



SPRINT 1

artXcape

By Codezilla & the boys



Sommaire

01

1. Introduction

02

2. Sprint Backlog

03

**3. Diagramme de
séquence objets**

04

**4. Diagramme de
classe de conception**

05

**5. Tableau blanc
(Trello)**

Notre Equipe



Karkoub Souhaila



Belmahdi Med Liwa



Mansour Aziz



Sayari Med Amin



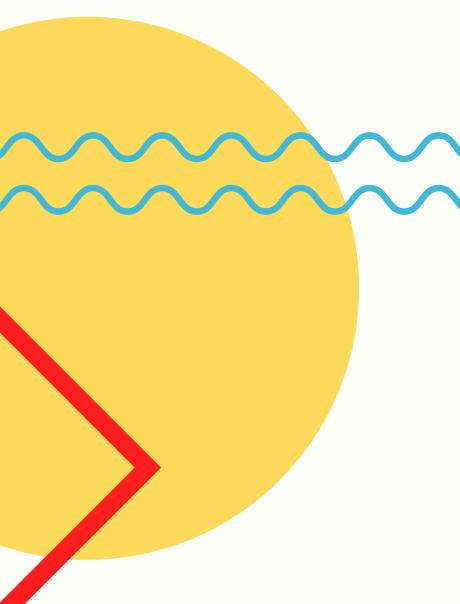
Weslati Jesser



Rouissi Nidhal

1. Introduction

2. Sprint Backlog



definition



Le Sprint Backlog est un élément clé de la méthodologie Agile, en particulier dans le cadre de Scrum. Il s'agit d'une liste des tâches et des éléments de travail que l'équipe s'engage à réaliser durant un sprint, qui est une période de développement fixe, généralement de deux à quatre semaines.



GESTION USER

1. Module CRUD pour les Utilisateur

2. Module API avatar et Mailing pour le mot de passe oublié



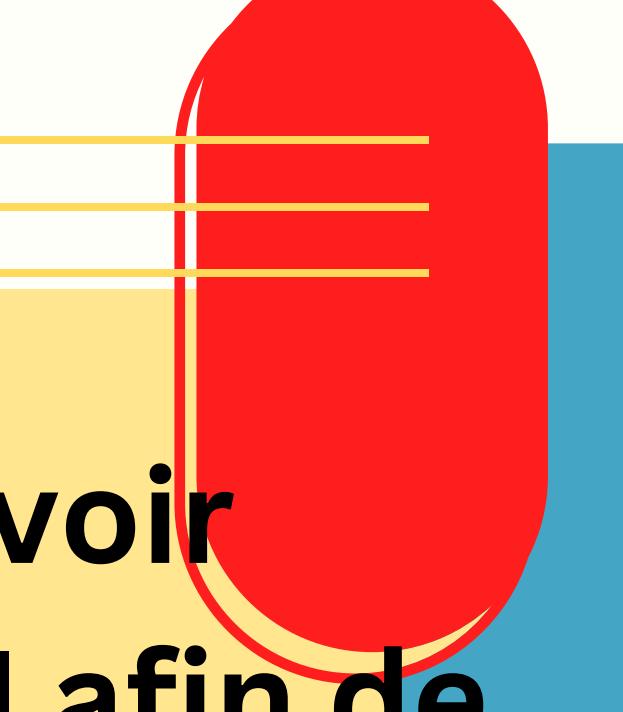
En tant qu'administrateur:

- User Story 1 : En tant qu'administrateur, je veux pouvoir créer un utilisateur afin de gérer les accès et permissions.
- User Story 2 : En tant qu'administrateur, je veux pouvoir lire les détails des utilisateurs afin de vérifier leurs informations.

- User Story 3 : En tant qu'administrateur, je veux pouvoir mettre à jour les informations d'un utilisateur en cas de modification nécessaire.
- User Story 4 : En tant qu'administrateur, je veux pouvoir supprimer un utilisateur afin de gérer efficacement la liste des utilisateurs.

- **Tâche 1** : Créer une interface utilisateur pour la création d'un utilisateur. (3 heures)
- **Tâche 2** : Développer la logique pour enregistrer un nouvel utilisateur dans la base de données. (2 heures)
- **Tâche 3** : Créer une page pour afficher la liste des utilisateurs. (2 heures)
- **Tâche 4** : Développer la logique pour récupérer les détails d'un utilisateur spécifique. (3 heures)

- 
- **Tâche 5** : Créer une interface pour mettre à jour les informations d'un utilisateur existant. (1 heure)
 - **Tâche 6** : Développer la logique pour mettre à jour les informations d'un utilisateur. (45 minutes)
 - **Tâche 7** : Créer une fonctionnalité pour supprimer un utilisateur. (2 heures)
 - **Tâche 8** : Tester toutes les fonctionnalités CRUD pour s'assurer qu'elles fonctionnent correctement.



User Story 5 : En tant qu'utilisateur, je veux pouvoir générer un avatar automatiquement via une API afin de personnaliser mon profil



User Story 6 : En tant qu'utilisateur, je veux pouvoir réinitialiser mon mot de passe via un lien envoyé par email afin de retrouver l'accès à mon compte en cas d'oubli.



GESTION EVENEMENT

1. Module CRUD pour les Événements

2. Module API pour la Génération d'Événements avec l'API Gemini

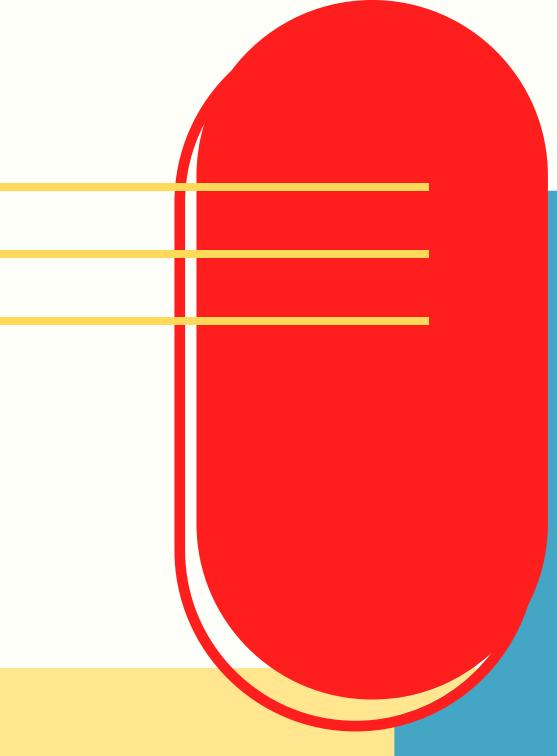
En tant qu'administrateur:

- **User Story 1** : je veux pouvoir créer un événement afin de planifier des activités.
- **User Story 2** : je veux pouvoir lire les détails des événements afin de vérifier les informations.

- **User Story 3** : je veux pouvoir mettre à jour un événement afin de modifier les informations si nécessaire.
- **User Story 4** : je veux pouvoir supprimer un événement afin de gérer efficacement la liste des événements.

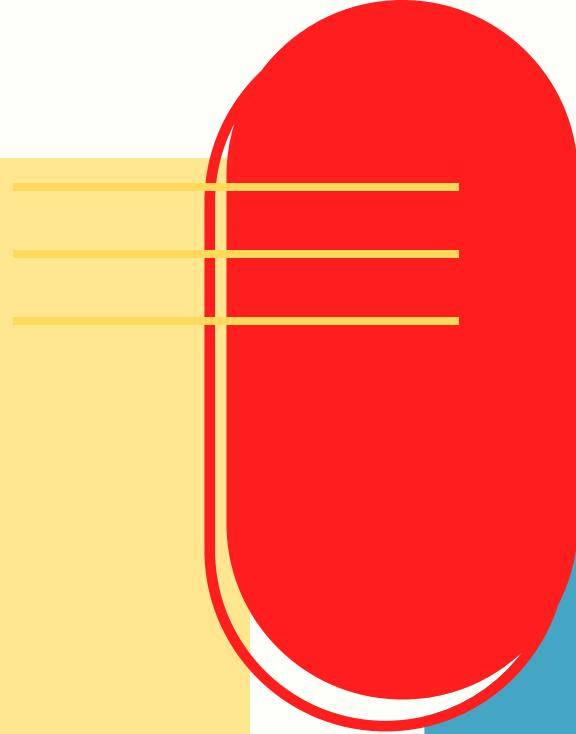
- **Tâche 1** : Créer une interface utilisateur pour la création d'événements. (3 heures)
- **Tâche 2** : Développer la logique pour enregistrer un nouvel événement dans la base de données.(2heures)
- **Tâche 3** : Créer une page pour afficher la liste des événements.(2heures)
- **Tâche 4** : Développer la logique pour récupérer les détails d'un événement spécifique.(3heure)

- 
- **Tâche 5** : Créer une interface pour mettre à jour les événements existants. (1heure)
 - **Tâche 6** : Développer la logique pour mettre à jour les informations d'un événement.(45minutes)
 - **Tâche 7** : Créer une fonctionnalité pour supprimer un événement.(2heure)
 - **Tâche 8** : Tester toutes les fonctionnalités CRUD pour s'assurer qu'elles fonctionnent correctement.



User Story 1 : En tant qu'administrateur, je veux pouvoir générer des événements automatiquement via l'API Gemini afin de simplifier le processus de planification





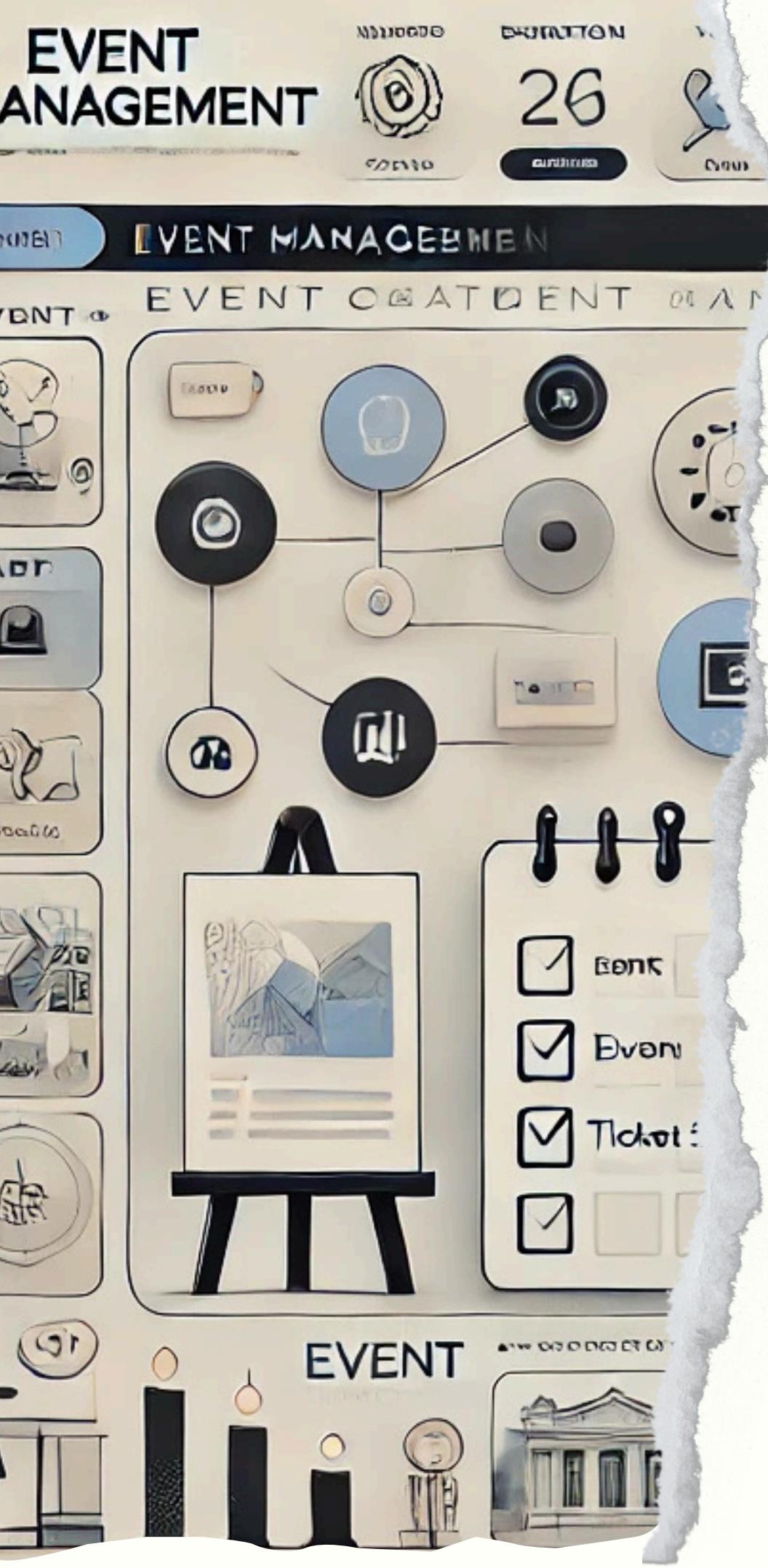
Tâche 1 : Intégrer l'API Gemini dans le projet. (2heures)

Tâche 2 : Créer une interface utilisateur pour permettre à l'administrateur de spécifier les paramètres de l'événement (date, type, etc.). (1heure)



Tâche 3 : Développer la logique pour appeler l'API Gemini et générer des événements basés sur les paramètres fournis. (1heure)

Tâche 4 : Tester la génération d'événements pour s'assurer que les données sont correctes et que l'interface fonctionne comme prévu. (1heure)



GESTION FORMATION

- 1. Module CRUD pour les Formations**
- 2. Module API Whisper pour transcrire et resumer les vidéos**



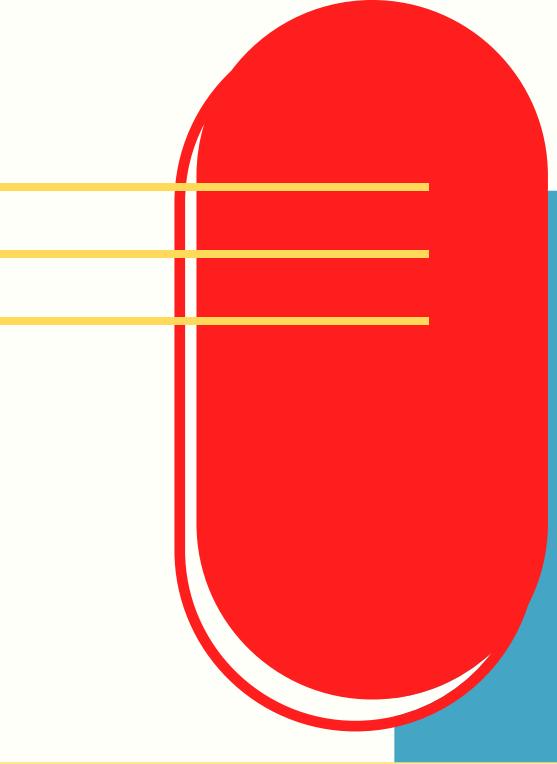
En tant qu'administrateur:

- User Story 1 : En tant qu'administrateur, je veux pouvoir créer une formation afin de planifier des sessions d'apprentissage.
- User Story 2 : En tant qu'administrateur, je veux pouvoir lire les détails des formations afin de vérifier les informations.

- User Story 3 : En tant qu'administrateur, je veux pouvoir mettre à jour une formation afin de modifier les informations si nécessaire.
- User Story 4 : En tant qu'administrateur, je veux pouvoir supprimer une formation afin de gérer efficacement la liste des formations.

- **Tâche 1** : Créer une interface utilisateur pour la création de formations.
- **Tâche 2** : Développer la logique pour enregistrer une nouvelle formation dans la base de données.
- **Tâche 3** : Créer une page pour afficher la liste des formations.
- **Tâche 4** : Développer la logique pour récupérer les détails d'une formation spécifique.

- 
- **Tâche 5** : Créer une interface pour mettre à jour les formations existantes.
 - **Tâche 6** : Développer la logique pour mettre à jour les informations d'une formation.
 - **Tâche 7** : Créer une fonctionnalité pour supprimer une formation.
 - **Tâche 8** : Tester toutes les fonctionnalités CRUD pour s'assurer qu'elles fonctionnent correctement.

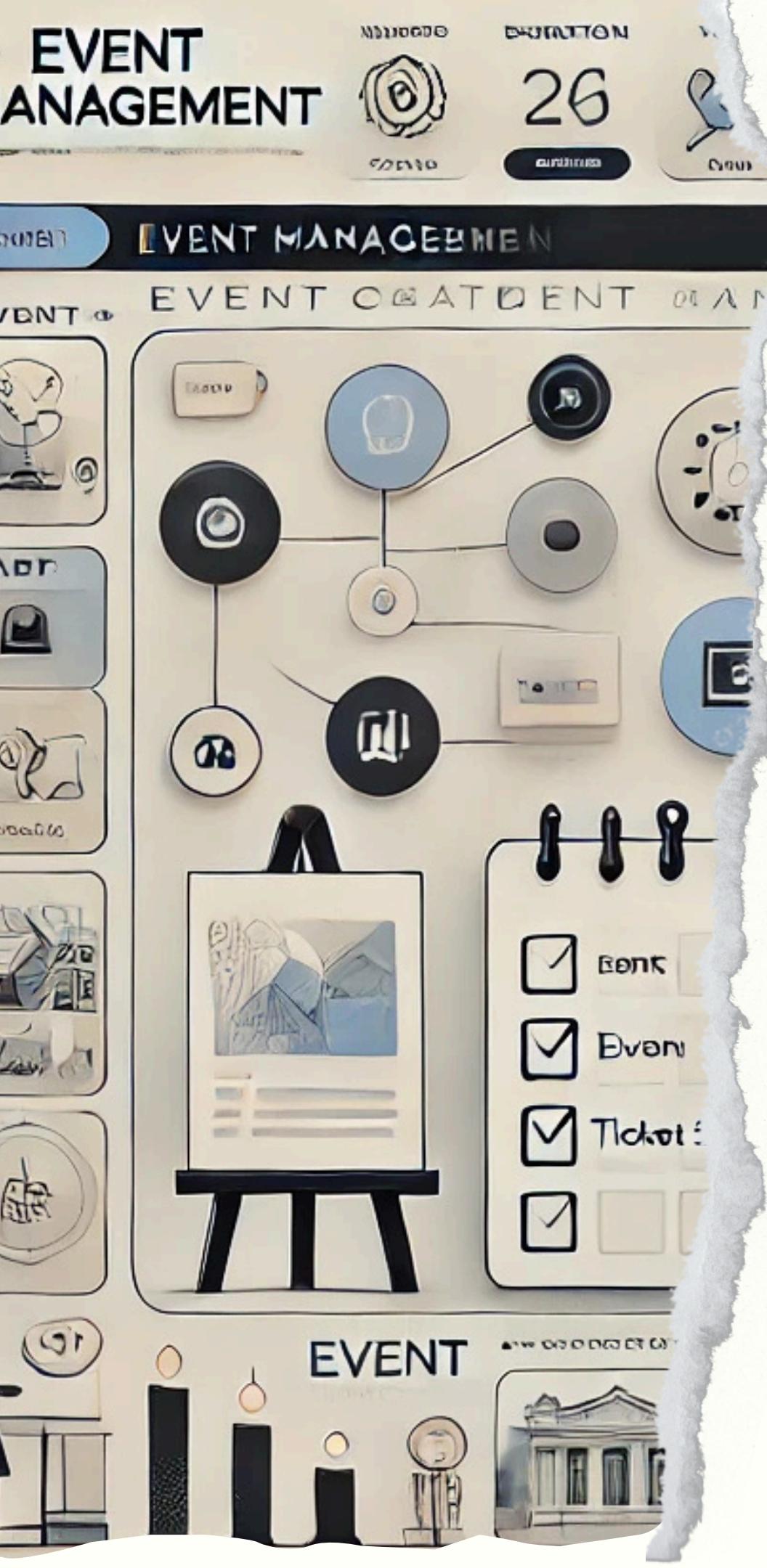


User Story 1 : En tant qu'administrateur, je veux pouvoir transcrire et résumer les vidéos de formation en texte à l'aide de l'API Whisper afin de rendre le contenu accessible sous forme écrite et d'offrir un aperçu concis du contenu.



- **Tâche 1** : Intégrer l'API Whisper. (2 heures)
- **Tâche 2** : Créer une interface utilisateur pour télécharger des vidéos. (4 heures)

- 
- **Tâche 3** : Développer la logique pour transcrire le contenu audio. (3 heures)
 - **Tâche 4** : Développer la logique pour résumer le texte transcrit. (3 heures)
 - **Tâche 5** : Tester la transcription et le résumé. (2 heures)



GESTION FORUM

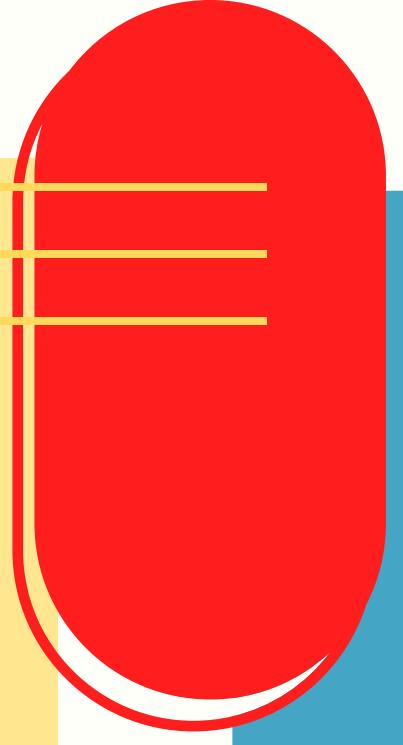
- 1. Module CRUD pour les Forum**
- 2. Module API pour la Recommandation de Posts et le Filtrage des Mots Inappropriés**



En tant qu'administrateur:

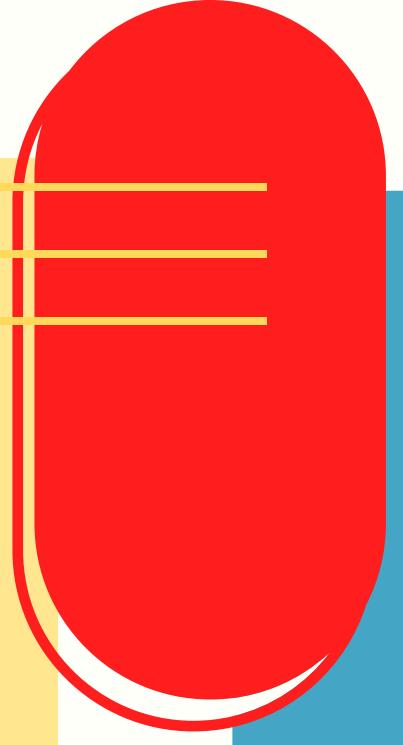
- User Story 1 : En tant qu'administrateur, je veux pouvoir créer un post afin de lancer une discussion sur le forum.
- User Story 2 : En tant qu'administrateur, je veux pouvoir lire les détails d'un post et de ses commentaires afin de vérifier le contenu et les interactions.

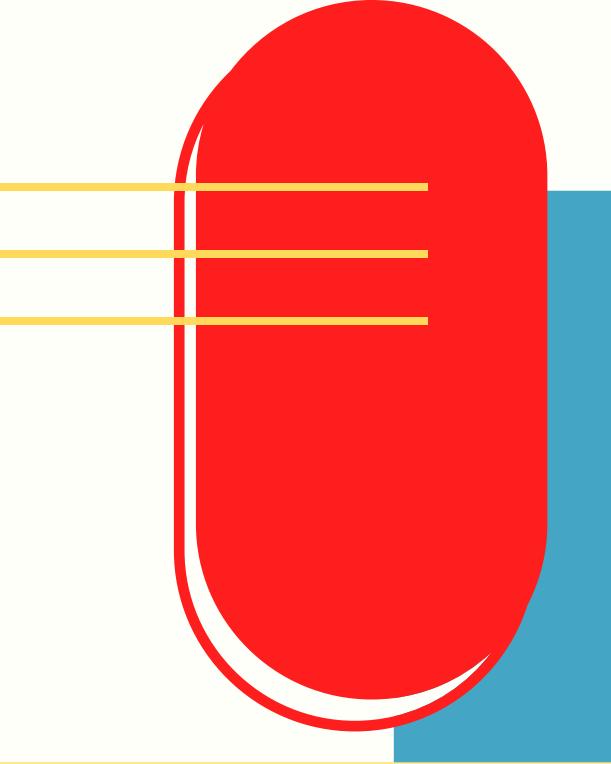
- User Story 3 : En tant qu'administrateur, je veux pouvoir mettre à jour un post ou un commentaire afin de corriger ou modifier les informations si nécessaire.
- User Story 4 : En tant qu'administrateur, je veux pouvoir supprimer un post, un commentaire ou un like afin de maintenir la qualité et la pertinence des échanges sur le forum.



- **Tâche 1** : Créer une interface utilisateur pour la création de posts sur le forum.
- **Tâche 2** : Développer la logique pour enregistrer un nouveau post dans la base de données.
- **Tâche 3** : Créer une page pour afficher la liste des posts du forum.
- **Tâche 4** : Développer la logique pour récupérer les détails d'un post spécifique avec ses commentaires.

- **Tâche 5** : Créer une interface pour mettre à jour les posts et commentaires existants.
- **Tâche 6** : Développer la logique pour mettre à jour les informations d'un post ou d'un commentaire.
- **Tâche 7** :Créer une fonctionnalité pour supprimer un post, un commentaire ou un like.
- **Tâche 8** :Tester toutes les fonctionnalités CRUD du forum pour s'assurer qu'elles fonctionnent correctement.





User Story 1 : En tant qu'administrateur, je veux pouvoir recommander des posts pertinents et filtrer les mots inappropriés dans les discussions du forum à l'aide d'une API afin d'améliorer l'expérience des utilisateurs et maintenir la qualité des échanges.



- **Tâche 1** : Intégrer l'API pour la recommandation de posts et le filtrage des mots inappropriés. (2 heures)
- **Tâche 2** : Créer une interface permettant à l'administrateur de configurer les paramètres du filtrage et de la recommandation. (1 heures)
- **Tâche 3** : Développer la logique pour analyser les posts et détecter les mots inappropriés.(1 heures)

- 
- **Tâche 4** : Implémenter l'algorithme de recommandation de posts en fonction des interactions des utilisateurs.(1 heures)
 - **Tâche 5** : Tester le filtrage des contenus et la recommandation de posts pour s'assurer que le système fonctionne correctement. (1 heures)



GESTION EMPLOI

1. Module CRUD pour les emplois
 2. Module API pour la détection automatiquement des fausses offres d'emploi avec l'API Google Perspective

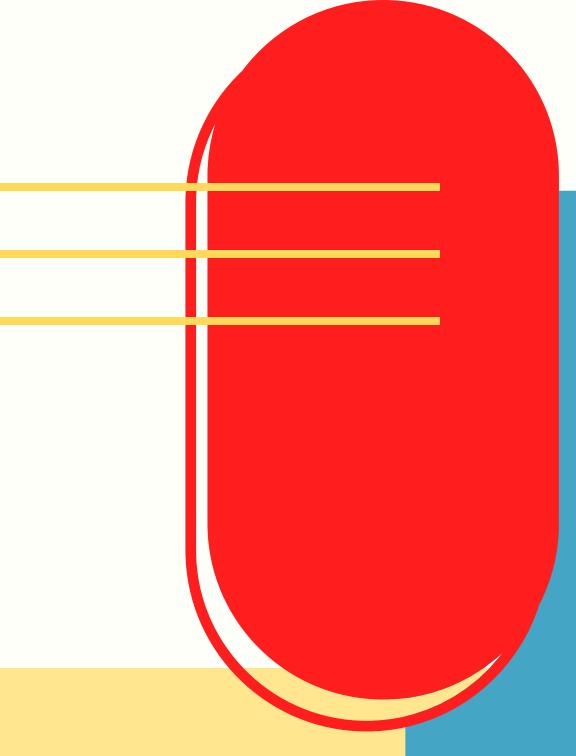
En tant qu'administrateur:

- User Story 1 : je veux pouvoir créer des offres d'emplois
- User Story 2 : je veux pouvoir lire les détails des emplois et des candidatures afin de vérifier les informations.

- User Story 3 : je veux pouvoir mettre à jour des emplois afin de modifier les informations si nécessaire.
- User Story 4 : je veux pouvoir supprimer des emploi afin de gérer efficacement la liste des emplois.

- **Tâche 1** : Créer une interface utilisateur pour la création des emplois. (3 heures)
- **Tâche 2** : Développer la logique pour enregistrer un nouvel emploi dans la base de données.(2heures)
- **Tâche 3** : Créer une page pour afficher la liste des emplois.(2heures)
- **Tâche 4** : Développer la logique pour récupérer les détails d'un emploi spécifique.(3heure)

- 
- **Tâche 5** : Créer une interface pour mettre à jour les emplois existants.(1heure)
 - **Tâche 6** : Développer la logique pour mettre à jour les informations d'un emploi.(45minutes)
 - **Tâche 7** : Créer une fonctionnalité pour supprimer un emploi.(2heure)
 - **Tâche 8** : Tester toutes les fonctionnalités CRUD pour s'assurer qu'elles fonctionnent correctement.



User Story 1 : En tant qu'administrateur, je veux pouvoir détecter automatiquement les fausses offres d'emploi via l'API Google Perspective afin de garantir l'authenticité des offres publiées.



Tâche 1 : Intégrer l'API Google Perspective dans le projet pour détecter les fausses offres d'emploi.
(3 heures)

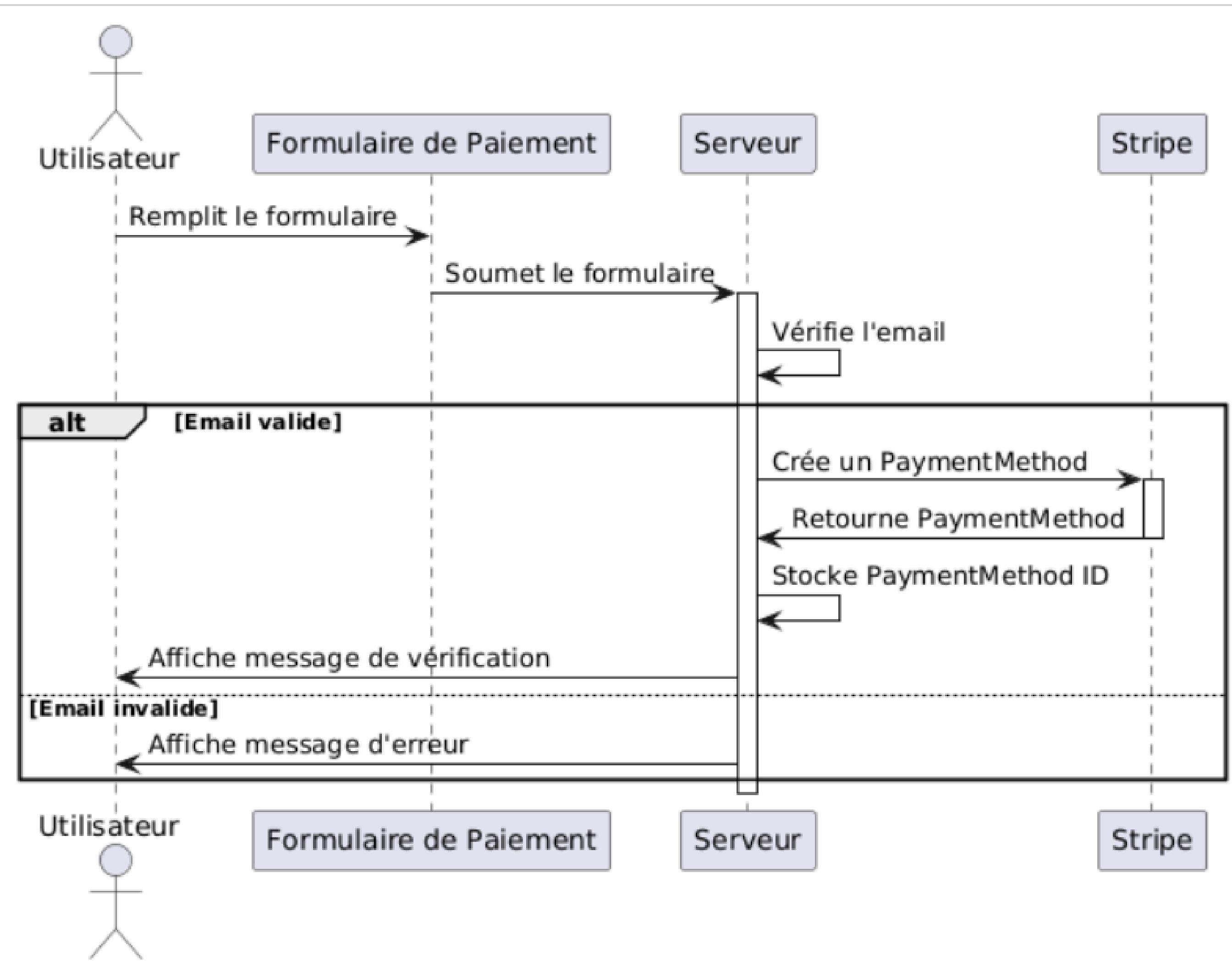
Tâche 2 : Créer une interface utilisateur permettant à l'administrateur de spécifier les paramètres de l'offre d'emploi (titre, description, catégorie, etc.). (1 heure)



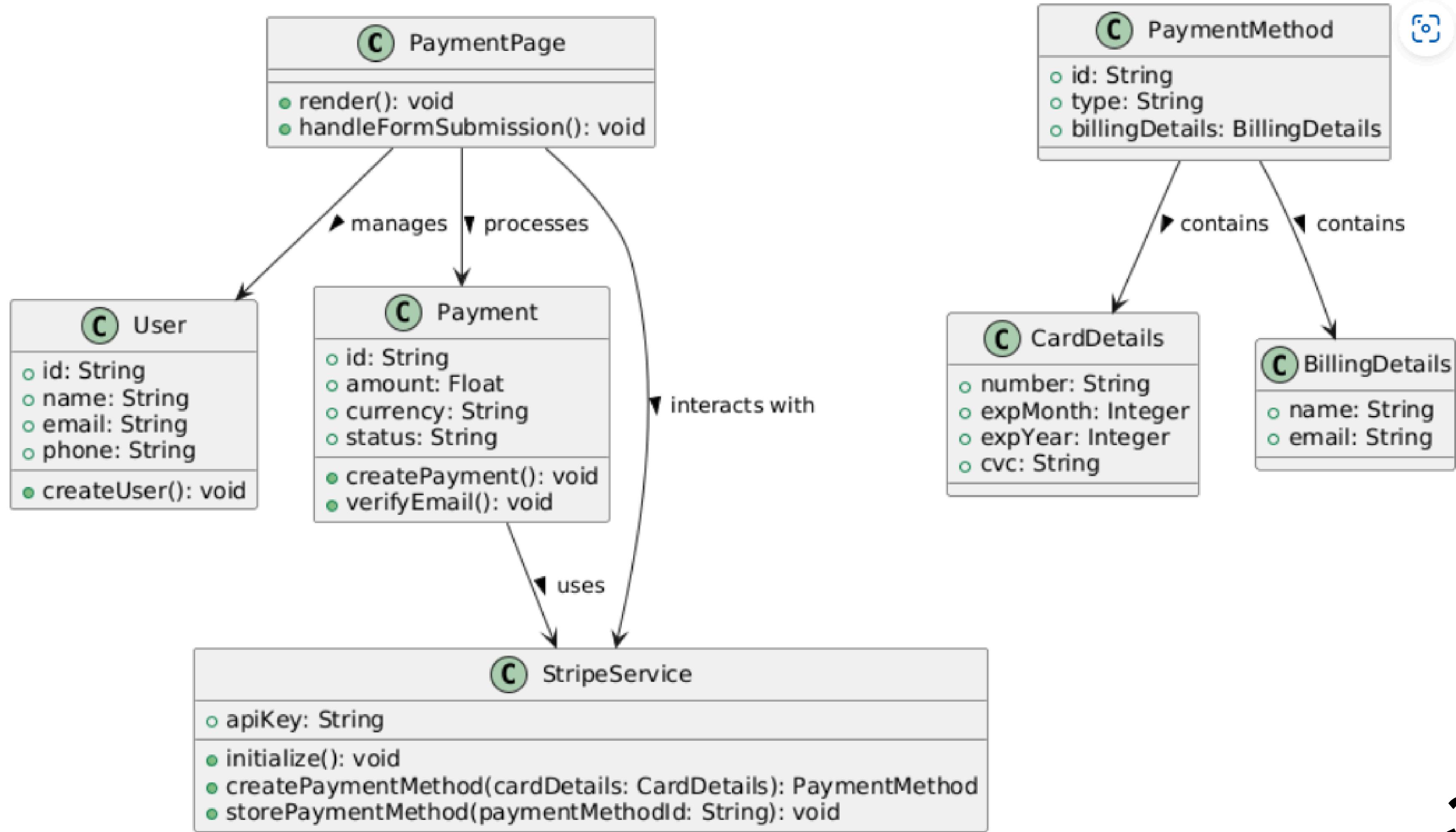
Tâche 3 : Développer la logique pour appeler l'API Google Perspective et analyser les offres d'emploi en fonction des paramètres fournis. (1heure)

Tâche 4 : Tester la détection des fausses offres d'emploi pour s'assurer que les données sont correctes et que l'interface fonctionne comme prévu. (1heure)

3. Diagramme de séquence objets



4. Diagramme de classe de conception





5. Tableau blanc (Trello)



▼ 17/02/2025

	Tâche		Statut	Date
<input type="checkbox"/>	Planification & Initialisation du projet	<input type="button" value="+"/>	Fait	févr. 17
<input type="checkbox"/>	Développement Backend (Entités & Relations)	<input type="button" value="+"/>	Fait	févr. 18
<input type="checkbox"/>	Développement des fonctionnalités CRUD	<input type="button" value="+"/>	Fait	févr. 19
<input type="checkbox"/>	Développement Frontend & Affichage des Offres	<input type="button" value="+"/>	Fait	févr. 20
<input type="checkbox"/>	Tests, Debugging & Améliorations	<input type="button" value="+"/>	Fait	févr. 21
<input type="checkbox"/>	Revue de Sprint & Documentation des api	<input type="button" value="+"/>	En cours	févr. 22
<input type="checkbox"/>	intigration avec github	<input type="button" value="+"/>	En cours	févr. 23
<input type="checkbox"/>	+ Ajouter tâche			
				févr. 17 - 23



Merci pour
votre Attention