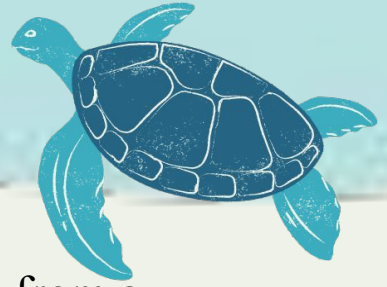
Web Application for Aqualab Sensor Monitoring and Analysis - Milestone 2

Ruth Garcia, Haley Hamilton, Greg Thompson

Milestone 2 Overview:



Implement, test, and demo *Communicating with Sensors:*

Sensor classes were created with functions to connect and disconnect from a serial port and read data from the port.

Implement, test, and demo *User Interface*

Mockups were developed into a User Interface.

Implement, test, and demo *Recording Data*

Data is being read from a set of ‘sensor’ classes and stored in a database.

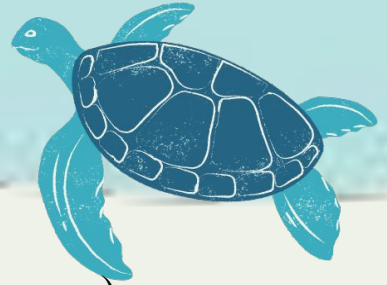
Implement, test, and demo *Uploading to Cloud*

Remote access is functional. Cloud storage and computing in the works.

Milestone 2 Progress Matrix:

Task	Completion	Greg	Haley	Ruth	To do
Implement, test, and demo *Communicating with Sensors*	70%	10%	80%	10%	Test with real sensors once arrived, implement check data range and send alerts functionality.
Implement, test, and demo *User Interface*	80%	0%	5%	95%	Complete user authentication functionality.
Implement, test, and demo *Recording Data*	100%	60%	40%	0%	None.
Implement, test, and demo *Uploading to Cloud*	30%	30%	70%	0%	Need to add further functionality, test, and demo.

Communicating with Sensors:



Sensor Classes:

3 sensor classes were created for each sensor type (water, air, and pressure).

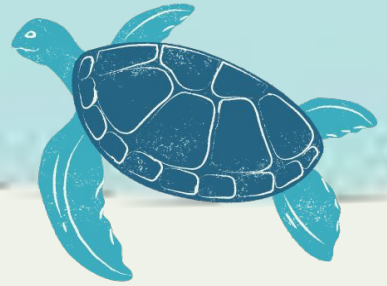
Class Functions:

Each sensor class has 3 functions, connect to serial port, disconnect from serial port, and read data.

The system connects to the serial port of the sensor at the start of the run with the user specified COM port. At the end of the run or to reconfigure the system, the system disconnects from the serial port.

At this moment, the read data function returns placeholder data. Once we get the sensors and see how the data is read, it will be parsed and returned.

Recording Data:



Sensor Monitoring:

System implements multiple threads to simultaneously monitor each sensor.

Data Synchronisation:

On a clock, a central thread records the most recent value recorded by each sensor.

Data Checking:

The central thread ensures data is in acceptable ranges and notifies users if that is not true.

Data Storage:

The Central thread then sends the data to our database.

Recording Data (Demo):

```
current readings in main [{'water-CO2': 102, 'numbers': 3}, {'air-CO2': 201, 'numbers': 7}, {'pressure': 77, 'numbers': 9}]
current readings of sensor  <w_sensor.Water_Sensor object at 0x000001E750FAA4B0> :  {'water-CO2': 102, 'numbers': 3}
current readings of sensor  <p_sensor.Pressure_Sensor object at 0x000001E752A9D9D0> :  {'pressure': 77, 'numbers': 9}
current readings of sensor  <a_sensor.Air_Sensor object at 0x000001E752A9EF90> :  {'air-CO2': 201, 'numbers': 7}
current readings of sensor  <w_sensor.Water_Sensor object at 0x000001E750FAA4B0> :  {'water-CO2': 102, 'numbers': 3}
current readings of sensor  <p_sensor.Pressure_Sensor object at 0x000001E752A9D9D0> :  {'pressure': 77, 'numbers': 9}
current readings of sensor  <a_sensor.Air_Sensor object at 0x000001E752A9EF90> :  {'air-CO2': 201, 'numbers': 7}
end of loop
current readings in main [{'water-CO2': 102, 'numbers': 3}, {'air-CO2': 201, 'numbers': 7}, {'pressure': 77, 'numbers': 9}]
current readings of sensor  <w_sensor.Water_Sensor object at 0x000001E750FAA4B0> :  {'water-CO2': 102, 'numbers': 3}
current readings of sensor  <p_sensor.Pressure_Sensor object at 0x000001E752A9D9D0> :  {'pressure': 77, 'numbers': 9}
current readings of sensor  <a_sensor.Air_Sensor object at 0x000001E752A9EF90> :  {'air-CO2': 201, 'numbers': 7}
current readings of sensor  <w_sensor.Water_Sensor object at 0x000001E750FAA4B0> :  {'water-CO2': 102, 'numbers': 3}
current readings of sensor  <p_sensor.Pressure_Sensor object at 0x000001E752A9D9D0> :  {'pressure': 77, 'numbers': 9}
current readings of sensor  <a_sensor.Air_Sensor object at 0x000001E752A9EF90> :  {'air-CO2': 201, 'numbers': 7}
Program Terminated
```

Recording Data (Demo):

a_sensor_collection

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	89	48.00 B	1	36.86 kB

p_sensor_collection

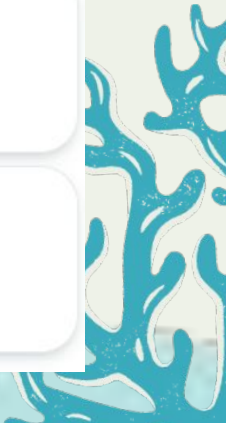
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	89	49.00 B	1	36.86 kB

user_collection

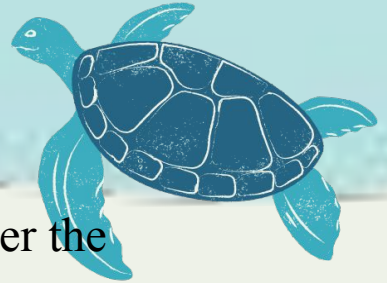
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	2	100.00 B	1	36.86 kB

w_sensor_collection

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	89	50.00 B	1	36.86 kB



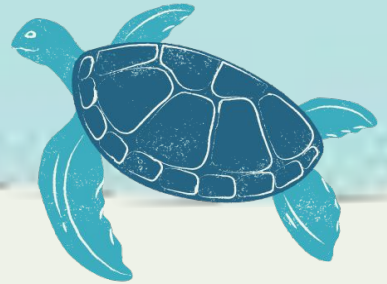
Uploading to Cloud:



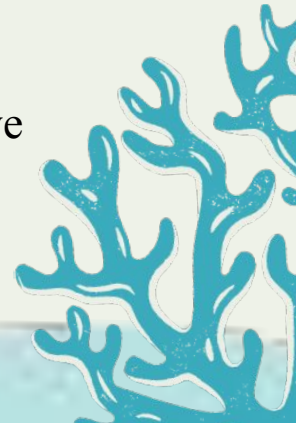
The remote access to the application is functional. A client can connect over the network to the host application to view the webpage.

We have done research into cloud computing options. Considering expected data loads, we do not currently plan on utilizing a third party for cloud storage. Once we receive the final sensors, we will do a comprehensive test on the rate of data creation to create test cases. If it becomes necessary, we will likely utilize AWS for cloud storage and processing resources.

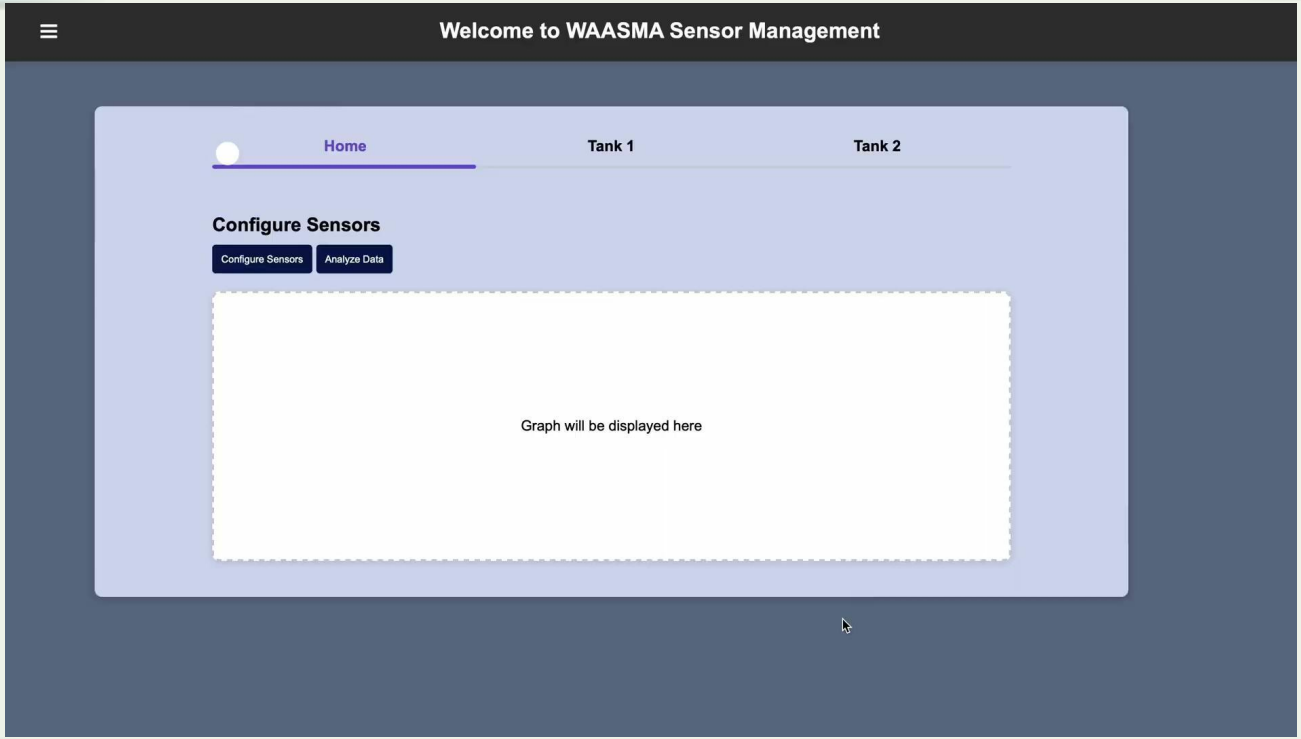
User Interface:



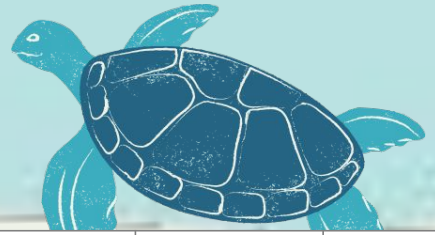
- We created a UI similar to our MockUp
- Created three tabs
  - Created a data analyzation tool and a sensors configuration tool
  - Graphs will be applied later on, as well as functionality of buttons
- Created a settings page
  - Control sensor ranges and the frequency of data updates
- Created a user page
  - Admin can control who is being added
  - Full working functionality will be coming soon!
    - Waiting on approval of client for the UI and making changes before we get to the fun stuff



Website Demo:

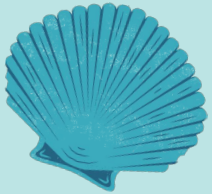


Milestone 3:



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

Task	**Greg**	**Haley**	**Ruth**
Implement, test, and demo *Displaying the data*	5%	5%	90%
Implement, test, and demo *Data Analysis Tools*	90%	5%	5%
Implement, test, and demo *Accessing Recorded Data*	5%	90%	5%
Implement, test, demo *Login Page*	5%	5%	90%



Questions?