**3. Chapitre 3 :**

*Analyse et conception*

**3.1. Introduction**

Dans ce chapitre nous aborderons l’identification des besoins pour la réalisation d’un système d’air monitoring, par la suite nous mettrons en évidence le côté conceptuel de note projet qui constitue une étape fondamentale et qui précède la partie implémentation.

Cette partie permettra de détailler les différents diagrammes et scenarios à implémenter dans la phase suivante.

Les diagrammes utilises sont les suivants :

* Le diagramme de flux de processus
* Le diagramme UML de classe

Les deux diagrammes ont été réalisés avec la plateforme en ligne [www.lucidchart.com](http://www.lucidchart.com).

**3.2. Analyse des besoins**

L’analyse des besoins est la première phase dans le cycle de vie de tout projet. Elle consiste à définir les services qui seront rendus à l’utilisateur et les contraintes sous lesquelles ce dernier devra fonctionner.

Notre objectif est de créer un prototype d’un système de qualité d’air avec la plateforme Arduino, qui se constitue d’un capteur de gaz et d’un module GPS, puis représenter les résultats dans une interface WEB.

**3.2.1. Les besoins fonctionnels**

En ce basant sur les projets déjà existants, on a identifié les différents besoins pour la conception d’un prototype d’un système d’évaluation de la qualité d’air :

* Arduino Uno board.
* Un capteur de gas, on a utilisé le MQ- 135
* Un module GPS.
* Une application WEB pour l’affichage des résultats sur une map.

**3.3. Etude préliminaire**

**3.3.1. Identification des acteurs :**

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Il peut consulter et ou modifier directement l'état du système en émettant et ou recevant des messages éventuellement porteurs de données.

Dans notre cas, nous avons identifié deux acteurs matériaux et un utilisateur.

Les acteurs matériaux sont :

* **Le capteur de Gas :** le model MQ- 135 que nous utilisons donne offre la possibilité de mesurer le Monoxyde de Carbone.
* **Le module GPS :** qui nous permettra d’avoir la longitude et l’attitude de notre point d’enregistrement.
* **L’utilisateur :** l’acteur qui consulte le produit final qui est une carte géographique qui montre des zones colorées selon la qualité d’air à ce point-là.

**3.3.2. Identification des messages :**

Pour les différents acteurs il faut chercher les messages qui déclenchent un comportement du système attendu par ces derniers dans le cadre de leurs activités et de l’autre côté, chercher les messages émis par le système à l’intention de ces acteurs.

Exemple de messages émis par le système :

* Calcul de AQI selon les valeurs de CO enregistre
* Interface graphique pour l’affichage des résultats
* Exemple de messages reçu par le système :
* Capture des valeurs de CO dans l’air
* Demande de la qualité d’air dans un endroit.

**3.4. Capture des besoins :**

La capture des besoins est la première étape dans la branche fonctionnelle. Elle consiste à modéliser et à détailler les besoins précédemment exprimés lors de l'étude préliminaire sous forme de cas d'utilisation. Cette étape se compose des phases suivantes :

**3.4.1. Diagramme de classes UML :**

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet. Alors que le diagramme de cas d'utilisation montre un système du point de vue fonctionnel, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Il permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Pour notre type nous avons besoin de deux classes une pour la base de donnes et l’autre pour le service WEB qui sera envoyé en format JSON.

* **CO\_DATA :**

Cette classe sert a enregistré les données capturées par le capteur MQ135 qui est le PPM et les coordonnées géographiques (lat, long) reçu par le GPS.

* **AQI :**

Le AQI est calculé selon une formule qui prend en paramètres plusieurs valeurs de ppm enregistrées la même position géographique, cette classe comprends la valeur de l’AQI, la gravité qui est conclue à travers la valeur selon une certaine échelle et au final les coordonnées géographiques (lat, long).

Nous utiliserons cette classe quand on enverra les données à partir du serveur vers le Front End en format JSON.

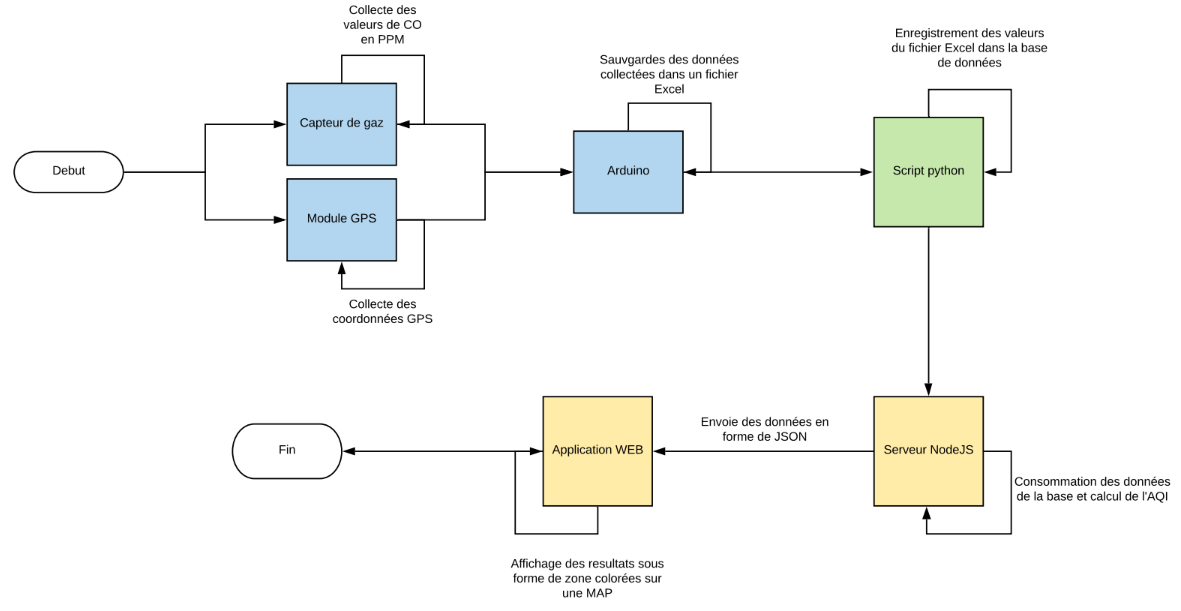
**3.4.2. Digramme de flux de processus :**

Figure Diagramme de flux de processus

Dans ce diagrammes les processus sont divisés en trois catégories, les processus en bleus sont purement liés au composants physiques, celui en vert est le middleware entre les matériaux et l’application web, et ceux en jaunes représentent le traitement fait pour l’application.

* **Le Hardware :**

Pour notre prototype on a utilisé un Arduino Uno avec un capteur de gas le MQ135 spécifiquement pour détecter le CO2 en PPM grâce à la bibliothèque ‘MQ135.h’, en addition a un aussi un module GPS le G28U7FTTL pour la sauvegarde des coordonnées géographique la longitude et la lattitude.

* **Le Middleware :**

Puisqu’on n’avait pas un module de communication, on sauvegarde les données du capteur de gas et du GPS dans un fichier Excel. Puis avec on lance un script python pour faire la migration des données à partir du fichier Excel vers la base de données.

* **L’application web :**

Pour le back-end on a besoin seulement d’un service qui calcule le AQI et envoie le résultat vers le Front End en un format JSON, pour cela on va utiliser un serveur NodeJS.

Le Front End est implémenté avec le Framework Angular et utilisant la librairie Leaflet. L’idée principale est de montrer sur une carte différentes zones coloré selon le AQI.

**3.5. Conclusion :**

Ce chapitre a eu pour but d’éclaircir l’aspect conceptuel de notre prototype, à travers le diagramme de flux de processus et le diagramme de classe UML.

Dans le chapitre suivant on va présenter les différents outils qui nous ont permis la réalisation du prototype ainsi qu’une explication détaillée sur l’utilisation de cette dernière.