Rendu 1er juillet

THOUVENIN Sébastien | SERRON Vincent
KAHRAMAN Mumin | REYNAUD Ludovic
CASSAIGNE Antoine | LECONTE Tony

Référent : AHMADI Laifa

Table des matières

I. Introduction et contexte

Très difficilement détectable, dangereuse pour les données des personnes ou de l'organisme concernée, les attaques Man-In-The-Middle dans lesquels des données sont interceptées lors d'un transfert restent courantes. En effet, récemment une startup israélienne s'est faite volée près d'un million d'euros de cette manière. Pour contrer ces attaques, une des meilleures solutions est le cryptage de bout en bout car de cette manière même si quelqu'un intercepte les données elles restent illisibles à première vue. Silver Cipher s'inscrit dans ce cadre car notre projet est de réaliser un système de transfert de dossier crypté de bout en bout. Afin de réaliser ceci, nous pourrons au cours du projet nous inspirer d'outils comme Firefox Send, Wormhole ou encore Swiss Transfer utilisent déjà cette solution pour partager des messages ou des fichiers afin de les sécuriser durant l'entièreté de l'envoi.

II. Outils

1. OpenSSL via Crypto

OpenSSL est un outil qui peut être utilisé pour le chiffrement de fichiers. C'est un outil très complet que nous pouvons utiliser sur tous les types de chiffrement de notre projet. Nous l'avons utilisé pour la génération et vérification des clés et des certificats. Pour utiliser OpenSSL nous avons utilisé le module *Crypto* de node.js.

2. ElectronJS

ElectronJS est un framework qui peut être utilisé pour créer des applications en utilisant notamment NodeJS. Nous l'avons utilisé pour notre application car tous les éléments nécessaires à la création de l'application étaient réalisables avec ElectronJS et que nous étions tous plus familier avec le langage JavaScript qu'avec d'autres langages comme le C#. La version utilisée est, par soucis de compatibilité avec certaines bibliothèques comme axios, electron-store ou crypto, une version antérieure à la dernière version.

3. RSA

Le chiffrement RSA est un chiffrement asymétrique qui consiste en deux clés, une clé publique servant à chiffrer les données et une clé privée servant au déchiffrement. Nous nous en sommes servi pour chiffrer la clé de déchiffrement symétrique qui sert au déchiffrement du fichier chiffré transféré. La clé publique de chaque utilisateur sera stockée dans notre base de données et leur clé privée elle restera localement dans l'appareil de l'utilisateur.

4. AES

AES est une méthode de chiffrement symétrique. Cela signifie que la clé de chiffrement et de déchiffrement est la même. Nous utilisons une clé de chiffrement de 256 bits ce qui nous procure 1,1 X 10^{77} combinaisons possibles. Il n'est donc quasi impossible de les déchiffrer avec les méthodes et les puissances de calculs dont on dispose aujourd'hui. Dans notre solution, une nouvelle clé est générée à chaque transfert et est utilisée pour le chiffrement du fichier.

5. Certificat

Un certificat est utilisé pour authentifier un individu. Il est validé par une Autorité.

6. AWS

Le service AWS (Amazon Web Service) est une filiale d'Amazon qui fournit des serveurs web. Il nous permet de contenir notre base de données ce qui nous permet d'y accéder à n'importe quel instant.

7. MySQL

MySQL est un système de gestion de bases de données relationnelles. Nous l'avons utilisé pour créer notre base de données qui stockera les informations publiques des utilisateurs et les transferts de fichiers chiffrés.

8. React

React est une bibliothèque JavaScript qui nous permet de gérer l'interface de notre application.

9. Github

GitHub est un des services de partage de données les plus utilisés dans le domaine de l'informatique. Il nous sert à mettre en commun tout le code de l'application de manière efficace.

10. Figma

Figma est un outil de création de prototypes et de design. Il nous a servi à établir la ligne directrice pour l'interface graphique de notre application.

11. Notion

Notion est un outil qui permet de gérer l'organisation de l'équipe. Il nous sert notamment à recenser les documents, à gérer le travail de l'équipe et à suivre l'avancement du projet.

12. Express

Express est un framework qui peut être utilisé pour la création de serveurs en JavaScript. Nous l'avons utilisé pour créer notre API qui fait le lien entre le serveur AWS et l'application.

13. Axios

Axios est une bibliothèque JavaScript qui peut être utilisée pour effectuer des requêtes http. Nous l'avons utilisée afin de faire nos requêtes à la base de données MySQL.

III. Méthode de travail

Tout d'abord, au sein de l'équipe le travail a été défini lors de réunions. Nous avons tenu un rythme de deux réunions par semaine. Avant ces réunions nous définissions le ou les thèmes à aborder lors de la réunion pour que tout le monde puisse y réfléchir. Durant ces rencontres tout le monde a pu proposer et défendre ses idées et ensuite chacun donnait son avis afin peser le positif, le négatif et finalement prendre une décision. Ces décisions étaient prises sur l'accord de l'entièreté de l'équipe et les quelques fois un quelqu'un n'était pas d'accord nous avons toujours su trouver un terrain d'entente afin d'obtenir son consentement.

Notre équipe était également basée sur la communication. Nous étions en échange permanant et personne n'a été mis à l'écart car tout a toujours été partagé à l'entièreté du groupe. Pour pouvoir communiquer, nous avons principalement utilisé *WhatsApp* pour les messages et *discord* pour faire des réunions avec des personne en distanciel, notamment lorsque certains membres de l'équipe ont eu le covid. Nous avons utilisé d'autre outils comme *notion* pour partager des documents et définir des « to do list » et *GitHub* pour partager notre code afin que toute l'équipe ait accès à tout le projet à tout instant.

Enfin, en ce qui concerne le travail en lui-même, nous avons toujours laissé la liberté aux membres de travailler comme bon leur semble. Certains travaillaient ensemble en présentiel quand d'autre préféraient rentrer chez eux pour effectuer leur tâche. Nous avons essayé au maximum d'appliquer la méthode Agile et nous nous sommes efforcés de respecter ses 12 fondements.

IV. Réponse à l'appel d'offre

1. Rappel de l'offre

L'information est au cœur des enjeux de notre société et la sécurisation du partage des données est essentielle, notamment avec La loi RGPD qui établit le cadre du traitement des données personnelles. On peut retrouver de nombreux exemples de données volé qui ont eu de lourd impactes comme lors des élections présidentielles américaine de 2016. En effet durant cette élection la partie démocrate semblait en tête mais suite à la fuite d'email volé le cours des élections a changé et c'est finalement le parti républicain qui s'est imposé. C'est dans ce contexte qu'il nous a été demandé de proposer une solution de partage de fichier offrant le chiffrement de bout en bout.

2. Le marché et les cibles

Le marché du transfert de données chiffré est vaste. De nombreuses grosses entreprises se sont lancé dans ce domaine et ont proposé des solutions au grand public. Ce qui nous différencie c'est notre offre sur mesure car certes nous proposons notre application au grand public mais nous proposons également une personnalisation sur mesure et sur demande pour des entreprises par exemple. De plus nous sommes une entreprise à l'écoute de ses utilisateurs. En effet un service utilisateur sera mis à disposition pour être à l'écoute de ces derniers afin de toujours pouvoir améliorer notre application.

Notre application a pour cible tous les niveaux de la société qu'il soit public ou privé. En effet nous offrons nos services aussi bien au particulier qui souhaiterai garder leur échange sécurisé qu'au entreprise pour qui le secret et la sécurité des données est essentiel.

3. Notre entreprise

Notre équipe *Silver Cipher* est composée de 6 étudiants très ambitieux qui partagent tous une même passion pour l'informatique et la cybersécurité. Nous partageons également un certain nombre de valeurs notamment social avec le respect de chacun de nos membres et de nos collaborateurs.

4. Solution technique

La solution proposée par notre équipe est une application de bureau.

Au premier lancement de l'application, l'utilisateur est invité à créer un compte. L'application génère ensuite une clé privée qui va servir au déchiffrement des données, elle sera stockée localement et l'utilisateur en aura la responsabilité. Une clé publique sera également générée et celle-ci permettra le chiffrement des données lors de l'envoi.

Une fois connecté, à la manière d'une messagerie électronique classique, l'utilisateur peut consulter les transferts qu'il a effectué dans sa boîte d'envoi et ceux qu'il a reçu dans sa boîte de réception où il peut télécharger les fichiers. Il peut bien entendu créer un nouveau transfert qui contient un fichier et potentiellement un message.

Lors d'un envoi, l'application chiffre d'abord le fichier et le message à l'aide d'une clé symétrique générée pour le transfert. La clé symétrique est ensuite chiffrée par la clé publique du destinataire, puis le fichier chiffré ainsi que la clé symétrique chiffrée sont envoyés sur notre base de données. Le destinataire peut ensuite télécharger le fichier chiffré transféré, l'application va déchiffrer la clé symétrique avec la clé privée locale généré lors de l'inscription. Enfin le fichier va lui-même être déchiffré par cette clé symétrique qui vient d'être déchiffrée.

Dans notre base de données les seuls éléments susceptibles d'être sensible sont complétement chiffré et donc illisible en cas de fuite. Le reste ne sont que des éléments publics comme le répertoire.

5. Proposition commerciale

Devis			
Intitulé	Prix unité	Quantité	TOTAL
Serveur web	1 150,99 €	1	1 150,99 €
Licence d'utilisation	4 922,00 €	1	4 922,00 €
Main d'œuvre	2 267.00 €	6	13 602,00 €
		TOTAL	19 674.99 €

V. Annexes

1. UCs et UCD

Nom du Use Case : **UC-1 S'inscrire**
Objectif du Use Case : Un Client non inscrit s'inscrit
Contexte : Le Client veut s'inscrire
Portée : Sous-système du Client
Niveau : Actor Goal
Acteur principal : Le Client
Participants : Aucun
Déclencheur : Le Client lance l'application pour la première fois
Condition initiale : Le Client veut être inscrit
Garantie minimale : Aucune
Scénario principal :

1. Le Client lance le logiciel
2. Le Client renseigne ses informations
3. Une paire de clés RSA est générée sur l'ordinateur du Client
4. Les informations sont traitées et les données à transférer sont envoyées au serveur
5. Le serveur envoie la confirmation au Client que le compte a bien été créé

Extensions :

Nom du Use Case : **UC-2 Supprimer le compte**
Objectif du Use Case : Un Client inscrit veut supprimer son compte
Contexte : Le Client veut supprimer son compte
Portée : Sous-système du Client
Niveau : Actor Goal
Acteur principal : Le Client
Participants : Aucun
Déclencheur : Le Client va sur l'onglet de suppression du compte
Condition initiale : Le Client veut supprimer son compte
Garantie minimale : Aucune
Scénario principal :

1. Le Client va sur la page de paramètre
2. Le client clique sur le bouton de suppression de compte
3. Le Client prend connaissance des risques
4. Le Client valide la suppression du compte en cliquant sur le bouton de validation

Extensions :

Nom du Use Case : **UC-3 Envoyer**
Objectif du Use Case : Un Client inscrit veut envoyer un fichier
Contexte : Le Client veut envoyer un fichier
Portée : Sous-système du Client
Niveau : Actor Goal
Acteur principal : Le Client
Participants : Aucun
Déclencheur : Le Client va sur l'onglet d'envoi de fichiers
Condition initiale : Le Client veut envoyer un fichier
Garantie minimale : Aucune
Scénario principal :

1. Le Client se rend sur la page de création de nouveau message
2. Le Client charge le fichier et renseigne les champs d'envoi
3. Le Client lance la procédure de cryptage et d'envoi
4. Le fichier crypté est envoyé vers le serveur

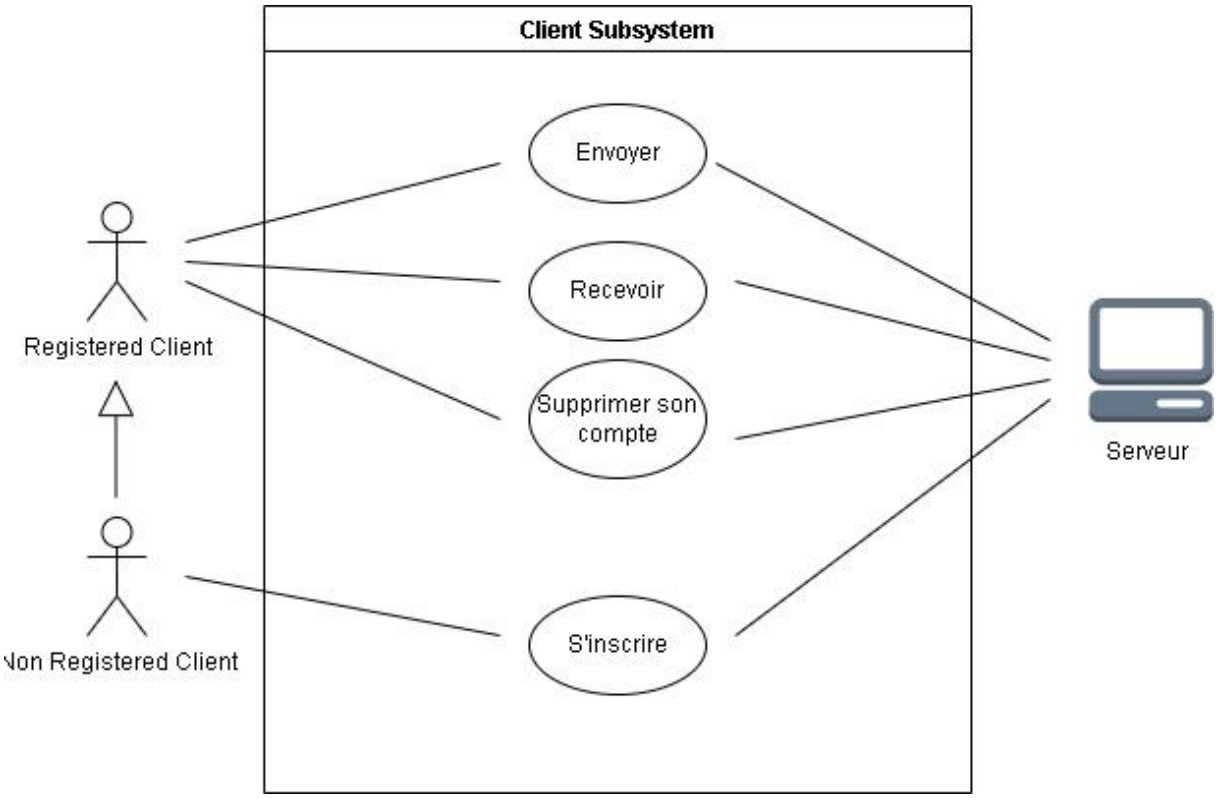
Extensions :

Nom du Use Case : **UC-4 Recevoir**
Objectif du Use Case : Un Client inscrit a reçu des fichiers
Contexte : Le Client a reçu des fichiers
Portée : Sous-système du Client
Niveau : Actor Goal
Acteur principal : Le Client
Participants : Aucun
Déclencheur : Le Client va sur la boîte de réception
Condition initiale : Le Client a reçu un ou des messages
Garantie minimale : Aucune
Scénario principal :

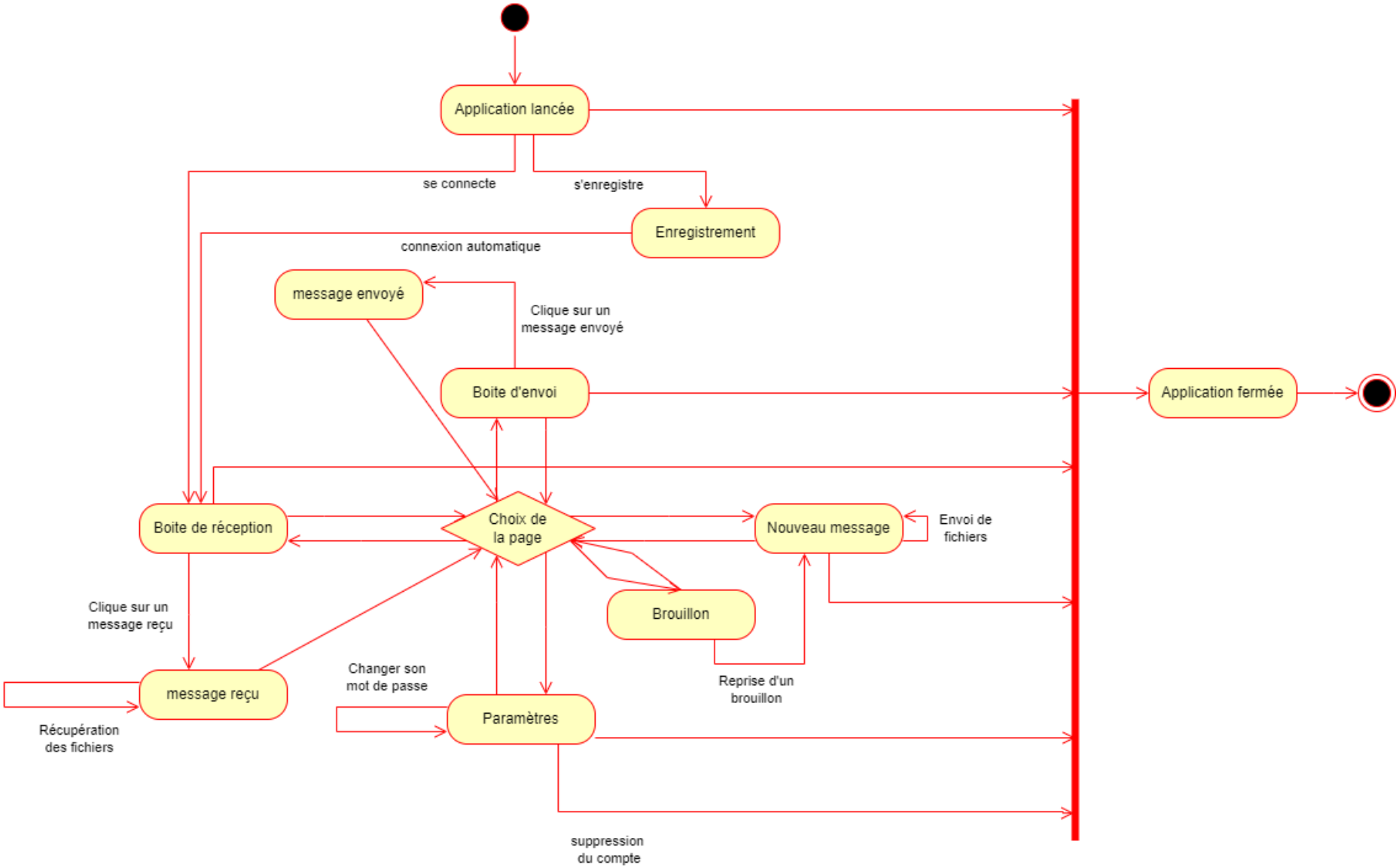
1. Le Client se rend sur la boîte de réception
2. Le Client reçoit les fichiers cryptés sur son ordinateur
3. Le Client clique sur l'icône de téléchargement du fichier
4. Le fichier est décrypté et téléchargé sur l'ordinateur du Client

Extensions :

5. Le Client supprime le fichier de la base de données


Client Subsystem
Envoyer
Recevoir
Supprimer son compte
S'inscrire
Registered Client
Non Registered Client
Serveur

2. Activity Diagram du Client



3. Backlog produit de Silver Cipher

Vision : « Un logiciel permettant l'envoie de fichiers chiffrés de bout en bout ».

ID	Titre	Description	Estimation durée (jours)
0	Système de notification	L'utilisateur devrait pouvoir recevoir une notification lorsqu'il reçoit un nouveau message.	3
1	Système de recherche dans les boites	L'utilisateur doit pouvoir rechercher dans sa boite d'envoie et sa boite de réception des échanges précis a l'aide de mot clés.	2
2	Création page web de révocation	Une page web permettant la suppression du compte doit être créer pour pouvoir supprimer son compte de n'importe où en cas de problème et si aucun ordinateur n'est disponible.	2
3	Système de certification	L'utilisateur doit pouvoir être authentifier avec un certificat qui prouve son identité.	5
4	Brouillon	Un onglet qui permet de retrouver tout transferts qui n'ont pas été envoyer.	2
5	Favoris	Une liste de favoris pour trouver facilement les utilisateurs à qui envoyer des fichiers.	2

4. Macro-planning

Détails du projet	
Projet	Silver Cipher
Jours ouvrés uniquement	NON
Date de début	02/06/2022
Commentaires :	

	Tâches	Date de début	Durée	Date de fin	Réalisé
1	**Phase 1**				
1.1	Documentation	09/06/2022	5	13/06/2022	100%
1.2	Word 1er itération	13/06/2022	3	15/06/2022	100%
1.3	Figma	13/06/2022	3	15/06/2022	100%
1.4	class diagramme	13/06/2022	3	15/06/2022	100%
1.5	activity diagramme	13/06/2022	3	15/06/2022	100%
1.6	use case diagramme	13/06/2022	3	15/06/2022	100%
2	**Phase 2**				
2.1	Interface graphique	17/06/2022	7	23/06/2022	100%
2.2	Création de la base de données	17/06/2022	1	17/06/2022	100%
2.3	Création des onglets	18/06/2022	3	20/06/2022	100%
2.4	Création de l'onglet paramètre	24/06/2022	2	25/06/2022	100%
2.5	Système de création de compte	17/06/2022	12	28/06/2022	100%
2.6	système de génération de clé	17/06/2022	12	28/06/2022	100%
2.7	Système d'authentification	20/06/2022	1	20/06/2022	100%
2.8	Système de certification	20/06/2022	5	24/06/2022	0%
2.9	Système de création de clé de suppression	21/06/2022	2	22/06/2022	0%
2.10	Système de chiffrement/d'échange de clé et déchiffrement	21/06/2022	5	25/06/2022	100%
2.11	Création page web de révocation	24/06/2022	2	25/06/2022	0%
2.12	Préparation de la présentation avec entrainement	02/07/2022	4	05/07/2022	0%
2.13	Éventuels imprévus	28/06/2022	3	30/06/2022	0%
2.14	Création d'une API	17/06/2022	4	20/06/2022	100%
2.15	Système de modification de mot de passe	01/07/2022	1	02/07/2022	0%
2.16	Système d'envoie/réception	26/06/2022	5	30/06/2022	80%