

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title:
Water Sampling with Applied Microcontrollers

Team Members: Jesus Elespuru, Angie Martinez, Steven Luque, Christopher Perez

Product Owner(s): Jesus Elespuru

Mentor(s): Dr. Gregory Murad Reis

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document presents the information necessary to gain a good understanding of ...

This project focuses on the design and development of an autonomous surface vehicle (ASV) system for real-time water quality monitoring in Biscayne Bay. The ASV addresses the limitations of traditional methods, such as manual sampling and fixed monitoring stations, by offering scalable, efficient, and continuous data collection. Equipped with water quality sensors, GPS navigation, and a solar-powered energy system, the ASV autonomously navigates predefined routes while collecting critical environmental data, including temperature, pH, salinity, and dissolved oxygen levels. The system transmits data in real-time to a cloud platform, where it is processed, stored, and visualized for user access via web and mobile applications.

Testing and field trials validated the ASV's performance in real-world conditions, demonstrating accurate sensor readings, reliable navigation, energy efficiency, and durability. The ASV provides a cost-effective and sustainable solution for large-scale environmental monitoring, empowering researchers and policymakers with actionable insights to preserve and manage aquatic ecosystems. This project highlights the potential of marine robotics in advancing ecological sustainability and improving environmental decision-making.

Table of Contents

INTRODUCTION	5
CURRENT SYSTEM	5
PURPOSE OF NEW SYSTEM	5
USER STORIES	6
IMPLEMENTED USER STORIES	6
PENDING USER STORIES	13
PROJECT PLAN	14
HARDWARE AND SOFTWARE RESOURCES	14
SPRINTS PLAN	14
<i>Sprint 1</i>	14
<i>Sprint 2</i>	16
<i>Sprint 3</i>	19
<i>Sprint 4</i>	21
<i>Sprint 5</i>	23
<i>Sprint 6</i>	26
<i>Sprint 7</i>	29
SYSTEM DESIGN	32
ARCHITECTURAL PATTERNS	32
SYSTEM AND SUBSYSTEM DECOMPOSITION	32
DEPLOYMENT DIAGRAM	34
DESIGN PATTERNS	35
SYSTEM VALIDATION	35
GLOSSARY	36
APPENDIX	37
APPENDIX A - UML DIAGRAMS	38
APPENDIX B - USER INTERFACE DESIGN	38
APPENDIX C - SPRINT REVIEW REPORTS	43
APPENDIX D - MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/ WISHLIST DOCUMENT AND OTHER DOCUMENTS	49
REFERENCES	51

INTRODUCTION

Biscayne Bay, a vital ecosystem for South Florida, faces increasing threats from pollution, rising temperatures, and human activity, which adversely impact water quality and marine life. Current methods for monitoring water quality often rely on manual sampling, which is labor-intensive, costly, and limited in coverage. This creates a significant gap in real-time data collection, making it difficult to identify trends or respond to environmental changes effectively. To address these challenges, there is a pressing need for an efficient, scalable, and autonomous solution that can provide comprehensive, high-resolution data to support environmental preservation efforts.

Current System

The current system for water quality monitoring in Biscayne Bay relies primarily on manual sampling and fixed monitoring stations. These methods are time-consuming, require significant human effort, and are limited in spatial and temporal coverage. Fixed stations provide localized data but fail to capture dynamic changes across the bay, while manual sampling is sporadic and lacks real-time capability. Additionally, deploying and maintaining these systems can be expensive and resource-intensive, further limiting their effectiveness. As a result, critical changes in water quality can go unnoticed, delaying responses to environmental threats and reducing the overall effectiveness of conservation efforts.

Purpose of New System

The purpose of the new autonomous surface vehicle (ASV) system is to provide an efficient, scalable, and cost-effective solution for real-time water quality monitoring in Biscayne Bay. By addressing the limitations of traditional methods, such as manual sampling and fixed monitoring stations, the ASV aims to deliver comprehensive and high-resolution data across large and dynamic water environments. Equipped with advanced navigation, solar-powered sustainability, and real-time data transmission capabilities, the system empowers researchers and environmentalists with actionable insights to support the preservation and management of aquatic ecosystems. The ASV's autonomous operation minimizes human intervention while reducing costs and improving the accuracy and frequency of environmental monitoring efforts.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

- User Story 1-0 Discuss and plan schematics for microcontroller and additional electrical components
 - Tasks: Be able to draw a schematic to understand the connections and wiring of the electrical components
- User Story 1-1 Communicate with Professor Gregory Reis for mentorship to be able to work with his research team to further develop the project.
 - Tasks: Reaching out to the professor and discussing future plans for the project
- User Story 1-2 Communicate with team members to allocate a weekly schedule to meet up in person
 - Tasks: Be able to have a reoccurring in-person meeting with all, or most, available team members.
- User Story 1-3 Discuss with group members which programming tools and features we will be using and implementing
 - Tasks: Agree on a common programming language and ideas for features to further develop the project.
- User Story 1-4 After working on electrical schematics, order the electrical components so they will arrive on time.
 - Tasks: Research websites and stores to order the electrical components, extra components are considered in case of emergency

User Story 2-0 Configure GPS module to work with Raspberry Pi for water sampling system

- Tasks: Set up the GPS module, install necessary libraries, and test GPS data output on the Raspberry Pi.

User Story 2-1 Write code to integrate GPS data with water sampling data

- Tasks: Modify Raspberry Pi code to collect GPS coordinates along with water quality data and log them together.

User Story 2-2 Optimize GPS data logging for accuracy and performance

- Tasks: Refactor code to ensure GPS data is captured accurately and at regular intervals without affecting system performance.

User Story 2-3 Test GPS accuracy in various environmental conditions

- Tasks: Perform tests to validate GPS data collection in different locations, such as open fields, near water bodies, and urban areas.

User Story 2-4 Design a 3D-printed case for the Raspberry Pi and GPS module

- Tasks: Create a 3D model that accommodates the Raspberry Pi, GPS module, and necessary wiring while ensuring protection from the elements.

User Story 2-5 3D print and assemble the case for the Raspberry Pi system

- Tasks: Print the designed case, assemble the Raspberry Pi, GPS module, and other components inside it, and check for fitting issues.

User Story 2-6 Test the waterproof properties of the 3D-printed case

- Tasks: Ensure the 3D-printed case protects the electronics from water exposure during field deployments.

User Story 2-7 Optimize the case design for heat dissipation and portability

- Tasks: Modify the case to improve airflow around the Raspberry Pi and make the system easier to transport.

User Story 2-8 Optimize Raspberry Pi code for efficient sensor data collection

- Tasks: Review the Raspberry Pi code and reduce inefficiencies, ensuring that the system runs faster and with less power consumption.

User Story 2-9 Implement multithreading to improve Raspberry Pi performance

- Tasks: Modify the code to use multithreading, allowing the Raspberry Pi to handle multiple tasks such as GPS data and sensor readings concurrently.

User Story 2-10 Test Raspberry Pi system under heavy load for performance bottlenecks

- Tasks: Run stress tests with all sensors and modules active, analyze performance, and identify code bottlenecks for further optimization.

User Story 2-11 Enhance error handling in the Raspberry Pi code

- Tasks: Improve error detection and handling in the code to ensure the system recovers from issues such as GPS signal loss or sensor malfunctions

User Story 2-12 Install a Raspberry Pi touch Display

- Tasks: Install a raspberry pi touch display for easier access to the raspberry configuration.

User Story 2-13 Add real-time feedback for system status on Raspberry Pi display

- Tasks: Update the Raspberry Pi's code to display status information like GPS lock, sensor readings, and battery life on a connected display.

User Story 3-0 Set up GPS module for accurate data output

- Tasks: Set up the GPS module and configure it for accurate data output.

User Story 3-1 Install necessary drivers and libraries for GPS module

- Tasks: Install necessary drivers and libraries on the device to enable GPS functionality.

User Story 3-2 Test GPS data collection in real-time scenarios

- Tasks: Test GPS data collection and confirm accuracy in real-time scenarios.

User Story 3-3 Write code to integrate GPS data into system software

- Tasks: Write code to integrate GPS data into the system software.

User Story 3-4 Log GPS coordinates alongside system outputs

- Tasks: Ensure GPS coordinates are logged alongside system outputs for tracking purposes.

User Story 3-5 Test integration of GPS module with other system components

- Tasks: Test the integration with other system components, ensuring smooth data flow.

User Story 3-6 Review and finalize the design of the mounting bracket

- Tasks: Review mounting bracket design for structural integrity and fit.

User Story 3-7 Finalize materials and dimensions for mounting bracket production

- Tasks: Finalize materials and dimensions for manufacturing the mounting bracket.

User Story 3-8 Test the mounting bracket under simulated conditions

- Tasks: Test the bracket under simulated environmental conditions to ensure stability.

User Story 3-9 Prepare case for screen installation

- Tasks: Prepare the case to fit the screen securely, ensuring proper alignment.
- User Story 3-10 Install the screen into the case
- Tasks: Install the screen in the case without damaging internal components.
- User Story 3-11 Test screen functionality and durability post-installation
- Tasks: Test screen visibility, touchscreen sensitivity, and overall durability post-installation.
- User Story 3-12 Design internal layout for component placement
- Tasks: Design an internal layout for optimal placement of components (GPS module, screen, Raspberry Pi, etc.).
- User Story 3-13 Install and secure internal components in the case
- Tasks: Install and secure all internal components within the case.
- User Story 3-14 Verify power and data connections between internal components
- Tasks: Verify connections (power, data) between components to ensure functionality.
- User Story 4-0 Finalize the design and production of the mounting brackets
- Tasks: Finalize materials and dimensions for manufacturing the mounting brackets.
- User Story 4-1 Install internal components into the case
- Tasks: Install and secure internal components such as the Raspberry Pi, GPS module, and screen within the case.
- User Story 4-2 Install external components and brackets to the case
- Tasks: Attach external components and finalize mounting brackets.
- User Story 4-3 Prepare and cut wires for internal connections
- Tasks: Measure and cut wires to the appropriate lengths for power and data connections between internal components.
- User Story 4-4 Mount the case onto the boat
- Tasks: Secure the case onto the boat using the finalized brackets and ensure stability during movement.
- User Story 4-5 Test stability of mounted case on the boat
- Tasks: Perform stability tests to ensure the case remains secure during operation and movement of the boat.
- User Story 4-6 Connect power supply to internal components
- Tasks: Connect the power supply to the internal components and test the system's power flow.
- User Story 4-7 Test all components after installation in the case
- Tasks: Perform a full system check to ensure all components (GPS, screen, Raspberry Pi) are functioning after installation in the case.
- User Story 4-8 Secure all wiring inside the case
- Tasks: Organize and secure all wires inside the case to prevent movement and interference.

User Story 4-9 Waterproof and weatherproof the case

- Tasks: Apply necessary seals and coatings to make the case waterproof and weatherproof for use on the boat.

User Story 4-10 Test the case and internal components in a water-based environment

- Tasks: Test the system on the boat in a water environment to ensure functionality under real conditions.

User Story 4-11 Finalize and document the installation process

- Tasks: Create detailed documentation for the installation process, including steps for mounting, wiring, and testing.

User Story 5-0 Check the temperature of the Raspberry Pi

- Tasks: Monitor and record the temperature of the Raspberry Pi to ensure it operates within safe limits.

User Story 5-1 Install the GPS module in the case

- Tasks: Securely install the GPS module within the case to ensure accurate data capture.

User Story 5-2 Test GPS module functionality

- Tasks: Run tests on the GPS module to confirm it is accurately capturing and transmitting location data.

User Story 5-3 Design and build a new piece for the box

- Tasks: Work on creating a new piece for the box to accommodate additional components or improve structure.

User Story 5-4 Integrate GPS data with system software

- Tasks: Modify the software to capture and log GPS data in synchronization with other system outputs.

User Story 5-5 Ensure stable operation of the GPS module during temperature variations

- Tasks: Check GPS functionality at various temperatures to ensure stable performance in different conditions.

User Story 5-6 Secure the Raspberry Pi against overheating

- Tasks: Implement cooling measures, such as adding heat sinks or fans, to prevent the Raspberry Pi from overheating.

User Story 5-7 Install additional supports for the GPS module inside the case

- Tasks: Add extra supports to keep the GPS module stable and secure during movement or vibrations.

User Story 5-8 Perform stress tests on the new box piece

- Tasks: Test the new piece to ensure it meets durability and performance requirements under simulated conditions.

User Story 5-9 Calibrate GPS data accuracy

- Tasks: Perform calibration to enhance GPS accuracy, adjusting configurations as needed for precise data.

User Story 5-10 Document GPS installation and testing process

- Tasks: Create a guide documenting the steps for GPS installation, calibration, and testing procedures.

User Story 5-11 Verify system performance with GPS and temperature checks

- Tasks: Run a final system check to ensure both GPS and Raspberry Pi operate smoothly under normal temperature ranges.

User Story 6-0 Document the wiring layout inside the case

- Tasks: Create a detailed diagram and description of the wiring setup inside the case for future reference.

User Story 6-1 Fix minor wiring issues

- Tasks: Identify and resolve any loose or misaligned wires to ensure stable connections and prevent shorts.

User Story 6-2 Organize internal components in the case

- Tasks: Reorganize and secure internal components for optimal space utilization and structural stability.

User Story 6-3 Document the final installation process

- Tasks: Write up detailed steps for the entire installation process, including wiring, component placement, and testing.

User Story 6-4 Address and resolve case fitting issues

- Tasks: Adjust case components to eliminate any misalignment or obstruction that affects component fit or stability.

User Story 6-5 Identify and fix minor bugs in the software

Tasks:

- Run diagnostic tests, identify bugs in the system software, and implement fixes for improved functionality.

User Story 6-6 Test wiring connections for consistency

- Tasks: Verify that all wiring connections maintain consistent power and data transmission, making adjustments as needed.

User Story 6-7 Document known bugs and fixes

- Tasks: Create a log of known software and hardware bugs, along with the implemented solutions.

User Story 6-8 Add labels to internal components and wiring

- Tasks: Label wires and components inside the case to make future maintenance and troubleshooting easier.

User Story 6-9 Test case internals for interference

- Tasks: Run tests to ensure that components within the case do not interfere with each other's operation.

User Story 6-10 Perform final quality checks on case internals

- Tasks: Conduct a thorough review of all internal components and connections to ensure everything is secure and functioning.

User Story 6-11 Compile full project documentation

- Tasks: Organize and finalize all documentation for the project, covering setup, wiring, installation, testing, and troubleshooting.

User Story 7-0 Develop a draft layout for the project poster

- Tasks: Create an initial draft layout for the project poster, including sections for problem statement, methodology, and results.

User Story 7-1 Finalize the content for the project poster

- Tasks: Edit and refine the text, visuals, and layout of the poster to ensure clarity and professionalism.

User Story 7-2 Print the final version of the project poster

- Tasks: Coordinate with a printing service to produce the final poster in high quality and proper dimensions.

User Story 7-3 Prepare a presentation outline for the showcase

- Tasks: Develop an outline for the presentation, highlighting key project aspects such as objectives, methods, and outcomes.

User Story 7-4 Practice presenting the project for the showcase

- Tasks: Conduct multiple rehearsals to refine delivery, timing, and responses to potential questions.

User Story 7-5 Gather and organize project materials for the showcase

- Tasks: Collect all necessary materials, including the poster, device, and any supplementary items, for display at the showcase.

User Story 7-6 Test all project components for functionality before the showcase

- Tasks: Perform a final system check to ensure all hardware and software components are working correctly.

User Story 7-7 Create a detailed user guide for the project

- Tasks: Develop a comprehensive user guide that explains setup, usage, and maintenance of the system.

User Story 7-8 Document lessons learned and future improvements

- Tasks: Write a report summarizing challenges faced, lessons learned, and potential future enhancements for the project.

User Story 7-9 Prepare a Q&A document for the showcase

- Tasks: Create a list of anticipated questions and well-prepared answers to address inquiries during the showcase.

User Story 7-10 Set up the project display at the showcase venue

- Tasks: Arrange the poster, system, and other materials in a visually appealing and functional setup at the event.

User Story 7-11 Engage with showcase attendees and gather feedback

- Tasks: Present the project to attendees, answer questions, and collect feedback for future improvements.

Pending User Stories

User Story 5-7 Install additional supports for the GPS module inside the case

- Tasks: Add extra supports to keep the GPS module stable and secure during movement or vibrations.

User Story 5-8 Perform stress tests on the new box piece

- Tasks: Test the new piece to ensure it meets durability and performance requirements under simulated conditions.

User Story 5-9 Calibrate GPS data accuracy

- Tasks: Perform calibration to enhance GPS accuracy, adjusting configurations as needed for precise data.

User Story 5-10 Document GPS installation and testing process

- Tasks: Create a guide documenting the steps for GPS installation, calibration, and testing procedures.

User Story 5-11 Verify system performance with GPS and temperature checks

- Tasks: Run a final system check to ensure both GPS and Raspberry Pi operate smoothly under normal temperature ranges.

User Story 6-1 Fix minor wiring issues

- Tasks: Identify and resolve any loose or misaligned wires to ensure stable connections and prevent shorts.

User Story 6-2 Organize internal components in the case

- Tasks: Reorganize and secure internal components for optimal space utilization and structural stability.

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- **Raspberry Pi 5**
- **Ender 3 3D Printer**
- **EXO 2**
- **EXO 2 Signal Adapter**
- **EXO 2 3-M Cable**
- **SolidWorks**
- **Python**
- **Streamlit**
- **QgroundControl**
- **Ublox-7**
- **Github**
- **Computers**
- **Cura Maker**

Sprints Plan

Sprint 1

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 2:25 PM

End time: 2:30 PM

We picked the following functions/features to use/test during this sprint, which were approved by Prof. Masoud Sadjadi. Here is the list of user stories and/or tasks ordered based on their priority.

User Story 1-0 Discuss and plan schematics for microcontroller and additional electrical components

- Tasks: Be able to draw a schematic to understand the connections and wiring of the electrical components

User Story 1-1 Communicate with Professor Gregory Reis for mentorship to be able to work with his research team to further develop the project.

- Tasks: Reaching out to the professor and discussing future plans for the project

User Story 1-2 Communicate with team members to allocate a weekly schedule to meet up in person

- Tasks: Be able to have a reoccurring in-person meeting with all, or most, available team members.

User Story 1-3 Discuss with group members which programming tools and features we will be using and implementing

- Tasks: Agree on a common programming language and ideas for features to further develop the project.

User Story 1-4 After working on electrical schematics, order the electrical components so they will arrive on time.

- Tasks: Research websites and stores to order the electrical components, extra components are considered in case of emergency

The team members indicated their willingness to work on the following user stories/tasks:

Angie Martinez

- User Story 1-2

Jesus Elespuru

- User Story 1-0
- User Story 1-4

Steven Luque

- User Story 1-1 Darniel Padron
- User Story 1-3

Christopher Perez

- User Story 1-0
- User Story 1-1

- User Story 1-2
- User Story 1-3
- User Story 1-4

Sprint 2

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 6:00 PM

End time: 8:00 PM

We picked the following Nextcloud functions/features to use/test during this sprint, which were approved by Prof. Masoud Sadjadi. Here is the list of user stories and/or tasks ordered based on their priority.

User Story 2-0 Configure GPS module to work with Raspberry Pi for water sampling system

- **Tasks:** Set up the GPS module, install necessary libraries, and test GPS data output on the Raspberry Pi.

User Story 2-1 Write code to integrate GPS data with water sampling data

- **Tasks:** Modify Raspberry Pi code to collect GPS coordinates along with water quality data and log them together.

User Story 2-2 Optimize GPS data logging for accuracy and performance

- **Tasks:** Refactor code to ensure GPS data is captured accurately and at regular intervals without affecting system performance.

User Story 2-3 Test GPS accuracy in various environmental conditions

- **Tasks:** Perform tests to validate GPS data collection in different locations, such as open fields, near water bodies, and urban areas.

User Story 2-4 Design a 3D-printed case for the Raspberry Pi and GPS module

- **Tasks:** Create a 3D model that accommodates the Raspberry Pi, GPS module, and necessary wiring while ensuring protection from the elements.

User Story 2-5 3D print and assemble the case for the Raspberry Pi system

- **Tasks:** Print the designed case, assemble the Raspberry Pi, GPS module, and other components inside it, and check for fitting issues.

User Story 2-6 Test the waterproof properties of the 3D-printed case

- **Tasks:** Ensure the 3D-printed case protects the electronics from water exposure during field deployments.

User Story 2-7 Optimize the case design for heat dissipation and portability

- **Tasks:** Modify the case to improve airflow around the Raspberry Pi and make the system easier to transport.

User Story 2-8 Optimize Raspberry Pi code for efficient sensor data collection

- **Tasks:** Review the Raspberry Pi code and reduce inefficiencies, ensuring that the system runs faster and with less power consumption.

User Story 2-9 Implement multithreading to improve Raspberry Pi performance

- **Tasks:** Modify the code to use multithreading, allowing the Raspberry Pi to handle multiple tasks such as GPS data and sensor readings concurrently.

User Story 2-10 Test Raspberry Pi system under heavy load for performance bottlenecks

- **Tasks:** Run stress tests with all sensors and modules active, analyze performance, and identify code bottlenecks for further optimization.

User Story 2-11 Enhance error handling in the Raspberry Pi code

- **Tasks:** Improve error detection and handling in the code to ensure the system recovers from issues such as GPS signal loss or sensor malfunctions.

User Story 2-12 Install a Raspberry Pi touch Display

- **Tasks:** Install a raspberry pi touch display for easier access to the raspberry configuration.

User Story 2-13 Add real-time feedback for system status on Raspberry Pi display

- **Tasks:** Update the Raspberry Pi's code to display status information like GPS lock, sensor readings, and battery life on a connected display.

The team members indicated their willingness to work on the following user stories/tasks:

Angie Martinez:

- User Story 2-2
- User Story 2-3
- User Story 2-5
- User Story 2-6
- User Story 2-7

Jesus Elespuru:

- User Story 2-0
- User Story 2-1
- User Story 2-2

- User Story 2-3
- User Story 2-4

Steven Luque:

- User Story 2-5
- User Story 2-7
- User Story 2-9
- User Story 2-10
- User Story 2-13

Darniel Padron:

- User Story 2-4
- User Story 2-8
- User Story 2-9
- User Story 2-10
- User Story 2-11

Christopher Perez:

- User Story 2-5
- User Story 2-7
- User Story 2-9
- User Story 2-10
- User Story 2-12

Sprint 3

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 6:00 PM

End time: 8:00 PM

We picked the following functions/features to use/test during this sprint, which were approved by Prof. Masoud Sadjadi. Here is the list of user stories and/or tasks ordered based on their priority.

User Story 3-0 Set up GPS module for accurate data output

Tasks:

- Set up the GPS module and configure it for accurate data output.

User Story 3-1 Install necessary drivers and libraries for GPS module

Tasks:

- Install necessary drivers and libraries on the device to enable GPS functionality.

User Story 3-2 Test GPS data collection in real-time scenarios

Tasks:

- Test GPS data collection and confirm accuracy in real-time scenarios.

User Story 3-3 Write code to integrate GPS data into system software

Tasks:

- Write code to integrate GPS data into the system software.

User Story 3-4 Log GPS coordinates alongside system outputs

Tasks:

- Ensure GPS coordinates are logged alongside system outputs for tracking purposes.

User Story 3-5 Test integration of GPS module with other system components

Tasks:

- Test the integration with other system components, ensuring smooth data flow.

User Story 3-6 Review and finalize the design of the mounting bracket

Tasks:

- Review mounting bracket design for structural integrity and fit.

User Story 3-7 Finalize materials and dimensions for mounting bracket production

Tasks:

- Finalize materials and dimensions for manufacturing the mounting bracket.

User Story 3-8 Test the mounting bracket under simulated conditions

Tasks:

- Test the bracket under simulated environmental conditions to ensure stability.

User Story 3-9 Prepare case for screen installation

Tasks:

- Prepare the case to fit the screen securely, ensuring proper alignment.

User Story 3-10 Install the screen into the case

Tasks:

- Install the screen in the case without damaging internal components.

User Story 3-11 Test screen functionality and durability post-installation

Tasks:

- Test screen visibility, touchscreen sensitivity, and overall durability post-installation.

User Story 3-12 Design internal layout for component placement**Tasks:**

- Design an internal layout for optimal placement of components (GPS module, screen, Raspberry Pi, etc.).

User Story 3-13 Install and secure internal components in the case**Tasks:**

- Install and secure all internal components within the case.

User Story 3-14 Verify power and data connections between internal components**Tasks:**

- Verify connections (power, data) between components to ensure functionality.

The team members indicated their willingness to work on the following user stories/tasks:

Angie Martinez:

- User Story 3-2
- User Story 3-3
- User Story 3-5
- User Story 3-6
- User Story 3-7

Jesus Elespuru:

- User Story 3-0
- User Story 3-1
- User Story 3-2
- User Story 3-3
- User Story 3-4

Steven Luque:

- User Story 3-5
- User Story 3-7
- User Story 3-9
- User Story 3-10
- User Story 3-13

Darniel Padron:

- User Story 3-4

- User Story 3-8
- User Story 3-9
- User Story 3-10
- User Story 3-11

Christopher Perez:

- User Story 3-5
- User Story 3-7
- User Story 3-9
- User Story 3-10
- User Story 3-12

Sprint 4

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 6:00 PM

End time: 8:00 PM

User Story 4-0 Finalize the design and production of the mounting brackets

Tasks:

- Finalize materials and dimensions for manufacturing the mounting brackets.

User Story 4-1 Install internal components into the case

Tasks:

- Install and secure internal components such as the Raspberry Pi, GPS module, and screen within the case.

User Story 4-2 Install external components and brackets to the case

Tasks:

- Attach external components and finalize mounting brackets.

User Story 4-3 Prepare and cut wires for internal connections

Tasks:

- Measure and cut wires to the appropriate lengths for power and data connections between internal components.

User Story 4-4 Mount the case onto the boat

Tasks:

- Secure the case onto the boat using the finalized brackets and ensure stability during movement.

User Story 4-5 Test stability of mounted case on the boat**Tasks:**

- Perform stability tests to ensure the case remains secure during operation and movement of the boat.

User Story 4-6 Connect power supply to internal components**Tasks:**

- Connect the power supply to the internal components and test the system's power flow.

User Story 4-7 Test all components after installation in the case**Tasks:**

- Perform a full system check to ensure all components (GPS, screen, Raspberry Pi) are functioning after installation in the case.

User Story 4-8 Secure all wiring inside the case**Tasks:**

- Organize and secure all wires inside the case to prevent movement and interference.

User Story 4-9 Waterproof and weatherproof the case**Tasks:**

- Apply necessary seals and coatings to make the case waterproof and weatherproof for use on the boat.

User Story 4-10 Test the case and internal components in a water-based environment**Tasks:**

- Test the system on the boat in a water environment to ensure functionality under real conditions.

User Story 4-11 Finalize and document the installation process**Tasks:**

- Create detailed documentation for the installation process, including steps for mounting, wiring, and testing.

The team members indicated their willingness to work on the following user stories/tasks:

Angie Martinez:

- User Story 3-2
- User Story 3-3
- User Story 3-5
- User Story 3-6
- User Story 3-7

Jesus Elespuru:

- User Story 3-0
- User Story 3-1
- User Story 3-2
- User Story 3-3
- User Story 3-4

Steven Luque:

- User Story 3-5
- User Story 3-7
- User Story 3-9
- User Story 3-10
- User Story 3-11

Darniel Padron:

- User Story 3-4
- User Story 3-8
- User Story 3-9
- User Story 3-10
- User Story 3-11

Christopher Perez:

- User Story 3-5
- User Story 3-7
- User Story 3-9
- User Story 3-10
- User Story 3-11

Sprint 5

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 6:00 PM

End time: 8:00 PM

User Story 5-0 Check the temperature of the Raspberry Pi

Tasks:

- Monitor and record the temperature of the Raspberry Pi to ensure it operates within safe limits.

User Story 5-1 Install the GPS module in the case**Tasks:**

- Securely install the GPS module within the case to ensure accurate data capture.

User Story 5-2 Test GPS module functionality**Tasks:**

- Run tests on the GPS module to confirm it is accurately capturing and transmitting location data.

User Story 5-3 Design and build a new piece for the box**Tasks:**

- Work on creating a new piece for the box to accommodate additional components or improve structure.

User Story 5-4 Integrate GPS data with system software**Tasks:**

- Modify the software to capture and log GPS data in synchronization with other system outputs.

User Story 5-5 Ensure stable operation of the GPS module during temperature variations**Tasks:**

- Check GPS functionality at various temperatures to ensure stable performance in different conditions.

User Story 5-6 Secure the Raspberry Pi against overheating**Tasks:**

- Implement cooling measures, such as adding heat sinks or fans, to prevent the Raspberry Pi from overheating.

User Story 5-7 Install additional supports for the GPS module inside the case**Tasks:**

- Add extra supports to keep the GPS module stable and secure during movement or vibrations.

User Story 5-8 Perform stress tests on the new box piece**Tasks:**

- Test the new piece to ensure it meets durability and performance requirements under simulated conditions.

User Story 5-9 Calibrate GPS data accuracy**Tasks:**

- Perform calibration to enhance GPS accuracy, adjusting configurations as needed for precise data.

User Story 5-10 Document GPS installation and testing process**Tasks:**

- Create a guide documenting the steps for GPS installation, calibration, and testing procedures.

User Story 5-11 Verify system performance with GPS and temperature checks**Tasks:**

- Run a final system check to ensure both GPS and Raspberry Pi operate smoothly under normal temperature ranges.

The team members indicated their willingness to work on the following user stories/tasks:

Angie Martinez:

- User Story 5-2
- User Story 5-3
- User Story 5-5
- User Story 5-6
- User Story 5-7

Jesus Elespuru:

- User Story 5-0
- User Story 5-1
- User Story 5-2
- User Story 5-3
- User Story 5-4

Steven Luque:

- User Story 5-5
- User Story 5-7
- User Story 5-9
- User Story 5-10
- User Story 5-11

Darniel Padron:

- User Story 5-4
- User Story 5-8
- User Story 5-9
- User Story 5-10
- User Story 5-11

Christopher Perez:

- User Story 5-5
- User Story 5-7
- User Story 5-9
- User Story 5-10
- User Story 5-11

Sprint 6

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 6:00 PM

End time: 8:00 PM

User Story 6-0 Document the wiring layout inside the case

Tasks:

- Create a detailed diagram and description of the wiring setup inside the case for future reference.
-

User Story 6-1 Fix minor wiring issues

Tasks:

- Identify and resolve any loose or misaligned wires to ensure stable connections and prevent shorts.
-

User Story 6-2 Organize internal components in the case

Tasks:

- Reorganize and secure internal components for optimal space utilization and structural stability.
-

User Story 6-3 Document the final installation process

Tasks:

- Write up detailed steps for the entire installation process, including wiring, component placement, and testing.
-

User Story 6-4 Address and resolve case fitting issues

Tasks:

- Adjust case components to eliminate any misalignment or obstruction that affects component fit or stability.

User Story 6-5 Identify and fix minor bugs in the software**Tasks:**

- Run diagnostic tests, identify bugs in the system software, and implement fixes for improved functionality.

User Story 6-6 Test wiring connections for consistency**Tasks:**

- Verify that all wiring connections maintain consistent power and data transmission, making adjustments as needed.

User Story 6-7 Document known bugs and fixes**Tasks:**

- Create a log of known software and hardware bugs, along with the implemented solutions.

User Story 6-8 Add labels to internal components and wiring**Tasks:**

- Label wires and components inside the case to make future maintenance and troubleshooting easier.

User Story 6-9 Test case internals for interference**Tasks:**

- Run tests to ensure that components within the case do not interfere with each other's operation.

User Story 6-10 Perform final quality checks on case internals**Tasks:**

- Conduct a thorough review of all internal components and connections to ensure everything is secure and functioning.

User Story 6-11 Compile full project documentation**Tasks:**

- Organize and finalize all documentation for the project, covering setup, wiring, installation, testing, and troubleshooting.
-

The team members indicated their willingness to work on the following user stories/tasks:

Angie Martinez:

- User Story 6-2
- User Story 6-3
- User Story 6-5
- User Story 6-6
- User Story 6-7

Jesus Elespuru:

- User Story 6-0
- User Story 6-1
- User Story 6-2
- User Story 6-3
- User Story 6-4

Steven Luque:

- User Story 6-5
- User Story 6-7
- User Story 6-9
- User Story 6-10
- User Story 6-11

Darniel Padron:

- User Story 6-4
- User Story 6-8
- User Story 6-9
- User Story 6-10
- User Story 6-11

Christopher Perez:

- User Story 6-5
- User Story 6-7
- User Story 6-9
- User Story 6-10
- User Story 6-11

Sprint 7

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 6:00 PM

End time: 8:00 PM

User Story 7-0 Develop a draft layout for the project poster**Tasks:**

- Create an initial draft layout for the project poster, including sections for problem statement, methodology, and results.
-

User Story 7-1 Finalize the content for the project poster**Tasks:**

- Edit and refine the text, visuals, and layout of the poster to ensure clarity and professionalism.
-

User Story 7-2 Print the final version of the project poster**Tasks:**

- Coordinate with a printing service to produce the final poster in high quality and proper dimensions.
-

User Story 7-3 Prepare a presentation outline for the showcase**Tasks:**

- Develop an outline for the presentation, highlighting key project aspects such as objectives, methods, and outcomes.
-

User Story 7-4 Practice presenting the project for the showcase**Tasks:**

- Conduct multiple rehearsals to refine delivery, timing, and responses to potential questions.
-

User Story 7-5 Gather and organize project materials for the showcase**Tasks:**

- Collect all necessary materials, including the poster, device, and any supplementary items, for display at the showcase.
-

User Story 7-6 Test all project components for functionality before the showcase**Tasks:**

- **Perform a final system check to ensure all hardware and software components are working correctly.**

User Story 7-7 Create a detailed user guide for the project**Tasks:**

- **Develop a comprehensive user guide that explains setup, usage, and maintenance of the system.**

User Story 7-8 Document lessons learned and future improvements**Tasks:**

- **Write a report summarizing challenges faced, lessons learned, and potential future enhancements for the project.**

User Story 7-9 Prepare a Q&A document for the showcase**Tasks:**

- **Create a list of anticipated questions and well-prepared answers to address inquiries during the showcase.**

User Story 7-10 Set up the project display at the showcase venue**Tasks:**

- **Arrange the poster, system, and other materials in a visually appealing and functional setup at the event.**

User Story 7-11 Engage with showcase attendees and gather feedback**Tasks:**

- **Present the project to attendees, answer questions, and collect feedback for future improvements.**

The team members indicated their willingness to work on the following user stories/tasks:

Angie Martinez:

- User Story 7-2
- User Story 7-3
- User Story 7-5
- User Story 7-6
- User Story 7-7

Jesus Elespuru:

- User Story 7-0
- User Story 7-1
- User Story 7-2
- User Story 7-3
- User Story 7-4

Steven Luque:

- User Story 7-5
- User Story 7-7
- User Story 7-9
- User Story 7-10
- User Story 7-11

Darniel Padron:

- User Story 7-4
- User Story 7-8
- User Story 7-9
- User Story 7-10
- User Story 7-11

Christopher Perez:

- User Story 7-5
- User Story 7-7
- User Story 7-9
- User Story 7-10
- User Story 7-11

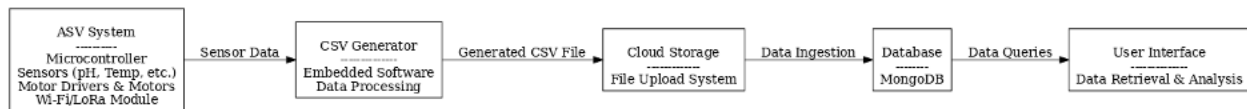
SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

1. Architectural Patterns

The architectural pattern of the ASV system follows a distributed architecture to achieve scalability, flexibility, and real-time operation. The system comprises three key layers: the edge layer, where the ASV collects and preprocesses sensor data; the communication layer, which handles real-time data transmission to the cloud via cellular or satellite networks; and the cloud layer, where data is stored, analyzed, and visualized for end-users. This distributed approach ensures seamless integration of hardware and software components while maintaining modularity for future upgrades. The architecture enables efficient processing and real-time access to environmental data, supporting machine learning-based predictive analytics for proactive decision-making.

Diagram 1. System Architecture of Project



2. System and Subsystem Decomposition

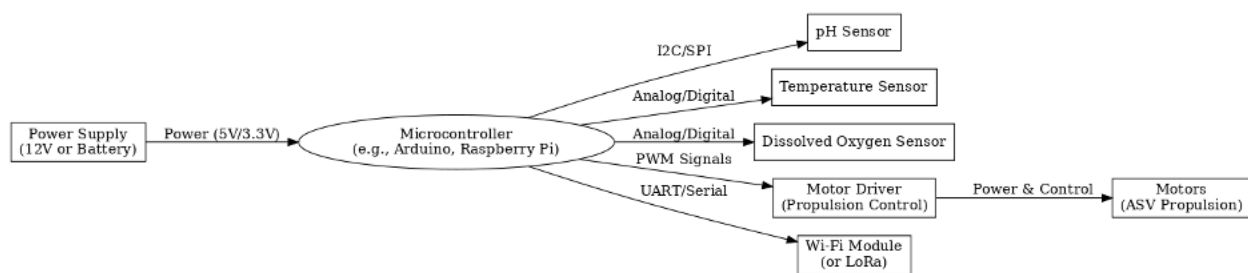
The ASV system is organized into multiple subsystems to simplify development, integration, and testing:

- **Hardware Subsystems:**
 - **Hull and Structure:** Provides stability and durability in various water conditions.
 - **Power System:** Includes solar panels and a rechargeable battery for sustainable energy supply.
 - **Sensor Suite:** Measures water quality parameters, such as temperature, pH, salinity, and dissolved oxygen.
 - **Navigation Components:** Consist of a GPS module and IMU for precise location tracking and orientation.
- **Software Subsystems:**
 - **Navigation Control:** Handles autonomous pathfinding and obstacle avoidance algorithms.
 - **Data Processing:** Preprocesses sensor data on the ASV and packages it for transmission.
 - **Communication Interface:** Manages real-time data transmission to the cloud via cellular or satellite networks.

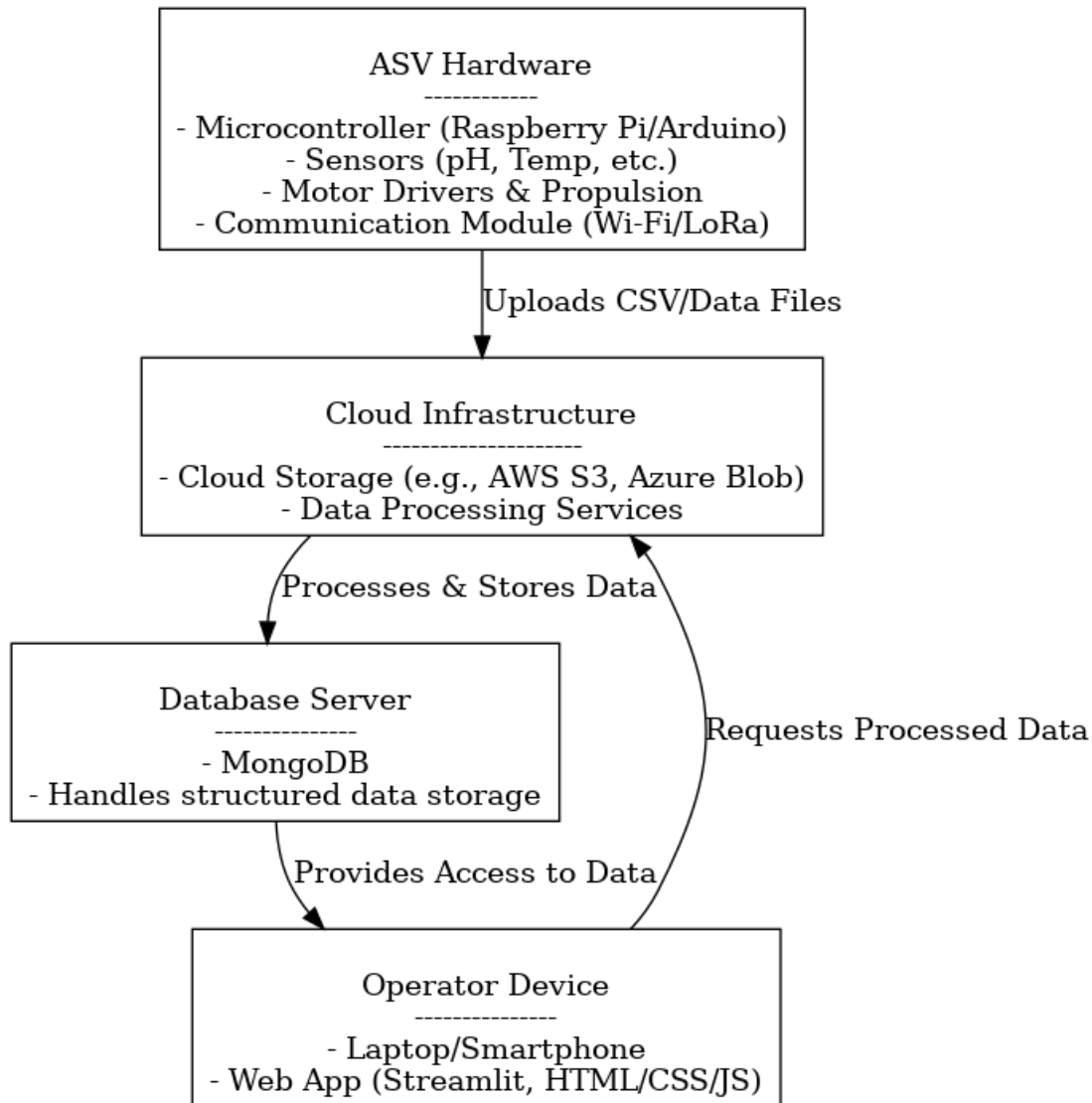
- Cloud Platform: Stores, analyzes, and visualizes data, enabling machine learning-based predictions.
- User Applications: Provides a web and mobile interface for real-time monitoring, historical data access, and ASV control.

Decomposing the system into subsystems ensures modularity, making it easier to troubleshoot, upgrade, and enhance individual components without affecting the overall system. This approach streamlines development while maintaining flexibility for future improvements.

Diagram 2. Electrical Schema



Deployment Diagram



Design Patterns

The ASV system utilizes the Observer Design Pattern to manage real-time data transmission and updates. The ASV acts as the subject, continuously gathering sensor data, while the cloud platform and user applications function as observers. Whenever new data is collected, the observers are notified and updated in real time, ensuring immediate synchronization between the ASV, cloud storage, and user interfaces. This design pattern simplifies the process of keeping all

system components aligned and ensures data consistency. Additionally, the Factory Design Pattern is employed in the initialization of sensors, allowing for easy replacement or addition of new sensor types without significant code changes, enhancing system flexibility and maintainability.

SYSTEM VALIDATION

System validation ensures that the autonomous surface vehicle (ASV) meets its intended objectives and performs reliably in real-world conditions. Validation was conducted through a combination of laboratory testing, simulation, and field trials.

- **Sensor Validation:** Each water quality sensor was calibrated and tested to ensure accurate readings for parameters like temperature, pH, salinity, and dissolved oxygen. Results were compared with manual sampling to confirm precision.
- **Navigation Validation:** The GPS and IMU were tested for accuracy in guiding the ASV along predefined routes, with obstacle avoidance mechanisms verified using controlled scenarios.
- **Data Transmission:** Real-time data transmission was validated by assessing the reliability and speed of communication between the ASV and the cloud platform under different network conditions.
- **Field Trials:** Comprehensive trials in Biscayne Bay validated the ASV's ability to navigate autonomously, withstand environmental challenges, and collect continuous, high-resolution data.
- **Predictive Analytics:** Machine learning models were tested using historical and live data to ensure accurate forecasting of water quality trends.

Successful validation confirmed the ASV's performance and readiness for deployment in environmental monitoring applications.

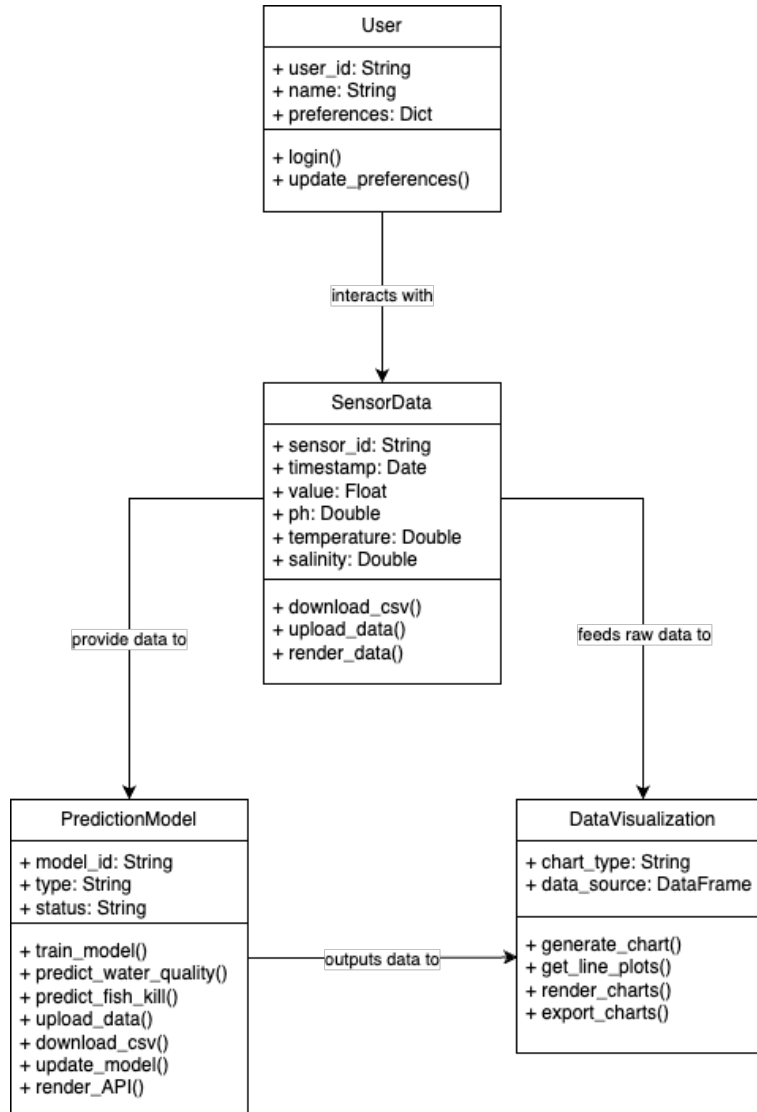
GLOSSARY

Term	Definition
User Management	Administration of user accounts, permissions, and access control
Integration Capabilities	The ability to integrate with other tools, applications, or services, enhancing its functionality and usability
Activity Logging	Recording and monitoring of user actions within the database
Autonomous Surface Vehicle (ASV)	A robotic vessel capable of navigating water environments without human intervention, used for tasks such as environmental monitoring
Water Quality Parameters	Key metrics used to assess water health, including temperature, pH, salinity, and dissolved oxygen levels
Obstacle Avoidance	A system integrated into the ASV that detects and avoids physical obstacles during navigation to ensure safety and operational efficiency
Real-Time Data Transmission	The process of sending data from the ASV to a cloud platform immediately after collection for instant access and analysis
Field Trials	Tests conducted in real-world environments, such as Biscayne Bay, to evaluate the performance and reliability of the ASV under natural conditions
Environmental Monitoring	The process of observing and measuring environmental conditions, such as water quality, to assess the health of an ecosystem

Data Integration	The process of combining data from multiple sources (e.g., sensors, cloud storage) into a unified platform for analysis and visualization
Pathfinding Algorithm	A computational method used to determine the most efficient route for the ASV to navigate through predefined areas
Predictive Analytics	The use of historical and real-time data, often with ML models, to forecast future trends in water quality or environmental changes
Machine Learning (ML)	A subset of artificial intelligence that uses algorithms to analyze data, identify patterns, and make predictions about future outcomes
Global Positioning System (GPS)	A satellite-based navigation system that provides real-time location data for the ASV's movement and tracking
Solar-Powered Energy System	A renewable energy system that uses solar panels to power the ASV and a rechargeable battery for energy storage during low-light conditions

APPENDIX

Appendix A - UML Diagrams



Appendix B - User Interface Design

Layout Overview

1. Header Section:

- **Title:** Display the project name (e.g., "FlowCast").
- **Subtitle:** A tagline such as "Real-Time Water Quality Monitoring and Predictions."
- **Navigation Menu:** Include options like "Home", "Global Dashboard," "Real-Time Analysis," "Background", "Sign Up" and "About Us."

2. Main Dashboard Section:

- **Real-Time Metrics:**
 - Highlight key water quality parameters (e.g., pH, temperature, dissolved oxygen, chlorophyll).
 - Use cards or widgets for each parameter, with color coding to indicate status (e.g., green for safe, red for critical).
- **Geospatial Map:**
 - Display an interactive map with markers for monitoring locations.
 - Markers show the current status or alerts for each location.
- **Trends and Charts:**
 - Include line charts, bar charts, and heatmaps for historical data and predictive trends.
- **Alerts and Notifications:**
 - Show any active alerts (e.g., "High chlorophyll levels detected").

3. Sidebar Section:

- Provide filtering options:
 - Select locations, parameters, or date ranges.
 - Toggle between real-time and historical data views.
- Include a search bar for quick navigation.

4. Footer Section:

- Add a footer with links to external resources (e.g., NOAA, local water authorities) and contact information.

Suggested Page Layout

1. Homepage

- **Hero Section:**
 - Large banner with a welcome message and call-to-action buttons ("Learn More," "Explore Data").
- **Quick Overview:**
 - Small widgets summarizing water quality trends.
 - A "Get Started" button linking to the main dashboard.

2. Dashboard

- **Key Features:**
 - Real-time data widgets with color-coded statuses.
 - Interactive map showing monitoring locations and their statuses.
 - Data visualization charts (e.g., trends over the last week/month).
 - Alert notifications for abnormal conditions.

3. Data Explorer

- **Interactive Data Table:**
 - Show raw and processed data.
 - Include filters for location, parameter, and date.
 - Export options for CSV or Excel files.

4. Settings

- **Customization Options:**
 - Set alert thresholds for each parameter.
 - Configure monitoring locations or add custom ones.
 - Choose display themes (light/dark mode).

5. About Page

- **Content:**
 - Describe the project's mission, purpose, and collaborators.
 - Acknowledgment of funding sources and data providers.

UI Components

Widgets

- **Data Cards:**

- Summarize metrics like pH, temperature, dissolved oxygen, and chlorophyll.

Example:

pH

7.8

Status: Safe

○

- **Charts:**

- **Line Charts:** Trends over time for each parameter.
- **Heatmaps:** Visualize correlations between parameters.
- **Bar Charts:** Compare averages for different locations or months.

- **Map:**

- Interactive map with clickable markers.
- Tooltip displays details like water quality, last update, and alerts.

Buttons

- Call-to-action buttons like:
 - **Refresh Data:** Pull the latest data from sensors.
 - **Export Data:** Download selected datasets.
 - **Submit Feedback:** Allow users to send feedback or report issues.

Wireframe Example

Here's a suggested structure:

Header

| FlowCast Logo | Dashboard | Data Explorer | Settings | About |

Main Content

| Metric Widgets (Cards) | Geospatial Map | Alerts & Notifications | | Charts/Trends for Selected Data |

Sidebar

| Filters (Locations, Parameters, Date Ranges) | | Settings (Thresholds, Custom Locations) |

Footer

| External Links | Contact Info |

Recommended Tools for UI Design

- **Wireframing/Prototyping:**
 - Figma, Adobe XD, or Sketch for designing mockups.
 - **Frontend Frameworks:**
 - Streamlit for fast prototyping.
 - Bootstrap or Tailwind CSS for responsive styling.
-

Appendix C - Sprint Review Reports

Sprint 1:

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 6:30PM

End time: 6:45 PM

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - User Story 1-1: Communicate with Professor Reis
 - User Story 1-0: Discuss and Plan Schematics
 - User Story 1-2: Plan Weekly Meetings
 - User Story 1-3: Discuss Programming tools with Team
 - User Story 1-4: Order Electrical Components

Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

- NA
 - Features/functions that were tested thoroughly and would benefit from some improvement/enhancements. Here is the list of suggested enhancements/improvements.
 - Create New schematics for the boat and to develop additional Raspberry Pi features

Sprint 2:

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 12:00 PM

End time: 12:20 PM

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Bracket for the case
 - Raspberry Pi

Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

- Mounting bracket for the screen
 - Parts ordered by the department have to be approved then need time to ship
- Features/functions that were tested thoroughly and would benefit from some improvement/enhancements. Here is the list of suggested enhancements/improvements.
 - Mounting the case
 - Schematics for the internals of the case
 - A port on the outside of the case for external connections

Sprint 3:

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 12:00 PM

End time: 12:20 PM

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Bracket for the case
 - Raspberry Pi
 - Wiring and case for the raspberry Pi

Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

- Mounting bracket for the screen
 - Parts ordered by the department have to be approved then need time to ship
 - Mounting the Raspberry Pi
- Features/functions that were tested thoroughly and would benefit from some improvement/enhancements. Here is the list of suggested enhancements/improvements.
 - Mounting the case
 - Schematics for the internals of the case
 - A port on the outside of the case for external connections
 - Wiring for the case as well as cooling

Sprint 4:

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 12:00 PM

End time: 12:20 PM

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Bracket for the case
 - Raspberry Pi
 - Wiring and case for the raspberry Pi

Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

- Mounting bracket for the screen
 - Parts ordered by the department have to be approved then need time to ship
 - Mounting the Raspberry Pi
- Features/functions that were tested thoroughly and would benefit from some improvement/enhancements. Here is the list of suggested enhancements/improvements.
 - Mounting the case
 - Schematics for the internals of the case
 - A port on the outside of the case for external connections

- Wiring for the case as well as cooling

Sprint 5:

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 12:00 PM

End time: 12:20 PM

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Wiring
 - Raspberry Pi
 - Field testing
 - Documentation
 - Out reach

Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

- Electric calculations of power drainage
- Features/functions that were tested thoroughly and would benefit from some improvement/enhancements. Here is the list of suggested enhancements/improvements.
 - CPU temperature testing
 - A port on the outside of the case for external connections
 - Wiring for the case as well as cooling

Sprint 6:

Project: Water Sampling with Applied Microcontrollers

Attendees: Angie Martinez, Jesus Elespuru, Steven Luque, Christopher Perez, Darniel Padron

Start time: 12:00 PM

End time: 12:20 PM

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Documentation
 - Out reach

Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

- Electric calculations of power drainage
 - GPS
- Features/functions that were tested thoroughly and would benefit from some improvement/enhancements. Here is the list of suggested enhancements/improvements.
 - CPU temperature testing

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Installation Guide

Option 1: Set up locally via Command Line

Open your terminal (or command prompt) and execute the following commands:

1. Clone the repository

git clone <https://github.com/your-username/flowcast-web-app.git>

Navigate into the Project Directory

```
cd flowcast-web-app
```

2. Set Up Your Environment

- Windows: Use python commands.
- macOS/Linux: Use python3 or python based on your setup.

3. Create a Virtual Environment (optional but highly recommended):

- `python -m venv venv`

4. Activate the Environment

- Windows: `venv\Scripts\activate`
- macOS/Linux: `source venv/bin/activate`

5. Install Required Dependencies

Install all required Python packages: `pip install -r requirements.txt`

6. Run the Streamlit App

Start the app by running the following command: `streamlit run app.py`

7. Access the App

Once the app starts, you will see this in the terminal:
You can now view your Streamlit app in your browser.

Local URL: <http://localhost:XXXX>

Open it in your browser to view the app.

Option 2: Set Up via GitHub Desktop

1. Clone the Repository Using GitHub Desktop

- Open GitHub Desktop. Then, press Click on File > Clone Repository....
- Select the repository: If the repository is listed under your GitHub account, choose it. If it's not listed, click the URL tab and enter the repository URL: <https://github.com/your-username/flowcast-web-app.git>.
- Choose a local path for the repository and click Clone.

2. Navigate to the Repository Folder

After cloning, open the folder in your terminal or command prompt: `cd`

3. Set Up Your Environment

Create a virtual environment (recommended): `python -m venv venv`

4. Activate the Environment

- Windows: `venv\Scripts\activate`
- macOS/Linux: `source venv/bin/activate`

5. Install Dependencies

Install all required Python packages: `pip install -r requirements.txt`

6. Run the Streamlit App

Start the app by running the following command: `streamlit run app.py`

7. Access the App

Once the app starts, you will see this in the terminal:
You can now view your Streamlit app in your browser.
Local URL: <http://localhost:XXXX>
Open it in your browser to view the app.

Notes for All Operating Systems

Prerequisites

1. Git: Install Git from git-scm.com (if not already installed).
 2. Python: Make sure Python 3.7 or later is installed. You can download it from python.org.
 3. Pip: Pip is included with Python; ensure it's up to date: `python -m pip install --upgrade pip`
 4. Deactivate Virtual Environment: To deactivate your virtual environment: `deactivate`
- Troubleshooting If you encounter errors during dependency installation: `pip install --upgrade pip setuptools wheel` Then retry: `pip install -r requirements.txt`
- For permission issues on macOS/Linux, use: `sudo pip install -r requirements.txt`

REFERENCES

Aguirre, L. (2021, April 2). Saving biscayne bay: Researchers worry about risk of another fish kill this summer. WPLG. <https://www.local10.com/news/local/2021/04/02/saving-biscayne-bay-researchers-worry-about-risk-of-another-fish-kill-this-summer/>

Biscayne Bay Commission. Biscayne Bay Commission | Water Quality Dashboard. (n.d.). <https://protectingfloridatogether.gov/BiscayneBayCommission>

Clearpath Robotics. (n.d.). The CLEARPATH robotics heron USV. MIT Marine Autonomy Lab : Robot - Heron browse. <https://oceanai.mit.edu/autonomylab/pmwiki/pmwiki.php?n=Robot.Heron>

Denise Maheho, D. M., Mahelo, D. M. and D., Makal, D., Kimbrough, L., Kluepfel, B., Hindrichs, B., Latam, M., Annesha, F. A., Kidangoor, A., Kamnitzer, R., & Alexander, I. (2023, February 4). Building back Miami's Biscayne Bay: Do Natural Solutions Hold Hope? Mongabay Environmental News. <https://news.mongabay.com/2021/07/building-back-miamis-biscayne-bay-do-natural-solutions-hold-hope/>

NOAA. (n.d.). National Oceanic and Atmospheric Administration. <https://www.noaa.gov/>

EXO2 multiparameter sonde. Multiparameter Water Quality Sonde for Unattended Monitoring. (n.d.). <https://www.ysi.com/exo2?srsId=AfmBOooi9qt7bUnsZvUS28ct9Nd039EOdgiIda6Dhfh1xqcceDBaxlx2>