

CECS 490A Spring 2025

Automated Planter System

By Ivan Martinez, Andy Madjedi, James Sanchez

May 7, 2025

**Biography**  
  
Ivan Martinez  
  
 I am currently pursuing a Bachelor of Science in Computer Engineering at California State University, Long Beach, where I am passionate about exploring new technologies and programming languages. With a strong interest in embedded systems, microcontrollers, and software development, I have worked on a range of engineering projects, including an FPGA Minesweeper game, an automated analog line-following robot, and a smart house with a stepper motor car. My goal is to become a Systems Engineering Officer in the United States Space Force or Air Force, where I can contribute to advanced aerospace and space innovations. Outside of academics, I enjoy competitive gaming, having served as the captain of my college’s Valorant eSports team, and I also lead engineering projects, demonstrating my leadership skills. I am committed to continuous learning, problem-solving, and contributing to both technical and non-technical teams.



Andy Madjedi

I’m pursuing a Bachelor’s degree in Computer Engineering at California State University Long Beach. Being the son of a mother who was a computer scientist at Hewlett Packard and Cal Poly San Luis Obispo, and a father who worked as a commercial architect, I've always had a passion for both technology and creativity. I've been employed at the tech repair company iFixit for six years, working in component testing and quality assurance, operations systems, and web design. The varied nature of these positions has given me a wide breadth of knowledge and experience along the entire spectrum from hardware to software. Because of iFixit's focus on the Right to Repair movement, I believe my work there has instilled in me a sense of ethical responsibility for how I approach my career path. Ideally, my career goals involve working in the aerospace, agricultural, or medical fields as a Computer Engineer, using the skills I’ve learned at CSULB to better the world around me.



James Sanchez

I am pursuing a Bachelor’s Degree in Computer Engineering at The California State University, Long Beach where I am set to graduate in the Fall semester of 2025. I am passionate about anything related to technology, how they work, and its effects on society as a whole. I enjoy the fields of Hardware Design, Software Development, and Embedded Systems. Some examples in which I exemplify my knowledge in these areas include previously collaborating with the CSULB team in designing a drone that is participating in the Raytheon competition in 2025 as well as developing an asset tracking system for my job using Python(with Flask), SQLite, and HTML/CSS for a web interface. My goal after obtaining my Bachelor’s degree is to work for a company in which innovation and aspiring to better society is the main priority. Although my academic journey is coming to an end, I aspire to continuously learn and apply my knowledge of technology and engineering as well as my leadership and team-collaborating qualities in my future endeavors.



**Project Overview**  
  
 Our Automated Planter system is designed to make plant care easier and more efficient for anyone, from beginners to experienced gardeners. The system automatically takes care of watering your plants, ensuring they receive the right amount of water without the risk of overwatering or under-watering. The planter is equipped with sensors that measure the moisture level in the soil and the amount of light the plant is receiving, allowing it to adjust watering schedules and notify the user when the planter's water tank is at a certain level. The system is also connected to a simple web interface that lets users monitor their plants from their computer or phone, providing data on the plant's health and environmental conditions.

The end-user of this product is anyone who enjoys having plants, but may struggle with remembering to water them regularly or worry about over-watering. This could be ideal for busy individuals, plant enthusiasts, or even those who are new to gardening. It’s a solution for anyone looking to take the stress out of plant care, offering both convenience and peace of mind. The automated planter is also portable, allowing users to place it anywhere in their home, and it’s designed to be energy-efficient and environmentally friendly, making it a sustainable option for urban gardening.

It is important to note that the motive of the Automated Planter is to not take the place of the human, but instead to assist. Like previously stated, sometimes people are too busy or distracted and forget to tend to their plant. Other cases may include plant enthusiasts who know how to take care of plants, but just need a little help especially when the plant is new and they want to ensure it starts off strong prior to transferring the plant outside or in a bigger area. This product should not only assist, but teach the user the various aspects that need to be considered when taking care of a plant.   
  
**Customer Needs**  
 Minimum Needs (Standard Features that are Necessary):  
The system will use a Raspberry Pi 5 (8GB) as the central processing unit, interfacing with all the sensors and servers to act as the brain of the system. The planter will be powered by a 15W power supply, with an input range of 94-264V AC from a wall connection to a USB-C cable connected to the Raspberry Pi. This ensures the system is reliable and energy-efficient. The planter needs to include a water tank, whose size is yet to be finalized, providing sufficient capacity to minimize the need for frequent refills. The system will feature a water pump connected to the Raspberry Pi that will move water from the tank to the plant at a rate of 100 liters per hour (L/H) when activated. Essential sensors for humidity, soil moisture, and light will be incorporated to ensure proper plant care. These sensors will be waterproof (IPX-5 rated) to prevent malfunctions due to exposure to water.

Assumed Needs (Basic Operation, Safety, Reliability):  
The planter must be easy to move, with a design that is not too bulky but still durable enough to withstand accidental falls without breaking into pieces. The system will need to monitor and display the current water tank capacity. This will allow users to keep track of water levels easily without constantly checking the tank. A web-based app hosted on the Raspberry Pi will be accessible from any computer or smartphone, allowing users to connect wirelessly to view real-time data from the planter, including sensor readings, plant care history, and water tank capacity. The system will log sensor data into a local database on the Raspberry Pi at regular intervals. Additionally, SMS alerts will be sent to the user when the water tank reaches low capacity thresholds, such as 50%, 30%, 10%, to ensure the user is notified in time before the tank is completely empty.

Unrecognized Needs (Features the Customer Didn’t Know They Needed):  
In addition to SMS alerts, a display screen on the planter will show the water tank’s capacity status (high, medium, low, extremely low). This provides a quick, visual way for users to check water levels in case they miss or forget the SMS notifications. The use of waterproof and water-resistant sensors ensures that the system is built to last. These sensors are designed to withstand exposure to water and will continue to function correctly over time, providing customers with a reliable and low-maintenance system.

By addressing these needs, the Automated Planter will offer users a practical, reliable, and efficient solution for plant care. The combination of automation, sensor-based monitoring, and user-friendly data access ensures that customers can easily manage their plants while enjoying the convenience of technology.

**Design Specifications**

There are numerous aspects that can be tested on the final product to ensure customer satisfaction.

1. The watering system not only works, but is efficient and reduces waste. This includes not having to be constantly filled by the user. Once the user fills the reservoir with water, that should last an extended amount of time so that it actually becomes convenient for the user.
2. All sensors connected to the Raspberry Pi are functioning and track the plant health. This allows for the Automated Planter to work as intended by constantly updating throughout the day so that the user does not need to do this themselves.
3. The planter itself is durable. Since it is going to be 3D printed, the filament must not be easily destroyed or easily get defects. Environmental aspects such as water and soil should not impact the planter.
4. The portability of the planter itself. The components will not cause the planter to not be as portable as expected. One thing to consider currently is the fact that it has to be plugged into a wall outlet. This eliminates some portability and is something that the team is analyzing in order to come up with different solutions.
5. Data is easily visible to the user upon looking on the built-in display or web browser. This also includes receiving notifications in a timely manner so that proper action can be taken by the user in case water is needed, soil needs replacing, and other possible events.
6. All electronic components are operating at a safe power level ensuring a seamless experience for the customer. This adds to the care-free aspect of the project allowing the automated planter to continue taking care of the plant without having to replace components.

**Constraints**

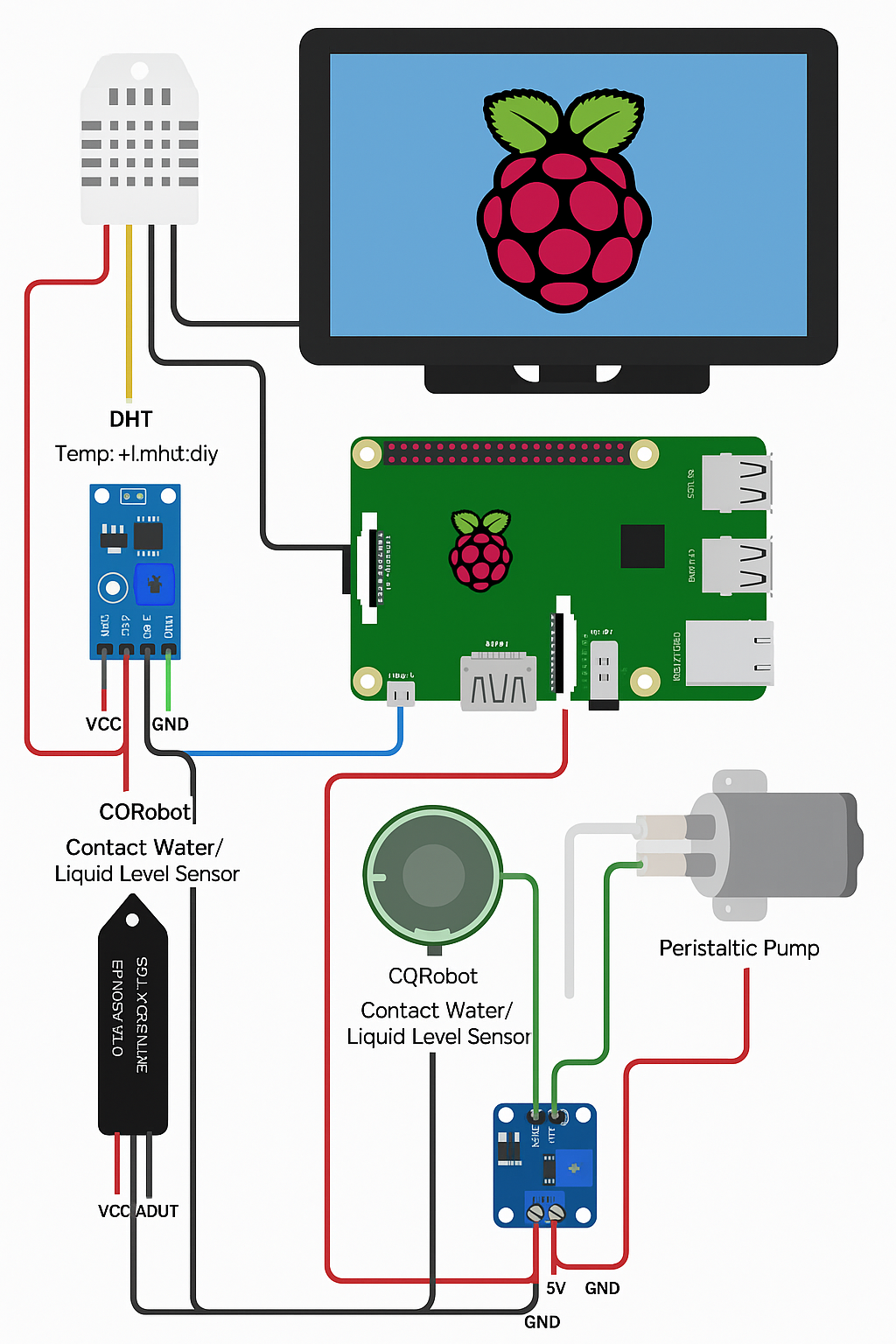
One of the biggest constraints we currently have are the power limitations. Since the Raspberry Pi is acting as the “brain” of this project, this means there are going to be many components connected to the Raspberry Pi. We must ensure that components, such as the sensors that are detecting factors like the humidity or light, are operating within safe power and thermal limits. The GPIO pins on the Raspberry Pi 5 operate at 3.3 V logic levels. Applying more than 3.3 V can permanently damage the Raspberry Pi causing the automated planter to fail. The max current per GPIO pin is around 16 mA and across all pins is approximately 50 ma. All things considered, we need to strategically pick components that can safely operate under these conditions.

Environmental exposure on our components is another constraint we are working with. Since this planter involves water, soil, and plants, this leaves electronic components at risk of malfunctioning and potentially damaging key components that run this planter. This means we must take extra precautions when considering what type of filament we are using for the 3D printing, applying print coating, and applying sealant to any areas in which we have small crevices or holes for components. Other things to consider is the exact placement of the Raspberry Pi so that it is not in a position where it can be compromised to elements such as water.

A constraint imposed by the Raspberry Pi itself is that there is no built-in analog input. This came as a surprise to the team and was something we did not consider prior to ordering some of our parts. For this reason, we are considering the use of a Pi Hat which enhances the functionality of the Pi beyond the basic hardware connections allowing us to utilize some sensors we ordered that involve analog inputs. In case this does not function in the way we hope, we are analyzing other options such as an Analog Digital Converter(ADC) or perhaps going with alternative sensors that do not involve analog inputs.

**Project Implementation**

Our project aims to develop a self\_stsuatining, microcontroller-based smart planter capable of automating the core plant care functions such as watering, environmental monitoring, and data logging. We intend to address the common aches of plant care for both new plant owners and experienced by reducing a lot of the manual intervention needed. This prevents over and under-watering as well as providing environmental feedback through a graphical interface making it easier for the customer to understand.



The core controller in this project is the Raspberry Pi 5 with 8 GB of RAM. This Pi serves as the central processing unit of the system which is exemplified by the diagram above. It executes the control algorithms, manages GPIO-level communication with all the peripherals connected, and hosts a lightweight web server. Moreover, the Raspberry Pi also drives the touchscreen GUI that is crucial in displaying valuable information to the user.

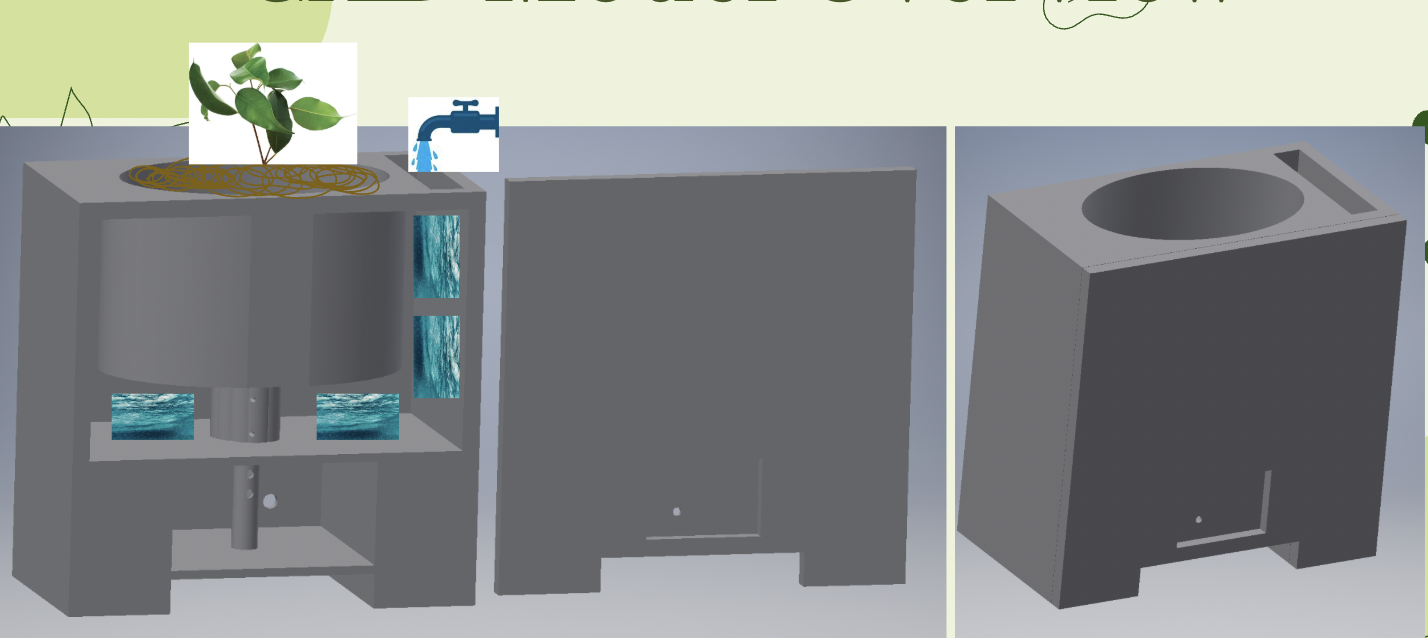
There are multiple sensors we currently have integrated to the planter. The first is the DHT22 which accounts for the temperature and humidity. This is a digital sensor interfaced via GPIO. Our light sensor is a digital light sensor that monitors the ambient light using I2C communication. Our soil moisture sensor outputs analog voltage representing soil moisture, but an ADC is needed in order to communicate with the Pi. Currently, the last sensor we have is the CQRobot contact water level that will be placed in our reservoir. These sensors, three in total, will be placed at the top, middle, and bottom to determine water level via GPIO input. The DIP switch will be configured for 3.3V output to operate a safe level for the Pi. All of these sensor readings will be taken at defined intervals and processed in order to make decisions on irrigation and to populate the system’s data logs.

A crucial aspect to the project is the watering subsystem. The irrigation system will be powered by a 5V peristaltic pump, but will not be directly driven by the Pi due to high-current and the Electromotive Force that could occur causing voltage spikes that may put the Pi at risk. This pump will be activated conditionally based on soil moisture thresholds that will be defined in software. The software for the UI rendering will more than likely be written in Python while all the sensor values will be logged in a local SQLite database.

An important aspect for the aesthetic of the planter is the user interface and visualization from the touchscreen display. This 7’’ IPS display will be connected to the Pi via HDMI and USB. This will display real-time sensor readings, the water level status, graphs of the data that has been logged, and the system status that displays if watering is occurring, the reservoir level, and potential decorative screens for personal customization.

One thing that we have yet to work on, but absolutely plan to do is PCB integration. A custom PCB interface board to connect all components to the Raspberry Pi will lead to more organization and overall a better design inside the planter. This will include breakout headers for all sensors, house components such as an ADC, and also ensures safe voltage handling and organized power routing. We have yet decided if the PCB interfaces with the Pi via GPIO ribbon cable or directly through the 40-pin header.

Some critical safety and reliability aspects are to be considered in the final product. Enclosures for components such as the Raspberry Pi need proper ventilation to prevent overheating. The system must operate with minimal energy and water waste in order to target sustainability. We intend to encourage users to engage in low-maintenance urban gardening this way in order to actually have an environmental impact.

**Diagrams  
  
  
  
Power Management**

| **Component** | **Voltage** | **Current Estimation** | **Power (W)** | **Notes** |
| --- | --- | --- | --- | --- |
| Raspberry Pi 5 (\*GB) | 5V | 1.5-2.5 A | 7.5-12.5 | Base system + GUI processing |
| 7" Touchscreen | 5V | 0.9-1 A | 4.5-5 | HDMI + USB powered |
| Peristaltic Pump | 5V-12V | 0.4-0.6 A | 2-7 | Possibly powered separately |
| DHT22 Sensor | 3.3-5V | 2.5 mA | 0.01 | Low draw, digital |
| Light Sensor | 3.3V | 1-2 mA | 0.01 | Digital / I2C |
| Soil Moisture | 3.3V | 10 mA | 0.03 | Analog / ADC |
| 3 Water Level Sensors | 5V | 12 mA Each | 0.18 Total | Digital outputs |

This is the initial power distribution we have based on the parts we initially ordered. These are estimates and are definitely going to change based on potential additions or subtractions of sensors and other components. It is important to note that the current power supply is the Official Raspberry Pi 27W USB-C Power Supply. The input is 100-240V Ac (wall outlet) and the output is 5.1V DC at up to 5 A via USB-C. This will power the Raspberry Pi 5 as well as all sensors and control circuits via the GPIO power rails on the Pi. As mentioned earlier, this is subject to change and this is just the initial power management we have for the planter. For instance, a possible alternative to wall power is a battery to increase portability which is an important aspect of the planter.

**WBS and Gantt Chart**

The WBS and Gantt Chart is yet to be determined. In the next section explaining the 5 Demos Contract, we have a general idea of the demos and milestones we want to hit. Once defining when the Demo 1 is due, we shall have a clearer idea of the week by week breakdown and what we are focusing on. As for roles, Andy Madjedi is more or less in charge of software development. Ivan Martinez has been focusing on the design of the planter itself and the hardware. James Sanchez has also been focusing on the components and overall hardware design. These roles are not yet clearly defined though as we all have been contributing to each aspect of the project. By the time fall semester begins, we will have a more clearly defined schedule and we will create the WBS and Gantt Chart in a more formal manner.

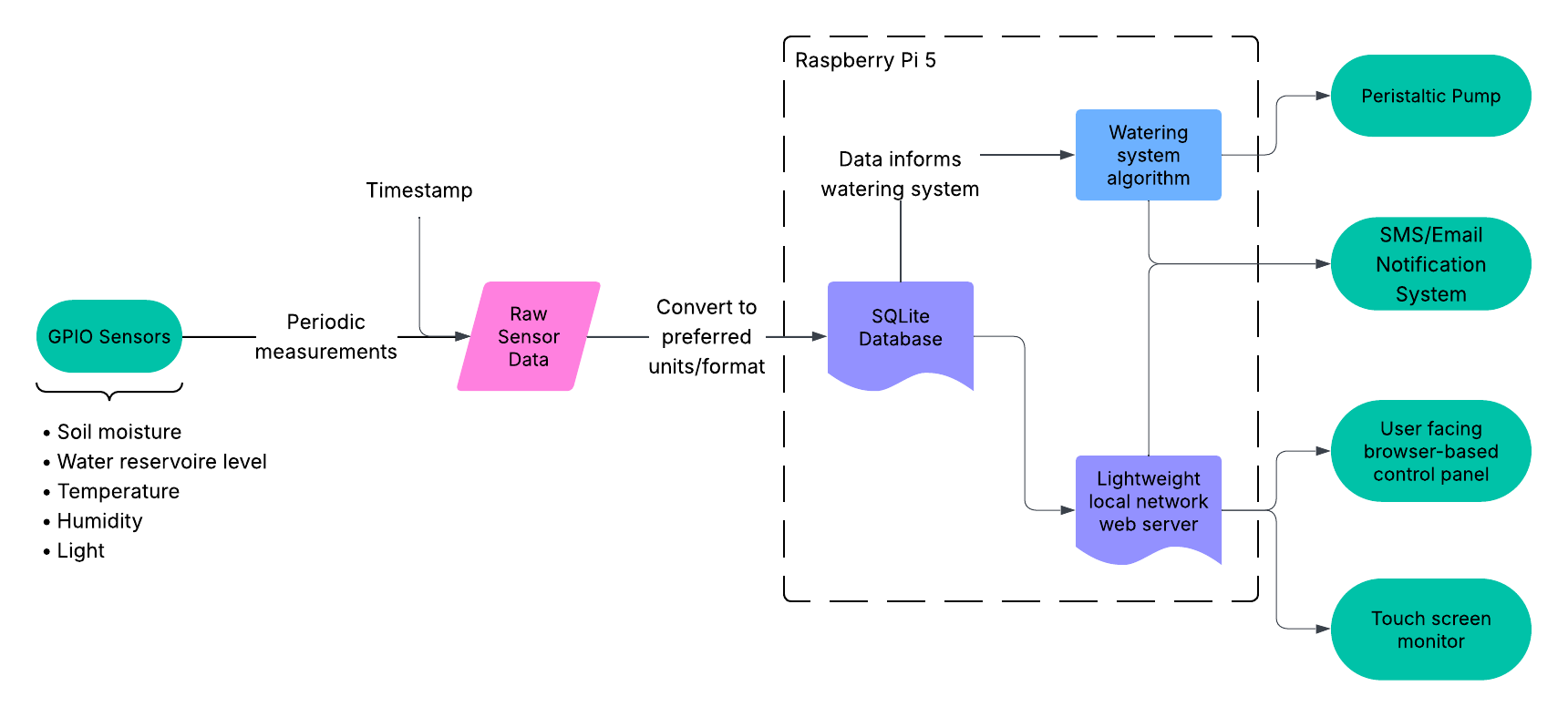
**5 Demos Contract (“Agile Sprints”)**

This is a rough draft of the demos we will have next semester. Demo 1 will involve sensor integration and data acquisition. The objective for this demo is to demonstrate working sensor connections and real-time data acquisition on the Raspberry Pi. In order to be successful, we would want to see correct detection that include stable , repeatable values. Demo 2 will focus on the pump control. We would like to demonstrate control of the peristaltic pump via the Raspberry Pi. The pump should activate when manually triggered by the Pi. Success in this demo will be determined by correct activation at appropriate times as well as no unintended actuation.Demo 3 will focus on the GUI dashboard on the touchscreen display. Here the objective will be to create and display a local dashboard using the touchscreen that visualizes sensor readings and system status. The criteria for success in this demo is to have a touchscreen that responds to input, refreshes data at set intervals, and has an acceptable delay for the updates. Demo 4 will involve remote monitoring of the data being logged. The objective in demo 4 is to ensure the user will be able to see the data being logged from the sensor on a web browser as well as receiving updates via messages and email. Success in this demo will involve seeing the data being updated in real time with an acceptable lag between device and remote update. For demo 5, this will involve having the fully integrated system in the actual planter. This means all components are connected, running, and the automated planter is ready to use. Success in this demo involves the planter to be functioning with no intervention needed. The design is clean and all hardware components are neatly designed and placed inside the planter. All data being collected is stored and the user is able to see it visually and obtain updates via messages and email. These demos are subject to change, but for now the team feels confident in these deadlines and sees them as realistic targets to strive for.

**Societal, Environmental Impact, and Sustainability**  
 Our project has the potential to positively impact society and the environment by promoting efficient use of resources, reducing energy consumption, and encouraging sustainable technology practices. By designing the system with low-power components and optimizing energy usage, we ensure that the device operates efficiently, conserving battery life and minimizing electrical waste. This focus on energy efficiency is especially important in a world where reducing carbon footprints is a priority. Furthermore, the use of durable, recyclable materials in the construction of our project helps minimize electronic waste, supporting a more sustainable product lifecycle.  
  
 This product is designed with sustainability in mind, ensuring efficient use of materials, energy, and resources. By incorporating low-power components and optimizing energy consumption, it minimizes its environmental footprint while maintaining reliable functionality. The device is constructed using durable, recyclable materials, which helps reduce electronic waste and extends the product's lifespan. In addition to being environmentally conscious, the product is built with user safety and reliability as top priorities, ensuring that it provides consistent performance without unnecessary resource consumption. Ultimately, this product serves as a responsible and efficient solution, balancing functionality and sustainability without compromising on performance.  
  
**Ethical Considerations** In designing this project, we have carefully considered several ethical principles to ensure that the product is both responsible and beneficial. Safety has been prioritized throughout the design process. Our system is built to minimize risks to users and the environment by using safe voltage levels, properly insulated components, and secure connections. We have also ensured that any interactive elements, such as buttons or interfaces, are intuitive and reduce the risk of user error.

User autonomy is respected by giving users full control over how they interact with the system. The product is designed to be user-friendly, allowing individuals to make informed choices about how they use it without being restricted by complex or hidden features. This emphasis on user control ensures that the product enhances user experience rather than manipulating or limiting it. Privacy has been maintained by avoiding unnecessary data collection. Our design does not require or store any user data, ensuring that individuals can use the product without concerns about their personal information being accessed, stored, or shared. This approach not only protects user privacy but also aligns with ethical principles of transparency and trust. By adhering to these ethical principles, our project aims to provide a product that is not only functional and efficient but also responsible and respectful of users and the environment.

**Software Flowchart**

****

**Software Verification, Validation, and Revision**

Currently, our plan is to implement all GPIO interaction and decision making algorithms with C. This would include readings from all analog and digital sensors, and control of the peristaltic pump. Fortunately, SQLite offers a C library for easily interfacing with the database.

The web server that serves the browser and touch screen interfaces will likely be implemented with a lightweight and easy to use framework such as Express, which is a small Node.js framework. Node.js also offers a third party library for interfacing with an SQLite database. Any charts and graphs we wish to display in the control panel from sensor data will likely be implemented with third party graphics libraries.

We plan to use Git and GitHub for revisioning and versioning. This will not only help facilitate easy collaboration between all team members, but will also make deployment of our code onto the Raspberry Pi, which will be connected to the internet, much more seamless. It will also allow us to quickly revert unnecessary software changes as hardware specifications or requirements change.

References

Flaura - Smart Plant Pot. (2021, October 3). *Capacitive Soil Moisture Sensors don’t work correctly + Fix for v2.0 v1.2 Arduino ESP32 Raspberry Pi* [Video]. YouTube. https://www.youtube.com/watch?v=IGP38bz-K48

Hansson, L. J., Sten, G., Rossander, M., Lideskog, H., Manner, J., Van Westendorp, R., Li, S., Eriksson, A., Wallner, A., Rönnqvist, M., Flisberg, P., Edlund, B., Möller, B., & Karlberg, M. (2024). Autoplant—Autonomous site preparation and tree planting for a sustainable bioeconomy. *Forests*, *15*(2), 263. https://doi.org/10.3390/f15020263

Instructables. (2021, June 2). *Automated garden system built on Raspberry Pi for outdoors or indoors - MUDPI*. Instructables. https://www.instructables.com/Automated-Garden-System-Built-on-RaspberryPi-for-O/

K, A., & Mahadevaswamy, U. B. (2018). Automatic IoT Based Plant Monitoring and Watering System using Raspberry Pi. *International Journal of Engineering and Manufacturing*, *8*(6), 55–67. https://doi.org/10.5815/ijem.2018.06.05

Kansara, K., Zaveri, V., Shah, S., Delwadkar, S., Jani, K., Babu Madhav Institute of Technology, & Babu Madhav Institute of Information Technology. (2015). Sensor based Automated Irrigation System with IOT: A Technical Review. *Sensor Based Automated Irrigation System With IOT: A Technical Review*.

Nikodem Bartnik. (2022, July 13). *Automatic garden watering system* [Video]. YouTube. https://www.youtube.com/watch?v=s-xkdfNeIVw

Piltch, A. (2023, October 23). *How to set up a headless Raspberry Pi, without ever attaching a monitor*. Tom’s Hardware. https://www.tomshardware.com/reviews/raspberry-pi-headless-setup-how-to,6028.html

Practical Engineering. (2015, April 2). *Arduino Garden Controller - automatic watering and data logging* [Video]. YouTube. https://www.youtube.com/watch?v=O\_Q1WKCtWiA