DOSSIER DE CONCEPTION

Auteurs : AYMARD Florian, CACHARD Sylvain, HUCHOT Pierre, LEONG Loris, NICOLAS Julien

TABLE DES MATIERES

Planning 1
Risque de sécurité propre à l'application : 2
Les risques de l'application : 2
Les mesures prises pour contrer ces risques 2
Diagramme de Cas d'utilisation 3
Diagramme de classe 4
Diagrammes de Séquence 5
Description des algorithmes 6
Algorithme de calcul de la qualité de l'air 6
Identifier des capteurs au fonctionnement similaire (sur une période donnée) 7
Calculer la moyenne des données collectées dans une certaine région géographique à un instant donné/sur une période donnée 7
Identifier des données frauduleuses produites par les contributeurs 8
Trouver l'impact d'un AirCleaner sur la qualité de l'air : 8
Architecture du projet 8
Plans de tests 9

PLANNING

Date	Tâches
30/03/2020	Rédaction du dossier de spécification, notamment des exigences fonctionnelles et non-fonctionnelles de l'application et du manuel d'utilisation Diagramme de cas d'utilisation Première version du diagramme de classe
16/04/2020	Détermination de la logique des différents algorithmes (qualité de l'air, capteurs aux comportements similaires, calcul de la moyenne, identification données frauduleuses, impact d'un AirCleaner) et écriture en pseudo-code
04/05/2020	Première partie du développement de l'application (classes et méthodes principales)
25/05/2020	Seconde partie du développement de l'application (implémentation des différents algorithmes) et tests

03/06/2020	Rendu de la version finale

RISQUE DE SECURITE PROPRE A L'APPLICATION :

LES RISQUES DE L'APPLICATION :

AirWatcher est une application concentrant un grand nombre de données. Celles liées à sa fonctionnalité tout d'abord, comme la qualité de l'air ou la localisation des AirCleaner. Ces dernières sont rassemblées dans une grande base de données uniquement accessible par l'agence gouvernementale. L'utilisation de l'application nécessite également le stockage de données personnelles relatives aux utilisateurs de l'application comme les utilisateurs ou les entreprises : notamment la localisation de leur capteur au cours du temps. Par conséquent, des assaillants pourraient avoir accès à l'ensemble de ces données en récupérant le mot de passe d'un utilisateur. Les dégâts causés pourraient alors être de l'ordre de l'envoi de données frauduleuses, voir de la modification ou la suppression partielle ou totale de la base de données. Les assaillants pouvant aussi avoir accès à la localisation de chacun des utilisateurs et des AirCleaner installés, cela pourrait avoir pour conséquence des dégradations physiques. Ces données pourraient aussi potentiellement être utilisées à des fins commerciales.

LES MESURES PRISES POUR CONTRER CES RISQUES

Afin de contrer aux mieux les risques présentés ci-dessus, on peut envisager un système de vérification de la fiabilité des mots de passe de tous les utilisateurs, mais aussi de systématiquement vérifier chaque donnée envoyée par un utilisateur privé afin de s'assurer de son exactitude. Concernant la base de données, des sauvegardes régulières et à plusieurs endroits peuvent être effectuées afin de permettre la restitution de données. On pourrait aussi imaginer un système sécurisé d'anonymisation des données qui empêcherait un attaquant ayant accès à la base de données d'associer les mesures et les capteurs à un compte utilisateur.

DIAGRAMME DE CAS D'UTILISATION

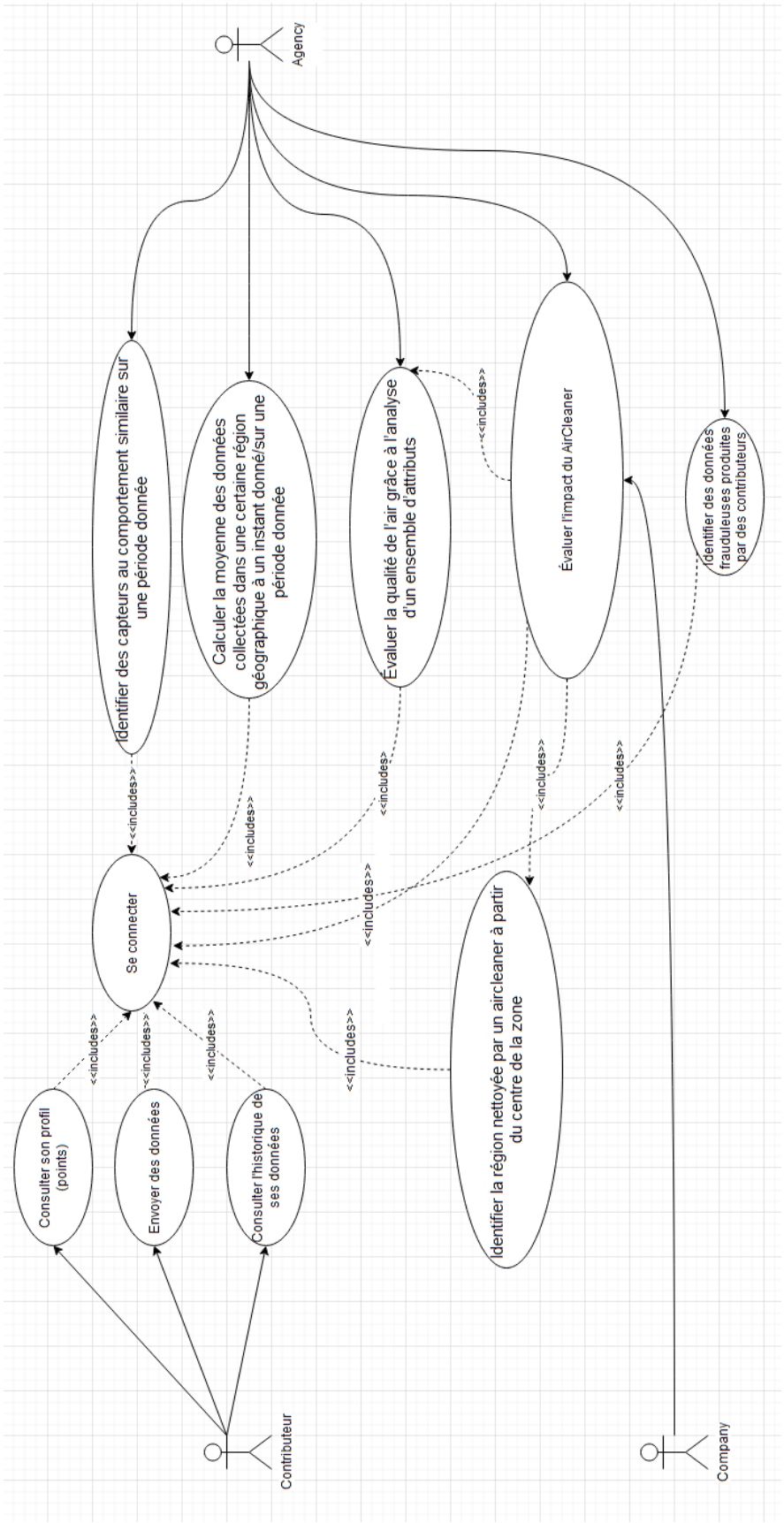
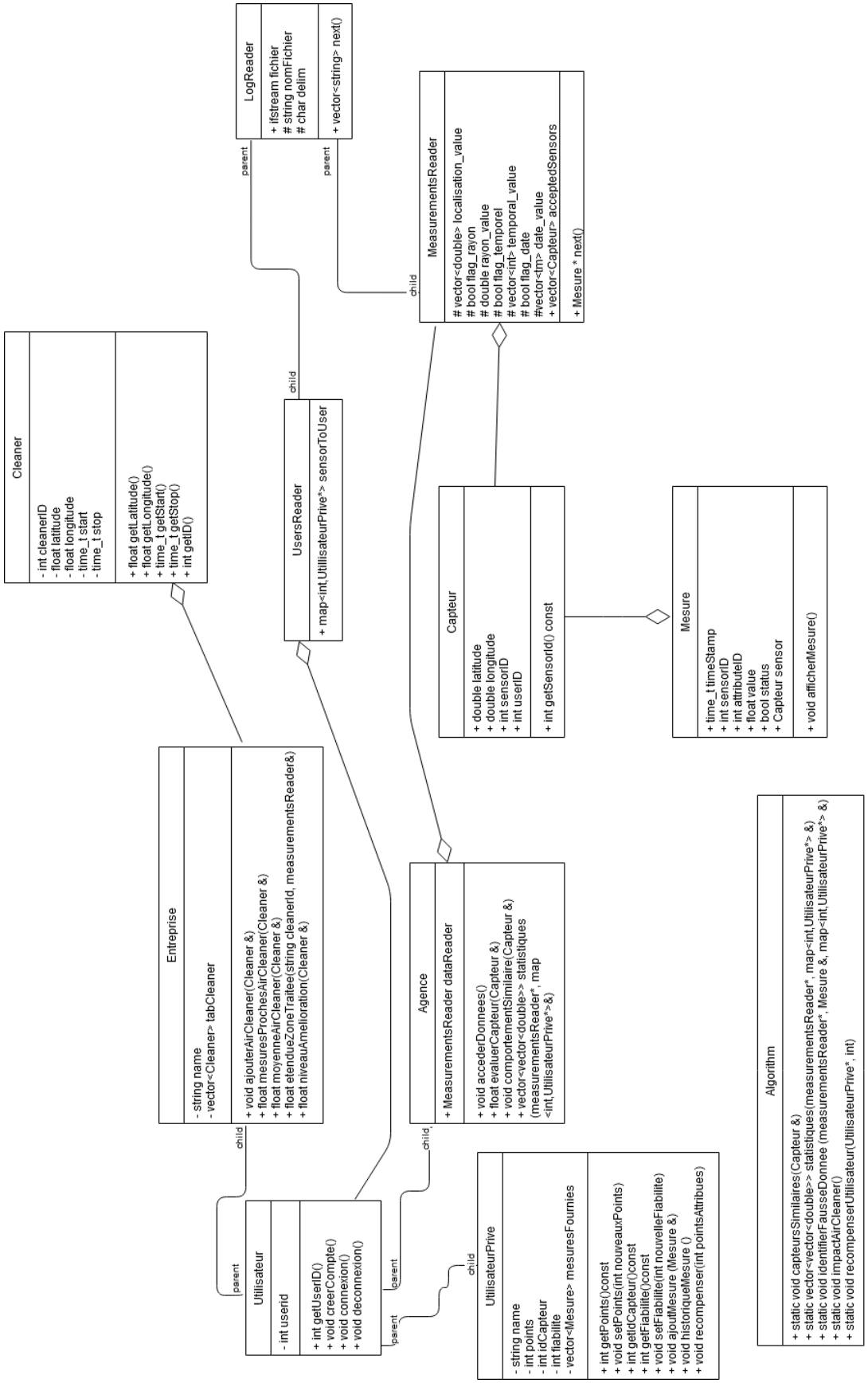


DIAGRAMME DE CLASSE



DIAGRAMMES DE SEQUENCE

Diagramme de séquence pour la recherche du rayon d'impact d'un cleaner donné :

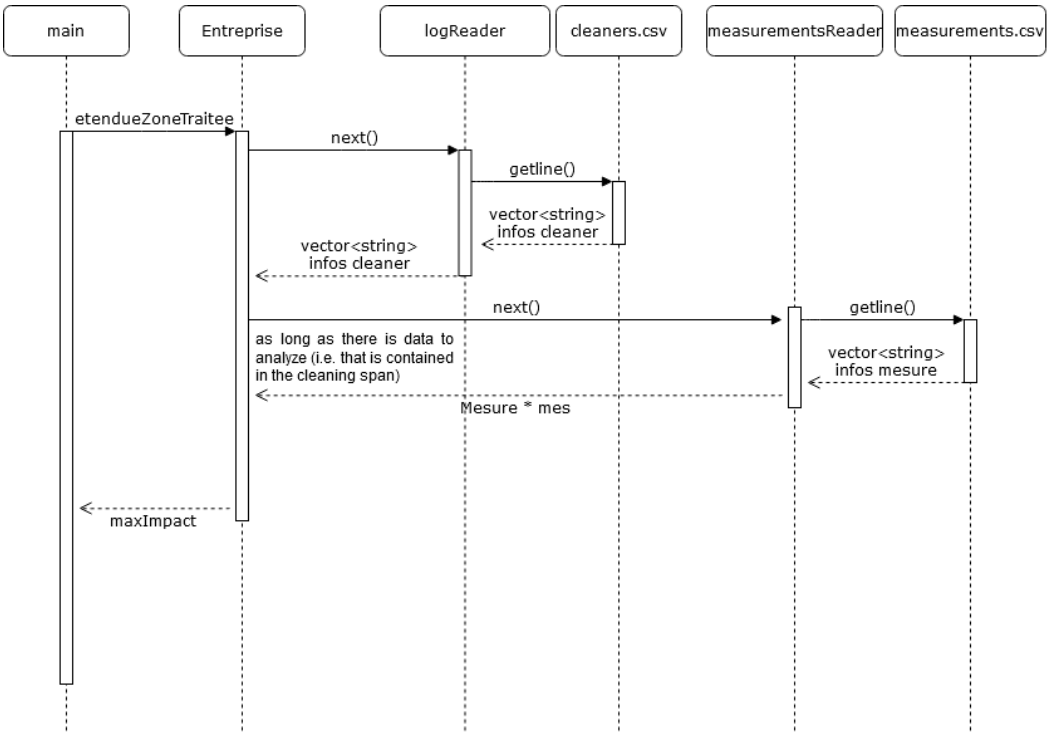
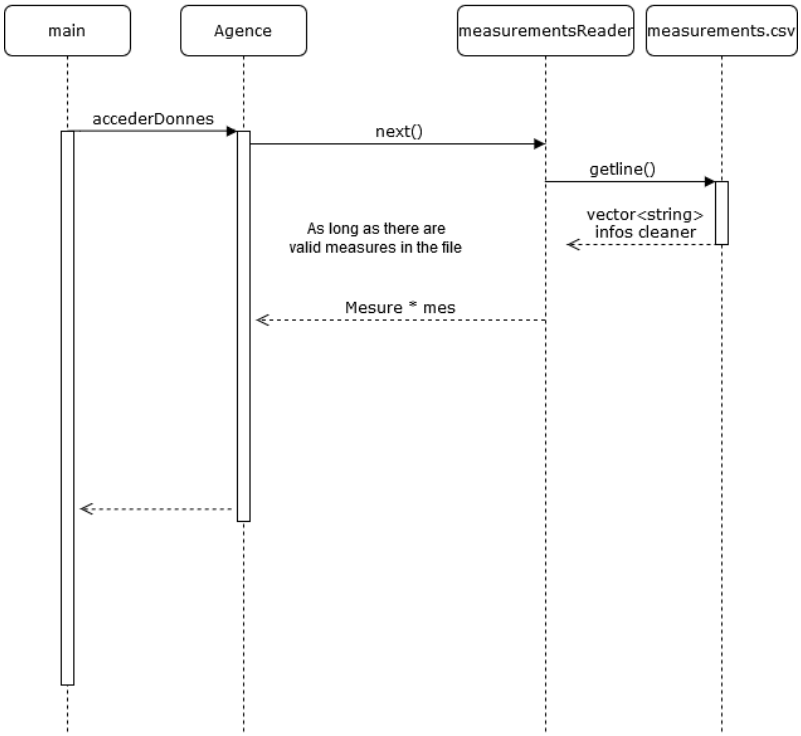


Diagramme de séquence pour l'affichage des données brutes :



DESCRIPTION DES ALGORITHMES

ALGORITHME DE CALCUL DE LA QUALITE DE L'AIR

Pour estimer la qualité de l'air, l'application se base sur la méthode de calcul de l'indice ATMO, qui est un indice journalier. Pour les valeurs SO_2, NO_2 et O_3, il faut récupérer la valeur maximale sur chaque heure, et en faire une moyenne sur la journée. Pour PM_{10}, il suffit uniquement de moyenner avec toutes les valeurs, donc on ne s'attardera pas sur cette partie du calcul. Ensuite, on se base sur la table suivante et le pire indice obtenu sera l'indice ATMO du jour.

Indice ATMO[20, 18]	O_3	SO_2	NO_2	PM_{10}	Niveau
1	0 à 29	0 à 39	0 à 29	0 à 6	Très bon
2	30 54	40 à 79	30 à 54	7 à 13	Très bon
3	55 à 79	80 à 119	55 à 84	14 à 20	Bon
4	80 à 104	120 à 159	85 à 109	21 à 27	Bon
5	105 à 129	160 à 199	110 à 134	28 à 34	Moyen
6	130 à 149	200 à 249	135 à 164	35 à 41	Médiocre
7	150 à 179	250 à 299	165 à 199	42 à 49	Médiocre
8	180 à 209	300 à 399	200 à 274	50 à 64	Mauvais
9	210 à 239	400 à 499	275 à 399	65 à 79	Mauvais
10	≥ 240	≥ 500	≥ 400	≥ 80	Très mauvais

Ainsi, le déroulement de l'algorithme est le suivant : pour chaque tranche d'une heure (00h-01h, 01h-02h,…) on cherche le maximum pour un attribut, puis on calcule la moyenne des valeurs récupérées.

Algorithme en pseudo-code

Entrée :
date_debut, date_fin, latitude, longitude, rayon
Sortie :
moyenne, maximum, minimum, variance *(de l'indice)*
Variables locales :
sum_SO2, sum_NO3, sum_O3, sum_PM,
max_SO2, max_NO3, max_O3,
count_SO2, count_NO3, count_O3, count_PM,
tab_atmo
cur_date, cur_heure
toutes les variables sont initialisées à 0 en début d'algorithme
Algorithme :
Initialisation des variables à 0
Naviguer dans le fichier CSV jusqu'à atteindre la date_debut
cur_date ← ligne.date
cur_heure ← ligne.hour
cur_jour ← 0
NB/TODO : pour des raisons de lisibilité j'omets la partie du traitement de la localisation du capteur, mais évidemment si le capteur de la ligne en cours d'analyse ne rentre pas dans le rayon alors on saute la ligne (voir note de même couleur)
Tant que ligne.date ≤ date_fin **faire**
    Tant que ligne.date == cur_date **faire**

Tant que ligne.hour == cur_heure **faire**
Switch(ligne.attribute)

Cas SO2
Si ligne.value ≥ max_SO2 **faire**
max_SO2 ← ligne.value
Cas NO3
Si ligne.value ≥ max_NO3 **faire**
max_NO3 ← ligne.value
Cas O3
Si ligne.value ≥ max_O3 **faire**
max_O3 ← ligne.value
Cas PM
sum_PM ← sum_PM + ligne.value
count_PM ← count_PM + 1
Fin switch
ligne ← ligne + 1 (le traitement se fait ici : on saute une ligne tant qu'on est pas sur un capteur qui satisfait le critère de localisation)
Fin Tant que
Ajouter les valeurs analysées aux sommes (sum_*) et incrémenter les compteurs (count_*) si les valeurs sont différentes de 0.
cur_heure ← ligne.hour
Fin Tant que
Diviser les sommes (sum_*) par les compteurs correspondants (count_*)
tab_atmo[cur_jour] ← pire indice obtenu grâce aux moyennes calculées au-dessus
cur_date ← ligne.date
cur_jour ← cur_jour + 1
Fin Tant que

IDENTIFIER DES CAPTEURS AU FONCTIONNEMENT SIMILAIRE (SUR UNE PERIODE DONNEE)

On compare les deux séquences temporelles de vecteurs de mesures des deux capteurs grâce au Dynamic Time Warping. Cet algorithme permet d'aligner les deux séquences dans le temps et de calculer une note correspondant à la somme des différences entre ces deux séquences, où à une somme de fonction de ces différences.
Implémentation du DTW : https://github.com/lemire/lbimproved/

CALCULER LA MOYENNE DES DONNEES COLLECTEES DANS UNE CERTAINE REGION GEOGRAPHIQUE A UN INSTANT DONNE/SUR UNE PERIODE DONNEE

- Identification des capteurs se trouvant dans la zone géographique (infos capteurs) autour d'un point, dans un rayon choisi par l'utilisateur
- Ici, on calcule la valeur de la moyenne incrémentalement pour ne pas stocker toutes les valeurs de mesures en mémoire : on ajoute les valeurs des mesures (qui ont été prise dans la zone géographique et la fenêtre temporelle choisie) à une variable sum au fur et à mesure du parcours du fichier

IDENTIFIER DES DONNEES FRAUDULEUSES PRODUITES PAR LES CONTRIBUTEURS

- Identifier les mesures proches en temps (infos mesures, fenêtre d'un jour) et en localisation (infos capteurs, capteurs dans un rayon de 80km de celui à vérifier).
- Calculer la moyenne et l'écart-type de l'ensemble formé par ces mesures.
- Si la valeur de la mesure examinée ne se situe pas dans un intervalle de 95% de confiance autour de la moyenne des valeurs des mesures de la même fenêtre spatiale et temporelle alors l'utilisateur qui a envoyé cette donnée est considéré comme fraudeur (sa note de fiabilité va baisser), et la mesure passe en statut fausse.
- La mesure en question ne rapporte pas de points à l'utilisateur si elle est utilisée.

$$Intervalle\ 95\% = \left[\mu - t_{0,05/2,n}\frac{\sigma}{\sqrt{n}}; \mu + t_{0,05/2,n}\frac{\sigma}{\sqrt{n}}\right]$$

µ= moyenne ; t~=2 ; sigma = écart-type des valeurs proches de la valeur à ajouter ; n : nombre de mesures proches de la valeur à ajouter

TROUVER L'IMPACT D'UN AIRCLEANER SUR LA QUALITE DE L'AIR :

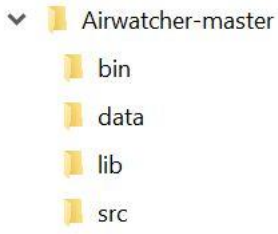
On cherche à calculer le rayon maximal de la zone circulaire autour du AirCleaner dans laquelle tout les capteurs présents ont eu un impact sur la qualité de l'air.
On initialise un rayon d'impact nul.
On initialise un rayon de non impact minimum.

Pour chaque capteur :
On calcule la moyenne des taux mesurés avant le nettoyage dans la zone autour du AirCleaner : m1
On calcule la moyenne des taux mesurés après le nettoyage dans la zone autour du AirCleaner : m2

On calcule le ratio m2/m1 pour chaque attribut, et on considère que le AirCleaner a eu un impact à l'emplacement du AirCleaner si m2/m1 < 0.34

ARCHITECTURE DU PROJET

L'application AirWatcher est organisée selon l'arborescence de fichiers suivantes :



Le répertoire *src* contient l'ensemble des interfaces et réalisations des classes de l'application. Le répertoire *lib* quant à lui contient les deux fichiers statiques utiles pour implémenter les différents algorithmes. *Data* contient les jeux de données de l'application en format *.csv*. Pour finir, le répertoire *bin* contient l'exécutable de l'application. Enfin, à la racine du répertoire *AirWatcher-master* se trouve aussi le *MakeFile* utile pour compiler et exécuter l'application, ainsi qu'une note des programmeurs.

PLANS DE TESTS

Nous avons pris de soin de ne pas subir de régression entre deux versions de l'application. C'est-à-dire que nous testions toutes les fonctionnalités présentes dans l'ancienne version avant de valider la nouvelle version.

Pour les méthodes calculant des valeurs numériques (moyennes, max, min, …) les valeurs étaient confrontés à un tableur Excel ce qui nous a permis de les valider.

Pour les méthodes d'affichage nous comparions l'affichage console obtenu avec ce qu'on attendait sur un petit jeu de données/capteurs (typiquement restreindre à quelques jours et quelques capteurs ou un capteur et une durée plus longue).