

RAPPORT DE STAGE

Théo BOUTROUX

08/01/2024 - 09/02/2024

Tuteur : Damien LE SCIELLOUR

Formation : EPSI Nantes – SN2 (option BTS SIO SLAM)

Entreprise d'accueil : **TIBCO Services** – Le Bois Cholet,
BP9 2 Route de la Forêt Bâtiment,
44860 Saint-Aignan-Grandlieu

Table des matières

Remerciements	3
Introduction	4
I – Présentation de l’entreprise	5
A – TIBCO Services	5
B – L’implantation	5
C – Les Clients	6
D – TIBCO en chiffres	7
E – Politique de l’entreprise	8
II – Présentation du CDC Cybersécurité	9
A – Histoire du service	9
B – Les Clients du SOC	9
C – Les Outils	10
D – Les prestations	10
E – L’équipe	11
III – Présentation de l’activité	12
IV – La Mission	14
💡 Qu’est-ce qu’une main courante ?	14
En quoi cette application est-elle utile ?	15
Fonctionnement de l’application	16
Déroulement de la mission	16
Difficultés rencontrées	30
V – Conclusion	31
VI – Bibliographie	33
VII – Annexes	35
Annexe 1 : Lettre de recommandation	35
Annexe 2 : “Visites à venir”	36
Annexe 3 : La page d’accueil de l’interface siem	38
Annexe 4 : La documentation de l’application	39

Remerciements

Avant de plonger dans le récit de mon stage, je voudrais juste remercier les personnes qui ont rendu cette expérience vraiment géniale. Leur soutien et leurs conseils ont vraiment été le petit plus qui a rendu tout cela possible.

Je tiens à remercier :

L'équipe de TIBCO Services, pour leur chaleureux accueil lors de mon arrivée et tout au long de mon stage.

Damien LE SCIELLOUR, Responsable de production (**RP**) , pour son accueil et sa sympathie.

Mickaël ARGANT, Responsable Opérationnel (**RO**), qui m'a donné l'opportunité de réaliser ce stage.

Armel SOUFFRAN, Dev SecOps, pour son aide précieuse, son partage d'expérience et ses conseils tout au long de mes missions.

L'ensemble du Service Cybersécurité, pour leur accueil et leur soutien durant la totalité du stage.

Le personnel de mon école (EPSI - Nantes), sans qui je n'aurais pas eu la chance de faire ce stage.

Un immense merci à tous ceux qui ont contribué à rendre cette expérience de stage aussi enrichissante et stimulante. Leur présence et leur engagement ont véritablement donné vie à cette période inoubliable.

Introduction

Ce rapport a pour objectif de relater mon expérience professionnelle lors de mon stage de deuxième année, qui s'est déroulé sur une période de 5 semaines, du 08/01/2024 au 09/02/2024, dans le cadre de ma formation à l'EPSI - Nantes.

J'ai pu réaliser ce stage chez TIBCO Services, une entreprise spécialisée dans les services numériques fondée en 1984. Elle offre de nombreuses solutions à leurs clients telles que la sécurité de leurs systèmes d'information ou encore du déploiement et de la maintenance chez ces derniers.

Pour mener à bien mon stage, j'ai adopté une approche méthodique. J'ai commencé par me familiariser avec les outils utilisés par le service de cybersécurité ainsi qu'avec les processus internes de l'entreprise. J'ai également suivi les directives données par mon tuteur pour réaliser mes missions afin de répondre à ses besoins.

Dans le cadre des missions qui m'ont été confiées, j'ai utilisé les ressources qui m'étaient mises à disposition par TIBCO ainsi que celles disponibles en ligne. J'ai également adopté une approche collaborative en travaillant avec mes collègues et en les sollicitant pour leur expertise. Enfin, j'ai également fait preuve d'autonomie en prenant des initiatives et en proposant des solutions.

En optant pour Tibco, mon objectif principal était de concrétiser les connaissances acquises de manière autodidacte et à l'EPSI. Ce stage représente une opportunité précieuse pour mettre en pratique mes compétences dans le domaine du numérique au sein d'une entreprise renommée. Il m'a également permis de renforcer mon intérêt pour le développement informatique et d'approfondir mes connaissances dans ce domaine en constante évolution.

I – Présentation de l'entreprise

A – TIBCO Services

Fondée en 1984 par M. Gérard Le Calvé, Tibco développe une première filiale dès 1989 pour vendre du matériel en support des prestations fournies par la société mère. En 2000, pour répondre à la demande du marché, Tibco propose une offre globale de services en se développant sur le marché de l'infogérance modulaire.

En 2017, Tibco acquiert, à travers sa filiale Tibco Télécoms, la société Networks-Technologies, dont l'activité principale est la maintenance des réseaux de télécommunications pour le compte d'équipementiers. Cette acquisition permet notamment au groupe de se renforcer sur les marchés d'exploitation et de maintenance des réseaux de télécommunication.

TIBCO met en place de nombreuses actions auprès des professionnels telles que :

- Faire évoluer les réseaux
- Cyberdéfendre les Systèmes d'Information des clients
- La Digital Workplace
- La souveraineté du Cloud et des Infrastructures

B – L'implantation



TIBCO compte plus de 2000 salariés répartis sur 113 points de présence répartis sur toute la France.

Parmi ces derniers, on retrouve :

- 4 centres de services : Nantes, Lens, Laval, Lyon
- 3 centres logistiques : Nantes, Lens, Paris
- 3 centres de support utilisateurs : Nantes, Lens, Laval
- 1 SOC / NOC certifié ISO 27001 : Nantes

TIBCO vise principalement les TPE/PME de proximité, c'est-à-dire les petites et moyennes entreprises situées localement. Ces entreprises constituent une cible importante pour TIBCO car elles représentent un vaste marché de clients potentiels qui ont besoin de solutions informatiques efficaces pour améliorer leurs opérations commerciales. En se concentrant sur les TPE/PME de proximité, TIBCO peut offrir des solutions adaptées à leurs besoins spécifiques, tout en fournissant un service personnalisé et un soutien local. C'est pourquoi l'entreprise est implantée dans toute la France, afin d'être proche de ses clients et de pouvoir répondre rapidement à leurs demandes et attentes. Cette proximité géographique permet à TIBCO de renforcer ses relations avec les entreprises locales, de mieux comprendre leurs défis et leurs priorités, et de leur offrir des solutions sur mesure qui les aident à prospérer dans un environnement concurrentiel en constante évolution.

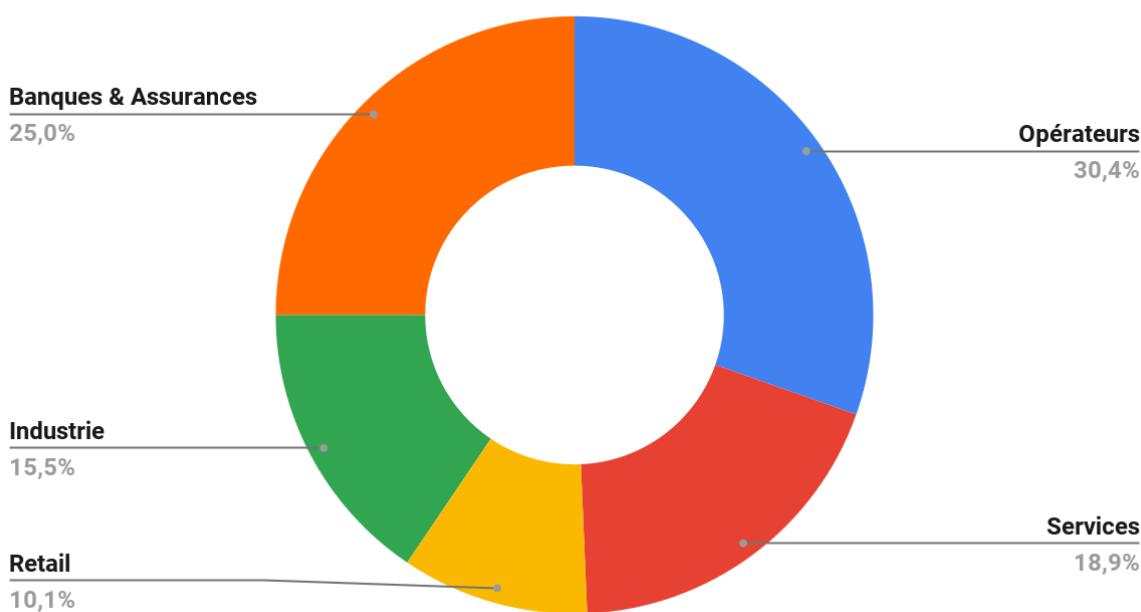
C – Les Clients

Les clients de TIBCO proviennent d'une variété de secteurs, notamment les banques et assurances, les opérateurs télécom, les services, l'industrie et le retail. Avec un nombre impressionnant de plus de 1700 clients à travers toute la France, TIBCO est reconnu comme un partenaire de confiance pour les entreprises cherchant à optimiser leurs opérations, à améliorer leurs performances et à rester compétitives dans un environnement économique en constante évolution. Ces clients représentent un large éventail d'industries et de marchés, et ils font confiance à TIBCO pour leur fournir des solutions innovantes et des technologies de pointe qui répondent à leurs besoins spécifiques et leur permettent d'atteindre leurs objectifs commerciaux.

D – TIBCO en chiffres

Les chiffres de TIBCO témoignent de sa solidité financière et de sa croissance constante. Avec 51 millions d'euros de fonds propres, l'entreprise dispose d'une base financière solide qui lui permet d'investir dans l'innovation et le développement de nouvelles solutions. De plus, un chiffre d'affaires dépassant les 150 millions d'euros atteste de la confiance accordée par ses clients et partenaires, ainsi que de la pertinence de ses offres sur le marché des logiciels et services informatiques. Ces chiffres démontrent l'engagement continu de TIBCO à fournir des solutions de connectivité et d'analyse de données de premier plan, tout en renforçant sa position en tant que leader dans son domaine.

Répartition des recettes par secteur



Graphique représentant la répartition du chiffre d'affaire par secteur d'activité

Sur l'année 2023, TIBCO c'est :

- **3 000 000** de Tickets traités
- **700 000** Interventions de maintenance
- **35 000** Équipements supervisés
- **400 000** Actes de déploiement
- **75 000** Utilisateurs protégés

E – Politique de l'entreprise

TIBCO place la responsabilité sociale et environnementale au cœur de ses priorités. Avec une notation de 837/1000 au label numérique responsable (3ème meilleure note du label), ils se positionnent fièrement comme l'un des leaders dans ce domaine. Leur engagement envers l'environnement se manifeste à travers toutes leurs activités et leurs solutions. Ils s'efforcent de fournir des solutions durables qui allient une forte compétitivité économique à une conscience écologique.

Dans le cadre de leurs offres et prestations écoresponsables, ils mettent en œuvre une approche holistique qui intègre des pratiques innovantes telles que l'utilisation de véhicules électriques, des interventions à distance et la promotion de la durée de vie des matériels. Leur démarche vise à réduire non seulement leur propre impact environnemental, mais aussi celui de leurs clients. En favorisant le réemploi des matériels numériques et en encourageant des usages numériques responsables, ils contribuent activement à la préservation de l'environnement et à la lutte contre le changement climatique.

Chez TIBCO, ils croient en une approche intégrée qui conjugue performance, confort et sobriété, tout en favorisant la fierté de leurs collaborateurs à contribuer à un avenir durable. Leur stratégie éco responsable et leur démarche RSE (Responsabilité Sociétale des Entreprises) témoignent de leur engagement envers l'amélioration continue de leur empreinte écologique, économique et sociale dans le domaine du numérique.

II – Présentation du CDC Cybersécurité

A – Histoire du service

L'évolution du service de cybersécurité au sein du CDC est marquée par plusieurs étapes clés depuis ses débuts. En 2018, le département a initié la mise en œuvre d'un Proof of Concept (POC) du SOC TIBCO, posant ainsi les bases de son infrastructure de sécurité opérationnelle. L'année suivante, en 2019, marque un tournant majeur avec la pleine mise en service du SOC, accompagnée de l'obtention de la certification ISO 27001, attestant de la conformité aux normes de sécurité internationales. En 2020, le service a élargi son offre avec la création de Cyberdéfense Sentinelle, répondant ainsi aux besoins croissants en matière de protection des données et des infrastructures numériques. Cette expansion s'est poursuivie en 2021 avec l'ajout de dix nouveaux clients réguliers au SOC, démontrant la confiance accordée par les entreprises au service. Les années suivantes ont été marquées par une croissance exponentielle, avec le déploiement de versions améliorées de Sentinelle en 2022 et 2023, accompagnées du déploiement d'Alacrité, témoignant de l'engagement continu du service à rester à la pointe de l'innovation et à répondre aux besoins croissants en matière de sécurité numérique, avec un portefeuille de clients qui a atteint 30 en 2023.

B – Les Clients du SOC

Le SOC du CDC Cybersécurité dessert une large gamme de clients provenant de divers secteurs, reflétant ainsi sa polyvalence et sa capacité à répondre aux besoins de différents types d'organisations. Parmi ses clients figurent des entreprises de l'industrie agro-alimentaire, des collectivités locales telles que les conseils départementaux et les communautés d'agglomération, ainsi que des mutuelles, des coopératives agricoles, des associations et des centres hospitaliers. Cette diversité témoigne de l'adaptabilité du service aux défis spécifiques rencontrés dans des environnements variés. Les clients du SOC varient en taille, avec

des effectifs allant de 100 à 5 000 postes, ce qui démontre sa capacité à s'adapter aussi bien aux petites structures qu'aux grandes organisations. Les contrats avec ces clients sont établis sur des périodes allant de 1 à 3 ans, garantissant ainsi une relation à long terme et un engagement continu dans la protection des systèmes d'information et des données sensibles de chaque client.

C – Les Outils

Au sein du CDC Cybersécurité, les équipes du SOC s'appuient sur une combinaison d'outils technologiques robustes pour assurer une surveillance et une protection efficaces des environnements numériques de ses clients. Parmi ces outils figurent le SIEM Fortinet, qui constitue le cœur du système en intégrant la collecte, la corrélation et l'analyse des données de sécurité en temps réel. En complément, le service utilise des outils tiers développés en interne, offrant ainsi une personnalisation adaptée aux besoins spécifiques des clients. Des solutions open source, reposant sur des bases CVE, sont également intégrées pour une veille active des vulnérabilités et une réponse rapide aux menaces émergentes. Pour détecter les comportements suspects, le SOC s'appuie sur des analyses comportementales à l'aide de l'UEBA (User and Entity Behavior Analytics). Enfin, pour les clients disposant de moins de 250 postes, le SOC propose une solution EDR (Endpoint Detection and Response) WithSecure, offrant une protection avancée au niveau des points d'accès critiques du réseau. Cette combinaison d'outils diversifiés garantit une couverture complète des menaces potentielles et une réactivité optimale face aux incidents de sécurité.

D – Les prestations

Le service Cybersécurité de TIBCO propose une gamme variée de six prestations essentielles pour répondre aux besoins complexes de sécurité informatique des entreprises. Ces services comprennent le SOC (Security Operation Center) pour la surveillance proactive des menaces, le NOC (Network Operation Center) pour la gestion optimale des réseaux, le MCS

(Maintien en Condition de Sécurité) Fortinet pour une protection continue des infrastructures, la FIR (Force d'Intervention Rapide) pour une réaction immédiate aux incidents, le PM (Patch Management) Windows pour assurer la sécurité des systèmes avec des mises à jour régulières, et l'AM (Antivirus Management) pour une gestion efficace des logiciels antivirus. Découvrez ci-dessous une illustration de nos services en action.



E - L'équipe

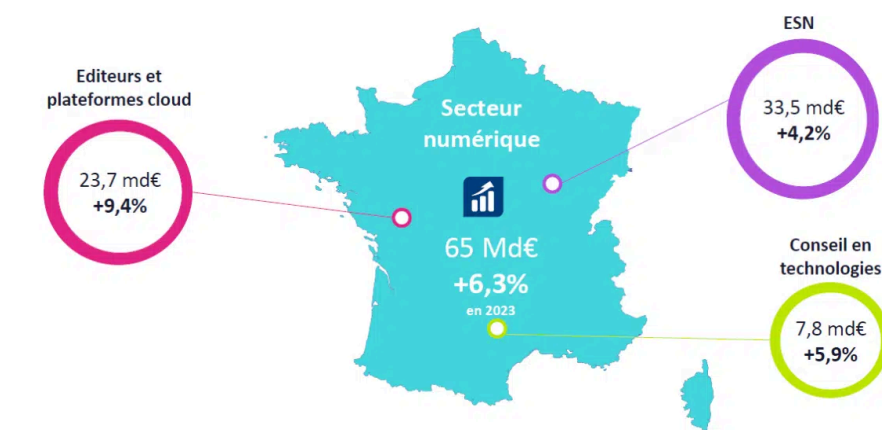
L'équipe de cybersécurité de TIBCO est organisée de manière à garantir une surveillance et une protection efficace des infrastructures numériques de nos clients. Le NOC dédié compte sept tib's (salariés), assurant une attention personnalisée et une gestion proactive des réseaux spécifiques à chaque entreprise. Trois tib's sont dédiés au NOC mutualisé, fournissant un support partagé pour plusieurs clients. Trois autres tib's se concentrent sur le PM/AM/Remédiation (MCS), assurant ainsi une maintenance continue et des actions correctives rapides. Le SOC mutualisé est composé de deux analystes N3, deux analystes N2 et trois analystes N1, travaillant en étroite collaboration pour détecter, analyser et répondre aux menaces de manière efficace. Un responsable opérationnel, un pilote de transition et un DevOps veillent au bon fonctionnement des opérations et à la transition fluide entre les différents services. Enfin, cinq ingénieurs de prestation et deux coordinatrices assurent le support

opérationnel et la coordination des activités, sous la direction d'un manager dédié. Cette répartition d'équipe garantit une approche holistique et réactive pour répondre aux défis complexes de la cybersécurité avec professionnalisme et expertise.

III – Présentation de l'activité

TIBCO Services est solidement ancrée dans le marché dynamique et en pleine croissance des Entreprises de Services du Numérique (ESN). En tant qu'acteur majeur de ce secteur, TIBCO Services s'engage à fournir des solutions innovantes et des services de haute qualité qui répondent aux besoins complexes de ses clients en matière de transformation numérique. Forte de son expertise technologique et de son engagement envers l'excellence opérationnelle, TIBCO Services joue un rôle essentiel dans la création de valeur pour ses clients et dans la croissance continue du marché des ESN.

En 2023, le marché des ESN continue de prospérer avec l'organisation professionnelle du numérique en France. Les prévisions indiquent une croissance du secteur numérique, avec une augmentation estimée à 6,3% pour cette année, comparé à 2022 où la croissance était de 5,9%. Cette progression témoigne de la vitalité du marché du numérique en France, qui se positionne en 3^e position en Europe en termes de croissance, derrière l'Espagne et le Royaume-Uni. Les principaux métiers du secteur connaissent une augmentation du chiffre d'affaires : les éditeurs et plateformes cloud avec une croissance de 9,4%, les ESN avec 4,2%, et le conseil en technologies (ICT) avec 5,9%. Le marché global du numérique est estimé à 65 milliards d'euros, dont les ESN représentent 51,5% avec un chiffre d'affaire de 33,5 milliards d'euros.



La croissance du secteur du numérique en 2023 (Source : boondmanager.com)

Cette croissance est soutenue par plusieurs facteurs, notamment l'essor du cloud, le Big data, les services IoT, la sécurité et la transformation digitale. Les entreprises clientes des acteurs du numérique accordent une importance particulière à la sécurité du système d'information, à l'amélioration de l'expérience client et à l'analyse des données. Par ailleurs, le numérique responsable gagne en importance, avec de nombreuses entreprises mettant en œuvre des actions spécifiques en faveur d'une approche responsable. En parallèle, le secteur du numérique continue de créer des emplois, avec plus de 47 000 emplois créés en 2022, bien que la pénurie de talents dans certains domaines clés reste un défi. Les entreprises adoptent des stratégies pour fidéliser leurs talents, telles que le travail à distance, les formations et les plans d'évolution de carrière. La montée en puissance du numérique responsable constitue également un levier de croissance, avec de nombreuses entreprises intensifiant leurs actions en faveur d'une approche responsable.

IV – La Mission

L'objectif de mon stage de seconde année était de me créer une nouvelle expérience professionnelle au sein d'une entreprise du secteur dans lequel j'étudie. De plus, il m'a permis d'appliquer les compétences que j'ai pu acquérir jusqu'ici et d'en acquérir de nouvelles. Durant ce dernier, j'ai été amené à travailler sur une mission majeure : la main courante.

 **Attention, certaines informations présentes dans les images sont pixélisées dans le cadre de la norme ISO 27001.**

Qu'est-ce qu'une main courante ?

Dans le cadre de la norme ISO 27001, la main courante, ou journal des événements, occupe une place centrale dans la gestion de la sécurité de l'information au sein des organisations. Il s'agit d'un enregistrement continu et chronologique des activités, incidents et événements significatifs liés à la sécurité de l'information. La main courante est mise à jour en temps réel ou à intervalles réguliers pour refléter de manière précise les événements survenus, et elle contient des informations détaillées telles que la date et l'heure, la description de l'événement et les actions prises pour y remédier. La tenue d'une main courante permet aux organisations de surveiller et de gérer efficacement les incidents de sécurité, de détecter les tendances et les problèmes récurrents, et de prendre des mesures correctives pour renforcer la sécurité de l'information, contribuant ainsi à démontrer leur conformité aux exigences de la norme ISO 27001.

En quoi cette application est-elle utile ?

Lorsqu'une personne extérieure (Exemple : Un client) entre dans le service cybersécurité (**CDC**), elle doit avoir validé 3 conditions indispensables sinon l'accès lui sera refusé.

- Un NDA valide : Un NDA (accord de confidentialité) est un document qui exige que la personne qui le signe protège la confidentialité des informations sensibles ou confidentielles lors de sa visite. Sa durée de validité est de 5 ans.
- Un horaire de présence doit être établi pour que toutes les parties sachent quand des personnes extérieures sont présentes dans le CDC.
- Une validation de la part du Responsable de production ou du Responsable opérationnel qui consiste à créer une autorisation d'accès via la main courante.

Une application a donc été créée afin de visualiser un planning des visites à venir ainsi que pour vérifier si une personne a le droit ou pas d'entrer dans le CDC à un instant T. Cependant, cette dernière n'était pas assez complète, ce qui obligeait le RO et le RP à travailler sur plusieurs plateformes différentes pour gérer la signature des NDA, la planification des visites...

L'objectif de ma mission était donc d'ajouter des fonctionnalités sur cette application déjà existante afin de centraliser un maximum de données et de réunir toutes les fonctionnalités au sein d'une seule plateforme, la main courante.

Fonctionnement de l'application

Afin de faciliter l'accès à la main courante, cette dernière a été développée en Python et est affichée sur une interface Web. Nous avons utilisé plusieurs technologies pour mener à bien le projet. On retrouve par exemple les langages de base en ce qui concerne le Web (HTML, CSS, JavaScript), mais également des frameworks¹ CSS et JS tels que JQuery et Bootstrap. Nous avons également eu l'occasion de travailler avec plusieurs API permettant de recueillir des données comme celle de DocuSign² et celle de Sharepoint. Enfin, pour stocker ces dernières, nous avons utilisé une base de données avec SQL Server.

Déroulement de la mission

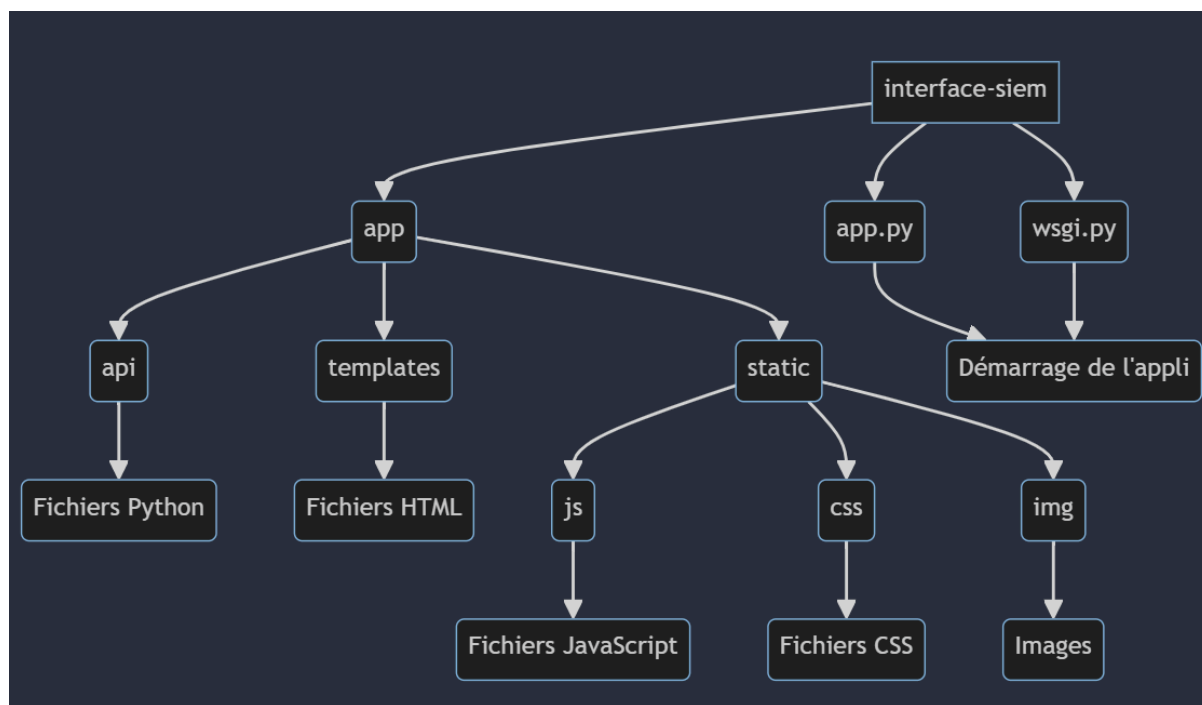
Lors de mon arrivée, j'ai eu l'opportunité de découvrir les différents bâtiments et services de l'entreprise, ce qui m'a permis de prendre mes repères et de faciliter mon intégration dès le début. Puis, nous avons eu une réunion avec les responsables du service (RO et RP) afin de parler de la mission principale de mon stage, et de réaliser un cahier des charges. Durant cette dernière, j'ai pu me familiariser avec l'outil déjà existant et comprendre le protocole complet d'accès au service. J'ai aussi pu découvrir les technologies utilisées auparavant pour développer cette application, à savoir du Python et des langages du Web.

Par la suite, j'ai pu installer les outils dont j'ai eu besoin tout au long du stage sur ma machine. J'ai donc installé Python et les dépendances de base, Git, mais aussi Visual Studio Code afin de développer dans une IDE performant. Si j'ai choisi cet IDE, c'est parce qu'il offre la possibilité de développer dans n'importe quel langage grâce à de nombreuses extensions mise à disposition des développeurs au sein même du logiciel. De plus, il permet d'avoir directement un terminal, ce qui permet d'exécuter des commandes sans avoir à quitter l'éditeur.

Le lendemain, j'ai rencontré Armel, développeur de la première version du projet. Nous avons eu l'occasion d'échanger à propos de ce dernier et il m'a aussi expliqué le fonctionnement du repo Gitlab utilisé pour stocker le code de l'appli.

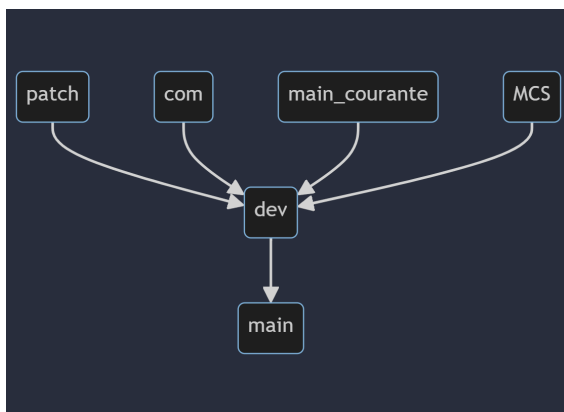
Fonctionnement du repo Gitlab :

Le code de la main courante était stocké sur un repo Gitlab nommé "interface-siem". Ce dernier contient plusieurs applications utilisées exclusivement en interne par les diverses équipes du service. Vous trouverez ci dessous l'arborescence complète du repo. On y retrouve une arborescence classique Flask, contenant un dossier "static" dans lequel on y met les fichiers CSS, JS et les images. Il y a également un dossier "templates" contenant l'ensemble des fichiers HTML de l'interface. Puis, il y a le dossier "api", qui rassemble les fichiers Python permettant la création des routes pour assurer le fonctionnement de l'application. Enfin, on retrouve les fichiers app.py et wsgi.py, qui permettent respectivement de démarrer l'application en développement, et en production.



Arborescence du repo Gitlab de l'interface siem

Lorsque l'on travaille sur un repo GitHub ou GitLab, il est important de pouvoir versionner son code. Cela permet un déploiement final plus simple et plus rapide. Cependant, lorsque l'on travaille à plusieurs sur un repo, ce qui était le cas durant la durée du stage, on peut être amenés à faire des modifications en même temps, ce qui peut entraîner des problèmes de compatibilité. Pour remédier à ce problème, il faut créer des branches³ afin de pouvoir faire des modifications sans altérer le fonctionnement du projet.



On retrouve ici un schéma qui illustre le fonctionnement du repo du projet. De mon côté, je travaillais sur la branche "main courante" et, quand les fonctionnalités étaient mises à jour, on faisait une "Pull Request"⁴ permettant de mettre à jour la branche dev pour faire des tests de pré-production. Enfin, on réalisait une deuxième "Pull Request" depuis la branche dev vers la main, pour mettre en production les nouvelles fonctionnalités.

Processus de déploiement :

1. On réalise un commit sur la branche main_courante afin d'actualiser le code de cette dernière
2. On fusionne la branche main_courante avec la branche dev pour faire des tests finaux avec le déploiement définitif
3. On fusionne la branche dev avec la branche main pour mettre en production

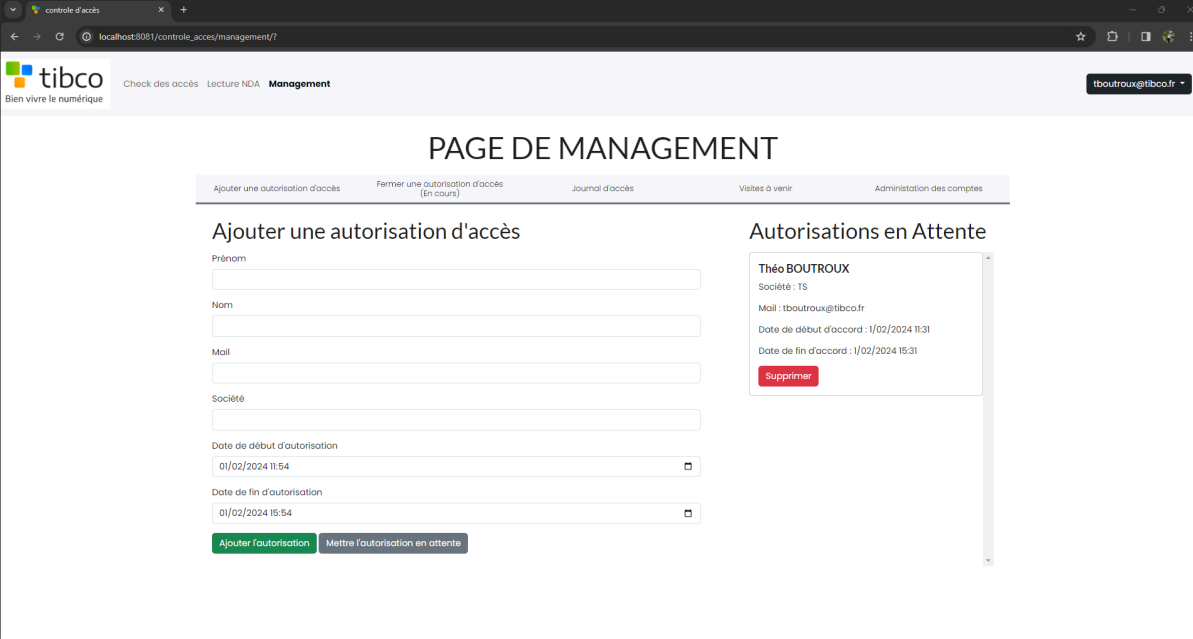


Schéma du fonctionnement des commit etc...

J'ai pu commencer à travailler sur le projet à partir du 3^{ème} jour. La première tâche que j'ai réalisée est une autocomplétion sur des champs de texte dans un formulaire. Cette dernière permettait d'éviter de faire des erreurs lors de la saisie des informations dans les champs. Afin de rendre dynamique cette fonctionnalité, j'ai utilisé JQuery, un framework JavaScript, et plus particulièrement AJAX pour envoyer des requêtes vers le Python quand un caractère était saisi. J'ai également retouché le style du formulaire avec Bootstrap afin d'avoir une meilleure UX⁵ et UI⁶ lors du remplissage de ce dernier.

Les jours qui ont suivi m'ont permis de travailler sur une nouvelle tâche du projet, la gestion des autorisations d'accès. Jusqu'alors, cette page permettait de créer une autorisation afin de programmer une visite client. Elle y regroupait uniquement un formulaire, celui de l'autocomplétion. Mon objectif était de pouvoir sauvegarder en tampon une autorisation, la modifier, et la supprimer en cas de besoin afin de faire gagner du temps lors de l'utilisation de la main courante. En effet, avant l'ajout de ces fonctionnalités, les responsables étaient obligés d'attendre la signature des NDA pour créer ces autorisations. Ils peuvent désormais les créer et les valider uniquement lorsque le NDA sera signé. Pour ce faire, j'ai créé de nouvelles routes en Python permettant de récupérer les informations du formulaire et de les stocker dans un fichier JSON. Parmi ces dernières, on retrouve le nom, le prénom, le mail, la société et les horaires de visite d'une personne. Pour mettre en tampon une autorisation, j'ai également créé un bouton permettant de réaliser cette tâche et qui crée par la suite une carte contenant les informations. Au début, on voulait créer trois boutons sur ces dernières (Ajouter, Modifier, Supprimer), mais on a finalement opté pour implémenter uniquement un bouton "Supprimer" afin d'améliorer l'UX de l'application. Pour modifier une autorisation, il faut simplement cliquer sur la carte correspondant à l'autorisation d'une personne, ce qui remplira automatiquement les champs du formulaire. Il faudra de nouveau enregistrer l'autorisation une fois les informations

mis à jour. Enfin, pour ajouter une nouvelle autorisation, il suffit de cliquer sur un bouton en bas du formulaire (Voir la photo ci-dessous).



Page concernant l'ajout d'autorisation d'accès

Pour conclure cette page, j'ai fini par ajouter une vérification en cas de doublon de visite, ce qui évite de créer plusieurs fois la même visite.

On arrive maintenant à la deuxième semaine du stage. Pour commencer, nous nous sommes interrogés au sein de l'équipe sur la manière dont on allait gérer l'annulation d'une visite pour respecter la norme ISO 27001. On a donc choisi d'ajouter un champ booléen dans la base de données, qui permet d'annuler les visites sans l'effacer dans le but de pouvoir tracer toute activité réalisée sur la main courante.

J'ai donc créé une nouvelle page nommée "Visites à venir" contenant trois onglets différents. Le premier contient un calendrier comprenant les visites à venir, le deuxième permet d'afficher les visites à venir mais sous forme de cartes, et le troisième rassemble un autre calendrier, affichant quant à lui les visites annulées. (Voir les images des pages dans l'annexe 1)

Afin de pouvoir annuler une visite, nous avons choisi d'implémenter un bouton situé en bas des cartes et un autre disponible dans un modal affiché lorsque l'on clique sur un événement du tableau des visites à venir. Cependant, lors du développement, j'ai été confronté à un problème. En effet, les visites enregistrées dans la base de données n'avait pas d'id assigné, ce qui empêchait de différencier correctement chaque visite. Nous avons donc ajouté ce champ au sein de la BDD afin de pallier à ce problème. Si nous n'avions pas réalisé cela, il aurait été probable qu'une autre visite de la même personne soit impactée.

J'ai également été confronté à un autre problème, le dynamisme de la page. Ce problème était causé car ces données ne se rafraichissaient pas automatiquement lorsque j'annulais une visite. Pour faire face à cela, j'ai donc modifié l'envoi des requêtes pour qu'elles me renvoient les nouvelles données dynamiquement.

Après la création de toutes ces fonctionnalités, nous avons mis à jour la branche dev du projet afin de réaliser une batterie de tests sur l'ensemble du projet, afin de vérifier le fonctionnement de ces dernières. J'ai donc corrigé les différents bugs présents tels que des problèmes de rafraîchissement. J'ai également ajouté la possibilité de trier les différents tableaux de l'application. En effet, la main courante contient deux autres pages qui permettent de rassembler de nombreuses informations concernant les autorisations d'accès et sur les entrées et sorties des personnes extérieures au sein du service. (Voir image ci dessous)

The screenshot shows a web browser window with the URL `localhost:8081/controle_acces/management/`. The page title is "PAGE DE MANAGEMENT". It features a navigation bar with "Check des accès", "Lecture NDA", and "Management". A user profile "tboutroux@tibco.fr" is visible in the top right. The main content is a table with the following columns: "Ajouter une autorisation d'accès", "Fermer une autorisation d'accès (En cours)", "Journal d'accès", "Visites à venir", and "Administration des comptes". The table lists several access permissions for a user named "Théo BOUTROUX".

Prénom ↓	Nom ↓	Date de début ↓	Date de fin ↓	Supprimer
		06/02/2023 0:00	06/02/2035 0:00	Supprimer
		02/06/2023 0:00	02/06/2040 0:00	Supprimer
		02/06/2023 0:00	02/06/2040 0:00	Supprimer
Théo	BOUTROUX	01/02/2024 11:31	01/02/2024 15:31	Supprimer
		02/06/2023 0:00	02/06/2040 0:00	Supprimer
		06/02/2023 0:00	06/02/2035 0:00	Supprimer
		02/06/2023 0:00	02/06/2040 0:00	Supprimer
		02/06/2023 0:00	02/06/2040 0:00	Supprimer
		06/02/2023 0:00	06/02/2035 0:00	Supprimer
		06/02/2023 0:00	06/02/2035 0:00	Supprimer
		18/09/2023 15:00	18/09/2028 19:00	Supprimer
		18/09/2023 15:00	18/09/2028 19:00	Supprimer
		18/09/2023 15:08	18/09/2028 19:08	Supprimer

Tableau de la page "Fermer une autorisation d'accès"

Après avoir terminé cette étape de test et de correction, on a pu réaliser la première mise en production de la nouvelle version de la main courante. Pour ce faire, on a donc fusionné la branche dev à la branche main, comme indiqué sur le schéma précédent. Puis, on s'est connecté au serveur qui assure le bon fonctionnement de l'interface siem afin de redémarrer le service et mettre à jour l'application.

Une fois que l'application à été mise en production, j'ai pu commencer à travailler sur un nouveau sujet important, l'authentification sur l'interface. Pour rappel, l'interface siem contient plusieurs applications différentes permettant aux équipes du service de pouvoir réaliser toutes leurs tâches sur la même application. Cependant, cela veut dire que tout le monde peut accéder à toutes les applications, y compris à la section management de la main courante qui est pourtant censée être disponible uniquement pour les responsables. C'est pourquoi, nous avons décidé de mettre en place un système d'authentification et de gestion des comptes au sein de l'interface. Cette solution est temporaire car au sein de l'entreprise, les salariés sont censés utiliser un seul mot de passe, celui de leur compte Windows stocké dans l'Active Directory⁷ de TIBCO. De plus, l'interface ne possède pas de certificat SSL, ce qui veut dire que

l'application est en HTTP et non en HTTPS. Par conséquent, même si l'interface n'est disponible qu'en interne, les données des utilisateurs circulent en clair sur le réseau TIBCO. C'est pourquoi il doivent utiliser un mot de passe destiné uniquement à l'utilisation de cette application.

Pour commencer cette partie, j'ai réalisé une version test à côté du projet afin de me familiariser avec les différents concepts à assimiler pour créer une solution fiable et durable. J'ai donc pu découvrir ce qu'est une session⁸ dans Flask, ainsi que les différentes méthodes de hachage⁹ des mots de passe. Dans notre cas, nous avons choisi d'utiliser la méthode de hachage SHA512, permettant de chiffrer des mots de passe en chaîne d'une longueur de 512 caractères. Une fois cette version bêta fonctionnelle, j'ai donc commencé à implémenter l'authentification au sein de l'interface.

La première étape a été de créer des comptes et une table correspondant aux comptes dans la base de données. On a donc réalisé la table "comptes_interface_siem" qui contient diverses informations sur les personnes et sur les comptes. En effet, elle rassemble le nom, prénom, mail, mot de passe et quadrigramme de la personne, mais aussi la date de création, d'activation et de suppression du compte. Ces dates permettent encore une fois de rassembler toutes les informations sur tous les comptes (actif ou non) afin de respecter la norme ISO 27001. Nous nous sommes basés sur la politique de TIBCO en ce qui concerne la nomenclature des mots de passe. Par conséquent, il doit faire 12 caractères et contenir au moins une minuscule, une majuscule, un chiffre et un caractère spécial (Exemple : Pa\$\$w0rd!234). J'ai par la suite implémenté un affichage dynamique lors de la création du mot de passe permettant de voir si toutes les règles ont été respectées. On retrouve ci-dessous le formulaire d'inscription.

controlé d'accès x controlé d'accès x Task Error: current_app issue x Link - Bootstap v3.3 x Découverte du service REST S... x Plate-forme de partage de fct... x Photos de la main courante - G... x

127.0.0.1:5000/authentication/register

tibco
Bien vivre le numérique

Se connecter S'inscrire

Inscription

Prénom : Nom :

Quadrigramme Akuteo :

Adresse Mail :

Mot de passe :

▲ Au moins 12 caractères
▲ Au moins une majuscule
▲ Au moins une minuscule
▲ Au moins un chiffre
▲ Au moins un caractère spécial (sauf l'antislash)

Confirmez le mot de passe:

Déjà un compte? Connectez-vous ici.

Page d'inscription

La seconde étape a été de créer une page de connexion afin d'accéder à son espace. Pour le moment cela n'a pas d'impact sur l'interface donc il n'est pas nécessaire de créer un compte, à l'exception des responsables, car ils possèdent des accès exclusifs sur la page management de la main courante. J'ai ajouté par la suite la déconnexion de l'application. Puis, après une courte réflexion, on s'est vite rendu compte qu'il serait utile de pouvoir modifier ces informations personnelles en cas de faute de frappe lors de l'inscription. C'est pourquoi une fois connecté, on a accès à une page nommée "Mon compte" permettant de réaliser ceci.

Enfin, il semblait indispensable de créer un système de gestion des comptes. En effet, il est important que seul le personnel du service Cybersécurité puisse avoir la possibilité de créer un compte. C'est pourquoi j'ai créé un nouvel onglet dans la page management, afin de pouvoir réaliser ceci. Son fonctionnement est simple, lorsqu'un compte est créé par un utilisateur, une nouvelle carte contenant deux boutons s'affiche et permet à un administrateur d'accepter ou refuser la création de ce dernier. En fonction de l'action réalisée, un tableau contenant l'ensemble des renseignements se met à jour afin d'assurer la traçabilité

de toutes les actions concernant les comptes. J'ai par la suite ajouté la possibilité de rechercher un compte dans le tableau en cas de besoin. Ce système permet donc de gérer l'accès à l'interface. Tant qu'un utilisateur n'est pas validé ou s'il est archivé, ce dernier ne peut pas se connecter à son espace. Après la réalisation de cette page, l'authentification sur l'interface a donc été finalisée.

On arrive maintenant à la fin de la troisième semaine de stage et j'ai une nouvelle mission à réaliser sur la main courante. En effet, je dois maintenant récupérer la date de signature d'un NDA pour l'afficher dynamiquement dans des cartes sur une page nommée "Lecture des NDA" contenant le nom, le prénom et un bouton renvoyant vers le NDA d'une personne afin de vérifier sa validité. Cette partie m'a permis de découvrir plusieurs nouveaux concepts, les Threads¹⁰ et les WebSockets¹¹.

La première étape a été de réussir à extraire une date d'un fichier PDF. Après de multiples tentatives en utilisant différents modules Python pour réaliser ceci, j'ai enfin réussi à extraire une date grâce au module pdfminer.six. La fonction que j'ai créé afin de récupérer cette date permet de lire le texte d'un fichier PDF et de récupérer une date sous une certaine forme grâce à une Regex¹². Elle permet également de formater la date pour la rendre exploitable (YYYY-MM-DD).

Ensuite, j'ai créé de nouveau une version test afin de me familiariser avec les Threads et les WebSockets en Python. Les NDA étant stockés sur un Sharepoint, j'ai également dû apprendre le fonctionnement de son API¹³. Le fonctionnement de cette version était simple. En effet, je lançais une requête vers l'API de Sharepoint lorsque j'étais sur une page d'accueil via un Thread, et les données se mettaient à jour dynamiquement dans une autre grâce à un WebSocket (ici Socketio). Dans le but de gagner du temps et de la performance, j'ai choisi au début de stocker la date de signature et les données concernant les personnes dans un fichier JSON. Ce système permettait alors de ne pas envoyer de requête pour chaque

personne s'il y avait déjà l'information sur cette dernière. Cela permettait de réduire le temps d'exécution, passant alors d'environ 2 minutes à 40 secondes. De plus, cela limitait le nombre de requêtes à réaliser, passant d'environ 150 à quelques-unes si jamais il y avait des nouvelles données à insérer dans le fichier JSON .

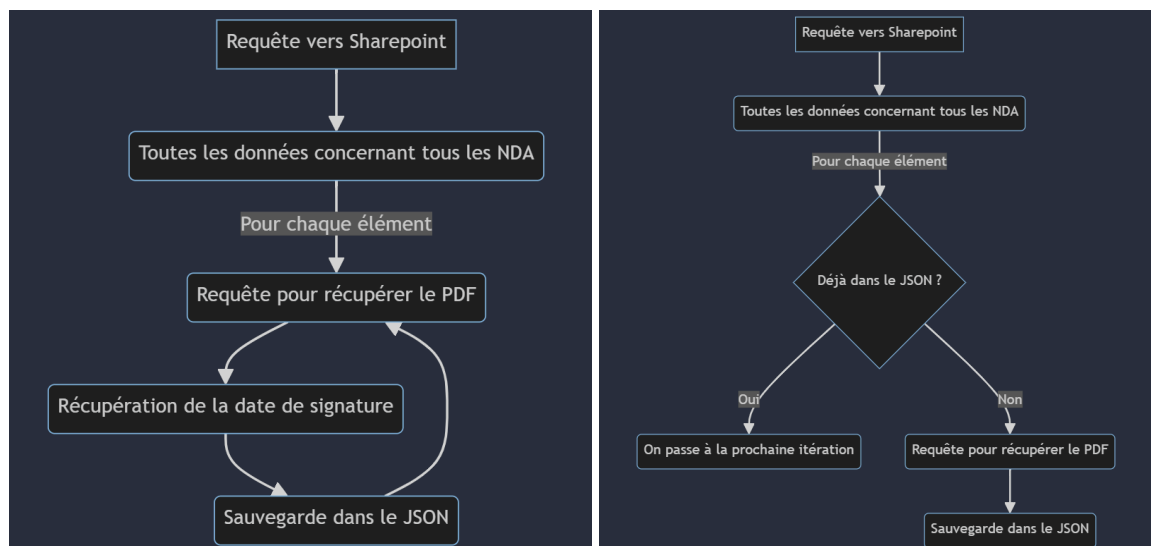
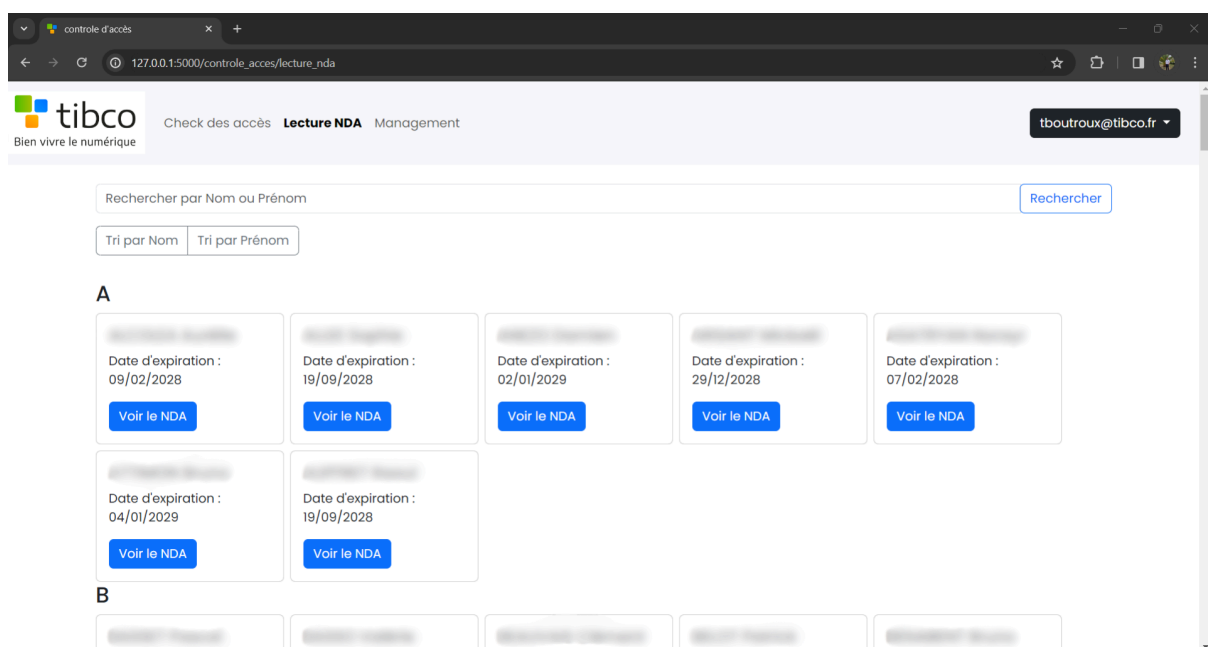


Schéma du fonctionnement de la récupération des données avant et après optimisation

Une fois ma version test opérationnelle, j'ai pu l'implémenter dans l'application. Pour commencer, j'ai dû faire un changement concernant le stockage des données. En effet, on a créé une nouvelle table qui allait recueillir celles stockées auparavant dans le JSON. Cela a donc engendré quelques modifications majeures au sein de la fonction permettant de récupérer les données. Une fois ces changements opérés, j'ai été confronté à plusieurs problèmes. Premièrement, il faut savoir que pour ajouter cette fonctionnalité, j'ai travaillé sur plusieurs fichiers Python différents. Il y avait celui comprenant les routes¹⁴ Flask ou encore celui qui rassemblait les autres fonctions utilisées dans ces dernières. Le premier problème que j'ai rencontré a été un souci de portée des variables. Par exemple, lorsque j'appelais une fonction à partir d'une route, certaines fonctionnalités ne marchaient pas car il y avait un problème de compréhension de contexte. Cela veut dire que le programme ne reconnaissait pas la fonction comme faisait partie de mon application Flask. Pour résoudre cet ennui, j'ai donc dû ajouter plusieurs paramètres tels que la configuration de l'application, qui permettait d'accéder aux fichiers de config (comportant toutes les

infos sur les BDD et sur les mots de passe des API) et aux variables spécifiques à l'application (Exemple : Utilisateur connecté, clé secrète...). Le deuxième problème que j'ai rencontré lors de l'implémentation de cette fonctionnalité était le formatage des dates au moment d'insérer les données au sein de la base de données. Pour le résoudre, il suffisait de modifier les dates pour les rendre compatibles avec Python.

Une fois cette fonctionnalité créée, j'ai pu m'occuper du WebSocket. Ce dernier permettait d'injecter dynamiquement la date d'expiration des NDA dans la page ci-dessous grâce à la date de signature récupérée et stockée dans la base de données (Un NDA est valide 5 ans).



Page "Lecture NDA"

Quand le WebSocket a été ajouté, j'ai pu créer une barre de recherche permettant de trouver le NDA d'une personne plus rapidement. J'ai également corrigé les différents bugs qui étaient présents au sein de l'application.

On a désormais pu mettre en production ces nouvelles fonctionnalités. Cependant, lorsque nous avons redémarré le service sur le serveur, plus rien ne fonctionnait. Ce problème était dû au WebSocket, car j'ai dû toucher aux fichiers permettant le démarrage du serveur Flask.

Malgré les différentes tentatives pour régler ce dernier, nous en avons conclu qu'il fallait malheureusement enlever cette fonctionnalité et la remplacer par quelque chose de plus classique, donc non dynamique, ce que j'ai fait par la suite. Quelques heures plus tard, nous avons pu recommencer la mise en production, réussie cette fois-ci. On a pu conclure de cet événement qu'il fallait mieux se préparer à ce genre d'incident. On a donc mis en place une manière de procéder au cas où ce problème venait à surgir de nouveau.

On arrive à la dernière semaine du stage et il me reste une dernière partie à réaliser, la partie Docusign. L'objectif de cette dernière fonctionnalité est de pouvoir, lors de l'ajout d'une nouvelle autorisation d'accès, envoyer un mail à la personne concernée directement au sein de l'application. Pour commencer, j'ai étudié la documentation proposée par Docusign afin de me renseigner sur le fonctionnement de leurs services ainsi que de leur API. Après avoir passé plusieurs heures à regarder des tutoriels et lire de la documentation, j'ai pu comprendre le fonctionnement de l'API. Docusign met, à disposition des développeurs, des outils pour différents langages permettant de simplifier le développement des fonctionnalités utilisant leurs services (appelé SDK). J'ai donc choisi celui offrant la possibilité de développer en Python. Encore une fois, j'ai commencé par travailler sur une version test afin de pouvoir tester avant d'implémenter dans la version finale.

Cependant, avant de pouvoir faire des tests, il fallait configurer l'API. Pour ce faire, il a fallu tout d'abord créer un compte développeur Docusign, ce qui permettait par la suite de créer une application pour récupérer les identifiants nécessaires à la connexion de l'API. Ensuite, il fallait récupérer un code qui permet par la suite de récupérer un token pour se connecter. Il fallait donc envoyer une première requête pour récupérer ce dernier. Puis, il faut envoyer une deuxième requête afin de récupérer le token, cette fois-ci en utilisant le code reçu lors de la première. Cette étape était complexe car la documentation fournie n'était pas très explicite et les noms donnés

au différents mots de passe n'étaient pas les mêmes que ceux habituellement utilisés dans les autres API.

Quand j'ai réussi à me connecter à l'API, j'ai fait quelques requêtes pour m'assurer que cette dernière fonctionnait. J'ai par la suite commencé à vouloir envoyer des mails via ma version test. Pour ce faire, j'ai créé quelques routes Flask et fonctions dans cette version afin de pouvoir tester correctement et dans les mêmes conditions que l'application finale. Je suis rapidement parvenu à des résultats concluants, ce qui m'a permis d'avancer plus rapidement que prévu sur cette étape.

J'ai par la suite déployé cette fonctionnalité dans l'interface siem. Pour ce faire, j'ai ajouté un bouton sur le champ société du formulaire d'ajout d'autorisation permettant d'envoyer le mail à la bonne personne une fois toutes les informations remplies. J'ai également ajouté un tag sur les cartes correspondant aux autorisations en attente afin de voir le statut d'un NDA (En attente, signé, annulé). Tout était prêt à être mis en production, cependant, il aurait fallu un compte développeur appartenant à TIBCO ou au service pour ajouter correctement tout ça, ce qui n'a pas été possible d'avoir à temps. On a donc décidé de créer une nouvelle branche dans laquelle le code à implémenter était stocké afin de pouvoir rapidement mettre à jour la main courante une fois ce compte obtenu.

Enfin, après quelques nouvelles modifications et corrections de bugs, nous avons réalisé la mise en production finale de la main courante le dernier jour de mon stage.

Difficultés rencontrées

Comme énoncé précédemment, j'ai rencontré quelques petites difficultés durant l'intégralité de la mission.

1. Le formatage des dates : Durant le stage, j'ai souvent été confronté à des dates. Cependant, elles n'étaient jamais du même format entre le JavaScript, le Python et la base de données, ce qui a engendré sans cesse des problèmes de compatibilité ou de type.
2. Code déjà existant : Malgré le fait que le code déjà fourni était bien commenté, j'ai eu quelques difficultés de compréhension par moment, dû au fait que nous n'avions pas la même manière de procéder sur certains points.
3. Portée des variables : Pendant l'implémentation du Thread au sein de l'application, on a eu des gros problèmes d'incompréhension de contexte à partir des fonctions, ce qui a fait perdre un peu de temps. Cependant les problèmes ont été résolus correctement en analysant correctement ces derniers.
4. Parsing des fichiers PDF : J'ai dû utiliser plusieurs librairies Python différentes pour enfin réussir à obtenir la date que je recherchais dans les fichiers PDF grâce à pdfminer.six.
5. API Docusign : Il aura fallu plusieurs jours pour comprendre le fonctionnement de l'API et son protocole de connexion complexe.

V – Conclusion

Ces cinq semaines de stage ont été pour moi l'occasion de pouvoir me créer une nouvelle expérience au sein d'une entreprise du numérique. Elles m'ont également permis de pouvoir découvrir de nouveaux concepts en développement tels que le fonctionnement en mode API, l'utilisation d'API REST (Docusign, Sharepoint), ou encore de nouveaux concepts comme les Threads et les WebSockets. De plus, j'ai pu apprendre une nouvelle manière de travailler, notamment grâce au repo Gitlab et aux bonnes pratiques mises en place au sein du code de l'application. Puis, j'ai découvert ce qu'est la norme ISO 27001 et ce qu'il faut faire pour la respecter. Cette expérience m'a permis de participer à toutes les étapes de production d'un projet professionnel, allant du cahier des charges à la mise en production, en passant par le développement et les moments d'échanges avec les demandeurs (les responsables du service).

D'un point de vue personnel, ce stage m'a permis de confirmer mon intérêt pour le développement Web grâce aux nombreuses choses que j'ai pu apprendre et développer durant l'intégralité de ce dernier. De plus, j'ai eu l'occasion de découvrir une nouvelle manière de procéder dans le développement d'un projet, ce qui m'a permis de me professionnaliser encore plus pour mes projets futurs. Puis, grâce au développeur, qui m'a accompagné durant tout le stage, j'ai pu apprendre de nouvelles bonnes pratiques à utiliser dans le code, que j'adopterai désormais dans mes futurs développements d'applications et de sites Web.

D'un point de vue professionnel, ce stage a été pour moi l'occasion de rencontrer des personnes spécialisées dans l'IT, qui ont pu me partager leurs compétences et expériences. Cela m'a également permis de me familiariser avec le monde professionnel. J'ai aussi pu améliorer mes soft skills, éléments clés à avoir en entreprise. Parmi ceux-ci, j'ai pu notamment développer mes compétences en matière de communication et de coopération avec les autres. J'ai eu la chance d'avoir des personnes

passionnées dans le service cybersécurité, qui ont pu m'aider lorsque je rencontrais des difficultés. De plus, j'ai également pu développer ma capacité d'écoute, notamment en écoutant les besoins et les conseils qu'on a pu me transmettre lors de mon stage. Ce stage m'a appris à être plus persévérant pour pouvoir atteindre les objectifs donnés malgré les difficultés rencontrées.

Enfin, vous retrouverez ci dessous la liste des compétences que j'ai pu acquérir ou développer durant mes cinq semaines de stage chez TIBCO :

- HTML : langage utilisé pour la structure des pages de l'interface
- CSS : Pour le style des pages
- Bootstrap : Un framework permettant de styliser les pages plus simplement
- JavaScript : Utilisé pour dynamiser des pages Web
- JQuery : Framework utilisé pour simplifier la syntaxe JavaScript.
- Python : Utilisé pour gérer tout le back-end de l'application développée
- Git/Gitlab : Outil de versionning du code
- SQL Server : Langage utilisé pour les bases de données

VI – Bibliographie

Framework¹ : Un framework est une plateforme de développement logiciel qui offre une structure organisée, des outils et des bibliothèques de code pré-écrits pour faciliter la création d'applications. Il fournit un ensemble de conventions, de bonnes pratiques et de modèles de conception qui permettent aux développeurs de construire des applications de manière efficace et cohérente. En utilisant un framework, les développeurs peuvent accélérer le processus de développement en réutilisant du code existant, en évitant la réinvention de la roue et en se concentrant sur les aspects spécifiques de leur application plutôt que sur des tâches génériques. En résumé, un framework est un outil essentiel qui simplifie et rationalise le processus de développement logiciel.

Docusign² : Il s'agit d'une société américaine spécialisée dans la signature électronique de documents et la gestion des transactions numériques. Dans notre cas, Docusign est utilisé afin d'envoyer et signer les NDA.

Branche³ : Une branche dans Git est une version parallèle du code, permettant aux développeurs de travailler sur des fonctionnalités ou des correctifs isolément, sans affecter directement la version principale du projet. Une fois les modifications testées, elles peuvent être fusionnées dans la branche principale. Les branches facilitent le développement collaboratif en maintenant une séparation claire entre les différents travaux en cours.

Pull Request⁴ : Une pull request est une demande pour intégrer des modifications proposées dans un projet de développement de logiciel, facilitant ainsi la collaboration et la révision avant leur intégration définitive.

UX⁵ : User Experience

UI⁶ : User Interface

Active Directory⁷ : Un Active Directory est un service de répertoire développé par Microsoft, utilisé principalement dans les environnements informatiques d'entreprise pour centraliser et gérer les ressources réseau telles que les utilisateurs, les ordinateurs, les groupes et les politiques de sécurité. Il fournit des fonctionnalités d'authentification, d'autorisation et de gestion des ressources, facilitant ainsi l'administration et la sécurisation des réseaux informatiques.

Session Flask⁸ : Dans Flask, une session est un moyen de stocker des données spécifiques à un utilisateur entre les requêtes HTTP. Cela permet de maintenir l'état de l'application pour un utilisateur donné tout au long de sa navigation sur le site. Flask utilise des cookies sécurisés pour stocker les données de session côté client, ce qui permet de maintenir la continuité de l'expérience utilisateur. Les données de session peuvent être utilisées pour stocker des informations telles que l'authentification de l'utilisateur, les préférences personnalisées ou d'autres données spécifiques à la session.

Hachage⁹ : Les méthodes de hachage de mots de passe sont des techniques qui convertissent un mot de passe en une forme difficile à décrypter avant de le stocker dans une base de données.

Cela garantit la sécurité même si la base de données est compromise, car les mots de passe originaux ne sont pas stockés directement.

Thread¹⁰ : Un thread est une entité d'exécution légère qui permet à un programme d'effectuer plusieurs tâches simultanément. Les threads permettent d'améliorer l'efficacité en exécutant des parties de code en parallèle, ce qui est particulièrement utile pour les opérations d'entrée/sortie ou les calculs intensifs.

WebSocket¹¹ : Un WebSocket est un protocole de communication bidirectionnel qui permet une communication en temps réel entre un client et un serveur via une connexion TCP persistante. Contrairement aux requêtes HTTP traditionnelles, qui sont de nature unidirectionnelle, les WebSockets permettent une communication continue, facilitant ainsi les mises à jour en temps réel, les notifications et les échanges de données interactifs entre le client et le serveur.

Regex¹² : Une regex, ou expression régulière, est une séquence de caractères qui forme un motif de recherche. Elle est utilisée pour trouver des correspondances dans les chaînes de texte en fonction de ce motif. Les regex sont largement utilisées pour la recherche de motifs spécifiques dans les données textuelles, la validation de formats d'entrée utilisateur et la manipulation de texte dans de nombreux langages de programmation et outils de traitement de texte.

API¹³ : Une API (Interface de Programmation Applicative) est un ensemble de règles, de protocoles et de définitions qui permettent à différentes applications informatiques de communiquer entre elles. Elle définit les méthodes standardisées par lesquelles les logiciels peuvent interagir, échangeant des données et des fonctionnalités de manière efficace et sécurisée. En résumé, une API facilite l'intégration et l'interaction entre différents systèmes logiciels.

Route¹⁴ : Une route Flask est une URL spécifique définie dans une application Flask, qui correspond à une fonction ou à une méthode spécifique. Ces routes sont utilisées pour définir les points d'entrée de l'application web Flask, spécifiant comment l'application doit réagir lorsque certaines URL sont invoquées par un client HTTP (comme un navigateur web). Les routes Flask sont généralement associées à des fonctions ou à des méthodes qui effectuent des opérations spécifiques, telles que la génération de contenu HTML, le traitement de formulaires ou l'accès à des données dans une base de données. En résumé, les routes Flask sont des points de terminaison qui définissent le comportement de l'application Flask en réponse à des requêtes HTTP spécifiques.

VII – Annexes

Annexe I : Lettre de recommandation



Lettre de recommandation

LE SCIELLOUR Damien
Responsable de Production
CDC Cyber Sécurité
TIBCO SERVICES
Le Bois Cholet – BP 9 – 44860 Saint-Aignan-de-Grand-Lieu
Tél portable : 06 36 16 40 07
E-mail : dlesciellour@tibco.fr

Objet : lettre de recommandation

Madame, Monsieur,

En tant que responsable de production du Centre de Compétences Cyber Sécurité chez TIBCO SERVICES, j'ai eu l'honneur d'avoir comme stagiaire Théo BOUTROUX pour un stage d'une durée de 5 semaines.

Enthousiaste, curieux, investi, il a su mener à bien le projet que je lui avais confié. Il est même allé au-delà de ce qui était attendu. Le projet consistait à refondre complètement l'interface d'une application web qui nous permettait de faire un suivi des accès (entrées/sorties) dans notre open space de production. Il a parfaitement su trouver l'équilibre entre l'obtention du résultat final sans se perdre dans les détails et cela en proposant un produit abouti et directement opérationnel. Durant tout le stage il est resté à l'écoute et attentif à nos demandes d'évolution du cahier des charges de départ. Sur un plan plus global, il s'est parfaitement et très rapidement intégré au contexte de l'entreprise. C'est au regard de tous ces atouts que je me permets de recommander la candidature de Théo au sein de votre entreprise.

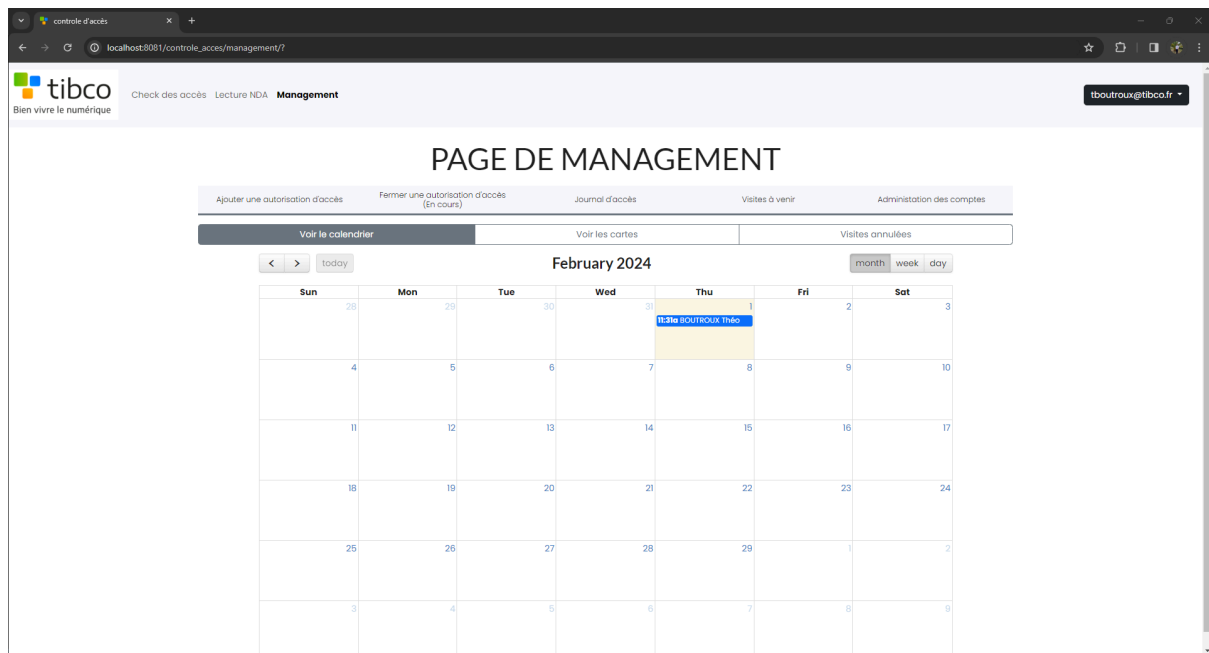
C'est avec un réel plaisir que je reste à votre disposition tous renseignements supplémentaires dont vous jugeriez utiles.

Veuillez agréer, Madame, Monsieur, mes meilleures salutations.

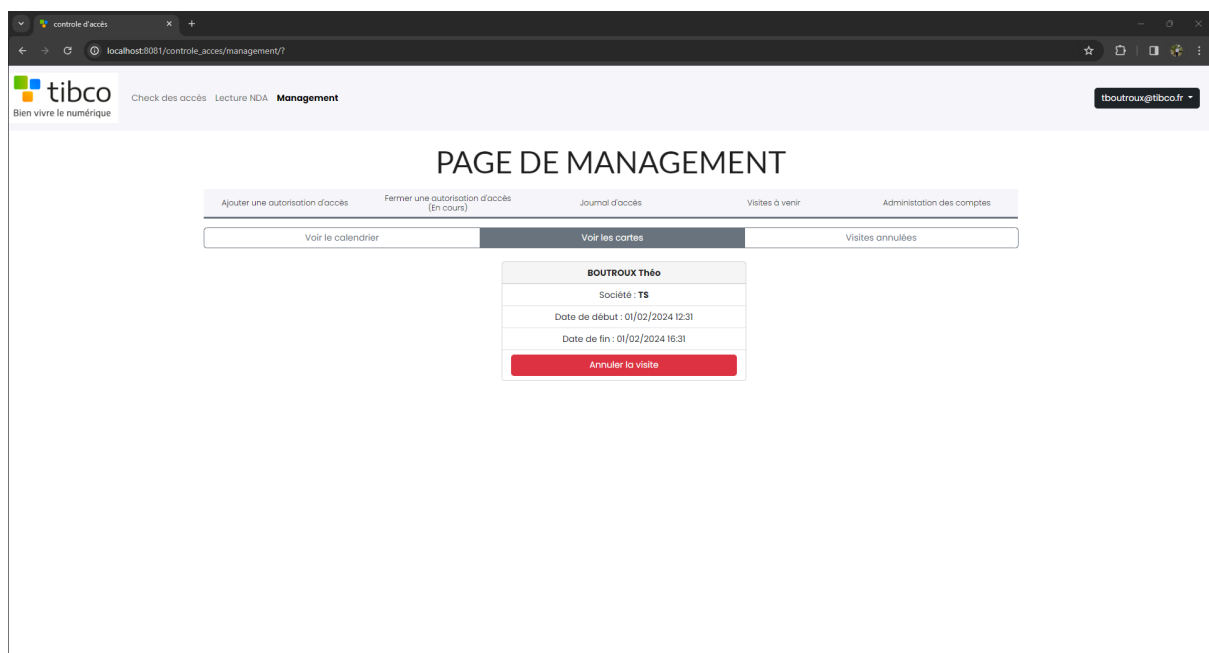
Damien LE SCIELLOUR



Annexe 2 : “Visites à venir”



Onglet correspondant au visites à venir sous la forme d'un calendrier



Les visites à venir sous forme de cartes

contrôle d'accès

localhost:8081/contrôle_accès/management/?

tibco

Bien vivre le numérique

Check des accès

Lecture NDA

Management

tboutroux@tibco.fr

PAGE DE MANAGEMENT

Ajouter une autorisation d'accès

Fermer une autorisation d'accès (En cours)

Journal d'accès

Visites à venir

Administration des comptes

Voir le calendrier

Voir les cartes

Visites annulées

<>today

February 2024

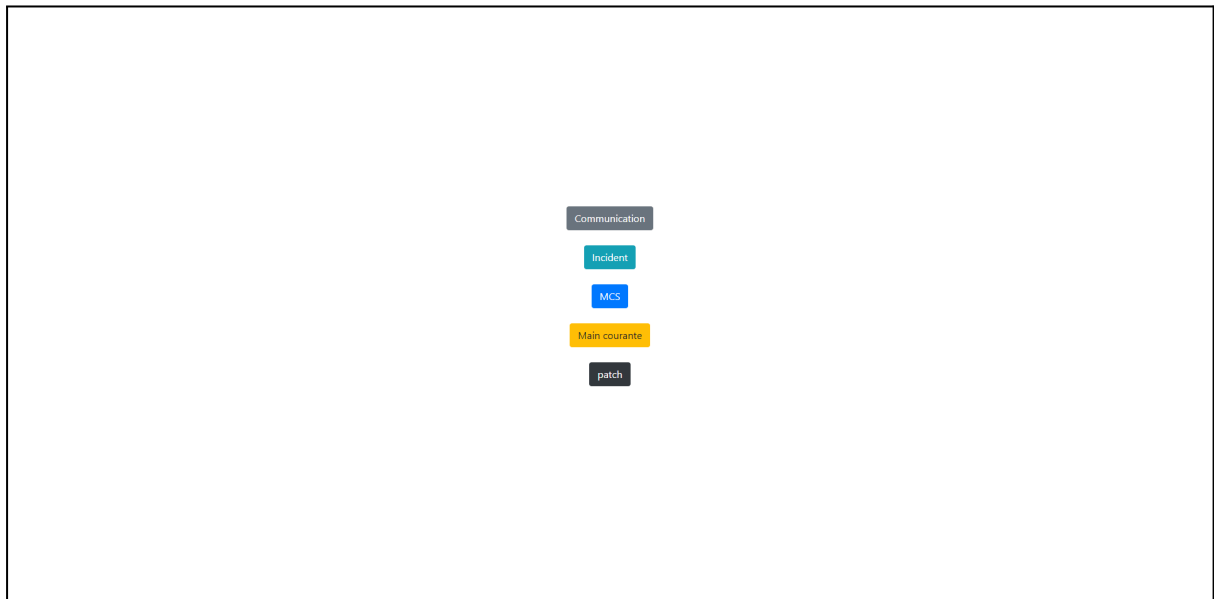
monthweekday

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1 13:30e BOUTROUX Théo	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	1	2
3	4	5	6	7	8	9

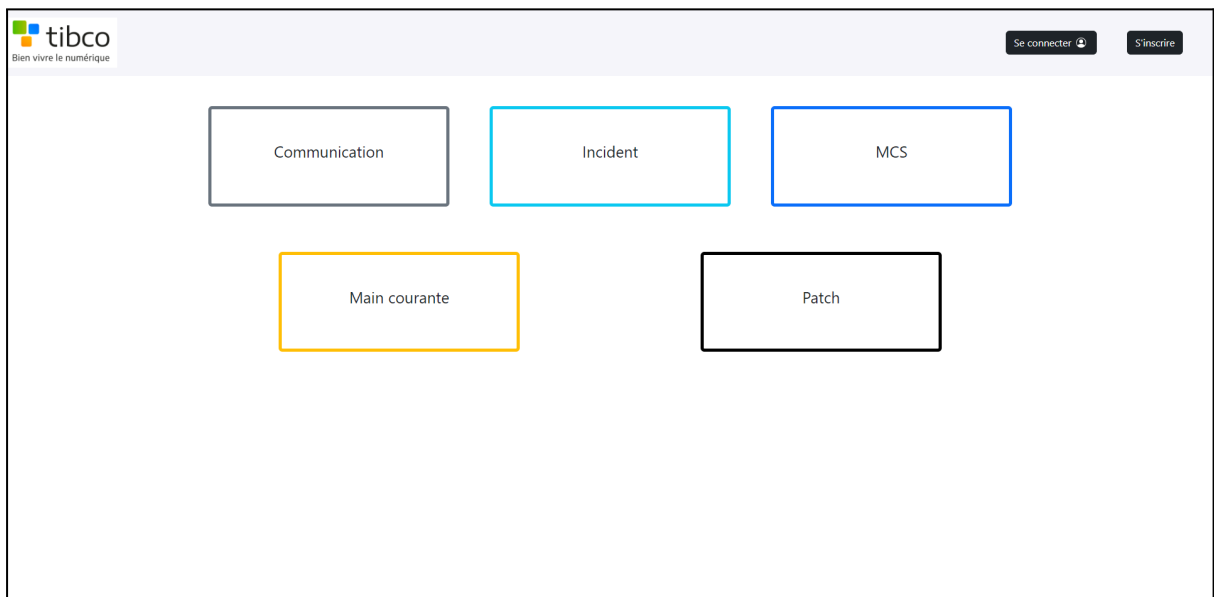
Le calendrier des visites annulées

37

Annexe 3 : La page d'accueil de l'interface siem






Ancienne page d'accueil



Nouvelle page d'accueil

Annexe 4 : La documentation de l'application

1. Structure de la BDD

compte_interface_siem	controle_acces_validation_master	informations_NDA	journaux_entrees_sorties
id 	nom	id_NDA	nom
nom	prenom	date_de_signature	prenom
prenom	societe	prenom	date_entree
quadrigramme_akuiteo	mail	nom	date_sortie
mail	date_saisie		type
date_de_creation	date_debut_accord_RP		id 
date_de_suppression	date_fin_accord_RP		
est_actif	id 		
mot_de_passe	est_annule		
est_archive			
date_de_validation			

Les tables :

- controle_acces_validation_master :
 - > Champs : id, nom, prenom, societe, mail, date_saisie, date_debut_accord_RP, date_fin_accord_RP, est_annule (bool)
- journaux_entrees_sorties :
 - > Champs : nom, prenom, date_entree, date_sortie, type ("entree" ou "sortie")
- comptes_interface_siem :
 - > Champs : id, nom, prenom, quadrigramme_akuiteo, mail, date_de_creation, date_de_suppression, est_actif, mot_de_passe, est_archive, date_de_validation
- informations_NDA :
 - > Champs : id, date_de_signature, nom, prenom

Aucune relation entre les tables

2. Fonctionnalités

Gestion de comptes utilisateurs :

- Création d'un compte
- Connexion à son compte quand il a été activé
- Modification de ses informations
- Déconnexion

Contrôle des accès :

- Rentrer un nom et prénom afin de pouvoir savoir si une personne respecte les conditions requises pour accéder au CDC.
- Pouvoir savoir qui est dans le CDC en temps réel
- Créer une autorisation pour le personnel de ménage et pour le CTI de Nantes
- Calendrier disponible uniquement en lecture pour recenser l'ensemble des visites prévues

Management :

- **Ajouter une autorisation d'accès :**
 - > Ajout d'une autorisation
 - > Possibilité de sauvegarder une autorisation et de visualiser l'état d'un NDA (Signé, En attente) (Affichage des autorisations sur la droite du formulaire)
 - > Modification d'une autorisation (Lors d'un clic sur une autorisation sauvegardée, les champs du formulaire sont automatiquement remplis)
 - /!\ Attention à bien sauvegarder le formulaire
 - > Suppression d'une autorisation sauvegardée
 - > Envoi d'un NDA à partir du formulaire
 - > Fermer une autorisation d'accès :
 - > Suppression d'une autorisation d'accès (en cliquant sur le bouton correspondant à la personne)
 - > Outil de tri sur le tableau

- **Journal d'accès :**

- > Visualisation des événements passés (tableau contenant des informations sur les personnes entrées dans le CDC : Nom, Prenom, Date_Entree, Date_Sortie, Duree)
- > Outil de tri sur le tableau

- **Visites à venir :**

- > Contient 3 boutons permettant de voir différentes infos
- > Voir Calendrier : Calendrier contenant l'ensemble des visites non annulées. Possibilité de cliquer sur les événements pour faire apparaître une pop-up dans laquelle on y retrouve un bouton pour annuler la visite.

/!\ le fait d'annuler une visite ne la supprime pas de la BDD mais change juste le champ 'est_annule'

- > Voir Cartes : Permet d'afficher les mêmes infos que le calendrier mais sous forme de cartes (Possibilité également d'annuler une visite)
- > Visites annulées : Calendrier contenant l'ensemble des visites annulées.

- **Administration des comptes :**

- > Activation et archivage des comptes
- > Recherche de compte
- > Visualisation des informations concernant les comptes

Lecture des NDA :

- Lecture du NDA de n'importe qui lors d'un clic sur le bouton
- Possibilité de rechercher un NDA

3. Les routes

Nombre de routes : **29**

Routes globales

/controle_acces/API_get_all_NDA

Cette route permet de récupérer l'ensemble des données des NDA présent dans le dossier partagé (Nom, prénom, Société) en utilisant l'API sharepoint

/controle_acces/API_recherche_toutes_autorisations

Route permettant de récupérer l'ensemble des autorisations

Route utilisées dans la page d'accueil

/controle_acces/

Route permettant d'accéder à la page principale de la main courante

/controle_acces/API_check_access

Route permettant de faire les check successifs de présence des éléments obligatoire pour accorder l'accès à l'espace cyber

/controle_acces/list_acces_en_cours

Route permettant de récupérer la liste des accès en cours

/controle_acces/API_log_access_enter

Route permettant de logger l'entrée d'une personne dans le fichier journaux_entrees_sorties

Routes utilisées dans la page de management

/controle_acces/management/

Route permettant d'accéder à la page de management de la main courante

`/controle_acces/callback`

Cette route permet de gérer le callback de l'API Docusign

`/controle_acces/API_get_all_visits`

Cette route permet de récupérer l'ensemble des prochaines visites prévues à partir de la BDD

`/controle_acces/API_cancel_visit`

Cette route permet d'annuler une visite prévue dans la BDD

`/controle_acces/API_add_new_access`

Route permettant d'ajouter une nouvelle autorisation dans la base de donnée

`/controle_acces/API_check_if_authorization_exists`

Cette route permet de vérifier si une autorisation existe dans la BDD

`/controle_acces/API_add_new_saved_authorization`

Cette route permet d'ajouter une nouvelle autorisation dans le fichier `all_saved_authorizations.json`

`/controle_acces/API_delete_saved_authorization`

Cette route permet de supprimer une autorisation dans le fichier `all_saved_authorizations.json`

`/controle_acces/API_update_saved_authorization`

Cette route permet de mettre à jour une autorisation dans le fichier `all_saved_authorizations.json`

`/controle_acces/API_get_all_saved_authorizations`

Cette route permet de récupérer l'ensemble des données des autorisations enregistrées dans le fichier `all_saved_authorizations.json`

/controle_acces/API_tout_le_journal_access

Route permettant de récupérer l'ensemble des entrées et sorties

/controle_acces/API_recherche_autorisation_en_cours

Cette route permet de récupérer l'ensemble des autorisations en cours

/controle_acces/API_recherche_toutes_autorisations

Route permettant de récupérer l'ensemble des autorisations

/controle_acces/API_sortie

Cette route permet de mettre à jour la base de donnée pour la sortie d'une personne

/controle_acces/API_revoque_autorisation_master

Cette route permet de révoquer une autorisation

/controle_acces/API_get_invalid_accounts

Cette route permet de récupérer l'ensemble des comptes invalides

/controle_acces/API_validate_account

Cette route permet de valider un compte invalide dans la BDD, c'est-à-dire de changer le champ est_actif à 1

/controle_acces/API_archive_account

Cette route permet d'archiver un compte dans la BDD, c'est-à-dire de changer le champ est_archive à 1

/controle_acces/API_search_account

Cette route permet de rechercher un compte dans la BDD

/controle_acces/API_send_docusign_envelope

Cette route permet d'envoyer un document à signer à une personne

Routes utilisées dans la page de lecture des NDA

/lecture_nda

Route permettant d'accéder à la page Lecture NDA

/controle_acces/API_get_NDA_content

Cette route permet de récupérer le contenu d'un NDA

/controle_acces/API_get_sharepoint_NDA

Cette route permet de récupérer l'ensemble des NDA présent dans le dossier partagé (Nom, prénom, Société) en utilisant l'API sharepoint

/controle_acces/API_search_NDA

Cette route permet de rechercher un NDA dans la BDD

4. Docusign

Les fonctions du module.py :

get_access_token(code, client_secret, client_id, current_app, session)

Fonction permettant de récupérer le token d'accès à partir du code d'autorisation

Paramètres :

- code : Le code d'autorisation obtenu après l'authentification de l'utilisateur
- client_secret : Le secret client de l'application DocuSign
- client_id : L'ID client de l'application DocuSign
- current_app : L'application Flask en cours d'exécution
- session : L'objet session de Flask

Sortie : Le token d'accès

`send_envelop(args: dict, current_app, session, response: dict)` Fonction permettant d'envoyer un document à signer

Paramètres :

- `args` : Un dictionnaire contenant les informations nécessaires pour envoyer le document à signer
- `current_app` : L'application Flask en cours d'exécution
- `session` : L'objet session de Flask
- `response` : Un dictionnaire pour stocker les réponses ou erreurs

Sortie : Aucune

`get_authorization_url(current_app, session)` Fonction permettant de générer l'URL d'autorisation pour DocuSign

Paramètres :

- `current_app` : L'application Flask en cours d'exécution
- `session` : L'objet session de Flask

Sortie : L'URL d'autorisation pour DocuSign

› Comment faire marcher l'envoi de NDA ?

Il suffit de décommenter le code ayant pour commentaire "DOCUSIGN"

› Où sont les morceaux de code ?

Début du fichier `all_app.py` (Variable `code_verifier`)

Route `/controle_acces/management`

Route `/controle_acces/API_get_all_saved_authorizations`

Route `/controle_acces/API_send_docusign_envelop`

Gestion d'envoi des NDA dans `global.js` (vers la ligne 100)