Parler du fait qu’on réutilise le starter kit existant qui n’est pas fait pour le micro-service et qu’on l’adapte en micro-service.

Couverture

Parler du fait que grâce à Amilapp, l’entreprise va enfin créer ses propres packages sur npmjs, évite la redondance de code dans les projets, surtout dans un projet micro-service où chaque service est une API.

Permet de se préparer au Cloud, Amiltone ne l’utilise pas pour le moment.

Micro-services adaptés à la méthode agile, faire des livraisons à chaque fin de sprints. Grâce aux micro-services, il est même possible de livrer et de mettre en route directement l’application sans que le tout soit prêt.

Explications sur les micro-services (11 juin 2020) : [https://www.technologies-ebusiness.com/enjeux-et-tendances/architectures-micro-services-objectifs-benefices-defis-partie-1](https://www.technologies-ebusiness.com/enjeux-et-tendances/architectures-micro-services-objectifs-benefices-defis-partie-1%20(11) <https://www.technologies-ebusiness.com/enjeux-et-tendances/architectures-micro-services-objectifs-benefices-defis-partie-2>

Parler du développement de Babyltone V2 quand j’étais en 3ème année, montrer l’importance d’un chef de projet, d’un lead dev, des MR, etc… Montrer qu’Amiltone à appris de ses erreurs et à amélioré le déroulement des projets interne.

Parler du fait que le lead Dev est parti en mission (probablement à cause du confinement)

Parler du possible passage sur ESlint au lieu de tsLint qui est déprécié : permettra une meilleure réutilisation avec une meilleure visualisation des erreurs dans le code.

Présentation de l’entreprise

Question à poser à Alex (idéalement) :

* Qui sont les concurrents d’Amiltone ?
* Quels partenaires ?
* Quelle politique de développement ?
* Organigramme de la société
* Comment se passe un projet client, du début à la fin (va me permettre de comparer avec le projet interne) (peut-être à demander aux chefs de projet plutôt)
* Quels ont été les plus gros clients au cours de l’histoire d’Amiltone (avec rapide explication du projet) ?
* Demander plus de description sur la branche ingénierie industrielle d’Amiltone
* Quelques chiffres (chiffre d’affaire sur 2019)
* Pourquoi s’arrêter à 300 salariés ?
* Parler de la branche design, pourquoi avoir fondé studio Bluck ?
* Pareil pour DNR ?
* Pourquoi sommes nous partis sur les microservices pour le backend d’Amilapp ?
* Comment a été gérée la crise du coronavirus ?

Présenter les différentes certifications qu’à acquis Amiltone.

Principaux éléments relatifs à l’entreprise notamment ceux qui ont un rapport avec le sujet du mémoire :  
Entreprise assez jeune dans son état d’esprit et qui a à cœur d’innover dans les nouvelles technos, ce qui m’a permis d’apprendre énormément (parler du panel de technologie d’Amiltone,

Analyse du fonctionnement de l’équipe de travail :

Organigramme :

Directeurs  
commerciaux – Chefs de projets  
lead dev  
dev

Fonctionnement :  
Suivi de la méthode agile du mieux possible sachant que je travaille sur un projet interne et donc moins « important » qu’un projet client. Daily meeting tous les matins avec toutes l’équipe, le chef de projet, ainsi que le lead dev de temps en temps, et designer qui s’occupe des maquettes du projet s’il y en a des nouvelles. Sprint de 2 semaines pour permettre une meilleure réactivité aux possibles évolutions et améliorations.   
Lorsqu’il y a des nouvelles grosses évolutions avec des nouvelles page sur le site web, grosse réunion avec chef de projet, designer et les développeurs. On parle des maquettes, de ce qui est faisable ou non, plus ergonomique, chacun peut donner son avis, tout le monde est écouté. Nouvelle réunion avec les développeurs et le chef de projet qui a prévu un tableau avec toutes les nouveautés, estimations de toutes les tâches. Etant sur le projet depuis le début, je m’occupe de créer les différentes tâches sur le tableau Jira. A chaque fonctionnalité développée, une merge request doit être faite avant de merger sur la branche principale.

Communication :  
utilisation d’outils professionnel pour la communication comme Teams. Communication très facile avec tous les développeurs puisque nous sommes tous dans le même bureau. Possibilité de demander de l’aide à tout le monde. Si quelque chose ne va pas, que ce soit au sein de l’équipe ou autre part, possibilité d’en parler au chef de projet qui essaiera de changer les choses. L’entreprise demande des retours à intervalles réguliers à ses développeurs ce qui rentre bien dans la méthode agile.

Condition de travail :   
dans un openspace avec environ 8 personnes. Climatisé.

Relations :  
De nouvelles têtes arrivent sur le projet de temps en temps. Besoin qu’ils se forment sur les nouvelles technos, plus simple de se former sur un projet interne qu’un projet client. L’équipe change donc souvent. De mon point de vue, cela peut faire perdre pas mal de temps au projet car je dois expliquer le projet aux nouveaux arrivants, les former, etc. Mais, point positif, cela me permet de former ces personnes justement, j’apprends de nouvelles choses aussi et finalement, j’aime bien le faire.

Spécificités :   
Projet interne donc à priori pas de deadline mais plutôt des objectifs.

Répartition des tâches :  
chacun peut choisir la tâche qu’il préfère accomplir tout en respectant l’ordre logique de réalisation. Il faut d’abord créer l’architecture du projet avant de commencer à créer une page de connexion par exemple.

Place du stagiaire.

Sur le projet Amilapp depuis 2 ans maintenant. Un des piliers du projet.  
Lead dev sur le projet uniquement. Je m’occupe de corriger les merge request faites par les autres développeurs de l’équipe. Quant à moi, j’envoie les merge request au lead dev de la web factory.

**Analyse du contexte**

Présentation générale de l’environnement :  
Amilapp vient de sortir en version 1.0. C’est une application qui permet une communication de l’entreprise envers ses collaborateurs plus simple et plus efficaces que les habituels courriels. Elle permet par exemple la création de sondages ou d’événements, habituellement communiqués via Teams ou mail. Tous les collaborateurs pourront y répondre facilement grâce à une application iOS ou Android. Amilapp dispose donc d’une application mobile, d’un backoffice administrateur et d’un frontoffice. Je me suis occupé des parties front et backoffice. Elles ont été codées en Angular 8. Amilapp ne dispose pas de backend à proprement parlé puisque nous utilisons la solution cloud de Google : Firebase. Elle permet la création rapide d’une application sans se poser les questions du « langage » de la base de données. J’ai utilisé le package AngularFire2 qui permet une intégration simple de Firebase avec Angular. Cette solution est pratique lorsque l’on souhaite développer une petite application rapidement. Mais dès lors que l’application devient plus complexe, on arrive rapidement aux limites de Firebase qui ne se retrouve pas suffisamment mature dans son fonctionnement pour nous satisfaire. Malgré beaucoup d’optimisations faites niveau code, on se retrouve forcément bloqué par la version gratuite de Firebase, et nous sommes alors obligé de passer par la case porte-monnaie. Pour une entreprise comme Amiltone, qui a développé ses propres solutions pour la gestion des collaborateurs, gestions des voitures de fonctions, ce n’est pas dans la philosophie de l’entreprise. De plus, cela permet aux collaborateurs en inter-contrat ou aux nouveaux arrivants de se former.

Comme Amilapp est un projet interne, un projet sans deadline, qui permet de progresser, le but est d’utiliser de nouvelles technologies, que les collaborateurs n’ont pas encore l’habitude d’utiliser. C’est le cas des micro-services. Le backend sera donc fait en micro-services et remplacera complètement Firebase. Le projet servira de base de lancement à cette façon de créer un projet.

Chez amiltone, le département web, la web factory, a créé un starter-kit, un projet de base avec énormément de fonctionnalité de base, tels que les différentes façons de se connecter en tant qu’utilisateur, une gestion de rôle, d’envoie de mail, de planificateur de tâches, qui peuvent potentiellement servir sur n’importe quel projet client. L’objectif du starter-kit est d’éviter de repartir de zéro à chaque nouveau projet.

Amilapp permettra la création d’un nouveau starter-kit, basé sur l’ancien, fonctionnant grâce aux micro-services.

Analyse du contexte

Acutalité dans le secteur de l’informatique :

Microservices de plus en plus en vogue, le cloud pareil.

Problématique : Comment adapter une application existante en marque blanche grâce aux micro-services.

Possible nouvelle problématique ? Comment créer un starter-kit backend en NodeJS grâce aux micro-services. OU Mise en place d’un starter-kit évolutif (en fonction des besoins) compatible avec les micro-services

Connaissances spécifiques nécessaires pour traiter le sujet du mémoire :

« Microservices are small, autonomous services that work together » (18 juillet 2020) <https://www.oreilly.com/library/view/building-microservices/9781491950340/ch01.html>

Méthodes habituellement utilisées pour une situation présentant des similitudes.

Parler de springboot (le starterkit java) et le .NET (démarrage d’un projet vide, possibilité de sélectionner une partie authentification soit déjà faite.

(11 Juillet 2020) <https://github.com/sahat/hackathon-starter>

(11 Juillet 2020) <https://github.com/w3tecch/express-typescript-boilerplate>

Exposé des décisions prises et des interventions menées par le stagiaire pour résoudre le problème

Les micro-service ?

**Définition des micro-services** :  
Le but de l’architecture est de découper une application en plusieurs petits services, autonomes et indépendants les uns des autres, contrairement à une architecture monolithique où l’application forme un tout indissociable. La plupart du temps, chaque service expose une API REST qui sera accessible par l’interface utilisateur ou n’importe quel autre micro-service. Chaque service aura alors son propre environnement et pourra être complètement différent des autres, que ce soit au niveau des technos utilisés ou des versions de celles-ci. On peut très bien imaginer, par exemple, un service codé en NodeJS avec une base de données MongoDB qui tourne grâce à Nginx et un autre service codé en Java avec une base de données Postgresql qui tourne grâce à Apache. Pour la plupart du temps, Docker est utilisé pour les micro-services.

**Avantage des micro-services :**

Est adapté au cloud, qui est de plus en plus utilisé.  
meilleure scalabilité : petit composants, autonomes et légers, simple à conteneuriser et à répliquer.  
Lors d’une mise à jour, il sera possible de cibler directement le micro-services en question

Inconvénients :  
très récent, pas de définition claire, évolue très rapidement, à besoin d’une veille technologique plus importante que pour les autres starterkit (spring, etc)

Pourquoi les micro-services ?

Nouvelle techno que personne ne connait dans l’entreprise.  
Sur le starterkit actuel, on se retrouve avec énormément de fonctionnalité et on devra supprimer celle qu’on ne veut pas utiliser, avec le risque casser des fonctionnalités. Avec les micro-services, on commencera un projet avec une brique de base très petites, et on pourra réutiliser d’autres micro-services pour les modifier ou les réutiliser comme on le souhaite.

Démonstration d’une originalité dans l’élaboration et la mise en œuvre de la solution :

Différents des solutions existantes sur le marché, adapté au besoin de l’entreprise.

Démonstration d’une originalité dans l’élaboration et la mise en œuvre de la solution

Analyse de l’approche choisie :

Demander aux chefs de projet le temps que prenait un nouveau projet sans starter-kit, puis avec starter-kit. Va permettre d’analyser les avantages et inconvénients de chaque méthode.

Comment serait réutilisable la solution dans une autre infrastructure ou sur un autre projet.