POLITECNICO MILANO 1863

Software Engineering for Geoinformatics 2022

Requirement Analysis and Specification Document

Pollution prediction by

Bigai S., Calcaterra M., Leonardi J., Rzewuski M., Zallemi N.

AIR Cloud

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Bigai S., Calcaterra M., Leonardi J., Rzewuski M., Zallemi N.
Version:	2.0
Date:	June 7, 2022
Download page:	https://github.com/Rkomi98/S4GEO
Copyright:	Copyright © 2017, Bigai S., Calcaterra M., Leonardi J., Rzewuski M., Zallemi N. – All rights reserved

Contents

List of Figures

List of Tables

AIR Cloud

1 Introduction

1.1 Purpose

The main purpose of our project is to involve and inform the users about the air pollution scenario of the 5 most polluted cities in Europe, which are specified below, through a web application planned as an Open-Source.
The topic we are treating, nowadays is essential and crucial in every field, like urban planning, product marketing, and most importantly environment and health.
Here we can see some examples of how much this topic is important nowadays:

"When exposed to particle pollution, patients with COPD usually have more emergency room visit, hospital admission, or even death in some cases."([1]).

Chronic Obstructive Pulmonary disease (COPD) is a lung disease characterized by long-term respiratory symptoms and airflow limitation. Air pollution impedes breathing for patients for whom it is already hard to breath.

"*Air pollution can affect asthma prevalence, onset, symptoms, and the reaction to treatment. Air quality plays an important role in the inception of asthma early in life and as trigger of asthma exacerbations in later time.*"([2])

"*Although it might be difficult to define an air quality sensor network's place in the smart city, innovative communities are showing that place does exist. The coronavirus crisis has demonstrated the importance of improving air quality instead of dismissing it as something that is nice to have*." by Grant Samms, research analyst at Guidehouse Insights -June 25, 2020.

"*The global air quality monitoring systems market is valued at USD 4.4 billion in 2021 and is expected to reach USD 5.9 billion by 2026, at a CAGR of 6.5% during the forecast period.*" in Dublin, Sept. 14, 2021 by Hòobe Newshire.

1.2 Scope

The scope of this project is to design and develop a web-based application that gives the possibility to users to query, visualize, analyze data, and save their analysis of the chosen data. The data are retrieved from the web, by virtue of explicit REST API(s) and contain information about certain pollutants, temperature, and humidity, from urban sensors of the five most polluted cities in Europe[1], specifically:

1. Skopje;
2. Krakow,
3. Belgrade;
4. London,
5. Paris.

The software let the user to see the analysis done by our application and see the possible correlation between pollutants (like carbon monoxide, ozone and Nitrogen dioxide) and some environmental factor (like relative humidity, temperature and pressure)

The dataset comes from `https://aqicn.org/`, which redirect to the specific environmental agency which treats data for the specific city. We will see more in deep in 2.2

[1] According to this website and to this one. We chose only big polluted cities, so with an high population and with an high value of AQI

1.3 Short overview

In a more detailed view, the software will contain functionalities to visualize data chosen by the user with different tools such as:

- with an interactive map centered in the city.
- classical data view in a metadata table

Moreover, the software will offer processing functionalities, such as:

- basic statistics about the air quality and all the pollutants.
- comparative analysis between two pollutants with other pollutants or atmospheric factors.
- predictions about the air quality is implemented in "beta form" in a jupyter notebook. We focused our analysis on a simple Linear regression and Gaussian processes.

1.4 Ancronyms and other definitions

On the table below we have listed some ancronyms and terms, we are going to use on this document, which will make the document more straightforward.

Name	Definition
Web application	It is an application program or software that is stored on a remote server and runs or is accessible over the Internet through a web browser.
REST	Representational state transfer is a software architectural style that was created to guide the design and development of the architecture for the Web. It defines a set of constrains for how the Web should behave.
REST API	It is an Application Programming Interface that satisfies the constrains of REST architecture an allows the interaction with RESTful web services.
GitHub	Web-based hosting service used for the development of our Open- Source project.
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.
PostgreSQL	It is an open-source Database Management System, which uses a client-server model where the server program manages the database files and accepts connections to the database from client applications.
DBMS	Software system we are going to use to store, retrieve, and run queries on data. Being an interface between the end-user and the database, it allows us to create, read, delete data in the database.
Choropleth Map	Shows interval data using shading, coloring, or placing of symbols within predefined areas, to indicate mean values of a specific quantity in those areas.
Cartogram Map	Shows geographical statistical diagrams of various quantities by using dots, curves, or shades.

Table 1: Acronyms and definitions

2 Application domain and phenomena description

2.1 Users

Based in our predictions the users can be divided into the following groups:

- The **main group of users**, according to our predictions, are municipalities, agencies, experts. They can use the application for:
  - environmental analysis
  - predictions about pollution that help in policy making, decision-making and urban planning.
- From a more economical point of view, we can also include as users **different companies**, which tend to increase their incomes from production strategy making. Having information about the specific pollutants they decide the proper materials for their products.
- On this wider group we can mention as users the **researchers** and **students**. They can access every tool of the app just out of their curiosity or for academic research.

2.2 Dataset

The dataset consists of 61 stations points which measure environmental pollution, temperature, pressure R.H and Wind coming from the AQCIN website. In our analysis we focused in the 5 most polluted cities in Europe, particularly in:

1. Skopje (6 stations);
2. Krakow (4 stations);
3. Belgrade (20 stations);
4. London (21 stations);
5. Paris (10 stations).

As we have already said in the last section, the cities have been chosen considering the big[2] most polluted cities according to this website and to this one.
The main features of the dataset are summarized below:

- **Point position**: expressed both in WGS84 geodetic coordinate reference system (latitude and longitude in decimal degrees) and UTC zone 43N projected coordinate reference system (east and north in meters).
- Then we have measurements of the following pollutants and other environmental factors with min, max and current value:
  - AQI (Air Quality Indicator)
  - PM 2.5
  - PM 10
  - O_3
  - NO_2
  - SO_2
  - CO
  - Temperature
  - Atmospheric Pressure
  - Relative Humidity
  - Wind

[2]Big means that the cities have at least 650000 people

Data collection Data can be retrieved in 2 different way:

- **Forecast data** consists in a table which has 7 rows:
  - on the first 2 there are the observations of the two days before the data retrieval.
  - on the other 5 rows the estimate of values of the day when the day has asked for the analysis and of the 4 days after.

  On the columns there are the minimum, maximum and average value of:
  - Ozone
  - PM 2.5
  - PM 10
  - UV index

- **Real time data** is a "single row table" with the values of the last measurement made by the sensors in the city chosen. On the column of the table the user can find the measurement[3] of
  - AQI (Air Quality Indicator).
  - Name of the city chosen.
  - CO: Carbon monoxide.
  - Relative humidity.
  - NO_2: Nitrogen dioxide.
  - O_3: Ozone
  - Atmospheric pressure
  - PM 10
  - PM 2.5
  - SO_2: Sulphur dioxide
  - Temperature
  - Wind
  - Latitude
  - Longitude
  - Geometry on the map

  Every time the user requests data from the API, it is automatically added to the local Database.

2.3 Operations

The operations included in the web application are divided into two categories: **User operations** and **System operations**.
The **user operations** are those that users are aware of and that can be enabled by interacting with the application. This way, the user will be able to perform operations in order to request information from the server or web services (as GitHub) by the web application, in particular:

- Request environmental data collected by the station points
- Download the data
- Visualize the data requested by different methods
- Sort and filter the data

[3] All these indicators are well explained in 2.5.1

The **system operations** are not visible to users. Essentially system operations are phenomena controlled by the machine in order to “run” the users operations, in particular:

- Web client – web server: server responds the browser requests with a page containing HTML, CSS, JS and Python code.
- Server responds the client requests by retrieving the data from the data base and returning it in JSON format

2.4 Product function

The software has some functionalities to visualize data chosen by the user:

- **Mapping Environmental Pollution Data**: Pollution data is showed in the "Create Project" page, through a point indicating the location of the data with the basic real-time data: the air quality, temperature, wind, atmospheric pressure, PM10, PM2.5 and the date of the data retrieval.
  It is also possible to visualize all the stations involved in the cities considered in our analysis. It

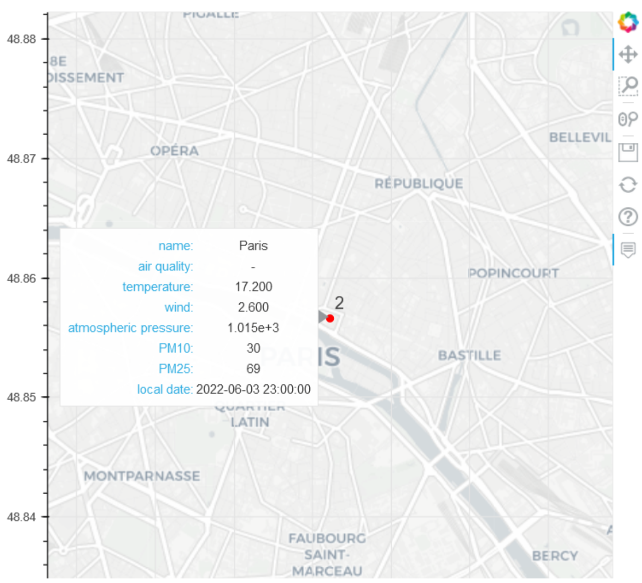

Figure 1: Mapping Environmental Pollution Data

  is showed in the "Data" section in the "More Info" page. Here for each station are listed the name of the station with the corresponding AQI value registered and the coordinates.

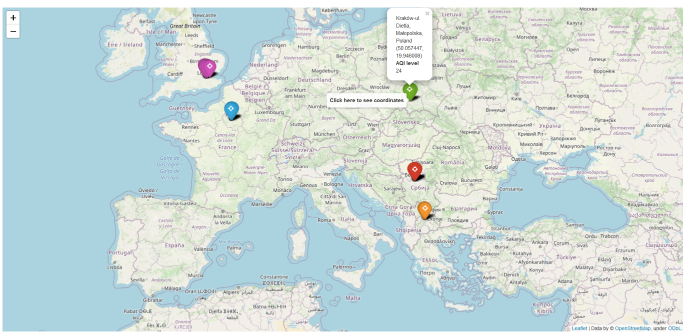

Figure 2: More info Map

- **Data Exploration**: Users will be able to select for each of the 5 available cities real-time and forecast data in a classical table. Furthermore, there will be the possibility to sort data in the table and filter data, choosing the column name, the filtering type, by using one of the following operators $\geq, >, ==, \leq, <$ and the value.
- **Data Analysis Tools**: In the website the user will generate graphs as well as basic statistics about the retrieved data by clicking the 'export data' button.

2.5 Relevant phenomena

In this section we will concentrate on the “real world” phenomena, those in the “machine world” and those that belong to both worlds. With well-defined relevant phenomena we will then be able to better understand the possible use cases and requirements and go into a deeper analysis of them in the following sections. With the issue of air pollution in big cities and the growing need of decision and policy making based on the condition of the air in a given place we will concentrate mainly on those phenomena.

2.5.1 World Phenomena

The most relevant ‘real world’ phenomenon to our study is the air pollution in a given city at a given time. The air quality is estimated in real time by parameters measured by sensors in the chosen city:

AQI Air Quality Index is a way of showing changes in the amount of pollution in the air. It is a measure of how clean or polluted the air is.

PM2.5 particulate matter 2.5 are tiny particles or droplets in the air that are 2.5 microns or less in width.

PM10 particulate matter 10 are tiny particles or droplets in the air that are 10 microns or less in width.

O_3 refers to ground-level ozone. Ground-level ozone comes from pollution emitted from cars, power plants, industrial boilers, refineries, and chemical plants.

NO_2 refers to nitrogen dioxide, which primarily gets in the air from the burning of fuel (emissions from cars, trucks, power plants etc.)

SO_2 refers to sulfur dioxide, which results from the burning of either sulfur or materials containing sulfur.

CO – refers to carbon monoxide. The main sources of CO to outdoor air are cars and other vehicles or machinery that burn fossil fuels.

T Temperature in degrees Celsius

P Pressure [hPa]

Φ Relative humidity [%]

v Wind [m/sec]

The second most important “real world” phenomenon, is the need of accessing, analyzing and understanding the air quality data by policy making actors and by other actors interested in the condition of the air. It could be done by various technological means, but we are focusing on a web app interface, from which (f.e.) a policy making actor can retrieve and analyze the data.

2.6 Machine phenomena

The most important elements are:

- **API requests** to retrieve real-time environmental data.
- **Database queries** to retrieve data from the database and store real-time data, results of analysis on data and comments on data in the database.

- **Data visualization** using maps, tables and graphs with tools provided by pandas and geopandas, bokeh and contextily. The latter is used to have a base map coming from OpenStreetMap.
- **Data manipulation** sorting by column header from the table, clicking on the header, and filtering data by column name, type and value.
- **Data analysis calculations** using Python packages such as "geopandas" for spatial analysis and "pandas profiling" to have a simple report analysis

2.7 Shared phenomena

In this section, we will introduce phenomena relevant in the intersection of the 'real-world' and 'machine-world'. They are described as user actions in the context of machine competences, they are going to be described in depth in the Use Cases section:

- **Register**: reserved for no registered users. A user can registered simply putting name and password (see tab 2).
- **Log In**: The registered user has first to login to use all the functionalities of our application (see tab 3).
- **Log Out**: The logged in user logouts of the system after finishing working with the system. (see tab 4).
- **Retrieve data from database** is a phenomenon combining the 'real-world' need of accessing environmental data with the 'machine world' capability of database queries (see table 5).
- **Visualize retrieved data** is a phenomenon that help users to understand the data with the possibility of visualizing data provided by the machine (see table 6).
- **Sort and filter data** combines the 'real-world' need to understand the data by the user and the 'machine-world' capability of data manipulation (see table 7).
- **Spatial analysis of data** combines the possibility of using models and algorithms provided by the 'machine world' with the collected data and the need to simulate and understand the environmental phenomena from the 'real-world' (see table 8).

3 Use case

In this section we will show the possible use cases of our software, showing software functionalities, possible users and considering possible exceptions in different situations.

Use Case 1: Register	
Name	Register
User:	All no-registered users
Condition:	The users can register, going on the website, clicking on "Register"
Flow of Events	The events follow this order: 1. A user opens the web-app 2. Clicks on the Log In button 3. Click on “Register” button 4. User puts in the name and password 5. After this procedure the user will be registered as a new user in the system and is redirected to the Log In page.
Exceptions	The user’s name already exists in the database.

Table 2: Register

Use Case 2: Log In	
Name	Log In
User:	Registered users
Condition:	The user has previously successfully signed up
Flow of Events	The events follow this order: 1. A user opens the application through a browser 2. The user clicks on the "Log In" button on the top right. 3. User puts hers/his credentials in the “Name” and “Password” fields 4. The user clicks on the “Log In” button 5. The user successfully login and the system automatically redirect the user to the homepage.
Exceptions	The user is not registered yet or has put the wrong name or password.

Table 3: Log In

Use Case 3: Log Out	
Name	Log Out
User:	Registered users
Condition:	The user has previously successfully logged in
Flow of Events	The events follow this order: 1. The user clicks on the "Log Out" button on the top right. 2. The user successfully login and the system automatically redirect the user to the homepage.
Exceptions	Log Out was unsuccessful due to browser error.

Table 4: Log Out

Use case 4: Query Data	
Name	Query Data
User:	Registered users
Condition:	The actor has accessed the web application through a browser, successfully logged in and has internet access
Flow of Events	The events follow this order: 1. The actor goes to the ‘New Project’ section 2. The actor starts a new query by choosing a city and the data type in the drop down menus 3. The query is processed 4. The system returns the query result as a table
Exit condition	The user retrieved the correct data from the database
Exceptions	The system crashed

Table 5: Query Data

Use case 5: Geographic visualization	
Name	Geographic visualization
User:	Registered users
Condition:	The entry conditions are 1. The actor has accessed the web application through a browser and has internet access. 2. The user has previously successfully logged in and then retrieved the needed dataset.
Flow of Events	The events follow this order: 1. The actor clicks on the ‘Visualize’ button next to the previously retrieved data tables 2. The user can export the map clicking on "save" button see 3
Exit condition	The system returns the correct visualization
Exceptions	The Openstreet map server or our system crashed

Table 6: Visualize and download the map

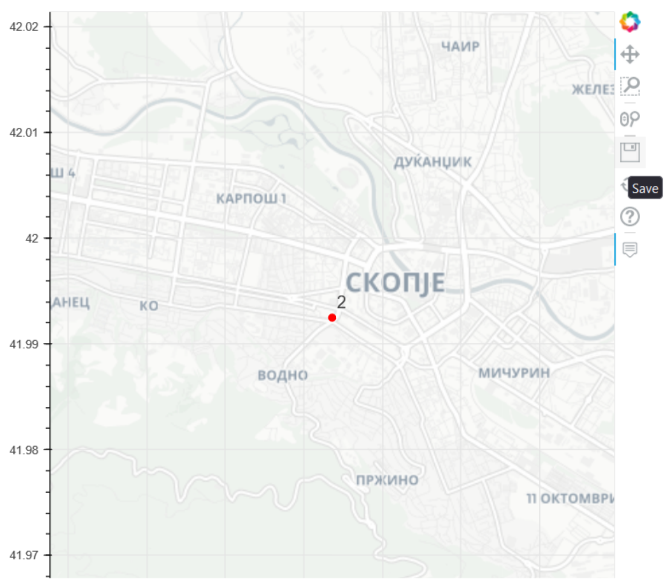

Figure 3: Save button

Use case 6: Data manipulation	
Name	Data manipulation
User:	Registered users
Condition:	The entry conditions are 1. The actor has accessed the web application through a browser and has internet access. 2. The user has previously successfully logged in and then retrieved the needed dataset, choosing one between "Forecast" and "Both".
Flow of Events	The events follow this order: 1. The user scrolls to the table retrieved from the API. 2. The actor chooses a sorting column, clicking on it. 3. The user can filter data, choosing the column name, the filtering type, by using one of the following operators $\geq, >, ==, \leq, <$ and the value. 4. The system returns the data or the visualization according to the sorting or/and filtering defined by the actor.
Exit condition	The system returns the correct sorted data.
Exceptions	The exceptions are: • The system crashed • The users sorting/filtering was not logically defined

Table 7: Sort, filter the data

Use case 7: Analyze data	
Name	Analyze data
User:	Registered users
Condition:	The entry conditions are 1. The actor has accessed the web application through a browser and has internet access. 2. The user has previously successfully logged in and then retrieved the needed dataset.
Flow of Events	The user can choose to visualize the analysis of the data in 2 different ways 1. Clicking on "Analyze" button. In this case the user will see some simple analysis on the data retrieved 2. Clicking on the "Export Analysis" button a new page is automatically opened. Here there is a small report on all the data retrieved.
Exit condition	The system returns the correct result of the analysis
Exceptions	The possible exceptions are: • The system crashed. • The parameters were not logically defined by the actor.

Table 8: Analyse the data

3.1 Use Case Diagram

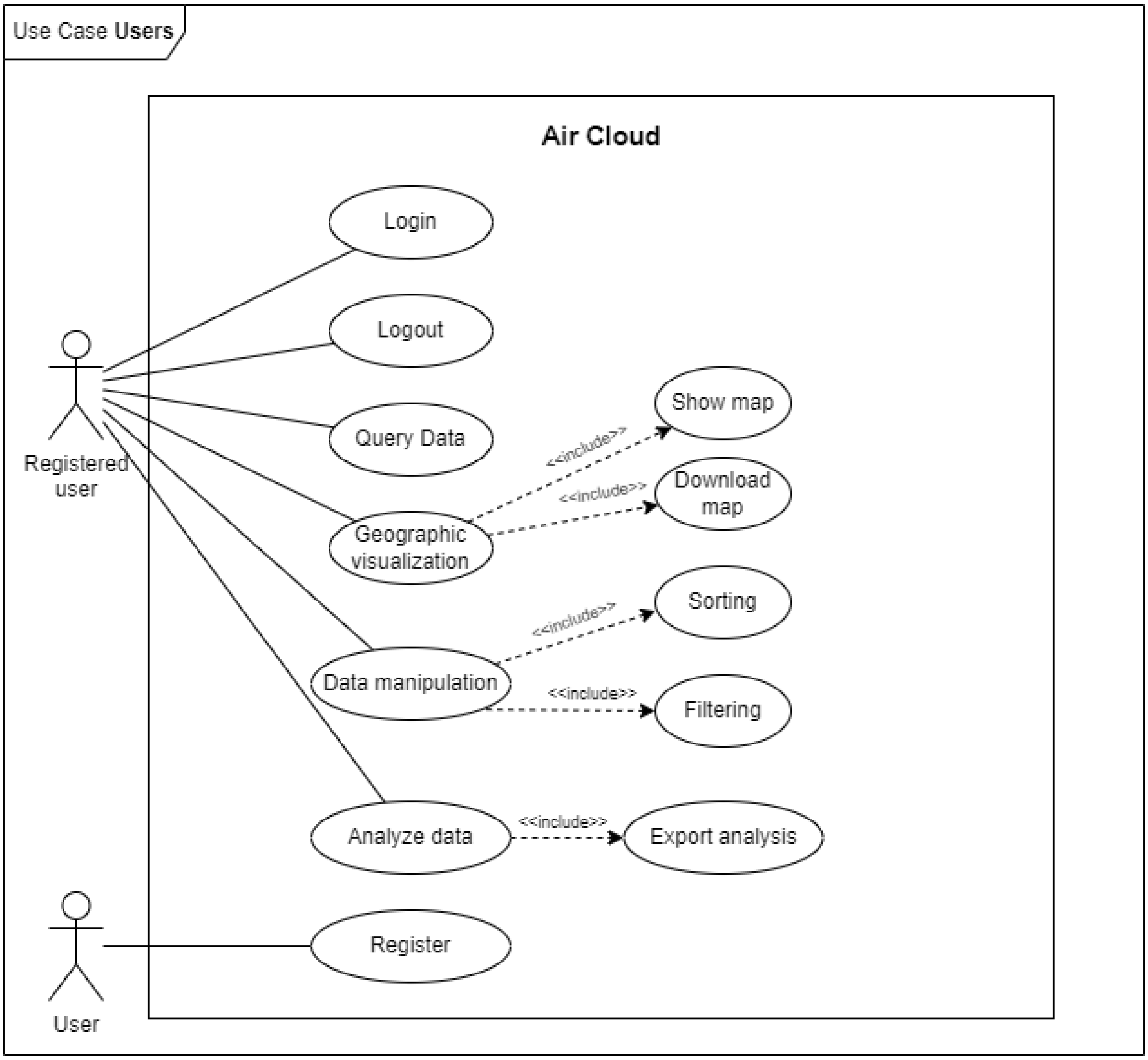

Figure 4: UML of our application

4 Functional requirements and Domain Assumptions

4.1 General user interface requirements

Our purpose is constructing a web application that offers to the users an attractive, easy to use and more importantly, a useful and comfortable environment, for analyzing and in all, for interpreting the data they are interested in. According to these properties we listed some general requirements:

1. The language of the website will be in English, since it is a common language for everyone.
2. Label buttons will be clearly specified to not confuse the user.

4.2 Functional requirements

Considering all the use cases, we can identify what are required functionalities of the system. User must be able to:

3. Query air quality data.
4. Sort and filter queried data.
5. Visualize data in form of a table and in a map realized in form of savable view.
6. Perform analysis on data.

4.3 Domain Assumptions

Furthermore we can include in this section the below Domain Assumptions:

1. The users can find the GitHub link of the project repository.
2. The users can send feedback to the development team, to suggest improvements or express their satisfiability with the service provided by the application contacting us on Linkedin. The user can find our Linkedin profile in the home page of the application

5 Effort Spent

Here we put some information on how much effort each group member spent in working at this document and what we did:

Name	Hours spent (h)	Details
Bigai	10	Operation in Application domain section
Calcaterra	10	Some changes in the Latex template, in the document: Dataset, Product function and Software used in Application domain section and the first 3 use cases
Leonardi	10	All the phenomena and all the use cases since the 4^{th} to the end
Rzewuski	10	UML of the application and some functional requirement
Zallemi	10	All the introduction, System and user in the Application domain section and most of the functional requirements

Table 9: Effort spent with details

References

[1] Robert L. Lux and C. Arden. Pope. Air quality index. *U.S. Environmental Protection Agency*, 2009.

[2] Teague WG and Bayer CW. Outdoor air pollution. *Pediatr Clin North Am*, 2001.

