Table of Contents

1.0 Introduction

1.1 Problem Statement

Due to the rise in climate variability, inefficient resource use and limited access to modern technological tools among small and mid-sized farms, agriculture continues to face significant challenges. Not only that, farmers also struggle with maintaining optimal environmental conditions like soil pH, temperature, humidity and air quality in order to maximise crop yield and ensure sustainability. Although there are several existing smart farming solutions, they are often fragmented, expensive and difficult to scale, which causes poor adoption in developing countries. The lack of real-time insights and automation in agriculture also causes an increase in manual labour, resource wastage and inconsistent agricultural productivity. Therefore, a cost-effective and intelligent IoT-based system that enables data-driven decision-making, environmental automation and remote farm management is needed to support precision agriculture.

1.2 Objectives

Our assignment aims to design and implement an IoT-based smart agriculture system named GrowSync which empowers farmers to monitor, control and optimize key environmental conditions through an integrated sensor-actuator-cloud architecture. The main objectives is to develop a multi-layer IoT architecture that seamlessly incorporates real-time sensing, data transmission, cloud analytics, and automated actuation. We will achieve this by integrating sensors for temperature, humidity, soil pH, gas levels, and pest activity with appropriate actuators for irrigation, ventilation, and pH adjustment. Additionally, the system will leverage cloud platforms such as AWS and Agroview to ensure secure data storage, enable trend visualization, and support machine learning-based forecasting for more informed decision-making. Moreover, a user-friendly mobile application will be developed using Flutter allowing farmers to remotely monitor and control their farm systems from anywhere. Lastly, by carefully selecting optimal components, communication protocols and encryption mechanisms, the system is designed to ensure scalability, affordability, and data security.

2.0 Detailed System Architecture Diagram

2.1 Explanation of System Architecture Diagram

The GrowSync agriculture system is designed to optimize agricultural environments by monitoring and controlling key parameters such as air quality, temperature, humidity, soil conditions, and pest activity. It is built on a multilayered architecture comprising of perception, gateway, cloud, application, and actuation layers and it enables intelligent data-driven farming through seamless sensor integration, cloud analytics, and real-time control.

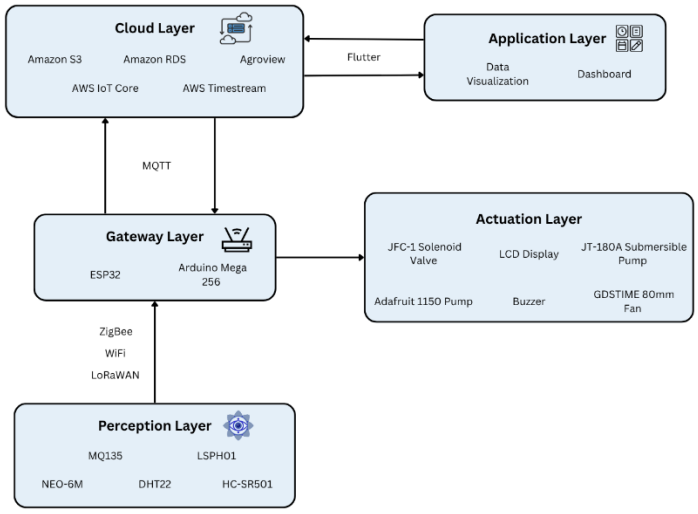

Figure 1 - System Architecture Diagram

Figure 1 illustrates the comprehensive system architecture for the proposed smart agriculture IoT system. The perception layer comprises an array of sensors responsible for monitoring various environmental parameters. The DHT22 sensor provides precise temperature and humidity readings crucial for climate control and irrigation scheduling (SparkFun Electronics, 2018). Then, the MQ135 gas sensor monitors air quality by detecting harmful gases such as ammonia, CO_2 and smoke to ensure a safe environment (Winsen, 2022). For soil analysis, the LSPH01 LoRaWAN sensor offers real-time pH values critical for nutrient management (Dragino Technology Co., Ltd., 2025). Moreover, motion activity is detected by the HC-SR501 PIR sensor which can trigger pest deterrents or alert systems while the NEO-6M GPS module offers geolocation data for spatial tracking and mapping of farm zones.

Then, the data from these sensors is transmitted to the gateway layer, which features both the ESP32 and Arduino Mega 2560 microcontrollers. The ESP32 handles wireless communication via ZigBee, Wi-Fi, and LoRaWAN and they do initial data filtering and forward to the cloud using MQTT protocol (Espressif Systems, 2022). Meanwhile, the Arduino Mega 256 with its ample GPIO pins and memory manages actuator control and real-time sensor interfacing (Arduino, 2015). As a result, this layer ensures efficient data handling and system responsiveness. The cloud layer utilizes AWS services including AWS IoT Core, Amazon S3, Amazon RDS, and Timestream for device management, data storage, and time-series analytics while integration with platforms like Agroview supports visualization and analytics for large-scale decision-making.

Then, data is securely communicated to the application layer via Flutter framework allowing users to monitor and interact with the system through mobile apps or web dashboards. The actuation layer responds dynamically based on sensor input and user-defined threshold using a JFC-1 solenoid valve and the JT-180A submersible pump to automate irrigation based on soil moisture and pH. For pH correction, the Adafruit 1150 peristaltic pump can dispense acidic or basic solutions while an LCD display and buzzer provide local feedback and alerts, and the GDSTIME 80mm fan maintains environmental comfort based on temperature or gas readings.

2.2 Flow of Data

The GrowSync system for agriculture is designed to intelligently monitor and manage farm environments by integrating sensors, controllers, cloud services, and actuators in a unified multilayered architecture. The data flow begins at the perception layer where sensors like DHT22, MQ135, LSPH01, and HC-SR501 collect environmental data such as temperature, humidity, air quality, soil pH, and motion. This data is then transmitted via wireless protocols such as Zigbee, Wi-Fi, or LoRaWAN to the gateway layer, where the ESP32 and Arduino Mega 2560 handle preprocessing and forward the information to the cloud layer such as AWS IoT Core, Amazon RDS using the MQTT protocol. In the application layer, platforms like Agroview or mobile/web dashboards enable users to visualize real-time data, analyze trends, and send commands. Based on these insights, the actuation layer responds using devices such as solenoid valves, water pumps, fans, and buzzers to maintain optimal growing conditions.

3.0 Hardware Component

3.1 Soil Moisture and pH Sensor

Figure 2 - LSP01 LoRaWAN Soil pH Sensor

The Dragino LSPH01 LoRaWAN Soil pH Sensor which also measures soil temperature is ideal for long-term agricultural use due to its durable IP68-rated waterproof probe and stable AgCl and metal electrodes (Dragino Technology Co., Ltd., 2025). This is because it supports ultra-long-range, low-power LoRaWAN communication which is perfect for remote farms lacking Wi-Fi or cellular coverage. Besides, it is powered by an 8500mAh Li-$SOCl_2$ battery and it enables up to 5 years of maintenance-free operation (LSPH01 - LoRaWAN Soil PH Sensor, 2024). Therefore, all these features such as remote configuration, temperature compensation, and multi-band support enhance accuracy and adaptability making it a reliable tool for precise soil monitoring and sustainable crop management.

3.2 Temperature and Humidity Sensor

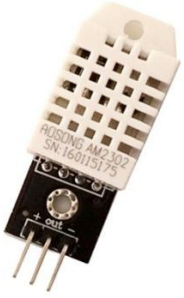

Figure 3 - DHT22 (AM2302) Temperature and Humidity Sensor

The DHT22 (AM2302) is a reliable, low-cost digital sensor used in this GrowSync system to measure temperature and humidity. It offers a wide sensing range from -40 °C to +80 °C with ±0.5 °C accuracy and 0–100% RH with ±2–5% RH accuracy (SparkFun Electronics, 2018). Furthermore, its single-wire digital interface simplifies microcontroller integration while its low power consumption and robust performance make it ideal for long-term, field-based deployment (Devanand et al., 2020). The DHT22 is frequently used in scalable IoT architectures and commonly connected to microcontrollers like the ESP32 enabling regular data uploads to cloud platforms for real-time monitoring and automation. All and all, its proven role in integrated sensor networks and smart farming solutions confirms its suitability for enhancing climate-aware crop management (Rajesh et al., 2020; Arshad et al., 2022).

3.3 Gas Sensor

Figure 4 - MQ135 Gas Sensor

The MQ135 gas sensor is chosen for this GrowSync system due to its affordability, analogue compatibility and ability to detect a wide range of harmful gases such as CO_2, NH_3, benzene, smoke and VOCs making it ideal for both greenhouse and open-field environments. The sensor outputs analogue voltage based on gas concentration which is easily interpreted by microcontrollers like the ESP32 for real-time monitoring and automation (Devanand et al., 2020). Besides, it operates at 5V and consuming under 200 mA and includes a heating element for accurate sensing and requires a preheating time of 24–48 hours to stabilize readings (Olimex, 2021). With detection ranges typically spanning 10–1000 ppm for CO_2 and 10–300 ppm for NH_3, it effectively triggers safety responses such as activating buzzers or ventilation fans when gas

levels exceed safe thresholds (Winsen, 2022). Its wide detection range and ease of integration make it a vital component in the system's air quality assurance mechanism.

3.4 PIR Sensor

Figure 5 - HC-SR501 Infrared Sensor

The HC-SR501 Passive Infrared (PIR) Motion Sensor is integrated into the GrowSync system to detect motion from humans or animals using its pyroelectric sensing mechanism and dual-element probe design. It is ideal for securing greenhouses, open fields or storage zones by triggering alarms or deterrent systems when motion is detected as it has a wide detection angle of up to 110° and a range of approximately 7 meters (Rajesh et al., 2020). Moreover, the sensor's sensitivity and delay time are adjustable and its simple TTL output ensures seamless integration with microcontrollers for real-time responses. The HC-SR501 is energy-efficient and suited for battery or solar-powered systems as it operates on 5V–20V and consuming under 50µA in standby mode while maintaining performance in temperatures ranging from -15°C to +70°C (HC-SR501 Datasheet, 2013; Devanand et al., 2020). As a result, its affordability, adaptability, and low power consumption make it a key component in building a cost-effective and scalable motion detection system for agriculture.

3.5 GPS Module

Figure 6 - NEO-6M GPS Module

The u-blox NEO-6M GPS module is chosen for field mapping and geolocation tracking due to its compact size, reliability, and seamless integration with IoT systems. It is ideal for battery-powered precision agriculture devices as it has a fast hot start time under 1 second, up to 2.5 m accuracy and low power consumption (u-blox, 2015). Additionally, it supports a 2.7–3.6 V supply and UART communication, making it fully compatible with microcontrollers like the ESP32. Moreover, features like Assisted-GPS (A-GPS), a -161 dBm sensitivity and up to 5 Hz data refresh rate ensure reliable performance even in obstructed rural conditions. In smart agriculture, it enables geofencing, yield mapping, variable rate application, and spatial analysis which is key to optimizing inputs and boosting sustainability (Jayanthi et al., 2007).

3.6 Wi-Fi Module

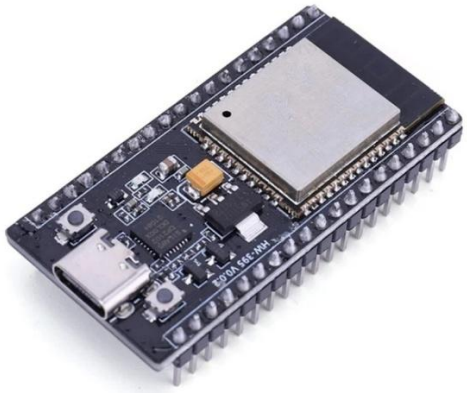

Figure 7 - ESP32 Wi-Fi Module

The ESP32 is selected as the core microcontroller and gateway for the GrowSync system due to its built-in Wi-Fi and Bluetooth enabling seamless wireless communication and real-time remote monitoring (Pereira et al., 2023). When powered by a dual-core 32-bit LX6 processor and offering 48 GPIO pins, it can efficiently handle multiple sensor inputs such as DHT22, soil moisture, GPS and actuators like irrigation pumps. Its support for analogue and digital I/O along with energy-saving sleep modes makes it ideal for battery-powered field deployment (Espressif Systems, 2020). Moreover, its affordability, compact integration, and robust performance enable a streamlined architecture without needing separate communication modules supporting reliable sensing and control in smart greenhouses and precision agriculture (Manzanero-Vazquez et al., 2021).

3.7 Arduino Module

Figure 8 - Arduino Mega 256 Module

The Arduino Mega 2560 is selected as the supporting microcontroller due to its extensive I/O capabilities and strong compatibility with various sensors and actuators. With 54 digital I/O pins, 16 analogue inputs and 4 UART ports, it interfaces efficiently with modules like the MQ135, DHT22, PIR sensor, fans, and buzzers making it ideal for managing multiple data streams and automation tasks in precision agriculture. It also operates at 5V and supports PWM outputs enabling precise control of devices like irrigation valves (Arduino, 2015). Additionally, its ATmega2560 processor runs at 16 MHz, along with 256 KB flash memory, 8 KB SRAM and 4 KB EEPROM supports real-time control and logic execution. When paired with the ESP32 which handles cloud connectivity, the Mega strengthens I/O management and boosts system scalability.

3.8 Actuators

Several actuators are needed to support automated control in the smart agriculture system and each of them is selected for specific environmental management tasks. One of the first actuators is a DC ventilation fan such as the GDSTIME 12V 80mm Fan which is used to regulate air temperature and expel excessive CO_2 or other pollutants when gas levels exceed safe thresholds. Then, a mini water pump like the JT-180A 5V DC submersible pump is needed for humidity control and is deployed to drive a misting system increasing air moisture when levels fall below the desired range. These actuators are essential for maintaining optimal plant growth conditions and are easily triggered through relay modules connected to the Arduino or ESP32.

Moreover, peristaltic dosing pumps such as the Adafruit 1150 12V DC pump is required for soil pH regulation and they are employed to inject acidic or alkaline solutions based on sensor feedback. This closed-loop feedback system ensures precise correction of soil chemistry. Lastly, a solenoid valve like the JFC-1 12V solenoid valve is used to control irrigation lines enabling efficient water distribution based on soil moisture data. These actuators not only automate routine interventions but also respond dynamically to real-time environmental changes enhancing system intelligence and reducing manual labour.

The 16x2 Liquid Crystal Display (LCD) and buzzer also serve as key actuators for local user interaction and alert mechanisms within the smart agriculture system. The LCD offers real-time visual feedback on parameters such as temperature, humidity, soil pH, gas concentration and irrigation status making it ideal for on-site monitoring in areas with limited connectivity (Rajesh et al., 2020; Devanand et al., 2020). Lastly, a 5V piezoelectric buzzer provides immediate auditory alerts for critical conditions like hazardous gas levels or pest presence with a sound level of 85 dB at 2.3 kHz (Mouser Electronics, 2018). As a result, the buzzer ensures prompt response even when visual feedback is missed enhancing safety and responsiveness in semi-supervised field environments.

3.9 Hardware Placement

Hardware Component	Placement Location	Justification
LSPH01 Soil Moisture and pH Sensor	Buried 5–10 cm near roots across multiple rows	Monitors root-zone moisture and pH value for better irrigation control
DH22 Temperature and Humidity Sensor	Centrally located and under a shaded area at crop height	Accurate ambient temperature and humidity without sun interference
MQ135 Gas Sensor	Slightly above crops near chemical storage or enclosed spots	Detects harmful gases or air pollutants in critical zones
HC-SR501 PIR Motion Sensor	Facing inward at field entrance or perimeter	Detects animal/human intrusion and triggers alarms
NEO-6M GPS	Centrally located on a pole or box with sky view	Provides real-time location data for mapping and tagging
ESP32 Microcontroller	Centrally placed in a weatherproof control box	Manages sensors, actuators, and wireless communication
Arduino Mega 2560	Inside the same control box as ESP32	Handles high I/O tasks and actuator control
LCD Display	Mounted on the control box or near entrance	Displays sensor readings for quick on-site monitoring
Buzzer	With control box or near crops	Alerts users to critical conditions via sound
Solenoid Valve (JFC-1)	In line with irrigation pipe near pump or water source	Controls water flow based on moisture or pH
Submersible Water Pump (JT-180A)	Submerged in water tank or nearby reservoir	Pumps water for irrigation when activated
Peristaltic Pump (Adafruit 1150)	Near nutrient or pH solution container in a sheltered spot	Dispenses solution for pH/nutrient adjustment
GDSTIME 80mm Fan	Inside greenhouse or crop zone mounted on pole or wall	Improves airflow when temperature or gas levels are high

Table 1 - Hardware Placement

4.0 Software Platform

4.1 Cloud Platform

A modern smart agriculture system relies heavily on robust software platforms and a mobile app development framework that can support multiple tasks being carried out simultaneously, especially in real time. For example, data collection, data storage, data processing and decision making. To make this happen, a scalable cloud-based infrastructures need to be built and combined with mobile-accessible interfaces. In this way, farmers can monitor and manage their farm operations remotely and efficiently. There are several cloud software platforms and mobile application frameworks that are relevant to smart agriculture and can be integrated to make it practical.

4.1.1 Generic Cloud-Based Platforms

The first example of online software platforms is the generic cloud-based platforms. These platforms serve as foundational infrastructure for many smart agriculture applications. These systems function as centralized repositories where sensor data from different sources like weather stations, soil sensors and imaging devices are collected, processed and analysed. They offer options like Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) which enable on-demand resource allocation. These systems provide key features like automated crop recommendations, forecasting, market updates and ICT services such as online expert consultations and weather predictions. The advantages of these platforms are secure, scalable and cost-effective. In this way, it provides farmers with real-time insights and decision support at scale while reducing computational load on local devices (Junaid, 2021).

4.1.2 AGRICLOUD

AGRICLOUD is a proposed AI-enhanced cloud system that is developed to support complex smart farming scenarios. It can process agricultural data in multiple formats like images, text, video and maps, using Support Vector Machine (SVM). The platform shows significant improvements in performance such as reductions in overhead time, energy consumption and execution delays after

being tested using CloudSim4.0. It allows farmers to use it at a supervisory level as it turns massive volumes of sensor and field data into understandable, useful insights. It is good for applications that use large datasets for automated classification as it can run them very fast (Junaid, 2021).

4.1.3 Agroview

Agroview, which is a cloud platform based on AWS can also be used for smart agriculture. It is an AI-based application that processes and visualizes data collected from Unmanned Aerial Vehicles (UAVs) for precision agriculture. It uses the high scalability and efficiency of AWS to carry out tasks that require a lot of CPU and GPU power like mapping crop health, estimating crop height, and detecting gaps between trees. The platform is designed to convert aerial imagery into actionable agricultural insights rapidly using technologies like machine learning and artificial intelligence. It helps to reduce manual scouting and also supports high-precision farming decisions. Agroview is able to serve various farm sizes while ensuring its reliable performance and cost-effectiveness using AWS's flexible infrastructure. In this way, it serves as an ideal cloud platform for large-scale, data-driven agricultural operations (Soussi, 2023).

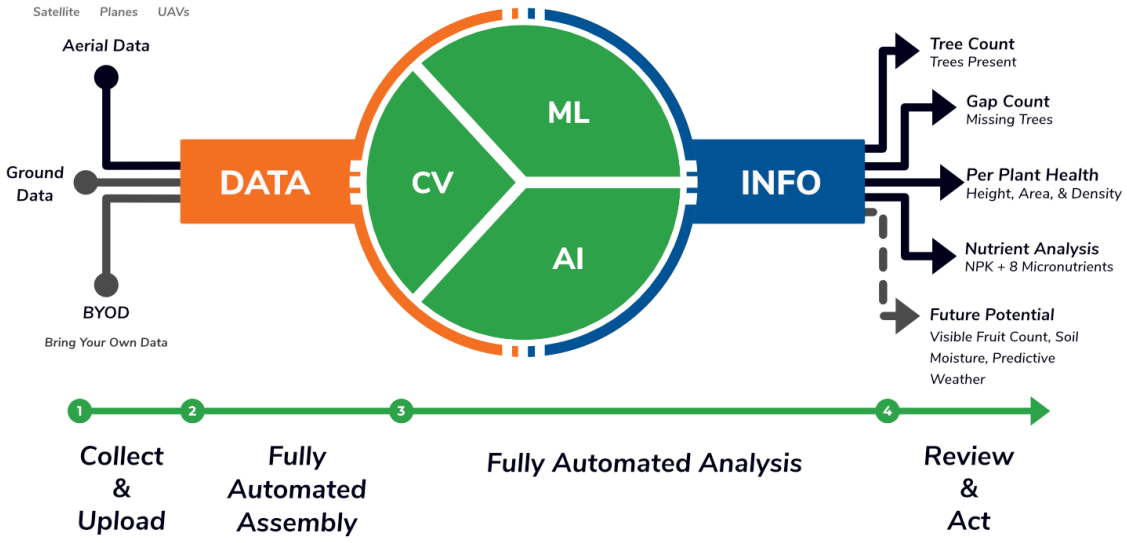

Figure 9 - How AGROVIEW Works (AGROVIEW)

4.1.4 ThingSpeak Cloud

ThingSpeak Cloud is another widely used cloud platform used to collect and visualize IoT sensor data, which is suitable for smart agriculture. It is effective to be used in educational, experimental and small-to-medium scale agricultural setups. ThingSpeak allows farmers to log values including temperature, humidity, and soil moisture for next crop planning and forecasting. It supports MATLAB analytics and has a user-friendly dashboard that allows jobs like setting up, trend visualizing, and data exporting to be done simply. Its simplicity and affordability make it a good tool for farms that do not have high infrastructure costs but need basic data analytics and cloud storage (Soussi, 2023).

4.1.5 ONENE-Based Greenhouse System

Not only that, the ONENET-based greenhouse system is able to monitor controlled environments like greenhouses remotely. It gathers information on soil moisture, light intensity, air temperature, and humidity using an ESP8266 Wi-Fi module, an STM32 microcontroller, and the ONENET IoT cloud platform. In this way, farmers can monitor and manage greenhouse conditions in real time using this data, which is sent to a web dashboard that is accessible from a desktop or mobile device. This system is simple, cost-efficient and has high detection accuracy, which allows it to be used for small-scale commercial greenhouses that need automation and real-time data access (Soussi, 2023).

4.2 Mobile App Development Framework

4.2.1 Flutter

For the mobile app development framework, Flutter which is developed by Google is widely recognized for its performance and flexibility across platforms. It enables programmers to create a single Dart codebase and publish apps for web, iOS, Android and embedded platforms. Flutter's hot reload feature is crucial for smart agriculture as it allows for real-time user interface modifications when incorporating dynamic sensor data like soil moisture or meteorological conditions. Not only that, its adaptable widget system enables developers to design user-friendly

farm monitoring and control dashboards. Its embedded device compatibility also makes it appropriate for IoT gateways or on-field tablets. These features are suitable for building interactive agriculture applications that require real-time responsiveness (Grace, 2025).

4.2.2 React Native

React Native which is maintained by Meta is another cross-platform framework built on JavaScript. It offers fast refresh capabilities and component-based architecture that helps accelerate development and testing. It is suitable to be used for smart agriculture that requires frequent UI updates, data entry forms, GPS-based tracking or integration with external APIs. It also lowers the barrier to entry for front-end developers as it relies on JavaScript. The open-source ecosystem provides a variety of ready-to-use libraries for integrating sensor data and enabling Bluetooth-based field control with IoT devices. The flexibility, ease of use and robust feature set of React Native are suitable to be use as a mobile app development framework (Grace, 2025).

4.2.3 JetBrains' Kotlin Multiplatform

JetBrains' Kotlin Multiplatform allows developers to share core logic across Android, iOS, and other platforms while retaining native UI performance. This makes it suitable to be used as smart agriculture prioritizes backend consistency and real-time data syncing. Not only that, Kotlin is perfect for improving legacy systems without requiring a complete redevelopment because of its smooth Android integration and compatibility with already-existing native applications. Shared business logic ensures effective development, and its support for native APIs provides high performance. All in all, Kotlin Multiplatform is a good choice for building a scalable and performance-critical smart agriculture mobile app (Grace, 2025).

4.3 Selection and Justification

Among the online software platforms, Agroview is chosen as the most suitable platform for smart agriculture. Unlike generic cloud platforms like ThingSpeak or ONENET which only collect sensor data and provide basic visualization, Agroview converts aerial imagery into actionable

insights which allow farmers to take suitable action. Not only that, Agroview which is built on Amazon Web Services gains advantages in scalability, reliability and computational power. Agroview can be integrated with several AWS services like Amazon S3, Amazon IoT Core, Amazon RDS and AWS Timestream to enhance its capabilities. Amazon S3 acts as a safe and scalable storage system for drone imagery and processed output. Amazon IoT Core controls real-time data streams from field-deployed IoT devices like soil moisture, temperature and weather sensors. Apart from that, Amazon RDS can be used to store structured data like crop health reports, event logs and decision records while AWS Timestream is used to handle time-series data from the IoT sensors and enable historical trend analysis and predictive forecasting.

Flutter is chosen as the mobile app development framework because it strongly supports real-time data visualization, responsive design, and multi-platform deployment. Flutter offers a rich set of customizable widgets and native-like performance out of the box, unlike React Native which relies heavily on third-party libraries for UI consistency and may face performance issues with complex interfaces. While Kotlin Multiplatform excels in backend logic sharing and native Android integration, it requires more platform-specific coding for UI and is less mature for full cross-platform development. Flutter's hot reload feature, seamless integration with cloud APIs, and support for web, mobile, and embedded devices make it the most flexible and efficient choice for building real-time, sensor-driven smart agriculture applications.

5.0 Communication Protocol Selection and Justification

There will be various data transmission paths in the IoT system, each utilizing different communication protocols depending on different factors such as range and latency. The figure below illustrates a simplified data flow throughout different components within the IoT system.

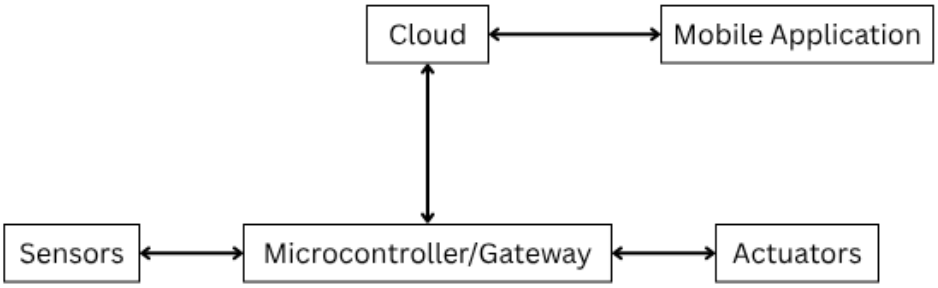

Figure 10 - Data Transmission Pathways

The sensors and actuators within the system communicate with each other through a microcontroller, which processes the collected data from the sensor and sends control signals to the appropriate actuator. There will be two types of microcontrollers in the system:

- A standard microcontroller, either embedded within a device, or positioned at a central location to collect data and manage signals sensors and actuators remotely.
- A gateway microcontroller, capable of transferring and receiving data to and from the cloud for processing and storage.

The final transmission path is between the cloud and the mobile application, where the mobile application can access data stored in the cloud, and send commands or configuration settings back to the system.

5.1 Sensors, Actuators and Microcontrollers

There will be 3 different ranges for data transmission between sensors, actuators and microcontrollers, which are short, medium and long distances, depending on the devices used. Each range will use different types of communication protocols based on its suitability for the task. Table ??? outlines the type of communication protocol used for each distance, alongside the factors associated with it.

Distance	Short	Medium	Long
Communication Protocol	ZigBee	Zigbee Mesh (One gateway per zone)	LoRaWAN
Bandwidth	2MHz		125 – 500KHz
Frequency Band	2.4GHz		915 MHz
Latency	20 – 30ms	30 – 60ms	~10s
Power consumption	Low		Very Low
Range	50 – 100m	Coverage area of approximately 300m^2	2 – 5km
Security	High (AES-128)		

Table 2 - Communication Protocols for Sensors, Actuators, and Microcontrollers (Avşar & Mowla, 2022) (Jawad et al., 2017)

- Short

  Short distance data transmissions consist of sensors close to the central microcontroller, such as temperature, humidity, UV, and gas sensors. Since these sensors are capable of covering a wide range of land area at once, they can be placed close to the microcontroller, which reduces the distance for data transmission. Therefore, Zigbee is used as the communication protocol due to its low latency of approximately 20 to 30ms at short ranges. Considering that sensor data is usually small in size, Zigbee's 2MHz bandwidth is well suited for transmitting it.

- Medium

  Sensors such as pH and soil moisture sensors are usually distributed across the field to cover more land area as soil conditions may differ across short distances. This same applies to PIR sensors, which has a short detection range and has to be installed throughout the field to cover the entire area. As a result, a Zigbee mesh network is used, where each sensor acts as a node with rerouting capabilities, allowing data to be transmitted hop by hop until the central node, which is the microcontroller (Navarro et al., 2020). This topology allows sensors and actuators to communicate over a wide area. Although this slightly increases the latency of data transmission to the central node, the trade-off is worth it due to being able to cover a larger area otherwise considered impossible. Additionally, Zigbee consumes a low amount of power, allowing it to be widely deployed across the area.

- Long

  Long distance transmissions involve agricultural zones which are located far away from the central building, potentially up to 1km and above. The sensor and actuator data collected by the central microcontroller of each zone has to be transmitted to the main gateway microcontroller to be transferred to the cloud. Additionally, technologies utilizing GPS modules may be used for spatial analysis of agricultural zones for geofencing, which are usually located at the borders of the agricultural zones. LoRaWAN is used as the communication protocol for these long-distance transmissions since the amount of data sent is typically very small, making it ideal for its narrow bandwidth range of 125 – 500KHz. As the data doesn't require constant monitoring with a high refresh rate, LoRaWAN's latency of approximately 10 seconds is suitable for the task. LoRaWAN's very low power consumption allows it to be used in a large-scale across long distances, without the need for frequent battery replacements (Ojo et al., 2021).

All communication protocols used for data transmission between sensors, actuators, and microcontrollers uses Advanced Encryption Standard (AES)-128 algorithm for encryption and integrity checking, making the communication highly reliable and safe from external interception (Prodanović et al., 2020). This ensures a uniform security standard across all devices in this transmission path which reduces development complexity. In addition, AES-128 has lower power consumption and processing time than AES-192 and AES-256, thus reducing overall computational overhead on devices (Phithak Thaenkaew et al., 2023).

5.2 Gateway Microcontroller and Cloud

Communication Protocol	MQTT over WiFi
Bandwidth	22MHz
Frequency Band	2.4GHz
Latency	50ms
Power consumption	High
Range	20 – 100m

Security	High (TLS)

Table 3 - Communication Protocol for Gateway and Cloud (Avşar & Mowla, 2022) (Jawad et al., 2017)

Instead of each sensor having microcontrollers capable of transferring data directly to the cloud, a central gateway microcontroller will be used, minimizing the need to purchase high-cost sensors. This approach reduces the overall power consumption of the system and allows for a centralized network topology, making managing batches of sensor data easier and safer.

This data transmission path uses Message Queuing Telemetry Transport (MQTT) operating over a Wi-Fi network as its communication protocol. Wi-Fi is chosen due to its widespread usage for internet connectivity, as well as having a bandwidth of 22MHz at 2.4GHz frequency band, allowing the gateway microcontroller to transfer a large amount of data at once. The minor latency of approximately 50ms is negligible since it doesn't affect the normal operations of the IoT system. Due to the high power consumption of Wi-Fi networks, the gateway microcontroller will be positioned at a central building located within the range of a Wi-Fi router, providing internet access to transfer data to the cloud.

MQTT is used as the messaging protocol due to its high degree of reliability and scalability. It sends data using a streamlined data transmission method, minimizing the loss of power and preserving network performance during the transmission process (Hsu, 2024). The MQTT protocol is encrypted using Transport Layer Security (TLS) with the AES-256 algorithm, which ensures the confidentiality and integrity of data during transmission. TLS is a widely used protocol for providing a secure communication channel between two parties, namely a server and a client. It prevents unauthorized parties from intercepting and tampering with the data during transmission (NCSC, 2021). The combination of MQTT and TLS further strengthens the security of data transmission from potential attacks (Shin & Jeon, 2024).

5.3 Cloud and Mobile Application

Communication Protocol	HTTPS over WiFi
Bandwidth	22MHz
Frequency Band	2.4GHz
Latency	50ms
Power consumption	High
Range	20 – 100m
Security	High (TLS)

Table 4 - Communication Protocol for Cloud and Mobile Application (Avşar & Mowla, 2022) (Jawad et al., 2017)

Similar to the transmission path between the gateway microcontroller and the cloud, the path between the cloud and mobile application uses a Wi-Fi network, with the primary difference being that Hypertext Transfer Protocol Secure (HTTPS) is used instead of MQTT. In this case, the device running the mobile application must be within the range of a Wi-Fi source to obtain internet access to the cloud server. HTTPS is a widely accepted protocol based on the request/reply model, suitable for client/server architectures such as communications between cloud and mobile applications. In this process, the mobile application sends a request message to the cloud, which processes it and returns a response message. HTTPS uses TLS as its security protocol to encrypt data and establish a secure communication channel between the cloud and the mobile application (Dizdarević et al., 2019).

5.4 Justification of Protocols

Sensors, actuators and microcontrollers in the local network utilize low-power communication protocols such as ZigBee and LoRaWAN to reduce the overall power consumption since they will be deployed at a large scale in precision agriculture. Moreover, the aforementioned devices are designed to be cheaper and have a lower computational capacity to minimize costs, so simpler encryption methods such as AES-128 are used to reduce processing requirements.

A gateway microcontroller will be used to connect the local network to the cloud. This device collects data from local sensors and actuators and transmits it to the cloud using MQTT operating over a Wi-Fi network, leveraging Wi-Fi's higher bandwidth and reliability to transmit large amounts of data at once. Due to the higher computational capacity of the gateway microcontroller, it can utilize stronger security protocols such as TLS to secure data transmission to the cloud.

Finally, HTTPS will be used as the communication protocol between the cloud and mobile applications since it is a widely adopted and reliable protocol for transmitting data over the internet. HTTPS is a secure version of the HTTP protocol, since it uses TLS to encrypt its communication channel, thus ensuring data confidentiality and integrity. Wi-Fi is used as the medium for this communication due to its high bandwidth, allowing for efficient data exchange between the mobile application and the cloud server, such as downloading sensor data, receiving alerts or updating configuration settings.

6.0 Data Storage and Management

Table 5 outlines the 3 main data types existing in the IoT system, namely sensor data, environmental logs, and system settings, with their respective data structures, processing types, and database technologies used for storage.

Data Type	Sensor Data	Environmental Logs	System Settings
Data Structure	Structured	Semi-structured	Structured
Processing Type/Update Frequency	Stream	Batch	Event-based
Database Technology	Amazon Timestream	Amazon S3	Amazon RDS

Table 5 - Comparison of Data Structures, Processing Methods, and Database Technologies for Different Data Types

6.1 Sensor Data

Sensor data typically exist as structured data and will be continuously transferred to the cloud for storage and real-time processing, which requires stream processing. This approach ensures that time-sensitive data, such as soil moisture, pH, and temperature readings can be processed and responded to immediately with the appropriate actuators. Amazon TimeStream will be used as the database technology to store sensor data for time-series analytics (Kumar Sinha, 2024). TimeStream can effectively and efficiently store, query, and retrieve large volumes of time-series data for analysis and visualization purposes (Rashmi R et al., 2023).

6.2 Environmental Logs

Environmental logs, such as system status, errors, and security breaches are usually in the form of semi-structured or unstructured data and do not need to be updated regularly. Therefore, they will be transferred to the cloud in batches for storage in Amazon S3, which provides durable and scalable data storage (Rashmi R et al., 2023). S3's log storage capability allows for the collection

and storage of environmental logs securely in tamper resistant storage, while also providing evaluating, monitoring, alerting, and auditing of access and actions in the IoT system (AWS, 2025).

6.3 System Settings

System settings will be stored as structured data and processed upon event triggers, such as when a user reconfigures the settings for a specific sensor. The changes will be updated and stored in the cloud almost immediately, making it a process comparable to real-time processing. These settings will be managed using Amazon RDS, a relational database service that provides SQL databases for storing system settings and configurations (Filip et al., 2022).

6.4 Data Security Measures

6.4.1 Data Encryption

- At Rest

  While data is at rest, for example, when pending for transmission in the gateway microcontroller or stored in the cloud, it is encrypted using the AES-256 algorithm, which offers stronger security than the AES-128 algorithm. Although AES-256 requires more power and computational time, this trade-off is justified due to the sensitivity and importance of the collected data (Raheja & Kumar Manocha, 2022). Additionally, when the data is at rest in the cloud, the power consumption and computational time required to encrypt will be handled by AWS services, indicating that the local network infrastructure does not need to expend resources on encryption operations.

- In Transit

  Before data is transmitted from the gateway to the cloud, it will be encrypted using TLS, which employs the AES-256 algorithm to ensure data confidentiality and integrity during transmission. However, data transmission between sensors, actuators, and microcontrollers will only use the AES-128 algorithm for encryption since they have limited power and processing capabilities (Phithak Thaenkaew et al., 2023).

7.0 Environmental Control Algorithm

The use of smart control algorithms will assist our system, GrowSync, in managing key environmental conditions, resulting in the best possible crop growth environment, lowering disease risk, and maximizing resource utilization. This section will be covering the most impactful environmental conditions for optimal crop growth, such as temperature & humidity control, air quality control, pH regulation control, and lastly, pest control.

7.1 Temperature and Humidity Control

Temperature and humidity are two climatic factors that influence plant growth, metabolism, and productivity by regulating processes such as photosynthesis, respiration, transpiration, and nutrient transport. Therefore, maintaining optimal temperature ranges and both soil and air humidity levels is essential for healthy crop development (Logicsun, 2024).

7.1.1 Temperature and Humidity Control Parameters

Here are some control parameters to consider:

1. **Temperature Setpoints:** Ideal range (e.g., 20–30°C depending on crops)
2. **Humidity Levels:** Often kept between 50%–80% RH
3. **Time of Day:** Can affect heating/cooling needs
4. **Seasonal Weather:** May require dynamic adjustment

7.1.2 Sensors & Actuators

Sensors: A low-cost combined temperature/humidity sensor such as the DHT22 (AM2302) is suitable (RichardP4, 2017). It is more accurate and responsive than the cheaper DHT11.

Actuators: Typical actuators include cooling/heating fans, a heater or exhaust vent, and a water pump or misting system for humidity. For example, a relay-controlled DC fan or ventilation fan can cool air, while a heater can warm it. A water pump can drive misting nozzles to increase humidity. These can be driven by relay modules or MOSFETs from Arduino GPIO pins (AdruinoGetStarted, 2025).

7.1.3 Control Logic and Operation

The Arduino repeatedly reads temperature (T) and humidity (H) and compares them to desired setpoints. A simple on/off control often suffices. For example, if T > upperThreshold, the fan will be turned on or the vent will be open. If T < lowerThreshold, it will be off. Likewise, if humidity H falls below a lower setpoint, a mist pump will be started. If H exceeds an upper setpoint, the mist pump will stop or the vent will be opened. This approach is energy efficient because actuators run only when needed. In practice, one can use hysteresis or two thresholds to prevent rapid toggling. For more stable control and large greenhouses, a PID controller can be implemented (AdruinoGetStarted, 2025).

7.1.4 Pseudocode

Example Scenario: If the temperature goes above 30°C and the humidity falls below 50%, the system will automatically turn on the fan and start the misting system to bring the environment back to optimal levels.

```
float T = dht.readTemperature();    // read from DHT22
float H = dht.readHumidity();
if (isnan(T) || isnan(H)) return;  // sensor error check

// Temperature control
if (T > 30) {
  digitalWrite(FAN_PIN, HIGH);      // turn fan on
} else if (T < 28) {
  digitalWrite(FAN_PIN, LOW);       // turn fan off
}

// Humidity control
if (H < 50) {
  digitalWrite(MIST_PIN, HIGH);     // start misting pump
} else if (H > 60) {
  digitalWrite(MIST_PIN, LOW);      // stop misting
}
```

Figure 11 - Pseudocode for Temperature & Humidity Control

7.2 Air Quality Control

While CO_2 enrichment can benefit greenhouse crops, exposure to harmful air pollutants such as ammonia or smoke can cause visible leaf damage, reduced growth, and yield loss, potentially leading to premature plant death (Memoona et al., 2023). Therefore, these air pollutants must be removed in order to provide the plants the best environment to grow.

7.2.1 Air Quality Control Parameters

Here are some control parameters to consider:

1. **CO_2 levels:** Ideally 800–1200 ppm for many crops
2. **Pollutants:** Such as NH_3, CO, or general VOCs
3. **Ventilation Needs:** Varies with weather and occupancy

7.2.2 Sensors & Actuators

Sensors: Use an air quality sensor such as the MQ-135 gas sensor, which can detect CO_2, NH_3, benzene, smoke and other pollutants. The MQ-135 outputs an analogue voltage proportional to gas concentration after calibration. It is inexpensive and suitable for monitoring overall air quality (Agarwal, 2022).

Actuators: Key actuators include a ventilation fan or vent opening and a CO_2 injection system. For example, a solenoid valve on a CO_2 tank. A vent fan helps to purge polluted or excess CO_2 air. Meanwhile, a CO_2 injector enriches air when CO_2 is too low. A buzzer or alarm could also be used to alert when harmful gas levels are detected.

7.2.3 Control Logic and Operation

The Arduino reads the MQ-135 and CO_2 sensor and checks against safe ranges. For example, if the MQ-135 analogue value indicates gas concentration above a threshold, the ventilation fan will be turned on to exhaust air. Conversely, many crops benefit from elevated CO_2, typically 800–1200 ppm. So if measured CO_2 falls below a lower setpoint (< 800 ppm), it will activate the CO_2 injection valve until the setpoint is reached. If CO_2 is excessively high (>1200 ppm), the fan or vent will be switched on to reduce until ideal level (DFRobot, 2023). This maintains healthy air for plants.

7.2.4 Pseudocode

Example Scenario:

If the MQ135 sensor detects excess harmful gas or low CO_2, the fan will turn on and CO_2 is injected until values return to optimal range.

```
int gasLevel = analogRead(MQ135_PIN);
float co2ppm = getCO2ppm();

if (gasLevel > 300) {
  digitalWrite(FAN_PIN, HIGH);   // Turn on fan
} else {
  digitalWrite(FAN_PIN, LOW);    // Turn off fan
}

if (co2ppm < 800) {
  digitalWrite(CO2_VALVE, HIGH); // Inject CO2
} else if (co2ppm > 1200) {
  digitalWrite(CO2_VALVE, LOW);  // Stop injection
}
```

Figure 12 - Pseudocode for Air Quality Control

7.3 pH Regulation Control

Soil pH plays a crucial role in determining nutrient solubility and root absorption efficiency. Maintaining appropriate pH levels is essential, as extreme values can cause nutrient precipitation and limit availability, thereby significantly reducing plant productivity (Nutricontrol, 2021).

7.3.1 pH Regulation Control Parameters

Here are some pH parameters to consider:

1. **Optimal pH Range:** Generally pH 5.5–7.0 for most crops
2. **Soil Composition:** Different soils buffer pH differently
3. **Fertilizer Use:** Can acidify or alkalize soil

7.3.2 Sensors & Actuators

Sensors: A soil pH probe, such as the LSPH01 analogue pH sensor, is used to measure soil acidity. This sensor outputs a voltage signal that corresponds to the pH level after calibration. It can be connected directly to an Arduino's analogue input pin (Bohrer, 2020).

Actuators: To adjust soil pH, peristaltic pumps or solenoid valves are used to dispense acidic or alkaline solutions into the soil or water system. For example, one pump draws from an acid reservoir, and another from a base (alkaline) reservoir. These pumps are controlled via digital pins on the Arduino, using relay modules or MOSFET drivers (Bohrer, 2020).

7.3.3 Control Logic and Operation

The Arduino continuously reads the soil pH and compares it against a predefined range. Most vegetables prefer slightly acidic soil, typically within the pH range of 6.0 to 7.0 (Liu & Hanlon, 2024). Therefore, a target pH of 6.5 is set, with a tolerance range of ± 0.2 to 0.5. If the measured pH falls below the lower threshold (target – tolerance), the soil is considered too acidic, and the system activates the alkaline pump to increase the pH level. Conversely, if the pH exceeds the upper threshold (target + tolerance), the acid pump is activated to lower the pH. When the pH is within the acceptable range, all pumps are deactivated (Bohrer, 2020).

7.3.4 Pseudocode

Example Scenario: If the pH drops to 5.8, the system activates the alkaline pump to bring it back to 6.5.

```
float pH = readPH();  // Get pH value

if (pH < 6.2) {
  digitalWrite(BASE_PUMP, HIGH);
  digitalWrite(ACID_PUMP, LOW);
} else if (pH > 6.8) {
  digitalWrite(ACID_PUMP, HIGH);
  digitalWrite(BASE_PUMP, LOW);
} else {
  digitalWrite(BASE_PUMP, LOW);
  digitalWrite(ACID_PUMP, LOW);
}
```

Figure 13 - Pseudocode for pH Regulation Control

7.4 Pest Control

Pests pose serious threats to agriculture and ecosystems by damaging crops, disrupting waterways, harming wildlife habitats, and causing disease and economic losses (National Institute of Food and Agriculture, 2025). In greenhouse environments, early detection and rapid response are essential to minimize losses while maintaining healthy growing conditions.

7.4.1 Pest Control Parameters

Here are some parameters to consider:

1. **Movement:** Detected by PIR or ultrasonic sensors
2. **Day/Night Conditions:** Pests may be more active during nighttime
3. **User Alert Preferences:** Alerts may be sent via buzzer or mobile app

7.4.2 Sensors & Actuators

Sensors: A PIR (Passive Infrared) motion sensor such as HC-SR501, detects infrared radiation from animals or humans, making it suitable for pest detection. Ultrasonic or beam break sensors can be used for detecting larger pests. A light sensor (LDR) can be used to limit detection to nighttime conditions, improving control accuracy (Prasetyo et al., 2024).

Actuators: When motion is detected, the system activates a buzzer to deter pests. Simultaneously, a notification is sent to the user's mobile app via Wi-Fi, enabling real-time alerts and remote monitoring (Prasetyo et al., 2024).

7.4.3 Control Logic and Operation

The pest detection system in GrowSync operates by continuously monitoring input from a PIR (Passive Infrared) motion sensor, which is strategically placed around the crop area. When the PIR sensor detects movement, it is interpreted as a potential pest intrusion. Upon detecting motion, the system immediately triggers deterrent mechanisms, which may include activating a buzzer, flashing LED lights, or an ultrasonic pest repellent to scare off the intruder without harming the crops (Prasetyo et al., 2024).

7.4.4 Pseudocode

Example Scenario: If motion is detected at night, the buzzer is triggered for 10 seconds and the user receives a mobile notification.

```
if (digitalRead(PIR_PIN) == HIGH) {
  digitalWrite(BUZZER_PIN, HIGH);
  digitalWrite(LIGHT_PIN, HIGH);
  sendAlert("Motion Detected in Greenhouse");
  delay(10000);  // Deterrent duration
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(LIGHT_PIN, LOW);
}
```

Figure 14 - Pseudocode for Pest Control using Motion Detection

8.0 Data Processing and Analysis

In the GrowSync system, data collected from sensors is transmitted from the Arduino microcontroller to a cloud platform for processing. This centralized cloud processing layer plays a crucial role in converting raw sensor inputs into meaningful insights. This section outlines the processes involved in data filtering, aggregation, visualization, and the application of machine learning to support automated control and informed decision-making. The figure below illustrates the data flow architecture of the GrowSync system.

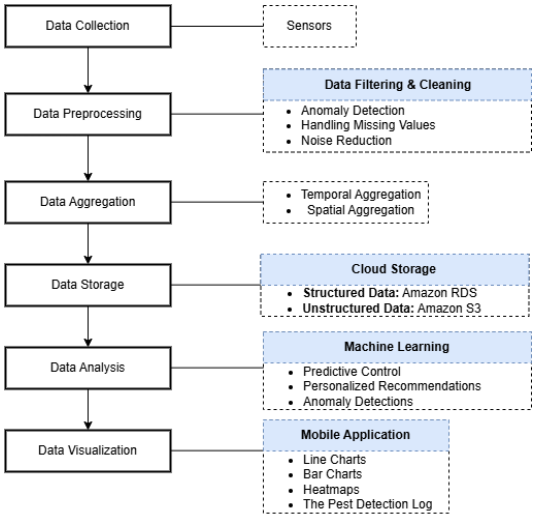

Figure 15 - Flowchart for Data Processing & Analysis

8.1 Data Filtering & Cleaning

Sensor data collected from a greenhouse environment can be highly variable due to environmental noise, sensor drift, and transmission errors. To ensure the data is suitable for analysis, a three-step data cleaning process is implemented:

8.1.1 Anomaly Detection

First, anomaly detection techniques are applied to flag impossible or incorrect readings (Wei et al., 2025). For example, if the temperature sensor (DHT22) suddenly reports 75°C in a greenhouse where average values range between 22–30°C, such a reading is marked as an outlier. A range-based rule or statistical method like Z-score can identify such anomalies. Similarly, if the soil pH sensor returns a value of 1.2, which is extremely acidic and biologically improbable, the system will automatically discard or replace with average value.

8.1.2 Missing Data

Missing data may occur due to network loss or temporary sensor failure. In these cases, the system fills in the gaps using methods such as last-known-value substitution or linear interpolation (Mahajan, 2024). For instance, if no humidity data is received for five minutes, the last available value is temporarily used for real-time visualization, while interpolated values are applied for historical trend analysis.

8.1.3 Noise Reduction

Additionally, noise reduction is particularly important for sensitive sensors, such as the analogue pH probe and the MQ135 air quality sensor. To minimize fluctuations, a moving average filter is applied (Poonia et al., 2025). For example, if the pH sensor returns inconsistent readings such as 6.4, 6.7, 6.2, 6.8 within a short time frame, the system calculates a moving average over a rolling window (5–10 samples) to stabilize the input before executing actuator commands.

8.2 Data Aggregation

Once the data has been cleaned, the next step is aggregation to extract meaningful trends and support efficient long-term analysis. Aggregation is performed in two primary forms: temporal aggregation and spatial aggregation.

8.2.1 Temporal Aggregation

In temporal aggregation, sensor data is summarized over fixed time intervals (Rajaguru et al., 2025). For real-time monitoring, readings are averaged every 15 minutes. This allows the mobile app to display current trends without overwhelming the user with minute-level fluctuations. For example, daily summaries of temperature, humidity, and soil pH allow farmers to evaluate if environmental conditions remained within optimal ranges for crop health. If the system records that soil moisture consistently drops below 50% during midday, the farmer may decide to adjust irrigation schedules accordingly.

8.2.2 Spatial Aggregation

Spatial aggregation involves combining readings from multiple sensors deployed in different zones of the greenhouse (Yoo et al., 2025). For example, three soil pH sensors placed in different corners of the greenhouse may report slightly different values (e.g., 6.3, 6.5, 6.4). The system calculates an average of these readings to determine whether to activate the acid or base pump for pH regulation across the entire zone. This approach balances local accuracy with practical decision-making for larger areas.

In addition, data aggregation supports insight generation. The system tracks daily water usage based on misting pump runtime and estimates how many litres were consumed. Similarly, energy consumption from ventilation fans is tracked, allowing weekly reports to show the total usage and efficiency. Historical trends, such as rising internal temperature patterns during summer, help the system recommend long-term improvements like additional shading or cooling units.

8.3 Data Visualization

After cloud-based processing, the system presents the information through a mobile app dashboard, providing both real-time monitoring and access to historical insights. The live dashboard displays key readings such as greenhouse temperature, humidity, CO_2 levels, and actuator status (e.g., "Fan: ON", "Misting: OFF") for quick system assessment (Wilberforce & Mwebaze, 2025).

Users are provided with timely and reliable data to support well-informed decision-making, as all visualizations are updated in real time and securely stored in the cloud for universal accessibility. To further facilitate efficient analysis and decision-making, historical data is systematically archived and presented through various graph types:

1. **Line charts** are used to monitor trends in temperature, humidity, and pH levels over time, enabling users to identify gradual changes that may indicate equipment issues like fan underperformance.
2. **Bar charts** allow for comparison of resource consumption, such as weekly water or CO_2 usage, providing insights into operational efficiency.
3. **Heatmaps** offer a spatial representation of environmental variables, such as soil pH, helping to pinpoint areas of concern.

8.4 Machine Learning

To enhance system intelligence and adaptability, our system uses machine learning (ML) models for predictive control, personalized recommendations and anomaly detections.

8.4.1 Predictive Control

For predictive control, machine learning models are trained on historical temperature data and weather forecasts to anticipate temperature spikes and activate fans before thresholds are crossed (Abou-Mehdi-Hassani et al., 2025). This helps to enhance environmental stability and reduce crop stress. Similarly, irrigation models analyze soil moisture patterns, crop type, and time of day to initiate misting precisely when needed, thereby preventing both under and overwatering.

8.4.2 Personalized Recommendations

Machine Learning also enables personalized recommendations based on user behaviour and crop history (Abou-Mehdi-Hassani et al., 2025). For instance, if a user consistently maintains CO_2 levels at 900 ppm and observed improved crop growth, the system may recommend this value as a standard for future operations. For pest control, if motion is frequently detected around 7 PM, the system can recommend scheduling deterrents measures in advance.

8.4.3 Anomaly Detections

Additionally, machine learning eases anomaly detection by identifying unusual sensor behaviour (Abou-Mehdi-Hassani et al., 2025). For example, if the misting system is active but the humidity level remains unchanged, this may indicate a blocked nozzle or system malfunction. In such cases, the system issues an alert to the user: “Warning: Misting system active but humidity unchanged. Possible malfunction.”

9.0 Mobile App Functionality

Our mobile application is the core user interface that connects farmers to their smart agriculture system. Developed using Flutter for cross-platform support, the app interacts seamlessly with AWS IoT Core through MQTT (Singh, 2024). Designed with simplicity and usability, the app enables farmers in remote areas with limited connectivity to efficiently monitor and manage their farms. The figure below presents a mock-up of the mobile app interface.

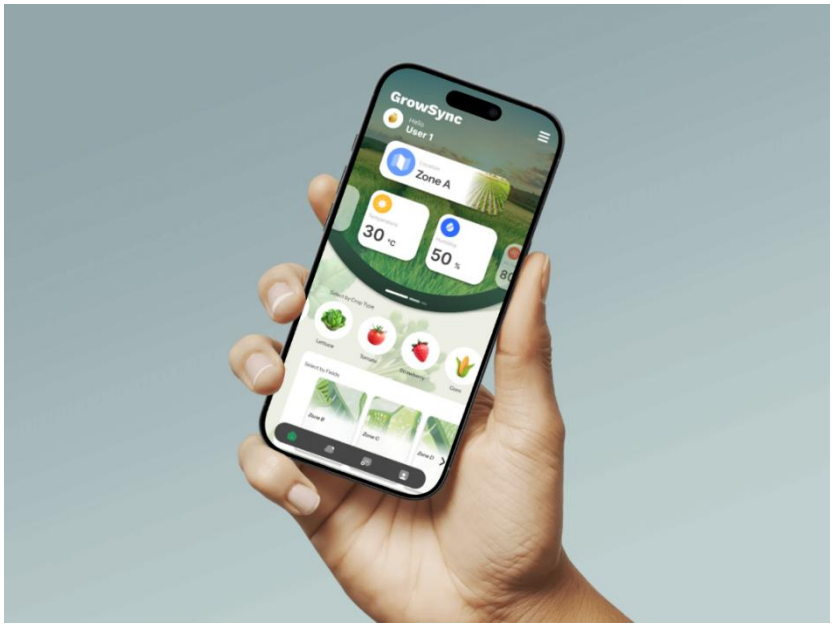

Figure 16 - GrowSync Mock-up

9.1 Real-Time Monitoring

The mobile app provides a live dashboard that displays real-time environmental conditions and actuator status. Real-time data is streamed from AWS Timestream, ensuring minimal latency between sensor input and user interface (flareAI Services, 2025). It will visualize key parameters such as Temperature & Humidity, Air Quality, pH Level and motion alerts.

For example, if the air quality drops below the threshold, users will see a red warning icon on the dashboard next to the “CO_2 Levels” indicator. Similarly, actuators such as misting systems, ventilation fans, and pH regulation pumps show their current status ("Active", "Standby", "Offline") in real time. This feature helps farmers to notice and respond quickly to environmental changes without being physically present.

9.2 Remote Control & Scheduling

The app enables users to override automatic controls, manually manage and schedule actuators from anywhere (Huynh et al., 2023). For example:

1. **Irrigation System:** Users can manually start or stop irrigation, adjust water flow, or activate specific zones based on observed moisture levels.
2. **Environmental Controls:** Farmers can increase ventilation fan speed if the temperature of greenhouses is high or manually activate misting during dry weather.
3. **CO_2 Control:** If CO_2 levels are low, the user can activate the valve directly from the app to improve photosynthesis conditions.
4. **Pest Control:** Deterrents such as lights or buzzers can be turned on if pest movement is detected.

In addition to remote controlling, the app also provides users the option to schedule automated tasks. For example, users can schedule an irrigation schedule to water Zone A every morning at 6 AM for 10 minutes.

9.3 Notifications & Alerts

The GrowSync system also provides real-time alerts based on sensor thresholds or system anomalies (Wilberforce & Mwebaze, 2025). Types of alerts include:

1. **Environmental Warnings:** “Temperature exceeds 35°C” or “Soil moisture below critical threshold.”
2. **System Malfunctions:** “Fan not responding,” “Sensor disconnected,” or “Misting system active but no humidity change detected.”
3. **Pest Activity:** “Motion detected near tomato greenhouse at 7:10 PM.”

Furthermore, the app generates summary reports at the end of each day or week, highlighting key environmental conditions, actuator activities, and resource usage, enabling farmers to reflect on system performance and plan more effectively.

9.4 Personalization

The mobile app includes a range of personalization features that allow users to modify the system to specific crop types, farm zones, and individual preferences (Wilberforce & Mwebaze, 2025). For example:

1. **Crop Profiles:** Users can create and manage profiles for different crops, for instance, define temperature, humidity, and pH targets for crops like lettuce, chili, or strawberries.
2. **Custom Dashboard Layout:** Farmers can rearrange widgets, for example, placing pH levels and nutrient tank status at the top if those are high priority.
3. **Multi-user Access & Permissions:** Farm owners can grant access to field workers or agronomists with different roles such as view-only, control, or admin.

10.0 Security Considerations

Ensuring secure security is fundamental to the success and reliability of the GrowSync system. GrowSync operates across multiple layers, including edge devices, cloud platforms, mobile applications, and communication protocols. Each of which is a potential target for security threats if not properly protected. This section outlines the key security mechanisms implemented to protect sensor data, actuator commands, user identities, and the overall integrity of the system.

10.1 Device Authentication

Device authentication is crucial to ensuring that only trusted and authorized sensors, actuators, and gateways can establish a connection with the system. Each IoT device is provisioned with a unique **X.509 digital certificate**, serving as a digital identity used during secure communication. These certificates are verified through Public Key Infrastructure (PKI). The system also implements **Mutual TLS (mTLS)**, allowing both the device and the server to authenticate each other before any data is exchanged (AWS, 2025).

For example, when a Raspberry Pi edge gateway receives temperature data from a DHT22 sensor, both devices must present valid certificates before the data is forwarded to AWS IoT Core. This prevents unauthorized devices from infiltrating the network. To further strengthen device integrity, all edge devices use secure boot and hardware-base roots of trust, verifying firmware authenticity at startup and safeguarding cryptographic keys within tamper-resistant modules.

10.2 Data Encryption

To prevent data interception and manipulation, our platform secures all data in both transit and at rest using industry-standard encryption techniques.

10.2.1 Encryption During Transmission

All communications between sensors, actuators, edge devices, cloud services, and mobile apps are encrypted using **Transport Layer Security (TLS 1.2 or higher)** (Manohar, 2025). Communication between field devices and AWS IoT Core utilizes **MQTT over TLS**, while the mobile application uses HTTPS to access cloud-hosted dashboards and APIs. This ensures that

environmental readings such as pH levels or CO_2 concentrations cannot be intercepted or altered during transmission (Manohar, 2025). In high-security farm environments, Virtual Private Networks (VPNs) may be employed to create additional encrypted tunnels across network segments.

10.2.2 Encryption in Storage

Data stored either locally or in the cloud is encrypted to protect its confidentiality and integrity. On the cloud side, our platform stores time-series data in AWS Timestream and object data in S3 buckets, both protected using **AES-256 encryption** (Manohar, 2025). This includes system environmental logs, actuator status history, and user profiles. On edge devices like Raspberry Pi, any temporarily cached data such as soil moisture readings or misting schedules is encrypted using the same AES standard. This ensures that even if a device is physically compromised, the data remains secure.

10.3 Access Control

GrowSync implements **Role-Based Access Control (RBAC)** to restrict access to different parts of the system based on user roles. This structure ensures that users only see or control components relevant to their responsibilities (Manohar, 2025). For example,

1. A Farm Owner has full access to all devices, schedules, and analytics dashboards.
2. A Farm Manager can configure automation for specific zones,
3. Farm Workers are granted access to view sensor data and operate assigned actuators only.

External consultants may be granted read-only access for analytical purposes. All user logins on both mobile and web platforms are protected using **Multi-Factor Authentication (MFA)**, which requires a password and a one-time verification code generated via SMS or an authenticator app. Strong password policies enforce complex, frequently updated credentials, while secure API access is managed through OAuth 2.0 tokens, API keys, and rate-limiting to prevent misuse by third-party integrations (Manohar, 2025).

10.4 Firmware Updates

Firmware updates for GrowSync's IoT devices are delivered using secure **Over-the-Air (OTA)** update mechanisms (Manohar, 2025). All firmware images are cryptographically signed by GrowSync, and each device verifies the digital signature before installation to ensure authenticity. These updates are encrypted during transmission to protect against tampering or interception. To maintain system stability, the update process is atomic, meaning that if an update fails mid-process, the device automatically reverts to the previous stable version. A secure bootloader ensures that only signed firmware can be executed on the device (Manohar, 2025).

For example, if a misting actuator receives a firmware patch to improve humidity calibration, it will only install it after validating the signature and ensuring the image is complete and untampered. Firmware version management is centralized, enabling administrators to track deployment status across all farm devices and ensure timely patching of known vulnerabilities.

11.0 References

Abou-Mehdi-Hassani, F., Zaguia, A., Bouh, H. A., & Mkhida, A. (2025). Machine Learning-Driven Greenhouse Management: A comparative analysis of prediction models. *2025 5th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, 1–7. https://doi.org/10.1109/iraset64571.2025.11008061

AdruinoGetStarted. (2025, June 8). *Arduino - cooling system using DHT sensor: Arduino Tutorial*. Arduino Getting Started. https://arduinogetstarted.com/tutorials/arduino-cooling-system-using-dht-sensor#:~:text=float%20temperature%3B%20%2F%2F%20temperature%20in,The%20fan%20is%20turned%20off.

Agarwal, T. (2022, June 10). *MQ135 Air Quality Sensor Datasheet : Working & Its Applications*. ElProCus. https://www.elprocus.com/mq135-air-quality-sensor/#:~:text=It%20is%20a%20semiconductor%20air,gas%20detection%20and%20monitoring%20applications.

Arduino. (2015). *Arduino Mega 2560 Rev3 Datasheet (A000067)*. [PDF]. https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf

Arshad, J., Aziz, M., Al-Huqail, A. A., Muhammad, Muhammad Husnain, Rehman, A. U., & Shafiq, M. (2022). Implementation of a LoRaWAN Based Smart Agriculture Decision Support System for Optimum Crop Yield. *Sustainability*, *14*(2), 827–827. https://doi.org/10.3390/su14020827

Avşar, E., & Mowla, Md. N. (2022). Wireless communication protocols in smart agriculture: A review on applications, challenges and future trends. *Ad Hoc Networks*, *136*, 102982. https://doi.org/10.1016/j.adhoc.2022.102982

AWS. (2025). *Security best practices in AWS IOT Core - AWS IOT Core*. https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html

AWS. (2025). *Guidance for Log Storage on AWS*. Amazon Web Services, Inc. https://aws.amazon.com/solutions/guidance/log-storage-on-aws/

Bohrer, J. (2020, November 4). Medium. https://jjbskir.medium.com/arduino-ph-controller-dfcfb749c7f0.

Devanand, W. A., Raghunath, R. D., Baliram, A. S., & Kazi , K. S. (2020, January 12). *Smart Agriculture System Using IoT*. Academia.edu. https://www.academia.edu/download/61762899/IJIRT147761_PAPER20200112-36455-1dz4prw.pdf

DFRobot. (2023, May 26). *CO2 concentration monitoring in Greenhouses*. https://www.dfrobot.com/blog-13282.html?srsltid=AfmBOopjakC8IZBDqJBFEVyjhrE3jH9ACO2tTrj2MFchF0_sfTJmMYYP.

Dizdarević, J., Carpio, F., Jukan, A., & Masip-Bruin, X. (2019). A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration. *ACM Computing Surveys*, *51*(6), 1–29. https://doi.org/10.1145/3292674

Dragino Technology Co., Ltd. (2025). *Datasheet-LSPH01-LoRaWAN Soil pH Sensor* [PDF]. https://www.dropbox.com/sh/jqebyb0l23w7rpt/AAC9bQKKQcFeRsQywSFrqOLja?dl=0

Espressif Systems. (2020). *ESP32 Series Datasheet Version 4.9*. [PDF]. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Filip, I.-D., Iliescu, C., & Pop, F. (2022). Assertive, Selective, Scalable IoT-Based Warning System. *Sensors*, *22*(3), 1015–1015. https://doi.org/10.3390/s22031015

flareAI Services. (2025, February 17). *Mobile app development for IOT: Best practices*. StudioLabs. https://www.studiolabs.com/mobile-app-development-for-iot-best-practices/

HC-SR501 Datasheet. (2013). *Passive Infrared Motion Detector Sensor Module*. [PDF]. https://www.mpja.com/download/31227sc.pdf

Hsu, T.-C. (2024). Designing a Secure and Scalable Service Agent for IoT Transmission through Blockchain and MQTT Fusion. *Applied Sciences*, *14*(7), 2975–2975. https://doi.org/10.3390/app14072975

Huynh, H. X., Tran, L. N., & Duong-Trung, N. (2023). Smart Greenhouse Construction and irrigation control system for optimal brassica juncea development. *PLOS ONE*, *18*(10). https://doi.org/10.1371/journal.pone.0292971

Jawad, H., Nordin, R., Gharghan, S., Jawad, A., & Ismail, M. (2017). Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review. *Sensors*, *17*(8), 1781. https://doi.org/10.3390/s17081781

Kumar Sinha, G. (2024). Sensor Data Analytics for Optimized Methane Leak Detection and Mitigation. *International Journal of Science and Research (IJSR)*, *13*(4), 223–231. https://doi.org/10.21275/sr24330010938

Logicsun. (2024, March 4). *Temperature and humidity management in agriculture.* Nido Pro. https://www.nidopro.com/temperature-and-humidity-management-agriculture/

LSPH01 -- LoRaWAN Soil pH Sensor. (2024). Dragino.com. https://www.dragino.com/products/agriculture-weather-station/item/184-lsph01.html

Mahajan, I. (2024, September 18). *Missing value imputation, explained*. Data Pragmatist. https://datapragmatist.com/p/missing-value-imputation-explained

Manohar, D. (2025, May 20). *IOT security: Comprehensive guide to safeguarding connected devices*. cavliwireless.com. https://www.cavliwireless.com/blog/nerdiest-of-things/iot-security-challenges-solutions

Manzanero-Vazquez, D. J., Manrique-Ek, J. A., Cardozo-Aguilar, G., & Decena-Chan, C. A. (2021). Internet of things applied to agriculture using the ESP32 module in connection with the Ubidots platform. *Journal of Technological Prototypes*, *7*(No.20 12-20), 12–20. https://doi.org/10.35429/jtp.2021.20.7.12.20

Memoona, Faazal, B., Qasim, M., Mumtaz, S., Iftikhar, M., Khalid, I., Muzaffar, M. J., Nisar, H., & Adrees, M. (2023). Crop quality and quantity as influenced by important air pollutants in Pakistan. *Advances in Botanical Research*, 109–144. https://doi.org/10.1016/bs.abr.2023.03.002

Mouser Electronics. (2018). *Piezoelectric Sounder PS1240P02BT Datasheet*. Mouser Electronics. [PDF]. https://www.mouser.com/datasheet/2/400/ef532_ps-

13444.pdfhttps://www.mouser.com/datasheet/2/400/ef532_ps-13444.pdf?srsltid=AfmBOoqLO3iSlCHqyetoiHgd2LHYDq8nNJJ3MK1gPEu19iNNrnVbV68S

National Institute of Food and Agriculture. (2025, June 30). *Pest management*. Nation Institute of Food and Agriculture. https://www.nifa.usda.gov/topics/pest-management#:~:text=Importance%20of%20Pest%20Management,life%20impacts%20in%20populated%20areas.

Navarro, E., Costa, N., & Pereira, A. (2020). A Systematic Review of IoT Solutions for Smart Farming. *Sensors*, *20*(15), 4231. https://doi.org/10.3390/s20154231

NCSC. (2021, July 21). *Using TLS to protect data*. Www.ncsc.gov.uk. https://www.ncsc.gov.uk/guidance/using-tls-to-protect-data

NEO-6 series. (2015, June 25). u-Blox. https://www.u-blox.com/en/product/neo-6-series

Nutricontrol. (2021, June 22). *The importance of ph control in crops*. https://nutricontrol.com/en/the-importance-of-ph-control-in-crops/.

Ojo, M. O., Viola, I., Baratta, M., & Giordano, S. (2021). Practical Experiences of a Smart Livestock Location Monitoring System Leveraging GNSS, LoRaWAN and Cloud Services. *Sensors*, *22*(1), 273. https://doi.org/10.3390/s22010273

Olimex. (2021). MQ135 Gas Sensor Module Datasheet. [PDF]. *https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf*

Pereira, G. P., Chaari, M. Z., & Fawwad Daroge. (2023). IoT-Enabled Smart Drip Irrigation System Using ESP32. *IoT*, *4*(3), 221–243. https://doi.org/10.3390/iot4030012

Phithak Thaenkaew, Quoitin, B., & Meddahi, A. (2023). Leveraging Larger AES Keys in LoRaWAN: A Practical Evaluation of Energy and Time Costs. *Sensors*, *23*(22), 9172–9172. https://doi.org/10.3390/s23229172

Poonia, A., Nath, P. R., Giriraj, Y. S., & Onkar, D. (2025b, June 17). *IOT-based adaptive farming system using acoustic sensing for Smart Agriculture*. International Journal of Engineering

Research & Technology. https://www.ijert.org/iot-based-adaptive-farming-system-using-acoustic-sensing-for-smart-agriculture

Prasetyo, N. A., Syamsu, M., & Arman, S. A. (2024). Design of pest detection and repellent device based on Arduino Uno using PIR (Passive Infrared Receiver) sensor. *Journal of Computer Science Advancements*, *2*(5), 285–296. https://doi.org/10.70177/jsca.v2i5.1326

Prodanović, R., Rančić, D., Vulić, I., Zorić, N., Bogićević, D., Ostojić, G., Sarang, S., & Stankovski, S. (2020). Wireless Sensor Network in Agriculture: Model of Cyber Security. *Sensors*, *20*(23), 6747. https://doi.org/10.3390/s20236747

Raheja, N., & Kumar Manocha, A. (2022). IoT based ECG monitoring system with encryption and authentication in secure data transmission for clinical health care approach. *Biomedical Signal Processing and Control*, *74*, 103481. https://doi.org/10.1016/j.bspc.2022.103481

Rajaguru, G., Lim, S., & O'Neill, M. (2025). A review of temporal aggregation and systematic sampling on time-series analysis. *Journal of Accounting Literature*, *47*(5), 110–128. https://doi.org/10.1108/jal-09-2024-0237

Rajesh, T., Thrinayana, Y., & Srinivasulu, D. (2020). IOT BASED SMART AGRICULTURE MONITORING SYSTEM. In *International Research Journal of Engineering and Technology*. https://www.irjet.net/archives/V7/i3/IRJET-V7I3346.pdf

Rajesh, T., Thrinayana, Y., & Srinivasulu, D. (2020). IOT BASED SMART AGRICULTURE MONITORING SYSTEM. In *International Research Journal of Engineering and Technology*. https://www.irjet.net/archives/V7/i3/IRJET-V7I3346.pdf

Rashmi R, Hrishikesh Prahalad, H Srujan Kumar, & Anuj Hoslok. (2023). *Exploring Digital Twins for Plant Growth Monitoring*. https://doi.org/10.1109/csitss60515.2023.10334087

RichardP4. (2017, October 13). *Arduino Greenhouse Control - humidity and temperature*. Instructables. https://www.instructables.com/Arduino-Greenhouse-Control-Humidity-and-Temperatur/

Shin, Y., & Jeon, S. (2024). MQTree: Secure OTA Protocol Using MQTT and MerkleTree. *Sensors*, *24*(5), 1447. https://doi.org/10.3390/s24051447

Singh, K. (2024, June 27). *Building an MQTT app using FlutterFlow to interface with IOT devices*. Medium. https://medium.com/@khajan_singh/building-an-mqtt-app-using-flutterflow-to-interface-with-iot-devices-8bf656d755d1

SparkFun Electronics. (2018). *DHT22 Temperature-Humidity Sensor Datasheet*. [PDF]. https://cdn.sparkfun.com/assets/f/7/d/9/c/DHT22.pdf

Suriano, D. (2024). Using Low-Cost Gas Sensors in Agriculture: A Case Study. *Using Low-Cost Gas Sensors in Agriculture: A Case Study*, 74–74. https://doi.org/10.3390/ecsa-11-20503

Tayari, E., Jamshid, A., & Goodarzi, H. (2015). Role of GPS and GIS in precision agriculture. *Journal of Scientific Research and Development*, *2*(3), 157–162. https://www.researchgate.net/profile/Nirmala-Svsg/post/Any_peer-reviewed_research_papers_articles_related_to_use_of_Geographic_Information_System_in_agronomy_agriculture/attachment/59d649b679197b80779a4275/AS%3A472255508553728%401489605895083/download/Role+of+GPS+and+GIS+in+precision+agriculture.pdf

u-blox. (2015). *NEO-6 u-blox 6 GPS Modules Data Sheet*. [PDF]. https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf

Wei, C., Shan, Y., & Zhen, M. (2025). Deep learning-based anomaly detection for precision field crop protection. *Frontiers in Plant Science*, *16*. https://doi.org/10.3389/fpls.2025.1576756

Wilberforce, N., & Mwebaze, J. (2025, January 17). *A framework for IOT-enabled Smart Agriculture*. arXiv.org. https://arxiv.org/abs/2501.17875

Winsen Electronics Technology. (2022). Semiconductor Gas Sensor MQ135 Datasheet (Version 1.4). [PDF]. *https://www.winsen-sensor.com/d/files/PDF/Semiconductor%20Gas%20Sensor/MQ135%20(Ver1.4)%20-%20Manual.pdf*

Yoo, B. H., Kim, K. S., Kim, J., Beresford, R. M., & Fleisher, D. H. (2025). The development of a soil data aggregation method for the spatial impact assessment of climate change on soybean yield. *Computers and Electronics in Agriculture*, *235*, 110347. https://doi.org/10.1016/j.compag.2025.110347