

B3124 - DALAOUI Riad, JOURDA Florian

B3129 - CAPGRAS Evan, DRAVET Eléonore, STEPHAN Justine

GÉNIE LOGICIEL ET MODÉLISATION UML

PROJET UML 3IF
-
AIRWATCHER

SOMMAIRE

Partie 1

Introduction

Planning

Equipe :

Diagramme de Gantt

Spécifications

Diagramme de cas d'utilisation

Prérequis fonctionnels et non-fonctionnels du système

Analyse des risques de sécurité

Tests de validation

Service : computeAQIMoment

Test : Valider la Précision du Calcul de la Moyenne de l'IQA

Service : computeAQIPeriod

Objectif

Vérifier que le service calcule correctement la moyenne de l'IQA dans un cercle pour une période spécifiée.

Service : compareSensor

Test : Classer les capteurs par similarité sur une période donnée

Objectif

Service : analyseSensor

Test : vérifier si un capteur est cohérent

Objectif

Service : measureImprovement

Test : Valider la Mesure de l'Amélioration de la Qualité de l'Air

Objectif

Service : measureAreaEffect

Test : Valider le Calcul du Rayon de la Zone d'Amélioration

Objectif

Service : computeAQIAtPoint

Test : Valider la Production d'une Valeur de Qualité de l'Air

Objectif

Manuel d'utilisation - interface console

Document de design

Architecture

Diagramme de classes

Diagrammes de séquence des trois scénarios majeurs

Descriptions et pseudo-codes de trois algorithmes

Unit tests

Service : allSensors

Cas de Test : Retourner l'ensemble des Capteurs du fichier Sensors.csv

Service : sensorsInCircle

Cas de Test : Retourner l'ensemble des Capteurs situé dans le rayon d'une coordonnée

Service : rankedSensors

Cas de Test : Établir un classement des capteurs par proximité géographique par rapport à un point donné

Service : markMalicious

Cas de Test : marquer un utilisateur comme malicious

Service : addPoint

Cas de Test : ajouter un point à un utilisateur si son capteur est utilisé

Service : allAttributes

Cas de Test : Retourner l'ensemble des attributs du fichier attributes.csv

Service : measurementsInCircleInPeriod

Cas de Test : Retourner les mesures dans la zone donnée, sur une période donnée

Service : measurementsInCircleAtMoment

Cas de Test : Retourner les mesures dans la zone donnée, à un instant donné

Service : measurementOfSensor

Cas de Test : Retourner l'ensemble des mesures produites par un capteur donné

Annexe 1 : jeux de tests

Jeux de tests

Jeu de test 1

Jeu de test 2

Jeu de test 3

Jeu de test 4

Introduction

Le projet AirWatcher vise à développer une application logicielle pour une agence gouvernementale chargée de la protection de l'environnement. L'application analysera, surveillera et rendra compte de la qualité de l'air sur un vaste territoire en utilisant des données collectées auprès de divers capteurs. Ce projet englobe les principes de génie logiciel pour garantir la création d'une solution logicielle de haute qualité et fiable.

Planning

Equipe :

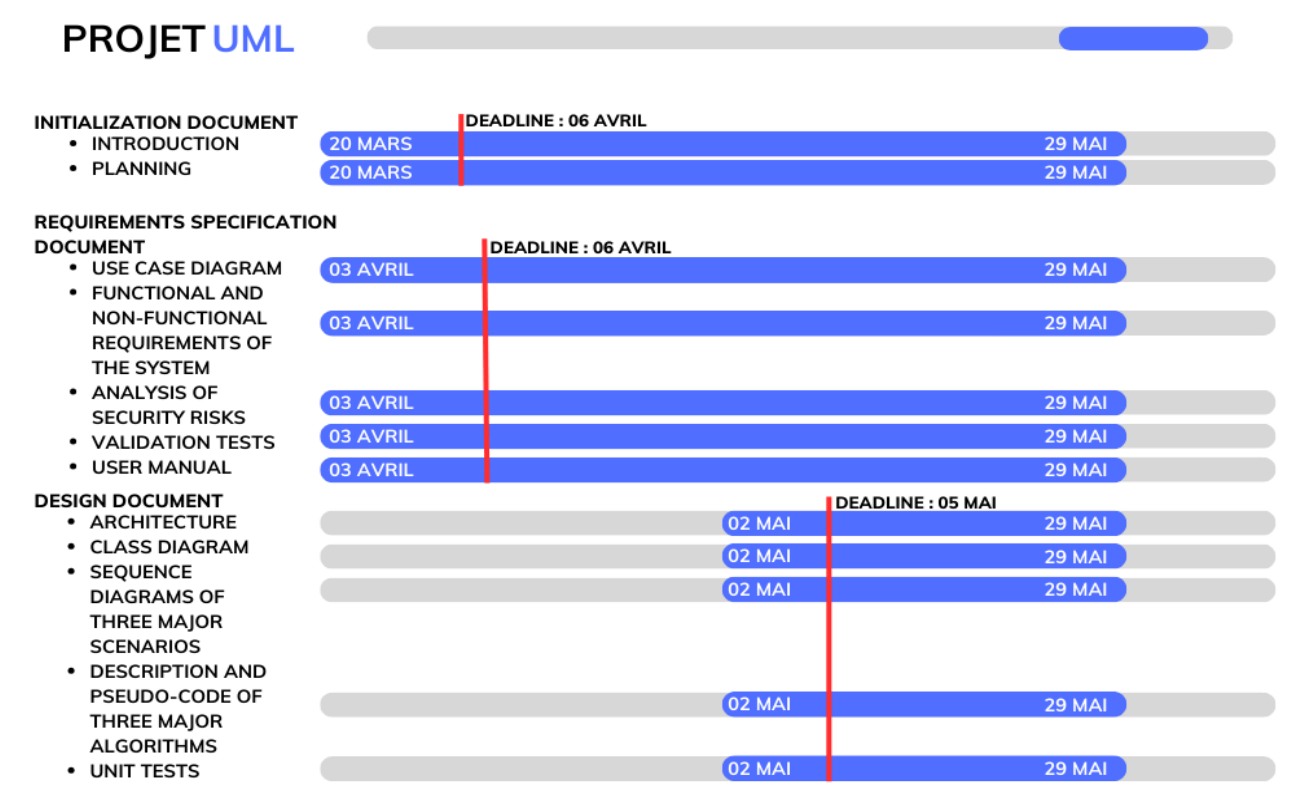
B 3129

Eléonore DRAVET
Evan CAPGRAS
Justine STEPHAN

B 3124

Florian JOURDAN
Riad DALAOUI

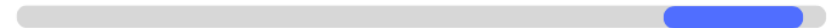
Diagramme de Gantt



PROJET UML

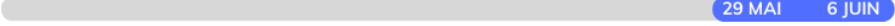
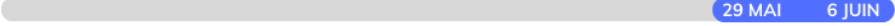
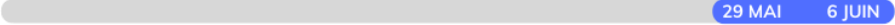
SOFTWARE APPLICATION DEVELOPMENT

- SOURCE CODE + GUIDE TO BUILD THE EXECUTABLE
- TESTS



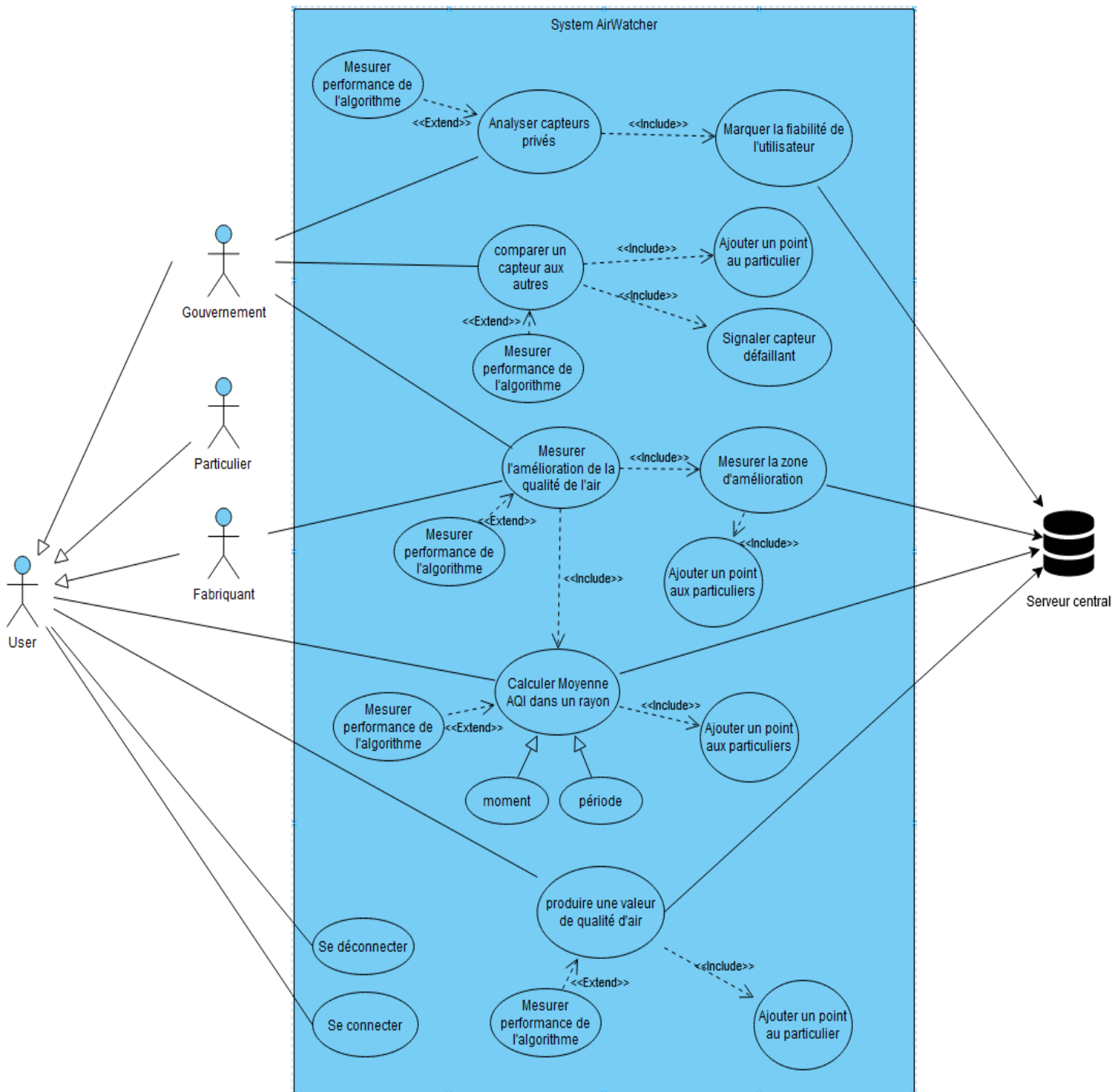
PRESENTATION AND DEMO

- PRESENTATION OF THE SPECIFICATION AND DESIGN
- LIVE DEMO OF THE IMPLEMENTED FUNCTIONALITIES
- PERFORMANCE EVALUATION OF THE EXECUTED ALGORITHMS
- SOURCE CODE REVIEW



Spécifications

Diagramme de cas d'utilisation



Prérequis fonctionnels et non-fonctionnels du système

1) Fonctionnels

Service	analyzeSensor
Description	Calcule la valeur estimée d'AQI du capteur avec les valeurs des 5 capteurs les plus proches (moyenne pondérée par la distance). Si la différence entre la valeur estimée et la vraie valeur >10% de la vraie valeur, le capteur est considéré comme corrompu. L'utilisateur est noté malveillant. Ajoute un point aux particuliers dont un capteur a été utilisé dans la requête (plusieurs points ajoutés si plusieurs capteurs). Mesurer la performance de l'algorithme.
Inputs	String sensorID
Outputs	Affiche si le capteur est considéré non-valide ou valide.
Source	measurements.csv, attributes.csv, users.csv
Destination	console gouvernement
Précondition	Le capteur appartient à un utilisateur privé.
Postcondition	La console a affiché le résultat. Si le capteur est corrompu, son propriétaire est noté dans malicious.csv. Le temps d'exécution de l'algorithme est affiché .

Service	compareSensor
Description	Établit un classement avec score de tous les capteurs par rapport à leur similarité au capteur indiqué. Ajoute un point aux particuliers dont un capteur a été utilisé dans la requête (plusieurs points ajoutés si plusieurs capteurs). Mesurer la performance de l'algorithme.
Inputs	String sensorID, date debut, date fin
Outputs	Affiche le classement avec les scores.
Source	measurements.csv, attributes.csv
Destination	console gouvernement
Précondition	
Postcondition	La console a affiché le classement avec les scores. Le temps d'exécution de l'algorithme est affiché.

Service	computeAQIMoment
Description	<p>Retourne la moyenne des différentes valeurs de la qualité de l'air des capteurs dans le cercle, en utilisant les données récoltées le jour-même du moment choisi (de 00:00 à 23:59). Le pourtour du cercle est inclus. Précision 10^{-2}. Si le rayon est nul, si un capteur se situe à l'endroit précis, sa valeur est renvoyée, sinon une erreur est renvoyée. Si le rayon est négatif ou qu'aucun capteur ne se situe dans le cercle, une erreur est renvoyée.</p> <p>Ajoute un point aux particuliers dont un capteur a été utilisé dans la requête (plusieurs points ajoutés si plusieurs capteurs). Mesure la performance de l'algorithme.</p>
Inputs	float latitude, float longitude, int radius, Date moment
Outputs	float Moyenne de la qualité de l'air
Source	measurements.csv, attributes.csv
Destination	consoles gouvernement, particulier et fabricant de purificateurs
Précondition	
Postcondition	La moyenne des données de qualité de l'air des capteurs est renvoyée et affichée sur la console. Le temps d'exécution de l'algorithme est affiché.

Service	computeAQIPeriode
Description	<p>Retourne la moyenne des différentes valeurs de la qualité de l'air des capteurs dans le cercle, entre le début et la fin incluses.</p> <p>Ajoute un point aux particuliers dont un capteur a été utilisé dans la requête (plusieurs points ajoutés si plusieurs capteurs). Mesurer la performance de l'algorithme.</p>
Inputs	float latitude, float longitude, int radius, Date debut, Date fin
Outputs	float Moyenne de la qualité de l'air
Source	measurements.csv, attributes.csv
Destination	consoles gouvernement, particulier et fabricant de purificateurs
Précondition	
Postcondition	<p>La moyenne des données de qualité de l'air des capteurs est renvoyée et affichée sur la console.</p> <p>Le temps d'exécution de l'algorithme est affiché .</p>

Service	measureImprovement
Description	Retourne la différence d'AQI du capteur le plus proche entre la mesure juste avant la mise en marche et la mesure juste avant l'arrêt.
Inputs	CleanerID, date debut, date fin
Outputs	float diff
Source	measurements.csv, attributes.csv, cleaner.csv
Destination	console fabricant ou console gouvernement
Précondition	
Postcondition	

Service	measureAreaEffect
Description	Retourne le rayon de la zone où la qualité de l'air est améliorée par le cleaner. Parcourt les capteurs du plus proche au plus loin du Cleaner. Dès qu'un capteur a <10% de différence entre la valeur avec Cleaner la plus récente et la valeur sans Cleaner la plus récente, l'algorithme s'arrête. La distance du dernier capteur avec >10% de différence est renvoyée, précision 10^{-2} .
Inputs	String cleanerID
Outputs	float rayon
Source	measurements.csv, attributes.csv, cleaner.csv
Destination	console fabricant ou console gouvernement
Précondition	
Postcondition	

Service	computeAQIAtPoint
Description	Si un capteur est présent, retourne la valeur du capteur, sinon retourne une estimation. Cette estimation est faite en prenant la moyenne pondérée des cinq capteurs les plus proches, précision 10^{-2} .
Inputs	float latitude, float longitude, Date moment
Outputs	float valeur
Source	measurements.csv, attributes.csv
Destination	console gouvernement ou particulier ou fabricant de purificateur

Précondition	
Postcondition	

2) Non-fonctionnels

Requirement	Objectif	Métrique/mesure
Performance	Les algorithmes des services doivent être mesurables en temps, et les temps de calcul ne doivent pas dépasser 1000ms.	millisecondes
Opérationnel	Les utilisateurs doivent s'authentifier avec leur identifiant et leur mot de passe. Les services auxquels ils ont accès diffèrent selon leur profil.	
Robustesse	Le système ne doit pas planter dans plus de 5% des requêtes.	Pourcentage d'événements provoquant une erreur
Ergonomie	L'interface doit être facile à prendre en main, l'utilisateur doit être en mesure d'accéder à n'importe quelle fonctionnalité en moins de 4 lignes de commande.	nombre de lignes de commande
Fiabilité	Le système doit être accessible à l'utilisateur à tout instant.	
Développement	développement en C++, utilisation de l'outil Git, méthode de développement en cascade	
Stockage	L'ensemble des données doit être accessible à plusieurs utilisateurs qui se connectent en même temps.	
Sécurité-innocuité	L'ensemble des données est préservé en cas de défaillance système.	Pourcentage de données perdues
Sécurité-confidentialité	Les données ne sont pas accessibles sans compte et sont protégées contre les attaques potentielles. Le code du programme lui-même est également inaccessible sans compte et protégé.	

Analyse des risques de sécurité

Attaquant	<p>Particulier qui veut nuire au système</p> <p>Membre interne voulant nuire au système</p> <p>Concurrents</p> <p>Groupe ou individu malveillant</p>
Atout	<p>L'ensemble des données gérées par le système : donnés sur les utilisateurs, les capteurs et les cleaners</p> <p>ressources serveur</p> <p>fonctionnalités de l'application</p>
Vulnérabilité	<ul style="list-style-type: none">- Des mots de passe faibles sont permis- Les données saisies ne sont pas vérifiées / validées- Transmission et stockage de donnés sans chiffrement- pas de mesures anti-DoS- Utilisation des privilèges accordés aux membres internes
Attaque	<ul style="list-style-type: none">- l'attaquant devine le mot de passe- l'attaquant saisit des données inattendues- l'attaquant intercepte la communication- l'attaquant utilise du code malveillant : Virus, Worm, Trojan- Attaque par déni de service (DoS)
Probabilité d'attaque	<ul style="list-style-type: none">- moyenne- haute- faible- faible- faible

Risque	1) l'attaquant falsifie les données 2) l'attaquant obtient des données privées 3) l'attaquant met hors service le système 4) l'attaquant rend les fonctionnalités erronées
Niveau d'impact	1) moyen 2) moyen 3) élevé 4) moyen
Contre-mesure	<ul style="list-style-type: none"> - Les mots de passe sont vérifiés et les mots de passe faibles ne sont pas acceptés - Vérification des requêtes SQL et des liens http - Transmission et stockage de données avec chiffrement - protection contre les attaques DoS avec limitation du trafic - vérification régulière de l'intégrité des données - surveillance de l'activité des utilisateurs internes.

Tests de validation

Service : computeAQIMoment

Test : Valider la Précision du Calcul de la Moyenne de l'IQA

Objectif	S'assurer que le service calcule avec précision la moyenne de l'IQA dans un cercle spécifié.
Résultats attendus	La moyenne correcte de l'IQA (précision 10^{-2}) calculée à partir de données de capteurs dans le cercle.
Étapes du Test	<ul style="list-style-type: none"> ○ Appeler le service ○ Comparer le résultat du service avec la moyenne de l'IQA calculée manuellement des capteurs à l'intérieur du cercle.

Critère de réussite	Le résultat du service correspond à la moyenne de l'IQA calculée manuellement avec une tolérance de $\pm 0,01$.
Résultat attendu avec jeu de test 1	Entrée : 44 0; 1; 1; 2019-01-01 Sortie : résultat; temps d'exécution en ms avec $7,32 \leq \text{résultat} \leq 7,34$ <i>Calcul : les capteurs 0,1,2,3 sont pris en compte</i> <i>IQA capteur 0 : 7 IQA capteur 1 : 8 IQA capteur 2 : 7</i>
Résultat attendu avec jeu de test 1	Entrée : <i>latitude longitude; rayon; date</i> 44 0; -1; 1; 2019-01-01 Sortie : erreur, le rayon est négatif
Résultat attendu avec jeu de test 1	Entrée : <i>latitude longitude; rayon; date</i> 20 0; 1; 1; 2019-01-01 Sortie : erreur, il n'y a aucun capteur dans le cercle sélectionné. Agrandissez ou déplacez votre zone.

Service : computeAQIPeriod

Test : Valider le Calcul de la Moyenne de l'IQA Limité dans le Temps

Objectif	Vérifier que le service calcule correctement la moyenne de l'IQA dans un cercle pour une période spécifiée.
Résultats Attendus	Moyenne correcte de l'IQA (double) pour la période spécifiée.
Étapes du Test	<ul style="list-style-type: none"> ○ Invoquer le service avec les entrées données. ○ Vérifier le résultat contre une moyenne calculée manuellement des points de données pertinents.
Critère de Réussite	La moyenne calculée par le service correspond à la valeur attendue avec une tolérance de $\pm 0,01$.
Résultat attendu avec jeu de test 2	Entrée : <i>latitude; longitude; rayon; dateDebut; dateFin</i>
Résultat attendu avec jeu de test 2	Entrée : 44; 0; 2; 2019-01-01; 2019-01-02 Sortie : résultat, avec $7,42 \leq \text{résultat} \leq 7,44$

	<i>Calcul : le capteur 4 n'est pas pris en compte car sa mesure est hors de l'intervalle - Capteur 0 : 7x2, Capteur 1 : 8x2, capteur 2 : 7x2, Capteur 3 : 8</i>
Résultat attendu avec Dataset fourni	Entrée : 44;0; 1; 2019-01-01; 2019-01-02 Sortie : résultat, avec $7,85 \leq \text{résultat} \leq 7,87$ <i>Calcul : Capteur 0 : 7, Capteur 1: 8, Capteur 2: 7, Capteur 11: 8, Capteur 12: 8, Capteur 21: 9, Capteur 22 : 8</i>
Résultat attendu avec Dataset fourni	Entrée : 44;0; 2; 2018-01-01, 2018-01-02 Sortie : erreur, il n'y a pas de mesure dans l'intervalle de temps donné
Résultat attendu avec Dataset fourni	Entrée : 44;0; -1; 2019-01-01; 2019-01-12 Sortie : erreur, le rayon est négatif
Résultat attendu avec Dataset fourni	Entrée : 0;0; -1; 2019-01-01; 2019-01-02 Sortie : erreur, pas de capteur dans la zone indiquée

Service : compareSensor

Test : Classer les capteurs par similarité sur une période donnée

Objectif	Classer les capteurs par similarité au capteur sélectionné.
Résultats Attendus	Tableau 2D avec pour chaque capteurID, son score de similarité.
Étapes du Test	<ul style="list-style-type: none"> ○ Exécuter le service pour comparer le capteur sélectionné aux autres. ○ Comparer le tableau obtenu au tableau attendu.
Critère de Réussite	Le tableau est complet. On tolère une différence de 10^{-2} de score. L'ordre peut différer tant que l'ordre croissant est respecté.
Résultat attendu avec le jeu de test 2	Entrée : Sensor1;2019-01-01; 2019-01-02;0 Sortie : tableau Sensor0 1 Sensor3 1 Sensor2 2 <i>Calcul : Le capteur 4 n'est pas comparé car n'a aucune mesure dans le même laps de temps.</i> $0+1^2$; 1^2+1^2 ; 1^2
Résultat attendu avec le jeu de test 2	Entrée : Sensor200;2019-01-01; 2019-01-02;0 Sortie : null. Affichage "Ce capteur n'existe pas".

Service : analyseSensor

Test : vérifier si un capteur est cohérent

Objectif	Évaluer le processus de validation pour les capteurs privés et les capteurs associés à l'utilisateur.
Résultats Attendus	Statut de validité du capteur et de tout autre capteur appartenant au même utilisateur.
Étapes du Test	<ul style="list-style-type: none">○ Invoquer le service pour analyser le capteur privé.○ Vérifier le statut de validité du capteur analysé et des autres capteurs appartenant à l'utilisateur.
Critère de Réussite	Le service marque renvoie le booléen correspondant. malicious.csv contient bien les capteurs du propriétaire si le capteur a été considéré corrompu..
Résultat attendu avec le jeu de test 2	Entrée : Sensor4 Sortie : true
Résultat attendu avec jeu de test 1	Entrée : Sensor1 Sortie : false

Service : measureImprovement

Test : Valider la Mesure de l'Amélioration de la Qualité de l'Air

Objectif	Vérifier que le service calcule précisément la différence de qualité de l'air avant et après l'utilisation d'un purificateur d'air.
Résultats Attendus	Une valeur double indiquant la différence de qualité de l'air.
Étapes du Test	<ul style="list-style-type: none">○ Sélectionner un purificateur et identifier deux moments pour comparer les mesures de qualité de l'air.○ Invoquer le service avec ces entrées.○ Évaluer le résultat par rapport aux métriques d'amélioration attendues basées sur des données fictives.
Critère de Réussite	Le résultat du service reflète correctement la différence de qualité de l'air, dans une tolérance de $\pm 0,01$.
Résultat attendu avec jeu de test 3	Entrée : Cleaner0 Sortie : 0

Résultat attendu avec jeu de test 3	Entrée : Cleaner1 Sortie : 6
--	---------------------------------

Service : measureAreaEffect

Test : Valider le Calcul du Rayon de la Zone d'Amélioration

Objectif	Confirmer que le service peut déterminer le rayon dans lequel la qualité de l'air s'est améliorée grâce à un purificateur d'air.
Entrées	Identifiant du purificateur d'air, deux horodatages pour la comparaison.
Résultats Attendus	Une valeur double représentant le rayon d'amélioration.
Étapes du Test	<ul style="list-style-type: none"> ○ ○ Exécuter le service avec ces entrées. ○ Comparer le résultat du service à un rayon d'amélioration calculé manuellement.
Critère de Réussite	Le rayon calculé correspond étroitement au rayon attendu, en tenant compte des données prédéfinies dans une marge d'erreur.
Résultat attendu avec Dataset fourni	Entrée : Cleaner0 Sortie : 0
Résultat attendu avec Dataset fourni	Entrée : Cleaner1 Sortie : 0.7
Résultat attendu avec Dataset fourni	Entrée : CleanerInnexistent Sortie : erreur, Mauvais ID

Service : computeAQIAtPoint

Test : Valider la Production d'une Valeur de Qualité de l'Air

Objectif	S'assurer que le service fournit des valeurs de qualité de l'air précises, soit directement à partir de capteurs, soit par estimation.
Entrées	Coordonnées géographiques (latitude, longitude), date et heure.
Résultats Attendus	Une valeur précise de qualité de l'air (double) pour l'emplacement et le moment spécifiés.

Étapes du Test	<ul style="list-style-type: none"> ○ Préparer un ensemble de points de données de capteur connus ou d'estimations pour un emplacement et un moment spécifiques. ○ Invoquer le service avec les entrées spécifiées. ○ Vérifier le résultat par rapport aux données connues ou aux valeurs estimées.
Critère de Réussite	La valeur de qualité de l'air du service est précise, avec une tolérance de $\pm 0,01$, que ce soit à partir de données de capteur directes ou d'estimations.
Résultat attendu avec le jeu de test 4	Entrée : 44;-1;2019-01-01 Sortie : 7
Résultat attendu avec le jeu de test 4	Entrée : 44;0;2019-01-01 Sortie : <i>Calcul :</i> <i>capteur 0 : 7 distance : 1</i> <i>capteur 1 : 8 distance : 0.3</i> <i>capteur 2 : 7 distance : 0.4</i> <i>capteur 3 : 8 distance 1.1</i> <i>capteur 4 : pas de valeur distance 1.8</i> <i>capteur 5 : 5 distance 2.5</i> <i>capteur 6 : 5 distance 3.2</i> <i>résultat : $(7/1+8/0.3+7/0.4+8/1.1+5/2.5) / (1+1/0.3+1/0.4+1/1.1+1/2.5) = 7.42$</i>
Résultat attendu avec le jeu de test 4	Entrée : 44;0;2000-01-01 Sortie : pas de mesures disponibles à cette date

Manuel d'utilisation - interface console

Bienvenue dans AirWatcher

AirWatcher permet aux agences gouvernementales, aux individus privés et aux fournisseurs de purificateurs d'air d'analyser efficacement les données sur la qualité de l'air. Ce manuel vous guide dans l'utilisation de l'interface utilisateur basée sur console.

Pour Commencer

Lors du lancement d'AirWatcher, vous êtes accueilli par la Console Principale. À partir de là, vous pouvez vous connecter (connexion non-implémentée) pour accéder à des fonctionnalités spécifiques en fonction de votre rôle d'utilisateur.

Pour démarrer l'application :

- Ouvrez votre interface de ligne de commande.
- Naviguez vers le répertoire de l'application AirWatcher.

- Si le projet n'a pas été compilé, écrivez `make` dans la console.
- Lancez l'application en tapant `./AirWatcher` et appuyez sur Entrée.

Fonctionnalités Principales

Après avoir lancé l'application, vous aurez accès à différentes fonctionnalités en fonction de votre rôle d'utilisateur. Les services communs disponibles incluent :

- **Calcul de la Moyenne de l'IQA dans un Cercle sur une Période (implémenté)**
- **Calcul de la Moyenne de l'IQA dans un Cercle à un Moment Spécifique**
- **Comparaison d'un Capteur avec les Autres**
- **Analyse des Capteurs Privés**
- **Mesurer l'Amélioration de la Qualité de l'Air**
- **Mesurer la Zone d'Amélioration (implémenté)**
- **Produire une Valeur de Qualité de l'Air**

Calcul de la Moyenne de l'IQA sur une Période :

- **Description:** Calcule la moyenne de l'IQA dans un cercle sur une période déterminée.
- **Utilisation:**
 1. Choisissez l'option dédiée au calcul périodique de l'IQA.
 2. Fournissez les coordonnées du centre du cercle, le rayon, ainsi que les dates et heures de début et de fin.
 3. Le service retourne la moyenne de l'IQA pour la période spécifiée.

Calcul de l'IQA à un Moment Spécifique :

- **Description:** Détermine la moyenne de l'IQA dans un cercle à un moment précis.
- **Utilisation:**
 1. Sélectionnez l'option pour le calcul de l'IQA à un moment spécifique.
 2. Saisissez les coordonnées du centre, le rayon, et la date.
 3. La moyenne de l'IQA dans le cercle pour le jour demandé sera affichée.

Comparaison des Capteurs :

- **Description:** Compare un capteur sélectionné avec d'autres pour évaluer sa cohérence.
- **Utilisation:**
 1. Choisissez l'option de comparaison de capteurs.
 2. Entrez l'identifiant du capteur que vous souhaitez comparer.
 3. Le système indiquera si le capteur est cohérent avec les autres.

Analyse des Capteurs Privés :

- **Description:** Analyse les données fournies par les capteurs privés pour déterminer leur validité.
- **Utilisation:**
 1. Sélectionnez l'option d'analyse des capteurs privés.
 2. Entrez l'identifiant du capteur privé concerné.
 3. L'application indique si le capteur (et potentiellement d'autres du même utilisateur) est considéré comme valide ou non.

Ajouter un Point aux Particulier

- **Description:** Ajoute un point par capteur de particulier utilisé dans la requête.
- **Utilisation:**
 1. Effectuez une requête impliquant un ou plusieurs capteurs de particuliers.
 2. Le système met automatiquement à jour le fichier points.csv, attribuant un point par capteur utilisé.

Mesurer l'Amélioration de la Qualité de l'Air

- **Description:** Calcule la différence de la qualité de l'air avant et après l'utilisation d'un purificateur.
- **Utilisation:**
 1. Entrez l'identifiant du purificateur et les moments à comparer.
 2. Le système retourne la différence de qualité de l'air comme une valeur double.

Mesurer la Zone d'Amélioration

- **Description:** Détermine le rayon de la zone d'amélioration de la qualité de l'air due à un purificateur.
- **Utilisation:**
 1. Entrez l'identifiant du purificateur et spécifiez le cadre temporel.
 2. Le système calcule et retourne le rayon de la zone d'amélioration comme une valeur double.

Produire une Valeur de Qualité de l'Air

- **Description:** Fournit une valeur réelle de la qualité de l'air d'un capteur ou une estimation en l'absence de capteur à l'emplacement spécifié.
- **Utilisation:**
 1. Entrez les coordonnées géographiques (latitude, longitude), la date et l'heure.
 2. Le système retourne une valeur de qualité de l'air comme une valeur double, basée sur les données de capteur disponibles ou des estimations

Déconnexion

Pour vous déconnecter et retourner à la Console Principale, sélectionnez simplement l'option **Logout** depuis votre console actuelle. Cela garantit que votre session est correctement fermée.

Conseils Supplémentaires

- Les commandes sont sensibles à la casse. Assurez-vous de les entrer exactement telles qu'elles sont montrées.

Dépannage

- **Données Non Affichées Correctement :** Vérifiez vos entrées pour toute erreur de frappe ou de format. Assurez-vous que les données existent dans les paramètres spécifiés.

Ce manuel est conçu pour aider les utilisateurs à naviguer efficacement dans l'application AirWatcher. Pour une assistance supplémentaire, veuillez vous référer aux informations de contact fournies par votre administrateur système ou le service d'assistance.

Document de design

Architecture

L'architecture "Layered architecture pattern" sera utilisée ici, elle offre une structure claire et une organisation logique qui aide à gérer la complexité des systèmes logiciels.

Caractéristiques Principales	<ul style="list-style-type: none">• Chaque couche se concentre sur une fonctionnalité distincte, ce qui facilite la compréhension, la maintenance et l'extension du système.• Les couches dépendent uniquement des couches directement en dessous d'elles, favorisant une structure hiérarchique.
Avantages	<ul style="list-style-type: none">• Encourage la séparation des préoccupations, rendant le code plus facile à comprendre et à maintenir.• Permet l'évolution et le remplacement des couches individuelles sans affecter l'ensemble du système.• Des fonctionnalités redondantes peuvent être fournies dans chaque couche pour augmenter la fiabilité du système.
Inconvénients	<ul style="list-style-type: none">• Surcharge: Peut introduire une complexité supplémentaire, en particulier dans les petites applications où les avantages pourraient ne pas être pleinement réalisés.• Surcharge de Communication: La communication stricte entre les couches peut entraîner une latence accrue dans certains scénarios. Les performances peuvent poser problème en raison des multiples niveaux d'interprétation d'une demande de service lors de son traitement à chaque couche.• En pratique, il est souvent difficile d'assurer une séparation nette entre les couches et une couche de niveau supérieur peut devoir interagir directement avec des couches de niveau inférieur plutôt qu'à travers la couche située immédiatement en dessous.

Console based user interface

Application management with Business Objects and Application services : AQI computing,
Sensors comparison and analyse etc...

Application functionalities with csv file interaction services

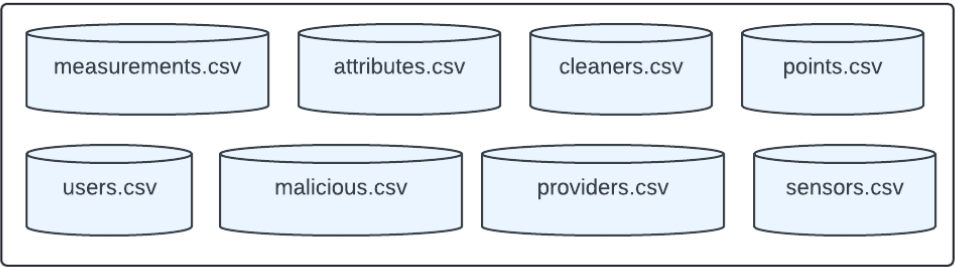
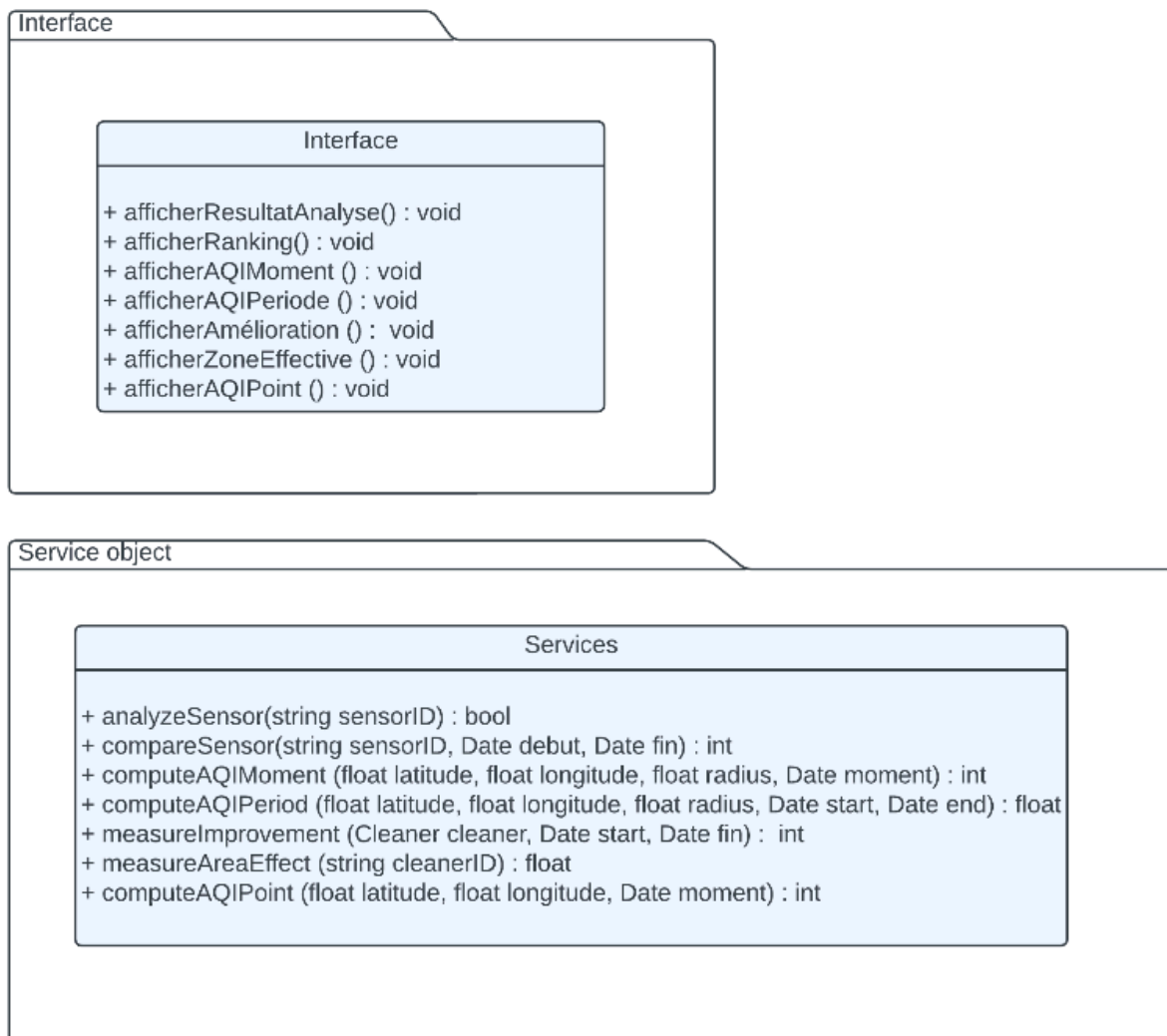
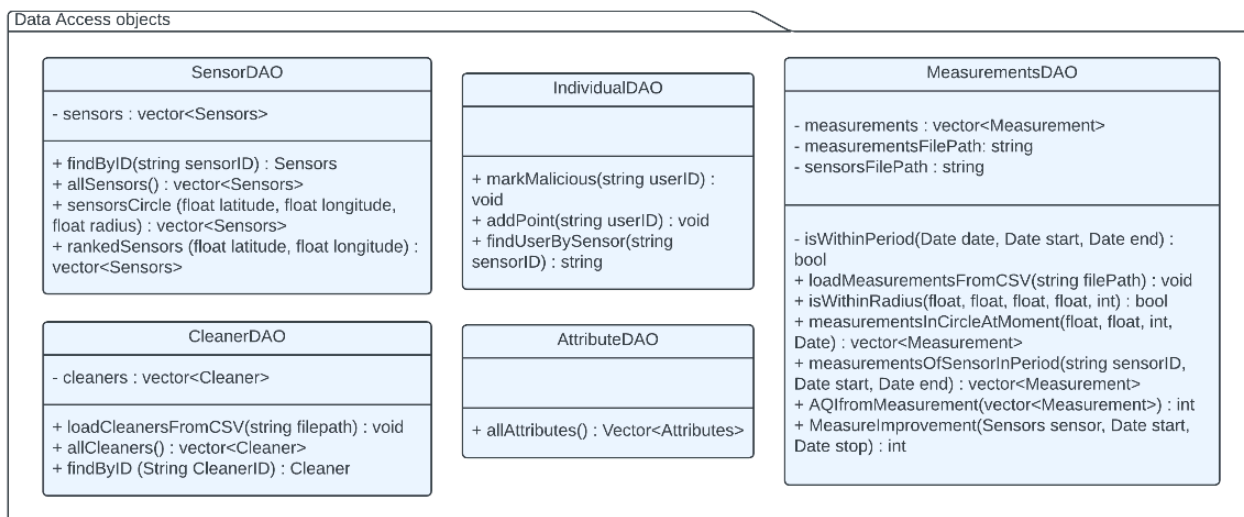
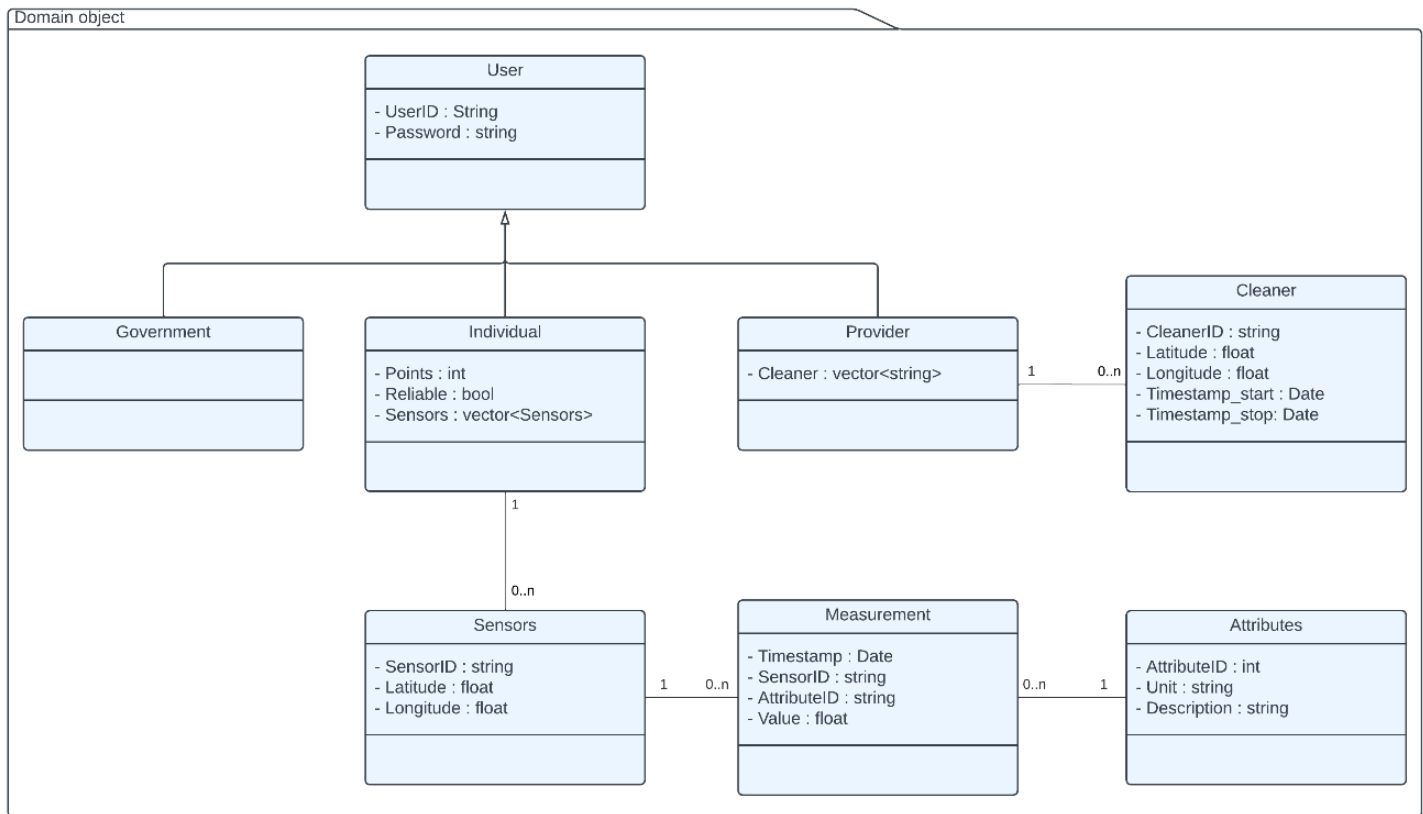


Diagramme de classes





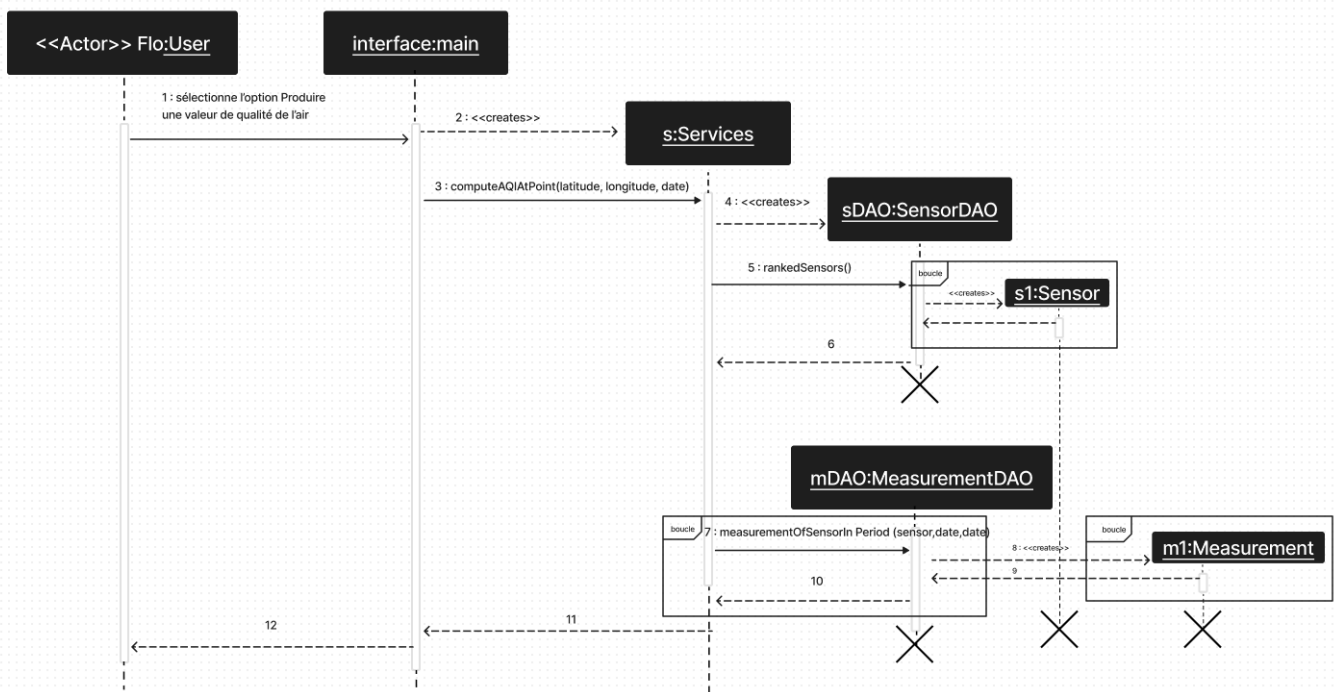
Diagrammes de séquence des trois scénarios majeurs

I- L'utilisateur produit une valeur de qualité de l'air

Description :

- Début de la séquence : Un utilisateur demande la valeur de qualité de l'air pour une localisation et un moment donné à travers l'interface système.
- L'interface fait appel au service computeAQIAtPoint() en créant une instance de Services
- Le service crée une instance de SensorsDAO et appelle sa fonction rankedSensors(latitude, longitude) pour récupérer un classement des capteurs des plus proches aux plus loins de l'endroit demandé.
- Le service crée une instance de MeasurementsDAO et fait appel à sa fonction measurementsOfSensorInPeriod(sensor,date,date), soit pour le capteur situé à l'endroit demandé, soit pour les 5 capteurs les plus proches s'il n'y en a pas, renvoyés par l'étape précédente.
- Le service calcule la valeur de qualité de l'air et la renvoie à l'interface.
- L'interface affiche la valeur pour l'utilisateur.

L'utilisateur produit une valeur de qualité de l'air



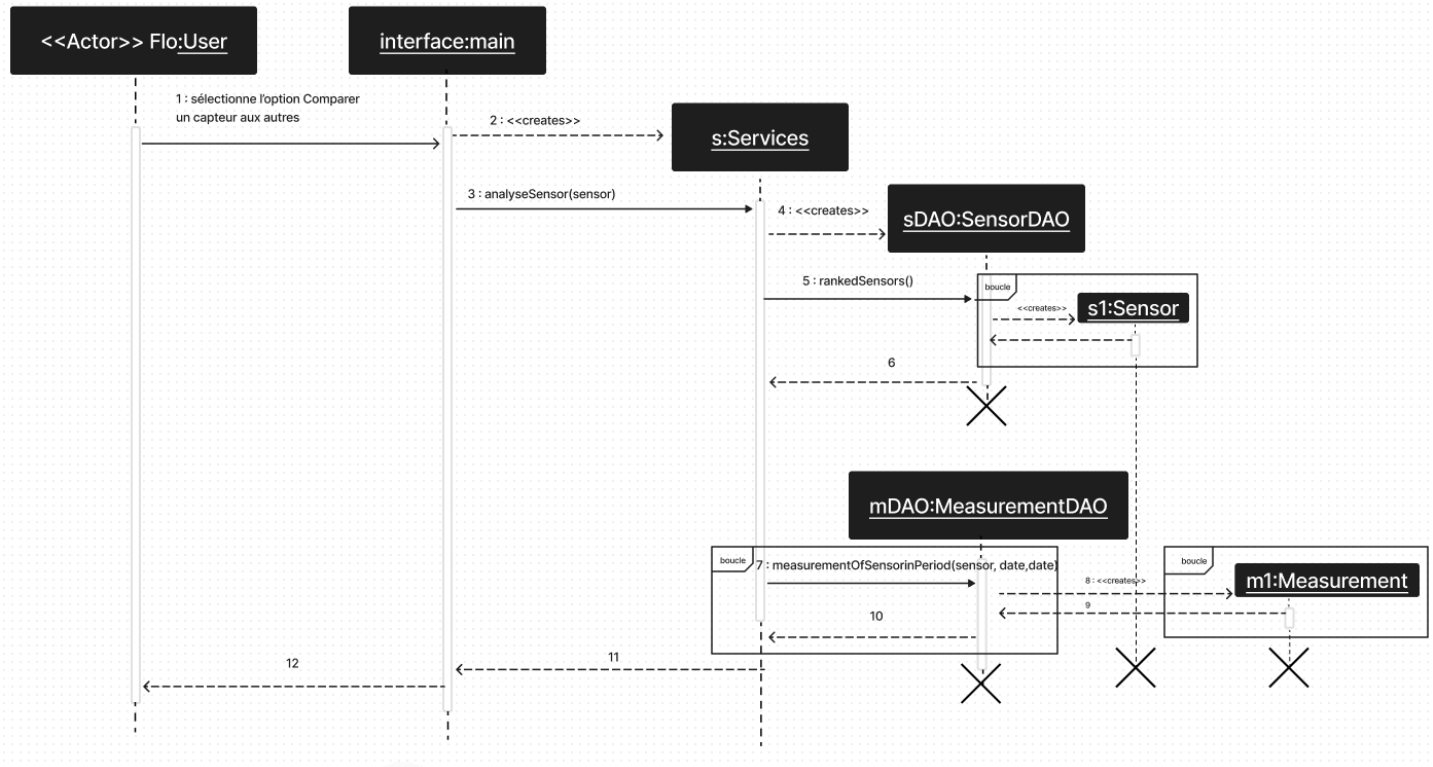
II- Le gouvernement veut comparer un capteur aux autres

Description :

- Début de la séquence : Un utilisateur gouvernemental initie une demande de comparaison de capteurs à partir de l'interface système.
- L'interface fait appel au service analyseSensor() en créant une instance de Services
- Le service crée une instance de SensorsDAO et appelle sa fonction rankedSensors(latitude, longitude) pour récupérer un classement des capteurs des plus proches aux plus loins de l'endroit demandé.

- Le service crée une instance de MeasurementsDAO et fait appel à sa fonction measurementsOfSensorInPeriod(sensor,date,date) pour le capteur en question ainsi que pour les 5 capteurs les plus proches, renvoyés par l'étape précédente.
- Le service calcule la valeur de qualité de l'air attendue et la compare à la valeur effective. Si la différence est supérieure à 10% : l'utilisateur est indiqué malicious.
- L'interface affiche la valeur pour l'utilisateur.

Le gouvernement compare un capteur aux autres

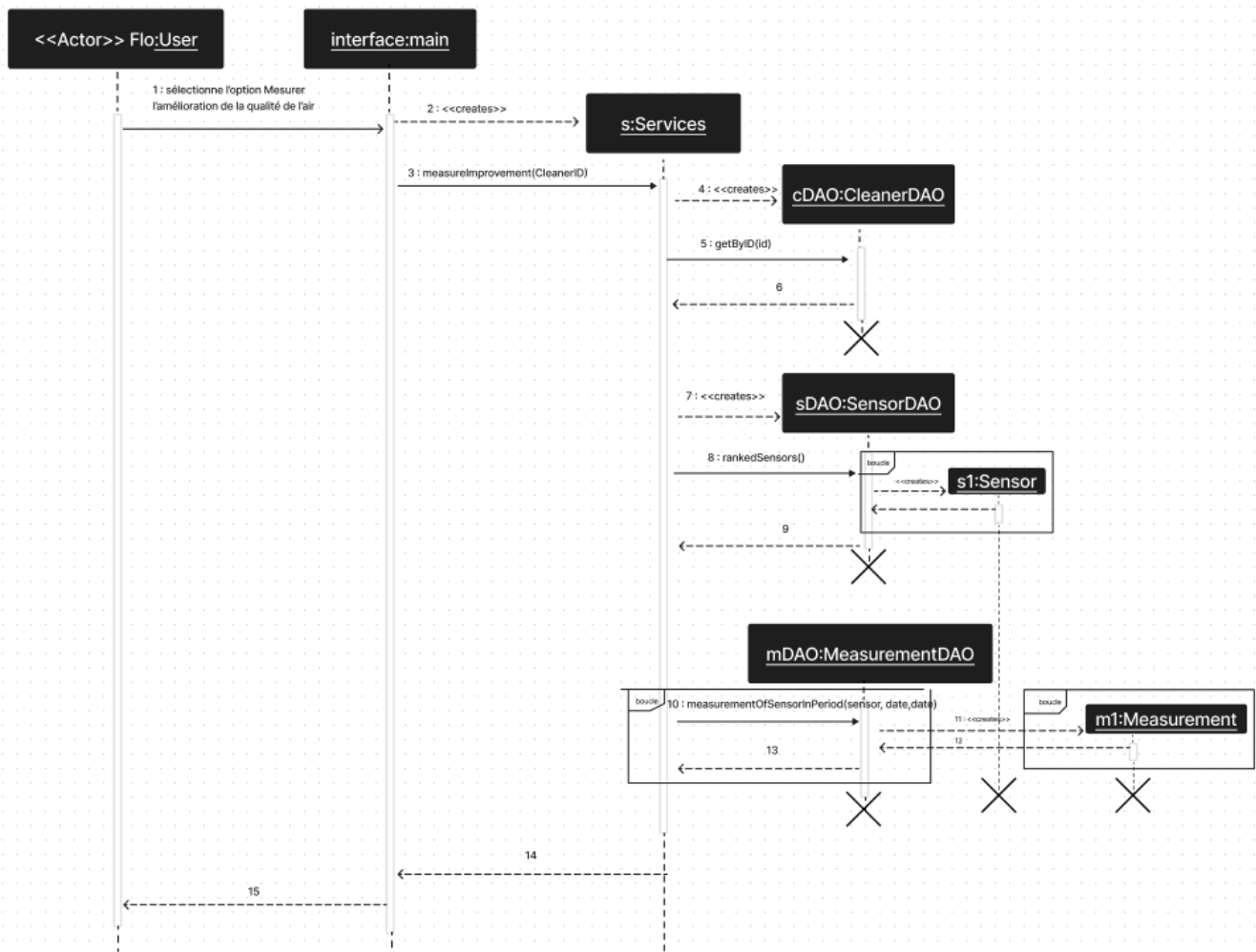


III- Le fabricant veut mesurer l'amélioration de la qualité de l'air

Description :

- Début de la séquence : Un fabricant demande la mesure de l'amélioration de la qualité de l'air due à un purificateur d'air via l'interface système.
- Récupération des données : Le système extrait les données des purificateurs d'air du fichier cleaners.csv.
- Traitement des données : La couche DataAccess utilise CleanerDAO.allCleaners() pour récupérer les informations des purificateurs concernés.
- Calcul de l'amélioration : Ces informations sont traitées par l'objet de domaine, qui utilise le service MeasureImprovement pour calculer l'amélioration de la qualité de l'air.
- Affichage des résultats : Les statistiques d'amélioration sont ensuite retournées et affichées sur l'interface du fabricant.

Le fabricant veut mesurer l'amélioration de la qualité de l'air



Descriptions et pseudo-codes de trois algorithmes

I- Calculer moyenne AQI dans un rayon (pour une période) - 1^{er} implémentation

```

computeAQIPeriod (latitude_centre, longitude centre, rayon, datedeb, datefin) {
    list_sensors = [], list_mesures = []
    for each Sensor uncaptuer { //chargé depuis SensorDAO
        distance = sqrt((latitude_centre - uncaptuer.latitude)^2 + (longitude_centre -
uncaptuer.longitude)^2)
        if distance <= rayon && capteur !estCorrompu() { //fonction séparée dans le code
            ajouter uncaptuer à list_sensors
            rechercher les mesures associées à l'ID de uncaptuer //grâce à
MeasurementDAO
            for each mesure {
                if mesure.dateMesure >= datedeb et mesure.dateMesure < datefin {
                    si la mesure vient d'un capteur d'un particulier: lui ajouter un
point
                }
            }
        }
    }
}
    
```

```

                                ajouter mesure à list_mesures
                                }
                            }
                        }
                    }
                }
            }
            grouper les mesures de list_mesures par attribut
            For each Attribut { // fonction séparée dans le code
                faire une moyenne des valeurs mesurées
                déterminer l'indice ATMO associé à cette moyenne
            }
            aqi = min(atmo)
            retourner aqi
        }
    }
}

```

II- Evaluer et classer les capteurs en fonction de leur similarité à un capteur donné - **non implémenté**

```

CompareSensor (Sensor lecapteur, Date datedeb, Date datefin) {
    differences; //2D array qui associe chaque capteur à son score de différence
    pour chaque capteur c { //chargé depuis SensorsDAO
        diffGlobale = 0;
        pour (t=debut;t<=fin;t++){
            calculer l'atmo de lecapteur;
            calculer l'atmo du capteur c;
            calculer la différence entre les deux diff;
            diffGlobale += diff;
        }
        stocker diffGlobale dans le 2D array differences;
    }
    classer en ordre croissant differences en fonction du score;
    afficher differences;
}

```

III- Mesurer la zone d'amélioration de la qualité de l'air - **2nd implémentation**

```

MeasureAreaEffect(String purifID) {
    récupérer le cleaner lepurif grâce à l'ID; //grâce à CleanerDAO
    rayon = 0
    val_ref = MeasureImprovement(lepurif, datedeb-1, datefin-1)
    si val_ref !=0 {
        list_sensors = rankedSensors(lepurif.latitude, lepurif.longitude) //fonction qui trie tous
        les capteurs par ordre croissant de distance à un point donné en paramètre
        int i = 0
        uncapteur = list_sensor[i]
        val = MeasureImprovement(uncapteur.latitude, uncapteur.longitude, datedeb-1,
        datefin-1)
    }
}

```

```

        Tant que (val>val_ref-10) {
            rayon = sqrt((lepurif.latitude-uncleaner.latitude)^2 +
(lepurif.longitude-uncleaner.longitude)^2)
            uncapteur = list_sensor[i]
            val = MeasureImprovement(uncapteur.latitude, uncapteur.longitude,
datedeb-1, datefin-1)
            i = i+1
        }

    retourner rayon
}
}

```

Unit tests

Test des méthodes des Data Access Objects.

Service : allSensors

Cas de Test : Retourner l'ensemble des Capteurs du fichier Sensors.csv

Objectif	S'assurer que la méthode soit capable de lire le fichier csv et de convertir chaque ligne lue en un objet <i>Sensors</i> qu'il ajoute à une liste.												
Résultats attendus	Une liste d'objets de type <i>Sensors</i>												
Étapes du Test	<ul style="list-style-type: none"> ○ Appeler le service ○ Comparer le résultat du service avec une liste de capteurs créée manuellement selon le jeu de test. 												
Critère de réussite	Le résultat du service correspond à la liste créée.												
Résultat attendu avec jeu de test 1	Entrée : Sortie : <table> <tr> <td><i>Indice</i></td><td><i>Objet</i></td></tr> <tr> <td>0</td><td>Sensor0;44;-1</td></tr> <tr> <td>1</td><td>Sensor1;44;-0.3</td></tr> <tr> <td>2</td><td>Sensor2;44;0.4</td></tr> <tr> <td>3</td><td>Sensor3;44;1.1</td></tr> <tr> <td>4</td><td>Sensor4;44;1.8</td></tr> </table>	<i>Indice</i>	<i>Objet</i>	0	Sensor0;44;-1	1	Sensor1;44;-0.3	2	Sensor2;44;0.4	3	Sensor3;44;1.1	4	Sensor4;44;1.8
<i>Indice</i>	<i>Objet</i>												
0	Sensor0;44;-1												
1	Sensor1;44;-0.3												
2	Sensor2;44;0.4												
3	Sensor3;44;1.1												
4	Sensor4;44;1.8												
Résultat attendu si fichier sensors.csv vide	Entrée : Sortie : liste vide <table> <tr> <td><i>Indice</i></td><td><i>Objet</i></td></tr> </table>	<i>Indice</i>	<i>Objet</i>										
<i>Indice</i>	<i>Objet</i>												
Résultat attendu si fichier inexistant	Entrée : Sortie : erreur lors de l'ouverture du fichier												

Service : sensorsInCircle

Cas de Test : Retourner l'ensemble des Capteurs situé dans le rayon d'une coordonnée

Objectif	S'assurer que la méthode soit capable de lire le fichier csv, de sélectionner les capteurs situés dans la zone demandée et de les ajouter à une liste.								
Résultats attendus	Une liste d'objets de type <i>Sensors</i>								
Étapes du Test	<ul style="list-style-type: none">○ Appeler le service○ Comparer le résultat du service avec une liste de capteurs créée manuellement selon le jeu de test.								
Critère de réussite	Le résultat du service correspond à la liste créée.								
Résultat attendu avec jeu de test 1	Entrée : 44;0;1 Sortie : <table><thead><tr><th><i>Indice</i></th><th><i>Objet</i></th></tr></thead><tbody><tr><td>0</td><td>Sensor0;44;-1</td></tr><tr><td>1</td><td>Sensor1;44;-0.3</td></tr><tr><td>2</td><td>Sensor2;44;0.4</td></tr></tbody></table>	<i>Indice</i>	<i>Objet</i>	0	Sensor0;44;-1	1	Sensor1;44;-0.3	2	Sensor2;44;0.4
<i>Indice</i>	<i>Objet</i>								
0	Sensor0;44;-1								
1	Sensor1;44;-0.3								
2	Sensor2;44;0.4								
	Entrée : 44;0;-1 Sortie : Erreur : le rayon est négatif								
	Entrée : 40;0;1 Sortie : Erreur : aucun capteur dans cette zone								

Service : rankedSensors

Cas de Test : Établir un classement des capteurs par proximité géographique par rapport à un point donné

Objectif	
Résultats attendus	Une liste d'objets <i>Sensors</i> avec leur rang
Étapes du Test	<ul style="list-style-type: none">○ Appeler le service○ Comparer le résultat du service avec une liste de capteurs créée manuellement selon le jeu de test.
Critère de réussite	Le résultat du service correspond à la liste créée.

Résultat attendu avec jeu de test 1	Entrée: 44;0 Sortie: <i>Indice</i> <i>Objet</i> 0 Sensor1;44;-0.3 1 Sensor2;44;0.4 2 Sensor0;44;-1 3 Sensor3;44;1.1 3 Sensor4;44;1.8
Résultat attendu si le paramètre est null	Entrée : null Sortie : Erreur, coordonnées null

Service : markMalicious

Cas de Test : marquer un utilisateur comme malicious

Objectif	interagir avec le fichier malicious.csv afin d'y ajouter l'utilisateur passé en paramètre
Résultats attendus	Mise à jour du fichier malicious.csv
Étapes du Test	<ul style="list-style-type: none"> ○ Appeler le service ○ Comparer le fichier csv avec un fichier créé manuellement
Critère de réussite	Le fichier mis à jour correspond au fichier créé
Résultat attendu avec jeu de test 1	Entrée : User0 Sortie : <i>malicious.csv</i> User0;
Résultat attendu lors du 2e appel	Entrée : User2 Sortie : <i>malicious.csv</i> User0; User2;
Résultat attendu si le paramètre est null	Entrée : null Sortie : Erreur, utilisateur null

Service : addPoint

Cas de Test : ajouter un point à un utilisateur si son capteur est utilisé

Objectif	interagir avec le fichier points.csv pour incrémenter de 1 le nombre de point de l'utilisateur concerné
-----------------	---

Résultats attendus	Mise à jour du fichier points.csv
Étapes du Test	<ul style="list-style-type: none"> ○ Appeler le service ○ Comparer le fichier csv avec un fichier créé manuellement
Critère de réussite	Le fichier mis à jour correspond au fichier créé
Résultat attendu avec jeu de test 1	Entrée : User0 Sortie : <i>points.csv</i> User0;1
Résultat attendu lors du 2e appel	Entrée : User0 Sortie : <i>points.csv</i> User0;2
Résultat attendu si le paramètre est null	Entrée : null Sortie : Erreur, utilisateur null

Service : allAttributes

Cas de Test : Retourner l'ensemble des attributs du fichier attributes.csv

Objectif	S'assurer que la méthode soit capable de lire le fichier csv et de convertir chaque ligne lue en un objet <i>Attributes</i> qu'il ajoute à une liste.										
Résultats attendus	Une liste d'objets de type <i>Attributes</i>										
Étapes du Test	<ul style="list-style-type: none"> ○ Appeler le service ○ Comparer le résultat du service avec une liste d'attributs créée manuellement selon le jeu de test. 										
Critère de réussite	Le résultat du service correspond à la liste créée.										
Résultat attendu avec jeu de test 1	Entrée : Sortie : <table> <thead> <tr> <th><i>Indice</i></th><th><i>Objet</i></th></tr> </thead> <tbody> <tr> <td>0</td><td>O3;µg/m3;concentration d'ozone</td></tr> <tr> <td>1</td><td>SO2;µg/m3;concentration de dioxyde de soufre</td></tr> <tr> <td>2</td><td>NO2;µg/m3;concentration de dioxyde d'azote</td></tr> <tr> <td>3</td><td>PM10;µg/m3;concentration de particules fines</td></tr> </tbody> </table>	<i>Indice</i>	<i>Objet</i>	0	O3;µg/m3;concentration d'ozone	1	SO2;µg/m3;concentration de dioxyde de soufre	2	NO2;µg/m3;concentration de dioxyde d'azote	3	PM10;µg/m3;concentration de particules fines
<i>Indice</i>	<i>Objet</i>										
0	O3;µg/m3;concentration d'ozone										
1	SO2;µg/m3;concentration de dioxyde de soufre										
2	NO2;µg/m3;concentration de dioxyde d'azote										
3	PM10;µg/m3;concentration de particules fines										
Résultat attendu si fichier attributes.csv vide	Entrée : Sortie : liste vide										

	<i>Indice</i> <i>Objet</i>
Résultat attendu si fichier inexistant	Entrée : Sortie : erreur lors de l'ouverture du fichier

Service : measurementsInCircleInPeriod

Cas de Test : Retourner les mesures dans la zone donnée, sur une période donnée

Objectif	S'assurer que la méthode soit capable de sélectionner les mesures situées dans la zone demandée, pour la période demandée et de les ajouter à une liste, à condition qu'elles ne proviennent pas d'un capteur corrompu.																										
Résultats attendus	Une liste d'objets de type <i>Measurement</i>																										
Étapes du Test	<ul style="list-style-type: none"> ○ Appeler le service ○ Comparer le résultat du service avec une liste de capteurs créée manuellement selon le jeu de test. 																										
Critère de réussite	Le résultat du service correspond à la liste créée.																										
Résultat attendu avec jeu de test 2	Entrée : 44;0;1;2019-01-02;2019-01-07 Sortie : <table> <thead> <tr> <th><i>Indice</i></th><th><i>Objet</i></th></tr> </thead> <tbody> <tr><td>0</td><td>2019-01-02 12:00:00 Sensor0 O3 50.5</td></tr> <tr><td>1</td><td>2019-01-02 12:00:00 Sensor0 NO2 72</td></tr> <tr><td>2</td><td>2019-01-02 12:00:00 Sensor0 SO2 39.25</td></tr> <tr><td>3</td><td>2019-01-02 12:00:00 Sensor0 PM10 50.5</td></tr> <tr><td>4</td><td>2019-01-02 12:00:00 Sensor1 O3 65.08</td></tr> <tr><td>5</td><td>2019-01-02 12:00:00 Sensor1 NO2 60.33</td></tr> <tr><td>6</td><td>2019-01-02 12:00:00 Sensor1 SO2 32.88</td></tr> <tr><td>7</td><td>2019-01-02 12:00:00 Sensor1 PM10 54.58</td></tr> <tr><td>8</td><td>2019-01-02 12:00:00 Sensor2 O3 47.68</td></tr> <tr><td>9</td><td>2019-01-02 12:00:00 Sensor2 NO2 43.22</td></tr> <tr><td>10</td><td>2019-01-02 12:00:00 Sensor2 SO2 46.98</td></tr> <tr><td>11</td><td>2019-01-02 12:00:00 Sensor2 PM10 45.1</td></tr> </tbody> </table>	<i>Indice</i>	<i>Objet</i>	0	2019-01-02 12:00:00 Sensor0 O3 50.5	1	2019-01-02 12:00:00 Sensor0 NO2 72	2	2019-01-02 12:00:00 Sensor0 SO2 39.25	3	2019-01-02 12:00:00 Sensor0 PM10 50.5	4	2019-01-02 12:00:00 Sensor1 O3 65.08	5	2019-01-02 12:00:00 Sensor1 NO2 60.33	6	2019-01-02 12:00:00 Sensor1 SO2 32.88	7	2019-01-02 12:00:00 Sensor1 PM10 54.58	8	2019-01-02 12:00:00 Sensor2 O3 47.68	9	2019-01-02 12:00:00 Sensor2 NO2 43.22	10	2019-01-02 12:00:00 Sensor2 SO2 46.98	11	2019-01-02 12:00:00 Sensor2 PM10 45.1
<i>Indice</i>	<i>Objet</i>																										
0	2019-01-02 12:00:00 Sensor0 O3 50.5																										
1	2019-01-02 12:00:00 Sensor0 NO2 72																										
2	2019-01-02 12:00:00 Sensor0 SO2 39.25																										
3	2019-01-02 12:00:00 Sensor0 PM10 50.5																										
4	2019-01-02 12:00:00 Sensor1 O3 65.08																										
5	2019-01-02 12:00:00 Sensor1 NO2 60.33																										
6	2019-01-02 12:00:00 Sensor1 SO2 32.88																										
7	2019-01-02 12:00:00 Sensor1 PM10 54.58																										
8	2019-01-02 12:00:00 Sensor2 O3 47.68																										
9	2019-01-02 12:00:00 Sensor2 NO2 43.22																										
10	2019-01-02 12:00:00 Sensor2 SO2 46.98																										
11	2019-01-02 12:00:00 Sensor2 PM10 45.1																										
Résultat attendu avec jeu de test 2	Entrée : 44;0;1;2019-01-04;2019-01-07 Sortie : liste vide (aucune mesure correspondante n'a été trouvée)																										
	Entrée : 40;0;-1;2019-01-02;2019-01-07 Sortie : Erreur : le rayon est négatif																										

Service : measurementsInCircleAtMoment

Cas de Test : Retourner les mesures dans la zone donnée, à un instant donné

Objectif	S'assurer que la méthode soit capable de sélectionner les mesures situées dans la zone demandée, à l'instant spécifié et de les ajouter à une liste, à condition qu'elles ne proviennent pas d'un capteur corrompu.																										
Résultats attendus	Une liste d'objets de type <i>Measurement</i>																										
Étapes du Test	<ul style="list-style-type: none">○ Appeler le service○ Comparer le résultat du service avec une liste de capteurs créée manuellement selon le jeu de test.																										
Critère de réussite	Le résultat du service correspond à la liste créée.																										
Résultat attendu avec jeu de test 2	<p>Entrée : 44;0;1;2019-01-02</p> <p>Sortie :</p> <table><thead><tr><th>Indice</th><th>Objet</th></tr></thead><tbody><tr><td>0</td><td>2019-01-02 12:00:00 Sensor0 O3 50.5</td></tr><tr><td>1</td><td>2019-01-02 12:00:00 Sensor0 NO2 72</td></tr><tr><td>2</td><td>2019-01-02 12:00:00 Sensor0 SO2 39.25</td></tr><tr><td>3</td><td>2019-01-02 12:00:00 Sensor0 PM10 50.5</td></tr><tr><td>4</td><td>2019-01-02 12:00:00 Sensor1 O3 65.08</td></tr><tr><td>5</td><td>2019-01-02 12:00:00 Sensor1 NO2 60.33</td></tr><tr><td>6</td><td>2019-01-02 12:00:00 Sensor1 SO2 32.88</td></tr><tr><td>7</td><td>2019-01-02 12:00:00 Sensor1 PM10 54.58</td></tr><tr><td>8</td><td>2019-01-02 12:00:00 Sensor2 O3 47.68</td></tr><tr><td>9</td><td>2019-01-02 12:00:00 Sensor2 NO2 43.22</td></tr><tr><td>10</td><td>2019-01-02 12:00:00 Sensor2 SO2 46.98</td></tr><tr><td>11</td><td>2019-01-02 12:00:00 Sensor2 PM10 45.1</td></tr></tbody></table>	Indice	Objet	0	2019-01-02 12:00:00 Sensor0 O3 50.5	1	2019-01-02 12:00:00 Sensor0 NO2 72	2	2019-01-02 12:00:00 Sensor0 SO2 39.25	3	2019-01-02 12:00:00 Sensor0 PM10 50.5	4	2019-01-02 12:00:00 Sensor1 O3 65.08	5	2019-01-02 12:00:00 Sensor1 NO2 60.33	6	2019-01-02 12:00:00 Sensor1 SO2 32.88	7	2019-01-02 12:00:00 Sensor1 PM10 54.58	8	2019-01-02 12:00:00 Sensor2 O3 47.68	9	2019-01-02 12:00:00 Sensor2 NO2 43.22	10	2019-01-02 12:00:00 Sensor2 SO2 46.98	11	2019-01-02 12:00:00 Sensor2 PM10 45.1
Indice	Objet																										
0	2019-01-02 12:00:00 Sensor0 O3 50.5																										
1	2019-01-02 12:00:00 Sensor0 NO2 72																										
2	2019-01-02 12:00:00 Sensor0 SO2 39.25																										
3	2019-01-02 12:00:00 Sensor0 PM10 50.5																										
4	2019-01-02 12:00:00 Sensor1 O3 65.08																										
5	2019-01-02 12:00:00 Sensor1 NO2 60.33																										
6	2019-01-02 12:00:00 Sensor1 SO2 32.88																										
7	2019-01-02 12:00:00 Sensor1 PM10 54.58																										
8	2019-01-02 12:00:00 Sensor2 O3 47.68																										
9	2019-01-02 12:00:00 Sensor2 NO2 43.22																										
10	2019-01-02 12:00:00 Sensor2 SO2 46.98																										
11	2019-01-02 12:00:00 Sensor2 PM10 45.1																										

Service : measurementOfSensorInPeriod

Cas de Test : Retourner l'ensemble des mesures produites par un capteur donné

Objectif	S'assurer que la méthode soit capable de lire le fichier csv, de sélectionner les mesures associées au capteur donné et de les ajouter à une liste, si le sensor n'appartient pas à un user inscrit dans malicious.csv.
-----------------	---

Résultats attendus	Une liste d'objets de type <i>Measurement</i>
Étapes du Test	<ul style="list-style-type: none"> ○ Appeler le service ○ Comparer le résultat du service avec une liste de mesures créée manuellement selon le jeu de test.
Critère de réussite	Le résultat du service correspond à la liste créée.
Résultat attendu avec jeu de test 1	Entrée : Sensor0, 2019-01-01, 2019-01-05 Sortie : <i>Objet</i> 2019-01-01 12:00:00 Sensor0 O3 50.25 2019-01-01 12:00:00 Sensor0 NO2 74.5 2019-01-01 12:00:00 Sensor0 SO2 41.5 2019-01-01 12:00:00 Sensor0 PM10 44.75
Résultat attendu si fichier attributes.csv vide	Entrée : Sensor2 Sortie : null
Résultat attendu avec jeu de test 3	Entrée : Sensor17 Sortie : null

Annexe 1 : jeux de tests

Jeux de tests

Jeu de test 1

measurements.csv

Timestamp;SensorID;AttributeID;Value;

```

2019-01-01 12:00:00 Sensor0 O3 50.25 /2
2019-01-01 12:00:00 Sensor0 NO2 74.5 /3
2019-01-01 12:00:00 Sensor0 SO2 41.5 /2
2019-01-01 12:00:00 Sensor0 PM10 44.75 /7
2019-01-01 12:00:00 Sensor1 O3 63.04 /3
2019-01-01 12:00:00 Sensor1 NO2 61.92 /3
2019-01-01 12:00:00 Sensor1 SO2 34.42 /1
2019-01-01 12:00:00 Sensor1 PM10 51.12 /8
2019-01-01 12:00:00 Sensor2 O3 47.84 /2

```

2019-01-01 12:00:00	Sensor2	NO2	43.32	/2
2019-01-01 12:00:00	Sensor2	SO2	46.24	/2
2019-01-01 12:00:00	Sensor2	PM10	45.02	/7
2019-01-01 12:00:00	Sensor3	O3	24.97	/1
2019-01-01 12:00:00	Sensor3	NO2	42.05	/2
2019-01-01 12:00:00	Sensor3	SO2	44.54	/2
2019-01-01 12:00:00	Sensor3	PM10	50	/8
2019-01-07 12:00:00	Sensor4	O3	25.93	
2019-01-07 12:00:00	Sensor4	NO2	60.42	
2019-01-07 12:00:00	Sensor4	SO2	57.67	
2019-01-07 12:00:00	Sensor4	PM10	40.46	

sensors.csv

SensorID;Latitude;Longitude;

Sensor0	44	-1
Sensor1	44	-0.3
Sensor2	44	0.4
Sensor3	44	1.1
Sensor4	44	1.8

attributes.csv

AttributeID;Unit;Description

O3	µg/m3	concentration d'ozone
SO2	µg/m3	concentration de dioxyde de soufre
NO2	µg/m3	concentration de dioxyde d'azote
PM10	µg/m3	concentration de particules fines

providers.csv

Provider0	Cleaner0
Provider1	Cleaner1

cleaners.csv

Cleaner0	45.333333	1.333333	2019-02-01 12:00:00	2019-03-01 00:00:00
Cleaner1	46.666667	3.666667	2019-02-01 12:00:00	2019-03-01 00:00:00

users.csv

User0	Sensor70
User1	Sensor36

Jeu de test 2

measurements.csv

Timestamp;SensorID;AttributeID;Value;

2019-01-01 12:00:00	Sensor0	O3	50.25	/2
2019-01-01 12:00:00	Sensor0	NO2	74.5	/3
2019-01-01 12:00:00	Sensor0	SO2	41.5	/2
2019-01-01 12:00:00	Sensor0	PM10	44.75	/7

2019-01-02 12:00:00	Sensor0	O3	50.5	/2
2019-01-02 12:00:00	Sensor0	NO2	72	/3
2019-01-02 12:00:00	Sensor0	SO2	39.25	/2
2019-01-02 12:00:00	Sensor0	PM10	50.5	/8
2019-01-01 12:00:00	Sensor1	O3	63.04	/3
2019-01-01 12:00:00	Sensor1	NO2	61.92	/3
2019-01-01 12:00:00	Sensor1	SO2	34.42	/1
2019-01-01 12:00:00	Sensor1	PM10	51.12	/8
2019-01-02 12:00:00	Sensor1	O3	65.08	/3
2019-01-02 12:00:00	Sensor1	NO2	60.33	/3
2019-01-02 12:00:00	Sensor1	SO2	32.88	/1
2019-01-02 12:00:00	Sensor1	PM10	54.58	/8
2019-01-01 12:00:00	Sensor2	O3	47.84	/2
2019-01-01 12:00:00	Sensor2	NO2	43.32	/2
2019-01-01 12:00:00	Sensor2	SO2	46.24	/2
2019-01-01 12:00:00	Sensor2	PM10	45.02	/7
2019-01-02 12:00:00	Sensor2	O3	47.68	/2
2019-01-02 12:00:00	Sensor2	NO2	43.22	/2
2019-01-02 12:00:00	Sensor2	SO2	46.98	/2
2019-01-02 12:00:00	Sensor2	PM10	45.1	/7
2019-01-01 12:00:00	Sensor3	O3	24.97	/1
2019-01-01 12:00:00	Sensor3	NO2	42.05	/2
2019-01-01 12:00:00	Sensor3	SO2	44.54	/2
2019-01-01 12:00:00	Sensor3	PM10	50	/8
2019-01-07 12:00:00	Sensor4	O3	20.42	/1
2019-01-07 12:00:00	Sensor4	NO2	20.42	/1
2019-01-07 12:00:00	Sensor4	SO2	10.90	/1
2019-01-07 12:00:00	Sensor4	PM10	10.13	/2

//attention corrompu

sensors.csv

SensorID;Latitude;Longitude;

Sensor0	44	-1
Sensor1	44	-0.3
Sensor2	44	0.4
Sensor3	44	1.1
Sensor4	44	1.8

attributes.csv

AttributeID;Unit;Description

O3	µg/m3	concentration d'ozone
SO2	µg/m3	concentration de dioxyde de soufre
NO2	µg/m3	concentration de dioxyde d'azote
PM10	µg/m3	concentration de particules fines

providers.csv

Provider0	Cleaner0
Provider1	Cleaner1

cleaners.csv

Cleaner0	45.333333	1.333333	2019-02-01 12:00:00	2019-03-01 00:00:00
Cleaner1	46.666667	3.666667	2019-02-01 12:00:00	2019-03-01 00:00:00

users.csv

User0	Sensor70
User1	Sensor36

Jeu de test 3

measurements.csv

Timestamp;SensorID;AttributeID;Value;

2019-01-31 12:00:00	Sensor33	O3	49.77
2019-01-31 12:00:00	Sensor33	NO2	51.36
2019-01-31 12:00:00	Sensor33	SO2	42.33
2019-01-31 12:00:00	Sensor33	PM10	57.43 /8
2019-02-28 12:00:00	Sensor33	O3	48.23
2019-02-28 12:00:00	Sensor33	NO2	51.37
2019-02-28 12:00:00	Sensor33	SO2	44.47
2019-02-28 12:00:00	Sensor33	PM10	55.56 /8

2019-02-28 12:00:00	Sensor71	O3	42.78
2019-02-28 12:00:00	Sensor71	NO2	40.77
2019-02-28 12:00:00	Sensor71	SO2	56.01
2019-02-28 12:00:00	Sensor71	PM10	50.65
2019-01-31 12:00:00	Sensor71	O3	43.26
2019-01-31 12:00:00	Sensor71	NO2	39.58
2019-01-31 12:00:00	Sensor71	SO2	56.51
2019-01-31 12:00:00	Sensor71	PM10	51.07

2019-01-31 12:00:00	Sensor73	O3	58.11
2019-01-31 12:00:00	Sensor73	NO2	29
2019-01-31 12:00:00	Sensor73	SO2	61.95
2019-01-31 12:00:00	Sensor73	PM10	63.74
2019-02-28 12:00:00	Sensor73	O3	59.84
2019-02-28 12:00:00	Sensor73	NO2	29.52
2019-02-28 12:00:00	Sensor73	SO2	61.54
2019-02-28 12:00:00	Sensor73	PM10	64.52

2019-01-31 12:00:00	Sensor76	O3	55.3
2019-01-31 12:00:00	Sensor76	NO2	37.12
2019-01-31 12:00:00	Sensor76	SO2	36.39

2019-01-31 12:00:00	Sensor76	PM10	56.14
2019-02-28 12:00:00	Sensor76	O3	3.43
2019-02-28 12:00:00	Sensor76	NO2	2.44
2019-02-28 12:00:00	Sensor76	SO2	2.26
2019-02-28 12:00:00	Sensor76	PM10	3.37
2019-01-31 12:00:00	Sensor77	O3	44.58
2019-01-31 12:00:00	Sensor77	NO2	29.88
2019-01-31 12:00:00	Sensor77	SO2	34.54
2019-01-31 12:00:00	Sensor77	PM10	47.41 /7
2019-02-28 12:00:00	Sensor77	O3	2.75
2019-02-28 12:00:00	Sensor77	NO2	1.82
2019-02-28 12:00:00	Sensor77	SO2	2.06
2019-02-28 12:00:00	Sensor77	PM10	2.87 /1

sensors.csv

SensorID;Latitude;Longitude;

Sensor0	44	-1
Sensor1	44	-0.3
Sensor2	44	0.4
Sensor3	44	1.1
Sensor4	44	1.8

attributes.csv

AttributeID;Unit;Description

O3	µg/m3	concentration d'ozone
SO2	µg/m3	concentration de dioxyde de soufre
NO2	µg/m3	concentration de dioxyde d'azote
PM10	µg/m3	concentration de particules fines

providers.csv

Provider0	Cleaner0
Provider1	Cleaner1

cleaners.csv

Cleaner0	45.333333	1.333333	2019-02-01 12:00:00	2019-03-01 00:00:00
//Sensor33				
Cleaner1	46.666667	3.666667	2019-02-01 12:00:00	2019-03-01 00:00:00
//Sensor77				

users.csv

User0	Sensor70
User1	Sensor36

Jeu de test 4

measurements.csv

Timestamp;SensorID;AttributeID;Value;

2019-01-01 12:00:00	Sensor0	O3	50.25	/2
2019-01-01 12:00:00	Sensor0	NO2	74.5	/3
2019-01-01 12:00:00	Sensor0	SO2	41.5	/2
2019-01-01 12:00:00	Sensor0	PM10	44.75	/7
2019-01-02 12:00:00	Sensor0	O3	50.5	/2
2019-01-02 12:00:00	Sensor0	NO2	72	/3
2019-01-02 12:00:00	Sensor0	SO2	39.25	/2
2019-01-02 12:00:00	Sensor0	PM10	50.5	/8
2019-01-01 12:00:00	Sensor1	O3	63.04	/3
2019-01-01 12:00:00	Sensor1	NO2	61.92	/3
2019-01-01 12:00:00	Sensor1	SO2	34.42	/1
2019-01-01 12:00:00	Sensor1	PM10	51.12	/8
2019-01-02 12:00:00	Sensor1	O3	65.08	/3
2019-01-02 12:00:00	Sensor1	NO2	60.33	/3
2019-01-02 12:00:00	Sensor1	SO2	32.88	/1
2019-01-02 12:00:00	Sensor1	PM10	54.58	/8
2019-01-01 12:00:00	Sensor2	O3	47.84	/2
2019-01-01 12:00:00	Sensor2	NO2	43.32	/2
2019-01-01 12:00:00	Sensor2	SO2	46.24	/2
2019-01-01 12:00:00	Sensor2	PM10	45.02	/7
2019-01-02 12:00:00	Sensor2	O3	47.68	/2
2019-01-02 12:00:00	Sensor2	NO2	43.22	/2
2019-01-02 12:00:00	Sensor2	SO2	46.98	/2
2019-01-02 12:00:00	Sensor2	PM10	45.1	/7
2019-01-01 12:00:00	Sensor3	O3	24.97	/1
2019-01-01 12:00:00	Sensor3	NO2	42.05	/2
2019-01-01 12:00:00	Sensor3	SO2	44.54	/2
2019-01-01 12:00:00	Sensor3	PM10	50	/8
2019-01-02 12:00:00	Sensor3	O3	24.61	
2019-01-02 12:00:00	Sensor3	NO2	41.87	
2019-01-02 12:00:00	Sensor3	SO2	46	
2019-01-02 12:00:00	Sensor3	PM10	50.52	
2019-01-07 12:00:00	Sensor4	O3	25.93	/1
2019-01-07 12:00:00	Sensor4	NO2	60.42	/3
2019-01-07 12:00:00	Sensor4	SO2	57.67	/2
2019-01-07 12:00:00	Sensor4	PM10	40.46	/6
2019-01-02 12:00:00	Sensor4	O3	25.44	
2019-01-02 12:00:00	Sensor4	NO2	59.98	
2019-01-02 12:00:00	Sensor4	SO2	58.33	
2019-01-02 12:00:00	Sensor4	PM10	41.25	
2019-01-01 12:00:00	Sensor5	O3	27.92	/1
2019-01-01 12:00:00	Sensor5	NO2	63.56	/3

2019-01-01 12:00:00	Sensor5	SO2	57.46 /2
2019-01-01 12:00:00	Sensor5	PM10	30.72 /5
2019-01-02 12:00:00	Sensor5	O3	26.74
2019-01-02 12:00:00	Sensor5	NO2	64.83
2019-01-02 12:00:00	Sensor5	SO2	59.72
2019-01-02 12:00:00	Sensor5	PM10	32.04

2019-01-01 12:00:00	Sensor6	O3	44.82 /2
2019-01-01 12:00:00	Sensor6	NO2	55.43 /3
2019-01-01 12:00:00	Sensor6	SO2	57.08 /2
2019-01-01 12:00:00	Sensor6	PM10	27.95 /5
2019-01-02 12:00:00	Sensor6	O3	44.62
2019-01-02 12:00:00	Sensor6	NO2	54.8
2019-01-02 12:00:00	Sensor6	SO2	60.79
2019-01-02 12:00:00	Sensor6	PM10	29.51

sensors.csv

SensorID;Latitude;Longitude;

Sensor0	44	-1
Sensor1	44	-0.3
Sensor2	44	0.4
Sensor3	44	1.1
Sensor4	44	1.8

attributes.csv

AttributeID;Unit;Description

O3	µg/m3	concentration d'ozone
SO2	µg/m3	concentration de dioxyde de soufre
NO2	µg/m3	concentration de dioxyde d'azote
PM10	µg/m3	concentration de particules fines

providers.csv

Provider0	Cleaner0
Provider1	Cleaner1

cleaners.csv

Cleaner0	45.333333	1.333333	2019-02-01 12:00:00	2019-03-01 00:00:00
Cleaner1	46.666667	3.666667	2019-02-01 12:00:00	2019-03-01 00:00:00

users.csv

Sensor0	44	-1
Sensor1	44	-0.3
Sensor2	44	0.4
Sensor3	44	1.1
Sensor4	44	1.8
Sensor5	44	2.5

Sensor6 44 3.2