

# 496 Software Project: Plant Watch

Brenden Bokino and Nishant Gurung

2026-01-21

## 1 Client Information

By sharing this client information and the rest of this document, you are stating that this client has provided this project as something they want (not something you created and asked if they wanted), and that they are interested in having you complete this project for your capstone.

- Client name: Dr. Maren Blohm
- Client title: Professor of Biology
- Client email address: mblohm@loyola.edu
- Client employer: Loyola University Maryland
- How you know the client: Loyola professor, met in person.

## 2 Project Description

### 2.1 Overview

Our project will be to build a mobile application that can be used to manage plant life. The app will allow users to scan plants and assess their condition, suggesting potential watering schedules and nutritional supplements to help their growth. The app will also allow users to manage greenhouses and interact with devices that might be used in them. Some of the devices Dr. Blohm is interested in are devices that can monitor moisture levels in soil, weather conditions, and automatic sprinklers.

These devices would be in the greenhouse and the garden beds we have around the campus. The information provided by our software will help them make better decisions about caring for these plants. It will also save time for the department, requiring them to check on the plants less often. With the Biology Department relocating, having to travel to the greenhouses less would be ideal.

### 2.2 Key Features

1. Scanning a plant to identify it and possibly note nutrient deficiencies and watering needs.
2. Individualized profiles for users that lets them manage their greenhouses and their devices.
3. Connect with and pull information from devices in the greenhouses such as weather monitors and moisture monitors.
4. Allow users to activate devices remotely or configure devices to activate in certain conditions (ex. Automatically activate sprinklers when moisture in the soil is below a certain threshold)

## 2.3 Why this Project is Interesting

This project bridges the two fields of Computer Science and Biology by showing how technology can contribute directly to environmental conservation efforts. This capstone gives us a chance to make a real, measurable impact in preserving the biodiversity of Loyola's campus. Additionally, it will build a model that can be adapted by other institutions seeking to monitor and protect the nature around them. The skills that will be learned are interesting as well. Computer vision is something we both do not have worked with and learning to connect different devices together will be a good experience.

## 2.4 Areas of CS required

1. Mobile app development
2. Computer Vision
3. Networking
4. Database

## 2.5 Potential Concerns and Questions

1. We are unsure of the difficulty of developing with computer vision. There are likely resources out there on it, but we are concerned with how accurate it can get.
2. We are unfamiliar with the devices used when monitoring plants and how programmable they will be.

## 2.6 Summary of Efforts to Find a Project

(Not necessary for 482) [Briefly list out when/how you've discussed with this client, and if you've discussed with other clients who either didn't work out or didn't respond. If you considered a different project and it didn't work out, why didn't it work out?]

We've been discussing about this project with Dr. Blohm since the last semester. She has guided us towards the right direction and answered our questions about the greenhouse, plant health, and the requirements of the app.

## 2.7 Comparison to Draft

The project was initially proposed by me (Nishant) during the drafting phase. While the core concept of building an app that uses image recognition to inform the users about its water cycle, we've changed the proposal to incorporate more fields of Computer Science. The initial draft was largely centered on Computer Vision, particularly image recognition for species identification. But the present proposal is not limited to a single subfield; instead, it is divided into several subfields of Computer Science that include:

- Computer Vision for automated species detection
- Database design and management for storing and organizing plant observations, weather patterns, and health assessment data to enable long-term tracking of garden performance.
- Networking for real-time data synchronization with the weather monitoring devices and multi-user access, which will send immediate alerts on temperature drops, and soil moisture to prevent crop loss.
- Mobile App Development for creating an intuitive, user-friendly interface.

# 3 Requirements

## 3.1 Non-Functional Requirements

Table 1 presents the NFRs for this software project.

ID	NFR Title	Category	Description
NFR1	App Launch Time	Performance	The application must launch and display the home screen in under 3 seconds.
NFR2	Plant Identification Accuracy	Reliability	The plant identification feature must have a minimum accuracy rate of 85% across common plant species.
NFR3	Scan Process Efficiency	Usability	The scan-to-identify process must require a maximum of 3 user taps to complete.
NFR4	Outdoor Accessibility	Usability	The application must use accessible font sizes and appropriate color contrast for outdoor visibility and readability.
NFR5	Visual Design Consistency	Usability	The application must maintain a consistent color scheme with green as the primary color and white as the secondary color.
NFR6	Platform Compatibility	Compatibility	The application must support iOS and Android.
NFR7	Battery Efficiency	Performance	The application must consume less than 5% of device battery per hour during active use.
NFR8	Alert Delivery Speed	Performance	Emergency alerts for critical conditions (temperature drops, low moisture) must be delivered to users within 30 seconds of detection.
NFR9	Password Encryption	Security	Passwords should be encrypted before being stored in the database.

Table 1: Non-Functional requirements

### 3.2 Functional Requirements (User Stories)

Table 2 presents the functional requirements for this software project.

ID	Story Title	Points	Description
S0	Scan Plants for General Information	3	As a user I want to scan a plant with my camera and get information about what plant it is so I can learn more about plants.
S1	Scan Plants for Watering Schedule	5	As a user I want to scan a plant with my camera and get a watering schedule for it so I know how to care for the plant.
S2	Scan Plants for Nutritional Deficiencies	5	As a user I want to scan a plant with my camera and get information and guidance about its nutritional deficiencies so that I can take steps to improve the plant's health.
S3	Add Plants to Greenhouse by Scan	2	As a user I want to add a plant that I have scanned to my greenhouse profile so that I do not manually have to input the information I learned from the scan.
S4	View Plant Scan History	3	As a user I want to view the history of plants I have scanned so I don't have to rescan the same plant repeatedly.
S5	Create Account	3	As a user, I want to create an account so that I can access the app features.
S6	Login/Logout Feature	2	As a user I want to be able to login and logout so I can safely access my greenhouse data and plant scan history.
S7	Greenhouse Profile Management	8	As a user, I want to create a greenhouse profile so I can manage my plants and plant monitoring devices.
S8	Sync AcuRite Weather Sensor	8	As a user I want to be able to view information from my AcuRite weather sensor from the app so I know the conditions my plants are in.
S9	Add Plants to Greenhouse Manually	8	As a user I want to be able to upload the types of plants in my greenhouse, so that I can build a complete catalog and plan the layout of the greenhouse.

S10	Adding Users to Greenhouse	2	As a user I want to add other users to my greenhouse so they can view its information and manage it.
S11	Sprinkler Activation	5	As a user I want to activate the sprinklers remotely, so that I can water plants without having to go to the greenhouse, saving time and energy.
S12	User Permissions in Greenhouses	2	As a user I want to be able to create role groups with different permissions for users in the greenhouse that I own.
<b>Total:</b>		56	

Table 2: Functional requirements as User Stories.

## 4 System Design

### 4.1 Architecture

[Which type of software architecture are you team following? Layered architecture, MVC, other? What are the main modules for your software?] [Main modules are not the same as Layers. If you adopted any form of layered architecture (MVC included), then your layers already group components based on responsibility. Therefore, for modules, think about semantically related components. For example, in a parking lot, I could have a User, Payment, Parking (Vehicles), Contact/Issues modules.]

### 4.2 Diagrams

[CS482, on sprints/iterations 2-3, you need to create and update a diagram (check Iteration 2-3 assignment for which type of diagram). On CS496, since before sprint/iteration 1, you should have a class diagram and keep it up-to-date. In CS496, if your class diagram changes at each sprint, then create a Class Diagram subsection for the sprints, and show the changes; while keeping the one here the most up-to-date version. ]

### 4.3 Technology

[ Which technologies are you going to use to implement your project? This should include the chosen programming language, main frameworks/libraries, and database or data storage. Testing framework is essential here as well.]

### 4.4 Coding Standards

[Are your team going to follow any coding standards? For example, using a naming convention for Database tables (like only singular lowercase names). Another example, only allowing code with unit tests and above 60% coverage to be committed (good convention since testing is going to be evaluated). If you need inspiration to define your coding standards, the Extreme Programming approach has a set of coding, design, and test rules.]

### 4.5 Data

[What is the main structure of your data? In SQL-like databases, this would be the planning of the main tables, their attributes, and interactions with other tables (basically an ER diagram). In NoSQL databases, this would be the main collections and general attributes of the JSON you will store in each collection.]

[Tip to better find and write the data your system will need. Go back to your User Stories and for each one, think to yourself: which attributes/fields do I need to store for this to work?]

[Tip 2. When a system has many different roles for people, those are usually done in a single User table/collection. Especially when they share many common attributes/fields.]

## 4.6 UI Mocks

[Define and draw/sketch/code the main UIs your user will interact with in your software. Add your UI mocks here and a short caption about it. Do not forget about the main forms and CRUD UIs.]

# 5 Iterations

## 5.1 Iteration Planning

[In CS496, you plan all iterations beforehand. In CS482, you update the planning here at each iteration. ]

Table 3 shows the iteration planning.

It.	Dates	Stories	Points	
			Planned	Done
1	01/01 - 02/01	S0 Scan Plants for General Information, S4 View Plant Scan History, S5 Create Account, S6 Login/Logout Feature	11	–
2	02/01 - 03/01	S1 Scan Plants for Watering Schedule, S7 Greenhouse Profile Management	13	–
3	03/01 - 04/01	S3 Add Plants to Greenhouse by Scan, S9 Add Plants to Greenhouse Manually	10	–
4	04/01 - 05/01	S8 Sync AcuRite Weather Sensor , S11 Sprinkler Activation	13	–
5	05/01 - 06/01	S2 Scan Plants for Nutritional Deficiencies, S10 Adding Users to Greenhouse, S12 User Permissions in Greenhouses	09	–
Total:			56	–

Table 3: Iteration Planning for Incremental Deliveries

## 5.2 Iteration/Sprint 1

### 5.2.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

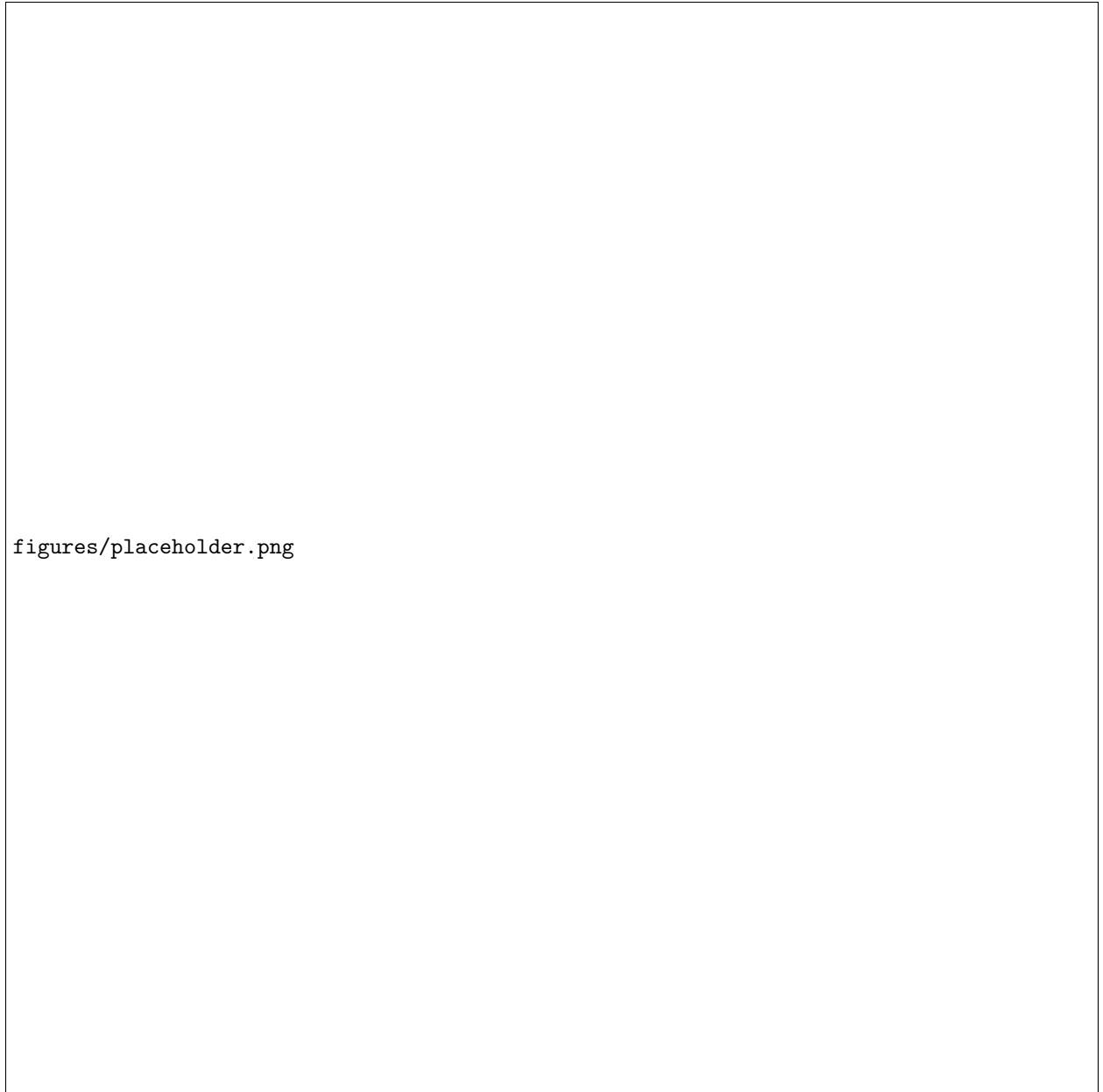
### 5.2.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

### 5.2.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

Figure 1 shows the test coverage



figures/placeholder.png

Figure 1: Iteration 1 test coverage report

#### 5.2.4 Retrospective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

## **5.3 Iteration/Sprint 2**

### **5.3.1 Planning**

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

### **5.3.2 Work Done**

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

### **5.3.3 Testing Coverage**

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

### **5.3.4 Retrospective & Reflection**

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

## **5.4 Iteration/Sprint 3**

### **5.4.1 Planning**

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

### **5.4.2 Work Done**

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

### **5.4.3 Testing Coverage**

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

### **5.4.4 Retrospective & Reflection**

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

## **5.5 Iteration/Sprint 4**

[CS496 has 5 sprints. CS482 only has only 3 sprints (remove Iterations 4 and 5 from this doc if you are writing a doc for 482)]

### **5.5.1 Planning**

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

### **5.5.2 Work Done**

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

### **5.5.3 Testing Coverage**

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

### **5.5.4 Retropective & Reflection**

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

## **5.6 Iteration/Sprint 5**

### **5.6.1 Planning**

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

### **5.6.2 Work Done**

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

### **5.6.3 Testing Coverage**

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

### **5.6.4 Retropective & Reflection**

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]



## 6 Final Remarks

### 6.1 Overall Progress

[Have you completed everything? If so, present evidence on how you brought value to your client, and the overall client satisfaction. Otherwise, estimate how much progress you done and how long it would take to finish this project. Be concrete about your progress, you know how many story points your software is, how many points you completed (this shows your progress). You also how many points your team delivers at each iteration, therefore you can estimate how many more iterations it would take to finish the leftover points (show the math).]

### 6.2 Project Reflection

[Your personal reflection on the project. What lessons did you learned. What would you have done differently? How can you do better work in future projects? You may write this as a team or per person (or both — if all your iterations were team reflections, then it would be better to write individual reflections here)]

## Appendix

[Appendix section if needed]