# PLANT MANAGEMENT SYSTEM

Maea, Tiana Leata

# Table of Contents:

Customer Problem Statements & System Requirements..... 1

**Functional Requirement Specification .....4**

System Sequence Diagram (SSD) for Updating Plant Care Information.....4

Activity Diagram for Adding a New Plant.....5

Project Plan..... 6

## Customer Problem Statements & System Requirements

**Problem Statement:**

- Many plant owners often struggle with keeping track of their plants' specific care needs, especially when they have several plants in their home. This can include everything from watering schedules, sunlight requirements, to soil preferences. It is challenging to manage multiple plants with different care routines and needs, especially when someone has a busy schedule. To have a simple, yet interactive, system that helps with organizing plants, receive reminders for their care, and access reliable information on best care practices for each plant type, would benefit a household plant owner. A digital platform that acts like a personal plant library would ensure their plants can thrive and reduce the stress of plant care management.

**Glossary of Terms:**

- Plant Library: A digital collection of plants owned by the user, including plant types, care routines, and other details.
- Watering Alert: A notification sent to the user to remind them of when to water their plant(s).
- Care Database: A built-in repository containing best care practices for different types of plant types.
- Soil Preferences: Specific types of soil required for different plants to thrive in.
- Sunlight Requirements: The amount of sunlight each plant needs daily for healthy growth.

**System Requirements:**

Functional Requirements

| No. | Priority Weight | Description |
|---|---|---|
| REQ-1 | High | The system should allow users to add, edit, and delete plant entries within their library. |
| REQ-2 | High | The system should send watering alerts based on the plant's watering schedule. |
| REQ-3 | High | Users can input plant details, including name, type, soil preferences, and sunlight needs. |
| REQ-4 | High | System should provide care suggestions to improve or correct plant care data. |
| REQ-5 | Medium | Users should be able to submit suggestions to improve or correct plant care data. |
| REQ-6 | Medium | Users should be able to search and filter their plant library by type, care needs, or name. |
| REQ-7 | Low | The system should allow users to upload images of their plants. |

Nonfunctional Requirements

| No. | Priority Weight | Description |
|---|---|---|
| NFR-1 | High | The system should have an intuitive interface for easy navigation. |
| NFR-2 | High | The system should reliably send timely alerts without delays. |
| NFR-3 | Medium | The system should securely store plant data and user contributions. |
| NFR-4 | Medium | The system should perform efficiently even with large amounts of data. |

User Interface Requirements

| No. | Priority Weight | Description |
|---|---|---|
| UI-1 | High | Home screen displaying the plant library with options to add or remove plants. |
| UI-2 | High | Plant detail page showing care requirements, next watering date, and user suggestions. |
| UI-3 | Medium | Contribution page where users can suggest changes or additions to care data. |
| UI-4 | Medium | Search and filter feature for finding plants by type, care needs, or name. |

## Functional Requirement Specification

- The system should provide an easy-to-use interface for managing plant collections.
- Users should be able to receive real-time notifications for plant care.
- The search functionality should support filtering by plant type, watering frequency, and light requirements.
- System interactions should be logged to track plant care history.

## System Sequence Diagram (SSD) for Updating Plant Care Information

**Actor:** User

**Objects**: Plant System, Plant Database

**Steps of Interaction:**

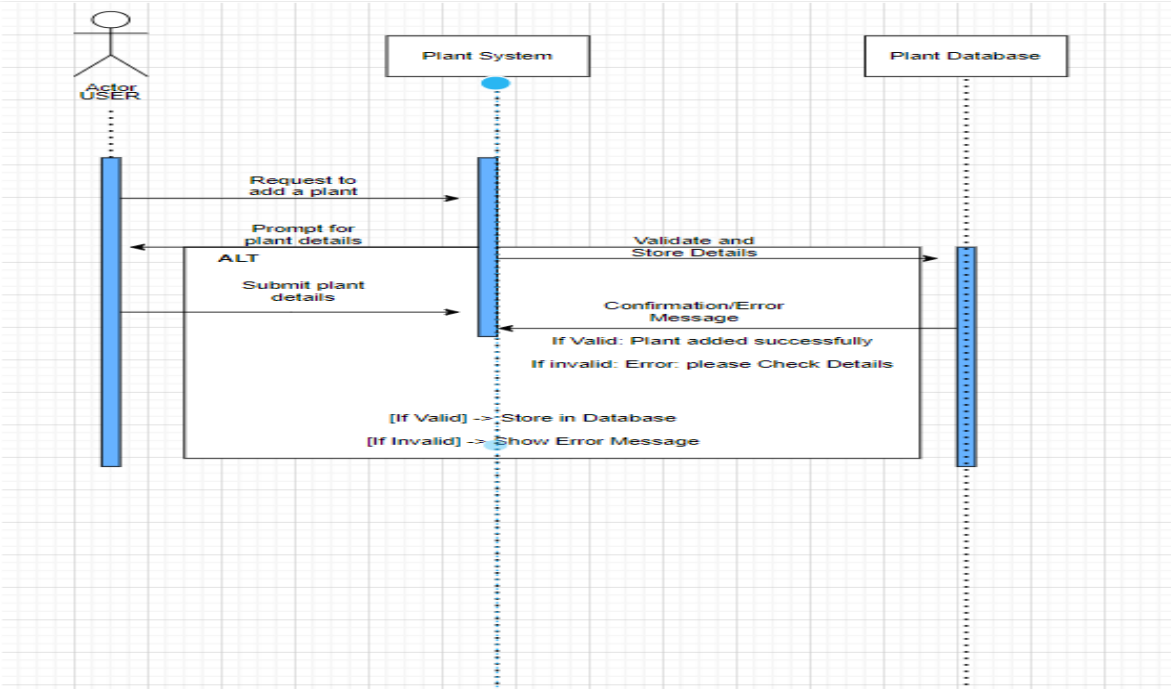
1. The User selects an existing plant from their list.
2. The Plant System retrieves plant details from the Plant Database.
3. The User updates the care information (e.g., new watering schedule).
4. The Plant System validates the update:

- If valid, proceed to Step 5.
- If invalid, display an error message.

5. The Plant System saves the updated information in the Plant Database.
6. The Plant System confirms that the update was successful.

**Alternative Scenarios:**

- If the update fails, the system prompts the user to retry.
- If there is an issue with the database, an error message is displayed.




**Activity Diagram for Adding a New Plant**

**States:**

- Initial State: The user opens the plant system.
- Final States:
  1. The plant is successfully added.
  2. The user enters invalid data and receives an error message.

**Actions:**

1. The User accesses the plant entry system.
2. The System prompts the user to enter plant details.
3. The User submits the details.
4. The System validates the input:
    - If valid, the plant is added to the database.
    - If invalid, the system prompts the user to correct the data.
5. The System confirms the plant has been added.

## Project Plan:

**Progress Made So Far:**

1. **Weeks 1-2:** Completed planning and system design. Defined system requirements and created UML diagrams. Initial database schema drafted.
2. **Weeks 3-4:** Set up the development environment and project repository on GitHub. Began backend implementation, including CRUD operations for plant management.
3. **Week 5:** Continuing backend development and database integration. Implemented basic plant library functionality.

**Remaining Development Timeline (15 Weeks Total):**

4. **Weeks 6-8:** Complete backend implementation and begin frontend development. Design UI using HTML, CSS, and JavaScript.
5. **Weeks 9-10:** Implement notifications, search functionality, and integrate the care database.
6. **Weeks 11-12:** Conduct testing and debugging to refine system performance.
7. **Weeks 13-14:** Optimize the UI for better user experience and finalize documentation.
8. **Week 15:** Prepare for project submission and presentation.

## References:


Elmasri, R., & Navathe, S. (2017). *Fundamentals of database systems* (7th ed.). Pearson.

Duckett, J. (2011). *HTML and CSS: Design and build websites*. Wiley.

Duckett, J. (2014). *JavaScript and jQuery: Interactive front-end web development*. Wiley.

W3Schools. (n.d.). *Java tutorial*. Retrieved from https://www.w3schools.com/java/