



# **Rapport de stage**

## **Développeur web**

Augustin MARIE  
3 janvier - 18 février

Entreprise: ADTi  
Tuteur: David TORZ

# Remerciements

Je tiens tout d'abord à remercier **David TORZ** pour cette opportunité, son accueil et la confiance qu'il m'a accordé en me confiant la reprise d'un projet pour un client final.

Je remercie également le reste de l'équipe ADTi et plus particulièrement **Antoine Bonifacio** et **Teddy Bossuyt** pour leur accueil dans l'entreprise et leur aide pour m'intégrer dans l'équipe le temps du stage.

Enfin, je tiens à remercier **Sophie Larbalestrier** pour ces conseils et le temps investi dans la relecture de mon rapport.

<b>Remerciements</b>	<b>1</b>
<b>Introduction</b>	<b>3</b>
<b>Présentation de la mission</b>	<b>4</b>
A-Présentation de l'entreprise	4
B-Présentation de la mission	5
C-Problèmes rencontrés, contraintes et limites	7
a- Contrainte du travail à distance	7
b- Contrainte d'accès au serveur	7
c- Contrainte de gestion de version	8
<b>Organisation du travail</b>	<b>9</b>
A-Processus et méthodes suivies	9
a- Développement du logiciel	9
b- Installation des mises à jour	9
B-Organisation des journées	11
<b>Réalisation des missions</b>	<b>12</b>
A- Mise en fonctionnement de l'application web	12
a- apache et php	12
b- base de données	13
c- websocket	14
B- Intégrer une continuité de service	15
a- sauvegarde de l'état	15
b- affichage à l'utilisateur	15
C- Transformation du style	17
a- Utilisation de composants bootstrap	17
b- Transformation du css	18
D- Intégration d'un système de rôles	19
a- Modifications côté serveur	19
b- Modifications visuelles	20
E- Affichage de l'historique	21
a- Enregistrement de l'historique	21
b- affichage de l'historique	21
<b>Evaluation des compétences mobilisés</b>	<b>23</b>
A- Adéquation du travail	23
a- Rendu final	23
b- Améliorations possibles	24
B- Compétences mises en oeuvre	25
a- Compétences techniques	25
b- Autonomie	25
c- Communication	26
<b>Conclusion</b>	<b>27</b>

# Introduction

Dans le contexte de mon cursus de Bachelor en informatique, j'ai été amené à effectuer un stage de 7 semaines chez ADTI, une entreprise de création de logiciels informatiques. J'ai voulu intégrer cette entreprise, principalement connue pour ses logiciels de GMAO, afin de profiter d'une expérience sur la création de logiciel en entreprise.

A mesure que les entreprises gagnent en taille, il devient de plus en plus difficile pour elles de gérer le matériel d'une entreprise et d'avoir une vision globale de son parc de machines. L'utilisation d'un logiciel dédié pour être aidé dans ces tâches devient donc indispensable.

La mission que l'on m'a proposée a été la récupération d'une application de suivi de machines web faite en javascript et en php ayant été abandonnée par le précédent développeur celui-ci ayant quitté le projet de manière inattendue.

Ma principale mission était donc de voir ce que nous devions mettre en place pour faire fonctionner ce projet et l'améliorer autant que possible.

Dans ce rapport, j'expliquerai plus en détail le contexte du stage au travers d'une analyse de l'entreprise et de la mission. Puis j'évoquerai ces enjeux pour moi et pour l'entreprise. Je pourrai ensuite m'attarder plus en détail chaque mission m'ayant été confiée, c'est-à-dire, les enjeux, les problèmes rencontrés et les solutions trouvés pour chaque sous missions m'ont été confiés.

Pour finir, je conclurai en détaillant les compétences apprises lors du stage, les éléments qui ont été plus difficiles à appréhender et, à l'issue de ce rapport, je ferai le point sur mon ressenti personnel en ce qui concerne les conditions de travail et les expériences acquises avant d'expliquer quel type d'expériences je chercherai à acquérir par la suite.

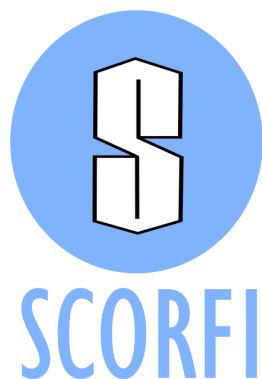
# I. Présentation de la mission

## A-Présentation de l'entreprise

La société ADTI est une petite société située à Wasquehal, une ville proche de Lille. Elle emploie beaucoup de stagiaires et d'alternants, et travaille aujourd'hui principalement en créant des applications web à l'aide de technologies variées avec, par exemple, du javascript et du php mais aussi avec des framework comme angular ou vue.js.

Les produits principaux de cette société sont ses logiciels de GMAO (Gestion de Maintenance Assistée par Ordinateur) ces logiciels permettent de gérer les tâches de maintenance d'une entreprise. Plus largement, ils permettent d'automatiser des opérations de gestion d'équipements (inventaire, recensement de caractéristiques), la maintenance (gestion des documents, demandes d'intervention), gestion des stocks, gestion des coûts et la gestion du personnel.

Les deux principaux logiciels de la société ADTI sont OPTIMa, leur logiciel de GMAO et SCORFI, logiciel de rapprochement et de communication bancaire.



Ces logiciels sont des applications natives traditionnelles qui ont été transformées en application web pour des raisons d'accessibilité (volonté de rendre ces applications disponibles sur des appareils comme des tablettes) mais aussi une simplicité de mise à jour (il suffit de mettre à jour l'application sur le serveur d'hébergement sans que les clients aient à télécharger quoi que ce soit et sans soucis de différence de version entre les clients). C'est la raison pour laquelle l'application sur laquelle mon stage s'est porté a été développée avec les technologies du web.

## B-Présentation de la mission

ADTI propose également la création de logiciels sur mesure et c'est dans ce contexte que mon stage s'est déroulé.

Une entreprise avec un grand nombre de machines à suivre avait demandé la création d'un logiciel permettant le suivi de celles-ci d'une manière simple et ergonomique. Le développement d'une première version du logiciel appelé followIt avait été faite cependant la personne responsable de son développement a quitté l'entreprise en ne laissant qu'une petite documentation utilisateur et le code de l'application. Ma mission était donc de récupérer cette application, paramétrer un serveur Apache et une base de données pour la faire fonctionner en me servant de la documentation utilisateur avant d'améliorer l'application existante.

Pour la partie graphique, l'application utilise html, css et javascript avec les librairies, ces langages de base constituent la majorité du code de l'application. Et permettent l'affichage des données.



Bien que l'application n'utilise pas de framework front-end, elle utilise en revanche de nombreuses librairies, les principales étant JQuery qui permet d'écrire du code javascript plus lisible et donc plus facilement maintenable. Bootstrap qui se qualifie sur son site comme une "boîte à outil" permettant la création de sites responsive et met disposition de nombreux éléments préconstruits. Et Datatables.js qui est utilisé pour automatiser la création d'un tableau avec des fonctions de bases (la recherche d'éléments, tri des éléments par rapport à une colonne, pagination automatique) qui sont des opérations difficiles à mettre en place par soi-même.

Du côté serveur, on utilise php pour permettre un affichage dynamique des pages mais aussi pour récupérer les informations sur les différentes machines. Ces informations sont mises à jour en utilisant une technologie de l'entreprise IBM. Cette partie est complètement détachée de l'application et est déjà mise en place en interne dans l'entreprise. Chaque machine communique son état dans une base de données appelée AS400 et le but de l'application est de faciliter la visualisation de ces informations (consultation des états, alertes en cas de panne, visualisation de l'historique des machines).

Une technologie appelée websocket permet au serveur de mettre à jour l'état des machines chez les clients sans que le client en face explicitement la demande. La librairie php pour mettre en place ce protocole est ratchet php websocket.



## C-Problèmes rencontrés, contraintes et limites

### a- Contrainte du travail à distance

Dans le cadre de cette mission, j'ai été confronté à plusieurs problèmes, le premier de ceux-ci était la quantité de travail à distance, cela rendait parfois la communication difficile. L'entreprise avait des petits locaux et préférait privilégier le travail à distance 2 à 4 jours par semaine. Pour communiquer à distance, l'entreprise utilise un logiciel nommé Mattermost, un service de messagerie instantanée libre. Celle-ci est auto-hébergée ce qui veut dire que le serveur sur lequel cette application est hébergée appartient à l'entreprise.

Un chat était à ma disposition pour communiquer mes avancements de la journée ou les problèmes que je rencontrais. On y communiquait également les bugs sur le serveur et les demandes de nouvelles fonctionnalités.

Cette organisation du travail a cependant quelques contraintes. Il peut par exemple arriver que quelqu'un mette du temps à voir un message sur une application et les explications peuvent parfois être moins claires que dans le cadre de travail en présentiel.

### b- Contrainte d'accès au serveur

Le second était l'absence d'accès au serveur du client. En effet, pendant la durée du stage, on ne pouvait accéder au serveur du client que quelques heures par jour. La raison était que l'on accédait au serveur à l'aide de team viewer et que je devais attendre qu'un responsable chez le client nous donne un accès manuellement pour pouvoir faire des mises à jour. De plus, l'accès à la base de données AS400 n'était disponible que sur ce serveur, ils pouvaient donc être compliqué de développer de nouvelles fonctionnalités utilisant cette base de données, puisque que je ne pouvais pas tester régulièrement mon code. En plus de cela, les différences entre l'architecture du serveur et la machine que l'on m'avait confiée provoquaient parfois des différences de fonctionnement qu'il fallait corriger.

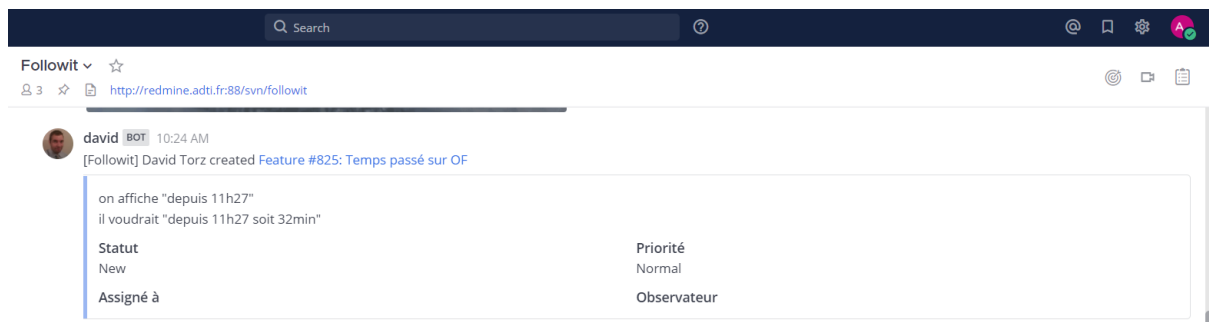
Une solution mise en place pour contourner ce problème était de simuler les données de l'AS400 au lieu de faire un vrai appel à la base de donnée, permettant ainsi de vérifier toute la partie "traitement des données" de l'application web et de trouver un maximum de bugs avant d'utiliser le serveur du client. Les deux solutions possibles pour simuler les données étaient de créer une base de données locale qui ressemble à celle de l'entreprise cliente et d'interroger temporairement celle-ci. Une autre solution utilisée était de créer un objet ressemblant au résultat de l'appel à la base de données et utiliser celui-ci sans faire appel à aucune base de données.



## c- Contrainte de gestion de version

Enfin, je n'ai pas eu accès à un logiciel de versionning pour la plus grande partie du stage, il était donc complexe de continuer l'intégration des fonctionnalités d'un côté et la correction des bugs sur le serveurs de l'autre puisque je devais gérer manuellement des versions en copiant le dossier sur lequel je travaillais. Plus tard, ce problème à été réglé quand on m'a donné un accès à l'application de gestion de projet utilisée par l'entreprise, redmine, qui a l'avantage d'être open-source. Celui-ci gérait le projet en hébergeant un dépôt svn (pour apache subversion). Contrairement à git, apache subversion repose sur un dépôt unique et centralisé.

A partir de ce moment-là, le partage des nouvelles versions est devenu beaucoup plus facile, et la demande de nouvelles features était facilitée grâce à un bot dans mattermost qui affichera automatiquement les demandes de nouvelles fonctionnalités dans le chat du projet.



J'ai cependant dû apprendre à utiliser redmine et à gérer des dépôts svn puisque ce sont des technologies auxquelles je ne suis pas habitué, j'ai donc eu besoin d'un temps d'adaptation pour correctement utiliser l'outil.

## II. Organisation du travail

### A-Processus et méthodes suivies

#### a- Développement du logiciel

Les missions portaient d'abord d'une demande de fonctionnalité de la part du client final, du maître de stage ou bien d'une de mes suggestions. Avant de développer, je devais décider de la meilleure façon d'implémenter la fonctionnalité, cette réflexion était généralement sur papier, pour visualiser les modifications à faire dans le code ou la base de données. Etant donné que je suis parti d'un logiciel existant, il était assez rare de devoir réfléchir à la nécessité d'implémenter de nouvelles librairies à ajouter.

Ensuite, je commençais l'intégration des fonctionnalités en précisant à la fin de mes journées sur le chat en ligne mon état d'avancement et mes potentiels blocages. Mes compétences me bloquent rarement étant donné que le logiciel utilise le php et le java script que je connais bien grâce à mes cours. Cependant, certaines librairies comme ratchet m'étaient inconnues donc je devais parfois me renseigner sur certaines fonctions avant de commencer.

A la fin du développement d'une fonctionnalité, je documentais mes fonctions et je partageais le fonctionnement de celles-ci avec des captures d'écrans sur le chat de l'application ce qui permettait de potentiellement voir les points à modifier si la fonctionnalité était correctement implémentée. Dans le cas où l'implémentation se déroulait sans problème, je pouvais avancer sur la fonctionnalité suivante.

#### b- Installation des mises à jour

Le serveur web sur lequel l'application devait être installée était interne à l'entreprise, ce qui veut dire que le serveur et l'application n'était pas disponible via une url. Pour accéder au serveur, on devait donc attendre d'avoir un accès sur une des machines de l'entreprise avec teamviewer et sur celle-ci accéder au serveur avec putty.

Putty permettait d'utiliser le protocole ssh pour faire des opérations comme changer les fichiers du serveur afin d'installer la dernière mise à jour mais aussi exécuter des commandes, comme redémarrer le serveur web. On pouvait ensuite observer le résultat de la mise à jour en se connectant à l'application web depuis la machine à laquelle on avait accès via teamviewer.

Il était parfois possible d'attendre quelque temps avant d'avoir accès à la machine et la résolution de bugs apparus pouvait temporairement arrêter le

développement des nouvelles fonctionnalités. Le dépôt en ligne a rendu plus facile la résolution des bugs et plus rapide le déploiement de la nouvelle version rendant le développement plus efficace et diminuant le temps d'arrêt du développement lors de l'apparition d'un bug.

## B-Organisation des journées

Les journées s'organisent de manière différente si la journée était en présentiel ou en distanciel. Cependant dans les deux cas, la semaine de 35 heures était divisée en 5 journées de travail de 7 heures, commençant à 9h00 et se terminant à 17h30, avec une pause de 1h30 à midi. Les matinées étaient dédiées au développement des nouvelles fonctionnalités et l'après midi était parfois plus chargé puisque c'était à ce moment-là que nous avions accès au serveur web du client, le développement des fonctionnalités pouvait être perturbé pour régler les bugs observés sur les machines du client.

Chaque mission partait d'une modification sur le programme à faire. En général, il y avait une trace écrite sur le logiciel mattermost (soit la modification à effectuer, soit un screenshot d'un bug par exemple), ce qui permettait de se rappeler précisément ce qu'il y avait à faire, ce qui était d'autant plus important à distanciel, car il était difficile de communiquer de manière fluide.

En distanciel, chaque matinée commençait par un appel de groupe sur mattermost, pour s'assurer que tout le monde sache ce qu'il avait à faire durant la journée. Cela permettait aussi d'avoir un contact entre collègues de l'entreprise, même sans être sur le site. A la fin de la journée, on demandait à chaque personne d'envoyer, sur le salon dédié à l'application sur laquelle il travaillait, un compte rendu du déroulé de la journée contenant les fonctionnalités créées, les endroits ayant posé problème et les points qui demandaient encore du travail.

## III. Réalisation des missions

### A- Mise en fonctionnement de l'application web

Comme expliqué, ma première mission a été de faire fonctionner la version du logiciel existant. J'avais pour ça à ma disposition une documentation utilisateur et une explication de l'utilisation des fonctions dans des fichiers au format markdown. Je devais également être capable de fournir une liste d'étapes car le serveur web du client était vierge.

#### a- apache et php

La première partie de l'installation consistait à initialiser le serveur apache et lui permettre d'exécuter des scripts php. J'ai choisi d'utiliser la dernière version stable de php, la version 8.1.3. Il est intéressant de prendre la dernière version stable pour s'assurer qu'elle reste stable aussi longtemps que possible. Pour pouvoir utiliser les bases de données, php a besoin de drivers, c'est ce qui permet à php d'exécuter des requêtes SQL dans une base de données. Pour notre application, on doit utiliser une base de données locale et la base de données as400 de l'entreprise, il a donc fallu installer ces deux drivers. J'ai eu quelques difficultés à installer le driver pour la base as400 puisque c'est une base de données propriétaire. En effet, elle est moins souvent utilisée, les ressources et l'aide sont donc plus difficiles à trouver et le driver est moins facilement accessible. J'ai cependant réussi à l'installer et j'ai pu communiquer les étapes pour que son installation soit plus facile sur le serveur web de l'entreprise.

Le serveur apache a demandé un peu de configuration notamment pour qu'il puisse savoir où se situe l'application et comment utiliser php. Ce sont des éléments de configuration de base qui n'ont pas posé beaucoup de problèmes.

Cependant la configuration initiale du serveur apache ne suffisait pas à faire fonctionner l'application, celle-ci possédant un fichier .htaccess qui sert à réécrire l'url des requêtes que le serveur web reçoit. Ici, il permettait la réécriture d'une url de forme "nom\_du\_site/ParcMachine" en "nom\_du\_site?page=ParcMachine", ce qui permet de cacher le fonctionnement interne de l'application web. Le reste de l'application avait besoin de cette réécriture pour fonctionner correctement et j'ai donc dû autoriser Apache à exécuter le contenu de ces fichiers.

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

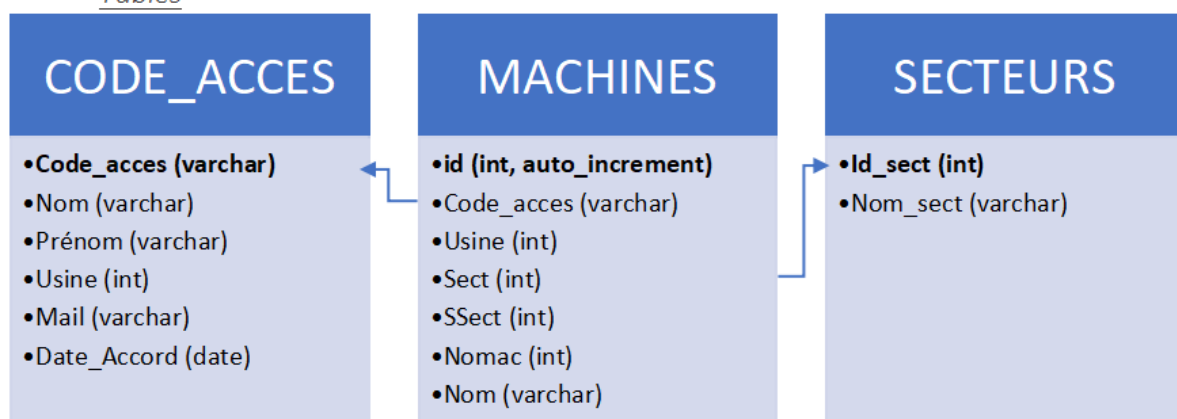
Pour que apache sache qu'il doit exécuter les fichiers .htaccess, il faut modifier l'option AllowOverride dans les fichiers de configuration du serveur Apache.

## b- base de données

La base de données locale n'était pas présente, j'ai donc dû la remonter en me servant de l'usage que les fonctions en font, mais aussi en utilisant la documentation existante. Le schéma de la base de données de départ était le suivant:

### Modèle de données

#### Tables



La table CODE\_ACCES recense les différents codes d'accès distribués aux utilisateurs.

La table MACHINES recense les machines créées dans les parcs machines des utilisateurs.

La table SECTEURS est une table de paramètre qui recense les différents secteurs.

Certains champs étaient soumis à des contraintes, l'une d'elles était "usine ne peut prendre que les valeurs 2, 3, 4 ou 7". J'ai d'abord pensé utiliser le type "enum", qui permet de proposer plusieurs valeurs qu'un champ peut prendre. Le problème est que la valeur d'un enum est stockée comme une chaîne de caractères en base de données, alors que le programme s'attend à le trouver sous forme de nombre entier. J'ai donc dû chercher comment imposer un certain nombre de valeurs prédéfinies, et j'ai fini par utiliser la commande Check. Celle-ci permet de vérifier qu'un champ respecte certaines conditions et remonte une erreur si ce n'est pas le cas.

```
--
-- Contraintes pour la table `usine`
--
ALTER TABLE `usine`
  ADD CONSTRAINT `chk_usine` CHECK (`num_usine`=2 OR `num_usine`=3 OR `num_usine`=4 OR `num_usine`=7);
COMMIT;
```

Une fois la base de données montée, j'ai créé un script pour qu'il puisse être directement exécuté sur le serveur web du client final, celui-ci permet de recréer la base de données à partir d'une seule commande ce qui a permis de gagner du temps par la suite.

### c- websocket

Le websocket est ce qui permet de mettre à jour les informations sur les machines connectées à l'application en temps réel. Contrairement au reste de l'application, le serveur websocket doit être lancé à part pour pouvoir fonctionner. On peut le lancer manuellement grâce à la commande php WebSocketServer.php et, ne nécessite pas de configuration particulière, cependant, ce n'est pas une bonne façon de le démarrer dans l'environnement de production, car si le websocket crash ou que le serveur web redémarre, le serveur websocket ne sera pas relancé automatiquement. Heureusement, linux possède une commande pour faire exactement ça, cela n'a donc pas posé problème.

Le deuxième problème rencontré a été la nécessité pour le websocket d'avoir accès à la base de données AS400. Je ne pouvais au départ rien tester en rapport avec la websocket car celle-ci crashait à chaque fois qu'elle n'arrivait pas à se connecter à la base de données. La solution que j'ai trouvée à ce problème a été de mettre les requêtes à la base de données AS400 dans un bloc try catch.

```
try{
    foreach ($this->clients->getByUsine($usine) as $client) {
        $client->send(json_encode(array("type"=>"EVENTS", "data"=>Event::getAllByUsine($usine))));
    }
    break;
}
catch (PDOException $exc){
    echo "INSERT : Impossible de se connecter à l'AS400\n";
}
```

Si le programme rencontre une erreur dans ce bloc, au lieu de crash, il va exécuter le code dans le bloc catch avant de continuer, ce qui permet de faire remonter l'information qu'il y a eu une erreur et de continuer d'utiliser la websocket dans l'environnement de test sans être obligé de la redémarrer à chaque requête ayant échoué.

Le dernier problème rencontré lors du paramétrage du websocket a été lors de la première installation sur le serveur du client. En effet, par défaut, le websocket écoutait le port 8080 qui était bloqué par le réseau de l'entreprise, j'ai donc rendu le port d'écoute du websocket configurable dans un fichier de configuration pour pouvoir éviter le problème, et permettre à l'entreprise de changer facilement le port d'écoute par la suite si nécessaire.

## B- Intégrer une continuité de service

Après avoir fini ma première mission, il m'a été demandé d'assurer un fonctionnement minimal de l'application en cas de panne du websocket, l'application devant alors pouvoir afficher à l'utilisateur le temps depuis la dernière mise à jour et les dernières données enregistrées.

### a- sauvegarde de l'état

La première partie était donc la sauvegarde régulière des informations du websocket. Pour ce faire, j'ai décidé de déclencher une sauvegarde chaque fois que les données de l'AS400 changent. C'était une opération plus facile qu'il n'y paraissait car la détection du changement des données était déjà en place dans l'application de base. En effet pour ne pas envoyer trop de requêtes d'actualisation aux clients, une fonction était déjà en place pour envoyer des données uniquement en cas de changement, je n'avais donc plus qu'à sauvegarder dans la base de données locale les nouvelles informations.

J'ai ensuite créé une nouvelle page en php répondant à des requêtes similaires à celles du websocket, mais utilisant la base de données locale. En cas de panne du websocket, un client affichant l'application pourra recevoir les données sauvegardées sur la base de données locale et pourra donc continuer d'utiliser l'application normalement, mais sans avoir accès à la mise à jour des informations. Elle retourne au client un format de données qui est lui aussi similaire à celui que retourne le websocket, ce qui permet d'utiliser des fonctions identiques dans les deux cas. De plus, comme cette page est juste un petit script appelé à chaque requête, il ne peut pas être injoignable, ce qui permet d'être sûr qu'elle soit toujours accessible.

### b- affichage à l'utilisateur

A ce moment-là, la question posée est de savoir comment détecter l'absence de websocket du côté du client affichant l'application pour qu'il puisse rediriger les requêtes, soit vers le websocket, soit vers le script php qui prend le relais en cas de panne. Pour ce faire, j'ai utilisé un paramètre de la connexion avec le websocket qui s'appelle le ready state.

Value		
Value	State	Description
0	CONNECTING	Socket has been created. The connection is not yet open.
1	OPEN	The connection is open and ready to communicate.
2	CLOSING	The connection is in the process of closing.
3	CLOSED	The connection is closed or couldn't be opened.



Comme on peut le voir, si l'état de la connexion entre le client affichant l'application web et le websocket server est différent de 1, cela signifie que le websocket n'est pas accessible, je peux donc à ce moment diriger les requêtes au bon endroit.

```
/**
 * Fonction de redirection des requêtes en cas de panne de la websocket
 * @param {Object} data données à envoyer
 */
function directRequest(data){
  if (websocket_server.readyState != 1){
    noWs_sendMessage(data);
  }
  else {
    websocket_server.send(JSON.stringify(data));
  }
}
```

Sur cette capture d'écran, on peut observer la comparaison de la propriété `readyState` par rapport à 1 pour permettre de rediriger les requêtes (ici appelées `data`) vers la bonne fonction. Comme les deux fonctions s'attendent à recevoir des données similaires, il n'y a que peu de changements à effectuer.

Après cela, la réponse à la requête est traitée pour afficher à l'utilisateur les informations nécessaires. Une dernière chose qui a été rajoutée est un bandeau affichant le temps depuis la dernière mise à jour des données. Cette valeur est stockée dans un petit dossier au format json car je ne voulais pas créer une table dans la base de données pour une seule valeur. Celle-ci est mise à jour régulièrement et récupérée en cas de problème.

la connexion au serveur a été perdue, veuillez recharger la page. Si le problème persiste, redémarrez l'application  
Dernière maj il y a 1h

Cette capture d'écran montre le bandeau affiché lorsque l'application n'arrive pas à se connecter avec le websocket. On affiche une durée au format "Dernière maj il y a *jj* jours et *hh* h", mais n'affiche le nombre de jour que si celui ci est supérieur à 0.

## C- Transformation du style

### a- Utilisation de composants bootstrap

Une fois la continuité de service mise en place, et quelques corrections sur l'ancienne version du logiciel effectuées, La question de l'affichage des informations s'est posée. L'application d'origine affichait les informations des machines de la manière suivante:



Sur cette capture d'écran, on voit l'affichage de données de test sur la page principale. Cet affichage posait problème pour plusieurs raisons. Premièrement, l'affichage n'était pas esthétiquement plaisant, et deuxièmement, il n'y avait aucune limite pour l'affichage de la taille du titre, ce qui posait problème, car l'utilisateur final voulait afficher l'application sur des machines n'ayant pas forcément accès à une souris et donc l'affichage peu compact pouvait remplir l'écran avec peu de machines.

Refaire le style de cette page avait 2 buts, afficher plus d'informations de manière plus esthétique.

Pour refaire le style du site j'ai décidé d'utiliser les composants de bootstrap. Pour rappel, Bootstrap est une boîte à outils" mettant à disposition de nombreux éléments préconstruits et pré-stylisés. L'utilisation de cette bibliothèque permet de gagner du temps et d'avoir un meilleur style. De plus, cette bibliothèque était déjà installée dans le projet de base, je n'avais donc aucune raison de ne pas m'en servir. Je pouvais ainsi créer rapidement différents éléments, et rajouter un minimum de style pour leur donner l'apparence désirée.

## b- Transformation du css

Pour apporter plus de visibilité sur l'opération en cours, j'ai choisi d'écrire le nom complet de l'opération dans la pastille de couleur en haut à droite, un élément de bootstrap qui s'appelle "badge". Les titres ne peuvent s'afficher que sur la ligne du haut pour économiser de la place. Pour économiser la place j'ai aussi choisi de réduire la police d'écriture, étant donné que les écrans sur lesquels sera affichée l'application sont de grande taille. Grâce à bootstrap si quelqu'un essaye d'afficher l'application sur portable, les secteurs (représentés par les grands conteneurs avec un titre orange) seront affichés sur une ligne pour préserver une certaine visibilité. Finalement, j'ai choisi de mettre les cadres correspondants à des machines en panne en jaune et de faire jouer un son, pour insister sur les erreurs et les rendre plus faciles à voir.

Gestion du parc machine

Follow It !

Deconnexion

TEST Suivi des machines Parc machine Suivi des connexions Liste des utilisateurs Gestion des secteurs

Le serveur est pour le moment inaccessible  
Dernière maj : il y a 1 jours et 20 h

1. Impression

FEG KBA 145

Roulage

(7-1-4-1) OF : 9949-1 depuis 14h45 soit 71h19min

R 900

Roulage

(7-1-4-4) OF : 13016-10 depuis 16h10 soit 69h54min

3. Contre collage

Contrecolleuse experflute

Roulage

(7-3-1-2) OF : 54123-3 depuis 8h05 soit 77h59min

5. Découpe

142 CER

Panne

(7-5-1-3) OF : 8456-2 depuis 6h33 soit 79h31min

162 CER

Casse-croûte

(7-5-1-4) OF : 16685-2 depuis 11h39 soit 74h25min

145 PER

à l'arrêt

(7-5-1-6)

8. Assemblage

Plieuse X115

Roulage

(7-8-2-6) OF : 34525-1 depuis 6h58 soit 79h06min

Sur cette capture d'écran, on peut voir le nouvel affichage des différentes informations. Si il n'y a pas d'action sur une machine, la case correspondant à la machine écrit juste "à l'arrêt". On peut aussi remarquer une barre de navigation que j'expliquerai plus tard.

Ce que l'on ne voit pas ici en revanche c'est que l'application ne sait pas à l'avance combien de secteurs elle aura à afficher, il faut donc automatiquement trouver le meilleur arrangement des secteurs. J'ai décidé d'utiliser une librairie nommée masonry. Celle-ci permet d'automatiquement trouver le moyen le plus efficace pour ranger sur plusieurs colonnes différents objets, ce qui m'a permis de ne pas avoir à créer cette fonctionnalité complexe.

## D- Intégration d'un système de rôles

Une des dernières missions qui m'a été confiée a été l'intégration d'un système de rôle. Après avoir ajouté un champ "rôle" dans la table des utilisateurs de la base de données, j'ai dû gérer la façon dont le programme gère le rôle du côté serveur, mais aussi du côté client. J'ai également suggéré l'intégration d'un mot de passe dans le fonctionnement de l'application, ce qui a été considéré comme non-nécessaire dans l'immédiat par le maître de stage.

### a- Modifications côté serveur

Quatres rôles ont été rajoutés : lecteur, utilisateur, admin usine et super admin. Les lecteurs peuvent uniquement accéder à la page pour visualiser les machines de leur usine. Les utilisateurs peuvent le faire aussi, mais eux peuvent gérer les machines suivies par l'application. Les admin usine peuvent, en plus des autorisations utilisateur, gérer les utilisateurs de leur usine ( ajouter, supprimer ou modifier les données les concernant) et enfin le super admin peut voir les machines et gérer les utilisateurs de toutes les usines.

Il a donc fallu développer un moyen aussi simple que possible de trouver si un utilisateur peut modifier un élément. J'ai donc créé des fonctions de test (une pour les machines et une pour les utilisateurs), appelées à chaque requête de la part d'un client.

```
public function canBeModifyBy($session){
    //Vérifie que le code d'accès est correct
    $ac = Acces::getById($session);
    if($ac){
        switch ($ac->getRole()){
            case "super admin":
                //le super admin a tout les droits
                return true;

            case "admin usine":
            case "utilisateur":
                //Les admins d'usine et les utilisateurs peuvent modifier que les machines de leur usine
                if($ac->getUsine()!=$this->getUsine()){
                    return false;
                }
                return true;

            default:
                //Si un utilisateur est ni admin ni super admin ni utilisateur, alors il ne peut rien faire
                return false;
        }
    } else {
        return false;
    }
}
```

Sur cette capture d'écran, on voit la fonction qui vérifie que la machine sur laquelle la fonction est appelée peut être modifiée par l'utilisateur avec le code d'accès (\$session) donné en paramètre. if(\$ac) sert à vérifier que le code d'accès donné en paramètre est valide. On vérifie ensuite le rôle correspondant à cet utilisateur. Si c'est un super admin, il a le droit de modifier n'importe quelle

machine, donc on ne fait pas d'autres vérifications (case "super admin"). Si c'est un admin usine ou un utilisateur, il faut vérifier que la machine est bien dans la même usine que la leur, ce que je fait avec `$ac->getUsine()!=$this->getUsine()`, `$this` étant la machine sur laquelle la fonction a été appelée. Enfin, le cas "default" veut dire "si ce n'est aucun de ces cas là" et si un utilisateur n'a pas un de ces trois rôles alors il ne peut modifier aucune machine.

Appeler cette fonction à chaque fois qu'un client demande l'ajout, la suppression ou la modification d'une machine permet d'être sûr que les utilisateurs ne fassent rien qu'ils n'aient pas le droit de faire. Mais le mieux reste de ne pas leur donner la possibilité de tenter une action qu'ils ne peuvent pas faire, c'est ce que l'on va voir lors de la section suivante.

## b- Modifications visuelles

La première modification visuelle a été de faire apparaître une barre de navigation pour les administrateurs.



Celle-ci leur permet de naviguer vers des pages comme la liste des utilisateurs connectés à l'application, ou la liste des utilisateurs. Ces pages ne sont destinées qu'aux administrateurs. Les utilisateurs de base ne peuvent pas accéder à ces pages, ils naviguent en utilisant les liens en haut de la page. En plus de ne pas voir les pages auxquelles ils n'ont pas accès, les utilisateurs ne peuvent pas non plus changer leur page en tapant dans l'url grâce au code suivant:

```
if($_SESSION["role"]!="super admin" && $_SESSION["role"]!="admin usine"){  
    error401();  
    exit;  
}
```

Sur cette capture d'écran, on peut voir que le programme appelle la fonction `error401` si le rôle de l'utilisateur ne correspond pas à un type d'administrateur. Cette fonction redirige vers la page d'erreur 401 (pour non autorisé) sans laisser l'utilisateur voir la moindre donnée.

Ensuite du côté du serveur, on vérifie quelles machines on doit montrer à quelles personnes. Pour ce faire, on utilise une logique similaire à celle pour vérifier si un utilisateur a le droit de modifier une machine. On vérifie quel est le rôle et l'usine d'un utilisateur, et on sélectionne les machines à afficher en conséquence.

## E- Affichage de l'historique

Quelle application de suivi de machine serait complète sans une visualisation de l'évolution dans le temps ? C'est pourquoi ma dernière mission a été d'enregistrer les états des machines dans une nouvelle table "Historique".

### a- Enregistrement de l'historique

Pour enregistrer dans la base de données locale les données concernant l'évolution de l'état des machines dans le temps, j'ai eu besoin de créer une table "Historique". Celle-ci contient les mêmes champs que ceux présents dans les événements récupérés dans l'AS400 avec en plus un champ "durée".

Mon idée de base était d'enregistrer les événements récupérés d'une manière similaire à l'enregistrement des derniers événements. Je me suis rendu compte que je ne pouvais pas le faire aussi simplement, notamment à cause du fait que j'aurais trop rapidement une base avec beaucoup d'entrées. J'ai donc décidé de comparer les événements en cours avec les derniers événements en cours dans l'historique.

Cette solution était un peu complexe à mettre en place notamment à cause du fait qu'il n'y avait pas de moyen direct d'associer un événement d'historique avec une machine. En effet, les événements de l'AS400 arrivent sans référence à la machine qu'ils représentent et chaque machine de l'AS400 n'est pas nécessairement suivie par l'application. Pour retrouver la machine, il faut se servir du combo usine, secteur, sous secteur et numéro de machine d'un événement pour vérifier si la machine est suivie par l'application.

De plus, comme les enregistrements d'historique sont, eux, liés à une machine dans la base de données, si un utilisateur veut supprimer une machine, il faut penser à supprimer ces enregistrements d'historique à ce moment-là sous peine d'avoir une erreur "violation de la contrainte de clé étrangère". J'avais réglé ce problème en exécutant une requête supprimant les enregistrements d'historique d'une machine avant de supprimer la machine elle-même. Aujourd'hui, avec mes nouveaux cours de base de données, j'aurais probablement utilisé la propriété "ON DELETE CASCADE" dans la base de données. Celle-ci permet d'avoir le même effet mais d'une manière automatique et plus sûre.

### b- affichage de l'historique

Je voulais que l'historique par machine soit accessible depuis la page de suivi des machines, mais je ne voulais pas juste ajouter un bouton à cliquer car les cadres affichant les machines, qui étaient déjà compacts, ne pouvaient pas afficher un bouton supplémentaire de manière ergonomique.

J'ai donc décidé d'afficher l'historique d'une machine quand on clique sur celle-ci dans l'interface. Pour montrer que l'interaction est possible, j'ai fait en sorte que la carte de la machine dans l'interface paraisse plus sombre quand on passe la souris dessus. Quand on affiche l'historique, voilà l'écran sur lequel on arrive.

The screenshot shows a web application interface for machine management. A modal window titled "Suivis de la machine Contrecolleuse experflute (7-3-1-2)" is open, displaying a table of machine operations. The background interface includes a top navigation bar with "Gestion du parc machine" and "Deconnexion", a left sidebar with "TEST", "Suivi des machines", "Parc machine", and "Suivi des connexions", and a main content area with sections for "1. Impression" and "8. Assemblage".

Opération	OF	Séquence	Durée	Jour debut	Heure debut
Roulage	54123	3	en cours	15/02/2022	8:05
Panne	13452	1	2:28	11/02/2022	12:15
Roulage	25355	1	0:55	11/02/2022	12:12
Casse-croûte	54123	4	0:15	11/02/2022	11:32
Roulage	54123	3	1:30	11/02/2022	10:32

The modal window also includes a close button (X) in the top right, a pagination control showing "Afficher 10 éléments", and navigation buttons "Précédent", "1", and "Suiv".

On voit donc sur cette capture d'écran l'affichage des différents états au cours du temps avec leur durée. A noter que pour ne pas surcharger l'affichage, on n'enregistre pas quand une machine est éteinte, seulement quand une opération est en cours.

Ici, la librairie Datatables.js permet de mettre automatiquement une fonction de pagination pour ne pas être submergé par un immense tableau, dans le cas où une machine aurait une centaine d'enregistrements par exemple. Cliquer sur la croix ou sur la zone grise permet de fermer cette fenêtre et de revenir à l'affichage normal des états des machines.

## IV. Evaluation des compétences mobilisés

### A- Adéquation du travail

#### a- Rendu final

Le rendu final se présente sous la forme d'un dépôt svn sur redmine (qui est leur logiciel de versionning). Ceci simplifie les ajouts de mises à jour et les corrections de bugs. De plus, à la fin du stage, tous les bugs majeurs de l'application avaient été réglés, il n'y avait donc plus de travail de développement à effectuer de la part de l'entreprise ADTi.

Chaque fonction est également documentée dans un fichier au format markdown ce qui permet de voir plus facilement et de façon plus détaillée ce que fait chaque fonction.

#### `getSecteurs()`

Cette méthode renvoie les informations sur le secteur de la machine sous la forme d'un objet (`stdClass`) avec deux attributs (`id_secteur` et `nom_secteur`) ou *False* si aucun secteur correspondant n'est trouvé. Elle fait un appel à la base de données locale avec la requête SQL suivante :

#### `modify($nom,$prenom,$role,$mail)`

Cette méthode modifie dans la base de données les colonnes *nom*, *prenom*, *role*, *mail* pour l'**Acces** sur lequel cette fonction a été appelée en les remplaçant par les valeurs données en paramètre. Ces quatre propriétés sont les seules qui peuvent être changées chez un **Acces**.

#### `insert()`

Cette méthode insère dans la base de données l'objet **Acces** sur lequel cette fonction a été appelée.

Sur cette capture d'écran, on peut voir un exemple de documentation au format markdown. L'avantage d'utiliser ce format pour la documentation est que le navigateur ou autre logiciel utilisé pour le visualiser peut générer automatiquement un menu de navigation pour trouver rapidement ce que l'on cherche.

En plus du code, j'ai aussi rendu un script pour créer la base de données, dans le cas où l'on voudrait refaire des modifications majeures sur l'application.



## b- Améliorations possibles

Cependant le code final est améliorable. En effet, certaines fonctionnalités étaient codées de manière sous-optimale. J'aurai voulu améliorer certaines fonctionnalités existantes ou modifier certaines fonctions que j'ai moi-même créées mais j'ai pu me rendre compte que, souvent, il valait mieux prendre plus de temps pour réfléchir à la meilleure intégration possible plutôt que de se dire que l'on pourra améliorer plus tard, car dans la pratique on a rarement le temps de le faire.

Une autre fonctionnalité que j'aurais aimé améliorer est l'utilisation de la base de données. En effet, pour le moment, pour supprimer une machine de la base de données, on fait d'abord une requête pour supprimer son historique et son état enregistré avant de supprimer la machine elle-même. Ceci pose problème car on pourrait avoir des situations où une partie des suppressions serait correctement effectuée et d'autres échoueraient. Or j'ai découvert récemment une façon de faire ce type de requêtes en même temps et d'ensuite effectuer toutes les suppressions en une seule fois uniquement si toutes les requêtes ont un résultat positif. Cela renforcerait la sécurité et la fiabilité de ces opérations.

Enfin, bien que l'on m'ait dit que la sécurité de l'application n'était pas une priorité j'aurais aimé sécuriser l'application un peu plus. Car même si l'application est destinée à n'être utilisée qu'en interne, on est jamais à l'abri d'un utilisateur malveillant. En effet, une grande partie des cyber-attaques informatiques contre les entreprises sont initiées par quelqu'un d'interne dans l'entreprise. Et même si l'application ne permet qu'un suivi d'état des machines, il est important de maintenir la sécurité des comptes. J'aurais donc aimé ajouter un mot de passe pour les comptes, passer l'application en https et ajouter un système de log pour savoir quel compte fait quelles requêtes pour faciliter l'identification d'activités suspectes.

## B- Compétences mises en oeuvre

Pendant ces 7 semaines, j'ai pu développer certaines compétences valorisables dans un environnement professionnel. Je ferai dans cette partie le point sur les plus importantes d'entre elles.

### a- Compétences techniques

Dans ce stage, j'ai travaillé avec les mêmes technologies que lors de mon premier stage. J'ai pu constater mon évolution, notamment sur java script, où il m'a été beaucoup moins difficile de travailler avec des requêtes AJAX. J'ai aussi eu beaucoup moins de difficultés à travailler avec du php orienté objet ce qui m'a permis de me sentir plus efficace.

Travailler en partant du code de quelqu'un d'autre m'a aussi permis de voir un certain nombre de bonnes pratiques que je n'aurais pas mises en place par moi-même. Notamment mettre des points virgules à la fin de chaque ligne en javascript même si il ne sont pas obligatoires. Le précédent développeur écrivait certains noms de variables en anglais et d'autres en français ce qui est une mauvaise pratique. En effet, il est généralement conseillé d'utiliser des noms de variable en anglais ou éventuellement en français, mais en aucun cas de mixer les deux.

Une compétence technique à revoir est cependant la conception des bases de données. J'ai senti, à certains moments, que je n'utilisais pas ce langage de manière optimale. Je pense que la base de données est la chose la plus importante à travailler pour moi par la suite.

### b- Autonomie

En terme d'autonomie, j'ai pu me servir de mon expérience de l'année dernière et celle des projets scolaires. J'ai su prendre en main ce projet seul et répondre aux exigences. Cependant c'était la première fois que je devais réaliser un projet seul de bout en bout (front end, back end et base de données), ce qui m'a permis de continuer à travailler cette compétence.

Néanmoins, j'ai pu me rendre compte que autonomie ne veut pas dire coder dans sa bulle, et parfois, j'aurais apprécié profiter d'une revue de code, une bonne pratique consistant à faire relire son code par quelqu'un d'autre. J'ai le sentiment que cela m'aurait aidé à éviter des bugs facilement trouvables, ainsi que d'améliorer certaines parties de mon code un peu complexe, en profitant de l'expérience de mon relecteur. Cette pratique permet aussi de s'assurer que le code est compréhensible pour quelqu'un d'extérieur ce qui est une garantie supplémentaire quant à la maintenabilité à long terme de celui-ci. J'espère donc que dans mes expériences professionnelles à venir je pourrai profiter de ce type de suivi.

## c- Communication

Ce stage m'a également permis de me rendre compte que, même si ma communication orale est bonne, le travail à distance rend les choses beaucoup plus complexes. J'ai pu travailler la communication à l'écrit, expliquer les problèmes et les avancées de manière concise et rapidement compréhensible.

Cette communication n'est pas que pratique car il est aussi très important de maintenir la cohésion d'équipe et de favoriser une ambiance positive avec ses collègues, même à distance. Sans cela, il a été prouvé que les conséquences d'une mauvaise ambiance peuvent causer une baisse de productivité. Cet aspect de la communication est très important surtout si ma carrière s'oriente vers des positions de chef d'équipe / chef de projet.

C'est une compétence que j'ai eu du mal à acquérir, comme j'en ai parlé dans la partie "I.C- Contrainte du travail à distance", mais cette compétence sera valorisable dans le futur, car la tendance est à l'augmentation du travail à distance pour des raisons sanitaires, pratiques mais aussi de confort pour les employés. Il est donc probable que je sois confronté régulièrement à ce type de travail dans ma vie professionnelle.

# Conclusion

Les conditions de travail dans l'entreprise ont été souvent difficiles à cause de la difficulté à communiquer, et cela particulièrement au début du stage.

De plus, le travail à distance régulier, et le fait de ne pas avoir accès à l'environnement sur lequel l'application sera déployée, ont créé des conditions tout à fait uniques pour moi. Ces conditions particulières ont vraiment ralenti mon avancée, surtout au début du stage. J'ai cependant réussi à m'adapter à ces conditions particulières de manière satisfaisante, puisque le rendu final a été jugé satisfaisant par l'entreprise cliente et le tuteur de stage.

Aussi, bien que l'autonomie totale sur le projet pouvait parfois être difficile à assumer, j'ai pu me servir de mes compétences sur le langage utilisé pour mener à bien la mission.

La mission qui m'a été confiée a donc été remplie avec succès. L'entreprise cliente possède maintenant une interface web pour suivre en temps réel l'état de ces nombreuses machines qu'elle faisait auparavant avec un tableur excel. Et pour l'entreprise ADTi, le contrat avec l'entreprise a été rempli, fidélisant et satisfaisant un client qui utilisait déjà le logiciel optima.

Ayant gardé contact avec David TORZ, le dirigeant, celui-ci m'a confié qu'une semaine après mon départ aucun bug majeur n'avait été trouvé sur le logiciel.

Ce stage m'a donc permis d'améliorer mes compétences sur des langages que je connaissais, tout en me faisant me rendre compte des progrès qu'il me restait à faire en matière de base de données et en communication à distance.

Bien que j'aime le développement web, après deux stages dans ce domaine, j'aimerais découvrir d'autres domaines de développement pour ne pas déjà me fermer. Le développement d'applications dites natives, plus traditionnelles, est une de ces autres matières que je souhaiterais explorer. Pour mon alternance, ou dans le cadre d'un hypothétique nouveau stage, j'aimerais donc utiliser d'autres langages que le java script ou le php, pourquoi pas quelque chose de plus bas niveau comme le C, C# ou rust que j'ai commencé à apprendre. Une expérience professionnelle plus centrée sur la création, gestion et protection d'une base de données me serait également très profitable et me permettrait de faire, de manière confiante, une application web de A à Z de la base de données jusqu'au front-end.

Finalement, j'aimerais que ma prochaine expérience professionnelle se fasse avec une équipe de professionnels, car lors de mon premier stage, j'ai travaillé avec une équipe de stagiaires, et dans le second, j'étais seul sur un projet. J'aimerais donc travailler avec une équipe de professionnels plus expérimentés, afin de voir les avantages que cela apporte, et si ce genre de conditions de travail me conviendrait ou pas, afin de préciser mon projet professionnel.