

Nini pereira Adrien
Fauconnier Simon
Heydari Nazanin
Henry de Generet

Goossens Jonathan
de Mahieu Benoit
Duquaine Cassandra

Rapport Projet d'intégration : **Sprint 2**

Lien vers la release produite durant le Sprint

- <https://trello.com/b/fCD5mNpZ/sensory-garden>
- <https://sensorygarden.be/>
- <https://github.com/EPHEC-Enovatech>
- <https://a360.co/2NLokCs> (Modèle 3D)

Rapport Bilan de sprint

Pour ce sprint-ci, au niveau de la board, nous avons ajouté un capteur de qualité de l'air et également commandé un capteur mesurant l'humidité de la terre. Ensuite, nous avons testé la batterie et le panneau solaire pour vérifier leur fonctionnement. Pour finir nous avons contacté une entreprise de miniaturisation (VMC électronique) pour avoir une idée du coût de l'opération.

Pour protéger la board nous avons modélisé une boîte grâce à fusion 360. La board va pouvoir se fixer grâce à des petits embouts. La boîte se refermera grâce à un même système. Par la suite nous pourrions imprimer la boîte grâce à une imprimante 3D.

Nous avons continué à optimiser le site web pour qu'il soit opérationnel pour les mobiles et tablettes. Nous avons ajouté des photos présentant l'équipe de travail, une partie messagerie qui permettra de recevoir des mails d'utilisateurs ainsi qu'une carte indiquant les futurs endroits de vente du produit final.

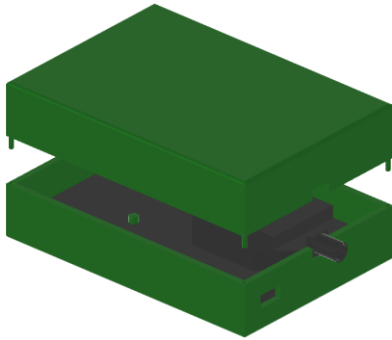
Au niveau du back-end, nous avons terminé l'API permettant de transmettre les données reçues par les capteurs dans la base de données. Il y a également le serveur mail qui est opérationnel avec comme adresse de contact contact@sensorygarden.be et noreply@sensorygarden.be pour nous permettre d'envoyer une réponse. Nous avons également mis en place un système de déploiement continu grâce à Travis-CI et github.

Récapitulatif des objectifs du sprint

Pour ce sprint nous avons comme objectif d'ajouter des capteurs sur la board ainsi qu'envoyer dans la base de données. Mettre en place le responsive design pour les mobiles et tablettes. Nous voulions également terminer le design de la boîte afin d'avoir déjà un premier modèle à présenter pour la review. Enfin nous voulions faire une page de connexion et d'inscription allant avec la base de données mise en place.

Schémas et analyses réalisée durant le sprint

Box protégeant la board



Tests unitaires de l'API

```
Adrien@MacBook-Pro-de-Adrien-Mini: ~/Dev/projet-integration/sensorygarden-api [dev] bundle exec rake spec
/Users/Adrien/.rbenv/versions/2.5.1/bin/ruby -I/Users/Adrien/Dev/projet-integration/sensorygarden-api/vendor/bundle/ruby/2.5.0/gems/rspec-core-3.8.0/lib:/Users/Adrien/Dev/projet-integration/sensorygarden-api/vendor/bundle/ruby/2.5.0/gems/rspec-support-3.8.0/lib:/Users/Adrien/Dev/projet-integration/sensorygarden-api/vendor/bundle/ruby/2.5.0/gems/rspec-core-3.8.0/exe/rspec --pattern spec/\*\*\{,\/\*\*\*\}\/*_spec.rb
.....
Finished in 2.82 seconds (files took 2.39 seconds to load)
58 examples, 0 failures

COVERAGE: 99.01% -- 601/607 lines in 30 files

+-----+-----+-----+-----+
| coverage | file | lines | missed | missing |
+-----+-----+-----+-----+
| 85.71% | spec/rails_helper.rb | 14 | 2 | 30-31 |
| 96.00% | app/controllers/users_controller.rb | 25 | 1 | 29 |
| 96.00% | app/controllers/sensors_controller.rb | 25 | 1 | 29 |
| 97.14% | app/controllers/devices_controller.rb | 35 | 1 | 40 |
| 97.62% | app/controllers/datas_controller.rb | 42 | 1 | 48 |
+-----+-----+-----+-----+
25 file(s) with 100% coverage not shown
Adrien@MacBook-Pro-de-Adrien-Mini: ~/Dev/projet-integration/sensorygarden-api [dev]
```

Site web



Monitoring

Monitors

Status	Last 7 Days	Type	18 Oct	17 Oct	16 Oct	15 Oct	14 Oct	13 Oct	12 Oct
	100.00%	Sensory Garden	http	100.00%	0.000%	0.000%	0.000%	0.000%	0.000%

Overall Uptime

100.00% (last 24 hours)
(no downtime)


100.00% (last 7 days)
(no downtime)


100.00% (last 30 days)
(no downtime)


Latest Downtime


No downtime recorded.

Quick stats

 All Clear
Refreshing in 41 secs

 Up 1

 Down 0

 Paused 0

Tests unitaires du sprint

Pour ce sprint, nous avons mis en place un système de déploiement continu grâce à Travis-CI et Github. Ce système nous permet de tester notre API grâce à des tests unitaires automatisés et de déployer automatiquement en production lorsque les modifications sont validées.

Estimation de la vélocité de l'équipe

Au niveau de la vélocité, nous avons mis 24 points pour ce sprint. Nous avons utilisé pour la première fois le planning poker pour planifier le sprint. Nous déciderons par la suite si nous gardons ce rythme pour les sprints suivants. Nous n'avons pas tout terminé mais nous avons réalisé un système de déploiement continu en plus de ce qui était prévu.

Forces/Faiblesses

Pour ce sprint nous avons eu l'occasion de tester une première fois le planning poker permettant de coter nos users stories. Nous remarquons malgré tout que nos users stories sont trop longues et sont divisées en un trop gros nombre de points. Nous avons également vécu un premier problème que nous n'avions pas prévu.

Nous avons toujours gardé une bonne vélocité au niveau des daily scrum. Cela nous permet chaque jour de voir l'avancement de chacun dans le travail et de voir ce qui se termine ou pas. Ensuite nous avons également mis timecamp en place pour voir le temps de travail de chacun sans jugement.

Objectifs du prochain sprint

Pour le sprint suivant, nous allons diviser les users stories en plus petites user stories comme cela nous pourrons voir plus facilement l'avancement des tâches. Grâce à cela nous pourrons voir l'avancement et l'accomplissement de notre travail en temps réel.

Ensuite nous allons terminer la partie connexion et inscription. Avec cela nous aurons une base de données prête avec les utilisateurs et les capteurs qu'ils pourront encoder par la suite.

Nous choisirons également d'autres users stories à implémenter durant le sprint 3.