NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READINESS FOR INNOVATION EMPLOYMENT AND ENTREPRENEURSHIP

PROJECT REPORT

PROJECT NAME : SMART SOLUTIONS FOR RAILWAYS

TEAM ID : PNT2022TMID44252

TEAM LEADER : NAVEEN T

TEAM MEMBERS : NAVEENKUMAR J
NAVEENKUMAR S
SUGESH N

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION

BUILDERS ENGINEERING COLLEGE
NATHAKADAIYUR

ANNA UNIVERSITY
CHENNAI-600025

INDEX

1. INTRODUCTION

1.1 Project Overview

- ❖ The device will detect the animals and birds using the Clarifai service.
- ❖ If any animal or bird is detected the image will be captured and stored in the
- ❖ IBM Cloud object storage.
- ❖ It also generates an alarm and prevents animals from destroying the crop .
- ❖ The image URL will be stored in the IBM Cloudant DB service.
- ❖ The device will also monitor the soil moisture levels, temperature, and humidity values and send them to the IBM IoT Platform.
- ❖ The image will be retrieved from Object storage and displayed in the web application.
- ❖ A web application is developed to visualize the soil moisture, temperature, and humidity values.
- ❖ Users can also control the motors through web applications.

1.2 PURPOSE

An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop. This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field. The motors and sprinklers in the field can be controlled using the mobile application.

2. LITERATURE SURVEY

2.1 Existing Problem

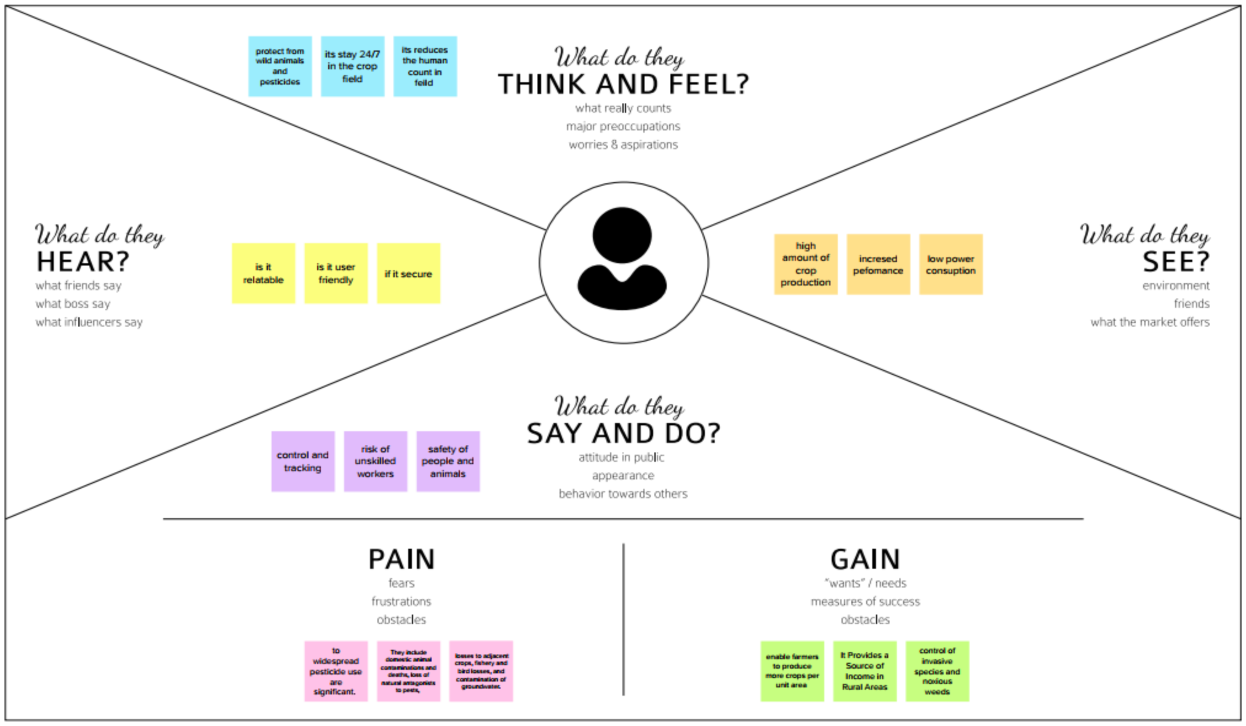
Most of the farmers are facing many problems nowadays due to many reasons. Our problem to solve is the invasion of various species such as birds and animals that harm the crops that are being cultivated. Various types of species such as birds and animals come to the cultivation field according to the crop that is being cultivated and also according to the season of cultivation. Some wild animals enter the field during night times when the field is near a forest region or when the farm cultivates some fruits and other crops that attract animals. Some animals cross the field in search of food and water and also the birds enter the field for food and they damage all the crops. When the animals enter the field they not only eat food but they also damage the entire field by walking upon the crops and also by spoiling the food crops. The birds, by entering the field they come to eat seeds of the crops and also they tend to drag the crops and ruin the entire field. Some birds enter the field to eat the insects and pests in the field.

2.2 Problem Statement Definition

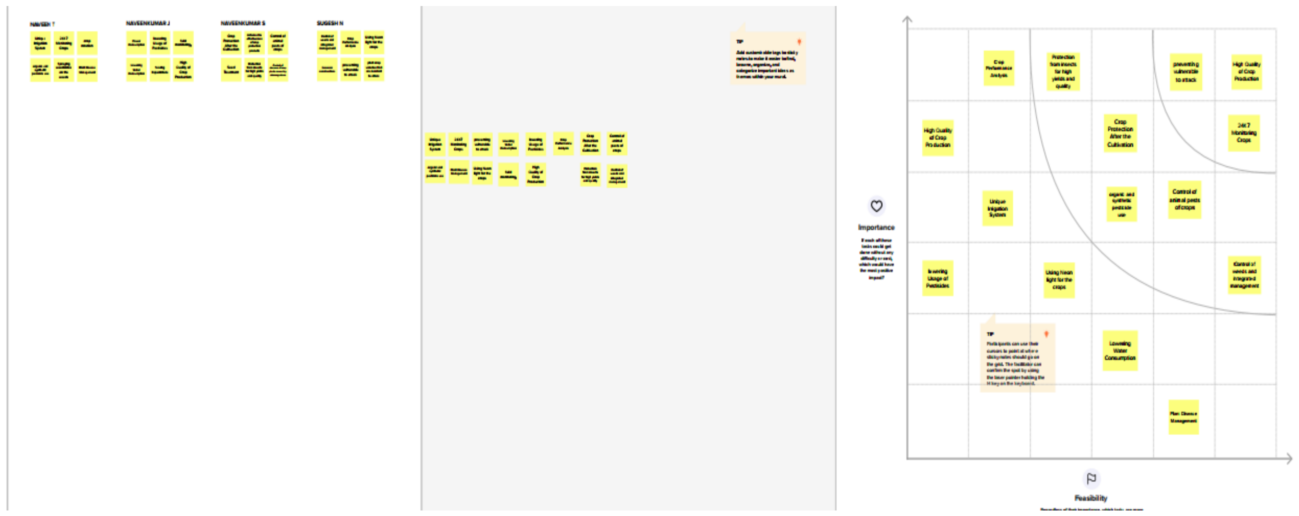
Most of the farmers are facing many problems nowadays due to many reasons. Our problem to solve is the invasion of various species such as birds and animals that harm the crops that are being cultivated. Various types of species such as birds and animals come to the cultivation field according to the crop that is being cultivated and also according to the season of cultivation. Some wild animals enter the field during night times when the field is near a forest region or when the farm cultivates some fruits and other crops that attract animals.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


protect from wild animals and pesticides
its stay 24/7 in the crop field
its reduces the human count in feild
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they
HEAR?
what friends say
what boss say
what influencers say
is it relatable
is it user friendly
if it secure
high amount of crop production
incresed pefomance
low power consuption
What do they
SEE?
environment
friends
what the market offers
control and tracking
risk of unskilled workers
safety of people and animals
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
to widespread pesticide use are significant.
They include domestic animal contaminations and deaths, loss of natural antagonists to pests,
losses to adjacent crops, fishery and bird losses, and contamination of groundwater.
GAIN
"wants" / needs
measures of success
obstacles
enable farmers to produce more crops per unit area
It Provides a Source of Income in Rural Areas
control of invasive species and noxious weeds

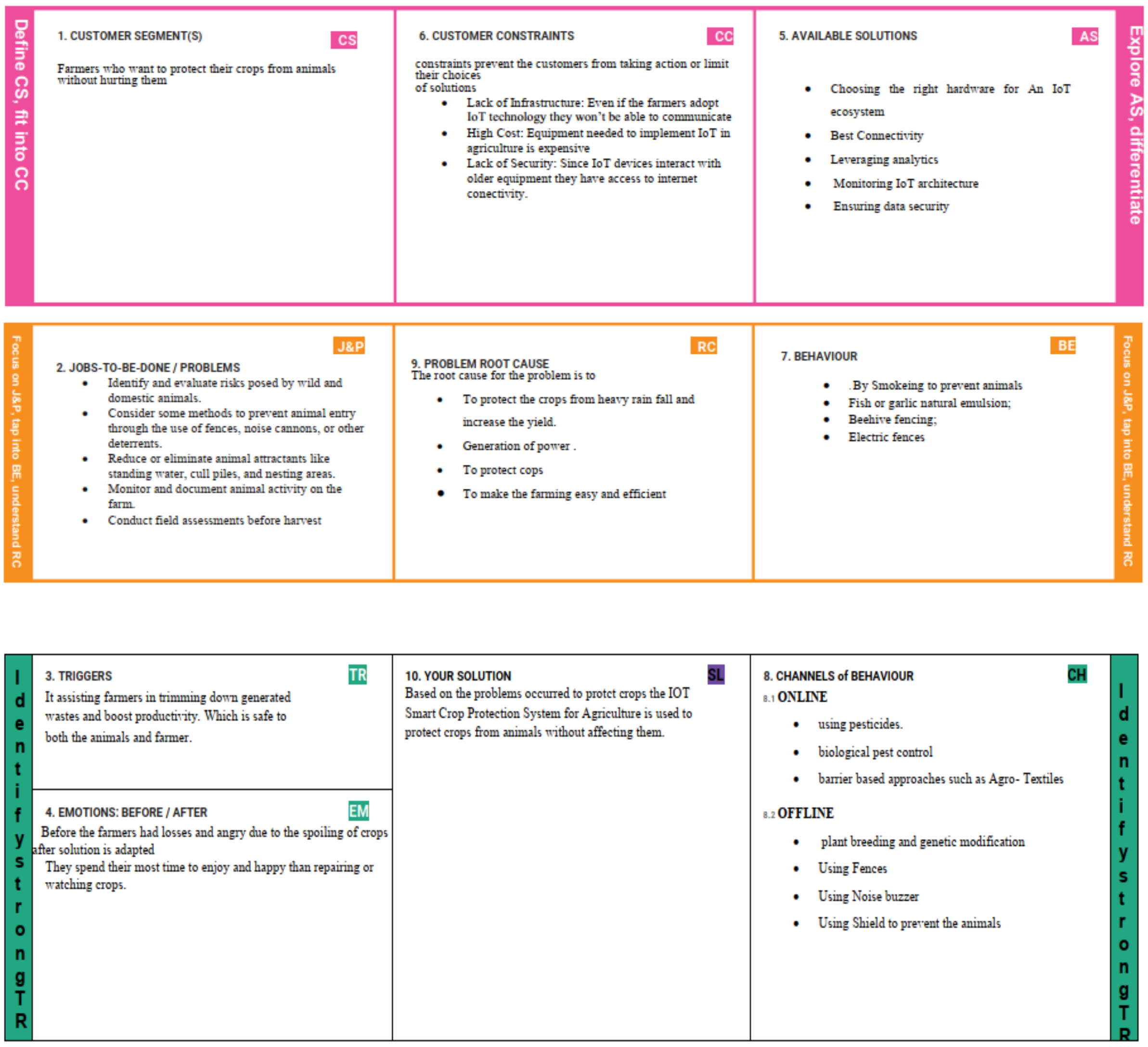
3.2 Ideation and Brainstorming


NAVEEN T
NAVEENKUMAR J
NAVEENKUMAR S
SUGESH N
TIP
Crop Performance Analysis
Protection from insects for high yields and quality
preventing vulnerable to attack
High Quality of Crop Production
High Quality of Crop Production
Crop Protection After the Cultivation
24X7 Monitoring Crops
Unique Irrigation System
organic and synthetic pesticide use
Control of animal pests of crops
Importance
lowering Usage of Pesticides
Using Neon light for the crops
Control of weeds and integrated management
TIP
Lowering Water Consumption
Plan Disease Management
Feasibility

3.3 Proposed Solution

S.No	Parameter	Description
1.	• Problem Statement (Problem to be solved)	• Crops are not irrigated properly due to insufficient labour forces. • Improper maintenance of crops against various environmental factors such as temperature climate, topography and soil quality which results in crop destruction. • Requires protecting crops from Wild animals attacks, birds and pests.
2.	• Idea / Solution description	• Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON and OFF the motor pump for managing the excess water level. It will be updated to authorities through IOT. • Temperature sensor connected to microcontroller is used to monitor the temperature in the field. • Image processing techniques with IOT is followed for crop protection against animal attacks.
3.	Novelty / Uniqueness	• Automatic crop maintenance and protection using embedded and IOT technology.
4.	Social Impact / Customer Satisfaction	• This proposed system provides many facilities which helps the farmers to maintain the crop field without much loss.
5.	Business Model (Revenue Model)	• This prototype can be developed as product with minimum cost with high performance .
6.	Scalability of the Solution	• This can be developed to a scalable product by using sensors and transmitting the data through Wireless Sensor Network and Analysing the data in cloud and operations is performed using robots .

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Farmers who want to protect their crops from animals without hurting them

6. CUSTOMER CONSTRAINTS CC

constraints prevent the customers from taking action or limit their choices of solutions

- Lack of Infrastructure: Even if the farmers adopt IoT technology they won't be able to communicate
- High Cost: Equipment needed to implement IoT in agriculture is expensive
- Lack of Security: Since IoT devices interact with older equipment they have access to internet conectivity.

5. AVAILABLE SOLUTIONS AS

- Choosing the right hardware for An IoT ecosystem
- Best Connectivity
- Leveraging analytics
- Monitoring IoT architecture
- Ensuring data security

Explore AS, differentiate

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

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

- Identify and evaluate risks posed by wild and domestic animals.
- Consider some methods to prevent animal entry through the use of fences, noise cannons, or other deterrents.
- Reduce or eliminate animal attractants like standing water, cull piles, and nesting areas.
- Monitor and document animal activity on the farm.
- Conduct field assessments before harvest

9. PROBLEM ROOT CAUSE RC

The root cause for the problem is to

- To protect the crops from heavy rain fall and increase the yield.
- Generation of power .
- To protect cops
- To make the farming easy and efficient

7. BEHAVIOUR BE

- .By Smokeing to prevent animals
- Fish or garlic natural emulsion;
- Beehive fencing;
- Electric fences

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

Identify strong TR & EM

3. TRIGGERS TR

It assisting farmers in trimming down generated wastes and boost productivity. Which is safe to both the animals and farmer.

4. EMOTIONS: BEFORE / AFTER EM

Before the farmers had losses and angry due to the spoiling of crops after solution is adapted

They spend their most time to enjoy and happy than repairing or watching crops.

10. YOUR SOLUTION SL

Based on the problems occurred to protct crops the IOT Smart Crop Protection System for Agriculture is used to protect crops from animals without affecting them.

8. CHANNELS of BEHAVIOUR CH

8.1 **ONLINE**

- using pesticides.
- biological pest control
- barrier based approaches such as Agro- Textiles

8.2 **OFFLINE**

- plant breeding and genetic modification
- Using Fences
- Using Noise buzzer
- Using Shield to prevent the animals

Identify strong TR

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Data Collecting	Smart farming based IOT Extract Data from Sources
FR-2	Data Supervision and Management	Transform data into proper format Effective Environmental Indicators Performance Evalution of Proposed system Mobile App for Farming Staffs
FR-3	Feature Selection	Wrapper feature selection approach to analyze the environment indicators,which selects the effective indicators on the farming system
FR-4	Analysis of Agricultural data	Effective environmental indicators
FR-5	Performance evalution of proposed algorithm	Dashboard For Farming Staffs and Mobile Apps

4.2 Non Functional requirement

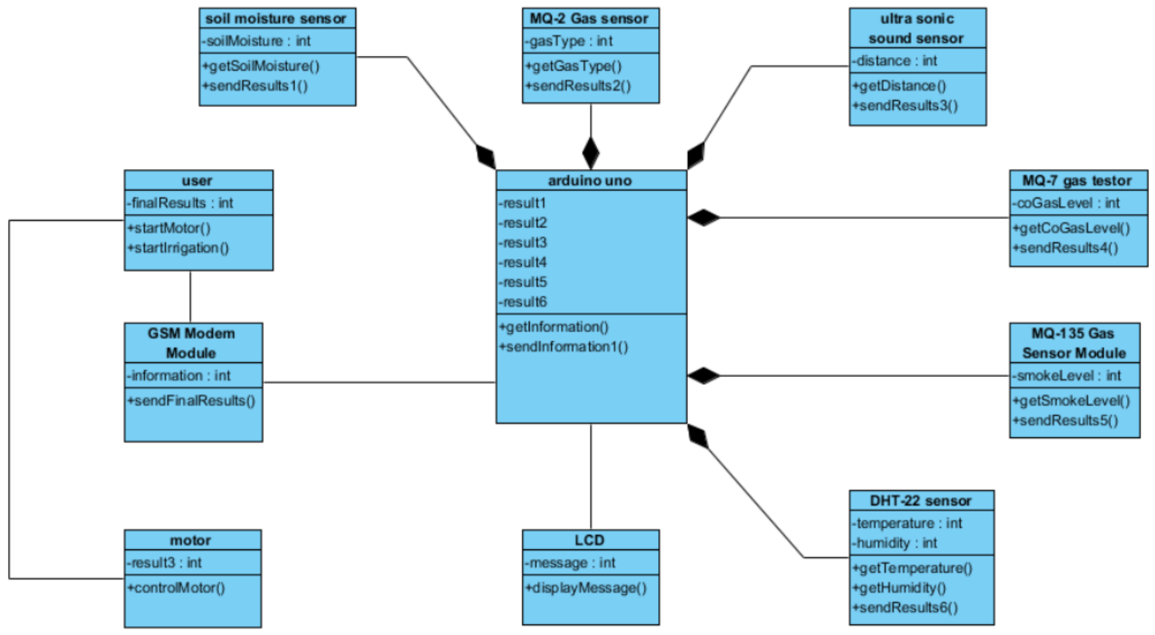
Following are the non-functional requirements of the proposed solution.

- NFR-1 Usability You will able to allocate money to different priorities and also help you to cut down on unnecessary spending
- NFR-2 Security More security of the customer data and bank account details.
- NFR-3 Reliability Used to manage his/her expense so that the user is the path of financial stability. It is categorized by week, month, and year and also helps to see more expenses made. Helps to define their own categories.
- NFR-4 Performance The types of expense are categories along with an option.Throughput of the system is increased due to light weight database support.

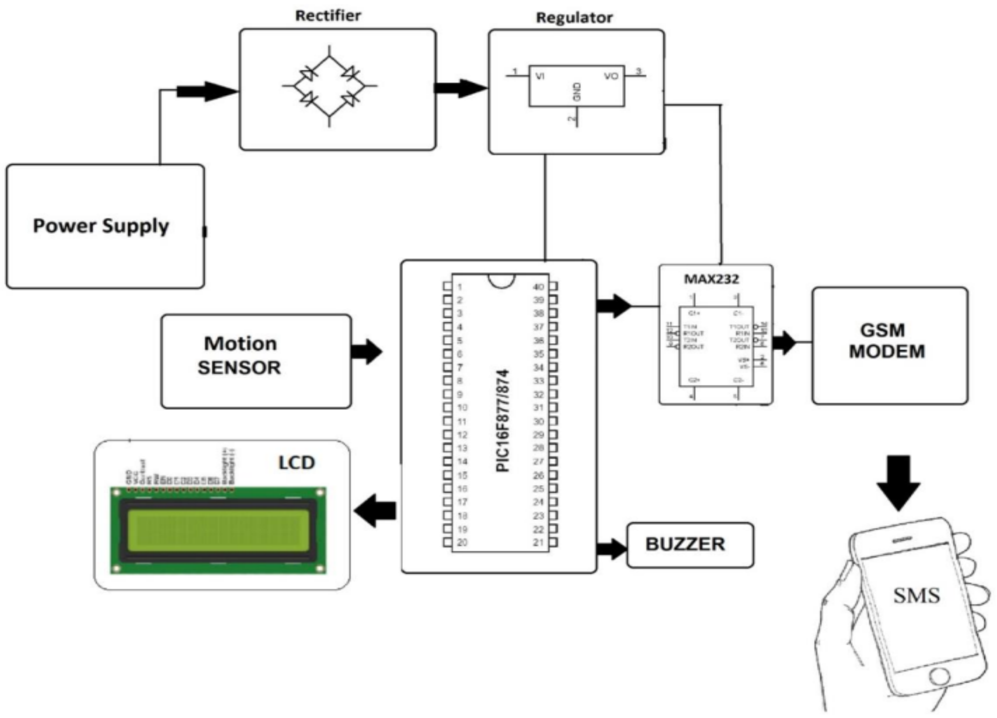
- NFR-5 Availability Able to track business expenses and monitor important for maintaining healthy cash flow. NFR-6 Scalability The ability to appropriately handle increasing demands.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution Architecture Diagram:

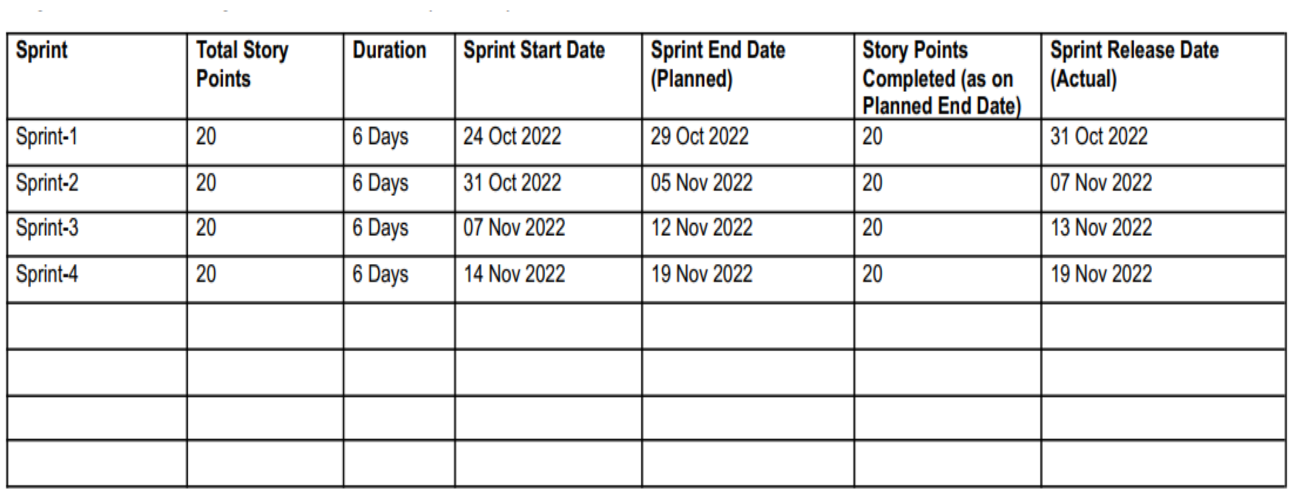


6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Naveen T
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Naveen kumar J
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Naveen Kumar S
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Sugesh N
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	
	Dashboard					

Project Tracker, Velocity & Burndown Chart :



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	24 Oct 2022	29 Oct 2022	20	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING AND SOLUTIONING

7.1 Features Feature

- 1 Detect the Temperature Feature
- 2 Detect the Humidity Feature
- 3 Detect the Moisture Feature
- 4 Detect the Animals

Codes:

PYTHON CODE TO IBM:

```
import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
import sys
#IBM Watson Device Credentials.
organization = "op701j"
deviceType = "Lokesh"
deviceId = "Lokesh89"
authMethod = "token"
```

```
authToken = "1223334444"
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="sprinkler_on":
    print ("sprinkler is ON")
 else :
    print ("sprinkler is OFF")
 #print(cmd)


try:
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
 deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
sys.exit()
#Connecting to IBM watson.
deviceCli.connect()
while True:
#Getting values from sensors.
 temp_sensor = round( random.uniform(0,80),2)
 PH_sensor = round(random.uniform(1,14),3)
 camera = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not Detected",]
 camera_reading = random.choice(camera)
 flame = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not Detected",]
 flame_reading = random.choice(flame)
 moist_level = round(random.uniform(0,100),2)
 water_level = round(random.uniform(0,30),2)

#storing the sensor data to send in json format to cloud.

 temp_data = { 'Temperature' : temp_sensor }
 PH_data = { 'PH Level' : PH_sensor }
 camera_data = { 'Animal attack' : camera_reading}
 flame_data = { 'Flame' : flame_reading }
 moist_data = { 'Moisture Level' : moist_level}
 water_data = { 'Water Level' : water_level}

# publishing Sensor data to IBM Watson for every 5-10 seconds.
 success = deviceCli.publishEvent("Temperature sensor", "json", temp_data,
qos=0)
 sleep(1)
 if success:
   print (" ............................publish ok............................ ")
 print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson")

 success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
 sleep(1)
 if success:
   print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")
 success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
 sleep(1)
 if success:
    print ("Published Animal attack %s " % camera_reading, "to IBM Watson")
 success = deviceCli.publishEvent("Flame sensor", "json", flame_data, qos=0)
 sleep(1)
 if success:
    print ("Published Flame %s " % flame_reading, "to IBM Watson")

 success = deviceCli.publishEvent("Moisture sensor", "json", moist_data, qos=0)
 sleep(1)
 if success:
     print ("Published Moisture Level = %s " % moist_level, "to IBM Watson")

 success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
 sleep(1)
 if success:
    print ("Published Water Level = %s cm" % water_level, "to IBM Watson")
 print ("")
 #Automation to control sprinklers by present temperature an to send alert message
to IBM Watson.

 if (temp_sensor > 35):
    print("sprinkler-1 is ON")
 success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is
high, sprinkerlers are turned ON" %temp_sensor }
, qos=0)
 sleep(1)
 if success:
    print( 'Published alert1 : ', "Temperature(%s) is high, sprinkerlers are turned
ON" %temp_sensor,"to IBM Watson")
 print("")
else:
 print("sprinkler-1 is OFF")
 print("")

 #To send alert message if farmer uses the unsafe fertilizer to crops.

 if (PH_sensor > 7.5 or PH_sensor < 5.5):
    success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH
level(%s) is not safe,use other fertilizer" %PH_sensor } ,
qos=0)
 sleep(1)
 if success:
    print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other
fertilizer" %PH_sensor,"to IBM Watson")
 print("")

 #To send alert message to farmer that animal attack on crops.

 if (camera_reading == "Detected"):
    success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on
crops detected" }, qos=0)
```

```
 sleep(1)
 if success:
    print('Published alert3 : ' , "Animal attack on crops detected","to IBM 
Watson","to IBM Watson")
 print("")
 #To send alert message if flame detected on crop land and turn ON the splinkers to 
take immediate action.

 if (flame_reading == "Detected"):
    print("sprinkler-2 is ON")
 success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected 
crops are in danger,sprinklers turned ON" }, qos=0)
 sleep(1)
 if success:
    print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers 
turned ON","to IBM Watson")

 #To send alert message if Moisture level is LOW and to Turn ON Motor-1 for 
irrigation.
 if (moist_level < 20):
    print("Motor-1 is ON")
 success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) 
is low, Irrigation started" %moist_level }, qos=0)
 sleep(1)
 if success:
    print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started" 
%moist_level,"to IBM Watson" )
```

```
 print("")
 #To send alert message if Water level is HIGH and to Turn ON Motor-2 to take
water out.
 if (water_level > 20):
    print("Motor-2 is ON")
 success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is
high, so motor is ON to take water out "
%water_level }, qos=0)
 sleep(1)
 if success:
    print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water
out " %water_level,"to IBM Watson" )
    print("")
 #command recived by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```