

**Auteur :** Zhaojie LU, Chengyu YANG

**08/10/2021**

**Type :** STBE

**Equipe :** Virgilia Muselli,  
Michiel Chinchole,  
Zhaojie Lu, Chengyu Yang

**Chemin :** PISTL/STBE

**Statut :**Après correction

**Destinataire(s) :** Virgilia Muselli, Michiel Chinchole

**Page(s) :** 20

## SafetyLine-Chiffrage

## Méta Modèle de la STBE

Version	Date	Modifications	Rédacteur
1.0	08/10/2021	Initial	Zhaojie LU,Chengyu YANG
1.1	15/10/2021	Définitions	Zhaojie LU,Chengyu YANG
1.2	22/10/2021	Solution	Zhaojie LU,Chengyu YANG
1.3	28/10/2021	En attente de correction	Zhaojie LU,Chengyu YANG
2.0	06/11/2021	Après correction	Zhaojie LU,Chengyu YANG

## Table des matières

1. Définition du projet	3
1.1. Les objectifs	3
1.2. Le contexte	3
2. Définition du système à réaliser	4
2.1. Fonctionnalités du système	4
2.2. Cas d'utilisation	6
2.2.1. Diagramme de Use Case	6
2.2.2. Détail du cas d'utilisation	7
2.2.3. Diagramme de séquence	11
2.3. Validation	13
2.3.1. Détail du test de validation	13
3. Contraintes	17

3.1. Contraintes techniques	17
3.2. Analyse des risques	17
4. Planification	18
4.1. WBS	18
4.2. PERT	19
4.3. GANTT	19
5. Annexes	20
5.1. Bibliographie	20

## 1. Définition du projet

### 1.1. Les objectifs

La cérémonie des “chiffrages” est une partie importante de la gestion de projet d'une entreprise. C'est le moment où les équipes se réunissent pour discuter des tâches à traiter lors du prochain sprint. Une fois les tâches acceptées, assignées par l'équipe de développement, chaque membre va planifier le temps d'achèvement prévu pour sa tâche et envoie l'ensemble de ses estimations par mail au chef de projet. Le chef de projet examine et approuve ces plans. Il va ensuite se connecter au GitLab, et ajouter les temps d'achèvement pour chaque tâche.

Objectif principal:

Mettre en place une application qui automatise cette cérémonie de chiffrage.

### 1.2. Le contexte

Safety Line est un leader de l'analyse des données pour le marché de l'aviation. Safety Line a développé des solutions logicielles innovantes qui permettent aux compagnies aériennes et aux aéroports de faire face aux enjeux majeurs du transport aérien en matière de sécurité, d'économies carburant et d'environnement.

Notre projet sera dirigé par deux encadrants de Safety Line. Nous utilisons l'Email et Microsoft Teams pour communiquer et analyser le projet. Dans leur environnement de travail, ils utiliseront principalement gitlab comme plate-forme de projet. Et notre projet va simplifier la cérémonie de chiffrage au sein de leur gestion de projet.

## 2. Définition du système à réaliser

### 2.1. Fonctionnalités du système

Notre système est une web application qui peut automatiser tous les processus de chiffrage. Nous allons utiliser Angular qui est une plateforme pour la création d'une web application au niveau de l'entreprise. Et aussi, nous allons connecter avec gitlab (plateforme principale du projet safety-line) pour réaliser la dernière étape de la cérémonie de chiffrage.

Nous avons 3 points principaux dans notre web application : User, Admin et Tâche.

Les 2 premiers points sont des utilisateurs qui vont réaliser le fonctionnement de chiffrage.

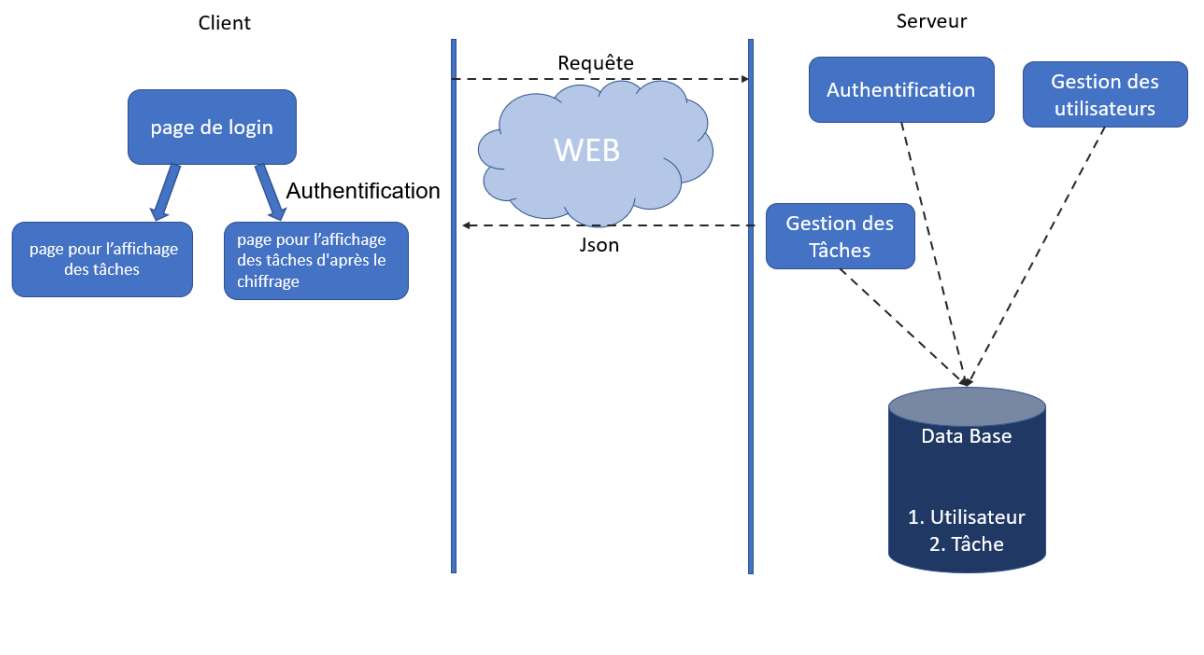
- User est capable de voir les tâches(par Gitlab) et d'ajouter le temps d'achèvement prévu à leur tâche.
- Admin est un rôle spécifié d'un utilisateur, il peut créer et supprimer des users, en plus il peut voir tous les plannings créés par user et valider ses plans.

Le tâche est le principe de notre projet, tous les processus sont autour de lui. On doit obtenir ces tâches par l'api de Gitlab en initial, puis d'après la modification par user et la validation par admin, on met à jour ces tâches dans gitlab.

Pour maintenir les données d'utilisateur et les tâches, nous avons aussi besoin de créer une base de données.

Composant du système :

- Page Web en Angular
  1. page de login
  2. page pour l'affichage des tâches
  3. page pour l'affichage des tâches d'après le chiffrage
- Base de données
  1. utilisateurs
  2. tâches
- Serveur pour connecter au base de données



Fonctionnalité attendu:

- Système de login
  1. authentification User/Admin
- Interaction avec gitlab
  1. prendre des tâches sur gitlab
  2. mise à jour ces tâches avec le temps estimé
- Chiffrage
  1. ajouter le temps d'achèvement estimé
  2. modification des tâches

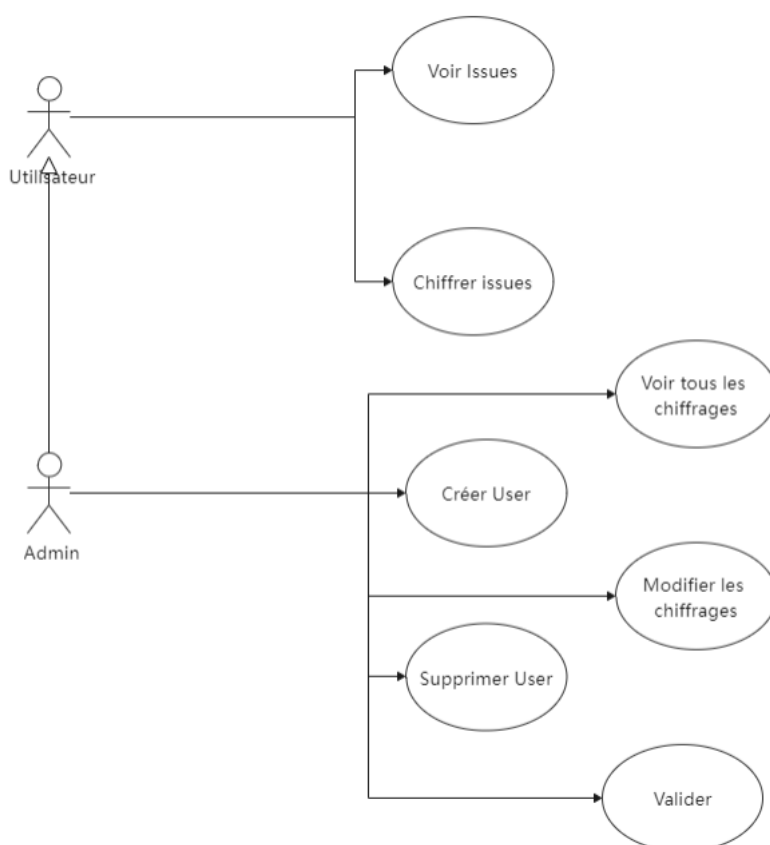
## 2.2. Cas d'utilisation

### User/Admin

<b>Nom</b>	user
<b>Type</b>	humain
<b>Rôle</b>	User peut voir et chiffrer les issues/tâches.

<b>Nom</b>	admin
<b>Type</b>	humain
<b>Rôle</b>	Admin peut supprimer et créer les users et voir ses chiffrages. Il peut aussi éditer et valider les chiffrages.

### Diagramme de Use Case



## Détail du cas d'utilisation

### Fiche détaillées

<b>Nom du cas d'utilisation</b>	Voir les issues
<b>Acteur principal</b>	User : u1
<b>Acteurs secondaires</b>	Aucun
<b>Objectif</b>	L'utilisateur u1 veut voir toutes les issues.
<b>Pré-condition(s)</b>	Les issues sont déjà obtenues par git. L'utilisateur u1 a déjà authentifié.
<b>Hypothèses</b>	Aucun
<b>Scénario</b>	1.L'utilisateur u1 se trouve dans la page d'utilisateur. 2.L'utilisateur u1 clique sur le bouton <Voir les issues>. 3.Le système affiche les issues. 4.L'utilisateur u1 peut voir toutes les issues sur l'écran.
<b>Post-condition(s)</b>	Les issues sont bien affichées sur l'écran.
<b>Exigences fonctionnelles</b> <b>non</b>	Aucun
<b>Exceptions, alternatives</b>	Aucun

<b>Nom du cas d'utilisation</b>	Chiffrer les issues
<b>Acteur principal</b>	User : u1
<b>Acteurs secondaires</b>	Aucun
<b>Objectif</b>	L'utilisateur u1 veut chiffrer une issue.
<b>Pré-condition(s)</b>	Les issues sont déjà affichées sur l'écran. L'utilisateur u1 a déjà authentifié.
<b>Hypothèses</b>	Aucun
<b>Scénario</b>	1.L'utilisateur u1 se trouve dans la page d'utilisateur. 2.L'utilisateur u1 clique une issue :issue 1. 3.Le système affiche les détails d'issue 1 et lui demande les autres informations. 4.L'utilisateur u1 saisie le temps estimé. 5.L'utilisateur u1 clique sur le bouton <Chiffrer>. 6.Le système envoie cette issue chiffrée dans la database et affiche une notification sur l'écran.
<b>Post-condition(s)</b>	L'issue chiffrée est bien stockée dans la database.
<b>Exigences fonctionnelles</b> <b>non</b>	Aucun
<b>Exceptions, alternatives</b>	Aucun

<b>Nom du cas d'utilisation</b>	Créer un utilisateur
<b>Acteur principal</b>	Admin : ad1
<b>Acteurs secondaires</b>	La liste d'utilisateurs
<b>Objectif</b>	L'utilisateur ad1 veut créer un utilisateur.
<b>Pré-condition(s)</b>	L'utilisateur ad1 a déjà authentifié avec rôle administrateur.
<b>Hypothèses</b>	Aucun
<b>Scénario</b>	1.L'utilisateur ad1 se trouve dans la page d'administrateur. 2.L'utilisateur ad1 clique sur le bouton <Créer un utilisateur>. 3.Le système lui demande de saisir le nom et le mot de passe. 4.L'utilisateur ad1 saisie les informations demandés. 5.Le système envoie cet utilisateur créé dans la database et affiche une notification sur l'écran.
<b>Post-condition(s)</b>	L'utilisateur créé est stocké dans la database. L'utilisateur créé peut être affiché dans la liste d'utilisateurs.
<b>Exigences fonctionnelles</b> <b>non</b>	Aucun
<b>Exceptions, alternatives</b>	Aucun

<b>Nom du cas d'utilisation</b>	Supprimer un utilisateur
<b>Acteur principal</b>	Admin : ad1
<b>Acteurs secondaires</b>	la liste d'utilisateurs
<b>Objectif</b>	L'utilisateur ad1 veut supprimer un utilisateur.
<b>Pré-condition(s)</b>	L'utilisateur ad1 a déjà authentifié avec rôle administrateur.
<b>Hypothèses</b>	Aucun
<b>Scénario</b>	1.L'utilisateur ad1 se trouve dans la page d'administrateur. 2.L'utilisateur ad1 clique sur le bouton <Supprimer un utilisateur>. 3.Le système affiche une liste d'utilisateurs et lui demande de choisir un utilisateur à supprimer. 4.L'utilisateur choisit un utilisateur et clique sur le bouton <Supprimer> . 5.Le système affiche une notification sur l'écran et supprime cet utilisateur supprimé dans la database.
<b>Post-condition(s)</b>	L'utilisateur supprimé n'est plus dans la database. L'utilisateur supprimé n'est plus dans la liste d'utilisateurs.
<b>Exigences fonctionnelles</b> <b>non</b>	Aucun
<b>Exceptions, alternatives</b>	Aucun

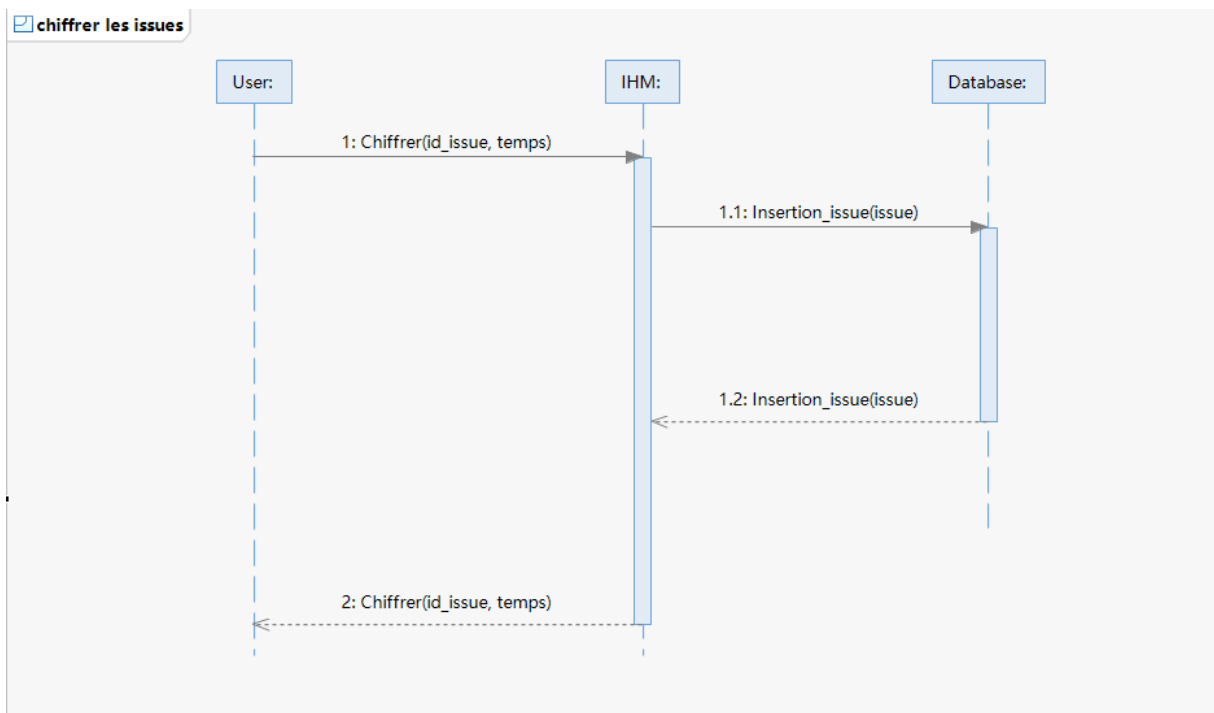
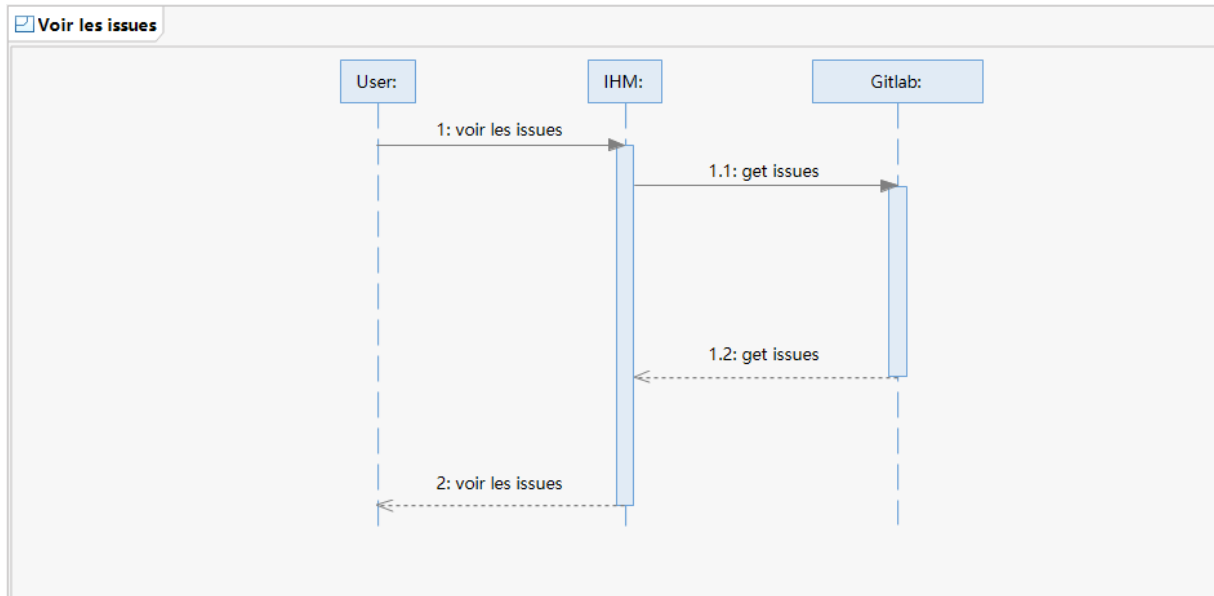


<b>Nom du cas d'utilisation</b>	Voir tous les chiffreages
<b>Acteur principal</b>	Admin : ad1
<b>Acteurs secondaires</b>	Aucun
<b>Objectif</b>	L'utilisateur ad1 veut voir tous les chiffreages.
<b>Pré-condition(s)</b>	L'utilisateur ad1 a déjà authentifié avec rôle administrateur.
<b>Hypothèses</b>	Aucun
<b>Scénario</b>	1.L'utilisateur ad1 se trouve dans la page d'administrateur. 2.L'utilisateur ad1 clique sur le bouton <Voir chiffreage>. 3.Le système affiche tous les chiffreages sur l'écran.
<b>Post-condition(s)</b>	Aucun
<b>Exigences fonctionnelles</b> non	Aucun
<b>Exceptions, alternatives</b>	Aucun

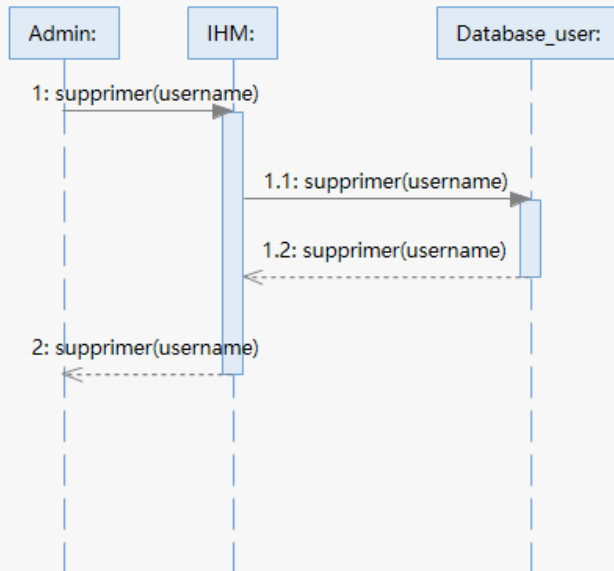
<b>Nom du cas d'utilisation</b>	Modifier un chiffreage
<b>Acteur principal</b>	Admin : ad1
<b>Acteurs secondaires</b>	Aucun
<b>Objectif</b>	L'utilisateur ad1 veut modifier un chiffreage.
<b>Pré-condition(s)</b>	L'utilisateur ad1 a déjà authentifié avec rôle administrateur. L'utilisateur ad1 a déjà cliqué sur le bouton <Voir chiffreage>.
<b>Hypothèses</b>	Aucun
<b>Scénario</b>	1.L'utilisateur ad1 choisit un chiffreage : chiffreage 1 . 2.L'utilisateur ad1 clique sur le bouton <Modifier>. 3.Le système lui demande de saisir les informations. 4.L'utilisateur ad1 saisie les informations demandés. 5.Le système envoie ce chiffreage modifié dans la database et affiche une notification sur l'écran.
<b>Post-condition(s)</b>	Le chiffreage modifié est mis à jour dans la database.
<b>Exigences fonctionnelles</b> non	Aucun
<b>Exceptions, alternatives</b>	Aucun

<b>Nom du cas d'utilisation</b>	Valider un chiffrage
<b>Acteur principal</b>	Admin : ad1
<b>Acteurs secondaires</b>	Aucun
<b>Objectif</b>	L'utilisateur ad1 veut valider un chiffrage.
<b>Pré-condition(s)</b>	L'utilisateur ad1 a déjà authentifié avec rôle administrateur. L'utilisateur ad1 a déjà cliqué sur le bouton <Voir chiffrage>.
<b>Hypothèses</b>	Aucun
<b>Scénario</b>	1.L'utilisateur ad1 choisit un chiffrage : chiffage 1 . 2.L'utilisateur ad1 clique sur le bouton <Valider>. 3.Le système envoie ce chiffage validé dans la database et affiche une notification sur l'écran.
<b>Post-condition(s)</b>	Le chiffage validé est mis à jour dans la database.
<b>Exigences fonctionnelles</b> <b>non</b>	Aucun
<b>Exceptions, alternatives</b>	Aucun

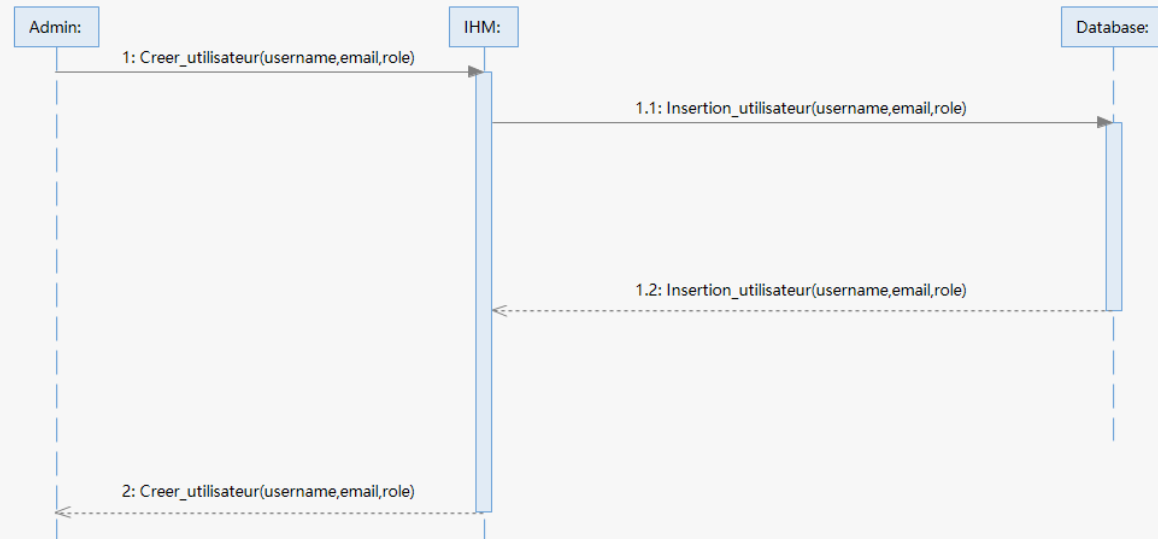
## Diagramme de séquence

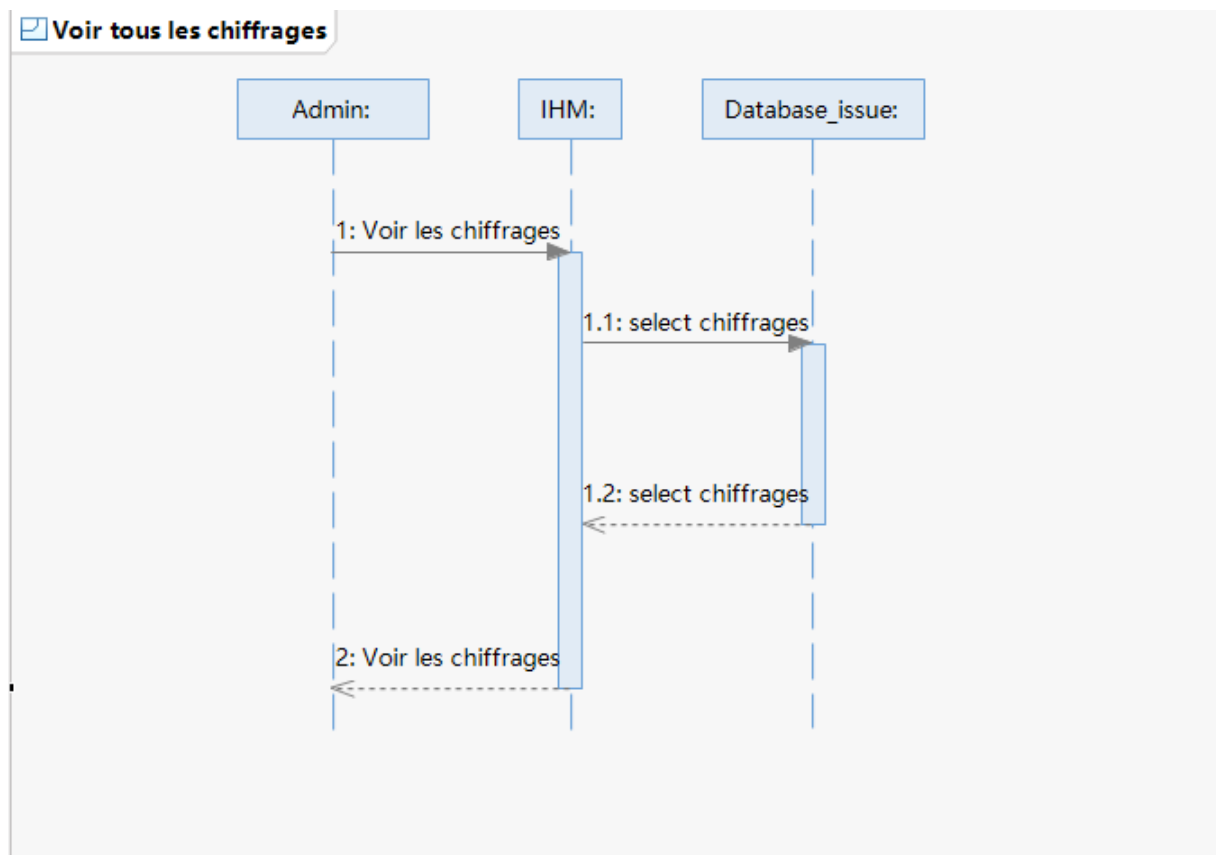


### Supprimer un utilisateur



### Créer un utilisateur





## 2.3 Validation

### Détail du test de validation :

#### Voir les issues

<b>ID</b>	<b>A001</b>
<b>Use Case correspondant</b>	Voir les issues
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle administrateur et il veut voir toutes les issues.
<b>Entrée(s)</b>	Rien
<b>Scénario</b>	1.Le testeur t1 clique sur le bouton <Voir les issues>.
<b>Résultat attendu</b>	Les issues sont affichées sur l'écran.
<b>Post-condition(s)</b>	Rien
<b>Moyen de vérification</b>	Le testeur t1 vérifie que les issues affichées sont bien les issues obtenues par git.

<b>ID</b>	<b>A002</b>
<b>Use Case correspondant</b>	Voir les issues
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle utilisateur et il veut voir toutes les issues.
<b>Entrée(s)</b>	Rien
<b>Scénario</b>	1.Le testeur t1 clique sur le bouton <Voir les issues>.
<b>Résultat attendu</b>	Les issues sont affichées sur l'écran.
<b>Post-condition(s)</b>	Rien
<b>Moyen de vérification</b>	Le testeur t1 vérifie que les issues affichées sont bien les issues obtenues par git.

### Chiffrer les issues

<b>ID</b>	<b>A101</b>
<b>Use Case correspondant</b>	Chiffrer les issues
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle administrateur et il veut voir toutes les issues.
<b>Entrée(s)</b>	Issue : issue 1
<b>Scénario</b>	1.Le testeur t1 choisit l'issue. 2.Le testeur t1 saisit le temps estimé. 3.Le testeur t1 clique sur le bouton <Chiffrer>.
<b>Résultat attendu</b>	Le système affiche un message <L'issue choisie est chiffrée>.
<b>Post-condition(s)</b>	L'issue 1 est stockée dans la database.
<b>Moyen de vérification</b>	Le testeur t1 vérifie que l'issue chiffrée est bien l'issue stockée dans la database.

<b>ID</b>	<b>A102</b>
<b>Use Case correspondant</b>	Chiffrer les issues
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle utilisateur et il veut voir toutes les issues.
<b>Entrée(s)</b>	Issue : issue 1
<b>Scénario</b>	1.Le testeur t1 choisit l'issue. 2.Le testeur t1 saisit le temps estimé. 3.Le testeur t1 clique sur le bouton <Chiffrer>.
<b>Résultat attendu</b>	Le système affiche un message <L'issue choisie est chiffrée>.
<b>Post-condition(s)</b>	L'issue 1 est stockée dans la database.
<b>Moyen de vérification</b>	Le testeur t1 vérifie que l'issue chiffrée est bien l'issue stockée dans la database.

### Créer un utilisateur

<b>ID</b>	<b>ID: A201</b>
<b>Use Case correspondant</b>	Créer un utilisateur
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle administrateur et il veut créer un utilisateur.
<b>Entrée(s)</b>	String: username, String password
<b>Scénario</b>	1.Le testeur t1 clique sur le bouton <Créer un utilisateur>. 2.Le testeur t1 saisit le nom et le mot de passe. 3.Le testeur valide la création.
<b>Résultat attendu</b>	L'utilisateur créé est affiché sur l'écran.
<b>Post-condition(s)</b>	Le compte créé peut être authentifié avec rôle utilisateur.
<b>Moyen de vérification</b>	Le testeur t1 vérifie que l'utilisateur créé est bien l'utilisateur stocké dans la database..

### Supprimer un utilisateur

<b>ID</b>	<b>A301</b>
<b>Use Case correspondant</b>	Supprimer un utilisateur
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle administrateur et il veut supprimer un utilisateur.
<b>Entrée(s)</b>	User : user 1
<b>Scénario</b>	1.Le testeur t1 clique sur le bouton <Supprimer un utilisateur>. 2.Le testeur t1 choisit l'utilisateur <user 1> à supprimer. 3.Le testeur t1 vérifie cette opération.
<b>Résultat attendu</b>	Les messages sont affichés sur l'écran.
<b>Post-condition(s)</b>	Il n'y a plus <user 1> dans la liste des utilisateurs.
<b>Moyen de vérification</b>	Le testeur t1 vérifie que l'utilisateur supprimé n'est plus dans la database..

### Voir tous les chiffreages

<b>ID</b>	<b>A401</b>
<b>Use Case correspondant</b>	Voir tous les chiffreages
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle administrateur et il veut voir tous les chiffreages .
<b>Entrée(s)</b>	Rien
<b>Scénario</b>	1.Le testeur t1 clique sur le bouton <Voir chiffreage>.
<b>Résultat attendu</b>	Les chiffreages sont affichés sur l'écran.
<b>Post-condition(s)</b>	Rien
<b>Moyen de vérification</b>	Le testeur t1 vérifie que les chiffreages affichés sont bien les chiffreages stockés dans la database.

### Modifier un chiffage

<b>ID</b>	<b>A501</b>
<b>Use Case correspondant</b>	Modifier un chiffage
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle administrateur et il a déjà cliqué sur le bouton <Voir chiffage>.
<b>Entrée(s)</b>	Chiffage : chiffage 1
<b>Scénario</b>	1.Le testeur t1 choisit <chiffage 1>. 2.Le testeur t1 clique sur le bouton <Modifier>. 3.Le testeur t1 saisit les informations à modifier. 4.Le testeur valide sa modification.
<b>Résultat attendu</b>	Le message est affiché sur l'écran.
<b>Post-condition(s)</b>	Le chiffage modifié est dans la liste des chiffreages.
<b>Moyen de vérification</b>	Le testeur t1 vérifie que le chiffage modifié est bien dans le database.

### Valider un chiffage

<b>ID</b>	<b>A501</b>
<b>Use Case correspondant</b>	Modifier un chiffage
<b>Contexte</b>	Le testeur t1 a déjà authentifié avec rôle administrateur et il a déjà cliqué sur le bouton <Voir chiffage>.
<b>Entrée(s)</b>	Chiffage : chiffage 1
<b>Scénario</b>	1.Le testeur t1 choisit <chiffage 1>. 2.Le testeur t1 clique sur le bouton <Valider>.
<b>Résultat attendu</b>	Le message est affiché sur l'écran.
<b>Post-condition(s)</b>	Le chiffage validé est dans la liste des chiffreages validés.
<b>Moyen de vérification</b>	Le testeur t1 vérifie que le chiffage validé est bien dans le database.



### 3. Contraintes

#### 3.1. Contraintes Techniques

Notre contrainte sur le projet est principalement par rapport à l'aspect technique. Le client demande la réalisation du projet en Angular (partie front end). En outre, pour des raisons de sécurité, nous sommes obligés de travailler sous nos propres projets.

#### 3.2. Analyse des risques

Risque technique (Projet)	
Description	Besoin de notre propre environnement
Causes	Peut être différent de l'environnement réel
Conséquences	amélioration supplémentaire
Prévention	Aucun
Gravité	Moyen

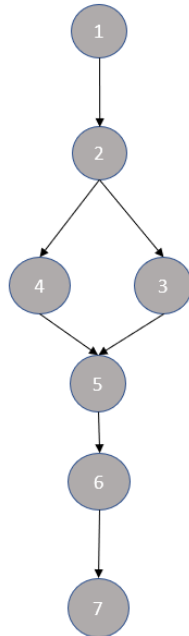
Risque technique (Angular)	
Description	Angular nécessaire
Causes	Peut entraîner des retards
Conséquences	Besoin le temps d'apprendre le technique
Prévention	Aucun
Gravité	Moyen

## 4. Planification

### 4.1. WBS

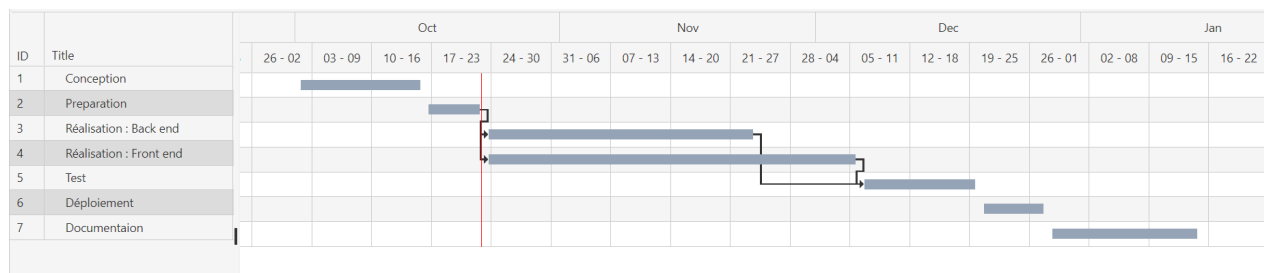


## 4.2. PERT



Numéro de tâche	Description de tâche	Durée minimum	Durée maximum
1	Analyse de projet	21 jours	21 jours
2	Préparation	7 jours	8 jours
3	Réalisation Front End	27 jours	34 jours
4	Réalisation Back End	34 jours	44 jours
5	Intégration Test	14 jours	14 jours
6	Déploiement	3 jours	7 jours
7	Rédaction Documentaire	20 jours	20 jours

## 4.3. GANTT



## 5. Annexes

## 5.1. Bibliographie

GITLAB Document : <https://docs.gitlab.com/ee/api/>

Angular Document : <https://ng.ant.design/docs/introduce/en>

Spring Boot : <https://spring.io/projects/spring-boot#learn>

Mybatis plus: <https://baomidou.com/en/>