Software Requirements Specification

For

Real-Time Factory Air Condition Monitoring for Improved Health

Version 1.0 approved.

Prepared by BSE25-5

Group 5

17th October 2024

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Revision History

Name	Date	Reason For Changes	Version
BSE25-5	17th/10/2024	Creation of the first draft of the SRS, covering all primary sections and initial requirements.	1.0
BSE25-5	2nd/11/2024	Added new features, clarified requirements, or made updates based on stakeholder feedback.	1.1

1. Introduction

1.1 Purpose

This document outlines the software requirements for the factory's environmental air quality monitoring system, version 1.0, specifically designed to enhance workplace conditions by tracking airborne pollutants in real-time and alerting users when conditions exceed permissible thresholds. This Software Requirements Specification (SRS) defines the monitoring system's scope, features, and functionalities, aligning it with organizational goals related to workplace safety, productivity, and regulatory compliance. The system will monitor critical airborne pollutants, including carbon monoxide (CO), carbon dioxide (CO_2), volatile organic compounds (VOCs), nitrogen dioxide (NO_2), particulate matter (PM2.5 and PM10), and temperature and humidity, to protect workers from exposure to unsafe environmental conditions and support compliance effort.[1]

Increased levels of pollutants, even by increments as small as 10 $\mu g/m^3$, have been associated with a 3- to 4-fold increase in respiratory complications, highlighting the importance of proactive monitoring in maintaining a safe and productive work environment.

In addition to air pollutants, the system will track dust levels and hazardous airborne vapors, such as ammonia (NH_3) and other chemical vapors, which pose specific risks in industrial environments. By offering a comprehensive overview of these airborne contaminants, the system enables factory personnel to receive timely alerts and access actionable data for mitigating risks, ensuring air quality, and optimizing the factory environment for enhanced productivity and equipment longevity.

1.2 Document Conventions

This Software Requirements Specification (SRS) document adheres to specific standards and typographical conventions to ensure clarity, consistency, and ease of navigation. The following conventions are used throughout the document:

Table 1 Document Conventions

Element	**Convention**
Font	The primary text of the document is written in Times**, 12 pt**, for easy readability.
Headings	Section and subsection headings follow a hierarchical structure with **bold formatting** for clarity: • **Main Headings (e.g., 1, 2, 3)**: Times, 18 pt, Bold • **Subheadings (e.g., 1.1, 2.1)**: Times, 14 pt, Bold • **Sub-subheadings (e.g., 1.1.1, 2.1.1)**: Arial, 12 pt, Bold
Requirements Statements	Every requirement is uniquely identified with an ID (e.g., REQ-1, REQ-2) and is written in **bold**, followed by the description in **regular text**.
Prioritization	Priorities for higher-level requirements are inherited by detailed requirements unless explicitly stated otherwise.
Terminology	Important terms and definitions are **italicized** to indicate that they are defined in the glossary section.

Highlighting	• **Bold**: Used for important terms, definitions, and section titles. • **Italicized Text**: Emphasizes special terms or concepts. • **Underlined Text**: Used for clickable links or references.

1.2.1 Document Structure Conventions

- Each requirement statement includes a unique identifier (e.g., REQ-001) for traceability.
- In cases where the priority of a requirement is critical (e.g., **High**, **Medium**, **Low**), it should be explicitly mentioned next to the requirement.

1.3 Intended Audience and Reading Suggestions

The Software Requirements Specification (SRS) document for the factory's environmental air quality monitoring system is crafted to address the diverse needs of stakeholders involved in the development, deployment, and operational phases of the project. This document delineates technical requirements, user interface designs, and non-functional aspects, ensuring that all project participants have a clear understanding of the system's objectives and functionalities. Each stakeholder is encouraged to concentrate on the sections pertinent to their role, facilitating efficient reading and comprehension. The table below offers guidance on the intended audience for each section of the SRS, allowing stakeholders to easily locate and prioritize the information most relevant to their responsibilities.

Table 2 Stakeholder Reading Guide for SRS Document

Stakeholder	Role Description	Relevant Sections	Reading Priority
Developers	Responsible for coding, integrating, and implementing the system's functionalities	• Functional Requirements • Interface Requirements • Non-functional Requirements	Start with detailed **technical requirements** to understand the coding and integration tasks.
Project Managers	Oversee project timeline, resource allocation, and ensure project stays on track	• System Scope • Operating Environment • Project Constraints	Begin with **system overview** sections for project scope and timeline, then move to constraints.
Testers	Validate the system's functionality, performance, and compliance with requirements	• System Features • User Interface Design • Functional Requirements	Focus on **system features** and **user interaction models** to prepare test cases and validation.
System Operators	Responsible for the daily operations and monitoring	• User Interface Design	Review **UI design** and **operating environment**

	of the system in the factory environment	• Operating Environment • System Maintenance Guidelines	to understand system usage and maintenance.
End Users	Factory staff who should interact with the system for real-time monitoring of environmental conditions	• System Features • User Interface Design	Prioritize **user interaction models** and **core functionalities** to understand system capabilities.
Maintenance Team	Responsible for troubleshooting, updating, and ensuring system longevity	• System Maintenance Guidelines • Operating Environment • Hardware and Software Requirements	Start with **maintenance and environment** sections to understand upkeep needs and operating specs.
Business Analysts	Gather business requirements, define system objectives, and ensure alignment with organizational goals	• System Scope • Functional Requirements • Business Constraints	Review **system scope** and **functional requirements** to understand how the system supports goals.

This structured guide aims to streamline the SRS review process, helping each stakeholder navigate the document efficiently and enhancing overall project alignment.

1.4 Product Scope

The factory's environmental air quality monitoring system is a comprehensive hardware and software solution aimed at monitoring and enhancing the working environment in factory settings by providing real-time monitoring and alerts regarding air quality conditions. Its primary goal is to ensure a safe and healthy atmosphere by detecting and reporting fluctuations in air quality that could impact worker well-being and operational efficiency.

The hardware component consists of sensors strategically placed throughout the factory to monitor various parameters, including temperature, humidity, volatile organic compounds (VOCs), and pollutant levels. By delivering timely alerts and visual reports, this embedded solution facilitates proactive adjustments to ventilation and air conditioning systems, ensuring compliance with safety standards and fostering improved overall productivity [4]. This system aligns with corporate objectives of maintaining a safe workplace, minimizing downtime associated with poor air quality, and enhancing employee satisfaction by promoting a healthier working environment. Moreover, by optimizing air quality control, the system contributes to cost-saving strategies related to energy consumption and operational maintenance.

2. Overall Description

2.1 Product Perspective

The **Factory's Environmental Air Quality Monitoring System** is a cutting-edge, self-sufficient solution aimed at enhancing air quality in industrial settings. It offers real-time monitoring and alerts to tackle the concerns related to air quality and its effects on worker health, safety, and productivity. By leveraging ***IoT* sensors**, **cloud computing**, and **real-time alerts**, this system provides a proactive and comprehensive strategy for maintaining a safe factory environment.

2.1.1 Context

- **Response to Need**: This system aims to meet the increasing demand for indoor air quality monitoring in industrial settings, where air conditions can lead to health risks and reduced productivity[1].
- **Market Gap**: Current solutions predominantly target outdoor environments or provide limited monitoring for indoor spaces. This product addresses that gap by delivering advanced indoor air quality monitoring specifically tailored for factory environments[2].
- **Based on Insights**: Drawing on insights from outdoor air quality systems, which effectively monitor air quality, this product is uniquely designed for indoor use in factories.

2.1.2 Development Origin

The system shall be developed to tackle air quality issues specific to factory environments, focusing on reducing workers exposure to harmful pollutants and volatile organic compounds (VOCs)[3]. Unlike disjointed solutions, this product provides a comprehensive approach that integrates **real-time data collection**, **cloud-based processing**, and **automated alerts** to improve workplace safety.

2.1.3 Relationship to Existing Systems

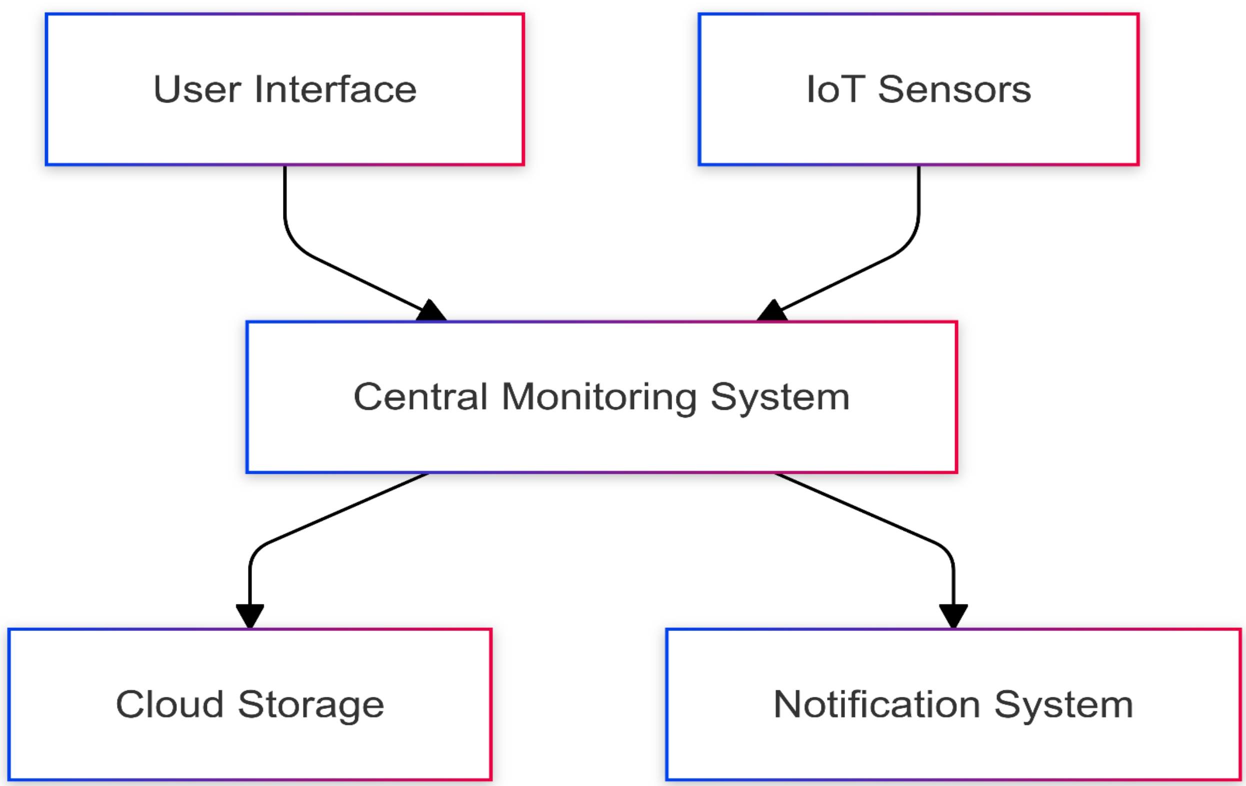
While other air quality monitoring systems are available, this product represents a **novel solution** rather than a mere upgrade. Its key enhancements include:

- ***IoT* Sensors**: These sensors monitor temperature, humidity, pollutants, and VOC levels across various zones within the factory[6].
- **Cloud Infrastructure**: This feature facilitates centralized data management, historical data analysis, and real-time reporting. The collected data is aggregated, analyzed, and stored on a cloud platform, facilitating scalable processing, remote access, and long-term data storage

- **User-Friendly Interfaces**: The system provides a web dashboard designed for operational staff to streamline management tasks.
- **Web Interfaces**: Users can access real-time reports and alerts through a web dashboard, enabling swift responses to potential air quality concerns.

This system represents a significant advancement in factory air quality monitoring by integrating **advanced hardware** and **software** to deliver actionable insights. It enhances workplace safety, operational efficiency, and regulatory compliance. The integration with external systems and **cloud technology** ensures scalability and flexibility to meet future needs.

Figure 1General Overview of the system architecture



2.2 Product Functions

The factory air quality monitoring hardware system shall be designed to perform several essential functions to ensure real-time oversight and management of air quality within the facility. The primary functions include:

- **Real-Time Air Quality Monitoring**: Continuously gather data on temperature, humidity, and VOC levels from a network of sensors distributed throughout the factory environment.
- **Alerts and Notifications**: Generate immediate alerts to inform users when air quality parameters exceed established safety thresholds.
- **Data Visualization**: Offer graphical representations of both historical and real-time air quality data through user-friendly dashboards, including trends and statistics. [4]
- **Report Generation**: Produce comprehensive reports on air quality over specified time periods, available for export for further analysis and compliance documentation.

These functions enable users to effectively monitor, analyze, and respond to air quality conditions within the factory, ensuring a safe and healthy working environment.

2.3 User Classes and Characteristics

The factory's environmental air quality monitoring system should be designed to cater to various user groups, each possessing distinct roles, responsibilities, and access levels. These user classes engage with the system in diverse ways, influenced by their technical expertise, frequency of use, and security requirements. By recognizing the specific characteristics and needs of each user class, the system is structured to provide relevant features and data access, thereby ensuring operational efficiency while upholding security protocols.

Table 3 User Classes and Characteristics

User Class	Characteristics	Functions Used	Security Level
Facility Managers	Moderate technical expertise; oversee air quality management; ensure compliance with safety standards	Access to monitoring dashboards, generate reports, configure alert settings, and adjust alert thresholds based on operational requirements	High-level access; limited control over system administration and technical settings
Workers	Moderate technical expertise; periodically observe sensors and report issues to maintenance staff	Monitor sensor LEDs for major changes and report issues	Low-level access
Maintenance Staff	Lower technical expertise: responsible for maintaining HVAC systems based on air quality data	Receives alerts for immediate action, performs system adjustments using data insights, but has restricted access to system configurations	Limited access; primarily for operational alerts and basic data insights
Executives and Decision Makers	Low technical expertise; infrequent users; focus on high-level trends for strategic decisions	View reports and summaries to evaluate factory air quality trends and conditions over time	View-only access; lowest security level

This structured approach ensures that each user class has access to pertinent features tailored to their specific roles within the factory, while safeguarding the system's integrity and data security. Facility managers possess the highest access to operational and alert settings, whereas executives are granted a more restricted view, concentrating on strategic insights. This balance promotes usability, security, and operational control across different user levels.

2.4 Operating Environment

The operating environment for the factory's environmental air quality monitoring system encompasses both software and hardware components, each with specific requirements and configurations. These elements are crucial for ensuring the seamless functionality and interoperability of the system, facilitating reliable data collection, storage, processing, and user interaction.

2.4.1 Software Operating Environment

The software environment defines the technological framework that supports the system's backend, frontend, and data management functionalities.[5] With a focus on security, compatibility, and efficient communication between components, the system's software setup ensures a responsive, secure, and user-friendly experience across all interfaces.

Table 4 Software Operating Environment

Component	**Specifications**
Operating System	• Backend services on Linux (Ubuntu 20.04 or later) • Web UI compatible with Chrome, Firefox, Edge, and Safari
Backend Technology	• Next.js for both server-side application logic and web frontend • MySQL for structured data management
Data Storage	• ThingSpeak for real-time sensor data storage • MySQL for user authentication and credentials
External APIs	• Integration with ThingSpeak IoT platform APIs to provide air quality data to the backend
Security	• Secure MySQL-based authentication • TLS 1.2+ encryption for all client-server communications

2.4.2 Hardware Operating Environment

The hardware environment comprises IoT devices, edge computing units, and network components, all designed to capture, process, and transmit air quality data in real-time. The hardware setup is optimized for durability, low power consumption, and reliable connectivity, even in constrained factory settings with limited internet access.

Table 5 Hardware Operating Environment

Component	Specifications
Hardware Platform	• IoT devices with air quality monitoring sensors, connected to network devices (GSM modules)
Gateway Devices	• Edge devices like Arduino Mega to aggregate and transmit sensor data
Sensor Specifications	• Sensors for temperature, humidity, CO2, VOCs, and PM2.5/PM10 detection • Designed for low-power and limited-resource environments
Communication Protocols	• Low-power protocols with GSM for long-range data transmission and connectivity
Cloud Servers	• Cloud environment hosted on Vercel for efficient data processing and accessibility
Power Supply & Durability	• Long-life 5V batteries to ensure uninterrupted sensor operation

This structured software and hardware setup ensures that each component of the factory air quality monitoring system is optimized for performance, reliability, and security. By leveraging modern technologies and efficient design, the system can operate effectively in an industrial environment, providing critical air quality insights for safer and healthier workspaces [2].

2.5 Design and Implementation Constraints

The development of the factory air quality monitoring system faces various design and implementation constraints stemming from corporate policies, regulatory compliance, hardware capabilities, and specific technology choices. These factors significantly influence the system's design, architecture, and implementation, and must be carefully considered to ensure the solution is compliant, efficient, and scalable.

2.5.1 Regulatory and Corporate Compliance

- **Data Privacy and Security Compliance**: The system must adhere to stringent data protection laws, such as the Data Protection and Privacy Regulations of 2021. This ensures that user data is collected, stored, and processed securely, in compliance with legal requirements. Protecting sensitive information is especially critical in environments where health-related monitoring takes place.
- **Audit Trails**: Maintaining audit trails for actions performed by users within the system is essential. In contexts where air quality monitoring has legal and health implications, audit logs

help ensure compliance, accountability, and traceability. Guaranteeing the integrity and accessibility of these logs aligns with the best practices outlined in industrial monitoring solutions.

2.5.2 Hardware and Environmental Constraints

- **Processing Power and Memory Limitations**: Due to the limited processing power and memory of IoT sensors, it is crucial to develop efficient, low-overhead software for edge data processing. Lightweight processing allows sensors to concentrate on capturing critical environmental data without overloading the system, aligning with findings from other research on real-time air monitoring systems in resource-constrained environments [1].
- **Network Bandwidth and Data Transmission**: The system relies on cellular data networks for data transmission, which often have limited bandwidth. Therefore, data transmitted from sensors must be streamlined and optimized to include only essential metrics, ensuring that the network can support real-time monitoring without delays [3].
- **Real-Time Data Processing**: Near real-time data processing is required to provide actionable insights into air quality. This demands a robust backend that can manage high-frequency data processing and ensure prompt updates, as demonstrated in other real-time IoT-based monitoring systems [4].
- **Environmental Durability**: IoT sensors and associated hardware must operate effectively in challenging factory environments characterized by varying temperatures, dust, and humidity. Ensuring durability is critical for reliable data collection over extended periods, even in potentially harsh conditions, as recommended in industrial air monitoring studies [2].

2.5.3 Technology, Tool, and Protocol Constraints

- **Data Analytics Tools**: The system's analytics capabilities may be limited to open-source or low-cost solutions due to budget constraints. This restriction limits the use of advanced commercial analytics tools, requiring the development of alternative methods for data visualization and trend analysis.
- **Concurrency and Parallel Operations**: The system must support multiple users accessing the dashboard simultaneously while processing real-time data from numerous IoT sensors. The backend needs to accommodate high concurrency and asynchronous operations to ensure system responsiveness, even under high loads, like approaches used in industrial air quality applications [6].

- **Programming Language and Framework**: The dashboard is developed using Next.js, chosen for its compatibility with responsive, real-time interfaces. This constraint necessitates that the development teamwork within the Next.js framework, which supports scalability and modularity.
- **Wireless Communication Protocols and Network Reliability**: Cellular data serves as the primary communication protocol with the ThingSpeak platform, which limits data transmission volume and may introduce latency. Reliable data communication is crucial for continuous monitoring and alerting, especially in industrial settings where timely responses are necessary [3].

2.5.4 Security and Access Control

- **Data Encryption**: To protect sensitive information, user credentials, and data must be encrypted, adhering to standards such as TLS 1.2 or higher. This is essential for ensuring data integrity and preventing unauthorized access.
- **Access Control**: Restricted access guarantees that sensitive air quality data and system functionalities are available only to authorized personnel, including industry managers and environmental compliance officers.

2.5.5 Design and Programming Standards

- **Modular Design Principles**: The system should adhere to modular design principles, allowing for independent development, testing, and maintenance of components such as the dashboard and backend services. Modularity enables easier updates and scalability, particularly for complex systems with multiple interacting components.
- **Code and Testing Standards**: The development process should follow ESLint coding standards to enhance consistency, readability, and maintainability. Comprehensive testing, including unit, integration, and end-to-end tests, should be conducted using frameworks such as Vitest to ensure functionality and reliability in the final deployment.

2.6 User Documentation

Thorough user documentation should be provided to ensure ease of use and proper functionality of the factory air quality monitoring system. This documentation should help various stakeholders understand the system's setup, usage, troubleshooting, and maintenance procedures.

2.6.1 Documentation Components

User Manual
A comprehensive guide that covers the installation, setup, and operational aspects of the system:

- **Installation and Configuration Instructions**: Step-by-step guidance on installing hardware and setting up the software.
- **Operational Guidelines**: Clear instructions for primary functionalities, including real-time monitoring, threshold settings, and report generation.
- **Troubleshooting and FAQs**: Solutions to common issues and answers to frequently asked questions to aid users in resolving problems independently.

Online Help Resources
Online resources offer quick, accessible support for system features:

- **Articles and Guides**: Blog posts and documentation providing insights into system features, troubleshooting, and air quality standards.
- **Visual Tutorials**: Step-by-step video tutorials to assist users in navigating the system and understanding air quality metrics [1].

Quick Start Guide
A concise guide designed to help users get the system up and running quickly, with a focus on essential functionalities only, ensuring a smooth onboarding process.

Delivery Formats
Documentation should be provided in multiple formats to accommodate user preferences:

- **PDF**: Downloadable guides for offline access.
- **HTML-Based Online Help**: A searchable online help center accessible directly from the software.
- **In-App Help**: Contextual help within the software for immediate access to instructions.

2.7 Assumptions and Dependencies

The following assumptions and dependencies may affect the requirements and development of the system as outlined in this Software Requirements Specification (SRS). These factors should be considered throughout the project lifecycle, as any changes or deviations could impact on the project's success.

2.7.1 Assumptions

- **Stable Operating Environment**: The operating system (OS), hardware, and related infrastructure (servers, networking) are assumed to remain stable and supported during development and deployment phases.
- **Availability of Third-Party Libraries**: It is assumed that all third-party or open-source libraries, frameworks, or APIs planned for use should be available, maintained, and compatible with the system's requirements.
- **User Access to Internet**: Users are assumed to have reliable internet access, as the system relies heavily on network communication for features such as real-time updates and cloud synchronization.
- **Adequate User Training**: It is assumed that users should possess a baseline level of technical proficiency or should undergo training to effectively use the system.
- **Availability of Development Tools**: The project assumes that required development tools, such as IDEs, version control systems, testing tools, and continuous integration servers, should be accessible and functional throughout the development lifecycle.
- **Access to Low-Cost Hardware**: It is assumed that there shall be a reliable supply of low-cost sensors available in the market.
- **Durability and Accuracy of Hardware**: The project assumes that the available hardware meets industry standards and is resilient enough to withstand the factory's harsh environment.

2.7.2 Dependencies

- **Third-Party Software Components**: The system shall depend on third-party software components, that is, libraries, APIs, or SaaS solutions. Changes in licensing, availability, or compatibility of these components could delay or impact project delivery.

- **Hardware and Network Infrastructure**: The performance and functionality of the system depend on the availability and reliability of the hardware infrastructure (e.g., servers, sensors, routers) and network connectivity.
- **External Data Services**: The system shall rely on external data services, which are cloud databases, ThingSpeak a cloud-based *IOT* platform. Any changes to these services, such as updates, downtime, or adjustments to pricing models, could affect system functionality.
- **Regulatory Compliance**: The project is dependent on adherence to relevant regulatory standards, which are data privacy laws and industry-specific regulations. Changes in these regulations may necessitate software modifications.
- **Development Team Availability**: The project assumes that the full development team should be available as scheduled. Any disruptions in staffing or availability could affect the project timeline.

These assumptions and dependencies are subject to ongoing review throughout the project lifecycle, and any significant changes should be addressed promptly to mitigate potential risks.

3. External Interface Requirements

This section outlines the necessary external interfaces for the system, detailing user interfaces, hardware interfaces, software interfaces, and communication protocols.

3.1 User Interfaces

The user interface (UI) of the system shall prioritize a consistent, intuitive, and user-friendly experience across all devices for the user dashboard.

Table 6 User Interfaces

Aspect	Description
Screen Layout	• **Navigation Bar**: Positioned at the top for easy access to core functions. • **Footer**: Contains legal information and contact details. • **Content Organization**: Clearly defined areas for content and user actions to optimize interaction and navigation.
Responsive Design	• The UI should fluidly adapt to various screen sizes, ensuring usability on both desktop and mobile devices.
Standard Components	• **Navigation Menu**: A consistent navigation bar across all screens (e.g., Home, Settings, Profile). • **Action Buttons**: Key actions such as Save, Cancel, and Submit should be uniformly positioned for ease of access. • **Error Messaging**: Errors should be prominently displayed, providing clear explanations and potential solutions to guide users in resolving issues.
Keyboard Shortcuts	• Common shortcuts should enhance efficiency, including: - Ctrl + S: Save - Ctrl + Z: Undo - Esc: Cancel
Mobile Interface	• The design should be touch-friendly, incorporating: **Navigation Drawer**: A collapsible menu for easy access to settings and options. **Larger Buttons**: Buttons optimized for touch interaction. **Swipe Gestures**: For seamless navigation and dismissing notifications.

3.2 Hardware Interfaces

The system shall integrate with various IoT sensors and gateways to facilitate data collection and transmission.

Table 7 Hardware Interfaces

Component	Description
Sensor Protocols	• Support for communication protocols to ensure low-power, efficient communication.
Data Transmission	• Sensors shall transmit data to the cloud using Ethernet or cellular networks.

3.3 Software Interfaces

The system shall connect to external software components through the following mechanisms:

Table 8 Software Interfaces

Interface Type	Description
Database	• MYSQL shall serve as the structured data storage solution, ensuring efficient data management.
Operating Systems	• Compatibility with Linux-based servers should be prioritized to leverage stability and performance.
External APIs	• Integration with platforms like ThingSpeak for enhanced data capabilities and analytics.

3.4 Communications Interfaces

The system shall implement robust communication protocols to ensure secure data transmission:

Table 9 Communication Interfaces

Communication Aspect	Description
Transmission Protocols	Support for HTTP/HTTPS for all data exchanges.
Encryption	All client-server interactions shall utilize TLS 1.2 or higher to ensure data security during transmission.
Alert Mechanisms	Notifications and alerts should be delivered through in-App Notifications For real-time user engagement. The LEDs and the buzzer shall also be used to notify the users.
Retry Mechanisms	The system should incorporate retry mechanisms for alerts and notifications in case of transmission failures, ensuring critical information is reliably communicated.

4. System Features

This section outlines the major functional capabilities of the system, focusing on features that enable users to effectively monitor and manage air quality in industrial environments. Each feature is described with **Use Case**, **Primary Actor**, **Preconditions**, **Normal Course**, **Exceptions**, **Post-conditions**, and **Functional Requirements**. The features are prioritized and rated for their impact, cost, and risk.

4.1 Real-time Air Quality Monitoring

4.1.1 Description and Priority

This feature should monitor and display real-time air quality data in industrial environments, capturing key pollutants such as **CO**, **VOCs**, **NO2**, and **PM2.5/PM10**. It should provide crucial information to factory managers and compliance officers about air quality conditions.

- **Priority**: High
- **Component Ratings**:
  - **Benefit**: 9 (Critical for worker safety)
  - **Penalty**: 8 (Failure could result in health risks)
  - **Cost**: 5 (Requires reliable sensor integration)
  - **Risk**: 4 (Minimal risk, but requires consistent network connectivity)

4.1.2 Stimulus/Response Sequences

Table 10 Stimulus/Response for Real-Time Air Quality Monitoring

User Action	System Response
1. IoT sensors collect air quality data in real-time	Data is transmitted to the central processing system
2. System processes the data	System displays pollutant levels (CO, VOCs, NO2, PM2.5/PM10) on the dashboard
3. Users monitor the dashboard	Real-time updates on air quality levels are displayed continuously

Exceptions:

- **2.1**: Data transmission fails due to network or sensor issues.
- **3.1**: Data received from the sensors is corrupted.
- **3.1.1**: Dashboard fails to update in real-time due to a system error.

4.1.3 Functional Requirements

Table 11 Functional Requirements for Real-Time Air Quality Monitoring

Requirement ID	Description
REQ-1	The system shall collect air quality data from IoT sensors.
REQ-2	The system shall process and display air quality data on a user interface in real-time.
REQ-3	The system shall update the dashboard in real-time with new data.
REQ-4	The system shall manage invalid sensor data by logging errors and notifying the administrator.

4.2 Analytics for Air Quality Trends

4.2.1 Description and Priority

This feature should analyze air quality trends based on historical data, enabling users to take initiative-taking measures to prevent poor air quality conditions. The analysis helps users mitigate potential health risks before they occur.

- **Priority**: Medium
- **Component Ratings**:
  - **Benefit**: 8 (Proactively improves air quality management)
  - **Penalty**: 6 (Lack of analysis data reduces initiative-taking capability)
  - **Cost**: 7 (Involves data analytics setup)
  - **Risk**: 4 (Low risk, but accuracy may vary)

4.2.2 Stimulus/Response Sequences

Table 12 Stimulus/Response for Predictive Analytics for Air Quality Trends

User Action	System Response
1. System collects and stores historical data	Data is stored in the system's database for future analysis
2. System analyzes trends	The dashboard displays air quality trends.
3. Users select specific time frames for analysis	The system provides more granular insights based on the selected time

Exceptions:

- **1.1**: Historical data is incomplete or corrupted.
- **2.1**: The system did not receive any data from the sensors.
- **2.1.1**: The Dashboard does not display any trends.

4.2.3 Functional Requirements

Table 13 Functional Requirements for Predictive Analytics for Air Quality Trends

Requirement ID	Description
REQ-5	The system shall store air quality data collected from the sensors.
REQ-6	The system shall analyze historical data to predict air quality trends for the next 24-48 hours.
REQ-7	The system shall display air quality trends on the dashboard.
REQ-8	The system shall notify users when predicted air quality is expected to exceed safety limits.

4.3 Real-time Alerts and Notifications

4.3.1 Description and Priority

This feature should provide real-time notifications to users when pollutant levels exceed predefined safety thresholds. Alerts are sent via the user's preferred communication channels (in-app notifications).

- **Priority**: High
- **Component Ratings**:
    - **Benefit**: 9 (Critical for worker safety)
    - **Penalty**: 8 (Failure could result in health risks)
    - **Cost**: 5 (Requires integration with communication channels)
    - **Risk**: 4 (Minimal risk, but requires reliable connectivity)

4.3.2 Stimulus/Response Sequences

Table 14 Stimulus/Response for Real-time Alerts and Notifications

User Action	System Response
1. Pollutant levels exceed safety thresholds	System triggers an alert in real-time
2. System sends alert through preferred channels	User receives alert via in-app notification
3. User acknowledges or dismisses the alert	System logs the response for future reference

Exceptions:

- **1.1**: The pollutant sensor provides incorrect data due to malfunction.
- **3.1**: The alert fails to send due to communication channel failure.
- **3.1.1**: The user fails to acknowledge the alert, prompting the system to retry.

4.3.3 Functional Requirements

Table 15 Functional Requirements for Real-time Alerts and Notifications

Requirement ID	Description
REQ-9	The system shall trigger alerts when pollutant levels exceed safety thresholds.
REQ-10	The system shall send alerts via in-app notifications based on user preferences.
REQ-11	The system shall log all sent alerts for future review.
REQ-12	The system shall retry sending alerts if the initial attempt fails.

4.4 Report Generation for Air Quality Data

4.4.1 Description and Priority

This feature generates detailed reports of historical air quality data for compliance audits or deeper analysis. The reports cover pollutants like **CO**, **VOCs**, **NO2**, and particulate matter.

- **Priority**: Medium
- **Component Ratings**:
    - **Benefit**: 8 (Important for compliance and audits)
    - **Penalty**: 6 (Failure could result in regulatory issues)
    - **Cost**: 4 (Requires data processing)
    - **Risk**: 3 (Minimal risk)

4.4.2 Stimulus/Response Sequences

Table 16 Stimulus/Response for Report Generation for Air Quality Data

User Action	System Response
1. User selects a time for the report	System retrieves and processes historical air quality data for that period
2. System generates the report in PDF/CSV	User downloads or views the report

Exceptions:

- **2.1**: The data logs for the selected period are incomplete.
- **3.1**: The report generation process encounters an error.
- **3.1.1**: The generated report is corrupted or cannot be opened.

4.4.3 Functional Requirements

Table 17 Functional Requirements for Report Generation for Air Quality Data

Requirement ID	Description
REQ-13	The system shall allow users to select time periods for report generation.
REQ-14	The system shall generate reports in PDF or CSV format.
REQ-15	The system shall store generated reports for future access.
REQ-16	The system shall notify the user when the report is ready for download.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The **Factory Air Quality Monitoring System** should meet rigorous performance requirements to ensure the effectiveness and reliability of its real-time capabilities. Key performance requirements include:

- **Real-Time Data Processing**: The system shall process sensor data in real-time, with a maximum latency of 3 seconds from the time data is captured by IoT sensors to when it is displayed on the dashboard.
- **Throughput**: The system should support the concurrent operation of up to 20 IoT sensors, each transmitting data every minute, without performance degradation.
- **Scalability**: The system should be able to scale seamlessly to manage additional sensors as the factory expands. This includes maintaining response times and processing speed even as the number of sensors increases.
- **System Uptime**: The system shall achieve an availability of **99.9%**, ensuring minimal downtime. Any downtime should not exceed 1 hour per month, accounting for scheduled maintenance.
- **Data Retrieval Time**: Historical data queries, such as air quality trends or report generation, should be completed within 2 seconds for a 30-day period and within 5 seconds for periods up to 6 months.

The system shall also be optimized for efficient memory usage and bandwidth management, ensuring that only essential data is transmitted over constrained network environments.

5.2 Safety Requirements

To protect the safety of both factory workers and equipment, the system should meet the following safety requirements:

- **Air Quality Thresholds**: The system shall trigger alerts when pollutant levels exceed **OSHA** and **EPA** safety thresholds for CO, NO_2, VOCs, and particulate matter [5].
- **Fail-Safe Mechanisms**: In the event of sensor or communication failure, the system shall trigger a default alert to notify factory management of potential air quality risks. This ensures that system malfunctions do not compromise worker safety.
- **Compliance with Regulatory Standards**: The system shall comply with relevant safety regulations such as **ISO 45001** for Occupational Health and Safety, ensuring that the air quality monitoring mechanisms adhere to legal safety requirements.
- **Auditing and Logging**: All safety alerts and system malfunctions should be logged and auditable to ensure that corrective actions are taken in the event of safety breaches. These logs should be stored securely for a minimum of **12 months**.

5.3 Security Requirements

Given the sensitivity of the air quality data and the need for secure operations, the following security measures are mandatory:

- **Data Encryption**: All data transmitted between IoT sensors, gateways, and the backend should be encrypted using **AES-256** encryption to prevent unauthorized interception.
- **Authentication**: Users accessing the system shall authenticate using credentials stored in MYSQL database ensuring secure access control.
- **Data Privacy**: The system should comply with data privacy regulations, including **The Data Protection and Privacy Regulations, 2021**. Personal data collected (e.g., from user interactions) should be stored securely and used only for operational purposes.
- **Incident Management**: In case of a security breach, the system shall follow a predefined incident response plan and notify administrators immediately.

5.4 Software Quality Attributes

- **Adaptability**:
  IoT sensors and gateways should support flexible deployment across different factory setups without requiring major configuration changes.
- **Availability**:
  Redundancy should be built into both hardware and software, ensuring that if a sensor, gateway, or server fails, backups should maintain **99.5% uptime**.
- **Interoperability**:
  Using **standard protocols** (*HTTP, MQTT*) and **data formats**, the system should easily integrate with **ThingSpeak** as the external platforms. **IoT sensors** should seamlessly communicate with cloud services for data aggregation and analysis.
- **Maintainability**:
  The system's **modular design** should allow hardware components like **sensors** and software features to be updated independently. ***CI/CD* pipelines** should ensure automated updates without disrupting system operations.
- **Testability**:
  Automated testing of both hardware and software should ensure at least **80% code coverage**. **Continuous testing** in the ***CI/CD* pipeline** identifies and resolves issues from sensor data collection to data visualization.
- **Usability**:
  The **user interface** should be designed for simplicity, with clear visualizations of air quality data from **sensors**. Intuitive dashboards should ensure minimal training for both technical and non-technical users.
- **Reliability**:
  A **retry mechanism** should ensure that sensor data transmission is dependable even during network failures. If a sensor or gateway fails, the system should log the issue and alert administrators for timely resolution.

5.5 Business Rules

The system should enforce several business rules to ensure consistent and secure operation:

- **User Roles**: Only **facility managers** and **maintenance staff** may monitor the air quality trends or manage the sensor network. Regular workers should have view-only access to air quality data.
- **Alert Escalation**: If air quality breaches persist for more than **5 minutes**, alerts should be escalated to upper management and safety officers, ensuring swift action is taken.
- **Data Retention**: Historical data should be stored for a minimum of **6 months**, enabling compliance with regulatory audits, and enabling long-term trend analysis.

Energy Efficiency: The system shall minimize energy consumption, particularly in sensor operations, by leveraging low-power communication protocols and optimizing data transmission intervals during non-critical hours.

6. References

1. Andarini, R. &. (2022). Monitoring indoor air quality for thermal comfort using Internet of Things. *Seventh International Conference on Informatics and Computing (ICIC)* (pp. 1-6). IEEE. doi:https://doi.org/10.1109/ICIC56845.2022.10006955
2. Hikmah, A. N. (2023). Sensor fusion with seven parameters for indoor air quality monitoring system. *10th International Conference on Electrical Engineering, Computer Science, and Informatics (EECSI)*. IEEE. doi:https://doi.org/10.1109/EECSI59885.2023.10295770
3. Islam, F. B.-S.-M. (2020). IoT-based HVAC monitoring system for smart factory. *International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. doi:https://doi.org/10.1109/ICTC49870.2020.9289249
4. Samal, A. S. (2023). Integrated IoT-based air quality monitoring and prediction system: A hybrid approach. *IEEE International Symposium on Smart Electronic Systems (iSES)*. IEEE. doi:https://doi.org/10.1109/iSES58672.2023.00099
5. Suryawanshi, R. G. (2024). IoT-based real-time environment monitoring and safety for factory workplace. *1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU)*. IEEE. doi:https://doi.org/10.1109/IC-CGU58078.2024.10530721
6. Thepudom, T. S. (2014). Indoor air quality monitoring system based on polymer/functionalized-SWNT gas sensors. *3rd Global Conference on Consumer Electronics (GCCE)*. IEEE. doi:https://doi.org/10.1109/GCCE.2014.7031331

Appendix A: Glossary

- **I*oT***: Internet of Things – a network of physical devices connected to the internet, collecting, and sharing data.
- ***HVAC***: Heating, Ventilation, and Air Conditioning systems.
- ***API***: Application Programming Interface – a set of rules that allow one system to communicate with another.
- ***MQTT***: Message Queuing Telemetry Transport – a lightweight messaging protocol for small sensors.
- **ThingSpeak**: An *IoT* platform used for monitoring and analyzing sensor data.

- ***CI/CD***: Continuous Integration/Continuous Deployment – a set of practices for automating software testing and deployment.

Appendix B: Analysis Models

- **Data Flow Diagram (DFD):** A DFD showing how data from *IoT* sensors is collected, processed in the cloud, and presented on the dashboard for user analysis.
- **Entity-Relationship Diagram (ERD):** An ERD outlining the relationships between key system entities, such as users, sensors, air quality data, and reports.

Appendix C: To Be Determined List

- ***TBD-1****: Additional sensor types that may be included in future system expansions (e.g., detecting more specific pollutants).*