

SOFTWARE REQUIREMENTS  
SPECIFICATION - Machine Learning Platform  
for Intelligent Water Systems Management -  
Senior Design Project Dashboard

Robert Castro  
Calvin Chau  
Yvan Michel Kemsseu Yobeu  
Laila Velasquez  
Kassandra Vera

## Table of Contents:

Table of Contents ..... pg2

Revision History ..... pg3

1. Introduction ..... pg4
  - 1.1 Purpose ..... pg4
  - 1.2 Intended Audience ..... pg4
  - 1.3 Product Scope ..... pg4
  - 1.4 Overview ..... pg4
2. External Interface Requirements ..... pg5
  - 2.1 User Interface. .... pg5
  - 2.2 Software Interface. .... pg5
  - 2.3 Communications Interface ..... pg5
  - 2.4 Functional Requirements ..... pg5
  - 2.5 Design Constraints ..... pg6
3. Ethical and Legal Considerations. .... pg6
  - 3.1 User Data ..... pg6
  - 3.2 User Storage ..... pg6
  - 3.3 Security Requirements ..... pg6
  - 3.4 Safety Requirements ..... pg7
  - 3.5 Moral Dilemma ..... pg7
4. Glossary ..... pg8

## Revision History

Name	Date	Reason For Changes	Version
Calvin Chau	5/5	1.0-2.5	1.0
Calvin Chau	5/7	3.0-4.0	1.1

## **1) Introduction**

### **1.1 Purpose**

The purpose of our Software Requirements Specification (SRS) document is to outline the necessary requirements for this dashboard while providing additional information about the dashboard.

### **1.2 Intended Audience**

The intended audience for our Software Requirements Specification (SRS) document includes:

Building Owners and Faculty Managers who oversee water systems in densely occupied buildings

Developers and Designers involved in developing the dashboard site

Saya Life Stakeholders who may integrate the dashboard into their website

### **1.3 Product Scope**

Our dashboard aims to use Saya Life's dataset to generate a building access report that details potential leakages.

We plan to potentially integrate our dashboard into the Saya Life website. We hope that users of our dashboard can make better informed decisions regarding possible leakages in buildings, such as insurance companies.

### **1.4 Overview**

This SRS document provides a detailed specification of the dashboard for the Machine Learning Platform for Intelligent Water Systems Management. It will present information regarding General Overview, Functional Requirements, External Requirements, and Ethical/Legal Considerations. Groups/Individuals are recommended to read this document before accessing/using the dashboard. It serves as a reference for developers, users, and stakeholders to understand system requirements and use cases. By following the guidance and requirements in this SRS, users can better manage water usage and address potential leakages in a timely and efficient manner.

## **2) External Interface Requirements**

### **2.1 User Interface**

The UI for the dashboard will be web-based and focus on clarity and ease of use.

The dashboard will be able to:

- Use charts and tables to display real-time and historical water usage data.
- Visually highlight outliers and potential leaks to display anomalies
- Provide user-friendly navigation through the dashboard, such as having building-specific views.
- Have optimized performance for responsiveness on both desktops and tablets
- Contain login functionality and unique features for users and admins

### **2.2 Software Interface**

The dashboard will use the following software systems:

- Saya Life API provides access to data regarding building water usage and sensor data.
- Database Management System for storing processed data and user information.
- Backend services, such as Node.js, to handle API requests, process data, and communicate with the frontend.
- Docker is used for containerizing the application components (frontend, backend, database), and Docker Hub hosts versioned images of the dashboard, allowing streamlined deployment and consistent environments across machines.
- OAuth for secure user login.

### **2.3 Communications Interface**

Operation communications within the dashboard include:

- HTTPS Protocol secures the transmission of data between frontend, backend, and external APIs.
- Jira compiles bug/error reports sent from users to admins for inspection and possible fixes.

### **2.4 Functional Requirements**

The dashboard supports the following functional requirements:

- User Authentication: Users must be able to register, log in, and log out securely. Admins are granted higher-level permissions for bug management and moderation.
- Data Visualization: Users must be able to view related graphics and data of provided time instances in real-time. Data displayed must be clear and concise for ease of access.
- Leak Detection Alerts: Dashboard/System must be able to detect anomalies

in water usage and flag potential leaks to related users, preferably users who are viewing or are near the specified area.

- Building Specific Filtering: Users must be able to filter water usage data by specific locations or buildings for specific viewing.
- Report Generation: Users must be able to generate reports related to water usage trends and anomalies. These reports must be downloadable for users to use for any purpose.
- Bug Reporting: Users must be able to send downloadable reports using the bug reporting feature. Reports sent must be able to be logged and notify admins/designers through Jira for review.

## **2.5 Design Constraints**

The dashboard is subjected to the following constraints:

- Technologies: The dashboard is deployed by modern web technologies (React and Django), OAuth 2.0 is used for user/admin authentication, and Jira is used for project management and bug tracking.
- Containerization: Project components must be containerized using Docker to ensure operational consistency across targeted environments.
- Cross-Platform Accessibility: The dashboard must be usable on modern web browsers on both desktop and tablet devices.
- Security: Communications between dashboard operations must occur over HTTPS. Sensitive data such as passwords and user tokens must be stored securely in an encrypted database.

## **3) Ethical and Legal Considerations**

### **3.1 User Data**

The dashboard may handle sensitive user-related data. These include user login credentials, usage behavior, and location-specific water usage patterns. For full transparency for our users, all data collection must be minimized to what's necessary for functionality, and we will inform users about data collection and its purpose.

### **3.2 User Storage**

User credentials and system activity must be securely stored in an encrypted format using practices such as password hashing and data encryption. Personally identifiable information will not be stored without a proper reason, and any stored data must comply with current data protection regulations.

### **3.3 Security Requirements**

Our dashboard and systems fit the following criteria:

- All data transmissions must occur over HTTPS

- User login credentials (Username, password, email) must be encrypted (bcrypt) and never be stored in plain text for security
  - Session tokens secured by JWT or passport will be used for maintaining authenticated sessions within the system
  - Regular vulnerabilities and code tests must be upheld regularly to stabilize the system
- These methods must be upheld by the system to promote a safe and secure environment for users of the dashboard.

### **3.4 Safety Requirements**

Although the dashboard does not directly control physical systems, inaccurate or delayed leak alerts may result in potential cases of water damage or health risks (mold, erosion, etc). The system should prioritize alert accuracy and clear communication, and all warnings should be labeled appropriately.

### **3.5 Moral Dilemma**

Potential moral dilemmas brought by the dashboard mainly involve how the system's display of water usage data may be interpreted or used. Two main problems may arise regarding this system.

Insurance companies or landlords may use this information to penalize their tenants or reject claims using the dashboard's leak detection data without proper context. There is a risk that this data may be misused to misrepresent context for malpractices.

The availability of users to view and download building water usage may unintentionally expose private information through pattern recognition. This could raise ethical concerns regarding non-consensual surveillance and breach of privacy.

## 4) Glossary

The Glossary defines terms, acronyms, and abbreviations used throughout the SRS to give clarity and consistency for all readers.

- SRS (Software Requirements Specification): Represents this document. Defines the functional and non-functional requirements of the dashboard, along with additional insights.
- UI (User Interface): The visual and interactive parts of the system that users interact with to traverse the dashboard.
- API (Application Programming Interface): A set of protocols that allows different software systems to communicate, such as retrieving data from Saya Life.
- JWT (JSON Web Token): A secure and compact way to represent session data and maintain authenticated sessions within the dashboard.
- OAuth 2.0: An open standard protocol used for secure user authorization.
- HTTPS (HyperText Transfer Protocol Secure): The secure version of HTTP used to encrypt communications over the web.
- Jira: A project management tool used for tracking tasks, bugs, and project progress.
- Docker: A platform used to package and deploy applications in containers to ensure consistency across environments.
- Docker Hub: A cloud-based registry where Docker images can be stored and shared.
- Saya Life API: The data provider API that supplies sensor and water usage information.
- Bcrypt: A password hashing algorithm used to securely store user credentials.