Software Requirements Specification

for

AMS(Atmospheric Management System)

Version 1.0

Prepared by

Lakshya Bhardwaj-	**IIT2020097**
Tarun Pal-	**IIT2020098**
Kunal Prajapati-	**IIT2020099**
Mohit Kumar Mina-	**IIT2020100**

Indian Institute of Information Technology, Allahabad

9/Mar/2022

Table of Contents

1. Introduction

Atmospheric management system is a GUI enabled air quality management system for IIIT-A campus. The system should be able to monitor the levels of hazardous gasses present in the environment.

1.1 Purpose

Air pollution has been a matter of grave concern since time immemorial. The adverse repercussions of air pollution are still being observed, which have indicated uncertainties and gaps in existing Atmospheric management policies and control strategies. Hence, to reduce the detrimental consequences of air pollution, our application is designed to make a GUI application(**AMS(Atmospheric Management System)**), the system must also be capable of measuring the humidity and temperature present in the atmosphere.
The main purpose of our application is to:

- The system should be able to monitor the level of hazardous gasses present in the environment (via means of random number generation).
- In an alarming situation, the system must be able to generate an alarm (also add an auto-generated message and mail system).

1.2 Definitions, Acronyms, and Abbreviations

Acronyms and Abbreviations:

1. “AMS©”: Copyrighted app name.
2. SRS: Software Requirement Specification
3.

Definitions:

1. Pollutants - PM10, No2, O3, PM2.5
1. “AMS©” - A software to manage the air quality control for IIIT-A campus.

1.3 Intended Audience and Reading Suggestions

This document is intended for the audiences such as developers, users, project managers and document writers. This SRS contains data of Pollutants of air. The rest of this SRS is organized as follows: Section 2.3 gives an overall description of the User Classes and Its Characteristics. It gives information about the various user classes that users anticipate when the individual will use this GUI. Section 4 gives specific information about the system features of **AMS(Atmospheric Management System)**. Requirements which the software is expected to deliver. Functional requirements are given in various cases. Some performance requirements and design constraints are also given.

1.4 Product Scope

AMS(Atmospheric Management System) is a GUI software for monitoring the level of hazardous gasses present in the environment. The data will be collected through API, which will give an estimate of the composition of Pollutant Gasses. The objective of this project is to measure the quality of air that we breathe so that we can improve it [PM10, NO2, O3, PM2.5], by analyzing, the density of pollutant Gasses, on daily basis, and in an alarming situation, the system must be able to generate alarm and add the auto-generated message and mail system and tell the hazard ness of the air.

2. Overall Description

2.1 Product Perspective

In today’s day, we find our country has a majority stake in polluted cities. So, it is our concern to measure the quality of air that we breathe so that we can improve it. To solve this problem we are going to create a GUI named **AMS(Atmospheric Management System)**©.

AMS(Atmospheric Management System)© is a GUI that is designed specifically for getting the composition of Pollutants in the air so that we can grade the air according to human health and decide the levels for it.
There will be a GUI, accessed by the user so that we can monitor the air quality, temperature and humidity and he will also be able to view the graphs of the previous records.

So that it can predict the next day's temperature.

This Project’s Perspective is to monitor the air quality in the atmosphere and also keep a track of its different characteristics.

- Keeps track of the air contents, humidity, temperature, etc.
- Works in real-time.
- Computes the grading levels and gives remarks on air quality.
- Gives Warnings when the quality drops under critical level and informs all campuses (using email)in the locality.
- Stores the complete track record which can be used for multiple purposes.

2.2 Product Functions

AMS(Atmospheric Management System) supports the following use cases:

Use Cases	**Description of Use Cases**
Login	User may log in/log out to the system
Password verification	Every time a user input the password it will be verified.
Detect density of Pollutants	Users can view details of the density of Pollutants.
Detect the Temperature and Humidity	Users can view the Temperature and Humidity
View History	Users can view the history.

2.3 User Classes and Characteristics

This project is intended to be used by various user classes. These classes can be

1. Check
2. Campus
3. Hostel
4. Professor Residences
5. Canteen

In every class (except Check class) we have attributes to denote the level of a particular gas(like NO2, SO3, CO2 etc.) in that area, we also have attributes for temperature and humidity.
And we have a function to find the concentration of different gasses(like NO2, SO3, CO2 etc.), also have a function to find the quality of air and a function to send the data to the Check class.
The Check class collects the data from Campus, Hostel, Professor Residences and Canteen classes. And this class decides whether the quality of air is good or not. If the air quality is not good then an email is to send to the user about the poor quality of air.

2.4 Operating Environment

- In this project, IoT devices would be used to obtain data. Using it we will grade the quality of air. In critical conditions, it will generate the email to all the users.
- We Will store the gathered data and computed results in databases deployed on a web server that would be needed and use this data for the forecast.
- MySQL will be used as a database to store the gathered data.

2.5 Design and Implementation Constraints

- Web-based systems are platform dependent.
- Work products such as documents will be in compliance with IEEE standards.

2.6 Assumptions and Dependencies

- User(Manager/Admin) must have/use valid UserId and Password.
- Users must login in to the system to access.
- GUI is only in English.
- Users must be accessing it through browsers like Chrome, Firefox or Safari.

3. External Interface Requirements

3.1 User Interfaces

In this project, we will collect data from the “**Indian Meteorological Department**” and we analyze the data. The system will monitor the level of hazardous gasses present in the environment. And in a situation where the quality of air is very harmful then the system will generate an email and send it to the users.

3.2 Hardware Interfaces

Various interfaces for the product:

- Touch Screen/Monitor
- KeyPad and Mouse
- Continues battery backup

3.3 Software Interfaces

- Any Operating System, such as Windows or any Linux Distribution.
- MySQL for a database storing and providing data servers, in run time.
- A GUI built using JavaApplet, or Java desktop application.

3.4 Communications Interfaces

Since it will be a GUI application, having a User Interface for the manager to result, after data is interpreted, there must be a connection to our software to the User. This data collected from API should be collected in a database (we will use MySQL) for storing it timely. So, there must be SQL server connection to our software.

4. System Features

Use Case 1:

Name: Authorized login

Summary: Allows User to login

Actors: Users

Pre-conditions:

- Internet connectivity

Main Success Scenario:

- User clicks on the login button.
- GUI application checks for the authorization of login.

Extension:

- Username or password incorrect. Shows error.

Post-Condition:

Users can access all the features of Gui.

Use Case 2:

Name: Detect density of Pollutants.

Summary: Allows users to view the details about the density of each and every particular Pollutant.

Actors: Users

Pre-conditions:

- Internet connectivity

Main Success Scenario:

- Users can view all the details about the density of Pollutants.

Extension: NIL

Post-Condition: NIL

Use Case 3:

Name: Detect the Temperature and Humidity

Summary: Allows User to view the details about the Temperature and Humidity.

Actors: Users

Pre-conditions:

- Internet connectivity

Main Success Scenario:

- Users can view the Temperature and Humidity.

Extension: NIL

Post-Condition: NIL

Use Case 4:

Name: View history

Summary: Allows User to view the history of Air Quality, Temperature, Humidity and the density of Pollutants.

Actors: Users

Pre-conditions:

- Internet connectivity

Main Success Scenario:

- Users can view the history of every atmospheric perspective which we stored.

Extension: NIL

Post-Condition: NIL

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Data should be updated every 30 minutes.
- Load time of UI should be not more than 15 sec.
- Sensors should be able to give data for every 5 minutes.

5.2 Hardware Requirements

- Standard pc
- Internet connection with good speed.
- Pentium IV 1.7Ghz dos or better processor.
- 1 GB ram or more
- At least 256 Gb Hard Disk Space.

5.3 Software Requirements

- Front End,
- Server,
- Back End,
- Database: MySql.

5.4 Software Quality Attributes

- This software will provide a user-friendly user interface.

- This app will be mobile-friendly too so that the Manager or the Caretaker can view it anywhere, anytime, and does not need to rely upon the desktop to view it.

- To further advancement of this application timely data will be added and graphs will be plotted timely, so as to get a very fair and accurate estimate.

5.5 Design Constraint

Security: The files in which the information regarding the Density of Pollutants should be secured.

Fault Tolerance: Data should not become corrupted in case of a system crash or power failure.

5.6 Other Requirements

Our application will use MySQL server for the database, and will timely sync with the data provided by the "**Indian Meteorological Department**".