SOFTWARE ARCHITECTURE DESCRIPTION

Farm Bot

CENG350 Software Engineering

GROUP-1

Muhammet Akyüz - **2521185**
Alkım Doğan - **2521482**

Contents

List of Figures

List of Tables

Revision History

Version	Date	Explanation
1	10.05.2024	Part 1 completed
2	16.05.2024	Some parts are fixed according to the feedback.

Table - 1 Revision History

1 Introduction

1.1 Purpose and Objectives of FarmBot

The purpose of the CNC system is to create an open and accessible technology that helps people (possible farmers) about grain, farming and growing plants and crops. This efficiency is achieved by a remote farming bot that can be controlled via web application. It also aims to help education, research related to farming, as well as home use for ordinary people.

1.2 Scope

The main scope of the FarmBot is to provide reliable, fast, efficient and cheap crop growing at the same time, provided the bot is properly installed on the corresponding area. To be specific, there is no one type of FarmBot. There are various of FarmBots. Each has different features with different size and cost.

- FarmBot will maintain an automated planting, seeding, weeding, maintenance, watering, crop monitoring and crop rotation.
- There are a lot of sensors that are attached to bot. The bot has a camera, soil sensors and a watering tool for the both data and maintenance purposes.
- FarmBot collects data on various aspects of crop growth and environmental conditions, which allows users to analyze trends, identify patterns, and optimize the farming practices.
- There is a graphically designed application with a game-like interface to help users grow crops easily via an interface map.
- The system will provide a flexible behaviour. The FarmBot can be configured and customize easily. All can be done with changing the basic parameters, meaning there is no coding knowledge required.
- There will be an extensive software documentation with how-to guides, descriptions of every feature and pro-tips.

1.3 Stakeholders and Their Concerns

- **Customers:** Customers include mainly farmers whose purpose is to make profit out of growing crops as well as ordinary users who make use of FarmBot for their personal space and purposes.

- **Developers:** Developers are those supporting the open source project available on GitHub. They can integrate new features, fix the current bugs or make a configuration as other stakeholders give feedback. Their purpose is to main the system along with its requirements and specifications.

- **Education Partnerships:** Education stakeholders involves both educators and their students. Those education institutes have partnership with FarmBot. Educators can use FarmBot to show and educate their students about the process of growing crops. They can also suggest further improvements to FarmBot to make it suit their educational needs. Students can use FarmBot to understand the process of growing crops more effectively and intuitively.

  Universities and Institutions adapted Farmbot are The University of California Berkeley, Darmouth, Nasa, California State University, Illıonis State University, Liberty University, California Polytechnic State University, Virginia State University.

2 References

1] “Open-source CNC farming,” FarmBot, https://farm.bot/ (accessed Mar. 12, 2024).

The standarts are from:

ISO/IEC/IEEE International Standard Systems and software engineering Life cycle processes Requirements engineering. (2018). IEEE. https://doi.org/10.1109/ieeestd.2018.855968623

3 Glossary

AI	Artificial Intelligence
MQTT	Message Queuing Telemetry Transport
TCP	Transmission Control Protocol
TypeScript	A strongly typed programming language that builds on JavaScript
Raspbery Pi	The name of the microboard
Arduino	Open-source electronic prototyping platform
Elixir	Dynamic function language for building scalable and maintainable applications
Ruby	Dynamic open source programming language with focus on simplicity productivity
Nema 17 Stepper	A motor that has a 1.8 degree step angle with 1.7 x 1.7 inch faceplate
HTTP	Hyper Text Transfer Protocol, an applicaiton layer protocol
RabbitMQ	an open- source message-broker software that implements the Advanced Message Queuing Protocol (AMQP) and several other messaging protocols
STOMP	Simple Text Oriented Messaging Protocol

Table 1: Definitions Table

4 Architectural Views

4.1 Context View

4.1.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners.

A few materials are used to explain the context of the system, such as a context diagram, an external interfaces diagram, and an interaction scenarios diagram.

Stakeholders may make use of this viewpoint in order to understand different perspectives and interactions between the FarmBot and any kind of users. The users use this view to grow the crops faster, making their process easier. The developers use this view to plan the new features and making the project more effective. As of education partnerships, they use it to arrange experiments and conduct researches using the FarmBot.

4.1.2 Context Diagram

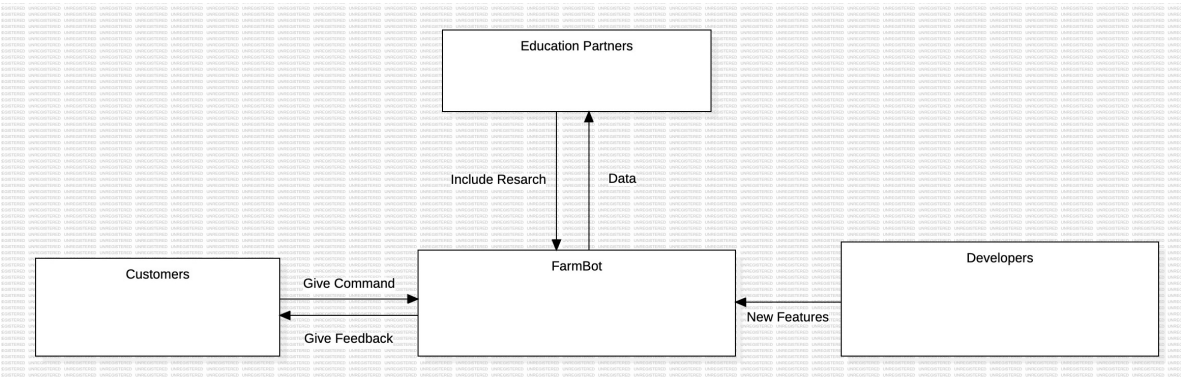

Figure 1: Context Diagram

- **Customers:** They can give command to FarmBot and request feedback from it.
- **Developers:** They can add new features to FarmBot to improve it.
- **Education Partners:** They can use FarmBot to conduct research by configurating it. They can also collect the data of the research.

4.1.3 External Interfaces

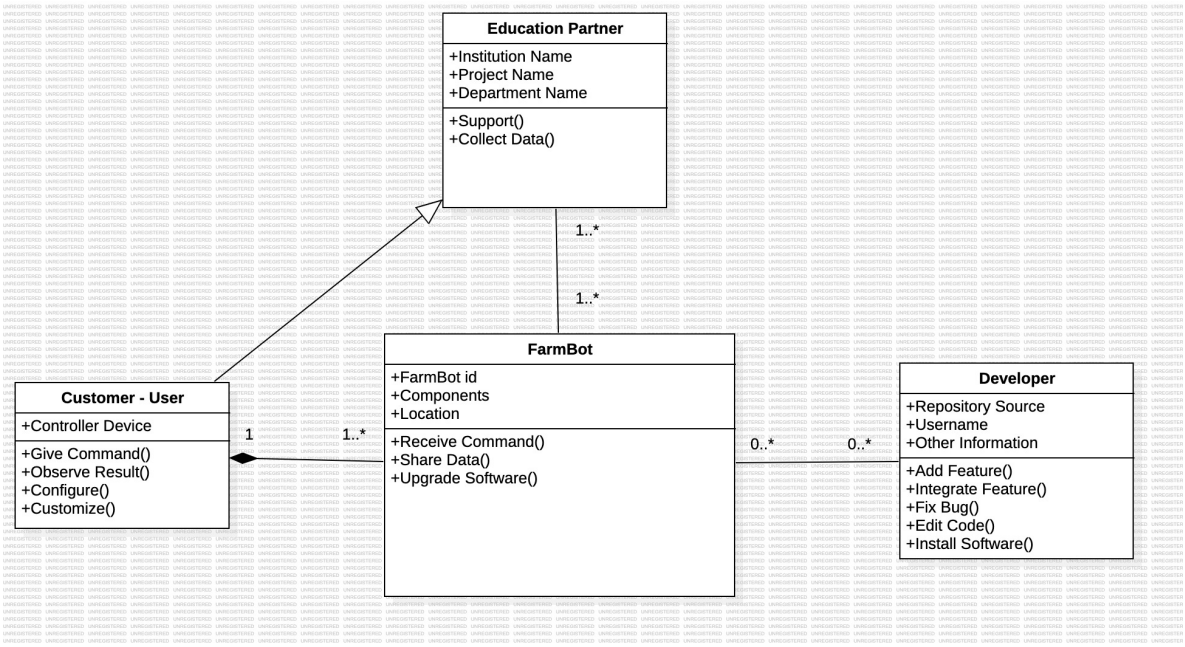

Figure 2: External Interfaces Diagram

Operation	**Description**
Receive Command	By analyzing the correct input section from the web app, FarmBot can receive the commands given by the customers or education partners
Share Data	By sending the information it gathered by its sensors to web app, Farmbot can share the data which it collected.
Upgrade Software	It configures and updates FarmBot according to the updates done by the developers.
Give Command	It enables users the give command to FarmBot.
Observe Result	Customers can use this function to analyze the data shared by the FarmBot to make the correct adjustments and control the functionality of it.
Configure	Customers can adjust the FarmBot according to their needs.
Customize	Customers can customize the FarmBot to make it more efficient and meet their needs.
Support	Education Partners can support the FarmBot by making observations about its functionality and making new experiments.
Collect Data	Collects data that FarmBot shared according to specification made by the Education Partners.
Add Feature	Enables developers to add new features to FarmBot to improve it.
Integrate Feature	By adding new technical parts or codes, developers can integrate the features they added.
Fix Bug	Developers can fix the bugs that occurred
Edit Code	Developers can edit codes to add new features or fix bugs.
Install Software	It installs software to FarmBots.

Table 2: External Interface Operation Descriptions

4.1.4 Interaction Scenarios

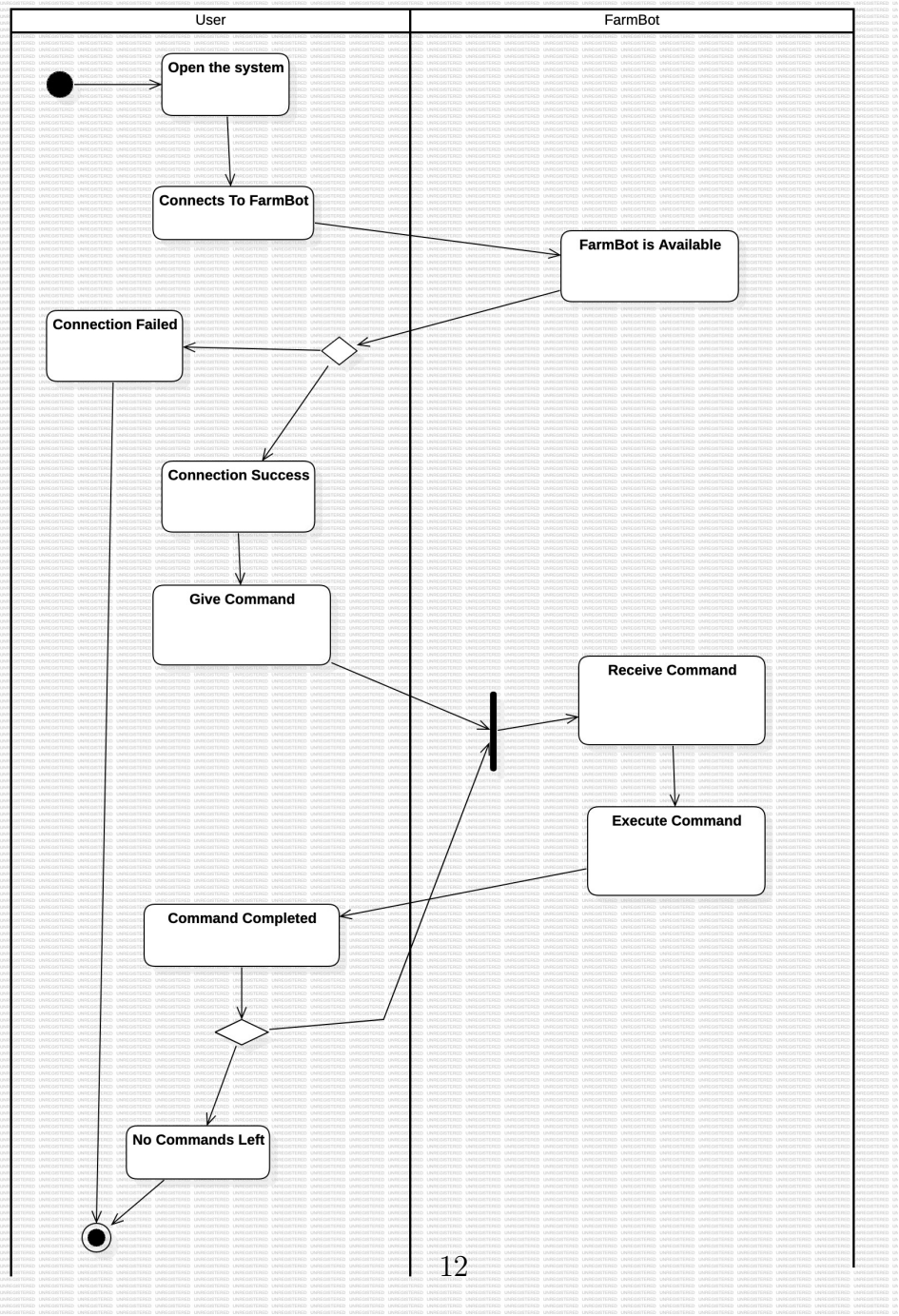

Figure 3: Give Command Activity Interfaces Diagram

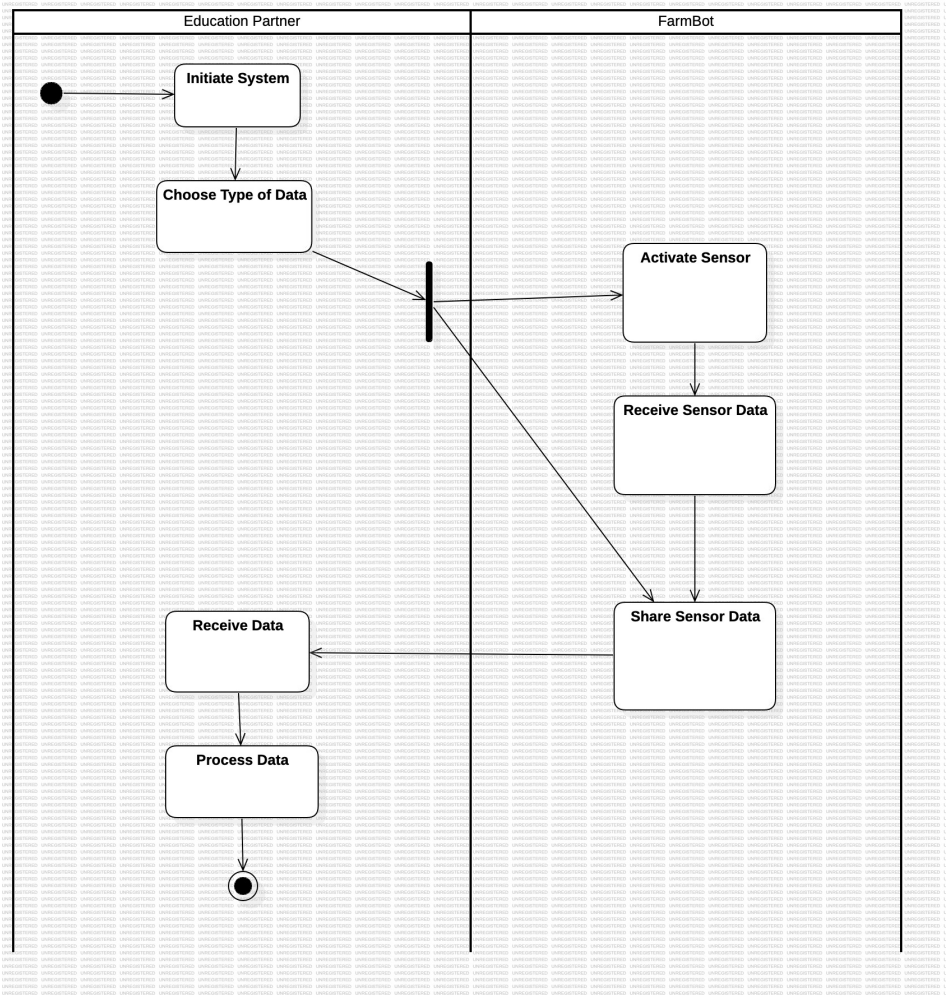

Figure 4: Collect Data Activity Diagram

4.2 Functional View

4.2.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners. Customers will use this view to understand the technical capabilities of the FarmBot, while education partners will use this view to generate new ideas for FarmBot, and also understand its functionality to use it in their education. The developers can utilize this view to understand functional design implemented in the FarmBot.

4.2.2 Component Diagram

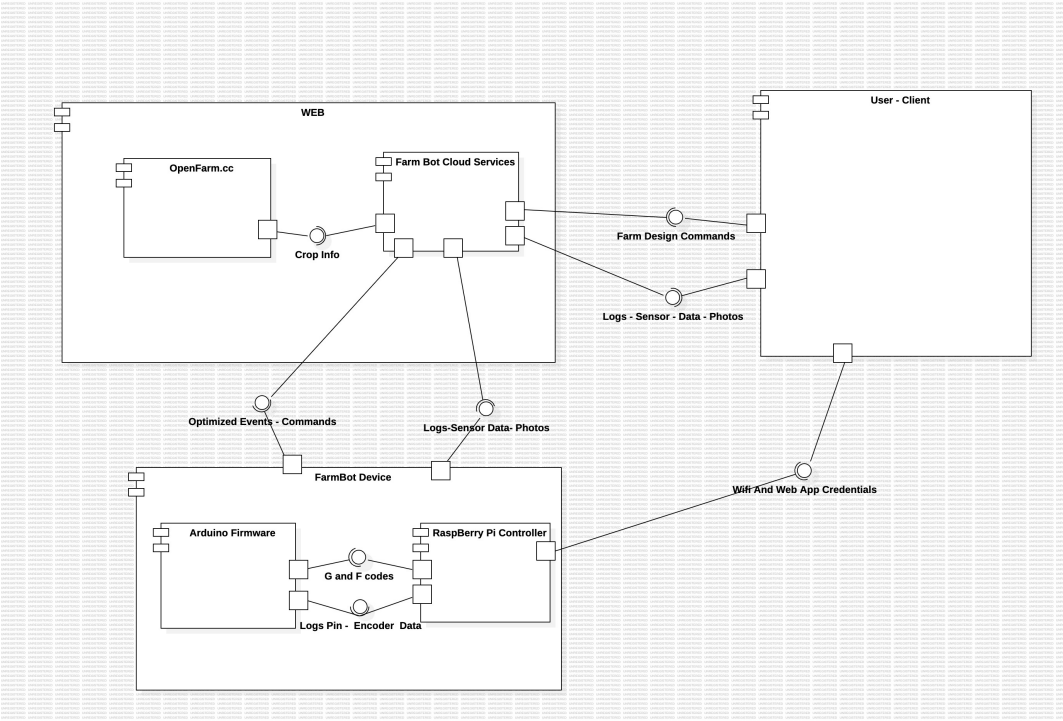

Figure 5: Collect Data Activity Diagram

- There are three subsystems which are Web, FarmBot device, and clients.
- FarmBot device consists of RaspBerry Pi Controller, and Arduino Firmware.
- Web consists of OpenFarm and FarmBot Cloud Services.
- Arduino Firmware can read sensors and write to tools. It can also control the stepper motors and rotary encoders.
- Arduino Firmware requires G and F codes interface which passes G and F codes to Arduino to make it work properly.
- RaspBerry Pi Controller can takes photos of the farm and process the data according to FarmBot configurations.
- RaspBerry Pi Controller needs Logs Pin - Encoder data interface in which Arduino Firmware provide the data it read to RaspBerry Pi.
- FarmBot device needs Logs- Sensor Data - Photos interface which provides the data collected by the Farmbot and the photos taken by the FarmBot device.
- FarmBot device needs Optimized Events- Commands interface where the Web provides it with new and optimized events and commands.
- User needs Farm Design Commands interface which provides Web with the users needs.
- User needs Logs- Sensor Data - Photos interface which provides users with the data and photos gathered by FarmBot.

4.2.3 Internal Interfaces

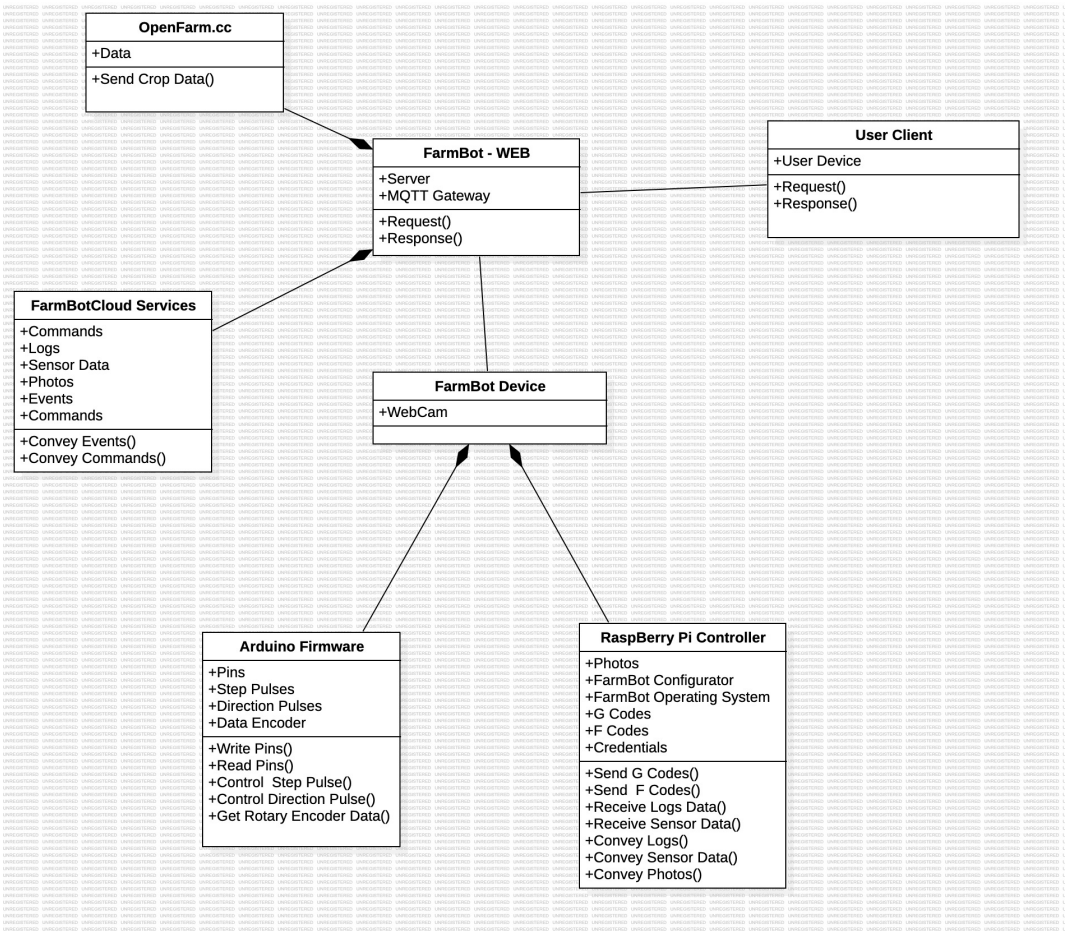

Figure 6: Internal Interfaces Diagram

Operation	**Description**
Send Crop Data	OpenFarm.cc database will provide the FarmBot-Web with the data it requested.
Request	Makes the needed request
Response	Response to the other components
Convey Events	Conveys the events that are requested
Convey Commands	Conveys the commands from components to other components
Write Pins	Writes the external pins of Arduino Firmware
Read Pins	Reads the external pins of Arduino Firmware
Control Step Pulse	Controls the step pulse of Arduino Firmware
Control Direction Pulse	Controls the direction of the pulses of Arduino Firmware
Get Rotary Encoder Data	Collects the data from the rotary encoder
Send G Codes	Sends the G codes from RaspBerry Pie Controller to Arduino Firmware
Send F Codes	Sends the F codes from RaspBerry Pie Controller to Arduino Firmware
Receive Logs Data	RaspBerry Pie Controller receive logs data from Arduino Firmware
Receive Sensor Data	RaspBerry Pie Controller receive sensor data from Arduino Firmware
Convey Logs	Conveys the logs from RaspBerry Pie Controller to FarmBot Web App
Convey Sensor Data	Conveys the sensor data from RaspBerry Pie Controller to FarmBot Web App
Convey Photos	Conveys the photos from RaspBerry Pie Controller to FarmBot Web App

Table 3: Internal Interface Operation Descriptions

4.2.4 Interaction Patterns

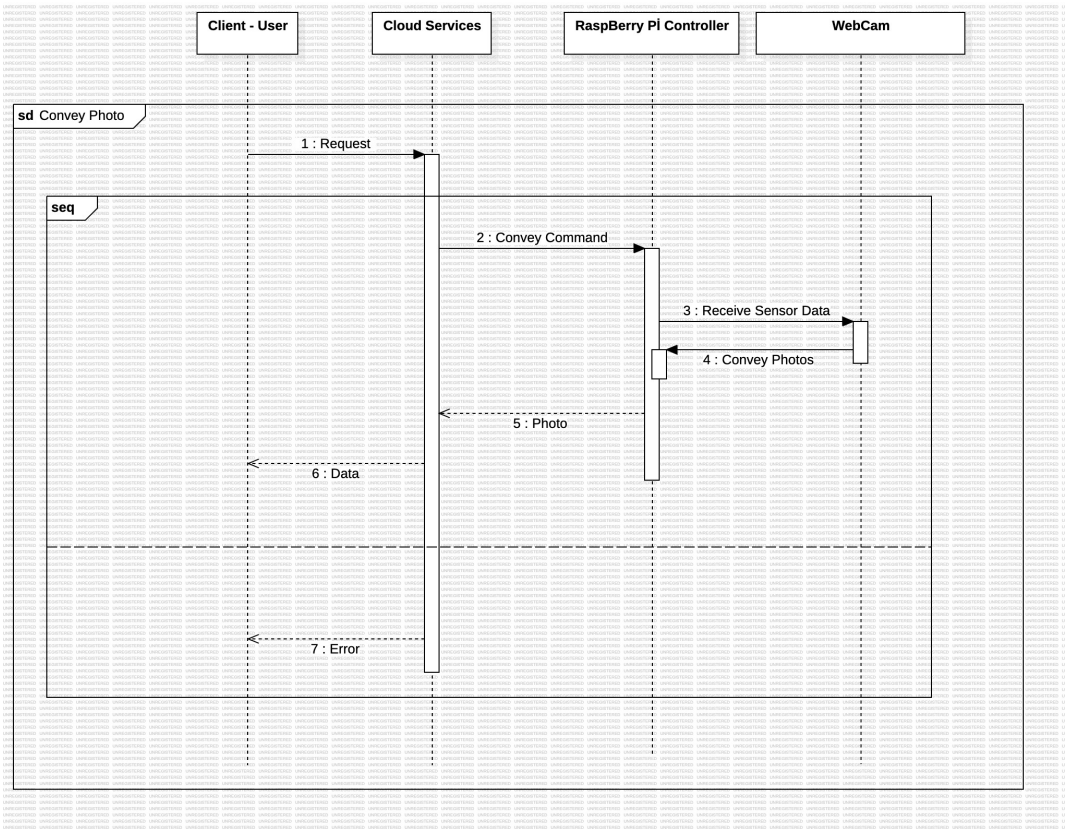

Figure 7: Convey Photos Sequence Diagram

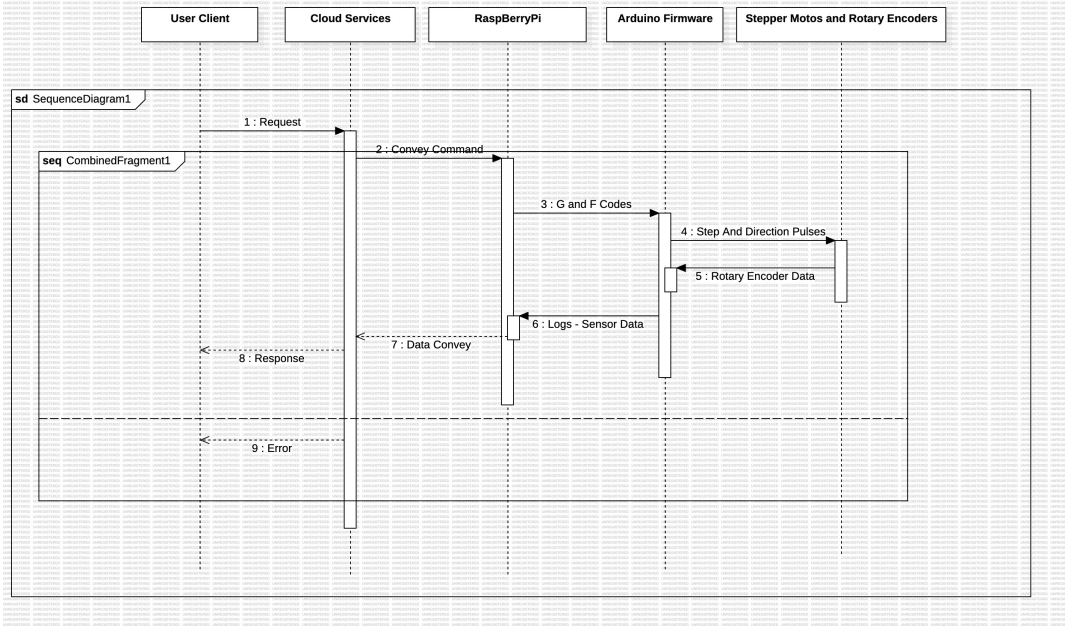

Figure 8: Get Rotary Encoder Data Sequence Diagram

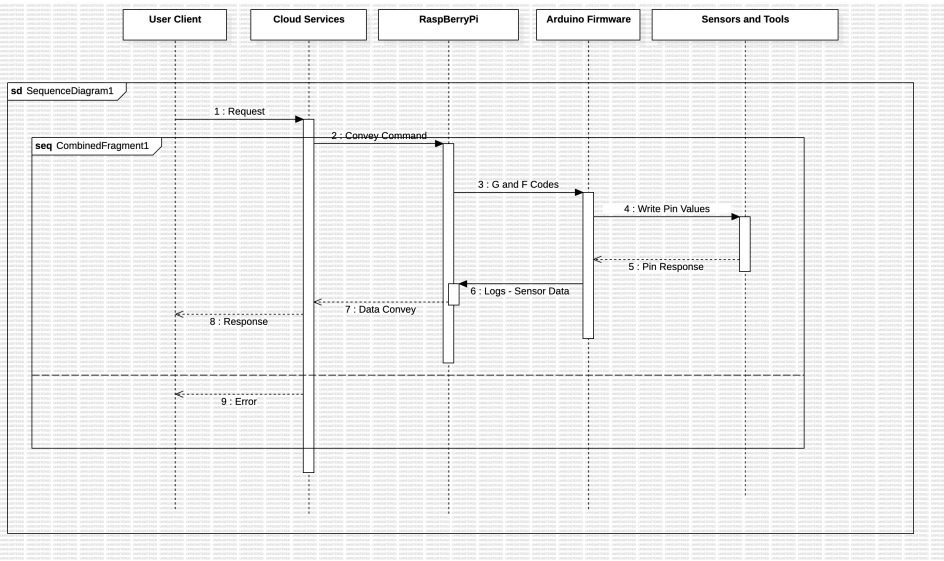

Figure 9: Write Pin Sequence Diagram

4.3 Information View

4.3.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners.Developers use this view to understand the strategies used within the process of data transfer, data modeling, and structuring of the data. Customers use this view to understand how to contribute information to the system or take information from the system. Educative partners utilize this view to understand how to improve the system as well as how to configure the system to meet their educational purposes.

4.3.2 Database Class Diagram

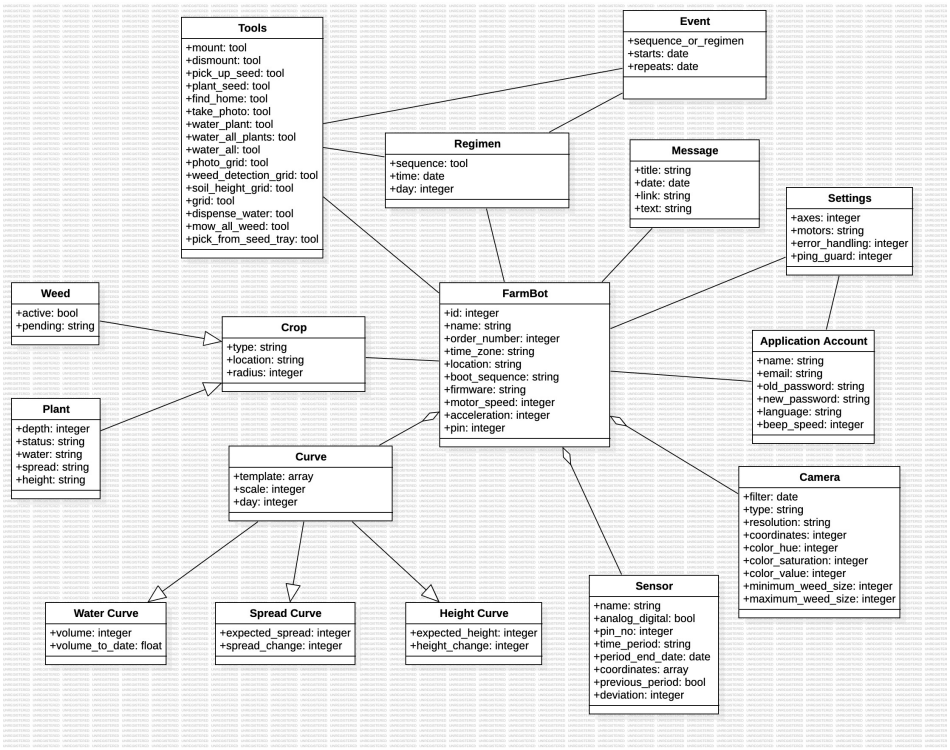

Figure 10: Logical Database Requirements

4.3.3 Operations on Data

Operation	**Description**
GetFarmBotId	Create: Read: Information Update: Delete:
ChangeName	Create: Read: Update: Information Delete:
GetTimeZone	Create: Read: Time Update: Delete:
ChangeLocation	Create: Read: Update: City Delete:
ArrangeMotorSpeed	Create: Read: Update: Float Delete:
ChangeAcceleration	Create: Read: Update: Float Delete:
ArrangePIN	Create: Read: Update: Information Delete:

Table 4: Internal Interface Operation Descriptions

GetLocation	Create: Read: Integer Update: Delete:
GetType	Create: Read: Information Update: Delete:
ArrangeRadius	Create: Read: Update: Integer Delete:
IsActive	Create: Read: Boolean Update: City Delete:
ArrangeDepth	Create: Read: Update: Integer Delete:
ChangeStatus	Create: Read: Update: String Delete:
GetHeigh	Create: Read: Information Update: Delete:

Table 5: Internal Interface Operation Descriptions

GetDay	Create: Read: Integer Update: Delete:
GetType	Create: Read: Information Update: Delete:
IsMounted	Create: Read: Boolean Update: Delete:
ChangeScale	Create: Read: Update: Integer Delete:
GetVolume	Create: Read: Integer Update: Delete:
GetVolumeToDate	Create: Read: Information Update: Delete:

Table 6: Internal Interface Operation Descriptions

GetExpectedSpread	Create: Read: Integer Update: Delete:
ChangeSpread	Create: Read: Update: Integer Delete:
GetExpectedHeight	Create: Read: Integer Update: Delete:
ChangeHeight	Create: Read: Update: Integer Delete:
GetSensorName	Create: Read: String Update: Delete:
GetCoordinates	Create: Read: Integers Update: Delete:

Table 7: Internal Interface Operation Descriptions

ChangeAccountName	Create: Read: Update: String Delete:
ChangeEmail	Create: Read: Update: String Delete:
ChangePassWord	Create: Read: Update: String Delete:
Create Password	Create: String Read: Update: Delete: Language
ChangeTitle	Create: Read: Update: String Delete:
ChangeDate	Create: Read: Update: Date Delete:

Table 8: Internal Interface Operation Descriptions

DeleteLink	Create: Read: Update: Delete: String
DeleteText	Create: Read: Update: Delete: String
AddLink	Create: Read: Update: String Delete:
AddText	Create: String Read: Update: Text Delete:
AddSequence	Create: String Read: Update: Delete:
SetDate	Create: Read: Update: Date Delete:

Table 9: Internal Interface Operation Descriptions

4.4 Deployment View

4.4.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners. Developers use this view to understand the technologies and strategies used within the deployment process. Customers use this view to understand the capabilities of the system. Educational partners use this view to improve and customize the technologies and strategies used within the deployment process for their educational purposes.

4.4.2 Deployment Diagram

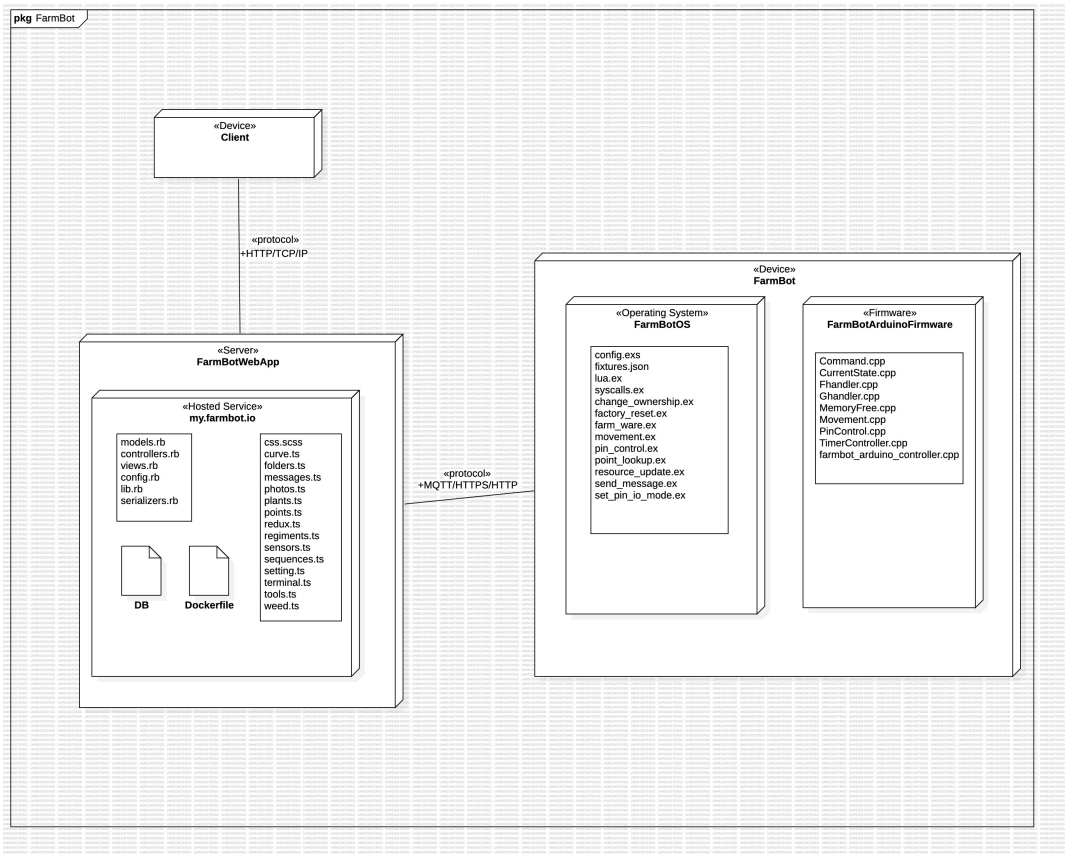

Figure 11: Deployment Diagram For FarmBot

- The device of the user, alternatively client, can be any device that can connect to internet, and use the FarmBot web application.
- Client device requests to FarmBot Web Application server by using HTTP protocol for application layer. This communication also uses TCP for transport layer and IP for network layer.

- The back-end part of the web application is mostly written in a programming lanuage called Ruby, which has the extension ".rb".
- Back-end architecture makes uses of MVC (Model-Controller-View) archite. Related files are coded in Ruby.
- There are database files that have the extension ".sql".
- PostgreSQL is used for the effective database queries. This can be seen in the configurations as "adapter: postgresql".
- There is also a Dockerfile for the usage of RabbitMQ. RabbitMQ is an open-source message-broker software that implements the Advanced Message Queuing Protocol (AMQP) and several other messaging protocols such as MQTT (Message Queuing Telemetry Transport) and STOMP (Simple Text Oriented Messaging Protocol).
- Front-end is widely written in TypeScript along with the related HTML/CSS.
- Communication between the Web Application and the FarmBot robot is also established via HTTPS/HTTP (insecure version of HTTPS).
- FarmBot operating system is coded in Elixir,a functional, concurrent programming language designed for building scalable and maintainable applications. Corresponding documents have the extension ".ex", which corresponds compiled code, and ".exs", which means scripting file.
- The firmware of FarmBot is coded in C++, a powerful, high-level programming language that offers a wide range of features suitable for developing a variety of application.

4.5 Design Rationale

4.5.1 Context View

In the context of design rationale, the context view of a project provides a high-level overview of the project's environment, stakeholders, goals, and constraints. It provides understanding for the broader context within which design decisions are made, and provides insights into the rationale behind those decisions.

4.5.2 Functional View

In the context of design rationale, the functional view of a project focuses on describing the functional requirements, capabilities, and interactions of the system being designed. It provides a detailed understanding of how the system will behave and what functionality it will offer to users. It also touches on scalability, modularization, requirements validation and verification concerns of the system.

4.5.3 Information View

In the context of design rationale, the information view of a project focuses on describing the information architecture, data models, and information flow within the system being designed. It provides view into how data is organized, stored, processed, and communicated within the system. . It also touches on consistency, interoperability, integration, security, and privacy concerns of the system.

4.5.4 Deployment View

In the context of design rationale, the deployment view of a project focuses on describing how the system is deployed, distributed, and managed within its operational environment. It provides view into the infrastructure, platforms, configurations, and deployment strategies used to make the system available to users. It also touches on scalability, performance, availability, fault tolerance, communication between components, and resource management factors of the system.

5 Architectural Views for Your Suggestions to Improve the Existing System

5.1 Context View

5.1.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners. A few materials are used to explain the context of the system, such as a context diagram, an external interfaces diagram, and an interaction scenarios diagram.

Stakeholders may make use of this viewpoint in order to understand different perspectives and interactions between the FarmBot and any kind of users. The users use this view to grow the crops faster, making their process easier. The developers use this view to plan the new features and making the project more effective. As of education partnerships, they use it to arrange experiments and conduct researches using the FarmBot.

5.1.2 Context Diagram

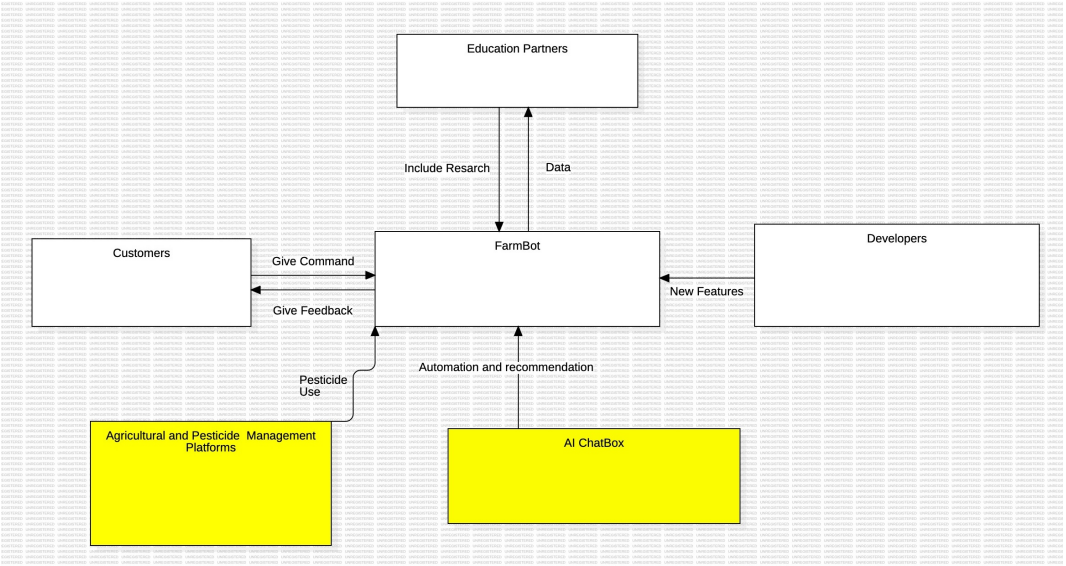

Figure 12: Context Diagram For Improvements

- **AI Tool :** The FarmBot can need some **external** artificial assistance both for users and itself to facilitate the work. Since most of the user are not qualified enough about the farming topics and how to grow to crops efficiently and correctly, they will need some guidance towards these topics. Moreover, FarmBot can take some assistance from AI to accomplish the tasks more quickly and efficiently. Providing this kind of different types of support can be done by some external AI Tool.

- **Agriculture and Pesticide Management Platforms :** To grow the plants more efficiently and to protect them from the harm that can be caused by harmful instincts and bacteria, FarmBot can be able to spray pesticide to the crops. By accessing official government websites, it can correctly choose most suitable pesticide for the farm. Agricultural and Pesticide Management Platforms interface can help the FarmBot regarding this topic.

5.1.3 External Interfaces

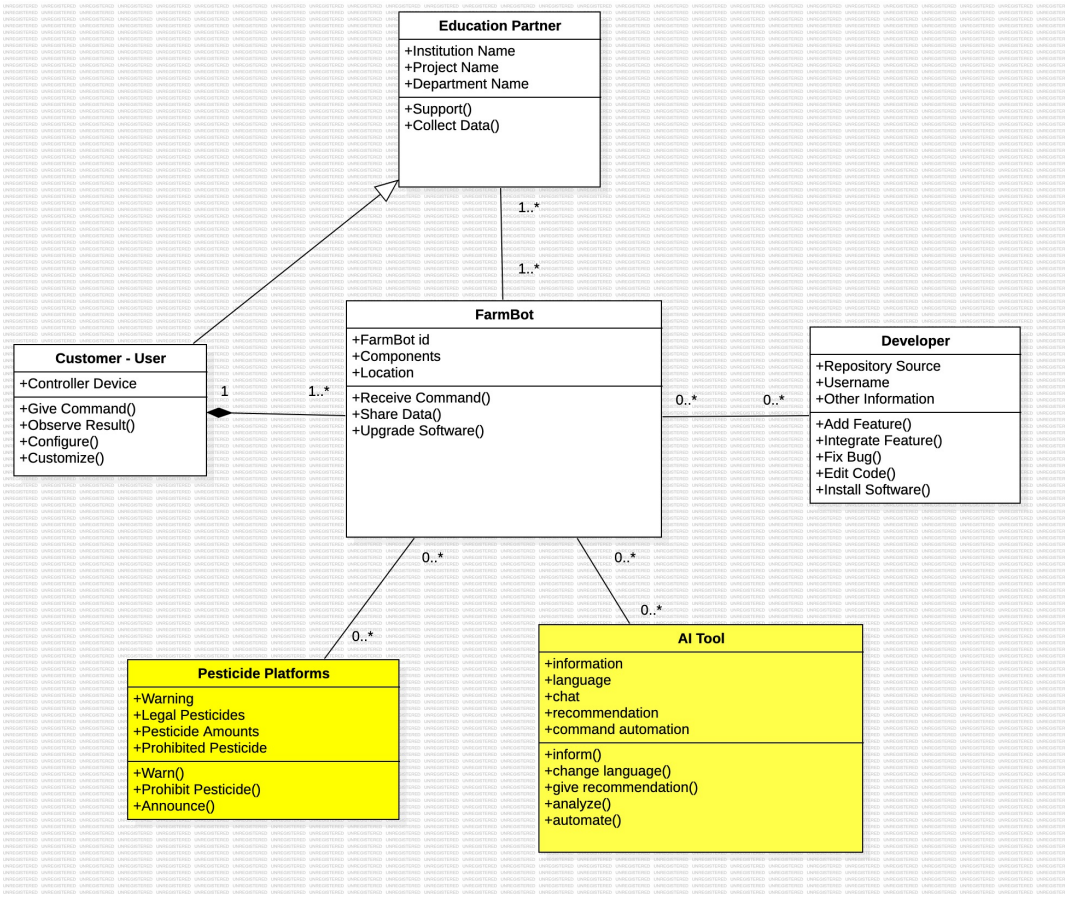

Figure 13: External Interface Diagram For Improvements

Operation	**Description**
Inform	By analyzing the current conditions or the questions it has been asked, AI Tool can generate respond and inform the user about these topics.
Change Language	User can change the language settings of the both AI Tool and FarmBot by using API.
Give Recommendation	According to the users' wishes and conditions of the farm, AI tool can give recommendations to the users to facilitate their work.
Analyze	Make the AI tool and FarmBot to analyze the current conditions of the farm.
Automate	Automates the trivial parts of the farm to facilitate users work
Warn	Warns the user about the incorrect or non-legal pesticide usage
Prohibit Pesticide	Prohibits and prevents the user from using illegal or out-dated pesticides
Announce	Announces about current regulations about pesticide usage

Table 10: Improved External Interface Operation Descriptions

5.1.4 Interaction scenarios

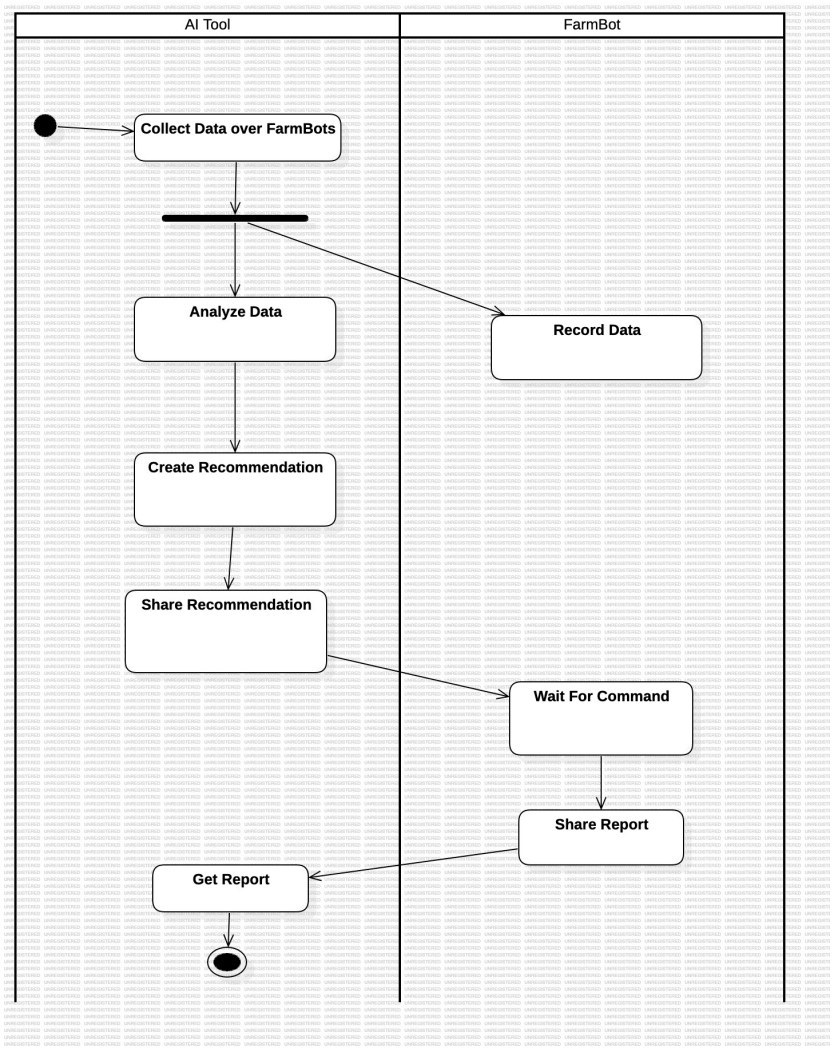

Figure 14: Activity Diagram Of AI ChatBox Interaction For Improvements

5.2 Functional View

5.2.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners. Customers will use this view to understand the technical capabilities of the FarmBot, while education partners will use this view to generate new ideas for FarmBot, and also understand its functionality to use it in their education. The developers can utilize this view to understand functional design implemented in the FarmBot.

5.2.2 Component Diagram

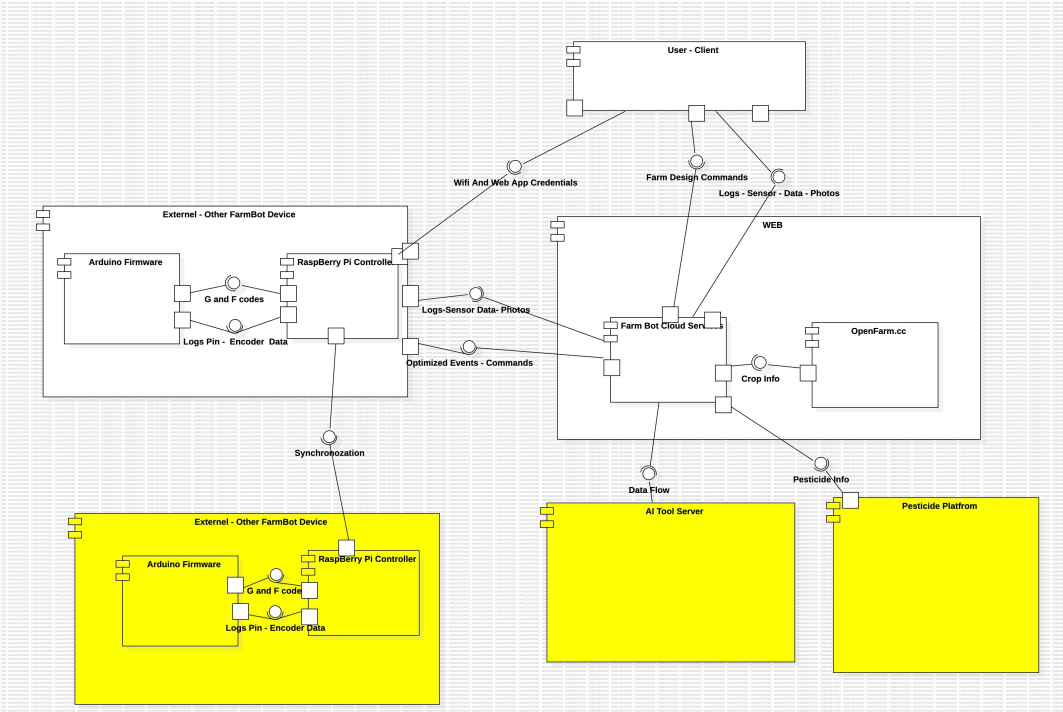

Figure 15: Component Diagram For Improvements

- FarmBots need synchronization interface to communicate with other FarmBot directly via hardware.

- AI Tool need Data Flow interface which enables the AI Tool to interact and communicate with the Web.
- Pesticide Platform need Pesticide Info interface which provides connection between the Pesticide Platform and Web.

5.2.3 Internal Interfaces

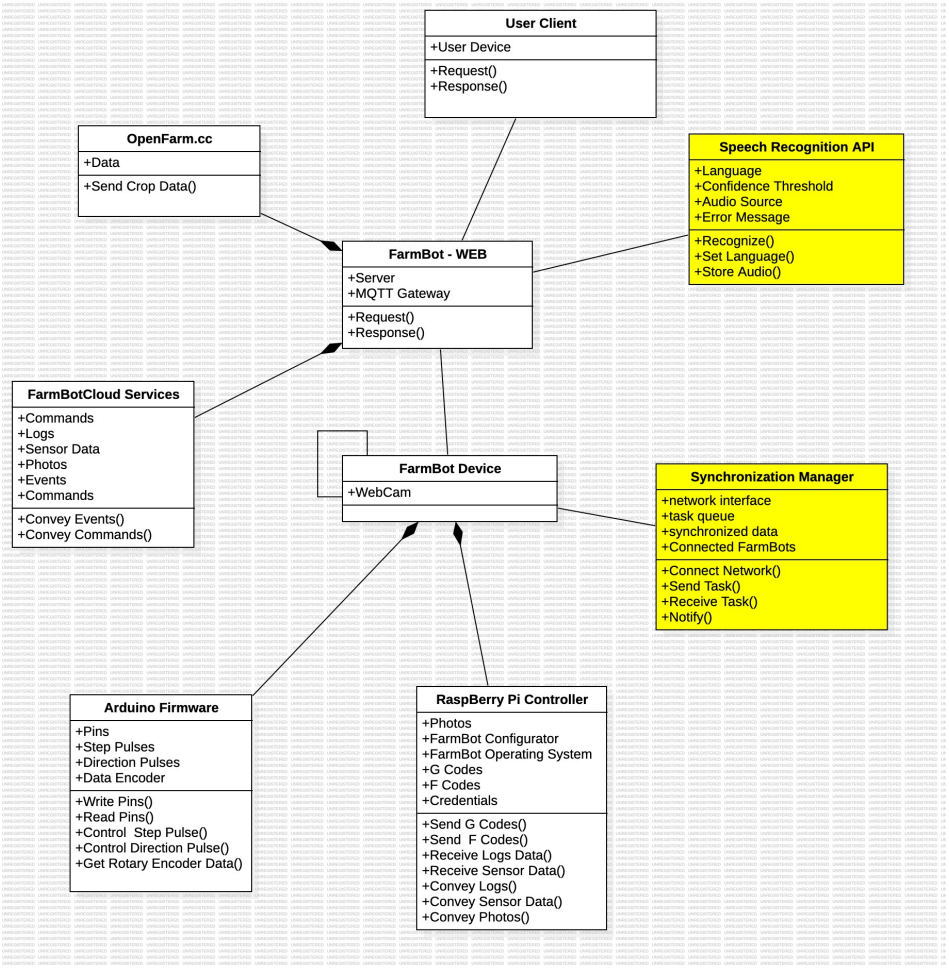

Figure 16: Internal Interface Class Diagram For Improvements

Operation	Description
Connect Network	Connects the FarmBot to the FarmBot network
Send Task	Sends tasks from a FarmBot to other FarmBots
Receive Task	Enables a FarmBot to receive tasks from other FarmBots
Notify	Notify FarmBots about available tasks
Recognize	Recognizes the voice command given by the user
Set Language	Sets the language
Store Audio	Stores the voice commands of the user

Table 11: Improved External Interface Operation Descriptions

5.2.4 Interaction Patterns

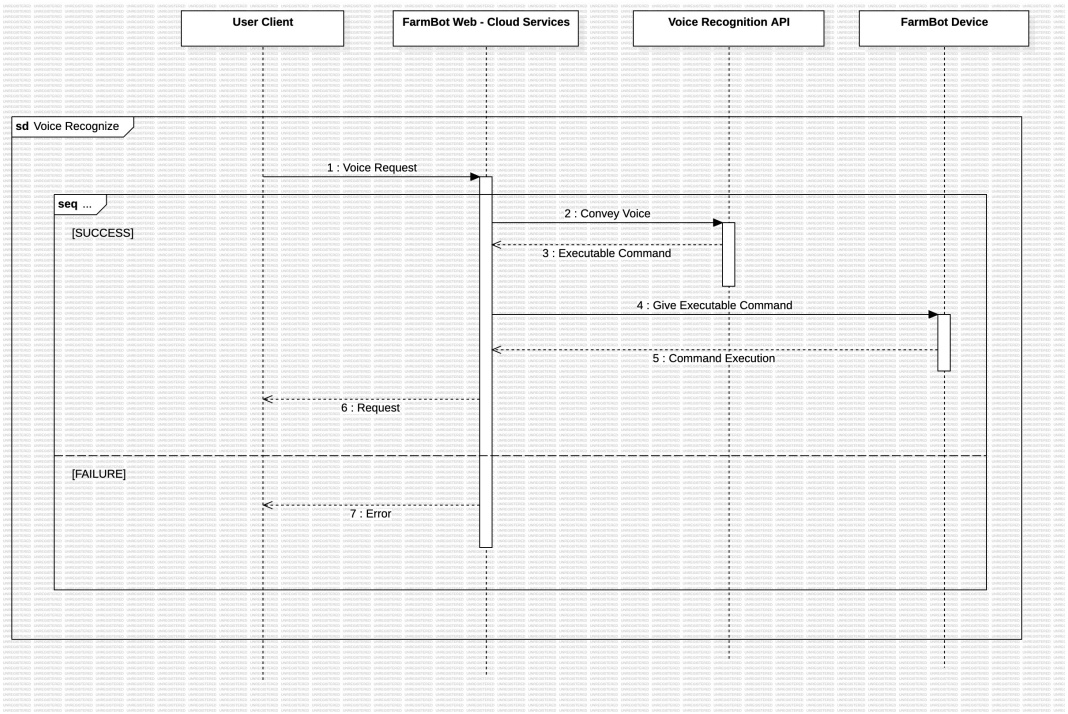

Figure 17: Voice Recognition Sequence Diagram For Improvements

5.3 Information View

5.3.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners.Developers use this view to understand the strategies used within the process of data transfer, data modeling, and structuring of the data. Customers use this view to understand how to contribute information to the system or take information from the system. Educative partners utilize this view to understand how to improve the system as well as how to configure the system to meet their educational purposes.

5.3.2 Database Class Diagram

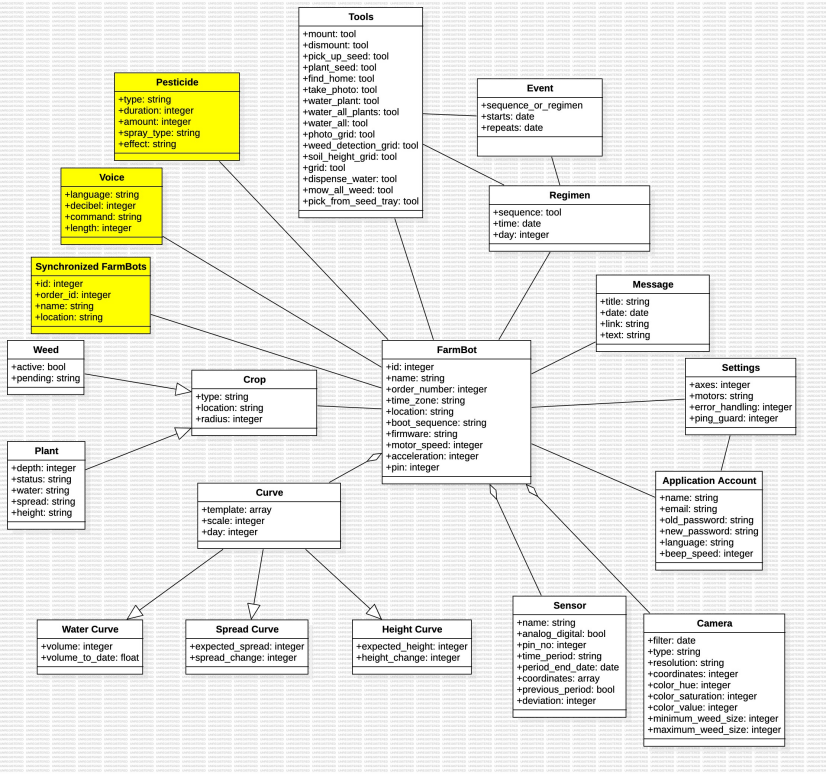

Figure 18: DataBase Class Diagram For Improvements

5.3.3 Operations on Data

Operation	Description
GetPesticideType	Create: Read: String Update: Delete:
ChangeDuration	Create: Read: Update: Integer Delete:
ChangeAmount	Create: Read: Update: Integer Delete:
GetSprayType	Create: Read: Information Update: Delete:
GetEffect	Create: Read: String Update: Delete:
SetVoiceLanguage	Create: Language Read: Update: Float Delete:
SetDecibelThreshHold	Create: Read: Update: Integer Delete:

Table 12: Internal Interface Operation Descriptions

Operation	Description
GetTranslatedCommand	Create: Read: String Update: Delete:
SetVoiceLength	Create: Read: Update: Integer Delete:
GetConnectedFarmBotsID	Create: Read: IntegerArray Update: Delete:
GetConnecetedFarmBotsLocation	Create: Read: City Update: Delete:

Table 13: Internal Interface Operation Descriptions

5.4 Deployment View

5.4.1 Stakeholders' uses of this view

There are 3 main stakeholders of the system: the customers, developers and educative partners. Developers use this view to understand the technologies and strategies used within the deployment process. Customers use this view to understand the capabilities of the system. Educational partners use this view to improve and customize the technologies and strategies used within the deployment process for their educational purposes.

5.4.2 Deployment Diagram

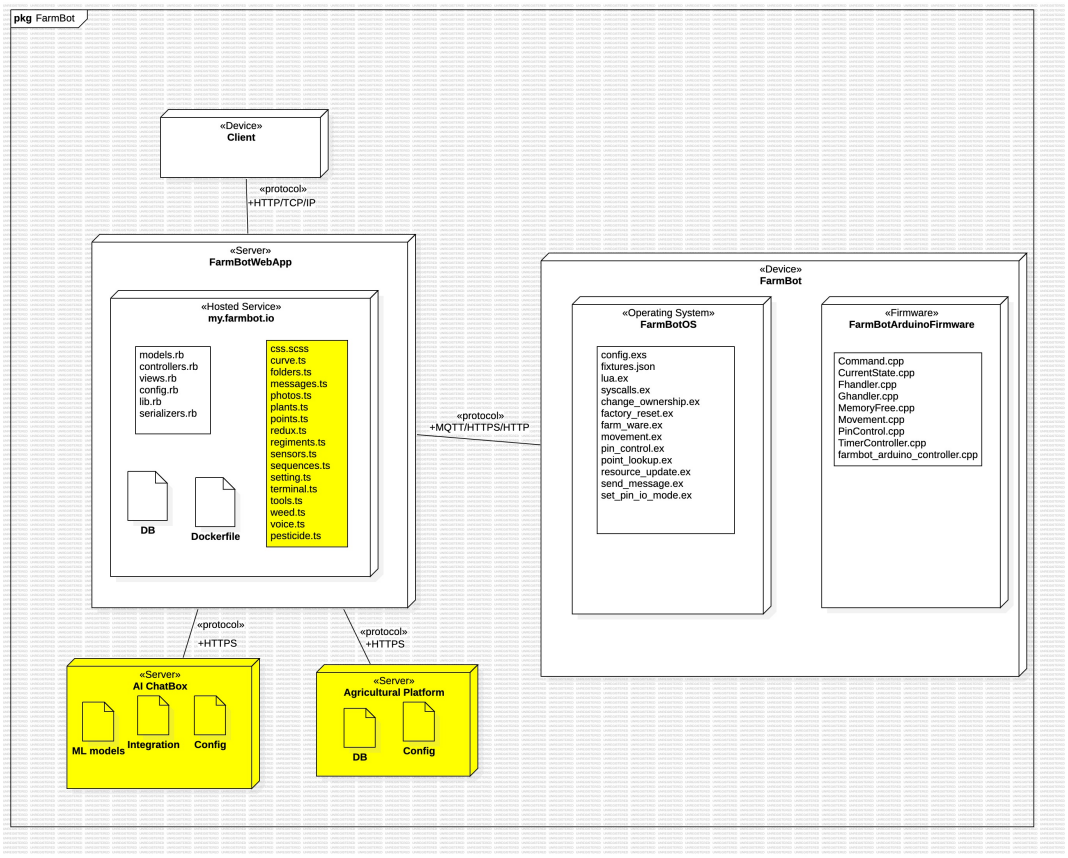

Figure 19: Deployment Diagram For Improvements

- Both AI server and Agricultural servers communicate with FarmBot web server by using HTTPS via port 80 similar to typical web application.
- voice.ts and pesticide.ts are the interfaces of voice command and pesticide usage that are newly added as an improvement.
- AI server contains suitable machine learning models for the appropriate use of ChatBox integration.

5.5 Design Rationale

5.5.1 Context View

In the context of design rationale, the context view of a project provides a high-level overview of the project's environment, stakeholders, goals, and constraints. It provides understanding for the broader context within which design decisions are made, and provides insights into the rationale behind those decisions. Integrating voice commands reduces the need for manual input, enhancing user experience and enabling hands-free operation, thus improving efficiency and safety.

5.5.2 Functional View

In the context of design rationale, the functional view of a project focuses on describing the functional requirements, capabilities, and interactions of the system being designed. It provides a detailed understanding of how the system will behave and what functionality it will offer to users. It also touches on scalability, modularization, requirements validation and verification. Voice command integration expands the accessibility of the FarmBot system, allowing users with limited mobility or those working in noisy environments to interact with the system effectively, thereby increasing and usability. concerns of the system.

5.5.3 Information View

In the context of design rationale, the information view of a project focuses on describing the information architecture, data models, and information flow within the system being designed. It provides view into how data is organized, stored, processed, and communicated within the system. . It also touches on consistency, interoperability, integration, security, and privacy concerns.Voice commands provide real-time interaction feedback, facilitating seamless communication between the user and the FarmBot system. This enhances user understanding and control over the system's actions, leading to more informed decision-making of the system.

5.5.4 Deployment View

In the context of design rationale, the deployment view of a project focuses on describing how the system is deployed, distributed, and managed within its operational environment. It provides view into the infrastructure, platforms, configurations, and deployment strategies used to make the system available to users. It also touches on scalability, performance, availability, fault tolerance, communication between components, and resource management factors of the system. Implementing voice command integration utilizes existing technology and APIs, enabling rapid deployment across various platforms and devices, ensuring scalability and interoperability with minimal development overhead.