**3.1 Project Management/Tracking Procedures**

The management style we have chosen for our project is waterfall style project management strategy. This strategy has been chosen because of our project’s dependency on hardware which is costly enough to eat up our project budget making it difficult to reverse on design decisions. This quality necessitated a protracted analysis and planning period and prevented us from returning to previous design steps, making a waterfall based strategy seemingly the best option.

**Project tracking:**

Informal project communication will occur via discord due to its versatility allowing for easy VOIP communication and image sharing allowing for quick communication of ideas. Formal progress software will be stored on the provided team git lab repository, hardware progress will be shared via the team discord.

**3.2 Task Decomposition**

* Task 1: Complete the user interface design and implement in react native

1. Design UI in figma for app and determine software to implement frontend and backend
2. Get software development environment setup for React Native
3. Develop the UI in React Native
4. Test UI
5. Deploy on app store

* Task 2: Set up AWS using SQL test backend and complete a round trip through Pi AWS and app.

1. Create on paper model of database structure
2. Get familiar with AWS and how to deploy a backend on their servers
3. Develop backend for local host
4. Deploy backend on AWS servers
5. Work on connecting mobile app AWS backend and PI
6. Complete round trip and test functionality

* Task 3: Implement the necessary hardware for the device to work as intended. Once functionality has been established, create a PCB that will handle all design and functional requirements

1. Sensor Selection
2. Work on receiving valid sensor data
3. Actuator Selection
4. Work to control water/fertilizer release with actuators
5. Breadboarding circuit to incorporate sensor and actuator power.
6. Ensure Sensor Data can be Read and Formatted in a Form that can be Useful. Initial software to command actuators.
7. Connect backend to Pi to get sensor data and complete round trip

**3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA**

Task 3 milestones:

* Receive accurate understandable data from sensors
* Send/receive parseable RESTfull requests to the server
* Utilize actuators to release water/fertilizer in a controlled manner
* Custom PCB is created
* Actuators are activated remotely by user via app
* Final working prototype circuit is created

Task 2 Milestones:

* Successful local database update
* Database structure complete on paper
* Successful backend deployment on AWS server
* APP is able to update database on AWS server

Task 1 Milestones:

* App mockup is finished
* UI works as intended

**3.4 PROJECT TIMELINE/SCHEDULE**

**First Semester:**

**Target 1: Complete the User Interface Design and Implement it in React Native**

**February:**

**Week 1-4:** Design UI in Figma for app and determine software to implement frontend and backend.

**March:**

**Week 1:** Set up software development environment setup for React Native.

**Week 2-4:** Develop the UI in React Native.

**April:**

**Week 1-4:** Develop the UI in React Native.

**May:**

**Week 1-2:** Develop the UI in React Native.

**Week 3-4:** Test UI.

**Week 3:** Deploy on the app store.

**Target 2: Set up AWS using SQL Test Backend and Complete a Round Trip through Pi AWS and the App**

**March:**

**Week 1:** Get familiar with AWS and how to deploy a backend on their servers.

**Week 2-3:** Develop backend for local host.

**Week 4:** Deploy backend on AWS servers and make break points.

**April:**

**Week 1-3:** Deploy backend on AWS servers and make break points.

**Week 4:** Work on connecting mobile app AWS backend and PI.

**May:**

**Week 1-3:** Work on connecting mobile app AWS backend and PI

**Week 1-2:** Complete round trip and test functionality.

**Target 3: Implement the necessary hardware for the device to work as intended. Once functionality has been established, create a PCB that will handle all design and functional requirements.**

**February:**

**Week 3-4:** Sensor functionality.

**March:**

**Week 1-2:** Sensor functionality.

**Week 2-4:** Actuator Selection and Individual Testing

**Week 4:** Breadboarding circuit to incorporate sensor and actuator power.

**April:**

**Week 1-4:** Breadboarding circuit to incorporate sensor and actuator power.

**Week 1:** Actuator Selection and Individual Testing

**Week 2-4:** Ensure Sensor Data can be Read and Formatted in a Form that can be Useful. Initial software to command actuators.

**Week 4:** Connect the backend to Pi to get sensor data and complete round trip.

**May:**

**Week 1:** Ensure Sensor Data can be Read and Formatted in a Form that can be Useful. Initial software to command actuators.

**Week 1-2:** Connect the backend to Pi to get sensor data and complete round trip.

**August:  
Week 2-4:** PCB Design.

**Week 2-4:** Ensure Sensor Data can be Read and Formatted in a Form that can be Useful. Initial software to command actuators.

**September:**

**Week 1:** Ensure Sensor Data can be Read and Formatted in a Form that can be Useful. Initial software to command actuators. **Week 1-3:** PCB Design.

**Week 4:** Hardware Testing.

**October:**

**Week 1-4:** Hardware Testing.

**November:**

**Week 1:** Hardware Testing.

**Target 4: Additional database and app functionality.**

**August:  
Week 1-4:** Research needs of other plant types.

**September:  
Week 1-2:** Research needs of other plant types.

**Week 2-4:** install and link new sensors.

**Week 2-4:** Update database columns/backend code.

**October:**

**Week 1:** Install and link new sensors.

**Week 1:** Update database columns/backend code.

**3.5 RISKS AND RISK MANAGEMENT/MITIGATION**

**3.6 PERSONNEL EFFORT REQUIREMENTS**

| **Task** | **Subtask** | **Estimated hours** |
| --- | --- | --- |
| Task 1: UI Design and React Native Implementation | Design UI in Figma | 12 |
| Task 1: UI Design and React Native Implementation | Determine software for frontend and backend | 2 |
| Task 1: UI Design and React Native Implementation | Setup React Native development environment | 1 |
| Task 1: UI Design and React Native Implementation | Develop UI in React Native | 30 |
| Task 1: UI Design and React Native Implementation | Test UI | 5 |
| Task 1: UI Design and React Native Implementation | Deploy on app store | 5 |
| Task 2: AWS Backend and Pi Integration | Model database structure on paper | 1 |
| Task 2: AWS Backend and Pi Integration | Learn AWS deployment | 2 |
| Task 2: AWS Backend and Pi Integration | Develop backend for local host | 3 |
| Task 2: AWS Backend and Pi Integration | Deploy backend on AWS servers | 1 |
| Task 2: AWS Backend and Pi Integration | Connect app with AWS backend and Pi | 4 |
| Task 2: AWS Backend and Pi Integration | Complete round trip and test functionality | 2 |
| Task 3: Hardware Implementation and PCB Design | Select sensors | 10 |
| Task 3: Hardware Implementation and PCB Design | Validate sensor data | 5 |
| Task 3: Hardware Implementation and PCB Design | Control water/fertilizer release with actuators | 10 |
| Task 3: Hardware Implementation and PCB Design | Breadboard sensor and actuator circuit |  |
| Task 3: Hardware Implementation and PCB Design | Format sensor data for backend |  |
| Task 3: Hardware Implementation and PCB Design | Connect hardware with backend |  |
| Task 4: Database and App Functionality Extension | Research plant type needs |  |
| Task 4: Database and App Functionality Extension | Install and link new sensors |  |
| Task 4: Database and App Functionality Extension | Update database and backend code |  |
| Task 4: Database and App Functionality Extension | Test extended functionality |  |
|  |  |  |
|  |  |  |

**3.7 OTHER RESOURCE REQUIREMENTS**

**Prototyping Components:**

* **Raspberry Pi Pico:** The central microcontroller for sensor data processing and actuator control.
* **Soil and Plant:** Essential for real-world testing of soil sensors.
* **NPK Sensor:** For measuring soil composition, including nitrogen, phosphorus, potassium, temperature, moisture, and pH.
* **Light Sensor:** To monitor ambient light levels affecting plant growth.
* Humidity and Temperature Sensor: For tracking the air conditions around the plant environment.
* **Salinity Sensor:** To assess the salt content in the soil, which is crucial for plant health.
* **Actuators:** Solenoid valves or similar mechanisms for water and liquid fertilizer dispensing.

**Hardware Assembly and Enclosure:**

* **Enclosure:** A case to house the electronics with modifications for sensor and actuator mounting.
* **Relays and Wiring:** For interfacing actuators with the Raspberry Pi Pico.

**Connectivity and Control:**

* **Power Supplies:** Adequate for powering the Raspberry Pi Pico, sensors, and actuators.
* **PCB:** Custom board for neatly organizing and connecting electronic components.

**Supplementary Materials:**

* **Tubing and Fittings:** For constructing the water and fertilizer dispensing system.
* **Fasteners and Mounting Hardware:** For securing components within the enclosure.

**Testing Supplies:**

* **Testing Equipment:** Tools like a multimeter and potentially an oscilloscope for circuit testing.
* **Consumables:** Solder, wire, and other materials for assembly and maintenance.

**Software and Development:**

* **Development Environment Subscriptions:** For programming the Raspberry Pi Pico and backend server (AWS).
* **Mobile Development Framework:** Such as React Native for app development connected to the hardware.