PROJECT

SmartFarmer - IoT Enabled Smart Farming Application

DONE BY : TEAM ID: PNT2022TMID27964

VikkramSrinivasan(311519106110)

Vikram.S(311519106111)

Sashanth.O(311519106083)

Ravindran K(311519106075)

Vignesh M(311519106107)

Project Report

1. Introduction

Problem Statement - SmartFarmer : IoT Enabled Smart Farming Application

What is a problem statement - Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

1.1 Project Overview

Agriculture, the major sector which defines the growth of the nation and any community that exists on this earth. It requires very intensive care, discipline and patience to reap the yield. In this evolving modern world the number of people who adopt agriculture are only a handful, in the next few years there will be very minimal people who do farming.

This project will help the farmers to grow crops in a whole new way so that they can reap good yield and healthy foods. This system does the following:

1. Autonomous crop monitoring.
2. Irrigation control.
3. Environment sensing.

1.2 Purpose

The purpose of this project is to help farmers to grow crops better and have yield, it emanates the need to manually look for the soil condition and decide whether to water the plantor not, this process becomes very difficult when plants are maintained over a very large area. This project helps the farmers to find a better solution and make them aware of their surroundings.

1.3 Purpose

The purpose of this project is to help farmers to grow crops better and have yield, it emanates the need to manually look for the soil condition and decide whether to water the plantor not, this process becomes very difficult when plants are maintained over a very large area. This project helps the farmers to find a better solution and make them aware of their surroundings.

2. Literature Survey

2.1 Existing Problem

Farmers must meet the challenging needs of our planet and the experiences and the expectations of the regulations, consumers and food produced. There are increasing pressure from climate change, soil erosion and biodiversity loss and from consumers changing tastes in food and concerns about how it is produced and the natural world that farming works with plants,pests and diseases.

Problems that farmers face:

1. Cope up with climate change, soil erosion and biodiversity.
2. Satisfy the customer needs.
3. Meet rising demands for more food of higher quality.
4. Invest in farm productivity.
5. Adopt and learn new technologies.

2.2 References

LITERATURE SURVEY : SMART FARMER - IOT BASED SMART FARMING

TITLE	**AUTHORS**	**ABOUT**	**LIMITATIONS**
1. IoT-Equipped and AI-Enabled Next Generation Smart Agriculture: A Critical Review, Current Challenges and Future Trends [2nd march 2022]	• SAMEER QAZI 1 (Senior Member, IEEE) • BILAL A. KHAWAJA (Senior Member, IEEE) • QAZI UMAR FAROOQ	The author speaks about the evolvement of wireless sensor networks over the years and various sensors used in the field to collect data and by using AI the analysis is done which will help farmers to make better decisions and will understand in depth analysis of the crops.	• Possible hacking of smart machinery like smart tractors or UAV's. • Not every farmers could able to implement this smart farming as it require a lot of investment • Paradigm shift from cloud based to edge AI applications for smart agriculture.

2. A Virtual Soil Moisture Sensor for Smart Farming Using Deep Learning [2022]	• Gabriele Patrizi , (Member, IEEE) • Alessandro Bartolini (Graduate Student Member, IEEE) • Lorenzo Ciani , (Senior Member, IEEE) • Vincenzo Gallo , (Graduate Student Member, IEEE) • Paolo Sommella , (Member, IEEE) • Marco Carratù , (Member, IEEE)	A virtual soft sensor is created by using artificial neural networks which can be used to track various necessary values for agriculture such as moisture, temperature, humidity, atmospheric pressure, Co2 concentration etc. by combining various normal sensors such ah temperature and humidity sensors and using DL (deep learning) algorithms to identify plant diseases.	• Long training time for installation of the node. • Complexity in designing the node.

TITLE	**AUTHORS**	**ABOUT**	**LIMITATIONS**
3. Comparative analysis of wireless technologies for internet- of-things based smart farm [March 2017]	- Asad Abbas - Muhammad Taha Jilani1, - Muhammad Khalid Khan	In this paper an overview of different wireless technologies is presented to provide connectivity to the physical things, particularly for a poultry farm. In a farm, connecting several sensors and automate various tasks and some data analytics, can be used to improve farming. It can also be used to easily monitor the environment of the poultry farm, thus providing better monitoring and control. The analysis of different technologies is carried out and the ZigBee and LTE technologies are found to be the most feasible and economical solutions for a smart poultry farm.	- The paper analysis various different technologies that can be used but does not give a clearer info on how various technologies can be implied in real life scenario. - It has some critical issue and one of the most important challenges affecting sensor lifetime, especially in those applications where power resources are limited. - Also most of these solutions which are implied to improve zigbee are simulation-based and have not been validated practically. - SMS alert system is mentioned for in case of risks while this system lacks in response time and efficiency while costing high.

TITLE	**AUTHORS**	**ABOUT**	**LIMITATIONS**
4. IoT based Intelligent irrigation support system for smart farming applications [2019]	• Neha Kailash Nawandar • Vishal Satpute	This paper presents an irrigation management system with sensor data fetching and compression, compressed data transfer, data processing, decision making and action invoke capabilities. A network of sensors implanted for the plants and three basic blocks form the whole system, compress the sensed data, send it to the FTP server which reconstructs it back into original form. A 2-layer Neural Network that utilizes the 4 inputs is used here for decision making. The proposed system monitors the test object 24×7 and it is capable to monitor a farm for its water and other requirements. It has compression and decision making capabilities which	• The output is notified to user via email which might cause unwanted power and storage usage which in turn might reduce battery life. • The use of neural networks for decision making increases the system complexity and the data flow process might get difficult to understand.

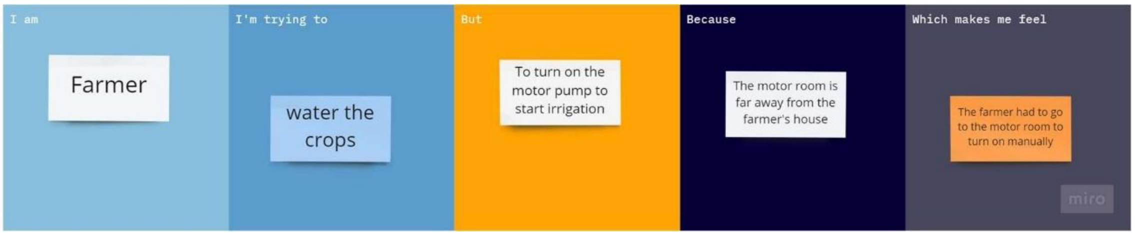
		makes it useful for home gardens, greenhouses, etc.	

TITLE	**AUTHORS**	**ABOUT**	**LIMITATIONS**
5. A Novel Framework for Smart Crop Monitoring Using Internet of Things (IOT) [2018]	• Ghanshala K.K. , • Chauhan.R , • Joshi R. C.	Unlike the traditional agricultural trends followed in India, this system mainly focussed on the soil nutrients and its adequate utilization. Earlier most of the crop monitoring techniques were based on temperature and humidity and limited to user end only but proposed system is based on soil nutrients and based on IOT. Thus increase the reach and use of edge computing and cloud computing make it globally accessible for data analytics. Cloud data analysis can be easily done for generating the soil nutrient requirement through machine learning techniques. Thus this system ensures that access is available to the farmers and ultimately fertilizers can be efficiently utilized for high yield .	• Power consumption is neglected here which is one of the prime factors to be taken care of while implementing the system . • The Zigbee wifi module used (ESP8266) is a 3.3V device so it might not be much compatible with all the peripherals used and the wifi code takes more CPU power . • The error percentage in calculating accurate Soil nutrients might be higher when compared to Temperature and humidity and also different crops require different ideal nutrients

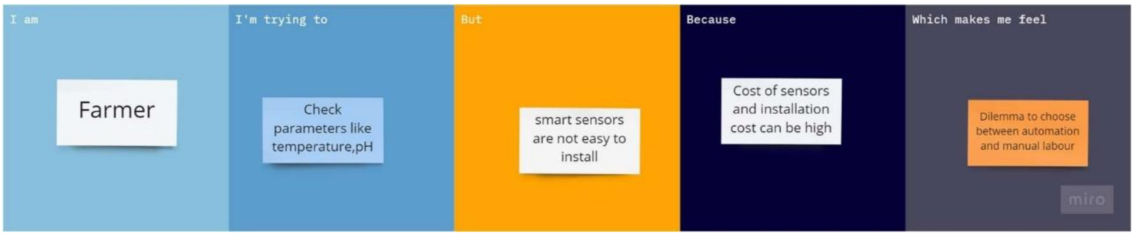
2.3 Problem statement definition

Customer Problem Statement:

A young farmer who want to start his farming business by growing various crops on his field.Since he is young he may not know about the traditional methods of farming. But the farmer is literate and knows how to operate applications on certain degree. Therefore by building a mobile application which will show data about various factors such as temperature, humidity, pH value of the soil etc. Also the farmer can operate motor pump from remote area to turn on using the same mobile application and can also set a schedule based on when to irrigate the field. This will ease the farmer's work and it will reduce the manpower required for farming. Thereby the farmer can reduce the cost of labour be more efficient.



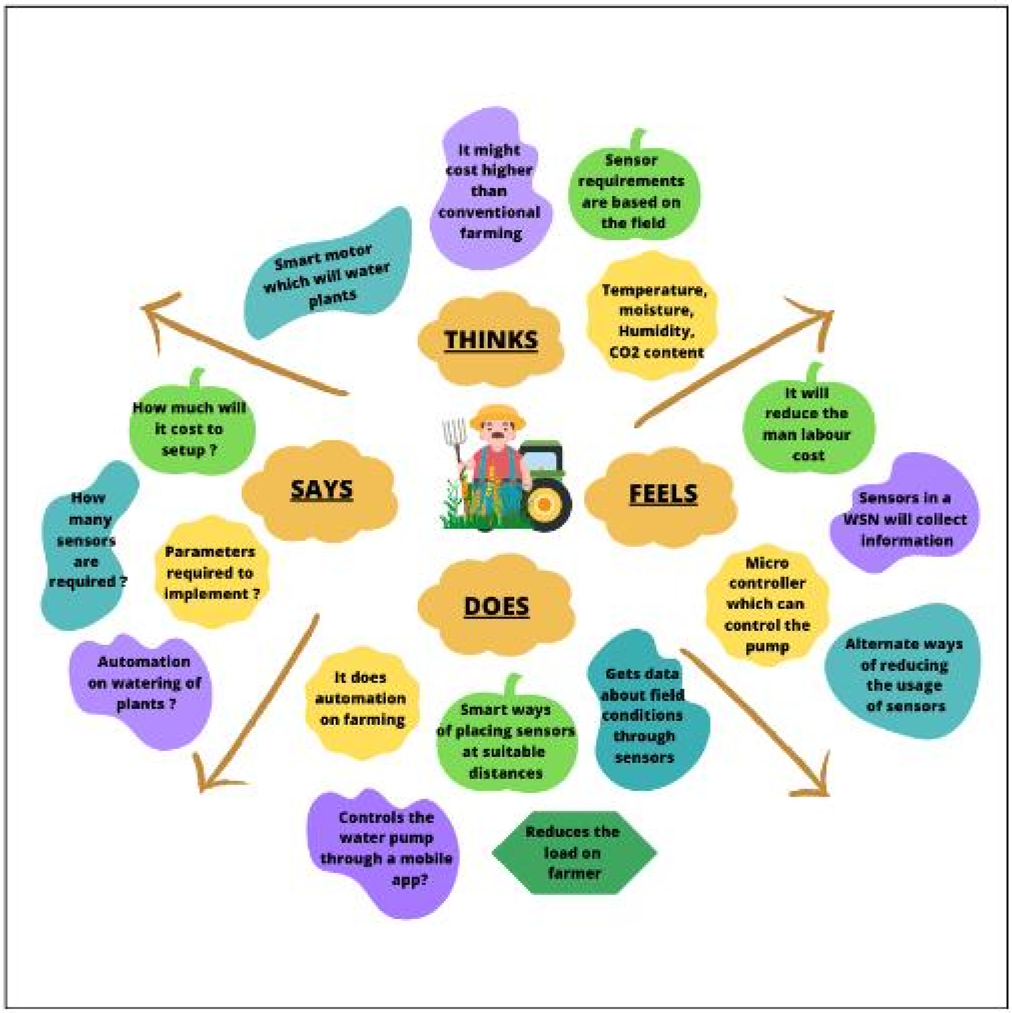
I am	I'm trying to	But	Because	Which makes me feel
Farmer	water the crops	To turn on the motor pump to start irrigation	The motor room is far away from the farmer's house	The farmer had to go to the motor room to turn on manually



I am	I'm trying to	But	Because	Which makes me feel
Farmer	Check parameters like temperature,pH	smart sensors are not easy to install	Cost of sensors and installation cost can be high	Dilemma to choose between automation and manual labour

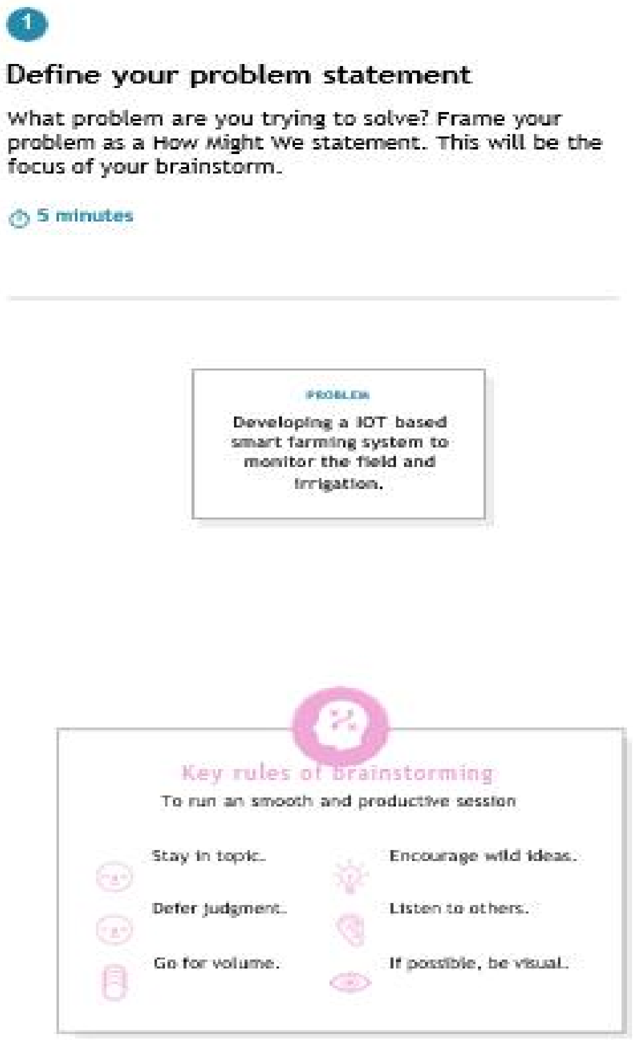
3.1Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. **Empathy Map:**


THINKS
It might cost higher than conventional farming
Sensor requirements are based on the field
Temperature, moisture, Humidity, CO2 content
Smart motor which will water plants
SAYS
How much will it cost to setup ?
How many sensors are required ?
Parameters required to implement ?
Automation on watering of plants ?
FEELS
It will reduce the man labour cost
Sensors in a WSN will collect information
Micro controller which can control the pump
Alternate ways of reducing the usage of sensors
DOES
It does automation on farming
Smart ways of placing sensors at suitable distances
Gets data about field conditions through sensors
Controls the water pump through a mobile app?
Reduces the load on farmer

3.2Ideation & Brainstorming:

Step-1: Team Gathering, Collaboration and Select the Problem Statement



1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

Developing a IOT based smart farming system to monitor the field and irrigation.

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

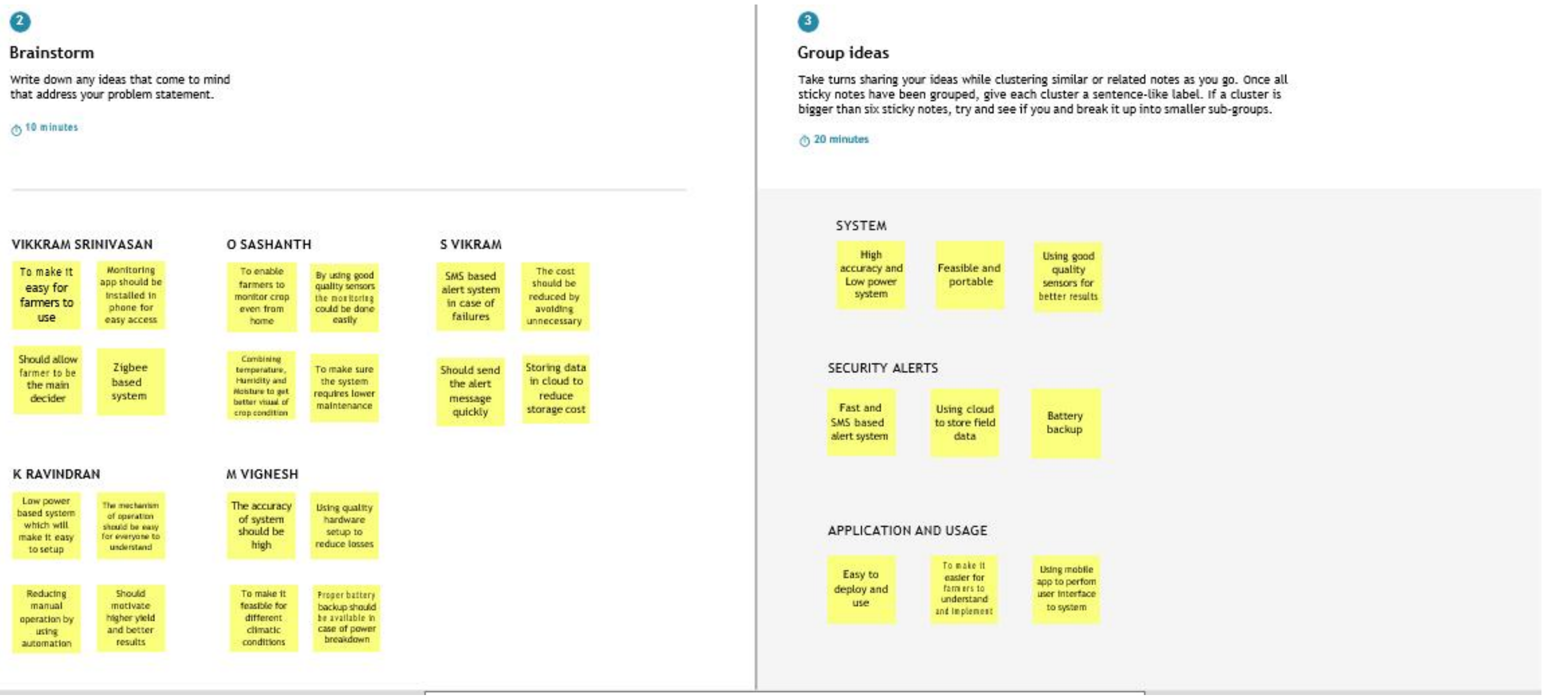
Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

VIKKRAM SRINIVASAN

To make it easy for farmers to use

Monitoring app should be installed in phone for easy access

Should allow farmer to be the main decider

Zigbee based system

O SASHANTH

To enable farmers to monitor crop even from home

By using good quality sensors the monitoring could be done easily

Combining temperature, Humidity and Moisture to get better visual of crop condition

To make sure the system requires lower maintenance

S VIKRAM

SMS based alert system in case of failures

The cost should be reduced by avoiding unnecessary

Should send the alert message quickly

Storing data in cloud to reduce storage cost

K RAVINDRAN

Low power based system which will make it easy to setup

The mechanism of operation should be easy for everyone to understand

Reducing manual operation by using automation

Should motivate higher yield and better results

M VIGNESH

The accuracy of system should be high

Using quality hardware setup to reduce losses

To make it feasible for different climatic conditions

Proper battery backup should be available in case of power breakdown

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

SYSTEM

High accuracy and Low power system

Feasible and portable

Using good quality sensors for better results

SECURITY ALERTS

Fast and SMS based alert system

Using cloud to store field data

Battery backup

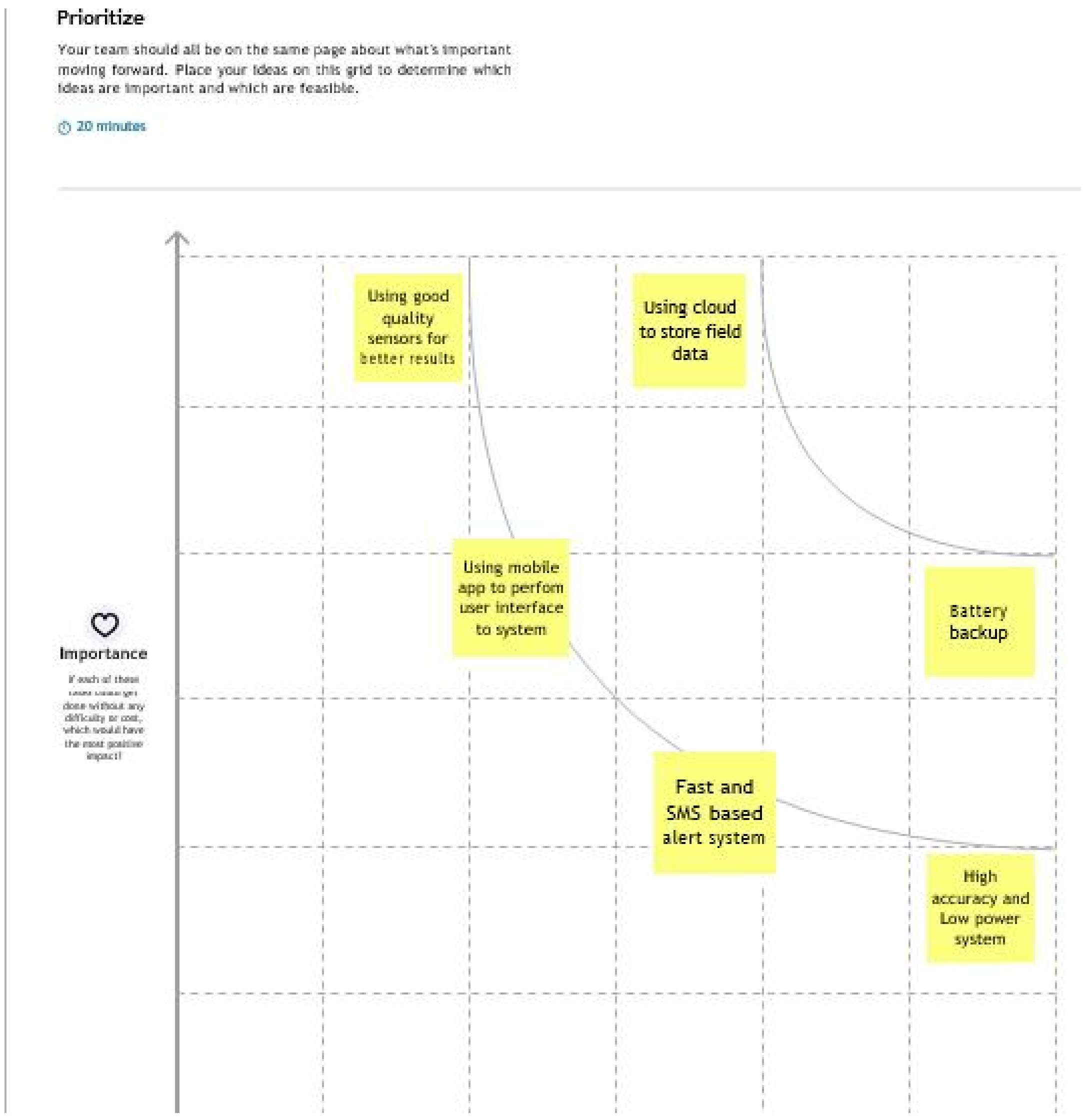
APPLICATION AND USAGE

Easy to deploy and use

To make it easier for farmers to understand and implement

Using mobile app to perfom user interface to system

Step-3: Idea Prioritization



Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact!

Using good quality sensors for better results

Using cloud to store field data

Using mobile app to perfom user interface to system

Battery backup

Fast and SMS based alert system

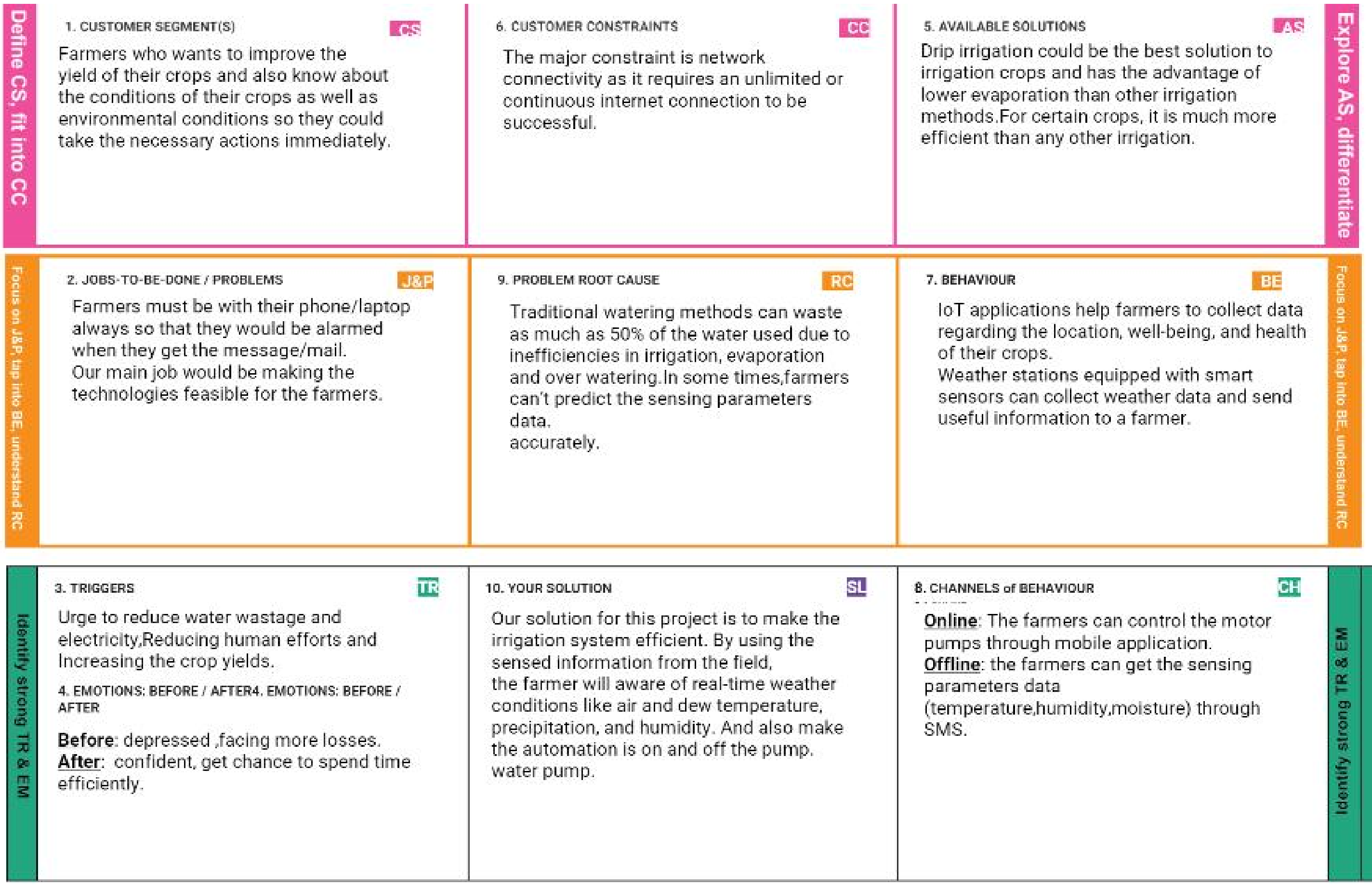
High accuracy and Low power system

3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	IoT-based agriculture systems help the farmer in monitoring different parameters of his field like soil moisture, temperature, and humidity using some sensors.
2.	Idea / Solution description	Our system comprises the following elements to come up with a solution: 1. Pest control. 2. Timely irrigation. 3. Constant nutrient monitoring. 4. Estimated time for cultivation. 5. Additional nutrient supplements. 6. Estimated crop yield. 7. Environment monitoring.
3.	Novelty / Uniqueness	Our system could function in both solar and battery mode. The inbuilt battery delivers power during the necessary times. It also delivers remote sensing facilities.
4.	Social Impact / Customer Satisfaction	Upon implementing customers feel : 1. Seeing nearby adopting better agriculture practice. 2. Better income rates. 3. Better yield. 4. Feeling motivated. 5. Stable income. 6. Happy to work. 7. Feeling comfortable with the practices

5.	Business Model (Revenue Model)	Our system comprises of hardware and software part: Hardware: 1. Controller(Brain) - 8000 2. Solenoid valves - 5000/piece 3. Pipe materials - needed to be provided by the land owner(May vary from place to place). 4. Cloud storage of data - 10000/Month(For n customers can be scaled up on demand) Roughly sums around - 25000
		Additionally we can generate income by increasing the number of controllers since it is limited to a specific area. Additional income - Ads posted on our mobile and web application platform
6.	Scalability of the Solution	This system of ours is like a lego which can be stacked and scaled up for a larger growth area.

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Farmers who wants to improve the yield of their crops and also know about the conditions of their crops as well as environmental conditions so they could take the necessary actions immediately.

6. CUSTOMER CONSTRAINTS CC

The major constraint is network connectivity as it requires an unlimited or continuous internet connection to be successful.

5. AVAILABLE SOLUTIONS AS

Drip irrigation could be the best solution to irrigation crops and has the advantage of lower evaporation than other irrigation methods.For certain crops, it is much more efficient than any other irrigation.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Farmers must be with their phone/laptop always so that they would be alarmed when they get the message/mail.
Our main job would be making the technologies feasible for the farmers.

9. PROBLEM ROOT CAUSE RC

Traditional watering methods can waste as much as 50% of the water used due to inefficiencies in irrigation, evaporation and over watering.In some times,farmers can't predict the sensing parameters data.
accurately.

7. BEHAVIOUR BE

IoT applications help farmers to collect data regarding the location, well-being, and health of their crops.
Weather stations equipped with smart sensors can collect weather data and send useful information to a farmer.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Urge to reduce water wastage and electricity,Reducing human efforts and Increasing the crop yields.

4. EMOTIONS: BEFORE / AFTER4. EMOTIONS: BEFORE / AFTER

Before: depressed ,facing more losses.
After: confident, get chance to spend time efficiently.

10. YOUR SOLUTION SL

Our solution for this project is to make the irrigation system efficient. By using the sensed information from the field,
the farmer will aware of real-time weather conditions like air and dew temperature, precipitation, and humidity. And also make the automation is on and off the pump.
water pump.

8. CHANNELS of BEHAVIOUR CH

Online: The farmers can control the motor pumps through mobile application.
Offline: the farmers can get the sensing parameters data (temperature,humidity,moisture) through SMS.

Identify strong TR & EM

REQUIREMENT ANALYSIS:

1.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Account Creation	Users should create an account in the web application to order the product.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Order Confirmation	Order is confirmed by a prompt message and redirected to the payment page.

FR-4	Payment	Payment can be made either through any online platform or can be paid at the time of delivery.
FR-5	Field Visit	A small field visit is made by the expert team to plan the product installation.
FR-6	Adding user data	A few data are added in the device before the installation to send sms and cloud storage.

1.2 Non-functional Requirements:

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

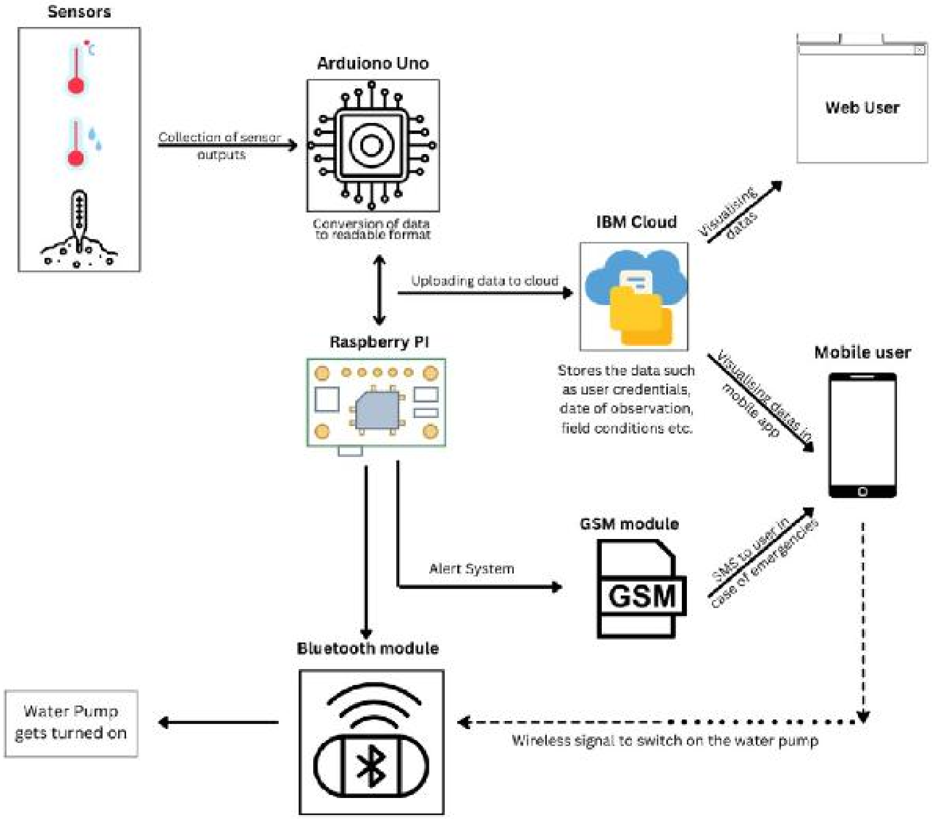
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Since the product is completely autonomous there is no need for extensive care.
NFR-2	**Security**	Since the has its own login credentials for both user and cloud login it makes it secure against data theft.

NFR-3	**Reliability**	The physical structure of the product is enclosed using waterproof and non corrosive materials it makes it reliable to use.
NFR-4	**Performance**	The performance of the device increases gradually based on the sensor data collected from the filed.
NFR-5	**Availability**	This system requires an active internet connection for better functioning and offsite control. But it can also work offline but users cannot monitor remotely.
NFR-6	**Scalability**	It can be scaled easily according to the area.

4.Project Design:

4.1Data 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.


Sensors
°C
Collection of sensor outputs
Arduiono Uno
Conversion of data to readable format
Web User
Visualising datas
IBM Cloud
Uploading data to cloud
Raspberry PI
Stores the data such as user credentials, date of observation, field conditions etc.
Visualising datas in mobile app
Mobile user
GSM module
Alert System
GSM
SMS to user in case of emergencies
Bluetooth module
Water Pump gets turned on
Wireless signal to switch on the water pump

4.2Solution & Technical Architecture:

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

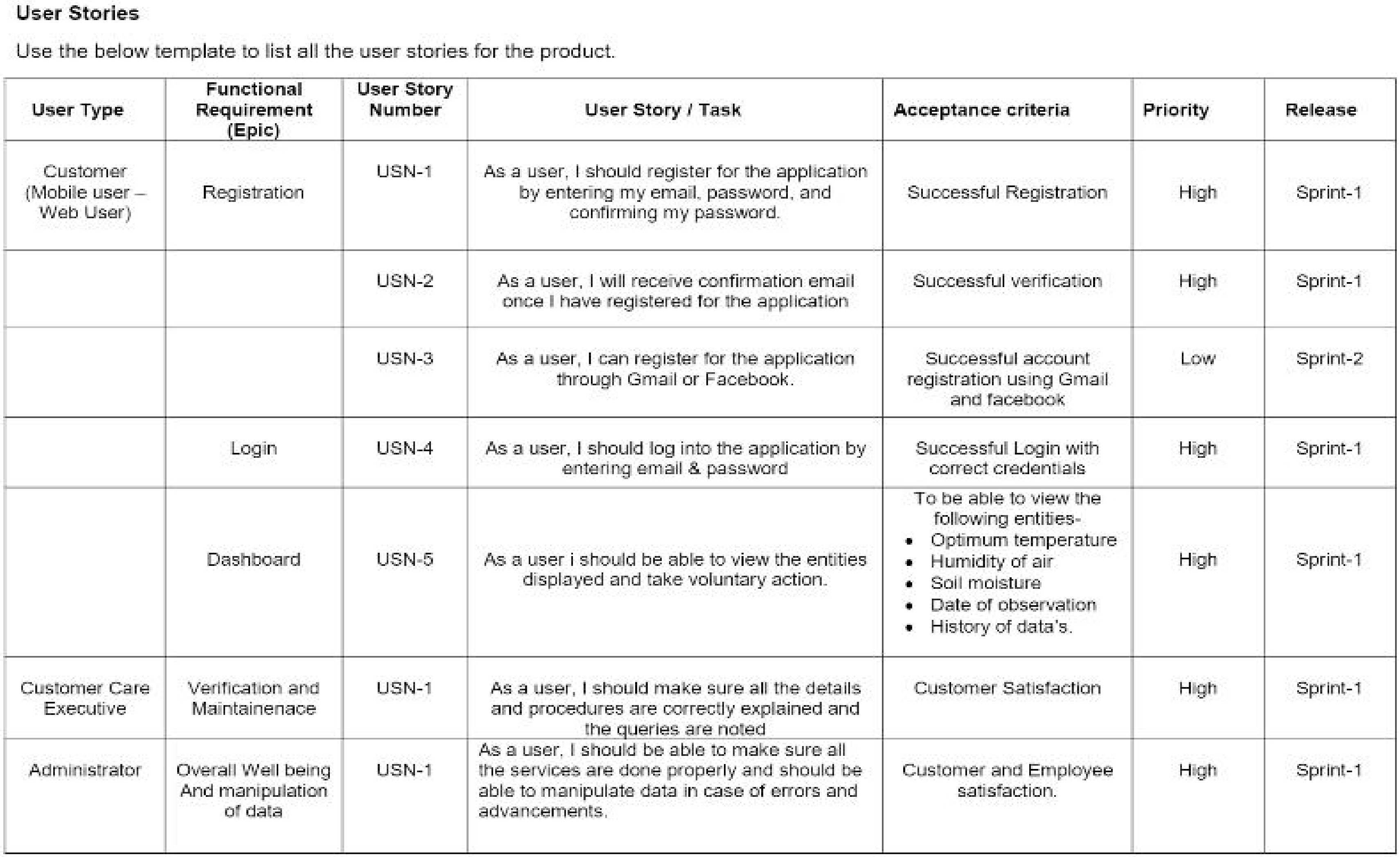
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram

4.3User Stories

Figure 1:

Architecture and data flow of the voice patient diary sample application



User Stories

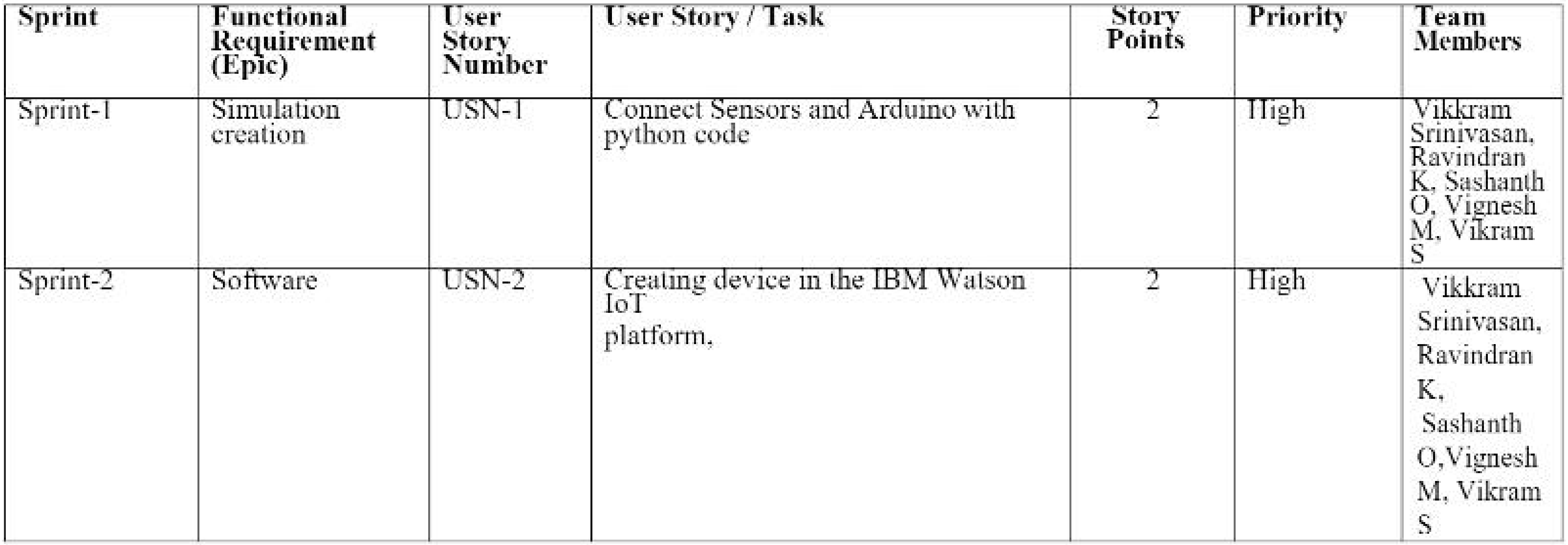
Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user – Web User)	Registration	USN-1	As a user, I should register for the application by entering my email, password, and confirming my password.	Successful Registration	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	Successful verification	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail or Facebook.	Successful account registration using Gmail and facebook	Low	Sprint-2
	Login	USN-4	As a user, I should log into the application by entering email & password	Successful Login with correct credentials	High	Sprint-1
	Dashboard	USN-5	As a user i should be able to view the entities displayed and take voluntary action.	To be able to view the following entities- • Optimum temperature • Humidity of air • Soil moisture • Date of observation • History of data's.	High	Sprint-1
Customer Care Executive	Verification and Maintainenace	USN-1	As a user, I should make sure all the details and procedures are correctly explained and the queries are noted	Customer Satisfaction	High	Sprint-1
Administrator	Overall Well being And manipulation of data	USN-1	As a user, I should be able to make sure all the services are done properly and should be able to manipulate data in case of errors and advancements.	Customer and Employee satisfaction.	High	Sprint-1

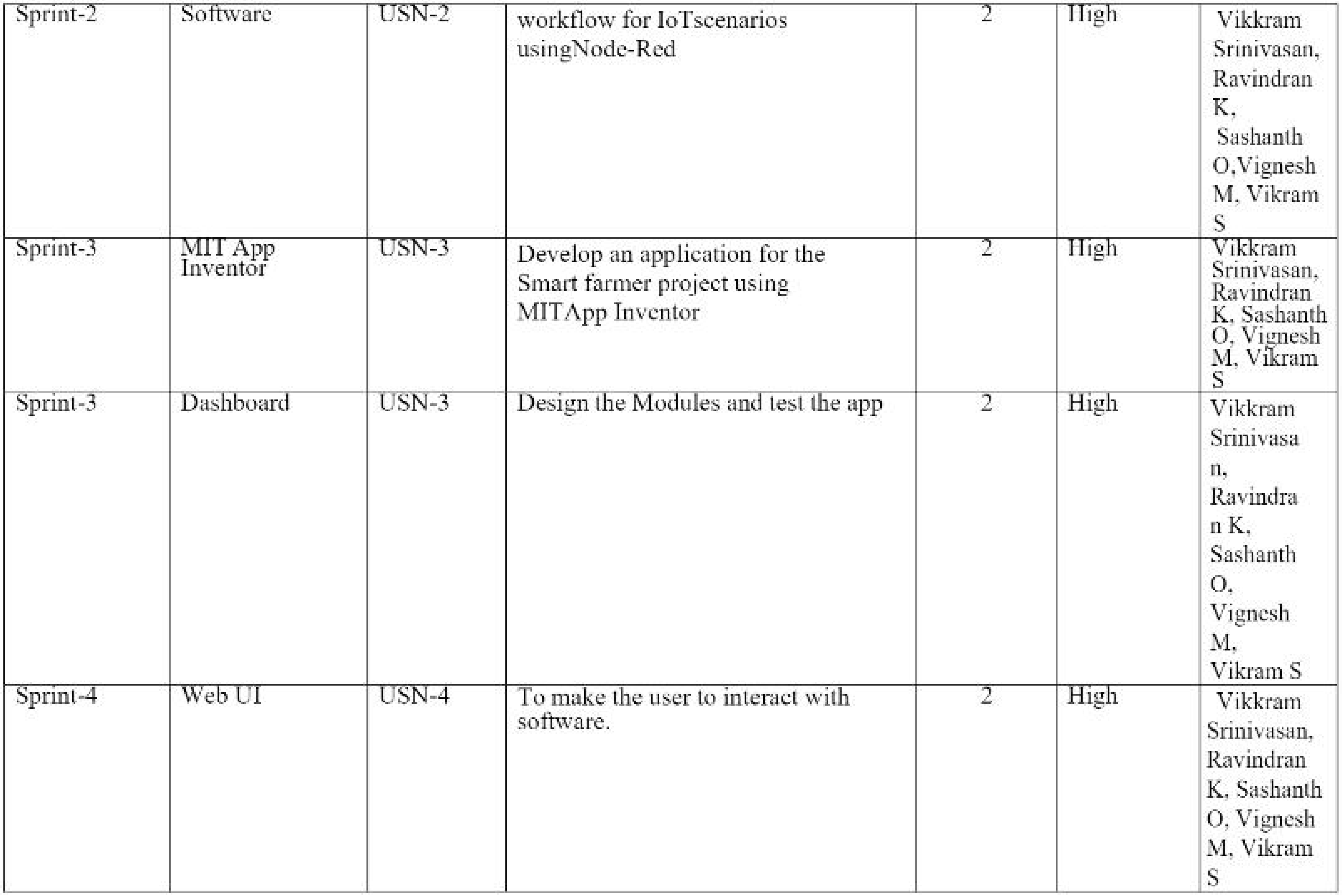
5.PROJECT PLANNING & SCHEDULING:

5.1Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Simulation creation	USN-1	Connect Sensors and Arduino with python code	2	High	Vikkram Srinivasan, Ravindran K, Sashanth O, Vignesh M, Vikram S
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform,	2	High	Vikkram Srinivasan, Ravindran K, Sashanth O,Vignesh M, Vikram S



Sprint-2	Software	USN-2	workflow for IoTscenarios usingNode-Red	2	High	Vikkram Srinivasan, Ravindran K, Sashanth O,Vignesh M, Vikram S
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmer project using MITApp Inventor	2	High	Vikkram Srinivasan, Ravindran K, Sashanth O, Vignesh M, Vikram S
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	Vikkram Srinivasan, Ravindran K, Sashanth O, Vignesh M, Vikram S
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Vikkram Srinivasan, Ravindran K, Sashanth O, Vignesh M, Vikram S

Velocity:

Project Tracker, Velocity & Burndown Chart: (4 Marks)

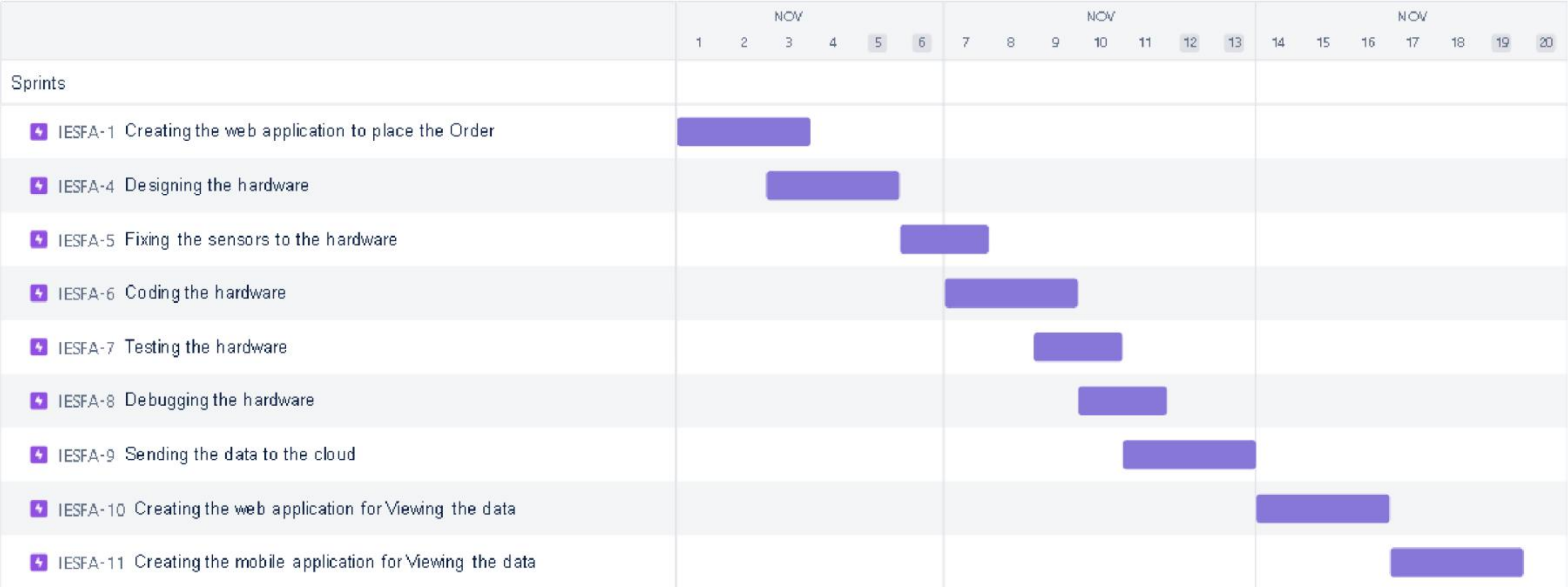
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	01 Nov 2022	07 Nov 2022		12 Nov 2022
Sprint-2	20	2 Days	08 Nov 2022	10 Nov 2022		13 Nov 2022
Sprint-3	20	5 Days	11 Nov 2022	16 Nov 2022		16 Nov 2022
Sprint-4	20	2 Days	17 Nov 2022	19 Nov 2022		19 Nov 2022

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.CODING & SOLUTIONING (Explain the features added in the project along with the code):

6.1 Feature 1

The unique feature of our project is that we can add up to 5 soil moisture and humidity sensors and we can add up to 4 pumps. The device is also capable of sustaining solar power so that it could operate without any power shortages during day time. It is capable of operating autonomously without any human intervention

6.2 Feature 2

The person who connected with the device can only view the data, other than the person connected with the device will notbe able to view the sensor readings. It enables a simple device security principle that others can not view and control the sensor readings from the device.

6.3 Feature 3

The user who is connected to the device can view the readings in mobile as well as the desktop. It is both mobile and webresponsive so there is no need to install a separate mobile application in the mobil;e devices to view the device status.

7.CODING & SOLUTIONING:

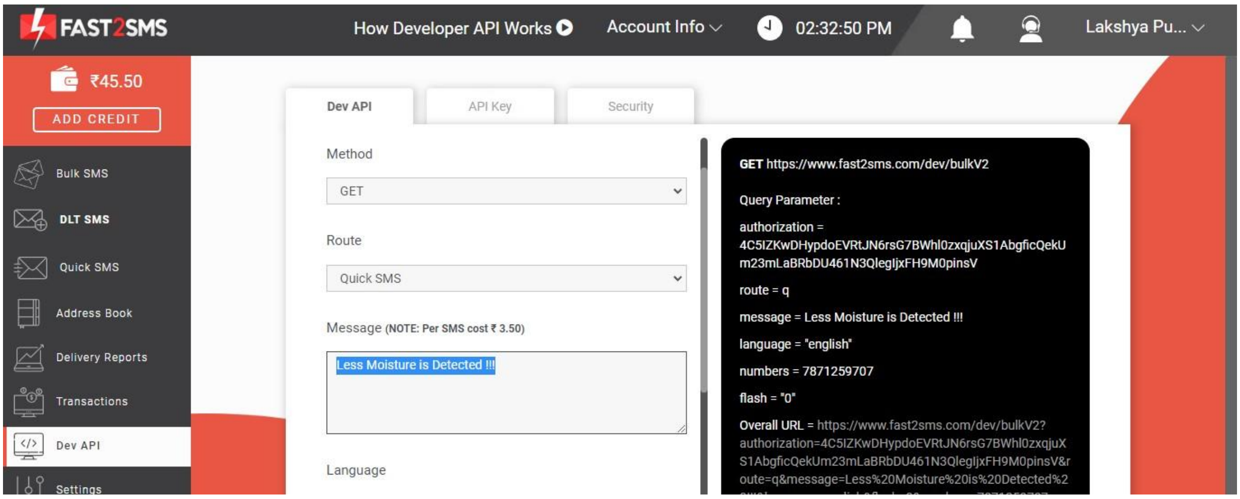
7.1Feature-1

To indicate the less moisture level in the field, so that the user can switchon motor to reach the sufficient water level. We gave a condition using python:

```
print("Published data Successfully: %s", myData)if(soil<20):
    print("Less moisture is detected")else:
    print("Moisture is sufficient")
```

7.1 Feature-2

We are using Fast2sms to send message to the user. Fast2SMSprovides a very unique and useful feature which is not available in any other bulkSMS service provider. You can send SMS to DND and Non DND numbers even if you are not registered in the DLT portal.



Method: GET

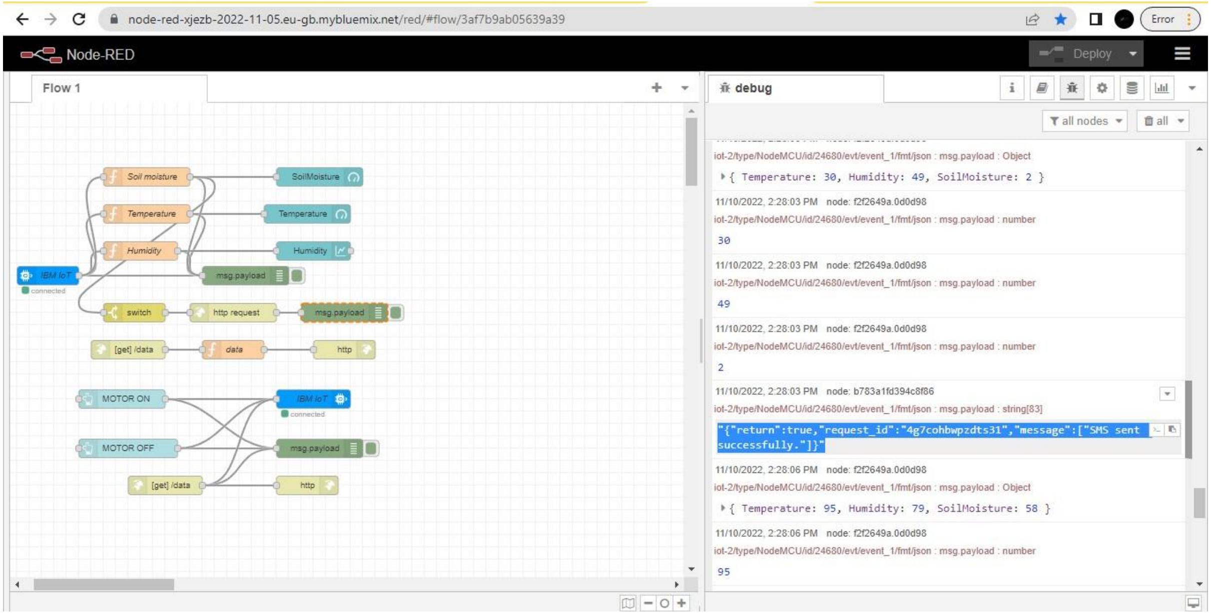
Route: Quick SMS

Message (NOTE: Per SMS cost ₹ 3.50): Less Moisture is Detected !!!

Language

```
GET https://www.fast2sms.com/dev/bulkV2

Query Parameter :

authorization =
4C5IZKwDHypdoEVRtJN6rsG7BWhl0zxqjuXS1AbgficQekU
m23mLaBRbDU461N3QlegIjxFH9M0pinsV

route = q

message = Less Moisture is Detected !!!

language = "english"

numbers = 7871259707

flash = "0"

Overall URL = https://www.fast2sms.com/dev/bulkV2?
authorization=4C5IZKwDHypdoEVRtJN6rsG7BWhl0zxqjuX
S1AbgficQekUm23mLaBRbDU461N3QlegIjxFH9M0pinsV&r
oute=q&message=Less%20Moisture%20is%20Detected%2
```

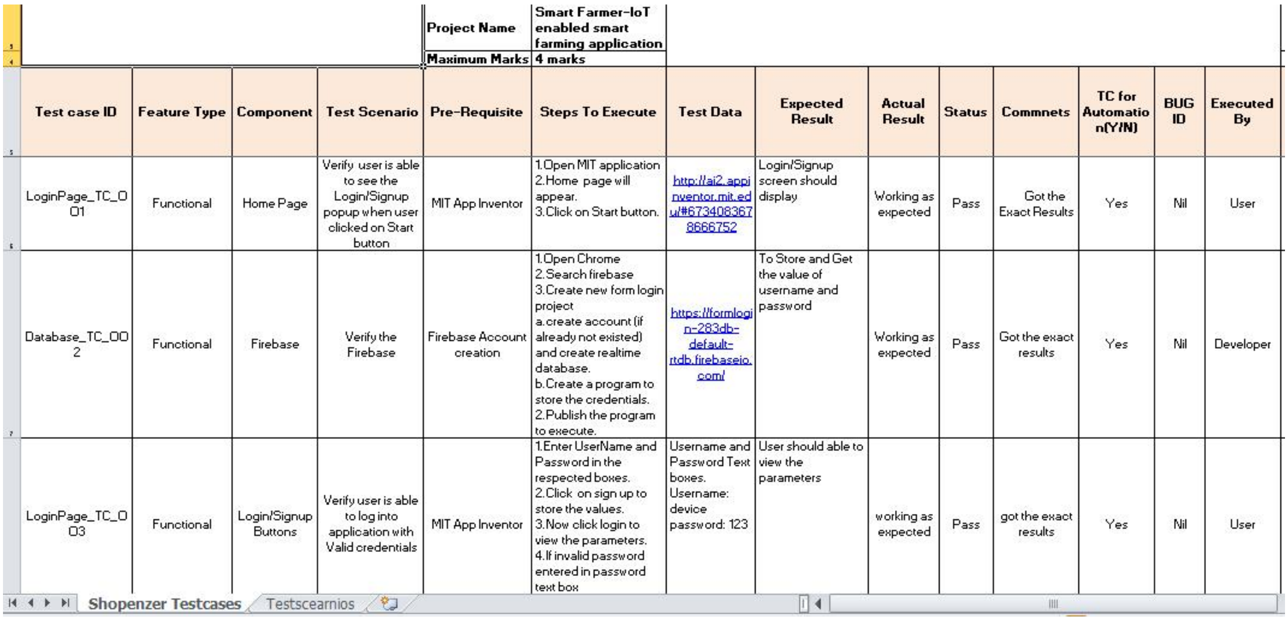
After created the fast2sms service, copy the overall url and paste in the Node redflow to send SMS to whenever the moisture level went below the threshold limit.



```
iot-2/type/NodeMCU/id/24680/evt/event_1/fmt/json : msg.payload : Object
▸{ Temperature: 30, Humidity: 49, SoilMoisture: 2 }

11/10/2022, 2:28:03 PM   node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/24680/evt/event_1/fmt/json : msg.payload : number
30

11/10/2022, 2:28:03 PM   node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/24680/evt/event_1/fmt/json : msg.payload : number
49

11/10/2022, 2:28:03 PM   node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/24680/evt/event_1/fmt/json : msg.payload : number
2

11/10/2022, 2:28:03 PM   node: b783a1fd394c8f86
iot-2/type/NodeMCU/id/24680/evt/event_1/fmt/json : msg.payload : string[83]
"{"return":true,"request_id":"4g7cohbwpzdts31","message":["SMS sent successfully."]}"

11/10/2022, 2:28:06 PM   node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/24680/evt/event_1/fmt/json : msg.payload : Object
▸{ Temperature: 95, Humidity: 79, SoilMoisture: 58 }

11/10/2022, 2:28:06 PM   node: f2f2649a.0d0d98
iot-2/type/NodeMCU/id/24680/evt/event_1/fmt/json : msg.payload : number
95
```

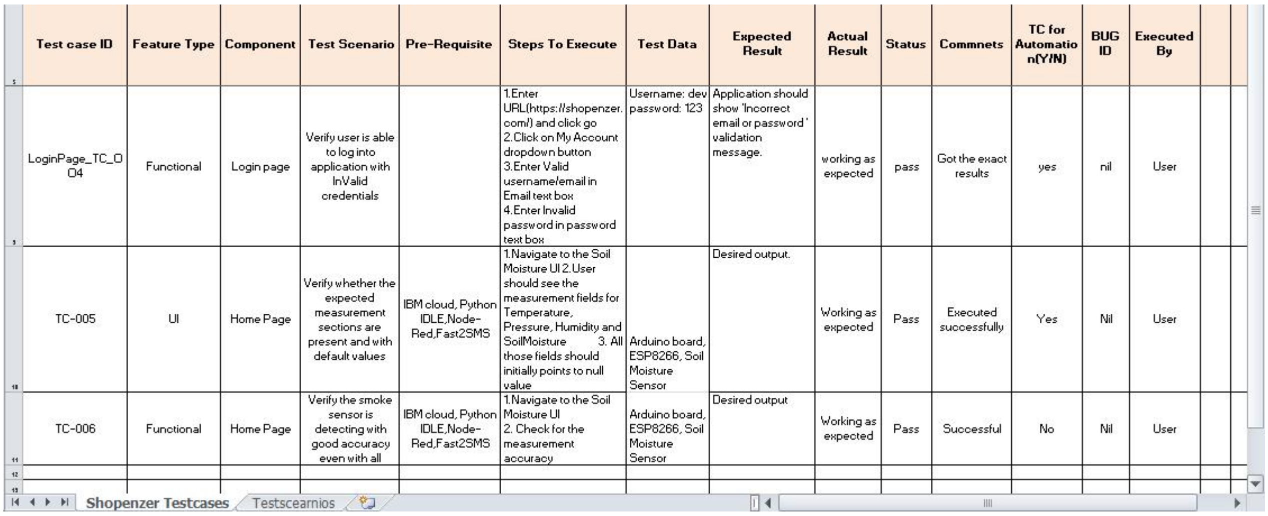
8.TESTING

8.1Test Cases



Project Name	Smart Farmer-IoT enabled smart farming application
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_O01	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on Start button	MIT App Inventor	1.Open MIT application 2.Home page will appear. 3.Click on Start button.	http://ai2.appinventor.mit.edu/#6734083678666752	Login/Signup screen should display	Working as expected	Pass	Got the Exact Results	Yes	Nil	User
Database_TC_OO2	Functional	Firebase	Verify the Firebase	Firebase Account creation	1.Open Chrome 2.Search firebase 3.Create new form login project a.create account (if already not existed) and create realtime database. b.Create a program to store the credentials. 2.Publish the program to execute.	https://formlogin-283db-default-rtdb.firebaseio.com/	To Store and Get the value of username and password	Working as expected	Pass	Got the exact results	Yes	Nil	Developer
LoginPage_TC_OO3	Functional	Login/Signup Buttons	Verify user is able to log into application with Valid credentials	MIT App Inventor	1.Enter UserName and Password in the respected boxes. 2.Click on sign up to store the values. 3.Now click login to view the parameters. 4.If invalid password entered in password text box	Username and Password Text boxes. Username: device password: 123	User should able to view the parameters	working as expected	Pass	got the exact results	Yes	Nil	User



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box	Username: dev password: 123	Application should show 'Incorrect email or password ' validation message.	working as expected	pass	Got the exact results	yes	nil	User
TC-005	UI	Home Page	Verify whether the expected measurement sections are present and with default values	IBM cloud, Python IDLE,Node-Red,Fast2SMS	1.Navigate to the Soil Moisture UI 2.User should see the measurement fields for Temperature, Pressure, Humidity and SoilMoisture 3. All those fields should initially points to null value	Arduino board, ESP8266, Soil Moisture Sensor	Desired output.	Working as expected	Pass	Executed successfully	Yes	Nil	User
TC-006	Functional	Home Page	Verify the smoke sensor is detecting with good accuracy even with all	IBM cloud, Python IDLE,Node-Red,Fast2SMS	1.Navigate to the Soil Moisture UI 2. Check for the measurement accuracy	Arduino board, ESP8266, Soil Moisture Sensor	Desired output	Working as expected	Pass	Successful	No	Nil	User

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Farmer IoT enabled smart farming application project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Improper network connectivity	10	6	4	2	22
Humidity alone is detected.	12	10	6	4	32
Continuous Battery Consumption	20	9	5	2	36
Detection Coverage Area	14	6	2	2	24
Altering the Calibration Curve	20	9	7	6	42
Maintenance	11	3	2	1	17
Accuracy detection of parameters	17	9	6	3	35
Totals	104	52	32	20	208

3. Test Case Analysis

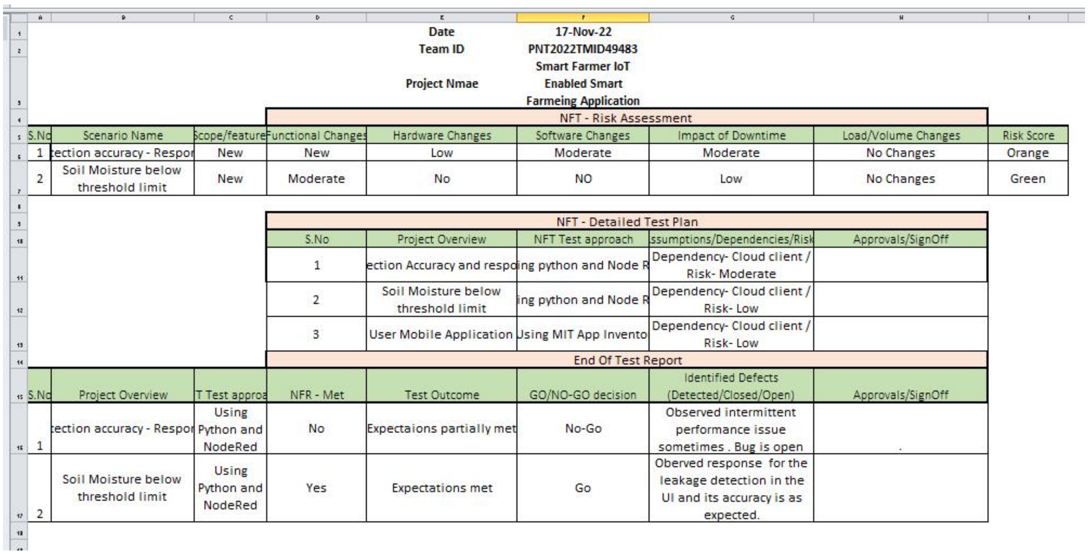
This report shows the number of test cases that have passed, failed,and untested

Section	Total Cases	Not Tested	Fail	Pass
Improper network connectivity	6	2	1	1
Humidity alone is detected.	15	0	0	15
Continuous Battery Consumption	12	0	0	12
Detection Coverage Area	5	0	1	4
Altering the Calibration Curve	4	0	0	4
Maintenance	5	0	0	5
Accuracy detection of parameters	1	0	0	1

9. RESULTS:

9.1 Performance Metrics

Performance metrics are defined as figures and data representative of an organization's actions, abilities, and overall quality.



Date	17-Nov-22
Team ID	PNT2022TMID49483
Project Nmae	Smart Farmer IoT Enabled Smart Farmeing Application

NFT - Risk Assessment

S.No	Scenario Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score
1	tection accuracy - Respor	New	New	Low	Moderate	Moderate	No Changes	Orange
2	Soil Moisture below threshold limit	New	Moderate	No	NO	Low	No Changes	Green

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	ssumptions/Dependencies/Risk	Approvals/SignOff
1	ection Accuracy and respo	ing python and Node R	Dependency- Cloud client / Risk- Moderate	
2	Soil Moisture below threshold limit	ing python and Node R	Dependency- Cloud client / Risk- Low	
3	User Mobile Application	Using MIT App Invento	Dependency- Cloud client / Risk- Low	

End Of Test Report

S.No	Project Overview	T Test approa	NFR - Met	Test Outcome	GO/NO-GO decision	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	tection accuracy - Respor	Using Python and NodeRed	No	Expectaions partially met	No-Go	Observed intermittent performance issue sometimes . Bug is open	.
2	Soil Moisture below threshold limit	Using Python and NodeRed	Yes	Expectations met	Go	Oberved response for the leakage detection in the UI and its accuracy is as expected.	

10.ADVANTAGES AND DISADVANTAGES :

Advantages:

1. Remote monitoring.
2. Autonomous watering system.
3. Environment monitoring.
4. Enables data security over sharing.
5. Can view the data in both mobile and desktop.

Disadvantages

1. Limited to small area - Requires similar units to cover larger area.
2. Few data discrepancies during bad weather.

11.CONCLUSION:

The aim of this project is to make the life and work of the farmer much easier. This can be achieved using the technique Precision Farming, this involves autonomous monitoring of crops and other environmental parameters which has an effect on the crop, these environmental conditions are:

1. Environmental Humidity
2. Environmental Temperature.
3. Soil Moisture.
4. Rain Sensing.

Above mentioned are some of the conditions monitored autonomously, threshold parameters for various crops are automatically setupon user input of crop variety to be monitored. By this system one could achieve a good yield and better nutritional crops in their agricultural produce.

12.FUTURE SCOPE:

Future scope of our project relies on the farmers and their feedbacks, in future we are planning to add the following features:

6. One device one farm - Cover the entire farm area with a single device.
7. Pest monitoring system.
8. Estimated yield calculator.
9. Estimated time of cultivation.
10. Individual cloud management dashboard.

13. APPENDIX :

Source Code:

IoT :

```
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include <Adafruit_BMP280.h>
#include <math.h>
#include <Wire.h>
#define BMP_SDA 21
#define BMP_SCL 22
#include <DFRobot_DHT11.h>
DFRobot_DHT11 DHT;
#define DHT11_PIN 4
#define rainAnalog 35
#define rainDigital 34
#define moistureDigital 32

Adafruit_BMP280 bmp280;
const char* ssid     = ""; const char*
```

```
password = ""; AsyncWebServer
server(80); AsyncEventSource
events("/events"); unsigned long
lastTime = 0; unsigned long timerDelay
= 1000;
int soil; int
rain; int
rainA;
float
temperat
ure; float
humidity;
float
pressure;
float
altitude;
long lastMsg
=
0; int
pumpRel
ayPin =
26;

#define ORG "asgkbm"
#define DEVICE_TYPE
"smart_farming"
#define DEVICE_ID "69696969"
 #defineTOKEN 12345678

char servers[] = ORG
".messaging.internetofthings.ibmcloud.com"; char pubTopic1[] =
"iot-2/evt/temperature/fmt/json"; char pubTopic2[] = "iot-
2/evt/humidity/fmt/json"; char pubTopic3[] = "iot-
2/evt/pressure/fmt/json"; char pubTopic4[] = "iot-
2/evt/altitude/fmt/json"; char authMethod[] = "use-token-auth"; char
token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(servers, 1883, NULL, wifiClient);
```

```
// Init BME280 void
initBME() {
  if (!bmp280.begin(0x76)) {
    Serial.println("Could not find a valid BMP280 sensor, check wiring!"); while(1);
  }
}


void getSensorReadings()
  { DHT.read(DHT11_PIN); temperature
  =
  DHT.temperature; humidity =
  DHT.humidity; pressure =
  bmp280.readPressure() / 100; soil =
  digitalRead(moistureDigital); rain =
  digitalRead(rainDigital); rainA =
  analogRead(rainAnalog); altitude =
  bmp280.readAltitude(1011.18); if(soil ==
  1){
    digitalWrite(pumpRelayPin, LOW);
  } else{ digitalWrite(pumpRelayPin,
  HIGH);
  }
}


// Initialize WiFi void
initWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print('.'); delay(1000);
  }
  Serial.println(WiFi.localIP());
}
```

```
String processor(const String& var)
  { getSensorReadings();
  //Serial.println(var);if (var ==
  "TEMPERATURE") {
    return String(temperature);
  } else if (var ==
  "HUMIDITY") { return
  String(humidity); } else if
  (var ==
  "PRESSURE") { return
    String(pressure);
  } else if (var ==
  "ALTITUDE") { return
  String(altitude);
  } else if (var ==
  "RAINING") { return
  String(rain);
  } else if (var ==
  "SOIL") { return String(soil);
  } return
  String();
}

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
< head >
  <title>Grow Greens Smart</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
   <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHp
vLbHG9Sr" crossorigin="anonymous">
  <link rel="icon" href="data:,">
  <style> html {font-family: Arial; display: inline-block; text-align: center;
background-color:#FCF8E8} p { font-size: 1.2rem;}
```

```
    body {  margin: 0;}
    .topnav { overflow: hidden; color: #6D9886; font-size: 1rem; }
    .content { padding: 20px; }
.card { background-color: #F2AA4CFF; box-shadow: 2px 2px 12px
      1px rgba(140,140,140,.5); border-radius: 30px;}
                .cards { max-width: 800px; margin: 0 auto; display: grid; grid-gap: 2rem;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); }
    .reading { font-size: 1.4rem; }
   </style>
  < /head >
  < body >
   <div class="topnav">
    <h1>Grow Greens Smart</h1>
   < /div >
   <div class="content">
    <div class="cards">
     <div class="card">
       <p><i class="fas fa-thermometer-half" style="color:#101820FF; font-size:25px"></i>
  Temperature</p><p><span class="reading" style = "color:#101820FF"><span id="temp"
  style="font-size:1rem; font-weight:bolder;">%TEMPERATURE%</span>

  &deg;C</span></p>
     < /div >
     <div class="card">
              <p><i class="fas fa-tint" style="color:#101820FF; font-
  size:25px"></i> Humidity</p><p><span class="reading" style="color:#101820FF;
  fontsize:1rem;"><span id="hum" style="font-size:1rem;
  fontweight:bolder;">%HUMIDITY%</span>

  &percnt;</span></p> < /div
     >
     <div class="card">
             <p><i class="fas fa-angle-double-down" style="color:#101820FF;
  font-size:25px"></i> Pressure</p><p><span class="reading" style="color:#101820FF;
  font-size:1rem;"><span id="pres" style="font-size:1rem;
```

```
font-weight:bolder;">%PRESSURE%</span> hPa</span></p>
     < /div >
     <div class="card">
               <p><i class="fas fa-mountain" style="color:#101820FF; font-size:25px"></i>
Altitude</p><p><span class="reading" style="color:#101820FF"><span
id="alti" style="font-size:1rem; font-weight:bolder;">%ALTITUDE%</span>
m</span></p>
     < /div >
     <div class="card">
               <p><i class="fas fa-cloud-rain" style="color:#101820FF; font-size:25px"></i>
Raining</p><p><span class="reading" style="color:#101820FF"><span id="rain"
style="font-size:1rem; font-weight:bolder;">%RAINING%</span></span></p> < /div >
     <div class="card">
                     <p><i class="fas fa-tree" style="color:#101820FF; font-
size:25px"></i> Moisture</p><p><span class="reading"
style="color:#101820FF"><span id="soil" style="font-size:1rem;
fontweight:bolder;">%SOIL%</span></span></p> < /div > < /div >
  < /div > <script> if
(!!window.EventSource) { var source =
new EventSource('/events');


source.addEventListener('open', function(e)
  {console.log("Events Connected");
}, false); source.addEventListener('error', function(e)
{
  if (e.target.readyState != EventSource.OPEN)
    {console.log("Events Disconnected");
  }
}, false);


source.addEventListener('message', function(e)
  {console.log("message", e.data);
}, false);


source.addEventListener('temperature', function(e) {
  console.log("temperature", e.data); document.getElementById("temp").innerHTML
  = e.data;

}, false);


source.addEventListener('humidity', function(e) {

  console.log("humidity", e.data);
  document.getElementById("hum").innerHTML = e.data;

}, false);


source.addEventListener('pressure', function(e) {

  console.log("pressure", e.data);
  document.getElementById("pres").innerHTML = e.data;

}, false);
source.addEventListener('altitude', function(e) {

  console.log("latitude", e.data);
  document.getElementById("alti").innerHTML = e.data;

}, false);


source.addEventListener('rain', function(e) { console.log("Rain",
e.data);

  if(e.data == '0')

  document.getElementById("rain").innerHTML = "Raining";

  else document.getElementById("rain").innerHTML = "Not

  Raining";

}, false);


source.addEventListener('soil', function(e) { console.log("Soil
Moisture", e.data);

  if(e.data == '1')

    document.getElementById("soil").innerHTML = "Less Water";

  else document.getElementById("soil").innerHTML = "Enough

  Water";

}, false);
```

```
}
</script>
< /body > </html>)rawliteral";


void setup() { Serial.begin(115200);
 pinMode(rainDigital, INPUT);
 pinMode(moistureDigital, INPUT);
 pinMode(pumpRelayPin, OUTPUT); initWiFi();
  initBME();
  // Handle Web Server server.on("/", HTTP_GET,
  [](AsyncWebServerRequest * request) { request->send_P(200,
  "text/html", index_html, processor);

  }) ;
  // Handle Web Server Events
  events.onConnect([](AsyncEventSourceClient * client) { if
  (client->lastId()) {
                Serial.printf("Client reconnected! Last message ID that it got is: %u\n",
client->lastId());
    }
    // send event with message "hello!", id current millis
    // and set reconnect delay to 1 second client-
    >send("hello!", NULL, millis(), 10000);
  }) ;
  server.addHandler(&events)
  ; server.begin(); if
  (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(servers); while
    (!client.connect(clientId, authMethod, token))
    {Serial.print("."); delay(500);
    }
    Serial.println("Bluemix connected");
  }
}
void loop() {
```

```
client.loop(); long now =
millis(); if (now - lastMsg
> 3000) { lastMsg
 = now;



 String payload =
 "{\"temperature\":"; payload += temperature;
 payload += "}";
 Serial.print("Sending payload: ");
 Serial.println(payload); if (client.publish(pubTopic1,
 (char*) payload.c_str())) { Serial.println("Publish ok");
 }  else  {
   Serial.println("Publish failed");
 }
 String payload1 = "{\"humidity\":";
 payload1 += humidity; payload1
 += "}";
 Serial.print("Sending payload: ");
 Serial.println(payload1); if (client.publish(pubTopic2,
 (char*) payload1.c_str())) { Serial.println("Publish ok");
 }  else  {
   Serial.println("Publish failed");
 }



 String payload2 = "{\"pressure\":";
 payload2 += pressure; payload2
 += "}";
 Serial.print("Sending payload: ");
 Serial.println(payload2); if (client.publish(pubTopic3,
 (char*) payload2.c_str())) { Serial.println("Publish ok");
 }  else  {
   Serial.println("Publish failed");
 }
```

```
    String payload3 = "{\"altitude\":";
    payload3 += altitude; payload3
    += "}";
    Serial.print("Sending payload: ");
    Serial.println(payload3); if (client.publish(pubTopic4,
    (char*) payload3.c_str())) { Serial.println("Publish ok");
    }  else  {
      Serial.println("Publish failed");
    }

  } if ((millis() - lastTime) > timerDelay)
  {
    getSensorReadings();
    Serial.printf("Temperature = %.2f ºC \n", temperature); Serial.printf("Humidity = %.2f
    \n", humidity);
    Serial.printf("Pressure = %.0f hPa \n", pressure);
    Serial.printf("Altitude = %.0f m \n", altitude);
    Serial.printf("Rain = %d\n", rain);
    Serial.printf("Rain = %d\n", rainA);
    Serial.printf("Soil = %d\n", soil);
    Serial.println();

    // Send Events to the Web Server with the Sensor Readings
    events.send("ping", NULL, millis());
    events.send(String(temperature).c_str(), "temperature", millis());
    events.send(String(humidity).c_str(), "humidity", millis());
    events.send(String(pressure).c_str(), "pressure", millis());
    events.send(String(altitude).c_str(), "altitude", millis());
    events.send(String(rain).c_str(), "rain", millis());
    events.send(String(soil).c_str(), "soil", millis());

    lastTime = millis();
  }
}
```

github link : https://github.com/IBM-EPBL/IBM-Project-18654-1659688068