

RAPPORT DE STAGE

(6 janvier – 14 février)

**CENTRE D'ACTION SOCIALE
DE LA VILLE DE PARIS**



→ Département des études de projet numérique

ERWAN

THOURY

Sommaire

I/ Présentation de l'entreprise → pages 3 et 4

- a) Activités, organisation → page 3
- b) Organigramme → page 3
- c) Quelques chiffres → page 4

II/ Projet → pages 4 à 13

- a) Etude de l'existant → page 4
- b) Objectifs → page 4
- c) Moyens utilisés → page 5
- d) Méthodologie de travail → pages 5 et 6
- e) Analyse → pages 6 à 9
- f) Schémas → pages 9 à 12
- h) Difficultés rencontrées → page 13

III/ Conclusion → page 13

I/ Présentation de l'entreprise

a) Activités, organisation

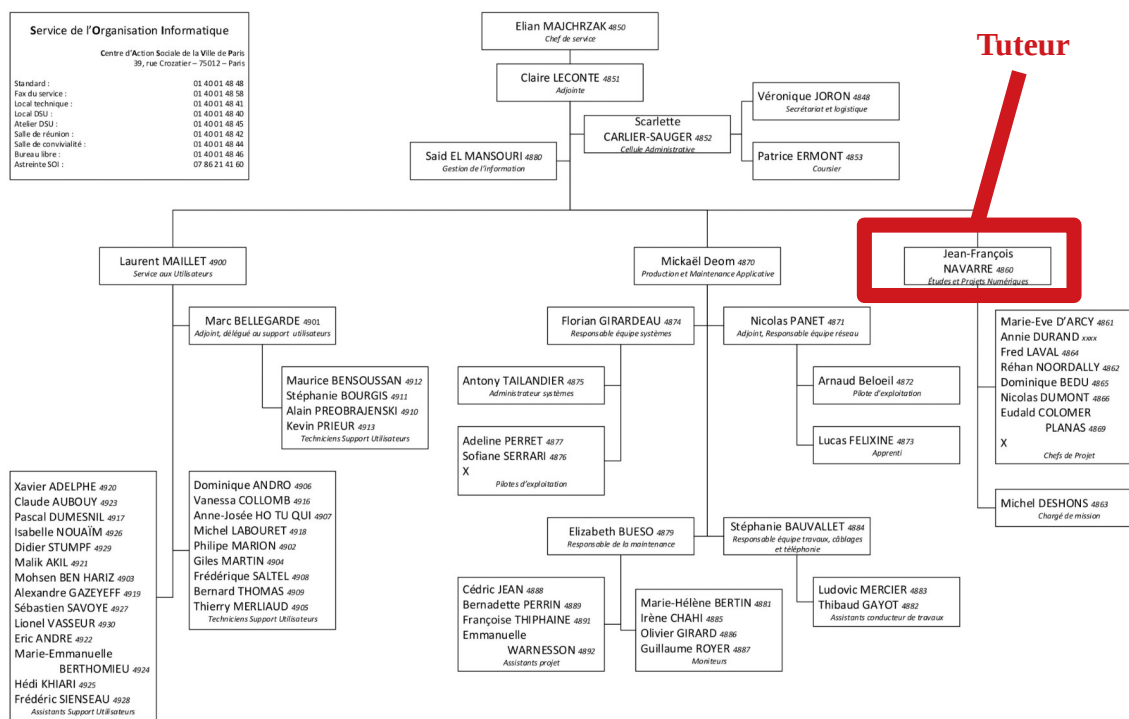
Le Centre d'action sociale de la Ville de Paris (CASVP) est un établissement public municipal.

Il obéit aux règles de droit commun d'un centre communal d'action sociale, sous réserve des textes spécifiques liés à l'organisation administrative de Paris.

Le CASVP remplit une mission de prévention et de développement social sur le territoire parisien. Il met en œuvre la politique d'action sociale définie par la Ville de Paris : instruction et délivrance des nombreuses prestations sociales de la Ville ; gestion de plus de 140 établissements pour l'accueil ou l'hébergement des personnes âgées, en difficulté et sans abri ; développement de dispositifs de services à la personne. Le CASVP exerce aussi, par délégation, certaines missions d'État, telles que l'hébergement d'urgence et l'aide aux personnes SDS (sans domicile stable). D'autres missions lui sont confiées par le Département de Paris : les aides financières d'aide sociale à l'enfance, certaines aides sociales légales, le service social de proximité... Engagé dans de nombreux projets, le CASVP évolue en permanence pour adapter son organisation à ses missions, à son public, et au contexte financier.

b) Organigramme

Organigramme du service informatique du CASVP:



c) Quelques chiffres

6000 agents permanents,

580 millions € de budget,

180 millions € d'aides et d'allocations,

+ de 140 établissements au service des usagers.

II/ Projet

a) Etude de l'existant

Avant que j'arrive, le service financier a demandé à mon équipe de leur créer un site qui permettrait de convertir un fichier en .csv en fichier .xml. J'ai donc continué le projet qui avait déjà été démarré. J'avais alors à ma disposition les codes du site pour la conversion ainsi qu'un serveur distant pour stocker les fichiers convertis. J'avais aussi en ma possession une base de données qui avait déjà été créée mais qui n'était pas très optimisée. Le plus gros du projet, l'algorithme de conversion, avait déjà été bien commencé.

b) Objectifs

Il me restait cependant beaucoup de choses à faire:

- créer un compte sur phpMyAdmin dédié à la BDD

- ajouter l'extension .xml au fichier issu de la conversion

- améliorer l'ergonomie (ajout d'un fil d'ariane + bouton pour navigation)

- enregistrer chaque action sur le site dans une table, les afficher sur un tableau de bord (connexion, conversion échouée et réussie...)et indiquer dans le tableau de bord si la conversion est réussie ou non, si elle a échoué on doit pouvoir accéder à l'erreur dans le tableau de bord

- envoyer suite à la conversion le fichier qui en est issu vers un serveur distant avec un envoi SFTP puis lorsque le fichier sur le serveur distant est traité par l'utilisateur de son côté

- le récupérer lors du rafraichissement du tableau de bord et affichage sur la ligne de la conversion réussie qui est liée.

c) Moyens utilisés



J'avais à ma disposition plusieurs outils pour me permettre de mener mon projet à savoir:

-Deux serveurs → un local pour stocker mon projet, les messages d'erreur, les fichiers retours ainsi que les fichiers temporaires et un distant pour permettre aux usagers de traiter sur un autre logiciel les fichiers issus de la conversion.

-PhpMyAdmin pour la gestion de base de données

-NotePad++ m'était imposé comme IDE

-Je devais utiliser uniquement php comme langage de programmation pour ce projet (pas de javascript)

-Pour le design, j'ai utilisé exclusivement bootstrap.

(-Pour l'autre projet, j'ai aussi utilisé JQuery car mon maître de stage voulait un site dynamique)

d) Méthodologie de travail

Pour commencer, il faut savoir que j'étais assez autonome dans la création du site bien que j'avais chaque semaine une liste de chose à faire. Lorsque j'avais fini, mon tuteur voulait voir ce que j'avais fait puis regarder le code pour voir si c'était bien optimisé. Si ça ne l'était pas, et ça m'est arrivé, je devais chercher une meilleure solution. En plus de ces réunions, j'ai rencontré deux fois la MOA afin qu'elle m'indique si mon site lui convenait et comment voulait-elle que celui-ci évolue. J'ai aussi participé à deux réunions du département dans lequel j'étais où chacun présentait l'avancée de leur projet. Pour finir, j'ai eu une réunion avec le service communication afin qu'il me fournisse la charte graphique du CASVP et qu'il me confirme que mon site rentre bien dans les clous. De mon côté, il y a quelques fonctions que j'ai dû apprendre à utiliser en recherchant sur la doc de php comme la fonction d'envoi SFTP d'un fichier ou encore les fonctions permettant de remplir un fichier xml. Je devais au fur et à mesure de l'avancement de mon projet remplir la documentation technique (MCD de la nouvelle BDD, explication de l'utilité de chaque fichier... etc) ainsi que mettre des commentaires pour chaque fonction.

Premier mail de mon tuteur qui me résume ce que je dois faire:

info stage
Navarre Jean-François
Catégorie rouge
Envoyé : lun. 06/01/2020 09:11
À : Thoury Erwan

Srv-re7-ast01-v
Erwan / Stage2020

Dev convertisseur ASTRE

- Sécurité : compte dédié pour accès BD
- Extension .XML
- Répercuter modif sur écran de test.
- Ergonomie :
 - Fil d'ariane
 - Retour à la page d'accueil
- Trace cnx, upload, dépôt
- Envoi SFTP
- Affichage historique
- Récupérer fichier retour et affichage

Page intraCSAVP (Lutece) du SOI :

- Doc W10 : état des lieux, maquette,

Mail de la MOA qui me demande de vérifier mon code car elle a un problème dans le tableau de bord (en l'occurrence, elle avait un problème lié à la connexion internet et pas à mon code):

Test d'intégration
Arrahaoui Saleoua
Vous avez répondu à ce message le 14/02/2020 11:32.
Envoyé : ven. 14/02/2020 11:18
À : Thoury Erwan

Bonjour Erwan,

J'ai fait un test d'intégration hier, mais le fichier n'apparaît pas.
Peux-tu vérifier ?

Tout	Connexion	Conversion					
n°	Action	Libellé du fichier	Base	Date	Réussite	Retour	Erreur
1	connexion			14/02/2020			
2	conversion		Maquette-TEST	13/02/2020	Non		/erreur_130220_181945.txt
3	conversion		Maquette-TEST	13/02/2020	Non		/erreur_130220_181812.txt
4	connexion			13/02/2020			

Merci

e) Analyse

Je devais donc faire un site qui remplit tous les objectifs cités précédemment. J'ai commencé par refaire tout le design du site avec Bootstrap en mettant un bandeau avec:

- des boutons de navigations qui se mettent à jour à chaque changement de page (lorsque nous sommes sur le tableau de bord, le bouton «Tableau de bord» disparaît, lorsque nous sommes sur la page d'accueil, le bouton «Accueil» disparaît ...etc)

- un bouton de déconnexion avec le prénom de l'utilisateur à côté. Cela permet de confirmer que nous sommes bien sur notre compte.



Ensuite j'ai créé le tableau de bord qui est la page principale d'un usager qui s'est connecté. Sur ce tableau de bord, on a alors un visuel de toutes les actions enregistrées. Au dessus du tableau, on peut choisir quel type d'action on veut voir dans le tableau de bord. Les actions ont des caractéristiques en commun:

- «n°» numéro de l'action dans le tableau (ne correspond pas au numéro de l'action dans la bdd mais bien au numéro d'ordre d'apparition dans le tableau)
- «Action» est le type d'action (soit connexion, soit conversion)
- «Date» correspond à la date de l'action
- Bouton supprimer à chaque ligne.

Il y a cependant des différences selon le type d'action mais aussi selon si la conversion est réussie ou non:

- Pour une connexion, il n'y a aucune colonne de plus.
- Pour les conversions, on ajoute le nom du fichier de base
- Pour une conversion réussie, on affiche le libellé du fichier .xml généré, un «oui» vert pour indiquer la réussite de la conversion et lorsqu'il y a un fichier retour généré sur le serveur distant par un autre logiciel de l'utilisateur, on le récupère et le site peut le lire grâce au lien de la colonne Retour.
- Pour une conversion ratée, on affiche juste le lien vers le message d'erreur. Cela permet à l'utilisateur de pouvoir corriger son fichier .csv qui est mal agencé.

Tout	Connexion	Conversion						
n°	Action	Libellé du fichier	Base	Date	Réussite	Retour	Erreur	
1	conversion		FC240120-err	29/01/2020	Non		/erreur_290120_172513.txt	supprimer
2	connexion			29/01/2020				supprimer
3	conversion	/FC_290120_143122.XML	FC270120	29/01/2020	Oui			supprimer

J'ai aussi écrit le code permettant l'envoi du fichier converti sur le serveur distant dont voici le code (\$resultat étant la requête permettant de récupérer les paramètres dans la BDD):

```
//Tous ces paramètres sont ceux qui permettent de se connecter au répertoire distant
$host = $resultat->host;
$port = $resultat->port;
$username = $resultat->identifiant;
$password = $resultat->motDePasse;
$fichierDistant = $resultat->fichierDistant.$nomfic;

$connection = @ssh2_connect($host, $port); //On se connecte donc au serveur distant
$auth = @ssh2_auth_password($connection, $username, $password); //Connexion et authentification au serveur sftp
$sftp = @ssh2_sftp($connection); //Initialise le sous-système SFTP
$sftpStream = @fopen('ssh2.sftp://' . $sftp . $fichierDistant, 'w');//On ouvre le serveur distant
```

Et ici on peut voir toutes les vérifications pour éviter une erreur fatale qui empêcherait l'utilisateur de continuer sa navigation, en bas on peut aussi voir la fonction fopen() et fwrite() pour l'écriture sur le serveur distant:

```
try { //Si il y a des problèmes dans l'envoi ou dans la connexion, retourne une erreur
    if (!$sftpStream)
    {
        throw new Exception("Impossible d'ouvrir le fichier distant, veuillez réessayer <br>");
    }

    $data_to_send = @file_get_contents($fichierLocal);

    if ($data_to_send === false)
    {
        throw new Exception("Impossible d'ouvrir le fichier local, veuillez réessayer <br>");
    }

    if (@fwrite($sftpStream, $data_to_send) === false)
    {
        throw new Exception("Impossible d'envoyer le fichier local, veuillez réessayer");
    }

    fclose($sftpStream);
} catch (Exception $e)
{
    error_log('Exception: ' . $e->getMessage());
    $erreur_message = $e->getMessage();
}

$handle = @fopen('/var/www/html/tmp/' . $nomfic, "w"); // On crée fichier en fonction du nombre de ligne dans le fichier csv et de la date que l'on place dans le dossier de l'utilisateur de la session
if ($handle)
{
    fwrite($handle, $strxml); // On écrit dans le fichier le fichier xml
    fclose($handle);
}
```


Enfin, lorsque l'utilisateur a traité le fichier xml avec un autre logiciel, je fais la démarche inverse à précédemment et je récupère le fichier traité sur le serveur distant lorsque je rafraîchis le tableau de bord. Une fois récupéré, je mets dans la BDD les informations de chaque ligne traitées.

```
$fichierSFTP_Distant_src = fopen($fichierSFTP_Distant, 'r'); //On ouvre le fichier que l'on veut envoyer
if($fichierSFTP_Distant_src)
{
    $fichierLocal_des = fopen($fichierLocal, 'w'); //On ouvre le fichier que l'on veut remplir
    if($fichierLocal_des)
    {
        $envoi = stream_copy_to_stream($fichierSFTP_Distant_src, $fichierLocal_des); //On remplit le fichier avec ce que l'on a envoyé
        fclose($fichierSFTP_Distant_src); //On referme
        fclose($fichierLocal_des); //On referme
    }
}
// aller chercher le nom du fichier en base plutot que faire un traitement de chaine
$requete = mysql_query("SELECT libelle
                        FROM Action
                        WHERE idaction=(SELECT numeroAction FROM LigneTraitee WHERE `idLigneTraitee`='".$numeroLigneChiffre."');") OR die(mysql_error());
$fichierOrigine = mysql_fetch_object($requete);

if($envoi) //Si le fichier a bien été envoyé, on supprime le fichier sur le serveur distant
{
    $suppression=ssh2_sftp_unlink( $sftp , $repertoireDistant.$fichierOrigine->libelle );
    $suppression=ssh2_sftp_unlink( $sftp , $repertoireDistant."/".$resultat );
}
```

Voilà la fin de l'analyse du premier projet qui m'a pris 4 semaines à finir, j'ai mis 1 semaine et demi pour finir le deuxième qui a été fait sous ajax.

f) Schémas

MCD de la BDD pour le projet:

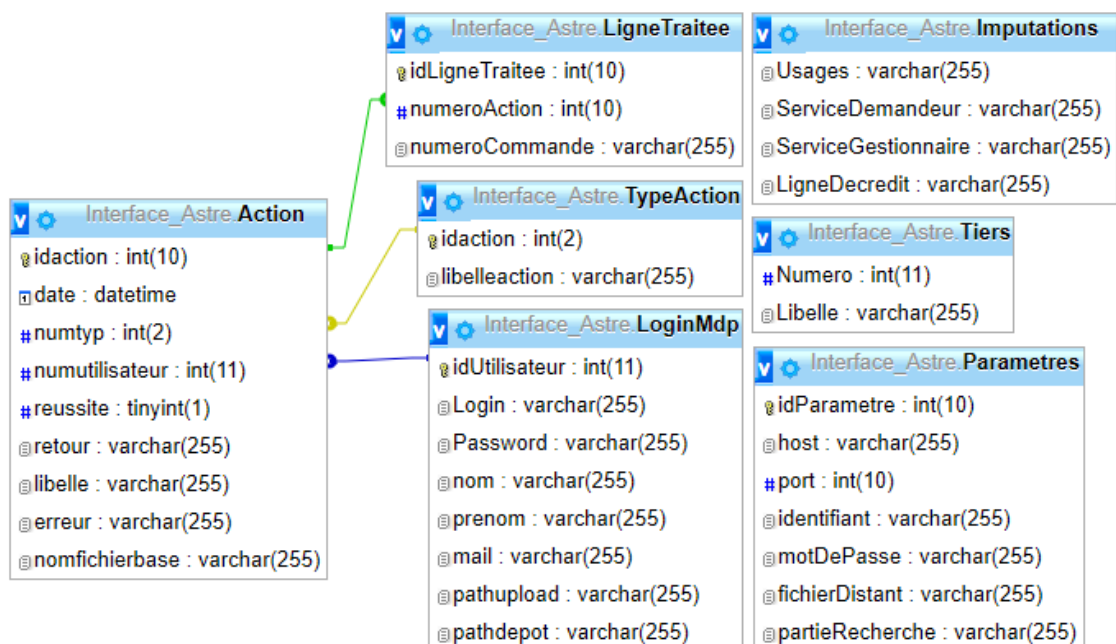


Table Action :

Cette table répertorie toutes les actions qui sont à mettre dans le tableau de bord. On a comme type d'action la connexion et la conversion, qu'elle soit réussie ou qu'elle ait échoué.

Ces attributs sont :

- idaction (int): est la clef primaire et s'auto-incrémente.
- date (datetime) : permet de conserver la date de l'action.
- numtyp (int): permet d'indiquer la nature de l'action. C'est aussi une clef étrangère qui permet de faire la jointure avec la table TypeAction.
- numutilisateur (int) : permet de conserver le numéro de l'utilisateur à l'origine de l'action. C'est aussi une clef étrangère qui permet de faire la jointure avec la table LoginMdp.
- reussite (tinyint) : booléen permettant d'indiquer la réussite ou non d'une conversion.
- retour (varchar): indique le nom du rapport d'une conversion réussie.
- libelle (varchar) : indique le nom du fichier xml résultant de la conversion.
- erreur (varchar) : indique le nom du fichier erreur.txt d'une conversion ratée.
- nomfichierbase (varchar) : indique le nom du fichier avant la conversion.

Table LigneTraitee :

Cette table répertorie toutes les informations importantes à récupérer sur chaque ligne traitée lors d'une conversion réussie.

Ces attributs sont :

- idLigneTraitee (int) : est la clef primaire et s'auto-incrémente.
- numeroAction (int) : permet de savoir à quelle conversion est liée la ligne traitée. C'est aussi une clef étrangère qui permet de faire la jointure avec la table Action
- numeroCommande (varchar) : permet de conserver le numéro de commande qui se trouve dans le rapport

Table LoginMdp :

Cette table répertorie toutes les informations des utilisateurs ainsi que le chemin pour déposer les fichiers.

Ces attributs sont :

- idUtilisateur (int) : est la clef primaire et s'auto-incrémente.
- Login (varchar) : est le login de l'utilisateur qui lui permettra de se connecter.
- Password (varchar) : est le mot de passe de l'utilisateur qui lui permettra de se connecter.
- nom (varchar) : permet de conserver le nom de l'utilisateur.
- prenom (varchar) : permet de conserver le prénom de l'utilisateur.
- mail (varchar) : permet d'avoir l'adresse de l'utilisateur s'il a un problème.
- pathupload (varchar) : permet d'avoir le lien vers le fichier où sont chargés les .csv à convertir.
- pathdepot (varchar) : permet d'indiquer le lien vers le fichier où il faut ranger les fichiers .xml convertis, les fichiers .txt d'erreur et enfin les rapports

Table TypeAction :

Cette table répertorie tous les types d'action qui sont à répertorier dans la table Action.

Ces attributs sont :

- idaction (int) : est la clef primaire et s'auto-incrémente.
- Libelleaction (varchar) : est le libellé (soit connexion, soit conversion).

Table Parametres :

Cette table répertorie tous les paramètres utiles que ce soient les identifiants de connexion au serveur sftp ou le mot à rechercher lors de la récupération du rapport.

Ces attributs sont :

- idParametre (int) : est la clef primaire et s'auto-incrémente.
- host (varchar) : est l'hôte nécessaire pour se connecter au serveur distant.
- port (int) : est le port nécessaire pour se connecter au serveur distant.
- identifiant (varchar) : est l'identifiant nécessaire pour se connecter au serveur distant.
- motDePasse (varchar) : est le mot de passe nécessaire pour se connecter au serveur distant.
- fichierDistant (varchar) : est le lien vers le répertoire distant.
- partieRecherche (varchar) : est le mot à rechercher lors de la récupération du rapport.

Table Tiers :

Cette table répertorie tous les tiers.

Ces attributs sont :

- Numero (int) : est la clef primaire.
- Libelle (varchar) : est le libellé du tiers.

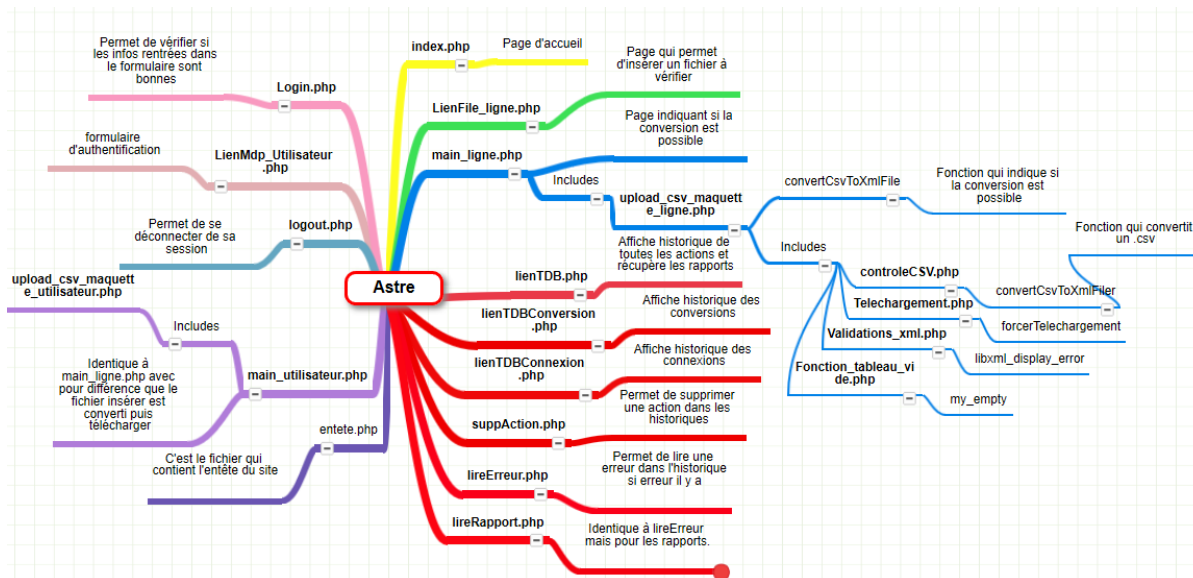
Table Imputation :

Cette table répertorie toutes les imputations.

Ces attributs sont :

- Usages (varchar) : permet d'indiquer l'usage.
- ServiceDemateur (varchar) : est le numéro du service demandeur.
- ServiceGestionnaire (varchar) : est le numéro du service gestionnaire.
- LigneDeCredit (varchar) : est le numéro de la ligne de crédit.

Carte des fichiers qui indique les différents chemins:



g) Problèmes rencontrés

Au cours du projet, j'ai eu quelques problèmes que je vais lister ci-dessous:

- Langage imposé et IDE pour le premier projet: j'ai dû faire le premier projet qu'avec php alors que certaines fonctionnalités auraient pu être codé avec un langage dynamique tel qu'ajax et j'aurais préféré utiliser Visual studio comme IDE plutôt que notepad++ qui est bien plus ergonomique avec ces extensions.
- Beaucoup de fonctionnalités que je ne savais pas comment ajouter; par exemple, je devais faire des envois SFTP avec des fonctions php, chose que je ne savais absolument pas faire avant et qui m'a pris du temps en recherche.
- Je devais aussi écrire la documentation liée à mon projet et mettre des annotations dans mon code. Je sais que c'est une obligation lorsque l'on code mais j'ai l'impression de perdre mon temps.

III/ Conclusion

Ce stage m'a beaucoup appris et j'ai été très content d'y être allé pour de multiples raisons:

- J'ai appris à travailler dans une entreprise, ce qui est différent du travail en cours ou à la maison.
- J'ai dû m'adapter aux technologies mises à ma disposition.
- J'ai dû apprendre à écouter ma MOA ainsi que le service responsable de la charte graphique.
- J'ai dû aussi faire beaucoup de recherche seul ce qui a développé dans le même temps mon autonomie.

En conclusion, j'ai passé un très bon stage qui, a n'en pas douter, m'a fait grandir. Mon tuteur a été satisfait de mon travail et m'a demandé si je voulais faire mon alternance au CASVP ce qui pourrait être une bonne chose si je décide de faire mes études en alternance.