

*Trust'e Pot*

**Projet soutenu le 19/01/2018**

**Membres du Groupe**

*Geoffroy Bernard*

*Valentin Kohler*

*Virgile Saint-Pé*

*Pierrick Jeandel*

*Samet Alaca*

**Projet effectué au département informatique de l'IUT de Metz**

# **Sommaire**

## **I. Sujet du projet**

1. Description du projet
2. Motivation du projet
3. Utilisateurs
4. Exemple d'utilisation

## **II. Gestion du projet**

1. Répartition du travail
2. Diagramme de GANTT
3. Répartition des tâches prévues
4. Comparaison

## **III. Analyse du Sujet**

1. Schéma général du projet
2. Simulateur de données
3. Application Web
4. Capteurs

## **IV. Perspective et conclusions**

1. Perspective
2. Conclusion

## **I)Sujet du projet**

### **1)Description du projet**

Le sujet sur lequel nous avons du travailler est la réalisation d'un pot connecté.

Ce pot connecté doit permettre à un utilisateur lambda de pouvoir surveiller l'évolution de différents paramètres relatifs aux plantes qu'il contient.

Par exemple, l'utilisateur peut choisir de surveiller l'évolution de la température, de l'humidité, ou bien encore du pH de l'environnement de la plante.

L'idée était donc de construire un capteur permettant de surveiller les différentes évolutions des paramètres souhaités au cours du temps et en permanence.

Pour surveiller l'évolution des paramètres, il était nécessaire de designer une interface comme par exemple un site web qui ferait office d'intermédiaire entre le capteur et l'utilisateur.

L'utilisateur devait pouvoir se créer un profil et y ajouter des capteurs.

L'association du capteur à une espèce de plante et la sélection des différents paramètres se font également via cette interface.

Par ailleurs, les données devaient pouvoir être accessibles par l'utilisateur à tout moment.

Les données relatives aux différents utilisateurs, les mesures récoltées par les différents capteurs mais également les différentes espèces de plantes devaient être stockées.

### **2)Motivation du projet**

Ce projet a été intéressant à réaliser pour tout notre groupe.

En effet, certains ont été attirés par l'aspect science et biologie qui était présent dans ce projet. D'autres par les technologies que nous devrions aborder (IoT).

Le professionnel qui nous a encadré a également été d'une grande aide. Il nous a permis d'avancer continuellement au fur et à mesure aux briefings auxquels nous avons participé. De plus, il nous a apporté son regard de professionnel et nous a permis de penser à des éléments que nous n'aurions pas pris en compte.

Cela a permis à chaque membre du groupe de progresser au fur et à mesure des séances.

Chacun a donc pu trouver un intérêt dans la réalisation de ce projet.

### **3)Utilisateurs**

Les utilisateurs visés par notre projet se divisent en deux catégories.

D'un côté les utilisateurs souhaitant construire leur propre capteur et utiliser notre projet de manière simple.

D'un autre côté les utilisateurs expérimentés voulant pouvoir améliorer le code de notre projet.

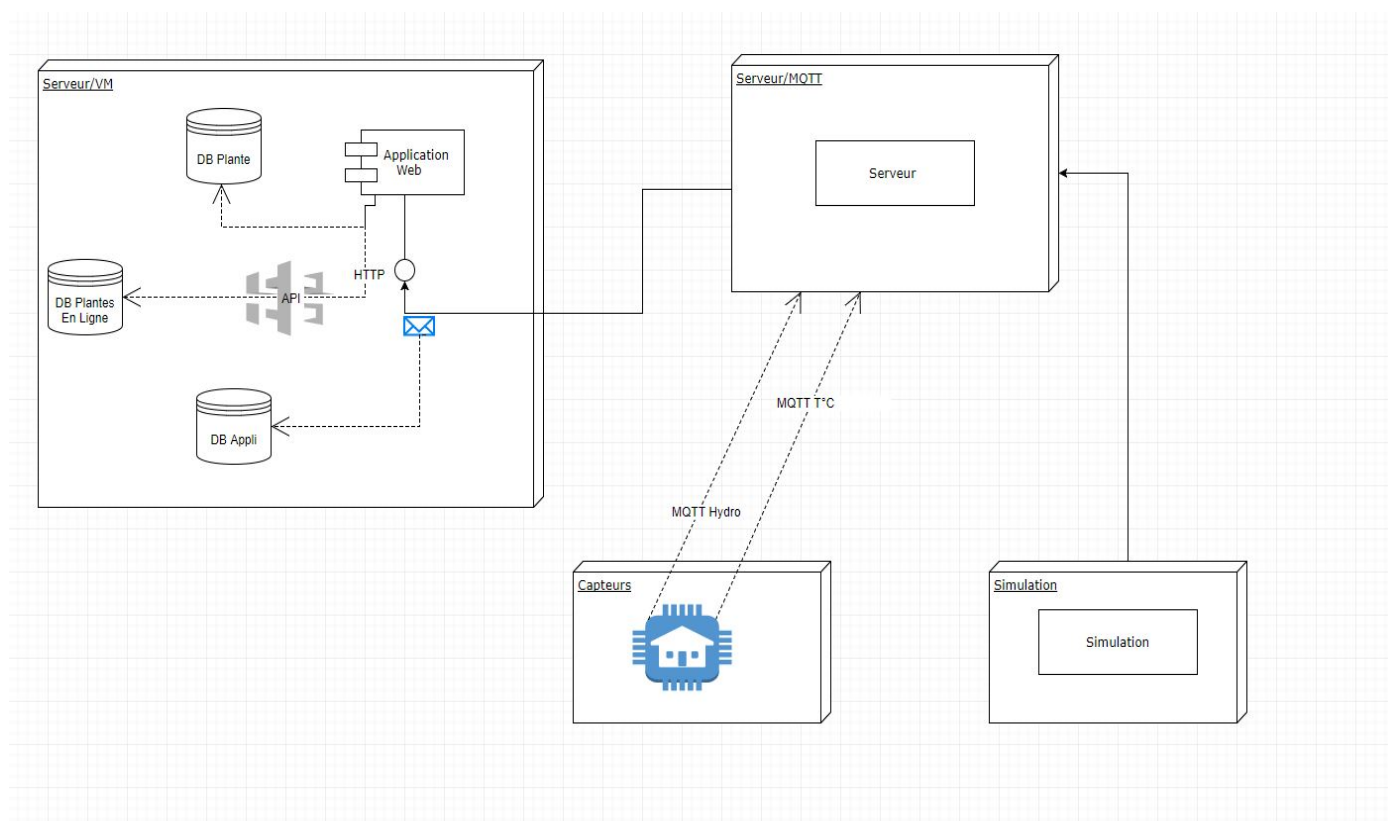
En effet nous avons pour objectif de le rendre disponible en open source un fois le projet terminé.

De ce fait, ce projet pourrait évoluer grâce au travail collectif d'utilisateurs passionnés.

*Description du projet, Motivation et Utilisateur réalisées par Valentin Kohler*

### **4)Exemple d'utilisation**

Dans cet exemple, nous allons tout d'abord voir le schéma de fonctionnement de l'application, puis nous verrons comment tout fonctionne du côté utilisateur.



En résumé, le capteur envoie par protocole MQTT des informations au serveur, celui-ci les envoie à l'application web ainsi qu'à notre base de données de l'application. Cette base de données contient donc toutes les informations nécessaires à l'application (les capteurs, les plantes de l'utilisateur, les données des capteurs...). L'application web compare ensuite les données reçues avec notre base de données des plantes à l'aide d'un API afin de d'envoyer d'éventuelles notifications.

D'autre part, nous avons réalisé un simulateur permettant de tester l'application sans avoir de capteur, celui-ci peut envoyer des données comme un capteur le ferait et nous permet ainsi de visualiser la réaction de notre application face à ces données.

Du point de vue de l'utilisateur, voici comment l'application fonctionne :

L'utilisateur va sur l'application et crée un compte.

Il se connecte et accède à son profil.

D'ici, il peut configurer quel type de notification il veut avoir (Mail, notification sur son application). Il peut également sélectionner l'envoi de certains types de notification uniquement (quand la température est trop élevée, manque d'eau...). Enfin, il peut choisir la fréquence de ces notifications.

L'utilisateur peut maintenant ajouter ses plantes à l'aide de la base de donnée et leur associer un ou plusieurs capteurs.

Finalement, l'utilisateur peut regarder l'évolution de ses plantes à l'aide de graphes et connaître ses caractéristiques actuelles (température, pH, etc...).

Ainsi, l'application compare les données reçues par les capteurs de l'utilisateur avec celle de notre base de données, de cette façon, si une des caractéristique vitale de la plante n'est pas respectée (manque d'eau, de chaleur ...) l'application avertit l'utilisateur.

*Exemple rédigé par JEANDEL Pierrick.*

## II)Gestion du projet

- Comparer la répartition prévue et le travail réellement effectué à la fin des 5 semaines.

### 1)Répartition du travail

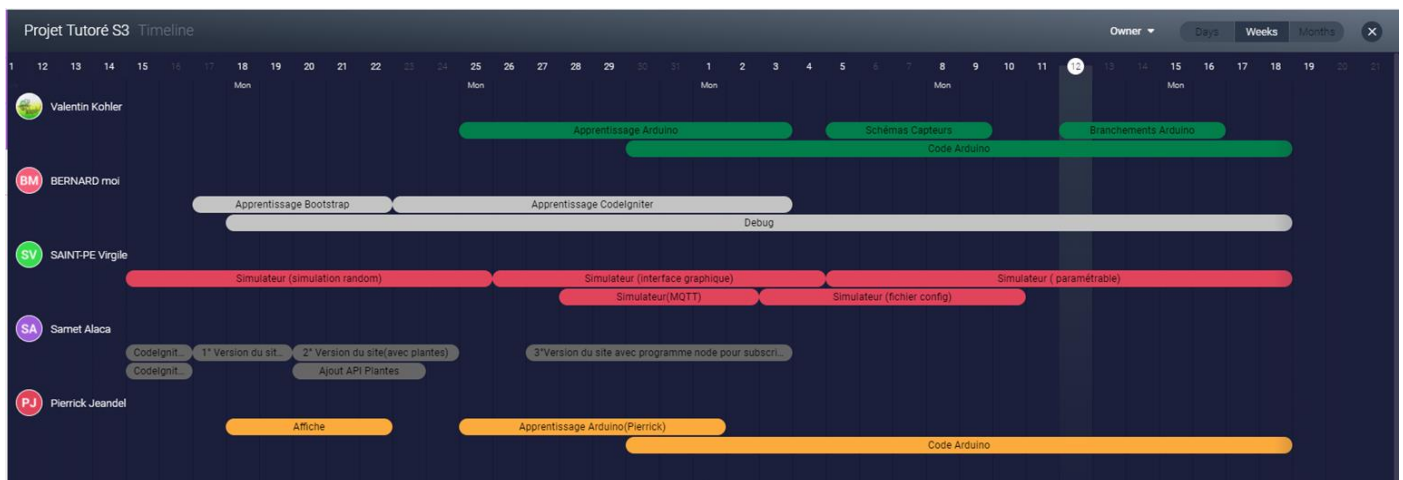
Nous nous sommes réparti le travail selon nos motivations à chacun. Certains cherchaient à s'améliorer et à pratiquer dans un langage qu'ils connaissaient déjà.

D'autres ont choisi de découvrir des langages et autres techniques qu'ils ne connaissaient pas.

La répartition des tâches a été effectuée dès la première séance avec le professionnel comme suit :

- Virgile s'est occupé de la partie simulateur.
- La partie WEB a été réalisée par Geoffroy et Samet.
- Le capteur a été réalisé par Valentin et Pierrick
- L'affiche a été réalisée par Pierrick et chaque membre du groupe a donné son avis sur comment l'améliorer

### 2)Diagramme de GANTT



### **3) Répartition des tâches prévues**

Notre répartition des tâches ainsi que notre diagramme de GANTT sont disponibles à l'adresse suivante : <https://truste-pot.monday.com/boards/71176830>

Voir aussi Annexes 1 et 2.

### **4) Comparaison**

Dans l'ensemble, les tâches que nous avons prévues ont été effectuées dans les temps impartis.

Quelques points ont cependant pris du retard, le code Arduino a en effet été un peu retardé puisque nous n'avions pas pu nous réunir pendant les vacances pour en discuter.

Le simulateur a pris également du retard car des difficultés ont été rencontrées lors de la mise en route de plusieurs simulateurs à la fois.

Pour tous les autres points, le travail a été effectué dans les délais prévus.

*Partie Gestion du Projet réalisée par Valentin Kohler*

### **III)Analyse du Sujet**

#### **1)Schéma général du projet**

Durant le premier rendez-vous avec notre responsable de projet nous nous sommes mis d'accord sur l'organisation générale du projet et les différents éléments que nous allions devoir créer et implémenter à notre solution.

Cette première séance était un brainstorming ou nous avons tous exposé et discuté nos idées. Nous avons ensuite réorganisé les idées que nous avons retenu dans un schéma qui, plus tard, a été refait sur ordinateur (voir le schéma dans la partie exemple d'utilisation).

Le projet allait donc être organisé de la manière suivante : Une application Web serait l'interface ou l'utilisateur pourrait créer son compte, puis se connecter pour gérer ses plantes. Une fois connecté, il peut ajouter des capteurs à son profil puis lier des plantes à ces capteurs, grâce à une recherche dans une base de données de plus de 48 000 plantes.

Après avoir associé une plante à un capteur, il peut paramétrer ce dernier : la fréquence à laquelle les données sont transmises, le type de notifications qu'il veut recevoir ( mail, sms ), si le capteur est en intérieur ou en extérieur, ...

A la fin de ce paramétrage, l'utilisateur est donc notifié si sa plante est dans de mauvaises conditions : via les données reçues par les capteurs si la plante est en intérieur, ou via les prévisions météo d'une API si la plante est en extérieur. Ces données sont comparées avec les meilleures conditions de vie de la plantes trouvées dans la BDD.

Les données reçues, sont archivées dans une autre BDD pour former un historique de données et pouvoir actualiser des graphes que l'utilisateur peut consulter.

Les capteurs ( humidité, luminosité, pH, température, sont programmés en Arduino ) et utilisent le protocole MQTT.

D'un autre coté un simulateur de données paramétrable doit être développé pour pouvoir faire des tests durant la phase de développement.

*Rédigé par SAINT-PE Virgile*



## **2)Simulateur de données**

Pour développer le simulateur nous avons choisi d'utiliser Java. C'est un langage qui nous est familier et avec lequel nous pensions pouvoir réaliser toutes les fonctionnalités dont nous allions avoir besoin.

Nous avons donc commencé par développer un simulateur qui puisse instancier des capteurs ( avec un id, des types de mesures, des fréquences d'envoi, ... ). L'interface graphique a été réalisée à l'aide du logiciel Scene Builder couplé au plugin JavaFX d'eclipse.

Pour envoyer des données par le protocole MQTT, nous avons du installer un broker MQTT ( mosquitto ), ainsi le simulateur publierait des données encodées en JSON sur un topic du server MQTT d'eclipse, et ces données seraient ensuite récupérables par l'application Web.

Nous avons très vite rencontré un problème : il était difficile de faire fonctionner plusieurs capteurs simultanément. Il était impératif qu'il soit possible de faire marcher plusieurs capteurs en même temps comme avec des vrais capteurs.

Pour remédier à ce problème nous avons décidé que l'interface graphique servirait seulement a générer un fichier de configuration encodé en JSON. Ainsi le simulateur serait lancé en ligne de commande via un terminal avec un fichier de configuration en paramètre. Cela résout le problème pour lancer plusieurs simulateurs à la fois.

D'autres fonctionnalités sont prévues dans la semaine à venir, comme l'ajout de nouveaux types de mesures personnalisées, et d'autres choix dans les variations de données.

*Rédigé par SAINT-PE Virgile*

## **3)Application Web**

Pour l'application web, nous avons décidé d'utiliser PHP, ainsi qu'un framework afin de simplifier les développements : CodeIgniter.

Nous avons choisi CodeIgniter car il est beaucoup plus simple et rapide à prendre en main que la plupart de ses concurrents, et suffisait pour notre projet.

L'application web utilise également beaucoup d'outils externes, toujours pour simplifier les développements. Nous utilisons notamment Bootstrap pour l'interface graphique.

L'application devait être capable de récupérer des informations venant de différents capteurs, cela impliquait une écoute sur socket, ce qui, dans notre cas et avec nos connaissances, n'était pas envisageable en PHP.

Nous avons donc résolu le problème en développant un petit programme NodeJS qui est en communication directe avec l'application web et qui se charge d'écouter les informations envoyés par les capteurs et les transfère à l'app.

#### **4)Capteurs**

Afin de pouvoir communiquer avec l'application web, nous avons besoin de coder un programme intégré aux capteurs. Ce programme devra utiliser Mosquitto pour envoyer les données vers un topic standard et prédéfini. Ces données devront être récupérées par le programme NodeJS et transférées vers l'application.

Pour effectuer nos tests et développer ce programme, nous disposons de :

- Une carte arduino ELEGOO UNO R3
- Les composants nécessaires aux capteurs
- Un breadbord ou plaque d'essai qui nous permettra de brancher nos composants
- Des fils de connexion

*Partie capteur rédigée par Valentin Kohler*

## **IV) Perspective et conclusions**

### **1) Perspective**

Notre professionnel nous a dit qu'il y avait le must have (ce qu'il faut impérativement) , et le nice to have (ce qui serait bien à avoir). Ce qu'il fallait impérativement ayant été défini très rapidement, nous avons eu le temps de penser à ce qui pourrait être intégré au site dans le futur, et donc de laisser la porte ouverte à plusieurs améliorations.

La plus importante était la création d'une application mobile. La développer entièrement "from scratch" une fois l'application web finie, semblait clairement non envisageable en termes de temps. Utiliser le framework Zend nous aurait permis de facilement transformer le site web en application mobile, mais cela demandait de laisser CodeIgniter de côté alors qu'il était déjà maîtrisé. Finalement nous avons opté pour l'utilisation de Apache Cordova qui nous permettrait éventuellement de créer l'application mobile une fois le projet fini en l'adaptant un peu.

D'autres améliorations sont mineures, par exemple l'utilisation de capteurs personnalisés qui permettraient d'enregistrer des variables définies par l'utilisateur, (eg. Si utilisation en extérieur, ajout de la vitesse du vent). Ou encore l'envoi des notifications, non pas par mail, mais par message (SMS), mais les seules solutions sont payantes et donc non envisageables pour l'instant. Cependant ces modifications sont facilement opérables même une fois le projet fini.

### **2) Conclusion**

En conclusion, le projet avait été placé en premier dans notre liste car il semblait intéressant et permettait d'utiliser des technologies sur lesquelles nous avons déjà des bases solides, mais présentait aussi des points sur lesquels nous devrions approfondir un peu.

Finalement le projet correspondait totalement à nos attentes et nous avons pu prendre un très bon départ. En effet notre équipe est très polyvalente et la répartition des tâches fût brève, les problématiques majeures ont toutes été abordées dès le premier rendez-vous avec le professionnel ce qui nous a permis de poser les bases le soir même.

Le rythme d'avancement était soutenu ce qui a permis aux rendez-vous suivant d'être tous utiles à l'avancement du projet tout en prenant le temps de poser les problématiques suivantes et les prochains objectifs à atteindre.

*Partie "Perspective et Conclusion" réalisée par BERNARD Geoffroy*

Annexe 1

Samet	1* Version du site(formulaire inscription)	SA	Done	Medium	7h	Dec 17 - 19
	2* Version du site(avec plantes)	SA	Done	Low	6h	Dec 20 - 24
	3*Version du site avec programme node pour subscribe aux topics	SA	Done	High	8h	Dec 27 - Jan 3
	Ajout API Plantes	SA	Done	Medium	5h	Dec 20 - 23
	Codeigniter(Configuration)	SA	Done	Medium	1h	Dec 15 - 16
	Codeigniter(Installation)	SA	Done	Medium	1h	Dec 15 - 16
	+ Create a New Pulse				28h sum	
Valentin	Apprentissage Arduino		Done	Medium	5h	Dec 25 - Jan 3
	Branchements Arduino		Working on it	High	2h	Jan 12 - 16
	Code Arduino		Working on it	High	14h	Dec 30 - Jan 18
	Schémas Capteurs		Done	Low	1h	Jan 5 - 9
	+ Create a New Pulse				22h	
Pierriick	Affiche	PJ	Done	Medium	4h	Dec 18 - 22
	Apprentissage Arduino(Pierriick)	PJ	Done	Low	4h	Dec 25 - Jan 1
	Code Arduino	PJ	Working on it	High	14h	Dec 30 - Jan 18
	+ Create a New Pulse				22h sum	

22h  
sum

Virgile

Owner	Status	Priority	Time Est.	Timeline
SV	Working on it	Medium	6h	Jan 5 - 18
SV	Done	High	4h	Jan 3 - 10
SV	Done	Low	5h	Dec 26 - Jan 4
SV	Done	Medium	6h	Dec 15 - 25
SV	Done	High	4h	Dec 28 - Jan 2
+ Create a New Pulse			25h sum	

Geoffroy

Owner	Status	Priority	Time Est.	Timeline
BM	Done	Low	5h	Dec 17 - 22
BM	Done	High	7h	Dec 23 - Jan 3
BM	Working on it	High	9h	Dec 18 - Jan 18
+ Create a New Pulse			21h sum	