III. Test Plan

To execute the different tests, we created new csv files dedicated to the tests. We choose to take only 2 different sensors 0 and 1 and took one measurement for each of them. We also created a file for the passwords and for the government’s logins.

III.1. Unitary Tests

### III.1.1. UserController functions

### LoadCSV functions

These tests will verify if the files passed as parameters in the functions loadCsv are correctly loaded. Not only the files’ names have to be correct but also the data loaded. We can check if the all data are correctly loaded by comparing the size of the users list and the sum of each files’ size.

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.loadCSV(string userFile, string providerFile, string governmentFile, string passwordFile) | Paths to the following CSV files : users.csv providers.csv government.csv passwords.csv | true |
| Additional Test: Verification of the obtained user list after loading the different files | Size of userController.users should be equal to the sum of the entries in users.csv, providers.csv and government.csv | 6 |
| Additional Test: entry integrity check | Check that list entries aren’t empty. Check that first and last entries contains valid users, and all have password differing from null or string.empty() | N/A |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.loadCSV(string userFile , string providerFile, string governmentFile, string passwordFile) | Non-exisiting files and empty / not valid files | false |
| Additional test: user list should remain unchanged | Before the first test, empty the users list. Then, check it is still empty. | Lists should be equal |

### Authentication

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.authenticate(string login, string pass) | Provider0  provider0 | User object with identifier provider0 and cleaner object identifier Cleaner0 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.authenticate(string login, string pass) | Provider5  provider0 | Null |

### GetIndividualUsers

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.getIndividualUsers() | void | Vector of IndividualUsers objects with identifiers : User0 and User1 |

### GetProviders

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.getProviders() | void | Vector of Providers objects with identifiers : Provider0 and Provider1 |

### GetPrivilege

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.getPrivilege(string identifier) | User0 | INDIVIDUAL |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.getPrivilege(string identifier) | Provider0 | PROVIDER |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.getPrivilege(string identifier) | Government0 | GOVERNMENT |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| userController.getPrivilege(string identifier) | User8 | null |

### III.1.2. SensorController functions

### GetSensors

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController.getSensors() | void | Vector of Sensor objects with identifiers : Sensor0 and Sensor1 |

### GetSensor

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController.getSensors(string identifier) | Sensor0 | Sensor object with identifiers : Sensor0, latitude: 44 and longitude : -1 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController.getSensors(string identifier) | Sensor101 | null |

### Malfunctionning analysis

This method uses several other functions to produce the wanted result:

1. removeAllMeasurementsFromSensor(Measurement[] measurements, Sensor sensor)

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. removeAllMeasurementsFromSensor(Measurement[] measurements, Sensor sensor) | getAllMeasurements() Sensor0 | Will give all the measurement except sensor0’s ones : Her we only have sensor1’s measurements |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. removeAllMeasurementsFromSensor(Measurement[] measurements, Sensor sensor) | getAllMeasurements() sensor101 | null |

1. FR5\_malfunctioningAnalysis(in sensorToCheck:Sensor): double

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR5\_malfunctioningAnalysis(Sensor sensorToCheck) | sensor0 | 0.08 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR5\_malfunctioningAnalysis(Sensor sensorToCheck) | sensor101 | null |

### MeanAir Quality

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. meanAirQuality(double latitude, double longitude, double radius, time\_t start, time\_t stop) | 45  -2  5  01/01/2019 12:00:00 01/01/2019 12:00:00 | mediocre |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. meanAirQuality(double latitude, double longitude, double radius, time\_t start, time\_t stop) | 45  -2  5  01/01/2019 12:00:00  15/15/2025 12:00:00 | null |

### Compare Sensors

This method uses several other functions to produce the wanted result:

1. FR7\_averageValue(Sensor sensor, Attibute targetAttribute, time\_t t1, time\_t t2)

Compute the average value of all its measurements for the given attribute

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR7\_averageValue(Sensor sensor, Attibute targetAttribute, time\_t t1, time\_t t2) | sensor0  03  01/01/2019 12:00:00  01/01/2019 12:00:00 | 50.25 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR7\_averageValue(Sensor sensor, Attibute targetAttribute, time\_t t1, time\_t t2) | sensor0  o2  01/01/2019 12:00:00  01/01/2019 12:00:00 | null |

1. FR7\_sensorComparison (Sensor sensorToCompare, timestamp t1, timestamp t2)

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR7\_sensorComparison (Sensor sensorToCompare, timestamp t1, timestamp t2) | Sensor0  01/01/2019 12:00:00  01/01/2019 12:00:00 | A map with Sensor object and a double value (similarity 0 to 1) :  Sensor1 and 0.846 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR7\_sensorComparison (Sensor sensorToCompare, timestamp t1, timestamp t2) | Sensor101  01/01/2019 12:00:00  01/01/2019 12:00:00 | null |

### Air Quality

This method uses several other functions to produce the wanted result:

1. isGivenTimeInsideTimePeriod(time\_t start, time\_t : stop, time\_t: time): bool

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. isGivenTimeInsideTimePeriod(time\_t start, time\_t : stop,time\_t: time): bool | 01/01/2019 12:00:00  01/01/2019 12:00:00  01/01/2019 12:00:00 | true |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. isGivenTimeInsideTimePeriod(time\_t start, time\_t : stop, time\_t: time): bool | 01/01/2019 12:00:00  01/01/2020 12:00:00  01/01/2025 12:00:00 | null |

1. distanceBetweenPositions (double latitudeA, double longitude, double latitudeB, double longitudeB)

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. distanceBetweenPositions (double latitudeA, double longitude, double latitudeB, double longitudeB) | 44  -1  45  -2 | 1.41 |

1. FR8\_qualityAttributes (double latitude, double longitude, time\_t time)

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR8\_qualityAttributes (double latitude, double longitude, time\_t time) | 45  -2  01/01/2019 12:00:00 | Returns a Map<Attribute,double> with the Attribute object’s identifiers o3, No2, So2 and PM10 and their corresponding measurement:  O3 : 55.56  NO2 : 69.28  S02 : 38.56  PM10 : 47.39 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR8\_qualityAttributes (double latitude, double longitude, time\_t time) | 45  -2  15/15/2025 12:00:00 | null |

1. FR8\_quality (double latitude, double longitude, time\_t time)

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR8\_quality (double latitude, double longitude, time\_t time) | 45  -2  01/01/2019 12:00:00 | String equal to “Mediocre” |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController. FR8\_quality (double latitude, double longitude, time\_t time) | 45  -2  15/15/2025 12:00:00 | null |

### LoadCSV

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController.loadCSV(string sensorFile, string measurementFile, string attributeFile) | Path to the following CSV files: sensors\_test.csv, measurements\_test.csv, attributes.csv | true |
| Additional Test: Verification of the obtained sensors list after loading the different files | sensorController.sensors.size() | 2 |
| Additional Test: Verification of the obtained measurements list after loading the different files | Check the total number of measurements (sum of the number of measurements for each sensor) | 8 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| sensorController.loadCSV string sensorFile, string measurementFile, string attributeFile) | Non-existing or invalid files | false |
| Additional Test: check sensors list integrity | Empty list before test. Check that sensorController.sensors list is empty | N/A |

### III.1.3. CleanerController functions

### LoadCSV

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| cleanerController.loadCSV(string file) | Path to the cleaners.csv file | true |
| Additional Test: Verification of the obtained cleaners list after loading the different files | Check the total number of cleaners (sum of the number of cleaners for each provider) | 2 |

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| cleanerController.loadCSV(string file) | Non-existing of invalid file | false |
| Additional Test: check cleaners list integrity | Empty lists before previous test. Check that all the providers have no cleaners registered. | N/A |

### Compute Cleaner Statistics

|  |  |  |
| --- | --- | --- |
| Tested Function | Test description / input | Expected result |
| cleanerController.computeStatistics(Cleaner cleaner) | Cleaner with identifier Cleaner0 | null |

III.2. Functional Tests

To test the different functionalities of our application we need to define what happens at each step of the scenario.  

**1 - Login fail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Input** | **Results** |
| 1 | The user has to login | Id : User101  Password: banana | You don’t have access to the application. Please enter a good id or password. |

**2 - Access sensors list**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use\_sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | get sensors | Name : Sensor0 Longitude: 44 Latitude: -1  Name : Sensor1 Longitude: 44 Latitude: -0.3  (The options are once again displayed) |

**3 - Retrieve sensor’s measurements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | use sensor <0> | Name: Sensor0 Longitude: 44 Latitude: -1  Sensor Menu :  1- measurements  2- measurements<date>  3- evaluate  4- disable  5- enable  6- compare  7- menu  Which functionality do you want to use? |
| 3 | He then has to choose an option in the new menu | measurements | Date : 01/01/2019 12:00  O3 : 50.25  NO2: 74.5  SO2: 41.5  PM10: 44.75  (The options are once again displayed) |

**4 - Retrieve sensor’s measurements at a special date**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | use sensor <0> | Name: Sensor0 Longitude: 44 Latitude: -1  Sensor Menu :  1- measurements  2- measurements<date>  3- evaluate  4- disable  5- enable  6- compare  7- menu  Which functionality do you want to use? |
| 3 | He then has to choose an option in the new menu | Measurements<01/01/2019 12:00> | Date : 01/01/2019 12:00  O3 : 50.25  NO2: 74.5  SO2: 41.5  PM10: 44.75  (The options are once again displayed) |

**5 - Run malfunctioning sensor detection analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | use sensor <0> | Name: Sensor0 Longitude: 44 Latitude: -1  Sensor Menu :  1- measurements  2- measurements<date>  3- evaluate  4- disable  5- enable  6- compare  7- menu  Which functionality do you want to use? |
| 3 | He then has to choose an option in the new menu | evaluate | 0.08  *(The options are once again displayed)* |

**6 - Mark specific sensor as unreliable *(Only for the government agency)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | A government agent has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | use sensor <0> | Name: Sensor0 Longitude: 44 Latitude: -1  Sensor Menu :  1- measurements  2- measurements<date>  3- evaluate  4- disable  5- enable  6- compare  7- menu  Which functionality do you want to use? |
| 3 | He then has to choose an option in the new menu | disable | Sensor0 has been disabled.  *(The options are once again displayed)* |

Similar test but for the functionality: Mark specific sensor as reliable. The command is now enabled, and the result should be displayed: Sensor0 has been enabled

**7 - Compare the similarity between the other sensors**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | use sensor <0> | Name: Sensor0 Longitude: 44 Latitude: -1  Sensor Menu :  1- measurements  2- measurements<date>  3- evaluate  4- disable  5- enable  6- compare  7- menu  Which functionality do you want to use? |
| 3 | He then has to choose an option in the new menu | compare | Sensor1 : 85%  *(The options are once again displayed)* |

**8 - Consult a sensor: failed attempt**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | use sensor <102> | No sensor found  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |

**9 - Retrieve mean quality of air on a specified area**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | airQ <45> <-2> <01/01/2019 12:00><01/01/2019 12:00>” | Mean air quality : mediocre  (The options are once again displayed) |

Similar test but with an error in the dates, the latitude or the longitude.   
Result: “Error in the input, please try again”

**10 - Retrieve quality of air at a given position and time**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The user has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | Enter the command “airQ <45> <-2> <01/01/2019 12:00>” | Air Quality : mediocre  (The options are once again displayed) |

Similar test but with an error in the date, the latitude or the longitude.   
Result : “Error in the input, please try again”.

**11 – Access the list of owned air cleaners (Only for providers)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The Provider has to login | Id : Provider0  Password: provider0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | consult cleaner<0> | Name : Cleaner0  Latitude: 45.3333  Longitude: 1.33333  Installation date : 01/02/2019 12:00:00  (The options are once again displayed) |

**12 - Access individual users list *(Only for the government agency)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The Government agent has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | get users | Name: User0  Sensor: Sensor70  Name: User1  Sensor: Sensor36  (The options are once again displayed) |

**13 - Access all air cleaners providers list (Only for the government agency)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Actions** | **Results** |
| 1 | The Government agent has to login | Id : Government0  Password: government0 | You are connected.  Menu  1- get\_sensors  2- use sensor <num>  3- airQ <latitude> <longitude> <start\_date><end\_date>  4- airQ <latitude> <longitude> <date>  5- consult cleaner <num>  6- get users  7- get providers  8- get\_cleaners  Which functionality do you want to use? |
| 2 | He then has to choose a functionality | get providers | Name: Provider0  Sensor: Cleaner0  Name: Provider1  Sensor: Cleaner1  (The options are once again displayed) |