

Rapport de stage: **stage de 2eme année de BUT INFO**

**Mise en place d'un outil de
monitoring.**



Effectué par :

Abelard Rodlens

Fait du 24/01/2024 au 19/03/2024

Encadré par :

-Tuteur en entreprise : Demri Rayan, responsable dev

-Enseignant responsable : Bjerrum Marie

Sommaire

.....	1
Remerciements.....	2
Introduction.....	2
I Présentation de l'entreprise	3
I.I Avistel	3
I.II Secteur d'activités.....	3
I.III Implantation géographique	4
I.IV Chiffre d'affaires.....	4
I.V Structure organisationnelle	5
I.VI Evolution et Croissance	6
I.VII Produit phare et services	6
I.VIII Infrastructures	9
I.IX Exigences	10
II Service d'affectation	10
II.I Service R&D	10
II.II Environnement technique et ressources	11
II.III Environnement humain	11
III Projet accomplie.....	11
III.I Présentation générale	11
III.II Environnement	11
III.III Problématique	11
III.IV Raisons.....	12
III.V Objectifs.....	12
III.VI Exigences Fonctionnelles.....	12
III.VII Processus de réalisation	13
III.VIII Compétences mobilisées	21
III.IX Gestion du temps.....	22
IV Conclusion	22
V Table des illustrations.....	23
Annexes.....	23

Remerciements

Tout d'abord avant d'aller plus loin dans mon rapport, je tiens à exprimer ma gratitude envers tout le corps enseignant de l'IUT de Villetaneuse qui m'a accompagné durant ces 3 semestres de BUT, leurs enseignements et leurs soutiens ont été essentiels pour la réussite de mon projet. Je souhaite adresser mes plus sincères remerciements à mon tuteur de stage Mr Demri Rayan, pour m'avoir accompagné durant ces 2 mois de stage, il m'a donné de précieux conseil qui assurément me serviront tout au long de mon parcours professionnel, ainsi que dans la vie en général. Je remercie également Mr Kneller, ainsi que Mr Germain de m'avoir si chaleureusement accueilli dans leur entreprise. Enfin je remercie mon tuteur académique Mme Bjerrum, pour son suivi attentif et ces précieuses directives qu'elle m'a fournies tout au long du stage. Leur soutien et enseignement ont grandement contribué au bon déroulement de mon stage.

Introduction

Etant actuellement en 2ème année de Bachelor Universitaire de Technologie au sein de l'institut universitaire de technologie de Sorbonne Paris Nord, j'ai récemment eu l'opportunité de réaliser un stage de 2 mois au sein de la société Avistel en tant que Développeur. Ce stage vise à nous offrir une première expérience professionnelle dans le milieu de l'informatique, nous permettant ainsi d'appliquer les connaissances que nous avons durement acquis durant les 3 premiers semestres de notre formation, tout en nous préparant pour les semestres à venir.

Avistel est une société opérant dans le domaine des solutions multimédias, dans divers secteurs d'activité, tel le domaine de la santé, de la justice et des plateformes pétrolières. De plus elle possède une filiale nommée 4K France spécialisé dans l'exploitation multimédia, cette filiale est chargée du recrutement et de la gestion du personnel, de la commercialisation des abonnements TV et Wifi, et gère plusieurs hôpitaux en France, dont 2 CHU, 1 GHT et 2 AP-HP.

Dans le cadre de ce stage, j'ai eu pour mission principale de consolider les différentes bases de données de la solution AVITAX (solution de taxation). Dans ce but j'ai travaillé sur la conception d'un outil de monitoring permettant aux clients d'Avistel, plus particulièrement ceux se situant dans le secteur de la santé. Cet outil permet aux clients de visualisés différentes données liées à leurs différents sites, telles que le chiffre d'affaires selon une

période et/ou un site donné, ou encore la possibilité de visualiser le service le plus et/ou le moins rentable.

Comment mené à bien la réalisation de cet outil par le biais des compétences acquises durant mes années de formations ?

Dans la suite de ce rapport, vous trouverez une présentation approfondie de l'entreprise qui m'a accueillie, ainsi qu'une description des missions du service auquel j'ai été affecté. Ainsi qu'une présentation détaillée de mon projet, dans lequel figurera les différentes étapes qui m'ont permis de le réaliser. Puis nous conclurons par l'expérience que j'ai acquis durant ce stage et mes perspectives d'évolution/orientation futures.

Présentation de l'entreprise

Avistel

Avistel est une entreprise détenue par Marc Germain et David Kneller qui opère dans le domaine des solutions multimédias depuis plus de 20 ans en France, offrant ses services à divers secteurs tels que les établissements pénitentiaires, les hôpitaux, les plateformes pétrolières, les écoles et même les chaînes de restauration.

Secteur d'activités

C'est une entreprise qui opère principalement dans le secteur des solutions multimédias, elle fournit des services et des technologies qui sont adaptés à différents secteurs d'activités.

Ces secteurs d'activités sont :

La santé :

Avistel fournit de nombreuses solutions multimédias aux établissements de santé, comme les hôpitaux, ou encore les centres hospitaliers universitaires. Parmi ces solutions proposées par la société figure la solution de divertissement AVIMOVIE, qui peut inclure des services tels que MyCanal, et Bein Sport pour divertir les patients. Également il leur propose des solutions tels que AVICOM permettant de gérer les systèmes de communication, et de gestion de la sécurité de leurs infrastructures.

La justice :

Ils fournissent aussi leurs solutions multimédia aux établissements pénitentiers, plus précisément les établissements tels que les centres de détentions, le ministère de la justice

et les prisons. Tout comme le secteur de la santé les solutions proposées par Avistel à ces établissements peuvent contenir des solutions de divertissement, de communication, ou encore de surveillance pour assurer un bon déroulement dans ces établissements.

L'éducation :

Avistel propose de multiples solutions pour les établissements scolaires qui visent à améliorer l'apprentissage des étudiants. En proposant des outils d'affichage permettant aux enseignants de projeter leurs enseignements, ou encore de permettre à l'établissement d'améliorer son système de communication.

Plateformes pétrolières :

Ils fournissent également des solutions multimédias aux entreprises opérant dans ce secteur, les solutions fournies sont principalement celles de divertissement pour les employés, ainsi que les solutions de communication et de surveillance pour assurer la sécurité et la productivité.

Restauration :

Leurs solutions sont aussi proposées aux chaînes de restauration rapides, il propose plus particulièrement leurs solutions d'affichage permettant à ces établissements de pouvoir afficher leurs différents menus, ou de tout simplement proposer du divertissement à leur clientèle.

En résumé, Avistel fournit ces solutions à tous types d'établissement, tels que les établissements de santé, de justice, d'éducation ou encore de restauration en mettant l'accent sur la communication, la sécurité, ainsi que la gestion et le divertissement que peuvent apporter leurs solutions.

Implantation géographique

Son siège social se situe en région parisienne, plus précisément à Houilles-Carières-sur-Seine, à environ 10 minutes de la gare du même nom. De plus, Avistel mène des activités dans plusieurs pays comme la Belgique, l'Angleterre et l'Espagne.

Chiffre d'affaires

Durant l'année 2023 le chiffre d'affaires de la société Avistel s'étend à 4.6 millions d'euros, soit une augmentation de 4.5%. Avec leur filiale 4K France, le chiffre d'affaires de la société monte jusqu'à 6.1 millions d'euros, ce qui représente une augmentation de celui-ci de 22%. Ce chiffre d'affaires représente l'approvisionnement de plus de 100 000 écrans, dont 270

établissements de santé équipés, ainsi que 9 plateformes pétrolières, et 94 prisons dont 56 se situant en France, et 38 autres en Belgique.

Structure organisationnelle

Avistel opte pour une structure organisationnelle hiérarchique classique, divisé en 2 arbres (cf Annexe 1). Le premier arbre est dirigé par le président Mr Germain Marc, dans lequel est présente trois branches.

Ces branches sont :

Commerce :

Ce service comprend toutes les activités commerciales, y compris la gestion des ventes, des relations clients, des partenariats commerciaux, etc.

R&D :

Ce service est dédié à la recherche et au développement logiciel, ce qui implique la création et la maintenance des solutions logicielles de l'entreprise.

Informatique :

Ce service regroupe toutes les activités liées à son domaine d'activité, notamment la gestion des systèmes informatiques, la maintenance du réseau, le support technique aux utilisateurs, etc.

Le second arbre est dirigé par le Directeur Général Mr Kneller David, dans lequel on retrouve 2 branches.

Ces branches sont :

Opérations :

Ce service supervise et coordonne les activités opérationnelles de l'entreprise. Plus précisément il gère les processus de production, la planification des ressources, ainsi que la gestion logistique et le maintenant des équipements.

Administration :

Cette branche est exclusivement dédiée à la comptabilité et à toutes les tâches administratives telles que la gestion des contacts, des comptes bancaires, des rendez-vous, etc. Soit toutes les activités typiques d'un service administratif classique.

Evolution et Croissance

A l'heure actuelle je n'ai aucune information sur l'évolution et la croissance de l'entreprise.

Produit phare et services

Avistel propose un service de location de télé, mais ce n'est pas ce service en lui-même qui fait la force de la société, ou qui intéresse la clientèle, mais ce sont tous les services développés par Avistel qui sont proposés ou non avec la location de télé. Parmi ces services proposés par l'entreprise on retrouve :

Avipay :

La solution de paiement en ligne sécurisé, compatible avec tout type d'appareil. Elle offre la possibilité à la clientèle d'Avistel d'offrir ces services de façon autonome 24h/24 et 7j/7 sous forme d'abonnement. Par le biais de la solution GTVX intégré à AVIPAY, il offre la possibilité de gérer les abonnements des services TV et compte proposé, tout en ayant la possibilité de consulter ces statistiques de vente.

Avicom :

Service d'affichage proposé par l'entreprise, elle permet de créer, diffuser et gérer du contenu multimédia. Il offre également la possibilité d'importer du contenu multimédia de différents types tels que PPT, PDF. Ou encore il permet de planifier le contenu diffusé selon un horaire prédéfini. De plus il offre la possibilité de tracer les actions effectuées et de gérer la diffusion de ces appareils à distance.

Avicontrol :

Service de gestion de réseau coaxial ¹, il a pour but de simplifier et centraliser la gestion des droits d'abonnement des télévisions fournis par Avistel via un réseau coaxial. C'est une solution qui permet de contrôler l'accès aux chaînes de télévision. Il permet également de gérer les abonnements et les droits de la télévision, via l'outil d'administration.

Avidev :

Service de développement de solutions personnalisées, il permet de faire faire des produits personnalisés répondant à l'environnement du client. L'entreprise propose la personnalisation de Tablettes, Téléviseurs, ou encore de Terminaux conçus par Avistel eux

même. Mais encore Avidev offre la possibilité de faire personnaliser le logiciel du composant par leurs développeur.

Avilock :

Service de sécurisation matérielle et logicielle, il garantit la sécurisation logiciel que ce soit sur site ou à distance, ainsi qu'une sécurisation matérielle prévenant des vols. Il assure également une compatibilité avec tous systèmes d'exploitation ², et appareils provenant ou non d'Avistel. De plus il propose des fonctionnalités tels que l'encapsulation de sites Web dans des applications, la gestion des applications des appareils, dont la possibilité d'ajout d'applications interne, d'autre part il propose la gestion de profils, le chiffrement de données, ou encore la possibilité de choisir le lieu dans lequel on souhaite installer le service.

Avilone :

Service IPTV permettant d'accroître ces offres de contenu, ainsi que d'améliorer l'intégration avec ces autres solutions. Il offre la possibilité de gérer les droits des abonnements de télévision via le réseau IP ³, ainsi que la capacité gérer son flotte multimédia ⁴ via se même réseau, et la possibilité d'interagir avec le service de facturation via une infrastructure nouvelle ou non. Avilone propose également de nombreux services utilisateurs, tels que la HD en direct, ainsi que la Vidéo à la demande, ou encore la Définition de Musique. Enfin des services back-office ⁵ offrant la possibilité de faciliter la gestion de son réseau IP grâce à de nombreuses fonctionnalités sont pareillement disponible.

Aviteach :

Service de multidiffusion IP conçue pour répondre aux besoins des salles de classe modernes. Il offre une plateforme intuitive qui permet aux intervenants de gérer facilement les flux ⁶ IP dans leur environnement d'enseignement. Il simplifie la diffusion et la réplique de contenu dans les salles de classe par le biais ces fonctionnalités offrant une gestion du flux IP simplifié. Il permet aussi de gérer une multitude d'appareils connecté tels que les projecteurs, les ordinateurs, les haut-parleurs, ou encore les microphones, qu'il soit fourni ou non par l'entreprise et tout ça via le réseau IP existant. Enfin le service promet une simplification de la gestion de l'affichage durant les cours, offrant ainsi une meilleure expérience d'apprentissage/d'enseignement aussi bien pour les élèves, pour que les enseignants.

Avimap :

Service de navigation indoor avancée, bien que similaire au service Avilone dans son approche. Il propose une plateforme complète permettant de guider de manière efficace les visiteurs des structures. De plus il permet aux entreprises de créer des plans interactifs sur des terminaux tactiles fournis par Avistel. Aussi il propose la conception de solution sur mesure adapté au mieux à l'environnement des clients, d'autre part un suivi est assuré, ainsi qu'un service après-vente. En résumé AVIMAP est un service permettant d'optimiser l'expérience des visiteurs et l'efficacité des entreprises.

Aviplay :

Service Android pour la gestion du contenu sur les écrans tactiles des établissements de santé. Il permet de diffuser et de classer des chaînes TV HD et des stations de radio numériques, tout en offrant une personnalisation avancée de l'affichage. De plus en ajoutant aisément des contenus variés tels que des films, des livres électroniques et des jeux, le service enrichit l'expérience des utilisateurs. Avec sa gestion des prix et des paiements flexibles, il offre aussi la possibilité de régler en ligne, à l'accueil ou directement sur le terminal tactile. D'autre part ses outils de présentation des données, de gestion des factures et des comptes utilisateurs permettent un contrôle précis et complet. Son système dédié aux techniciens assure un fonctionnement optimal et sécurisé, conforme aux normes en vigueur. Enfin AVIPLAY est vendu comme un service indispensable pour divertir et informer les résidents des établissements de santé.

Aviservices :

Service de support et de services qui offre une assistance complète de l'installation à la maintenance. Il permet de bénéficier d'un accompagnement personnalisé répondant à tous besoins technologiques. Le service permet également au client d'identifier ces besoins afin de trouver la solution la plus adéquate. Tout ça via l'analyse technique de leur système effectué par leurs techniciens dans le but de proposer de nouveaux services. Aussi l'équipe de gestion assure la mise en place des services, ça inclut les intégrations de systèmes avec nos partenaires. D'autre part le service garantit la réussite des opérations d'installation, incluant tous les aspects depuis les serveurs et réseaux jusqu'aux terminaux utilisateurs.

Enfin Aviservices fournit une documentation complète pour accompagner dans l'utilisation et la maintenance des solutions technologiques.

Lexique :

- ¹ Réseau de télécommunication utilisant un câble coaxial pour transmettre des données.
- ² Ensemble de programmes permettant le fonctionnement et le contrôle d'outils informatiques.
- ³ Réseau informatique utilisant le protocole IP dans le but de transmettre des données entre des appareils connectés.
- ⁴ Désigne une collection d'appareils.
- ⁵ Partie d'une entreprise, gérant les opérations internes et les processus administratifs.
- ⁶ Fais référence à un transfert de donnée.

Infrastructures

Avistel dispose d'un grand espace de travail pour une entreprise de cette taille. L'espace est découpé en 2 parties, dont l'espace de travail de bureau (cf Annexe 2). Tout d'abord les bureaux sont de taille moyenne, ils sont tous équipés d'ordinateur et de chaise de bureau ergonomique de grande qualité afin de préserver le dos des employés. De plus dans cette espace de travail il y a un espace pour les informaticiens, dans lequel on y trouve un plan de travail, des installations réseau, ainsi que des cartes réseau et d'autres outils dépassant mes connaissances. Le second espace de travail est l'atelier (cf Annexe 3) dans lequel on retrouve les appareils défectueux, mais également un grand plan de travail et une multitude d'outils. Je tiens à préciser qu'il est bien évidemment équipé de chaise. Sinon pour ce qui est de l'agencement des bureaux la société a opté pour des bureaux partagés, ce qui permet d'être dans un environnement de travail collaboratif, mais les bureaux partagés n'est pas la seule raison. Il faut savoir que les bureaux sont très proches les uns des autres, en plus de ça ils sont équipés de devanture en verre ce qui permet de faciliter la communication ou encore de voir si chaque membre de l'équipe est opérationnel.

Les locaux disposent également d'appareils permettant de prendre des boissons chaudes comme froides, ainsi que d'une salle à manger/détente. Dans cette salle on y retrouve les appareils qu'on retrouve couramment dans les cuisines, soit un frigo, un micro-onde, également un lavabo et de la vaisselle. Sinon de l'autre côté de la pièce se trouve une table accompagnée de chaises et d'une télévision. Il y a aussi une salle de réunion assez classique.

Exigences

Pour ce qui est des exigences vestimentaires chez Avistel, il n'y a pas vraiment d'exigence particulière. L'entreprise privilégie le confort, laissant à chacun la liberté de s'habiller comme il le souhaite. On peut ainsi voir des personnes en jogging, comme en cargo, ou encore pantalon ou encore en jean pour ce qui est du bas, et le choix est tout aussi varié pour ce qui est du haut. Cette liberté est sûrement due au fait que l'entreprise est principalement composée de personnes assez jeunes.

En ce qui concerne la communication verbale, aucune exigence particulière n'est imposée, tant que tout se déroule dans le respect mutuel. Le vocabulaire est varié, bien que le langage familier soit beaucoup plus fréquent. C'est un environnement dans lequel tout le monde se tutoie, créant ainsi une atmosphère familiale. De plus, la plupart des employés sont appelés par un surnom.

Service d'affectation

Service R&D

J'ai été affecté au service Recherche et Développement qui a pour objectif principal le développement et l'entretien des solutions logicielles de l'entreprise. Nous collaborons principalement avec les équipes commerciales, l'atelier, la direction ainsi que la gestion de projet, afin de répondre au mieux aux besoins de l'entreprise. De plus nous sommes également en contacts Directs avec nos fournisseurs dans le but de trouver les matériels les plus adéquats.

De plus nous sommes engagés dans plusieurs projets simultanés, avec comme mission de rendre nos solutions disponibles sur notre site. En tant que service R&D, nous avons pour mission de créer des solutions efficaces et fonctionnelles, puis de les rendre moins onéreux. Enfin c'est un service qui a des échéances strictes à respecter.

Environnement technique et ressources

Concernant notre environnement technique et nos ressources, nous disposons d'une grande variété de matériels informatiques, ce qui comprend des serveurs, des Switches, des Environnements de Virtualisation (ESXI), des ordinateurs portables, des Workstation, etc. Nous adoptons une méthodologie agile dans nos méthodes de travail, en utilisant les documents techniques fournis par les constructeurs et éditeurs.

Environnement humain

Notre équipe est dirigé par Mr Demri notre responsable développement. Nous collaborons également avec des freelances venant des quatre coins du globe. De plus, nous disposons aussi d'un bureau dédié ainsi que d'un environnement de test, qui comprend une baie et une zone de préproduction pour tester nos solutions avant de les déployer.

Projet accomplie

Présentation générale

La société Avistel cherche à mettre en place un outil de monitoring pour ses clients, plus particulièrement ceux travaillant dans le secteur de la santé. Il doit utiliser les bases de données de la solution de taxation Avitax.

Environnement

Le projet est conçu pour les clients du secteur de la santé, principalement des hôpitaux. Ces établissements sont destinés à prendre en charge des patients de tout âge et problèmes de santé. C'est un environnement qui dispose de moyen de surveillance, afin d'assurer la sécurité des patients. De plus il dispose également de moyen de communication, et de divertissement tels que les télévisions permettant aux patients d'accéder à l'information ou de tout simplement se divertir.

Problématique

L'entreprise fait face à un problème majeur, qui est de savoir comment il pourrait consolider efficacement les données provenant des différentes bases de données de la solution Avitax dans le but de fournir à ses clients des informations sur leurs revenus tels que leur chiffre d'affaires, par exploitant, par site, et ce sur des périodes données, ou encore le nombres de chambres non loués ? Pour l'instant ces données sont dispersées, ce qui fais qu'il est impossible pour les clients d'avoir une vue d'ensemble des performances de leurs sites.

Raisons

Il y a plusieurs raisons pour lesquels Avistel veut mettre en œuvre ce projet. Tout d'abord l'amélioration de la gestion financière, en fournissant des données consolidées et simple d'interprétation, les clients pourront prendre des décisions plus réfléchies sur la gestion de leurs finances. Puis ça permettrait une optimisation opérationnelle, fournir une vue d'ensemble du chiffre d'affaires par exploitant et par site permettra au client de pouvoir voir ce qui va, ou ne va pas sur son site, donc également d'optimiser ces opérations commerciales. Enfin proposé gratuitement ce service pourrait permettre à Avistel de renforcer sa force de proposition, et donc de fidéliser ses clients.

Objectifs

Le projet à plusieurs objectifs :

La conception d'un outil de monitoring : Concevoir et développer un outil qui consolide les différentes bases de données de la solution Avitax et les présente de manière claire et concise.

Simplifié l'accès aux données : Offrir aux clients un accès simplifié aux informations sur leur chiffre d'affaires par exploitant, par site et sur des périodes données.

Déployer l'outil via Docker : Assurer que l'outil soit facilement déployable et extensible en utilisant Docker, offrant ainsi une solution pratique pour les clients.

Interface intuitive : Concevoir une interface utilisateur intuitive et esthétique, afin d'améliorer l'expérience utilisateur.

Exigences Fonctionnelles

Visualisation de données : L'outil doit permettre la visualisation de données en fonction de plusieurs facteurs tels que le site, l'exploitant ou encore une période donnée. Il doit également offrir la possibilité de visualiser le nombre de chambres non loués, ainsi que la date de la dernière location.

Fonctionnalité de service : Doit donner la possibilité d'activer ou non une fonctionnalité permettant de faire apparaître le service le moins et le plus rentable.

Connexion sécurisée : L'outil doit disposer d'un système de connexion sécurisée.

Intégration avec Avitax : L'application doit solliciter les bases de données de la solution Avitax afin d'y récupérer les informations nécessaires.

Gestion des rôles : La plateforme doit comporter un système de limitation d'accès en fonction du rôle de l'utilisateur.

Gestion utilisateur : l'application doit permettre à l'utilisateur admin d'ajouter, de suppression ou encore de modifier des comptes client.

Interface Utilisateur : L'interface utilisateur doit être intuitive, facile d'utilisation et adaptable à différents types d'appareils.

Déploiement avec Docker : L'outil doit être déployer sur docker afin de facilité sa maintenance et son évolution.

Processus de réalisation

Documentation :

Avant de commencer le projet, une base de données de la solution avitax (cf Annexe 4) m'a été fournie par Avistel, bien sûr les informations relatives aux clients ont été modifié avant qu'on me transmette celui-ci. Base de données dans laquelle on retrouve des exemples de factures clients, ainsi que des informations sur les types de bouquets et leurs prix, ou encore des informations sur les clients et leurs chambres.

Etude de la Base de données :

Tout d'abord j'ai commencé le projet par une étude approfondie de la base de données. Pour cela, j'ai pris le temps de lire le script plusieurs fois pour comprendre la manière dont les tables ont été conçues, ainsi que les liens entre elles. Cette étape est essentielle avant de commencer à se lancer dans le code, puisque je dois respecter la structure de leur base de données telle qu'elle a été constituée. En comprenant la disposition des tables et les relations entre elles, je pourrais planifier et effectuer le développement de l'application conformément aux exigences.

Etude des besoins utilisateur :

Puis, j'ai identifié les différents besoins qu'un utilisateur pourrait avoir en utilisant un outil tel que celui que je dois concevoir. Pour cela j'ai réalisé un diagramme UML des cas d'utilisations (cf Annexe 5), dans lequel j'ai défini le comportement attendu d'un utilisateur classique, ainsi que celui d'un utilisateur admin. Pour ce qui est de l'utilisateur classique, on s'attend à ce qu'il se connecte/déconnecte et visualise son chiffre d'affaires. Tandis que pour l'utilisateur admin on s'attend plutôt à ce qu'il ajoute, supprime, modifie un utilisateur, ou consulte tout simplement la liste des utilisateurs (une fois connecté).

Réalisation de la maquette :

Tout d'abord pour la réalisation de la maquette, j'ai longuement réfléchi à la manière dont je voulais agencer l'outil. J'ai opté pour un tableau de bord (cf Annexe 6), puisque c'est un format dans lequel sont regrouper différents types d'informations représenté de manière simple et organisée, et ce en temps réel.

Ensuite j'ai conçu la maquette des différentes pages constituant le site web telles que la page de connexion, la page d'accueil, la page de paramètre, ainsi que la page de gestion des utilisateurs, et les versions ou il y a les fenêtres ouvertes. De plus j'ai également réalisé la maquette de la version mobile de l'outil.

Mise en place de l'environnement de développement :

J'ai mis en place mon environnement de développement en installant Visual studio code, qui est l'éditeur de code avec lequel je suis le plus à l'aise, ainsi que WampServer, l'environnement de développement que j'ai appris à utiliser durant ma formation. J'ai également installé Ubuntu Server qui est une variante de l'OS Ubuntu spécialement conçue pour être utilisée sur des serveurs, sur mon poste de travail, ainsi que le logiciel PuTTY, qui est un client SSH me permettant de me connecter au serveur via une connexion de type SSH. Une connexion de type SSH est une méthode qui permet d'établir une communication entre deux ordinateurs de façon sécurisée.

Choix des solutions :

J'ai réalisé un synoptique de la solution, dans lequel je spécifie le comportement de l'application, ainsi que les langages de développement qui seront utilisés pour sa conception. Pour le Front-End, j'ai opté pour HTML, le CSS et le JavaScript, car ce sont les langages avec lesquels je suis le plus à l'aise, et ce sont ceux que j'ai appris lors de ma formation. Pour ce qui est du Back-End, j'ai choisi PHP, qui est l'un des rares langages back end que je maîtrise en plus de Python, et c'est selon moi celui qui convient le mieux à la conception de l'outil de monitoring. En ce qui concerne le Système de gestion de base de données relationnelle j'ai décidé de prendre MySQL, notamment parce que PostgreSQL n'était pas compatible avec la base de données Avitax. Malgré le fait que ce soit le SGBDR que je maîtrise le mieux après celui-ci, la différence de nom de fonction a été un léger problème. De plus j'ai opté pour PhpMyAdmin comme outil de gestion de base de données, pour des raisons de confort.

Mise en place et configuration de l'architecture MVC :

J'ai configuré et mis en place mon architecture MVC (Model-View-Controller), qui est un patron de conception qui consiste à diviser son application en 3 composants :

Model :

Le modèle est la partie qui communique avec la base de données, pour y récupérer des informations et les transmettre au contrôleur, ou les insérer selon sa demande. Pour ce faire il utilise la classe PDO (PHPDataObject) qui permet d'interagir avec différentes bases de données, tels que PostgreSQL, MySQL, SQLite, etc.

View:

La vue correspond tout simplement à ce qui est affiché à l'utilisateur, plus précisément c'est ce avec quoi il interagit.

Controller :

Le contrôleur joue le rôle d'intermédiaire entre la Vue et le Modèle. Selon les actions effectuées par l'utilisateur, il mettra à jour la vue et/ou le modèle.

Pour faire simple, l'utilisateur interagit avec la Vue, déclenchant ainsi une action qui est transmise au contrôleur. Le contrôleur demande ensuite au Modèle de récupérer des informations dans la base de données pour traiter l'action (si nécessaire), puis met à jour la

vue. J'ai opté pour ce patron de conception puisqu'il permet d'organiser le code de manière claire, le rendant ainsi plus facile à lire et à maintenir. De plus, il permet d'éviter la répétition de code et facilite les tests.

Configuration environnement docker :

J'ai installé Docker et créé les différents fichiers de configuration (cf Annexe 7) en fonction des solutions que j'avais choisies. Puis, j'ai débuté par la création du fichier de configuration pour mon conteneur Docker, appelé `docker-back` (cf Annexe 8). Dans ce fichier, j'ai spécifié l'image PHP que j'utilise, ainsi que l'emplacement de mes fichiers PHP et le répertoire dans lequel ils doivent être copiés dans le conteneur. Ensuite, j'ai ajouté les instructions effectuant les dernières mises à jour si cela est nécessaire. Enfin, j'ai configuré l'exposition de mon image sur le port que je souhaitais écouter. Par exemple si j'expose mon image PHP sur le port 8000:80 (ce qui signifie que les requêtes effectuées sur le port 8000 de mon serveur sont redirigées vers le port 80 de mon conteneur Docker), j'aurais accès à mes fichiers PHP en utilisant le lien `Adresse_IP_Server:8000`.

Ensuite, j'ai fait de même pour mon conteneur `Docker-front` (cf Annexe 9), qui contient uniquement l'image Nginx, prenant en charge le HTML, le CSS et le Javascript. De plus, il fonctionne sur un port différent de celui du PHP.

Enfin, j'ai créé le fichier `compose.yaml` (cf Annexe 10), qui est le fichier de configuration de Docker Compose. Ce fichier permet de relier plusieurs conteneurs Docker ensemble. Puis j'ai spécifié la version de Docker Compose que je souhaitais utiliser, ici la version 3. Ensuite, j'ai indiqué les différents services composant le Docker Compose, à savoir le Front-end, Back-end, et PhpMyAdmin. Enfin j'ai paramétré la base de données, le réseau et les volumes du Docker Compose.

Développement sans style :

J'ai fait le choix de commencer à développer les différentes parties de l'outil sans utiliser de style puisque trouver ça beaucoup plus pratique.

Page de connexion :

Tout d'abord j'ai commencé par développer la page de connexion (cf Annexe 11), qui permet à l'utilisateur d'accéder à l'outil de manière sécurisée, ou de modifier son mot de passe. La page est uniquement constituée d'un formulaire classique comprenant deux zones de texte pour le login et le mot de passe, d'un texte menant vers le changement de mot de passe, ainsi qu'un bouton « Se connecter ».

Ensuite, pour faire fonctionner le formulaire il était nécessaire de créer une table "Account" comprenant 6 attributs. Le premier est l'attribut "user_id" qui constitue la clé primaire de la table et est représenté par une suite de chiffres tel que « 127854 », permettant d'identifier les utilisateurs. Puis, nous avons l'attribut "username" qui est unique et équivalent au login, ainsi que l'attribut "pass", permettant à l'utilisateur de s'identifier. Pour des raisons de sécurité, nous n'utilisons jamais de mots de passe uniques. L'utilisation de mots de passe uniques pourrait fournir des indices à un utilisateur malveillant tentant une attaque par Brut Force. L'attaque par Brut Force consiste à tester une multitude de combinaisons de mots de passe jusqu'à trouver le bon. Ensuite l'attribut "adresse" représente l'adresse e-mail de l'utilisateur. L'attribut admin qui ne peut avoir que deux valeurs, 0 ou 1, il a comme valeur par défaut 0, l'attribut sert à définir si l'utilisateur est administrateur ou non.

Enfin, l'attribut "token" de type text représente une chaîne hexadécimale de 64 caractères aléatoires. Ayant rencontré à des problèmes avec les sessions PHP, je les ai remplacées par des tokens API, qui offre une solution plus flexible et sécurisée que l'authentification par session PHP.

L'Authentification s'opère de cette manière (cf Annexe 12) :

Lorsque l'utilisateur s'authentifie, le nom d'utilisateur ainsi que le mot de passe saisi sont stockés dans des variables. Ensuite, le programme vérifie que la combinaison de l'username et du mot de passe, préalablement haché (les mots de passe stockés dans la base de données sont cryptés pour des raisons de sécurité), correspond à l'un des utilisateurs présents dans la base de données. Si la combinaison correspond à une ligne de la table, l'utilisateur se voit accorder un token API, qui est stocké à la fois dans la base de données et dans un Cookie PHP côté client, ayant une durée de validité de 12 heures. Sinon, il est redirigé vers une page d'erreur lui indiquant que ses informations ne correspondent à aucun utilisateur. A noter qu'un Cookie est un petit fichier texte, créé et stocké sur le navigateur. Il contient des informations envoyées par le serveur web.

De plus, une fonction de vérification de la validité du token (cf Annexe 23) est appelée dans chaque page de l'outil. Lorsque que le token est valide (`token.cookie = token.bd`), la fonction crée un "token d'expiration" de même durée que le token en Cookie, s'il n'existe pas déjà, afin de contrôler l'expiration du token. Dans le cas où le token est invalide, l'utilisateur est tout simplement déconnecter.

Page de changement de mot de passe :

Pour ce qui est de la page du changement de mot de passe (cf Annexe 13), il était nécessaire de pouvoir envoyer des mails de manière automatique et personnalisée. J'ai donc opté pour PHPMailer, une bibliothèque PHP qui permet d'envoyer des e-mails via des serveur SMTP. J'ai donc paramétré mon instance PHPmailer avec les différentes informations telles que l'hôte, le protocole de chiffrement, les informations de la boîte mail, le corps du mail, l'entête etc. Dans mon cas, j'ai utilisé le serveur SMTP de Gmail.

La page fonctionne de la manière suivante (cf Annexe 14) :

Elle attend la saisie de l'e-mail lié au compte de l'utilisateur. Une fois récupéré, l'e-mail est entré dans une requête qui a pour but de vérifier si elle correspond à l'un des utilisateurs de la base de données. Si ce n'est pas le cas, l'utilisateur est simplement renvoyé sur un message d'erreur lui indiquant cela. Dans le cas contraire, un mail est envoyé à l'adresse avec code à 6 chiffres (cf Annexe 14.1). Ce code à 6 chiffres est stocké côté serveur pour sa vérification, et son adresse est également conservée pour la suite. Une fois le code saisi, s'il correspond à celui stocké côté serveur, la page redirige vers la page permettant de modifier son mot de passe (cf Annexe 14.2). Sinon, l'utilisateur reçoit simplement un message lui indiquant que le code n'est pas bon. Une fois arrivé sur la page de modification du mot de passe, son formulaire gère toutes les situations qui pourraient se passer, comme ne pas avoir respecté les exigences au niveau du mot de passe, avoir rentré 2 mots de passe différents, voire ne rien rentrer du tout.

Page de sélection de base de données :

Puis j'ai conçu la page de sélection de base de données (cf Annexe 15), qui permet à l'utilisateur de sélectionner une base de données parmi celles auxquelles il a accès. Elle est composée d'un champ de sélection, ainsi que d'un bouton "Continuer". Pour son fonctionnement, j'ai ajouté une table que j'ai nommé "user_bd_access" dans laquelle se trouvent l'attribut "user_id", qui est une clé à la fois primaire et étrangère faisant référence à l'identifiant de l'utilisateur, ainsi que l'attribut "bd_name", qui est une chaîne de caractères, mais également clé primaire faisant référence au nom de la bd.

Pour assurer son fonctionnement, j'ai dû modifier les paramètres de l'instance PDO. Avant, je spécifiais la base de données à laquelle je souhaitais accéder. Désormais, je spécifie directement le nom de la base de données dans mes requêtes via mon paramètre de fonction \$bd, ce qui me permet de changer les informations de l'outil selon la base de données choisie.

Le programme fonctionne de la manière suivante (cf Annexe 16) : lorsque l'utilisateur est connecté et qu'il arrive sur la page, je récupère son identifiant via le cookie "user", qui me permet d'accéder à la liste des noms de base de données de ma table "user_bd_access". Je tiens à préciser que cette table, tout comme la table "account", se trouve dans une base de données à part nommée "User" où sont stockées toutes les données relatives aux utilisateurs. Une fois la liste récupérée, chaque élément est inséré dans l'input de sélection, permettant ainsi à l'utilisateur de sélectionner une base de données, qui sera ensuite stockée dans un Cookie.

J'ai fait le choix de stocker cette information dans un Cookie, tout simplement car la variable d'environnement \$_POST ne conserve les données envoyées par un formulaire jusqu'au prochain chargement de la page. Par exemple, imaginons que j'utilise cette variable et que j'accède une première fois à ma page d'accueil, j'ai bien toutes les données associées à ma base de données puisque \$_POST existe. Cependant, une fois que je quitte la page, la valeur stockée dans \$_POST n'existe plus. Ainsi, lorsque je reviens sur la page, aucune information ne sera affichée.

Page de gestion des utilisateurs :

Ensuite, je suis passé à la page de gestion des utilisateurs (cf Annexe 17), qui permet de consulter la liste des utilisateurs en temps réel, de les modifier, de les supprimer et même d'en ajouter de nouveaux. Cette page est uniquement composée de paragraphes, de boutons et d'un formulaire caché. Durant la conception de la page, j'ai rencontré un problème ; je ne savais pas comment éviter de devoir recharger la page pour visualiser la nouvelle liste après chaque ajout, suppression ou la modification d'un utilisateur, ce qui n'est vraiment pas pratique.

Ne sachant pas comment faire, je suis allé me renseigner sur le web. J'y ai trouvé la classe XML HTTP Request, une classe permettant de faire des requêtes au serveur web depuis JavaScript. Par le biais de cette classe j'ai pu faire en sorte que ma liste se mette à jour à chaque action effectuée, cela en envoyant une requête au serveur web, qui va déclencher ma fonction PHP, qui lui-même va récupérer la liste des users de ma table "account" et la transmettre au serveur pour qu'il puisse répondre à la requête.

C'est la fonction **ActualiserListUsers** (cf Annexe 18) qui se charge de tout ça. Son code fonctionne de la manière suivante :

Elle récupère la liste des utilisateurs présents dans la table "account", puis vérifie si chaque user est déjà présent sur la page. Si c'est le cas, elle met à jour ses informations. Dans le cas contraire, elle ajoute l'utilisateur à la liste, ainsi qu'un bouton portant son identifiant pour le différencier des autres. Ce bouton contient un gestionnaire d'événement qui affiche le formulaire caché, et le préremplit en fonction de l'utilisateur. Pour ce qui est du formulaire, il renvoie un message d'erreur lors de sa validation si aucune information n'a été modifiée. De plus, il est composé de plusieurs champs de texte cachés contenant les informations actuelles de l'utilisateur, afin d'offrir la possibilité de ne changer qu'une seule information.

Tableau de bord :

Pour réaliser le tableau de bord, cela nécessitait de créer des graphiques de tous types. Pour ce faire, j'ai opté pour Chart.js une bibliothèque JavaScript qui permet de générer des graphiques interactifs, notamment des graphiques en barres, en camemberts, ou encore linéaires. J'ai commencé par concevoir des fonctions composées de requêtes SQL (cf Annexe 19), afin de calculer le chiffre d'affaires par site, par exploitant, ou encore le nombre de chambres inactives, etc. Ensuite, j'ai tenté de générer des graphiques avec les données récupérées par le biais de mes fonctions PHP, mais j'ai rencontré un problème qui empêchait les graphiques de se générer. Ce problème était dû au fait que les données récupérées lors de ma requête XMLHttpRequest n'étaient pas au bon format, soit au format objet JSON. J'ai réglé le souci en utilisant la fonction JSON.parse() qui a pour fonction de convertir les chaînes de caractères en objet JSON.

Son programme fonctionne de la façon suivante (cf Annexe 20) :

Une fois arrivé sur la page, une requête XMLHttpRequest est lancée. Elle a pour but d'appeler les fonctions qui calculent les informations nécessaires pour remplir le tableau de bord et ses graphiques. Une fois les informations récupérées, elles sont transmises aux graphiques et aux autres parties de la page, qui peuvent les afficher de manière organisée.

En parallèle, les formulaires (cf Annexe 21) permettant de choisir la période et/ou le département pour lequel on souhaite visualiser son chiffre d'affaires, ainsi que le nombre de factures réglées avec tel ou tel moyen de paiement, sont configurés. Ces formulaires, une fois validés, envoient une requête au serveur contenant les informations saisies, afin de récupérer les données demandées. Une fois récupérées, l'ancien graphique est supprimé pour être remplacé par un nouveau comprenant les nouvelles informations. Ces formulaires sont composés d'input de type Date, Week, Month, et Year. Certains de ces inputs ne renvoyaient pas l'information voulue. Par exemple, pour les graphiques, il était nécessaire

d'avoir tous les chiffres d'affaires de chaque semaine dans le mois. Ne trouvant aucun input pour le faire, j'ai dû utiliser l'input mois permettant de récupérer le mois de l'année saisie, que j'ai couplé à une fonction "découperMois" (cf Annexe 22) qui prend un mois dans l'année, et qui le découpe en semaines en indiquant la date de début et de fin de chaque semaine du mois. J'ai fait cela pour plusieurs d'entre eux.

Compétences mobilisées

Dans le cadre du projet, j'ai mobilisé plusieurs compétences acquies durant ma formation :

Compétence 1 : Réaliser un développement d'application

J'ai débuté ce projet en étudiant de manière détaillée la base de données Avitax, afin de comprendre sa structure et de planifier le développement de l'outil conformément aux exigences. Ce qui m'a permis de concevoir une maquette de l'outil en accord avec les besoins des utilisateurs, et de mettre en place un environnement de développement adapté.

Compétence 2 : Optimiser des applications informatiques

Durant le développement, j'ai constamment cherché à optimiser les performances de l'outil, en faisant des requêtes SQL ne récupérant seulement des données utiles, et en réduisant le nombre de requêtes. J'ai également fait en sorte de toujours regrouper les requêtes XMLHttpRequest qui pouvaient l'être. Aussi, j'ai usé des bonnes pratiques de développement en réalisant des tests assurant le bon fonctionnement du code.

Compétence 3 : Administrer des systèmes informatiques communicants

J'ai mis cette compétence en application lors de la configuration du serveur Ubuntu et de l'environnement Docker.

Compétence 4 : Gérer des données de l'information

J'ai appliqué cette compétence depuis l'étude de la base de données Avitax jusqu'au développement des pages d'authentification sécurisée et de gestion des utilisateurs. De plus, j'ai dû utiliser une méthode de cryptage afin de protéger les données sensibles, tout en les manipulant les données efficacement.

Compétence 5 : Conduire un projet

Dans ce projet, j'ai occupé le rôle de chef de projet, ce qui m'a amené à planifier les différentes étapes mener le projet à bien. J'ai également dû gérer les problèmes rencontrés et analyser l'environnement du projet.

Gestion du temps

Durant ce projet, j'estime avoir mal géré mon temps. Il faut savoir que les horaires de l'entreprise sont de 9h à 13h et de 14h à 17h30, avec une pause de 20 minutes le matin et l'après-midi. J'aurais pu faire le choix d'importer du travail chez moi, mais je ne l'ai pas fait car je pensais avoir plus de temps que nécessaire. Malheureusement, j'ai perdu énormément de temps lors de la configuration de l'environnement Docker.

Conclusion

Globalement, je suis plutôt satisfait de ma première expérience professionnelle en entreprise. J'ai acquis de nouvelles compétences en travaillant sur un environnement de développement différent, principalement en utilisant un serveur Ubuntu distant via Visual Studio Code en SSH. De plus, J'ai découvert l'utilité que pouvait avoir Docker pour le déploiement d'applications et appris à configurer un environnement Docker. D'autre part, cette expérience m'a également permis d'approfondir mes connaissances dans plusieurs langages de programmation vu durant mes années de formation, me permettant ainsi de rafraîchir ma mémoire par rapport à Javascript, et combler certaines lacunes que je pouvais avoir en PHP, notamment au niveau de ma compréhension de l'architecture MVC, ainsi que ma compréhension des requêtes SQL. Par ailleurs, je pense également avoir amélioré ma capacité à résoudre les problèmes grâce aux difficultés rencontrés pendant ce projet.

Cependant, cette expérience m'a aussi permis de prendre conscience de mes faiblesses. Selon moi, je suis loin du niveau nécessaire pour être intégré dans un projet professionnel. Mais au moins, cela m'a fait prendre conscience de l'importance des projets personnels dans ce milieu. Malheureusement, ce n'est pas avec le peu de projets que l'on fait dans ma formation que je vais m'améliorer. Je pense que je dois voir cela plus comme un supplément qu'autre chose. En conséquence, je suis déterminé à consacrer plus de temps à des projets personnels et à continuer d'apprendre de manière autonome pour combler mes lacunes et atteindre un bon niveau.

Le projet étant loin d'être terminé, il ne risque pas d'être utilisé par l'entreprise, mais je prévois le finir de mon côté, afin d'enrichir d'avantage mon portfolio.

Actuellement, je réfléchis sérieusement à la possibilité de faire de l'alternance l'année prochaine. Ce qui est sûr, c'est que ce n'est pas le genre d'entreprise dans laquelle j'ai envie de faire de l'alternance. En raison de sa taille, l'entreprise n'a pas spécialement le temps de m'encadrer comme je souhaiterais l'être.

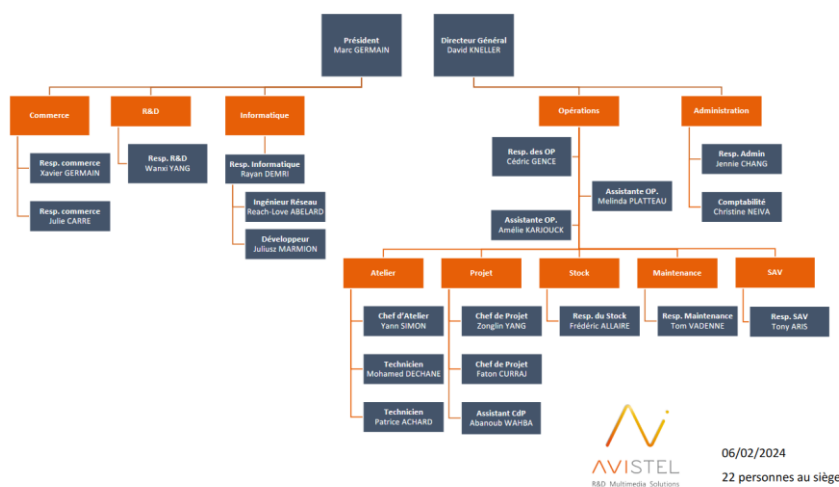
Au moment présent, je ne peux pas dire que cette expérience m’ait aidé à y voir plus claire au niveau de mon projet professionnel, puisque je n’ai pas été réellement impliqué dans un projet ayant un quelconque intérêt pour l’entreprise. Ce qui est tout à fait normal en raison de mon manque de compétence.

En conclusion, cette première expérience en entreprise a été enrichissante autant techniquement que professionnellement. Elle m’a permis d’acquérir de nouvelles compétences, de consolider mes connaissances en programmation et d’améliorer ma capacité à résoudre des problèmes. De plus, cela m’a également permis de prendre conscience de mes faiblesses et de l’importance que peuvent avoir les projets personnels. Je suis reconnaissant envers l’équipe d’Avistel pour cette opportunité, qui me servira tout au long de mon parcours professionnels.

Table des illustrations

I Annexes

Annexe 1 - Organigramme Avistel



Annexe 2 - Espace de travail



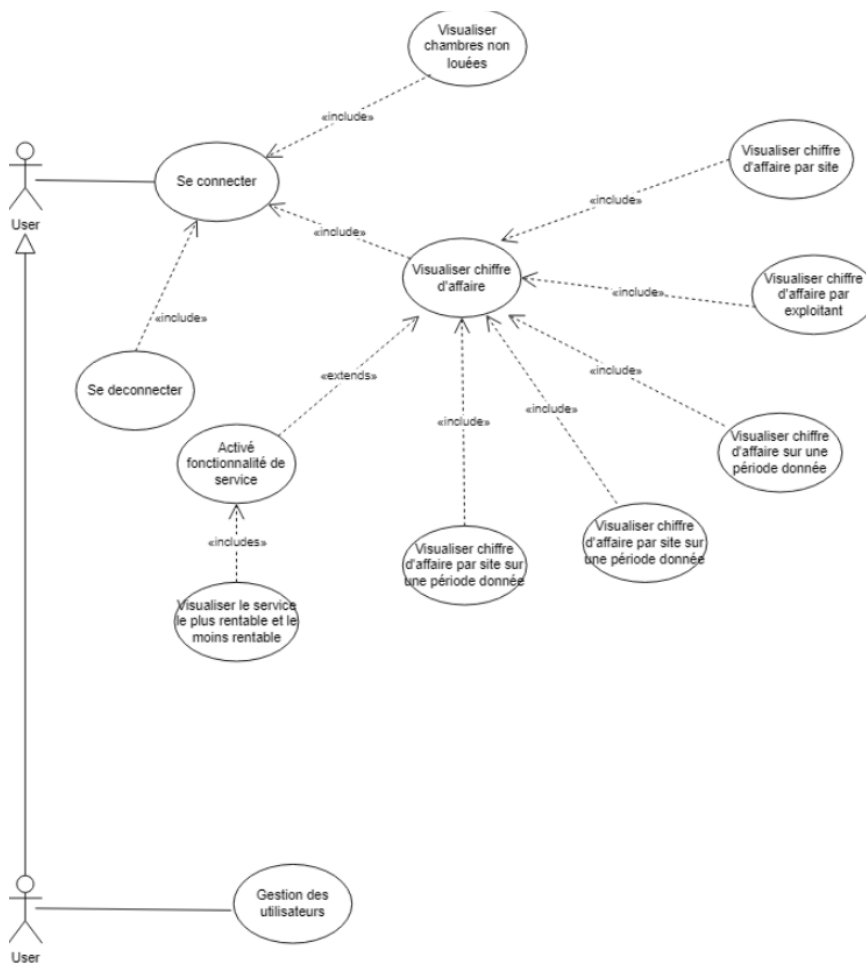
Annexe 3 - Atelier



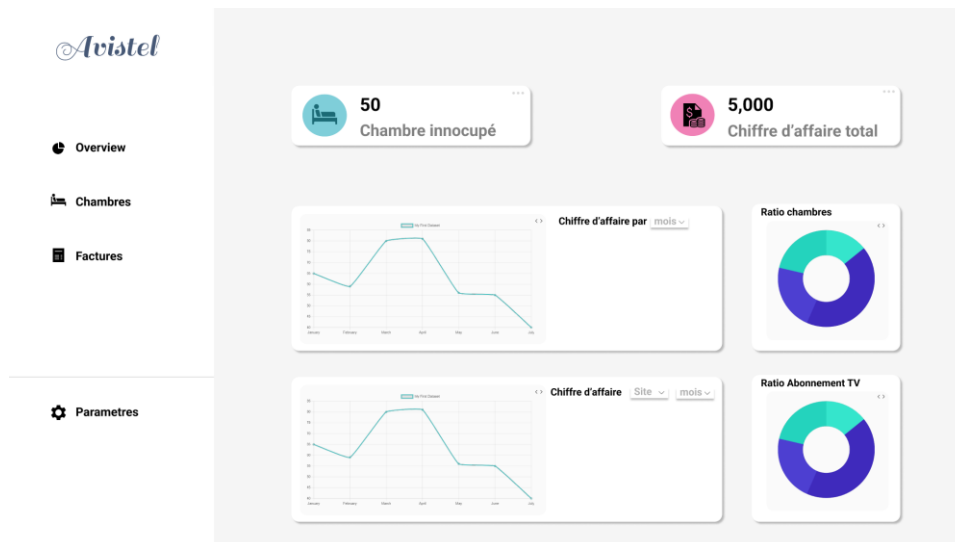
Annexe 4 – BD Solution Avitax

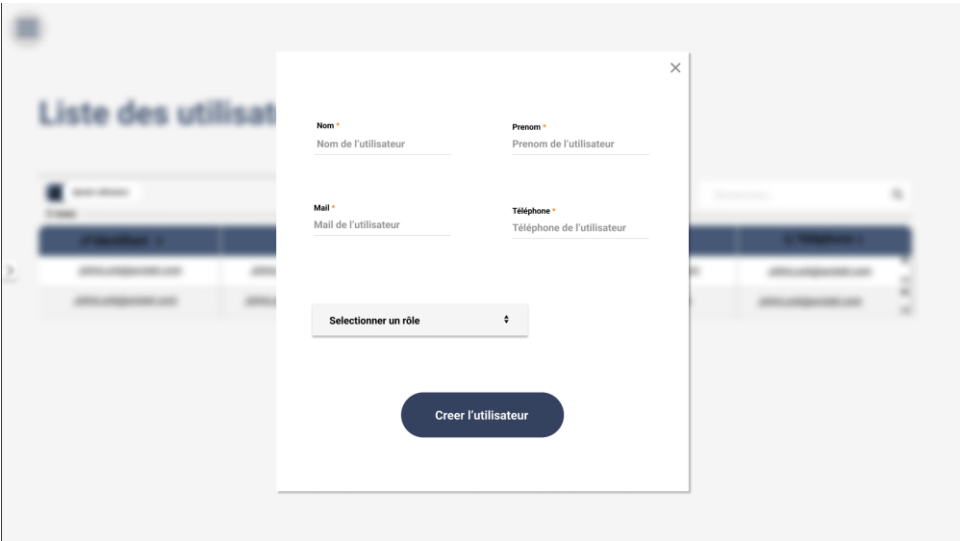
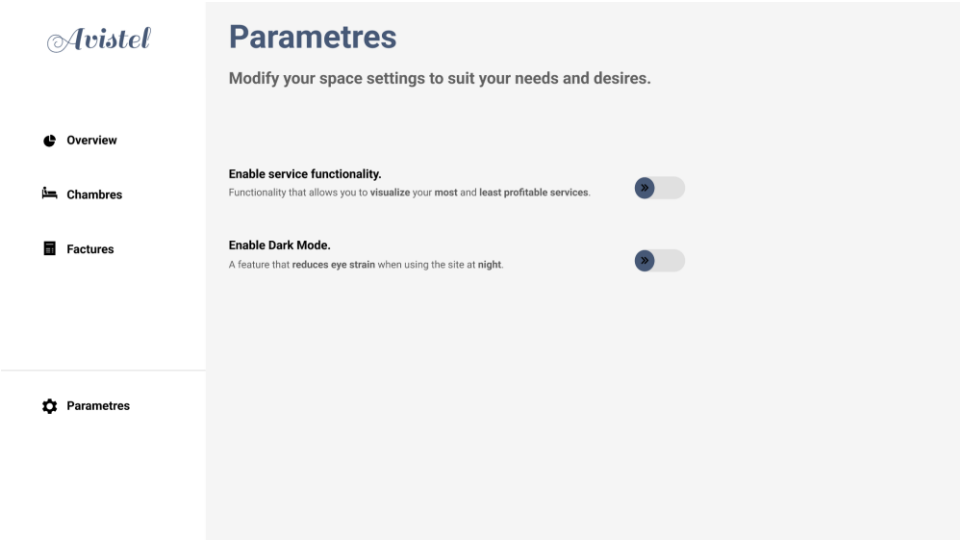
<input type="checkbox"/> bouquets	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	16	InnoDB	latin1_swedish_ci	32,0 kio	-
<input type="checkbox"/> chambres	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	812	InnoDB	latin1_swedish_ci	160,0 kio	-
<input type="checkbox"/> clients	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	932	InnoDB	latin1_swedish_ci	224,0 kio	-
<input type="checkbox"/> configuration	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	1	InnoDB	latin1_swedish_ci	16,0 kio	-
<input type="checkbox"/> factures_tv	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	1 285	InnoDB	latin1_swedish_ci	320,0 kio	-
<input type="checkbox"/> operations	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	16 333	InnoDB	latin1_swedish_ci	2,1 Mio	-
<input type="checkbox"/> ordres_tv	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	0	InnoDB	latin1_swedish_ci	16,0 kio	-
<input type="checkbox"/> ordre_wifi	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	0	InnoDB	latin1_swedish_ci	16,0 kio	-
<input type="checkbox"/> passlogs	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	2	MyISAM	utf8mb4_0900_ai_ci	4,1 kio	-
<input type="checkbox"/> systemes	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	1	InnoDB	latin1_swedish_ci	16,0 kio	-
<input type="checkbox"/> systemes_wifi	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	1	InnoDB	latin1_swedish_ci	16,0 kio	-
<input type="checkbox"/> tarifs	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	730	InnoDB	latin1_swedish_ci	80,0 kio	-

Annexe 5 - Diagramme UML



Annexe 6 – Maquette Figma





Liste des utilisateurs

Ajouter utilisateur

Rechercher...

2 rows

Identifiant	Nom	Prenom	Adresse	Téléphone	
1	JohnLock@avistel.com	JohnLock@avistel.com	JohnLock@avistel.com	JohnLock@avistel.com	...
2	JohnLock@avistel.com	JohnLock@avistel.com	JohnLock@avistel.com	JohnLock@avistel.com	...

Annexe 7 - Fichier de configuration Docker

```
GNU nano 6.2 nginx.conf
server {
    listen 80;
    server_name 192.168.10.51;

    root /var/www/html;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

[Read 18 lines]

Help Write Out Where Is Cut Execute Location M-U Undo
Exit Read File Replace Paste Justify Go To Line M-E Redo

```

RUN docker-php-ext-install mysqli pdo pdo_mysql
RUN apt-get update && apt-get install -y nginx
COPY ./ /var/www/html
COPY nginx.conf /etc/nginx/sites-available/default
EXPOSE 80
CMD service php7.4-fpm start && nginx -g "daemon off;"

avistel@lxstage:~/app$ ls
back          compose.yaml.old  docker-front  html          requirements.txt
compose.yaml  docker-back       front         nginx.conf    test.php
avistel@lxstage:~/app$ cd back
avistel@lxstage:~/app/back$ ls
config.php  index.php  Model.php
avistel@lxstage:~/app/back$

```

Annexe 8 - Docker-back

```

GNU nano 6.2                                docker-back
FROM php:fpm

COPY ./back/* /var/www/html

RUN apt-get update && apt-get install -y php-fpm
RUN docker-php-ext-install mysqli pdo pdo_mysql
RUN apt-get update && apt-get install -y nginx
COPY ./ /var/www/html
COPY nginx.conf /etc/nginx/sites-available/default
EXPOSE 80
CMD service php7.4-fpm start && nginx -g "daemon off;"

[ Read 18 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^_ Go To Line ^B Redo

```

Annexe 9 - Docker-front

```
GNU nano 6.2 docker-front
FROM nginx:alpine

EXPOSE 80
```

[Read 3 lines]

^G Help ^O Write Out ^W Where Is [Read 3 lines] ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^_ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

Annexe 10 - Compose.yaml

```
GNU nano 6.2 compose.yaml
version: "3"

services:
  frontend:
    build:
      context: ./
      dockerfile: docker-front
    ports:
      - "8000:80"
    volumes:
      - ./front/src:/usr/share/nginx/html/
    networks:
      - app-network

  php:
    build:
      context: ./
      dockerfile: docker-back
    ports:
      - "8001:80"
    volumes:
      - ./back/:/var/www/html/
    networks:
      - app-network

  web:
    image: nginx:latest
    ports:
      - "8003:80"

  phpmyadmin:
    container_name: phpmyadmin
    image: phpmyadmin/phpmyadmin:latest
    environment:
      PMA_HOST: database
```

[Read 64 lines]

^G Help ^O Write Out ^W Where Is [Read 64 lines] ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^_ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

```

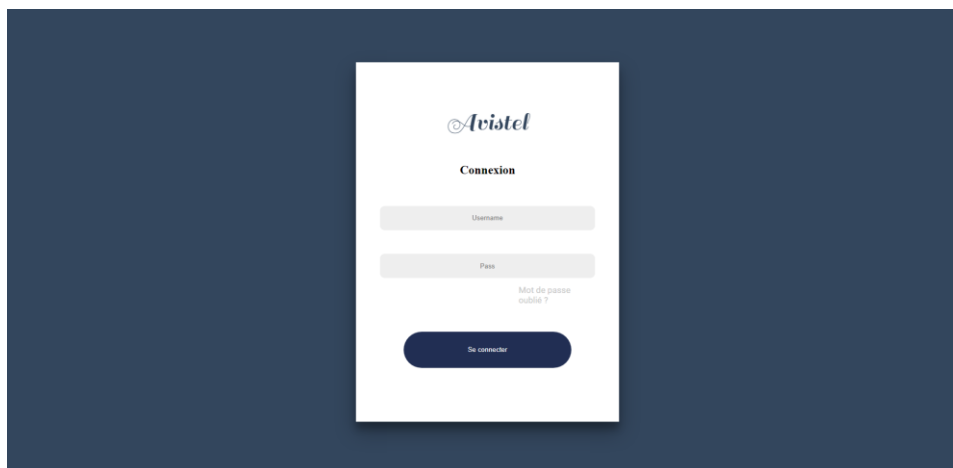
GNU nano 6.2                                compose.yaml *
PMA_HOST: database
UPLOAD_LIMIT: 64M
ports:
  - "8002:80"
depends_on:
  - database
networks:
  - app-network
volumes:
  - /etc/localtime:/etc/localtime:ro

database:
  container_name: database
  image: mysql:latest
  environment:
    MYSQL_ROOT_PASSWORD: avistel
    MYSQL_DATABASE: avistel
    MYSQL_USER: avistel
    MYSQL_PASSWORD: avistel
    MYSQL_CHARSET: utf8
    MYSQL_COLLATION: utf8_general_ci
  volumes:
    - db-data:/var/lib/mysql
    - /etc/localtime:/etc/localtime:ro
  networks:
    - app-network
networks:
  app-network:
    external: true

volumes:
  db-data:

```

Annexe 11 - Page login



Annexe 12- Code Page Login

```

public function action_login()
{
    if ((isset($_POST["submit"]) && isset($_POST["username"]) && isset($_POST["pass"])) && ($_POST["username"] != "" && $_POST["pass"] != "")) {
        if($_POST["username"]){
            $m = Model::getModel();
            $passhash = hash("sha256", $_POST["pass"]);
            $user = $m->getUser($_POST["username"], $passhash);

            if ($user != false) { // Si l'utilisateur est trouvé, alors on continue.
                $token = bin2hex(random_bytes(32));

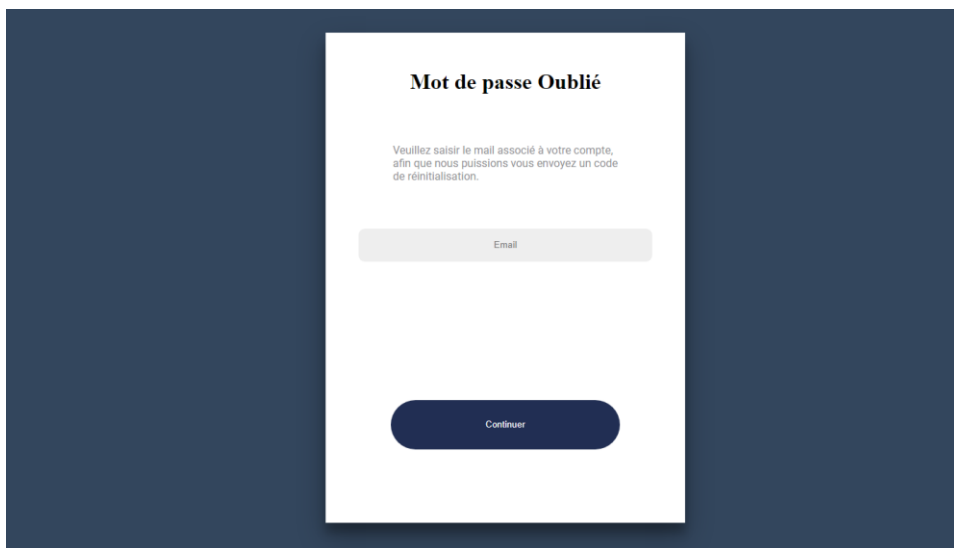
                // Mise à jour du token utilisateur dans la base de données
                $m->updateUserToken($token, $user['user_id']);

                // Création des cookies pour l'utilisateur et le token
                setcookie("user", $user['user_id'], time() + 1800, "/");
                setcookie("token", $token, time() + 1800, "/");

                // Redirection vers la page d'accueil avec un paramètre unique pour éviter le cache
                header("Location:index.php?controller=select_bd&no-cache=" . uniqid());
                exit;
            } else {
                echo "Mot de passe ou nom d'utilisateur incorrect.";
            }
        } else {
            echo "Veuillez remplir tout les champs.";
        }
    }
}

```

Annexe 13 - Page mot de passe oublié



Annexe 14 - Code Page Mot De Passe Oublié

Annexe 14.1 - Code d'envoi de mail

```
public function action_passforgot(){
    if(isset($_POST["submit"])){
        echo "submit";
        $m=Model::getModel();
        $user=$m->getUserByAddress($_POST["address"]);
        if($user == true){

            $mail = new PHPMailer(true);

            try {
                $code=random_int(100000,999999);
                if(session_status() == PHP_SESSION_NONE){
                    session_start();
                }

                $_SESSION["code"]=$code;
                $_SESSION["address"]=$_POST["address"];
                // Paramètres du serveur SMTP (Gmail)
                $mail->isSMTP();
                $mail->Host = 'smtp.gmail.com';
                $mail->SMTPAuth = true;
                $mail->Username = 'rycou123@gmail.com';
                $mail->Password = 'rycou123@gmail.com';
                $mail->SMTPSecure = 'tls'; // ssl ou tls
                $mail->Port = 587;

                // Paramètres du message
                $mail->setFrom("rycou123@gmail.com", "AVI...");
                $mail->addAddress($_POST["address"], $user["username"]);
                $mail->Subject = "AVI... Changement De Mot De Passe";
                $mail->Body = "Bonjour, voici votre code:".$code;

                // Ajoute l'en-tête Return-Path
                $mail->addCustomHeader('Return-Path: rycou123@gmail.com');

                // Envoi de l'e-mail
                $mail->send();
                echo "e-mail a été envoyé avec succès.";
                extract([$code]);
                header('Location:index.php?controller=verif_code&action=default');
                exit;

            } catch (Exception $e) {
                echo 'Erreur lors de l\'envoi de l\'e-mail : ', $mail->ErrorInfo;
            }
        }
    }
}
```

Annexe 14.1 - Code changement mdp

```

public function action_resetpass(){
    if(isset($_SESSION['address'])) && (isset($_POST['newpass'])) && (isset($_POST['confirm_pass'])) && $_POST['newpass'] != "" && $_POST['confirm_pass'] != "" ){ // Si les saisies ne sont pas vides on continue.

        if($_POST['confirm_pass'] == $_POST['newpass']){ // Si la saisie est la même on continue.

            if(preg_match("/^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#%^&*~`-+=~(){}]).{8,}$/", $_POST['confirm_pass'])){ // Si le mdp contient au moins 1 majuscule, 1 chiffre et 1 caractère spécial on continue.

                if($_POST['newpass'] == $_POST['confirm_pass']){ // Si les 2 saisies sont les mêmes on continue.

                    $m = Model::getModel();
                    $user = $m->getUserByAddress($_SESSION['address']); // Récupère les infos de l'utilisateur.
                    $userid = $user->id; // Récupère seulement son id.

                    $verif = $m->checkIfPassAlreadyUsed($userid, $_POST['newpass']); // Recherche le mot de passe dans l'historique des mdp de l'utilisateur.

                    if($verif == null){ // Si le mot de passe n'a jamais été utilisé, on autorise le changement.
                        $passhash = sha256($_POST['newpass']);
                        $m->updatePassByAddress($_SESSION['address'], $passhash);
                        $m->addInfoPassing($userid, $passhash);
                        unset($_SESSION['address']);
                        $this->render("login", []);
                    } else{echo "Merci de saisir un mot de passe que vous n'avez jamais utilisé";}

                } else{echo "Les mots de passe diffèrent";}

            } else{echo "Le mot de passe doit contenir au minimum 1 majuscule, 1 chiffre et 1 caractère spécial";}

        } else{echo "Le mot de passe doit faire au minimum 8 caractères";}

    }
}

```

Annexe 15 - Page selection bd

gtvx

Annexe 16 - Code page selection bd

```

<?php
require_once 'Controller.php';
require_once __DIR__ . '/../Models/Model.php';
require_once 'verification.php';

class Controller_select_bd extends Controller{

    public function action_select_bd(){
        if(isset($_COOKIE['token']) && isset($_COOKIE['user'])){ // Si client connecter on continue
            $expiration_cookie = $_COOKIE['user'];
            setcookie('bd_select', $_POST['bd'], $expiration_cookie - time(), "/");
            $this->render('home');
        }

    }

    public function action_default(){
        verification();
        $this->render('select_bd');
    }

}

if (($_SERVER['REQUEST_METHOD'] == "POST" && isset($_POST['getBd'])) ) { // Si requête post ayant comme clé getBd, alors on renvoie au serveur l'id du bd sélectionné
    $m = Model::getModel();
    $tab = $m->getAllUserBd($_COOKIE['user']);

    echo json_encode($tab);
}

?>

```

```

// Page Select

// Remplir le champ select de la liste des bd de l'utilisateur
var xhr = new XMLHttpRequest();
xhr.open('POST', './Controllers/Controller_select_bd.php', true);
xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {
        var response = JSON.parse(this.responseText);
        var input_select_bd = document.getElementById("bd_select"); // Récupère l'input select

        response.forEach((bd) => { // Insère chaque bd lié à l'utilisateur dans l'input
            console.log(bd);
            var option = document.createElement("option");
            option.value = bd.bd_name;
            option.textContent = bd.bd_name;
            input_select_bd.appendChild(option);
        });
    }
};

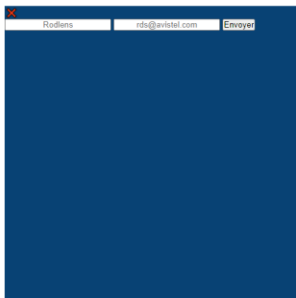
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhr.send('getBd');

```

Annexe 17 - Page gestion users



username= Aiq address= rds@avistel.com
(modifier/supprimer)
username= Avistel address= avistel@gmail.com
(modifier/supprimer)
username= Arcanus address= keishura12@gmail.com
(modifier/supprimer)
username= Doc address= doc@doc.com
(modifier/supprimer)
username= duchojacq address= duchosal.jacqueline@avistel.com
(modifier/supprimer)



username= Rodlens address= rds@avistel.com
(modifier/supprimer)
username= Avistel address= avistel@gmail.com
(modifier/supprimer)
username= Arcanus address= keishura12@gmail.com
(modifier/supprimer)
username= Doc address= doc@doc.com
(modifier/supprimer)
username= duchojacq address= duchosal.jacqueline@avistel.com
(modifier/supprimer)

Annexe 18 - Code page gestion utilisateurs

```
class Controller_userlist extends Controller{

    public function action_userlist(){

    }

    public function action_modifyuser(){

        if (isset($_POST["oldusername"]) && isset($_POST["newusername"])){
            var_dump($_POST["oldusername"]);
            var_dump($_POST["newusername"]);
            $m=Model::getModel();
            $m->updateUserUsername($_POST["oldusername"], $_POST["newusername"]);
            $this->render("userlist",[]);
        } else{ $this->render("error404", []);}

    }

    public function action_default(){

        $this->render("userlist",[]);

    }

}
```

```
if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["oldaddress"]) && isset($_POST["oldusername"]) && isset($_POST["newusername"]) && $_POST["newusername"] != "" ) {
    // Récupérez les données du formulaire
    $m=Model::getModel();
    $newusername = $_POST["newusername"];
    $oldusername = $m->getUserByAddress($_POST["oldaddress"]);
    $oldusername=$oldusername['username'];
    // Appel la fonction permettant de modifier l'username de l'utilisateur

    $m->updateUserUsername($oldusername, $newusername);

    echo "Username change";

}

if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["oldaddress"]) && isset($_POST["newaddress"]) && $_POST["newaddress"] != "" ) {
    // Récupérez les données du formulaire
    $newaddress = $_POST["newaddress"];

    // Appel la fonction permettant de modifier l'adresse de l'utilisateur
    $m=Model::getModel();
    $user= $m->getUserByAddress($_POST["oldaddress"]);
    $oldaddress=$user['adresse'];

    $m->updateUserAddress($oldaddress, $newaddress);
    // Vous pouvez envoyer une réponse JSON au client si nécessaire

    echo "Adresse: $oldaddress ->modifié par ". $newaddress;

}

if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["oldaddress"]) && isset($_POST["newaddress"]) && isset($_POST["oldusername"]) && isset($_POST["newusername"]) && $_POST["newaddress"] != "" && $_POST["newusername"] != "" ) {
    // Récupérez les données du formulaire
    $newaddress = $_POST["newaddress"];
    $newusername=$_POST["newusername"];

    // Appel la fonction permettant de modifier l'adresse de l'utilisateur
    $m=Model::getModel();
    $user= $m->getUserByAddress($_POST["oldaddress"]);
    $oldaddress=$user['adresse'];
    $oldusername=$user['username'];
    $m->updateUserAddress($oldaddress, $newaddress);
    $m->updateUserUsername($oldusername, $_POST["newusername"]);
    // Vous pouvez envoyer une réponse JSON au client si nécessaire

    echo "Adresse: $oldaddress ->modifié par ". $newaddress. " et ". $oldusername. " en ". $newusername;

}

if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["asktab"]) ) {
    $m=Model::getModel();
```

The Django Service could not be started because the specified port or ports (3000) are already in use.
Source: IBM Cloud for xCloud Developer Extension

```

if (($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["asktab"]) ) {

    $m=Model::getModel();
    $users=$m->getAllUser();

    echo json_encode($users);

}

if (($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["updateUser"])) {

    $m=Model::getModel();
    $userlist=$m->getAllUser();

    echo json_encode($userlist);

}

if (($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["getUsers"])){
    $m=Model::getModel();
    $userlist=$m->getAllUser();

    echo json_encode($userlist);
}

if (($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST['userid'])) {

    $m=Model::getModel();
    $m->delUser($_POST['userid']);
    echo "Utilisateur ".$_POST['userid']." Supprimé";

}

```

Code ActualiserUsers

```

function ActualiserListUsers() { // Actualise la liste des utilisateur
var xhr = new XMLHttpRequest();
xhr.open("POST", "../controllers/Controller_userlist.php", true);

xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {
        var users = JSON.parse(this.response);
        var userRows = document.querySelectorAll('.user_row');

        users.forEach(function(user) {
            var existingRow = Array.from(userRows).find(function(row) { // Recherche le bouton relié à l'utilisateur courant
                return row.querySelector('.voirplus').name === user.user_id;
            });

            if (!existingRow) { // Si l'utilisateur est présent dans la bd, mais pas dans la liste alors il l'ajoute à celui-ci.
                var div = document.createElement("div");
                div.classList.add('user_row');

                var para = document.createElement("p");
                para.textContent = "username= " + user.username + " address= " + user.adresse;
                div.appendChild(para);

                var input = document.createElement("input");
                input.type = "submit";
                input.classList.add('voirplus');
                input.name = user.user_id;
                input.value = "modifier";
                div.appendChild(input);

                var input_del = document.createElement("input");
                input_del.type = "submit";
                input_del.classList.add('del');
                input_del.value = "supprimer";
                input_del.name = user.user_id;
                div.appendChild(input_del);

                document.body.appendChild(div);

                // Ajoute un gestionnaire d'événements pour ce nouveau bouton voirplus
                input.addEventListener("click", function() {
                    var userinfos = document.getElementById('userinfos');
                    let newusername_userlist = document.getElementById('newusername-userlist');
                    let newadress_userlist = document.getElementById('newadress-userlist');

                    newusername_userlist.value="";
                    newadress_userlist.value="";

                    newusername_userlist.placeholder = user.username;
                    newadress_userlist.placeholder = user.adresse;
                });
            }
        });
    }
};

```

```

        var oldusername = document.getElementById('oldusername-userlist');
        oldusername.value = user.username;

        var oldaddress = document.getElementById('oldaddress-userlist');
        oldaddress.value = user.adresse;

        userinfos.style.visibility = 'visible';
    });

    // Ajoute un gestionnaire d'événements pour button supprimer
    inputs=Array.from(document.querySelectorAll(".del"));

    inputs.forEach((val, key) =>{
        inputs[key].addEventListener("click", function(){

            xhr= new XMLHttpRequest();
            xhr.open('POST', './Controllers/Controller_userlist.php', true);
            xhr.onreadystatechange=function(){
                if(xhr.readyState ==4 && xhr.status==200){
                    inputs[key].parentNode.remove();
                }
            }

            let userid=this.name;

            xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

            xhr.send("userid="+userid);

        });
    });

    // Si il existe il met seulement à jour les valeurs username et adresse.
    var userPara = existingRow.querySelector("p");
    userPara.textContent = "username= " + user.username + " adresse= " + user.adresse;
});

userRows.forEach(function(row) { // Verifie que chaque lignes de la liste correspondent à un utilisateur, sinon il supprime celui-ci.
    var userid = row.querySelector(".voirplus").name;
    var existingUser = users.find(function(user) {
        return user.user_id == userid;
    });
});

```

```

//
userRows.forEach(function(row) { // Verifie que chaque lignes de la liste correspondent à un utilisateur, sinon il supprime celui-ci.
    var userid = row.querySelector(".voirplus").name;
    var existingUser = users.find(function(user) {
        return user.user_id == userid;
    });
});

if (!existingUser) {
    row.remove();
}
});

xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
xhr.send("getUsers=2");

```

Annexe 19 - Requetes SQL

```

public function getAllChiffreAffaire()
{
    $query = $this->bd->prepare("SELECT SUM(montant_total) AS Chiffred'affaire,id_client FROM factures_tv WHERE type <> 'remboursement' GROUP BY id_client");
    $query->execute();

    $Allchiffred'affaire = $query->fetchAll(PDO::FETCH_ASSOC);
    return $Allchiffred'affaire;
}

public function getAllDep($bd)
{
    $query = $this->bd->prepare("SELECT DISTINCT departement_name FROM ".$bd.">.factures_tv WHERE departement_name <> ''");
    $query->execute();
    $res=$query->fetchAll(PDO::FETCH_ASSOC);
    return $res;
}

public function getPaymentMethodStatByDepAndYear($dep,$year,$bd)
{
    $query = $this->bd->prepare("SELECT COUNT(reglement) AS nb_facture, reglement, departement_name FROM ".$bd.">.factures_tv WHERE departement_name=:dep AND YEAR(date_saisie)=:year GROUP BY reglement;");
    $query->bindValue("dep", $dep);
    $query->bindValue("year", $year);
    $query->execute();

    $PaymentMethodStat = $query->fetchAll(PDO::FETCH_ASSOC);
    return $PaymentMethodStat;
}

public function getAllPaymentMethodStatByYear($year,$bd)
{
    $query = $this->bd->prepare("SELECT COUNT(reglement) AS nb_facture, reglement, departement_name FROM ".$bd.">.factures_tv WHERE YEAR(date_saisie)=:year GROUP BY reglement;");
    $query->bindValue("year", $year);
    $query->execute();

    $PaymentMethodStat = $query->fetchAll(PDO::FETCH_ASSOC);
    return $PaymentMethodStat;
}

public function getPaymentMethodStatByDepAndDate($dep,$date,$bd)
{
    $query = $this->bd->prepare("SELECT COUNT(reglement) AS nb_facture, reglement, departement_name FROM ".$bd.">.factures_tv WHERE departement_name=:dep AND date_saisie=Y:date GROUP BY reglement;");
    $query->bindValue("dep", $dep);
}

```

Annexe 20 - Programme Tableau de bord

```
if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["Dep"]) && isset($_POST["Date"])) {
    $m = Model::getModel();
    $tab=[];
    $i =0;
    $a=$i;
    while($i<12){
        if($i < 10){
            $a='0'.strval($i);
            $tab[$i]=$m->getTurnoverByDepByMonthAndYear($_POST["Dep"], $_POST["Date"], $a);
        } else{
            $tab[$i]=$m->getTurnoverByDepByMonthAndYear($_POST["Dep"], $_POST["Date"], $i);
        }
        $i+=1;
    }

    // Convertir le tableau en JSON et l'afficher
    echo json_encode($tab);
}

if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["Dep"]) && isset($_POST["Year"]) && isset($_POST["Month"])) {
    $m = Model::getModel();

    $semaines=decouperMois($_POST["Year"], $_POST["Month"]);
    $i=0;
    $tab=[];
    while($i<count($semaines)){
        $tab[$i]=$m->getTurnoverByWeek($_POST["Dep"], $_POST["Year"], $_POST["Month"],$semaines[$i][0],$semaines[$i][1]);
        $i++;
    }

    // Convertir le tableau en JSON et l'afficher
    echo json_encode(array($tab,$semaines));
}

if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["Dep"]) && isset($_POST["WeekDate"])) {
    $m = Model::getModel();

    $date=explode("/",$_POST["WeekDate"]);
    $jours=weekToDay(intval($date[0]),intval($date[1]), intval($date[2]));
    $i=0;
    $tab=[];
    while($i<count($jours)){
        $tab[$i]=$m->getTurnoverByDepByDate($_POST["Dep"], $jours[$i]);
        $i++;
    }

    // Convertir le tableau en JSON et l'afficher
    echo json_encode(array($tab,$jours));
}
```

```
if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["getDeps"])) {
    $m = Model::getModel();
    $tab_sites = $m->getALLDep('gtvx');

    // Convertir le tableau en JSON et l'afficher
    echo json_encode($tab_sites);
}

if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["getChartsData"])) {
    $m = Model::getModel();

    $AllDepCAForCurYear=[];
    $dep=$m->getALLDep('gtvx');
    foreach($dep as $value ){
        $AllDepCAForCurYear[]=$m->getTurnoverByDepByYear($value['department_name'],2023,$_COOKIE["bd_select"]);
    }

    $AllDepPmStat=[];
    foreach($dep as $value){
        $AllDepPmStat[]=$m->getPaymentMethodStatByDepAndYear($value['department_name'],date("Y"),$_COOKIE["bd_select"]);
    }

    $NotRentRooms = $m->countNotRentRooms();
    $RentRooms = $m->countRentRooms();

    $NRR_ratio=($NotRentRooms['nb_not_rent_rooms']/ $RentRooms['nb_rent_rooms']) * 100;
    $RR_ratio= 100-$NRR_ratio;
    $NRR_ratio=round($NRR_ratio);
    $RR_ratio= round($RR_ratio);

    $Sub_ratio=$m->countSubscription($_COOKIE["bd_select"]);
    $CA_Site=$m->getSiteTurnover($_COOKIE["bd_select"]);
    $Site_MP_Stat=$m->countPaymentMethod($_COOKIE["bd_select"]);

    // Convertir le tableau en JSON et l'afficher
    echo json_encode(array([$NRR_ratio,$RR_ratio],$Sub_ratio,$AllDepCAForCurYear, $AllDepPmStat,$CA_Site,$Site_MP_Stat));
}

if (($SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["mpDep"]) && isset($_POST["mpDate"])) {
    $m = Model::getModel();

    if($_POST["mpDep"]=="tout"){
        $PM_date=$m->getALLPaymentMethodStatByYear($_POST["mpDate"],$_COOKIE["bd_select"]);
    } else{
```

```

if (($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["mpDep"]) && isset($_POST["mpDate"])) {
    $m = Model::getModel();

    if($_POST["mpDep"]=="Tout"){
        $PM_date=$m->getAllPaymentMethodStatByYear($_POST["mpDate"],$_COOKIE["bd_select"]);
    } else{
        $PM_date=$m->getPaymentMethodStatByDepAndYear($_POST["mpDep"],$_POST["mpDate"],$_COOKIE["bd_select"]);
    }

    // Convertir le tableau en JSON et l'afficher
    echo json_encode($PM_date);
}

if (($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["mpDep"]) && isset($_POST["mpYear"]) && isset($_POST["mpMonth"])) {
    $m = Model::getModel();

    $pm_Month=$m->getPaymentMethodStatByMonthAndYear($_POST["mpDep"],$_POST["mpYear"],$_POST["mpMonth"],$_COOKIE["bd_select"]);

    // Convertir le tableau en JSON et l'afficher
    echo json_encode($pm_Month);
}

if (($_SERVER["REQUEST_METHOD"] == "POST") && isset($_POST["mpDep"]) && isset($_POST["mpWeekDate"])) {
    $m = Model::getModel();

    $date=explode("/", $_POST["mpWeekDate"]);
    $jours=weekToDay(intval($date[0]),intval($date[1]), intval($date[2]));
    $week_start=$jours[0];
    $week_end=$jours[count($jours)-1];

    $mp_weekdate=$m->getPaymentMethodStatByWeek($_POST["mpDep"],$week_start,$week_end,$_COOKIE["bd_select"]);

    // Convertir le tableau en JSON et l'afficher
    echo json_encode(array($week_start,$mp_weekdate));
}

```

Extrait code d'un graphique

```

function caValiderFormulaire() {
    var xhr = new XMLHttpRequest();
    xhr.open('POST', '../Controllers/Controller_home.php', true);

    xhr.onreadystatechange=function(){
        let input_year=document.getElementById('CA_input_year');
        let input_month=document.getElementById('CA_input_month');
        let input_week=document.getElementById('CA_input_week');
        let input_date=document.getElementById('CA_input_date');
        if(xhr.readyState==4 && xhr.status==200){
            response=JSON.parse(this.responseText);

            graphique=Chart.getChart('CA_chart');
            if(graphique){
                graphique.destroy();
            }

            if(input_year.style.visibility=="visible"){
                // Définir les données pour le graphique
                var data = {
                    labels: ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre'],
                    datasets: [{
                        label: "Chiffre d'affaires "+document.getElementById('CA-Select-Dep').value+" "+input_year.value,
                        backgroundColor: "red",
                        borderColor: "black",
                        data: [reponse[0].Montant_total, reponse[1].Montant_total, reponse[2].Montant_total, reponse[3].Montant_total, reponse[4].Montant_total, reponse[5].Montant_total, reponse[6].Montant_total]
                    }]
                };
            }
            else if(input_month.style.visibility=="visible"){
                label_reponse=[];
                data_reponse=[];
                reponse[0].forEach((donnée,index)->{
                    var date=reponse[1][index][0];
                    date=date.split('-');
                    var day=date[2];
                    var month=date[1];
                    var year=date[0];
                    data_reponse.push(donnée.Montant_total);
                    label_reponse.push("Semaine du "+day+month+year);
                })
            }
            var data = {
                labels: label_reponse,
                datasets: [{
                    label: "Chiffre d'affaires",

```



```

var data = {
  labels: label_reponse,
  datasets: [{
    label: "Chiffre d'affaires",
    backgroundColor: 'red',
    borderColor: 'black',
    data: data_reponse
  }]
};

}
else if(input_week.style.visibility=="visible"){
  label_reponse=[];
  data_reponse=[];
  reponse[0].forEach((donnée,index)=>{
    var date=new Date(reponse[1][index]);
    var date_name='';
    var day=date.getDay();
    var month=date.toLocaleString('fr-FR', { month: 'long' });
    var year=date.getFullYear();

    data_reponse.push(donnée.Montant_total);
    if(date.getDay()==1){
      date_name='Lundi';
    }
    else if(date.getDay()==2){
      date_name='Mardi';
    }
    else if(date.getDay()==3){
      date_name='Mercredi';
    }
    else if(date.getDay()==4){
      date_name='Jeudi';
    }
    else if(date.getDay()==5){
      date_name='Vendredi';
    }
    label_reponse.push(date_name+' '+day+' '+month+' '+year);
  })
  var data = {
    labels: label_reponse,
    datasets: [{
      label: "Chiffre d'affaires",
      backgroundColor: 'red',
      borderColor: 'black',
      data: data_reponse
    }]
  }
}

```

```

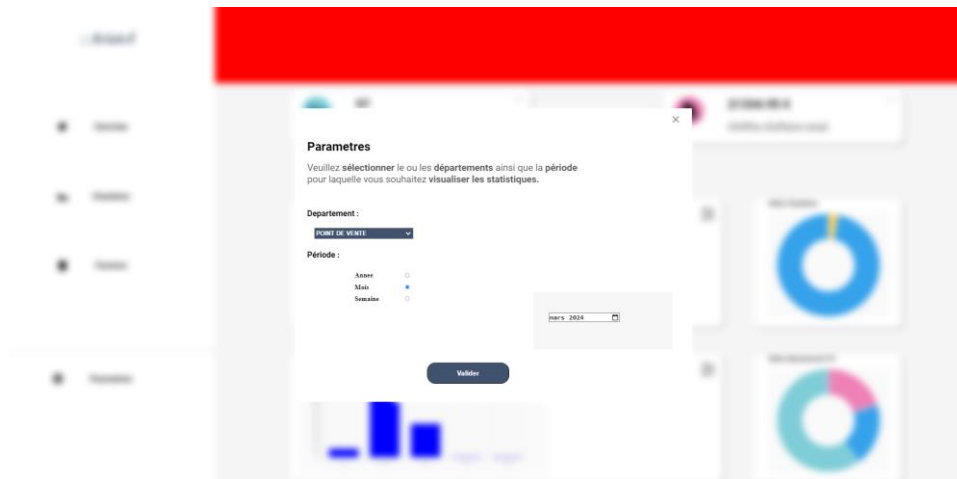
// Configurer les options du graphique
var options = {
  responsive: true,
  maintainAspectRatio: false,
  scales: {
    x: {
      ticks: {
        font: {
          size: 17,
        }
      },
    },
    y: {
      beginAtZero: true
    }
  },
  plugins: {
    tooltip: {
      titleFont: {
        size: 20
      },
      bodyFont: {
        size: 15
      }
    }
  }
};

// Récupérer le canvas
var canvas = document.getElementById('CA_chart');
ctx=canvas.getContext('2d');

// Créer le graphique
var myChart = new Chart(ctx, {
  type: 'bar',
  data: data,
  options: options
});
}
}

```

Annexe 21 - Tableau de bord Formulaire (non finalisé)



Annexe 22 - fonction découperMois

```
function decouperMois($year, $month) {
    $ListeSemaines = [];
    $firstDayOfMonth = strtotime($year . '-' . $month . '-01');
    $lastDayOfMonth = strtotime('last day of ' . $year . '-' . $month);

    $currentDay = $firstDayOfMonth;
    $weekStartDate = $currentDay; // Premier jour du mois = début semaine.

    while ($currentDay <= $lastDayOfMonth) {
        if (date('N', $lastDayOfMonth) == 7) { //
            $lastDayOfMonth-=2;
        }
        elseif(date('N', $lastDayOfMonth) == 6){
            $lastDayOfMonth-=1;
        }
        // Si c'est le jour est lundi, on commence une nouvelle semaine.
        if (date('N', $currentDay) == 1) {
            $weekStartDate = $currentDay;
        }
        // Si c'est un vendredi ou le dernier jour du mois, c'est la fin de la semaine
        if (date('N', $currentDay) == 5 || $currentDay == $lastDayOfMonth) {
            $ListeSemaines[] = [
                date('Y-m-d', $weekStartDate),
                date('Y-m-d', $currentDay)
            ];
            $weekStartDate +=2; // Passer au lundi suivant
        }

        // Passage au jour suivant
        $currentDay = strtotime('+1 day', $currentDay);
    }

    return $ListeSemaines;
}
```

Annexe 23 - fonction vérification de l'expiration du token de connexion

```
function verification()
{
    $n = $_SERVER['HTTP_REFERER'];
    $userToken = $_SESSION['token']; // recupere le token qui est dans la bd
    if ((isset($userToken) && ($userToken['token'] != '')) && !isset($_COOKIE['token']) && !isset($_COOKIE['user'])) { // Vérifie l'existence des variables d'environnement
        //si modification
        if ($COOKIE['token'] != $userToken['token'] || !isset($_COOKIE['expire']) || $_COOKIE['expire'] < time()) {
            setcookie('user', $_COOKIE['user'], time() + 1000, "/");
            setcookie('token', $_COOKIE['token'], time() + 1000, "/");
            setcookie('expire', time() + 1000, 0, "/"); // le nouveau cookie pour contrôler l'expiration
        }
    } else {
        header('Location: index.php'); //Renvoie sur la page d'accueil si token en cookie différent de celui en bd
        exit;
    }
}
```