# **Project Scope and Goals**

The **Urban Environmental Investigation** collected environmental information on location (including time and location) in the SAN SILVESTRE(Lima, Peru), and obtained actual results according to different climatic conditions. According to the location, we can collect different parameters related to the location, such as: average noise level, average light intensity, wind direction, wind speed, cloud cover, cloud type, photo of cloud conditions, visibility, traffic count, temperature, humidity and air pollution. There are a lot of data about the urban environment in different location. Where are the locations? What are the environment conditions? You cannot analysis the data effectively if you do not know their locations and environmental conditions. The **Urban Environment Investigation** is an effective tool of management of these data.

During this project which is done by a group of Geoinformatics Engineering Students of Politecnico di Milano, after acquisition of data of the Urban Environmental Investigation by the “San Silvestre Geography IA 2020” available on [Epicollect website](https://five.epicollect.net/project/san-silvestre-geography-ia-2020/data). These data with 451 entries need to be treated in order to understand and use them. So we are required to design a web application which follows these goals:

* Processing and exposing on the web the data using some original manipulation strategy, by leveraging both the geographic content (map-based view) as well as attributes (interactive exploratory graphs).
* Allowing users to extract custom views of the data, leave comments, and eventually discover how to contribute to the data collection.

# **Domain Analysis**

This web application is designed for real estate customers, urban planner and urban environmental researcher.

Our software will not interact with other software of hardware the whole manipulations will be done online by the users.

This web application can be used for surveying, monitoring and information exchanging. 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 following: shapely, pandas, geopandas for the representation of data; psycopg2 which is the python SQL version; flask for web implementation.

# **World and Machine Phenomena Scheme**



**World Phenomena:**

1. Condition of the Environment:

This refers to the conditions of environment in different times and locations, including average noise level, average light intensity, wind direction, wind speed, cloud cover, cloud type, photo of cloud conditions, visibility, traffic count, temperature, humidity and air pollution.

**Shared Phenomena:**

1. SignUp:

The process allows each new user to register their accounts by filling the fields.

2. Login:

The process allows user to go to their Home Page after they signed-up (If user is new) or the process allows user to go to their Home Page (If user already registered).

3. Logout:

The process allows user to logout their accounts after login.

4. Upload, Download, Delete, Comments:

The processes allows user to do after they signed up and Logged in to our Web App.

5. Location:

Exact position of the condition in a specific reference system

**Machine Phenomena:**

Data Base Queries:

To Retrieve the data from the database (User Data, Environmental Conditions Data)

Visualization of Data by Map View:

The data of the environment is shown on an OSM base Map with an Interactive View.

# **Use cases**

In the following section we are going to discuss different general scenarios of the use of the system

# **Requirements:**

**We can classify the requirements into three categories:**

1. Technical requirements

2. Functional requirements

3. Non-Functional requirements

**Technical requirements:**

The system should be implemented in:

1. The implementation language must be Python.

2. The web page language must be HTML, CSS, JAVASCRIPT.

3. The interface should be Easy.

4. The web page display language should be English.

5. The Database to store the information should be PostgreSQL.

**Functional requirements:**

1. The system should allow multiple users to access the services at same time.

2. The system should allow registered users to enter their home page.

3. The system should allow registered users to upload the data, view data, and be able to use other available services.

4. The system shall allow non-registered users to see the about of Web app.

**Non-Functional Requirements:**

The Web App should be available 24h/7.

The data should be updated in the real time when the user uploads data or deletes his data.

The web app should be mobile phone compatible.

# **Effort Spent:**

# **Reference**