# SMART AIR POLLUTION MONITORING SYSTEM

### Real Time Project Review-1

**Submitted in partial fulfilment of the requirements for the award of the Degree of**

**Bachelor of Technology (B. Tech) In**

## COMPUTER SCIENCE AND ENGINEERING

**INTERNET OF THINGS(CSO)**

**By**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | P.VAISHNAVI |   SUSMITHA | |  | | --- | | 22AG1A6958 |   22AG1A6913 |
| TARUN KUMAR | 22AG1A6955 |
| V.CHOWSHIK | 22AG1A6959 |

Under the Esteemed Guidance of

## INTERNAL & PROJECT GUIDE: MRS PRIYANKA

(Assistant Professor)



# Department of Computer Science and Engineering

***ACE ENGINEERING COLLEGE***

### An Autonomous Institution

**(NBA ACCREDITED B. TECH COURSES: EEE, ECE, MECH, CIVIL & CSE, ACCORDED NAAC ‘A’ GRADE)**

**(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana)**

**Ghatkesar , Hyderabad - 501 301**

### Contents

**INDEX** Page no

[Abstract 3](#_TOC_250017)

Feasibility Study

1. Introduction 4
   1. [Purpose 4](#_TOC_250016)
   2. [Scope 5](#_TOC_250015)
   3. [Intended Audience 6](#_TOC_250014)
   4. Definitions, Acronyms and Abbreviations 6
   5. [References 7](#_TOC_250013)
   6. [Overview 7](#_TOC_250012)
2. [Overall Description 7](#_TOC_250011)
   1. [User Interfaces 7](#_TOC_250010)
   2. [System Interfaces 8](#_TOC_250009)
   3. Constraints, Assumptions and Dependencies 8
   4. [User characteristics 9](#_TOC_250008)
3. [System Analysis 9](#_TOC_250007)
   1. Existing System 9
   2. [Proposed System 10](#_TOC_250006)
   3. [Requirements Specification](#_TOC_250005)
      1. [User Requirements 11](#_TOC_250004)
      2. [System Requirements 12](#_TOC_250003)
      3. [Domain Requirements 12](#_TOC_250002)
      4. [Functional Requirements 13](#_TOC_250001)
      5. [Non-Functional Requirements 14](#_TOC_250000)

# ABSTRACT

This paper introduces an IoT-based air pollution monitoring system leveraging ESP32 microcontrollers and gas sensors, including MQ135, MQ7, and MQ5, to provide comprehensive real-time air quality assessment.

The system deploys sensor nodes across diverse locations, continuously measuring key air quality parameters. The ESP32 microcontroller serves as the central processing unit, facilitating data collection, analysis, and transmission to a centralized database or cloud platform.

Specifically, the MQ135 sensor detects gases such as ammonia and nitrogen oxides, while the MQ7 sensor monitors carbon monoxide levels, and the MQ5 sensor identifies flammable gases like methane. Through low-power operation and efficient data transmission, the system ensures scalability and suitability for deployment in remote or off-grid areas. Stakeholders, including environmental agencies, city planners, researchers, and the general public, can access real-time air quality information through intuitive web-based dashboards. This IoT-based approach offers a cost-effective, scalable, and user-friendly solution for monitoring and managing air quality, thereby contributing to public health enhancement and environmental sustainability.

Top of Form

## FEASIBILITY STUDY 1.INTODUCTION

### Purpose

An IoT-based air pollution monitoring system utilizing ESP32 and gas sensors fulfills a multifaceted purpose in contemporary society. It offers real-time monitoring capabilities, enabling continuous assessment of air quality and thereby safeguarding public health by promptly identifying and addressing potential risks posed by harmful pollutants like particulate matter, carbon monoxide, sulfur dioxide, and nitrogen dioxide. Additionally, it fosters environmental consciousness by shedding light on air quality trends and pollution hotspots, empowering communities to advocate for cleaner air policies and take collective action to mitigate pollution sources. Serving as an early warning system for pollution events, such as industrial accidents or wildfires, it facilitates rapid response measures to minimize health impacts and environmental damage. Moreover, the data collected by the system supports urban planning decisions, aids in research and policy development, informs personal health management choices, and assists in compliance monitoring with air quality regulations. In essence, this IoT-based system acts as a cornerstone for promoting public health, environmental sustainability, and informed decision-making across various sectors of society.

Furthermore, the system serves as a crucial tool for urban planners and policymakers, providing essential data for designing sustainable cities and infrastructure that minimize pollution exposure and enhance quality of life for residents. Researchers leverage the collected data to conduct in-depth studies on air pollution trends, assess the efficacy of pollution control measures, and develop evidence-based policies to improve air quality and mitigate health risks. At the individual level, access to real-time air quality information empowers people to make informed decisions about outdoor activities, exercise routines, and commuting routes, thereby reducing personal exposure to harmful pollutants and promoting overall well-being. Additionally, industries can utilize the system for self-monitoring purposes, ensuring compliance with air quality regulations and emission standards while fostering corporate responsibility and environmental stewardship.

1. Top of Form

### Scope:

1. **Project Overview:** Provide a concise yet informative summary of the project's goals, emphasizing the development of an IoT-based air pollution monitoring system using ESP32. Highlight its significance in addressing environmental concerns and improving public health.
2. **Hardware Selection**: Explain the rationale behind choosing ESP32 as the central microcontroller due to its capabilities in wireless communication and data processing. Detail the selection process for sensors, considering factors such as accuracy, sensitivity, and compatibility with ESP32. Mention additional hardware components required for power management, communication modules (Wi-Fi, Bluetooth), and any necessary peripherals.
3. **Sensor Integration:** Describe the integration process of sensors with the ESP32 microcontroller, providing clear wiring diagrams and relevant code snippets for data acquisition. Highlight any challenges encountered during integration and the corresponding solutions.
4. **Data Processing and Analysis:** Discuss the methods employed for processing and analyzing sensor data on the ESP32. Include details on calibration techniques to ensure the accuracy of measurements and algorithms for aggregating and storing data efficiently.
5. Communication Protocol: Explain the choice of communication protocol (e.g., MQTT, HTTP) for transmitting sensor data from the ESP32 to a central server or cloud platform. Elaborate on security measures such as encryption and authentication to safeguard data during transmission.
6. **User Interface:** Outline the design of a user-friendly interface for visualizing real-time and historical air quality data. Discuss options for displaying data on local interfaces (e.g., LCD screens) and remote platforms such as web dashboards or mobile applications.
7. **Alerting Mechanism:** Detail the implementation of an alerting mechanism to notify users when air pollution levels exceed predefined thresholds. Explain how alerts are generated, transmitted, and escalated in case of critical events.
8. **Power Management:** Present strategies for optimizing power consumption to prolong battery life or minimize energy usage, considering the system's deployment environment and operational requirements.
9. Describe the **testing plan** devised to validate the accuracy and reliability of the air pollution monitoring system. Discuss simulation methods for replicating various environmental conditions during testing.
10. **Deployment Strategy:** Discuss considerations for deploying the system in different environments, addressing factors such as installation, maintenance, and scalability. Highlight any adjustments made for indoor versus outdoor deployments.

### Intended Audience

1. **Environmental Agencies and Authorities:** Officials responsible for monitoring and regulating air quality at local, regional, or national levels. They can utilize the system's data to assess pollution levels, identify sources of contamination, and formulate policies for environmental protection and public health.
2. **Urban Planners and City Administrators:** Professionals involved in urban development and infrastructure management. They can leverage the system's insights to design sustainable cities, optimize traffic flow, and mitigate pollution hotspots through zoning regulations, green spaces, and public transportation initiatives**.**
3. **Healthcare Professionals:** Doctors, epidemiologists, and public health researchers interested in understanding the correlation between air pollution exposure and health outcomes. They can use the system's data to conduct epidemiological studies, assess respiratory risks, and advocate for preventive healthcare measures in vulnerable communities.
4. **Educational Institutions:** Students, educators, and researchers in environmental science, engineering, and related fields. They can incorporate the system into academic curricula, research projects, and outreach activities to raise awareness about air quality issues, promote STEM education, and foster innovation in pollution monitoring technologies.
5. **Non-Governmental Organizations (NGOs):**Environmental advocacy groups, community organizations, and citizen scientists engaged in grassroots activism and environmental justice campaigns. They can collaborate with the system's developers to deploy sensor networks in underserved communities, empower residents with pollution data, and advocate for policy reforms.
6. **Industry Stakeholders:** Industrial firms, manufacturing facilities, and energy producers seeking to minimize their environmental footprint and comply with regulatory standards. They can deploy the system to monitor emissions, optimize production processes, and demonstrate corporate social responsibility through transparent pollution reporting**.**
7. **General Public:** Residents, commuters, and consumers concerned about air quality in their surroundings. They can access the system's user-friendly interfaces, mobile apps, or public displays to check real-time pollution levels, receive personalized alerts, and make informed decisions about outdoor activities, travel routes, and lifestyle choices.
8. **Technology Enthusiasts and DIY Communities:** Hobbyists, makers, and tinkerers interested in experimenting with IoT devices and sensor technologies. They can explore open-source designs, contribute to community forums, and develop custom applications or sensor configurations for air quality monitoring projects in their homes, schools, or local neighbourhood’s.

### Definitions, Acronyms, Abbreviations

IoT: Internet of Things - a network of interconnected devices that can communicate and exchange data.

**MQ:** message queue

### References

IoT-Based Smart Air Pollution Monitoring and Control System" by V. P. Jain and P. K. Tripathy, in the International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT): This article discusses the design and development of a smart air pollution monitoring and control system based on IoT technology. "IoT-Based Air Pollution Monitoring System: A Review" by Anand Nayyar, in the journal Sensors: This paper provides a comprehensive review of IoT-based air pollution monitoring systems, discussing various technologies, methodologies, and challenges.

### Overview

The MQ135 sensor is a popular gas sensor widely used for detecting various gases such as carbon dioxide (CO2), ammonia (NH3), nitrogen oxides (NOx), and volatile organic compounds (VOCs), which are common pollutants in the environment. It detects these gases based on changes in electrical conductivity when exposed to the target gases which detects Carbon monoxide in air and MQ-5 sensor it detects methane lpg gas . A microcontroller such as ESP32 is used to interface with the MQ135 sensor,MQ7,MQ5 and process the sensor readings. The microcontroller collects data from the sensor and determines the pollution levels. And levels on website shows the pollution levels in the environment. The components of the pollution detector may be housed in an enclosure to protect them from environmental factors and ensure durability. The sensor may need to be calibrated initially to ensure accurate measurements. Additionally, periodic testing and calibration may be required to maintain accuracy over time. A air pollution monitoring sensor provides real-time monitoring of air quality, allowing users to assess the level of pollution caused by vehicles and take appropriate measures to mitigate its impact on health and the environment.

## OVERALL DESCRIPTION

### User Interfaces

1. **Real-Time Data Visualization:** The web dashboard displays real-time air quality metrics such as concentrations of various pollutants (CO, NO2, SO2, PM2.5, PM10) in graphical format (pie charts, bar graphs, etc.). Each pollutant has its dedicated graph, allowing users to monitor changes over time.
2. **Customizable Alerts:** Users can set personalized thresholds for each pollutant and when pollution levels exceed predefined limits indicates pollution level. settings can be adjusted from the dashboard.
3. **Historical Data Analysis:** The dashboard provides options to view historical pollution data for trend analysis and comparison over different time periods (daily, weekly, monthly). Users can select specific date ranges and generate customizable reports.

### System Interfaces

**Sensor Module (MQ135):** The MQ135 sensor module is used to detect pollutants in the environment , such as carbon dioxide (CO2), nitrogen oxides (NOx), and volatile organic compounds (VOCs).

**Senser Module (MQ7):** It is suitable for detecting H2, LPG, CH4, CO, Alcohol. Due to its high sensitivity and fast response time, measurements can be taken as soon as possible.

**Sensor Module(MQ5):** The MQ7 is a simple-to-use Carbon Monoxide (CO) sensor suitable for sensing CO concentrations in the air. It can detect CO-gas concentrations anywhere from 20 to 2000ppm

**Microcontroller (ESP32):** The microcontroller interfaces with the sensor module to read the pollution levels. It processes the sensor data and executes control logic for showing the levels in the user dashboard

**Testing and Calibration:**

The system may undergo testing and calibration to ensure accurate detection of pollution levels and to reliable values on the dashboard. Calibration involves comparing sensor readings with reference measurements and adjusting parameters as needed to improve accuracy**.**

### Constrains, Assumption and Abbreviations:

#### Constraints

**Power Consumption**: ESP32 devices should be optimized for low power consumption to ensure longer battery life or efficient use of power sources**.**

**Sensor Calibration:** Gas sensors may require periodic calibration to maintain accuracy. The system should incorporate mechanisms for automated or user-initiated calibration routines.

**Sensor Sensitivity and Accuracy**: The gas sensors must be sensitive and accurate enough to detect pollutants within acceptable limits. Regular sensor maintenance and periodic replacement may be necessary to ensure reliable performance.

**Environmental Conditions**: The system should be designed to withstand various environmental conditions, including temperature extremes, humidity, and exposure to pollutants, to ensure reliable operation over time.

**Scalability**: The system architecture should allow for scalability to accommodate additional sensors or monitoring nodes as needed, without significant redesign or performance degradation.

**Real-Time Monitoring**: The system should provide real-time monitoring capabilities, enabling timely detection and response to changes in air quality levels.

**User Interface**: A user-friendly interface should be provided to allow users to view air quality data, set alerts, and configure system parameters easily.

.

#### Assumptions

**Sensor Accuracy:** The pollution sensor, such as the MQ135, is assumed to provide accurate and reliable measurements of pollutant levels in the air.

**Threshold Setting:** Threshold levels for pollutant concentrations triggering an alert message are predetermined based on regulatory standards or health guidelines. These thresholds may vary for different pollutants**.**

**Microcontroller Reliability:** The microcontroller used in the system (ESP8266) is assumed to function reliably and be capable of processing sensor data and sending SMS alerts without significant errors or malfunctions.

#### Dependencies

**1Programming Environment:** You'll need knowledge of programming languages like C/C++ (for Arduino) or Python (for Raspberry Pi) to develop the firmware or software for the

microcontroller. Libraries for interfacing with the MQ135 sensor and GSM module are available and may need to be installed.

2**.Alert Thresholds and Logic:** Define pollution threshold levels that trigger SMS alerts. Implement logic in the microcontroller to compare real-time pollution data with these thresholds and give pollution levels accordingly.

### User Characteristics

**Citizens and Residents**: Individuals living in urban or industrial areas who are concerned about their exposure to air pollution and seek real-time information to make informed decisions about outdoor activities, health precautions, and lifestyle choices.

**Community Organizations:** Local community groups, environmental organizations, and advocacy groups interested in monitoring air quality in their neighbourhood’s, raising awareness about pollution issues, and advocating for cleaner air policies and initiatives.

**City Planners and Urban Developers**: Professionals involved in urban planning, infrastructure development, and city management who require air quality data to assess the impact of urbanization, traffic congestion, industrial activities, and other factors on local air quality and public health.

**Health Professionals**: Medical professionals, public health officials, and researchers who utilize air quality data to study the health effects of air pollution, identify vulnerable populations, and develop strategies for mitigating health risks associated with poor air quality.

**Environmental Agencies and Regulatory Bodies**: Government agencies responsible for monitoring and regulating air quality, enforcing environmental laws and regulations, and ensuring compliance with emission standards and pollution control measures.

**Industry and Business Owners**: Industrial facilities, manufacturing plants, and commercial establishments that use gas sensors for internal air quality monitoring, emission control, and compliance reporting to regulatory authorities.

**Educational Institutions**: Schools, colleges, and universities that incorporate air quality monitoring projects into their curriculum to educate students about environmental science, pollution control, and sustainability practices.

**Technology Enthusiasts and DIY Hobbyists**: Individuals interested in electronics, IoT development, and DIY projects who use ESP32 and gas sensors to build their air pollution monitoring systems for personal use, experimentation, or educational purposes.

.

# System Analysis

### Existing systems

Traditional air pollution monitoring systems rely on manual methods and stationary monitoring stations to measure air quality. These systems typically involve the use of specialized equipment and trained personnel to collect air samples at specific locations, which are then analysed in laboratories to determine pollutant concentrations.

Stationary monitoring stations are set up in strategic locations, such as urban centres, industrial areas, and near sources of pollution, to continuously monitor air quality. These stations may be equipped with instruments such as gas analysers, particulate matter samplers, and meteorological sensors to measure various pollutants, meteorological parameters, and atmospheric conditions.

In addition to stationary monitoring stations, mobile monitoring units equipped with air quality monitoring equipment may be deployed to assess air quality in different areas or to investigate pollution sources and hotspots. These mobile units can be mounted on vehicles or carried by personnel to collect real-time data as they traverse various locations.

Data collected from stationary and mobile monitoring stations are typically recorded manually or transmitted to centralized databases for analysis and reporting. This data is used by environmental agencies, researchers, policymakers, and the public to assess air quality trends, identify pollution sources, evaluate compliance with air quality standards, and develop strategies for pollution control and mitigation.

Overall, traditional air pollution monitoring systems rely on manual sampling and analysis methods, which can be labor-intensive, time-consuming, and limited in spatial coverage. Advances in technology, such as IoT-based monitoring systems, offer the potential for real-time monitoring, greater spatial coverage, and more efficient data collection and analysis.

Disadvantages on existing system:

1. **Limited Spatial Coverage**: Due to the stationary nature of monitoring stations and the need for manual sampling, coverage is often limited to specific locations, which may not adequately represent overall air quality across a region.
2. **Time-Consuming and Labor-Intensive**: Manual sampling and analysis require significant time and effort from trained personnel, leading to delays in data collection, analysis, and reporting. This can hinder timely response to pollution events or changes in air quality.
3. **Costly Operation and Maintenance**: Operating and maintaining stationary monitoring stations and mobile monitoring units can be expensive due to equipment costs, personnel expenses, calibration, and periodic maintenance.
4. **Limited Real-Time Monitoring**: Traditional systems typically provide data at discrete time intervals rather than real-time, making it challenging to capture short-term fluctuations in air quality or respond quickly to emerging pollution issues.
5. **Vulnerability to Human Error**: Manual data collection and analysis are prone to human error, including sampling inconsistencies, data entry mistakes, and interpretation errors, which can affect the accuracy and reliability of air quality measurements.
6. **Weather Dependency**: Weather conditions can impact air quality measurements, and traditional systems may not always account for meteorological factors such as wind speed, temperature, and humidity, leading to potential inaccuracies in data interpretation**.**
7. **Limited Mobility:** Stationary monitoring stations are fixed in location, limiting their ability to capture air quality variations in different areas or respond to pollution events in real-time. Mobile monitoring units offer more flexibility but may still be constrained by logistical challenges and resource limitations.
8. **Inadequate Coverage of Pollution Sources**: Traditional systems may not adequately capture emissions from mobile sources such as vehicles or localized sources such as industrial facilities, which can contribute significantly to air pollution in certain areas.

### Proposed System

### An existing IoT-based air pollution monitoring system employs ESP32 microcontrollers along with gas sensors like MQ135, MQ7, and MQ5 to provide comprehensive real-time monitoring of air quality. These systems consist of sensor nodes distributed across various locations, continuously measuring key air quality parameters. The ESP32 microcontroller acts as the central processing unit, facilitating data collection, analysis, and transmission to a centralized database or cloud platform. Specifically, the MQ135 sensor detects gases such as ammonia and nitrogen oxides, while the MQ7 sensor monitors carbon monoxide levels, and the MQ5 sensor focuses on flammable gases like methane. This integration enables the system to collect real-time data on pollutant concentrations, which can be accessed by stakeholders such as environmental agencies, city planners, and the public through web-based dashboards. With low-power operation and efficient data transmission, the system is suitable for deployment in remote areas, offering scalability for expanded monitoring coverage. Overall, this IoT-based air pollution monitoring system provides a cost-effective and user-friendly solution for managing air quality, supporting efforts to promote public health and environmental sustainability

### .Requirements Specification

### User Requirements

1. **Real-time Monitoring:** The system should provide real-time monitoring of air quality metrics including levels of pollutants such as carbon monoxide (CO), nitrogen dioxide (NO2), sulfur dioxide (SO2), particulate matter (PM2.5 and PM10), and ozone (O3).
2. **Accuracy and Precision:** Users expect accurate and precise measurements of air pollutants to ensure reliable data for decision-making and analysis**.**
3. **Remote Access:** Users should be able to access air quality data remotely through a user-friendly interface, such as a mobile application or web dashboard, allowing them to monitor air quality from anywhere at any time.
4. **Alerts and Notifications:** The system should have the capability to send alerts and notifications to users when air pollutant levels exceed predefined thresholds, enabling timely responses to potential health hazards.
5. **Historical Data Logging:** Users require the system to store historical air quality data for analysis, trend identification, and reporting purposes. This data logging feature helps in understanding long-term patterns and identifying pollution hotspots.
6. **Integration with Smart Devices:** Integration with other smart home devices or automation systems can enhance user experience. For example, the system could trigger actions like turning on air purifiers or sending alerts to HVAC systems for ventilation adjustments based on air quality readings.
7. **Modularity and Scalability**: Users may need the flexibility to expand the system by adding more sensors or integrating with additional IoT devices in the future. The system should be modular and scalable to accommodate future upgrades or expansions**.**
8. **Low Power Consumption:** Since the system may operate continuously, it should be designed for low power consumption to prolong battery life or reduce energy costs when connected to mains power.
9. **Easy Installation and Maintenance:** Users should be able to easily install, set up, and maintain the system without requiring specialized technical skills. Clear instructions and troubleshooting guides can aid in user adoption and satisfaction.
10. **Data Privacy and Security:** Protection of sensitive air quality data is crucial. The system should implement robust security measures to ensure data privacy, integrity, and protection against unauthorized access or tampering.

### System Requirements

1. **Compatibility with ESP32 Module**: The system should be compatible with the ESP32 microcontroller platform, ensuring seamless integration and communication between the sensors, microcontroller, and network connectivity components.
2. **Support for Multiple Gas Sensors:** The system should support different types of gas sensors capable of detecting various air pollutants such as carbon monoxide (CO), nitrogen dioxide (NO2), sulfur dioxide (SO2), particulate matter (PM2.5 and PM10), and ozone (O3). It should allow for easy integration and management of these sensors.
3. **Wireless Connectivity:** The system should include Wi-Fi or Bluetooth connectivity options to enable communication between the ESP32 module and external devices such as smartphones, tablets, or computers. This facilitates remote monitoring and data retrieval.
4. **Data Acquisition and Processing:** The system should be able to acquire sensor data at regular intervals and process it to obtain accurate measurements of air pollutant concentrations. It should implement algorithms for calibration, noise reduction, and data filtering to ensure reliable measurements.
5. **Real-time Data Transmission:** The system should transmit air quality data in real-time to a centralized server or cloud platform for storage and analysis. This requires efficient data transmission protocols and reliable network connectivity to prevent data loss or delays.
6. **User Interface:** The system should provide a user-friendly interface for accessing air quality data, configuring settings, and receiving alerts or notifications. This interface can be in the form of a mobile application, web dashboard, or dedicated display connected to the ESP32 module.
7. **Threshold Monitoring and Alerts:** The system should allow users to set threshold levels for air pollutant concentrations and trigger alerts or notifications when these thresholds are exceeded. This feature enables timely responses to changes in air quality and potential health hazards.
8. **Data Logging and Visualization:** The system should log historical air quality data for analysis, trend identification, and reporting purposes. It should provide visualization tools such as charts, graphs, or maps to help users interpret and understand the data.
9. **Power Management:** The system should implement efficient power management techniques to optimize energy consumption and extend battery life, especially in battery-powered deployments. This may include sleep modes, low-power sensor configurations, and power-saving strategies.
10. **Security Measures:** The system should incorporate security measures to protect sensitive air quality data from unauthorized access, tampering, or interception. This includes encryption of data transmission, authentication mechanisms, and secure storage practices.

### Domain Requirements

1. **Environmental Compatibility:** The system should be designed to withstand various environmental conditions, including temperature fluctuations, humidity levels, and exposure to outdoor elements such as rain and dust.
2. **Sensor Calibration:** Calibration procedures should be established to ensure the accuracy and reliability of sensor measurements over time. This includes periodic calibration checks and adjustments to compensate for sensor drift or degradation.
3. **Pollutant Detection Range:** The system should support gas sensors with detection ranges suitable for monitoring relevant air pollutants at concentrations typically encountered in urban, industrial, or residential environments.
4. **Pollutant Specificity:** Gas sensors should exhibit high specificity to target pollutants to minimize cross-sensitivity and false readings from interfering gases commonly found in the environment.
5. **Response Time:** Sensors should have fast response times to detect rapid changes in air pollutant concentrations and provide timely updates to the monitoring system.
6. **Sensitivity:** The system should utilize gas sensors with sufficient sensitivity to detect low concentrations of air pollutants, especially for pollutants with stringent regulatory limits or health implications at low exposure levels.
7. **Long-Term Stability:** Sensors should demonstrate long-term stability to maintain consistent performance over extended periods without significant degradation in accuracy or reliability.
8. **Data Validation and Quality Assurance**: Mechanisms should be implemented to validate sensor data integrity, identify outliers or anomalies, and ensure data quality before storage, analysis, or presentation to users.
9. **Regulatory Compliance:** The system should adhere to relevant environmental regulations, standards, and guidelines governing air quality monitoring, including calibration requirements, sensor performance criteria, and data reporting protocols.
10. Where applicable, the system should support integration with existing environmental monitoring networks operated by government agencies, research institutions, or non-governmental organizations to contribute to broader air quality monitoring efforts and data sharing initiatives.

### Functional Requirements

1. **Sensor Data Acquisition**: Collect real-time data from various gas sensors measuring air pollutants.
2. **Data Processing and Analysis:** Process sensor data to calculate pollutant concentrations, apply calibration algorithms, and ensure data accuracy.
3. **Real-time Monitoring and Display:** Display air quality metrics in real-time, updating regularly.
4. Alerting and Notification: Send alerts when pollutant concentrations exceed predefined thresholds, via email, SMS, or push notifications.
5. **Remote Access and Control:** Enable users to access data and control settings remotely through a mobile app or web interface, with secure authentication.
6. **Data Logging and Storage:** Log historical data locally or transmit it to a remote server for storage and analysis.
7. **Visualization and Reporting:** Provide visualizations and reports summarizing air quality trends and variations.
8. **Integration with External Systems**: Integrate with external monitoring networks or platforms for data sharing and interoperability.
9. **Power Management:** Implement power-saving features for energy optimization and extended battery life.
10. **Error Handling and Diagnostics:** Detect and handle errors, providing diagnostic tools and logs for troubleshooting.

#### Non-Functional Requirements

1. **Performance:** The system should be capable of processing and analyzing pollution data in real-time with minimal latency, ensuring timely notifications to users.
2. **Scalability:** The system should be scalable to accommodate a growing number of vehicles and users without sacrificing performance or reliability.
3. **Reliability:** The system should have high availability, minimizing downtime and ensuring continuous monitoring of vehicle emissions.
4. **Accuracy:** Pollution sensors should provide accurate measurements within acceptable margins of error to maintain the system's credibility and effectiveness.
5. **Availability:** The system should be available 24/7 to provide timely notifications to users, with minimal scheduled maintenance downtime.
6. **Security:** Implement robust security measures to protect sensitive data, including encryption of data transmissions, access control mechanisms, and protection against cyber threats.