

**MAHENDRA ENGINEERING
COLLEGE WOMEN**

Category:INTERNETOFTHINGS

A PROJECT REPORT

Submittedby

K.SUBASHREE(TEAM LEADER)

P. SOBIYA

M. SUMITHRA

M. THILOTHAMA

ECHNOLOGTECHN

COMPUTER SCIENCE AND ENGINEERING

NALAIYATHIRAN2022TeamId:PNT2022TMID3041PNT2022TM

ID30400

PROJECTGUIDES

IndustryMentor:MANIMEGALAI

FacultyMentor:J P KAVITHA

INDEX

1. INTRODUCTION

a. ProjectOverview

b. Purp

ose2.

LIT

ERA

TUR

ESU

RV

EY

a. Existingproblem

b. Problem Statement

Definition3.IDEATI

ON&PROPOSEDS

OLUTION

- a. **EmpathyMapCanvas**
- b. **Ideation&Brainstorming**
- c. **ProposedSolution**
- d. **Problem**

Solution

fit4.REQ

UIREME

NTANAL

YSIS

- a. **Functionalrequirement**
- b. **Non-Functional**

requirements5.PR

OJECTDESIGN

- a. **DataFlowDiagrams**
- b. **Solution&TechnicalArchitecture**
- c. **UserStories**

1. PROJECT PLANNING & SCHEDULING

a. Sprint Planning & Estimation

b. Sprint Delivery Schedule

2. CODING & SOLUTION

a. Feature 1

3. CONCLUSION

4. FUTURE SCOPE

5. APPENDIX

a. Source Code

i. INTRODUCTION

a. Project Overview:

Smart crop protection system

Smart crop protection system solutions use sensors placed in crop yields to measure humidity ,temperarture , moisture and to notify farmers when crops are ready to be emptied. Over time, historical data collected by sensors can be used to identify crop patterns . The cost of these sensors is steadily decreasing, making IoT crop protection more feasible to implement and more attractive to farmer.

b. Purpose:

1. At present, we can see crop are being damaged due to many reasons. Our primary goal is to protect the crop from being damaged .
—
2. Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield.
3. In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops .
4. So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.
—

ii. LITERATURE SURVEY

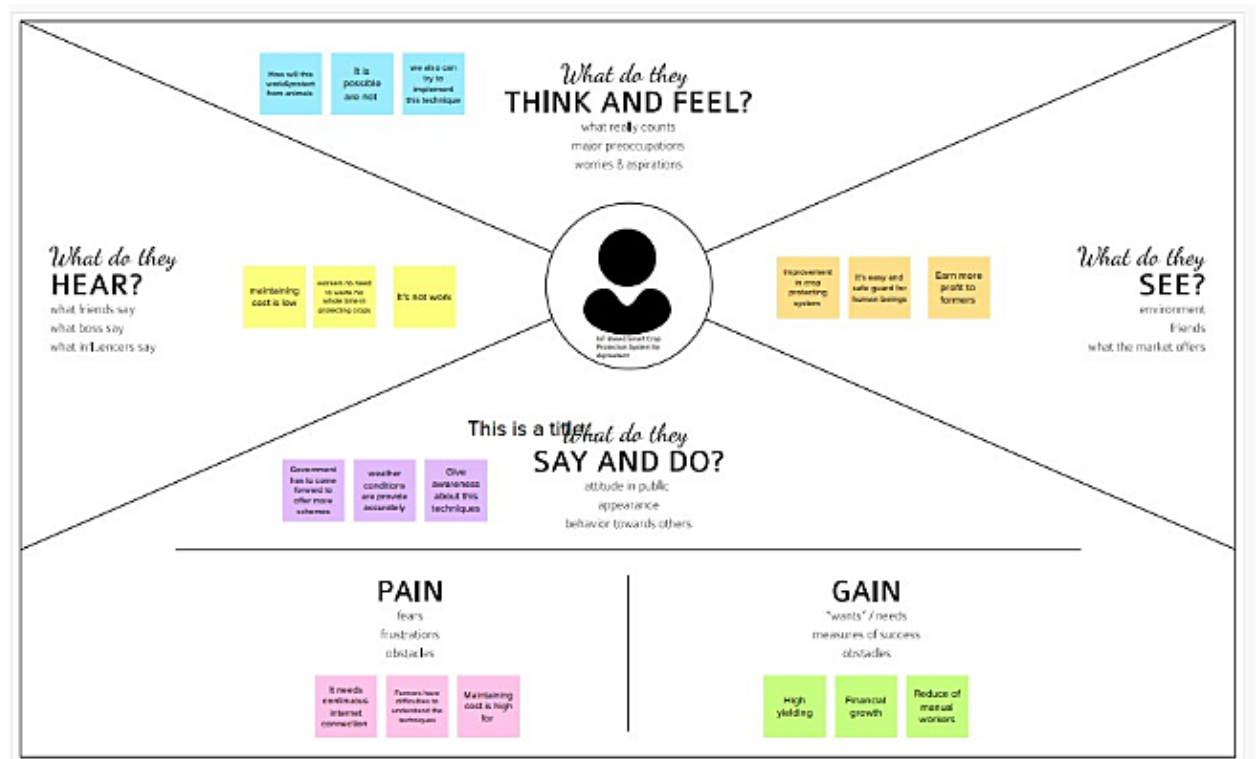
a. Existing Problem:

5. At present, we can see crop are being damaged due to many reasons. Our primary goal is to protect the crop from being damaged.
—
 - a. Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield.
 - b. In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops
 - c. So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

b. Problem Statement Definition:

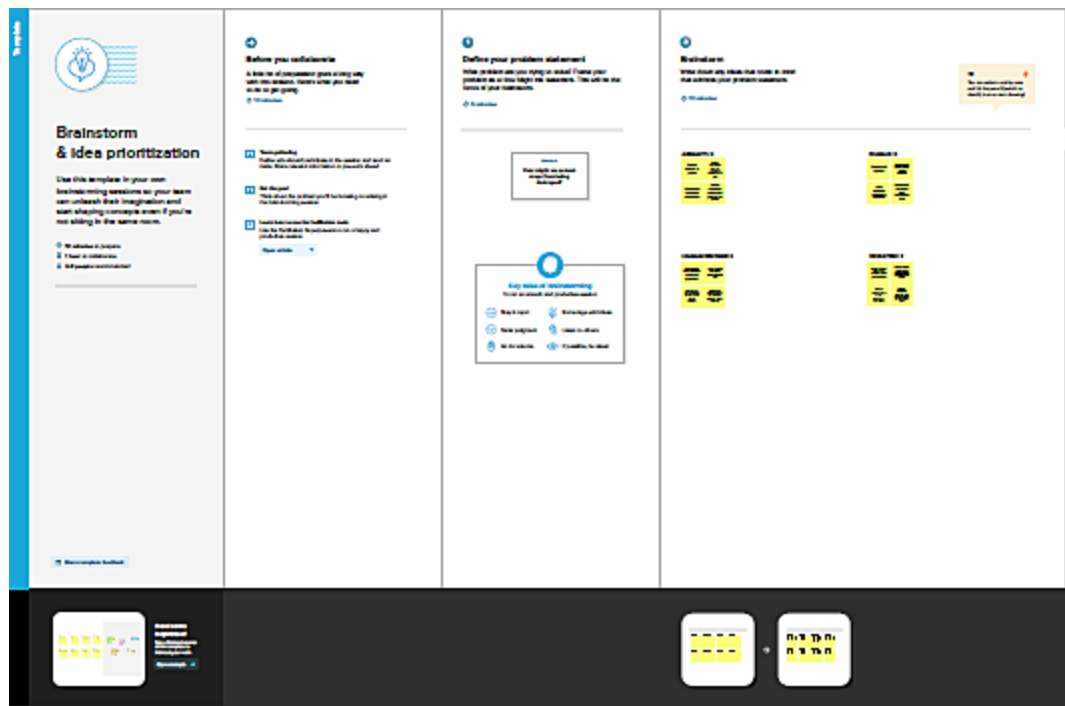
6. Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield .In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops .So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

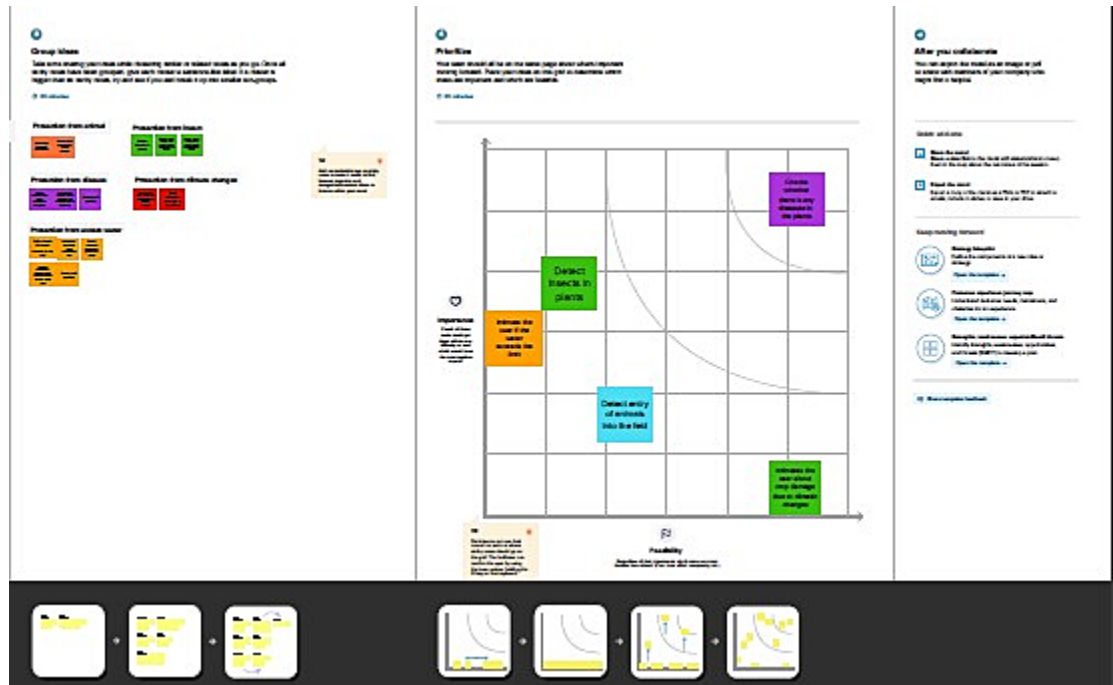
IDEATION&PROPOSEDSOLUTION



a. Ideation&Brainstorming:

Ideation and Brainstorming Ideation is often closely related to the practice of brainstorming, a **specific technique that is utilized to generate new ideas**. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.





a. ProposedSolution:

S.No.	Parameter	Description
1.	ProblemStatement(Problemtoresolved)	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

2.	Idea/Solutiondescription	<p>The aim of the proposed work is to develop an IOT device for smart crop field monitoring system and automated irrigation system using the wireless sensor networks(WSN) . To create an IOT device for monitoring the crop field using sensors (soil moisture ,temperature ,Humidity ,etc.,) To automate the irrigation by comparing the level of soil moisture with the threshold value .</p>
3.	Novelty/Uniqueness	<p>Daily update about the Condition of the land send to the farmers via mail</p>
4.	SocialImpact/CustomerSatisfaction	<ul style="list-style-type: none"> • Cost effective to the society • Modernaization to the society • High protection and High yield

5.	Business Model(RevenueMode l)	Outcome based model Data based model Platform based model
6.	ScalabilityoftheSolution	Start small and build out

Problem-Solution Fit canvas

IOT based Smart Crop Protection for Agriculture

Version:1.0

Focus on PR, tap into BE, understand RC	1. CUSTOMER SEGMENT(S) CS <p>Our customers are farmers who are affected by damage of crops due to various reasons like insect attacks, animal invasion, Excess water flow, etc.</p>	6. CUSTOMER LIMITATIONS CL <small>EG. BUDGET, DEVICES</small> <p>Customer must have minimum knowledge about using the technology.</p>	5. AVAILABLE SOLUTIONS AS <small>PLUSES & MINUSES</small> <ul style="list-style-type: none"> • Complete control and elimination of yield-threatening weeds • Protection from diseases for healthier farm output • Protection from insects for high yields and quality 	Explore AS, differentiate
	2. PROBLEMS / PAINS PR <small>+ ITS FREQUENCY</small> <p>Major problems farmers face is crops being damaged by insects, animals, water and various climatic changes.</p>	9. PROBLEM ROOT / CAUSE RC <ul style="list-style-type: none"> - Sense the animals and insects in the crop field. - Sense the water required for the crop. - Sense the required climate for the crop. 	7. BEHAVIOR BE <small>+ ITS INTENSITY</small> <p>Crop protector informs customer about the crop from being damaged by insect, animal, excess water flow, climatic changes, etc.</p>	
Identify strong TR & EM	3. TRIGGERS TO ACT TR <p>This triggers to protect the crops from insects, animals, excess water, climatic changes, unwanted plant growth etc.</p>	10. YOUR SOLUTION SL <p>The aim of the proposed work is to develop an IoT device for smart crop field monitoring system and automated irrigation system using the wireless sensor networks (WSN). To create an IoT device for monitoring the crop field using sensors (soil moisture, temperature, Humidity, etc...). To automate the irrigation by comparing the level of soil moisture with the threshold value.</p>	8. CHANNELS of BEHAVIOR CH <p>ONLINE</p> <p>Notifies the customer about the crops being damaged</p>	Extra Fast
	4. EMOTIONS EM <small>BEFORE / AFTER</small> <p>Before there is no technology to protect the crop from insects, animals, excess water, climatic changes, unwanted plant growth, etc, so many farmers committed suicides.</p>		<p>OFFLINE</p> <p>Senses the crops</p>	

4 REQUIREMENT ANALYSIS

a. 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 Registration through Gmail .
FR-2	User Confirmation	Confirmation via Email .
FR-3	Interfacing with hardware	Interface the sensors with the software application so as to alert the farmers in ps .
FR-4	Database Connection	Databases are retrieved from IBM Cloudant .
FR-5	Mobile Application	Alarm and motors can be accessed from the mobile app .

b. Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The smart crop protection alerts the farmers in case of any obstacles and helps in protecting
NFR-2	Security	Smart Agriculture can improve the farming practices and maintain sustainable production especially by preventing the animals into the agricultural land through IoT enabled devices
NFR-3	Reliability	With a proper power supply, SD card and programming the processor should be able to run 24/7 for years. The SD card and power supply will likely wear out faster than the Pi. The possible reasons behind Raspberry Pi failure can be power breakdowns, SD card failures, and ineligible environments.
NFR-4	Performance	Usage of an SD card module that helps to store as specified sound to scare the animals. Crop damaged due to animal attack can be sensed. Network and Design Evaluation
NFR-5	Availability	Agriculture for different variety of crops is based on the monsoon changes, indoor outdoor climatic temperatures, availability of rain fall and irrigation methods.

NFR-6	Scalability	The product shall be made available to everyone especially in remote areas for better efficiency of crop yield with the better safety of crops as well as the farmers.
-------	-------------	--

5 PROJECT DESIGN

a. Data Flow Diagrams:

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

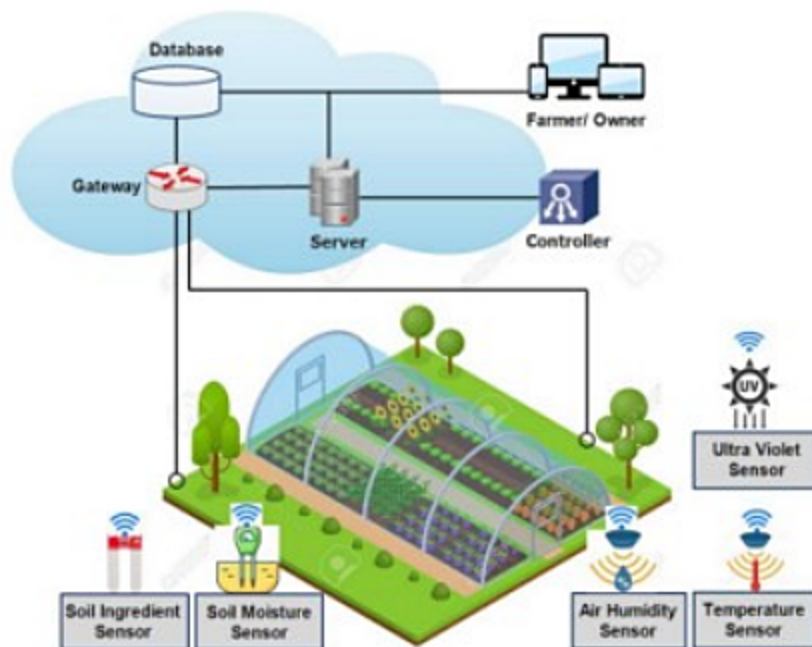
a. SOLUTION AND TECHNICAL ARCHITECTURE

i. Summary:

The smart crops are constructed based on the sensor application and raspberry pi.

can also act as a transceiver since it is connected to the mobile phone of the user . The overall process of the sensors and raspberry pi is monitored using real time monitor which can help data transmission.This is stored and formulated using cloud data . Through which the admin can access the data and then track the location from GPS .

SOLUTION ARCHITECTURE :



Components

i. Components&Technologies:

S. No	Component	Description	Technology

1.	User Interface	How user interacts with application. e.g. WebUI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript/AngularJs/ReactJs etc.
2.	Application Logic-1	Logic for a process in the application	Java/Python
	Database	Data Type, Configuration etc.	MySQL, NoSQL, etc
4.	IoT	To collect the data and alert the users	IBM Watson IoT Platform, Node Red.
5.	Cloud Database	Database Service on Cloud	Cloudant DB

ii. **Application Characteristics:**

S. No	Characteristics	Description
1.	Open Source framework	List the open-source frameworks used

2.	Security implementation	List all the security/access controls implemented, use of firewall etc .,
3.	Scalable Architecture	Justify the scalability of architecture (3-tier, Micro-services)
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.

a. 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
-----------	-------------------------------	-------------------	-----------------

Customer(Farmer)	Maintain fields	USN-1	As a user, I can monitor the growth of crops and protect the crops against animals
	Analyzing problem	USN-2	As a user, I collect the required information about the problems on agriculture fields
Project Designers	Identifying the problem and provides solutions	USN-3	As a user, I can sense the water level and flame in the field using sensor and monitor using IOT
Customer field Maintainer	Problem solution	USN-4	As a user, an area can be monitored from a remote place
	Final process	USN-5	This proposed smart IOT-based crop protection device is found to be cost-effective and efficient

1. PROJECT PLANNING AND SCHEDULING

a. SPRINT PLANNING & SCHEDULING:

TITLE	DESCRIPTION
Literature Survey & Information Gathering	Literature survey on the selected project is done by gathering information about related web browsing.

Prepare EmpathyMap	PreparedEmpathy Map Canvastocapture the user Pains & Gains list of problem state
Ideation	List the organizing the brainstorming session and prior feasibility and importance.
ProposedSolution	Preparedtheproposedsolution whichincludesthenovelty,feasibilityofidea,businessmodel.
ProblemSolutionFit	Prepared problem -solutionfitdocument.
Solution Architecture	Prepare solution architecture document.

b. **SPRINTDELIVERYSCCHEDULE**

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule.

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Fpic)	User Story Number	User Story / Task	Story Points (40)	Priority (Low to High)	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the required dataset by entering my email, password, and confirming my password.	3	High	Umareshwari
Sprint-1		USN-2	As a user, I will receive confirmation email and the SMS once I have registered for the application.	2	High	Umareshwari
Sprint-2	Cloud services	USN-3	As a user, I can register for the application through Facebook or any social media.	1	Low	Abhinav
Sprint-4		USN-4	As a user, I can register for the application through Gmail/web service.	2	Medium	Dhanu
Sprint-3	Login	USN-5	As a user, I can log into the application network by entering email & password.	4	High	Venkaresh
Sprint-2	Pre processing	USN-6	As a farmer, the user must be able to find the system easy to access so pre-processing and other task must be perfect.	3	High	Dhanu
Sprint-1	Collecting Dataset	USN-7	To collect various sources of animal threats and keep developing a dataset.	3	Medium	Umareshwari
Sprint-4	Integrating	USN-8	To integrate the available dataset and keep improving the accuracy of finding animals.	2	High	Abhinav
Sprint-3		USN-9	To find and use appropriate compiler to run and test the data so that we can implement our program.	1	Low	Venkaresh
Sprint-2		USN-10	Request Saveetha Engineering College to deploy the project in our campus and test.	1	Low	Dhanu
Sprint-1	Training	USN-11	As programmer, we need to train our data perfectly so that the program runs smoothly.	3	High	Umareshwari
Sprint-3		USN-12	Train the data using our available servers and IFM dataset from server and improve that.	2	Medium	Venkaresh
Sprint-4	Coding	USN-13	To modify the code according to our program and improve the efficiency of that code.	4	High	Abhinav
Sprint-2		USN-13	To improve performance.	1	Low	Venkaresh
Sprint-2	Record	USN-5	To record the data and plot the graph to show the characteristics officially.	4	High	Venkaresh
Sprint-1	Planning	USN-4	Plan the programming language and feasibility.	3	Medium	Umareshwari, Abhinav
Sprint-4		USN-14	Demonstrate the working and improve accuracy overall.	2	Low	Abhinav

Project Tracker, Velocity & Burndown Charts

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (EndDate)
Sprint-1	20	5 Days	20 Oct 2022	24 Oct 2022	20
Sprint-2	20	5 Days	25 Oct 2022	29 Oct 2022	20

Sprint-3	20	5Days	31Oct 2022	4Nov2022	20
Sprint-4	20	7Days	5Nov 2022	11Nov2022	20

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	5 Days	20 Oct 2022	24 Oct 2022	20	21 Oct 2022
Sprint-2	20	5 Days	25 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint-3	20	5 Days	31 Oct 2022	4 Nov 2022	20	2 Nov 2022
Sprint-4	20	7 Days	5 Nov 2022	11 Nov 2022	20	8 Nov 2022

Velocity:

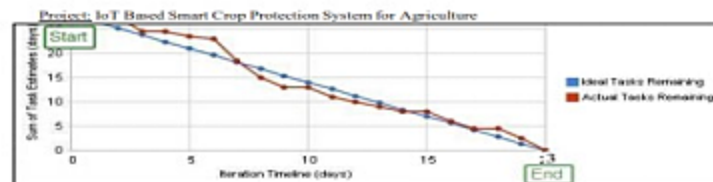
We have a 23-day sprint duration, and the velocity of the team is 20 (points per sprint).

To Find: Calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{23}{20} = 1.15$$

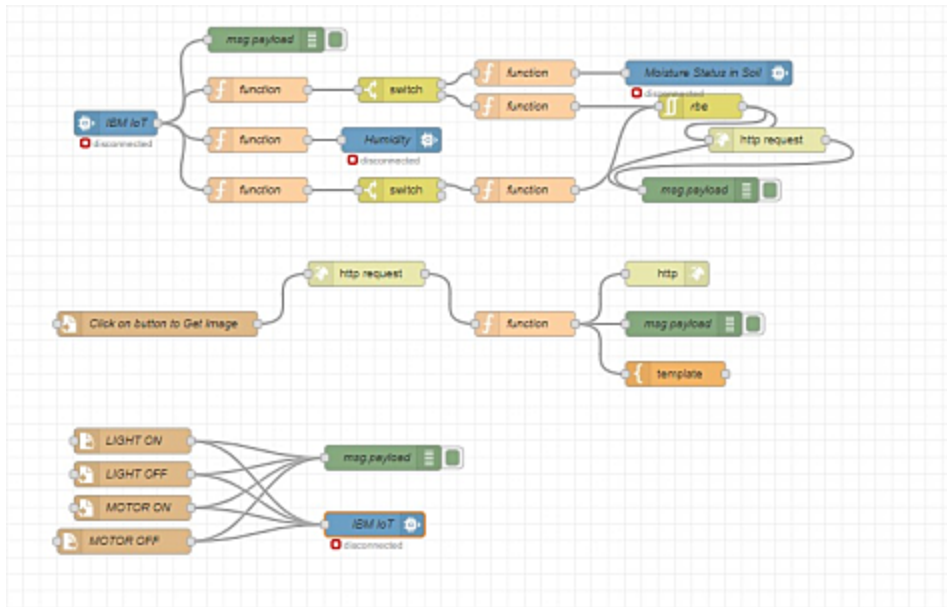
Burndown Chart:

A burn down 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.



1. CODINGANDSOLUTIONING

NODEREDSERVICEASSOCIATEDWITHIBMCLLOUD:



IBM Watson IoT Platform

umasevi814@gmail.com
ID: jcgym

Browse Action Device Types Interfaces

Add Device

Identity Device Information **Recent Events** State Logs

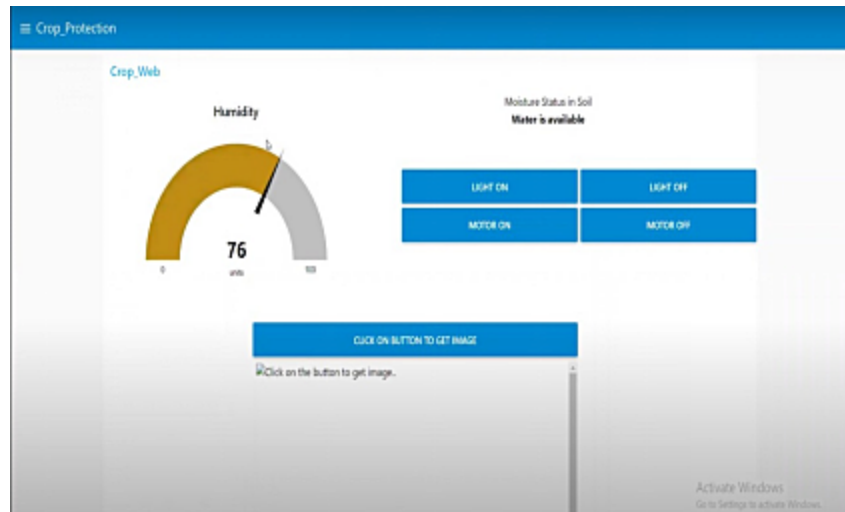
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Humidity	{"randomNumber":36}	json	a few seconds ago
Temperature	{"Temperature":3}	json	a few seconds ago
Moisture	{"Moisture":54}	json	a few seconds ago
Humidity	{"randomNumber":70}	json	a few seconds ago
Temperature	{"Temperature":68}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 Simulation running

NoderedDashboard:



a. CONCLUSION

We presented an intelligent Smart crop protection system. The system is based on IoT sensors. It is responsible for measuring the waste level in the smart crop. When the smart crop gets affected almost there will be information received by the admin, Since the admin can access the data and location of the crop. Later send this data (through Internet) to a server for storage and processing.

This data helps to compute the optimized collection routes for the workers. In future, we would like to enhance the system for different kind of crop management system .

b. FUTURESCOPE

The advantage of this work is its contribution in making a Smart crop. Among the many challenges that a city faces, crop protection management is of utmost importance. This is because, it is directly related to food of people living in the area. We are further extending this work to address problems of segregating different kind of crops (e.g., paddy, wheat, etc.,), and identifying different agricultural department for collecting it. The optimization algorithms may be devised accordingly depending on the requirements. In future, we would like to enhance the system for different kind of crops.

11.APPENDIX

a. SOURCECODE:

PYTHONCODETOPUBLISHDATA

```
import cv2  
import numpy as np
```

```

import wiotsdk.device
import playsound
import random
import time
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError

# CloudantDB
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
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resource_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

# This is how you authenticate
metadata = (('authorization', 'key 0620e202302b4508b90eab7efe7475e4'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "g5d4qO8EIgv4TWUCJj4hfEzgalqEjrDbE82AJDWlAOHo"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-  
storage:global:a/c2fa2836eaf3434bbc8b5b58fefff3f0:62e450fd-4c82-4153-ba41-  
ccb53adb8111::"
clientdb = cloudant("apikey-

```

```
W2njldnwtjO16V53LAVUCqPwc2aHTLmlj1xXvtdGKJBn",
"88cc5f47c1a28afbfb8ad16161583f5a", url="https://d6c89f97-cf91-48b7-b14b-
c99b2fe27c2f-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,
ibm_auth_endpoint=COS_AUTH_ENDPOINT,
config=Config(signature_version="oauth"),
endpoint_url=COS_ENDPOINT
)
```

```
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))
```

```
    #set 5 MB chunks
```

```
part_size = 1024 * 1024 * 5
```

```
    #set threadhold to 15 MB
```

```
file_threshold = 1024 * 1024 * 15
```

```
    #set the transfer threshold and chunk size
```

```
transfer_config = ibm_boto3.s3.transfer.TransferConfig(
```

```
multipart_threshold=file_threshold,
```

```
multipart_chunksize=part_size
```

```
)
```

```
    #the upload_fileobj method will automatically execute a multi-part upload
```

```
    #in 5 MB chunks 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))

```

```

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    command = cmd.data['command']
    print(command)
    if (command == "lighton"):
        print('lighton')
    elif (command == "lightoff"):
        print('lightoff')
    elif (command == "motoron"):
        print('motoron')
    elif (command == "motoroff"):
        print('motoroff')
    myConfig = {
        "identity": {
            "orgId": "chytun",
            "typeId": "NodeMCU",

```

```
    "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
```

```
client = wiot.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

```
database_name = "sample"
my_database = clientdb.create_database(database_name)
if my_database.exists():
    print(f'"{database_name}" successfully created.')
cap=cv2.VideoCapture("garden.mp4")
if(cap.isOpened()==True):
    print('File opened')
else:
    print('File not found')
```

```
while(cap.isOpened()):
    ret, frame = cap.read()
    gray = cv3.cvtColor(frame, cv2.COLOR_BGR@GRAY)
    imS= cv2.resize(frame, (960,540))
    cv2.imwrite('ex.jpg',imS)
    with open("ex.jpg", "rb") as f:
        file_bytes = f.read()
```

#This is the model ID of a publicly available General model. You may use any other

public or custom model ID.

```
request = service_pb2.PostModelOutputsRequest(  
model_id='e9359dbe6ee44dbc8842ebe97247b201',  
inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base  
64=file_bytes))  
))
```

```
response = stub.PostModelOutputs(request, metadata=metadata)
```

```
if response.status.code != status_code_pb2.SUCCESS:
```

```
raise Exception("Request failed, status code: " + str(response.status.code))
```

```
detect=False
```

```
for concept in response.outputs[0].data.concepts:
```

```
    #print('%12s: %.f' % (concept.name, concept.value))
```

```
if (concept.value>0.98):
```

```
    #print(concept.name)
```

```
if (concept.name=="animal"):
```

```
print("Alert! Alert! animal detected")
```

```
playsound.playsound('alert.mp3')
```

```
picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
```

```
cv2.imwrite(picname+'.jpg',frame)
```

```
multi_part_upload('Dhakshesh', picname+'.jpg', picname+'.jpg')
```

```
    json_document={"link":COS_ENDPOINT+'/'+Dhakshesh+'/'+picname+'.jpg'}
```

```
new_document = my_database.create_document(json_document)
```

```
if new_document.exists():
```

```
print(f"Document successfully created.")
```

```
time.sleep(5)
```

```
detect=True
```

```
moist=random.randint(0,100)
```

```
humidity=random.randint(0,100)
myData={'Animal':detect,'moisture':moist,'humidity':humidity}
print(myData)
if(humidity!=None):
    client.publishEvent(eventId="status",msgFormat="json", daya=myData, qos=0,
onPublish=None)
print("Publish Ok..")
client.commandCallback = myCommandCallback
cv2.imshow('frame',imS)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
client.disconnect()
cap.release()
cv2.destroyAllWindows()
```

-

-

```
*IDLE Shell 3.8.8*
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:10:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/HP/Desktop/crop/crop_protect.py =====
2021-04-06 12:52:19,640 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:hj5fmy:NodeMCU:12345
'sample' successfully created.
File opened
({'Animal': False, 'moisture': 17, 'humidity': 41})
Publish OK..
({'Animal': False, 'moisture': 84, 'humidity': 16})
Publish OK..
({'Animal': False, 'moisture': 48, 'humidity': 43})
Publish OK..
({'Animal': False, 'moisture': 0, 'humidity': 3})
Publish OK..
({'Animal': False, 'moisture': 73, 'humidity': 68})
Publish OK..
({'Animal': False, 'moisture': 26, 'humidity': 26})
Publish OK..
({'Animal': False, 'moisture': 96, 'humidity': 59})
Publish OK..
I
```

IBM Watson IoT Platform umasek014@gmail.com ID: jegvm Add Device

Browse Action Device Types Interfaces

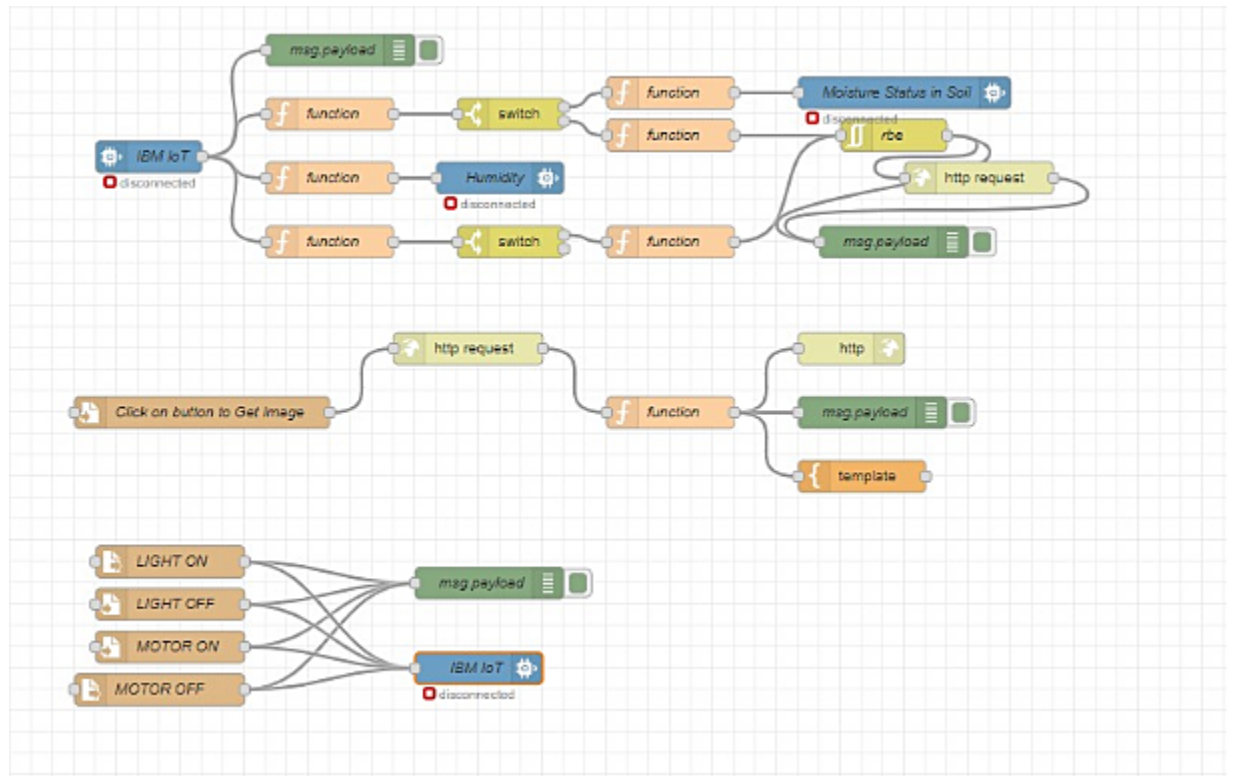
Identity Device Information **Recent Events** State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Humidity	{"randomNumber":36}	json	a few seconds ago
Temperature	{"Temperature":3}	json	a few seconds ago
Moisture	{"Moisture":54}	json	a few seconds ago
Humidity	{"randomNumber":70}	json	a few seconds ago
Temperature	{"Temperature":68}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 Simulation running



TECH TO SPEECH:

```
fromibm_watson import TextToSpeechV1
fromibm_cloud_sdk_core.authenticators import IAMAuthenticator
importplaysound
```

```
authenticator = IAMAuthenticator('v9n8Zn4r5VpcMVz_HyRY0DrS13jSzph2IEFioVj4-vmT')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)
```

```
text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm')
```

```
with open('alert.mp3', 'wb') as audio_file:
    audio_file.write(
        text_to_speech.synthesize(
            'Alert! Alert! Animal Detected.',
            voice='en-US_ALLisonV3Voice',
            accept='audio/mp3'
        ).get_result().content)
    playsound.playsound('alert.mp3')
```

Crop_Web



Moisture Status in Soil
Water is available


LIGHT ON

LIGHT OFF

MOTOR ON

MOTOR OFF

CLICK ON BUTTON TO GET IMAGE

 Click on the button to get image.

Activate Windows
Go to Settings to activate Windows.