Department of Computer Engineering

CENG350 Software Engineering

Software Requirements Specification for FarmBot

Group 52

By

Emre Çam

Tufan Özkan

Sunday 24th March, 2024

Contents

List of Figures iii

List of Tables iv

1 Introduction 1

1.1 Purpose of the System . . . 1

1.2 Scope . . . 2

1.3 System Overview . . . 3

1.3.1 System Perspective . . . 3

1.3.1.1 System Interfaces . . . 4

1.3.1.2 User Interfaces . . . 6

1.3.1.3 Hardware Interfaces . . . 8

1.3.1.4 Software Interfaces . . . 9

1.3.1.5 Communication Interfaces . . . 9

1.3.1.6 Memory Constraints . . . 9

1.3.1.7 Operations . . . 10

1.3.2 System Functions . . . 11

1.3.3 Stakeholder Characteristics . . . 11

1.3.4 Limitations . . . 11

1.4 Definitions . . . 14

2 References 15

3 Specific Requirements **16**

3.1 External Interfaces . . . 16
3.2 Functions . . . 16
3.3 Logical Database Requirements . . . 16
3.4 Design Constraints . . . 17
3.5 System Quality Attributes . . . 18
3.5.1 Usability Requirements . . . 18
3.5.2 Performance Requirements . . . 19
3.5.3 Reliability Requirements . . . 21
3.5.4 Availibility Requirements . . . 21
3.5.5 Security Requirements . . . 21
3.5.6 Maintability Requirements . . . 22
3.5.7 Portability Requirements . . . 23
3.6 Supporting Information . . . 24

4 Suggestions to Improve The Existing System **25**

4.1 System Perspective . . . 25
4.2 External Interfaces . . . 25
4.3 Functions . . . 25
4.4 Logical Database Requirements . . . 26
4.5 Design Constraints . . . 26
4.6 System Quality Attributes . . . 26
4.7 Supporting Information . . . 27

List of Figures

1.1 Context Diagram of FarmBot Express 3
1.2 Example 12

List of Tables

1 Table Example v

Revision History

(Clause 9.2.1)

You can use a table to show your reports' versions and their date.

To create tables, you can use https://www.latex-tables.com/; the Table 1 is generated by it.

A	B	C	D

Table 1: Table Example

1. Introduction

This document provides the Software Specification Requirement (SRS) of an open-source precision agriculture CNC farming project that lets the user grow food from anywhere with a web app. The website for this system is https://farm.bot

1.1 Purpose of the System

The purpose of the FarmBot system is to revolutionize agricultural practices through automation. FarmBot aims to regularize farming methods, enhance crop yields and minimize resource waste. Additionally; FarmBot provide a platform to manage and monitor their crops and agricultural operations efficiently. Through automation and monitoring, the system aims to:

- Automate planting, watering, weeding and harvesting processes to reduce human work and improve efficiency.
- Provide precise control over the environmental factors such as soil moisture, harmful weeds.
- Facilitate real-time monitoring from web application to enhance crop health.
- Provide an environment to design a layout for crops and implement that design in real world.

1.2 Scope

In the scope of this system, Farmbot is humanity's open-source CNC farming machine that automatically grows food right in your backyard keeping you in complete control, as opposed to the current food production system which no longer has control over how the food is produced.

- Farmbot is provided with 95 pre-assembled elements to be set up quickly and easily and perform the basic functions needed to grow a garden.
- Raspberry Pi computers operate the system with its hardware components (webcam, Arduino firmware microcontroller combined with powerful stepper motors and dynamic devices, ph sensors) and user application requests.
- Farmduino microcontrollers allow positioning of the tool head with millimetre accuracy for sowing, and watering plants in any pattern and frequency with given built-in features of plants based on their type, age, and soil conditions. The sensors also send the condition of soil and weather data of pH, temperature, and moisture.
- The Onboard camera system allows user to monitor their garden, and capture images, and the farmbot can take action by detecting the weeds utilising advanced computer vision.
- The web and mobile application allows users to control the garden remotely, and configure or update the software of Raspberry Pi computers without any need for hardware change.
- The farm designer and sequence editors allow users to lay out their plants with built-in data for plants, creating optimal layouts each season, and taking care of their plants in the way they want without requiring any coding.
- The farmbot software development allows professionals to modify and use it for

any purpose (e.g. education, business) in hardware tools or develop completely new versions of the farmbot based on the existing library of components.

1.3 System Overview

This section of the document will provide detailed information about the system including all components.

1.3.1 System Perspective

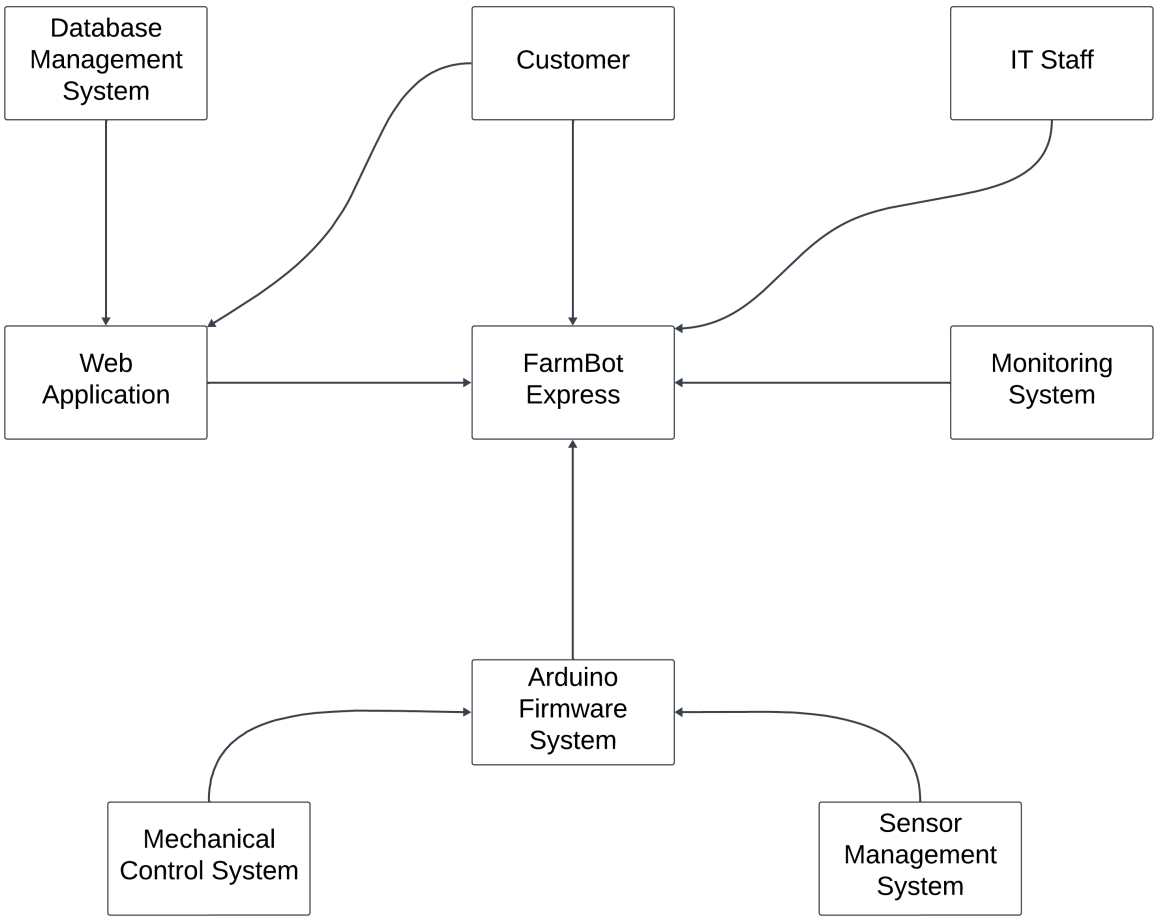

Figure 1.1: Context Diagram of FarmBot Express

FarmBot is not part of a large system nor necessarily requires the download of an application. Hence customers shall go into the web browser of the app and configure their FarmBot using their laptop, tablet, or smartphone. The web application features real-time manual controls and logging, a sequence builder for custom routines for FarmBot to execute.The web application keeps the data and receives the built-in data for plants from the database management system. The machine maintains a connection and synchronises with the web application via the message broker. Besides that, Farmbot communicates with the Arduino firmware system over a serial connection to send commands for motor and tool positioning and receive collected from sensors and rotary encoders. Moreover, the FarmBot is connected to a monitoring system through a webcam to provide real-time view, image-capturing, and detecting weeds, taking actions according to it. Furthermore, the system provides an IT staff interface in case of loggings, errors and customary configurations to help customers.

Refer to *(Clause 9.6.4, 9.5.4.1)*

1.3.1.1 System Interfaces

- **Monitoring System:** In the FarmBot Express, the camera is provided within the kit. The kit is connected to Raspberry Pi through a USB cable or serial connection. The monitoring system receives data from the camera and using built-in computer vision functions, it detects the weeds and layout of the garden in general. According to the status of the garden, it takes actions autonomously or alerts the customer through the web application using the MQTT Gateway service. Moreover, it provides a real-time view or image-capturing from recorded data.

- **Arduino Firmware System:** The system consists of two subsystems named Mechanical Control System and Sensor Management System. 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. The processed data returns as a command from the main system to position tool heads to set light, sow weeds, and water plants.

- **Mechanical Control System:** This system is primarily constructed from V-slot aluminium extrusions and aluminium plates and brackets. They are driven by four NEMA 17 stepper motors, the Farmduino Express microcontroller, and a Raspberry Pi Zero 2 W computer. The bots roll directly on the wood-raised bed. FarmBot Express features a three-in-one tool head and does not require any additional precision tool. The gantry allows the tool head to be moved in the X-direction and serves as a guide for the Y and Z directions. The gantry also supports the structure for mounting the electronics box and seed troughs.

- **Sensor Management System:** This system has soil moisture, temperature, light, pH, humidity, and weather sensors. The abrupt measurements are sent to the Farmduino Express microcontroller. The firmware and Raspberry Pi computers process data and take actions autonomously or if it requires further investigation, it sends messages to the web server.

- **Web Application:** This system allows users to control and configure their FarmBot from most electronic devices. It does not require installation since it is a web-based application. The core features include the farm designer, event scheduler, sequence editor, regimen editor, controls, management of FarmBot's tools and seed containers, management of FarmBot's settings, storage and viewing of logs and sensor readings. Moreover, the application cloud is connected to the crop database.

- **Database Management System:** This system serves as a comprehensive repository of information about botanics and personal gardens. The general built-in data details are planting information, growing requirements, watering needs, nutritional needs, companion planting, pest and disease management, and harvesting and maintenance. Overall, the crop database is a valuable resource for users, so

that they do not have to search for plants or to be experts.

1.3.1.2 User Interfaces

FarmBot Express users and IT staff interact with the system primarily through a web-based application accessible via computer and mobile device browsers. The interface facilitates various functionalities essential for managing and monitoring farm operations. Furthermore; embedded operating system of the FarmBot Express is also available as open-source code. So a customer or an IT member can access the operating system code and configure the embedded system of the FarmBot Express by using means of embedded microcontroller chip. The following parts constitute the user interface:

- **Customer Interface:** Users are required to create an account by providing necessary details such as name, email address, and password. After signing, users can log in using their credentials to access their accounts. Upon logging in, users are greeted with a dashboard displaying key information such as current garden layout, planted crops and real photos of the layout. The dashboard provides tabs of the farming operations that allow users to navigate to different sections of the application. These tabs can be grouped as below:
  - **Layout Screen:** Shows the layout of the garden. It shows real photos of the garden, harmful weeds, the spread area of the planted crops, condition of the soil etc. directly on the garden layout.
  - **Plants:** In this tab, user can view and manage all of the plants in the garden. Plants can be dragged and dropped to the garden layout to make a design.
  - **Weeds:** This tab stands for viewing and managing the weeds in the garden. Weeds that are detected by FarmBot Express appear in this tab. User can select which weeds to be removed and what is the period of the removal.
  - **Points:** This tab provides information about soil in the different point of the garden. User can select points to monitor soil condition.

- **Curves:** This tab provides information about water, spread and height curves for the plants. User can determine amount of water, spread and height of the plants in this tab.
- **Sequences:** In this tab, user can configure sequences by putting some steps in order like first plant the seed, water it and then take a photo of it.
- **Regimens:** This tab stands for scheduling sequences based on the age of the plants.
- **Events:** This tab provides an overview of schedulend sequences and regimens for the user.
- **Sensors:** This tab provides information about historical sensor readings. It also allows user to view current sensor readings and manage sensors.
- **Phoros:** In this tab, user can view and manage photos of the garden taken by FarmBot Express. It also allows user to calibrate camera and weed detection settings of the FarmBot Express.
- **Tools:** This tab allows user to view detailed information about and manage tools, seed containers, and slots in the garden.
- **Messages:** This tab includes messages and announcements for the user.
- **Help:** In this tab, user can see the documentation of the web-application and can get support from IT members.
- **Settings:** This tab is the settings tab for web-application. Mainly for configuring accounts of the user.
- **Contols:** This tab provides a controller interface for user to control movements of FarmBot Express manually.
- **Jobs and Logs:** In this tab, user can view running and completed jobs.
- **Connectivity:** This tab provides information about the connection between the web browser, FarmBot Express, and the FarmBot web-application servers.

Besides this web-application user interface, the web-application and the embedded operating system of the FarmBot Express are open-source. So advanced users can configure the application and operating system as they want. Last thing to mention is that, FarmBot provides a REST API to edit information such as sequences and the gardan layout without using web-application directly.

- **IT Staff Interface:** While there isn't an interface exclusively designed for IT staff within the FarmBot Express system, IT members have access to the source code of both the web-application and the embedded operating system of the FarmBot Express as they are open source. This access enables them to support users and perform configurations as needed.

1.3.1.3 Hardware Interfaces

The system requires a borescope camera, Raspberry Pi computer, Farmduino, several sensors, four motors, rotary encoders, Vacuum Pump, audio jack, micro SD Card, Wi-Fi radio antenna, Ethernet connector, several USB cables, UTM cable, power supply, LED light strip, water solenoid valve and the universal tool mount. The camera supports advanced computer vision technologies. The Raspberry Pi computer, Farmduino and many mechanical devices are usually connected through USB cables. To receive data and send commands, the vacuum pump, water solenoid valve, encoders, motors, and light strip are integrated into the Raspberry Pi. Another hardware requirement is an electronic device that has a connection to the web server. Hence the Ethernet connector or the Wi-fi radio antenna must be implemented into the Farmduino firmware. Overall, the Raspberry Pi serves as a communication node between the Web App and the Farmduino via message broker, MQTT gateway. This communication allows Raspberry Pi to get instructions, customary configurations, and modifications manually from the web app to the Farmduino and hence to other hardware products.

1.3.1.4 Software Interfaces

- **Database:** OpenFarm which is a free and open source database for farming and gardening knowledge is used for this service. This service stores and provides crop and growing information to the web app for a streamlined user experienced. Any information in the tabs mentioned in the User Interface stored in the database. This information includes data accumulated by the sensors of the FarmBot Express such as moisture of the soil, temperature; data obtained from the photos provided by the webcam of the FarmBot Express such as weeds in the garden; data related to planned garden layout; data of jobs and scheduled sequences and much more.
  - Name: OpenFarm
  - Version Number: Last stable version
  - Source: https://openfarm.cc/

1.3.1.5 Communication Interfaces

The communication protocol between FarmBot Express and the web application is established using the MQTT Gateway protocol. MQTT is a lightweight messaging protocol ideal for communication between constrained devices and network servers. MQTT operates on a publish-subscribe messaging pattern, allowing FarmBot Express and the web application to exchange messages asynchronously. Messages are published by either FarmBot Express or the web application and subscribed to by the respective counterpart to facilitate bidirectional communication. The message content includes commands, sensor readings, job logs, data obtained by the hardware and photos taken by the webcam.

1.3.1.6 Memory Constraints

The system's application is web-based, therefore there is no requirement to have storage in the user's device. On the other hand, the Farmbot Express is a microcontroller after

all, hence it has built-in preserved area storage for program and data memories. Hence there is a memory constraint to keep webcam images, recordings, and data of several sensors. The data should be sent to web cloud services before overwriting. Moreover, it is an open-source system, hence the functions, OS and more can be configured by customers which affects the preserved program memory.

1.3.1.7 Operations

The operations of the Farmbot Express system can be partitioned into:
User-Initiated Operations:

- Customising garden
- Planting the garden
- Mounting and dismounting tools
- Measuring soil moisture
- Performing actions on many plants
- Using Farmbot's buttons
- Scanning the garden for weeds
- Using the rotary tool

Periods of Interactive Operations and Periods of Unattended Operations:

- Optimised Events
- Automated Planting Schedule
- Scheduled Sowing
- Scheduled Watering
- Real-time Viewing

- Capturing Images
- Connecting Farmbot to the Internet
- Connecting Farmbot to the Web App

Data Processing Support Functions

- Sensor Data Analysis
- Image Processing
- Logging
- Rotary Encoder Data Analysis
- Wi-fi and Web App credentials configuration

Backup and Recovery Functions

-

1.3.2 System Functions

Refer to *(Clause 9.6.5, 9.5.4.2)*. If you want to add any figure or diagram, you can use a figure environment.In Figure 1.2

1.3.3 Stakeholder Characteristics

Refer to *(Clause 9.6.6, 9.5.4.3, 9.4.5)*

1.3.4 Limitations

- **Regulatory Policies:** The system's web app holds personal information, like name, email address, telephone number or credit card to store with the account. Moreover, it collects all FarmBot web application data (e.g. photos, garden layout, weeds), device, log, location pieces of information, and local storage. Hence the system must hold data as encrypted.

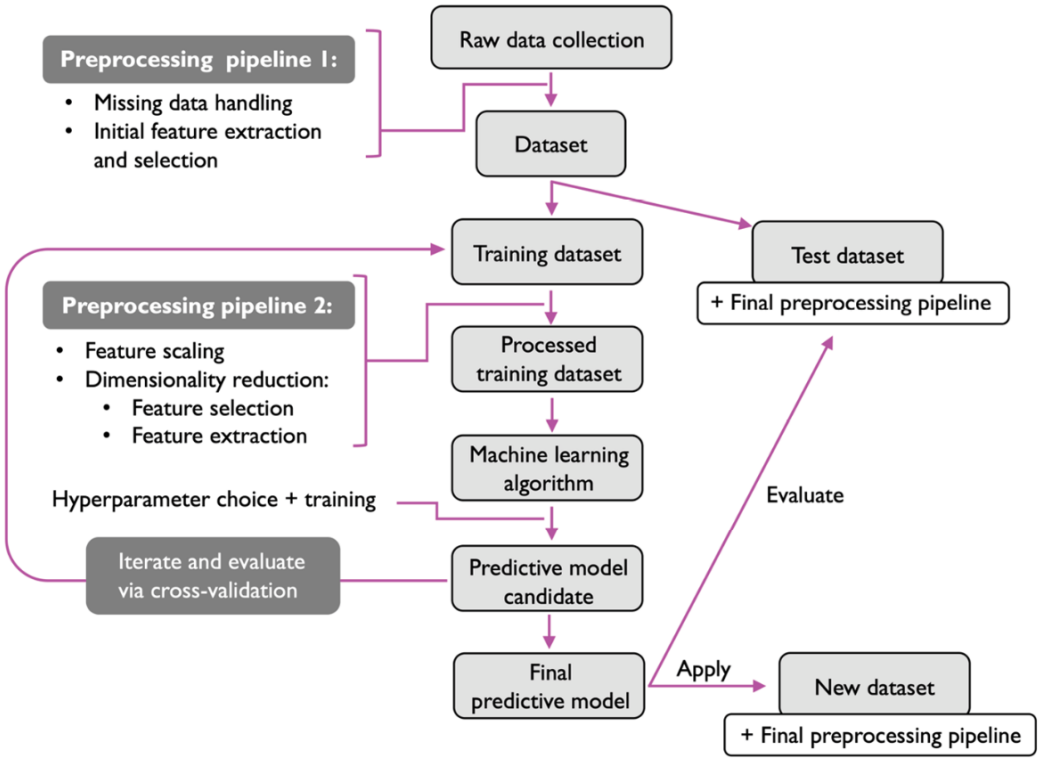

Figure 1.2: Example

- **Hardware Limitations:** The system is kept all day and night outside with the rain, sun, bugs, birds, dirt and different temperatures. Hence, it is suggested to inspect the hardware for loose screws, visible damage such as corrosion, bending, or cracking of components every three months. These are important to receive and process the correct image, sensor, and encoder data to send to the cloud and apply built-in or customary instructions.

- **Interfaces to other applications:** FarmBot Express must be compatible with web app services, Arduino firmware system, and monitoring system. Furthermore, the operating system and version of the customer's device must be compatible with FarmBot Express.

- **Parallel Operation:** FarmBot Express is a standalone system controlled by a web app registered by a particular account. Hence parallel operation is not required.

- **Audit Functions:** The system is an open-source, hence it can be configured by IT staff or customers. However, the inventory of weeds, water and the maintenance of the Arduino firmware system and the system's hardware shall be checked manually in person.
- **Control Functions:** The system has built-in advanced control-automation technologies. However, it is an open-source project and the user can customise and configure the system freely.
- **Higher-order Language Requirements:** The system and its Farmduino system are written in microprocessor-specific assembly. The system's firmware can be configured with C++, and its web app is written with Ruby on Rails, ReactJS, and TypeScript.
- **Signal Handshake Protocols:** The communication protocol between the FarmBot Express and its web services is satisfied by MQTT Gateway.
- **Quality Requirements:** The maintenance of the hardware, robots, bed, and sensors shall be checked physically regularly.
- **Criticality of the application:** The system is a single unit used by specific people. Hence the failure of the machine affects only one user.
- **Safety and security considerations:** The system's web app stores personal data. Hence the IT Staff shall check security regularly.
- **Physical/Mental considerations:** The system's web application does not require any coding and is easy to use. The built-in crop database provides information about Botanics and there is no physical action required after assembling the FarmBot Express. Hence physically disabled, young or old, expert or not expert, anyone can use it.

1.4 Definitions

You should add acronyms and abbreviations here. Refer to *(Clause 9.6.7)*

For any citation, refer to it as [1].

2. References

[1] M. C. Younis and H. Abuhammad, "A hybrid fusion framework to multi-modal biometric identification," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 25799–25822, 2021.

3. Specific Requirements

Refer to *(Clause 9.6.10)*.

3.1 External Interfaces

External Interfaces Class Diagram and its explanations go here. Plus, other content as appropriate

Refer to *(Clause 9.6.11, 9.5.8)*.

3.2 Functions

Use-case diagram goes here; **detailed use-case descriptions in a reasonable template** follow. You are expected to have **about 10 use cases covering major system functionality**. Have some associations in your use-case diagram, e.g. include, extend, specialization. Choose *three* most complicated use cases. Construct three diagrams **(one sequence diagram, one activity diagram, and one state diagram)** to elaborate on these three use cases. Plus, other content as appropriate.

Refer to *(Clause 9.6.1)*.

3.3 Logical Database Requirements

Key data objects (persistent or not) and their major attributes. Draw the **Class Diagram** with associations. A class dictionary can be omitted, provided that the naming is understandable.

Refer to *(Clause 9.6.15)*.

3.4 Design Constraints

- **Environmental Regulations:** FarmBot Express is designed for agricultural applications, including the cultivation of edible plants. Therefore; one important rule FarmBot hardware must follow is about not using harmful materials. This helps protect people and the environment from dangerous substances. This way, FarmBot can be safe for farmers to use and prevent harmful chemicals from getting eat or getting into the soil or water. The ecological footprint shall be kept as small as possible.

- **Safety Standards:** The design must comply with safety standards, ensuring the safety of operators, bystanders, and the environment. This includes features like emergency stop mechanisms, collision avoidance, and fail-safe operation.

- **Resource Constraints:** There are constraints which are rooted from the resources that potential users have, such as availability of skilled personnel who can manage hardware, and technological infrastructure which is required for system deployment such as electricity and wifi.

- **Energy Efficiency:** FarmBot shall be designed with energy efficiency in mind, utilizing low-power components and optimizing algorithms to minimize energy consumption during operation.

- **Affordability and Accessibility:** If FarmBot is intended for small-scale or resource-constrained farmers, the system shall be affordable and accessible, with consideration for cost-effective design solutions.

3.5 System Quality Attributes

3.5.1 Usability Requirements

- **Requirement 2:** FarmBot Express shall precisely use the tools, keeping sensitivity and needs of crops and soil in mind. For example, it shall precisely dispense water to crops, ensuring enough coverage and adequate (neither more nor less) hydration for their health.
- **Requirement 3:** FarmBot shall adjust its actions dynamically to accommodate variations in sensor data. Therefore; it can meet the needs of crops and weeds more accurately.
- **Requirement 4:** FarmBot Express shall follow commands given by the user in sequences, executing tasks according to the specified sequence without any deviation or interruption in normal circumstances (except user interruption or emergency stop).
- **Requirement 7:** FarmBot Express shall generate and transmit logs to the user interface upon completion of each job or command, providing detailed information about the executed tasks.
- **Requirement 8:** FarmBot shall provide feedback on task completion status, indicating when each task is initiated, in progress, or successfully completed.
- **Requirement 9:** The system shall provide intuitive user interface for both novice and experienced users. It should be kept in mind that users may be people who are far from technology.
- **Requirement 10:** The system shall support multiple languages for user interfaces to cater to international users.
- **Requirement 11:** The system shall minimize training length by presenting information in a clear and organized manner.

- **Requirement 12:** Users shall be able to customize settings and preferences to tailor the system to their needs.
- **Requirement 27:** Clear documentation, including user manuals, shall be provided to assist users in troubleshooting and customization. This simplifies user training.
- **Requirement 34:** Peripherals of the FarmBot Express shall be readily available from authorized vendors or through an official supply chain to minimize effort to find them.
- **Requirement 35:** The hardware design shall incorporate fault-tolerant mechanisms to mitigate the impact of component failures. Fault tolerance is important as the hardware works outside and stays in harsh conditions when necessary.
- **Requirement 40:** The hardware interfaces, including buttons and switches, shall be ergonomically designed and labeled clearly for intuitive operation by users. Replicates of the hardware interfaces shall be shown in user interface if necessary.
- **Requirement 42:** The hardware assembly shall be designed to minimize sharp edges or protrusions that could cause injury to users during installation or maintenance. This requirement also prevents animals and plants from being injured.
- **Requirement 51:** FarmBot shall detect and alert users to any errors or deviations from expected task outcomes such as rotary tool stall.
- **Requirement 52:** FarmBot shall provide clear and comprehensible error handling messages or feedback, preferably including suggestions and solutions, to assist users in resolving issues efficiently and resuming operations.

3.5.2 Performance Requirements

- **Requirement 1:** FarmBot Express shall accurately move to locations at specified coordinations within a tolerance of ±1mm (the difference between two coordination points).

- **Requirement 5:** FarmBot Express shall execute scheduled tasks, such as watering crops, at the specified times with a deviation tolerance of no more than ±1 minute from the scheduled time.

- **Requirement 6:** FarmBot Express shall repeat specified tasks, such as removing weeds, the designated number of times as instructed by the user, with no tolerance.

- **Requirement 13:** The system shall support a full garden configuration with crops, weeds and points simultaneously without degradation of performance. The grid of the FarmBot Express is 1.2 meters to 3 meters. So it can hold maximum of around 100 plants.

- **Requirement 14:** Response time of the system shall be less than 1 seconds under normal connection conditions. It is important for critical functions such as emergency stop.

- **Requirement 15:** The system shall accurately identify the crops and weeds grown in the garden. Therefore: more appropriate maintenance can be performed and unwanted losses can be prevented.

- **Requirement 43:** The webcam shall provide high-resolution images to allow FarmBot to process images, detect crops and weeds, and differantiate plants. Additioanlly; this allows to monitor their garden remotely.

- **Requirement 44:**The webcam shall have adjustable angles and calibration settings to provide users with clear views of the farming environment.

- **Requirement 45:** Sensors integrated into FarmBot Express shall provide accurate measurements within a reasonable tolerance of the actual value under normal operating conditions Accurate sensor measurements are essential for precise control of agricultural parameters such as soil moisture.

3.5.3 Reliability Requirements

- **Requirement 16:** The system shall efficiently manage memory and system resources of its embedded hardware to prevent system crashes.
- **Requirement 17:** The system shall have a mean time between failures (MTBF) of at least 336 hours (2 weeks). This requirement aims to ensure the system's reliability during extended periods of operation without supervision, such as during holidays, covering approximately 90 percent of typical holiday durations.
- **Requirement 25:** Communication between the user interface and hardware of FarmBot shall be seamless, even in harsh conditions. Any connection loss or data loss during transactions shall be promptly reported to the user to ensure accurate work.This requirement ensures correct operation and accuracy of FarmBot.

3.5.4 Availibility Requirements

- **Requirement 20**: The user interface shall guarantee 98% uptime excluding scheduled maintenance windows. Because user interface is crucial for FarmBot Express to implement its regular tasks.
- **Requirement 36:** The FarmBot Express system shall require the presence of trained maintenance staff available for on-site assistance and support.

3.5.5 Security Requirements

- **Requirement 21:** User authentication shall be enforced for accessing sensitive functionalities or data. This protects the user garden from external threats.
- **Requirement 22:** Passwords shall be securely stored using encryption techniques to prevent unauthorized access. This protexts the user gard from externak threats.

- **Requirement 23:** The system shall implement emergency actions to prevent hazards during hardware operations. This requirement is critical to ensure the safety of personnel, crops, animals and the environment.

- **Requirement 24:** The system shall log and audit user activities, including login attempts, command executions, and configuration changes, for security monitoring and forensic analysis.

3.5.6 Maintability Requirements

- **Requirement 26:** The FarmBot web application code and operating system code shall be modularized and well-documented to facilitate easy maintenance and future enhancements. This speeds up the development and update process of the system.

- **Requirement 27:** Clear documentation, including technical guides, shall be provided to developers in troubleshooting and development. This simplifies troubleshooting and development.

- **Requirement 28:** The hardware components shall be modularized to allow for easy replacement and upgrades without requiring extensive disassembly. This makes the moving and shipping processes safer.

- **Requirement 29:** Dependency of the modules to other modules of the embedded system shall be as small as possible, allowing for independent testing and maintenance.

- **Requirement 30:** Standard components shall be preferred in the hardware design to facilitate easy replacement and availability.

- **Requirement 31:** Comprehensive documentation shall be provided for all hardware components, including specifications, schematics, and assembly instructions.

- **Requirement 33:** Critical hardware components such as motors and sensors shall be designed for easy replacement by end-users, with minimal tools and technical expertise required.

- **Requirement 50** The FarmBot Express system shall be designed for easy disassamble to accommodate changes in farm layout. As farm layout may wear out or be damaged easily, faster re-assamble provides convenience.

3.5.7 Portability Requirements

- **Requirement 32:** The assembly of FarmBot Express shall be designed to be easy and require a minimal number of tools .This requirement aims to reduce the time, effort and number of tools required for assembly.

- **Requirement 39:** The FarmBot interface shall be responsive and adaptive, providing consistent user experiences across various screen sizes and resolutions. This is important to provide a user interface available from different devices.

- **Requirement 38:** The system shall support standard communication protocols, such as MQTT or RESTful APIs, to enable interoperability with third-party devices and services.

- **Requirement 39:** Installation and update procedures shall be automated (if possible) and well-documented to simplify deployment on different platforms and systems.

- **Requirement 48:** The webcam integrated into FarmBot Express must be designed to be replaceable, allowing for easy upgradeability of the system.As technology evolves, there may arise the need to upgrade the webcam component of FarmBot Express to take advantage of improved camera capabilities.

3.6 Supporting Information

Despite the many functionalities it offers, FarmBot's main purpose is not large-scale agriculture.The initiative to introduce the FarmBot technology aims to increase interest in gardening and plant cultivation. Having a Farmbot would enable groups to increase interest in the idea of gardening and raising plants in a number of ways. Through gamification students would learn the best way to place plants and discover the needs of plants and how to observe and check on those needs. Additionally the Farmbot would give the opportunity to teach additional skills such as robotics and computer programming. Users, particularly students, can learn fundamental principles of robotics and automation by programming FarmBot to perform specific tasks such as planting, watering, and harvesting.

4. Suggestions to Improve The Existing System

4.1 System Perspective

Context diagram and explanations of context diagram go here for suggestions to improve the existing system. Plus, other content as appropriate.

Refer to *(Clause 9.6.4, 9.5.4.1)*.

4.2 External Interfaces

External Interfaces Class Diagram and its explanations go here for suggestions to improve the existing system. Plus, other content as appropriate

Refer to *(Clause 9.6.11, 9.5.8)*.

4.3 Functions

Use-case diagram for suggestions to improve the existing system goes here; **detailed use-case descriptions in a reasonable template** follow. You are expected to have **about 4 use cases covering suggestions to improve the existing system**. Have some associations in your use-case diagram, e.g. include, extend, specialization. Choose three most complicated use cases. Construct three diagrams **(one sequence diagram, one activity diagram, and one state diagram)** to elaborate on these three use

cases. Plus, other content as appropriate.

Refer to *(Clause 9.6.12, 9.5.5, 9.5.10)*.

4.4 Logical Database Requirements

Key data objects (persistent or not) and their major attributes for suggestions to improve the existing system. Draw the **Class Diagram** with associations. A class dictionary can be omitted, provided that the naming is understandable.

Refer to *(Clause 9.6.15)*.

4.5 Design Constraints

Specify constraints on the system design imposed by external factors, such as official standards, regulatory requirements, or organizational/managerial limitations for suggestions to improve the existing system.

Refer to *(Clause 9.6.16)*.

4.6 System Quality Attributes

- **Requirement 18 Suggestion:** Automated backup mechanisms shall be implemented to prevent data loss in the event of system failure.
- **Requirement 19 Suggestion:** The system shall be capable of recovering from faults or errors without manual intervention.
- **Requirement 37: Suggestion** The FarmBot software shall be platform-independent and compatible with major operating systems, including Windows, macOS distributions.
- **Reqirement 41: Suggestion?** Sensors and actuators shall provide feedback to users through visual indicators or audible signals to indicate their status or operation.

- **Requirement 46: Suggestion:** Sensors shall be calibrated regularly to maintain accuracy and reliability over time, with calibration intervals not exceeding [specify duration].

- **Requirement 47: Suggestion** FarmBot Express hardware shall be equipped with a locking and alarm mechanism to prevent theft of its components. Theft prevention is essential as it resides outside.

Important quality attributes (Usability *(Clause 9.6.13, 9.5.6)*, Performance *(Clause 9.6.14, 9.5.7)*, Dependability properties, Maintainability, and so on) in the order of priority with associated requirements for the improved system.

Refer to *(Clause 9.6.18)*.

4.7 Supporting Information

Refer to *(Clause 9.6.20)*.