## **Functional Requirements**

### **1. Sensor Functionality Analysis**

* **Objective: Ensure sensors are functioning correctly.**
* **Inputs: Concentrations of O3, SO2, NO2, PM10 from** measurements.csv**; sensor coordinates from** sensors.csv**.**
* **Outputs: List of malfunctioning sensors.**
* **Process: Compares sensor data against the five geographically closest sensors, accepting a maximum deviation of 10% per indicator.**
* **Tests:**
  + **Detect a sensor providing one incorrect concentration.**
  + **Detect a sensor providing entirely incorrect concentrations.**

### **2. Air Quality Statistics Production**

* **Objective: Generate air quality statistics for a specified time and location.**
* **Inputs: Time (moment or period), Location (coordinates), Sensor data (**measurements.csv**), Optional radius (default 10 km).**
* **Outputs: ATMO index value, air quality level for the area.**
* **Process: Calculates average concentrations to measure air quality based on the ATMO index.**
* **Tests:**
  + **Verify the lowest quality level is displayed when measurements are between quality levels.**
  + **Expand search radius automatically if no sensors are found in the initial area.**
  + **Notify the user if no measurements are available for the specified time.**

### **3. Sensor Similarity Scoring and Ranking**

* **Objective: Score and rank sensors based on similarity to a selected sensor during a specified time.**
* **Inputs: Sensor ID (user-selected), Time period (user-selected).**
* **Outputs: Selected sensor's score, top 10 similar sensors with scores and distances.**
* **Process: Displays sensors with the closest air quality readings, within a 10% similarity tolerance, ranking further sensors higher in case of equal scores.**
* **Tests:**
  + **Handle fewer than 10 sensors within the similarity score threshold.**
  + **Address identical scores and distance ranking.**

### **4. Air Cleaner Impact Observation**

* **Objective: Assess the impact of air cleaners on air quality.**
* **Inputs: Cleaner data (**cleaners.csv**), Sensor locations (**sensors.csv**).**
* **Outputs: The furthest sensor within the impact zone, improvement levels in percentages.**
* **Process: Compares data from the three closest sensors during and outside cleaner operation to calculate improvement.**
* **Tests:**
  + **Identify cleaners with no measurable impact on air quality.**
  + **Provide feedback when data is insufficient for assessment.**

### **5. Data Sharing Points Allocation**

* **Objective: Reward private individuals for sharing sensor data.**
* **Inputs: User information (**users.csv**), Data usage instances.**
* **Outputs: Points awarded, total points per user.**
* **Process: Tracks sensor data usage and allocates 10 points per instance to the contributing user.**
* **Tests:**
  + **Verify functionality with varied data usage scenarios.**
  + **Ensure no points are allocated for non-private sensors.**

### **6. Individual Reliability Classification**

* **Objective: Determine the reliability of data from private sensors.**
* **Inputs: User and sensor IDs (**users.csv**), Sensor data (**measurements.csv**).**
* **Outputs: Updated reliability status in** measurements.csv **and** users.csv**.**
* **Process: Compares user sensor data against neighboring sensors to detect anomalies beyond a 20% tolerance.**
* **Tests:**
  + **Validate with both reliable and unreliable sensor data.**
  + **Ensure user reliability status is accurately updated and communicated.**

### **Dataset Context**

* **Total Sensors: 100 sensors, mapped across France.**
* **Measurements: Daily readings of O3, SO2, NO2, PM10, taken simultaneously.**
* **User Types: Two private users, one honest, one dishonest.**
* **Air Cleaners: Two, with differing effectiveness.**
* **Task: Identify dishonest users, effective air cleaners, and impact zones through AirWatcher functionalities.**

**Functional Requirements :**

| **Functionalities** | **Inputs** | **Outputs** | **Description** | **Tests** |
| --- | --- | --- | --- | --- |
| 1. Analyze **data** to make sure sensors **function correctly** | **data :**   * concentrations of O3, SO2, NO2, PM10 of each sensor, stored in measurements.csv * coordinates of the sensors stored in sensors.csv | List of malfunctioning sensors | The system compares the coherence of the measures provided by each sensor with the ones provided by the five other closest sensors to pinpoint malfunctioning sensors. The tolerated margin of error will be fixed at 10% for each indicator. | * Pinpoint a sensor that gives one incorrect concentration * Pinpoint a sensor that gives fully incorrect concentrations |
| 1. Produce **statistics** about **air quality** at a certain **time** and **place** | **Time :** Given moment or specific period of time  **Place :** Coordinates of a point in space  **data :** concentrations of O3, SO2, NO2, PM10 of each sensor, stored in measurements.csv  **(Radius)** Optional : 10 Km by default ; radius of the circular for gathering data. | The value of the ATMO index for the area  The level of air quality in the specified area | The quality of air is measured according to the ATMO index after computing the average value of each concentration provided by all the sensors in the area | * State a case in which the concentrations are horseback between several levels of quality of air and make sure the lowest level is displayed * State a case in which no sensor was found in the designated area and inform the user that the area will be automatically expanded for computation * State a case in which no measure was found in the given timeslot and warn the user that no statistic could be given in this case |
| 1. **Score** and **rank** sensors in terms of similarity to a **selected sensor** at a specified **period of time** | **Selected sensor :** Id of a sensor entered by the user  **Period of time :** specific period of timechosen by the user | Score of the selected sensor  Top 10 similar sensors with their score and their distance to the selected sensor | The system computes the quality of air of all the sensors, and displays the 10 sensors with the closest ATMO to the selected sensor, with a tolerance of 10%. If two or more sensors have the same ‘similarity score’, the furthest is ranked first | * State a case in which less than 10 sensors share the same similarity score * State a case in which less than 10 sensors are in the margin of tolerance of the score of the selected sensor * State a case in which more than 10 sensors are in the margin of tolerance of the score of the selected sensor * State a case in which two or more sensors compete for a same rank and the furthest is ranked first * State a case in which no ranking could be established due to absence of similar scores in within the margin of tolerance |
| **4) Observe the impact of cleaners on air quality** | **One cleaner with its period of use and localisation stored in cleaners.csv Localisations of each sensors stored in sensors.csv** | **The furthest sensor of the cleaned zone around the specified cleaner and the level of improvement given by a relative percentage for each concentration** | **Select the three closest sensors, and compare their data during and outside the period of use of the cleaner to calculate the level of improvement. As long as the ATMO index is improved by at least one level, search for the next closest sensor.** | **• Cleaners with no improvement on air quality**  **• If the three closest sensors didn’t register during the time of use of the cleaner, inform the user that no data can be provided on the level of improvement**  **• If the the ATMO index didn’t change, return a message saying it**   * **if none of the sensors register during the period of use are before, no data are provided** |
| 1. Grant **points** to **private individuals** for **data sharing** | **Private individuals :** users from user.csv  **Data sharing :** number of times a user sensor’s data was used in the system’s queries | **Points :** 10 points granted to the user per data used  Return the total of points gathered by each private individual | The system enlists all the sensors made available for data sharing by a certain UserId, increments the number of times the data provided by these sensors was used each time they are, and grants 10 points per data used to the user | * Test the functionality when no private users’ data was used * Test the functionality when a user’s data was used several times * Test the functionality when a user was banned * A data is used but is not from a private individual : no points are granted for the user |
| 1. Classify individuals as unreliable | **Private individuals :** UserId and SensorId from user.csv  Data provided by these SensorId from measurements.csv | Update all the values provided by an unreliable user as False in measurements.csv  Update the status of a user as BANNED in user.csv | The system compares the measures provided by a user’s sensor to the average of the measures provided by the three closest sensors at the same date. If more than 10 measures don’t meet a 20% tolerance on a certain concentration and if the closest sensor is under 5 km away, the user is BANNED. | * Test it with a user considered as reliable and a user considered as unreliable (all sensors wrong) * Add a “reliable” sensor to the user considered as unreliable : this latter will still be banned * An individual with one sensor that have 9 measures in O3 unreliable is still considered honest * An individual with only one sensor but no sensor within a 5 km radius is still reliable no matter the measures, but a warning is displayed saying that the verification can’t be made * If there are no correlations in the date, a warning is displayed saying that the verification can’t be made |

Some information about the data set is given below.

* There are a total of 100 sensors located on a grid on the map of France.
* There are 4 measurements (one of each of the 4 types: O3, SO2, NO2, PM10) per sensor per day for a period of 1 year.
* All measurements are taken at the same time every day.
* The measurements of the sensors are coherent with the sensors in the neighborhood. This means that it is probable to find similar measurement values of the same type among sensors that are in immediate vicinity.
* There are two private individual users.
* One of the private individual users is honest. Whereas, the other one is dishonest. The

dishonest user’s sensor generates false data.

* There are two air cleaners.
* One of the air cleaners is ineffective and thus has no impact on the air quality. Whereas, the other one is very effective and has a strong impact on measurements generated by surrounding sensors during the period of its operation.
* You are required to discover information such as the identity of the dishonest user, the identity of the effective air cleaner, the radius of the cleaner’s impact, etc. through the functionalities provided by your implementation of AirWatcher.