Web Application for Aqualab Sensor Monitoring and Analysis

System Requirements Document

Prepared by: Gregory Thompson - gthompson2022@my.fit.edu
Haley Hamilton - hamiltonh2021@my.fit.edu
Ruth Garcia - ruth2021@my.fit.edu

Project Advisor: Dr. Slhoub - kslhoub@fit.edu
Project Client: Dr.Turingan

Version: 0.1
Date Created: 09/02/2024

Table of contents

1. Introduction

1.1 Purpose

The purpose of this project is to develop a system that reads and displays data for use in Dr. Turingan's lab research. The product allows him and his lab team to connect their laboratory sensors/apparatuses to a system and view the data remotely and in real time as well as give alerts when measurements are outside of desired ranges. This document aims to clearly and concretely describe the requirements of this project.

1.2 Document Conventions

In this document, the term *system* always refers to the product being developed and the term *client* refers to Dr. Turingan and his lab team, who will be receiving the product and using it for their research. The functional requirements are organized into features based on similar functionalities. Each feature contains a list of numbered and specific requirements with the format REQ-# (e.g. REQ-1, REQ-2). Each feature is also assigned a priority level (e.g., High, Medium, Low) and are ranked on their benefit to the product, penalty if the feature is not implemented correctly or at all, cost to implement, and risk level involved with implementing the requirement. Ratings are on a scale of 1-10, 1 being the lowest (e.g., low benefit, low cost, etc.) and 10 being the highest (e.g., high penalty, high risk, etc.)

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by the client, developers, and the project advisors. This document contains specifications about the product and its requirements including the different external interfaces, functional, and non-functional requirements.

1.4 Product Scope

The scope of this product is to develop a web application that will connect to and read in data from sensors and display it to allow the client to view the data and analyze it. This capability gives the client a centralized system that automatically reads and records data from different sensors, alerts them when sensor measurements are not in the expected range, and gives them a versatile tool to compare and analyze the data. By implementing this system, the client will streamline their research, making the process more efficient and reducing the likelihood of errors.

1.5 References

This document follows the IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998.

2. Overall Description

2.1 Product Perspective

Dr. Turingan, the Director of the Aquaculture Laboratories at Florida Tech, is analyzing how much carbon dioxide is absorbed in seawater as it is 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. The data is measured via a water quality sensor that collects data on how much CO2 is in the water, an air quality sensor to collect data on the levels of CO2 in the air, and a pressure gauge to measure the pressure of the top controlled environment tank.

The lab team described a need for having 24/7 access to constantly updating information and measurements and for detecting and alerting when measurements do and do not fall within a desired range.

2.2 Product Functions

The product will be an intuitive, easy to use web application customized for the lab team to improve their research efficiency and minimize time wasted from errors. The application will have the capability to connect with and receive data from the sensors. The application will display current sensor measurements to be accessed remotely and give alerts when measurements are out of the desired range to ensure the lab team can respond to errors quickly. The application will record all sensor data, automatically plot the data, and allow the user to filter through the data to simplify data analysis. The application will simplify disk storage management by informing users of the currently used local disk storage and allowing them to move or delete data.

2.3 User Classes and Characteristics

There are three user classes: Admin, Operator, and Observer.

The Admin role is reserved for the lab team leader, Dr. Turingan. The admin is in charge of the lab team and is the main researcher. This role has overall authority in all researching decisions, is the most important, and has the most access and privilege. A user with this role uses the product frequently.

The Operator role is reserved for the lab team assistants. The operator works for/under the lab team leader, supports the research effort, and reports back to the lab team leader. This role is of average importance and has average access and privilege. A user with this role uses the product most frequently.

The Observer role is reserved for the lab mechanical engineers. The observer also works for/under the lab team leader and supports the lab equipment and sensors. This role ensures the research environment is properly set up and has the least access and privilege. A user with this role uses the product somewhat frequently.

2.4 Operating Environment

The system shall operate on a Windows 10 or 11 laptop. The sensors will be connected to the laptop via USB/ethernet cables or may be connected to an Arduino that is connected to the laptop. TBD.

2.5 Design and Implementation Constraints

2.5.1 Budget: We have to ensure that all sensors and hardware purchased remains in the budget of the client while also ensuring that the system's technical and quality needs are met.

2.5.2 Knowledge: We have to conduct additional research on web application development, database creation and management, and methods to connect to sensors.

2.5.3 Maintenance: The client must maintain software after delivering, including resolving any issues and ensuring the system continues to function as intended.

2.6 User Documentation

The product will consist of documentation of the general software structure as well as list each class and their role. The product will also consist of User Guide documentation. This will walk the user through all of their possible actions with all of the different system functions.

2.7 Assumptions and Dependencies

2.7.1 We assume the client has a sufficient budget for the purchasing of needed sensors that will work both for the specifications and the compatibility needed as well as any additional hardware to connect the sensors and run the system.

2.7.2 We assume there will be little to no issues with the sensors being delivered in a reasonable time and they will be fully functional once they arrive.

3. External Interface Requirements

3.1 User Interfaces

The user interacts with the software via a web application consisting of different screens and pages to organize the different sensor displays and different user actions. The user can click through the different screens, click buttons, and submit input information using the user interface.

3.2 Hardware Interfaces

The system interfaces with the three different sensors (water quality, air quality, and pressure) using different wired connections. This includes RS232-USB cables as well as an Arduino and Arduino Interface Shield.

3.3 Software Interfaces

The system interfaces with various APIs and libraries to read in data from the different sensors (e.g. pyserial). The system stores and processes this data by interfacing with a database, MongoDB. The backend of the system is implemented with a Python web framework tool, Flask, and the frontend display of the data is implemented with React and HTML/CSS/JS.

3.4 Communications Interfaces

The system will utilize HTTP/HTTPS protocol for secure web application communication. The system will also communicate with users via phone and email push notification so they are alerted when sensor measurements are out of the desired range.

4. System Features

The following is a list and description of each of the core functional requirements of the product.

4.1 Sensor Connection

4.1.1 Description and Priority

The system must be able to connect with each type of sensor. This includes a water quality probe (Manta+40 from Eureka Water Probes) that measures CO2 in the water, an air quality sensor (CO2 Gas Sensor from Vernier) that measures CO2 in the air, and a pressure sensor (Pressure Sensor 400 from Vernier) that measures the pressure of the environment. As each sensor is different and has different modes of connection, the system must have specific code implemented to connect with and read data from each type of sensor. Priority: High.

- Benefit: 9
- Penalty: 8
- Cost: 8
- Risk: 3

4.1.2 Stimulus/Response Sequences

1. Admin user configures the sensors by selecting how many they are working with, the types of each sensor, and the connection information (e.g. COM port). System uses inputs to configure and read data from the sensors.

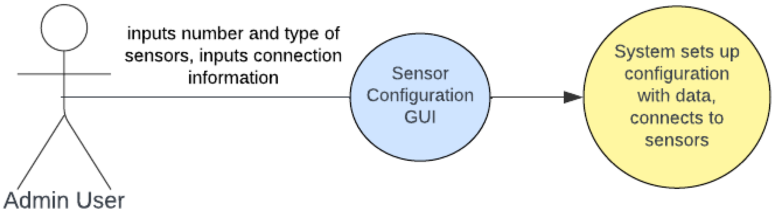
4.1.3 Functional Requirements

- REQ-1: The system shall utilize the necessary physical hardware as well as libraries or API’s to connect with and read from the water quality sensor, model Manta+40 from Eureka Water Probes.
- REQ-2: The system shall utilize the necessary physical hardware as well as libraries or API’s to connect with and read from the air quality sensor, “CO2 Gas Sensor” from Vernier.

- REQ-3: The system shall utilize the necessary physical hardware as well as libraries or API's to connect with and read from the pressure sensor, "Pressure Sensor 400" from Vernier.
- REQ-4: The system shall allow Admin users to input connection information about the sensors so the system can connect to them.
- REQ-5: The system shall allow Admin users to configure the number of sensors and the type of sensors they are working with.

4.1.4 Use Case Diagrams

- REQ-1 - REQ-5: Sensor Connection and Admin User Inputs


inputs number and type of sensors, inputs connection information
Sensor Configuration GUI
System sets up configuration with data, connects to sensors
Admin User

4.2 Monitoring Current/Recent Sensor Data

4.2.1 Description and Priority

The system must display current/recent measurements from the sensor remotely and in real time via a web application. The system must allow only Admin users the functionality to set desired values ranges for the expected measurements from each sensor. If the measurements are not in the desired ranges/values, the system will notify all users via an on screen warning message or/and a push notification. Priority: Medium.

- Benefit: 9
- Penalty: 8
- Cost: 3
- Risk: 1

4.2.2 Stimulus/Response Sequences

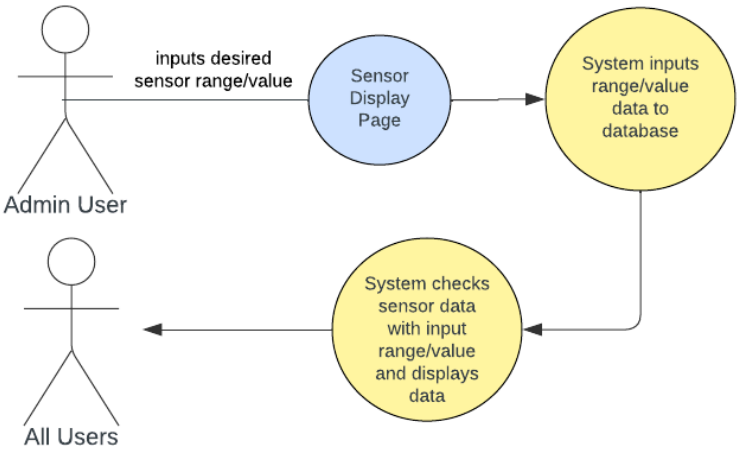
1. User navigates to the home page and selects a tab to view sensor data for (all tanks or specific tanks).
   System displays the current/recent measurements of the water quality, air quality, and pressure sensor for all tanks or a specific tank.
2. User navigates to the settings display and inputs desired ranges/values for specific sensors (settings page or in specific tank sensor area).
   System uses input to check all incoming data from that sensor and ensures it matches the range/value. If it does not, the system alerts the user via an on-screen alert and a push notification.

4.2.3 Functional Requirements

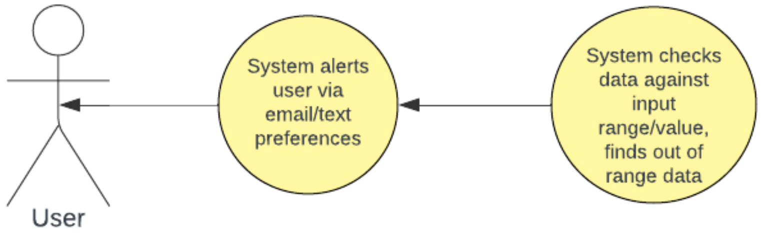
- REQ-6: The system shall display for the user the current and recent measurements read from the water quality, air quality, and pressure sensors.
- REQ-7: The system shall allow Admin users to enter desired ranges/values for each sensor.
- REQ-8: The system shall alert users if the sensor data does not fall within the specified range/value via an on screen alert and a push notification.

4.2.4 Use Case Diagrams

- REQ-6 and REQ-7: Data Displaying and Range/Value Inputs


Admin User
inputs desired sensor range/value
Sensor Display Page
System inputs range/value data to database
System checks sensor data with input range/value and displays data
All Users

- REQ-8: Sensor Alerts


User
System alerts user via email/text preferences
System checks data against input range/value, finds out of range data

4.3 Analysis of Past Measurements

4.3.1 Description and Priority

The system must record past measurements and plot them over time in graphs to allow Admins and Operators the functionality to view trends in sensor data. The system must allow Admins and Operators the functionality to filter through past measurements by type of data and by date to view and plot only the necessary sensor data from a given period of time. The system must use recorded data to calculate relationships between sensor data that the Admin requests (Ex: the relationship between the amount of CO2 in the air and in the water). The system must allow Admins and Operators the functionality to export desired collected measurements (filtered or unfiltered) into a CSV file for use with analysis tools like Excel. Priority: Medium.

- Benefit: 6
- Penalty: 3
- Cost: 4
- Risk: 3

4.3.2 Stimulus/Response Sequences

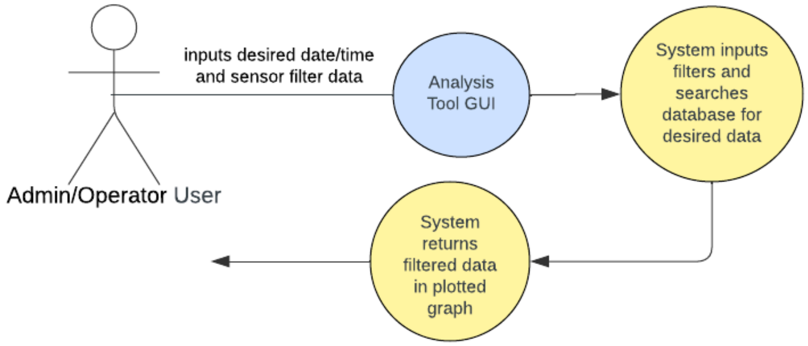
1. User navigates to the analysis tool.
   System displays the tool and all past recorded data plotted on a graph.
2. User specifies which sensor data they would like to view and from what specific time period.
   System displays this filtered data plotted on a graph.
3. The user exports the data to use with Excel.
   System downloads the data to users computer

4.3.3 Functional Requirements

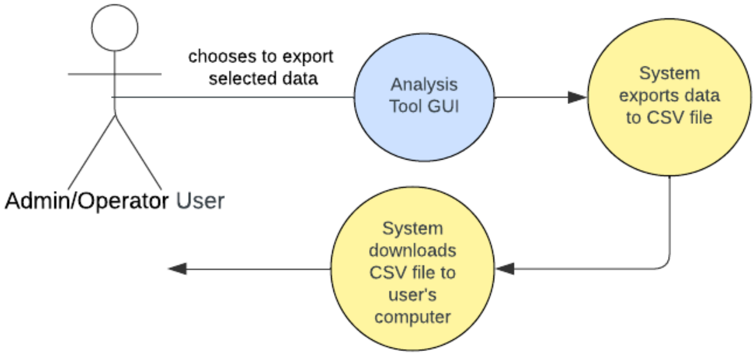
- REQ-9: The system shall record past measurements for the water quality, air quality, and pressure sensors to a database for use with analysis tools and general collection of past data.
- REQ-10: The system shall plot all recorded data overtime in a graph.
- REQ-11: The system shall receive user input to filter through recorded measurements by type of sensor data and by date.
- REQ-12: The system shall use recorded data to calculate and display relationships between sensor data as requested by the client.
- REQ-13: The system shall allow users to export collected measurements (filtered or unfiltered) into a CSV file that can be downloaded to their computer.
- REQ-14: The system shall allow the Admin user to change the frequency of when data is recorded to the database.

4.3.4 Use Case Diagrams

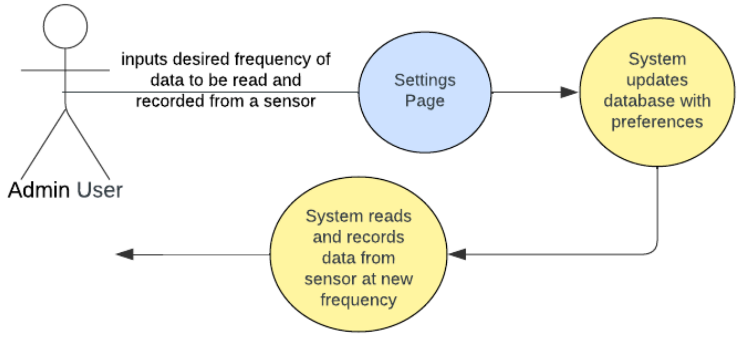
- REQ-11: Data Filtering and Plotting


inputs desired date/time and sensor filter data
Analysis Tool GUI
System inputs filters and searches database for desired data
Admin/Operator User
System returns filtered data in plotted graph

- REQ-13: Data Exporting


chooses to export selected data
Analysis Tool GUI
System exports data to CSV file
Admin/Operator User
System downloads CSV file to user's computer

- REQ-14: Data Frequency Preferences


inputs desired frequency of data to be read and recorded from a sensor
Settings Page
System updates database with preferences
Admin User
System reads and records data from sensor at new frequency

4.4 Mitigate Disk Overflow Risk

4.4.1 Description and Priority

The system must display how much local disk storage is currently being taken up. The system must alert the users on screen when local disk storage is getting full (ex: ~70% full). As a default, the system will archive recorded measurements to CSV files and regularly uploaded to a cloud every month which Admins and Operators can access and use with analysis tools. The system must allow Admins and Operators the functionality to move recorded data to chosen secondary storage and/or delete chosen data. Priority: Low.

- Benefit: 5
- Penalty: 8
- Cost: 4
- Risk: 3

4.4.2 Stimulus/Response Sequences

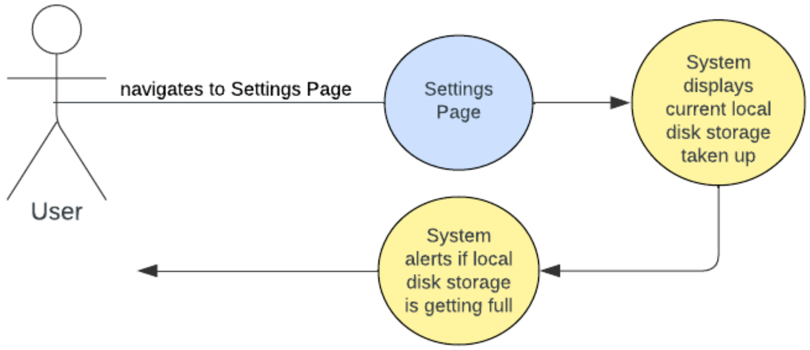
1. User navigates to view how full the local disk storage is.
   System displays the current local disk storage and alerts the user when it is getting full.
2. User changes the settings of how often data is backed up to the cloud.
   System updates this information.
3. Admin user chooses to move/delete some sensor data and specifies which sensor data from what time.
   System deleted the chosen data from the database and (if User chose to move the data) downloads the chosen data onto the user's computer.

4.4.3 Functional Requirements

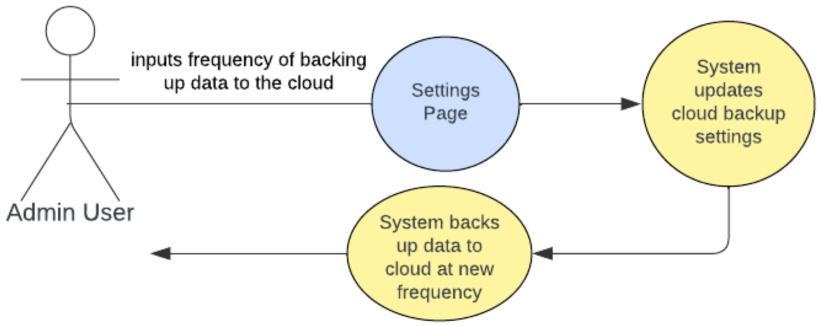
- REQ-15: The system shall display how much local disk storage is currently being taken up.
- REQ-16: The system shall alert the user when local disk storage is getting full (ex: ~70% full).
- REQ-17: The system shall have a default backup method where recorded measurements are archived to CSV files and regularly uploaded to a cloud.
- REQ-18: The system shall allow Admin users to change the data cloud backup settings.
- REQ-19: The system will allow Admin users to move recorded data to chosen secondary storage and/or delete chosen data.

4.4.4 Use Case Diagrams

- REQ-15 and REQ-16: Local Disk Storage Display and Alerts


User
navigates to Settings Page
Settings Page
System displays current local disk storage taken up
System alerts if local disk storage is getting full

- REQ-17: Data Backup to Cloud


inputs frequency of backing up data to the cloud
Admin User
Settings Page
System updates cloud backup settings
System backs up data to cloud at new frequency

4.5 User Authentication and Security

4.5.1 Description and Priority

The system must have a user authentication system that allows users to log into the system using their email and a specified password. To create new users, the system must have a feature for Admins to input a user's email, password, and role. The system must have three different role types with different user privileges and access. The system must allow users to specify if they would like to receive email or text notifications if a sensor measurement is out of range and the system must be able to take in a user's phone number. The system must log when a user has logged into or out of the application.
Priority: Medium.

- Benefit: 5
- Penalty: 3
- Cost: 3
- Risk: 5

4.5.2 Stimulus/Response Sequences

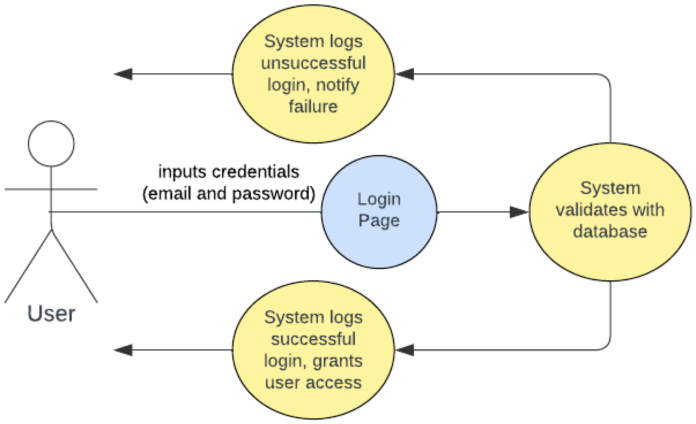
1. Admin user chooses to create a new user and inputs the user's email, role, and a password for the users and submits the information.
   System updates the database with the new user information.
2. User enters their email and password to the login screen and selects submit.
   System checks the user email and password against the database information and allows the user access if the email is listed and the password is correct, denies user access otherwise.
3. User is logged into the system and it has a particular role.
   System restricts and grants access to certain pages and tools depending on the user role.
4. User selects whether they would like email and/or text notifications and submits their phone number.
   System updates the database with the preferences and data. If sensor measurements are out of range, the system retrieves data from the database to know which users to send alerts to and to which email/phone number.
5. User logs in successfully or unsuccessfully to the system. User logs out of the system.
   System records the user's email, the user's action, and the date and time to a log file.

4.5.3 Functional Requirements

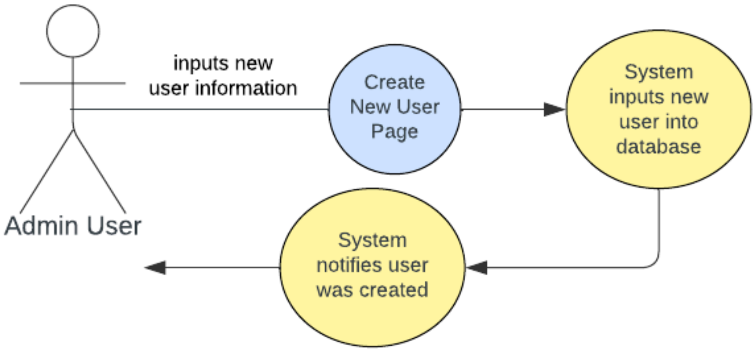
- REQ-20: The system shall allow users to log into the system using their registered email and a password. Upon successful login, the system shall grant the user access to the application.
- REQ-21: The system shall allow Admin users to create a new user by inputting the new user's email, password, and role.
- REQ-22: The system shall have three different role types, Admin, Operator, and Observer, each with different levels of user privileges and access. The Admin user can access all the system features, the Operator can access all the system features except for Admin specific features, and the Observer can only access the data display page and the user settings.
- REQ-23: The system shall allow users to specify if they would like to receive email or text notifications if a sensor measurement is out of range. If text notification is selected, the system must be able to take in the user's phone number.
- REQ-24: The system shall log when a user has logged into the application, both successfully or unsuccessfully, and logged out of the application.

4.5.4 Use Case Diagrams

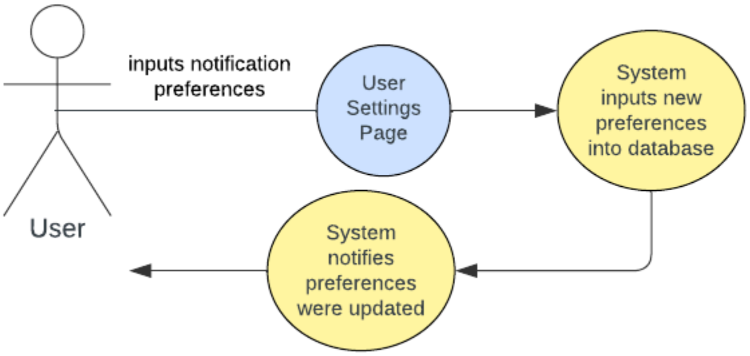
- REQ-20 and REQ-24: User Login and Activity Logging


System logs unsuccessful login, notify failure
inputs credentials (email and password)
Login Page
System validates with database
User
System logs successful login, grants user access

- REQ-21: User Creation


inputs new user information
Create New User Page
System inputs new user into database
Admin User
System notifies user was created

- REQ-23: User Preferences


inputs notification preferences
User
User Settings Page
System inputs new preferences into database
System notifies preferences were updated

5. Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 The system shall display data from the sensors within 2 seconds of reading the data for 90% of all data read during nominal operating conditions. This is to ensure a minimal delay from when sensor data is read and displayed and ensure accurate data is being presented to users.

5.1.2 The system shall respond to user requests (e.g., button clicks, page switching, submitting information) within 2 seconds for 90% of all requests during nominal operating conditions. This is to ensure optimal user experience and efficiency when navigating and interacting with the application.

5.1.3 The system shall respond to user analysis tool filter requests, i.e. filtering the type of sensor data from a given time to display in the analysis tool, within 5 seconds for 90% of all requests during nominal operating conditions. This is to ensure optimal user experience and efficiency when using the data analysis tool.

5.1.4 The system shall ensure that data uploaded to the cloud is accessible to users within 1 minute after the upload is completed, under nominal operating conditions. This is to ensure optimal user experience and efficiently when accessing data that is uploaded to the cloud.

5.2 Safety Requirements

As the lab experiment includes large tanks of water, it is imperative to remember to exercise caution around the computer, wires, and other equipment. Getting this hardware wet could cause water damage and prevent it from working as intended.

5.3 Security Requirements

As mentioned in the functional requirements, users will have to log into the application with their specific email and password. Once logged, they will only be able to access the features associated with their user role. The system will also record when a user successfully logs in, unsuccessfully logs in, and logs out of the system.

5.4 Software Quality Attributes

The system shall provide a user-friendly user interface that is easy to navigate and intuitive, allowing the client to use the application efficiently and with minimal training. User feedback will be collected to ensure that the user interface meets the user's expectations and needs.

The system shall be scalable to accommodate the user's need for different sensor numbers and configurations. Additionally, the system shall be reliable and robust to ensure data is properly being streamlined from the sensors to the database, user interface, analysis tool, and recorded to the cloud.

Appendix A: Glossary

API: Application Programming Interface, a way for two or more computer programs/components to communicate with each other.

IEEE: Institute of Electrical and Electronics Engineers, professional organization.

RS232: Recommended Standard 232, standard for serial communication of data.

Appendix B: To Be Determined List

2.4 Operating Environment