

AIRWATCHER



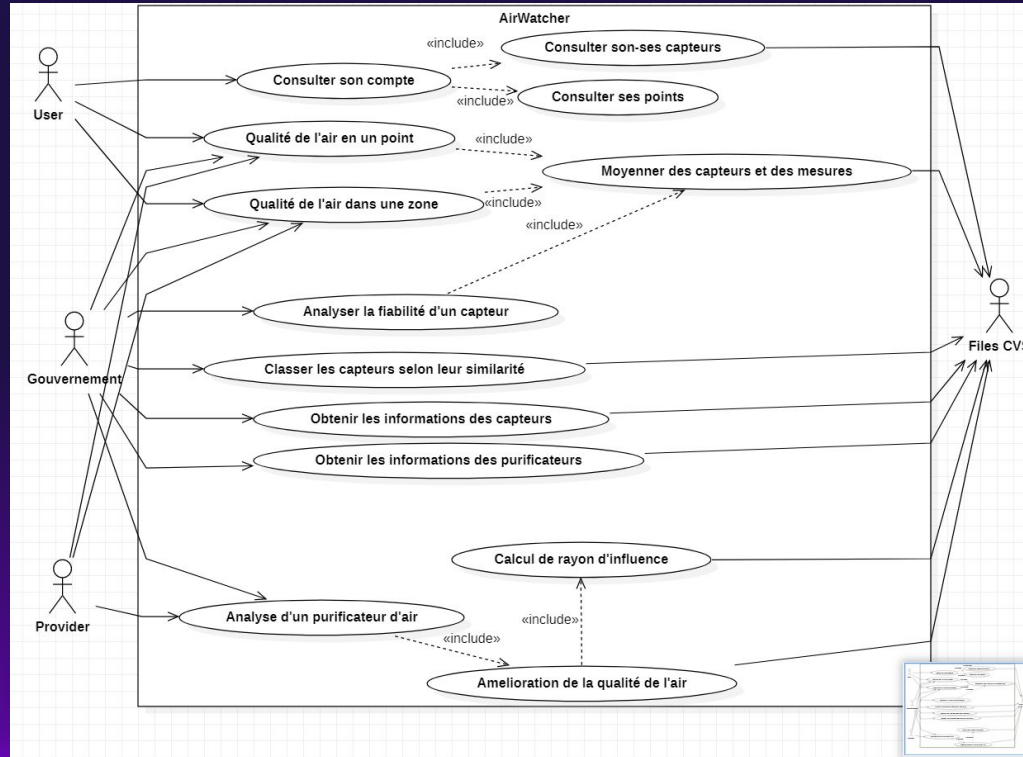
PROJET UML

B3201
ABHAY Annie
NGOV Sophanna

B3209
GOLL Sébastien
HASENFRATZ Louis

TÂCHES À RÉALISER

TROUVER LES BESOINS FONCTIONNELS



ET LES BESOINS NON FONCTIONNELS

Non-functional requirements	Measurements
Performance	Durée d'exécution des l'algorithmes (durée limite:1s pour nos fonctions, 3s pour le chargement des données)
Reliability	Availability (Durée de panne par jour acceptables:10min)
Usability	Nombre de menus faible Nombre de commandes à taper Taille du manuel d'utilisation Clarté des commandes
Robustesse	Nombre de tests échoués Nombre de bugs



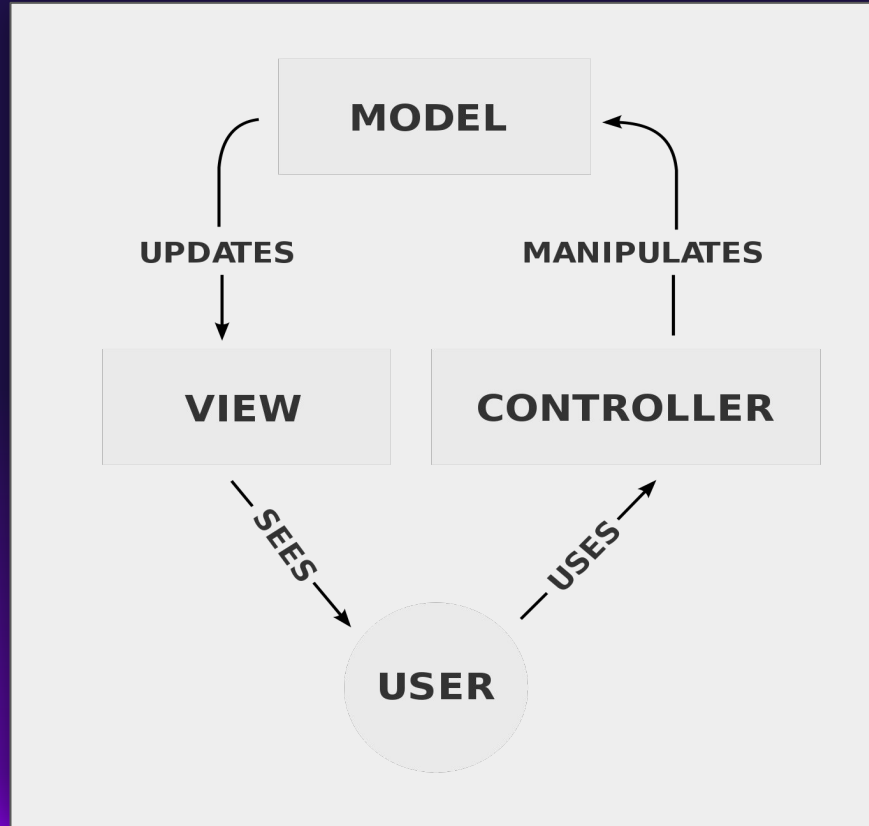
CONNAÎTRE NOS RISQUES

Système	Airwatcher
Atout	<ul style="list-style-type: none">- les coordonnées des capteurs- les informations des purificateurs
Vulnérabilité	<ul style="list-style-type: none">- les comptes du gouvernement et des providers ne sont pas assez sécurisés (mdp trop faible)
Attaque	<ul style="list-style-type: none">- l'attaquant insère du code dans les données attendues- l'attaquant insère des données non valides- l'attaquant vole/devine le mot de passe d'un compte- l'attaquant sature le serveur (DDoS)
Risque	<ul style="list-style-type: none">- l'attaquant obtient les données ; typiquement les coordonnées des capteurs : risque de destruction de matériel, ou encore les coordonnées des purificateurs : risque de sabotage- plus personne ne peut accéder au service

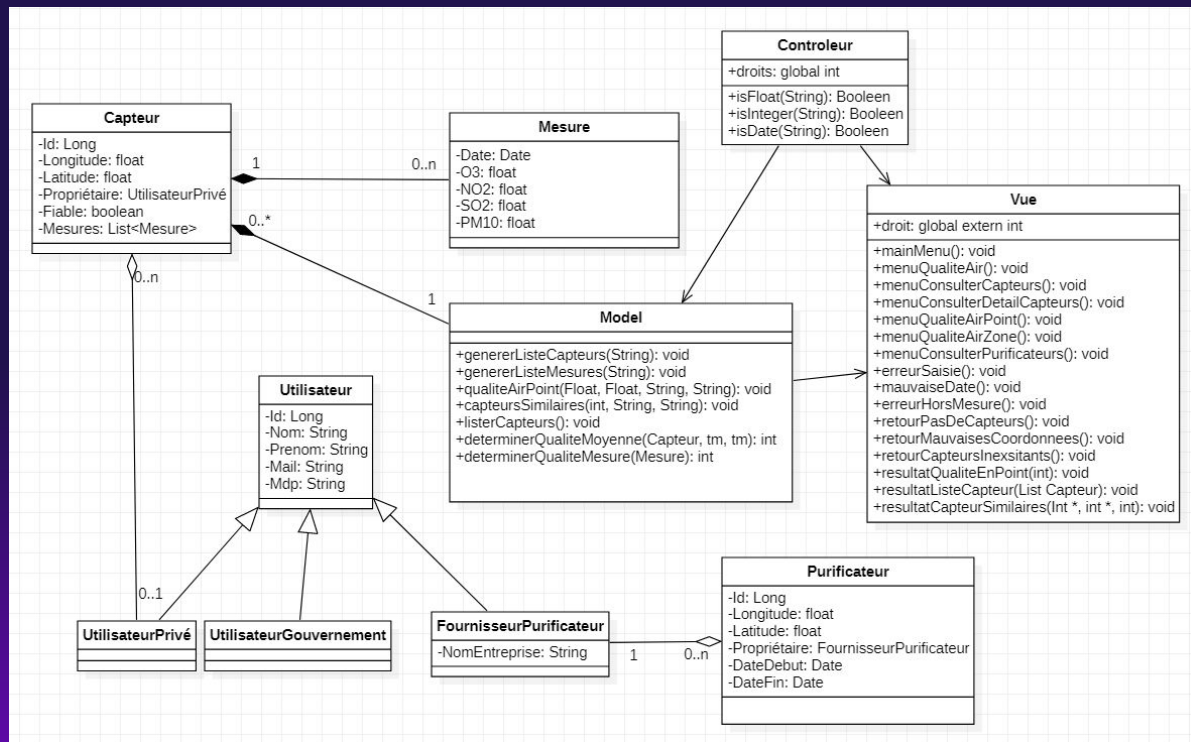


COMMENT S'Y PRENDRE ?

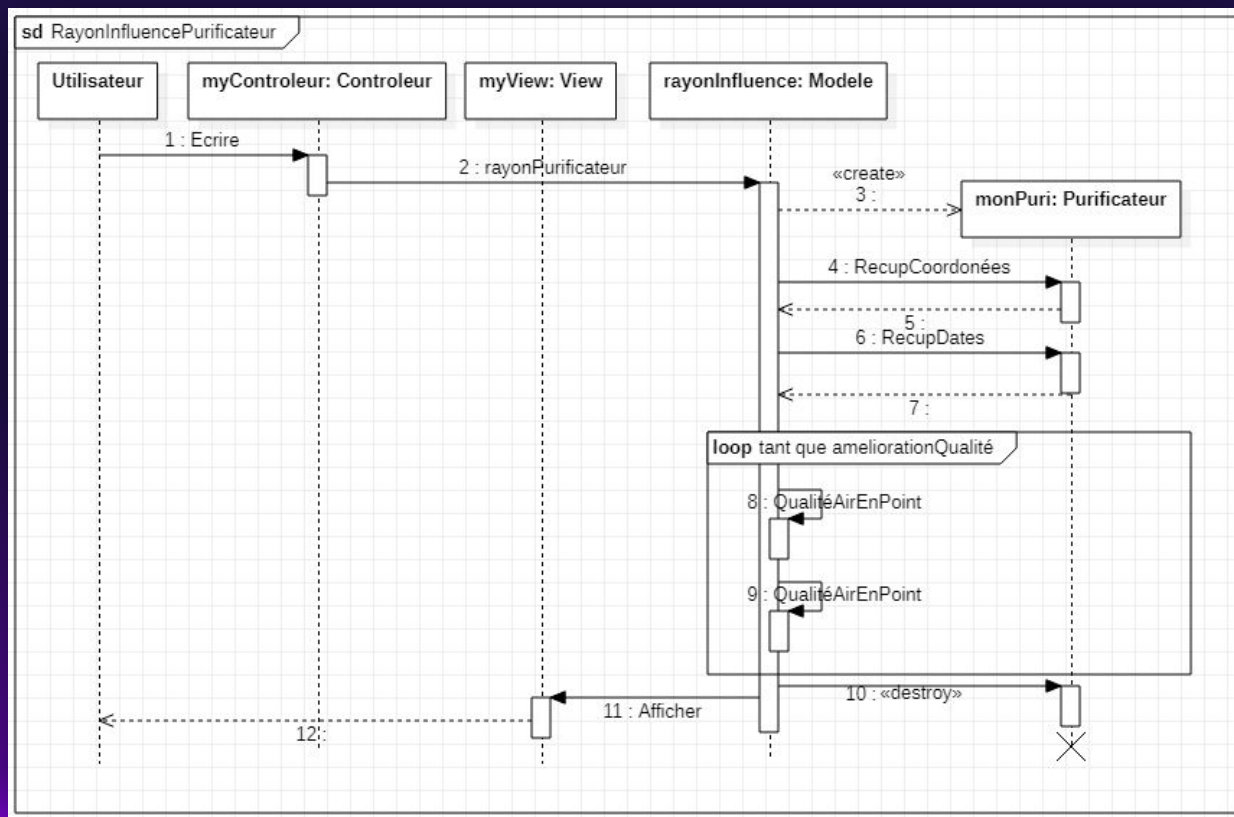
ARCHITECTURE



STRUCTURATION DU CODE



EBAUCHE DES ALGORITHMES



VÉRIFICATION DU FONCTIONNEMENT

CHOIX DES TESTS

Le data set modifier initial comprend 5 capteurs (id de 0 à 4) comportant chacun des données ont donc ces valeurs: (en qualité de l'air)

Jour de la mesure:	01	02	03	04	05
Sensor0:	5	5	5	5	5
Sensor1:	1	1	1	1	1
Sensor2:	1	3	5	7	9
Sensor3:	2	4	6	8	10
Sensor4:	9	7	5	3	1

Test 0:

Un test avec un data set réduit mais sans particularité

Resultat attendu(score en fonction des capteurs) en partant du capteur 2:

Sensor0:	12
Sensor1:	1
Sensor2:	0
Sensor3:	5
Sensor4:	24

Test 1:

Un test où toutes les mesures sont identiques à celles du capteur 2:

Resultat attendu(score en fonction des capteurs) en partant du capteur 2:

Sensor0:	0
Sensor1:	0
Sensor2:	0
Sensor3:	0
Sensor4:	0

Test 2:

Un test où un seul capteur est disponible dans la base de données

```
void test0();  
void test1();  
void test2();  
void test3();  
void test4();  
void test5();  
void test6();  
void testPerformance();
```

```
int main(){  
  
    test0();  
    test1();  
    test2();  
    test3();  
    test4();  
    test5();  
    test6();  
    testPerformance();  
  
    return 0;
```

```
}
```



CONCLUSION