

## **Analyse détaillée**

# **SOMMAIRE**

**I/ Persistance des données**

**II / Algorithmes**

**III / Gestion du temps**

## **I/ Persistance des données**

La conception d'une base de données doit être réalisée dans le but de garder en mémoire les données de l'utilisateur et des sondes.

Les règles de gestion de cette base de données sont :

- Une espèce végétale peut-être l'espèce de plusieurs végétaux et un végétal appartient à une seule espèce
- Une zone contient plusieurs végétaux, ou n'en contient pas et un végétal peut-être dans plusieurs zones.
- Une zone historise plusieurs données et une donnée historisée appartient à une zone.
- Une zone attend plusieurs données et une donnée attendu appartient à une zone.
- Une alerte se déroule dans une zone et une zone peut avoir plusieurs alertes ou aucune.
- Une alerte n'est due qu'à un seul type d'alerte et un type d'alerte peut concerner plusieurs alertes.
- Une sonde peut-être concerné par plusieurs types d'alerte et un type d'alerte est concerné par une sonde.
- Une zone peut correspondre à plusieurs types d'alerte et un type d'alerte correspond à une zone.
- Une alerte envoie un message à potentiellement plusieurs personnes responsables et une personne responsable peut-être concerner par plusieurs alertes.
- Une personne responsable peut-être responsable de plusieurs zones et une zone possède plusieurs personnes responsables.
- Une sonde peut concerner plusieurs données reçues et une donnée reçu appartient à une seule sonde à une date donnée
- Une sonde peut concerner plusieurs données attendues et une donnée attendu appartient à une seule sonde à une date donnée

A l'issue de ces règles de gestion un dictionnaire des données ( Annexe 2) puis un modèle conceptuel des données à été construit ( Annexe 1).

## **II / Algorithmes**

### **Ajout de type d'alerte dans une zone :**

Input - Le type d'alerte à ajouter

Ouput - boolean

1 – Stocker tous les types d'alertes d'une zone

2 - Si il existe un type d'alerte qui a la même condition d'alerte et la même sonde que les types d'alertes stocker alors

3 - Retourner faux

4 – Sinon associer le type d'alerte à la zone dans la base de donnée

5 – Retourner vrai

### **Vérification donnée reçue par rapport aux données attendues :**

Input – Données attendues ( couple date / donnée / marge ), donnée reçue ( couple date / donnée )

Ouput – boolean

1 – prec ← dernière valeur attendue avant la donnée reçue et la stocker.

2 – next ← prochaine valeur attendue après la donnée reçue et la stocker

3 – Si il existe une valeur prec et next alors

3.1 – Si prec et next ont la même valeur alors

3.2 – Si la donnée reçue est compris dans l'intervalle de la donnée attendu ( [prec – prec.marge ; prec + prec.marge ] ) alors

3.2.1 – Retourner vrai

3.2.2 – Sinon retourner faux

3.3. – Sinon

3.3.1 – moyenne ← (prec + next )/ 2

3.3.2 – moyenne marge ← (prec.marge + next.marge)/2

3.3.3 – Si la donnée reçue est compris dans l'intervalle de la donnée attendue ( [moyenne – moyenne marge ; moyenne + moyenne marge ] ) alors

3.3.3.1 – Retourner vrai

3.3.4 – Sinon retourner faux

4 – Sinon retourner faux.

### Envoyer Alerte :

Input – Données attendues de la zone ( couple date / donnée / marge ), donnée reçue ( couple date / donnée ), zone

Output - Rien

1 – vérification ← Vérifier la valeur reçue ( utiliser le pseudo-code précédent

2 – Récupérer les personnes responsables de la zone

3 – Si vérification est faux alors

3.1 Récupérer le type d'alerte correspondant à la donnée reçu

3.2 Si ce type alerte existe alors

3.2.1 Récupérer le message d'alerte, et l'envoyer par mail à toutes les personnes responsables avec pour objet la zone concernée et la sonde concernée.

3.3 Sinon envoyer un message d'erreur, expliquant qu'il n'y a pas de type d'alerte pour cette situation

### **III / Gestion du temps**

Dû à un temps de conception / développement très court ( 3 semaines ) et un manque de main d'œuvre ( une personne ).

La gestion du projet c'est fait de manière à maximiser le temps réservé pour le développement du projet (interface et liaison avec la base de donnée ), et minimiser le temps de réflexion ( conception logiciel ).

La conception finale de chaque module du logiciel c'est faite au fur et à mesure de la production du logiciel , lors de l'ajout des fonctionnalités.

Une première production du cahier des charges, conception de la base de données et logiciel c'est faite le premier jour du projet. Et par la suite vont évoluer jusqu'à identifier tous les besoins réalisables et nécessaires pour le projet.

Les jeux de test s'effectueront à la fin du projet, par manque de temps aucun test unitaire sur la base de donnée ne sera effectué , ceux-ci seront validés lors du test des fonctionnalités de l'interface graphique.

Voir l'annexe 4 pour la répartition et l'estimation de la durée des tâches. ( Diagramme de Grantt)