David Golay

IUT METIERS DU MULTIMEDIA   Dijon

Mémoire d’alternance

|  |  |
| --- | --- |
| Rapport de stage de fin de DUT Informatique | |
| Poste occupé | Développeur Front-end |
| Période | 4 avril 2022 – 8 septembre 2023 |
| Entreprise d’accueil | Owlnext 12 rue du Golf, Quetigny |
| Maître d’apprentissage | Arnaud Francois |
| Signataire de la convention | Laurent Maluski |
| Tutrice pédagogique | Céline Moreira Dos-Santos |
| Etablissement scolaire | IUT MMI de Dijon 7 Boulevard Docteur Petitjean, Dijon |

  Remerciements

Tout d’abord je souhaite remercier mon maitre d’apprentissage, Arnaud Francois ainsi que le gérant d’Owlnext, Laurent Maluski, sans qui cette expérience aurait été . grâce à qui j’ai d’abord trouvé mon stage qui s’est poursuivi pour avoir rendu cette alternance possible, pour m’avoir encadré, et partagé volontairement ses connaissances sur des technologies innovantes.

Je remercie Quentin Chaplier qui a été mon chef de projet pendant la première période de cette alternance. Il m’a aussi donné bon nombre d’explications sur le fonctionnement de son domaine de compétence, ce qui par la suite, m’a permis de mieux comprendre l’étendue des responsabilités de chacune de nos disciplines respectives.

Je remercie à Axel Breniaux pour m’avoir épaulé et pour tous ses conseils très pratiques et son implication qui ont souvent contribué à me faire progresser plus rapidement.

De manière générale, je remercie toute l’équipe d’Owlnext pour la richesse des solutions qu’ils apportent.

Pour finir, Je remercie Monsieur Noirot pour son naturel bienveillant ainsi que l’équipe pédagogique et administrative de l’IUT Métiers du Multimédia (Mention Métiers du numérique) de Dijon pour la formation qu’elle m’a donnée ainsi que pour l’ouverture d’esprit dont elle a fait preuve.

Table des matières

[A. Introduction au contexte 4](#_Toc131615958)

[présentation générale de owlnext 4](#_Toc131615959)

* [La société Owlnext 4](#_Toc131615960)
* [L’organisation de l’entreprise 5](#_Toc131615961)
* [Moyens de collaboration interne 6](#_Toc131615962)

[Activités d’Owlnext 7](#_Toc131615963)

* [Formation de tiers & Intervention 7](#_Toc131615964)
* [Accompagnement & Conseil orienté tech 7](#_Toc131615965)
* [Le développement de solution (Activité majoritaire) 8](#_Toc131615966)
* [Provenance des clients 8](#_Toc131615967)

[relationnel client 8](#_Toc131615968)

[Thématique de l’alternance 10](#_Toc131615969)

[B. Thématique de l’alternance 10](#_Toc131615970)

[Introduction à la méthode de travail 10](#_Toc131615971)

* [Environnement de développement 10](#_Toc131615972)
* [Vue d’ensemble des tâches effectuées 11](#_Toc131615973)

[C. Mes missions 13](#_Toc131615974)

* [Impulsa 13](#_Toc131615975)
* [Une application riche en fonctionnalité 13](#_Toc131615976)

[le projet AgriEchange 15](#_Toc131615977)

* [Transition technologique 15](#_Toc131615978)

[De nombreux apports personnels 16](#_Toc131615979)

* [Identification des besoins 16](#_Toc131615980)
* [Réflexion autour de la généricité 17](#_Toc131615981)
* [Rétention des informations essentielles 18](#_Toc131615982)
* [Méthode et priorisation des tâches 18](#_Toc131615983)
* [Provenance des erreurs 18](#_Toc131615984)
* [La communication écrite 19](#_Toc131615985)
* [Prévoyance et feuille de route 19](#_Toc131615986)

[Les difficultés rencontrées 22](#_Toc131615987)

* [La mise à jour progressive de l’architecture 23](#_Toc131615988)
* [Passage à une technologie plus stricte 24](#_Toc131615989)
* [L’ajout de témoins visuels 24](#_Toc131615990)

[D. Conclusion 25](#_Toc131615991)

[Apports personnels 25](#_Toc131615992)

* [L’importance du partage 25](#_Toc131615993)
* [La formation continue 26](#_Toc131615994)
* [La stratégie du design to cost 26](#_Toc131615995)

[E. Annexes 27](#_Toc131615996)

* [Front-End 27](#_Toc131615997)
* [React.js 27](#_Toc131615998)
* [Next.js 28](#_Toc131615999)
* [Flutter 28](#_Toc131616000)

# Introduction au contexte

## présentation générale de owlnext

### La société Owlnext

Etablie 12 rue du Golf à Quetigny, Owlnext est une jeune société accompagnant différentes organisations dans l’indentification, l’analyse et la réponse à leurs problématiques informatiques. Elle accorde une importance absolument cruciale à la communication avec ses clients. La place donnée au dialogue et au partage de l’information est très marquée dans les interactions qu’entretiennent les équipes avec leurs interlocuteurs. L’idée étant de les accompagner sur une logique de long terme et de leur proposer une solution viable rapidement (MVP[[1]](#footnote-1)). Cette évolution est facilitée par une conception bien anticipée de l’architecture système et logicielle, en perpétuelle amélioration.

### L’organisation de l’entreprise

Cette entreprise a été créée par quatre anciens collègues et amis : Chloé Cochet, Arnaud Francois, Adrien Gras et Laurent Maluski, qui, depuis le 1er juin 2021 ont progressivement vu leur effectif se doubler.

Orienté Back-end

Orienté Front-End

Figure : Organigramme du personnel

Le statut de Sarl SCOP[[2]](#footnote-2) de Owlnext confère une horizontalité des membres dans la hiérarchie. Les rôles que chacun occupe diffèrent selon les projets et les ordres du jour et l’avis de chacun peut influencer les prises de décisions collectives. Cette volonté de rassemblement dans le décisionnel est une caractéristique fondamentale qui définit cette entreprise.

Bien entendu, les chargés de projet traitant directement avec les clients, prennent les décisions nécessaires à bonne la conduite du projet dont ils ont la charge en assignant à l’ordre du jour les différentes tâches à effectuer. Bien que les développements d’application soient très denses, il règne dans ses murs, une atmosphère très propice au dialogue et à la prise de recul. La bienveillance et la confiance permet d’apaiser les tensions et de souder les membres entre eux.

### Moyens de collaboration interne

Afin d’assurer un suivi proximal avec les avancées de chacun, la communication orale directe et spontanée est très utilisée entre les membres des différents services, notamment pour les demandes ou explications nécessitant d’être détaillé oralement ce qui se complète parfois avec le dessin de schéma et d’algorithmes. Ces situations arrivent, entre autres dans les moments de briefing, démonstration, points de synchronisation et de fusion des développements faits séparément.

|  |  |
| --- | --- |
| Outils de communication et de collaboration | |
| Communication *(interne et externe)* | Discord, Slack, Teams, Outlook |
| Partage du code | Github |
| Répertoires des tâches | As[[3]](#footnote-3)ana |

## Activités d’Owlnext

L’entreprise qui m’a accueilli propose une large gamme d'activités informatiques couvrant différentes thématiques telles que la formation de tiers/intervention, l'accompagnement, l'architecture, le conseil orienté tech/veille, le développement complet de solutions, la reprise de projet et l'optimisation de processus métier. En résumé :

* Architecture
* Accompagnement / Conseil orienté tech / Veille
* Développement complet de solution
* Reprise de projet
* Optimisation de processus métier

### Formation de tiers & Intervention

En ce qui concerne la formation de tiers/intervention, l'entreprise offre des formations personnalisées pour aider des organismes publics ou privés à acquérir les compétences nécessaires pour gérer leur propre système informatique. De même, elle propose des services d'intervention pour résoudre les problèmes liés aux systèmes informatiques des entreprises.

### Accompagnement & Conseil orienté tech

Pour l'accompagnement, l'entreprise accompagne ses clients dans toutes les étapes de la mise en place et de l'utilisation de leurs systèmes informatiques. Elle propose des services de maintenance pour garantir la continuité des opérations et des mises à jour régulières pour assurer la sécurité des systèmes. Elle propose également des conseils de mise en place d’architecture performantes. L'entreprise se tient constamment informée des dernières tendances et des nouvelles technologies pour proposer les solutions les plus adaptées à ses clients. Elle offre également des services de veille pour aider les entreprises à anticiper les changements futurs et à s'adapter aux évolutions technologiques.

### Le développement de solution (Activité majoritaire)

La majeure partie du temps, l'entreprise conçoit de A à Z des solutions sur mesure pour répondre aux besoins spécifiques de chaque typologie de client, débouchant sur des projets pouvant s’étaler de deux mois à deux ans. Ces développements ne ciblent pas particulièrement de secteur, elle cherche avant tout à comprendre une typologie de client et de challenger ses idées afin de lui proposer une solution qui pourra être étendue sur le long terme. Cela lui a permis de collaborer étroitement avec des praticiens du paramédical, dentistes, administration de pharmacie, réseaux de l’agriculture, cabinet d’experts comptables, sociaux et juridiques, acteurs du sport, écoles de commerce. Enfin, dans certains cas rares, l’entreprise reprend des projets existants.

### Provenance des clients

* Le bouche à oreille
* Les clients satisfaits
* Réseaux de connaissance
* Développer Group Dijon

### Relationnel client

Owlnext se veut très proche des besoins de ces clients, cela passe par l’installation d’un climat de confiance et de chalenging perpétuel des propositions clientes. Cela permet de mieux comprendre leurs besoins et leurs attentes, ce qui peut aider à concevoir des produits et des services plus adaptés à leurs besoins. Cela peut également contribuer à améliorer la qualité des produits et des services en identifiant rapidement les problèmes et en y apportant des solutions.

Ainsi, cette entreprise peut établir des relations plus fortes et plus durables avec eux, il s’agit là de sa stratégie de fidélisation. Les clients se sentiront entendus et respectés, ce qui peut les inciter à rester fidèles à l'entreprise et à recommander ses produits et services à leur entourage.

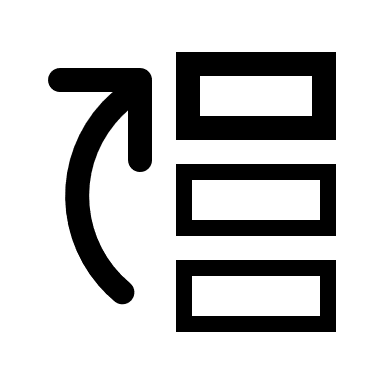
Cependant, il y a aussi des inconvénients à avoir des rapports très à l'écoute des clients. Tout d'abord, cela peut être très coûteux en temps et en ressources, car l'entreprise devra consacrer du personnel et des ressources pour recueillir les retours des clients qui de leur côté n’ont pas même ordres de priorité.

De plus, si l'entreprise est trop à l'écoute des clients, elle peut être tentée de suivre leurs demandes sans prendre en compte les implications à long terme pour l'entreprise. Cela peut entraîner des décisions prises trop rapidement, sans prendre en compte les considérations stratégiques à long terme. C’est pour cela qu’il est important de prendre en compte les commentaires des clients de manière équilibrée et d’apporter un cadre qui n’entrainera pas des surcouts de développement.

# Thématique de l’alternance

## Introduction à la méthode de travail

### Environnement de développement

Grâce à un ensemble de script et des commandes développées par Adrien Gras (Architecte et Administrateur système[[4]](#footnote-4)), notre poste de travail est configuré pour le développement. Dès lors qu’un membre d’équipe doit intervenir sur un projet, il récupère les sources existantes du projet ou les crée à partir d’un squelette sans cesse mis à jour. C’est ce qu’on appelle ‘monter une stack’. Il s’agit d’un empilement de composants logiciels nécessaires pour faire fonctionner une application web ou un client lourd[[5]](#footnote-5).

Bien que ce stage suive une certaine continuité, on peut aisément le scinder en deux temps différents. En effet, tout juste un mois après mon arrivée, je suis devenu la ressource principale en développement front-end ? J’ai pris le relais d’Axel Breniaux qui devînt une ressource annexe (sur ce projet). Dès lors, je eu l’opportunité de développer des fonctionnalités encore inexistantes.

### Vue d’ensemble des tâches effectuées

Mon rôle en tant que développeur front-end a été de participer au développement de toute la partie visible et avec laquelle les utilisateurs de l’application interagissent.

Divers

* Correction de bugs
* Ajustements graphiques
* Ajout de fonctionnalité supplémentaires sur l’existant
* Rafraîchissement de l’interface
* Ajout de fonctionnalités inexistantes

Les composants et utilitaires

* Création
* Adaptation
* Extension
* Optimisation

Les formulaires

* Règles de validation de formulaires
* Pré-remplissage des champs de formulaire
* Gestion d’erreurs
* Discrimination de l’affichage selon les droits d’accès

Les flux de données

* L’intégration de connecteurs API[[6]](#footnote-6)
* Mapping (intégration dans l’interface)

# Mes missions

### Impulsa

Owlnext m’a d’abord accueilli dans le cadre de son contrat de développement avec un groupe pluridisciplinaire d’experts-comptables, avocats et consultants proposant des services de production comptable, juridique, d’audit, de conseils. C’est une équipe constituée d’avocats inscrits au barreau de Paris mais aussi de juristes disposant d’expériences significatives en droit des affaires.

Possédant un réseau de plus de 1700 clients, cette constitution a fait appel à l’entreprise m’accueillant, afin de concevoir une **application web progressive**[[7]](#footnote-7)permettant de centraliser et d’organiser leurs différents services.

Mes missions n’ont pas débordé du contexte du développement front-end. Je n’ai absolument pas agit sur la partie back-end du projet.

### Une application riche en fonctionnalité

Cet équivaillent d’ERP étant assez vaste, vous trouverez ci-dessous un graphique légendé permettant de se faire une idée des différentes sections de l’application en concordance avec les développements sur ces derniers.

## le projet AgriEchange

Le projet agriEchange est à destination des agriculteurs et des réseaux d’agriculture, elle permet à ces derniers de faire du troc basée une monnaie nommée l’agri Euro. Elle permet à ceux-ci de partager du matériel et des services sous forme de services rendus. Elle permet aussi de réaliser de véritables transactions. D’un point de vue plus technique c’est une solution multi-plateforme (Android, IOS, Desktop, Web App) développer sur Flutter, le nouveau framework que l’entreprise souhaite désormais utiliser pour la majorité de ces projets. A l’heure actuel, je viens de passer en période d’apprentissage sur Flutter et j’intervient en temps que renfort sur des parties localisées de l’application.

### Transition technologique

Une transition sur Flutter qu’est-c eque cela implique ?

Effectuer un tel changement de paradigme technologique dans une entreprise nécessite une grande audace qui implique de sortir de sa zone de confort, d’avoir des profils de pionniers et comme celui d’Arnaud qui porte vigoureusement cette transition. C’est prendre des risques importants en évoluant sur des terrains nouveaux, difficile à quantifier en temps de travail qu’il faut justifier au client. Cela peut donner lieu à des premiers projets pouvant mal se terminer et ne pas combler les attentes des clients. Cela peut être difficile et créer des réticences pour les employés qui sont habitués à des méthodes de travail et des processus établis.

Mais en prenant du recul sur ce changement majeur, cette audace est importante pour plusieurs raisons. Tout d'abord, elle permet d'identifier et de saisir de nouvelles opportunités qui peuvent être bénéfiques pour l'entreprise et les employés. En effet, les changements de paradigme comme tel peuvent souvent conduire à des innovations, ce qui est un argument commercial d’Owlnext auprès de ces clients, services qui peuvent aider l'entreprise à se démarquer sur le marché.

## De nombreux apports personnels

### Identification des besoins

Bien que le projet Impulsa implique une grande partie de communication, d'analyse et de restructuration des besoins clients, mon travail sur ce projet dès le début de mon stage en avril 2022 était principalement axé sur les tâches de programmation sans interaction avec les clients. Ce projet a été l’un des premiers projets de l’entreprise. Il organisait des thématiques complexes relative à l’exercice comptable et toutes ces spécificités.

Étant donné la complexité de l'application sur laquelle j’étais, il était primordial de bien comprendre les enjeux auxquels le nouveau développement devait porter. La modélisation logique était principalement assurée par le chargé de projet / développeur Back-End, sans qu'il y ait d'intérêt spécifique à ce que je participe aux échanges avec le client. Par conséquent, je n'ai pas analysé les besoins clients, mais plutôt les besoins de développement nécessaires pour découper les tâches et arriver au résultat que nous avions défini ensemble.

Initialement, j'ai suivi les maquettes graphiques élaborées par le designer UX de l'entreprise et j'ai fortement utilisé l'existant pour intégrer les nouveaux écrans. Cependant, de nombreux cas d’usage à traiter ont été découvert au fur et à mesure des implémentations. Cela était majoritairement dû à un manque de spécifications et des besoins sans cesse évoluant. Cela m’a permis de réaliser à quel point il est important d’avoir un cahier des charges très spécifiques au début du projet pour ne pas avoir à trop compensé les imprévus qu’il font naitre.

Afin de répondre aux besoins de développement en constante évolution, j'ai dû m'adapter en temps réel aux demandes et aux retours des clients. Bien que j'aie eu à cœur de réaliser des refactorisations de code pour anticiper d'éventuels remaniements de l'interface, j'ai rapidement compris que cette pratique était contre-productive dans le contexte de contraintes de délais et de ressources allouées. J'ai donc appris à développer rapidement du fonctionnel, tout en gardant en tête les priorités de temps et de budget.

### Réflexion autour de la généricité

La réutilisation d’un bout de code dans d’autres environnements est très souvent une idée sur laquelle mise les développeurs, cela permet d’aller plus vite en dupliquant le composant et de réduire les coûts de développements. Mais force est de constater que ce n’est pas si simple que cela.

Il peut arriver que l’on pense à tort qu’en réutilisant un composant dans un autre projet cela aura pour effet de réduire les coûts de développement. Cela n’est pas valable dans tous les cas et pour posséder un composant unique réutilisable dans plusieurs autres outils informatiques, il faut être prudent.

Notamment sur la première moitié de mon alternance, j'ai beaucoup cherché à rendre mes développements génériques majoritairement parce qu’il s’agissait d’une bonne pratique apprise en contexte scolaire, mais j'ai vite compris que cela pouvait être coûteux en temps et en ressources, voir même contre-productif dans les typologies de projet auxquels j’étais confronté. Souvent les cas étaient trop spécifiques et donc difficile à généraliser.

Chercher à rendre mes composant réutilisable dans le plus de contextes différents nécessite de bien réfléchir à l’implication temporelle et budgétaire que cela va impliquer et finalement.

J'ai ainsi appris à trouver des solutions de contournement moins coûteuses pour répondre aux exigences et à opter pour des solutions non-bloquantes. Au lieu de rendre un composant générique, il vaut mieux privilégier un découpage atomique de celui-ci.

Un des compromis les plus efficaces que j’ai pu trouver était de découper des éléments très atomiques en petites brique isolées dès lors que des réutilisations au sein de la même application était certaines. Par exemple, j’ai isolé des logiques de validation de champ de formulaire, des formateurs de date et de temps passé, des composants permettant de générer des adresses, des interpréteurs de graphiques, des petits morceaux de codes redondants.

### Rétention des informations essentielles

Afin de gagner en autonomie et d’éviter le plus possible à mes collègues de se déconcentrer de leur propre mission, j’ai vite compris l’enjeu de la prise de note efficace, pour savoir où retrouver les informations pertinentes pour que je puisse travailler en autonomie une fois le breiffings terminés. En outre, c’est indispensable pour bien comprendre et d’avoir une vue d’ensemble sur les différentes tâches dispatchées à différents moments de la journée. Cette prise de note, m’a beaucoup rassurée et m’a servi, par exemple à localiser dans l’arborescence des applications des exemples desquels je pouvais m’inspirer pour développer.

### Méthode et priorisation des tâches

Dans des délais impartis, il faut nécessairement faire des choix sur ce qui devra être absolument présent et consistent dans la production et ce qui pourra être simplifié ou renégocier avec le client. Sur la solution Il faut privilégier les aspects fonctionnels avant tout. Assurer que chaque étape fonctionne avant de l’embellir graphiquement. Bien sûr, il ne s’agit pas de laisser de côté l’aspect visuel, mais d’assurer un bon fonctionnement logique avant tout. J’ai appris à faire des compromis en privilégiant ce qui compte le plus en évitant de perdre trop de temps sur des détails.

### Provenance des erreurs

C’est notamment par l’observation des pratiques de débogage de mes collègues que j’ai pu progressivement déterminer les points d’attention sur lesquels se focaliser afin d’assurer un débogage plus efficace. En effet, le débogage représente un temps difficilement quantifiable qu’il faut donc désamorcer de manière pratique. C’est aussi par une meilleure compréhension du pipeline des applications qui permet de reconnaitre les nœuds où porter notre attention. En connaissance bien l’architecture, on sait mieux ou placer sa paire de jumelle, on sait d’où peuvent naitre les erreurs de programmation et donc à quel niveau les corriger.

N’ayant pas d’expérience significative en Javascript il y a tout juste un an, le fait d’avoir commencé par manipuler ce language sans utiliser une surcouche plus stricte (TypeScript) m’a permis de me pencher sur des comportements dangereux que permet la liberté totale donnée par le Javascript. Je comprends mieux les problématiques qu’il faut avoir à l’œil quand on développe en Javascript. Par exemple, les opérateurs de comparaison qu’il faut éviter. Il faut compenser le manque de typage par une logique plus La conformité des types de variables

### La communication écrite

L'un des défis que j'ai dû surmonter était de maintenir une consultation régulière de l'outil de communication interne de l'entreprise. Pendant une période prolongée, j'ai trouvé difficile de me détourner de mes tâches de programmation pour vérifier mes notifications. J'ai donc dû acquérir plus d’attention afin de m'assurer que j'étais informé des informations pertinentes en temps opportun et de pouvoir répondre aux questions pouvant faire avancer les autres.

### Prévoyance et feuille de route

Lorsque que j’ai travaillé sur le projet Impulsa, j’ai été amené à travailler de manière autonome sur de nombreuses fonctionnalité, mais celle-ci, fût celle qui m’. J’ai donc décidé de détailler l’un d’elle en particulier pour illustrer le caractère autonome dont j’ai fait preuve. La fonctionnalité DataViz correspond à un ensemble de graphique permettant de d’expliciter efficacement différents amas de données aux managers des différents services. Ainsi, il a fallu implémenter une vingtaine de graphiques dans l’application ‘Impulsa’.

J’ai choisi de parler de cette fonctionnalité car c’est un élément sur lequel j’ai pu déterminer moi-même une architecture pour répondre à un besoin précis dans un temps limité. J’ai eu le champ libre pour trouver une solution technique efficace qui une fois mis en place, devait se configurer simplement pour ajouter de nouveaux graphiques tout en offrant un degré de liberté afin de pouvoir intégrer d’autres type de graphique par la suite.

Ce système de graphique devait être configurable par d’autres personnes pendant mes périodes de cours.

Fin de contrat avec Impulsa

Les projets ont une fin.

De nombreux freins ont perturbé la bonne conduite de se projet.

* Manque de ressources alloué par les conducteurs
* Manque d’information sur le métier
* Un manque de précision
* Un manque d’expérience
* Des standards inconstants
* Difficulté liée à la complexité des disciplines juridiques et comptables
* Un projet rempli

Apprendre sur une architecture rendue obsolète

Il est important d’avoir à l’esprit qu’en informatique, les outils évoluent très rapidement, notamment dans le domaine du web. Dès lors, de nombreux facteurs entre en jeu quant aux choix des technologies qu’elles utiliseront à leurs avantages et au détriment de leur inconvénients.

Owlnext tend à externaliser ses outils

* Minidiag
* Notification API
* Demander la liste des outils externalisé

Ce découplage hermétique entre le back end et le front end est un choix judicieux et React étant facile à apprendre dès lors qu’on a des bases en HTML et en programmation,.

Toutefois, faire les bons choix de structure ne se découvre vraiment qu’à la pratique. Parfois vouloir séparé dans un fichier la déclaration de constantes peut être plus chronophage et même moins pratique (cf : les fichiers de déclaration d’action). En ayant travaillé sur une architecture plus ancienne, je comprends vraiment mieux comment des choix simples ont simplifier des pratiques redondantes.

Exemple de changements mineurs permettant de gagner de nombreuses minutes cumulées

* Arborescence des routes
* anonymiser les fichiers de Model (simplifie la duplication de composant)
* Snack

Exemple de changements majeurs

* Le typage strict (perte de temps lors de la définition des variables, énorme gain de temps lors de l’écriture de la logique)
* Création d’un reducer pour gérer le contenu des panneaux
* Composant de gestion de Rôle possible à tout les niveaux
* SecurityWrapper ?
* CustomForm

## Les difficultés rencontrées

Ce projet est l’application la plus complexe que j’ai pu observer et **analyser** jusqu’à ce jour. Comme tout projet, c’est une infrastructure qui a évolué dans le temps. C’est fascinant de se rappeler qu’il n’y avait rien de tout cela, il y a près d’un an et pour être tout à fait sincère, je reste encore à ce jour subjugué par le nombre de fonctionnalités que comporte cette application.

Bien qu’au départ, j’ai ressenti une une forme d’intimidation vis-à-vis de l’arborescence très fournie des fichiers de code et des intrications de composants, mais au fur et à mesure cette masse d’information s’est détaillé dans mon esprit.

### La mise à jour progressive de l’architecture

En un an, les standards de programmation ont évolué et se sont affranchis de certaines contraintes pour en découvrir d’autres. Dès le départ, Arnaud m’a donné pour mission de convertir des composants dans la forme de la nouvelle architecture suivant :

* Amélioration des standards
* Evolution des technologies
* Nouvelles fonctionnalités

Je comprenais l’intérêt de ce changement de motif architectural, toutefois, il y a eu beaucoup de composants parcourus à convertir. Il m’apparaît encore difficile aujourd’hui de déterminer lequel(s) améliorer en priorité. Tous les remplacer serait long et fastidieux, et représente une masse considérable de configurations à tester. C’est une répercussion du refactoring[[8]](#footnote-8).

### Passage à une technologie plus stricte

### L’ajout de témoins visuels

Dans le meilleur des mondes, les utilisateurs ont la fibre. Mais qu’en est-il des utilisateurs rencontrant des problèmes de connexion internet occasionnels ou quotidien. Quelles informations leur présenter, comment les rassurer quant à la réalisation des actions qu’ils ont déclenchées ? si aucun témoin visuel ne leur signale que leur demande a bien été traitée et qu’il faudra attendre que l’information leur revienne, ils pourront se sentir désorientés. Pour palier à cela, il faut donc ajouter un maximum de témoins visuels marquant la prise en charge et le résultat de leur action. Par exemple, Il faut leur montrer des barres de chargement ou des messages d’alerte, de réussite ou d’avertissement.

Quand il faut traverser les différentes couches de l’application, parfois des milliers de fois par jour, il faut s’armer des stratagèmes de circulation et les plus adéquats.

# Conclusion

## Apports personnels

Gain en autonomie

Prise

Priorisation des tâches

Simplification des solutions

### L’importance du partage

D’un point de vue technique, cette expérience m’a donné une vision bien plus précise des enjeux du développement Front-End. Elle m’a permis de faire progresser mon algorithmique et ma compréhension des enjeux de l’asynchronisme en javascript ~~et~~ d’acquérir des compétences sur des technologies incontournables du web.

Grâce aux réponses qu’Arnaud Francois m’apportait régulièrement, j’ai pu porter mon attention sur des problèmes d’optimisation et d’architecture qui se sont avérés être des sources d’intérêt insatiables. Souvent, lors de ses passages en revue de mon code, Arnaud s’arrêtait sur des points clés essentiels qu’il vulgarisait plus ou moins précisément. Il me donnait une explication, une amélioration possible, un équivalent respectant les derniers standards de programmation, et je ne pourrais que le remercier pour toute la matière à penser qu’il a su partager. J’ai pu renforcer mon idée qu’être bien entouré permet de se sentir à l’aise et de révéler le meilleur de soi-même.

### La formation continue

C’est lorsqu’on est entouré de professionnels du web, qu’on se rend vraiment compte qu’il reste encore énormément de choses à apprendre. J’ai particulièrement apprécié de progresser dans un environnement en clin au partage de connaissances et à la résolution des problèmes. Dans la programmation, la concentration est indispensable afin qu’aucun détail n’échappe à notre vigilance. Il est très facile d’introduire une erreur générant une faille suffisamment large pour faire stopper l’application. On peut passer à côté de certains cas d’usage[[9]](#footnote-9) que l’on n’a pas traité ou tout simplement oublié.

Conscient que je rencontrais des difficultés à suivre scrupuleusement un fil conducteur dans l’exécution de mes tâches, j’ai appris une certaine rigueur et méthodologie de façon à gagner en concentration. Cela m’a permis d’éviter une forme d’éparpillement au sein des fichiers et des développements partiels que je pouvais parfois entreprendre dans un ordre incohérent.

### La stratégie du design to cost

Le design to cost est donc une approche proactive de la gestion des coûts, qui vise à minimiser les coûts de production dès la conception du produit. C’est une méthode de conception qui consiste à concevoir un produit en prenant en compte dès le départ les coûts de production, afin de maximiser la rentabilité de celui-ci. Cette méthode implique de déterminer dès le début du processus de conception les coûts de chaque composant et de chaque étape de fabrication, et de les intégrer dans la conception du produit de manière à optimiser les coûts tout en répondant aux spécifications du client. Ce n’est pas le cas sur tout les projets d’Owlnext, cela dépend de la typologie de client, mais une tendance s’est développé au cours de mon alternancve, il s’agit de l’approche Front-first [[10]](#footnote-10) des projets, se concentrer sur l'expérience utilisateur et l'interface graphique du site web, en veillant à ce que l'utilisateur final dispose d'une expérience de navigation fluide et agréable.

Merci de votre attention

# Annexes

### Front-End

Le **Front-End** c’est la partie du code qui est reçue par le navigateur Web. Il s’agit finalement d’éléments du site web que l’on aperçoit à l’écran et avec lesquels on pourra interagir. Ces éléments sont composés de 3 langages suivant :

* **HTML**[[11]](#footnote-11)(structure)
* **CSS** [[12]](#footnote-12) (aspect)
* **Javascript**[[13]](#footnote-13) (interactivité)
* **TypeScript (surcouche de typage strict)**

### React.js

**Une image contenant texte, clipart

Description générée automatiquementReact** (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM[[14]](#footnote-14) virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité. La bibliothèque est utilisée par Netflix, Yahoo, Airbnb, Sony, WhatsApp ainsi que par les équipes de Facebook.

### Next.js

Une image contenant texte, clipart, ciel nocturne

Description générée automatiquementNext.js est une surcharge de la bibliothèque React permettant de faire du rendu côté serveur (**SSR)**[[15]](#footnote-15). Créé en 2016 par l’entreprise Vercel, Next.js permet de créer des applications web multi-plateformes performantes.

C’est un outil qui va permettre de faciliter la vie des développeurs React en réduisant les tâches de configurations courantes. L’objectif principal consiste à disposer d’une solution simple pour créer, déployer et partager des applications web.

### Flutter

1. MVP : Minimum Viable Product - Pratique souvent mise en place avec la méthode Agile. Le MVP est la version la plus minimaliste possible d’un produit. Les retours effectués permettront d’améliorer le MVP, de l’enrichir avec de nouvelles fonctionnalités, par itération, jusqu’à devenir le produit fini. [↑](#footnote-ref-1)
2. SCOP : Société coopérative de production dont les associés majoritaires sont les salariés. Réunis autour d'un même projet économique et des mêmes valeurs, ils s'impliquent totalement dans l'entreprise. [↑](#footnote-ref-2)
3. Asana – Asana est un gestionnaire de communication d'équipe. Ce produit prend en charge de nombreuses fonctionnalités, notamment les espaces de travail, des projets, des tâches, des étiquettes, des notes, des commentaires. [↑](#footnote-ref-3)
4. Administrateur Système et Réseau - En informatique, le titre d'administrateur systèmes désigne la personne responsable des serveurs d'une organisation. [↑](#footnote-ref-4)
5. Client lourd – Logiciel exécutable [↑](#footnote-ref-5)
6. API – Une API est une interface de programmation d’application, elle représente le principal protocole d'échange utilisé pour connecter les plateformes entre elles. [↑](#footnote-ref-6)
7. PWA - Les Progressive Web App sont des applications web qui combinent les fonctionnalités offertes par un navigateur web avec les fonctionnalités présentes sur les smartphones. On y constate une amélioration du temps de chargement des pages, amélioration de l’expérience utilisateur. La PWA permet des gains de temps de maintenance : elle s’administre beaucoup plus facilement qu’une application native (plus besoin de passer par un informaticien pour agir sur le contenu). [↑](#footnote-ref-7)
8. Refactoring– Redécoupage plus atomique du code. Réduit les redondances de code et sépare mieux les responsabilités de chacune des briques de JavaScript. [↑](#footnote-ref-8)
9. Cas d’usage – Use Case - désigne la manière d’utiliser un système. [↑](#footnote-ref-9)
10. L'approche "Front First" en développement web consiste à concevoir d'abord l'interface utilisateur avant de se concentrer sur le développement de la partie "back-end". [↑](#footnote-ref-10)
11. **HTML - Hyper Text Markup Langage -** est un langage composé de balises jouant le rôle de**squelette de nos pages Web. Par e**xemple : un titre, un paragraphe, une image ou une liste. [↑](#footnote-ref-11)
12. Le **langage** **CSS** - **Cascading Styles Sheets** - est un langage qui va **définir la forme de nos pages Web** et permettre de les décorer. Par exemple : régler les couleurs, les polices d’écriture, la mise en page, les marges etc. [↑](#footnote-ref-12)
13. Javascript - Le langage Javascript est responsable de l’interactivité et de la logique qu’il y a derrière nos **pages web**. Par exemple : ouverture de menu. [↑](#footnote-ref-13)
14. DOM - Le Document Object Model ou DOM (pour modèle objet de document) est une interface de programmation pour les documents HTML, XML et SVG. Il fournit une représentation structurée du document sous forme d'un arbre et définit la façon dont la structure peut être manipulée par les programmes, en termes de style et de contenu. [↑](#footnote-ref-14)
15. SSR – Server Side Rendering – Rendu côté serveur - méthode de rendu où toutes les ressources de votre page sont conservées sur le serveur. Lorsque la page est demandée, elle est téléchargée depuis le serveur et livré au navigateur de l’utilisateur. [↑](#footnote-ref-15)