

**Berdah
Clément
Saphir | BUT 2**

Stage du 22/01/24 au 29/03/24

Lieu
MGC | Connecting. Groupe Figaro.
9 Rue de Caumartin
75009 Paris

Tuteur
M. MOREAU Bruno
Directeur Exécutif
06 29 96 50 80

Rapport de stage

MGC | CONNECTING

U N I V E R S I T É
SORBONNE
PARIS NORD

Stage de Développeur Full-Stack au sein de l'équipe informatique.

1. Remerciements



Je tiens à remercier toute l'équipe de **MGC | Connecting** avec laquelle j'ai passé la durée de ce stage.

J'aimerais également remercier toutes les personnes qui ont pu me consacrer une partie de leur temps afin de m'accompagner durant ce stage, et ainsi le rendre passionnant.

Je remercie les développeurs avec lesquels j'ai pu travailler en étroite collaboration, **Mr Hermann HOL** pour la conception de l'interface web, ainsi que **Mr Laurent ESPANA** pour l'expansion de la base de données et du script.

Enfin, je remercie tout particulièrement mon maître de stage **Mr Bruno MOREAU** qui m'a très largement aidé durant ce stage, et qui m'a accordé sa confiance afin de complètement m'intégrer à l'équipe.

2. Sommaire

I. Introduction.

1. Recherche du stage.4
2. Mon arrivée au sein de l'entreprise (service informatique).....4

II. Présentation de l'entreprise.

1. Contexte.6
2. Multidiffusion.7
3. Feedback.7
4. Reporting.8

III. Cadre et visée du projet.

1. Contextualisation du projet.....9
2. Objectifs du projet.....9

IV. Expansion de la base de données.

1. Présentation de la mission.10
2. Réalisation de la mission.11
3. Bénéfice de ce travail.12

V. Création d'un script d'alimentation.

1. Présentation de la mission.13
2. Réalisation de la mission.13
3. Bénéfice de ce travail.15

VI. Interfacer les données : visualisation via une page web.

1. Présentation de la mission.16
2. Réalisation de la mission.16
3. Bénéfice de ce travail.19

VII. Gain en efficacité.

1. Meilleure gestion d'organisation.19
2. Meilleure communication.20

VIII. Conclusion.

1. Résumé.21
2. Ressenti personnel.21
3. Orientation (BUT3).22

IX. Lexique.23

X. Annexes.24

I. Introduction.

1. Recherche du stage.

Lors de cette année de BUT2, une période de stage en entreprise est obligatoire dans le cursus. Ce stage s'étale à partir de la fin du mois de janvier pour une durée de 8 semaines minimum jusqu'à 10 semaines maximum. A l'approche de la fin du premier semestre, je me suis mis en recherche active du stage. Avec seulement 1 an et demi d'étude et une durée de stage relativement courte, beaucoup d'entreprises sont réticentes à accueillir des stagiaires. En effet, il y avait beaucoup d'offres de stage mais avec des exigences qui ne correspondaient pas avec le profil de mon stage. Il a fallu ainsi optimiser mes chances de trouver ce stage.

Les entreprises souhaitent évidemment avoir affaire à un étudiant sérieux et professionnel. Certaines offres exigeaient de voir le profil LinkedIn du potentiel stagiaire. Ainsi j'ai créé mon profil sur ce réseau et je l'ai rendu le plus professionnel possible. Mon CV devait également être soigné, de même que ma lettre de motivation. Mon entourage et surtout mes professeurs m'ont aidé à améliorer ces deux documents. L'objectif était que ma candidature intéresse les rares offres de stage qui correspondent à mon profil. En parallèle de ces recherches sur des sites d'offres d'emplois/stages, j'ai aussi activé mon réseau personnel, ce qui a fini par être l'option la plus efficace.

Après quelques semaines, **Mr Bruno MOREAU** a validé ma candidature au sein de l'entreprise **MGC | Connecting**, pour un poste de stagiaire, du 22 janvier au 29 mars 2024. C'est ainsi que s'achève ma recherche et que je trouve le stage en tant que **développeur FULL STACK**.

L'avantage d'avoir un stage en tant que développeur FULL STACK est qu'on se complète et se renforce en compétences et connaissances sur plusieurs aspects du métier de développeur, ce qui est un élément très important pour moi. Je souhaitais élargir mon champ d'intervention afin de développer le plus de compétences possibles.

2. Mon arrivée au sein de l'entreprise (service informatique)

C'est le lundi 22 janvier que je suis arrivé pour la première fois chez MGC | Connecting. A mon arrivée, j'ai pu visiter les locaux et faire connaissance avec une

partie de l'équipe, notamment les personnes avec lesquelles je serai amené à travailler.



Afin que je puisse être parfaitement intégré à l'activité de l'entreprise, un ordinateur a été mis à ma disposition, et configuré avec tous les éléments nécessaires pour que je puisse accomplir mes tâches.

C'est à cet endroit que j'ai pu également récupérer mon badge d'accès ainsi que mon ordinateur de travail.

Le "Carré IT" du Figaro, Rue de Provence, à Paris.

Mr Moreau, mon maître de stage, a validé l'organisation de mon planning de travail. Il était indispensable que je sois toujours accompagné d'un des deux développeurs lors de ma présence sur site. Nous avons donc défini que mes jours de télétravail seraient les mardis et les vendredis, calqués sur l'organisation de l'équipe. Cela me permet donc d'être toujours en présentiel avec toujours au moins un des 2 développeurs.

Les communications internes se font avec le logiciel Cisco Jabber ainsi que Microsoft Outlook (pour les mails).



Cisco Jabber était utilisé pour de la messagerie instantanée, mais aussi pour les appels, lorsque j'étais en télétravail. C'est grâce à cet outil que je pouvais contacter les développeurs en cas de besoins.

Grâce à Microsoft Outlook, j'accède à ma boîte mail et reçoit, entre autres, les invitations pour les réunions.



Ces deux messageries m'ont été configurées par le Carré IT.



Concernant la partie développement, j'ai choisi d'utiliser le logiciel Microsoft Visual Studio Code. C'est le logiciel avec lequel je code tout le temps chez moi, et aussi pour les travaux divers pour l'université. J'avais besoin également d'un logiciel qui prenne en compte plusieurs langage comme le `c#`, `php`, `javascript`...

J'utilisais également les services déjà présents dans l'entreprise comme Microsoft SQL Server Management studio pour la gestion des bases de données.



Enfin, le serveur FTP, serveur sur lequel toute l'entreprise peut partager des dossiers ou des fichiers. Cet outil me sera utile pour la création de mon script.

Enfin, j'ai utilisé XAMPP, qui est un logiciel qui permet de simuler une adresse serveur local sur mon pc. Cela a été très utile, notamment dans la dernière partie de mon stage, lors de la création de ma page web.



Tout était au point pour que je puisse travailler dans de bonnes conditions, que ce soit sur site, ou bien en télétravail.

Une fois que tout fut bien configuré, l'équipe de MGC a pu m'attribuer un rôle dans le service informatique et un projet complet sur lequel j'allai travailler pendant les 10 semaines de stage.

II. Présentation de l'entreprise.

2.1. Contexte.

MGC | Connecting (MGC) est une agence de communication. Cette entreprise fait partie du **Groupe Figaro** qui lui-même appartient au **Groupe Dassault**. Plus précisément, il s'agit d'une agence de communication immobilière.

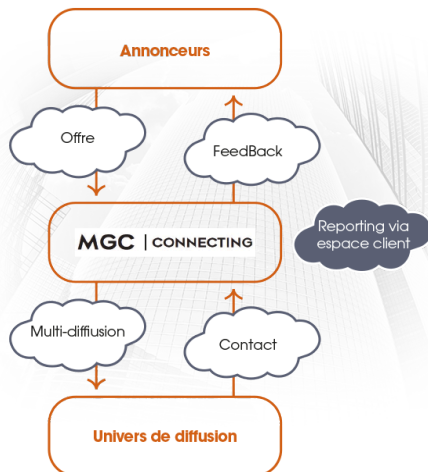
MGC intervient comme prestataire de services, auprès des constructeurs et promoteurs immobiliers, pour la gestion des offres immobilières et la gestion des leads clients. MGC décharge ses clients de ces tâches.

Ainsi, MGC agit en support de ses clients afin qu'ils puissent gagner du temps et se concentrer sur leur cœur de métier.

MGC intervient sur la **multidiffusion**, le **feedback** mais aussi le **reporting**.

2.2. Multidiffusion (flux descendant).

Lorsqu'une agence immobilière désire promouvoir ses offres, elle peut faire appel à MGC. Une agence immobilière a toujours des offres de ventes. Ces offres peuvent être des biens (une maison précise, un appartement précis...) ou bien des lots (un



groupement de biens avec les mêmes caractéristiques). Dans le but de faciliter la vente de ces biens, l'agence immobilière peut faire appel au service de **multidiffusion** proposé par MGC. Dans ce cas, plusieurs solutions existent. En premier lieu, le client envoie ses annonces dans des fichiers et précise (ou a déjà précisé) les sites d'annonces immobilières (portails) sur lesquels il souhaite diffuser ses offres, comme **SeLogger** ou bien **Immoneuf**. L'autre proposition mise en place par MGC,

est d'utiliser une API permettant d'aller récupérer directement, auprès du client, les annonces + portails. Enfin, MGC met à disposition à ses clients le site **One**. Celui-ci permet à chaque client de venir modifier manuellement les portails sur lesquels seront diffusées les annonces. Une fois que MGC a récupéré les informations nécessaires, la multidiffusion sur les différents portails est opérationnelle. Une fois la publication des annonces faite, l'étape suivante de ce processus est le retour client (Feedback).

2.3. Feedback (flux ascendant).

Le feedback est le retour des résultats obtenus à la suite de la multidiffusion. Chacun de ces sites, ont sur leur portail, un espace dédié avec un formulaire de demande de contact. Cet espace est utilisé par les internautes afin de montrer leurs intérêt pour une offre.

MGC intervient ici en récupérant les leads (retour internaute pour le client). Sur ces portails, lorsqu'un internaute saisit le formulaire de contact, MGC reçoit un mail avec les informations correspondantes. MGC renvoie donc à son client, les informations contenues dans les leads. Le client peut demander à MGC de formater les leads dans le but de rendre la collecte d'informations plus facile. MGC a mis en place un programme qui va, en fonction du client et du lead reçu, récolter les informations nécessaires.

Une fois les informations récupérées, l'objectif est de les envoyer au client concerné. Pour ce faire, en fonction de la préférence du client, MGC renvoie les informations. Ce feedback peut être fourni par mail, fichiers, ou quand le client possède un CRM, MGC peut l'intégrer directement dedans. Afin de compléter cette prestation, MGC donne également au client la possibilité de visualiser sa performance.

2.4. Reporting.

Le reporting, c'est ce qui regroupe les éléments de suivi et de mesures des retours. Le reporting sert à donner aux clients, une idée de ce qui marche le mieux pour eux. MGC propose un espace client dans lequel chacun d'entre eux peut aller voir les statistiques relatives à ses annonces, ses portails...

Lorsque MGC récupère les informations pour le feedback, il les calcule et les rentre dans une autre base de données. Cette base permet ensuite aux clients de voir les statistiques. Toutes ces informations sont disponibles pour le client sur l'espace client du site.

Sur ce graphique (*cf annexe 1.*), on peut voir un graphique permettant à un client x de visualiser le nombre de formulaires en fonction du support (portail).

De même sur le graphique suivant (*cf annexe 2.*), le client peut avoir le même graphique mais par mois, et donc voir la répartition détaillée sur toute l'année.

Ces statistiques permettent au client de voir les performances sur un mois, sur un portail.

Grâce à ce service de reporting proposé par MGC, les clients n'ont pas à s'occuper de cette partie, gagnent du temps et profitent d'un service leur permettant de mesurer l'activité et l'efficacité de la diffusion de leurs offres.

III. Cadre et visée du projet.

1. Contextualisation du projet.

Lors de mes premiers jours, j'ai pu prendre connaissance des projets en cours de l'entreprise et donc ce sur quoi j'allais pouvoir apporter mon aide. Ainsi dans le cadre de mon stage de développeur full stack, et MGC qui souhaitait compléter et retravailler son Espace Client, j'ai ainsi pu participer à ce projet, et proposer à ses utilisateurs de bénéficier d'une expérience plus complète. Comme exposé précédemment, cet espace client est assez complet sur les rapports de statistiques relatifs à la performance réalisée par un client en fonction de ses annonces, de ses portails, au cours de l'année. Ce qui intéresse également les clients, ce sont les coûts des prestations de MGC. En effet, pouvoir vérifier ses performances est un service très important proposé par l'Espace Client, mais voir si les coûts de ces prestations est également primordial afin d'en mesurer la rentabilité. Les clients ont la nécessité de vérifier si la performance liée à tel portail est financièrement rentable. Sans ces informations, le client naviguait de manière aveugle sans savoir si sa stratégie de diffusion est financièrement pertinente.

De plus, le manque de détail sur les facturations existait aussi en interne chez MGC. Il manquait un système facile et pratique pour l'interne afin de suivre les finances, le détail des factures.

C'est alors que comme pour le client ou bien le service interne de MGC | Connecting, il fallait absolument enrichir cet espace client.

2. Objectifs du projet.

Ce projet était très important, visait à apporter du confort pour tous les utilisateurs, qu'ils soient internes ou bien externes (les clients). Il faut donc que ce projet soit optimisé et bien réfléchi afin d'anticiper tout problème mais aussi qu'il puisse durer dans le temps, même avec l'ajout de nouvelles fonctionnalités. Enfin, ce projet doit absolument s'intégrer avec l'existant.

Améliorer l'espace client peut se traduire par beaucoup de moyens : optimisation de l'espace du site web, embellissement des pages web... Dans le cas de MGC, l'objectif est de rajouter une nouvelle fonctionnalité. Cela se traduira par l'ajout d'un nouvel onglet qui sera spécialement dédié aux données financières. Pour pouvoir avoir accès à ces données financières, il fallait revoir la manière de les traiter mais aussi de les afficher. Ces données financières sont vues différemment par les clients ou l'interne de MGC. En effet, ces données sont des **factures** et donc il y a l'achat, la vente et donc le calcul du chiffre d'affaires est possible. L'objectif est de faire la

dissociation entre ce que les clients verront sur leur espace client, et ce que doit pouvoir regarder le service interne de MGC.

Au démarrage, j'ai pu assister à des réunions pour qu'on commence à définir l'objectif principal de ce projet. L'objectif est d'obtenir une page web, sur laquelle une navigation fluide, simple et optimisée permettra à quiconque voulant faire des recherches dessus, de visualiser sa demande de manière claire, nette et rapide.

Pour arriver à ce résultat, il a fallu d'abord découper le projet en plusieurs missions.

Tout d'abord il faut avoir une structure de base de données claire, pour pouvoir manipuler plus facilement les données. Nous avons donc déterminé qu'il était nécessaire de rajouter une table dans l'une des bases de données. Ensuite, une fois que toutes les données avaient un endroit pour être stockées, il fallait bien les insérer dans la base de données. En termes de volumétrie, on parle de plusieurs milliers de lignes provenant d'un fichier csv, à rentrer tous les mois. C'est donc évidemment impossible de faire tout cela à la main. L'alternative proposée est donc de créer un script (un .exe) permettant de mettre les données nécessaires dans la table créée, et cela de manière totalement automatique. Une fois arrivé à un stockage des données stables, l'objectif principal rentre en jeu : interfacier ces données sur une page web pour permettre aux utilisateurs de naviguer librement entre ces milliers de lignes.

Afin de mener à bien ce projet, j'ai pu être entouré des deux développeurs de l'équipe, Laurent ESPANA (pour les deux premières étapes du projet) ainsi que Hermann HOL (pour l'interface web).

IV. Expansion de la base de données.

1. Présentation de la mission.

Afin de bien démarrer ce projet et dans le but de pouvoir créer mon script d'insertion automatique, puis par la suite l'interface web, nous devons créer une table dans l'une des bases de données. Même avec une présentation complète du système des bases de données de **MGC | Connecting** prodiguée par Laurent ESPANA, le développeur sénior, j'ai dû valider avec lui afin de placer cette table au bon endroit. En effet, étant donné que pour une entreprise comme celle-ci les données sont réparties sur plusieurs bases de données, j'ai dû commencer le projet avec de l'aide. Dans cette première mission j'ai pu utiliser mes connaissances en termes de logistique de base de données. J'ai pu également utiliser mes compétences en SQL.

2. Réalisation de la mission.

Le projet que nous sommes en train de commencer à créer, vise à exposer des données financières. Comme dit précédemment, dans l'espace client, MGC met en place une partie permettant de voir des statistiques détaillées, le reporting. Et comme nous allons manipuler des statistiques, nous décidons donc de créer notre table SQL dans la base de données Reporting, donc la base de données qui recense toutes les statistiques nécessaires au client.

Une fois l'important emplacement de la table choisie, il faut définir les champs. Comme dit précédemment, les données que nous comptons rentrer dans cette table sont des lignes provenant d'un fichier csv. La définition des colonnes s'est donc faite sur les colonnes du fichier csv en question.

Afin de suivre également un pattern permettant à l'équipe interne de gérer les données, nous décidons de rajouter 3 champs, CREATED, MODIFIED ainsi que DELETED. Les champs CREATED et MODIFIED sont initialisés à la date de l'insertion des données. Le dernier champ DELETED lui est de type boolean, c'est à dire 1 ou 0. Initialement à 0, il sera modifié à un 1 si modification nécessaire en suite. Afin de prévoir une facilitation de navigation dans notre future interface web, nous comptons laisser l'utilisateur choisir d'afficher telles données en fonction de telle temporalité. Nous avons donc décider de rajouter 6 autres champs :

- Le semestre de la date de Facturation et de la date de Parution
- Le trimestre de la date de Facturation et de la date de Parution
- L'année de la date de Facturation et de la date de Parution
- Le mois de la date de Facturation et de la date de Parution

La table SQL compte 43 champs différents, dont 11 qui ont été ajoutés. Le reste fait partie du fichier csv qui contient les données intéressées.

3. Bénéfice de ce travail.

En commençant cette première partie du projet j'avais sans doute mal jugé la professionnalisation et l'exigence que je devais avoir. J'ai cru qu'une simple table SQL allait être une mission simple. En effet, créer une simple table SQL n'a jamais été difficile. Mais j'ai tout de même dû poser plusieurs questions au développeur sénior afin d'obtenir de l'aide.

Afin de garder la cohérence du modèle de données, il fallait au préalable vérifier que certaines informations du fichier csv, ne soient pas déjà présentes dans la base de données, afin d'éviter de potentiels doublons.

Plusieurs éléments chacun plus complexe que d'autres étaient à prendre en compte. J'ai dû beaucoup communiquer et je dirai donc que ce travail m'a apporté en communication. En entreprise, il faut savoir exposer un problème de manière claire.

En plus de cela, il faut noter que je codais mon SQL sur le logiciel SQL Serveur à contrario de PSQL comme nous faisons à l'université. Bien que ces 2 manières de coder de la création et/ou implémentation en base de données soient très similaires, des légères différences de syntaxe étaient présentes. J'ai donc dû faire des recherches afin de faire l'accord entre ce que je connais, et la nouvelle syntaxe.

Je dirai donc que ce travail, qui à première vue, n'est pas le plus compliqué de tous, m'a tout de même apporté de l'expérience et de la consolidation en communication. Le travail seul m'a forcément obligé à gagner en autonomie. Et enfin, je dirai que cette première mission m'a fait gagner en connaissance sur le logiciel SQL Serveur.

Ce travail a aussi apporté à l'entreprise une fonctionnalité qui leur permet donc de gagner en informations et en statistiques. Cela est utile pour MGC afin qu'ils puissent connaître davantage les informations liées à leurs factures : leurs finances (ventes et/ou achats).

Une fois que ce travail fut terminé, j'ai pu continuer avec la deuxième étape, la création d'un script qui alimente cette nouvelle table.

V. Création d'un script d'alimentation.

1. Présentation de la mission.

A la suite de la première mission qui consistait à créer une table SQL pour y stocker des informations provenant d'un fichier CSV, il fallait alimenter cette table. Ces informations de factures sont très nombreuses et donc cela implique de pouvoir traiter des fichiers à grande taille.

Pour cette mission, j'ai échangé avec mon maître de stage, Bruno, car ce sera lui le principal utilisateur de ce script. La mission est claire : activer ce script afin qu'il puisse automatiquement remplir la base de données. Pour ce faire nous avons plusieurs éléments à prendre en compte. Tout d'abord, l'endroit où se trouve le fichier. Dans notre cas, le fichier se trouvera dans un dossier présent dans le serveur FTP, serveur de partage. Une autre problématique à prendre en compte, est le type des champs présents dans la table. En effet, même si en lisant le fichier csv, le type des données est assez clair, il faut bien faire attention à ce que chacune de colonnes soit traitée correctement.

Durant ce travail j'ai pu utiliser ma logique que j'ai consolidé au fur et à mesure de mes années d'études au sein de l'IUT. J'ai pu également utiliser mes connaissances en Java car j'ai pu faire un rapprochement entre ce langage et celui que j'allais utiliser.

2. Réalisation de la mission.

Après avoir utilisé SQL Serveur pour créer la table dans la base de données, je suis passé à Visual Studio Code pour le développement de ce script. Au début je pensais rédiger ce code en python ou java, mais on m'a informé que la grande majorité des codes de MGC sont fait en langage C# (C sharp). J'ai donc dû apprendre dans les grandes lignes ce langage afin de me sentir plus à l'aise dans la rédaction du code.

À première vue, j'ai trouvé ce langage très similaire au Java. Les deux sont orientés objets, les deux possèdent des classes, des objets, de l'héritage, du polymorphisme... La syntaxe est également la même (avec quelques différences), les fonctions sont entre {}. Une différence que l'on peut noter entre ces deux langages est l'utilisation de .NET. ".NET" (appelé Dotnet), est une plateforme créée par Microsoft permettant de coder dans plusieurs langages mais en particulier comme ici, le c#. Cette plateforme est indispensable au développement en c#, elle permet notamment d'obtenir de multiples fonctionnalités grâce à ses bibliothèques. Ses bibliothèques sont nombreuses et offrent des éléments tels que : la gestion de

fichiers, le traitement de données, la sécurité, la connectivité réseau et bien plus encore...

J'ai donc tout configuré et pu commencer à coder. Avant de commencer le développement pur, j'ai pu voir avec le développeur sénior pour la logique que nous devons aborder afin d'effectuer à bien ce travail.

Ce travail a donc comme finalité d'avoir un script qui se connectera automatiquement au serveur FTP, puis récupérera le fichier csv. Il lira ensuite toutes les lignes et pour chacune des lignes de ce fichier, il créera une commande SQL INSERT INTO qu'il stockera dans un tableau. Ce tableau contiendra autant de commandes qu'il y a de lignes dans le fichier. Une fois le fichier entièrement parcouru, le script se connecte au serveur SQL et exécute chacune des commandes créées auparavant.

Le script (*cf annexe 3.*), possède donc plusieurs fonctions. En premier lieu, le script utilise la fonction DownloadFileFromFTP pour se connecter au serveur FTP. Par souci de sécurité, les informations sensibles permettant de se connecter au serveur FTP (login, mot de passe) sont stockées dans un fichier externe afin d'éviter de mettre tout directement dans le code de base. Comme on peut le notifier dans la fonction main (*cf annexe 4.*), fonction permettant de lancer l'exécution du script, nous appelons une fonction GetValue qui va lire un tableau qui contient les informations sensibles (qu'on aura récupéré auparavant).

Une fois que le script s'est bien connecté au serveur FTP, il visera le dossier prévu pour y récupérer le fichier csv et le téléchargera 2 fois. Une fois pour le manipuler, et une fois pour le stocker en local. A noter également qu'un sous dossier "Archives" est présent dans le serveur FTP et donc le fichier sera également déplacé dedans.

Le script qui a stocké toutes les lignes du fichier csv dans un tableau, va ensuite appeler une fonction GenerateInsertCommandsFromCsv qui va permettre de créer les commandes SQL INSERT INTO (avec l'aide de la fonction ProcessCsvLine). Le script prend chaque ligne du tableau (donc du fichier) sépare les valeurs et les insère chacune dans la ligne de la commande INSERT INTO. La première ligne (en tête du fichier), permettra de spécifier les champs dans lesquels nous voulons insérer telle valeur.

Il ne faut pas oublier que pour chaque valeur du fichier, il faut qu'elle corresponde avec le type de données prévues dans la table SQL. C'est pour ça qu'avant de finaliser la commande SQL, le script définit bien le type d'une valeur. Par exemple, une valeur nulle pour le champ NetAchat, sera écrit dans le fichier comme "-". Cela signifie que le script pourrait lire cette valeur comme une chaîne de caractères, une chaîne contenant qu'un tiret. Alors que la valeur représente à un float (un nombre décimal) qui est égal à 0.00 (en l'occurrence ici 0.00 €).

Toutes les valeurs sont prises en compte avant d'être retransmises dans la commande INSERT INTO. Nous avons aussi d'autres champs à recalculer :

- $\text{NetVente} = \text{BrutVente} + \text{RemiseClient} + \text{Majo1} + \text{Majo2} + \text{Majo3} + \text{AutresMajo}$
- $\text{NetAchat} = \text{BrutAchat} + \text{RemiseAchat} + \text{Majo3} + \text{AutresMajo}$
- $\text{Nego} = \text{NetVente} - \text{NetAchat}$

Il y avait aussi les fonctions `Get(Month)(Semester)(Trimestre)(Year)FromDate` qui calculent le mois, semestre, trimestre, année des dates correspondantes. Ces fonctions étaient appelées lorsque le script croisé une des deux colonnes : `DateFacturation` et `DateParution`.

Il était donc important de le faire avant de tout mettre dans la commande.

Une fois que nous avons toutes nos commandes de prêtes et qui étaient stockées dans un tableau, il était temps de les exécuter. Dans le même principe que le serveur FTP, le script avait besoin de se connecter au serveur SQL afin de pouvoir procéder à l'exécution des commandes. Les informations sensibles de connexion sont également stockées dans un autre fichier hors du script, et ce afin de gagner en sécurité. Le script utilise donc ces informations pour se connecter à la base de données, et en particulier ici, dans la table que j'ai créée auparavant.

Une fois connecté, le script reprend le tableau qui contient absolument toutes les lignes de commandes correspondantes aux lignes du fichier csv, et les exécute une par une. Afin de rendre l'utilisation de ce script plus intuitive, j'ai rajouté une fonctionnalité qui permet aussi de calculer le pourcentage des commandes exécutées. Cela donne à l'utilisateur une idée du temps qu'il lui reste.

3. Bénéfice de ce travail.

Une de mes plus grandes craintes au début de ce stage, était d'être dans l'incapacité de pouvoir venir en aide à l'entreprise, de peur d'avoir un travail plus important que mes connaissances. Cette mission-là est évidemment un travail que je n'aurai pas imaginé faire mais aussi que je ne savais pas faire. J'ai beaucoup apprécié celui-ci car il change de ce que j'ai l'habitude de faire et me force à sortir de ma zone de confort. J'ai beaucoup appris en réalisant cette mission, j'ai pu gagner des connaissances en C#, un nouveau langage pour moi. J'ai pu également expérimenter la manipulation d'un serveur FTP, de fichiers CSV mais également d'une base de données (sans utiliser du php ou du SQL). Durant cette mission j'ai fait face à de nombreux obstacles, erreurs... et j'ai donc fait énormément de recherches afin de comprendre ce qu'il ne marchait pas.

MGC | Connecting bénéficie aussi maintenant d'un nouvel outil de travail qui leur permet d'alimenter la base de données de manière rapide et 100% automatique.

VI. Interfacer les données : visualisation via une page web

1. Présentation de la mission.

Comme dit précédemment, l'objectif de tout ce projet est de permettre une navigation plus fluide et plus simple dans toutes ces données de facturations et cela, que ce soit pour le client ou bien l'entreprise. L'idée la plus simple pour arriver à ce résultat, est de créer une page web complète permettant à l'utilisateur de choisir précisément les données qu'il veut voir.

Après avoir passé la plupart du temps de mon stage à effectuer du travail considéré comme BACK-END, je m'attaque au développement FRONT-END. Sur cette nouvelle partie la du travail, je serai accompagné du développeur junior, Hermann. Après avoir vu avec lui, le mieux afin que mon travail se coordine correctement avec celui de l'entreprise déjà en place, est que je code en utilisant la méthode MVC (Model, Views, Controllers). Cette méthode est une que je connais beaucoup car c'est évidemment celle que nous avons commencé à étudier en début d'année à l'IUT. De plus, nous sortions d'un projet SAÉ dans lequel cette méthode était la colonne vertébrale du projet.

Dans ce travail j'ai utilisé beaucoup de mes connaissances. En effet, le PHP était le langage prédominant dans cette mission. Afin d'utiliser à bien le langage PHP, il faut également suivre avec le HTML et le CSS. J'ai donc utilisé toutes ces connaissances pour l'élaboration de ma page web. Enfin, afin de rendre la page web plus dynamique, j'ai utilisé du javascript. Grâce à mes souvenirs de cette matière en BUT1, j'ai pu rendre cette page plus attractive, plus fluide et dynamique.

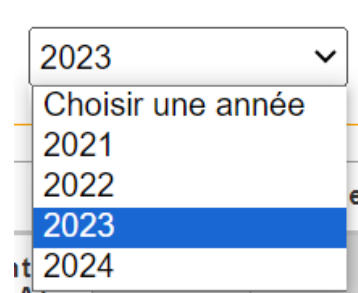
2. Réalisation de la mission.

Le travail en FRONT-END signifie que c'est le travail présent devant, ce que l'utilisateur verra. Pour cette mission il fallait allier efficacité, fluidité mais aussi esthétisme. Afin de correspondre et de corréler à l'image de l'espace client de base, j'ai d'abord créé une maquette (*cf annexe 5.*) sur laquelle je m'inspire du véritable espace client. J'ai ensuite basé mon travail sur cette maquette.

Pour commencer, j'ai recréé l'accueil de base qui est présent sur le site (*cf annexe 6.*). Bien que facultatif, j'aime bien lors d'un projet, qu'il soit fait de A à Z. J'ai donc recréé le parcours qu'un utilisateur pourrait faire s'il voulait voir la nouvelle fonctionnalité, celle que je suis en train de développer. Une fois que la copie du site a été créée, j'ai donc commencé la "View" de ma page web (*cf annexe 7.*), en me basant sur la maquette que j'avais imaginée. J'ai donc réutilisé les codes couleurs de MGC, le noir et le orange. J'ai commencé par reprendre la structure de base qu'on a fait pour la saé pour l'organisation de mes fichiers (*cf annexe 8*). Je suis parti sur une vue/view nommée "detail_factu" et j'ai donc accordé avec ceci le controller. J'ai également une page de javascript qui se trouve dans le sous-dossier Js de Contents. Tous ces fichiers me permettaient de pouvoir créer une bonne page fonctionnelle, sans oublier le css pour l'esthétisme.

D'un point de vue technique, j'ai voulu optimiser en rajouter la navbar dans mon fichier "view_begin.php" avec son css associé, que j'appelle au tout début de ma page "view_detail_factu.php". De même pour le footer qu'on peut retrouver sur la page, je l'ai placé dans le fichier "view_end.php" que cette fois j'appelle à la fin afin de cloturer mon code.

L'idée de base est que nous puissions avoir accès aux données qu'on a rentrées dans notre table avec notre script et ce, de manière claire et précise.



Pour commencer, la page web est dédiée au service interne de MGC. Je voulais créer une page qui était simple, alors j'ai opté pour que la page s'affiche avec uniquement 2 menus de sélection, celui du choix du client, ainsi que celui de l'année. Étant donné que la page est pour l'instant adaptée pour le service interne de MGC, le choix des clients peut se faire avec tous ceux présents dans la base.

En revanche, si on est dans l'espace client, en tant que client et qu'on est connecté, alors là, le choix du client n'est évidemment pas possible. Les années qui sont affichées ici s'actualisent automatiquement. Cela signifie qu'à chaque passage à une année supérieure, le code ajoute automatiquement le choix de faire un choix pour la nouvelle année. Cela est dû au fait que les options des années ici présentes sont définies grâce à cette commande :

```
SELECT DISTINCT AnneeFacturation FROM  
[REPORTING].[dbo].[ListeApresFacturation] ORDER BY AnneeFacturation
```

et non saisi manuellement. Afin de rendre cette page dynamique, j'ai ajouté une fonctionnalité qui est entièrement gérée avec mon fichier javascript, celle de préciser la période en fonction de l'année choisie.

En effet, si dans le menu de sélection “Année”, une année est choisie, alors un autre menu apparaît. Ce second menu de sélection permet à l'utilisateur de préciser la période sur laquelle il veut prendre connaissance de données. L'utilisateur peut choisir de voir les statistiques sur l'année complète, ou bien d'uniquement voir des informations sur une temporalité encore plus précise. Dans le nouveau cas où une option autre que “Année entière” serait choisie dans ce menu, alors un dernier menu apparaît, et celui-ci change en fonction de ce que l'utilisateur aurait choisi comme temporalité.

Par exemple, si l'utilisateur choisit “TRIMESTRIEL”, il pourra donc préciser le trimestre dont il veut prendre connaissance. Comme on a vu précédemment, c'est ici qu'intervient l'importance de ce qu'on avait prévu auparavant, le rajout des champs Trimestre/Semestre/Mois/Année de la Facturation/Parution. Grâce à ces champs là, la fonctionnalité de recherche par temporalité précise, est

énormément facilitée.

Tous ces menus sont codés dans un formulaire. Une fois que l'utilisateur a fait tous ses choix, il n'a plus qu'à appuyer sur le bouton “Rechercher” afin que les données s'affichent dans un tableau (cf annexe 7.).

Lorsque le formulaire de recherche est soumis, on appelle le controller, qui permet de voir quels ont été les choix faits par l'utilisateur. Une fois tout cela défini, on appelle les fonctions présentes dans le Model (cf annexe 9.). Une autre fonctionnalité que j'ai rajoutée afin de rajouter de la visibilité à l'utilisateur, c'est de pouvoir afficher le détail financier. En effet, pour l'instant l'utilisateur ne peut que voir des chiffres purs. L'idée est tout de même d'avoir le plus d'informations disponibles. Alors en + afficher détail financier. dessous du tableau (cf annexe 7.) j'ai rajouté un bouton permettant d'afficher un autre tableau dans lequel se trouve toutes les factures qui composent les calculs affichées initialement. Une fois ce bouton appuyé, nous voyons un tableau qui liste toutes les factures.

- cacher détail financier.

Client	Secteur	Date de Facturation	Ventes (CA) en €	Achats en €	Marge en €	Date de Parution	Numero de la Facturation
	PAFIG	2023-10-31	.00	.00	.00	2021-03-27	31277255
	PAFIG	2023-10-31	-114.00	.00	-114.00	2021-10-02	31277256

Une dernière fonctionnalité que j'ai rajoutée, en plus de pouvoir regarder les factures avec les calculs de

ventes et d'achats, c'est de pouvoir filtrer les recherches par secteur. En effet, MGC effectue ses ventes auprès de ses clients sur différents secteurs. Il est alors important pour l'interne de MGC de pouvoir filtrer ces recherches sur les différents secteurs. Cela permet donc de voir quel secteur génère plus d'argent ou autre.

3. Bénéfice de ce travail.

Ce travail là était particulièrement différent des autres. Moi qui ne suis pas un adepte de la programmation web, j'ai bien aimé travailler sur cet aspect là. Cette mission m'a permis de consolider et de renforcer énormément mes compétences en PHP. J'avais besoin de faire de la méthode MVC car j'estimais mon niveau faible dans cette méthode là. Je suis dorénavant satisfait de mon travail et du niveau que j'ai pu prendre durant la période de travail sur cette page web. En plus du PHP, j'ai eu l'occasion de refaire du CSS, un langage purement esthétique dont on oublie trop souvent l'importance et la complexité. J'ai pu gagner en connaissance sur ce langage là. Enfin, le langage Javascript que j'ai beaucoup utilisé surtout pour rendre le côté dynamique à mon site, m'est vite revenu et j'ai pu également m'améliorer sur ce langage là.

Une qualité en plus de ce travail là, c'est que j'ai dû particulièrement être attentif aux besoins suggérés par Bruno, mon maître de stage. Être à l'écoute des personnes qui font appel à mes services est un aspect très important de mes études.

VII. Gain en efficacité.

1. Meilleure gestion d'organisation.

Le monde de l'entreprise et celui des études à l'IUT, est totalement différent. Le point commun entre ces deux endroits est l'importance de l'organisation. En effet, sans une bonne organisation, on se perd très facilement.

Une première organisation essentielle, c'est l'adaptation aux processus existants. Ce que j'ai fait pendant ce stage a dû forcément être en adéquation avec ce que MGC avait déjà mis en place. Voilà la raison pour laquelle j'ai rédigé mon script de création de table dans la même forme que les autres tables déjà inscrites. Cela va de soi pour le script d'auto-alimentation de la table. Je l'ai rédigé en langage C# afin que cela colle avec la plupart du code qu'on peut retrouver dans le service

informatique de MGC | Connecting. Enfin, pour le site web, j'ai dû m'adapter aux codes couleurs de l'entreprise afin que cela s'intègre parfaitement. J'ai pu facilement m'adapter tout au long du stage, notamment grâce à l'aide des deux développeurs qui m'ont accompagné.

La gestion du temps c'est une aussi une notion très importante et primordiale de la vie en entreprise. Le statut de stagiaire aurait pu négliger cette partie là, mais quoi qu'il arrive, il faut savoir où on en est par rapport au temps qu'on dispose. J'essayais alors de gérer mon temps, de savoir combien de temps une tâche allait me prendre, mais aussi dans combien de temps j'aurais fini.

Enfin, quel que soit le projet, le domaine ou l'endroit, il peut toujours y avoir des problèmes qui impliquent de la flexibilité ou de l'adaptabilité. Je pense que durant l'entièreté de ce stage, j'ai pu améliorer mon organisation dans ces 3 notions là et ce, pour les raisons énoncées.

2. Meilleure communication.

Bien que j'ai effectué le travail globalement seul, j'ai tout de même fait partie d'une équipe. Nous étions donc 3 (en plus de moi) dans le service informatique. Étant donné que j'ai pour chaque tâche était accompagné d'un des deux développeurs, il était important que je tiens au courant de mon avancée à l'un d'entre eux. La communication était donc importante pendant ces 10 semaines de stages. J'ai fait l'effort durant ce stage de venir assez régulièrement vers les développeurs confirmés afin de demander des conseils, ou bien faire des suggestions. J'ai essayé d'apporter une collaboration active. En plus de prendre la parole, l'écoute est d'autant plus importante, encore plus quand elle provient de développeurs confirmés. A chaque fois qu'on me parlait je faisais preuve d'une écoute attentive et active.

Je pense que durant ce stage mon organisation et ma communication ont été améliorées. Le fait d'être dans le cœur du métier m'a poussé à faire de mon mieux et donc de m'améliorer de jour en jour.

VIII. Conclusion.

1. Résumé.

Après 10 semaines de stage avec 30 jours sur site et 20 en télé-travail, j'ai pu effectuer mes 3 principales tâches présentées ci-dessus. J'ai pu faire des missions complètement différentes les unes des autres mais qui pourtant étaient reliées :

La **table SQL** a mené à un **script d'auto-alimentation** qui a mené à une **interface web**.

L'objectif du projet que nous avons mis en place a été avancé voir même touché. Nous avons atteint le but final, bien qu'il soit toujours possible de le compléter encore plus et donc nous ne pouvons pas considérer le travail complet fini.

Durant ces 10 semaines, j'ai pu améliorer énormément mes capacités. J'ai commencé à apprendre le langage C#. J'ai solidifié mes compétences en développement web. D'un point de vue non-technique, j'ai pu améliorer ma communication, mon organisation.

2. Ressenti personnel.

Quand j'ai commencé le stage, je m'attendais à revivre plus ou moins une situation assez similaire à mon stage de troisième : c'était ma seule référence. Quand je suis arrivé, c'est ce que j'ai ressenti, des open spaces du même genre, des salles de réunions similaires...

Le lieu était bien situé, les locaux étaient spacieux (*cf annexe 10.*). L'équipe de MGC était peu nombreuse, mais j'ai trouvé ça meilleure en termes de conditions de travail plutôt que d'avoir de grands services avec des dizaines de personnes. J'étais la douzième personne à venir travailler ici. Tout le monde était très sympa et je trouve que l'ambiance de travail était très agréable. On était dans un mélange de bonne humeur, de détente, mais également de sérieux et assiduité, je trouve que c'est la méthode de travail la plus agréable, et celle qui donne le plus envie de travailler.

Pour ce qui est de mes missions, j'ai beaucoup apprécié pouvoir toucher à tous types de développement (BACK et FRONT - END). J'ai trouvé le projet particulièrement intéressant et le fil rouge de mon stage, l'enchaînement de mes missions avait un sens. Je pense que j'aurais moins aimé si on m'avait dit de coder des travaux qui étaient "éparpillés", sans que tout le fruit de mon travail n'ait aucun sens.

Je suis également très satisfait de mes efforts. Tout ce que j'ai pu développer pendant ce stage est l'image exacte de ce que j'avais en tête et de ce que j'ai voulu faire.

J'ai beaucoup aimé travailler en entreprise. Je pense que j'ai été plus motivé qu'à l'IUT et donc cela m'a poussé à apprendre. Le fait d'être au cœur de véritables projets à enjeux, me donne plus envie de fournir de l'effort.

3. Orientation (BUT3).

Pour l'année prochaine, j'aimerais bien décrocher une alternance. En effet, comme dit précédemment, le travail en entreprise, l'environnement professionnel, en plus de l'IUT, m'a particulièrement épanoui.

J'apprends beaucoup à l'IUT, je pense que le fait d'être entouré de développeurs confirmés au sein d'une entreprise avec des projets à enjeux, me donne encore plus l'envie d'apprendre.

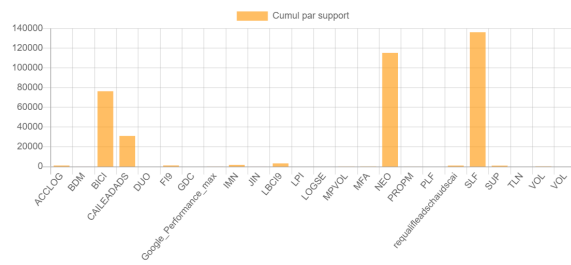
C'est pour ça que je pense essayer de trouver une alternance pour continuer ma troisième année d'études afin de pouvoir apprendre le programme basique de l'université, mais aussi pour les tester en entreprise par la suite.

Je suis déjà en train de voir avec MGC pour une éventuelle alternance, car je pense qu'il serait une bonne idée de continuer sur cette belle lancée que nous avons commencé pendant ces 2 mois.

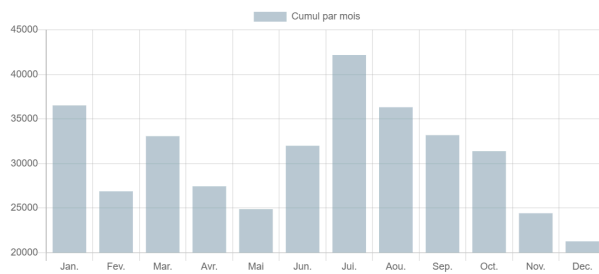
IX. Lexique.

- *Portail* : Site internet sur lequel sont déposées les offres des annonceurs, des clients.
- *Leads* : Communication directe ou indirecte entre le client et ses clients (autrement dit, entre les clients de MGC et les internautes des portails).
- *Biens* : Maison, terrain ou autre à vendre sur un portail, donné par les clients.
- *Lots* : Groupement de biens qui possèdent des caractéristiques similaires.

X. Annexes.



1.



2.

```

Programme X
class Program
{
    // ...
    static void DownloadFileFromFtp(string ftpHost, string ftpUser, string ftpPassword, string ftpFolderPath, string localPath, string ftpArch)
    {
        // ...
    }
    static void MoveFileToFolderArchive(FtpClient ftp, FtpListItem file, string ftpArchivePath)
    {
        // ...
    }
    static List<string> GenerateInsertCommandsFromCsv(string csvFilePath)
    {
        // ...
    }
    static string NormalizeCurrency(string value)
    {
        // ...
    }
    static void ProcessCsvLine(string[] headers, string[] values, List<string> insertCommands, int lineNumber)
    {
        // ...
    }
    static string FormatIsoValue(string value, string header)
    {
        // ...
    }
    static string FormatValue(string value, string header)
    {
        // ...
    }
    static int GetMonthFromDate(string data, int lineNumber)
    {
        // ...
    }
    static int GetYearFromDate(string data)
    {
        // ...
    }
    static int GetDayOfMonthFromDate(string data)
    {
        // ...
    }
    static int GetQuarterFromDate(string data)
    {
        // ...
    }
    static string GetValue(string key, string[] infos)
    {
        // ...
    }
    static void InsertIntoSqlServer(string connectionString, List<string> commands)
    {
        // ...
    }
    static void DeplacerDernierFichierArchives(string directoryPath, string archiveDirectory)
    {
        // ...
    }
    static void DeleteFile(string filePath)
    {
        // ...
    }
}

```

3.

```

class Program
static void Main(string[] args)
{
    // Extrait toutes les informations pour se connecter du fichier FTP_settings.
    string[] infosFTP = File.ReadAllLines(@"C:\PROGRAM_facture_FTP_to_SQLSERV\app2\FTP_settings.txt");
    string ftpUser = GetValue("User", infosFTP);
    string ftpPassword = GetValue("Password", infosFTP);
    string ftpHost = GetValue("Host", infosFTP);
    string ftpFolderPath = GetValue("FTP_FilePath", infosFTP);
    string localPath = GetValue("Local_OW_PATH", infosFTP);
    string ftpArchivePath = GetValue("FTP_ArchivePath", infosFTP);
    DownloadFileFromFtp(ftpHost, ftpUser, ftpPassword, ftpFolderPath, localPath, ftpArchivePath);

    string csvFilePath = Path.Combine(localPath, "ListeAF.csv");
    List<string> insertCommands = GenerateInsertCommandsFromCsv(csvFilePath);

    // Extrait toutes les informations pour se connecter du fichier SQL_settings.
    string[] infosSQL = File.ReadAllLines(@"C:\PROGRAM_facture_FTP_to_SQLSERV\app2\SQL_settings.txt");
    string NomServeur = GetValue("NomServeur", infosSQL);
    string NomBaseDeDonnees = GetValue("BaseDD", infosSQL);
    string Connexion = GetValue("Connexion", infosSQL);
    string Password = GetValue("Password", infosSQL);
    string connectionString = $"Data Source={NomServeur};Initial Catalog={NomBaseDeDonnees};User Id={Connexion};Password={Password};";

    InsertIntoSqlServer(connectionString, insertCommands);
    DeleteFile(Path.Combine(localPath, "ListeAF.csv"));
    string archiveDirectory = Path.Combine(localPath, "Archives");
    DeplacerDernierFichierArchives(localPath, "Archives");
}
}

```

4.

6. **comparatif : vrai site**

7. mon site

8 [+ afficher détail financier.](#)

```
9 | view_home.php
  | index.php
```

10.

```
public function getAllNomClientsMGC() { ...  
}  
public function getAllYear() { ...  
}  
public function getAllSecteurByClient($name, $year, $month = null,  
}  
public function getAllSecteur() { ...  
}  
public function getAllInfoByTempo($name, $year, $month = null, $se  
}  
public function getCAByTempo($name, $year, $month = null, $semestr  
}  
public function getAchatsByTempo($name, $year, $month = null, $sem  
}  
public function getCAGroupByMonth($name, $year, $month = null, $se  
}  
public function getAchatsGroupByMonth($name, $year, $month = null,  
}
```

11.

