



Rapport Projet Sport & Co M2 MIAGE 2020-2021

29 MAI

Master 2 MIAGE Bordeaux : Formation en alternance

Thibault DE CET, Domonkos DENES, Amine KHELIFI, Nathan JOUBERT, Allan MENDJELI

MIAGE RÉSEAU
DES MIAGE
DE FRANCE
Bordeaux

Table des matières

Contexte du projet.....	3
Outils utilisés.....	3
Mode de travail.....	4
Planning de travail.....	5
Matrice de compétences de l'équipe au début du projet (sur 3 points)	6
Initialisation du projet.....	6
Schéma de l'architecture projet.....	8
Principales difficultés rencontrées.....	9
Matrice de compétence de l'équipe en fin de projet (sur 3 points)	10
Idées d'ouverture.....	10
Conclusion.....	11

Contexte du projet

Le but de ce projet était de créer un site Web permettant aux utilisateurs de louer du matériel sportif et à des loueurs de proposer à la location leur matériel sportif. Ce site propose 3 types de parcours :

- Le parcours client : il permet aux utilisateurs de louer du matériel en fonction d'une période donnée et de la quantité désirée suivant les stocks disponibles.
- Le parcours propriétaire : il permet aux propriétaires de mettre en location leurs équipements.
- Le parcours administrateur : il offre la possibilité d'accepter ou de refuser le statut de loueurs aux utilisateurs, il peut également voir la liste de tous les loueurs présents sur le site.



Figure 1 : Logo de Sport & Co

Pour ce faire, nous avons choisi d'utiliser la technologie **Angular.js** (version 11.2.5) pour la partie Front-end, **Node.js** (version 14.16.0) pour la partie Back-end et concernant notre base de données nous avons utilisé MySQL. Nous avons, logiquement, fait le choix de partir sur du **Angular.js** car tout le monde au sein de notre projet en avait déjà plus ou moins fait. Concernant **Node.js**, seul Domonkos n'en avait jamais fait, même s'il avait déjà travaillé sur du Back-End en JEE, le reste de l'équipe avait déjà pu travailler dessus lors des projets de l'UE "Méthodes Agiles". C'est pour cela que ce choix de technologie côté Back nous paraissait pertinent. Pour le choix de "MySQL" la question ne s'est pas vraiment posée étant donné que nous avons tous travaillé avec dans le cadre de notre licence et Master MIAGE.

Outils utilisés

Durant cette période d'un mois, nous avons utilisé divers outils informatiques pour mener à bien ce projet :

- Google Drive : cette plateforme nous a permis de centraliser en ligne la plupart des informations et fichiers nécessaires pour notre projet comme les modèles MCD et scripts SQL pour notre base de données ou encore les éléments visuels, les installations, la documentation et normes liées à notre projet.
- Figma : cet outil en ligne nous a permis de mettre en place les maquettes graphiques pour notre site web. Son utilisation nous permettait également de récupérer les paramètres de mise en forme pour nos fichiers ".css".

- Trello : ce tableau de bord en ligne nous a permis de faire le suivi des tâches à faire, en cours et terminées tout au long de notre projet afin d'avoir une meilleure visibilité globale.
- Discord : c'est sur ce logiciel de chat et visio-conférences que nous nous réunissions pour travailler chaque jour. Nous avons travaillé sur plusieurs salons vocaux différents pour que chacun puisse se déplacer entre eux pour poser des questions sur des tâches ou des problèmes.
- Zoom : pour pouvoir effectuer nos réunions de suivi du projet hebdomadaires avec M. Maabout.
- IntelliJ Ultimate ou Visual Studio Code : ces environnements de développement intégré (ou IDE) nous ont permis d'effectuer nos développements côté "Front-end" et "Back-End".
- GitHub : outil de travail collaboratif qui offre les fonctionnalités de versionnement de Git, et permet également de récupérer, envoyer et mettre en commun les développements effectués par chaque personne/groupe. Son utilisation était indispensable tout au long du projet. Chaque US était développé sur sa branche propre, qui était ensuite raccordé à la branche principale d'un sprint une fois la relecture de code effectuée.
- MySQL Workbench : sur ce logiciel nous avons pu nous occuper de mettre en place toute la base de données du projet, tels qu'avec les scripts de création et de remplissage de la base avec des données exemples. Nous avons également créé des vues dessus afin de pouvoir accéder à certains champs/données plus facilement.
- Postman : il s'agit d'une plateforme servant à tester son API Back-End, ce grâce à l'envoi des requêtes de test.
- Scrum Poker-Online.org : site en ligne nous permettant d'effectuer le chiffrage de nos différentes US.

Mode de travail

Pour ce projet, nous avons choisi de travailler de manière "Agile", en effectuant du pair programming en début de projet afin que les membres avec le plus de compétences puissent aider ceux qui avaient plus de difficultés avec les technologies utilisées. De plus, nous fonctionnions sous forme de "sprint" d'une durée de 2 semaines. Au début de chaque sprint, on choisissait les tâches à effectuer en fonction de leur importance à nos yeux par rapport au fonctionnement du site.

Thibault s'est chargé de rédiger des User Story (US) en début de projet afin que nous puissions cadrer et comprendre les différentes fonctionnalités qui allaient proposer notre site Web. Il s'est également occupé de classer ces US par ordre de priorité vis-à-vis du projet (Vital, Essentiel ou Nice to have) et en décrivant le type de développement à y faire (Front ou Back). Dans la continuité de la méthode "Agile", les US étaient chiffrés, avec l'outil **Scrum Poker-Online.org** par l'ensemble afin de savoir combien de temps nous devions passer dessus. Ceci dans le but d'avoir une meilleure visibilité sur les temps de développements nécessaires et ainsi savoir si nous pourrions tenir les deadlines que nous nous étions fixés.

Ces US étaient répertoriés sur l'outil **Trello**, qui permet d'attribuer ces tâches aux différents membres du groupe, mais également de les classer en diverses catégories :

- Backlog : US rédigée mais pas encore commencée/attribuée
- A faire : US rédigée et pas encore commencée/attribuée, mais que l'on souhaite effectuer lors du sprint en cours
- En cours : US en cours de développement par une ou plusieurs personnes de l'équipe

- Bloqué : US terminée en termes de développement, mais qui n'a pas été vérifiée par un ou plusieurs autres membres n'ayant pas travaillé dessus
- A merge : US à fusionner avec une autre branche sur Git, dans le cas d'US dépendante ou des US de fin de sprint.
- Terminé (sprint) : US terminée et vérifiée par un ou plusieurs autres membres n'ayant pas travaillé dessus

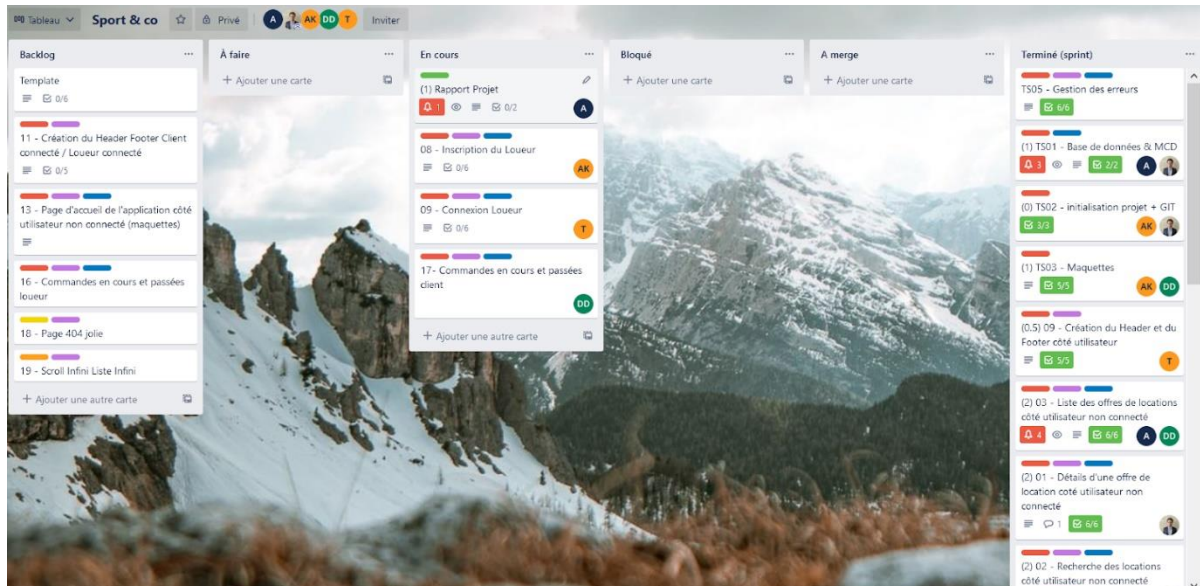


Figure 2 : Capture d'écran de Trello utilisé pour ce projet

Vers la fin du projet nous nous sommes peu à peu éloignés du "pair programming" pour avancer plus vite sur chacune des US et puissions également gagner en autonomie tout en ayant l'aide de certains si nécessaire sur Discord.

Planning de travail

Pour bien nous répartir les US et faire le point sur l'avancement, nous avons planifié des réunions tous les matins du lundi au vendredi sur le logiciel **Discord** afin de pouvoir discuter des éventuels problèmes rencontrés ou des difficultés de certains avec les technologies. Le reste du temps était réservé au temps de développement, nous restions sur le discord dédiés afin que chacun puisse communiquer avec les autres membres du groupe.

Chaque jeudi avait lieu notre réunion hebdomadaire avec M. Maabout pour pouvoir faire le point sur l'avancement et voir si le déroulement du projet correspondait bien à ses attentes.

Planning	Lundi	Mardi	Mercredi	Jeudi	Vendredi
9h	Daily Meeting	Daily Meeting	Daily Meeting	Daily Meeting	Daily Meeting
9h30	Développement	Développement	Développement	Développement	Développement
11h	Développement	Développement	Développement	Réunion avec M.Maabout	Développement
12h	Pause de midi	Pause de midi	Pause de midi	Pause de midi	Pause de midi
14h	Reprise du développement	Reprise du développement	Reprise du développement	Reprise du développement	Reprise du développement

18h	Fin du développement	Fin du développement	Fin du développement	Fin du développement	Fin du développement
-----	----------------------	----------------------	----------------------	----------------------	----------------------

Planning du projet

Matrice de compétences de l'équipe au début du projet (sur 3 points)

Voici la matrice de compétence de toute l'équipe au sein de ce projet, on peut voir que le niveau est assez hétérogène du groupe. On remarque également que Nathan était celui ayant le plus de compétences au sein de ce projet, nous l'avons donc sollicité pour que la montée en compétence dans ce groupe se fasse plus rapidement.

Membre / Techno	Angular	Node JS	SQL	Git
Allan	1	1.5	2	1.5
Thibault	1.5	1.5	3	2
Nathan	2.5	2	3	2
Amine	1	1.5	3	2
Domonkos	1.5	1	2	3

Matrice de compétence de l'équipe au départ du projet

Initialisation du projet

Pour bien commencer ce projet, il était primordial de bien cerner les différents enjeux et besoins tout en y cadrant bien les limites qui y seraient associées. Pour cela nous nous sommes divisés en 3 équipes qui s'occupaient respectivement de rédiger les User Story décrivant le fonctionnement du site Web, de mettre en place le modèle MCD et de designer les maquettes graphiques.

Comme nous l'avons dit précédemment, Thibault s'est proposé pour la rédaction tandis que Allan et Nathan eux se sont occupés de modéliser le schéma MCD. Concernant ce modèle, il a évolué au fur et à mesure du projet étant donné que certaines US ont révélé des erreurs ou des oublis dans notre modèle initial, ce qui nous a permis également de le revoir et de le repenser pour d'éventuelles évolutions.

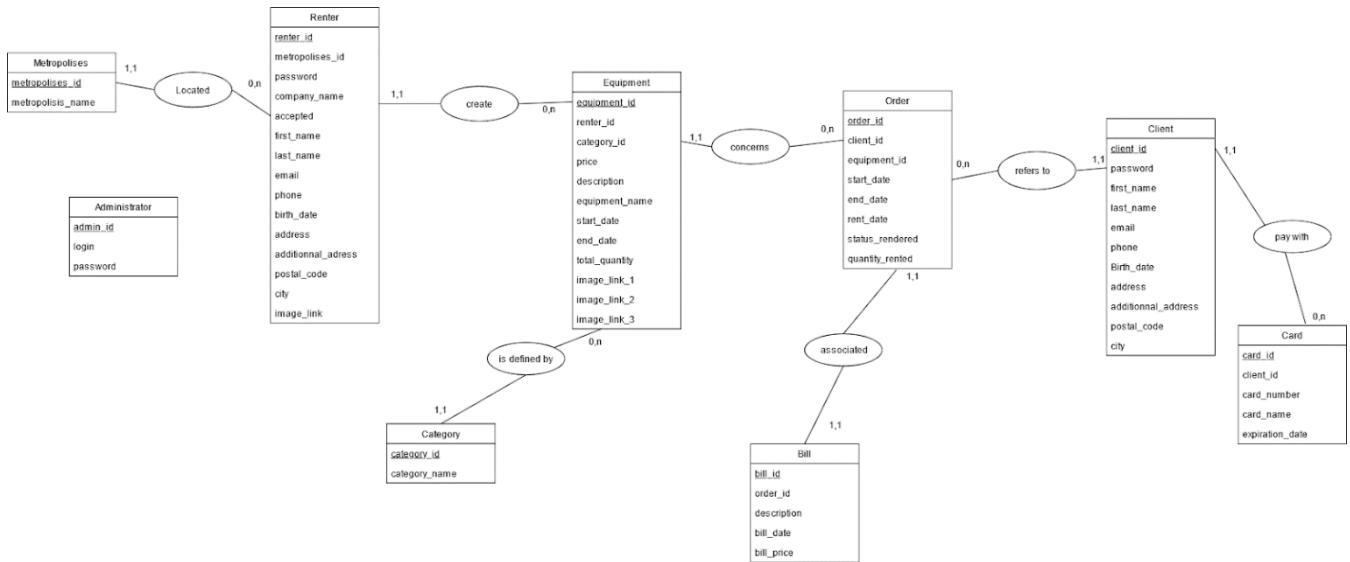


Figure 3 : Modèle MCD de Sport & Co

De leur côté Domonkos et Amine se sont occupés des maquettes graphiques via l'outil Figma. Ils ont ainsi pu modéliser les différentes pages qui composent notre site tout en récupérant les informations graphiques de chaque champ grâce à cet outil ce qui nous a grandement aidés pour le développement HTML/CSS.

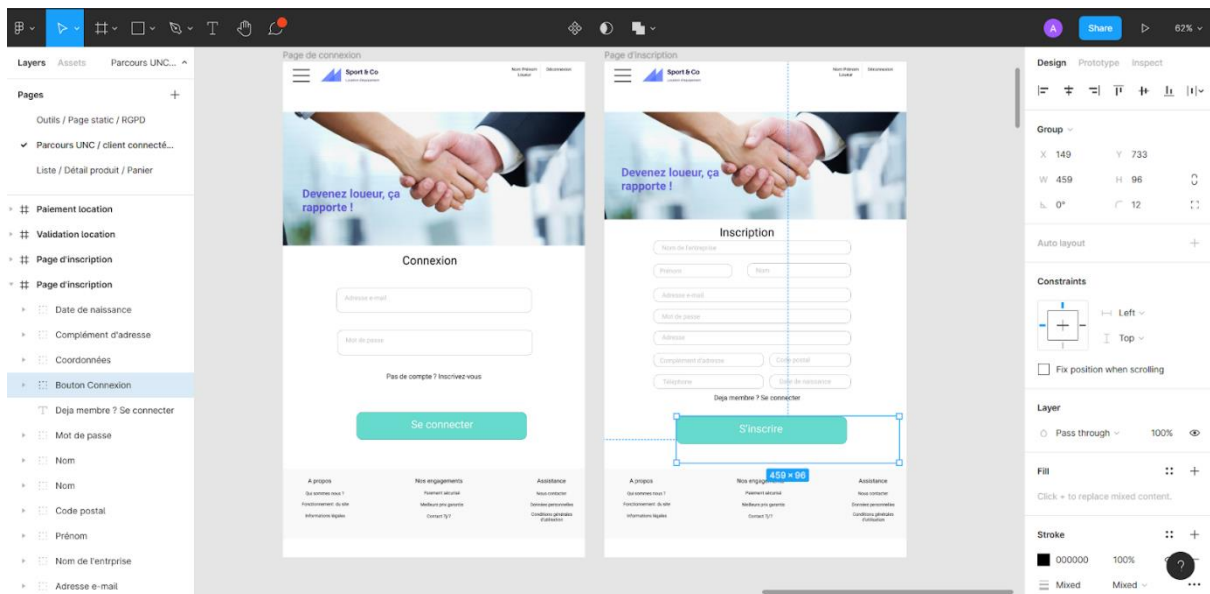


Figure 4 : Maquettes de Sport & CO

Schéma de l'architecture projet

Le fonctionnement de notre application repose sur une architecture assez simple et facilement utilisable, ce qui facilite la maintenance ou la reprise.

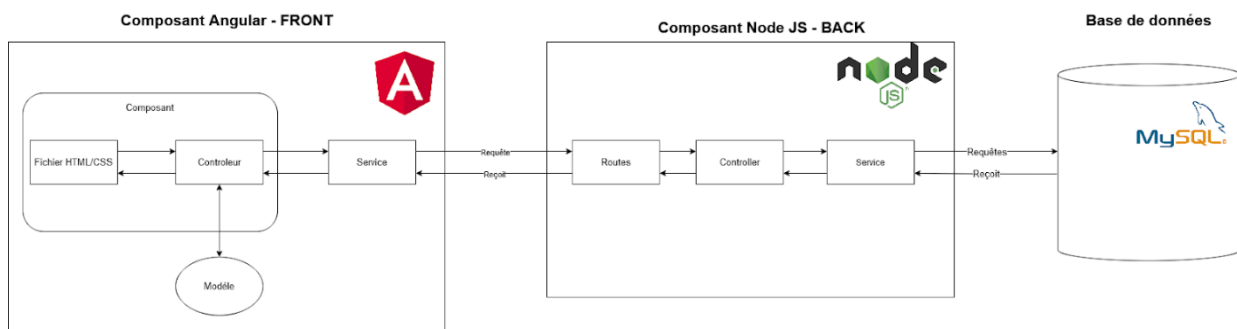


Figure 5 : Fonctionnement simplifié de Sport & Co

Les fichiers HTML et CSS sont respectivement ceux qui vont contenir le code de la partie strictement visuel de l'application. Grâce au framework Angular, ils se reposent sur des fichiers .ts, notamment le conteneur, qui va contenir les fonctions servant directement au fichier HTML. Mais ces fichiers seuls ne suffiraient pas pour notre application, et pour communiquer avec la partie Back du logiciel, un service est nécessaire. Celui-ci va donc envoyer et recevoir des requêtes. Les composants utilisent également les modèles, qui servent à représenter les données sous forme d'objet, mais également à récupérer les données au format JSON et de les convertir en modèles.

La requête est donc envoyée sous forme de requête HTTP au Back. Les requêtes que nous avons utilisées auront été les suivantes :

- GET : Cette requête permet de récupérer les données, sans effets sur les ressources. Elle peut être paramétrée pour récupérer des données en particulier.
- POST : Elle est utilisée pour transmettre des données, notamment utilisées ici pour la création de ressources.
- PUT : Nous nous en sommes servis pour la mise à jour des données des ressources
- DELETE : Cette dernière requête sert à supprimer une ressource du serveur.

Les ressources étant ici la base de données. Cependant, la connexion n'est pas directe entre le service d'Angular et la base de données, le back se trouve entre les deux. Il sert de tampon pour la réalisation d'opérations de transformation des données, notamment pour transformer les requêtes originelles en requête SQL que la base de données comprendra.

La porte d'entrée du Back sont les fichiers routes, un par table de la base de données. Ils sont eux-mêmes reliés à un fichier contrôleur. Ce dernier, comme son nom l'indique, va effectuer les différents contrôles sur les données, et enfin appeler les services. Ce sont eux qui communiqueront avec la base.

Principales difficultés rencontrées

Durant ce projet, nous avons dû faire face à plusieurs difficultés, qu'elles soient d'ordre technique ou organisationnel.

L'un des points de blocage aura été une mauvaise gestion du git par moment. Cela s'est notamment expliqué par des mauvaises compréhensions d'informations, comme la branche censée servir pour réaliser le css, qui n'aura pas été tenu à jours d'autres modifications, entraînant des conflits par la suite. Ou encore des bugs réglés sur d'anciennes branches plus utilisées et donc plus à jour avec l'actuel.

Concernant les difficultés techniques, elles auront plus été individuelles, et se sont principalement produites en Front, avec Angular.

Par exemple pour Nathan, la première difficulté aura été sur le fonctionnement et l'intégration de la recherche pour relancer cette dernière une fois sur la liste (le composant de recherche se trouvant à la fois sur la page d'accueil, mais aussi de liste). La solution aura été trouvée en demandant conseil à une connaissance pratiquant l'angular. Les composants ont pu être transformés.

Thibault aura lui rencontré beaucoup de difficultés pour la mise en place de l'architecture CSS ainsi que de son bon maintien et évolution tout au long du projet. Il aura également dû faire face à de nombreux conflits lors de différents merge de branches.

Pour Allan, la difficulté était au niveau du Front où il ne parvenait pas à bien utiliser les différentes fonctionnalités des composants et leur liaison avec les "services" et "models" notamment sur la partie concernant le paiement des clients. Il a notamment été aidé par Domonkos et Nathan afin de pouvoir lui permettre de monter en compétence dessus plus rapidement, même s'il avoue avoir toujours un peu de mal à travailler côté "Front". Il avait également quelques problèmes avec certaines fonctionnalités de git qui se sont rapidement effacé à force de pratique.

En ce qui concerne Domonkos, la difficulté était la nouveauté de la technologie du Back, qu'il n'avait jamais utilisée avant. Heureusement il a pu vite apprendre les bases, et n'a pas eu beaucoup de difficultés avec cela plus tard. Une autre difficulté qu'il a eue pendant le projet était concernant la mise en forme de l'affichage des données au Front, mais il a été beaucoup aidé par ses collègues.

Une autre difficulté concernait la partie loueur, notamment l'inscription avec la récupération des données du formulaire. Un blocage a été identifié du côté d'Amine à ce sujet. Après l'intervention de Nathan, Amine a pu être débloqué.

D'une manière plus générale, la gestion du projet n'a pas été la plus optimale du fait que l'équipe a misé sur une autonomie de ses membres. Le gros inconvénient était que certaines tâches ont pris plus de temps qu'estimées lors du chiffrage dû généralement soit à un blocage soit à une perte de temps sur des tâches annexes. Avec un peu de recul sur le projet, la solution aurait été d'identifier un développeur avec une double casquette de "chef de projet" afin de mieux cadrer l'avancement des tâches.

Matrice de compétence de l'équipe en fin de projet (sur 3 points)

En fin de projet, on voit que l'équipe a globalement progressé sur les différentes technologies utilisées. Cette progression est, selon nous, en grande partie liée au "pair programming" en début de projet et l'entraide qu'il y a eu au sein du groupe tout au long de ce projet.

Membre / Techno	Angular	Node JS	SQL	Git
Allan	1.5	2	2.5	2.5
Thibault	1.5	2	3	2.5
Nathan	2.5	2.5	3	2.5
Amine	1.5	1.5	3	2
Domonkos	2	2	2.5	3

Matrice de compétence de l'équipe à la fin du projet

Idées d'ouverture

Bien que notre site soit assez complet, toutes les fonctionnalités que nous aurions voulu implémenter ne l'ont pas été, ce principalement par manque de temps et de moyen.

En voici quelques-unes non exhaustives :

- Un système d'avis client et de notation des loueurs et des produits
- La possibilité de modifier ses informations personnelles (que ce soit pour le client ou le loueur)
- Une amélioration visuelle de la page 404
- La possibilité de commander plusieurs produits.
- Une amélioration de la protection du site
- La possibilité pour les loueurs de modifier leurs produits, et de les supprimer
- La possibilité pour l'administrateur de supprimer des produits ou loueurs

Conclusion

Ce projet aura été pour nous l'occasion de réaliser un site se rapprochant le plus d'un site d'e-commerce, ce qui a permis de mettre en valeur de nombreuses fonctionnalités essentielles, mais pas toujours évidentes à décerner (telles que les systèmes d'inscription et de connexion pour des différents usages de l'application).

Il nous aura fait remettre en perspective l'utilité et même, la vitalité d'un chef de projet dédié, ce afin de veiller au bon fonctionnement de l'application, et éviter des retards dans le développement.

Enfin, nous avons tous pu progresser sur les technologies, principalement en Angular et en Node JS, ce qui est un atout non négligeable, ces technologies et plus généralement celle des webservices prenant de plus en plus d'ampleur dans le développement.