Senior Design 1: Divide and Conquer

Name Here //still playing with a few + better image

Group 12:

Roman Luknicki	Electrical Engineering
Rohit Goeptar	Computer Engineering
Victor M. Arteaga-Cuenca	Computer Engineering
Thomas Parker	Electrical Engineering

Reviewer Committee:

Dr. Mark Maddox
Dr. Mike Borowczak
Dr. Saleem Sahawneh
Dr. Justin Phelps

ECE Mentor:

Dr. Lei Wei Dr. Chung Yong Chan Dr. Arthur Weeks

//we also know the formatting is off, if shifts so much with 4 people editing a google doc

Chapter 1 - Executive Summary

Future home of the summary

Chapter 2 - Project Description

2.1 Project Background

For thousands of years, agriculture has depended on the careful observation of natural elements. Early civilizations like those in Mesopotamia and along the Nile developed irrigation systems to channel water to their crops, relying heavily on environmental cues to guide planting and harvesting. The success of early farming was directly tied to how well people could monitor sunlight, soil conditions, and water availability, long before the invention of modern tools or automation. This deep-rooted relationship between humans and plants formed the basis of not only agricultural development but also societal growth.

As technology and infrastructure evolved, large-scale agriculture became more industrialized, reducing the need for individuals to grow their own food. However, interest in plants never disappeared, it simply shifted. In urban areas and residential spaces, people began caring for houseplants, small gardens, and community green spaces as a way to reconnect with nature, improve indoor air quality, and enhance living environments. In particular, the rise of indoor gardening and the popularity of houseplants in the 21st century has transformed plant care into both a creative hobby and a method of personal wellness.

This cultural shift has coincided with rapid advancements in consumer technology. The growing accessibility of sensors, microcontrollers, and open-source platforms has enabled individuals to explore how everyday tasks like watering a plant can be improved through automation and real-time monitoring. As interest in sustainability and smart living has grown, more people have turned to do-it-yourself (DIY) solutions to solve problems in their personal environments. Online communities of hobbyists and students now regularly build simple systems that integrate electronics with nature, learning engineering principles in the process.

These developments represent a fusion of traditional practices and modern innovation. While the fundamentals of plant care: light, water, and attention remain unchanged, the tools available to support those tasks have become far more advanced. The background of this project lies in that intersection: where ancient agricultural wisdom meets new technological possibilities, and where hands-on experimentation helps bring the future of home gardening to life.

2.2 Motivation

Houseplants have seen a significant rise in popularity, especially in recent years, as people seek to enhance their living environments with natural beauty, improved air quality, and a sense of calm. However, despite their benefits, many indoor plants are poorly maintained due to simple but impactful mistakes. Studies and surveys have shown that a large percentage of houseplants are lost each year due to either neglect or improper care. Among the most common issues are **overwatering**, which can lead to root rot and the proliferation of fungus gnats, and **underwatering**, which stresses the plant and inhibits growth. Additionally, **improper sunlight exposure**, whether too much or too little, can result in scorched leaves, poor color, and even plant death. These challenges are not uncommon among casual plant owners and can be discouraging, especially when plants that are meant to bring joy become a source of stress.

This dilemma inspired our group of four electrical and computer engineering students to develop a smarter, more reliable approach to plant care. As engineering students, we're constantly looking for ways to apply our technical knowledge to solve real-world problems, and this project offered the perfect intersection of embedded systems, sensor integration, power management, and user interaction design. It allows us to practice and demonstrate skills in microcontroller programming, analog and digital signal processing, and system-level integration—skills we've developed throughout our coursework and labs. The Smart Garden System not only fits well within our academic field, but also gives us the freedom to be creative and work on something that's meaningful to us personally.

Each of us has, at some point, struggled to keep a plant healthy, whether due to a busy schedule, lack of knowledge, or inconsistent conditions in our homes. This personal connection to the problem strengthens our motivation, as we want to build something that we would use ourselves. There's something deeply rewarding about nurturing a plant and watching it grow, and this project gives us the opportunity to make that experience more accessible and successful for others.

Beyond our own interests, we recognize the broader value of plant care in households. Plants like the **snake plant** are known to filter airborne toxins such as formaldehyde, benzene, and xylene, improving indoor air quality and contributing to a healthier home. Studies have also shown that indoor plants can help reduce stress and improve mood—benefits that are especially important in today's fast-paced, often digitally overwhelming environments. Maintaining healthy plants isn't just a hobby; it's a lifestyle choice that supports wellness, sustainability, and mental clarity.

The **post-pandemic gardening boom** has also played a role in our decision. Many people began turning to indoor and balcony gardens during lockdowns as a way to stay grounded and productive. Even now, the trend of **natural living and homegrown food** continues to grow, with individuals and families looking for ways to bring a little more green into their homes. Our Smart Garden System aligns perfectly with this shift and could eventually evolve into

something scalable—perhaps a larger, multi-plant system, or even a community garden platform in the future.

Ultimately, this project represents a fusion of our technical training and our appreciation for the natural world. It challenges us to think critically about automation and sustainability, while offering a product that could make a real difference in people's lives. For us, it's more than just a senior design project—it's a chance to build something practical, meaningful, and potentially expandable far beyond its initial scope.

2.3 Existing Projects

The intersection of technology and horticulture has given rise to a variety of smart plant care solutions, ranging from commercial products to DIY projects. These innovations aim to simplify plant maintenance by automating tasks such as watering, lighting, and monitoring environmental conditions.

2.3.1 Commercial Solutions

LetPot Automatic Watering System 2.0: This system automates plant watering by delivering controlled water amounts to each plant's base through adjustable drippers, all managed via a smartphone app. It's suitable for different scenarios, including indoor and outdoor plants, and offers features like customizable watering schedules and real-time monitoring.

GrowCube Smart Watering System: GrowCube is an intelligent plant watering system based on four independent soil moisture sensors and water outlets. Using an exclusive app, it allows users to monitor soil moisture in real-time and control watering accordingly. The system supports up to four plants simultaneously and provides detailed data visualization for plant care.

Click and Grow Smart Garden: Click and Grow offers a range of indoor gardening products, including the Smart Garden series. These systems use self-irrigating technology and "plant pods" containing seeds, nutrients, and a growing medium. The Smart Garden 3 and 9 models are designed for home use, while the Wall Farm can grow up to 51 plants, catering to more extensive indoor gardening needs.

Gardyn Home System: Gardyn provides a high-tech indoor gardening solution that allows users to grow up to 30 plants, including vegetables, fruits, and herbs, indoors with minimal effort. The system features full-spectrum LED lighting, AI-based plant care monitoring via the Gardyn app, and a sleek design suitable for modern living spaces.

2.3.2 DIY and Open-Source Projects

Arduino-Based Smart Gardens: Enthusiasts have developed various DIY smart garden projects using Arduino microcontrollers. These projects often include soil moisture sensors, temperature sensors, and light sensors to monitor plant conditions. For instance, one project integrates an Arduino Mega with a TFT touchscreen to display real-time data and control watering schedules.

Raspberry Pi IoT Garden: Leveraging the power of the Internet of Things (IoT), some DIY projects utilize Raspberry Pi to create smart gardens. These systems can monitor real-time status of plants, including soil moisture and temperature, and allow users to access data and control watering through a smartphone app.

OpenAg Personal Food Computer: Developed as an open-source platform, the Personal Food Computer is designed for controlled-environment agriculture. It allows users to monitor and control environmental variables such as temperature, humidity, and light, providing a comprehensive system for plant growth research and education.

2.3.3 Comparative Analysis

While commercial systems like LetPot and GrowCube offer user-friendly interfaces and plug-and-play functionality, they may come with limitations in terms of customization and scalability. On the other hand, DIY projects provide flexibility and a deeper understanding of the underlying technology but require a higher level of technical expertise and time investment.

Our proposed Smart Garden System aims to bridge the gap between these two approaches by offering a customizable, scalable, and user-friendly solution. By integrating affordable sensors, microcontrollers like the ESP32, and optional features such as solar power and mobile app integration, our system aspires to provide a comprehensive plant care solution suitable for both novice and experienced users.

2.4 Goals and Objectives

Goals:

This project aims to provide an efficient and reliable solution for monitoring and maintaining the health of indoor plants using automation, environmental sensing, and user-centered design. The Smart Garden System is intended to reduce the burden of daily plant care while promoting sustainability and increased user engagement. The platform combines hardware and software subsystems to create a modular, extensible system capable of adapting to a wide variety of plant types and environments. The goals listed below represent the broad, high-level vision for the system, with each goal followed by specific technical objectives that define the key deliverables needed to fulfill it.

🌿 Goal 1: Build an optimized plant environment for health and growth

To ensure plant longevity and promote consistent development, this goal focuses on sensing and maintaining the plant's physical environment primarily hydration and sunlight exposure through intelligent, automated responses.

- Develop a soil moisture sensing system capable of continuously monitoring the plant’s hydration level.
- Implement a light intensity sensing system to measure and log the amount of ambient sunlight the plant receives throughout the day.
- Record real-time environmental data from the soil moisture and light sensors at regular intervals for analysis and feedback.
- Establish a structured data storage system to archive all recorded sensor data for long-term tracking and comparison.
- Include plant care profiles with ideal moisture and light ranges for various species, allowing the system to interpret sensor data accordingly.
- Define automated threshold logic to trigger watering or light supplementation when environmental readings fall outside optimal ranges.
- Design a stable power delivery system with support for future expansion to solar or battery options.
- Calibrate all sensors against known values to ensure accurate readings for reliable feedback control.

📷 Goal 2: Develop a Camera System for Daily Growth Tracking

To visualize and document plant growth over time, the system will include a time-lapse camera module. This feature not only enhances user engagement but also serves as a useful tool for analyzing environmental effects on plant development. Captured images will be stored with timestamps, allowing for playback of the plant’s progress over days or weeks.

- Integrate a compact, low-power camera module capable of capturing periodic images (e.g., once per day).
- Automate the image capture process using the microcontroller, triggered by a daily timer.

- Store images locally with metadata (date/time) for future retrieval and time-lapse generation

- Ensure power-efficient operation of the camera system to support long-term deployment.

- Design system architecture to allow later integration with cloud or mobile app viewing features.

📱 Goal 3: Create a Functional Mobile Application Interface

The mobile application will serve as the primary interface for users to access their plant's environmental data in a clean and readable format. The goal is to provide users with an easy-to-understand interface for tracking their plant's condition over time without requiring complex interaction or real-time processing. This will help support user awareness and engagement while maintaining system simplicity.

- Develop a mobile app interface that displays current moisture and light readings in an organized and user-friendly format.
- Enable the app to store and display past sensor readings over time, allowing users to view environmental changes and identify patterns.

- Configure the app to receive and display data transmitted from the system's microcontroller at fixed time intervals.
- Integrate a basic plant database within the app that allows users to select their plant type and apply pre-programmed optimized care levels.

🔌 Goal 4: Create a Modular and Expandable System Design

This goal focuses on ensuring the project is built with long-term scalability in mind. By designing the hardware and software architecture to be modular, future additions such as more sensors, plant zones, or new features can be integrated with minimal rework. This also allows other developers or users to customize and adapt the system to suit different use cases or plant types beyond the initial prototype.

- Design the system architecture to support multiple sensor inputs and actuator outputs, allowing additional plant zones to be added in the future.
- Organize the hardware layout, including the custom PCB, to allow for additional modules such as temperature sensors, humidity monitors, or alternative lighting.

- Ensure software components, including data logging and mobile interface functions, are structured to accommodate future expansion without major redesign.

- Enable clear documentation and code structure to support future teams in continuing or building upon the project after the initial implementation.

Objectives:

Project Objectives

💧 Water Distribution System Objectives

- Deliver water to the plant using a compact pump or solenoid valve system controlled by the microcontroller
- Respond to moisture sensor data and initiate watering when hydration levels fall below a pre-set threshold
- Regulate water flow duration for each watering cycle to avoid overwatering
- Allow for simple calibration and tuning of moisture thresholds during testing and refinement

☀️ Light Management System Objectives

- Use an LED grow light to provide supplemental lighting in environments with low natural sunlight
- Activate the light module based on light sensor input and defined exposure thresholds
- Support scheduled or triggered lighting routines that simulate a healthy day-night cycle
- Ensure efficient power usage to minimize energy consumption and heat buildup

📱 App and User Interface Objectives

- Develop a mobile application that displays soil moisture and light readings from the system
- Visualize past sensor readings through a basic history log or time stamped record system
- Allow the user to select a plant profile from a preset database with optimized care values
- Receive and display sensor data via Bluetooth communication from the microcontroller
- Create a clean and user-friendly layout tailored for casual plant owners

📷 Camera System Objectives

- Integrate a small camera module to capture daily images of the plant
- Store images locally or forward them to the app for later viewing

- Associate each image with a timestamp to enable time-lapse creation
- Operate with low power consumption between image captures

🌱 Environmental Sensor Objectives

- Accurately measure soil moisture and ambient light conditions using reliable sensors
- Collect and log data at consistent intervals to track environmental trends
- Ensure sensor accuracy through calibration and validation during integration
- Reduce power usage by placing sensors into low-power modes between active readings

🔌 Microcontroller and System Integration Objectives

- Manage all system functions including data collection, actuator control, and communication
- Coordinate timing and logic to run the watering, lighting, and camera subsystems
- Package sensor and system data for Bluetooth transmission to the app
- Interface with a custom PCB for streamlined hardware integration and power routing

//i gotta fix formatting here

Stretch Goals / Future Additions

📳 Notification and Alerts

• Add a mobile alert system to notify users of low moisture, poor lighting, or other plant stress indicators
• Include customizable thresholds so users can set personalized alert conditions

🌡️ Environmental Expansion

• Integrate temperature and humidity sensors to monitor additional environmental factors
• Create alerts when values fall outside the optimal range for a selected plant type

🏡 Enclosed Growth Environment

• Design a mini greenhouse enclosure with passive or active climate control features
• Maintain a stable microclimate for sensitive plants or exotic species

🍓 Edible Plant Support

• Expand system profiles to support fruits, vegetables, and herbs
• Include care parameters for different growth phases like seedling, flowering, and fruiting

🧩 **Modular Scalability**

• Support multi-plant setups with separate sensor-actuator modules per plant
• Enable easy hardware expansion through plug-and-play connectors or addressable nodes

📸 **AI Plant Identification**

• Use the onboard camera and an integrated AI model to recognize plant species
• Auto-load matching care profiles based on plant type for optimal automation

☁️ **Cloud Integration**

• Sync data to the cloud for backup, multi-device access, and remote monitoring
• Enable long-term trend analysis and personalized plant insights via web or mobile dashboard

📈 **Growth Analytics**

• Analyze image and sensor data to estimate plant growth trends and detect early signs of stress
• Provide predictive care suggestions based on historical patterns

// keeping this for now

Chapter 3 - Description of Features/Functionalities

3.1 General

The smart plant care system is designed to automate and optimize the maintenance of houseplants through a user-friendly, scalable platform. Its general functionality revolves around monitoring environmental variables, delivering targeted care, and providing meaningful user feedback. This system is intended to serve both novice and experienced plant owners, offering reliable and informative features to support healthier plant growth.

A core feature of the system is its modular scalability. The initial prototype is built to support a single plant, but it is designed to expand easily to accommodate multi-plant configurations. Each module includes its own sensor suite and actuator components to ensure individual plant needs are met without interference. This allows the system to grow alongside a user's plant collection, creating a flexible long-term solution.

The system also integrates a basic user interface designed for accessibility. The UI provides real-time sensor data, historical trends, and options to configure care parameters such as light thresholds or watering schedules. For enhanced interaction, the project envisions a mobile application interface that may eventually support remote monitoring and manual overrides.

Another general capability of the system is the ability to function semi-independently when unattended. The goal is to ensure consistent plant care even while users are traveling or otherwise unavailable. Timed watering, light triggering, and data logging are all features intended to reduce the amount of manual upkeep required.

Additionally, the project is built with educational and sustainability-oriented goals. Users can learn more about their plant's needs over time and may consult a built-in database for optimal growth conditions. Power options may include standard DC supply or, in future revisions, solar panel integration for low-energy off-grid setups. Overall, the general system behavior emphasizes reliability, user empowerment, and extensibility.

3.2 Hardware

The hardware implementation is structured around three primary subsystems: environmental sensing, plant care delivery, and growth monitoring. Each subsystem is composed of key components selected to achieve precise functionality and reliable integration.

Environmental Sensing Subsystem: This includes the soil moisture sensor and the light sensor. The soil moisture sensor is embedded in the plant's soil and is responsible for collecting data on current hydration levels, which are then transmitted to the microcontroller. The light sensor, positioned to detect ambient conditions, captures sunlight exposure levels. Together, these components allow the system to continuously monitor plant care variables and determine whether intervention is needed.

Plant Care Delivery Subsystem: This subsystem handles active plant maintenance tasks. It includes a water pump or solenoid valve system that is triggered based on sensor thresholds or time logic. A grow light module may also be included for indoor environments with insufficient natural light. These elements are connected to a custom PCB, which organizes power distribution and circuit integration. The microcontroller manages all actuator commands, ensuring that water and light are delivered in the correct amount and at the right times.

Growth Monitoring Subsystem: To add a visual element to plant progress tracking, the system integrates a compact camera module. This module captures daily images that are logged with a timestamp for future time-lapse generation. The camera's operation is controlled by the central microcontroller and configured to function at low power to accommodate continuous deployment.

Additional Considerations: The system includes a power management unit that supports either wall outlet DC input or, in extended applications, solar panel integration. All sensors,

actuators, and communication modules connect through the custom-designed PCB, which also supports power regulation and expansion ports for future scalability.

While the database and user interface reside on the software side, the microcontroller and PCB enable the bridge between hardware readings and digital logging. This ensures sensor data is routed to storage and the user interface seamlessly, maintaining synchronization between physical measurements and software feedback.

3.3 Software

The software architecture of the smart plant care system is responsible for processing sensor data, executing logic for automated actions, managing user interaction, and maintaining data logs. It is divided into several core modules corresponding to the system's primary functions: sensing and data processing, care logic and actuation, image logging, and user interface management.

Environmental Data Processing:
Sensor readings from the soil moisture probe and light sensor are sampled at predefined intervals by the microcontroller's firmware. The raw analog or digital values are filtered and normalized for consistency. Threshold-based comparisons are then applied to determine if the soil is too dry or if the light level is inadequate. These results are stored locally in memory and periodically written to a database for historical tracking.

Automated Care Logic:
Based on the processed data, the software determines whether to activate the water pump and/or grow light. Watering logic can operate on two modes: threshold-based (triggered when soil is dry) or time-based (watering at set intervals). Similarly, the light module can be toggled if natural light falls below a configured lux threshold. Logic routines also include cooldown timers to avoid repeated or excessive actions. These automated decisions are coordinated by a state machine programmed on the microcontroller.

Image Capture and Logging:
The time-lapse feature is managed by a scheduled software task that triggers the camera once daily. Each image is stamped with a date and time, then saved to a local or cloud-connected storage directory. This enables long-term visual analysis of plant growth, which may be displayed in the user interface or exported externally. Storage management includes naming conventions and retention policies to avoid overflow.

User Interface and Database Integration:
All sensor readings, watering events, light activation logs, and image data are synchronized with a centralized database. This allows the mobile application to retrieve and display real-time and historical plant data. The UI provides users with tools to configure watering and lighting thresholds, view graphs of moisture/sunlight trends, browse time-lapse photos, and manually

override system actions. Notifications and alerts are also generated here, informing users of critical plant conditions (e.g., low moisture or excessive heat).

Communication and Expandability:
The software includes serial, Bluetooth, or Wi-Fi communication protocols, depending on final platform selection. These channels support communication between the microcontroller and mobile application, as well as future modules such as multi-plant expansions. The codebase is modular to support scalability and eventual updates such as additional plant profiles or sensor types.

//need to redo this

4.0 Engineering Specifications:

To ensure the Smart Garden System meets both performance and usability goals, a clear set of engineering specifications has been defined. These specifications guide component selection, system behavior, and testing benchmarks. The criteria are based on functionality, safety, user experience, and cost constraints, ensuring the final product is both technically sound and accessible to casual plant owners.

Table 1: Parameters

Parameter	Criteria
Max Power Consumption	≤ 10 Watts
Soil Moisture Sensor Accuracy	±5% moisture content
Soil Moisture Sensor Response Time	≤ 2 seconds
Light Sensor Accuracy	±10% lux value
Light Sensor Range	0 – 100,000 lux
Automated Watering Response Time	≤ 5 seconds after sensor trigger
Water Reservoir Capacity	0.5 – 1 Liter
Water Dispensing Accuracy	±20 mL
Grow Light Control Logic	Activate < 1000 lux; deactivate > 2000 lux
Grow Light Power Consumption	≤ 5 Watts
System Operating Temperature	50 – 100 °F (10 – 38 °C)
Multi-Plant Support	Minimum of 2 independent soil sensor zones
Camera Resolution (Stretch Goal)	1080p for time-lapse or plant analysis
Firmware Update Capability	OTA (Wi-Fi)
System Cost	< $100 per unit

For Demonstration:

The Smart Garden System prototype will be demonstrated in a controlled environment to validate the following core specifications and requirements:

- Real-time Monitoring (Specification: Sensor Accuracy & Response Time):
  Live display of soil moisture, light level, and temperature readings on the LCD or mobile interface. Demonstrates adherence to:

    1. Soil Moisture Accuracy: ±5%
    2. Light Sensor Accuracy: <2 NTU (Turbidity proxy)
    3. Sensor Response Time: ≤ 2 minutes

- Automated Watering (Specification: Dispenser Accuracy & Control Logic):
  The system will autonomously activate the water pump when soil moisture drops below a calibrated threshold, validating:

    1. Water Dispenser Accuracy: 2–4 oz
    2. Response Time of Moisture Trigger: < 1 s
    3. Integration with sensor input and control logic

- User Override (Specification: Manual Input & Interface):
  Watering will also be manually triggered via:

    1. Physical button press on the unit

- Data Logging (Specification: Logging Interval & Data Accessibility):
  Sensor data (moisture, light, temperature) will be logged and viewable over a rolling 1-minute interval, demonstrating:

    1. Logging Resolution: 1 data point per 5 seconds
    2. Local or remote access via interface

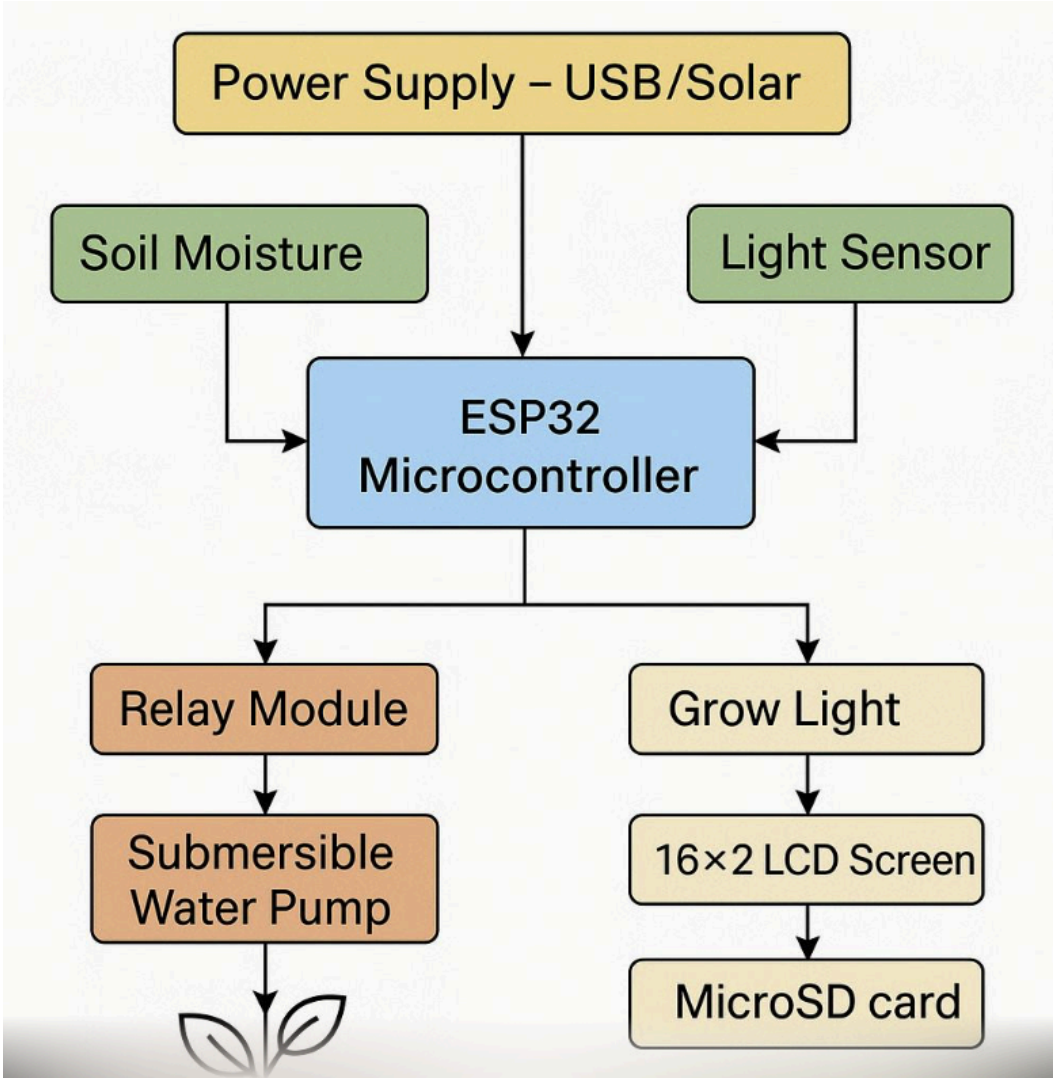
These demonstrations will validate the integration of hardware, firmware, and user interaction in a real-world context while showcasing the system's potential for expansion and scalability.

5.0 Hardware diagram/flowchart

The hardware design of the Smart Greenhouse System centers around the ESP32 microcontroller as the core processing unit. This microcontroller interfaces with a soil moisture sensor and a light sensor to gather real-time environmental data. The system also includes an automated watering mechanism, which consists of a small submersible pump controlled by a relay. Power is supplied via a USB adapter, with the option for solar panel integration. A custom-designed PCB will house these components, ensuring a compact and reliable setup. The hardware flowchart will visually represent the flow of data from sensors to the

microcontroller, the activation of the watering system, and the power distribution throughout the system.

//chan wants to know .. WHY DIDNT YOU KEEP THE PLANT PIC



Yellow - Whole team

Green - Thomas Parker

Blue -Victor Cuenca

Orange - Roman Luknicki

Baige - Rohit Goeptar

Figure 1: Hardware Diagram

5.1 Software diagram/flowchart

The software for the Smart Greenhouse System will be developed for the ESP32 microcontroller. It will include modules for sensor data acquisition, data processing, and control logic for the watering system. The software will also incorporate data logging functionality, storing moisture and light readings for trend analysis. For user interaction, either a local LCD display or a potential future mobile app will be integrated, allowing for real-time data viewing and manual overrides. The software flowchart will illustrate the program flow, including sensor reading loops, decision-making processes for automated watering, data storage routines, and communication with the user interface

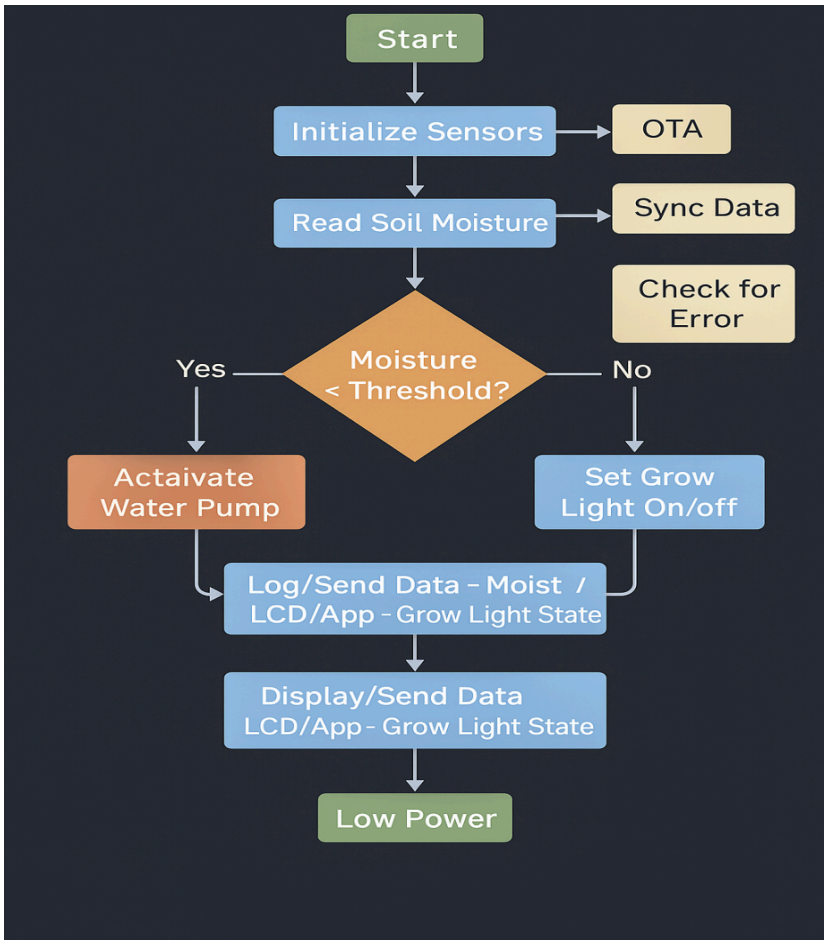


Figure 2: Software Diagram

5.3 House of Quality

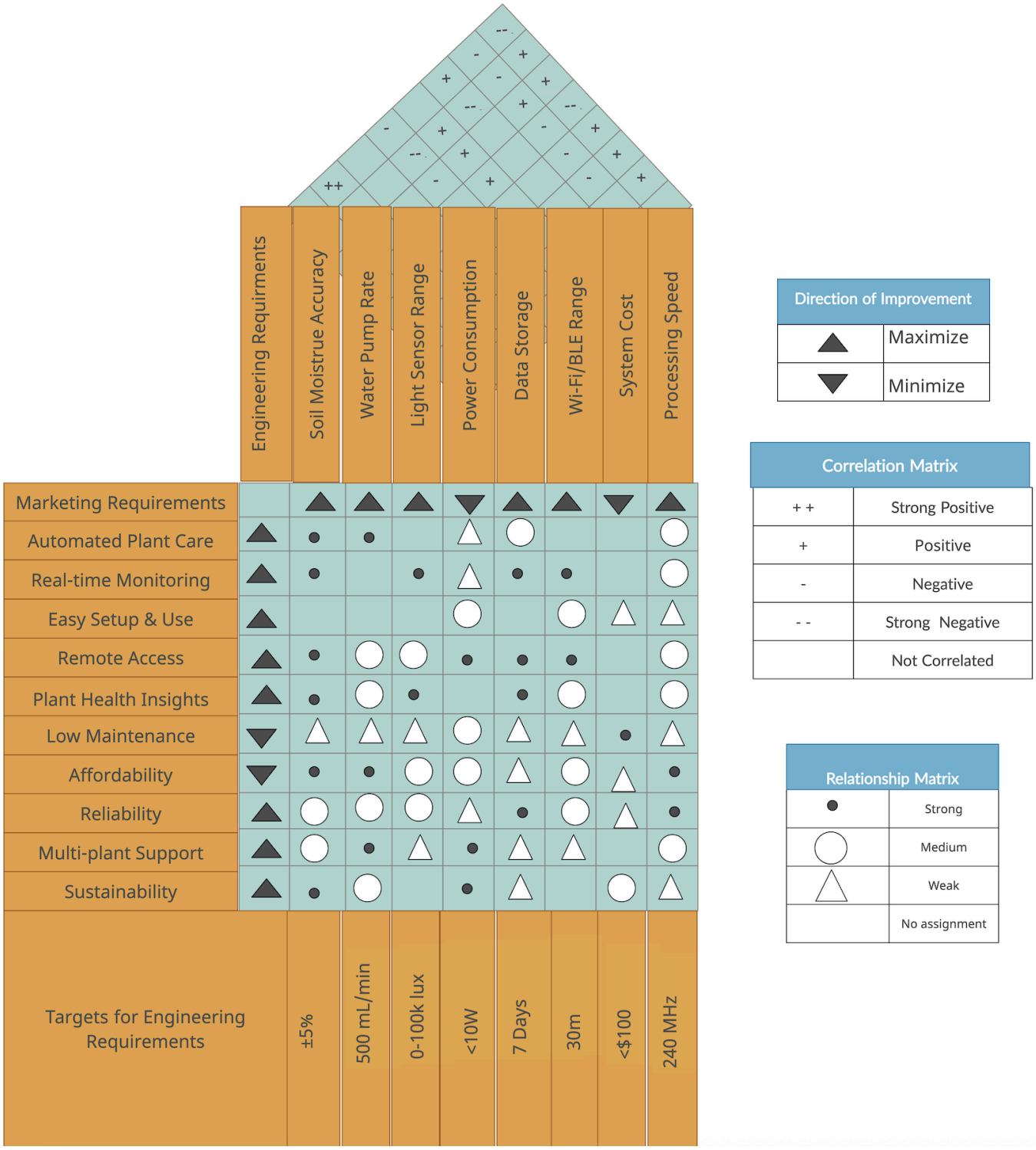


Figure 3: House of Quality

6.0 Budget and Financing

Below we will list the items and components we will need during this project but are subject to change as we personally have never built something like this before. Below will be the main components and we will add a plus and minus buffer range in case we need to add or remove any of the components.

Table 2: Component list

Component	Purpose	Qty	Est. Cost	Notes
ESP32 Dev Board	Main microcontroller for automation and sensor control	1	$6–10	Handles watering logic, BLE/Wi-Fi
Raspberry Pi 4 (2GB+)	Secondary controller for UI and data logging	1	$35–55	Interfaces with touchscreen and hosts web dashboard
Raspberry Pi Official 7" Touchscreen	Touch-based UI for control/settings	1	$60–70	Used with Raspberry Pi for local interface
Soil Moisture Sensor (Capacitive)	Detect soil dryness for each plant	2	$2–4 each	Capacitive type is durable and accurate
Peristaltic Water Pump	Pump to deliver water to plants	2	$8–15 each	Silent and precise water control
4-Channel Relay Module	Switching control for pumps/lights	1	$6–8	Controlled by ESP32
Real Time Clock (DS3231)	Keeps accurate time for schedules	1	$2–4	Enables timed watering
Light Sensor (BH1750)	Measure ambient light levels	1	$3–5	Optional stretch goal
DHT11/DHT22 Sensor	Temperature and humidity measurement	1	$3–6	Optional for environment tracking
Breadboard + Jumper Wires	Prototyping connections	1 set	$5–7	Reusable and necessary for testing
5V Power Supply	Powers ESP32 and sensors	1	$5–10	Wall adapter or battery pack
Power Supply for Pi (USB-C, 5V 3A)	Powers Raspberry Pi and screen	1	$8–12	Official Pi power supply recommended
MicroSD Card (32GB+)	Storage for Raspberry Pi	1	$6–10	Stores OS and logs
Custom PCB (optional)	Cleaner final version of circuit	1	$5–20	For final version
Waterproof Enclosure	Protect electronics from moisture	1	$5–15	Can be 3D printed or prebuilt
Grow Light Panel	Optional light source for indoor plants	1	$10–25	Stretch goal feature

These are average prices so on the low end, cheaper shipping we are looking at $179, on the higher end we are looking at $295. We are also giving ourselves a buffer range of plus or minus $200 in case of any mishaps, price adjustments, addition of components or removal of components.

7.0 Project Milestones

The timeline below is an estimation of how things will go for our group. They are subjected to change in the case of certain due dates that are assigned during this semester and the fall semester for Senior Design 2. Below is a pseudo schedule that we hope to follow in order to progress smoothly through this project.

Table 3: Project Milestones table

Semester	Week(s)	Milestone
Summer	Weeks 1–2	Define project scope, goals, and deliverables; assign team roles
Summer	Weeks 3–4	Conduct research on plant needs, soil sensors, microcontrollers, and touchscreen options
Summer	Week 5	Compare component options and finalize bill of materials
Summer	Week 6	Order essential components and test soil sensor readouts using ESP32
Summer	Week 7	Create block diagrams and software architecture plans
Summer	Week 8	Write pseudocode for core features and interface design layout
Summer	Week 9	Draft early documentation and prepare progress presentation
Summer	Finals Week	Submit Summer Research Summary + Preliminary System Design
Fall	Weeks 1–2	Begin full system integration: sensors, pumps, UI, and ESP32–Pi communication
Fall	Week 3	Add real-time scheduling (RTC module or Pi time)
Fall	Week 4	Implement sensor-driven watering logic with live testing
Fall	Week 5	Build touchscreen interface with manual override features
Fall	Week 6	Add grow light control (optional stretch goal)
Fall	Week 7	Assemble full prototype in waterproof housing
Fall	Week 8	Run multi-day system validation tests
Fall	Week 9	Optimize logic, calibration, and UI feedback
Fall	Week 10	Prepare Final Report, Poster, and User Guide
Fall	Week 11	Polish final demo and presentation
Fall	Finals Week	Deliver Final Presentation + Submit Completed System