A Software Requirements Specification

On

IoT based

Remote Weather Controlling

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Computer Science and Engineering from Rajasthan Technical University, Kota



**MENTOR: SUBMITTED BY:**

**Mr. Sushant Kumar Keshav Bhandari (17ESKCS079)**

(Dept. of Computer Science & Engineering)

**COORDINATOR:**

**Dr. Mukesh Gupta [HOD]**

(Dept. of Computer Science & Engineering)

**Mrs. Anjana Sangwan**

(Dept. of Computer Science & Engineering)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN**

**Ramnagaria (Jagatpura), Jaipur – 302017**

**SESSION 2020-21**

Table of Contents

[**1. Introduction** 3](#_Toc50832196)

[1.1 Purpose 3](#_Toc50832197)

[1.2 Intended Audience and Reading Suggestions 3](#_Toc50832198)

[1.3 Scope 4](#_Toc50832199)

[1.4 Document Conventions 4](#_Toc50832200)

[1.5 References 5](#_Toc50832201)

[**2. Overall Description** 6](#_Toc50832202)

[2.1 Product Perspective 6](#_Toc50832203)

[2.2 Product Functionality 6](#_Toc50832204)

[2.3 Operating Environment 7](#_Toc50832205)

[2.4 Design and Implementation Constraints 8](#_Toc50832206)

[2.5 User Documentation 8](#_Toc50832207)

[2.6 Assumptions and Dependencies 8](#_Toc50832208)

[**3. External Interface Requirements** 9](#_Toc50832209)

[3.1 User Interfaces 9](#_Toc50832210)

[3.2 Hardware Interfaces 10](#_Toc50832211)

[3.3 Software Interfaces 12](#_Toc50832212)

[3.4 Communication Interfaces 13](#_Toc50832213)

[**4. System Features** 14](#_Toc50832214)

[4.1 Read Sensor Data 14](#_Toc50832215)

[4.2 Uplink 14](#_Toc50832216)

[4.3 Dashboard 14](#_Toc50832217)

[4.4 Trigger Actions 15](#_Toc50832218)

[4.5 Downlink 15](#_Toc50832219)

[**5. Other Non-Functional Requirements** 16](#_Toc50832220)

[5.1 Performance Requirements 16](#_Toc50832221)

[5.2 Safety Requirements 16](#_Toc50832222)

[5.3 Security Requirements 16](#_Toc50832223)

[5.4 Software Quality Attributes 17](#_Toc50832224)

[**6. Other Requirements** 18](#_Toc50832225)

[6.1 Appendix A: Glossary 18](#_Toc50832226)

[6.2 Appendix B: Analysis Models 18](#_Toc50832227)

# Introduction

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.

## Purpose

The purpose regarding to this 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.

## Intended Audience and Reading Suggestions

Our intended audience includes developers, testers, coders and business people in industrial sector, or any other sector that finds this project useful. This report can be read in manner by firstly reading the introduction of the report and then proceeding in the same order as described in the table of contents. Developers may focus on the parts that contain the diagrams and design of the project.

## 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 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.

## Document Conventions

* **Font Type**: Times New Roman
* **Heading 1**: Font Size (20)

Bold

Centre- Aligned

* **Heading 2**: Font Size (16)

Bold

Left-Aligned

* **Heading 3**: Font Size (12)

Bold

Left-Aligned

* **Content**: Font Size(12)
* **Line Spacing**: 1.5

## References

* 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>

* Ubidots IoT Server Connection

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

* TTN Docs

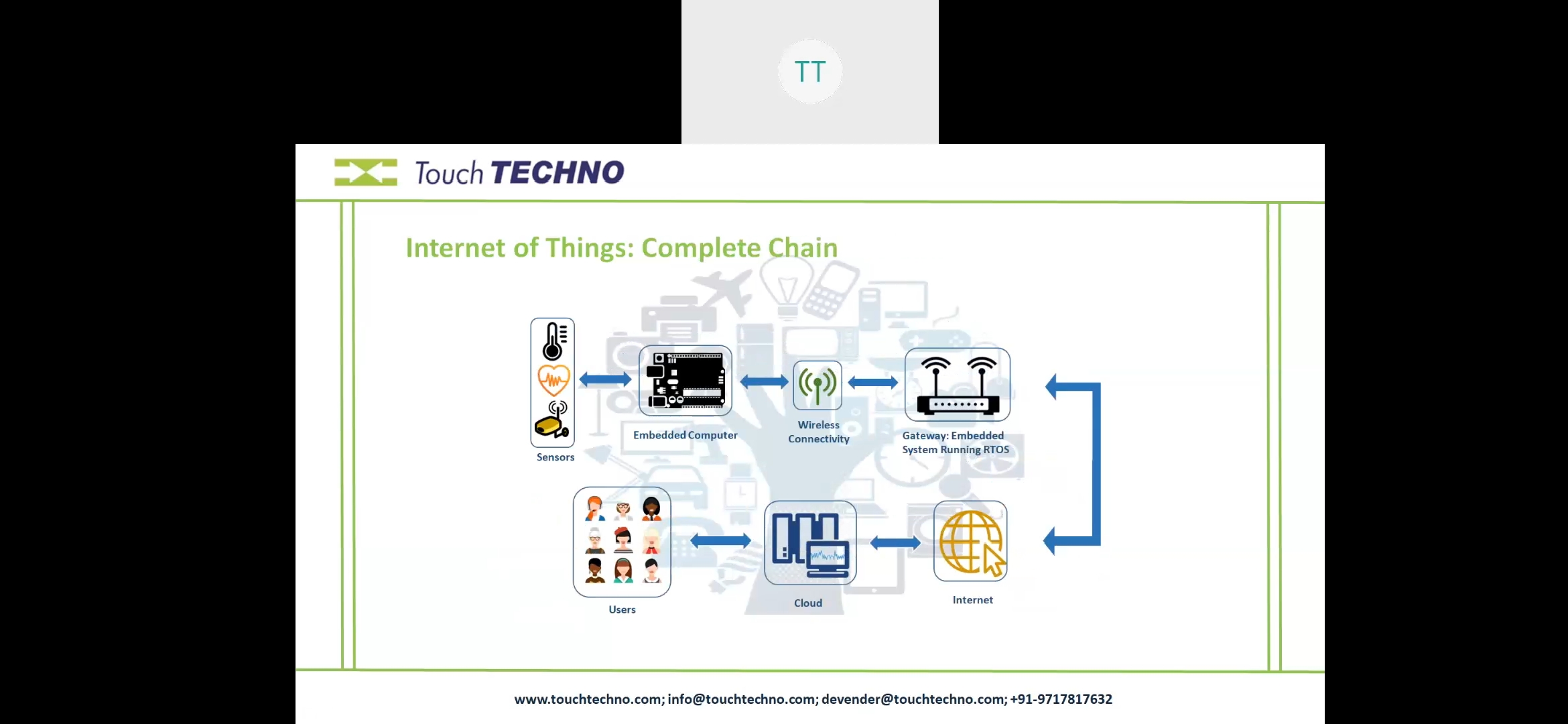
<https://www.thethingsnetwork.org/docs/>

* 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>

# Overall Description

## Product Perspective

****

**Fig 2.1: Product Perspective**

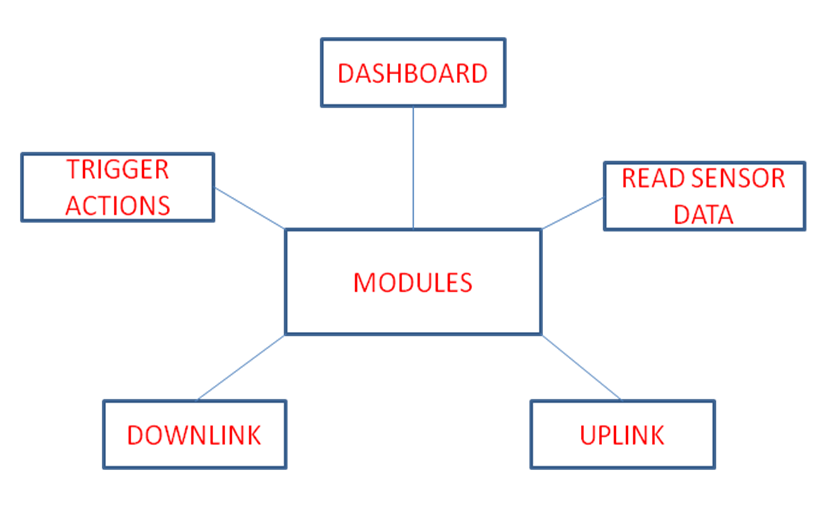
This diagram represents the entire IoT chain that has been involved in the project that depicts the perspective of the product. So, the data is read through sensors by the use of embedded system and is sent to the gateway through wireless connectivity. From the gateway, the data is sent to the network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

## Product Functionality

1. Read Sensor data:

* Temperature
* Humidity

1. Data uplink to application server (Ubidots)
2. Dashboard for monitoring temperature and humidity values
3. Relay Control through downlinks based on Trigger Actions and Events
4. Automation and Remote Monitoring



## Operating Environment

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.

## Design and Implementation Constraints

* More complex to configure than wired network
* Affected by surroundings.
* Wall blocking
* Interference
* Attenuation (far distance)
* Arduino is susceptible to surrounding conditions.
* At extreme high or low temperatures, Arduino gets withered off
* Range bounded to few kilometers
* Bandwidth of the data transmitted is low

## User Documentation

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.

## Assumptions and Dependencies

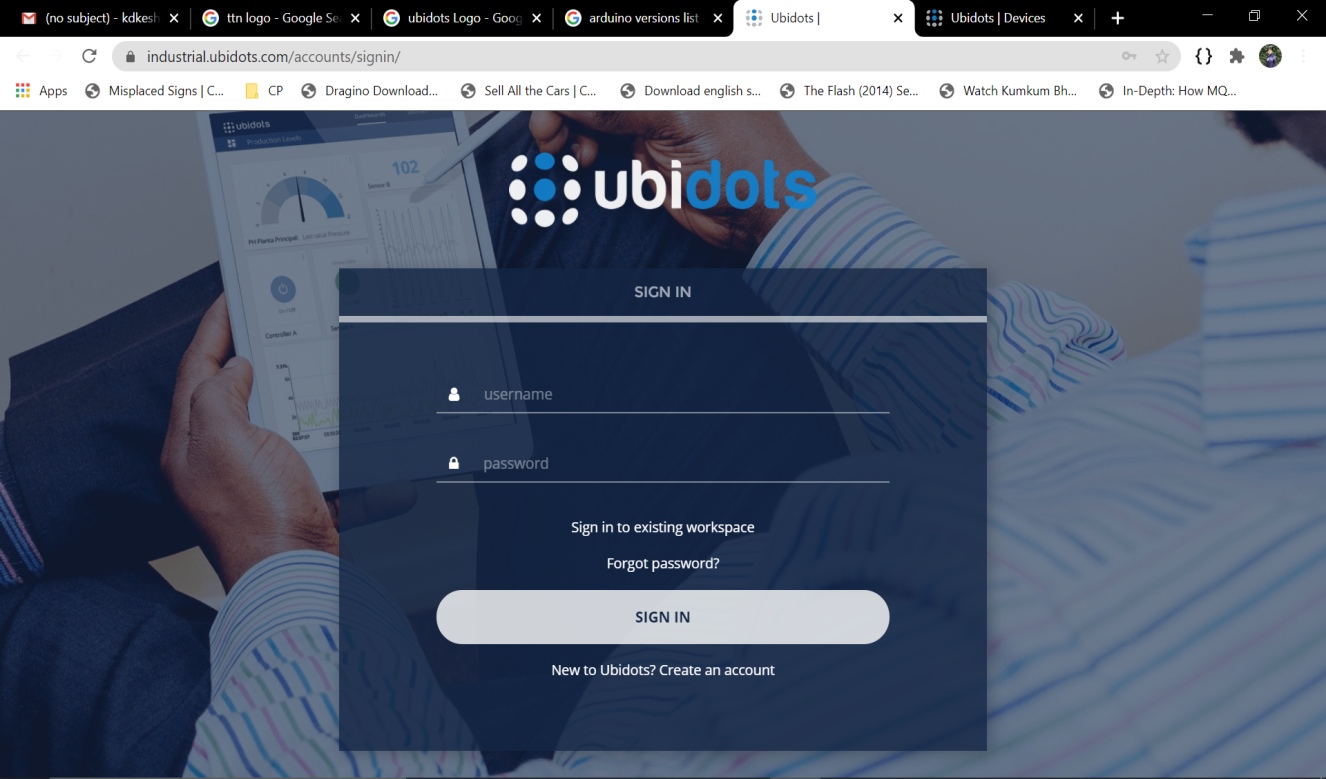
It is assumed that the internet provided to the gateway has a good stability and the user should have a good internet connection in order to run the application. The person using this application is assumed to be the one who know the basics of computer and know how to operate an application.

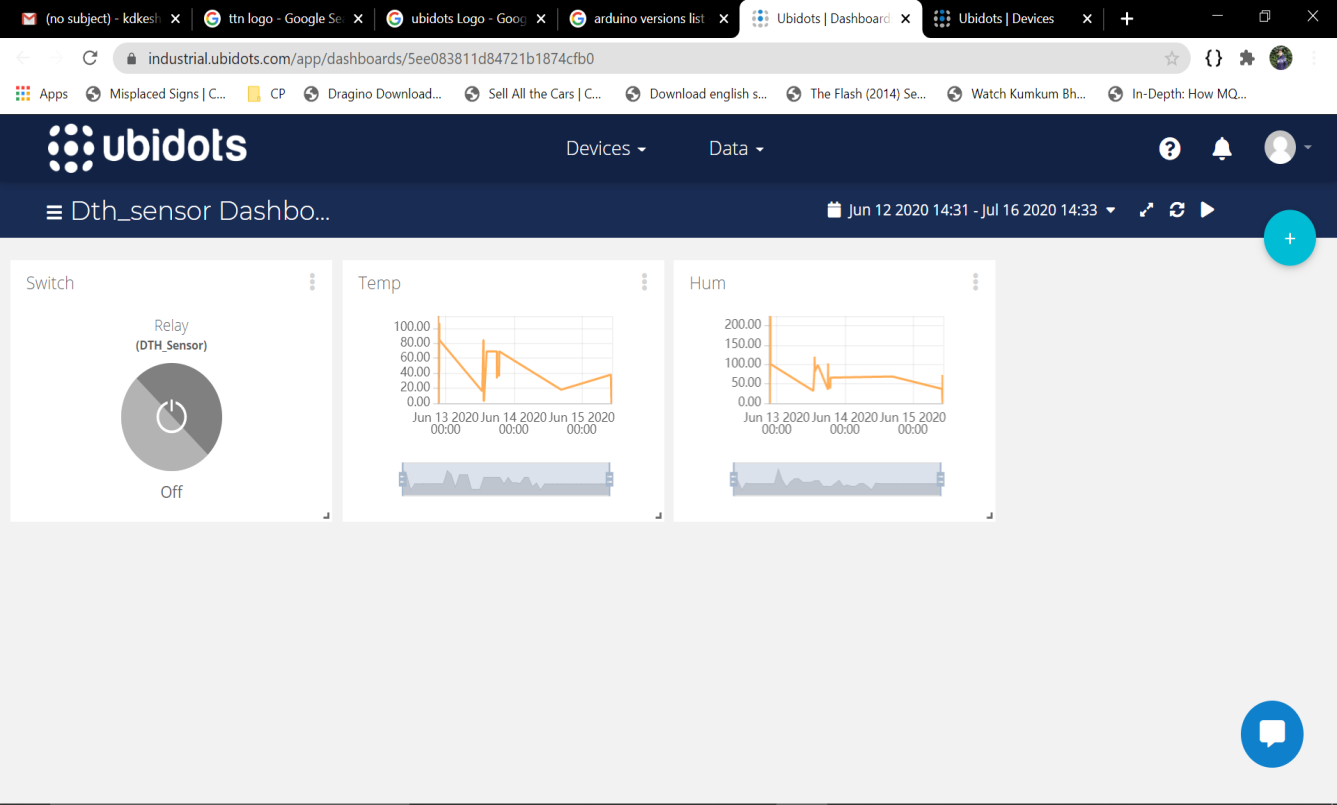
Since our project is not dependent on any other project, there is no dependency.

# External Interface Requirements

## 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.



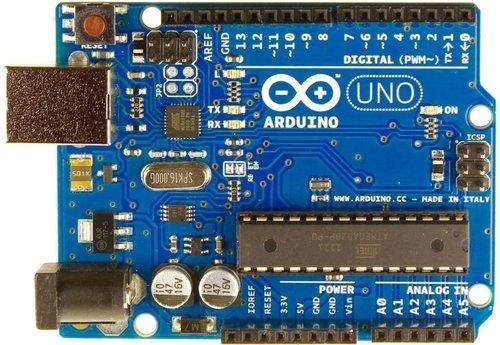


## Hardware Interfaces

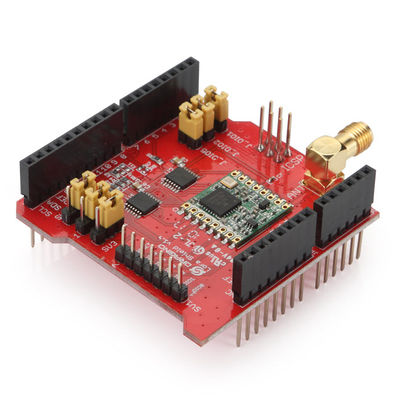
* **LG-01N Single Channel LoRa Gateway:** It is the hardware interface required for sending data from Arduino to TTN.



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

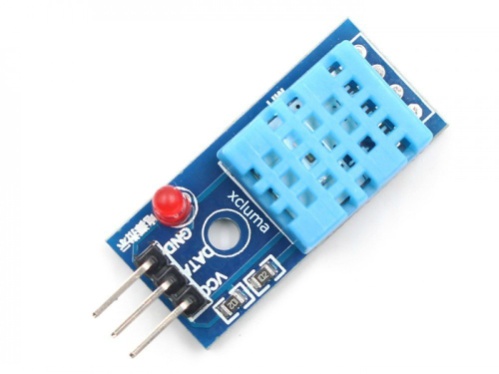


(Arduino UNO)

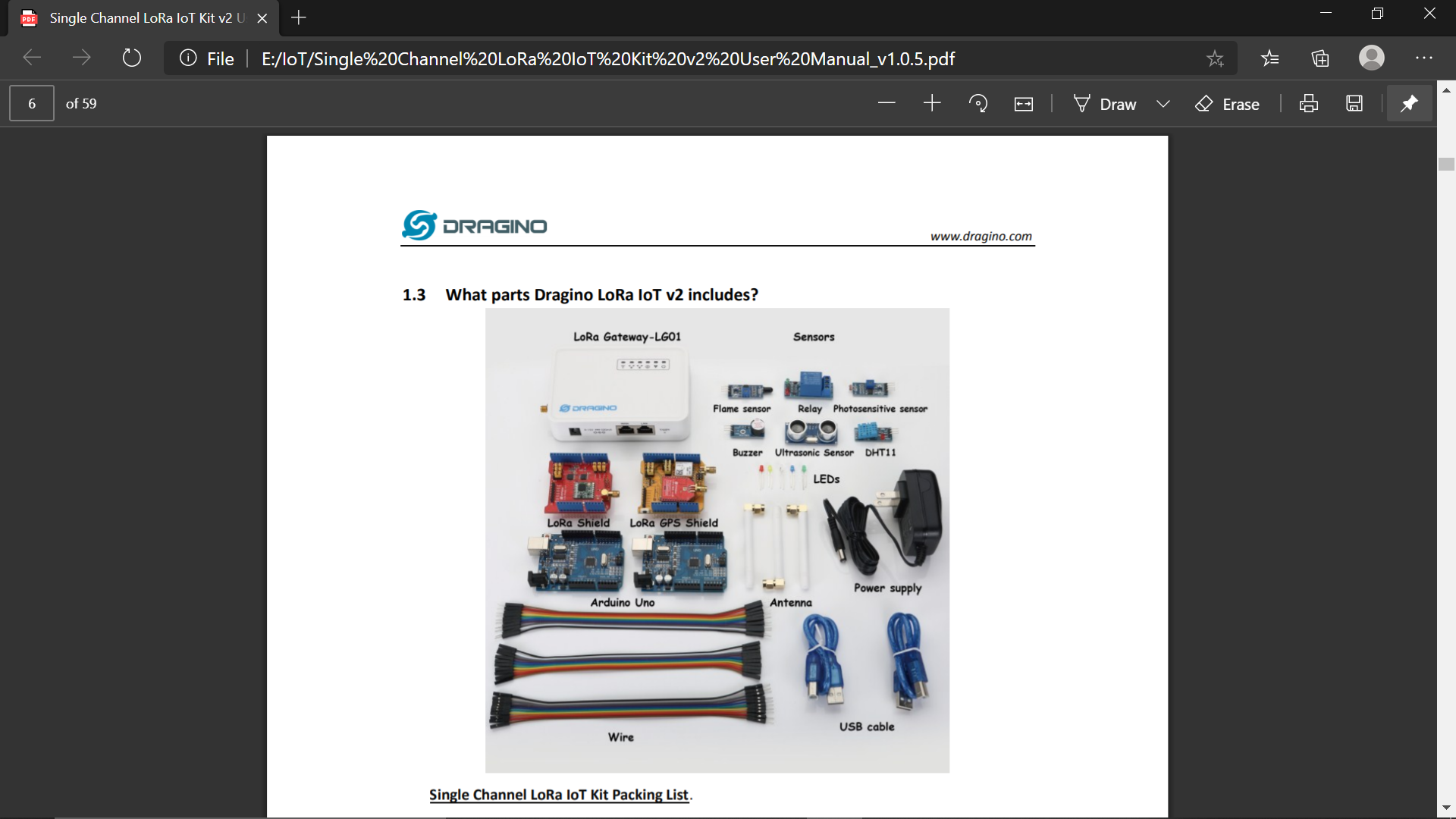


(LoRa Shield)

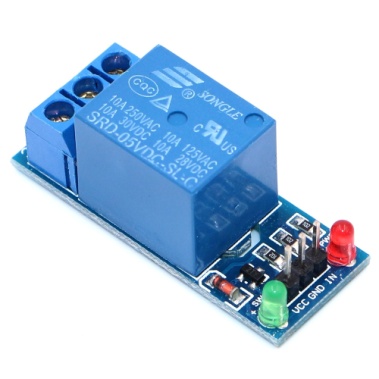
* **DHT11 Sensor:** It is a temperature and humidity sensor.



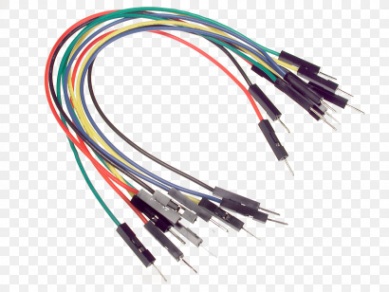
* **Antenna**: It is attached with the gateway to increase the range.

****

* **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.



## Software Interfaces

* **Sensor Interfacing:** Arduino Software IDE



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



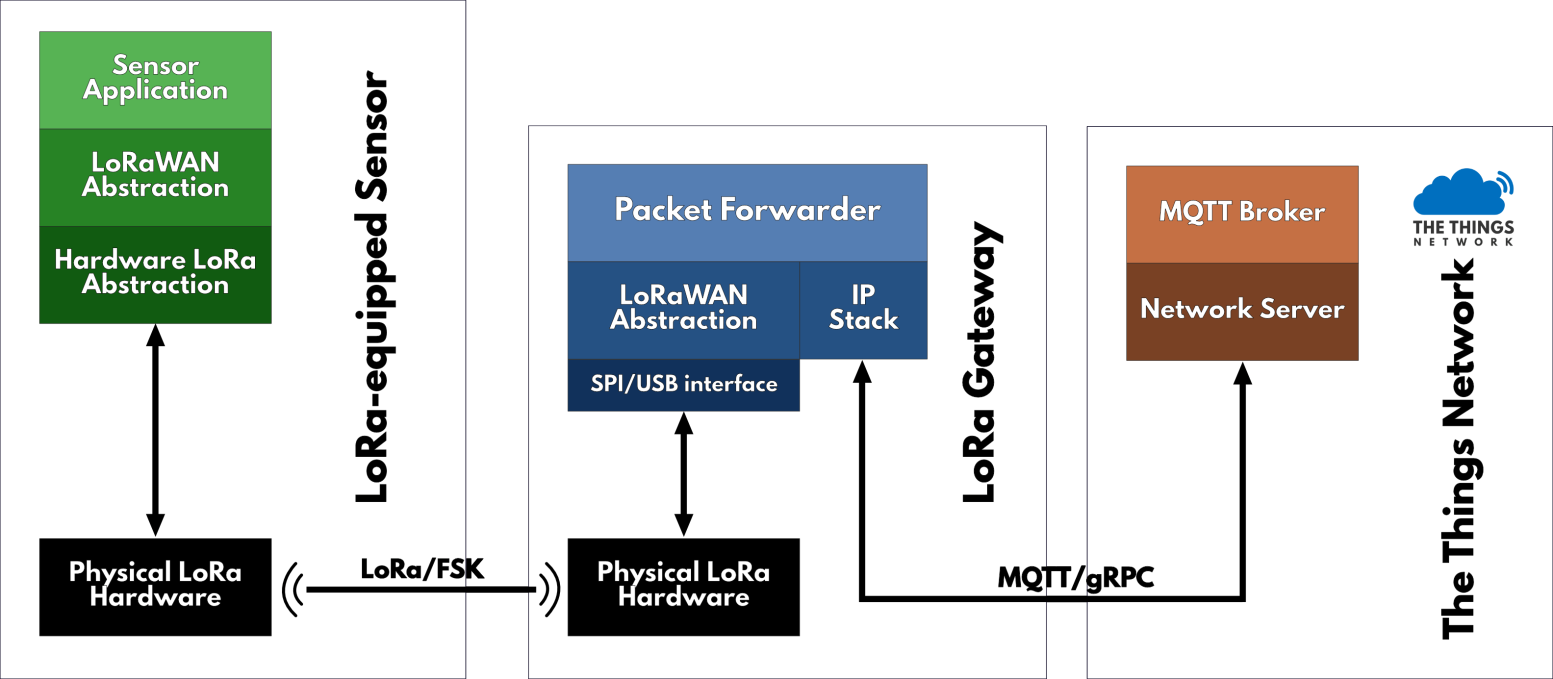
* **Application Server:** Ubidots Application



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



## Communication Interfaces

****

# System Features

## Read Sensor Data

DHT11 Sensor calculates the temperature and humidity values and these values are to be read by the help of Arduino UNO.

Steps:

1. Connect the corresponding pins to Arduino through connecting wires
2. Write the corresponding code in Arduino software IDE
3. Connect the Arduino to computer with the help of USB cable
4. Upload the code to Arduino
5. Sensor values are read by the Arduino

## Uplink

The sensor values read by the Arduino is to be sent to TTN and then to Ubidots with the help of LoRa shield.

Steps:

1. Configure the gateway
2. Mount LoRa shield on Arduino for transmitting data to gateway
3. With the help of gateway, data sent to TTN
4. Write corresponding payload
5. Integrate TTN with Ubidots
6. Data sent to Ubidots

## Dashboard

For monitoring the temperature and humidity in an efficient and user-friendly way, a dashboard is taken into account.

Steps:

1. Create a new dashboard on Ubidots
2. Add our device on that
3. Add the corresponding parameters on the Dashboard
4. Add one relay on Dashboard for downlink

## Trigger Actions

In order to control and maintain our required temperature and humidity range, we need to trigger our relay.

Steps:

1. Create events in Ubidots
2. Select the variables for triggering actions
3. Set the ‘if’ conditions for specifying when the action to be triggered
4. Set what action to be performed in case ‘if’ condition satisfies
5. Set when to get back to normal conditions
6. Based on above triggering, relay is turned on or off

## Downlink

In order to turn on or off the relay, downlink needs to be sent that gives the end node a message to turn it on or off.

Steps:

1. When trigger event occurs, downlink is sent
2. How the end node respond to different downlinks is set
3. The downlink is confirmed
4. Action corresponding to given downlink is performed

# Other Non-Functional Requirements

## 5.1 Performance Requirements

In order to run our application effectively, the performance requirements are as follows:

* A good internet connection for our application to run smoothly
* Internet stability should be there. Absence of this, may lead to loss of data packets.
* The Arduino should be kept safe from extreme physical conditions
* The battery need to be changed in case it drains out

## 

## 5.2 Safety Requirements

Since our application requires controlling the conditions, in case the device stops working, the chances of accident may occur which is hardly the case. But in that case, our system notifies the concerned person through mail or message and then the issue can be resolved out.

Moreover, since the relay gets heated up, one should never touch it or it may prove to be fatal since there is great chances of electric shock.

## 5.3 Security Requirements

* In order to maintain security, 128-bit encryption is done
* OTAA Activation dynamically changes the keys
* Frame Counters are used in order to stop re-sending attacks
* Every device has a unique Device ID, Apps Key, Network Session Key

## 5.4 Software Quality Attributes

The various quality attributes are there in this project such as:

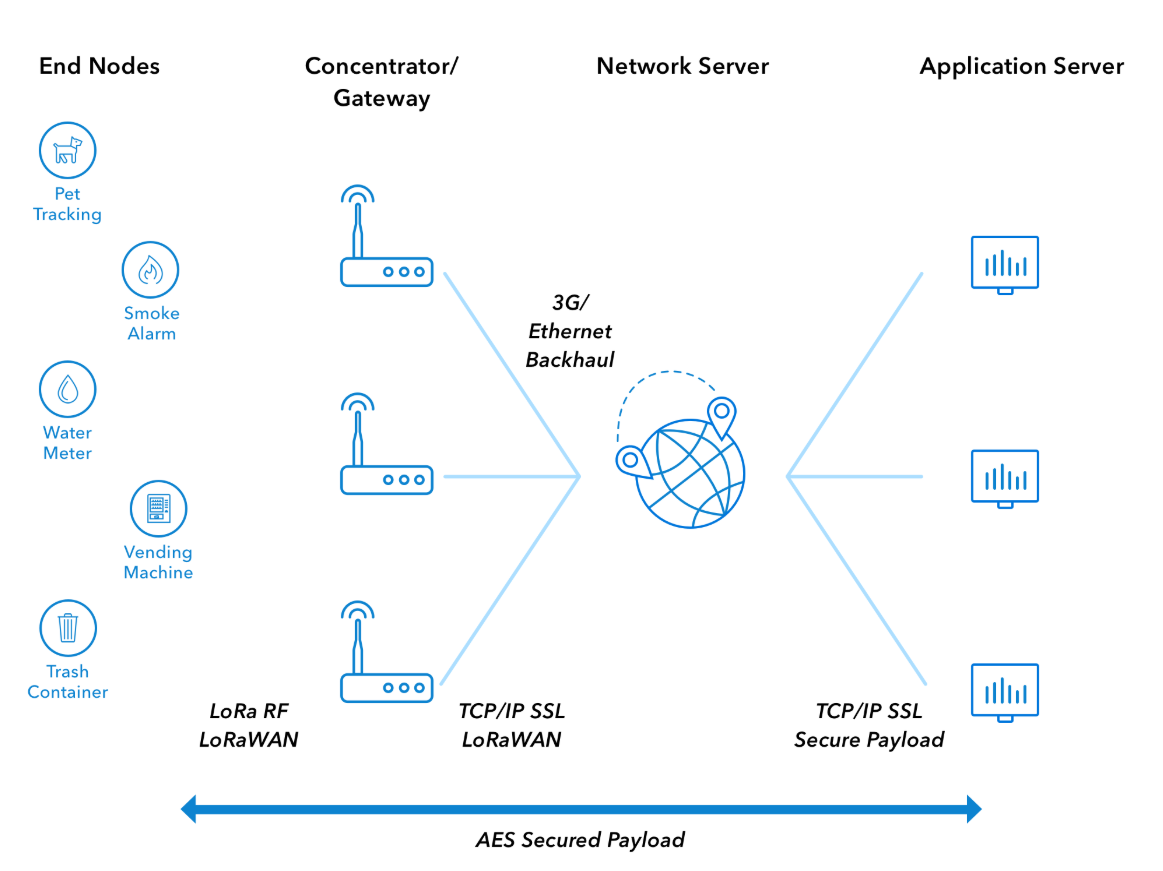
* **Eco-Friendly**: All the components used in our project do not harm our environment and thus, it is eco-friendly
* **Less Skills Required**: In order to operate our application, no hard core technical skills is required.
* **Automation and Digitalization**: Since this project is focused on automation, once the things are set up, no one is required i.e. the project works automatically.

# Other Requirements

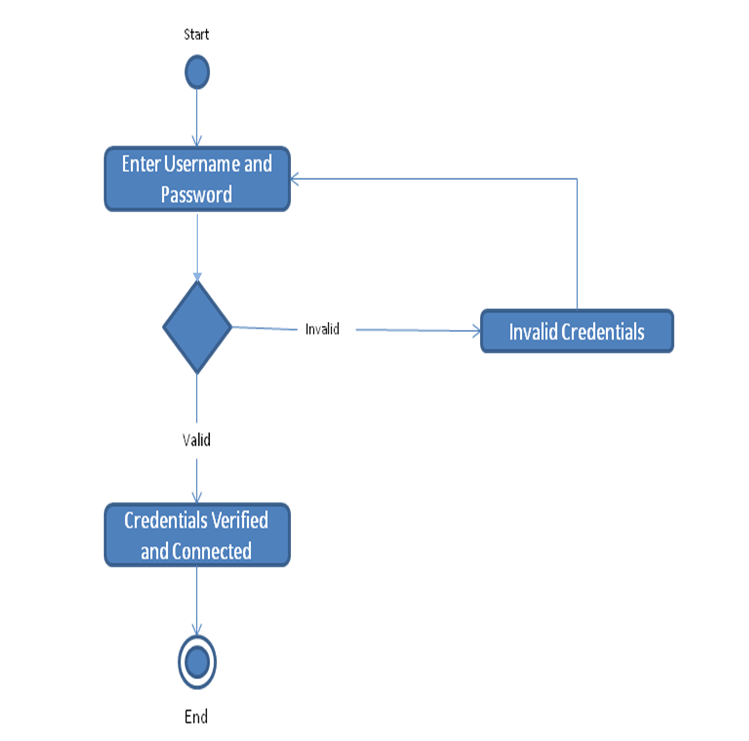
## 6.1 Appendix A: Glossary

* ABP (Activation By Personalization)
* IoT (Internet of Things)
* JSON (Java Script Object Notation)
* LoRa (Long Range Technology)
* OTAA (Over The Air Activation)
* TTN (The Things Network)

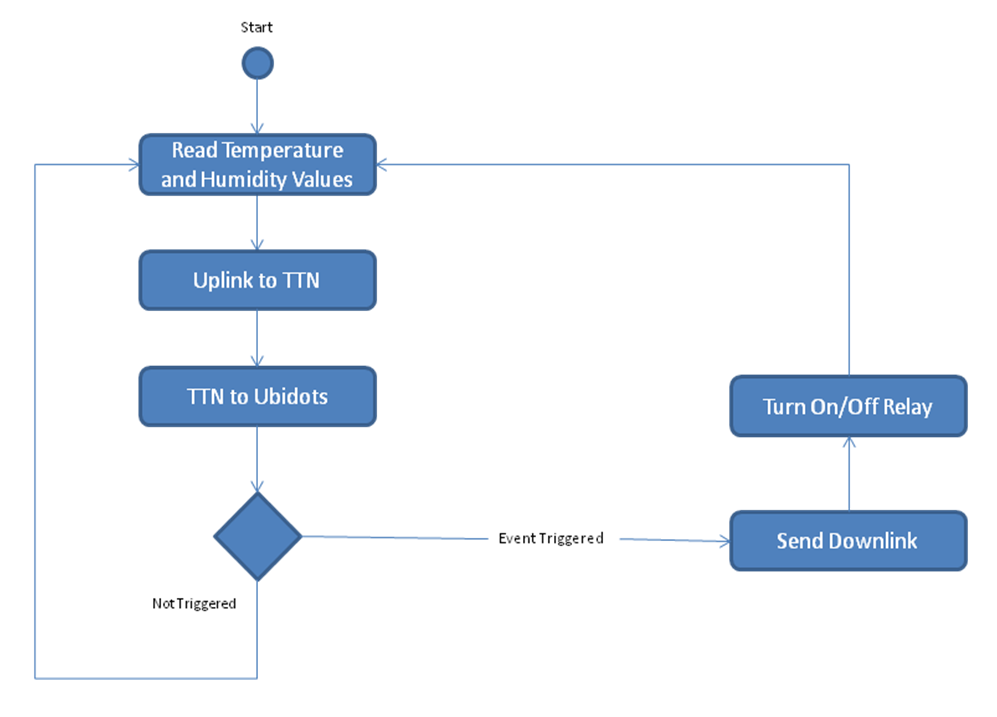
## 6.2 Appendix B: Analysis Models

****

(Architecture Model)



**Activity Diagram (Login)**



**Activity Diagram (System)**