

2018 - 2019

Projet d'intégration SensoryGarden SPRL

Groupe 1 IT :

Simon FAUCONNIER
Jonathan GOOSSENS
Adrien NINI PEREIRA
Benoit DE MAHIEU

Groupe MARKET :

Cassandra DUQUAINE
Nazanin HEYDARI
Yves HENRY DE GENERET

Référents :

L. VAN DORMAEL
A. DEWULF

Table des matières

1 Présentation du projet	1
1.1 Pourquoi ce produit ?	1
1.2 Board	1
1.3 Site Web	1
1.4 Notre projet à l'avenir	1
2 Qualité de réalisation	2
2.1 Description initiale du projet	2
2.2 Analyse du produit développé	2
2.2.1 Objectifs du projet	2
2.2.2 Nombre d'heures investies dans le projet	2
2.2.3 Produit final	2
2.2.4 Points forts/faibles	2
2.2.5 Améliorations envisageables	3
2.2.5.1 Board	3
2.2.5.2 API	3
2.2.5.3 Forum	3
2.2.6 Difficultés rencontrées	3
3 Aspects techniques et intégration	5
3.1 Arduino - Board - Proximus	5
3.2 Back-end	5
3.3 Front-end	5
3.4 Box	6
4 Environnement et méthodes de développement et de maintenance	7
4.1 Problématique	7
4.2 Solutions mises en place	7
4.2.1 Déploiement continu	7
4.2.2 Tests unitaires & d'intégration	7
4.2.3 Code quality & coverage	7
5 Méthodologie agile	8
5.1 Dynamique du groupe	8
5.2 Outils de communication	8
5.3 Utilisation de scrum	8
5.4 Avantage de la méthodologie scrum	9
5.5 Problèmes vs solutions	9
5.5.1 Développement	9
5.5.2 Production	9
5.5.3 méthode agile	9
5.5.4 communication avec le client	9
5.6 Compétences apportées	9
6 Sécurité	11
6.1 Analyse	11
6.2 Solutions mises en place	11
6.2.1 Virtual Private Server - VPS	11
6.2.2 Serveur Web	11
6.2.3 Base de données	12

6.2.4	Application Programming Interface - API	12
6.2.5	Failles classiques	12
6.2.6	GDPR	12
6.3	Améliorations futures	13
7	Approche entrepreneuriale	14
7.1	Cible visée	14
7.2	Marché de référence	14
7.3	Concurrents	14
7.3.1	Xiaomi Mi flora	14
7.3.2	Flower Power	14
7.4	Nos avantages	15
7.4.1	Connexion LoraWan ou WIFI	15
7.4.2	Site web + graphiques	15
7.4.3	Forum	15
7.4.4	Lieu de vente	15
7.5	Budget	15
7.5.1	Passifs	15
7.5.2	Actifs	15
7.6	Rentabilité	15
7.7	Frais - Revenus	16
7.8	Études qualitatives et quantitatives	18
7.8.1	Qualitative en B2B	18
7.8.2	Qualitative en B2C	18
7.8.3	Quantitative	18
8	Aspect RSE de l'entreprise	19
8.1	Accessibilité du site web	19
8.2	Green-it	19
A	Annexes	23

1. Présentation du projet

1.1 Pourquoi ce produit ?

Face au millier de plantes qui meurent à cause des personnes qui n'ont pas le temps de s'en occuper ou qu'ils l'oublient tout simplement. La société SensoryGarden propose la solution pour sauver toutes ces plantes. Le Sensory Captor va penser pour son propriétaire et va l'aider à prendre soin de sa plante.

Grâce à des graphiques, des tableaux et un forum, la plante parlera le même langage que son propriétaire. À lui maintenant d'écouter attentivement et de s'activer face aux pleurs envoyés par sa précieuse plante.

1.2 Board

Le Sensory Captor sera vendu dans des magasins spécialisés nature tel que Bepositive, Urban Gardener ou Dille et Kamille sur Bruxelles ou GreenPeas, Pop up shop ou SuperBien sur Louvain-la-Neuve.

Le Sensory Captor sera planté dans votre potager ou dans votre pot de fleurs. Grâce à sa taille et forme adaptée, il sera facile de le déplacer, le planter dans la terre et le manipuler sans crainte de le casser en retour.

Le Sensory Captor récupère des données telles que l'humidité de la terre, la température, l'humidité et la pression ambiante. Il capte également la luminosité et la qualité de l'air de la pièce. Il possède également en plus un relais qui permet de contrôler tous les outils électriques telle qu'une lampe UV, un arrosoir automatique ou autre chose selon les envies.

Une fois toutes ces données captées, elles sont envoyées dans notre base de données grâce au réseau LoraWan de Proximus. Ensuite les données sont utilisées et envoyées pour notre site web.

1.3 Site Web

Pour notre site web, nous avons implémenté toutes les idées du produit initial au niveau technologie. Nous avons mis en place notre plate-forme web pour l'utilisateur. Il peut y prendre contact avec l'équipe technique, il y a un accès rapide aux points de vente de notre produit ou alors différents liens vers nos pages Github, Trello, Facebook et Instagram.

Une fois inscrit ou connecté, l'utilisateur a accès à plusieurs options pour gérer ses différents capteurs. Il a à sa disposition des graphiques ou tableaux présentant les valeurs captées durant la semaine, la journée ou les dernières heures.

Pour finir, si une question lui vient à l'esprit, il accède à notre forum où des experts sont présents pour le conseiller face aux mesures obtenues par le Sensory Captor. Un système de catégories est mis en place pour permettre aux utilisateurs de préciser le thème de la conversation.

1.4 Notre projet à l'avenir

Au vu du travail des étudiants en marketing notre projet peut tenir la route avec la vente de nos Sensory Captors. Des magasins ont déjà donné leur accord pour vendre notre produit. De plus, les enquêtes montrent que le produit intéresse un nombre conséquent de personnes.

Malheureusement, nous n'avions pas le temps en douze semaines de construire un réel prototype mais nous avons cherché toutes les informations possibles à la bonne réalisation de celui-ci.

2. Qualité de réalisation

2.1 Description initiale du projet

Le but du projet est d'implémenter une serre connectée facile à utiliser. Le projet se scinde en deux grosses parties :

- La serre en tant que telle permettant, grâce à de nombreux capteurs, de surveiller ses plantations.
- Une plate-forme web servant à accéder aux relevés des capteurs mais aussi à de nombreuses ressources partagées par et pour les utilisateurs via un forum participatif.

Une web app permettra également d'accéder aux informations directement sur un smartphone.

2.2 Analyse du produit développé

2.2.1 Objectifs du projet

Au début de notre projet nos objectifs étaient assez simples et vagues. Grâce à trello nous avons scindé nos objectifs en différentes users stories. Cela nous a permis de définir le travail à effectuer pour chaque sprint. En début de sprint, nous définissions le nombre de points à effectuer et travail de chacun. Au début du projet nous faisions des daily scrum très régulièrement pour connaître l'avancement du travail.

Au fur et à mesure des sprints, nos objectifs ont évolué et nous avons amélioré notre manière de travailler. À la fin de chaque sprint, les objectifs étaient presque terminés sauf lorsque nous avons essayé d'augmenter notre vélocité.

Au final, nous pouvons être fier de notre projet. Nous avons réussi à atteindre nos objectifs de début de projet. Le travail fourni pour quatre informaticiens est conséquent et permet d'intégrer énormément d'anciennes et de nouvelles technologies.

2.2.2 Nombre d'heures investies dans le projet

Sur Trello nous avons mis en place l'extension TimeCamp. Elle permet de tracker le temps de travail de tous les membres du groupe. Grâce à TimeCamp, nous savons que nous avons travaillé exactement 313h30. Il faut ajouter à ça le temps des réunions que nous avons fait, donc 7h30 à ajouter. Au final nous avons au minimum un total de 321h pour ce projet avec une équipe de quatre informaticiens. C'est un minimum car de nombreuses reprises nous avons oublié de lancer le timer et les heures de documentation n'ont pas été comptée non plus.

2.2.3 Produit final

Après toutes ces heures de travail, nous sommes heureux de présenter notre produit. Face aux objectifs du début du projet, nous sommes plus que content face au produit final. La box est construite et fonctionne en accord avec le site web. Le tout est accessible avec la progressive web app et nous avons implémenté notre forum personnel. Même si la box doit être encore réfléchie, le site web est apporté avec de nombreuses fonctionnalités et est très complet aussi bien du point de vue de la sécurité ainsi que du dynamisme.

2.2.4 Points forts/faibles

Le gros point fort de notre équipe est l'implication de chacun dans ce projet. Nous avons distribué les tâches de chacun sans crainte sur la réalisation de celles-ci. Nous sommes aussi tous les quatre prêts à travailler pleinement pour obtenir un résultat parfait. Nous sommes également une équipe très diversifiée, une personne très douée dans le développement back-end, un autre front-end, le troisième expert arduino et le dernier spécialiste fusion 360.

Le point faible du groupe est le nombre d'informaticien qui travaillent dessus. Nous nous sommes répartis les tâches et donc il nous était difficile de pouvoir reprendre le travail d'un autre. Nous voulons aussi arriver à la perfection au final, ce qui peut poser problème dans l'avancement d'une tâche qui bloque toutes les autres.

2.2.5 Améliorations envisageables

2.2.5.1 Board

Les grosses améliorations de notre projet sont d'assurer la connexion entre le Sensory Captor et le site web. Ici nous utilisons LoraWan qui est lié à Proximus car nous avons reçu le matériel en prêt pour développer un prototype. Rien n'empêche à l'avenir d'avoir notre propre réseau de communication ou alors travailler avec du WIFI.

De plus pour la board, nous avons besoin encore de quelque temps pour proposer un produit final. Nous avons déjà regardé au prix des différents composants mais aucun contrat n'est fait pour le produire en grande quantité. À l'avenir, nous prendrons contact avec une entreprise spécialisée en électronique pour confectionner le produit final.

2.2.5.2 API

Nous pouvons améliorer l'API grâce à twilio. Cela permettra d'envoyer des notifications aux utilisateurs lorsqu'un problème survient. Nous avons constaté qu'il y avait une forte demande lors de nos enquêtes quantitatives et que ça apporterait un intérêt supplémentaire pour acheter notre Sensory Captor.

2.2.5.3 Forum

Pour le forum, nous avons plusieurs améliorations possibles.

La première permettrait à un utilisateur de supprimer ses posts. Pour le moment seul les administrateurs ont le droit de le faire. Cela va de pair avec la suppression de messages à caractère non approprié. Avec un grand nombre d'utilisateurs, des modérateurs seront désignés pour supprimer tous ces messages.

Ensuite, un système de like permettrait de mettre en avant les messages ainsi que les commentaires les plus appréciés et les plus utiles pour les utilisateurs de notre produit. De plus ces messages seront plus facilement accessibles via Google si une personne ne connaît pas encore notre site.

Pour faire le lien entre les graphiques et le forum, nous pouvons implémenter une méthode permettant de partager rapidement et proprement les résultats obtenus par un Sensory Captor sur le forum. Cela favoriserait le partage et éviterait des désagréments aux utilisateurs.

Pour finir, un système d'avatar peut être mis en place. L'utilisateur se sentira plus représenté sur notre forum. Il sera également plus reconnaissable par d'autres utilisateurs.

2.2.6 Difficultés rencontrées

Les grosses difficultés rencontrées dans ce projet sont dues à enco et OVH. Tout d'abord nous avons dû implémenter les cloud engine à la place des cloud channel. Ceux-ci n'étaient plus maintenus et nous n'étions pas au courant. Certains jours la board envoyait des données mais enco ne les recevaient plus. À cause de cela, les données n'étaient pas envoyées sur notre site. Avec OVH, nous avons eu des problèmes de performance et donc notre site web était down durant toute une journée.

Ensuite nous avons rencontré des problèmes avec l'impression 3D. Une fois les plans finis, l'impression a complètement raté. En effet, l'imprimante s'est complètement bouchée et les autres imprimantes présentes au MakiLab ont décidé de ne plus fonctionner... Heureusement nous avons pu rebondir et faire une découpe laser afin d'obtenir une jolie boîte en bois. Nous devons par contre traiter cette boîte pour qu'elle supporte les arrosages.

Nous avons également eu quelques difficultés au niveau du site web. Au départ, nous comptions intégrer un lecteur de QR Code directement dans l'application à l'aide de librairies comme "instascan.js", "qr-scanner.js", "jsQR.js" et "jsQRScan.js". L'implémentation de ces librairies a été faite mais aucune ne fonctionnait correctement sur IOS. Nous avons donc abandonné cette idée étant donnée qu'Apple représente une part de marché non négligeable.

La page des graphiques fut longue à mettre en place étant donné qu'elle est entièrement dynamique. Chaque action déclenche une multitude d'événement à gérer. Heureusement, la communauté de la librairie "chart.js" est très active et de nombreuses informations sont disponibles sur le net.

Lors de la réalisation du forum, l'utilisation du "select2" a posé énormément de problèmes... Dès qu'une modification de style était faite, tout le "select2" bougeait... Nous avons quand même décidé de le garder parce qu'il est très pratique. Nous nous sommes justes adapté à sa versatilité.

La librairie pour écrire en MarkDown a été changée à plusieurs reprises car elles n'étaient pas toutes responsives, elles ne permettaient pas de faire tout ce que l'on désirait ou n'était pas compatible avec tous les appareils... Nous nous sommes finalement fixé sur l'une d'elle qui est plutôt efficace !

3. Aspects techniques et intégration

3.1 Arduino - Board - Proximus

Pour créer un Sensory Capteur, nous utilisons un kit reçu par l'EPHEC. Celui-ci se programme comme un arduino et est lié avec Proximus. Nous avons branché différents capteurs sur la board. Ceux-ci récupèrent les informations toutes les deux heures et les envoient sur le site de Proximus. Ensuite, grâce à des cloud engine et l'API que nous avons mis en place, les informations sont envoyées jusqu'à notre site internet <https://sensorygarden.be>.

Pour la programmation arduino, les étapes sont assez simples :

- Initialiser les sensors et ouvrir le bus vers Proximus. Cette étape est importante car elle indique si les pins sont en entrée ou sortie.
- Toutes les deux heures, lire les valeurs de chacun des capteurs
- Envoyer toutes ces données du côté de Proximus.

Évidemment il faut ajouter quelques librairies fournies par AllThingsTalk, la société qui est à l'origine des kits que nous avons reçus à prêter.

3.2 Back-end

Pour mettre en place un système complexe comme celui de ce projet nous avons besoin d'une bonne infrastructure solide pour que tout fonctionne ensemble. Nous avons plusieurs technologies qui doivent s'intégrer ensemble : Des modules IoT doivent avoir toutes leurs données centralisées, ainsi elles pourront ensuite être redistribuée sur une plateforme web.

- Nous avons choisi un VPS chez OVH pour leur fiabilité, rapidité et facilité d'installation.
- Une API (Application Programming Interface) sert d'interface entre nos modules IoT et notre base de données. Cette API a été développée en Ruby avec le Framework Ruby on Rails.
- Nous avons installé une base de données MySQL.
- Nous avons configuré un serveur mail qui tourne dans un container docker. Ce serveur mail nous permet d'avoir des adresses emails professionnelles pour le projet. Nous l'utilisons aussi en tant que relay SMTP pour envoyer des emails depuis notre API.
- Nous utilisons le serveur DNS de OVH pour gagner du temps et faciliter la configuration. Nous avons ajouté tous les records nécessaires au bon fonctionnement de notre serveur Mail (MX, SPF, DMARC, DKIM). Cela nous a permis d'avoir un score de 10/10 sur le site mail-tester.com (voir figure A.4 en annexe)

3.3 Front-end

La plateforme web a été réalisée en HTML5, CSS3 et Javascript ainsi que les librairies listées ci-dessous :

Page principale :

- "Jquery" (v 3.3.1) - Nous avons utilisé cette bibliothèque Javascript car elle est rapide et riche en fonctionnalité. De plus, de nombreux plugins offrant des fonctionnalités supplémentaires sont développés par la communauté.
- "Waypoint.js" - Ce plugin permet de lancer des événements spécifiques lorsque l'utilisateur atteint un point particulier de la page en scrollant. À la base, nous voulions l'utiliser pour réaliser des animations, une demande de l'équipe marketing, mais suite à la conférence green IT, où cette pratique était déconseillée, nous avons décidé d'abandonner l'idée des animations lourdes et dommageables pour l'environnement. Dans la version de production, nous utilisons ce plugin pour mettre à jour le menu en fonction de la position de l'utilisateur dans la page.

Page personnelle de l'utilisateur :

- "BigSlide.js" - Cette librairie très légère (1kb) gère le menu hamburger pour la version mobile de la page utilisateur.
- "Charts.js" - Cette librairie est essentielle pour notre projet. Elle permet de générer des graphiques en fonction d'un set de données. Nous avons choisi cette dernière car elle est très utilisée par la communauté, il est donc très facile de trouver des informations sur son fonctionnement.
- "Moment.js" - "Moment" permet de gérer les dates et est utile pour la page des graphiques. En effet, les données des dates sont stockées dans un autre fuseau horaire dans notre base de données, "moment" s'occupe de faire la conversion et permet de formater les dates en français facilement.
- "datepicker.js" - Au début, nous étions simplement utiliser le tag HTML <input> de type date mais ce dernier n'est pas supporté par Safari (IOS/OSX).
- "qrcode.js" - Cette librairie nous permet de générer des qrcode facilement.

Forum :

- "select2.js" - Cette librairie permet de créer des sélecteurs très particuliers avec un style bien à eux. Nous pouvons, grâce à la librairie, créer des select à choix multiple pour un filtre plus optimisé sur les différentes catégories du forum.
- "jPages.js" - Cette librairie est très simple et permet de gérer le système de pagination du forum. Nous avons choisi d'utiliser cette librairie car elle est très facile d'utilisation.
- "marked.js" - Cette librairie nous permet de transformer le MarkDown écrit lors de la création de publications ou de commentaire.
- "tail.writer.js" - Cette librairie nous permet d'avoir un système de MarkDown lors de la création d'une publication. La textarea sera modifiée afin d'avoir un entête qui permet d'ajouter des caractères de markdown.

Progressive Web App (PWA) :

Nous avons décidé pour ce projet de ne pas faire d'application native mais d'opter pour une PWA, qui est une nouvelle technologie principalement développée par Google. Cela nous permet de développer une seule plateforme en gardant la plupart des avantages d'une application native. Pour nous, cela nous simplifie le développement, et pour l'utilisateur, l'expérience est la même qu'une application native.

- "Cache API", est une API native de JavaScript qui permet le stockage d'assets dans la cache du navigateur. Cela nous permet de stocker les données statiques du site pour que l'utilisateur puisse y accéder hors-ligne.
- "Service Worker", est une technologie développée par Google. Le principe est assez simple, c'est du code JavaScript qui tourne en arrière plan côté client. Il se place entre le navigateur du client et le serveur, cela nous permet d'intercepter les requêtes faites au serveur et de les rediriger vers la cache. C'est ce qu'on appelle une optique Offline-First.
- "Fetch API", est une API native de JavaScript, utilisée pour effectuer des requêtes HTTP asynchrones. Nous l'utilisons principalement dans le Service Worker, le principe est le même qu'avec l'AJAX mais il fonctionne sur un système de promesses (Promises) à la place des callback.

3.4 Box

Pour protéger notre système des intempéries, nous avons décidé de réaliser un boîtier en 3D. Nous étions tout d'abord parti sur de l'impression en 3D, les schémas ont donc été réalisé à l'aide de Fusion 360. Le MakiLab a été de très bon conseil dans la réalisation de ce projet. Différents prototypes d'impression ont été réalisé. Ensuite, nous avons imprimé la Box finale mais l'impression a échoué et les imprimantes se sont bloquées. Nous nous sommes rabattu sur de la découpe Laser, les schémas ont donc été réalisé à l'aide d'InkScape, un petit logiciel de dessin qui ressemble un peu à Paint... Tout s'est déroulé au mieux, nous avons ensuite collé les faces ensemble et notre Board pouvait rentrer dans le boîtier.

4. Environnement et méthodes de développement et de maintenance

4.1 Problématique

L'une des principales problématiques de ce projet est que nous ne sommes que 4 étudiants à travailler sur la technique. Nous devions donc optimiser au mieux notre temps pour espérer produire autant de travail que les autres équipes.

Pour nous aider à atteindre nos objectifs, nous avons mis en place plusieurs outils qui ont amélioré le développement et l'intégration des différentes technologies que nous avons utilisé.

4.2 Solutions mises en place

4.2.1 Déploiement continu

Nous avons mis en place un pipeline de déploiement continu avec Travis-CI et capistrano pour notre API. Ce système fut long et compliqué à mettre en place (environ 30h sur le premier et second sprint) mais il nous a permis d'accélérer le développement de notre API. Cela nous a permis de nous concentrer à 100% sur le développement sans se soucier du déploiement et de la migration des différentes version de la base de donnée.

4.2.2 Tests unitaires & d'intégration

Toujours dans le cadre du développement de notre API, nous avons appliquée une méthodologie TDD (Test Driven Development). Nous avons donc environ 130 tests d'intégrations pour tous les endpoints de notre API. Ces tests nous ont permis de réduire le temps de débug mais aussi d'améliorer la qualité de notre code. Ces tests sont intégrés à notre pipeline de déploiement, ainsi notre API n'est déployée que si et seulement si tous les tests sont verts. Nous avons utilisé le Framework RSpec pour écrire ces tests.

4.2.3 Code quality & coverage

Nous avons ajouté à notre pipeline un système de tests de qualité et de couverture. Ce système nous donne un rapport complet sur la "santé" de notre code, il nous indique différentes métriques nous donnant des informations sur la qualité du code (Complexité, répétition de blocs de code, code inutile) ainsi que des informations sur du code qui n'est pas couvert par les tests unitaires.

5. Méthodologie agile

5.1 Dynamique du groupe

Notre groupe plus réduit que les autres a obligé chacun à travailler sur des tâches différentes à accomplir lors de chaque sprint. Nous nous sentons tous indispensable à la réalisation du projet et voir les autres travailler motive grandement les troupes. Nous pouvons plus facilement nous réunir pour discuter de notre avancement. À chaque réunion, nous sommes tous présent et chacun peut s'exprimer librement.

Dès le sprint 1, nous avons distribué les tâches séparément et durant tout le projet les tâches sont restées restreintes pour chacun. C'est une difficulté du groupe réduit mais comme tout le monde était motivé et présent à chaque fois, nous n'avons eu aucun problème pour continuer de cette manière.

5.2 Outils de communication

Un point fort de notre groupe est la communication entre nous, étudiant d'informatique. Nous avons utilisé Facebook évidemment mais en plus le système d'issue sur github. Pour les tâches à faire ou accomplie, nous avons utilisé trello. Nos codes sont mis sur github régulièrement.

Avec les étudiants en marketing, nous avons mis un groupe Facebook pour partager les documents avec eux. De plus une conversation WhatsApp permet de discuter avec eux. Cette association entre les étudiants de marketing et nous n'a pas été facile. Les rythmes de travail, la communication, l'investissement ou les présentations sont différents de part et d'autre. Chaque groupe travaille de manière différente et se complète du mieux que nous pouvions.

5.3 Utilisation de scrum

Nous avons attaqué ce projet en utilisant scrum. Nous avons choisi un product owner, un scrum master et les autres sont l'équipe de développement. Naturellement pour ce projet le product owner et le scrum master développaient également. Nous avons joué le jeu avec les marketings en les considérant comme des clients qui désiraient un produit final mais le manque de communication à empêcher le bon déroulement du travail. Nous n'avions aucune demande particulière et aucun retour non plus.

Le projet a été divisé en cinq sprints de deux semaines. Dès le premier sprint, nous avons mis en place notre github, trello, définis nos users stories. Chacune avec une description expliquant son utilité. Ce premier sprint a été primordiale pour le bon déroulement du projet car c'est à celui-ci que nous avons installer toutes les structures du projet.

À la fin de chaque sprint, nous nous réunissions pour une rétrospective. Durant cette réunion nous prenions en compte les remarques des enseignants, de nos coéquipiers de marketing et autres idées reçues des autres étudiants. Directement après nous attaquions la réunion pour le sprint suivant, nous y définissions les tâches à accomplir et les points à leur attribuer.

Durant tout le projet, nous avons pris du temps pour faire des séances de travail tous ensemble au CHE². Cela a permis de créer une bonne cohésion dans le groupe. De plus, à de nombreuses reprises, nous avons mis en place du pair programming. Deux informaticiens sur le même ordinateur. Cela permettait de réfléchir sur une problématique et d'être plus performant. Comme outils que nous avons utilisé :

- Product backlog/sprint backlog qui est fait dès le départ du sprint 1. Même s'il a été modifié durant le projet, nous avions une ligne directrice durant ce projet ou durant chaque sprint.
- Github qui a permis de faire du versionning. Chaque personne pouvait travailler sur sa version sans crainte de casser quelque chose chez les autres. En fin de sprint nous remettons tout ensemble et réglons les conflits. Le résultat était mis en production uniquement après cette résolution grâce à un système de déploiement continu expliqué plus haut.

- Trello qui a été utile pour la bonne démarche de ce projet. Grâce à celui-ci, nous savions le travail à accomplir chaque sprint. Nous connaissions aussi les tâches les plus importantes et les plus longues.
- Planning poker qui nous a servi pour noter les tâches à accomplir. Nous donnions des points à chaque tâche donnant une idée du temps qu'il fallait pour la terminer. Bien que très utile, nos estimations n'étaient pas toujours correct. De plus cet outil nous permettait de garder une vitesse minimum à chaque sprint qui était de 24 points.

Un outil que nous n'avons pas utilisé bien connu est le burn down chart. Pour ce projet nous n'avons pas pris le temps de définir des deadline pour les tâches. Comme nous étions largement limité dans le temps, nous avons préféré entamer rapidement le projet avec la fin de sprint comme deadline. De plus nos sprints étaient de deux semaines et nous voyions facilement l'avancement des tâches sur Trello.

5.4 Avantage de la méthodologie scrum

La méthode scrum nous a permis de structurer notre projet du début jusqu'à la fin. Les rôles définis au début ont lancé tout le monde dans nos tâches respectives et chaque fin de sprint s'est bien déroulé pour chacun. Cette méthodologie est une super méthode de travail pour une équipe IT. Elle pousse chaque membre à terminer son travail dans un délai assez court. De plus si un soucis arrive les autres membres peuvent sans problème l'aider pour faire avancer le projet. Petit bémol face à cette méthode, il faut que le client soit présent pour faire avancer le projet et ait une idée de ce qu'il veut avant de le lancer.

5.5 Problèmes vs solutions

5.5.1 Développement

Au tout début de ce projet, nous avions des problèmes pour le développement du site web. Nous n'avions qu'une seule branche développement donc dès qu'une personne faisait des changements, il y avait un risque d'effacer ce qu'un autre avait fait. Pour régler ce problème nous avons mis en place un serveur uniquement de développement avec des sous-espaces pour chacun. Avec cela nous avons créé les branches correspondantes sur github et résolu le conflit.

5.5.2 Production

Dans le premier sprint, la mise en production stressait l'équipe, nous n'étions pas sûrs que tout allait bien fonctionner. Pour résoudre ce problème, une fois que les soucis de versionning sont résolus, nous mettons les fichiers sur le staging pour faire toutes nos vérifications. Une fois que tous les tests sont passés, nous mettons le site officiel à jour. Nous savons que tout fonctionne correctement après cela. Pour automatiser et simplifier notre déploiement en staging et production, nous avons mis en place un système qui envoie automatiquement les fichiers en SFTP sur le serveur. Cela nous permet de ne plus devoir nous connecter à notre serveur pour une mise en production. Cela nous évite aussi des problèmes de droits sur les fichiers sur notre serveur de production.

5.5.3 méthode agile

Pour le premier sprint également, nous n'avions pas encore imprimé le planning poker donc nous avons travaillé durant ce sprint sans comptage de notre vitesse. Par chance nous avons terminé toutes les user stories mais nous manquons d'informations sur le nombre de points accomplis.

5.5.4 communication avec le client

Durant tout le projet, nous manquions de retour de la part des étudiants en marketing pour nous aider dans notre travail. Tout le travail effectué était accessible via notre site web. Pour remédier à ce problème, nous avons organisé des réunions chaque lundi, laissant l'opportunité aux étudiants de marketing de donner leurs avis et conseils. Cela a permis d'avoir un retour même si ce n'était pas suffisant.

5.6 Compétences apportées

Avec ce projet, nous étions vraiment une équipe d'informaticiens au sein d'une entreprise et nous appliquions la méthodologie scrum du mieux possible. Cela représentait vraiment la vie active que nous allons avoir dans deux mois. Nous avons observé tous les points positifs et négatifs ainsi que les changements personnels que nous pouvions apporter avec la personnalité de chacun. Les rôles de product owner et scrum master ont permis à

certains de travailler sur une autre facette. Ils n'étaient pas juste présent pour programmer mais également dans la gestion du projet.

6. Sécurité

6.1 Analyse

Notre projet intégrant plusieurs technologies, nous avions plusieurs aspects dont nous devions tenir compte.

Tout d'abord, l'un des points les plus sensibles était le VPS. Ce serveur est le point central où toutes les informations de nos utilisateurs passent et sont stockées. Il était donc très important d'avoir le moins de portes d'entrées sur le serveur.

Ensuite, il était important que les utilisateurs se sentent en sécurité sur notre site. Il nous tenait à cœur que les données de nos utilisateurs ainsi que leurs actions soient les plus sécurisées possible.

Enfin, le transfert des données récoltées par les capteurs ne doivent pas être corrompues ou récupérées par une tierce personne.

Pour répondre à ces problématiques, nous avons mis plusieurs solutions en place.

6.2 Solutions mises en place

6.2.1 Virtual Private Server - VPS

Le VPS, étant le points sensible du projet, nous avons tout de suite attaqué sa sécurisation dès le premier sprint. Nous avons suivi la procédure suivante :

- Première connexion en root.
- Création d'un premier utilisateur (non sudoer) avec un mot de passe.
- Envoi d'une clé publique pour cet utilisateur puis vérification connexion SSH via cette paire clé privé/clé publique.
- Configuration du serveur OpenSSH, ce qui comprend : la désactivation de connexion pour l'utilisateur Root.
- Promotion de ce premier utilisateur en sudoer.
- Création de comptes (non sudoers) pour les autres utilisateurs avec un mdp provisoire.
- UNIQUEMENT lorsque les mots de passes de ces utilisateurs ont été changé ET que les clés publiques sont sur le serveur, désactivation de la connexion par mot de passe sur le serveur et promotion des utilisateurs en sudoers.
- Installation et configuration de Fail2Ban sur le serveur OpenSSH : 3 échecs sur 10 minutes = 10 minutes de ban, 6 échecs en 1h = 10 heures de ban, 9 échecs en 3 jours = 10 jours de ban.

Cette configuration basique nous permet de nous prémunir contre les attaques les plus courantes, par exemple le bruteforce de mot de passe. La configuration de plusieurs "prisons" (jails) Fail2Ban nous permet de réduire le trafic attaquant notre serveur et donc de réduire les baisses de performances. Ce point d'entrée que nous savons maintenant sécurisé sera le seul moyen d'interaction (ligne de commande) que nous aurons avec le serveur. Tous les transferts de fichiers que nous opérerons se feront par le biais de cette connexion (SFTP).

6.2.2 Serveur Web

Le serveur web est l'interface entre le client et les données stockées sur le serveur. Il était donc important que les communications soient sécurisées au mieux pour que le client soit le plus en confiance lors de l'utilisation de nos outils.

Pour cela, tous nos serveurs web sont sécurisés via HTTPS et utilisent la version de protocole HTTP/2. Notre infrastructure se compose d'un reverse proxy nginx qui va servir les différentes ressources statiques du site web ainsi que les données dynamiques venant de notre API.

Notre serveur a été configuré via l'outil certbot qui se charge de demander les certificats via letsencrypt et d'adapter la configuration de nginx pour HTTPS. Ceci nous a permis d'obtenir un score A sur le site ssllabs.com (voir figure A.3 en annexe).

6.2.3 Base de données

La base de données est aussi un point extrêmement sensible, elle contient toutes les informations relatives à nos clients ainsi que les données liés à l'utilisation de nos produits.

Notre base de données est configurée pour n'être accessible que localement sur notre serveur, il est donc impossible de communiquer avec depuis l'extérieur. Nous avons donc travailler sur des bases de données de test et de développement localement sur nos machines et ensuite déployé les schémas via notre système de déploiement continu. Ainsi la seule façon de récupérer et manipuler des données est via notre API ou en ligne de commande en étant directement connecté au serveur (SSH).

Les mots de passes utilisés pour se connecter à la base de données sont stockés dans des variables d'environnement sur le serveur et ne se retrouve nulle part dans le code ou dans notre système de versionning.

6.2.4 Application Programming Interface - API

Pour ce projet, nous avons choisi de développer une API pour plusieurs raisons, l'une d'elle est de séparer complètement la partie front-end (utilisateur) et le back-end (système). De cette manière toutes les interactions avec la base de données sont clairement définies et limitées par les endpoints de l'API. Il devient alors difficile de faire des attaques de type injection SQL.

Mais pour empêcher n'importe qui d'accéder à n'importe quel endpoint de l'API, nous avons mis en place un système de token d'authentification ainsi qu'un système de droit. Ce token d'authentification est ce qu'on appelle un JSON Web Token (JWT) qui permet une authentification stateless. Ce token est valide pour une durée de 24 heures et contient des informations relatives à l'utilisateur, notamment son id et son niveau de droit. Cela nous permet plusieurs choses, d'une part avoir ces données à portée de main coté client pour effectuer des affichages dynamiques selon les droits par exemple. D'autre part, pour pouvoir identifier si l'utilisateur tente de récupérer des données aux-quelles il ne doit pas avoir accès. Les JWT étant stateless, le serveur ne garde pas en mémoire qu'un utilisateur est connecté, donc ces informations (contenues dans le token) nous permettent d'identifier l'utilisateur et de vérifier que ce qu'il demande ne dépasse pas les droits qui lui sont donnés.

Le fonctionnement des JWT permet aussi de vérifier que le token n'a pas été modifié par l'utilisateur. Cette vérification se fait via une signature unique se trouvant dans une variable d'environnement sur le serveur. En effet, les JWT sont encodés en base64 et donc facilement décodable via le site jwt.io, l'utilisateur peut donc voir les informations qu'il contient et les modifier (raison pour laquelle aucune information sensible n'y est stocké), mais cela entraînera une erreur lors de la vérification de la signature et donc un refus de l'API.

Au niveau de la gestion des mots de passes, nous utilisons la librairie "Knock" qui nous génère les JWT. Cette librairie utilise la propriété "has_secure_password" du framework Ruby on Rails qui s'occupe du hashage des mots de passe ainsi que de l'authentification de l'utilisateur. Cela nous permet de ne pas nous occuper de ces fonctions sensibles relatives au stockage, hashage et vérification des mots de passes.

6.2.5 Failles classiques

Lors d'une première version du forum sur notre site, nous nous sommes rendu compte de la présence d'une faille XSS assez grave. Suite à cette découverte nous avons changé le fonctionnement de notre API pour la partie forum pour éviter que les utilisateurs puissent injecter du code qui serait ensuite exécuté. Cette faille fut rapidement réglée avec au final peu de code mais cela nous a permis de nous rendre compte de l'importance de tester tous les cas et surtout ceux qui ne devraient jamais arriver.

Nous voulons aussi éviter qu'un utilisateur "vole" virtuellement le capteur d'un autre utilisateur. Pour s'en prémunir nous avons défini au niveau de notre base de donnée qu'un capteur ne peut être assigné qu'à un et un seul utilisateur à la fois. Donc pour éviter qu'un utilisateur distrait se trompe dans le numéro de série lors de l'enregistrement de son capteur ou qu'un utilisateur mal veillant s'approprie le capteur de quelqu'un d'autre, nous avons mis en place un système de code de vérification similaire à celui que l'on peut voir sur des cartes de banques ou le numéro de registre national. Nous avons développé cet algorithme nous même et seuls les administrateurs du site pourront générer des codes de vérification.

6.2.6 GDPR

Concernant la protection des données de l'utilisateur, nous avons programmé notre système de façon à supprimer toutes les données relatives à un utilisateur suite à la suppression de son compte. Cela nous permet deux choses : la première est de se débarrasser de données inutiles dans notre base de données pour faire de la

place, la seconde est une question d'engagement que nous avons pris envers nos utilisateurs de respecter leur privacité digitale ainsi que le droit à l'oubli.

6.3 Améliorations futures

En terme d'améliorations futures, on peut dire qu'aucun système n'est jamais parfait et n'est jamais sécurisé à 100%. Néanmoins quelques détails peuvent être amélioré dans notre système actuel.

Premièrement, il faudrait mettre en place un système de backup automatisé au niveau de notre base de données. Pour le moment, nous sommes sans filet à ce niveau et si nous avions le malheur d'avoir un problème qui causerait la perte de toutes nos données, nous n'avons pas de moyen simple de les récupérer.

Deuxièmement, notre base de données devrait idéalement être stocké sur un serveur physique différent que le reste de notre infrastructure. Cette séparation permet de diminuer les risques de fuites de données en cas d'infiltration sur le serveur web.

Troisièmement, il faudrait mettre en place de la redondance au niveau de nos serveurs, pour le moment, nous n'avons qu'un seul VPS qui tourne avec notre système, si il plante, tous nos services sont inaccessibles. Nous devrions avoir au moins un autre serveur, identique au premier, qui prendrait le relais si le premier vient à planter.

En conclusion, n'importe quel système peut toujours être amélioré, néanmoins nous avons essayé de sécuriser au maximum notre système dans les limites de nos moyens pour que l'expérience de nos utilisateurs soit la plus paisible et sécurisée possible.

7. Approche entrepreneuriale

7.1 Cible visée

Notre première cible sont des jeunes adultes entre 25 et 45 ans. Un public pouvant se permettre d'acheter un outil comme celui-la. Nous touchons surtout les personnes qui aiment cultiver leurs propres légumes, fruits ou autres. De plus, ces personnes aiment utiliser la technologie dans leur vie de tous les jours. Ici ils pourront combiner les deux en même temps.

Ensuite ce produit va viser les personnes voulant changer leur alimentation avec des produits génétiquement modifiés par des aliments cultivés directement dans leur jardin.

Pour finir, nous nous concentrerons d'abord sur la clientèle belge. Avant d'attaquer un marché étranger, il est important de voir si le produit fonctionne en Belgique. Ce pays possède une culture très ouverte et variée. Si le produit séduit, rien nous empêche de dépasser nos frontières pour offrir aux personnes notre aide.

7.2 Marché de référence

Avec notre produit nous sommes clairement sur un marché du jardinage d'intérieur. C'est une pratique assez tendance actuellement avec le "Do it yourself" et le "eco-friendly". Notre produit s'installe également avec la tendance du circuit court. Le consommateur connaît d'où viennent les aliments qu'il mange. Notre produit fonctionne pour la favorisation du bio. Si la personne veut améliorer son alimentation, il ne va sûrement pas la polluer avec des produits nocifs.

7.3 Concurrents

7.3.1 Xiaomi Mi flora

- Le Xiaomi Mi Flora donne les informations tel que quantité d'eau à donner, qualité de la terre, la température et l'ensoleillement de la plante.
- L'appareil coûte 27,50 €.
- L'application se télécharge sur smartphone. Les données sont envoyées via Bluetooth et l'appareil fonctionne avec des piles.
- Xiaomi Mi Flora s'achète uniquement en ligne et disponible pour le territoire belge. Cette marque ne possède aucun site internet officiel.

7.3.2 Flower Power

- Flower Power est un capteur qui surveille en temps réel la santé des plantes elle est liée à une application gratuite disponible sur smartphone.
- L'appareil coûte 29,99 € et mesure la température, la luminosité, l'engrais et l'humidité. Elle permet de suivre plusieurs plantes à la fois, à l'intérieur ou à l'extérieur.
- Les données sont envoyées via Bluetooth et l'appareil fonctionne avec des piles.
- Flower Power s'achète uniquement en ligne et est disponible sur le territoire belge.
- La marque Parrot possède un site officiel.
- La production a été arrêté en 2017 mais le produit est encore disponible sur Amazon.

Ces deux produits existent également en pot connecté mais ne se rapprochent pas de notre produit au niveau du design.

7.4 Nos avantages

7.4.1 Connexion LoraWan ou WIFI

Premier point important c'est la connectivité de notre Sensory Captor. Contrairement aux autres concurrents, le nôtre fonctionne avec le réseau LoraWan ou avec du WIFI. Cela permet à l'utilisateur de récupérer des données à plus grande distance. Pas besoin d'être juste à coté pour voir les résultats. Nous avons soit le choix du réseau LoraWan qui permet de mettre le capteur partout en Belgique, cela favorise notre clientèle mais augmente le prix, le WIFI permettra de diminuer le prix mais la zone du réseau est restreinte.

7.4.2 Site web + graphiques

Le second point est notre site web accessible pour les personnes souffrant d'un handicap. Il est facile d'utilisation et d'accès. Les graphiques présentant les données sont très pratiques pour comprendre la plante. L'utilisateur peut prendre contact avec l'équipe technique directement sur le site. Il peut voir également les différents points de vente de notre produit sur une map directement accessible sur notre site.

7.4.3 Forum

Troisième point indispensable au bon fonctionnement, c'est notre forum ! Lorsqu'un utilisateur a une question, il peut aller sur notre forum où des experts, d'autres utilisateurs du Sensory Captor ou alors un étranger pourront le conseiller. Ce point était très important pour nous. Nous voulions permettre à une personne ne connaissant rien d'avoir la chance de changer ses habitudes.

7.4.4 Lieu de vente

Dernier point en opposition aux autres, notre produit sera en vente uniquement en magasin pour commencer. Nous avons trouvé plusieurs magasins prêts à vendre plusieurs de nos Sensory Captor. De cette manière, les consommateurs pourront visualiser notre produit directement et voir son efficacité.

7.5 Budget

7.5.1 Passifs

Pour ce projet, le gros coût sera pour la création de l'entreprise. Nous y comptons les frais de la création de l'entreprise, les frais de lancement et tous les frais liés aux taxes, transports, assurances et autres.

Second coût mais qui sera réinvestit ce sont les Sensory Captors. Il faut acheter en grande quantité pour faire diminuer le prix de chaque composant et également de la conception du produit final.

7.5.2 Actifs

Pour réaliser ce projet, nous avons besoin d'un financement de départ. Tout d'abord, il y a des apports personnels. Cela permettra de lancer notre entreprise sous forme de SPRL. Le montant demandé est de 18550 €.

Ensuite nous nous sommes inscrits sur une plateforme de crowdfunding **KissKissBankBank**. Nous désirons arriver à un montant de 15000 €.

Le gros point de notre actif viendra de la vente des Sensory Captors. Nous avons effectué des enquêtes qualitatives pour estimer le nombre de capteurs que nous pourrions vendre dans les différents magasins.

7.6 Rentabilité

Après plusieurs calculs effectués par nos étudiants en marketing, nous pouvons dire que notre projet est viable assez rapidement. Notre avantage, aucun besoin d'investisseurs externes autre que par crowdfunding.

Avec un montant de 33550 €, nous pouvons couvrir tous les frais de création et nos investissements dès le premier mois. De plus nous sommes rentables dès la première année ce qui est rare pour une nouvelle société.

Bien que tout nous donne raison dans ce projet, nous devons garder un oeil sur la vente des Sensory Captors. Le taux de vente doit respecter la courbe fixée par nos calculs.

7.7 Frais - Revenus

1^{ère} année :

	Frais fixes annuel HTVA	TVA	Frais variables unitaire HTVA
Carburant (41,32€/mois)	495,84€	21%	Prix d'achat 7,12 €
Contrôle technique	26,20€	21%	Etiquette 0,99 €
Taxe de circulation	193,91€	/	Packaging 0,88€
Assurance auto	912,50€	/	
Abonnement Téléphone	793,44€	21%	
Abonnement Co-Working	2 023,27€	21%	
Foire commerciale	2 352,07€	21%	
Assurance complète (hors voiture)	4 435,87€	/	
Cotisation annuelle des sociétés	347,50€	/	
Provision cotisation sociale	12 798,08€	/	
Salaires	60 000€	/	
Frais de publicité	991,68€	21%	
FF total	85 370,45€		FV unitaire total 8,99 €

FIGURE 7.1 – Coûts récapitulatifs pour la première année.

Quantités vendues aux magasins 2019														
	Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre	Total	Chiffre d'affaires
SuperBien	20	30	40	50	40	30	20	20	20	15	25	40	350	6 479,20 €
Aveve (2)	80	80	120	180	120	100	80	80	80	100	100	120	1220	22 584,64 €
Central Jardin (2)	140	140	200	220	200	160	140	140	140	140	180	200	2000	37 024,00 €
On Green (3)	270	240	450	750	270	240	240	240	240	240	300	360	3840	71 086,08 €
OOAK	20	30	40	50	40	30	20	20	20	15	25	40	350	6 479,20 €
Pépinières de Boitsfort	50	50	60	80	60	50	50	50	50	50	60	70	680	12 588,16 €
Les terres d'ici,	20	20	40	60	40	20	20	20	20	20	30	40	350	6 479,20 €
Eco Farm du long Fond	20	20	40	45	30	20	20	20	20	20	20	30	305	5 646,16 €
GreenPeas	0	0	0	0	0	0	0	0	0	0	3 980	0	3880	105 210,09 €
Foire Cocoon	20	20	30	50	30	20	20	15	15	15	20	20	275	5 090,80 €
BePositive	30	30	20	50	20	40	20	20	15	20	30	315	5 831,28 €	
Urban Gardener	30	30	20	50	20	30	30	30	20	20	30	330	6 108,96 €	
Culture Indoor	60	40	70	80	50	40	40	40	40	40	40	560	10 386,72 €	
Dille et Kamille (2)	TOTAL	760	730	1130	1665	920	780	700	695	680	675	4830	990	14554,5
	TOTAL cumulé	760	1490	4285	5205	5985	6685	7380	8060	8735	13564,5	14554,5	300 974,49 €	

FIGURE 7.2 – Quantité vendues en magasins pour la première année.

Key Partners	Key Activities	Value Propositions	Customer Segments
 <p>Who are our key partners? Who are our suppliers? Which suppliers are weourcing from? Which key alliances do we have?</p> <p>What key activities does our value proposition require? Our Customer Care? Customer Experience? Customer Support?</p>	<p>Capteurs connectés Gestion du site web et une application Gestion d'une base de données via l'inscription au forum</p> <p>Magasins locaux Concept Store Magasin de jardinage Clients (forum)</p>	<ul style="list-style-type: none"> • Service après-vente Community via le forum de notre site internet • Valeur : o Excellence o Facilité o Accessibilité o Environnement o Valeur par rapport aux concurrents 	<ul style="list-style-type: none"> • Cibles souhaitées o Jeunes adultes : 25-45 ans o Distributeurs en B2B • Portrait de notre cible o Cultive ses propres légumes, fruits ou herbes aromatiques o Vegan o Conscient de l'environnement o Mange sain o Personnes actives o Aiment jardiner o Aliment les plantes de toute sorte o Sachant utiliser la technologie o Vivant en appartement o Ayant le souhait de jardiner
 <p>Who are the key resources required for our business model? Who are our key assets? Which key resources are weourcing from? Which key assets do we have?</p> <p>What key resources does our value proposition require? Our Customer Care? Customer Experience? Customer Support?</p>	<p>Fournisseurs Lieu de stockage Co-working Base de données</p>	<ul style="list-style-type: none"> o Site web propre et application mobile o Points de vente physique o 5000 références de plantes o Possibilité d'utiliser le capteur avec plusieurs plantes o Fonctionne à l'aide de wifi o Entreprise Belge 	 <p>What key channels does our customer service require? Our Customer Care? Customer Experience? Customer Support?</p> <p>How are we going to reach our customers? How will they get to us? How much does each channel contribute to overall revenue?</p>
 <p>What are the most important costs inherent in the business model? Which key resources are most expensive? What key assets are most expensive?</p> <p>What value streams does our revenue model generate? How much does each revenue stream contribute to overall revenue?</p>	<p>Cost Structure</p>	<ul style="list-style-type: none"> • Production Site web et application mobile Achat de marchandise Frais généraux liés à l'exploitation 	 <p>Prix de vente Crowdfunding avant le lancement</p>

FIGURE 7.3 – Business model canvas - Sensory Garden

7.8 Études qualitatives et quantitatives

7.8.1 Qualitative en B2B

- Certains concepts store étaient très intéressés et d'autres pas du tout car cela ne correspondait pas aux valeurs et positionnement de l'entreprise.
- Les prix mis en avant étaient entre 20 et 30 euros. Selon eux, le bouche à bouche est ce qui aide le plus dans la promotion.
- Ils attendent que notre produit soit facile à utiliser et facilement vendable

7.8.2 Qualitative en B2C

- En général, les personnes que nous avons interrogés aiment jardiner chez eux et s'informe sur leurs plantes via internet. Nous devons donc penser à standardiser le forum afin de pouvoir répondre aux questions de nos clients. Cela pourra se faire grâce aux informations que nous allons récolter via leurs expériences.
- Ils préfèrent les plantes aromatiques. Nous devons donc nous concentrer sur le référencement des plantes aromatiques plus que les autres plantes.
- Les interrogés préfèrent acheter notre produit dans un point de vente physique. A l'heure actuelle, nous n'allons pas développer notre site pour la vente en ligne.
- L'application doit rester gratuite pour nos clients et les notifications se feront via cette application.

7.8.3 Quantitative

- D'une manière générale, les répondants s'occupent de plantes très diverses : légumes, fruits, fleurs, plantes d'intérieur, herbes aromatiques et plantes d'extérieur. Il est donc important que notre base de données soit alimentée d'informations relatives à toutes ces types de plantes. De plus, il est important de noter que les plantes les plus souvent mentionnées sont les plantes d'intérieur et les herbes aromatiques. Il est donc primordial que notre capteur soit adapté à ce type de plantes.
- La plupart des répondants effectuent leurs recherches concernant leurs plantes via internet. Il est donc important que notre base de données soit bien référencée, afin de devenir une source d'informations pour nos utilisateurs.
- La majorité des répondants désirent que notre capteur soit de couleur verte.
- En ce qui concerne les fonctionnalités de l'application, la grande majorité désire avoir accès à une page synthétique leur donnant les informations sur les besoins de leur plante. Ils désirent également avoir accès à une page d'informations sur tous types de plante. Moins d'une personne sur dix souhaite faire des achats via notre app. Il nous semble donc judicieux de ne pas développer cette option.
- La majorité des répondants désirent être informés sur la santé de leur plante via une notification sur leur smartphone.
- La grande majorité des répondants se disent prêts à télécharger notre application. Par contre, si elle est payante, seuls 2 personnes sur 10 la téléchargeraient.

8. Aspect RSE de l'entreprise

8.1 Accessibilité du site web

La plateforme web est entièrement accessible au personne présentant un handicap, que ce soit moteur ou visuel. En effet, la navigation avec la touche "TAB" et "SHIFT" + "TAB" est entièrement gérée afin d'avoir une expérience utilisateur la plus agréable possible. Pour cela, nous avons utilisé les attributs ARIA (*Accessible Rich Internet Applications*) et l'attribut tabindex afin que les lecteurs d'écran puisse comprendre la page. Nous avons également ajouté un "trap focus" afin de "piéger" le focus à l'intérieur des boites de dialogues modales.

Nous avons également ajouté un mode daltonien pour notre page présentant les données sous forme de graphique. Il s'agit simplement d'une case à cocher afin d'ajouter des formes dans les couleurs, cela permettra aux personnes souffrant d'une défaillance de la perception des couleurs de différencier les valeurs à améliorer des valeurs correctes.

Toutes les images possèdent un attribut alt qui permet d'afficher un texte alternatif si cette dernière ne charge pas. Ce texte est aussi accessible par les lecteurs d'écran. Nous avons aussi fait en sorte que les graphiques soient interprété comme des tableaux de données par les outils d'accessibilité.

Afin de s'assurer de l'accessibilité de nos pages, nous avons utilisé des outils de test tel que WAVE (*web accessibility evaluation tool*) et *Lighthouse*, un plugin chrome permettant de tester un site web (accessibilité, temps de chargement,...).

Nous avons également configuré accessLint sur GitHub, un outil de vérification des bonnes pratiques pour l'accessibilité lors du déploiement.

8.2 Green-it

Pour notre site web, nous avons choisi un hébergeur respectueux de l'environnement : OVH. En effet, cette entreprise a mis en place plusieurs technique afin de réduire le coût énergétique de leur serveur.

- Ils développent leur propre système de refroidissement.
- Ils ont complètement supprimé la climatisation
- L'électricité ne représente plus que 10% du coût de fonctionnement des serveurs.

Vous pouvez retrouver ces informations sur : <https://www.ovh.com/fr/apropos/green-it.xml>

Suite à la conférence Green IT, nous avons abandonné l'idée d'ajouter des animation sur notre page d'accueil car ces dernières alourdissent la page web.

Nous avons également essayé de réduire au maximum le temps de chargement de notre page en faisant le moins de requête possible aux serveurs. Nous avons également minifier nos fichiers CSS et JS et compressé nos images afin de réduire notre empreinte environnementale.

Le seul problème que nous rencontrons c'est la carte de Google Map intégrée à notre site web. Lors du chargement de la page, cette dernière effectue de nombreux appels réseaux et réduis notre score sur les outils de test. (Plus ou moins 30 requêtes)

Le fait que nous avons codé une Progressive Web App est aussi une approche Green-IT. En effet, du fait que tout notre site web s'enregistre dans la cache du navigateur du client, nous ne faisons réellement des requêtes au serveur qu'à la première ouverture du site. De plus, notre site charge sans devoir communiquer avec le serveur afin de fonctionner hors-ligne et d'économiser de la bande passante sans surcharger le réseau et le serveur.

Pour tester si notre site respectais bien les pratiques du green IT, nous avons utilisé le site ecoindex.fr où nous obtenons un score de A. Le fait que nous avons implémenté une Progressive Web App, permet d'améliorer notre score de C à A grâce à la mise en cache des fichiers statiques.

8. Bibliographie

- [1] almeida. Documentation du système de pagination jpages. <http://luis-almeida.github.io/jPages/>.
- [2] ascott1. Documentation bigslide.js. <http://ascott1.github.io/bigSlide.js/>.
- [3] autodesk. Formation fusion 360. <https://f360ap.autodesk.com/courses/getting-started-in-fusion-360>.
- [4] A Een bedrijf van Premier Farnell. Farnell. <https://be.farnell.com/fr-BE>.
- [5] bruno. Configuration capitano pour rails. <https://gist.github.com/bruno-/9808201>.
- [6] Capistrano. Gem ruby capistrano. <https://github.com/capistrano/capistrano>.
- [7] Chart.js. Documentation chart.js. <https://www.chartjs.org/docs/latest/>.
- [8] Easy Concept. Internet simple pour tous. <https://www.easy-concept.com/blog/?2009/05/28/788-algorithme-de-verification-d-un-numero-de-compte-bancaire-belge#.W8mbXfmYTIU>.
- [9] James Coonce. Tutoriel authentification knock. <https://codebrains.io/rails-jwt-authentication-with-knock/>.
- [10] Refsnes Data. w3schools. <https://www.w3schools.com/css/>.
- [11] davidshimjs. Documentation qrcodejs.js. <https://davidshimjs.github.io/qrcodejs/>.
- [12] draw.io. Site de dessin en ligne (design bdd). <https://www.draw.io/>.
- [13] fengyuanchen. Documentation datepicker.js. <https://fengyuanchen.github.io/datepicker/>.
- [14] Marc G Gauthier. Formation ruby - openclassrooms. <https://openclassrooms.com/fr/courses/2913686-lancez-vous-dans-la-programmation-avec-ruby>.
- [15] Marc G Gauthier. Formation ruby on rails - openclassrooms. <https://openclassrooms.com/fr/courses/3149156-initiez-vous-a-ruby-on-rails>.
- [16] Google. Conférence offline web apps par google en 2016. <https://www.youtube.com/watch?v=cmGr0RszHc8>.
- [17] Google. Cours offline-first web apps. <https://classroom.udacity.com/courses/ud899>.
- [18] Google. Documentation google. <https://developers.google.com/web/>.
- [19] Google. Outils d'audit lighthouse. <https://developers.google.com/web/tools/lighthouse/>.
- [20] Nick Hartunian. Tutoriel authentification via token jwt. <https://medium.com/@nick.hartunian/knock-jwt-auth-for-rails-api-create-react-app-6765192e295a>.
- [21] hiddedevries. Fonction trap focus. <https://hiddedevries.nl/en/blog/2017-01-29-using-javascript-to-trap-focus-in-an-element>.
- [22] imakewebthings. Documentation waypoints.js. <http://imakewebthings.com/waypoints/guides/getting-started/>.
- [23] HK Weiku information & Technology Co. Pcb prototype the easy way. <https://www.pcbway.com/>.
- [24] inkscape. Site pour l'outil de dessin de découpe laser. <https://inkscape.org/fr/>.
- [25] Creston Jamison. Configuration serveur production rails sur ubuntu 18.04. <https://www.rubytreehardware.com/resources/ruby-on-rails-41-ubuntu-1404-server-deployment/>.
- [26] javascripter. Convertir couleur hex en rgb. <http://www.javascripter.net/faq/hextorgb.htm>.
- [27] Jquery. Documentation jquery. <http://api.jquery.com/>.
- [28] Hazel Virdó & Kathleen Juell. Configuration https sur ubuntu 18.04. <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04>.
- [29] JWT.io. Site officiel json web token. <https://jwt.io>.
- [30] Austin Kabiru. Tutoriel api restful rails. <https://scotch.io/tutorials/build-a-restful-json-api-with-rails-5-part-one>.

- [31] James Kropp. Tutoriel authentification via token. <https://engineering.musefind.com/building-a-simple-token-based-authorization-api-with-rails-a5c181b83e02>.
- [32] lucasmonteiro001. Exporter json vers csv. <https://gist.github.com/lucasmonteiro001/6c6c0ad27507d2281fdd8cb2165a2a67>.
- [33] mail tester.com. Site de test serveur mail. <https://www.mail-tester.com>.
- [34] makilab. Wiki du makilab. <https://wiki.makilab.org/index.php/Accueil>.
- [35] marked. Documentation de marked.js. <https://marked.js.org/>.
- [36] Traversy Media. Simple ruby on rails 5 rest api from scratch. <https://www.youtube.com/watch?v=QojnRc7SS9o>.
- [37] minifier. Minifier fichier css et js. <https://www.minifier.org/>.
- [38] moment.js. Documentation moment.js. <https://momentjs.com/>.
- [39] Monoso. Test driven rspec - episode 02, testing a rails api. <https://www.youtube.com/watch?v=Wb3oIfiLdZU>.
- [40] mozilla foundation. Guide sur l'accessibilité. <https://developer.mozilla.org/en-US/docs/Web/Accessibility>.
- [41] MXToolBox. Boite à outils pour records dns et serveur mail. <https://mxtoolbox.com>.
- [42] Mozilla Developpers Network. Documentation cache api. <https://developer.mozilla.org/en-US/docs/Web/API/Cache>.
- [43] Mozilla Developpers Network. Documentation fetch api. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API.
- [44] Mozilla Developpers Network. Documentation mozilla. <https://developer.mozilla.org/fr/>.
- [45] nsarno. Librairie knock authentification jwt. <https://github.com/nsarno/knock>.
- [46] Ruby on Rails. Documentation création api avec rails. https://guides.rubyonrails.org/api_app.html.
- [47] Ruby on Rails. Documentation ror. <https://api.rubyonrails.org>.
- [48] Ruby on Rails. Guide ruby on rails. <https://guides.rubyonrails.org>.
- [49] Ruby on Rails. Ruby on rails site. <https://rubyonrails.org>.
- [50] RSpec. Librairie de tests pour rails - rspec. <http://rspec.info>.
- [51] SamBrishes. Documentation de tail.writer. <https://github.com/pytesNET/tail.writer>.
- [52] Codemy School. Rails api : Json web token part 1. <https://www.youtube.com/watch?v=HxvLVF2n4LY>.
- [53] Codemy School. Rails api : Setting up basic crud. <https://www.youtube.com/watch?v=V1xIa6JemeI>.
- [54] Ltd Seeed Technology Co. seeed the iot hardware enabler. <http://wiki.seeedstudio.com/>.
- [55] select2. Documentation select 2. <https://select2.org/>.
- [56] Kirill Shevch. Déploiement continu avec capistrano pour rails. https://medium.com/@kirill_shevch/continuous-deployment-with-circleci-and-capistrano-for-web-applications-d8fea0192114.
- [57] tomav. Image docker pour le serveur mail. <https://github.com/tomav/docker-mailserver>.
- [58] Travis-CI. Documentation travis-ci pour ruby. <https://docs.travis-ci.com/user/languages/ruby/>.
- [59] Travis-CI. Site travis-ci. <https://travis-ci.org>.
- [60] Auteurs variés. Bible du développeur. <https://stackoverflow.com/questions/>.
- [61] Auteurs variés. Site de répertoriassions d'icônes. <https://www.flaticon.com>.
- [62] Wojciech Wnętrzak. Déploiement capistrano depuis travis-ci. <https://cowbell-labs.com/2014-03-06-capistrano-deploys-from-travis.html>.

A. Annexes

Box

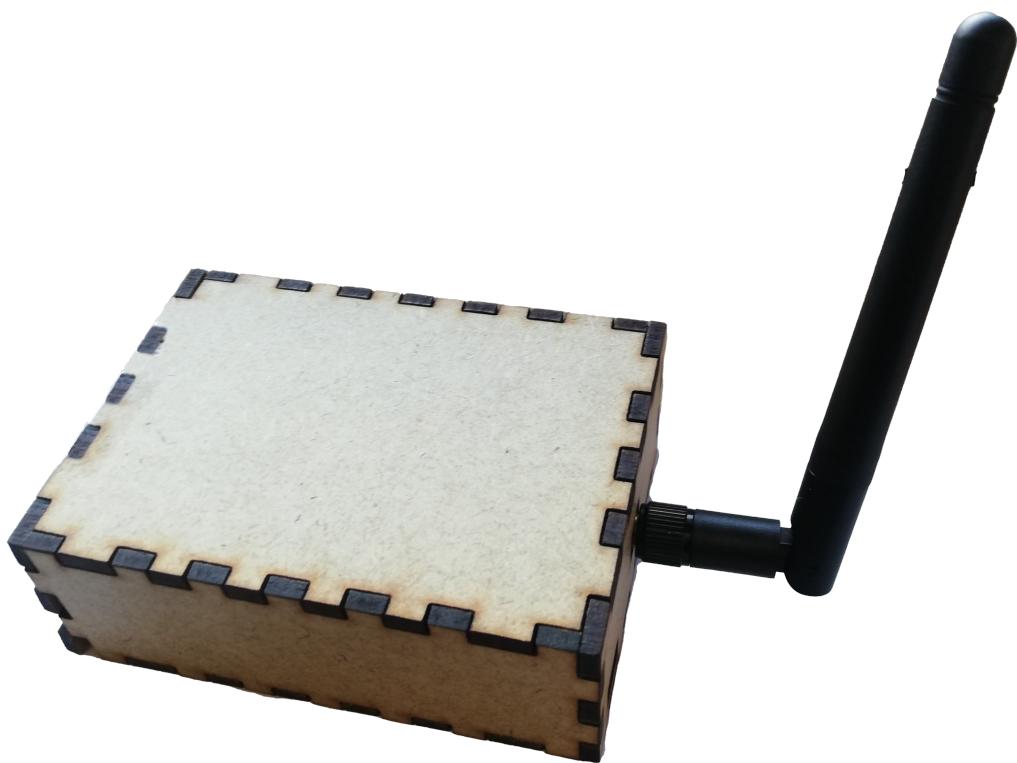


FIGURE A.1 – La boîte pour la board

Base de données

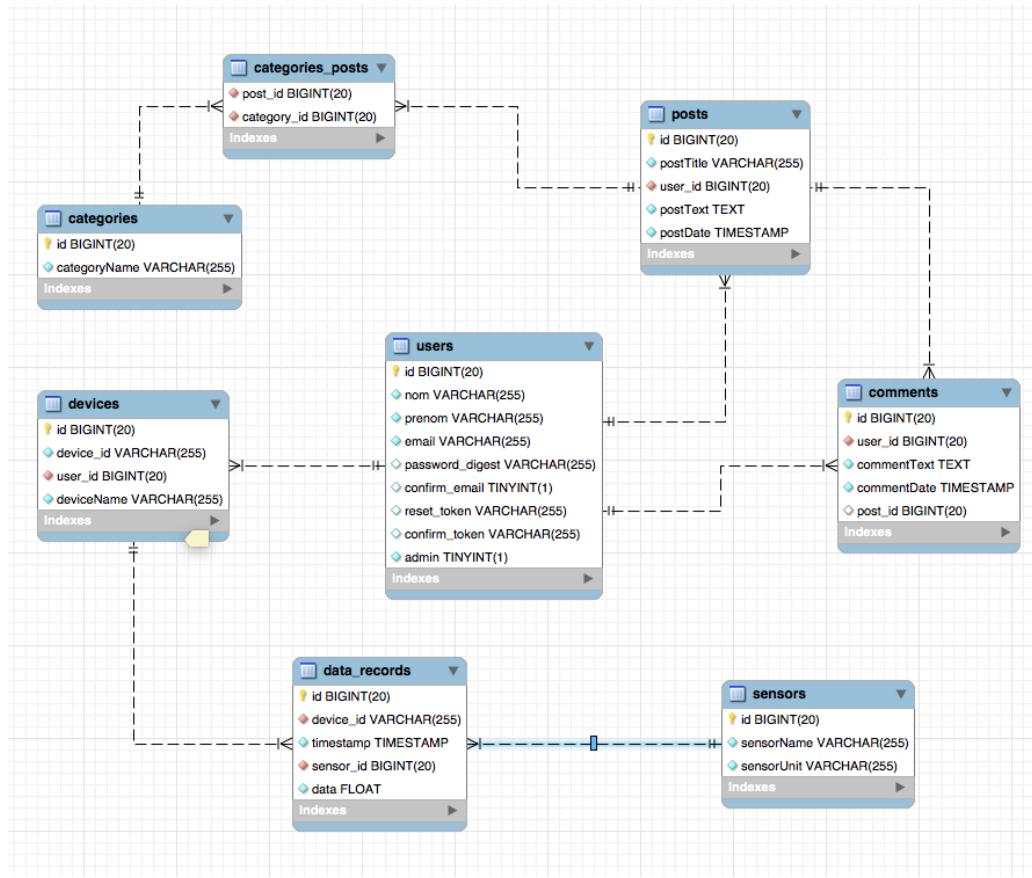


FIGURE A.2 – Schéma de la base de données en production

HTTPS

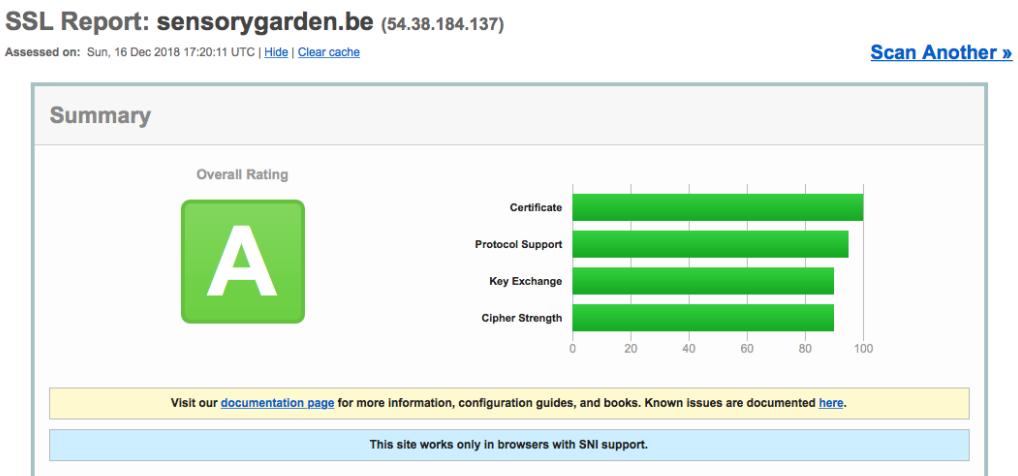


FIGURE A.3 – Résultat audit sécurité - Qualys SSL Labs

Mail tester



FIGURE A.4 – Résultat test serveur mail - mail-tester.com

Lighthouse

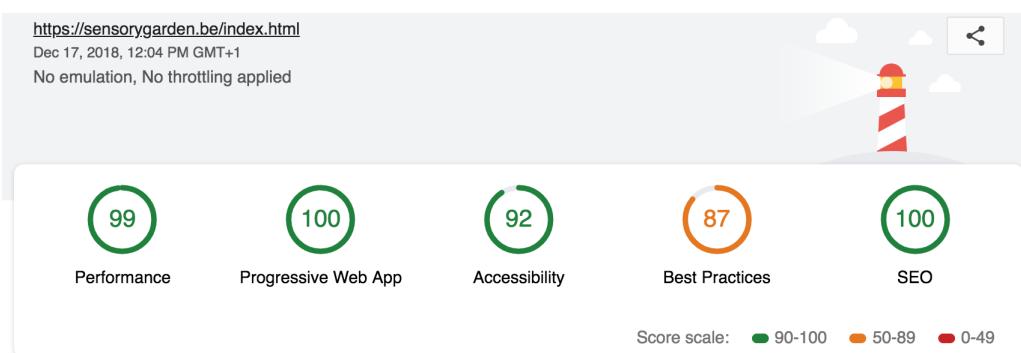


FIGURE A.5 – Résultat test audit Lighthouse

Ecoindex.fr



→ sensorygarden.be

calculé le 17 / 12 / 2018

Renouveler le test pour
<https://sensorygarden.be>

→ Performance environnementale ⓘ



FIGURE A.6 – Résultat test Green-IT - ecoindex.fr

Certificat DNS



Voici les données de la personne ou de l'organisation qui, au moment de la demande, est mentionnée comme titulaire de ce nom de domaine dans la base de données de DNS Belgium.

NOM DE DOMAINE [sensorygarden.be](#)

Statut:	Registered
Inscription:	2 octobre 2018
Dernier changement:	3 octobre 2018 07:45
Date d'expiration:	2 octobre 2019 00:49
Organisation:	
Contact:	Adrien Nini Pereira
Langue:	Français
Adresse:	Rue Raymond Pieret 38
	1480 Clabeck
	Belgique
Téléphone:	+32.23760142
Fax:	
E-mail:	mtk6rbmkouosxtzvrq2h@c.o-w-o.info

FIGURE A.7 – Certificat d'enregistrement DNS

Accessibility checker

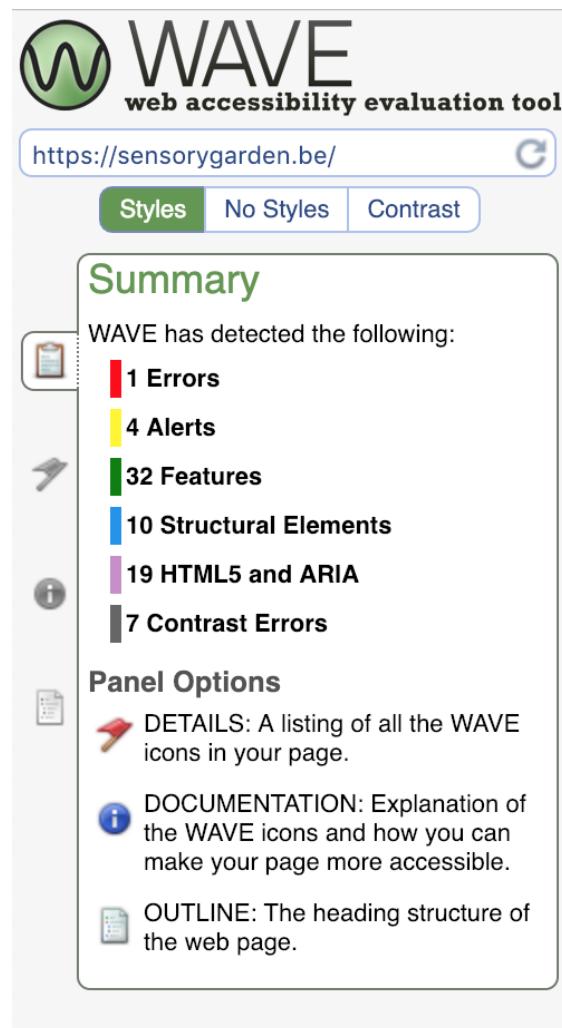


FIGURE A.8 – WAVE - L'erreur est due à Google Maps