MAKERERE UNIVERSITY

Software Requirements Specification

for

Low-Cost Water Quality Monitoring System (LCWQMS)

Version 1.0 approved

MAKERERE UNIVERSITY, DEPARTMENT OF NETWORKS

02 NOVEMBER, 2024

Prepared by BSE 25-30

Name	**Registration Number**	**Student Number**
KISEJERE RASHID	21/U/11543/EVE	2100711543
SSEMAGANDA TREVOUR	21/U/18348/EVE	2100718348
SSENTEZA EMMANUEL	21/U/13955/PS	2100713955
GUM PRICILLA	21/U/17674/EVE	2100717674

Table of Contents

Table of Contents

List of Figures

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document specifies the requirements for the Low-Cost Water Quality Monitoring System designed to offer real-time monitoring and predictive insights into water quality using biosensors, artificial intelligence and machine learning. This document describes version 1.0 of the system, detailing the initial scope of the project, which includes the system's capacity to measure key water quality parameters (such as pH, conductivity and total dissolved solids) and its ability to use gathered data for predictive analysis. This SRS covers the entire monitoring system, including data acquisition, processing, machine learning algorithms and reporting features, focusing on providing an affordable and effective solution for water quality assessment.
The SRS plays a crucial role in aligning stakeholders on the objectives and functional expectations of the Low-Cost Water Quality monitoring System. By clearly documenting each requirement, this SRS provides a shared understanding among developers, data scientists and end users, ensuring that the project stays focused on addressing key water quality challenges in a reliable, structured manner. Additionally, the SRS serves as a foundation for quality assurance, enabling thorough testing and validation of the system's features, while also supporting future system scalability and maintainability.

1.2 Document Conventions

The SRS document follows specific conventions to enhance clarity and readability:

- **Text Formatting:** Bold text is used to emphasize section headings and significant terms. Italicized text indicates definitions or explanations of specific terms, justified alignment, Times New Roman, 12pt size and 1.5 line spacing.
- **Numbering System**: Each requirement and sub-requirement is uniquely numbered for easy reference. Major sections are numbered using integers (e.g 1, 2, 3) and sub-sections follow a decima format (e.g 1.1, 1.2, 1.2.1).

- **Requirement Prioritization:** Requirement priority levels are designated as “High”, “Medium” or “Low” at the beginning of each requirement statement. higher-level requirements inherit the priority level of their parent requirement, unless explicitly noted otherwise.
- **References:** External documents and standards are cited by title and are included in the references section.
- **Technical Terminology:** Specialized terms, especially those related to bio-sensors and machine learning, are defined in the Definitions section for clarity.

This document’s formatting and visual aids are designed to promote accessibility and comprehension for a wide range of stakeholders. The combination of clear text formatting, illustrative visuals, and explicit prioritization will result in effective communication and understanding of the software requirements.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) will serve as a valuable resource for various stakeholders involved in the development, deployment and management of the Low-Cost Water Quality Monitoring System. It offers essential insights and guidance for the following readers:

- **Developers:** To use this SRS as a guide for implementing system functionalities, focusing on the System Features and Requirements sections.
  Developers are advised to start with the section 1 for “Overview” to comprehend the project’s scope. They can explore the “System Features” for detailed technical requirements and specifications.
- **Project Managers:** To reference the overall scope, timelines, and constraints, with an emphasis on Section 1 introduction and and 2 which is Overall Description sections. These will also use this document to oversee and guide the design and development process, offering essential insights for the supervising and evaluating the Low-Cost Water Monitoring System prototype against the specified requirements.
  Project Managers should initiate their reading with the “Introduction” and “Overall Description” for an overview. Consequently, they can focus on the “System Features” and other “Non Functional Requirements” to align project goals with system attributes.

- **Testers:** To plan testing strategies based on functional and non-functional requirements.
  Testers should begin with the "Introduction" and "Overall Description" to gain a broad understanding. These can also focus on "System Features" and "Performance Requirements" to comprehend testing scenarios and acceptance criteria.
- **End Users:** To gain an overview of the system's features and capabilities particularly in the product Scope and System Features sections.
  The end users should start with the "Introduction and Purpose" such that they get to know how the system is to serve their needs.

This SRS document is divided into nine sections, each serving a purpose as described in the table below;

Section	**Purpose**
Introduction	This section provides an overview of the document and describes the purpose, intended system audience, document conventions and references.
Overall Description	This section details the product perspective, functions, user classes and characteristics, constraints, assumptions and dependencies of the Low-Cost Water Monitoring System. It addresses user categories, operating environment, design and implementation constraints, system functionality, developer assumptions and the rationale behind design decisions.
External Interface Requirements	This section covers the user interfaces, hardware interfaces, communication protocols and interaction methods essential for achieving the functionalities of the Low-Cost Water

	Quality Monitoring System.
System Features	This section outlines the intended functionalities of the Low-Cost Water Quality Monitoring System, presented through use cases, use case narratives and detailed functional requirements for the system.
Non-Functional Requirements	This section provides a description of the performance, safety, security and software quality requirements for the Low-Cost Water Quality Monitoring System.
Other Requirements	This section covers any extra requirements that haven't been mentioned elsewhere in the document.
Appendix A: Glossary	This section explains the abbreviations and acronyms used in the document..

1.4 Product Scope

The Low-Cost Water Quality Monitoring System is designed to offer an efficient, real-time water quality monitoring solution. The system uses biosensors to continuously measure essential water quality indicators, providing data for AI-driven analysis that forecasts potential water quality issues, pollution levels and public health goals by offering an affordable, easy-to-use monitoring solution, aligning with broader initiatives for sustainable water management.

The water quality monitoring system will:

- **Hardware Component:**
    - ❖ Employ a microcontroller-based platform to interface with various sensors, including pH, conductivity and dissolved oxygen sensors.
    - ❖ Incorporate a power supply and communication module for data transmission.
- **Software Component:**

- Acquire and process sensor data.
- Implement machine learning algorithms to analyze data and generate insights.
- Provide a user-friendly interface for the system configuration and data visualization.

2. Overall Description

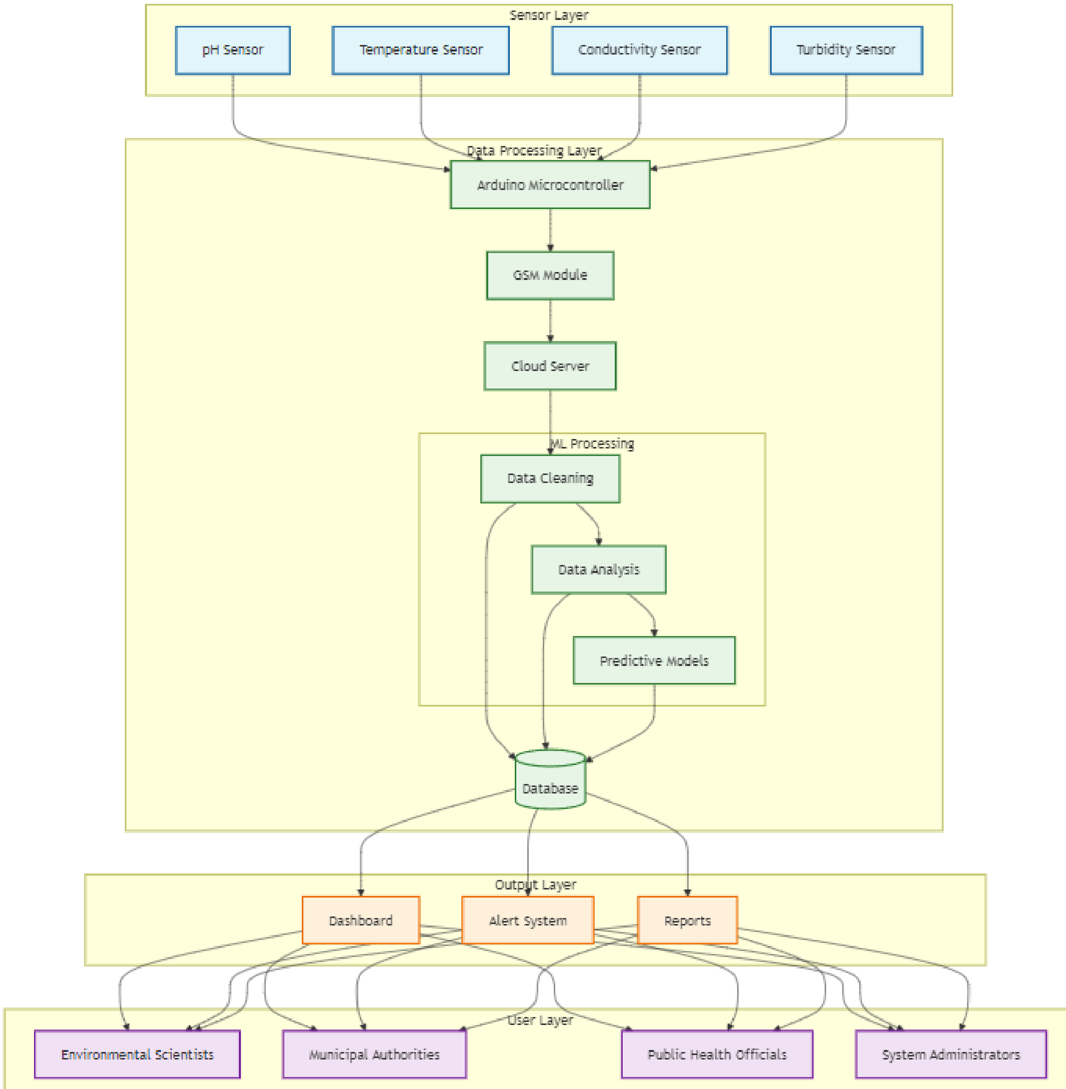

Figure 1 Data flow Diagram of the system

2.1 Product Perspective

The proposed product is an Artificial Intelligence of things (AIoT) system designed for real-time water quality monitoring using a set of specialized sensors. These include sensors for pH, temperature, water conductivity, and turbidity, which measure the respective properties of the water. The system aims to address the challenges associated with manual water quality testing by providing continuous and automated data collection. Each sensor captures key metrics, which are transmitted via a GSM (Global System for Mobile-communications) module to an online web interface. This interface displays the data in real-time, allowing users to monitor trends and anomalies as they occur. This product is a new, self-contained system and is not a follow-on or replacement for any existing solutions currently in use. Existing water quality monitoring methods typically involve manual data collection and testing by professionals, which is both labor-intensive and prone to delays. The novelty of our system lies in automating the entire process, offering seamless, round-the-clock monitoring.

In addition to real-time monitoring, the system integrates two machine learning models that enhance its functionality:

- Forecasting Model: This model analyzes historical and real-time data to predict potential future issues, such as contamination events or critical changes in water quality parameters. Alerts are generated to help users take proactive measures.
- Water Potability Model: This model evaluates the water's overall safety for consumption based on the collected data and presents a clear recommendation on whether the water is suitable for drinking.

The system components are illustrated in Figure 1, which outlines the interconnection between the sensors, the GSM module, the online web interface, and the machine learning models. The GSM module serves as a bridge between the physical sensors and the cloud platform, ensuring seamless data transmission. The web interface not only provides real-time visualization but also triggers alerts based on the machine learning predictions.

In summary, the system offers a novel approach to water quality monitoring in Uganda by automating data collection, providing real-time insights, and leveraging machine learning for predictive analysis. This will address the current limitations of manual testing methods and enable stakeholders to make informed decisions regarding water safety and resource management.

2.2 Product Functions

The proposed IoT-based water quality monitoring system must perform the following high-level functions:

- **Data Collection and Sensing:**
    - ❖ Measure pH, temperature, conductivity, and turbidity of water through integrated sensors.
- **Real-Time Data Transmission:**
    - ❖ Transmit collected data to an online web interface through a GSM module for continuous monitoring.
- **Data Visualization and Monitoring:**
    - ❖ Display sensor readings on a user-friendly web interface in real-time.
    - ❖ Provide trend analysis for better tracking of water quality changes.
- **Machine Learning-Based Forecasting:**
    - ❖ Predict potential future water quality issues (e.g., contamination) through a forecasting model.
    - ❖ Generate alerts to warn users about upcoming risks.
- **Water Potability Assessment:**
    - ❖ Evaluate whether water is safe for drinking using the potability model.
    - ❖ Display clear recommendations about water safety.

2.3 User Classes and Characteristics

The primary user classes anticipated for this IoT-based water quality monitoring system are:

2.3.1 Environmental Scientists and Water Quality Analysts

- ❖ **Frequency of Use:** Frequent, monitoring data daily or weekly.
- ❖ **Functions Used:**
    - ➢ Access real-time and historical water quality data.
    - ➢ Analyze trends and generate reports for environmental studies.
    - ➢ Utilize the **forecasting model** to predict future water quality issues.

2.3.2 Municipal Water Authorities / Public Health Officials

- **Frequency of Use:** Regular, monitoring data weekly or monthly.
- **Functions Used:**

  - Use the **potability model** to assess whether water is safe for public consumption.
  - Monitor real-time alerts for potential contamination events.

2.4 Operating Environment

The IoT-based water quality monitoring system will operate in a distributed environment, comprising physical sensors, cloud infrastructure, and user interfaces. Below is a detailed description of the hardware, software, and other dependencies required for smooth operation:

2.4.1. Hardware Platform

- **IoT Sensors:**
  - pH sensor, temperature sensor, conductivity sensor, and turbidity sensor installed at the monitoring site.
- **GSM Module:**
  - Ensures wireless transmission of collected data to the cloud via mobile networks.
- **Power Supply:**
  - The sensors and GSM module will use battery-powered systems for uninterrupted operation in remote areas.

2.4.2. Software Components and Applications

- **Cloud Platform:**
  - The system will leverage **cloud services** (e.g., AWS IoT, Microsoft Azure, or Google Cloud) for:
    - **Data storage**: Storing real-time and historical water quality data.
    - **Data processing and analytics**: Running machine learning models for forecasting and potability assessments.
    - **Web hosting**: Hosting the user interface for data visualization and reporting.
- **Web Interface:**
  - A responsive web application accessible via modern browsers.
  - The web interface will provide real-time data visualization, and alerts.

2.4.3. Operating System

- **IoT Device Firmware:**
    - The GSM module and sensors will run on lightweight firmware using Arduino microcontrollers.
- **Server Environment:**
    - The backend will use a Python framework for managing data pipelines and API endpoints.

2.4.4. Network Requirements

- **Internet Connectivity:**
    - Reliable mobile network coverage (3G/4G) is necessary for continuous data transmission from the sensors to the cloud.

2.4.5. Dependencies and Integration

- **Machine Learning Models:**
    - Forecasting and potability models will rely on Python-based frameworks such as TensorFlow or scikit-learn.

2.4.6. Environmental Constraints

- **Operating Conditions:**
    - Sensors must function reliably in outdoor environments, withstanding temperature variations, humidity, and water exposure.

In summary, the system's operation depends on hardware platforms (sensors, GSM modules), cloud services for storage and processing, and web-based interfaces for user interaction. Compatibility with cloud infrastructure, web browsers, and machine learning frameworks will guarantee smooth integration and usability.

2.5 Design and Implementation Constraints

The development of the IoT-based water quality monitoring system will be subject to the following constraints:

2.5.1 Hardware Limitations

- **Sensor Accuracy and Calibration:**

  The sensors must be regularly calibrated to maintain accuracy, which could impact data reliability if not managed properly.

- **Power Limitations:**

  Remote installations may rely on solar power or batteries, restricting power-hungry components and limiting the use of high-performance hardware.

- **Network Availability:**

  The system depends on 3G/4G mobile networks, which may not be consistently available in some deployment areas.

2.5.2 Timing and Performance Requirements

- **Real-Time Data Transmission:**

  Data collected from sensors must be transmitted and displayed on the web interface with minimal delay to ensure timely decision-making.

- **Machine Learning Model Performance:**

  Forecasting models must provide quick predictions without introducing significant latency, even with limited cloud resources.

2.6 User Documentation

The following documentation components will be provided to ensure users can effectively operate and maintain the system:

2.6.1 User Manual

- **Description:**

A comprehensive guide covering the system's features, setup procedures, and troubleshooting tips and instructions on accessing the web interface, viewing data, and generating reports.

- **Target Audience:**

  Environmental scientists, municipal water authorities, and general users.

- **Delivery Format:**

  PDF version and printed copies.

2.6.2 System Administration Guide

- **Description:**

  A detailed manual for technicians and system administrators, focusing on system setup, configuration, and maintenance tasks.

  Covers firmware updates, sensor calibration, and network troubleshooting.

- **Target Audience:**

  System administrators and technicians.

- **Delivery Format:**

  PDF version accessible from the admin interface.

2.7 Assumptions and Dependencies

The development and deployment of the IoT-based water quality monitoring system rely on the following assumptions and external dependencies. If these assumptions are incorrect, or if dependencies change during development, the project may encounter delays, increased costs, or the need for redesign.

2.7.1 Assumptions

- **Network Availability**

  Assumes that **3G/4G mobile network coverage** will be available at all monitoring sites to ensure uninterrupted data transmission.

- **Accurate Sensor Performance**

  Assumes that the pH, temperature, conductivity, and turbidity sensors will perform reliably and maintain accuracy over time with routine calibration.

  Impact if Incorrect: Inaccurate sensor data may compromise the machine learning models and lead to faulty decision-making.

- **Cloud Service Stability and Accessibility**

  Assumes that the **cloud platform** (e.g., AWS, Azure) will be available and stable for real-time data processing, storage, and analytics.

  Impact if Incorrect: Outages or service disruptions could delay data processing and result in missing critical alerts.

2.7.2 Dependencies

- **Third-Party Hardware Components**

  The project depends on **commercial sensors** (pH, temperature, conductivity, turbidity) and **GSM modules** from external suppliers.

  Impact if Unavailable: Delays in hardware procurement could push back deployment schedules.

- **Cloud Services and APIs**

  The system relies on **third-party cloud platforms** (e.g., AWS, Google Cloud) for data storage, APIs, and machine learning model hosting.

Impact if Discontinued: Changes in cloud services could force migration to alternative platforms, increasing development time and cost.

- **Machine Learning Libraries and Models**

  The project assumes the continued availability of **open-source libraries** like **TensorFlow** or **scikit-learn** to implement forecasting and water potability models.

  Impact if Unavailable: The development team would need to find or develop alternative libraries, adding complexity.

3. External Interface Requirements

3.1 User Interfaces

The **Low-Cost Water Quality Monitoring System** includes a web-based interface, accessible via both desktop and mobile devices, that allows users to view water quality metrics in real time and historical trends, with alerts for any critical water quality events. This section details the specific characteristics and requirements of this user interface.

1. **Dashboard Overview**:
   - **Purpose**: The dashboard is the main screen where users can monitor water quality parameters (pH, temperature, conductivity, turbidity) in real-time.
   - **Components**:
     - A live data feed section that displays current readings from each sensor.
     - Historical trend graphs for each parameter, showing changes over time to help users identify patterns in water quality.
     - A notification panel that shows alerts or warnings if any parameter falls outside the safe thresholds.
   - **Design**:
     - **Real-Time Display**: Data is updated every few seconds to show live readings from the sensors.
     - **Data Visualizations**: Graphs and charts (such as line graphs for trends or bar graphs for comparison) are used to visualize data. Each parameter has its own graph, allowing users to view historical trends over different time periods (e.g., last 24 hours, past week, etc.).
2. **GUI Standards**:
   - **Consistency**: All screens follow a consistent color scheme (e.g., blue for safe, yellow for caution, red for danger), font styles, and button placements to improve usability and provide a cohesive experience.
   - **Navigation**: A navigation bar on the top or side provides easy access to different sections (e.g., **Home**, **History**, **Settings, Analytics**).

- **Standard Icons**:
  - Icons are used to represent each sensor and parameter type to make the interface more intuitive. For example, a water droplet icon for pH, thermometer for temperature, etc.
- **Mobile Responsiveness**: The layout adjusts to fit various screen sizes, ensuring that the application is easy to use on both desktop and mobile devices.

3. **Screen Layout Constraints**:
   - **Layout Structure**:
     - **Top Section**: Displays the system's status, such as connection status, data update frequency, and last update time.
     - **Main Content Area**: Shows the current readings, visualizations, and historical data for each sensor. Each sensor's data is presented in its own box or card for clarity.
     - **Notification Panel**: Located at the top, this panel displays warnings and error messages. For example, if the pH level exceeds safe limits, a red alert message is shown.
   - **Responsive Design**: The layout adjusts based on the screen size, with a priority on readability and ease of navigation.
4. **Standard Buttons and Functions**:
   - **Help Button**: A help button is present on each screen. When clicked, it opens a pop-up with frequently asked questions and tips on interpreting the data.
   - **Refresh Button**: Allows users to manually refresh the data if there are connectivity issues or a delay in data updates.
   - **Export Data**: An option to download data as a CSV or PDF file, allowing users to analyze it offline. This is especially useful for researchers or water quality officers who need long-term data for reporting.
5. **Keyboard Shortcuts**:
   - **Purpose**: Keyboard shortcuts enable quicker navigation, especially for advanced users.
   - **Examples**:
     - Ctrl + D to go to the **Dashboard**.

- Ctrl + H to open **History**.
- Ctrl + S to open **Settings**.
- Ctrl + E to trigger **Export Data**.

6. **Error Messages**:
    - **Standardized Messages**:
        - **Device Not Found**: Shown if the system cannot connect to the sensors or the GSM module. The message will provide suggestions to check device connections.
        - **Data Loading Error**: Appears when there's an issue retrieving data from the server. The message will prompt users to check their internet connection.
        - **Connection Lost**: A message indicating that the GSM module cannot send data. It provides troubleshooting steps, like verifying the network signal strength.
    - **User-Friendly Language**: Error messages are written in simple language to ensure that users can understand and act on the issues.

3.2 Hardware Interfaces

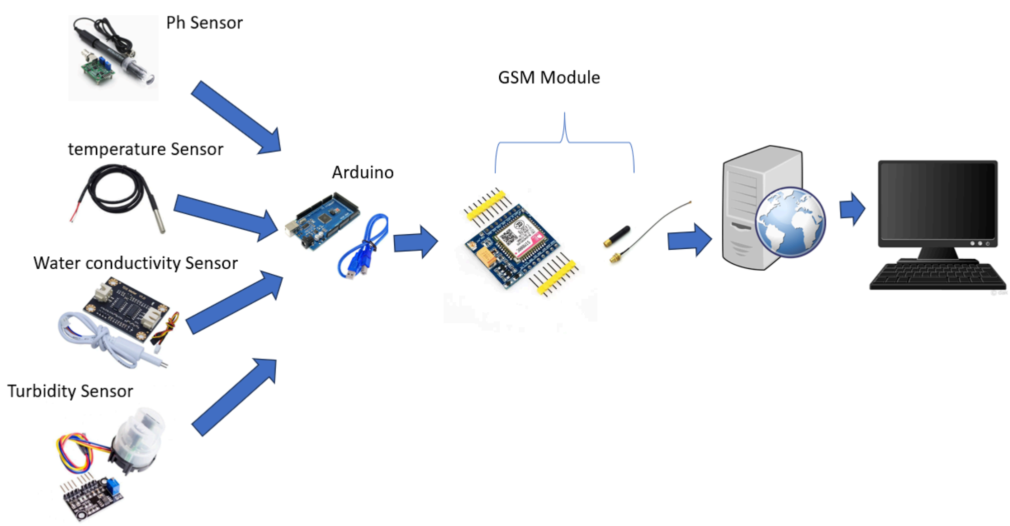

Figure 2 Showing the System Architecture

This section describes the connections between the software and various hardware components, detailing how data flows through the system from sensors to the server and user interface.

1. **Sensor Interface:**
    - **pH Sensor**:
        - **Description**: A pH sensor measures the hydrogen ion concentration in the water, which is used to determine its acidity or alkalinity. pH values typically range from 0 to 14, with values below 7 indicating acidity, 7 being neutral, and values above 7 indicating alkalinity. pH is an essential factor in water quality, as deviations from the normal range can harm aquatic life and indicate pollution.
        - **Function**: Measures the acidity or alkalinity of the water. A change in voltage output indicates a change in pH. Monitoring pH helps assess if the water quality is safe for consumption and supports aquatic life. Abnormal pH values can indicate chemical pollution, acid rain, or other contaminants.
        - **Connection**: The pH sensor can be interfaced with the Arduino using an analog-to-digital converter (ADC). The sensor's probe will produce an analog voltage corresponding to the pH level, which the ADC will convert to a digital value. Arduino can then process this value to calculate the pH level.
        - **Calibration**: To ensure accurate readings, the pH sensor needs to be calibrated periodically using buffer solutions with known pH values, typically pH 4, 7, and 10.

Figure 3 Shows the PH Sensor

- **Temperature Sensor**:
  - **Description**: The temperature sensor measures the water temperature. Temperature is a critical factor in water quality, affecting both chemical and biological processes within the water.
  - **Function**: Measures the water temperature. Monitoring temperature is essential because temperature changes can impact dissolved oxygen levels and metabolic rates of aquatic organisms. Sudden temperature shifts might indicate industrial discharge or thermal pollution.
  - **Connection**: A common temperature sensor for this type of project is the DS18B20 or LM35. The DS18B20 is a digital temperature sensor with a 1-wire interface, making it easy to connect to Arduino using a single data line. For this project, we shall use the DS18B20 temperature sensor because of its advantages of being waterproof, chemical resistant and good for harsh environments.
  - **Calibration**: Temperature sensors usually do not require frequent calibration but can be compared with a known thermometer for verification.

Figure 4 Shows the Temperature Sensor

- **Water Conductivity Sensor**:
  - **Description**: A water conductivity sensor measures the electrical conductivity of the water, which correlates with the amount of dissolved ions or salts.

- **Function**: Measures the ability of water to conduct electricity, which correlates with the level of dissolved salts or impurities. High conductivity can indicate pollution from agricultural runoff, industrial waste, or saltwater intrusion in coastal areas.
- **Connection**: The sensor is connected to an analog pin on the Arduino, where it measures the resistance between two electrodes. This resistance is then converted into conductivity units.
- **Calibration**: Conductivity sensors should be calibrated using solutions of known conductivity to ensure accuracy in readings. Calibration allows the Arduino to interpret the analog output correctly as conductivity measurements.

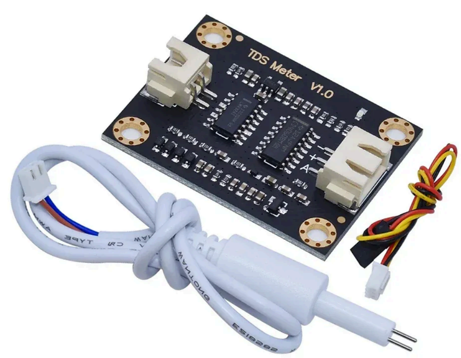

Figure 5 Shows the Water Conductivity Sensor

- **Turbidity Sensor**:
  - **Description**: A turbidity sensor measures the cloudiness or haziness of the water, often due to the presence of suspended particles. Turbidity is a critical indicator of water quality as it directly affects light penetration and can result from sediment, organic material, and pollutants.

- **Function**: Measures water clarity by determining the amount of suspended particles. High turbidity levels can indicate erosion, runoff, industrial pollution, or bacterial contamination.

Figure 6 shows the Water turbidity Sensor

- **Connection**: Turbidity sensors typically output an analog signal that varies with the amount of light passing through the water sample. The sensor is connected to an Arduino analog input pin, where the analog readings are then processed to determine turbidity levels.
- **Calibration**: Turbidity sensors need to be calibrated against samples with known turbidity levels. This helps create a reference to convert the analog readings into Nephelometric Turbidity Units (NTU).

2. **Microcontroller (Arduino):**
    - **Function**: Acts as the central processing unit, gathering data from all sensors and preparing it for transmission.
    - **Connectivity**:
        - **GPIO Pins**: The sensors connect to specific GPIO (General Purpose Input Output) pins on the Arduino, allowing it to receive input data.
        - **Analog Pins**: Sensors with analog outputs connect to these pins for data conversion.
        - **Digital Pins**: Used for sensors with digital outputs.

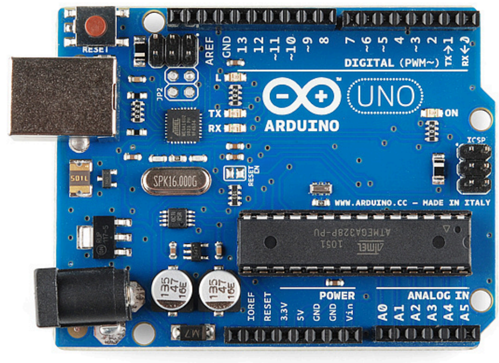

Figure 7 shows the Arduino Micro-controller

3. **Communication Module (GSM):**
   - **Purpose**: Enables wireless data transmission to the remote server.
   - **Interface Type**: UART (Universal Asynchronous Receiver/Transmitter) serial communication.
   - **Data Transmission**: The GSM module takes data from the Arduino and sends it to the server. It uses the cellular network, providing coverage even in remote areas.
   - **Communication Protocol**: Data is typically sent using HTTP or MQTT protocols. HTTP is used for sending data via POST requests to the server, while MQTT can be used for lightweight, low-power communication.

Figure 8 Shows the GSM Module

3.3 Software Interfaces

This section describes the interfaces between the monitoring system's software components and any other software products or services it interacts with, such as the database, APIs, and libraries.

1. **Database Interface:**
    - **Type**: Cloud-based or local database, depending on the deployment environment.
    - **Structure**:
        - Tables store readings for each parameter (pH, temperature, conductivity, turbidity) along with timestamps.
    - **Data Flow**:
        - Sensor data is transmitted from the GSM module to the server and stored in the database in real time.
        - The database stores data securely, and only authorized users can access it.
    - **Data Format**: Data is formatted in JSON (JavaScript Object Notation) before being saved, making it easier to process and visualize later.
2. **Operating System:**
    - The Arduino microcontroller runs a real-time firmware OS designed to handle sensor data collection and GSM communication efficiently.
    - For user-facing applications (desktop or mobile), the system supports major OS platforms such as Windows, macOS, and Linux for desktops, and Android and iOS for mobile devices.
3. **APIs:**
    - **D3.js**: For data visualization on the dashboard, showing real-time and historical water quality data.
    - **Server API**: Receives data from the GSM module and processes it for storage in the database. This API is also used to retrieve historical data when requested by the user interface.
4. **Libraries:**
    - **Arduino Libraries**: Libraries such as TinyGSM are used to facilitate communication between the Arduino and the GSM module.

- **Sensor-Specific Libraries**: Each sensor has a library (or code snippets) that helps read data and interpret sensor values.

5. **Tools and Frameworks**
   - **Tools**
     - **Visual Studio Code:** An Integrated Development Environment that will be used as a code editor and local development environment
     - **Git and GitHub:** Git will be used for version control, allowing team members to collaborate on code development, track changes, and manage project versions. GitHub can be used for repository management and issue tracking, promoting efficient team collaboration and code sharing.
     - **Arduino IDE:** The Arduino Integrated Development Environment (IDE) will be used to write, compile, and upload code to the Arduino board. It provides a simple interface to control sensor data acquisition, process readings, and manage data transmission to the server.
     - **Jupyter Notebooks**: Jupyter notebooks offer an interactive environment for data analysis and experimentation. They will be used to test and iterate on machine learning models, visualize data trends, and document the development process in a collaborative and easily shareable format.
   - **Frameworks**
     - **Django:** Django is a high-level Python web framework designed to build secure, scalable, and maintainable web applications quickly. It follows the Model-View-Template (MVT) architectural pattern, which makes it ideal for applications that require robust data handling, user management, and data visualization. In the **Low-Cost Water Quality Monitoring System**, Django will serve as the backend framework to handle data from biosensors, process it, and provide real-time monitoring and analysis.

     - **React:** React is a popular JavaScript library for building user interfaces, particularly single-page applications where performance and interactivity are essential. It enables developers to create reusable components, manage state efficiently, and update the UI dynamically in response to data changes, making it a suitable choice for interactive dashboards and real-time data visualization. In the **Low-Cost Water Quality Monitoring System**, React will be used as the frontend framework to build an interactive and user-friendly interface, providing a seamless experience for users monitoring water quality.

3.4 Communications Interfaces

This section describes the communication requirements for transmitting data between the system components and external systems

1. **Communication Protocols**:
   - **HTTP Protocol**: Used for sending data from the GSM module to the server via POST requests, ensuring data reaches the server in near real-time.
2. **Data Transfer Rates**:
   - **Sensor to Arduino**: The data transfer rate is instantaneous as it occurs locally.
   - **Arduino to GSM Module**: The serial communication is set at a baud rate of 9600 bps.
   - **GSM to Server**: The rate depends on network speed but aims for one request every 5 minutes to conserve bandwidth and avoid overloading the server.
3. **Data Synchronization**:
   - Data is sent at fixed intervals (e.g., every 5 minutes). If there's a failure, the module retries until successful transmission, preventing data loss.
4. **Communication Security**:
   - **Data Encryption**: AES encryption could be applied to protect data before sending it through the GSM module, securing it against interception.
   - **Authentication**: API keys are used for authorizing data transfers between the GSM module and the server, ensuring only trusted devices can send data.

5. **JSON (JavaScript Object Notation):** In the **Low-Cost Water Quality Monitoring System,** JSON will be a crucial format for transferring data between different components of the system, specifically between the Django backend and the React frontend.

4. System Features

This section details the Predictive Analysis aspect of the system, which leverages machine learning to forecast key water quality parameters and identify risks. It also covers AI-driven Data Processing, which includes data cleaning, anomaly detection, and trend analysis, all of which support accurate predictions. Finally, it describes Real-Time Monitoring through biosensors that continuously track water quality indicators, providing instant alerts and updates. Together, these elements enable proactive management of water quality, combining foresight with real-time monitoring for comprehensive water safety and compliance. .

4.1 Predictive Analysis

4.1.1 Description and Priority

This involves the use of Machine learning based forecasting of water quality parameters like the PH,Turbidity and chlorine levels among others and the consequences of having any abnormalities in these features.

The **priority** of this feature is categorized as **high** since this is the primary obligation and role of the system,to predict future trends and possible consequences of any changes in the physicochemical properties of the water.

- Cost Rating : 6

  Requires constant data processing and model maintenance

- Risk Rating :5

  Medium risk if models are not updated or trained adequately.

- Penalty Rating :5

  Without this feature,predictive capabilities are lost,but real time alerts still cover immediate risks.

- Benefit Rating :8

  Provides foresight into potential water quality issues and alerts on future risks

4.1.2 Stimulus/Response Sequences

a. User Action:Access Predictive Analysis Results

System Response:The system displays the main dashboard with real-time monitoring and a menu to access the Predictive Analysis module. The user selects the "Predictive Analysis" option.

b. User Action:Select Water Quality Parameters and Time Frame

System Response:The system prompts the user to select the specific water quality parameters they want predictions for ,for example pH, nitrate levels,Turbidity,Chlorine levels. The user chooses the parameters and sets a desired time frame For example next week, next month, or a specific date range).

c. User Action:Set Up Automated predictions

System Response:The user opts to schedule automated predictions by setting a frequency For example daily, weekly and desired parameters. The system saves the schedule and sends confirmation to the user.

d. User Action: View Detailed Results

System Response:The system allows the user to click on any parameter in the graph or trend to view more detailed predictions, analysis summaries, and recommendations. The user can expand graphs or view detailed data tables if need

4.1.3 Functional Requirements

REQ-1: The system must allow users to select specific water quality parameters for example pH, nitrate levels, dissolved oxygen from a predefined list.

REQ-2: The system must provide options for the user to set a time frame for predictions for example next week, next month

REQ-3: The system must run the predictive machine learning models using the preprocessed historical data for each selected parameter.

REQ-4: The system must present predicted values in a visual format, including graphs, charts, and trend lines.

REQ-5: The system must allow users to schedule automated predictions for specified parameters and time intervals.

REQ-6: The system must ensure that all historical and predicted data is stored securely, with restricted access based on user roles.

REQ-7: The system must log errors internally for all failed operations and provide meaningful notifications to users for each type of failure.

TBD1: The specific predictive model types (e.g., ARIMA, LSTM) to be used based on performance testing and accuracy assessments.

TBD2: Final thresholds for alerting users based on predicted parameter values (to be determined based on environmental regulations or expert recommendations).

4.2 AI and Machine learning Data Processing

4.2.1 Description and Priority

The system will use AI algorithms to clean and preprocess raw sensor data to remove noise and inconsistencies.On top of that ,Machine learning models shall detect anomalies in water quality based on historical and real-time data patterns, flagging any abnormal values.Also Identification of trends over time in various water quality parameters to predict changes and potential issues using ML models.Identification of trends over time in various water quality parameters to predict changes and potential issues using ML models will also be one of the responsibilities of this feature.

The **priority** of this feature is rated as **high** because the data processing stage is the foundation of the system and hence will be the basis of predictions to be made.

- Benefit Rating:9

Offers high value by automating insights that would otherwise require manual data analysis.

- Penalty Rating:7

Without AI and ML, the system would lack the ability to analyze large datasets efficiently, missing out on critical insights and future trends.

- Cost Rating:7

Implementing AI and ML typically involves substantial costs related to data processing, model development, maintenance, and ongoing updates together with hardware with necessary specifications.

- Risk Rating:6

AI and ML implementations come with risks such as biases in the models, incorrect predictions, or the need for continuous updates to reflect new data trends

4.2.2 Stimulus/Response Sequences

User Action:Upload Historical Data for Model Training

Using python libraries ,the user will have to upload required and necessary datastes for appropriate training of the model

System Process: Train the Machine Learning Model

The system begins model training using the selected dataset and parameters. It displays a progress indicator showing the current status of the training (e.g., percentage complete). During this stage, the system monitors training for overfitting or underfitting.

User Action: Run Predictions

The system allows the user to select specific parameters or datasets for predictions. The user initiates predictions by selecting these options and clicking "Run Prediction."

System Process 4: Generate Predictions and Alerts

The system uses the deployed AI model to generate predictions based on real-time or selected historical data. It displays predicted values and highlights any deviations or alerts based on predefined thresholds for example a sudden spike in nitrate levels).

System Process 3: Evaluate Model Performance

The system evaluates the trained model using a reserved validation dataset. Performance metrics like accuracy, precision, recall, and RMSE are displayed to the user. If the model's accuracy is below a predefined threshold, the system recommends adjustments to training parameters or additional training data.

Dialog Elements Associated with this Use Case

Uploading and Validating Data:

- **Dialog:** "Select file" → "Upload historical data" → "Perform data validation"

Evaluating and Deploying the Model:

- **Dialog:** "Evaluate model performance" → "Review performance metrics" → "Save and deploy model"

Viewing AI Insights:

- **Dialog:** "Click on alerts" → "View detailed insights and recommendations"

4.2.3 Functional Requirements

REQ-1: The system must allow users to upload historical water quality data files in supported formats for example CSV ,Excel.

REQ-2: The system must provide a preview of the uploaded data, showing a few rows to help users verify their upload.

REQ-3: The system must automatically clean and preprocess uploaded data, including handling missing values,removing outliers.

REQ-4: The system must evaluate the trained model using a separate validation dataset and display performance metrics for example accuracy, precision, recall, RMSE.

REQ-5: The system must run the trained model to generate predictions and display the results in both graphical and tabular formats.

TBD1: The criteria for threshold-based alerts for specific water quality parameters (based on environmental guidelines or domain expert recommendations).

TBD2: Selection of specific AI and ML models for different water quality parameters for example time-series models, neural networks, based on extensive model testing.

TBD3: Implementation of advanced explainability techniques to allow users to understand the basis of each prediction.

4.3 Real-Time Water Quality Monitoring

4.3.1 Description and Priority

There will be deployment of biosensors for detecting physical, chemical, and biological parameters such as pH, temperature, dissolved oxygen, turbidity, nitrate levels, heavy metals, and microbial contaminants.

The priority of this feature is equally high since sensors deployed will be incharge of reporting real time changes in the state of the water.

- Benefit Rating:9

Real-time monitoring provides immediate insights into water quality, enabling swift responses to contamination events, which is crucial for public health and environmental safety.

- Penalty Rating:8

Without real-time monitoring, there is a high risk of delayed responses to water quality issues, potentially leading to health risks and environmental damage.

- Cost Rating:7

Implementing a real-time monitoring system will be quite costly due to the need for advanced sensors, data transmission technologies, and ongoing maintenance which requires a good fortune of funds to access.

- Risk Rating:6

While real-time system is essential, it will come with risks such as potential sensor malfunctions, data transmission issues, and cybersecurity threats together with sensors being stolen by malicious individuals

4.2.2 Stimulus/Response Sequences

User Action: Configure Monitoring Parameters

The system displays a configuration interface where the user can set parameters for monitoring (e.g., specific water quality indicators like pH, turbidity, temperature)

User Action: Initiate Real-Time Monitoring

The user clicks on "Start Monitoring." The system activates the sensors in the water body, establishing a connection for real-time data transmission. It displays a message: *"Real-time monitoring has started."*

System Process: Data Collection

The system begins collecting data from the sensors at predefined intervals (e.g., every 10 seconds). It logs this data for processing and analysis, ensuring data integrity and quality.

User Action: View Real-Time Data Dashboard

The user accesses a dashboard that visualizes real-time water quality data through graphs and metrics. The system updates the dashboard automatically with the latest data and highlights any critical values (e.g., pH levels exceeding safe limits).

System Process: Continuous Monitoring and Reporting

The system continues real-time monitoring, logging data and generating periodic reports on water quality trends. It sends daily/weekly summaries to users, allowing for long-term analysis and decision-making.

User Action 3: View Real-Time Data Dashboard

The user accesses a dashboard that visualizes real-time water quality data through graphs and metrics. The system updates the dashboard automatically with the latest data and highlights any critical values (e.g., pH levels exceeding safe limits).

4.2.3 Functional Requirements

REQ-1: The system must support integration with various water quality sensors (e.g., pH, turbidity, dissolved oxygen, temperature) for data collection.

REQ-2: The system must continuously collect data from the sensors at predefined intervals (e.g., every 10 seconds).

REQ-3: The system must process incoming data in real-time, applying algorithms to evaluate water quality based on predefined thresholds.

REQ-4: The system must detect anomalies in real-time data and flag them for further review.

REQ-5: The system must provide a user-friendly dashboard that displays real-time water quality metrics in graphical and tabular formats.

REQ-6: The dashboard must be updated automatically with the latest data without requiring manual refresh.

REQ-7: The system must log all incoming data along with timestamps for historical analysis.

TBD1: Additional parameters or features based on specific environmental regulations or user requirements.

TBD2: Integration with third-party systems or APIs for extended data analysis or sharing.

5. Other Nonfunctional Requirements

NFREQ-1: The system must achieve an uptime of 99.9% to ensure continuous monitoring without interruptions.

NFREQ-2: The system must enforce strong user authentication mechanisms, including multi-factor authentication (MFA) for access to sensitive functionalities.

NFREQ-3: The system must be designed to easily scale up to accommodate an increase in the number of sensors and data volume without significant architectural changes.

NFREQ-4: The system must include comprehensive documentation for both users and developers, detailing functionalities, APIs, and maintenance procedures.

NFREQ-5: The system must be designed to operate effectively in varying environmental conditions (e.g., temperature, humidity) typical of outdoor water monitoring setups.

5.1 Performance Requirements

Response Time

- Dashboard updates must occur within 2 seconds of new data arrival
- Alerts must be generated within 5 seconds of detecting anomalous conditions
- ML predictions must be completed within 30 seconds of request initiation

Availability

- System uptime of 99.9% excluding scheduled maintenance
- Maximum scheduled downtime of 4 hours per month
- Automatic failover within 60 seconds for critical components

5.2 Safety Requirements

Water Quality Monitoring

- Immediate alerts when water quality parameters exceed safe thresholds
- Triple redundancy for critical measurements affecting public health

Hardware Safety

- All sensors must be waterproof with IP68 rating
- Electrical components must be properly insulated and grounded
- Battery-powered components must include overcharge protection

Environmental Safety

- Sensor materials must be non-toxic and environmentally safe
- System must comply with environmental protection standards
- Proper disposal procedures for replaced sensors and components

5.3 Security Requirements

Data Security

- End-to-end encryption for all data transmission
- AES-256 encryption for stored data
- Secure backup system with 30-day retention

Network Security

- Firewall protection for all system components
- Regular security audits and penetration testing

Compliance

- Regular security assessment documentation
- Incident response plan documentation

5.4 Software Quality Attributes

Reliability

- Maximum of one critical failure per year
- 99.9% accuracy in sensor readings
- Automated error detection and correction

Maintainability

- Modular design for easy component replacement
- Comprehensive documentation of all APIs
- Automated testing coverage of at least 80%

Usability

- Maximum 1-hour training required for basic users
- Intuitive dashboard design with consistent layout
- Mobile-responsive interface

Scalability

- Linear performance scaling with increased load
- Automated resource scaling capabilities

5.5 Business Rules

User Roles and Permissions

System Administrator: Full system access and configuration

Environmental Scientist: Access to all data and analysis tools

Water Quality Officer: Access to monitoring and alerts

Public Health Official: Access to potability assessments

General User: Access to public dashboards only

Operational Rules

Mandatory daily system health checks

Weekly backup of all system data

Monthly performance reports generation

Quarterly system audit requirements

Data Management

Data retention period of 5 years minimum

Automated archiving of data older than 1 year

Regular data quality assessments

6. Other Requirements

Internationalization

- Localized date and time formats
- Region-specific water quality standards

Legal Requirements

- Compliance with local water quality regulations
- Data privacy law compliance

Documentation Requirements

- User manuals in all supported languages
- Technical documentation for system maintenance
- API documentation for integration
- Regular updates to reflect system changes

Integration Requirements

- API support for third-party systems
- Standard data format support (JSON, CSV)

References

1. "Water quality monitoring and assessment using IoT technology in developing countries: Current status and future directions," *Modeling Earth Systems and Environment*, Springer. [Online]. Available: https://link.springer.com/article/10.1007/s40808-020-01041-z
2. "A real-time water quality monitoring and control system using IoT and machine learning for sustainable water resource management," *IEEE Xplore*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9358675/
3. "Smart water quality monitoring system for real-time data analysis and environmental impact assessment," *HEG-Geneva*. [Online]. Available: https://arodes.hes-so.ch/record/8076?ln=fr&v=%5B%27pdf%27%5D
4. "Recent advances in low-cost water quality monitoring systems using IoT-based biosensors," *Scientific African*, ScienceDirect. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666285X2100090X
5. "Emerging biosensor technologies for water quality monitoring: Applications and challenges," *Sensors Journal*, MDPI. [Online]. Available: https://www.mdpi.com/1424-8220/22/11/4078
6. NWSC, "WATER QUALITY AT THE HEART OF NWSC OPERATIONS – Water Herald," *NWSC Water Herald*. [Online]. Available: https://nwscwaterherald.co.ug/water-quality-at-the-heart-of-nwsc-operations/

Appendix A: Glossary

Acronyms and Abbreviations

- **AI**: Artificial Intelligence
- **AIoT**: Artificial Intelligence of Things
- **API**: Application Programming Interface
- **ARIMA**: AutoRegressive Integrated Moving Average
- **AES**: Advanced Encryption Standard
- **CSV**: Comma-Separated Values
- **GSM**: Global System for Mobile Communications
- **GPIO**: General Purpose Input/Output
- **HTTP**: Hypertext Transfer Protocol
- **IDE**: Integrated Development Environment
- **IP68**: Ingress Protection rating 68 (dust-tight and protected against long-term water immersion)
- **JSON**: JavaScript Object Notation
- **LSTM**: Long Short-Term Memory
- **MFA**: Multi-Factor Authentication
- **MQTT**: Message Queuing Telemetry Transport
- **NTU**: Nephelometric Turbidity Units
- **RMSE**: Root Mean Square Error
- **SRS**: Software Requirements Specification
- **UART**: Universal Asynchronous Receiver/Transmitter

Technical Terms

- **Anomaly Detection**: Process of identifying unexpected patterns or outliers in data that deviate from normal behavior.
- **Biosensor**: An analytical device that converts biological responses into electrical signals.
- **Conductivity**: A measure of water's capability to pass electrical flow, indicating the concentration of dissolved solids.

- **Machine Learning Model**: A program that can learn from data to make predictions or decisions without being explicitly programmed.
- **pH**: A measure of how acidic/basic water is, ranging from 0 (acidic) to 14 (basic).
- **Predictive Analysis**: Use of statistical algorithms and machine learning techniques to identify likely future outcomes based on historical data.
- **Real-time Monitoring**: Continuous collection and analysis of data as it is generated.
- **Turbidity**: A measure of water clarity or cloudiness caused by suspended particles.

System-Specific Terms

- **Dashboard**: Web-based interface displaying real-time and historical water quality data.
- **Data Processing Pipeline**: Series of steps for cleaning, validating, and analyzing sensor data.
- **Forecasting Model**: Machine learning model specifically designed to predict future water quality parameters.
- **Potability Model**: AI model that determines if water is safe for human consumption.
- **Water Quality Parameters**: Measurable properties of water that indicate its quality (pH, temperature, turbidity, etc.).

Appendix B: To Be Determined List

Predictive Analysis TBDs

1. TBD1: Specific predictive model types (ARIMA, LSTM) to be selected based on performance testing and accuracy assessments.
2. TBD2: Final thresholds for alerting users based on predicted parameter values.

AI and Machine Learning Data Processing TBDs

3. TBD1: Criteria for threshold-based alerts for specific water quality parameters.
4. TBD2: Selection of specific AI and ML models for different water quality parameters.
5. TBD3: Implementation of advanced explainability techniques for predictions.

Real-Time Water Quality Monitoring TBDs

6. TBD1: Additional parameters or features based on specific environmental regulations or user requirements.
7. TBD2: Integration with third-party systems or APIs for extended data analysis or sharing.

Status Tracking

Each TBD item will be tracked with:

- Priority Level
- Assigned Owner
- Target Resolution Date
- Dependencies
- Current Status