Project ID: 23-24/IT/G2

A PROJECT THESIS ON

SMART AGRICULTURE SYSTEM USING IOT

Submitted
for fulfillment of award of
Bachelor of Technology
in
Information Technology
by

Aradhya Priyadarshi (2000270110021)
Harsh Tiwari (2000270130071)
Khushi Singh (2000270130087)

Under the guidance of

Mr. Sanjay Sonker

to

AJAY KUMAR GARG ENGINEERING COLLEGE, GHAZIABAD

May 2024

Declaration

We hereby declare that the work presented in this report, entitled "SMART AGRICULTURE SYSTEM USING IOT", was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors and sources for all the words, ideas, diagrams, graphics, computer programs, experiments, and results that are not my original contribution.We have used quotation marks to identify verbatim sentences and given credit to the original authors and sources.

We affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

Name : Aradhya Priyadarshi
Roll No. : 2000270110021

Name : Harsh Tiwari
Roll No. : 2000270130071

Name : Khushi Singh
Roll No. : 2000270130087

Certificate

This is to certify that the report entitled **SMART AGRICULTURE SYSTEM USING IOT** submitted by **Aradhya Priyadarshi (2000270110021)**, **Harsh Tiwari (2000270130071)** and **Khushi Singh (2000270130087)** to the Dr. A.P.J. Abdul Kalam Technical University, Lucknow in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Information Technology is a bonafide record of the project work carried out by us under my guidance and supervision. This report in any form has not been submitted to any other university or institute for any purpose, to the best of my knowledge.

Mr. Sanjay Sonker
Assistant Professor
Department of Information Technology
Ajay Kumar Garg Engineering College

Dr. Rahul Sharma
Professor & HOD
Department of Information Technology
Ajay Kumar Garg Engineering College

Place: Ghaziabad
Date: May 2024

Acknowledgement

Apart from our efforts, the success of the project depends largely on the encouragement and guidelines of many others.We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to express a deep sense of gratitude to Professor Rahul Sharma, our Head of the department, Dr. Sunil Kumar and Dr. Nitin Sharma, the project coordinators for their cordial support as they gave the permission to use all required equipment and the necessary material to complete the project.We would like to extend our sincerest gratitude to Mr. Sanjay Sonker for their guidance and supervision as well as for providing necessary information regarding the project and also for the support in completing the project.

Finally, We also extend our heartiest thanks to our parents, friends, and well-wishers for being with us and extending encouragement throughout the project.

Contents

Declaration . . . . **i**

Certificate . . . . **ii**

Acknowledgements . . . . **iii**

Abstract **1**

1 Introduction **2**

1.1 Problem Statement 2

1.2 Scope 5

1.3 Why is this topic chosen? 7

2 Literature Review **9**

3 Proposed Model **11**

3.1 NodeMCU 14

3.2 Arduino 17

3.3 Soil Moisture sensor 20

3.4 Temperature Sensor 22

3.5 Humidity Sensor 24

4 Designing of Project **26**

4.1 Zero and One Level DFD 26

4.2 Use Case Diagram 28

5 Methodology Used **29**

5.1 Class Diagram 31

5.2 Sequence Diagram 34

5.3 ER Diagram 37
5.4 Flowchart 40

6 Results and Discussions 42

7 Gantt Chart 47

8 Conclusion and Future Scope 48
8.1 Conclusion 48
8.2 Future Scope 49

Bibliography 51

A Research Paper 54

B UNAI Certificate 58

C Source Code 59

D Screenshots 85

List of Figures

4.1 Zero Level DFD for Smart Agriculture System 26
4.2 One Level DFD for Smart Agriculture System 27
4.3 Use Case for Smart Agriculture System 28

5.1 Class Diagram for Smart Agriculture System 31
5.2 Sequence Diagram for Smart Agriculture System for observing data on the current soil pH level 34
5.3 Sequence Diagram for Smart Agriculture System for observing data on soil moisture from a farm 35
5.4 Sequence Diagram for Smart Agriculture System for sending a Help Request 36
5.5 ER Diagram for Smart Agriculture System 37
5.6 Flowchart for Smart Agriculture System 40

6.1 Experimental Setup Architecture 43
6.2 Number of packets received by Blockchain per minute . . 44
6.3 Number of transactions published in Blockchain per minute 45

7.1 Gantt Chart 47

A.1 Acceptance Acknowledgemnt 54

C.1 Arduino 59

D.1 Dashboard : Real-time Monitoring 85
D.2 Monitoring Environmental Conditions and Crop Health . . 87

Abstract

In response to the rising global demand for food production, there is an imperative need to integrate innovative technologies into conventional agricultural practices. This abstract introduces a transformative Smart Agriculture System poised to revolutionize crop management and bolster sustainability within the agricultural sector. Leveraging advanced sensor technologies, Internet of Things (IoT) devices, and artificial intelligence, our system establishes a seamless ecosystem. This cohesive infrastructure actively monitors, analyzes, and optimizes diverse facets of crop cultivation, providing farmers with unprecedented insights and tools for informed decision-making.

Our proposed system utilizes state-of-the-art sensor technologies to gather real-time data on soil conditions, weather patterns, and crop health. This data is then transmitted via IoT devices to a centralized platform where it is analyzed using sophisticated artificial intelligence algorithms. These algorithms can detect patterns and predict potential issues such as pest infestations, water shortages, or nutrient deficiencies long before they become critical problems.

In conclusion, the implementation of this advanced Smart Agriculture System is crucial for the future of farming. It represents a significant leap forward in our ability to produce food in a sustainable manner, meeting the needs of a growing population while protecting our natural resources. Through the convergence of cutting-edge technologies, we can transform traditional agriculture into a more efficient, productive, and environmentally friendly enterprise.

Chapter 1

Introduction

1.1 Problem Statement

In the face of escalating global demand for food production and the imperative to ensure agricultural sustainability, there exists a pressing challenge: how can traditional agricultural practices be augmented to meet these demands while mitigating environmental impact? Traditional farming methods often lack real-time monitoring and predictive capabilities, leading to inefficiencies, resource wastage, and vulnerability to various risks such as pest infestations, water shortages, and nutrient deficiencies. Consequently, there is an urgent need to develop and implement a transformative Smart Agriculture System that integrates advanced sensor technologies, Internet of Things (IoT) devices, and artificial intelligence to enable proactive monitoring, analysis, and optimization of crop cultivation processes. This system aims to revolutionize crop management practices, empower farmers with actionable insights, and enhance overall agricultural productivity and sustainability.

Traditional agricultural practices, while time-tested and foundational, often fall short in addressing the complex challenges posed by modern-day food production needs. These methods typically rely on manual labor and observation, which can result in delayed responses to emerging issues such as pest outbreaks or soil nutrient depletion. The lack of precise data and real-time feedback mechanisms hampers the ability of farmers to make informed decisions, leading to suboptimal use of resources like water, fertilizers, and pesticides. Furthermore, traditional practices are often not scalable to meet the increasing food demands driven by population growth and urbanization.

The integration of advanced sensor technologies in a Smart Agriculture System can significantly mitigate these challenges. Sensors placed throughout the farming environment can continuously collect data on various parameters such as soil moisture, temperature, humidity, and light intensity. This real-time data collection allows for immediate detection of anomalies and facilitates timely interventions. For instance, if sensors detect a drop in soil moisture levels, automated irrigation systems can be activated to provide the necessary water, thereby preventing crop stress and ensuring optimal growth conditions.

IoT devices play a crucial role in connecting these sensors and enabling seamless communication between different components of the Smart Agriculture System. Through IoT, data collected from various sensors can be transmitted to a centralized platform where it can be analyzed using advanced algorithms. This connectivity ensures that all parts of the agricultural ecosystem are synchronized and can respond dynamically to changing conditions. Additionally, IoT devices can be used to control machinery such as drones and automated tractors, which can perform tasks like planting, spraying, and harvesting with high precision and efficiency.
Artificial intelligence (AI) is another pivotal component of the Smart Agriculture System. AI algorithms can analyze the vast amounts of data generated by sensors and IoT devices to identify patterns and predict future trends. For example, AI can forecast weather patterns, predict pest infestations, and recommend optimal planting times and crop varieties based on historical data and current conditions. These predictive capabilities enable farmers to adopt a proactive approach to crop management, reducing the risk of crop failure and enhancing yield.

Moreover, AI-powered decision support systems can provide farmers with actionable insights tailored to their specific needs. By integrating data on soil health, crop performance, and environmental conditions, these systems can suggest precise interventions such as the exact amount of fertilizer needed or the best time to apply pesticides. This precision farming approach minimizes resource wastage, reduces environmental impact, and maximizes productivity.

The implementation of a Smart Agriculture System holds the promise of transforming traditional farming practices into a more efficient, sustainable, and resilient agricultural model. By leveraging the power of advanced technologies, farmers can optimize resource use, improve crop yields, and ensure food security in the face of growing global challenges. This holistic approach not only benefits individual farmers but also contributes to the broader goal of sustainable development and environmental conservation.

In conclusion, the integration of advanced sensor technologies, IoT devices, and artificial intelligence into agriculture is essential for addressing the multifaceted challenges of modern food production. A Smart Agriculture System enables real-time monitoring, predictive analysis, and optimized resource allocation, revolutionizing crop management practices and enhancing agricultural sustainability. As the global population continues to rise, adopting these innovative technologies will be crucial in meeting the increasing food demand while preserving our natural resources for future generations.

Furthermore, smart agriculture can play a pivotal role in mitigating climate change impacts by promoting sustainable land use practices and enhancing carbon sequestration. By adopting these innovative approaches, agriculture can become more adaptable to changing climatic conditions, safeguarding food production against adverse weather events and contributing to global food security.

In summary, the Smart Agriculture System is a transformative approach that addresses the complex challenges of modern agriculture. Through the integration of cutting-edge technologies, it enhances efficiency, sustainability, and resilience, ensuring a prosperous future for farmers and a secure food supply for the growing global population. The journey towards smart agriculture is a collaborative effort that requires commitment, investment, and continuous innovation to achieve long-term success and sustainability in the agricultural sector.

1.2 Scope

1. IoT enables easy collection and management of tons of data collected from sensors: With the integration of cloud computing services like agriculture field maps, cloud storage, and data analytics platforms, data can be accessed live from anywhere and everywhere. This enables live monitoring and end-to-end connectivity among all the parties concerned. Farmers can monitor soil conditions, weather forecasts, and crop health in real-time, making it easier to make informed decisions promptly. This real-time access to data ensures that any anomalies or issues can be addressed immediately, reducing the risk of crop damage and optimizing the use of resources such as water and fertilizers.

2. IoT is regarded as a key component for Smart Farming: With accurate sensors and smart equipment, farmers can increase food production by 70 percent by the year 2050, as depicted by experts. IoT technologies provide precision agriculture tools that allow for the precise application of water, fertilizers, and pesticides, which enhances crop yields and quality. The ability to collect and analyze data on crop performance, soil conditions, and environmental factors allows farmers to implement best practices and innovations that drive productivity. The adoption of IoT in agriculture not only supports increased food production but also promotes sustainable farming practices that are vital for long-term agricultural success.

3. With IoT, production costs can be reduced to a remarkable level: This will, in turn, increase profitability and sustainability. Lower production costs can be achieved through the efficient use of resources, reduction of waste, and automation of labor-intensive tasks. By leveraging IoT technologies, farmers can build other advanced systems such as:

- Intelligent Soil Cultivation System: This system would plow, weed, prepare the seedbed, and sow the field soil in order to prepare it for harvest. Automated soil cultivation systems use sensors to monitor soil health and ensure optimal conditions for planting, which reduces the need for manual labor and increases the precision of soil preparation activities.

- Efficient Irrigation Mechanisms: This technology would automate the controlled artificial supply of water needed for plant growth. Smart irrigation systems use sensors to monitor soil moisture levels and weather conditions, applying water only when and where it is needed. This reduces water waste, ensures plants receive the right amount of water, and enhances crop yield.

- Smart Fertilizer Systems: These systems automate the application of fertilizer to a field while allowing for precise control over the kind, amount, and timing of fertilizer. By using data from soil sensors and crop health monitors, these systems ensure that crops receive the exact nutrients they need at the right time, improving growth and reducing the environmental impact of over-fertilization.

- Intelligent Pest Detection and Treatment Systems: These systems keep an eye out for pest infestations, analyze agricultural damage, and incorporate methods to manage the infestation. By using advanced imaging and sensor technologies, these systems can detect pests early, apply targeted treatments, and reduce the use of chemical pesticides, promoting a healthier and more sustainable farming environment.

- Smart Farm Management System: This kind of technology would aim to offer analytics on data to increase field productivity and yield. By integrating various data sources, such as weather forecasts, soil conditions, and crop health, smart farm management systems provide comprehensive insights that help farmers make better decisions and optimize their operations.

- Intelligent Groundwater Quality Management System: The final product is greatly influenced by the quantity and quality of the groundwater. As a result, this system uses IoT approaches to maintain appropriate groundwater levels. Sensors monitor groundwater quality and levels, ensuring that crops receive clean water and preventing over-extraction, which can deplete water resources and harm the environment.

Overall, the implementation of IoT in agriculture represents a significant advancement in how farming operations are conducted. By harnessing the power of data and connectivity, IoT technologies enable smarter, more efficient, and sustainable agricultural practices that can meet the growing demands for food production while minimizing the environmental impact.

1.3 Why is this topic chosen?

Every aspect of the 21st century has undergone a revolution because of the Internet of Things (IoT) and smart computing technologies. These technologies are applied in many different ways, from monitoring the state of crops and the moisture level of the soil in real-time to using drones to help with chores such as spraying pesticides. The extensive integration of both recent IT and conventional agriculture has brought in the phase of agriculture 4.0, often known as smart agriculture. Agriculture intelligence and automation are addressed by smart agriculture, encompassing a wide array of advanced tools and systems to enhance productivity and sustainability in farming practices.

1. Population Growth and Hunger: With the global population projected to exceed two billion, particularly in urban areas, achieving Sustainable Development Goals (SDGs), such as zero hunger, becomes a major challenge. This surge in population will significantly increase the demand for food, putting immense pressure on existing agricultural systems to produce more with limited resources. Therefore, innovative solutions through smart agriculture are crucial to meet this growing demand and ensure food security.

2. Resource Scarcity: Scarcity of water and available farming land presents substantial obstacles. Meeting 40 percent of the water supply by 2030 is challenging, and there is insufficient farming land to support the food supply. This necessitates more efficient use of available resources. Smart agriculture aims to address these issues by optimizing resource use through precise irrigation systems, soil health monitoring, and crop rotation strategies that ensure sustainability.

3. Efficiency and Resource Allocation: Agriculture 4.0 aims to optimize crop cultivation using environmental monitoring, forecasting, and smart devices. It is vital to efficiently use resources like water and energy. Advanced technologies enable farmers to analyze vast amounts of data to make informed decisions about planting, watering, and harvesting. This leads to increased yield, reduced waste, and a more sustainable farming practice overall. Efficient resource allocation not only helps in maximizing productivity but also minimizes the environmental impact of agricultural activities.

4. Security Risks in Agriculture: Wireless Sensor Networks (WSNs) and IoT technologies in Agriculture 4.0 offer significant advantages but are susceptible to security risks due to obsolete or unpatched firmware and software. The connectivity of various devices in smart agriculture systems makes them vulnerable to cyber-attacks, which can disrupt operations and lead to data breaches. Ensuring robust cybersecurity measures and regular updates are essential to protect these systems from potential threats and maintain their integrity and reliability.

These problem statements highlight the multifaceted challenges and security concerns associated with modern agriculture, particularly in the context of Agriculture 4.0. Addressing these issues requires a collaborative effort from technology developers, policymakers, and the farming community to leverage the benefits of smart agriculture while mitigating its risks. This holistic approach will be instrumental in achieving sustainable and resilient agricultural systems for the future.

Chapter 2

Literature Review

This research by Vangala explores the integration of blockchain technology for authentication in smart farming. The study highlights the importance of secure authentication methods in modern agriculture, emphasizing the use of smart contracts. The authors propose an innovative scheme that leverages blockchain's transparency and security features to authenticate users and devices in smart farming systems.

1. Security and Privacy in Smart Farming: Challenges and Opportunities (2020): Gupta address the essential aspects of security and privacy in smart farming. The paper identifies challenges and opportunities in this domain, focusing on safeguarding sensitive data and ensuring the integrity of smart farming systems. It highlights the need for robust security measures to protect against potential threats.

2. Innovating Digitally The New Texture of Practices in Agriculture 4.0 (2022): Lioutas and Charatsari delve into the digital innovations shaping Agriculture 4.0. The research explores how technology is revolutionizing agricultural practices, emphasizing the digital transformation of the industry. It sheds light on the integration of advanced technologies like IoT, AI, and automation in modern farming.

3. IoT Based Smart Crop-Field Monitoring and Automation Irrigation System (2018): Rao and Sridhar present an IoT-based smart crop-field monitoring and automation irrigation system. This research focuses on the practical implementation of IoT technology in agriculture. It outlines a system for monitoring crop fields and automating irrigation, showcasing the potential for increased efficiency and resource optimization.

4. Challenges for Agriculture Through Industry 4.0 (2021): Bernhardt discuss the challenges faced by agriculture in the context of Industry 4.0. The paper explores the integration of advanced technologies and automation in agriculture, highlighting the potential benefits and obstacles. It offers valuable insights into the transformation of farming practices.

5. Agrifusion: An Architecture for IOT And Emerging Technologies Based on A Precision Agriculture Survey (2021): Singh introduce the AgriFusion architecture, designed to harness IoT and emerging technologies in precision agriculture. The research focuses on creating a comprehensive framework for integrating advanced technologies into farming practices.

6. Role Of Internet Of Things For Adopting And Promoting Agriculture 4.0 (2021): Raj provide a survey of the role of the Internet of Things (IoT) in promoting Agriculture 4.0. The paper explores how IoT technologies are driving the adoption of modern farming.

7. Self-Automated Agriculture System Using IOT (2020): Krishnan present a self-automated agriculture system that leverages IoT technology. The research focuses on practical implementations of IoT in agriculture, emphasizing automation and control systems. It demonstrates how IoT can be harnessed for efficient resource management in farming.

The work provides a comprehensive overview of IoT-based smart irrigation systems. They discuss the recent trends in sensor technology and IoT systems for precision agriculture. The paper emphasizes the critical role of sensors and IoT in optimizing irrigation practices, ultimately contributing to resource efficiency and increased crop yields.

These literature reviews collectively provide a comprehensive understanding of various aspects of smart farming, including technology integration, security, IoT applications, and the digital transformation of agriculture. Researchers and practitioners can draw valuable insights from these works to further advance the field of smart farming.

Chapter 3

Proposed Model

1. Sensor Technologies Integration: The Smart Agriculture System will integrate state-of-the-art sensor technologies to gather real-time data on crucial agricultural parameters, including soil conditions, weather patterns, and crop health. These sensors will be deployed strategically across farmland to ensure comprehensive coverage and accurate data collection. By monitoring factors such as soil moisture, temperature, and nutrient levels, these sensors will provide a detailed and dynamic view of the agricultural environment. This comprehensive data collection will be the foundation for precise and informed decision-making in farming practices.

2. Internet of Things (IoT) Connectivity: The collected data from the sensors will be transmitted wirelessly via IoT devices to a centralized platform. This connectivity will enable seamless data transfer and communication between various components of the agricultural ecosystem, facilitating real-time monitoring and analysis. IoT devices will ensure that data flows smoothly from the field to the cloud, where it can be processed and analyzed. This interconnected network of devices will create a cohesive system that responds dynamically to changing conditions, ensuring optimal management of resources and activities.

3. Centralized Data Analytics Platform: A centralized data analytics platform will be established to process the incoming data streams from the sensors. Using sophisticated artificial intelligence algorithms, this platform will analyze the data to detect patterns, identify trends, and predict potential issues such as pest infestations, water shortages, or nutrient deficiencies. By leveraging machine learning and big data analytics, the platform will provide deep insights into crop performance and environmental conditions, enabling proactive management and early intervention to prevent problems.

4. Predictive Insights and Decision Support: Leveraging the analytical capabilities of the system, farmers will be provided with actionable insights and decision support tools. These insights will enable farmers to make informed decisions regarding crop management practices, resource allocation, and mitigation strategies for potential risks. For example, predictive analytics can forecast weather conditions and recommend optimal times for planting or harvesting, while decision support tools can suggest the best types and amounts of fertilizers to use based on soil analysis.

5. Automation and Optimization: The Smart Agriculture System will support the automation and optimization of various farming processes based on real-time data and predictive analytics. This may include automated irrigation systems, precision application of fertilizers and pesticides, and adaptive crop management strategies tailored to specific environmental conditions. Automated systems will ensure that resources are used efficiently and effectively, reducing waste and enhancing crop yields. For instance, smart irrigation systems can adjust water supply based on soil moisture levels, while automated pest control systems can target infestations precisely, minimizing the use of chemicals.

6. Scalability and Flexibility: The proposed model will be designed to be scalable and adaptable to diverse agricultural settings and crop types. Whether in small-scale subsistence farming or large-scale commercial agriculture, the system will be flexible enough to accommodate varying needs and requirements. This adaptability ensures that the benefits of smart agriculture can be realized across different farming contexts, from local family farms to expansive industrial operations. The system's scalability will allow it to grow with the needs of the agricultural sector, providing ongoing support and innovation as demands evolve.

7. Continuous Improvement and Innovation: The Smart Agriculture System will be developed as an evolving platform, with provisions for continuous improvement and innovation. Feedback mechanisms and data-driven insights will be utilized to refine algorithms, enhance predictive capabilities, and incorporate emerging technologies to further optimize agricultural productivity and sustainability. By continuously integrating new research findings, technological advancements, and user feedback, the system will remain at the forefront of agricultural innovation, driving ongoing improvements in efficiency and effectiveness.

Overall, the proposed Smart Agriculture System represents a comprehensive and transformative approach to modernizing conventional farming practices. By harnessing the power of advanced technologies such as sensors, IoT, and artificial intelligence, this system aims to revolutionize crop management, enhance productivity, and promote sustainability in the agricultural sector. This holistic approach not only addresses immediate challenges but also lays the groundwork for a resilient and sustainable future in agriculture. By implementing such a system, farmers can achieve greater yields, reduce environmental impact, and ensure food security for the growing global population.

Technologies used

- HTML, CSS, Javascript, Reactjs - Frontend
- SQLite - Database
- NodeMCU
- Arduino
- Soil Moisture sensor
- Temperature Sensor
- Humidity Sensor

3.1 NodeMCU

NodeMCU is an open-source IoT platform based on the ESP8266 Wi-Fi module. It integrates a microcontroller with Wi-Fi capability, making it an ideal choice for IoT projects. Here's a detailed description of NodeMCU and its application in smart agriculture systems:

What is NodeMCU? NodeMCU is a low-cost, low-power microcontroller board that features:

ESP8266 Wi-Fi Module: Provides robust Wi-Fi capabilities, allowing for easy connection to wireless networks and communication with other IoT devices.

Microcontroller: A Tensilica Xtensa LX106 core, which can run at 80 or 160 MHz and includes 32KB instruction RAM, 80KB user data RAM, and 4MB flash memory.

GPIO Pins: Multiple General Purpose Input/Output pins that can interface with sensors, actuators, and other hardware components.

Lua Scripting Language: Originally, NodeMCU firmware used Lua, but it also supports the Arduino IDE, making it accessible for many developers.

USB Port: Facilitates easy programming and power supply via a USB connection.

NodeMCU in Smart Agriculture Systems Smart agriculture leverages IoT technologies to enhance farming practices. NodeMCU plays a pivotal role in creating efficient, automated, and data-driven agricultural systems. Here's how:

1. Environmental Monitoring

Soil Moisture Sensors: NodeMCU can interface with soil moisture sensors to monitor soil water levels. Data is sent to a cloud server or a local database for real-time analysis.

Temperature and Humidity Sensors: By connecting these sensors, NodeMCU can provide critical data about environmental conditions, helping in climate control within greenhouses or open fields.

2. Automated Irrigation Systems

Control Relays: NodeMCU can control water pumps and irrigation valves through relays based on soil moisture readings. This ensures optimal watering schedules, conserving water and improving crop yields.

Real-time Adjustments: Automated systems can adjust irrigation based on weather forecasts or real-time sensor data, minimizing water wastage and enhancing crop health.

3. Pest and Disease Management

Image Processing: By integrating with cameras and machine learning models, NodeMCU can help identify pests and diseases in crops. Early detection allows for timely intervention, reducing crop damage.

Environmental Conditions: Monitoring conditions conducive to pest and disease outbreaks can help in preemptive actions.

4. Data Logging and Analysis

Cloud Integration: NodeMCU can send data to cloud platforms like ThingSpeak, Firebase, or custom servers. Farmers can access and analyze this data via web or mobile applications.

Predictive Analytics: Historical data analysis can predict trends, such as when certain pests might appear or when crops need more water.

5. Cost-Effective and Scalable

Affordable: NodeMCU's low cost makes it accessible for small and large-scale farmers.

Scalability: Multiple NodeMCUs can be deployed across large fields, providing comprehensive data coverage and control.

6. Remote Monitoring and Control

Smartphones and PCs: Farmers can monitor and control their systems remotely via smartphones or computers, receiving alerts and updates in real-time.

Integration with Other IoT Devices: NodeMCU can work with other IoT devices like weather stations, drones, and automated machinery, creating a cohesive smart farming ecosystem.

In summary, NodeMCU provides a flexible, cost-effective, and efficient solution for implementing smart agriculture systems. It enables precise monitoring and control of various farming processes, leading to enhanced productivity, resource efficiency, and sustainable agricultural practices.

3.2 Arduino

Arduino enables the creation of smart agriculture systems by facilitating the integration and automation of various farming processes. Here's how Arduino is utilized in smart agriculture:

1. Environmental Monitoring

Soil Moisture Sensors: Arduino can read data from soil moisture sensors to monitor soil water content. This data helps in determining the right irrigation schedules.

Temperature and Humidity Sensors: By connecting sensors like DHT11 or DHT22, Arduino can monitor environmental conditions crucial for crop health.

Light Sensors: Arduino can use light sensors (e.g., LDR, photodiodes) to measure sunlight exposure, helping in understanding plant growth conditions.

2. Automated Irrigation Systems

Control Relays: Arduino can control irrigation pumps and valves through relays based on sensor data, ensuring plants receive the right amount of water.

Irrigation Scheduling: Programmable timers and real-time clocks (RTCs) allow Arduino to implement precise irrigation schedules.

3. Pest and Disease Management

Camera Integration: By integrating with camera modules and image processing algorithms, Arduino can help in detecting pests and diseases.

Sensor Networks: Creating a network of sensors to monitor environmental conditions that may lead to pest and disease outbreaks.

4. Data Logging and Analysis

SD Card Modules: Arduino can store sensor data on SD cards for later analysis.

Wireless Communication: Using modules like ESP8266/ESP32, GSM, or RF, Arduino can transmit data to cloud platforms for remote monitoring and analysis.

5. Weather Stations

Weather Sensors: Arduino can be used to build weather stations that measure parameters like temperature, humidity, atmospheric pressure, rainfall, and wind speed.

Data Collection and Transmission: Collected data can be used to make informed decisions about planting, irrigation, and harvesting.

6. Automation and Control

Greenhouse Automation: Arduino can control various aspects of a greenhouse, such as ventilation, heating, cooling, and lighting, based on sensor feedback.

Robotic Systems: Arduino can be used to develop robotic systems for tasks like planting, weeding, and harvesting, reducing labor and increasing efficiency.

7. Cost-Effective and Customizable

Affordable: Arduino boards are relatively inexpensive, making them accessible for small-scale farmers and hobbyists.

Customizable: The open-source nature of Arduino allows for extensive customization to meet specific agricultural needs.

8. Remote Monitoring and Alerts

SMS and Email Alerts: Using GSM modules, Arduino can send SMS alerts to farmers about critical conditions (e.g., low soil moisture, extreme temperatures).

Mobile Apps: Data from Arduino can be sent to mobile apps, allowing farmers to monitor their fields remotely and make timely decisions.

Practical Applications in Smart Agriculture

Precision Farming: Use of precise data to optimize planting, watering, and harvesting, leading to increased yield and resource efficiency.

Climate Control: Automated control of greenhouse climates, ensuring optimal growing conditions year-round.

Livestock Monitoring: Monitoring animal health and environment conditions to improve livestock management.

In summary, Arduino provides a flexible, cost-effective, and user-friendly platform for developing smart agriculture systems. It enables farmers to implement precise monitoring and control mechanisms, leading to improved crop management, resource conservation, and increased agricultural productivity.

3.3 Soil Moisture sensor

A soil moisture sensor is a sophisticated device used to measure the volumetric water content in soil. It is an essential component in smart agriculture systems, enabling precise monitoring of soil moisture levels to optimize irrigation practices and improve crop health. The accurate and real-time data provided by these sensors allow farmers to make informed decisions about watering schedules, ensuring that crops receive the right amount of water at the right time.

A typical soil moisture sensor consists of a pair of conductive probes or a capacitive plate. When inserted into the soil, the sensor measures the electrical properties that change with soil moisture levels. These properties, such as electrical resistance or capacitance, vary depending on the amount of water present in the soil. The sensor outputs an analog or digital signal proportional to the moisture content. This signal can then be interpreted by connected devices, such as microcontrollers or data loggers, to provide real-time insights into soil conditions.

Soil moisture sensors are crucial for implementing smart agriculture systems, enabling efficient water management, and improving crop health. By continuously monitoring soil moisture levels, these sensors help prevent overwatering or underwatering, which can lead to crop stress, reduced yields, and wasted resources. Optimal water management not only enhances plant growth and productivity but also conserves water, a critical resource in many agricultural regions.

Integrating soil moisture sensors with platforms like Arduino allows farmers to create automated and data-driven solutions to optimize their farming practices. For instance, an Arduino-based system can be programmed to read data from the moisture sensor and activate irrigation systems when moisture levels fall below a certain threshold. This automation ensures that crops receive water precisely when needed, reducing the labor and guesswork involved in traditional irrigation methods.

Moreover, soil moisture sensors can be linked to larger data analytics platforms that aggregate information from multiple sensors across a farm. This aggregated data can provide a comprehensive view of soil moisture distribution, helping farmers identify patterns and areas that require special attention. Advanced algorithms can analyze this data to offer predictive insights, such as anticipating dry spells or periods of high water demand, allowing farmers to plan their irrigation schedules proactively.

By adopting these advanced technologies, farmers can enhance sustainability and increase productivity. Efficient water use reduces the environmental impact of farming practices, conserving water resources and reducing runoff that can lead to soil erosion and water contamination. Healthier crops, resulting from optimal irrigation, contribute to higher yields and better-quality produce, supporting food security and profitability.

In summary, soil moisture sensors are a vital tool in the advancement of smart agriculture. They provide precise, real-time data that enables efficient water management, enhances crop health, and supports sustainable farming practices. When integrated with platforms like Arduino, these sensors form the backbone of automated and data-driven agricultural systems, paving the way for a more productive and environmentally friendly approach to farming. By leveraging such technologies, the agricultural sector can meet the growing global food demand while preserving vital natural resources for future generations.

3.4 Temperature Sensor

Temperature sensors are essential components in smart agriculture systems, providing critical data for monitoring and managing the environmental conditions that affect crop growth and health. Understanding and controlling temperature is crucial because it directly influences plant metabolism, photosynthesis rates, and overall crop development. Accurate temperature data helps farmers make informed decisions to ensure optimal growing conditions.

Temperature sensors measure temperature by using the physical properties of materials that change with temperature. These properties might include resistance, voltage, or current, which vary predictably with changes in temperature. Common types of temperature sensors used in agriculture include thermistors, thermocouples, and semiconductor sensors. When exposed to different temperatures, these sensors convert the thermal variations into an electrical signal (voltage, resistance, or current) that can be read and processed by a microcontroller or other digital devices. This data conversion is crucial for integrating temperature measurements into automated agricultural systems.

Temperature sensors are vital for implementing smart agriculture systems. They enable precise monitoring and control of environmental conditions, contributing to better crop management, improved resource efficiency, and enhanced productivity. For example, temperature data can be used to control greenhouse climates, ensuring that plants are grown in optimal temperature ranges. Maintaining the correct temperature can help in avoiding conditions that could stress plants, such as excessive heat or frost, both of which can significantly impact crop yields.

By integrating temperature sensors with platforms like Arduino, farmers can create automated solutions to optimize their farming practices. An Arduino-based system can continuously monitor temperature data and trigger actions such as adjusting ventilation, heating, or cooling systems in response to temperature fluctuations. This automation not only reduces the manual labor required for managing environmental conditions but also ensures that interventions are timely and precise, leading to healthier plants and better yields.

Moreover, temperature sensors can be part of a broader network of environmental sensors that collectively provide a comprehensive picture of the farm's microclimate. When combined with other sensors, such as soil moisture sensors, humidity sensors, and light sensors, temperature sensors help create a detailed environmental profile. This integrated approach allows for sophisticated data analytics and modeling, which can predict potential issues and suggest optimal farming strategies.

The benefits of using temperature sensors in agriculture extend beyond immediate crop health. Improved temperature management can lead to more efficient use of resources such as water and energy. For instance, controlling temperature in greenhouses can reduce the need for excessive watering or heating, conserving resources and lowering operational costs. Additionally, better temperature control can extend growing seasons and increase the variety of crops that can be cultivated in a given area, enhancing the overall productivity and profitability of the farm.

In conclusion, temperature sensors are a crucial technology in the advancement of smart agriculture. They provide precise, real-time data that enables effective monitoring and control of environmental conditions, essential for optimizing crop growth and health. When integrated with platforms like Arduino, temperature sensors facilitate automated and data-driven agricultural practices. This integration leads to sustainable and profitable farming, capable of meeting the demands of a growing global population while preserving the environment. By adopting these technologies, farmers can ensure that their practices are both efficient and sustainable, securing the future of agriculture.

3.5 Humidity Sensor

Humidity sensors are crucial components in smart agriculture systems, providing vital data for monitoring and managing environmental conditions. Accurate humidity measurement is essential for optimizing plant growth, preventing diseases, and ensuring the overall health of crops. Humidity levels affect a variety of plant processes, including transpiration, nutrient uptake, and disease susceptibility. By monitoring humidity, farmers can create ideal growing conditions, enhance crop quality, and increase yields.

Humidity sensors measure the amount of water vapor in the air. They typically work by using materials that change their electrical properties based on the humidity level. Common types of humidity sensors include capacitive, resistive, and thermal conductivity sensors. These sensors convert the humidity level into an electrical signal, which can be read and processed by a microcontroller or other digital devices. This signal provides a precise measurement of the ambient humidity, allowing for real-time monitoring and control.

Humidity sensors are vital for implementing effective smart agriculture systems. They enable precise monitoring and control of environmental humidity, contributing to better crop management, disease prevention, and resource efficiency. For example, high humidity levels can create favorable conditions for fungal and bacterial diseases, which can devastate crops. By using humidity sensors, farmers can detect these conditions early and take preventive measures, such as adjusting ventilation or applying fungicides, to protect their crops.

Moreover, humidity sensors help optimize irrigation practices. Understanding the relationship between soil moisture and air humidity can guide irrigation schedules, ensuring that plants receive the right amount of water without over-irrigation, which can lead to root diseases and water wastage. This precise control of water use not only conserves a critical resource but also improves plant health and productivity.

By integrating humidity sensors with platforms like Arduino, farmers can develop automated and data-driven solutions to optimize their farming practices. For instance, an Arduino-based system can monitor humidity levels and automatically control ventilation systems, irrigation systems, and other environmental controls to maintain optimal humidity conditions. This automation reduces the need for manual interventions, saving labor and ensuring that environmental conditions are consistently managed.

Furthermore, humidity sensors can be part of a broader network of environmental sensors that work together to provide a comprehensive view of the farm's microclimate. When combined with temperature sensors, soil moisture sensors, and light sensors, humidity sensors enable a detailed analysis of the growing environment. This integrated data can be used to develop predictive models and decision support systems, helping farmers make proactive adjustments to their practices.

The benefits of using humidity sensors in agriculture extend beyond immediate crop health. Improved humidity management can lead to more efficient use of resources such as water and energy. For example, by controlling humidity levels, greenhouses can reduce the need for excessive watering or heating, conserving resources and lowering operational costs. Additionally, maintaining optimal humidity levels can enhance the quality and shelf-life of harvested produce, increasing market value and reducing post-harvest losses.

In conclusion, humidity sensors are a crucial technology in the advancement of smart agriculture. They provide precise, real-time data that enables effective monitoring and control of environmental conditions, essential for optimizing plant growth and health. When integrated with platforms like Arduino, humidity sensors facilitate automated and data-driven agricultural practices. This integration leads to sustainable and profitable farming, capable of meeting the demands of a growing global population while preserving the environment. By adopting these technologies, farmers can ensure that their practices are both efficient and sustainable, securing the future of agriculture.

Chapter 4

Designing of Project

4.1 Zero and One Level DFD

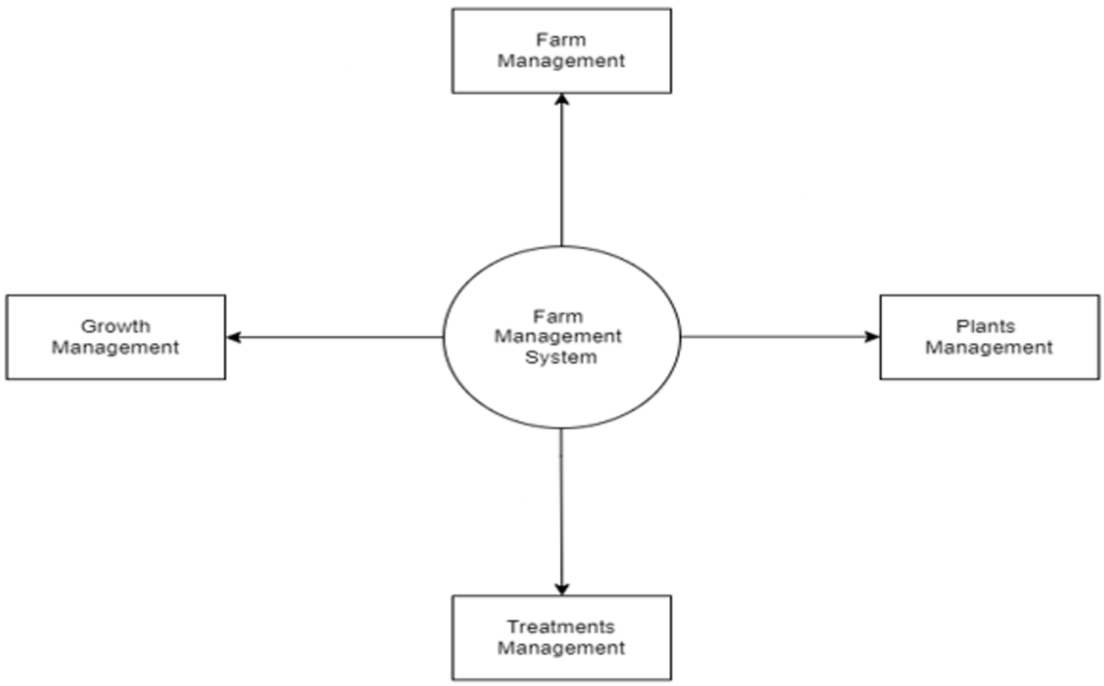

Figure 4.1: Zero Level DFD for Smart Agriculture System

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows. This is the 0-level of the Farm management system, where we have elaborated the high-level processes of the farm. It's designed to be an at-a glance view of Growth, user, login showing the system as a single high-level process, with it's relationship to external entities of farm, treatments.

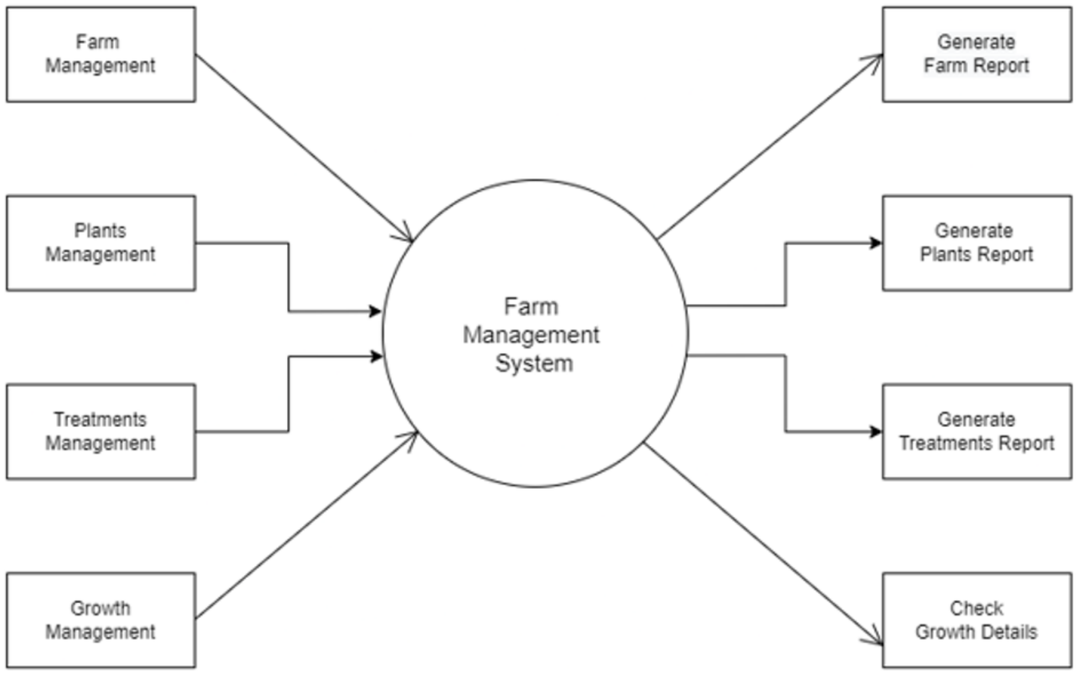

Figure 4.2: One Level DFD for Smart Agriculture System

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes. 1-level DFD of Farm management system shows how the system is divided into sub-systems, each of which deals with one or more of the data flow to or from an external agent, and which together provide all the functionality of the farm management system as a whole.

4.2 Use Case Diagram

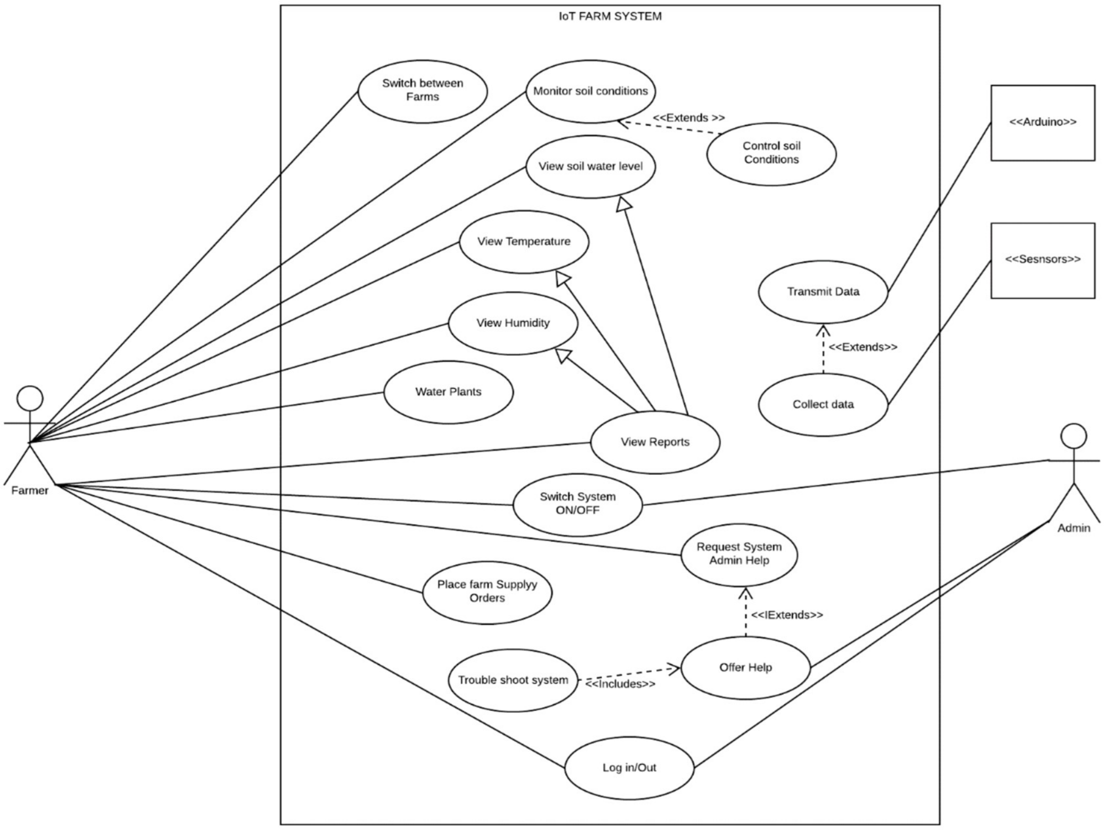

Figure 4.3: Use Case for Smart Agriculture System

To model functionality of the system, we use different actors, external entities ("roles"), and the associated use cases represented on the use case diagram. The use case diagrams are also used to show the functions, actions and services that the systems will perform.
The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements ac countable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.
Following are the purposes of a use case diagram given below:
1)It gathers the system's needs.
2)It depicts the external view of the system.
3)It recognizes the internal as well as external factors that influence the system.
4)It represents the interaction between the actors.

Chapter 5

Methodology Used

1. **Requirements Assessment and Requirements Gathering**:- Conduct a thorough requirements assessment to understand the challenges and needs of the agricultural community.- Identify key stakeholders, including farmers, agronomists and agronomists, to gather information on needs and priorities.- Define the goals and scope of the Smart Farming System project based on identified needs and requirements.

2. **Research and Technology Assessment:-** Conduct comprehensive research on IoT technologies, mobile devices and agricultural monitoring systems.- Assess the effectiveness of various sensor technologies to monitor soil conditions, weather patterns, crop health and other relevant parameters.- Explore IoT communication protocols and platforms for data transmission and connectivity.

3. **System Design and Architecture:-** Develop a system architecture that describes the components, interfaces and interactions of an intelligent agriculture system.- Design a sensor network to collect real-time data in agriculture, taking into account aspects such as sensor placement, coverage, and scale.- Defines the communication infrastructure, including IoT gateways, data transfer protocols, and cloud-based and analytical platforms.

4. **Speaker Design and Installation:**- Send the sensor equipment to the agricultural field according to the sensor network design.- Installs an IoT gateway and establishes a connection between the sensors and the compiled database.- Sensor devices must be properly adjusted and configured for accurate data collection.

5. **Data Collection and Transmission:-** Implement a data collection method to collect real-time data from deployed sensors.- Develop protocols

to efficiently send sensor data to a central database.- Uses data processing techniques to clean and validate incoming data streams before storing and analyzing them.

6. **Data Storage and Management:-** Establish a centralized data storage infrastructure, such as a cloud-based database or data lake, to store incoming sensor data.- Define data management policies, including data retention period, access control and data backup strategy.- Complete data encryption and security measures to protect sensitive agricultural data against unauthorized access or modification.

7. **Data Analysis and Knowledge Generation:-** Develop machine learning models and algorithms to analyze sensor data and extract meaningful information.- Apply predictive analytics to detect patterns, trends and trends in agricultural data.- Create information and recommendations for farmers based on sensor data analysis, such as irrigation schedules, pest management strategies and crop optimization techniques.

8. **User Interface and Visualization:-** Design simple interfaces and dashboards to view sensor data, analysis results, and useful insights.- Develops interactive tools and visualization techniques to help farmers explore and interpret agricultural data.- Includes functions to notify users of important events or deviations from normal, such as severe weather warnings or dangerous outbreaks.

9. **Testing and Validation:-** Conduct pilot tests of the smart farming system in a real farming environment to evaluate its functionality and performance.- Collect feedback from end users, including farmers and agronomists, to identify areas for improvement and improvement.- Verify the effectiveness of the system to improve indicators of agricultural productivity, resource efficiency and sustainability.

10. **Scaling and Scaling:-** Deploy smart agriculture systems to a wider range of agricultural areas and farms, taking into account scale and resource requirements.- Provide training and support to farmers and agricultural stakeholders to use the system and interpret data-based insights.- Monitor system performance and scalability over time and update design and implementation as needed to meet changing requirements and challenges.

5.1 Class Diagram

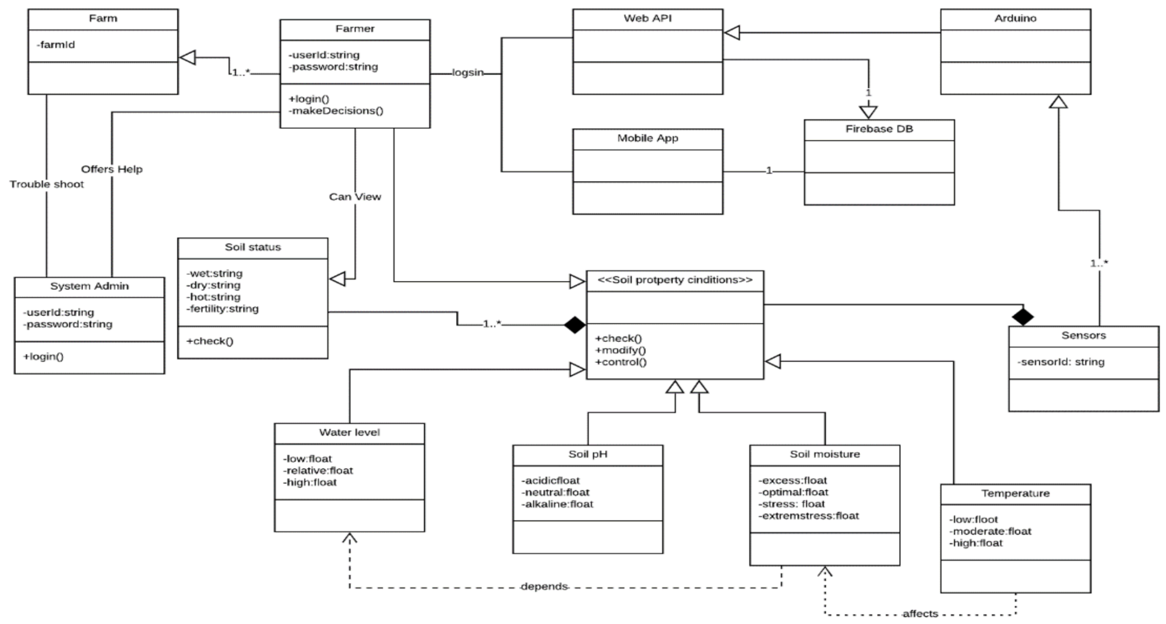

Figure 5.1: Class Diagram for Smart Agriculture System

Here's a breakdown of the system:

Farm class: This class represents the physical farm land. It has an attribute, farmID, which is likely a unique identifier for the farm.

Farmer class: This class represents the person managing the farm. It has attributes, userID and password, which are likely used for the farmer to log in to the system. The class also has a method, login(), which would presumably be used for the farmer to authenticate themselves.

System Admin class: This class presumably represents someone who manages the entire system, rather than a specific farm. It has similar attributes (userID and password) and a login() method as the Farmer class.

Web API class: This class likely represents a web application that interacts with the rest of the system. It doesn't have any attributes shown in the diagram, but it does have a method, makeDecisions(), which suggests it might use the data collected by the system to make recommendations for the farm.

Mobile App class: This class represents a mobile application that farmers can use to interact with the system. It doesn't have any attributes or methods shown in the diagram.

Firebase DB class: This class represents a Firebase database, which is a cloud-based NoSQL database that can be used to store data from the system.

Sensors class: This class represents the various sensors that are deployed on the farm to collect data. It doesn't have any attributes shown in the diagram, but it does have methods for:

check(): This method is likely used to collect data from the sensors.

modify(): This method is likely used to change the configuration of the sensors.

Soil Properties class: This class represents the various properties of the soil that are measured by the sensors. It has attributes for:

soil moisture: This attribute can have values of wet, dry, or excess.

fertility: This attribute likely refers to the nutrient content of the soil.
soil pH: This attribute refers to the acidity or alkalinity of the soil, and it can have values of acidic, neutral, or alkaline.

Water Level class: This class represents the water level in the soil. It has attributes for:

low: This attribute likely represents a boolean value indicating whether the water level is low.

relative: This attribute likely represents a numerical value indicating the relative water level.

high: This attribute likely represents a boolean value indicating whether the water level is high.

Temperature class: This class represents the temperature of the soil. It has attributes for:

stress: This attribute likely represents a numerical value indicating the amount of temperature stress on the crops.

extremeStress: This attribute likely represents a boolean value indicating whether the temperature is causing extreme stress on the crops.

moderate: This attribute likely represents a boolean value indicating whether the temperature is moderate.

high: This attribute likely represents a boolean value indicating whether the temperature is high. The arrows in the class diagram show how the classes relate to each other. Here are some of the relationships shown in the diagram:

The Farm class interacts with the Sensors class. This suggests that the sensors are deployed on the farm to collect data.

The Sensors class sends data to the Firebase DB class. This suggests that the sensor data is stored in the Firebase database.

The Web API class interacts with the Firebase DB class. This suggests that the web API retrieves data from the database to make decisions about the farm. The Mobile App class interacts with the Firebase DB class. This suggests that the mobile app can be used to view data from the database.

The System Admin class and Farmer class can both interact with the Web API class. This suggests that both the system admin and the farmer can use the web API to manage the farm.

Overall, this class diagram shows a design for a smart agriculture system that uses IoT sensors to collect data about the farm environment. The data is then stored in a cloud database and accessed by a web API and a mobile app. The system can be used by farmers and system administrators to monitor and manage their farms.

5.2 Sequence Diagram

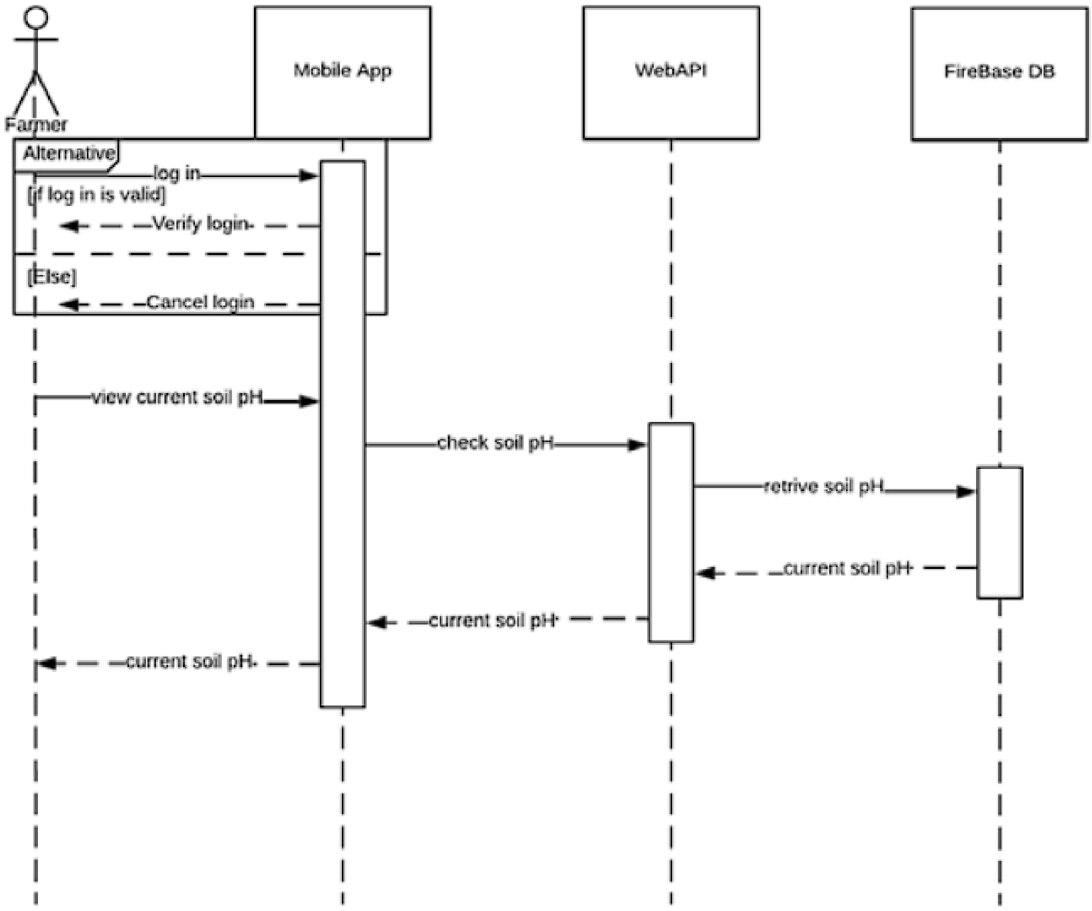

Figure 5.2: Sequence Diagram for Smart Agriculture System for observing data on the current soil pH level

In order to demonstrate the interaction among classes in terms of message exchange over time/events we use "sequence diagrams" also called "event diagrams". Sequence diagrams help us validate and visualize several system events for predicting and analyzing how the system will behave. Here are some of the sequences overtime. Samples of the sequence diagrams are shown in above figs.

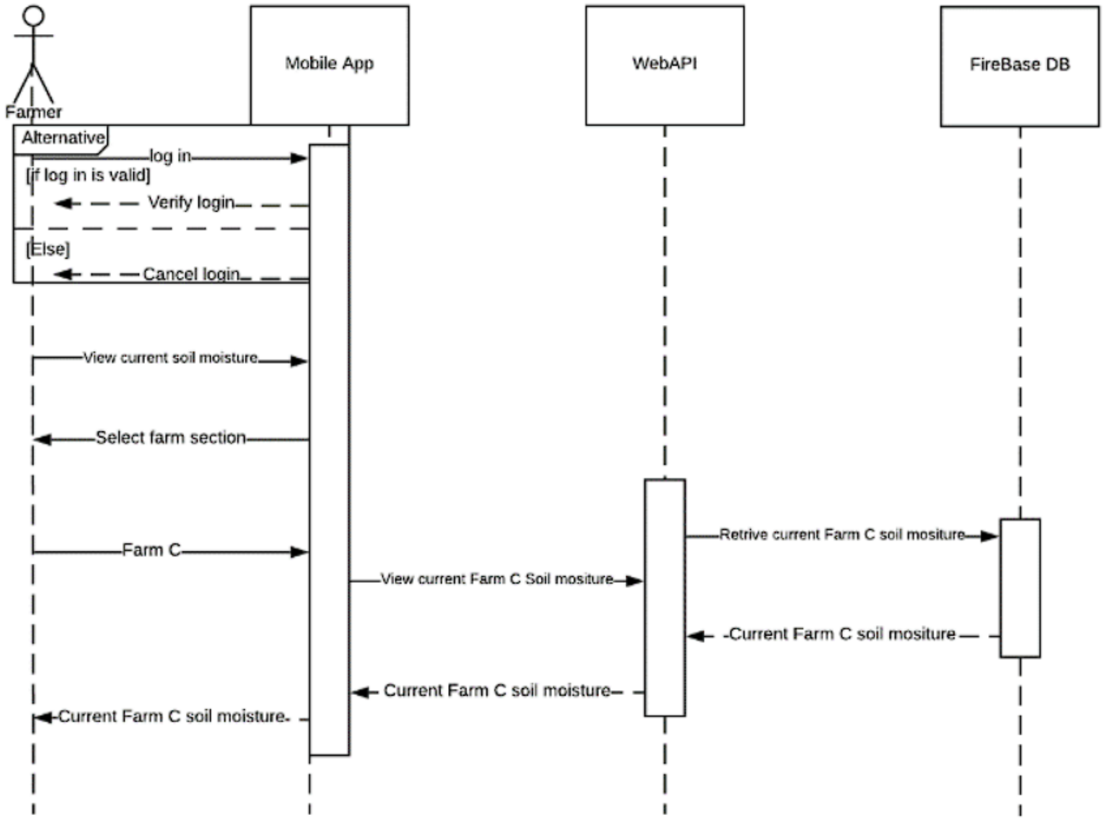

Figure 5.3: Sequence Diagram for Smart Agriculture System for observing data on soil moisture from a farm

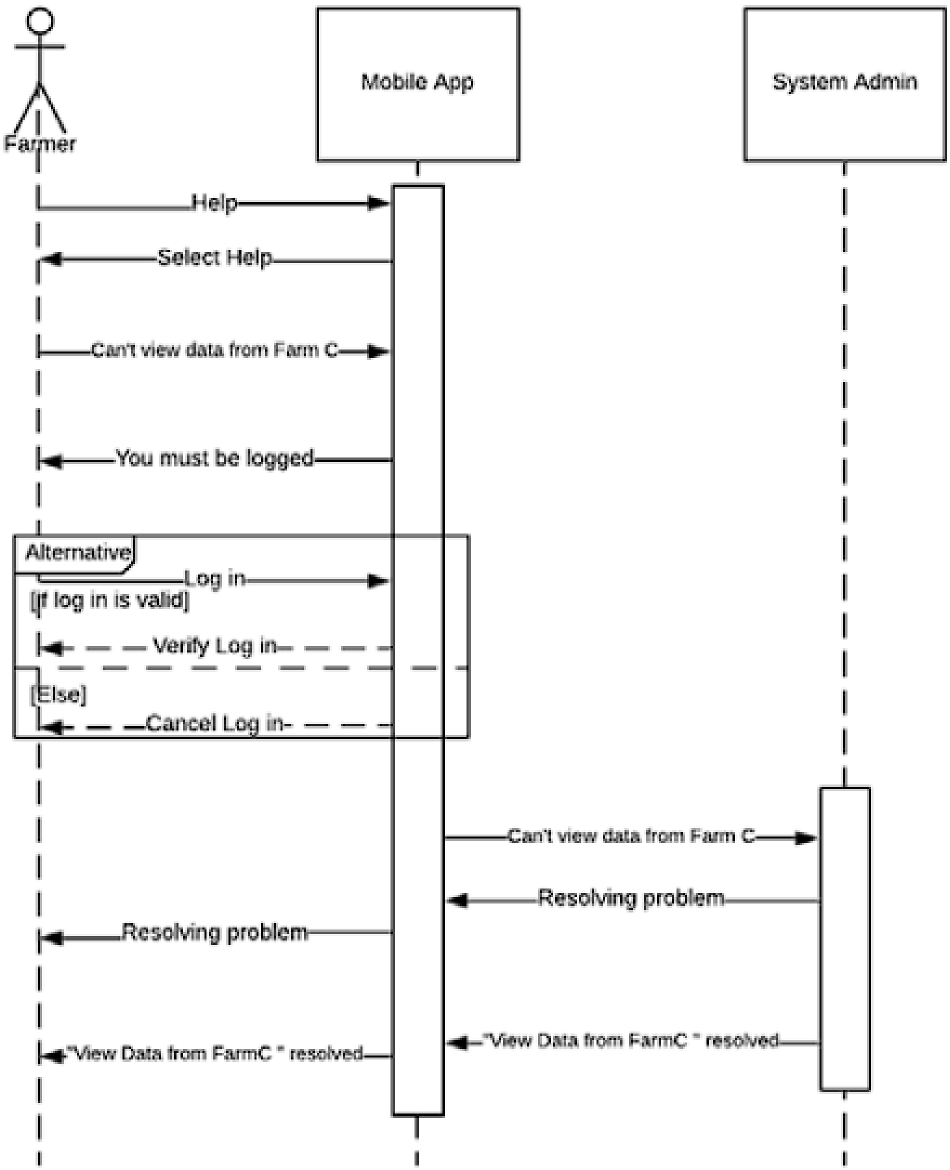

Figure 5.4: Sequence Diagram for Smart Agriculture System for sending a Help Request

5.3 ER Diagram

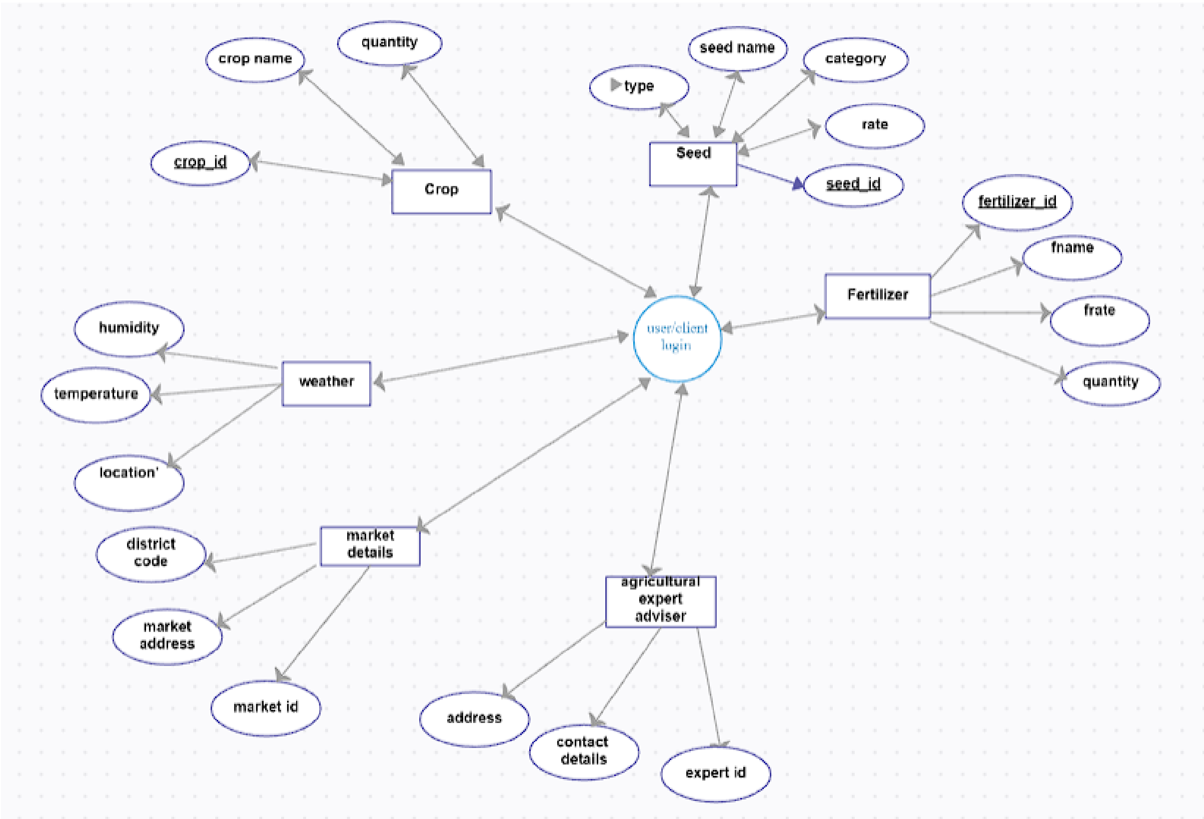

Figure 5.5: ER Diagram for Smart Agriculture System

Figure 5.6 illustrates an Entity-Relationship (ER) diagram showcasing the entities and their relationships in a system designed for Smart Agriculture System using IOT.
Here are the entities and their relationships:

1. **Crop**
Attributes:

- crop id (primary key)
- crop name
- quantity
- seed id (foreign key)

2. **Seed**
Attributes:

- seed id (primary key)
- seed name
- category
- type
- rate

3. **Fertilizer**

Attributes:

- fertilizer id (primary key)
- fname
- frate

4. **Weather**
Attributes:

- temperature
- humidity
- Location

5. **Market Details**
Attributes:

- district code
- market id (primary key)
- Market id

6. **Agricultural Expert/Adviser**
Attributes:

- expert id (primary key)

- contact details

The relationships between the entities are as follows:

- Crop can be grown from one kind of Seed.
- Crop can be fertilized with one kind of Fertilizer.
- The weather data is associated with a particular Location.
- Crops are grown in a particular Location.
- Crops are sold in a particular Market.
- The agricultural management system has one or more User/Client.
- The agricultural management system has one or more Agricultural Expert/Adviser.

5.4 Flowchart

The flowchart shows a decision-making process that automatically controls a watering system based on sensor readings and pre-set thresholds.

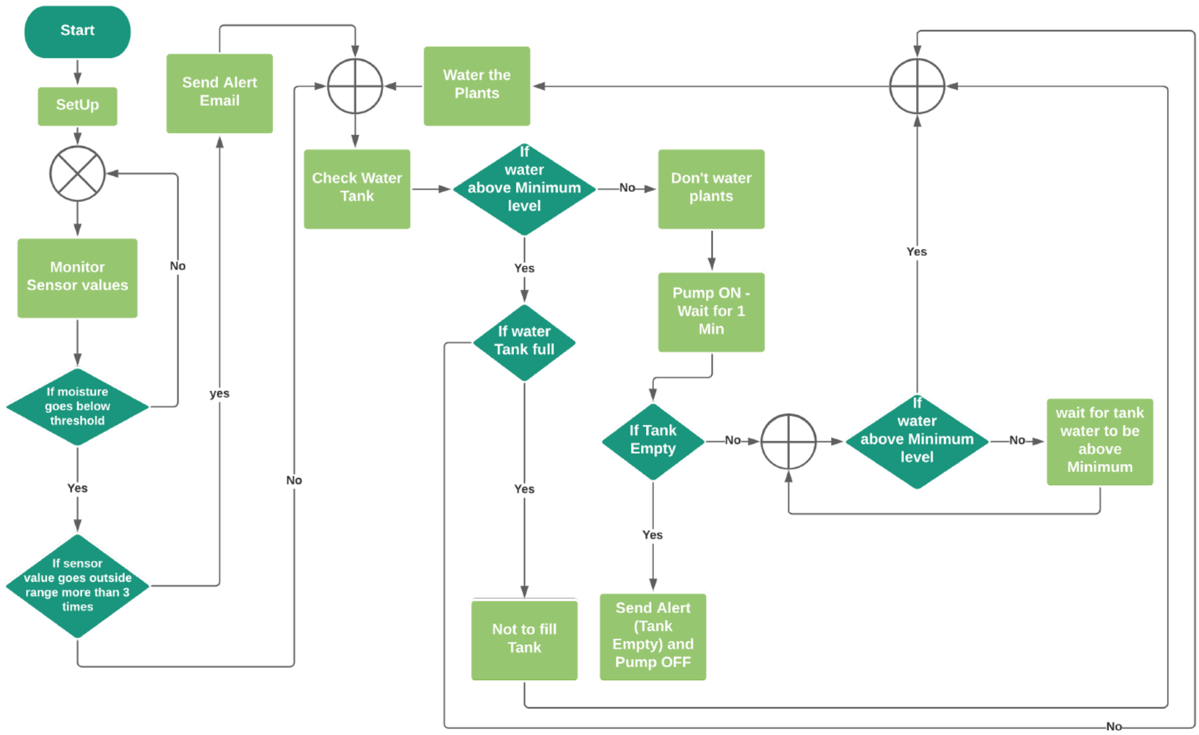

Figure 5.6: Flowchart for Smart Agriculture System

Here's how the process works:

- The system starts by going through a setup phase.
- During operation, the system continuously monitors sensor values.
- If the sensor readings indicate that the moisture level has fallen below a certain threshold, the system checks the water tank.
- If the water tank is full, the pump is turned on for one minute to water the plants.
- After one minute, the system checks the sensor readings again.
- If the moisture level is still below the threshold, the cycle repeats until the level is reached.
- If the water tank is empty, the system sends an alert and the pump is shut off.

The system also includes safeguards to prevent overwatering and to conserve water.

- If the sensor value goes outside the range more than three times, the system sends an alert and the pump is turned off.
- The system won't refill the tank if the sensor reading indicates that the water level is above the minimum threshold.

Chapter 6

Results and Discussions

To test the security framework, we allowed DDoS attacks against the Blockchain Network. A distributed denial-of-service (DDoS) attack targets websites and servers by disrupting network services. A DDoS attack attempts to exhaust an application's resources.

1. Host Machine (HM): Here we have installed an open-source based IoT framework called ThingsBoard, which make it possible to con struct fog layers while seamlessly syncing with the cloud, on the host system and constructed a scenario of a smart farm with two silos, each with a different number of IoT devices. Every second, each gadget reports a pseudo-random value. We utilized a code to simulate a Blockchain client, which accepts the provided values and generates transactions. Mininet network emulation tool is then used to commit the transaction onto the Blockchain via the Open vSwitch module. We also connected the ONOS SDN con troller and sFlow4 collector to the Mininet virtual network.

2. Virtual Machine (VM): We have installed the Ethereum Blockchain network and its Supply Chain AssetTrack application in the Vir tual Machine. To keep records of their updated information, we formed a user for the host system (fog node) and various assets. Every reported value will be used by the blockchain client to update the specified asset in the Ethereum blockchain.

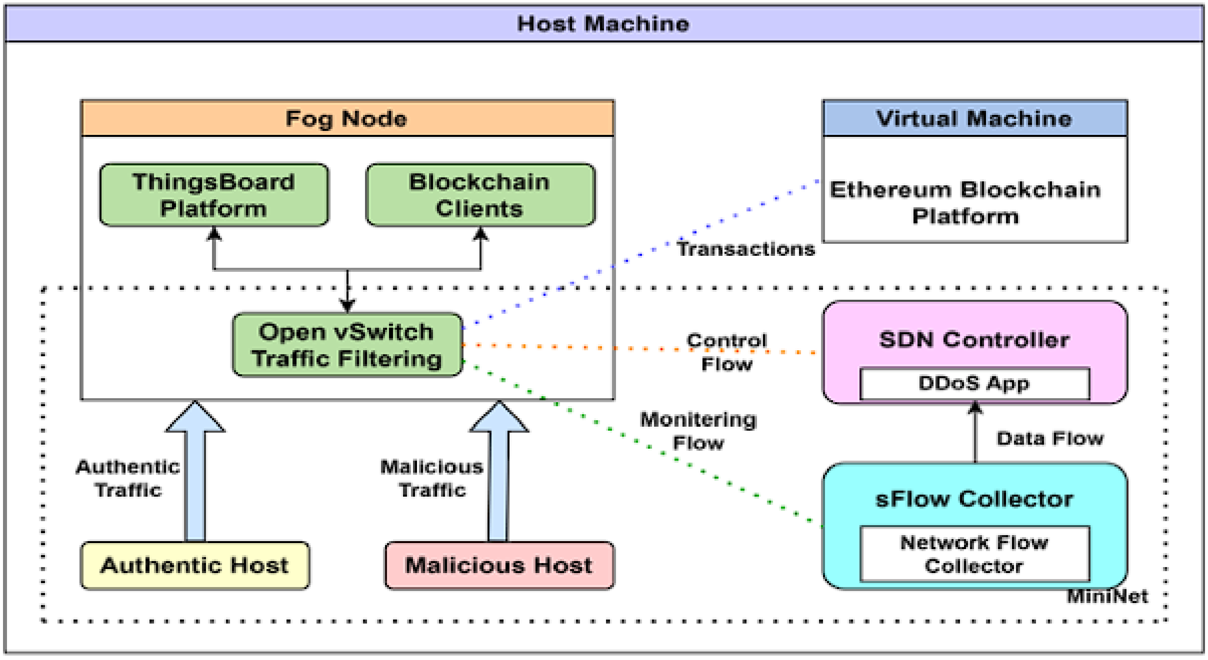

Figure 6.1: Experimental Setup Architecture

To put our framework architecture to the test, we conducted three case studies.

Case 1: We examined the platform's usual workflow, which means we did not conduct any attacks on the Ethereum blockchain.

Case 2: Using the Hping programme on the host PC, we launched a DDoS assault on the Blockchain network. We implemented a DDoS mitigation programme on the SDN controller.

Case 3: We launched a DDoS attack against the Blockchain network, by turning off the DDoS mitigation programme in the SDN controller. During the simulation, we noticed two crucial metrics in each case:

- The number of network packets received by the Blockchain per minute.
- The number of transactions published in the Blockchain network.

Taking the case 1 findings as ideal, with 120 published transactions in 10 min at an average of 100 Kbps; the third scenario revealed a really dismal performance with about 45 published transactions, representing a loss of more than 60 percent compared to the 1st case, and an unreliable

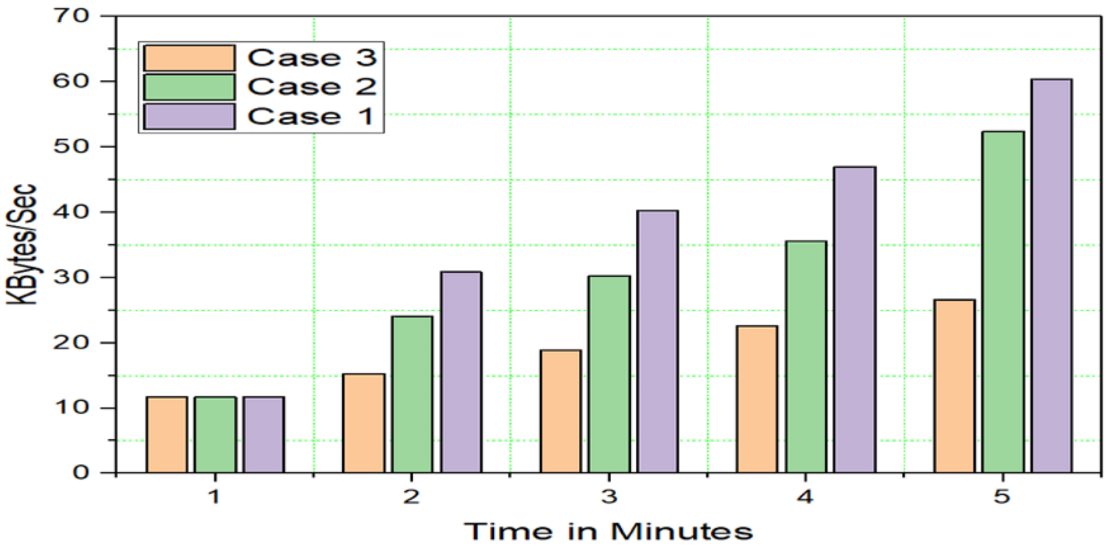

Figure 6.2: Number of packets received by Blockchain per minute

Blockchain network. To understand the outcomes of the second scenario, we must first recognize what took place in the network during the test. The Open vSwitch sends all network traffic data to the Flow forecast system, a real-time reporting system that gives real time notice of network assaults, involving attackers and target information, to the SDN controller security framework via commands. The SDN controller interacts with the switch to discard DDoS traffic while allowing valid traffic to continue through. The attack is stopped fast enough to prevent it from rapidly growing and resulting in 95 blocks published, which is approximately 40 percent more than the 3rd case. The project research discusses the importance of security measures in the context of smart agriculture and proposes a security framework for the agricultural Internet of Things (IoT) that integrates blockchain technology, fog computing, and software-defined networking (SDN). Here's a summary of the key points:

1. Security in Smart Agriculture: The text acknowledges that despite various security issues affecting farming production, there Figure 6.3: Number of transactions published in Blockchain per minute are limited security measures in place. This could be due to the early stages of technology development and the constraints of computing resources. However, as agriculture advances, the need for robust security features becomes critical, especially with the introduction of precision agriculture, which presents new complexities.

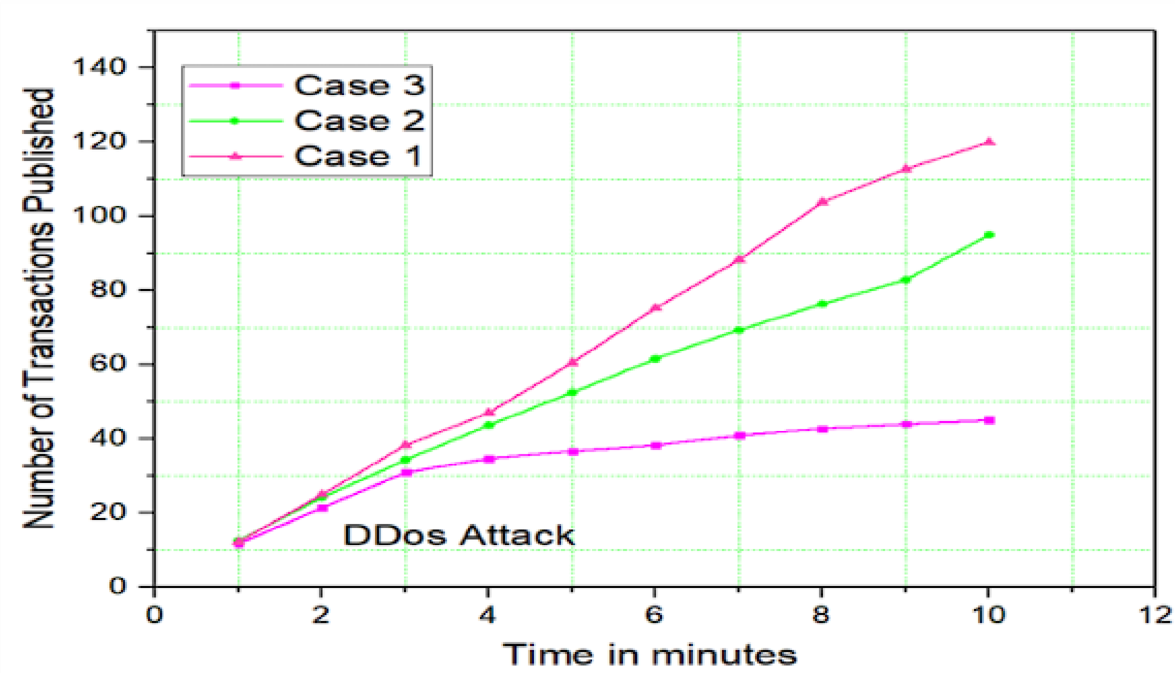

Figure 6.3: Number of transactions published in Blockchain per minute

2. Proposed Security Framework: The study suggests a security frame work for the agricultural IoT. This framework incorporates blockchain technology, fog computing, and SDN. Notably, it aims to enhance communication security for IoT devices, addressing the challenges of limited computing resources and ensuring data reliability.

3. Blockchain Integration: The proposed architecture links each SDN controller to a blockchain, facilitating secure communication be tween SDN controllers and IoT devices. This integration leverages the transparency and security features of blockchain technology.

4. Distributed Trust-Based Verification: The text highlights the use fullness of distributed trust-based verification, particularly for IoT 42 devices with limited resources. This approach enhances security and data reliability in a distributed controller context.

5. Future Directions: The text suggests the potential addition of intrusion detection systems using deep learning algorithms to prevent the insertion of fake sensor data in intelligent agricultural fields. This indicates a focus on further enhancing security in smart agriculture.

The adoption of these technologies fosters a data-driven farming culture where farmers can make informed decisions based on real-time data and predictive analytics. This leads to more efficient farming practices, lower costs, and higher profitability. Additionally, the system promotes the use of renewable energy sources, such as solar-powered sensors and equipment, further reducing the carbon footprint of agricultural activities.

The research also hints at future directions for improving security further, such as the development of more advanced encryption techniques, decentralized data management systems, and enhanced threat detection algorithms. As the landscape of smart agriculture evolves, continuous advancements in security measures will be essential to maintaining the integrity and reliability of the system. This forward-thinking approach ensures that smart agriculture remains resilient against emerging cyber threats, enabling a secure and sustainable agricultural future.

In summary, the research emphasizes the importance of security in smart agriculture and proposes a comprehensive security framework that leverages blockchain, fog computing, and SDN technologies to ad dress the evolving challenges in this field. It also hints at future directions for improving security further.

Chapter 7

Gantt Chart

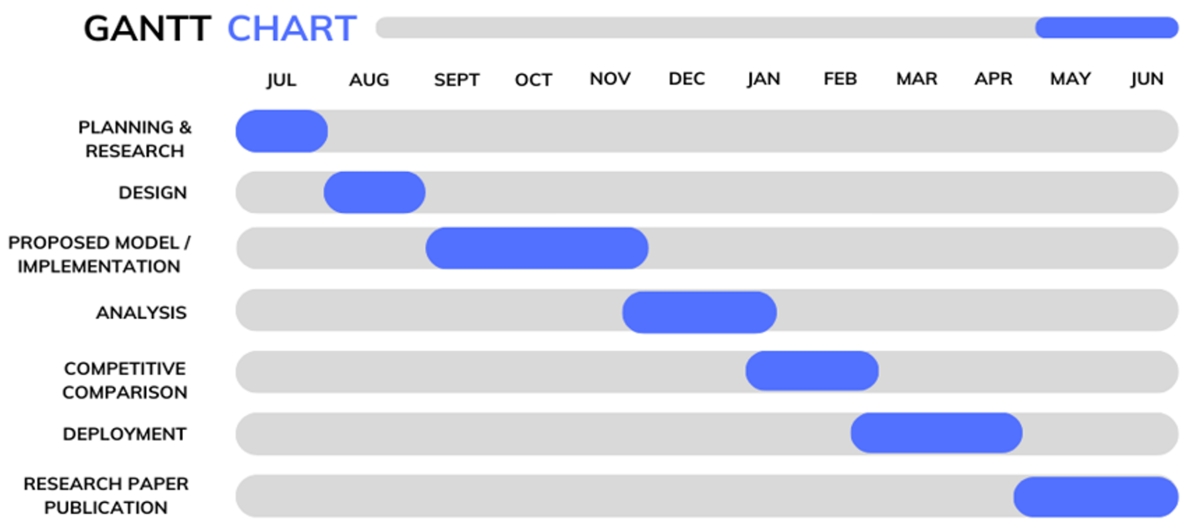

Figure 7.1: Gantt Chart

The Gantt Chart Figure 7.1 represents the project called "Smart Agriculture System". It visually breaks down the project into its different tasks and shows how much time is allocated to each which includes Planning and Research, Design, Proposed Model, Analysis, Deployment and Research Paper Publication. This task suggests that the project will be completed till May 2024.

Chapter 8

Conclusion and Future Scope

8.1 Conclusion

The Smart Agriculture System using IoT designed and implemented in this project demonstrates the potential of integrating modern technology with traditional farming practices. By utilizing sensors to monitor soil moisture, temperature, and humidity, the system provides real-time data that can be used to optimize water management and fertilizer application. The use of NodeMCU and Arduino microcontrollers ensures that the system is both cost-effective and scalable. This project has shown that IoT-based solutions can significantly enhance agricultural productivity and resource management, leading to more sustainable farming practices.

The successful deployment of this system highlights the following key benefits:

Efficient Water Usage: By continuously monitoring soil moisture levels, the system can precisely determine when and how much water is needed, reducing water waste and ensuring crops receive adequate hydration.

Improved Crop Health: Monitoring temperature and humidity helps in maintaining optimal growing conditions, which can improve crop yields and quality.

Resource Optimization: The data-driven approach allows for better management of fertilizers, ensuring that nutrients are applied in the right amounts and at the right times, which can reduce costs and environmental impact.

Scalability: The modular nature of the system, based on widely available microcontrollers and sensors, makes it easy to expand and customize according to different crop requirements and farm sizes.

8.2 Future Scope

The project lays a strong foundation for further development and enhancement. Here are some potential future scopes:

Integration with Advanced Analytics

Implementing machine learning algorithms to analyze historical data can help predict future conditions and optimize resource allocation even further. Utilizing big data analytics to provide deeper insights into crop health and environmental conditions.

Automated Actuation

Incorporating automated irrigation systems that are directly controlled by the sensor data can eliminate the need for manual intervention. Developing a closed-loop system where fertilization is automatically adjusted based on real-time soil nutrient levels.

Expansion of Sensor Types

Adding additional sensors for parameters such as soil pH, nutrient levels, and pest detection can provide a more comprehensive overview of crop health. Implementing remote sensing technologies, such as drones, to cover larger areas and gather more detailed environmental data.

Enhanced User Interface

Developing a mobile app or web portal with a user-friendly interface for farmers to monitor real-time data, receive alerts, and manage their fields remotely. Incorporating visualization tools that provide easy-to-understand insights and recommendations.

Wireless Communication Technologies

Exploring the use of advanced wireless communication protocols such as LoRaWAN or NB-IoT for better connectivity and data transmission over large agricultural fields. Ensuring secure and reliable data transmission to protect the integrity and privacy of the collected data.

Integration with Weather Forecasting Services

Integrating weather forecasting APIs to adjust irrigation and fertilization schedules based on predicted weather conditions. Using forecast data to prevent over-irrigation before rain events or to prepare for extreme weather conditions.

Sustainability and Energy Efficiency

Exploring the use of solar-powered sensors and controllers to make the system more sustainable and reduce dependency on external power sources. Implementing energy-efficient algorithms to extend the battery life of wireless sensors. By incorporating these future developments, the Smart Agriculture System can evolve into a more sophisticated and comprehensive solution, further enhancing agricultural productivity, sustainability, and resource efficiency. This project not only contributes to the advancement of smart farming technologies but also promotes the adoption of sustainable agricultural practices, which are essential for meeting the global food demands of the future.

Bibliography

[1] Vangala, A.; Sutrala, A.K.; Das, A.K.; Jo, M. Smart Contract- Based Blockchain-Envisioned Authentication Scheme for Smart Farming. IEEE Internet Things J. (2021)

[2] Garc´ıa, L.; Parra, L.; Jimenez, J.M.; Lloret, J.; Lorenz, P. IoTbased smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture. Sensors (2020)

[3] Lioutas, E.D.; Charatsari, C. Innovating digitally: The new texture of practices in agriculture 4.0. Sociol. Rural. (2022)

[4] Rao, R.N.; Sridhar, B. IoT Based Smart Crop-Field Monitoring and Automation Irrigation System. In Proceedings of the 2018 2nd Insternational Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January (2018)

[5] Bernhardt, H.; Bozkurt, M.; Brunsch, R.; Colangelo, E.; Herrmann, A.; Horstmann, J.; Kraft, M.; Marquering, J.; Steckel, T.; Tapken, H.; et al. Challenges for Agriculture through Industry 4.0. Agronomy (2021)

[6] Singh, R.K.; Berkvens: Architecture for IoT and Emerging Technologies Based on Precision Agriculture Survey. IEEE Access (2021)

[7] Research Journal of Engineering and Technology, [2022], https://www.irjet.net/archives/V9/i2/IRJETV9I228.pdf.

[8] Tanmay Baranwal and Nitika Pushpendra Kumar Pateriya, "Development of IoT based Smart Security and Monitoring Devices for Agriculture" in 6th International Conference - Cloud System and Big Data Engineering, IEEE, pp. 978–1-4673-8203-8/16, 2016.

[9] P. Rajalakshmi and S. Devi Mahalakshmi, "IOT Based Crop-Field Monitoring And Irrigation Automation" in 10th International conference on Intelligent systems and control (ISCO) 7–8 Jan 2016, published in IEEE Xplore, Nov 2016.

[10] Mohamed Rawidean Mohd Kassim, Ibrahim Mat and Ahmad Nizar Harun, "Wireless Sensor Network in Precision agriculture application" in International conference on computer Information and telecommunication systems (CITS), published in IEEE Xplore, July 2014.

[11] Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra NietoGaribay, and Miguel Ángel PortaGándara, "Computerized Irrigation System Using a Wireless Sensor Network and GPRS Module", IEEE Transactions on Instrumentation and Measurements, 0018-9456,2013

[12] Paparao Nalajala, P Sambasiva Rao, Y Sangeetha, Ootla Balaji, K Navya," Design of a Smart Mobile Case Framework Based on the Internet of Things", Advances in Intelligent Systems and Computing,Volume 815, Pp. 657-666, 2019.

[13] k.lakshmisudha, swathi hegde, neha cole, shruti iyer, " good particularity most stationed cultivation spinning sensors", state-of-theart weekly going from microcomputer applications (0975-8887), number 146-no.11, july 2011

[14] G. Manogaran, M. Alazab, K. Muhammad, V.H.C. De AlbuquerqueSmart sensing based functional control for reducing uncertainties in agricultural farm data analysis. IEEE Sensor J, 21 (16) (2021), pp. 17469-17478

[15] A.D. Boursianis, M.S. Papadopoulou, A. Gotsis, S. Wan, P. Sarigiannidis, S. Nikolaidis, et al. Smart irrigation system for precision agriculture—the AREThOU5A IoT platform IEEE Sensor J, 21 (16) (2020), pp. 17539-17547

[16] M.S. Farooq, S. Riaz, A. Abid, T. Umer, Y.B. Zikria. Role of IoT technology in agriculture: a systematic literature review Electronics, 9 (2) (2020), p. 319

[17] Hsiu-Wen Wang, Chi-Hua Chen, Ding-Yuan Cheng, Chun-Hao Lin. "A Real-Time Pothole Detection Approach for Intelligent Transportation System",Hindawi Publishing CorporationMathematical Problems in Engineering, [2015]

[18] M. Alkahtani, Q.S. Khalid, M. Jalees, M. Omair, G. Hussain, C.I. Pruncu E-agricultural supply chain management coupled with blockchain effect and cooperative strategies Sustainability, 13 (2) (2021), p. 816

[19] A. Kamilaris, A. Kartakoullis, F.X. Prenafeta-Boldú .A review on the practice of big data analysis in agriculture .Comput Electron Agric, 143 (2017), pp. 23-37

[20] N. Tantalaki, S. Souravlas, M. Roumeliotis - Data-driven decision making in precision agriculture: the rise of big data in agricultural systems J Agric Food Inf, 20 (4) (2019), pp. 344-380

[21] Y. Hajjaji, W. Boulila, I.R. Farah, I. Romdhani, A. Hussain - Big data and IoT-based applications in smart environments: a systematic review Computer Science Review, 39 (2021), Article 100318

[22] A. Kochhar, N. Kumar Wireless sensor networks for greenhouses: an end-to-end review. Comput Electron Agric, 163 (2019), Article 104877

[23] D. Thakur, Y. Kumar, A. Kumar, P.K. Singh. Applicability of wireless sensor networks in precision agriculture: a review Wireless Pers Commun, 107 (2019), pp. 471-512

[24] Z. Zhai, J.F. Martínez, V. Beltran, N.L. Martínez Decision support systems for agriculture 4.0: survey and challenges Comput Electron Agric, 170 (2020), Article 105256

Appendix A

Research Paper

Gmail

Aradhya Priyadarshi <aradhyapriyadarshi@gmail.com>

Acceptance : ICOTET 2024

ICOTET2024 <icotetdgi2024@gmail.com> Mon, May 27, 2024 at 4:09 PM
To: Aradhya Priyadarshi <aradhyapriyadarshi@gmail.com>
Cc: icotet@gnindia.dronacharya.info

Greetings from ICOTET 2024!

Dear Author (s)

We are pleased to inform you that **Paper ID 2750** entitled **" Smart Agriculture System Using IoT "** submitted by you has been accepted by the 2nd International Conference on Optimization Techniques in Engineering and Technology Engineering (ICOTET 2024).

You are advised to register for the conference by **27th of May,** 2024! Payment details for registration can be found at the bottom of this email.

You are requested to fill out the following Google form for the registration and payment information etc.:

https://forms.gle/mqyRFhx45hqkJ6cx7

All the registered and presented papers for the 2nd ICOTET 2024 will be published in the AIP Conference Proceedings (Scopus Index) and Springer Nature Conference Proceedings (Scopus Index). Please note that the plagiarism level of the paper should not exceed 15%.

For further details, please visit the official website: https://www.icotet.in/registration

Thanks & Regards

Organizing Committee

ICOTET 2024.

Figure A.1: Acceptance Acknowledgemnt

The following image, Figure 6.1, shows an Acceptance Mail application submitted to Conference ICOTET 2024 by Aradhya Priyadarshi for a paper titled "Smart Agriculture System Using IoT." The paper, identified by ID 2750, focuses on leveraging Internet of Things (IoT) technology to enhance agricultural practices.

Smart Agriculture System Using IoT

1st Aradhya Priyadarshi
AKGEC Ghaziabad, U.P, India
aradhyapriyadarshi@gmail.com

2nd Harsh Tiwari
AKGEC Ghaziabad, U.P, India
harshtiwari09092000@gmail.com

3rd Khushi Singh
AKGEC Ghaziabad, U.P, India
sngkhushi2003@gmail.com

4th Sanjay Kumar Sonkar
AKGEC Ghaziabad, India

***Abstract*—Intelligent agricultural systems have emerged as an effective way to address challenges facing farmers, including climate change, resource scarcity and changing food demand. This overview provides an overview of the concepts, functions, and benefits of smart farming systems. Smart farming systems use a variety of technologies, such as the Internet of Things (IoT), Blockchain, analytics data and remote sensing. come in Increase agriculture and improve yields. These systems allow critical factors such as soil moisture, temperature, crop health and pests to be monitored and controlled in real time. Smart Agriculture System (SAS) automates operations, reduces resource waste, and improves decision-making processes using sensors and actuators. The simple use of a agricultural system has many benefits. These include increased productivity, reduced resource consumption and improved quality of life. SAS contributes to the economic viability of agriculture by optimizing resource allocation and reducing input costs. In addition, these systems promote sustainable agriculture, reduce environmental impacts and promote philanthropy. SAS uses the power of technology to help farmers overcome challenges and achieve better, more productive and more sustainable food production. Continued research, development and adoption of these systems is essential to ensure food security and meet the needs of a growing world population.**

I. Introduction

The 21st century has witnessed a transformative revolution across various domains, largely propelled by the emergence of the Internet of Things (IoT) and smart computing technologies. These advancements have found diverse applications, ranging from real-time monitoring of crop conditions and soil moisture levels to the utilization of drones for tasks like pesticide spraying. This widespread amalgamation of modern IT and traditional agricultural practices marks the advent of Agriculture 4.0, characterized by enhanced intelligence and automation in farming processes.

The article commences by offering an overview of the evolution of Agriculture 4.0, delineating its advantages and disadvantages. This research delves into the intricacies of layered architectural design, identifies prevalent security concerns, and delineates security requirements and forthcoming opportunities within the realm of smart agriculture. Moreover, it proposes a robust security architectural framework tailored specifically for Agriculture 4.0, integrating blockchain technology, fog computing, and software-defined networking.

Sr. no.	Name of the paper	Author	Year of publication	Technology used	Limitations
1	Automation in Agriculture and IoT	Vaishali Puranik, Ankit Ranjan, Anamika Kumari	2019	Arduino Circuit, 2X16 Liquid Crystal Display, GSM Module	Cost is high
2	An IoT Instrumented Smart Agricultural Monitoring and Irrigation System	Anil Kumar Saini, Susmita Banerjee, Himanshu Nigam	2020	ThingSpeak, NodeMCU, Sensor, E-mail	Increased channel maintenance
3	Smart Agriculture Based on IoT and Cloud Computing	Sriveni Namani, Bilal Gonen	2020	Sky Drone FPV2, Cloud Computing	It requires strong network
4	Providing Smart Agricultural Solutions to	M.K.Gayatri , J.Jayasakthi ,	2015	Sensor module, Processor module, Communication	There could be wrong analysis of weather

Fig. 1. Literature Review

The proposed framework amalgamates Ethereum blockchain and software-defined networking solutions within an open-source IoT platform. Subsequently, it undergoes rigorous testing across three distinct scenarios subjected to Distributed Denial of Service (DDoS) attacks. The findings of the performance evaluation indicate that the suggested security framework exhibits commendable efficacy in mitigating potential threats and safeguarding agricultural operations in the digital age.

II. Literature Survey

A literature review is a critical and comprehensive analysis of existing literature on a specific topic or research question.

III. Methodology

A. Block Diagram

The block diagram consists of three crucial sensors: temperature, soil moisture, and humidity sensor. In order to record, show, or indicate temperature changes, a temperature sensor is an electrical device that senses the temperature of its immediate surroundings and converts the information into electronic data.The Soil Moisture Sensor uses capacitance to

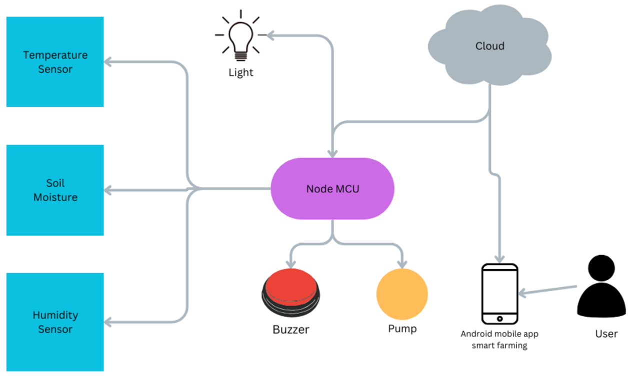

Fig. 2. Block Diagram

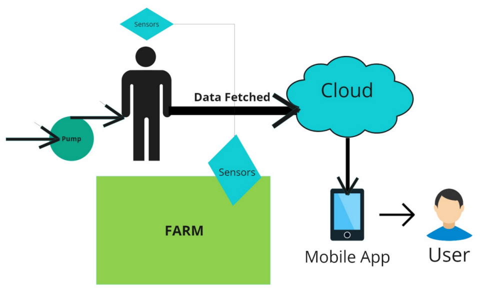

Fig. 3. System Archirecture

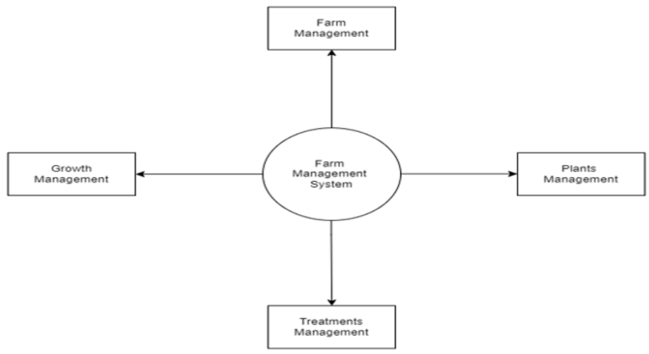

Fig. 4. DFD Level 0 Diagram

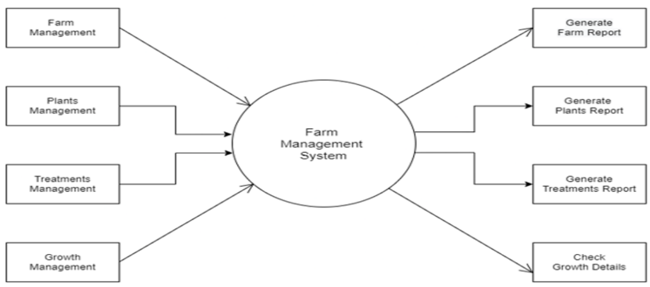

Fig. 5. DFD Level 1 Diagram

measure the dielectric permittivity of the surrounding medium. The dielectric permittivity of soil is influenced by its water content. The sensor generates a voltage in relation to the dielectric permittivity, and along these lines, the water content of the dirt. A humidity sensor monitors, cycles, and reports humidity and air temperature. Relative humidity is the ratio of visible humidity to the highest measured humidity at a given air temperature. When seeking relief, it's important to consider relative humidity.

B. *System Architecture*

Smart agriculture uses IoT and sensor technology to optimize farming operations, increase crop yield, and improve resource efficiency. The system architecture for such an application consists of multiple components that collaborate smoothly to gather, process, and analyze data from agricultural contexts. Sensors, IoT devices, cloud platforms, and web apps are all components of system architecture.

C. *DFD Level 0 Diagram*

The level-0 (DFD) diagram depicts the primary serviceable sections of the system under evaluation. Similar to the context diagram, all system under examination should be represented by a single level-0 (DFD) diagram. The figures below help to visualize the DFD diagram.Three distinct approaches provide a realistic way to begin the analysis. The technique involves sensing live temperature and humidity data, configuring procedures using an application, and controlling buzzers, motors, pumps, and ropes using sensor data.

D. *DFD Level 1 Diagram*

The Level-1 (DFD) graphic depicts additional key functional aspects of the system under consideration. As with level 1 (DFD).There is no method for determining what is and isn't. The Level-1 (DFD) process diagram should only include the core serviceable sections of the system, avoiding include lower-level operations. This section displays the system's overall processing. The level-1 DFD provides a clear description of what is done and how. Only the main subprocesses are explained here.

IV. Result

The project is dependable, efficient, and saves time by providing easy control. Users can store agricultural land information in the cloud and use the Android app to monitor temperature, soil moisture, and crop watering needs. Users can view information and notifications at any time. This project offers enhanced security and streamlines traditional operations, saving time and resources. Based on our analysis, we believe that the suggested system will automate farming, digitize the process, and improve mapping and land upkeep for farmers and owners.

V. Conclusion

The project research discusses the importance of security measures in the context of smart agriculture and proposes a security framework for the agricultural Internet of Things (IoT) that integrates blockchain technology, fog computing, and software-defined networking (SDN).

Here's a summary of the key points:

1. Security in Smart Agriculture: The text acknowledges that despite various security issues affecting farming production, there are limited security measures in place. This could be due to the early stages of technology development and the constraints of computing resources. However, as agriculture advances, the need for robust security features becomes critical, especially with the introduction of precision agriculture, which presents new complexities.

2. Proposed Security Framework: The study suggests a security framework for the agricultural IoT. This framework incorporates blockchain technology, fog computing, and SDN. Notably, it aims to enhance communication security for IoT devices, addressing the challenges of limited computing resources and ensuring data reliability.

3. Blockchain Integration: The proposed architecture links each SDN controller to a blockchain, facilitating secure communication between SDN controllers and IoT devices. This integration leverages the transparency and security features of blockchain technology.

4. Distributed Trust-Based Verification: The text highlights the usefulness of distributed trust-based verification, particularly for IoT devices with limited resources. This approach enhances security and data reliability in a distributed controller context.

5. Future Directions: The text suggests the potential addition of intrusion detection systems using deep learning algorithms to prevent the insertion of fake sensor data in intelligent agricultural fields. This indicates a focus on further enhancing security in smart agriculture.

In summary, the research emphasizes the importance of security in smart agriculture and proposes a comprehensive security framework that leverages blockchain, fog computing, and SDN technologies to address the evolving challenges in this field. It also hints at future directions for improving security further.

VI. Reference

1. Vangala, A.; Sutrala, A.K.; Das, A.K.; Jo, M. Smart Contract Based Blockchain-Envisioned Authentication Scheme for Smart Farming. IEEE Internet Things J. (2021)

2. Garc´ıa, L.; Parra, L.; Jimenez, J.M.; Lloret, J.; Lorenz, P. IoTbased smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture. Sensors (2020)

3. Gupta, M.; Abdelsalam, M.; Khorsandroo, S.; Mittal, S. Security and Privacy in Smart Farming: Challenges and Opportunities. IEEE Access (2020)

4. Lioutas, E.D.; Charatsari, C. Innovating digitally: The new texture of practices in agriculture 4.0. Sociol. Rural. (2022)

5. Rao, R.N.; Sridhar, B. IoT Based Smart Crop-Field Monitoring and Automation Irrigation System. In Proceedings of the 2018 2nd Insternational Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January (2018)

6. Bernhardt, H. Bozkurt, M. Brunsch, R. Colangelo, E. Herrmann, A. Horstmann, J. Kraft, M. Marquering, J. Steckel, T. Tapken, H. et al. Challenges for Agriculture through Industry 4.0. Agronomy

7. Singh, R.K.; Berkvens: Architecture for IoT and Emerging Technologies Based on Precision Agriculture Survey. IEEE Access (2021)

Appendix B

UNAI Certificate

Certificate of Compliance with United Nations Sustainable Development Goals

This is to certify that the project titled SMART AGRICULTURE SYSTEM USING IOT, submitted by Aradhya Priyadarshi, Harsh Tiwari and Khushi Singh, final year students of the Bachelor of Technology in Information Technology program at Ajay Kumar Garg Engineering College, Ghaziabad, have been reviewed and found to be in alignment with the following United Nations Sustainable Development Goals (SDGs). All efforts have been made to the best of our ability and knowledge that no other SDGs are compromised or negatively impacted.

SDG No.	SDG Name	Relevance	SDG No.	SDG Name	Relevance
1	No Poverty	☐	10	Reduced Inequalities	☐
2	Zero Hunger	☐	11	Sustainable Cities and Communities	☐
3	Good Health and Well-being	☐	12	Responsible Consumption and Production	☐
4	Quality Education	☐	13	Climate Action	☐
5	Gender Equality	☐	14	Life Below Water	☐
6	Clean Water and Sanitation	☐	15	Life on Land	☐
7	Affordable and Clean Energy	☐	16	Peace, Justice, and Strong Institutions	☐
8	Decent Work and Economic Growth	☐	17	Partnerships for the Goals	☐
9	Industry, Innovation, and Infrastructure	☐			

Signature of the Students

Aradhya Priyadarshi (2000270110021)

Harsh Tiwari (2000270130071)

Khushi Singh (2000270130087)

Signature of the Supervisor

Mr. Sanjay Sonker

Appendix C

Source Code

Arduino Code(Backend)

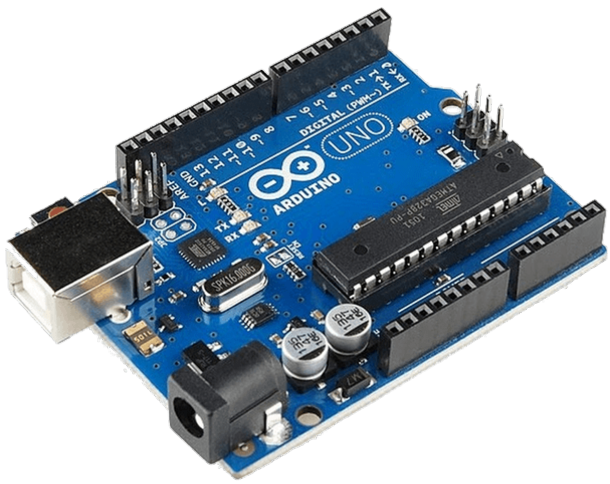

Figure C.1: Arduino

Arduino code, typically written in C++, consists of two main functions: ‘setup()‘ and ‘loop()‘. The ‘setup()‘ function runs once at the start, initializing sensors and outputs, while the ‘loop()‘ function runs continuously, executing the main tasks. For a smart agriculture system, Arduino can be programmed to read soil moisture levels, control irrigation based on sensor data, and manage environmental conditions like temperature and humidity. By interfacing with various sensors and actuators, Arduino can automate watering, monitor crop health, and optimize resource use, enhancing efficiency and productivity in agriculture.

source_code.ino

```
#include <DHT.h>
#include <DHT_U.h>
#include <Bridge.h>
#include <BridgeServer.h>
#include <BridgeClient.h>
#include <Process.h>
#include <FileIO.h>

#define DHTPIN    2
#define DHTTYPE   DHT22
#define GROUND_PUMP 5
#define WATERPUMP_PIN  10
#define MOIST_IN   A0
#define LEVEL_IN   A2

//For timers
#define NORMAL_OPERATION_TIME   10
#define ONE_MINUTE_PAST_PUMP_ON  6000 // 100 iteration (10 ms each) * 60 secs in a min
#define ONE_SECOND_MS   1000

BridgeServer server;

//System States
typedef enum
{
  A_CHECK_WATER_TANK = 0,
  A_WAIT_FOR_TANK_PUMP_ON,
  A_WAIT_FOR_TANK_PUMP_OFF,
  A_TAKE_FIELD_ACTION
```

```
}working_cycle_t;

typedef enum
{
  TO_WATER = 0,
  NOT_TO_WATER
}field_action_t;

typedef enum
{
  TO_FILL_WATER=0,
  NOT_TO_FILL_WATER
}tankaction_t;

typedef enum
{
  STILL_EMPTY_TANK = 0x03,
  TANK_OK_NOW = 0x04
}mail_type_t;



//Sensor readings
float temp;
float humi;
float moist;
int waterlevel;
const int AirValue = 620;
const int WaterValue = 310;
sensor_t sensor;
uint32_t min_delay_DHT_ms;
DHT_Unified dht(DHTPIN, DHTTYPE);
```

```
// time strings
String dataString;
String dataStringHMax;
String dataStringHMin;
String dataStringTMax;
String dataStringTMin;
String dataStringMMax;
String dataStringMMin;
int space = 2;
int LogSpace = 10;
long int time1=0;
long int time2=0;
unsigned long lastMillis = 0;
unsigned long MillisEmptytank=0;
static boolean reset_dht_waiting_cycles = false;

//Threshold Values
float hThMax;
float hThMin;
float tThMax;
float tThMin;
float mThMax;
float mThMin;

float humiMax=0;
float humiMin=100;
float tempMax=0;
float tempMin=100;
float MoistMax=0;
```

```
float MoistMin=100;

int crossed_temp_thresh=0;
int crossed_humi_thresh=0;
int crossed_moist_thresh=0;

int email_OnOff;
int maxCalls = 4;
int calls = 0;

void setup() {

  Serial.begin(9600);

  //Bridge connection between microcontroller and microprocessor
  Bridge.begin();
  dht.begin();
  dht.temperature().getSensor(&sensor);

  // Listen for incoming connection
  server.listenOnLocalhost();
  server.begin();

  //Linux FileSystem Setup
  FileSystem.begin();
  FileSystem.remove("/mnt/sda1/www/Sensor_data.txt");
  FileSystem.remove("/mnt/sda1/www/log.csv");

  //Check configureemail.txt file to enable or disable email alerts
  File confFileEmail = FileSystem.open("/mnt/sda1/www/configureemail.txt", FILE_READ);
  email_OnOff = confFileEmail.parseInt();
```

```
  confFileEmail.close();

  // set the threshold values to be stored in limit.txt file
  File confFileThresh = FileSystem.open("/mnt/sda1/www/limit.txt", FILE_READ);
  tThMin = confFileThresh.parseFloat();
  tThMax = confFileThresh.parseFloat();
  hThMin = confFileThresh.parseFloat();
  hThMax = confFileThresh.parseFloat();
  mThMin = confFileThresh.parseFloat();
  mThMax = confFileThresh.parseFloat();
  confFileThresh.close();
  dataString += getTimeStamp();
  dataStringHMax = dataString;
  dataStringHMin = dataString;
  dataStringTMin = dataString;
  dataStringMMax = dataString;
  dataStringMMin = dataString;
  min_delay_DHT_ms = sensor.min_delay / 1000;
  pinMode(GROUND_PUMP, OUTPUT);
  pinMode(WATERPUMP_PIN, OUTPUT);
}

void loop() {

  static working_cycle_t activity = A_CHECK_WATER_TANK;
  static uint8_t current_delay_ms = NORMAL_OPERATION_TIME;

  MonitorMode();

  BridgeClient client = server.accept();  // Get clients coming from server
  if (client)
```

```
{
  String command = client.readStringUntil('/'); // Process request
  if (command == "conf")
  {
    confCommand(client);
  }
  client.stop();    // Close connection and free resources.
}

switch(activity)
{

  case A_CHECK_WATER_TANK:

  if(false == MinWaterLevelOk())
  {
    WaterPlants(NOT_TO_WATER);
    activity = A_WAIT_FOR_TANK_PUMP_ON;
  }
  else if(true == MaxWaterLevelReached())
  {
    FillTank(NOT_TO_FILL_WATER);
    activity = A_TAKE_FIELD_ACTION;
  }
  else
  {
    activity = A_TAKE_FIELD_ACTION;
  }
  break;

  case A_WAIT_FOR_TANK_PUMP_ON:
```

```
   static uint16_t alarm_counter = 0;

   if(true == MinWaterLevelOk())
   {
     alarm_counter = 0;
     reset_dht_waiting_cycles = true;  //to make sure to have the first value reading from the DHT
sensor immediately
     activity = A_TAKE_FIELD_ACTION;
   }
   else
   {
     FillTank(TO_FILL_WATER);
     if((millis() - MillisEmptytank) >= 60000){
       MillisEmptytank= millis();
       alarm_counter = 0;
       runPythonScript(STILL_EMPTY_TANK);
       FillTank(NOT_TO_FILL_WATER);
       activity = A_WAIT_FOR_TANK_PUMP_OFF;
     }
   }
   break;

   case A_WAIT_FOR_TANK_PUMP_OFF:  // case for the manual refill of the tank

   if(true == MinWaterLevelOk())
   {
     runPythonScript(TANK_OK_NOW);
     Serial.print("sending email..Tank Ok");
     reset_dht_waiting_cycles = true;  //to make sure to have the first value reading from the DHT
sensor immediately
     activity = A_TAKE_FIELD_ACTION;
```

```
    }
    else
    {
      // nothing
    }
    break;

    case A_TAKE_FIELD_ACTION:

    field_action_t field_action;

    ElaborateData(moist);
    activity = A_CHECK_WATER_TANK;
    break;
  }
  delay(current_delay_ms);
}

void MonitorMode()
{
  String dataString;

  delay(50); // Poll every 50ms
  time2 = millis();
  temp = AcquireTemperature();
  humi = AcquireHumidity();
  moist = AcquireGroundMoisture();
  waterlevel = AcquireWaterLevel();

  if((millis() - lastMillis) >= (LogSpace * 1000)) //for the presentation it has been set to 5 seconds
  {
```

```
  AcquireLog();
  lastMillis =millis();
 }
 if((time2 - time1) >= (space * 1000))
 {
  dataString += getTimeStamp();

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (it is a very slow sensor)
  time1 = millis();

  if(humi > humiMax)
  {
   humiMax = humi;
   dataStringHMax = dataString;
  }
  if(humi < humiMin)
  {
   humiMin = humi;
   dataStringHMin = dataString;
  }
  if(temp > tempMax)
  {
   tempMax = temp;
   dataStringTMax = dataString;
  }
  if(temp < tempMin)
  {
   tempMin = temp;
   dataStringTMin = dataString;
  }
```

```
if(moist > MoistMax)
{
  MoistMax = moist;
  dataStringMMax = dataString;
}
if(moist < MoistMin)
{
  MoistMin = moist;
  dataStringMMin = dataString;
}

// send the e-mail (if needed)
if(email_OnOff == 1)
{
  if (temp <= tThMin || temp>=tThMax)
  {
    crossed_temp_thresh ++;
  }
  if (humi <= hThMin || humi>=hThMax)
  {
    crossed_humi_thresh ++;
  }
  if (moist <= mThMin || moist >= mThMax)
  {
    crossed_moist_thresh++;
  }
  if (calls < maxCalls)
  {
    if(crossed_temp_thresh == 3)
    {
      Serial.println(F("Triggered"));
```

```
        runPythonScript(0x00);

        calls++;
        crossed_temp_thresh = 0;
      }
      if(crossed_humi_thresh == 3)
      {
        Serial.println(F("Triggered"));
        runPythonScript(0x01);
        calls ++;
        crossed_humi_thresh = 0;
      }
      if(crossed_moist_thresh == 3)
      {
        Serial.println(F("Triggered"));
        runPythonScript(0x02);
        calls ++;
        crossed_moist_thresh = 0;
      }
    }
    else
    {
      Serial.println("\nTriggered! Skipping to save smtp calls.");
    }
  }
 }
}

float AcquireTemperature(void)
{
 sensors_event_t event;
```

```
  dht.temperature().getEvent(&event);
  if (isnan(event.temperature))
  {
    Serial.println(F("Error reading temperature!"));
  }
  else
  {
    temp = event.temperature;
  }
  return temp;
}

float AcquireHumidity(void)
{
  sensors_event_t event;

  dht.humidity().getEvent(&event);
  if (isnan(event.relative_humidity))
  {
    Serial.println(F("Error reading humidity!"));
  }
  else
  {
    humi = event.relative_humidity;
  }
  return humi;
}

int AcquireWaterLevel(void)
{
```

```
  int level;

  level = analogRead(LEVEL_IN);
  int templevel = map(level, 330, 390, 0, 10);
  if (templevel >= 10)
  {
    level = 10;
  }
  else if(templevel <= 0)
  {
    level = 0;
  }
  else if(templevel > 0 && templevel < 10)
  {
    level = templevel;
  }
  level = level * 10;
  return level;

}

float AcquireGroundMoisture(void)
{
  float moistpercent;

  moistpercent = analogRead(MOIST_IN);
  moistpercent = map(moistpercent, AirValue, WaterValue, 0, 100);
  if(moistpercent >= 100)
  {
    moist = 100;
  }
```

```
  else if(moistpercent <= 0)
  {
    moist = 0;
  }
  else if(moistpercent > 0 && moistpercent < 100)
  {
    moist = moistpercent;
  }
  return moist;
}

void AcquireLog()
{
  String time = getTimeStamp();
  Serial.println(time);
  String tString = String(temp);
  String hString = String(humi);
  String mString = String(moist);
  File log = FileSystem.open("/mnt/sda1/www/log.csv", FILE_APPEND);
  if(log)
  {
    log.print('\n');
    log.print(time);
    log.print(',');
    log.print(temp);
    log.print(',');
    log.print(humi);
    log.print(',');
    int moisti = moist;
    log.print(moisti);
  }
```

```
  if((time=="22:59:55")||(time=="22:59:54"))
  {
    runPythonScript(0x06);
  }

  File dataFile = FileSystem.open("/mnt/sda1/www/Sensor_data.txt", FILE_APPEND);
  if (dataFile)
  {
    dataFile.print(time);
    dataFile.print("  Temperature= ");
    dataFile.print(tString);
    dataFile.print("*C ");
    dataFile.print("Humidity= ");
    dataFile.print(hString);
    dataFile.print("% ");
    dataFile.print("Moisture= ");
    dataFile.print(mString);
    dataFile.println("%");
    dataFile.close();
  }
}

field_action_t ElaborateData(unsigned int moist)
{
  if (moist <= mThMin)
  {
    WaterPlants(TO_WATER);
  }
  else if (moist >= mThMax)
  {
```

```
   WaterPlants(NOT_TO_WATER);
  }
  else if (moist > mThMin && moist < mThMax)
  {
   WaterPlants(TO_WATER);
  }
}

void WaterPlants(field_action_t action)
{
  switch(action)
  {
   case TO_WATER:

   digitalWrite(WATERPUMP_PIN, HIGH);
   break;

   case NOT_TO_WATER:

   digitalWrite(WATERPUMP_PIN, LOW);
   break;
  }
}

void FillTank(tankaction_t tankaction)
{
  switch(tankaction)
  {
   case TO_FILL_WATER:

   digitalWrite(GROUND_PUMP, HIGH);
```

```
    break;

    case NOT_TO_FILL_WATER:

    digitalWrite(GROUND_PUMP, LOW);
    break;
  }
}

boolean MinWaterLevelOk(void)
{
  boolean ret = false;

  if(AcquireWaterLevel() >= 30)
  {
    ret = true;
  }
  return ret;
}

boolean MaxWaterLevelReached(void)
{
  boolean ret = false;

  if(AcquireWaterLevel() >= 100)
  {
    ret = true;
  }
  return ret;
}
```

```
void confCommand(BridgeClient client)
{
  String config_command = client.readStringUntil('/');
  float temptemp;
  float humtemp;
  float moisttemp;

  if(config_command == "email")
  {
    FileSystem.remove("/mnt/sda1/www/configureemail.txt");
    File mail = FileSystem.open("/mnt/sda1/www/configureemail.txt", FILE_WRITE);
    email_OnOff = client.parseInt();
    mail.print(email_OnOff);
    mail.close();
    if(email_OnOff)
    {
      calls = 0;
      Serial.println("Email notification has been enabled");
    }
    else
    {
      Serial.println("Email notification has been disabled");
    }
  }
  if(config_command == "limit")
  {
    String type_limit = client.readStringUntil('/');
    if(type_limit == "tmin")
    {
      tThMin = client.parseFloat();
    }
```

```
if(type_limit == "tmax")
{
  tThMax = client.parseFloat();
}
if(type_limit == "hmin")
{
  hThMin = client.parseFloat();
}
if(type_limit == "hmax")
{
  hThMax = client.parseFloat();
}
if(type_limit == "mmin")
{
  mThMin = client.parseFloat();
}
if(type_limit == "mmax")
{
  mThMax = client.parseFloat();
}

// swap values if user sets lower limit grater then upper limit
if(tThMin > tThMax)
{
  temptemp = tThMin;
  tThMin = tThMax;
  tThMax = temptemp;
}
if(hThMin > hThMax)
{
  humtemp = hThMin;
```

```
    hThMin = hThMax;
    hThMax = humtemp;
  }
  if(mThMin > mThMax)
  {
    moisttemp = mThMin;
    mThMin = mThMax;
    mThMax = moisttemp;
  }
  FileSystem.remove("/mnt/sda1/www/limit.txt");
  File limitConf=FileSystem.open("/mnt/sda1/www/limit.txt", FILE_WRITE);
  limitConf.print(tThMin);
  limitConf.print('\n');
  limitConf.print(tThMax);
  limitConf.print('\n');
  limitConf.print(hThMin);
  limitConf.print('\n');
  limitConf.print(hThMax);
  limitConf.print('\n');
  limitConf.print(mThMin);
  limitConf.print('\n');
  limitConf.print(mThMax);
  limitConf.close();
}

if(config_command == "read")
{
  String type_config = client.readStringUntil('\r');
  if(type_config == "email")
  {
    client.print("Email: " + String(email_OnOff));
```

```
  }
  if (type_config == "sendlog")
  {
    runPythonScript(0x05);
  }
  if(type_config == "json")
  {
    client.println("{\"email\":\"" + String(email_OnOff) + "\",\"tmin\":\"" +
    String(tThMin) + "\",\"tmax\":\"" +
    String(tThMax) + "\",\"hmin\":\"" +
    String(hThMin) + "\",\"hmax\":\"" +
    String(hThMax) + "\",\"mmin\":\"" +
    String(mThMin) + "\",\"mmax\":\"" +
    String(mThMax) + "\",\"Temperature\":\"" +
    String(temp) + "\",\"Humidity\":\"" +
    String(humi) + "\",\"Moisture\":\"" +
    String(moist) + "\",\"WaterLevel\":\"" +
    String(waterlevel) + "\"}");
  }
}



if(config_command == "plant")
{
  String type_plant = client.readStringUntil('\r');
  if(type_plant == "Tomato")
  {
    tThMin = 8.00;
    tThMax = 30.00;
    hThMin = 60.00;
```

```
  hThMax = 80.00;
  mThMin = 70.00;
  mThMax = 85.00;
}
if(type_plant == "Wheat")
{
  tThMin = 16.00;
  tThMax = 30.00;
  hThMin = 60.00;
  hThMax = 80.00;
  mThMin = 55.00;
  mThMax = 100.00;
}
if(type_plant == "Rice")
{
  tThMin = 20.00;
  tThMax = 35.00;
  hThMin = 60.00;
  hThMax = 80.00;
  mThMin = 70.00;
  mThMax = 100.00;
}
if(type_plant == "Potato")
{
  tThMin = 15.00;
  tThMax = 30.00;
  hThMin = 60.00;
  hThMax = 80.00;
  mThMin = 80.00;
  mThMax = 95.00;
}
```

```
    if(type_plant == "Grapes")
    {
      tThMin = 20.00;
      tThMax = 35.00;
      hThMin = 60.00;
      hThMax = 80.00;
      mThMin = 65.00;
      mThMax = 85.00;
    }

    FileSystem.remove("/mnt/sda1/www/limit.txt");
    File limitConf = FileSystem.open("/mnt/sda1/www/limit.txt", FILE_WRITE);
    limitConf.print(tThMin);
    limitConf.print('\n');
    limitConf.print(tThMax);
    limitConf.print('\n');
    limitConf.print(hThMin);
    limitConf.print('\n');
    limitConf.print(hThMax);
    limitConf.print('\n');
    limitConf.print(mThMin);
    limitConf.print('\n');
    limitConf.print(mThMax);
    limitConf.close();
  }
}

String getTimeStamp()
{
  String result;
  Process time;
```

```
  // date is a command line utility to get the date and the time
  // in different formats depending on the additional parameter
  time.begin("date");
  time.addParameter("+%T");
                          // T for the time hh:mm:ss
  time.run();             // run the command

  // read the output of the command
  while (time.available() > 0)
  {
    char c = time.read();
    if (c != '\n')
      result += c;
  }
  return result;
}

void runPythonScript(byte paramter)
{
  Process p;
  p.begin("python");
  if(paramter == 0x00)
  {
    p.addParameter("/mnt/sda1/www/tempnot.py");
  }
  if(paramter == 0x01)
  {
    p.addParameter("/mnt/sda1/www/humnot.py");
  }
  if(paramter == 0x02)
```

```
  {
    p.addParameter("/mnt/sda1/www/moistnot.py");
  }
  if(paramter == 0x03)
  {
    p.addParameter("/mnt/sda1/www/TankEmpty.py");
  }
  if(paramter == 0x04)
  {
    p.addParameter("/mnt/sda1/www/TankOkNow.py");
  }
  if(paramter == 0x05)
  {
    p.addParameter("/mnt/sda1/www/logsend.py");
  }
  if(paramter == 0x06)
  {
    p.addParameter("/mnt/sda1/www/deletelog.py");
  }
  p.run();
}
```

Appendix D

Screenshots

Screenshot 1

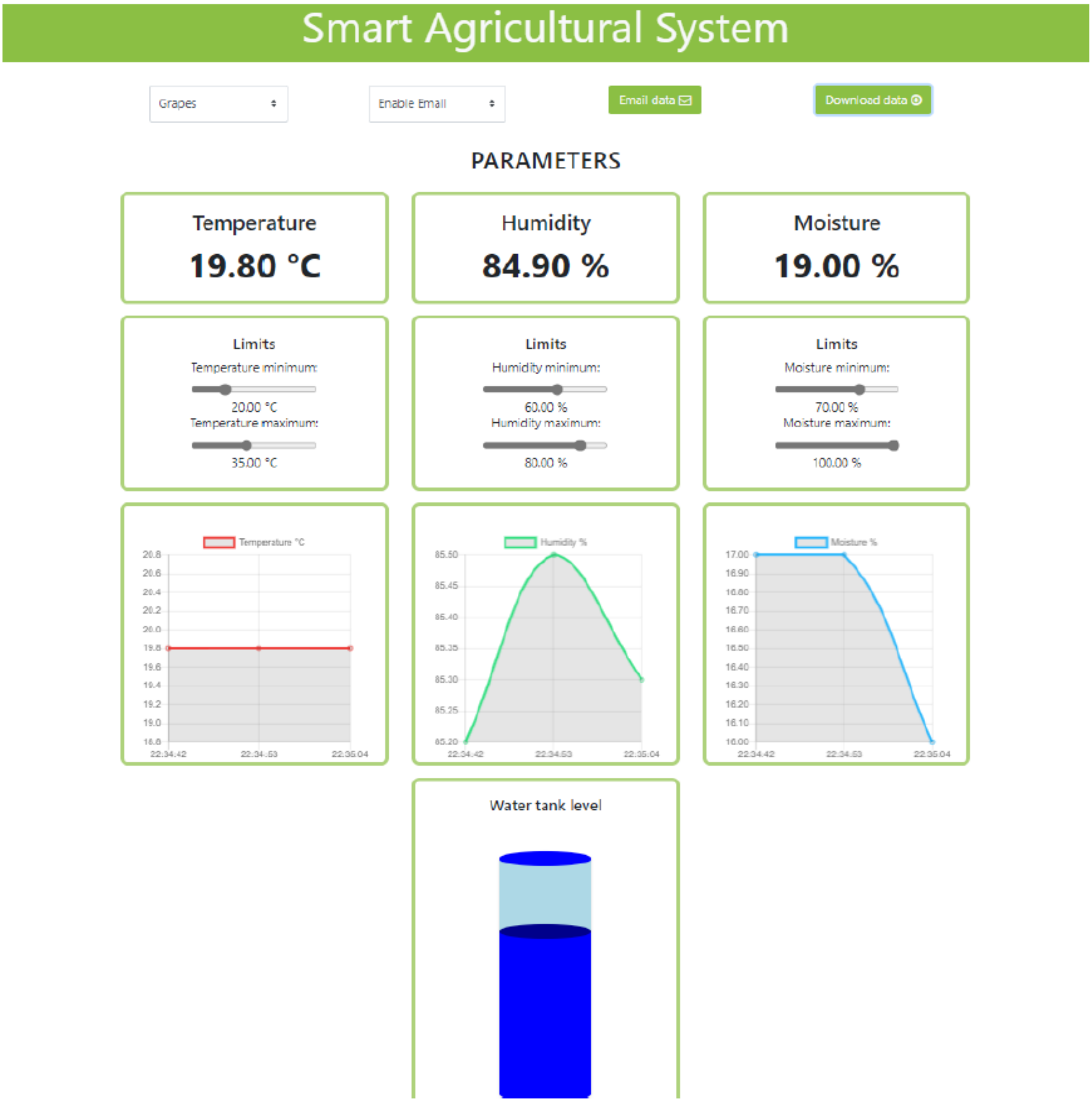

Figure D.1: Dashboard : Real-time Monitoring

Figure D.1 shows "Real-time Crop Monitoring" dashboard provides a comprehensive view of crucial environmental parameters essential for optimal crop management. This interactive dashboard displays real-time data on temperature, moisture, humidity, and the water level in the farmer's water tank. Each metric is presented clearly, allowing farmers to make informed decisions for various crops.

Temperature: This section shows the current temperature readings across different parts of the crops, helping to monitor and adjust to ensure the vines remain in optimal growing conditions.

Moisture: Real-time soil moisture levels are displayed, highlighting areas that may require irrigation to maintain ideal soil conditions for the crops.

Humidity: This feature monitors the ambient humidity, providing data crucial for preventing issues like mildew and ensuring the vines thrive.

Water Level: The dashboard includes a real-time display of the water level in the farmer's water tank, ensuring that there is always enough water available for irrigation needs.

Each of these parameters is presented in an easy-to-read format with visual aids such as graphs and color-coded indicators. This allows for quick assessment and immediate action, promoting efficient vineyard management and enhancing crop yield and quality.

The dashboard serves as a vital tool for modern farmers, integrating technology into agriculture to streamline processes and improve overall productivity. With this real-time monitoring system, farmers can maintain a healthy vineyard by ensuring that the environmental conditions are always optimal for crop growth.

Screenshot 2

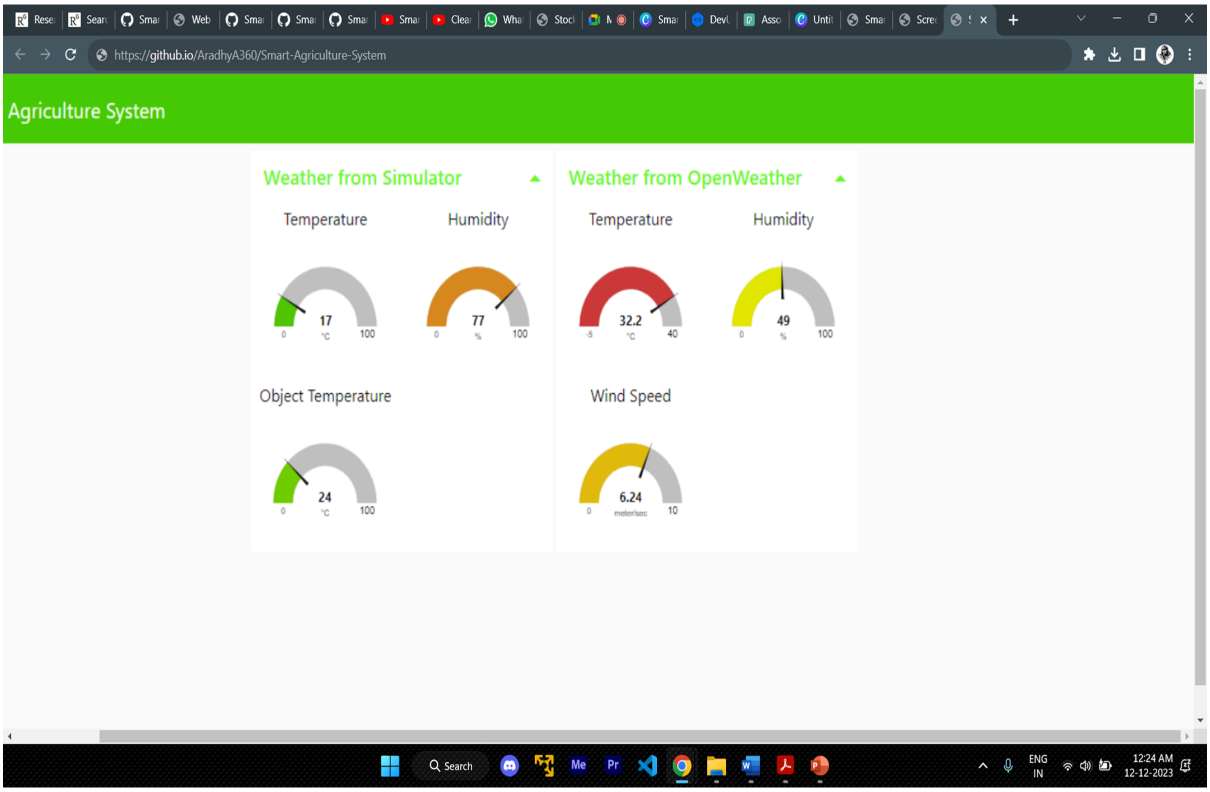

Figure D.2: Monitoring Environmental Conditions and Crop Health

Figure D.2 The temperature is represented with a color gradient, where different colors indicate varying temperature ranges. Similarly, wind speed is illustrated using arrows or lines, where the length and direction of the arrows denote the speed and direction of the wind, respectively.

By integrating these two crucial environmental parameters into a single visual representation, the image allows for a comprehensive understanding of the current conditions affecting the cropland. This integrated approach facilitates timely and precise interventions, helping to ensure that the crops receive the best possible care.

Overall, this detailed depiction of temperature and wind speed serves as an indispensable tool for modern agriculture, promoting efficient and sustainable farming practices.