

# RAPPORT DE STAGE



VERDIER Quentin

22/02/2023

EPSI PSN2



## TABLES DES MATIERES

REMERCIEMENTS .....	4
INTRODUCTION .....	5
CONTEXT .....	6
Présentation de l'entreprise .....	6
Mission attribuée .....	7
ÉTUDE DE L'ENVIRONNEMENT ET DES TECHNOLOGIES .....	8
Étude de l'environnement.....	8
Étude des technologies.....	8
RÉALISATION TECHNIQUE ET LES MESURES .....	10
Présentation des logiciels, normes et principes.....	10
Les mesures.....	11
GESTION DU PROJET .....	12
DÉROULER DU STAGE.....	14
CONCLUSION .....	21
ANNEXES .....	22

## REMERCIEMENTS

Je souhait tout d'abord remercier mon tuteur de stage Arnaud LIMOUZIN pour m'avoir permis de faire ce stage et de travailler sur ce projet. Je le remercie pour le temps qu'il a pris pour répondre à mes questions. Je souhaite le remercier pour la confiance qu'il m'a donnée pour ce projet qui m'ont permis d'enrichir mes compétences en développement frontend et backend, malgré le fait que je connaissais très peu le langage JAVA SCRIPT. Je remercie aussi les personnes qui était présente dans l'open-space avec qui j'ai pu échangé et beaucoup apprendre et pour leur bienveillance.

## INTRODUCTION

Du 9 janvier au 10 février, j'ai effectué un stage au sein de l'entreprise Infotel Conseil (situé au 11 Imp. Serge Reggiani, 44800 Saint-Herblain). Au cours de ce stage que j'ai pu m'intéresser au développement front-end et back-end d'une application de référencement des clients et confrère de Infotel.

Plus largement, ce stage a été une opportunité pour moi de m'immerger en tant que développeur front-end et back-end dans une entreprise nationale. Au-delà d'enrichir mes connaissances en développement, ce stage m'a permis de comprendre dans quelle mesure le développement était l'une des voies dans l'informatique dans laquelle je pourrais me complaire dans mon futur professionnel. Si ce secteur m'attirait, notamment par les enjeux grandissant de l'informatique dans la société, ces cinq semaines à travailler sur ce projet m'ont confortée dans mon projet professionnel.

Ainsi, je commencerai par contextualiser le stage dans le cas de ce stage, en vous présentant notamment l'entreprise dans laquelle j'ai opéré, puis les missions qui m'ont été attribué. Par la suite, je vous présenterais l'étude de l'environnement et des technologies, qui fut d'avantage un travail de recherche et de veille afin de trouver les informations utiles à mes différentes missions. Ensuite, je vous exposerai la réalisation technique pour mes missions, composée notamment de la présentation des logiciels, normes et principes utilisés, et des mesures. Enfin, sera évoqué le stage en tant que tel, c'est-à-dire les différentes missions réalisées. Après avoir présenté la gestion du projet sera détaillé son déroulé durant les cinq semaines. Y seront exposés les objectifs à réaliser, les contraintes, les difficultés auxquelles j'ai fait face, les réponses apportées à ces défis, ainsi que la manière dont j'ai réalisé les différentes missions.

## CONTEXT

Dans cette rubrique vous sera présenté l'entreprise du stage. Il sera évoqué tout d'abord la présentation de Infotel Conseil et son organisation. Après cela vous sera exposer les Missions qui m'ont était attribué.

### Présentation de l'entreprise

Infotel est une Entreprise de services numériques (ESN) ainsi qu'un éditeur de logiciels possédant 730 clients dans des domaines variés : bancaire, financières, du secteur public parmi d'autres. Nous pouvons citer la Banque Populaire, Airbus ou bien la Poste. Ce groupe a été fondé en 1979 par des anciens employés d'Airbus et Capgemini.

J'ai intégré l'équipe LAB Nantes. Elle est composée de Lead Techniques qui travaillent un jour par semaine au sein des locaux parmi leurs autres engagements. Des Product Owners sont également présents. Les autres membres de l'équipe sont composés de stagiaires, d'alternants, ou bien des salariés intercontrat, qui ont du temps dédié sur leur planning pour travailler sur les projets du Laboratoire technique. Les projets gérés par cette équipe sont internes, c'est-à-dire que leurs portées se limitent au sein d'Infotel Conseil. Les projets maintenus sont les suivants :

- AR2C : Application de Référencement des clients et des confrères.
- ARHI : Application dédiée aux Ressources Humaines d'Infotel pour le processus d'embauche de nouveaux candidats.
- GDR : Définition des risques liés aux projets d'Infotel.
- Mon agence : Application de gestion d'agence.
- SQIL : Suivi de l'évolution des compétences des utilisateurs.

## Mission attribuée

Pour commencer mon stage ma première mission a été de me renseigner sur les méthodes et principes qui sont utilisé dans l'entreprise ainsi que de me renseigner sur le projets qui est de créer une application permettant de référencer les clients et leurs relations avec les confrère de infotel.

Ensuite j'ai commencé ma mise en veille technique pour trouver une solution d'intégration une carte dynamique a un site en React.

Après m'être informée sur les technologies et méthodes, j'ai donc pu commencer ma missions qui était d'intégrer une carte permettant de visualiser les clients d'Infotel sur la pages d'accueil ainsi qu'ajouté un moyen de position les clients a leurs création ou leurs modifications.

Plus tard j'ai finaliser la partie back-end des relations entre les clients et les confrères qui n'était pas encore créé.

# ÉTUDE DE L'ENVIRONNEMENT ET DES TECHNOLOGIES

A présent, penchons-nous sur l'étude de l'environnement, en d'autre-terme, la veille pour les librairies permettant d'intégrer une carte dynamique sur un site en Reactjs. Ensuite, nous verrons l'étude des technologies possibles dont celles qui existent déjà pour n'en retenir que les plus pertinentes.

Pour rappel, les librairies sont des ensembles de fonctions et classe déjà codé et conçu pour être réimplémenter dans d'autre code pour faciliter la création de service ou fonction souvent utiliser.

## Étude de l'environnement

Avant de commencer le projet je me suis donc renseigner dessus. J'ai appris qui allais permettre de simplifier le référencement des clients de infotel ainsi que les relations qu'ils ont avec les confrère d'infotel. Ce qui facilitera le travail des personnes chargé de recrutement pour la mise a jours de ses informations mais aussi la visualisation de c'est information comme l'ajout de la carte qui permettra une vision géographique. Et ce qui veux aussi dire que le projet sera utiliser par un grand nombre de personnes et pourra être en charge d'un grand nombre d'information.

## Étude des technologies

Tout d'abord intéressons-nous aux projet dans son état actuel. Il est réparti en 2 sous projet. Le front-end qui est notre application React qui va permettre au utilisateur d'avoir un site web comme interface. Ce front-end utilise comme librairies principales *Matériel UI* et *MobX*. *Materiel UI* est une librairies qui permet de proposer des design simple mais efficaces tout en étant responsive c'est-à-dire qu'il s'adapte a la taille de l'écran. *MobX* est une librairies qui va nous permettre de créer des « stores » qui sont des composants nous permettant de stocké des données qui vont être utilise a de nombreux endroit dans le site. Nous les utilisons donc pour toutes les informations que nous récupéreront de la base de données grâce à l'API. Ensuite nous avons le back-end



qui est une API, c'est un service qui permet la communication entre différents services ou logiciel qui dans notre cas nous permet la communication entre notre site et notre base de données. Les bibliothèques principales sont *Spring*, *Hibernate* et *Mokito*. *Spring* est une bibliothèques qui intègrent beaucoup de fonctionnalités pour simplifier la création des classes et des configurations pour le démarrage de l'API. *Hibernate* est une bibliothèques qui nous permet de créer la connexion à la base de données et de gérer la transformation entre les données stockées dans la base de données et les objets qui sont utilisés en Java qui permet de les utiliser et modifier dans l'API. Enfin *Mokito* qui permet de créer des « mokes », qui sont des attributs façades que l'on utilise pour faire les tests et renvoyer au code les données qui nous intéressent dans notre test.

Ensuite j'ai donc effectué ma veille pour les différents outils que je pourrais utiliser. Pour comment, il y avait OpenStreetMap qui est une carte ouverte et collaborative du Monde. Ce qui nous permet d'avoir une carte mise à jours régulièrement par de nombreux utilisateurs et ce gratuitement et libre de service. Ce qui en fait le candidat idéal. Ensuite il fallait donc une technologie permettant d'intégrer OpenStreetMap au site React. Pour cela il y avait OpenLayers qui est une bibliothèques javascript open sources mais qui n'est pas encore adaptée pour le react de manière simple et efficace. J'ai aussi vu google map qui proposait sa propre bibliothèques react mais qui était limitée dans la quantité d'utilisation ce qui limite fortement le développement du projet. Et enfin j'ai découvert Leaflet qui est une bibliothèques Java-Script légère et libre de service et qui possède une bibliothèques dédiées à l'adaptabilité React sans être limitée.

## RÉALISATION TECHNIQUE ET LES MESURES

Ici l'IDE, environnement de développement intégré, choisie, les extensions, les logiciels, les normes et les principes vont vous être présentés et expliqués afin d'offrir un aperçu précis des conditions dans lesquelles les missions ont été réalisées, avant d'évoquer les mesures du projet.

### Présentation des logiciels, normes et principes

Pour faire ce projet, j'ai utilisé 2 environnement de développement *IntelliJ* et *Visual Studio Code*. L'ide *IntelliJ*, éditée par *JetBrains*, est un IDE dédié au développement de logiciel sous Java, qui est un langage dédié à la création d'application embarqué, ou ses déclinaisons. Il est très utile car il dispose d'une assistance intelligente de codage qui vérifie le code pendant la saisie et analyse l'intégralité du projet. Il a aussi une refactorisation rapide et sécurisée et offre une interface visuelle très utile pour gérer le projet avec *Git*. Mais a l'inconvénient d'être très gourmand en ressource. Et l'ide *Visual Studio Code*, éditée par Microsoft, est un IDE polyvalent. Il est très léger et permet de travailler sur plusieurs langage en simultanée et possède une communauté active qui permet d'ajouter des extensions pour améliorer la productivité. C'est pour quoi comme mon travail s'orienté majoritairement vers le react et que le débogueur est donc dans le navigateur chrome. J'ai préféré garder les 2. *IntelliJ* m'a été utile pour travailler sur l'API et utiliser GIT de manière simplifiée et *Visual Studio Code* m'est utile pour travailler sur le site et faire de légères modifications sur l'API.

Ce tableau ci-dessous montre comment ce choix a été orienté, et pourquoi je me suis organisé ainsi pour ce projet :

	IntelliJ	Visual Studio Code
Accessibilité	Gratuit	Gratuit
Spécialité	Développement JAVA	Un peu de tout
Débogage	5/5	0/5
Examen du code	5/5	5/5
Développement de logiciel	5/5	0/5

Facilité d'utilisation de Git	5/5	4/5
Prise en main	4/5	2/5

tableau de comparaison entre IntelliJ et Visual Studio Code

*Git* est un logiciel de gestion de version. Il permet de travailler de façon collaborative sur un seul projet. J'ai donc dû apprendre comment utiliser les branches correctement pour avoir un travail organisé et cohérent. Chez Infotel les branches sont organisé par une branche = une fonctionnalité. Le nom de la branche doit être explicite par rapport à la tâche effectuée, afin de faciliter les recherches dans l'historique des modifications.

Le projet utilisais une base de donnée *MariDB*, avec des valeurs fictive mais qui permettait de tester l'application.

Lors de ma veille j'ai appris les principes qui étaient utilisés dans l'entreprise, telle que le principe de développement du Test-Driven Development. Cela consiste à commencer la réflexion d'abord par les tests qui seront effectuer dans le code. Cela implique de prévoir tous les cas possibles mais a l'avantage de simplifier l'algorithmie et réduire les oublies. Ainsi que la Sémantique des versions avec un référentiel distinct. Par exemple. Les APIs ont des versions 1.0.2, elles sont toutes composer de 3 nombres. Le premier nombre représente les mises à jour majeurs. Le second les mises à jour mineurs. Et le dernier, les corrections de bugs.

## Les mesures

Durant ce projet nous avons pu faire plusieurs mesures grâce à des tests local à l'aide de la base de données fictives et de docker pour héberger l'API en local. Un exemple de message envoyé est présentée à l'annexe n°1. Mais la librairie n'a été incorporé dans les APIs seulement après mon départ alors je n'ai pas pu voir le service d'alerte teams fonctionner mais j'ai eu des retours grâce à l'équipe CRM pour me dire que tout se passait bien.

## GESTION DU PROJET

Dans cette partie sur la gestion de projet vous sera expliquée l'organisation des missions. En effet, je présenterai ici la réalisation des différentes et leurs planifications.

Etant donné que mon tuteur de stage était généralement cher des clients nous nous voyons que une journée par semaine. Je devais donc travailler en autonomie mais j'étais toujours accompagné des autre personnes dans l'open-space à qui je pouvais demandé des conseils.

Pour commencer, J'ai pris connaissance du projet grâce à la documentation dans Confluence qui est un outils permettant le partage de connaissances.

J'ai donc pour comprendre comme le projets était architecturé avec les différentes couches. Pour le back-end, l'API comporte 3 couches. La premiers couche est appelé « Repository », cette couche est destiné à faire le liens entre l'Api et la base de données. La seconde couches est nommé « Service », dans cette couche l'Api est libre de modifier et transformer les données. C'est aussi a cette couche que l'on effectue les vérifications des données pour les envoies en uniquement si elle corresponde à se qu'on attend sinon l'on l'envoie un message d'erreur et l'en stop le processus avant que cela ne créer des problèmes. Et enfin la dernière couche qui est appeler « Controller », c'est la couche qui permet de configurer les appels de l'Api et de vérifie les authentication d'accès. Et pour le front-end les couches sont aussi au nombre de 3. Elles sont appelé « Service », « Store » et « Composant ». La couche « Service » sert à effectuer tous les appelés vers l'Api. La couche « Store », elle nous permet de stocké les réponses de Api et de les réutiliser dans plusieurs composants. La couche « Composants » est la couche ou l'on organise tous nos composants react c'est tous les morceau ou partie du site qui vont s'assemblé et créer notre site web.

J'ai pu voir les maquettes avec les idées qui avais était trouvé pour réaliser le site web.

Et il y avait aussi un backlog qui est une boîte à idées avec des nouvelles idées à implémenter au site ou des fonctionnalités qui pourraient être intéressantes à ajouter pour le confort des utilisateurs.

## DÉROULER DU STAGE

A présent, dans cette partie, vous pourrez suivre en détail le déroulement des missions qui ont été accomplies durant ces cinq semaines. De l'objectif à réaliser jusqu'aux moyens utilisés pour livrer mon projet, en passant par les différentes contraintes et les difficultés auxquelles j'ai fait face, et les solutions utilisées pour y remédier.

Pour comprendre la mission qui m'a été affectée, il était important que je m'imprègne du contexte et de l'organisation de l'entreprise et de l'équipe. J'ai donc commencer par me renseigner en lisant les documentations du projets et de l'entreprise pour en apprendre plus sur le projet et sur l'organisation de l'entreprise. J'ai donc appris que le projets allais servir au chargé de recrutement pour référencé les clients et les relations avec les confrère de infotel. Ce qui leurs permettra d'avoir une vision plus claire des niveaux de relations entre les confrère et les clients, ce qui facilitera le relai des demande d'offre des clients et de pouvoir les relayer plus efficacement a leurs confrère si il n'ont pas encore eu l'information et de l'envoyé uniquement au confrère qui peu réponde au attente de l'offre et qui ont les compétences. Ensuite, j'ai lu la documentation interne pour comprendre quelles étaient les technologie et méthode de développements internes. Ce qui est toutes les normes telle que le versionning qui est une convention de numérotage des version d'un projet en utilisant 3 nombre sous la forme x.x.x , le premier nombre est la version majeur, l'on l'augmente lors de changement majeurs sur l'application telle qu'un ajout d'une fonctionnalité. Le second nombre est pour la version mineur, elle corresponds au mise à jours. Et enfin le dernier nombre corresponds au patch c'est les corrections de bugs ou les légère modifications.

Après avoir pris connaissance du projet et des normes j'ai donc commencer a installer mon environnement de développement. J'ai donc du téléchargé mes IDEs et les logiciels et outils qui me serais nécessaire pour tests le projets. Après avoir fait mes installations je me suis donc tourné vers une personnes de l'open-space pour qu'il m'aide a récupérer le projet et a faire mes demande d'accès. Comme je n'avais pas

encore accès au Git, il a dû m'envoyer le projet par ZIP plutôt que d'attendre les droits. J'ai donc pu commencé a regardé l'avancement du projets.

Dans la journée de nous avions remarqué que le PC avais des problèmes étrange avec les souris qui ne fonctionnais pas il n'y avait pas les effets de survoltage sur les boutons et les clics à de nombreuses reprise ne fessais aucun effet. Et malheureusement le lendemain matin, j'ai eu la mauvaise surprise de me rendre compte que le poste qui m'avais été attribué ne voulais plus démarré. Et demandais le code bitlocker qui est enregistré sur le PC après avoir été saisie une premières fois à l'initialisation du poste. Mais même en demandant au service de maintenance le code le pc essayer de corrigeais le problème et redémarrais mais nous redemandais le code bitlocker nous faisais la même boucle. Nous avons donc changé mon postes et j'ai réinstaller mon environnements de développements et je commençais a remarqué que la souris commençais a avoir les même problèmes que l'autre pc nous avons donc tester avec une autre souris et ceci semblais fonctionnais correctement. Nous pensons donc je c'était le pilote de la souris qui pouvait être à l'origine de se problèmes. Plus tard je pu donc commencer a tests le projets et le faire tourné sur mon poste mais des données étais nécessaires pour accédé au site telle que des codes d'accès pour passer la pages de connections. Je me suis donc dit que pour m'occuper en attendant mon tuteur je pouvais faire un petit projet react à coté et tests des librairies pour intégrer une carte interactive sur un site en react. J'ai donc fait ma veille et trouvé 3 librairies OpenLayers, Google maps et Leaflet React. OpenLayers est une librairies java script open sources mais qui n'est pas encore adapter pour le react de manière simple et efficace. Google Maps proposais sa propre librairies react mais elle limité dans la quantité d'utilisation pour la version gratuite, ce qui limite fortement le développement du projet ou nécessitait des dépenses supplémentaire, qui je pense n'étais vraiment pas envisagé. Et enfin j'ai découvert Leaflet qui est une librairies Java-Script légère et libre de service et qui possède un librairies dédiés à l'adaptabilité React sans être limité. J'ai donc commencer mon projets test avec la librairies leaflet. Mais je ne comprenais pas pourquoi les images de la cartes s'affichait de manière désorganiser. J'ai donc pris du temps a comprendre mais la documentations de leaflet react était un complément de la documentations JS pour la simplifier en react. J'ai donc trouvé les éléments qui me manquais sur internet.

J'ai donc fait un mini projet où j'affichait la carte et permettait d'afficher ou non les régions et les départements de France et de mettre un petit pop-up lorsque l'on clique dessus ou j'affichait le nom. Car sur les maquettes qui avaient été faites la navigation entre les clients sur la carte était par rapport au département ou régions sélectionner que nous affichons une liste sur le côté.

Après avoir fini mon projet test il était arrivé le jour de travail en interne de mon tuteur je pu donc lui demander ou récupérer les données nécessaires pour tester et faire fonctionner le site. Et nous avons aussi pu configurer mon git pour lequel la demande d'accès avait eu une réponse. Je pu donc tester le site et me rendre compte de l'avancement du projet et des modifications que j'allais devoir apporter. Mon tuteur m'a ainsi parlé de plutôt enregistrer la position des clients et de les afficher afin de les marquer sur la carte. J'ai donc réfléchi à l'enregistrement des localisations des clients et proposer d'ajouter des champs à la table client, le premier pour la latitude et le deuxième pour la longitude. Après un petit échange nous validons l'idée à je commençais donc à modifier la base de données pour ajouter les deux champs. Et ensuite je modifiai l'API pour ajouter les deux nouveaux champs à la classe de l'API pour pouvoir les utiliser dans l'API et faire le lien entre ces champs en base de données et dans la classe Java. Plus tard mon tuteur me montra que la librairie Hibernate qui était utilisée pour faire les liens entre la base de données et la classe Java crée le script de la base de données avec les informations qu'elle disposait avec les classes et comme nous n'avions pas plus d'informations que celle utilisée par l'API donc renseigner dans les classes. Donc Hibernate génère le script complet.

Ensuite j'ai commencé à intégrer la carte à la page d'accueil ce qui fut un peu complexe car le projet utilisait une syntaxe objet alors que React est fait pour une syntaxe fonctionnelle j'ai donc pris un peu de temps pour réussir à trouver une solution pour utiliser la syntaxe objet qui m'est obligatoire pour utiliser les informations de l'API de manière correcte en respectant les attentes de la librairie MobX qui est pensée pour les syntaxes objets. Après avoir trouvé une solution je commence donc à créer mes marqueurs pour les clients qui possèdent des positions définies pour ne pas avoir de problème qui pourraient faire planter le site. Ensuite j'ajoute aux marqueurs de petites bulles lorsque les survolent qui affiche le diminutif ou le nom du client qui sont uniques car vérifiés lors de la création du client. Mais avoir uniquement le diminutif ou le nom n'était pas



suffisant j'ai donc regarder comment je pouvais afficher plus d'information. Dans une autre parti du projets les informations des clients sont renseigner dans une grand fenêtre/pop-up toutes les informations du client. Je décide donc de les réutiliser et de l'afficher lorsque l'on clique sur le marqueur. J'ai donc fait un point avec mon tuteur pour validé mon idée ou non et il a bien aimé et à validé.

Ensuite je me tourne vers le formulaire pour trouvé une solution pour enregistre les coordonnées du client de manière efficace. Dans la documentations de Leaflet il y a un marqueur que l'on peut déplacé. J'ai donc créer un petit carte dans le formulaire de création et de modification avec un marqueur déplaçable pour positionner les clients. Ce qui facilite la saisie des informations plutôt que de rentrée les coordonnées à la mains.

Mais déplacé un marqueur n'était pas optimale ni agréable pour avoir une position assez précise j'ai donc commencé a me renseigner sur des Apis qui pourrais renvoyer des coordonnées gps après avoir fait une recherche d'adresse. Je trouvais donc l'api de google map qui proposais se service mais avec un quantité de recherche limité ce qui ne me plaisait pas j'ai donc continuer mes recherche et je suis tombé sur l'api des adresses du gouvernement français qui laisse à disposition, gratuitement et sans limite cette api. Le seul inconvénient et que l'api ne fonctionne que en France métropole, DOM et TOM. Mais étant donné que infotel et une entreprise français et qui n'ai pas présente sur le marché international tout c'est client et confrère son situé en France ou pose des agences en France. Je commence donc à ajouter un champ de texte permettant de faire notre recherche d'adresse et j'ai mis un bouton qui déclenche la recherche avec l'api et qui prend en compte l'adresse saisie pour faire ca recherche et qui me renvoie qu'un seul résultat en fonction des réponse et du pourcentage de correspondance le plus haut calculer par l'api. Avec le résultat de l'api je récupérer donc les coordonnées qu'il a trouvé pour déplacé la carte et quelle se centre au-dessus de manière animé comme si l'on y volais. En testant je me suis rendu compte que si l'on cherchais une adresse c'était pratique mais si l'on chercher une commune ou une ville le zoom était trop grand et l'on ne voyais rien. J'ai donc pensé a changé le zoom en fonction du type de résultat de l'api. Si le résultat renvoyer est une adresse ou un lieux dieu l'on zoom de manière proche pour identifier l'adresse. Sinon l'on garde un zoom plus faible pour avoir une vue d'ensemble au niveau de la commune ou de la ville.

Ensuite j'ai donc synchroniser l'emplacement du marqueur et le résultat de la recherche de l'api pour que l'on place le marqueur directement sur la recherche pour rendre la fonctionnalité vraiment efficace. Trouvant la fonctionnalité de recherche efficace je l'ai donc intégré a la carte de la page d'accueil pour se placé ou cherche de manière géographique des clients.

Après quelque utilisation du formulaire j'ai trouvé dommage que le sélecteur de commune du clients n'ai pas synchroniser avec la carte. J'ai donc commencer a chercher une solution permettant de mettre le sélecteur a jour en fonction de la position du marqueur ou de la recherche avec l'adresse mais ceci était trop complexe pour moi. Je me suis donc rabattu sur le fait de mettre à jour la carte en fonction du sélecteur. Si une commune était sélectionner alors la carte et le marqueur se positionner dessus grâce a l'api. J'ai donc du faire en sorte que lorsque la valeur de la commune changeais alors je fessais une recherche avec l'api et je position le marqueur et la carte dessus mais je devais faire en sorte de ne pas faire trop de recherche sur l'api pour éviter de la surchargé inutilement et de ne pas ralentir le site.

Après avoir fait ses fonctionnalités qui était ma mission pour ce stage et que mon tuteur était cher des clients donc ne pouvais pas vérifier et validé ce que j'avais fait. J'ai commencé à regarder dans le projet des chose qu'il restait à faire qui sont généralement signalé par des commentait commençant par TODO ce qu'il restait à faire dans le projets. Je suis donc tombé sur des tests unitaires à mettre à jour dans l'api. Les tests unitaires sont des tests que l'on utilise dans le projet pour vérifier que avec des entrés données l'on obtiennes un telle résultat, l'on les utiliser pour vérifier le bon déroulement d'une fonctionnalité. Par exemple dans l'api les tests unitaires sont utile pour vérifier le bon déroulement de la couche « service » qui est la couche centrale de l'api et qui modifie les données et fait des vérifications. Lors de c'est tests l'on vérifie si les vérification fonctionne bien et stop bien le fonctionnement du service si une erreur arrive.

Après ses mises à jours je commençais a regardé un autre « TODO » qui était ici pour actualiser automatiquement le token de l'utilisateur si il etait encore connecter après la date d'expiration du token pour évité a l'utilisateur de se reconnecter. J'ai donc commencer à regarder le fonctionnement de la gestion des accès dans l'api mais je ne

comprenais pas tout et je me suis retrouvé bloqué. Heureusement le lendemain mon tuteur de stage est de retour et j'ai donc pu faire un points sur les ajouts que j'avais fait avec la barres de recherche et les formulaires. Il m'a aidé à optimiser et à rendre mon code plus lisible. Après l'on a vérifier mes tests qui avais bien été fait et enfin je pu lui demandé de l'aide pour qu'il m'explique les tokens. Il m'a donc expliqué que ce système de sécurité allait être changé car il n'est pas assez efficace et que l'entreprise avais trouvé une alternative permettant d'avoir une sécurité commune sur tout les sites. L'on a donc regarde une nouvelles taches que je pourrais faire avant la fin de mon stage est l'on a vue que la fonctionnalité des relations entre clients et confrère n'avais pas encore été fini dans l'api.

J'ai d'abord regardé ce qui avais été fait. J'ai vue que la partie Hibernate été créé donc les relations était bien dans la base de données. Mais par contre dans la partie service et controller il n'y avait que de quoi lire les données. Il manquais la partie pour la création, la modification et la suppression des relations. J'ai donc commencer à faire les services de création, de modification et de suppression. C'est donc avec les informations récupérer par les appels de l'api l'on va faire les vérifications de l'existence du client et du confrère. Apres avoir faites les services je fait donc faite les tests ce qui me permettre aussi de vérifier ce que je viens de faire si je n'ai pas fait d'erreur. Les tests sont simple à faire étant donnée qu'il n'y a que trois cas possible se sont le premier réussite, le deuxième erreur le clients n'existe pas, troisième et dernier le confrère. Donc je créer mes tests avec des valeurs par défaut qui sont ici une relation, un client, un confrère. Ensuite je fait mon premiers cas, c'est-à-dire je lance ma fonction avec ma relation grâce à la librairie Mockito, je peux mocker/imiter un service et définir la réponse de se service. Ce qui va me permettre que lors de la vérification du client et du confrère je puis renvoyé ce que j'ai créer donc il les considère qu'il existe et va créer la relation, sauf que Mockito va validé le test et bloqué la création et arrêter le processus juste avant pour évité de surchargé la base de données. Pour le second cas l'on va faire en sorte que pour les vérifications le service ne va pas renvoyer de client alors nous auront donc a vérifier si notre fonction renvoie donc bien un message d'erreur et qui correspond a celui que l'on a définie. Et pareil pour le troisième cas juste mais avec le confrère manquant à la place du client. Enfin je peux lancé mes tests pour vérifier ce que j'ai fait et validé ma couche service.

Ensuite je commence à créer mes nouvelles fonction dans le controller pour pouvoir appeler les services que je viens de créer depuis l'extérieur de l'api pour créer je rajouté un appel en POST, qui est un verbe http qui permet de définir l'action d'une requête, et le nomme « /relation » c'est-à-dire que si je fais une requête le lien vers api avec /relation a la fin et que je mets dans la requête la relations a créer alors l'api utilisera cette fonction. Pour la modification j'utilise un appel PATCH, qui est le verbe http pour les modification. Et pour la suppression l'on utilise le verbe DELETE.

Maintenant que l'api est fini je commence donc a rajouté les appels api pour les création de chaque relation dans le formulaire de création du clients et du confrère car la parti de drag and drop permettant de renseigner les relations était déjà créé. Mais pour les formulaires de modifications il y avait plus de travail. Il fallait pense a comparer les relations du formulaire et les anciennes dans la base de données. Les nouvelles relations non existante avant vont donc avoir un appel api pour les créer. Ensuite les relations qui existait est existe encore uniquement si elle sont modifier auront un appel api de modification. Et enfin les relations qui existait mais qui n'existe plus auront un appel api pour les supprimer.

Mais je rencontrais un problème lorsque je me retrouvais à faire beaucoup d'appel api, l'api crashais car toutes les requetés arrivais en même temps. J'ai donc pris du temps mais fini par trouvé sur internet que d'utiliser un for plutôt qu'un foreach permet de faire les appels api en série plutôt qu'en parallèle.

## CONCLUSION

Lors de ce stage chez Infotel Conseil sous la tutelle de Arnaud LIMOUZIN, j'ai pu effectuer la missions qui m'avais était attribué et ce plus vite qu'attendu ce qu'y m'a permis de découvrir encore plus de chose coté back-end. Ce projet m'a permis de me rendre compte que je suis capable de bien avancé et d'être efficace même en autonomie, en me renseigner, à organiser un projet et fournir un travail propre et simple à reprendre.

Malgré le fait que mon tuteur ne soit disponible que un jour par semaine j'ai su me débrouillé et avancé sur ce projet de façon autonome. Et les personnes de l'équipe on fait tous leurs possible pour m'aider et me m'être à l'aise.

Ce stage, en adéquation avec mon projet professionnel m'a permis de découvrir plus amplement le Java et le Java script, des langage que je connaissais peu auparavant. Il m'a conforté dans mon projet de travailler dans le développement, puisque j'ai pu réaliser les missions demandées dans le temps qui m'était imparti. J'ai su trouver des solutions aux problèmes auxquels j'étais confronté. J'ai apprécié me confronter à la réalité du développement en entreprise et travailler sur un projet. J'ai hâte de mettre à profit les enseignements des expériences de ce stage de cinq semaines.

## ANNEXES

### Sources :

JAVA : <https://www.lebigdata.fr/java-guide-complet>

Java Hibernate : <https://hibernate.org/orm/>

Java-script REACT : <https://fr.reactjs.org/>

Leaflet-React : <https://react-leaflet.js.org/>

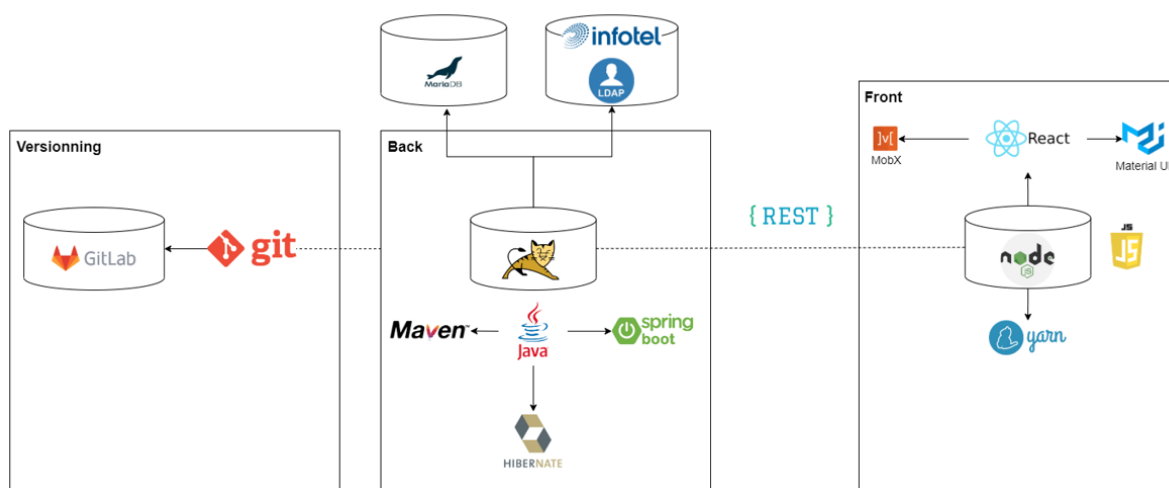
IntelliJ / IDEA : <https://www.jetbrains.com/fr-fr/idea/>

Visual Studio Code: <https://code.visualstudio.com>

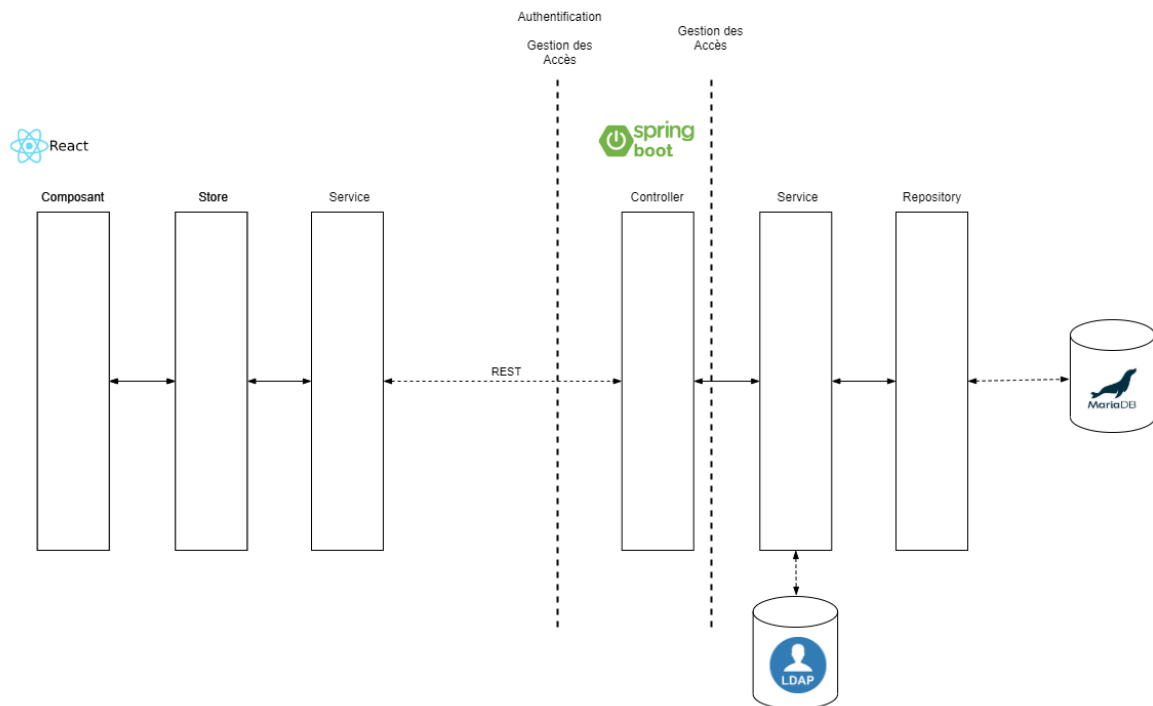
GIT : <https://www.atlassian.com/fr/git/tutorials/what-is-git>

Version Sémantique : <https://semver.org/lang/fr/>

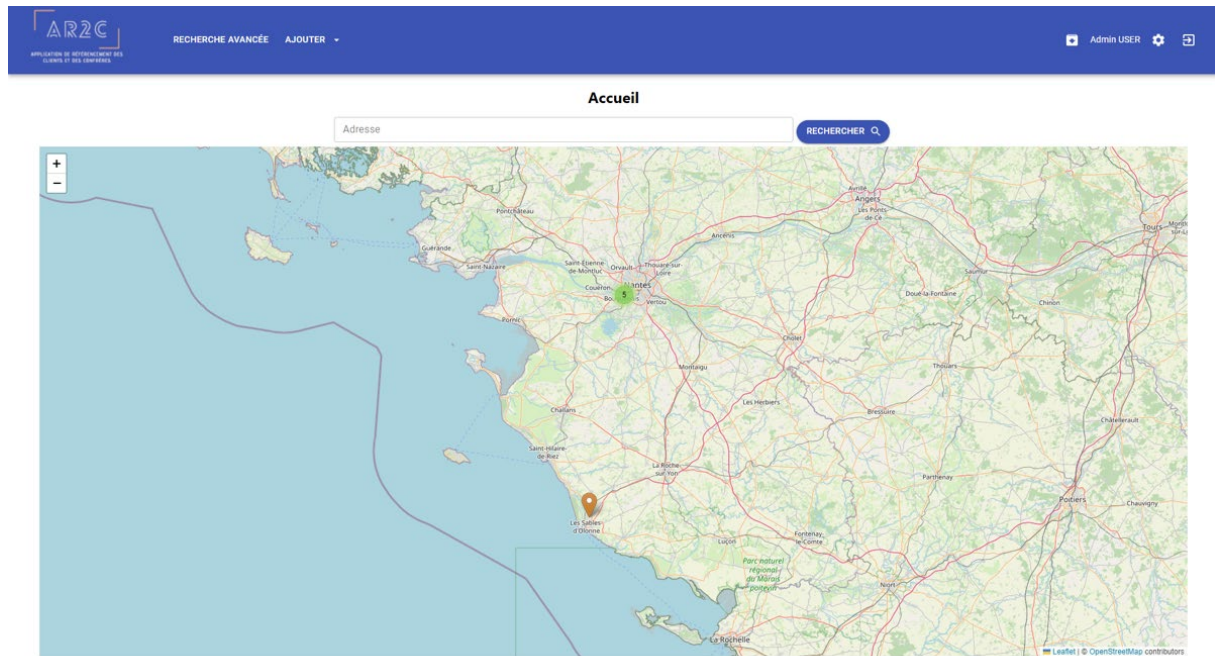
### Schémas :



Annexe n°1 : Schéma des technologies et communication de l'application

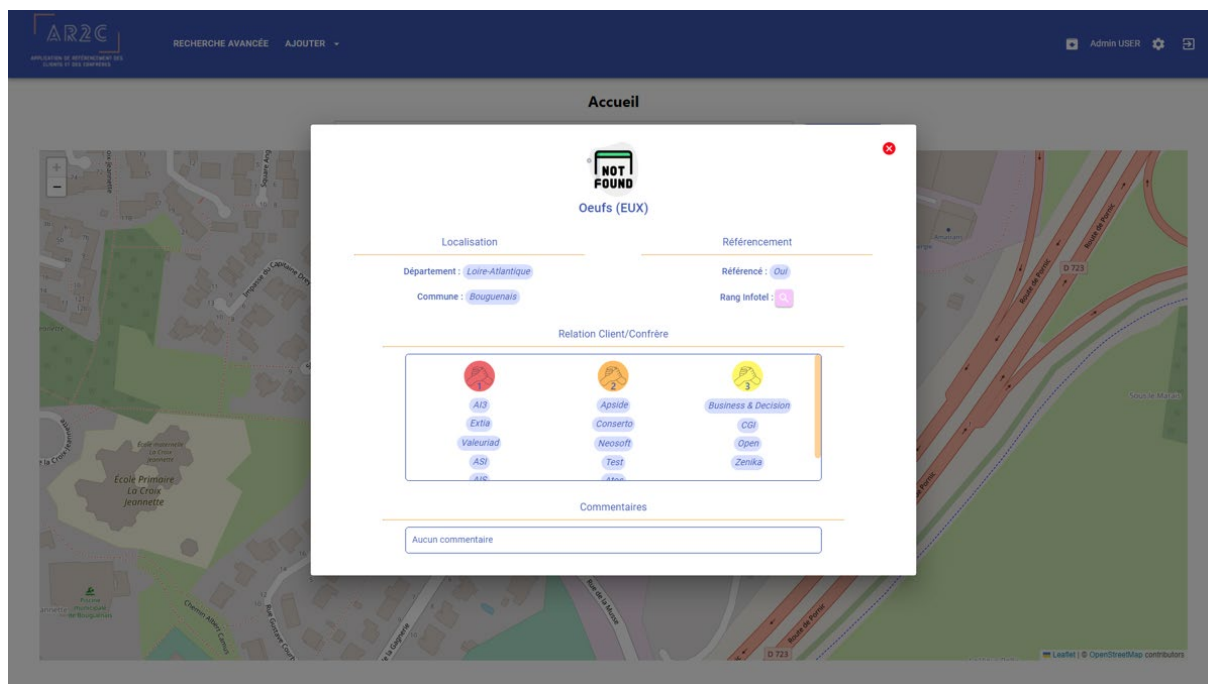


Annexe n°2 : Schéma d'architecture des couches



Annexe n° 3 : page d'accueil à fin de stage





Annexe n° 4 : pop-up info client à fin de stage

The screenshot shows a web application interface with a dark blue header containing the AR2C logo and navigation links. The main content area is titled 'Création d'un nouveau client' and is divided into several sections: 'Informations générales' (Nom, Diminutif, Logo), 'Localisation' (Département, Commune, Adresse), 'Référencement Infotel' (Référéncé, Rang Infotel), 'Relation Client/Confrère' (a grid of icons and names), and 'Commentaires'. The background is a map of France.

Annexe n° 5 : page formulaire création client à fin de stage