KARAPAKKAM, CHENNAI - 600097

IoT Based Smart Crop Protection System for Agriculture

TEAM ID: PNT2022TMID27380

TEAM MEMBERS:

RADHA PRABHAKARAN (311019106701)

PAARVATHI S. (311019106046)

PRASHANTHI B. (311019106051)

PREETHI S. (311019106054)

Project Report

1. INTRODUCTION

1.1 Project Overview

The project title is real-time river quality monitoring and control system. Nowadays most of the rivers are filled with dirty water, so we are not able to draw the water and people who only depend on the stream of water near their houses are not able to make use of it. River water plays a crucial role in so many lives. The purity of water mainly depends upon the pH level. In this project, we control the river water pH level with the help of IoT. The current water quality monitoring system is a manual system with a monotonous process and is very time-consuming. This paper proposes a sensor-based water quality monitoring system. The main components of a Wireless Sensor Network (WSN) include a microcontroller for processing the system, a communication system for inter and intra-node communication, and several sensors. Real-time data access can be done by using remote monitoring and Internet of Things (IoT) technology.

1.2 Purpose

In this project, we depict the design of a Wireless Sensor Network (WSN) that assists to monitor the quality of water with the support of information sensed by the sensors dipped in water. Using different sensors, this system can collect various parameters from water, such as pH, dissolved oxygen, turbidity, conductivity, temperature, and so on. The rapid development of WSN technology provides a novel approach to real-time data acquisition, transmission, and processing. The clients can get ongoing water quality information from far away. Now a day's The Internet of things (IoT) is an innovative technological phenomenon. It is shaping today's world and is used in different fields for collecting, monitoring, and analysis of data from remote locations. IoT-integrated networks are everywhere, starting from smart cities, smart power grids, and smart supply chains to smart wearables. Though IoT is still under-applied in the field of environment, it has huge potential. It can be applied to detect forest fires and early earthquakes, reduce air pollution, monitor snow levels, prevent landslides, and avalanches, etc. Moreover, it can be implemented in the field of water quality monitoring and control systems. Water quality monitoring has gained more interest among researchers in this twenty-first century. Numerous works are either done or ongoing on this topic, focusing on various aspects of it. The key theme of all the projects was to develop an efficient, cost-effective, real-time water quality monitoring system that will integrate a wireless sensor network and the internet of things.

2. LITERATURE SURVEY

2.1 Existing problem

The existing system mainly provides surveillance functionality. These systems also do not provide protection from wild animals, especially in such an application area. They also need to take measures based on the type of animal that tries to enter the area, as different methods are adopted to prevent different animals from entering restricted areas. The other commonly used methods used by farmers to prevent crop vandalization by animals include building physical barriers, use of electric fences and manual surveillance and several other exhaustive and dangerous methods.

2.2 References

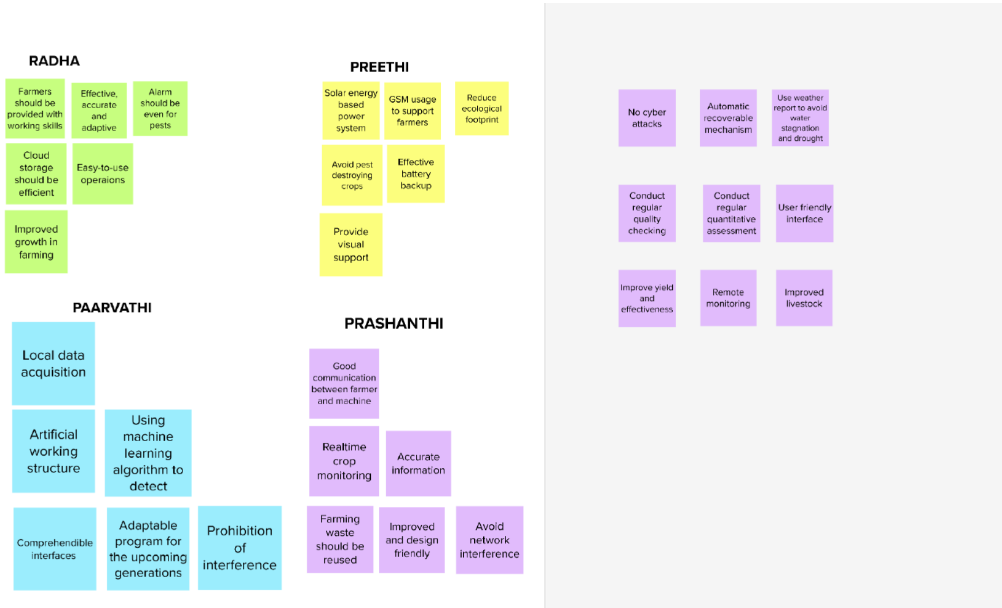
i. Mr.Pranav shitap, Mr.Jayesh redij, Mr.Shikhar Singh, Mr.Durvesh Zagade, Dr. Sharada Chougule. Department of ELECTRONICS AND TELECOMMUNICATION ENGINEERING, Finolex Academy of Management and technology, ratangiri, India.
ii. N.Penchalaiah, D.Pavithra, B.Bhargavi, D.P.Madhurai, K.EliyasShaik,S.Md.sohaib.Assitant Professor, Department of CSE, AITS, Rajampet, India UG Student, Department of CSE, AITS, Rajampet, India.
iii. Mr.P.Venkateswara Rao, Mr.Ch Shiva Krishna, MR M Samba Siva ReddyLBRCE, LBRCE, LBRCE.
iv. Mohit Korche, Sarthak Tokse, ShubhamShirbhate, Vaibhav Thakre, S. P. Jolhe(HOD). Students,

2.3 Problem Statement Definition

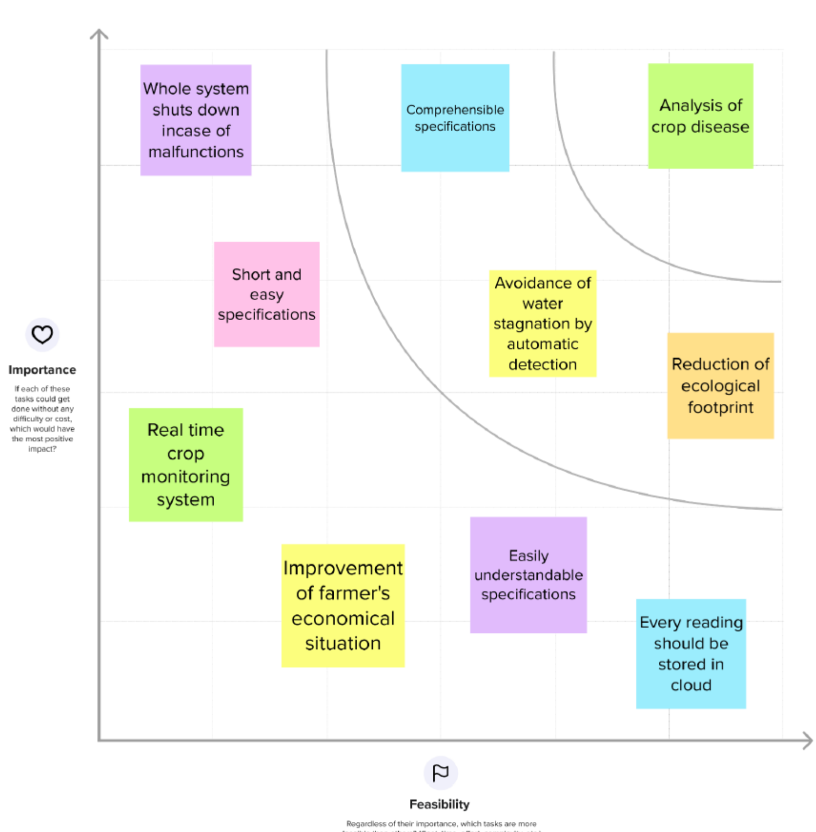
In spite of economic development, agriculture is the backbone of the economy. Crop farms are many times ravaged by local animals like buffaloes, cows, goats, and birds and fire etc. this leads to huge losses for the farmers. It is not possible for farmers to blockade entire fields or stay 24 hours and guard them. Agriculture meets the food requirements of the people and produces several raw materials for industries. But because of animal interference and fire in agricultural lands, there will be a huge loss of crops. Crops will be totally getting destroyed

3. **IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

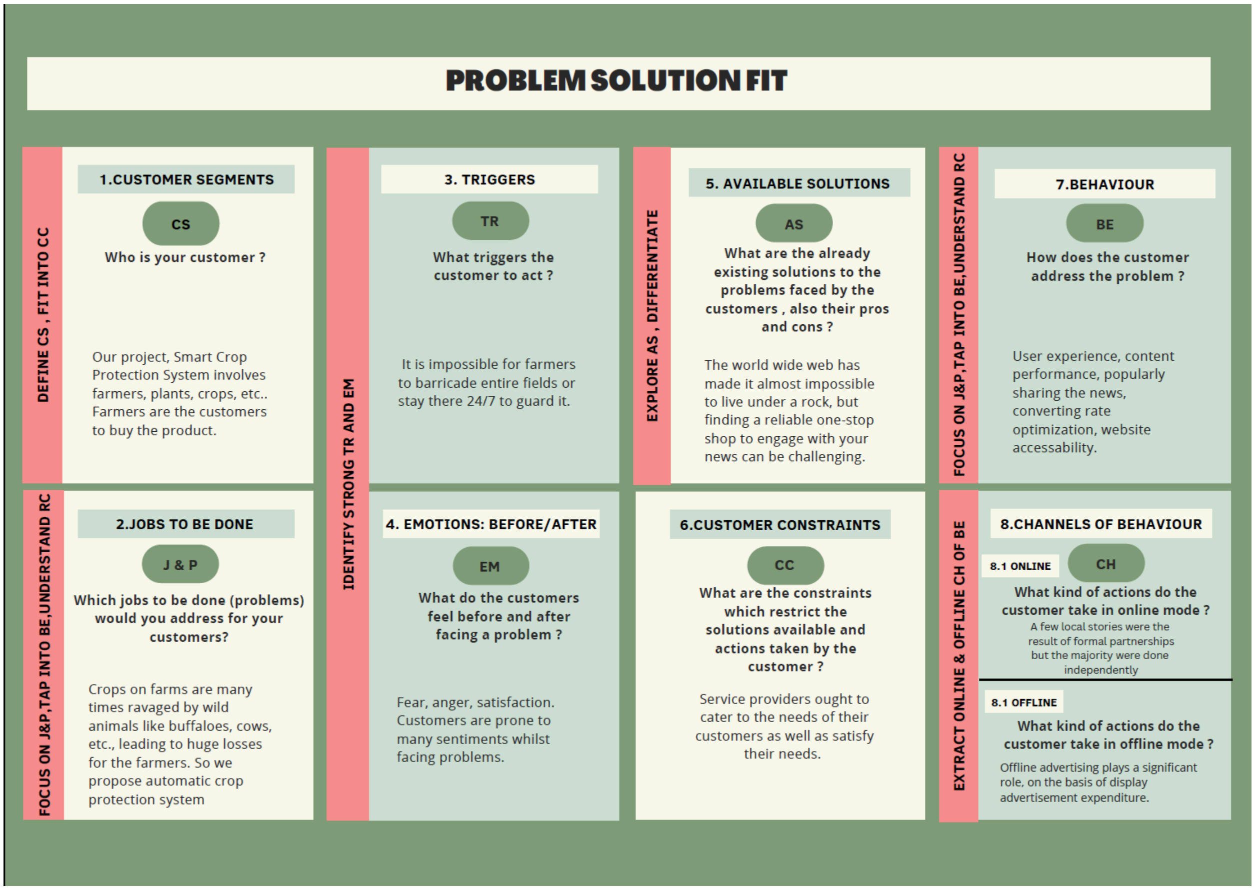


3.3 Proposed Solution

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The crop protection system helps the farmers in protecting the crop from animals,birds and pest which destroys the crop.
2.	Idea / Solution description	This project is based on surveillance with an animal ward-off system employed in farmlands in order to prevent crop vandalization by wild animals. In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID's, various PIR sensors are deployed in the area to detect any motion and hence turns ON a camera when movement is detected, thereby providing real time monitoring.
3.	Novelty / Uniqueness	Agriculture is the backbone of the economy but,because of the animals,birds and pest interference in the agricultural lands,there will be huge loss for the crops.
4.	Social Impact / Customer Satisfaction	With the use of Sound Trap,we can deliver environmental benefits in the form of safe,Healthy and affordable food.By using sound trap will probably maintain and even increase the yield.
5.	Business Model (Revenue Model)	We advocate for establishment of crop protection adequacy standards that would allow a market system to maximize efficiency.
6.	Scalability of the Solution	*This can be developed to a scalable product by using sensors and transmitting the data and analysing the data in cloud.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

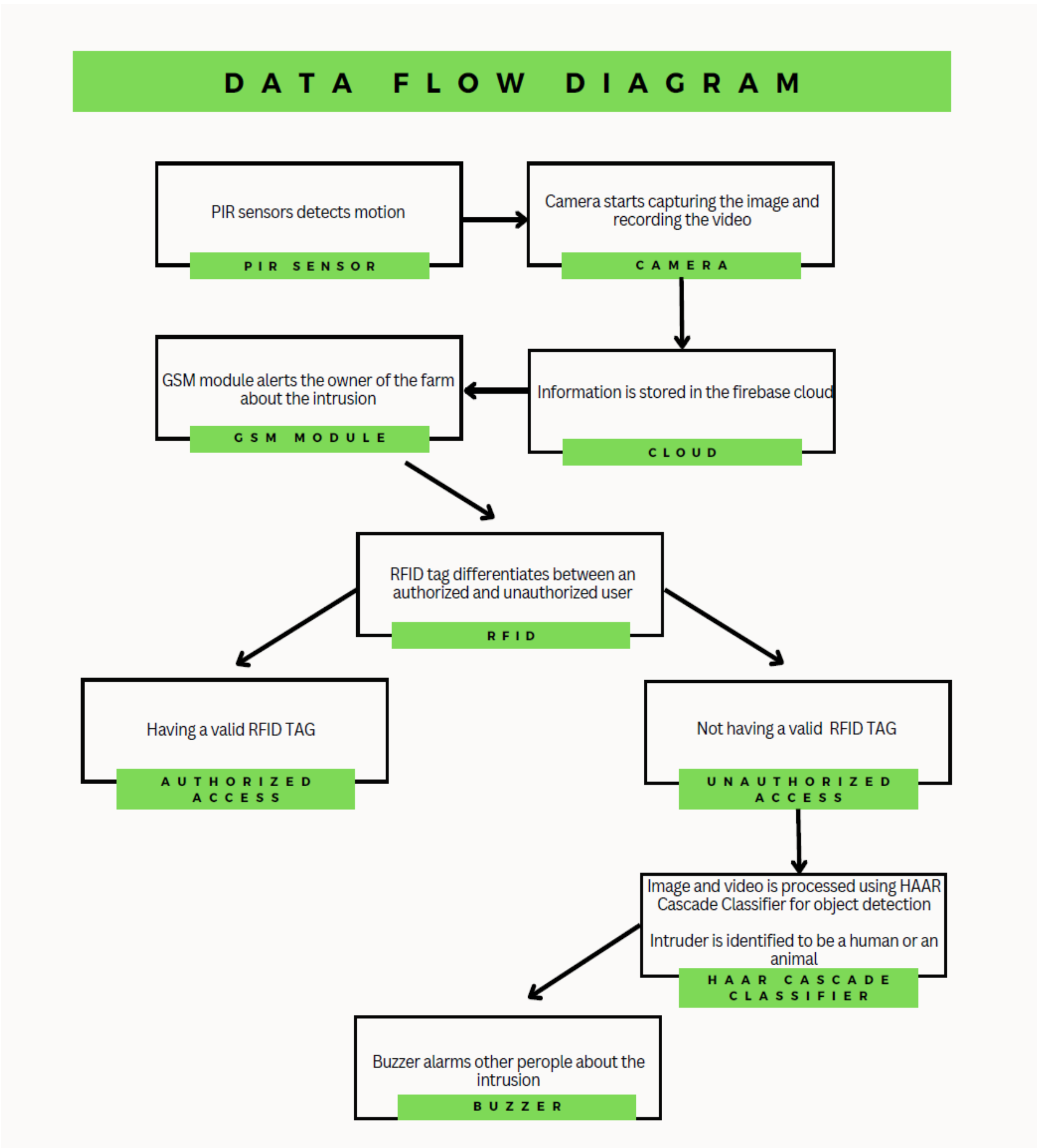
S.NO.	FUNCTIONAL REQUIREMENT	SUB-REQUIREMENT
1.	User Visibility	Sense animals nearing the crop field and sound an alarm to scare them away as well as send SMS to farmers using cloud service.
2.	User Reception	The data like values of temperature, humidity, and soil moisture sensors are received via SMS.
3.	User Understanding	Based on the sensor data value to get the information on the presence of farming land.
4.	User Action	The user needs to take actions like destruction of crop residues, deep plowing, crop rotation, fertilizers, etc..

4.2 Non-Functional requirements

S.NO.	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
1.	Usability	Mobile support users must be able to interact in the same roles & tasks on computers and devices where practical, given mobile capabilities.
2.	Security	Data requires secure access to register and communicate securely on devices and authorized users of the system who exchange information must be able to do so.
3.	Reliability	It has the capacity to recognize the disturbance near the field and does not give a false caution signal.
4.	Performance	Must provide acceptable response time to users regardless of the volume of data that is stored and the analytics that occurs in the background.
5.	Availability	IoT solutions and domains demand highly available systems for 24x7 operations isn't a critical production application.
6.	Scalability	System must handle expanding load and data retention needs that are based on the upscaling of the solution scope.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form, Gmail, LinkedIn
FR-2	User Confirmation	Confirmation via Email, OTP
FR-3	User Understanding	For better understanding of the farmer, the signs should be clear and legible. It can include illustrations to make it easily understandable for the farmers
FR-4	User Convenience	The display should be big enough that it should be visible clearly even from a far distance

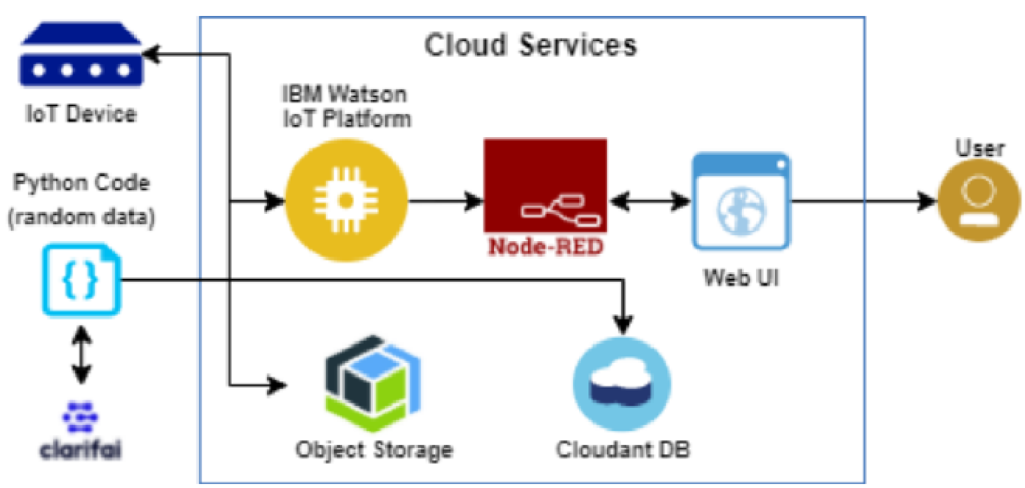
Non-functional Requirements:

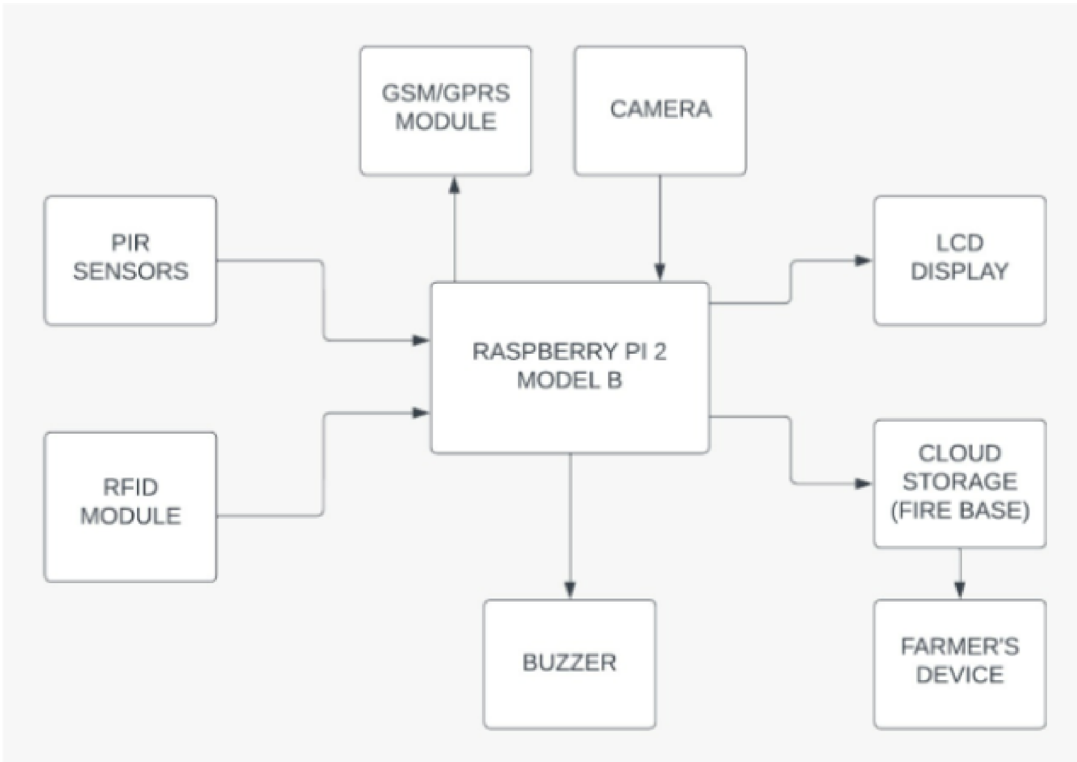
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It should have a good security system so no other person is able to hack and display their own directions.
NFR-2	**Security**	It should able to display information without any errors.
NFR-3	**Reliability**	It should have the ability to automatically update itself when certain weather problems arise.
NFR-4	**Performance**	It should be available 24/7 for the benefit of the customers.
NFR-5	**Availability**	It should have the ability to change and upgrade easily according to the requirements.
NFR-6	**Scalability**	It should be able to upgrade and update when there is a need for it.

Technical Architecture:

The architectural diagram is shown below and the technology used is listed down in table 1 and 2





Block Diagram

Description: This project is based on surveillance with an animal ward-off system employed in farmlands in order to prevent crop vandalization by wild animals. In addition to providing protection, this system distinguishes between an intruder and an authorised person using RFID, various PIR sensors are deployed in the area to detect any motion and hence turn ON a camera when movement is detected, thereby providing real-time monitoring.

Table-1:Components & Technologies:

S.No	Components	Description	Technology
1.	Raspberry Pi 2 Model B	The Raspberry Pi is an open-source microcontroller.	C++/ Python
2.	Application Logic-1	Used to capture pictures and videos	Camera
3.	Application Logic-2	After detecting the presence of movement, it emits sound	Buzzer/ Alarm

4.	Cloud Database	Database server on cloud	IBM Watson IoT platform or Firebase
5.	External API - 1	Purpose of external API used in the application to locate the crops	Google maps Geolocation API
6.	External API - 2	A GSM/GPRS Module is used to enable communication between a microcontroller (or a microprocessor) which sends SMS to the farmers device	GSM
7.	External API - 3	To sensors utilise the detection of infrared that is radiated from all objects that emit heat.	PIR sensor
8.	External API - 4	This technology is used to record the presence of an object using radio signals.	RFID sensor module
9.	Machine Learning Model	To displays some basic and vital information	LCD Display
10.	User interface	Mobile App	IoT

Table 2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	SDK that is freely available for the users to use the source code	Raspberry pi 2 Model B
2.	Scalable Architecture	By utilising the sensors, it supports higher workload to work with the IoT	IoT
3.	Security Implementations	The user can alone view the issued data	Authorization and Authentication accessibility
4.	Availability	Web features can be accessed from any remote location	IBM Watson IoT platform
5.	Performance	As the sensors provide accurate data, it is effective in its performance	Python

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	Preethi
Sprint-1	Confirmation	USN-2	As a user, I will receive a confirmation email once I have registered for the application	1	High	Paarvathi
Sprint-2	Confirmation via Facebook	USN-3	As a user, I can register for the application through Facebook	2	Low	Prasanthi
Sprint-1	Confirmation via Gmail	USN-4	As a user, I can register for the application through Gmail	2	Medium	Radha Prabhakaran
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Radha Prabhakaran

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	49	06 Oct 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	07 Oct 2022

7. CODING & SOLUTION

7.1 Feature 1

Output: Digital pulse high (3V) when triggered (motion detected) and digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor. Power supply: 5V-12V input voltage for most modules (they have a 3.3V regulator), but 5V is ideal in case the regulator has different specs.

BUZZER

Specifications

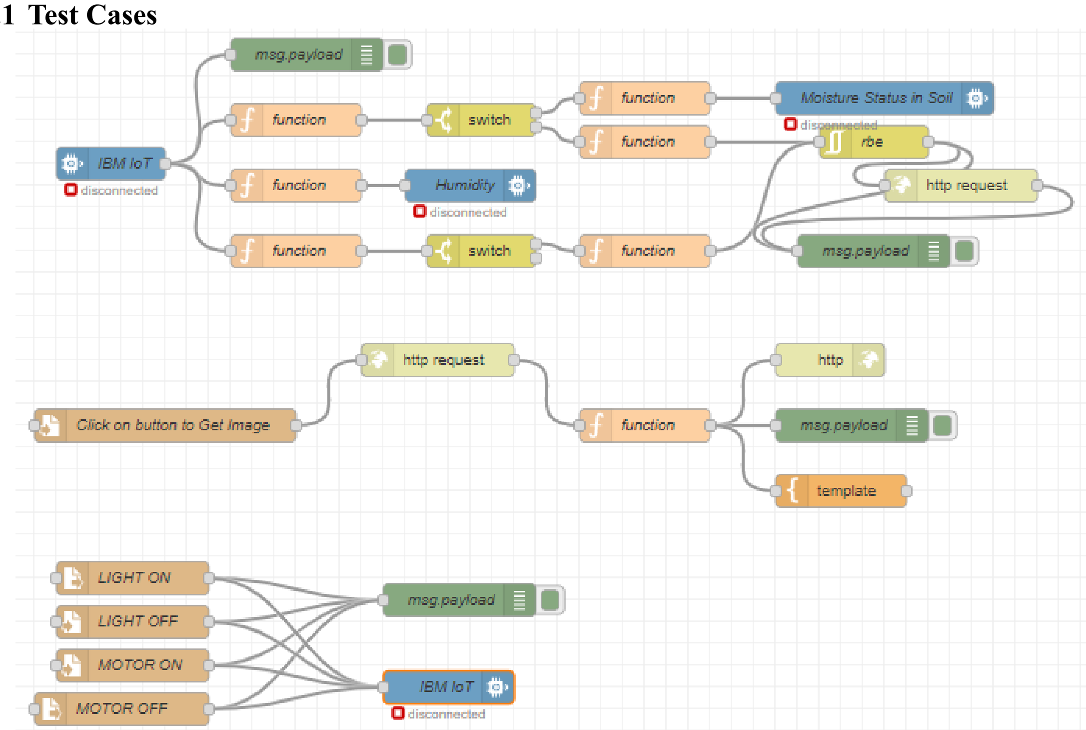
- Rated Voltage : 6V DC
- Operating Voltage : 4 to 8V DC
- Rated Current*: ≤30mA
- Sound Output at 10 cm* : ≥85dB
- Resonant Frequency : 2300 ±300Hz

7.2 Feature 2

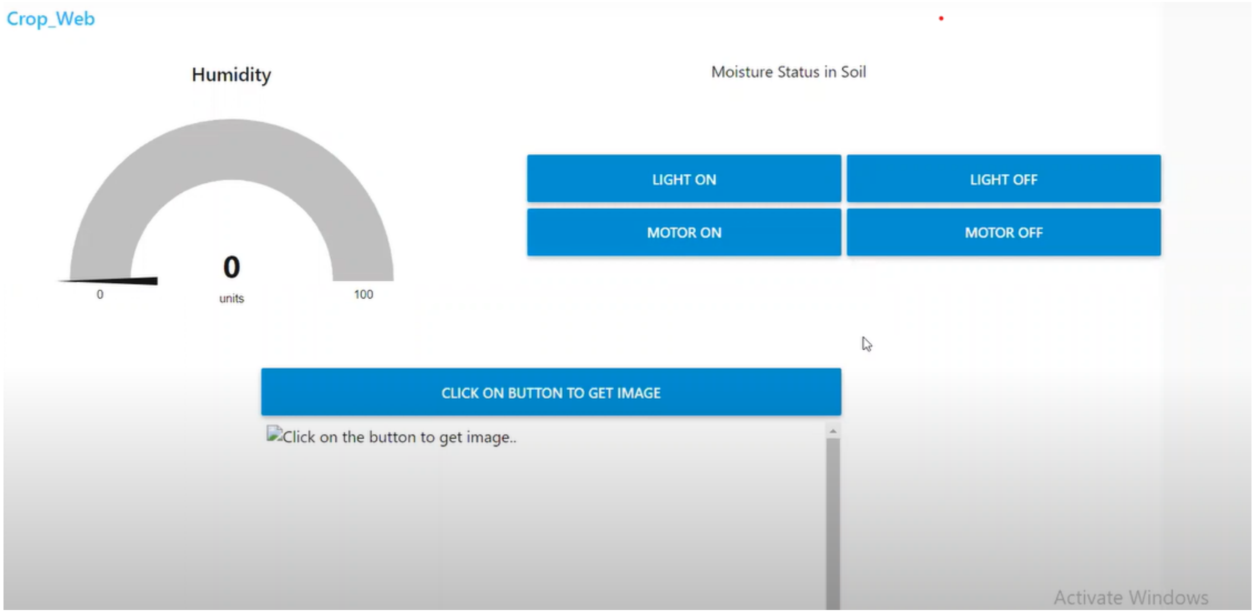
i. Good sensitivity to Combustible gas in a wide range.
ii. High-sensitivity to LPG, Propane and Hydrogen.
iii. Long-life and low cost.
iv. Simple drive circuit.

8. TESTING

8.1 Test Cases



8.2 User Acceptance Testing



9. RESULTS

9.1 Performance Metrics

The problem of crop vandalization by wild animals and fire has become a major social problem in current times. It requires urgent attention, as no effective solution exists to date for this concern. Thus, this project carries great social relevance as it aims to address this issue. This project will help farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection of their fields. This will also help them in achieving better crop yields, thus leading to economic well-being.

10. ADVANTAGES & DISADVANTAGES

Controllable food supply. You might have droughts or floods, but if you are growing the crops and breeding them to be hardier, you have a better chance of not starving. It allows farmers to maximize yields using minimum resources such as water, and fertilizers.

The main disadvantage is the time it can take to process the information.in order to keep feeding people as the population grows, you have to radically change the environment of the planet.

11. CONCLUSION

An IoT Web Application is built for smart agricultural systems using Watson IoT platform, Watson simulator, IBM cloud and Node-RED.

12. FUTURE SCOPE

In the future, there will be a very large scope, this project can be made based on Image processing in which wild animals and fire can be detected by cameras and if it comes towards the farm then the system will be directly activated through wireless networks. Wild animals can also be detected by using wireless networks such as laser wireless sensors and by sensing this laser or sensor's security system will be activated.

13. APPENDIX

13.1 Source Code

```
import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
import sys
#IBM Watson Device Credentials.
Device ID="12345"
Device Type="NodeMCU"
Date Added="Nov 19, 2022 1:39 PM"
Added By="aaradha245@gmail.com"
Connection Status="Disconnected"
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()
```

13.2 GitHub & Project Demo Link

https://drive.google.com/file/d/1cm0DmsT2GE1IiTVZNZIkSNJXjwgVFNf7/view