

# **Document d'initialisation**

## **1. Introduction:**

L'application AirWatcher est conçue pour faciliter la surveillance et l'analyse des données sur la qualité de l'air recueillies par des capteurs déployés sur un vaste territoire. Développé pour un organisme gouvernemental de protection de l'environnement, AirWatcher vise à assurer la fonctionnalité des capteurs, à regrouper les données pour l'analyse statistique, à identifier les zones présentant une qualité de l'air similaire et à fournir des informations sur l'impact des purificateurs d'air sur la qualité de l'air. Le présent document décrit les premières étapes et la planification nécessaires au développement et à la mise en œuvre de l'application.

## **2. Planification :**

### **a. Composition de l'équipe :**

L'équipe de développement se compose de quatre membres : Sami TAIDER, Samuel LOUVET DE MONTELLA, Jixiang SUN, Adam ALLAZ.

### **b. Rôles et responsabilités :**

**Samuel** : Chef de projet, dirige l'équipe, s'assure que les jalons du projet sont atteints et supervise la qualité globale du projet.

**Jixiang** : Responsable de documenter tous les aspects du projet, y compris les défis rencontrés, les solutions mises en œuvre et les progrès réalisés. Assure une documentation complète pour la présentation finale.

**Sami** : évalue rigoureusement les fonctionnalités de l'application, en s'assurant qu'elles répondent aux normes de qualité grâce à des tests approfondis, à l'identification des bogues et à la collaboration avec l'équipe pour les correctifs, garantissant ainsi la stabilité et la fiabilité du rendu final.

**Adam** : responsable de la soutenance finale ainsi que la mise en forme des documents écrits. Responsable de la création de diapositives et assure une présentation fluide et engageante.

Les tâches communes à tous les membres de l'équipe incluent la modélisation, le choix des fonctionnalités, l'écriture du code.

### **c. Organisation :**

**TP 1 :**

Au cours de cette session, notre objectif est de rassembler les exigences, de définir les fonctionnalités du système et d'analyser les cas d'utilisation de l'application. Nous examinerons les exigences du projet, discuterons des besoins fonctionnels et non fonctionnels et créerons un diagramme de cas d'utilisation pour illustrer les interactions du système et les rôles des utilisateurs. De plus, nous analyserons les risques de sécurité afin d'identifier les menaces et les contre-mesures, en veillant à une compréhension complète de la portée du projet et des exigences des utilisateurs.

## **TP 2 :**

Dans cette session, nous nous concentrons sur la conception de l'architecture de l'application et la planification de ses éléments structurels. Nous développerons des diagrammes de classe pour représenter la structure du système, des diagrammes de séquence pour les scénarios majeurs, et décrirons les algorithmes majeurs avec des pseudo-codes. À la fin, nous visons à avoir un plan clair pour guider nos efforts de développement.

## **TP 3 :**

Cette session se concentre sur la mise en œuvre du code source, la réalisation de tests unitaires et l'assurance de la fonctionnalité. Nous développerons le code source pour les fonctionnalités clés, créerons des tests unitaires pour vérifier l'exactitude et validerons le code source par des tests. L'objectif est de disposer de fonctionnalités robustes et testées prêtes à être intégrées à la fin de la session.

## **TP 4 :**

Au cours de cette séance, nous nous concentrerons sur la finalisation des documents de présentation, la réalisation d'un examen final de la documentation et la préparation de la présentation finale. Nous polissons les diapositives de présentation, répétons la démonstration en direct et effectuons les améliorations nécessaires pour assurer une soutenance fluide. De plus, nous finaliserons toute la documentation et les préparatifs restants pour le rendu.

## **TP 5 :**

Au cours de cette dernière séance, nous nous concentrerons sur les ajustements de dernière minute, sur la vérification de tous les produits livrables et sur la vérification que tout est en ordre pour la présentation.

# **Exigences fonctionnelles et non-fonctionnelles du système**

## **Exigences fonctionnelles :**

**Fonction** : Calculer les Moyennes des Mesures des Capteurs par Zone

### **Description :**

Cette fonction calcule la valeur moyenne mesurée par tous les capteurs dans chaque zone géographique pour chaque type de données, en utilisant les informations provenant de deux fichiers CSV : l'un contenant les descriptions des zones géographiques avec les plages de latitude et de longitude ainsi que leurs identifiants de zone correspondants, et un autre détaillant les identifiants des capteurs et leurs identifiants de zone respectifs. De plus, les mesures des capteurs sont obtenues à partir du fichier "measurements.csv".

### **Entrées :**

Aucun paramètre

### **Source :**

- Descriptions des zones géographiques avec les plages de latitude et de longitude (à partir du fichier CSV).
- Identifiants des capteurs et leurs identifiants de zone correspondants (à partir du fichier CSV).
- Mesures des capteurs à partir du fichier "measurements.csv".

### **Sorties :**

Liste des moyennes des mesures des capteurs par zone pour chaque type de données dans le format d'une map qui a pour clés le type de donnée concernée et pour valeur une liste contenant la moyennes des mesures pour chaque zone géographique dans l'ordre des identifiants de zone ("type de donnée" : [moyenneZone0, moyenneZone1 .....])

### **Destination :**

Fonction de contrôle des capteurs défaillants

### **Action :**

Pour chaque zone géographique :

Calculer la valeur moyenne pour chaque type de données mesurées par les capteurs dans la zone.

**Exigences :**

- Disponibilité des descriptions des zones géographiques avec les plages de latitude et de longitude, chaque zone contient un nombre de capteurs suffisant (pas de capteurs isolés).
- Mappage correct des identifiants des capteurs à leurs identifiants de zone respectifs.
- Données de mesures des capteurs valides et complètes dans le fichier "measurements.csv".

**Pré-condition :**

- Les fichiers CSV contenant les descriptions des zones et les informations sur les capteurs sont accessibles et à jour.
- Le fichier "measurements.csv" contient des données de mesures de capteurs valides.

**Post-condition :**

Les moyennes des mesures des capteurs par zone pour chaque type de données sont calculées et stockées dans la map de sortie.

**Effets secondaires :**

Aucun.

**Fonction : Fonction de contrôle des capteurs défaillants :**

**Description :**

Cette fonction utilise la fonction précédemment implémentée pour calculer les moyennes des mesures des capteurs par zone afin de déterminer l'état de chaque capteur. Elle génère un fichier CSV listant tous les capteurs avec leur état correspondant, indiquant s'ils fonctionnent correctement ou non (vrai ou faux).

**Entrées :**

Aucun paramètre

**Source :**

- Identifiants des capteurs et leurs identifiants de zone correspondants (à partir du fichier CSV).

- Mesures des capteurs à partir du fichier "measurements.csv".

**Sorties :**

Fichier CSV contenant une liste des capteurs et de leur état.

**Destination :**

Main loop

**Action :**

1. Récupérer la liste des capteurs et de leurs identifiants de zone respectifs.
2. Pour chaque capteur :
  - a. Déterminer l'identifiant de zone du capteur.
  - b. Obtenir les moyennes des mesures des capteurs pour la zone correspondante.
  - c. Analyser les mesures du capteur pour déterminer son état de santé :
    - i. Si les mesures du capteur se situent dans les plages attendues basées sur les mesures moyennes pour sa zone, le marquer comme fonctionnant correctement (vrai).
    - ii. Sinon, marquer le capteur comme défectueux (faux).
3. Générer un fichier CSV contenant la liste des capteurs avec leur état de santé (vrai ou faux).

**Exigences :**

- Disponibilité des moyennes des mesures des capteurs par zone pour chaque type de données.
- Liste des capteurs et de leurs identifiants de zone respectifs.

**Pré-condition :**

Bon fonctionnement de la fonction précédemment implémentée pour calculer les moyennes des mesures des capteurs par zone.

**Post-condition :**

Un fichier CSV est généré et contient une liste des capteurs avec leur état de santé.

**Effets secondaires :**

Aucun.

**Fonction : Calculer la Moyenne de la Qualité de l'Air dans une Zone Circulaire :**

**Description :**

Cette fonction agrège les données de qualité de l'air collectées pour calculer des statistiques qui aident l'agence à prendre des décisions pour améliorer la qualité de l'air. Plus précisément, l'application permet aux utilisateurs de calculer la moyenne de la qualité de l'air dans une zone circulaire spécifiée par l'utilisateur. La moyenne de la qualité de l'air peut être calculée pour un moment donné ou une période de temps spécifiée.

**Entrées :**

- Centre et rayon de la zone circulaire spécifiée
- Indication de si on veut calculer la moyenne pour un instant donné ou sur un intervalle de temps
- Période de temps pour laquelle on veut calculer la moyenne le cas échéant

**Source :**

- Liste des capteurs et leur localisation à partir du fichier "sensors.csv".
- Mesures des capteurs à partir du fichier "measurements.csv".

**Sorties :**

Valeur moyenne de la qualité de l'air pour la zone circulaire spécifiée.

**Destination :**

Main loop

**Action :**

1. Récupérer les données de qualité de l'air des capteurs en fonction des coordonnées de localisation et du rayon fournis par l'utilisateur.
2. Calculer la valeur moyenne de la qualité de l'air pour les données filtrées.
3. Retourner la valeur moyenne de la qualité de l'air calculée.

**Exigences :**

- Disponibilité des données de qualité de l'air collectées à partir des capteurs.
- Spécification correcte de la zone circulaire par l'utilisateur.

**Pré-condition :**

- Les données des capteurs sont disponibles et à jour.
- L'utilisateur spécifie des coordonnées et un rayon valides pour la zone circulaire.

**Post-condition :**

La valeur moyenne de la qualité de l'air pour la zone circulaire spécifiée est calculée et retournée.

**Effets secondaires :**

Aucun.

**Fonction : Fonction Noter et Classer les Capteurs par Similarité :****Description :**

Cette fonction permet aux utilisateurs de sélectionner un capteur, puis de noter et de classer tous les autres capteurs en fonction de leur similarité avec le capteur sélectionné. La similarité est déterminée en comparant les données générées par les capteurs au cours d'une période de temps spécifiée. Le but de cette fonctionnalité est d'identifier les zones avec une qualité de l'air similaire.

**Entrées :**

Période de temps pour laquelle on veut calculer la moyenne.

**Source :**

- Liste des capteurs et leur localisation à partir du fichier "sensors.csv".
- Mesures des capteurs à partir du fichier "measurements.csv".

**Sorties :**

- Mesures de la qualité de l'air fournie par le capteur le plus proche de la position spécifiée.
- Distance jusqu'au capteur le plus proche.

**Destination :**

Main loop

**Action :**

1. Recevoir les coordonnées de la position géographique précise en entrée.
2. Déterminer le capteur le plus proche de la position spécifiée en utilisant des méthodes de calcul de distance (par exemple, distance euclidienne).

3. Récupérer les mesures de la qualité de l'air à partir du capteur le plus proche.
4. Calculer la distance entre la position spécifiée et le capteur le plus proche.
5. Renvoyer les mesures de la qualité de l'air et la distance jusqu'au capteur le plus proche.

**Exigences :**

- Disponibilité des données des capteurs générées au cours de la période de temps spécifiée.
- Sélection appropriée d'un capteur de référence par l'utilisateur.

**Pré-condition :**

- Les données des capteurs pour la période de temps spécifiée sont disponibles.
- L'utilisateur sélectionne un identifiant de capteur de référence valide.

**Post-condition :**

Une liste classée des capteurs ainsi que leur score en fonction de leur similarité avec le capteur de référence est générée et retournée.

**Effets secondaires :**

Aucun.

**Fonction : Fonction Déterminer la Qualité de l'Air à une Position Géographique Précise :**

**Description :**

Cette fonction a pour objectif de fournir la valeur de la qualité de l'air à une position géographique précise sur le territoire à un moment donné. Elle prend en entrée les coordonnées en tant que paramètres et renvoie les mesures des quatre types de données fournies par le capteur le plus proche de la position spécifiée. De plus, elle renvoie la distance jusqu'à ce capteur.

**Entrées :**

Coordonnées de la position géographique précise (latitude et longitude).

**Source :**

Mesures des capteurs à partir du fichier "measurements.csv".

**Sorties :**



Liste de tuples (classement, score)

**Destination :**

Main loop

**Action :**

1. Récupérer les données générées par les capteurs au cours de la période de temps spécifiée.
2. Calculer la similarité entre le capteur de référence et chacun des autres capteurs.
3. Pour calculer la similarité, pour chaque type de données, additionner les différences absolues des valeurs mesurées entre le capteur de référence et chaque autre capteur.
4. Trier tous les capteurs en fonction de leur similarité avec le capteur de référence.
5. Générer une liste classée des capteurs en fonction de leurs attribuer un score sur 100 de similarité.

**Exigences :**

- Disponibilité des données du capteur contenant les mesures des quatre types de données.
- Spécification correcte des coordonnées pour la position géographique précise.

**Pré-condition :**

- Les données du capteur contenant les mesures des quatre types de données sont disponibles.
- Des coordonnées valides pour la position géographique précise sont fournies.

**Post-condition :**

- Les mesures des quatre types de données fournies par le capteur le plus proche de la position spécifiée sont renvoyées.
- La distance jusqu'au capteur le plus proche est calculée et renvoyée.

**Effets secondaires :**

Aucun.

**Fonction : Fonction Suivi de l'Impact des Cleaners d'Air :**

**Description :**

Cette fonction facilite l'observation de l'impact des Cleaners d'air fournis par diverses entreprises sur la qualité de l'air. Ces machines sont capables de purifier l'air dans la zone environnante. Les fournisseurs installent ces Cleaners d'air dans le territoire. L'agence gouvernementale et les fournisseurs utilisent l'application AirWatcher pour surveiller l'impact des Cleaners sur la qualité de l'air. Cela inclut l'évaluation de paramètres tels que le rayon de la zone nettoyée et le niveau d'amélioration de la qualité de l'air.

**Entrées :**

Identifiant du Cleaner (CleanerID)

**Source :**

- Liste des cleaners et leur localisation (fichier "Cleaners.csv")
- Liste des capteurs et leur localisation à partir du fichier "sensors.csv".
- Mesures des capteurs à partir du fichier "measurements.csv".

**Sorties :**

- Rayon de la zone impactée par le Cleaner.
- Pourcentage moyen de l'amélioration apportée par le Cleaner.

**Destination :**

Main loop

**Action :**

1. Récupérer la localisation du Cleaner concerné
2. Calculer le rayon de la zone nettoyée en fonction des emplacements géographiques des capteurs, en comparant les données avant / après le déploiement du Cleaner.
3. Évaluer l'amélioration des indicateurs de qualité de l'air

**Exigences :**

Disponibilité des données provenant des purificateurs d'air installés

**Pré-condition :**

Les données provenant des purificateurs d'air installés sont collectées et accessibles.

**Post-condition :**

Le rayon impacté par le Cleaner est calculé et renvoyé.

L'amélioration moyenne des différents indicateurs de qualité de l'air est calculée et renvoyée.

**Effets secondaires :**

Aucun.

**Fonction : Fonction Gérer la Participation des Particuliers :**

**Description :**

Cette fonction gère la participation des particuliers à la génération de données sur la qualité de l'air en leur permettant d'installer des capteurs fixes à leur domicile. Les particuliers sont incités à contribuer aux données en gagnant des points pour l'utilisation de leurs données. Chaque fois que les données de leur capteur sont utilisées dans une requête, ils reçoivent un point. Si un des capteurs d'un utilisateur est marqué comme défaillant (cf. fonction décrite précédemment), celui-ci ne pourra plus gagner de points et sera signalé sur un csv généré, contenant des informations sur les particuliers, leurs scores et s'ils sont bloqués ou autorisés à gagner des points de récompense.

**Entrées :**

Aucune entrée

**Source :**

- Liste des individus et leurs capteurs (fichier "users.csv")
- Liste des capteurs et de leur état (défaillant ou pas).

**Sorties :**

- Fichier CSV contenant des informations sur les particuliers, leurs scores et leur statut d'éligibilité aux points de récompense.

**Destination :**

Main loop

**Action :**

1. Faire le lien pour chaque requête de l'application entre l'Id de capteur et l'Id de l'utilisateur correspondant..
2. Déterminer le statut d'éligibilité aux points de récompense de chaque particulier.
3. Mettre à jour les scores des particuliers en fonction de l'utilisation de leurs capteurs

4. Générer un fichier CSV contenant des informations sur les particuliers, leurs scores et leur statut d'éligibilité aux points de récompense.

**Exigences :**

Bonne vérification de l'utilisation des capteurs dans les requêtes de l'application.

**Pré-condition :**

- Vérification de l'éligibilité du particulier avant la mise à jour de ces points.
- Les données provenant des capteurs des particuliers sont collectées et accessibles.

**Post-condition :**

Un fichier CSV contenant des informations sur les particuliers, leurs scores et leur statut d'éligibilité aux points de récompense est généré.

**Effets secondaires :**

Aucun.

**Exigences non-fonctionnelles :****Vitesse :**

Vitesse de traitement : Viser un minimum de 1000 lectures de capteurs traitées par seconde.

Temps de réponse : Requêtes des utilisateurs traitée dans un délai de 500 millisecondes.

Temps d'exécution des algorithmes : Visez un temps d'exécution moyen de moins de 50 millisecondes pour les principaux algorithmes tels que l'analyse des capteurs, le calcul de la moyenne et la comparaison des capteurs.

**Taille :**

Taille du stockage des données : Limiter la taille totale de l'ensemble de données stocké par l'application à 100 Mo.

Utilisation de la mémoire : Viser à maintenir l'utilisation de la mémoire en dessous de 500 Mo pendant le fonctionnement normal.

**Facilité d'utilisation :**

Réactivité de l'interface utilisateur : S'assurer que l'interface utilisateur répond aux interactions des utilisateurs dans un délai de 100 millisecondes.

**Fiabilité :**

Résilience aux erreurs : Viser ce qu'au moins 95 % des événements ou conditions du système ne provoquent pas d'erreurs de l'application.

Intégrité des données : Maintenir une probabilité de corruption ou de perte de données inférieure à 0,01 % en cas de panne.

**Portabilité :**

Compatibilité avec les plates-formes : S'assurer de la compatibilité avec au moins 95 % des systèmes d'exploitation et des navigateurs web courants.

Dépendance à un matériel ou un logiciel spécifique : Minimiser les dépendances et assurez-vous de la compatibilité avec les configurations matérielles et logicielles standard.

# Ananalysis of security risks

Système	Atout	Vulnérabilité	Attaque	Risque	Niveau d'impact	Contre-mesure
Collecte de données	Large couverture des capteurs sur une grande zone	Manque de chiffrement pendant la transmission des données	Attaque de l'homme du milieu	Accès non autorisé aux données des capteurs	Élevé	Implémenter des protocoles de chiffrement pour la transmission des données
	Collecte régulière de données à intervalles fixes	Format de fichier CSV vulnérable	Attaques par injection (par exemple, injection SQL)	Corruption ou manipulation des données	Élevé	Implémenter des mécanismes de validation des entrées pour prévenir les attaques par injection
	Stockage centralisé simplifie la gestion	Accès non autorisé au serveur central	Intrusion sur le serveur	Modification ou vol non autorisé de données	Élevé	Renforcer les mesures de sécurité du serveur, telles que le contrôle d'accès et les systèmes de détection des intrusions
	Utilisation de capteurs basse consommation pour une empreinte écologique réduite	Manque de protection physique des capteurs contre la manipulation ou le vol	Altération physique des capteurs	Perte ou vol de capteurs	Faible	Installer des capteurs dans des boîtiers sécurisés et surveiller les sites de déploiement pour prévenir la manipulation ou le vol
	Utilisation de capteurs alimentés par énergie solaire pour une autonomie prolongée	Manque de redondance dans la collecte des données pour éviter les pertes	Panne temporaire ou définitive des capteurs	Perte de données en cas de panne	Faible	Mettre en place des mécanismes de redondance pour assurer la disponibilité continue des données collectées
Analyse des capteurs	Identifier et maintenir les capteurs défaillants	Manque d'authentification pour les données des capteurs	Attaques de spoofing (par exemple, spoofing IP)	Données de capteur fausses entraînant une analyse incorrecte	Élevé	Implémenter des mécanismes d'authentification pour les données des capteurs afin d'assurer l'intégrité et l'authenticité
	Agréger des données pour l'analyse statistique	Manque de vérifications d'intégrité des données	Manipulation des données	Résultats statistiques trompeurs	Moyen	Implémenter des vérifications d'intégrité des données pour détecter et prévenir la manipulation des données
	Noter et classer les capteurs pour leur similitude	Manque de contrôles d'accès sur le processus de comparaison des capteurs	Accès non autorisé aux données de comparaison	Accès non autorisé à des informations sensibles	Moyen	Implémenter des mécanismes de contrôle d'accès pour restreindre l'accès à la fonctionnalité de comparaison des capteurs

Système	Atout	Vulnérabilité	Attaque	Risque	Niveau d'impact	Contre-mesure
Requête géographique	Positionnement géographique précis	Manque de vérifications d'autorisation pour les requêtes géographiques	Accès non autorisé aux résultats de la requête	Exposition d'informations de localisation sensibles	Moyen	Implémenter des vérifications d'autorisation pour restreindre l'accès à la fonctionnalité de requête géographique
	Capacités de requête en temps réel	Manque de validation des entrées pour les coordonnées géographiques	Manipulation des entrées	Résultats de requête invalides ou trompeurs	Moyen	Implémenter des mécanismes de validation des entrées pour garantir la validité des coordonnées géographiques
Performance	Algorithmes efficaces pour l'analyse des données	Manque de surveillance de l'utilisation des ressources	Attaques par déni de service	Performance de l'application dégradée	Élevé	Implémenter des mécanismes de surveillance des ressources et de limitation de débit pour atténuer les attaques par déni de service
	Temps d'exécution minimal pour les algorithmes	Manque de mesures d'optimisation	Exploitation des faiblesses algorithmiques	Exécution inefficace des algorithmes	Élevé	Optimiser les algorithmes pour améliorer les performances et l'utilisation des ressources
Interface Utilisateur	Interface utilisateur sur mesure en fonction des rôles	Manque d'authentification pour les rôles utilisateur	Accès non autorisé aux données sensibles	Exposition d'informations sensibles à des utilisateurs incorrects	Moyen	Implémenter des mécanismes d'authentification pour les rôles utilisateur afin d'assurer le contrôle d'accès
	Accès restreint aux données	Manque de communication sécurisée entre l'interface utilisateur et les données	Interception ou manipulation des données	Accès ou modification non autorisés des données	Élevé	Implémenter des protocoles de communication sécurisée entre l'interface utilisateur et le stockage des données
	Interface utilisateur conviviale avec une courbe d'apprentissage réduite	Manque de gestion des sessions utilisateur pour prévenir les accès non autorisés	Session volée ou usurpée	Accès non autorisé à l'interface utilisateur	Faible	Mettre en œuvre des mécanismes de gestion de session robustes pour empêcher les accès non autorisés
	Accès restreint aux fonctionnalités de l'interface utilisateur basé sur les privilèges	Manque de journalisation des activités utilisateur pour l'audit et la surveillance	Activité malveillante ou suspecte	Incapacité à suivre ou à détecter les activités malveillantes	Faible	Mettre en œuvre une journalisation efficace des activités utilisateur pour la détection et la réponse aux incidents
Surveillance des performances	Capacité à surveiller les performances des systèmes pour détecter les anomalies	Manque de sécurité dans les mécanismes de surveillance des performances	Attaques visant à altérer les données de surveillance	Détection inexacte ou altération des performances du système	Élevé	Implémenter des mécanismes de surveillance sécurisés et des contrôles d'intégrité pour les données de performance

Système	Atout	Vulnérabilité	Attaque	Risque	Niveau d'impact	Contre-mesure
	Temps de réponse rapide aux événements critiques	Manque de tolérance aux pannes	Panne ou indisponibilité du système	Temps de réponse retardé ou défaillance du système	Élevé	Concevoir une architecture résiliente avec des mécanismes de redondance et de récupération en cas de panne
Gestion des autorisations	Attribution de droits d'accès en fonction des rôles et des responsabilités	Manque de vérifications d'autorisation lors de l'attribution des droits d'accès	Attribution de droits d'accès inappropriés ou excessifs	Accès non autorisé à des données sensibles ou à des fonctionnalités critiques	Moyen	Implémenter des mécanismes de contrôle d'accès basés sur des règles et des politiques de sécurité
	Gestion sécurisée des identités et des informations d'authentification	Manque de sécurité dans le stockage et la transmission des informations d'identification	Vol ou compromission des informations d'identification	Utilisation frauduleuse des identités ou accès non autorisé	Élevé	Implémenter des protocoles et des techniques de cryptage sécurisés pour protéger les informations d'identification
Intégrité des données	Maintien de l'intégrité des données pour garantir leur fiabilité et leur précision	Manque de mécanismes de vérification de l'intégrité des données	Altération ou corruption des données	Perte de fiabilité ou d'exactitude des données	Moyen	Implémenter des mécanismes de vérification d'intégrité pour détecter et prévenir la corruption des données



Aspect	Problème potentiel	Test de validation	Méthode
Analyse du fichier CSV	Fichier CSV manquant ou corrompu	Vérification de l'existence du fichier CSV et de son intégrité	Utilisez les fonctionnalités de manipulation de fichiers en C++ pour vérifier si les fichiers existent et lisez leur contenu pour assurer qu'ils ne sont pas vides.
	Format de données invalide	Vérification du format des données CSV selon les spécifications	Utilisez les fonctionnalités de lecture de fichiers en C++ pour lire les fichiers CSV ligne par ligne, puis utilisez des expressions régulières ou d'autres méthodes appropriées pour valider le format des données.
	Absence de données	Vérification si le fichier CSV contient des données	Vérifier si le fichier CSV est vide et si l'application gère cette situation correctement
	Données dupliquées	Vérification si les données CSV contiennent des doublons	Lisez les lignes de données des fichiers CSV et utilisez une table de hachage ou une autre structure de données pour suivre les lignes de données déjà rencontrées afin de vérifier les doublons.
	Taille excessive du fichier CSV	Gérer efficacement les fichiers CSV de grande taille	Créez un fichier CSV volumineux, puis testez les performances et la stabilité de l'application lors de la lecture et du traitement de ce fichier.
	Données en dehors de la plage attendue	Vérification si les valeurs des mesures sont plausibles (ex. latitude>90)	Écrivez des fonctions de test pour vérifier si les valeurs de mesure sont dans la plage attendue, par exemple, vérifiez si la latitude est comprise entre -90 et 90 degrés.
	Incohérence entre les fichiers CSV	Vérification de la cohérence entre les données de différents fichiers	Comparez les champs de données correspondants dans différents fichiers pour garantir leur cohérence.

Aspect	Problème potentiel	Test de validation	Méthode
Analyse des données	Erreurs d'analyse ou d'interprétation des données	Vérification de la précision des résultats d'analyse	Comparer les résultats d'analyse avec des données connues pour vérifier leur exactitude
	Manque de cohérence entre les résultats	Vérification de la cohérence des résultats d'analyse	Exécuter plusieurs analyses sur les mêmes données et vérifier si les résultats sont cohérents
	Calcul de la moyenne de la qualité de l'air dans une zone circulaire	Calculer la moyenne de la qualité de l'air dans une zone spécifiée pour une période de temps donnée	Créer une zone circulaire virtuelle et une période de temps, calculer la qualité de l'air et vérifier si le résultat correspond aux attentes
	Évaluation et classement des capteurs en fonction de la similarité	Évaluer et classer les autres capteurs en fonction de la similarité par rapport à un capteur sélectionné	Vérifier si le résultat correspond aux attentes
	Détection des données non fiables provenant des particuliers	Analyser les données fournies par les capteurs individuels pour déterminer leur fiabilité	Créer des données de capteurs virtuels provenant d'utilisateurs individuels, les analyser et vérifier si elles sont fiables. Vérifier si le résultat correspond aux attentes
Gestion de la mémoire	Fuites de mémoire	Utilisation d'outils de débogage pour surveiller la mémoire	Exécutez l'application et utilisez des outils de débogage de mémoire en C++, tels que Valgrind, pour vérifier les fuites de mémoire ou d'autres problèmes.
	Surallocation de mémoire	Vérification si la mémoire est allouée de manière adéquate	Écrivez des fonctions de test pour simuler l'allocation et la libération de mémoire de l'application, puis vérifiez si son comportement est conforme aux attentes.

Aspect	Problème potentiel	Test de validation	Méthode
Efficacité de l'algorithme	Temps d'exécution excessif	Mesurer le temps d'exécution de chaque algorithme	Utilisez les fonctionnalités de mesure du temps en C++ pour enregistrer les temps de début et de fin de chaque algorithme, puis calculez le temps d'exécution.
	Complexité algorithmique excessive	Vérifier si les algorithmes sont optimisés et efficaces	Évaluez l'optimisation et l'efficacité de chaque algorithme en analysant leur temps d'exécution et l'utilisation des ressources.
Gestion des erreurs	Erreurs de saisie utilisateur	Validation des entrées utilisateur pour éviter les erreurs	Écrivez des fonctions de test pour simuler différentes entrées utilisateur et vérifiez si l'application les gère correctement.
	Manque de feedback sur les erreurs	Assurer que les utilisateurs sont informés des erreurs survenues	Simulez des situations d'erreur et vérifiez si les messages d'erreur appropriés sont affichés pour informer les utilisateurs.
	Logging et journalisation des erreurs	Assurer que les erreurs sont correctement enregistrées et journalisées	Testez les fonctions de journalisation d'erreurs pour assurer qu'elles enregistrent les erreurs de manière appropriée et qu'elles peuvent être suivies par les développeurs.
	Erreurs liées aux autorisations d'accès	Vérifier si les erreurs liées aux autorisations d'accès sont correctement gérées	Simulez des erreurs d'autorisation d'accès et vérifiez si l'application les traite correctement en refusant l'accès ou en affichant des messages d'erreur appropriés.
Optimisation des performances	Utilisation excessive de la CPU	Surveiller l'utilisation du processeur pendant l'exécution	Utilisez des outils de surveillance de l'utilisation du processeur pour mesurer et analyser l'utilisation du processeur par l'application pendant son exécution.

Aspect	Problème potentiel	Test de validation	Méthode
	Accès disque excessif	Surveiller les opérations d'entrée/sortie pendant l'exécution	Mesurer la fréquence et la durée des accès disque pendant l'exécution de l'application
	Utilisation excessive de la mémoire	Surveiller l'utilisation de la mémoire pendant l'exécution	Observer l'utilisation de la mémoire pendant l'exécution et identifier les zones où une optimisation est nécessaire

```
user@DESK-A12DR34:~/Dir1/Dir2/AirWatcher/$
```

```
[0] Quitter
```

```
[1] Statistiques générales
```

```
[2] Statistiques personnalisées
```

```
[3] Capteurs défaillants
```

```
Entrer Sélection :
```

```
>
```

#### Page d'accueil:

-Menu principal-

La sélection se fait en entrant le numéro de l'option souhaitée dans l'invite de commande.

```
Statistiques générales :
```

```
[0] Retour au Menu principal
```

```
*LES STATS*
```

```
Entrer Sélection :
```

```
>
```

#### Page stats (1):

-Affichage des statistiques générales-

l'invite de commande ne sert qu'à retourner au Menu principal.

Permet de visualiser les statistiques sur la pollution, les capteurs, etc...

Statistiques personnalisées (Choix du capteur de départ) :  
[0] Retour au Menu principal

*\*LES CAPTEURS NON-DÉFAILLANTS\**

*[1] capteur 1 : {longitude : latitude}*

*[2] capteur 2 : {longitude : latitude}*

*...*

Sélectionner un capteur de départ :

>

Menu stats avancées (1) :

Sélectionner le capteur au centre de la zone qu'on l'on souhaite étudier en entrant son numéro dans l'invite de commande.

Statistiques personnalisées (Choix de la taille de la zone) :  
[0] Retour au Menu principal

*Capteur sélectionné : n° X*

*[1] Petite*

*[2] Moyenne*

*[2] Grande*

Sélectionner une taille de zone:

>

Menu stats avancées (2) :

Choisir une taille de zone autour du capteur sélectionné en entrant le numéro correspondant dans l'invite de commande.

Statistiques personnalisées (Choix de la temporalité) :

[0] Retour au Menu principal

*[1] Instant donné*

*[2] Durée donnée*

Sélectionner une temporalité:

>

Menu stats avancées (3) :

Sélectionner un type de temporalité (instantanée ou étendue) en entrant le numéro correspondant dans l'invite de commande.

Statistiques personnalisées (Choix de l'instant) :

[0] Retour au Menu principal

*\*LISTE DES INSTANTS DISPONIBLES\**

*[1] Instant 1 : {DD/MM/YYYY HH:MM}*

*[2] Instant 2 : {DD/MM/YYYY HH:MM}*

*...*

Sélectionner un instant:

>

Menu stats avancées (4) :

Sélectionner l'instant d'étude des données des capteurs en entrant le numéro correspondant dans l'invite de commande.

Statistiques personnalisées (Choix de la durée) :  
[0] Retour au Menu principal

*\*LISTE DES INSTANTS DISPONIBLES\**

[1] Instant 1 : {DD/MM/YYYY HH:MM}

[2] Instant 2 : {DD/MM/YYYY HH:MM}

...

Sélectionner un temps de départ :

>

Menu stats avancées (5) :

Sélectionner l'instant de départ d'étude des données des capteurs en entrant le numéro correspondant dans l'invite de commande.

Statistiques personnalisées (Choix de la durée) :  
[0] Retour au Menu principal

*\*LISTE DES INSTANTS DISPONIBLES\**

[1] Instant 1 : {DD/MM/YYYY HH:MM}

[2] Instant 2 : {DD/MM/YYYY HH:MM}

...

Sélectionner un temps de départ :

> 17

Sélectionner un temps de fin :

>

Menu stats avancées (6) :

Sélectionner l'instant de fin d'étude des données des capteurs en entrant le numéro correspondant dans l'invite de commande.



(  $instant_{départ} < instant_{final}$  )



Statistiques personnalisées:  
[0] Retour au Menu principal

*\*LES STATS\**

Entrer Sélection :  
>

#### Page stats (2):

-Affichage des statistiques personnalisées-  
l'invite de commande ne sert qu'à retourner au Menu principal.  
Permet de visualiser les statistiques sur la pollution, les capteurs, etc...

Capteurs défaillants:  
[0] Retour au Menu principal

*\*LES CAPTEURS\**

*[1] capteur 1 : {longitude : latitude}*

*[2] capteur 2 : {longitude : latitude}*

*...*

Voir les information d'un capteur -> Sélectionner un capteur :  
>

#### Liste des capteurs défaillants (1):

-Affichage des capteurs défaillants-  
l'invite de commande permet de retourner au Menu principal  
ainsi qu'à sélectionner un capteur défaillant afin d'en voir  
les informations.

Capteurs défaillants:  
[0] Retour au Menu principal

*\*LES CAPTEURS\**

*[1] capteur 1 : {longitude : latitude}*

*[2] capteur 2 : {longitude : latitude}*

*...*

Voir les information d'un capteur -> Sélectionner un capteur :  
> 7

*\*LES INFORMATIONS\**

- *Fournisseur : Sensor Co*
- *Contact : email@service.com*
- *localisation : {longitude : latitude}*

Sélectionner un autre capteur/Retour au Menu principal :  
>

Liste des capteurs défaillants (2):

-Affichage des capteurs défaillants-

l'invite de commande permet de retourner au Menu principal  
ainsi qu'à sélectionner un autre capteur défaillant afin d'en  
voir les informations.