Création d'une application de création / gestion de profils de certificats SSL/TLS*

*Possibilité de commercialisation engageant le secret professionnel

1) Définir les besoins du projet

- Genèse du projet : Ce projet est né d'une problématique de perte de temps. En effet la société RFIT pour laquelle je travaille développe des applications de traçabilité via routeur Bluetooth. Ces routeurs sont connectés à des serveurs MQTT. Pour des raisons de sécurité la connexion entre le serveur et les routeurs doit être chiffrer. Pour cela on utilise des certificats SSL/TLS auto-signés (car sur un intranet chez les clients). Pour créer ces certificats on utilise des outils en ligne de commande avec des scripts fastidieux à éditer pour chaque nouveau client/renouvellement. Ainsi une application desktop permettant la gestion d'une base de client et l'automatisation de la création de certificat ferai gagner beaucoup de temps aux techniciens de RFIT.

- Ce projet est dans le cadre de l'alternance de Robin Coblentz au sein de l'entreprise RFIT.

- Ce projet est nouveau et s'inscrit dans le cadre d'une amélioration continue des services disponibles aux employés de RFIT.

- Le projet doit permettre aux techniciens de créer les certificats en créant un profil client ou en utilisant un profil existant afin de mettre à jour les certificats. Les certificats créés doivent être trié par client/date de création.

L'expression des besoins pour ce projet « CertSSL » est constituée de :

- Mettre en place une documentation tutoriel de l'application.
- Proposer une application, à la fois simple et épuré, utilisable rapidement et facilement.
- L'application doit être légère et facilement installable.

Les enjeux du projet par rapport à l'Entreprise sont :

- Avoir une application Windows permettant de gagner du temps quant à la création des certificats SSL.
- Permettre d'enregistrer des profils de clients.
- Mettre en commun les profils entre les applications utilisées par les techniciens.

Les enjeux du projet par rapport à Robin Coblentz sont :

- De développer ces compétences en C#
- Découvrir l'environnement MongoDB sous C#
- Découvrir la création de Setup Windows.

La mise en place de l'application se fera via un setup et une base de données gratuite, les couts seront donc nul.

Cela devra permettre aux techniciens de gagner en temps et en productivité.

- L'objectif du projet :

  L'objectif et d'éviter la perte de temps en automatisant la tache de l'édition de script, l'accès aux ressources de profils clients, et de simplifier la gestion du partage des profils clients entre les techniciens.

1) Composer l'équipe de projet

- Identifier les participants :

  Membres du projet : Robin Coblentz est en alternance en tant que technicien chez RFIT dans le cadre du BTS SIO. En tant que technicien il a diverse mission tel que le développement d'applications et de projets pour des clients mais aussi la mise en place de solution interne à RFIT dans le but de gagner en productivité.

- Distribuer les rôles :

Robin Coblentz aura le profil de développeur. Ses compétences actuelles dans le secteur du numérique et de l'informatique sont :

- **C#** : niveau débutant
- **Base de données Mongodb** : niveau intermédiaire
- **.NET** : niveau débutant

2) Cahier des charges du projet « CertSSL »

- L'analyse fonctionnelle :

  - L'app Application devra être simple à utiliser.
  - 2 parties distinctes doivent être perceptibles : une pour les profils, une pour les certificats.
  - Profils : On pourra ajouter ou supprimer les profils, un clic dessus permet de le sélectionner.
  - Certificats : On peut choisir un chemin de destination, la durée de validitée, et le chemin des library OpenSSL.

- Les contraintes

  - Contraintes de couts : Le cout de développement informatique doit être nul, pour cela nous utiliserons des outils de développement gratuit (Visual Studio 2019 Community.
  - Contraintes d'utilisation : Le poste sur lequel tourne l'application doit être connecté à internet pour fonctionner. OpenSSL doit être installer sur le poste car ses Library permettent la signature des certificats.
  - Contraintes de délais : Le projet doit être finalisé pour fin avril.

- Planification du projet

  - La 1ere phase : évaluation des besoins et 1ere rédactions du document de suivi (01/03/2021→ 15/03/2021)

  - La 2eme phase : développement frontend C# (15/03/2021 → 17/03/2021)

  - La 3eme phase : Formation MongoDB C# (17/03/2021 → 20/03/2021)

  -La 4eme phase : Préparation MongoDB et Visual Studio 2019 (20/03/2020 → 25 /03/2021)

  -La 5eme phase : Développement (25/03/2020 → 10 /04/2021)

  -La 6eme phase : Test (10/04/2020 → 12/04/2021)

  -La 7eme phase : Mise en place Setup (12/04/2020 → 14/04/2021)

- Chiffrage du Projet

Tous les coûts liés aux ressources techniques, logicielles et matériels ne sont pas à chiffrer étant donné qu'ils sont inexistants pour ce projet.

Chaque profil en base pèse quelque ko, or l'hébergeur atlas MongoDB est gratuit jusqu'à 500mo de stockage. On peut donc déduire que nous n'aurons pas besoins de payer d'abonnement MongoDB.

3) Etude de marché

Sécurité :

Dans notre cas, la mise en place de certificats est en pleines essors. En effet les entreprises cherchent à sécuriser au maximum leurs réseaux depuis la vague de piratage via WannaCry. Les entreprises ont pris conscience de l'importance de la sécurité des moyens de communication et de transfert d'informations. De nombreuses technologies sans fils sont vulnérables à ce type d'attaque et beaucoup de technologie sont utilisées en milieu industriel de manière croissante.

https://www.rs-online.com/designspark/la-communication-sans-fil-en-environnement-industriel-fr-3

https://fr.wikipedia.org/wiki/WannaCry

Sans-fils :

On observe également que les technologies sans fils sont de manière générale de plus en plus utilisés en milieu industriel. Elles ont un côté pratique et peu couteux indéniable par rapport aux câbles. Seulement la sécurité en prend un coup : ici l'absence d'accès physique n'est plus un problème pour mes pirates qui peuvent même non connecter essayer de décrypter et interpréter les signaux grâce à différents outils. Pire, une fois connecté grâce à des failles de sécurité ils peuvent visualiser les données transitant de manière transparente sans être détecté (attaque type Men In The Middle).

https://fr.wikipedia.org/wiki/Attaque_de_l%27homme_du_milieu

Chiffrage :

Ainsi les certificats SSL/TLS offre un réel avantage d'un point de vue sécurité en chiffrant les données de bout en bout grâce à un système de clé privée/public rendant inefficace les interceptions sans le certificat permettant de les déchiffrer. De plus le server permet de confirmer son identité au client et de valider la sécurité sur un réseau.

https://www.sslmarket.fr/ssl/certificats

4) Définition de la solution

Application Windows :

L'application Windows est un programme exécutable, développé en C#. Il comporte une partie graphique avec laquelle l'utilisateur peut interagir. Cette application se divise en 2 parties :

Une partie profil permet de créer des profils pour les différents clients de RFIT et de lister les existants. On peut ajouter un profil en remplissant les champs « nom de société », « fonction de l'entreprise », « pays », « ville », « nom de domaine/IP » qui sont les champs nécessaires à la signature d'un certificat. En cliquant sur un profil dans la liste on peut également choisir de le supprimer. Une fois un profil choisi il se charge dans la section « certificat ».

Une partie certificat permet de sélectionner un emplacement ou sauvegarder les certificats générés. On peut choisir le temps de validité de celui-ci, entrée une passpharse qui servira de variables pour encoder le certificat, et on pourra sélectionner le chemin d'accès à OpenSSL dont les Library permettent la création des certificats. Un bouton permet ensuite de générer les certificats server/client/ca et leurs clés respectives.

Base de données :

Nous choisirons pour des raisons de performance le service mongodb car ce type de base nosql est particulièrement léger, simple à mettre en place et est tout à fait adapté à un usage léger car il a été conçu pour la gestion de fichier json au format. Il offre ainsi des performances honorables et peut être setup extrêmement rapidement.

Setup :

Afin de pouvoir installer rapidement l'application, j'ai créé un setup d'installation permettant de télécharger et installer celle-ci de manière simple et intuitive en 3 clics. Le setup est développé à l'aide du plugin Microsoft Setup Installer conçu pour fonctionner avec les projets C# de Visual Studio 2019.

5) Cout et contraintes de la solution :

Les couts de développement seront nuls car les services, api et d'outils de développement utilisés dans ce projet sont gratuits.

L'application doit être légère et utilisable en le moins de clics possible afin de la rendre la plus efficace possible.

Une connexion à internet est requise pour que l'application puisse communiquer avec la base donnée.

6) Test :

Une phase de test eu lieu afin de vérifier le bon fonctionnement de l'application.

Tout d'abord l'ajout de plusieurs milliers de profils a été essayer afin de voir sur l'application rester réactive en les chargeant. Elle le reste car il lui faut environ 10 secondes pour charger 5000 comptes client.

Ensuite des tests en coupant puis remettant le réseau ont été fait, l'application n'actualise que s'il y a du réseau.

Pour finir un grand nombre de certificats ont été créer sans la moindre difficulté.

7) Versionning :

Le versionning est assuré par visual studio 2019 qui permet de monter automatiquement les numéros de build.

A noter que le plugin github aide également à visualiser le code et les améliorations mises en place.

Chaque mise à jour majeur incrémentera le numéro de version de 1

Chaque mise à jour mineur incrémentera le numéro de version de 0.1

Chaque mise à jour corrective incrémentera le numéro de version de 0.0.1