

# Deliverable #1 Template : Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

**Tutorial Number:** T01

**Group Number:** G4

**Group Members:** Lukas Buehlmann, David Olejniczak, Vanessa Lai, Saqib Khan, Suzanne Abdullah

## 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) provides an overview of the software requirements for the Smart City Environmental Monitoring and Alert System (SCEMAS). SCEMAS is a software application designed to collect, analyze, and visualize environmental data such as air quality, temperature, humidity, and noise levels to support city-level monitoring and decision-making.

## 1.1 Purpose

This document outlines the system's purpose, scope, characteristics of intended users, and the functional and non-functional requirements that guide the design and development of the system.

This document is intended for internal SCEMAS stakeholders, including but not limited to project managers, developers, domain experts, and SCEMAS team members/investors. No prior readings are required.

## 1.2 Scope

Smart City Environmental Monitoring and Alert System (SCEMAS) is an environmental monitoring software system that allows users to quickly and easily monitor environmental data for metrics such as air quality, noise levels, temperature, UV levels, and humidity across geographical zones, as well as configure alerts and reminders.

The software products in the system include a dashboard for real-time data visualization of active alerts and geographical maps of sensor locations. Real-time data aggregation will be performed within geographical zones to transform data into actionable insights for time-sensitive alerts in the system. A read-only REST API for non-sensitive data will be publicly available for any user, which uses HTTP methods to ensure safe access and rate limiting to prevent abuse and security threats. The system will use MQTT for transmitting all sensor telemetry securely through encryption, validating incoming messages for correct format, adherence to the defined scheme and validating value ranges. The telemetry stored in a time-series database will not store or process PII. Audit log history of all significant events, alert triggers and user management will be stored for a minimum of 1 year.

The system will have role-based access control, which will authorize user-specific access to data based on a user's predefined role: City Operator and System Administrator. The operators will primarily use the application to monitor environmental conditions, view raised alerts and respond to events quickly using in-app tools. Operators will have write access to change event statuses and create alert rules. Administrators will have elevated privileges to approve and verify any alert rules being made based on specific thresholds or historical data trends.

SCEMAS will have a separate public facing dashboard interface that will allow any public user to view the visualized non-sensitive data from the REST API.

Our software will have Service Level Agreements (SLA) to guarantee a dashboard response time of 30 seconds upon logging in. A goal of the software is to increase the number of public users accessing the REST API to improve data quality and product development.

## 1.3 Definitions, Acronyms, and Abbreviations

SCEMAS: Smart City Environmental Monitoring and Alert System

PIPEDA: Personal Information Protection and Electronic Documents Act. RTO: Recovery Time Objective is the maximum amount of time that it should take to restore system operations after a service disruption.

PII: Personally Identifiable Information

UI: User Interface is the graphical display of the application the users interact with

UX: Refers to the user experience using a given user interface

MQTT: Message Queuing Telemetry Transport

RBAC: Role-Based Access Control

IoT: Internet of Things

UV: Ultraviolet

REST API: Representational State Transfer Application Programming Interface

#-px: Refers to the pixel size of attributes on a user interface, e.g. 14-px sized font

## 1.4 References

- [1]. K. Erik, "The Responsive Website Font Size Guidelines," Learn UI Design. Accessed: Feb. 01, 2026. [Online]. Available: <https://learnui.design/blog/mobile-desktop-website-font-size-guidelines.html>
- [2]. Altamira, "Common Screen Sizes for Responsive Web Design." Accessed: Feb. 01, 2026. [Online]. Available: <https://www.altamira.ai/blog/common-screen-sizes-for-responsive-web-design/>
- [3]. L. Bruton, "User interface guidelines: 10 essential rules to follow - UX Design Institute." Accessed: Feb. 01, 2026. [Online]. Available: [urlhttps://www.uxdesigninstitute.com/blog/10-user-interface-guidelines/](https://www.uxdesigninstitute.com/blog/10-user-interface-guidelines/)
- [4]. "Project Outline for Software Design I" [Online]. Available: <https://avenue.cllmcmaster.ca/d2l/le/lessons/727030/topics/5498994>[Accessed:12-Feb-2026]
- [5]. J. Dunn, "Why you need a FAQ page and how to create an effective FAQ template," Knowledge-BaseBlog ,<https://www.knowledgebase.com/blog/why-you-need-a-faq-page-and-how-to-create-an-effective-faq-template/> (accessed Feb. 3, 2026).
- [6]. M. Hertzem, "Understanding preference: A meta-analysis of user studies", <https://doi.org/10.1016/j.ijhcs.2024.103408> (accessed Feb. 3, 2026).
- [7]. ILAC, "English proficiency levels: How to determine your level," <https://ilac.com/blog/how-to-determine-english-levels/> (accessed Feb. 3, 2026).
- [8]. E. Magner, "Digital Signage Font Guide: Chapter 2: Font colors," RSS, <https://www.mvix.com/blog/digital-signage-font-guide-font-colors> (accessed Feb. 3, 2026).
- [9]. Web.dev (Google), "Why Speed Matters for Your Website and Its Users," 2024. [Online]. Available: <https://web.dev/articles/why-speed-matters>. [Accessed: Jan. 31, 2026].
- [10]. Bodytrak, "The Role of Geolocation Technology in Enhancing Emergency Responses for Worker Safety," May 2024. [Online]. Available: <https://bodytrak.co/news/worker-safety-geolocation-technology-emergency-responses/>. [Accessed: Jan. 31, 2026].
- [11]. UptimeRobot, "What Does 99.9% Uptime Mean?," June 2024. [Online]. Available: <https://uptimerobot.com/blog/what-does-999-uptime-mean/>. [Accessed: Jan. 31, 2026].
- [12]. IBM, "What is Disaster Recovery?" 2024. [Online]. Available: <https://www.ibm.com/topics/disaster-recovery>. [Accessed: Feb. 2, 2026].
- [13]. Microsoft, "Understand the offline capabilities of IoT Edge devices and gateways," 2024. [Online]. Available: <https://learn.microsoft.com/en-us/azure/iot-edge/offline-capabilities>. [Accessed: Feb. 2, 2026].
- [14]. GeeksforGeeks, "Graceful Degradation in Distributed Systems," 2025. [Online]. Available: <https://www.geeksforgeeks.org/system-design/graceful-degradation-in-distributed-systems/>. [Accessed: Feb. 3, 2026].
- [15]. Netdata, "What is Database Concurrency?" 2024. [Online]. Available: <https://www.netdata.cloud/academy/what-is-database-concurrency/>. [Accessed: Feb. 3, 2026].

- [16]. StatCounter Global Stats, “Desktop Operating System Market Share Worldwide.” Accessed: Feb. 01, 2026. [Online]. Available: <https://gs.statcounter.com/os-market-share/desktop/worldwide/>
- [17]. P. Solutions, “How to minimize downtime during your control system upgrade,” Process Solutions, Inc., <https://processsolutions.com/how-to-minimize-downtime-during-your-control-system-upgrade/> (accessed Feb. 3, 2026).
- [18]. K. O’Brien, “What is application monitoring?,” IBM, <https://www.ibm.com/think/topics/application-monitoring> (accessed Feb. 3, 2026).
- [19]. R. Davydov, “How to build scalable, future-proof custom software,” Built In, <https://builtin.com/articles/build-future-proof-software> (accessed Feb. 3, 2026).
- [20]. R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, “Role-based access control models,” in *Computer*, vol. 29, no. 2, pp. 38-47, Feb. 1996, doi: 10.1109/2.485845.
- [21]. Canadian Centre for Cyber Security, “Annex 3A: Security control catalogue,” Guidance on the Management of IT Security Risks (ITSG-33). [Online]. Available: <https://www.cyber.gc.ca/en/guidance/annex-3a-security-control-catalogue-itsg-33>. [Accessed: Feb. 10, 2026].
- [22]. Government of Canada, “Official Languages Act (R.S.C., 1985, c. 31 (4th Supp.)),” Justice Laws Website. [Online]. Available: <https://laws-lois.justice.gc.ca/eng/acts/o-3.01/>. [Accessed: Jan. 31, 2026].
- [23]. Government of Canada, “Criminal Code (R.S.C., 1985, c. C-46), s. 319,” Justice Laws Website. [Online]. Available: <https://laws-lois.justice.gc.ca/eng/acts/c-46/section-319.html>. [Accessed: Jan. 31, 2026].
- [24]. Criminal Code, R.S.C. 1985, c. C-46, s. 372. [Online]. Available: <https://laws-lois.justice.gc.ca/eng/acts/c-46/section-372.html>
- [25]. Government of Canada, “Criminal Code (R.S.C., 1985, c. C-46), s. 184,” Justice Laws Website. [Online]. Available: <https://laws-lois.justice.gc.ca/eng/acts/c-46/section-184.html>. [Accessed: Jan. 31, 2026].
- [26]. Government of Canada, “Personal Information Protection and Electronic Documents Act (S.C. 2000, c. 5),” Justice Laws Website. [Online]. Available: <https://laws-lois.justice.gc.ca/eng/acts/p-8.6/>. [Accessed: Jan. 31, 2026].
- [27]. Government of Ontario, “Integrated Accessibility Standards, O. Reg. 191/11,” ServiceOntario e-Laws. [Online]. Available: <https://www.ontario.ca/laws/regulation/110191>. [Accessed: Jan. 31, 2026]
- [28]. Government of Ontario, “Municipal Freedom of Information and Protection of Privacy Act, R.S.O. 1990, c. M.56,” ServiceOntario e-Laws. [Online]. Available: <https://www.ontario.ca/laws/statute/90m56>. [Accessed: Jan. 31, 2026].
- [29]. Government of Canada, “Canadian Environmental Protection Act, 1999 (S.C. 1999, c. 33),” Justice Laws Website. [Online]. Available: <https://laws-lois.justice.gc.ca/eng/acts/c-15.31/>. [Accessed: Jan. 31, 2026].
- [30]. World Wide Web Consortium, “Web Content Accessibility Guidelines (WCAG) 2.1,” W3C Recommendation, June 5, 2018. [Online]. Available: <https://www.w3.org/TR/WCAG21/>. [Accessed: Jan. 31, 2026].

## 1.5 Overview

Section 2 discusses the overall product description including the product functions, user characteristics, constraints, assumptions, and the perspective of the product relative to existing projects. Section 3 contains a use case diagram for the system. Section 4 highlights the functional requirements including the use cases organized by business event, different viewpoints for each event and associated scenarios, and a global scenario for each business event. Section 5 discusses the non-functional requirements including the look and feel requirements, usability and human requirements, performance requirements, operational requirements, maintainability, security, and political requirements.

## 2 Overall Product Description

### 2.1 Product Perspective

SCEMAS is a smart city monitoring dashboard developed for environmental monitoring of the average smart city built for a web application. Related products include Siemens MindSphere, AxxonSoft (Safe City) and Cisco Kinetic for Cities, where city-wide data is monitored from various IoT devices based on geographic zones. However SCEMAS differentiates itself by focusing on unifying all environmental data into one centralized dashboard and database with the ability to respond to events via built-in tools rather than just observing them. The product will facilitate various roles of access, allowing users to be created, edited, and removed, while keeping extensive historical logs of all user actions, alerts, and collected data for auditing similar to the compliance standards found in the related products. Various alerts and reminders can be created based on geographical zones and real-time aggregated data like in AxxonSoft.

The product will interact with the larger system of distributed IoT sensors utilizing the MQTT protocol to protect the data and ensure system privacy and security. SCEMAS will act as a central ingestion hub, receiving high-volume telemetry streams from sensors to detect threshold violations in real-time. The product will also utilize real-world mapping software such as OpenStreetMap or GoogleMaps to visualize these telemetry streams. Similar to AxxonSoft map view, SCEMAS will overlay live heatmaps and alert markers on the geography, allowing City Officials to pinpoint the exact location of environmental hazards and acknowledge them directly from the map interface.

The product will have a sub-system to export data for public use, using a secure gateway allowing the city to share non-sensitive environmental information to an external public facing website and a read-only API. This website will be open to the public and will have a heatmap on limited metrics. This ensures City Officials have full command-and-control capabilities of public facing data.

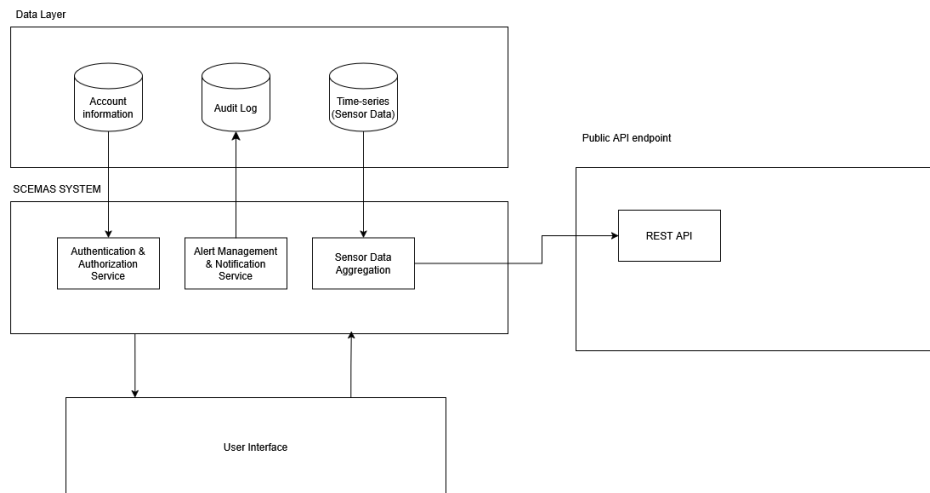


Figure 1: SCEMAS System Diagram

### 2.2 Product Functions

The system functionality will be focused on both data collection and processing done automatically as well as interactions between users and the system. At a high level, the system is responsible for collecting environmental data, storing this data, calculating aggregations of data series by city zone, and comparing stored alert configurations to present data. The city operators and administrators can interact with this system through a dashboard interface while public users can use a public facing API that provides different functionality from the city dashboard. An important property of the system is that it has functionality to

authenticate users and limit the access of different users based on their role.

Function	Description
Environmental Data Collection	The system collects data from many sensors throughout a city. The system ensures the incoming data is in the correct format, within bounds, and valid before accepting it.
Data Storage and Management	Sensor data is stored for real-time and historical use, organized by time and stored by city zone.
Data Processing and Aggregation	Data is aggregated over time and by city zone for both private use and public access depending on the sensitivity of the data.
Alert Detection and Management	Administrators can manage and add new alert rules. Alerts occur when real time data triggers the rule of an alert. Alerts are logged and notifications are sent to the dashboard
Dashboard for City Operators	The dashboard is a secure interface for city operators to view real time data, current alerts, and view trends.
Public Data Access Interface	A read-only public interface is exposed which only allows access to aggregated, non-sensitive environmental data.
RBAC Management	The system supports Role-Based Access Control to differentiate administrators from city operators.
Authentication	All user sign-ins are authenticated and all data in transit is encrypted. All significant events, such as alert triggers and administrative changes, are logged by the system.

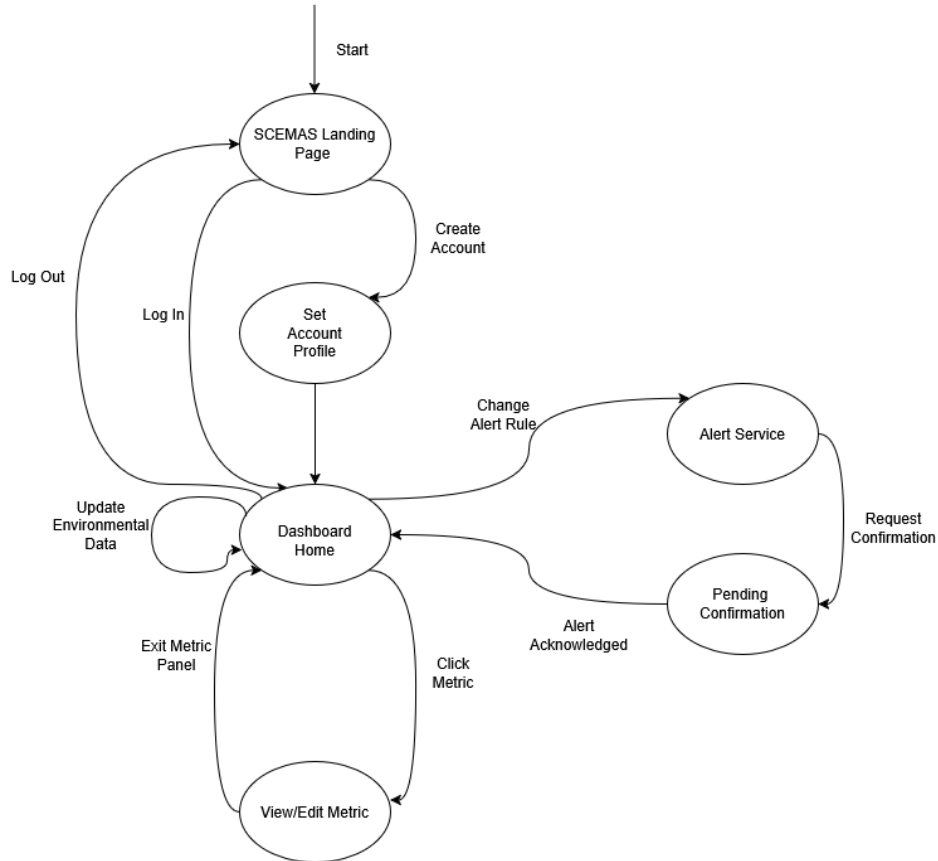


Figure 2: SCEMAS State Diagram

## 2.3 User Characteristics

As the app will have different interfaces for different types of users, it is important for us to define the distinct needs and abilities of each category of user.

### City Operators

City operators will be responsible for monitoring the application's dashboard to put out alerts for events.

1. **Educational level:** Secondary school education
  - City operators are expected to have at least a high-school level of education to properly interpret the data on the dashboard.
2. **Experience:** Prior training in using the system
  - City operators are expected to be trained in how to use the dashboard prior to starting their role. Prior experience in using a similar dashboard would also be acceptable experience.
3. **Technical experience:** Average computer knowledge
  - City operators are expected to have a medium level of expertise in using computers, including an understanding of file systems and more complex graphical user interfaces.

### City Residents

City residents will view city-wide alerts notifications from a different interface than city operators.

1. **Educational level:** Basic reading and comprehension ability
  - City residents fall into a wide range of age and education levels, so in order to best accommodate the target audience, the application should be accessible to those with only a primary school education level.
2. **Experience:** None
  - City residents should be able to learn to use the app without having any prior experience with it, or similar products.
3. **Technical experience:** Minimal understanding of computers
  - City residents are expected to have a minimal understanding of how computers work in order to launch the application, as well as some understanding of common app navigation methods (e.g. basic menus, buttons)

### Application Administrators

Administrators will be able to define alert rules for the city operator dashboard.

1. **Educational level:** Post-Secondary education
  - The administrators of the system are expected to have completed a four year program at a university or college relating to city management or administrative systems.
2. **Experience:** Prior training in using the system
  - The application administrators of the system are expected to have prior training in using the system, as well as prior experience in using similar software.
3. **Technical experience:** Mid-High level of computer knowledge
  - System admins are expected to have a mid to high understanding of computers, and should have a good understanding of file systems, complex system interfaces, trouble-shooting, and potentially command-line interfacing.

The app should also be accessible to individuals with varying levels of visual, auditory and mobility impairments on all user-facing-interfaces to ensure it is usable for all members of the general population.



## 2.4 Constraints

The following are constraints which may limit the developer's options and approach:

**Communication Protocol:** The system must accept and send all environmental telemetry data using the MQTT protocol. This is mandated by the existing IoT infrastructure already deployed in smart city environments. The system cannot use alternative protocols such as HTTP.

**Regulation Compliance:** The system must comply with all data protection and privacy regulations, such as the PIPEDA in Canada.

**Public Data Access:** Only non-sensitive environmental data should be exposed through the public interface. Any sensitive data must be restricted for authorized users. This constraint stems from the privacy-by-design requirement specified in the system design.

**Budget Constraint:** The system must be developed and deployed with zero monetary budget. All software tools, cloud services, databases, and third-party APIs must utilize free-tier offerings.

**Fixed Timeline:** The system must be developed within the time allocated. This fixed deadline constrains the scope of features that can be implemented and requires prioritization of core functionality over advanced features.

## 2.5 Assumptions and Dependencies

1. We are a private company being contracted by the city government to create this platform.
2. The project is assumed to be funded by the Government of Canada.
3. The system is assumed to be limited to operation within one city.
4. It is assumed that there is valid data flowing into the system from various IoT devices for Business Events to be successful (as stated in the Precondition for BE1).
5. It is assumed that accompanying training modules will be provided for City Operators, and they can read the system documentation.

## 2.6 Apportioning of Requirements

N/A

### 3 Use Case Diagram

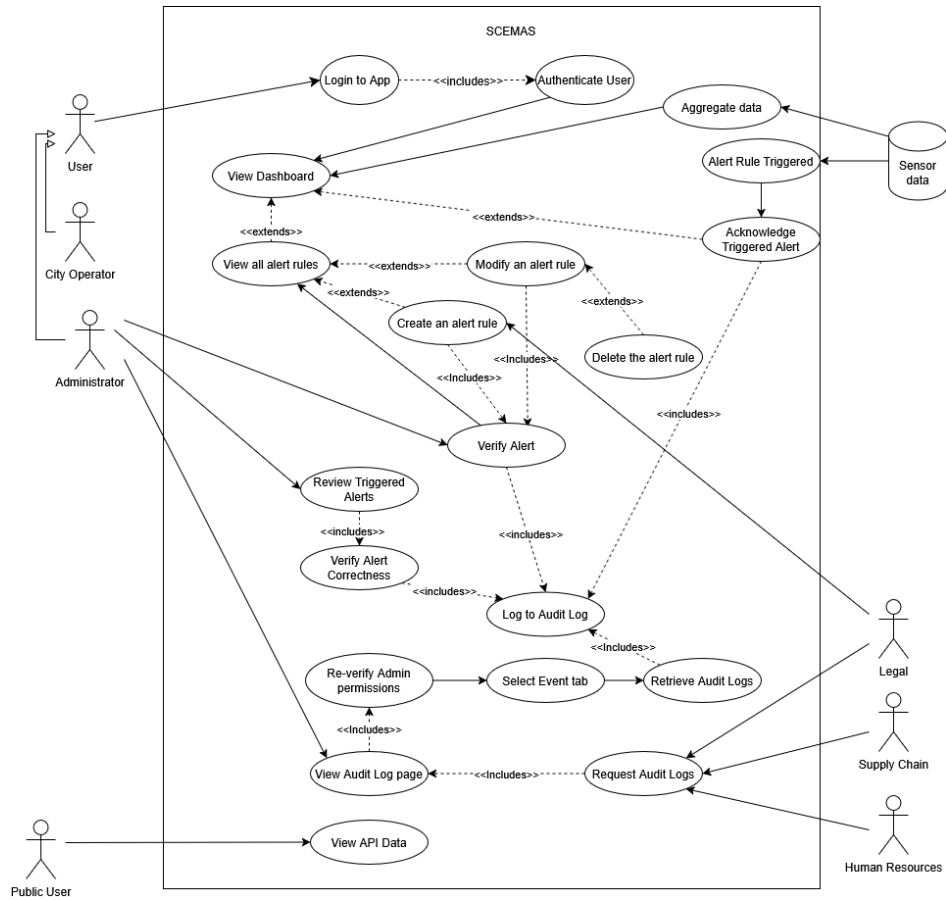


Figure 3: SCEMAS Use-Case Diagram

## 4 Highlights of Functional Requirements

The business events identified for the project are as follows:

- **BE1.** Creating an Alert Rule
- **BE2.** Editing Existing Alert Rule
- **BE3.** Sunlight is very high on the UV index
- **BE4.** View Dashboard
- **BE5.** Public API Request to Access Data
- **BE6.** Checking Audit log
- **BE7.** User authentication

The viewpoints identified for the project include:

- **VP1.** Administrator
- **VP2.** City Operator
- **VP3.** Public User
- **VP4.** Human Resources
- **VP5.** Supply Chain
- **VP6.** Legal

### **BE1.** Creating an Alert Rule

**Pre-Condition:** The user is logged in and has a valid account and is knowledgeable with the system. There is valid data flowing into the system from various IoT devices.

#### **VP1.** Administrator

- 5i. The administrator approves the alert.
- 6i. Administrator does not approve the alert.
  - 6i.1. Feedback box appears and admin gives feedback on alert.
  - 6i.2. Sends feedback back to operator.
- 7i.2. The administrator is contacted by the user about a failed alert setup.

#### **VP2.** City Operator

##### **Main Success Scenario:**

1. User clicks on create alert tool in the dashboard navbar.
2. System displays the alert configuration tool.
3. User selects relevant environmental metrics, geographic area, threshold value(s) and time-frame for this alert.
4. User selects the alert visibility, public facing vs internal.
5. User reviews the summary and sends it to administrator for approval.
6. Once approved the system checks if the alert is valid and saves to database.
7. System creates the alert and displays a success message on screen.

##### **Secondary Scenario:**

- 1i. User selects the wrong tool.
  - 1i.1. User selects go back.

- 3i. User enters invalid metric, or configuration.
  - 3i.1. System highlights the input box in red.
  - 3i.2. System prevents user from proceeding.
- 4i. If a public facing alert has been created the system checks if the alert is permitted within the law.
- 5i. User cancels the alert configuration.
  - 5i.1. User clicks cancel alert.
  - 5i.2. System discards all inputs and returns user to dashboard.
- 6i. Administrator does not approve the alert.
  - 6i.1. Feedback box appears and admin gives feedback on alert.
  - 6i.2. Sends feedback back to operator.
- 7i. System fails to create alert.
  - 7i.1. System displays error message on screen.
  - 7i.2. System provides option to contact the administrator.

**VP3.** Public User  
N/A

**VP4.** Human Resources  
N/A

**VP5.** Supply Chain  
N/A

**VP6.** Legal

- 4i. If a public facing alert has been created the system checks if the alert is permitted within the law.

#### **Global Scenario:**

**Precondition:** The user is logged in and has a valid account and is knowledgeable with the system. There is valid data flowing into the system from various IoT devices.

#### **Main Success Scenario:**

- 1. User clicks on create alert tool in the dashboard navbar.
- 2. System displays the alert configuration tool.
- 3. User selects relevant environmental metrics, geographic area, threshold value(s) and timeframe for this alert.
- 4. User selects the alert visibility, public facing vs internal.
- 5. User reviews the summary and sends to administrator for approval.
- 6. Once approved the system checks if the alert is valid and saves to database.
- 7. System creates the alert and display a success message on screen.

#### **Secondary Scenario:**

- 1i. User selects the wrong tool.
  - 1i.1. User selects go back.
- 3i. User enters invalid metric, or configuration.
  - 3i.1. System highlights the input box in red.
  - 3i.2. System prevents user from proceeding.
- 4i. If a public facing alert has been created the system checks if the alert is permitted within the law.
- 5i. User cancels the alert configuration.
  - 5i.1. User clicks cancel alert.
  - 5i.2. System discards all inputs and returns user to dashboard.

- 6i. Administrator does not approve the alert.
  - 6i.1. Feedback box appears and admin gives feedback on alert.
  - 6i.2. Sends feedback back to operator.
- 7i. System fails to create alert.
  - 7i.1. System displays error message on screen.
  - 7i.2. System provides option to contact the administrator.

## **BE2. Modify Existing Alert Rule**

**Pre-Condition:** At least one alert rule must already exist in the system database. The user must be logged in and is knowledgeable with the system and have a valid account with appropriate privileges.

### **VP1. Administrator**

- 6i. The administrator approves the alert.
- 7i. Administrator does not approve the alert.
  - 7i.1. Feedback box appears and admin gives feedback on alert.
  - 7i.2. Sends feedback back to operator.
- 8i.2. The administrator is contacted by the user about a failed alert setup.

### **VP2. City Operator**

#### **Main Success Scenario:**

- 1. User selects "Manage Alerts" section.
- 2. System fetches and displays a list of currently active and inactive alert rules.
- 3. User selects an alert rule to modify.
- 4. User edits relevant environmental metrics, geographic area, threshold value(s) and time-frame for chosen alert.
- 5. User chooses whether to keep current visibility or change alert visibility, public facing vs internal.
- 6. User reviews the summary and sends to administrator for approval.
- 7. Once approved the system checks if the alert is valid and saves to database.
- 8. System displays a success message indicating the rule has been updated.
- 9. System logs the changes to Audit log.

#### **Secondary Scenario:**

- 1i. User selects the wrong tool.
  - 1i.1. User selects go back.
- 3i. User cannot find the specific alert rule.
  - 3i.1. User uses the search/filter bar to locate the rule by ID or name.
- 4i. User enters an invalid metric or configuration.
  - 4i.1. System highlights the input box in red.
  - 4i.2. System prevents user from proceeding until the error is corrected.
- 6i. User cancels the edit.
  - 6i.1. User clicks "Cancel."
  - 6i.2. System discards all unsaved modifications and returns the user to the "Manage Alert Rules" list.
- 7i. Administrator does not approve the alert.
  - 7i.1. Feedback box appears and admin gives feedback on alert.
- 8i. System fails to update the Alert.
  - 8i.1. System displays a "Error" message.
  - 8i.2. System provides an option to contact the Administrator.

**VP3.** Public User

N/A

**VP4.** Human Resources

N/A

**VP5.** Supply Chain

- 9i. Reviews the audit log to ensure alert modifications align with current city safety protocols.

**VP6.** Legal

- 5i. If an alert has been edited to be a public-facing alert, the system checks if the alert is permitted within the law.

**Global Scenario:**

**Pre-Condition:** At least one alert rule must already exist in the system database. The user must be logged in and is knowledgeable with the system and have a valid account with appropriate privileges.

**Main Success Scenario:**

1. User selects "Manage Alerts" section.
2. System fetches and displays a list of currently active and inactive alert rules.
3. User selects an alert rule to modify.
4. User edits relevant environmental metrics, geographic area, threshold value(s) and timeframe for chosen alert.
5. User chooses whether to keep current visibility or change alert visibility, public facing vs internal.
6. User reviews the summary and sends to administrator for approval.
7. Once approved the system checks if the alert is valid and saves to database.
8. System displays a success message indicating the rule has been updated.
9. System logs the changes to Audit log.

**Secondary Scenario:**

- 1i. User selects the wrong tool.
  - 1i.1. User selects go back.
- 3i. User cannot find the specific alert rule.
  - 3i.1. User uses the search/filter bar to locate the rule by ID or name.
- 4i. User enters an invalid metric or configuration.
  - 4i.1. System highlights the input box in red.
  - 4i.2. System prevents user from proceeding until the error is corrected.
- 6i. User cancels the edit.
  - 6i.1. User clicks "Cancel."
  - 6i.2. System discards all unsaved modifications and returns the user to the "Manage Alert Rules" list.
- 7i. Administrator does not approve the alert.
  - 7i.1. Feedback box appears and admin gives feedback on alert.
- 8i. System fails to update the Alert.
  - 8i.1. System displays a "Error" message.
  - 8i.2. System provides an option to contact the Administrator.

**BE3.** Sunlight is very high on the UV index

**Pre-Condition:** SCEMAS is operational and is receiving sensor data. An Administrator has defined and activated an alert rule for high UV sunlight.

**VP1. Administrator**

**Main Success Scenario:**

1. The system detects incoming sensor data that violates an alert rule.
2. Alert is automatically generated.
3. Alert severity, timestamp, and affected zone are recorded.
4. The system notifies city operator dashboards in affected areas.
5. An Administrator reviews the triggered alert.
6. Administrator verifies that the alert triggered correctly.

**Secondary Scenario:**

- 2i. System fails to generate alert.
  - 2i.1. System logs the failure and notifies administrators.
- 6i. Administrator finds fault in alert rule configuration.
  - 6i.2. Administrator modifies the faulty alert configuration.

**VP2. City Operator**

**Main Success Scenario:**

1. An alert is triggered and appears on the operator dashboard.
2. The operator notices the alert.
3. The operator clicks on the notification to view alert details.
4. Alert status is updated to be acknowledged by the system.
5. The operator and viewing timestamp is logged by the system.

**Secondary Scenario:**

- 1i. Multiple alerts appear at once.
  - 1i.1. Dashboard prioritizes the alert with higher severity.
  - 1i.2. Continue to 2.
- 2i. No operator notices the alert within 5 minutes of alert trigger.
  - 2i.1. The system notifies administrators.
- 4i. Operator flags alert as false alarm.
  - 4i.1. System records the report and notifies administrators for review.

**VP3. Public User**

**Main Success Scenario:**

1. The alert rule is triggered by the system.
2. The system checks that the alert can be shown to the public.
3. The alert is found to be publicly visible.
4. The alert is set to be visible through the public API.
5. A user requests access to the alert.
6. The system sends the alert information to the user.

**Secondary Scenario:**

- 3i. The alert is not found to be publicly visible.
  - 3i.1. The alert is not set to be visible through the public API.
- 6i. The system delays sending information due to rate limiting.

**VP4. Human Resources**

N/A

**VP5. Supply Chain**

N/A

**VP6. Legal**

N/A

### **Global Scenario:**

**Pre-Condition:** SCEMAS is operational and is receiving sensor data. An Administrator has defined and activated an alert rule for high UV sunlight.

#### **Main Success Scenario:**

1. The system detects incoming sensor data that violates an alert rule.
2. Alert is automatically generated.
3. Alert severity, timestamp, and affected zone are recorded.
4. The system checks that the alert can be shown to the public.
5. The alert is found to be publicly visible.
6. The system notifies city operator dashboards in affected areas.
7. The alert appears on the operator dashboard.
8. A City Operator notices the alert.
9. The City Operator clicks on the notification to view alert details.
10. Alert status is updated to be acknowledged by the system.
11. The acknowledging City Operator and timestamp is logged by the system.
12. The alert is set to be visible through the public API.
13. A Public User requests access to the alert.
14. The system sends the alert information to the Public User.
15. An Administrator reviews the triggered alert.
16. The Administrator verifies that the alert triggered correctly.

#### **Secondary Scenario:**

- 2i. System fails to generate alert.
  - 2i.1. System logs the failure and notifies administrators.
- 5i. The alert is not found to be publicly visible.
  - 5i.1. The alert is not set to be visible through the public API.
- 7i. Multiple alerts appear at once.
  - 7i.1. Dashboard prioritizes the alert with higher severity.
  - 7i.2. Continue to 8.
- 8i. No operator notices the alert within 5 minutes of the alert trigger.
  - 8i.1. The system notifies Administrators.
- 10i. Operator flags alert as false alarm.
  - 10i.1. System records the report and notifies administrators for review.
- 14i. The system delays sending information due to rate limiting.
- 16i. Administrator finds fault in alert rule configuration.
  - 16i.2. Administrator modifies the faulty alert configuration.

### **BE4. View Dashboard**

**Pre-Condition:** SCEMAS is online, and the user has logged in with a verified account on SCEMAS with the correct level account permissions.

#### **VP1. Administrator**

##### **Main Success Scenario:**

1. Administrator accesses the SCEMAS application home screen.



2. System retrieves the user's last-used geographical area and sets it as the current geographical location.
3. System returns currently active nearby environmental and health warnings and displays them in a real-time dashboard map.
4. Administrator clicks the toggle bar to select "View Alerts".
5. System identifies the user as an Admin and loads all current environmental alerts to the dashboard.
6. Administrator monitors active alerts and system logs for consistency.
7. Administrator clicks log out.
8. System invalidates the session and clears the cache.
9. System invalidates the current session, clears cached map data and returns the user to the login screen.

## **VP2. City Operator**

### **Main Success Scenario:**

1. Operator accesses the SCEMAS application home screen.
2. System retrieves the user's last-used geographical area and sets it as the current geographical location.
3. System returns nearby currently active nearby environmental and health warnings and displays them in a real-time dashboard map.
4. Operator clicks on a warning on the map to view more information.
5. System opens a side panel showing all official information, including severity levels, specific metrics, and health precaution recommendations.
6. Operator views information and closes the side panel.
7. System closes the side panel and returns to the dashboard map.
8. Operator selects a data type (air quality, noise levels, temperature, and humidity) from the metric toggle menu.
9. System updates the map to display the selected data type on the real-time map with colour-coded environmental maps based on the severity levels of nearby regions.
10. Operator clicks log out.
11. System invalidates the current session, clears cached map data and returns the user to the login screen.

### **Secondary Scenario:**

- 2i. System fails to retrieve the user's last-used geographical area.
  - 2i.1. System attempts to fetch the last-used geographical area and receives a null response from the database.
  - 2i.2. System retrieves the user's default home location from the user's account information.
  - 2i.3. System sets the default area as the current location and loads the map.
  - 2i.4. Operator updates their geographical area by searching on the map.
- 3i. No Critical Alerts found.
  - 3i.1. System returns no critical alerts nearby.
  - 3i.2. System displays a "No Active Critical Alerts" message.
  - 3i.3. System continues to display the base map and environmental metrics side panel.
- 4i. Login Timed Out.
  - 4i.1. System notices over 15 minutes since the last user interaction.
  - 4i.2. System clears all sensitive environmental data from the screen and clears any unsaved changes.
  - 4i.3. Redirects the user to the login page.
- 5i. Critical Service Outage.
  - 5i.1. System detects that a microservice is down.

5i.2. System displays a message indicating the metric information is unavailable and not up-to-date.

5i.3. System continues to serve cached data for other functional metrics.

**VP3. Public User**

N/A

**VP4. Human Resources**

N/A

**VP5. Supply Chain**

N/A

**VP6. Legal**

N/A

**Global Scenario:**

**Pre-Condition:** SCEMAS is online, and the user has logged in with a verified account on SCEMAS with the correct level account permissions.

**Main Success Scenario:**

1. External User accesses the SCEMAS application home screen.
2. System retrieves the user's last-used geographical area and sets it as the current geographical location.
3. System returns nearby currently active environmental and health warnings and displays them in a real-time dashboard map.
4. External User clicks on a warning on the map to view more information.
5. System opens a side panel showing all official information, including severity levels, specific metrics, and health precaution recommendations.
6. External User views information and closes the side panel.
7. System closes the side panel and returns to the dashboard map.
8. External User selects a data type (air quality, noise levels, temperature, UV levels, and humidity) from the metric toggle menu.
9. System updates the map to display the selected data type on the real-time map with colour-coded environmental maps based on the severity levels of nearby regions.
10. External User clicks the toggle bar to select "View Alerts."
11. System identifies the user's account permissions and loads all current environmental alerts and system logs to the dashboard.
12. External User monitors active alerts and system logs for consistency.
13. External User clicks log out.
14. System invalidates the current session, clears cached map data, and returns the user to the login screen.

**Secondary Scenario:**

- 2i. System fails to retrieve the user's last-used geographical area.
  - 2i.1. System attempts to fetch the last-used geographical area and receives a null response from the database.
  - 2i.2. System retrieves the user's default home location from the user's account information.
  - 2i.3. System sets the default area as the current location and loads the map.
  - 2i.4. External User updates their geographical area by searching on the map.

- 3i. No Critical Alerts Found.
  - 3i.1. System returns no critical alerts nearby.
  - 3i.2. System displays a "No Active Critical Alerts" message.
  - 3i.3. System continues to display the base map and environmental metrics side panel.
- 4i. Login Timed Out.
  - 4i.1. System notices over 15 mins since the last user interaction.
  - 4i.2. System clears all sensitive environmental data from the screen and clears any unsaved changes.
  - 4i.3. System redirects the user to the login page.
- 5i. Critical Service Outage.
  - 5i.1. System detects that a microservice is down.
  - 5i.2. System displays a message indicating the metric information is unavailable and not up-to-date.
  - 5i.3. System continues to serve cached data for other functional metrics.

#### **BE5.** Public API Request to Access Data

**Pre-Condition:** The public user has read the API documentation. There is valid data flowing from the app to the API.

#### **VP1.** Administrator

##### **Secondary Scenario:**

- 3i. Administrator can manually block an IP address.
  - 3i.1. Admin can view past metrics from an IP address from the API request database to access if an IP needs to be blocked.
  - 3i.2. Admin can block an IP address by blacklisting it.

#### **VP2.** City Operator

N/A

#### **VP3.** Public User

**Pre-Condition:** SCEMAS is online, the API portal is configured by the administrator and valid aggregated data is in the system.

##### **Main Success Scenario:**

- 1. External user sends a HTTP request to the endpoint for environmental data.
- 2. System receives HTTP request, and processes it by checking if the request is asking for valid data.
- 3. System checks rate limiting counter from requesting IP address and the API request is counted in the database for metrics.
- 4. System retrieves data requested from the database.
- 5. System appends legal disclaimer to request.
- 6. Systems sends HTTP response back to user.
- 7. External user receives data.

##### **Secondary Scenario:**

- 2i. HTTP request is not valid.
  - 2i.1. System disregards HTTP request.
  - 2i.2. System responds with not found HTTP response.
- 3i. Rate limit exceeded.
  - 3i.1. System drops the request and does not process it.
  - 3i.2. System returns too many request HTTP response back to user.

#### **VP4.** Human Resources

N/A

**VP5.** Supply Chain  
N/A

**VP6.** Legal

**Secondary Scenario:**

- 5i. Disclaimer is provided in the HTTP response to protect the system and city from liability.

**Global Scenario:**

**Pre-Condition:** SCEMAS is online and the API portal is configured by admin. Valid aggregated data is in the system.

**Main Success Scenario:**

1. External user sends a HTTP request to the endpoint for environmental data.
2. System receives HTTP request, and processes it by checking if the request is asking for valid data.
3. System checks rate limiting counter from requesting IP address and the API request is counted in the database for metrics.
4. System retrieves data requested from the database.
5. System appends legal disclaimer to request.
6. Systems sends HTTP response back to user.
7. External user receives data.

**Secondary Scenario:**

- 2i. Http request is not valid.
  - 2i.1. System disregards HTTP request.
  - 2i.2. System responds with not found HTTP response.
- 3i. Rate limit exceeded.
  - 3i.1. System drops the request and does not process it.
  - 3i.2. System returns too many HTTP requests response back to user.

**BE6.** Checking Audit log

**Pre-Condition:** SCEMAS system is online and actively logging activity in the audit log, Administrator has logged into the system.

**VP1.** Administrator

**Main Success Scenario:**

1. Administrator clicks on the "Audit Log" button.
2. System re-verifies Administrator's permissions and grants them access to the audit logs.
3. System displays the audit log page.
4. Administrator clicks on the event tab they want to view records of (e.g. user management, alert triggers, device lifecycle changes).
5. System retrieves relevant audit logs and displays them on the page.
6. Administrator views audit logs of the specified event they were interested in.

**Secondary Scenario:**

- 2i. System denies user access to admin controls.
  - 2i.1. Administrator authentication is unsuccessful.
  - 2i.2. Checking audit log fails.
- 5i. No audit logs are present for the type of event being viewed.
  - 5i.1. System presents a message indicating there are no audit logs for the given event type.

- 5ii. System fails to retrieve audit logs.
  - 5ii.1. System presents an error message to notify user of failure.
  - 5ii.2. Checking audit log fails.

**VP2. City Operator**

**Secondary Scenario:**

- 5i. City operator's activity will appear within audit logs retrieved by administrator.

**VP3. Public User**

N/A

**VP4. Human Resources**

**Secondary Scenario:**

- 6i. Audit logs may be used to flag suspicious or inappropriate activities of city operators for reporting to HR.

**VP5. Supply Chain**

**Secondary Scenario:**

- 6i. Audit logs may be passed on to the supply chain to monitor the health of the system.

**VP6. Legal**

**Secondary Scenario:**

- 6i. Audit logs may be passed on to the legal department as documented evidence of all the system's activity if required.

**Global Scenario:**

**Pre-Condition:** SCEMAS system is online and actively logging activity in the audit log, Administrator has logged into the system.

**Main Success Scenario:**

1. Administrator clicks on the "Audit Log" button.
2. System re-verifies Administrator's permissions and grants them access to the audit logs.
3. System displays the audit log page.
4. Administrator clicks on the event tab they want to view records of (e.g. user management, alert triggers, device lifecycle changes).
5. System retrieves relevant audit logs and displays them on the page.
6. Administrator views audit logs of the specified event they were interested in.

**Secondary Scenario:**

- 2i. System denies user access to admin controls.
  - 2i.1. Administrator authentication is unsuccessful.
  - 2i.2. Checking audit log fails.
- 5i. No audit logs are present for the type of event being viewed.
  - 5i.1. System presents a message indicating there are no audit logs for the given event type.
- 5ii. System fails to retrieve audit logs.
  - 5ii.1. System presents an error message to notify user of failure.
  - 5ii.2. Checking audit log fails.

**BE7. User authentication**

**Pre-Condition:** The user is knowledgeable with the system. There is valid data flowing into the system from various IoT devices.

**VP1. Administrator**

- 4i.2. The administrator is prompted to review suspicious login.
- 4i.3. The administrator is contacted by the user to help with the login issue.

**VP2. City Operator**

**Main Success Scenario:**

- 1. User opens up the SCEMAS webapp in their browser.
- 2. System requires the user to log in and displays the username and password fields.
- 3. User enters username and password.
- 4. System authenticates the user.
- 5. System loads the dashboard and displays all options.

**Secondary Scenario:**

- 4i. System fails to authenticate the user.
  - 4i.1. System displays login error on screen.
  - 4i.2. System prompts the administrator to review suspicious log in.
  - 4i.3. System asks if the user wants to contact the administrator for help.

**VP3. Public User**

N/A

**VP4. Human Resources**

N/A

**VP5. Supply Chain**

N/A

**VP6. Legal**

- 11i. If a public facing alert has been created the system checks if the alert is permitted within the law.

**Global Scenario:**

**Pre-Condition:** The user is knowledgeable with the system. There is valid data flowing into the system from various IoT devices.

**Main Success Scenario:**

- 1. User opens up the SCEMAS webapp in their browser.
- 2. System requires the user to log in and displays the username and password fields.
- 3. User enters username and password.
- 4. System authenticates the user.
- 5. System loads the dashboard and displays all options.

**Secondary Scenario:**

- 4i. System fails to authenticate the user.
  - 4i.1. System displays login error on screen.
  - 4i.2. System prompts the administrator to review suspicious login.
  - 4i.3. System asks if the user wants to contact the administrator for help.

## 5 Non-Functional Requirements

### 5.1 Look and Feel Requirements

#### 5.1.1 Appearance Requirements

- LF-A1. The application should use a distinct colour scheme that is in adherence with the companies brand colours.  
**Rationale:** The application’s UI should be consistent and professional to establish a sense of trust in the product’s users [1].
- LF-A2. All text in the application should be at least 14-px in size.  
**Rationale:** The text in the application should follow font-size conventions to be large enough so it is easy to read for a wide range of individuals with varying vision levels [1].
- LF-A3. The background colours of pages should contrast with the interactive elements (alert notifications, buttons, icons) directly on top of them.  
**Rationale:** The interactive elements on the page should stand out from the background elements in order to indicate what can be interacted with on the page [1].

#### 5.1.2 Style Requirements

- LF-S1. The application’s interface should be built for a resolution of 1920x1080 and be compatible with various screen sizes across desktop devices.  
**Rationale:** As the interface will be used across many different desktop devices with varying screen sizes, the interface’s size must be adaptable so all users can view the application as intended. 1920x1080 has been selected as the default size as it is the most common resolution on modern desktop devices [2].
- LF-S2. The application’s interface must utilize modern conventions of professional software UI/UX to achieve a professional and minimalist design.  
**Rationale:** Adhering to commonly used UI/UX design principles will make the application look polished and professional, instilling trust in the user. Using well-known conventions of web design will also help the application to be more user-friendly [3].
- LF-S3. All main pages of the application interface should be accessible from one menu (e.g. a navigation bar).  
**Rationale:** The application should be simple to navigate, and users should not have to search through many menus to find crucial features [4].

### 5.2 Usability and Humanity Requirements

#### 5.2.1 Ease of Use Requirements

- UH-EOU1. The system must include a help menu for users, including a frequently asked questions page, and tutorial pages for how to navigate.  
**Rationale:** A frequently asked questions page will be a helpful resource for users by providing answers to repetitive inquiries [5].

#### 5.2.2 Personalization and Internationalization Requirements

- UH-PI1. The system must allow users to configure personal preferences in the account settings. Features that must be selectable are light or dark mode and accessibility routes highlighted on the maps for users with disabilities.  
**Rationale:** Enabling user-defined preferences improves the overall user experience and decreases task time [6]. The software will provide equitable utility for different users, making the application more inclusive.

### 5.2.3 Learning Requirements

UH-L1. The system must allow any third-party developer to successfully request and interpret environmental data within a two-hour time frame.

**Rationale:** A short learning curve ensures the API is intuitive and well-documented for the public. A short onboarding period encourages adoption of the REST API and improves developer productivity [4].

### 5.2.4 Understandability and Politeness Requirements

UH-UP1. The proficiency requirements for understanding any language being used in the system shall not exceed the B1 language proficiency level.

**Rationale:** The system application should be understandable to a wide range of people and a B1 proficiency level indicates the user can handle everyday situations and express opinions [7].

### 5.2.5 Accessibility Requirements

UH-A1. The system must have a setting to turn off the use of bright colours in the dashboard and map.

**Rationale:** Highly saturated colours are hard on the eyes and can cause visual fatigue when viewing for extended periods of time [8].

## 5.3 Performance Requirements

### 5.3.1 Speed and Latency Requirements

PR-SL1. The system shall ensure the latency of the data appearing on the City Operator dashboard does not exceed 1.5 seconds.

**Rationale:** Any delay in getting information to a city official directly increases the risk to public safety [4].

PR-SL2. The system shall authenticate a user and load their specific role-based dashboard in less than 3 seconds after the user clicks login.

**Rationale:** For City Operators, they would need to respond to events in a fast manner, as a delay in logging in during a crisis could delay critical decision-making [4].

PR-SL3. The Public API shall return the requested data within 1 second.

**Rationale:** Fast access to public data is essential for transparency and third-party developers. If the public-facing tools are slow, users are less likely to use the system for daily planning [9].

### 5.3.2 Safety-Critical Requirements

PR-SC1. N/A

### 5.3.3 Precision or Accuracy Requirements

PR-PA1. The system shall store and display all numerical data (e.g., temperature, humidity, and particulate matter) to at least two decimal places.

**Rationale:** High-resolution data is needed to identify subtle environmental trends. Rounding to the nearest whole number can hide small changes in events, which could prevent the alert of an event. This level of detail is necessary to detect the beginning trends before they cross the thresholds [4].

PR-PA2. The system shall represent sensor locations and alert markers on the geographical map with coordinate precision of at least 1 meter.

**Rationale:** As stated in Section 2.1, the system must allow officials to pinpoint exact locations of hazards. Low-precision coordinates could place an alert on the wrong area, leading to responders to the incorrect location [10].



### 5.3.4 Reliability and Availability Requirements

PR-RA1. The system shall maintain an availability of 99.99%, limiting the system updates and maintenance to no more than an hour a year.

**Rationale:** As the system is responsible for detecting hazards, it must be readily available. Despite the industry standard being an uptime of 99.99% [11], it is not attainable even for most services. Most cloud service providers' setups could achieve 99.99% availability [11].

PR-RA2. If the system or database experiences an unplanned failure, it shall automatically recover and restore core services within 120 seconds, without manual intervention.

**Rationale:** Disaster recovery aims to minimize business disruption by restoring service quickly after failures. This requirement sets an explicit RTO to reduce the operational "blind spot" during time-sensitive environmental incidents [12].

### 5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1. The system shall store all incoming sensor data in a local cache for up to 48 hours during a network or internet failure. Upon reconnection, the system shall automatically upload this data to the main database in chronological order.

**Rationale:** Smart-city sensor networks must tolerate intermittent connectivity, including extended outages during major emergencies. IoT architectures commonly support offline operation by using local caching so telemetry is not lost and can be reliably forwarded when connectivity returns [13].

PR-RFT2. The system shall continue to provide core functions, even if non-critical subsystems become unavailable.

**Rationale:** If one part of the app breaks, the whole system should not crash. Using "graceful degradation" ensures that essential tools remain available [14].

### 5.3.6 Capacity Requirements

PR-C1. The system shall support at least 50 concurrent City Operators performing dashboard actions simultaneously without data collision, inconsistent state, or system failure.

**Rationale:** Multiple city operators may operate the dashboard simultaneously. The system must handle concurrent access safely to prevent conflicting updates (such as simultaneous alert edits) and ensure operational continuity under peak load [15].

### 5.3.7 Scalability or Extensibility Requirements

PR-SE1. The system shall be able to scale to support deployment in multiple cities of varying population sizes, with increases in the number of sensors, monitored zones, and concurrent users, without requiring an architectural redesign.

**Rationale:** Designing for scalability allows the system to grow in users, sensors, and coverage while maintaining performance and avoiding major redesign [4].

### 5.3.8 Longevity Requirements

PR-L1. The system shall be designed with a minimum operational lifespan of at least 10 years.

**Rationale:** This system is a large investment for a city, so to provide a high return on investment and ensure safety, the software must remain viable and maintainable without requiring a complete replacement [4].

## 5.4 Operational and Environmental Requirements

### 5.4.1 Expected Physical Environment

OE-EPE1. N/A

### 5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1. The system must implement a read-only REST API endpoint allowing third-party and public access to non-sensitive aggregated environmental data.

**Rationale:** It is important for the system to be able to get important safety information out to the general public, and providing this API to other services will facilitate this communication [4].

OE-IA2. The system should be able to reliably receive and process data from designated sensors.

**Rationale:** The application must be able to receive sensor data in order to properly activate alerts and notify third party systems [4].

### 5.4.3 Productization Requirements

OE-P1. The application should be packaged in a zip file containing a README.

**Rationale:** The application should be quick to download and be easy to install and set up [4].

### 5.4.4 Release Requirements

OE-R1. The application should be compatible with the last 2 major versions of the Windows operating system.

**Rationale:** The system should be compatible with the most common desktop operating system worldwide [16].

## 5.5 Maintainability and Support Requirements

### 5.5.1 Maintenance Requirements

MS-M1. The system must be designed for modular component (microservices) upgrades independent of the entire system.

**Rationale:** A long maintenance window can result in missed alerts during critical environmental events. This will reduce risk and overall system downtime by completing individual software patches [17].

### 5.5.2 Supportability Requirements

MS-S1. The system must show application health metrics such as API latency, error rates, traffic and host saturation for all administrators.

**Rationale:** Administrators must be able to resolve data ingestion failures and offline issues immediately [18].

### 5.5.3 Adaptability Requirements

MS-A1. The system architecture must support the addition of new data ingestion for different environmental metric types.

**Rationale:** Environments are constantly changing and there will be new climate trends to be monitored. The system must be sustainable long-term and support easy additions without changing core logic [19].

## 5.6 Security Requirements

### 5.6.1 Access Requirements

SR-AC1. The system should only allow administrators to modify authenticated users and alert rules.

**Rationale:** There should be a way to add, manage, and delete users so that city operators can be successfully authenticated without public users having the same abilities. A Role-Based Access Control model [20] fits this need by defining a hierarchy of authentication levels through various roles.

- SR-AC2. The system should only allow city operators and system administrators to view the dashboard.  
**Rationale:** Public users should not be permitted to access all data through the system dashboard since this could reveal sensitive data to the public. A Role-Based Access Control model [20] fits this need by defining a hierarchy of authentication levels through various roles.

### 5.6.2 Integrity Requirements

- SR-INT1. Data should be encrypted while in transit.  
**Rationale:** Sensitive sensor data could be read if not encrypted leading to potential security risks and data leaks [21]. By keeping data encrypted, users and clients can trust the product to keep their data safe which will improve trust.

### 5.6.3 Privacy Requirements

- SR-P1. The system shall not collect, process, or store any personally identifiable information (PII).  
**Rationale:** This will ensure compliance with privacy regulations while reducing legal, ethical, and security risks [21].
- SR-P2. The public API shall only expose non-sensitive environmental data aggregated by city zone.  
**Rationale:** Revealing precise sensor data to the public could reveal the location of sensors used in the city [21]. Giving aggregated data balances transparency with the security of the system.

### 5.6.4 Audit Requirements

- SR-AU1. The system shall maintain an immutable log of all significant events, including alert triggers, alert acknowledgement and alert rule modifications.  
**Rationale:** Maintaining an unmodifiable log of all events can be used to audit the system and to track administrative changes. This follows the guidelines for cybersecurity set out by the Government of Canada [21].

### 5.6.5 Immunity Requirements

- SR-IM1. Public-facing interfaces shall enforce rate limits to protect against denial-of-service attacks and excessive data scraping.  
**Rationale:** Limiting the rate public users can access the API should reduce the strain on servers and limit the risk of a successful denial-of-service attack. This method of resisting denial-of-service attacks is outlined in ITSG-33 (Security Control SC-5) [21].

## 5.7 Cultural and Political Requirements

### 5.7.1 Cultural Requirements

- CP-C1. The system shall support all official languages of the country it is operating in.  
**Rationale:** National language laws state that for government systems all official languages must be available [22].
- CP-C2. The app and system will not use any offensive or hurtful language towards any individual or people group based on their; citizenship, gender, sexual orientation, religion, political views, disability status, or any other common identifier.  
**Rationale:** Hate speech laws must be followed and all users should feel safe when using the app [23].

### 5.7.2 Political Requirements

- CP-P1. The public facing app should not use fear inducing terms when describing alerts. Only scientifically backed terms will be used.  
**Rationale:** To not create panic or fear when a fire or pollution alert is out [24].

## 5.8 Legal Requirements

### 5.8.1 Compliance Requirements

- LR-COMP1. The app will not process any raw audio data. IoT devices will process audio and never transmit raw audio streams only sound levels.  
**Rationale:** Live audio streaming without consent is wiretapping under the Personal Information Protection and Electronic Documents Act and the Criminal Code of Canada Section 184 [25] [26].
- LR-COMP2. The app’s public facing app will meet the AODA accessibility requirements.  
**Rationale:** The app must meet the accessibility requirements by law [27].
- LR-COMP3. All app data will be stored on servers inside the country of origin.  
**Rationale:** This is the law on how municipalities have to collect and store data [27].
- LR-COMP4. The app will contain immutable audit logs and have frequent backups all while being secure for a minimum of 7 years.  
**Rationale:** The law states municipalities have to retain records in a secure manner to fulfill freedom of information requests and audits [28].
- LR-COMP5. Environmental action taken by a user in the dashboard when responding to an event has to be legal.  
**Rationale:** All environmental actions regarding pollution, toxicity, and waste, need to follow the Canadian Environmental Protection Act [29].

### 5.8.2 Standards Requirements

- LR-STD1. The app will follow the W3C WCAG 2.1 Accessibility Guidelines.  
**Rationale:** This is to meet AODA web standards [30].

## 6 Innovative Feature

The chosen innovative feature we will be implementing is a public interface for our SCEMAS application that will allow any public user to view dashboards which will visualize all of the publicly available environmental metrics available from the REST API. This feature will benefit the product by increasing the number of users for our application. Since the REST API will be mainly used by third-party developers, we want to allow regular public users to also view and be informed of environmental issues. We have also come up with other innovative features, as listed below:

- Provide access to external resources that guide users on how to navigate specific environmental concerns whenever a warning is issued.
- The dashboard map uses highlighted areas to emphasize zones that require extra precaution, such as school zones and residential neighbourhoods.
- An integrated RAG (Retrieval-Augmented Generation) model that leverages city documentation, such as covering water treatment, waste management, snow removal, and emergency services, to provide intelligent suggestions. Users can review these insights before choosing to take action via dedicated buttons, such as calling 911 or routing the issue to the appropriate department.
- The system generates insights based on real-time weather data. For example, it can identify heavy snow accumulation and automatically recalculate and transmit optimized routes to snow plow operators.
- Administrators can utilize biometric authentication, including fingerprint and retinal scans.
- Rather than simply displaying current sensor data, the platform predicts potential risks by analyzing trends. If particulate matter concentration begins to rise rapidly toward the  $100 \mu\text{g}/\text{m}^3$  threshold, the system will preemptively alert operators before the limit is officially breached.

- Notification settings allow organizations to assign specific employees to monitor particular alert types, such as air quality, or to oversee designated geographic regions.
- The interface fully supports both light and dark modes to accommodate different lighting environments and user preferences.

## A Division of Labour

This sheet indicates the contributions of each team member and is signed by all team members.

Khan, Saqib

- Introduction
- System Diagram
- Use-case Diagram as a group
- 2.4 Constraints
- Brainstorming requirements, business events
- BE2. Modify Existing Alert Rule
- 5 Non-Functional requirements discussion
- 5.1 Look and Feel Requirements contribution
- 5.3 Performance Requirements
- Discussed Innovative feature



Abdullah, Suzanne

- 2.3 User Characteristics
- System Diagram contribution
- Use-case Diagram with group
- BE6. Checking Audit Log
- 5.1 Look and Feel Requirements
- 5.4 Operational and Environmental Requirements
- Innovative feature main idea
- Brainstorming requirements, business events



Olejniczak, David

- 1.4 References + formatting references in Non-Functional Requirements

- 2.1 Product Perspective Paragraph
- Use-case Diagram as a group
- 2.4 Assumptions
- BE1. Creating an Alert Rule
- BE5. Public API Request to Access Data
- BE7. User authentication
- 5.7 Cultural and Political Requirements
- 5.8 Legal Requirements
- Brainstormed Innovative feature with group

*David Olejniczak*

Lai, Vanessa

- 1.2 Scope
- BE4. View Dashboard
- 5.2 Usability and Humanity Requirements
- 5.5 Maintainability and Support Requirements
- State Diagram
- Use-case Diagram as a group
- Brainstormed Innovative feature with group
- 6. Innovative Feature

*Vanessa Lai*

Buehlmann, Lukas

- 1.5 Overview
- 2.2 Product Functions
- Use-case Diagram as a group
- BE3. Sunlight is very high on the UV index
- 5.6 Security Requirements
- Brainstormed Innovative Ideas with group
- Reviewed and edited final document

*L Buehlmann*