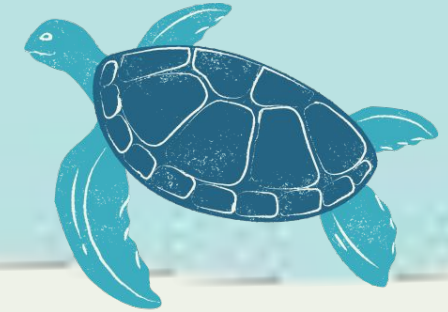
Web Application for Aqualab Sensor Monitoring and Analysis

Ruth Garcia, Haley Hamilton, Greg Thompson

Introduction



- Dr. Turingan, the Director of the Aquaculture Laboratories at Florida Tech, is analysing how much carbon dioxide is absorbed in seawater as it used in food-production by marine algae (seaweeds).
- There are 2 tanks, one below which contains an environment of water and marine algae, and one above which contains a controlled environment of carbon dioxide.
- Multiple sensors/apparatuses are utilized to measure data including a water quality sensor, an air quality sensor, and a pressure gauge.

Overall goal and motivation:

Goal: Develop a customized web application to improve research efficiency and minimize time wasted from errors. The application will have the capability to:

- Connect with and receive data from the sensors.
- Display all current sensor measurements
- Alert the team when measurements are out of the desired range
- Record all sensor data, automatically plot the data, and allow the user to filter through the data

Motivation: Current lab sensors are not connected to any system — Data and measurements from sensors only available in the lab and cannot be monitored remotely or automatically recorded.

Different User Types:

Lab Team Leader

In charge of the lab team and is the main researcher, has overall authority in all researching decisions.

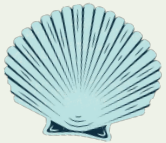
Lab Team Assistants

Works for/under the lab team leader, supports the research effort and reports back to the lab team leader.

Lab Mech Eng

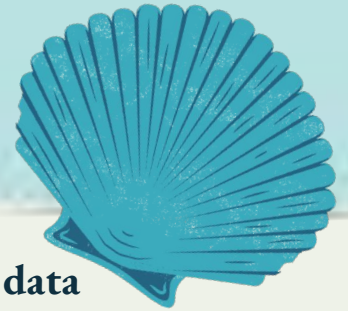
Works for/under the lab team leader, supports the lab equipment and sensors, ensures the research environment is properly set up.

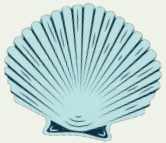
Approaches (key system features):



- **Allows all users to connect sensors to the web application.**
    a. Sensors include water quality sensor measuring amount of CO_2 in seawater, air quality sensor measuring amount of CO_2 in air, and pressure sensor measuring the pressure of the environment.
- **Allows users to monitor current/recent data measurements from the sensors**
    b. All users can view the current data measurements given by the sensors
    c. All users can receive an alert if measurements aren't in specified values/range
    d. The Lab Team Leader can set expected values/ranges for sensor measurements

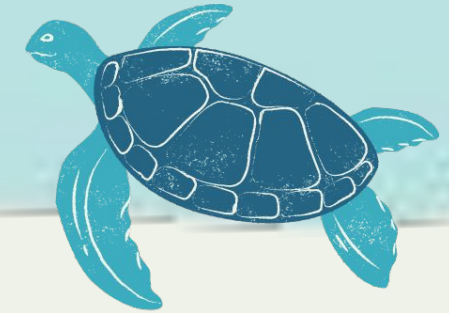
Approaches (key system features):





- **Allows users (Lab Team Leader and Assistants) to view and analyze recorded data**
  a. Users can view recorded data in plotted graphs
  b. Users can apply filters to data to view desired data
  c. Users can view calculated relationships between sensor data
  d. Users can export data into a CSV file and retrieve archived data files from the cloud
- **Allows users (the Lab Team Leader and Assistants) to easily manage disk storage**
  e. Users can view current used local disk storage from the web application
  f. Users can receive alerts when local disk storage is getting full
  g. Users can move recorded data to chosen secondary storage and/or delete chosen data.

Algorithms and Tools:



- Coding Languages: Python and HTML/CSS/JavaScript
- Sensor Connectivity: Sensor APIs/libraries (Ex: RS232 and pyserial library)
- Code Collaboration: Github
- IDE: Visual Studio Code/IntelliJ
- Alerting System: JavaScript for on screen alerts/OneSignal for push notifications
- Data Plotting: JavaScript Plotting Library (Plotly.js/Chart.js/D3.js)
- Hosting Service: AWS or Google Cloud Platform

Technical Challenges:

- Connecting to different sensors via different APIs/connections and libraries
- Collecting data and displaying it accurately in real time
- Hosting a server for 24/7 access that is accessible anywhere
- Displaying/plotting data over time in an easy to read graph
- Archiving recorded data and uploading it to a cloud
- Allowing users the ability to move/delete recorded data

Milestone 1 (Sep 30):

Compare and select technical tools for:

-communicating with sensors, displaying the data, data analysis tools, user interface, recording data, accessing recorded data

Provide demos to evaluate tools for:

-communicating with sensors, displaying the data, data analysis tools, user interface, recording data, accessing recorded data

Resolve technical challenges:

-Connecting to different sensors via APIs/connections and libraries

-Collecting data and displaying it accurately in real time

-Hosting a server for 24/7 access that is accessible anywhere

-Displaying/plotting data over time in an easy to read graph

Compare and select collaboration tools for software development, documents/presentations, communication, task calendar

Create Requirement Document

Create Design Document

Create Test Plan

Milestone 2 (Oct 28):

- Implement, test, and demo Communicating with Sensors
- Implement, test, and demo User Interface
- Implement, test, and demo Recording Data and Uploading to Cloud

Milestone 3 (Nov 25):

- Implement, test, and demo Displaying the data
- Implement, test, and demo Data Analysis Tools
- Implement, test, and demo Accessing Recorded Data

Task matrix for Milestone 1:

Task	Greg	Haley	Ruth
Compare and select Technical Tools	data analysis tools, accessing recorded data	communicating with sensors, recording data	user interface, displaying the data
"hello world" demos	data analysis, accessing recorded data	communicating with sensors, recording data	user interface, displaying the data
Resolve Technical Challenges	Archiving recorded data + uploading to cloud, user ability to move/delete recorded data	APIs/tools/libraries for connecting to different sensors, collecting data and displaying it in real time	Hosting services for remote access, tools/libraries for plotting recorded data
Compare and select Collaboration Tools	programs	documents/presentations	communication, task calendar
Requirement Document	write 20%	write 60%	write 20%
Design Document	write 60%	write 20%	write 20%
Test Plan	write 30%	write 30%	write 40%