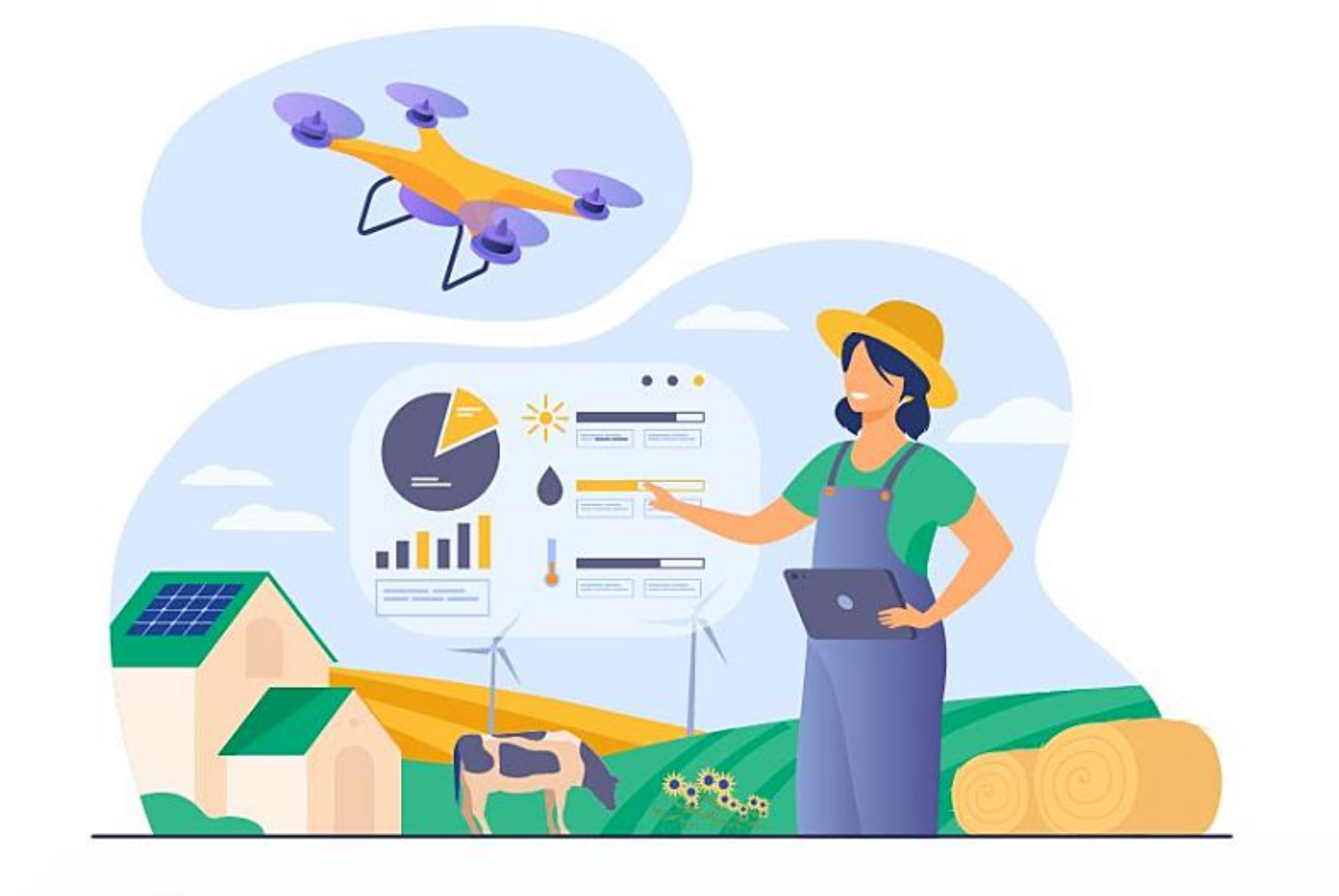
SmartFarmer - IoT Enabled Smart Farming Application

TEAM ID: PNT2022TMID40121



TEAM MEMBERS:

Deepika J
Janani J
Poornima Pooja S
Siva Shankari K

Content

1. INTRODUCTION

Even today, different developing countries are also using traditional methods and backward techniques in agriculture sector. Smart Farming has enabled farmers to reduce waste and enhance productivity with the help of sensors (humidity, temperature, soil moisture, etc.) and automation of irrigation systems. Further with the help of these sensors, farmers can monitor the field conditions from anywhere. Internet of Things based Advanced Farming is highly efficient when compared with the conventional approach. The applications of intelligent Agriculture solutions not only targets conventional, large farming. With operations, but could also be new levers to uplift other growing or common trends in agricultural like organic farming, family farming (complex or small spaces, particular cattle and/or cultures, preservation of specific or high-quality varieties, etc.), and enhance highly transparent Farming.

1.1 Project Overview

In Internet of Things based smart agriculture, a system is formed to monitor the farmland with the help of sensors, which senses components like temperature, light, humidity, soil moisture, etc. Then, automate the irrigation system and allow farmers to monitor their field conditions from anywhere through IoT Analytics Platform. To make the agricultural process even smarter and accurate, precision agriculture is used. This makes agricultural practice more controlled and precise in terms of raising livestock and farming. Internet of Things based Advanced Farming plays a vital role when it comes to the use of IT and other elements like sensors, agricultural drones, autonomous vehicles, control systems, automated hardware, robotics, variable speed technology, and others.

The below highlighted are the applications of Internet of Things in smart farming:

- ***Weather Monitoring:***

Weather plays a very significant role when it comes to the Agriculture sector. In agriculture, there is almost everything dependable upon the climate condition. In smart Farming, temperature humidity, light intensity, and soil moisture can be monitored through various sensors. These are again used by the reactive system to trigger alerts or automate the process such as water and air control.

- ***Smart Irrigation on Agriculture Land:***

In smart irrigation, automated sprinkler systems or intelligent pumps are used. Soil moistures sensors are used in different areas to get the moisture of the soil in agricultural land. Based on the results from the soil moisture sensors, theintelligent pumps or intelligent sprinklers are turned On/Off.

⭐ ***Monitoring Soil Quality:***

Farmers usually use a sampling method to calculate soil fertility, moisture content. Fortunately, this sampling doesn't give accurate results as chemical decomposition varies from location to location. Meanwhile, this not much helpful. To resolve this thing, it plays an essential role in Farming. Sensors can be installed at a uniform distance across the length and breadth of the farmland to collect the accurate soil data, which can be further used in the dashboard or mobile application for the farm monitoring.

1.2 Purpose

Little or very less technological advancement is found here that has increased the production efficiency significantly. To increase the productivity, a novel design approach is presented in this paper. Smart farming with the help of Internet of Things (IOT) has been designed. The controller keeps monitoring the temperature, humidity, soil condition and accordingly supplies water to the fiel

- They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.

Smart farming is a management concept focused on providing the agricultural industry with the infrastructure to leverage advanced technology – including big data, the cloud and the internet of things (IoT) – for tracking, monitoring, automating and analyzing operations.

2.LITERATURE SURVEY

A literature survey on smart farming can provide an overview of the latest research and developments in the field of smart farming technology. It can also identify gaps in the current knowledge and suggest areas for future research.

2.1 Existing problem

Agriculture is the back bone of our nation. In olden days farmers used to guess the fertility of soil and made assumption to grow which type of crop. They didn't know about the moisture, level of water and particularly weather condition which terrible a farmer more. They use pesticides based on some assumption which made lead a serious effect to the crop if the assumption is wrong. The productivity depends on the final stage of the crop on which farmer depends. Farmers need to deal with many problems, including how to: Cope with climate change, soil erosion and biodiversity loss.

2.2 References

[1].<u>Smart Farming using IOT</u>: Amandeep, Arshia Bhattacharjee, Pabonic Das, Debjit Bass, Somudit Roy, Spandan Ghosh, Sayan Saha, Sowvik Pain, Sourav Daj, T.K.Rana- Nov - 2017

A remote controlled vehicle operates on both automatic and manual modes, for various agriculture operations like spraying, cutting. The controller keeps monitoring the temperature, humidity, soil condition.

Link: **https://ieeexplore.ieee.org/document/8117219**

[2].<u>Smart Agriculture Using Internet of Things with Raspberry Pi:</u> Zuraida Muhammad, Muhammad Azri Asyraf Mohd Hafez, Nor Adni Mat Leh, Zakiah Mohd Yusoff, Shabinar Abd Hamid- Sept - 2020

The purpose of this Smart Agriculture with IoT using Raspberry Pi is to find the suitable system to be applied in future agriculture system. The aims of this project are to reduce time and water consumption as well as maximizing agriculture product and to improve the efficiency of management and control for agriculture farm.

Link: **https://ieeexplore.ieee.org/document/9204927**

[3].<u>Smart Irrigation Based On Crops Using IOT:</u>Shyam Peraka, Reddy Sudheer, Bandi Narashmha Rao, Allu Ravi, Esai Naveen Kumar- Feb - 2021

Proposed the irrigation system, which based on the latest IOT technology to reduce the wastage of water and it reduced manual labor to irrigate the crops based on the type and stage of the crop.

Link: **https://ieeexplore.ieee.org/document/9342736**

[4].<u>AgriSegNet: Deep Aerial Semantic Segmentation Framework for IoTassisted Precision Agriculture</u>: Tanmay Anand, Soumendu Sinha, Murari Mandal, Vinay Chamola, Fei Richard Yu.- Apr - 2021

In this work, a deep learning framework AgriSegNet for automatic detection of farmland anomalies using multiscale attention semantic segmentation of UAV acquired images. The proposed model is useful for monitoring of farmland and crops to increase the efficiency of precision farming techniques.

Link: **https://ieeexplore.ieee.org/document/9395478**

[5].<u>Sensor Based Smart Agriculture with IoT Technologies:</u> Pyingkodi M, Thenmozhi K, Nanthini K, Karthikeyan M, Suresh Palarimath, Erajavignesh V, Bala Ajith Kumar G. -Mar - 2022

This paper narrates the role of IoT application in smart agriculture. Smart farming is a precision farming hence, it uses accurate information to draw outcomes. It demonstrates the different sensors, applications, challenges, strength and weaknesses that support the IoT and agriculture.

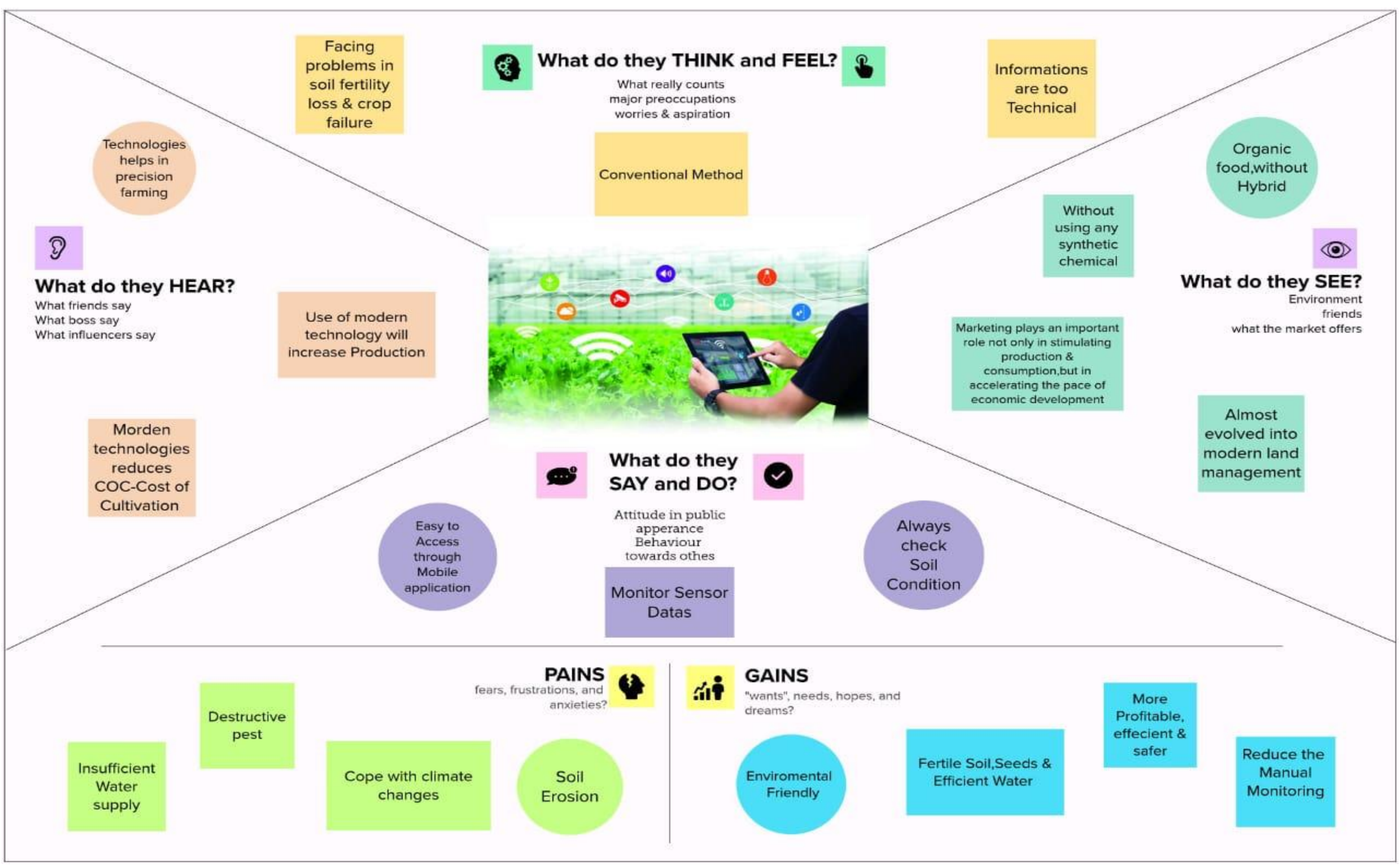
Link: **https://ieeexplore.ieee.o rg/document/9741001**

2.3 Problem Statement Definition

By Using different types of Sensors we collected the data & monitor the feld continuously, and automatically pour water in the crops & Manual requirement is not necessary Farmer can easily known in their situated place.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming


Brainstorm
10 minutes
Poornima Pooja S
ESP232 Microprocessor is used
Current datas are frequently updated & monitored by Mobille Application
The accurate soil moisture level will be noted
Using IoT core the Manual work is decreased
Watson IoT Platform designed for allowing user to connect, visualize & analyze their Sensor data over the cloud
Siva Shankari K
DHT22 & Soil Moisture Sensor
Use Open weather map to get the Global Weather Data
Node - Red dashboard is connected to Moible Application to get Sensor Values
Online simulation will be done by Wokwi platform
Datas will be stored in Cloud Server
Deepika J
Crops are monitored regularly
L293D Motor Driver is used
Capacitive soil Moisture sensor used to Test the Moisture level & check the capacity of soil
It can notify the problems that occur in the field
It can be easy to use by control the device in any area
Janani J
Automatic water pouring method is used
Proper alert has been used
Relay is used to act as a Electrical switch
MQTT Websocket Server is used to receive MQTT data directly into a web browser
Capacitive soil moisture sensor is also used for checking the water level in water tank

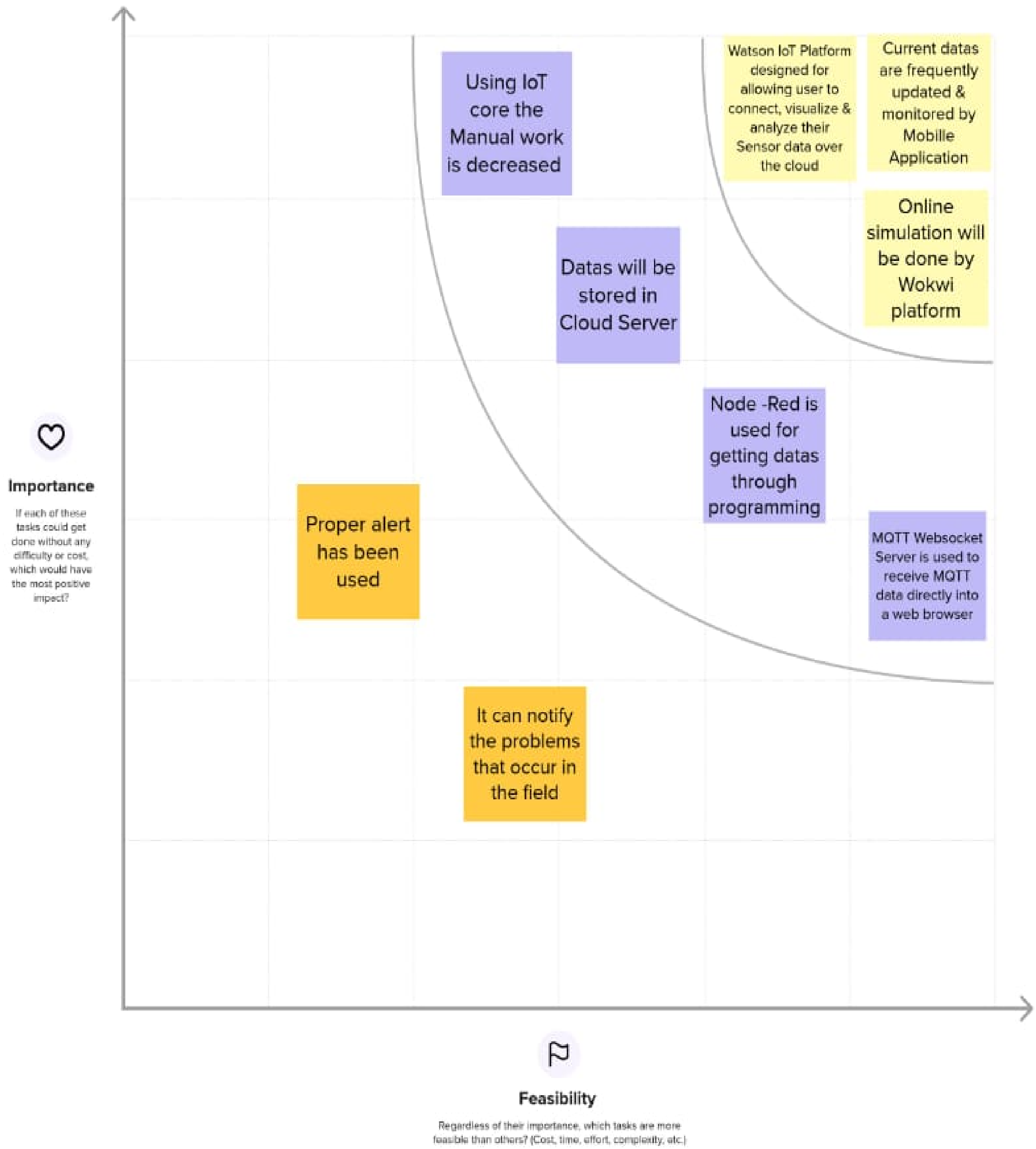
Group ideas

20 minutes

- ESP232 Microprocessor is used
- Current Datas are Frequently Updated
- DHT22 & Soil Moisture Sensor is ued
- Node -Red is used for getting datas through programming
- In Wokwi platform, Online simulation will be done
- Watson IoT Platform is used for analyze, visualize the data
- It can be easy to use by control the device in any area
- L293D Motor Driver is used
- Crops are monitored regularly
- The accurate datas will be get
- Proper alert should be given
- Suitable for all types of soil
- User - Friendly
- Using IoT Core the manual work will be decreased
- Open weather map use to get Weather data
- Here soil sensor is used to Test the soil Capacity along with sensing moisture level
- Data should be store in Cloud
- MQTT Websocket is used to receive MQTT data directly into a web browser

Prioritize

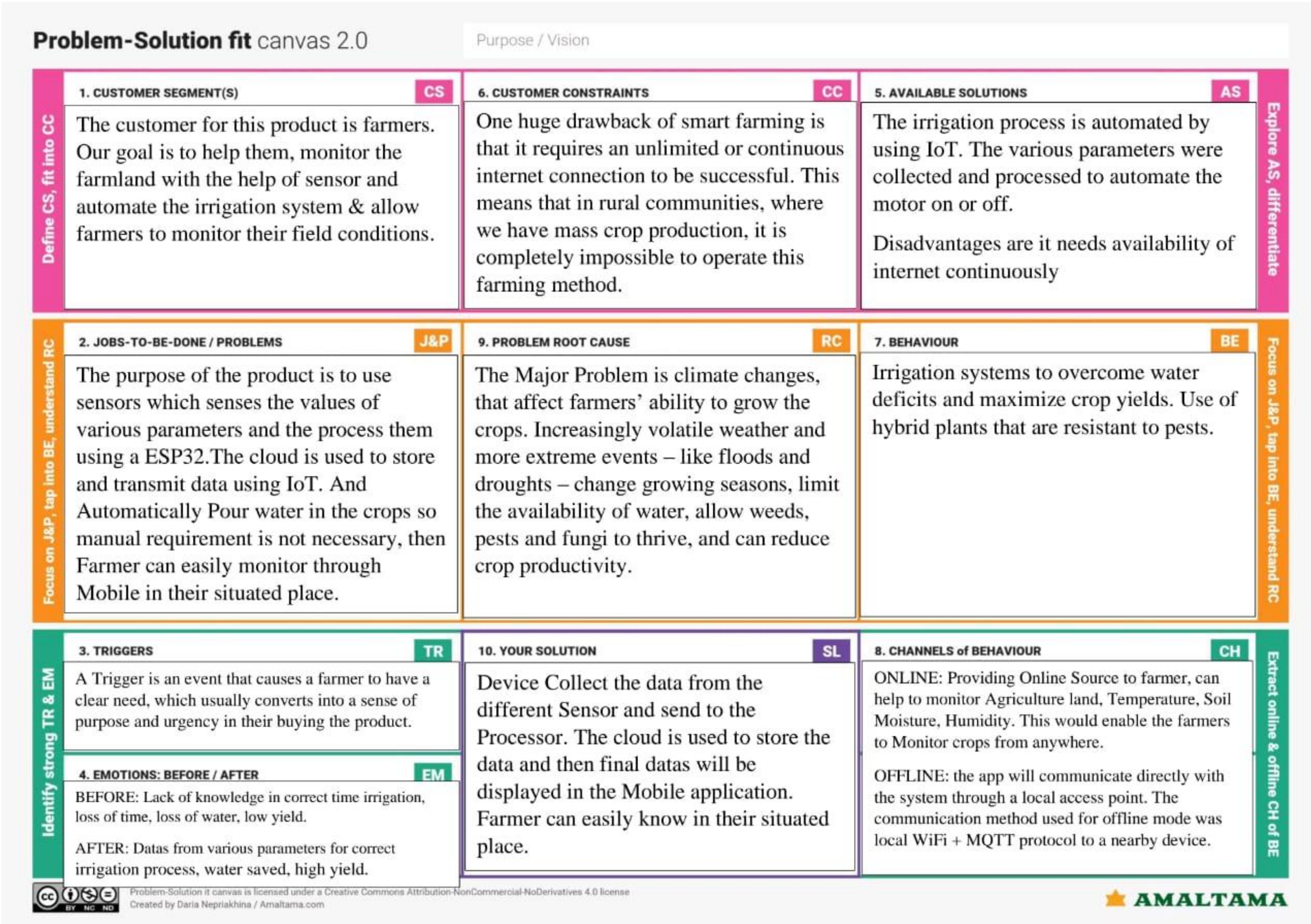
20 minutes



3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Farmers are under pressure to produce more food & use less energy and water in the process. A remote monitoring and control system will help farmers deal effectively with these pressures. • IoT applications may assist in controlling the irrigation pump, opening and closing water flowing gates and also data logging the soil health conditions for present and future purpose, each field should get just the right amount of water at just the right time. • Smart Farm's systems can be retrofitted on existing sites and provide immediate impact with a very short return on investment time period.
2.	Idea / Solution description	It collects the data from different types of sensors and it sends the value to the Cloud server. It also collects the weather data from the Open weather Map. Watering to plants in the right period of time automatically.
3.	Novelty / Uniqueness	Various eminent researchers have been making efforts for smart farming by using IoT concepts in agriculture. Smart Farming based IoT technologies to reduce waste and enhance productivity.
4.	Social Impact / Customer Satisfaction	• Increased production, the optimisation of all the processes related to agriculture increases production with limited lose. • Water saving , weather forecasts and sensors that measure soil moisture means watering only when necessary and for the right length of time.
5.	Business Model (Revenue Model)	• IoT can connect every unit, device, asset, machinery or equipment to a single network. With smart sensors, businesses can then track assets and control equipment . • A popular IoT business model is the data-driven model powered by the data generated by your devices, that provide value to customers. • Even small farmers can gather dat from various sources ,which helps them with decision making that will help lower cost and increase yields.
6.	Scalability of the Solution	• Smart farming can make agriculture more profitable for the farmers. Decreasing resources input will save the farmer money and labor and increased reliability of spatially explicit data will reduce risks. • The goals is to increase the quality and quantity of crops while decreasing the human labor used for such purposes.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

Agriculture is considered as one of the major sources in maintaining a nation's GDP. Most of the developing countries and under developed countries are relying on cultivation to improve their economic wealth. In this modern technology era, technology can play a tremendous role in the agriculture sector.
The advanced technology has the capability to automate various cultivation phases like watering, fertilizing, harvesting and much more.

We analyzed the various standard IoT techniques used in Agriculture sector based on hardware and software, and thereby deriving the existing challenges for making farming much smarter and efficient.

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

S.No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1.	Objective	Automatic Irrigation System is used to irrigate the land without the help of manpower. It works by using various sensors which senses the values of various parameters and the process them using ESP32.The cloud is used to store and transmit data using IoT. The field need water then automatically motor will get ON and it will get OFF when it's get enough.
2.	Product Features	The smart irrigation system is self-explanatory and user friendly plant watering system. In this system, ESP32 is compatible to operate the hardware module. For the monitoring and controlling the water pump and, the multiple sensors are used. This system is used for monitoring and controlling water pump with the help of software applications. Wi-Fi connectivity and android app are provided for field testing. This smart system has software to view a sensor’s real time graph analysis on mobile.
3.	User Requirement	The users of this system will require a simple yet effective and fully functioning automatic system. The mobile application should be simple and easy to use allowing users to navigate all of the sensor data with ease. Users will expect a system that once installed will actually work and water their plants when needed. This is the main requirement for users as the project will be a failure if it does not water the plants when they need to be watered.
4.	Use Case	The data that will be collected from the sensors and sent to the cloud. These readings will then be displayed on the mobile app in a clear and easy to read manner. This use case describes the communication between the mobile app and the user. The application will mainly be used for displaying the sensor data in a user-friendly manner. The user will be given notifications with certain sensor data readings.

4.2 Non-Functional requirements

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

S.No	Non-Functional Requirement	Description
1.	Usability	By using automatic irrigation system we are using various equipment like temperature sensor, humidity sensor, and soil moisture sensor. These sensors will find the various situations of the soil and based on soil moisture percent, land gets automatically irrigated, when field needs water then automatically motor will get ON and it will get OFF when it's get enough. These sensed parameters and motor status will be displayed on user devices.
2.	Security	There is no deploying of this application and the only instance of the app will be the single local android app that we used for developing.
3.	Reliability	Thankfully due to the robustness of the hardware and the simple nature of the software, the project is unlikely to fail. There may be some cable damage over time but that would be a simple fix.
4.	Performance	The project should update the application with the relevant information quickly but there is no need for extremely fast data transfer. Cloud MQTT data transfers seem nearly instantaneous to the user so this should be sufficient.
5.	Availability	The data gathered from the sensors and transfer the data to the cloud will be available to the project and the application as long as the Python code is running.
6.	Scalability	Scalability is a major concern for IoT platforms. It has been shown that different architectural choices of IoT platforms affect system scalability and that automatic irrigation system.

5. PROJECT DESIGN

Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

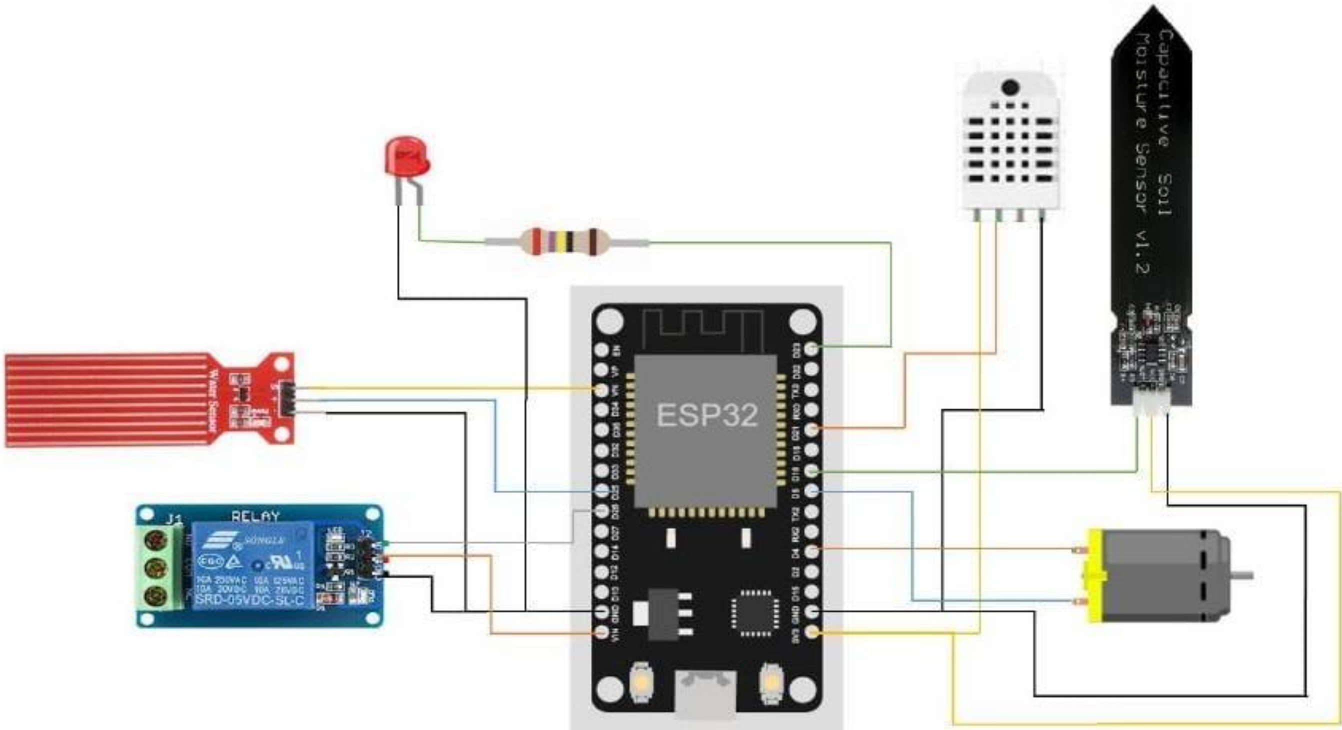


Fig: Block Diagram

5.1 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. Here is the DFD for the smart

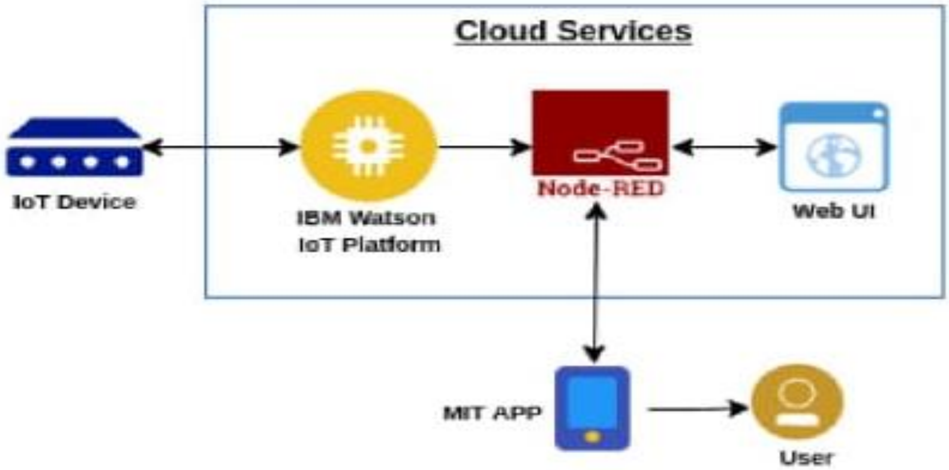


Fig: Simplified Data flow Diagram

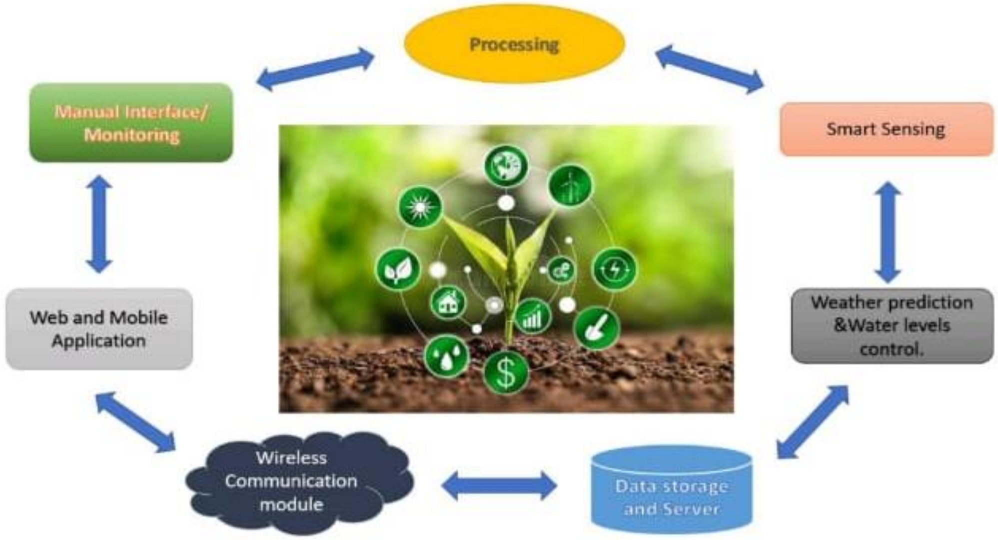


Fig: Data Flow Diagram

5.2 Solution & Technical Architecture

<u>Solution Architecture:</u>

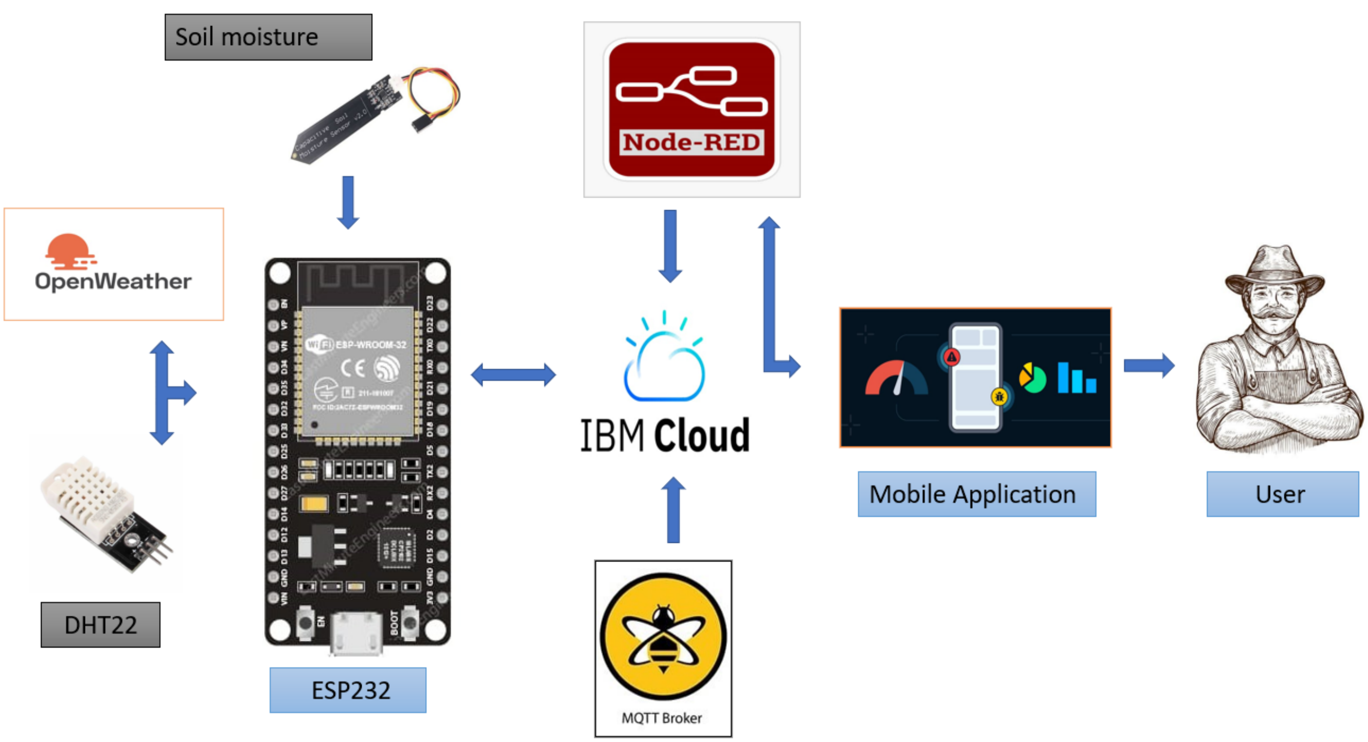


Fig: Solution Architecture

Technical Architecture:

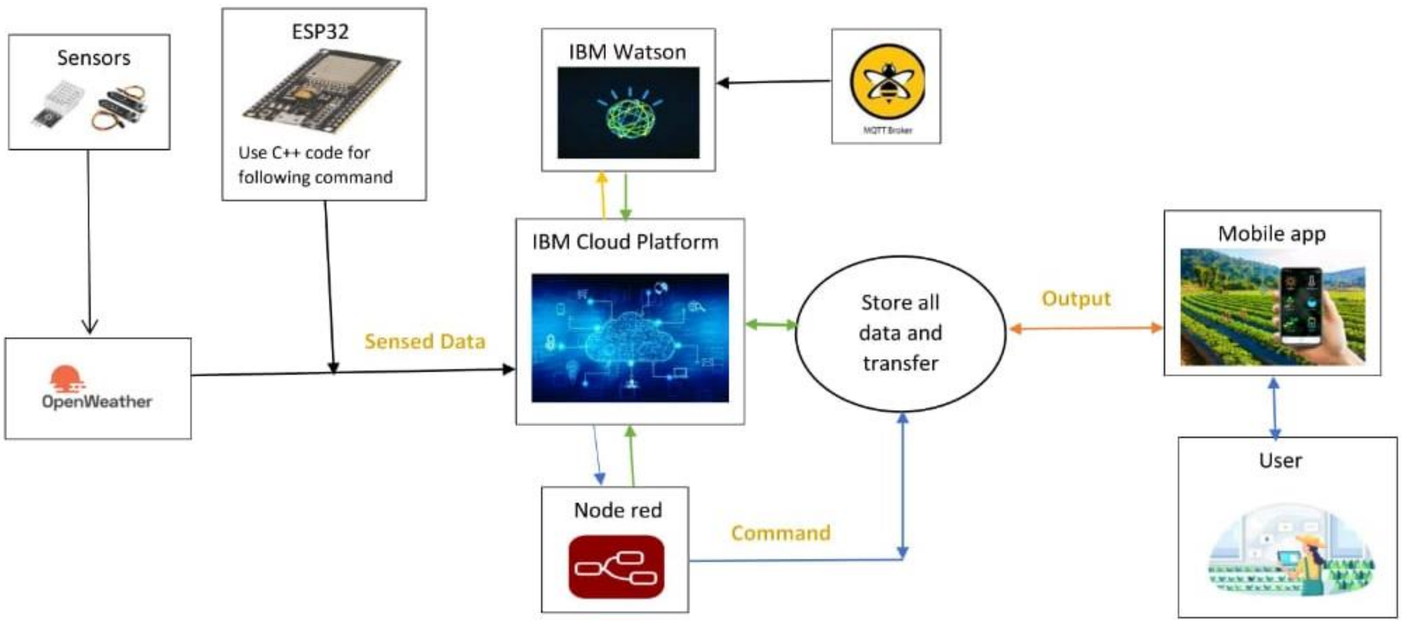


Fig: Technical Architecture

• The different sensors (temperature, humidity, soil moisture) are sensed through the ESP232 processor kit and the weather API.

• ESP232 is used as a processing unit that processes the data obtained from sensors and weather data from weather API.

• Node-red is used for programming, as the tool to wire the hardware, software and APIs. The MQTT protocol is followed as a mediator.

• The overall data given by ESP232, open weather, node-red are stored in the IBM cloud.

• The data which was collected are provided to the user through a mobile application that was developed using the MIT app invertor.

• The water irrigation to plants is automatically done in the right period of time by the instructions which was already programmed.

5.3 User Stories

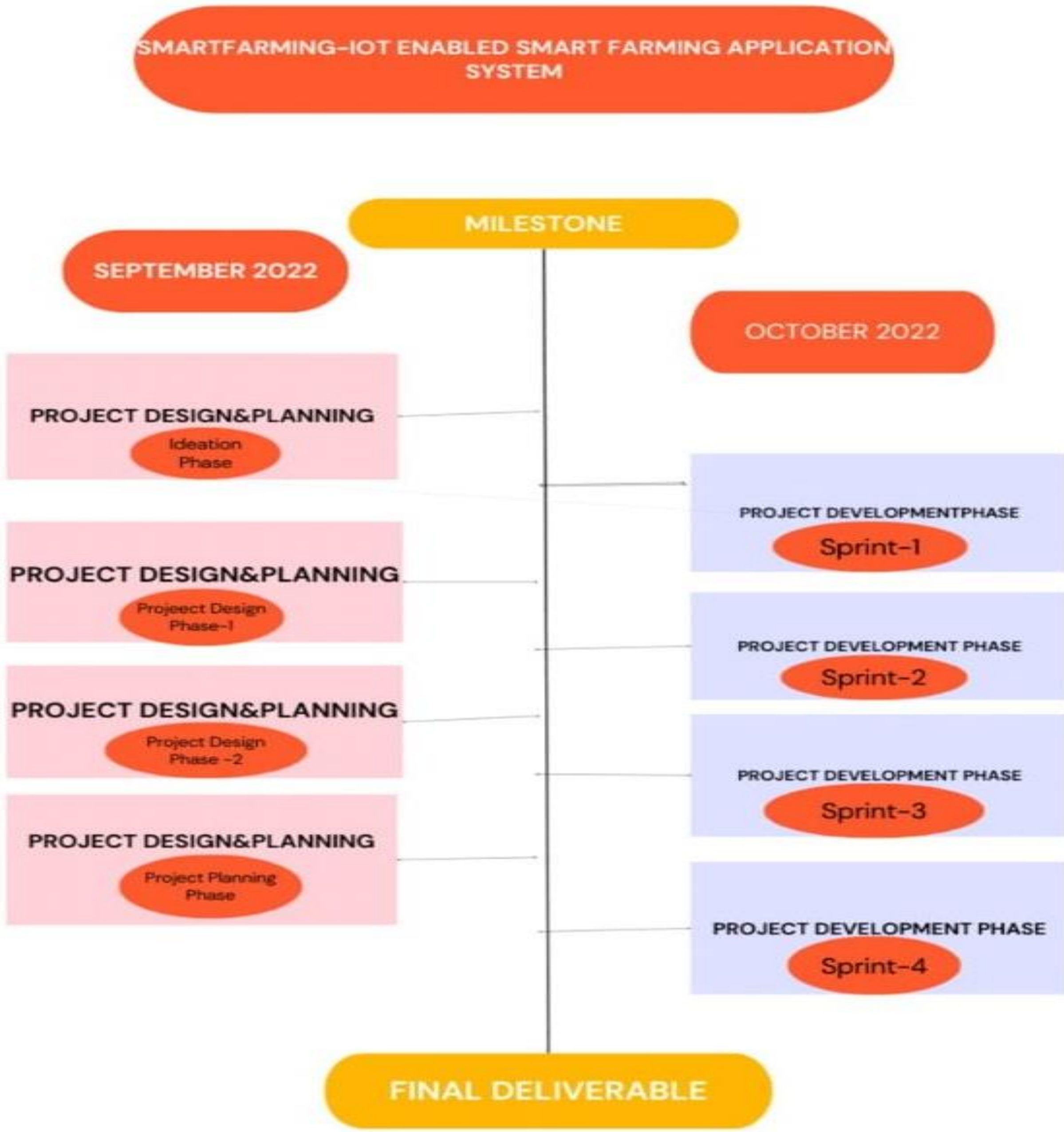
Use the below template to list all the user stories for the product

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can directly scan the QR code	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation details in the linked application	I can receive confirmation details & access it	High	Sprint-1
		USN-3	As a user, I can register for the application through the link shared in the social media.	I can register & access the dashboard with social media Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through URL link.	I can access the application easily in the direct login link.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application given Link.	I can log into the application and monitor.	High	Sprint-1
	Dashboard					
Customer (Web user)	Use Case	USN-1	To make the user to interact with the software.	User can use it with review given by the users.	Medium	Sprint-4
Customer Care Executive	User Requirements	USN-1	Have issues in the monitored message.	Issues in receiving the messages due to internet access.	Medium	Sprint-3
Administrator	Product Features	USN-1	IoT devices are Familiar to use.	Technology utilization is more accepted to the present generation development.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

Project planning is the process of defining your objectives and scope, your goals and milestones (deliverables), and assigning tasks and budgetary resources for each step. A good plan is easily shareable with everyone involved, and it's most useful when it's revisited regularly. Scheduling in project management is the listing of activities, deliverables, and milestones within a project. A schedule also usually includes a planned start and finish date, duration, and resources assigned to each activity.

6.1 Sprint Planning & Estimation



6.2 Sprint Delivery Schedule

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	4
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	4
Sprint-1		USN-3	As a user, I can register for the application through cloud	2	Low	4
Sprint-1		USN-4	As a user, I am shown what I can do in the product so I know whether or not this product will fill my needs	2	Medium	4
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	4
Sprint-3	Dashboard	USN-6	As a user,I can see all the collected data in the dashboard	1	High	4

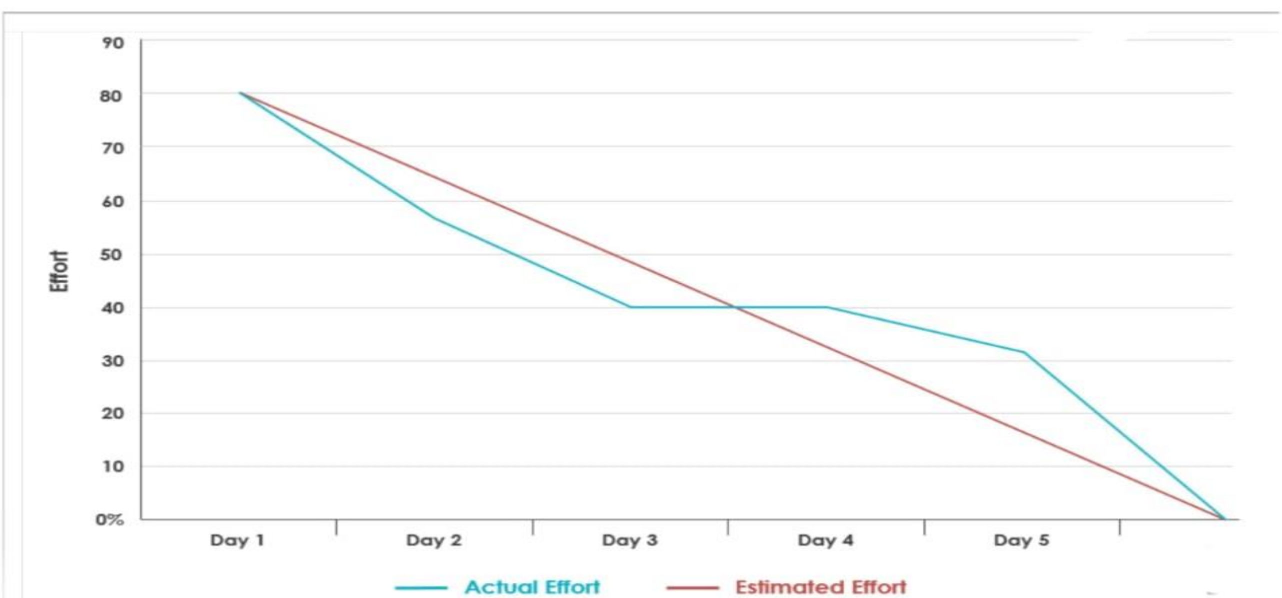
Project Tracker, Velocity:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint-1	20	10 Days	29 Aug 2022	08 Sep 2022	20	17 Sep 2022
Sprint-2	20	10 Days	19 Sep 2022	29 Oct 2022	20	01 Oct 2022
Sprint-3	20	10 Days	03 Oct 2022	13 Oct 2022	20	15 Oct 2022
Sprint-4	20	6 Days	16 Oct 2022	22 Oct 2022	20	22 Oct 2022
Sprint-5	20	10 Days	24 Oct 2022	02 Nov 2002	20	19 Nov 2022

Velocity: Imagine we have a 12-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

6.3 Reports from JIRA



7. CODING & SOLUTIONING

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHT_SENSOR_PIN  21 // ESP32 pin GIOP21 connected to DHT22 sensor
#define DHT_SENSOR_TYPE DHT22
#define RELAY_PIN    17 // ESP32 pin GIOP17 that connects to relay
#define AOUT_PIN 36
#define MOISTURE_PIN 36 // ESP32 pin GIOP36 (ADC0) that connects to AOUT pin of
moisture sensor
#define THRESHOLD 1000
#define POWER_PIN  17 // ESP32 pin GIOP17 connected to sensor's VCC pin
#define SIGNAL_PIN 36 // ESP32 pin GIOP36 (ADC0) connected to sensor's signal pin
#define SENSOR_MIN 0
#define SENSOR_MAX 521
#define LED 23
DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);// creating the instance by
passing pin and typr of dht connected
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-------credentials of IBM Accounts------
#define ORG "31b6or"//IBM ORGANITION ID
#define DEVICE_TYPE "smart_farm"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "Agriculture"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "-4+ZnfIxFunHoCkyIu"     //Token
#define METHOD "use-token-auth"
String data4;
float h, t, m, w;
//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined
client id by passing parameter like server id,portand wificredential
int value = 0; // variable to store the sensor value
int level = 0; // variable to store the water level

void setup()// configureing the ESP32
```

```
{
  Serial.begin(115200);
  dht_sensor.begin();// initialize the DHT sensor
  pinMode(RELAY_PIN, OUTPUT);
  pinMode(POWER_PIN, OUTPUT);   // configure D7 pin as an OUTPUT
  digitalWrite(POWER_PIN, LOW); // turn the sensor OFF
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()
{
  // read humidity
  float hum  = dht_sensor.readHumidity();
  // read temperature in Celsius
  float tempC = dht_sensor.readTemperature();

  // check whether the reading is successful or not
  if ( isnan(tempC) || isnan(hum))
  {
    Serial.println("Failed to read from DHT sensor!");
  }
  else
   {
    Serial.print("Humidity: ");
    Serial.print(hum);
    Serial.print("%");

    Serial.print("  |  ");

    Serial.print("Temperature: ");
    Serial.print(tempC);
    Serial.print("°C  ~  ");

   }
  m = analogRead(AOUT_PIN); // read the analog value from sensor
  m = analogRead(MOISTURE_PIN); // read the analog value from sensor
  if (m < THRESHOLD)
  {
    Serial.print("The soil is DRY (");
    Serial.print("The soil is DRY => turn pump ON");
    digitalWrite(RELAY_PIN, HIGH);
  }
  else
  {
```

```
    Serial.print("The soil is WET (");
    Serial.print("The soil is WET => turn pump OFF");
    digitalWrite(RELAY_PIN, LOW);
  }
  Serial.print(" (");
  Serial.print(m);
  Serial.println(")");
  delay(500);
  digitalWrite(POWER_PIN, HIGH);  // turn the sensor ON
  delay(10);                      // wait 10 milliseconds
  value = analogRead(SIGNAL_PIN); // read the analog value from sensor
  digitalWrite(POWER_PIN, LOW);   // turn the sensor OFF
  level = map(value, SENSOR_MIN, SENSOR_MAX, 0, 4); // 4 levels
  Serial.print("Water level: ");
  Serial.println(level);
  delay(1000);
  PublishData(t, h, m, w);
  delay(1000);
  if (!client.loop())
  {
    mqttconnect();
  }
}

/*....................................retrieving to
Cloud...............................*/
void PublishData(float temp, float hum, float moist, float water)
{
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temperature\":";
  payload += temp;
  payload += "," "\"humidity\":";
  payload += hum;
  payload = "{\"moisture\":";
  payload += moist;
  payload += "}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str()))
  {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
  }
  else
```

```
  {
    Serial.println("Publish failed");
  }

}
void mqttconnect()
{
  if (!client.connected())
  {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token))
    {
      Serial.print(".");
      delay(500);
    }
     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice()
{
  if (client.subscribe(subscribetopic))
  {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  }
  else
  {
    Serial.println("subscribe to cmd FAILED");
```

```
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++)
  {
    //Serial.print((char)payload[i]);
    data4 += (char)payload[i];
  }

  Serial.println("data: "+ data4);
  if(data4=="lighton")
  {
Serial.println(data4);
digitalWrite(LED,HIGH);

  }

  else
  {
Serial.println(data4);
digitalWrite(LED,LOW);

  }
data4="";
}
```

OUTPUT:

```
Connecting to ..
WiFi connected
IP address:
10.10.0.2
Reconnecting client to 31b6or.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Humidity: 40.00%  |  Temperature: 24.00°C  ~  The soil is WET (The soil is WET => turn pump OFF
(2911.00)
Water level: 21
Sending payload: {"moisture":2911.00}
Publish ok
Humidity: 40.00%  |  Temperature: 24.00°C  ~  The soil is WET (The soil is WET => turn pump OFF
(2583.00)
Water level: 18
Sending payload: {"moisture":2583.00}
Publish ok
Humidity: 40.00%  |  Temperature: 24.00°C  ~  The soil is WET (The soil is WET => turn pump OFF
(2253.00)
Water level: 17
```

7.1 Feature 1

ESP32 DEVKIT V1

- ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process. It is a successor to the ESP8266 microcontroller.

- The ESP-WROOM-32 is a versatile Wi-Fi + BT+ BLE MCU module Two CPU cores can be individually controlled or powered on.

- The sleep current of the ESP32 chip is less than 5 µA, making it suitable for battery powered and wearable electronics applications.

ESP32 DEVKIT V1 Operational Modes

• Active mode: The chip radio is powered on.

• Modem-sleep mode: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.

• Light-sleep mode: The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP co-processor are running.

•Deep-sleep mode: Only the RTC memory and RTC peripherals are powered on. Wi-Fi and Bluetooth connection data are stored in the RTC memory. The ULP co-processor can work.

• Hibernation mode: The internal 8-MHz oscillator and ULP co-processor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and some RTC GPIOS are active. The RTC timer or the RTC GPK can wake up the chip from the Hibernation mode.

Processors:

- CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
- Ultra low power (ULP) co-processor

Memory: 320 KiB RAM, 448 KiB ROM
Wireless connectivity:

- Wi-Fi: 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)

Peripheral interfaces:

- 34 × programmable GPIOs
- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit DACs
- 10 × touch sensors (capacitive sensing GPIOs)
- 4 × SPI
- 2 × I²S interfaces
- 2 × I²C interfaces
- 3 × UART
- SD/SDIO/CE-ATA/MMC/eMMC host controller
- SDIO/SPI slave controller
- Ethernet MAC interface with dedicated DMA and planned IEEE 1588 Precision Time Protocol support
- CAN bus 2.0
- Infrared remote controller (TX/RX, up to 8 channels)
- Pulse counter (capable of full quadrature decoding)
- Motor PWM
- LED PWM (up to 16 channels)
- Hall effect sensor
- Ultra low power analog pre-amplifier

Security:

- IEEE 802.11 standard security features all supported, including WPA, WPA2, WPA3 (depending on version)[5] and WLAN Authentication and Privacy Infrastructure (WAPI)
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customer
- ryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)
- Power management:
- Internal low-dropout regulator
- Individual power domain for RTC
- 5 µA deep sleep current

Wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt.

Specifications:

ESP-32	DESCRIPTION
Core	2
Arquitecture	32 bits
Clock	Tensilica Xtensa LX106 160-240MHz
WiFi	IEEE802.11 b/g/n
Bluetooth	Yes - classic & BLE
RAM	520KB
Flash	Extern QSPI - 16MB
GPIO	22
DAC	2
ADC	18
Interfaces	SPI-I2C-UART-I2S-CAN

Fig: Specifications

Function Block Diagram:

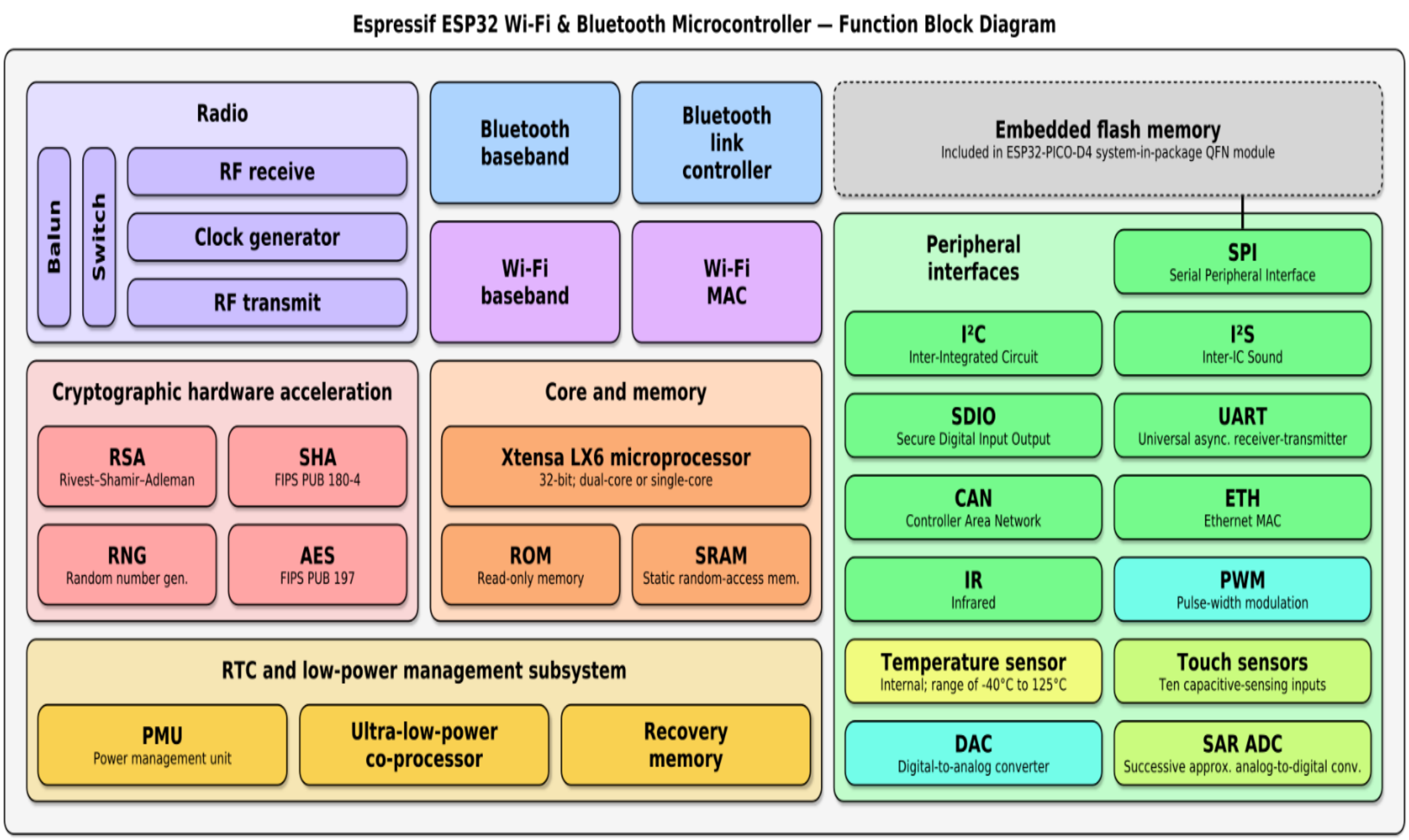


Fig: Functions of ESP32

Pin Diagram of ESP32:

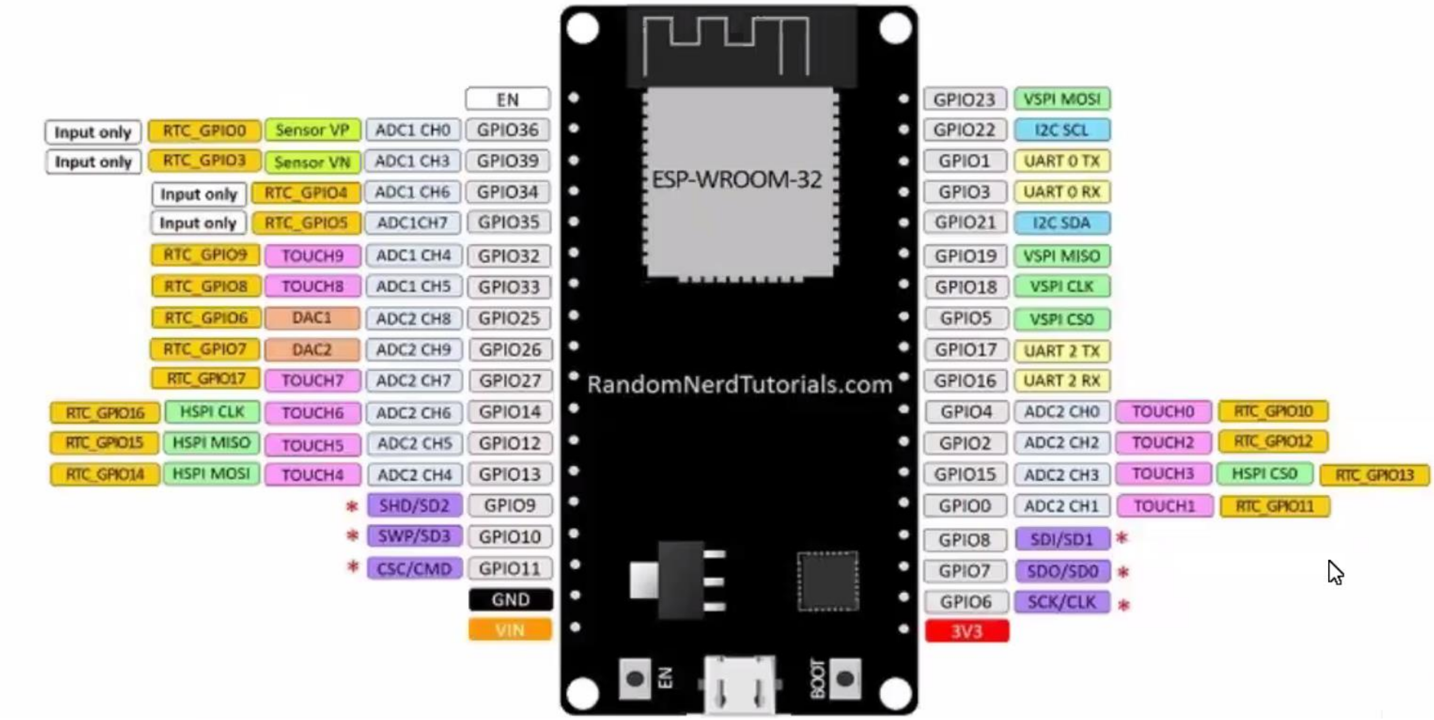
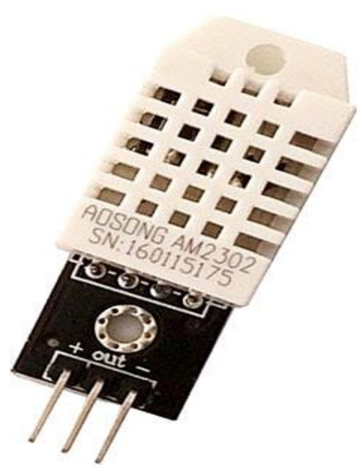


Fig: Pin Diagram

☞ ***Pins SCK/CLK, SDO/SDO, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.***

7.2 Feature 2

1.DHT22- Temperature and Humidity Sensor



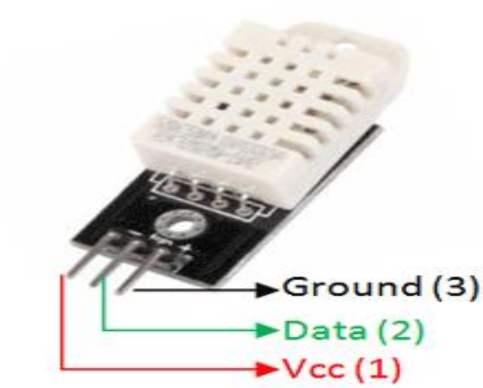


Fig:(a)DHT22 *Fig:(b) DHT22-Sensor Pinout*

For DHT22 Module

1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

DHT22 Specifications

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: -40°C to 80°C
- Humidity Range: 0% to 100%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±0.5°C and ±1%

Applications

- Measure temperature and humidity
- Local Weather station
- Automatic climate control
- Environment monitoring

2.Capacitive Soil Moisture Sensor

Fig: Capacitive Soil Moister Sensor

This **Capacitive soil moisture** sensor measures soil moisture levels by capacitive sensing rather than resistive sensing like other sensors on the market. It is made of corrosion-resistant material which gives it excellent service life. Insert it into the soil around your plants and impress your friends with real-time soil moisture data.

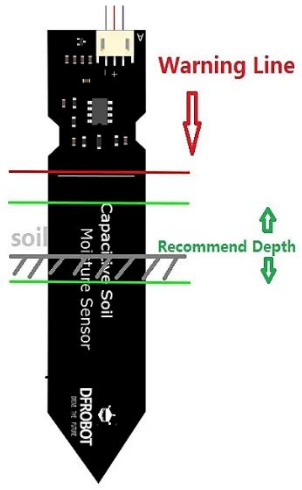


Fig: Recommended line

Fig: Length of soil moister sensor

Specifications:

Operating Voltage	3.3 ~ 5.5 VDC
Output Voltage	0 ~ 3.0VDC
Operating Current	5mA
Interface	PH2.54-3P
Dimensions mm(LxWxH)	98 x 23 x 4
Weight (gm)	15
Shipment Weight	0.018 kg
Shipment Dimensions	12 × 4 × 1 cm

Features:

1. Supports 3-Pin Gravity Sensor interface
2. Analog output

Applications:

1. Garden plants
2. Moisture detection
3. Intelligent agriculture

7.3 Database Schema

IBM Cloud:

IBM cloud services are a full-stack consisting of a huge collection of more than 170 products and cloud computing services for business-to-business (B2B) organizations, which are deployed around the world. As is the case with many other universal cloud computing services – such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud – IBM Cloud includes the three main service models (or types) of cloud computing. These include: infrastructure as a service (IaaS), platform as a service (PaaS), and, recently, software as a service (SaaS), with a special focus on (IaaS). It is also offered through the deployment models of public, private, and hybrid cloud.

IBM Cloud computing provides its services for startups that have small development teams, alongside large enterprise businesses and organizations. In addition to public, private, and hybrid cloud delivery models, it positions these services within three umbrellas: SmartCloud Foundation, SmartCloud Services, and SmartCloud Solutions, in order to increase productivity and effectiveness.

IBM Cloud Services

As we mentioned above, cloud services from large companies are usually a world without borders or a bottomless ocean. The following list is merely an attempt to confine the most common and important IBM Cloud Services in addition to some third-party vendors' support services.

Watson Iot Platform

A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualization and data storage.

Watson IoT Platform features

Completely manage your IoT landscape and make better business decisions. Using a secure, smart and scalable platform as the hub of your IoT, get real-time analysis of user, machine and system-generated data, including speech, text video and social sentiment. You need contextual insight for truly cognitive IoT applications.

8.TESTING
8.1 Test Cases

Table 1- Components and Technologies

S.No	Component	Description	Technology
1.	User Interface	Smart farming solutions work through sensors. Farmers can monitor various conditions like soil moisture, water level, Temperature & humidity from anywhere by combining sensors through Mobile Application.	MIT App Inventor.
2.	Application Logic-1	The code will be builded based on certain conditions like assigning task to get the Sensor datas & then based on moisture level the water flow will be controlled and if moisture level exceeds certain level it will also intimate through message.	C++,Python
3.	Application Logic-2	Here we create a device and then connected to the Node-red to develop a software.	IBM Watson STT service .
4.	Application Logic-3	Here sensed data will be displayed on the IBM dashboard.	IBM Watson Assistant.
5.	Database	We can store all the datas, so that user can retrieve the data whenever required .	Drive, NoSQL, etc.
6.	Cloud Database	In cloud we can store the data and get back the data at any moment & it will displayed on the dashboard. This datas will be stored here safely for future purpose.	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements.	IBM Block Storage or Other Storage Service or Local File system.
8.	External API	Using open weather map as a external source we can collect the weather information.	IBM Weather API.
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Wokwi Cloud Server Configuration : IBM	Local, Cloud Foundry

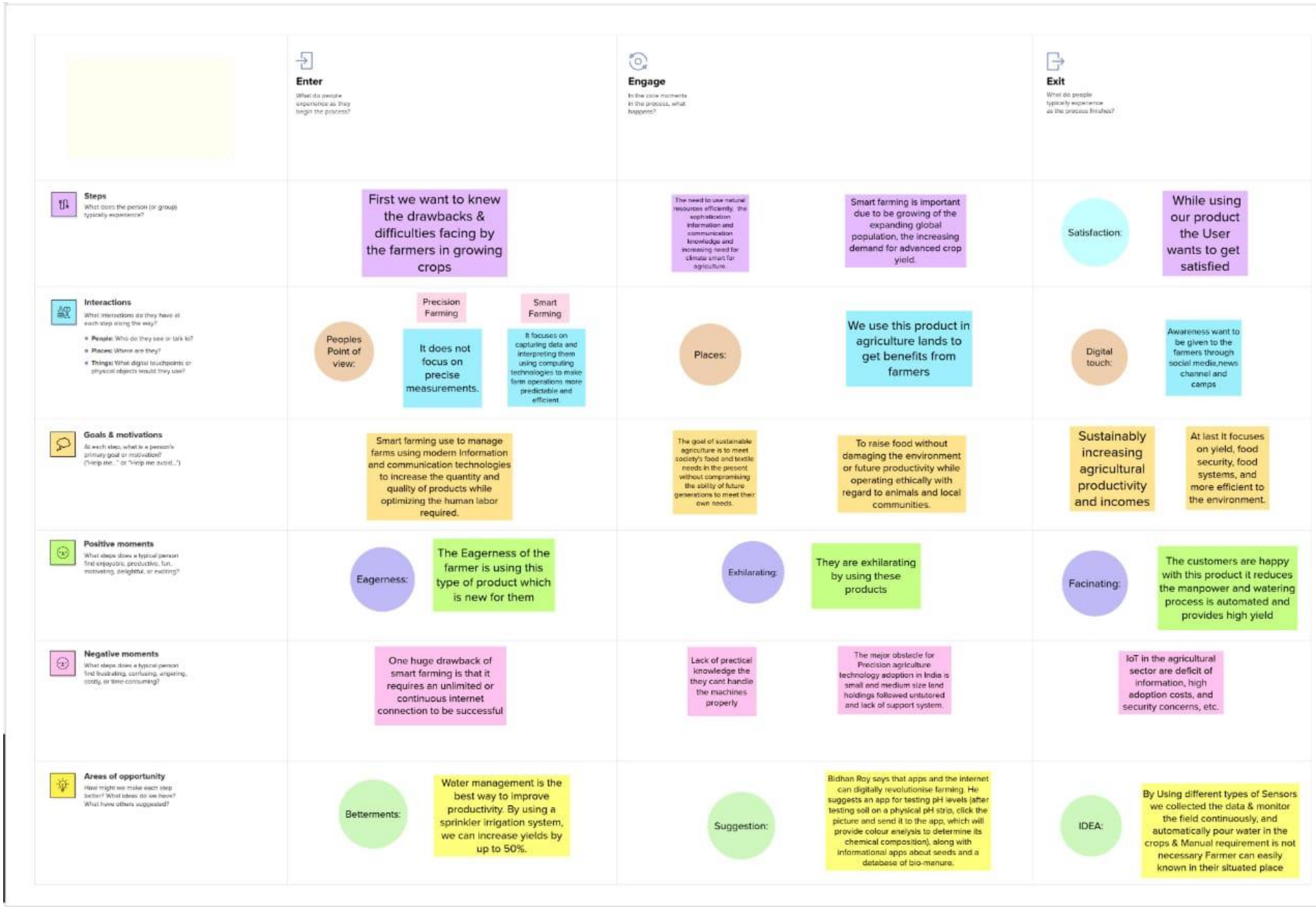
Table 2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Farmers can able to control their own data using this technology.	Open weather Map, MIT App Inventor, Wokwi.
2.	Security Implementations	We are using secured Platform like IBM Cloud to store all the datas & we can collect the data whenever we required.	IBM Cloud, IBM Watson Assistant
3.	Scalable Architecture	It can able to handle more amount of works by adding resources to the system. It supports higher workload without any fundamental changes to it.	IBM cloud
4.	Availability	The data gathered from the sensors transfer to the farmers can monitor all information using phone.	Sensors, IBM Watson, Node-red, MIT App
5.	Performance	The received data from the sensors are all together stored in the IBM cloud drive, the message is given to the user application to perform the process. By the data the automatic performance is also done.	Sensors, IBM Watson, Node-red, MIT App

8.2 User Acceptance Testing

User Story Number	User Story / Task	Acceptance criteria
USN-1	As a user, I can directly scan the QR code	I can access my account / dashboard.
USN-2	As a user, I will receive confirmation details in the linked application	I can receive confirmation details & access it.
USN-3	As a user, I can register for the application through the link shared in the social media.	I can register & access the dashboard with social media Login .
USN-4	As a user, I can register for the application through URL link.	I can access the application easily in the direct login link.
USN-5	As a user, I can log into the application given Link.	I can log into the application and monitor.

9. RESULTS
9.1 Performance Metrics



10. ADVANTAGES & DISADVANTAGES

Advantages:

- Soil sensing is also employed to appropriately control the application of real-time variable rate equipment. This allows you to understand the scale of your grounds, making you also, in this process, device effective ways of conserving necessary farming resources like water.
- You also get to get important information about the the levels of humidity, and temperature of your environment.
- Lower driver stress.
- Ease of use.
- Easier recording and reading.

Disadvantages:

- One huge disadvantage of smart farming is that it requires an unlimited or continuous internet connection to be successful. This means that in rural communities, especially in the developing countries where we have mass crop production, it is completely impossible to operate this farming method. In places where internet connections are frustratingly slow, smart farming will be an impossibility.
- As pointed out earlier, smart farming makes use of high techs that require technical skill and precision to make it a success. It requires an understanding of robotics and ICT. However, many farmers do not have these skills. Even finding someone with this technical ability is difficult or even expensive to come by, at most. And, this can be a discouraging factor hindering a lot of promising farmers from adopting it.

11. CONCLUSION

Smart farming can make agriculture more profitable for the farmer. Decreasing resource inputs will save the farmer money and labor, and increased reliability of spatially explicit data will reduce risks.

Will enhance new sectors and technologies that will be the main sources of economic development and growth of the future.

IoT based smart farming system can prove to be very helpful for farmers since over as well as less irrigation is not good for farming. Threshold values for climatic conditions like humidity, temperature, moisture can be fixed based on the environmental conditions of that particular region. This system generates irrigation schedule based on the sensed real time data from field and data from the weather repository. This system can recommend farmer whether or not, is there a need for irrigation.

12. FUTURE SCOPE

The sensors and microcontroller are successfully interfaced and wireless communication is achieved between various nodes. All observations and experimental tests prove that this project is a complete solution to field activities and irrigation problems. Implementation of such a system in the field can definitely help to improve the yield of the crops and overall production.

In day to day life agriculture is one of the most important parts in every human's life. Many farmers are carried out on road and on field operations this will increase the time and man power. This paper is effectively used for thefarmers for reducing the man power and accesses their crops in an efficient way. This might reduce the time and work of the farmers. This is a less cost process that contains various sensors that are used to sense various parameters and this can be used effectively by the farmers using their mobile phone. And this can be carried out.

One of the limitations of this system is that continuous internet connectivity is required at user end which might prove to be costly for farmer. This can be overcome by extending the system to send suggestion via SMS to the farmer directly on his mobile using GSM module instead of mobile app. Weather data from the meteorological department can be used along with the sensed data to predict more information about the future which can help farmer plan accordingly and improve his livelihood.

13. APPENDIX

Source Code GitHub & Project Demo Link

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-1831-1658416607

Project Demo Link:
https://drive.google.com/file/d/10ESa_Ep8jK1erwx3dBolWKMLGlKtgNOp/view?usp=drivesdk