

Cahier des charges 2025-2026

L'objectif de ce mini-projet est de mettre en place une application (API backend uniquement pour ce semestre) permettant le **suivi d'un PFE façon Jira mais en incluant aussi le suivi des rapports de PFE et le suivi des réunions**, avec trois rôles : **Étudiant**, **Encadrant entreprise**, **Encadrant universitaire**.

Le système doit couvrir : **Projet** → **Sprints** → **User Stories** → **Tâches**, **validations** par encadrants, **réunions** de suivi (planifiées, puis complétées), **versions de rapports**, et un **dashboard** d'avancement et de journal.

1) Durée et Date de soutenance

- Le projet dure **4 à 5 semaines**
- **Date de soutenances** : les 2 dernières semaines avant les vacances
- **Date du dernier push sur github**: le 07 décembre
- Le projet est réparti en 4 équipes avec 2 étudiants/équipe (8 étudiants)

2) Acteurs

- **Étudiant** : crée le projet PFE, crée les sprints, les user stories et les tâches, met à jour l'avancement, planifie/modifie les réunions, uploades les rapports.
- **Encadrant entreprise** : consulte et **valide/invalid**e l'avancement des **tâches**.
- **Encadrant universitaire** : consulte et **valide/invalid**e les **tâches** et **valide** le **contenu des réunions**.

3) Périmètre (ce semestre)

- **API backend uniquement** (pas d'interface front).
- Tests avec **Postman** (collections à livrer).
- Authentification & rôles disponibles comme socle commun (vus en cours).
- Semestre suivant : développement du **front React** qui consommera ces API.

4) Fonctionnalités attendues

4.1 Gestion du projet

- L'étudiant **crée un projet** PFE.
- Dans le projet, l'étudiant **crée des sprints**.
- Dans chaque sprint, l'étudiant **crée des user stories**.
- Dans chaque user story, l'étudiant **crée des tâches**.

4.2 Tâches & workflow

- Statuts d'une tâche : **ToDo**, **InProgress**, **Standby** (bloquée), **Done**.
- L'étudiant peut **changer** le statut ; l'historique des changements est **consultable**.
- génération d'un rapport html de suivi par sprint
- Génération d'un rapport html global projet

4.3 Validations

- **Encadrant entreprise** et **encadrant universitaire** peuvent **valider/invalid**er l'avancement d'une **tâche**.
- Chaque validation est **horodatée** et **réf**érencée à une **réunion** (ou "hors réunion" si pertinent).

4.4 Réunions

- L'étudiant **planifie** une réunion (date + **ordre du jour**).
- Après la réunion, il **complète** le **compte rendu réel**.
- La réunion peut **réf**érencer : une **User Story**, une **Tâche**, ou une **Version de rapport**.
- **Encadrant universitaire** : **valide** le **contenu** de la réunion.

4.5 Rapports

- L'étudiant peut **uploader plusieurs versions** de son rapport (métadonnées version, date, notes).

- Les encadrants peuvent **consulter l'historique** des versions.
- Une réunion peut **pointer** une version de rapport précise.

4.6 Dashboard

- **Journal du stage** : fil chronologique (réunions, validations, changements de statuts, dépôts de rapports).
- **État d'avancement** global : % tâches Done par projet/sprint/US, items en Standby, validations en attente, dernières réunions et dépôts.

5) Règles de gestion

- **Seul l'étudiant** peut **créer** : projet, sprints, user stories, tâches ; planifier/modifier les réunions ; uploader des rapports.
- Les **encadrants** ne modifient pas le contenu créé par l'étudiant, mais **valident/invalident** là où c'est prévu.
- La validation de tâche ne concerne que les tâches mises à "Done"
- Chaque validation de tâche contient : **statut validé, auteur (encadrant), date, réunion liée** (ou "hors réunion").
- Le **contenu de réunion** doit être **validé** par l'**encadrant universitaire**.

6) Livrables intermédiaires (au moins 4 deadlines)

- Dans le fichier Readme, mettre la répartition du travail par deadline et par équipe et les dates de chaque deadline
- Alimenter le fichier readme avec le travail déjà réalisé
- le travail ne doit pas être fait la dernière semaine avant la livraison finale

7) Livrables

- **API fonctionnelles** couvrant le périmètre.
- **Collections Postman** démontrant les scénarios principaux (de bout en bout).
- Diagramme de classes

8) Organisation (4 équipes, 2 étudiants/équipe)

- **Équipe A : Projets & Sprints** (+ Dashboard agrégations projet/sprint).
- **Équipe B : User Stories & Rapports** (versions).
- **Équipe C : Tâches & workflow** (historiques).
- **Équipe D : Validations & Réunions** (liens vers US/Tâches/Rapports, validation du contenu de réunion).