[Date]

USER

[nom de la société]

[Titre du document]

[Sous-titre du document]

Table des matières

**Aucune entrée de table des matières n'a été trouvée.**

1. **Contexte du projet**

Ce projet a pour objectif de concevoir une plateforme web de gestion des activités des apprentis ingénieurs tout au long de leur cursus en alternance.

Cette plateforme vise à centraliser plusieurs processus clés présents dans la formation et le suivi pédagogique d’un alternant.

* L’inscription de l’apprenti
* Le livret d’apprentissage (journal de formation)
* L’organisation des entretiens semestriels
* Les évaluations pédagogiques et professionnelles
* La planification des soutenances et des jurys

Actuellement, ces activités sont gérées via Microsoft SharePoint, un outil de GED (gestion électronique de documents) qui permet de stocker, classer et accéder aux documents liés aux apprentis. Cependant, l’usage exclusif de SharePoint limite certaines possibilités d’évolution, notamment en ce qui concerne :

* L’automatisation des processus,
* La traçabilité structurée des données,
* L’analyse ou l’exploitation des informations stockées (rapports, évaluations, suivi pédagogique).

Dans une optique, d’automatisation des processus, de traçabilité des donnes, d’analyse et d’exploitation centralisée des informations stockées, le projet propose aux étudiants de réaliser une plateforme web personnalisée, fondée sur une architecture trois tiers, permettant de mieux gérer ces processus et d’expérimenter toutes les dimensions de la conception d’un système d’information.

* Modélisation des donnes (dimension SI)
* Développement logiciel en environnement web (dimension GL)
* Méthodologie rigoureuse de gestion de projet (Scrum)

Ce document présente une version restructurée du cahier des charges enrichies de propositions d’amélioration visant à rendre le projet plus clair, plus pertinent pédagogiquement, et plus cohérent avec les attentes professionnelles.

1. **Limites identifiés dans l’ancien CDC**

|  |  |
| --- | --- |
| **Thèmes** | **Explications** |
| **Formulation du cahier des charges** | * Le document est verbeux et peu structure, rendant la compréhension difficile * Il mélange parfois les besoins fonctionnels, techniques et organisationnels, sans distinction claire * Les schémas présents difficiles à interpréter ou encore à exploiter ce qui nuit à la visualisation globale du système attendu |
| **Ciblage pédagogique peu adapte** | * Le projet est riche mais trop ambitieux pour la durée allouée (1 semestre), surtout pour un étudiant avec peu d’expérience * Certains objectifs ne sont ni priorises ni découpes clairement en versions itératives (MVP, V2…) |
| **Modularité et découpage fonctionnel** | * Le projet propose trop de modules en parallèle, sans logique progressive * Il n’existe pas de flux de navigation utilisateur clair, ce qui complique la conception de l’interface et de l’UX |
| **Méthodologie et outillage** | Aucun socle technique (structure de projet de base) n’est fourni ce qui oblige chaque groupe a repartir de 0 en handicapant d’entrée de jeu les étudiants sans expérience dans la construction technique d’un projet. |
| **Suivi et démonstrabilité** | * Bien qu’ils aient été évoqués plus tard en cours, la mise en production, l’intégration d’un pipeline CI/CD et la mise en place de tests ne sont pas clairement exiges * Aucune base de référence ou critère minimum de qualité (MVP, maquettes, UX, responsive …) n’est clairement précise, ce qui laisse place à des écarts importants d’un groupe a l’autre. |

1. **Objectifs pédagogiques clairs**

* **Comprendre la modélisation des systèmes d’information :** Ce projet permet aux étudiants de s’exercer à la conception d’un système d’information en partant de la modélisation des données. Ils apprennent à construire un schéma conceptuel (MCD), le traduire en schéma logique (MLD), puis en schéma physique (MPD). Cela leur permet de bien comprendre la structure des bases de données relationnelles et leur rôle central dans une application métier. Ils découvrent aussi comment choisir entre stockage structuré (BD) ou non structuré (documents).
* **Appliquer les méthodes du génie logiciel :** Les étudiants doivent développer une application web fonctionnelle en respectant les bonnes pratiques du génie logiciel. Ils découvrent comment découper le code en couches (présentation, métier, données), structurer leurs fichiers, écrire un code lisible et maintenable, mais aussi documenter leur travail. Ils sont encouragés à tester leur code et à utiliser des outils de qualité comme SonarQube ou Jenkins.
* **Travailler en mode agile avec scrum :** Le projet se déroule selon la méthode agile Scrum. Les étudiants apprennent à organiser leur travail en sprints, à planifier leurs tâches, à suivre un backlog et à organiser des revues et rétrospectives. Chaque membre de l’équipe a un rôle précis (Scrum Master, Product Owner, etc.), ce qui favorise la collaboration, la communication et l’autonomie dans l’avancement du projet.
* **Développer une vraie application web :** À travers ce projet, les étudiants développent une application web complète. Ils conçoivent des interfaces simples et efficaces, gèrent des rôles utilisateurs (apprentis, tuteurs, admin...), créent des formulaires et affichent des données dynamiques. Ils se familiarisent avec les bases du développement fullstack et comprennent comment connecter leur application à une base de données.
* **Suivre un projet du début à la fin :** Le projet SIGL permet de vivre tout le cycle de vie d’un projet informatique. De l’analyse des besoins jusqu’au déploiement d’un prototype fonctionnel, les étudiants passent par toutes les étapes clés : rédaction du cahier des charges, modélisation, développement, tests, démonstration. Ils apprennent aussi à produire des livrables clairs et professionnels, comme un Gantt, une documentation technique ou une démo vidéo.

1. **Acteurs et rôles**
2. **Apprenti**

L’apprenti est l’utilisateur principal de la plateforme. Il utilise l’outil pour saisir ses activités en entreprise, compléter son journal de formation, déposer ses livrables (rapports, fiches de synthèse, slides), et organiser ses entretiens semestriels. Il peut consulter ses évaluations et être notifié des échéances importantes.

1. **Tuteur pedagogique**

Le tuteur pédagogique suit un ou plusieurs apprentis du côté école. Il lit et commente les documents déposés, participe aux entretiens, évalue le travail de l’apprenti et peut attribuer des notes. Il veille à la progression académique de l’étudiant en entreprise.

1. **Maitre d’apprentissage**

Il encadre l’apprenti dans son entreprise. Il complète la grille de compétences à la fin de chaque semestre, commente les livrables et participe aux entretiens semestriels. Il a accès aux documents de l’apprenti liés à son activité professionnelle.

1. **Coordinatrice d’alternance**

Elle gère la logistique de l’alternance au sein de l’école. Elle crée les comptes des entreprises partenaires, saisit les tuteurs et maîtres d’apprentissage, programme les périodes d’entretiens et de soutenances. Elle est au centre de l’organisation pédagogique et administrative.

1. **Responsable du cursus**

Il valide les missions proposées pour les apprentis, suit l’évolution de chaque promotion, et intervient dans les prises de décision importantes. Il a une vision globale sur les activités et peut intervenir en cas de problème de progression.

1. **Assistante des departements**

Elle apporte un soutien administratif pour la gestion des documents ou la planification. Son rôle peut inclure la gestion de fichiers de suivi, les relances, ou la mise à disposition d’éléments administratifs.

1. **Entreprise partemaire**

L’entreprise partenaire est responsable de fournir les informations nécessaires pour inscrire un apprenti. Elle accède à certaines informations concernant le suivi de l’apprenti (via le maître d’apprentissage ou un représentant RH).

1. **Professeurs ESEO**

Ils peuvent intervenir comme membres de jury lors des soutenances. Le professeur président du jury saisit la note attribuée à l’oral et peut accéder aux livrables de l’apprenti pour évaluation.

1. **Administrateur de la plateforme**

Il gère techniquement les comptes utilisateurs, attribue les rôles, réinitialise les mots de passe, supprime les comptes obsolètes, et veille au bon fonctionnement de la plateforme. Il peut aussi configurer des événements ou activer des modules.

1. **Direction des etudes**

Elle supervise l’ensemble du dispositif pédagogique de l’alternance. Elle a une vision transverse sur l’utilisation de la plateforme, la qualité des suivis et l’adéquation entre les missions en entreprise et les objectifs de formation.

1. **Fonctionnalités principales (simplifiées ou clarifiées)**

La plateforme doit permettre de centraliser et de faciliter la gestion des activités liées à l'alternance pour les apprentis, les tuteurs, les maîtres d'apprentissage et les responsables pédagogiques. Voici les grandes fonctionnalités, réorganisées pour plus de lisibilité:

1. **Gestion des utlisateurs et roles**

* Connexion et déconnexion sécurisées (authentification par rôle).
* Gestion des mots de passe (changement ou réinitialisation).
* Création de comptes par l’administrateur (apprentis, tuteurs, maîtres d’apprentissage…).
* Attribution automatique ou manuelle des rôles à chaque utilisateur.

1. **Journal de formation (Livret d’apprentissage)**

* Saisie périodique des activités professionnelles par l’apprenti (note mensuelle).
* Dépôt de documents à différentes étapes (fiche de synthèse, rapports, slides).
* Consultation et annotation du journal par le tuteur pédagogique et le maître d’apprentissage.
* Historique structuré et daté des contributions de l’apprenti.

1. **Organisation des entretiens semestriels**

* Planification des entretiens (créneaux proposés selon les disponibilités).
* Enregistrement automatique des périodes de suivi par semestre.
* Interface de validation/participation aux entretiens pour les 3 parties : apprenti, tuteur, maître d’apprentissage.

1. **Evaluations**

* Saisie des grilles d’évaluation par le maître d’apprentissage à chaque fin de semestre.
* Attribution des notes par le tuteur pédagogique pour certains modules (conduite de projet, etc.).
* Historique des évaluations accessible depuis la fiche de l’apprenti.

1. **Gestion des soutenances et des jurys**

* Planification des soutenances à partir du semestre 7.
* Affectation des membres de jury (interne + externe).
* Dépôt et visualisation des supports de soutenance (slides).
* Saisie des notes par les membres du jury directement depuis la plateforme.

1. **Notifications et rappels**

* Paramétrage de notifications (mail, ou affichage interne) pour rappeler :
  + Dates d’entretiens
  + Dépôt de documents
  + Échéances importantes
* Affichage d’une timeline personnalisée pour chaque utilisateur

1. **Documents et livrables**

* Dépôt des documents attendus selon le semestre :
  + Note mensuelle
  + Fiches de synthèse
  + Rapports de projet (PING, final)
  + Slides de soutenance
* Visualisation et téléchargement des documents autorisés selon le rôle

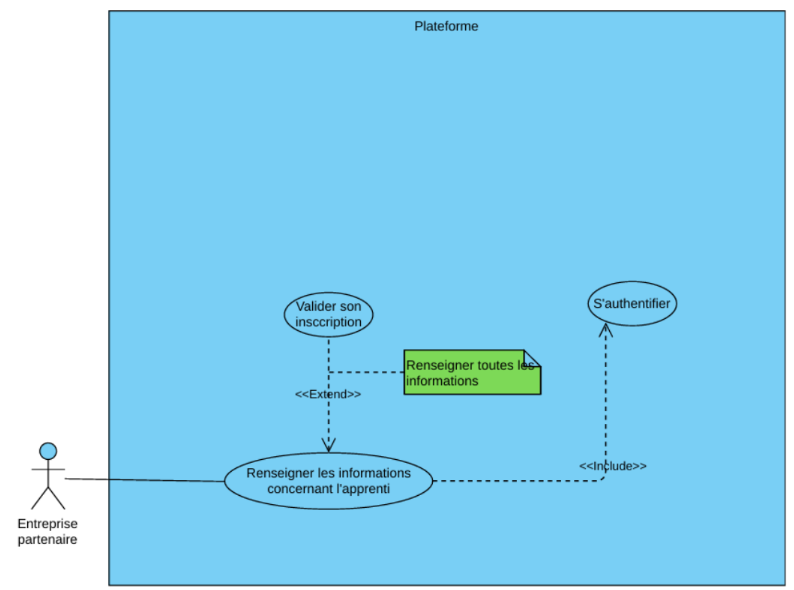
1. **Suivi global de la progression**

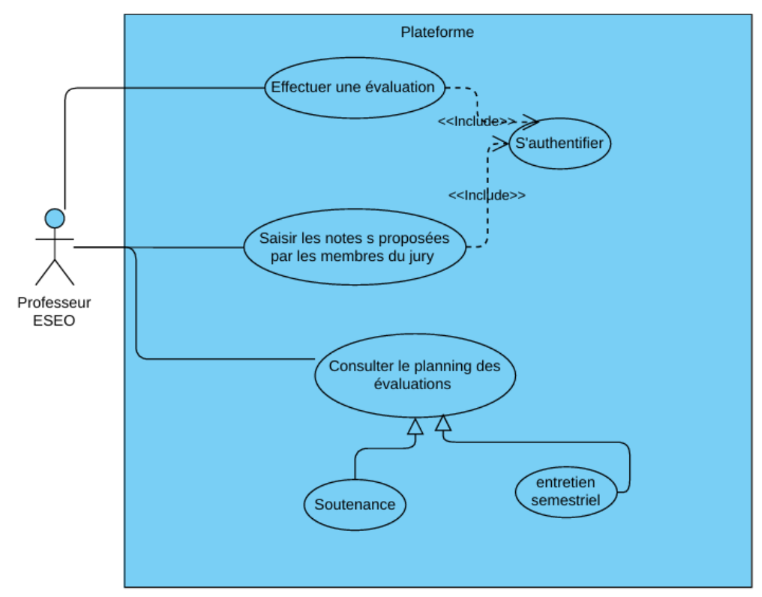
* Tableau de bord pour chaque acteur :
  + Apprenti : état des livrables, entretiens à venir, évaluations
  + Tuteur et maître d’apprentissage : liste des apprentis suivis, alertes
  + Administrateur / coordinateur : suivi de la promo, statistiques, relances

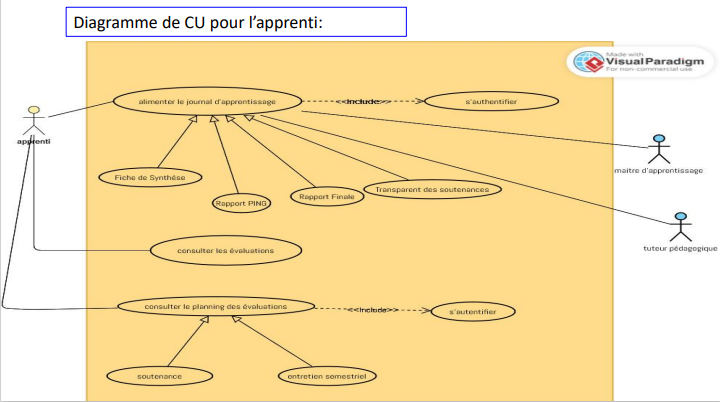
1. **Cas d’utilisation + diagrammes**

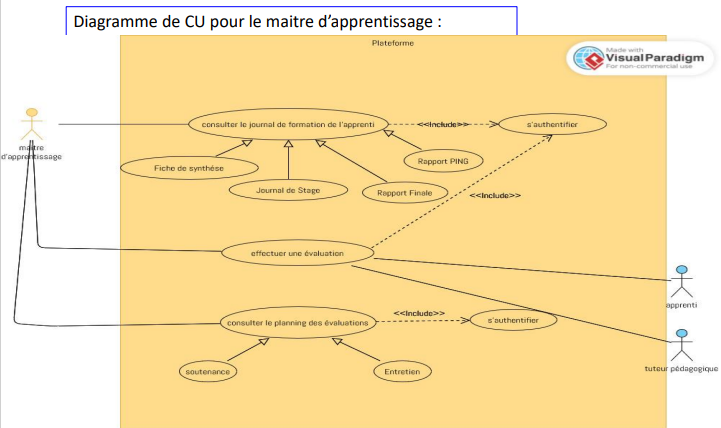
Les cas d’utilisation définissent les interactions attendues entre les utilisateurs (acteurs) et la plateforme. Pour simplifier l’analyse et la rendre plus accessible aux étudiants, ils sont regroupés en 5 profils d’acteurs principaux : Apprenti, Maître d’apprentissage, Tuteur pédagogique, Coordinatrice de l’alternance, professeurs ESEO, responsable de cursus, Administrateur.

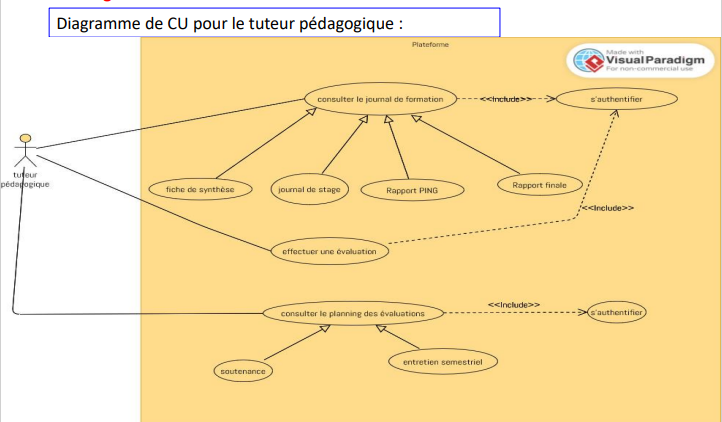
Chaque acteur interagit avec la plateforme à travers un ensemble limité mais clair de fonctionnalités centrées sur son rôle. Voici les diagrammes de cas d’utilisation des utilisateurs pour lesquels on peut recenser des interactions directes avec la plateforme :

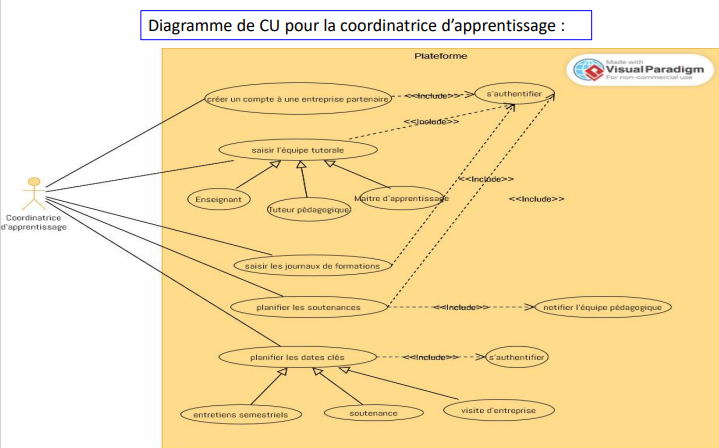


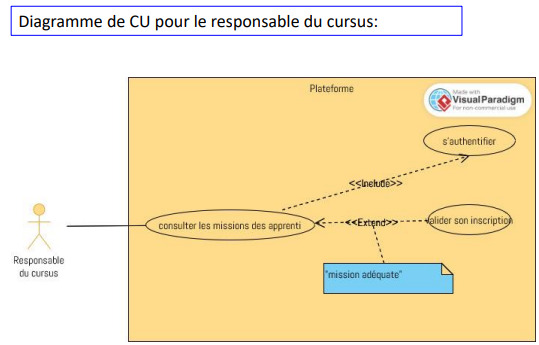












1. **Analyse des besoins (fonctionnels + non fonctionnels)**
2. **Besoins fonctionnels**

Les besoins fonctionnels décrivent l’ensemble des actions que le système doit permettre de réaliser. Ils sont organisés ici par domaines fonctionnels:

1. Gestions des utilisateurs
   * Connexion et déconnexion sécurisées
   * Création, modification et suppression de comptes utilisateurs
   * Attribution automatique ou manuelle de rôles selon le profil (apprenti, tuteur, etc…)
   * Réinitialisation de mot de passe par l’administrateur
2. Journal de formation
   * Saisie des activités mensuelles par l’apprenti
   * Dépôt des documents (fiche de synthèse, rapport PING, rapport final, slides)
   * Accès au journal pour les tuteurs et maitres d’apprentissage
   * Historique structure et date des contributions
3. Entretiens semestriels
   * Planification des entretiens avec proposition de créneaux
   * Validation de la présence par les 3 parties (apprenti, tuteur, maitre)
   * Renseignement d’un compte rendu d’entretien
4. Evaluations
   * Dépôt des grilles d’évaluation par les maitres d’apprentissage
   * Saisie des notes par les tuteurs pédagogiques
   * Historique des évaluations par semestre
5. Soutenances et jurys
   * Planification des soutenances (créneaux, jurys, salles)
   * Dépôt des supports de présentation
   * Attribution des notes et commentaires par les membres du jury
6. Notifications et rappels
   * Paramétrage et envoie automatique de rappels (emails, interface)
   * Affichage d’une timeline personnalisée pour chaque utilisateur
7. Suivi et visualisation
   * Tableau de bord de progression pour chaque utilisateur
   * Vue consolidée pour les coordinateurs et responsables pédagogiques
   * Export données (Excel et PDF)
8. **Besoins non fonctionnels**

Les besoins non fonctionnels concernant la qualité d’exécution, la sécurité et l’accessibilité du système.

1. Accessibilité et ergonomie
   * Interface intuitive et responsive (desktop + mobile)
   * Navigation adaptée a chaque rôle utilisateur
   * Design simple, clair, sans surcharge d’informations
2. Sécurité
   * Authentification obligatoire
   * Gestion des autorisations selon les rôles
   * Respect des règles de confidentialité des données personnelles
3. Performance et fiabilité
   * Temps de réponse inferieur a 2 secondes pour les opérations standards
   * Gestion des erreurs et messages explicites
   * Sauvegarde régulière des données sensibles
4. Modularité et évolutivité
   * Code structure en modules fonctionnels
   * Possibilité d’ajouter de nouveaux rôles ou documents dans le futur
5. Traçabilité
   * Historique des modifications sur les journaux et les évaluations
   * Journalisation des connexions, dépôts et actions sensibles
6. Déploiement
   * Déploiement simple via GitHub Actions ou autres outils CI/CD
7. **Contraintes techniques**

La plateforme SIGL doit être conçue en tenant compte de plusieurs contraintes techniques, adaptées à un contexte académique, sans nécessité de déploiement public ou de pipeline CI/CD.

1. Environnement de développement
   * L’application sera une application web en mode client-serveur
   * Le développement pourra s’appuyer sur un framework backend (Django, flask, Node.js) et framework frontend (React, Vue.js…)
   * L’environnement devra être compatible avec les postes étudiants (Windows/Linux/Mac)
2. Architecture logicielle
   * L’architecture retenue suivra un modèle en 3 couches :
     + Présentation (interface utilisateur)
     + Métier (logique applicative)
     + Données (accès à la base)
   * Le code sera organisé de manière modulaire pour faciliter la maintenance
3. Base de données
   * Le système reposera sur une base de données relationnelle (PostgreSQL)
   * Les documents déposés par les utilisateurs (fiches, rapports) seront enregistrés sur le serveur, avec référence en base
4. Sécurité et gestion des accès
   * Une authentification utilisateur sera obligatoire
   * Chaque fonctionnalité devra être accessible selon le rôle de l’utilisateur connecte (apprenti, tuteur, etc …)
   * Les mots de passe seront stockes de manière sécurisée (hachage)
5. Gestion des livrables et documentation
   * La documentation du projet sera gérée localement sous forme de fichiers classiques (Word, PDF…)
6. Outils et gestion de projet
   * Le projet sera géré en méthode Scrum, avec organisation par sprints
   * Les taches, stories et plannings seront suivis via JIRA
   * L’utilisation de Git et Github est encouragée pour le versionnage du code
7. **Définition du périmètre MVP (Minimum Viable Product)**

Le Minimum Viable Product (MVP) correspond à la version minimale de la plateforme SIGL qui sera développée et présentée à l’administration le 30 juin 2025. Il s’agit d’un produit fonctionnel intégrant les fonctionnalités essentielles, permettant de démontrer les bénéfices des améliorations proposées, tout en restant réalisable dans le temps imparti.

1. **Objectifs du MVP**

* Mettre en œuvre les fonctionnalités fondamentales du parcours alternant.
* Illustrer l’amélioration de l’expérience utilisateur grâce à une interface claire et structurée.
* Fournir une base stable pour une éventuelle extension future par d’autres groupes d’étudiants.

1. **Fonctionnalités incluses dans le MVP**

|  |  |
| --- | --- |
| **Module** | **Fonctionnalité** |
| **Authentification** | Connexion avec rôles (Apprenti, tuteur, maitre, coordonnateur) |
| **Journal de formation** | Saisie de la note mensuelle + dépôt des documents associes |
| **Espace utilisateur** | Tableau de bord avec rappels de livrables et entretiens |
| **Entretiens semestriels** | Création d’un entretien par l’apprenti, consultation par le tuteur |
| **Evaluations** | Dépôt d’une évaluation simple par le maitre d’apprentissage |
| **Gestion des utilisateurs** | Création des comptes par la coordinatrice de l’alternance |

1. **Fonctionnalités exclues (hors MVP)**

|  |  |
| --- | --- |
| **Module** | **Fonctionnalité** |
| **Soutenance et jury** | Planification, note finale, affectation des membres |
| **Notifications automatiques** | Envoi de mails/SMS, alertes système |
| **Echéancier complet** | Vue calendrier par utilisateur |
| **Export PDF/Excel** | Génération de documents formates |
| **CI/CD et hébergement** | Déploiement en ligne, tests automatises |

1. **Planning previsionnel**

|  |  |  |
| --- | --- | --- |
| **Semaine** | **Sprint** | **Objectifs principaux** |
| Semaine 1 | **Sprint 0 : cadrage** | Finalisation du cahier des charges, planification, création des user stories dans JIRA |
| Semaine 2 | **Sprint 1** | Mise en place du projet (base code, login, rôles), création base de données, première interface |
| Semaine 3 | **Sprint 2** | Module Journal de formation : saisie + dépôt, affichage apprenti + lecture tuteur |
| Semaine 4 | **Sprint 3** | Module Entretien : création par apprenti, consultation par encadrants, affichage planning |
| Semaine 5 | **Sprint 4** | Évaluation : saisie d’une note et commentaire par le maître d’apprentissage |
| Semaine 6 | **Sprint 5** | Tableau de bord utilisateur + gestion des comptes par coordinatrice |
| Semaine 7 | **Sprint 6 : finalisation** | Tests manuels, corrections, présentation, préparation démo + livrables finaux |

1. **Améliorations pédagogiques proposes**

* Séparation explicite des besoins fonctionnels et non fonctionnels.
* Ajout d’une section dédiée au MVP pour cadrer les livrables.
* Hiérarchisation claire des modules inclus/exclus dans le développement.
* Recentrage du périmètre sur les fonctionnalités prioritaires.
* Mise en valeur du MVP comme outil central de démonstration de la valeur.
* Meilleure articulation entre les objectifs d’apprentissage et les livrables attendus.
* Intégration progressive des modules fonctionnels selon la logique d’un parcours alternant.
* Vision plus claire des interactions entre les acteurs pédagogiques et professionnels.