**IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE**

**NALAIYA THIRAN PROJECT BASED LEARNING ON**

**PROFESSIONAL READLINESS FOR INNOVATION,**

**EMPLOYNMENT AND ENTERPRENEURSHIP**

**A PROJECT REPORT**

***SUBMITTED BY***

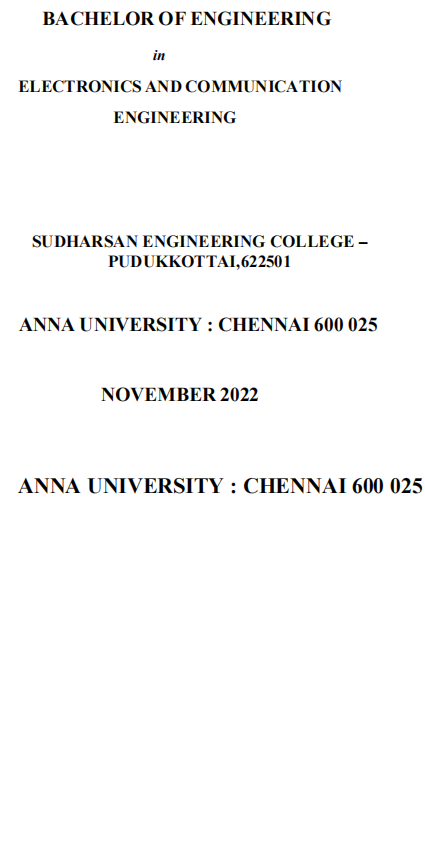
1. ***HARIHARASUTHAN-814419106002***

***S.ARUN-814419106001***

***S.KARTHIKEYAN-814419106003***

***S.DHILEEPKUMAR-814419106303***

***S.SIVAKUMAR-814419106304***



**1. INTRODUCTION:**

**1.1 PROJECT OVERVIEW:**

The Internet of Things Smart technology allows new digital agriculture. Today, technology has become a requirement in order to handle contemporary difficulties, and many industries are utilizing cutting-edge technologies to automate their processes. Advanced agriculture, based on Internet of Things technology, is intended to help producers and farmers decrease waste and increase output by optimizing fertilizer use to raise plant efficiency. It allows farmers more control over their animals, crops, and costs and resources.

With the widespread use of the Internet of Things (IoT), linked devices have permeated every part of our lives, from health and fitness to home automation, automotive and logistics, smart cities, and industrial IoT. As a result, it is only natural that IoT, linked devices, and automation would make their way into agriculture, vastly improving practically every aspect of it. In the previous several decades, farming has undergone a variety of technical revolutions, becoming increasingly industrialized and technology-driven. Farmers have acquired improved control over the process of producing animals and growing crops by employing die rent smart agricultural technologies, making it more predictable and e client.

The Smart protection system defines that this project help to farmer for the protection of a farm. We have designed this project for the only secure from animals but we this project have the provision to secure from the human begins also. This can achieve by the help of IOT device that we are discuss in this paper. The SCPS work on the battery so that this project can be easily portable and also we are add solar panels and converter modules this can help the battery to charge from solar energy.

The IOT device is used to indicate the farmer by a message while someone enter into the farm and we are used SD card module that helps to store a specified sound to fear the animals. This project is smart crop protection system for protect the farm from animals as well as unknown person. This projects contents ardiuno UNO, Nodemcu, LCD display, PIR sensor, flame sensor, sd card module ,solar panel, solar charges converter. This whole project is work on 12v dc supply from battery. We used solar panel to charge the battery.

**1.2 PURPOSE:**

The project aims at making use of evolving technology i.e. IOT and smart agriculture using automation. Monitoring environmental conditions is the major factor to improve yield of the efficient crops.

**2. LITERATURE SURVEY**

**2.1 EXISTING PROBLEM:**

Iot based smart agriculture monitoring system. Rajalakshmi.P and S. Devi Mahalakshmi, “IOT Based Crop Field Monitoring and Irrigation Automation”, 10th International conference on Intelligent systems and control (ISCO), 2016. An IOT Based Crop-field monitoring an irrigation automation system describes how to monitor a crop field. A system is developed by using sensors and according to the decision from a server based on sensed data, the irrigation system is automated. Through wireless transmission the sensed data is forwarded to web server database. If the irrigation is automated then the moisture and temperature fields are decreased below the potential range. The user can monitor and control the system remotely with the help of application which provides a web interface to user. By smart Agriculture monitoring system and one of the oldest ways in agriculture is the manual method of checking the parameters. In this method farmers by themselves verify all the parameter and calculate the reading .The system focuses on developing devices and tool to manage, display and alert the users using the advantages of a wireless sensor network system. It aims at making agriculture smart using automation and IoT technologies. The cloud computing devices are used at the end of the system that can create a whole computing system from sensors to tools that observe data from agriculture field. It proposes a novel methodology for smart farming by including a smart sensing system and smart irrigator system through wireless communication technology. This system is cheap at cost for installation. Here one can access and also control the agriculture system in laptop, cell phone or a computer.

**LIMITATIONS**

1. There could be a wrong analysis of weather conditions.

2. Devices are to be altered according to the farmers, it will involve equipment which will be expensive.

3. If there are faulty data processing equipment or sensors, then it will lead to a situation where the decisions are taken wrong .

**2.2 REFERENCES:**

[1] Sinung Suakanto, Ventje J. L. Engel, Maclaurin Hutagalung, Dina Angela, “Sensor networks data acquisition and task management for decision support of smart agriculture,” in 2016 International Conference on Information Technology Systems and Innovation (ICITSI) Bandung – Bali, pp. 24–27, Oct. 2016.

[2] Chetan Dwarkani M, Ganesh Ram R, Jagannathan S, R. Priyatharshini “Smart agriculture system using sensors for agricultural task automation,” in 2015 IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development (TIAR 2015).

[3] Nikesh Gondchwar, R. S. Kawitkar, “IOT based smart agriculture,” International journal Of Advanced research in computer and Communication Engineering (IJARCCE), vol. 5, no. 6, Jun. 2016.

[4] Narayut Putjaika, Sasimanee Phusae, Anupong Chen-Im, Phond Phunchongharn and Khajonpong Akkarajit Sakul, “A control system in intelligent agriculture by using arduino technology,” in Fifth ICT International Student Project Conference(ICT-ISPC), 2016.

[5] Tejas Bangera, Akshar Chauhan, Harsh Dedhia, Ritesh Godambe, Manoj Mishra, “IOT based smart village,” International Journal of Engineering Trends and Technology (IJETT), vol. 32, no. 6, Feb. 2016, ISSN: 2231- 5381.

[6] Jeetendra Shenoy, Yogesh Pingle “IOT in agriculture,” 978-9-3805- 4421-2/16/, IEEE. 2016.

[7] Rajalakshmi P and S. Devi Mahalakshmi, IOT Based Crop-Field Monitoring and Irrigation Automation.

[8] Abdullah Na, William Isaac, “Developing a human-centric agricultural model in the IOT environment,” in 2016 International Conference on Internet of Things and Applications (IOTA) Maharashtra Institute of Technology, Pune, India 22 Jan - 24 Jan, 2016, 978-1-5090-0044-9/16, 2016 IEEE.

[9] Syed Mubarak and S. Sujatha “International journal of advance research in science and engineering,” IJARSE, vol. 4, no. 01, May 2015, ISSN23198354(E).

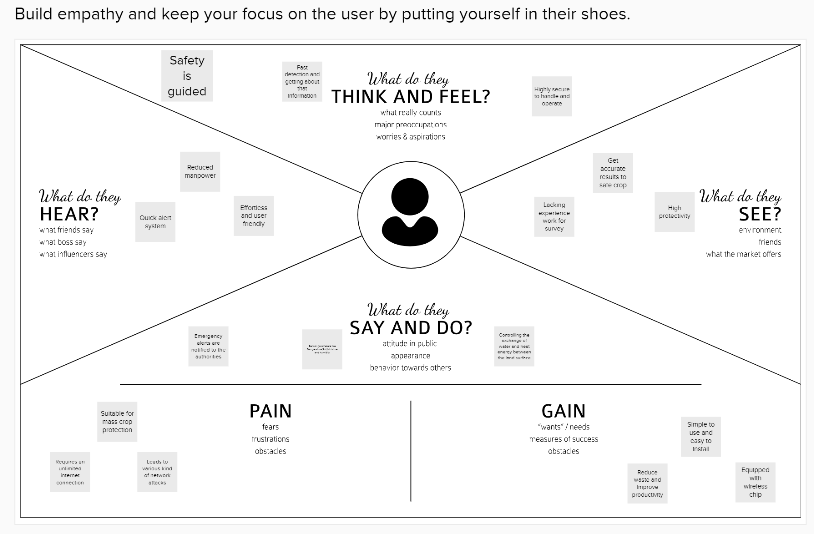
[10] Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Ángel Porta- Gándara”Automated Irrigation System Using a Wireless Sensor Network and GPRS module” , Ieee Transactions On Instrumentation And Measurement, Vol. 63, No. 1, January 2014.

**2.3 PROBLEM STATEMENT DEFINITION:**

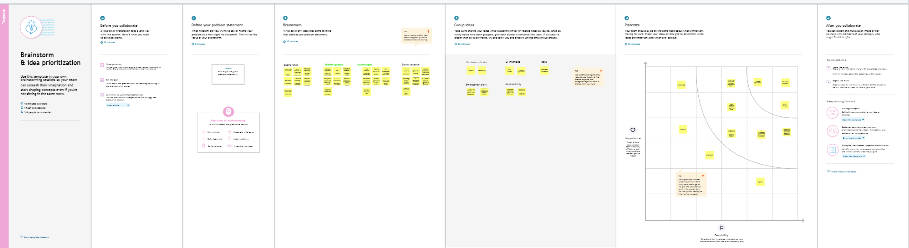
Proposes the automatic irrigation system using Arduino for smart crop field productivity. This system consists of sensor like moisture sensor, temperature sensor, rain sensor and ultrasonic sensor. Moisture sensor used for detecting the moisture content in soil, temperature sensor is used to measure the temperature value and ultrasonic sensor is used to measure the water level. If the excess of water in field is detected, then motor will ON to remove it. The measured parameters are uploaded to IOT through this farmer can monitor anywhere. Based on the command from IOT water motor will ON. The measured parameters are displays on LCD.

**3. IDEATION & PROPOSED SOLUTION**

**3.1 EMPATHY MAP CANVAS:**



**3.2 IDEATION & BRAINSTORMING:**



**3.3 PROPOSED SOLUTION:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | Protecting the crops from wild animals and also monitor the soil moisture levels in the field. |
| 2. | Idea / Solution description | Monitoring the crops using IOT based technology by 24x7.Detecting animals and scared them away. The motors and sprinklers in the field can be controlled using the mobile application. |
| 3. | Novelty / Uniqueness | Accessibility due to the alarm system, being helpful to farmers.  Scalability. |
| 4. | Social Impact / Customer Satisfaction | User friendly.  Its increase rate of good yield of crops. |
| 5. | Business Model (Revenue Model) | Cost efficient.  It’s reduce the anxiety and fear of losing crops meanwhile it’s help to increase the profit comparing to previous year. |
| 6. | Scalability of the Solution | In future it can be enhanced by sending message directly to the fire department in case there is a mass wild animals attacks the fields. The controlling and monitoring of the soil moisture level can be automated by taking care of the crops in case of low moisture level, without notifying the farmers. |

**3.4 PROBLEM SOLUTION FIT:**



**4. REQUIREMENT ANALYSIS**

**4.1 FUNCTIONAL REQUIREMENT:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **FUNCTIONAL REQUIREMENT.** | **SUB REQUIREMENT.** |
| 1. | User Visibility | Sense animals nearing the crop field & sounds alarm to woo them away as well as sends SMS to farmer using  cloud service. |
| 2. | User Reception | The Data like values of Temperature, Humidity, Soil moisture Sensors are  received via SMS. |
| 3. | User Understanding | Based on the sensor data value to get the information about the present of farming  land. |
| 4. | User Action | The User needs take action like destruction of crop residues, deep plowing, crop rotation, fertilizers, strip  cropping, scheduled planting operations. |

# 4.2 NON-FUNCTIONAL REQUIREMENTS:

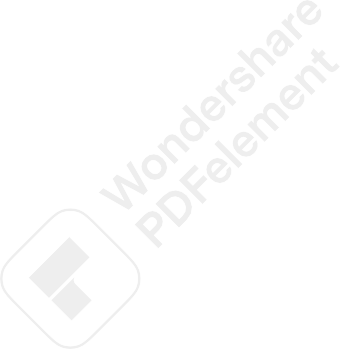
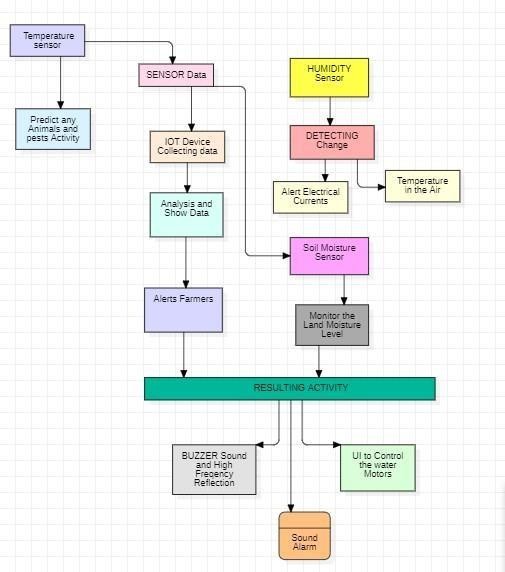
Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **NON-FUNCTIONAL REQUIREMENT.** | **DESCRIPTION.** |
| 1. | Usability | Mobile Support Users must be able to interact in the same roles & tasks on computers & mobile devices where practical, given mobile  Capabilities. |
| 2. | Security | Data requires secure access to must register and communicate securely on devices and authorized users of the system who exchange  Information must be able to do. |
| 3. | Reliability | It has a capacity to recognize the disturbance near the field and doesn’t give a false  Caution signal. |
| 4. | Performance | Must provide acceptable response times to users regardless of the volume of data that is stored and the analytics that occurs in background. Bidirectional, near real-time communications must be supported. This requirement is related to the requirement to support industrial and  Device protocols at the edge. |
| 5. | Availability | IOT Solutions and domains demand highly available systems for 24 x 7 operations. Isn’t a critical production application, which means that operations or production  don’t go down if the IOT solution is down. |
| 6. | Scalability | System must handle expanding load & data retention needs that are based on the up scaling of the solution scope, such as extra  Manufacturing facilities and extra buildings. |

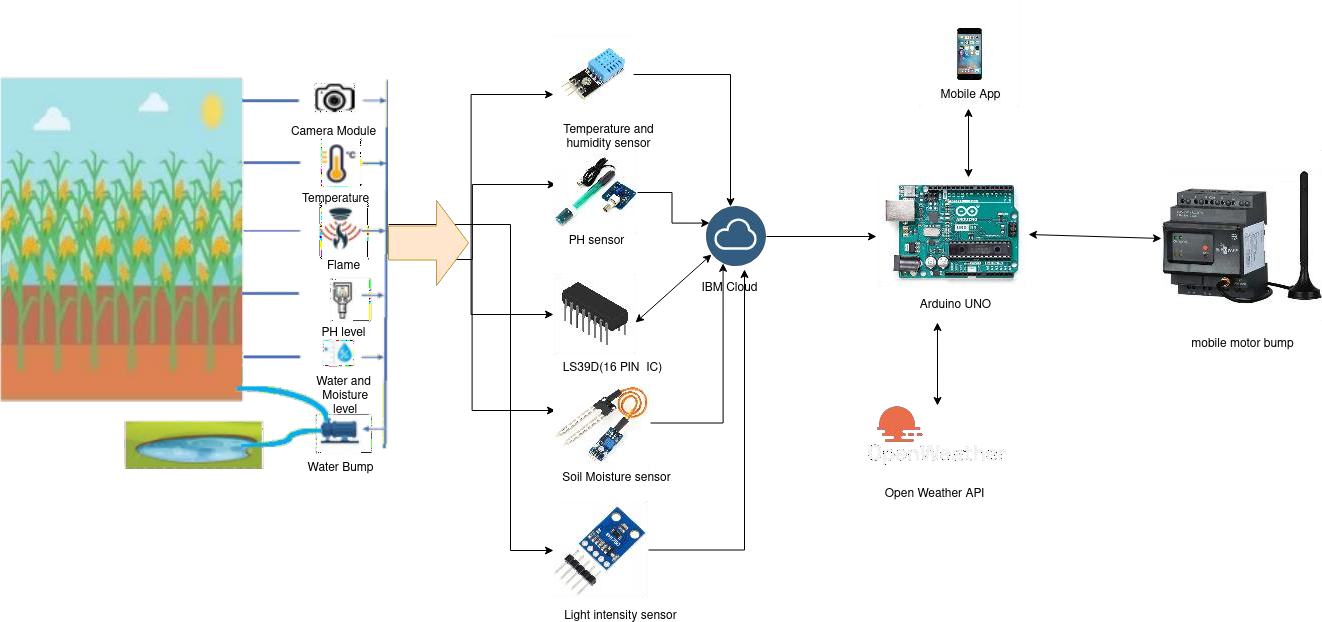
**5. PROJECT DESIGN**

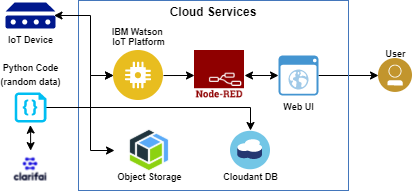
**5.1 DATA FLOW DIAGRAMS:**

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.



**5.2 SOLUTION & TECHNICAL ARCHITECTURE:**

**SOLUTION ARCHITECTURE:**

**TECHNICAL ARCHITECTURE:**

**5.3 USER STORIES:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional requirement (Epic)** | **User Story number** | **User Story/Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Mobile  user) | Registration | USN-1 | User can enter into the web application | I can access my account  /dashboard | High | Sprint 1 |
|  |  | USN-2 | User can register their credentials like email id and password | I can receive confirmation email &click  confirm | High | Sprint 1 |
|  | Login | USN-3 | User can log into the  application y entering email and password | I can login to my account | High | Sprint 1 |
|  | Dashboard | USN-4 | User can View the temperature | I can view the data given by the device | High | Sprint 2 |
|  |  | USN-5 | User can view the level of sensor monitoring value | I can view the data given by the  device | High | Sprint 2 |
| Customer (Web user) | Usage | USN-1 | User can view the web page and get the information | I can view the data given by the device | High | Sprint 3 |
| customer | Working | USN-1 | User act according to the alert given by the device | I can get the data work according to it | High | Sprint 3 |
|  |  | USN-2 | User turns ON the water motors/Buzzer/Sound Alarm when occur the disturbance on field | I can get the data work according to it |  | Sprint 4 |

**6. PROJECT PLANNING & SCHEDULING**

**6.1 SPRINT PLANNING & ESTIMATION:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 |  | US-1 | Create the IBM Cloud services which are being used in this project. | 6 | High | Hariharasuthan,  Arun,Karthikeyan,  Siva,Dhileepkumar |
| Sprint-1 |  | US-2 | Configure the IBM Cloud services which are being used in completing this project. | 4 | Medium | Hariharasuthan,  Arun,Karthikeyan,  Siva,Dhileepkumar |
| Sprint-2 |  | US-3 | IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform. | 5 | Medium | Hariharasuthan,  Arun,Karthikeyan,  Siva,Dhileepkumar |
| Sprint-2 |  | US-4 | In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials. | 5 | High | Hariharasuthan,  Arun,Karthikeyan,  Siva,Dhileepkumar |
| Sprint-3 |  | US-1 | Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform. | 10 | High | Hariharasuthan,  Arun,Karthikeyan,  Siva,Dhileepkumar |
| Sprint-3 |  | US-2 | Create a Node-RED service. | 1 | High | Hariharasuthan,  Arun,Karthikeyan,  Siva,Dhileepkumar |

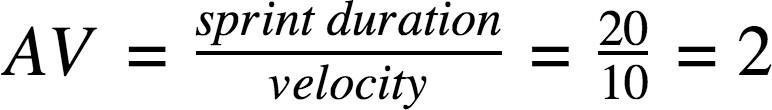
**6.2 SPRINT DELIVERY SCHEDULE:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 20 | 6 Days | 12.11.22 | 18.11. 22 | 20 | 20.11.22 |
| Sprint-2 | 20 | 6 Days | 12.11.22 | 18.11. 22 | 20 | 20.11.22 |
| Sprint-3 | 20 | 6 Days | 12.11.22 | 18.11. 22 | 20 | 20.11.22 |
| Sprint-4 | 20 | 6 Days | 12.11.22 | 18.11. 22 | 20 | 20.11.22 |

**6.3 Reports from JIRA:**

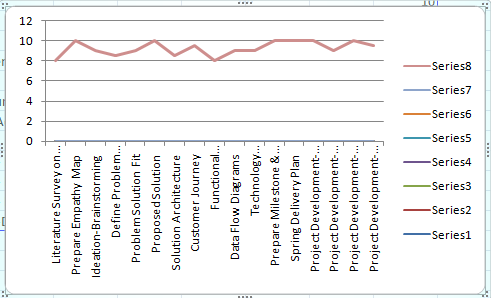
**Velocity:**

Imagine we have 10-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)



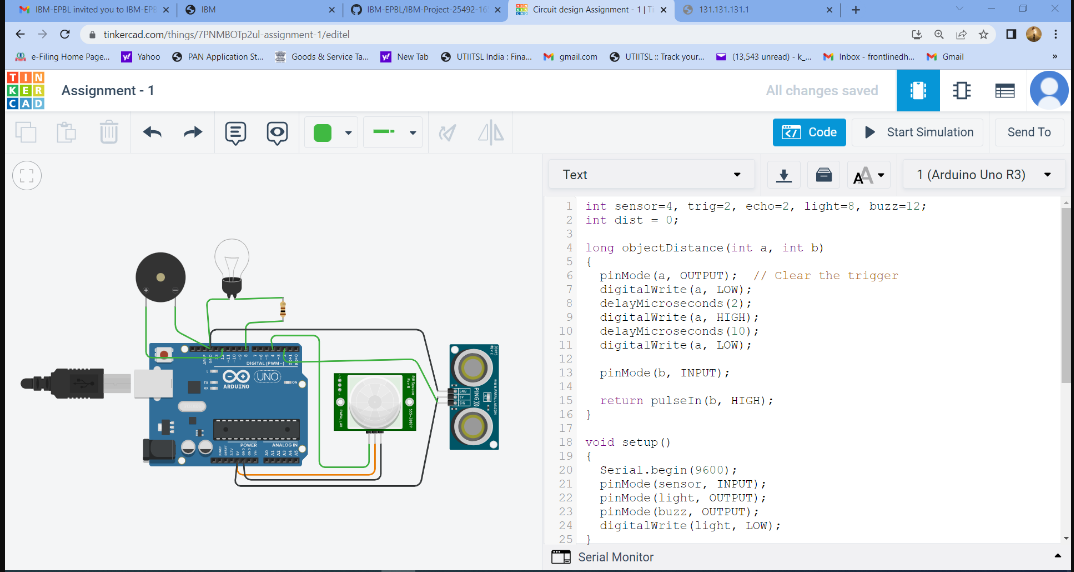
**Burn down Chart:**

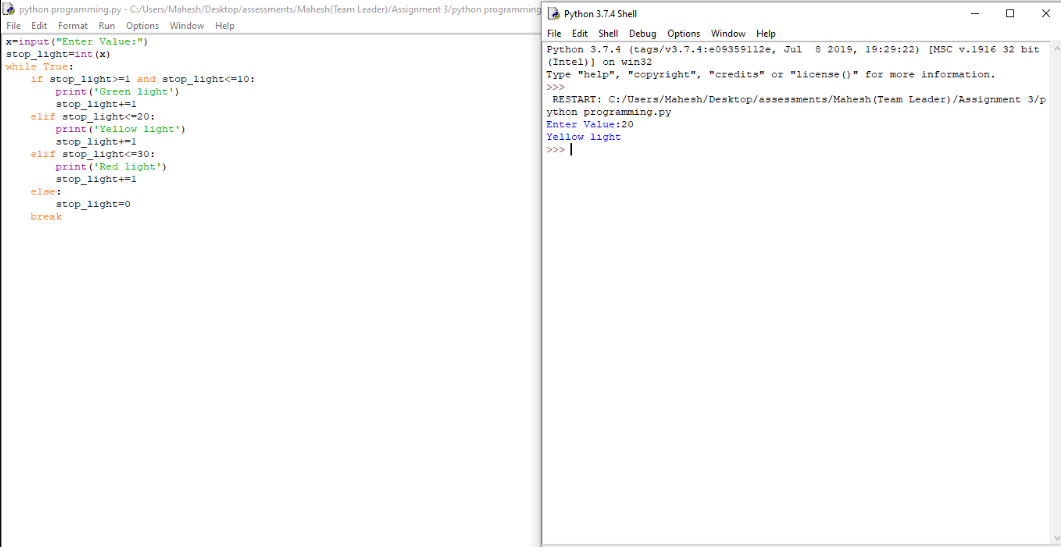
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress overtime.

****

**7. CODING & SOLUTIONING:**

|  |
| --- |
| int sensor=4, trig=2, echo=2, light=8, buzz=12; |
|  | int dist = 0; |
|  |  |
|  | long objectDistance(int a, int b) |
|  | { |
|  | pinMode(a, OUTPUT); // Clear the trigger |
|  | digitalWrite(a, LOW); |
|  | delayMicroseconds(2); |
|  | digitalWrite(a, HIGH); |
|  | delayMicroseconds(10); |
|  | digitalWrite(a, LOW); |
|  |  |
|  | pinMode(b, INPUT); |
|  |  |
|  | return pulseIn(b, HIGH); |
|  | } |
|  |  |
|  | void setup() |
|  | { |
|  | Serial.begin(9600); |
|  | pinMode(sensor, INPUT); |
|  | pinMode(light, OUTPUT); |
|  | pinMode(buzz, OUTPUT); |
|  | digitalWrite(light, LOW); |
|  | } |
|  |  |
|  | void loop() |
|  | { |
|  | //readUltrasonicDistance(7, 7) |
|  | dist = 0.01723 \* objectDistance(trig, echo); |
|  | Serial.print("Distance is "); |
|  | Serial.print(dist); |
|  | Serial.println("cm"); |
|  | if(dist>50 && dist<100) |
|  | { |
|  | tone(buzz, 50); |
|  | delay(2000); |
|  | noTone(buzz); |
|  | //delay(1000); |
|  | if(digitalRead(sensor)) |
|  | { |
|  | digitalWrite(light, HIGH); |
|  | delay(2000); |
|  | } |
|  |  |
|  |  |





**8 TESTING**

### 8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**8.1 TYPES OF TESTS**

**8.1.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.1.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**8.1.3 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**8.1.4 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.1.5 White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**8.1.6 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**8.2 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**8.2.1 Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**8.2.2 Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**8.2.3 Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# 8.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

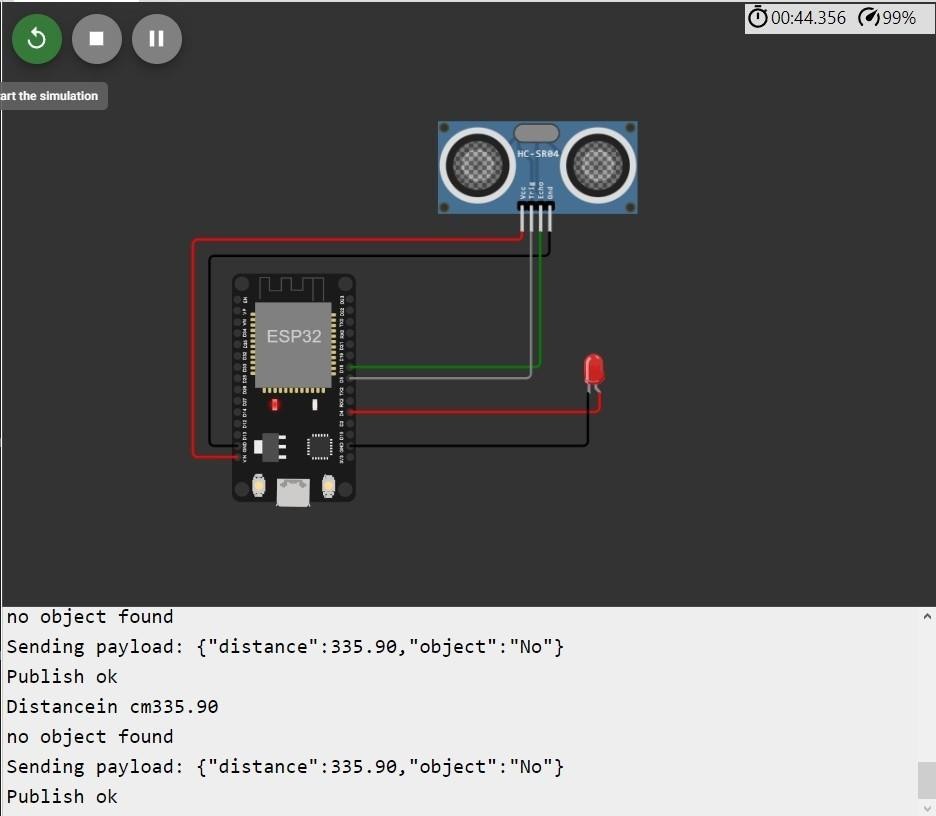
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

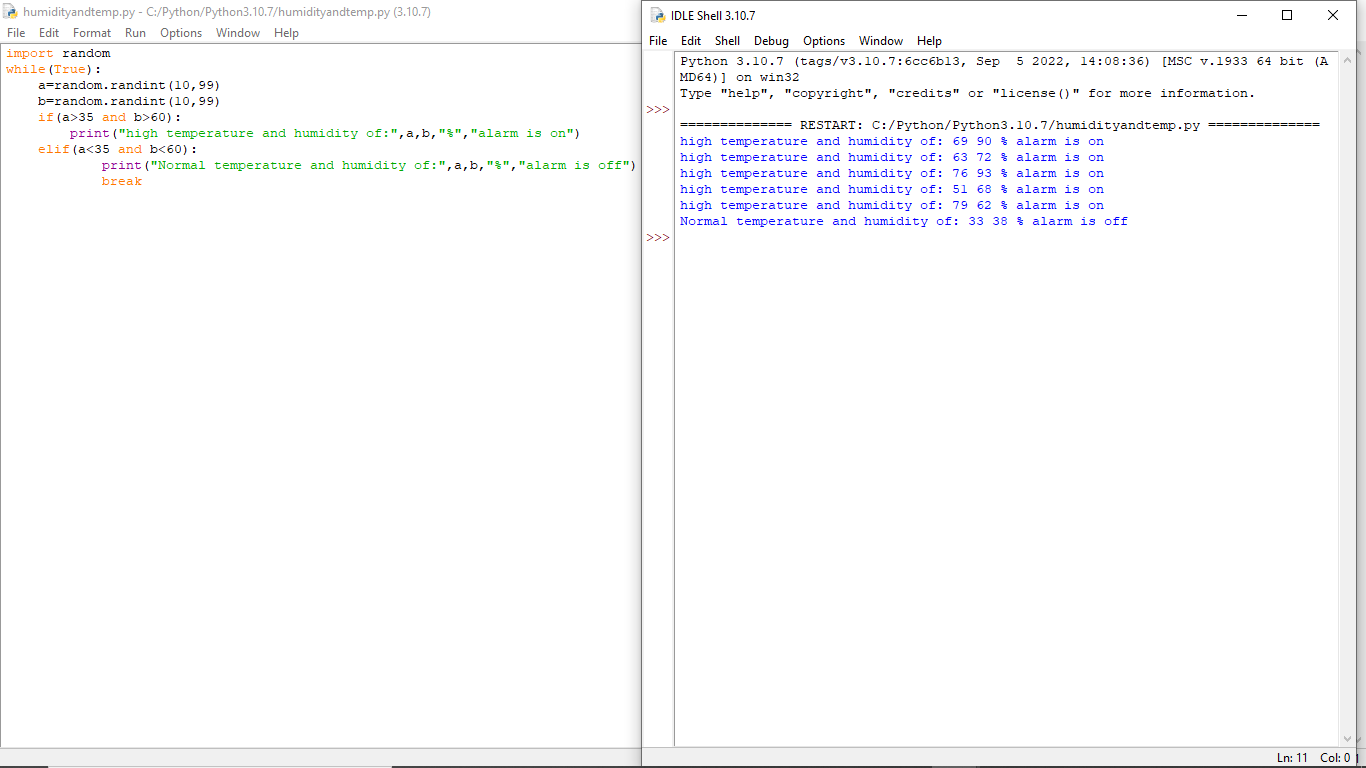
**8.4 Acceptance Testing**

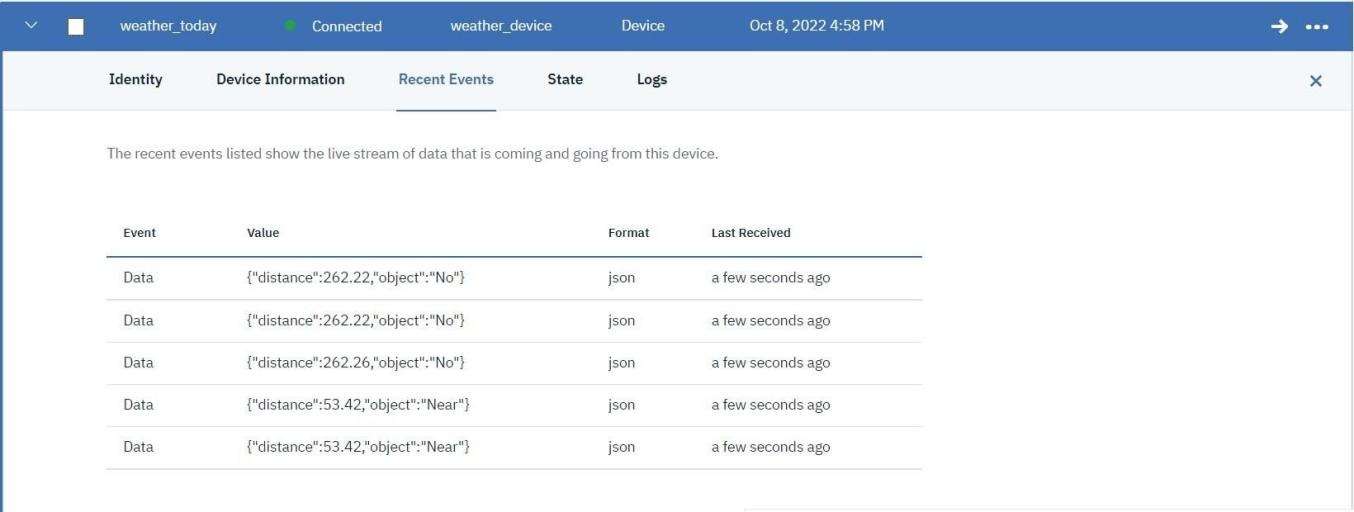
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**9 RESULTS**







**10 ADVANTAGES & DISADVANTAGES:**

**ADVANTAGES:**

#### Lower operating costs

#### Increased productivity and workplace safety

#### Better customer experiences

* High efficient
* User friendly
* Easy to install

**DISADVANTAGES:**

* Determining the right process.
* Feeling constrained.
* Spending more than necessary.
* Less efficient
* Covers short area.

**11 CONCLUSIONS**

IOT based smart agriculture system can prove to be very helpful for farmers since over as well as less irrigation is not good for agriculture. Threshold values for climatic conditions like humidity, temperature, moisture can be fixed based on the environmental conditions of that particular region. The system also senses the invasion of animals which is a primary reason for reduction in crops. 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. Continuous internet connectivity is required.

**12 FUTURE SCOPES:**

Indian agriculture is diverse ranging from impoverished farm villages to developed farms utilizing agricultural technologies. Promoting application of modern information technology in agriculture will solve a series of problems facing by farmers. Lack of exact information and communication leads to the loss in production. Our paper is designed to overcome these problems. This system provides an intelligent monitoring platform framework and system structure for facility agriculture ecosystem based on IOT.

**13 APPENDIXES:**

**SOURCE CODE:**

#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt

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

//-------credentials of IBM Accounts------

#define ORG "5qpnhq"//IBM ORGANITION ID

#define DEVICE\_TYPE "weather"//Device type mentioned in ibm watson IOT Platform #define DEVICE\_ID " weather1"//Device ID mentioned in ibm watson IOT Platform #define TOKEN "?4I@lLnUpZttANL9MsJ&M"

//Token String data3; float dist;

//-------- 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 LED = 4; int trig = 5; int echo = 18; void setup()

{

Serial.begin(115200); pinMode(trig,OUTPUT); pinMode(echo,INPUT); pinMode(LED, OUTPUT); delay(10); wificonnect(); mqttconnect();

}

void loop()// Recursive Function

{

digitalWrite(trig,LOW); digitalWrite(trig,HIGH); delayMicroseconds(10); digitalWrite(trig,LOW); float dur = pulseIn(echo,HIGH); float dist = (dur \* 0.0343)/2; Serial.print ("Distancein cm"); Serial.println(dist); PublishData(dist);

delay(1000); if (!client.loop()) { mqttconnect()

}

/\*.....................................retrieving to Cloud \*/

void PublishData(float dist) { mqttconnect();//function call for connecting to ibm

/\* creating the String in in form JSon to update the data to ibm cloud

\*/ String object; if (dist <100)

{

digitalWrite(LED,HIGH); Serial.println("object is near"); object = "Near";

}

else

{

digitalWrite(LED,LOW); Serial.println("no object found"); object = "No";

}

String payload = "{\"distance\":"; payload += dist; payload += "," "\"object\":\""; payload += object; 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]); data3 += (char)payload[i];

}

data3="";

}

**GITHUB LINK:**

https://github.com/IBM-EPBL/IBM-Project-49745-1660837504