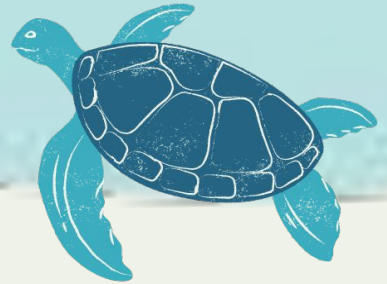
Web Application for Aqualab Sensor Monitoring and Analysis - Milestone 3

Ruth Garcia, Haley Hamilton, Greg Thompson

Milestone 3 Overview:



Implement, test, and demo *Accessing Recorded Data*

- Ensure parts of the system have access to the data stored in the database
- CRUD functions

Implement, test, and demo *Displaying the Data*

- Ensure frontend can read and display data from backend
- React Modules

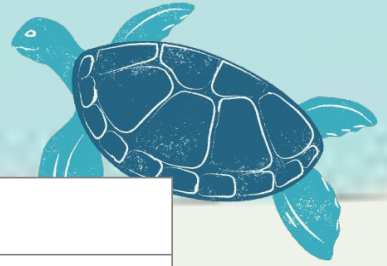
Implement, test, and demo *Data Analysis Tools*

- Tool allows users to filter through and display desired data in a graph

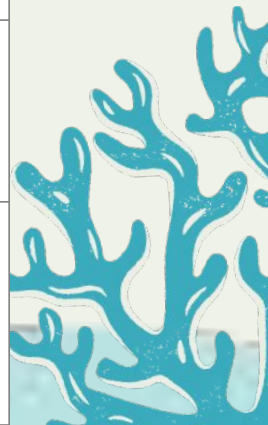
Implement, test, and demo *Data Checking and Notifications*

- System checks data to ensure its in the desired range
- System sends notifications if out of range

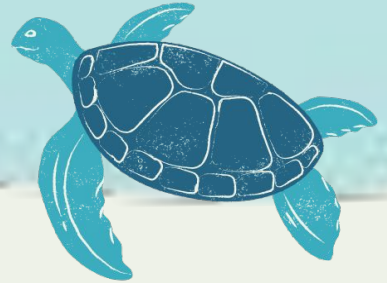
Milestone 3 Progress Matrix:



Task	Completion	Greg	Haley	Ruth	To do
Implement, test, and demo *Displaying the data*	70%	0%	40%	60%	Test functionality with sensors and refine data handling.
Implement, test, and demo *Data Analysis Tools*	70%	30%	40%	30%	Test functionality with sensors and refine data handling.
Implement, test, and demo *Accessing Recorded Data*	50%	20%	60%	20%	Develop the move/delete data functionality.
Implement, test, and demo *Data Checking and Notifications*	70%	100%	0%	0%	Test functionality with sensors and additional development and testing with notification system..



New Backend Multithreading:



So program synchronisation is complicated.

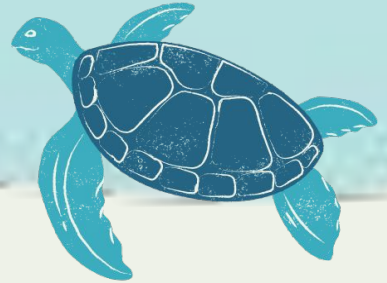
The technical details here are beyond the scope of this presentation, but the tasks performed by threads and the methods of sharing data between threads have been reworked. This will mitigate potential thread starvations and eliminate data corruption issues.

User Notifications:

Sensor APIs now check whether received data is in expected ranges.
The functionality for an authorized user to modify these ranges has been developed and implemented.

System currently prints to console to avoid spamming. Once real datasets are available for testing, we will discuss with the client the exact specifications of notification messages. (what information must be sent and how often messages should be sent)

Accessing Recorded Data



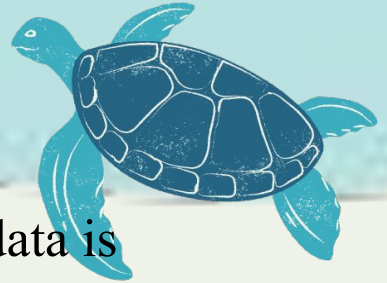
- **Frontend to Backend Connection -**
  - CRUD functions were made in the Flask app
  - allows different react modules to read data from the backend

@self.app.route("/current_sensor_data/", methods=["GET"])
- allows graphs on home page to read and display data

@self.app.route("/config_sensors", methods=["PATCH"])
- allows home page to update the sensor configuration

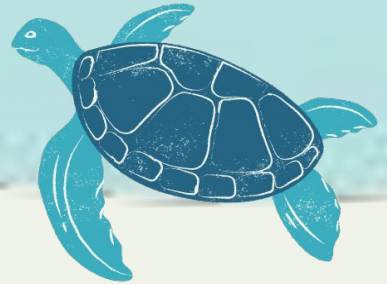
@self.app.route("/create_user", methods=["POST"])
- allows user page to create new users

Data Analysis Tools:



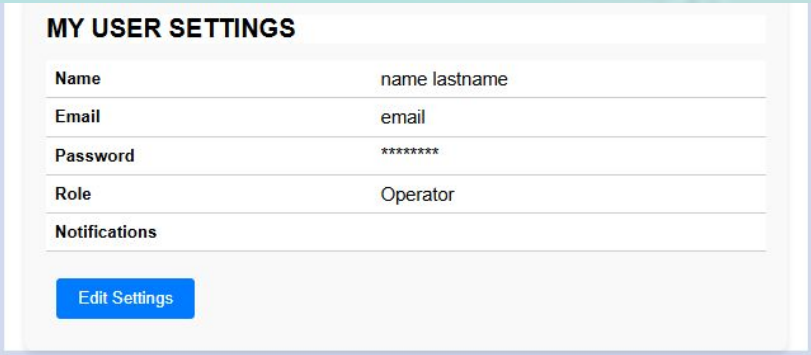
- **The Goal:** Users input data filters and a graph of the desired data is displayed.
- **Current progress:**
  - User can select filters and apply them, these filters are sent to the backend and query the data
  - We don't have properly simulated data, so right now this function is not sending back data to display
  - Right now, a real time react chart is displayed instead, showing we know how to use data to display a chart (same for the home page)
  - The tank tabs DO communicate with the backend to receive updating data. It gets a number every couple of seconds and adds it as a point on the graph

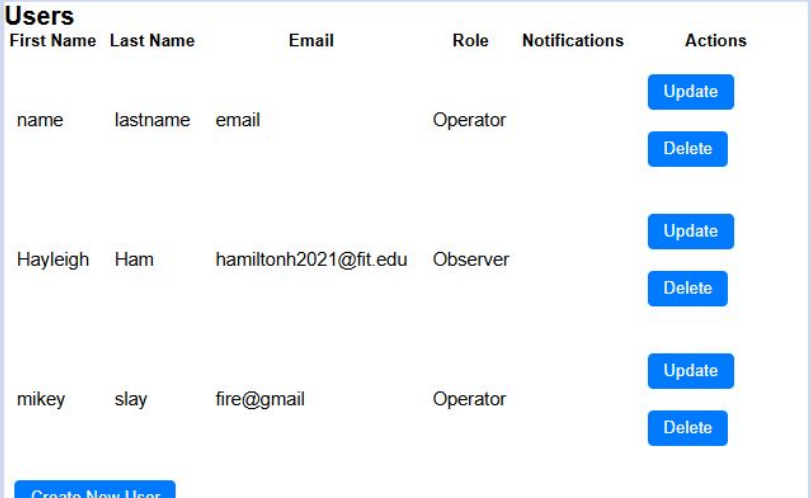
Displaying the Data and User Interface:



- **Consists of <u>Flask functions</u> and <u>React modules</u>**
  - Allows the frontend and backend to communicate and React properly displays all needed data and forms
- **UI styling is a work in progress**
  - Integration of our existing UI styling and our evolving React modules was difficult, trying to say true to mockups.
- **Full Implementation Awaiting Sensors:**
  - Sensors are ordered and will be here for M4 (fingers crossed)
  - Right now we are displaying dummy data, code and measurements will be tailed once we have proper example data

Frontend Backend Connection (Demo) - User:



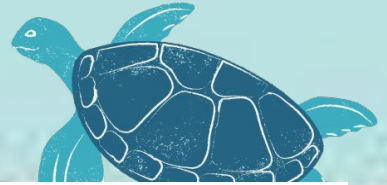



```
_id: ObjectId('67339694bc5ff38d4a7eacb7')
firstName : "name"
lastName : "lastname"
email : "email"
role : "Operator"
password : "password"
```

```
_id: ObjectId('6743383f48ce505f5fb7be0a')
firstName : "Hayleigh"
lastName : "Ham"
email : "hamiltonh2021@fit.edu"
role : "Observer"
password : "password"
```

```
_id: ObjectId('6743e9b3f769525b2bb32baf')
firstName : "mikey"
lastName : "slay"
email : "fire@gmail"
role : "Operator"
password : "password"
```

Frontend Backend Connection (Demo) - Settings:



SENSOR SETTINGS

Sensor Type	Tank #	Communication	Range	
Water	1	COM3	1-222	Change Range
Air	1	COM4	7-9	Change Range
Pressure	1	COM5	8-98645	Change Range
Water	2	COM6	5-2	Change Range
Air	2	COM7	7-3	Change Range
Pressure	2	COM8	9-3	Change Range

DATA SETTINGS

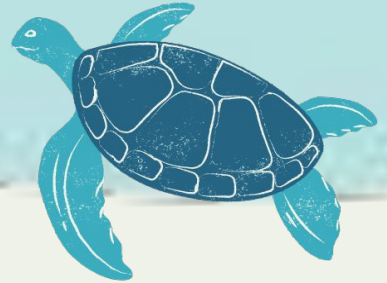
Frequency of Data Reading:	**7**	Update
Frequency of Data Storing:	**8**	Update


```
_id: ObjectId('674354a15e5844b0220b807a')
type : "Water"
tank : "1"
coms : "COM3"
range_high : "222"
range_low : 1
```

```
_id: ObjectId('674354b67c7b0f93bf4c1d09')
type : "Air"
tank : "1"
coms : "COM4"
range_high : "9"
range_low : "7"
```

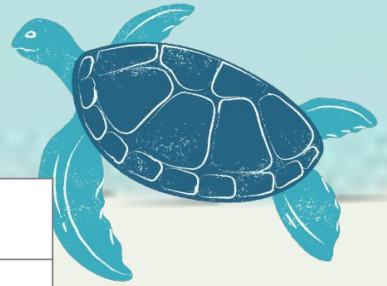
```
_id: ObjectId('674354cb50844dc982c63a8b')
type : "Pressure"
tank : "1"
coms : "COM5"
range_high : "98645"
range_low : "8"
```

Website Functionality Demo:

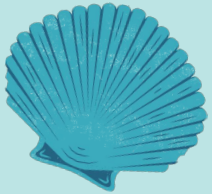


One Moment Please
(grabbing flash drive)

Milestone 4 Progress Matrix:



Task	Greg	Haley	Ruth
Testing and Refactoring System with Sensors and Data	40%	40%	20%
Implement, test, and demo *Move/Delete Data from Database*	20%	40%	40%
Implement, test, and demo *Data Backup in Cloud Storage*	40%	20%	40%
Implement, test, and demo *User Role Logic*	30%	30%	40%



Questions?