POLITECNICO
MILANO 1863

Software Engineering
for Geoinformatics 2023

Requirement Analysis and Specification Document

EarthTech Group RASD Presentation by

Mohammad Kordgholiabad

Fatemeh Soleimaniansomarin

Ghulam Abbas Zafari

EarthTech

Air Quality Monotoring

Deliverable: RASD

Title: Requirement Analysis and Verification Document

Authors: Mohammad Kordgholiabad, Fatemeh Soliemaniansomarin
Ghulam Abbas Zafari

Version: 1.0
Date: April 17, 2023

Revision history

Version	Date	Change
1.0	17/4/2023	First submitted version

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this project is to outline the requirements and verification process for developing an interactive web application that presents air quality data. The web map will allow users to view, analyse, and visualize air quality data in an interactive and informative manner. The web map will be accessible via a web browser and will provide real-time or near-real-time data on air quality levels in a specific cities such as Paris, Amsterdam, Barcelona, Milan, and Dublin.

Air pollution is a critical environmental issue that affects the health and well-being of communities worldwide. With the help of data visualization, we can better understand the distribution and magnitude of air pollution and identify areas that require attention for mitigation and remediation efforts.

1.2 Scope

The scope of this project or an interactive web map is a powerful tool for visualizing air quality data, enabling users to explore, analyse, and gain insights into air quality indexes.
In this web application we used prediction model which utilize historical data, real-time data from air quality monitoring stations, weather data, and other relevant data sources to make accurate predictions.

1.3 Short overview

Air pollution is a significant global challenge that requires collective efforts from individuals, communities, governments, and industries to address. By taking steps to reduce our contribution to air pollution and advocating for policies and practices that promote cleaner air, we can help protect the environment and safeguard our health for generations to come.

In a time of information saturation and global climate crisis, there is a need for distributed accessible high-quality information. This combined with the increasing propensity for the individual to find their own specific information sources tailored to their own needs means that tools are needed which allow individuals to cross-reference the information they receive using reliable and objective sources.

The application to be developed, therefore, aims to be a tool that provides easy access to air quality data in order to empower them both in their personal lives and as members of a community. The application is not a source in and of itself, but rather a visualization tool for existing and established data sources. Therefor to create this web map, we used modern web technologies such as python, CSS, JavaScript, and mapping libraries.

1.4 Anacronyms and other definitions

“The web app” or “app” : The application that is being developed.
“Use case” : An example of user interaction with the system.
“User” : any individual using the website that satisfies the description under the “ user characteristics” section.
“Bookmark” : Saving a snapshot of the information presented at a location on the map.
JSON : JavaScript Object Notation.
JS : javascript.
df : Dataframe, a pandas dataframe object.
gdf : geodataframe, a geopandas dataframe object
RASD : Requirements Analysis and Specification Document, easy later access.
“OpenAQ” : An organization providing open air quality data through API (OpenAQ, n.d)
“API” : Application Programming Interface

"REST" : Representational State Transfer

“Python” : 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.

"AQI" : Air Quality Index.

2 Application domain and Use Case:

2.1 Users:

Our major user is the public, like students, families, clerks and workers. This application can be used for non-expertise uses, helping people to check the weather for their trips, camping and their other routine and daily plans. Also for checking if the air is polluted or not, so they can decide whether or not live in a region or go on a trip.

This makes our application to be user-friendly and as simple and as clear as possible, since our users are the public.

2.2 Dataset:

Our data is collected from different stations of the cities below, which are popular cities in Europe and are visited by tourists and other people a lot:

1. Paris
2. Amsterdam
3. Barcelona
4. Milan
5. Dublin

The collected data consists of below information:

Air quality data: $PM_{2.5}$, PM_{10}, O_3, NO_2, and SO_2

Weather data: Temperature, Pressure, Humidity, andWind

The data can be shown to the users in different ways:

1. on the map: By keeping the mouse cursor on the desired location on the map, user can see a brief pop-up of the latest AQI and Temperature of the location.

2. on a diagram: By choosing the location, user can see a time series of changes in a location over time. They can also choose two locations and compare their diagram.

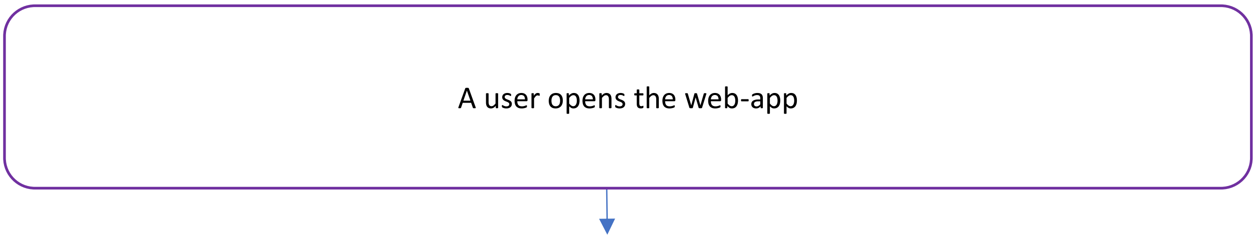
3. by choosing time and location: User chooses the city from the search bar and chooses the time and sees the result.

2.3 Use case:

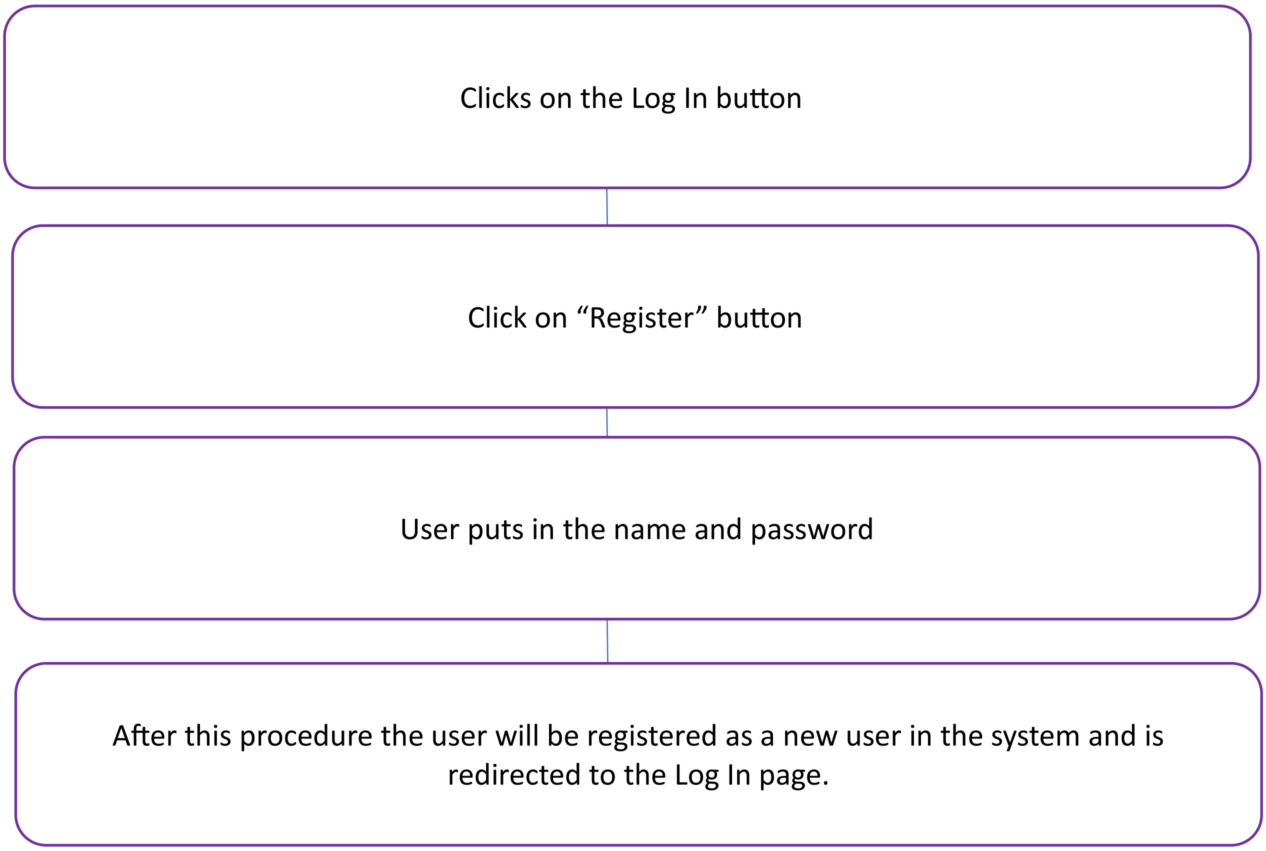
Here are some major user cases of operations that registered and non-registered users might do.

2.3.1 Register as a user:

User: not-registered users



A user opens the web-app



Clicks on the Log In button

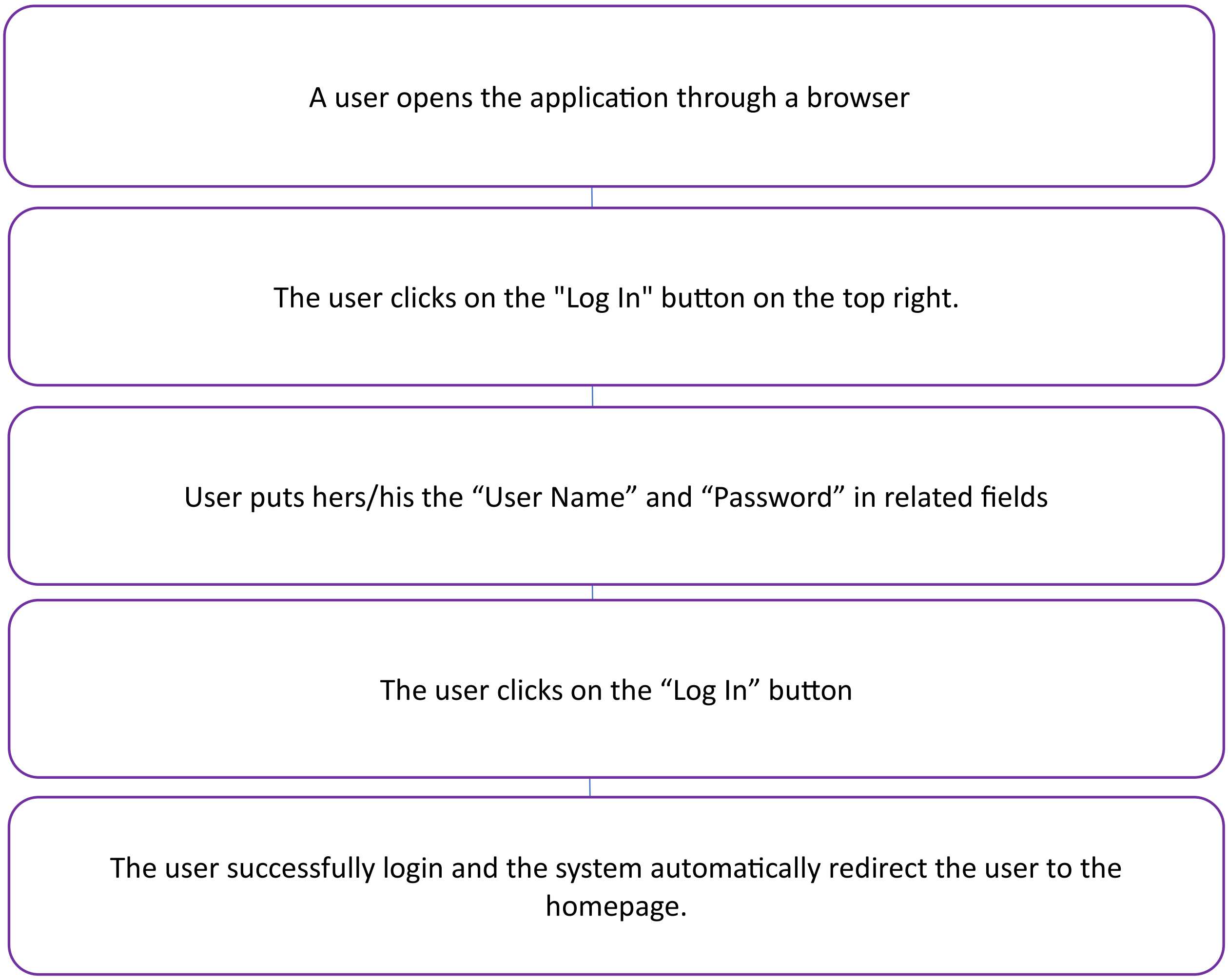
Click on “Register” button

User puts in the name and password

After this procedure the user will be registered as a new user in the system and is redirected to the Log In page.

2.3.2 Login as a user:

User: registered users



A user opens the application through a browser

The user clicks on the "Log In" button on the top right.

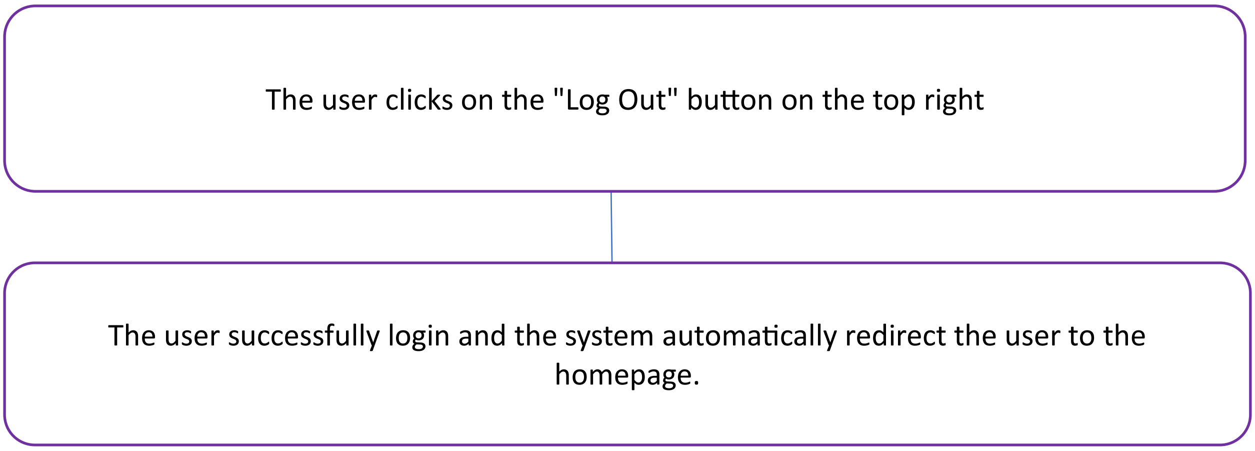
User puts hers/his the “User Name” and “Password” in related fields

The user clicks on the “Log In” button

The user successfully login and the system automatically redirect the user to the homepage.

2.3.3 Log out:

User: registered users

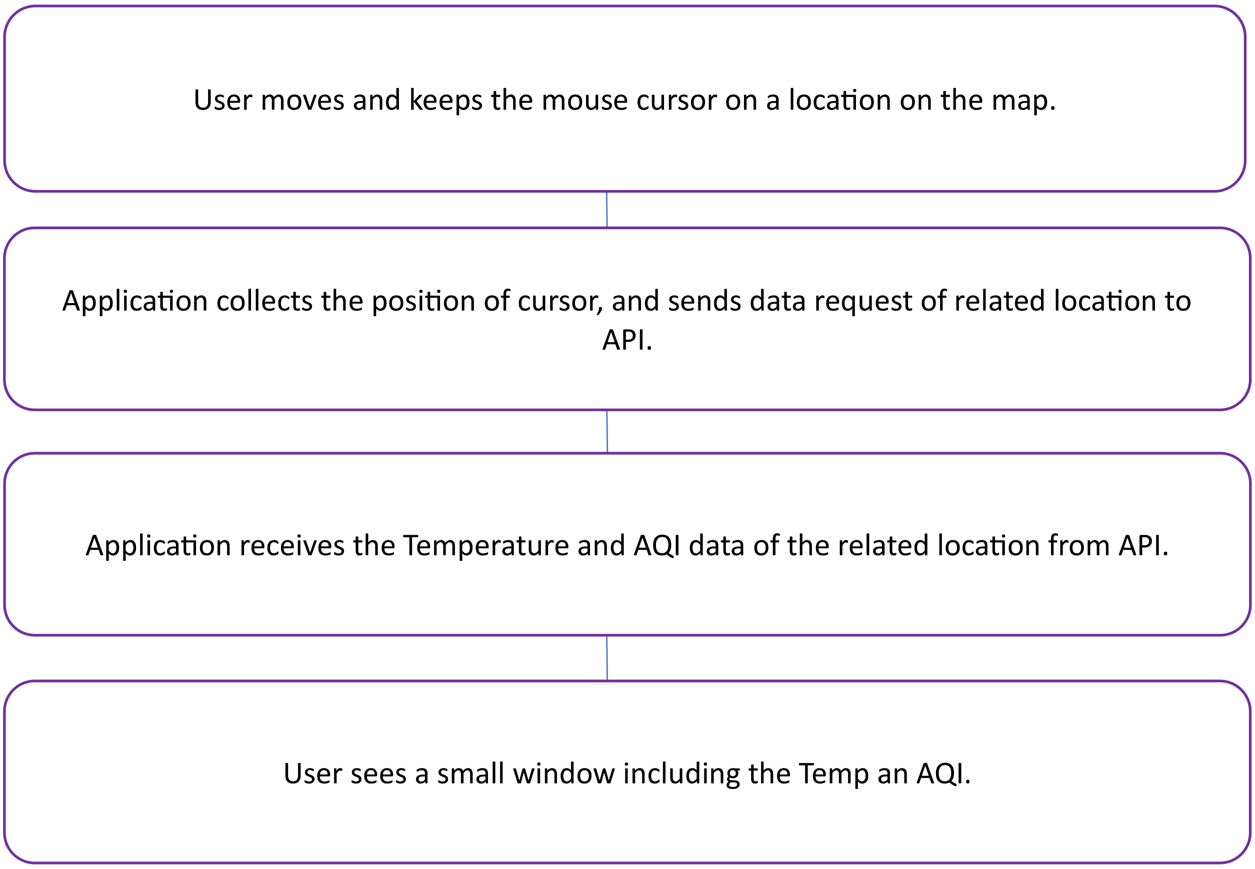


The user clicks on the "Log Out" button on the top right

The user successfully login and the system automatically redirect the user to the homepage.

2.3.4 Pop-ups on the map:

User: not-registered users
Exit condition: user sees the needed data.



User moves and keeps the mouse cursor on a location on the map.

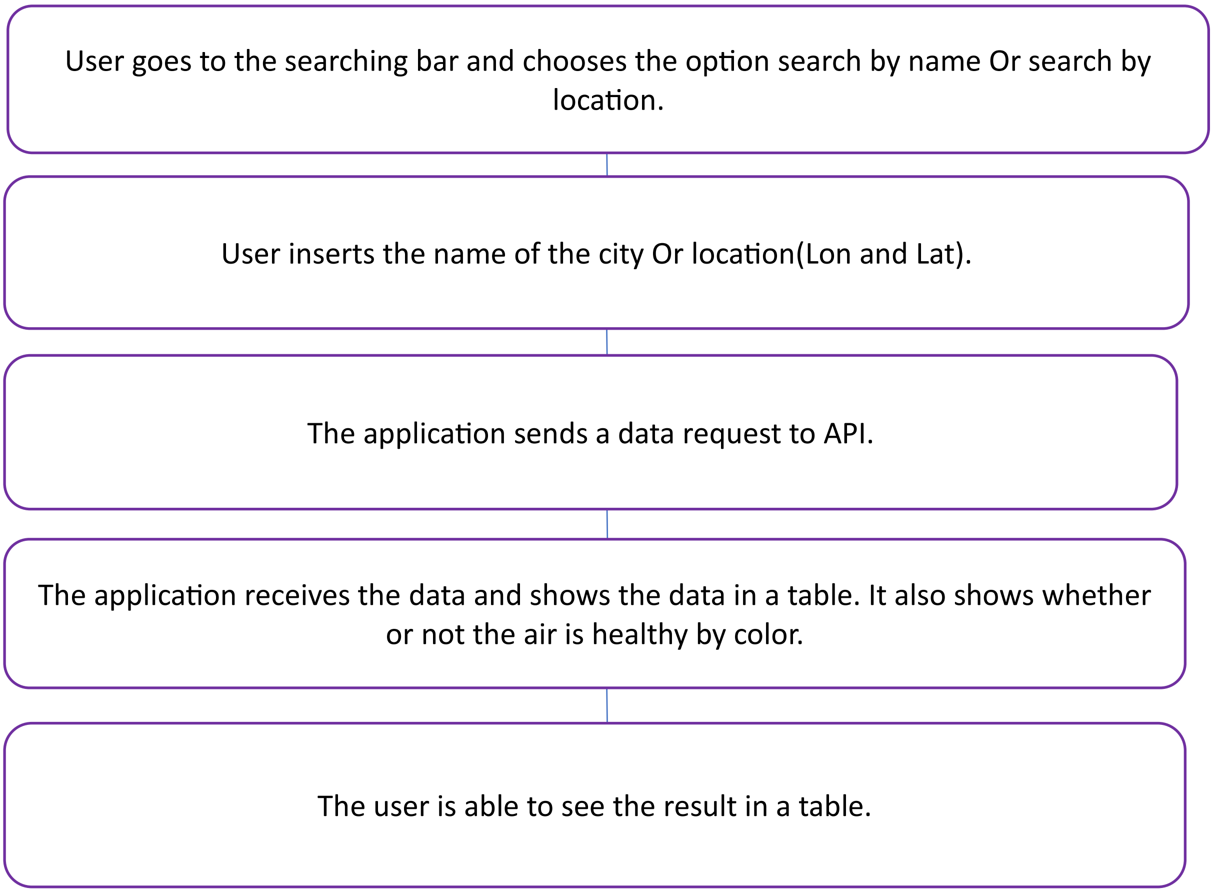
Application collects the position of cursor, and sends data request of related location to API.

Application receives the Temperature and AQI data of the related location from API.

User sees a small window including the Temp an AQI.

2.3.5 Searching:

Exit condition: User sees the correct data with wanted location.
User: registered users



User goes to the searching bar and chooses the option search by name Or search by location.

User inserts the name of the city Or location(Lon and Lat).

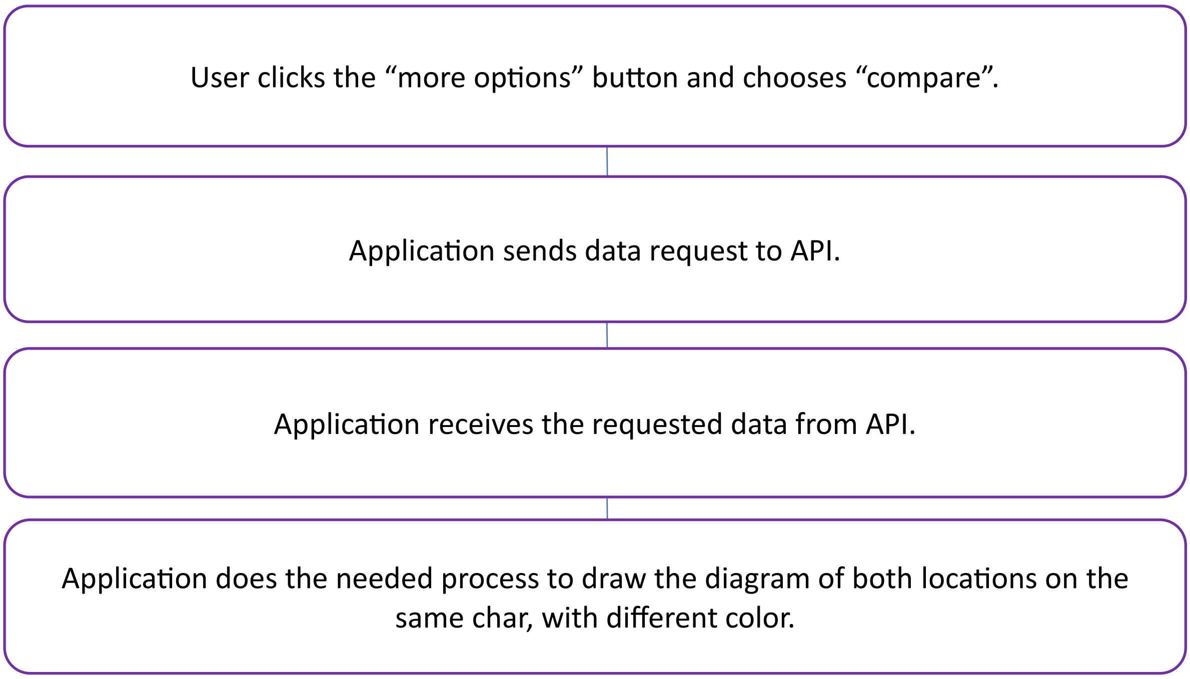
The application sends a data request to API.

The application receives the data and shows the data in a table. It also shows whether or not the air is healthy by color.

The user is able to see the result in a table.

2.3.6 Comparing data:

Exit condition: user sees the diagram with desired information.
User: registered users.

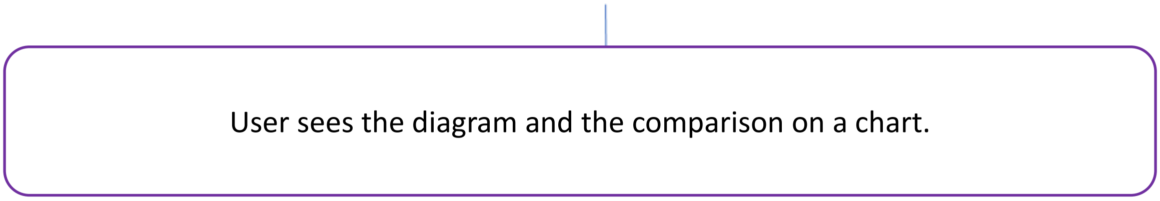


User clicks the “more options” button and chooses “compare”.

Application sends data request to API.

Application receives the requested data from API.

Application does the needed process to draw the diagram of both locations on the same char, with different color.



User sees the diagram and the comparison on a chart.

3 Requirements

3.1 Functional Requirements

Functional requirements describe the functions and features that "web app" plan to have in order to meet the needs of our users. Here are some functional requirements for our air quality monitoring application:

1. **Map Feature**: The system shall provide a map-based view of the sensor data for visualizing the geographic distribution of air quality and weather observations.

2. **Dashboard Feature**: The system shall provide a dashboard that enables users to visualize and analyze the sensor data in various formats.

3. **Log In/Log Out Feature**: The system shall provide a secure login functionality for registered users to access the system's features and data. It shall also include a log-out feature to log out the user securely. Also, it shall include a feature to modify the account like changing the password.

4. **Data Ingestion**: The system shall be able to ingest air quality and weather sensor data from public digital archives. The system shall be able to process the data and store it in a database.

5. **Data Retrieval**: The system shall provide a REST API for querying and retrieving air quality and weather sensor data from the database.

6. **Data Visualization**: The system shall provide an interactive dashboard for visualizing air quality and weather sensor data. The dashboard shall allow users to create custom views of the data. Also, dashboard shall allow users to interact with the data, such as selecting different cities.

7. **Data Processing**: The system shall support various data processing functions, such as analyzing trends or sorting the maximum and minimum amount of each factor.

3.2 Non-Functional Requirements

Non-functional requirements define the quality attributes, attributes that cannot be directly traced to the functionality of the software, such as performance, usability.

1. **Usability**: The application should be user-friendly and easy to navigate, with clear instructions and feedback provided to users.

2. **Performance**: The application should be able to handle the predicted volume of user requests and data processing in the fastest time possible.

3.3 Technological Requirements

Technological requirements are the hardware and software specifications that are necessary to support the system's functional and non-functional requirements.

1. Database: PostgreSQL
2. Web server: Flask
3. Dashboard: Jupyter Notebooks, Jupyter widgets
4. Programming language: Python 3.x
5. Data processing and utilities: Pandas, Geo pandas

3.4 Domain Assumption

1. The users can find the GitHub link of the project repository.
2. The system assumes that the web server and database will be hosted on reliable and secure servers with high-speed internet connectivity.
3. The system assumes that users accessing the application have basic knowledge of air quality and weather measurements.
4. The system assumes that the selected public digital archives have a stable and consistent API for accessing data.
5. The system assumes that the data provided by the public digital archives are available for free and do not require any additional licenses or fees.
6. The system assumes that the application will only be used for monitoring and visualization of air quality and weather data and will not be used for any other purposes.

4 Refrences

- https://cordis.europa.eu/docs/projects/cnect/3/609023/080/deliverables/001-MYWAYD21RequirementSpecificationAndAnalysiscnectddg1h520142537023.pdf
- https://webeep.polimi.it/pluginfile.php/968190/mod_folder/content/0/RASD-1.pdf?forcedownload=1
- https://webeep.polimi.it/pluginfile.php/968190/mod_folder/content/0/RASD-2.pdf?forcedownload=1
- Robert L. Lux and C. Arden. Pope. Air quality index. U.S. Environmental Protection Agency, 2009.