

Table des matières

Introduction Générale	3
1 Présentation et contexte du projet	4
1.1 Introduction	4
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Organisme d'accueil	4
1.2.2 Domaines d'activités	4
1.2.3 Effectifs et organisation	5
1.3 Présentation du projet	5
1.3.1 Problématique	5
1.3.2 Objectifs du projet	5
1.4 Étude et critique de L'existant	6
1.4.1 Analyse de l'existant	6
1.4.2 Coursera	6
1.4.3 Udemy	7
1.4.4 edX	7
1.4.5 Khan Academy	7
1.5 Solution proposée	8
1.6 Cadre du travail adopté	9
1.7 Environnement de travail	9
1.7.1 Environnement matériel	9
1.7.2 Méthode de gestion de projet	9
1.7.3 Outils et Technologies	9
1.7.4 Approches de gestion de projet : Agile vs Traditionnelle	14
1.7.5 Choix méthodologique	15
1.7.6 Présentation de la méthodologie Scrum	15
1.7.7 Outils SCRUM	16
1.7.8 Planning	17
1.8 Conclusion	17
2 Analyse et spécification des besoins	18
2.1 Introduction	18
2.2 Analyse des besoins	18
2.2.1 Identification des acteurs	18
2.2.2 Besoins fonctionnels	19
2.2.3 Besoins Non fonctionnelle	20
2.3 Pilotage du projet avec SCRUM	21

2.3.1	Equipe et rôles	21
2.3.2	L'équipe du projet	21
2.4	Backlog de produit	22
2.5	Diagramme de cas d'utilisation global	23
2.5.1	Diagramme de cas d'utilisation global	23
2.6	Architecture de l'application	25
2.6.1	Communication client-serveur	25
2.6.2	Architecture logicielle :	26
2.7	Planification des sprints	27
3	SPRINT 1 :	28
3.1	Introduction	28
3.2	Backlog du sprint 1	28
3.3	Spécification fonctionnelle	29
3.3.1	Diagramme de cas d'utilisation du sprint 1	29
3.3.2	diagramme de cas d utilisation du sprint 1	29
4	SPRINT 2 :	32
4.1	Introduction	32
4.2	Backlog du sprint 2	33
4.2.1	Diagramme de séquence du sprint 1	33
5	SPRINT 3 :	34
5.1	Introduction	34
5.2	Backlog du sprint 3	35
5.3	*****	35
6	SPRINT 4 :	36
6.1	Introduction	36
6.2	Backlog du sprint 4	36

Introduction Générale

Dans une ère où la technologie transforme profondément nos modes de vie, l'éducation n'échappe pas à cette évolution. Le développement rapide du numérique, l'expansion mondiale d'Internet et la multiplication des appareils connectés ont modifié la manière dont les individus accèdent à l'information et se forment. Aujourd'hui, il n'est plus nécessaire d'être physiquement présent dans une salle de classe pour apprendre. L'apprentissage en ligne, aussi appelé *e-learning*, s'est imposé comme une solution efficace, flexible et accessible, répondant à des besoins variés dans des contextes aussi bien scolaires que professionnels.

Cette nouvelle forme de formation permet aux apprenants de suivre des cours à leur propre rythme, de consulter des ressources éducatives à tout moment, et de s'adapter à des contraintes personnelles ou professionnelles. Grâce à des supports variés vidéos explicatives, documents téléchargeables, quiz interactifs, forums de discussion, l'expérience d'apprentissage devient plus dynamique et mieux adaptée à chaque profil. Ces plateformes attirent un public très large : étudiants en quête de renforcement, professionnels en reconversion, ou simples curieux souhaitant acquérir de nouvelles compétences.

L'un des grands atouts de ces environnements numériques est la possibilité de personnaliser le parcours de formation. En suivant les progrès de chaque utilisateur, les plateformes peuvent adapter les contenus, suggérer des exercices ciblés ou proposer un accompagnement plus précis. Certains systèmes intègrent également des outils d'évaluation automatique, qui permettent un retour immédiat et contribuent à renforcer la motivation des utilisateurs. D'autres fonctionnalités, telles que les espaces de discussion entre apprenants et formateurs, renforcent l'aspect collaboratif de la formation, en créant une véritable communauté d'apprentissage.

De plus, l'apprentissage en ligne joue un rôle important dans la démocratisation de l'accès à l'éducation. Il permet à des personnes vivant dans des zones éloignées, ou ne pouvant se déplacer, de bénéficier de contenus pédagogiques de qualité. Il favorise aussi la formation tout au long de la vie, dans un contexte où les exigences du marché du travail évoluent rapidement et où la mise à jour des compétences devient indispensable.

Cependant, la création d'une plateforme d'apprentissage en ligne ne se résume pas à l'ajout de ressources pédagogiques sur Internet. Il s'agit d'un projet complexe, qui nécessite de prendre en compte l'expérience utilisateur, la qualité pédagogique des contenus, la sécurité des données, la performance technique de l'outil, ainsi que la capacité à évoluer selon les besoins des utilisateurs. Il convient également de concevoir des interfaces simples, intuitives, et accessibles à tous, tout en garantissant une navigation fluide.

Ce rapport présentera les différentes étapes de conception et de développement d'une telle solution, en mettant en lumière les choix technologiques réalisés, les contraintes rencontrées, ainsi que les stratégies adoptées pour assurer la fiabilité, l'efficacité et l'accessibilité du produit final.

Chapitre 1

Présentation et contexte du projet

1.1 Introduction

Ce premier chapitre, intitulé « Contexte général », vise à replacer notre projet dans un cadre global. Nous commencerons par la présentation de la société d'accueil « CES LAB ». Ensuite, nous procéderons à une étude de l'existant suivie d'une critique. Nous présenterons ensuite la solution proposée. Enfin, ce chapitre se terminera par la présentation de la méthodologie de gestion de projet adaptée.

1.2 Présentation de l'organisme d'accueil

1.2.1 Organisme d'accueil

Le laboratoire de recherche *Computer and Embedded Systems* (CES LAB) a été créé en 2002 au sein du Département de Génie Électrique de l'École Nationale d'Ingénieurs de Sfax (ENIS). Le laboratoire CES a une vocation polyvalente. Les travaux de recherche menés au sein du laboratoire se concentrent sur des thématiques ciblées dans les domaines des Nouvelles Technologies de l'Information et des Communications ainsi que de la Maîtrise de l'Énergie. On peut citer notamment l'informatique, la microélectronique et les télécommunications.

Une grande partie des recherches réalisées s'inscrit dans le cadre de la préparation des thèses de doctorat et des stages de Master, ainsi que dans des collaborations avec des partenaires universitaires et industriels, tant nationaux qu'internationaux.

1.2.2 Domaines d'activités

La vocation du CES est de contribuer :

- au développement de la recherche dans les domaines des Nouvelles Technologies de l'Information, des Communications et de la Maîtrise de l'Énergie;



FIGURE 1.1 – Enter Caption

- au développement de l'économie nationale en ciblant la recherche appliquée à l'industrie locale ;
- à la mise en place de projets de recherche nationaux et internationaux ;
- à la formation et à l'encadrement de formateurs et de jeunes chercheurs ;
- à l'organisation de manifestations scientifiques nationales et internationales.

1.2.3 Effectifs et organisation

Dirigé par le professeur Mohamed ABID, le laboratoire CES compte 87 chercheurs et enseignants-chercheurs répartis comme suit :

- 4 Professeurs
- 3 Maîtres de conférences
- 27 Maîtres-Assistants
- 20 Assistants
- 5 Technologues
- 50 Doctorants
- 30 Étudiants en Master Recherche

1.3 Présentation du projet

1.3.1 Problématique

Avec l'essor constant des technologies numériques, l'apprentissage en ligne s'est imposé comme une solution incontournable dans le domaine de l'éducation. Bien qu'efficaces sur plusieurs plans, les plateformes actuelles présentent encore des lacunes importantes. L'absence d'interaction humaine réelle, les parcours peu personnalisés et les dispositifs d'évaluation souvent standards nuisent à l'efficacité de l'apprentissage. De plus, la manière dont l'accès aux contenus est structuré, ainsi que certains modèles économiques, limitent l'accessibilité pour une partie des utilisateurs. Ces éléments témoignent d'un besoin réel d'amélioration afin de proposer des environnements plus adaptés, plus engageants et plus souples, capables de répondre aux attentes d'une communauté d'apprenants de plus en plus diversifiée.

1.3.2 Objectifs du projet

L'objectif général de ce projet est de développer une plateforme d'apprentissage en ligne qui améliore la qualité de l'expérience utilisateur, tant pour les apprenants que pour les formateurs. Les objectifs spécifiques sont les suivants :

- Offrir un accès flexible à la formation, permettant aux apprenants d'acheter et de suivre des cours en ligne à leur propre rythme.
- Faciliter la création et la gestion des cours, en offrant aux formateurs la possibilité de publier du contenu sous différents formats (vidéos, PDF, exercices interactifs, etc.).



FIGURE 1.2 – logo coursera

- Améliorer l'interaction entre utilisateurs via un système de messagerie entre étudiants et enseignants.
- Intégrer un chatbot intelligent pour fournir une assistance instantanée aux apprenants.
- Mettre en place un système de certification pour récompenser les étudiants ayant terminé une formation avec succès.
- Assurer un suivi personnalisé des apprenants avec recommandations adaptées.
- Proposer un modèle économique adapté, notamment un système de paiement à l'unité pour garantir une accessibilité financière.
- Favoriser l'apprentissage interactif, avec des fonctionnalités telles que la correction automatique des exercices et l'évaluation des compétences.

1.4 Étude et critique de L'existant

Dans cette partie, nous allons étudier les solutions présentes sur le marché. Cette étude permettra d'identifier les points forts et les points faibles chez chaque solution ainsi les technologies existantes afin de définir l'allure générale de notre solution proposée.

1.4.1 Analyse de l'existant

Des nombreuses plateformes d'apprentissage en ligne ont vu le jour ces dernières années, proposant une grande variété de contenus pédagogiques et de modalités d'enseignement. Parmi les plus connues, on retrouve Coursera, Udemy, edX ou encore Khan Academy, qui offrent des formations dans des domaines divers comme la programmation, le marketing, la gestion de projet, ou les langues.

1.4.2 Coursera

Coursera, par exemple, collabore avec des universités prestigieuses pour proposer des cours en ligne certifiants. Elle permet un apprentissage structuré avec des vidéos, des quiz, des travaux notés et parfois même des projets à rendre. Cependant, malgré sa qualité, l'interaction entre apprenants et formateurs reste limitée, et certaines formations nécessitent un abonnement payant pour accéder aux ressources complètes.

1.4.3 Udemy

Udemy, de son côté, se distingue par un modèle plus libre où chaque formateur peut publier ses propres cours. Bien qu'elle offre une grande diversité de contenus, la qualité pédagogique varie fortement d'un cours à l'autre, et le suivi personnalisé y est quasi inexistant.



FIGURE 1.3 – logo udemy

1.4.4 edX

edX, similaire à Coursera, se focalise aussi sur des cours universitaires, avec un accent sur la rigueur académique. Certaines de ses fonctionnalités avancées, comme les parcours diplômants ou les certifications vérifiées, ne sont toutefois accessibles qu'aux apprenants payants.



FIGURE 1.4 – logo edX

1.4.5 Khan Academy

Khan Academy propose un modèle gratuit destiné principalement aux élèves du primaire et du secondaire. La plateforme est appréciée pour sa simplicité et son efficacité, mais elle s'adresse davantage à un public scolaire qu'à des professionnels ou étudiants du supérieur.



FIGURE 1.5 – logo Khan Academy

Dans l'ensemble, bien que ces plateformes aient permis de démocratiser l'apprentissage à distance, elles présentent des limites communes : manque d'interaction humaine, peu de personnalisation du parcours, accès restreint à certains contenus, ou encore difficultés à évaluer précisément les acquis. Ces constats soulignent l'importance de concevoir une solution qui allie accessibilité, flexibilité, suivi personnalisé et richesse pédagogique.

1.5 Solution proposée

Pour relever ces défis, l'introduction d'une plateforme éducative en ligne représente une opportunité stratégique. Grâce à des outils interactifs et à l'intelligence artificielle, elle permettra d'améliorer l'expérience d'apprentissage, d'automatiser certaines tâches administratives et de faciliter la gestion des contenus pédagogiques.

Les fonctionnalités clés comprennent :

- Développement d'une application mobile pour les étudiants et enseignants :
 - Accès aux cours et aux ressources pédagogiques (vidéos, PDF, quiz, etc.).
 - Suivi de la progression des apprenants avec recommandations personnalisées.
 - Réception de notifications sur les nouveaux cours, événements et mises à jour.
 - Communication via un chat intégré entre enseignants et étudiants.
- Développement d'une application web pour l'administrateur et les enseignants :
 - Gestion des inscriptions des utilisateurs.
 - Ajout, modification et suppression de cours et contenus pédagogiques.
 - Suivi des paiements et gestion des accès aux cours payants.
 - Gestion des évaluations et attribution des certifications.
 - Analyse des statistiques d'apprentissage via un tableau de bord interactif.
 - Gestion des interactions (questions, forums, assistance en ligne).

Cette solution vise à offrir une expérience utilisateur fluide et optimisée, en intégrant les meilleures pratiques en matière d'e-learning et de gestion de contenu éducatif.

1.6 Cadre du travail adopté

Dans notre projet nous avons choisi d'utiliser Jira Kanban comme outil de gestion de projet pour suivre les tâches et les tickets de manière centralisée. Kanban nous permet de visualiser l'état de chaque tâche, de suivre leur progression et de gérer efficacement les priorités. Nous avons également intégré GitHub, une plateforme de développement collaboratif, pour la gestion du code source et des versions du logiciel. Cette intégration nous permet de suivre les modifications apportées au code en temps réel et de gérer efficacement les branches et les pull requests. Nous organisons des réunions régulières avec l'équipe pour passer en revue l'état du projet, discuter des problèmes potentiels et prendre des décisions en collaboration. Nous sommes convaincus que l'utilisation de Jira Kanban et de GitHub nous permettra de livrer le projet dans les délais impartis tout en maintenant un haut niveau de qualité et de collaboration au sein de l'équipe.

1.7 Environnement de travail

1.7.1 Environnement matériel

L'architecture matérielle utilisée pour la réalisation est la suivante :

Spécifications techniques	Ordinateur portable DELL Latitude 3520	Ordinateur portable HP
Marque et modèle	DELL Latitude 3520	HP
Mémoire	16 Go	8 Go
Stockage	SSD 256 Go	HDD 1 To
Écran	Écran interne	Écran interne
Système d'exploitation	Windows 10 Professionnel	Windows 10 Famille

TABLE 1.1 – Caractéristiques du PC portable utilisé

1.7.2 Méthode de gestion de projet

La gestion de projet regroupe l'ensemble des activités visant à garantir la planification et la réalisation satisfaisante du projet dans les délais fixés, tout en respectant les objectifs établis. Avant d'entamer le projet, il est crucial de choisir une méthode adaptée pour assurer la réussite de sa gestion.

1.7.3 Outils et Technologies

Dans cette section, nous décrivons les outils de travail logiciels, plateformes et les langages de programmation utilisés durant les différentes phases du projet.

Environnements de développement

- **VSCode** Visual Studio Code est un éditeur de code source développé par Microsoft. il est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur

de code multiplateforme, open source et gratuit, supportant une dizaine de langages.

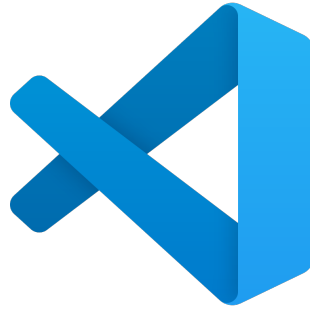


FIGURE 1.6 – logo visual studio code

Langages de programmation

- **java Script** interprété principalement utilisé pour créer des applications web interactives. Il est généralement adopté pour créer du contenu dynamique et ajouter des fonctionnalités telles que les effets d'animation, la validation de formulaire, la mise à jour du contenu de la page sans rechargement etc.



FIGURE 1.7 – Logo javascript

- **CSS** est l'acronyme de « Cascading Style Sheets ». c'est un langage de style pour les pages web . Il offre des fonctionnalités évoluées de conception, de mise en page et de stylisme pour créer des expériences utilisateur cohérentes et adorables sur le World Wide Web.



FIGURE 1.8 – Logo css

- **HTML** est un acronyme pour "HyperText Markup Language" . C'est le langage de balisage standard utilisé pour la création de pages web .Il est utilisé pour créer la structure de base d'une page web de manière efficace .



FIGURE 1.9 – Logo html

- **React** est une bibliothèque JavaScript utilisée pour construire des interfaces utilisateurs (UI). Elle permet de créer des composants réutilisables et imbriquables, qui peuvent être combinés pour former des applications web complexes. React se concentre sur la gestion des vues, ce qui permet de créer des interfaces dynamiques et réactives aux interactions de l'utilisateur.

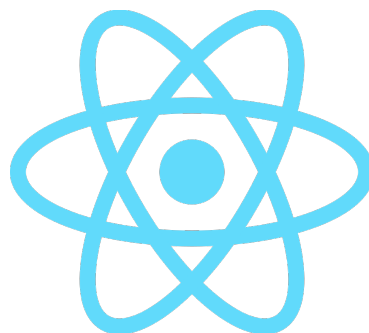


FIGURE 1.10 – Logo react

- **Node** est une plateforme de développement JavaScript qui permet d'exécuter du code JavaScript en dehors du navigateur, notamment sur des serveurs. Elle utilise le moteur V8 de Google Chrome et se caractérise par son architecture non bloquante et sa boucle d'événements qui permettent un traitement asynchrone efficace.



FIGURE 1.11 – Logo Node

- **overleaf** est un système de composition de documents nous permet de structurer notre contenu de manière précise, de gérer facilement les références bibliographiques et de générer des tableaux et des graphiques de haute qualité pour une présentation cohérente de nos résultats



FIGURE 1.12 – Logo Overleaf

- **Bootstrap** est un framework CSS open source, largement utilisé pour le développement web, qui facilite la création de sites web et d'applications en utilisant des classes pré-définies pour le style et la mise en page



FIGURE 1.13 – Logo bootstrap

- **MongoDB** est une base de données NoSQL orientée document, open source, qui stocke les données dans des documents flexibles de type JSON. Elle est conçue pour gérer des données structurées, semi-structurées et non structurées, offrant une grande flexibilité dans la gestion des schémas.



FIGURE 1.14 – Logo MongoDB

- **Express******* est un framework Node.js minimaliste et flexible pour le développement d'applications web et d'API. Il est conçu pour faciliter la construction de serveurs web et l'intégration de fonctionnalités essentielles à l'application, tout en laissant une grande flexibilité pour l'utilisateur. Express permet de gérer les requêtes HTTP



FIGURE 1.15 – Logo Express

Logiciels

- **StarUML** est un logiciel de modélisation UML (Unified Modeling Language) open source développé par MKLabs. Il permet de créer divers diagrammes UML, tels que les diagrammes de cas d'utilisation, les diagrammes de classes, et les diagrammes de séquence, pour modéliser des systèmes logiciels.



FIGURE 1.16 – Logo StarUML

- **Draw.io** est un outil web open source gratuit qui permet de créer divers types de diagrammes, notamment des organigrammes, des diagrammes UML, des relations d'entité, des diagrammes de réseau et des maquettes



FIGURE 1.17 – Logo Draw.io

- **GitHub** Nous avons utilisé GitHub pour partager et gérer notre code source. C'est une plateforme de développement collaboratif qui utilise Git, offrant des fonctionnalités de collaboration et de suivi des modifications pour faciliter l'amélioration continue entre les développeurs.



FIGURE 1.18 – Logo GitHub

- **Figma** est un outil de conception d'interface utilisateur en ligne que nous avons utilisé pour concevoir notre application avant le développement.



FIGURE 1.19 – Logo Figma

1.7.4 Approches de gestion de projet : Agile vs Traditionnelle

Les approches traditionnelles et agiles sont largement utilisées dans le développement de solutions logicielles. L'approche Agile met l'accent sur la collaboration, l'adaptabilité et l'itération. Les équipes travaillent en cycles courts, appelés *sprints*, pour se concentrer sur des objectifs spécifiques et livrer fréquemment des produits fonctionnels.

À l'inverse, l'approche traditionnelle suit un processus linéaire et séquentiel, avec une planification détaillée dès le début. Les phases du projet sont exécutées dans un ordre prédéfini,

de l'analyse des besoins à la livraison finale. Cette méthode est moins flexible mais permet une meilleure planification initiale.

Chaque approche a ses avantages et est adaptée à différents types de projets. Nous sélectionnerons celle qui correspond le mieux aux spécificités de notre projet et à nos exigences.

1.7.5 Choix méthodologique

Le choix de la méthodologie conditionne la réussite du projet. Nous avons comparé plusieurs méthodes courantes : cycle en V, Waterfall et Scrum.

- **Cycle en V** : englobe toutes les étapes du cycle de développement. Avantages : simplicité, amélioration de la qualité, productivité accrue. Inconvénients : manque de souplesse et de communication entre parties prenantes.
- **Waterfall** : modèle en cascade avec étapes strictes. Avantages : structure claire, définition précoce des objectifs, communication efficace. Inconvénients : difficile de gérer les changements et peu adapté aux besoins évolutifs.
- **Scrum** : approche agile basée sur l'itération, la collaboration et l'adaptation. Avantages : simplicité des processus, adaptation aux changements, communication améliorée. Inconvénients : besoin d'une équipe expérimentée, risques de déviation sans définition claire des tâches.

Après analyse, la méthode Scrum sera privilégiée pour sa flexibilité et son adéquation avec les besoins évolutifs du projet.

Critère	Cascade (Waterfall)	Cycle en V	Méthode Agile
Changement	Très difficile, coûteux et impacte tout le cycle.	Difficile, les changements nécessitent une refonte importante.	Facile, intégré dans le processus via des itérations courtes.
Risques	Réduction des risques grâce à la validation continue, mais rigide.	Risque élevé si des erreurs ne sont détectées qu'à la fin du projet.	Risque plus faible grâce aux retours continus et aux ajustements rapides.

TABLE 1.2 – Comparaison entre les différentes méthodes de gestion de projet

1.7.6 Présentation de la méthodologie Scrum

La méthode **Scrum** repose sur un développement logiciel **incrémental**, avec une transparence totale sur les demandes d'évolutions ou de corrections à implémenter. Grâce à des livraisons fréquentes, en général toutes les quatre semaines, le client reçoit à chaque itération une version du logiciel disposant de nouvelles fonctionnalités, en parfait état de fonctionnement.

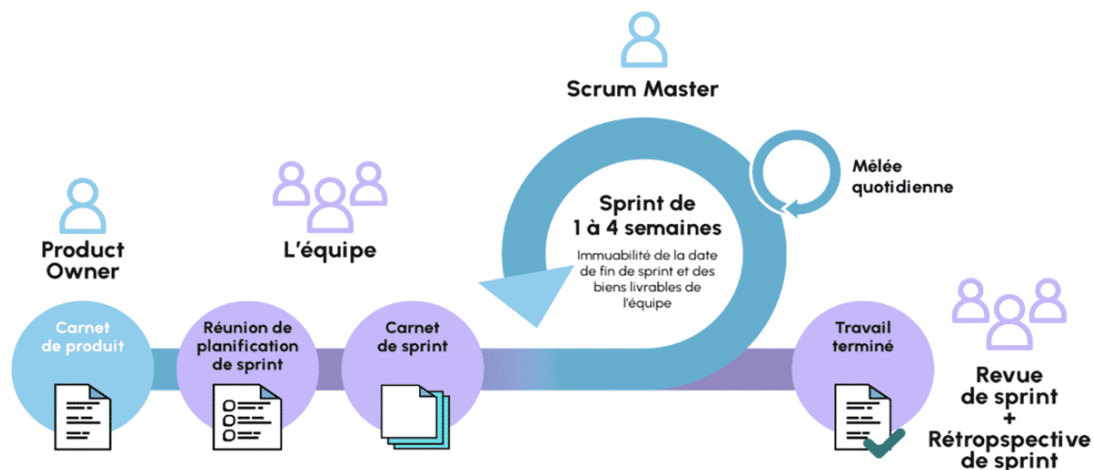


FIGURE 1.20 – Gestion de projet SCRUM

1.7.7 Outils SCRUM

Dans le cadre de notre projet, et afin d'assurer le bon déroulement des différentes phases de ce dernier, nous avons opté pour le framework agile **Scrum**. Il repose sur une gestion de projet collaborative et un cycle de développement :

- **Itératif** : répété plusieurs fois, de l'idée initiale à une version de plus en plus aboutie ;
- **Incrémental** : progressif, tâche après tâche ;
- **Adaptatif**.

Ainsi, les trois piliers fondamentaux de Scrum sont :

- **La transparence** : tous les membres de l'équipe ont accès aux informations pertinentes concernant le produit à développer. Cela favorise la communication et une compréhension commune des objectifs.
- **L'inspection régulière** : des évaluations régulières permettent de suivre l'avancement du projet, de mesurer les résultats obtenus et de détecter d'éventuels écarts.
- **L'adaptation** : en cas d'écart identifié, l'équipe peut ajuster sa stratégie pour corriger la trajectoire du projet. Cette flexibilité permet une amélioration continue du processus.

La conception et le développement de notre système s'inscrivent naturellement dans une logique Scrum, pour des raisons bien déterminées. En effet, ce processus s'adapte parfaitement à la décomposition de notre projet de fin d'études et repose sur les avantages suivants :

- Une plus grande souplesse et réactivité ;
- Une capacité d'adaptation importante au changement grâce à des itérations courtes ;
- Une meilleure satisfaction des besoins du client.

1.7.8 Planning

Afin de garantir une bonne compréhension de l'avancement du projet et une communication efficace, nous avons utilisé un **diagramme de Gantt** qui illustre les tâches à effectuer ainsi que leurs dates de réalisation prévues.

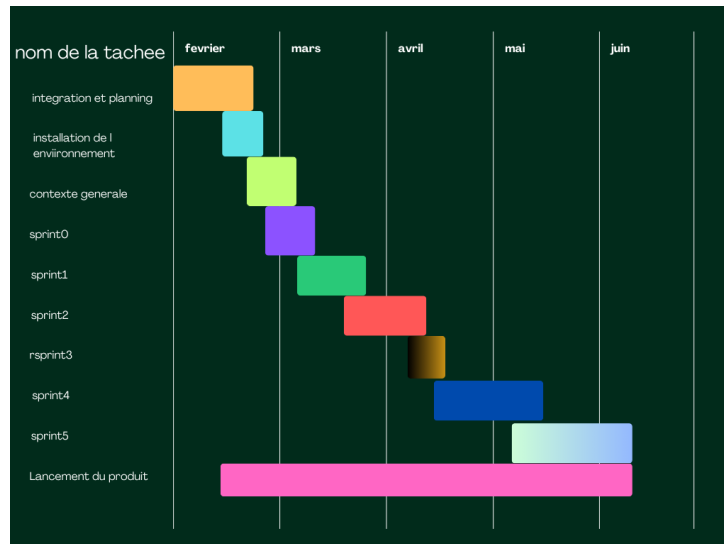


FIGURE 1.21 – diagramme de gant

1.8 Conclusion

Ce chapitre constitue une étape essentielle dans la structuration de notre projet. Nous avons présenté le contexte de l'entreprise d'accueil, analysé l'existant, et introduit la méthodologie Scrum adoptée pour la réalisation. Cette approche nous a permis d'avoir une vue globale du projet, en mettant en évidence les besoins et les enjeux que nous approfondirons dans les chapitres suivants.

Chapitre 2

Analyse et spécification des besoins

2.1 Introduction

Ce chapitre est consacré à la spécification des besoins fonctionnels et non fonctionnels de notre système, au diagramme de cas d'utilisation globale, au backlog du produit avec une description de l'environnement de travail.

2.2 Analyse des besoins

La partie spécification des besoins est une étape primordiale afin d'identifier les besoins fonctionnels de notre application et d'aboutir à notre objectif, d'une part, et, d'autre part, pour éviter le développement d'une interface non satisfaisante.

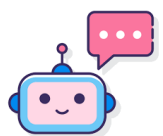
2.2.1 Identification des acteurs



Étudiant : L'étudiant peut créer un compte sur la plateforme afin de s'inscrire. Il a la possibilité de rechercher et d'acheter des cours, puis de les télécharger. Il peut accéder aux ressources associées aux cours achetés, contacter les enseignants pour poser des questions ou demander des clarifications, et enfin évaluer les cours suivis.



Enseignant : L'enseignant peut s'inscrire en créant un compte. Il peut ensuite modifier les cours en ajoutant, supprimant ou mettant à jour le contenu pédagogique. Il a aussi la possibilité de créer des évaluations (tests, quiz) et d'interagir avec les étudiants pour répondre à leurs questions.



Chatbot : Le chatbot permet aux étudiants de poser des questions à l'intelligence artificielle intégrée à la plateforme. Il fournit des réponses automatiques et guide les utilisateurs dans leur navigation et l'utilisation des différentes fonctionnalités disponibles.



Administrateur : L'administrateur est responsable de la gestion globale de la plateforme. Il peut gérer les comptes utilisateurs, organiser le contenu pédagogique, surveiller les paiements effectués par les étudiants et générer des rapports statistiques et analytiques.

2.2.2 Besoins fonctionnels

1. Gestion des utilisateurs

- Permettre l'inscription et la connexion des utilisateurs (étudiants et enseignants).
- Authentification via e-mail/mot de passe et éventuellement via Google ou Facebook.
- Gérer les profils des utilisateurs (photo, biographie, informations personnelles).

2. Gestion des cours

- Permettre aux enseignants de créer, modifier et supprimer des cours.
- Ajouter différents types de contenus : vidéos, fichiers PDF, documents, exercices interactifs.
- Organiser les cours en catégories et sous-catégories.

3. Accès aux cours

- Permettre aux étudiants d'acheter et d'accéder aux cours après paiement.
- Afficher les leçons sous forme de modules progressifs.

4. Système de paiement

- Intégrer un système de paiement en ligne sécurisé (PayPal, Stripe, carte bancaire, etc.).
- Gérer les transactions et générer les factures.

5. Interaction et communication

- Fournir un système de messagerie entre étudiants et enseignants.
- Mettre en place des espaces de discussion (forums ou commentaires liés aux cours).
- Permettre aux utilisateurs de soumettre des réclamations.
- Ajouter un formulaire de contact pour envoyer des messages à l'équipe de support.

6. Évaluations et suivi

- Proposer des quiz et examens pour évaluer les connaissances des étudiants.
- Générer automatiquement un certificat après la validation complète d'un cours.
- Suivre la progression des étudiants dans les cours suivis.

7. Système d'intelligence artificielle

- Intégrer un chatbot capable de répondre aux questions des étudiants.
- Fournir un correcteur automatique pour les exercices pratiques.

8. Notation et feedback

- Permettre aux étudiants de noter et commenter les cours suivis.
- Mettre en place un système de recommandation basé sur les préférences et les évaluations des utilisateurs.

9. Rôle assistant

- Permettre aux utilisateurs de postuler pour devenir assistant ou tuteur pédagogique.

10. Visualisation des données

- Fournir à l'administrateur une interface de visualisation des données (utilisation, ventes, progression...).

11. Fonctionnalité complémentaire à définir

- Prévoir une fonctionnalité supplémentaire à spécifier selon les besoins futurs (3 jours estimés).

2.2.3 Besoins Non fonctionnelle

1. Performance et rapidité

- La plateforme doit rester rapide et fluide, même en cas de forte affluence d'utilisateurs.
- Optimisation du temps de chargement des pages et des vidéos.

2. Sécurité

- Chiffrement des mots de passe et des données sensibles.
- Protection contre les attaques XSS, CSRF, injections SQL, etc.
- Sécurisation des transactions financières.

3. Disponibilité et accessibilité

- Plateforme accessible 24h/24 et 7j/7, sans interruption de service.
- Compatibilité avec les navigateurs web modernes et les appareils mobiles.

4. Scalabilité

- Capacité à gérer un grand nombre d'utilisateurs simultanément.
- Possibilité d'ajouter de nouvelles fonctionnalités sans dégradation des performances.

5. Expérience utilisateur (UX/UI)

- Interface intuitive et facile à utiliser.
- Design responsive, adapté aux mobiles, tablettes et ordinateurs.

6. Sauvegarde et récupération des données

- Sauvegarde automatique et régulière des données des utilisateurs et des cours.
- Système de récupération de compte en cas d'oubli du mot de passe.

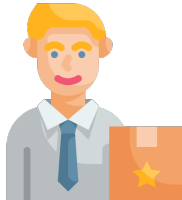
7. Conformité aux normes

- Respect du Règlement Général sur la Protection des Données (RGPD).
- Respect des normes d'accessibilité pour les personnes en situation de handicap.

2.3 Pilotage du projet avec SCRUM

2.3.1 Equipe et rôles

Les rôles d'une équipe Scrum : L'équipe dans notre projet se compose de différents rôles, chacun contribuant de manière essentielle à sa réussite :



Product Owner



Scrum Master



Developer Team

FIGURE 2.1 – Équipe SCRUM.

2.3.2 L'équipe du projet

La répartition des rôles au sein de notre projet Scrum est présentée dans le tableau ci-dessous :

Équipe	Rôle
Le Product Owner ()	Représentant les parties prenantes, joue un rôle crucial dans notre projet de chaîne d'approvisionnement chez Leanios. Il définit et hiérarchise les exigences, gère le backlog produit, communique la vision et collabore avec les parties prenantes pour assurer le succès du projet.
Le Scrum Master ()	Responsable de faciliter la mise en œuvre du cadre Scrum. Il veille à ce que l'équipe respecte les pratiques Scrum, élimine les obstacles et favorise un environnement de travail collaboratif et autonome.
Developer team ()	Membres de l'équipe chargés de concevoir, coder, tester et livrer les fonctionnalités du produit. Ils travaillent en collaboration étroite avec le Product Owner pour comprendre les besoins, développer les fonctionnalités et assurer la qualité du produit final.

TABLE 2.1 – Présentation de l'équipe Scrum.

2.4 Backlog de produit

Le Backlog du produit est l'artefact le plus important de Scrum. En effet, c'est l'ensemble des caractéristiques fonctionnelles qui compose le produit. Ces caractéristiques sont appelées user story. Chaque user story est caractérisée par une priorité définie par le Product Owner. Le backlog produit de notre projet est le suivant :

ID	Épique	User Story	Complexité (1–5)	Priorité
US-1	Gestion des utilisateurs	En tant qu'utilisateur, je peux m'inscrire, me connecter et gérer mon profil afin d'accéder à la plateforme selon mon rôle.	3	Élevée
US-2	Gestion des cours	En tant qu'enseignant, je peux créer, modifier et supprimer des cours afin de proposer du contenu pédagogique aux étudiants.	3	Élevée
US-3	Accès aux cours	En tant qu'étudiant, je peux acheter et accéder aux cours afin de suivre une formation complète.	3	Élevée
US-4	Système de paiement	En tant qu'étudiant, je peux effectuer un paiement en ligne afin d'accéder aux cours.	4	Élevée
US-5	Interaction et communication	En tant qu'utilisateur, je peux interagir via une messagerie ou des discussions pour échanger avec les enseignants ou étudiants.	3	Moyenne
US-6	Évaluations et suivi	En tant qu'étudiant, je peux passer des quiz et suivre ma progression afin de valider mes compétences et obtenir un certificat.	3	Moyenne
US-7	Système d'intelligence artificielle	En tant qu'étudiant, je peux poser des questions au chatbot et recevoir de l'aide automatisée afin de progresser plus facilement.	4	Moyenne
US-8	Notation et feedback	En tant qu'étudiant, je peux noter et commenter les cours suivis afin d'aider les autres utilisateurs et améliorer le contenu.	2	Moyenne
US-9	Devenir assistant	En tant qu'utilisateur, je peux postuler pour devenir assistant afin de contribuer à l'encadrement pédagogique.	2	Faible
US-10	Visualisation des données	En tant qu'administrateur, je peux consulter des tableaux de bord pour visualiser l'activité de la plateforme.	4	Moyenne

ID	Épique	User Story	Complexité (1–5)	Priorité
US-11	Réclamations	En tant qu'utilisateur, je peux envoyer une réclamation afin de signaler un problème rencontré.	1	Faible
US-12	Message de contact	En tant qu'utilisateur, je peux envoyer un message via le formulaire de contact afin d'entrer en communication avec l'équipe.	1	Faible
US-13	Fonctionnalité complémentaire à définir	En tant qu'administrateur, je souhaite pouvoir ajouter une fonctionnalité personnalisée à définir selon les besoins futurs.	3	Faible

TABLE 2.2 – Backlog produit des fonctionnalités principales de la plateforme.

2.5 Diagramme de cas d'utilisation global

Tout d'abord, identifions les acteurs qui utiliseront le système :

2.5.1 Diagramme de cas d'utilisation global

CHANGEMENT!!!!

- **Étudiant** : L'étudiant est l'utilisateur principal qui s'inscrit, consulte et achète des cours, suit les modules, passe des évaluations, interagit avec les enseignants et utilise le chatbot pour poser des questions.
- **Enseignant** : L'enseignant conçoit, publie et met à jour les cours et les évaluations. Il communique avec les étudiants pour répondre à leurs questions et les accompagner dans leur apprentissage.
- **Administrateur** : L'administrateur supervise la plateforme. Il gère les utilisateurs, contrôle les contenus publiés, suit les paiements, et génère des rapports pour l'analyse globale du système.
- **Chatbot (IA)** : Le chatbot assiste les étudiants en répondant automatiquement à leurs questions, en les guidant dans l'utilisation de la plateforme et en corrigeant certains exercices pratiques.

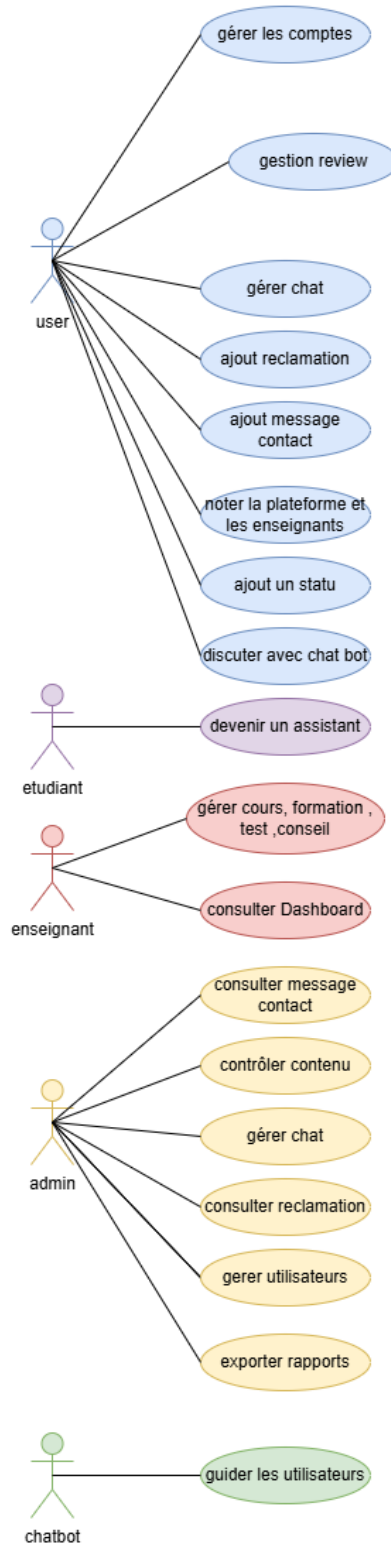


FIGURE 2.2 – Diagramme de cas d'utilisation global

Sur la base de ces acteurs, nous pouvons identifier les cas d'utilisation suivants :

- **User (Utilisateur général)** : Ce rôle regroupe les fonctionnalités communes à tous les types d'utilisateurs (étudiant, enseignant, visiteur). Il peut :

- Gérer les comptes
- Gérer le chat
- Ajouter une réclamation
- Envoyer un message de contact
- Noter la plateforme et les enseignants
- Ajouter un statut
- Discuter avec le chatbot
- Consulter les avis (reviews)
- **Étudiant** : Hérite des droits du rôle “user” avec la capacité supplémentaire de :
 - Devenir un assistant
- **Enseignant** : Dispose des fonctionnalités suivantes :
 - Gérer les cours, formations, tests et conseils
 - Consulter un tableau de bord (Dashboard)
- **Administrateur (Admin)** : Possède des droits avancés pour superviser la plateforme :
 - Consulter les messages de contact
 - Gérer les réclamations
 - Gérer les utilisateurs
 - Contrôler le contenu
 - Gérer le chat
 - Exporter des rapports
- **Chatbot** : Agent conversationnel chargé de :
 - Guider les utilisateurs dans l'utilisation de la plateforme

Ce diagramme joue un rôle fondamental dans la conception fonctionnelle de l'application. Il aide à structurer les responsabilités des utilisateurs et constitue une référence pour le développement orienté cas d'utilisation.

2.6 Architecture de l'application

L'architecture générale de ce projet est optimisée pour garantir la **fiabilité**, la **sécurité** et l'**efficacité**, en utilisant des technologies et des pratiques de conception éprouvées.

La plateforme e-learning repose sur un modèle **client-serveur** distribué via Internet, permettant une interaction fluide entre les différents utilisateurs : **administrateur**, **enseignants** et **étudiants**. Cette architecture est développée en utilisant la **MERN Stack**, une pile technologique moderne et complète composée de **MongoDB**, **Express.js**, **React.js** et **Node.js**.

2.6.1 Communication client-serveur

Chaque utilisateur accède à l'application web via un navigateur, en envoyant des **requêtes HTTP** vers le **serveur principal** hébergé sur Internet. Le serveur traite ces requêtes, communique avec la base de données, puis retourne des **réponses HTTP** contenant les données ou les pages demandées.

Les rôles des utilisateurs sont définis comme suit :

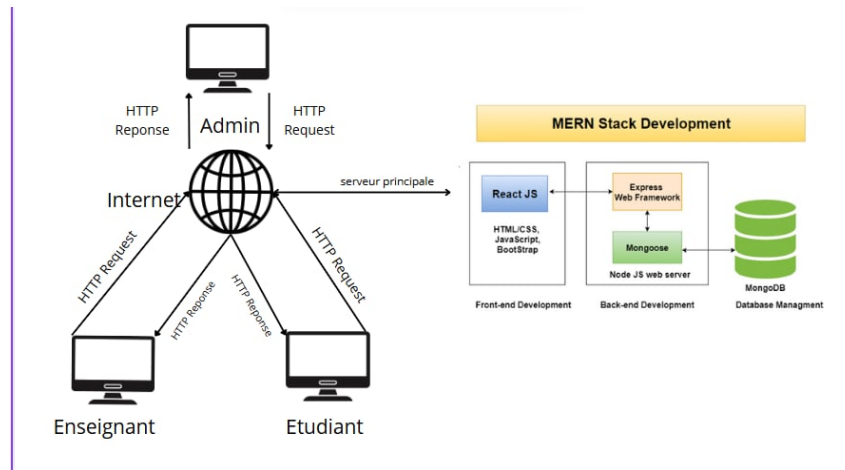


FIGURE 2.3 – Architecture Physique

- **Administrateur** : gère les utilisateurs (étudiants et enseignants), les cours, et le contenu pédagogique.
- **Enseignant** : publie les cours, suit la progression des étudiants, et évalue leurs rendus.
- **Étudiant** : consulte les cours, participe aux activités pédagogiques, et interagit avec les enseignants.

2.6.2 Architecture logicielle :

L'architecture logicielle repose sur la **MERN Stack**, qui permet de couvrir l'ensemble des besoins de développement front-end, back-end et de gestion de base de données.

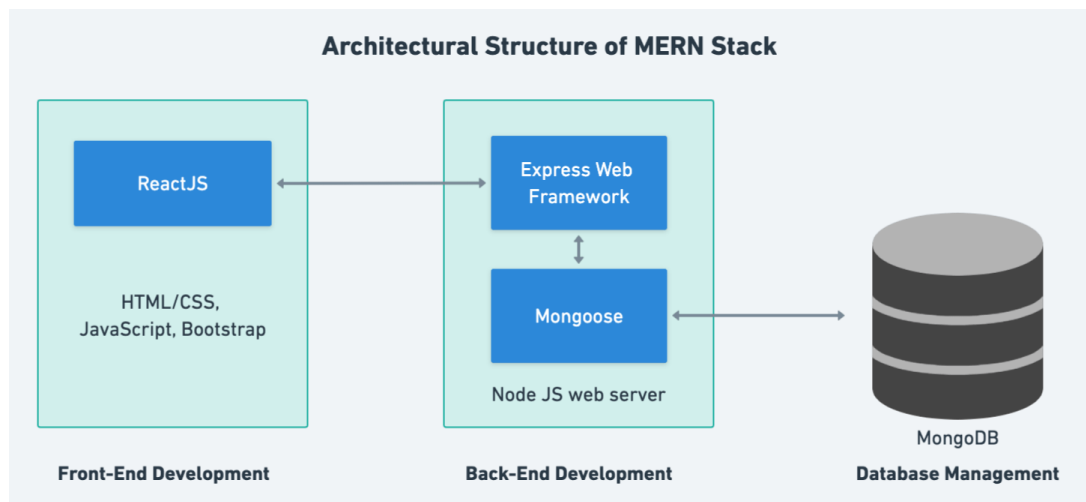


FIGURE 2.4 – Architecture logicielle

— Front-end (React.js)

L'interface utilisateur est développée en *React.js*, une bibliothèque JavaScript performante pour la création de composants dynamiques. L'interface est enrichie à l'aide de *HTML5*, *CSS3*,

Bootstrap et *JavaScript*, offrant une expérience utilisateur réactive et intuitive.

— **Back-end (Express.js & Node.js)**

Le serveur d'application est construit avec *Node.js*, une plateforme JavaScript côté serveur, couplée avec le framework *Express.js*. Ce dernier permet de gérer les routes, les API REST, l'authentification, ainsi que la logique métier de la plateforme.

— **Base de données (MongoDB & Mongoose)**

Les données (utilisateurs, cours, évaluations, etc.) sont stockées dans *MongoDB*, une base de données NoSQL orientée documents. L'outil *Mongoose* est utilisé pour modéliser les schémas de données et faciliter les opérations entre le serveur *Node.js* et la base *MongoDB*.

2.7 Planification des sprints

Chapitre 3

SPRINT 1 :

3.1 Introduction

Dans ce chapitre nous allons détailler le travail réalisé durant le premier sprint. En effet, chaque sprint, qui est l'ensemble de items, représente une vision distribuée de la période de la production du livrable. Le développement de chaque item passe par les étapes d'analyse, de conception et de réalisation.

3.2 Backlog du sprint 1

TABLE 3.1 – Sprint 1

ID	Fonctionnalité globale	Tâche	Estimation (points)
US-1	Gestion des utilisateurs	Inscription, connexion, gestion profil	3
US-2	Gestion des cours	Création, modification, suppression des cours	3

3.3 Spécification fonctionnelle

3.3.1 Diagramme de cas d'utilisation du sprint 1



FIGURE 3.1 – diagramme de cas d'utilisation du sprint1

3.3.2 diagramme de cas d'utilisation du sprint 1

Gestion des utilisateurs

Description textuelle du cas d'utilisation Gestion des utilisateurs

Cas d'utilisation	Permettre à un utilisateur de s'inscrire, se connecter et gérer son profil selon son rôle (étudiant, enseignant, admin).
Acteurs	Utilisateur (nouveau ou existant).
Pré-condition	L'utilisateur doit avoir accès à la page d'inscription ou connexion.
Post-condition	L'utilisateur est authentifié et peut accéder aux fonctionnalités selon son rôle.
Scénario nominal	<ul style="list-style-type: none"> — L'utilisateur ouvre la page d'inscription. — Il remplit le formulaire avec ses informations (nom, email, mot de passe, rôle). — Il soumet le formulaire. — Le système valide les données. — Le compte est créé et un email de confirmation est envoyé. — L'utilisateur se connecte avec son email et mot de passe. — Le système authentifie l'utilisateur. — L'utilisateur accède à son tableau de bord personnalisé selon son rôle.
Scénario alternatif	<ul style="list-style-type: none"> — Si l'email est déjà utilisé, le système affiche un message d'erreur "Email déjà pris". — Si les mots de passe ne correspondent pas, affichage d'un message "Mot de passe non confirmé". — Si les champs obligatoires sont vides, le formulaire indique les erreurs. — En cas d'erreur de connexion, un message "Email ou mot de passe incorrect" est affiché. — L'utilisateur peut tenter à nouveau la connexion ou réinitialiser son mot de passe.

Gestion des cours

Description textuelle du cas d'utilisation Gestion des cours

Cas d'utilisation	L'enseignant peut créer, modifier et supprimer des cours pour proposer du contenu pédagogique aux étudiants.
Acteurs	Enseignant connecté.
Pré-condition	L'enseignant est authentifié et possède les droits pour gérer les cours.
Post-condition	Le cours est créé, mis à jour ou supprimé avec succès.
Scénario nominal	<ul style="list-style-type: none"> — L'enseignant accède à la gestion des cours. — Il clique sur "Créer un cours". — Il remplit les champs obligatoires (titre, description, contenu). — Il soumet le formulaire. — Le système valide les données et enregistre le cours. — Le cours apparaît dans la liste des cours disponibles. — Pour modifier, il sélectionne un cours, effectue les changements, puis enregistre. — Pour supprimer, il sélectionne un cours et confirme la suppression.
Scénario alternatif	<ul style="list-style-type: none"> — Si des champs obligatoires sont manquants, le système affiche un message d'erreur. — Si la suppression d'un cours est bloquée (ex. étudiants inscrits), un message prévient l'enseignant. — En cas de problème technique (base de données inaccessible), afficher un message d'erreur générique.

Chapitre4

SPRINT 2 :

4.1 Introduction

Dans ce chapitre nous allons détailler le travail réalisé durant le deuxième sprint. En effet, chaque sprint, qui est l'ensemble de items, représente une vision distribuée de la période de la production du livrable. Le développement de chaque item passe par les étapes d'analyse, de conception et de réalisation.

4.2 Backlog du sprint 2

ID	Fonctionnalité globale	Tâche	Description
US-3	Achat et accès	Achat et accès aux cours	<p>Cas d'utilisation : L'étudiant achète un cours et y accède.</p> <p>Acteurs : Étudiant.</p> <p>Pré-condition : Étudiant connecté, cours disponible.</p> <p>Post-condition : Accès au cours débloqué.</p> <p>Scénario nominal : L'étudiant choisit un cours, paie, puis suit le cours.</p> <p>Scénario alternatif : Paiement refusé ou cours indisponible, message d'erreur.</p>
US-4	Paiement en ligne	Intégration du paiement en ligne	<p>Cas d'utilisation : Permettre un paiement sécurisé pour accéder aux cours.</p> <p>Acteurs : Étudiant, système de paiement.</p> <p>Pré-condition : Étudiant prêt à payer, méthode valide.</p> <p>Post-condition : Paiement validé ou refusé, accès accordé ou bloqué.</p> <p>Scénario nominal : Étudiant saisit ses infos, paiement validé, accès activé.</p> <p>Scénario alternatif : Paiement échoué, proposition de réessayer.</p>

4.2.1 Diagramme de séquence du sprint 1

Chapitre 5

SPRINT 3 :

5.1 Introduction

Dans ce chapitre nous allons détailler le travail réalisé durant le troisième sprint. En effet, chaque sprint, qui est l'ensemble de items, représente une vision distribuée de la période de la production du livrable. Le développement de chaque item passe par les étapes d'analyse, de conception et de réalisation.

5.2 Backlog du sprint 3

5.3 *****

Cas d'utilisation	L'étudiant peut acheter et accéder aux cours pour suivre une formation complète.
Acteurs	Étudiant connecté.
Pré-condition	L'étudiant doit être authentifié.
Post-condition	L'étudiant dispose d'un accès au cours acheté.
Scénario nominal	<ul style="list-style-type: none"> — L'étudiant navigue dans la liste des cours. — Il sélectionne un cours à acheter. — Il est redirigé vers la page de paiement. — Il saisit ses informations de paiement. — Le paiement est validé par le système. — L'étudiant reçoit une confirmation d'achat. — Le cours devient accessible depuis son tableau de bord.
Scénario alternatif	<ul style="list-style-type: none"> — En cas de paiement refusé, un message d'erreur s'affiche. — Si les informations de paiement sont invalides, le système demande la correction. — Si le cours est déjà acheté, afficher un message d'information.

Chapitre 6

SPRINT 4 :

6.1 Introduction

Dans ce chapitre nous allons détailler le travail réalisé durant le quatrième sprint. En effet, chaque sprint, qui est l'ensemble de items, représente une vision distribuée de la période de la production du livrable. Le développement de chaque item passe par les étapes d'analyse, de conception et de réalisation.

6.2 Backlog du sprint 4

TABLE 6.1 – Sprint 4

ID	Fonctionnalité globale	Tâche	Estimation (points)
US-9	Devenir assistant	Postuler comme assistant	2
US-10	Visualisation des données	Tableaux de bord administrateur	4
US-11	Réclamations	Envoi et traitement de réclamations	1
US-12	Message de contact	Formulaire de contact	1
US-13	Fonctionnalité complémentaire	Fonctionnalité personnalisée à définir	3