



Rapport d'activité en entreprise

Entreprise : Eurice

Tuteur en entreprise : Yannick Lavoix (Eurice)

Septembre 2017 - Septembre 2019, Contrat de professionnalisation

Remerciements

Tout d'abord, je remercie M. Yannick Lavoix, mon maître d'apprentissage, pour m'avoir encadré durant cette alternance.

Je souhaite remercier l'ensemble de l'entreprise Eurice qui a rendu cet apprentissage possible et m'a permis d'acquérir de nombreux nouveaux savoirs et compétences.

Enfin, je remercie l'ESGI pour avoir rendu cette alternance possible et m'avoir fait bénéficier de cours enrichissants avec une équipe pédagogique de qualité tout au long de l'année.

Table des matières

Remerciements	1
Table des matières	2
1 Introduction	3
2 Contexte Entreprise	4
2.1 Entreprise d'accueil	4
2.2 Organisation de l'entreprise	5
2.3 Contexte Métier	5
2.3.1 Callibri	5
2.3.2 Site client	6
2.4 Focus sur le service du stage	6
3 Les Missions réalisées	7
3.1 Enjeux et cadre des missions effectuées	7
3.2 Présentation d'une mission à fort enjeux : le site client	7
3.3 Bilan et recul sur les missions	10
4 Conclusion	11

1 Introduction

Dans le cadre de ma 4^{ième} année d'architectures des logiciels en apprentissage, je devais intégrer une entreprise, j'ai continué cette année l'alternance que j'avais commencé dans l'entreprise Eurice pendant mon année de licence professionnelle PRISM.

Le but de cette alternance est de nous confronter au monde professionnel, d'acquérir une expérience en entreprise et enfin communiquer notre ressenti et notre travail effectué en entreprise à travers la rédaction de ce rapport.

Eurice fournit des solutions logicielles pour les centres d'appel que ce soit pour l'accueil téléphonique ou la gestion d'agenda. Je travaille au sein de l'équipe de développement.

Mes missions pendant ces 12 mois ont été de développer une application mobile pour la gestion d'agenda et la création d'une nouvelle version du site client .

Dans une première grande partie je présenterai l'entreprise. Dans une seconde partie je ferai une introduction des logiciels réalisés à Eurice, nécessaire pour comprendre le contexte de ma mission puis je présenterai ma mission et les différentes tâches effectuées. Dans une troisième partie, j'effectuerais un bilan professionnel et un bilan personnel de mon expérience au sein d'Eurice.

2 Contexte Entreprise

2.1 Entreprise d'accueil

L'accueil téléphonique en France est principalement utilisé par les professions libérales mais également pour les entreprises de toutes tailles souhaitant avoir un accueil téléphonique. Cette profession dispose de sa chambre professionnelle le « SIST ».

Le SIST regroupe 60 centres d'accueil partout en France cela représente 8000 hôtesses d'accueil téléphonique et 150 millions d'appels entrant traités par an. Le principe de l'accueil téléphonique est de ne rater aucun appel téléphonique pour ne pas perdre de potentiel client.

Le fait de confier sa ligne téléphonique à des professionnels permet d'avoir un accueil téléphonique de qualité.

Cela permet également de réduire drastiquement les coûts qu'aurait une entreprise à engager une ou plusieurs personnes à temps plein pour répondre aux appels.

Créée en 1992 par ses actuels dirigeants, Eurice est spécialisée dans la gestion d'appels entrants. Eurice est à la fois opérateur et fournisseur de solutions techniques de par ses services :

- L'accueil téléphonique fourni par un plateau de réception des appels au sein des locaux de l'entreprise, qui peut aussi parfois servir d'environnement de pré-production avant le déploiement de mise à jour chez les centres d'appels client.
- Le développement de solutions pour les centres d'appels : logiciel de gestion d'agenda, répartiteur d'appel, portail client et applications mobiles, tous les logiciels utilisés à Eurice et vendus au client, cela permet entre autre de connaître le besoin avec plus de précision car nous sommes clients de nos propres produits.

Chaque année, Eurice gère pour le compte de ses clients près d'un million d'appels décrochés en moins de 3 sonneries.

2.2 Organisation de l'entreprise

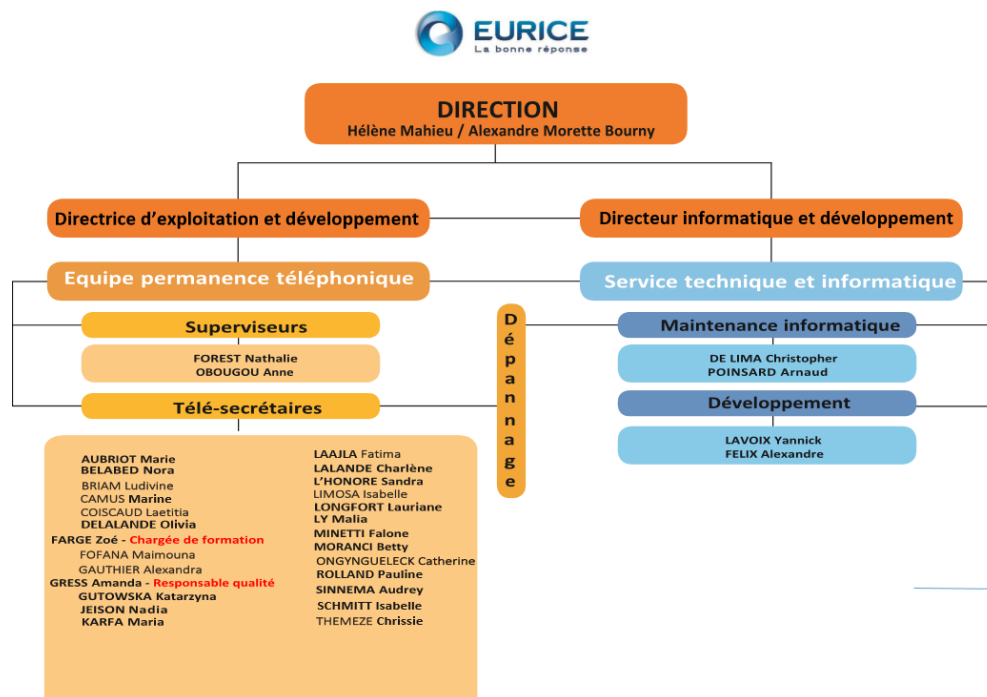


FIGURE 2.1 – Organigramme Fonctionnel

2.3 Contexte Métier

2.3.1 Callibri

Il s'agit de l'application principale pour un centre d'appel. Ce logiciel permet la gestion de plusieurs dossiers avec leur agenda, avoir une communication entre la secrétaire et le client via les instructions et les messages. un système de scripting soutient l'opératrice lors du traitement d'un appel, c'est le support du pré-traitement et du post-traitement de l'appel.

Callibri s'interface avec le système d'appel pour automatiquement ouvrir le dossier correspondant au numéro appelé, cela permet de gagner un temps conséquent et de rendre transparent l'appel aux yeux de l'appelant qui ne sait pas que son appel est décroché dans un centre d'appel.

L'application utilise pour la majeure partie de l'interface des pages web grâce à la librairie Essentials Objects utilisant un moteur chrome embarqué. Le reste de l'interface qui comprend le menu et la fenêtre native est réalisé en WPF.

Callibri étant utilisé sur plusieurs postes accédant à des ressources communes, le logiciel communique via messages UDP aux autres postes, pour, par exemple, éviter de prendre un rendez-vous sur le même créneau libre.

Fonctionnalités

Les fonctionnalités principales de [Callibri](#) sont divisés en plusieurs parties :

- Dossiers : [Callibri](#) permet la création de dossier qui représente un client, il contient les coordonnées de ce dernier, les différents numéros de téléphone ainsi que les éléments de facturation et de l'agenda.
- Gestion des instructions : ce sont les démarches à suivre ou des informations laissées par le client aux opératrices, cela peut indiquer les habitudes du client, des créneaux d'absence, informations d'accès ou autres.
- Scripting : le scripting permet de personnaliser le fonctionnement du dossier, cela peut être dans le traitement d'un appel ou d'un ajout de rendez-vous mais peut aller aussi jusqu'à l'affichage de fenêtres personnalisées lors d'un appel pour le dossier
- Messages : les messages sont le moyen de communication privilégié pour les opérateurs, ils permettent une communication direct au client.
- Gestion de l'agenda : c'est la partie centrale de [Callibri](#) ou sont affichés les rendez-vous, les créneaux libres et les informations pertinentes pour les créneaux pris
- Organigrammes d'entreprise : [Callibri](#) a été prévu pour répondre aux besoins de clients divers et variés. L'organigramme d'entreprise permet rapidement de visualiser la hiérarchie de l'entreprise, ses différents services ainsi que chaque salarié attaché à ces services.

2.3.2 Site client

Le site client comme son nom l'indique est la partie accessible au client de l'agenda [Callibri](#), les fonctionnalités diffèrent légèrement par rapport à l'agenda de l'opérateur mais l'interface visuelle reste exactement la même, le site permet aussi de consulter les messages reçus et de les archiver, de même pour les instructions, le client a aussi accès à l'agenda web RDV Médicaux s'il fait partis de leurs.

2.4 Focus sur le service du stage

Je fais partie de l'équipe développement qui est composé de Yannick Lavoix et de moi-même, notre objectif est d'ajouter des fonctionnalités aux logiciels existant, les maintenir et créer de nouvelles solutions. notre interlocuteur principal est le directeur de l'entreprise Mr. Morette-Bourny, nous sommes parfois mis en relation avec des clients ou des partenaires tel que RDVMedicaux ou Doctolib.

3 Les Missions réalisées

3.1 Enjeux et cadre des missions effectuées

Ma première mission était la réalisation d'une application mobile multi-plateforme reprennant les fonctionnalités du site web client tout en prenant en compte les différences d'expérience utilisateur qu'apporte une interface tactile.

Les outils et technologies utilisées pour cette mission sont les suivantes :



- Ionic + cordova
- Git comme outils de versionning de code source
- Npm comme gestionnaire de librairie
- Visual Studio Code
- GitLab pour l'hébergement du code source et la gestion de projet

Mon choix c'est porté sur Ionic pour la rapidité avec laquelle on peut développer une application et ce sur les plateformes iOS et Android sans avoir à faire du code natif spécifiques aux plateformes de plus j'avais déjà des compétences en technologie web.

La seconde mission était la refonte totale du site web client qui contient aussi l'API pour l'application mobile, ce nouveau site nommé web3 doit avoir les mêmes fonctionnalités que la version précédente tout en utilisant la puissance des nouvelles technologies.

- client-side : version Progressive Web App de l'application mobile
- server-side : Asp.net Core
- persistance : base de données NexusDB
- Git comme outils de versionning de code source
- GitLab pour l'hébergement du code source et la gestion de projet
- Visual Studio Code

3.2 Présentation d'une mission à fort enjeux : le site client

Le site web client est la facade directement visible de nos clients c'est donc un produit crucial pour l'entreprise, il m'a été confié la création du nouveau site web, appelé en interne web3 car il s'agit de la troisième refonte du site.

J'ai eu totale liberté pour le choix des technologies utilisées, le web2 est basé sur un stack vieillissant : asp.net webforms et jquery. De plus, la qualité du code et sa maintenabilité est extrêmement faible rendant difficile et ce de manière exponentielle, l'ajout de fonctionnalités et la correction de bugs.

Pour les choix techniques j'ai choisi de rester sur l'utilisation du langage C# pour deux raisons, tout d'abord je maîtrise ce langage depuis de nombreuses années et ensuite c'est le langage utilisé dans

la réalisation de tout les autres logiciels au sein de l'entreprise ce qui assure une certaine homogénéité et enfin cela me permet aussi de reprendre le code de l'ancien site client plus facilement.

L'architecture du web2 était complètement incorrecte, utilisant le C# comme langage purement procédural plutôt que comme langage objet, aucune des bonnes pratiques ne sont respecté ce qui rend le code hautement dupliqué et difficile à maintenir, de plus aucun test unitaires n'était mis en place rendant difficile la non-regressions.

La première étape fut d'analyser ce qui appartenait purement à la partie métier et d'isoler les composants techniques relatif à l'application et à la persistance des données :

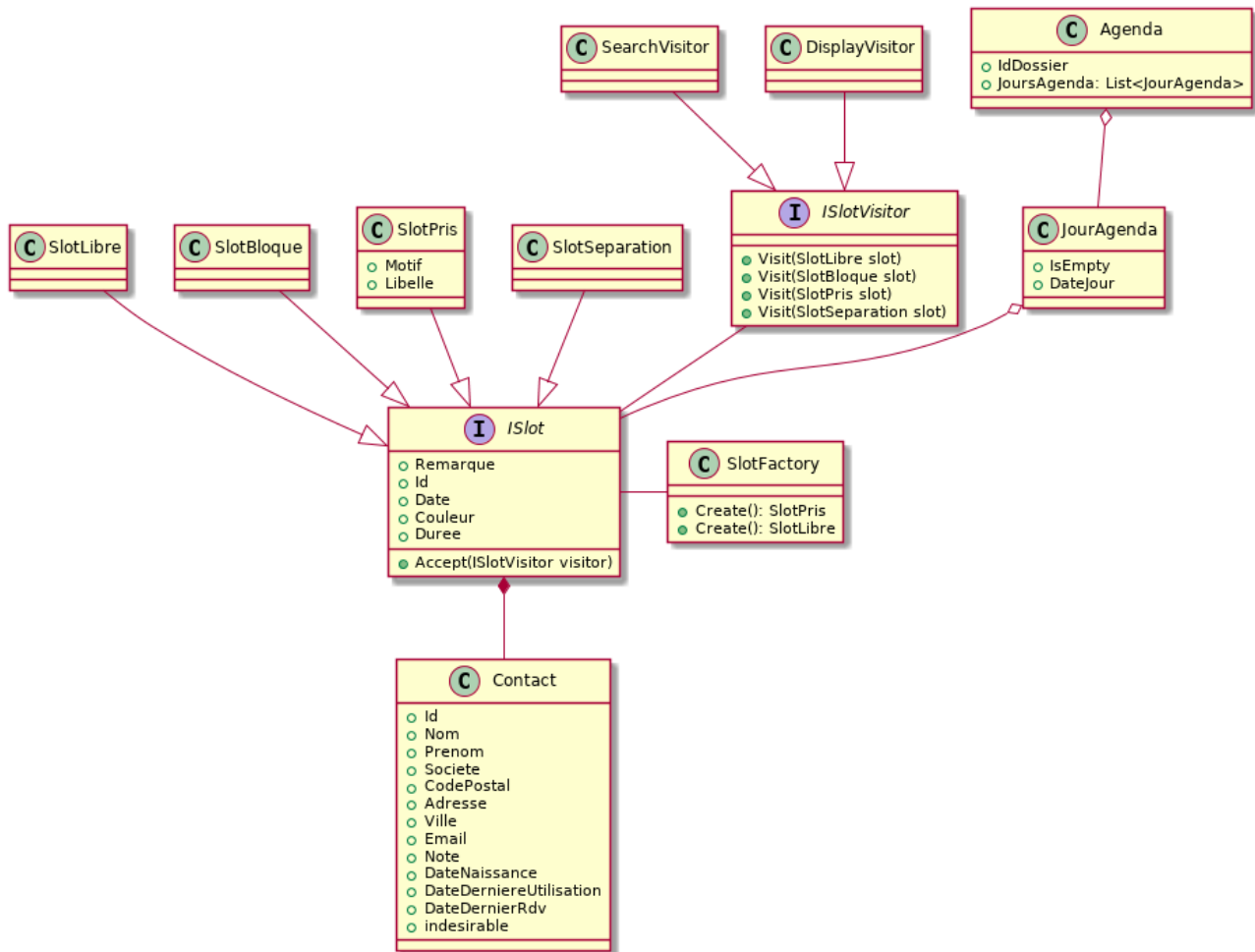


FIGURE 3.1 – classes métier pour l'agenda

Cette partie ne concerne que les classes métier de l'agenda, chaque agenda est composé de jour qui sont eux-mêmes composés de créneaux (slot) qui peuvent avoir plusieurs états, tout les états ne peuvent pas être représenté de la même manière car ils correspondent plutôt à des sous état tel que le fait qu'un rdv pris soit un rdv web pris via le site web Doctolib ou de l'un de nos autres partenaires. J'ai du trouver et isoler les propriété et différent états des slot par moi pour avoir une architecture correcte, non présente sur le web2.

Avec ces classes réalisées je me suis penché sur l'implémentation du reste de l'architecture, j'ai opté pour une architecture hexagonale, la partie métier serait un noyau central avec lequel le reste de l'application interagirait, cela permettrait ainsi de vérifier les règles métier avec des tests unitaires et d'isoler ces classes avec le reste de l'application permettant par ailleurs de porter ce noyau rapidement dans d'autres applications futures.

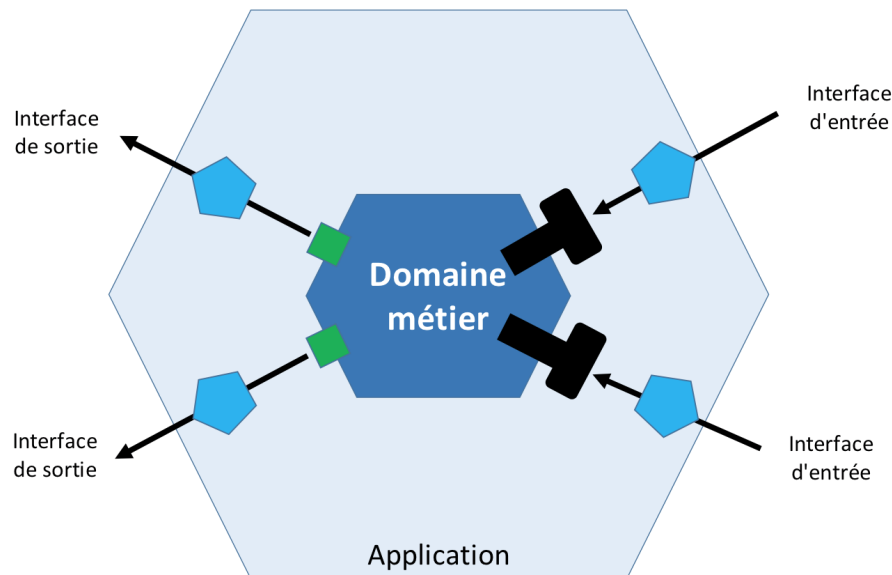


FIGURE 3.2 – Architecture Hexagonale

Avec une architecture hexagonale, les classes métier communiquent vers l'extérieur via des "port" et des "adapter", les ports sont en fait des interfaces qui définissent des rôles à remplir et les adapters sont les implémentations de ces interfaces, les adapters "remplissent un contrat".

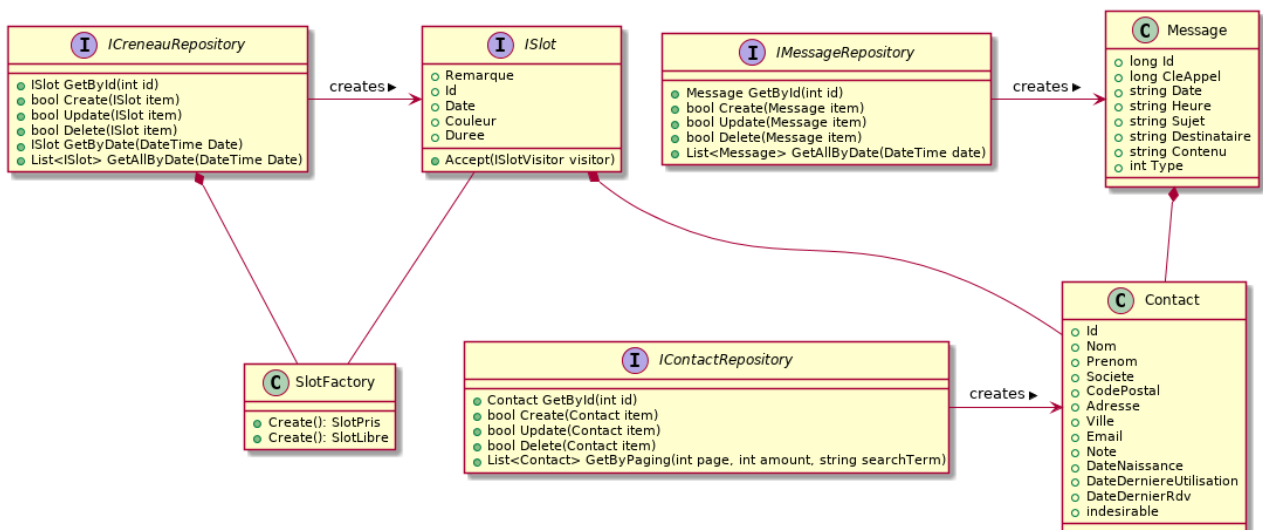


FIGURE 3.3 – implémentation des repository

Les Repository sont l'implémentation d'une interface d'entrée, ils permettent l'interaction de façon abstraite avec un support de persistance tel qu'une base de données, le noyau métier ne connaît pas les dessous techniques, cela facilite les tests unitaires puisque l'on peut alors utiliser des adapters qui retournent des données fixes pour les tests. Dans le cas du web3 la grande partie des interfaces d'entrée

vont être de l'accès aux données tandis que pour les interfaces de sortie cela peut être les controller d'ASP.net core ou les requêtes http envoyées par l'application mobile.

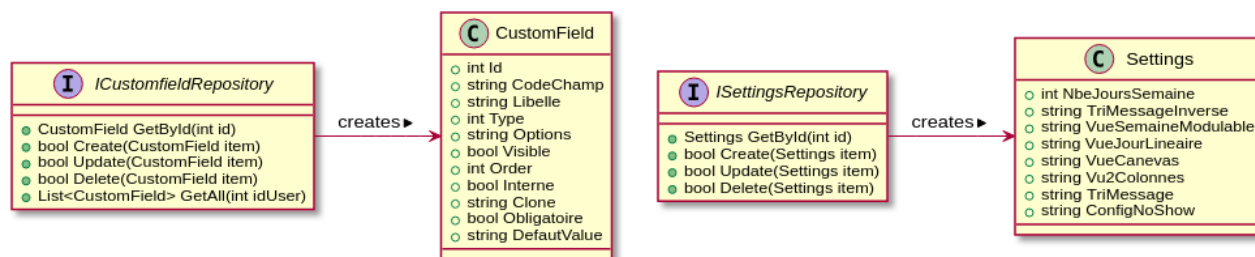


FIGURE 3.4 – reste des classes métier et repository

j'ai ensuite entièrement séparé le noyau métier dans une librairie à part référencé dans le projet principal pour plus de clareté et pour permettre l'évolution indépendante du site et du noyau qui ont chacun leur depot git, le code client-side est dans le depo du site mais j'ai pour objectif de rendre ce code commun entre l'application mobile et le site web.

3.3 Bilan et recul sur les missions

Pour les deux missions j'ai eu totale liberté dans mes choix qui fut à la fois un avantage mais aussi une preuve du manque de gestion de projet au sein de l'entreprise, je n'avais aucun cahier des charges et les règles métiers n'étaient pas clairement définies, j'ai du donc établir par moi même le cahier des charge et lister les nouvelles fonctionnalités en me basant sur l'ancien site web et son code qui n'a d'ailleurs pas facilité la tâche.

pour l'application j'ai du aussi me baser sur le website et m'auto-former aux bonnes pratiques à appliquer à l'expérience utilisateur et l'interface graphique puisqu'une application n'a pas du tout les mêmes contraintes qu'un site web.

En prenant du recul je me suis rendu que le choix du framework Ionic pour l'application mobile était un choix judicieux mais seulement à court terme, il est en effet assez difficile de maintenir toute application basé sur le javascript et l'environnement node car les outils et la qualité d'un environnement de développement basé sur ces techno est assez médiocre, les librairie deviennent incompatibles ou non d'une version sur l'autre, il est assez difficile de débogger certaines erreurs à cause de l'épaisseur conséquente du stack de technologies utilisées. Si je n'avais pas eu d'aussi forte contraintes de temp pour produire un résultat j'aurais utilisé le framework Xamarin qui aurait encore plus augmenté la portabilité du code puisque j'aurais pu utiliser le noyau métier du web3 dans l'application ce qui aurait permis de réduire le temp de développement et peut être arriver à produire un résultat aussi rapidement qu'avec l'utilisation du framework Ionic, au moment des choix technologique j'ai peut être donc fait preuve d'un manque de vision générale et sur le long terme.

4 Conclusion

Sur le plan professionnel cette année a été très enrichissante, j'ai pu renforcer mes compétences dans les framework Angular, Ionic et .NET que j'avais commencé à acquérir lors de mon année de licence professionnelle, sur le projet de site web j'ai pu appliquer pleinement mes compétences dans l'architecture des logiciels avec la mise en place de méthodes de développement et de gestion de projet agile dans un premier temps puis avec l'application des bonnes pratiques de développement pour les langages orientés objet avec l'utilisation correcte de design pattern et le respect des règles de base pour la création de logiciels maintenable, performant et fiables.

D'un point de vu personnel j'ai appris beaucoup de choses sur la gestion d'agenda et les centres d'accueil téléphonique notamment en terme de contraintes techniques qui sont imposées par des clients réticent au changement ainsi que les enjeux technologiques de ce cœur de métier qui doit se faire peau neuve pour rester compétitif.

Le marché de la prise de rendez-vous évolue et se tourne de plus en plus vers le web et le mobile, j'ai apporté à Eurice les compétences pour la réalisation d'application Mobile sur les plateformes Android et iOS.

