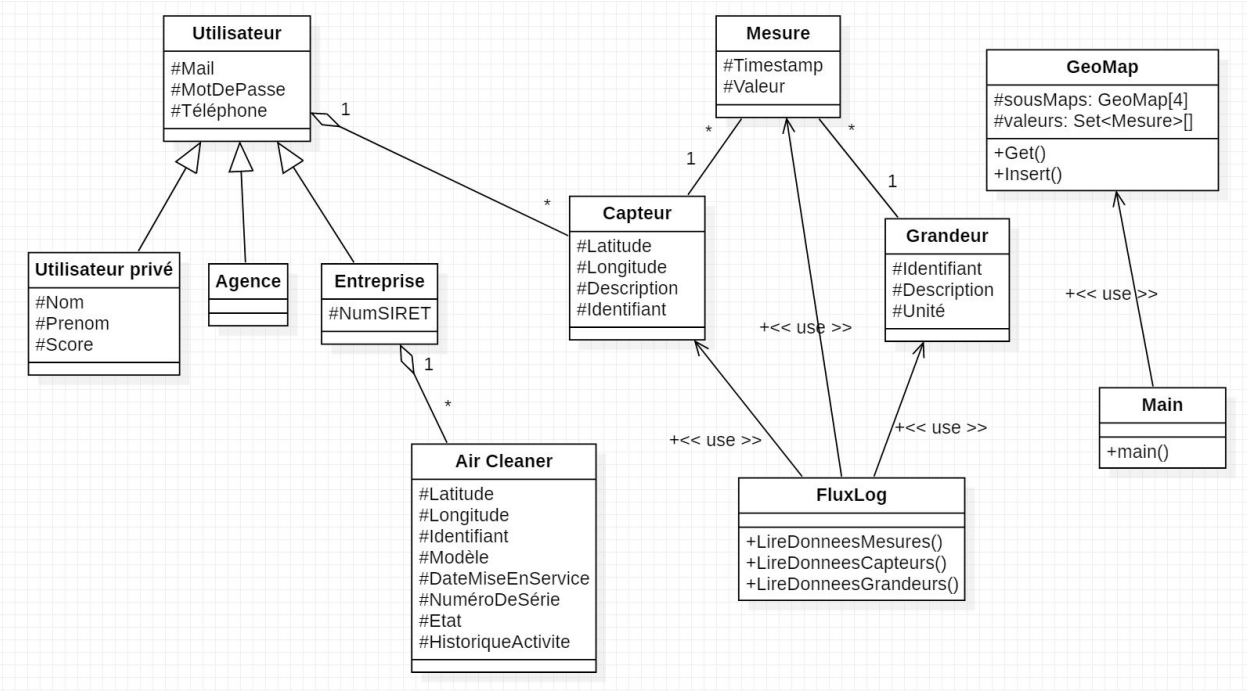
Dossier de design

Sommaire

1. Historique des versions

Version 0.1 : 18.03.20 - Création du document
Version 1.0 : 17.04.20 - Premier draft
Version 1.2 : 07.05.20 - Correction du document suite à la vérification par un enseignant
Version 2.0 : 04.06.20 - Mise en page et relecture

2. Diagramme de classes



3. Choix de l’architecture en couche

Description

Pour réaliser notre projet Air Cleaner nous avons choisi une architecture en couche avec en surface une interface utilisateur dans lequel celui-ci peut notamment insérer des données de mesures prélevés par son capteur ou bien se renseigner sur la qualité de l’air pour une entreprise ou agence.

Cette couche est suivi d’une couche d’authentification et d’autorisation (non-implémentée), qui est suivie à son tour de la couche service qui incorpore les fonctionnalités principales de l’application, à savoir les algorithmes de détection de fausses données ou encore ceux de la synthèse de la qualité de l’air.

Ensuite nous avons la couche entités, qui contient les différents objets manipulés par la couche service comme l'objet Mesure ou encore l'objet Capteur (cf. diagramme de classe).

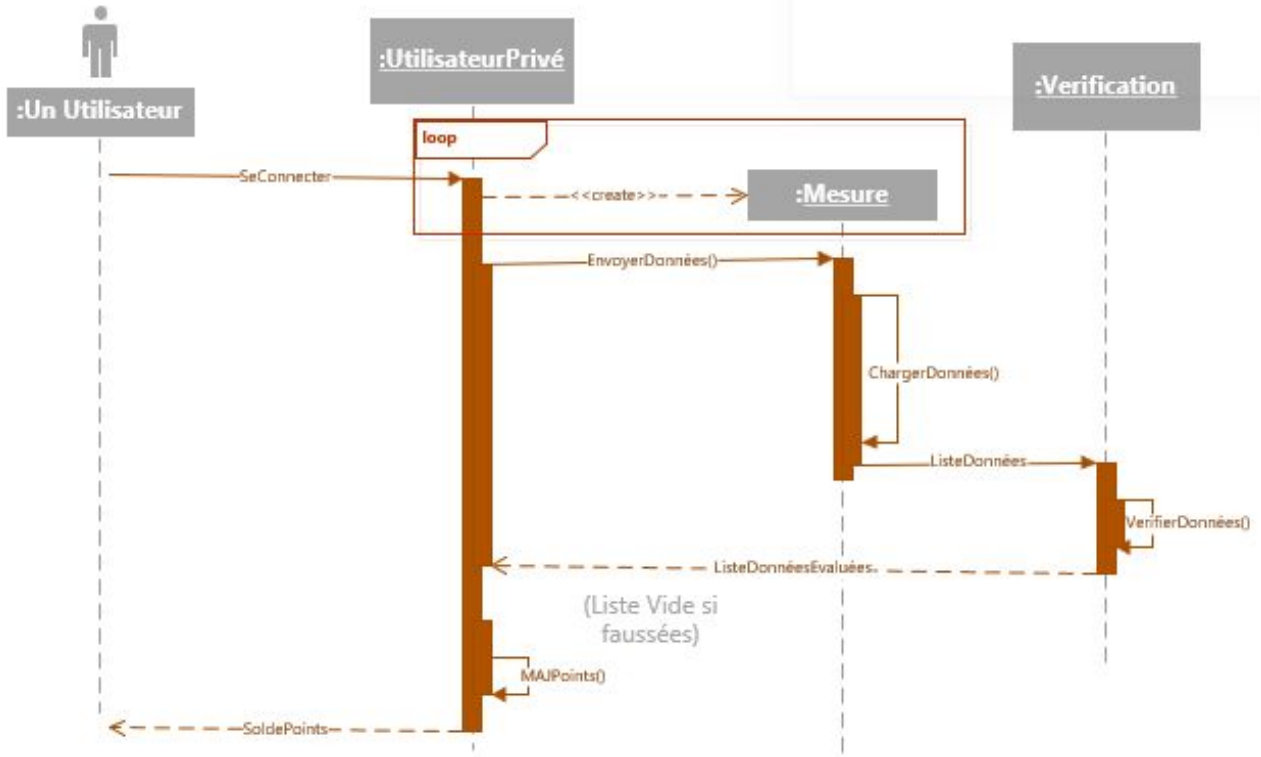
Enfin, nous avons une couche base de données (non-implémentée) qui devra contenir les données relatives aux capteurs, c'est-à-dire les identifiants uniques, localisation… mais aussi l'ensemble des mesures validées par l'algorithme de détection de fausses données et de capteurs similaires. Les données relatives aux authentifications se retrouvent aussi sur cette couche, de même pour les logs de performances.

Justification

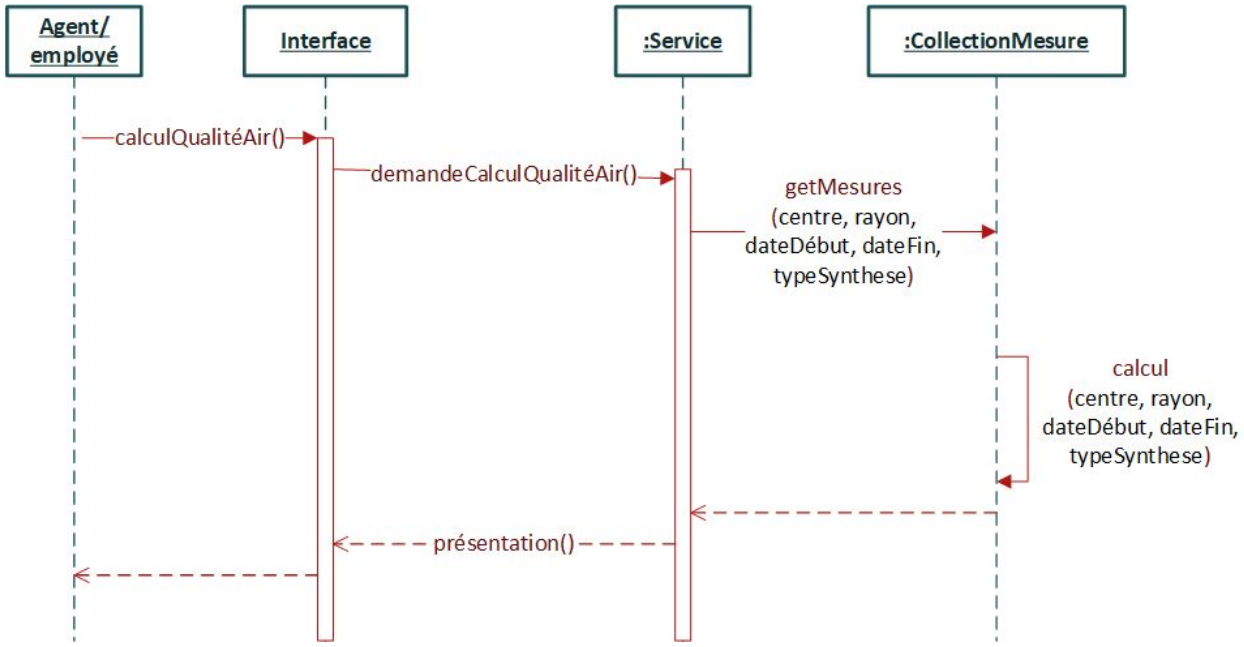
Cette architecture est surtout adapté au besoin des utilisateurs d'interagir à travers une interface maintenue pendant toute la durée de vie de l'application. Elle permet également de rajouter des couches de sécurité supplémentaire si besoin au niveau des autres couches que la couche authentification (par exemple sur la couche Base de données).

4. Diagramme de séquence

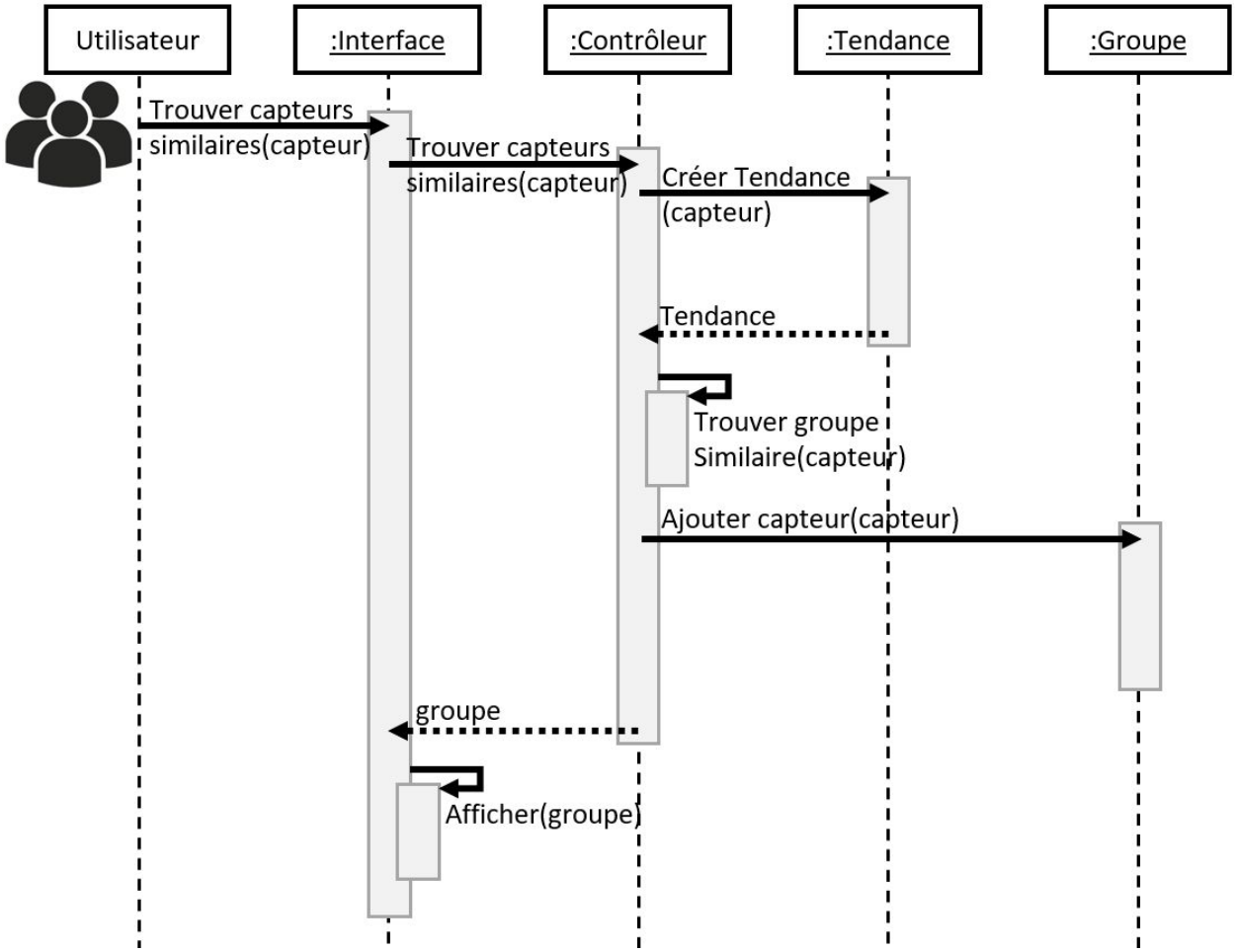
Envoi de données



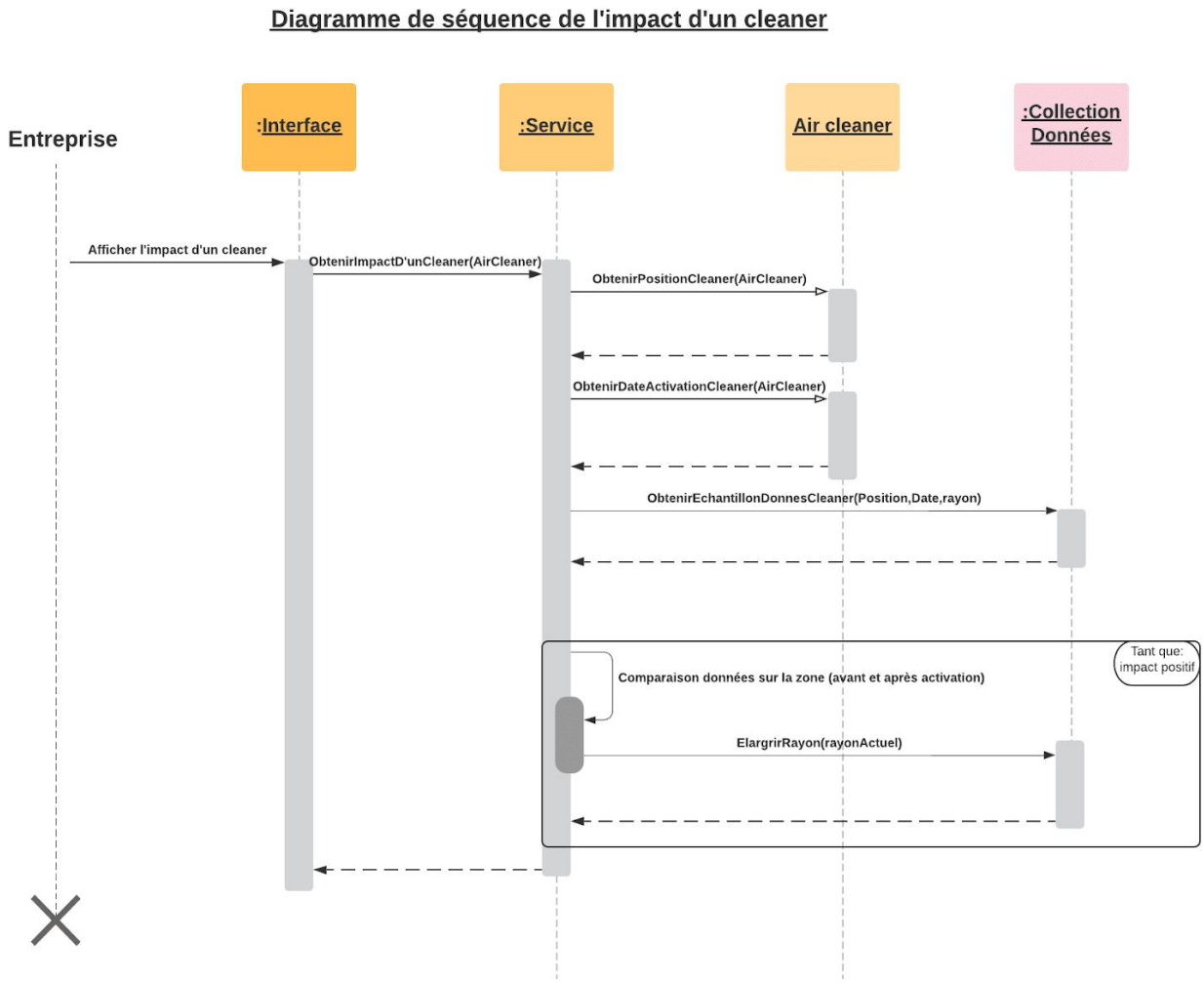
Calcul de la qualité de l'air



Détection de capteurs similaires (gagner des points)

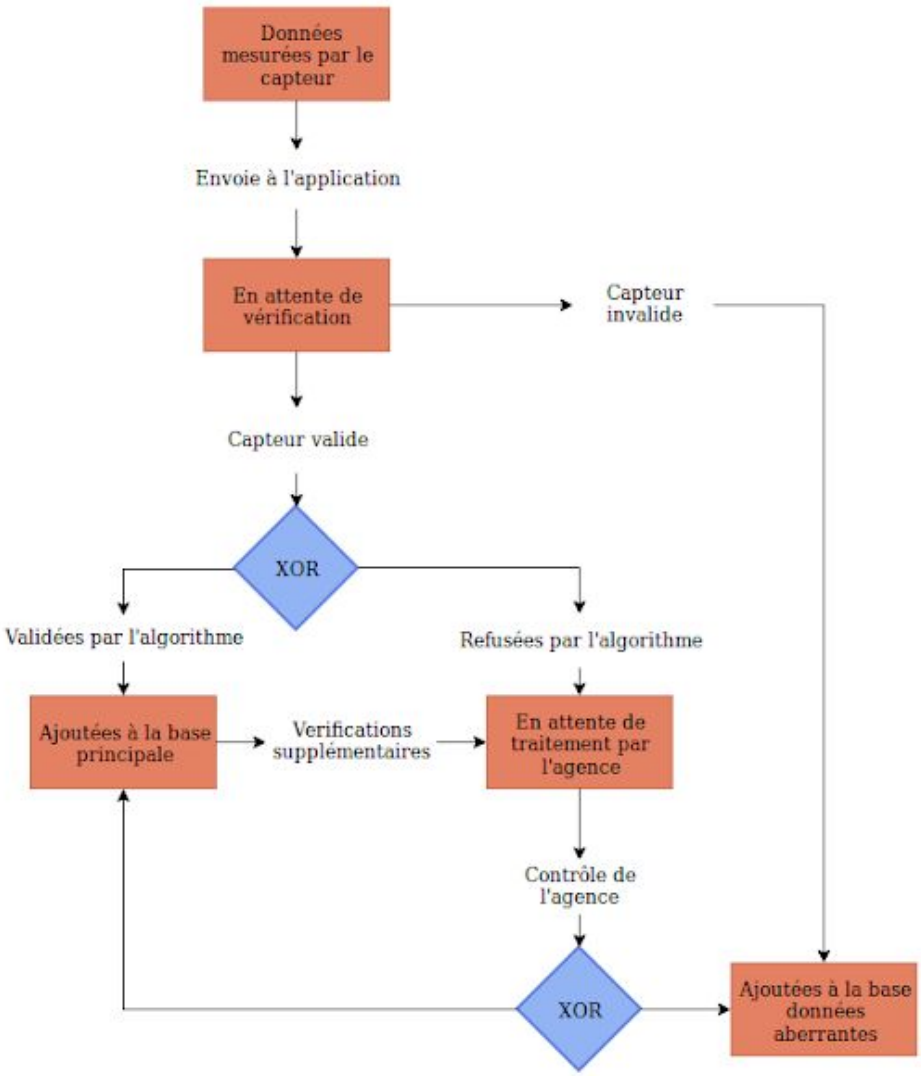


Consulter l'impact des cleaners



5. Diagramme etat-transition

Diagramme état-transition des données

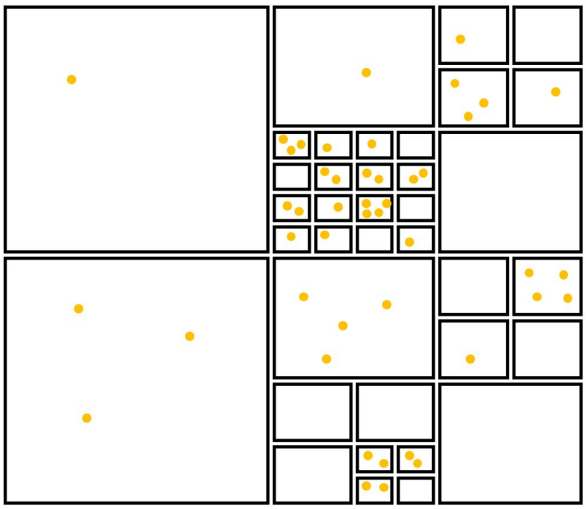


6. Description des algorithmes importants

Indexation des données par coordonnées

Les mesures sont stockés dans une structure de données assez proche d'un arbre quaternaire. A chaque niveau de l'arbre, on a soit un certain nombres de données, soit 4 sous arbres, chacun couvrant un quart de l'espace que le niveau supérieure recouvrait. La division va se

faire lorsque la quantité de données accumulées à un niveau est trop grande. Cette quantité limitée de données par niveau va dépendre d'un paramètre qu'ici nous imaginerons égal à 4. Tant qu'il n'y aura pas plus de 4 coordonnées différentes pour l'ensemble des mesures sur un noeud, alors on pourra y stocker des données. Si on vient à dépasser ce nombre, on ajoute un niveau en dessous de ce noeud, et les données du noeud sont dispatchés dans les arbres du niveau inférieur.
Concrètement, on découpe l'espace en secteurs, et la taille des secteurs va être inversement proportionnelle à la densité des mesures dans l'espace.



Ici, on peut voir un découpage de l'espace, avec 4 coordonnées maximum par niveau. Après découpage.
Cette structure permet de facilement faire de la recherche dans l'espace. Par exemple, si on veut tous les points se trouvant à au plus une distance D d'un centre C, on peut éliminer un plan entier de l'arbre si la projection du point C sur son contour est à une distance supérieure à D du point C. Si on cherche si il existe ou non une mesure à une coordonnée donnée, on peut aller directement au secteur englobant la coordonnée.

Détection de fausses données
Entree : mesure ou liste de mesures à tester

Sortie : liste des valeurs aberrantes ou critiques parmis l'échantillon testé

Construction d'un échantillon témoin de valeurs à partir des valeurs déjà présentes dans l'application, relevées dans des conditions similaires (en fonction de la région et l'intervalle de temps).

Option 1 : Détection des mesures aberrantes (par le test de Chauvenet)

1) Déterminer le nombre de mesures n de l'échantillon témoin
2) Calcul moyenne m de l'échantillon témoin

3) Calcul de l'écart type de l'échantillon témoin
4) Pour chaque mesure de valeur v à tester:
   Test de Chauvenet:
   Si $n^* P (| X - m | \geq | v - m |) < 0.5$
   Alors ajouter la mesure à la liste des mesures aberrantes
5) retourner la liste des valeurs aberrantes

Option 2 : Détection des mesures critiques (mesures plus grande ou plus petites que 95% des mesures témoins)

1) Calcul de la borne bInf au dessus de laquelle se trouve 95% des valeurs de l'échantillon témoin
2) Calcul de la borne bSup au dessous de laquelle se trouve 95% des valeurs de l'échantillon témoin
3) Pour chaque mesure à tester:
   Si la mesure n'appartient pas à l'intervalle [bInf;bSup], ajouter la mesure à la liste des mesures critiques
4) retourner la liste des valeurs critiques

Un nombre anormal de valeurs critiques parmis l'échantillon testé peut entraîner l'exclusion de celui-ci

Synthèse de la qualité de l'air

Entrée : mesures indexées, coordonnées du centre, rayon, date de début, date de fin

Sortie : moyenne, maximum, minimum, ecart-type

Déroulement :

1) Filtre spatio-temporel des mesures
2) Pour chaque mesure :
   a) Si mesure de O3:
      i) moyenneO3 += mesure
      ii) Mise à jour min et max
   b) Si mesure de NO2:
      i) moyenneNO2 += mesure
      ii) Mise à jour min et max
   c) Si mesure de SO2:
      i) moyenneSO2 += mesure
      ii) Mise à jour min et max
   d) Si mesure de PM10:
      i) moyennePM10 += mesure
      ii) Mise à jour min et max
3) Calcule des moyennes par une division par le nombre de mesures pour chaque gaz
4) Pour chaque mesure

    a) Si mesure de O3:
        i) ecartTypeO3 += (valeur-moyenne)²
    b) Si mesure de NO2:
        i) ecartTypeNO2 += (valeur-moyenne)²
    c) Si mesure de SO2:
        i) ecartTypSO2 += (valeur-moyenne)²
    d) Si mesure de PM10:
        i) ecartTypePM10 += (valeur-moyenne)²
5) Calcul des ecarts-types par racines carrés du quotient ecartType/nombreMesure
6) Return moyenne, min, max, ecartType

7. Planning de l'organisation

Répartition des tâches relatives au développement de l'application :

- Eric s'occupera de la couche interface (main) ainsi que d'une partie de la couche service (objet catalogue de mesures GeoMap avec indexation géo-spatiale des données)
- Sylvain se chargera d'implémenter une des deux fonctionnalités principales qui est la détection de fausses données (couche service)
- Iyad s'occupera lui de la lecture des données fournies par les utilisateurs .csv (couche interface) en entrée et des instances d'objets Mesures, Capteurs et Grandeur (couche entités)
- Alexandre se chargera enfin d'implémenter la deuxième fonctionnalité principale qui est la synthèse de la qualité de l'air (couche service)