

Rapport de stage de Ludovic ANGENARD.

Entreprise et lieu du stage : Uneek à Nantes

Ecole : Campus Academy

Date du stage : 13 mai au 10 juillet

Sommaire :

1. Introduction
 - a. Général
 - b. Mes attentes
2. L'entreprise
 - a. Présentation générale
 - b. L'organisation interne
 - c. Sa marque de fabrique, son marché et sa concurrence.
 - d. La vie chez Uneek
3. Mon rôle
 - a. Le service intégré
 - b. Ma mission
 - c. Les enjeux de la mission
4. Préparation et organisation du projet
 - a. Les outils de travaux
 - b. Mode de travail
5. Déroulement du projet
 - a. Maquettage et catégorisation
 - b. Reproduction visuelle
 - c. Ajout du javascript
 - d. Stocker les icônes Fontawesome
 - e. Le coté serveur
 - i. Une application sur Ruby on rails
 - ii. Node modules
 - f. Résultat final du sélecteur
 - g. Documentation
 - h. Préparation du service
6. Conclusion
 - a. Identification des erreurs et solution pour la suite
 - b. Elévation des compétences
 - c. La réponse à mes attentes
7. Glossaire.

1 - Introduction

A - Général

Pour commencer, le stage est ma première expérience professionnelle dans le développement informatique. Cette expérience s'est déroulée du 13 Mai au 10 Juillet 2020. Cette expérience a plusieurs objectifs pour mon cursus professionnel.

Le premier objectif, étant d'ajouter une expérience professionnelle, non pas uniquement dans le développement mais dans n'importe quel domaine. Cela a pour but de développer les soft skills uniquement, faire évoluer l'apprenant de façon professionnelle, comprendre quelles sont les attentes d'une entreprise et adapter son comportement vis-à-vis de l'entreprise. Par exemple, être autonome ou obéir strictement à la demande de la hiérarchie.

Le second objectif, est de développer les hard skills en ayant une expérience professionnelle dans le domaine de notre spécialisation pour la deuxième année. En effet, si la possibilité s'offrait à moi de choisir un stage en tant que développeur, il fallait la saisir pour plusieurs raisons. La première, est de mettre en pratique les compétences apprises chez Campus Academy. La deuxième, est d'améliorer mes compétences déjà acquises. Enfin, la troisième est d'augmenter mon panel de compétences en apprenant de nouvelles technologies comme des langages, des bibliothèques mais surtout des outils comme des Framework. Tout cela, a pour but de nous faire progresser et de nous aiguiller quant à la spécialisation qu'il faut choisir pour la deuxième année.

B – Mes attentes

Tout d'abord, mon but était de trouver impérativement un stage en tant que développeur informatique plutôt qu'une autre expérience professionnelle dans un domaine quelconque. Car cela apporte bien plus de progresser dans le domaine d'étude pour l'année suivante qu'uniquement acquérir des compétences sur l'attitude à adopter en entreprise. Bien que ses softs skills soient important, j'ai déjà eu des expériences professionnelles et en acquérir une nouvelle dans un domaine quelconque apportera peu de valeur ajoutée dans mon Curriculum Vitae. En effet, ce stage de développement informatique sert aussi à apporter plus de crédibilité à mon Curriculum Vitae afin de trouver plus facilement une alternance pour ma deuxième et troisième année.

Ensuite, personnellement j'ai une préférence pour le back-end dans le développement informatique. Car cela permet d'avoir plus d'emprise dans la création de nouvelles fonctionnalités et cela est pour moi plus satisfaisant et plus stimulant une fois le code fonctionnel. De plus, le back-end permet la création de plus de fonctionnalité avec la gestion des bases de données et laisse donc une marge de création plus importante. Cependant, cette plus grande liberté implique des technologies plus complexes avec la gestion de serveurs ce qui peut être bloquant pour un stage de deux mois uniquement car le projet ne peut pas être assez conséquent pour y intégrer du back-end.

Enfin, le stage court me permet de créer et d'appliquer mes compétences sur de plus petits projets. Ce qui me fera travailler également voire principalement sur des technologies front-end. Le Front-end me pose quelques difficultés puisqu'il faut un regard sur l'esthétisme et faire attention à l'UX, UI. De plus, les technologies front-end vu en cours comme le langage javascript et sa bibliothèque jQuery ont été apprises en fin de première année, juste avant l'expérience professionnelle.

Le stage est donc pour moi une occasion de retravailler les technologies front-end, apprendre ou approfondir des technologies back-end, améliorer mes compétences dans l'expérience utilisateur et avoir un regard plus critique vis-à-vis de l'esthétisme du projet si possible.

2 – L’entreprise

A – Présentation générale

L’entreprise qui a accepté de me prendre en tant que stagiaire développeur informatique sur deux mois, est l’entreprise Uneek situé à 5 bd de la Louettrie à Nantes. J’ai travaillé avec l’équipe qui s’occupe de la marque Kosmopolead, situé également à Nantes mais qui a ses propres bureaux se trouvant à 168 route de saint joseph.

L’entreprise Uneek SARL est un éditeur de logiciel qui se focalise sur le développement d’applications web sous la marque Kosmopolead. Kosmopolead a donc une équipe à taille humaine et ce qui en découle c’est l’ambiance de groupe décontractée et une forte cohésion d’équipe. Aussi, l’équipe Kosmopolead est régie par une forte écoute et entraide de chacun. Ce qui favorise la rapidité quant à la réponse apportée lorsqu’il y’a un problème, que ça soit un client qui remonte une idée de nouvelles fonctionnalités ou un bug testé en interne sur un produit en cours de réalisation. L’entreprise prône une rigueur quant à l’approche du client que ce soit dans l’écoute de ses besoins ou dans le service apporté. Uneek fait au mieux pour que le client ait le service qu’il lui faut le plus rapidement possible. Ainsi, chaque service proposé pour le client est adapté à celui-ci.

Aussi, Uneek n’hésite pas à innover dans ses services pour anticiper la demande client et augmenter la qualité fournie par ses services afin que l’expérience utilisateur soit de plus en plus facilité au quotidien et ainsi faire gagner du temps aux clients.

Enfin, dans le service de développement, la productivité est importante comme dans toute entreprise. Cependant Uneek était une ancienne startup et devait produire très vite à ses débuts en délaissant certaines bonnes pratiques du développement. Ce qui ne leur convient désormais plus car le service de développement informatique chez Uneek prend en grande considération la propreté, la sécurité et le fait de tester la fiabilité du code afin d’éviter quelque souci technique lors de la mise en production.

B – L’organisation interne

Uneek est organisé en plusieurs services. Les services sont au nombre de trois, il y’a le service marketing, le service de développement informatique et le service de test d’expérience utilisateur. Je vais définir brièvement les rôles de chacun et plus m’attarder sur le service du développement informatique puisque c’est dans ce service que j’ai fait mes deux mois de stages.

Tout d’abord, lorsqu’il s’agit de s’occuper des rapports clients, c’est au service marketing que l’on fait appel. En effet, ce service a pour rôle de trouver de nouveaux clients mais surtout de les accompagner au mieux dans la recherche de leurs besoins afin de trouver le service qui leur correspond le plus. Ils sont aussi la passerelle entre le client et le service de développement, puisque dès qu’il y’a un soucis technique quelconque, c’est le service marketing qui identifie les manipulations faites par le client pour remonter le rapport au service informatique qui doit résoudre le souci.

Ensuite, lorsqu’une fonctionnalité est finie et avait passé tous les tests unitaires, Uneek fait appel à son service de test d’expérience utilisateur. Leur rôle est d’assurer que le produit soit ergonomique, intuitif à utiliser et que tous les services de la marque Kosmopolead soient régis sous les mêmes règles d’utilisation et que le client ne soit pas perdu s’il utilise deux produits de la marque Kosmopolead.

Enfin, le service de développement informatique qui est le service que j'ai intégré en tant que stagiaire, a pour rôle de proposer des idées, développer et maintenir les produits que propose Uneek. En effet, le service de développement ne fait pas que créer les produits pour les clients. Le service peut faire preuve de force de propositions puisque c'est le service qui sait ce qu'il peut être fait et ce que les nouvelles technologies vont permettre de faire. De plus, le service de développement doit prendre en compte les aléas de l'utilisation des clients sur leur produit et concevoir des solutions techniques afin que ses utilisations non prévues ne perturbent plus le bon fonctionnement du produit. Enfin, leur rôle est aussi de développer de nouvelles fonctionnalités pour améliorer des produits existants ou pour en créer de nouveaux.

L'organisation de l'entreprise dépend donc de ces trois services qui travaillent toujours en commun pour que tout le produit finit soit raccord avec la demande client et fonctionnel.

C – Marque de fabrique

Uneek a une marque déposée qui s'appelle Kosmopolead. Cette marque contient un ensemble de solutions qui sont conçus, développés et maintenus par eux-mêmes. Uneek est spécialisé dans la conception d'intranet pour les collectivités. Ils ont donc plusieurs produits qui sont adaptés à plusieurs marchés. Uneek propose le produit *Kosmopolead Entreprise* qui est un outil permettant une alternative aux mails et informe toutes les personnes travaillant dans l'entreprise via un historique de version et une messagerie instantanée. Aussi, *Kosmopolead Entreprise* permet le partage de tous types de documents à des contacts, des utilisateurs des groupes ou encore des adresses mails. *Kosmopolead Entreprise* est donc sécurisé en encryptant toutes les données puisque c'est un outil adressé aux entreprises et que les données doivent être confidentielles. De plus, Kosmopolead Entreprise a un outil de CRM pour que le client puisse interagir en continue avec ses propres clients. Le but de *Kosmopolead Entreprise* est de faire gagner du temps à ses clients en évitant les pertes de temps dans la recherche d'information dans les mails, en pouvant planifier des réunions et en améliorant la communication en interne avec les employés et en externe avec leurs clients.

La marque Kosmopolead, n'est pas seulement limité à un intranet pour les entreprises. Mais elle a aussi à son effectif, deux autres produits. Le premier s'appelle *Kosmopolead Campus & Alumni* et le second *Kosmopolead Mairie et collectivités*. Le premier, est un outil dédié aux Campus et Universités qui souhaitent retrouver leurs anciens diplômés, regrouper les informations des étudiants et de permettre aux associations étudiantes de trouver des entreprises pour financer leurs projets. Le second outil, est adressé aux collectivités souhaitant gérer et paramétrer leurs réseaux téléphoniques, dématérialiser leurs documents et les mettre en réseaux, rediriger les mails dans le service souhaité, avoir un portail des usagers et gérer ses réseaux sociaux.

Uneek conçoit donc des intranets de la marque Kosmopolead qui sont sophistiqués et intelligents en répondant systématiquement à une demande spécifique. Par exemple, un Campus souhaitant trouver ses anciens élèves ou une Entreprise souhaitant optimiser son temps pour récupérer ses informations. Uneek tant à perfectionner ses outils de recherches en étudiant le domaine de l'intelligence artificielle.

D – LA VIE CHEZ UNEEK.

Tout d'abord, Uneek permet à ses employés le télétravail. En effet, à cause de la crise du sars-cov2 que nous connaissons, nous ne pouvons pas être plus de deux à quatre personnes à travailler dans un même espace confiné pendant plusieurs heures. Aussi, nous avons l'obligation de porter le masque dans les bureaux et une borne de gel hydroalcoolique était à notre disposition à l'entrée du

bâtiment. L'entreprise dispose donc de trois bureaux. De plus, nous ne pouvions pas être plus de huit à Neuf personnes dans l'entreprise la même journée. Pour respecter cela, nous devions planifier nos jours d'entreprise et de télétravail grâce à l'agenda disponible dans les outils *Kosmopolead Entreprise*. Personnellement, je faisais au moins deux jours en entreprise par semaine puisque c'est ma première expérience professionnelle dans le développement informatique et par conséquent, la première fois que je travaillais chez Uneek. Le premier mois, je restais trois à quatre jours en entreprise pour pouvoir parler de la mission qui m'a été confiée, demander de l'aide si je bloquais trop longtemps et avoir quelques pistes pour la suite de ma mission. Puis le second mois, je ne restais plus que deux jours en entreprise puisque j'avais les pistes pour continuer seul.

Les horaires de travaux sont officiellement de neuf heures à douze heures et de quatorze heures à dix-huit heures. Les horaires ne sont pas fixes, on peut commencer avant et finir plus tôt et inversement si on n'est pas en retard dans nos plannings. Aussi, lors de la pause déjeuner, nous nous attendions pour manger afin de passer un moment convivial ensemble. Nous sommes tous séparés d'au moins un mètre afin de respecter les règles de distanciations sociales. Enfin, une fois notre journée terminée, nous devions désinfecter notre partie du bureau à l'aide d'un spray virucide puisque nous n'avons pas de bureaux attitrés et toute personne peut prendre le bureau d'une autre personne le lendemain selon les besoins.

Les horaires de travaux et l'attente de productivité en télétravail sont les mêmes qu'en entreprise. En effet, le fait de s'imposer le même rythme avec les mêmes horaires, me permet de pouvoir discuter avec mon maître de stage à distance. Si nous sommes chacun en télétravail et que personne n'a les mêmes horaires, alors nous ne pourrions pas échanger de façon productive et cela nous mettrait tous les deux en retard dans nos plannings respectifs.

Cependant, la vie professionnelle a quelques différences en télétravail. En effet, le fait de ne pas avoir à se déplacer en voiture en ville pour aller dans les locaux fait que l'on commence une journée plus calmement et sereinement. Même si c'est une différence qui m'est totalement personnelle et non liée à Uneek précisément, cela fait partie des changements de vie en télétravail que j'ai pu notifier. De plus, le fait d'être chez soi me permettait de réaliser des activités personnelles lors des pauses déjeuner. Cela permet d'avoir une réelle coupure entre les demi-journées de travail et donc de pouvoir continuer sans être fatigué.

Pour finir, le télétravail a ses avantages comme ses inconvénients. Il faut savoir s'imposer un rythme chez soi et ce n'est pas toujours évident. Aussi, le fait de passer une journée sans voir ses collègues est à prendre en compte. En effet, être en entreprise permet d'avoir un environnement chaleureux, convivial et propice à la productivité dû au fait d'être entouré de collègues travaillant en même temps et d'avoir accès à des écrans plus grands. Cependant, les journées en entreprise étaient pour moi plus fatigantes en fin de journées dû au fait qu'elles étaient plus stimulantes. C'est donc un choix qu'il faut faire en fonction de la tâche à accomplir et de son énergie personnelle.

Pour conclure sur cette partie, Uneek est une entreprise en pleine croissance où chaque employé a son importance puisque Uneek est organisé de façon à ce que chaque service soit complémentaire les uns envers les autres. Ainsi, l'entreprise propose des services complets, fait pour répondre à une problématique du client et ainsi transformer son activité en réussite. L'ambiance y est agréable à vivre puisque les valeurs de l'entreprise mettent en valeur ses employés tant humainement que professionnellement en développant proposant des produits qui demande à relever certains défis.

3 – Mon rôle

A – Le service de développement informatique

J’ai donc rejoint le service de développement informatique partiellement décrit dans la partie 2.B. Ce service est constitué de cinq chercheurs développeurs informatique auquel s’ajoute trois stagiaires en développement informatique dont moi. Le service informatique a donc beaucoup à faire, puisque comme dit dans la partie 2.B, le service est chargé de proposer, développer et maintenir les fonctionnalités et produits d’Uneek. La maintenance passe par la mise à jour des technologies avec lesquels sont conçus les produits Uneek. Par exemple, mettre à jour ruby 2.5 à la version supérieur 2.6, permet d’avoir accès aux nouvelles possibilités du langage ou d’éviter d’avoir des failles de sécurités qui sont maintenant connues. Ensuite, l’équipe fait preuve de force de propositions et cela peut prendre plusieurs formes. Cela peut aller d’une suggestion de fonctionnalités pour un client, à la proposition de certaines technologies afin d’apporter un réel plus pour les prochains produits d’Uneek. En effet, la veille informatique effectué par les employés permet de savoir quelles technologies sortent et surtout ce qu’elles permettent de faire. C’est très important pour Uneek afin d’anticiper l’innovation. Mon rôle dans ce service est de prendre en charge le développement d’une fonctionnalité qui sera réutilisable pour les produits Kosmopolead et ainsi pouvoir réduire la charge de travail de mon maître de stage.

B – Ma mission

Pour commencer, ma mission était de créer un sélecteur d’icône from scratch afin que l’utilisateur puisse consulter les icônes et en sélectionner une en limitant le nombre d’interactions que l’utilisateur doit effectuer. Aussi lors de ce stage, je devais créer un service permettant à l’utilisateur d’ajouter ses propres icônes. Enfin, il a fallu le rendre réutilisable par tous les développeurs d’une part en faisant en sorte que techniquement il puisse l’être et d’autre part en fournissant une documentation claire et précise afin de l’installer, de l’enrichir et de le personnaliser correctement. Créer un sélecteur d’icônes permet en interne de pouvoir chercher les icônes identifier les tables des bases de données plus facilement et ainsi faire gagner du temps aux employés. Aussi, le projet peut être réutilisé pour ajouter une plus-value à une fonctionnalité existant dans un produit de la marque Kosmopolead. Par exemple, permettre à l’utilisateur de pouvoir envoyer des icônes dans un chat de façon intuitive.

De ce fait, le sélecteur doit répondre à certains critères. Il faut que les icônes soient accessibles grâce à des tags. Ce qui permet à l’utilisateurs de pouvoir trouver l’icône qu’il souhaite grâce à des cheminements différents. De plus, il faut que les icônes soient catégorisées. C’est-à-dire qu’il doit y avoir des catégories et possiblement des sous-catégories distinctes les unes des autres avec des titres uniques où sont rangées les icônes. Ainsi, il faut que l’utilisateur puisse accéder et naviguer à travers ces ensembles de catégories. Enfin, il faut bien évidemment que l’utilisateur puisse voir les icônes et avoir un avant-goût de son icône choisit.

Un sélecteur d’icônes en trois parties se dessine donc. Une première partie réservée à la recherche, où l’utilisateur peut taper des mot clefs, ce qui affichera un ensemble d’icônes correspondant à ces mots. Une deuxième partie liée à la navigation, où l’utilisateur peut cliquer pour accéder directement aux icônes liées à la catégorie sélectionnée. Enfin la dernière partie de prévisualisation, où l’utilisateur peut regarder les icônes et cliquer sur celle qu’il souhaite afin de la sélectionner.

Le sélecteur d'icônes sera un module, ce qui permettra au service de développement informatique de pouvoir facilement l'intégrer dans différents projets. C'est pourquoi un README doit être rédigé en y explicitant, l'installation, le fonctionnement et son utilisation pour un développeur.

C – Les enjeux de ma mission

Tout d'abord, ma mission est double. En effet, en plus de devoir créer un sélecteur d'icône, je devais créer un service complet au tour de celui-ci, ce qui hausse les enjeux. La première partie est de créer un sélecteur d'icônes. C'est une fonctionnalité en plus qui peut être utilisée dans les produits Kosmopolead. Il faut donc que le client n'ait aucun problème technique lors de l'utilisation du sélecteur. Si cela arrivait, alors Uneek perdrait en crédibilité devant ses clients, ce qui n'est pas dans leurs objectifs.

Cependant, ce cas de figure est peu probable puisque le service de développement informatique et d'expérience utilisateur testent les fonctionnalités avant de les mettre en production. Cela est donc plus probable que les conséquences se limitent à une augmentation de la charge de travail pour mon maître de stage et l'équipe de développement informatique en général, ce qui causerait un retard dans le développement des produits Uneek.

De plus, ce retard peut-être accru si la documentation fournie avec le sélecteur d'icônes et le code de celui-ci ne sont pas clairs. En effet, la perte de temps occasionnée par une documentation peu précise et un code approximatif est gigantesque. Ainsi, il faudrait que mon maître de stage comprenne et décrypte mon code pour ensuite réparer ce qui n'allait pas.

Donc, si ma double mission est faite de façon incorrecte, cela peut engendrer des pertes de temps pharamineuses pour le service de développement. Néanmoins, le service d'ajout d'icônes lié au sélecteur d'icônes reste secondaire et n'est donc pas un objectif prioritaire pour Uneek. Si l'échec de la mission a des conséquences sur le service de développement informatique, alors la réussite de celle-ci en a tout autant.

Ainsi, le succès de la création du sélecteur d'icônes aura pour conséquences une baisse de charge de travail pour le service en plus d'avoir une plus-value pour les produits Kosmopolead ainsi qu'en interne, pour identifier les bases de données. Le service d'ajout d'icônes quant à lui reste une plus-value importante mais ne sera pas une baisse de charge de travail puisque cela reste un objectif secondaire.

4 – Préparation et organisation du projet

A – Les outils de travail

Pour commencer, le stage a nécessité des outils tant pour communiquer avec les autres employés que dans la construction même du sélecteur d'icônes.

Ainsi, afin de pouvoir communiquer que ça soit en télétravail ou dans les locaux avec mon maître de stage, nous utilisons l'intranet de l'entreprise qui est le produit développé par elle-même, *Kosmopolead Entreprise*. Cet intranet, nous permettait de pouvoir échanger à l'aide d'une messagerie instantanée. Mais également, de pouvoir échanger des documents, des liens ce qui permettait de stocker les informations et de pouvoir les réutiliser facilement si cela était nécessaire.

Ensuite, la partie graphique du sélecteur est créée grâce à du HTML et du CSS. Le HTML, sert donc à baliser les différents éléments du sélecteur afin de pouvoir les mettre en forme grâce à du CSS. Ces deux technologies sont prévues pour de l'édition de texte mais sont maintenant utilisées pour tous

les affichages web. Certains comportements ont posé problèmes car le sélecteur d'icônes est très éloigné d'une page web classique avec des éléments textuels.

Pour continuer, le CSS est attribué à une classe ou à un identifiant appelé couramment id. Or, le fait d'attribuer un style à un identifiant ou une classe sur ce projet est problématique. En effet, comme dit plus haut, le projet est censé être greffé à d'autres projets. Si j'utilisais un id ou une classe sur le sélecteur alors, l'élément peut récupérer le style pré-fait du projet auquel il est greffé, ou pire dans le cas de l'identifiant qui est censé être unique, où il peut empêcher le véritable identifiant de pouvoir afficher le bon style. Pour pallier ce problème Bootstrap est la solution idéale, puisque ce framework utilise du CSS en interne et seul des classes suffisent pour appeler ce CSS. Ces classes peuvent donc être utilisées dans le sélecteur puisque Bootstrap va les associer à des styles pré-fait. De plus, Bootstrap permet une uniformisation des styles puisque si le même outil est utilisé dans tous les sélecteurs d'icônes, alors ce sont les mêmes styles qui vont être affichés. Ainsi, l'esthétisme du sélecteur d'icônes est cohérent.

Aussi, l'HTML est généré automatiquement grâce à du javascript afin de ne pas avoir besoin de copier tout le HTML pour chaque icône. Le javascript est le langage web le plus adapté pour cette tâche puisqu'il ne sollicite que très peu le serveur avec des requêtes http. En effet, le langage s'exécute du côté client. De plus, le javascript est un langage asynchrone ce qui a pour caractéristique de pouvoir charger plusieurs éléments en même temps. Cela est très utile dans un projet comme la création de ce sélecteur, puisque les icônes vont pouvoir commencer à être chargées en même temps que le corps du sélecteur, ce qui diminue le temps de réaction de l'affichage lorsque l'utilisateur clique sur le bouton.

De plus, le fait que le javascript s'exécute du côté client impose certaines contreparties techniques. En effet, c'est le navigateur qui exécute le code javascript et c'est donc du navigateur que le javascript est dépendant. Les navigateurs ne supportent pas certaines fonctionnalités du javascript qui doit donc trouver une alternative pour chaque navigateur. En tant que développeur, on ne va pas construire plusieurs sélecteurs différents et afficher celui qui correspond au navigateur choisi par l'utilisateur. De ce fait, jQuery est indispensable pour l'aboutissement de la mission, car il va permettre d'uniformiser le code pour tous les navigateurs et en interne, il va détecter le navigateur et écrire le javascript de la bonne façon. Cela me permet de pouvoir créer un seul sélecteur d'icône, disponible sur tous les navigateurs.

Pour finir avec les outils qui conçoivent visuellement le sélecteur, Fontawesome est une police de caractère qui va être utile pour le sélecteur puisqu'il affiche des icônes. Les icônes sont en fait des caractères unicode que Fontawesome relie aux visuels. Leur utilisation se fait grâce à des classes spécifiques. En effet, la classe est lue par le navigateur qui cherche une correspondance dans les feuilles de styles du sélecteur. La classe est trouvée dans une feuille de styles de Fontawesome qui applique l'index de la table unicode correspondante, ce qui va conduire à l'icône. Enfin, l'icône est affichée grâce à l'astuce du `:before` en CSS qui permet de passer l'icône, de la classe au contenu même de la balise.

Les outils permettant la construction visuelle du sélecteur ne sont pas suffisants pour qu'il soit fonctionnel. En effet, il faut que le sélecteur soit réutilisable facilement par les développeurs de l'entreprise. Pour cela, il faut réaliser un module grâce à Nodejs qui va mettre le code du sélecteur sur un serveur et de ce fait, il va pouvoir être appelé par n'importe quel développeur.

Aussi, il faut que le Node module soit disponible pour les autres développeurs. Ainsi, on utilise NPM qui est un gestionnaire de bibliothèques Javascript. Je l'utilise pour pouvoir utiliser Yarn qui est une couche supérieure à NPM et permet d'avoir un fichier appelé `yarn.lock` qui répertorie notamment les

dépendances liées au sélecteur d'icônes et les met à jour pendant le développement. Donc, c'est grâce à NPM et Yarn que nous rendons le Node module disponible aux autres développeurs à condition d'avoir le lien du registre Yarn et d'avoir l'accès.

Enfin, je devais tester l'application qui en l'occurrence représentait uniquement mon sélecteur d'icône grâce à Ruby on rails. Ruby on rails est un framework du langage Ruby exécuté côté serveur contrairement au Javascript. Le framework Ruby on rails quant à lui permet de créer des applications grâce à une arborescence réglée. Lors de la création du sélecteur ces technologies m'ont servi pour tester celui-ci en récupérant le Node modules et en utilisant les gems des dépendances grâce à Yarn.

En ce qui concerne le système d'exploitation, je travaillais sur le WSL2 qui est donc le Windows subsystem for linux 2 soit grossièrement, ubuntu intégré dans windows 10. J'ai choisi cette solution car j'utilisais mon ordinateur personnel qui avait donc windows 10 et qui était dépourvue de double boot et de machine virtuelle. Je pouvais utiliser les commandes linux sur un windows 10.

Pour terminer sur les outils, l'IDE que j'utilisais s'appelle Visual Studio code et m'était pratique pour développer avec la correction syntaxique qu'il propose mais surtout la couleur dans le code afin de mieux repérer les différents éléments. Cela est un réel avantage pour développer plus rapidement contrairement à Nano que j'utilisais dès que je développais directement sur le WSL2. De plus, Visual Studio Code permet une gestion des fichiers que l'on peut pousser ou tirer très rapidement dans le GitLab qui est la forge logicielle utilisée par Uneek.

Pour conclure, les outils pour réaliser cette mission sont nombreux mais ont différent but. En effet, certains permettent la construction directe du sélecteur tant dans la création visuelle que dans l'élaboration algorithmique qui permet la réalisation des fonctionnalités. Quand d'autres mettent en communication le code entre les développeurs ou permettent la rédaction du code avec un gain de rapidité de développement.

B – Mode de travail

Les outils de construction du sélecteur ont été présentés, mais ça ne permet pas au sélecteur d'être construit à temps. En effet, le stage était de deux mois, et pour respecter les délais, il faut avoir une méthodologie.

Pour commencer, Uneek a ses propres méthodologies. En effet, au sein du service de développement informatique, des réunions spontanées sont mises en place quand il y'a un sujet qui mérite le point de vue de toutes les personnes du service. Dès qu'un souci ou une information concerne plusieurs personnes, alors un des employés qui a l'information ou qui a découvert le souci, informe les autres. Ainsi, les informations circulent rapidement en interne.

Lors de mon stage, je travaillais donc avec mon maître de stage. Ma mission n'engageait personne d'autre pendant le développement, alors la diffusion des informations se faisaient par oral ou grâce à la messagerie instantanée. De ce fait, mon tuteur m'accompagnait en me donnant des pistes seulement grâce à des liens internet ou en expliquant oralement ce à quoi servait telle ou telle technologie. Cela avait pour but de me guider sans donner les réponses pour que j'apprenne. Internet était ma seule documentation lors de ce stage. Cela m'a beaucoup fait progresser dans la recherche d'informations dans les documentations officielles comme dans la recherche d'exemples concrets sur git hub. Le fait de s'appuyer entièrement sur internet, permet également de ne pas être perdu lorsque l'on utilise une technologie qui n'a pas été encore utilisée par l'entreprise ou du moins qui l'a été partiellement. Cela a été le cas pour la construction du sélecteur lors de l'utilisation d'un élément de Bootstrap.

Cette méthodologie de travail m'a donc permis d'améliorer mon autonomie et mon efficacité dans la recherche. En conséquence, je devais retenir les informations soit en écrivant une liste de tâches à accomplir dans la journée. Dès que j'avais une question sur un procédé que je n'arrivais pas à saisir alors, j'éliminais certaines possibilités grâce à de simples tests afin que mon maître de stage ne perde pas trop de temps pour chercher la cause du souci. Cependant, lorsque la tâche à accomplir était plus compliquée, l'application de cette méthodologie prenait beaucoup de temps. En effet, lorsqu'il fallait apprendre une technologie plus complexe notamment en réseau, il y'a énormément de montée en compétences en amont. C'est dans ce cas de figure que devenait nécessaire de passer quelque temps avec mon maître de stage pour que la mission puisse être effectuée à temps. Aussi, à chaque tâche inscrite sur le planning terminée, j'effectuais des tests afin de savoir si le sélecteur d'icônes était toujours fonctionnel. Les tests effectués ne sont pas des tests unitaires où l'on écrit du code qui compare la solution attendue avec la solution reçue. C'étaient des tests visuels à effectuer où j'effectuais une liste de manipulations que j'avais conservée sur un bloc note. Donc, à chaque ajout de fonctionnalités ou d'éléments affichés sur le sélecteur d'icônes, j'effectuais cette liste de manipulations.

Enfin, afin d'avoir un suivi de la progression du développement du sélecteur d'icônes, la mise en place d'un planning s'avérait primordiale. En effet, un planning est un moyen parfait pour d'une part, permettre aux supérieurs de savoir où le développement en est et d'autre part de pouvoir anticiper les tâches et savoir les estimer. Cela aide donc à savoir quelles technologies il faudra utiliser et si la durée du stage prévue de deux mois est suffisante. La mise en place d'un planning nécessite qu'il soit tenu et mis à jour à chaque fin de tâche et si une tâche se rajoute, il faut le notifier, sinon le planning perd de son intérêt.

5 – Déroulement du projet

A – Maquettage et catégorisation

Le maquettage est une partie essentielle au projet puisque c'est dès cette partie que l'on définit les possibilités et les différentes fonctionnalités du projet même si dans le cas du sélecteur d'icône, ce n'est pas très dérangeant de changer les plans en cours de route au vu de l'envergure du projet. En effet, il n'est pas si dérangeant de faire des correctifs de conceptualisation du sélecteur ou d'ajout de fonctionnalité en cours de développement puisque ça ne peut être que des petits correctifs. Il faut cependant être claire dès le départ pour ne pas avoir une quantité de correctifs conséquente, ce qui pourrait retarder la concrétisation du projet.

Pour la réalisation de la maquette, je devais d'abord faire une catégorisation claire qui me permettait de savoir le nombre de catégories. La catégorisation est longue, car il faut prendre en compte toutes les icônes qui sont très nombreuses. Cela a pris plusieurs jours de trier les icônes et de choisir le nom des catégories. Cette liste de catégories est amenée à évoluer au cours du développement afin d'être affinée. Ainsi, une icône peut être dans plusieurs catégories afin que des utilisateurs différents puissent retrouver leur icône avec leur propre cheminement. Par exemple, l'icône « ambulance » peut se trouver dans la catégorie véhicule comme dans la catégorie santé. Si trop d'icônes font doublon dans deux mêmes catégories alors on essaye de trouver un autre nom à la catégorie pour fusionner les deux anciennes catégories. S'ajoute à cela le fait qu'il y'ait des sous-catégories qui contiennent parfois elles-mêmes des catégories.

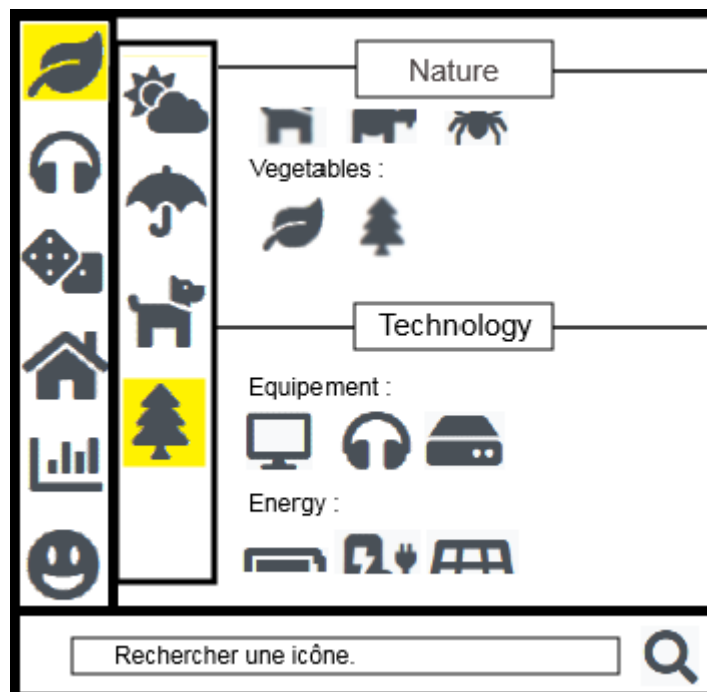
Un schéma est plus parlant pour illustrer la conceptualisation des catégories.

Schéma de l'arborescence des catégories.

- **Catégorie principale 1**
 - Catégorie secondaire 1
 - Icône 1
 - Icône 2
 - Catégorie secondaire 2
 - Catégorie tertiaire 1
 - Icône 1
 - Icône 2
 - Catégorie tertiaire 2
 - Icône 1
 - Icône 2
- **Catégorie principale 2**
 - Catégorie secondaire 1
 - Icône 1
 - Icône 2
 - Catégorie secondaire 2
 - Icône 1
 - Icône 2

La principale problématique que cela engendre, c'est de créer le sélecteur en rendant la lecture des catégories simple et intuitive. Pour cela, la maquette permet de savoir sur quelles bases on part. La première maquette proposée dispose d'une barre de navigation à chaque niveau de catégories.

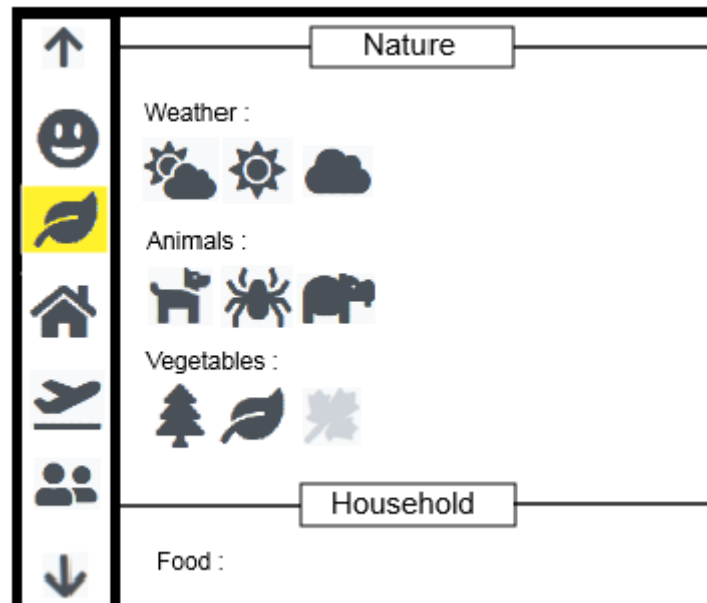
Première maquette du sélecteur d'icônes.



Ici, les icônes représentent une catégorie. Car l'utilisateur repère mieux une image plutôt que du texte. Seulement, il y'a trop d'icônes avec les deux barres de navigations, ce qui ne plaisait pas à mon maître de stage. Alors, j'ai enlevé la barre de navigations pour n'en laisser que la principale. Aussi, j'ai rajouté des flèches pour indiquer à l'utilisateur qu'il y'a d'autre catégories de disponibles. Les titres

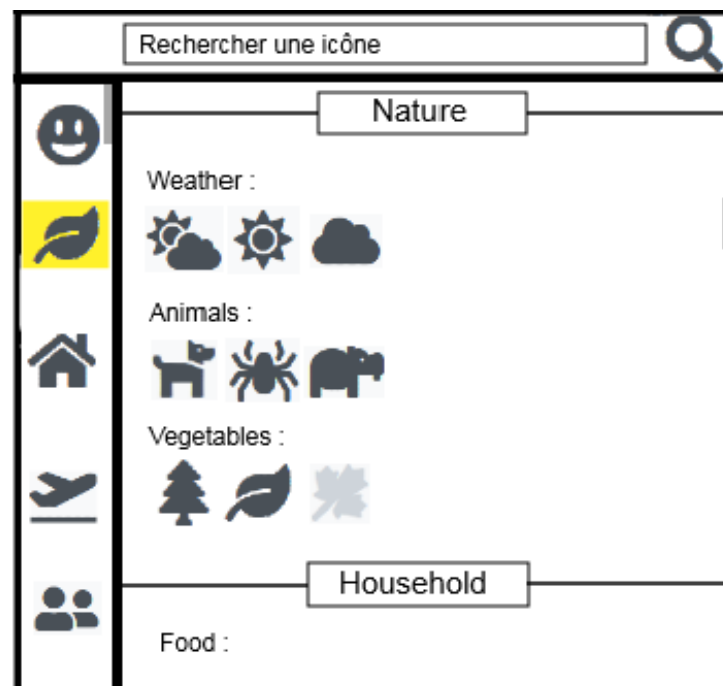
des catégories de différents niveaux sont toujours indiqués et marqués par une différence de style, dans la partie de visualisation des icônes qui est la partie principale du sélecteur. L'objectif de cette maquette est de rendre le sélecteur le plus épuré possible. Cela a donné ce résultat :

Deuxième maquette du sélecteur d'icônes.



La maquette était claire mais manquait d'élément. Il fallait rajouter la fonctionnalité de recherche mais la mettre plus en évidence, au-dessus de la partie principale. De plus, les flèches étaient à remplacer par un ascenseur, ce qui est plus parlant pour l'utilisateur. Ce qui amène à une troisième proposition de maquette.

Proposition finale de la maquette pour le sélecteur d'icônes.

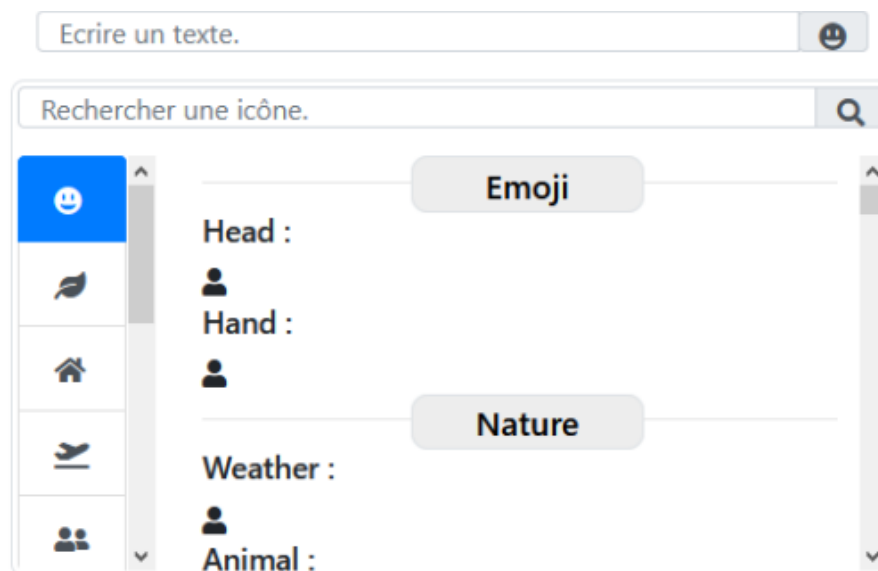


La maquette est donc épurée avec une barre de navigation simple d'utilisation et une fonctionnalité de recherches. Cependant, la maquette ne représente pas le produit final mais n'est qu'une ébauche de ce qu'il pourrait être. Cela signifie que la maquette est amenée à évoluer au cours du développement si l'on se rend compte que l'affichage d'un élément ne convient plus. De même, la catégorisation est amenée à évoluer au cours du temps ou au moins on va y revenir plus tard pour les ajouts de tags.

B – Reproduction visuelle

Une fois que la maquette a été terminée, elle a été réalisée visuellement grâce à du HTML et CSS en premier lieu. Le but était d'organiser les éléments au bon endroit et d'appliquer un style. En effet, même si une maquette était déjà conçue, elle ne présentait uniquement l'organisation des différents éléments et aiguillait sur le style que l'on voulait appliquer au sélecteur d'icônes. Ainsi, il fallait réaliser différents tests graphiques avant d'obtenir une version qui plaisait.

Sélecteur d'icônes créé en HTML et CSS



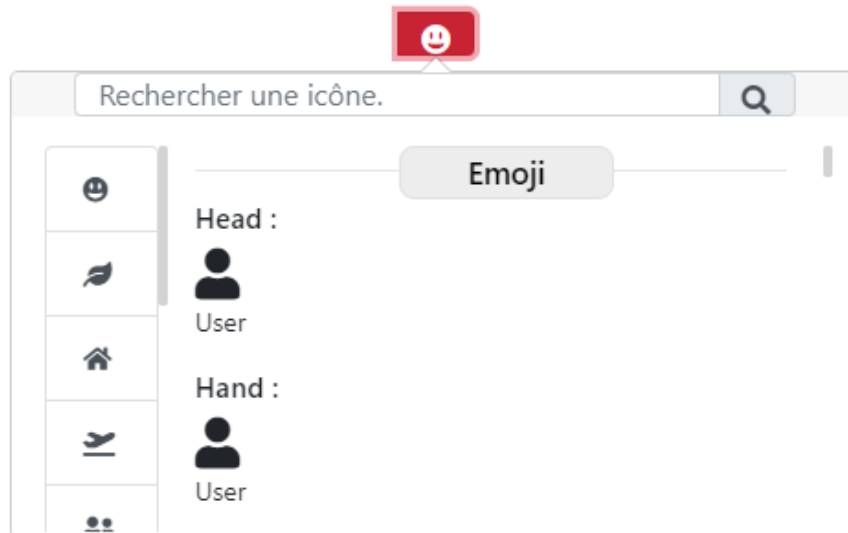
Cependant, il fallait ensuite remplacer tout le CSS par des classes Bootstrap pour toutes les raisons décrites dans la partie 4.A. Pour la création même du widget, la fonctionnalité popover de Bootstrap qui se base sur la bibliothèque Popper nous paraissait le plus adapté à ce que nous voulions faire contrairement au modal qui nous aurait bloqué sur le système de position qui est plus souple que le popover. Plusieurs problèmes suite à ce choix s'ensuivent. En effet, c'était la première fois que j'utilisais cette fonctionnalité et de même pour mon maître de stage qui d'habitude utilisait les modals. Cela nous posait des problèmes pour que les éléments s'emboîtent parfaitement et ne se chevauchent pas. L'alignement vertical est parfois problématique sur Bootstrap. En revanche, ce n'est pas un problème des Flexbox qui au contraire nous permettait d'aligner verticalement les éléments du sélecteur d'icône et Bootstrap propose des classes qui permettent d'utiliser le modèle Flexbox.

Enfin, une feuille de styles est conservée pour changer l'aspect des ascenseurs au moins sur le navigateur Google Chrome. Cela n'était pas une bonne solution de garder une feuille de styles puisqu'il fallait supprimer toutes les classes ne venant pas de Bootstrap mais cela permettait un

rendu plus léger au sélecteur et aussi cela permettait de supprimer l'écrasement d'un ascenseur contre la barre de navigation.

Ainsi, grâce à la fonctionnalité popover du framework Bootstrap et du modèle Flexbox, le sélecteur d'icônes a un squelette qui est créé mais qui pouvait encore légèrement évoluer au cours du développement pour qu'il soit parfait.

Sélecteur d'icône créé à partir de Bootstrap et avec le style des ascenseurs.



Le squelette du sélecteur est créé, maintenant il faut y appliquer toutes les icônes mais cela est beaucoup trop long à intégrer en HTML car il y'a presque un millier d'icônes différentes. Il faudrait donc plusieurs milliers de lignes HTML ce qui n'est pas raisonnable. C'est pourquoi nous avons opté pour l'utilisation d'un langage et c'est donc sur du javascript que j'ai travaillé.

C – Ajout du Javascript

Tout d'abord, j'utilisais le langage javascript avec la bibliothèque jQuery qui permet d'avoir un même code pour tous les navigateurs. En effet, javascript est exécuté sur le navigateur et étant donné la diversité des navigateurs aujourd'hui, le langage en lui-même ne peut pas être compatible avec tous. C'est donc la bibliothèque jQuery qui s'occupe de cette compatibilité.

Aussi, tout comme le CSS, il ne devait pas y avoir de HTML pour le sélecteur. En effet, il ne devait rester en HTML que le bouton qui permet d'afficher le sélecteur une fois appuyé. Ainsi, il a fallu créer un écouteur en javascript. Cela permet de vérifier si le bouton est bien appuyé et dès qu'il est appuyé, il exécute le code.

Ensuite, j'ai rendu fonctionnel les différentes possibilités qu'offre le sélecteur d'icônes pour un utilisateur. Ainsi, l'utilisation du bouton, la recherche d'icônes grâce aux noms de celles-ci, l'utilisation des icônes et l'utilisation de la barre de navigation ont été créés assez vite mais ce n'était absolument pas optimisé et le code n'était pas propre. Le but était de voir les différents bugs qu'il pouvait y avoir sans pour autant gérer l'optimisation et les possibles soucis techniques que cela allait amener en plus. Cependant, c'était une erreur de ne pas avoir optimisé un minimum le code. Notamment la partie de l'affichage des icônes dans les bonnes catégories. Car cela m'a valu une journée pour nettoyer cette partie-là. Aussi, je n'avais pas pris en compte la recherche des icônes via des tags car cela est venu plus tard dans le développement.

Puis, le sélecteur a été nettoyé une première fois pour que l'ajout des icônes dans les catégories se fasse proprement. De plus, j'ai pris l'initiative de créer une classe ce qui rend le code lisible et propre. Mais cela a des conséquences notamment dans l'utilisation de l'objet. En effet, il faut que le sélecteur d'icônes puisse être utilisé plusieurs fois sur la même page. C'est-à-dire que l'utilisateur peut choisir sur un sélecteur une icône et sur un autre sélecteur une autre icône sans pour autant que les deux sélecteurs transmettent leur icône à l'autre. Ainsi, une instance de la classe permet de créer un sélecteur d'icône qui est différent d'une autre instance de la même classe. Donc le sélecteur d'icônes peut être présent plusieurs fois sur la même page.

Néanmoins, l'utilisation d'une classe en javascript peut avoir des conséquences qui nuisent au bon fonctionnement de celle-ci. En effet, le système d'exploitation windows 7 propose comme navigateur par défaut internet explorer. Internet explorer est un vieux navigateur maintenant abandonné au profit de microsoft edge. De plus, javascript qui est exécuté grâce aux navigateurs nécessite des normes pour qu'il y ait un standard. ECMAScript est un ensemble de normes et ECMAScript 6 n'est pas supporté par internet explorer alors que c'est ECMAScript 6 qui apporte l'utilisation des classes à javascript. Ainsi, les utilisateurs de window 7 qui sont encore nombreux puisque c'est un bon système d'exploitation, ne pourront pas utiliser le sélecteur d'icônes via internet explorer mais via un autre navigateur qu'ils devront choisir.

Enfin, lors de la création de la classe, l'utilisation du this était compromise à cause du javascript qui a un comportement spécifique lorsqu'il sélectionne un élément HTML. En effet, javascript considère l'élément sélectionné comme un objet et ainsi l'utilisation du this à l'intérieur de cette sélection pointe vers l'élément et non vers l'instance en elle-même. Afin d'y remédier, il a fallu créer une variable dès que le sélecteur est instancié dans le constructor. Ainsi, c'est cette variable qui représente l'instance et non this.

En conclusion, le Javascript est un langage très utile pour la gestion d'éléments HTML et sur son automatisation et sa capacité à être asynchrone ce qui est optimal pour du web. Cependant certains comportements atypiques de javascript sont à connaître pour ne pas subir des bugs lors du développement.

D – Stocker les données Fontawesome

Tout d'abord, les icônes sont stockées dans un fichier YAML. En effet, le format YAML est un format qui représente des tableaux, des objets ou autre grâce à une liste et une syntaxe très simple. Cela était utile pour l'étape de la catégorisation qui était fastidieuse. La syntaxe simplifiée était beaucoup de temps gagné sur le long terme. En effet, le sélecteur contient plus de neuf-cents icônes et écrire uniquement le nom de l'icône dans une liste pour qu'elle soit répertoriée dans un tableau représente un gain de temps conséquent. De plus, le format YAML a été mis à jour lors de l'utilisation des tags. Ainsi, avec plus de trois tags à chaque icône, le moindre caractère en moins à écrire est un gain de temps conséquent puisqu'il y avait trois à quatre tags par icônes selon leur nombre de catégories. Le fichier YAML avait triplé uniquement à cause de l'ajout des tags.

Ensuite, le fichier YAML devait être converti en fichier Json pour que Fontawesome puisse le lire et ainsi afficher les icônes dans le sélecteur. En effet, le fichier YAML seul ne pouvait pas être utilisé. Il fallait le convertir en Json. Néanmoins, il était judicieux de le convertir automatiquement via du code javascript et non à la main via un convertisseur de fichiers sur internet. En effet, cela n'est pas une bonne pratique car en premier lieu, l'automatisation de la conversion fait gagner du temps non négligeable pour le développement. En second lieu, le fichier était donc téléchargé dans le convertisseur pour le ressortir en Json. Ainsi, le site qui converti a potentiellement une copie du

fichier YAML que j'ai transmis. Cependant, les seules données que j'ai transmises sont disponible gratuitement sur Fontawesome donc ce n'est pas dérangeant qu'un inconnu puisse avoir le fichier.

Ainsi, stocker les icônes Fontawesome grâce à un fichier YAML est pratique pour gagner du temps grâce à sa syntaxe simplifiée au maximum. Le format YAML ne se suffit pas en lui-même, il faut le convertir en Json afin de pouvoir utiliser les icônes de Fontawesome. L'automatisation de la conversion est donc très utile pour gagner du temps de développement en éliminant des procédures. Mais également, cela empêche d'utiliser un convertisseur de fichier et d'envoyer ses propres données au site en question.

E – Le côté serveur

Pour commencer, la construction même du sélecteur d'icône ne nécessitait pas de serveur, mise à part pour la police de caractère de Fontawesome. En revanche, il y'en avait besoin pour la distribution du sélecteur d'icônes afin que tous les développeurs puissent l'utiliser. En effet, la décision de créer un module était prise mais il fallait pouvoir le tester dans une application et c'est pourquoi le choix s'est tourné vers le langage Ruby et son framework Ruby on rails qui sont des outils déjà utilisés par Uneek et c'est donc plus facile pour mon maître de stage d'avoir un suivit.

I – Une application sur Ruby on rails

Tout d'abord Ruby on rails est un framework qui permet de créer des applications facilement en suivant l'arborescence précise. Pour installer ruby on rails il est plus facile de le faire à l'aide des commandes linux. Or, mon ordinateur personnel n'a que Windows 10 en système d'exploitation. Heureusement, Windows 10 dispose d'une partie linux appelée WSL2 pour Windows Subsystem for Linux 2. Pour utiliser les commandes Linux il suffit d'installer un client ubuntu depuis le Windows store.

Ensuite, grâce à la documentation officielles de Ruy on rails, j'ai obtenu la liste des commandes pour installer Ruby dans un premier temps et rails dans un second ainsi que toutes les dépendances nécessaires pour que cela fonctionne. Pour ce faire il faut utiliser RVM (Ruby version Manager). Dès que les dépendances ont été installés, j'ai commencé à créer l'application. Au début, il suffit d'écrire la commande ***rails new iconSelector*** et un dossier iconSelector est créé. Dès que cela est fait, cela est devenu plus compliqué pour moi. L'aide de mon maître de stage était donc plus que bienvenu à ce moment. En effet l'une des particularités de Ruby est de créer des gems qui sont donc des modules qui apportent des fonctionnalités. Les gems sont créés par d'autre développeur et on peut les réutiliser. On peut également les rassembler dans un Gemset qui est simplement un regroupement de ces gems. Pour que notre application fonctionne il fallait créer un Gemset qui regroupe les gems nécessaires. Il fallait également que dès que l'on ouvre le dossier iconSelector, que le Gemset s'active automatiquement. Cela demande des capacités plus avancées dans le langage Ruby que je n'avais pas le temps d'apprendre seul pour finir le projet du sélecteur d'icônes à temps. Dès que ce Gemset était créé et son activation automatique, nous avions accès au serveur rails. Grâce à la commande ***rails s*** un serveur est accessible via un Localhost avec le port que l'on souhaite sachant que le port 3000 est le port par défaut.

Dès que cela était installé on pouvait vérifier que la connexion au serveur rails fonctionnait grâce à un localhost. Le localhost était donc mon moyen de tester si l'application fonctionnait correctement. L'environnement était préparé, je suis passé à l'étape suivante qui est la création d'un Node modules

II – Node modules

Tout d'abord, un Node modules est une bibliothèque en Javascript sur NodeJs qui va exécuter le code. Cela permet de créer des bibliothèques Javascript et ainsi appeler des modules via NPM (Node Package Manager) ou Yarn. J'utilisais Yarn car c'était la technologie utilisée dans l'entreprise.

Cependant, mon maître de stage était moins disposé à m'aider sur cette étape puisqu'il n'avait pas créé de node modules. Ainsi, je devais avoir recours à la documentation sur internet qui explique comment faire un node modules et comment l'appeler. C'est sur cette étape que la création de la classe m'a beaucoup aidé puisque je devais exporter le code et NodeJs permet d'exporter une classe uniquement ce qui était plus simple.

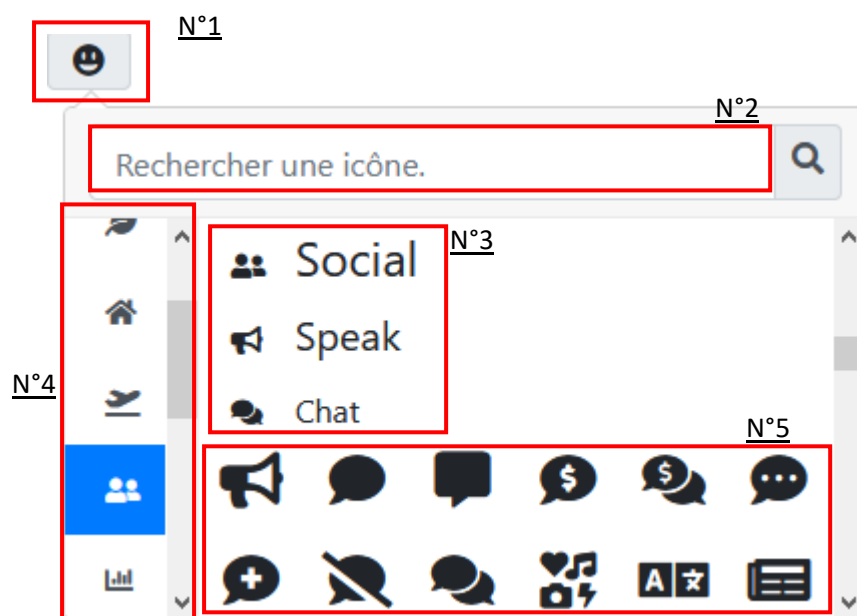
Ainsi, la création du node modules iconSelector s'est faite d'une part grâce à NodeJs qui permet d'exporter le module mais aussi grâce à Ruby on Rails qui permet de le tester en réseau. Cela a posé des problèmes lors des tests car cela était beaucoup plus long que si j'avais simplement une page web à recharger. En effet, à chaque changement dans mon module, je devais effacer le cache de Ruby on Rails, réactiver le serveur et attendre que cela se reconnecte. A partir de cette étape, le développement a commencé à ralentir puisqu'il y'avait moins de temps de développement et dès qu'il y'avait des corrections à apporter ou des modifications pour rendre le code propre, cela devenait tout de suite plus long à manipuler.

Enfin, une fois que le module était propre et bien exporté, il fallait réussir à le récupérer. Cela se faisait grâce à un registre Yarn et de le récupérer via Verdaccio pour que le GitLab du projet prenne en compte chaque nouveau commit. Le sélecteur d'icônes est donc prêt à l'emploi.

F –Résultat final

Le sélecteur d'icônes est fin prêt à être utilisé et est divisé en plusieurs parties afin de répondre à la demande client.

Voici une capture d'écran de la dernière version du sélecteur où l'on peut distinguer ses différentes parties :



N°1 : Bouton permettant d'ouvrir et de fermer le sélecteur. L'icône dans le bouton, change en fonction de l'icône choisie par l'utilisateur

N°2 : C'est ici que l'utilisateur peut rechercher le mot clef pour trouver son icône plus facilement.

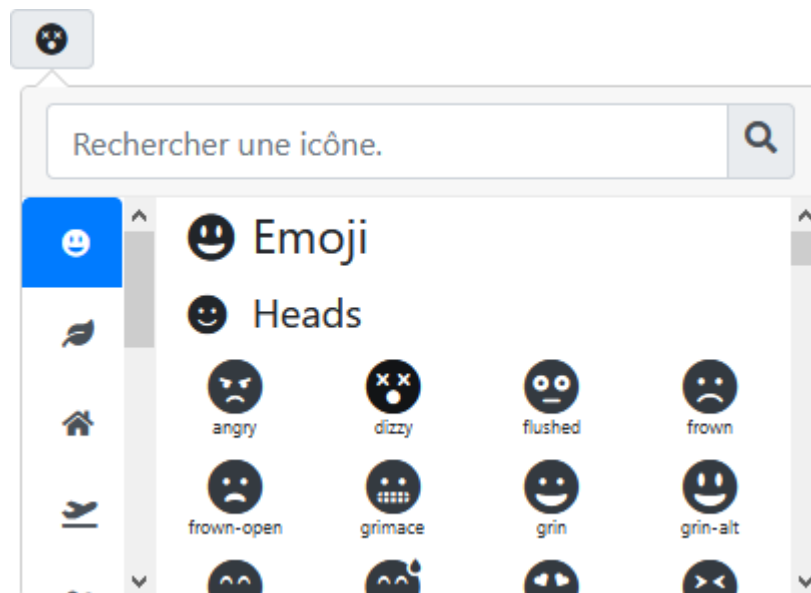
N°3 : Ce sont les titres de catégories. Il y'a trois niveaux de catégories. Leur niveau est indiqué grâce à leur taille. Ainsi le titre ayant la plus grande taille correspond à la catégorie qui contient la deuxième, qui contient elle-même la dernière.

N°4 : La barre de navigation qui contient des icônes, correspondant chacune à une catégorie. La catégorie sélectionnée est marquée par un surlignage bleuté.

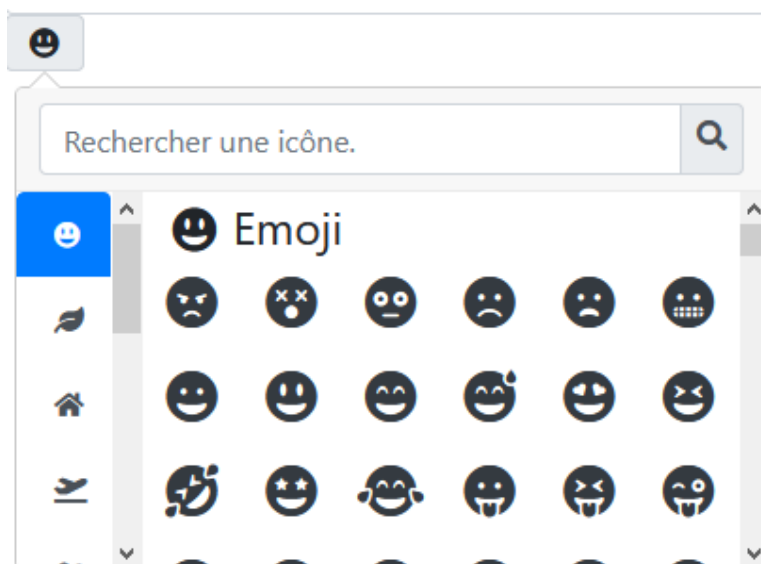
N°5 : La zone de prévisualisation. L'utilisateur peut scroller et cliquer sur une icône afin de la sélectionner.

Enfin, le sélecteur d'icônes dispose de quatre options accessibles au développeur, permettant une personnalisation et une souplesse d'utilisation selon le rôle du sélecteur.

La première, permet au développeur de choisir s'il affiche ou non les noms des icônes sous celles-ci.



La deuxième, permet au développeur de choisir s'il affiche ou non les sous catégories des icônes.



La troisième, permet au développeur de choisir si un clic sur une icône enlève l’affichage du sélecteur.

Enfin la dernière, qui est similaire à la précédente, permet au développeur de choisir si un clic en dehors du sélecteur enlève l’affichage du sélecteur.

G - Documentation

Pour qu’un projet soit utile à l’ensemble des membres et futurs membres de l’équipe de développement informatique, il faut que ce projet soit facile à installer. Pour ce faire, j’ai écrit une documentation dans un Readme sur Gitlab.

Dans un premier temps, le Readme explique ce qu’est le sélecteur d’icônes en y décrivant les fonctionnalités et sur quelles technologies il fonctionne. Cela passe par l’explication de l’installation en donnant les lignes de commandes à exécuter et l’HTML du bouton qui déclenche l’apparition du sélecteur d’icônes. S’ensuit une description des fonctionnalités mais surtout des différents modes que l’on peut appliquer au sélecteur comme le mode avec les noms d’icônes. Cela est très important d’indiquer tout cela aux futurs développeurs car là est tout le but d’avoir créé un Node modules. Il faut que tout le monde puisse utiliser le Node modules facilement et ainsi faire gagner du temps à tout le monde.

Dans un second temps, le Readme a aussi un mode expérimental sans serveur. Comme dit dans la partie 5.E. II les tests sont très contraignants à effectuer si l’on veut changer une partie du code juste pour de l’affichage Html par exemple. Pour cela un mode sans serveur a été créé pour que chaque développeur teste à partir de leur navigateur en actualisant simplement la page. Néanmoins, un mode sans serveur apporte quelques contraintes car je ne pouvais pas séparer le fichier contenant la liste des icônes et le fichier contenant la classe du sélecteur car le Javascript seul ne le permet pas. J’ai donc tout mis dans un même fichier ce qui n’est pas forcément très lisible mais qui est fonctionnel. De plus, je devais trouver une substitution pour l’Ajax. En effet, l’Ajax fonctionne grâce à des requêtes http ce qui nécessite un serveur. Pour ce faire, je ne pouvais pas rendre fonctionnel une option qui est celle qui permet la possibilité au développeur de changer de liste d’icônes.

Ainsi, le Readme permet d’avoir une documentation claire sur le GitLab du projet, ce qui réduit le temps de travail des développeurs lors de l’utilisation ou de la modification du sélecteur d’icônes.

H – Préparation du service

La mission consistait au départ à créer un sélecteur d’icônes et d’y associer un service permettant à l’utilisateur d’ajouter une icône. Cependant, avec un stage de deux mois, je n’ai pu que créer le sélecteur d’icône mais j’ai dû rechercher comment effectuer le service. La recherche était conséquente puisque c’est une partie plus complexe à effectuer. Pour ajouter une icône et lui attribuer un index de la table unicode il faut utiliser FontForge qui utilise la gem FontCustom. Ainsi, je devais indiquer comment installer Fontforge et Fontcustom et aussi expliquer les premières manipulations permettant d’initialiser le projet. Le but n’est pas d’expliquer dans la documentation comment faire pour que le service soit effectué complètement rien qu’en lisant la documentation mais plutôt de faire gagner du temps de recherches à la personne qui va s’occuper du service.

Par manque de temps, je n'ai pas pu élaborer une documentation claire du service mais cela est un départ qui permet de connaître les technologies dont ils auront besoin dans un premier temps.

6 – Conclusion

A – Identification des erreurs et solution pour la suite

Lors de ce stage, j'ai commis plusieurs erreurs qui m'ont coûté du temps de développement. Je pouvais être éparpillé sur différentes tâches et mal organisé. En effet, la création du sélecteur d'icônes demandait la réalisation de plusieurs tâches et celles-ci pouvaient évoluer au fur et à mesure du développement. Cela se produisait essentiellement sur l'aspect graphique du sélecteur mais cela pouvait se produire sur des comportements de celui-ci. C'est pourquoi est apparu les différents modes du sélecteur avec lesquels le développeur peut choisir s'il affiche les noms de sous-catégories où que le sélecteur s'efface dès que l'utilisateur clic hors de celui-ci. Aussi, la mauvaise organisation s'est retrouvée lors des tests visuels. Ainsi, j'aurais dû créer une liste des manipulations à effectuer dès le départ et ne pas attendre l'apparition de certaines erreurs qui ont fait perdre beaucoup de temps.

Enfin, développer sous WSL2 pouvait être une bonne idée sur le papier mais cela a causé une perte de temps conséquente pour moi et mon maître de stage. En effet, mon IDE Visual Studio Code était sur le windows 10 et le projet iconSelector avec le gemset de Ruby on rails était sous le WSL2. Cela posait un problème dès que j'ouvrais le projet iconSelector via VSCode et que j'ajoutais ou modifiais des fichiers. Cela avait pour conséquence une perturbation des droits des fichiers et cela perturbait l'exécution des fichiers. Ainsi, à chaque fichier de l'application de test Ruby on rails modifiés via VS Code, il fallait mettre les droits utilisateurs correctement. Puis, après avoir diagnostiqué le problème, j'ai effectué les modifications de l'application test sur Ruby on Rails grâce à l'IDE Nano qui est directement intégré dans le WSL2. Si c'était à refaire, j'aurais téléchargé une machine virtuelle comme Virtual Box avec une image Ubuntu ou Debian ou bien mettre mon ordinateur en double boot mais cela est plus complexe à réaliser.

En plus de me faire progresser grâce à la pratique et me faire découvrir des technologies et le monde de l'entreprise. Ce stage me permet aussi de remettre en question mes méthodes de travaux en me montrant mes erreurs et en m'aiguillant sur les solutions à apporter. C'est donc avec des solutions que je ressors de ce stage.

B – Elévation des compétences

Tout d'abord, lors de ce stage j'ai pu me perfectionner sur des outils. En effet, grâce au sélecteur d'icônes, j'ai appris d'autres fonctionnalités du framework Bootstrap comme popover mais surtout j'ai appris à aller plus loin que juste prendre quelques bouts de code par ci par là dans ce Framework. J'ai pris conscience de l'utilité de ce framework pour uniformiser les classes et donc les styles sur un même projet. Sur le langage Javascript et la bibliothèque JQuery j'ai appris énormément également. La notion d'écouteur en Javascript était quelque chose que je ne connaissais pas avant ce stage par exemple. Avant ce stage j'avais également quelques difficultés à développer avec ce langage asynchrone. Mais j'ai vite appris à utiliser des déclencheurs pour éviter certains comportements surprenant du langage. Des automatismes se sont créés à force de développer sur ce même langage pendant deux mois.

Ensuite, lors de ce stage j'ai appris à utiliser de nouveaux outils. En effet, j'ai découvert le langage Ruby et plus particulièrement son framework Ruby on rails avec le gestionnaire de bibliothèque Yarn ou encore NodeJs. Le langage Ruby est exécuté côté serveur et ainsi permet le partage de morceaux de codes. Ces morceaux de codes sont appelés Gem et j'en ai utilisé pour créer un Gemset. Je n'ai pas particulièrement écrit d'algorithme en Ruby, cependant j'ai pu en utiliser et voir la syntaxe et son utilité. Ruby on rails est donc le nouveau framework où j'ai beaucoup travaillé dessus lors de la seconde partie de mon stage. J'ai vu son fonctionnement, comment l'utiliser et en quoi ce framework est utile. De la même façon j'ai pratiqué sur NodeJs que je ne connaissais pas mais comme il fonctionne grâce au langage Javascript, je ne partais pas de zéro. J'ai donc créé un Node modules.

Enfin, j'ai mis en pratique des compétences que j'ai pu apprendre de l'école. En effet, outre le fait d'avoir travaillé sur des outils que j'ai déjà vu en cours, je devais adopter des bonnes pratiques apprises en cours. Notamment, la création d'une classe pour rendre le code plus propre et lisible pour les autres développeurs. Mais aussi, pour la maquette du sélecteur d'icône car les catégories sont identifiées par des icônes. C'est donc de l'iconographie, qui consiste à représenter ce qui peut l'être par des icônes afin de stimuler la mémoire visuelle de l'utilisateur. L'iconographie est surtout utilisée pour améliorer l'interface utilisateur. J'ai pu mettre en pratique des Soft-skills également. En effet, pour ce stage j'ai eu besoin de persévérer et d'être autonome. Je pense avoir développé mon autonomie au cours de ce stage puisque j'ai su où chercher et j'ai appris l'importance de regarder et analyser des exemples sur git hub notamment. Pour finir, j'ai travaillé pour être plus méthodique dans mon développement pour gagner du temps puisque c'est primordial en entreprise.

C – La réponse à mes attentes

Tout d'abord, ce stage court m'a permis de pratiquer essentiellement sur des technologies Front-end et en moindre partie sur du Back-end. Je préférais le Back-end avant ce stage mais depuis, je n'ai plus d'appétences plus importantes pour l'un que pour l'autre car les deux me plaisent autant. J'ai donc progressé sur les technologies Front-end comme javascript où j'avais du mal avec sa particularité d'être asynchrone mais depuis je m'y suis habitué et j'ai su m'adapter.

Ensuite j'ai appris d'autre technologie. En effet, j'ai eu l'occasion de travailler sur Ruby on rails notamment chose que je n'aurais pas pu faire dans toutes les entreprises. Je suis content d'avoir pu apprendre et voir autant d'autre technologies que je n'avais pas vu dans mon école car cela s'ajoute dans mon panel de compétences et cela permet aussi de pouvoir utiliser ces technologies pour d'autres projets.

Ensuite, j'ai pu travailler sur mon regard esthétique. En effet, tout le long du stage le visuel du sélecteur d'icônes n'a cessé d'évoluer pour atteindre un visuel plus épuré et sobre. Aussi, l'expérience utilisateur du sélecteur était à prendre en compte et cela m'a forcé à aller dans le sens de l'utilisateur et à créer un visuel permettant à l'utilisateur de prendre en main rapidement le sélecteur.

Pour conclure, cette première expérience professionnelle m'a beaucoup apporté aussi bien techniquement que pour ma culture personnelle. Puisque mon maître de stage m'a donné des informations ou des anecdotes sur des technologies qui sont bonnes à prendre et permettre de comprendre pourquoi elles sont créées ainsi. J'ai passé un stage avec une bonne équipe accueillante et qui n'hésitait pas à m'aider s'il le fallait et cela joue énormément pour qu'un stage se déroule dans la meilleure des façons et je tenais à les remercier pour ça.

7 – Glossaire

A.P.I. qui est le sigle de Application Programming Interface, est un ensemble de protocoles utilisés par un logiciel permettant d'interagir avec d'autre logiciel pour accéder à ses fonctionnalités.

Back-end ou arrière-plan en français, est un terme qui indique le côté serveur de l'application.

Bibliothèque est un ensemble de morceaux de codes prêt à l'emploi afin de pouvoir les utiliser sans devoir les réécrire. Le but est d'éviter de créer des répétitions de codes.

Bootstrap est un framework créé par Twitter permettant de créer des styles CSS parfois animé avec du javascript.

Constructor ou constructeur en français, est une méthode dans une classe qui exécute son contenu dès l'instanciation de sa classe.

CRM qui est le sigle de Customer Relationship Management, est un système qui permet de relier ses clients en permanence avec l'entreprise afin d'améliorer leur rentabilité.

CSS qui est le sigle de **Cascading Style Sheets**, est un langage de style qui permet de modifier l'apparence des pages HTML.

Double boot est le fait d'avoir deux systèmes d'exploitation partitionnés sur le même disque dur.

ECMAScript est un ensemble de normes pour le langage Javascript. Les navigateurs ont donc ce standard comme référence.

Fontawesome est une police de caractère permettant d'afficher des icônes.

FontCustom est un logiciel permettant de construire ses propres icônes.

FontForge est un logiciel permettant d'éditer ses propres polices.

Framework est un ensemble de composants créant la structure générique d'un logiciel.

From scratch veut dire de zéro en français, et signifie le fait de commencer un projet sans aucune base.

Front-end ou frontal en français, désigne la partie de l'application du côté client.

GitLab est une forge logicielle permettant aux développeurs de pouvoir développer en groupe sur un même projet.

Hard-skill ou compétences dures en français, est un terme qui désigne les compétences d'une personne sur de la technique contrairement aux compétences appelés **soft-skills** qui se basent sur l'attitude, la méthodologie et la compréhension de l'environnement d'une personne.

HTML qui est le sigle de hyper text markup language, est un langage de balisage et permet de séparer des parties de textes dans un site internet.

Id est l'abréviation d'identifiant et c'est la syntaxe dans l'HTML pour attirer un identifiant à un élément.

IDE est le sigle de integrated development environment, et c'est un logiciel pour développeur qui permet l'édition de texte principalement mais ils ont plus de fonctionnalités au cours du temps comme relier le code à la forge logicielle du projet.

Javascript est un langage asynchrone qui s'exécute du côté client et qui est normalisé par ECMAScript.

Machine virtuelle est un logiciel permettant d'émuler un poste de travail grâce à une image.

Nano est un éditeur de texte qui s'utilise depuis le terminal.

NodeJs est une technologie qui permet d'exécuter du javascript côté serveur. Il s'apparente donc comme un langage à part.

NPM est un gestionnaire de bibliothèques javascript.

Popper est une Bibliothèque de Bootstrap permettant de créer des popover.

Rails est un framework du langage ruby permettant de créer des applications rapidement.

README ou lis moi en français, est le nom que l'on donne à un fichier contenant la documentation essentielle à un projet.

Ruby est un langage de programmation qui s'exécute côté serveur. Soft-skill

Tags ou étiquette en français, permet de ranger des éléments selon un mot. Ainsi, on peut catégoriser ses éléments.

This est un mot clef utilisé dans la programmation objet comme préfixe pour désigner le fait que l'objet suivant (une variable comme une méthode par exemple) appartient à la classe où il se trouve.

UI est un sigle de user interface, désigne donc l'interface utilisateur et est donc la partie avec lequel l'utilisateur peut directement interagir comme dans un site internet par exemple.

UX est le sigle de user experience, est le quotidien de l'utilisateur de son utilisation d'un produit.

Veille informatique est une pratique qui consiste à s'informer des nouvelles technologies, des failles trouvées sur des outils ou d'autres nouvelles concernant l'informatique.

WSL2 est le sigle de Windows subsystem for linux 2, est un client linux dans windows, ce qui permet de pouvoir installer l'application ubuntu sur windows à partir du Windows store par exemple.

Yarn qui est un gestionnaire de bibliothèques javascript se basant sur NPM et développé par Facebook.