

A Smart Aquatic Ecosystem

DESIGN DOCUMENT

sdmay26-21

Md Maruf Ahamed (Advisor/Client)

Carson Irving (Frontend/Backend/AI)

Max Purvis (Hardware)

Daniel Pender (Hardware)

Aden Koziol (AI)

Marina Mannan (Embedded Systems)

Griffin Urban (Frontend)

Ryan Holden (Backend)

sdmay26-21@iastate.edu

sdmay26-21.sd.ece.iastate.edu

Revised: Sept. 29th, 2025

Executive Summary

Traditional aquarium maintenance requires frequent manual monitoring of water quality and fish health, which can lead to inconsistent care and increased stress for aquatic life. Our smart aquatic ecosystem aims to use AI and automation to eliminate the need for constant manual adjustment and create a more stable environment for fish or other aquatic life.

The two most important design requirements are automated monitoring and controlling of the environment inside the fish tank. Ideal water conditions can be created by controlling measurable variables of the water quality, including pH, temperature, dissolved oxygen, and turbidity.

Our implementation will use smart sensors to track this data in real time and visualize it on a web-based application for the user's benefit. Automated controls will regulate temperature, feeding, and water exchange/flow. Additionally, the system will track the fish population with a camera using an AI-driven computer vision model. This data will be used along with water quality patterns to model population trends and predict fish health issues. The web app will show records of every adjustment made to the environment, and will notify the user to recommend any necessary action to ensure the continued health of the ecosystem.

Our progress so far has involved researching and ordering the equipment for setting up the fish tank. We ordered the tank itself, pumps, filters, water quality sensors, and an Arduino. We have also researched quality cameras and AI camera detection libraries.

Our next step is to build the tank and begin testing. After we receive the parts, we can begin water treatment to create a stable environment. Then we can buy the fish and begin training the AI model to track population growth.

Put together, this system addresses all the needs of the user by eliminating the need for constant manual care. It monitors water quality and fish health, performs automatic feedings and water replacement, and alerts the user to any urgent changes in the environment.

Learning Summary

Development Standards & Practices Used

List all standard circuits, hardware, and software practices used in this project.
List all the Engineering standards that apply to this project that were considered.

Software

- camelCase and 4-space tabs for programming language
- ASP.NET Core WEB API for backend.
- Descriptive Git commits and work on individual branches before merge approval

Hardware

- Sensor testing and calibration
-

Summary of Requirements

List all requirements as bullet points in brief.

- AI computer vision fish population and growth analysis
- Automatic feeding mechanism using actuators
- Basic Aquarium Equipment (Tank, Filter, Air Pump)
- Advanced Aquarium Sensors (pH, turbidity, water quality, dissolved oxygen, temperature)
- Web-based application to monitor insights, control systems, and real-time visualization of data

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

SE

- 2280 - *Introduction to Data Structures*
- 3090 - *Software Development Practices*
- 3170 - *Introduction to Software Testing*
- 3190 - *Construction of User Interfaces*
- 3630 - *Introduction to Database Management*

CS

- 3110 - *Introduction to the Design and Analysis of Algorithms*

CPRE

- 2880 - *Embedded Systems I*

EE

- 1850 - *Introduction to Electrical Engineering and Problem Solving I*
- 2010 - *Electric Circuits*
- 2300 - *Electronic Circuits and Systems*
- 2850 - *Problem-Solving Methods and Tools for Electrical Engineering*

New Skills/Knowledge acquired that were not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

Table of Contents

1.	Introduction	5
1.1.	PROBLEM STATEMENT	5
1.2.	INTENDED USERS	5
2.	Requirements, Constraints, And Standards	5
2.1.	REQUIREMENTS & CONSTRAINTS	5
2.2.	ENGINEERING STANDARDS	5
3	Project Plan	6
3.1	Project Management/Tracking Procedures	6
3.2	Task Decomposition	6
3.3	Project Proposed Milestones, Metrics, and Evaluation Criteria	6
3.4	Project Timeline/Schedule	6
3.5	Risks And Risk Management/Mitigation	7
3.6	Personnel Effort Requirements	7
3.7	Other Resource Requirements	7
4	Design	7
4.1	Design Context	7
4.1.1	Broader Context	7
4.1.2	Prior Work/Solutions	8
4.1.3	Technical Complexity	8
4.2	Design Exploration	9
4.2.1	Design Decisions	9
4.2.2	Ideation	9
4.2.3	Decision-Making and Trade-Off	9
4.3	Proposed Design	9
4.3.1	Overview	9
4.3.2	Detailed Design and Visual(s)	9
4.3.3	Functionality	10
4.3.4	Areas of Concern and Development	10
4.4	Technology Considerations	10
4.5	Design Analysis	10

5	Testing	10
5.1	Unit Testing	11
5.2	Interface Testing	11
5.3	Integration Testing	11
5.4	System Testing	11
5.5	Regression Testing	11
5.6	Acceptance Testing	11
5.7	Security Testing (if applicable)	11
5.8	Results	11
6	Implementation	12
7	Professional Responsibility	12
7.1	Areas of Responsibility	12
7.2	Project Specific Professional Responsibility Areas	12
7.3	Most Applicable Professional Responsibility Area	12
8	Closing Material	12
8.1	Discussion	12
8.2	Conclusion	12
8.3	References	13
8.4	Appendices	13
9	Team	13
9.1	TEAM MEMBERS	13
9.2	REQUIRED SKILL SETS FOR YOUR PROJECT	13
	(if feasible – tie them to the requirements)	13
9.3	SKILL SETS COVERED BY THE TEAM	13
	(for each skill, state which team member(s) cover it)	13
9.4	PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM	13
	Typically Waterfall or Agile for project management.	13
9.5	INITIAL PROJECT MANAGEMENT ROLES	13
9.6	Team Contract	13

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1. Introduction

1.1. PROBLEM STATEMENT

Keeping an aquarium may sound simple, but it's actually a delicate process where small changes in water quality can have big effects on the fish's health. Most aquarium owners stick to a maintenance routine, but what happens when life gets in the way? Our project aims to solve this problem with a smart, AI-powered aquarium management system that monitors water conditions in real time and responds automatically. By using sensors and intelligent controls, the system can regulate temperature, feeding, water flow, and even perform water changes when needed. There will also be artificial intelligence tracking the growth and population of the fish to provide more insights for the owners. This creates a healthier aquatic environment for the fish while reducing the workload for busy owners. The insights gathered will be easily accessible remotely from either a web interface or application, providing peace of mind for the owners who might be away from the tank.

1.2. INTENDED USERS

Who will use the product you create? Who benefits from or will be affected by the results of your project? List as many users or user groups as are relevant to your project (at least three). For each user or user group, (1) describe the user and their key characteristics (e.g., a persona), (2) identify their need(s) related to the project (e.g., a needs statement), and (3) discuss how they might benefit or derive value from the product you create. Justify how these benefits/this value connects to your overarching problem statement.

User Group 1: Casual Aquarium Owners

Description: Individuals or families who have a fish tank at home. They typically have limited knowledge of aquarium care and little time for regular maintenance.

Needs Statement: They need an easy and low-effort way to keep their fish alive and healthy without having to constantly monitor things such as water quality, feeding, and cleaning.

Benefits: The AI tank provides automation for feeding, monitors water quality, and tracks the health of the fish. This makes for a more accessible and less stressful experience for casual fish owners.

User Group 2: Commercial Users

Description: Commercial users, such as pet stores, need to manage several tanks with different species of fish. The staff at these stores may not always be experts, making consistent care across many tanks challenging.

Needs Statement: They need a scalable solution that ensures consistent water quality, reduces labor costs, and keeps fish alive longer.

Benefits: AI automation will reduce manual monitoring and feeding, and maintain standardized care across tanks. This will save costs and increase reliability.

User Group 3: Enthusiasts and Hobbyists

Description: Experienced aquarium owners who might have more expensive fish and setups.

Needs Statement: They need accurate, detailed data and precise environmental controls to optimize their tank conditions, monitor fish health, and experiment with advanced setups.

Benefits: The AI system provides real-time feedback, such as temperature, water quality, and pH. This will allow enthusiasts to achieve higher success rates with their many difference fish, making advanced care more manageable.

Please include any user research documentation, empathy maps, or other artifacts as appendices.

2. Requirements, Constraints, And Standards

2.1. REQUIREMENTS & CONSTRAINTS

Functional requirements:

- Sensors:
 - pH sensor should read pH accurately within (+/-) 0.1 pH - **constraint**
 - Temperature sensor should read accurately within (+/-) 1°F - **constraint**
 - Turbidity sensor should read accurately within (+/-) 0.5% F*S - **constraint**
 - Dissolved oxygen sensor should read accurately within (+/-) 1 mg/L - **constraint**
- Feeding Mechanism:
 - Should hold up to a week's worth of fish food

User Experiential Requirements:

- Application:
 - Should have response time of no greater than 1 second - **constraint**
 - Should display any sensor range outside of their normal set value
 - Should allow user to schedule feeding intervals and times
 - Should monitor trends in sensor data and feeding patterns
 - Should display tank conditions, quality, and population statistics

Environmental Requirements:

- Electronics housing should be water-resistant
- Electronics should not interfere with water quality, dissolved oxygen, and other water metrics

Physical Requirements:

- Should fit inside the lid of the tank

Economic Requirements:

- The final product must cost less than \$500 to construct - **constraint**

Aesthetic Requirements:

- There should be minimal visible wires, boards, or electronics to the user

1.1. ENGINEERING STANDARDS

Standards Importance:

Engineering standards are important because they ensure safety, reliability, and consistency in design and production. They help engineers follow proven methods, make systems compatible with one another, and maintain quality across different projects. Standards also make communication clearer between teams and help prevent errors or failures in engineering work.

ISO/IEC 42001:2023 - Information Technology - Artificial Intelligence - Management System

- **ISO/IEC 42001:2023:** This is an international standard that specifies requirements for establishing, implementing, and maintaining an artificial intelligence system. It is intended for entities that are designing or utilizing AI-based products. It is important because it is the world's first AI management system standard.

IEEE 830-1998: Recommended Practice for Software Requirements Specifications

- **IEEE 830-1998:** This standard provides structured guidelines for writing and organizing the requirements for a software system. It helps ensure that every functionality, such as water monitoring, feeding automation, and data logging, is clearly described and testable. The goal is to improve communication between developers, engineers, and stakeholders so that everyone shares the same understanding of what the system must do.

IEEE 1016-2009: Systems and Software Engineering — Software Design Descriptions

- **IEEE 1016-2009:** This standard outlines how to properly document a system's software design. It includes how to describe system architecture, interfaces between modules, data flows, and logic control. For the fish tank project, this means clearly defining how sensors, microcontrollers, and the AI module interact, for example, how temperature or turbidity data are processed and used to adjust pumps or lights automatically.

Project Relevance:

- **IEEE 830-1998** is relevant because it helps define clear software requirements for each automated task, such as feeding, lighting, and water control. This ensures the system functions as intended and reduces design errors.
- **IEEE 1016-2017** is important for documenting how sensors, controllers, and the AI system work together. It supports easier debugging and future updates.
- **ISO/IEC 42001:2023** is relevant because it promotes safe and responsible use of AI. It helps ensure the AI makes reliable decisions and operates ethically within the fish tank system.

Additional Standards:

- IEEE 29119-2021: Software and Systems Engineering - Software Testing - Test Processes
- IEEE 16085-2020: Systems and Software Engineering - Life Cycle Processes - Risk Management

Modifications:

- IEEE 29119-2021: To incorporate this standard into our project, we will lay out a testing scope to make sure all components are properly tested. For example, our testing scope could be sensors, AI population tracking, the feeding system, and water quality calculations.
- IEEE 16085-2020: To incorporate this standard into our project, we will create a list of risks to make sure that all are accounted for. For example, a risk could be a wrong turbidity sensor reading. To account for this, we will calibrate the sensor weekly as well as get rid of outliers, and take averages.

3 Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

We are using agile for our project management style. While we have a general idea right now of the project goals, there is a lot of room for those goals to change. It would be a mistake for us to lock ourselves into a design when we aren't 100% sure of all of the possibilities with the project. Being able to quickly pivot to add a new feature or change our focus is required in our project, and agile is the best project management style to support these rapid changes.

We are tracking progress and assigning roles in Plaky, which is an issue-tracking board like Trello. The entire team can see what each person is currently working on and get a general idea of their progress. If more detailed information is needed, we can also look at git commits in GitLab to see exactly what changes each person is making. We chose to do this because not all aspects of our project can be tracked through git commits, as there are some non-programming problems that we will have to solve. These include automated fish feeding, water cleaning, and water temperature control. While there will most likely be some code associated with solving these problems, there is more work that will need to be accomplished that cannot be tracked through git.

3.2 TASK DECOMPOSITION

With the help of Plaky mentioned above, we are able to split the project into smaller tasks and subtasks. Currently, our team has organized a file-tree-like structure to split our tasks:

1. Hardware
 - a. Arduino HTTP requests
 - b. Sensor setup
 - c. Feeding system design
 - d. Setup camera
 - e. Lid construction
2. Software
 - a. Frontend
 - i. landing page
 - ii. HTTP receive

- b. Backend
 - i. Receiving HTTP
 - ii. Database integration
 - iii. Server hosting
- c. AI
 - i. AI population Tracking
 - ii. Utilize a camera with an AI model

Each of these tasks may be broken down further into smaller subtasks in order to solve individual problems. The linear timeline of completion of tasks is described in section 3.4

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

To organize our progress, we have divided our tasks into three main categories: Hardware, Software, and AI. Each task within these categories has a certain milestone and metric/evaluation criteria we use to measure our project's progress.

Hardware

1. Task 1: Arduino HTTP requests
 - a. Milestone: Arduino successfully sends GET/POST requests to the server
 - b. Evaluation Criteria: > 95% requests received successfully
2. Task 2: Sensor setup
 - a. Milestone: Temperature, turbidity, pH, and oxygen sensors calibrated and stable
 - b. Evaluation Criteria: Sensor readings within +/- 5% of reference values
3. Task 3: Feeding system design
 - a. Milestone: Servo motor dispenses food reliably on schedule
 - b. Evaluation Criteria: < 5% deviation from scheduled feed times over 1 week
4. Task 4: Camera setup
 - a. Milestone: Camera captures clear images at defined intervals
 - b. Evaluation Criteria: > 90% usable image rate over 100 captures
5. Task 5: Lid construction
 - a. Milestone: Lid securely houses sensors and camera with waterproofing
 - b. Evaluation Criteria: Passes leak and sensor stability tests

Software

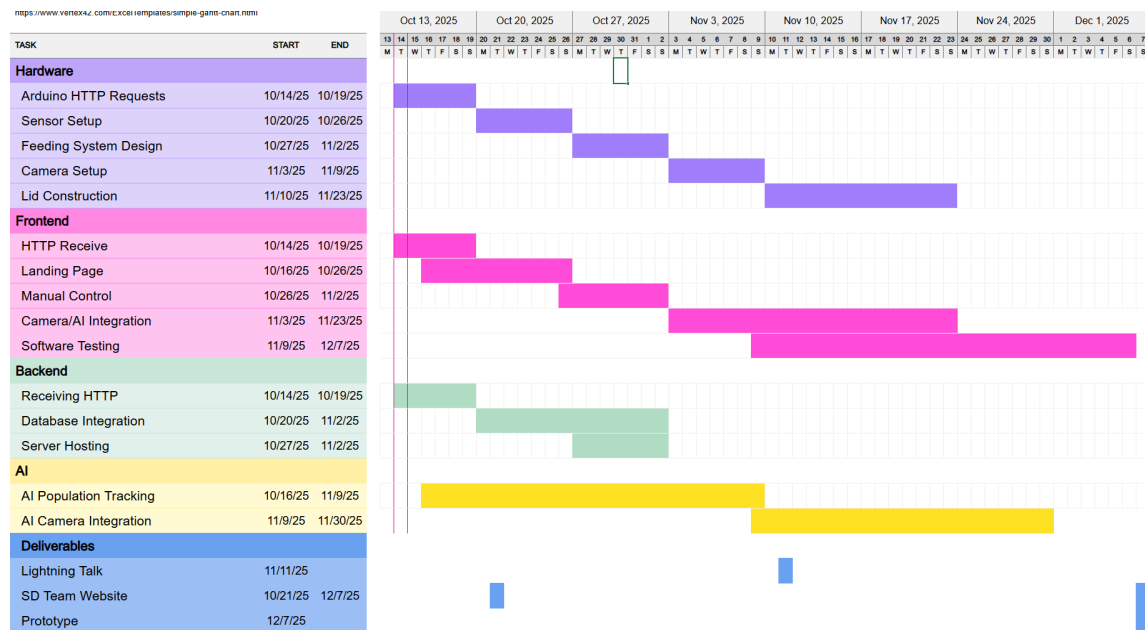
1. Task 1: Frontend landing page
 - a. Milestone: Responsive UI with tank status and manual controls
 - b. Evaluation Criteria: Loads in < 2s, compatible on mobile and desktop
2. Task 2: Frontend HTTP requests received
 - a. Milestone: Frontend receives and displays real-time sensor data
 - b. Evaluation Criteria: > 95% data sync accuracy
3. Task 3: Backend HTTP handling
 - a. Milestone: Server receives and parses Arduino data
 - b. Evaluation Criteria: > 95% parse success rate over 100 transmissions
4. Task 4: Backend database integration
 - a. Milestone: Sensor data stored and retrievable from the database
 - b. Evaluation Criteria: < 1s query latency, > 99% data integrity

5. Task 5: Backend server hosting
 - a. Milestone: Server hosted and accessible via local network
 - b. Evaluation Criteria: > 99% uptime

AI

1. Task 1: AI population tracking
 - a. Milestone: AI model detects and counts fish in tank images
 - b. Evaluation Criteria: > 80% accuracy on annotated test set
2. Task 2: AI camera integration
 - a. Milestone: AI processes live camera feed for fish tracking
 - b. Evaluation Criteria: < 2s latency per frame, > 80% frame classification rate

3.4 PROJECT TIMELINE/SCHEDULE



Prototyping and testing will be ongoing throughout the project. Our project management style is agile, so this schedule will be updated as it is necessary to address the requirements of our project.

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

For each task, identify all salient risks (certain performance target may not be met; certain tool may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5 and each high-severity risk, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

An agile project can associate risks and risk mitigation with each sprint.

1. Arduino HTTP requests
 - a. Risks
 - i. Poor signal strength could cause the backend requests to the Arduino to fail (P = .6)
 - ii. Arduino may have limited memory, and the pictures taken by the camera might be too large (P = .6)
 - b. Mitigation Plan
 - i. Use retries so that it will try multiple times and make sure the WiFi connectivity is strong.
 - ii. We could switch out the Arduino for a more expensive Raspberry Pi that has more memory.
2. Sensor setup
 - a. Risks
 - i. Sensors may need calibration after a certain period of time (P = .6)
 - ii. Electrical noise could cause readings to not be accurate (P = .6)
 - b. Mitigation Plan
 - i. Implement a warning email that is sent to the user once every certain period of time to recalibrate sensors
 - ii. We could replace the cheap sensors that we bought with higher-grade sensors
3. Feeding system design
 - a. Risks
 - i. The servo motor may misalign (P = .6)
 - ii. Food could run out without the user noticing. (P = .8)
 - b. Mitigation Plan
 - i. Upgrade the servo motor to a higher-grade servo
 - ii. Add some sort of weight sensor, or calculate how many times the fish can be fed with the container of food. Notify user when out
4. Setup camera
 - a. Risks
 - i. The camera could capture blurry or dark images (P = .7)
 - ii. The camera may not be the most compatible with Arduino (P = .2)
 - b. Mitigation Plan
 - i. Test with multiple lighting setups
5. Lid construction
 - a. Risks
 - i. Failure to waterproof could lead to electrical damage (P = .3)
 - ii. Parts and wires could come loose (P = .6)
 - b. Mitigation Plan
 - i. Make 3D printed mounts to secure equipment properly
6. Landing page
 - a. Risks
 - i. UI may not look correct on mobile (P = .9)
 - ii. Load time might be longer than 2 seconds (P = .6)
 - b. Mitigation Plan
 - i. Use a responsive framework like React
 - ii. Optimize scripts or lessen picture quality

7. HTTP receive
 - a. Risks
 - i. Websocket implementation errors (P = .2)
8. Receiving HTTP
 - a. Risks
 - i. Overload if too many requests are being made (P = .6)
 - b. Mitigation Plan
 - i. Grab data from a database instead of sending a request to the Arduino. Arduino updates the database every x seconds.
9. Database integration
 - a. Risks
 - i. Slow queries as data piles up (P = .6)
 - b. Mitigation Plan
 - i. Use a lighter-weight database like SQLite
10. Server hosting
 - a. Risks
 - i. Server overheating (P = .1)
11. AI population Tracking
 - a. Risks
 - i. The model does not perform well due to a small dataset (P = .7)
 - ii. False positives like reflections or bubbles (P = .7)
 - b. Mitigation Plan
 - i. Use object detection libraries that exist and have been extensively trained
 - ii. Use background subtraction and average out the fish count over multiple images
12. Utilize a camera with an AI model
 - a. Risks
 - i. Latency > 2 seconds because of AI response time (P = .9)
 - b. Mitigation Plan
 - i. Use a machine learning model instead, which is much faster

3.6 PERSONNEL EFFORT REQUIREMENTS

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.

Task	Explanation	Estimated Hours
Hardware Design & Assembly	Select and connect sensors (pH, turbidity, temperature, DO), actuators (heaters, pumps, feeders), and power supply circuits	30

AI & Computer Vision Development	Develop and train models for fish counting, growth tracking, and predictive analysis of water quality trends	20
Web Dashboard Development	Design and implement a web-based interface for monitoring and alerts	40
System Integration & Testing	Combine hardware, firmware, and software modules, test communication, reliability, and automation features.	40
Calibration & Optimization	Fine-tune sensors, control thresholds, and AI model accuracy under real conditions.	20

3.7 OTHER RESOURCE REQUIREMENTS

Identify the other resources aside from financial (such as parts and materials) that will be required to complete the project.

Hardware Components:

- Arduino Uno
- Turbidity Sensor
- Temperature Sensor
- Dissolved Oxygen Sensor
- pH Sensor
- Servo Motor
- Feeding Mechanism
- Power Supplies

Software Components:

- Arduino IDE
- .NET Core Web API (Backend)
- React JS/ Tailwind (Frontend)
- SQLite (Database)

Fish Tank Components:

- Tank
- Air Pump
- Filter

- Water Heater