

# Document de synthèse :

## Qu'est-ce qui nous a plu ou déplu ?

- Utiliser de nouvelles technos, pouvoir les apprendre dans le cadre d'un projet en autonomie
- Le fait de devoir utiliser des outils utiles et modernes et utiles dans le travail de groupe (GitHub)
- Très bonne ambiance au sein de l'équipe mais des difficultés à avancer progressivement à cause justement d'une ambiance un peu trop bonne

## Comment s'est on réparti les tâches ?

Nous avons découpé notre groupe de 5 personnes en 2 équipes :

- Une équipe MOA composée de Sami et Alexandre.
  - Le groupe MOA a principalement travaillé de pair en s'échangeant les idées, notamment en rédigeant les user story, , etc.
  - Mise en place de l'outil Travis : Il va permettre d'automatiser le lancement des tests unitaires
  - Docker : Docker est un logiciel libre qui automatise le déploiement d'applications dans des conteneurs logiciels. En d'autres termes il peut emballer une application et ses dépendances dans un conteneur isolé.
  - Réalisation de la documentation technique et fonctionnelle

Une équipe MOE composée de François, Guillaume et Vincent s'est partagée le travail entre la partie Front et back :

- Guillaume a fait l'identification ainsi que l'inscription et les jeux de tests unitaires sur le back-end.
- François s'est occupé de l'architecture de base de l'API (routing ainsi que la connexion à mongoDB)
- Vincent en collaboration avec François ont tout 2 travaillé sur l'implémentation de l'objet todolist

## Quelles ont été les difficultés rencontrées tout au long du projet ?

- Une des difficultés principales a été d'apprendre une nouvelle technologie sur le tas : NodeJS pour la partie "back" et VueJS pour la partie "Front". Ces deux

technologies n'étaient en effet bien connues que d'une personne au début du projet.

- Les répartitions efficaces des tâches à également été un élément peu évident à gérer. savoir comment allouer le temps de chacun pour telle ou telle tâche a nécessité des ajustements continus.
- Maîtriser git, le concept du markdown et de nouveaux outils tels trello a pu représenter un travail nouveau pour certains d'entre nous.
- 

#### Comment s'est axée la communication au sein du groupe ?

- Nous avons créé une conversation pour notre groupe afin de parler uniquement du projet
- Nous avons fait des points réguliers en physique pendant les cours sur l'avancement des différentes features, et ce en particulier pour les user Stories.
- nous avons également dû faire le point sur le rapport afin de mettre nos connaissances en commun, notamment expliquer les aspects techniques de l'architecture de l'appli à l'équipe MOA

# Spécifications fonctionnelles :

## User story de notre To do list

En tant **qu'utilisateur lambda** :

- Je voudrais pouvoir organiser et hiérarchiser des idées/tâches au sein de blocs et sous blocs afin d'organiser mon travail
- J'aimerais pouvoir indenter librement ces blocs afin d'apporter une classification efficace
- J'aimerais pouvoir ajouter des effets de style aux caractères ( Gras, souligné, Italique, police, taille, etc.) afin d'améliorer la lisibilité de ma To do list
- J'aimerais pouvoir retrouver mes todolist après avoir fermé le navigateur (système d'authentification et de sauvegarde) afin de ne pas perdre le travail que j'ai fait.
- J'aimerais pouvoir drag and drop des blocs afin de pouvoir basculer facilement des tâches d'une branche de ma todolist à une autre facilement.
- J'aimerais ajouter une tâche et modifier son contenu juste en cliquant dessus (sans passer par un formulaire) afin que l'outil soit plus rapide et facile à utiliser

En tant que **chef de projet** :

- je veux pouvoir avoir une visibilité claire sur un certain nombre de tâches successives à accomplir dans le but de pouvoir m'organiser efficacement avec mon équipe
- Je veux pouvoir créer un compte et m'authentifier sur le site, afin que je puisse choisir qui peut voir ma to-do-list
- Je veux pouvoir gérer les droits des personnes dont j'ai autorisé l'accès auparavant (droit d'écriture, de lecture) afin que tout le monde ne puisse pas modifier ce qui a été fait.
- Je veux pouvoir avoir un historique des actions qui ont été faites dans ma todolist (par qui, quand et quoi).

En tant que **chef de projet et/ou développeur** :

- je veux pouvoir donner une valeur de temps à chacune des tâches de ma to-do-list pour gérer les priorités
- je veux pouvoir affecter la valeur réelle de temps que j'ai finalement consacré à la tâche (après l'avoir finie).

Échelle de points :

- Pour la période de réalisation du projet, nous allons définir 100 points qui représentent la durée du projet.
- Chacune des fonctionnalités devront avoir un nombre de points fixés en fonction de la durée estimée de la tâche. Ainsi, une fois que les développeurs auront donné leurs estimations, nous prioriserons les fonctionnalités à développer.

## Spécifications techniques :

L'application repose sur une architecture MVVM (Model-View-ViewModel.)

La Vue reçoit toujours les actions de l'utilisateur et interagit seulement avec le ViewModel. Le Modèle communique avec le serveur et notifie le ViewModel de son changement.

Le ViewModel s'occupe de :

- Présenter les données du Model à la Vue,
- Recevoir les changements de données de la Vue,
- Demander au Model de se modifier.

Côté client, nous avons utilisé la technologie VueJS. Elle consiste à utiliser du javascript pour récupérer des informations du back end, et donc pouvoir afficher des informations dynamiques.

Côté back nous avons utilisé un serveur en node js.

Concernant la base de données, nous avons utilisé une base en MongoDB.

On a voulu réaliser de l'intégration continue et du déploiement continu. Concernant l'intégration continue, on a quasiment réussi à la faire. Travis lance les tests et lance les commandes pour tester l'environnement. Cependant, il y a un problème d'accès au port pour la base de données.

Concernant le déploiement continu, on a essayé de déployer Docker. Cependant, nous ne connaissions pas la technologie, et nous nous y sommes pris trop tard. Nous avons tout de même appris les bases du déploiement continu, et le fonctionnement de Docker.

Lors de l'utilisation de VueJs nous avons réalisé que ce Framework permet une personnalisation poussée de composants de base, encapsulés les uns dans les autres. Séduits par ce système, nous nous sommes lancés sur des terrains précis et soucieux du détail, nous faisant sacrifier énormément de temps, ce que nous n'avons réalisé que bien plus tard. Nous avons notamment personnalisé les champs d'input pour qu'ils se gèrent tout seuls, et qu'on puisse définir leur comportement grâce à un simple appel d'une méthode.

De plus, l'architecture Server-Front/API-Back est selon nous efficace car elle permet d'avoir une API Back-End indépendante du Front, ce qui nous permettrait éventuellement de créer une application mobile, ou une application desktop. Cependant, nous avons perdu énormément de temps sur la gestion des en-têtes liés au CORS (Cross-origin resource sharing). En effet, pour que deux API communiquent via des requêtes REST en local, il est nécessaire d'envoyer les bonnes en-têtes. Nous avons géré cela grâce à un middleware au sein de node

```
22  var allowCrossDomain = function(req, res, next) {
23      res.header('Access-Control-Allow-Origin', '*');
24      res.header('Content-Type', 'text/html; charset=utf-8');
25      res.header('Access-Control-Allow-Credentials', true);
26      res.header('Access-Control-Allow-Methods', 'GET,PUT,POST,DELETE,OPTIONS');
27      res.header('Access-Control-Allow-Headers', 'Content-Type,Authorization,Origin');
28      next();
29  }
```

# Manuel utilisateur :

Quelles sont les fonctionnalités et comment marchent-elles ? faire des procédures dans la mesure du possible, même si ça va être mince dans le mesure ou on fait une todolist

Ecran d'accueil de la Todolist, il est possible ici de s'inscrire ou de s'authentifier cette page redirige soit sur la création d'une première todolist en cas de première connexion

Bienvenue sur todo

Login

Pseudo

Mot de passe

Mot de passe

Connexion

Login

Pseudo

E-mail

E-mail

Mot de passe

Password

Confirmer votre mot de passe :

Confirm password

Inscription

Choix des todolist enregistrées après le login

## Mes todo

ma todo	<div>Modifier</div> <div>Gérer les accès</div> <div>Supprimer</div>
Tata	<div>Modifier</div> <div>Gérer les accès</div> <div>Supprimer</div>
Toto	<div>Modifier</div> <div>Gérer les accès</div> <div>Supprimer</div>

Création des taches de la todolist, possibilité de drag and drop, les taches et de voir ce qu'elles contiennent

Type a task

+

Add a description to your task

Get your list a name

Save List

Test



Retours personnels :

Sami : Ce fut une excellente matière, probablement l'une des meilleures que nous avons eu. Le cours était vraiment bien (même si une partie plus détaillée sur les tests unitaires aurait pu être pas mal). J'ai beaucoup aimé le fait qu'il y ait une partie fonctionnelle aussi.

Dans ce cours, les gens qui n'aiment pas trop le code n'ont pas été délaissé, et on avait vraiment moyen de s'organiser comme nous le souhaitions. J'aurais vraiment voulu avoir plus de cours comme ça. On a appris beaucoup de choses intéressantes (la partie Travis et Docker pour ma part notamment).

Je regrette de ne pas avoir plus travaillé la matière, j'ai travaillé en cours quand même (à part le dernier, mais je n'étais vraiment pas en état :/), mais j'aurais dû beaucoup plus travaillé en dehors.

En tous cas, continuez votre cours comme il est actuellement, c'était vraiment sympa.