CAPTEUR DE POLLUTION PORTABLE CONNECTE

Projet EPSI i4 - 2015 / 2016

## Présentation de l’équipe

Le projet est constitué autour de 5 membres :

* Benoit Grand : Chef de projet et Développeur mobile ;
* Pierre Hosteins : Développeur mobile et Community Manager ;
* Aymeric Mivielle : Développeur Arduino et Développeur web ;
* Jessica Lacaze : Développeuse Spring et Hadoop ;
* Julien Maupetit : Assistant chef de projet et Développeur Spring.

## Le contexte et objectifs du projet

### Le contexte

Le projet doit être réalisé au cours de la 4ème année du cursus ingénierie de l’EPSI.

Notre projet se place dans un cadre d’une sensibilisation à la pollution en zone urbaine à l’approche de la COP21 le 11 décembre à Paris 2015.

### Les objectifs

Le but est de créer une solution matérielle et logicielle permettant à chaque citoyen de pouvoir analyser les pollutions avec lesquelles il peut être en contact. Cette analyse permettra de voir sur le long terme quel peut être l’impact d’un projet comme le Grand Paris Plaine Commune qui a comme objectif d’améliorer la qualité de l’air dans l’agglomération en interdisant les véhicules d’avant 2000.

### Le périmètre

Nous nous concentrons sur les utilisateurs de terminaux Android en France.

## Description matérielle

### Capteur de pollution portable connecté

Le projet vise à concevoir un capteur de pollutions portable connecté. Celui-ci permettrait de récupérer des données brutes sur diverses pollutions :

* Pollution de l’air ;
* Pollution sonore.

Mais également d’autres données :

* Température ;
* Humidité ;
* Géo localisation ;
* Date / Heure.

L’ensemble de ces données permettrait de créer une carte dynamique de la pollution au sein de Bordeaux Métropole. Le capteur aura pour objectif d’être transporté dans la ville grâce à divers moyens de locomotions :

* Vélo ;
* Voiture ;
* Moto ;
* Bus ;
* Tram.

Cela afin de pouvoir prélever le plus de données possible dans divers zones non couvertes par des stations météos et des capteurs de pollutions, mais aussi de pouvoir les récupérer « en mouvement ».

(Voir PDF Présentation du projet).

## Description fonctionnelle

### L’application web

#### Objectifs

* Visualisation d’une carte dynamique des zones de pollutions (grâce aux informations de géo localisation) dans le temps avec une visualisation de l’évolution ;
* Moteur de rechercher permettant d’avoir les données chiffrées (Ex : Pessac entre juin et septembre 2015).

#### Arborescence

|  |  |
| --- | --- |
| NOM RUBRIQUE | DESCRIPTIF |
| Carte dynamique de la région | Une carte dynamique de la région Bordeaux Métropole affichant les niveaux de pollution avec un système de filtre. |
| Moteur de recherche | Un moteur de recherche permettant de consulter les données brutes pour une région et une période donnée. |
| Présentation du projet | Une page de présentation du projet, ainsi que de l’équipe avec un lien vers les différents repo. |
| Formulaire de contact | Une page avec un formulaire de contact. |
| Documentation assemblage capteur | Page avec un wiki pour l’assemblage du capteur arduino. |
| Documentation API | Page avec un wiki pour l’utilisation de l’API. |

#### Langues

|  |  |
| --- | --- |
| Langue | Type |
| Français | Site complet |
| Anglais | Site complet |

### L’application mobile

#### Objectifs

* Pont entre le capteur et le serveur (si l’utilisateur possède un capteur). Elle aura pour rôle de détecter le moment le plus adéquate pour lancer la récupération des données par le capteur. Par la suite le téléphone hôte enverra les données au serveur via l’appel d’un service web ;
* Permettre la visualisation d’une carte dynamique des zones de pollutions (grâce aux informations de géo localisation).

#### Arborescence

|  |  |
| --- | --- |
| NOM RUBRIQUE | DESCRIPTIF |
| Appairage au capteur | Bouton permettant de se connecter via Bluetooth au capteur afin de récupérer les données. |
| Carte dynamique de la région | Une carte dynamique de la région Bordeaux Métropole affichant les niveaux de pollution avec un système de filtre. |

#### Langues

|  |  |
| --- | --- |
| Langue | Type |
| Français | Application complète |
| Anglais | Application complète |

### L’application serveur

#### Objectifs

* L’application serveur devra permettre via l’intermédiaire de service web de pouvoir recevoir les données des capteurs via des requêtes POST ;
* L’application mobile et web devra également pouvoir récupérer les données via des services web GET avec un certain nombre de paramètre :
  + Récupération pour un code commune et une date (début et fin) ;
  + Récupération par une donnée de géo localisation (longitude et latitude) ainsi que période.

### La base de données

### Le but de la base de données sera d’héberger les données envoyées par l’ensemble des capteurs. La base de données sera non relationnelle. Elle sera performante et permettra la gestion d’un flux de données important.

## Livrables

### Développements

|  |  |
| --- | --- |
| TITRE | DESCRIPTION |
| Application mobile | L’application mobile Android. |
| Application web | L’application web. |
| Application serveur | L’application serveur permettant de traiter les données, mais également de les consulter via des web services. |
| Base de données | La base de données stockant l’ensemble des données brutes. |

### Documents

|  |  |
| --- | --- |
| TITRE | DESCRIPTION |
| Cahier des charges | Le cahier des charges (objectifs, besoins etc) |
| Spécifications techniques | L’ensemble des specs (Mockups, UML etc) |
| Documentation Arduino | Assemblage de l’Arduino. |
| Documentation API | Récupération des données via l’API. |

## Versions

|  |  |  |
| --- | --- | --- |
| NUMERO | FONCTIONNALITES | DETAILS |
| Version 1 | Arduino Bluetooth | Capteur |
|  | Application mobile | Appairage et carte |
|  | Application web | Carte, moteur de rechercher,  Contact, Présentation et documentation |
|  | Application serveur | Webservices POST et GET |
| Version 2 | Option capteur d’intérieur | Permet de spécifier que l’on va utiliser le capteur en intérieur, les données ne seront pas en opendata, et il y aura un compte. |
| Version 3 | Arduino Wifi ou GSM | Capteur |