POLITECNICO
MILANO 1863

Politecnico Di Milano

Software Engineering for Geoinformatics

REPORT

Requirement Analysis & Specification Document:

AQI-GDP Dashboard Application

Authors

Iyad Abdi

Filippo Bissi

Valerio Paoloni

.....Interruzione di colonna.....

Person Code

10808624

10438335

10401564

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this study is to design and develop an interactive client-server application to facilitate the efficient and effective querying, processing, and visualization of air quality data from ground sensor stations located in the top 10 cities globally by GDP, segmented by their historical and developmental group:

Global North Cities:

- Tokyo, Japan
- New York City, USA
- Los Angeles, USA
- Chicago, USA
- London, United Kingdom

Global South Cities:

- Shanghai, China
- Beijing, China
- Mumbai, India
- Sao Paulo, Brazil
- Shenzhen, China

By utilizing publicly available database management systems, web server development, and data visualization, this application aims to provide users with an informative platform to access and analyze environmental data. By comparing the air quality data from cities with the highest GDPs in both regions, this project seeks to shed light on the globally common sources of air pollution, correlation between productive output and pollution, and identify effective policy solutions to improve environmental conditions and protect public health. Through this application, the research team aims to contribute to the development of a more sustainable and livable environment for people in urban areas around the world.

1.2. Scope

The scope of this application is to design and develop an interactive client-server application that enables users to access, query, and visualize air quality and weather sensor data collected from the public digital archive available on the *World's Air Pollution: Real-time Air Quality Index* website https://waqi.info and/or the *Air Pollution in the World Real-time Air Quality Index (AQI)* https://aqicn.org. The application consists of three main components: a database where the data is ingested beforehand, a web server (backend) that exposes a REST API used for querying the database and retrieving data, and a dashboard that exploits the web server to provide means for requesting, processing, and visualizing data.

The database component should be capable of storing large time-series datasets if feasible, as well as associated metadata for each observation. The backend component should be able to provide an intuitive and secure API that allows users to query the data and perform various data processing tasks. The dashboard component should provide simple, yet interactive, visualization tools that enable users to explore the data in different ways, such as maps, perhaps dynamic graphs, and other visualization tools. The application should be scalable, reliable, and maintainable, and should adhere to industry-standard software development practices and guidelines as taught in this course. The final

deliverable aims to be a fully functional, user-friendly application that allows users to access, query, and visualize air quality sensor data of our 10 focus cities in a meaningful and informative way.

1.3. Short overview

The technical functionalities relate to the implementation and operation of the application, including the data integration, database management, API development, and data processing functionality. The user-enabled tools relate to the user experience and the ways in which users will interact with the data via the dashboard. The processing functionalities are integral to both the technical and user-enabled functionalities, as they will enable users to process and visualize the data in meaningful ways.

Technical functionalities:

- Ingestion of air quality and weather sensor data from https://waqi.info and/or https://aqicn.org into a database
- Design and development of a proper database schema to store the data and associated metadata
- Development of a REST API for querying and retrieving data from the database
- Implementation of data cleaning and preprocessing functionality within the web server
- Harmonization, aggregation, and filtering of data based on different patterns and criteria
- Space-time aggregations and other data transformations

User-enabled tools:

- Interactive dashboard that allows users to process, visualize, and explore the data retrieved from the web server using original manipulation strategies
- Customizable views that allow users to generate custom visualizations of the data
- Map-based views and interactive graphs that enable users to explore the data in different ways

1.4. Acronyms and other definitions

TERM	DEFINITION
AQI	Air Quality Index; a measure of air quality that takes into account the concentration of pollutants such as PM2.5, PM10, O_3, NO_2, SO_2, and CO
Choropleth Map	A map that shows interval data using shading, coloring, or placing of symbols within predefined areas, to indicate mean values of a specific quantity in those areas

Dash	A Python framework for building interactive web-based dashboards and data visualization application
DBMS	Database Management System; software system we are going to use to store, retrieve, and run queries on data
Flask	A Python web framework that allows us to easily build and deploy web applications
GitHub	A web-based hosting service used for the development of our open-source project
JSON	JavaScript Object Notation; a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate
Pandas	An open-source Python library for data manipulation, analysis, and cleaning
PostgreSQL	An open-source database management system that uses a client-server model where the server program manages the database files and accepts connections to the database from client applications
Python	A high-level programming language we are going to use for building our software. It will be the basis of every functional tool used in our web application
REST	Representational State Transfer; a software architectural style that was created to guide the design and development of the architecture for the Web. It defines a set of constraints for how the Web should behave
REST API	An Application Programming Interface that satisfies the constraints of REST architecture and allows the interaction with RESTful web services

2. Application specifications

2.1. Users

Based on the scope of the application, we envision the users falling into 3 main categories:

1. **Administrator**: Users in this category are responsible for managing the system. They have privileges to perform administrative tasks such as data updates, user access control, system configuration, and overall system maintenance.
2. **Data Analysts**: Users categorized as Data Analysts are focused on analyzing and interpreting the AQI data across the top ten cities by GDP. They require tools and functionalities for data exploration, filtering, visualization, and exporting to support their data analysis tasks.
3. **Data Explorer**: Users falling under the Data Explorer category are interested in exploring the AQI data across different cities and time periods. They seek interactive features to select cities, view historical trends, compare AQI levels, and analyze patterns or correlations in the data.

2.2. Dataset

Data will be retrieved using the REST API of our data source. The data will be collected and integrated from a sensor in each of the 10 cities. The data will be stored locally in our database and able to be retrieved on a continuous basis.

Point positions will provide precise information on the location of air quality sensors which can be used to plot this sensor data on a map, visualize spatial patterns in air quality observations, and filter data by location, allowing users to focus on specific areas of interest.

The AQI data variables represent different pollutants that can be measured by air quality sensors:

- **PM2.5**: fine particulate matter with a diameter of less than 2.5 micrometers, which can penetrate deep into the lungs and cause respiratory problems.
- **PM10**: larger particulate matter with a diameter of less than 10 micrometers, which can also contribute to respiratory problems.
- **O3**: ground-level ozone, which can cause respiratory problems and aggravate asthma.
- **NO_2**: nitrogen dioxide, which is a common pollutant from traffic and can cause respiratory problems.
- **SO_2**: sulfur dioxide, which is typically released from burning fossil fuels and can cause respiratory problems and acid rain.
- **CO**: carbon monoxide, which is a colorless and odorless gas that can cause headaches, dizziness, and nausea at high levels of exposure.

Collectively, these AQI data variables provide important information on the levels of different pollutants in the air, which can help inform policies and interventions aimed at reducing air pollution and improving public health.

2.3. Operations

Operations of the web application can be organized into two categories: **user** and **system**.

User operations regard the user's interaction with the application and front-end functionalities via the dashboard. The dashboard provides interactive visualization tools, such as maps and graphs, to allow users to explore air quality data. The frontend communicates with the backend through a REST API, allowing users to request data and retrieve it in the JSON format. The frontend is responsible for providing an intuitive user experience and allowing users to explore and visualize air quality data efficiently. These operations include city selection, AQI data retrieval, data filtering, and data visualization.

System operations regard the backend functionalities of the application. The database ingests air quality and sensor data from our selected city sensors using an API from the World Air Quality Index to be stored in a database where it can be retrieved by the web server. Once the data is ingested, the web server exposes a REST API to allow users to query and retrieve data based on their requirements. The web server performs data processing tasks before returning results to users such as data, spatial, and temporal aggregations, allowing users to retrieve data for specific time periods and geographic areas. The API returns results in the JSON format, which can be easily consumed and converted by the frontend of the web app.

2.4. Product function

The web application will function by displaying a global map-based view with dots of our selected cities. The map will allow for zooming in and out in order to assess the specific AQI readings of global regions. The dot on each city will represent the respective city's AQI reading and the corresponding color related to the air quality.

Users will be able to select a city, or grouping of cities to make comparative readings of AQI data. (E.g. Comparing AQI of Los Angeles with Sao Paolo.)

2.5. Relevant phenomena

Air quality and air pollution is related to economic activity, as industrial and transportation-related emissions contribute significantly to the concentration of pollutants in the air. As a result, large cities, which concentrate people and higher levels of economic activity, tend to have higher levels of air pollution. This relationship between economic activity and air quality has important implications for public health and policy. By exploring this relationship through the AQI web app, users can gain insights into how different cities' economic activities (among other factors) affects air quality.

2.6. Shared phenomena

The objective of this web application relates to public health and environmental livability. As these phenomena are not limited to a single individual or group, but rather have a broad societal impact, our user case will not require registration or login functionalities to access the web app. Rather, all visitors of the application's website will be able to interface with it.

3. Use case

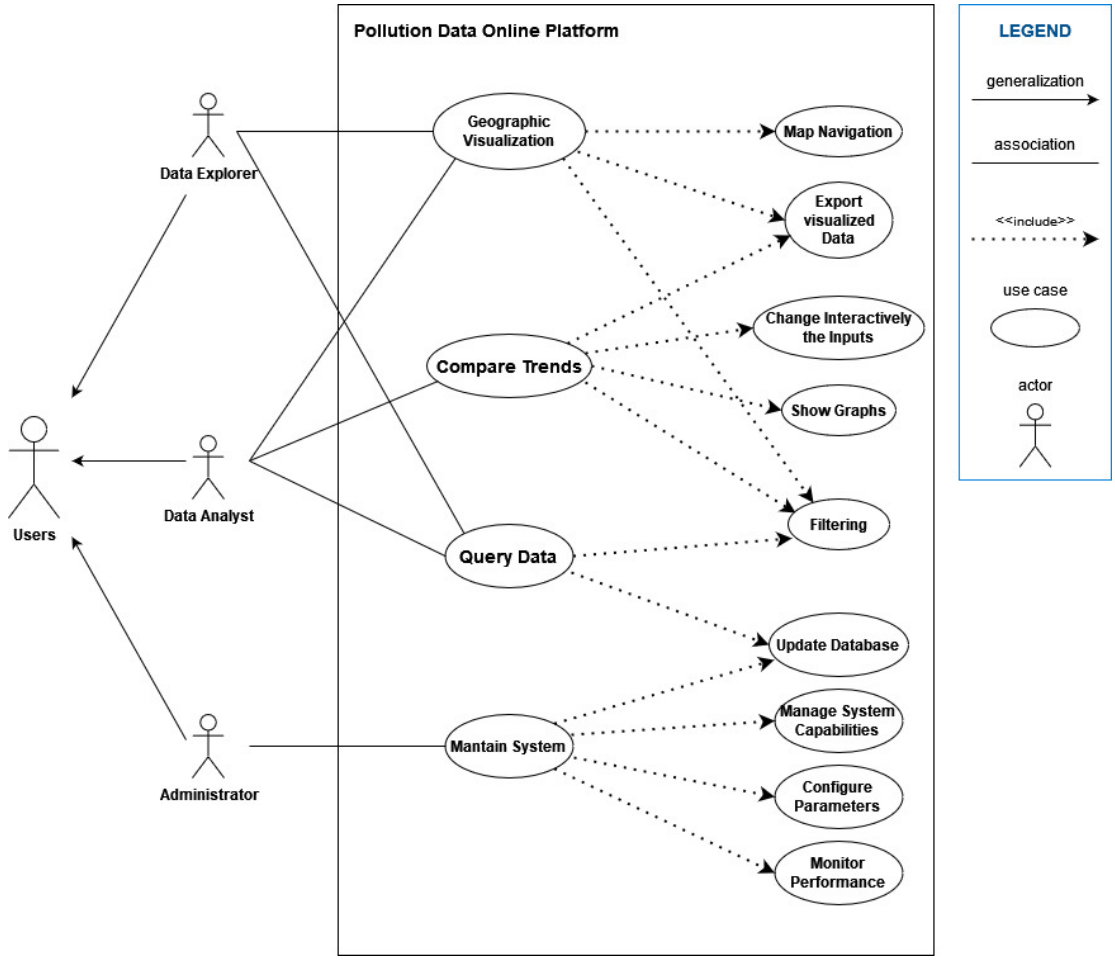
3.1. User Actions

According to the functionalities that the platform is aiming to provide, in the following chart each user is profiled by the potential activities he could undertake in the service.

User Action	Administrator	Data Analyst	Data Explorer
1		x	x
2		x	x
3		x	x
4		x	x
5		x	x
6	x		
7	x		
8	x		x
9	x		

List of the actions: (1) General navigation – (2) Querying the data picking specific periods and variables (pollutant/weather) – (3) Visualize correlation between pollutants and weather conditions in specific place (histogram, chart value) – (4) Change interactively the inputs to observe AQI dynamics – (5) Export a graphical output (map/data charts) – (6) Configure system parameters (7) Manage storage and processing capabilities – (8) Update or modify database – (9) Monitor system performance

3.2. Use Case Diagram



3.3. Use Case stories

As an administrator, I should be able to update AQI data in our database and customize the user interface.	As a data analyst, I need to filter, sort, and view AQI data of cities based on specific criteria to analyze trends or identify patterns.	As a data analyst, I want to explore the AQI data across the top ten cities by GDP and visualize it through interactive charts and graphs.

3.4. Flow of events

Each user case would have a unique and respective flow of events.

Administrator:

1. Data Source Integration:

a. The administrator accesses the dashboard app and navigates to the data source integration section.

b. They connect the dashboard to external data sources, such as public APIs or data feeds, to automatically import the latest AQI data.

c. The administrator configures the data source parameters, including authentication, refresh intervals, and data mapping.

2. Dashboard Customization:

    a. The administrator selects the customization feature within the dashboard.

    b. They customize the layout, design, and visual components of the dashboard, including the selection of specific AQI data metrics to display prominently.

    c. The administrator saves the customization settings, and the dashboard is updated with the selected design and data metrics.

3. System Monitoring and Maintenance:

    a. The administrator accesses the system monitoring section to ensure smooth operation.

    b. They review system performance, check for any errors or issues, and analyze resource usage.

    c. The administrator performs routine maintenance tasks, such as database optimization, data backups, or software updates, to maintain the system's stability and security.

Data Analyst:

1. Data Exploration:

    a. The data analyst selects from the list of cities or group of cities in the dropdowns.

    b. They choose specific cities and time periods of interest for analyzing the AQI data.

    c. The data analyst applies filters or criteria to further refine the displayed data, such base map toggle.

2. Visualization and Analysis:

    a. Based on the selected data and filters, the data analyst views interactive line charts and maps representing AQI trends across the selected cities.

    b. They zoom in on specific time periods, hover over data points for details, and compare AQI levels between different cities.

c. The data analyst identifies patterns, anomalies, or correlations in the visualizations, gaining insights into air quality trends.

3. Export and Reporting:

    a. The data analyst selects a subset of the analyzed data and chooses the export option.

    b. They export the selected data as a PNG file to perform further analysis or generate customized reports.

    c. The data analyst creates summary reports or visualizations based on the exported data for sharing with stakeholders or making informed decisions.

Data Explorer:

1. Date and Cities Selection and Overview:

    a. The data explorer navigates to the date-range picker and selects a specific date(s) of interest from the calendar.

    b. They view an overview of cities' AQI data, including a stacked bar graph showing AQI levels per day over that time.

    c. The data explorer explores additional information, such as AQI levels for a specific city on specific date.

2. Comparative Analysis:

    a. The data explorer selects multiple cities from the dropdown feature to compare.

    b. They visualize AQI rankings using a bar chart or table, identifying the highest and lowest AQI level cities.

    c. The data explorer examines the changes in AQI rankings over time, identifying cities with notable improvements or deteriorations.

3. Correlation Analysis:

    a. The data explorer selects dates from the date-range picker to explore the correlation of average AQI and group of cities by GDP.

    b. They generate line graph plots to visualize and analyze the relationship between these variables over time

    c. The data explorer identifies potential correlations, such as the impact of GDP on AQI levels, and draws insights from the analysis.

These flows of events highlight the interactions and actions performed by users in each user case within the application. The specific functionalities and visualizations provided in your dashboard will shape the actual user experience and the flow of events.

4. Functional Requirements and Domain Assumptions

4.1. General user interface requirements

The goal of our web application is to enable a user-friendly way to analyze, compare, query, and visualize the AQI data of our 10 cities. As such, we offer some general UI requirements:

1. Intuitive and user-friendly interface dashboard, with clear navigation and easy access to all features and functions.
2. English-only interface.
3. Open access site that does not require user registration.
4. Interactive and engaging visualizations that allow users to explore and analyze air quality data in various ways, such as maps and charts.
5. Clear and concise documentation and help resources.
6. Potential integration with social media and other external platforms to allow users to share their custom analysis.

4.2. Functional requirements

The functional requirements consider all our use-cases. As such, it will ensure:

1. Data Ingestion: The application should be capable of ingesting air quality and weather sensor data from selected public digital archives, such as the World Air Quality Index project.
2. Data Storage: The application should be able to store large time-series datasets and associated metadata for each observation.
3. Data Processing: The application should be able to perform data cleaning, preprocessing, and harmonization to ensure consistency and quality of the data.
4. Data Retrieval: The application should expose a REST API to allow users to query the database and retrieve data in a timely and efficient manner.
5. Data Visualization: The application should provide users with interactive visualization tools that enable them to explore the data in different ways, such as maps, graphs, and possibly other visualization tools.
6. Customization/filtering: The application should allow users to customize their views of the data by selecting variables, time ranges, and other parameters.
7. Performance: The application should be scalable and reliable, able to handle a large volume of data and users.
8. Accessibility: The application should be hosted on a website to ensure wide accessibility.

4.3. Domain Assumptions

Our domain assumptions in developing this web application are as follows:

1. The air quality and weather sensor data retrieved from the public digital archives are accurate and reliable.
2. The data is publicly available and can be accessed through an API.
3. The database system used is PostgreSQL and can store large time-series datasets, as well as associated metadata for each observation.
4. The web server can handle many requests simultaneously, and is designed to be scalable, reliable, and maintainable.
5. The frontend dashboard is designed to be user-friendly and intuitive, allowing users to easily explore and visualize air quality and weather sensor data retrieved from the backend.

5. Conclusion

In conclusion, the Requirements Analysis and Specifications Document has laid the foundation for the development of our interactive AQI Dashboard. This document has effectively captured and outlined the functional and non-functional requirements, along with the constraints and assumptions, paving the way for a structured and efficient development process.

As the project progresses, we understand the importance of flexibility and adaptability in adjusting to potential changes in requirements. We are committed to ongoing improvement, aligning our application with the evolving needs of our users and advances in technology.

This comprehensive document has provided the springboard for the subsequent design and development phases of our Air Quality Index Dashboard. We look forward to the journey ahead in building out our vision for an interactive tool that can contribute meaningfully to global environmental awareness and urban livability.

References

Requirements Analysis Document Template. Available at: https://www.cs.fsu.edu/~lacher/courses/COP3331/rad.html.

Requirements Document Example. Available at: http://web.cse.ohio-state.edu/~bair.41/616/Project/Example_Document/Req_Doc_Example.html.