Politecnico di Milano

Geoinformatics Engineering
Software Engineering for Geoinformatics

Requirement Analysis and Specification Document

Group members:
Lorenzo Carlassara, Angelica Iseni, Emma Lodetti, Virginia Valeri

Professor:
Giovanni E. Quattrocchi

Accademic Year 2022 - 2023

Contents

1 Introduction **2**

1.1 Context and motivations . . . 2

1.2 Definitions, acronym, abbreviations . . . 2

1.3 Solution overview . . . 2

1.4 Scope and limitations . . . 2

2 Requirements **2**

2.1 Stakeholders . . . 2

2.2 Actors . . . 3

2.3 Domain Assumption . . . 3

2.4 Uses cases . . . 3

2.5 User stories . . . 4

2.6 Functional requirements . . . 5

2.6.1 Data Ingestion . . . 5

2.6.2 Web Server . . . 5

2.6.3 Dashboard . . . 5

2.7 Non-functional Requirements . . . 5

2.7.1 Performance . . . 5

2.7.2 Usability . . . 5

2.7.3 Security . . . 5

2.7.4 Scalability . . . 6

2.8 Constraints . . . 6

3 Bibliography **6**

1 Introduction

The purpose of this document is to describe the requirements for the development of an interactive client-server application that enables users to access, query and visualize air quality and weather sensor data retrieved from existing public digital archives. The system consists of three main components: a database to ingest and store the selected data, a web server (backend) to expose a REST API for querying the database, and a dashboard to provide means for requesting, processing (e.g. space-time aggregations), and visualizing data (e.g. maps, dynamic graphs, etc.).

1.1 Context and motivations

The agreement between the European Commission on electronic fuels will make it possible to market heat-powered vehicles even after 2035 when the ban on the sale of gasoline and diesel-powered cars begins, as long as they are powered by synthetic, climate-neutral fuels. European energy ministers also ratified by a majority vote the regulation on stopping gasoline and diesel engines in 2035. Public authorities collect air quality and weather observations in near-real time from ground sensor stations and store them in digital archives. Ground sensors data are composed of long time series of observations and sensors metadata, including coordinates, type of measured variable, etc. Often, observations from different sensors and/or providers require different patterns for data accessing, harmonization, and processing. Interactive applications and dashboards, capable of facilitating such tasks, are key for supporting both public authorities as well as ordinary citizens in data processing and visualization.

1.2 Definitions, acronym, abbreviations

- **Ordinary citizens:** refers to an average person who is a member of a particular community, society, or country and who is not distinguished by any special status, profession, or achievement. In the context of government and politics, the term is often used to describe the majority of the population who are not involved in the decision-making process or who do not hold any formal political power.
- **Air quality:** refers to the degree to which the air is clean, clear, and free from pollutants.
- **Ground sensors:** refers to the sensors installed on the ground that monitor air quality and weather conditions.
- **Public digital archives:** refer to online repositories or databases of digital information that are publicly accessible and maintained by governmental or public entities.

1.3 Solution overview

1.4 Scope and limitations

Observations from different sensors and providers require different patterns for data accessing, harmonization, and processing. This can make it difficult to access and process data from different sources. Additionally, the system may not be able to handle large amounts of data or complex queries. It is also important to note that the system relies on existing public digital archives to retrieve air quality and weather sensor data. If these archives are not up-to-date or do not contain all of the necessary data, this could limit the effectiveness of the system.

2 Requirements

2.1 Stakeholders

- Ordinary citizens
- European Environmental Ministries
- Automotive production companies
- Public authorities

2.2 Actors

- European Ministries
- Public authorities

2.3 Domain Assumption

- The user can find the GitHub link of the project repository.
- The users can send feedback to the team , to suggest improvements or express satisfiability with the service contacting us.
- The ground sensor stations are distributed geographically and collect data on a regular basis.
- The air quality and weather observations collected by the ground sensor stations are stored in databases.
- The data collected by different ground sensor stations may have different formats, structures, and levels of quality.
- The application needs to be able to access and harmonize data from different sources and formats, using standard protocols and tools.
- The application needs to be able to perform basic data processing and analysis tasks, such as data filtering, aggregation, and visualization.
- The application needs to be user-friendly and accessible to both public authorities and ordinary citizens, who may have different levels of technical expertise.
- The application needs to be scalable and able to handle large amounts of data, as well as multiple concurrent users.
- The application needs to be secure and protect sensitive data, such as personal information or confidential data.
- The application needs to be regularly updated and maintained, to ensure the quality and reliability of the data and services provided.

2.4 Uses cases

- Sign up
  Users can create a profile in the application by using their email address and generating a password.
- User login
  The users can log in with their credentials
- Data Visualization
  The user selects the desired location for data visualization. The application then retrieves the data from the database and presents it to the user in a graphical format. The user can interact with the data visualization, zooming in and out, selecting specific variables to view, and adjusting the time range of the data displayed.
- Data download
  Users can download graphical representation of data previously selected.
- Admin data management
  This use case diagram describes the process for an admin to manage the air quality data stored in the database. The admin logs in to the application and is presented with a dashboard displaying the current data collection status, as well as the ability to add or remove sensor stations, update metadata, and adjust the data processing algorithms. The admin can also view logs of data processing and system performance to monitor the overall health of the

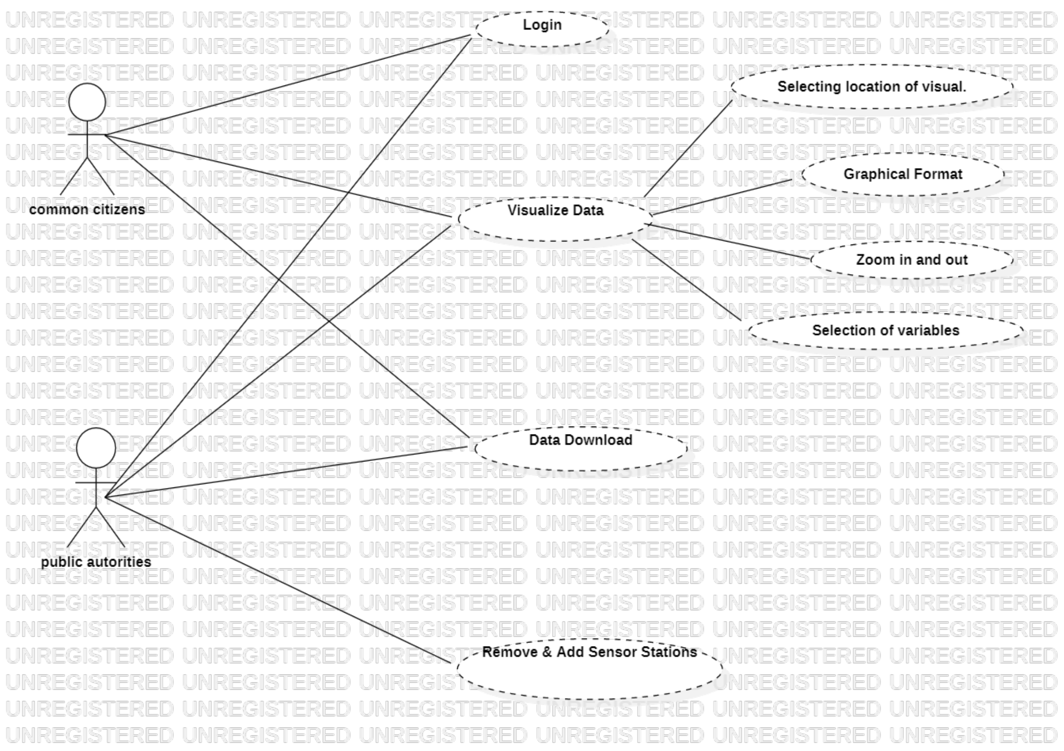

Figure 1: Uses cases diagram

2.5 User stories

- User login and data visualization:
  - As a user, I want to be able to log in to the application so that I can access air quality and weather data.
  - As a user, I want to be able to select the location for data visualization so that I can view data for a specific area.
  - As a user, I want to be able to interact with the data visualization so that I can zoom in and out, select specific variables to view, and adjust the time range of the data displayed.
- Data download:
  - As a user, I want to be able to download air quality and weather data from the application so that I can analyze it in other tools.
  - As a user, I want to be able to select the desired data variables and location so that I can download only the data I need.
  - As a user, I want to be able to download the data in a variety of formats (e.g., CSV, JSON, etc.) so that I can use it in different tools.
- Admin data management:
  - As an admin, I want to be able to log in to the application so that I can manage the air quality and weather data stored in the database.
  - As an admin, I want to be able to add or remove sensor stations so that I can update the data collection process.
  - As an admin, I want to be able to update metadata so that users have accurate information about the data.
  - As an admin, I want to be able to adjust the data processing algorithms so that users receive high-quality data.

2.6 Functional requirements

The following requirements must be met by the system:

2.6.1 Data Ingestion

- The system should allow data ingestion from public digital archives of air quality and weather sensor data.
- The selected dataset must be uploaded in the application database.
- Strategies for continuous data integration between the selected data archive and the database can be implemented and considered a plus, but they are not mandatory.

2.6.2 Web Server

- The system should provide a REST API that enables users to query and retrieve data from the database.
- The web server should perform data cleaning and preprocessing before returning the results to users.
- Data must be returned to users using the JSON format.

2.6.3 Dashboard

- The system should provide an interactive dashboard that enables users to process and visualize data retrieved from the web server using some original manipulation strategy.
- The dashboard should include both geographic content (map-based views) and attributes (interactive graphs).
- The dashboard should enable users to generate custom views of the data.
- Additional base maps and layer can be added to the map-based views and considered a plus, but they are not mandatory.

2.7 Non-functional Requirements

The following non-functional requirements must be met by the system:

2.7.1 Performance

- The system must handle a large volume of data, and queries should be processed efficiently.
- The system should provide timely and accurate data retrieval and processing.

2.7.2 Usability

- The system should have a user-friendly interface that enables users to easily access and interact with data.
- The dashboard should be visually appealing and enable users to easily generate custom views of the data.

2.7.3 Security

- The system must ensure data privacy and security.
- The system should implement proper authentication and authorization mechanisms.
- The system should enforce access controls to restrict access to sensitive data.

2.7.4 Scalability

- The system should be designed to scale horizontally and vertically.
- The system should be able to handle an increasing number of users and data sources.

2.8 Constraints

- The system must comply with data privacy regulations and standards.
- The system should be developed using open-source software.
- The system must be developed using Python programming language.
- The system must be hosted on a cloud platform.

3 Bibliography