Software Requirements Specification

Water Management System

Name:- Tanvi Amit Patel
College Name:- Deogiri Institute of Engineering and Management Studies,Chh. Sambhajinagar
Roll no. :- CS3155
Class:- TY-A
PRN:- 23021141242064
Title:- Water Management System

Table of Contents

1. Introduction 1
   1.1 Purpose 1
   1.2 Scope 2
   1.3 Definitions, acronyms, and abbreviations 2
   1.4 References 3
2. Overall description 4
   2.1 Product perspective 4
   2.2 Product functions 4
   2.3 User characteristics 5
   2.4 Constraints 5
   2.5 Assumptions and dependencies 6
   2.6 Apportioning of requirements 6
3. Specific requirements 7
      3.1.1 User interfaces 7
      3.1.2 Hardware interfaces 7
      3.1.3 Software interfaces 8
      3.1.4 Communications interfaces 8
   3.2 Functional requirements 9
      3.2.1 User Class 1 - The User 9
      3.2.2 User Class 2 - Restaurant Owner 10
      3.2.3 User Class 3 - Administrator 13
   3.3 Non-Functional requirements 15
   3.4 Design constraints 16
   3.5 Software system attributes 18
4. Analysis Models 20
5. Prioritization and Release Plan 21
   5.1 Choice of prioritization method 21
   5.2 Prioritization Results 22
   5.3 Release Plan 23

1. Introduction

This section provides the scope description and overview of all elements included in this Software Requirements Specification (SRS) document for the Water Management System (WMS)**.** It outlines the purpose of the system, its scope, the key definitions and acronyms, and the references that support this work.

The purpose of this document is twofold:

1. To act as a formal agreement between the customer, stakeholders, and the development team regarding the expectations and functionality of the system.
2. To serve as a guideline for the development team in designing and implementing the first version of the Water Management System.

1.1 Purpose

The primary purpose of this document is to provide a comprehensive description of the Water Management System (WMS). The WMS is designed to help individuals, communities, and organizations monitor, track, and manage water consumption efficiently, with the ultimate goal of promoting sustainable water usage and minimizing wastage.

The document serves multiple purposes:

1. For Customers/Stakeholders – it describes what the system is intended to do, why it is being developed, and how it will address the issues related to inefficient water usage. It provides a clear understanding of the services offered, so that customers and end-users can evaluate whether the system meets their needs.
2. For the Development Team – it acts as a blueprint during system design and implementation. Every requirement specified in this document will serve as a guideline for building the system, ensuring that developers and designers do not deviate from the intended objectives.
3. For Future Maintenance and Extension – this document also serves as a reference for future upgrades and modifications. As the WMS is expected to expand in later versions (e.g., integration with IoT-based smart water meters, leak detection sensors, and AI-driven predictions), the requirements here lay the foundation for scaling the system.

The Water Management System (WMS) will provide services to three main user groups:

- Individual Users: who will log their daily water usage, receive reminders, and monitor their consumption patterns.
- Water Resource Managers: who will supervise water consumption across communities, housing societies, or industries.
- Administrators: who will manage the system's operations, verify users/managers, and ensure system stability and accuracy.

In short, this document intends to clarify the objectives, requirements, and constraints of the WMS so that all parties involved share a common understanding of the system to be developed.

1.2 Scope

The Water Management System (WMS) is a web and mobile-based application that facilitates the tracking, analysis, and optimization of water consumption**.** It enables users to set reminders, log usage, and view analytics at both personal and community levels.

The scope of the system includes:

1.2.1 Individual-Level Functions

- User registration and profile creation.
- Manual logging of water usage (daily, weekly, monthly).
- Smart reminders and alerts (e.g., irrigation schedules, household water usage).
- Visualization tools such as dashboards, charts, and reports.
- Goal-setting and progress tracking to encourage conservation.

1.2.2 Community/Manager-Level Functions

- Centralized monitoring of water distribution within a community or industrial setup.
- Detection of abnormal consumption (e.g., sudden spikes indicating leakage or overuse).
- Report generation for decision-making and policy formulation.
- Scheduling notifications about water supply interruptions or maintenance work.

1.2.3 Administrative Functions

- User management (creation, deletion, role assignment).
- Verification of water managers before they can supervise communities.
- System updates and maintenance.
- Integration with third-party APIs and IoT sensors for future scalability.

The goals of WMS can be summarized as:

1. **Water Conservation** – promote sustainable practices and reduce wastage.
2. **Awareness Building** – help users understand their usage patterns.
3. **Efficient Resource Allocation** – assist managers in optimizing community water distribution.
4. **Data-Driven Decisions** – provide analytics for better planning and management.

The initial release will focus on manual usage logging and reminder notifications, with the system designed in a modular fashion to allow future enhancements like automatic IoT-based consumption tracking.

1.3 Definitions, acronyms, and abbreviations

Table 1 - Definitions

WMS	Water Management System – the software being developed.
User	An individual who uses the system to monitor and manage personal water usage.

Water Resource Manager	A user with extended privileges to monitor and manage water consumption at a community or organizational level.
Administrator (Admin)	A user with the highest level of control, responsible for managing the system, verifying users, and maintaining data integrity.
GUI	Graphical User Interface – the interface that users interact with.
DBMS	Database Management System – used to store and retrieve water usage data.
API	Application Programming Interface – a set of tools that allows communication with external systems (e.g., weather forecasts, IoT sensors).
IoT	Internet of Things – networked devices like smart water meters or leak detectors that could be integrated into WMS.
Analytics Dashboard	A visual reporting interface for presenting water usage trends and statistics.
Stakeholder	Any individual or group with an interest in the system, including users, managers, administrators, and developers.

1.4 References

The WMS SRS has been prepared with reference to:

1. IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications.
2. United Nations Sustainable Development Goal (SDG 6) – Ensure availability and sustainable management of water and sanitation for all.
3. Research Papers on Water Management:
    - ―Smart Water Management in Urban Areas Using IoT,‖ IEEE Access, 2023.
    - ―Data-Driven Decision Support for Water Distribution Networks,‖ Springer, 2022..
4. Course materials and textbooks on Software Engineering, Requirements Analysis, and Sustainable Systems Design.

1.5 Overview

The remainder of this document is structured as follows:

- Chapter 2: Overall Description provides a high-level view of the Water Management System and explains how it fits into its operational environment. It describes the overall product perspective, major functions, user characteristics, system constraints, assumptions, dependencies, and the allocation of requirements between system releases.
- Chapter 3: Specific Requirements contains a detailed account of the functional and non-functional requirements of the WMS.

2. Overall description

This section provides an overall perspective of the Water Management System (WMS), its environment, functionality, and the nature of its users. It describes how the system interacts with external entities, defines the product functions, identifies the stakeholders, and highlights assumptions, dependencies, and constraints. Since the WMS is envisioned to serve both individual users and organizations, this chapter also discusses the roles of the three main classes of users — the end-user, the water resource manager, and the administrator.

2.1 Product perspective

The Water Management System (WMS) is a data-driven application designed to operate as both a mobile and web-based platform. It will function as an independent system but is designed to allow future integration with IoT devices such as smart water meters, flow sensors, and leak detectors.

The WMS can be viewed as consisting of two main components:

1. **Mobile Application** – primarily designed for individual users to log water usage, set reminders, receive notifications, and view personal consumption patterns.
2. **Web Portal (Dashboard)** – designed for water resource managers and administrators. It provides advanced features such as aggregated data visualization, reporting tools, anomaly detection (e.g., leakage or unusual spikes in usage), and management of community-level resources.

Both the mobile and web components are backed by a centralized database server that stores user profiles, historical data, reminders, and analytical results. The system communicates with the database through a secure API layer over the internet.

A possible future enhancement is the integration with third-party services, such as:

- **Weather APIs** to predict water needs (e.g., irrigation schedules based on rainfall forecasts).
- **IoT-based smart meters** for automated water usage logging.
- **Municipal water supply APIs** to notify users about upcoming supply interruptions or rationing schedules.

To ensure portability and scalability, the WMS will follow a modular architecture. The mobile application will focus on personal tracking and notifications, while the web portal will concentrate on large-scale monitoring and analytics.

2.2 Product functions

The WMS offers a wide variety of functions tailored to different stakeholders. These can be summarized as follows:

2.2.1 For Individual Users

- Register and create a personal profile.
- Log daily, weekly, and monthly water usage.
- Set reminders and alerts for household or personal water consumption.

- View consumption statistics through interactive charts and dashboards.
- Receive personalized recommendations for reducing water usage.

2.2.2 For Water Resource Managers

- Access aggregated data from multiple users or communities.
- Detect and analyze abnormal consumption patterns.
- Generate reports for decision-making (e.g., usage distribution across households).
- Send notifications about water supply issues (outages, rationing, or quality alerts).
- Track long-term trends for better resource planning.

2.2.3 For Administrators

- Manage all user accounts (creation, modification, deletion).
- Verify and authorize water resource managers.
- Maintain and monitor system performance.
- Control integration with external APIs or IoT devices.
- Configure system-wide policies and usage thresholds.

These functions together make the WMS not only a personal water management app but also a community-level monitoring system**,** ensuring a balance between individual convenience and collective sustainability**.**

2.3 User characteristics

The WMS will be used by three categories of users:

1. End-Users (General Public):
    - Expected to have basic smartphone literacy.
    - Will primarily interact with the mobile application.
    - Require a simple, intuitive interface with minimal setup.
    - Motivated by personal health, financial savings, and environmental awareness.
2. Water Resource Managers:
    - Typically belong to housing societies, municipalities, or industries.
    - Need advanced tools for data monitoring, report generation, and community-level insights.
    - May have moderate technical knowledge, so the web portal must balance detail with usability.
3. Administrators:
    - Usually IT staff or authorized system operators.
    - Possess high technical skills and responsibility for maintaining the WMS.
    - Need full system access, including database monitoring, API management, and user account control.

Thus, the system is designed to accommodate a wide range of technical expertise, ensuring ease of use for general users while still providing advanced controls for experts.

2.4 Constraints

Several constraints limit the design and operation of the WMS:

- **Connectivity Requirement:** The WMS relies heavily on internet connectivity for syncing data with the central database. Without an internet connection, only limited offline functions (such as local reminders) may be available.
- **Device Limitations:** Mobile devices have constraints in terms of memory and storage. The mobile app must be lightweight (less than 50 MB) and optimized to run on mid-range smartphones.
- **Data Privacy:** User water consumption data must be securely stored and transmitted, complying with relevant data protection standards.
- **System Load:** The centralized database must handle requests from thousands of users simultaneously without significant delay.
- **Dependency on External APIs:** Features such as weather-based predictions or municipal alerts depend on third-party APIs. Failure or downtime of these services may affect system performance.

2.5 Assumptions and dependencies

The design of the WMS is based on several assumptions and dependencies:

- Users have access to smartphones and/or computers with stable internet connectivity.
- Community water managers and administrators have the necessary authority and willingness to adopt digital tools for monitoring.
- Weather forecasting APIs and IoT devices (when integrated) provide accurate and timely data.
- Users are motivated to log their water usage honestly and regularly (until full IoT automation is implemented).
- Municipalities or communities provide access to necessary data for broader water distribution monitoring.

If these assumptions do not hold, the system's effectiveness may be reduced.

2.6 Apportioning of requirements

In case of project delays or limited resources, certain requirements may be postponed to future releases:

- **First Release:** Core features such as user registration, manual logging of water usage, reminders, and basic dashboards.
- **Second Release:** Advanced analytics, report generation, and community-level monitoring features for managers.
- **Third Release:** Integration with IoT devices, predictive modeling using weather data, and large-scale municipal dashboards.

This approach ensures that even the first version of the WMS is a fully functional product, while advanced features can be gradually introduced without disrupting the core system.

3. Specific Requirements

This chapter describes the detailed functional and non-functional requirements of the Water Management System (WMS). It is organized into subsections covering user interfaces, hardware and software interfaces, communication interfaces, functional requirements for each user class, performance requirements, design constraints, and software system attributes.

The requirements are presented in a structured way to ensure that developers, users, and stakeholders all have a clear understanding of the system's intended behavior.

3.1 External interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1.1 User interfaces

The WMS will provide two main user interfaces:

1. Mobile Application Interface (for End Users):
    - Designed for ease of use with simple navigation and attractive design.
    - Key screens include:
        - Login & Registration Page – secure user authentication.
        - Home Dashboard – displays today's water usage, goals, and reminders.
        - Reminder Setup Screen – lets users configure household reminders (e.g., irrigation, personal drinking, or cleaning).
        - Analytics Page – provides visual charts of daily/weekly/monthly usage.
        - Settings Page – profile updates, notification preferences, and language selection.
    - Must follow standard UI/UX design principles (minimal steps, readable fonts, accessibility support).
2. Web Dashboard Interface (for Managers & Admins):
    - More data-heavy, designed for larger screens.
    - Key features include:
        - Community Overview Page – aggregated water consumption of multiple users.
        - Reports Page – generate and export usage reports in PDF/CSV formats.
        - Notification Panel – send announcements to groups of users.
        - Admin Panel – add/delete users, manage permissions, configure integrations.

Both mobile and web interfaces must ensure multilingual support (at least English and regional languages, e.g., Hindi, Marathi) and provide dark/light themes for usability.

3.1.2 Hardware interfaces

- **Mobile Devices:** Android (v8.0+) and iOS (v13+) smartphones.
- **Web Browsers:** Chrome, Firefox, Safari, and Edge (latest two versions).

- **Optional Future Hardware:** IoT-based smart meters, water flow sensors, and leak detectors. The WMS should allow easy plug-in of such devices through APIs.

3.1.3 Software interfaces

- **Operating Systems:** Android, iOS, Windows, macOS, and Linux.
- **Database:** MySQL/PostgreSQL for storing user data, reminders, and analytics.
- **Backend Framework:** Python (Flask/Django) or Node.js.
- **Frontend:** React.js for web, Flutter/React Native for mobile.
- **External APIs:**
    - Weather API (for rainfall and irrigation prediction).
    - Push Notification Services (Firebase Cloud Messaging or Twilio).
- **Authentication Services:** OAuth 2.0 for secure login and password recovery.

3.1.4 Communications interfaces

The WMS will use HTTPS protocol for secure communication between client apps and the server.

RESTful APIs will connect the mobile/web apps to the backend.

Notifications will be pushed via Firebase Cloud Messaging (FCM) for mobile apps and email/SMS gateways for web users.

Admins and managers will use secure role-based authentication to access higher-level data.

3.2 Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.

3.2.1 User Class 1 - The User

The End User represents an individual (household member, farmer, or citizen) who primarily uses the mobile application to monitor and manage personal or household water consumption. The system must provide a simple yet powerful interface to help users understand and optimize their water usage.

The following functional requirements are organized into logical groups. Each requirement includes an identifier, description, rationale, and priority (High/Medium/Low).

A. Account Management

- FR1.1 User Registration
    - Description: The system must allow new users to register using email, phone number, or social media accounts.
    - Rationale: Ensures user identity and enables personalized data tracking.
    - Priority: High.
- FR1.2 Secure Login
    - Description: Users must log in using a password or OTP. Forgotten passwords must be reset via email/SMS.
    - Rationale: Protects user accounts from unauthorized access.
    - Priority: High.
- FR1.3 Profile Management

    - Description: Users can update profile details such as name, age, household size, and daily water target.
    - Rationale: Personalization improves reminder accuracy and analytics.
    - Priority: Medium.

B. Logging Water Usage

- FR1.4 Manual Logging
    - Description: Users must be able to log daily water consumption manually in liters (e.g., drinking: 2L, gardening: 50L).
    - Rationale: Helps track usage and identify patterns.
    - Priority: High.
- FR1.5 Usage Categories
    - Description: The system must allow categorization (drinking, cooking, cleaning, gardening, irrigation, others).
    - Rationale: Categorization supports detailed analytics and conservation advice.
    - Priority: High.
- FR1.6 Historical Records
    - Description: The system must store and display at least one year of historical water usage data.
    - Rationale: Allows users to observe long-term trends.
    - Priority: High.

C. Reminders & Notifications

- FR1.7 Reminder Setup
    - Description: Users can configure reminders for specific water activities (e.g., ―irrigate garden at 6 am daily‖).
    - Rationale: Supports habit-building and prevents overuse.
    - Priority: High.
- FR1.8 Smart Alerts
    - Description: The system must send alerts when consumption exceeds user-defined thresholds.
    - Rationale: Prevents excessive usage and encourages conservation.
    - Priority: High.
- FR1.9 Missed Reminder Notifications
    - Description: If a user misses a logged reminder, the system must send a follow-up alert.
    - Rationale: Ensures accountability and accurate logging.
    - Priority: Medium.

D. Analytics & Visualization

- FR1.10 Daily/Weekly/Monthly Dashboard
    - Description: The system must generate visual charts (line, bar, pie) showing consumption.
    - Rationale: Provides feedback and increases user awareness.
    - Priority: High.
- FR1.11 Conservation Insights

    - Description: The system must provide text-based insights (e.g., ―You saved 10 liters compared to last week‖).
    - Rationale: Encourages users to adopt better practices.
    - Priority: Medium.
- FR1.12 Export Reports
    - Description: Users can export personal usage data in PDF/Excel format.
    - Rationale: Enables sharing with managers or for personal tracking.
    - Priority: Medium.

E. Personalization

- FR1.13 Goal Setting
    - Description: Users can set daily/weekly/monthly water usage goals.
    - Rationale: Goal tracking motivates responsible water use.
    - Priority: High.
- FR1.14 Language Options
    - Description: The app must support multilingual features (English, Hindi, Marathi).
    - Rationale: Enhances accessibility in regional communities.
    - Priority: Medium.
- FR1.15 Theme Preferences
    - Description: Users can choose between light/dark themes.
    - Rationale: Improves usability and accessibility.
    - Priority: Low.

F. Advanced Features (Future Release)

- FR1.16 Weather-based Suggestions
    - Description: The system should suggest reducing irrigation on days with forecasted rainfall.
    - Rationale: Optimizes resource usage.
    - Priority: Medium (Future).
- FR1.17 IoT Integration
    - Description: The system should automatically log water usage from smart meters (if available).
    - Rationale: Reduces manual effort and increases accuracy.
    - Priority: Low (Future)

3.2.2 User Class 2 – Water Resource Manager

The Water Resource Manager represents an individual or authority responsible for monitoring and managing water consumption at the community, industrial, or municipal level. This user interacts mainly with the web-based dashboard to oversee multiple users, detect anomalies, generate reports, and send notifications.

The requirements for this user class are more advanced and data-driven than for end-users. Managers need powerful analytical tools and the ability to influence community water behavior.

A. Account and Access Control

- FR2.1 Manager Account Creation

    - Description: The system must allow administrators to create accounts for verified managers.
    - Rationale: Ensures that only authorized personnel oversee community data.
    - Priority: High.
- FR2.2 Secure Login with Role-Based Access
    - Description: Managers must log in using strong authentication (username/password + OTP).
    - Rationale: Protects sensitive community-wide data.
    - Priority: High.
- FR2.3 Role Assignment
    - Description: Managers should only access the communities assigned to them by the administrator.
    - Rationale: Prevents unauthorized access to unrelated communities.
    - Priority: High.

B. Community Monitoring

- FR2.4 Aggregated Usage Dashboard
    - Description: The system must provide managers with a real-time overview of community water consumption.
    - Rationale: Helps managers monitor overall demand and detect abnormal usage.
    - Priority: High.
- FR2.5 Household/Unit Comparison
    - Description: Managers must be able to compare water usage across different households, departments, or blocks.
    - Rationale: Identifies outliers and promotes fair usage distribution.
    - Priority: High.
- FR2.6 Anomaly Detection
    - Description: The system must highlight sudden spikes or drops in usage (e.g., 200% increase in a day).
    - Rationale: Alerts managers to leaks, theft, or misuse.
    - Priority: High.
- FR2.7 Resource Allocation View
    - Description: The dashboard must display allocated vs. consumed water for each community segment.
    - Rationale: Ensures supply-demand balance.
    - Priority: Medium.

C. Reporting & Analytics

- FR2.8 Report Generation
    - Description: Managers must be able to generate weekly, monthly, and yearly water usage reports.
    - Rationale: Supports decision-making and long-term planning.
    - Priority: High.
- FR2.9 Export Options
    - Description: Reports must be exportable to PDF, Excel, and CSV formats.
    - Rationale: Enables offline analysis and sharing with stakeholders.
    - Priority: Medium.
- FR2.10 Comparative Analysis

    - Description: The system must allow managers to compare current consumption with historical trends.
    - Rationale: Identifies long-term patterns and evaluates conservation programs.
    - Priority: High.
- FR2.11 Predictive Forecasting (Future Release)
    - Description: The system should use historical + weather data to forecast community water demand.
    - Rationale: Helps prepare for peak demands.
    - Priority: Medium (Future).

D. Communication & Notifications

- FR2.12 Bulk Notifications
    - Description: Managers must be able to send community-wide announcements (e.g., ―Water outage on 12th March, 10 AM–4 PM‖).
    - Rationale: Keeps users informed about service interruptions.
    - Priority: High.
- FR2.13 Targeted Notifications
    - Description: Managers must be able to send alerts to specific households exceeding average consumption.
    - Rationale: Promotes accountability and fairness.
    - Priority: Medium.
- FR2.14 Advisory Messages
    - Description: Managers can send water conservation tips to all users (e.g., ―Reuse kitchen water for gardening‖).
    - Rationale: Encourages sustainable practices.
    - Priority: Medium.

E. Community Goal Setting

- FR2.15 Set Community Targets
    - Description: Managers can set monthly community-wide water usage targets.
    - Rationale: Encourages collective responsibility.
    - Priority: Medium.
- FR2.16 Reward/Warning System
    - Description: The system should allow managers to assign ―green scores‖ to households that conserve water and warnings to those who overuse.
    - Rationale: Adds gamification to promote conservation.
    - Priority: Low (Future).

F. Advanced Features (Future Release)

- FR2.17 Leak Detection Integration
    - Description: If IoT sensors are integrated, managers must receive alerts of leaks in pipelines or tanks.
    - Rationale: Reduces water wastage and cost.
    - Priority: Medium (Future).

3.2.3 User Class 3 - Administrator

The Administrator is the highest-level user in the Water Management System (WMS). Administrators are responsible for system configuration, maintenance, user management, role assignment, and ensuring data integrity and security. Unlike End Users and Water Managers, Administrators do not focus on water consumption data directly but ensure that the system functions reliably for all stakeholders.

The Administrator interacts mainly with the web-based control panel but may also have access to backend monitoring tools. The requirements below cover all necessary functionalities of the Administrator role.

A. User and Role Management

- FR3.1 User Account Management
  - Description: Administrators must be able to add, update, and remove accounts of End Users and Managers.
  - Rationale: Ensures smooth onboarding and removal of inactive or fraudulent accounts.
  - Priority: High.
- FR3.2 Role Assignment
  - Description: Administrators must assign roles (End User, Manager, Admin) with different levels of privileges.
  - Rationale: Supports access control and system security.
  - Priority: High.
- FR3.3 Role Modification
  - Description: Administrators must be able to promote or demote users (e.g., from User to Manager).
  - Rationale: Reflects changes in organizational responsibilities.
  - Priority: Medium.
- FR3.4 User Suspension and Blocking
  - Description: Administrators must be able to suspend accounts temporarily or permanently block users violating system policies.
  - Rationale: Protects system integrity and compliance with regulations.
  - Priority: High.

B. System Monitoring

- FR3.5 Dashboard for System Health
  - Description: Administrators must have access to a system health dashboard showing server uptime, database performance, and error logs.
  - Rationale: Helps ensure reliability and early detection of failures.
  - Priority: High.
- FR3.6 Data Backup and Recovery
  - Description: The system must allow administrators to perform scheduled backups and restore data in case of system failures.
  - Rationale: Ensures data security and continuity.
  - Priority: High.
- FR3.7 Error and Event Logging
  - Description: The system must record all critical events (logins, failed attempts, data modifications).

    - Rationale: Provides traceability and aids in auditing.
    - Priority: High.

C. Configuration and Policies

- FR3.8 Community Threshold Management
    - Description: Administrators must configure global or community-level water usage thresholds.
    - Rationale: Standardizes conservation targets across multiple users.
    - Priority: Medium.
- FR3.9 Notification Policies
    - Description: Administrators must define global notification rules (e.g., ―send alerts if daily usage > 500 liters per household‖).
    - Rationale: Ensures consistent alerts and avoids spam.
    - Priority: High.
- FR3.10 Integration with External Services
    - Description: Administrators must be able to configure external APIs (weather data, IoT devices, billing systems).
    - Rationale: Provides flexibility and future expansion.
    - Priority: Medium.

D. Security and Compliance

- FR3.11 Role-Based Access Control
    - Description: Administrators must enforce RBAC to ensure that Users, Managers, and Admins can only access their respective functions.
    - Rationale: Prevents unauthorized data access.
    - Priority: High.
- FR3.12 Data Encryption
    - Description: Administrators must ensure all sensitive data is encrypted at rest and in transit.
    - Rationale: Protects personal and community water usage data.
    - Priority: High.
- FR3.13 Audit Trails
    - Description: Administrators must have access to system-wide audit logs for compliance and investigations.
    - Rationale: Ensures accountability.
    - Priority: Medium.

E. Advanced Administration Features (Future Release)

- FR3.14 Automated Scaling Support
    - Description: Administrators should configure auto-scaling of servers during peak usage periods.
    - Rationale: Maintains performance under heavy load.
    - Priority: Low (Future).
- FR3.15 AI-based Anomaly Detection
    - Description: The system should notify administrators of unusual system activities (e.g., DDoS attack patterns, suspicious data manipulations).
    - Rationale: Adds proactive protection against threats.

    - Priority: Medium (Future).
- FR3.16 Integration with Government Water Data
    - Description: Administrators should be able to integrate WMS with official government water monitoring portals.
    - Rationale: Expands system utility for large-scale deployments.
    - Priority: Low (Future).

3.3 Non-Functional Requirements

While functional requirements define what the Water Management System (WMS) must do, non-functional requirements describe the quality attributes the system must exhibit to ensure it is usable, reliable, and scalable in real-world conditions. These requirements apply to the entire system and cut across all user classes.

Portability

The WMS must be portable across devices and platforms. The mobile app should work on both Android and iOS operating systems, while the web application should function properly on major browsers such as Chrome, Firefox, Edge, and Safari. The backend should be deployable on multiple cloud platforms such as Google Cloud, AWS, or Azure to provide hosting flexibility.

Interoperability

The system must integrate seamlessly with external services. This includes APIs for weather data, which will enable intelligent reminders, and IoT sensors that can provide automated water usage data. The WMS must use open standards to ensure compatibility with future technologies and devices.

Capacity

A key requirement for the WMS is the ability to store and process large amounts of historical data. Each user should be able to access at least one year's worth of their usage history, and community-level data should be stored and maintained for long-term planning. The system must also support archiving older records in such a way that they can still be retrieved quickly when needed. The analytics module should be capable of analyzing large datasets from multiple users and communities without performance issues.

Reliability

Reliability is central to building user trust in the WMS. The system should provide very high availability so that users and managers can rely on it every day. Notifications and reminders must be consistently delivered, even under heavy system load. In case of a server crash or technical failure, the system must recover quickly with minimal data loss. Administrators should be notified automatically of critical errors so that corrective action can be taken immediately.

Scalability

The WMS must be scalable to support both small and large-scale deployments. Initially, it will serve individual users and local communities, but it should be possible to extend it for use in entire cities or municipalities. The system's architecture should support horizontal scaling, allowing new servers and

resources to be added as the number of users increases. Similarly, the database design should be flexible enough to support sharding or replication to distribute load across multiple storage nodes.

Concurrency

The system must support multiple users performing tasks simultaneously. For example, many users may be logging water usage at the same time, while managers are generating reports and administrators are updating policies. The system should be able to handle these concurrent operations without creating bottlenecks or delaying responses for other users. User data consistency must be guaranteed so that no records are lost or overwritten due to simultaneous actions.

Efficiency

The mobile application must be lightweight and optimized to run smoothly on mid-range smartphones without excessive battery drain or memory consumption. The web application should be efficient in its use of system resources, ensuring that even under peak usage the servers remain stable. Database queries should be optimized to execute rapidly, and background tasks such as sending notifications or syncing logs should be processed without affecting the overall responsiveness of the system.

Regulatory and Compliance Requirements

Since water usage data can be considered sensitive, the WMS must comply with applicable regulations such as the General Data Protection Regulation (GDPR) or corresponding local privacy laws. Users must be informed about data collection practices, and explicit consent should be obtained. Data retention policies must also comply with government or municipal guidelines.

3.4 Design constraints

This section includes the design constraints on the software caused by the hardware. The Water Management System (WMS) is intended to be a scalable, reliable, and user-friendly application. However, its design and implementation must operate within several **constraints**. These constraints arise from factors such as technology limitations, compliance with standards, resource restrictions, and user expectations. Understanding these limitations is crucial for both the developers and stakeholders, as they shape the boundaries within which the system can be built.

Technological Constraints

The WMS must be compatible with commonly used platforms. The mobile application is required to function on both Android and iOS operating systems, which means that cross-platform frameworks like Flutter or React Native should be considered. Similarly, the web portal must work on all major browsers including Chrome, Firefox, Safari, and Edge. This cross-platform requirement restricts the use of niche technologies that may only support a limited ecosystem. The database technology selected must support scalability and reliability while also being cost-effective. Open-source relational database systems such as MySQL or PostgreSQL are preferable to avoid expensive licensing. The backend must rely on frameworks like Python Flask/Django or Node.js, as these are widely supported, lightweight, and scalable.

Legal and Compliance Constraints

Water usage data is considered sensitive, especially at the community level where it may reveal consumption patterns. The system must therefore comply with data protection regulations such as the

General Data Protection Regulation (GDPR) and any relevant local privacy laws. All personal information must be securely stored and encrypted, and user consent must be obtained before data collection. These requirements constrain the choice of storage methods, logging mechanisms, and even third-party services that may not meet compliance standards.

Resource Constraints

Since the WMS is envisioned as a project for wide adoption, it should remain cost-efficient. Open-source tools and frameworks must be prioritized to avoid licensing costs. The system must also be lightweight so that it can run smoothly on mid-range smartphones with limited memory and storage. This rules out the use of heavy software libraries or resource-intensive designs.

Network bandwidth may also be limited in some areas where the application is used. Hence, the system must be designed to function under moderate connectivity and provide at least basic offline features such as storing unsynced logs locally until a connection is available.

User Interface Constraints

The system must accommodate users with varying levels of digital literacy. End users are expected to include individuals with little to no technical background, while managers and administrators may have more advanced knowledge. Therefore, the interface must remain simple and intuitive for end users while still providing detailed analytical tools for managers and administrators. This design constraint rules out overly complex navigation structures and forces the development team to balance simplicity with functionality.

Additionally, accessibility standards must be followed. Users with visual impairments should be able to interact with the application through readable fonts, high-contrast themes, and screen-reader compatibility. Multilingual support is another constraint, as the system must be usable in multiple regional languages such as English, Hindi, and Marathi.

Performance and Security Constraints

The system must meet strict performance targets, including fast response times and the ability to handle concurrent users. This means that the architecture must be optimized for efficiency and scalability, without unnecessary complexity. Security is another major constraint. All communications must be secured using HTTPS, and sensitive data such as user credentials must be encrypted. Role-based access control must be enforced to ensure that users, managers, and administrators only access features relevant to their roles.

Future Expansion Constraints

While the WMS must initially support manual logging and reminders, it must also be designed in a modular way that allows easy integration with Internet of Things (IoT) devices such as smart water meters and leak detectors in the future. This constraint affects the choice of architecture, as the system must remain flexible enough to incorporate new technologies without requiring a complete redesign.

3.5 Software system attributes

The success of the Water Management System (WMS) does not depend solely on its ability to provide functional features such as water logging, reminders, and reporting. Equally important are the **non-functional attributes** that determine how well the system performs in real-world conditions. These attributes ensure that the system is reliable, usable, secure, and maintainable across diverse user groups and environments.

Reliability

The WMS must be a highly reliable platform, since users will depend on it to manage daily and community-level water usage. The system must guarantee that reminders are delivered on time, data is recorded accurately, and historical information is preserved without corruption. Users should be able to rely on the system for both short-term notifications and long-term analysis. Reliability also extends to data synchronization between the mobile app and the central server. If a user enters data offline, the system must reliably update the server once connectivity is restored without causing duplication or data loss.

Availability

The system must be designed to remain available to users at all times, except during planned maintenance windows. This is particularly important for community-level managers and administrators who may need to access real-time water usage reports during working hours. The WMS should be hosted on a cloud infrastructure to ensure redundancy and minimize downtime. Availability also applies to mobile notifications — even when the app is not actively in use, reminders and alerts must still reach users without fail.

Security

Security is a critical attribute given that the WMS stores sensitive personal and community water usage data. All data transfers between the mobile app, web portal, and server must be encrypted using secure protocols such as HTTPS. User authentication must be protected with secure login methods, including the use of multi-factor authentication for managers and administrators. Data at rest in the database must also be encrypted, ensuring that unauthorized access to the server does not expose sensitive information. Role-based access control must strictly enforce the boundaries between End Users, Water Managers, and Administrators, preventing accidental or malicious misuse of the system.

Maintainability

The system must be designed with maintainability in mind so that it can evolve with changing requirements. Code should follow modular and well-documented practices, making it easier for developers to fix bugs, introduce new features, or update existing ones without disrupting the system. The WMS should also support automated testing and logging, allowing administrators and developers to quickly identify and resolve issues. In addition, configuration files and system parameters should be easy to update without requiring extensive code changes, so that system policies can be adapted to new contexts.

Usability

The usability of the WMS is essential, since the system targets a wide audience with varying levels of technical expertise. End Users should find the mobile application intuitive, with simple navigation, clear icons, and a user-friendly design. Community Managers, on the other hand, require a dashboard with advanced analytical features, but it must still be designed in a way that avoids overwhelming them with unnecessary complexity. Accessibility standards must be respected, ensuring that users with visual impairments can rely on features such as large fonts, high-contrast themes, and compatibility with screen readers. Furthermore, the system must support multilingual features so that users in different regions can interact with the application in their preferred language.

Portability

The WMS must be portable across different environments and devices. The mobile application must support both Android and iOS devices, while the web application must function on all major browsers. The backend must be designed to be deployable on multiple cloud platforms such as Google Cloud, AWS, or Azure, ensuring flexibility in hosting. Portability also extends to future IoT integration, where the system must be able to accept data from various brands and models of smart meters and sensors without requiring major redesign.

Efficiency

The system must make efficient use of resources such as memory, storage, and processing power. The mobile app must be optimized to minimize battery usage and avoid consuming excessive storage on the device. The server must handle requests and process reports efficiently, even when serving thousands of users simultaneously. Database queries should be optimized to ensure that the system does not slow down as the dataset grows larger. Efficiency also applies to the notification service, which must deliver alerts quickly without excessive retries or delays.

Scalability

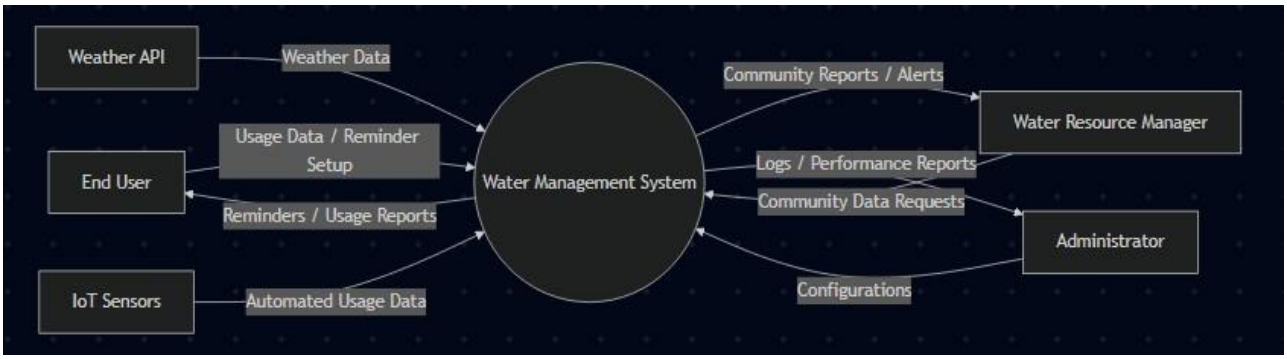
Although scalability is closely tied to performance, it deserves attention as a system attribute. The WMS must be scalable enough to serve a small household, a large residential community, or even an entire city. The architecture must support adding new servers, expanding the database, and integrating additional modules without requiring major changes. The system must also allow the addition of new user roles in the future, such as government regulators or environmental agencies, without disrupting existing functionality.

Interoperability

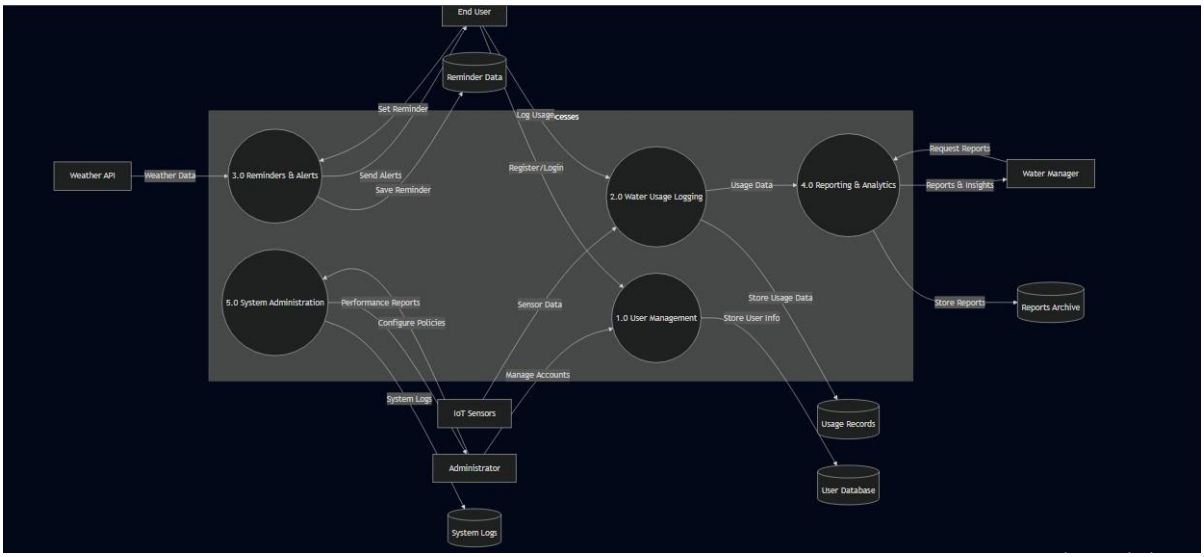
The WMS must be capable of integrating with external systems and devices. This includes APIs for weather forecasts, municipal water supply notifications, and IoT-based smart meters. The system should be designed using open standards and protocols to ensure smooth interoperability with current and future technologies.

4. Analysis Models

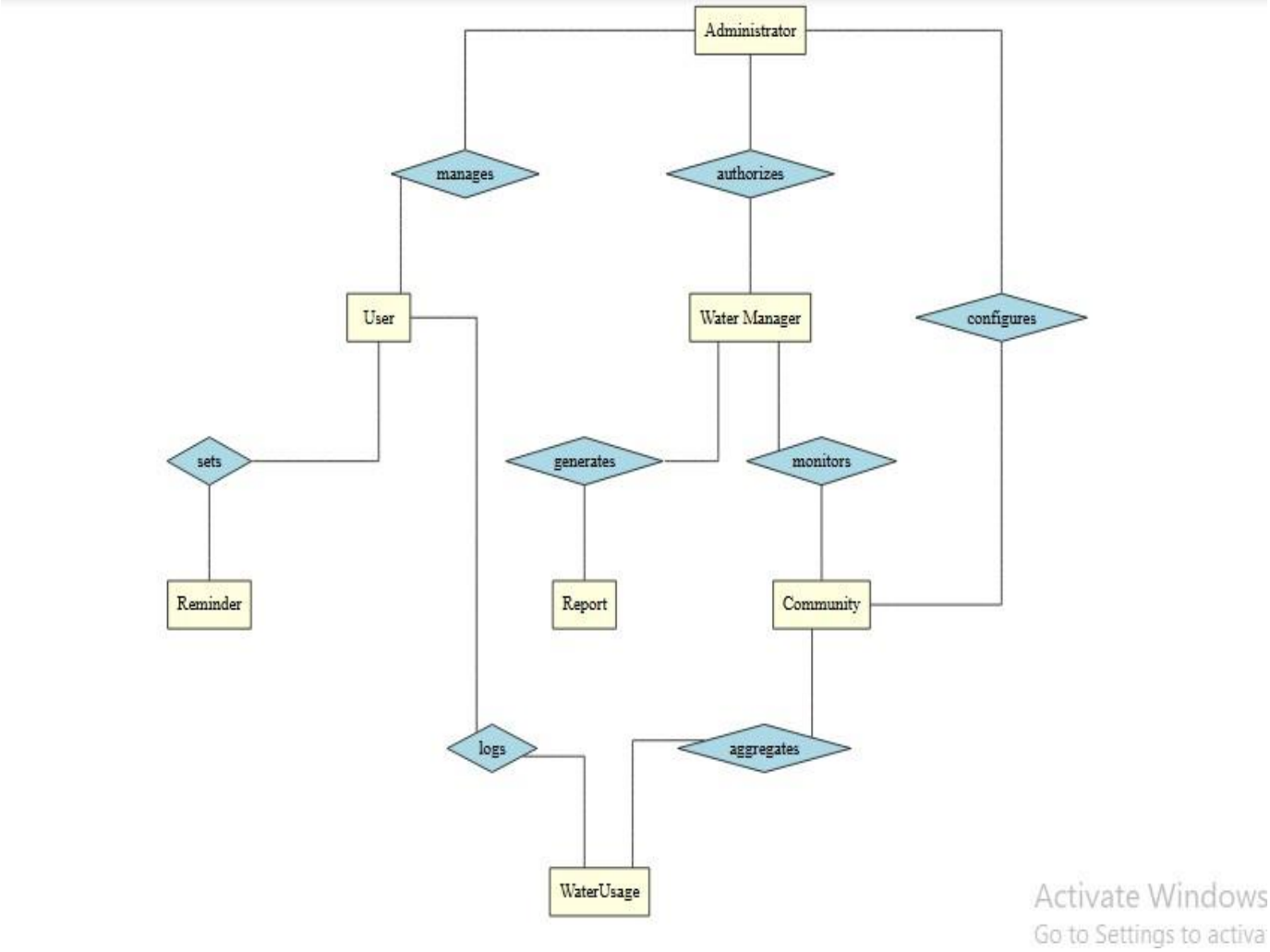
Context Diagram:



Data Flow Diagram(DFD):



E-R Diagram:



5. Prioritization and Release Plan

The Water Management System (WMS) contains a large set of functional and non-functional requirements that must be developed, tested, and delivered in a structured manner. Since it is not feasible to implement all requirements at once, especially in the initial release, a **prioritization strategy** is necessary. This ensures that the most critical features are developed first, while secondary features are scheduled for later iterations.

The following sections explain the prioritization method chosen, the results of prioritization, and the release plan for the system.

5.1 Choice of prioritization method

When prioritizing the requirements the ten most important ones were picked out first. This was done with a simple ―1 to 10‖ ranking method, with one being ―not important‖ and ten ―very important‖. Based on the elicitation meetings, and the perceived ideas of what was important to the different stakeholders, a number was set for each requirement. The numbers were then summed up for each requirement and the ten with the highest score were chosen to be prioritized with the cost value approach. The results, which are red-marked, can be seen in Appendix I and as shown, it turned out to be five functional requirements and five quality requirements. These requirements were then prioritized according to the cost value approach and the results can be viewed under Appendix II.

The remaining requirements were prioritized according to the ―Five-Way Priority Scheme‖ as shown in Appendix III. This method was chosen since it gives the different stakeholders the same importance and has an enough wide range for determining which requirement is more important than the other [3].
However, in this prioritization process, the development team was not included as a stakeholder since the different features were not considered to be as important to them as for the other stakeholders.

Other methods for prioritization, such as the hundred-dollar test and the yes-no vote, were also considered. The hundred-dollar test is quite similar to the five-way priority scheme, since it also gives a wide range for ranking the requirements. However, it is more easily misused since someone could save all their money and put them on a requirement that they think is very important [3]. Others might not agree that this requirement is important but it might still get the most votes since one person cared about it [3].

The yes-no vote method might be fairly simple to carry out, however the range is too narrow. For instance, if two requirements are not very important it would be hard to determine which of those requirements that is more important than the other [3].

In conclusion, weighing the disadvantages and advantages of these methods against each other lead us to choose the five-way priority scheme.

5.2 Prioritization Result

The results of applying the Cost-Value method indicate that the following requirements are of highest priority for the initial release:

- High Priority (Release 1)
  - User registration and authentication
  - Logging daily water usage
  - Setting and receiving reminders
  - Generating personal usage reports
  - Community data collection and aggregation
  - Basic administrator functions such as user management

These requirements provide the core functionality of the system and ensure that users can begin using the WMS effectively from the very first release.

- Medium Priority (Release 2)
    - Advanced reporting and analytics (comparative trends, graphs)
    - Integration with weather data for intelligent reminders
    - Manager dashboards with real-time visualization
    - Extended notification channels (email, push notifications)
    - Multilingual support

These features improve user experience and expand the system's usefulness but are not essential for initial deployment.

- Low Priority (Release 3 and beyond)
    - IoT integration with smart water meters
    - AI-based consumption prediction and forecasting
    - Advanced gamification features for user motivation
    - Large-scale municipal integration features

These are considered long-term goals and will be introduced once the system has matured and gained a stable user base.

5.3 Release Plan

Based on the prioritization results, the WMS will be released in phased iterations.

- Release 1 (Core System – 6 months)
    - End User: Registration, authentication, daily usage logging, reminders, personal reports
    - Manager: Basic community-level reporting
    - Administrator: User management and role assignment

This release establishes the foundation of the system and delivers immediate value to users.

- Release 2 (Enhanced Features – 6–12 months)
    - End User: Multilingual interface, weather-based reminders, advanced personal analytics
    - Manager: Detailed reporting with charts and visualization, comparative analysis
    - Administrator: Extended configuration and monitoring tools

This release enhances usability and provides advanced features for analysis and planning.

The prioritization and release plan ensure that the development of the Water Management System remains focused, resource-efficient, and user-centric. The chosen Cost-Value approach provides a balanced framework for decision-making, while the phased release plan ensures that stakeholders receive value at every stage of the project. This strategy also allows for continuous user feedback, enabling the system to evolve and expand over time in response to real-world needs.