COMPLEMENT

Ayant fini la réalisation du pipeline plutôt que prévu, j’ai pu faire d’autre tâches dans l’entreprise. La partie qui suit est un aperçu global et chorologique des tâches supplémentaires qui m’ont été conféré.

1. BASCAC – Blu Age (J’ai oublié la suite).

BASCAC est une petite application web qui permet de simplifier la gestion de la facturation de Velocity au client. Celle-ci s’appuis sur la base de données client (identique à la base de données qu’utilise le pipeline pour récupérer les ACL).

L’application permet entre autres, d’ajouter/retirer la permission à un client d’utilisé Velocity, ou encore de géré le pourcentage de réduction accorder au client. En conséquence, son accès doit être limité car elle gère des données de facturation.

L’objectif de cette tâche est de mettre en ligne BASCAC de manière sécurisée, en limitant son accès par un mot de passe. J’ai utilisé l’IDE Visual Studio Code pour y ajouter mes modifications.

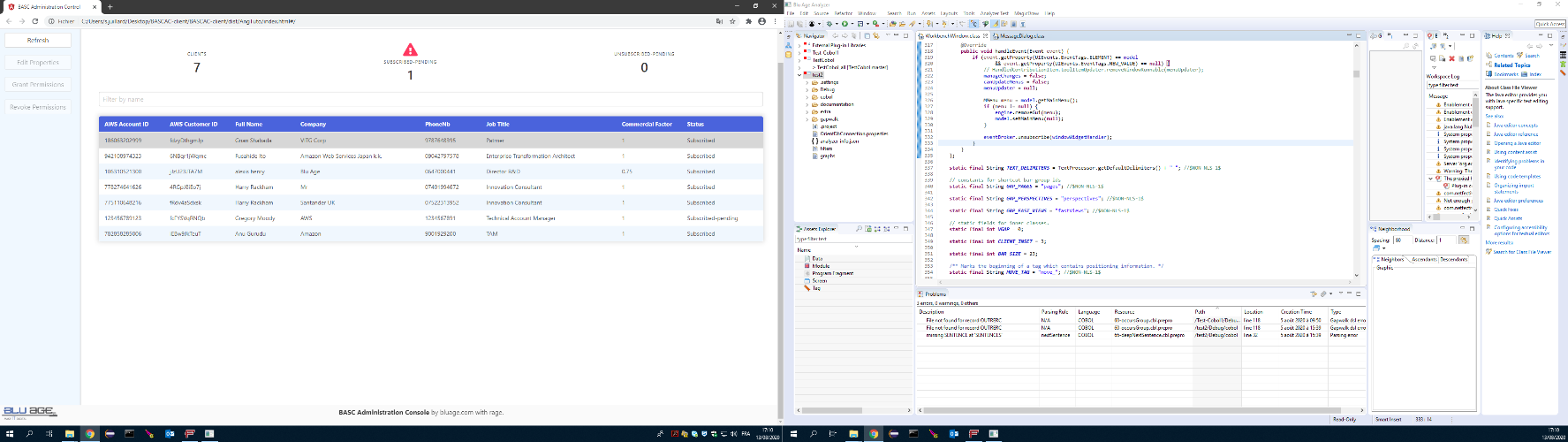
L’outils qui m’as été préconisé est AWS Cogito, un service d’amazone, pour les applications web, qui propose de gérer les mots de passe.

L’application à été réalisé avec Angular, soit codé en TypeScript, html et css. L’application été nativement une “single-page application” ou SPA. J’ai dû rajouter à cette application un mécanisme de routage afin d’y ajouter une page d’authentification.

Une fois cela terminer, j’ai essayé d’intégrer cogito à ma page d’authentification. C’est à cette étape que j’ai pris connaissance de Amplify. Un autre service d’amazone qui permet de mettre en ligne des applications. Pour la mise en place de cognito, son utilisation été nécessaire.

En me renseignant plus sur cet outil, je me suis aperçu que je pouvais déployer BASCAC, après l’avoir compilé, de manière non-programmatique sur la console graphique d’Amazone. Puis une fois cela fait, ajouter la gestion des mots de passe, qui apparait comme un pop-up en haut de la page. (Cf. image)

Au final, je suis revenu sur mon travail d’ajout de page d’authentification et d’utilisation de cognito et j’ai opté pour un déploiement et une gestion des mots de passe non-programmatique, à l’aide d’amazone.

N.B : Sur demande de son créateur, j’ai également modifié légèrement le style de la page en ajoutant du contenu au feuilles de style de l’application.

2 – Exportation de la Vue Problème de Blu Age Analyzer.

Analyzer est un produit de Blu Age de la V7 <ALORS LA J’AI BESOIN D’EXPLIQUATION POUR SON USAGE>.

Lorsqu’un client rencontre des problèmes avec son code, il peut contacter Blu Age afin de recevoir de l’aide. On lui demander alors le contenu de l’onglet « problème », l’onglet dans lesquels est affiché les erreurs et les warnings issue de l’a compilations du/des projets en cours.

L’objectif de cette tâche est d’ajouter, dans l’onglet problème une option pour exporter le contenu de la vue au format .csv (tableurs).

Blu age Analyser est une IDE développé sur la base d’Eclipse. Elle possède à la fois des onglet, vues et fonctionnalités développées par Blu Age et d’autre issue de Eclipse.

En xml, on peut ajouter des éléments graphiques à Analyzer, et les liés avec des handlers, commandé en java.

La difficulté de cette tâche réside dans l’adaptation de code interne à Eclipse. L’onglet « problèmes » d’Analyzer, est une vue dont le code est interne et il est donc difficile d’en récupérer les éléments. Pour cette raison, il n’est pas possible de récupérer le contenu de la vue problème à proprement parlé. Pour récupérer ce contenu, il a fallu aller à la source de ce contenu : On va récupérer les informations « à la source », soit dans les markers d’un projet. Un marker est un ensemble de donnée généré dans notre cas à la racine du projet dans lesquels les informations relatives à celui-ci son stocker.

C’est en récupérant ces informations que j’ai pu reconstituer la vue problème, puis l’exporter sous le format csv. Après son implémentation, j’ai pris connaissance d’une option similaire, qui avait déjà été implémenter, mais qui avais un usage différent. Cette option proposait d’exporter, à la racine du projet, dans plusieurs formats possibles, les problèmes relatifs à un projet en faisant une clique droite sur celui-ci dans la vue navigation.

J’ai donc dans un deuxième temps, adapter ma solution pour quelle s’intègre au travail déjà effectuer. Ainsi, sans retirer l’option déjà implémenté, j’ai ajouté à la vue problème une option qui propose d’exporter la vue problème (problèmes de tous les projets ouvert), dans plusieurs formats différents, dont le csv, et de l’enregistre par un menu contextuel en choisissant le nom et l’emplacement du rapport d’erreurs.

Une fois la tâche terminer, j’ai pris connaissance des protocoles de Blu Age afin de manager le suivit des commit. Avec l’aide de Oliver <Nom de famille> j’ai revu l’écriture de mon code, j’ai rempli mon message de commit avec un identifiant qui correspondais à ma tâche, puis on a rempli ensemble cette fiche pour présente l’option implémenté.

Conclusion :

Ce stage ma permis de prendre en compétence, et ce dans diverse domaine : devOps, devWeb et dev logiciel. J’ai codé dans plusieurs langages différents, utilisé et tester divers outils. Cette expérience m’as permis de développer ma capacité d’adaptation et ma flexibilité.

J’ai dû moi-même concevoir des solutions, chercher par moi-même les outils qui pourrais me servir et faire le choix des plus adéquats. Pour la tâche d’implémentation d’une fonctionnalité d’export, j’ai pu me familiariser avec les problématiques de la programmation d’un grand projet, ou plusieurs acteurs différent commit du code. Ce que l’on ne rencontre par forcément dans un projet en milieu scolaire. J’ai aussi du crée de la documentation pour le pipeline que j’ai créée, pour qu’elle puisse être reprise plus tard. J’ai eu la satisfaction de terminer les tâches que l’on m’a conféré et j’ai l’impression d’avoir énormément appris au cours de ces deux mois, non seulement d’un point de vue technique, mais aussi sur le fonctionnement en entreprise.