Deliverable #1 : Software Requirement Specification (SRS)

SE 3A04: Software Design III – Large System Design

Tutorial Number: T01
Group Number: G01
Group Members:

- Benjamin Bloomfield
- Vikram Chandar
- Trisha Panchal
- Yug Vashisth
- Hamza Nadeem

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
  - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.

1 Introduction

This Software Requirements Specification (SRS) defines the requirements for the Smart City Environmental Monitoring & Alert System (SCEMAS), a cloud-based platform for smart-city environmental monitoring and alerting. SCEMAS collects real-time environmental telemetry from a distributed sensor network, ingests it through an MQTT-based interface, validates incoming messages, and stores the data for both real-time and historical analysis. The platform aggregates readings, evaluates configurable threshold rules, and generates alerts to support timely city response. SCEMAS provides a secure operator dashboard for visualization and exposes a read-only REST API that returns aggregated, non-sensitive environmental data for public and third-party consumption. This document establishes the system scope, key stakeholders and users, core capabilities, and the constraints and quality attributes that guide implementation and verification.

1.1 Purpose

This document is used to establish the requirements, performance, and intended behaviour/uses of the Smart City Environmental Monitoring and Alert System (SCEMAS). The intended audience of the SRS is the stakeholders of SCEMAS, including developers and testers.

1.2 Scope

The Smart City Environmental Monitoring & Alert System (SCEMAS) is a cloud-based software product designed to collect, process, store, and distribute environmental telemetry data for urban monitoring and decision support. The system takes real-time data from a distributed network of environmental sensors, including air quality, temperature, humidity, and other environmental indicators. SCEMAS validates, aggregates, and persistently stores this telemetry in a secure database that supports both real-time and historical analysis. The software detects threshold violations and adverse environmental conditions and provides automated alerting functionalities for city operators. A secure dashboard allows authorized city personnel to visualize sensor data on maps and charts, monitor system health, and manage active alerts. Access to system functionality is controlled through a role-based access control framework to ensure that administrative and operational privileges are properly restricted. All telemetry and user data transmitted within the system is encrypted using industry-standard protocols, and stored data is protected through appropriate encryption mechanisms to maintain confidentiality and integrity. The scope of this project focuses on the architectural design and core software services of the system, including telemetry ingestion pipelines, secure data storage, alert evaluation logic, and the protected user interface. Physical sensor hardware, large-scale deployment, and advanced predictive analytics are outside the scope of the initial implementation.

1.3 Definitions, Acronyms, and Abbreviations

- MQTT: Message Queuing Telemetry Transport
- SCEMAS: Smart City Environmental Monitoring and Alert System
- CESI: Canadian Environmental Sustainability Indicators
- RBAC: Role based Access Control

1.4 References

[1] Government of Canada, "Canadian Environmental Sustainability Indicators (CESI)," *Government of Canada*. [Online]. Available: https://indicators-map.canada.ca/

[2] D. Murphy, "FAQ Best Practices: Benefits of a Well-Structured Website FAQ," *Masterful Marketing*, May 10, 2025. [Online]. Available: https://masterful-marketing.com/website-faq-benefits-best-practices/

[3] K. Moran, "Dealing with Technical or Professional Jargon," *Nielsen Norman Group*, March 19, 2023. [Online]. Available: https://www.nngroup.com/articles/technical-jargon/

[4] "What is Text-To-Speech (TTS) and why do we need it?" *Voxpow*, August 16, 2020. [Online]. Available: https://voxpow.com/blog/what-is-text-to-speech/

[5] D. Nichols, "Coloring for Colorblindness," *David Math Logic*. [Online]. Available: https://davidmathlogic.com/colorblind/

[6] "How fast should a Website Load in 2025?" *BrowserStack*, July 1, 2025. [Online]. Available: https://www.browserstack.com/guide/how-fast-should-a-website-load

[7] B. Farouk, "What is a Good API Response Time?" *Odown*, July 5, 2024. [Online]. Available: https://odown.com/blog/what-is-a-good-api-response-time/

[8] "Environmental Monitoring Precision," *Sustainability Directory*, Jan 12, 2025. [Online]. Available: https://pollution.sustainability-directory.com/term/environmental-monitoring-precision/

[9] "False Positive Rate," *vwo*. [Online]. Available: https://vwo.com/glossary/false-positive-rate/

[10] N. English, C. Zhao, K. L. Brown, C. Catlett, and K. Cagney, "Making Sense of Sensor Data: How Local Environmental Conditions Add Value to Social Science Research," *National Library of Medicine*, May 5, 2020. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8991303/

[11] U. C. Gorai, "The SOLID Approach: Principles for Maintainable and Scalable Code," *Medium*, December 20, 2024. [Online]. Available: https://medium.com/@ucgorai/the-solid-approach-principles-for-maintainable-and-scalable-code-1208473e1bf2

[12] Government of Canada, "User authentication guidance for information technology systems (ITSP.30.031 v3)," *Canadian Centre for Cyber Security*, April, 2018. [Online]. Available: https://www.cyber.gc.ca/en/guidance/user-authentication-guidance-information-technology-systems-itsp30031-v3

[13] Government of canada, "Secure your accounts and devices with multi-factor authentication (ITSAP.30.030)," *Canadian Centre for Cyber Security*, February, 2024. [Online]. Available: https://www.cyber.gc.ca/en/guidance/secure-your-accounts-and-devices-multi-factor-authentication-itsap30030

[14] Government of Canada, "Internet of Things (IoT) Security - ITSAP.00.012," *Canadian Centre for Cyber Security*, July, 2022. [Online]. Available: https://www.cyber.gc.ca/en/guidance/internet-things-iot-security-itsap00012

[15] Government of canada, "Using encryption to keep your sensitive data secure - ITSAP.40.016," *Canadian Centre for Cyber Security*, May, 2021. [Online]. Available: https://www.cyber.gc.ca/en/guidance/using-encryption-keep-your-sensitive-data-secure-itsap40016

[16] A. Ali, M. Ahmed, and A. Khan, "Audit Logs Management and Security - A Survey," *ResearchGate*, June, 2021. [Online]. Available: https://www.researchgate.net/publication/352745109

[17] Government of Canada, "Guidance on securely configuring network protocols (ITSP.40.062)," *Canadian Centre for Cyber Security*, January, 2025. [Online]. Available: https://www.cyber.gc.ca/en/guidance/guidance-securely-configuring-network-protocols-itsp40062

[18] "The Personal Information Protection and Electronic Documents Act (PIPEDA)," *Office of the Privacy Commissioner of Canada*, September 9, 2025. [Online]. Available: https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/

[19] "Audit and Traceability / Audit Logs," *Secoda*, June 23, 2025. [Online]. Available: https://www.secoda.co/glossary/audit-traceability-audit-logs

[20] M. Dreshner, "Security log retention: Best practices and compliance guide," *AuditBoard*, June 17, 2025. [Online]. Available: https://auditboard.com/blog/security-log-retention-best-practices-guide

[21] Government of Canada, "Cyber Security Readiness Goals: Securing Our Most Critical Systems," *Canadian Centre for Cyber Security*, October 29, 2024. [Online]. Available: https://www.cyber.gc.ca/en/cyber-security-readiness/cyber-security-readiness-goals-securing-our-most-critical-systems

[22] Government of canada, "Canadian Environmental Protection Act: monitoring," *Environment and natural resources*, June 7, 2018. [Online]. Available: https://www.canada.ca/en/environment-climate-change/services/canadian-environmental-protection-act-registry/monitoring-reporting-research/monitoring.html

[23] Government of Canada, "Directive on Open Government," *How Government Works*, OCtober 9, 2014. [Online]. Available: https://www.tbs-sct.canada.ca/pol/doc-eng.aspx?id=28108

1.5 Overview

The remainder of this Software Requirements Specification (SRS) is organized as follows. Section 2 provides an overall description of SCEMAS, including its context, major functions, intended users, constraints, and assumptions. Section 3 presents the system's use case diagram. Section 4 outlines the functional requirements using the main business events and viewpoint-based scenarios, ending with a consolidated global scenario. Section 5 describes the non-functional requirements, including usability, performance, operational/environmental constraints, maintainability/support, security, cultural/political, and legal considerations. Finally, the Appendix includes the division of labor and any supporting material required by the course template.

2 Overall Product Description

2.1 Product Perspective

SCEMAS is a cloud-native IoT software platform built for smart-city environmental monitoring and alerting. It is similar in purpose to Government of Canada environmental dashboards such as the Canadian Environmental Sustainability Indicators (CESI) [1] interactive indicators and maps, as well as national monitoring dashboards that report trends and exceedances (for example, water monitoring and wastewater surveillance). However, SCEMAS is designed for city operations where near real-time telemetry ingestion, automated threshold-based alerting, and zone-level aggregation are needed to support fast response and day-to-day decision-making. In many cities, environmental data is tracked in a scattered or manual way, which can slow down responses to issues like poor air quality, heatwaves, or noise disturbances. SCEMAS addresses this by bringing everything into one place: it collects telemetry in real time, automatically detects when conditions cross important thresholds, and shares clear, timely information with both city operators and the public.

At a high level, SCEMAS sits between a network of environmental sensing devices and the people or systems that need environmental insights. During development and testing, the sensors act like real devices by publishing telemetry messages into the platform. Incoming telemetry is accepted through an MQTT-based ingestion entry point, then passed through a validation and ingestion component that checks message structure and plausible value ranges before storing the data in persistent time-series storage.

Once data is stored, SCEMAS performs real-time aggregation to produce useful summaries for monitoring and reporting. A configurable, rule-based alerting engine continuously evaluates these readings and aggregates to detect threshold violations and track alert status over time. When an alert is triggered, SCEMAS can notify external subscriber systems, while also presenting the situation to operators through a secure web dashboard that includes maps, charts, current values, active alerts, system health, and historical trends.

SCEMAS also has a read-only REST API which can be provided to approved third-party organizations upon request. The API returns aggregated, non-sensitive environmental data, and can be paired with a simple demonstration client showing how to consume the endpoints. Potential consumers include mapping applications (e.g., Google Maps overlays) and external organizations that want to display SCEMAS data for their own users or services. To ensure secure administration and accountability, the platform includes

role-based access control (RBAC) and audit logging across operator-facing functions and administrative actions.

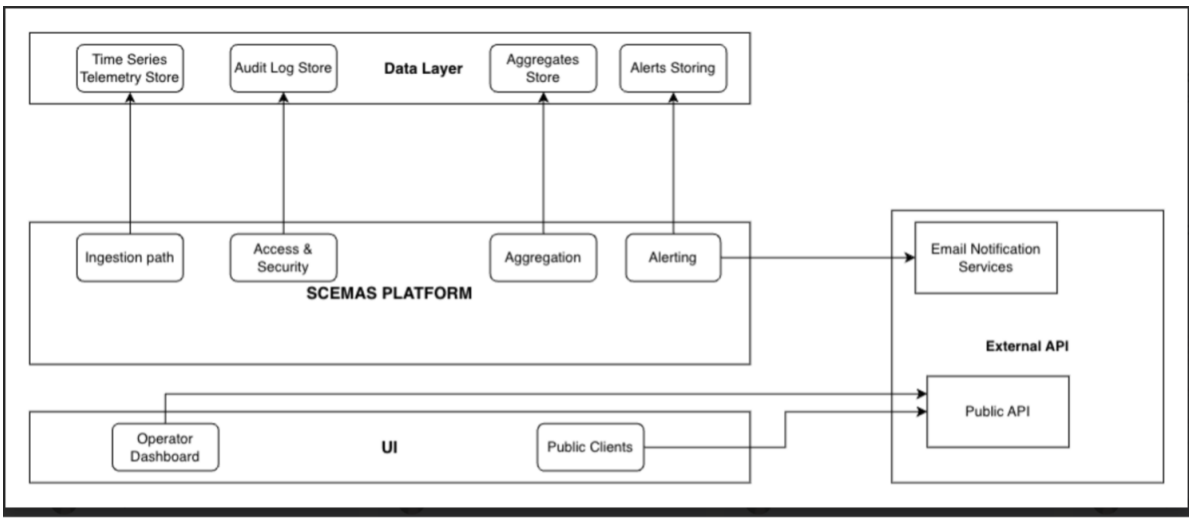


Figure 1: SCEMAS high-level block diagram and external interfaces.

2.2 Product Functions

Module	Functions
User Module	• Create an account – Allow the user to create an account. • Login – Allow the sign in to their account. • Account Permissions – Assigns each user with a role controlling their permissions. • Modify Account – Allow administrators to modify user accounts.

Module	Functions
Alert System	• Alert Rules – Allow administrators to define rules based on thresholds. • Evaluate Alert Conditions – Continuously evaluate incoming data with the defined alert rules. • Alert Triggers – Trigger an alert upon rule violation. • Alert Status – Keep a status indicator of each alert between active, acknowledged, or resolved. • Alert Log – Keep a log of all alerts that occur.
Data Handling	• Receiving data – Receives IoT data. • Message validation – Incoming messages are validated for correct format. • Data storage – All valid data is persistently stored. • Real time aggregation – Aggregates incoming data into measurements for predefined graphical zones within the city.
Private Dashboard	• Display real-time visualizations – Display any sensor locations, charts of environmental metrics, etc. • Display real-time visualizations – Provides overview of system health, active alerts, and access to historical data trends. • Display real-time visualizations – Ability to manage sensors including adding and deleting them.

Module	Functions
Public Dashboard	• Public Rest API – Exposes aggregated non confidential information for public users.

2.3 User Characteristics

2.3.1 User I: City Operators

City operators analyze the web dashboard to monitor key environmental conditions, and respond accordingly to alerts generated by the system.

- **Education Level**: City operators are expected to have some level of post-secondary education, in environmental science, engineering, urban planning, or another related field of study.
- **Experience**: Prior experience working in a similar role, such as environmental monitoring or public safety operations. Such users should be familiar with interpreting geographical maps, environmental metrics, charts, and trends to identify irregular environmental conditions.
- **Technical Expertise**: City operators are expected to have intermediate computer proficiency, such as the ability to navigate web-based applications, interact with dashboards, and operate data visualization tools.

2.3.2 User II: Public Users

The general public and citizens can access non-sensitive environmental information through public interfaces that use the read-only REST API, for awareness and safety purposes.

- **Education Level**: Public users are assumed to have basic literacy skills. Basic foundations in listening, reading, writing, and speaking are all the user needs to use the platform without any challenges.
- **Experience**: The user is expected to have little to no prior experience using monitoring systems. The system is designed to be user-friendly to ensure that interactions are straightforward for any user.
- **Technical Expertise**: Basic technical skills, such as understanding how to use a web browser, navigate a webpage, and view digital displays, are all that is necessary.

2.3.3 User III: Administrators

System administrators are responsible for building and maintaining the system. Their role focuses specifically on ensuring that the system operates securely, behaves correctly, and is maintainable over time.

- **Education Level**: System administrators should have at least a post-secondary education in software engineering, computer science, or another similar technical field.
- **Experience**: They are expected to have prior experience maintaining or managing software systems in a professional or academic setting. This experience includes responsibilities related to access control, system configuration, and operational oversight. Such users should also understand how to define system policies and supervise automated processes such as logging and alert handling.
- **Technical Expertise**: Advanced technical skills are required for system administrators. They should be familiar with configurable rule-based systems, authentication methods, role-based access control (RBAC) models, and audit logging.

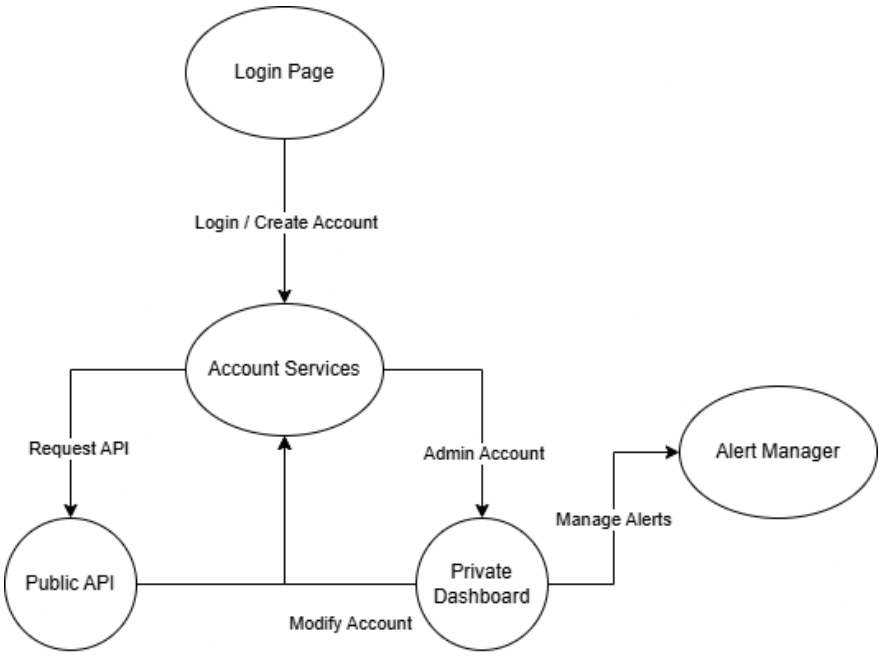


Figure 2: SCEMAS State Diagram.

2.3.4 User IV: Third-Party Developer

Third-party developers request and interpret environmental data from the public API to help build external applications.

- **Education Level**: Developers should have a post-secondary education in software engineering, computer science, or equivalent professional experience.
- **Experience**: The third-party developers are expected to have experience developing software that uses REST APIs and works with JSON data.
- **Technical Expertise**: Proficient technical skills in web application development and API integration, as well as the ability to comprehend API documentation.

2.4 Constraints

- Provide a general description of any constraints that will limit the developer's options

1. **Budget**: The project is constrained by a budget of $0. This impacts what features the developer can implement, as all technologies, frameworks, cloud services, and other development tools must be free for the team to use.
2. **Project Time**: The development of the system is constrained to the Winter 2026 academic calendar. This limits the overall scope of the system, and the amount of features that can be implemented at launch. It shapes the overall development timeline, and the ability to test and optimize.
3. **Communication Protocol**: All sensor telemetry must use the MQTT protocol. This restricts the system's communication model and requires that incoming messages align with the predefined data formats that MQTT messaging supports.
4. **Personal Identifiable Information**: The system is constrained to operate under privacy-by-design principles, and must not collect, process, or store any personally identifiable information. Environmental data must be aggregated and associated with general city zones rather than precise locations.

5. **Public API Scope**: The public and third-party facing API is constrained to be read-only, meaning that they can not submit data or modify the state. It must only provide access to aggregated, non-sensitive environmental data.
6. **Simulated Hardware**: Physical IoT sensor hardware is not available for this project and must be be generated using software simulations. This influences design decisions as the functionality cannot be verified using real-world sensors.
7. **Deployment Environment**: The system is constrained to a cloud architecture using available platforms and tools suitable for educational use. Therefore, developers cannot deploy the system on local machines or private servers.

2.5 Assumptions and Dependencies

Assumptions About What the Software Is Aiming to Achieve

- IoT sensors provide accurate environmental information to the web application.
- IoT sensors provide data in real-time.
- All telemetry messages follow the predefined MQTT message schema agreed upon by the system.
- All IoT sensors have already been installed and require no additional setup.
- No PII data will be collected or transmitted by IoT sensors

Other Assumptions That May Require Requirement Changes if They Fail

- The selected cloud provider provides sufficient uptime, scalability, and availability for development and deployment.
- The System has access to a database capable of efficiently storing and querying time-series data.

2.6 Apportioning of Requirements

- EMPTY

3 Use Case Diagram

- EMPTY

4 Highlights of Functional Requirements

Main Business Events: The business events that relate to the system include:

- BE1. Define a New Alert Rule For Incoming Hurricane.
- BE2. Update an Existing Alert Rule Due to New Pollution Limits.
- BE3. Remove or Disable an Alert Rule From the Winter Season Ending.
- BE4. System Detects Rule Violation From a Wildfire and Triggers Alert
- BE5. Confirm Alert After Detecting a Rule Violation Due to a Heat Wave.
- BE6. Mark the alert as resolved
- BE7. Request Access to Public API
- BE8. Modify Account

Viewpoints: The viewpoints the system considers include:

- VP1. City Operators
- VP2. Public Users
- VP3. Administrators
- VP4. Third Party Developers
- VP5. Environmental Monitoring Department
- VP6. IT Support

Business Events:

BE1. Define a New Alert Rule For Incoming Hurricane. #1
Pre-condition: The administrator is authenticated and logged into the system. There is no alert rule that already exists for the same threshold.

VP1. City Operators #1

N/A

VP2. Public Users #2

N/A

VP3. Administrators #3
Main Success Scenario

1. System displays the dashboard.
2. Administrator selects *Manage Alerts* tab
3. System displays the existing alert rules, and the option to create a new rule.
4. Administrator selects *Create New Alert Rule*.
5. System displays the alert rule setup form.
6. Administrator enters the new alert rule information.
7. Administrator submits the alert rule to be added to the system.
8. System validates the form.
9. System stores the new alert rule.
10. System reports to administrator that the alert has been successfully created.

Secondary Scenario

8i. System detects invalid or incomplete data.
- 8i.1 System prevents the alert rule from being submitted.
- 8i.2 System displays an error message describing the issue.
- 8i.3 System highlights the section(s) that need to be changed.
- 8i.4 Return to BE1-10.

9i. System fails to store the added alert rule.
- 9i.1 System displays an error message that the alert rule could not be created.
- 9i.2 System provides instructions for contacting IT Support.

VP4. Third-Party Developers #4

N/A

VP5. Environmental Monitoring Department #5
Main Success Scenario

1. Environmental Monitoring Department identifies a hurricane is forming nearby.

2. Environmental Monitoring Department determines that a new threshold is required to monitor wind conditions in new areas.
3. Environmental Monitoring Department submits the updated rule requirements to the administrator
4. Administrator reviews and approves the requested changes.

VP6. IT Support #6

4ii System should allow IT Support to restore administrator access after verifying their identity.

13i System should provide IT Support with error logs to resolve the alert rule storage failures.

Global Scenario:

Pre-condition: The administrator is authenticated and logged into the system. There is no alert rule that already exists for the same threshold.

Main Success Scenario

1. Environmental Monitoring Department identifies a hurricane is forming nearby.
2. Environmental Monitoring Department determines that a new threshold is required to monitor wind conditions
3. Environmental Monitoring Department submits the updated rule requirements to the administrator
4. Administrator reviews and approves the requested changes.
5. System displays the dashboard.
6. Administrator selects *Manage Alerts* tab
7. System displays the existing alert rules, and the option to create a new rule.
8. Administrator selects *Create New Alert Rule*.
9. System displays the alert rule setup form.
10. Administrator enters the new alert rule information.
11. Administrator submits the alert rule to be added to the system.
12. System validates the form.
13. System stores the new alert rule.
14. System reports to administrator that the alert has been successfully created.

Secondary Scenario

12i. System detects invalid or incomplete data.
- 12i.1 System prevents the alert rule from being submitted.
- 12i.2 System displays an error message describing the issue.
- 12i.3 System highlights the section(s) that need to be changed.
- 12i.4 Return to BE1-10.

13i. System fails to store the added alert rule.
- 13i.1 System displays an error message that the alert rule could not be created.
- 13i.2 System provides instructions for contacting IT Support.

BE2. Update an Existing Alert Rule Due to New Pollution Limits. #2

Pre-condition: There must exist at least one alert rule in the system. The administrator is authenticated and logged into the system.

VP1. City Operators #1

N/A

VP2. Public Users #2

N/A

VP3. Administrators #3

Main Success Scenario

1. System displays the dashboard.
2. Administrator selects *Manage Alerts* tab
3. System displays the existing alert rules.
4. Administrator selects an alert rule to update.
5. Administrator modifies the alert rule parameters.
6. Administrator submits the updated alert rule.
7. System validates the updated alert rule.
8. System updates the alert rule in the systems database.
9. System reports to the administrator that the alert rule has been successfully updated.

Secondary Scenario

7i. Updated alert information is invalid or incomplete.

7i.1 System prevents the updated alert rule from being submitted.

7i.2 System displays an error message describing the issue.

7i.3 System highlights the section(s) that need to be changed.

7i.4 Return to BE2-5.

8i. System fails to update the alert rule.

8i.1 System displays an error message indicating that the alert rule update failed.

8i.2 System provides instructions for contacting IT Support.

VP4. Third-Party Developers #4

N/A

VP5. Environmental Monitoring Department #5

Main Success Scenario

1. Environmental Monitoring Department identifies new pollution limits have been established.
2. Environmental Monitoring Department makes a particular threshold adjustment based on new data.
3. Environmental Monitoring Department submits the updated rule requirements to the administrator
4. Administrator reviews and approves the requested changes.

VP6. IT Support #6

8i. System provides IT Support with error logs to resolve the alert rule update failures.

Global Scenario:

Pre-condition: There must exist at least one alert rule in the system. The administrator is authenticated and logged into the system.

Main Success Scenario

1. Environmental Monitoring Department identifies an increase in rainfall in the upcoming season.
2. Environmental Monitoring Department makes a particular threshold adjustment based on environmental readings.
3. Environmental Monitoring Department submits the updated requirements to the administrator
4. Administrator reviews and approves the requested changes.
5. System displays the dashboard.

6. Administrator selects *Manage Alerts* tab
7. System displays the existing alert rules.
8. Administrator selects an alert rule to update.
9. Administrator modifies the alert rule parameters.
10. Administrator submits the updated alert rule.
11. System validates the updated alert rule.
12. System updates the alert rule in the systems database.
13. System reports to the administrator that the alert rule has been successfully updated

Secondary Scenario

11i. Updated alert information is invalid or incomplete.
- 11i.1 System prevents the updated alert rule from being submitted.
- 11i.2 System displays an error message describing the issue.
- 11i.3 System highlights the section(s) that need to be changed.
- 11i.4 Return to BE2-5.

12i. System fails to update the alert rule.
- 12i.1 System displays an error message that that the alert rule update failed.
- 12i.2 System provides instructions for contacting IT Support.

BE3. Remove or Disable an Alert Rule From the Winter Season Ending. #3
Pre-condition: There must exist at least one alert rule in the system. The administrator is authenticated and logged into the system.

VP1. City Operators #1

N/A

VP2. Public Users #2

N/A

VP3. Administrators #3

Main Success Scenario

1. System displays the dashboard.
2. Administrator selects *Manage Alerts* tab
3. System displays the existing alert rules.
4. Administrator selects an alert rule to remove or disable.
5. System prompts the administrator for confirmation.
6. Administrator confirms the action.
7. System removes or disables the alert rule.
8. System reports that the alert has been successfully removed or disabled.

Secondary Scenario

6i. Administrator cancels the confirmation.
- 6i.1 System does not remove or disable the alert rule.
- 6i.2 System returns to the list of existing alert rules.

7i. System fails to remove or disable the alert rule.
- 7i.1 System displays an error message that the action failed.
- 7i.2 System provides instructions for contacting IT Support.

VP4. Third-Party Developers #4

N/A

VP5. Environmental Monitoring Department #5

Main Success Scenario

1. Environmental Monitoring Department determines that seasonal environmental conditions (e.g., snow) have ended.
2. Environmental Monitoring Department reviews identifies a threshold that is no longer required.
3. Environmental Monitoring Department submits the no longer needed requirements to the administrator
4. Administrator reviews and approves the requested changes.

VP6. IT Support #6

7i System provides IT Support with system logs to investigate and resolve alert rule removal or disable failures.

Global Scenario:

Pre-condition: There must exist at least one alert rule in the system. The administrator is authenticated and logged into the system.

Main Success Scenario

1. Environmental Monitoring Department determines that seasonal environmental conditions (e.g., snow) have ended.
2. Environmental Monitoring Department reviews identifies a threshold that is no longer required.
3. Environmental Monitoring Department submits the no longer needed requirements to the administrator
4. Administrator reviews and approves the requested changes.
5. System displays the dashboard.
6. Administrator selects *Manage Alerts* tab
7. System displays the existing alert rules.
8. Administrator selects an alert rule to remove or disable.
9. System prompts user for confirmation.
10. Administrator confirms the action.
11. System removes or disables the alert rule.
12. System reports that the alert has been successfully removed or disabled.

Secondary Scenario

11i. Administrator cancels the confirmation.

11i.1 System does not remove or disable the alert rule.

11i.2 System returns to the list of existing alert rules.

12i. System fails to remove or disable the alert rule.

12i.1 System displays an error message that the action failed.

12i.2 System provides instructions for contacting IT Support.

BE4. System Detects Rule Violation From a Wildfire and Triggers Alert. #4

Pre-condition: The wildfire violates thresholds. The City Operator is authenticated and logged into the system.

VP1. City Operators #1

Main Success Scenario

1. System detects wildfire proximity within 15 km of city limits.
2. System triggers the related alert.
3. System sends a high-priority notification to the city operator.

4. City operator views the active (triggered) alerts.
5. City operator selects the alert from the dashboard.
6. System displays the alert details.
7. City operator reviews the alert information.
8. System logs the acknowledgment with the timestamp and operator ID.
9. System notifies subscribed external systems through the dedicated API endpoint..

VP2. Public Users #2
Main Success Scenario

1. Public users view the wildfire details and recommended safety instructions.

VP3. Administrators #3

N/A

VP4. Third-Party Developers #4
Main Success Scenario

1. Third-party developers confirm delivery of the alert.

VP5. Environmental Monitoring Department #5

N/A

VP6. IT Support #6

N/A

Global Scenario:

Pre-condition: The wildfire violates thresholds. The City Operator is authenticated and logged into the system.
Main Success Scenario

1. System detects wildfire proximity within 15 km of city limits.
2. System triggers the related alert.
3. System sends a high-priority notification to the city operator.
4. City operator views the active (triggered) alerts.
5. City operator selects the alert from the dashboard.
6. System displays the alert details.
7. City operator reviews the alert information.
8. System logs the acknowledgment with the timestamp and operator ID.
9. System sends a notification through the dedicated API endpoint.
10. Third-party developers confirm delivery of the alert.
11. Public users view the wildfire details and recommended safety instructions.

BE5. Confirm Alert After Detecting a Rule Violation Due to a Heat Wave. #5
Pre-condition: An alert has been triggered by a threshold violation and is marked as *Active*. The city operator is authenticated and logged into the system.

VP1. City Operators #1

N/A

VP2. Public Users #2

N/A

VP3. Administrators #3
Main Success Scenario

1. System displays the dashboard showing the triggered alerts.
2. City operator selects an alert from the dashboard.

3. System displays the alert details.
4. City operator reviews alert information.
5. City operator selects the option to acknowledge the alert.
6. System updates the alert status to *Acknowledged*.
7. System confirms that the alert has been acknowledged.

Secondary Scenario

4i. Alert has already been acknowledged by another operator.
4i.1 System refreshes the alert status.
4i.2 System informs the city operator that the alert is already acknowledged.

5i. System fails to update the alert status.
5i.1 System displays an error message to city operator.
5i.2 Alert remains in *Active* status.
5i.3 System provides instructions for contacting IT Support.

VP4. Third-Party Developers #4

N/A

VP5. Environmental Monitoring Department #5

N/A

VP6. IT Support #6

5i System provides IT Support with system logs to investigate and resolve alert acknowledgment issues.

Global Scenario:

Pre-condition: An alert has been generated by the system and is marked as *Active*. The city operator is authenticated and logged into the system.

Main Success Scenario

1. System displays the dashboard showing the triggered alerts.
2. City operator selects an alert from the dashboard.
3. System displays the alert details.
4. City operator selects the option to acknowledge the alert.
5. System updates the alert status to *Acknowledged*.
6. System confirms that the alert has been acknowledged.

Secondary Scenario

4i. Alert got acknowledged by another operator.
4i.1 System refreshes the alert status.
4i.2 System informs the city operator that the alert is already acknowledged.

5i. System fails to update the alert status.
5i.1 System displays an error message to city operator.
5i.2 Alert remains in *Active* status.
5i.3 System provides instructions for contacting IT Support.

BE6. City Operator Resolves an Alert #6

Pre-condition: An alert has been generated by the system and has been acknowledged by a city operator. The city operator is authenticated and logged into the system.

VP1. City Operators #1

N/A

VP2. Public Users #2

N/A

VP3. Administrators #3

Main Success Scenario

1. System displays the dashboard showing the active alerts.
2. City operator selects an alert.
3. System displays the alert details.
4. City operator marks the alert as resolved.
5. System updates the alert status to *Resolved*.
6. System records the time and the operators details.
7. System moves the alert to the *Completed* page.
8. System confirms that the alert has been resolved.

Secondary Scenario

4i. Alert got resolved by another operator
  4i.1 System refreshes the alert status.
  4i.2 System informs the city operator that the alert is already acknowledged.

5i. System fails to update the alert status.
  5i.1 System displays an error message to city operator.
  5i.2 Alert remains in *Acknowledged* status.
  5i.2 System provides instructions for contacting IT Support.

VP4. Third-Party Developers #4

N/A

VP5. Environmental Monitoring Department #5

N/A

VP6. IT Support #6

5i System provides IT Support with system logs to investigate and resolve alert resolution failures or audit logging issues.

Global Scenario:

Pre-condition: An alert has been generated by the system and has been acknowledged by a city operator. The city operator is authenticated and logged into the system.

Main Success Scenario

1. System displays the dashboard showing the active alerts.
2. City operator selects an alert.
3. System displays the alert details.
4. City operator marks the alert as resolved.
5. System updates the alert status to *Resolved*.
6. System records the time and the operators details.
7. System moves the alert to the *Completed* page.
8. System confirms that the alert has been resolved.

Secondary Scenario

4i. Alert got resolved by another operator
  4i.1 System refreshes the alert status.
  4i.2 System informs the city operator that the alert is already acknowledged.

5i. System fails to update the alert status.
  5i.1 System displays an error message to city operator.
  5i.2 Alert remains in *Acknowledged* status.

5i.2 System provides instructions for contacting IT Support.

BE7. Request Access to Public API #7

VP1. City Operators #1
N/A. City Operators have access to environmental data internally and do not require public API credentials.

VP2. Public Users #2
Pre-condition: User has a registered Public User SCHEMAS account and is logged in.
Main Success Scenario

1. User selects "Request Public API".
2. System redirects user to an information page.
3. System informs user that public API access is unavailable and that an account modification to a Third-Party Developer account is required.

VP3. Administrators #3
Pre-condition: User has a registered Administrator SCHEMAS account and is logged in.
Main Success Scenario

1. Administrator navigates to the admin dashboard.
2. Administrator selects "View API request tickets".
3. Administrator selects a ticket to view request details.
4. Administrator adds review comments.
5. Administrator approves the API access request.
6. System generates API credentials with appropriate rate limits.
7. System notifies the user that API access has been granted.

Secondary Scenario

5i. Administrator rejects API access request.
5i.1 System records the rejected request.
5i.2 System notifies the user that the request has been rejected.

VP4. Third-Party Developers #4
Pre-condition: User has a registered Third-Party Developer SCHEMAS account and is logged in.
Main Success Scenario

1. User selects "Request Public API".
2. System displays the API access request form.
3. User enters required information including project name, purpose, organization affiliation, expected request volume, and required endpoints.
4. User submits the API access request form.
5. System validates the submitted information.
6. System stores the API access request record.
7. User receives notification of approved API access.
8. User opens the notification.
9. System displays API credentials and a link to API documentation.

Secondary Scenario

4i. System fails to create API access request.
4i.1 System displays a link to contact IT support.
4i.2 User navigates to the IT support ticket page.
4i.3 User creates a support ticket describing the issue.
4i.4 User submits the support ticket.

4i.5 IT Support responds with remediation instructions.

4i.6 User follows instructions and returns to BE6-3.

6i. User receives notification that API access request was rejected.

6i.1 User opens the rejection notification.

6i.2 System displays rejection reason and administrator comments.

6i.3 User reviews comments and returns to BE6-1.

VP5. Environmental Monitoring Department #5
N/A.

VP6. IT Support #6
Pre-condition: User has a registered IT Support SCHEMAS account and is logged in.
Main Success Scenario

1. IT Support navigates to the IT support ticket dashboard.
2. IT Support selects a support ticket to view details.
3. IT Support resolves the issue and documents actions taken.
4. IT Support marks the ticket as resolved.

Global Scenario:

Pre-condition: User has a registered Third-Party Developer SCHEMAS account and is logged in.
Main Success Scenario

1. User selects "Request Public API".
2. System displays the API access request form.
3. User enters required project and usage information.
4. User submits the API access request form.
5. System validates and stores the request.
6. Administrator reviews and approves the request.
7. System generates API credentials with appropriate rate limits.
8. System notifies the user that API access has been granted.

Secondary Scenario

4i. System fails to submit API access request form.

4i.1 System prompts user to contact IT support.

4i.2 User submits an IT support ticket.

4i.3 IT Support resolves the issue.

4i.4 User retries the API request.

5i. System detects invalid form fields.

5i.1 User is prompted to correct the form.

5i.2 User resubmits the request.

6i. Administrator rejects the request.

6i.1 User receives rejection notification.

6i.2 User reviews comments and returns to BE6-1.

BE8. Modify Account #8

VP1. City Operators #1
N/A. City Operators do not have permission to modify user accounts.

VP2. Public Users #2
Pre-condition: User has a valid Public User SCHEMAS account and is logged in.
Main Success Scenario

1. User selects “Request Account Modification”.
2. System displays an account modification page containing a request form.
3. User enters the purpose of the request, organization affiliation, and developer credentials (if required).
4. User submits the account modification request form.
5. System records the account modification request.
6. System notifies the user of the account modification decision.
7. User opens the notification.
8. System displays confirmation of the change, updated permission details, and administrator comments.

Secondary Scenario

3i. User submits invalid or incomplete form details.
  3i.1 System prompts the user to correct the form and retry submission.

4i. System prompts user to contact customer support.
  4i.1 User navigates to the customer support ticket page.
  4i.2 User submits a support ticket describing the issue.
  4i.3 IT Support responds with remediation instructions.
  4i.4 User follows instructions and returns to BE7-1.

6i. Administrator rejects the account modification request.
  6i.1 User receives a rejection notification.
  6i.2 User reviews comments and returns to BE7-1.

VP3. Administrators #3
Pre-condition: User has a valid Administrator SCHEMAS account and is logged in.
Main Success Scenario

1. Administrator navigates to the admin dashboard.
2. Administrator selects “View Account Modification Request Tickets”.
3. Administrator selects a ticket to view request details.
4. Administrator adds review comments.
5. Administrator approves the account modification request.

VP4. Third-Party Developers #4
Pre-condition: User has a valid Third-Party Developer SCHEMAS account and is logged in.
Main Success Scenario

1. Developer selects “Request Account Modification”.
2. System displays an account modification page containing a request form.
3. Developer enters the purpose of the request, organization affiliation, and developer credentials (if required).
4. Developer submits the account modification request form.
5. System records the account modification request.
6. System notifies the developer of the account modification decision.
7. Developer opens the notification.
8. System displays confirmation of the change, updated permission details, and administrator comments.

Secondary Scenario

3i. Developer submits invalid or incomplete form details.
  3i.1 System prompts the developer to correct the form and retry submission.

4i. System prompts developer to contact customer support.
  4i.1 Developer navigates to the customer support ticket page.

4i.2 Developer submits a support ticket describing the issue.
4i.3 IT Support responds with remediation instructions.
4i.4 Developer follows instructions and returns to BE7-1.
6i. Administrator rejects the account modification request.
6i.1 Developer receives a rejection notification.
6i.2 Developer reviews comments and returns to BE7-1.

VP5. Environmental Monitoring Department #5
N/A.

VP6. IT Support #6
Pre-condition: User has a registered IT Support SCHEMAS account and is logged in.
Main Success Scenario

1. IT Support navigates to the IT support ticket dashboard.
2. IT Support selects a support ticket to view details.
3. IT Support resolves the issue and documents actions taken.

Global Scenario:

Pre-condition: User has a valid Public User or Third-Party Developer SCHEMAS account and is logged in.
Main Success Scenario

1. User selects "Request Account Modification".
2. System displays the account modification request form.
3. User submits required modification details.
4. Administrator reviews the modification request.
5. Administrator approves the request.
6. System records the account modification.
7. System notifies the user of the modification.
8. User reviews confirmation and updated permissions.

Secondary Scenario

3i. User submits invalid or incomplete form details.
3i.1 User is prompted to retry submission.
4i. System prompts user to contact customer support.
4i.1 User submits a support ticket.
4i.2 IT Support resolves the issue.
4i.3 User retries the modification request.
9i. Administrator rejects the request.
9i.1 User receives rejection notification.
9i.2 User reviews comments and returns to BE7-1.

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1. The system shall use a cool, non-distracting background that does not draw visual focus away from the operational dashboard content.
Rationale: SCEMAS is intended for monitoring environmental conditions. A background with colours that are not warm such as bright red/yellow ensures that users' attention is reserved for critical dashboard elements such as metrics, dashboards, and alerts.

LF-A2. The system should use a consistent font family across all screens.
Rationale: A standardized font enhances the user experience by providing a consistent experience across all screens making it easier to navigate.

LF-A3. Visual alert indicators shall be designed to clearly stand out from non-critical information.
Rationale: Environmental alerts often require immediate action. Distinct visual indicators ensure that critical conditions are quickly recognizable, even when operators are monitoring large volumes of data.

LF-A4. The system shall use a font size that is readable; the font size should also be consistent across all screens relative to the header type.
Rationale: This ensures that information can be categorized into classes such as title, heading, paragraph heading and paragraph body while being readable to users.

5.1.2 Style Requirements

LF-S1. The system shall adapt its layout to different screen resolutions.
Rationale: City operators may access the system from devices with varying screen sizes. Responsive layout behavior ensures a consistent experience across supported devices.

LF-S2. The system shall use a consistent layout that is the same for all dashboard views.
Rationale: A consistent layout structure helps users build familiarity with the interface, preventing unnecessary navigation or relearning on every new screen.

LF-S3. The system shall group related information and controls using spacing, alignment, and section boundaries.
Rationale: Clear visual grouping improves information organization and allows users to quickly identify related data.

LF-S4. The system shall limit the text density on each screen.
Rationale: Screens should be informative and require as little navigation as possible but not overwhelm the user with information cluttering the user experience.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1. The system should show brief tooltips for all major controls, indicators and buttons.
Rationale: Giving tooltips enhances user understanding and provides context for each element on the dashboard.

UH-EOU2. The system should allow users to report bugs to IT support.
Rationale: If users encounter a problem they should be able to receive a response from the SCHEMAS team to address their issue and implement fixes.

UH-EOU3. The system should maintain consistent terminology and terms throughout the interface.
Rationale: Consistent terminology improves learnability and prevents inconsistent descriptions across multiple screens.

UH-EOU4. The system shall cache user interface preferences, such as selected dashboards or view settings.
Rationale: A cache on the preference allows users to have a consistent configuration in which they view the dashboard which makes the screens tailor made for any particular user.

UH-EOU5. The system shall contain a Frequently Asked Question section.
Rationale: An FAQ section allows users to get quick and easy answers to commonly asked questions reducing the time required to get acclimated to the system [2].

5.2.2 Personalization and Internationalization Requirements

UH-PI1. The system shall allow users to configure personal interface preferences.
Rationale: Allowing users to change their default views, dashboard preference and interface preferences will help increase usability among users.

UH-PI2. The system shall allow users to change their language preferences.
Rationale: All users should be able to understand the information on the screen. This is essential for all users to be able to view the system equally and accurately.

UH-PI3. The system shall support internationalized measurement units, date formats and time zones.
Rationale: Environmental data must be able to be measured and accessed across all time zones. Supporting international viewing standards ensure data is accessible to people regardless of their location.

5.2.3 Learning Requirements

UH-L1. Users shall be able to learn how to navigate the main dashboard and interpret key environmental metrics within 20 minutes of first use.
Rationale: Operators must quickly understand the core interface to respond to environmental events effectively. Limiting the learning time ensures the system is intuitive and reduces onboarding effort.

UH-L2. Users shall be able to configure personal preferences, such as measurement units, default dashboards, and theme settings, within 10 minutes of first use.
Rationale: Users should quickly learn how best to tailor the application to them. If they are able to personalize the system quickly it will make sure that the rest of their time using the system is more enjoyable.

UH-L3. A trained city operator(someone that has 10+ hours of education on the system) must be able to log into the dashboard and successfully acknowledge a critical environmental alert within 30 seconds of the alert appearing.
Rationale: City operators that are familiar with the system should be able to navigate the dashboard and acknowledge an alert, this shows the system is user-friendly and accessible.

UH-L4. A trained developer (someone that has 4+ years of experience developing) must be able to understand the API documentation within a 1 hour time frame.
Rationale: Developers that are willing to subscribe to APIs should be able to understand the documentation in the short(1-hour) time frame. If they are able to understand the documentation within this time it means that the documentation is written clearly and helps improve usability of the service.

5.2.4 Understandability and Politeness Requirements

UH-UP1. The system shall use professional and neutral wording in all user-facing communications.
Rationale: Maintaining a professional tone is important to ensure users are not offended by any terminology and allows users to strictly focus on the data.

UH-UP2. The system shall avoid technical jargon or abbreviations in messages unless universally understood by target users.
Rationale: Using understandable and general knowledge ensures users of all levels of technicality can understand the information [3]. This prevents confusion and ensures understandability.

5.2.5 Accessibility Requirements

UH-A1. The system shall provide text-to-speech functionality for all essential interface elements.
Rationale: Text to speech allows users that are blind or low-vision to interpret environmental data.

UH-A2. The system shall provide captions or transcripts for all audio content, including notifications and messages.
Rationale: Users who are deaf or hard-of-hearing should be able to access spoken information like any other able user and a viable way to do this is using text-to-speech [4].

UH-A3. The system shall use a color palette that is accessible to users with common forms of color vision deficiency.
Rationale: City-operated systems must be inclusive and accessible. 1/16 men and 1/240 women have some form of colour deficiency and therefore should be accommodated [5]. Colorblind-friendly design ensures that all operators can correctly interpret system states and alerts without relying solely on colour perception.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1. The web dashboard for city operators should have a load time of less than 1 second.
Rationale: Having a fast-loading website will ensure users remain engaged. A study shows that 40% of users will leave a website if it takes longer than 3 seconds to load [6].

PR-SL2. The system should provide a REST API for public use with a response time of less than 1 second.
Rationale: APIs with lower response time lead to many benefits, like enhancing user satisfaction and increasing cost effectiveness of the system by allowing fewer resources to be used [7]. Longer API response times will likely result in user frustration and abandonment [7].

5.3.2 Safety-Critical Requirements

PR-SC1. The system should ensure that alerts are posted in less than 5 seconds of the internal system finding a threshold broken.
Rationale: A rapid alert system allows operations to respond to any alerts within a timely manner, staging interventions or putting out city wide alerts where possible.

5.3.3 Precision or Accuracy Requirements

PR-PA1. For each environmental indicator (air quality, noise levels, temperature, humidity) the system shall have a standard deviation of less than 15%.
Rationale: Inaccurate data could lead to a false sense of security or a false sense of urgency when trying to address the implications of the measurement [8], its important for the measurement to remain accurate to avoid these situations.

5.3.4 Reliability and Availability Requirements

PR-RA1. The database should have an up time of at least 99%
Rationale: This allows any authentication requests to properly go through and avoids cases where users cannot use the system due to authentication requests.

PR-RA2. The system should ensure that alerts have a false positive rating of less than 1%
Rationale: A high false positive rating can lead to improper use of resources, impaired efficiency and a loss of trust of users [9]. Its important to have a low false positive rate to avoid taking unnecessary measures to correct a false issue.

5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1. The System should not fail if one component of it fails.
Rationale: This is to ensure that the system stays functional even if a sensor fails, meaning the system will continue to collect data from the other sensors.

PR-RFT2. The system should detect and log any errors in under 5 seconds
Rationale: This is to ensure that all errors are properly logged as well as properly addressed within a timely manner.

5.3.6 Capacity Requirements

PR-C1. The system should be able handle and store information from at least 150 sensors across the city.
Rationale: A mid sized city requires 150 sensors [10] to allow the system to collect accurate data for accurate measurements. Ensuring the whole city is covered allows us to take action where its needed.

5.3.7 Scalability or Extensibility Requirements

PR-SE1. The internal code of our system should follow SOLID design principles to ensure scalability.
Rationale: The Solid design principles are a set of principles that help developers write clean, maintainable, and scalable code [11]. Adopting these principles in the system's design allows our system to be scalable and extensible.

5.3.8 Longevity Requirements

PR-L1. The sensors should use sustainable materials that guarantee a lifespan of at least 2 years
Rationale: This is to ensure that each sensor can work optimally without the need to constantly replace them to ensure proper data collection.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE-EPE1. SCEMAS shall operate in a cloud-hosted environment using commodity server infrastructure (virtual machines/containers) and shall not require any specialized physical hardware.
Rationale: The system should be deployable using free-tier or educational cloud resources, and sensor hardware is simulated rather than physically deployed.

OE-EPE2. SCEMAS shall support operation over standard internet connections and tolerate intermittent connectivity from simulated sensor publishers by continuing to run and ingest telemetry once connectivity resumes.
Rationale: IoT-style telemetry publishers may disconnect; the system should remain stable and continue ingesting when publishers reconnect.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1. SCEMAS shall ingest sensor telemetry through the MQTT protocol and validate each incoming message against a defined schema and plausible value ranges before storage.
Rationale: MQTT ingestion and validation are core requirements for reliable telemetry processing.

OE-IA2. SCEMAS shall expose a read-only REST API for public/third-party consumption that returns aggregated, non-sensitive environmental data.
Rationale: The public-facing API must be read-only and provide aggregated data rather than sensitive/raw details.

OE-IA3. When an alert is triggered, SCEMAS shall notify subscribed external systems through a dedicated notification API endpoint.
Rationale: External notifications on alert triggers are a required system capability.

5.4.3 Productization Requirements

OE-P1. SCEMAS shall provide configuration-driven deployment for thresholds, zone definitions, and alert rules so the same build can be used across environments (dev/test/demo).
Rationale: Configuration-based behavior supports different setups without rebuilding code and better fits a reusable platform.

OE-P2. SCEMAS shall include clear operator and developer documentation, including public API documentation sufficient for a competent third-party developer to request and interpret data within two hours.
Rationale: Good documentation is required so external developers can successfully use the API quickly.

5.4.4 Release Requirements

OE-R1. SCEMAS shall support separate deployment profiles for (at minimum) development and demonstration/staging, with distinct configuration for database endpoints, authentication secrets, and rate limits.
Rationale: Separating environments reduces demo risk and avoids mixing secrets/data across deployments.

OE-R2. Each SCEMAS release shall be versioned and include a rollback strategy for the dashboard and API services.
Rationale: Cloud deployments can fail; rollback supports service continuity.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1. SCEMAS shall be designed as modular services/components (e.g., ingestion, alerting, storage, dashboard/API) so that changes to one component do not require rewriting unrelated components.
Rationale: Modularity reduces regression risk and makes future extensions (new metrics, new rules) easier.

MS-M2. SCEMAS shall maintain automated logs for major runtime events (telemetry ingestion failures, validation rejections, alert triggers, and admin actions) to support debugging and ongoing maintenance.
Rationale: Logs are essential for diagnosing issues in distributed systems and supporting ongoing maintenance.

5.5.2 Supportability Requirements

MS-S1. SCEMAS shall provide a support-facing troubleshooting view or documented runbook steps that allow IT Support/Administrators to identify common failures (database unreachable, API rate-limit triggered) and the recommended recovery action.
Rationale: Support needs a fast way to narrow down what broke in a multi-component system (ingestion + storage + dashboards + API).

MS-S2. SCEMAS shall provide user-facing error messages on the dashboard/API that are actionable (what happened + what to try next) without exposing sensitive internal details.
Rationale: Clear errors reduce support load, and avoiding internal details supports secure operations.

5.5.3 Adaptability Requirements

MS-A1. SCEMAS shall allow new sensor types/metrics to be introduced by extending the telemetry schema and validation rules, without changing the operator dashboard's core navigation or authentication

flow.
Rationale: Monitoring evolves; the system should adapt to new indicators while keeping the operator experience stable.

MS-A2. SCEMAS shall be adaptable to multiple deployment providers (or local containers for demos) by avoiding provider-specific hard dependencies in core logic.
Rationale: Teams may switch hosting/deployment options; portability reduces rework.

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1. The system must require all users of the dashboard to authenticate using an email/username and password, being granted access to the system.
Rationale: This ensures that only people who are authorized can access the system, which prevents the risk of authorized individuals to alter environmental data [12].

SR-AC2. The system must support multi-factor authentication (MFA) for administrative accounts.
Rationale: MFA helps by eliminating the possibility of compromising the credentials of an administrators account. This is crucial, since the administrator is a high-privilege user who can define alert rules, and view sensitive system information [13].

SR-AC3. The system must enforce Role-Base Access Control (RBAC) to distinguish between the different user groups, such as administrator, city operator, and read-only public users.
Rationale: Enforcing an RBAC policy ensures that each user group can only perform actions that are appropriate to their respective role. This helps to reduce accidental misuse of the systems functions.

SR-AC4. The system must lock out a user account after five consecutive failed login attempts.
Rationale: Protects against repeated attempts of trying to login to an account that does not belong to the respective individual [12].

SR-AC5. All IoT devices must be authenticated with a unique key before sending telemetry data to the system.
Rationale: Prevents harmful telemetry to be injection into the system, by limiting entering data to registered and authorized sensors [14].

5.6.2 Integrity Requirements

SR-INT1. Any data transmitted between IoT devices, and the system must be encrypted, and include integrity protection.
Rationale: Ensures that telemetry is protected, and cannot be changed during transmission. As a result, this helps maintain confidentiality and integrity [15].

SR-INT2. All incoming sensor telemetry must be validated against a defined schema and acceptable value ranges before being stored.
Rationale: Ensures that only correct and meaningful data is allowed to enter the system. This restricts any corrupted/malicious data from affecting any environmental analytics and alerts.

SR-INT3. The system must maintain an audit log to keep track of any data modification, such as alert rule updates, or alert status changes.
Rationale: An audit log is essential to provide traceability to identify any potential modifications made accidentally, or by unauthorized personnel [16].

SR-INT4. The system should use cryptographic hashing for any alert generated, to ensure that the alert data remains secure, and unaltered after generation.
Rationale: Enforces data integrity, and prevents city operators from receiving false notifications [17].

5.6.3 Privacy Requirements

SR-P1. The system will not collect or store any personally identifiable information from public users.
Rationale: Protects individual privacy, and ensures compliance with PIPEDA regulations [18].

SR-P2. The system shall only collect environmental data at the level of general city zones.
Rationale: Ensures citizen privacy by preventing sensor data from tracking individual residents

SR-P3. Public APIs should only provide non-sensitive data to public citizens and third-party developers.
Rationale: Prevents misuse of the API, and protects privacy.

5.6.4 Audit Requirements

SR-AU1. The system shall log all user login attempts, including the timestamp, their distinct identification number, and the location of the request.
Rationale: Ensures accountability for access attempts [10].

SR-AU2. The system shall log all access to sensitive data, and data modification such as updating, adding or deleting rule alerts, or registering new sensors or accounts.
Rationale: Ensures sensitive data is properly handled, and traceability of critical changes [19].

SR-AU3. All audit logs should be retained for at least 1 years.
Rationale: Ensures there is enough time look at incidents, or check compliance [20].

SR-AU4. All audit logs should be protected with access controls to ensure that only authorized security can view or extract information from them.
Rationale: Prevents critical information from being shared without authorization [20].

5.6.5 Immunity Requirements

SR-IM1. The system shall keep receiving and storing sensor data even when the real-time processing system is down.
Rationale: Makes sure no sensor data is lost if part of the system stops [21].

SR-IM2. The system shall immediately try again when sending sensor data fails.
Rationale: Ensures that sensor data eventually reaches the system, even in the case of network problems [21].

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP-C1. The system shall not allow users to create an account with an inappropriate or discriminatory name.
Rationale: Prevents harassment, and hate speech, while ensuring that the system remains professional, inclusive and aligned with standards

CP-C2. The system shall not use any symbols, colours, and icons in the dashboards that can be considered offensive based one ones religion, ethnicity, disability or sexual orientation.
Rationale: Ensures the user interface stays inclusive for a diverse population. Helps reduces the risk of discrimination and reputational damage.

5.7.2 Political Requirements

CP-P1. The system shall comply with all federal, provincial, and municipal regulations regarding environmental monitoring and data sharing [22].
Rationale: Ensures legal compliance and helps to avoid political conflicts.

CP-P2. The system shall be transparent in how it collects and stores data by giving government authorities access to such data and system reports [23].
Rationale: Enables accountability to political authorities.

CP-P3. The system shall provide ways to share environmental data to other authorized groups.
Rationale: Supports collaboration between government organizations.

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1.

5.8.2 Standards Requirements

LR-STD1.

6 Innovative Feature

An innovative feature that could be added is an included simulation feature that would allow city operators to adjust environmental parameters and observe the projected outcome in a visual dashboard overlaid on top of the city. This would help city operators plan response strategies, visualize worst case outcomes, and help train employees without impacting live data.

A Division of Labour

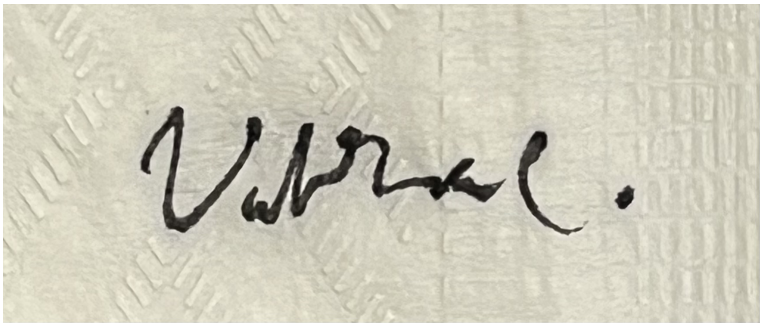
Benjamin Bloomfield

- 1.4 References
  - Managed all in-text citations.
  - Formatted each website in IEEE format.
  - Organized all citations in the references.bib file.
- Revised 2.2 Product Functions as a group.
- 2.3 User Characteristics.
- 2.4 Constraints.
- 4 Highlights of Functional Requirements:
  - Helped devise the list of viewpoints.
  - Wrote the LaTeX code structure for business events.
  - Wrote BE1-BE6.
- Non-Functional Requirements: 5.6, 5.7
- Organized storage and display of figures.
- Wrote the LaTeX code structure for Division of Labour

Date: February 13, 2025

Vikram Chandar

- Revised 1.2 Scope and 2.1 Product Perspective
- 1.3 Definitions, Acronyms, and Abbreviation
- Revised 2.2 Product Functions as a Group
- 2.5 Assumptions and Dependencie
- Functional Requirements
  - Helped create the list of viewpoints as a team
  - Wrote BE6,BE7.
- 5.1 Look and Feel Requirement
- 5.2 Style Requirements
- 6 - Thought of innovative feature and adapted project around this feature.



Date: February 13, 2025

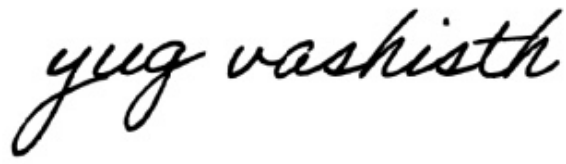
Trisha Panchal

-

Yug Vashisth

- Introduction
- Overview (1.5)
- 2.1 Product Purpose
- Figure 1(2.1) Diagram

- Wrote the Origial BE5 (now revised by and combined with other BE)
- 5.4 Operational and Environmental Requirements
- 5.5 Maintainability and Support Requirements
- General team contributions, engaging in discussions, helping create ideas etc.



Date: February 14, 2025

Hamza Nadeem

- 1.1 Purpose
- 2.2 Product Functions
  - Decided to separate functions into Modules as group.
  - Compiled list of modules and functions.
  - Created State diagram of module interactions.
- Helped come up with Business events and viewpoints.
- 5.3 Performance Requirements

Date: February 13, 2025