

# Rapport de stage

## 4eme Année Architecture des logiciels

Expaceo

ESGI



Damien Chaudois  
2017-2018

## Remerciements

Je tiens à remercier toute l'équipe d'Expaceo qui m'a accueillie, formé, encadré, et avec qui j'ai passé un très bon moment. Je pense en particulier à deux personnes : Cédric Del-Nibbio, PDG d'Expaceo, qui m'a suivie tout au long de mon stage malgré sa position, et Pierrick Gurlain, qui m'a enseigné toute ce que je sais d'une application MVC en ASP .net et ce malgré mes lacunes dans le domaine du web. Sans le sérieux, le professionnalisme et l'implication de ces personnes dans ma formation, non seulement je n'aurais jamais pu intervenir comme je l'ai fait dans les projets de l'entreprise, mais je n'aurais pas autant progressé tant dans les technologies web que dans les patterns qui les accompagnent.

J'ai aussi une pensée pour Pierre, un autre stagiaire avec qui j'ai travaillé en binôme sur ce projet, qui malheureusement a eu plus souvent raison que moi quand il s'agissait de résoudre les bugs.

## Sommaire

### **TABLE DES MATIERES**

---

1	Présentation d'Expaceo .....	5
1.1	L'activité d'Expaceo .....	5
1.2	L'organigramme .....	6
2	Le projet.....	7
2.1	Présentation de l'application .....	7
2.2	Présentation de l'architecture de l'application.....	8
2.2.1	Le JavaScript.....	8
2.2.2	L'ASP .net MVC 5.....	9
2.2.3	SQL Server .....	10
2.3	Gestion de projet .....	11
2.3.1	Avancement du projet .....	11
2.3.2	Intégration continue .....	11
2.3.3	Architecture .....	12

## Introduction

En tant qu'alternant en école d'ingénieurs, j'ai travaillé sous un contrat de professionnalisation dans l'entreprise Expaceo, d'octobre 2017 à aujourd'hui.

Ce rapport a pour but de présenter, résumer, et synthétiser les travaux que j'ai effectués et les connaissances que j'ai acquises au sein de cette entreprise pour la période 2017-2018.

Sur cette période, j'ai travaillé en tant que développeur, principalement web full-stack, sur le projet de gestion de feuille de présence et de congé de l'entreprise. Dans ce rapport, je commencerais par vous présenter le contexte de mon stage, j'exposerais ensuite la nature de mes travaux effectués, et je conclurais sur ma rétrospective de cette période et ce qu'elle m'a apporté.

# 1 PRESENTATION D'EXPACEO

---

## 1.1 L'ACTIVITE D'EXPACEO

Expaceo est un cabinet de conseil français spécialisé autour des nouvelles technologies Microsoft proposant des prestations de conseils et d'expertise.

Fondée en 2009 par son président M. Cédric Del Nibbio, cette société par actions simplifiées dispose d'un effectif de 64 personnes et d'un chiffre d'affaire de 3.8 Millions d'euros en fin d'années 2017.

Par rapport à fin 2016, Expaceo a connu une croissance d'effectif de l'ordre de 110%

Ses principaux domaines d'intervention sont :

- Le conseil
- Architecture / L'édition logicielle
- La formation

L'entreprise coopère avec des clients dans plusieurs secteurs comme :

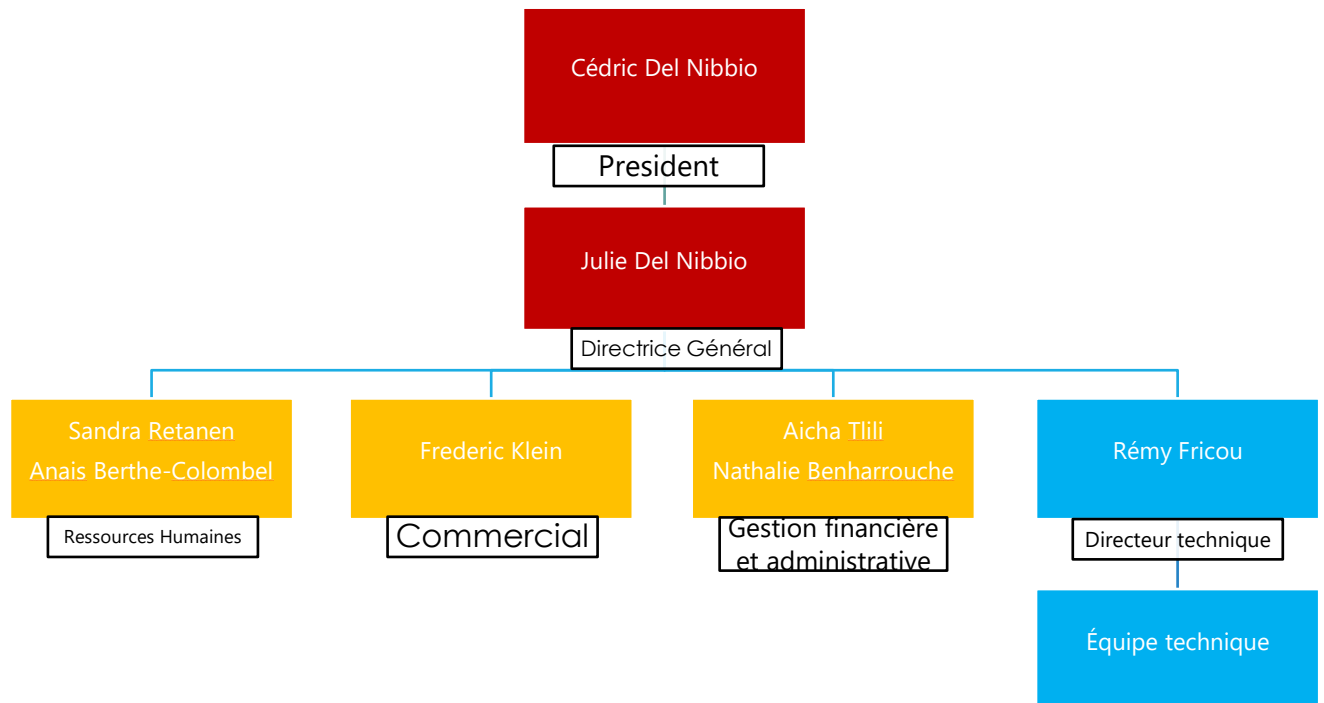
- L'industrie
- Le transport
- La finance
- La télécommunication

Parmi ses clients on peut citer : La SNCF, Bolloré Logistics, Qualitelis.

Les collaborateurs d'Expaceo apportent leurs expertises aux clients grâce à leurs compétences diverses dans de divers langages, des technologies diversifiées et des Framework performant.

L'entreprise cherche à toujours garder un fort taux de veille technologique en organisant chaque mois des présentations sur des technologies et où les collaborateurs échangent sur leurs connaissances et peuvent mieux se connaître car comme les employés travaillent pour des clients différents ils ne se côtoient pas tous les jours. Ces présentations servent à renforcer l'unité et la cohésion au sein de l'entreprise.

## 1.2 L'ORGANIGRAMME



En temps qu'alternant, je me situe bien entendu dans l'équipe technique.

## 2 LE PROJET

---

### 2.1 PRESENTATION DE L'APPLICATION



Au début d'Expaceo, les congés étaient pris en envoyant un mail aux ressources humaines. L'entreprise a alors commencé à développer son logiciel de gestion des congés : Clerks manager, et ceux-ci sont maintenant gérés via celle-ci.

La même logique existe pour les feuilles de présence des prestataires chez les clients. La mission principale a été de permettre à l'application de gérer celles-ci en addition des congés. Un compte rendu d'activité (CRA) est un document essentiel pour toute entreprise de service informatique : il résume le temps de travail qui sera par la suite facturé au client au tarif journalier du prestataire. Pour être valide, il doit contenir plusieurs informations :

- Le nom du client
- Le projet sur lequel le prestataire a travaillé
- Le nombre de demi-journées de travail effectués
- La signature du prestataire et du client

Il a donc été demandé à l'application de donner la possibilité à tous les collaborateurs de remplir leurs jours de présence directement sur Clerks Manager, puis de générer le PDF qui sera ensuite soumis à validation par le client. En complément, un administrateur doit pouvoir consulter la liste des clients et de leurs projets, les prestataires et les validateurs associés, et pouvoir éditer le tout.

Ce projet compte 3 membres : Pierre Said, stagiaire en 3e année à EPITECH, Pierrick Gurlain, Architecte logiciel chez Expaceo, et moi-même. Pierre et moi avons les mêmes attributions, en travaillant chacun sur nos tâches, et Pierrick nous encadrait en vérifiant l'avancée du projet tous les vendredis et en redistribuant les nouvelles tâches.

## 2.2 PRESENTATION DE L'ARCHITECTURE DE L'APPLICATION

L'application web, comme la plupart des applications de ce genre, se découpe en 3 grandes parties : le front, le back, et la base de données. Le front a été réalisé en HTML/CSS, et les composants ont été codés avec le Framework Vue.js, une librairie jeune et en pleine expansion. Le Back a été réalisé en C# .net, plus précisément avec le Framework ASP, et la base de données est sous Microsoft SQL Server.

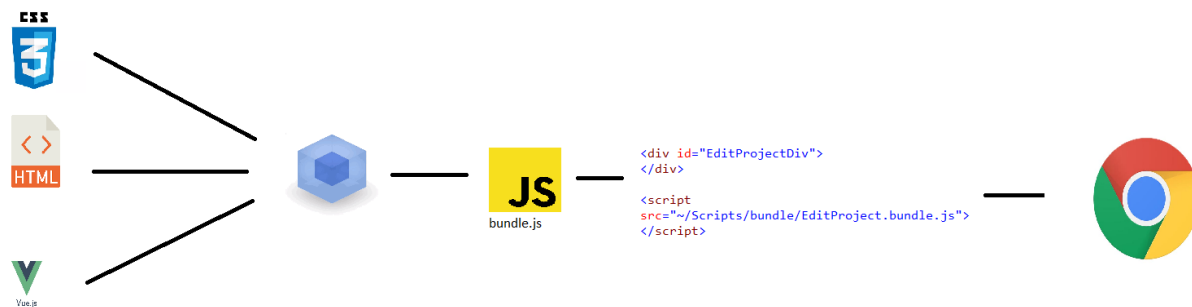
En addition de l'ASP, l'application suit l'architecture MVC, autour duquel est construit le Framework ASP.

### 2.2.1 Le JavaScript



# webpack

Le Vue.js est une bibliothèque Javascript dont le principal argument de vente est sa réactivité : il permet de lier des éléments HTML à des variables, et de reconstruire le DOM dès que ladite variable est mise à jour. Il s'agit d'une logique fortement orientée objet : une instance de l'objet vue est créée, et l'application va se lier aux attributs et méthodes de cette instance. Conjugué à webpack, il devient possible de faire des SPA (Single page application) pour simplifier la distribution du code au client. Webpack fonctionne avec Node.js : il va lire un fichier package.json à la racine du projet qui va lui indiquer quels sont les fichiers à surveiller, et va transformer toutes les dépendances de ces fichiers en un seul « bundle » qui sera ensuite envoyé au client, et qui regroupera le HTML, le CSS, et l'instance de vue.



L'objet vue va attendre un attribut très important : la 'Target' où il va insérer son Template, sous la forme de l'id de la balise cible. Ici, la cible est donc « EditProjectDiv », et elle sera remplie par le Template de l'objet Vue.js déclaré tout de suite au-dessus de lui entre des balises `<template>` `</template>` qui sont des composants propres au vue.js.



### 2.2.2 L'ASP .net MVC 5

Microsoft, au travers de son Framework ASP, met à disposition de ses développeurs un très puissant outil de gestion du routing des applications web. Bien qu'opaque pour un débutant comme moi, il finit par faire gagner un temps considérable en gérant de lui-même la distribution des vues à envoyer au client en fonction de sa requête. Par défaut, ASP se base sur les noms des fichiers pour savoir quoi renvoyer à qui. Ainsi, si un utilisateur demande l'URL /Project/EditProject, le code suivant :

```
public class ProjectController : Controller
{
    public ActionResult EditProject()
    {
        return View();
    }
}
```

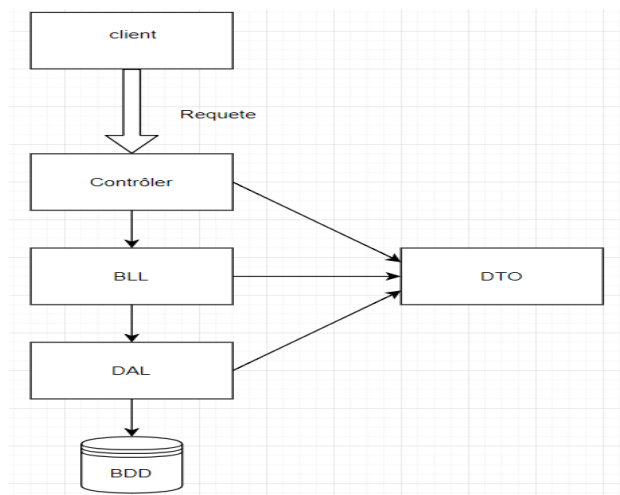
Renverra au client le fichier EditProjectVue.cshtml contenue dans le dossier View de la solution. Il est donc très simple de rajouter des pages, des routes, et des API à notre application. L'ASP possède un très puissant outil de parsing : préciser qu'une route attend un objet C# en paramètre autorise le Javascript à lui envoyer un objet à la structure similaire, et va automatique créer une instance de l'objet avec les données envoyé par le client. Il en va de même pour les variables de retour qui sont automatiquement sérialisées.

Comme énoncé plus haut, l'application suit une architecture Model-Vue-Contrôleur. Cela signifie qu'il y a trois grands axes dans le projet : les vues, représenté par le HTML/Css + vue js, les contrôleurs, décrit plus haut, et les modèles, qui ont pour rôle de faire transiter dans l'application une représentation des données stockées en base de données. À cela s'ajoute le découpage de la partie contrôleur en 3 sous-parties :

- La couche présentation, représentée par les objets contrôleurs
- La couche métier (Business Logic Layout - BLL), qui va s'occuper de faire les opérations et les calculs pour retourner des résultats cohérents au contrôleurs
- La couche d'accès au donné (Data Access Layout - DAL)

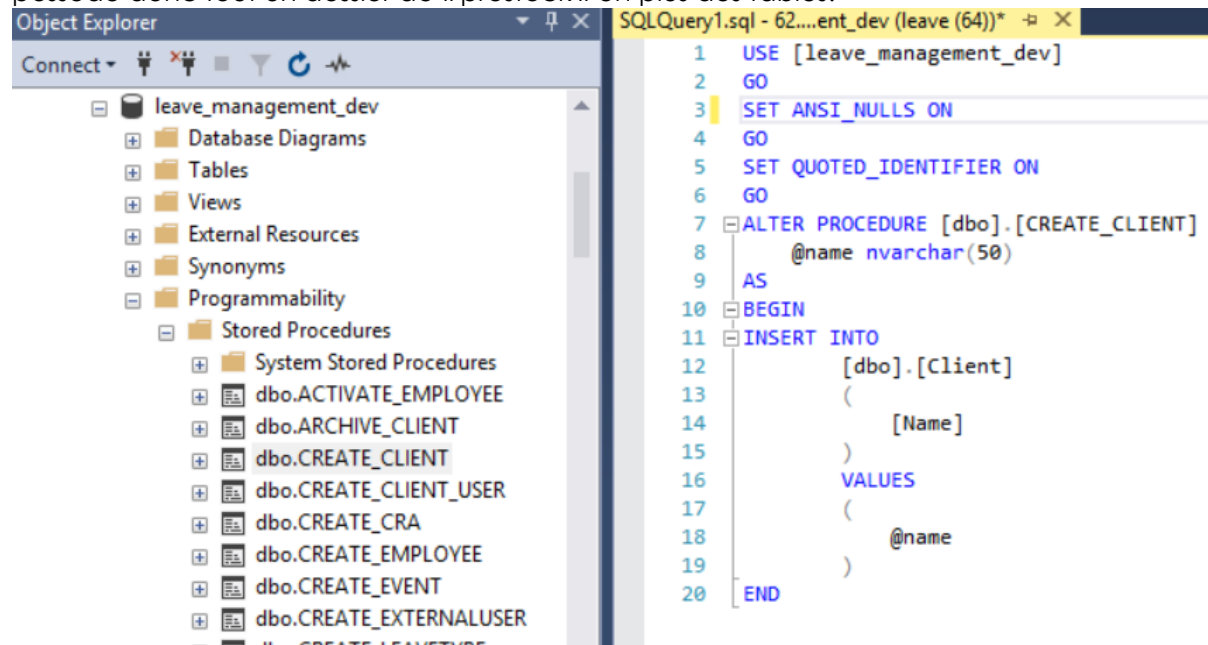
À cela s'ajoutent les modèles, appelés ici Data Transfert Object (DTO), qui vont circuler dans toute l'application.

Un tel découpage permet à une équipe de développeurs de facilement travailler d'une fonctionnalité à l'autre, puisque chaque section possède sa propre responsabilité. Si la demande de traitement des données devait être changé par le client, il suffit de modifier la BLL, si la base de données passe du SQL Server au MySQL, il suffit de modifier la DAL, etc....

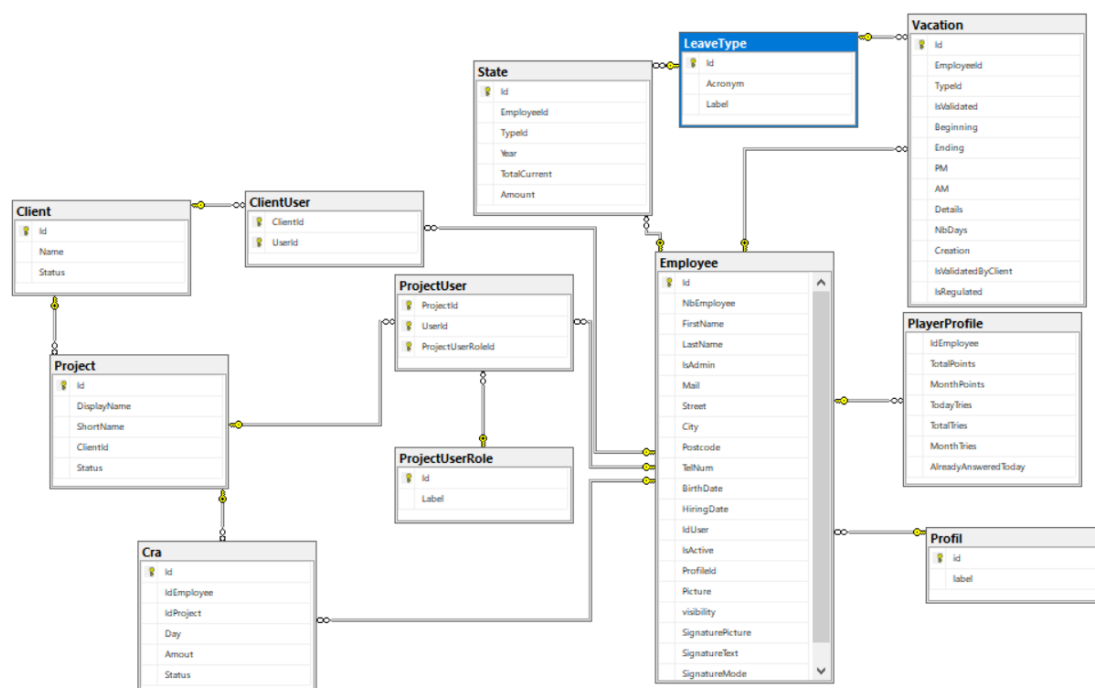


### 2.2.3 SQL Server

Dans l'application, seule la DAL a accès à la base SQL Server, mais même elle ne possède aucune requête, seulement des appels à des procédures stockées. Le projet SQL serveur possède donc tout un dossier de « prostock » en plus des tables.



La structure permet de comprendre quel est l'objectif métier de l'application : au centre, les employés qui sont rattachés au projet et au client via les tables n-n et qui ont une relation 1-n avec la table qui gère les congés. On voit ici qu'un CRA est le regroupement de deux éléments : un employé, et un projet. Ce projet est rattaché à un client, et enfin il y a une date pour chaque CRA. Un CRA est donc un employé pour un projet de un client sur une journée

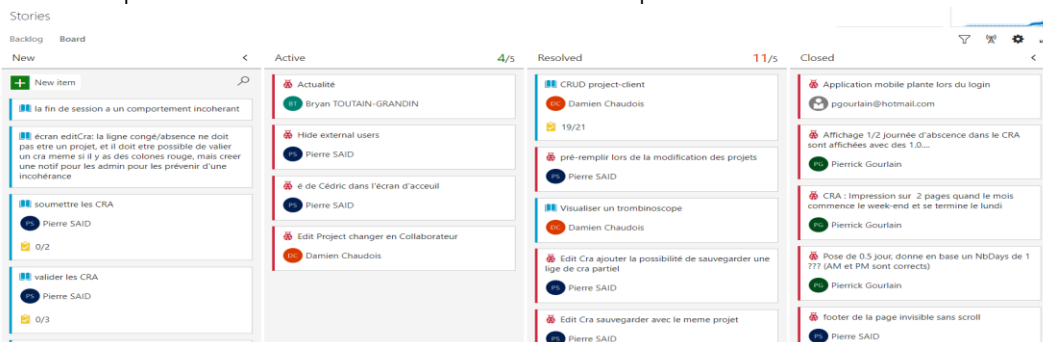


## 2.3 GESTION DE PROJET

Microsoft met à disposition son outil de gestion de projet VSTS (Visual Studio Team Services) qui est un couteau suisse de la gestion de projet. La plateforme web propose absolument tous les services nécessaires à la gestion, l'intégration, et le déploiement d'une application web.

### 2.3.1 Avancement du projet

Le développement de l'application fonctionne par la méthode AGILE, chaque développeur se voit attribué une user story, et il lui appartient par la suite de délimiter les tâches à effectuer pour la réaliser. VSTS offre une interface proche de celle de Trello :

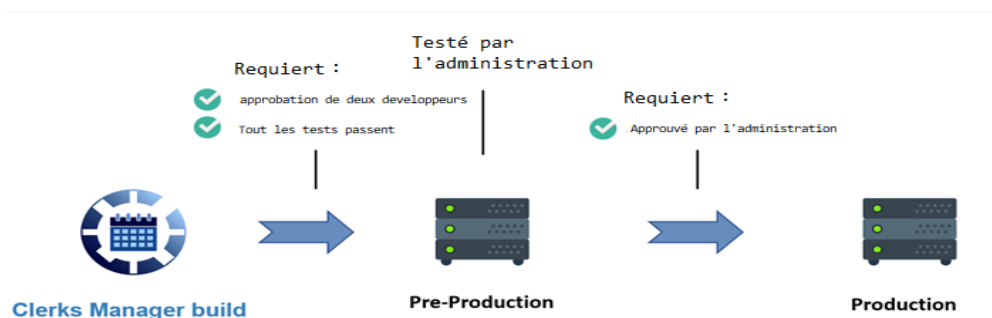


L'application étant développée en interne, il est très simple d'échanger avec le client puisqu'il se trouve dans le bureau d'à côté.

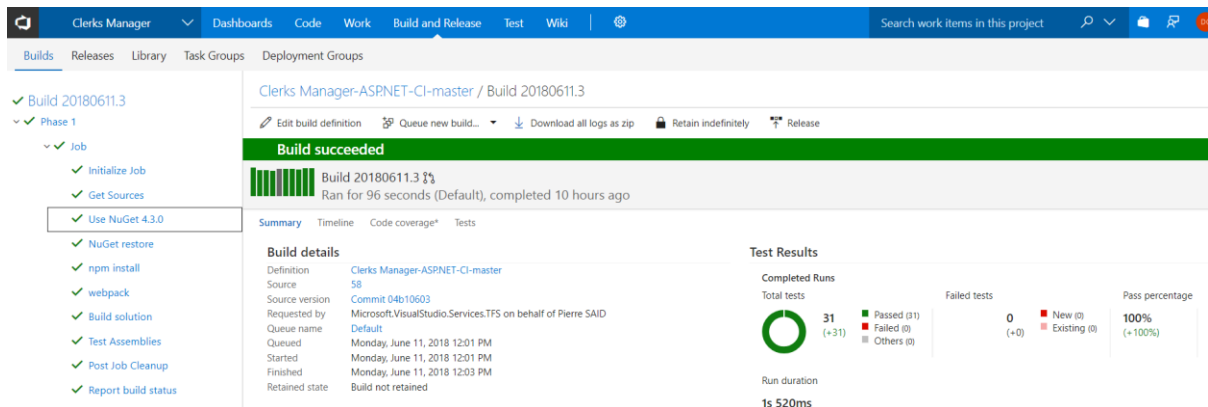
### 2.3.2 Intégration continue

Le versioning de l'application se fait sous Git et est totalement intégré dans VSTS : chaque commit déclenche un build, et tout une suite de JOB. Actuellement, il y a deux environnements dans lesquelles l'application peut être déployée : la pre-production et la production. Chaque déploiement est déclenché par un push sur la branche concernée. Il existe 3 branches :

- Master : impossible de commit sur cette branche, le développeur doit créer une nouvelle branche, développer sa fonctionnalité, puis faire une « pull request » pour merger ses modifications sur cette branche. Pour être accepté, un pull request doit valider un build, et être approuvé par au moins un autre développeur.
- Pre-production : seule la branche master peut faire un pull request sur cette branche. Une fois commit, le serveur de pre-production redémarre automatiquement avec la nouvelle version du build
- Production : seule la branche pre-production peut faire un pull request sur cette branche, et seul Pierrick (en charge du projet) ou Cédric (PDG d'Expaceo) peut valider cette action.

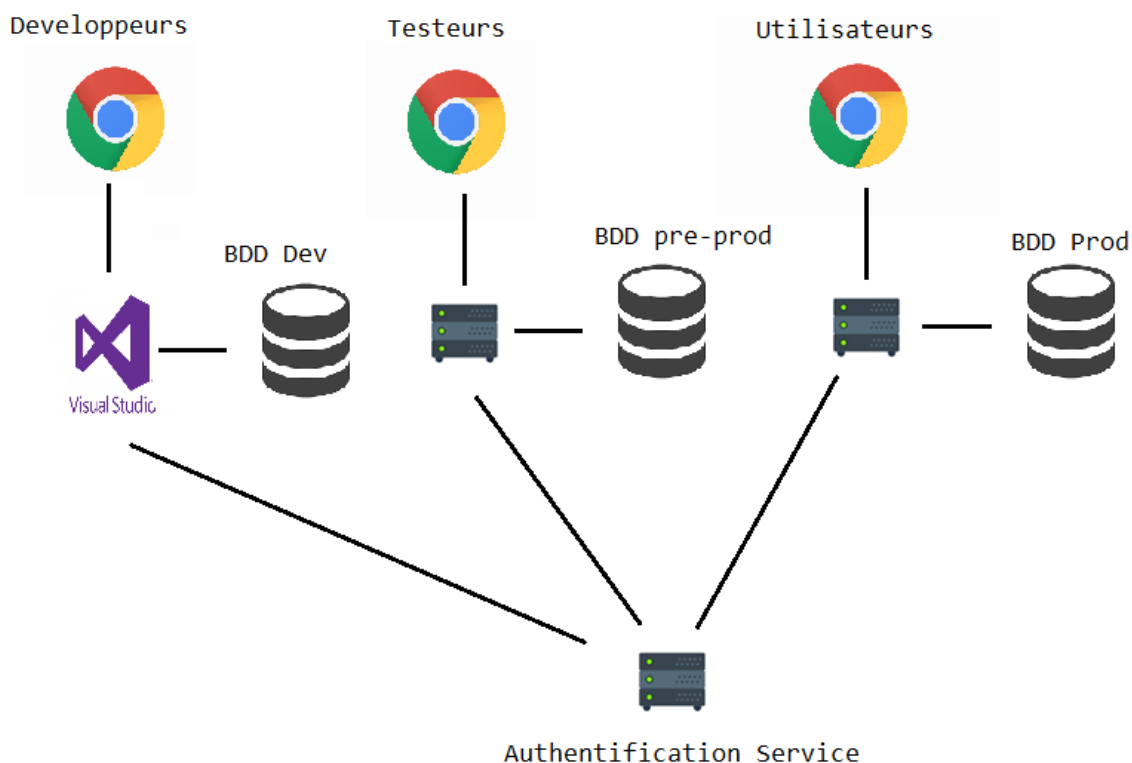


Dès qu'un merge est lancé, VSTS lance un build et affiche un rapport complet, les utilisateurs d'apache ne sont pas dépayés, le system est même plus poussé.



### 2.3.3 Architecture

En plus de la base de données standard, Clerks Manager repose sur un service d'authentification sur un serveur distant. Cela permet de limiter les risques d'intrusion en cas de corruption du serveur de production. Le déploiement se passe donc de la manière suivante :



Chaque développeur peut travailler sur sa branche locale exécutée par Visual Studio, mais travaillent tous sur la même base SQL, et chaque pull request, soit sur la branche pre-prod soit sur la branche production déclenche un redémarrage du serveur et une mise en production automatique.

# Conclusion

Déçue par mon année passée dans ma précédente entreprise lors de ma 3e année, j'ai décidé d'en changer pour ma 4e. L'expérience que je retiens d'Expaceo est incomparable que celle de mon ancienne mission, sur tous les niveaux. J'ai intégré l'équipe avec un niveau plus que médiocre en technologie web, et j'en ressors confiant dans mon poste de développeur full-stack. Tout au long de l'année, j'ai été suivie et formée par des professionnels qui m'ont encadré sans me tenir la main et m'ont toujours guidé vers le perfectionnisme de mon travail. Je peux dire sans hésiter que j'ai bien plus appris et progressé au sein de mon entreprise que dans l'école, et je suis très heureux des compétences que j'ai acquises chez eux cette année.

Grâce aux connaissances que m'ont transmises Expaceo, surtout au travers de Pierrick Gurlain, c'est tout un secteur du développement informatique qui s'est ouvert pour moi, et je leur en suis extrêmement reconnaissant pour cela.