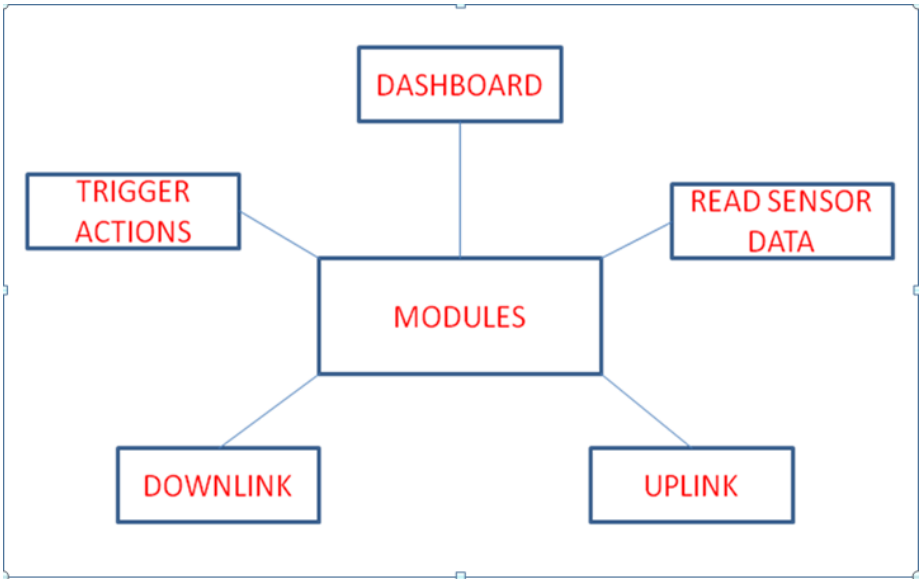
1) Read Sensor data:
- Temperature
- Humidity

2) Data uplink to application server (Ubidots)

3) Dashboard for monitoring temperature and humidity values

4) Relay Control through downlinks based on Trigger Actions and Events

5) Automation and Remote Monitoring



2.1.1.9 User Characteristics

The user documentation will include a manual for managers, employee as well as technicians. User manual for this application will guide them how to exactly use our application interface and produce more output. It will also teach that how one gets notified in case there is a fault. Manual of user will include the details about how and what are the features that are associated with our application and how one can use them in the most efficient way.

2.1.1.10 Constraints

• More complex to configure than wired network

• Affected by surroundings.
- Wall blocking
- Interference
- Attenuation (far distance)

• At extreme high or low temperatures, Arduino gets withered off

• Range bounded to few kilometers

• Bandwidth of the data transmitted is low

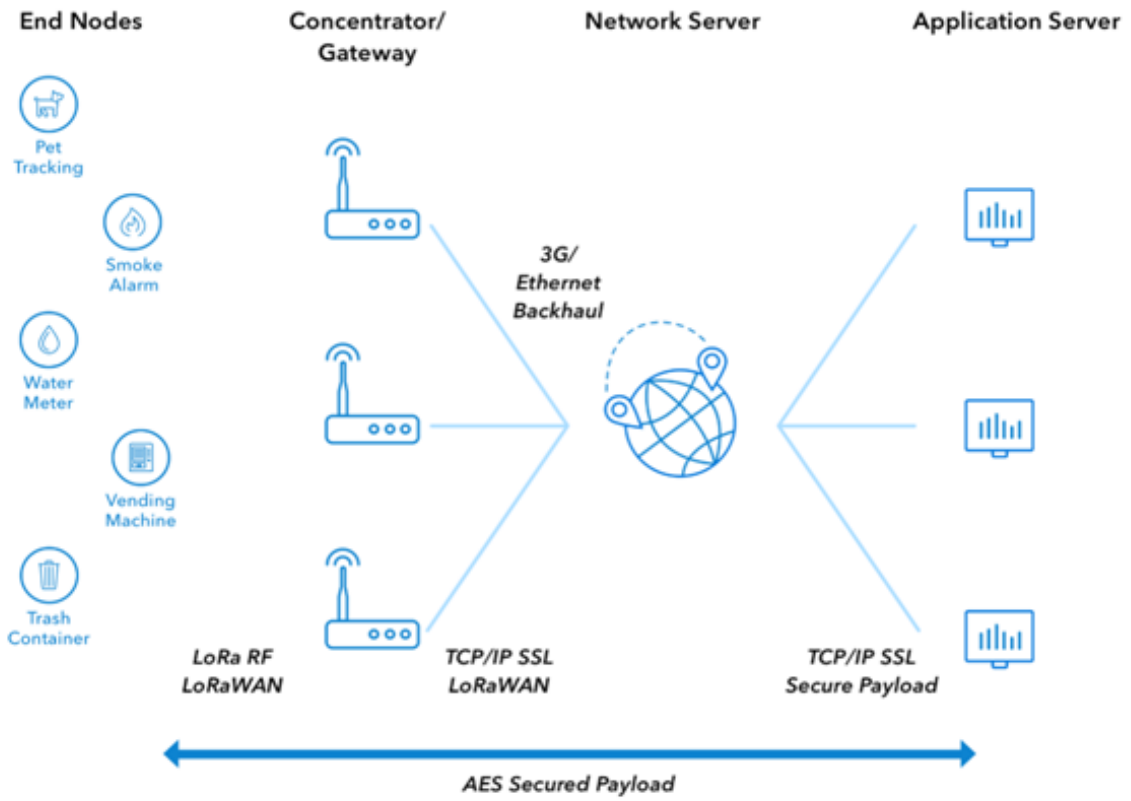
2.1.1.11 Assumption and Dependencies

It is assumed that the internet provided to the gateway has a good stability and the user should have a good internet connection in order to run the application. The person using this application is assumed to be the one who know the basics of computer and know how to operate an application.
Since our project is not dependent on any other project, there is no dependency.

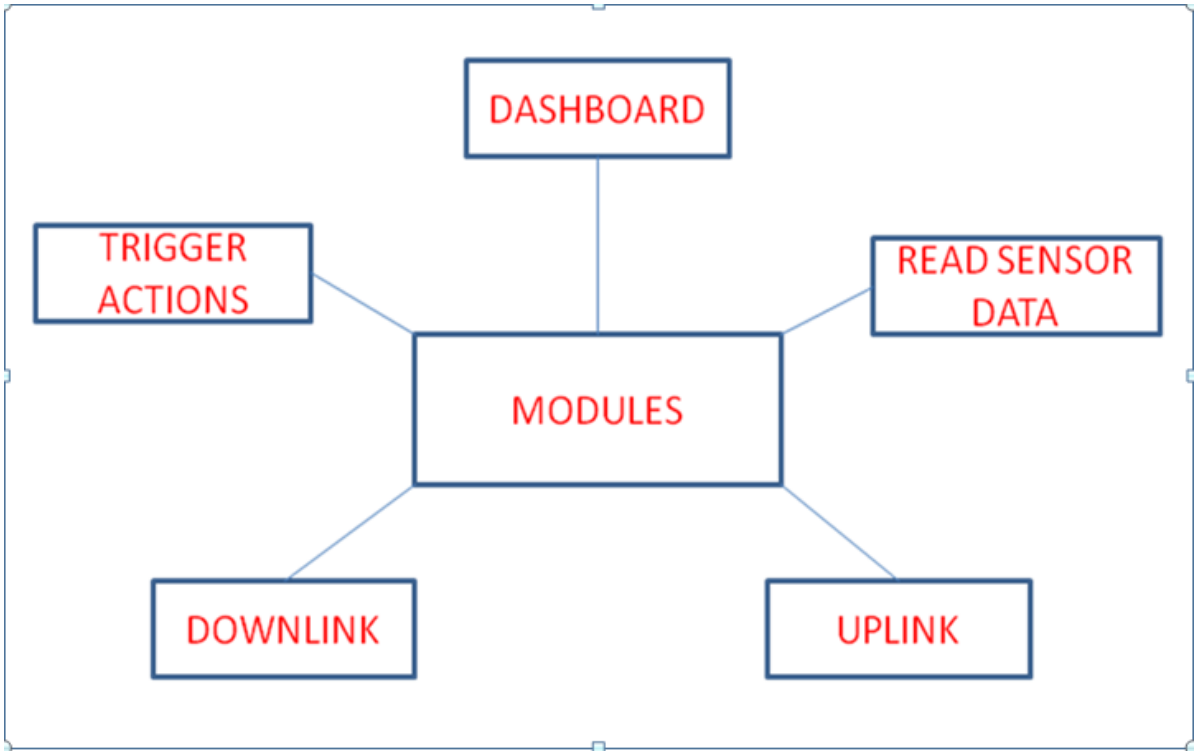
Chapter 3

SYSTEM DESIGN SPECIFICATION

3.1 System Architecture



3.2 Module Decomposition Description



3.3 High Level Design Diagrams

3.3.1 Activity Diagram (Login)

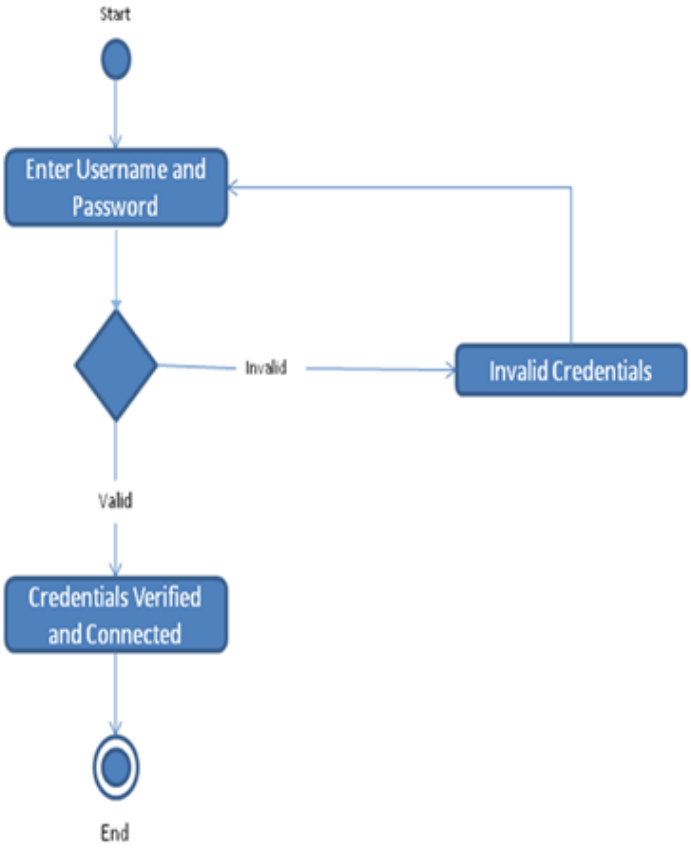


Figure 3.1: Login Activity Diagram

3.3.2 Activity Diagram (System)

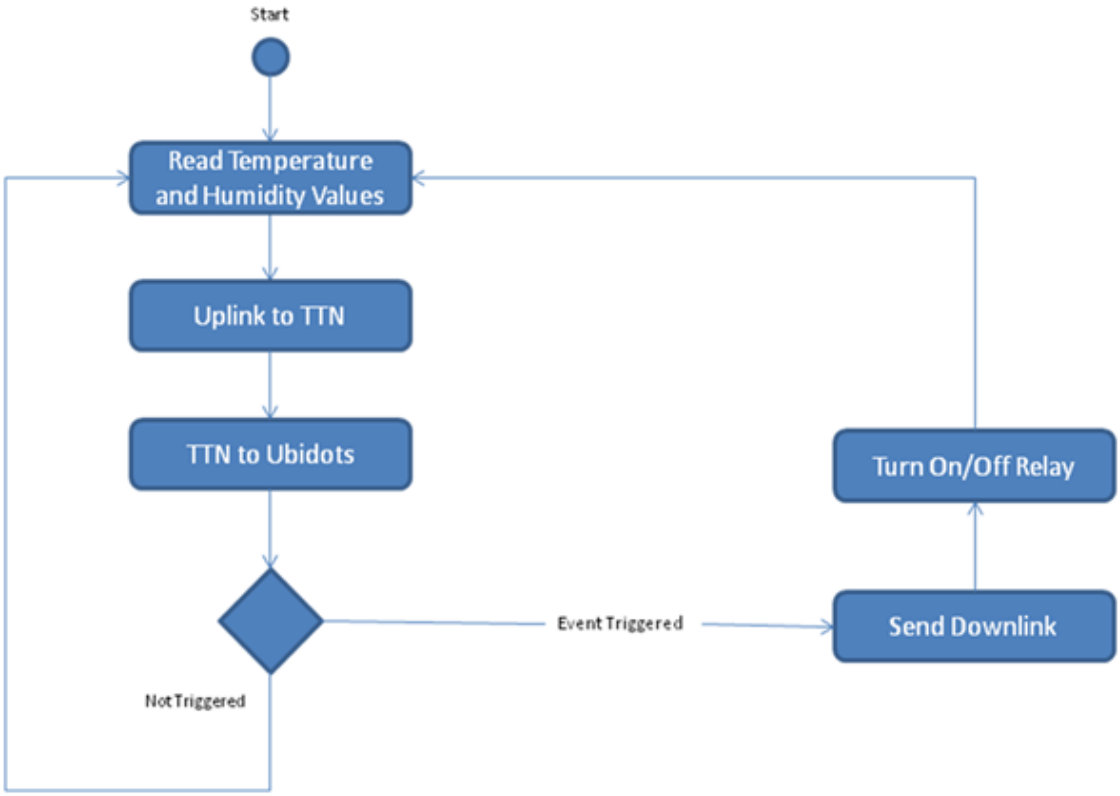


Figure 3.2: System Activity Diagram

Chapter 4

METHODOLOGY AND TEAM

4.1 Introduction to Waterfall Framework

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In ”The Waterfall” approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as an input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.

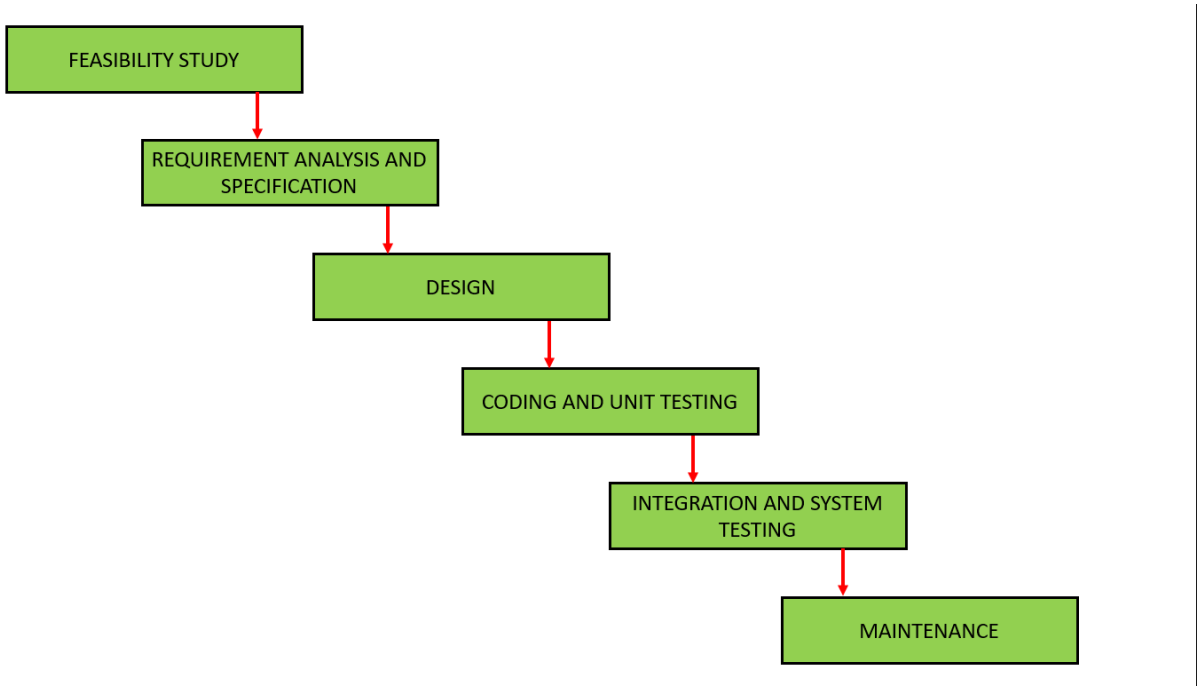


Figure 4.1: WaterFall model

The sequential phases in Waterfall model are-

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

3. **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

4. **Integration and Testing:** All the units developed in the imple-

mentation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of system:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

6. **Maintenance:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name ”Waterfall Model”. In this model phases do not overlap.

Waterfall Model Pros Cons

Advantage The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

4.2 Team Members, Roles & Responsibilities

Keshav Bhandari: Complete System (All modules)

Roles:

- Feasibility Study
- Requirement Analysis Specification
- Design
- Coding and Unit Testing
- Integration and System Testing
- Maintenance

Chapter 5

System Testing

The designed system has been testing through following test parameters.

5.1 Functionality Testing

In testing the functionality of the device, the following features were tested:

1. Uplinks
   (a) The uplinks are sent at regular intervals of time to check whether there is a loss of packets or not.
   (b) The response time between sensor data and the uplink is tested.
   (c) The uplink sent is received in the proper formats and desired place at network and application server.
2. Downlinks
   (a) Error message when exceeding the range of temperature and humidity by sending downlinks.

(b) Different conditions tested for downlinks so that in all the possible test cases, our system works properly.

(c) Testing for proper downlinks received in the required format at network and application server.

3. Dashboard

(a) Testing whether the temperature parameter is received and reflected perfectly on the dashboard.

(b) Testing whether the Humidity parameter is received and reflected perfectly on the dashboard.

(c) Testing that downlink relay is working perfectly and whether we are able to handle it through dashboard.

5.2 Performance Testing

- Testing for the loss of data packets.
- Testing for the connectivity of the internet and the speed of data packets transmission.
- Testing for the regular uplinks and downlinks for checking the performance of the system.
- Testing the updation of the temperarue and humidity values reflection on the dashboard.

5.3 Usability Testing

- The device used is portable enough to be moved into different segments of the industry.
- The equipment is smart enough to push not only the notifications but also the error messages, warnings etc.
- The system have an option to log all the events to provide clarity to the end users.
- The notifications are shown and handling of the display can be done properly in the devices [computers/mobile devices].
- Usability in terms of displaying data, processing data, pushing job tasks from the devices tested thoroughly.

Chapter 6

TEST EXECUTION SUMMARY

Execution Test Summary Report is an overall review of Testing Process from start to end. Test Plan comes at the starting of project while Test Summary Report comes at the end of the testing process.

The Test Summary Report contents are :

1. S.No.
2. No. of Uplinks Sent
3. No. of Uplinks Received
4. No. of Downlinks Sent
5. No. of Downlinks Received
6. Final Status

S.No	Uplinks Sent	Uplinks Received	Downlinks Sent	Downlinks Received	Final Status
1	1000	999	200	200	PASS
2	1500	1500	500	500	PASS
3	750	750	150	149	PASS
4	1250	1250	700	700	PASS
5	2000	2000	1200	1199	PASS

Table 6.1: Text Execution Summary Table

Chapter 7

PROJECT SCREENSHOTS

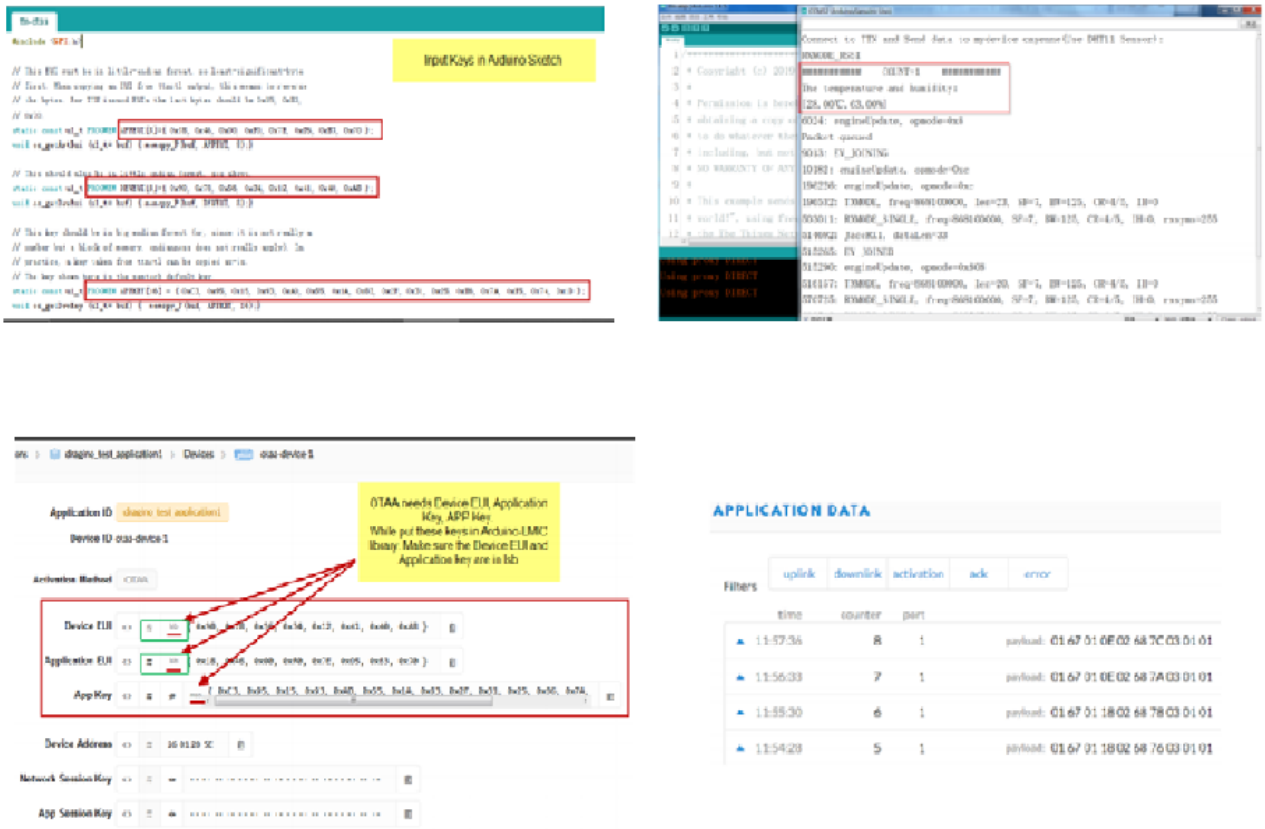


Figure 7.1: Project Screenshots

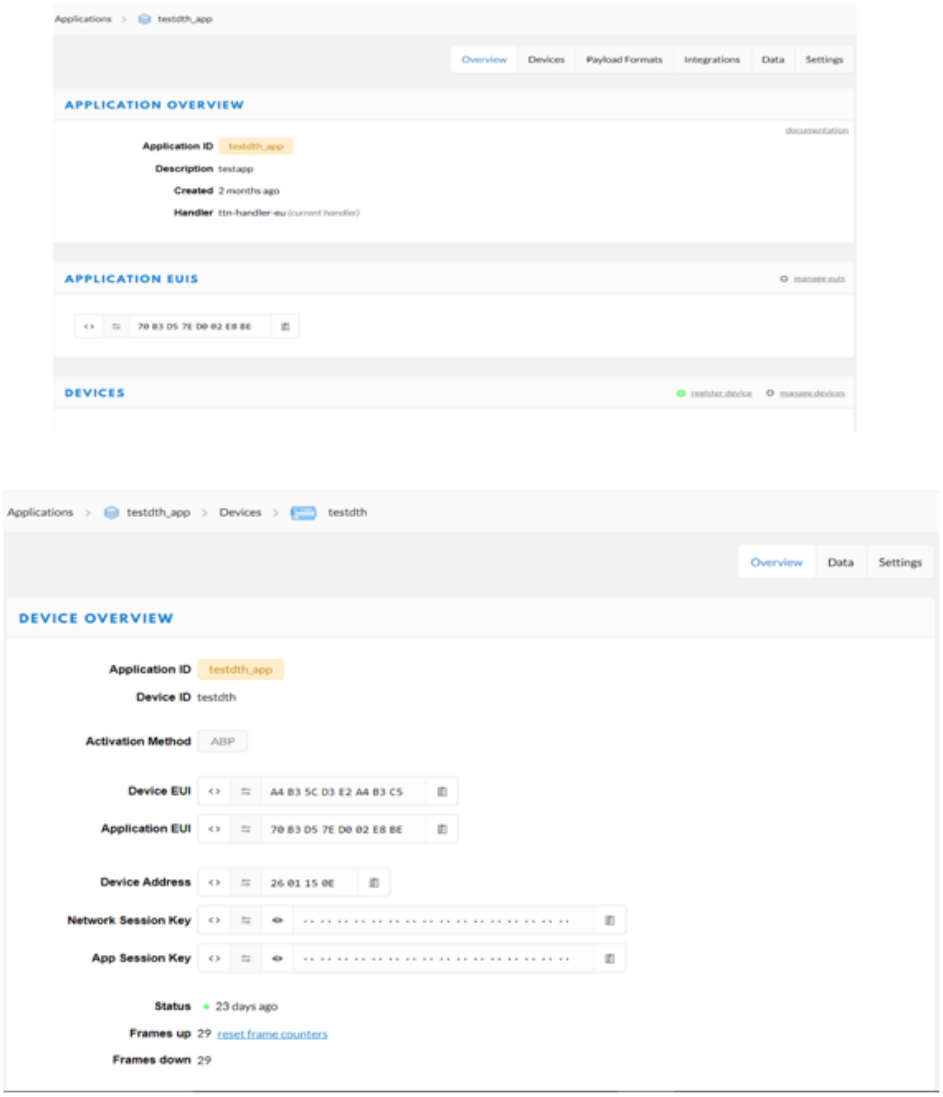


Figure 7.2: Project Screenshots

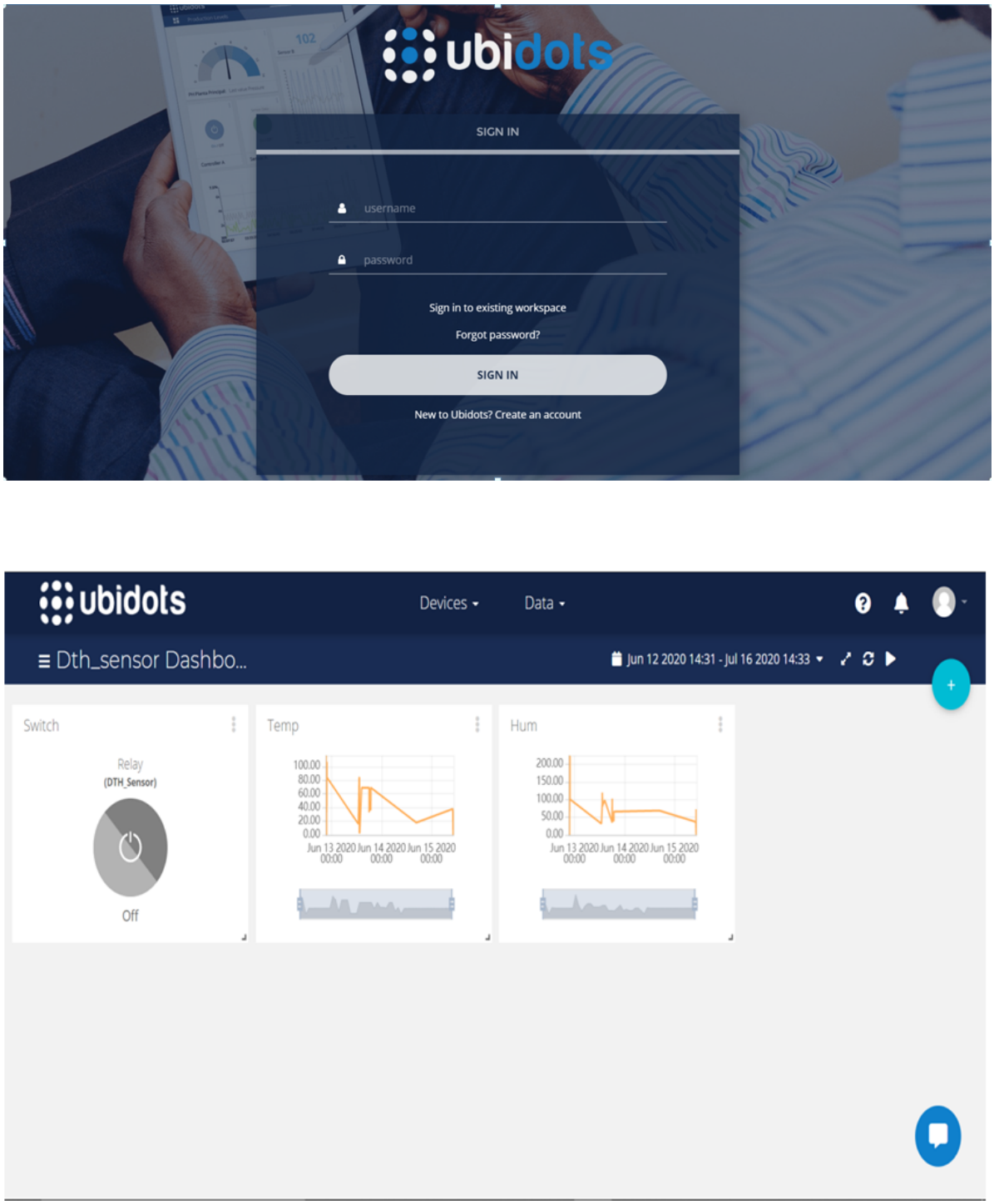


Figure 7.3: Project Screenshots

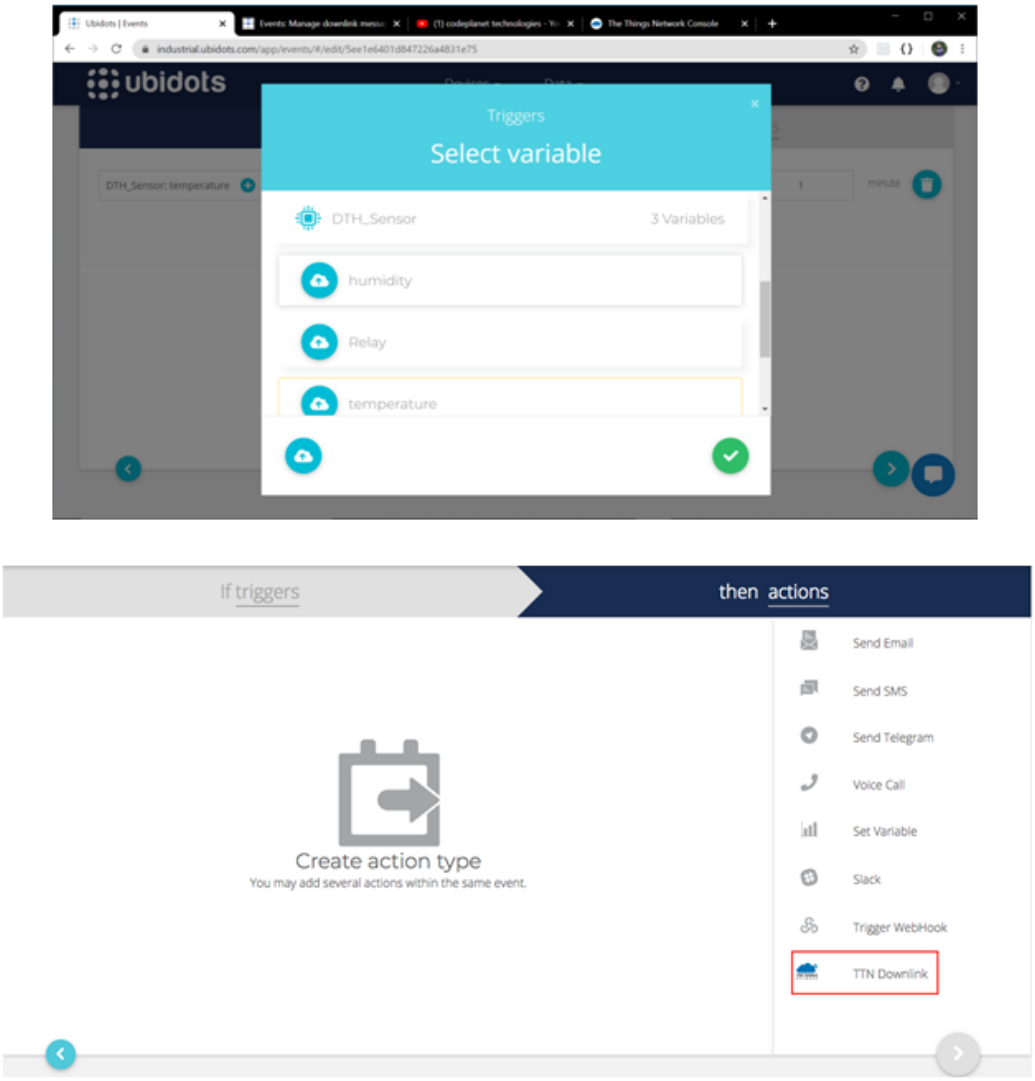


Figure 7.4: Project Screenshots

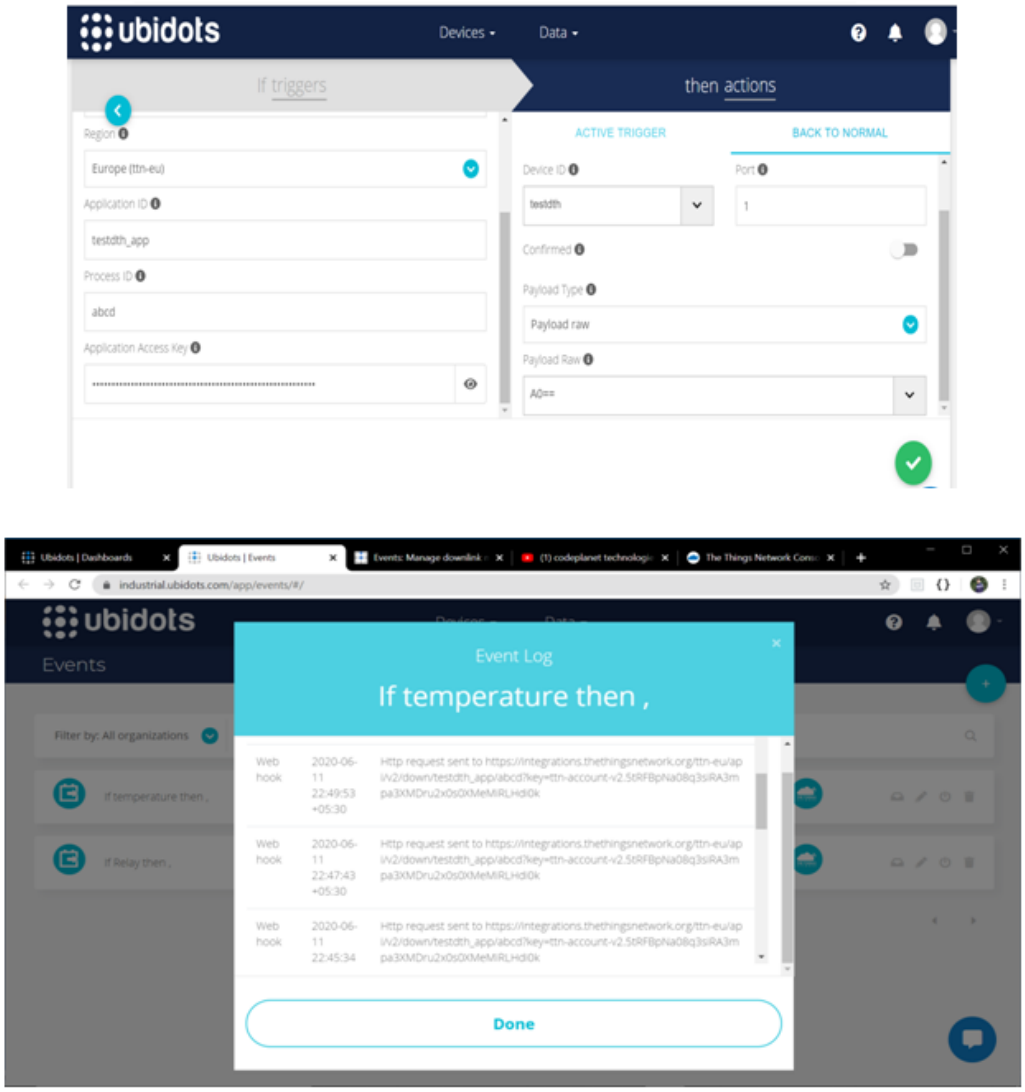


Figure 7.5: Project Screenshots

Chapter 8

PROJECT SUMMARY AND CONCLUSIONS

8.1 Conclusion

The project is considerable with a good number of parameters. Here are some parameters that describe the purpose of how this project can handle real-time issues and provide efficient solutions for them.

- **Automation and Digitalization:** Automation is done which reduces required human resource with an ideal solution providing remote monitoring and controlling.
- **Risk Minimization:** It eradicates the risk of accidents that may occur in case the temperature and humidity conditions are not met.

Chapter 9

FUTURE SCOPE

IoT based Remote Weather Controlling System will use Arduino sensors to detect temperature and humidity. It updates the sensor values at regular intervals and based on that turn on or off the relay devices (air conditioners, heaters, etc.) in order to maintain the condition range with least interaction with user.
The project covers:

- Remote Sensing
- Remote Monitoring
- Automation
- Event Based Action Triggering
- Large Area Covering through LoRa.

This project has a wide scope of application. Applicable in Industries, Hospitals, Storage-houses, Home, Swimming pools, Oven, etc.

References

[1] *Single Channel LoRa Iot Kit User Manual (Dragino)*

http://www.dragino.com/downloads/index.php?dir=LoRa_IoT_Kit/v2-Kit/&file=Single%20Channel%20LoRa%20IoT%20Kit%20v2%20User%20Manual_v1.0.5.pdf

[2] *Ubidots IoT Server Connection*

http://www.dragino.com/downloads/index.php?dir=LoRa_IoT_Kit/v2-Kit/&file=Ubidots_IoT_Server_Connection.pdf

[3] *TTN Docs*

https://www.thethingsnetwork.org/docs/

[4] *SRS IEEE Standard*

https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc#:~:text=IEEE%20Software%20Requirements%20Specification%20Template&text=%3CIdentify%20the%20product%20whose%20software,system%20or%20a%20single%20subsystem.%3E