Final Project Report

Team ID	PNT2022TMID30928
Project Title	Iot Based Smart Crop Protection System for Agriculture
Date	19 November 2022

Team Leader

S.Sowndarya(620119106094)

Team Members

Nandhini D.(620119106055)
A.Sowmya(620119106093)
H.Sneka(620119106090)

IoT Based Smart Crop Protection System For Agriculture

INTRODUCTION

Generally farmers face many problems in their lives. Some of them can be overcome by taking preventive measures and some of them can not be taken care of. The measures that they take may or may not solve the problem permanently. The problems for farmers reoccur every now and then. Some of the ways to solve these problems may not be known to them. It may cause various problems in their fields and in their lives too.

Project Overview

Our project is to create an application that is connected to various types of sensors that are connected to the fields. These sensors sense various types of datas and send those datas to the mobile application that is in the farmers mobiles. The information such as temperature, humidity level of the atmosphere, soil pH levels, weather reports, motion of any animals and birds, crops condition and so on.

Purpose

The main purpose of this project is to intimate the farmers about the condition of the field even when they are not at the fields. To intimate them about the condition of the field, the weather and humidity changes when they are not there to take necessary actions. When an animal enters the field, it damages the crops, so we use image processing to ensure the safety of the crops and animals by giving the farmers the detailed whereabouts of the animals in their fields.

LITERATURE SURVEY

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.

As a result of this system, we can detect the changes in the field easily and intimate the farmers about it and also we can take precautions and do remedies accordingly. Here we use very low power consuming highly efficient components that give us accurate results and also they perform at low data rate conditions without any lag and help in finding the remedies. This crop protection system helps in detection of all kinds of external dangers and it saves time and money to the farmers before any loss that may occur. With the help of this system the farmers can be in a peaceful environment at ease without any pressure.

Problem statement Definition:

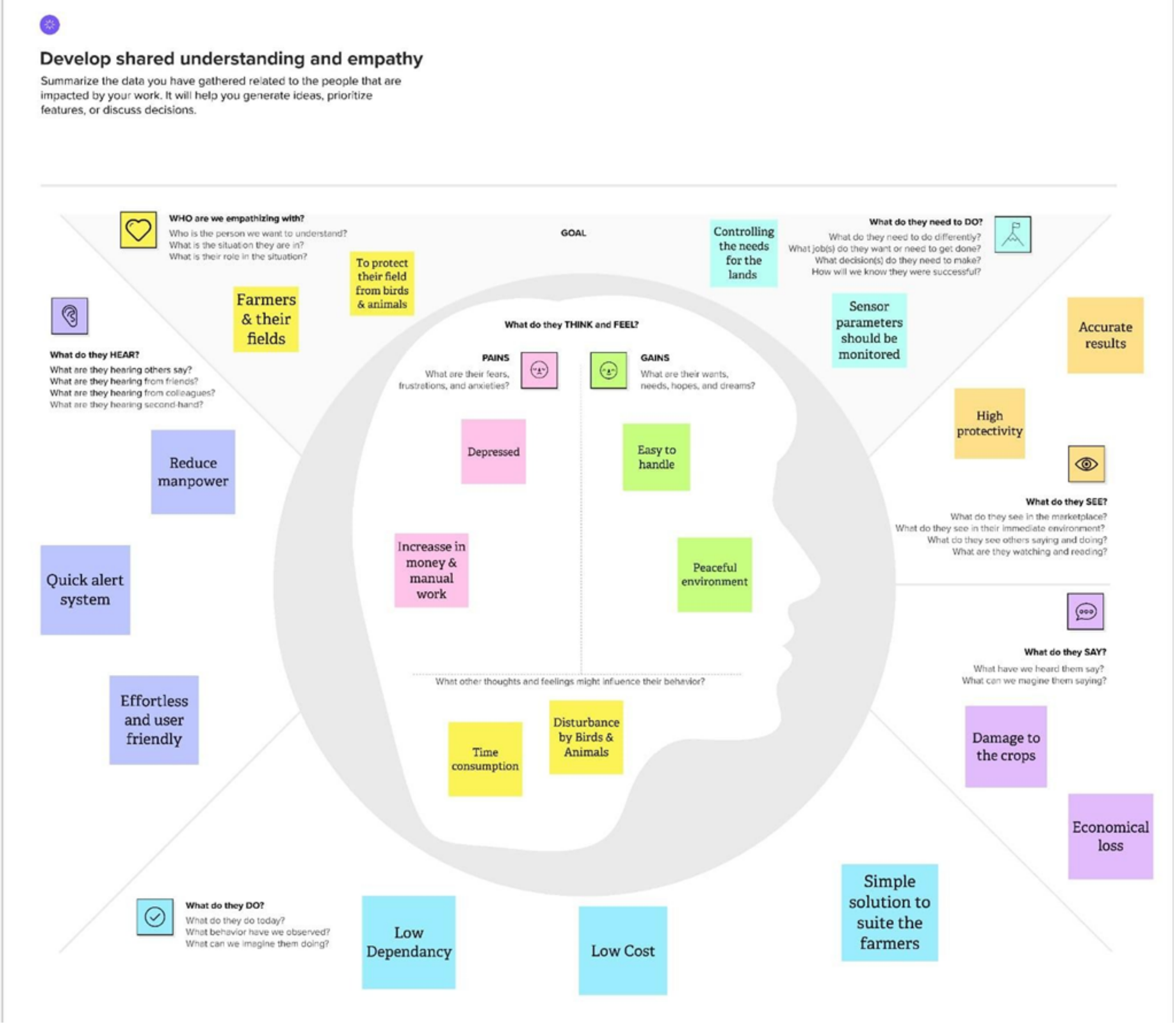
Iot based smart crop protection system for agriculture - is the project to avoid the entry of animals into the fields and if they enter , then to intimate the farmers about the intrusion of the animals and to help them take necessary actions to avoid further damage of crops.

Customer Problem Statement:

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Farmer	Monitor my crops	There are some disturbances	Of birds, animals & insects	Very frustrated and depressed about my field
PS-2	Farmer	Prevent animals from attacking my field	There is no easy and helpful technology	Of many kinds of birds & animals attack according to the type of cultivation	Unable to do anything many times

IDEATION & PROPOSED SOLUTIONS:

Enthalpy Map:



Develop shared understanding and empathy

Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.

WHO are we empathizing with?

Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

Farmers & their fields

To protect their field from birds & animals

GOAL

Controlling the needs for the lands

What do they need to DO?

What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

Sensor parameters should be monitored

Accurate results

High protectivity

What do they HEAR?

What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

Reduce manpower

Quick alert system

Effortless and user friendly

What do they THINK and FEEL?

PAINS

What are their fears, frustrations, and anxieties?

Depressed

Increasse in money & manual work

GAINS

What are their wants, needs, hopes, and dreams?

Easy to handle

Peaceful environment

What other thoughts and feelings might influence their behavior?

Time consumption

Disturbance by Birds & Animals

What do they SEE?

What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

What do they SAY?

What have we heard them say?
What can we magine them saying?

Damage to the crops

Economical loss

What do they DO?

What do they do today?
What behavior have we observed?
What can we imagine them doing?

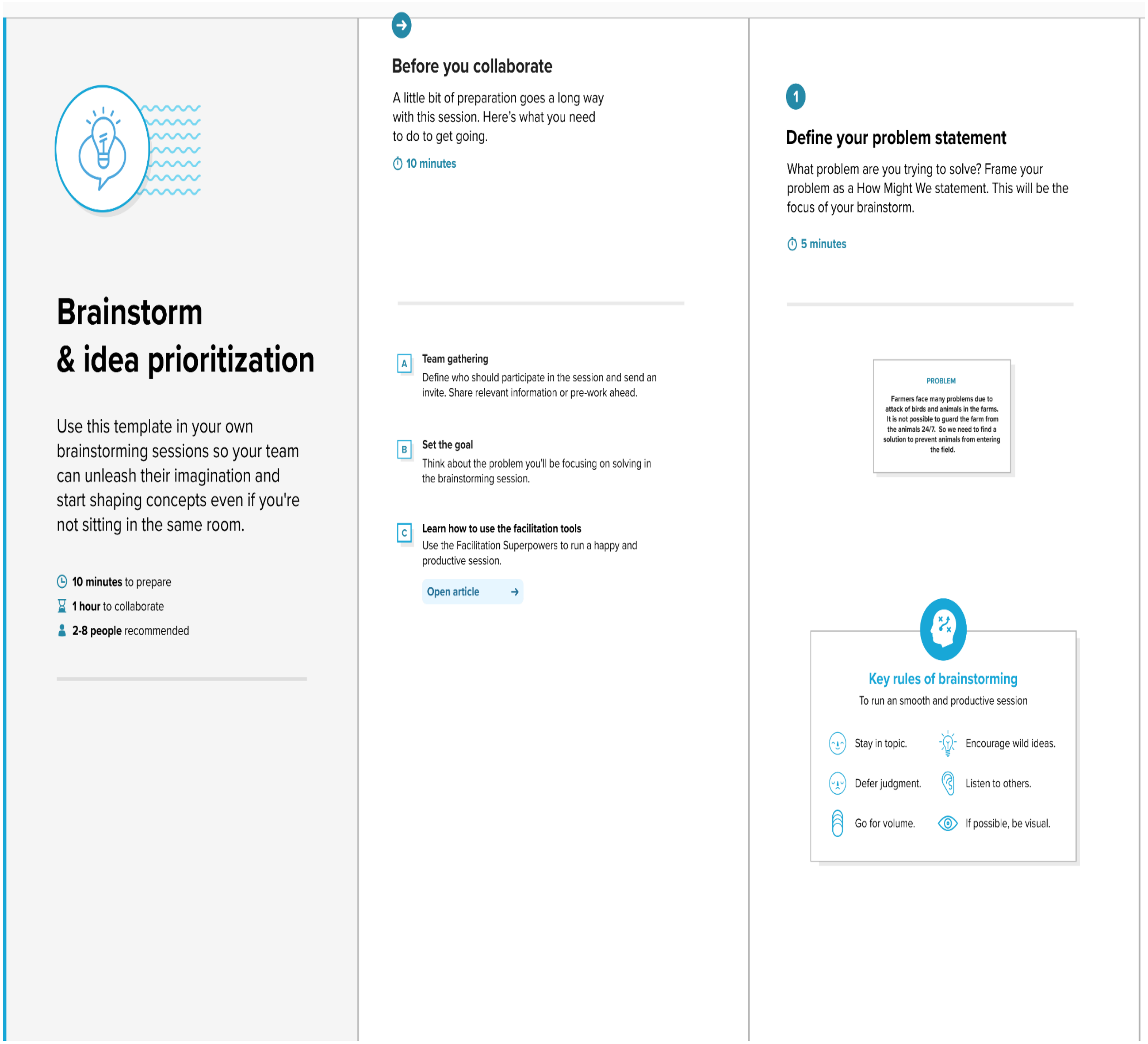
Low Dependancy

Low Cost

Simple solution to suite the farmers

Ideation & Brainstorming:

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



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

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

5 minutes

PROBLEM

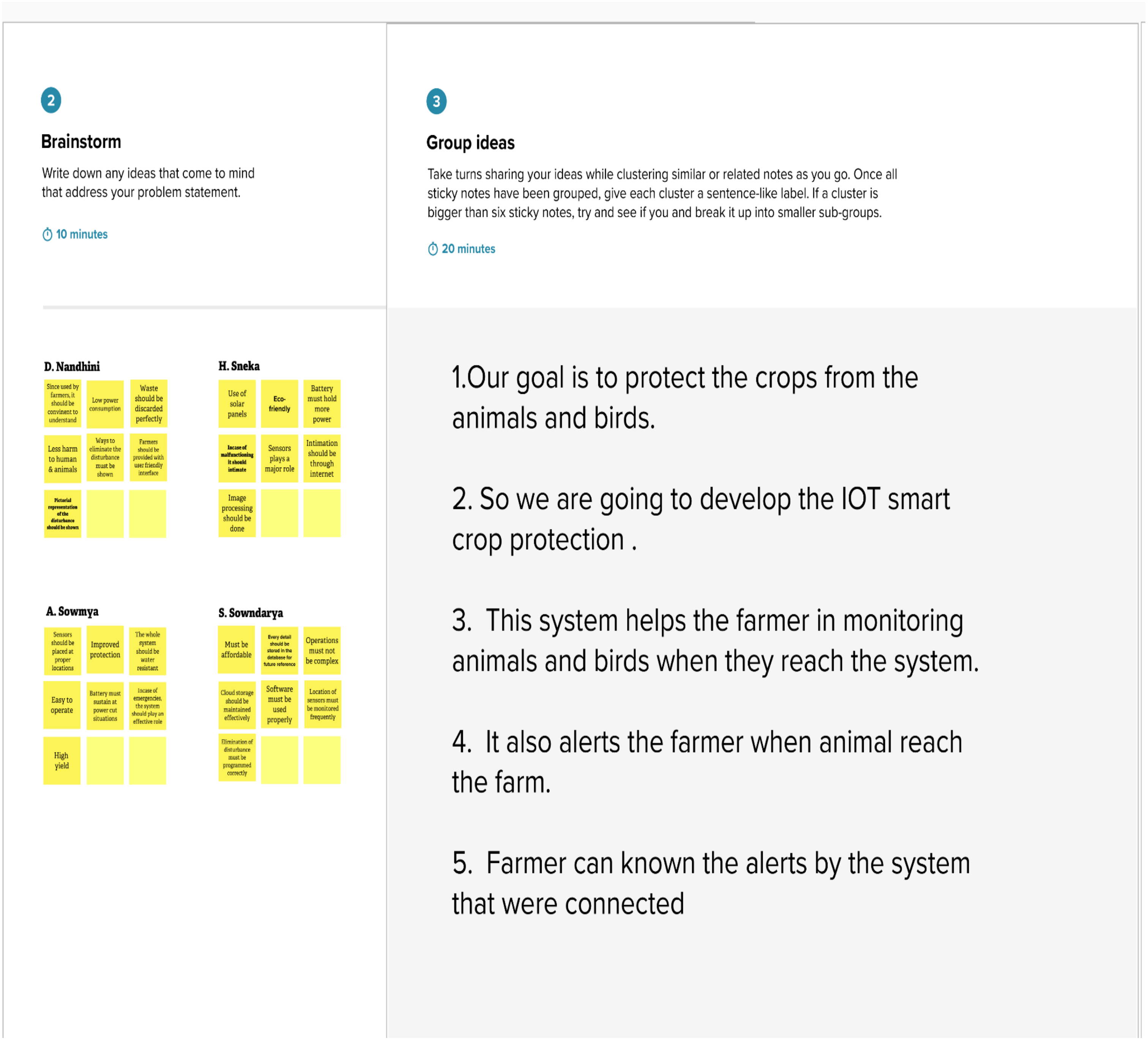
Farmers face many problems due to attack of birds and animals in the farms. It is not possible to guard the farm from the animals 24/7. So we need to find a solution to prevent animals from entering the field.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

D. Nandhini

Since used by farmers, it should be convinent to understand

Low power consumption

Waste should be discarded perfectly

Less harm to human & animals

Ways to eliminate the disturbance must be shown

Farmers should be provided with user friendly interface

Pictorial representation of the disturbance should be shown

H. Sneka

Use of solar panels

Eco-friendly

Battery must hold more power

Incase of malfunctioning it should intimate

Sensors plays a major role

Intimation should be through internet

Image processing should be done

A. Sowmya

Sensors should be placed at proper locations

Improved protection

The whole system should be water resistant

Easy to operate

Battery must sustain at power cut situations

Incase of emergencies, the system should play an effective role

High yield

S. Sowndarya

Must be affordable

Every detail should be stored in the database for future reference

Operations must not be complex

Cloud storage should be maintained effectively

Software must be used properly

Location of sensors must be monitored frequently

Elimination of disturbance must be programmed correctly

3

Group ideas

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

20 minutes

1.Our goal is to protect the crops from the animals and birds.

2. So we are going to develop the IOT smart crop protection .

3. This system helps the farmer in monitoring animals and birds when they reach the system.

4. It also alerts the farmer when animal reach the farm.

5. Farmer can known the alerts by the system that were connected

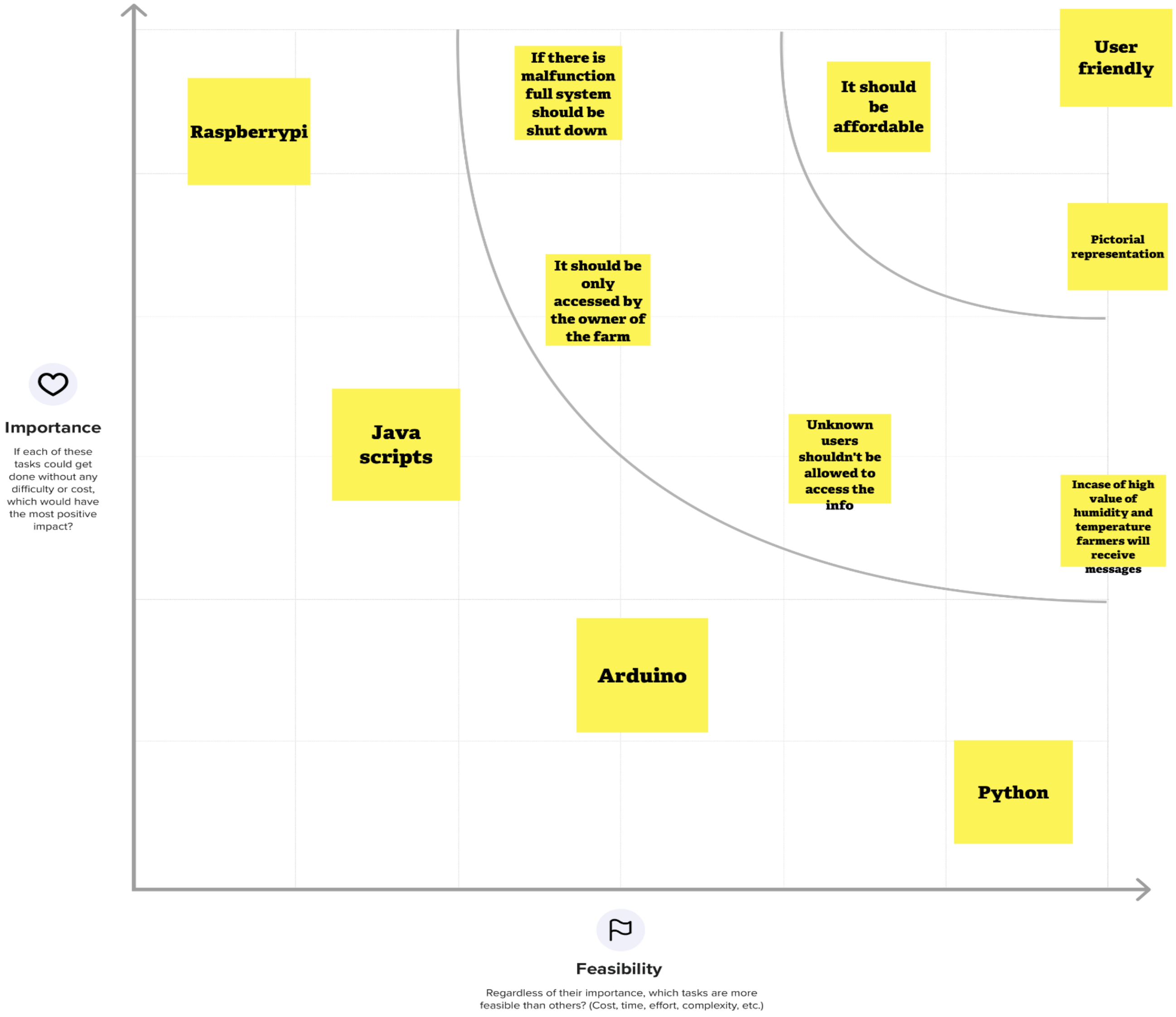
Step-3: Idea Prioritization

4

Prioritize

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

20 minutes



Raspberrypi

If there is malfunction full system should be shut down

It should be affordable

User friendly

Pictorial representation

It should be only accessed by the owner of the farm

Java scripts

Unknown users shouldn't be allowed to access the info

Incase of high value of humidity and temperature farmers will receive messages

Arduino

Python

Importance

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

Feasibility

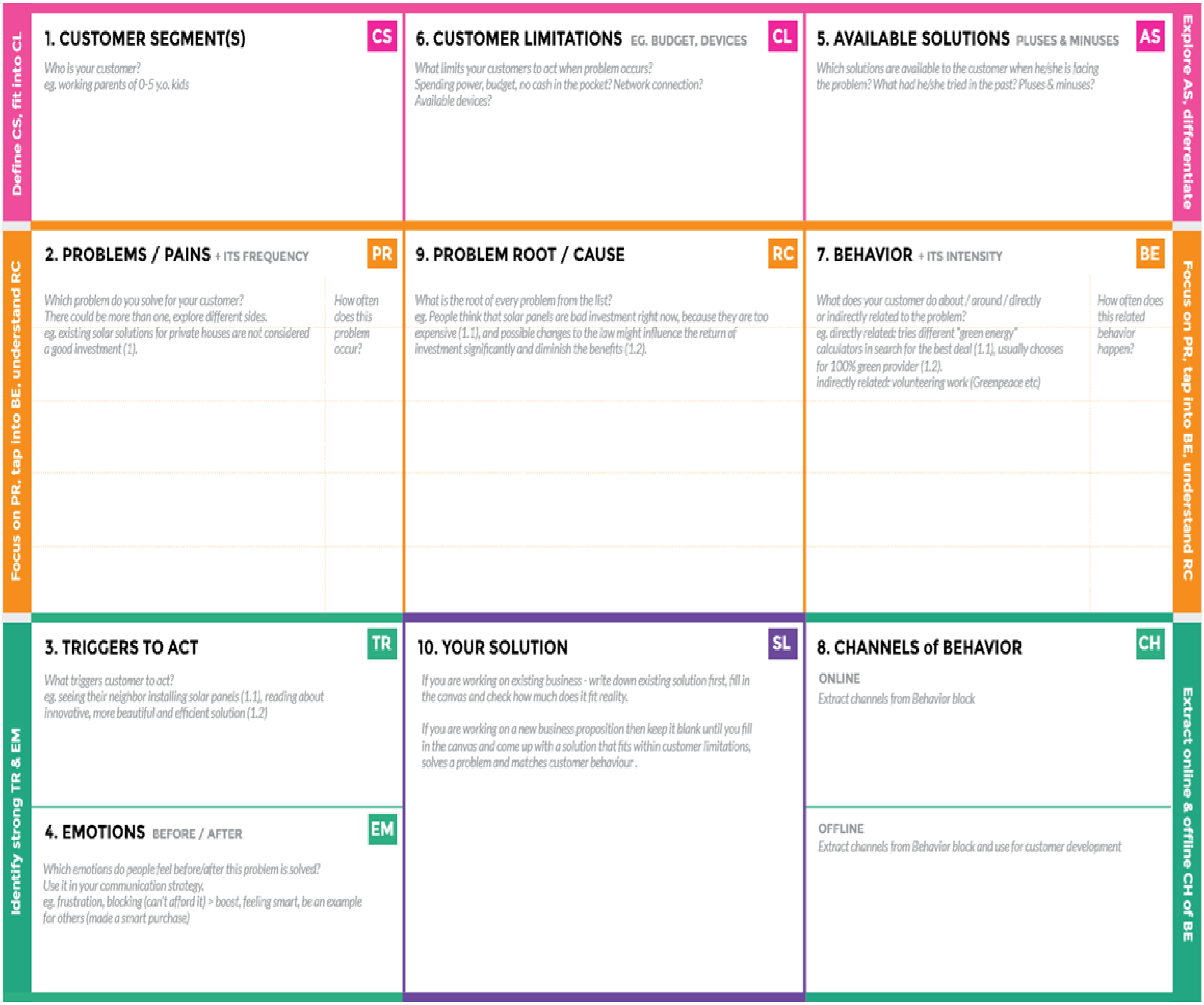
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop IOT based smart crop protection system
2.	Idea / Solution description	An IOT crop protection system is based on motion detection sensor and is developing especially for crop monitoring in agriculture fields,wet lands and farms
3.	Novelty / Uniqueness	· Conserving diversity · Preventing food related illness · Lowering the food cost
4.	Social Impact / Customer Satisfaction	· High yield · Increased quality · Lowering the food cost
5.	Business Model (Revenue Model)	The importance of crop protection system lies in conserving biodiversity and optimizing the resources used.

6.	Scalability of the Solution	Scalability in crop protection helps to protect the crops during different seasons

Problem Solution Fit:



Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
eg. working parents of 0-5 y.o. kids

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

What limits your customers to act when problem occurs?
Spending power, budget, no cash in the pocket? Network connection?
Available devices?

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

Which solutions are available to the customer when he/she is facing the problem? What had he/she tried in the past? Pluses & minuses?

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

Which problem do you solve for your customer?
There could be more than one, explore different sides.
eg. existing solar solutions for private houses are not considered a good investment (1).

How often does this problem occur?

9. PROBLEM ROOT / CAUSE RC

What is the root of every problem from the list?
eg. People think that solar panels are bad investment right now, because they are too expensive (1.1), and possible changes to the law might influence the return of investment significantly and diminish the benefits (1.2).

7. BEHAVIOR + ITS INTENSITY BE

What does your customer do about / around / directly or indirectly related to the problem?
eg. directly related: tries different "green energy" calculators in search for the best deal (1.1), usually chooses for 100% green provider (1.2).
indirectly related: volunteering work (Greenpeace etc)

How often does this related behavior happen?

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

What triggers customer to act?
eg. seeing their neighbor installing solar panels (1.1), reading about innovative, more beautiful and efficient solution (1.2)

4. EMOTIONS BEFORE / AFTER EM

Which emotions do people feel before/after this problem is solved?
Use it in your communication strategy.
eg. frustration, blocking (can't afford it) > boost, feeling smart, be an example for others (made a smart purchase)

10. YOUR SOLUTION SL

If you are working on existing business - write down existing solution first, fill in the canvas and check how much does it fit reality.

If you are working on a new business proposition then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour .

8. CHANNELS of BEHAVIOR CH

ONLINE
Extract channels from Behavior block

OFFLINE
Extract channels from Behavior block and use for customer development

Extract online & offline CH of BE

REQUIREMENT ANALYSIS:

Functional Requirements:

The main task for this unit is that it should help the farmers to be able to find the threats that occur to their fields.

This product has the feature of monitoring the changes that occur in the fields such as the humidity changes, temperature changes, ph levels modifications, motion sensing to detect animals and image processing of the animals.

The main focus is that the animals entering in the fields are captured by image processing and it is being intimated to the farmers.

The processed images must be captured by the system that helps the farmers to take necessary actions.

The functional units used are different types of sensors, cloud database , internet and image processing.

Non functional requirements:

The sensor captures the image and senses the datas in the field.

The sensed datas is then sent to the cloud database where all the datas is stored.

If any animals enter the field, then the image processing works and processes the image of the animal and sends them to the user/farmers to intimate them.

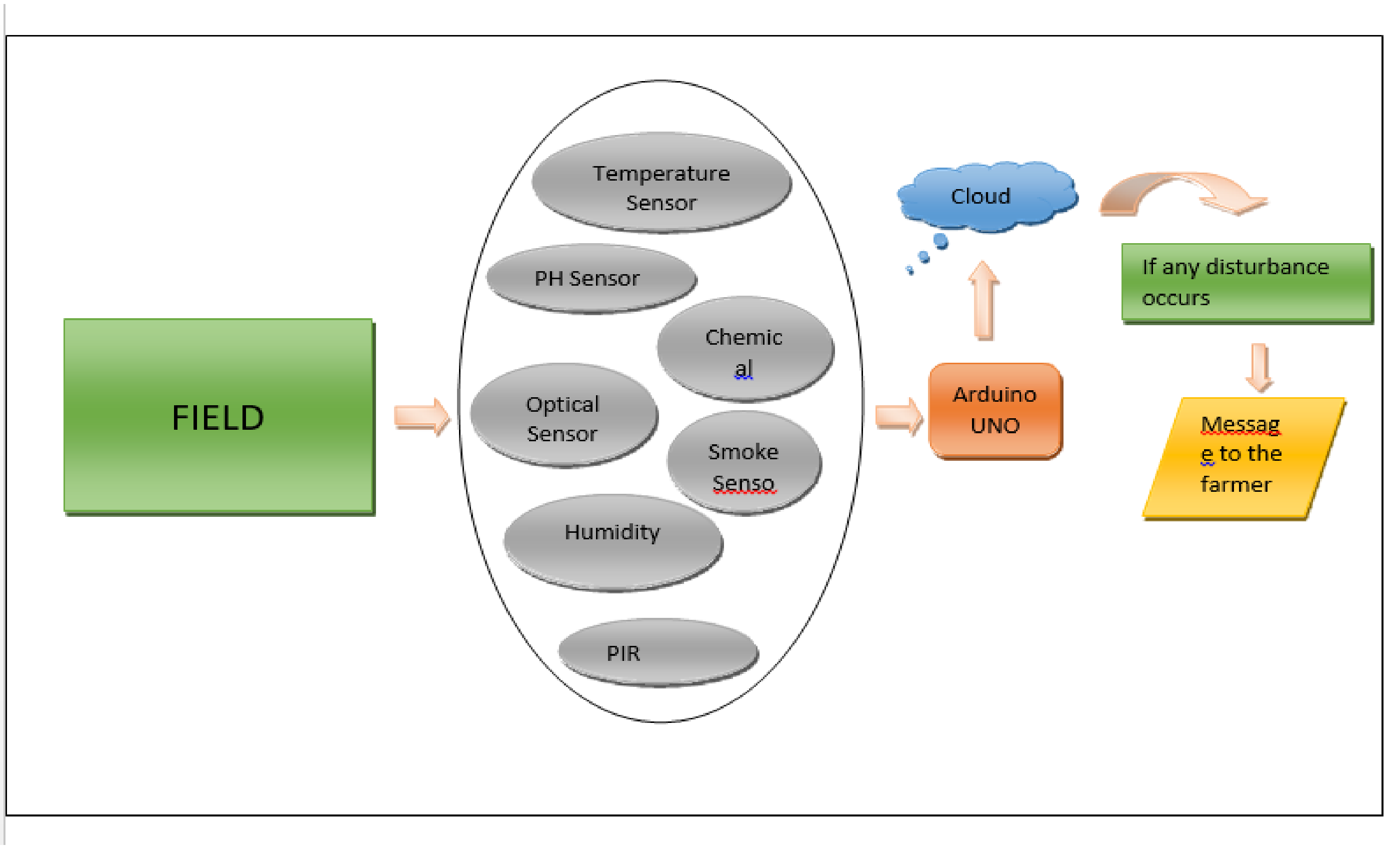
The other sensed datas such as the temperature, humidity, soil pH levels are updated every second by the cloud and sent to the user in the form of a graph.

If those levels exceed the normal range then a message intimation is sent to the user.

This helps the users to be able to solve the issue before it becomes a disaster and makes them at a loss. This also helps them to save time and also to take care of the crops more effectively.

PROJECT DESIGN:

Data flow diagram:



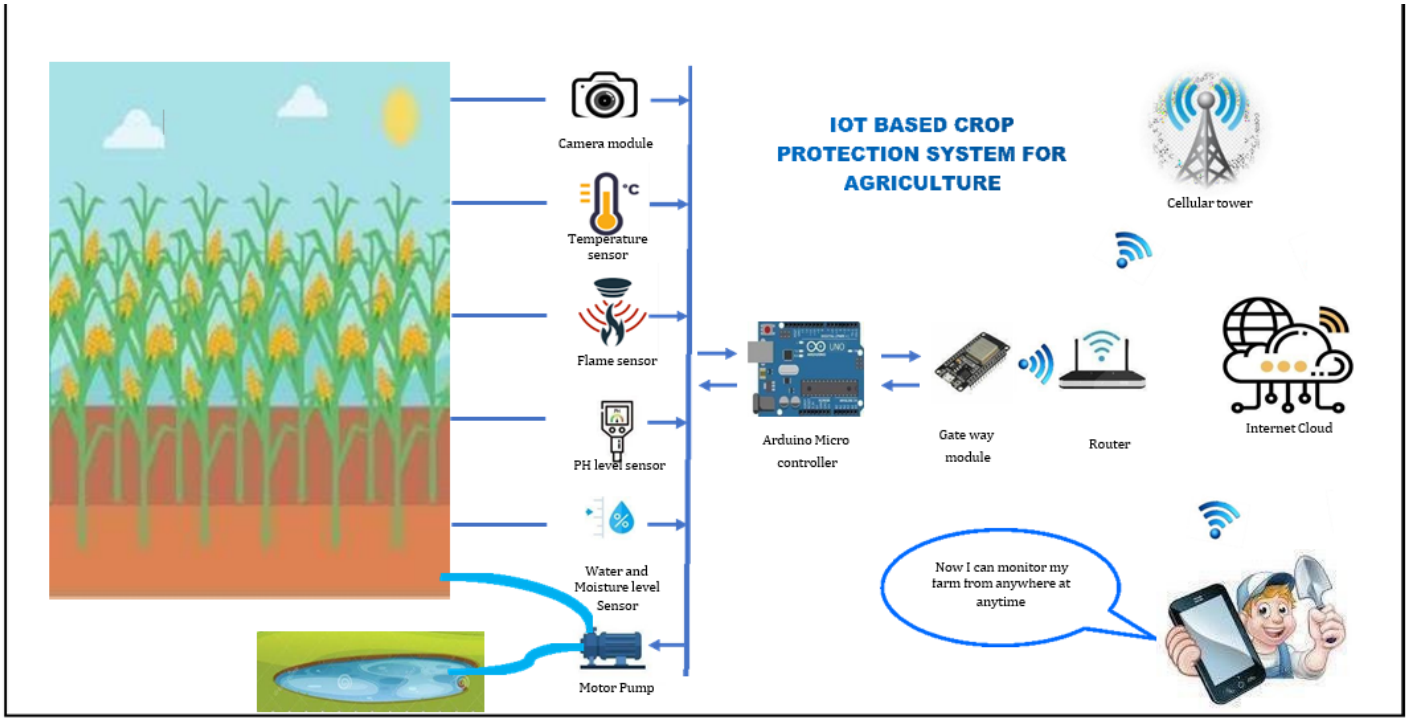
Solution architecture:

The different parameters for farming such as temperature, humidity, pH level, light intensity are sensed using different sensors and the obtained results are given in the form of graphs and the values are stored in the cloud.

Arduino UNO is used as the processing unit which processes the data obtained from the sensors and send them to the internet and the data's are saved and they are sent to the farmers for verification.

Node RED is used as the programming tool to wire the hardware, software and APIs. The MQTT protocol is used for communication.

All the collected data are sent to the user through the internet to their smartphones through the mobile application that was built specifically for this purpose. This application will be linked to their field 24/7 and the data will be updated frequently.



Technology architecture:

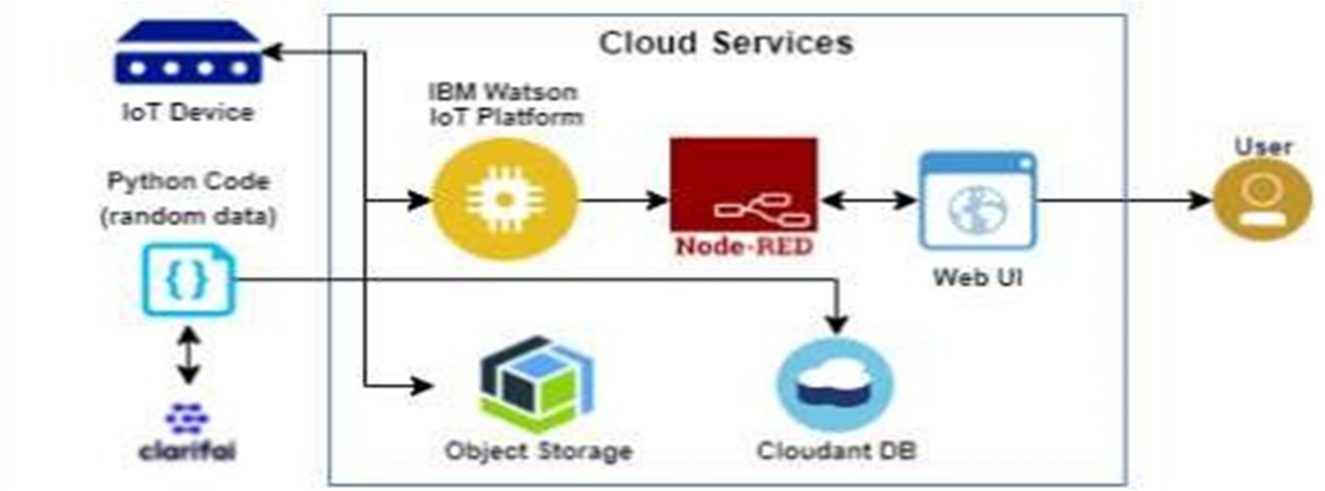


Table : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How the user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	In app development
2.	Application Logic-1	Logic for a process in the application	Python

3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	Influx DB,NoSQL
6.	Cloud Database	Database Service on Cloud	Cloudant.
7.	File Storage	File storage requirements	IBM Block storage
8.	External API-1	Purpose of External API used in the application	IBM Weather API
9.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local	Cloud Foundry

PROJECT PLANNING & SCHEDULING:

Sprint planning & estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (40)	Priority (Low to High)	Team Members
Sprint-1	Coding	USN-1	The python code for connecting the sensors with the system is created.	3	High	S.Soundarya
Sprint-1		USN-2	The code is tested for any form of errors and bugs and are rectified.	2	High	D.Nandhini
Sprint-2	Cloud services	USN-3	The python code is linked with the IOT Watson cloud platform services and to the Node RED platform.	1	Low	H.Sneka
Sprint-4		USN-4	The user will be able to login to the platform by using email and password and access data.	2	Medium	A.Sowmya
Sprint-3	Login	USN-5	The user must create a login to access the database of their field.	4	High	S.Soundarya
Sprint-2	Pre processing	USN-6	The access that is to be done by the farmer, so it must be easy to understand to them.	3	High	D.Nandhini

Sprint-1	Collecting Dataset	USN-7	To collect various sources of animal threats and keep developing a dataset.	3	Medium	A.Sowmya
Sprint-4	Integrating	USN-8	To integrate the available dataset and keep improving the accuracy of finding animals	2	High	D.Nandhini
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	H.Sneka
Sprint-2		USN-10	Testing the codes to find any interruptions and other factors and rectify them.	1	Low	A.Sowmya
Sprint-1	Training	USN-11	As programmer, we need to train our data perfectly so that the program runs smoothly	3	High	D.Nandhini
Sprint-3		USN-12	Train the data using out available services and IBM dataset from server and improve that	2	Medium	S.Soundarya

Sprint-4	Coding	USN-13	To modify the code according to our program and improve the efficiency of that code	4	High	D.Nandhini
Sprint-2		USN-13	Improving the performance by creating a reliable database and good infrastructure for easy access.	1	Low	S.Soundarya
Sprint-2	Record	USN-5	To record the data and plot the graph to show the characteristics officially	4	High	S.Soundarya
Sprint-1	Planning	USN-4	Plan the programming language and feasibility	3	Medium	D.Nandhini
Sprint-4		USN-14	Demonstrate the working and improve accuracy overall	2	Low	S.Soundarya

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	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

CODING & SOLUTION:

PYTHON CODE:

```
import cv2

import numpy as np

import wiotp.sdk.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, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

#This is how you authenticate
metadata = (('authorization', 'key 83ddcfb774c54cfd81d7a67ba69a0678'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "kn05el2QeCyawCFMRytUXLFirKVxw8v5HAIRvDKsIHmu"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/98d92dfd0ccf4f32a116d3d0fe24e15c:02d1fcad-1310-4403-93a6-a0eabc4c768b::"
clientdb = Cloudant("apikey-v2-d8mn8ful7bxv3pw2cq0o1p1d8z3icznh8qu8y2xsv5",
"400eef0a90d31fd7fa41c9dd0a2baa4b",
url="https://cbf0b64e-c2d3-4404-be21-36565dc150b9-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": "tw9ckq",
        "typeId": "node",
        "deviceId": "6020"
    },
    "auth": {
        "token": "27102001"
    }
}

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




database_name = "sample1"
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 = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    imS= cv2.resize(frame, (960,540))
    cv2.imwrite('ex.jpg',imS)
    with open("ex.jpg", "rb") as f:
        file_bytes = f.read()
    detect=False
    t=random.randint(-1,1)
    if(t==0):
        detect=True
        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('jadestorage', picname+'.jpg', picname+'.jpg')

json_document={"link":COS_ENDPOINT+'/'+'jadestorage'+'/'+picname+'.jpg'}
        new_document = my_database.create_document(json_document)


        if new_document.exists():
            print(f"Document successfully created.")
            time.sleep(5)


    moist=random.randint(0,100)
    humidity=random.randint(0,200)
    temperature=random.randint(0,100)

myData={'Animal':detect,'moisture':moist,'hum':humidity,'temp':temperature}
    print(myData)
    if(humidity!=None):
        client.publishEvent(eventId="status",msgFormat="json",
data=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()
```

Features:

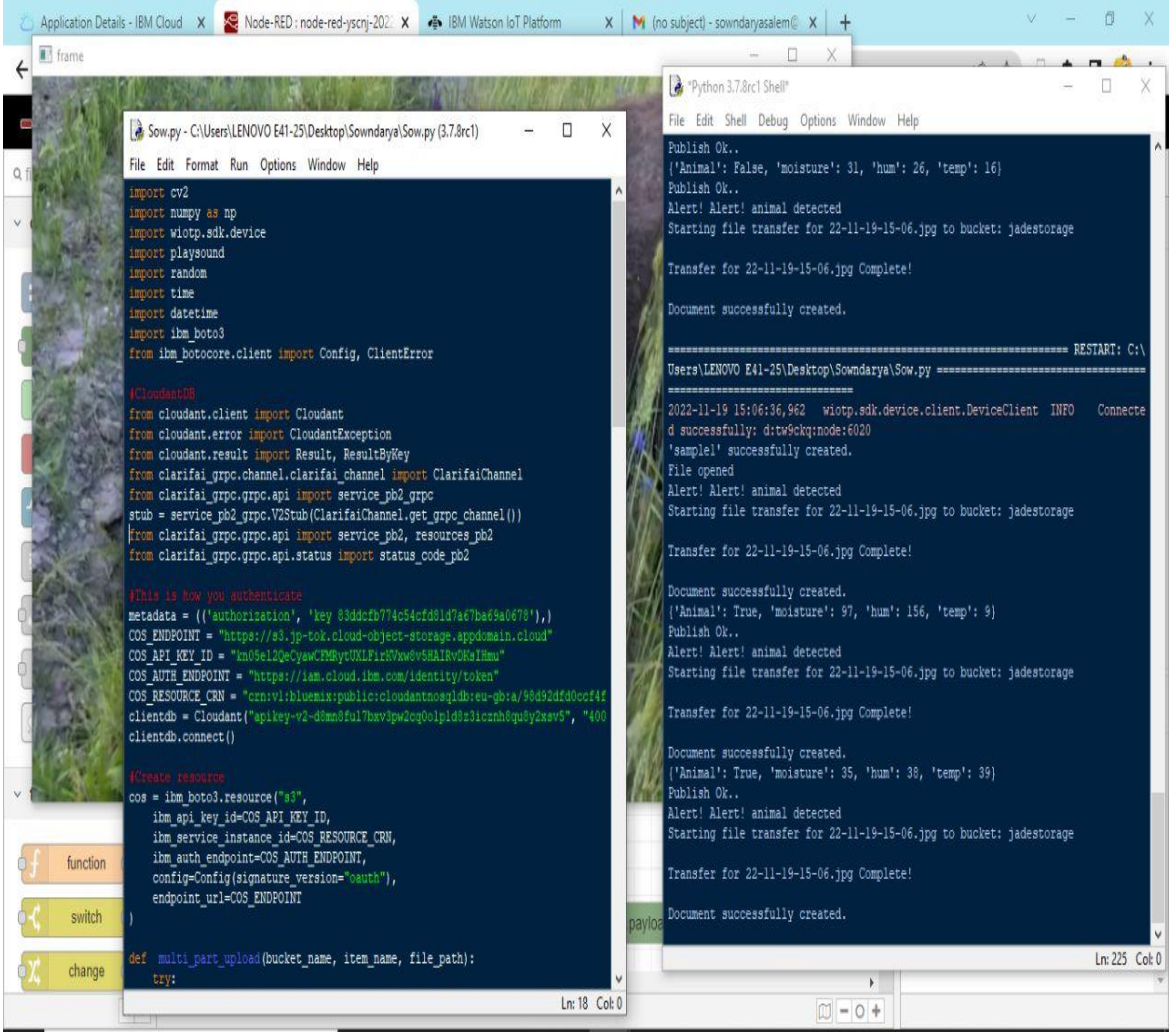
The special features used in this code are that it senses the datas such as the temperature, humidity, pH levels of the soil, water level of the soil, and detects the motion of the animals in the field.

The sensor senses all the animals entering the field and sends those datas were collected, it processes the datas and sends the information to the cloudant database.

The sensed datas is processed at Clarifi and processes the data and gives the output as an image. The image is sent to the user and the user is intimated about the intrusion.

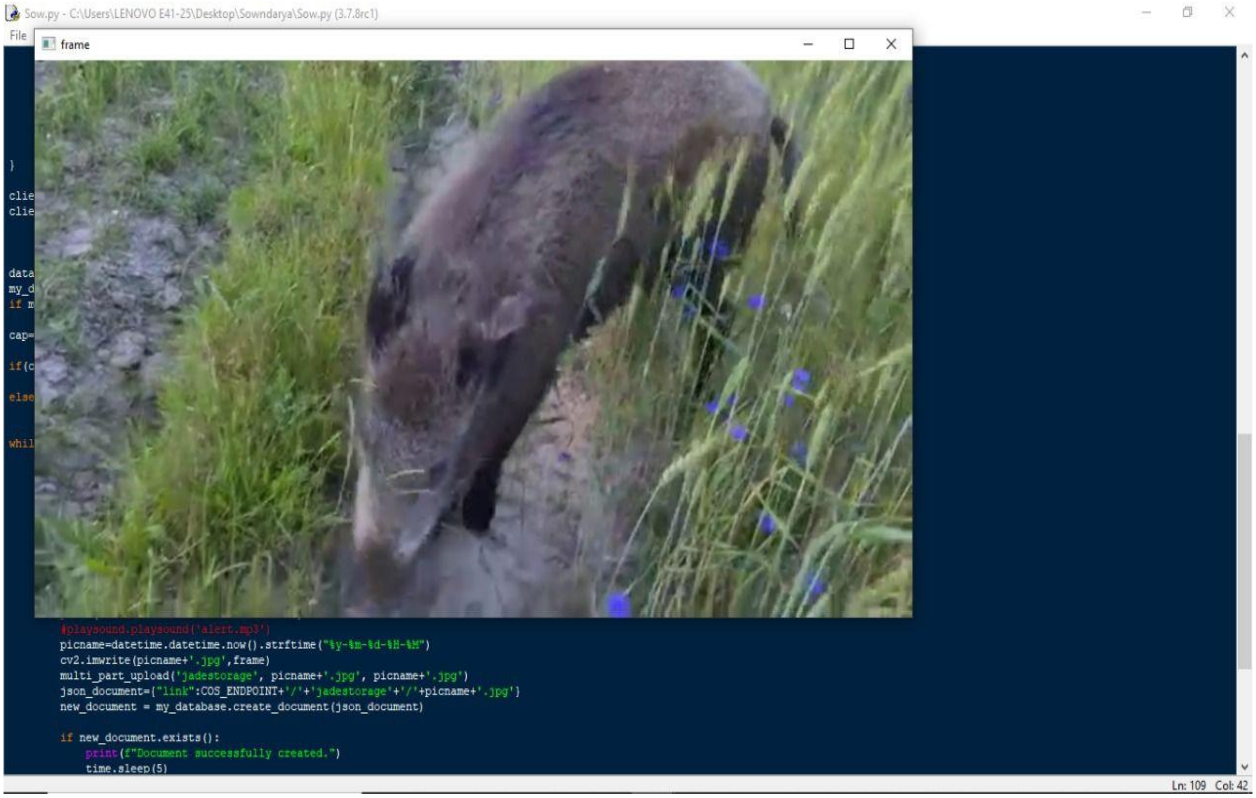
The datas from the different sensors such as the temperature, humidity, moisture, ph levels are updated frequently and if they exceed the normal values the user is intimidated by a message .

TESTING:

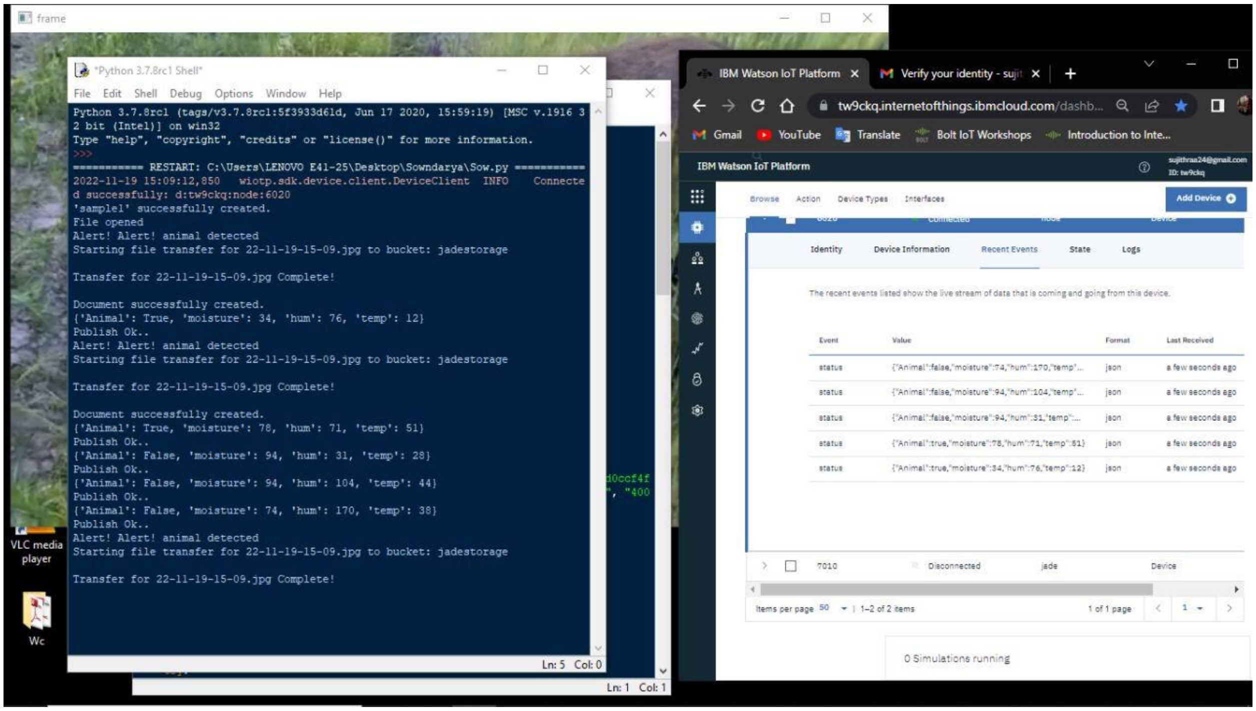

Application Details - IBM Cloud
Node-RED : node-red-yscnj-2022
IBM Watson IoT Platform
(no subject) - sowndaryasalem@
frame
Sow.py - C:\Users\LENOVO E41-25\Desktop\Sowndarya\Sow.py (3.7.8rc1)
File Edit Format Run Options Window Help
Python 3.7.8rc1 Shell
File Edit Shell Debug Options Window Help
function
switch
change
payloa
Ln: 18 Col: 0
Ln: 225 Col: 0

```
import cv2
import numpy as np
import wiotp.sdk.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, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

#This is how you authenticate
metadata = (('authorization', 'key 83ddcfb774c54cfd81d7a67ba69a0678'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "kn05e12QeCyawCFMRytUXLFirKVxw8v5HAIRvDKsIHmu"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/98d92dfd0ccf4f
clientdb = Cloudant("apikey-v2-d8mn8ful7bxv3pw2cq0olpld8z3icznh8qu8y2xsv5", "400
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:
```

```
Publish Ok..
{'Animal': False, 'moisture': 31, 'hum': 26, 'temp': 16}
Publish Ok..
Alert! Alert! animal detected
Starting file transfer for 22-11-19-15-06.jpg to bucket: jadestorage

Transfer for 22-11-19-15-06.jpg Complete!

Document successfully created.

================================================================ RESTART: C:\
Users\LENOVO E41-25\Desktop\Sowndarya\Sow.py ================================
==============================
2022-11-19 15:06:36,962   wiotp.sdk.device.client.DeviceClient  INFO    Connecte
d successfully: d:tw9ckq:node:6020
'sample1' successfully created.
File opened
Alert! Alert! animal detected
Starting file transfer for 22-11-19-15-06.jpg to bucket: jadestorage

Transfer for 22-11-19-15-06.jpg Complete!

Document successfully created.
{'Animal': True, 'moisture': 97, 'hum': 156, 'temp': 9}
Publish Ok..
Alert! Alert! animal detected
Starting file transfer for 22-11-19-15-06.jpg to bucket: jadestorage

Transfer for 22-11-19-15-06.jpg Complete!

Document successfully created.
{'Animal': True, 'moisture': 35, 'hum': 38, 'temp': 39}
Publish Ok..
Alert! Alert! animal detected
Starting file transfer for 22-11-19-15-06.jpg to bucket: jadestorage

Transfer for 22-11-19-15-06.jpg Complete!

Document successfully created.
```


Sow.py - C:\Users\LENOVO E41-25\Desktop\Sowndarya\Sow.py (3.7.8rc1)
File
frame
Ln: 109 Col: 42

```
    #playsound.playsound('alert.mp3')
    picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
    cv2.imwrite(picname+'.jpg',frame)
    multi_part_upload('jadestorage', picname+'.jpg', picname+'.jpg')
    json_document={"link":COS_ENDPOINT+'/'+'jadestorage'+'/'+picname+'.jpg'}
    new_document = my_database.create_document(json_document)

    if new_document.exists():
        print(f"Document successfully created.")
        time.sleep(5)
```



```
Python 3.7.8rc1 (tags/v3.7.8rc1:5f3933d61d, Jun 17 2020, 15:59:19) [MSC v.1916 3
2 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: C:\Users\LENOVO E41-25\Desktop\Sowndarya\Sow.py ===========
2022-11-19 15:09:12,850   wiotp.sdk.device.client.DeviceClient  INFO    Connecte
d successfully: d:tw9ckq:node:6020
'sample1' successfully created.
File opened
Alert! Alert! animal detected
Starting file transfer for 22-11-19-15-09.jpg to bucket: jadestorage

Transfer for 22-11-19-15-09.jpg Complete!

Document successfully created.
{'Animal': True, 'moisture': 34, 'hum': 76, 'temp': 12}
Publish Ok..
Alert! Alert! animal detected
Starting file transfer for 22-11-19-15-09.jpg to bucket: jadestorage

Transfer for 22-11-19-15-09.jpg Complete!

Document successfully created.
{'Animal': True, 'moisture': 78, 'hum': 71, 'temp': 51}
Publish Ok..
{'Animal': False, 'moisture': 94, 'hum': 31, 'temp': 28}
Publish Ok..
{'Animal': False, 'moisture': 94, 'hum': 104, 'temp': 44}
Publish Ok..
{'Animal': False, 'moisture': 74, 'hum': 170, 'temp': 38}
Publish Ok..
Alert! Alert! animal detected
Starting file transfer for 22-11-19-15-09.jpg to bucket: jadestorage

Transfer for 22-11-19-15-09.jpg Complete!
```

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

Event	Value	Format	Last Received
status	{"Animal":false,"moisture":74,"hum":170,"temp"...	json	a few seconds ago
status	{"Animal":false,"moisture":94,"hum":104,"temp"...	json	a few seconds ago
status	{"Animal":false,"moisture":94,"hum":31,"temp":...	json	a few seconds ago
status	{"Animal":true,"moisture":78,"hum":71,"temp":51}	json	a few seconds ago
status	{"Animal":true,"moisture":34,"hum":76,"temp":12}	json	a few seconds ago

7010	Disconnected	jade	Device

RESULTS:

The results for the above testing process are given and the output obtained are the outputs that we have aimed for getting from the start. As a result of this system, we can detect the changes in the field easily and intimate the farmers about it and also we can take precautions and do remedies accordingly. Here we use very low power consuming highly efficient components that give us accurate results and also they perform at low data rate conditions without any lag and help in finding the remedies. This crop protection system helps in detection of all kinds of external dangers and it saves time and money to the farmers before any loss that may occur. With the help of this system the farmers can be in a peaceful environment at ease without any pressure.

ADVANTAGES & DISADVANTAGES:

Advantages:

The main advantage of this design is that it helps the farmers to take care of their crops .

As it gives information immediately, it is easy to take necessary actions immediately.

The given information is of great use and that information at the correct time saves the field and the farmers.

This information reduces losses to the farmers and it helps in maintaining the field in a healthy manner.

Disadvantages:

As the sensors are very sensitive in nature, there are many possibilities of error occurrences or contaminations in the reports provided.

The location of the sensors should be in a place where it cannot be contaminated.

There is the use of internet connectivity. So there should be a good network connection at any time and also there should not be any power shortage as power shortage may cause the sensor to stop working.

The sensors used should be of good quality, if not it will give wrong information to the users.

CONCLUSION:

This project helps the farmers to take care of their fields from the animals and it helps them to maintain a good amount of crops in their fields. The devices used here require very little but uninterrupted power supply with uninterrupted network connectivity. This is easily accessible by the farmers and they are kept updated about their field every instant. It helps them to reduce loss in their fields and in their lives.

FUTURE SCOPE:

This will be used by the farmers as they give the processed images of the animals that enter in their fields and intimates them and also they give the temperature, humidity and all the other values that are needed for a good farming practice.

It is highly easy to use and is very user friendly. The need for power and network are very low and as a result of this we can expect various types of farmers to use this in the near future as they give us the needed information that helps in various forms. This information is the mandatory need for a farmer to be known at any instant.

APPENDIX:

Source code:

```
import cv2
import numpy as np
import wiotp.sdk.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, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2


#This is how you authenticate
metadata = (('authorization', 'key 83ddcfb774c54cfd81d7a67ba69a0678'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "kn05el2QeCyawCFMRytUXLFirKVxw8v5HAIRvDKsIHmu"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/98d92dfd0ccf4f32a116d3d0fe24e15c:02d1fcad-1310-4403-93a6-a0eabc4c768b::"
clientdb = Cloudant("apikey-v2-d8mn8ful7bxv3pw2cq0o1p1d8z3icznh8qu8y2xsv5", "400eef0a90d31fd7fa41c9dd0a2baa4b", url="https://cbf0b64e-c2d3-4404-be21-36565dc150b9-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": "tw9ckq",
        "typeId": "node",
        "deviceId": "6020"
    },
    "auth": {
        "token": "27102001"
    }
}

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



database_name = "sample1"
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 = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    imS= cv2.resize(frame, (960,540))
    cv2.imwrite('ex.jpg',imS)
    with open("ex.jpg", "rb") as f:
        file_bytes = f.read()
    detect=False
    t=random.randint(-1,1)
    if(t==0):
        detect=True
        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('jadestorage', picname+'.jpg', picname+'.jpg')
```

```
json_document={"link":COS_ENDPOINT+'/'+'jadestorage'+'/'+picname+'.jp
g'}
        new_document = my_database.create_document(json_document)

        if new_document.exists():
            print(f"Document successfully created.")
            time.sleep(5)

    moist=random.randint(0,100)
    humidity=random.randint(0,200)
    temperature=random.randint(0,100)

myData={'Animal':detect,'moisture':moist,'hum':humidity,'temp':temperat
ure}
    print(myData)
    if(humidity!=None):
        client.publishEvent(eventId="status",msgFormat="json",
data=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()
```

GitHub link :

https://github.com/IBM-EPBL/IBM-Project-28575-1660113919

Project demo link:

https://drive.google.com/file/d/18kNPGX7fCg5qxUY5KjMGebCVgVTbjeEF/view?usp=drivesdk