

Bethus Lilian
Magne Nicolas
Lafont Paul
Rousseau Yohan
Pasquereau Milan

Rapport Projet Qualité Logiciel

Sommaire

I°) Contexte du projet

II°) Structure

III°) Méthodologie

IV°) Difficultés rencontrées

V°) Conclusion

I°) Contexte du projet

L'objectif de ce projet est de développer une application Web permettant la gestion d'emprunts de matériels informatiques (téléphones, tablettes, pc) pour la société LocaMat. Les utilisateurs de l'application pourront consulter, modifier, ajouter, supprimer ou réserver du matériel informatique.

Ce projet s'inscrit dans une démarche interne à l'entreprise LocaMat et l'application développée sera utilisée par les collaborateurs de l'entreprise qui peuvent être soit des utilisateurs soit des administrateurs.

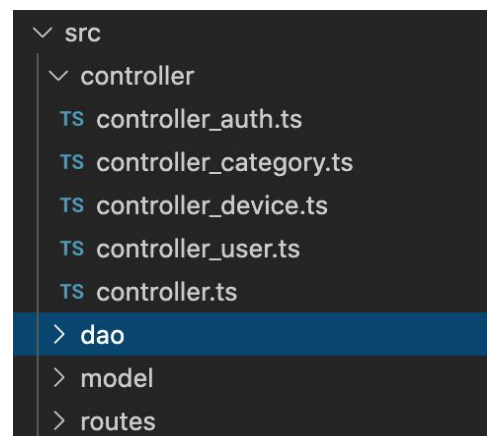
II°) Structure

Ce projet étant une application web, il est divisé en 2 parties : le front-end et le back-end. Le front-end est la partie avec laquelle l'utilisateur final de l'application va interagir. Le back-end est la partie qui permet le bon fonctionnement des fonctionnalités de l'application (qui va permettre d'interagir avec la base de données notamment). Le dossier dist contient la partie back-end et le dossier static contient la partie front-end

Back-End:

Nous avons choisi d'utiliser à travers ce projet le pattern MVC: modèle-vue-contrôleur. La structure de la partie back-end est la suivante: des packages controller, dao, model et routes qui eux mêmes contiennent les fichiers typescript:

Également, un dossier tests contient les tests portant sur les routes et sur le modèle de l'application.

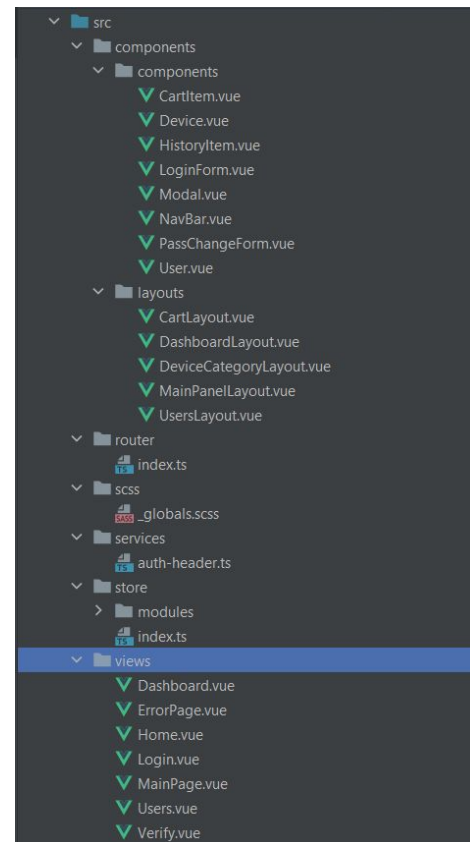


Front-End:

Nous avons choisi d'utiliser les standards à la mode pour les frameworks front-end et optant pour de la programmation orientée composant. Les composants sont séparés en deux classes : les composants purs, les composants layout. Les composants purs ne comprennent pas d'autres composants et les composants layout servent à la mise en page entre les différents composants purs.

Aussi nous avons intégré des extensions pour ne pas écrire du code trop redondant avec le service d'authentification qui ajoute les tokens utilisateurs dans les requêtes pour l'API. Nous avons aussi des extensions propres à VueJs comme le routing avec 'vue-router' et le store avec 'vuex'. Le routing n'as pas d'agencement particulier mais le store est divisé en plusieurs modules servant au management : du panier d'un utilisateur, de l'authentification, de la liste des appareils.

Enfin le dossier 'Views' contient les vues utilisées pour le router



III°) Méthodologie

La première étape a consisté à identifier les exigences du projet. Nous nous sommes donc mis tous ensemble afin de définir les fonctionnalités qui seront présentes dans l'application. Une fois que le périmètre de l'application fut bien déterminé nous avons pu ensemble décider de quelles technologies nous allions utiliser. Il en a résulté une application web interagissant avec une API REST.

L'appli étant distinctement divisée en deux, Nicolas et Paul se sont chargés de concevoir les interfaces tandis que Lilian, Milan et Yohan se sont occupés de concevoir

le modèle relationnel de la base données, le modèle de l'application ainsi que la définition des routes pour l'API REST.

Lors de la phase de développement nous avons plus ou moins gardé cette répartition. Nicolas s'est occupé du développement des IHM, de la mise en forme des données ainsi que la communication avec l'API.

De l'autre côté Lilian s'est occupé de la classe User, Milan de la classe Device et Category et Yohan de la classe Authentication, tout en aidant sur les classes précédentes, après avoir mis en place les outils NPM pour le projet. Le développement du back consistait à faire le lien entre les routes de l'API et les contrôleurs, la mise à jour du modèle et de la base de données.

La phase de test a démarré lorsque la majeure partie du backend était achevée. C'est Paul qui s'est occupé d'écrire et exécuter les tests unitaires sur la partie backend, épaulé par Yohan, en plus des tests d'intégration sur la partie frontend. De son côté Lilian s'est occupé de répertorier tous les tests dans un cahier de test.

Concernant les outils utilisés pour les tests nous avons utilisé ESLint pour l'analyse syntaxique, NYC pour le taux de couverture de test sur le projet. Nous avons utilisé Mocha pour gérer les structures des tests et Chai qui nous a fourni les outils pour les assertions lors des tests unitaires. C'est SuperTest qui a été utilisé pour vérifier les routes et les contrôleurs de l'API REST. Enfin pour le cahier de test c'est SquashTM qui a permis de le générer.

IV°) Difficultés rencontrées

Nous avons rencontré quelques difficultés de coordination lors de la réalisation du projet. Cela était en partie dû au fait que la majorité du projet a été fait durant les vacances ainsi qu'en distanciel. Il était donc plus compliqué pour nous de communiquer afin de se répartir les tâches, d'évaluer notre avancement ainsi que de discuter sur les problèmes et conflits que nous rencontrions.

Par exemple, un point qui nous a beaucoup retardé est le fait que nous n'avions pas discuté à l'avance de façon claire, précise et écrite (dans le cahier de spécifications notamment) dans quel format et par quelles URL les données transitent entre l'application web (front-end) et l'API REST (backend).

Cela nous a donc valu plusieurs réécritures partielles de nos méthodes aussi bien dans le backend que dans le frontend ainsi qu'une partie de notre temps de travail sur le projet "gâchée".

Du point de vue technique, certaines technologies que nous utilisions étaient complètement nouvelles pour certains d'entre nous (TypeScript, Express, Vue...). Cependant, nous n'avons pas rencontré de difficultés majeures ou insurmontables pour nous car nous avons su nous entraider sur la fin du projet afin de mettre en commun nos points de blocage et les surmonter ensemble.

V°) Conclusion

Les objectifs de ce projet ont été remplis. L'application web est fonctionnelle et prête à être déployée. Certaines fonctionnalités n'ont pas pu être implémentées faute de temps mais les fonctionnalités principales de l'application sont parfaitement fonctionnelles. Des tests ont été établis sur la partie back-end mais également front-end et permettent d'assurer la validité du développement de l'application.

Ce projet a permis à l'équipe de développement d'améliorer ses compétences en termes de développement web et également d'avoir une meilleure approche des tests portant sur une application web.