Team ID:PNT2022MID32605

IOT based Smart Crop Protection System for Agriculture

Team members

Ahmed Yahya A – 221005

Dheeraj Prakash S – 221025

Mukesh Manikandan M – 221056

Nagaraj R – 221059

INDEX

1. **INTRODUCTION**
   - 1.1. Project Overview
   - 1.2. Purpose

2. **LITERATURE SURVEY**
   - 2.1. Existing problem
   - 2.2. References
   - 2.3. Problem Statement Definition

3. **IDEATION & PROPOSED SOLUTION**
   - 3.1. Empathy Map Canvas
   - 3.2. Ideation & Brainstorming
   - 3.3. Proposed Solution
   - 3.4. Problem Solution fit

4. **REQUIREMENT ANALYSIS**
   - 4.1. Functional requirement
   - 4.2. Non-Functional requirements

5. **PROJECT DESIGN**
   - 5.1. Data Flow Diagrams
   - 5.2. Solution & Technical Architecture
   - 5.3. User Stories

6. **PROJECT PLANNING & SCHEDULING**
   - 6.1. Sprint Planning & Estimation
   - 6.2. Sprint Delivery Schedule
   - 6.3. Reports

7. **CODING & SOLUTIONING**
   - 7.1. Feature 1
   - 7.2. Feature 2
   - 7.3. Feature 3

8. TESTING
- **8.1. Test Cases**
- **8.2. User Acceptance Testing**

9. RESULTS
- **9.1 Performance Metrics**

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX
- **Source Code**
- **GitHub & Project Demo Link**

Project Report

1.Introduction:

1.1 Project Overview:

Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds etc. This Leads to huge losses for the farmers. It is not possible for farmers to barricade entire fields or stay on field on 24 hours and guard it. So here we propose smart crop protection system which prevents from animals and this project detects animal detection by sending alert to the farmers.

1.2 Purpose:

Animal intrusion are common thing happening now-a-days. Since it causes destruction to the farms as well as it destroys the livelihood of the farmers also. Our project ensures that the animal intrusion will gets avoided as well as the temperature and humidity of the crops will be monitored in a continuous manner.

2.Literature Survey:

2.1 Existing problem:

Animal intrusion and crop maintenance are two major issues for farmers especially who lives in rural parts of India. Farmers didn't have awareness to tackle all these stuffs and didn't have the knowledge of current technologies also and cost of the application which is existing now was not at nominal cost.

2.2 References:

1) Krunal Mahajan, Riya Prate, Ekta Zade, Shubham Khanate, Shishir Bagal5," REVIEW PAPER ON SMART CROP PROTECTION SYSTEM", International Research Journal of Engineering and Technology (IRJET), Volume: 08, issue 02 Feb 2021

2) Anjana, Sowmya, Charon Kumar, Manisha, Sahana, "Review on IoT in Agricultural Crop Protection and Power Generation", International Research Journal of Engineering and Technology (IRJET) , Volume 06, Issue 11 ,Nov 2019.

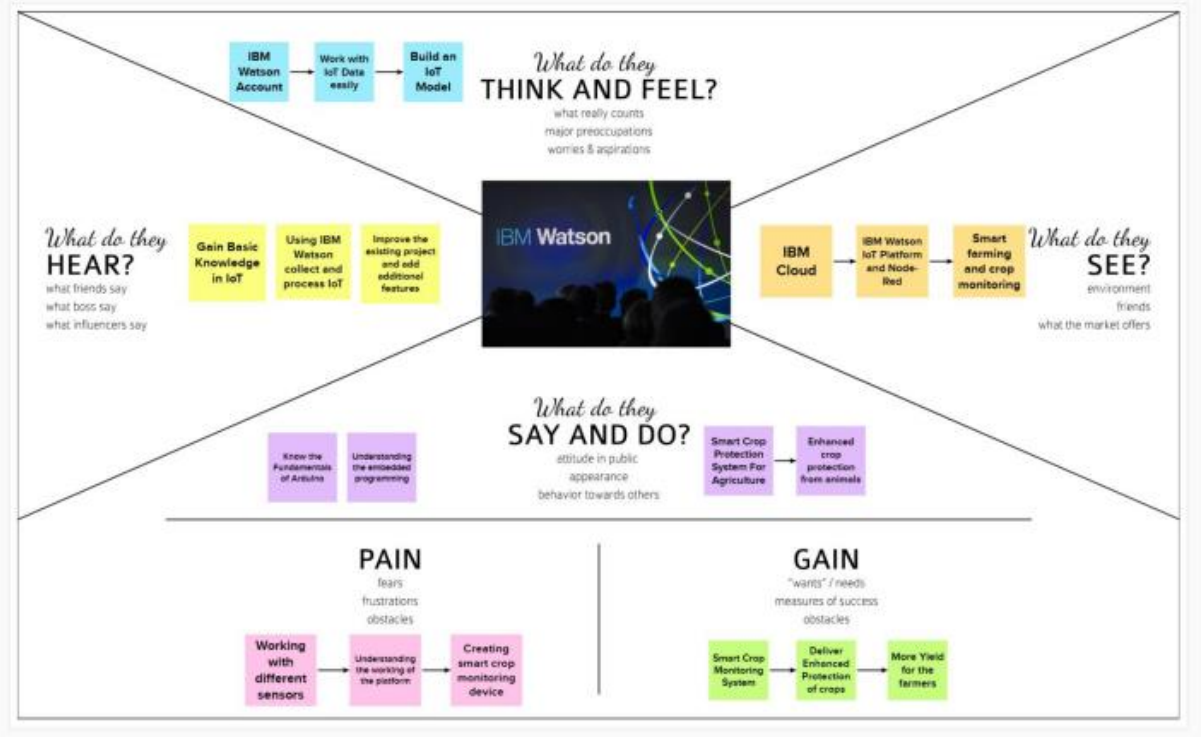
3) G. NaveenBalaji, V. Nandhini, S. Mithra, N.Priya, R. Novena, "IOT based Smart Crop monitoring in farmland", Imperial Journal of Interdisciplinary Research (IJIR), Volume 04,Issue 01,Nov 2018

2.3 Problem Statement Definition:

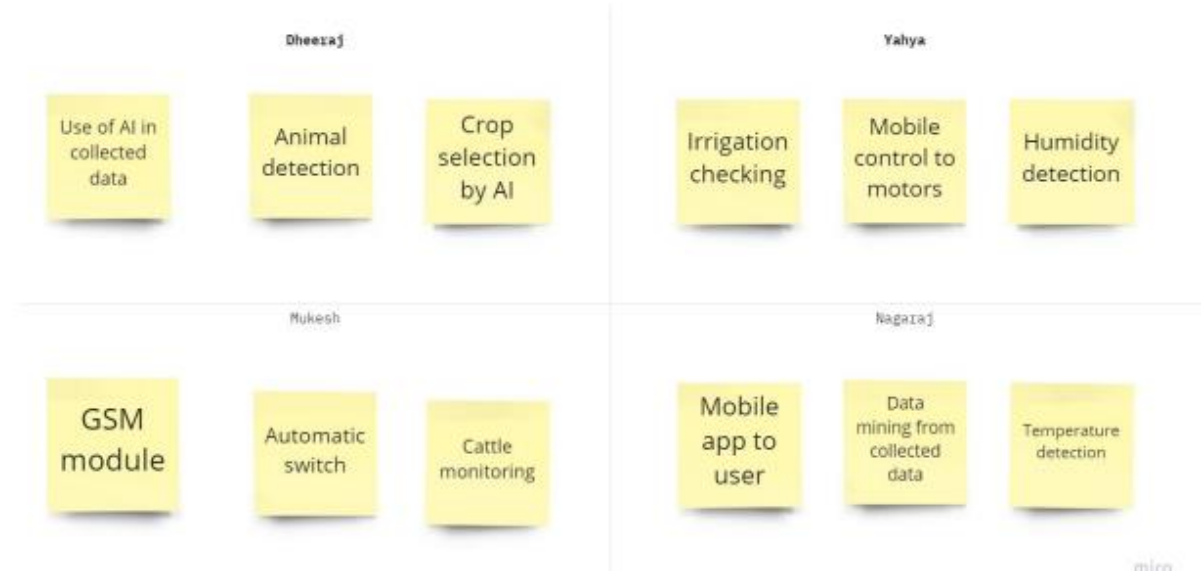
An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop. This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field. The motors and sprinklers in the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

3.Ideation and Proposed Solution:

3.1 Empathy Map Canvas:



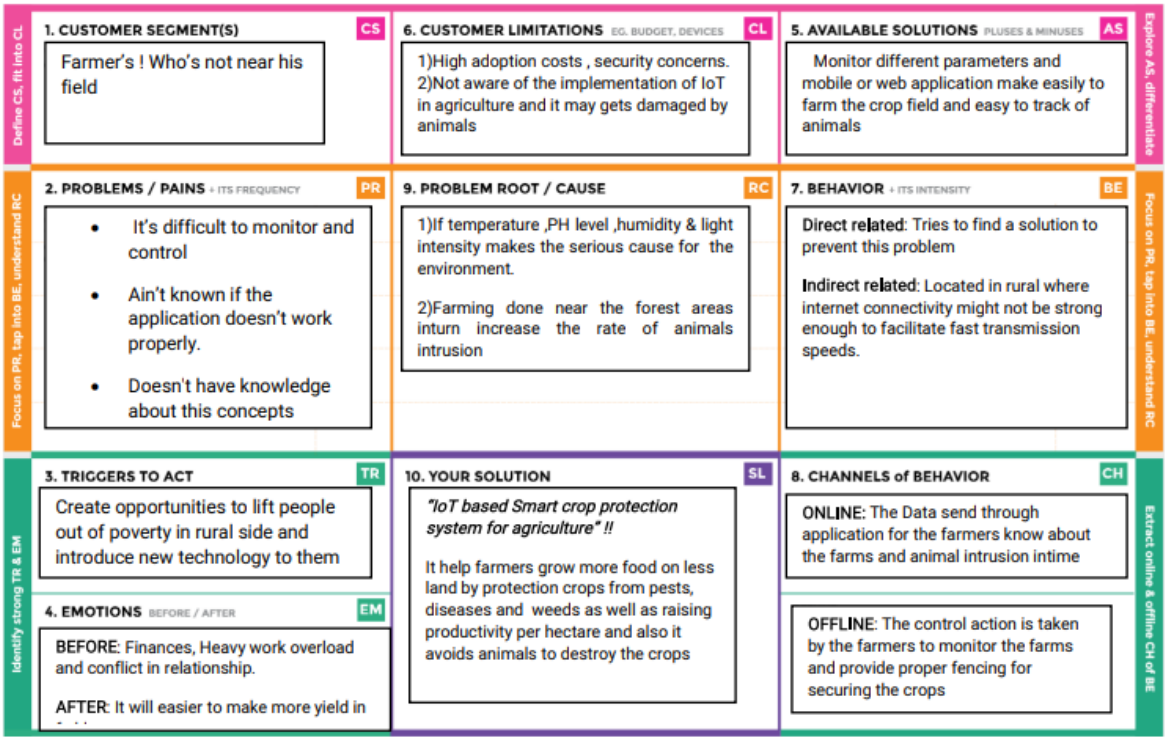
3.2 Ideation and Brainstorming:



3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Iot based Smart Crop Protection System for Agriculuture
2.	Idea / Solution description	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
3.	Novelty / Uniqueness	The complete safety of crops is ensured from animals without harming animals and with minimum loss to the farm
4.	Social Impact / Customer Satisfaction	To tackle the effect of climate change, water scarcity and to prevent crop from animals for better yields
5.	Business Model (Revenue Model)	Iot business models include platform,subscription,pay-per-usage and data driven models
6.	Scalability of the Solution	It supports increasing number of connected devices and users as well as application features

3.4 Problem Solution fit:



4 Requirement Analysis:

4.1 Functional Requirement:

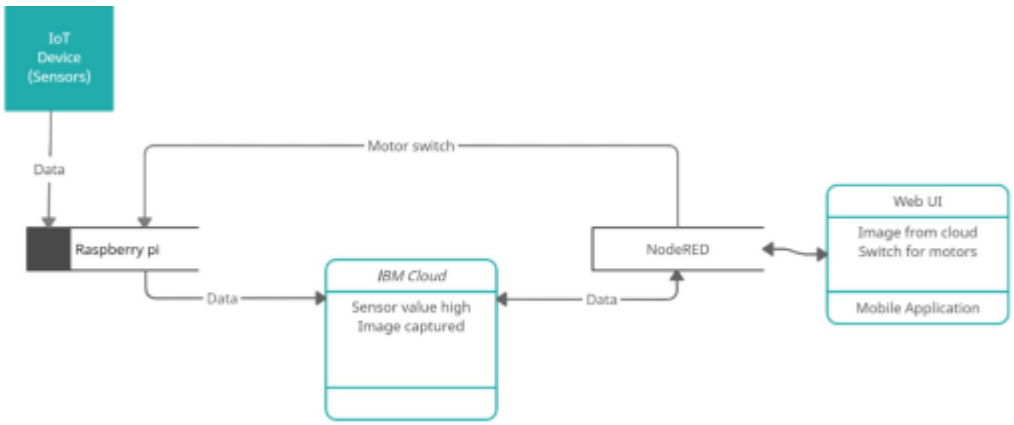
FRNo.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Motion Detection	Detection of animal movement in the farm land
FR-2	Temperature	Monitoring the ambient temperature and alerting
FR-3	Humidity	Monitoring the ambient humidity and alerting
FR-4	Controlling the Electrical Appliances	Switching the light ON and OFF Switching the Motor ON and OFF
FR-5	SMS	Alerting the user about the above instances

4.2 Non-Functional Requirement:

FRNo.	Non-Functional Requirement	Description
NFR-1	**Usability**	The Project is easy to use and implement. It is more user friendly.
NFR-2	**Security**	It is Based on IBM cloud hence the security will be high by default
NFR-3	**Reliability**	As we use Computer vision the reliability is very high
NFR-4	**Performance**	It can withstand any kind of environment as it is based on crop protection
NFR-5	**Availability**	It is always available since the sensors will sense the data and transmit to cloud frequently for processing.
NFR-6	**Scalability**	It is scalable to a great extent since it uses IBM Cloud and Node Red.

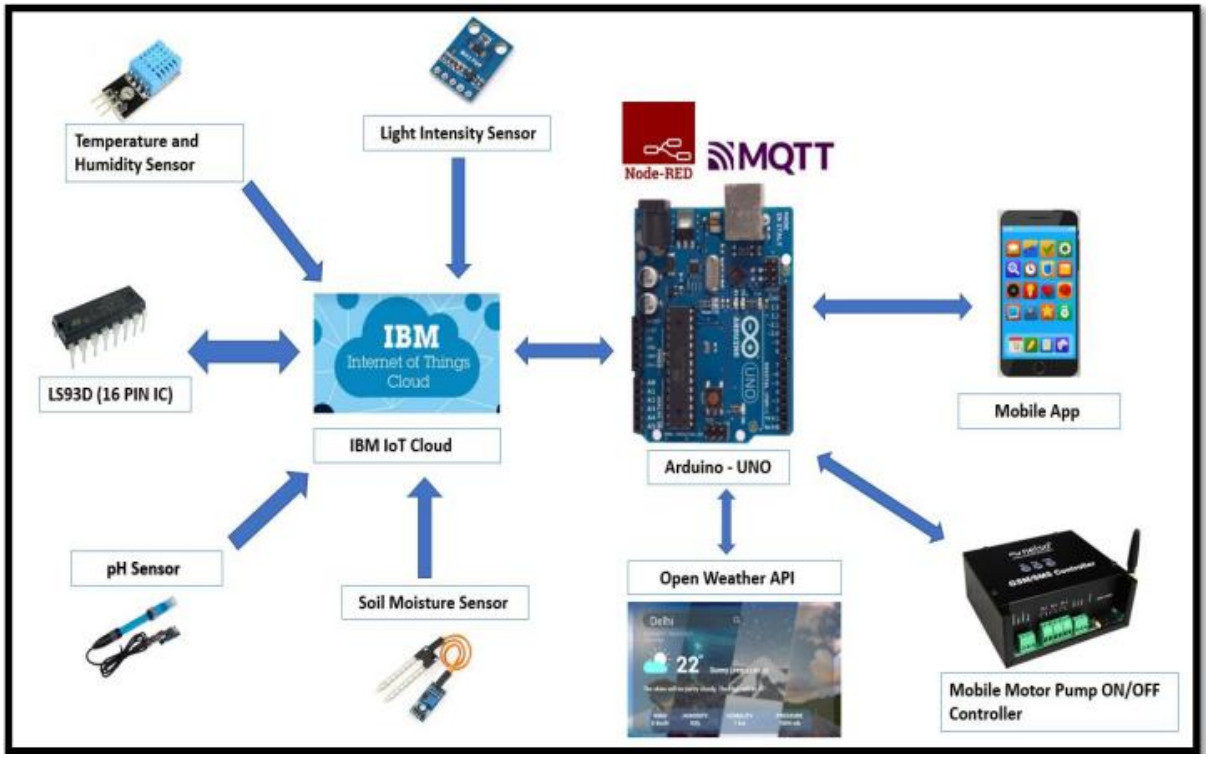
5.Project Design:

5.1 Data Flow Diagram:

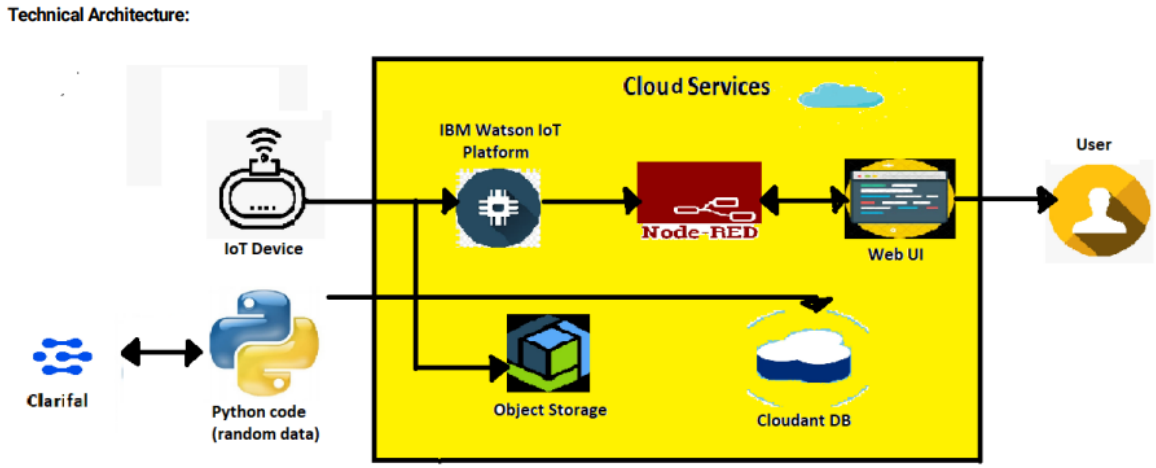


5.2 Solution and Technical Architecture:

Solution Architecture:



Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Farmer	Registration and setup	USN1	As a user - farmer, I can register for the system by setting up the sensors	-	High	Sprint-1
Farmer	Get notify	USN2	When an animal is detected I need to receive notification with Image captured	I can view the data in the web	High	Sprint-1
Farmer	On/off Motor	USN3	I can remotely turn on and off the water motor	UI for turning the water motor	Medium	Sprint-2
Farmer	On/off Light	USN4	I can remotely turn on and off the lights	UI for turning the Light	Medium	Sprint-2
Farmer	Temperature and Humidity	USN5	I need to view the current temperature and the humidity of the agriculture land	Web UI for viewing the data	Low	Sprint-2

6. Project Planning and Scheduling:

6.1 Sprint Planning and Estimation:

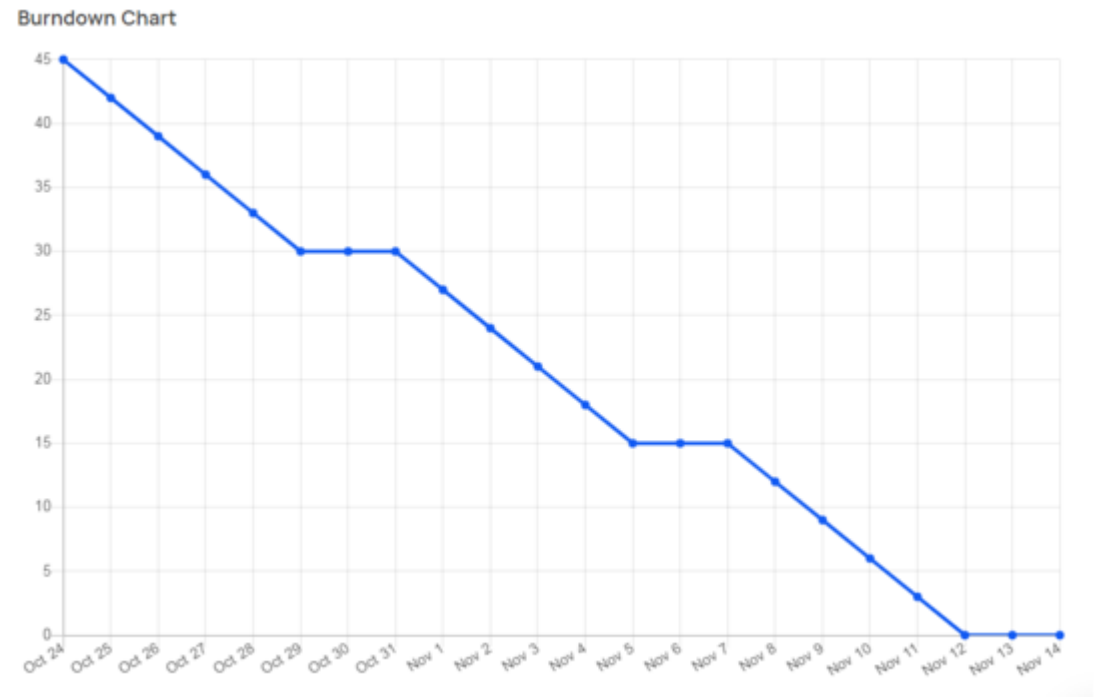
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration and setup	Farmer	As a user-farmer , I can register for the system by setting up the sensors.	20	High	Ahmed Yahya.A, Dheeraj Prakash.S, Mukesh Manikandan.M, Nagaraj.R
Sprint-2	Get notify	Farmer	When an animal is detected I need to receive notification with Image captured.	10	High	Ahmed Yahya.A, Dheeraj Prakash.S, Mukesh Manikandan.M, Nagaraj.R
Sprint-3	On/off Motor	Farmer	I can remotely turn on and off the water motor.	5	Medium	Ahmed Yahya.A, Dheeraj Prakash.S, Mukesh Manikandan.M, Nagaraj.R
Sprint-3	On/off Light	Farmer	I can remotely turn on and off the lights.	5	Medium	Ahmed Yahya.A, Dheeraj Prakash.S, Mukesh Manikandan.M, Nagaraj.R
Sprint-4	Temperature and Humidity	Farmer	I need to view the current temperature and the humidity of the agriculture land.	5	Low	Ahmed Yahya.A, Dheeraj Prakash.S, Mukesh Manikandan.M, Nagaraj.R

Estimation : 15,000 rupees including raspberry pi, sensor and cameras

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	01 Nov 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	08 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	14 Nov 2022
Sprint-4	5	6 Days	14 Nov 2022	19 Nov 2022	5	19 Nov 2022

6.3 Burndown Chart



7.Coding & Solutioning:

7.1 Feature 1:

```
detect = False
   img_url = None
   for concept in response.outputs[0].data.concepts:
      if (concept.name == "animal"):
         if(concept.value > 0.85):
            print("Animal is Detected")
            #playsound.playsound('alert.mp3')
            pic = datetime.datetime.now().strftime("%y-%m-%d-%H-%M-%S")
            img_url="https://dymn-crop-protection.s3.jp-tok.cloud-object-storage.appdomain.cloud/"+pic+".jpg"
            cv2.imwrite(pic + '.jpg',frame)
            multi_part_upload('dymn-crop-protection',pic+'.jpg',pic+'.jpg')
            json_document = {"link":COS_ENDPOINT+'/'+'dymn-crop-protection'+'/'+pic+'.jpg'}
            new_document = my_database.create_document(json_document)
            if new_document.exists():
               print(f"Document successfully created")
            detect=True
            time.sleep(5)
            break
   else:
      print("No Animal Detected")
```

Feature 1:

Image is captured continuously with the help of camera and checked simultaneously with the Image Classification Algorithm which is provided by Clarifai Service. In case of any animal crosses over the frame it is detected and the image is stored in the Object Storage of IBM Cloud and a copy of stored image is sent to NodeRed and MIT App which serves as the Frontend interface for the user.

7.2 Features 2:

```
#Temperature Data from sensor
    temp = random.randint(0, 100)
    #Humidity Data from sensor
    humidity = random.randint(0, 100)
    myData = {'detect': detect, 'temperature': temp, 'humidity': humidity, 'url':img_url}
    print("Data : ",myData)
    #Send to IBM Watson
    if ((temp != None)  and (humidity != None)):
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Published")
    #Commands from IBM
    client.commandCallback = myCommandCallback
    #Wait (in Seconds)
    time.sleep(45)
cap.release()
cv2.destroyAllWindows()
#Disconnect device
client.disconnect()
```

Feature 2:

Temperature and humidity can be recorded with the sensors and sent to the IBM Watson. This ensures that insects that attack crops will be reduced and the crop yields will be monitored to provide at the desired time. It also ensures that the right kind of crop is cultivated at the right time.

7.3 Features 3:

```
#Recieve Commands
def myCommandCallback(cmd):
    print("\nCommand Recieved")
    command = cmd.data['command']
    if(command == 'Lon'):
        print('Light is on')
    elif(command == 'Loff'):
        print('Light is off')
    elif (command == 'Mon'):
        print('Motor started running')
    elif (command == 'Moff'):
        print('Motor stopped')
    print()
```

Feature 3:

The motor pumps that are connected in the farm and the lights are made to be controllable by the respected farmer from any where using the Android app or the website. When the Farmer clicks the Motor On button, a command message is sent from Node Red to the IBM Watson device which is in turn connected by the python script and command is received and executed.

8.Testing:

8.1 Test Cases:

Test ID	Test Scenario	Test Data	Expected Result	Status
NodeRED_TC_001	Verify user is able to see the Login	https://cropprotection.eu-gb.cf.appdomain.cloud/ui/	Login page should display and redirect to home page after authentication	Pass
NodeRED_TC_002	Check if data receives from sensor	https://cropprotection.eu-gb.cf.appdomain.cloud/ui/	Data received is displayed in UI	Pass
NodeRED_TC_003	Check if alert box is triggered if animal is detected	Sample video containing animal image	An alert dialog box is obtained	Pass
NodeRED_TC_004	When 'View Image' button is clicked animal image should be shown	Sample video containing animal image	Animal Image is obtained	Pass
NodeRED_TC_005	Verify user is able to move to control Tab	Button is Clicked	Webpage should redirect to Control Tab	Pass
NodeRED_TC_006	Verify user is able to control motor	Button is Clicked	Command should be received in python to control motor	Pass
NodeRED_TC_007	Verify user is able to control Light	Button is Clicked	Command should be received in python to control light	Pass
NodeRED_TC_008	Verify user is able to move to Home Tab	Button is Clicked	Webpage should redirect to Home Tab	Pass
Python_TC_001	Image captured by camera should be recognized as animal or not	Sample video containing animal image	Animal Detected	Pass
Python_TC_002	Image captured by camera should be recognized as animal or not	Sample video without animal image	Animal not Detected	Pass
Python_TC_003	When button clicked On/Off either in Web/App command should receive in python	Command given in Website and Android app	Command received	Pass
Python_TC_004	When button clicked On/Off either in Web/App command should receive in python	Command given in Website and Android app	Command received	Pass

Python_TC_005	When Sensor data is received it should be sent to the Node-Red	Data received in IBM Watson device	Event data received	Pass
Python_TC_006	If Animal is detected it should be stored in the Object Storage	Sample video without animal image	Image stored in the cloud	Pass
App_TC_001	Check if data receives from sensor	Random Integer sent from Python	Data received is displayed in UI	Pass
App_TC_002	Check if alert box is triggered if animal is detected	Sample video containing animal image	An alert dialog box is obtained	Pass
App_TC_003	When 'View Image' button is clicked animal image should be shown	Sample video containing animal image	Animal Image is obtained	Pass
App_TC_004	Verify user is able to control motor	Button is Clicked	Command should be received in python to control motor	Pass
App_TC_005	Verify user is able to control Light	Button is Clicked	Command should be received in python to control light	Pass
App_TC_006	If Animal is detected, Alert notification should be pushed	Sample video containing animal image	Notification is raised	Pass

8.2 User Acceptance Testing:

Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	4	1	3	16
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	2	15	30
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	22	12	10	21	65

Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Node red service	5	0	0	5
Client Application	15	0	0	15
Security	2	0	0	2
Python script for animal detection	6	0	0	6
Exception Reporting	1	0	0	1
Final Report Output	4	0	0	4
Version Control	3	0	0	3

9 Results:

9.1 Performance Metrics:

Average Response Time: 45 seconds

The Accuracy of the project: 90%

Error Rates: 5%

Application Availability is ensured

10.Advantades & Disadvantages:

Advantages:

- It allows farmers to maximize yields using minimum resources such as water, fertilizers, seeds etc.
- It is cost effective method.
- It delivers high quality crop production.

Disadvantages:

- The smart agriculture needs availability of internet continuously.
- Rural part of most of the developing countries do not fulfil this requirement.
- The smart farming-based equipment’s require farmers to understand and learn the use of technology.

11.Conclusion:

This application is well suited for all kinds of farm environment by detecting the animal intrusion and reporting to the user. It also sends the data about the current Temperature and Humidity along with the forecasted weather at that particular place. The app is also able to control the Motor and Lighting facility from the remote place.

12.Future scope:

1. Motor functionality will be automated with auto cut-off facility when the water overflows.

2. Human detection by classifying the apart from authorized persons will be enabled.

13.Appendix:

Source code:

```
#packages
import cv2
import wiotp.sdk.device
import random
import time
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
from clarifai_grpc.grpc.api import service_pb2, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
#Clarifai
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
metadata = (("authorization", f"Key a3dbb3c91d514700b2ed37f0c5fdd4ec"),)
#CloudObjectStorage Credentials
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "a4K_AtOCafJlAW1lCqcFZWcVDspWB4DEK4C2MgzRRgf4"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN="crn:v1:bluemix:public:cloud-
objectstorage:global:a/1bdd4ed6ad914c8a852296af0c126df3:66d3a14a-26b2-4885-a1ac-
b137784a8487:bucket:dymn-crop-protection"
#Cloudant
clientdb=Cloudant("apikeyv2pziyonwwgbpyhhc5fm4pxkm8969a33a9yjmqiacw5r1","ea31c9
92e248c8fc9a285fd1291564c4",url="https://apikey-v2-
pziyonwwgbpyhhc5fm4pxkm8969a33a9yjmqiacw5r1:ea31c992e248c8fc9a285fd1291564c4
@2c780937-91bf-4dd9-94fa-fe2ae17340c9-bluemix.cloudantnosqldb.appdomain.cloud")
clientdb.connect()
```

```
#Create Resource
cos = ibm_boto3.resource("s3",ibm_api_key_id = COS_API_KEY_ID,
ibm_service_instance_id=COS_RESOURCE_CRN,config=Config(signature_version="oaut),
 endpoint_url = COS_ENDPOINT )
#Upload Image to Bucket
def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name,bucket_name))
        part_size = 1024 * 1024 * 5  #5MB
        file_threshold = 1024 *1024 * 15 #15MB
        transfer_config=ibm_boto3.s3.transfer.TransferConfig(multipart_threshold=
            file_threshold , multipart_chunksize = part_size )
        with open(file_path,"rb") as file_data:
            cos.Object(bucket_name,item_name).upload_fileobj
            (Fileobj=file_data,Config=transfer_config )
            print("Transfer for {0} Complete !!\n".format(item_name))
    except ClientError as be:
        print("Client Error : {0}\n",format(be))
    except Exception as e:
        print("Unable to Complete multi-part upload : {0}".format(e))
#Recieve Commands
def myCommandCallback(cmd):
    print("\nCommand Recieved")
    command = cmd.data['command']
    if(command == 'Lon'):
        print('Light is on')
    elif(command == 'Loff'):
        print('Light is off')
    elif (command == 'Mon'):
        print('Motor started running')
    elif (command == 'Moff'):
        print('Motor stopped')
```

```
#Device Config
myConfig = {
    "identity" : {
        "orgId":"yqzb6k",
        "typeId":"IoTDevice",
        "deviceId":"cropdymn"
    },
    "auth":{
        "token":"cropprotection"
    }
}

#Watson Device Connection
client = wiotp.sdk.device.DeviceClient(config = myConfig, logHandlers=None)
client.connect()

#Database Creation
database_name = "sample1"
my_database = clientdb.create_database(database_name)

#Camera
#cap = cv2.VideoCapture(0)

#Sample Video
cap = cv2.VideoCapture('sample.mp4')

#Animal Classifier
if(not(cap.isOpened() == True)):
    print('File Not Found')
```

```
#Code
while(cap.isOpened()):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    imS = cv2.resize(frame,(960,540))
    cv2.imwrite('temp.jpg',imS)
    with open('temp.jpg',"rb") as f:
        file_bytes = f.read()
    #Clarifai Image Classifier Model
    request = service_pb2.PostModelOutputsRequest(model_id='aaa03c23b3724a16a56b629203edc62c' , inputs = [resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes)))])
    response = stub.PostModelOutputs(request,metadata=metadata)
    if response.status.code != status_code_pb2.SUCCESS:
        raise Exception("Request failed code: " + str(response.status.code))


    #Checking for Animals
    detect = False
    img_url = None
    for concept in response.outputs[0].data.concepts:
        if (concept.name == "animal"):
            if(concept.value > 0.85):
                print("Animal is Detected")
                #playsound.playsound('alert.mp3')
                pic = datetime.datetime.now().strftime("%y-%m-%d-%H-%M-%S")
                img_url="https://dymn-crop-protection.s3.jp-tok.cloud-object-storage.appdomain.cloud/"+pic+".jpg"
                cv2.imwrite(pic + '.jpg',frame)
                multi_part_upload('dymn-crop-protection',pic+'.jpg',pic+'.jpg')
                json_document={"link":COS_ENDPOINT+'/'+'dymn-crop-protection'+'/'+pic+'.jpg'}
                new_document = my_database.create_document(json_document)
```

```
            if new_document.exists():
                print(f"Document successfully created")
            detect=True
            time.sleep(5)
            break
    else:
        print("No Animal Detected")
    #Temperature Data from sensor
    temp = random.randint(15, 50)
    #Humidity Data from sensor
    humidity = random.randint(0, 100)
    myData = {'detect': detect, 'temperature': temp, 'humidity': humidity, 'url':img_url}
    print("Data : ",myData)

    #Send to IBM Watson
    if ((temp != None)  and (humidity != None)):
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
        print("Published")
    #Commands from IBM
    client.commandCallback = myCommandCallback
    #Wait (in Seconds)
    time.sleep(45)

cap.release()
cv2.destroyAllWindows()
#Disconnect device
client.disconnect()
```

GitHub & Project Demo Link :

GitHub link : https://github.com/IBM-EPBL/IBM-Project-13844-1659533037

Project Demo Link : https://clipchamp.com/watch/V0kVedGYnAX