

Senior Design 2

AzulTank - A Fish Tank Helper



Group 14:

Gabriel Besana	<i>Electrical Engineering</i>
Rafael Nieves	<i>Computer Engineering</i>
Jazz Olario	<i>Computer Engineering</i>
Christian Rosado Arroyo	<i>Computer Engineering</i>

Reviewer Committee:

Mark Maddox	<i>ECE</i>
Mike Borowczak	<i>ECE</i>
Wei Sun	<i>ECE</i>

Mentor:

Dr. Lei Wei
Dr. Chung Yong Chan

Table of Contents

Chapter 1 - Executive Summary.....	1
Chapter 2 - Project Description.....	3
2.1 Project Motivation & Background.....	3
2.2 Pre-existent Developments.....	4
2.3 Goals and Objectives.....	5
2.4 Description of Features/Functionalities.....	6
2.4.1 General.....	6
2.4.2 Hardware.....	6
2.4.3 App and UI Integration.....	7
2.5 Engineering Specifications & Requirements.....	8
2.6 House of Quality.....	9
2.7 Hardware diagram/flowchart.....	11
2.8 Software diagram/flowchart.....	12
Chapter 3 - Research.....	12
3.1 Hardware Research.....	13
3.1.1 Temperature Readings.....	14
3.1.2 Temperature Sensor Selection.....	15
3.1.3 pH Readings.....	16
3.1.4 pH Sensor Selection.....	18
3.1.5 Turbidity Readings.....	19
3.1.6 Turbidity Sensor Selection.....	21
3.1.7 LED Lights.....	22
3.1.8 LED Light Strips Selection.....	24
3.1.9 Motors.....	25
3.1.10 Servo Motor Selection.....	27
3.1.10 Stepper Motor Selection	27
3.1.11 Connectivity.....	28
3.1.12 Wi-Fi Board Selection.....	30
3.1.13 Custom PCB & Development Boards.....	31
3.1.14 Microcontroller.....	33
3.1.15 Microcontroller Selection.....	34
3.1.16 AC - DC Converters.....	39
3.1.17 AC - DC Converter Selection.....	41
3.1.20 Voltage Regulation.....	43
3.1.21 Switching Regulator Selection.....	45
3.2 Software Research.....	47

3.2.1 PCB Design Software.....	47
3.2.2 Development Technologies.....	50
3.2.3 Database.....	53
3.2.4 Backend Technologies.....	54
3.2.5 Operating System.....	55
Chapter 4 - Related Standards and Design Constraints.....	58
4.1 Industrial Standards.....	59
4.1.1 PCB Standards.....	59
4.1.2 Sensor Standards.....	60
4.1.3 Communication Standards.....	61
4.1.3.1 UART Communication.....	62
4.1.3.2 Analog Communication.....	63
4.2 External Design Constraints.....	64
4.2.1 Time.....	64
4.2.2 Economic.....	65
4.2.3 Sustainability.....	66
4.2.4 Other Constraints.....	67
4.2.4.1 Political.....	67
4.2.4.2 Ethical.....	68
4.2.4.3 Health and Safety.....	68
Chapter 5 - ChatGPT.....	70
5.1 Case Study 1.....	71
5.2 Case Study 2.....	71
5.3 Case Study 3.....	72
5.4 Case Study 4.....	72
5.5 Case Study 5.....	73
5.6 Case Study 6.....	74
5.7 Impact of Artificial Intelligence.....	76
Chapter 6 - Hardware Design.....	76
6.1 Power Delivery/ Electrical Power System Schematics.....	77
6.2 MCU Board Schematics.....	78
6.2.1 ESP32 Module.....	78
6.2.2 Connectors.....	80
6.2.3 USB Connector.....	81
6.2.4 USB to Serial Conversion.....	81
6.2.5 Serial Signals Handling.....	83
6.2.6 Power Supply Section.....	83
6.2.7 User and Reset Buttons.....	85

6.2.8 Stepper Motor Controller.....	85
6.2.9 Terminals.....	86
6.2.9.1 Led Lights Terminal.....	87
6.2.9.2 Turbidity Sensor Terminal.....	87
6.2.9.3 Servo Motor Terminal.....	87
6.2.9.4 Ph Sensor Terminal.....	87
6.2.9.5 Temperature Sensor Terminal.....	88
6.2.9.6 Regulator Terminal.....	88
6.3 Turbidity Sensor Board Schematics.....	89
6.4 pH Sensor Board Schematics.....	90
Chapter 7 - Software Design.....	91
7.1 Landing Page.....	91
7.2 Settings Page.....	93
7.3 Schedule.....	94
7.4 Add New Schedule.....	95
7.5 Edit Schedule.....	96
7.6 Start Time.....	98
7.7 Repeat Day.....	99
7.8 Activity Diagram.....	100
7.9 Use Case Diagram.....	101
7.10 Class Diagram.....	103
Chapter 8 - System Fabrication.....	104
8.1 PCB.....	104
8.1.1 Regulator PCB Layout Design.....	105
8.1.2 MCU PCB Layout Design.....	106
8.1.3 Turbidity Sensor PCB Layout Design.....	108
8.1.4 pH Sensor PCB Layout Design.....	109
Chapter 9 - System Testing and Prototyping.....	110
9.1 Hardware Testing.....	111
9.1.1 Sensors.....	112
9.1.1.1 Temperature Sensor.....	112
9.1.1.2 pH Sensor.....	113
9.1.1.3 Turbidity Sensor.....	113
9.1.2 Dispensers.....	113
9.1.2.1 Food Dispenser.....	114
9.1.2.2 pH Dispenser.....	114
9.1.3 Motors.....	115
9.1.4 LED Lights.....	116

9.1.5 PCB.....	117
9.1.5.1 Testing Voltage Regulator PCB.....	117
9.1.5.2 Testing ESP32 PCB.....	117
9.1.5.3 Testing USB Connector in MCU PCB.....	118
9.1.5.4 Testing Sensor PCBs.....	118
9.1.6 Microcontroller.....	119
9.2 Software Testing.....	120
9.2.1 General User Interface.....	120
9.2.2 Data Storage.....	121
9.2.3 Notifications.....	122
9.3 Identifying and Resolving Issues.....	123
Chapter 10 - Administration.....	123
10.1 Budget and Financing.....	123
10.2 Distribution of Work.....	124
10.3 Project Milestones.....	125
Chapter 11 - Conclusion.....	127

List of Figures

Figure 2.1: House of Quality by authors	10
Figure 2.2: Hardware Diagram by authors	11
Figure 2.3: Software Diagram by authors	12
Figures 5.1 and 5.2: First attempt at DALL E image generation	75
Figures 5.3 and 5.4: Second attempt at DALL E image generation	75
Figure 6.1: Schematic of 5V to 12V Regulator	77
Figure 6.2: Schematic of ESP32 Module	79
Figure 6.3: Schematic of ESP32 Connectors	80
Figure 6.4: Schematic of USB Connector	81
Figure 6.5: Schematic of USB to Serial Connections	82
Figure 6.6: Schematics of Serial Signals Handling	83
Figure 6.7: Schematics of Power Supply Section.....	84
Figure 6.8: Schematics of Buttons.....	85

Figure 6.9: Schematics of Stepper Motor Controller.....	86
Figure 6.10: Schematics of Terminals Section.....	86
Figure 6.11: Schematics of Turbidity Board.....	88
Figure 6.12: Schematics of pH board.....	89
Figure 7.1: Landing Page	91
Figure 7.2: Settings Page	92
Figure 7.3: Schedule Page	93
Figure 7.4: Add Page	95
Figure 7.5: Edit Page 1	96
Figure 7.6: Edit Page 2	97
Figure 7.7: Start/Edit Time	98
Figure 7.8: Repeat Day	99
Figure 7.9: Activity Diagram.....	100
Figure 7.10: Smart Aquarium Use Case Diagram	101
Figure 7.11: Class Diagram	102
Figure 8.1: PCB Layout Design of Regulator board.....	104
Figure 8.2: PCB Layout Design of MCU board.....	105
Figure 8.3: PCB Layout Design of Turbidity board.....	107
Figure 8.4: PCB Layout Design of pH board.....	108

List of Tables

Table 2.1: Specifications for the project	8
Table 3.1: Tech Comparison of Temperature Reading devices	15
Table 3.2: Part Comparison of possible Temperature Sensors	16
Table 3.3: Tech Comparison of pH Reading devices	17
Table 3.4: Part Comparison of possible pH Sensors	19
Table 3.5: Tech Comparison of Turbidity Reading devices	20

Table 3.6: Part Comparison of possible Turbidity Sensors	21
Table 3.7: Tech Comparison of LED Bulbs and LED Strips	23
Table 3.8: Part Comparison of possible LED Lights Strips	24
Table 3.9: Tech Comparison of Servo and Stepper Motors	26
Table 3.10: Part Comparison of possible Servo Motors	27
Table 3.11: Part Comparison of possible Stepper Motors	28
Table 3.12: Comparison of Connectivity Technologies	30
Table 3.13: Part Comparison of possible Wi-Fi enabled devices	31
Table 3.14: Tech Comparison of PCB and Dev Boards	33
Table 3.15: Part Comparison of possible Microcontrollers	38
Table 3.16: Tech Comparison of AC-DC Converters	41
Table 3.17: Part Comparison of possible AC-DC Converters	43
Table 3.18: Tech Comparison of Linear and Switching Regulators	45
Table 3.19: Part Comparison of possible Voltage Regulators	46
Table 3.20: Tech Comparison of PCB Design Software	50
Table 3.21: Tech Comparison between Flutter and React Native	53
Table 3.22: Tech Comparison between backend technologies	55
Table 3.23: Tech Comparison of iOS and Android	58
Table 10.1: Estimated Budget and Financing for the project	122
Table 10.2: Distribution of Work for the project	123
Table 10.3: Estimated time of completion for Project Milestones for Senior Design 1	124
Table 10.4: Estimated time of completion for Project Milestones for Senior Design 2	125

Chapter 1 - Executive Summary

The driving force behind our senior design project, the AzulTank, a smart aquarium that simplifies and improves the experience of owning pet fish. Many fish owners struggle to maintain a consistent and healthy environment for their aquatic pets due to busy lifestyles, forgetfulness, or travel commitments. This is particularly challenging as finding a reliable person to care for fish during absences can be difficult, and transporting a fish tank is not practical. By incorporating advanced technology into a fish tank, we aim to ensure that fish receive optimal care without requiring constant attention.

The AzulTank is designed to monitor and regulate water quality, temperature, and oxygen levels, automate feeding schedules, and provide remote monitoring through a live stream. These capabilities are intended to foster healthier, longer-living fish while offering a more enjoyable and stress-free experience for fish owners. The smart aquarium goes beyond convenience, addressing the overall well-being of fish and making it easier for owners of all experience levels to maintain a healthy environment for their aquatic pets. For newcomers, taking care of fish can be a daunting task, requiring continuous monitoring to ensure the tank's health. A smart aquarium acts as a safety net by providing real-time feedback and automatic adjustments, making fishkeeping more accessible. Even for experienced aquarists, this technology offers advanced monitoring solutions, allowing them to focus on the more enjoyable aspects of their hobby. Ultimately, this technology not only improves the quality of life for fish but also encourages better care practices across the board.

In the sections that follow up, we dive into the motivation and background of the project, outlining our goals, objectives, and design requirements, which include basic, advanced, and stretch goals. The description of features and functionalities references input from customers and marketing analysis of comparable products and projects. We also review existing products, past projects, and prior related work, setting the stage for our project's approach. We included a table of key engineering specifications with quantitative measures, highlighting three demonstrable specifications for Senior Design 1.

In the Research and Investigation chapter, we provide a comprehensive comparison of various technologies and components used in the Smart Aquarium. Detailed tables and explanations showcase the rigorous selection process undertaken to choose the most suitable hardware and software components, ensuring optimal performance and cost-effectiveness. We also researched Standards and Design Constraints pertaining to our project, which provides guidelines for designing our PCB, and environmental management systems. This adherence helps reduce hazardous waste and promotes sustainability. This section discusses various standards and design constraints in detail, explaining their impact on the project.

Right after, we explore the idea of using ChatGPT to influence the AzulTank. Integrating AI could provide significant feedback that could improve our workflow by providing quick and accurate responses to various queries, saving time and enhancing efficiency. AI, particularly ChatGPT, offers enhanced problem-solving capabilities and automates routine tasks, allowing us to focus on creative and strategic aspects of the project. It continuously provides up-to-date information, which is crucial in our field. We also have a look at DALL E, an image generating model, that helps us imagine what the final version of the AzulTank could look like once it's built.

Following this, the Hardware Design and Software Design chapters will cover the overall design aspects of the Smart Aquarium. The hardware design will include various diagrams and illustrations, emphasizing the need for a solid design structure. Specifically we will be laying down the schematics and layout for all our modules and components, showing the architecture behind each circuit pertaining to the project. While the software design will feature different types of diagrams and user interfaces, showcasing how the software components are developed and integrated. Once the design for the project is completed, we move onto the System Fabrication/Prototype Construction section, we will discuss the PCB layout and other aspects related to creating a functional prototype. The System Testing and Evaluation section will detail the testing processes and performance evaluations of both hardware and software components to ensure the system's overall effectiveness.

Finally, the Administrative Content chapter will address the financial aspects of the project, including budget details, a bill of materials, and project milestones for both phases of the project. This section will also outline work distributions, highlighting the teamwork involved. We conclude by summarizing the project's achievements, reflecting on the overall process and outcomes. The Smart Aquarium project aims to solve common challenges faced by fish owners, demonstrating our ability to create innovative engineering solutions. By providing an automated solution for fish care, we enhance the experience for both novice and experienced aquarists. This project not only helps us tackle real-world problems but also prepares us for future careers in the engineering industry by showcasing our teamwork, project management, and technical development skills.

This project showcases our proficiency in collaboration, combining elements of electronics and software engineering. By developing a product that not only simplifies fish care but also enhances the owner's experience, we demonstrate the potential of engineering to improve everyday life. Our report on the AzulTank showcases the importance of using technology to create sustainable, efficient, and user-friendly solutions that address real-world problems.

Chapter 2 - Project Description

2.1 Project Motivation & Background

The motivation behind creating a smart fish tank aquarium that automates the care of fish stems from the desire to simplify and enhance the experience of being a pet fish owner. Many fish enthusiasts and pet owners often struggle to maintain a consistent and healthy environment for their aquatic pets due to busy schedules, forgetfulness, or the need to travel. The last concern is a significant contribution to our motivation behind the project, as it's not always easy for a fish owner to find someone who can take care of their fish pets. Moving a fish tank somewhere else while you are out of the country or far away from home is probably not a feasible idea, and finding someone trustworthy who can come into your house every day to take care of your fish may not always be possible.

By developing and integrating innovative technology into a fish tank, owners can ensure their fish receive optimal care without constant physical intervention. This innovation aims to monitor water quality, allow automatic feeding schedules, water treatment through a manual pH tablet dispenser and remote checks through a live stream. Consequently, this can lead to healthier, longer-living fish and a more enjoyable, stress-free experience for the owners by reducing the risk of common issues such as forgetting to feed the fish or letting poor water conditions go unnoticed for long periods of time.

The importance of a smart fish tank aquarium goes beyond convenience as it addresses the well-being of the fish regardless of experience level. For a beginner, taking care of fish may be a steep learning curve, requiring constant aquarium monitoring to ensure its well-being. Our approach provides a safety net by offering real-time feedback and automated adjustments, making fishkeeping more accessible to novices. On the other hand, seasoned aquarists won't likely face this issue. However, this technology is still beneficial in their case since it offers an advanced monitoring solution, allowing them to focus on the more enjoyable aspects of their hobby. Ultimately, this technology not only enhances the quality of life for the fish but also promotes better care practices across the board.

As engineering students, creating a smart fish tank aquarium offers us a unique opportunity to apply the knowledge and skills acquired throughout our studies to solve a real-world problem since the level of freedom this project provides is one we won't likely be able to experience under the rigorous schedule of an engineering company. Additionally, it tackles our challenges of owning and maintaining a fish tank while balancing personal and academic responsibilities, such as the need to travel out of the country or spend extended hours on campus. This project not only serves as a practical solution to our personal challenges or different experience level owners but also highlights our abilities by working as a team and applying project management skills and hardware and

software development skills, preparing us for future endeavors in the engineering industry.

2.2 Pre-existent Developments

There are currently several products on the market aimed at enhancing the experience of owning pet fish. However, many of these products focus on improving only a few specific aspects of fish care. Whether it's simplifying routine maintenance or providing notifications about the smallest details of your aquarium's ecosystem, these solutions often leave the impression that something is still missing. As we explored similar products and inventions, we found that many were either prohibitively expensive or lacked the durability and reliability needed for long-term use.

An example of a product currently available on the market is 'Glass Aqua.' This innovative solution simplifies aquarium creation and maintenance, offering both aesthetic appeal and functional design to ensure a healthy and happy environment for fish. While it enhances the consumer experience with its user-friendly features, 'Glass Aqua' falls on the pricier side of the spectrum, making it less accessible for the average consumer to fully utilize its potential.

Additionally, several other products on the market today incorporate similar innovations. For example, smart aquarium systems can monitor water conditions and alert users to irregularities, while some tanks feature built-in devices capable of feeding fish or filtering water with a simple tap on a mobile app. Notable examples include the Maxspect Lagoon, Fluval Flex, and Xiaomi Smart Fish Tank, among others, which showcase some or all of these advanced features.

A past senior design project we reviewed incorporated many features we aimed to include to enhance the experience of owning a fish. It utilized various technologies similar to those found in products like 'Glass Aqua,' but at a more affordable and efficient level. However, after analyzing the senior design project and existing products, we identified areas for improvement. Our goal as a group is to innovate further, building upon previous designs to create an enhanced and more comprehensive product.

As we delve deeper into our research, we recognize that our concept builds upon existing innovations in aquarium technology rather than introducing something entirely new. However, our aim is to reimagine these advancements by creating a solution that prioritizes practicality and affordability. We envision a product that simplifies the maintenance and care of aquariums, ensuring longevity and health for aquatic life. By focusing on budget-friendly design, we strive to offer consumers a comprehensive tool that provides all the essential features they need—delivered seamlessly through the press of a button. This approach ensures that advanced aquarium care becomes accessible to a broader audience without compromising on quality or functionality.

2.3 Goals and Objectives

Basic Goals:

- Provide a comprehensive system for managing an aquarium, ensuring the well-being of the fish and ease of use for the owner.
- Ensure effective monitoring and control.
- Maintain optimal water conditions.
- Automated feeding.
- Automatic Lighting.

Basic Objectives:

- Implement a pH sensor to accurately measure the water's pH level.
- Integrate an automated food dispenser to ensure fish are fed at scheduled intervals.
- Continuously monitor water temperature using a temperature sensor to ensure it remains within a suitable range for fish health.
- Incorporate a pH solution dispenser to automatically adjust the water's pH level based on sensor readings.
- Control a smart lighting system that simulates natural light cycles and enhances the tank's aesthetic appeal.
- Include a turbidity sensor to monitor water clarity and detect any quality issues.
- Develop an intuitive mobile application for remote monitoring and control, allowing users to manually override automated systems.

Stretch Goals:

- Integrate a live feed camera to provide real-time video of the fish tank, allowing for remote monitoring.
- Implement an air bubble system controlled by a Raspberry Pi or PCB to enhance oxygen levels while adding an aesthetic touch.
- Create a comprehensive analytics dashboard that logs historical data for all monitored parameters and sends alert notifications for critical conditions, such as pH imbalances and temperature extremes.

By achieving these goals and meeting the design requirements, the AzulTank will provide a comprehensive solution for maintaining a healthy and thriving aquarium environment.

2.4 Description of Features/Functionalities

After researching the available 'smart' aquariums on the market, we found that none offered an automatic pH solution dispenser or a combination of temperature, pH, and turbidity sensors. Some auto feeders included a live-stream camera, but like temperature sensors, they were sold as standalone products. Among the smart aquariums we found, most only offered basic features like self-cleaning systems or simple smart lighting with limited color options. Overall, we could not find any product that combined all the desired features, and none included a turbidity sensor or an automatic pH solution dispenser.

Between Spring and Summer 2020, a senior design team developed AquaEco, a smart aquarium that featured an automatic feeder, sensors to monitor water temperature, pH, and turbidity levels, and RGB lighting that allowed the user to manually select colors. While this project was impressive, it lacked several features, such as live-streaming functionality for monitoring the tank remotely, automation of a pH dispenser, a bubble system, and using the smart lighting as a visual cue to indicate water quality. By comparing current market products with past senior design projects, we identified a set of features to incorporate into our version of the smart aquarium.

2.4.1 General

- The enclosure for the feeder will be constructed using PLA, a cheap, easy-to-use, and compostable 3D printing material.
- The enclosure will house the necessary electronics, fish food, and pH tablets.
- A cutout in the enclosure will create a space for the camera to capture a live stream, positioned against the tank's glass for a clear view of the fish activity.

2.4.2 Hardware

- A custom PCB will be powered by DC power and serve as the main board, managing the inputs and outputs of the connected components.
- A Raspberry Pi will power the camera for the fish tank livestream. It will be connected to the main PCB, which will transmit the footage after receiving a command.
- A Wi-Fi module will be connected to the custom PCB, providing data to the mobile app and enabling remote control.

- A servo motor, powered by the PCB, will be used to dispense the fish food.
- A temperature sensor will track the water's temperature.
- A pH sensor will keep track of the water's pH level.
- A stepper motor, powered by the PCB will be used to dispense pH tablets.
- A turbidity sensor will monitor the water quality, triggering a smart lighting system and app notifications to inform the user when it's time to replace the water.

2.4.3 App and UI Integration

- The mobile application will focus on easy-to-use, featuring a modern interface, which will allow the user to control the smart capabilities of the fish tank. For example:
 - o Smart Lighting System
 - o Food Dispenser
 - o pH Tablet Dispenser
 - o Livestream Camera Access
 - o Scheduling System
 - o Alerts System
- Notifications:
 - o Notification sent when the feeder automatically dispenses food.
 - o Notification sent when the pH sensor detects the pH level is below or above optimal conditions.
 - o Notification sent when the turbidity sensor detects the water quality is below optimal conditions.
 - o Notification sent when the water sensor detects the temperature is above or below optimal conditions.

2.5 Engineering Specifications & Requirements

The table below lists our project's specifications that shows clear metrics for testing and assessing success. We will prioritize maintaining these specifications by thoroughly testing the prototype and final product.

Please note that these specifications are subject to change given our limited experience with large-scale projects such as this. We also included the overall project budget to further emphasize and highlight the importance of balancing cost-effectiveness with performance standards.

While each component has specific requirements that are crucial for maintaining optimal conditions for aquatic life, we chose to highlight the sensors, water pump control and servo motor control to illustrate how these components work together to support the health and the stability of the aquatic environment.

Table 2.1: Specifications for the project.

Specification	Criteria
Power Supply	Wall outlet power
Max Power Consumption	~24 Watts
pH Sensor Response time	Response Time: ≤ 2 min
Temperature Sensor Response Time	Response Time: ≤ 2 min
Temperature Sensor Accuracy	$\pm 0.9^{\circ}\text{F}$ from 14°F to 185°F
pH Sensor Accuracy	pH level must be between 4 and 10
Turbidity Sensor Accuracy	<5 NTU
Food Dispenser Accuracy	Food Dispenser must dispense between 0.189 Oz Average per 26 sec
pH Dispenser Accuracy	One tablet at a time
Dispensers Capacity	0.50 Oz
LED Response Time	2ms
Servo Motor Response Time	2ms
Stepper Motor Response Time	200ms
Cost	$<\$500$

For Demonstration:

- *pH Sensor Accuracy*
 - Ensures timely detection and changes of pH levels, which is vital for prompt actions to maintain optimal pH balance for the aquarium
 - Dispenser Accuracy must be between 4 and 10.
- *Food Dispenser Accuracy*
 - Ensures that the food dispenser will consistently dispense the same amount of ounces per sec.
 - Food Dispenser must dispense 0.189 Oz Average per 26 sec
- *pH Dispenser Accuracy*
 - Ensure that the pH dispenser will consistently dispense the same amount of tablet.
 - pH Dispenser must dispense 1 tablet at a time.

2.6 House of Quality

The House of Quality outlines the relationship between customer requirements and technical specifications. Strong positive correlations indicate that improving one aspect can significantly enhance customer satisfaction. For example, maximizing sensor accuracy (strongly correlated with accuracy, reliability, and automation) and ensuring robust Wi-Fi connectivity (vital for responsiveness and automation) are critical. Power consumption is to be minimized to enhance affordability and energy efficiency.

The relative weights of each technical specification reflect their overall importance, guiding the design focus towards features that will most effectively meet customer needs and expectations. By analyzing the House of Quality, the team can prioritize efforts and resources effectively. Features that show strong positive correlations with key customer requirements are given priority. This ensures that the design and engineering efforts are aligned with what customers value the most. This matrix helps ensure that the final product not only meets but exceeds customer expectations by focusing on key areas that drive satisfaction.

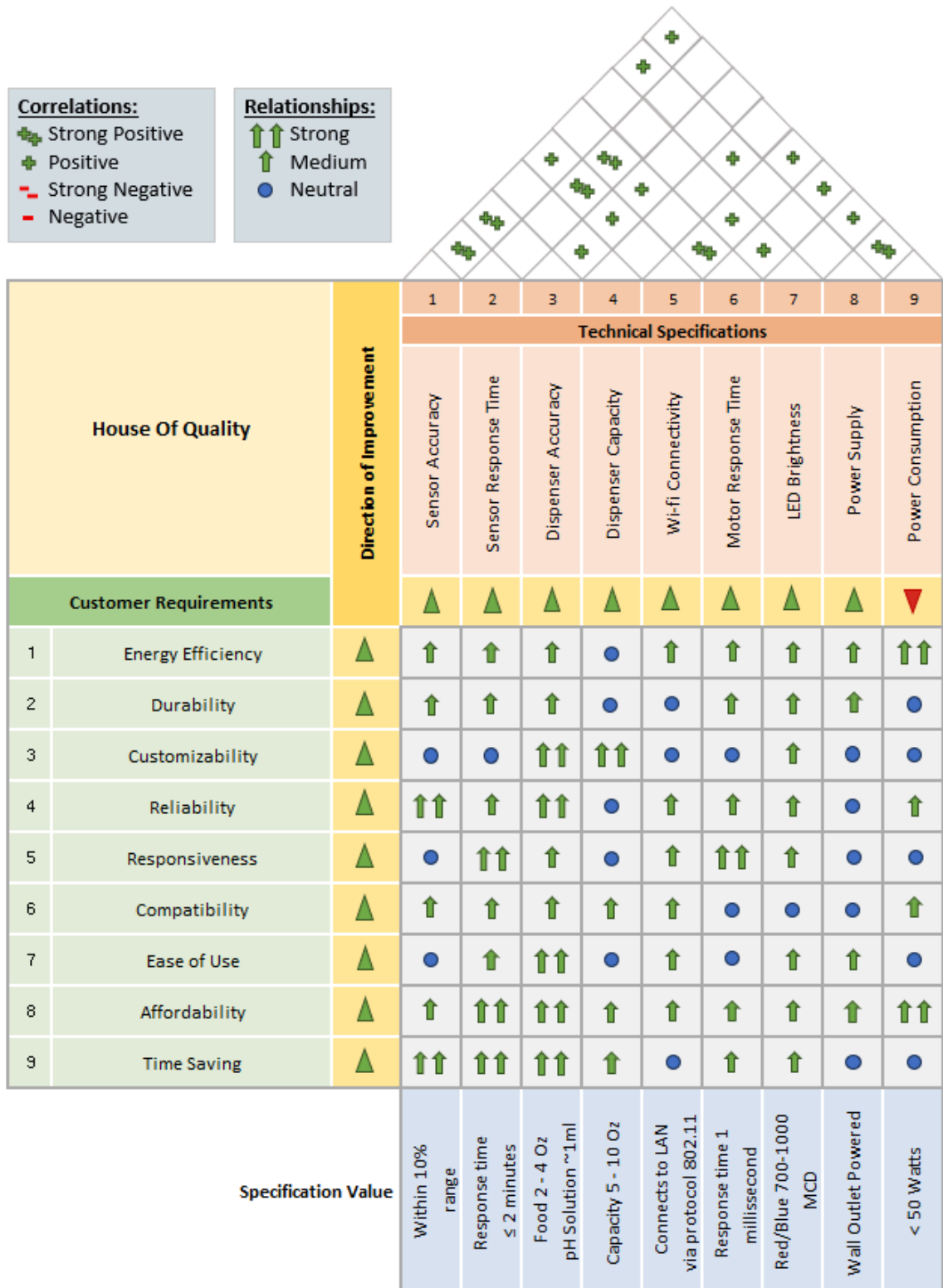


Figure 2.1: House of Quality by authors.

2.7 Hardware diagram/flowchart

The figure below shows the block diagram that we as a group designed to provide an adequate visual representation of the aspects of hardware that we will include in the project. Our goal is to have a system where we can have clear and concise interfacing and communication between the components. With the power supply unit, it will supply the necessary power to the main components which include our sensors, dispensers, Raspberry Pi, and microcontroller. The main purpose of the diagram is to present the roles that each member of the group will have within the creation of this design. Lastly, this diagram shows the specifications that we have set to meet the overall requirements.

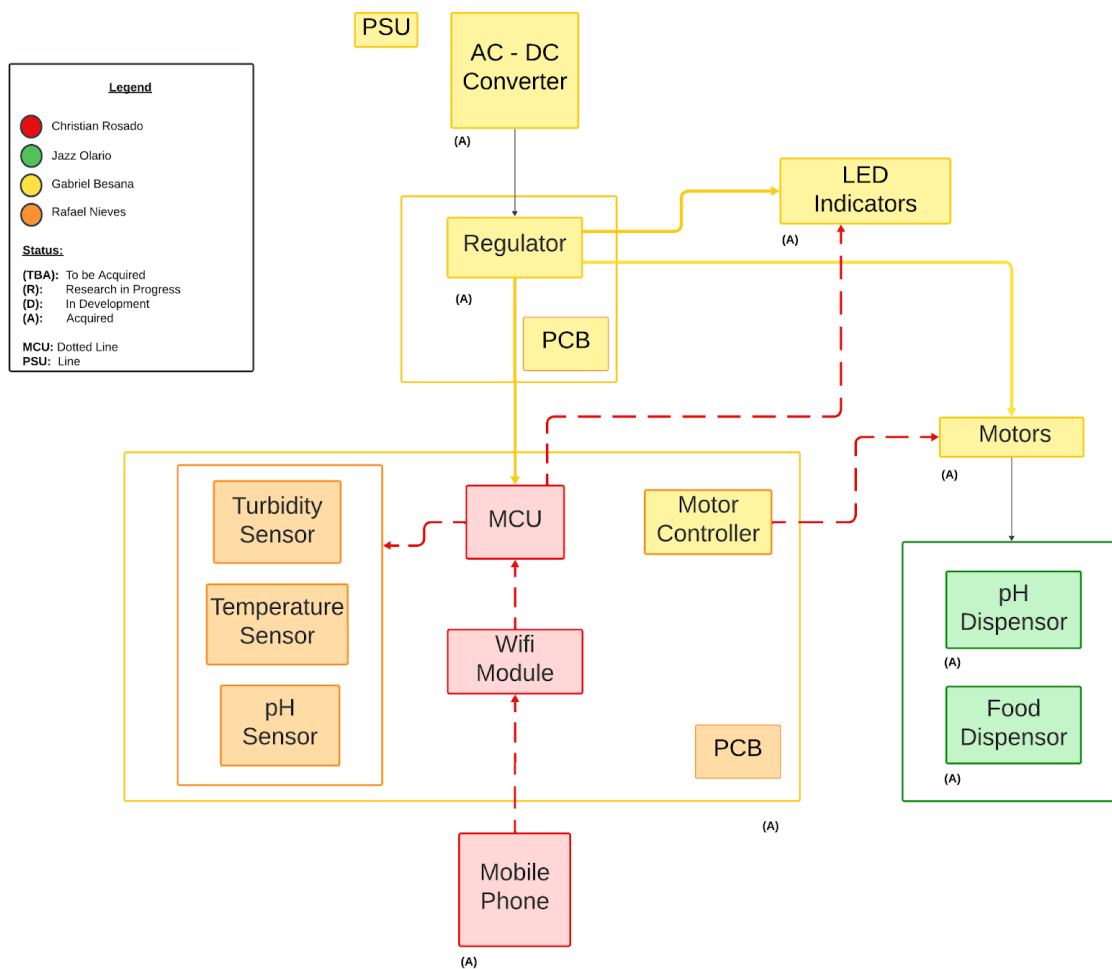


Figure 2.2: Hardware Diagram by authors.

2.8 Software diagram/flowchart

The figure below is the software flowchart/diagram, providing a visual description of the Azul Tank's functionality. It is divided into two blocks: the PCB software and the companion mobile app, illustrating the integration between them. The accompanying legend shows the work distribution among team members, as well as the status of each block, offering a clear overview of the project's progress.



Figure 2.3: Software Diagram by authors.

Chapter 3 - Research

Thorough research and careful part selection are crucial to ensure the project's success. This section provides an overview of the methodologies and criteria used to identify and select the most suitable components for our project. The goal was to choose parts that meet the essential requirements of accuracy, reliability, ease of integration, and cost-effectiveness. Each component was selected based on its performance characteristics, compatibility with other system elements, and ability to fulfill the project's overall objectives. By prioritizing high-quality, reliable parts, the system aims to achieve functionality, ease of use, and long-term durability, ensuring a healthy and well-maintained aquarium environment. Research helps identify and avoid potential complications that might cause costly delays and potential failures. Our research is further divided into two main categories: hardware and software.

The part selection phase is a critical section of our project, as it determines the performance, reliability, and overall success of the system. In this chapter, we will delve into the process of selecting the appropriate components to meet our design requirements and goals. Our objective is to identify and evaluate sensors, controllers, and other key hardware elements that will ensure optimal water conditions, automate feeding, and facilitate effective monitoring and control.

We will begin by examining the criteria used for selecting each component, considering factors such as accuracy, durability, compatibility, and cost-effectiveness. This process involves a thorough comparison of available options from various manufacturers, evaluating their specifications and performance metrics to ensure they align with our project needs.

For each category of components, we will provide a detailed analysis of the top candidates. This analysis can include technical specifications, pricing, and compatibility with our system architecture, as well as any trade-offs and considerations involved in the selection process.

The goal of this chapter is to provide a comprehensive overview of the part selection process, highlighting the rationale behind each choice and ensuring that our final selections will enable the successful implementation of a robust, efficient, and user-friendly fish tank management system. By carefully selecting the best components, we aim to create a reliable and effective solution that meets the needs of both the aquatic environment and the fish tank owner.

3.1 Hardware Research

When looking for ideal components for the project, hardware research and considerations are important to guarantee the highest possible performance as well as its compatibility and cost-effectiveness. Thorough research allows for the selection of components that meet the specific needs of the project, technical specifications and compatibility of each part. Another thing to consider is the

durability and scalability of the hardware to prevent issues such as system failures or inefficient performance.

3.1.1 Temperature Readings

Temperature readings are essential for maintaining a healthy and stable environment in a fish tank. They ensure the well-being of aquatic life, prevent stress and disease, support breeding, maintain water quality, and enhance the overall user experience. By continuously monitoring and regulating water temperature, aquarium owners can provide optimal living conditions for their fish and other aquatic organisms, ensuring a thriving and vibrant aquarium ecosystem.

Digital Thermometer

A digital thermometer is a straightforward and user-friendly option for monitoring the temperature of a fish tank. It features an easy-to-read LCD display and provides quick temperature readings with minimal effort. Digital thermometers are generally affordable, battery-powered, and can be placed inside the tank for continuous temperature monitoring. However, their limited connectivity options make them less suitable for automated systems that require continuous data logging.

Temperature Sensor

On the other hand, temperature sensors are highly accurate and designed for continuous monitoring in aquatic environments. These sensors are fully submersible and can provide real-time digital data, which can be easily integrated into an automated monitoring system. They are ideal for users who want to automate their fish tank monitoring and control systems, ensuring stable temperature conditions without constant manual intervention.

Infrared Sensor

Lastly, Infrared temperature sensors provide high measurement capability with non-contact surface temperature readings. They offer instantaneous data accessibility and digital connectivity options like I2C or SPI, making them suitable for advanced monitoring applications. Despite their higher cost and the need for microcontroller integration, IR sensors' ability to provide quick, accurate measurements without direct contact is advantageous in scenarios where immersion is impractical. However, their detection range is limited to the line of sight, restricting their use to surface temperature monitoring only.

Table 3.1: Tech Comparison of Temperature Reading devices.

Criteria	Digital Thermometer	Temperature Sensor	Infrared Sensor
Measurement Capability	Measures temperature at a single point	Continuous temperature monitoring	Noncontact, surface temperature
Data Accessibility	Requires manual reading	Provides real-time digital data	Provides real-time digital data
Cost	\$10 - \$50	\$5 - \$30	\$20 - \$50
Power Consumption	Minimal (battery-powered)	Typically < 1 Watt	Typically < 1 Watt
Connectivity	None or limited (basic LCD display)	Analog/Digital output, can interface with microcontrollers	Digital (I2C or SPI)
Power Supply	Battery	Direct DC power (3.3V - 5V from a microcontroller)	Direct DC power (3.3V - 5V from a microcontroller)
Detection Range	Limited to physical contact point	Typically -55°C to +125°C	Limited to line of sight

We selected a temperature sensor over a digital thermometer because it offers several significant advantages. Temperature sensors provide continuous, real-time monitoring, ensuring that any fluctuations in temperature are detected immediately, which is crucial for maintaining a stable environment for the fish. Unlike digital thermometers, which require manual readings and only provide snapshots of the temperature at the time of measurement, temperature sensors allow for constant oversight. Temperature sensors can be integrated into automated systems that manage other aspects of the fish tank. This level of automation and integration is not possible with standalone digital thermometers.

Additionally, the temperature sensor offers data logging capabilities when connected to a microcontroller, enabling the logging of historical data for analyzing trends, diagnosing issues, and ensuring long-term stability in the tank environment. This historical data is valuable for making informed decisions based on temperature patterns. It is versatile and can be used in various applications beyond just monitoring the fish tank. They can be part of a comprehensive environmental monitoring system, interacting with other sensors to provide a holistic view of the tank's conditions.

3.1.2 Temperature Sensor Selection

When designing a comprehensive system, selecting the appropriate waterproof temperature sensor is crucial to ensure accurate and reliable monitoring of water temperature. Temperature control is vital for maintaining a healthy environment for aquatic life, and the sensors must be robust enough to withstand prolonged submersion in water.

Table 3.2: Part Comparison of possible Temperature Sensors.

Manufacturer	Model	Price	Size	Key Features
Dallas/Maxim	DS18B20 Waterproof	From \$4.50	Cable length: 1m (Sensor: 6mm diameter)	High accuracy ($\pm 0.5^{\circ}\text{C}$) 1-Wire communication
TE Connectivity	TSYS01P Waterproof	From \$12.00	Cable length: 1.5m (Sensor: 7mm diameter)	High accuracy ($\pm 0.2^{\circ}\text{C}$) I2C interface
TE Connectivity	HTU21D Waterproof	From \$15.00	Cable length: 1m (Sensor: 8mm diameter)	High accuracy ($\pm 0.3^{\circ}\text{C}$) I2C interface

The DS18B20 Waterproof temperature sensor stands out as the superior option compared to the other two sensors due to its exceptional combination of accuracy, durability, and ease of integration. This sensor offers a high degree of precision with an accuracy of $\pm 0.5^{\circ}\text{C}$, ensuring that the water temperature is consistently monitored within optimal ranges for the fish's health. Its waterproof design makes it ideal for continuous submersion in the fish tank, providing reliable and long-lasting performance even in aquatic environments.

Additionally, the DS18B20 is widely supported and easily interfaced with popular microcontrollers, facilitating seamless integration into our automated system. The sensor's robust construction ensures durability, reducing the need for frequent replacements, which translates into cost savings and reduced maintenance efforts over time. Moreover, its affordability, combined with the high-quality performance, makes the DS18B20 a cost-effective choice that does not compromise on functionality, making it the best option for maintaining a stable and healthy aquatic environment in the fish tank.

3.1.3 pH Readings

Monitoring pH readings in a fish tank is essential for maintaining a healthy and stable environment for aquatic life. Consistent and optimal pH levels help reduce stress in fish, making them less susceptible to diseases and infections. Fluctuations in pH can cause significant stress, weakening the immune system and increasing vulnerability to pathogens. Integrating pH sensors into the fish tank system allows for automated monitoring and alerts. This ensures that any deviations from the optimal pH range are quickly detected and corrected.

pH Strips

pH test strips offer a very basic and manual method for measuring the pH of water. They are cost-effective, requiring no power supply or connectivity, making them the simplest option available. However, they are not suitable for continuous monitoring and do not provide digital data accessibility. Test strips are limited to single-point manual readings and require direct contact with the sample, which can be less accurate and less convenient compared to digital meters and sensors.

Digital pH Meter

On the other hand, a digital pH meter is a handheld device used to measure the pH level of water at a specific point. It features a built-in display that shows the pH reading, making it easy to use and read. Digital pH meters are generally affordable and are powered by batteries, making them portable and convenient for spot checks. However, their lack of connectivity and limited data logging capabilities make them less ideal for automated systems.

pH Sensor

Lastly, pH sensors are designed for continuous monitoring of pH levels in real-time. These sensors can be integrated into automated systems and provide digital or analog outputs that can be read by microcontrollers. pH sensors are typically more expensive than digital pH meters, but they offer greater functionality and precision. They require a direct DC power supply, usually provided by the monitoring system, and consume minimal power. pH sensors are ideal for applications that require constant pH monitoring and data logging

Table 3.3: Tech Comparison of pH Reading devices.

Criteria	pH Test Strips	Digital pH Meter	pH Sensor
Measurement Capability	Basic, single point, manual	Measures pH at a single point via manual reading	Continuous pH monitoring in real-time
Data Accessibility	Manual Reading	Requires manual reading on a built-in display	Provides real-time digital or analog data

Cost	\$5 - \$15	\$20 - \$100	\$30 - \$150
Power Consumption	None	Minimal (battery-powered)	Typically < 1 Watt
Connectivity	None	None or limited (basic LCD display)	Analog/Digital output, interfaces with microcontrollers
Power Supply	None	Battery (typically AA or button cell)	Direct DC power (3.3V - 5V from a microcontroller)
Detection Range	Limited to manual, sample contact	Limited to the specific point of immersion	Continuous monitoring, typically 0-14 pH

We selected a pH sensor over a digital pH meter because it provides numerous benefits crucial for maintaining the optimal environment for aquatic life. A pH sensor offers continuous, real-time monitoring of the water's pH level, ensuring that any fluctuations are immediately detected and addressed. This constant oversight is essential for the well-being of the fish, as even minor changes in pH can have significant impacts on their health. Unlike digital pH meters, which require manual readings and only provide periodic snapshots of the pH level, sensors ensure ongoing vigilance.

pH sensors offer higher precision and customization in readings. They can be configured to send alerts if the pH deviates from a specified range, allowing for immediate corrective actions. This level of accuracy and customization is not available with digital pH meters, which cannot be programmed to trigger alerts or automated responses based on specific thresholds. They can also be integrated into automated systems that manage other aspects of the fish tank, such as dosing systems to adjust pH levels or water changes.

3.1.4 pH Sensor Selection

Maintaining the correct pH level is crucial for the health and wellbeing of aquatic life in an aquarium. pH sensors enable a more proactive approach to maintaining the optimal environment for fish, providing the capability to quickly detect and respond to pH changes. Selecting an affordable and reliable waterproof pH sensor is essential for accurately monitoring and adjusting the pH levels. This comparison aims to identify the best-suited pH sensors that meet the project's requirements for accuracy, durability, and ease of integration at a more affordable price point.

Table 3.4: Part Comparison of possible pH Sensors.

Manufacturer	Model	Price	Size	Key Features
DFRobot	Gravity: Analog pH Sensor Pro	From \$29.50	Probe length: 60mm, diameter: 12mm	Good accuracy (± 0.1 pH) Analog output
Milone Technologies	eTape Liquid Level Sensor (pH)	From \$49.95	Probe length: 120mm, diameter: 8mm	Good accuracy (± 0.05 pH) Analog output
DFRobot	Gravity: Analog pH Sensor	From \$56.90	Probe length: 90mm, diameter: 12mm	Good accuracy (± 0.1 pH) Analog output

The Gravity: Analog pH Sensor Pro is the superior choice among the available options due to its specialized design, ease of calibration, and extensive support for integration. Unlike other pH sensors, the Gravity: Analog pH Sensor Pro is specifically designed for aquariums and hydroponic systems, providing high stability and reliable readings over long periods. Its built-in temperature compensation ensures that pH measurements remain accurate despite variations in water temperature, which is critical for maintaining the delicate balance in a fish tank.

The sensor comes with a pre-calibrated electrode and an easy-to-follow calibration procedure, making it user-friendly and ensuring consistent accuracy with minimal effort. This feature is particularly beneficial for users who may not have extensive experience with pH monitoring equipment. Much like the DS18B20, the Gravity: Analog pH Sensor Pro is compatible with a wide range of microcontrollers which simplifies the integration process into our automated fish tank system.

The sensor also benefits from robust documentation and community support, which facilitates troubleshooting and enhances the overall user experience. Its rugged construction and waterproof probe ensure longevity and resilience in the aquatic environment, reducing maintenance needs and providing dependable performance. Overall, the Gravity: Analog pH Sensor Pro offers a blend of precision, durability, and ease of use, making it the ideal choice for effectively managing the pH levels in our fish tank.

3.1.5 Turbidity Readings

Turbidity measures the cloudiness or haziness of water caused by suspended particles such as algae, detritus, and waste products. High turbidity levels indicate poor water clarity, which can affect the overall aesthetic and health of the aquarium. Fish and other aquatic organisms thrive in clear water. High turbidity levels can cause stress, which weakens their immune systems and makes them more susceptible to diseases. By keeping turbidity levels in check, owners can

perform necessary maintenance tasks, such as water changes and filter cleanings, on a regular schedule.

Digital Turbidity Meter

A digital turbidity meter is a handheld device designed to measure the turbidity of water at a specific point, it requires to take a manual sample into a vial, which is then placed into the reader. It features a built-in LCD display that shows the turbidity reading, making it easy to use and read. These meters are moderately priced, but are still somewhat accessible for many users. They are powered by batteries, providing portability and convenience for spot checks. Digital turbidity meters are suitable for hobbyists and small-scale applications where periodic manual measurements are sufficient.

Turbidity Sensor

A turbidity sensor is designed for continuous real-time monitoring of water clarity. These sensors provide digital or analog outputs that can be read by microcontrollers, making them ideal for integration into automated systems. Turbidity sensors are generally more cost-effective for continuous monitoring, with prices typically lower than those of digital turbidity meters. They require a direct DC power supply, often provided by the monitoring system, and consume minimal power. Turbidity sensors are perfect for applications that demand constant turbidity monitoring and data logging, such as automated fish tank management systems.

Optical Particle Counter

Lastly, an optical particle counter offers advanced turbidity measurements by counting and sizing particles in water, providing detailed continuous monitoring. They require integration with a microcontroller for data display and analysis. While more expensive, they offer high precision and detailed data, making them suitable for applications requiring granular water quality analysis. Their medium power consumption and requirement for microcontroller power supply add complexity but offer advanced capabilities. The wide detection range and detailed particle data make optical particle counters ideal for environments where precise turbidity control is critical.

Table 3.5: Tech Comparison of Turbidity Reading devices.

Criteria	Digital Turbidity Meter	Turbidity Sensor	Optical Particle Counter
Measurement Capability	Measures turbidity at a single point via manual reading	Continuous turbidity monitoring in real-time	Continuous and detailed monitoring in real-time

Data Accessibility	Requires manual reading on a built-in display	Provides real-time digital or analog data	Provides real-time digital or analog data
Cost	\$50 - \$200	\$20 - \$80	\$100 - \$300
Power Consumption	Minimal (battery-powered)	Typically < 1 Watt	Medium to high (varies depending on equipment)
Connectivity	None or limited (basic LCD display)	Analog/Digital output, interfaces with microcontrollers	Analog/Digital output, interfaces with microcontrollers
Power Supply	Battery (typically AA or button cell)	Direct DC power (3.3V - 5V from a microcontroller)	Direct DC power (10V - 15V)
Detection Range	Limited to the specific point of immersion	Continuous monitoring across the entire tank	Wide range with detailed particle data

We selected a turbidity sensor over a digital turbidity meter because it provides several key advantages crucial for maintaining water clarity and the overall health of the aquatic environment. A turbidity sensor offers continuous, real-time monitoring, allowing for immediate detection of any changes in water turbidity. This constant monitoring is essential, as it ensures that any issues affecting water clarity are quickly addressed, preventing harmful conditions for the fish. In contrast, digital turbidity meters require manual readings, providing only periodic snapshots of turbidity levels. This can result in delays in identifying water quality problems, potentially leading to adverse conditions for the fish before corrective measures are implemented.

3.1.6 Turbidity Sensor Selection

To ensure the clarity and quality of water in an aquarium, monitoring turbidity is essential. Selecting an affordable and reliable waterproof turbidity sensor is crucial for detecting changes in water clarity due to suspended particles. This table provides a comparison of affordable waterproof turbidity sensors, helping to select the most appropriate sensor for maintaining optimal water clarity within the fish tank without exceeding budget constraints.

Table 3.6: Part Comparison of possible Turbidity Sensors.

Manufacturer	Model	Price	Size	Key Features
Seed Studio	Grove - Turbidity Sensor v1.2	From \$33.90	Probe length: 45mm, diameter: 33mm	Good accuracy Analog output Up to 100 NTU

DFRobot	Gravity: Analog Turbidity Sensor	From \$9.90	Probe length: 42mm, diameter: 32mm	Good accuracy Analog output Up to 100 NTU
Vernier	Turbidity Sensor	From \$124.00	Probe length: 50mm, diameter: 30mm	Very High accuracy Analog output Up to 200 NTU

The Gravity: Analog Turbidity Sensor is the better choice over the other two options due to its precise measurement capabilities, robust construction, and straightforward integration process. This sensor is designed to provide accurate and reliable measurements of water clarity, which is crucial for detecting the presence of particles that can affect the health of the fish. Its high sensitivity allows it to detect even minor changes in turbidity levels, ensuring that the water quality remains optimal.

This sensor is supported by comprehensive documentation and a big community of users, which can be helpful for troubleshooting and optimizing the system. The combination of precision, durability, ease of use, and integration capabilities makes the Gravity: Analog Turbidity Sensor the best option for maintaining clear and healthy water conditions in the fish tank, ensuring the well-being of the fish and the overall success of the tank management system.

3.1.7 LED Lights

LED lights play a crucial role in simulating natural light cycles, enhancing the looks of the aquarium, and promoting the health and well-being of the fish. Proper lighting is essential for maintaining the biological rhythm of the fish and encouraging natural behaviors. The ability to control the intensity and color of the LED lights through our app further allows for custom lighting conditions that can adapt to different species' needs and various times of the day. By integrating LED lights, we ensure a balanced and vibrant aquatic environment that contributes to the overall success of the project.

LED Light Bulbs

LED Light Bulbs are designed to replace traditional incandescent and fluorescent bulbs in various fixtures, offering a direct, energy-efficient alternative. They are known for their ability to produce a high-quality, bright light while consuming significantly less energy. This efficiency results in lower electricity bills and a reduced environmental footprint, making them a sustainable choice for lighting. They can be used in standard fixtures, providing focused, directional light that is ideal for general room lighting, task lighting, and decorative purposes.

LED Light Strips

LED Light Strips consist of multiple LEDs mounted on a narrow, flexible circuit board. LED strips are highly versatile and can be cut to specific lengths, bent around corners, and installed in tight spaces, making them ideal for accent lighting, under-cabinet lighting, and creative projects. Another thing to consider is that one of the key benefits of LED strips is their ability to provide even, diffuse lighting over a large area. LED strips are available in a wide range of colors, including RGB options that can be programmed for dynamic lighting effects.

LED Panel Lights

LED panel lights are designed to provide high light output with a longer lifespan of around 50,000 hours, making them a robust option for aquarium lighting. They offer moderate customization options, with some flexibility in installation and configuration. Costing between \$20 and \$100 per panel, they require a power adapter or controller, similar to LED strips. LED panel lights consume low to moderate power, providing efficient lighting solutions for larger or more demanding aquarium setups.

Table 3.7: Tech Comparison of LED Bulbs and LED Strips.

Criteria	LED Light Bulbs	LED Light Strips	LED Panel Lights
Light Output	Typically provides directional lighting	Offers flexible, linear lighting for even distribution	Directional, high output directional lighting
Lifespan	Long lifespan (15,000-50,000 hours)	Long lifespan (20,000-50,000 hours), but can vary based on usage and quality	Long lifespan (20,000-50,000 hours), but can vary based on usage and quality
Customization	Limited to bulb shape and size.	Customizable length, color, and brightness.	Moderate (some flexibility)
Cost	Generally lower initial cost per unit. Typically < \$12	Can be more expensive overall, depending on length and features. Typically < \$22	\$20 - \$100 per panel, depending on length and features
Power Supply	Typically uses standard household AC voltage	Requires a DC power supply, usually 12V or 24V	Requires a DC power supply, usually 12V or 24V

Power Consumption	Generally low power consumption (6-15 watts)	Variable power consumption (4-20 watts per meter) depending on length and density	Variable power consumption (4-20 watts per meter) depending on length and density
--------------------------	--	---	---

We selected LED light strips because they offer several significant advantages over LED light bulbs, making them a superior choice for many applications. One of the primary benefits is their flexibility and versatility. LED strips can be bent, cut, and shaped to fit various spaces and designs. They provide continuous, linear illumination, resulting in even and diffused lighting across a surface.

Additionally, LED light strips often come with advanced features like color-changing capabilities, dimmability, and programmable lighting effects. Their slim profile allows for installation in tight or narrow spaces, making them suitable for the AzulTank. And in terms of energy efficiency, while both LED light strips and bulbs are energy-efficient, strips often allow for targeted lighting, reducing the need to illuminate an entire space and leading to additional energy savings, especially in accent or task lighting applications.

3.1.8 LED Light Strips Selection

When selecting an LED light strip, factors such as brightness, color options, control features, and waterproofing must be considered to ensure they meet the specific requirements of an aquarium environment. Additionally, compatibility with the MCU is essential. By considering these factors, we can choose an LED light strip that meets the aesthetic and biological needs of the aquarium.

Table 3.8: Part Comparison of possible LED Lights Strips.

Manufacturer	Model	Price	Size	Key Features
Adafruit	NeoPixel Digital RGB LED Strip	\$24.95 (per meter)	1 meter	Microcontroller compatible Highly customizable Flexible Weatherproof options
HitLights	LED Strip Light	\$29.99 (per 16.4 feet)	16.4 feet	Remote controlled Dimmable Flexible Waterproof options
Govee	Smart LED Strip Lights	\$25.99 (per 16.4 feet)	16.4 feet	App controlled Wi-Fi/Bluetooth Voice control Flexible

The Adafruit NeoPixel Digital RGB LED Strip stands out as the better option for several compelling reasons. Firstly, it features high-quality WS2812B RGB LEDs, which are known for their vibrant colors and precise control capabilities. These LEDs allow for intricate lighting effects and detailed color animations, making the strip ideal for projects that require high levels of customization and dynamic lighting.

Another significant advantage of the Adafruit NeoPixel strip is its integration with microcontroller platforms such as Arduino and Raspberry Pi. This compatibility enables advanced programming and control, providing users with the flexibility to create complex lighting sequences and responsive lighting systems. Despite being priced at \$24.95 per meter, which might be higher than some alternatives, the investment is justified by the superior performance and extensive features offered.

3.1.9 Motors

Motors are devices developed to transform electrical energy into mechanical energy. This allows movement and operation in various applications. From the refrigerators and washing machines in our household, to the electric motors that power trains and other vehicles, we use motors every day. These motors have become an essential part of modern technology in which they enable automation, improve efficiency and enhance the convenience of our daily life. For this project we are using the motor to be powered by the PCB and will be used to dispense the fish food. This will also be used to help control the amount of pH drops dispensed into the water. The choice for these motors narrows down between the servo motors and stepper motors, each comes with its unique characteristics and uses.

Servo Motors

Servo motors are well-known for their ability to precisely regulate angular or linear motion. This component functions by sending unique signals to the motor indicating how much to turn and in which direction.

This would be a good addition to use for our project to automate both the feeding process and the pH balance. It is known for its control method and precision. It uses a feedback loop with encoders that continuously monitor and modify the position, speed and torque.

This reduces manual effort to ensure the well-being of aquatic life and its environment which makes it a valuable component to our project.

Stepper Motors

Stepper Motors are a type of DC motor that breaks a full rotation into many equal steps. This component moves in discrete steps making them ideal for applications that need precise position and speed control.

Similar with servo motors, this would also be a good addition to use for our project to automate both the feeding process and pH balance. Unlike the servo motors though, it achieves its precise position without the feedback. Another thing for stepper motors is that it moves consistent torque at lower speeds, but it is limited to lower speeds due to its step-by-step movement.

Brushless DC Motors (BLDC)

Due to their adaptability, simplicity, ease of control and the ability to power consumer electronics, DC motors are an essential component that are widely used in various applications. These applications are toys, small appliances like electric toothbrushes and shavers as well as the cooling fans in computers.

This type of motor would be advantageous for our project. It has higher efficiency due because it has no brushes which reduces friction and energy loss. Another reason for this is its precision control using Electronic Speed Controller (ESC) to accurately dispense food and controlling the amount of pH drops.

Synchronous AC Motors

Synchronous AC Motors are known for their precise speed control and high efficiency. These types are in a form of alternating current (AC) in which the shaft rotates at the same frequency as the supply current making them ideal for applications where constant speed is essential.

This type of motor could be beneficial for our project if for precise speed control. We want to maintain the correct dosage of the food and pH dispensers. This motor maintains a constant speed regardless of load variations. Another thing is that due to its high efficiency at steady speeds. This can help with our power consumption making our overall project energy-efficient and cost-effective over time.

Table 3.9: Tech Comparison of Servo and Stepper Motors.

Criteria	Servo Motors	Stepper Motors	Brushless DC Motors (BLDC)	Synchronous AC Motors
Torque	High at low speeds	High, but can miss steps under load	High and consistent	High and consistent
Step Angle	Based on encoder	(200 full steps)	Based on encoder	Synchronized with supply frequency
Speed	600-5000 RPM	100-<1000 RPM	150-10000 RPM	600-3600 RPM
Power Consumption	5 or more Watts	5 or more Watts	Generally High	Generally High

Positioning	Continuous precise adjustments	Discrete steps	Continuous precise adjustments	Continuous precise adjustments
Cost	\$3-\$150	\$3-\$30	\$30-\$100+	\$10-\$100+

For our project, we need motors that are quick, dependable and precise to integrate to our food and pH dispensers. After further evaluation we chose servo motors for our food dispenser and a stepper motor for our pH tablet dispenser as each type of motor fits the needs of each application.

We chose the servo motor for our food dispenser because it was most well suited with its continuous and precise adjustments. Having high torque at low speeds allows for a consistent and precise dispense of food into the tank even with varying loads. With the stepper motor, we found that to be more ideal for our pH dispenser. With the discrete step-based movement, it allows us to have more precise control on the dispensing of individual tablets while being the most cost effective.

Compared to the other two selections, both the servo and stepper motors offer better performance based on their uses for each application. While Brushless DC motors are efficient at high speeds, it's not very suited for low power applications. To include, the Synchronous AC motors also don't suit our application needs due to the lack of precision even with its high reliability and consistency. Through the use of both the servo and stepper motor, we were able to have the precision and reliability needed for our aquarium system while having less energy consumption and costs.

3.1.10 Servo Motor Selection

When choosing the correct servo motor, it is important to consider the specific requirements for our food dispenser. For now, the thing that is the most important or a primary consideration for us is the precision in ounces of the amount of food dispensed. This will not only save us from preventing overfeeding, but also benefit our project by maintaining optimal feeding conditions for the overall aquarium itself.

Table 3.10: Part Comparison of possible Servo Motors.

Manufacturer	Model	Price	Power Consumption	Speed	Feedback System
TowerPro	SG90	~\$3	4.0-7.2 V	0.12 sec/60°	Potentiometer
TowerPro	MG996R	~\$10	4.8 - 7.2 V	0.19 sec/60°	Potentiometer

DSSERVO	DS3218	~\$15	4.8 - 6.8 V	0.16 sec/60°	Potentiometer
---------	--------	-------	-------------	-----------------	---------------

SG90 stands out for us primarily due to its size, precision and power consumption. Despite its size, this time of servo still offers sufficient precision for our components such as the food and pH dispensers where small controlled movements are crucial. Another thing is the cost because it is generally more affordable than the other servo motors in the market available.

3.1.11 Stepper Motor Selection

When choosing the correct stepper motor, it is important to consider the specific requirements for the pH dispenser. For now, the thing that is the most important or a primary consideration for us is the precision of how many tablets are dispensed with each cycle. This will allow us to prevent incorrect pH level maintenance when dispensing the tablets, but also benefit our project by maintaining optimal pH level conditions for the overall aquarium itself.

Table 3.11: Part Comparison of possible Stepper Motors.

Manufacturer	Model	Price	Power Consumption	Speed	Feedback System
Adafruit Industries LLC	NEMA17 Bipolar Hybrid	\$14.56	350mA at 12V	200 steps/rev (1.8°/step)	Open-Loop System
MIKROE	ROHS 28BYJ-48	\$3-5	92mA at 5V	64 steps/rev (5.625°/step)	Open-Loop System
STEPPER ONLINE	NEMA8 Bipolar Microstep	\$20-30	1A at 4V	200 steps/rev (1.8°/step)	Open-Loop System

For our project the ROHS 28BYJ-48 stood out the most due to its energy efficiency and precision at low-speed operations. Along with its affordability ranging at \$3-\$5 per unit, it saves us tremendously compared to the other two motors. With an energy consumption of 92mA at 5V, it allows for prolonged energy-efficiency while being able to have 64 steps per revolution and allowing for precise movements when tasked to dispense a single tablet per couple steps. While it may sacrifice torque and speed, it still meets our application demands where precision is more heavily prioritized.

3.1.12 Connectivity

In order to have real-time data transfer, remote monitoring and control various parameters within the aquatic environment, connectivity is an essential component of modern aquarium monitoring systems. Efficient connection enables the user to monitor environmental changes, fish behavior as well as their health and take immediate appropriate action when needed. This is important for maintaining ideal conditions and guarantees the health of aquatic life.

Another thing to consider for having an efficient connectivity allows for seamless integration of various components needed for the project. This includes the temperature and pH sensors, automated dispensers, air bubbles, LED lights and also the camera. These components are able to connect and communicate to one another with the use of either a mobile app or central control unit which provides an overall view of the aquarium's condition. Remote connectivity also benefits the user to observe and modify necessary settings from any location, which guarantees that the environment remains stable even when the user is not physically present.

The choice of connectivity for our project narrows down between Bluetooth and Wi-fi. These wireless communications are two of the common technologies used to connect devices and transfer data. Each of this technology has its own distinctive benefits and limitations depending on the particular requirements of the project.

Bluetooth

Bluetooth is known for connecting devices in close proximity for communication between devices. The ability to operate in the 2.4 Ghz ISM band and known for its low power consumption makes it perfect for battery-powered devices. This feature not only increases the battery life but also makes it more appropriate for devices and applications that require energy-efficient and seamless connectivity. This makes it a popular choice for straightforward, point-to-point communication between components.

Wi-fi

Wi-Fi in contrast to Bluetooth, it offers a much faster data transfer over greater distances and supports a broader range of devices in a more complex network environment. It offers a dependable connectivity that is suitable for large file transfers, high-speed internet access and streaming applications for it operates both on 2.4 GHz and 5Ghz bands. This connectivity is ideal for wide coverage and quick data throughput.

Below is a table highlighting the difference between the Bluetooth and Wi-Fi connectivity.

Ethernet

Ethernet in contrast to both Bluetooth and Wi-Fi, is widely used to this day because of its reliability, high speed, and cost effectiveness. Unlike Bluetooth and Wi-Fi, which allows the user to do wireless connectivity, ethernet uses a combination of hardware components such as cables and switches. It also supports speed from 10 Mbps to 100 Gbps and beyond, catering the growing demands for data transfer and communication. This type of connectivity makes it a popular choice for many people due to its ease of use, involving simple plug-and-play hardware components, as well as its enhanced security making it ideal for protecting sensitive data and ensures secure communication.

Table 3.12: Comparison of Connectivity Technologies.

Criteria	Bluetooth	Wi-Fi	Ethernet
Range	Short-range (approximately 10 meters)	Longe range (150 feet indoors and 300 feet outdoors)	Up to 100 meters
Data Transfer Rate	3-50 Mbps	Up to 600 Mbps	10 Mbps-10Gps
Power Consumption	0.5-3 Watts	1-5 Watts	0.3-5 Watts
Interference	2.4 GHz ISM (Less interference)	2.4-5 GHz (potential interference)	Minimal (shielded cable)
Security	SSP/AES-128 encryption	WPA2/WPA3	Varies
Cost	\$2-\$10	<\$20	\$2-\$20

We carefully considered this aspect, acknowledging that each connectivity option we compared has its own pros and cons. When choosing the right connectivity, the thing that is the most important or a primary consideration for this is having the user able to communicate anywhere around the world. This capability gives the user the ability to obtain real-time data collection and monitor well-being of aquatic life. Taking these into account, WiFi is our best choice for connectivity. This allows the user to track the overall environmental conditions and gives them the capability to make any changes throughout the tank as necessary as long as they have internet access.

3.1.13 Wi-Fi Board Selection

Wi-Fi is already integrated into the microcontroller unit (MCU). This device is included on for the project and having this existing connectivity capabilities of the MCU simplifies a lot of things. This includes the over design and process, reduces the cost, and gives us a little bit more confidence to ensure that the devices connected to the MCU will be compatible. Overall, taking advantage of the MCU's built-in features helps improve our project's system reliability and performance.

Table 3.13: *Part Comparison of possible Wi-Fi enabled devices.*

Manufacturer	Model	Price	Operating voltage	Op. Temp
Espressif Systems	ESP32-WROOM-32E Series N16R2	\$3-\$5	3.0V - 3.6V	-40°C to +85°C
Texas Instruments (TI)	MSP430	\$2-\$4	1.8V - 3.6V	-40°C to +85°C
Microchip Technology	PIC32MZ	\$8-\$15 USD	2.1V - 3.6V	-40°C to +85°C

ESP32 is our best bet for this considering that it has a dual-core CPU that enables for efficient multitasking and faster performance which is more than enough for our project. It also has an ease of integration where it simplifies the process of adding Wi-Fi capabilities while making it stable and reliable to our project.

3.1.14 Custom PCB & Development Boards

Choosing the type of central processing unit is one of the most important decisions, as it will control all the connected components and handle the data transfer from the sensors. This decision cannot be taken lightly, as selecting the correct MCU will enable the implementation of all the planned smart aquarium features. Additionally, the chosen approach must ensure reliable operation for years, providing peace of mind to the owner and maintaining optimal aquarium conditions for the well-being of the fish. With this in mind, we will now consider the use of a custom PCB versus an off-the-shelf development board.

Custom PCB

We first investigate the use of a custom PCB. This approach comes with several advantages over a development board. A custom PCB offers the opportunity to be engineered only to include essential components, thus reducing points of failure and enhancing its lifespan. This allows the product to be compact, offering the consumers easy installation while also being compatible with different types of aquarium setups. The ability to choose the components offers the opportunity to select high-quality or industrial-grade components, further increasing reliability. Another benefit is that this allows us to target the product's requirements precisely. By doing so, we can target specific tasks, resulting in better performance, unlike with a development board, which targets a broader set of applications. At the same time, this means better optimization of the circuit, reducing unnecessary overhead with improved efficiency, and reducing costs by eliminating unnecessary components.

Whether a PSU or a battery is utilized, power consumption matters to power the smart aquarium. We can optimize consumption and tailor an efficient design by selecting low-power components. This approach would benefit a battery-operated design where the total operation time is at a premium. However, even if we were to go through the PSU route, this would still be beneficial for electricity savings. Ideally, the device will remain on for years, meaning a greener approach would be desirable. In general, choosing a custom PCB route allows full control over the hardware and software, but the downside is that it takes longer to develop since both areas must be built before the custom PCB is ready for use.

Development Boards

On the other hand, we have the option of choosing an off-the-shelf development board. This approach skips the research, design and development steps, and in doing so, it further contributes to faster development. It can be highly beneficial due to its ease of use. For example, some development boards can be plug and play, such as a Raspberry Pi. Note that there are other plug-and-play options. These boards can ship with a preloaded OS on a micro-SD, so they are ready for use after their initial setup.

Development boards tend to have a large community behind them, meaning there's extensive documentation that accelerates development when working with them, from developers contributing to libraries to countless written or video tutorials to better compatibility with various IDEs facilitating coding and debugging. These adhere to standard interfaces and therefore have better support than custom PCBs.

Peripheral support is typically better for development boards as well. They tend to offer GPIO pins, USB ports, HDMI outputs, serial interfaces, etc. This is a notable advantage over a custom PCB that would need to be designed to include

any peripherals as needed. This can be a time-consuming/complex task, which would end up increasing development time. Therefore, the advantage of having built-in peripheral support allows for faster prototyping and reduced development time.

When it comes to its cost, the use of a development board could be beneficial for smaller projects such as this one. They offer a fixed cost so that we can choose a board based on the project's needs and budget. This board will be ready for use, helping avoid design costs, prototyping costs and components outsourcing.

In general, the key advantage of choosing a development board is its ease of use and ability to skip the hardware development phase. In addition, IDEs often offer features or plugins that can seamlessly interact with the hardware. More importantly, they are backed by a large community and abundant documentation. Last but not least, they can be easily repurposed as needed, therefore being a good investment in the long run.

Table 3.14: Tech Comparison of PCB and Dev Boards.

Criteria	Custom PCB	Development Board
Processing Power	Can be chosen to be as powerful as required.	The board can be chosen based on CPU power requirements.
Power Consumption	Can offer reduced power consumption based on design.	The board can be chosen based on power consumption requirements.
Scalability	Design is tailored as needed.	Fixed design.
Ease of Use	Hardware & Software must first be developed.	Ready to use.
Environmental Tolerance	Components can be chosen based on environmental requirements.	Limited board options based on environmental requirements.
Peripherals Support	Must be designed with the required support.	Includes a number of peripherals built-in.
Cost	Cheaper in the long run if limited to required components only.	Fixed cost at the start of a project.
Documentation	Limited by design changes.	Available.

3.1.15 Microcontroller

A microcontroller is a small integrated circuit that is designed for the control of specific tasks and peripherals. In essence, it's a smaller computer that has its own central processing unit (CPU), volatile and non-volatile memory, and input/output interfaces. They are widely used in embedded systems and consumer electronics products such as home appliances and gaming systems. However, they are often powered by a battery, so they are often designed to be low-powered. There are several types of microcontrollers, but among the common ones, we can find 8-bit, 16-bit and 32-bit microcontrollers. Choosing the microcontroller wasn't a decision taken lightly, as it is the brain of the operations of our project.

The main responsibilities of the microcontroller are receiving and transferring data from three sensors, controlling two motors for the dispensers, and managing an LED lighting system. Additionally, a live stream camera is implemented to monitor the tank's activity. It uses UART, one-wire digital, and analog communication protocols, with the main power source being DC. Fortunately, companies provide open-source schematics for microcontrollers built into their development board designs, allowing us to select a design and modify it to suit our project's needs.

To decide which microcontroller board we ended up using to create our custom PCB, there were several factors and project specifications to consider. Primarily, these included computing power to ensure the quick control of the connected peripherals and accurate processing of data, communication protocols such as UART and analog to digital, the number of pins needed, operating voltage, and the need for a compact-sized board. Considering the number of I/O pins needed, we had six peripherals connected for a total of 17 connections, from which we ensured we had six individual data pins. As for the operating voltages, the temperature sensor utilized a range of 0V to 3.3V, while the pH and turbidity sensors required 0 to 5V to provide accurate data readings.

Another important aspect to consider was memory size and total storage. We had to ensure there was enough non-volatile storage for our code. As for memory size, we mostly needed to consider the bit size of the data provided by the sensors to avoid losing accuracy. Since two of our sensors were analogue, there was no selectable resolution (bits) to be considered because the resolution of the sensor measurements was determined by the analog-to-digital converter of the microcontroller in use. However, our temperature sensor utilized a digital protocol with a 9-to-12-bit selectable resolution.

Finally, it was important that we chose a microcontroller design backed by robust and clear documentation. The development was more likely to go faster if the team was able to look into past problems and how they were solved by the community behind a design.

3.1.16 Microcontroller Selection

The following section focuses on the comparison of microcontroller options including their features and specifications. The requirements detailed in the previous section must be satisfied to be able to build the smart aquarium. Towards the end of the section, we expect to identify the most suitable microcontroller that's able to handle the needed tasks as quickly as possible and efficiently.

Texas Instruments: MSP430FR6989

The MSP430FR6989 is a highly integrated microcontroller from the MSP430 family, designed for ultra-low power applications and created by Texas Instruments. This board's most notable features are 16-bit RISC architecture accompanied by a single core 16MHz of CPU frequency, 2KB of FRAM, 128KB of non-volatile memory, 12-bit ADC, 16-bit timer, support for communication protocols such as UART, SPI, and I2C among other features. Its most impressive features are the addition of a screen that can be used for debugging purposes and FRAM or Ferroelectric Random Access Memory, which allows for fast writing speed, high endurance, and low energy consumption. These are RAM features that are considered ideal for data logging, which is an important aspect of our project when the data is being reported by the sensors and then used within the mobile application. With this configuration in mind, we move on to verify if they can satisfy the needs of our project.

First, we investigate the speed of the CPU, amount of RAM, and storage. While 16MHz would work well for basic tasks such as reading data off the sensors, it's a single-core processor running at a low frequency, which can result in a bottleneck when it comes to multiple queued requests by the app, working with the database and if the stretch goals were implemented adding a camera into the mix would be out of the question. The same problem is very likely to occur with the memory since 2KB RAM and 128 KB of storage is an extremely low spec, considering we are running a local serverless database, handling multiple inputs and outputs and storing code that handles all these tasks. On the bright side, this configuration supports a selectable resolution of up to 12-bit, which covers the sensor's requirements.

We now explore the required communication protocols, pin number and voltages. The project requires the communication protocols analogue to digital and UART, which the MSP430FR6989 satisfies. As for pin number, there's the necessary amount of PWD, DATA, VCC, and GND pins to check off the requirements for the motors, sensors, LEDs, and stretch goals if implemented. The project sensors require a range of voltages from 3V up to 5V, which the board is perfectly capable of handling while also having more than enough voltage headers.

We finally consider the documentation backing up the board. Ti provides the schematic for the board, which is highly important if we want to create our

custom PCB to cover the project's needs. There's also plenty of official documentation on the board's web page ranging from user's guide manuals to more technical literature. The community is still very active since it's a very popular board for education and learning as well as open-source projects. Additionally, there's many resources found on its forums.

With all of these considerations in mind, even though the board is still officially supported by TI and covers most of the project's needs, the team has decided to explore other microcontroller options. The CPU, RAM and storage configuration could potentially lead to a bottleneck during development based on the number of operations it must handle. The board design also does not support a Wi-Fi connection which is essential for the product's design.

Microchip Technology: Curiosity PIC32MX470

The Curiosity PIC32MX470 is among the most popular boards of the PIC32 family of microcontrollers designed by Microchip Technologies. It's a versatile board suited for various applications, from simple projects to more complex development. The microcontroller's configuration is a single-core 32-bit CPU running at 120 MHz paired with 128 KB RAM and 512KB of non-volatile flash memory with on-chip temperature measurement capability. The Curiosity PIC32MX470 also supports several communication protocols such as UART, SPI, I2C and analogue to digital with an ADC module supporting 10-bit resolution. Other features include a built-in Bluetooth 4.2 module, several 16-bit and 32-bit timers/counters, an X32 header for audio modules and two mikroBUS sockets, among other general development board features.

The addition of the mikroBUS sockets is one of the board's most notable features and highly beneficial to our project since it allows for the use of MikroElektronika Click boards addons. These boards provide several easy to add functionalities to the custom PCB. For example, sensors, actuators, communication modules such as Wi-Fi and Bluetooth, displays, and more making the PIC32MX470 a highly customizable board.

Now, we investigate if this microcontroller/board design is suitable for our project's requirements, starting off by looking into the CPU, RAM and storage. The PIC32MX470 sports a 7.5 times faster CPU than the previously discussed microcontroller. This would absolutely address our concern of needing to process peripheral and app requests quickly. However, this comes with the downside of remaining at a single core, which again could potentially be a bottleneck under peak performance scenarios. We've now seen a considerable increase in RAM memory, but unfortunately, the non-volatile memory that this configuration is paired with is still too low for the project's needs. This can be solved by utilizing external storage, such as a micro-SD, but the team prefers to keep this as a backup option for several reasons. Performance is worse than that of utilizing the MCU's flash memory, and the reliability of a micro-SD card is lower than that of flash memory. This last point is highly important since micro-SDs can suffer from

tear and wear much faster, leading to data corruption. Additionally, it would be bad practice to create a consumer product with such a flaw of having removable code or code that can be easily tampered with.

The PIC32MX470 comes with up to five UART modules, two 4-wire SPI models, two I2C modules and a 10-bit ADC module fulfilling the communication protocols requirements of the project. However, the ADC module is another shortcoming of this board since we would lose precision with our temperature sensor ranging from 8-bit to 12-bit resolution. The microcontroller board design also includes the necessary voltage headers ranging from 3.3V to 5V, which satisfies the project's peripherals requirements.

Documentation-wise, this microcontroller benefits from a strong community presence. Microchip's product web page contains all the necessary guides to understand the hardware and how to get started. For example, it provides user guides, data sheets, software libraries, and technical support, among other resources. There's also a space for the user community full of discussions about projects from developers, engineers, and hobbyists, which are packed with questions, examples, and valuable insight related to Microchip's products.

Overall, this microcontroller would be able to handle every task thrown at it. This controller has the ability to easily integrate extra functionality through addons, fulfills voltage and communication protocol requirements, and is backed by an active and strong community. However, it has a few notable shortcomings. The processor remains at a single core, which could lead to bottlenecks and affect the performance of the final product. There's low built-in storage, which could force the team to rely on external storage, putting the integrity of the entire device at risk by exposing its code and files to the consumer. Finally, its ADC module would reduce the precision of our temperature sensor. For these reasons, the team has decided to move on from this microcontroller option and investigate a multicore and higher storage solution.

Espressif: ESP32-WROOM-32E- N16R2

The ESP32-WROOM-32 E series N16R2 belongs to the ESP32 family of microcontrollers created by Espressif. It's designed for IoT applications, and it offers robust connectivity options and extensive peripheral support in a compact and cost-effective package. The microcontroller configuration is built around a 32-bit dual-core LX6 microprocessor at a frequency of 240MHz, a 40 MHz crystal oscillator, with 520 KB SRAM, 2MB of PSRAM, 448 KB ROM and 16MB of non-volatile SPI flash memory. It also supports several communication protocols such as UART, SPI, and I2C, among others and features an ADC with a selectable resolution of 12-bit. The board includes Bluetooth 4.2 and Wi-Fi capabilities featuring Wi-Fi 5 or the 802.11b/g/n standard on a 2.4GHz band reaching speeds up to 150Mbps with a built-in PCB antenna. Other features include IR, a pulse counter, a capacitive touch sensor and an SD card interface.

Now, let's delve into whether this microcontroller is well-suited for the project at hand. The configuration offered by this microcontroller sports a dual-core processor now at double the clock rate speed of the previously discussed PIC32 microcontroller. With the multicore requirement checked and much faster speeds, we can confidently say this microcontroller would be able to handle all the smart aquarium tasks quickly while also running a local database that interacts with the mobile application. As for RAM and flash storage, we are now presented with a much higher amount than for previously discussed microcontrollers, eliminating concerns about having enough for storing and running our code. This board design also supports all the necessary communication protocols, and thanks to the 12-bit ADC, we wouldn't have any precision issues with the sensors. There are also the necessary voltage headers ranging from 3.3V to 5V, covering our project's peripherals requirements.

As for documentation for the ESP32 family of boards and microcontrollers, developers consider it quite comprehensive and supportive for a wide variety of hobbyist projects and commercial products. Espressif provides extensive official documentation that includes datasheets, technical reference manuals, SDK documentation and hardware design guidelines. Due to the popularity of the ESP32, there are also numerous tutorials and forums covering all sorts of projects from developers, engineers and hobbyists. There are also various forums and platforms, such as GitHub, providing code examples. The maker, Espressif, is also very active in updating the available official documentation to stay up to date with SDK changes, bug fixes and the latest features.

In general, this microcontroller is perfectly capable of quickly handling the smart aquarium's requests as well as the prompts produced by the mobile application, thanks to its higher frequency than previously considered microcontrollers and its multi-core processor. It also sports higher non-volatile storage and more RAM, ensuring great overall performance. Bitwise, the ADC module is fully capable of handling the sensors with full precision, and one of the most important features is its capability of Wi-Fi. The final product connects to Wi-Fi to interact with the mobile application and access a camera live stream. For these reasons, the Espressif ESP32-WROOM-32E-N16R2 is the microcontroller of choice to develop Azul Tank.

Table 3.15: Part Comparison of possible Microcontrollers.

Criteria	MSP430FR6989	Curiosity PIC32MX470	ESP32WROOM32 E- N16R2
Architecture	16-bit	32-bit	32-bit
CPU Frequency	16-MHz	120MHz	240 MHz
RAM Size	2KB FRAM	128KB RAM	520 KB SRAM + 16 KB SRAM in RTC + s 2 MB PSRAM

Memory Size	128KB of nonvolatile FRAM	512KB Flash	16 MB
GPIO Pins	83	43	38
Low-Power Mode	Yes	Yes	Yes
UART	Yes	Yes	Yes
SPI	Yes	Yes	Yes
I2C	Yes	Yes	Yes
ADC Supported Resolution	12-bit	10-bit	12-bit
Wi-Fi	No	Via Add-ons	Yes
Operating Temperature	-40°C to +85°C	-40°C to +105°C	-40 to +85 °C
Operating Voltage	1.8 to 3.6 V	2.3 to 3.6 V	3.0 to 3.6 V
Package	LQFP-80	QFN-64	SMD-38
Cost	\$20.00	\$44.26	\$16.00

For our smart fish tank aquarium project, we choose to create our custom PCB based on Espressif's ESP32-WROOM-32E Series N16R2. This microcontroller offers several low-power modes; however, this doesn't pertain to our project since it means Wi-Fi is turned off when any of these modes are active. That said, there are several benefits to choosing this microcontroller design. It is important to find a compact design that fits within the enclosure we have in mind. Since this is an attachment to home aquariums, we want to avoid a bulky design so that it can be easily installed and look like it belongs to the tank's setup. The ESP32 has 38 GPIO pins, which is more than enough while remaining under more than half the amount present on Ti's board and slightly above Microchip's boards. This ensures that our compact design requirement is met, and any board replacements are much simpler and cheaper.

Moving onto key features, the ESP32 is equipped with a dual-core processor and sufficient memory to handle all the tasks required by the smart aquarium's design with ease. It also comes with a Wi-Fi module, which is an essential component for the aquarium since it is set up and controlled by a mobile application. All the communication protocols needed by the selected components are supported, and we don't lose the precision of the components thanks to the ESP32 ADC resolution. The ESP32 can be programmed in C and C++, and while the team wasn't as familiar with C++, the features and design were backed by excellent documentation and a very active community, which made development perfectly doable.

With these points in mind, the team concluded that Espressif's ESP32-WROOM-32-E Series N16R2 was the most suitable microcontroller for the project's needs since it included all the needed features at a low cost and without going overboard with unnecessary hardware.

3.1.17 AC - DC Converters

When choosing the proper topology for AC - DC converters, there is a lot to consider through a large variety of advantages and disadvantages between each. It's important for us to choose the proper topology because it can cause significantly cause impact to the efficiency and performance of our device. Given the scope of work and what requirements we have explained to allow our device to work to its full capabilities, we have laid out different topologies we have considered to fit what we need to make our device function at its peak. Below are the thorough examinations of the strengths, weaknesses, and applications of each topology and the key considerations that we as a team have decided that will benefit our design the most.

Flyback Converter

Flyback converters are a type of isolated buck-boost (step-up or down) AC-DC converter that has the inductor replaced with a transformer. It converts the AC input into a smooth DC voltage through rectification and filtering. When the power switch (like a MOSFET) is on, the DC voltage charges the transformers magnetic field but when off the transformers secondary winding has the energy transferred to it to once again rectify and filter the energy for a stable DC output. The transformer provides a safety isolation and a PWM controller that adjusts the switch's duty cycle. Flybacks are valuable with the efficiency and versatility this converter has due to its ability to handle a wide range of load conditions and inputs for a large diversity of applications. The feedback mechanism introduced in the converter allows for precise voltage regulation to ensure consistent performance even if there are varying load demands or fluctuations in the input voltage. Considerably more safe for electronic devices that are more sensitive due to the electrically isolated output from the input. Although the disadvantages that the flyback converter has is voltage stress, more noise , more complex design, lower efficiency at higher frequencies, and cost.

Push-Pull Forward Converter

The push-pull converter differs from the typical buck, boost, or buck-boost converters as it is a type of isolated DC - DC converter with an added rectifier stage prior to being fed into the push-pull converter. This converter rectified and filtered the AC input to generate the DC voltage that is then driven through the primary winding of the transformer. But unlike the flyback converter, this converter uses two power switches (either MOSFETs or BJTs) that creates an alternating magnetic field that transfers and stores the energy in the secondary winding during each cycle.

Within the secondary winding is where the energy is further filtered and rectified to provide a smooth DC output.. As a highly efficient converter, the symmetrical operation of two power switches allows for minimal stress on the components and enhances the durability and longevity of the controllers. Similar to the flyback converter, it also offers electrical isolation and a feedback control system to adjust the switching to ensure a stable output voltage and safety capabilities to protect sensitive electronics from the input voltage spikes and electrical noise. Like the Flyback, the disadvantages that the Push-pull converter has is the increased design complexity, switching losses due to needing more than one switch, complex control and drive requirements, lower efficiency under certain conditions, and cost.

Quasi - Resonant Flyback

The quasi-resonant flyback converter is an upgraded version of the traditional flyback converter, designed to reduce switching losses and lower electromagnetic interference (EMI). It operates in a quasi-resonant mode, utilizing zero-voltage or zero-current switching techniques to enhance efficiency and reduce EMI. This makes it ideal for applications where space and noise are critical factors, such as portable electronics or small power adapters. However, the increased control complexity of the quasi-resonant flyback converter requires meticulous design and implementation. Despite this complexity, its ability to offer high efficiency and low EMI in a compact form factor makes it a preferred choice for modern power supply applications.

Table 3.16: Tech Comparison of AC-DC Converters.

Criteria	Flyback (Step-Up or Step-Down)	Push-Pull Forward (Step-Up or Step-Down)	Quasi-Resonant Flyback Converter (Step-Down Buck)
Efficiency	Medium	High	High
Heat Generated	Medium - High	Low - Medium	Low - Medium
Low Load Efficiency	High	Low	Medium - High
Noise Output (Ripple)	High	Low	High
Design Flexibility	Narrow Input Voltage Range	Wide Input Voltage Range	Medium
Complexity	Low	High	Low - Medium
Cost:	High	Medium	Medium

As a group, we have chosen Flyback converters as they are ideal for small-scale projects that use AC to DC conversion. Due to simplicity, efficiency, and cost-effectiveness are suited for our low to medium-power applications because the design only needs a single compact component to save us space and reduce the design complexity. This design has an electrical isolation between the output and input and can handle various input voltages. With such adaptability and our cost, size, and simplicity constraints, the flyback was the right choice for our project.

3.1.18 AC - DC Converter Selection

As we don't have an external battery or power, having the AC to DC converter wall plug in allows for our devices that use DC power to receive the necessary voltage to work properly. At the same time, having this converter allows it to have a more stable and consistent DC output voltage and puts less stress on the other components that further filter and step-down or step-up the voltage to meet the different requirements of the overall system.

Desvorry B08QCBRYPT

Designed to be a reliable 60W AC-to-DC adapter that is able to convert a wide input voltage of 100-200V AC into a stable 12V DC output. Prioritizing safety for users, it has built in protections against overcurrent, overvoltage, short circuits, and overheating. With so many forms of safety measures, it allows for a smooth and consistent power delivery for devices like routers or LED strips. Also with its compact design and affordability, it is able to cater for a wide variety of applications where reliability and safety are very essential to sensitive equipment.

Alitove ALT-1205

Similar to the previous device this is also a durable AC-to-DC converter designed towards efficient operation for a variety of applications. Providing a 12V output with up to 60W of power that is also equipped with quite a few safety features. These protections include voltage surges, short circuits, and thermal overloads. One of the main features of the ALT-1205 is being able to maintain a consistent voltage output that is essential for more sensitive devices. Like the Desvorry brand, it has a similar input range at 100-240V AC allowing it to be another versatility choice.

LitStar GMFINE

With the LitStar GMFINE, it differs from the previous converters as it only outputs 24W of power. This is an AC-to-DC adapter that is made for more low-powered devices that doesn't require as much current. A key feature of this converter is that it has an automatic current adjustment range from 50mA to 2A, which allows for accommodation of a variety of devices. Like the previous two converters, it has an array of safety features that protects against short circuits, overcharging, and even overloading. Having the adaptability as it does with the current

adjustment, it makes it a practical and cost-effective choice for low power requirement equipment.

Table 3.17: Part Comparison of possible AC-DC Converters.

Model	B08QCBRYPT	ALT-1205	GMFINE
Brand	Desvorry	Alitove	LitStar
Input Voltage (Min-Max)	100-200V AC	100-240V AC	100-240V AC
Output Current (Amps)	5A	5A	50mA-2A
Output Voltage	12V	12V	12V
Operating Temperature Range (°C)	0°C to 40°C	0°C to 40°C	0°C to 40°C
Type	Wall Plug-in	Wall Plug-in	Wall Plug-in
Cost (Per Unit)	\$10.99	\$11.99	\$11.99

Overall, we chose to go with the LitStar GMFINE as it provided the features that we desired for our project. Since we aren't using any external battery or other power source, it was important to choose a proper converter that we can trust to handle high frequencies and be able to have a quick response time. Lastly, just having the flexibility, and precise control for efficient performance allows the other parts we have chosen to have an easier time handling the output voltage and current provided from this converter.

3.1.19 Voltage Regulation

The use of voltage regulation is a must due to the various voltage requirements required of many different components in the systems involved. To ensure the reliability of the operation of the devices and systems within the design, a stable and consistent voltage supply is crucial for the longevity and protection of the components.

Voltage regulators are a key component and have two primary types which are linear and switching regulators. Both types have their own individual advantages, applications, and characteristics which may make one or the other stand out more depending on the design specifications. Linear regulators have more simplicity in design, low cost and noise but in turn have lower efficiency if there's

a large disparity between the input and output voltage. While switching regulators, which has better efficiency and reliability at the cost of design simplicity and more noise.

Linear Regulator

As a step-down converter, linear regulators will always have an output voltage that is lower than the input. This means that there is a minimum voltage difference between the V_{in} and V_{out} needed to allow the regulator to work properly. There are two types of linear regulators, series and shunt. Series regulators are placed in a series between a load and power supply, while shunt regulators are placed in parallel between a load and power supply. A common linear series regulator is low-dropout (LDO) and they use a transistor that is controlled by a negative-feedback circuit that produces a stable output voltage.

Unlike the switching regulator, linear regulators aren't able to regulate the desired output voltage if the V_{out} is greater than V_{in} minus the $V_{dropout}$ and any excess power is dissipated in the pass transistor as heat. With a simple design, low noise, fast transient response and cost effectiveness it's worth considering for low power or noise sensitive applications.

Switching Regulator

A switching regulator is different than the linear as it uses a switching element to transform the incoming power supply to a pulsed voltage. By turning on a switch (MOSFET or FET), power is then supplied until the voltage that is desired is achieved. The energy stored in the inductors and capacitors that was supplied is delivered to the output.

Unlike the linear regulator, the way the switching regulator is set up allows for efficient voltage supply while keeping heat generation at a minimum for applications that require more out of the power supply. At the same time, it provides flexibility for designs as the switching regulator is offered in four common different output formats which are Buck, Boost, Buck-boost, and Inverting. The variety of formats allow for a range of efficiency for the output voltage if it needs to be stepped up or down for a design.

With the versatility and efficiency it's important to note that the switching regulator's downsides include the complexity of the design, slow transient response, louder noise, and cost. But should be highly considered for high current and power applications, wide input voltage range, flexibility, and or power-sensitive situations.

Table 3.18: Tech Comparison of Linear and Switching Regulators.

Criteria	Linear Regulator	Switching Regulator		
		Buck (Step-Down)	Boost (Step-Up)	Buck-Boost (Step-Up or Down)
Flexibility (Types)	Buck	Buck (Step-Down)	Boost (Step-Up)	Buck-Boost (Step-Up or Down)
Efficiency	Low - Medium	High	High	High
Heat Generated	Medium - High	Low	Low	Low
Complexity	Low	Medium	Medium - High	High
Noise Output (Ripple)	Low	Medium	High	Medium
Cost	Low	Medium	Medium - High	High

As a group we have chosen the Buck-type switching regulator because they are considered top-tier voltage regulators for multiple reasons. They are highly efficient, often exceeding 90%, by converting excess input voltage into current instead of heat, unlike linear regulators. This high efficiency is essential for battery-powered devices, where energy conservation is crucial. Buck converters are also very versatile, effectively stepping down voltage with minimal power loss, making them suitable for a broad range of applications, from small portable electronics to large industrial systems.

Their capability to handle wide input voltage ranges while providing stable output voltage ensures consistent performance under varying conditions. Modern buck regulators also feature integrated protections like overcurrent protection, thermal shutdown, and soft start, which enhance their reliability and simplify their use in complex electronic designs. With their outstanding efficiency, versatility, and advanced features, buck-type switching regulators are the best choice for efficient and reliable voltage regulation.

3.1.20 Switching Regulator Selection

With the inclusion of a DC to DC converter paired with the AC to DC converter a voltage regulator might be unnecessary but it's included for the added protection and fine tuning of our design. By allowing the DC to DC converter the freedom to step up or down the voltage, the voltage regulator helps to fine tune the output voltage to a more precise and stable value that helps to filter noise and maintain a constant output despite variations in input voltage or load conditions.

LMR51430

As a versatile simple switcher synchronous buck converter that's designed for industrial applications, this regulator is capable of delivering up to a 3A of current and has an input range of 4.5V to 36V. It is able to support 500 kHz and 1.1 MHz switching frequencies that enable the use of small inductors for a compact design. This device is available in both the PFM and FPWM versions and it offers high efficiency at light loads and consistent at frequencies with minimal output voltage ripple across the loads. The device also includes a built-in soft start and compensation circuits that minimize the need for external components while also offering protective features such as cycle by cycle current limit, short circuit protection, and thermal shutdown to prevent excessive power dissipation.

TPS54528DDAR

As a versatile synchronous buck converter, this operates using an adaptive on time CAP2 mode control. This was designed as a cost-effective solution that has low component requirements and a low standby current. With its main control loop the D-CAP2 it delivers a fast transient response without the need for any external compensation. Along with the adaptive on-time control, this ensures a seamless transition between the Eco-mode and heavy loads for high efficiency across a wide range of conditions. The TPS54228 is able to take an input voltage range of 4.5V to 18V and is able to output 0.76V to 6V with an adjustable soft start feature. Lastly, with its compatibility with low ESR capacitors makes it a viable option for efficiency with reliable temperature ranges at -40°C to 85°C .

LM22676MR-ADJ/NOPB

This device is a step-down switching regulator design towards high voltage applications and integrates all essential functions that require only a few external components. Featuring a 42V N-channel MOSFET switch that is able to supply up to a 3A load current, it is able to boast a staggering 90% efficiency with strong line and load regulations. Like the other two devices, the LM22676 is able to offer a wide range between the input and output voltages and includes an internal loop compensation that eliminates the need for external calculations. With the regulator having both a fixed 5V and adjustable output versions, it is able to operate at a 500kHz switching frequency for compact external components. With a built-in 500 μs soft-start, and protections for thermal shutdown and current limiting guards for overload, it provides all the necessary applications.

Table 3.19: Part Comparison of possible Voltage Regulators.

Model	LMR51430	TPS54528DDAR	LM22676MR-ADJ /NOPB
Brand	Texas Instruments	Texas Instruments	Texas Instruments
Input Voltage (Min-Max)	4.5V - 36V	4.5V - 18V	4.5V - 42V

Output Voltage (Min-Max)	0.6V - 34.2V	0.76V - 6V	1.3V - 37V
Switching Frequency (kHz)	500 kHz - 1.1 MHz	650 kHz	500 kHz
Output Current	3A	5A	3A
Type	Buck	Buck	Buck
Cost (Per Unit)	\$1.51	\$0.746	\$5.59

Overall, we chose to use the TPS54528 as our voltage regulator as it provided a versatile set of features that we desired to not only increase efficiency. By having the capability to support an input range of 4.5V to 18V, it allows for high efficiency at light loads and is consistent at frequencies with minimal output voltage.

It includes a soft start that minimizes the need for external components and offers necessary protective features to keep the longevity of our devices. With this component being manufactured by Texas instruments, it allows us to have a reliable data sheet to go off of and documentation to help properly implement this device into our design. Cost wise, it is the cheapest of the three and is more than capable of meeting the requirements and specifications of the design needing 5V for our smaller components.

3.2 Software Research

Software research and considerations are important the same way as hardware investigation. This way we can guarantee the highest possible performance and ensure the compatibility with microcontroller units (MCUs), sensors, dispensers and other essential components required for the project. This research helped us understand and ensure that the software chosen communicates well and works together efficiently.

Effective and efficient software research also allows the user to create reliable systems and develop software that performs well, meets user needs, and possibly supports future technology advancements.

3.2.1 PCB Design Software

Printed Circuit Board (PCB) not only serves as a base where each electronic component is mounted and connected but also shows how well the final product performs, how easily it can be manufactured and how reliable it is.

When it comes to designing a PCB, a carefully thought-out plan for the layout ensures that electrical signals flow efficiently between components. This includes setting up the components in a way that minimizes interference and maximizes

its performance. This way it prevents signal loss and electromagnetic interference.

Another thing that we need to consider alongside designing the PCB is finding the software for it. Selecting the right software for designing a project's printed circuit boards (PCBs) play a crucial role in determining the efficiency, precision, and ultimate success of a project. Choosing the correct software can greatly increase production output and efficiency during design by providing tools that make it easier to manage projects. The software itself also needs to be user-friendly to reduce learning curve and errors so the user can easily understand how to use it properly.

Altium Designer

"Altium Designer is a PCB and electronic design automation software package for printed circuit boards."

Developed by Altium Limited, It integrates various design elements into one cohesive platform, allowing designers to tackle complex projects with greater efficiency and accuracy. This integration minimizes errors and facilitates a smooth workflow through different design stages.

One of Altium Designer's key features is its advanced suite of tools, which includes interactive routing, auto-routing, and more. These tools are essential for managing high-speed design requirements and ensuring optimal PCB performance. The software's routing capabilities help designers maintain signal integrity, reduce electromagnetic interference, and optimize layouts for both analog and digital circuits.

Altium Designer also provides extensive simulation and analysis tools, enabling designers to test and validate circuits under different conditions before finalizing their designs. This early detection of potential issues saves time and resources. The software supports team collaboration with features for version control, project management, and real-time communication.

The user-friendly interface and comprehensive component library further enhance the design experience. Designers have access to a vast database of verified components, making it easier to select up-to-date parts. Additionally, Altium Designer integrates well with other engineering tools and platforms, promoting a seamless workflow.

Overall, Altium Designer is a powerful and versatile PCB design tool that streamlines the design process, boosts productivity, and ensures the creation of high-quality electronic circuits. Its extensive range of tools and features makes it a valuable asset for both beginners and seasoned designers.

EAGLE (Easily Applicable Graphical Layout Editor)

“EAGLE (Easily Applicable Graphical Layout Editor) is an electronic design automation (EDA) software that lets printed circuit board (PCB) designers seamlessly connect schematic diagrams, component placements, PCB routing, and comprehensive library content.”

Initially developed by CadSoft Computer GmbH but now owned by Autodesk, the software itself provides a robust suite of tools for schematic capture and PCB layout that makes it an excellent choice for designing and working on printed circuit boards. By including advanced features that cater to high-speed design needs, EAGLE ensures the PCB's optimal performance. Overall, the software also offers sophisticated interactive routing and auto-routing tools which are essential for maintaining signal integrity and reducing electromagnetic interference. These functionalities ensure that the finished product adheres to the highest standards of performance and reliability.

Another thing to consider for this software is that it is part of the Autodesk family. This means EAGLE can integrate easily with other Autodesk tools such as Fusion 360. This enhances the software's capabilities to create better and more complete electronic design products. EAGLE's user-friendly interface and extensive features cater to both novice and experienced designers. Additionally, the software supports team collaboration through tools like version control and project management, promoting efficient and seamless workflows.

Finally, EAGLE has a major benefit and that is the extensive component library that is installed right into the software system. By having a variety of ready-to-use parts, this helps to accelerate the design phase and ensure that anyone designing has access to the latest components in real time to truly encompass the user-friendly experience.

EasyEDA

EasyEDA is a cloud-based EDA tool designed to simplify the process of electronic design and PCB layout for users ranging from beginners to experienced professionals. This software is particularly popular among hobbyists and small-scale manufacturers due to its intuitive interface and seamless integration with fabrication services.

One of the most notable features of EasyEDA is its comprehensive online library of components, which includes symbols, footprints, and 3D models. This ensures that users can quickly find and use components without the need for manual creation, significantly speeding up the design process.

EasyEDA also excels in its collaborative features. Being cloud-based, it allows multiple users to work on the same project in real time, making it ideal for team-based projects. Additionally, its integration with fabrication services, such as JLCPCB, streamlines the transition from design to manufacturing, reducing time and potential errors in the process.

Another strength of EasyEDA is its cross-platform accessibility. Users can access their projects through a web browser or download the desktop version for offline use, ensuring flexibility and convenience.

While not as advanced as some professional-grade tools, EasyEDA provides an excellent balance of features, usability, and affordability, making it a go-to solution for many in the PCB design community.

Table 3.20: Tech Comparison of PCB Design Software.

Criteria	Altium Designer	EAGLE	EasyEDA
Simulation Tool	Altium's Mixed Signal Circuit Simulator	N/A Supports external simulation software such as LTspice	Built-in simulator for basic circuit testing
Component Libraries	Up to date with real-time update	Regularly updated and can be expanded with custom libraries	Extensive online library, updated regularly
OS Compatibility	Windows, macOS	Windows, macOS, Linux	Cloud-based, desktop app for Windows/mac OS
Memory Usage	8-16 GB RAM 2-4 GB VRAM	4-8 GB RAM 1-2 GB VRAM	Minimal (runs in browser or lightweight app)
User-Friendly	Yes	Yes	Yes
Cost	Free \$355/month	Free \$680/year	Free (some advanced features may cost extra)

When selecting PCB design software, we prioritized familiarity with the tool to streamline our workflow and avoid the steep learning curve of adopting a new platform. This decision not only saves us valuable time but also reduces the frustration that can come with mastering unfamiliar design software.

EasyEDA stood out as the ideal choice for our project due to its intuitive interface and seamless integration with our design needs. While professional-grade tools like Altium Designer offer extensive features, EasyEDA provides a well-rounded solution that aligns perfectly with our requirements. Its robust component libraries, regularly updated and easily accessible, ensure we have access to the necessary components for our designs without added complexity. Additionally, its cloud-based nature allows for flexibility and collaboration, which is crucial for managing and sharing our work efficiently.

In summary, EasyEDA offers the right balance of usability, functionality, and accessibility, making it the perfect fit for our PCB design goals while maintaining high productivity and quality in our work.

3.2.2 Development Technologies

Application software development involves multiple components, each playing a crucial role in creating a functional and seamless experience. These components can be divided into the front end, which handles the user interface and user experience, and the back end, responsible for the background logic and data processing. By carefully selecting the right technologies, we can ensure the creation of a modern, cross-platform solution. Below, we will explore the technologies we have considered that are key to the success of our project.

Flutter

Flutter is an open-source UI software development kit. It was created by Google and released in 2017. It allows the development of cross-platform applications for a variety of platforms of operating systems such as web, Fuchsia, Android, iOS, Linux, macOS and Windows. From a developer's perspective, Flutter is easier to learn than React Native. It utilizes Dart as a programming language, also created by Google, and while the team isn't familiar with it, it's easier to pick up than JavaScript. The language's structure is also closer to what native application development looks like.

Flutter's Command Line Interface (CLI) comes with a tool called Flutter Doctor. This tool helps developers set up their chosen IDE for iOS or Android development. When paired with CLI it becomes among the smoothest setup experiences compared to other stack technologies.

Moving onto performance, Flutter utilizes Impeller which offers runtime rendering. It optimizes GPU usage and UI performance and efficiency. With the use of powerful API's such as Metal and Vulkan, it offers more efficient rendering, lower power consumption and better framerates.

Flutter's documentation provides easy navigation and is meticulously crafted to cater to developers from different backgrounds. It comes with tutorials that offer hands-on experience. However, this comes with the downside of having a low number of active Dart developers compared to other communities, such as the JavaScript community. This can lead to problems when development problems and questions arise, which led our team to move away from this option.

React Native

On the other hand, React Native is a mobile development framework created by Facebook/Meta, released in 2015. It's another potential choice when developing mobile, web, and desktop applications, also allowing cross-platform development

on iOS and Android. React Native transforms JavaScript components or the "user interface design blocks" into native ones.

React Native's components approach can be highly beneficial when compared to Flutter, but in the long run, it can also become problematic. For example, a React Native application will be able to keep up with the latest native components design. The downside is that as this happens automatically, you can lose part of the application design. To keep a current design, this feature must be turned off.

React Native's powerful platform comes at a price. In general, developers consider the learning curve somewhat steeper. Although JavaScript is a much more popular programming language than Dart, that doesn't mean it's easier to work with.

For example, dealing with classes within classes adds to the complexity and nuances of React Native. This is an important point to consider when choosing our stack since it can lead to hours upon hours of troubleshooting or solving problems throughout development.

React Native's upper hand is called Expo Go. This tool offers a much easier way to design, build and deploy applications, including web applications. Most importantly, it enables developers to view and interact with their apps as they build them directly on their phones. This accelerates development since it removes the need to go through Android Studio or xCode, Apple's iOS development environment.

React Native's performance can be a little slower than Flutter's due to its JavaScript bridge needed for the interaction between JavaScript, the native user interface, and device components. However, this doesn't necessarily mean it's a worse approach. React Native might not be able to compile native code, but unlike other similar technologies, it can produce native-looking user interfaces.

Documentation isn't as meticulously organized or clear as Flutter's counterpart, however it is deemed more user friendly since it encompasses best practices and useful frequently asked questions. React Native on the other hand, has been around for much longer, and therefore, there are many more tutorials and topics being discussed on useful sites such as Stack Overflow and GitHub. This leads to a large and more mature community backing this stack. An important point to consider when it comes to trying to solve unexpected complex development problems.

Table 3.21: Tech Comparison between Flutter and React Native

Criteria	Flutter	React Native
Learning Curve	Easier Difficulty	Higher Difficulty
Development Environment	iOS:Limited on Windows with full support on MacOS Android: Works on both Windows and MacOS	iOS:Limited on Windows with full support on MacOS Android: Works on both Windows and MacOS
Key Development tools	Flutter Doctor	Expo Go
Automatic Component Updates	No	Yes
Performance	Faster	Limited by JavaScript bridge
Operating Systems	Windows & MacOS	Windows & MacOS
Cross Platform Development	Yes	Yes
Documentation	Easier to understand for newcomers.	More complex, but robust backed by a large community.

For development framework, the team has chose to develop the application via React Native. While its learning curve is somewhat more difficult than its counterpart, Flutter, it offers near native-looking user interfaces, which brings us closer to our goal of achieving a modern and easy-to-use UI. The use of Expo Go accelerates development since we can view and interact with the application as it's being built. Additionally, the community behind React Native is more mature. Therefore, documentation is widely available; there are many tutorials on how to build an application via React Native and code examples on forums and GitHub. Finally, employers are more interested in a candidate who can code in React Native instead of Flutter.

3.2.3 Database

A database is a structured collection of data usually stored either locally on a computer or on the cloud. They are designed to efficiently manage, retrieve and store data allowing easy access to it as needed. A common scenario of a need for a database is an application with usernames and password of its users.

Our project's design required the use of database to handle the data reported by the sensors in place which will then be sent over to a mobile application. Below we explore the type of databases under consideration.

SQL Database

SQL uses a relational database management system. This approach is suitable for structured data and requires the user to predefine a schema before working with data and inserting it into a table. This restriction means all the data must follow the same structure. Working with an SQL database means a safer beginner option since SQL is known to be one of the most versatile and widely used database structures backed by a large community and abundant documentation. This would also offer us the option of utilizing SQLite, a serverless approach that works as an embedded database within applications. When considering our options, it's a very tempting route since it would avoid any server headaches and with time being at a premium, it would make sense to choose it.

NoSQL Database

On the other hand, we have NoSQL databases. These use a non-relational database system, which means a dynamic and more flexible schema for unstructured data. Documents can be created without a defined structure, and each document can have its own structure. Thanks to this approach, data can be stored in many ways, such as document-oriented, column-oriented, graph-based, or organized as a key-value store. Choosing a NoSQL database would allow us to work on the data without worrying about disrupting the whole system if our data structures were to change at some point during development. MongoDB is known to work well with React Native, which is one of our possible technologies to incorporate in the development of the mobile app, therefore making a NoSQL database a viable option.

In conclusion, the project made use of an SQL database. The reason for this decision is that it provides the benefit of implementing it as a local database, eliminating the unnecessary complexity that comes with a cloud-based database. This opens the door to the use of technologies such as SQLite, which, as previously discussed, is an embedded serverless database. Additionally, the team was already familiar with SQL. Finally, while this approach is stricter, requiring structured data schemas, its excellent documentation made learning the language simpler than trying to understand the nuances that come with its NoSQL counterpart.

3.2.4 Backend Technologies

The development team carefully evaluated several backend technologies to determine which would best support the smart aquarium system's needs, including sensor data collection, device control, and integration with the mobile application. The team considered three options: Adafruit IO, a Node.js server, and Firebase. Each of these technologies offered distinct advantages and features that could contribute to the project's overall success.

Adafruit IO

Adafruit IO is a cloud-based platform specifically built for Internet of Things (IoT) applications, providing a set of tools to efficiently manage sensor data, device interactions, and cloud connectivity. The platform supports real-time data streaming, allowing devices like the ESP32 to communicate seamlessly with mobile applications and other connected devices. It is especially well-suited for small to medium-sized IoT projects, offering a straightforward setup and user-friendly interface. Adafruit IO's simple integration with microcontrollers, including the ESP32, ensures reliable data handling and minimal configuration, making it an excellent choice for the smart aquarium system.

Node.js Server

A Node.js server provides the flexibility to build a custom backend capable of handling HTTP requests, managing WebSocket connections, and maintaining a real-time database. Its non-blocking I/O and event-driven architecture make it ideal for efficiently managing high-concurrency applications. This approach is particularly beneficial for projects that require tailored backend logic. However, while Node.js offers significant control over server operations, it also involves the responsibility of managing and maintaining the server infrastructure. Since our smart aquarium project only requires simple commands for sending and receiving trigger keys, the team decided not to opt for this option.

Firebase

Firebase, a platform by Google, provides tools like real-time databases, user authentication, and cloud functions, among other features. Its NoSQL database is great for fast and flexible data handling, which is especially useful for apps needing real-time updates and offline access. The built-in user authentication makes managing users in an IoT setup much simpler. While Firebase's managed services reduce the hassle of server maintenance, it does come with some limitations, like less flexibility in data structuring and querying compared to traditional SQL databases. This approach does not follow a traditional server set up. Instead it combines various backend technologies services that arent needed since the team has already chosen specific ones. For this reason, firebase was not used since it overcomplicated what was needed to develop Azultank.

Table 3.22: Tech Comparison between backend technologies

Criteria	Adafruit IO	Node.js Server	Firebase
Setup Complexity	Plug-and-play for IoT projects	Moderate, requires setup and configuration	Easy to moderate, managed service

Data Handling	Real-time streaming, easy integration with sensors	Customizable data handling, real-time with WebSockets	Real-time database with synchronization
Flexibility	Limited customization options	Highly flexible, custom backend logic	Limited customization options
Real-time Capabilities	Excellent for IoT	Real-time with WebSockets	Excellent real-time data sync
Documentation	Extensive, IoT-focused	Extensive, Node.js community	Excellent, well-supported by Google

3.2.5 Operating System

Regardless of which technology we use, Flutter or React Native, both options support cross-platform development for Android and iOS. Before choosing a platform, we must consider particular factors such as the compatibility and limitations when working with the full stack technologies discussed above, the team's coding experience, development cost and the size of the user base. Below, we briefly describe each Operating System, why each was considered and which one was chosen as our developing target.

iOS

iOS is Apple's proprietary mobile operating system, which powers iPhones and iPads. Thanks to the stack technologies in consideration, the team doesn't have to concern itself with learning how to code in Swift, Apple's proprietary language for iOS App development. However, issues may arise when attempting to run the application on an iPhone. There are scenarios which require the use of xCode. This is Apple's integrated development environment (IDE), which was developed for macOS and used to develop iOS applications. We must keep in mind that none of the team members owns an Apple Mac, and while there exist alternatives, they just aren't as good as natively working with an Apple system.

To compile and distribute an iOS app the use of xCode is needed to generate an ".ipa" file or iOS app package. Now, since the distribution of the application is not the focus of this project, it isn't a primary concern, but we are still facing possible issues when attempting to run the app on our own devices without the use of xCode. If we choose to use Flutter, the team will hit a wall since xCode is an absolute requirement to complete the development of the app. If we went with React Native, we could simply use Expo Go to build and test our application. However, this doesn't come without its caveats.

Expo Go is a convenient and powerful tool, but unfortunately, it has its own limitations when working on an iOS application. For example, the limited access to native modules, lower performance, no offline use and limited access to some hardware features. Of all these, the biggest problem being the limited access to native modules since some application features may rely on custom native modules or third-party libraries.

According to online sources, as of 2024, the iOS user base in the US holds 58.81% of the market share and 28.32% of the global market share (Howarth, Josh). If we were to consider the distribution of the application, then this is an important piece of data we must keep in mind. Finally, to publish an iOS app in the App Store, developers must enroll in the Apple Developer Program, which has a cost of \$99 per membership per year.

Android OS

Android is a mobile operating system based on a version of the Linux kernel and while open source, it's primarily developed by Google. Now while Android applications are traditionally developed in Java and the team has Java experience, going with technologies such as Flutter or a framework such as React Native would speed up development since we wouldn't be coding from scratch.

A great benefit of working with this operating system is the team wouldn't run into similar proprietary "xCode" problems found when developing in iOS. Android's integrated development environment is available on both Windows OS and Mac OS. In the team's case, Windows is the main platform to be used when developing our application. Additionally, Android's IDE enhances development when used in conjunction with either React Native or Flutter, making this option viable for the team.

In the case of React Native we could potentially run into similar issues as with the iOS counterpart. Reiterating these issues, these were limited native module support, lower performance and limited offline use. However, there are solutions to them, such as simply ejecting the application from Expo Go and continuing to work outside of it. As discussed earlier, these problems wouldn't apply to Flutter since it takes a completely different developing approach with the use of IDEs such as Android Studio.

As for Android user stats, according to online sources, the Android user base in the US holds 40.81% of the market share and 70.23% of the global market share (Howarth, Josh). While the user base in the US may be smaller than that of iOS, it more than makes up for it with more than double the size on a global scale. The cost of publishing to such markets is a one-time fee of \$25 (Google Play).

Microsoft Windows OS

Microsoft Windows is a proprietary graphical operating system developed by Microsoft. Developing on a Windows platform opens countless possibilities and approaches for creating the project's companion application. Since this is a desktop operating system rather than a mobile OS we would no longer be limited to Flutter or React Native for mobile development purposes. However, full stack technologies such as Flutter and React Native can still be used to develop the application or web client.

Windows is considered the most popular operating system in the world, and it holds 70% of the global market share (Qasim). Looking at the global user base, developing for Windows seems like a viable option, but the team must also consider its limitations. The purpose of the application is to offer the user a convenient solution to be able to monitor and tend to their aquarium while away from home. Windows OS would be running on a desktop PC, laptop or on a more portable option such as Microsoft Surface device. Now, while the Surface is the most portable device, its user base is much smaller than the one of iOS and Android devices. Additionally, it must be considered that not all users are willing to carry around a device of this size. With these factors in mind, it's unlikely the team will develop the application for Microsoft Windows OS.

Table 3.23: Tech Comparison of iOS and Android.

Criteria	iOS	Android OS	Windows OS
IDE Requirements	Requires an Apple Mac system for native access to xCode IDE.	Supports a variety of IDEs on both Windows and Mac systems including Android Studio.	Supports a variety of IDEs larger than Android and iOS.
React Native Compatibility	App can be developed using the framework but cannot be fully tested or deployed without the use of xCode.	Full development can be completed within the framework but can be enhanced/facilitated with the use of Android Studio.	Full development can be completed within React Native.
Flutter Compatibility	App can be developed using the framework but cannot be fully tested or deployed without the use of xCode.	Full development can be completed within the framework but can be enhanced/facilitated with the use of Android Studio.	Full development can be completed within Flutter.

User Base	58.81% of the US market share and 28.32% of the global market share.	40.81% of the US market share and a 70.23% of the global market share.	70% of the global market share for desktop operating systems.
App Publishing Cost	\$99 per developer membership per year.	A one-time payment of \$25.	A one-time payment of \$19 per individual account or one-time payment of \$99 per company account.

For the operating system, although React Native supports cross-platform development, the team decided to focus on iOS initially. This choice stemmed from practical considerations. First, every team member owned an iPhone, providing an accessible and consistent testing environment. While the team lacked a macOS system, which is traditionally required for iOS development, Expo offered the necessary tools to create application builds that could be installed locally on the team's devices without the need for a Mac.

Focusing on iOS first also streamlined development, allowing the team to prioritize testing and debugging on a single platform before addressing the complexities of cross-platform compatibility.

Chapter 4 - Related Standards and Design Constraints

Adhering to established standards and navigating design constraints are important to ensuring the system's functionality, reliability, and compliance with industry norms. Standards offer a framework for interoperability, safety, and performance, guiding the design and implementation processes.

On the other hand, design constraints define the boundaries within which the project must operate, including technical, economic, and political limitations. This chapter goes into the various standards pertinent to our project and the specific design constraints we must consider. By addressing these aspects, we aim to navigate the complexities of the project effectively and deliver a robust, high-quality product that meets user expectations and requirements.

4.1 Industrial Standards

Industrial standards are established guidelines and specifications that ensure products and systems are safe, reliable, and consistent. They cover a wide range of areas, including materials, processes, safety protocols, and performance criteria. Following established standards gives direction to the design and

development process by providing clear guidelines and best practices, ultimately contributing to the efficiency and success of our project.

4.1.1 PCB Standards

The PCB design standards is a criteria set to ensure the quality, reliability, and manufacturability of printed circuit boards within the industry. For our project, we will be using the PCB manufacturer JLCPCB and will follow their set criteria and standards for PCB design. JLCPCB allows the user to be able to comply with a variety of different design standards that include IPC, ASME, and Military. For the most part, JLCPCB follows the general PCB design standard of IPC-2221 which provides a detailed guideline for the design, material selection, electrical and mechanical performance, and environmental resilience. The IPC-2221 standard encompasses a wide variety of design standards to meet high-performance, streamline production processes, and ensure compatibility with various electronic components and systems.

Subcategories to be aware of that take the IPC-2221 standard as a template and further fine tune specific aspects of the PCB technology on its physical design principles.

- The first sub-category is the IPC-2222 which is the rigid organic printed board structure design. This one focuses on materials, layout and mechanical properties.
- The second sub-category is the IPC-2223 which is the flexible printed board structure design. This one focuses on material selection, bend radius and other unique flexible circuitry.
- The third sub-category is the IPC-2224 which is the Organic, PC card format, printed board structure design. This one focuses on dimensional stability, thermal management and electrical performance.
- The fourth sub-category is the IPC-2225 which is the Organic, MCM-L, printed board structure design. This one focuses on integration of multiple ICs onto a single substrate
- The fifth sub-category is the IPC-2226 which is the High Density Interconnect (HDI) structure design. This one provides guidelines for feature finer lines and spaces, smaller vias, and higher connection pad densities.
- The sixth sub-category is the IPC-2227 which is the embedded passive devices printed board design. This one focuses on addressing the design considerations for incorporating embedded passive devices which aim to optimize space and performance.

When designing a PCB using EasyEDA, we also have to consider the manufacturer limitations to ensure the fabrication reliability. As the limitations may vary between other manufacturers, these are still common values for minimum and maximum sizes for JLCPCB. For our selected manufacturer we are limited to board thickness of 0.4mm-3.2mm and copper weight to be 0.5oz-3oz. The outer layers have to have a trace width between 0.1mm-0.5mm, trace spacing between

0.1mm-0.5mm which is subject to change depending on width and solder mask clearance at 0.075mm-0.15mm. For the hole sizes they are limited to a drill hole size of 0.2mm-6.3mm, annular ring size of 0.1mm, and a via hole size of 0.2mm-0.8mm. Lastly we also have to consider the clearances of the components which need a minimum of 0.5mm and a copper to edge minimum clearance of 0.25mm.

Besides following JLCPCB's own manufacturing standards, we will be following IPC-2221 general standards as it serves as a proper comprehensive guide to PCB design that will minimize costs when adhering to it. By utilizing these standards, it will aid us by providing the knowledge to create a reliable, quality, safe, reduced error and defective PCB.

4.1.2 Sensor Standards

Ensuring that the sensors used in our project are accurate and reliable is of great importance, and following established sensor standards is essential in achieving this objective. The ISO 14000 family of standards provides comprehensive guidelines for environmental management systems, which are crucial in ensuring that the sensors and electronic components utilized in the project do not negatively impact the environment.

Specifically, ISO 14001:2015 sets the criteria for an effective environmental management system (EMS). By following ISO 14001:2015, we can establish a system of steps that helps us manage our environmental responsibilities. This includes reducing hazardous waste, promoting resource efficiency, and minimizing our ecological footprint. By applying these standards we ensure that our components are operated in an environmentally friendly manner, contributing to sustainability and compliance with environmental regulations.

In addition, ISO 14004:2016 offers general guidelines on principles, systems, and support techniques, providing additional guidance for implementing and maintaining an EMS. By integrating these guidelines, we can enhance our project's environmental performance and ensure that all stages of our sensor lifecycle, from usage to disposal, follow sustainable practices. This not only helps by reducing the environmental impact but also improves the overall efficiency and effectiveness of our project management processes.

In addition to environmental standards, using specific sensor standards is crucial for achieving accurate and reliable measurements. ASTM D1886, for example, consists of standard test methods for turbidity, which is important for accurately measuring water clarity in the fish tank. Turbidity measurement is an essential parameter for maintaining water quality, as it affects light penetration and the overall health of the ecosystem. By following ASTM D1886, we ensure that our turbidity sensor provides precise and reliable readings, allowing for effective monitoring and maintenance of water quality. This standard follows the

procedures for calibrating and testing turbidity sensors, ensuring that they meet thorough accuracy and durability requirements.

Furthermore, ISO 7027:2016 specifies methods for determining turbidity, offering an alternative or complementary approach to ASTM D1886. By adhering to ISO 7027:2016, we can compare our turbidity measurements, enhancing the robustness and reliability of our data. This double check ensures that our water quality monitoring system is both accurate and consistent, enabling proactive management of the fish tank environment.

By thoroughly applying these international standards, we demonstrate our commitment to environmental health and technical qualities. This approach not only enhances the credibility and reliability of our project but also follows global best practices. Ultimately, integrating these standards into our project development process ensures that we deliver a high-quality, environmentally responsible, and technically sound solution for maintaining optimal conditions in the fish tank.

4.1.3 Communication Standards

Communication standards are essential guidelines and protocols that ensure operability, compatibility, and reliable data exchange between different devices and systems. These standards define how data is transmitted, formatted, and processed, providing a common framework that allows diverse hardware and software components to work together seamlessly. In the context of our project, adhering to communication standards is crucial for integrating various sensors, controllers, and user interfaces, ensuring that the system operates efficiently and effectively.

4.1.3.1 UART Communication

UART stands for Universal Asynchronous Receiver-Transmitter. This is a hardware communication protocol that provides asynchronous serial communication between devices. For UART, asynchronous means it's not synchronized by a clock signal, unlike other communication protocols such as SPI or I2C. It is considered one of the simplest methods for data exchange and is typically used in low-cost or simple data transfer applications. It consists of two primary data lines: the transmit line (TX) and the receive line (RX). In other words, this setup enables full-duplex communication and therefore, data can be sent and received at the same time.

As for its communication setup, the devices that will communicate through UART must be set to the same baud rate. UART transmits data in a series of bytes, typically 8-bits. Each byte is framed by a start bit, followed by the data bits and then the stop bits. However, in some cases, there may be optional parity bits after the data bits. With this set, the data to be transmitted is converted from

parallel data into serial form and then sent over to the receiving UART, where it is converted back to parallel data.

In general, UART benefits from being a straightforward communication protocol that is easy to implement. It requires minimal hardware resources and it's widely known and popular to use. Thanks to this, there's robust documentation available about implementing it into projects. The fact it uses no clock signal eliminates the complexity that comes with other synchronous communication protocols, such as SPI or I2C. It's more flexible being a full duplex protocol than other methods that can either transmit in one direction or in both ways, but not simultaneously. Additionally, while it includes a basic error detection mechanism, it's very welcome since it is optional, but it also enhances the reliability of the data transmission.

On the other hand, it has its disadvantages when compared to protocols such as SPI or I2C. Its straightforward point-to-point implementation comes at the price of being generally slower than both protocols. This means UART is less suitable for applications that require data transfers at high speeds, but in the case of our project, it wouldn't be a concerning factor since the aquarium automation consists of simple requests. Additionally, the data frame is limited to 9 bits, and the protocol suffers from signal degradation over long distances; therefore, it's typically used within short-distance communication setups. Since UART utilizes a fixed point-to-point communication method, it's also limited to a single master-single slave configuration. Unfortunately, this means it's a less efficient protocol for multi-device networks, but again, this wouldn't affect our project considering that, at most, we would have a single master-single slave setup. Finally, a large drawback of UART is the lack of flow control. This could lead to data loss if the receiver (RX) isn't ready to process incoming data.

With the communication protocol established, it is important to note how it will affect the project. A mobile application will control the smart aquarium, but before we move on to its development, critical components of the smart aquarium must be programmed and tested for demo purposes. Since we are demonstrating the capabilities of our sensors and no application UI is available, the data obtained by the sensors will be displayed on a PC's terminal program. In more detail, the ESP32 will run the code that configures the connected sensors to activate and provide the temperature, turbidity and pH level data. Then, the data will be displayed on the terminal through a USB to UART bridge connection by using functions provided by the ESP32's SDK, such as `Serial.print()` if using the Arduino framework.

In general, while UART has its disadvantages, its advantages far outweigh them when it comes to our project's requirements. The lack of a clock signal makes this protocol straightforward and simple to implement. Finally, it is a key communication protocol for the project since obtained components, such as the sensors, must be tested by printing data obtained by them into a terminal program until the project's mobile application is fully developed.

4.1.3.2 Analog Communication

Analog communication refers to the transmission of data using continuous signals that vary in amplitude, frequency, or phase. These standards ensure that analog devices can communicate accurately and consistently, even in the presence of noise and other environmental factors.

The ESP32 microcontroller is a highly versatile component, one crucial aspect of utilizing it is managing its analog connections. The ESP32 is equipped with multiple ADC (Analog-to-Digital Converter) channels, which are essential for interfacing with analog sensors such as those measuring temperature, pH, and turbidity.

The standard practice for analog connections involves several important considerations to ensure reliable and accurate data readings. First, the voltage levels of the analog signals must be within the ESP32's input range. This may require voltage dividers or level shifters to adjust the signal levels appropriately. Additionally, it is important to implement proper filtering techniques to minimize noise and ensure clean, stable readings. This can be achieved by adding capacitors and resistors in the analog circuitry to filter out high-frequency noise and stabilize the signal.

Proper grounding and shielding are required to prevent electromagnetic interference, which can distort the analog signals. Ensuring that the analog sensors and the ESP32 share a common ground helps maintain signal integrity. Moreover, the placement of components and routing of PCB traces should be designed to minimize interference and crosstalk.

4.2 External Design Constraints

External design constraints are factors outside the control of the team that can affect the design, development, and implementation of our project. These constraints come from various external sources, such as regulations, environmental conditions, market demands, and technological advancements. Recognizing and addressing these constraints is essential to ensure the project's completion and success. These constraints are essential for delivering a reliable and market-ready product that meets the needs of fish tank owners while adhering to all relevant standards and requirements.

4.2.1 Time

The main or probably the most important constraint for this project is time. Various factors limit the availability and effectiveness of how we use time for tasks and activities. For this project for example, we are only given a finite amount of time to design and accomplish the overall project. Because of this we considered some general issues for this constraint as well as required critical decisions along the way.

One of the general issues to consider is time management. This is crucial as we must balance the project with life, schoolwork, and other activities. Our group is a bit unfortunate because we could not decide on what project to do before the start of the semester, which has put us a bit behind schedule. However, we are carefully organizing our required tasks, and working diligently to catch up and stay on course to make up for this lost time.

Another general issue for this is dependency management. We need to understand that some tasks depend on the completion of other tasks. If a preceding task is delayed, it could possibly create a domino effect causing it to push back other tasks. This will also cause a halt and compromise the overall project schedule. One example for this is the PCB (Printed Circuit Board). Before we can proceed with assembly, testing, and integration into the final product, we must first build this component. If a case where it is not completed in time or is delayed in any way, the entire may come to a pause. An increase of costs can also cause an increase of costs due to rushed work or expedited shipping. Most importantly, because of the said delay, any issues found during testing cannot be addressed in time.

Unexpected Events is a general issue for time as well. Unforeseen events could be something like a technical issue, changes with the project or even personal emergencies disrupt the planned schedules. These incidents could require the team to spend more time troubleshooting or force them to change the original plans, which could cause delays in the overall time frame. Our project requires communication between the components and the app to monitor the aquatic life and its environment as well as control the sensors and dispensers remotely. For example, we encounter a technical issue where one of the sensors is not being accurately transmitted to the app because of a bug around the code. Debugging this issue could have us spend additional unplanned time identifying the root of the problem as well as fixing the overall code.

The following issues above could cause an outcome where the overall project does not reach the desired or necessary standards. This might force us to cut corners, make last-minute changes and possibly just do the bare minimum just to meet the deadline. This will also cause us to skip important details and thorough testing, reducing the overall quality and efficiency of our project. As a possible consequence, we could possibly risk failing our project and our grade if we do not address these constraints/issues as soon as possible.

Effective communication, continuous monitoring and careful planning can help avoid these constraints. These things could help the team to look ahead to spot potential problems and plan how to handle them before they arise. We also made sure that our team is ready to respond to any changes by promoting open communication and continuously reviewing our progress to make required modifications. With the implementation of these proactive methods, we will be able to better manage our time and keep the project on track for completion.

4.2.2 Economic

These constraints set the financial boundaries the project must operate with, influencing every decision from the selection of components to the methods used for prototyping and manufacturing. Given that our project lacks external funding or sponsorship, and all expenses are coming directly from our pockets, the challenge of managing economic constraints becomes even more pronounced.

The most immediate economic constraint is the budget. With no external funding sources, our financial resources are strictly limited to personal contributions from team members. This requires a delicate approach to budgeting, where every dollar spent must be accounted for and justified. The budget will cover a range of expenses including, but not limited to, purchasing components, tools, and materials, as well as any fees associated with prototyping services or testing facilities.

To minimize costs, we must explore cost-effective prototyping methods. This could include using 3D printing for custom parts, leveraging university facilities for access to tools and equipment, and utilizing open-source hardware and software wherever possible.

Another challenge we face managing economic constraints is balancing cost with quality. High-quality components, while better for their superior performance and durability, often come with a higher price tag. On the other hand, opting for cheaper components can lead to reliability issues and shorter durability, ultimately increasing the total cost of the product to the owner due to frequent replacements and repairs. We must conduct thorough market research to identify suppliers that offer the best value for money. This involves not only comparing prices but also evaluating the reputation of suppliers and the quality of their products.

Economic constraints also impact how scalable the project is. High production costs can make it hard to produce the AzulTank at a price point that attracts customers. Making sure the project is economically viable involves careful financial planning and smart use of resources. This includes using cost-effective methods for prototyping, optimizing the design for easier manufacturing, and considering long-term maintenance and support costs.

Innovation can also play a role in managing economic constraints. This might involve developing new techniques or methods that are more cost-effective than traditional approaches. Due to a lack of funding or sponsors, creative solutions are needed to make the most of available resources. This means using open-source software and hardware, and trying not to spend a large amount of money if it can be avoided.

Managing these constraints requires a careful balance between cost and quality, strategic sourcing of materials and components, cost-effective prototyping, and

efficient resource optimization. By considering long-term costs, exploring alternative funding opportunities, and incorporating cost-saving innovations, we can navigate these economic challenges effectively. Despite the lack of external funding, a thoughtful and strategic approach to budgeting and resource management will allow us to develop a high-quality, affordable AzulTank that meets the needs of fish tank owners.

4.2.3 Sustainability

For the design and construction of our smart aquarium product, we must consider both hardware and software components. We intend to create a compact device that can self-sustain the aquarium for as long as possible by auto-feeding the fish, regulating pH water levels and notifying users about the water quality in general. A large selling point of the product is to be able to monitor and take care of your fish while you are away from home; we must always ensure it's powered. Therefore, it should utilize a DC source. Additionally, its material must be able to withstand any accidental water contact.

Now, a large concern is ensuring the electrical components don't come into contact with water. Due to this constraint, the design must implement some sort of waterproofing method, whether it's by strategically placing the components within the product's enclosure, making the enclosure waterproof or the components themselves. The device will be positioned against the glass of a fish tank, so it's essential that its internals are protected at all costs. If the stretch goals are implemented, the product's enclosure will have a cut out for the positioning of a camera. This means there will be a need to implement certain protections against moisture that could affect the camera lens. The device must be able to withstand a range of operating temperatures. We must assume it won't always be utilized in a home with air conditioning, so tropical weather homes must be considered, especially when the device will always need to be powered on due to its Wi-Fi capabilities. In addition, it must be able to withstand the colder temperatures of the markets under consideration. Therefore, we must carefully choose each one of our components to ensure it can self-sustain home aquariums for as long as possible.

Moving onto the software side, the application must offer a modern and intuitive user interface, ensuring ease of use. A visually appealing user interface can keep users engaged longer and encourage the use of the application in the future. The team is aiming to create an application that is easy to maintain so that it can be updated as needed for years to come. This is essential to its lifecycle since it would require updates for new features and bug fixes as well as updates so that it remains compatible with multiple OS upgrades. Sometimes, phone OS will phase out applications that don't meet a certain criterion, such as SDK changes, which can increase the complexity of updating the app by requiring updates to the codebase, dependencies, or any third-party libraries used, which is why keeping the software sustainable is a high priority. A hardware product is only as good as its software. With this in mind, a well-designed application will not only

enhance customer loyalty and retention but also increase the chances that the product will achieve success.

Overall, with these constraints set, we have a clear goal on how our product should be designed to ensure it's useful, effective, and durable in terms of how long it'll be able to operate to become a competitive product in the market.

4.2.4 Other Constraints

4.2.4.1 Political

Dealing with these constraints is crucial to make sure the product meets all safety, environmental, and compliance standards, allowing it to be legally sold and used in different markets.

A key political constraint is compliance with local and international regulations. This means following guidelines from regulatory bodies like the FCC (Federal Communications Commission) for wireless devices. Complying with FCC regulations ensures that the ESP32 and other wireless components used in the Fish Tank Helper don't interfere with other electronic devices and operate within approved frequency ranges.

Trade policies also create political constraints, affecting the sourcing and cost of components and materials. Political stability and government support for technological innovation can influence the project's timeline and feasibility. For example, changes in trade policies on imported components can increase costs and disrupt supply chains, requiring adjustments to the project budget and timelines.

4.2.4.2 Ethical

For the development of this project we considered that it is important to take into account a number of ethical limitations in order to prioritize the ethical use of technology.

We make sure that no unethical methods such as labor exploitation or obtaining components from vendors with poor labor practices are involved in the technology being utilized. Working with vendors that comply with the highest moral standards in hiring practices is something we really value.

To guarantee that these ethical requirements are satisfied, we make sure that our project uses components that are obtained from trusted companies in which it includes a thorough search for suppliers and ensures that they uphold ethical standards. Overall, it is important to continue to improve the project while upholding our ethical ideas by routinely evaluating and modifying the system in response to user input and new technological advancements.

4.2.4.3 Health and Safety

Healthy and Safety constraints is another thing that must be factored into this project. We do this by making sure that the water temperature, pH levels and oxygen level are within the safe range for each specific fish being kept. Although we are not using any fish to test this project, we still take this into account to ensure that we provide ideal living conditions for the aquatic life and its environment.

Considering how our entire product is powered by electricity, user safety is an additional important factor to take into account. In order to ensure that our overall project is safe for users of all ages, non-toxic materials must be used to prevent accidental electrical shock and ensure that all components are safely enclosed to avoid injury for installation and testing.

Lastly, we also need to consider the mechanical safety of the fish and the user who will interact with the aquarium. Again, although this project will not involve any aquatic life, we nevertheless want to make sure that the system works properly in the event that aquatic life is later added to it. This will reduce the risk of injuries, and malfunctions if components are properly installed and protected. This will overall create a safe system and provide better experience for both the fish and the users themselves.

4.2.4.4 Manufacturability

Lastly, with the development of this project we have considered the constraint of Manufacturability. When it comes to accessibility of tools and or equipment, we as a group are limited to the equipment found in the university's lab. When constructing different portions of our project, like the electrical hardware components such as the power supply unit and pcb, we had to design for a more simple approach. The overall impact that this has on us as a group is that it prevents us from branching out to a more complex and innovative design that may require equipment that we aren't able to access easily or at all.

Although, there are portions of the project components that do not have to be created from scratch which reduces the manufacturing time and these include the sensors and motors that will be bought separately. Given the group's technical skill spread, this will allow the design to be less hardware intensive which can allow the group more fluidity in constructing with our limited equipment. This can be seen as a positive as we don't have sole access to the university facilities and may need to share with other students which may limit the manufacturing time of our electrical equipment.

Chapter 5 - ChatGPT

Artificial Intelligence or mainly known as AI refers to the development of machines and systems that are capable of performing activities that require human intelligence. These said activities include thinking, learning, problem solving and language comprehension to overall enhance human capabilities. In this day and age, AI is seen as a commodity; it has impacted our day to day life and businesses, from the most mundane to the most complex applications. One of the primary benefits is its ability to process and analyze vast amounts of data quickly and accurately, far beyond human capabilities. This allows AI to identify patterns, make predictions, and provide insights that can drive decision-making across various industries. Additionally, AI can automate repetitive and mundane tasks, increasing efficiency and freeing up workers to focus on more complex or creative activities. The consistency and precision of AI systems could also reduce the possibility of human error, improving the overall quality of outcomes. Some popular and well-known examples of AI are ChatGPT developed by Open AI, Copilot developed by Microsoft and Gemini developed by Google DeepMind.

ChatGPT is developed by OpenAI and launched in November, 2022. It is a chat bot and a virtual assistant. Machine learning algorithms are used to process and analyze vast amounts of data in order to provide responses to user inquiries. One of the platform's primary strengths is its capacity to provide responses in real-time. Users may spend hours researching on writing an essay, solve a math problem or debug a code but ChatGPT may provide well-written or solution alternatives in seconds.

Another good example of an AI worth mentioning is Gemini. Formerly known as Bard, this AI platform was developed by Google in March 2023. It works similarly as ChatGPT where they both understand and generate human-like text and provide real-time responses. However, Google's Gemini mainly focuses on increasing productivity within Google Workspace by providing smart ideas and data analysis that are intimately linked with Google's ecosystem.

However, we must also keep in mind that AI can sometimes be unreliable due to several factors. The quality of the data used to train AI models is the most important; data that is biased, untrue, or outdated can lead to inaccurate or incorrect results. Sometimes, AI's inability to understand context and difficulty handling unclear inputs contribute to its unreliability. AI models trained in one discipline might not apply well to others, and they often struggle with edge cases that cannot be related to training data. Technical issues, such as software bugs or hardware failures, can also affect AI performance. Lastly, ethical and security concerns, including copyright infringement and privacy violations, can undermine the reliability and trustworthiness of AI systems. Addressing these issues is essential for developing more trustworthy and dependable AI technologies.

5.1 Case Study 1

“What requirements do I need to maintain a healthy aquarium?”

When asked, OpenAI’s GPT-3.5 answered:

Refer to Chat Log A located in Appendix B

This is a very straightforward, accurate response. Nothing out of the ordinary, not to mention the question wasn’t too thorough or specific. It could have given more details on its answer but kept it simple. Most of the information, if not all, could’ve been used in the research for our project. These are all well known characteristics of a healthy aquarium that match the information we’ve gathered so far; if given more context, it might have been able to provide a more in-depth answer that fits our necessities.

However, for this case scenario, the objective will be to compare the different responses that we can obtain, to the exact same question, from different AI models. This way, we can analyze each answer and see how each model differs from each other.

Up next, Google’s Gemini:

Refer to Chat Log B located in Appendix B

The two answers differ in several key aspects, particularly in the way they are organized, detail level, and clarity. ChatGPT’s response was organized into ten sections, each addressing specific components of aquarium maintenance. It provided a broad summary and included some general recommendations for each aspect. On the other hand, Google Gemini’s response was structured into four main sections. Each section was concise and included specific recommendations, such as maintaining tropical fish temperatures between 72°F and 82°F and performing partial water changes of 15-25% every 1-2 weeks. This approach made the information more accessible and easier to follow for beginners.

While both answers are complete in their own right, Google Gemini’s response might be considered more complete for a beginner due to its clarity, conciseness, and specific guidance. ChatGPT’s response, however, might be more valuable for someone looking for a more detailed and thorough understanding of aquarium maintenance.

5.2 Case Study 2

“What is the price range for servo motors vs stepper motors? Which one should I use?”

When asked, ChatGPT-3.5 answered:

Refer to Chat Log C located in Appendix B

The overall response of chatGPT shows a straightforward response. The pros and cons for each component as well as the suggestions given provides a comprehensive view to help the user to have an idea and make an informed decision based on specific needs and requirements. However, one must understand that response is arbitrary and not accurate thus giving a result of an approximate price range which could be either out of date or estimated from a database.

For our project, an accurate result could be given if we used a utilized search engine such as Google, Bing or Brave. Prices from various vendors allows us to compare and contrast accurate price ranges as well as more detailed features for each component obtained from the vendor's website. This feature includes the speed, torque, model as well as power consumption.

5.3 Case Study 3

“What are some common lights used for aquariums?”

When asked, ChatGPT-3.5 answered:

Refer to Chat Log D located in Appendix B

As always, chatGPT was able to give us a straightforward answer. By giving us the common lights used in an aquarium, we are able to get a good start and a baseline for our research. It also gave us an understanding which light is beneficial for their type of fish and other aquatic species. The results above gave us a good starting point on what to do and saved us time by narrowing down our searches on several search engines such as Google. This then directly helped us to go to a certain website for further research on each light that the AI suggested to us.

Although this further proves that ChatGPT is a great resource, it is still important to keep in mind that this is 100% considering that GPT-3.5 was last updated in November 2023. This also means that we should still do further research on our own and also ensure that we are covering all the aspects thoroughly as well as not miss any critical details and information.

5.4 Case Study 4

From this case study onwards we decided to upgrade to ChatGPT Plus, which gives us access to model GPT-4o. This new language model features advanced natural language understanding, which allows it to comprehend and generate text with greater accuracy and contextual relevance. This version has access to an expanded knowledge base, therefore including a broader range of topics and more up-to-date information, providing enhanced support for various inquiries and tasks. It also features the capability of understanding and analyzing images.

This allows the model to process both text and visual inputs, enabling more sophisticated and comprehensive interactions.

To test the accuracy, contextual relevance and expanded knowledge base of this new model, we decided to repeat a question from a previous case study:

“What is the price range for servo motors vs stepper motors? Which one should I use?”

Refer to Chat Log E located in Appendix B

Comparing the responses from GPT-4o and GPT-3.5 shows both similarities and differences in detail and structure. GPT-3.5 provides a detailed price breakdown for servo and stepper motors, it categorizes them into low, mid and high end ranges, which is useful for budget planning. On the other hand GPT-4o offers a broader price range overview, focusing on factors that affect the cost of the motor.

As far as performance and application of the motor goes, both models delved into advantages and disadvantages of both types of motors. GPT-3.5 seemed to focus on specifics, while GPT-4o focused more on the main benefits and drawbacks of each.

When choosing between servo motors and stepper motors, GPT-4o gives broad recommendations based on application requirements, suggesting servo motors for applications requiring precise control, high speed, and high torque, and stepper motors for budget sensitive projects or those requiring precise control at low to moderate speeds. GPT-3.5, however, offers more detailed guidance on specific applications, like robotics and CNC machines for servo motors, and 3D printers and small CNC machines for stepper motors, making it easier for users to see practical use cases.

In summary, GPT-3.5 offers detailed, to-the-point information, useful for specific inquiries, while GPT-4o provides a concise, comprehensive overview, suitable for those preferring a summary. Both responses are informative, with the choice depending on the user’s need for detail versus conciseness.

5.5 Case Study 5

“This is the hardware diagram for the fish tank attachment we're designing. Would you make any changes to it?”

When asked, and showed a picture of Figure 2.2 (our hardware diagram), ChatGPT 4o answered:

Refer to Chat Log F located in Appendix B

GPT-4o gave immediate feedback by stating that the diagram looks “well-organized and clear”. However, it added several comments on how we could improve on our diagram. The feedback was detailed and constructive, it addresses key aspects such as label clarity, power supply unit specifications, component connections, color coding consistency, additional components, legend details, interfacing and communication protocols. Its criticisms are completely valid and we could consider making changes to this diagram in the near future.

This new language model does a very good job at analyzing visual content and providing a detailed explanation for any enquiries that the user might request. For example, when evaluating a hardware diagram, GPT-4o can identify areas for improvement in label clarity, ensuring all elements are distinctly marked and readable. It can suggest indicating specific power requirements for the PSU, helping streamline the acquisition process. GPT-4o can also verify that all necessary component connections are correctly represented, such as ensuring the Wifi Module's connectivity to both the MCU and the mobile phone.

In terms of interfacing and communication, GPT-4o can suggest specifying the protocols used, such as Bluetooth or WiFi, to provide clearer communication pathways. It also emphasizes the importance of error handling, advising on indicating any errors for critical components like motors or dispensers. Finally, GPT-4o can comment on physical layout considerations, offering notes on how to indicate the proximity of sensors to the fish tank.

These new capabilities demonstrate GPT-4o's ability to provide thorough and valid feedback on visual content, ensuring clarity and accuracy in complex diagrams and enhancing the overall design process. In conclusion, we believe the new model brings relevant feedback to the table and can be considered as an asset to the team.

5.6 Case Study 6

With the upgrade to Chat GPT Plus, we also acquired access to DALL E, a GPT model created by OpenAI that allows us to generate images from the ground up. To test this model, we decided to make it help us visualize what the final product of AzulTank might look like. The output of this model can be easily modified to produce a completely different picture with the slightest change in the prompt.

“Visualize a polycarbonate enclosure on top of an aquarium that has a cut out inside the enclosure to fit an aquarium filter. The enclosure has a power supply unit inside it as well. The enclosure has openings to have dispensers to dump food in the aquarium”

DALL E generated the following images:

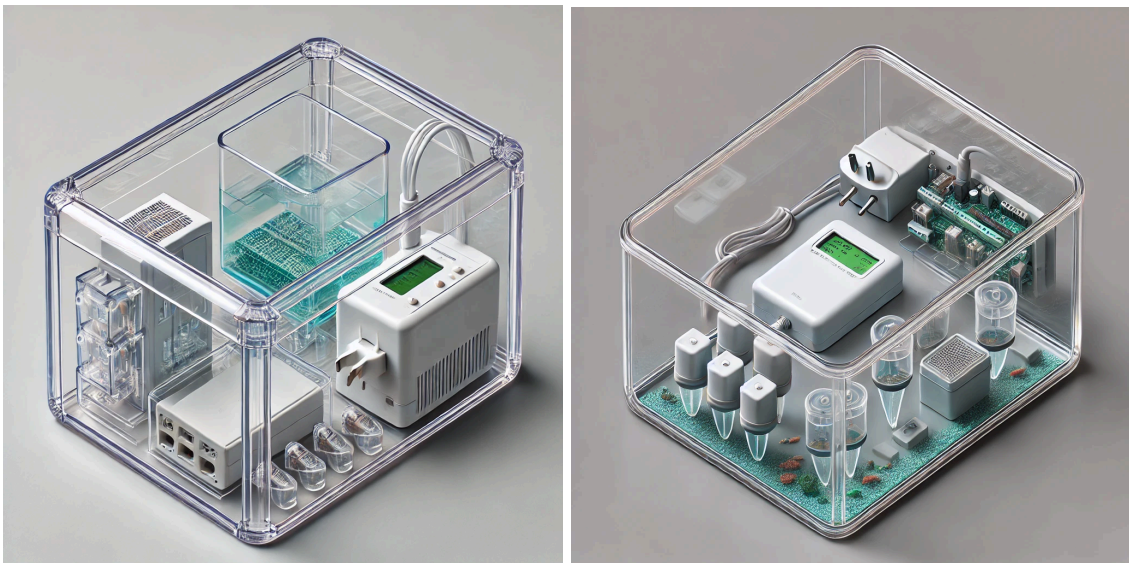


Figures 5.1 and 5.2: First attempt at DALL E image generation

The images that DALL E provided went below our expectations, they are not what we envisioned. It would seem like the model combined the enclosure with the aquarium and created a hybrid between both of these. These are not of much use, it didn't follow the specifications as given, and there is a possibility the prompt wasn't specific enough to warrant a design of our liking. Perhaps using a shorter, more detailed prompt could generate an image closer to our idea of what the AzulTank should look like.

"A simple polycarbonate rectangular box with a power supply unit, microcontroller, and food dispensers."

DALL E generated the following images:



Figures 5.3 and 5.4: Second attempt at DALL E image generation

The results of this prompt are more in-line with the expectations we have for the AzulTank. It seems to have completely succeeded at replicating some of the specifications provided, like for example the polycarbonate rectangular box, a power supply unit, a microcontroller (only on Figure 5.4) and several mechanisms that could serve as food dispensers .

DALL-E's ability to turn text prompts into images makes it a valuable tool for visualizing product ideas. By creating clear and detailed pictures from descriptions, it helps us see what the final product could look like. This makes it easier to discuss and improve designs, ensuring everyone is on the same page. Using DALL-E can speed up the design process and help catch potential issues early, making it a great asset for developing and refining new products.

5.7 Impact of Artificial Intelligence

AI has profoundly changed the way we live and how we see the world as a result of rapid advances in machine learning, and data processing. Human skills and capabilities have improved by making work more efficient and productive due to AI tools.

"Adapting to AI is not about pitting human intelligence against artificial intelligence, but about complementing and enhancing it." (Anonymous).

AI has enabled humans to automate repetitive tasks and provide intelligent suggestions. This has significantly reduced manual effort and allows humans to focus on more strategic and creative endeavors. It is essential to use AI wisely and thoughtfully as well as we integrate AI into our daily workflows.

AI can either have a positive and negative impact in regards to our Senior Design Project. The positive impact for this includes enhancing capabilities by automating complex tasks and helping enhance the overall quality of our project. But this also includes a negative impact where we could cause an over-reliance of using AI tools that could potentially result in a loss of critical thinking and problem-solving abilities among team members.

It is important to create balance between using AI skills and maintaining human oversight and judgment. It is also important that AI is there to only help and support human intelligence, not replace it.

Chapter 6 - Hardware Design

This section is a crucial part of the Azul Tank project, focusing on the detailed planning and structuring of the system's physical and electronic components. This chapter will provide an in-depth look at how each piece of hardware is selected, integrated, and organized to create a seamless and efficient system.

The main purpose of this section is the creation and analysis of schematic diagrams. These diagrams are crucial as they map out the electrical connections and circuitry, detailing how each component interacts within the system. By providing a clear and precise layout, schematics ensure that the design is both understandable and replicable. They serve as a blueprint for assembling the hardware, highlighting the pathways and connections that allow the system to function correctly.

Understanding the hardware design is essential for the successful implementation of the Azul Tank. It allows us to anticipate potential issues, streamline the integration of various components, and ensure that the system operates smoothly and efficiently. A meticulous approach to hardware design not only enhances the reliability and performance of the aquarium but also demonstrates our ability to apply engineering principles to solve real-world problems. By delving into the hardware design, we aim to create a robust foundation for our project, ensuring that the final product is both innovative and functional.

6.1 Power Delivery/ Electrical Power System Schematics

Figure 6.1 illustrates a DC-DC buck converter designed to step down a 12V input voltage to 5V. Buck converters are efficient power management systems that convert a higher input voltage to a lower output voltage using a combination of switching regulators, inductors, and capacitors. The circuit starts at the left with a DC barrel connector labeled U2 that provides the 12V input from our AC to DC wall plug, which is then filtered by the capacitors C2 and C3 to stabilize the input voltage. Closer to the middle of the regulator we have R1 which serves as the pull-up resistor for the enable pin to ensure the regulator is active. Then we have C4 which is there to filter the VREG5 output, while C8 ensures stable bootstrap operation for the high-side MOSFET driver.

Moving to the output side, the inductor L1 and capacitor C5 form the primary buck converter stage which filters the switching waveform for a smooth DC output. With capacitors C6 and C7 they further stabilize the voltage output while the voltage divider that is formed by the resistors R2 and R3 sets the desired output voltage. We then have an LED set to show if there's proper power flowing through the regulator before the output of 5V is delivered from the connector at U20.

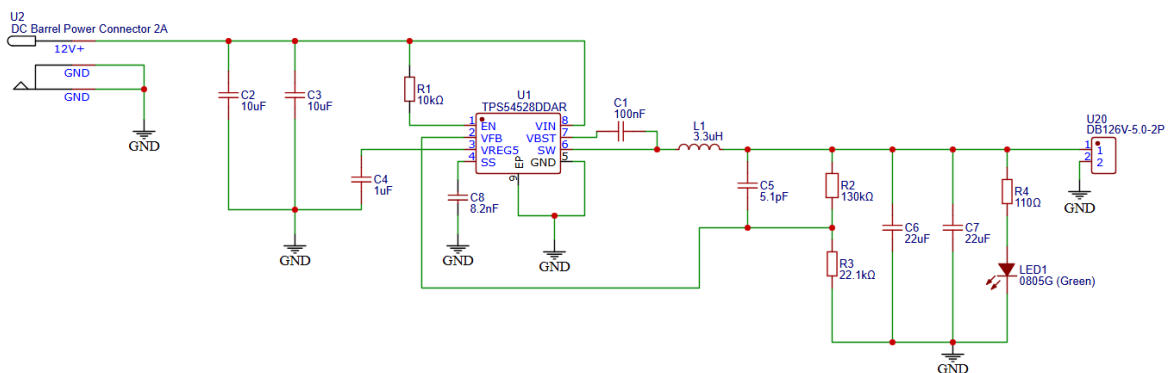


Figure 6.1: Schematic of 12V to 5V Regulator

By utilizing switching regulators, inductors, and capacitors, the circuits achieve smooth and stable voltage outputs. The feedback mechanisms within the regulators ensure precise control over the output voltages, making this converter

reliable for various sensitive electronic applications requiring specific voltage levels.

6.2 MCU Board Schematics

The ESP32-WROOM-32D microcontroller unit (MCU) serves as the brains of our system, meticulously guiding the interactions between various sensors, and other electronic components. This section will go through the detailed schematic involving ESP32-WROOM-32D, emphasizing its critical role and the complexity it manages within our hardware design. It is a versatile and powerful MCU that offers a broad range of functionality including WiFi and Bluetooth capabilities, making it an ideal choice for projects that require wireless communication and significant processing power. The schematic diagram we've created is designed to illustrate how this MCU connects to and controls the diverse hardware elements of the AzulTank, ensuring efficient and reliable operation.

The schematic includes detailed representations of each connection, from power supply inputs to signal outputs, and interfaces with various peripherals. It delineates the power management system ensuring that the ESP32-WROOM-32D receives the correct voltage levels for operation while also highlighting how data flows through the system, facilitating a comprehensive understanding of the communication between the MCU and sensors like the temperature, pH, and turbidity sensors, as well as output devices such as servo motor, stepper motor, and LED lights.

6.2.1 ESP32 Module

Figure 6.4 shows the core connections of the ESP32-WROOM-32D module. The module is powered by a +3.3V supply, which is crucial for its stable operation, and is further stabilized by capacitors to filter out any noise or voltage fluctuations that might affect performance. This setup ensures that the module receives a clean and stable power supply, which is essential for reliable operation.

The several GPIO pins that are labeled provide numerous options for connecting sensors and other peripherals. These GPIO pins are versatile and can be configured for different modes of operation, making them ideal for interfacing with a wide range of external devices. This connectivity is crucial for the AzulTank, as it allows for the integration of multiple sensors and actuators, each with its unique requirements.

The communication pins, RXD0 and TXD0, are specifically designated for UART communication, enabling serial data exchange with other devices. This is particularly important for tasks that require reliable and straightforward communication protocols, such as data logging or interfacing with other microcontrollers.

Additionally, control pins like EN, which is used to enable or disable the module, and SENSOR_VP and SENSOR_VN, which serve as ADC (Analog-to-Digital Converter) inputs, are detailed. These control pins are essential for the module's operation and configuration, allowing precise control over its functionality. SENSOR_VP and SENSOR_VN enable the ESP32 to interface with analog sensors, converting analog signals into digital data that the microcontroller can process. This capability is vital for reading data from sensors like pH meters and turbidity sensors, which provide analog outputs.

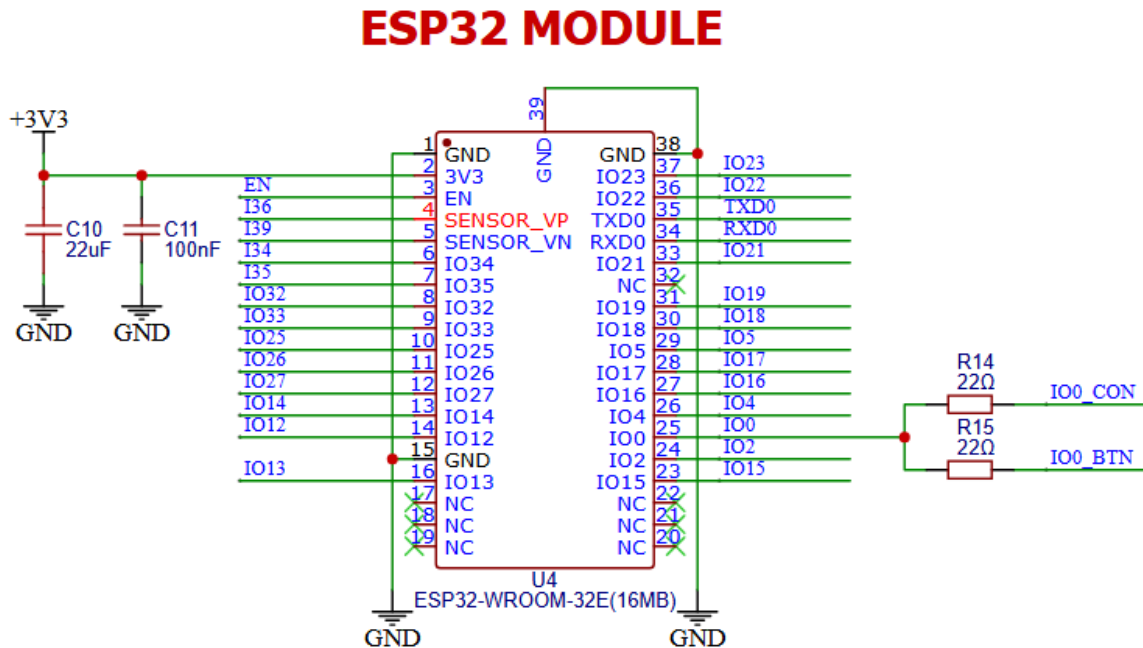


Figure 6.2 : Schematic of ESP32 Module

6.2.2 Connectors

Figure 6.5 below covers the connectors (CN5 and CN9), which play an important role in achieving the connection of the ESP32 with various other components and peripherals. These connectors are designed to ensure effective integration and communication between the microcontroller and external devices, which is essential for the overall functionality of the AzulTank.

These connectors include additional capacitors for power stabilization. Capacitors are important in any electronic design as they help get rid of fluctuations in the power supply, ensuring that all connected components receive an appropriate voltage. This is particularly important when dealing with sensitive electronic components, as power instability can lead to erratic behavior or even damage, which could end up as an additional expense. By adding these capacitors, we enhance the reliability and longevity of the system.

CONNECTORS

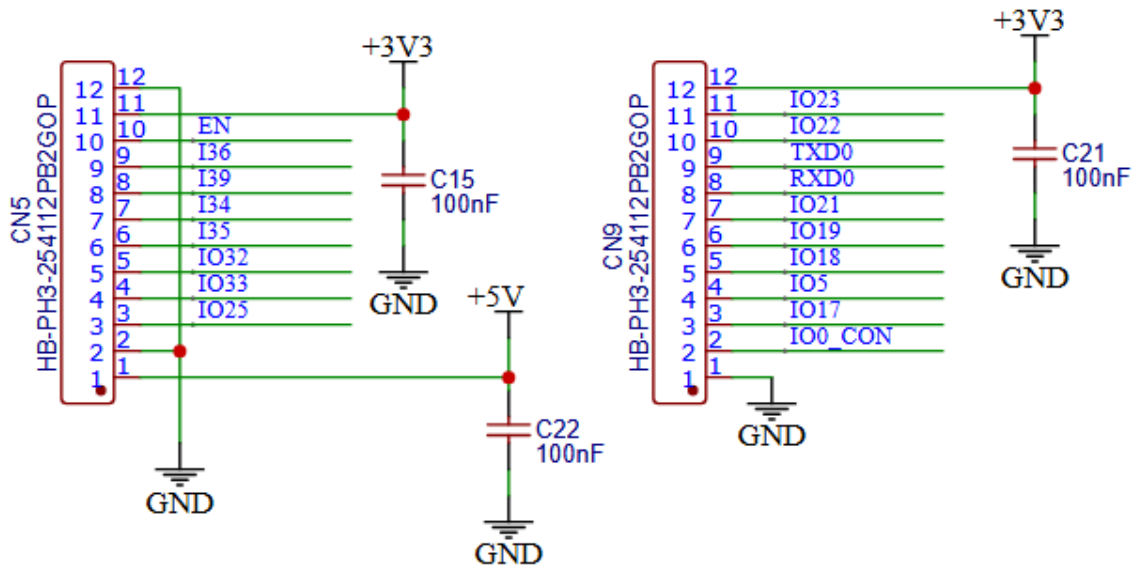


Figure 6.3: Schematic of ESP32 Connectors

The connectors also provide for extending the I/O pins, allowing for the easy connection of external sensors and modules. This extension is important for the flexibility of the AzulTank, it allows us to accommodate various sensors such as temperature, pH, and turbidity sensors, as well as actuators like servo motors and LED strips. Each I/O pin can be configured to serve different purposes, whether it's reading sensor data or controlling an output device.

Furthermore, external power input connections, such as +5V_EXT and +3V3, are provided for powering additional devices. These power inputs ensure that any external sensors or modules connected to the system have a reliable power source. The +5V_EXT connection is particularly useful for devices that require a higher voltage than the ESP32's operating voltage, such as servo motors or certain types of sensors. Meanwhile, the +3V3 connection is suitable for low-power devices that operate at the same voltage as the ESP32. This dual-power enhances the variety and scalability of the system, allowing it to support a broad range of peripherals without affecting power stability or system performance.

6.2.3 USB Connector

In Figure 6.6, we have the USB connector, which is a crucial component for programming and debugging the ESP32. It includes USB data lines (D+ and D-) that are responsible for data transmission. These lines enable bidirectional communication, allowing the ESP32 to send and receive data efficiently. This is essential for tasks such as code uploads, where the computer sends instructions to the ESP32, and debugging, where the ESP32 sends data back to the computer for analysis.

USB CONNECTOR

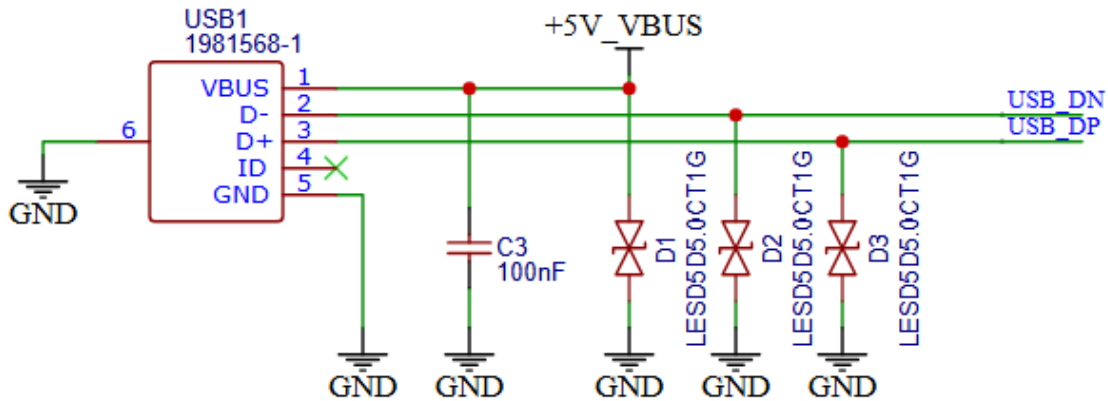


Figure 6.4: Schematic of USB Connector

Additionally, the USB connector provides a +5V_VBUS supply, which powers the ESP32 and other connected components during programming and debugging sessions. This power supply is filtered by capacitors to ensure a stable and noise-free voltage, preventing potential disruptions or damage to the circuitry. The capacitors smooth out any voltage spikes or drops, ensuring consistent power delivery.

6.2.4 USB to Serial Conversion

Figure 6.7 shows the USB to serial conversion, a critical process that enables programming and communication with the ESP32 by converting USB signals to serial communication. This conversion is managed by the USB Bridge CP2102N-A01-GQFN28R integrated circuit (IC), which efficiently translates USB data signals into UART signals that the ESP32 can understand and process. It ensures that data sent from a computer via a USB connection can be seamlessly converted into a format that the ESP32's UART interface can interpret. This conversion is crucial for loading programs onto the ESP32 and for serial communication during debugging and operation.

To ensure the integrity and reliability of the signals during this conversion process, various resistors and capacitors are employed for signal conditioning. These components play a vital role in maintaining the quality of the electrical signals by filtering out noise, stabilizing the voltage levels, and ensuring that the data is transmitted accurately without loss or corruption. Proper signal conditioning is essential to prevent errors in data transmission, which could lead to miscommunication between the computer and the ESP32.

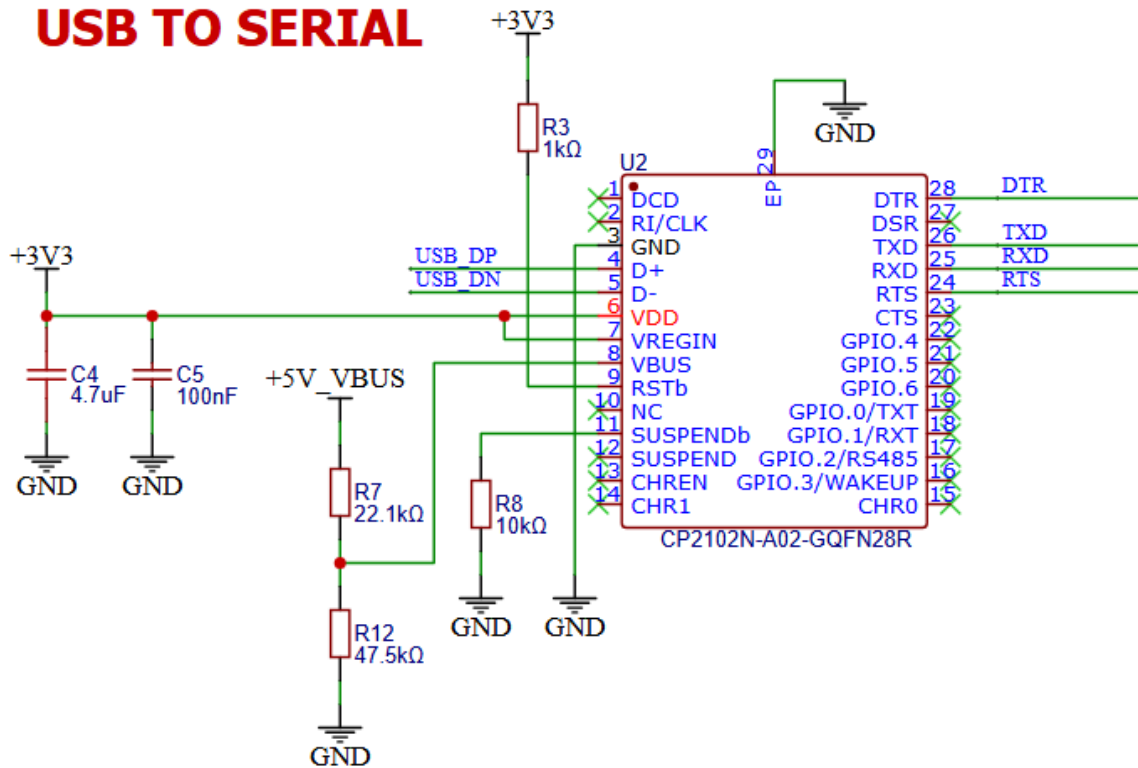


Figure 6.5: Schematic of USB to Serial Connections

The schematic also depicts the signal lines for USB data, labeled as USB_DP and USB_DN, which carry differential data signals from the USB port to the bridge. In addition to the data lines, control lines such as DTR (Data Terminal Ready) and RTS (Request to Send) are included. These control lines are used for managing the data flow and ensuring proper synchronization between the USB interface and the UART interface of the ESP32.

6.2.5 Serial Signals Handling

In Figure 6.8, the serial signals handling section of the schematic manages the signal handling for serial communication. Transistors Q1 and Q2 are used for level shifting and signal conditioning, which ensures that the voltage levels of the signals match the requirements of both the ESP32 and the connected peripherals. This is essential for maintaining signal integrity and preventing voltage mismatches.

SERIAL SIGNALS HANDLING

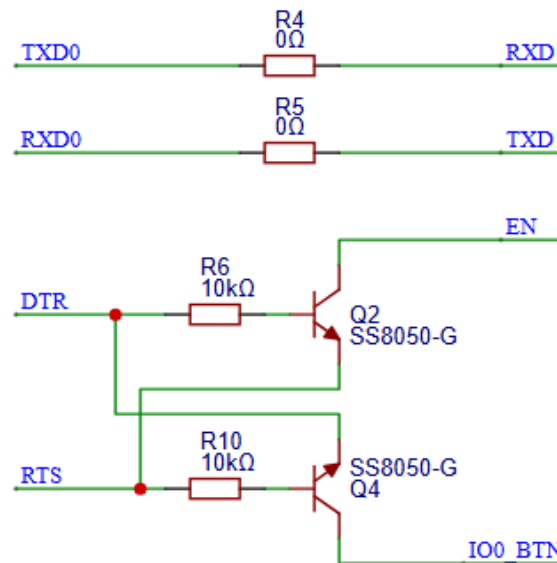


Figure 6.6: Schematics of Serial Signals Handling

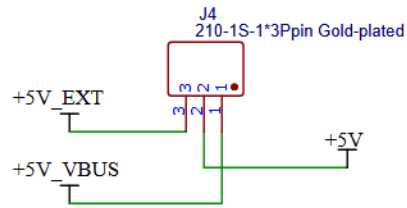
Additionally, the schematic includes resistors that ensure proper biasing and support signal integrity for the serial lines TXD0 (Transmit Data) and RXD0 (Receive Data). These resistors help maintain the correct voltage levels and currents, enhancing the quality of the electrical signals by filtering out noise and preventing signal degradation. This setup ensures accurate and reliable data communication, which is vital for the effective operation of the AzulTank system.

6.2.6 Power Supply Section

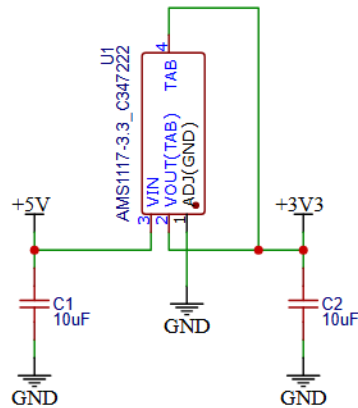
In Figure 6.9, the Power Supply section is crucial for ensuring that the ESP32 and its peripherals receive the correct voltage levels necessary for stable and reliable operation. This section of the schematic is designed to manage the power distribution efficiently, allowing the system to function seamlessly whether powered by USB or an external power source.

The power supply configuration allows for selection between USB power and external power, providing flexibility in how the AzulTank can be powered. This flexibility is essential for various use cases, whether the system is being programmed and debugged via a USB connection or running independently with an external power supply.

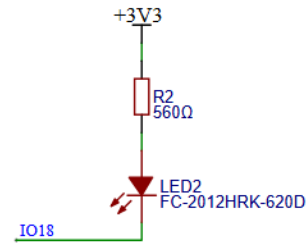
POWER INPUT SELECTION



5V to 3.3V



USER LED



POWER LED

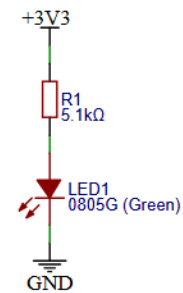


Figure 6.7: Schematics of Power Supply section

Additionally, the schematic includes power LEDs that serve as indicators of the power input and voltage regulation status. These LEDs provide immediate visual feedback on whether the system is correctly powered and whether the voltage is being regulated properly. The power LEDs help in diagnosing power-related issues quickly, ensuring that any problems can be fixed on time.

6.2.7 User and Reset Buttons

In Figure 6.10, the BOOT/User Button and the Reset Button sections are implemented to manage the user interaction and the reset functionality of the ESP32. These buttons provide critical hardware-level control for programming, debugging, and resetting the microcontroller, which enhances the reliability and usability of the system.

The BOOT/User Button circuit includes a momentary push button (SW10), a pull-up resistor (R9), and a decoupling capacitor (C13). The pull-up resistor ensures the pin remains in a defined state when the button is not pressed, while the capacitor debounces the button signal by filtering out noise and preventing false triggering. This setup allows for a reliable signal to the ESP32, particularly when initiating programming or user-defined tasks.

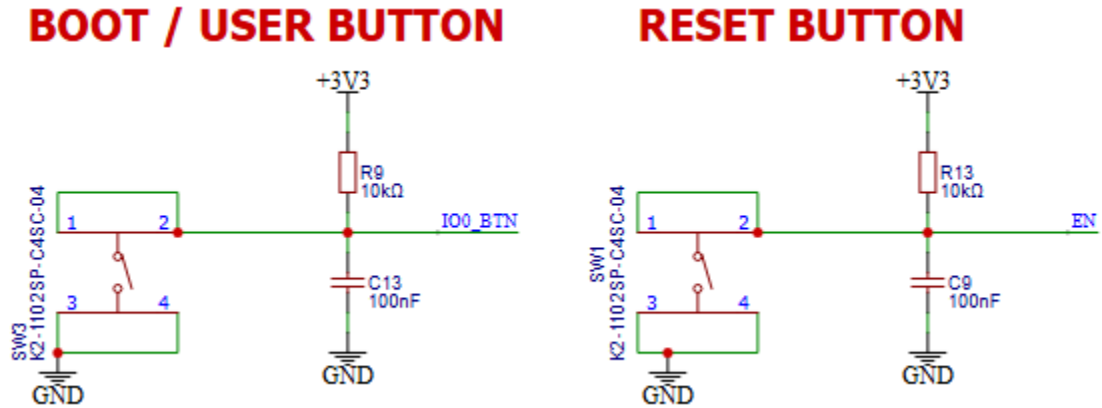


Figure 6.8: Schematics of buttons

Similarly, the Reset Button circuit consists of a momentary push button (SW11), a pull-up resistor (R13), and a decoupling capacitor (C9). This circuit allows the user to reset the ESP32 manually. The pull-up resistor maintains the EN (Enable) pin in a default high state, while the capacitor filters noise to ensure a clean and effective reset pulse when the button is pressed.

These button circuits are essential for interacting with the ESP32, providing both user-friendly operation and stable performance under various conditions, making them a vital part of the AzulTank system.

6.2.8 Stepper Motor Controller

In Figure 6.11, the Stepper Controller circuit is designed to manage the precise control of the stepper motor used in the AzulTank system. This subsystem relies on the DRV8833PWP motor driver (U9) to interface the ESP32 microcontroller with the stepper motor, providing the necessary power and signal modulation for smooth and reliable operation.

The motor driver features inputs (AIN1, AIN2, BIN1, BIN2) connected to the ESP32 GPIO pins (IO26, IO14, IO12, and IO13) to control the direction and speed of the motor. The motor's outputs (AOUT1, AOUT2, BOUT1, BOUT2) provide the appropriate current and voltage levels to the stepper motor windings.

The DRV8833PWP motor driver also features fault reporting (nFault) and sleep mode (nSleep) capabilities, improving the system's robustness by detecting errors and reducing power consumption when the motor is not in use.

STEPPER CONTROLLER

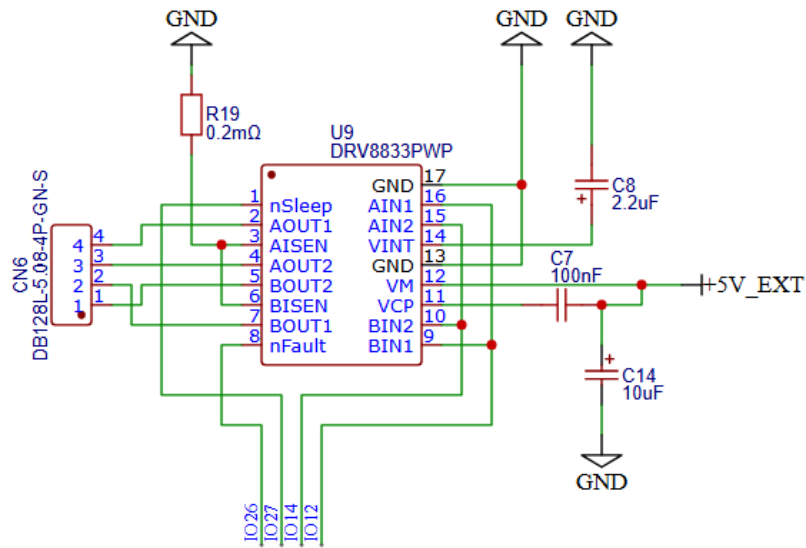


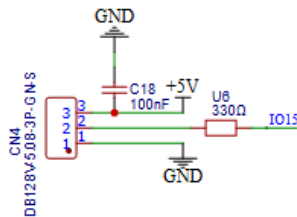
Figure 6.9: Schematics of Stepper Motor Controller

This setup provides precise control over the stepper motor, enabling the AzulTank system to perform accurate tasks such as pH tablet dispensing, which requires fine movement resolution and reliability.

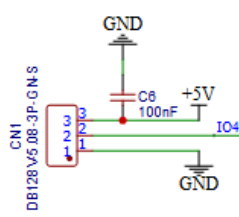
6.2.9 Terminals

The Terminals section of the schematic connects the core subsystems of the AzulTank project to the ESP32 microcontroller and the power supply. These terminals provide interfaces for critical components such as sensors, actuators, and power regulation. Each terminal ensures that signals and power are delivered reliably while minimizing noise and interference.

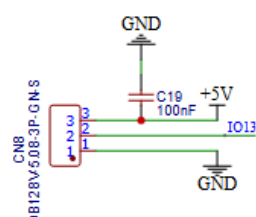
LED LIGHTS



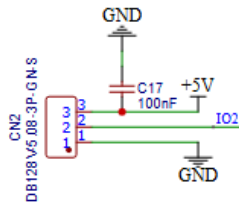
TURBIDITY SENSOR



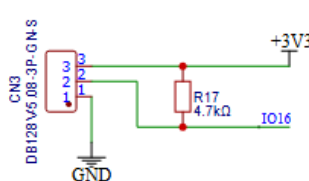
SERVO MOTOR



pH SENSOR



TEMPERATURE SENSOR



REGULATOR

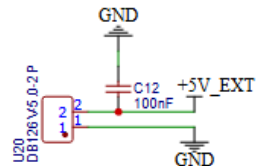


Figure 6.10: Schematics of Terminals section

6.2.9.1 Led Lights Terminal

The LED Lights terminal includes a current-limiting resistor (330Ω) and a decoupling capacitor (C18, 100nF) to stabilize the power supply. This ensures a consistent and flicker-free operation of the NeoPixel LED strip, which is controlled via GPIO pin IO2.

6.2.9.2 Turbidity Sensor Terminal

The Turbidity Sensor terminal is equipped with a decoupling capacitor (C8, 100nF) to filter noise from the power supply. The sensor is powered by the 5V rail and sends data to GPIO pin IO4. This setup ensures accurate water clarity measurements.

6.2.9.3 Servo Motor Terminal

The Servo Motor terminal includes a decoupling capacitor (C19, 100nF) to stabilize power delivery, preventing voltage fluctuations caused by motor operation. The servo motor is powered by the 5V rail and controlled via GPIO pin IO13, enabling precise food dispensing tasks.

6.2.9.4 Ph Sensor Terminal

The pH Sensor terminal incorporates a decoupling capacitor (C17, 100nF) for stable power delivery, minimizing noise in pH readings. The sensor is powered by the 5V rail and connected to GPIO pin IO2, ensuring reliable communication with the microcontroller.

6.2.9.5 Temperature Sensor Terminal

The Temperature Sensor terminal connects the DS18B20 temperature sensor, using a pull-up resistor (R17, 4.7kΩ) to ensure proper signal integrity on the one-wire communication line. The sensor operates on the 3.3V rail and communicates via GPIO pin IO16, providing accurate temperature readings.

6.2.9.6 Regulator Terminal

The Regulator terminal features a decoupling capacitor (C12, 100nF) to smooth power delivery from the 5V_EXT supply. This regulator ensures that the connected components receive a stable 5V supply, critical for the system's overall performance.

By organizing the connections through dedicated terminals, this design ensures a modular and robust system architecture, simplifying assembly, debugging, and maintenance of the AzulTank project.

The ESP32 module acts as the central controller, interfacing with sensors and actuators through its GPIO pins. The connectors expand the I/O capabilities and provide easy access to power and data lines. The USB connector and USB to

serial sections facilitate programming and debugging by allowing direct communication with a computer. Serial signals handling ensures that data communication is reliable and noise-free. Power input selection and regulation guarantee that all components receive the appropriate power levels, maintaining system stability and performance.

6.3 Turbidity Sensor Board Schematics

The turbidity sensor board processes and monitors water turbidity levels using the DFRobot SEN0189 Turbidity Sensor. The schematic integrates components such as operational amplifiers, resistors, capacitors, and LEDs to condition and visualize the turbidity signal.

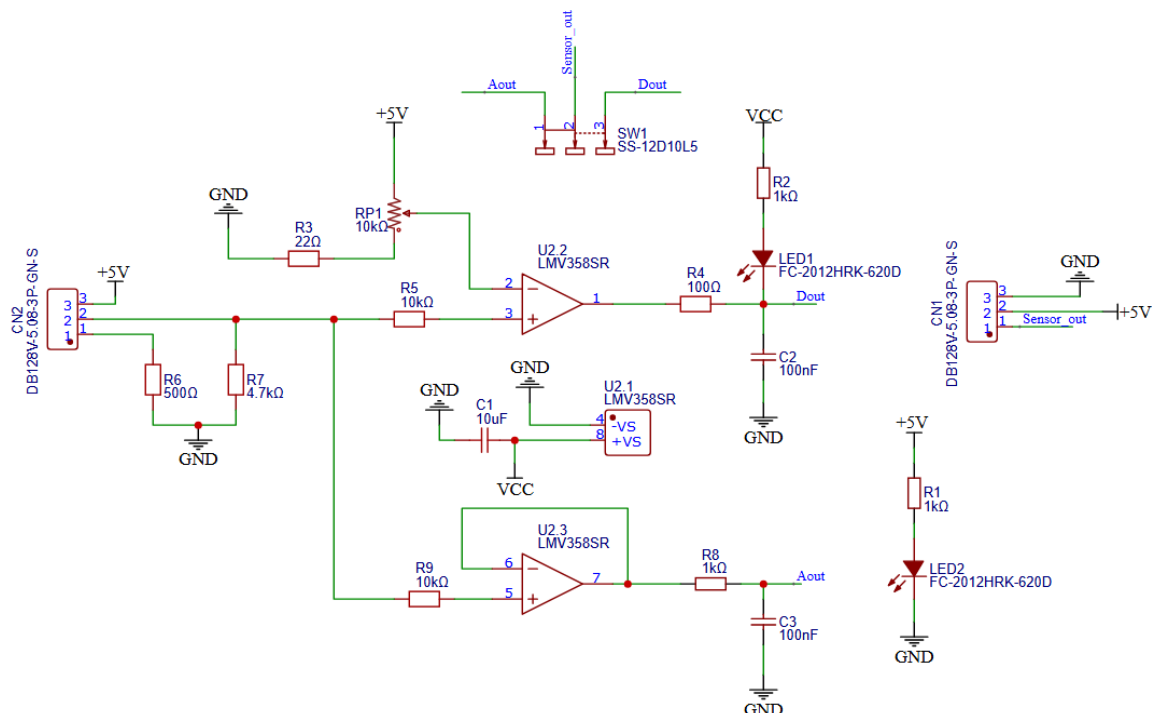


Figure 6.11: Schematics of Turbidity board

The board includes two connectors, CN1 and CN2, which provide +5V power, ground, and the analog output from the sensor. The sensor signal, labeled as Sensor_out, is scaled using a voltage divider comprising resistors R6 and R7. A potentiometer (RP1) allows for manual calibration of the reference voltage used to compare the sensor's output.

Signal conditioning is achieved through a dual operational amplifier (U2, LMV358SR). The first amplifier (U2.2) functions as a comparator, comparing the sensor's conditioned signal to the reference voltage. When the signal exceeds the threshold, the comparator outputs a high signal, activating an indicator LED (LED1) connected to the digital output (Dout). The second amplifier (U2.3) serves as a buffer, stabilizing and amplifying the analog output (Aout) for accurate monitoring or further processing. LED2, connected to Aout, provides a visual

indication of the analog signal strength, which corresponds to the water's turbidity level.

To ensure signal stability and noise reduction, decoupling capacitors (C1, C2, and C3) are strategically placed in the circuit. Current-limiting resistors (R2, R4, R8, and R1) protect the LEDs and maintain consistent operation. The circuit produces two types of outputs: a digital output (Dout), which provides a binary high/low signal when the turbidity exceeds the set threshold, and an analog output (Aout), which offers a continuous signal proportional to the turbidity level. This design allows real-time turbidity monitoring, making it an essential component of the AzulTank project for maintaining optimal aquatic conditions. The design is based on the specifications outlined in the DFRobot Turbidity Sensor documentation.

6.4 pH Sensor Board Schematics

The pH sensor board processes signals from a pH probe and outputs a stable, conditioned signal for monitoring water quality. The design uses operational amplifiers, precision resistors, and capacitors to amplify and filter the weak signals from the pH electrode.

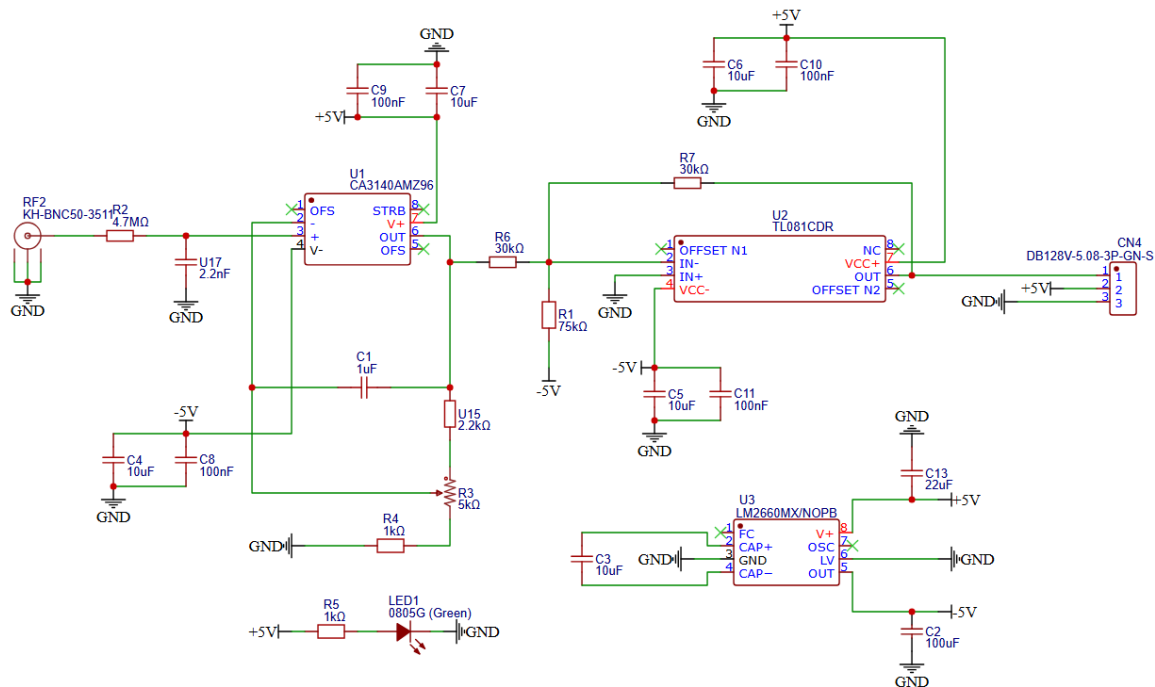


Figure 6.12: Schematics of pH board

The pH probe, connected via RF2 (Coaxial Connector) generates a small voltage proportional to the pH level of the water. This signal is stabilized and filtered by components such as R2 and U17 to reduce noise and ensure accuracy.

The signal enters the first operational amplifier (U1), which provides high input impedance and initial signal amplification. Decoupling capacitors (C9 and C7) are placed near the power supply pins of U1 to minimize noise. The output is further processed by U2, another operational amplifier that fine-tunes the signal and offsets it for proper scaling. Resistors R6, R7, and R1 determine the gain and offset configuration of the amplifiers, while capacitors such as C1 and C5 help stabilize the feedback loop and filter out unwanted high-frequency noise.

The board also includes an LED indicator (LED1), driven by resistors R4 and R5, which provides a visual signal when the circuit is active. The power supply for the board is managed by a charge pump voltage inverter (U3), which generates a -5V supply from the +5V input. This dual-rail power system ensures proper operation of the amplifiers, with capacitors C2, C3, and C13 providing stability and noise filtering.

The output signal is accessible via connector CN4, which provides a reliable interface for connecting the pH signal to the MCU board. This robust design ensures accurate and stable pH measurements, making it a critical part of the AzulTank project for maintaining optimal water conditions. The design is based on the specifications outlined in the DFRobot Gravity Analog pH Sensor V1 documentation.

Chapter 7 - Software Design

This section discussed the functionality of the application design providing a comprehensive overview of its capabilities. To cover the full scope of the application and its capabilities, the section itself has been separated into subsections based on the app's key functions and components. Each of this subsection went into detail of individual features, user interactions, and its design considerations associated with each aspect of the application. Overall, we aimed to present a clear and structured understanding of how the app enabled efficient and successful operation of the smart aquarium system.

7.1 Landing Page

The landing page was created to welcome the user to their smart aquarium's current status along with its main functionalities. At the bottom of the screen, the user encountered a control panel with six buttons that are dedicated to quick actions. These are for feeding, dispensing pH solution, accessing schedules, opening an LED colors menu, and if stretch goals are implemented, LED settings shortcut and activating a camera livestream. In the top right corner, there is a settings button dedicated to customizing the smart tank's features.

The top of the landing page focused on displaying the status of the aquarium's water conditions. These conditions are the turbidity level, which is a measurement of the water's clarity, water temperature and, most importantly, its

pH level. These three areas send out notifications if they go below or above a user-set threshold. In addition, each of the statuses represented the severity of the issue not only via metrics but also through colors. Green for when the water's quality is within appropriate conditions while red indicates immediate attention is required for the water quality has gone out of the user-set threshold.

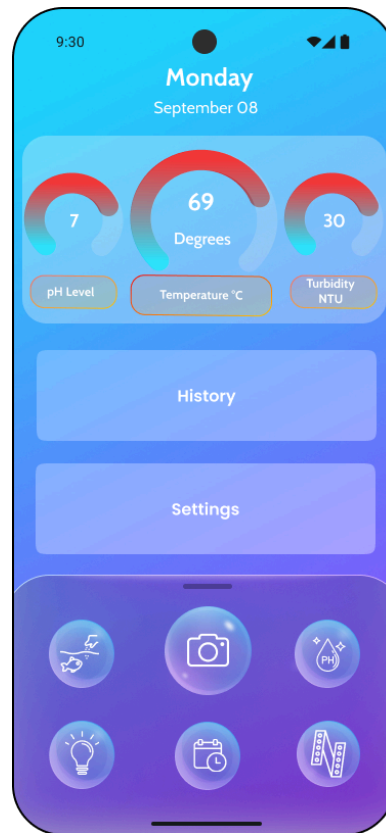


Figure 7.1: Landing Page

As for the control panel, on the top left, the food dispenser button allowed the user to instantly dispense fish food, which displayed a message once the action is in motion. The amount to be dispensed can be configured on the settings menu, which will be discussed later. Next, there's the camera button if the livestream stretch goal is implemented. This button simply activated a live stream of the aquarium, allowing the owner to monitor the fish even while away from home. On the top right, the user can instantly dispense the pH solution if needed, and a message is displayed once the action is in motion. Moving to the bottom left, this button is used to open a menu where different colors for the tank's LEDs can be set as well as configuring the colors for the warning system indicating when the aquarium requires attention. Next, there's the schedules button, which, when pressed, displays a menu that allows the user to configure feeding days/times and amounts of food to be dispensed. On the bottom right, the user can press this button to instantly navigate to the LED settings page for quick access to adjust the LED colors and brightness. However, this is also a stretch goal.

The front-end design has been developed through React Native in combination with Figma. Figma focused on the interface design, prototyping and graphics while React will set the code's structure. Finally, Figure 7.1 shows a figure of the application's landing page.

7.2 Settings Page

The settings page encompassed the configuration of individual features. A tab was added dedicated to each feature as required. This tab covers water conditions for turbidity, temperature and pH level.

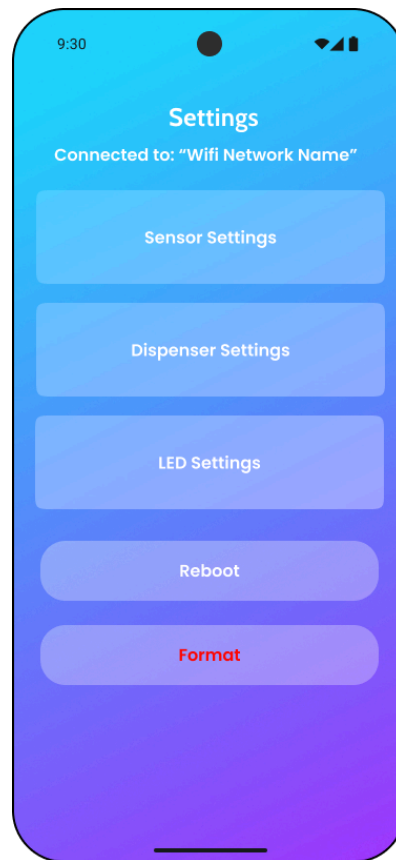


Figure 7.2: Settings Page

Under the title "Settings" shows which network the app is connected to. The first tab allowed the user to configure the sensors or water quality thresholds, which triggered a warning. In addition, the user can click on each water condition and LED color indicated the warning system uses to represent the severity of the situation when attention is required. The second next tab contained the configuration of the dispensers. Here, the user can set the amount of food to be dispensed when the feeding button is pressed, this goes both the manual and scheduled dispenses. The LED's menu, to which the user is redirected when pressing the quick action LED button, as discussed earlier. This allowed the user to change the color of the LED as well as allowed the user to adjust its

brightness. The Reboot button allowed the user to send in a command to the MCU to reboot the MCU itself. Lastly the format button allowed the user to send in a command to erase the entire file system stored in the MCU's flash memory. Figure 7.2 shows a figure of the application's settings page.

7.3 Schedule

The schedule page allowed the users to set and manage feeding schedules and LED schedules for the fish and the overall tank. This made it a critical component for the smart aquarium application. This is because the feeder functionality guarantees that the aquatic life is fed at regular intervals, which improves their health and overall well-being. The LED schedule was simply to turn on/off the LED lights whenever the user wanted, which provided a customizable lighting setup to enhance the aquarium's aesthetics and maintain a consistent day-night cycle for the aquatic environment. The overall design for this page mainly focuses on its usability, flexibility, and clarity in which allows the users to swiftly set up and change feeding schedules according to their specific needs and requirements. The photo shows the overall user interface of the application and its functionalities that we included into our project.

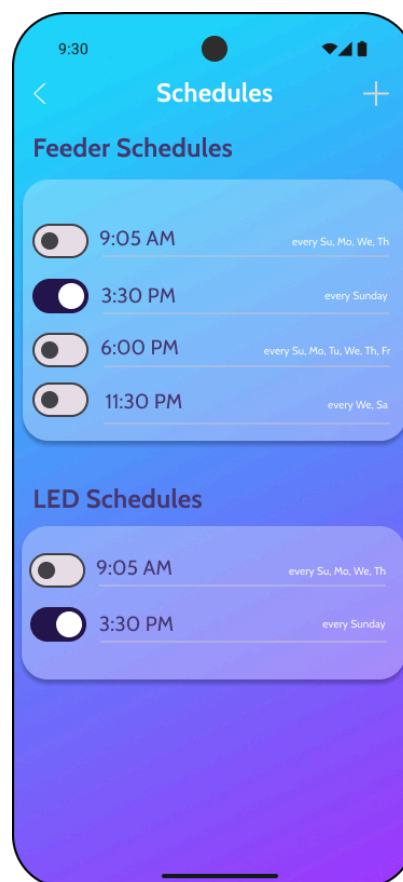


Figure 7.3: Schedule Page

The provided image illustrates the user interface of the application and its functionalities. The schedule page displays a list of schedules. This is separated between Feeder and LED, each with a toggle switch to enable or disable the schedule. Users can easily see the scheduled feeding and LED times, the frequency of intervals, and the days of the week when the schedule will occur.

7.4 Add New Schedule

The "Add New Schedule " is added to the Schedule Page and is designed to overall simplify the process of setting up automated feeding times for the aquarium. This is accessible through a big plus (+) button icon, making it immediately recognizable and simple to use. This button would allow the users to instantly navigate to the scheduling interface.

This feature also allowed the users to define a specific start for feeding via a user-friendly interface which ensures accuracy and ease of use. The time picker itself included its own feature where it allowed the user to scroll through limited hour (up to 12) and minute (up to 60) options. This made it easy to select the precise time for feeding. This also includes an AM/PM Toggle to allow the user to switch morning and afternoon or evening times.

Furthermore, we also aimed to provide versatile repetition choices in which it allowed the user to plan feedings to occur daily or specified days of the week and specific time duration. This medication takes into account the common if not all sorts of aquatic life that will possibly reside in the aquarium. This recognized that various species may have varied nutritional requirements and feeding frequencies.

By default, all fields are set to "None" and the Repeat Time are set to current time prompting users to actively select the desired times and ensuring no accidental schedules are created. Device feed is also then set to LED by default.

For example, some fish need daily feeding or might require multiple feedings per day, while others might need to be fed less frequently. Some users or hobbyists may want to fast their fish for a day or two a week to help them clear their digestive systems. So having this customization in our app has guaranteed that feeding schedules can be adapted to the aquarium's specific demands. This gives an overall stable and flexible solution for keeping aquatic life healthy and well-fed.

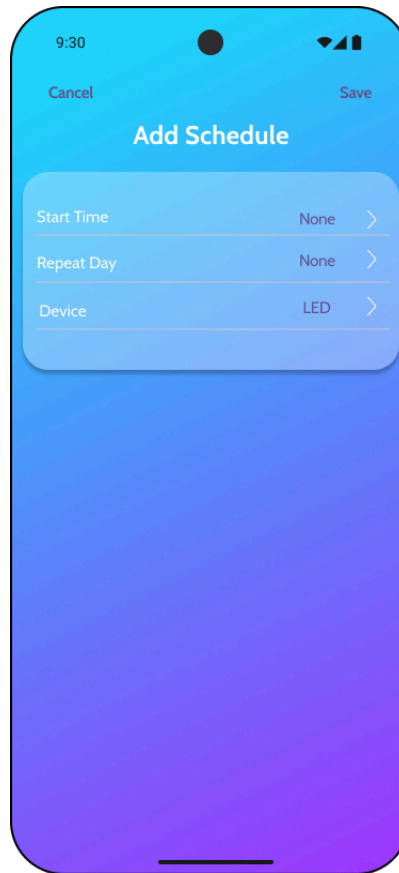


Figure 7.4: Add Page

To further improve the user experience, the time picker interface included “Save” and “Cancel” buttons. The “Save” button confirmed the specified feeding time and repetition settings. The “Cancel” button on the other hand allowed the user to exit the scheduling page without changing anything. Overall, these buttons were clearly visible and can be easily accessible ensuring that the user was able to efficiently and confidently set or cancel their feeding schedules.

7.5 Edit Schedule

We added an “Edit Schedule” feature for the app. This feature allowed the user the ability to manage existing feeding with ease. Each existing feeding schedule entry as an on/off toggle to allow the user to activate/deactivate each schedule as needed without actually deleting or creating a new schedule.

The user had the ability to click on an existing schedule to change the time and repetition settings of an existing schedule. Similar features from the “Add Schedule Feature” such as the time picker for the time, repetition options for day as well as the Save and Cancel buttons will be incorporated into this feature to ensure consistency and ease of use for the user.

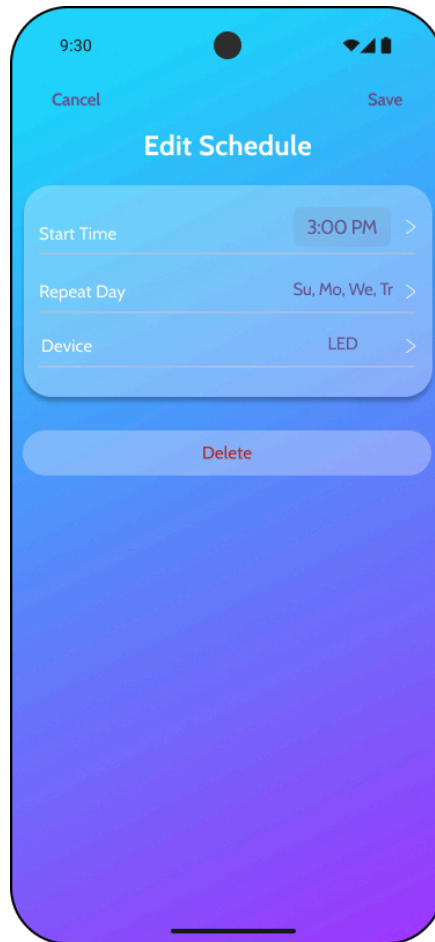


Figure 7.5: Edit Page 1

To add convenience and ease for the user, a delete button is also added and would be readily available to remove any existing schedule that is no longer needed or required by the user. This keeps the overall schedule page clean and up to date. This certain feature gives the user full control over their feeding schedules while keeping the flexibility and adaptability required for an optimal aquarium environment.

Another thing to highlight for this feature is that it allowed the user to set a feeding plan for a certain day and time as well as eliminated the need for repeated events. Users can simply select a single day and time for feeding which allows the feeding routine to be as simple or detailed as needed.

Lastly, we also added the delete button which allowed the user to delete that certain schedule without affecting the other existing schedules. This ensured seamless management and flexibility for updating or reorganizing according to the user's preferences.

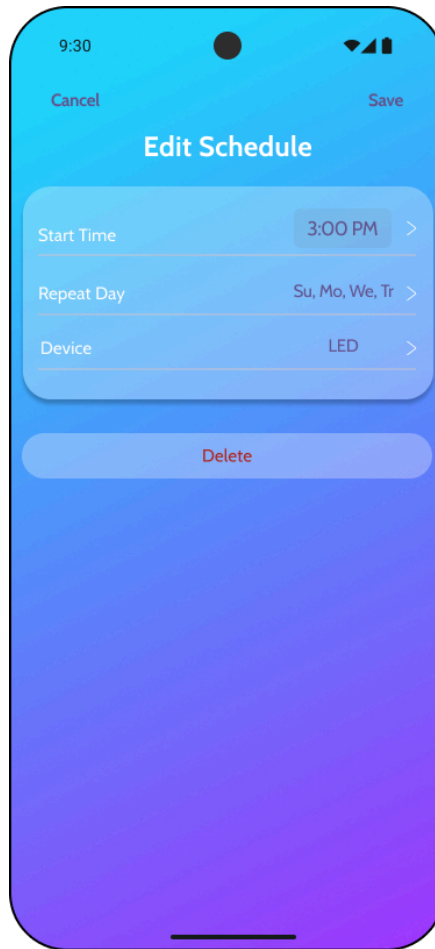


Figure 7.6: Edit Page 2

It ensured that the overall schedule could be adjusted to accommodate changes in the aquarium environment. Similar to the “Add New Schedule,” this feature and its buttons were designed clearly to be visible and can be easily accessible which ensured that the users were able to efficiently and confidently edit or delete an existing feeding schedule.

7.6 Start Time

Within the scheduling options, the user can set the start for their programmed feeding schedule. This feature showcased a user-friendly interface that allowed users to easily modify the starting hour for the feeding to begin. These pages are seamlessly integrated into both the “Add Schedule” and “Edit Schedule” workflows, ensuring a consistent and intuitive user experience.

When creating a new schedule, the user is guided through a straightforward process of assigning start time in which it is set to the current time by default. The interface provided clear prompts and input fields, enabling users to specify the exact times they want the feeding to occur. This step-by-step approach

ensures that even users who are not tech-savvy can easily configure their feeding schedules without confusion.

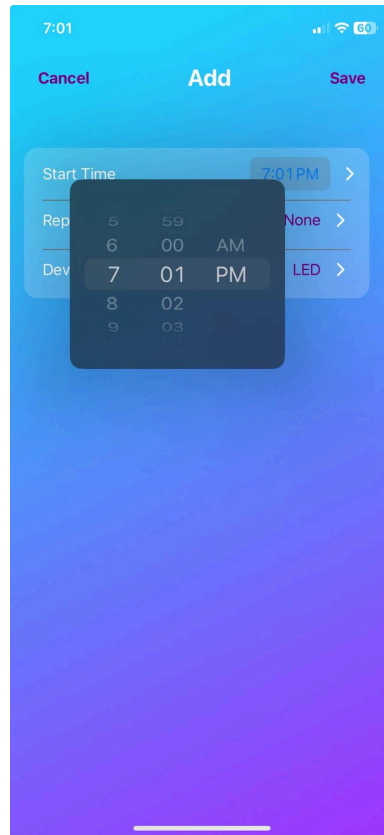


Figure 7.7: Start Time

Similarly, when editing an existing schedule, users can access the same interface to modify the start and end times. This consistency between adding and editing schedules helps users feel comfortable and familiar with the process, reducing the learning curve and potential for errors. Users can adjust their feeding schedules as their needs change, ensuring that the system remains flexible and adaptable to different situations.

7.7 Repeat Day

Much like the start and end times in the previous section, we have also designed a page for the user to set the days of the week that they want to repeat their hours depending on the device they chose. This interface can be seen when creating a new schedule or editing a previously created one as well.

The users are presented with a clear and user-friendly interface that allows them to select specific days by including simple toggles or checkmarks for each day of the week, from Monday to Sunday. This feature is useful for users who had consistent schedule patterns throughout the week, as it allowed them to automate the process. This kept their schedules maintained without requiring

them to manually input the information each day and ensured their fish remained well-fed and LED lights to turn/off at the desired day.

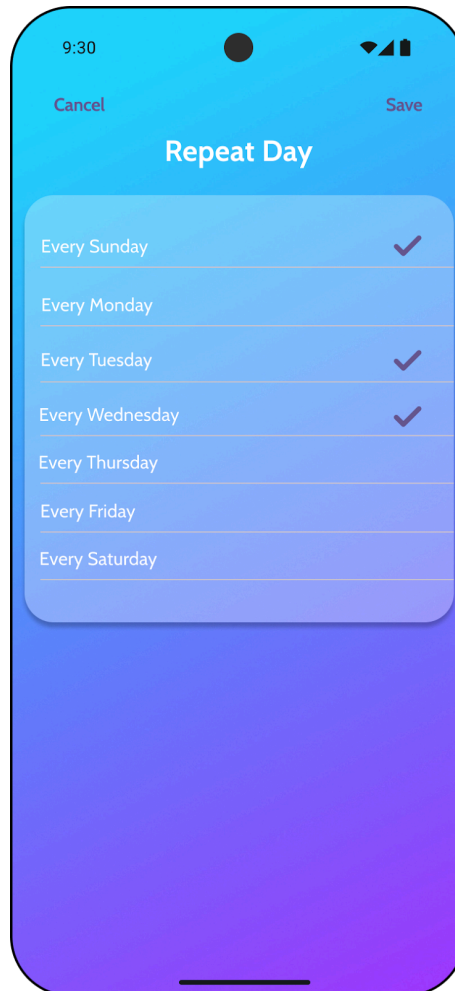


Figure 7.8: Repeat Day

7.8 Activity Diagram

An activity diagram is a simple way to map out the flow of a process in a system. It's a type of diagram used in software development to show how tasks are performed and connected. You'll see activities represented as rounded rectangles with arrows showing the direction of the process. Decisions are marked with diamond shapes, and you can even show things happening at the same time with bars to indicate parallel actions. These diagrams are helpful because they make it easy to spot problems, streamline workflows, and explain how things work to people who might not be technical.

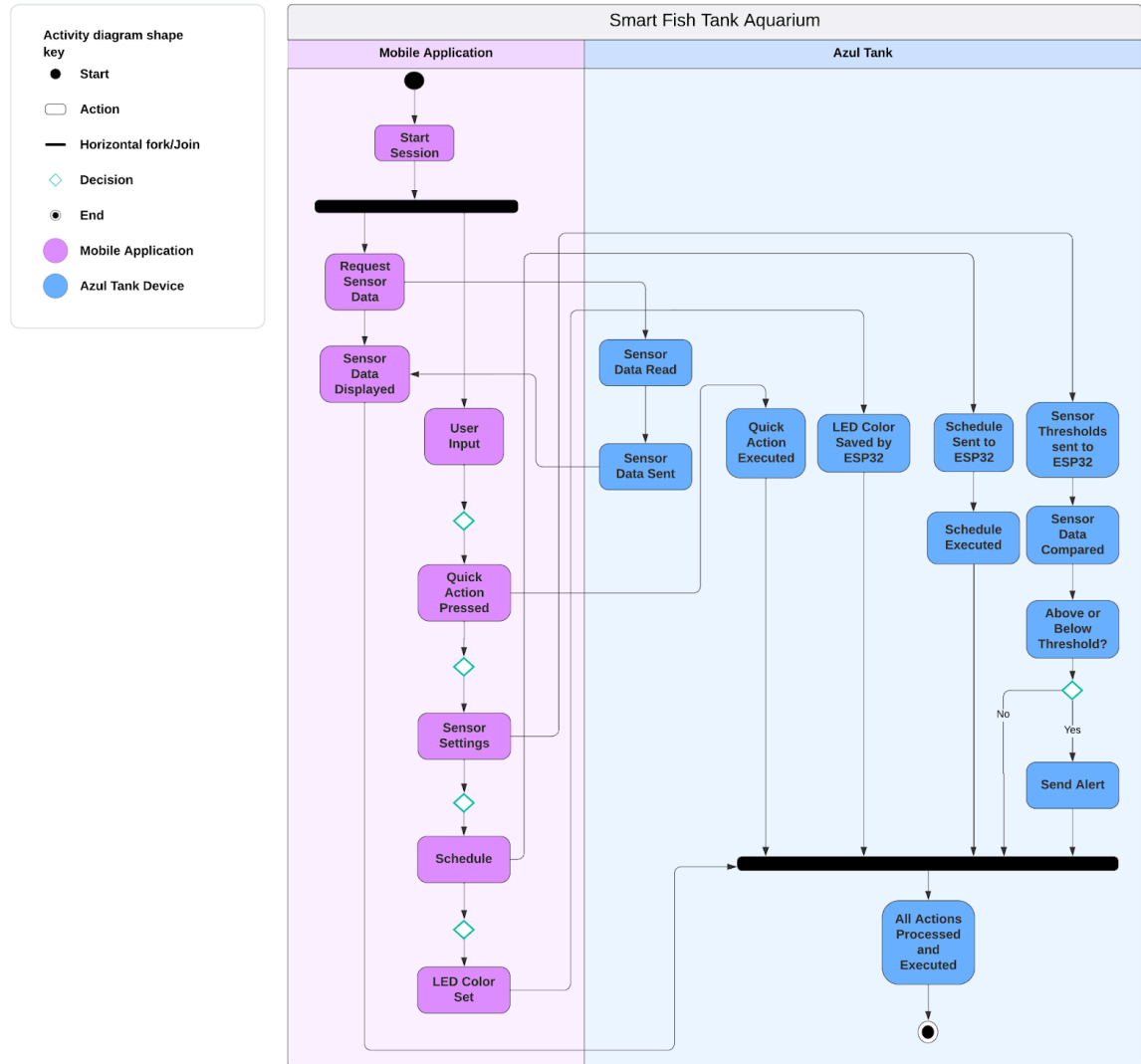


Figure 7.9: Activity Diagram

7.9 Use Case Diagram

A use case diagram displays the interactions between system components. This emphasizes the roles of various actors and the flow of information. Not only that it ensures that all components operate together harmoniously, but it also clarifies the relationships and dependencies within the system.

The overall diagram serves as a blueprint or a template to understand the system's functional needs and requirements from the user's perspective. This aids proper communication and guidance for the development process and ensures that the final implementation satisfies the intended needs and expectations. Overall, it helps discover possible areas for development and optimization which results in a more robust and user-friendly system.

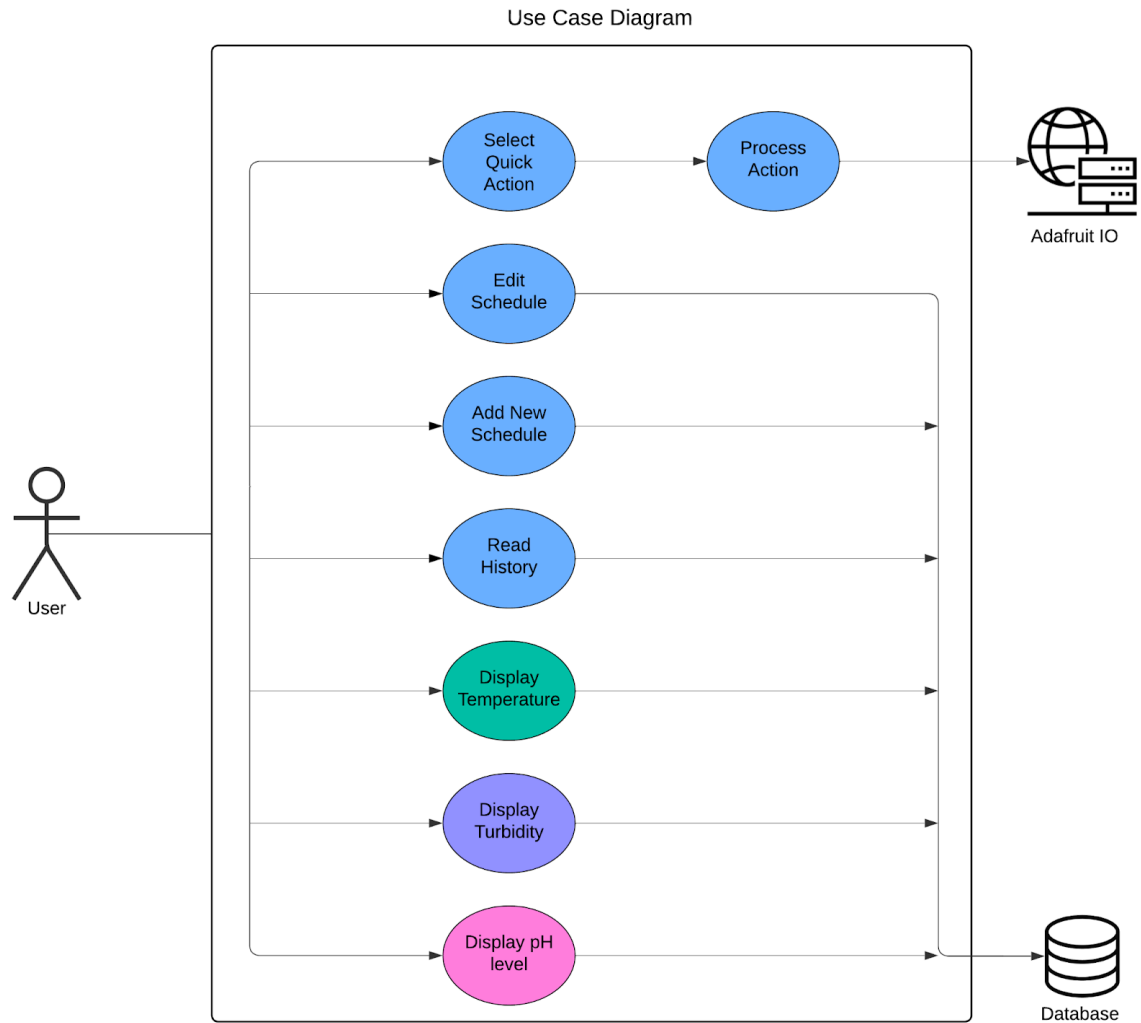


Figure 7.10: Smart Aquarium Use Case Diagram

When utilizing the AzulTank, the use case diagram highlights the essential interactions between the user, Adafruit IO, and the database. The user can perform several actions, such as selecting quick actions, editing schedules, and reading sensor data, including temperature, turbidity, and pH levels. These actions are processed by the system, ensuring that the appropriate commands are sent to the web server and the database is updated accordingly.

The user initiates actions such as "Select Quick Action" and "Edit Schedule," which are then processed by the system. The "Process Action" node indicates that the system handles these user inputs and communicates with Adafruit IO to send and execute the commands.

This interaction ensures that the user's commands are correctly interpreted and implemented by the system. On the other hand, the "Read Temperature," "Read Turbidity," and "Read pH Level" nodes illustrate how the system continuously

monitors the aquarium's conditions. This sensor data is then transmitted to the database, where it is stored and can be accessed for further analysis and decision-making.

7.10 Class Diagram

A class diagram is a type of static structure diagram in the Unified Modeling Language, also known as UML. It describes the structure of a system by presenting its classes, attributes, methods, and relationships among objects. It's a strong approach to help visualize, design, document and implement guidance. This way, offering a clear and detailed plan that developers can follow for implementation. It's also considered formal documentation as it provides reference for future updates. Additionally, it's helpful for helping stakeholders understand the relationships and dependencies of a system by providing them with a common language.

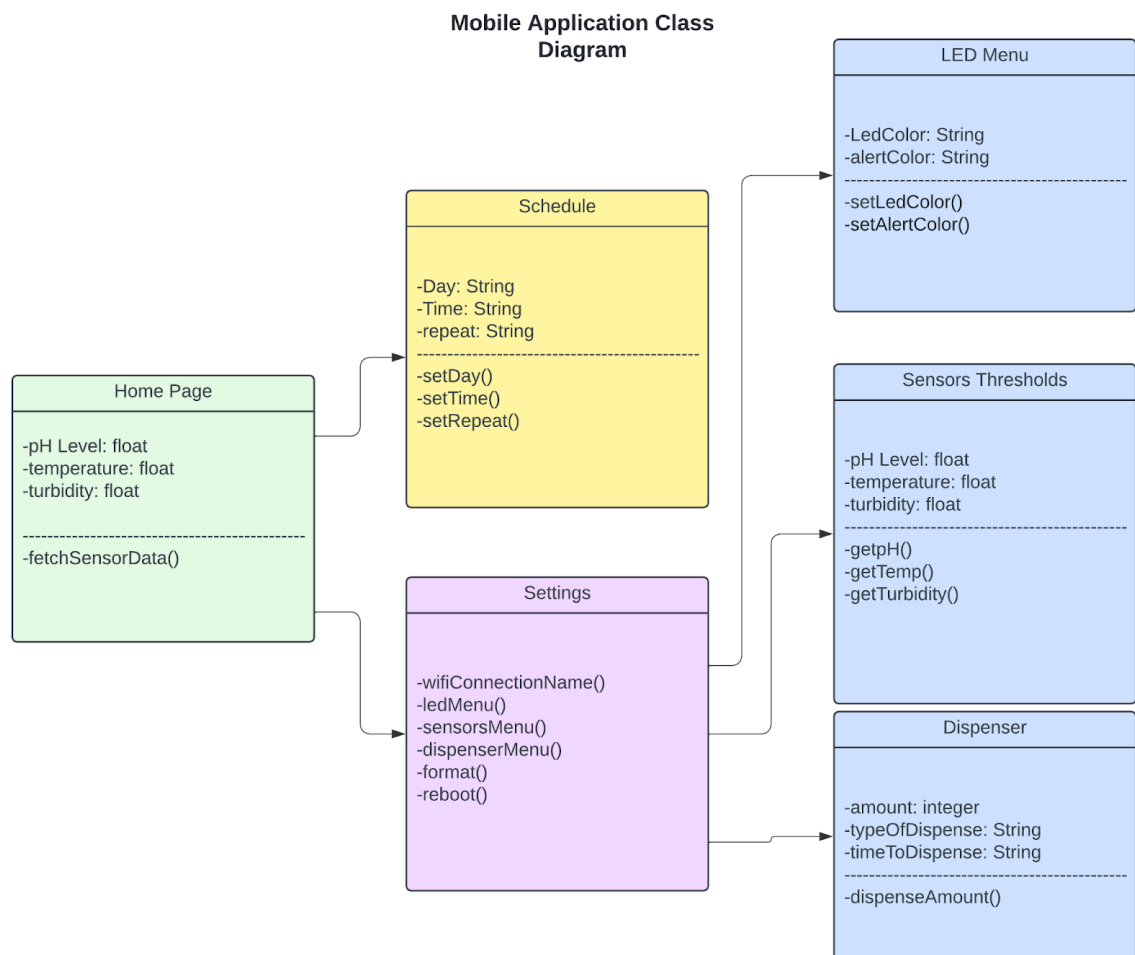


Figure 7.11: Class Diagram

The classes are represented by rectangles and contain the attributes and methods. Classes can be associated with each other and to represent this relationship different types of lines are used to connect them. There are lines for association, inheritance, aggregation, composition and dependency. Figure 7.10 shows a class diagram of the AzulTank's companion application.

Each class is interconnected through relationships that illustrate how they interact and depend on each other. For instance, the Landing Page class interacts with the Sensors class to display current readings, while the Schedule class interacts with the Dispenser class to manage feeding times. These interactions are depicted through lines that represent associations, ensuring that the functionality of the application is cohesive and well-integrated. It serves as an important tool for both the initial development and future enhancements, ensuring that the application remains robust and adaptable to new requirements.

Chapter 8 - System Fabrication

8.1 PCB

For designing our printed circuit boards (PCBs), we utilized EasyEDA, a user-friendly and versatile PCB design platform. This tool allowed us to efficiently create and refine our PCB layouts while ensuring they met the performance and reliability requirements of our AzulTank project. Once the designs were finalized, we partnered with JLCPCB, a trusted PCB manufacturer, to produce our boards with precision and high-quality standards.

Throughout the PCB design process, we made several key decisions to optimize circuit performance and minimize potential issues. A primary focus was maintaining short signal paths to reduce noise and signal degradation. This was especially important for sensitive analog signals and high-speed digital circuits. By keeping these paths as short as possible, we effectively minimized interference and ensured robust and accurate signal transmission.

To further enhance performance, we carefully separated the analog and digital signal paths. This separation prevented digital noise, typically caused by high-frequency switching, from interfering with delicate analog signals. This design strategy was crucial for the reliable operation of sensors and other precision components, enabling accurate data acquisition and processing.

For the Power Supply Unit (PSU) and voltage regulator PCBs, specific enhancements were incorporated to support stable and efficient operation. Power and ground traces were designed with increased width to handle higher current loads and reduce voltage drops, ensuring consistent and reliable power delivery. Thermal vias were strategically placed near high-power components to facilitate effective heat dissipation, preventing overheating and maintaining stable performance under various load conditions.

During the design phase, we utilized EasyEDA's simulation tools to test and refine the PSU and voltage regulator circuits under different voltage inputs. These simulations allowed us to identify and address potential issues, such as inconsistencies in the step-down process that could impact sensitive components. By analyzing and optimizing the designs, we ensured they could handle a broader range of voltage inputs while protecting critical parts of the circuit.

Once the designs were complete, we sent them to JLCPCB for manufacturing. Their precise and reliable production process ensured that our boards were fabricated to meet our specifications. After manufacturing, we conducted additional testing during integration to verify that the PCBs performed as expected within the complete system. This final stage ensured that the boards functioned seamlessly and reliably as part of the AzulTank project. By leveraging EasyEDA for design and JLCPCB for manufacturing, we successfully created high-quality PCBs that meet the performance and reliability requirements of our project.

8.1.1 Regulator PCB Layout Design

The regulator PCB serves as a critical component in our system because it provides a reliable 5V output to power our ESP32 PCB from a 12V DC input supplied from our AC-to-DC wall plug. This step down buck regulator ensures that our ESP32 that manages all our sensors and motors receives the stable voltage, while still being protected from overvoltage damage. Its inclusion between the wall plug and ESP32 is essential to maintaining proper functionality and extending the lifespan of our components, sensors, and motors.

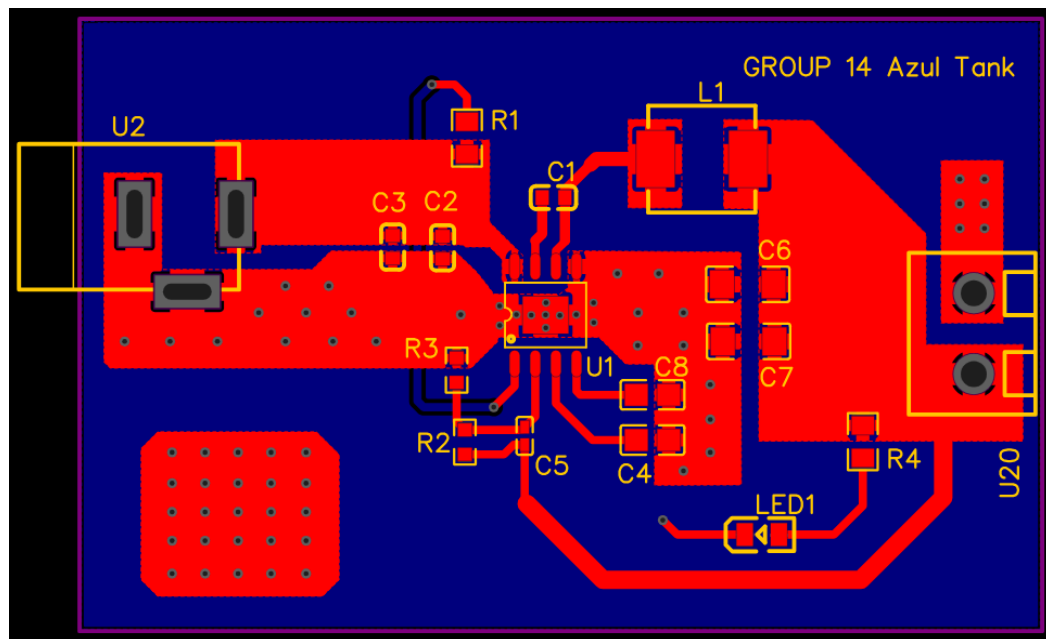


Figure 8.1: PCB Layout Design of Regulator board

The PCB layout of our regulator board was designed towards optimal performance and reliability. The key components like the input capacitors, output capacitors, and feedback resistors are centrally placed around the TPS54528 voltage regulator to ensure effective voltage filtering and stability. The input terminal U2 was placed on the left with the output connector board U20 was placed on the right which allows for efficient power flow across the board. Overall, all the components were strategically placed to allow for minimal noise and maintain the compact layout. On the bottom right side we have the LED indicator with a resistor to provide the visual indication that the regulator is operating properly for a more easy way of monitoring the PCB status. Lastly, the board's traces are designed to handle appropriate current levels with careful attention to grounding and signal integrity for more reliability.

8.1.2 MCU PCB Layout Design

The MCU board serves as the central hub for managing the various sensors and actuators in the AzulTank system. The layout is meticulously designed to accommodate the microcontroller (U4), power management circuits, and interface connectors for peripherals, ensuring robust functionality and minimal interference. The microcontroller is centrally positioned, with neatly routed traces radiating out to ensure short, direct signal paths to peripheral connectors, improving signal integrity and minimizing potential noise.

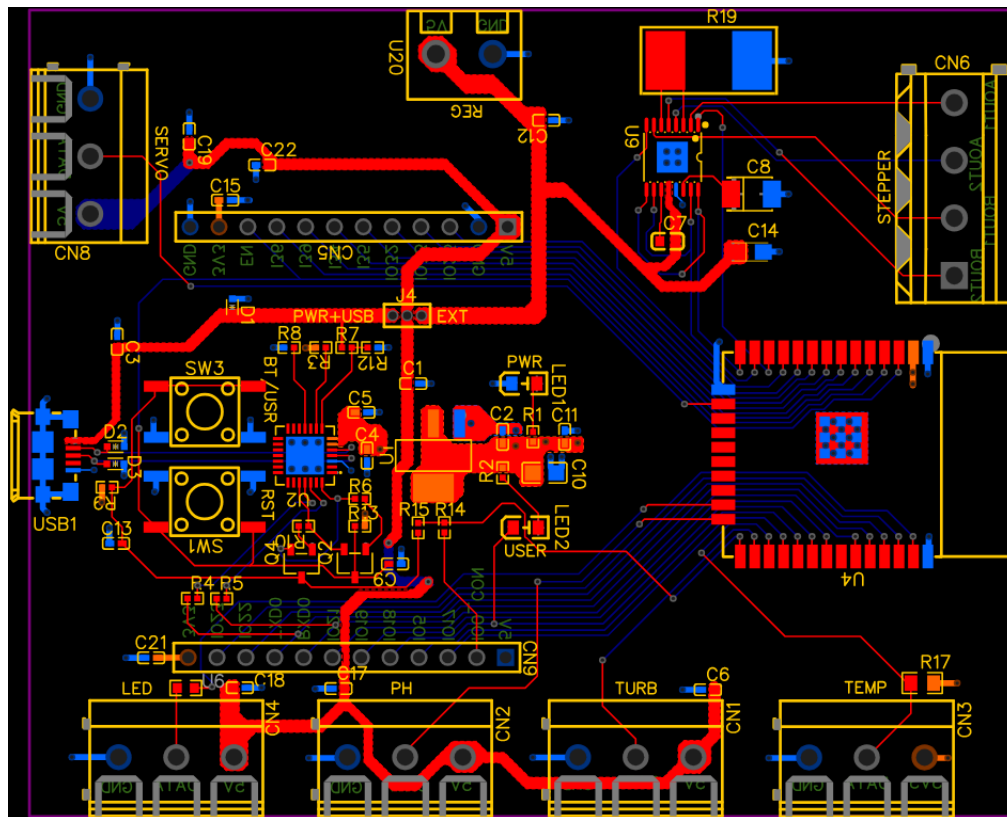


Figure 8.2: PCB Layout Design of MCU board

Connectors for the servo motor (CN8), stepper motor (CN6), and sensors, such as turbidity (CN1), pH (CN2), and temperature (CN3), are strategically placed along the edges of the board for easy access and organized cable management. Power input options are provided via USB (USB1) and an external connector (J4), allowing flexibility in powering the system. The board's power management circuitry, including voltage regulators (U2 and U9), is located near the power inputs and incorporates decoupling capacitors (C2, C8, C14) to stabilize voltage and minimize noise across the board.

The layout includes LED indicators (LED1 and LED2) to provide visual feedback for power status and user-defined functions, making it easier to monitor the system's operation. Components such as switches (SW3, SW5) and user inputs are arranged intuitively near the microcontroller for efficient debugging and interaction. Thick traces are used for power and ground connections to handle higher current loads effectively, while signal traces are kept thin and well-routed to avoid crosstalk and interference.

Decoupling capacitors (C4, C10, C11) are positioned close to the microcontroller and other critical components to filter out high-frequency noise, ensuring stable operation of both digital and analog circuits. The ground plane design is continuous and well-distributed, providing a low-impedance return path for signals and minimizing EMI. Additionally, the layout features thermal relief patterns around power-intensive components, improving heat dissipation and enhancing overall reliability.

This compact yet functional layout ensures the seamless integration of all AzulTank system components while maintaining high performance and reliability. The MCU board's design reflects a careful balance between functionality, manufacturability, and system requirements, making it a key component in managing and controlling the smart fish tank system.

8.1.3 Turbidity Sensor PCB Layout Design

The PCB layout for the turbidity board was designed with a focus on functionality, compactness, and ease of assembly. The layout groups components logically, with the turbidity probe connector placed on one edge of the board to ensure seamless interfacing with the external sensor.

Signal processing components, including the operational amplifier (U2) and associated resistors, are centrally positioned to minimize signal path lengths, ensuring reduced signal degradation and improved reliability. LED indicators (LED1 and LED2) are located near the output section, providing clear visual feedback on the board's operational status.

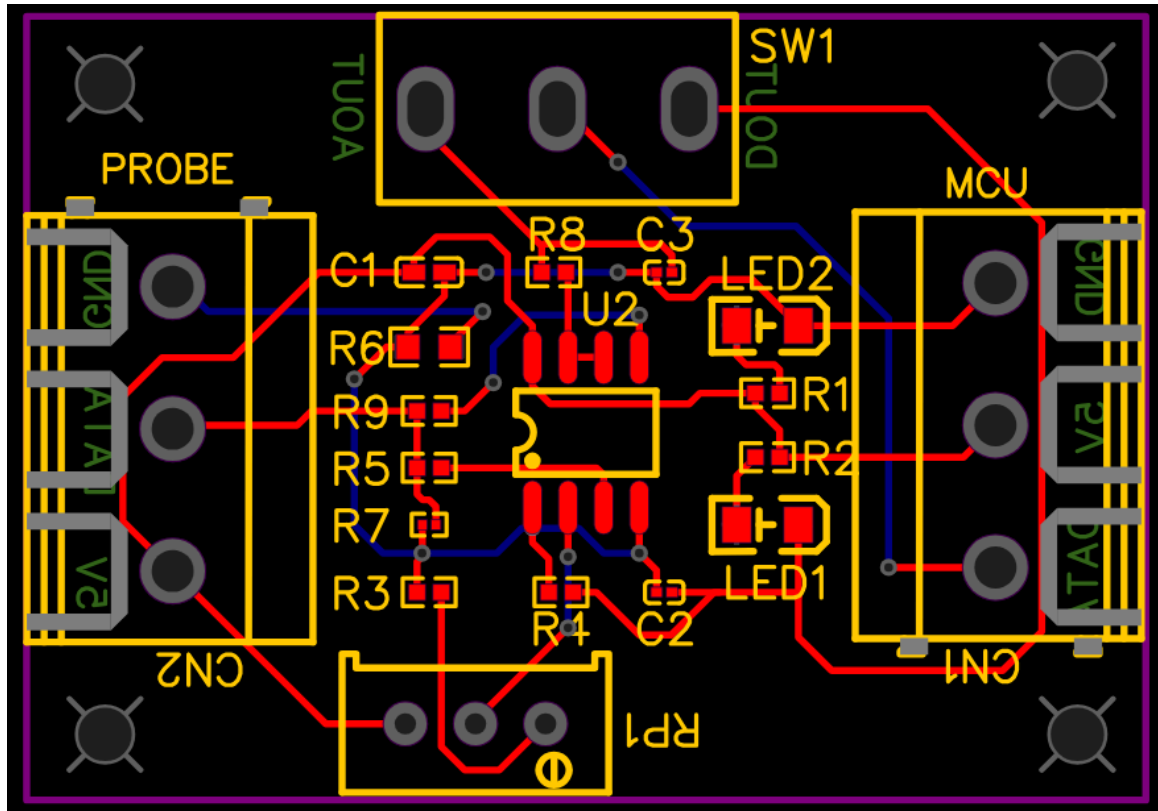


Figure 8.3: PCB Layout Design of Turbidity board

Trace routing was carefully planned to reduce electromagnetic interference (EMI) and crosstalk, with signal traces kept short and direct. The use of distinct trace colors for different PCB layers enhances the clarity of routing and simplifies debugging. Ground and power traces are designed with sufficient width to handle current loads effectively, maintaining stable voltage levels across the circuit. A well-distributed ground plane reduces noise and ensures reliable operation of both the analog and digital sections of the board.

Key interfaces are strategically positioned for convenience and performance. The PROBE connector allows for straightforward connection to the turbidity sensor, while the MCU connector, located on the opposite edge, facilitates easy integration with the microcontroller. The potentiometer (RP1) is placed in an accessible location, allowing for simple calibration without interfering with other components. LED indicators are positioned prominently to clearly signal the digital and analog outputs, making it easy to assess the board's status at a glance.

The layout also optimizes board space while maintaining a clean and organized design. This balance ensures the board is compact yet functional, meeting the performance and reliability requirements of the AzulTank system. By adhering to best practices in PCB design, the turbidity board delivers accurate and stable signal processing essential for real-time water quality monitoring.

8.1.4 pH Sensor PCB Layout Design

The PCB layout for the pH board was designed to ensure precise signal processing, efficient space utilization, and reliable performance.

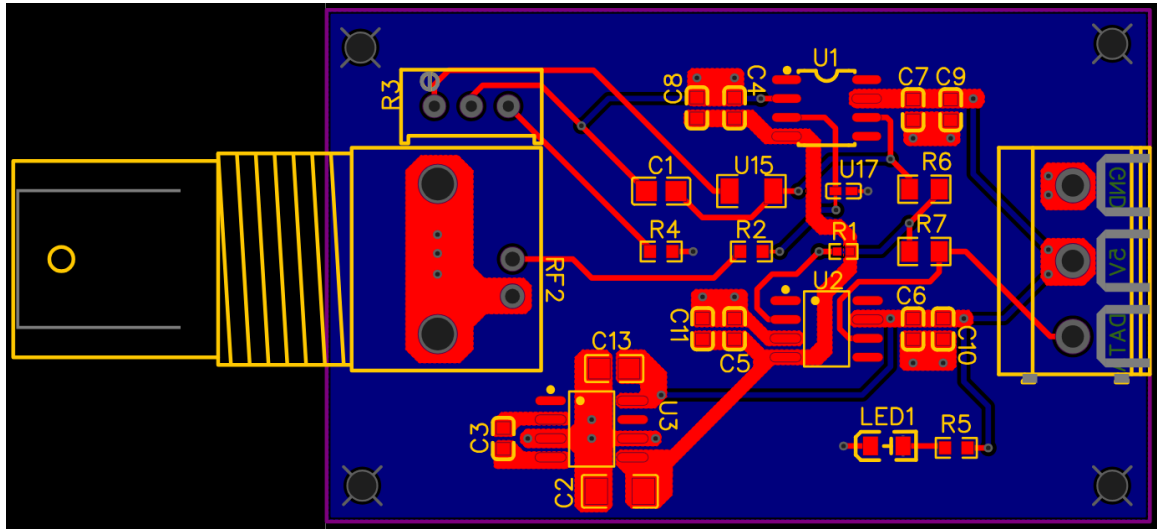


Figure 8.4: PCB Layout Design of pH board

The layout begins with the BNC connector (RF2) on one end, which interfaces with the pH probe and provides the weak analog signal for processing. The signal passes through various conditioning and amplification stages, with the components placed strategically to ensure short signal paths and minimal noise interference. The operational amplifiers, U1 and U2, are centrally positioned, allowing efficient routing of input and output signals while minimizing path length to reduce signal degradation.

Key passive components, such as resistors (R2, R6, R7) and capacitors (C1, C7, C9), are placed close to the amplifiers to ensure stable and accurate signal processing. Decoupling capacitors (C4, C6, C13) are positioned near power pins to minimize noise and stabilize voltage supply, ensuring reliable operation. The layout also includes thermal vias and ground planes that help reduce electromagnetic interference (EMI) and improve overall circuit stability.

The board features clear separation of high-frequency and low-frequency signal paths, preventing interference between the analog and digital sections. LED1, located near the edge, serves as a visual indicator for the circuit's operational status, making it easy to monitor functionality during testing or regular operation. The power management IC (U3) is strategically placed near the power inputs, with sufficient filtering capacitors (C2, C3) to provide clean dual-rail power supply for the operational amplifiers.

Connector CN4 is positioned at the opposite end of the board to provide a convenient interface with the microcontroller, ensuring a clean and efficient signal

output. The compact and organized design maximizes the use of the available PCB space while maintaining clear routing and adequate spacing between components to avoid short circuits or interference.

This layout ensures the pH board operates reliably, processes signals accurately, and integrates seamlessly with the AzulTank system. The design reflects careful consideration of performance, manufacturability, and ease of use, making it a vital part of the water quality monitoring solution.

Chapter 9 - System Testing and Prototyping

The system testing and prototyping are essential stages in the development of any device in production. This section outlines how we tested the hardware and software components of our AzulTank device. We broke it down further to hardware and software system testing and prototyping to ensure that the finished product meets the necessary requirements and performs properly under various conditions. This section will also help us identify issues and resolve them early in the development process which helps us reduce risk of costly and time-consuming fixes later on.

9.1 Hardware Testing

Hardware Testing and Prototyping helps facilitate continuous improvements and ensures that the final product is in high quality or exceeds user expectations. Risk mitigation is accomplished by testing in a variety of scenarios and conditions to ensure the products dependability and robustness.

We started our hardware testing with the microcontroller and the sensors for temperature, pH and turbidity. This is mainly because these are the components that we had on hand. The following below is a list of a system hardware components tested or we plan for prototyping throughout this procedure:

1. Microcontroller
 - a. Tested for proper UART functionality
2. Sensors
 - a. Temperature Sensor
 - i. Tested for accuracy to make sure the value and information matches with the digital thermometer.
 - ii. Tested for compatibility with microcontroller by using Arduino to make sure it shows the temperature on IDE's output terminal.
 - iii. Prototyped with different water conditions (hot/cold/lukewarm)
 - iv. Prototyped sending information over WiFi to another application/device.
 - b. pH Sensor

- i. Tested for accuracy to make sure the value and information matches with the pH strips.
 - ii. Tested for compatibility with the microcontroller by using Arduino to make sure it shows the pH level on the IDE's output terminal.
 - iii. Prototyped with different water conditions.
 - iv. Prototyped sending information over WiFi to another application/device.
 - c. Turbidity Sensor
 - i. Tested for accuracy based on known turbidity standards
 - ii. Tested for compatibility with microcontroller by using Arduino to make sure it shows the turbidity value on the IDE's output terminal.
 - iii. Prototyped with different water conditions (clear, murky and very dirty).
 - iv. Prototyped sending information over WiFi to another application/device.
- 3. Dispensers
 - a. Food Dispenser
 - i. Tested for accuracy to make sure the device dispenses the proper amount of fish food.
 - ii. Prototyped receiving information over WiFi from another application/device.
 - iii. Tested for device response time when information is received from application/device.
 - b. pH Dispenser
 - i. Tested for accuracy to make sure the device dispenses the proper amount of pH.
 - ii. Prototyped receiving information over WiFi from another application/device.
 - iii. Tested for device response time when information is received from application/device.
- 4. Motor
 - a. Tested for device response time when information is received from application/device.
 - b. Prototyped functionality with silicon tube and IV flow regulator to make sure it is stopping the dispensing the correct amount of pH.
- 5. LED lights
 - a. Prototyped receiving information over WIFI from another application/device.
 - b. Tested by measuring the LEDs brightness and responsiveness.
 - c. Microcontroller

9.1.1 Sensors

The sensors in our AzulTank project monitor various environmental parameters such as temperature, pH levels, and turbidity. These components play a vital role as it gathers real-time data and sends this information to the mobile application. This flow of information ensures that the user is always aware of the current conditions within the aquarium. The information received from the sensors allows the user to take appropriate actions to maintain optimal condition and well-being of aquatic life without doing any manual work.

9.1.1.1 Temperature Sensor

Testing temperature sensors are tested in a series of processes to ensure accuracy and reliability. We first calibrated and tested into a known temperature reference such as an ice-cold water, lukewarm water or a boiling water. We then compare it to the digital thermometer to confirm that the sensor's reading is accurate. We do this by connecting the sensor to the microcontroller through appropriate input pins. We programmed the microcontroller to read sensor data at regular intervals.

Once we validated the temperature sensor and its compatibility with the microcontroller, we made sure to set up necessary network configurations to establish network connection. This allowed the microcontroller to transmit the sensor data to the mobile application.

Lastly, the mobile application then processes the incoming data and displays the received data in a user-friendly format for better reading and understanding.

9.1.1.2 pH Sensor

The pH sensor is also tested in a series of processes similar to the temperature sensor to ensure accuracy and reliability. We first connected the sensor to the microcontroller through its designated input pins for accurate data transfer. We then calibrated the components and wrote a program using a standard buffer solution, typically at pH 4.0 and 7.0 we obtained from the kit.

After the calibration stage, we verified the sensor's accuracy by testing it in different known pH solutions. We then compared the sensor readings with pH strips bought to confirm that the calibration was successful.

Once we validated the pH sensor's accuracy and compatibility with the microcontroller, we ensured to set up the necessary network configuration to establish network connections that allowed the microcontroller to transmit sensor data to the mobile application. This enabled the mobile application to process the incoming data received and displayed it in a user-friendly format for better reading and understanding.

9.1.1.3 Turbidity Sensor

Testing the turbidity sensor has already had the similar process of testing with the temperature and pH sensor to ensure accuracy and reliability. We integrated it by connecting the component to the microcontroller through its designated input pins for the data transfer. We then calibrated it with known turbidity standards.

Once we validated the turbidity sensor and its compatibility with the microcontroller, we then ensured to set up the necessary configuration to establish a network connected to allow the microcontroller to transmit sensor data to the mobile application. The data received by the mobile application displays it in a user-friendly format, categorizing the water quality as “Clean”, “Murky”, or “Dark” based on the turbidity test levels. This display ensured that the user can easily monitor the water condition and take appropriate actions to maintain a healthy aquarium environment.

9.1.2 Dispensers

The dispensers for our AzulTank aids in maintaining the overall aquatic environment as well as the well-being of aquatic life. The food dispenser and pH dispenser play an important role in automating and optimizing the aquatic environment. The food dispenser ensured that the aquatic life is fed at consistent intervals, while the pH dispenser maintained the overall water quality and ensured the health of all the aquatic environment.

Together, this allowed for efficient management and reduced the manual workload for the user or owner for maintaining the aquarium's environment and the health of its aquatic inhabitants.

9.1.2.1 Food Dispenser

The food dispenser for our project comprises a series of processes to ensure its accuracy, reliability and functionality. To verify its accuracy, functionality and compatibility with the microcontroller, the component was connected to a microcontroller for data transfer and we then calibrated it to ensure it released the precise amounts of food. This process included setting up the component to deliver the exact amount required per feeding cycle.

Once we verified the compatibility as well as the component communicates well with the microcontroller. We proceeded to more advanced tests such as manual activation and timed dispensing. The manual activation verified the dispenser functionality where we triggered the dispenser manually. This also involved measuring the amount of food dispensed. We repeated this process numerous times to ensure the components accuracy and consistency for each feeding cycle.

When we conduct the timed dispensing, we programmed the microcontroller to activate the dispenser at specific times. We verified that it dispenses the correct amount of food for each feeding cycle. This process helped us ensure that the microcontroller accurately triggers the dispenser based on the programmed feeding schedule.

Another test we did as well is to perform a load testing. We do this by filling the dispenser with various amounts of food to ensure it can handle different loads without malfunctioning such as jams or blockages that may occur during operation. This helped us overall confirm that this component operates reliably within the system.

9.1.2.2 pH Dispenser

The pH dispenser for our project comprised a series of processes to ensure its accuracy, reliability and functionality. Initially, we aimed for the pH dispenser to operate automatically, where we release pH tablets on a schedule. However, to avoid the risk of random dispensing during scheduled times, we decided to only do manual dispensing instead. This approach ensured that the user had better control over the dispensing process and minimized the potential for accidental pH imbalances that could harm aquatic life.

We start by verifying the dispenser's accuracy, functionality and compatibility with the microcontroller. The component is connected to a microcontroller for data transfer and calibrated to ensure it releases the precise amounts of pH tablet.

Similar to the food dispenser, we validated the compatibility between the component and the microcontroller. This helped us ensure that both components can communicate well before proceeding to make some changes. We then proceeded to perform manual activation, repeating the process multiple times to verify that the dispenser dispensed the correct amount of pH tablets and ensured reliability.

Through this thorough testing we ensured that the dispenser's performance was accurate and dependable, providing the necessary precision to maintain a balanced pH level in the aquarium. This overall approach gave us the confidence that its functionality ensured the safety and well-being of the aquatic environment.

9.1.3 Motors

The motors used to control the AzulTank's dispensers, ensuring precise and efficient operation.

The stepper motor is used to control the pH dispenser. Its ability to move in discrete steps allowed us to rotate the dispensing mechanism accurately which ensured that it only dispenses one pH tablet at a time.

The servo motor on the other hand, is used for the food dispenser. Its ability to precisely control angles enabled it to release the specific amounts of food with consistency.

Both motors along with its own unique individual strengths, we were able to create a reliable and automated solution that managed the essential functions of the aquarium.

9.1.3.1 Servo Motor

The servo motor for our project involved a thorough process that ensured the precision, reliability and a smooth operation for our overall project. A test for this component is crucial for they are responsible for the precise control over the food dispensing mechanisms.

We started the test by and made sure that the motor is compatible and communicates well with the microcontroller. The servo motor for the food dispenser is calibrated to achieve the correct movement for dispensing the appropriate amount of food.

Manual activation test is then conducted to verify that the servo motor moves to the exact position needed for dispensing. We had to repeat this process multiple times to ensure its accuracy and consistent performance. This repetition process helped us detect possible errors or issues and inconsistencies that we may not have seen in a single test run.

Automation activation test was then followed after this, where the microcontroller was programmed to activate the servo motors based on predetermined schedules or sensor inputs. This ensured that the motors responded precisely and promptly without human intervention. This also ensured that the servo motors, dispensers and microcontroller work seamlessly together.

9.1.3.2 Stepper Motor

Similar to the servo motor, the stepper motor for our project involved a thorough process that ensured the precision, reliability and a smooth operation for our overall project. Testing this component was essential, as the stepper motor plays a critical role in accurately dispensing the appropriate number of pH tablets to maintain the water quality.

We started the test and verified the compatibility and communication between the stepper motor and microcontroller. A manual activation test was then conducted to confirm that the stepper motor performed the precise steps needed to dispense the appropriate amount of tablets. This process was repeated multiple times to ensure accuracy and consistency, which helped and allowed us to detect possible errors or issues and inconsistencies that we may not have seen in a single test run.

9.1.4 LED Lights

The LED lights for our projects are being used as indicators for our AzulTank. The testing process for this component can be considered to be particularly delicate or sensitive considering that it is critical to ensure that the LED lights do not harm the aquatic life in our tank.

We conducted a test on proper illumination and color accuracy. We also conducted a manual activation and automated activation for the LEDs to verify that they emit the correct colors. For example, we have red for warnings and green for normal conditions.

We first verified that the components are compatible as well as it communicates well with the microcontroller. We then manually activated the LEDs and observed them to ensure that they lit up correctly with the expected color and brightness.

The automated activation for the LEDs involved turning on the LEDs based on sensor inputs. We did this by setting thresholds for temperature, pH and turbidity levels that eventually triggered the LEDs to light up specific colors. This helped us ensure that the components respond accurately to various conditions.

For example, if the temperature sensor detected that the water temperature was outside the ideal range for the aquarium, the LED illuminated or lit up 'red.' Similarly, if the pH levels deviated from the ideal range or if the turbidity levels indicated murky water, the LEDs illuminated specific colors to signal the user to take appropriate action. This system provided a clear and immediate visual alert to help maintain optimal conditions in the aquarium.

A load test was also performed by having the LEDs run for extended periods of time to ensure their ability to tolerate continuous operation without failure. These tests gave us the confidence that LED indicators functioned correctly offering reliable feedback on the aquarium's status. This also helped us verify that LEDs did not emit light intensities or colors that prevent any potential stress or harm to aquatic life.

9.1.5 PCB

When testing our PCBs we needed to guarantee the functionality and reliability because of the importance it is to our electronic components. Testing various types of PCBs, such as voltage regulators, ESP32 modules, USB connectors, and sensor interfaces, requires specific approaches tailored to their unique functionalities. This is a quick outline that goes step-by-step on how we are using testing procedures for each type of PCB.

9.1.5.1 Testing Voltage Regulator PCB

The voltage regulators for our design maintain a consistent voltage level to protect more delicate electronic components. We begin with a visual inspection again to ensure proper component placement and solder quality.

For the initial input voltage test, we connect a variable DC power supply to the regulator's input and use a multimeter to confirm the input voltage is within the operating range. We then measure the output voltage with a multimeter to ensure it aligns with the expected regulated voltage that our design requires.

For load testing, we connect a load resistor or electronic load to the output and verify the regulator maintains a stable output under varying loads. We then perform a line regulation test by varying the input voltage and measuring the output to ensure stability. Optionally, we can also use an oscilloscope to check for output ripple and noise, confirming the regulator provides clean power. In general being able to check the noise can give us an idea as well if we have to move components or change them to allow for less noise and output ripple.

9.1.5.2 Testing ESP32 PCB

ESP32 modules are popular in IoT applications due to their integrated Wi-Fi and Bluetooth capabilities. We start with a visual inspection to check the placement of the ESP32 chip and other critical components.

We conduct a power test by connecting a separate and safe power supply to the PCB and measuring the voltage at the power pins to ensure it is within the ESP32's operating range. We then have a programming test that is crucial for us to see if it properly works as intended by connecting a USB to UART converter and uploading a simple test program (e.g., a blinking LED or something similar) using the Arduino IDE or ESP-IDF. We then verify if it was successfully uploaded and operational.

For the serial communication test, we open the serial monitor on a computer and ensure data can be sent and received accurately. We also test additional peripherals (GPIOs, ADCs, I2C, SPI) by writing and uploading appropriate test programs.

9.1.5.3 Testing USB Connector in MCU PCB

The USB connector in our MCU PCBs are essential for providing the connectivity for USB devices. We then begin with another visual inspection to check for correct placement and soldering of the USB connectors.

We perform continuity tests using a multimeter to ensure proper electrical connections between the USB connector pins and the PCB traces. For the power test, we connect the PCB to a USB power source and measure the voltage at the VBUS and GND pins to ensure it is within the expected range.

We use a USB tester to verify the proper operation of the D+ and D- data lines, ensuring data can be transferred without errors. Finally, we conduct a device recognition test by connecting a USB device (e.g., a flash drive or something similar) to the connector and verifying the computer recognizes and communicates with the device.

9.1.5.4 Testing Sensor PCBs

The Sensor PCBs interface with our various sensors (e.g., pH, temperature, and Turbidity sensor) for data acquisition. We also start with a visual inspection to ensure correct placement and soldering of sensor components.

We then conduct a power test by connecting a separate and safe power supply to the sensor PCB and measure the voltage at the sensor's power pins to ensure it is within the operating range. For our sensors with analog outputs (e.g., pH and Turbidity), we measure the output with a multimeter or oscilloscope. For the digital sensors (e.g., Temperature), we read data using a microcontroller or computer.

The Functionality tests are crucial to verify the sensor's response to stimuli as we are using sensors for live real-time data for our smartphone app. For example, we apply pressure to a pressure sensor or use various scenarios tailored for each sensor and measure the output signal. For our sensors using communication protocols like I2C or SPI, we connect to a microcontroller and verify the sensor data to ensure accurate communication.

Overall, testing our PCBs is a meticulous process that ensures the functionality, reliability, and safety as we use very delicate components on different sections of the overall design. Each type of PCB, from AC to DC converters and voltage regulators to ESP32 modules, USB connectors, and sensor interfaces, requires specific testing procedures tailored to their unique operational requirements. By following these testing methods, we can validate the performance and integrity of the PCB designs, guaranteeing they meet the necessary standards for our design.

9.1.6 Microcontroller

Testing the microcontroller before using it in the project is a crucial step to ensure its full functionality and reliability and avoid any potential issues during development. The microcontroller is the heart of the project since it handles critical tasks by interfacing with various components; therefore, rigorous testing of the microcontroller is necessary, and this process includes inspecting for physical damage, power-up tests, peripheral testing, memory tests, stress tests and the basic ability to execute code.

First, the team conducted a visual inspection for any physical damage or manufacturing defects. This step is essential to identify obvious flaws before the

MCU is powered up. If everything looks as expected, the microcontroller is then powered up to begin the electrical testing phase with a voltage test. This test was considered successful because voltage was found within the specifications range and if there are no signs of overheating.

Next, the microcontroller's basic functionality was tested. Here we loaded a series of classic programs such as flashing an LED. Blinking an LED, provided a clear, visual indication that code was executing, and communication has been established. This test was considered successful by observing the LED can blink at various set rates, therefore functioning as expected.

From here, we move on to test the peripheral's capability and functionality. This area covered a range of tests, starting with GPIO testing. A program was executed to toggle GPIO pins, and with a multimeter, we tested their outputs. With this done, we can configure some pins as inputs and read their states by connecting buttons to them. The test was considered successful as the measurements are within specification. The next test was for the ADC or analog-to-digital converter. This test was considered successful after connecting an analog voltage source to an ADC pin and running a program, we verified the digital output corresponds to the input voltage.

For the memory test, we wrote a program that was stored in both RAM and flash memory. We then read and output the contents of each memory module. The contents matched, confirming that the test was successful. With these fundamental areas validated, we proceeded to stress test the microcontroller to evaluate its performance under more demanding conditions.

Stress testing the system was a critical phase to evaluate the stability of the MCU. This process involved setting the microcontroller to its maximum operating frequency and temperature range, along with voltage variations to ensure it could handle demanding workloads of different types in the environment in which it would be installed. For this test, a comprehensive program was run utilizing various peripherals while executing multiple computations. The goal was to identify any situations in which the microcontroller could not operate or exhibited failure. The test was successfully completed, confirming that the MCU remained stable over the set period without any system crashes.

By implementing the tests above, the team was able to thoroughly test the microcontroller's capabilities to their full extent. This ensured the MCU functioned as expected, reducing the risk of encountering issues during development and providing the project with the stability and reliability it required.

9.2 Software Testing

This section focused on the software testing of the mobile application. The software testing and evaluation of the application were crucial to ensure the entire design operates as expected and fulfilled the team's proposed

requirements. By thoroughly evaluating the application's functionalities the team verified that all the integrated components work well together, this way ensuring frontend, backend and communication protocols are all functioning as intended.

Additionally, it allowed the team to identify bugs early in the development process. Addressing these issues before the application before deploying was essential for maintaining a high performance and reliability of the application. This approach provided the user with a seamless, reliable and high-quality experience on release.

9.2.1 General User Interface

To ensure a seamless and intuitive user interface experience, this section focused on rigorous testing of the UI and its interactive elements. Comprehensive UI and button testing enhanced the application's usability and aesthetic appeal while also ensuring functionality and reliability across various devices and screen sizes. The UI was tested in sections. The sections were the following: landing page, settings page, LED page and scheduling page. Every tab and button was tested to identify and address any potential issue. This approach allowed the team to create a polished and user-friendly mobile application, providing users with a smooth and responsive experience.

For the landing page, the tank's overall status indicators were prominently displayed, with all the buttons neatly arranged below them. The tests confirmed that the temperature, turbidity, and pH reading updated in real-time and matched the data provided by the sensors. Additionally, the status UI successfully changed based on the warning system, accurately reflecting the set thresholds for the temperature, turbidity and pH levels to alert the users effectively.

Each button was tested individually to ensure its functionality. The tests verified that every button fulfilled its assigned task, such as dispensing food, dispensing pH tablets, opening the LED menu, and accessing the scheduling menu.

For the scheduling page the tests focused on each one of the tabs assigned to each individual day. The test confirmed that each tab allowed the user to set feeding times and LED times and were able to successfully save those settings. Additionally, the users were able to specify and save the amount of food to be dispensed which ensured the precise control over the feeding schedule.

For the LED page, the test confirmed that each LED preset button and color swatches successfully set the aquarium's lighting and warning system colors. Each button visibly adjusted the lighting to its assigned color, ensuring functionality. Additionally, the color swatches allowed the user to select and set custom colors, visibly reflecting the user's choice in the aquarium's lighting.

Finally for the settings page there was conducted for each of the tabs contained on the page. To pass the tests, each tab needed to direct the user to the correct

configuration menu such as the LED menu, dispenser's configuration, schedules and any others that may be added throughout development.

Once all UI and button tests have been executed and successfully passed, we achieved a fully functional and robust interface. The comprehensive testing of every page and element ensured that they worked seamlessly and intuitively, providing users with a smooth and responsive experience.

9.2.2 Data Storage

The data storage test ensured that the most up-to-date sensor data was properly stored. It was an essential software component since this data represented the status of the fish tank on the landing page of the mobile application. The test was divided into several areas to confirm that all functionalities worked as expected. The tests included reading the sensor data in preparation for storage, transferring it to a database, and ultimately sending it over Wi-Fi to be displayed on the mobile application's landing page.

For the first test, which focused on reading the data we first ensured the basics such as making sure all the sensor connections to the ESP32 are correctly in place. Utilizing a serial monitor connection the data read by the sensors was displayed for inspection. To pass this test, the team verified the data being read was consistent and fell within expected ranges.

From here, we moved on to the next test, which involved sending the data over to the database. This test focused on verifying integrity of the data after it was passed over to the database. The sensor data was written to the database and immediately compared to the sensor's local data to ensure they matched. To pass this test, various checks at different intervals and the percentage of matches needed to come back as 100% otherwise the database tables would be considered inaccurate.

The last test ensured that the sensor data displayed on the mobile application's landing page was accurate. For this step, the system underwent stress testing under various data collection loads. To pass this test, the mobile application fish tank status needed to successfully store and update the displayed water quality values accurately. Additionally, general testing included data entry checks to verify that was being logged at the expected intervals without any missing entries.

9.2.3 Notifications

The notifications system is a critical component of the smart aquarium, ensuring the user's peace of mind and the fish's well-being. It is responsible for reporting all activity within the fish tank back to the user, overseeing the proper functioning of the dispensers and water quality of the aquarium. The notifications tests covered various activities that are being monitored within the tank that were

designed to trigger alerts and warnings. Each of the tests differed based on the type of activity and what's being tested. Below we begin by discussing the alerts.

The first test focused on the food dispenser, ensuring fish food was dispensed once the notification alerted the user. The test confirmed that notifications were received precisely as the food was being visibly dispensed, validating the system's accuracy and eliminating the possibility of false notifications.

The second test revolved around the pH dispenser and was successfully completed. The test ensured that the pH tablet was visibly dispensed when the notification was sent to the user. Again, this confirmed the system's accuracy and eliminated the possibility of false notifications, even when the notifications were received.

The third test centered around notifying the user of temperature changes and was successfully completed. The test ensured that the user received a notification as soon as a temperature change was detected to be either below or above a set threshold. This notification was critical for the safety of the fish, as it allowed the user to take immediate action to address potentially harmful temperature fluctuations.

The fourth test focused on notifying the user of turbidity level changes on the water. The test was successfully completed and ensured that a notification was sent to the user as soon as the turbidity of the water went above a certain threshold. Once again, an important precaution that was vital to ensure the well-being of the fish.

The final test revolved around the pH solution dispenser. The test confirmed that the notification was sent when the pH dispenser visibly verified dispensed the tablet into the water. This necessary precaution ensured the system's reliability and safeguarded the well-being of the fish.

9.3 Identifying and Resolving Issues

One of the primary goals of the testing and prototyping stages is to identify issues and resolve them early in the development process. By conducting thorough tests and creating detailed prototypes, we are able to uncover potential problems before they could escalate into more significant issues. This allows us, not only to improve the overall quality of the AzulTank device, but also reduce the risk of costly and time-consuming fixes later on.

In conclusion, the system testing and prototyping phases are extremely important in the development of the AzulTank. By meticulously testing both hardware and software components and iteratively refining our designs, we ensured that the AzulTank met all necessary requirements and could perform reliably under various conditions. This thorough approach allowed us to deliver a robust and

user-friendly smart aquarium management system, ultimately contributing to the success of our project.

Chapter 10 - Administration

10.1 Budget and Financing

The table below represents the finalized budget, detailing the actual costs of all essential components for our project's successful implementation. Each component listed is critical to the overall functionality and effectiveness of the system, ensuring that all necessary elements are in place for optimal performance.

Additionally, the overall budget includes the primary components and additional costs, ensuring that all necessary expenses have been accounted for in the total project costs. This allocation confirms that we can acquire the required materials while effectively managing costs, which enables us to maintain the balance of affordability and quality.

Table 10.1: *Estimated Budget and Financing for the project.*

Budget and Financing	
Component	Estimated Price
pH Sensor	\$60
Temperature Sensor	\$10
Turbidity Sensor	\$20
Dispenser(s)	\$20
LED Lights	\$16
Development boards	\$36
Micro SD Card Module	\$8
Servo Motor	\$3
Stepper Motor	\$5
PSU	\$8
Custom PCB	\$182
pH tablet	\$11
Fish Food	\$14
Total	\$393

10.2 Distribution of Work

Efficient work distribution is key to the AzulTank's success. By assigning tasks based on each team member's strengths in hardware design, software development, system integration, and testing, we ensure high-quality and timely

completion. This allowed us to collaborate, share our knowledge, and support each other, ensuring the delivery of this final report. The following table outlines each team member's specific tasks and responsibilities:

Table 10.2: *Distribution of Work for the project.*

Electrical Engineering	Responsibilities
Gabriel Besana	Overall Project Design and layout
	Design Constraints
	PCB Design and Implementation
	PSU Design and Implementation
Computer Engineering	Responsibilities
Jazz Olario	Administrative Documentation
	Software Design and Implementation
	Motor Selection and Dispenser Design
	MCU Selection and Implementation
Computer Engineering	Responsibilities
Rafael Nieves	Sensor Research and Selection
	MCU and Sensor Schematics
	Design Constraints
	MCU and Sensor Design
Computer Engineering	Responsibilities
Christian Rosado Arroyo	Project Lead
	Website Design and Management
	Software Design and Implementation
	MCU Selection and Implementation

10.3 Project Milestones

With the established due dates, we followed this time frame as closely as possible. Most of these milestones involved paper throughout the semester.

Much of our research was conducted during Senior Design 1, focusing on data collection and conceptual development.

We also initiated PCB design and began developing the front-end application during this semester in which it helped us prepare for the tasks ahead in Senior Design 2. Although it was challenging to achieve this milestone, especially during the summer semester with limited time, we successfully started developing the front-end, which significantly enhanced the overall progress of our project.

Table 10.3: Project Milestones for Senior Design 1.

Senior Design 1			
Overview	Description	Planned Duration	Status
Group formation	Identify group members and define individual roles	1 week	Completed
Project ideas	Come together and come up with project ideas	1 week	Completed
Divide and conquer	Start writing 10-page proposal for approval	1 week	Completed
Divide and conquer submission		Due 5/31/24	Completed
Research and planning	Begin writing 60-page document draft	4 weeks	Completed
Review draft	Review and revise draft based on feedback from peers and reviewers	1 week	Completed
60-page draft submission		Due 7/5/24	Completed
HW & SW design	Initiate PCB design as well as the develop front-end application development	1 week	Completed
Finalize and review report		1 week	Completed
120-page final report due		Due 7/23/24	Completed

The planned duration for the Senior Design 2 milestones is based on Fall Semester and will remain unpredictable. Even so, we have our project foundation all set along with a clear direction for our main idea. As we approach the start of Senior Design 2, our focus is on thorough preparation.

PCB design and App Development is crucial for our project's progress. An early completion of these milestones will certainly help with our progress and possibly ensure our team to be ready to meet any challenges ahead for Senior Design 2.

Table 10.4: Project Milestones for Senior Design 2.

Senior Design 2			
Overview	Description	Planned Duration	Status
PCB Assembly	Building the PCB	4 weeks	Completed
Develop Application	Develop and integrate front-end application	4 weeks	Completed
Test Sensors & Dispensers	Test sensors & dispenser accuracy and Responsive	2 weeks	Completed
Test application	Test user interface responsiveness and ensure reliability of notification-based sensors	2 weeks	Completed
PCB Assembly & App Development	Ensure that hardware and software are communicating properly	4 weeks	Completed
Debug & Test	Identify and resolve issues in both hardware and software components	2 weeks	Completed
Finalize Paper		1 week	Completed
Review		1 week	Completed
Final Demo		Due 11/22/24	Completed
Final Paper		Due 11/26/24	Completed

Chapter 11 - Conclusion

In the rapidly advancing landscape of smart home technology, our smart fish aquarium 'AzulTank' stands out as a highly beneficial addition and innovation for both well-versed aquarists and beginners alike who want to step out of the home for longer periods of time. The adaptation of the traditional fish tank to a modern day version allows for more cutting-edge technology to create an environment

that optimally maintains the health and well-being of aquatic life with a touch of aesthetics for the eyes.

Our group aimed for the 'Azul Tank' to have an advantage in addition to automation for essential maintenance tasks. With a traditional aquarium, they require regular monitoring and manual hands-on adjustments to maintain water quality, temperature, and lighting. In contrast, the 'Azul Tank' comes equipped with automated systems, sensors, and real time data analytics that continuously monitor these parameters, allowing for real-time adjustments while moving away from the manual labor side when owning an aquarium. The automation provided reduces the risk of human error to ensure a stable environment. Like with each dispenser that is set to a specific margin of error that allows a certain amount of pH solution, and or food to dispense to prevent any drastic pH changes or over feeding scenarios. With the delicacy of aquarium ecosystems, it's crucial to keep everything in check at all times.

Another thing we aimed for this project was the integration of our smartphone app and local based systems, it allows aquarium owners to have unprecedented control and insight into the aquatic environments. Through the app, users can remotely monitor water quality and temperature, feed the fish, and even fix the tank's pH levels. Notifications can even alert owners to potential issues, such as temperature fluctuations, water turbidity, or pH quality imbalances that allows for the user to prompt intervention remotely to an extent. The level of connectivity that the AzulTank has not only enhanced the user experience but also ensured the long-term health of the aquarium's inhabitants. Another important feature to note is that the app is tailored for all ages and is created towards having a positive user friendly experience even for beginners.

For any homeowner, the aesthetic of their aquarium is just as important as the benefits of one that is optimally obtained. An alert can include a pH problem that will turn the LED lights green for a period of time to bring attention besides a phone notification. With the ability to program the lighting to change color based on aquarium needs and the overall design of the AzulTank, it stays pleasing to the eyes and fits comfortably with any interior design when installed to the aquarium. Besides enhancing the ambiance of a room significantly, allowing for the real time analytics to help control the installed LED add's to the functionality and eye appeal.

With the AzulTank providing a significant leap towards smart aquarium technology, it also integrates educational benefits. With the accurate real-time data being sent to our smartphone app, it allows it to be an excellent platform to learn more about aquatic ecosystems for your fishes and the delicate balance needed to keep these aquatic creatures healthy. It takes a lot of effort for both maintenance and upkeep of aquatic life and having a device aid in the whole process allows for a better understanding of the necessities of time, attention, and detail to save both time and money in the long run.

Lastly, going beyond the technical and aesthetic advantages allows for more users to move on from traditional fish tanks to a more sophisticated, cost effective, and user friendly system. Within a normal work week, people are gone at least eight hours a day and not able to physically monitor their tanks for small changes. Through the automation of maintenance tasks and providing remote access at the touch of an app provides enough relief for all types of aquatic owners who can't give as much time to their aquarium. As technology continues to evolve, the appeal for a non traditional aquarium route will be greater.

This document serves as a comprehensive report of the AzulTank's fundamental theories, selection of components, technology comparisons, limitations, design processes, and administrative costs and budget for implementation. Moving forward, we hope that this gives further insight in future development or expansion of the smart aquarium hands-off technology that is both functional and cost benefit for the general masses.

Appendix A - References

1. "Analog to Digital Converter (ADC)." ESP, docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html.
2. "Design Constraints." EMU, me.emu.edu.tr/en/capstone-design-projects/engineering-design/engineering-design-constraints.
3. "Gravity__analog_ph_sensor_meter_kit_v2_sku_sen0161-V2." DFRobot, wiki.dfrobot.com/Gravity__Analog_pH_Sensor_Meter_Kit_V2_SKU_SEN0161-V2.
4. Kenton, Will. "ISO 14000 Definition, Standards, Certification, and Costs." Investopedia, Investopedia, www.investopedia.com/terms/i/iso-14000.asp.
5. "Standard Test Methods for Nickel in Water." D1886, www.astm.org/d1886-14r21e01.html.
6. "Turbidity_sensor_sku__SEN0189." DFRobot, wiki.dfrobot.com/Turbidity_sensor_SKU__SEN0189.
7. "Waterproof_ds18b20_digital_temperature_sensor__sku_dfr0198_." DFRobot, wiki.dfrobot.com/Waterproof_DS18B20_Digital_Temperature_Sensor__SKU_DFR0198_.
8. "KiCad." Wikipedia, Wikimedia Foundation, 28 June 2024, en.wikipedia.org/wiki/KiCad.
9. "KiCad Eda." Schematic Capture & PCB Design Software, www.kicad.org/. Accessed 7 July 2024.
10. "Altium Designer." *Wikipedia*, Wikimedia Foundation, 27 June 2024, en.wikipedia.org/wiki/Altium_Designer.
11. "Chatgpt." *Wikipedia*, Wikimedia Foundation, 5 July 2024, en.wikipedia.org/wiki/ChatGPT.
12. "Eagle / Autodesk Fusion Software: Get Prices & Buy Official." *Autodesk*, 4 Mar. 2024, www.autodesk.com/products/eagle/overview?term=1-YEAR&tab=subscription.
13. "Gemini (Chatbot)." *Wikipedia*, Wikimedia Foundation, 24 June 2024, en.wikipedia.org/wiki/Gemini_(chatbot).

14. Pichai, Sundar. "Introducing Gemini: Our Largest and Most Capable AI Model." *Google*, Google, 6 Dec. 2023, blog.google/technology/ai/google-gemini-ai/#sundar-note.
15. *Imperial College London*. "Servo Motor SG90 Data Sheet", www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf. Accessed 8 July 2024.
16. *NEMA 8 Stepper Motor*. StepperOnline. (n.d.). https://www.omc-stepperonline.com/nema-8-stepper-motor?srsId=AfmBOopE43UyDq0yY0uyO5WfvBkDHsARsrcYsw_-xm3d2edsL2rZeyxe
- 17.
18. "Stepper Motors vs. Servo Motors: ISL Products." *ISL Products International*, islproducts.com/design-note/stepper-motors-vs-servo-motors/#:~:text=Servo%20motors%20can%20generate%20speeds,and%20generate%20higher%20peak%20torque. Accessed 7 July 2024.
19. "University of Central Arkansas: UCA." *Center for Excellence in Teaching and Academic Leadership*, uca.edu/cetal/chat-gpt/. Accessed 7 July 2024.
20. Wiggers, Kyle. "Google Gemini: Everything You Need to Know about the New Generative AI Platform." *TechCrunch*, 28 June 2024, techcrunch.com/2024/06/28/what-is-google-gemini-ai/#:~:text=Gemini%20recently%20came%20to%20Google's,re%20on%20to%20make%20recommendations.
21. "Brushless Vs Brushed DC Motors: When and Why to Choose One Over the Other." *Monolithicpower*, Pete Millett, www.monolithicpower.com/en/learning/resources/brushless-vs-brushed-dc-motors. Accessed 12 July 2024.
22. "Types of Motors: Working and Applications." *GeeksforGeeks*, GeeksforGeeks, 29 Apr. 2024, www.geeksforgeeks.org/types-of-motors/.
23. "DC/DC Converters: Circuit Analysis–Intro to Predicting Circuit Output - Technical Articles." *EEPower*, eepower.com/technical-articles/dc-dc-converters-circuit-analysisintro-to-predicting-circuit-output/#. Accessed 12 July 2024.
24. *MPs*, www.monolithicpower.com/en/learning/mpscholar/power-electronics/dc-dc-converters/boost-converters#:~:text=Boost%20converters%20are%20a%20type,voltage%20conversion%20is%20made%20possible. Accessed 13 July 2024.
25. "Home." *RECOM*, recom-power.com/en/support/technical-resources/book-of-knowledge/ac-dc-b

- ook-of-knowledge/acdc-bok-chapter8-topologies.html?0. Accessed 5 July 2024.
26. Storr, Wayne. "AC-DC Converters Including Buck, Boost and Flyback." *Basic Electronics Tutorials*, 2 Aug. 2022, www.electronics-tutorials.ws/connectivity/ac-dc-converters.html. Accessed 11 July 2024.
 27. *Power Supply Technology - Buck DC/DC Converters* | Mouser Electronics, www.mouser.com/applications/power-supply-topology-buck/. Accessed 9 July 2024.
 28. Proto-Electronics. "Why Detailed Evaluation of Your AC-DC Converter Topology Pays Off?" *Proto*, www.proto-electronics.com/blog/choosing-ac-dc-converters-topology. Accessed 11 July 2024.
 29. Apple. "Choosing a Membership - Support - Apple Developer." *Apple.com*, 2019, developer.apple.com/support/compare-memberships/. Accessed 28 June 2024.
 30. "CURIOSITY PIC32MX470 DEVELOPMENT BOARD." *Https://Www.microchip.com/En-Us/Development-Tool/Dm320103*. Accessed 5 July 2024.
 31. "Difference between SQL and NoSQL - GeeksforGeeks." *GeeksforGeeks*, 26 Oct. 2018, www.geeksforgeeks.org/difference-between-sql-and-nosql/. Accessed 4 July 2024.
 32. Geeksforgeeks. "Microcontroller and Its Types." *GeeksforGeeks*, 20 Feb. 2022, www.geeksforgeeks.org/microcontroller-and-its-types/. Accessed 27 June 2024.
 33. Howarth, Josh. "iPhone vs Android User Stats (2023 Data)." *Exploding Topics*, 13 Oct. 2023, explodingtopics.com/blog/iphone-android-users. Accessed 29 June 2024.
 34. "MSP-EXP430FR6989 Development Kit | TI.com." *Www.ti.com*, www.ti.com/tool/MSP-EXP430FR6989#support-training. Accessed 5 July 2024.
 35. Nowak, Maja. "Flutter vs. React Native in 2022." *Nomtek*, 1 Feb. 2024, www.nomtek.com/blog/flutter-vs-react-native.
 36. Sharma, Avinash. "Step-By-Step Process to Upload App to Google Play Store." *Appinventiv*, 25 Oct. 2019, appinventiv.com/blog/how-to-submit-app-to-google-play-store/#:~:text=How%20Much%20Does%20it%20Cost. Accessed 29 June 2024.

37. Skuza, Bartosz, et al. "Flutter vs React Native – Which Is Better for Your Project?" *Droids on Roids*, 5 May 2024, www.thedroidsonroids.com/blog/flutter-vs-react-native-comparison.
38. "Universal Asynchronous Receiver Transmitter (UART) Protocol." *GeeksforGeeks*, 19 Sept. 2022, www.geeksforgeeks.org/universal-asynchronous-receiver-transmitter-uart-protocol/. Accessed 5 July 2024.
39. "What Is Full Stack Development? | a Complete Guide." *MongoDB*, www.mongodb.com/resources/basics/full-stack-development.
40. Qasim. "Linux vs Windows vs Mac: The Ultimate CheatSheet." *Www.redswitches.com*, 5 Aug. 2023, www.redswitches.com/blog/linux-vs-windows-vs-mac/. Accessed 3 July 2024.
41. "How to Use the Play Console - Play Console Help." *Support.google.com*, support.google.com/googleplay/android-developer/answer/6112435?hl=en.
42. Espressif. "ESP32-DevKitC v4 Getting Started Guide - ESP32 - — ESP-IDF Programming Guide Latest Documentation." *Docs.espressif.com*, docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html.
43. Brent Rubell and 2 other contributors Contributors: Tyler Cooper Adam Bachman. (n.d.). *Welcome to Adafruit Io*. Adafruit Learning System. <https://learn.adafruit.com/welcome-to-adafruit-io>
44. Google. (n.d.). Google. https://firebase.google.com/?gad_source=1&gclid=CjwKCAiA3ZC6BhBaEiwAeqfvyJh1fwRT0QhtJ1C_paZ4PCd7nB_h8Kx5lGKi2YjvhKmjeV3x5BC8pRoC2O4QAvD_BwE&gclsrc=aw.ds
45. *Node.js - about node.js®*. Node.js -. (n.d.). <https://nodejs.org/en/about>
46. Step motor 5V - mikroelektronika. MIKROE. (n.d.). <https://www.mikroe.com/step-motor-5v>
47. TPS54528. TPS54528 data sheet, product information and support | TI.com. (n.d.). https://www.ti.com/product/TPS54528?utm_source=google&utm_medium=cpc&utm_campaign=app-bsr-null-44700045336317437_prodfolderdynamic-cpc-pf-google-ww_en_int&utm_content=prodfolddynamic&ds_k=DYNAMIC%2BSEARCH%2BADSDCM=yes&gad_source=1&gclid=Cj0KCQiAi_G5BhDXARIsAN5SX7qmvwglphHr-NJ9uWPqGcz_Uks8Si_Zd4ZSPrNUd9dw7dIFEmhKUEwaAIK3EALw_wcB&gclsrc=aw.ds

Appendix B - ChatGPT Logs

Chat Log A

“Maintaining a healthy aquarium involves several key requirements to ensure the well-being of aquatic life and the overall ecosystem. Here are the essential components:

Tank Size and Setup:

- Choose an appropriately sized tank for the type and number of fish or other aquatic animals.
- Ensure the tank is placed on a stable, level surface and away from direct sunlight to prevent algae growth.

Filtration:

- Use a suitable filtration system to remove waste, debris, and harmful chemicals from the water.
- Regularly clean and maintain the filter to ensure it functions effectively.

Water Quality:

- Test water parameters regularly (pH, ammonia, nitrite, nitrate, hardness, and alkalinity) to keep them within the suitable range for your aquatic inhabitants.
- Perform regular water changes (typically 10-20% weekly) to remove accumulated toxins and replenish essential minerals.
- Dechlorinate tap water before adding it to the aquarium.

Heating and Lighting:

- Use a reliable heater to maintain a stable temperature appropriate for your specific fish species.
- Provide adequate lighting to mimic natural day and night cycles and support plant growth if you have live plants.

Aeration:

- Ensure proper oxygenation of the water using an air pump, airstone, or surface agitation from the filter.
- Monitor oxygen levels, especially in densely stocked or planted tanks.

Substrate and Decorations:

- Choose a suitable substrate (gravel, sand, etc.) that matches the needs of your aquatic animals and plants.
- Add decorations like rocks, driftwood, and plants to provide hiding places and reduce stress for fish.

Feeding:

- Provide a balanced diet appropriate for the species you keep, avoiding overfeeding to prevent water quality issues.
- Remove uneaten food to prevent decay and maintain cleanliness.

Aquatic Plants:

- If keeping live plants, ensure they have adequate light, nutrients, and CO₂ if necessary.
- Regularly trim and remove dead or decaying plant material.

Monitoring and Maintenance:

- Observe your fish and other inhabitants daily for signs of stress, disease, or unusual behavior.
- Perform routine maintenance tasks like cleaning the glass, removing algae, and pruning plants.

Disease Prevention:

- Quarantine new fish or plants before adding them to the main tank to prevent introducing diseases.
- Treat diseases promptly with appropriate medications if necessary.
- By paying attention to these aspects, you can create a thriving and healthy environment for your aquarium inhabitants.” (OpenAI - GPT-3.5, 2024)

Chat Log B

“Keeping a healthy aquarium requires a balance between the biological, chemical and physical aspects of the environment. Here are some key requirements:

Setting up the tank:

- Choose the right size tank for the fish you want. Bigger is generally better as it provides more stable water parameters.
- Use a good quality filter system. Filtration is essential for removing waste products and keeping the water clean.
- Consider using a substrate like gravel or sand to house beneficial bacteria that help break down waste.
- Decorate your tank with live or artificial plants, rocks and driftwood. Live plants can help with maintaining water quality and provide hiding spots for fish.

Water Parameters:

- Maintain the proper water temperature for your chosen fish species. Most tropical fish prefer temperatures between 72°F and 82°F.
- Regularly monitor and adjust water chemistry, including pH, ammonia, nitrite and nitrate levels. You can buy test kits to measure these parameters.
- Perform partial water changes (15-25% of the tank's volume) every 1-2 weeks to remove built-up waste and replenish minerals.

Fish care:

- Research the specific needs of the fish you want to keep. Different species have different requirements for water parameters, tank space, and diet.
- Avoid overstocking your tank. Overcrowding can lead to stress, illness, and poor water quality.
- Feed your fish a high-quality diet appropriate for their species and feeding habits. Don't overfeed - uneaten food will decompose and pollute the water.

Maintenance:

- Regularly clean the filter media to ensure proper function.

- Remove uneaten food and debris from the tank to prevent ammonia spikes.
- Trim overgrown plants and remove dead leaves to maintain aesthetics and water quality.

By following these requirements and establishing a regular maintenance routine, you can create a thriving underwater world for your fish to enjoy. Remember, there are many resources available online and at pet stores to help you learn more about specific fish care and aquarium maintenance.” (Google - Gemini, 2024)

Chat Log C

“The price range for servo motors and stepper motors varies depending on their specifications and quality. Here's a general overview:

Servo Motors:

- **Low-end:** \$10 - \$30 (typically small hobby servos)
- **Mid-range:** \$30 - \$150 (used in more demanding applications like RC vehicles, robotic arms)
- **High-end:** \$150 - \$500+ (industrial servos with high precision and torque)

Stepper Motors:

- **Low-end:** \$10 - \$20 (small stepper motors for basic applications)
- **Mid-range:** \$20 - \$100 (larger stepper motors for 3D printers, CNC machines)
- **High-end:** \$100 - \$300+ (industrial stepper motors with high torque and precision)

Choosing Between Servo Motors and Stepper Motors:

The choice between servo motors and stepper motors depends on your specific application and requirements:

Servo Motors:

- **Advantages:**
 - High precision and accuracy
 - Closed-loop control with feedback
 - High torque at high speeds
 - Smooth operation
- **Disadvantages:**
 - Generally more expensive
 - More complex control systems

- **Best For:** Applications requiring precise control, high speed, and high torque, such as robotics, CNC machines, and RC vehicles.

Stepper Motors:

- **Advantages:**
 - Simpler control systems
 - High torque at low speeds
 - Precise position control in open-loop systems
 - Generally more affordable
- **Disadvantages:**
 - Less efficient at high speeds
 - Can lose steps under heavy loads without feedback
 - Potential for resonance issues
- **Best For:** Applications requiring precise control at low to moderate speeds, such as 3D printers, small CNC machines, and automated conveyor systems.

Application Considerations:

- **Precision:** If you need high precision and feedback control, a servo motor is generally a better choice.
- **Speed and Torque:** For high-speed applications with significant torque requirements, servo motors are usually more suitable.
- **Cost:** For budget-sensitive projects, stepper motors might be a better option, especially if the application's precision and speed requirements are moderate.
- **Control Complexity:** Stepper motors are simpler to control and might be more suitable for projects where ease of implementation is a priority.

For your specific project, consider the following factors:

- The required precision and torque
- Budget constraints
- Control system complexity
- Speed requirements

Based on these factors, you can make a more informed decision on whether a servo motor or a stepper motor is more suitable for your needs.” (OpenAI - GPT-3.5, 2024)

Chat Log D

“Common lights used for aquariums include:

1. **Fluorescent Lights:** These are energy-efficient and provide a broad spectrum of light. They are suitable for both freshwater and marine aquariums.
2. **LED Lights:** Popular due to their long lifespan, energy efficiency, and customizable spectrum. They can be used in both freshwater and marine setups and are ideal for planted tanks and coral reefs.
3. **Incandescent Lights:** Less common due to their inefficiency and heat production. They are typically used in smaller, basic setups.
4. **Metal Halide Lights:** Produce intense light and are commonly used in large, deep marine aquariums, particularly for reef tanks with coral.
5. **Compact Fluorescent Lights (CFLs):** Provide a good amount of light and are suitable for planted freshwater aquariums.
6. **T5 and T8 Lights:** These are types of fluorescent lights with different diameters. T5 lights are thinner and more efficient compared to T8 lights.
7. **VHO (Very High Output) Fluorescent Lights:** These produce more intense light than standard fluorescent lights and are suitable for larger aquariums, especially marine and reef tanks.
8. **Actinic Lights:** Emit light in the blue spectrum, enhancing the colors of marine life and promoting photosynthesis in corals and other marine plants.
9. **Moonlights:** These are LED lights that mimic the natural lunar light cycle, often used to simulate nighttime conditions in the aquarium.

Each type of light has its specific use cases, and the choice depends on the type of aquarium, the inhabitants, and the desired aesthetic and functional outcomes.” (OpenAI - GPT-3.5, 2024)

Chat Log E

“When deciding between servo motors and stepper motors for your application, it's important to consider several factors including cost, performance, precision, and application requirements. Here's a comparison of the price ranges and typical use cases for each type of motor:

Price Range

- **Servo Motors:**
 - **Price Range:** \$50 to \$500 or more, depending on the size, torque, and precision required. High-end industrial servo motors can be even more expensive.
 - **Factors Influencing Price:** Includes the motor, encoder, controller, and often a gearbox.
- **Stepper Motors:**

- **Price Range:** \$10 to \$100 for most common models. High-precision or high-torque models can go up to \$200.
- **Factors Influencing Price:** Generally less expensive as they do not require encoders for open-loop systems. Prices can increase with higher torque and precision requirements.

Performance and Application Considerations

Servo Motors:

- **Advantages:**
 - **High Precision and Accuracy:** Equipped with feedback systems (encoders) for precise control of position, speed, and torque.
 - **Smooth Operation:** Better performance at high speeds and can handle higher torque loads.
 - **Efficiency:** More efficient for continuous operation with variable loads.
- **Disadvantages:**
 - **Complexity:** Requires a more complex control system (controller, encoder).
 - **Cost:** Generally more expensive than stepper motors.
- **Typical Applications:**
 - CNC machinery
 - Robotics
 - Conveyor systems
 - Applications requiring precise position control and high torque at various speeds.

Stepper Motors:

- **Advantages:**
 - **Cost-Effective:** Generally less expensive and simpler to control.
 - **Simplicity:** Easy to implement for applications requiring less precise control.
 - **Good Torque at Low Speeds:** Provide high torque at low speeds, suitable for holding applications.
- **Disadvantages:**
 - **Step Accuracy:** Lower resolution and accuracy compared to servos.
 - **Torque Drop-off:** Torque decreases as speed increases.
 - **Heat Generation:** Can generate more heat during operation.
- **Typical Applications:**
 - 3D printers
 - Small CNC machines
 - Positioning systems (e.g., camera platforms)
 - Simple robotics and automation where high precision is not critical.

Which One Should You Use?

- **For Precision and Variable Load Applications:** If your application requires high precision, variable speed, and torque, or continuous feedback, servo motors are likely the better choice despite their higher cost and complexity.
- **For Cost-Effective and Simple Control Applications:** If your application involves simple position control, cost constraints, or lower speed operations with high torque at low speeds, stepper motors are generally suitable and more cost-effective.

Given your project is related to a smart fish tank helper, where precise control of components like dispensers and sensors is necessary but not at a high speed or torque, stepper motors might be a good fit due to their simplicity and cost-effectiveness. However, if you need more precision or have varying load conditions, considering servo motors might be beneficial despite the higher cost.” (OpenAI - GPT-4o, 2024)

Chat Log F

“Your hardware diagram looks well-organized and clear. Here are a few suggestions to consider for possible improvements:

1. **Label Clarity:**
 - Ensure all labels are clearly readable. The use of (TBA) for "To Be Acquired" is helpful, but ensure that each element that needs to be acquired is distinctly marked.
2. **Power Supply Unit (PSU):**
 - The PSU box and connections are clear, but consider indicating the specific power requirements or specifications if they are known, as this could help in the acquisition process.
3. **Component Connections:**
 - Double-check that all necessary connections are represented. For instance, ensure the Wifi Module is properly shown as connected to both the MCU and the mobile phone.
4. **Color Coding:**
 - The color coding is a great idea for indicating different statuses. However, ensure there is consistency in how the colors are used. For example, if the red dotted line represents communication lines, make sure this is uniformly applied throughout the diagram.
5. **Additional Components:**
 - If there are additional components or future expansions (e.g., additional sensors or modules), consider adding placeholders for these in the diagram.
6. **Legend Details:**

- Expand the legend if necessary to include more detailed information about the roles and responsibilities of each individual or component status.
- 7. Interfacing and Communication:**
- If the MCU communicates with the mobile phone over a specific protocol (e.g., Bluetooth, WiFi), consider specifying this on the diagram.
- 8. Error Handling:**
- If applicable, indicate any error handling or backup systems. For instance, if there is a fail-safe for the motor or dispensers.
- 9. Physical Layout Considerations:**
- If the physical layout of components is important (e.g., proximity of sensors to the fish tank), consider adding a note or a small diagram indicating this.

Implementing these suggestions can enhance clarity and ensure all team members and stakeholders are on the same page regarding the design and functionality of the system.” (OpenAI - GPT-4o, 2024)