Smartfarmer - Iot Enabled Smart Farming Application

A NAALAIYA THIRAN PROJECT REPORT

Submitted by

CH MOHAN (Team Leader)

D JAYESH (Team Member)

G MANIKANTA (Team Member)

A HAAREESH (Team Member)

G CHAKRAVARTHI (Team Member)

Team ID: PNT2022TMID14842

Smartfarmer - Iot Enabled Smart Farming Application

Project Report

Smartfarmer - Iot Enabled Smart Farming Application

ABSTRACT

From farm to fork, information and communication technology sector is being enhanced to facilitate the farmers, croppers and related users of intelligent services. Technological revolution integrates the development of smart devices and IoT services. To feed the ever growing global population, the agriculture industry needs to be extended.

Internet of Things opens the door wide for smart farming solution to increase the agricultural production. IoT technologies helps the farmers as a service by providing historical and real time data for predicting soil quality, weather conditions and crop's health. Smart farming provides the enhanced facility for process automation and evaluation and waste reduction. As a result, all these factors drastically increase the quality and quantity of the food products and decrease the production cost. This paper outlines the promising solutions applied in the sphere of agriculture.

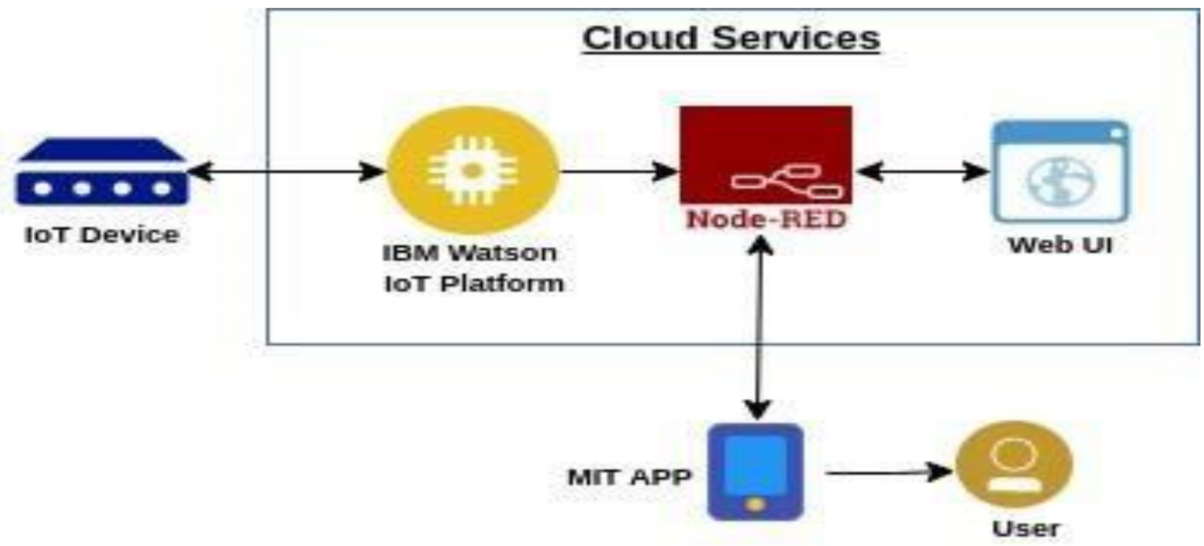
Keywords: Smart Farming, Internet of Things, Green House, IoT agriculture.

Smartfarmer - Iot Enabled Smart Farming Application

1. INTRODUCTION

1.1 PROJECT OVERVIEW

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.



1.2 PURPOSE

They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field.

Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.

In large farmland, Internet of Things equipped drone helps to receive the current state of crops and send the live pictures of farmland.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

The biggest challenges faced by IoT in the agricultural sector are lack of information, high adoption costs, and security concerns, etc. Most of the farmers are not aware of the implementation.

2.2 REFERENCE

[1] ISSN No:-2456-2165 Volume 4, Issue 2 Feb – 2019: "Solars' Energy: - A safe and reliable, eco-friendly and sustainable Clean Energy Option for Future India: - A Review."

[2] Universal Paper of advanced science and science and exploration technology. [2] GRD Journals- Global Research and Development Journal for Engineering | Volume 4 | Issue 3 | February (2019) ISSN: 2455-5703 “Design and Implementation of an Advanced Security System for Farm Protection from Wild Animals”.

[3] International Journal of Innovations in Engineering and Science, Impact Factor Value 4.046 e-ISSN: 2456-3463 Vol.4, No. 5, 2019 “Solar Powered Smart Fencing System for Agriculture Protection using GSM & Wireless Camera”.

[4] International Journal of Management, Technology and Engineering ISSN NO: 2249-7455

Volume 8, Issue VII, JULY/2018"Protecting Crops from Birds, Using Sound Technology In

Agriculture" [5] American Journal of Engineering Research (AJER) 2018 eISSN: 2320-0847 pISSN: 2320- 0936 Volume-7, Issue-7, pp-326-330 "Moisture Sensing Automatic Plant Watering System Using Arduino Uno".

2.3 PROBLEM STATEMENT DEFINITION

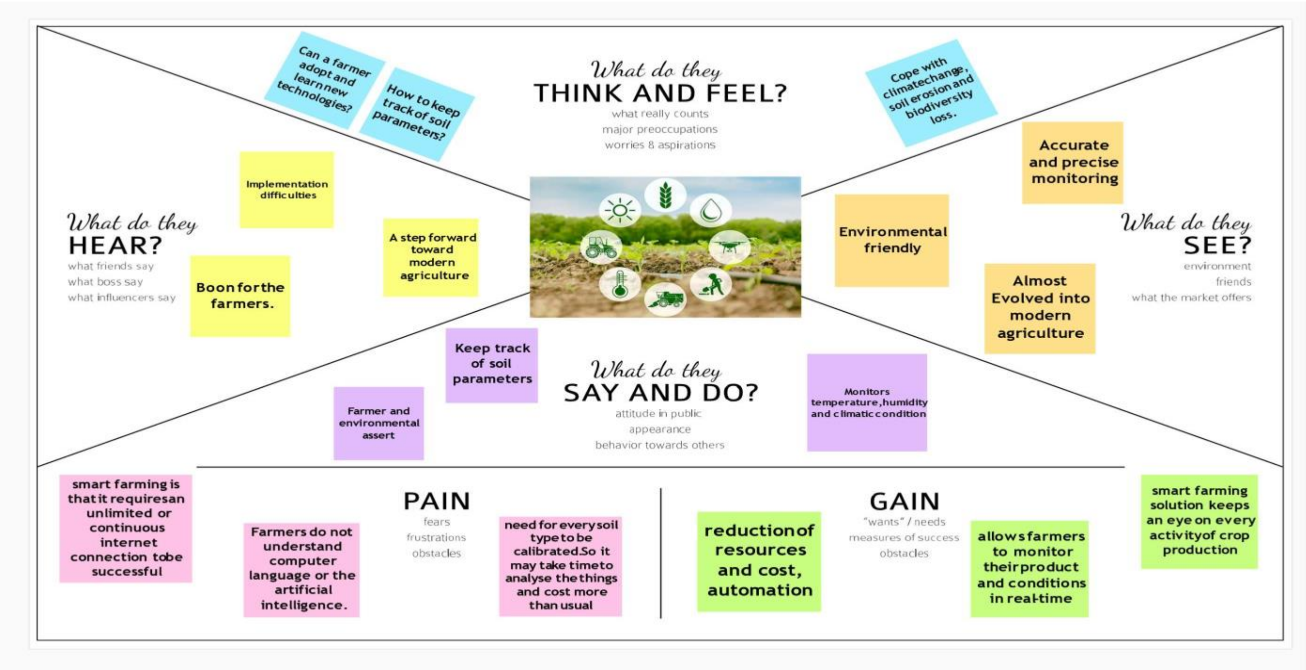
A strong customer problem statement should provide a detailed description of your customer's current situation. Consider how they feel, the financial and emotional impact of their current situation, and any other important details about their thoughts or feelings.

Creating a customer problem statement is easy with Miro. Using our collaborative online whiteboard, you can create an online problem statement that's easy to follow and shareable with your team. All you have to do is sign up for free, select this template, and follow your template.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem ant the person who is experiencing it The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



3.2 Ideation & Brainstorming

TEAM IDEAS:

CH MOHAN:

- Automate irrigation process using temperature of soil.
- Automate irrigation using measurement of moisture of soil.

A HAAREESH:

- We can use sensors on sensing.
- We can sense and program the moisture level.

D JAYESH:

- We can simplify the drip irrigation into time controlled irrigation.
- Automate irrigation using any Robots.

G CHAKRAVARTHI & G MANIKANTA:

- We can automate and design Audino for programming.
- We can make good design and programming of soil moisture and temperature.

Best Three Ideas:-

- Automate irrigation using measurement of moisture of soil.
- We can sense and program the moisture level.
- We can automate and design Audino for programming.

3.3 Proposed Solution

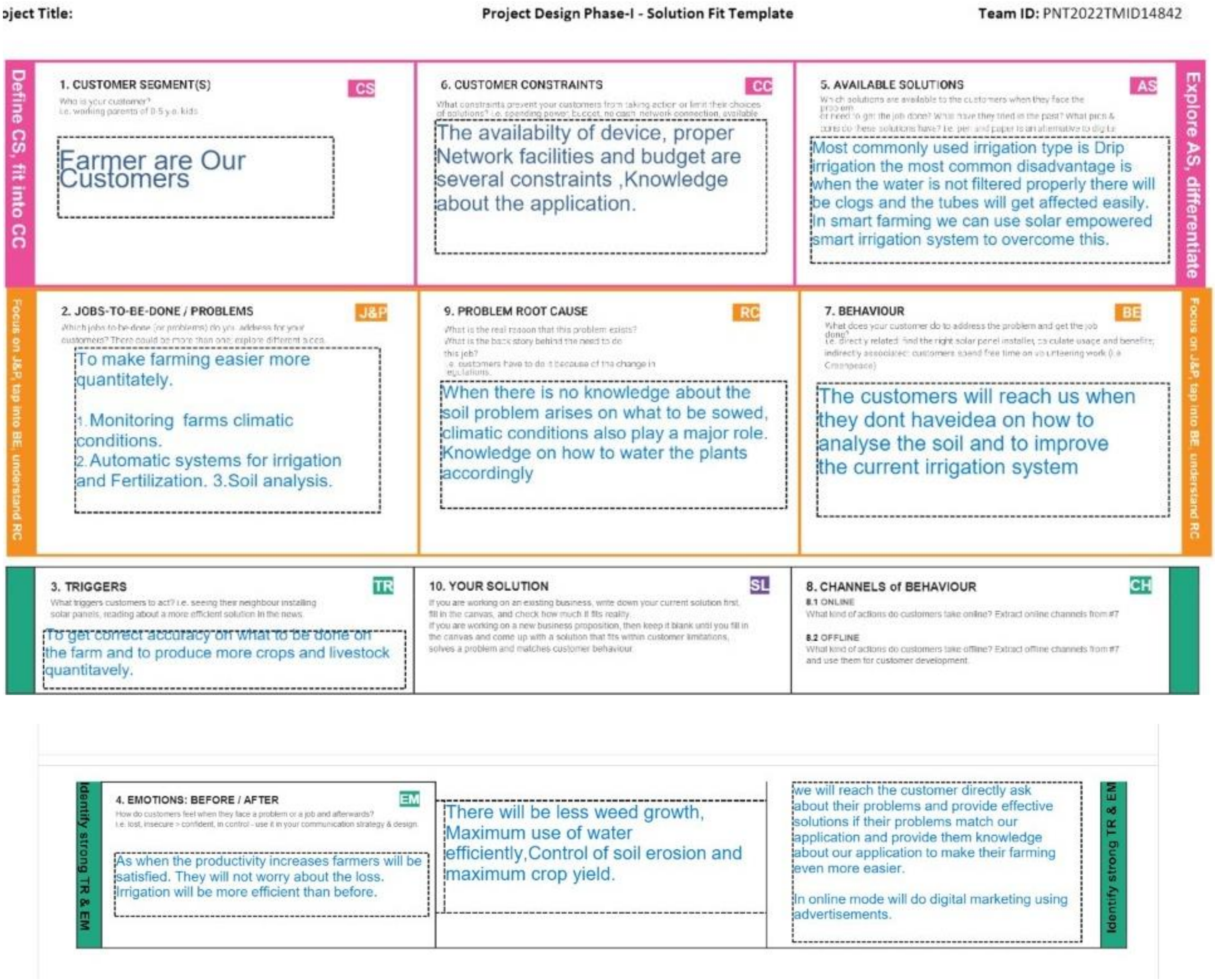
Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	To make farming easier by choosing several constraints in agriculture and to overcome those constraints, to increase production quality and quantity using IOT.
2.	Idea / Solution description	Using smart techniques like monitoring farms climate, smart irrigation and soil analysis.
3.	Novelty / Uniqueness	Solar power smart irrigation system which helps you to monitor temperature, moisture, humidity using smart sensors.
4.	Social Impact / Customer Satisfaction	It is better than the present modern irrigation system by using this method we can control soil erosion. There will be better production yield.
5.	Business Model (Revenue Model)	As the productivity increases customer satisfaction also increases and hence need for the application also increases, which increases the revenue of the business.
6.	Scalability of the Solution	It is definetly scalable we ca increase the constraints when the problem arises.

Smartfarmer - Iot Enabled Smart Farming Application

3.4 Problem Solution fit:



Smartfarmer - Iot Enabled Smart Farming Application

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Log in to system	Check Credentials Check Roles of Access.
FR-4	Manage Modules	Manage System Admins Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature details Humidity details
FR-6	Log out	Exit

4.2 NON-FUNCTIONAL REQUIREMENT

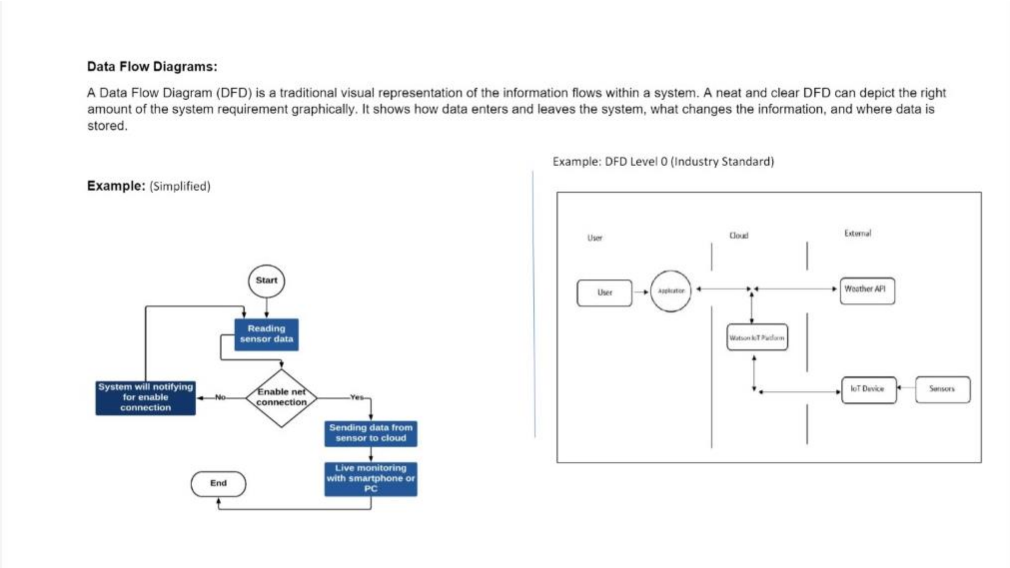
Following are the non-functional requirements of the proposed solution.

FR.No	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability includes easy learn ability, efficiency in use, remember ability, lack of errors in operation and subjective pleasure.
NFR-2	**Security**	Sensitive and private data must be protected from their production until the decision-making and storage stages.

NFR-3	**Reliability**	The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection schemes to avoid farm service outages.
NFR-4	**Performance**	the idea of implementing integrated sensors with sensing soil and environmental or ambient parameters in farming will be more efficient for Overall monitoring.
NFR-5	**Availability**	Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.
NFR-6	**Scalability**	Scalability is a major concern for IoT platforms. It has shown that different architectural choices of IoT platforms affect system scalability and that automatic real time decision-making is feasible in An environment composed of dozens of thousand.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

1. The different soil parameters temperature, soil moistures and then humidity are sense during different sensors and obtained value is stored in the IBMcloud.

2. Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weatherAPI.

3. NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed forth communication.

4. All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could plan through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch.

5.2 Solution & Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2 Guidelines:

1. The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM cloud.

2. Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.

3. NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.

4. All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could decide through an app, weather to water the crop or not depending upon the sensor values. By using the app, they can remotely operate the motor switch.

Smartfarmer - Iot Enabled Smart Farming Application

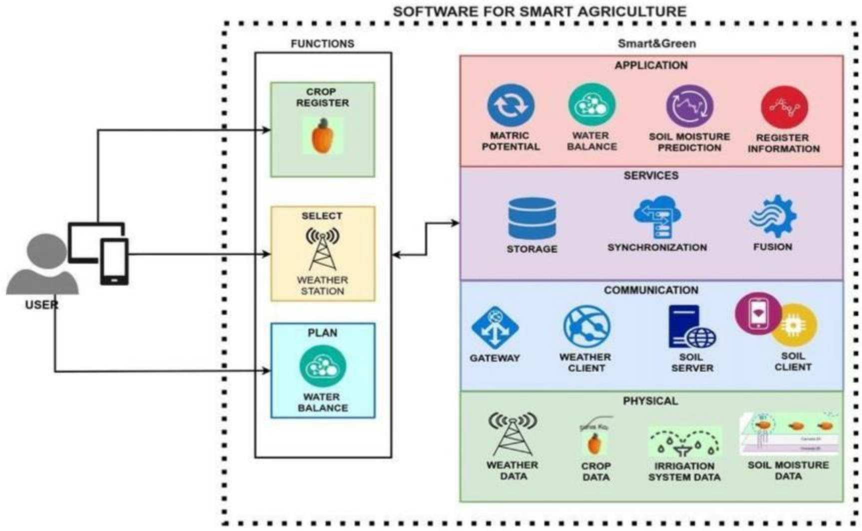


Table-1:

Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chabot etc.	MIT app
2.	Application Logic-1	Logic for a process in the application	Node red/IBM Watson/MIT app
3.	Application Logic-2	Logic for a process in the application	Node red/IBM Watson/MIT app
4.	Application Logic-3	Logic for a process in the application	Node red/IBM Watson/MIT app
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM cloud.
7.	Temperature sensor	Monitors the temperature of the crop	

8.	Humidity sensor	Monitors the humidity	
9.	Soil moisture sensor (Torsiometer's)	Monitors the soil temperature	
10.	Weather sensor	Monitors the weather	.
11.	Solar panel		.
12.	RTC module	Date and time configuration	
13.	Relay	To get the soil moisture data	

Table-2:

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	MIT app, Node-Red	Software
2.	Scalable Architecture	Drone technology, pesticide monitoring, Mineral identification in soil.	Hardware

5.3 USER STORIES

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile and web user)	User Registration	USN-1	As a user, I can register for the application by entering my username, password or by entering my phone number.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I can register for the application through Gmail		low	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering username and password or by entering my phone number.	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-4	As a user, I can check the soil temperature.	I can monitor the soil temperature	High	Sprint-2
		USN-5	As a user, I can check the humidity of the soil.	I can monitor the humidity	High	Sprint-2
		USN-6	As a user, I can check the temperature of the soil.	I can monitor the temperature	Medium	Sprint-2

Smartfarmer - Iot Enabled Smart Farming Application

6.PROJECT PLANNING AND SCHEDULING:6.1 Sprint planning and estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Member
Sprint-1	Registration (Farmer Mobile User)	UNS-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	CH MOHAN (Leader)
Sprint-1	Login	UNS-2	As a user, I will receive confirmation email once I have registered for the application	1	High	D JAYESH (Member 1)

Sprint-2	User Interface	UNS-3	As a user, I can register for the application through Facebook	3	Low	A HAAREESH (Member 2)
Sprint-1	Data Visualization	UNS-4	As a user, I can register for the application through GMAIL	2	Medium	G.MANIKANTA & G CHAKRAVARTHI (Member 3 & 4)
Sprint-3	Registration (Farmer -Web User)	USN - 1	As a user, I can log into the application by entering email and password	3	High	CH MOHAN (Leader)

Sprint - 2	Login	USN - 2	As a registered user, I need to easily login log into my registered account via the web page in minimum time .	3	High	D.JAYESH (Member 1)

Sprint - 4	Web UI	USN - 3	As a user, I need to have a friendly user interface to easily view and access the resources	3	Medium	A HAAREESH (Member 2)
Sprint - 1	Registration (Chemical Manufacturer - Web user)	USN - 1	As a new user, I want to first register using my organization email and create a password for the account.	2	High	G MANIKANTA & G CHAKRAVARTHI (Member 3 & 4)
Sprint - 4	Login	USN - 2	As a registered user, I need to easily log in using the registered account via the web page.	3	High	CH MOHAN (Leader)

Sprint - 3	Web UI	USN - 3	As a user, I need to have a user friendly interface to easily view and access the resources.	3	Medium	D JAYESH (Member 1)
Sprint - 1	Registration (Chemical Manufacturer - Mobile User)	USN - 1	As a user, I want to first register using my email and create a password for the account.	1	High	A HAAREESH (Member 2)
Sprint - 1	Login	USN - 2	As a registered user, I need to easily log in to the application.	2	Low	G MANIKANTA & G CHAKRAVARTHI (Member 3 & 4)

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	12	6 Days	24 Oct 2022	29 Oct 2022	20	14 Oct 2022
Sprint-2	6	6 Days	31 Oct 2022	05 Nov 2022	20	17 OCT 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	20	19 NOV 2022
Sprint-4	6	6 Days	14 Nov 2022	19 Nov 2022	20	21 NOV 2022

7. CODING & SOLUTIONING

7.1 Feature 1

import wiotp.sdk.device

import time import os

import datetime import random

myConfig = {
"identity": {
"orgId": "y5c2yt",

"typeId": "Devicel",

"deviceId": "12345"

},

"auth": {

"token": "12345678"

}

}

client = wiotp.sdk.device.DeviceClient (config=myConfig, logHandlers=None) client.connect () def myCommandCallback (cmd) : print ("Message received from IBM IoT Platform: %s" % cmd.data['command']) m=cmd.data['command'] if (m=="motoron"): print ("Motor is switched on") elif (m=="motoroff"): print

("Motor is switched OFF")
print (" ") while True:

soil=random.randint
(0,100) temp=random.r

andint (-20,

125) hum=random.r
andint (0, 100)
myData={'soilmoistu:
soil,'temperature':tem
p, 'humidity':hum
}
client.publishE
vent(eventId="status",
msgFormat="js on",

data=myData, qos=0 , onPublish=None)

print ("Published data Successfully: %s", myData) time.sleep (2)

client.commandCallback = myCommandCallback

client.disconnect ()

7.1 Feature 2

/*

Plant Watering System

The circuit:

- Water pump

Power supply: 4.5~12V DC Interface: Brown +; Blue - -

Temperature/moisture sensor Power supply: 3.3-5v

- Moisture sensor Power supply: 3.3-5v

*/

```
#include "DHT.h"

#define DHTPIN 2 // what digital pin we're connected to #define DHTTYPE
DHT22 // DHT 22 (AM2302), AM2321


DHT dht(DHTPIN, DHTTYPE);


const int SOIL_MOISTURE_SENSOR_PIN = A0; const int
WATER_PUMP_PIN = 4;


const int dry = 520; const int wet = 270; const

int moistureLevels = (dry - wet) / 3;


// TODO: Should we have a counter so if it waters for X times, then take a break?

// OPTIMIZE: how dry to start watering and for how long. const int
soilMoistureSartWatering = 400; const int soilMoistureStopWatering = 300;

// 60 seconds const long waterDuration =

1000L * 60L;

// 60 seconds const long sensorReadIntervals =

1000L * 60L;

// 2 hr
```

```
const long waterIntervals = 1000L * 60L * 60L * 2; long lastWaterTime =
waterIntervals - 1; boolean isWatering = false;


void setup()

{ Serial.begin(9600); pinMode(WATER_PUMP_PIN, OUTPUT);
waterPumpOff(); dht.begin();


}


void loop()

{ mainLoop ();

}


void mainLoop() { float temperature = getTemperature(); float

humidity = getHumidity(); long soilMoisture =

analogRead(SOIL_MOISTURE_SENSOR_PIN);

Serial.println("Soil Moisture: " + readableSoilMoisture(soilMoisture) + ", " +
soilMoisture);

Serial.println("Temperature: " + String(temperature) + "
*F");Serial.println("Humidity: " + String(humidity) + " %");


if (millis() - lastWaterTime > waterIntervals)
```

{waterPlants(soilMoisture); lastWaterTime = millis();

}

delay(sensorReadIntervals);

}

void waterPlants(int soilMoisture) {

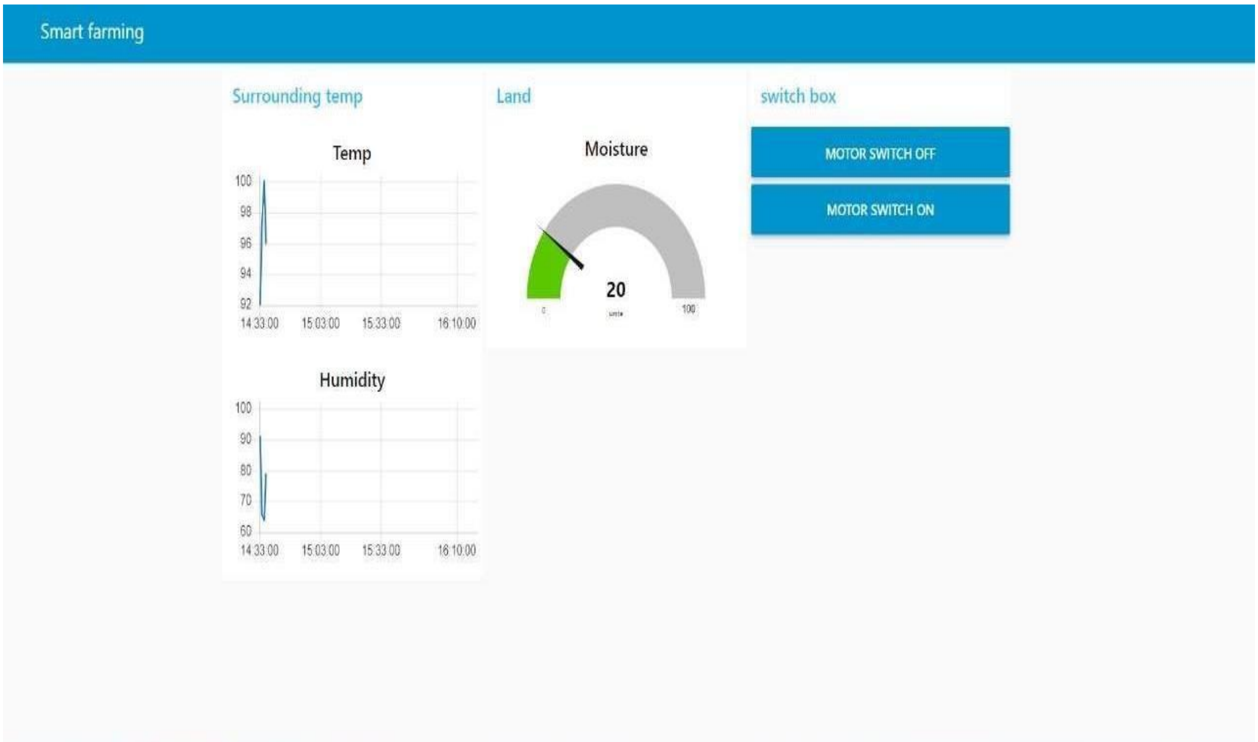
// Should this take a moving avg of the soilMoisture?

// Can get outliers on the right after watering. if (soilMoisture > soilMoistureSartWatering)

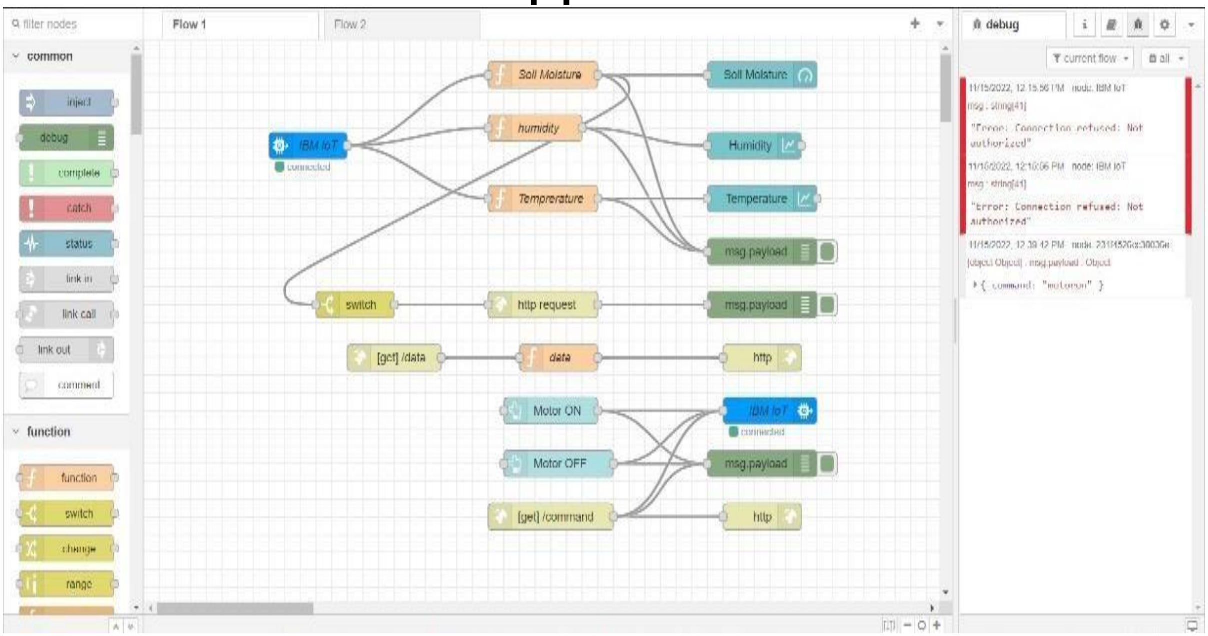
{ isWatering = true

8. TESTING

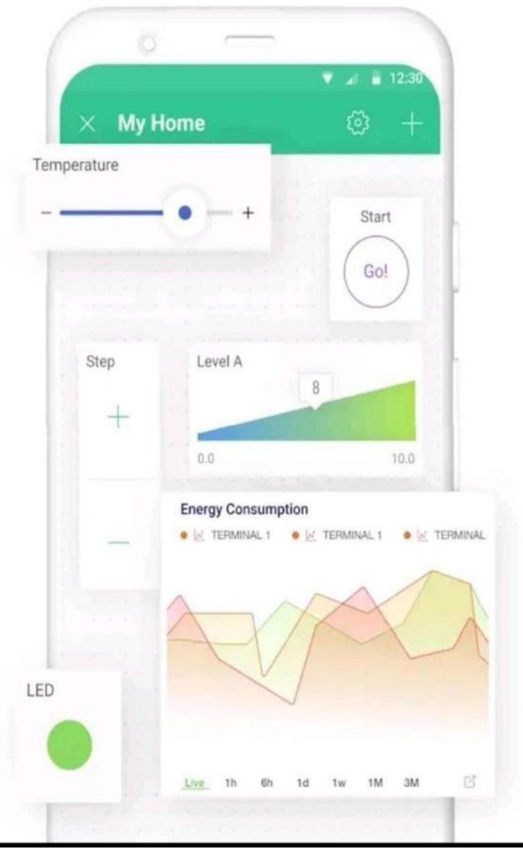
8.1 Test Cases



Smartfarmer - Iot Enabled Smart Farming Application

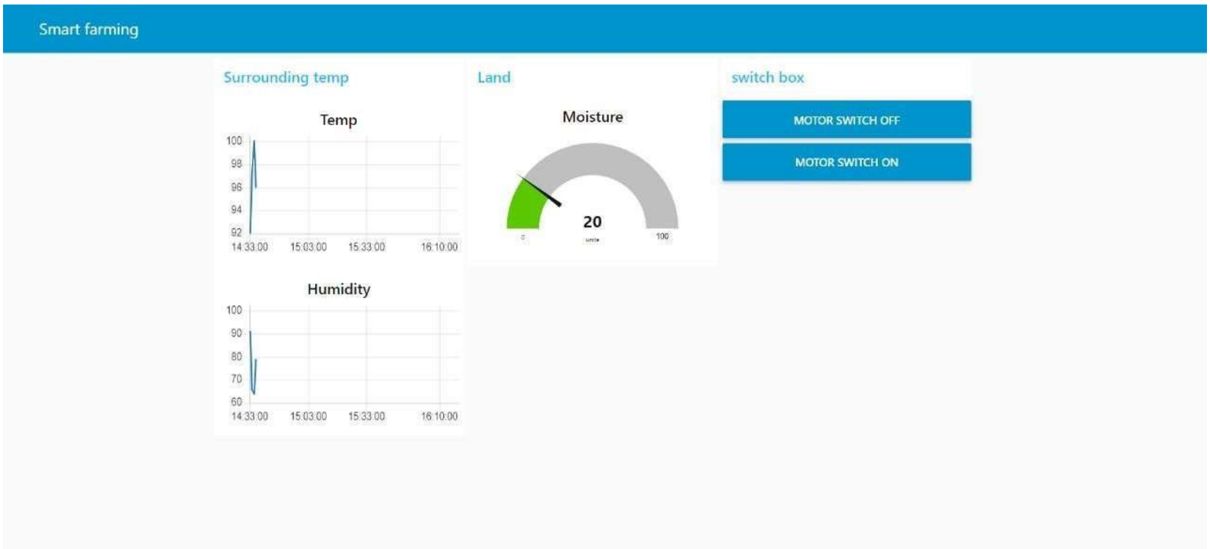


8.2 User Acceptance Testing



9. RESULTS

9.1 PERFORMANCE METRICS



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- A remote control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labor-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed.

- For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motordriven hardware become the next logical step.

- Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system.

- Submersible weight sensors or ultrasonic sensors can screen the degree of tanks, lakes, wells and different kinds of fluid stockpiling like fuel and compost. The product figures volume dependent on the tank or lake geometry after some time. It conveys alarms dependent on various conditions.

DISADVANTAGES:

- The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

- The smart farming based equipment require farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.

11. CONCLUSION

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of Farming irrigation, agriculture suffers. Climate factors such as humidity, temperature, and moisture can be adjusted dependent on the local environmental variables. This technology also detects animal invasions, which are a major cause of crop loss. This technology aids in the scheduling of irrigation based on present data from the field and records from a climate source. It helps in deciding the farmer to whether to do Smart farming irrigation or not to do. Continuous internet connectivity is required for continuous monitoring of data from sensors. This also can be overcome by using GSM unit as an alternative of mobile app. By GSM, SMS can be sent to farmer's phone.

12. FUTURE SCOPE

- In future due to more demand of good and more farming in less time, for betterment of the crops and reducing the usage of extravagant resources like electricity and water IOT can be implemented in most of the places.

- In the current project we have implemented the project that can protect and maintain the the crop. In this project the farmer monitor and control the field remotely. In future we can add or update few more things to this project.

- We can create few more models of the same project, so that the farmer can have information of an entire.

- We can update this project by using solar power mechanism. So that the power supply from electric poles can be replaced with solar panels. It reduces the power line cost. It will be a one-time investment. We can add solar fencing technology to this project.

- We can use GSM technology to this project so that the farmers can get the information directly to his home through SMS. This helps the farmer to get information if there is an internet issues.

- We can add camera feature so that the farmer can monitor his field in real time. This helps in avoiding thefts.

13. APPENDIX

SOURCE CODE

import wiotp.sdk.device

import time import os

import datetime import random

myConfig = { "identity":
{
"orgId": "y5c2yt",

"typeId": "Devicel",

"deviceId": "12345"

},

"auth": {

"token": "12345678"

}

}

client = wiotp.sdk.device.DeviceClient (config=myConfig, logHandlers=None) client.connect () def myCommandCallback (cmd) : print ("Message received from IBM IoT Platform: %s" % cmd.data['command']) m=cmd.data['command'] if (m=="motoron"): print ("Motor is switched on") elif (m=="motoroff"): print

("Motor is switched OFF")
print (" ") while True:
soil=random.ra ndint
(0,100) temp=random.r

andint (-20,

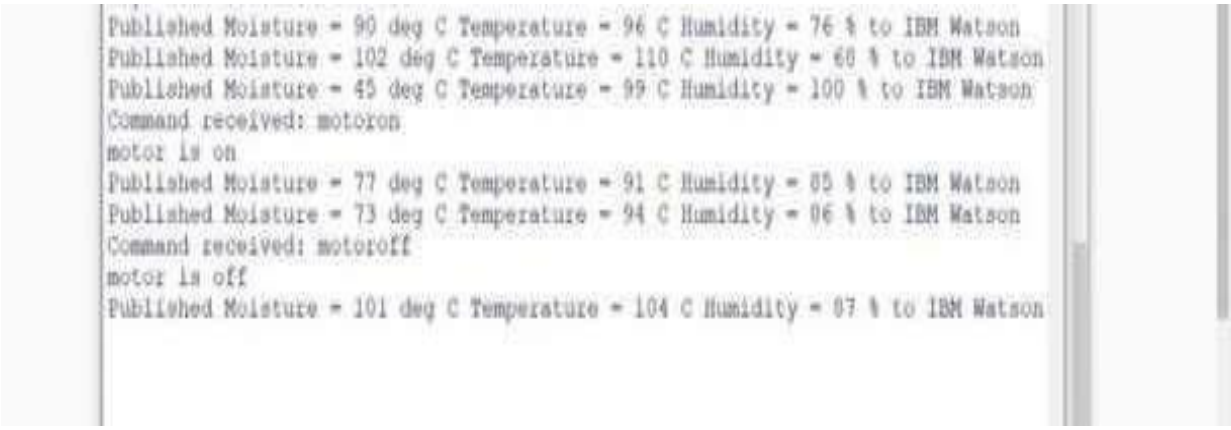
125) hum=random.r
andint (0, 100)
myData={'soil
moisture': soil,
'temperature':temp,
'humidity':hum
}
client.publishEvent
(eventId="status",
msgFormat="js on",

data=myData, qos=0 , onPublish=None)

print ("Published data Successfully: %s", myData) time.sleep (2)

client.commandCallback = myCommandCallback
client.disconnect ()

OUTPUT:



```
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 60 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 96 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```

GitHub link: https://github.com/IBM-EPBL/IBM-Project-34401-1660235409

Smartfarmer - Iot Enabled Smart Farming Application

Project Demo link: https://github.com/IBM-EPBL/IBM-Project-34401-1660235409/tree/main/Demo%20file