Automatic Plant Shop

Kritapas Chutiwanichyakul 62011145
Thawat Jarupenpat 62011278
Tinchupeam Weerayutvilai 62011281

Bachelor of Engineering in Software Engineering
Department of Computer Engineering
School of Engineering
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2022

Thesis – Academic Year 2022

Bachelor of Engineering in Software Engineering

Department of Computer Engineering, School of Engineering

King Mongkut's Institute of Technology Ladkrabang

Title: Automatic Plant Shop

Authors

1. Kritapas Chutiwaniychyakul Student ID: 62011145
2. Thawat Jarupenpat Student ID: 62011278
3. Tinchupeam Weerayutvilai Student ID: 62011281

Approved for submission

.....

(Dr. Pipat Sookvatana)
Advisor

Date ..27.../..05.../..2023..

Acknowledgement

We would like to express our deepest gratitude to Asst.Prof.Dr. Pipat Sookvatana who has provided us with his insight, guidance and meaningful advice in this project. His dedication and commitment to our team is tremendous which allows us to grow into better individuals characters and software engineers.

We would like to express our deepest appreciation to Asst.Prof.Dr. Kasin Vinchienchom for giving us insight and wisdom on this project. Without his teachings, it would be very difficult and challenging to complete the project.

We also would like to express to Dr. Nattapong Jungterapanich for giving us insight and useful recommendations on this project.

Abstract

The automatic plant shop thesis report takes an action into the urgent issue of global warming, aiming to address people's concerns by promoting plant cultivation and raising awareness about the current situation. In order to achieve our objective, we aim to tackle the lack of plant markets in Thailand, with the majority located in the capital city, Bangkok, making them inaccessible to a large portion of the population.

To overcome this challenge, we have developed a mobile plant shop machine system, which can be conveniently relocated to any area across the country. This innovation enables customers interested in purchasing plants to access the shop more easily than ever before. By implementing this solution, we hope to encourage widespread plant cultivation and contribute to spreading awareness about the critical environmental issues we face today.

The project consists of two essential components: the software and hardware aspects. The hardware side of the project is controlled by Raspberry Pi, a reliable and multipurpose hardware controller. Raspberry Pi serves as the central control unit, responsible for managing and executing various operations within the automatic plant shop.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The motivation behind the automatic plant shop project bases from the pressing global issue of climate change and the need to raise awareness about it. With increasing concerns over global warming and its negative effects on the environment, it has become crucial to encourage sustainable practices and empower individuals to make a positive impact.

Figure 1 - Flood Effect

Figure 2 - Cracked Land

One effective way to combat climate change is through the cultivation and maintenance of plants. Plants play a vital role in carbon sequestration, oxygen production, and overall ecosystem balance.

Figure 3 - Plant the Tree

However, the limited availability of plant markets in Thailand, particularly in regions outside the capital city, presents a significant challenge for individuals who wish to purchase plants and contribute to environmental conservation efforts.

Figure 4 - Plant Market

Based on `Kapook.com` (https://home.kapook.com/view53551.html), popular plant market in Thailand including,

- Chatuchak Park (สวนจตุจักร)
- 11th Infantry Regiment, King's Guard (กรมทหารราบที่ 11 รักษาพระองค์)
- Thewet Plant Market (ตลาดต้นไม้เทเวศร์)
- Thonburi Market - Sanam Luang 2 (ตลาดนัดธนบุรี - สนามหลวง 2)
- Plant Market Bangyai-Bangbuathong (ตลาดต้นไม้ บางใหญ่-บางบัวทอง)
- Taling Chan Floating Market (ตลาดน้ำตลิ่งชัน)
- Kanchanaphisek Ring Plant Market (ตลาดต้นไม้วงแหวนกาญจนาภิเษก)
- Market of Ornamental Flowers, Thai market (ตลาดไม้ดอกไม้ประดับและกิ่งพันธุ์ไม้ตลาดไท)
- Thansiri Tree Market - Khlong Hok (ตลาดต้นไม้ธัญศิริ - คลองหก)
- Ornamental Flower Village - Klong 15 (หมู่บ้านไม้ดอกไม้ประดับ - คลอง 15)
- Plant market along the Ekamai-Ramintra Expressway (ตลาดต้นไม้เลียบทางด่วนเอกมัย-รามอินทรา)
- Lertnimit Plant Market - Romklao (ตลาดต้นไม้เลิศนิมิตร - ร่มเกล้า)
- Chao Chom Plant Market (ตลาดต้นไม้เจ้าจอม)
- Tree market, Soi Wat Phra Ngoen (ตลาดต้นไม้ ซ.วัดพระเงิน)

All of them are located in Bangkok and nearby locations. This is the indicator that tells us most of them are inaccessible to a large portion of the population in Thailand.

To address this challenge, the automatic plant shop project aims to connect the gap between plant enthusiasts and accessible plant markets. This not only enables individuals to beautify their surroundings but also supports a sense of responsibility towards the environment and encourages sustainable practices on a larger scale.

By designing a mobile plant shop machine system that can be conveniently located anywhere in the country, the project intends to provide people with easier access to a wide variety of plants.

Figure 5 - Flower Vending Machine

The figure above shows one of the examples of idealistic vending machines that can be used for selling plants and flowers.

1.2 Objectives

- Develop a user-friendly and intuitive frontend interface to enhance the customer's experience.
- Create an efficient backend system to handle server-side operations and data management.
- Implement user authentication for staff users to ensure the ease of data management.
- Enable seamless communication between the software and hardware components using the MQTT protocol which is the modern standard for efficient data exchange of the IoT industry.
- Implement a comprehensive search functionality to allow customers to easily find and explore various plant options based on their preferences.
- Enable online payment integration for a smooth and convenient transaction process.
- Develop a comprehensive functionality that enables staff to efficiently care for each plant based on its specific needs and characteristics.
- Develop a comprehensive functionality that enables staff to efficiently manage the plant's data including adding and deleting its details.
- Implement features that allow staff members to easily track and monitor the state of each plant throughout the process of caring for and selling them.
- Ensure the scalability and extensibility of the software system to accommodate future enhancements and features.
- Implement a clear and informative display of plant specifications to ensure that customers have a thorough understanding of the plant they are considering purchasing, including care requirements and instructions.
- Ensure the data is collected for future purposes.

1.3 Scope of Work

- Implement a hardware control system using Raspberry Pi to manage and control various operations within the plant shop machine.
- Utilize the MQTT protocol for seamless and efficient communication between the server and the Raspberry Pi hardware controller.
- Develop an online payment workflow to enable convenient transactions.
- Design the system that allows staff members to efficiently care for each plant based on its specific needs and characteristics.

1.4 Thesis Structure

This thesis consists of seven chapters which are arranged as follows:

- Chapter 1 - Introduction
- Chapter 2 - Project Plan
- Chapter 3 - Background Knowledge
- Chapter 4 - Requirements Analysis / System Architecture / Design
- Chapter 5 - Software Development
- Chapter 6 - Result
- Chapter 7 - Conclusion and Future Work

1.5 Team Responsibilities

- Kritapas Chutiwanichyakul : Back-end, Database Design, REST API, Testing
- Thawat Jarupenpat : Front-end, UI Design
- Tinchupeam Weerayutvilai : Hardware Model / IoT Programming (Pi 4, MQTT)

Chapter 2

Related Works

This chapter primarily presents the idea of the other existing software related to the project, which is Green Up Vending Machine. Its related strength and weakness plays an important role in the designing process of this thesis project report.

2.1 Green Up (Vending Machine)

The Green Up Vending Machine is the first plant vending machine in Thailand. It offers a wide variety of plants for sale, with each plant displayed in a specific slot. Each slot is labeled with a unique 'slot number' and has a payment QR code. The machine features an LCD screen that shows the name of the plant associated with each slot.

Figure 6 - Green Up Vending Machine

To purchase a plant, customers simply scan the payment QR code located on the front of the slot door. The machine is designed to be simple and fast, with no touchscreen or complex user interface. This simplicity is one of its main selling points, ensuring a quick and efficient purchasing process for any plant.

For customers seeking additional information on plant care or wanting to ensure they choose the right plant, the machine provides a chatbot assistant on its Facebook page. This assistant guides customers through the steps and allows them to make their purchases using the QR code displayed on the LCD screen, rather than on the front of the slot door. This helps prevent misunderstandings and ensures a smooth buying experience.

Figure 7 - Green Up Facebook Chat

Figure 8 - Green Up LCD Screen

We have analyzed some of the strengths and weaknesses of Green Up Vending Machine below here.

Strengths	*Weaknesses*
Convenience : The vending machine provides a convenient and accessible way for customers to purchase plants anytime, without the need for human interaction or specific store hours.	**Limited Interaction** : The absence of a touchscreen or user interface on the machine may limit the ability to provide detailed information or customization options during the purchasing process.
Efficiency : The straightforward process of scanning the payment QR code makes the purchasing experience quick and hassle-free.	**Potential Misunderstanding** : The separation of the payment QR code from the slot door may lead to confusion or misunderstandings for customers unfamiliar with the system, requiring clear instructions and signage.
Customer Support : The chatbot assistant on the Facebook page offers additional guidance and support, helping customers make informed decisions and address any queries they may have.	**Language Barrier** : If the chatbot assistant is the primary source of additional information, language barriers or technological limitations may hinder effective communication or access to detailed plant care instructions, as it is currently only available in Thai.
-	**Lack of Physical Inspection** : Customers may have limited opportunities to physically inspect the plants before making a purchase, relying only on the displayed name and possibly the accompanying images.

2.2 Idea Conclusion

Overall, the Green Up Vending Machine system offers convenience and a variety of plants, but some limitations include limited interaction, potential language barriers, and the inability to physically inspect plants before purchase. These factors should be considered to ensure a positive customer experience and address potential challenges.

Chapter 3

Background Knowledge

3.1 Client Side

Client Side refers to the part of a computer application or system that runs on the user's device or client machine. In the context of web development, the client side typically refers to the code and processes that execute within a web browser or application on the user's device.

When a user interacts with a website or web application, the client side is responsible for handling and executing tasks directly on the user's device. This includes rendering the user interface, handling user input, and performing various operations and computations.

Key components of the client side include:

1. User Interface:
   The client side is responsible for rendering the visual elements and user interface components of a website or application. This involves HTML (Hypertext Markup Language) for structure, CSS (Cascading Style Sheets) for styling, and JavaScript for interactivity and dynamic behavior.

2. User Input Handling:
   The client side processes user actions such as button clicks, form submissions, and keyboard interactions, allowing users to interact with the application and trigger desired actions.

3. Data Manipulation and Validation:
   The client side often handles data manipulation, validation, and formatting before sending it to the server. This can include tasks such as form validation, data formatting, and basic data processing.

4. Client-Side Storage:
   The client side provides storage mechanisms, such as web storage (localStorage, sessionStorage) or cookies, to store and retrieve data on the user's device.

5. Client-Side APIs and Libraries:
   Various APIs and libraries are available on the client side, such as the Document Object Model (DOM) API for manipulating HTML elements,

AJAX (Asynchronous JavaScript and XML) for making asynchronous server requests, and front-end frameworks like React or Angular for building complex user interfaces.

3.2 Server Side

Server Side refers to the part of a computer application or system that runs on the server or backend infrastructure. In web development, the server side typically refers to the code and processes that execute on the server and handle the logic and data processing required to generate dynamic web content.

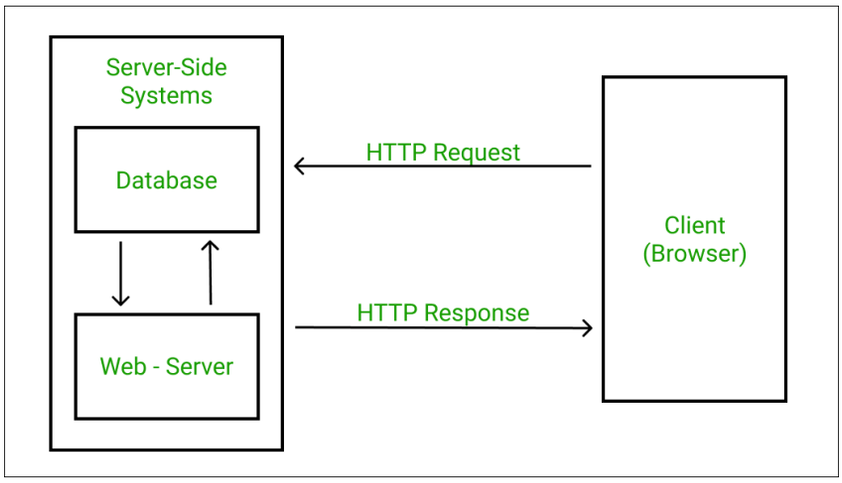

Figure 9 - Client Side and Server Side Communication

When a user interacts with a website or web application, the server side is responsible for receiving requests from the client side (user's device), processing those requests, and generating appropriate responses to be sent back to the client.

The server side refers to the part of an application or system that runs on the server, handling request processing, data storage, application logic, authentication, and providing APIs.

3.3 Communication

3.3.1 HTTP Request

An HTTP request is a message sent by a client (such as a web browser) to a server using the Hypertext Transfer Protocol (HTTP). It is the fundamental mechanism for communication between clients and servers on the World Wide Web. HTTP requests are used to initiate various actions or retrieve resources from a server.

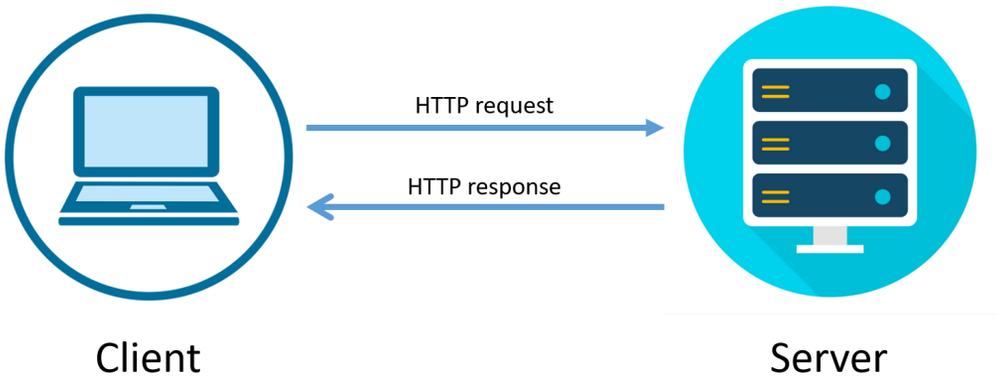

Figure 10 - HTTP Request and Response

An HTTP request consists of several components:

1. Request Line:
   It includes the HTTP method (e.g., GET, POST, PUT, DELETE) indicating the type of action to be performed, the URL or URI (Uniform Resource Identifier) specifying the resource location, and the HTTP version being used.

2. Request Headers:
   These are additional pieces of information sent along with the request, such as the client's user agent, accepted content types, cookies, and more.

3. Request Body (optional):
   Some types of requests, like POST or PUT, may include a request body that carries data to be sent to the server.

Figure 11 - HTTP Request Components

The most common HTTP methods used in requests are:

- GET: Retrieves a resource from the server.
- POST: Submits data to be processed by the server, often used for form submissions or creating new resources.
- PUT: Updates an existing resource on the server.
- DELETE: Removes a specified resource from the server.

When a client sends an HTTP request to a server, the server processes the request and sends back an HTTP response, which includes a status code indicating the success or failure of the request and the desired resource (if applicable).

3.3.2 HTTP Response

An HTTP response is a message sent by a server in response to an HTTP request made by a client (such as a web browser) using the Hypertext Transfer Protocol (HTTP). It contains the requested resource or information about the status of the request.

HTTP responses provide important information to clients, allowing them to interpret and process the server's reply. The status code helps the client understand the outcome of the request and take appropriate actions. For example, a 200 status code indicates a successful request, while a 404 status code indicates that the requested resource was not found.

3.3.3 MQTT

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for efficient communication between devices in constrained environments, such as low-bandwidth networks and devices with limited processing power. It follows a publish-subscribe pattern, where devices can publish messages to specific topics, and other devices can subscribe to those topics to receive the messages.

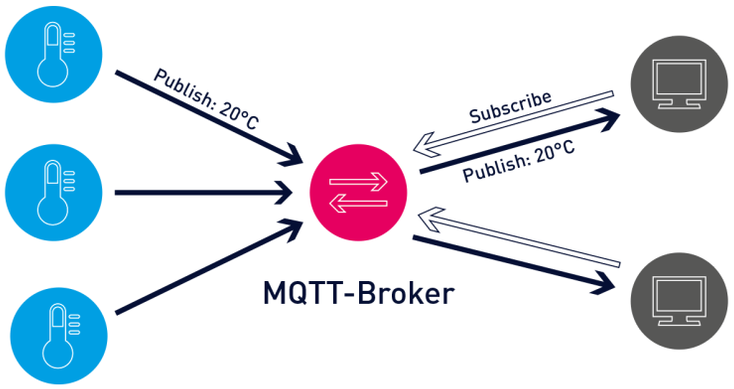

Figure 12 - MQTT

MQTT (Message Queuing Telemetry Transport) is used for efficient and reliable communication in various scenarios, especially in the field of IoT (Internet of Things) and machine-to-machine (M2M) communication.

3.3.4 Webhook

A webhook is a mechanism that allows applications or services to send real-time notifications or data to other applications or services through HTTP callbacks. It is a way for systems to communicate with each other by triggering an HTTP request to a predefined URL endpoint when a specific event or action occurs.

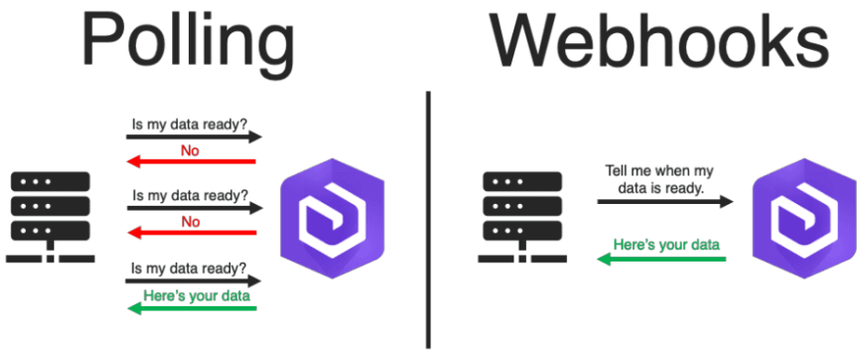

Figure 13 - Webhook

Here's how webhooks generally work:

1. Registration:
   The receiving application (webhook consumer) provides a URL endpoint and registers it with the sending application (webhook provider). The URL endpoint is where the webhook consumer wants to receive notifications or data.

2. Event Trigger:
   When a specific event or action occurs in the webhook provider application, such as a new order placed or a user registration, the webhook provider generates a payload of data related to that event.

3. HTTP POST Request:
   The webhook provider sends an HTTP POST request to the URL endpoint specified by the webhook consumer. The payload of data is included in the body of the request.

4. Webhook Consumer Processing:
   The webhook consumer receives the HTTP request at the registered URL endpoint. It can extract and process the data from the payload, perform additional actions, or update its own system based on the received information.

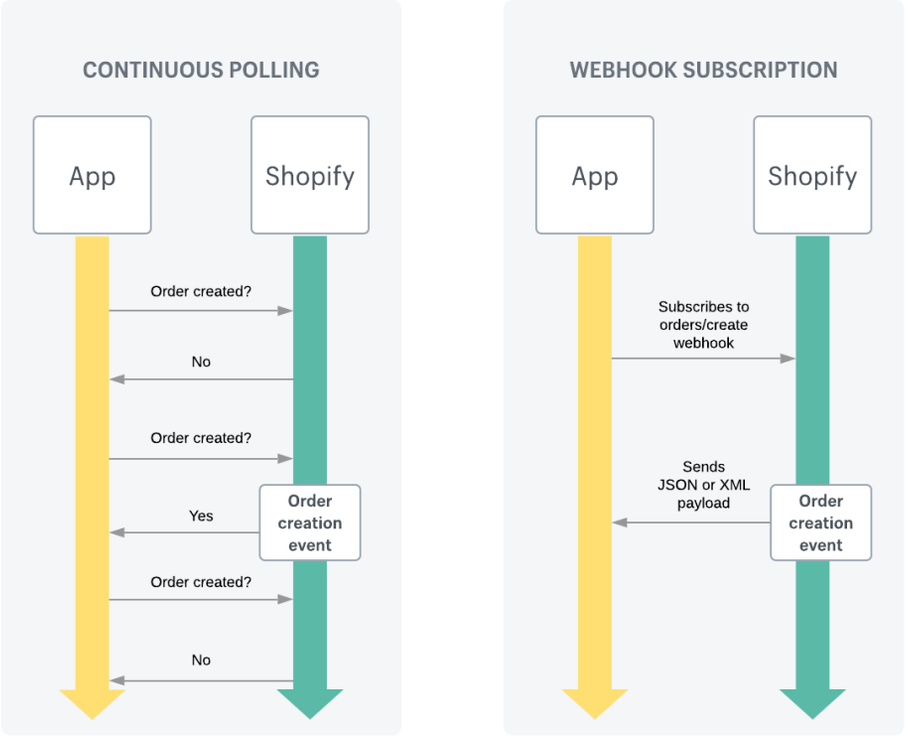

Figure 14 - Example of Webhook Workflow

3.4 Hardware Controller

Hardware Controller refers to a physical device or component that is responsible for managing and controlling various hardware operations and functionalities. It acts as an interface between the software and the underlying hardware components, facilitating communication and executing instructions to control hardware behavior.

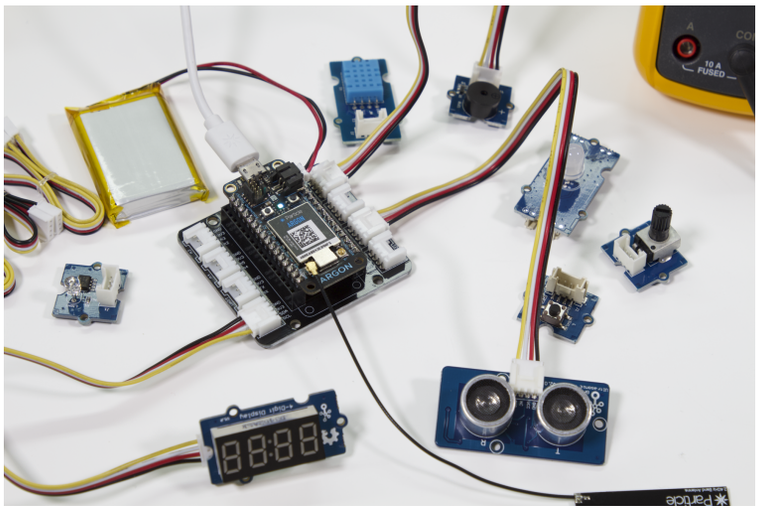

Figure 15 - Hardware Controller connected to other parts

3.5 Development Tools

3.5.1 React.js

React.js (also known as React) is an open-source JavaScript library widely used for building user interfaces (UIs) and front-end applications. It was developed by Facebook and is maintained by Facebook and a community of developers. React.js focuses on creating reusable UI components and provides a declarative syntax that simplifies the process of building interactive and dynamic user interfaces.

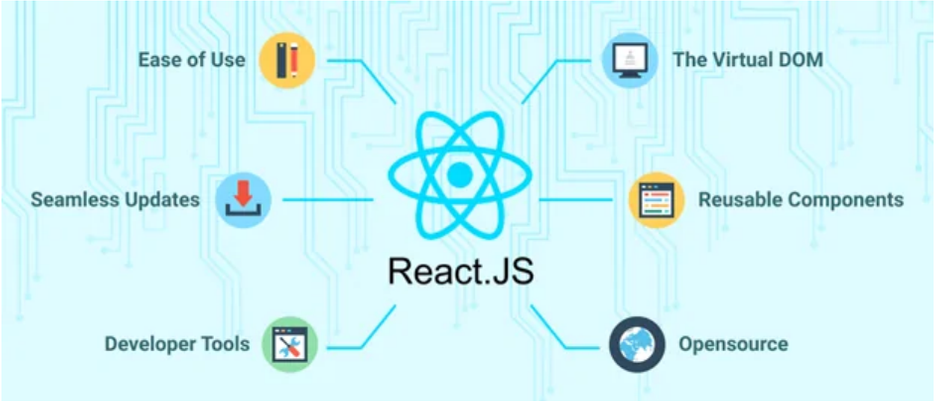

Figure 16 - React Introduction

3.5.2 Node.js

Node.js is an open-source, server-side JavaScript runtime environment built on Chrome's V8 JavaScript engine. It allows developers to execute JavaScript code outside of a web browser, enabling server-side scripting and the development of scalable and high-performance network applications. Node.js provides an event-driven, non-blocking I/O model, making it well-suited for handling concurrent connections and building efficient, real-time applications.

Figure 17 - JavaScript runtime environment

3.5.3 Express.js

Express.js is a fast and minimalist web application framework for Node.js. It provides a robust set of features and utilities for building web applications and APIs. Express.js simplifies the process of creating server-side applications in JavaScript by providing a lightweight and flexible framework.

3.5.4 Postgres

Postgres (or PostgreSQL), is an open-source relational database management system (RDBMS). It is known for its advanced features, scalability, and adherence to SQL standards. PostgreSQL offers a robust and reliable platform for storing, querying, and managing structured data.

Figure 18 - PostgreSQL Logo

3.6 PERN Stack

The PERN stack is a software development stack that combines several technologies to build web applications. PERN stands for PostgreSQL, Express.js, React.js, and Node.js. It is a full-stack JavaScript framework that allows developers to create modern, scalable, and efficient web applications.

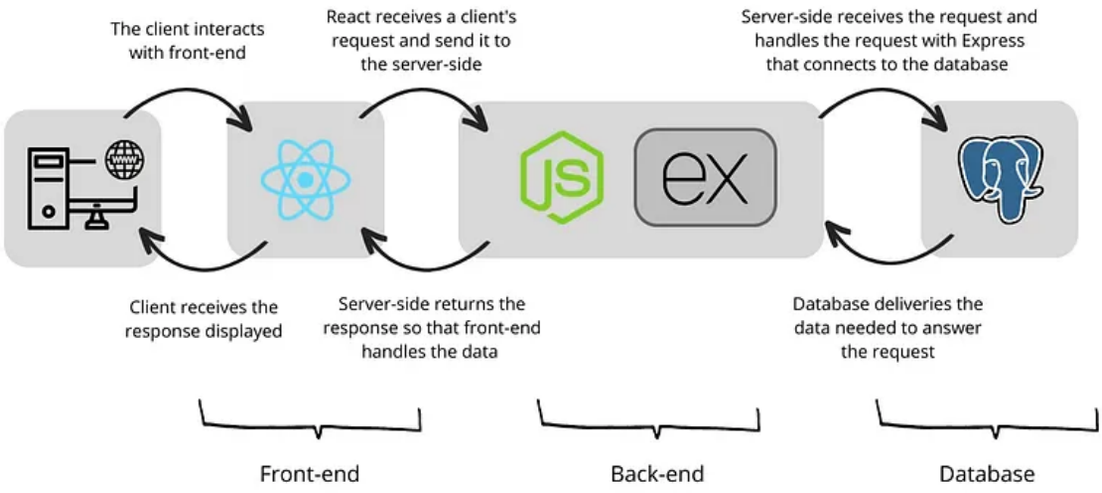

Figure 19 - PERN Stack

3.7 Raspberry Pi 4

The Raspberry Pi 4 is a small, single-board computer that serves as a versatile and affordable computing platform. It is part of the Raspberry Pi series, developed by the Raspberry Pi Foundation, and is the fourth generation of the Raspberry Pi line.

The Raspberry Pi 4 is widely used for a range of applications, including educational projects, DIY electronics, home automation, media centers, robotics, IoT devices, and prototyping. Its affordability, compact size, and broad community support make it accessible to beginners and professionals alike, supporting creativity and innovation in the world of computing.

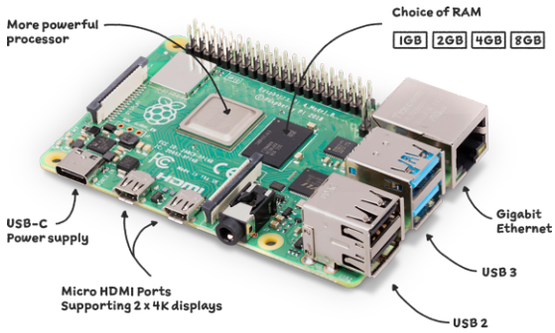

Figure 20 - Raspberry Pi 4

Chapter 4

Requirement Analysis and System Design

4.1 Requirements

4.1.1 Functional Requirements

1. The interface should allow users to browse available plants that are available in the specific machine.
2. The interface should provide a search bar to find their needed plants.
3. The interface should allow users to select a plant category they want.
4. The interface should provide details of the selected plant.
5. The interface should provide a suggestion of how to take care of each plant.
6. The user should be able to make a payment via QR following the suggestion steps on the screen.
7. The user should be able to pick up their plants after the payment process is done successfully.
8. The slot should be able to lock itself after the plant is picked up.
9. Staff should be able to add a new machine when its available location needs to be expanded.
10. Staff should be able to provide capacity for the specific machine via the staff interface.
11. Staff should be able to remove the specific machine that was made by mistake.
12. Staff should be able to add some new plants to the specific slot via the staff interface.
13. Staff should be able to provide important plant information to the system.
14. Staff should be able to edit and update plant information.
15. Staff should be able to remove plants that were made by mistake.
16. Staff should be able to open any slot door if there is a plant inside for the caring and watering purpose.
17. Staff should be able to mark if the plant has been already sold.
18. The whole system should be designed for ease of relocation purpose.

4.1.2 Non-Functional Requirements

1. The software should be developed as a web application.
2. The software will communicate to the hardware controller via the MQTT protocol.
3. The hardware controller should be able to open the door once the bank system confirms that the plant has been purchased.
4. The system should be able to analyze if there is a plant in a slot.
5. The Software Application should be developed based on Frontend and Backend aspects.
6. The hardware controller board will be mainly developed using Python.

4.2 Use-Case Diagram

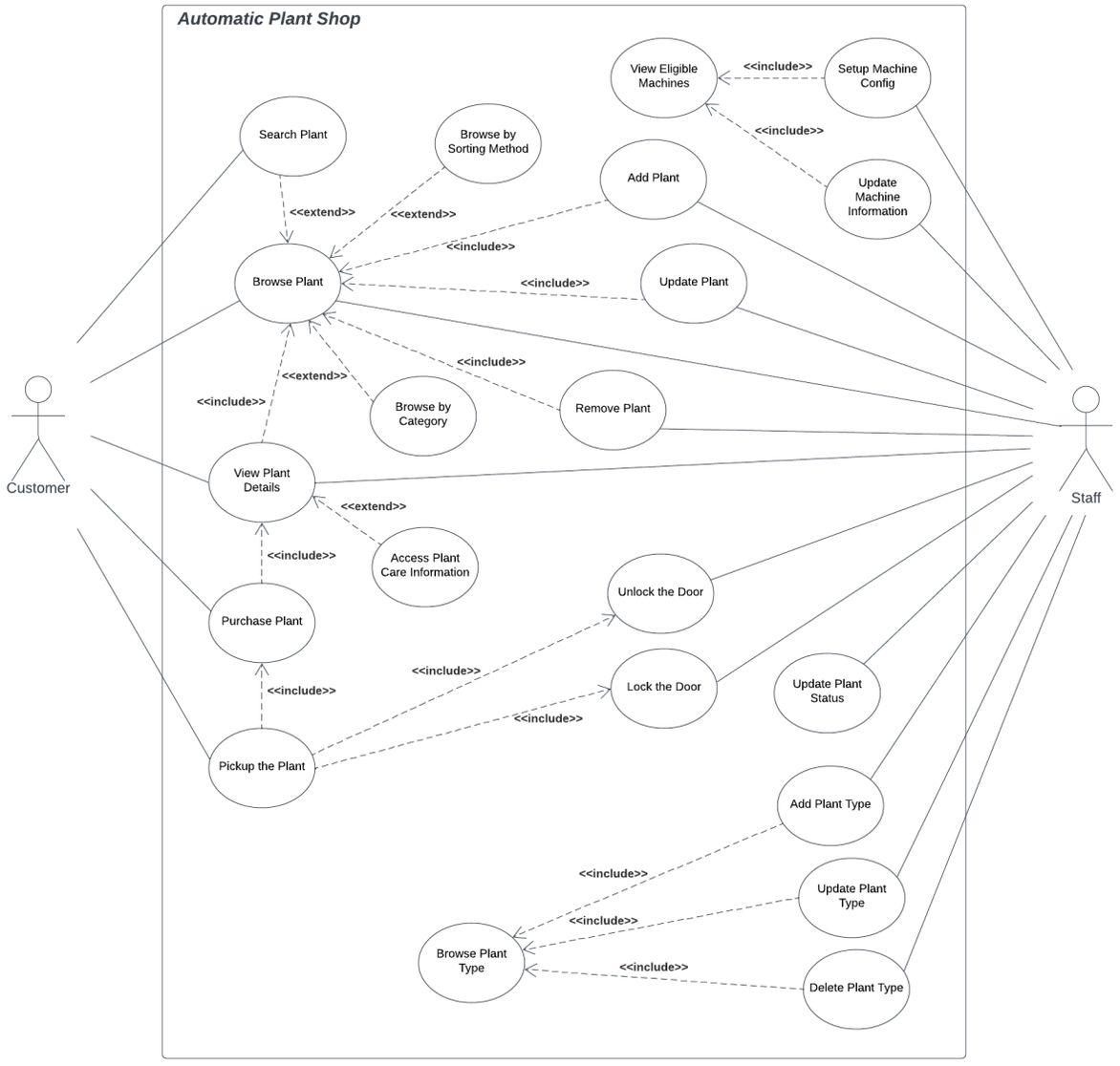

Figure 21 - Use-Case Diagram

Let's review all of the use-cases mentioned in the figure.

1. ***Customer***:

- ***Search Plants***: The customer can search for specific plants based on their preferences.
- ***Browse Plants***: The customer can browse the available plants in the machine.
- ***Browse Plant by Category***: The customer can browse plants by different categories.
- ***Browse Plant by Sorting Method***: The customer can browse plants according to specific criteria.

- ***View Plant Details***: The customer can view detailed information about a selected plant.
- ***Access Plant Care Information***: The customer can access information on how to care for a specific plant.
- ***Purchase Plants***: The customer can complete the purchase transaction for the selected plants.
- ***Open the Door***: The customer can open the door of the automatic plant shop after purchasing it.
- ***Close the Door***: The customer can close the door of the automatic plant shop.
- ***Pick up the Plant***: The customer can pick up a plant after the purchase is complete.

2. ***Staff***:

- ***Setup Machine Config***: Staff can configure the machine settings and parameters.
- ***View Eligible Machines***: Staff can view the eligible machines available for use.
- ***Update Machine Information***: Staff can update the information related to the machine.

- ***Browse Plant Types***: Staff can browse and view the available plant types in the system.
- ***Add Plant Types***: Staff can add new plant types to the system.
- ***Update Plant Types***: Staff can update the information and details of existing plant types.
- ***Delete Plant Types***: Staff can remove plant types from the system

- ***Search Plants***: Staff can search for specific plants based on various criteria.
- ***Browse Plants***: Staff can browse and view the available plants in the machine.
- ***Browse Plant by Category***: Staff can browse plants categorized by different categories.
- ***Browse Plant by Sorted Method***: Staff can browse plants sorted according to specific criteria.
- ***Add Plants***: Staff can add new plants to the machine.
- ***Update Plants***: Staff can update the information and details of existing plants.
- ***Remove Plants***: Staff can remove plants from the machine.

- ***Open the Door***: Staff can open the door of the automatic plant shop.
- ***Close the Door***: Staff can close the door of the automatic plant shop.
- ***Update Plant Status***: Staff can update the status or condition of plants in the inventory.

The use-case diagram showcases the various interactions and functionalities available to both customers and staff in the automatic plant shop system. Customers can perform actions related to browsing, searching, purchasing, accessing plant care information, and interacting with the physical door. Staff members have additional capabilities for managing machine configurations, plant types, inventory, and performing administrative tasks.

4.3 System Architecture

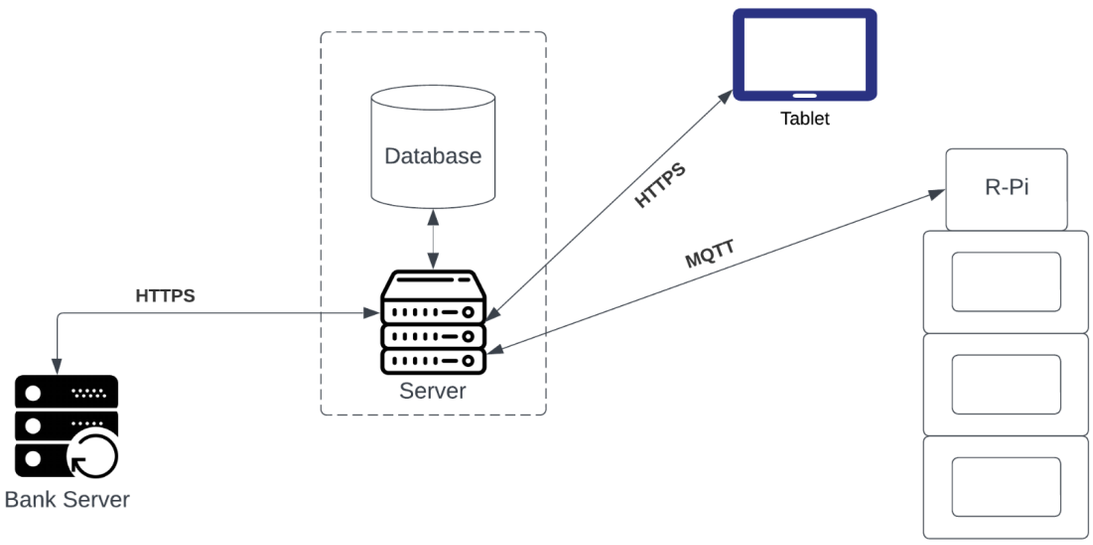

Figure 22 - System Architecture

Let's review all of the components mentioned in the figure.

1. ***Server***:

- The server acts as the central hub of the system, responsible for handling requests and managing the overall functionality.
- It hosts the backend application built using Node.js, which handles business logic, communication with the database, and integration with external services.
- The server receives requests from the user interface, processes them, and sends back the appropriate responses.
- It communicates with the Raspberry Pi for controlling the hardware aspects of the plant shop machine.

2. ***Database***:

- The database stores essential information about plants, machines, slots, customer details, and transactions.
- It provides persistent storage for the system, ensuring data integrity and availability.
- The server communicates with the database to retrieve and update relevant data when needed.

3. ***Screen***:

- The screen refers to the user interface component that displays the available plants, their details, and other relevant information.
- It allows customers to interact with the system, browse plants, search, view details, and access plant care information.
- The screen receives user input and sends requests to the server for processing and retrieving data.

4. ***Bank Server***:

- The bank server represents the external service that handles payment processing and authorization.
- When a customer proceeds with the purchase, the server communicates with the bank server to initiate the payment transaction securely.
- The bank server validates the payment details, deducts the appropriate amount, and provides confirmation or status updates to the server.

5. ***Raspberry Pi***:

- The Raspberry Pi serves as the hardware controller, responsible for controlling the physical aspects of the automatic plant shop machine.
- It communicates with the server, receiving instructions for opening or closing the door, updating the plant pickup status, or retrieving information.
- The Raspberry Pi interacts with the door mechanism to open or close the door when necessary.

These components work together to provide a seamless and integrated experience for customers while ensuring efficient management of data, transactions, and hardware control in the automatic plant shop system.

4.3.1 Server Design

In this section, we will introduce a separation of the server into two parts: the ***client server*** and the ***backend server***.

Additionally, there are two client servers: the **'customer server'** and the **'staff server'**. Let's further describe the design.

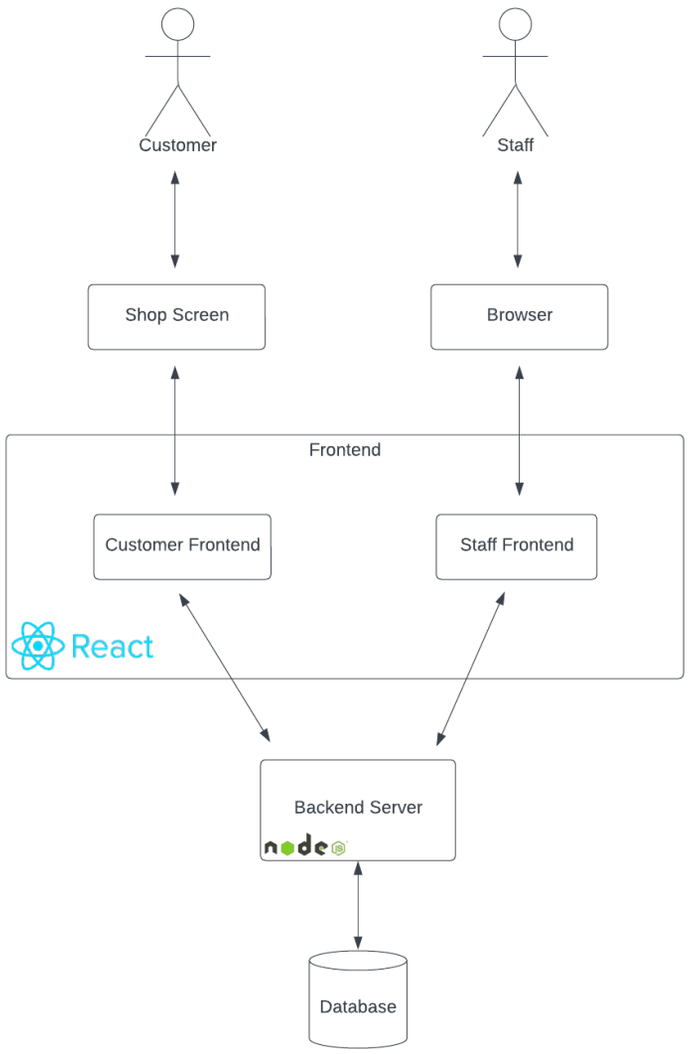

Figure 23 - Server Design

1. ***Client Server***:

- The client server component is responsible for serving the user interface (UI) to the clients (customers and staff members) who access the system.
- It handles the rendering of the UI and manages the interaction with the clients' web browsers.
- The client server consists of two separate instances: the 'customer server' and the 'staff server'.
- Each client server serves a specific UI tailored for either customers or staff members, providing a distinct user experience.

    1. ***Customer Server***:

- The customer server is responsible for serving the customer-facing UI, tailored specifically for browsing plants, making purchases, and accessing plant care information.
- It handles requests from the shop screen and communicates with the backend server to fetch data or perform transactions.
- The customer server ensures a smooth user experience for customers, providing a user-friendly interface and seamless interaction with the system.

    2. ***Staff Server***:

- The staff server serves the UI intended for staff members or shop owners, offering functionalities for managing data, plant types, and performing administrative tasks.
- It handles requests from the staff members' web browsers and interacts with the backend server to retrieve or update relevant information.
- The staff server provides a dedicated interface for staff members to efficiently manage the automatic plant shop system and perform their tasks.

2. ***Backend Server***:

- The backend server component handles the core business logic, data processing, and communication with the database.
- It serves as the central processing unit for the system, receiving requests from both the customer server and the staff server.
- The backend server processes the received requests, performs necessary operations (such as retrieving data from the database or updating records), and sends back the appropriate responses.
- It communicates with the client servers to exchange data and facilitate the required functionalities.

4.3.2 Payment Process Outline (Key Feature)

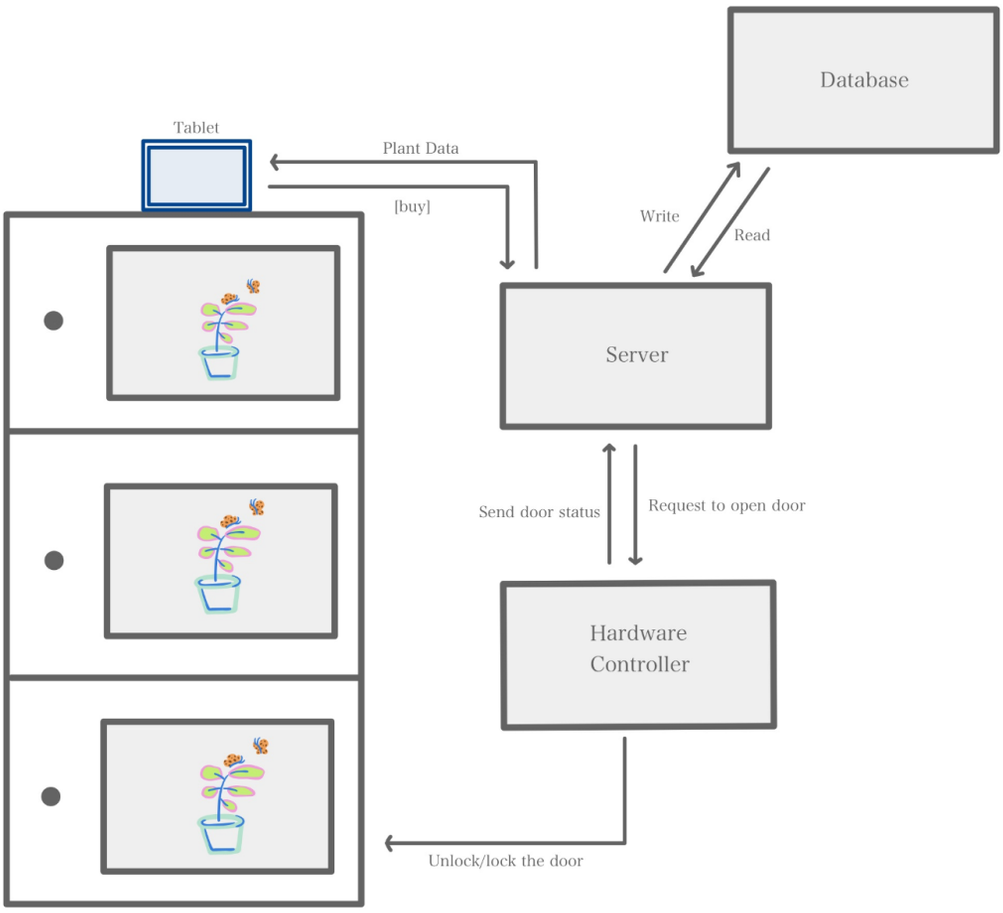

Figure 24 - Payment Process Outline

The payment process outline involves several components, including the screen, server, database, and hardware controller. Here's a description of the steps involved in the payment process once a customer selects a plant to be purchased:

1. ***Customer Interaction***:

- The customer interacts with the screen interface to select the plant they wish to purchase.
- The screen displays the available plants, their details, and the plant care instructions.
- The customer selects the desired plant from the screen interface.

2. ***Server Request***:

- The screen interface sends a request to the server, indicating the selected plant and initiating the payment process.

3. ***Server Processing***:

- The server receives the request and processes the payment transaction.
- It communicates with the database to validate the availability of the selected plant and retrieve relevant details.

4. ***Payment Authorization***:

- The server contacts the bank server to initiate the payment authorization process.

5. ***Transaction Update***:

- If the payment is authorized and successful, the server updates the transaction details in the database.
- The database records the purchase, including the plant details, customer information, payment status, and transaction timestamp.

6. ***Hardware Controller Instruction***:

- The server communicates with the hardware controller, typically a Raspberry Pi, to initiate the necessary actions.
- The hardware controller receives the instruction to open the door or release the selected plant for the customer.

7. ***Customer Notification***:

- The screen interface displays a notification to the customer, indicating the successful completion of the purchase.
- The customer is informed that they can proceed to pick up their plant from the designated area.

The above outline demonstrates the interaction between the components involved in the payment process. The customer interacts with the screen to select the plant, the server manages the payment processing and communication with the database and bank server, and the hardware controller carries out the physical actions, such as opening the door or releasing the plant.

Throughout the process, data is exchanged between the server, database, screen, and hardware controller to ensure accurate transaction records, payment authorization, and customer notifications. The coordination and integration of these components enable a smooth and secure payment process for the automatic plant shop system.

4.4 System Components Design

The automatic plant shop system comprises multiple interconnected components based on web application aspect, each serving a specific purpose and contributing to the overall functionality.

4.4.1 User Interface

The user interface, accessible via screens or web-based interfaces, allows customers and staff members to interact with the automatic plant shop system. The UI provides a visually appealing and intuitive platform for browsing available plants, making purchases, accessing plant care information, and managing administrative tasks. Separate UIs may be designed for customers and staff members to cater to their specific needs and roles.

1. ***Customer UI***

- ***Machine Setup Section***

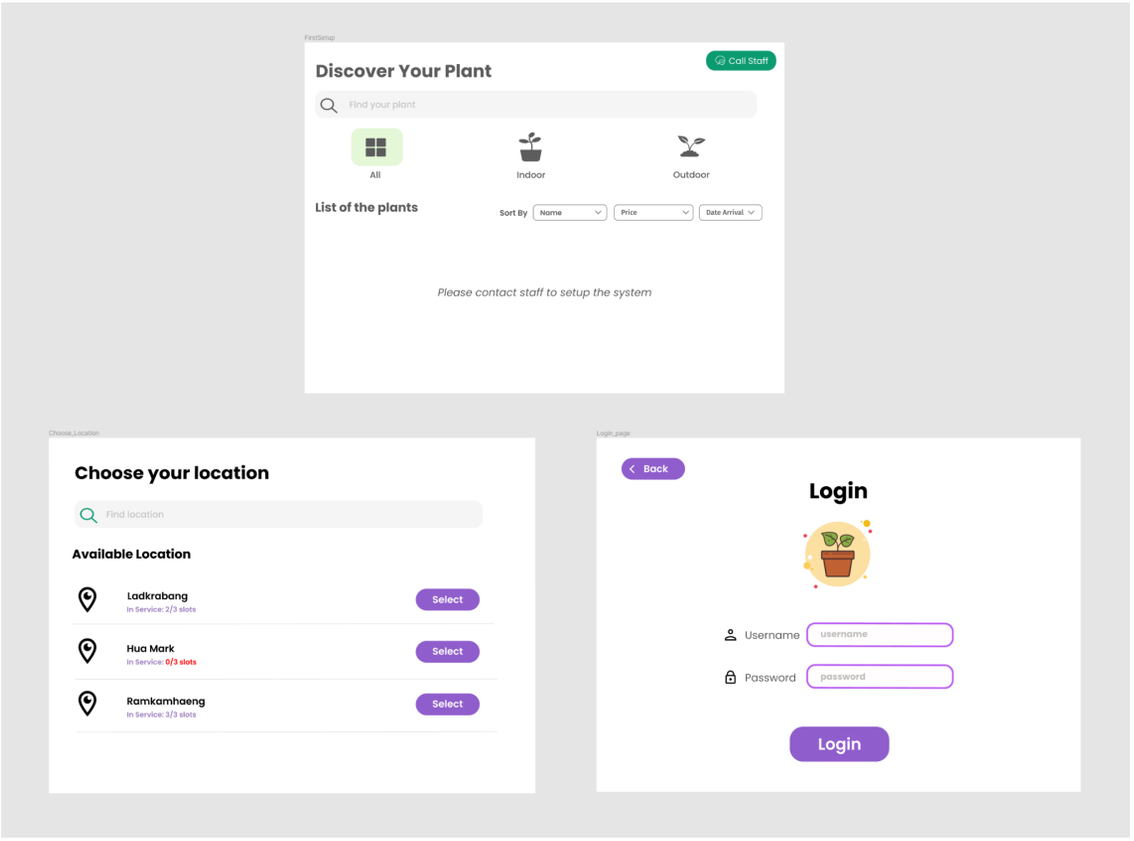

Figure 25 - Customer's Machine Setup UI

- The Machine Setup section is responsible for handling the initial setup process. It may include prompts and suggest customers to call staff of the automatic plant shop.

- This section ensures that customers have a smooth onboarding experience by providing clear instructions and necessary information to set up their interaction with the shop.

- Note that in the normal situation, customers do not need to go through this process. So, this page is designed for handling unusual situations or when system config needs to be set again.

- ***Browsing Section***

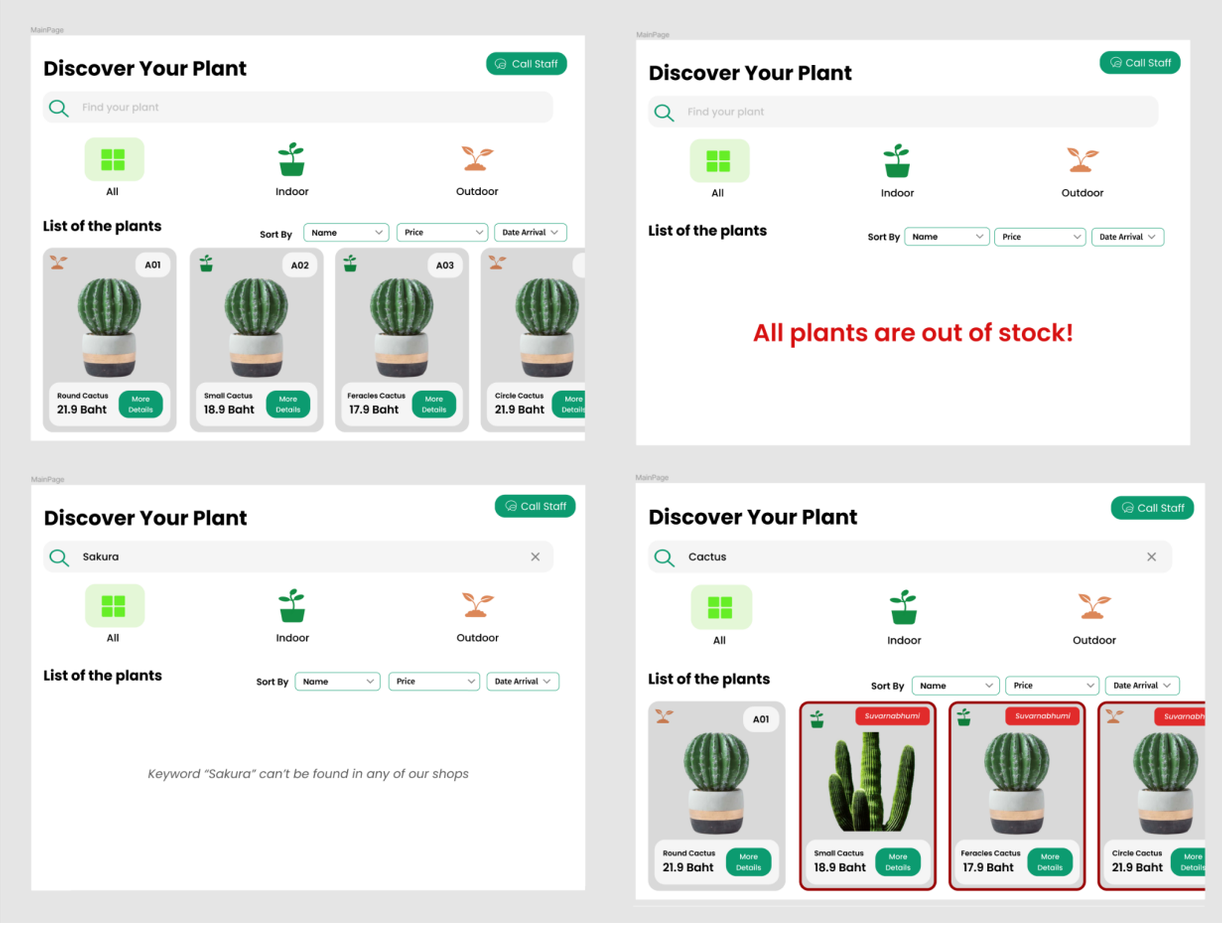

Figure 26 - Customer's Browsing UI

- The Browsing section of the customer UI presents an intuitive interface for customers to explore the available plants. It typically displays a list or grid view of plants, along with relevant details such as plant names, images, and prices.

- Customers can browse through the plants, filter them based on categories or sorting options, and view additional information about specific plants of interest.

- ***Information Providing Section***

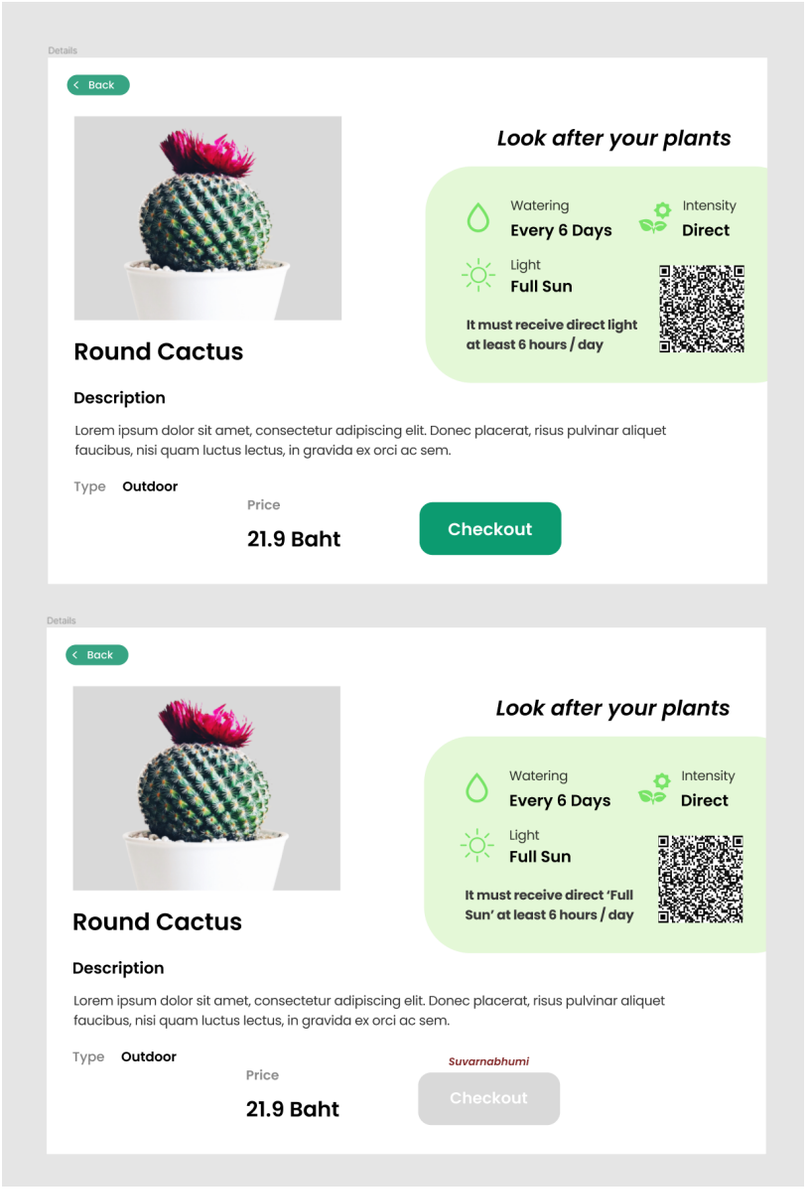

Figure 27 - Customer's Information Providing UI

- The Information section provides customers with detailed information about each plant. It may include sections such as plant care instructions, specifications, growth requirements, and any special considerations.

- This section helps customers make informed decisions by providing them with the necessary knowledge to choose plants that align with their preferences, living conditions, and gardening expertise.

Figure 28 - Example of Plant Care Instruction Presets

- ***Payment Section***

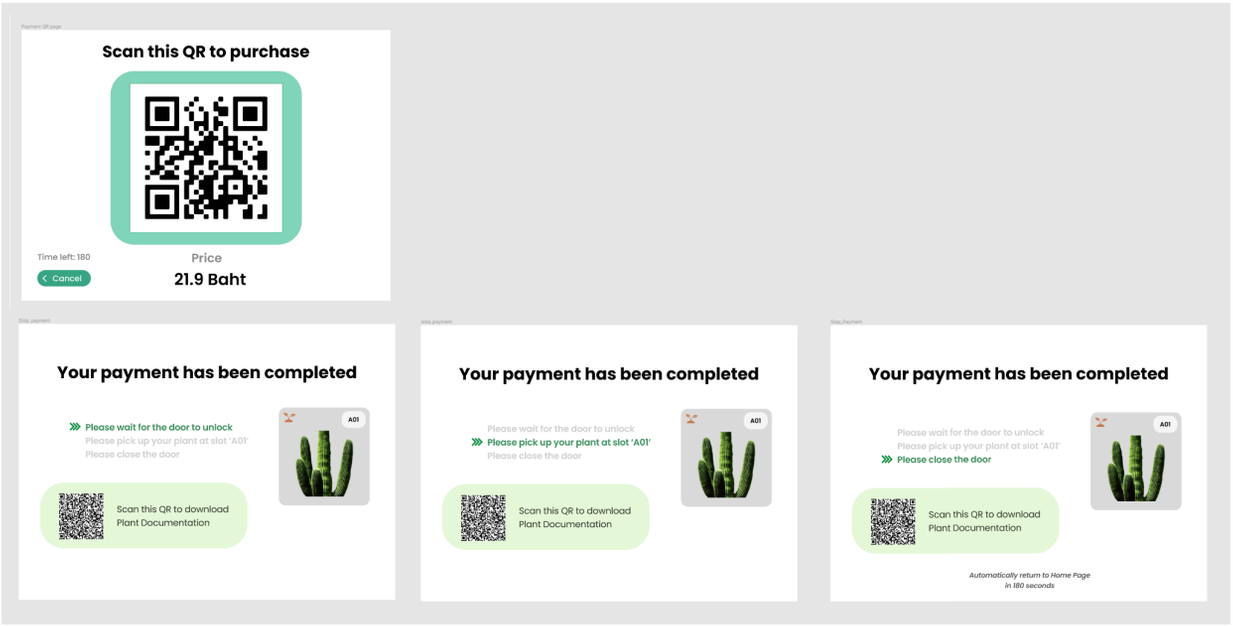

Figure 29 - Customer's Payment Section UI

- The Payment section facilitates the payment process for customers who wish to purchase selected plants. It offers a QR payment method to support the cashless payment in the present day.

- Once the payment is confirmed, the UI may provide a confirmation message, acknowledging the successful completion of the transaction. Then, the system will allow customers to pick up their purchased plant from its slot.

2. ***Staff UI***

- ***Machine Management Section***

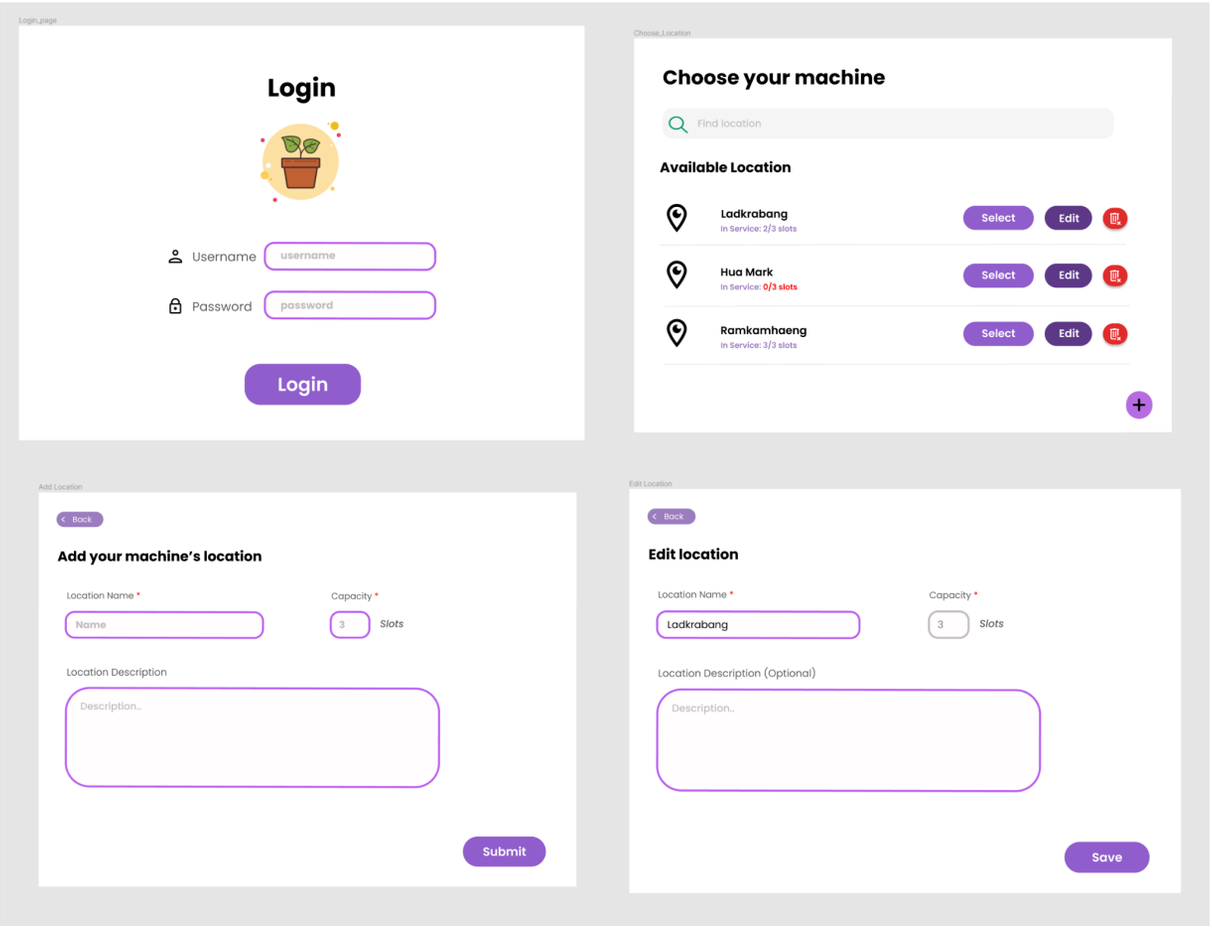

Figure 30 - Staff's Machine Management Section UI

- The Machine Management section provides staff members with tools to configure and manage the automatic plant shop machines. It may include functionalities like adding new machines, or performing maintenance tasks.

- Staff members can view and manage the machine data that is on their individual responsibility, ensuring the proper functioning and availability of the machines to customers.

- ***Plant Type Management Section***

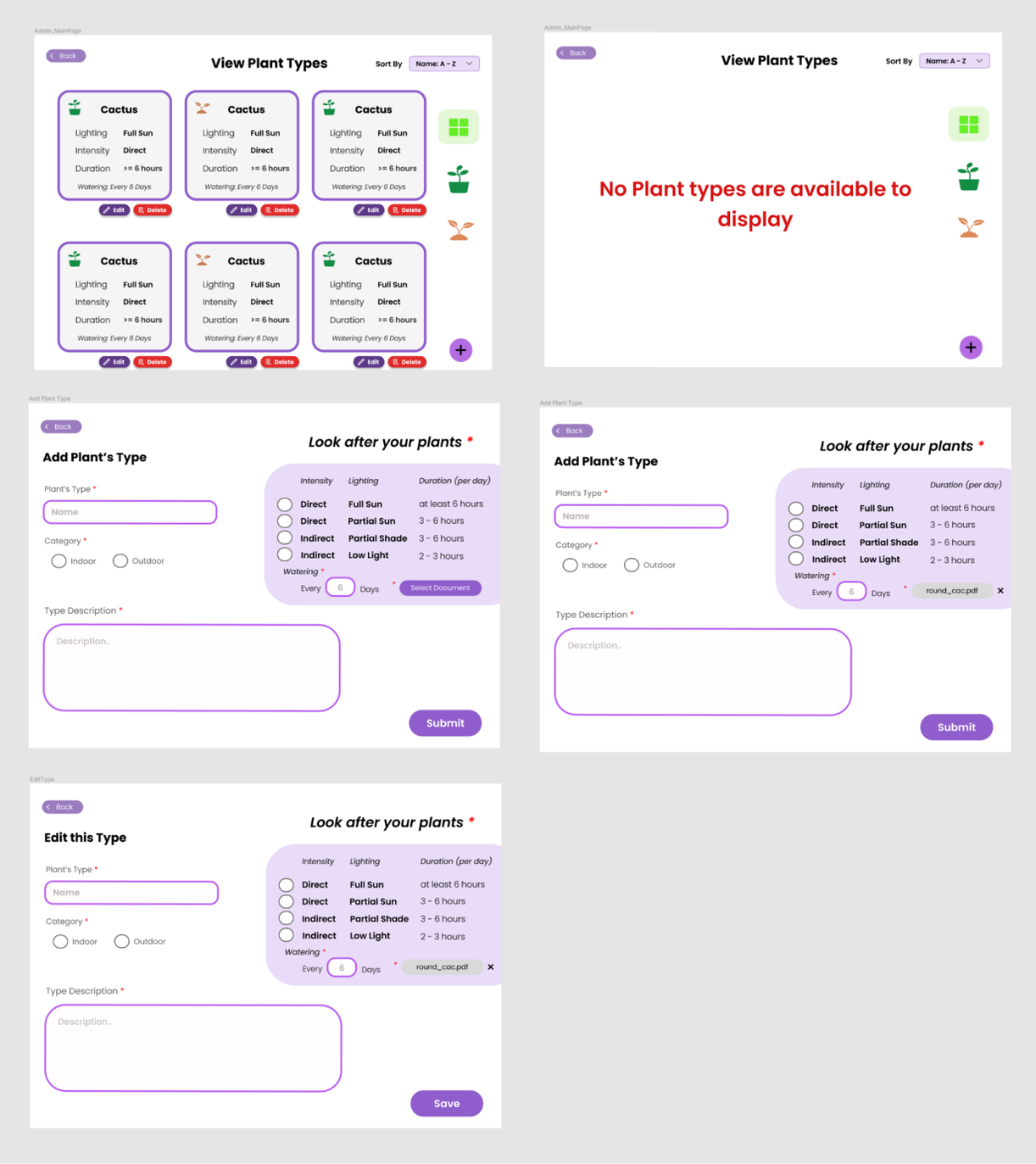

Figure 31 - Staff's Plant Type Management Section UI

- The Plant Type Management section allows staff members to manage the types of plants available in the automatic plant shop. Staff members can add new plant types, update existing ones, or remove plant types from the inventory.

- This section enables staff members to maintain up-to-date types, ensuring a wide variety of options for customers to choose from.

- ***Browsing Plant Section***

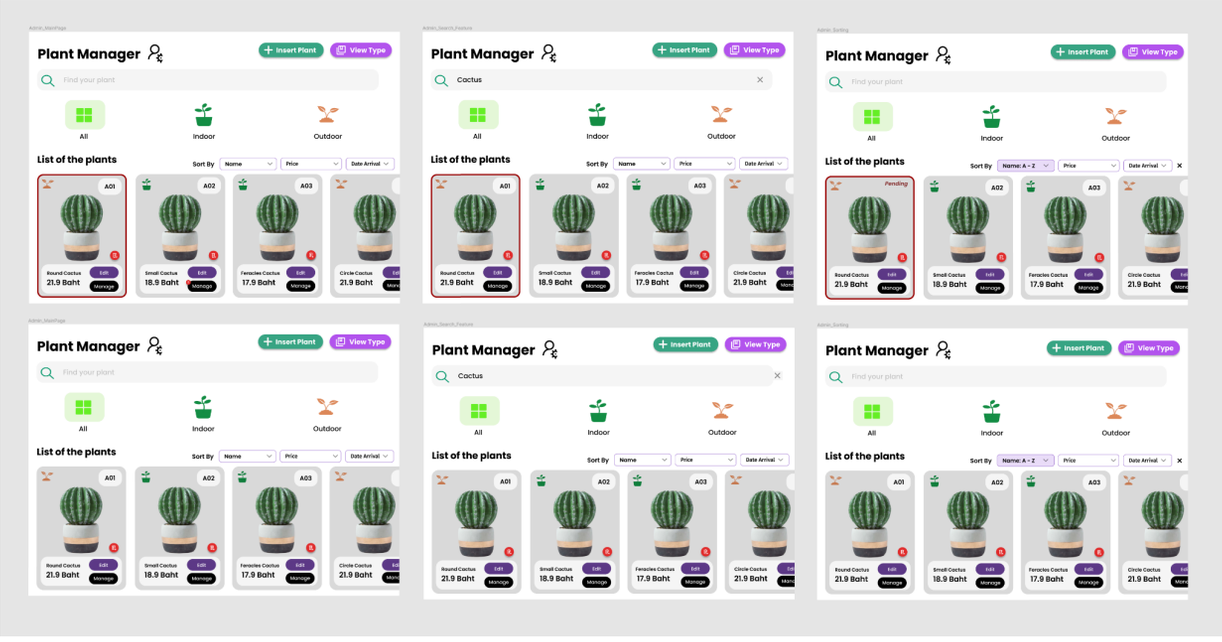

Figure 32 - Staff's Browsing Plant Section UI

- The Browsing Plant section allows staff members to browse through the available plants in the machine. It provides an interface similar to the customer browsing section, enabling staff members to view plant details, images, prices, and other relevant information.

- ***Plant Management Section***

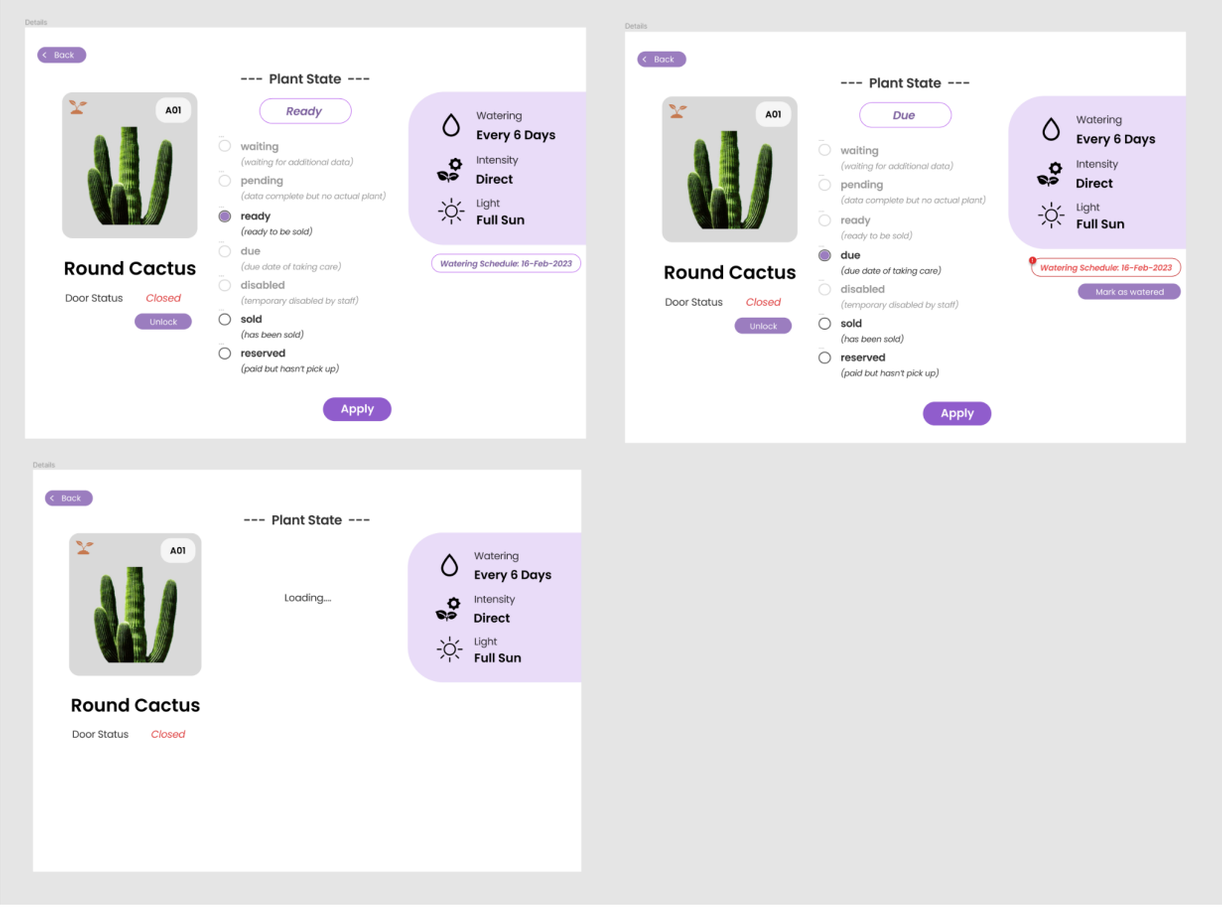

Figure 33 - Staff's Plant Management Section UI

- The Plant Management section provides staff members with functionalities to manage the existing plants in the automatic plant shop. This includes features such as updating plant status (e.g., sold, reserved).

- Staff members can use this section to keep track of plant data, or handle any plant care tasks by making a communication to the hardware controller.

- ***Plant Insertion Section***

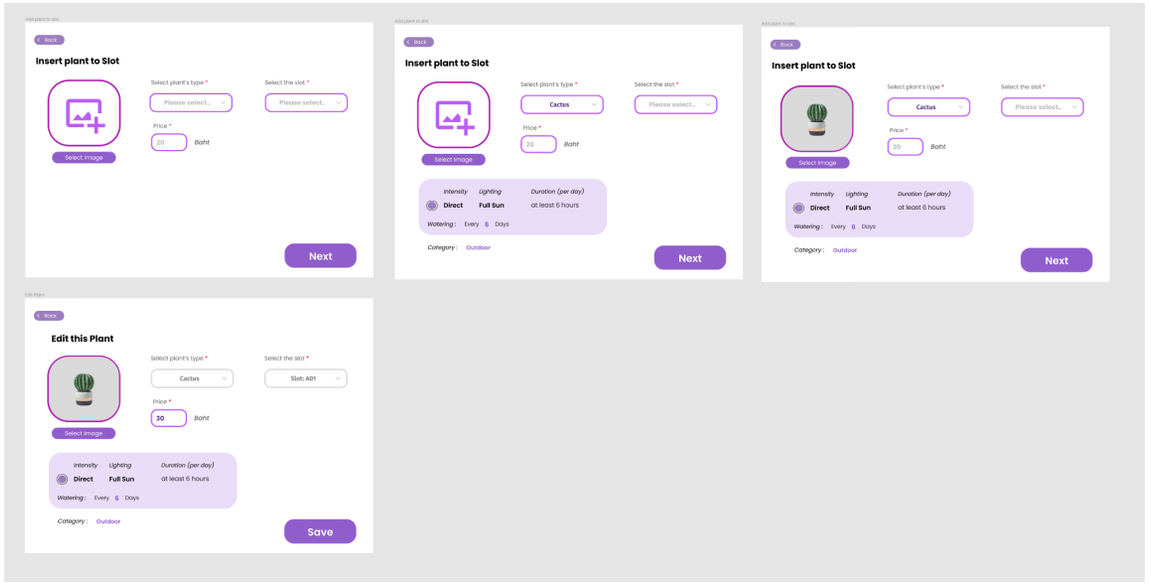

Figure 34 - Staff's Plant Insertion Section UI - Data Insertion

- The Plant Insertion section allows staff members to add new plants to the machine slot. It offers a form or interface where staff members can enter the required plant information, including plant type, slot to be inserted, images, price, and any other relevant attributes.

- Staff members can provide an accurate image of the new plants, ensuring not to make customers confused when browsing through the plant selection interface.

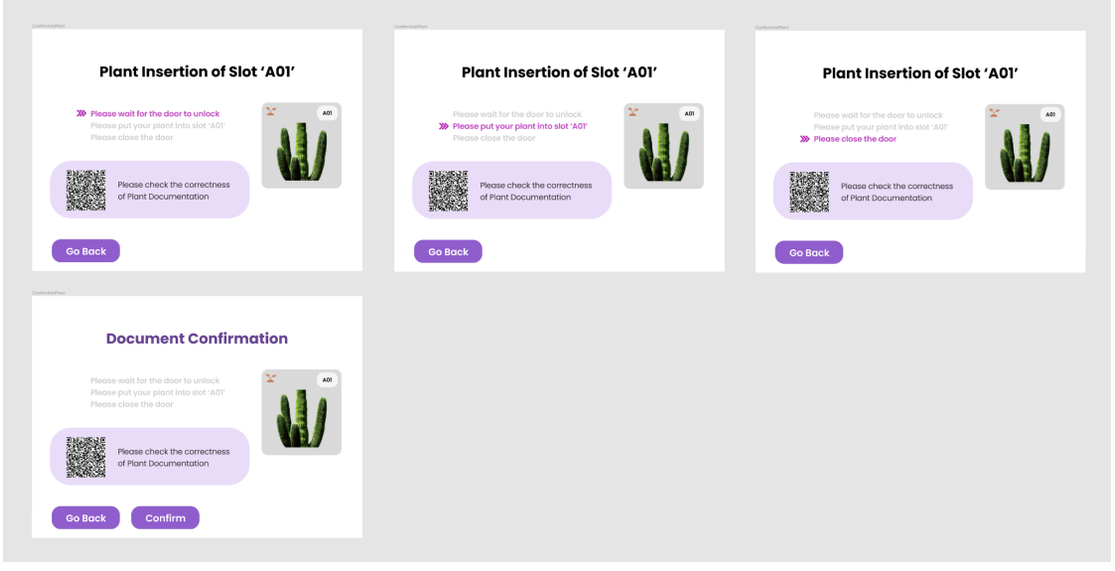

Figure 35 - Staff's Plant Insertion Section UI - Actual Plant Insertion

4.4.2 Backend Server

The server component serves as the backbone of the automatic plant shop system. It encompasses the backend application developed using technologies such as Node.js and Express.js. The server handles crucial functionalities like managing the database, processing customer requests, handling payment transactions, and coordinating communication between various components.

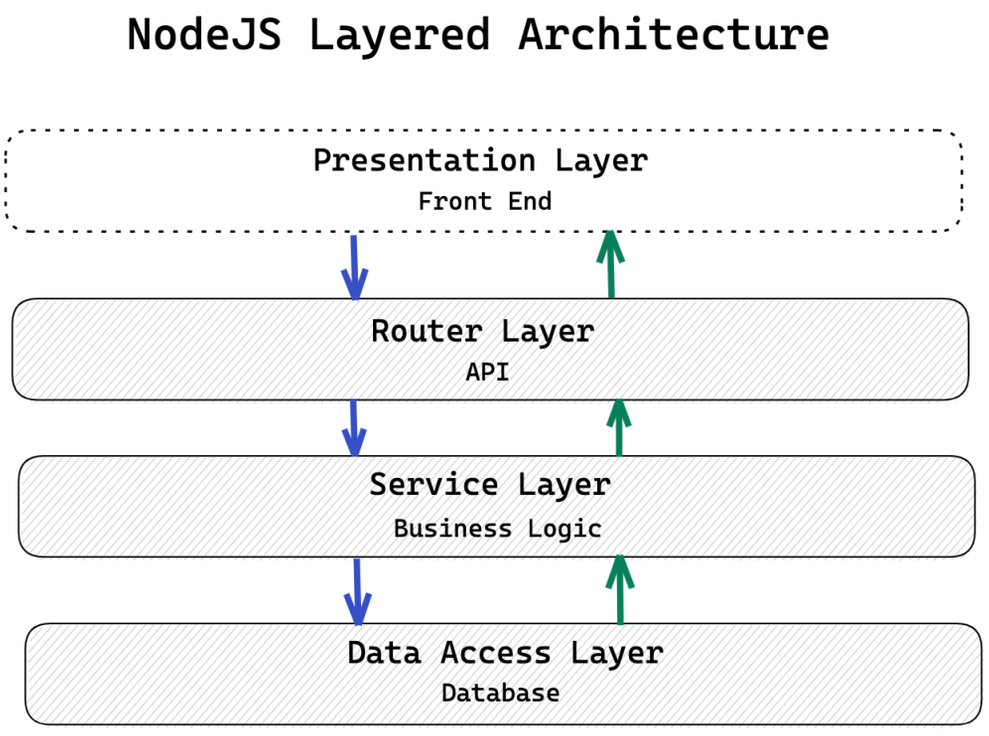

Figure 36 - NodeJS Layered Architecture

In the backend server of the automatic plant shop project, the development follows the NodeJS Layer Architecture, which includes the components: router, service, and DAO (Data Access Object). Additionally, the project incorporates the MVC (Model-View-Controller) pattern to handle a large number of requests. Let's explore each component and its role in the system:

1. ***Router***:

- The router component handles the incoming HTTP requests and directs them to the appropriate controllers based on the specified routes and endpoints.
- It defines the API routes, maps them to the corresponding controller functions, and facilitates the interaction between the client and the server.

2. ***Controller***:

- The controller is a crucial component in the MVC pattern. It receives requests from the router and acts as an intermediary between the router, service, and models.
- The controller handles request validation, calls the appropriate service functions to perform the required business logic, and prepares the response to be sent back to the client.
- It acts as a bridge, coordinating the flow of data and interactions between the various components involved in request processing.

3. ***Model***:

- The model layer defines the data schema, and validates incoming data.
- It plays a crucial role in enforcing data integrity, and performing data manipulations.

4. ***Service***:

- The service component encapsulates the business logic of the automatic plant shop system. It contains the core functionalities and implements the specific operations required for various actions and processes.
- The service layer handles the processing of requests received from the router and coordinates with other components, such as the DAO and models, to execute the necessary operations.

5. ***DAO (Data Access Object)***:

- The DAO component is responsible for interacting with the database or data storage layer. It abstracts the data access operations and provides an interface to perform CRUD (Create, Read, Update, Delete) operations on the database.
- The DAO layer communicates with the underlying database system, executes queries, retrieves data, and updates or persists information based on the service's instructions.

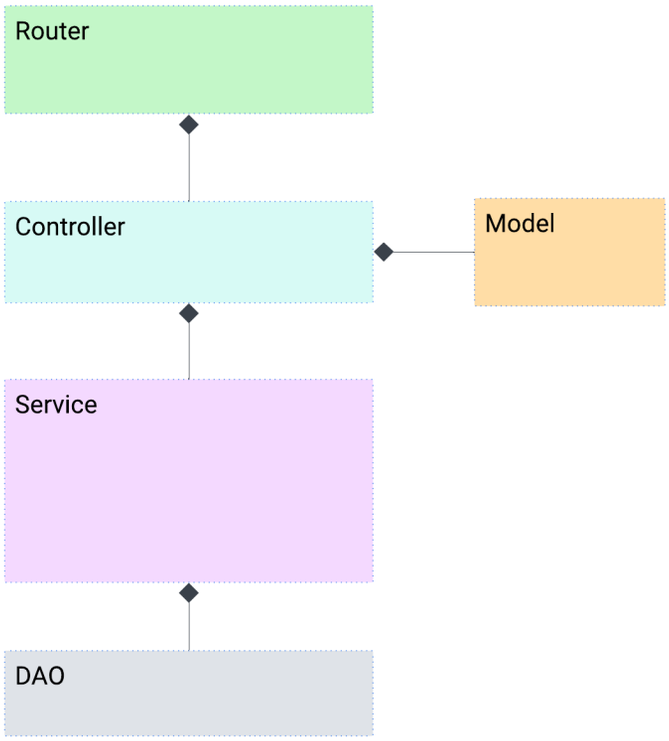

Figure 37 - Project Layered Architecture

4.4.3 Database

The database component stores essential data related to the automatic plant shop, including plant information, inventory details, customer records, and transactional data. It provides a secure and reliable means of storing and retrieving structured data. Technologies like PostgreSQL are commonly employed to ensure efficient data management and enable seamless integration with other system components.

The database design of the automatic plant shop project involves several entities and their relationships. Here's an overview of the database design:

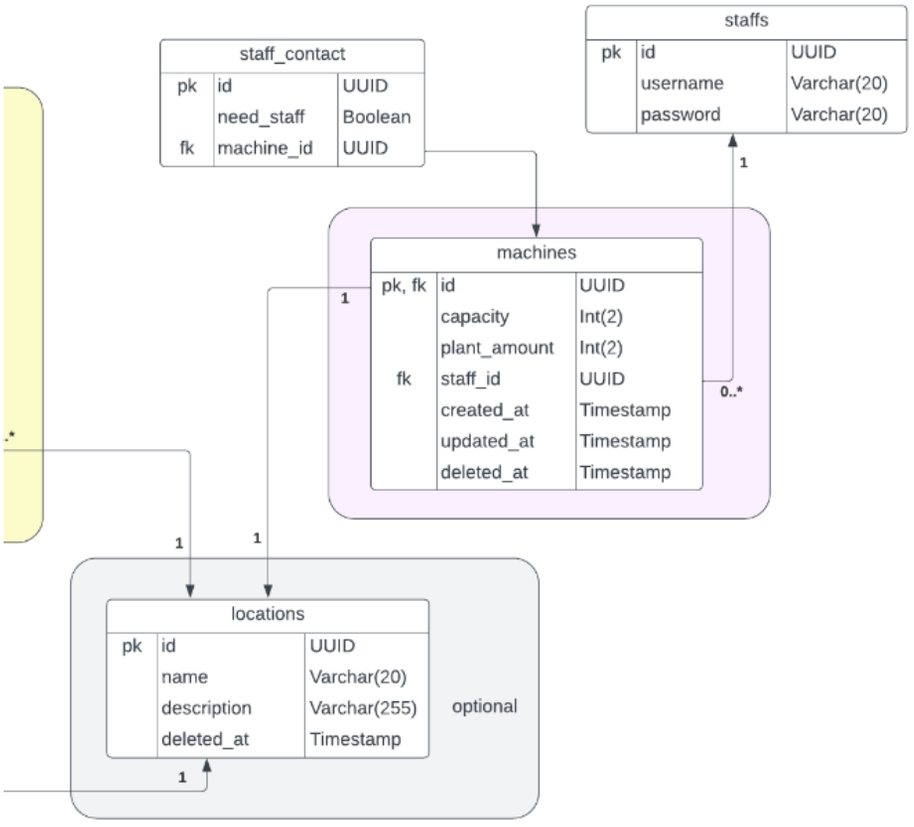

Figure 38 - Machine and Location Database

1. Machine and Location:

- The 'staff' table stores information about the staff members responsible for the automatic plant shop. It includes fields such as 'username' and 'password' for authentication purposes. Each staff member is associated with only one machine.

- The 'machine' table represents the individual machines in the automatic plant shop. It includes attributes like 'machine_id', 'location', and other relevant details specific to each machine. Each machine is associated with a staff member responsible for its management.

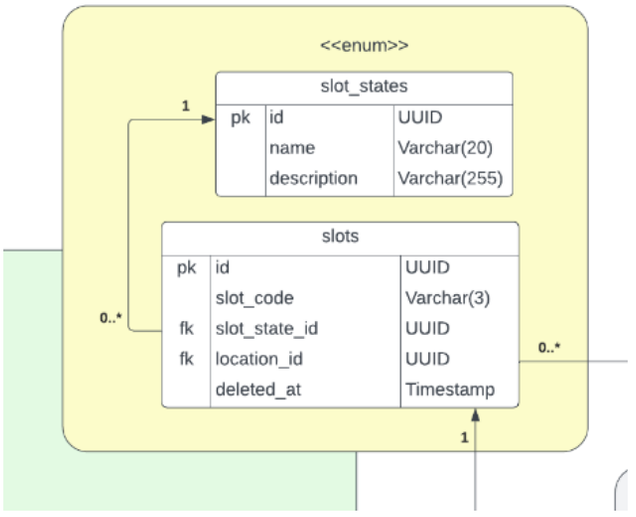

Figure 39 - Slot Database

2. Slot:

- The 'slot' table represents the slots within each machine. It keeps track of the availability and status of each slot. Each slot is associated with a specific machine and can hold one plant.

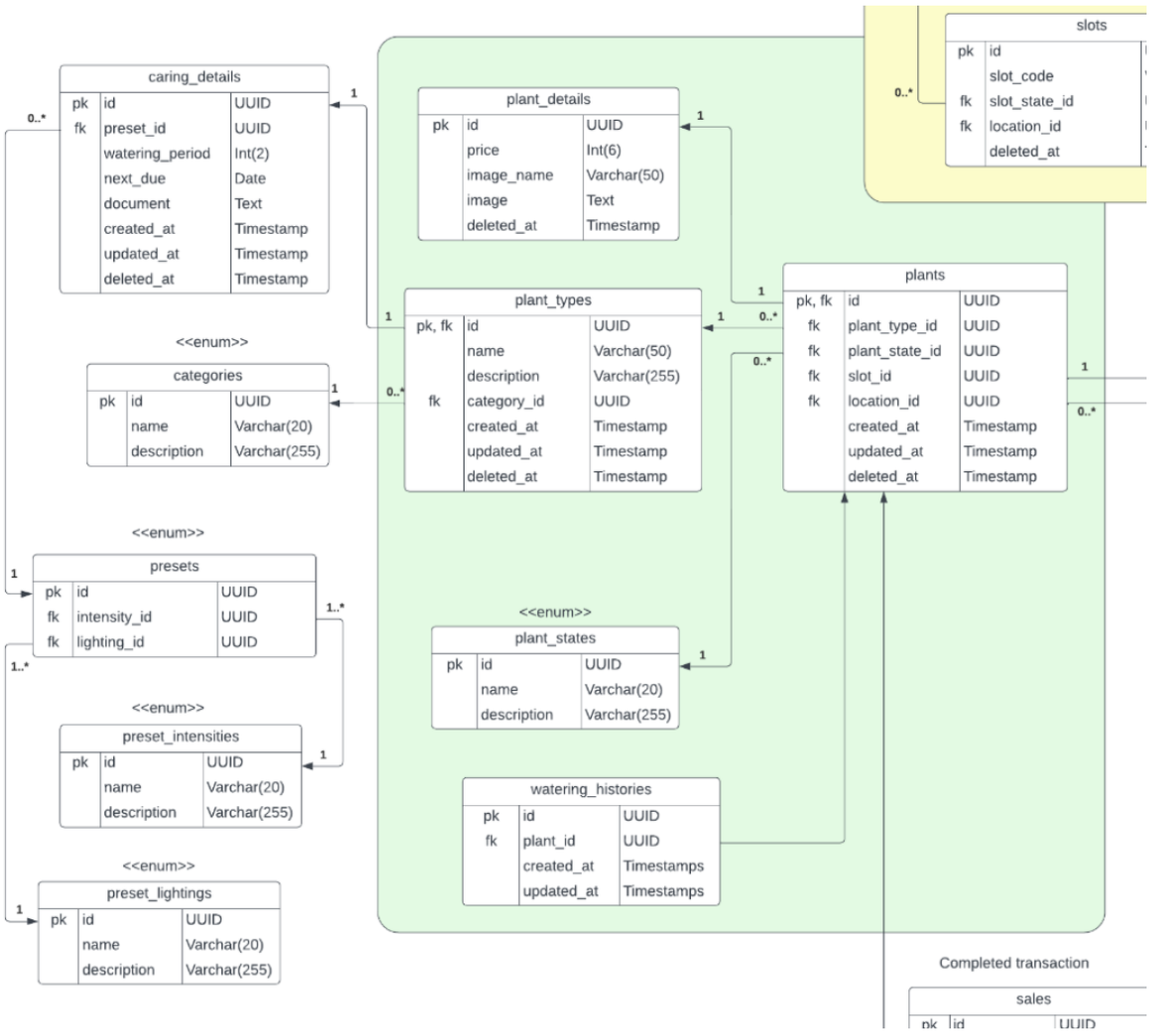

Figure 40 - Plant Type and Plant Database

3. Plant Type:

- The 'plant_type' table stores information about different types or categories of plants available in the automatic plant shop. It includes fields such as 'category', 'plant_care_info', and other relevant attributes specific to each plant type.

4. Plant:

- The 'plant' table represents individual plants in the inventory. It includes attributes such as 'plant_id', 'plant_type_id', 'plant_state', 'watering_history', and other specific details of each plant. Each plant is associated with a plant type.

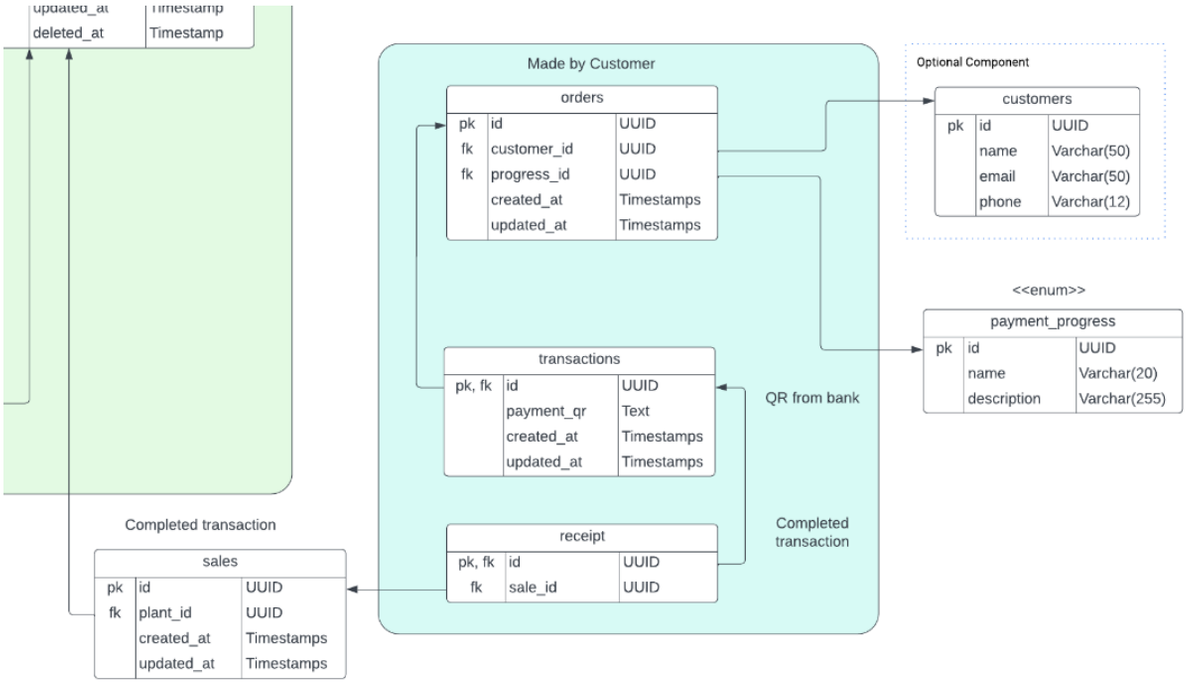

Figure 41 - Payment Transaction Database

5. Order History:

- The 'order_history' table keeps track of the orders placed by customers. It includes information such as 'order_id', 'customer_id', 'timestamp', and 'payment_progress'. This table allows for tracking the progress of payments for each order.

6. Transaction:

- The 'receipt' table is associated with completed transactions, including details such as 'transaction_id', 'order_id', 'transaction', and other relevant transaction-related information. This table records the completed transactions, providing a record of each purchase.

4.4.4 Hardware Controller

The hardware controller, often based on a Raspberry Pi, acts as the central control unit of the automatic plant shop. It communicates with other components, such as the server and physical mechanisms, to facilitate the operation of the shop. The hardware controller receives instructions from the server, controls the opening and closing of the shop's door, and manages other physical actions necessary for plant selection and retrieval.

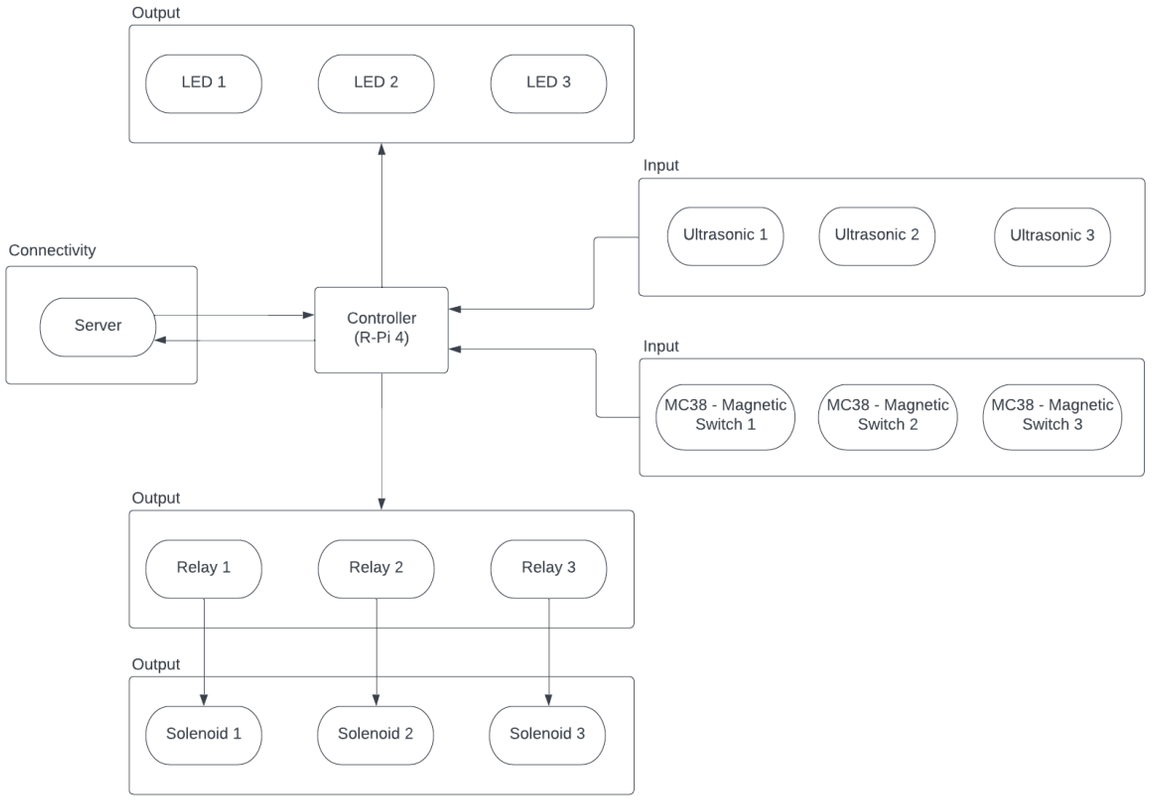

Figure 42 - Controller Block Diagram

4.5 MQTT Networking

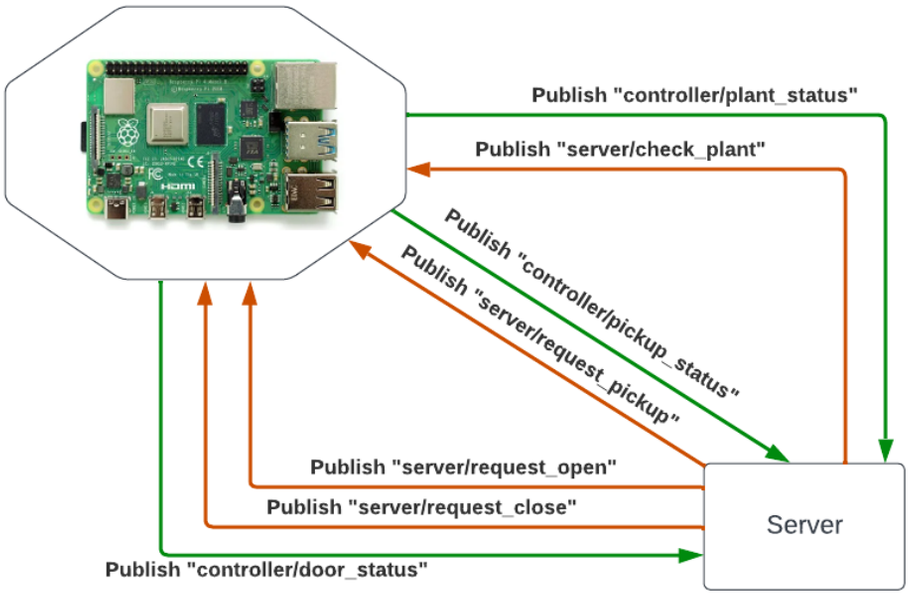

Figure 43 - Controller's MQTT Networking

This diagram shows how clients communicate with each other over the MQTT protocol using the "publish" and "subscribe" models.

Chapter 5

Software Development

5.1 Development Process

Over the course of four months, the entire project of the automatic plant shop has undergone a comprehensive redesign. This redesign involved a significant shift in the tools and technologies used, requiring the team to conduct a research and exploration of new solutions. Here's an explanation of the project's redesign process:

1. ***Evaluation of Existing System (Jan)***:

- At the start of the redesign process, the team assessed the existing system's strengths, weaknesses, and limitations. This evaluation provided insights into the areas that needed improvement and served as a foundation for the redesign efforts.

2. ***Identification of Requirements and Goals (Jan)***:

- The team defined the project's requirements and set clear goals for the redesign. This involved considering user expectations, and understanding the evolving needs of the automatic plant shop system.

3. ***System Designing (Feb):***

- This phase involved designing a database and user interface to match each group of users of the new system.

4. ***Research and Exploration (Mar)***:

- With the decision to change tools and technologies, the team embarked on an extensive research phase. We explored various options and evaluated potential alternatives that aligned with the project's requirements and goals.
- Research involved studying different frameworks, libraries, databases, and other technologies that could enhance the functionality, scalability, and performance of the automatic plant shop system.

5. ***Development and Integration (Apr - May)***:

- Following the research phase, the team began the development and integration of the redesigned system. They utilized their findings and insights to implement the new tools and technologies into the project.
- This phase involved coding, configuring, and integrating the components of the new system, including the backend server, user interfaces, database, and hardware controller.

6. ***Iterative Refinement (late May)***:

- Throughout the four-month redesign process, the team iteratively refined and improved the system. They also gathered feedback from users, and made necessary adjustments to enhance the performance, usability, and reliability of the automatic plant shop.

5.2 Backend Development

In the development of the automatic plant shop project, the backend was designed and structured into four distinct layers, each serving a specific purpose. These layers work together to ensure efficient and organized handling of data and requests. The figure below illustrates the combination and interaction of these layers:

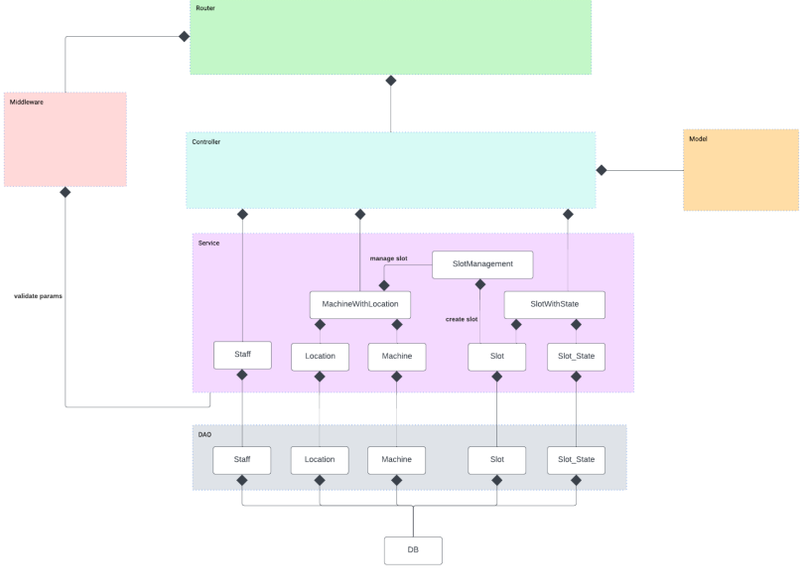

Figure 44 - Simplified Example of Backend Layers

5.2.1 Slot Management Section

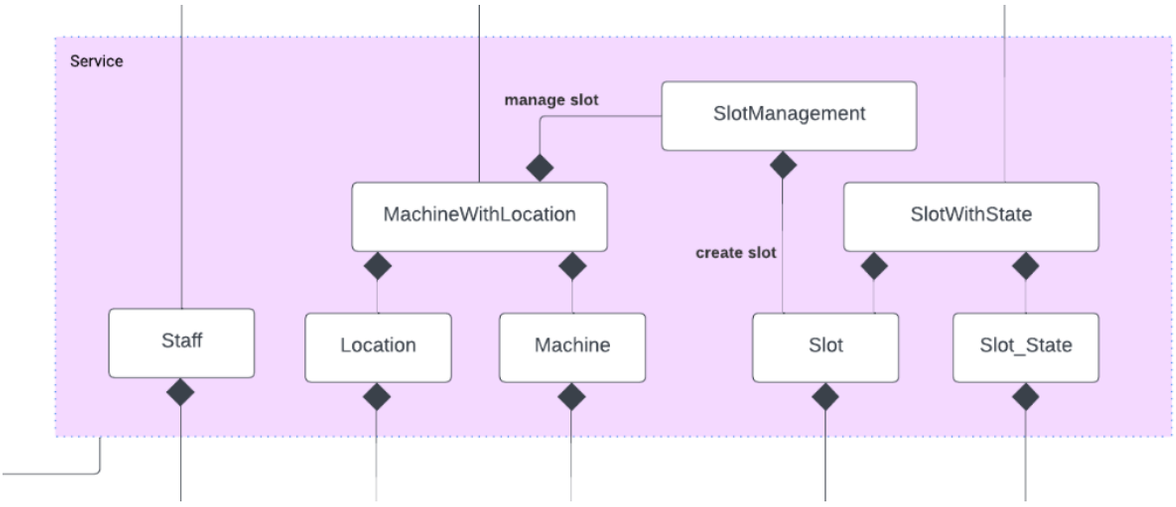

Figure 45 - Backend Service's Slot Management Section

The 'slot management section' is responsible for handling the management of slots within the machine. This section includes the 'slot management service', which is specifically designed to handle the allocation and management of slots once a machine is created.

5.2.2 Plant Type Management Section

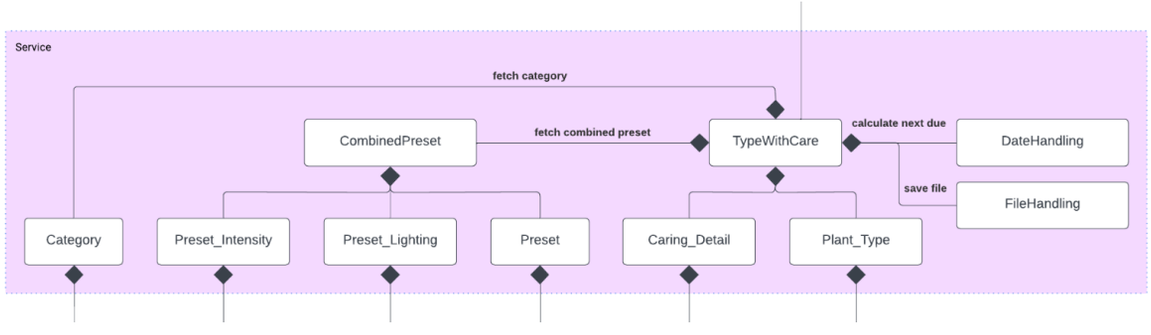

Figure 46 - Backend Service's Plant Type Management Section

The 'plant type management section' is responsible for managing the different types of plants available in the system. This section includes the 'type with care service' which provides functionalities related to plant types and their associated care information. Additionally, it utilizes a 'preset module' to fetch care presets and handles data related to dates and files.

5.2.3 Plant Data Management Section

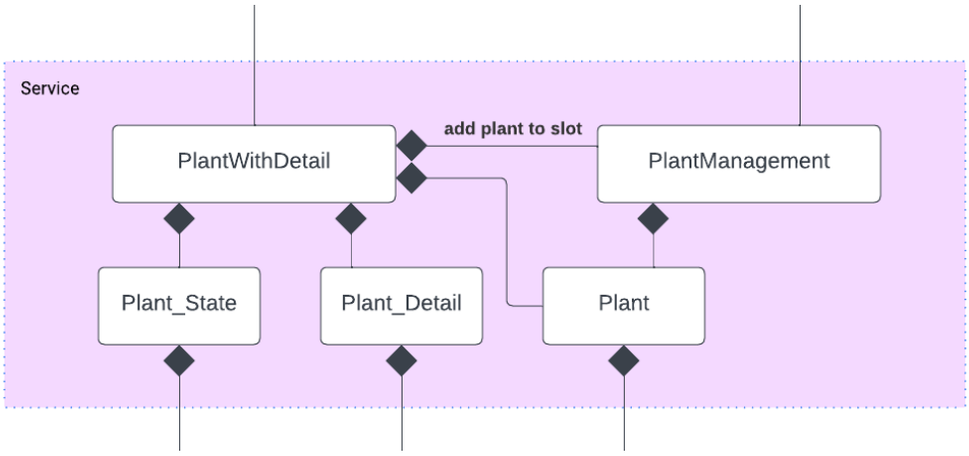

Figure 47 - Backend Service's Plant Data Management Section

The 'plant data management section' is responsible for managing the data of individual plants within the system. This section includes the 'plant management service' which facilitates the addition of plants to one of the available slots once the staff inserts the plant's data into the system.

5.2.4 State Management Service

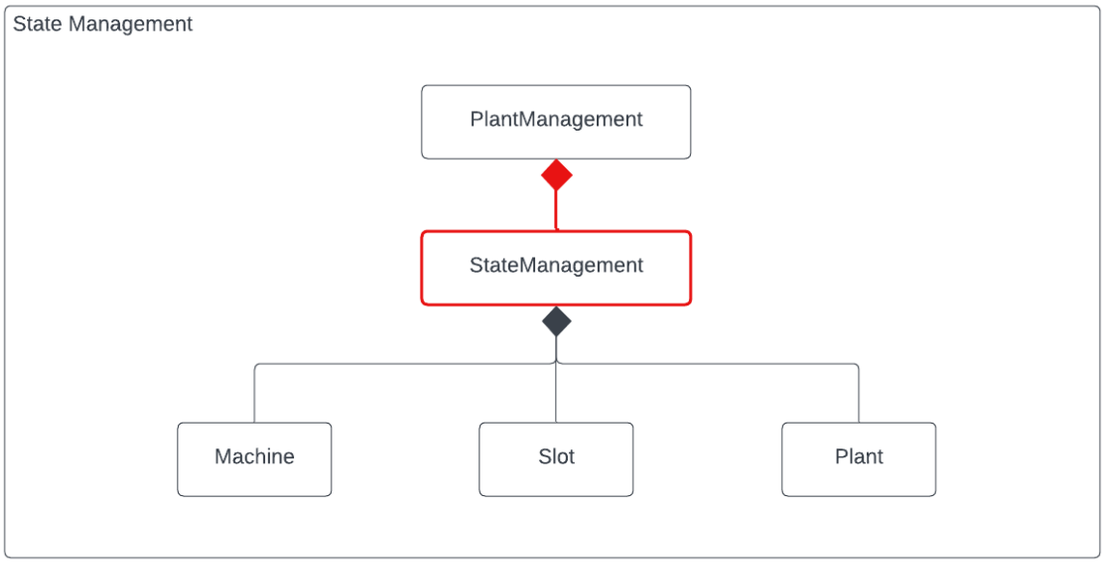

Figure 48 - Backend - State Management Service

In addition to the 'plant management service' discussed earlier, the backend of the project also includes a 'state management service'. This service is responsible for updating the state of both plants and slots within the system.

5.3 State Design

The automatic plant shop project incorporates a 'state design' approach for updating the states of slots and plants within the system. This design ensures efficient management and tracking of the current conditions and statuses of slots and plants.

5.3.1 Slot State

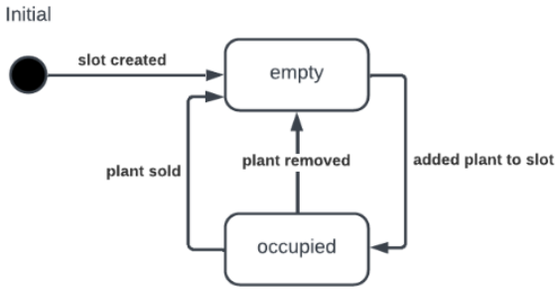

Figure 49 - Slot State

5.3.2 Plant State

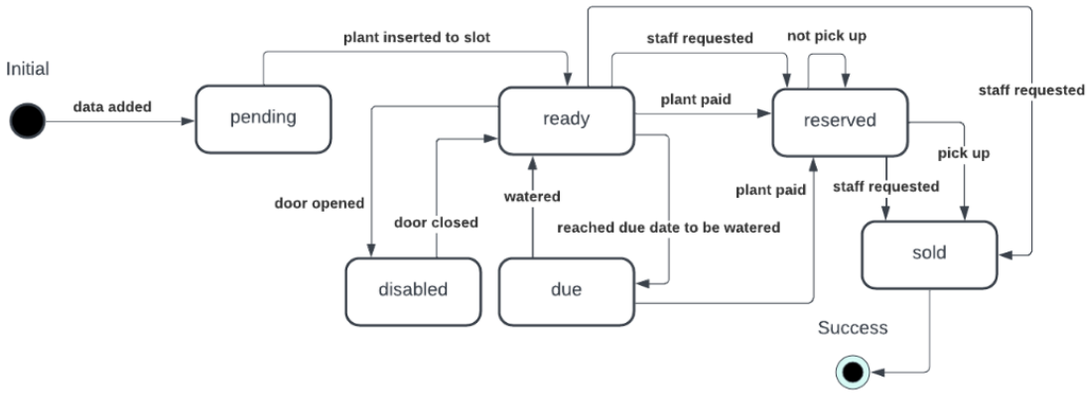

Figure 50 - Plant State

5.4 Components Communication

In the automatic plant shop project, certain tasks and operations require communication between the hardware components within the system. This communication allows for the coordination and synchronization of actions, ensuring the proper functioning and operation of the hardware components.

5.4.1 Request Unlock Slot

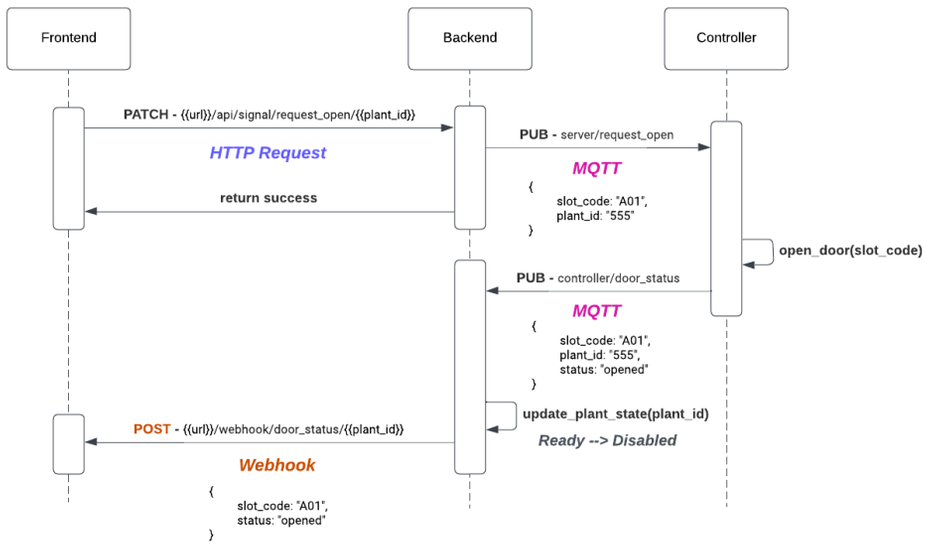

Figure 51 - Request Unlock Slot Flow

- When a staff selects a plant, the client-side user interface sends a request to the hardware controller to open the corresponding slot door.

- The request includes information such as the plant id or identifier to identify the specific slot associated with the selected plant.

- The client-side user interface communicates this request to the backend server, which in turn sends a command or message to the hardware controller to initiate the opening of the slot door.

5.4.2 Request Lock Slot

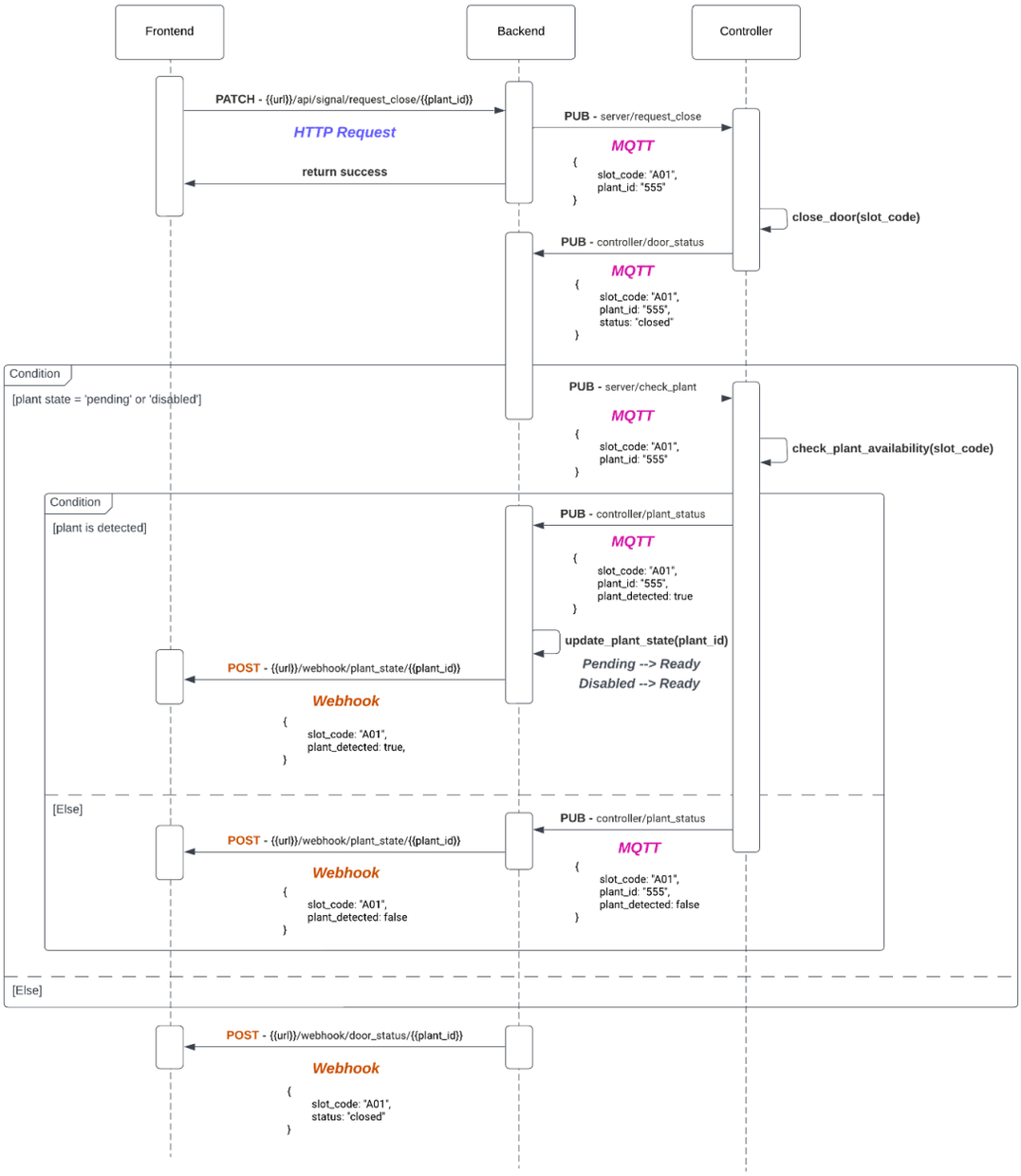

Figure 52 - Request Lock Slot Flow

- Once the staff has made their selection or if the slot door needs to be closed for any other reason, the client-side user interface sends a request to the hardware controller to close the slot door.

- Similar to the request for opening the slot door, the client-side user interface communicates this request to the backend server, which then sends a command or message to the hardware controller to close the slot door associated with the request.

5.4.3 Payment Processing

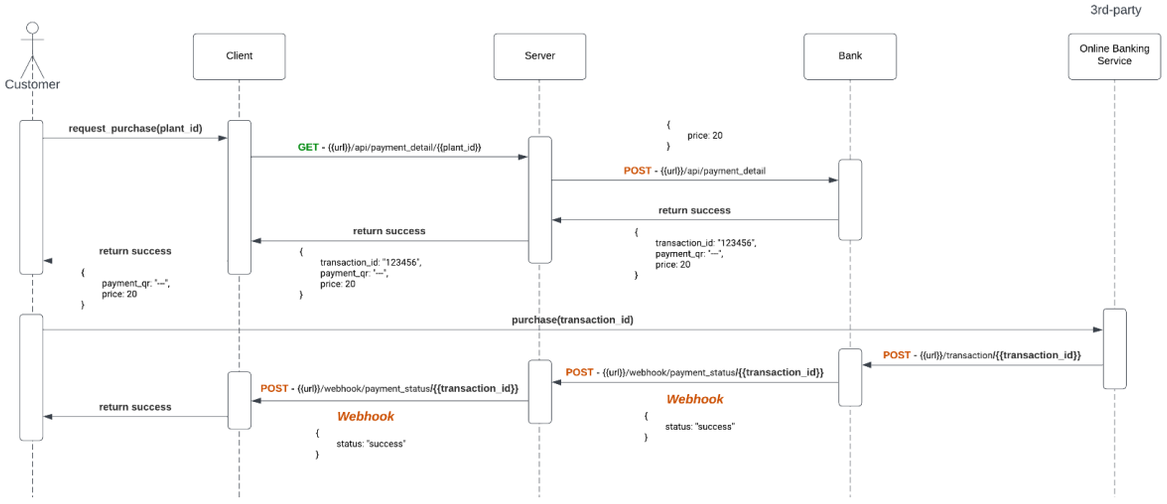

Figure 53 - Payment Processing Flow

- In the automatic plant shop project, the payment transaction for purchasing plants is confirmed through a bank webhook request. A webhook is a mechanism that allows communication between different systems or applications in real-time. In this case, the webhook is used to receive and process payment confirmation from the bank.

- When a customer initiates the payment process for a plant purchase, the necessary payment details are sent to the bank for verification and processing. Once the payment is successfully completed, the bank triggers a webhook request to the automatic plant shop system, notifying it about the payment confirmation.

5.4.4 Request Pick Up

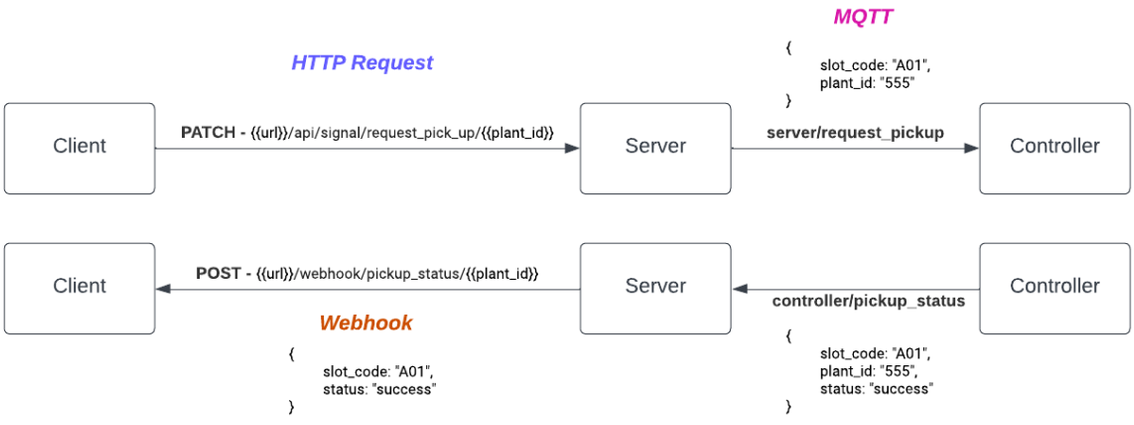

Figure 54 - Request Pick Up Flow

- After the customer has completed the purchase, the client-side user interface can send a request to the hardware controller to allow the customer to pick up their purchased plant.

- This request indicates that the customer is ready to collect their plant from the slot.

- The client-side user interface communicates this request to the backend server, which then sends a command or message to the hardware controller to enable the pickup process, such as unlocking or releasing the plant from the slot.

5.5 Hardware Controller

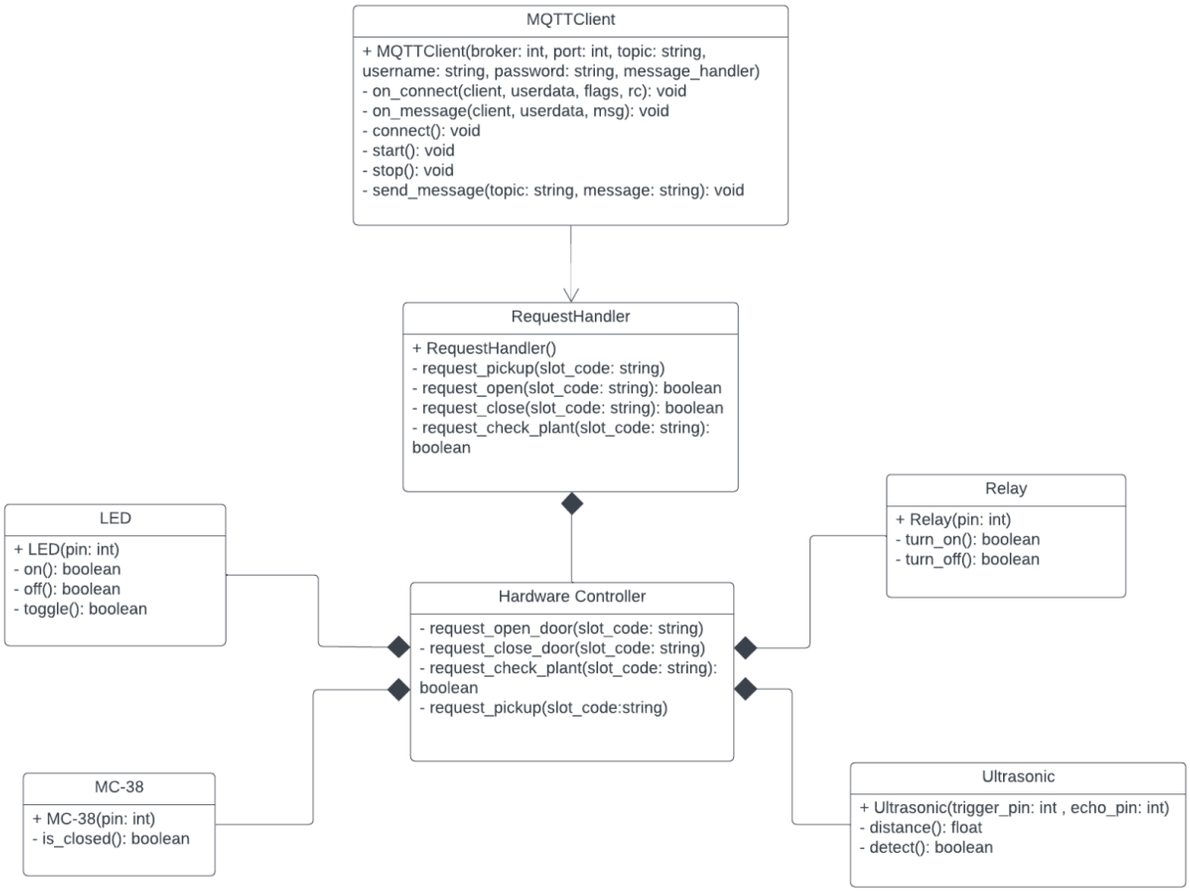

Figure 55 - Controller Class Diagram

In the class design of the hardware controller software, we have adopted a component-based approach where each hardware part is represented as a separate component. This design allows for modularization and encapsulation of functionality, making it easier to manage and maintain the hardware controller system.

Chapter 6

Results

6.1 User Interface

The result of the user interface (UI) meets certain specifications and requirements. However, due to time constraints and the need to complete the project within the given timeframe, some sections of the UI have been changed to ensure timely completion and the ability to demonstrate the functionality.

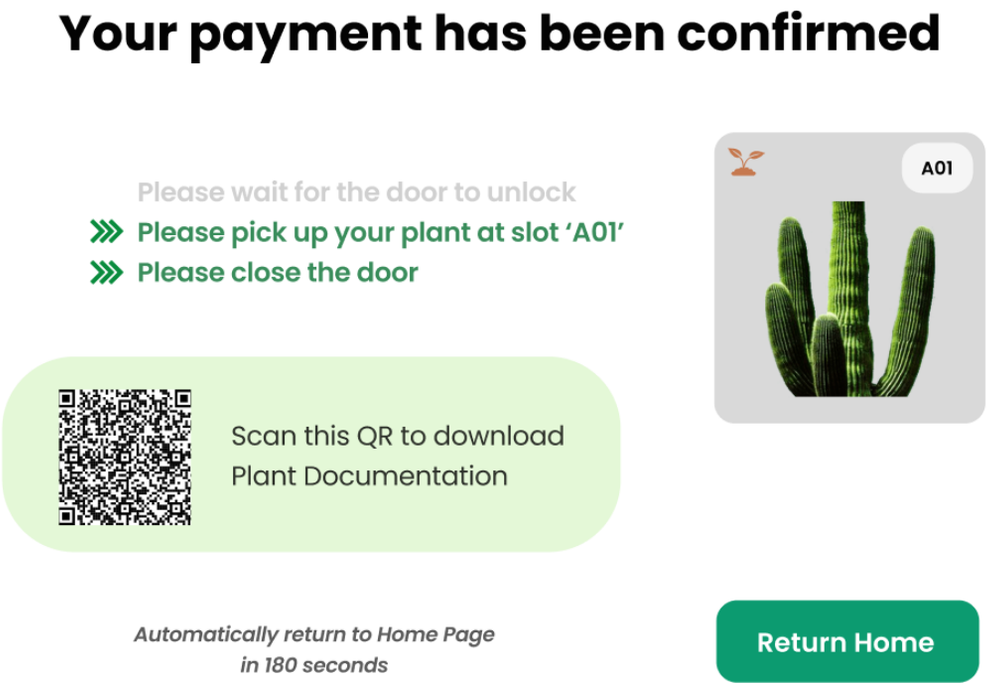

Figure 56 - Adapted Payment Confirmation UI

One of the changes made to the user interface is the simplification of the "Confirm Payment" section. Originally designed with three steps, the section was modified to combine the process into a single step while retaining all the necessary functionality.

6.2 Backend Service

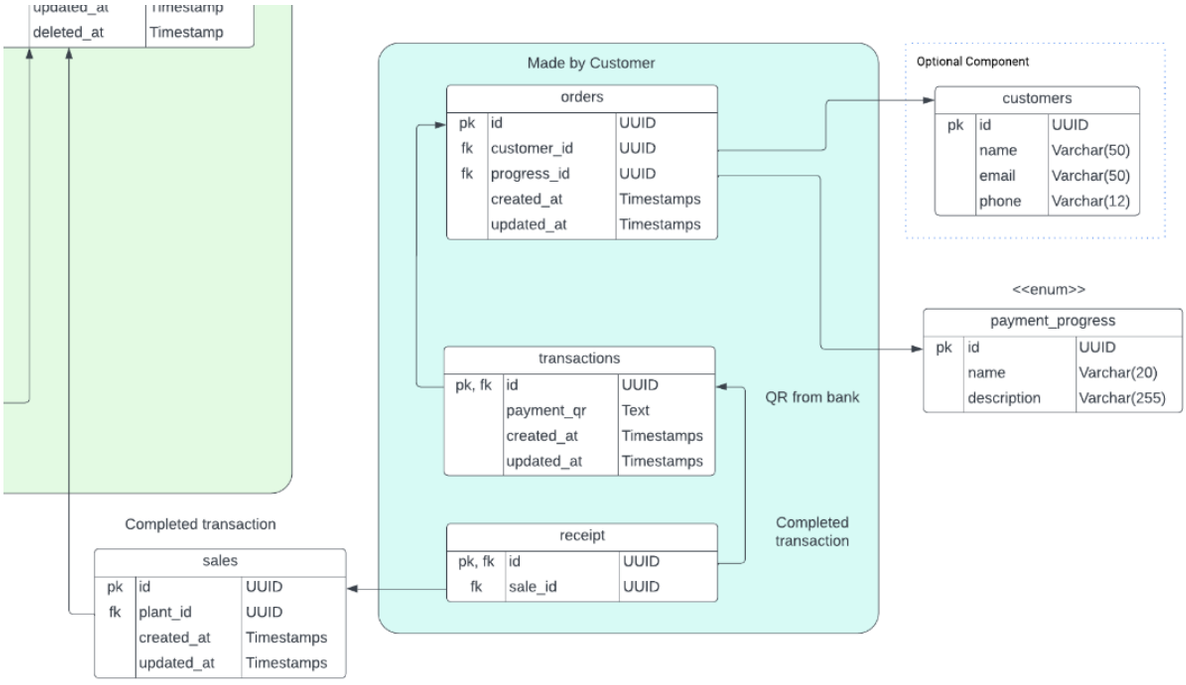

Figure 41 - Payment Transaction Database

Within the time limit of the project, it is important to note that the storage of transaction data by the backend has not been implemented yet. Although the transaction data plays a crucial role in tracking and recording completed purchases, due to time constraints, this specific functionality has not been fully developed.

6.3 Hardware Model

The shop machine hardware model is designed with an automatic unlocking mechanism that is triggered once the customer has successfully completed the payment for their chosen plant. This mechanism ensures a seamless and convenient experience for the customers. Additionally, the door slot of the machine is transparent, allowing customers to have a clear view of the plants available in each slot.

Figure 57 - Machine Model

Figure 58 - Inside the Machine Slot

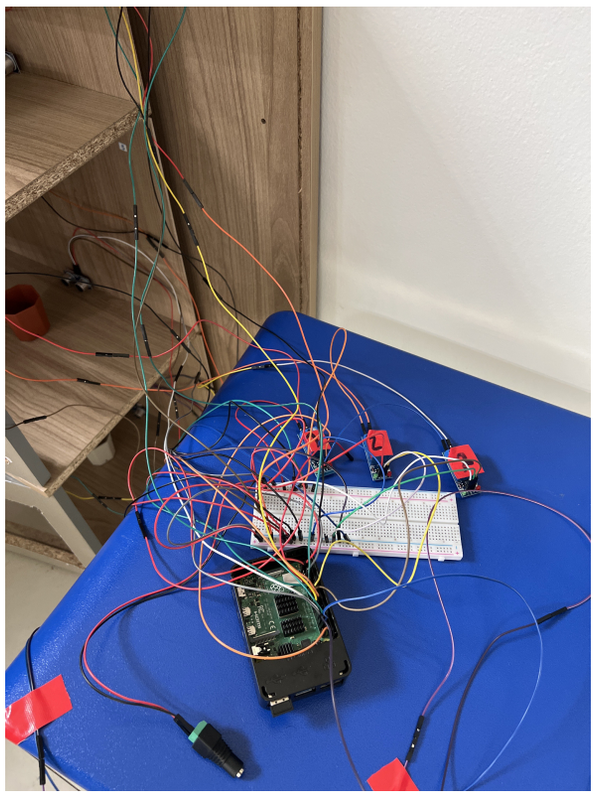

Figure 59 - Hardware Controller (Raspberry Pi 4)

Chapter 7

Conclusion and Future Work

In conclusion, the automatic plant shop project has successfully developed a novel and innovative solution to address the limited accessibility to plant markets in certain regions. By creating a mobile plant shop with an automatic unlocking system and transparent door slots, the project aims to encourage people to grow plants and raise awareness about the current environmental challenges.

While the project has achieved its primary objectives within the given timeframe, there are several potential future enhancements that can further improve the automatic plant shop:

- Implement a customer support feature that allows customers to call staff directly through the user interface in case of any issues or inquiries.

- Develop a reservation system that enables customers to reserve specific plants from their homes, ensuring availability upon their visit to the plant shop.

- Integrate a mapping functionality that shows the location of different plants within the shop, providing customers with a visual representation and easy navigation to find their desired plants.

By addressing these future tasks, the automatic plant shop can enhance the customer experience, and expand its reach to a broader customer. With continuous improvements and updates, the project has the potential to contribute to the promotion of green living and environmental awareness in the community.

Reference

14 ตลาดต้นไม้ ราคาถูก น่าช้อป ไว้แวะซื้อไปจัดสวนชิล ๆ. kapook.com. (2020, June 23). https://home.kapook.com/view53551.html

14, D. S. S. | J. (2020, July 14). *Bring nature into your home with Bangkok's Best Plant Markets*. BK Magazine Online. https://bk.asia-city.com/city-living/news/bring-nature-your-home-bangkoks-best-plant-markets

The 6 best iot hardware platforms (2021 update) – particle blog. Particle. (n.d.). https://www.particle.io/blog/iot-hardware-comparison-guide

Admin. (2020, October 28). *How HTTP request and Response Works*. BytesofGigabytes. https://bytesofgigabytes.com/networking/how-http-request-and-response-works

Admin. (2023, February 12). *What is REACT JS?*. FAQs. https://www.faqs.com.pk/what-is-react-js

Alves, R. (2023, March 2). *Get started with the Pern Stack: An introduction and implementation guide*. Medium. https://medium.com/@ritapalves/get-started-with-the-pern-stack-an-introduction-and-implementation-guide-e33c55d09994

Antonieta, Y. (2023, March 5). *NodeJs layered architecture*. ctrl. https://ctrly.blog/nodejs-layered-architecture

Denchak, M. (2022, May 23). *Are the effects of global warming really that bad?*. Consequences and Effects of Global Warming -- What is the Impact? https://www.nrdc.org/stories/are-effects-global-warming-really-bad

GeeksforGeeks. (2021, September 20). *State the core components of an HTTP response ?*. GeeksforGeeks. https://www.geeksforgeeks.org/state-the-core-components-of-an-http-response

HTTP primer for Frontend developers. Web Dev Drops. (2017, August 29). https://www.webdevdrops.com/en/http-primer-for-frontend-developers-f091a2070637

Impacts of global warming. WWF Australia. (n.d.). https://wwf.org.au/what-we-do/climate/impacts-of-global-warming

MQTT. Paessler. (n.d.). https://www.paessler.com/it-explained/mqtt

posttoday. (2018, November 26). *“กรีน อัพ” ตู้ขายต้นไม้อัตโนมัติ*. posttoday. https://www.posttoday.com/business/571985

Silva, M. H. da. (2020, September 14). *Creating a secure node.js REST API: Toptal®*. Toptal Engineering Blog. https://www.toptal.com/nodejs/secure-rest-api-in-nodejs

Somkiat, & Article by Somkiat PuisungnoenTo be Craftmanship. (n.d.). *ว่าด้วยเรื่องของ serial ใน PostgreSQL*. cc somkiat. https://www.somkiat.cc/serial-in-postgresql

Sufiyan, T. (2023, May 16). *What is node.js: A comprehensive guide*. Simplilearn.com. https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs

Vending Machines vector art PNG, Valentine S Day vending machine flowers, valentine S day, Tanabata, romantic PNG image for free download. Pngtree. (n.d.). https://pngtree.com/freepng/valentine-s-day-vending-machine-flowers_6971546.html

Webhooks Overview. Shopify. (n.d.). https://shopify.dev/docs/apps/webhooks

What is a webhook & how to use it to track email marketing activity. Brevo (ex Sendinblue). (2023, May 16). https://www.brevo.com/blog/what-is-a-webhook

What is the client side?: Definition by Elementor. Elementor. (2021, November 15). https://elementor.com/resources/glossary/what-is-the-client-side

มาแล้ว! “ตู้ขายต้นไม้อัตโนมัติ” แห่งแรกในไทย ต้นจริง ดอกไม้จริง พร้อมปลูกได้เลย. Mango Zero. (2018, February 15). https://www.mangozero.com/tree-vending-machine-green-up-world