Department of Computer Engineering

CENG350 Software Engineering

Software Requirements Specification for FarmBot

Group 7

By

Burak Yildiz 2449049

Ihsan Seha Polat 2443729

Saturday 6th April, 2024

Contents

List of Figures

List of Tables

Revision History

Revision	Date	Description
0.0.1	03.03.2024	Initialization
1.0.0	29.03.2024	SRS First Draft finalized
2.0.0	06.04.2024	SRS finalized

Table 1: Revision History

1. Introduction

1.1 Purpose of the System

The FarmBot system is designed to automate and optimize small-scale agricultural tasks, embodying the principles of precision farming and sustainability. Through its open-source CNC farming machine, FarmBot aims to:

- Automate planting, watering, weeding, and monitoring of plant health to reduce labor and increase efficiency.
- Employ precision farming techniques to optimize resource use and enhance crop yields, promoting sustainability.
- Offer customizability and flexibility, enabling users to tailor the system to their specific agricultural needs.
- Serve as an educational tool, facilitating hands-on learning in STEM fields and sustainable agricultural practices.
- Foster a community-driven approach to innovation in agriculture, encouraging collaboration and continuous improvement.

This system's purpose aligns with advancing sustainable, efficient, and accessible agriculture through technological innovation.

1.2 Scope

The scope of the FarmBot software encompasses several key modules designed to automate and optimize agricultural operations on a small scale. These modules include:

- **Plant Scheduler**: Automates the planting process by scheduling tasks based on crop types and planting patterns.
- **Water Management System**: Optimizes water usage by adjusting watering schedules in response to soil moisture levels and weather forecasts.
- **Weed Detection and Control**: Employs image recognition to identify and manage weed growth with minimal human intervention.
- **Health Monitoring**: Utilizes sensor data to monitor plant health and soil conditions, providing actionable insights to users.
- **User Interface (UI)**: Offers a web-based interface for users to interact with FarmBot, customizing settings, monitoring operations, and receiving updates.

These software products facilitate the core functionalities of FarmBot, enabling efficient, automated farming operations. They are designed to:

- Reduce manual labor and operational complexity in gardening and small-scale farming.
- Enhance crop yields and sustainability by optimizing resource usage and environmental adaptation.
- Provide educational opportunities through hands-on engagement with agricultural technology.

The application of FarmBot's software is targeted at individual gardeners, educators, and small-scale farmers, offering significant benefits such as increased efficiency, sustainability, and accessibility to precision agriculture. The objectives and goals of

this project align with promoting sustainable farming practices, advancing agricultural education, and fostering a community of innovation in farming technology. This scope is consistent with the higher-level specifications outlined in the system requirements, focusing on the integration of technology into agriculture to meet modern demands for food production and environmental stewardship.

1.3 System Overview

1.3.1 System Perspective

FarmBot is a standalone, autonomous precision agriculture system that enhances small-scale gardening through automation. FarmBot epitomizes the forefront of autonomous precision agriculture technology, designed to transform small-scale gardening through automation. It leverages external data sources, such as weather forecasts and agricultural databases, to offer an all-encompassing garden management interface. The system combines precision-engineered hardware components with an intuitive web-based software platform and supports both Wi-Fi and Ethernet for robust internet connectivity. Capable of performing a wide range of operations from seeding to plant health monitoring, FarmBot's design is inherently adaptable, facilitated by a modular construction that allows for the integration of additional sensors and tools, enhancing its capabilities over time.

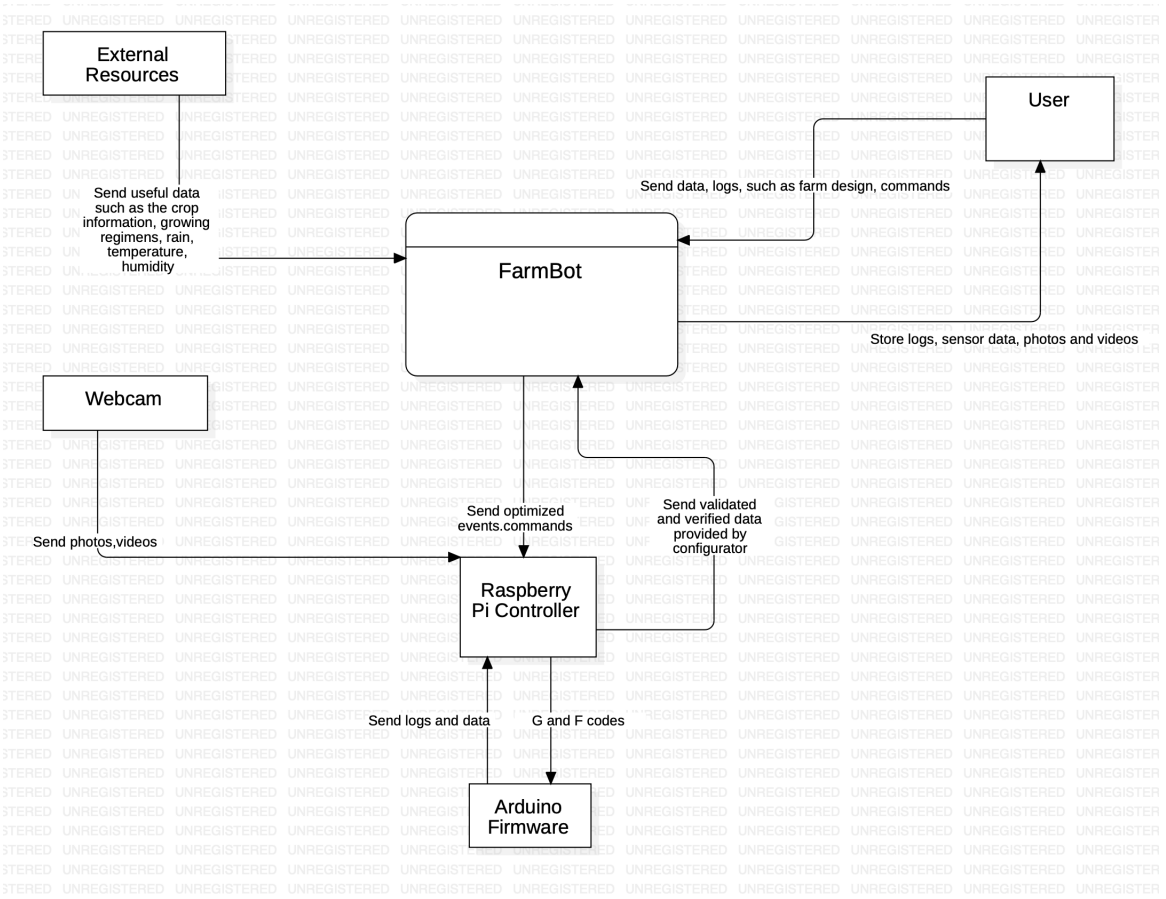

Figure 1.1: FarmBot Context Diagram

1.3.1.1 System Interfaces

FarmBot's system interfaces facilitate the seamless operation of hardware components and external data exchange, encompassing connections between the microcontroller and various sensors and actuators. These interfaces are governed by industry standards such as SPI and I2C for sensor data communication, ensuring reliable and efficient data transfer. Additionally, interfaces with external databases and cloud storage adhere to secure web standards, including HTTPS and RESTful APIs, to protect data integrity and privacy.

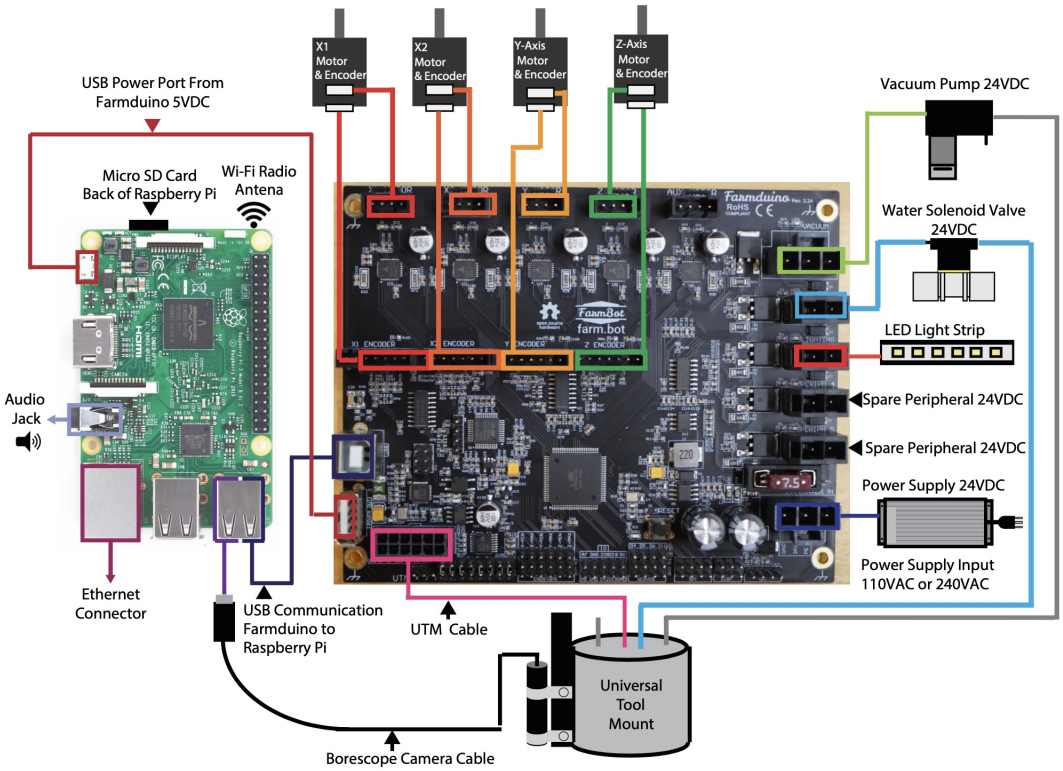

Figure 1.2: FarmBot System Diagram

1.3.1.2 User Interfaces

The user interfaces of FarmBot, including a web application and mobile app, are designed for ease of use and accessibility, enabling users to remotely control, monitor, and configure their FarmBot systems. These interfaces comply with web accessibility standards (WCAG) and best practices in UI/UX design, ensuring a seamless and inclusive user experience across different devices and platforms.

Figure 1.3: FarmBot Web App on Different Devices

1.3.1.3 Hardware Interfaces

FarmBot's hardware interfaces, which include connections to soil moisture sensors, the watering system, seed dispensers, and camera systems, adhere to electrical and communication standards to ensure safety and functionality.Their special Farmduino electronics board features a dedicated processor for monitoring the rotary encoders at high speed, while the Raspberry Pi 3 serves as the web-connected brain. The standards FarmBot uses here include IEEE 802 for wireless communication and USB standards for peripheral devices, ensuring compatibility and safe operation across the system's hardware components.

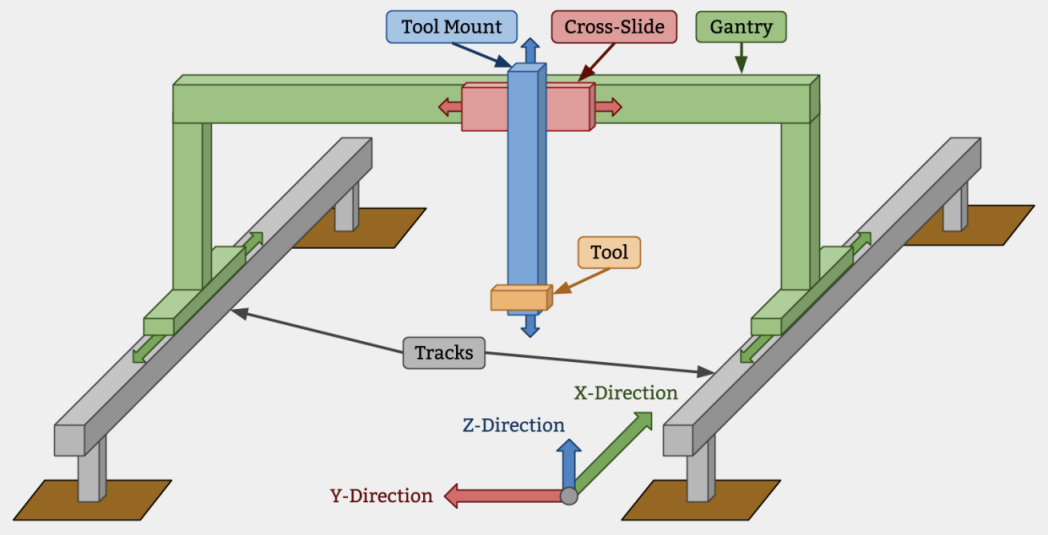

Figure 1.4: FarmBot High Level Hardware Overview and Coordinate System.

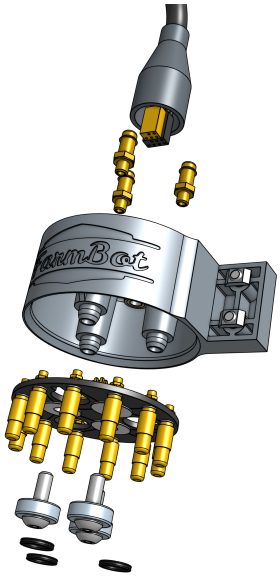

Figure 1.5: FarmBot Genesis Universal Tool Mount

1.3.1.4 Software Interfaces

The software infrastructure of FarmBot, based on a lightweight Linux operating system, supports various software interfaces for system operation and management. Compliance with open-source licensing and security standards, including the Secure Software

Development Framework (SSDF) by NIST, ensures that FarmBot's software remains secure, reliable, and maintainable.FarmBot's Raspberry Pi runs a custom operating system named FarmBot OS to maintain a connection and synchronize with the web application via the message broker. This allows FarmBot to download and execute scheduled events, be controlled in real-time, and upload logs and sensor data. The OS communicates with the Farmduino/Arduino over a USB cable or serial connection to send G and F code commands, and also receive collected data from sensors and rotary encoders.FarmBot OS has a built-in utility named configurator allowing you to easily enter WiFi and web app credentials from a WiFi enabled device (such as a laptop or smartphone). This is used for initial setup in order to get your FarmBot connected to your home WiFi and web app account. The firmware that is flashed onto the Arduino or Farmduino microcontroller is responsible for physically operating FarmBot's motors, tools, sensors, and other electronics. It receives G and F codes from FarmBot OS, and then moves the motors and reads and writes pins accordingly. It also sends collected data from the rotary encoders and pin reads back to the Raspberry Pi.[1]

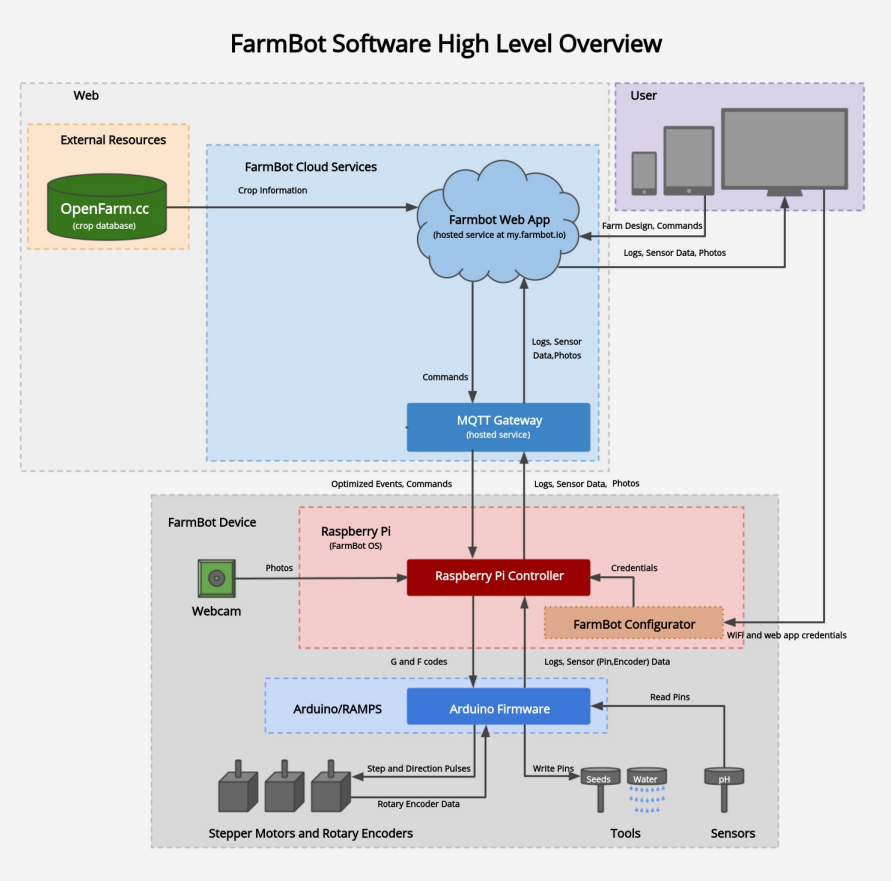

Figure 1.6: FarmBot Software High Level Overview

1.3.1.5 Communication Interfaces

Communication is the lifeblood of FarmBot, enabling not just the operation of the device itself but also its interaction with the broader digital world. FarmBot uses a configurator which is a piece of software built into FarmBot OS that makes it easy to connect your FarmBot to a WiFi network and user's web app account. FarmBot's communication interfaces, enabling Wi-Fi, Ethernet, and optional Bluetooth connectivity, are designed in accordance with IEEE standards, including IEEE 802.11 for Wi-Fi and IEEE 802.15.1 for Bluetooth. These standards ensure robust and secure data communication, allowing FarmBot to receive updates, synchronize data, and provide remote access functionalities efficiently.

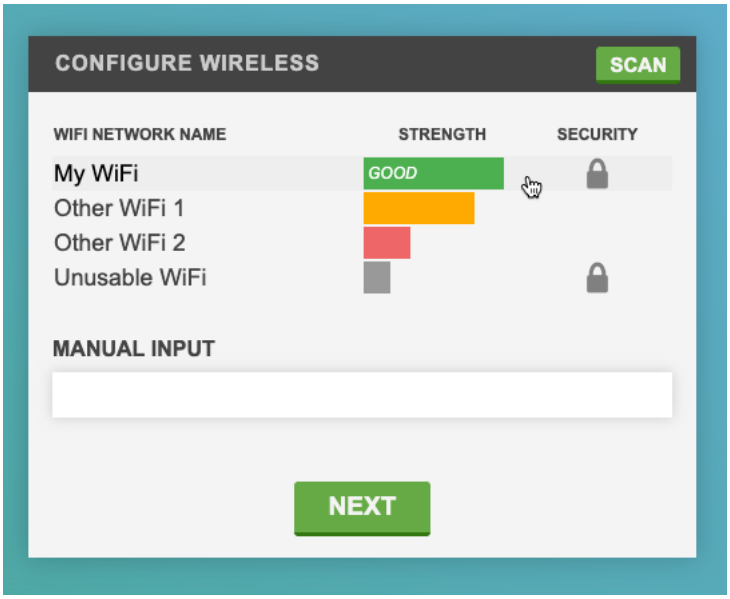

Figure 1.7: FarmBot Configurator in use for Wi-Fi Connection

1.3.1.6 Memory Constraints

Addressing memory constraints, FarmBot incorporates on-board and expandable memory solutions, following industry standards for flash memory and external storage, such as SD memory card standards.An external memory interface, such as an SD card slot, offers users the flexibility to expand storage capacity. This is particularly useful for logging detailed operational data and storing images captured for plant health monitoring, ensuring that memory limitations never compromise the system's performance or data retention capabilities.

Task	Transmitted	Received	Total
Log sent from FarmBot	200 bytes	200 bytes	400 bytes
Command sent to FarmBot	200 bytes	200 bytes	400 bytes
Partial sync	200 bytes	200 bytes	400 bytes
Full sync	30KB - 150KB	30KB - 150KB	60KB - 300KB
Image upload	100KB - 500KB	200 bytes	100KB - 500KB
Over-the-air update	1MB	60MB	61MB

Figure 1.8: Average data required by task

1.3.1.7 Operations

This section outlines the normal and special operations required by users of FarmBot, including modes of operation, interactive and unattended periods, data processing support functions, and backup and recovery operations.

Modes of Operations in the User Organization

User-Initiated Operations:

- **Garden Planning and Setup:** Users design their garden layout, specifying plant locations, types, and schedules using the web or mobile application.
- **Manual Control Mode:** Allows for direct control over FarmBot components for immediate tasks or testing purposes.
- **Scheduled Operations Mode:** Users schedule routine tasks such as watering, weeding, and fertilizing tailored to the garden's needs.

Special Modes:

- **Monitoring and Alerts Mode:** FarmBot monitors soil moisture, plant health, and growth, alerting users to any anomalies detected.
- **Learning Mode:** FarmBot suggests optimizations based on historical data and machine learning algorithms (if applicable).

Periods of Interactive and Unattended Operations

Interactive Operations:

- **Initial Setup and Configuration:** Users set up FarmBot and configure its initial settings interactively.
- **Regular Monitoring:** Users check on plant health and progress, adjusting operations as needed.

Unattended Operations:

- **Automated Gardening Tasks:** FarmBot performs scheduled tasks autonomously.
- **Night Operations:** Tasks like watering are performed at night to minimize evaporation, without user intervention.

Data Processing Support Functions

- **Soil and Plant Data Analysis:** FarmBot analyzes data from sensors to inform decisions on watering and plant health.
- **Weather Data Integration:** Adjusts operations based on anticipated weather conditions.
- **User Input Processing:** Tailors operations to specific garden needs based on user input.

Backup and Recovery Operations

Backup Operations:

- **Data Backup:** Periodic backup of plant growth data, system settings, and operation logs to the cloud.
- **Configuration Backup:** Backup of user settings and configurations for easy recovery or migration.

Recovery Operations:

- **System Restore:** Restores operations, settings, and schedules from backup after malfunctions or maintenance.
- **Disaster Recovery:** Enables system reinstallation and configuration recovery in case of severe issues.

These operational requirements ensure FarmBot is capable of meeting the diverse needs of precision agriculture, providing flexibility, user-friendliness, and resilience against disruptions.

1.3.2 System Functions

FarmBot is designed to automate the critical functions of precision agriculture. This includes initial soil preparation, seeding, daily monitoring of plant health through sensors, precise watering, fertilizing, and weeding. The seeding function utilizes a database of plant characteristics to optimize seed spacing and depth. Watering is dynamically adjusted by the system based on soil moisture content and weather forecasts to promote water conservation and plant health. The weeding function is executed by a vision system that identifies unwanted vegetation. These system functions are aimed at increasing crop yield, reducing resource waste, and enabling users to manage gardening tasks remotely with ease and precision.

1.3.3 Stakeholder Characteristics

Stakeholders of the FarmBot system include individual hobbyists who are keen on technology and sustainable living, educational institutions that aim to introduce students to the intersection of agriculture and technology, small-scale sustainable farmers looking for precision agriculture solutions, and research organizations focusing on agricultural technologies. Hobbyists may prioritize user experience and educational content, institutions may focus on the system's ability to be used as a teaching tool, farmers require reliability and efficacy in crop production, and research organizations might focus on data collection capabilities and customizability for experiments.

1.3.4 Limitations

Limitations of the FarmBot system are multifaceted, stemming from hardware constraints, environmental factors, and technical limitations. The hardware, including its size and robotic range of motion, limits the area of cultivation. While the software provides customization options, there is a reliance on user knowledge to fully exploit these features. Environmental limitations include susceptibility to extreme weather conditions which could disrupt the robotic operations and sensor readings. Technical limitations involve the dependence on consistent internet connectivity for updates and remote monitoring, and the need for continuous power supply. Additionally, while FarmBot can manage many day-to-day tasks, certain aspects of gardening such as dealing with large pests or diseases still require human intervention.

1.4 Definitions

CNC	Computer Numerical Control, a method used to control machine tools with software using programmable commands.
STEM	An acronym for Science, Technology, Engineering, and Mathematics, representing fields of education and work focused on these disciplines.
SPI	Serial Peripheral Interface, a synchronous serial communication protocol used for short distance communication, primarily in embedded systems.
I2C	Inter-Integrated Circuit, a serial protocol used to connect low-speed devices like microcontrollers and sensors over short distances.
HTTPS	Hypertext Transfer Protocol Secure, an extension of HTTP enhanced with security measures to ensure encrypted communication over the internet.
RESTful API	Representational State Transfer API, a set of guidelines for creating web services that allow for interaction with RESTful web services.
WCAG	Web Content Accessibility Guidelines, developed through the W3C process, aimed at making web content more accessible to people with disabilities.
IEEE 802	A family of IEEE standards dealing with local area networks and metropolitan area networks, covering physical and data link layers.
SSDF	Secure Software Development Framework, guidelines proposed by NIST for integrating security practices into the software development lifecycle.
NIST	National Institute of Standards and Technology, a U.S. Department of Commerce agency that develops and promotes measurement, standards, and technology.

Table 1.1: Definitions

2. References

[1] "Introduction to farmbot software documentation." https://software.farm.bot/v15/docs/introx, 2024. Accessed: 2024-03-27.

[2] "ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering." ISO/IEC/IEEE 29148:2018(E), Nov. 2018. doi: 10.1109/IEEESTD.2018.8559686.

This document is written with respect to the IEEE 29148-2018 standard [2].

3. Specific Requirements

3.1 External Interfaces

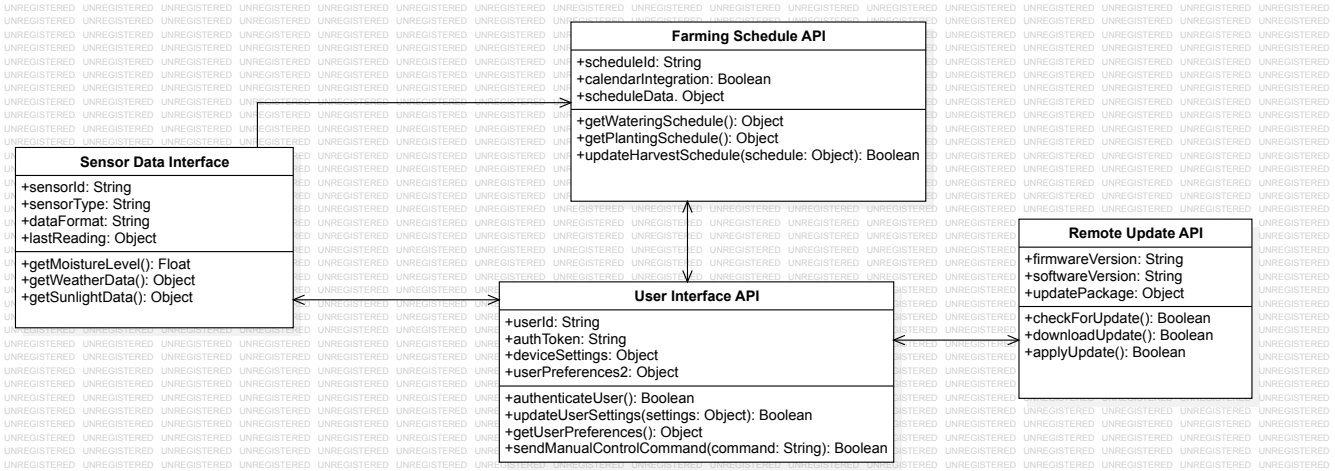

Figure 3.1: Class Diagram for External Interfaces

The FarmBot system is designed with multiple external interfaces to interact with system elements and external entities, including user input mechanisms and automated data retrieval systems. These interfaces are crucial for the operation of the FarmBot and define how it communicates with the external world.

3.1.1 User Interface API

- **Name:** User Interface API

- **Purpose:** To allow users to interact with the FarmBot system for monitoring and control purposes.
- **Source of Input:** User through web or mobile interface.
- **Destination of Output:** FarmBot hardware and software systems for execution of tasks.

3.1.2 Plant Database API

- **Name:** Plant Database API
- **Purpose:** To provide access to a database of plant types and their growing requirements.
- **Source of Input:** FarmBot system querying for data.
- **Destination of Output:** User interface for display, and FarmBot's scheduling system for planning.

3.1.3 Sensor Data Interface

- **Name:** Sensor Data Interface
- **Purpose:** To collect environmental data such as soil moisture and temperature for the FarmBot to make informed decisions.
- **Source of Input:** Environmental sensors deployed within the FarmBot operational area.
- **Destination of Output:** FarmBot's decision support system and user interface for real-time environmental data display.

3.1.4 Diagnostic and Logging API

- **Name:** Diagnostic and Logging API

- **Purpose:** To collect and maintain logs related to system diagnostics and operations.
- **Source of Input:** FarmBot hardware and software components.
- **Destination of Output:** Centralized logging storage, and the user interface for system status reports.

3.1.5 Weather API

- **Name:** Weather API
- **Purpose:** To provide weather forecasts to optimize FarmBot's farming operations.
- **Source of Input:** Online weather service.
- **Destination of Output:** FarmBot's scheduling system to adjust operations based on weather conditions.

3.1.6 Remote Update API

- **Name:** Remote Update API
- **Purpose:** To manage and deploy software updates to the FarmBot system.
- **Source of Input:** Remote update server.
- **Destination of Output:** FarmBot's internal systems to apply updates.

3.1.7 GPS Communication Interface

- **Name:** GPS Communication Interface
- **Purpose:** To provide real-time geolocation data for the FarmBot.
- **Source of Input:** GPS satellites.

- **Destination of Output:** FarmBot's control system for accurate positioning and tracking.

3.1.8 Farming Schedule API

- **Name:** Farming Schedule API
- **Purpose:** To handle the timing and execution of farming tasks.
- **Source of Input:** User-defined schedules and automated system recommendations.
- **Destination of Output:** FarmBot's actuators and task management system.

These interfaces incorporate bi-directional flows of information where applicable, facilitating a responsive and adaptable system. The inputs and outputs defined here are based on conceptual knowledge and should be verified against the actual FarmBot API documentation for accuracy.

3.2 Functions

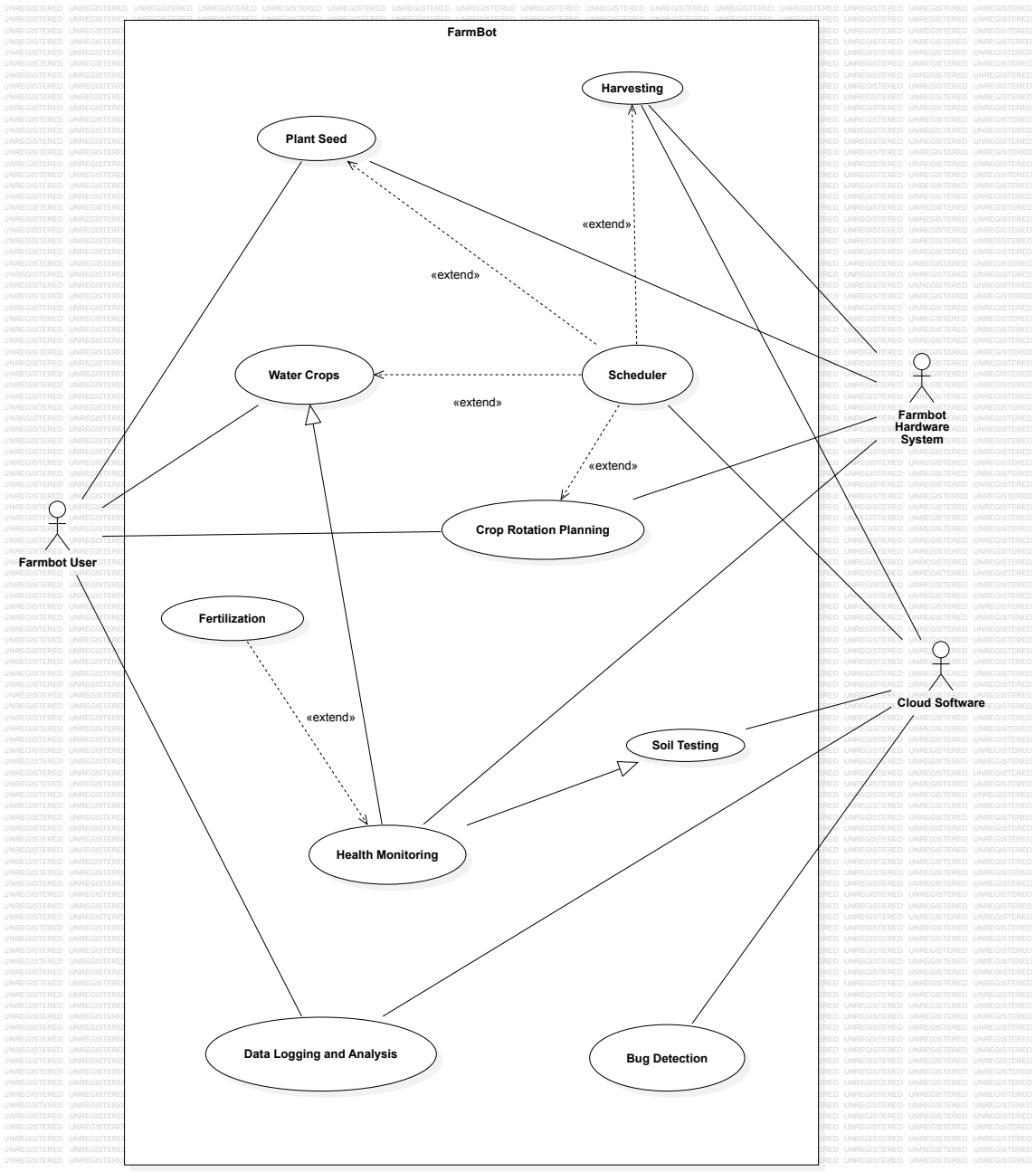

Figure 3.2: Use Case Diagram

Use case name	Seed Planting
Actors	FarmBot, Farmer
Description	Automate the process of planting seeds on the current state of the dynamic garden topography.
Data	Seed type, planting depth, spacing, garden layout, planting coordinates
Preconditions	FarmBot is operational and loaded with the correct seed types.
Stimulus	Farmer selects the seeding plan through the control interface.
Basic flow	Farmer inputs the garden layout and seed details → FarmBot plants seeds according to the specified parameters.
Alternative flow	Manual intervention for complex planting patterns not recognized by FarmBot.
Exception flow	Seed jam or misplacement triggers an alert to the farmer for manual resolution.
Post conditions	Seeds are planted according to the layout; system is ready for the next task.

Table 3.1: Use Case Model for Plant Seeding

Use case name	Watering
Actors	FarmBot, Environmental Sensors
Description	Automate watering of plants based on soil moisture levels.
Data	Soil moisture threshold, amount of water per plant
Preconditions	Water supply is connected and sensors are calibrated.
Stimulus	Soil moisture levels fall below the set threshold.
Basic flow	Sensors detect low moisture → FarmBot waters the plants accordingly.
Alternative flow	Scheduled watering if sensor data is unavailable.
Exception flow	Water supply error triggers an alert.
Post conditions	Plants are watered; soil moisture is adequate.

Table 3.2: Use Case Model for Watering

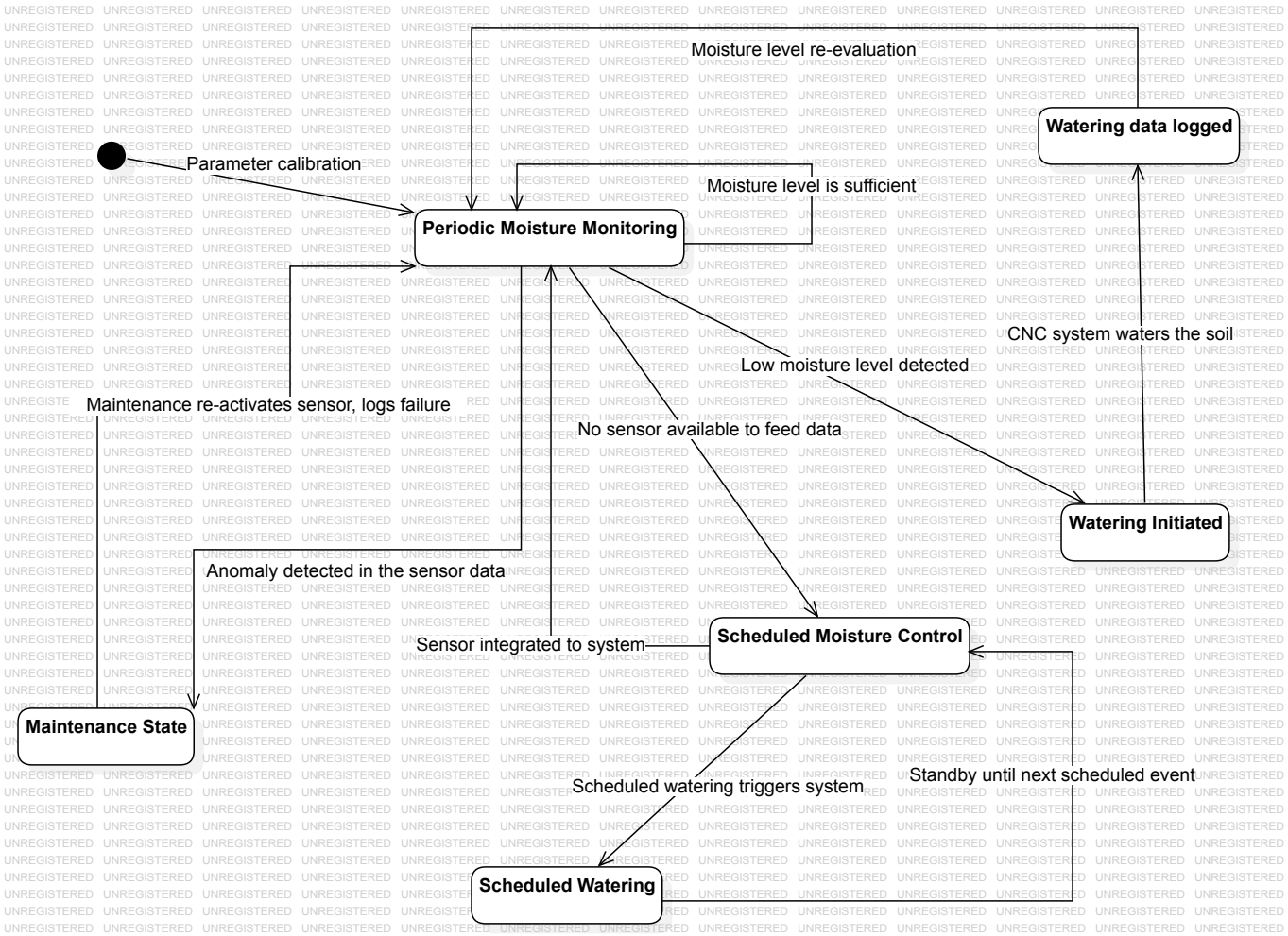

Figure 3.3: State Diagram for Watering

Use case name	Weeding
Actors	FarmBot
Description	Identify and remove weeds from the garden.
Data	Plant vs. weed identification parameters
Preconditions	FarmBot's vision system is calibrated for plant/weed differentiation.
Stimulus	Scheduled weeding or manual initiation by the farmer.
Basic flow	FarmBot identifies weeds → Removes weeds mechanically.
Alternative flow	Manual removal of weeds that FarmBot cannot identify.
Exception flow	Damage to a plant is detected, pausing operation for review.
Post conditions	Garden is free of weeds.

Table 3.3: Use Case Model for Weeding

Use case name	Soil Testing
Actors	FarmBot, Farmer
Description	Automate the collection and analysis of soil samples.
Data	pH, moisture content, nutrient levels
Preconditions	FarmBot is equipped with soil testing tools and reagents.
Stimulus	Scheduled testing or farmer request.
Basic flow	FarmBot collects soil samples → Analyzes for key parameters → Reports results.
Alternative flow	Samples sent to a lab for more comprehensive analysis.
Exception flow	Inconclusive results trigger re-sampling.
Post conditions	Farmer receives soil health report.

Table 3.4: Use Case Model for Soil Testing

Use case name	Harvesting
Actors	FarmBot, Farmer
Description	Automate the harvesting of ripe crops.
Data	Crop type, ripeness indicators
Preconditions	FarmBot's vision system can identify ripe crops.
Stimulus	Crops reach maturity according to the growth timeline or visual inspection.
Basic flow	Identification of ripe crops → Automated or assisted harvesting.
Alternative flow	Manual harvesting for delicate or complex crops.
Exception flow	Damage to crops during harvesting triggers an alert.
Post conditions	Ripe crops are harvested.

Table 3.5: Use Case Model for Harvesting

Use case name	Monitoring Plant Health
Actors	FarmBot, Environmental Sensors, Farmer
Description	Continuously monitor plant health and environmental conditions.
Data	Plant growth metrics, environmental data
Preconditions	Sensors and vision systems are operational.
Stimulus	Ongoing monitoring as part of the daily cycle.
Basic flow	Collect data → Analyze for health indicators → Alert farmer of issues.
Alternative flow	Request manual inspection if anomalies are detected.
Exception flow	System malfunction halts monitoring until fixed.
Post conditions	Farmer is informed of plant health status.

Table 3.6: Use Case Model for Monitoring Plant Health

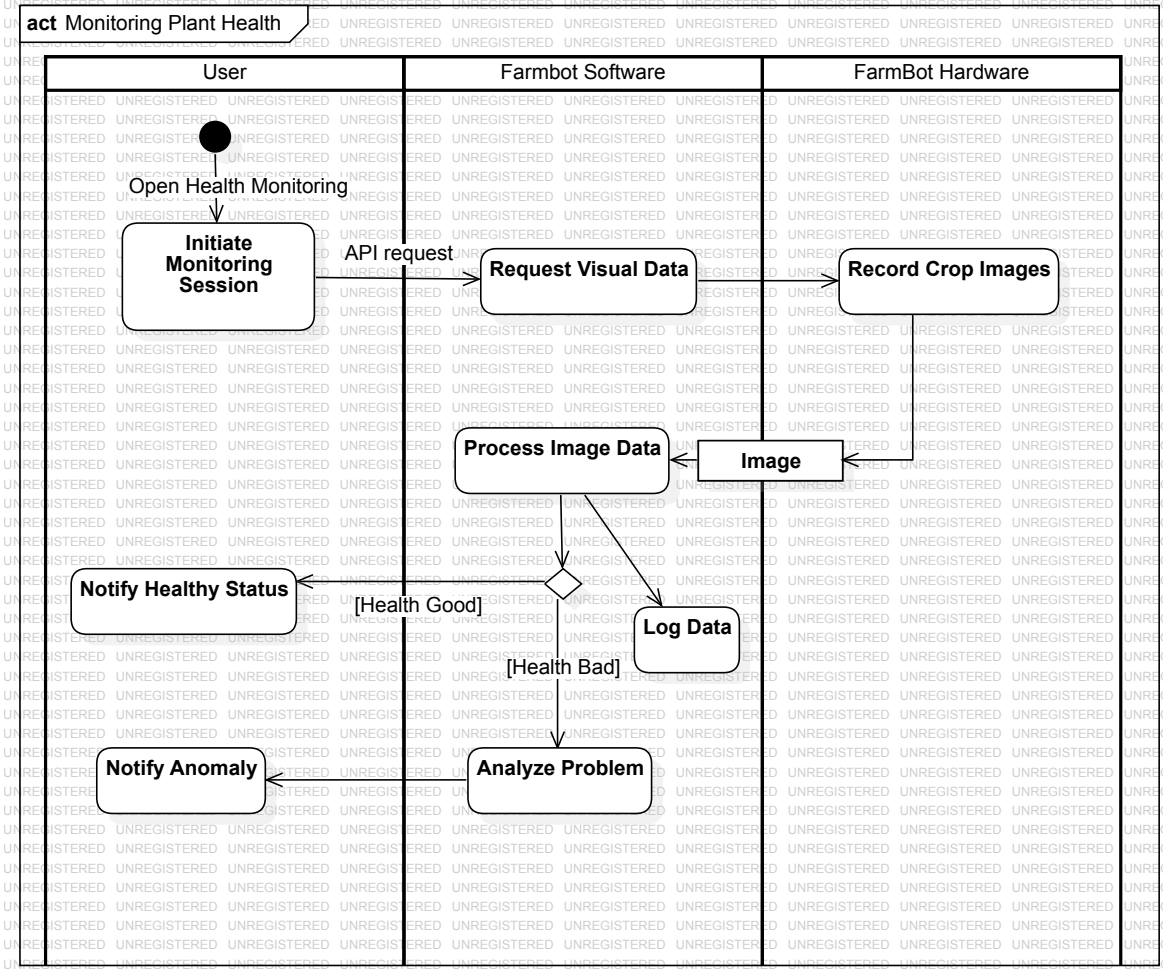

Figure 3.4: Activity Diagram for Monitoring Plant Health

Use case name	Fertilizing
Actors	FarmBot, Farmer
Description	Automate the distribution of fertilizers based on soil and plant needs.
Data	Fertilizer type, application rate, targeted plants
Preconditions	Fertilizer is loaded and FarmBot is calibrated for accurate distribution.
Stimulus	Scheduled application or as dictated by soil/plant health data.
Basic flow	Determine nutrient needs → FarmBot applies fertilizer accordingly.
Alternative flow	Hand application for areas FarmBot cannot reach or for special treatments.
Exception flow	Over or under fertilization detected, prompting adjustment or alert.
Post conditions	Plants receive the necessary nutrients; system ready for next operation.

Table 3.7: Use Case Model for Fertilizing

Use case name	Data Logging and Analysis
Actors	FarmBot, Farmer, Cloud Services
Description	Collect and analyze farming data to optimize future farming practices.
Data	Plant growth data, environmental conditions, FarmBot operational data
Preconditions	Data collection tools and analysis software are operational.
Stimulus	Continuous data collection as part of FarmBot's operations.
Basic flow	Data is logged → Transmitted to cloud for analysis → Insights are generated.
Alternative flow	Data analyzed locally for immediate adjustments.
Exception flow	Data corruption or loss triggers a system check and potential reconfiguration.
Post conditions	Farmer receives actionable insights and historical data is archived.

Table 3.8: Use Case Model for Data Logging and Analysis

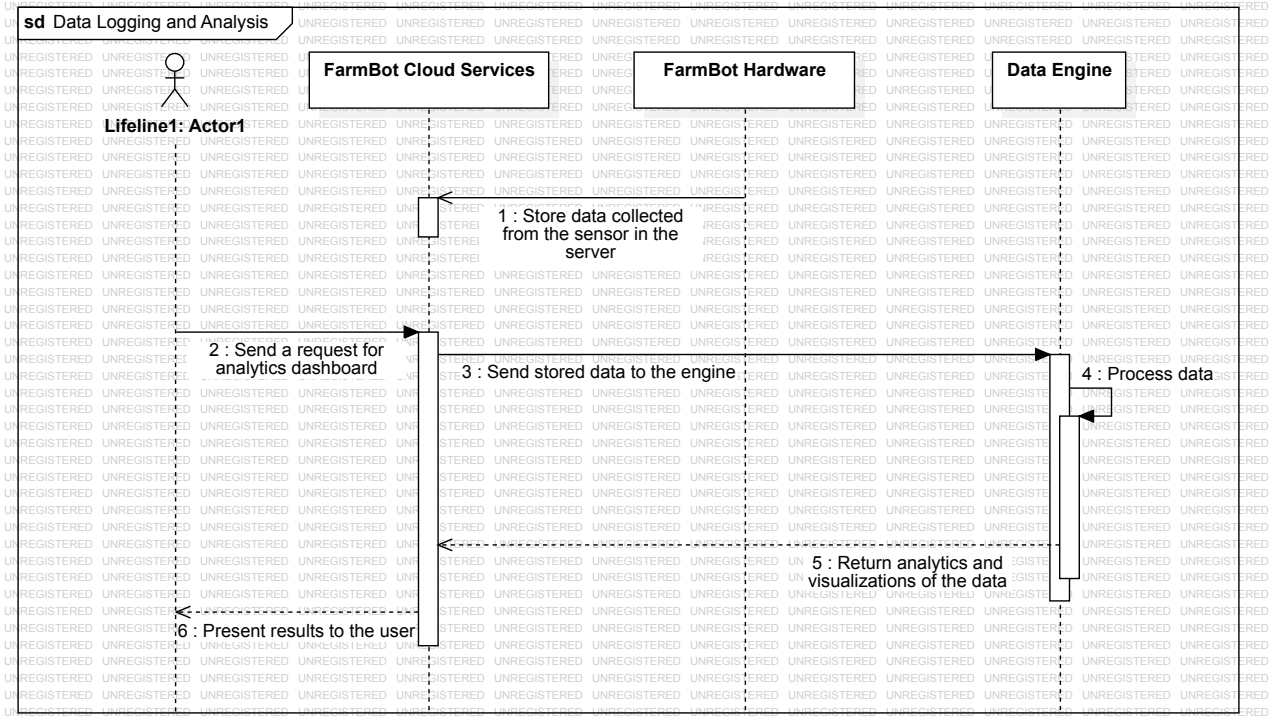

Figure 3.5: Sequence Diagram for Data Logging and Analysis

Use case name	Pest Detection and Management
Actors	FarmBot, Farmer
Description	Identify pests and apply appropriate management strategies.
Data	Pest types, affected plants, management methods
Stimulus	Regular monitoring or farmer initiates pest scan.
Basic flow	Detect pests → Identify → Apply non-chemical/chemical management.
Alternative flow	Manual intervention for pests not recognized or for delicate areas.
Exception flow	Incorrect pest management application triggers review and correction.
Post conditions	Pest levels are managed; minimal impact on crops.

Table 3.9: Use Case Model for Pest Detection and Management

Use case name	Crop Rotation Planning
Actors	FarmBot, Farmer
Description	Assist in planning crop rotation to maintain soil health and reduce pests.
Data	Historical crop data, soil health data, crop rotation strategies
Preconditions	Access to historical farming data and crop rotation models.
Stimulus	End of growing season or during planning phase.
Basic flow	Analyze data → Generate crop rotation plan → Implement plan with FarmBot assistance.
Alternative flow	Farmer adjusts plan based on personal knowledge or preferences.
Exception flow	Inadequate data for planning triggers manual planning process.
Post conditions	Crop rotation plan is established; ready for implementation in the next season.

Table 3.10: Use Case Model for Crop Rotation Planning

3.3 Logical Database Requirements

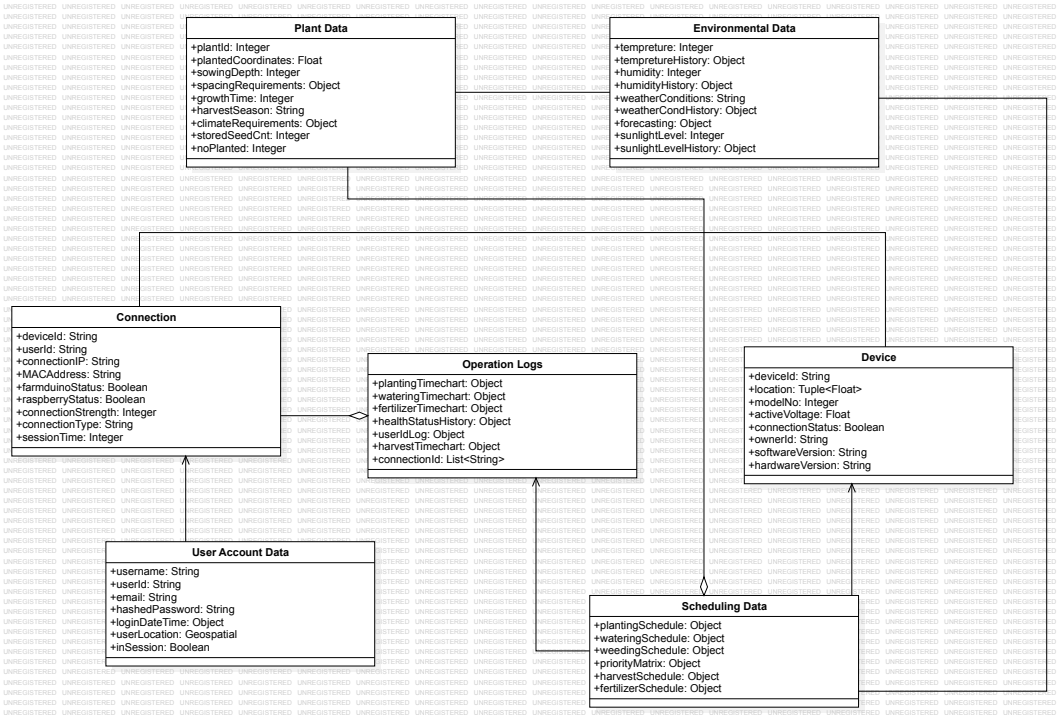

Figure 3.6: Logical Database Class Diagram

The FarmBot system's logical database is structured to optimize the automation of precision agriculture. Key data objects include:

Planting Data: Holds records for plant species, planting coordinates, sowing depth, and spacing requirements.

Environmental Data: Stores readings from environmental sensors, including moisture levels, temperature, and weather conditions.

Operation Logs: Contains a historical record of all operations performed by FarmBot, such as watering, weeding, and fertilizing actions, along with timestamps and outcomes.

User Account Data: Manages user information, including credentials, configuration preferences, and roles.

Scheduling Data: Keeps track of planting and watering schedules, task prioriti-

zation, and event triggers based on environmental data.

The class diagram for these data objects would illustrate their interrelationships, ensuring clear understanding without the need for additional dictionaries.

3.4 Design Constraints

The FarmBot system is subject to the following design constraints:

Environmental Conditions: FarmBot must withstand outdoor conditions and variable weather.

Compatibility: The system should be compliant with common agricultural software and hardware interfaces to ensure interoperability with various sensors and devices.

Regulatory Compliance: Adherence to international standards for safety, privacy, and environmental impact is mandatory.

Resource Efficiency: Design must promote energy and water conservation, fitting into sustainable agriculture practices.

3.5 System Quality Attributes

For FarmBot, the following quality attributes are essential:

Reliability: As a robotic system used in agriculture, consistent performance and minimal downtime are critical.

Usability: The user interface must be accessible to individuals with varying levels of technical expertise, supporting a diverse user base.

Maintainability: Ease of updating the software and replacing hardware components is crucial for long-term operation.

Scalability: The system should be capable of being expanded in functionality and size to accommodate larger farming operations.

3.6 Supporting Information

The supporting information for the FarmBot project encompasses comprehensive documentation available on the official website, including assembly guides, software and hardware documentation, FAQs, and a troubleshooting wiki. Additionally, the FarmBot community forum serves as a collaborative platform for users to share experiences, offer solutions, and provide feedback. The open-source nature of the project encourages contributions from developers, which are facilitated through repositories hosting the FarmBot software and hardware design files. Educational resources for schools and research institutions are also provided to promote STEM education through practical application in agriculture.

4. Suggestions to Improve The Existing System

4.1 System Perspective

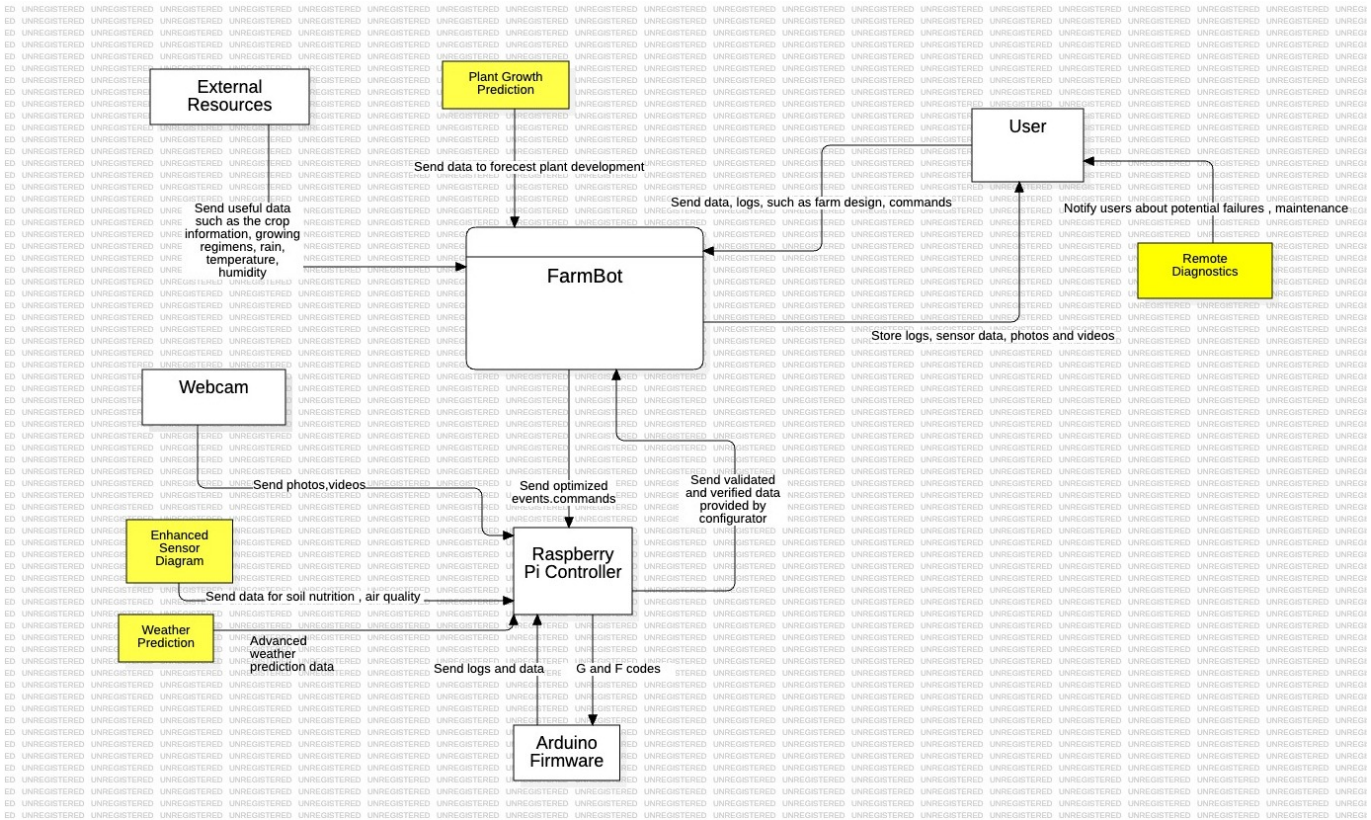

Figure 4.1: FarmBot Context Diagram with Suggestions

The system's architecture, designed to automate precision farming through FarmBot, interacts with various external and internal components. This section discusses improvements in the context of the system's interaction with its environment, highlighting four key areas for enhancement.

4.1.1 Enhanced Sensor Integration

Context and Explanation: Integrating a broader range of environmental sensors into FarmBot's system can significantly improve its adaptive capabilities. Enhanced sensors for detecting soil nutrition and air quality will allow FarmBot to respond more dynamically to the needs of the plants and the conditions of their environment. This could involve incorporating new sensor types that measure specific nutrient levels in the soil or detect pollutants in the air that could affect plant growth.

Suggested Improvements:

- Incorporate nitrogen, phosphorus, and potassium (NPK) soil sensors to optimize fertilizer application.
- Add air quality sensors to monitor the presence of pollutants that could harm plant health.
- Develop a sensor data aggregation module in the software to analyze data from multiple sources for comprehensive environmental assessment.

4.1.2 Advanced Weather Prediction Integration

Context and Explanation: Weather plays a crucial role in farming operations. By integrating advanced weather prediction models that utilize machine learning to process historical weather data, FarmBot can proactively adjust its operations—such as watering, planting, and harvesting schedules—based on accurate weather forecasts.

Suggested Improvements:

- Partner with advanced weather prediction services or invest in the development of proprietary weather prediction algorithms.

- Implement machine learning models that can analyze patterns in long-term weather data to forecast local weather conditions with higher accuracy.
- Automatically adjust FarmBot's operational schedules based on weather predictions to optimize plant growth and resource usage.

4.1.3 Plant Growth Prediction Models

Context and Explanation: Predicting plant growth stages and harvest times can significantly enhance the scheduling efficiency of FarmBot. Integrating plant growth prediction models that leverage historical growth data and current environmental conditions will allow for more precise planning of farming operations.

Suggested Improvements:

- Develop or integrate existing plant growth models into FarmBot's software, tailored to the variety of plants supported by the system.
- Use accumulated data from FarmBot's operations and external data sources to refine and validate the growth models continuously.
- Provide users with predictions on growth stages and optimal harvest times, enhancing the planning and execution of farming tasks.

4.1.4 Remote Diagnostics and Proactive Maintenance

Context and Explanation: Ensuring the reliability of FarmBot's operations requires a system for early detection of potential failures or maintenance needs. Implementing a framework for remote diagnostics and proactive maintenance can help in identifying issues before they lead to system downtime.

Suggested Improvements:

- Develop diagnostic algorithms that monitor system performance and predict potential failures based on operational data and sensor readings.

- Implement a maintenance scheduling module that alerts users to upcoming maintenance tasks and suggests optimal times for their completion.
- Allow for remote firmware updates and system checks to ensure that FarmBot operates with the latest software and within optimal performance parameters.

Each of these improvements aims to enhance the adaptability, efficiency, and reliability of the FarmBot system, leveraging advanced technologies and data-driven strategies to optimize precision farming operations.

4.2 External Interfaces

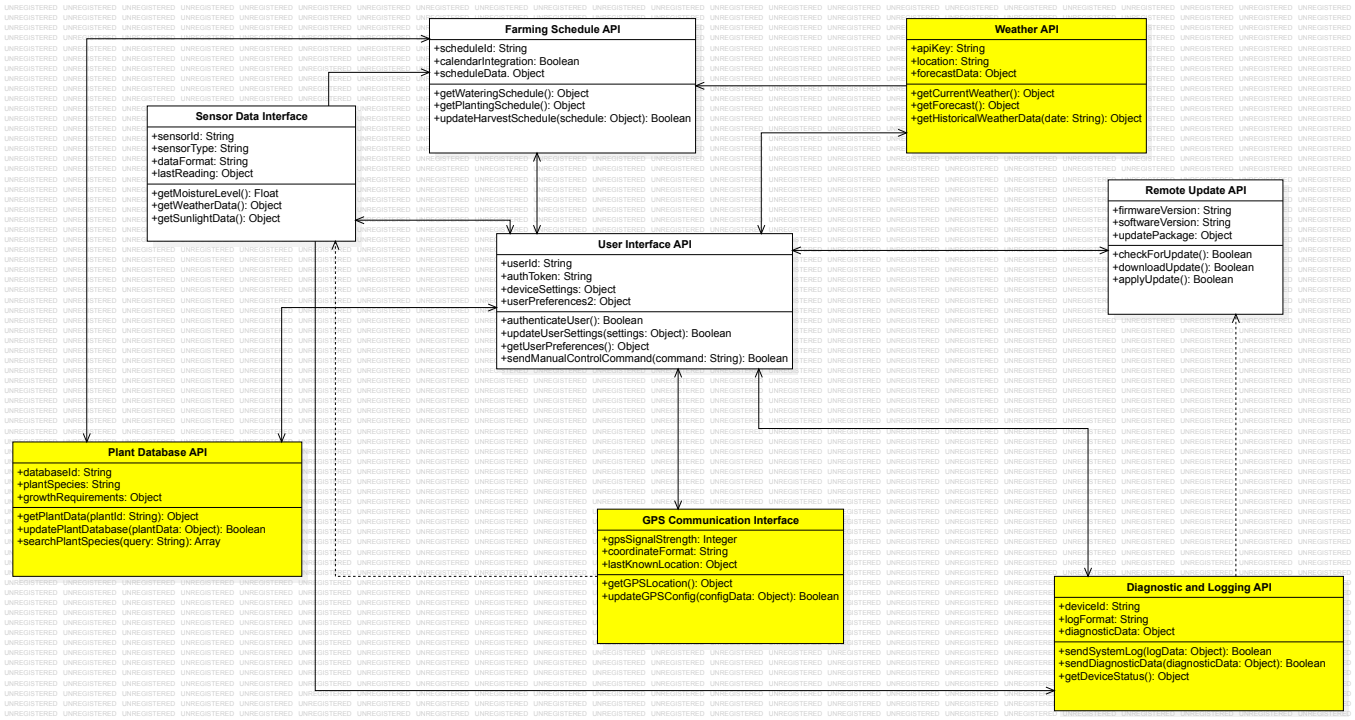

Figure 4.2: Class Diagram for External Interfaces

This section outlines the external interfaces of the FarmBot system, highlighting the integration of both original and suggested interfaces to enhance system functionality.

The context diagram (not included here) visually represents these interfaces and their connections. The following descriptions detail how suggested interfaces can complement and expand the capabilities of the existing system.

4.2.1 Suggested External Interfaces

Plant Database API

- **Purpose:** To enrich FarmBot's knowledge base with extensive plant species data and their specific growth requirements, facilitating more informed decision-making for planting strategies.
- **Improvements:** Integration of a comprehensive plant database allows for automated, species-specific care routines, optimizing growth conditions for a diverse range of crops and enhancing yield efficiency.

GPS Communication Interface

- **Purpose:** To provide precise geolocation capabilities for FarmBot, enabling accurate mapping of the farming area and plant locations.
- **Improvements:** With accurate GPS data, FarmBot can perform tasks with enhanced precision, such as targeted watering or planting, reducing resource waste and improving operational efficiency.

Weather API

- **Purpose:** To fetch real-time and forecasted weather data, allowing FarmBot to adapt its operations based on current and upcoming weather conditions.
- **Improvements:** Weather adaptability ensures that FarmBot can make proactive adjustments to its routines, such as delaying watering before rain, thus optimizing resource use and protecting plants from adverse weather.

Diagnostic and Logging API

- **Purpose:** To collect detailed diagnostics and operational logs from FarmBot, facilitating remote troubleshooting and system optimization.
- **Improvements:** Enhanced diagnostic capabilities enable predictive maintenance and quick resolution of issues, improving system reliability and uptime.

4.2.2 Integration with Existing Interfaces

The suggested interfaces complement the existing system's capabilities, forming a cohesive and intelligent farming solution:

- The **User Interface API** becomes a more powerful tool for users, offering access to a broader range of data and controls, from selecting plants from the **Plant Database API** to viewing weather forecasts via the **Weather API**.
- The **Sensor Data Interface** and **GPS Communication Interface** work in tandem to provide environmental and locational data, enhancing the precision of FarmBot's operations.
- Integration of the **Diagnostic and Logging API** with the **Remote Update API** ensures that FarmBot's software remains up-to-date and operates efficiently, with maintenance and updates informed by comprehensive diagnostic data.

4.2.3 Improvement Recommendations

To fully leverage the potential of the suggested external interfaces, the following recommendations are proposed:

- Data Synchronization: Implement robust data synchronization mechanisms between the FarmBot system and the external APIs to ensure real-time accuracy and responsiveness.
- User-Centric Design: Continuously refine the User Interface API to present the enriched data and controls in an intuitive and accessible manner, enhancing the user experience.

- Security and Privacy: Ensure all external communications are encrypted, and data privacy is maintained, especially when integrating third-party APIs like weather services and GPS data providers.

  By incorporating these suggested external interfaces and adhering to the improvement recommendations, the FarmBot system can achieve significant advancements in automation, efficiency, and user engagement.

4.3 Functions

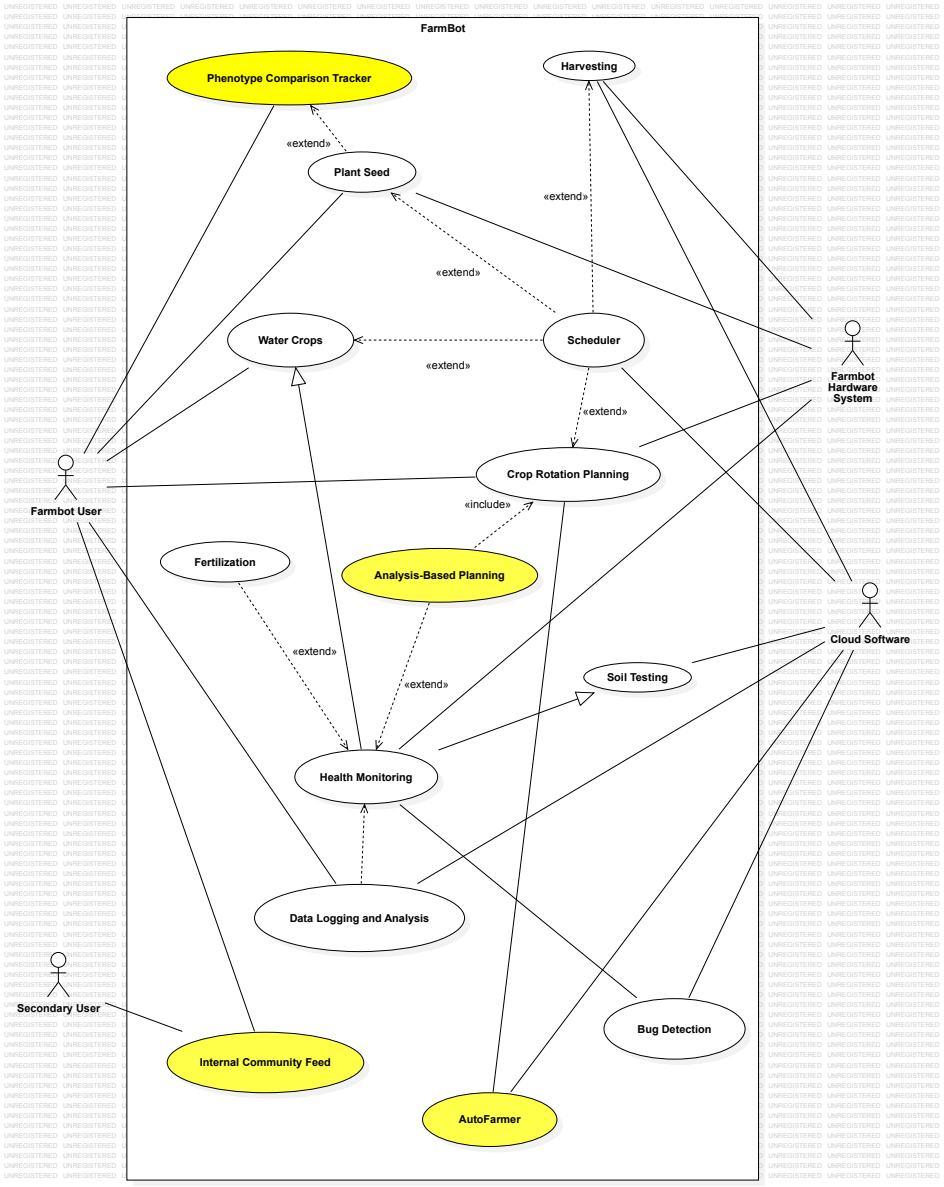

Figure 4.3: Use Case Diagram with Suggestions

Use case name	Internal Community Feed
Actors	Farmbot App, Users
Description	Embed the FarmBot Community Blog into the app for easy integration of utilities, schedule sharing, etc.
Data	User profile info, recent activities, image sharing, garden layout, planting history
Preconditions	Active internet connection and data sharing permissions.
Stimulus	FarmBot users connect and interact via. social tab.
Basic flow	Users share experiences through the social portal → Other users interact with content.
Alternative flow	Importation of the data and activities posted on the original community page.
Exception flow	Connectivity and legal social platform issues.
Post conditions	Users interact and improve the platform through bug reports and implementation suggestions.

Table 4.1: Use Case Model for Internal Community Feed

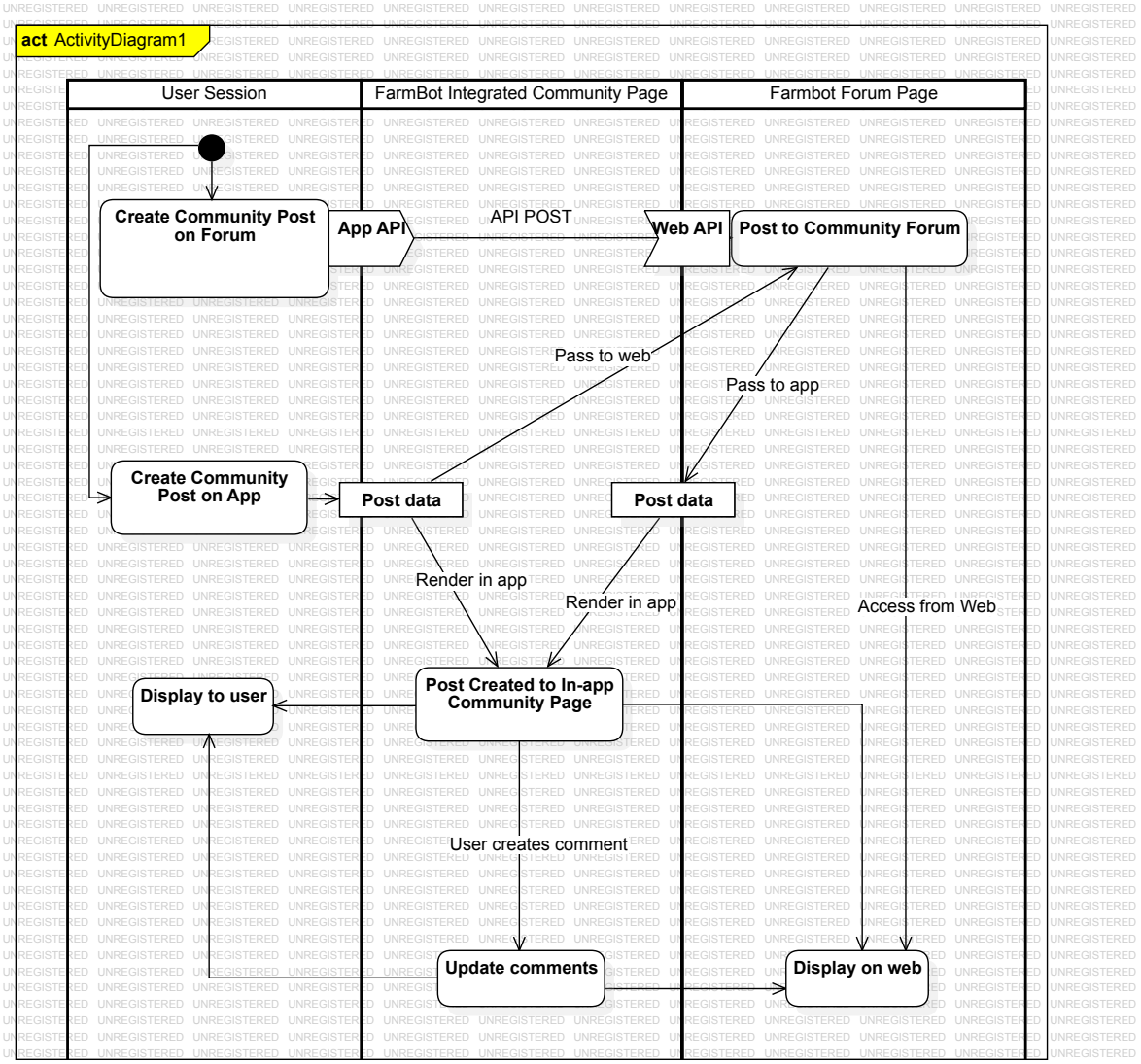

Figure 4.4: Activity Diagram for Internal Community Feed

Use case name	Analysis-based Planting
Actors	Scheduler, User, Farmbot
Description	Use past crop data to improve predictions and plant crops to have higher yields. Use historical data to build on the already existing analysis feature. Add past plant performance as an additional parameter to predict plant growth performance
Data	Past harvest data, weather data, geospatial data, soil quality and nutrient data, user specifications
Preconditions	FarmBot has harvested at least one crop cycle in the same location at the same season.
Stimulus	User decides to optimize scheduling and planting using the past data.
Basic flow	One crop cycle is completed → The data from the previous operation is used to optimize the next one.
Alternative flow	User opts to not use past data and planning is done as standard.
Exception flow	Past data is non-existent or corrupt. There are new variables introduced making the data obsolete.
Post conditions	Crop planning is done better resulting in higher yields.

Table 4.2: Use Case Model for Analysis-based Planting

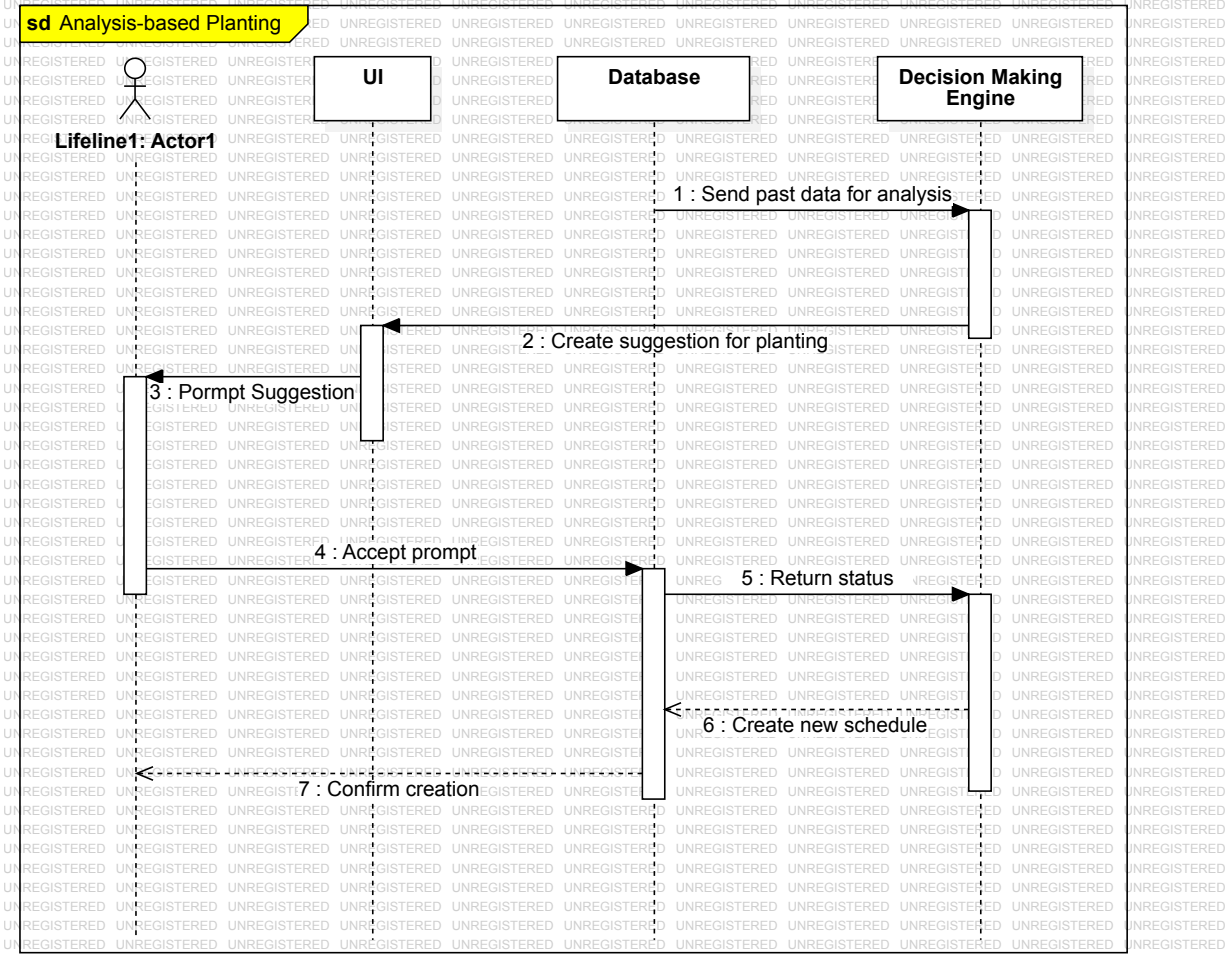

Figure 4.5: State Diagram for Analysis-based Planting

Use case name	Phenotype Comparison Tracker
Actors	User, FarmBot
Description	An automated dashboard for the tracking of scientific experiments on plant genetics and phenotype results. The FarmBot software is extended to support internal experimentation to support the scientific use of the system. Seed labeling and crop tracking is embedded to the system.
Data	Seed type, planting depth, spacing, garden layout, planting coordinates, genetic differences, test batch, control batch
Preconditions	All other variables are held constant except seed types of the same plant.
Stimulus	A hypothesis is tested regarding the effects of plant genes on growth and yield.
Basic flow	Experiment is conducted on a batch → The results are automatically analyzed and stored by the FarmBot system.
Alternative flow	Plants are sown and harvested normally.
Exception flow	Additional variables are introduced by accident, disabling the experiment.
Post conditions	Seeds are specifically tracked, making the experiment easier to conduct with less human error.

Table 4.3: Use Case Model for Phenotype Comparison Tracking

Use case name	Auto-Farmer
Actors	FarmBot, User, Scheduler
Description	Automate the process of planting seeds, watering, weeding, fertilizing, and other activities to make the system fully autonomous. The user is only tasked with base level error handling.
Data	Crop state, planting options, plant health, schedule, weather data
Preconditions	FarmBot is operational and loaded with the correct seed types, fertilizer, and water source.
Stimulus	The user specifies farming choices through the control interface such as plant types, maintenance frequency, and fertilizer amount. Constant weather data is supplied to the system.
Basic flow	User sets up the system with its parameters → FarmBot plants seeds according to the specified parameters and does continuous up-keeping of the plants automatically.
Alternative flow	AutoFarmer is engaged only for certain specific time frames.
Exception flow	Software or hardware errors are introduced or the system runs out of resources.
Post conditions	The system constantly tends to the crops allowing it to operate autonomously.

Table 4.4: Use Case Model for AutoFarmer

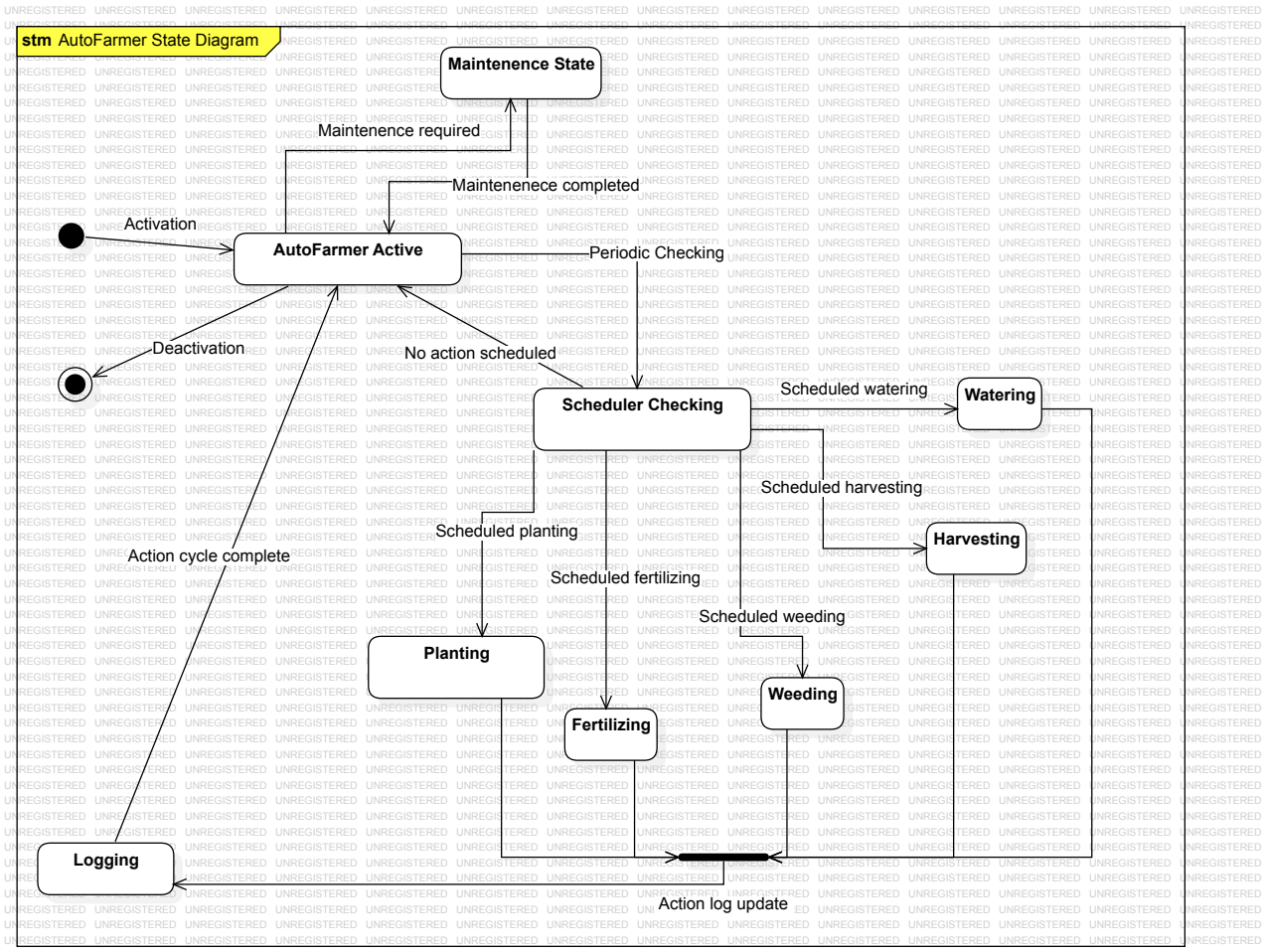

Figure 4.6: State Diagram for AutoFarmer

4.4 Logical Database Requirements

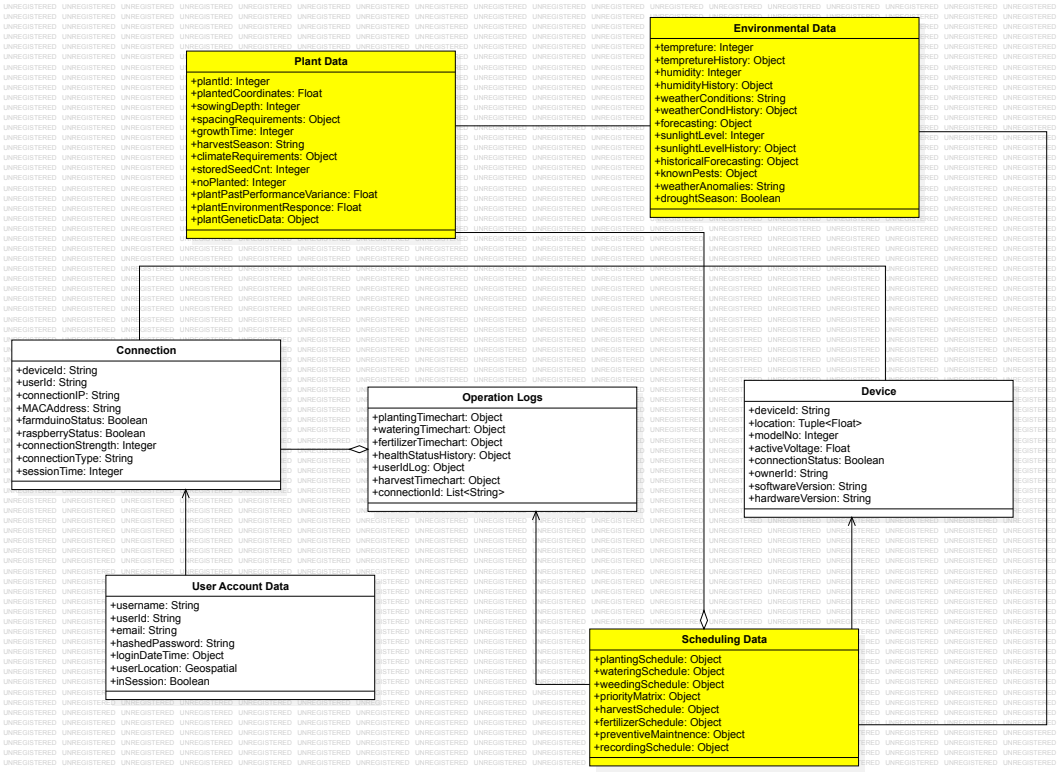

Figure 4.7: Logical Database Class Diagram with Suggestions

This section details the enhancements to the FarmBot system's logical database, specifically focusing on the addition of new attributes to Plant Data, Environmental Data, and Scheduling Data. These improvements aim to refine FarmBot's operational efficiency, adaptability, and data analytics capabilities.

4.4.1 Enhancements to Plant Data

Plant Data has been augmented with the following attributes to provide a more comprehensive understanding of each plant's performance and requirements:

- **plantPastPerformanceVariance (Float):** Measures the variance in past performance of plants, aiding in the prediction of future growth patterns and identifying plants that deviate from expected growth trajectories.

- **plantEnvironmentResponse (Float):** Quantifies each plant's responsiveness to environmental conditions, enabling more personalized care strategies that adapt to the plant's specific needs.
- **plantGeneticData (Object):** Contains genetic information of the plants, supporting advanced analysis for breeding optimization and genetic resilience to pests or diseases.

4.4.2 Enhancements to Environmental Data

Environmental Data now includes attributes that capture broader ecological and meteorological factors:

- **knownPests (Object):** A record of pests known to affect the farming area, facilitating targeted pest management strategies.
- **weatherAnomalies (String):** Describes historical weather anomalies in the area, providing insights for better preparation against unusual weather patterns.
- **droughtSeason (Boolean):** Indicates whether the current season is prone to drought, aiding in water conservation efforts and drought-resistant planting decisions.

4.4.3 Enhancements to Scheduling Data

Scheduling Data has been enhanced with attributes focused on maintenance and operational recording:

- **preventiveMaintenance (Object):** Outlines scheduled maintenance activities for FarmBot's machinery, aiming to prevent equipment failure and extend the system's lifespan.
- **recordingSchedule (Object):** Specifies the schedule for recording various operational data points, ensuring comprehensive data collection for analysis and optimization.

4.4.4 Class Diagram with Associations

Implementation Considerations: Incorporating these attributes requires updating the FarmBot software to leverage this enriched data for operational decision-making. This might involve algorithm updates, user interface adjustments for data visualization, and enhanced analytics capabilities to interpret the new data types.

These database enhancements are poised to significantly improve FarmBot's precision agriculture capabilities, making the system more responsive to plant and environmental variables and proactive in maintenance scheduling. The goal is to optimize plant health, yield, and system reliability through data-driven insights.

4.5 Design Constraints

To further enhance the FarmBot system, it is vital to address and adapt to additional design constraints that arise from proposed improvements:

Data Privacy and Security: With the integration of advanced data analytics and genetic information, ensuring the privacy and security of data becomes paramount. Adherence to data protection regulations like GDPR for users' and plants' genetic data is mandatory.

Advanced Sensor Compatibility: The incorporation of enhanced sensors for soil nutrition and air quality demands compatibility with an expanded range of sensor technologies. This necessitates modular and adaptable software drivers and interfaces.

Weather Prediction Technologies: The integration of advanced weather prediction requires handling large datasets and complex algorithms, which may constrain system processing capabilities and require optimization for efficient data processing.

Resource Intensity: The improvements in predictive maintenance, plant growth modeling, and environmental responsiveness increase the system's computational and energy resources. Optimizing for low-power operations and efficient data processing will be essential.

4.6 System Quality Attributes

Improvements to the FarmBot system should prioritize the following quality attributes to ensure the system's effectiveness and user satisfaction:

Security: Enhancements in data analytics and personalization features increase the need for robust security measures to protect user and plant data against unauthorized access and cyber threats.

Performance: Advanced functionalities, such as real-time environmental monitoring and predictive analytics, require the system to process and respond to data efficiently, maintaining high performance without lag.

Adaptability: The system must be flexible to accommodate new sensors, data models, and user preferences without significant overhauls, allowing for seamless updates and integration of new features.

User Experience: With the addition of complex data analytics and more detailed control options, maintaining an intuitive and user-friendly interface is crucial to ensure that users of all technical levels can effectively interact with the system.

4.7 Supporting Information

The suggested improvements to the FarmBot system necessitate updated and comprehensive supporting information:

Updated Documentation: Detailed documentation covering new sensors, weather prediction integration, and plant growth models will be essential for users and developers to understand and leverage the system's enhanced capabilities.

Educational Resources: To aid in the adoption of new features and technologies, providing updated educational resources and tutorials will help users maximize the benefits of the system's improvements.

Community Engagement: Encouraging feedback and contributions from the FarmBot community on the new features will be vital for iterative improvements and user satisfaction. A platform for sharing customizations, data models, and integration

tips should be supported.

Security Guidelines: With increased focus on data security, providing users with clear guidelines and best practices for data privacy and system security will be crucial to maintaining trust.

By addressing these design constraints and focusing on enhancing the system's quality attributes, the FarmBot project can achieve significant advancements in precision agriculture, making the system more responsive, efficient, and user-friendly.