Deliverable #1 Template : Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

Tutorial Number: T02
Group Number: G3
Group Members:

- Group Member Name (as listed in Avenue)
- You do not need to use student #s or macid (keep those private).

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 <u>underline</u>
- 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 document will provide an overview of the software requirements for SCEMAS. This document covers the purpose and objectives of SCEMAS, the scope of the system, characteristics of its intended users, product requirements, and an overview of the project structure.

1.1 Purpose

- The goal of this document is to define the software requirements, key functionalities, and user interactions for SCEMAS.
- It is intended for internal project stakeholders such as developers, project managers, system architects, and other concerned parties. No prior familiarity with SCEMAS is necessary to understand this specification.

1.2 Scope

- The product to be developed is SCEMAS, which contains many integrated components. This includes IoT data ingestion, database for time-series and geospatial data, aggregation engine, rule based alert engine, operator dashboard, and a public REST API.
- The IoT data ingestion is responsible for handling telemetry ingestion and validating it for correct formatting, compliance with schemas, and ensuring its within expected value ranges. The database will store the data validated by the IoT ingestion, and will be optimized for geospatial and time-series data. The aggregation engine will calculate various metrics such as 5-minute average, hourly maximums, and many more. The rule based alert engine will have administrator defined thresholds. It will compare incoming sensor data to the threshold to alert and log as needed. The operator dashboard must implement RBAC and display real time information from sensors, visualizations, and alert tools. The public REST API will provide the aggregated, non-sensitive data for public and third-party consumption
- SCEMAS continuously monitors air quality, noise levels, temperature, and humidity. Using this data, the system aims to improve urban environmental oversight, improve alerts during environmental events.

1.3 Definitions, Acronyms, and Abbreviations

- IoT - Internet of Things, a network of connected devices that transmit and exchange data.
- RBAC - Role Based Access Control, restricts access based on user roles.
- REST API - Representational State Transfer Application Programming Interface, a standard for web service communication.
- SCEMAS - Smart City Environmental Monitoring & Alert System.

1.4 References

1. "Deliverable #1 Template: Software Requirements Sepecification (SRS) Template," [Online]. https://avenue.cllmcmaste
[Accessed: 2026-02-05].

1.5 Overview

- Section 2 discusses the overall description for SCEMAS. This includes product perspectives, functions, user characteristics, constraints, assumptions and dependencies. Section 3 provies a use case diagram for the system which demonstrates typical interactions with the system, telemetry ingestion, and more. Section 4 describes the specific requirements for SCEMAS, going over the process in detail, covering events the system must handle, and how viewpoints interact with the system. Section 5 describes the specific non functional requirements, covering aesthetics, usability, performance, operations, security, and more.

2 Overall Product Description

2.1 Product Perspective

- Put the product into perspective with other related products, i.e., context
- If the product is independent and totally self-contained, it should be stated here
- If the SRS defines a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to the functionality of the software being developed. Identify interfaces between that larger system and the software to be developed.
- A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful

The Smart City Environmental Monitoring & Alert System (SCEMAS) is a cloud-native IoT software platform. Unlike systems such as SimpliCity, which focus on supporting city services, internal business process, and citizen engagement strategies, SCEMAS specifically collects, validates, stores, and analyzes environmental teleemtry to support real-time uran environmental monitoring and decision making.

SCEMAS is self-contained but interacts with multiple external systems. A network of IoT sensors collect environmental data and communicate them to SCEMAS. The system validates this incoming data, performs real-time aggregation, and calculates metrics to provide actionable insights into environmental conditions. Based on alert rules defined by administrators, SCEMAS triggers alerts upon detecting rule violations, logs all events, and notifies any subscribed external systems.

The system provides tailored interfaces for city operators and the public. City operators access a secure dashboard with real-time environmental metrics, system health, and active alerts. Public users access aggregated, non-sensitive environmental data through a simplified dashboard.

The system will interact with multiple external systems. SCEMAS will notify any subscribed external systems to notify them of any alert events and will also interface with various IoT sensors to collect environmental data.

2.2 Product Functions

- Provide a *summary* of the major functions that the software will perform.
  - **Example**: An SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.
- Functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time
- Present the functions in a list format - each item should be one function, with a brief description of it
- Textual or graphical methods can be used to show the different functions and their relationships
  - Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables

2.3 User Characteristics

- Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise
- Since there will be many users, you may wish to divide into different user types or personas

2.4 Constraints

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

2.5 Assumptions and Dependencies

- List any assumptions you made in interpreting what the software being developed is aiming to achieve
- List any other assumptions you made that, if it fails to hold, could require you to change the requirements
  - **Example**: An assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

2.6 Apportioning of Requirements

- Identify requirements that may be delayed until future versions of the system

3 Use Case Diagram

- Provide the use case diagram for the system being developed.
- You do not need to provide the textual description of any of the use cases here (these will be specified under "Highlights of Functional Requirements").

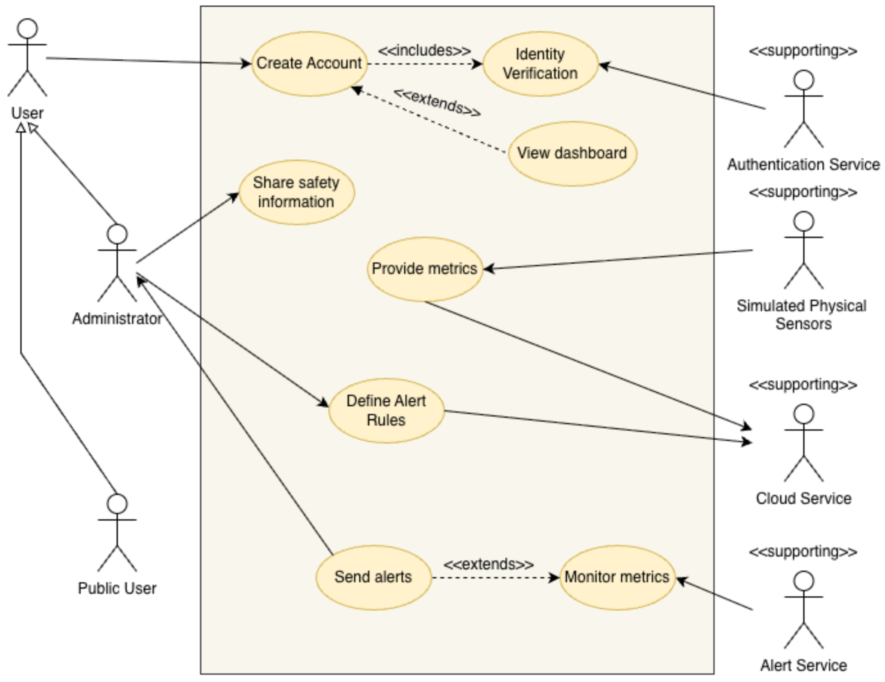

Figure 1: Use Case diagram for the 3A04 system being developed.

4 Highlights of Functional Requirements

- Specify all use cases (or other scenarios triggered by other events), organized by Business Event.
- For each Business Event, show the scenario from every Viewpoint. You should have the same set of Viewpoints across all Business Events. If a Viewpoint doesn't participate, write N/A so we know you considered it still. You can choose how to present this - keep in mind it should be easy to follow.
- At the end, combine them all into a Global Scenario.
- Your focus should be on what the system needs to do, not how to do it. Specify it in enough detail that it clearly specifies what needs to be accomplished, but not so detailed that you start programming or making design decisions.
- Keep the length of each use case (Global Scenario) manageable. If it's getting too long, split into sub-cases.
- You are *not* specifying a complete and consistent set of functional requirements here. (i.e. you are providing them in the form of use cases/global scenarios, not a refined list). For the purpose of this project, you do not need to reduce them to a list; the global scenarios format is all you need.
- Red text below is just to highlight where you need to insert a scenario - don't actually write it all in red.

Main Business Events: List out all the main business events you are presenting. If you sub-divided into smaller ones, you don't need to include the smaller ones in this list.

Viewpoints: List out all the viewpoints you will be considering.

Interpretation: Specify any liberties you took in interpreting business events, if necessary.

BE1. Business Event Name #1

VP1. Viewpoint Name #1
Insert Scenario Here

VP2. Viewpoint Name #2
Insert Scenario Here

Global Scenario:
Insert Scenario Here

BE2. Business Event Name #2

VP1. Viewpoint Name #1
Insert Scenario Here

VP2. Viewpoint Name #2
Insert Scenario Here

Global Scenario:
Insert Scenario Here

5 Non-Functional Requirements

- For each non-functional requirement, provide a justification/rationale for it.
  Example:
  SC1. *The device should not explode in a customer's pocket.*
  Rationale: Other companies have had issues with the batteries they used in their phones randomly exploding [insert citation]. This causes a safety issue, as the phone is often carried in a person's hand or pocket.
- If you need to make a guess because you couldn't really talk to stakeholders, you can say "We imagined stakeholders would want...because..."
- Each requirement should have a unique label/number for it.
- In the list below, if a particular section doesn't apply, just write N/A so we know you considered it.

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1.

5.1.2 Style Requirements

LF-S1.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1.

5.2.2 Personalization and Internationalization Requirements

UH-PI1.

5.2.3 Learning Requirements

UH-L1.

5.2.4 Understandability and Politeness Requirements

UH-UP1.

5.2.5 Accessibility Requirements

UH-A1.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1.

5.3.2 Safety-Critical Requirements

PR-SC1.

5.3.3 Precision or Accuracy Requirements

PR-PA1.

5.3.4 Reliability and Availability Requirements

PR-RA1.

5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1.

5.3.6 Capacity Requirements

PR-C1.

5.3.7 Scalability or Extensibility Requirements

PR-SE1.

5.3.8 Longevity Requirements

PR-L1.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE-EPE1.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1.

5.4.3 Productization Requirements

OE-P1.

5.4.4 Release Requirements

OE-R1.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1.

5.5.2 Supportability Requirements

MS-S1.

5.5.3 Adaptability Requirements

MS-A1.

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1.

5.6.2 Integrity Requirements

SR-INT1.

5.6.3 Privacy Requirements

SR-P1.

5.6.4 Audit Requirements

SR-AU1.

5.6.5 Immunity Requirements

SR-IM1.

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP-C1.

5.7.2 Political Requirements

CP-P1.

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1.

5.8.2 Standards Requirements

LR-STD1.

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.