

Analyse de la qualité de l'air (Etape 2)

Document de Conception

Génie logiciel et modélisation UML

Aymeric Cousaert, Quentin Ferro, Felix Fonteneau, Mathis Guilhin

10 mai 2019



Résumé

L'application proposée a pour but de pouvoir prendre des décisions afin d'améliorer la qualité de l'air sur le territoire national.

Elle permet aux membres de l'organisme gouvernemental en charge de la surveillance de la qualité de l'air de réaliser un certain nombre de selections sur des données produites par des capteurs répartis sur l'ensemble du territoire national.

Table des matières

1 Diagrammes	2
2 Description des algorithmes importants	9
3 Plan de tests de validation	9
3.1 Calcul de la qualité moyenne de l'air	9
3.2 Trouver les capteurs avec un comportement similaire	10
3.3 Trouver des capteurs caractérisant la qualité de l'air à un point précis	11
3.4 Surveiller le bon fonctionnement des capteurs	12
4 Autres tests	13
5 Planning	13

1 Diagrammes

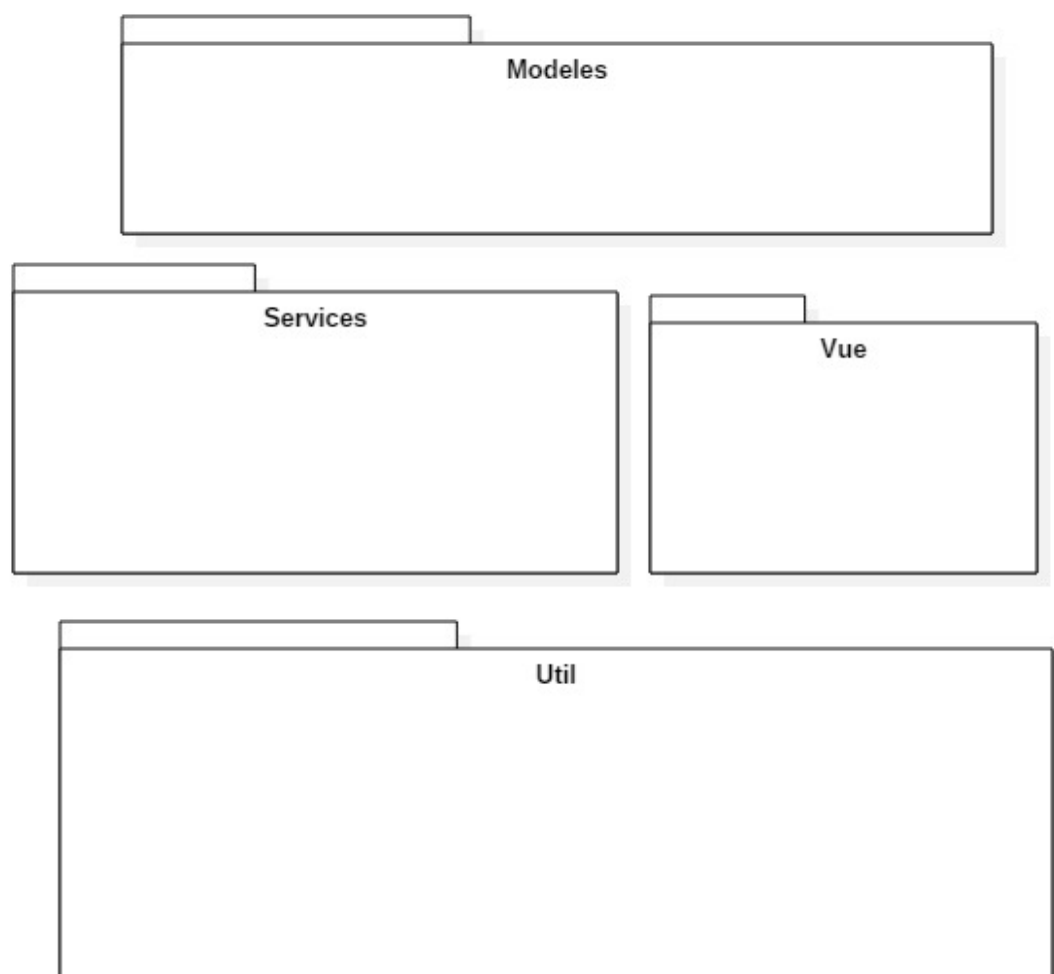


FIGURE 1 – Architecture

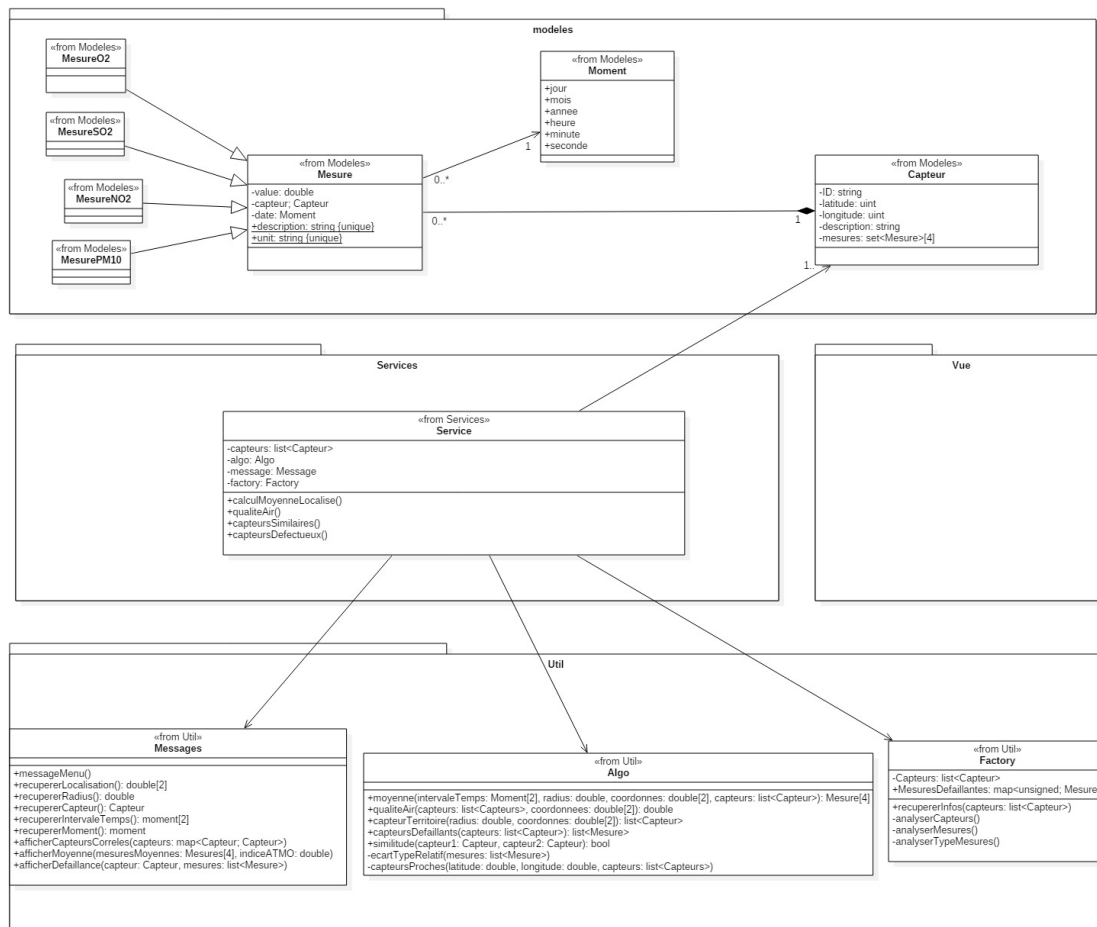


FIGURE 2 – Diagramme de classes complet

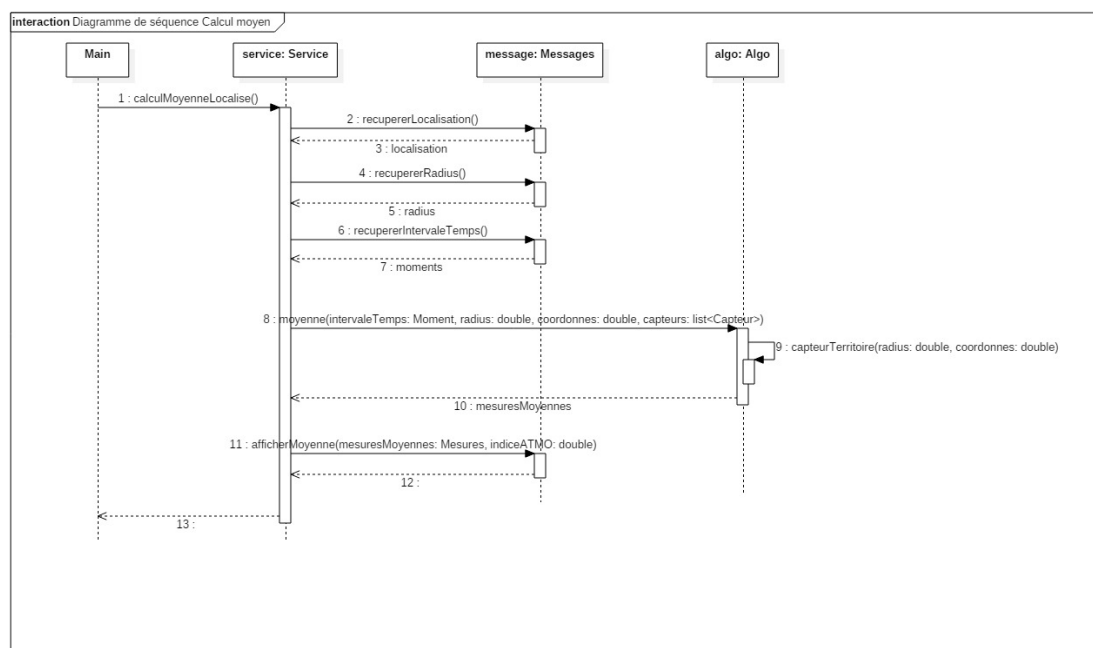


FIGURE 3 – Diagramme de séquence calcul moyen

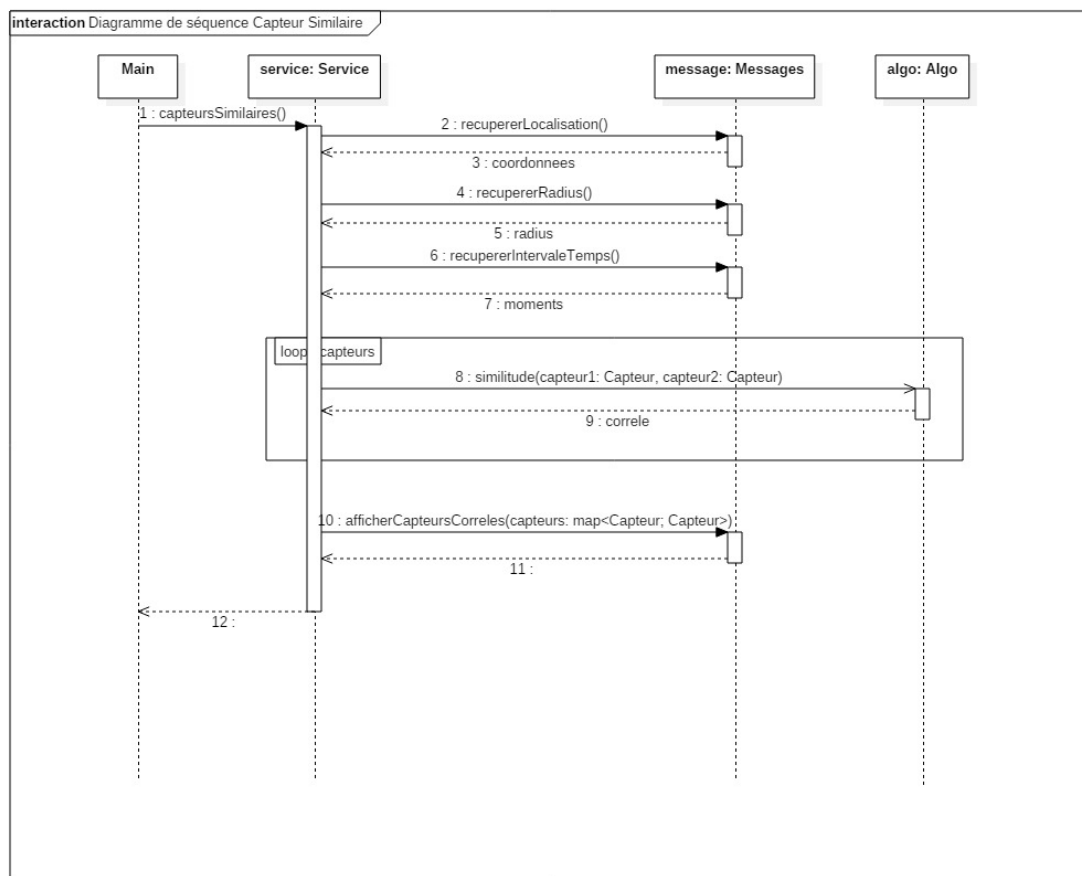


FIGURE 4 – Diagramme de séquence capteurs similaires

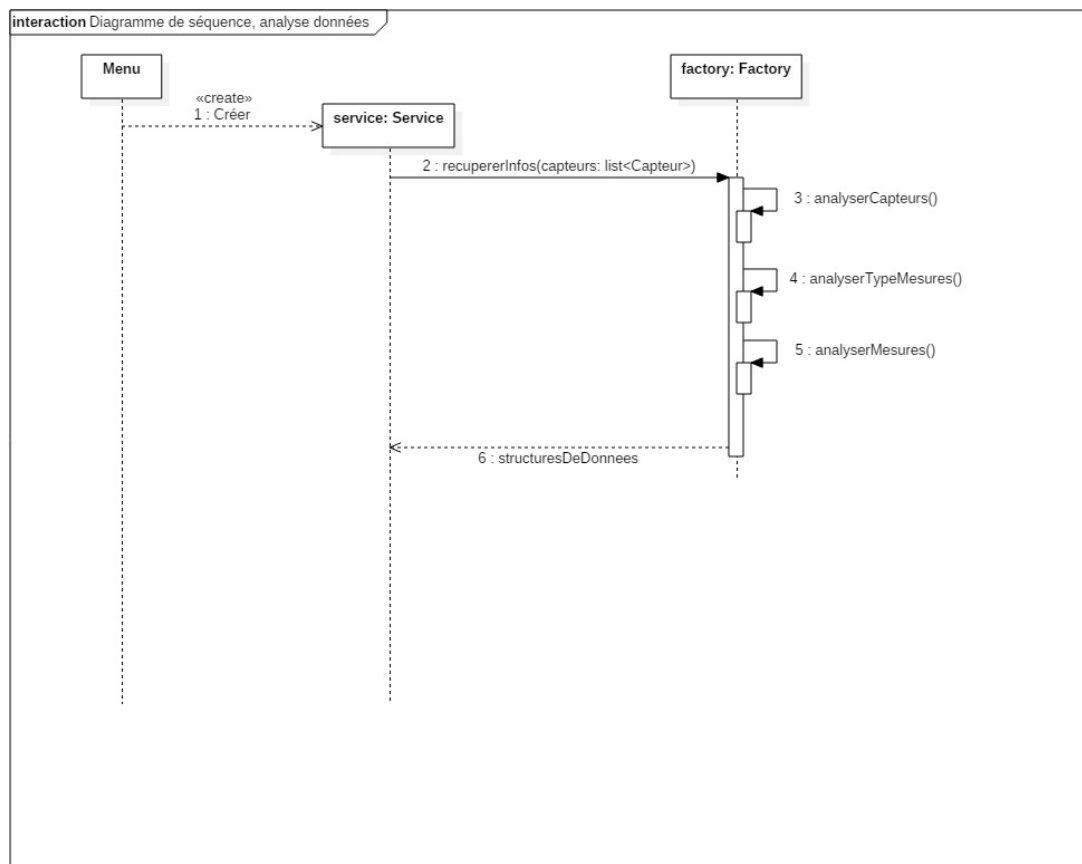


FIGURE 5 – Diagramme de séquence analyse de données

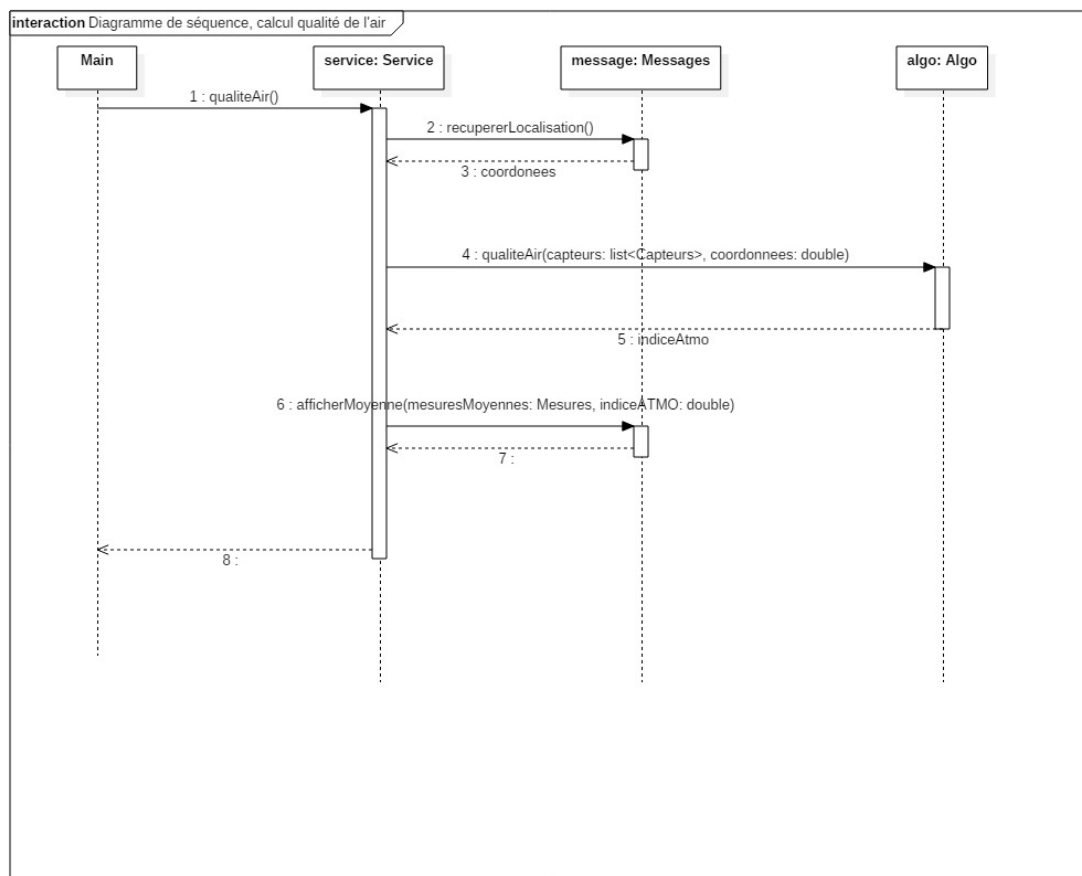


FIGURE 6 – Diagramme de séquence qualité de l'air

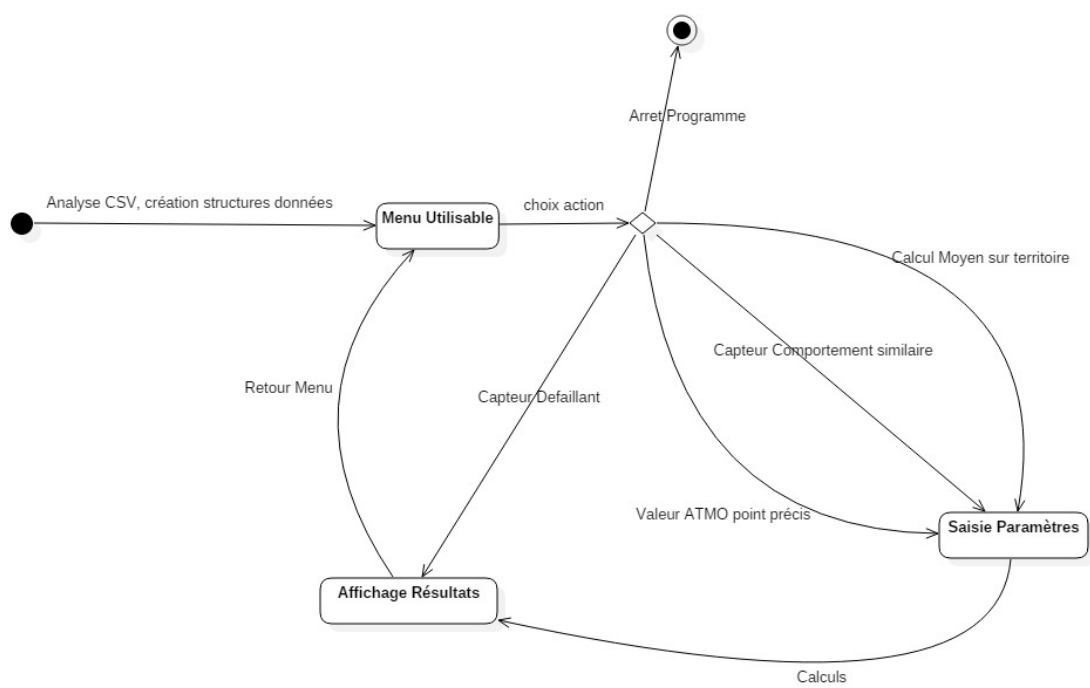


FIGURE 7 – Diagramme états transition

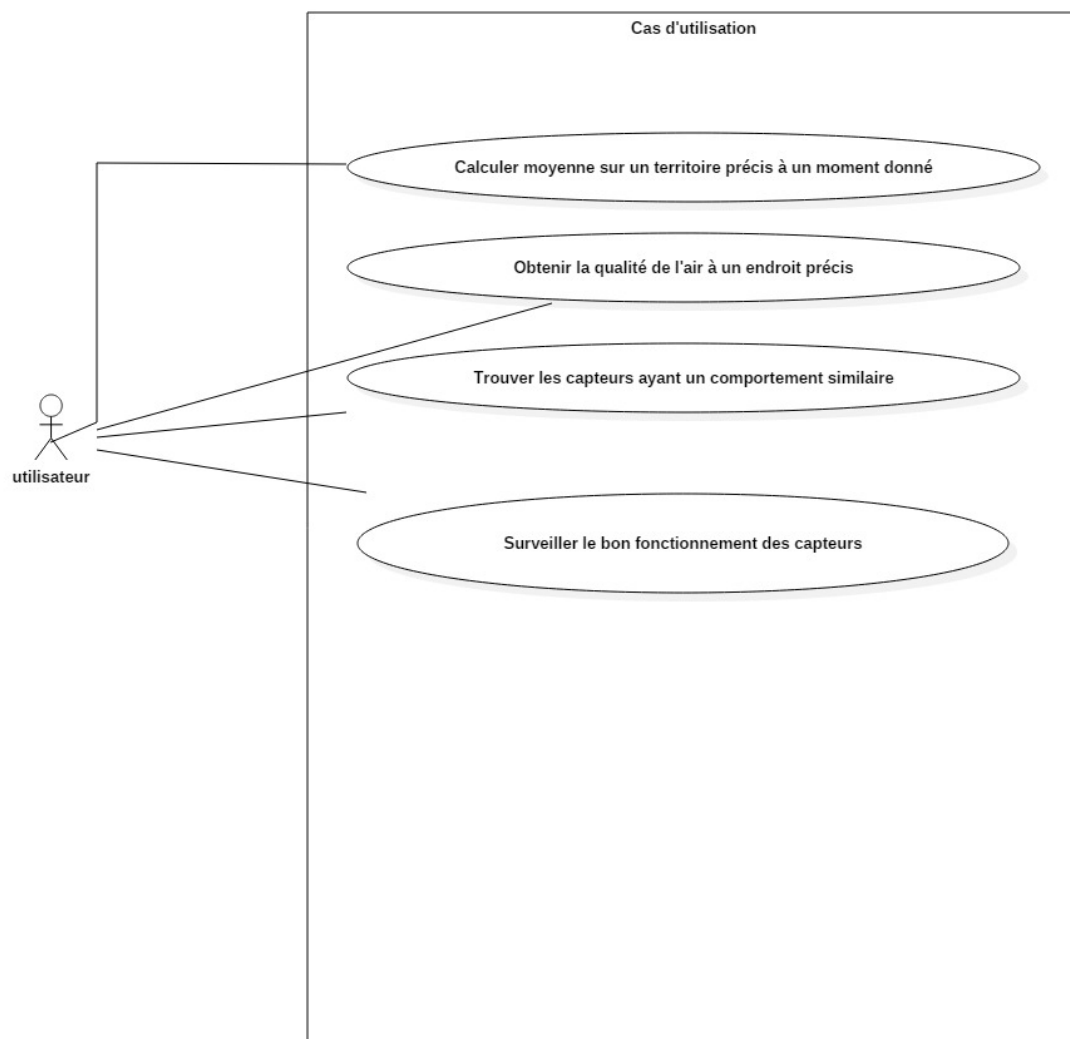


FIGURE 8 – Diagramme des cas d'utilisation

2 Description des algorithmes importants

- Analyse des fichiers logs et des capteurs défectueux :
Pour l'ensemble des lignes des fichiers csv sources
 A chaque ligne, créer les mesure/capteurs correspondants.

- Calcul de la moyenne sur un territoire à un instant/periode :
Récupération des données capteurs correspondantes
Pour chaque type de gaz/particule
 Calculer la moyenne
Qualité moyenne de l'air = max(moyennes)

- Capteur Comportement Similaire :
Récupération des données correspondantes à la zone géographique et tranche horaire
Pour tous les capteurs i de 1 à n
 Pour tous les capteurs j > i
 Pour chaque type de gaz/particule
 Calculer ecart type relatif
 Si tous les écarts type sont inférieurs à 10% des médianes
 Similarité entre i et j

- Caractériser point précis :
Pour tous les capteurs
 Calculer la distance au point géographique
Récupération des données des capteurs les plus proches et de leurs distances
Pour chaque type de gaz/particules
 Calculer la moyenne pondérée en fonction de l'inverse des distances
Qualité de l'air = max(moyennes)

3 Plan de tests de validation

3.1 Calcul de la qualité moyenne de l'air

Cette fonctionnalité fait appel au service calculMoyenneLocalise() qui va demander à l'utilisateur de renseigner à la fois le territoire de la recherche (avec un rayon et des coordonnées GPS) puis soit un moment, soit une période précise. Le logiciel va alors sélectionner les données correspondantes à la recherche, par exemple :

```
2017-01-01T00:01:20.6090000;Sensor0;O3;17.8902017543936;  
2017-01-01T00:01:20.6090000;Sensor0;NO2;42.4807462361763;  
2017-01-04T22:30:21.4120000;Sensor6;SO2;315.994525496237;  
2017-01-04T22:30:21.4120000;Sensor6;PM10;53.0141207251689;  
2017-01-05T16:00:23.5120000;Sensor4;O3;146.233103387634;  
2017-01-05T16:00:23.5120000;Sensor4;NO2;376.1238360793;
```

Sortie :

Les quantités respectives de gaz/particules fines en moyenne dans la zone sont (exprimées en microgrammes par mètre cube):

O3: 82.06,

NO2: 209.3,

SO2: 315.99,

PM10: 53.01,

Indice ATMO : 8 (Mauvais)

Données obtenues par moyenne de 3 capteurs

3.2 Trouver les capteurs avec un comportement similaire

Cette fonctionnalité fait appel au service capteursSimilaires() qui va demander à l'utilisateur de renseigner à la fois le territoire de la recherche (avec un rayon et des coordonnées GPS), puis une période (délimitée par une heure de début et de fin)

Le logiciel va alors sélectionner les données correspondantes à la recherche, et donner les similarités entre capteurs, par exemple :

```
2017-01-01T00:01:20.6090000;Sensor0;O3;17.8902017543936;
2017-01-04T00:01:20.6090000;Sensor1;O3;53.0141207251689;
2017-01-04T00:01:20.6090000;Sensor2;O3;19.0141207251689;
2017-01-04T01:01:20.6090000;Sensor0;O3;19.4554654678965;
2017-01-04T01:01:20.6090000;Sensor1;O3;53.0141207251689;
2017-01-04T01:01:20.6090000;Sensor2;O3;23.0141207251689;
2017-01-01T21:30:21.4120000;Sensor0;NO2;42.4807462361763;
2017-01-04T21:30:21.4120000;Sensor1;NO2;253.0141207251689;
2017-01-04T21:30:21.4120000;Sensor2;NO2;51.0141207251689;
2017-01-04T22:30:21.4120000;Sensor0;NO2;39.1348654654125;
2017-01-04T22:30:21.4120000;Sensor1;NO2;265.0141207251689;
2017-01-04T22:30:21.4120000;Sensor2;NO2;46.0141207251689;
2017-01-01T15:01:20.6090000;Sensor0;PM10;1.51548634564565;
2017-01-04T15:01:20.6090000;Sensor1;PM10;53.0141207251689;
2017-01-04T15:01:20.6090000;Sensor2;PM10;3.5.0141207251689;
2017-01-04T16:01:20.6090000;Sensor0;PM10;2.01454546498454;
2017-01-04T16:01:20.6090000;Sensor1;PM10;52.0141207251689;
2017-01-04T16:01:20.6090000;Sensor2;PM10;4.0141207251689;
```

Le logiciel va alors calculer l'écart type relatif entre les capteurs pour chaque particule :

Entre 0 et 1 :

Pour O3,

$$ET = \frac{\sqrt{(17 - 53)^2 + (19 - 53)^2}}{2} = 25 > 12 = 10\%$$

de la médiane des valeurs

Pour NO2,

$$ET = \frac{\sqrt{(42 - 253)^2 + (39 - 265)^2}}{2} = 154 > 20$$

(pareil)

Pour PM10,

$$ET = \frac{\sqrt{(1.5 - 53)^2 + (2 - 51)^2}}{2} = 36 > 4$$

(pareil)

Il n'y a pas de similarités.

Entre 1 et 2 :

Pour O3,

$$ET = \frac{\sqrt{(53 - 19)^2 + (53 - 23)^2}}{2} = 22 > 12 = 10\%$$

de la médiane des valeurs

Pour NO2,

$$ET = \frac{\sqrt{(253 - 51)^2 + (265 - 46)^2}}{2} = 148 > 20$$

(pareil)

Pour PM10,

$$ET = \frac{\sqrt{(53 - 3.5)^2 + (52 - 4)^2}}{2} = 34 > 4$$

(pareil)

Il n'y a pas de similarités.

Entre 0 et 2 :

Pour O3,

$$ET = \frac{\sqrt{(17 - 19)^2 + (19 - 23)^2}}{2} = 2.23 < 12 = 10\%$$

de la médiane des valeurs

Pour NO2,

$$ET = \frac{\sqrt{(42 - 51)^2 + (39 - 46)^2}}{2} = 5.7 < 20$$

(pareil)

Pour PM10,

$$ET = \frac{\sqrt{(1.5 - 3.5)^2 + (2 - 4)^2}}{2} = 1.41 < 4$$

(pareil)

Il n'y a pas de similarités.

Sortie :

Similarité entre les capteurs 0 et 2, de coordonnées (-8.15,-34.77) et (38.9,-89.2)

3.3 Trouver des capteurs caractérisant la qualité de l'air à un point précis

Cette fonctionnalité fait appel au service qualiteAir() qui va demander à l'utilisateur de renseigner les coordonnées GPS du point voulu. Le logiciel va alors sélectionner les données correspondantes à la recherche, par exemple :

```
2017-01-04T22:30:21.412000;Sensor0;NO2;100.1348654654125;  
2017-01-04T22:30:21.412000;Sensor1;NO2;265.0141207251689;  
2017-01-04T22:30:21.412000;Sensor2;NO2;46.0141207251689;  
2017-01-01T15:01:20.6090000;Sensor0;PM10;1.51548634564565;
```

```

2017-01-04T15:01:20.6090000;Sensor1;PM10;10.0141207251689;
2017-01-04T15:01:20.6090000;Sensor2;PM10;12.0141207251689
2017-01-01T00:01:20.6090000;Sensor0;O3;80.8902017543936;
2017-01-04T00:01:20.6090000;Sensor1;O3;60.0141207251689;
2017-01-04T00:01:20.6090000;Sensor2;O3;72.0141207251689;
2017-01-01T00:01:20.6090000;Sensor0;SO2;200.8902017543936;
2017-01-04T00:01:20.6090000;Sensor1;SO2;169.0141207251689;
2017-01-04T00:01:20.6090000;Sensor2;SO2;241.0141207251689;

```

Une moyenne pondérée va être effectuée pour chaque type de gaz/particules fines, et on donnera alors l'indice ATMO qui correspond au maximum des 4 indices. Imaginons que les sensors 0, 1 et 2 sont à respectivement 0.1, 0.5 et 1 km du point GPS. On prend l'inverse des distances à savoir 10, 2 et 1 comme facteur pour pondérer la moyenne

$$atmoNO2 = \frac{10 * 100 + 2 * 265 + 1 * 46}{10 + 2 + 1} = 121$$

Cela correspond à l'indice 5.

$$atmoPM10 = \frac{10 * 1.51 + 2 * 10 + 1 * 12}{10 + 2 + 1} = 3.62$$

Cela correspond à l'indice 1.

$$atmoO3 = \frac{10 * 80 + 2 * 60 + 1 * 72}{10 + 2 + 1} = 76$$

Cela correspond à l'indice 3.

$$atmoSO2 = \frac{10 * 200 + 2 * 169 + 1 * 241}{10 + 2 + 1} = 193$$

Cela correspond à l'indice 5.

Sortie :

La qualité de l'air au point (-8.451,19.457) est d'indice ATMO 5
(Moyenne réalisée à partir de 3 capteurs).

3.4 Surveiller le bon fonctionnement des capteurs

Cette fonctionnalité fait appel au service `getCapteursDefectueux()` qui va chercher des données capteurs non cohérentes. Par exemple, sur le jeu de données suivants :

```

2017-01-01T00:01:20.6090000;Sensor0;SO2;-12.8902017543936;
2017-01-04T00:01:20.6090000;Sensor1;SO2;169.0141207251689;
2017-01-04T00:01:20.6090000;Sensor2;SO2;100000.0141207251689;

```

Sortie :

Capteur 0 défectueux : valeur négative

Capteur 2 défectueux : valeur maximum dépassée

4 Autres tests

Tests de non-régression

Pour chaque nouvelle étape dans le processus de développement, refaire l'intégralité des tests pour vérifier que les dernières modifications apportées n'ont pas altéré le code qui avait précédemment été validé.

Test de performance

Test pour différents volumes de données dans le fichier csv, tout en vérifiant la spécification non fonctionnelle qui demande un temps de réponse de moins de 5 secondes.

Test d'intégration

Exécution combinée de deux ou plusieurs éléments du programme. Tester par exemple que la recherche de comportement similaire fonctionne avec la recherche de capteurs defectueux.

Test Système

Tester les méthodes qui font un appel à une fonction système. Tester l'ouverture du fichier .csv. Vérifier l'utilisation correcte de la mémoire et du disque. Tester la performance du système.

Test d'utilisabilité

Nécessite l'interaction avec des utilisateurs. Difficile à automatiser. Ici on pourra dans un premier temps se faire passer pour des utilisateurs afin de tester par nous même l'utilisabilité de notre application.

Limites des Tests à éviter

Les tests doivent traiter non seulement le bon fonctionnement de l'application en entrant des valeurs "classiques" (clean tests) mais doivent aussi tester les cas limites ou cas d'erreurs (dirty tests). Il faut prendre en compte les différentes branches, les différents chemins possibles du programme. Choisir des valeurs d'entrée qui ont des implications différentes.

5 Planning

Séance 1 : Réalisation des spécifications ainsi que des principales précisions concernant l'application.

Séance 2 : Conception de l'application et réalisation de tous les diagrammes nécessaires à la bonne compréhension de l'application, complétant ainsi les spécifications réalisées lors de la première séance. Parmi ces diagrammes : diagramme de cas d'utilisation, diagramme d'interaction, diagramme de classe, diagramme de séquence...

Séance 3 : Développement des différentes classes nécessaires au fonctionnement de l'application ainsi que des premières fonctionnalités et tests associés.

Fonctionnalités prévues opérationnelles : L'architecture globale est mise en place (Toutes les classes sont créées et possèdent leurs attributs). En particulier, développement des modèles de l'application, ainsi que le début de la classe factory afin d'analyser les fichiers csv.

Séance 4 : Développement fonctionnalité par fonctionnalité et tests associés. Classe

Fonctionnalités prévues opérationnelles : Liste des capteurs qui ne fonctionnent pas, valeurs de la qualité de l'air à un point précis (grâce à la triangulation des capteurs les plus proches).

L'utilisateur peut connaître la qualité moyenne de l'air dans un cercle qu'il décrit à une période ou instant donné

Séance 5 : Développement fonctionnalité par fonctionnalité et tests associés. Préparation au rendu, vérifier les tests.

Fonctionnalités prévues opérationnelles : liste des capteurs ayant un comportement similaire.
Le produit fini est prêt.