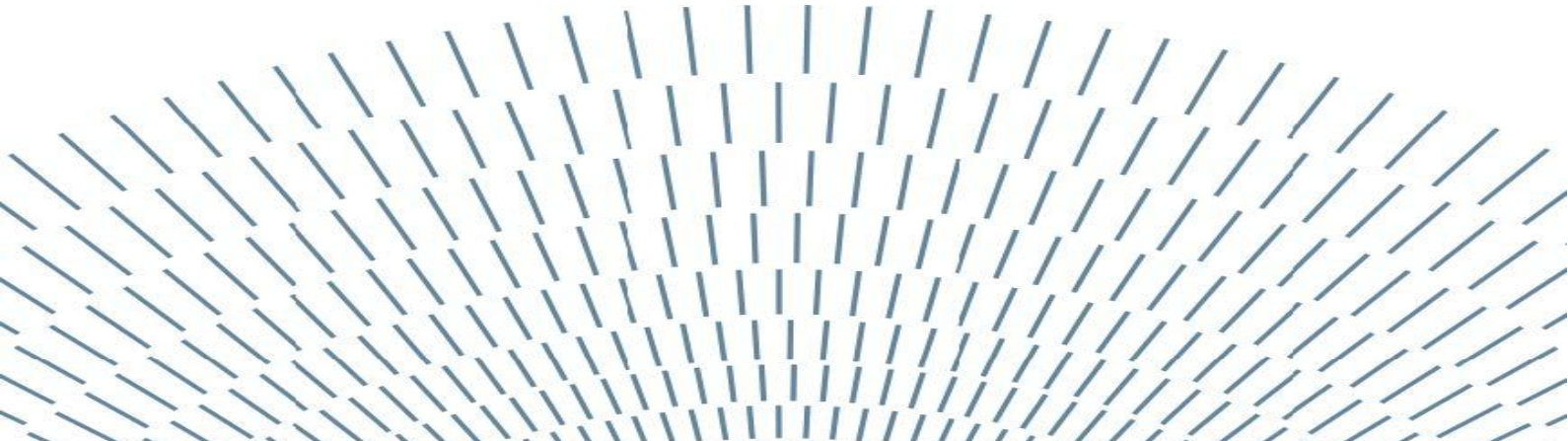
Politecnico di Milano
AA 2020-2021

Geoinformatics Engineering
Software Engineering for Geoinformatics Project

iUrban
RASD

Requirement Analysis and Specification Document
version 2.0 - 25/5/2021

Authors:	**Professors:**
Gao Jiawei	Di Nitto Elisabetta
Ding Guangyu	
Ernesa Ymeri	Oxoli Daniele
Song Xiangyang	



Section 1 - Project Scope and Goals

1.1 Project Scope

The scope of the project “Web-Based Geospatial Database Application on Urban Environmental Investigation in San Silvestre”, accomplished by a group of Geoinformatics students in Politecnico di Milano, is to design a web-based application, serving as a platform which not only displays the geographical contents in a map-based view, but also allows users to both access or retrieve existing data on “Urban Environmental Factors” (UEF in the following context) and contribute to the database of UEF in a given location through interactive mapping tools.

The application of our project is named “iUrban”. By using this application, users can find quantitative or qualitative data descriptions about the urban environment on the website, as well as instructions on how to contribute to the database. The permission of user-uploaded data is intended to make the application volunteer-based, so that the database can be continuously augmented. Nonetheless, the data uploaded by users must follow a predetermined regulation.

1.2 Project Goals

Firstly, the application “iUrban” of the project “Web-Based Geospatial Database Application on Urban Environmental Investigation in San Silvestre” enables users to visualize location-specific UEF data on a web page equipped with a base map.
Secondly, users can use the UEF data to perform certain data analysis, such as filtering, to inspire their decisions choosing the locations of their residential houses.
Thirdly, users can leave a comment on all the recordings, in which case this application acts as a forum for users to communicate with each other.
Finally, users can find descriptions and guidance on how to extract the arbitrarily-selected data and how to collect and upload the data in order to keep the database up to date and enlarged gradually.

Section 2 - Domain Analysis

2.1 Target Population

The web application “iUrban” of “Web-Based Geospatial Database Application on Urban Environmental Investigation in San Silvestre” is aimed at providing the UEF in the city for house buyers. And meanwhile municipalities could apply it on the urban health or urban economy thesis, for instance, to analyze the relationship between traffic and noise and make city park planning. It could also be conducive for research on urban environments.

This web application will be run online by the users, without interaction with other software or hardware.

Generally, our product is an application serving as a platform where users could monitor, update and exchange UEF information as references to help them make decisions according to their needs. To preserve the relevance of the website, log-in is required in order to access all the functions. As web managers, we have the privilege to remove any comment or user-uploaded data.

The representation of raw data in the web application will be performed by python scripts, and the libraries needed are as follows: shapely, pandas, geopandas for the representation of data; psycopg2 for SQL and also Flask for web implementation.

2.2 Project Database

The data for this project, including information about point positioning, will be retrieved from EpiCollect5, pre-processed and copied to a PostgreSQL database with a given frequency. The web app will then interact with DBMS and perform CRUD operations on the PostgreSQL database. The advantages for storing the data in a PostgreSQL database as opposed to fetch them directly from Epicollect5 include: i) to enable verification, pre-processing and storing of consistent data; ii) to ensure availability of data, decoupling our web application from EpiCollect5; iii) to reduce the risk of data loss; iv) to improve performance; v) to leverage DBMS capabilities and, in particular, the interface between Python and PostgreSQL.

In this project, we have chosen the dataset “SAN SILVESTRE GEOGRAPHY IA 2020” from the website Epicollect5. With 451 entries uploaded in the year of 2020, this dataset collects both the environmental and artificial geographical data in the city San Silvestre, such as temperature, humidity, wind direction, noise level and traffic count. Through processing and exposing the data above, users could choose an optimal location for residential, commercial, official areas as well as public service facilities.

Here is an example of a data entry:

Question	Answer
Name	Eileen Manning
Location	-12.110656, -77.034835
Date	12/10/2020
Time	15:30:00
Average noise level	55
Average light intensity	76894
Wind direction	W
Wind speed (Beaufort scale)	3
Cloud cover	8
Cloud type	Stratus
Visibility	2
Traffic count	5
Temperature	19
Humidity	68
Make a note of anything that may cause an anomaly	
Air pollution (number of particles) PM only.	2

Fig 1. Example of an Entry Showing the Attributes Table

A data entry contains the following attributes:

1) Name : The name of the surveyor;
2) Location: Geodetic coordinates (Latitude and Longitude) in WGS8[1] reference system. Coordinates are expressed in degrees, generated automatically by Epicollect5.
3) Date:DD/MM/YYYY format;
4) Time:hh:mm:ss format;
5) Average Noise Level: in dB;
6) Average Light Intensity[2]: is the perceived power per unit solid angle,in Candela (cd);
7) Wind Direction: reported by the direction from which it originates;
8) Wind Speed (Beaufort Scale): is an empirical measure that relates wind speed to observed conditions at sea or on land
9) Cloud Cover: refers to the fraction of the sky obscured by clouds when observed from a particular location
10) Cloud Type: Clouds are given different names based on their shape
11) Visibility:
12) Traffic Count: count of vehicular or pedestrian traffic, which is conducted along a particular road, path, or intersection..
13) Temperature: is a measure of the average kinetic energy of the particles in an object, in Celsius;
14) Humidity: is the amount of water vapor in the air,in fahrenheit;

[1] The World Geodetic System 1984 (WGS 84) is a 3-dimensional coordinate reference frame for establishing latitude, longitude and heights for navigation, positioning and targeting for the DoD, IC, NATO, International Hydrographic Office and the International Civil Aviation Organization.[1]
[2] Light irradiance collected by device

15) Note: anything that may cause an anomaly;
16) Air Pollution (Number of Particles): PM only.

Section 3 - Relevant Phenomena

3.1 Backgroud

With the growth in per capita income and living standard, people tend to choose locations of their own houses with more harmonious and pleasant environments surrounded. When making decisions for the residential location, people usually take such UEF into consideration as noise level, traffic conditions, temperature and humidity, which are crucial for the quality of living conditions.

However, the lack of a platform, where people have easy access to the information mentioned above, makes it difficult to find an answer to the dilemma in determining the optimal location for residences.

3.2 Phenomena

Depending on the conditions above, in this session we introduce both world phenomena and machine phenomena, and also the interactions between them.

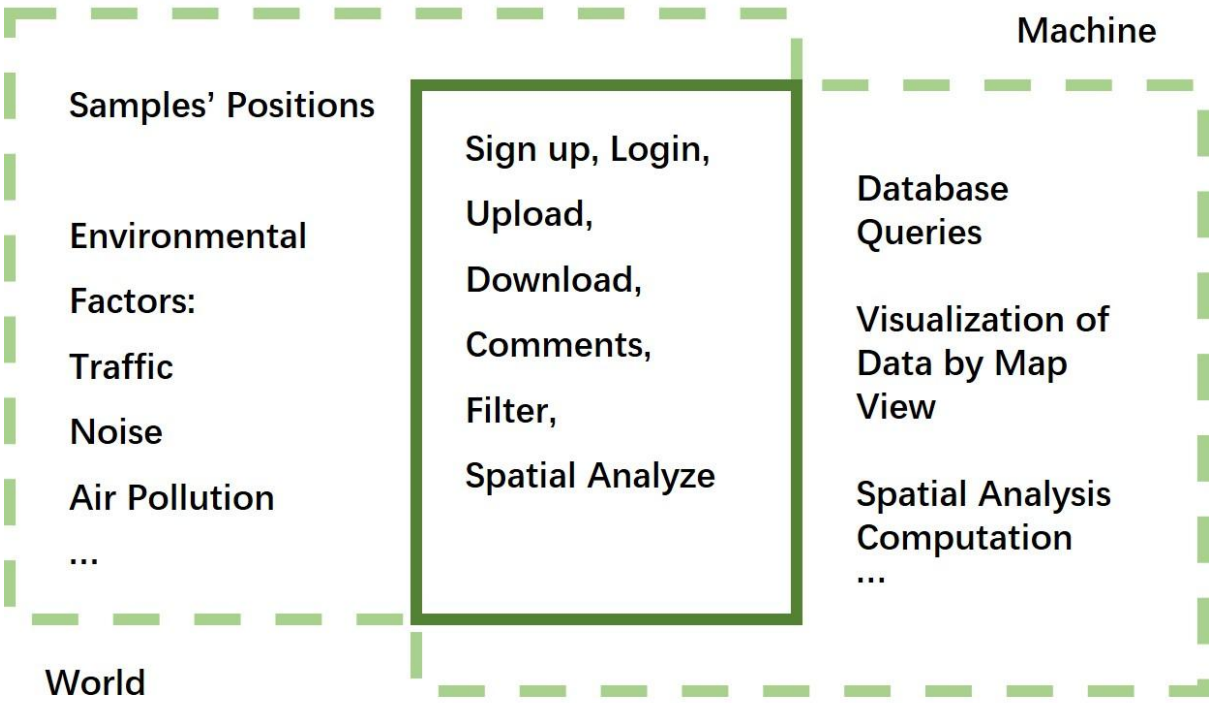


Fig 2. Relevant Phenomena

3.2.1 World Phenomena

The main world phenomena is the samples' positions and the environmental conditions.
Samples' Positions: The positions are collected by the contributors of the data using GPS devices, which refers to the coordinates of real places on the map.
Environmental Factors: Environmental Factors are recorded as the attributes of the samples such as traffic, average noise level, air pollution etc… (see details in Fig 1. Attribute Table of SAN SILVESTRE GEOGRAPHY IA 2020) .
Combining the two kinds of phenomena we could get the environmental data spatially distributed.

3.2.2 Machine Phenomena

Database Queries: using database queries to achieve the storage, updation, and retrieval of user and sample information.

Visualization of Data by Map View: using visualization of data by map view to distribute the samples on OSM basemap

Spatial Analysis Computation: some computation of spatial analysis is calculated

3.2.3 Shared Phenomena

SignUp: Process each new user will do by filling the fields with his data.

Login: The process user will do to go to their home page after they signed-up (If user is new) or the process user will do to go to their Home Page (If user already registered)

Upload, Download, Delete, Comments: Processes user can do after they signed up and logged in to our Web App

Filter by Environmental Conditions: Users are able to find specific samples by choosing the UEF conditions

Spatial Analyze: Users are able to use analytical tools such as statistical functions, to generate the properties of the samples

Section 4 - Use Cases

4.1 Use Cases Description

4.1.1 Register a user account

Name	Register a user account
Actors	Visitor
Entry Condition	The user has opened the application through a browser and has the internet access to connect and has access to the Web Browser on its device
Flow of Events	1. Open the application 2. Click on “Sign Up” button 3. Complies all the mandatory fields and provide the necessary information 4. Click on “Confirm” button 5. The system confirms the registration of the Visitor 6. The system return message
Exit Condition	The system returned the successful message. The user has successfully registered and now he’s able to use the application
Exceptions	1. The account that the user is used to register has already signed up, such as a username 2. The user didn’t fill all of the mandatory fields with valid data.

4.1.2 Login

Name	Login
Actors	User
Entry Condition	The user is previously successfully signed up
Flow of Events	1. The user opens the application through a browser 2. The user enters her/his credentials in the “Account” and “Password” fields of the home page of the app 3. The user clicks on the “Login” button 4. The user successfully login and the system automatically redirects to the function page
Exit Condition	The user is successfully redirected to the function page
Exceptions	1. The user enters invalid Account 2. The user enters invalid Password

4.1.3 Logout

Name	Logout
Actors	User
Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user clicks on the “Logout” button 4. The user successfully logout and the system automatically redirects to the login page
Exit Condition	The user is successfully redirected to the login page
Exceptions	Some system errors

4.1.4 Query Data by Filter Fields

Name	Query Data by Filter Fields
Actors	User
Entry Condition	The user has opened the application through a browser and has the internet access to connect and has access to the Web Browser on its device
Flow of Events	1. The user go to “Table” page 2. The user clicks on the “Search” box 3. The user inputs some filter fields 4. The system returns the query results in the table
Exit Condition	The system has successfully returned the results
Exceptions	1. There are something wrong for user’s inputs 2. There are something system wrong

4.1.5 Query Data by Range

Name	Query Data by Range
Actors	User
Entry Condition	The user has opened the application through a browser and has the internet access to connect and has access to the Web Browser on its device
Flow of Events	1. The user go to “Table” page 2. The user clicks on the “Query” button 3. The user inputs some search criteria in the query panel 4. The user clicks on the “confirm” button 4. The system returns the query results in the table
Exit Condition	The system has successfully returned the results

Exceptions	1. There are something wrong for user's inputs 2. There are something system wrong

4.1.6 Update Data

Name	Update Data
Actors	User
Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user clicks on the "Update" button 2. The user inserts or edits the field data 3. The user click on the "Confirm" button 4. The system check the type and numerical range of the input data 5. The system returns the check results, if there are something wrong, the user need to check their data again 6. The system returns the update results
Exit Condition	The system has returned the successful message
Exceptions	1. There are something wrong for user's inputs 2. There are something system wrong

4.1.7 Delete Data

Name	Delete Data
Actors	User
Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user go to the "Table" page 2. The user chooses the data that she/he has uploaded 3. The user clicks on the "Delete" button 4. The user click on the "Confirm" button 5. The system returns the delete results
Exit Condition	The system has returned the successful message
Exceptions	1. The user does not choose the data 2. There are something system wrong

4.1.8 Plotting Data

Name	Plotting Data
Actors	User
Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user clicks on the "Plotting" button

	2. The user clicks on the type of plotting she/he wants to visualize the data, such as, Chart, Graph or Map View 3. The user clicks the specific view Button and confirms 4. The system returns the result of the view option the user selected
Exit Condition	The system has successfully returned the results
Exceptions	There are something system wrong

4.1.9 Download Data

Name	Download Data
Actors	User
Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user clicks on the “Download” button 2. The user clicks on the “Confirm” button 3. The user chooses the location to download 4. The browser starts downloading
Exit Condition	The system has successfully download
Exceptions	There are something system wrong

4.1.10 Leave Comments

Name	Leave Comments
Actors	User
Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user chooses one data 2. The user views the comment of the data 3. The user inputs her/his comments 4. The user click on the “Confirm” button 5. The system returns the comment results and reflash this web
Exit Condition	The system has reflashed this page and displayed the comments
Exceptions	1. The user does not input the comments 2. There are something system wrong

4.1.11 Modify Comments

Name	Modify Comments
Actors	User

Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user chooses one data 2. The user views the comment of the data 3. The user click on the “Modify” button 4. The user inputs and edits the data 5. The user click on the “Confirm” button 6. The system returns the modify results 7. The system reflash this web
Exit Condition	The system has reflashed this page and displayed the comments
Exceptions	1. The user does not do the modify operation 2. There are something system wrong

4.1.12 Delete Comments

Name	Delete Comments
Actors	User
Entry Condition	The user has previously successfully logged in
Flow of Events	1. The user chooses one data 2. The user views the comment of the data 3. The user chooses her/his comments 4. The user click on the “Delete” button 5. The user click on the “Confirm” button 6. The system returns the delete results 7. The system reflash this web
Exit Condition	The system has reflashed this page and displayed the comments
Exceptions	1. The user does not do the delete operation 2. There are something system wrong

4.2 Use Case Diagrams

4.2.1 Visitor

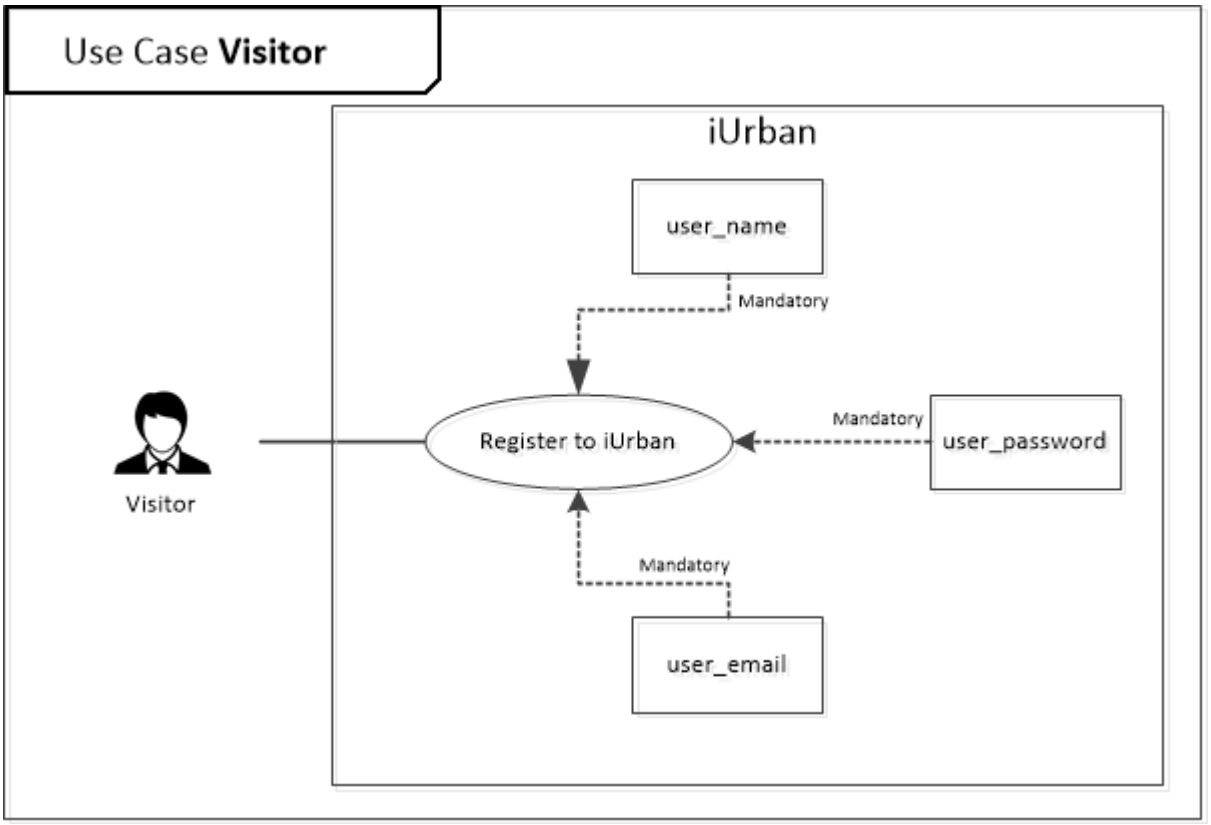


Fig 3. Visitor Diagram

4.2.2 User

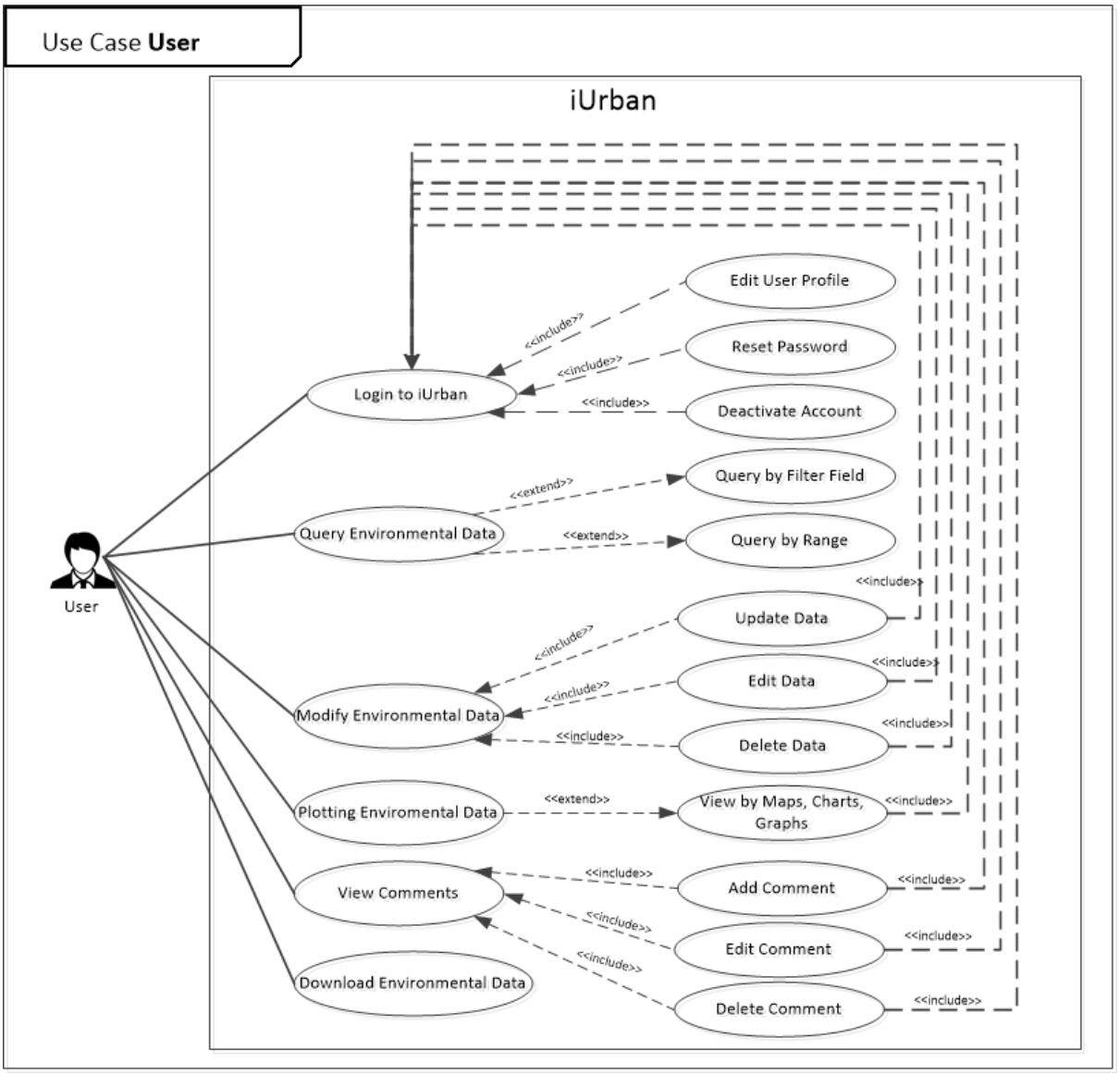


Fig 4. User Diagram

Section 5 - Requirements and domain assumptions

We can classify the requirements into three categories: functional requirements, non-functional requirements and technical requirements.

5.1 Functional Requirements

Base on the above user cases, the interactive user application must fulfill the following high priority functional requirements:

1) It must have an interactive map where layers of the data (measurement data points, heat map, dot map, etc.) and basemaps (e.g. OSM) can be selectable by the user in order to be visible.
2) Each measurement data point can be clicked by the user, so the details about that specific measurement are displayed (e.g.average noise, wind direction, photo of point clouds, temperature)
3) The raw measurement data table must be displayable in the website as well as available to be downloaded by the user in a set of formats (e.g. csv, shape, etc.).
4) The filters of the data can be personalized by the user. He/she can select a window of time, a specific spatial area or a filter based on a categorical or numerical data feature.
5) The user must be able to obtain descriptive statistics on the data as well as personalized plots using the available filters described previously.
6) The web-application may also implement the following functional requirements:
7) The user is able to find the GitHub repository in the page of the project.

Moreover, with reference to UC1, the web-application shall have a static section containing general information about noise pollution and instructions on how to contribute to data collection. This section will include hyperlinks for the users to navigate within the website or to access outside resources.

5.2 Non-Functional Requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability.

In this section the Non-Functional requirements will be described. These requirements define how a system is supposed to be. These requirements will be divided into two sections: the execution qualities and the evolution qualities.

5.2.1 Execution Qualities

1) In terms of security, the website must not put on risk the user while browsing or downloading the information, it is important that it is not possible for an external attacker to use the website in a malicious manner, modifying the information and affecting the user's integrity.
2) Regarding usability, the user may be able to download data from the website. The system shall always provide a safe environment which ensures that the downloaded content is virus and malware free.

3) The web application shall be a safe environment regarding data veracity, it must always display the real information uploaded to the EpiCollect5 interface without any malicious modification that can attempt against the data integrity.

5.2.2 Evolution Qualities

During the development and deployment, the web application must be subject to several quality checks that will ensure that the requirements and final product aimed quality are met.

1) Testability: due to the importance of having the least amount of bugs in our final product, our aim is to have a high testability; this refers to the fact that each part of the project will be tested as well as each of the components. Since the website will contain several layers and components as building blocks we will be able to test each of the parts as individual components.
2) Maintainability: As established in the testability specification, the project will be divided into several layers and components, due to this the maintainability will be high the following are the components divided into groups and their maintainability description.
   i. Server: the server system shall be highly maintainable because it will be hosted by an external service provider that will be in charge of maintaining all of the hardware that supports the website. The small amount of users is also important to consider this as a highly maintainable project
   ii. Static website: will require periodical maintenance with a low frequency to keep the static information up to date. This will not be time consuming and it may be done in a simple and fast way.
   iii. Dynamic website: this component of the system is considered to have a high maintainability given that it will be fully automated with the Epicollect Rest API, alongside with the Python modules. The most demanding component inside the dynamic website in terms of maintainability is the database management, since the data cleaning may be needed to change depending on the updates on the Epicollect5 database or API.
3) Scalability: the system shall be scalable, with the ability of extending to countries or areas that currently do not have service (outside the area that today contains information. The modular structure of the product will permit us to build more components around the website and simplify the scalability.

5.3 Technical Requirements

In this section the Technical requirements will be described. These requirements specification contains comprehensive knowledge about the purpose of the system, functionality and implementation methods

1) The program language must be Python
2) The web page language must be HTML, CSS, JS
3) The function interface should be Easy
4) The web page display language should be English
5) The system should use PostgreSQL Database to store the information

Section 6 - Effort Spent

For this project, the development team is composed of four persons. During the project, we use the Github to manage our project, the Google File to edit our RASD together and the zoom to discuss online if not allowed to meet in public.

Date	Task	Gao	Ding	Ernesa	Song
22/3/2021	Project Dataset	2h	2h	2h	2h
28/3/2021	Project Scope, Goals and Domain Analysis	3h	3h	3h	3h
11/4/2021	Use Cases	4h	4h	4h	4h
10/5/2021	Revision S1-S3	4h	4h	4h	4h
14/5/2021	Revision S4 S5	4h	4h	4h	4h

Section 7 - Revision History

Date	Modifications
11/4/2021	First Version
23/5/2021	Second Version
24/5/2021	Diagrams Revamped; Format Unified

Section 8 - Reference

[1] “World Geodetic System 1984 (WGS 84)”, National Geospatial-Intelligence Agency, Last modified: 12-Apr-21, https://earth-info.nga.mil/index.php?dir=wgs84&action=wgs84