

inputenc[]

**Université Sultan Moulay Slimane
Ecole Nationale des Sciences Appliquées
-Khouribga-**

Projet de fin d'étude
En vue de l'obtention du diplôme

INGENIEUR D'ETAT

Filière : Génie Informatique

Présenté par
Bahaa Eddine ELBAGHAZAOUI

Contribution à la réalisation de la plateforme Prospektor

Soutenu le 01/07/2019, devant le jury :

Meriem MANDAR	ENSA-K	Président
Mohammed NASRI	ENSA-K	Examineur
Mohamed AMNAI	ENSA-K	Encadrant Interne
Abdelhaq EL AIBI	Groupe OCP	Encadrant Externe

Résumé

Dans le cadre de ce Projet de Fin d'études, nous étions responsables de la conception des différentes fonctionnalités d'une solution moderne Prospektor. Cette solution vient dans un contexte de digitalisation du processus de gestion des anomalies dans les mines du groupe OCP lors de l'extraction de phosphates en se basant sur une architecture orientée API. Notre contribution dans ce projet a commencé par une analyse de l'existant pour préciser le point de départ, ensuite nous avons commencé à implémenter les fonctionnalités par Epics dans une approche Agile.

Ce stage de fin d'étude nous a été très bénéfique, il nous a aidé à projeter l'ensemble des connaissances acquises durant notre formation d'ingénieur à l'ENSA de Khouribga, ainsi que l'implication dans les différentes phases liées à un projet de développement logiciel. Une forte valeur ajoutée réside dans l'apprentissage des nouveaux concepts qui sont actuellement la tendance du domaine du génie logiciel à savoir l'utilisation d'une architecture orientée APIs basé sur les web service qui tient en compte le contexte actuel de la technologie ainsi que l'adoption d'une méthodologie agile qui nous a permis une marge de flexibilité pour innover et livrer plus de valeur. Encore plus, cette expérience de stage nous a aidé à intégrer le monde de l'entreprise, l'interaction et le travail au sein d'une équipe qui ont rendu ce projet une expérience professionnelle très réussie.

La solution Prospektor n'est pas encore terminée en terme de développement vu la voluminosité de l'ensemble des fonctionnalités. Il reste encore à développer des fonctionnalités en relation avec le reste des Epics de la RoadMap ainsi que les tâches liées aux analytiques.

Mots clés : Prospektor, Orienté API, Approche Agile.

Abstract

As part of the graduation project, we were responsible for designing the various functionalities of a modern prospektor solution. This solution is part of the digitalization of the OCP's anomaly management process, it's based on an API oriented architecture. Our contribution in this project started with an analysis of the existing tools to specify the departure point, then we started to implement the functionalities by Epics in an Agile approach.

This graduation internship was very beneficial, it helped us to project all the knowledge acquired during our engineering studies at ENSA Khouribga, as well as the involvement in the different steps related to a software development project. A strong added value lies in learning new concepts that are currently the trend of the software engineering field namely the use of an API oriented architecture based on web services that considers the current context of technology, as well as the adoption of an agile methodology that has given us a margin of flexibility to innovate and deliver more value. Even more, this internship experience helped us to integrate the business world, interaction and working within a team that made this project a very successful work experience.

The Prospektor solution is not yet finished in its development due to the volume of all the functionalities. It still remains to develop features related to the rest of the RoadMap Epics as well as the tasks related to analytics.

Keywords : Prospektor, API oriented architecture, Agile methodology.

Dédicaces

C'est avec plaisir que je dédie ce modeste travail

À mon père, décédé trop tôt, qui m'a toujours poussé et motivé dans mes études.
Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde.

À ma mère .Aucun hommage ne pourrait être à la hauteur de l'amour dont elle ne cesse de me combler. Que dieu la procure bonne santé et longue vie.

À ma sœur Azhar et à toute ma famille pour leur amour, leur tendresse et leur encouragement.

Au membres de digital factory, Bakr, Amine, Ayoub, Driss, Karima, Abdelhalim, Meryem, Ghita, Imad et Zaynab pour leur aide et leur serviabilité.

À tous ceux que j'aime.

Bref, à tous ceux qui ont contribué à la réalisation de ce travail. Que chacun y trouve l'expression de ma profonde gratitude.

Remerciement

Je présente mes remerciements les plus sincères à Monsieur **Mohamed AMNAI** d'avoir accepté de m'encadrer, de m'avoir soutenu et assisté tout au long de mon stage et de m'avoir fait part de ses remarques pertinentes et de ses idées constructives. Je tiens à lui exprimer toute mon admiration et ma reconnaissance.

Je tiens à remercier également Monsieur **Abdelhaq EL AIBI** et tous les membres de la digital factory pour leur encadrement, leurs précieux conseils tout au long de mon stage, ainsi que pour les remarques éclairées qu'ils m'ont prodiguées tout au long de ce stage.

Par la même occasion, j'adresse mes remerciements à tous mes enseignants pour leurs efforts qui ont guidé mes pas et enrichi mes travaux tout au long de mes études universitaires, et à ceux qui ont perfectionné mes connaissances théoriques et pratiques durant la période de la formation.

Table des matières

1	Contexte Général Du Projet	15
1.1	Introduction	16
1.2	Présentation de l'organisme d'accueil	16
1.2.1	Description de la société LeaderPos	16
1.2.2	Logo de la société LeaderPos	16
1.2.3	Informations de contact de la société LeaderPos	17
1.2.4	Services de la société LeaderPos	17
1.2.5	Organigramme fonctionnel de de la société LeaderPos	18
1.3	Étude de l'existant	18
1.3.1	Critique de l'existant et solution proposée	22
1.3.2	Solution proposée	25
1.4	Présentation du projet	26
1.5	Langage et méthodologie de conception	26
1.6	Pourquoi Scrum	27
1.7	Choix des technologies	28
1.8	Conclusion	29
2	Planification et Architecture	30
2.1	Introduction	31
2.2	Analyse des besoins	31
2.2.1	Identification des acteurs	31
2.2.2	Les besoins fonctionnels	33
2.2.3	Les besoins non fonctionnels	37
2.3	Planning du traitement de cas d'utilisation	39
2.3.1	Priorités	39

2.3.2	Risques	39
2.4	Prototypage des interfaces	40
2.5	Pilotage du projet avec Scrum	42
2.5.1	Les outils Scrum	42
2.5.2	Équipe et rôles	43
2.5.3	Le backlog du produit	43
2.5.4	Diagramme de cas d'utilisation global	50
2.5.5	Architecture de l'application	52
2.5.6	Planification des sprints	54
2.6	Conclusion	55
3	Etude et réalisation du premier Sprint	56
3.1	Introduction	57
3.2	Le premier sprint	57
3.2.1	Spécification fonctionnelle	67
3.2.2	Diagramme de cas d'utilisation détaillé du Sprint 1	67
3.2.3	Description textuelle de cas d'utilisation	69
3.2.4	La modélisation conceptuelle	81
3.2.5	Codage	86
3.2.6	Réalisation	86
3.2.7	Tests	87
3.2.8	Les tests unitaires	88
3.3	Conclusion	91
4	Etude et réalisation du deuxième Sprint	92
4.1	Introduction	93
4.2	Le deuxième sprint	93
4.2.1	Spécifications fonctionnelles	106
4.2.2	La modélisation conceptuelle	114
4.2.3	Tests	119
4.3	Conclusion :	119
5	La phase closure	121
5.1	Introduction	121
5.2	Environnement de développement	121
5.2.1	Environnement matériel	121
5.2.2	Environnement logiciel	122

5.2.3	Langages de programmation	124
5.3	Conclusion	125

Table des figures

1.1	Logo De La société LeaderPos	16
1.2	Organigramme Fonctionnel De La Société LeaderPos	18
1.3	Page D'Accueil De La Plateforme Udemy	19
1.4	Page D'Accueil De La Plateforme Coursera	19
1.5	Page D'Accueil De La Plateforme w3school	20
1.6	Processus Scrum	28
2.1	Diagramme De Contexte Statique	33
2.2	Page De Connexion	40
2.3	Interface D'Accueil De Visiteur	41
2.4	Interface De Profil Instructeur	42
2.5	Diagramme De cas D'Utilisation Global	51
2.6	Diagramme de paquetage	53
3.1	Diagramme De cas D'Utilisation Du Premier Sprint	68
3.2	Diagramme De Classes Du Premier Sprint	81
3.3	Diagramme De Séquence Du Cas D'Utilisation « S'authentifier »	82
3.4	Diagramme De Séquence Du Cas D'Utilisation « Ajouter Un Membre »	83
3.5	Diagramme De Séquence Du Cas D'Utilisation « Consulter Profil » .	84
3.6	Diagramme De Séquence Du Cas D'Utilisation « Consulter Statis- tiques »	85
3.7	L'interface d'authentification	87
3.8	L'interface d'ajout d'utilisateurs	87
3.9	Résultat réussi du mapping de la requête de sujet vers la réponse de sujet	89
3.10	Test unitaire de création réussie d'un sujet	90

3.11	Validation de test unitaire de création d'un sujet	90
4.1	: Diagramme De Cas D'Utilisation Détaillé du Sprint 2	107
4.2	: Diagramme De Classe Du Second Sprint	114
4.3	:Diagramme De Séquence Du Cas D'Utilisation « Changer Mot De Passe »	115
4.4	: Diagramme De Séquence Du Cas D'Utilisation « Ajouter Un Cours »	116
4.5	:Diagramme De Séquence Du Cas D'Utilisation « Supprimer Une Leçon »	118
4.6	:Test Unitaire Pour Une Liste Vide Des Leçons	119
4.7	:Résultat Réussi Pour Une Liste Vide Des Leçons	119

Liste des tableaux

1.1	Fiche Technique De LeaderPos	17
1.2	Analyse Comparative Des plateformes : Udemmy, Coursera Et W3Schools	20
1.3	Tableau Comparatif : Avantages Et Inconvénients Des Plateformes D'Apprentissage En Ligne	23
2.1	Description Des acteurs De La Plateforme D'Apprentissage	32
2.2	Exigences Fonctionnelles	33
2.3	Backlog de Produit	44
2.4	Backlog de Produit	55
3.1	Backlog Du Premier Sprint	58
3.2	Description textuelle du cas d'utilisation « Authentification »	69
3.3	Description textuelle du cas d'utilisation « Ajouter Utilisateur »	70
3.4	Description textuelle du cas d'utilisation « Supprimer un membre »	72
3.5	Description textuelle du cas d'utilisation « Modifier informations de profil »	73
3.6	Description textuelle du cas d'utilisation « Télécharger photo de profil »	75
3.7	Description Textuelle Du Cas D'Utilisation « Déconnexion »	76
3.8	Description textuelle du cas d'utilisation « Ajouter sujet »	77
3.9	Description textuelle du cas d'utilisation « Modifier sujet »	78
3.10	Description textuelle du cas d'utilisation « Supprimer sujet »	79
4.2	Description textuelle du cas d'utilisation « Modifier mot de passe »	108
4.3	Description textuelle du cas d'utilisation « Ajouter cours »	109
4.4	Description textuelle du cas d'utilisation « Ajouter cours »	111
4.5	Description textuelle du cas d'utilisation « Modifier leçon »	112
4.6	Description textuelle du cas d'utilisation « Passer un quiz »	113

5.1 Environnement Matériel 122

Table d'abréviations

API Application programming interface. [2](#), [126](#)

MVP Minimum Viable Product. [14](#)

OCP Office Chérifien des Phosphates. [2](#), [14](#), [126](#)

PFE Projet de Fin d'Étude. [14](#)

SMS Short Message Service. [126](#)

Introduction générale

La gestion des activités métiers devient de plus en plus un défi majeur pour les sociétés, un défi qui est aujourd'hui un point déterminant en termes d'optimisation des processus métiers ainsi que l'amélioration de leur visibilité et de leur gestion. Les entreprises manufacturières changent de stratégie au fur et à mesure de l'évolution des marchés. Soumises à de fortes pressions concurrentielles au cours des dernières décennies, les industries se sont orientées vers la digitalisation.

Dans ce cadre-là s'inscrit le sujet de mon [PFE](#) au sein de l'[OCP](#), dont le but est de concevoir et implémenter une solution informatique avec une architecture moderne pour digitaliser le processus métier. Dans notre situation est gestionné les anomalies dans les sites du groupe [OCP](#) lors de l'extraction de phosphates.

Le point de départ de notre projet est de faire une analyse profonde pour réaliser la première version [MVP](#) (Une réalisation qu'on peut la mettre en face des clients pour commencer à valider nos hypothèses), après on va faire des améliorations correspondants à nos besoins. L'équipe travaille avec une méthodologie Scrum selon les Epics tracés dans la RoadMap du projet.

Le présent rapport décrit l'ensemble du travail réalisé dans le cadre de ce projet, il contient quatre chapitres. Le premier chapitre contient une description du contexte général du projet notamment la présentation de la Digital Factory de l'[OCP](#) ainsi que la motivation et les objectifs du projet. Le deuxième chapitre présente une analyse de besoins fonctionnels et non fonctionnels. Par la suite le troisième chapitre mettra l'accent sur l'ensemble des éléments de l'étude conceptuelle. Enfin, le chapitre quatre présentera les résultats de l'implémentation.

Chapitre 1

Contexte Général Du Projet

1.1 Introduction

Ce premier chapitre est essentiel pour contextualiser le projet d'application d'apprentissage en ligne. Nous commençons par présenter brièvement l'organisme d'accueil où cette application sera élaborée. Ensuite, nous explorons les diverses solutions existantes dans le domaine de l'e-learning et nous déterminons les objectifs principaux que nous cherchons à atteindre avec cette application. Enfin, nous détaillons la méthodologie adoptée pour la gestion du projet, en mettant l'accent sur SCRUM comme cadre de travail agile. Ce chapitre jette ainsi les bases nécessaires pour une compréhension approfondie du projet. .

1.2 Présentation de l'organisme d'accueil

Dans le cadre de notre licence en technologie de l'informatique à l'Institut Supérieur des Études Technologiques de Mahdia, nous avons eu l'opportunité de réaliser notre projet de fin d'études au sein de la société LeaderPos.

1.2.1 Description de la société LeaderPos

Fondée en 2016, LeaderPos se positionne comme une entreprise experte dans divers domaines, notamment le matériel informatique, l'ingénierie logicielle, la création de sites Web, le développement d'applications mobiles et les systèmes de points de vente. Avec une expertise pointue dans ces domaines, LeaderPos offre une gamme complète de services de haute qualité pour répondre aux besoins variés de sa clientèle.

1.2.2 Logo de la société LeaderPos



FIGURE 1.1 – Logo De La société LeaderPos

TABLE 1.1 – Fiche Technique De LeaderPos

Catégorie	Informations
Emplacement	Rte Soukra, Km1, 5ème étage Rime Palace Sfax, Tunisie
Adresse Email	LeaderPos.tn@gmail.com
Numéro de Téléphone	frigui Adel +216 98 230 371 / Chahir +216 22 227 381

1.2.3 Informations de contact de la société LeaderPos

1.2.4 Services de la société LeaderPos

— Service Technique et Maintenance

LeaderPos dispose d'une équipe dédiée chargée d'assurer la maintenance préventive et corrective des équipements informatiques de ses clients. Cette équipe intervient rapidement pour résoudre les problèmes techniques et garantir le bon fonctionnement continu des systèmes informatiques, minimisant ainsi les interruptions d'activité et optimisant la productivité des clients.

— Service Installation et Déploiement

LeaderPos offre un service complet d'installation et de déploiement de ses solutions informatiques sur mesure. Cette prestation inclut la planification, l'installation physique des équipements, ainsi que la configuration et la mise en place des logiciels nécessaires. L'objectif est d'assurer une intégration harmonieuse des solutions au sein de l'infrastructure existante des clients, en garantissant une transition fluide vers les nouveaux systèmes.

— Service Développement

Avec une équipe expérimentée de développeurs, LeaderPos propose des services de développement sur mesure pour répondre aux besoins spécifiques de ses clients. Que ce soit pour la création d'applications mobiles, de logiciels sur mesure ou de plateformes web complexes, l'équipe de développement de LeaderPos travaille en étroite collaboration avec les clients pour concevoir des solutions innovantes et adaptées à leurs exigences.

— Service Commercial

Le service commercial de LeaderPos est dédié à comprendre et à répondre aux besoins des clients. Il agit comme un partenaire stratégique, offrant des conseils personnalisés et des solutions adaptées aux défis commerciaux des

clients. En plus de répondre aux demandes de renseignements et de fournir des informations sur les produits et les services, l'équipe commerciale de LeaderPos est également responsable de l'établissement de partenariats stratégiques avec d'autres acteurs du secteur de la technologie informatique, permettant ainsi aux clients de bénéficier des dernières innovations et des meilleures pratiques du marché.

1.2.5 Organigramme fonctionnel de de la société LeaderPos

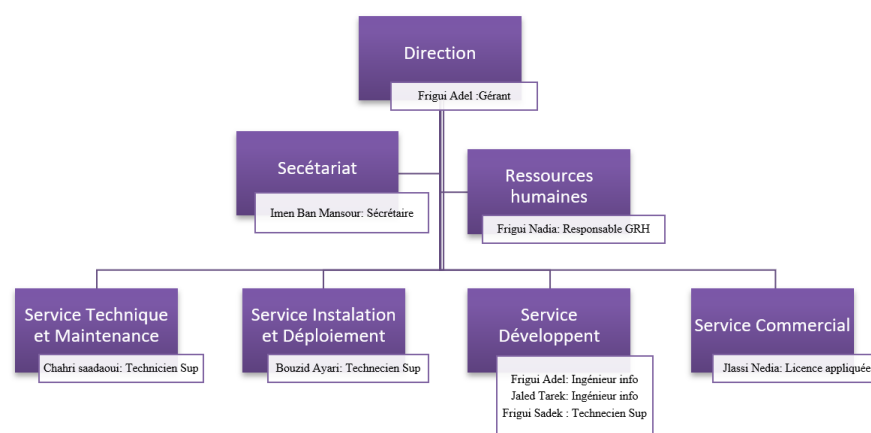


FIGURE 1.2 – Organigramme Fonctionnel De La Société LeaderPos

1.3 Étude de l'existant

L'étude de l'existant constitue le cœur de la phase d'analyse d'un projet. Cette étape est primordiale pour la mise en route de tout projet informatique ou autre, et permet de définir le contexte de fonctionnement, ainsi que le processus métier. Elle permet également de mettre en lumière les différentes imperfections dans le système actuel afin de les corriger. Dans cette partie, nous entreprendrons une analyse approfondie des solutions les plus reconnues sur le marché de l'éducation en ligne, qui exploitent diverses plateformes. L'objectif est d'identifier les avantages et les inconvénients de chacune d'entre elles.

- Udemy est un site internet de formation en ligne à destination des adultes et des étudiants. Il est en ligne depuis mai 2010. En janvier 2020, le site compte plus de 50 millions d'étudiants et 57 000 cours en 65 langues. [4]

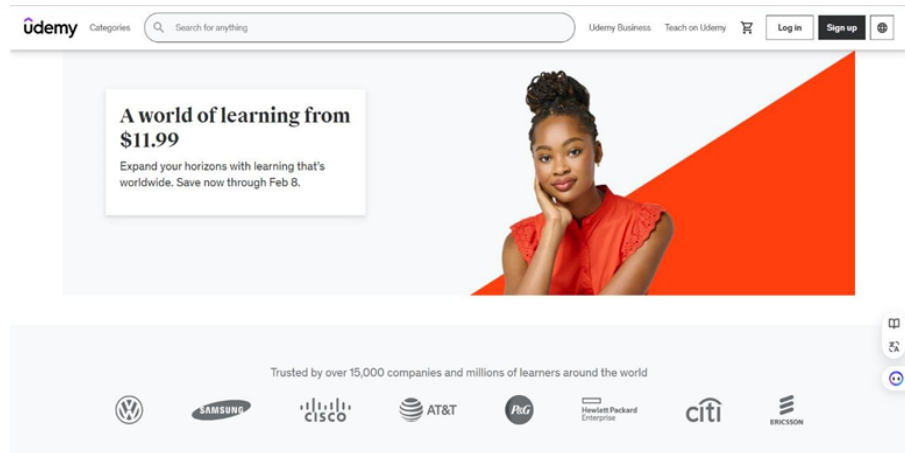


FIGURE 1.3 – Page D'Accueil De La Plateforme Udemy

- Coursera est un fournisseur de formation en ligne qui propose des cours en ligne, communément appelés MOOC ou Massive Open Online Courses, dispensés par les meilleures universités du monde. Elle compte actuellement plus de 275 partenaires dans une cinquantaine de pays [3].

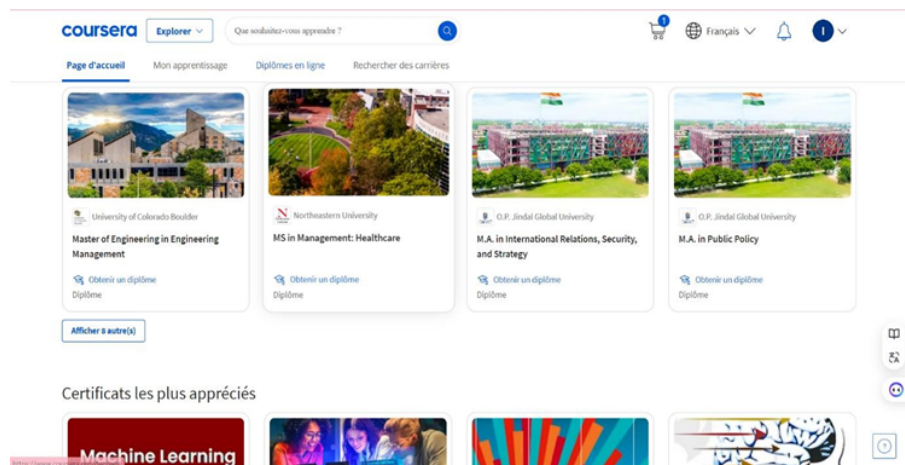


FIGURE 1.4 – Page D'Accueil De La Plateforme Coursera

- W3Schools : W3Schools est une plateforme en ligne populaire qui propose des didacticiels de développement Web et des références pour diverses technologies Web. Il constitue une ressource complète pour les personnes qui apprennent ou cherchent à améliorer leurs compétences en HTML, CSS, JavaScript et autres technologies connexes. [2].

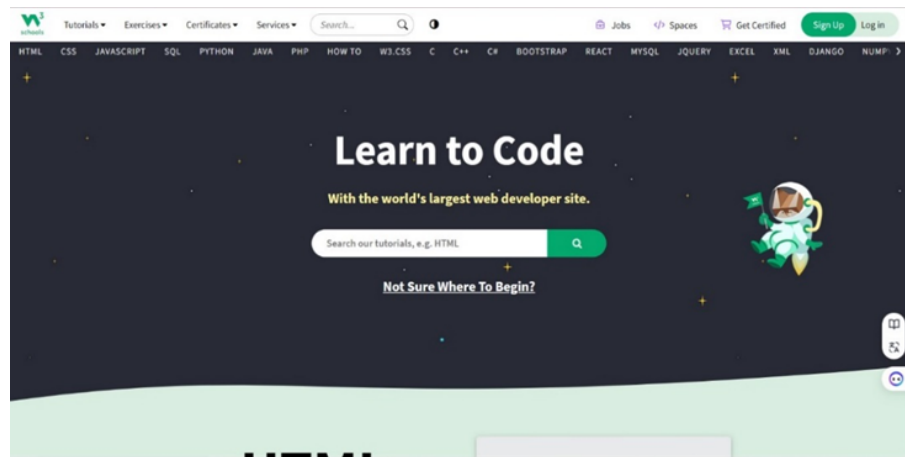


FIGURE 1.5 – Page D'Accueil De La Plateforme w3school

TABLE 1.2 – Analyse Comparative Des plateformes : Udemy, Coursera Et W3Schools

Critères	Coursera	Udemy	W3Schools
Facilitation de l'échange	✓ Communauté active	✓ Forums de discussion	✗
Partage de notifications	✓ Notifications par courriel et sur la plateforme	✓ Notifications par courriel	✗
Partage de cours/documents	✓ Possibilité de télécharger les ressources	✓ Partage de ressources pédagogiques	✓ Large gamme de tutoriels et ressources éducatives
Visioconférences	✓ Intégration de sessions en direct	✗	✗
Commentaires des enseignants	✓ Fonctionnalité de commentaire sur les cours	✗	✗
Création de quiz	✓ Possibilité de créer des quiz pour évaluation	✓ Outils de création de quiz	✗
Accès individuel aux profils	✓ Chaque utilisateur a son profil personnel	✓ Accès aux profils utilisateurs	✗

Critères	Coursera	Udemy	W3Schools
Accès gratuit	✗	✗	✓
Flexibilité	✗	✗	✓
Communauté	✓	✓	✓
Certification	✓	✓	✗
Navigation	Bien organisée	Facile	Facile
Support	Modéré	Limité	Limité
Prix	Coursera est moins cher. Le coût de la certification est de 49 \$	Udemy est très cher. Les cours varient de 50 \$ à 200 \$. Mais c'est très facile à obtenir des rabais qui réduisent les tarifs à 12,99 \$	Gratuit
Qui peut devenir instructeur ?	Les cours Coursera peuvent être dispensés uniquement par des éducateurs, des experts et des instructeurs professionnels.	Chez Udemy, presque tout le monde peut devenir instructeur.	✗
Assistance du forum communautaire	Coursera propose un support de forum communautaire.	Dans Udemy, il n'y a pas de support de forum communautaire.	✗
Aide du moniteur	L'assistance de l'instructeur concerne uniquement les cours Degree et MasterTrack.	L'assistance de l'instructeur est limitée, uniquement de la part des instructeurs du cours.	✗

Critères	Coursera	Udemy	W3Schools
Parcours d'apprentissage	Il existe un parcours d'apprentissage et des listes de lecture à travers des spécialisations de cours.	Il n'y a pas de parcours d'apprentissage ni de listes de lecture dans les cours Udemy.	✗

1.3.1 Critique de l'existant et solution proposée

1.3.1.1 Analyse de l'existant

Dans la partie précédente de notre recherche, nous avons repéré les avantages ainsi que les difficultés suivantes :

TABLE 1.3 – Tableau Comparatif : Avantages Et Inconvénients Des Plateformes D'Apprentissage En Ligne

A (+) / I (-)	Coursera	Udemy	W3Schools
Avantages (+)	<ul style="list-style-type: none"> - Partenariat avec des établissements d'enseignement du monde entier. - Structuré. - Certifications délivrées par le nom d'une institution professionnelle. - Choix entre des cours uniques, des spécialisations et des programmes menant à un diplôme. - Accessible sur téléphone mobile. - Vidéos téléchargeables. - Disponible dans plusieurs langues. - Communauté mondiale d'apprenants. - Planification des cours dans des logiciels comme Excel ou Google Docs. 	<ul style="list-style-type: none"> - Facile à suivre les progrès personnels. - Facile à obtenir des rabais. - Possibilité d'acheter des cours individuellement. - Disponible dans plusieurs langues. - Plus de 100,000 cours vidéo en ligne publiés chaque mois. - Inscription facile aux cours axés sur la carrière. - Certificats à la fin des cours. - Communauté amicale. - Possibilité d'auditer des cours gratuitement. - Cours visibles sur tous les appareils connectés à Internet. - Avis et évaluations des anciens étudiants disponibles. 	<ul style="list-style-type: none"> - Contenu gratuit et accessible à tous. - Large gamme de tutoriels et de ressources éducatives sur les technologies web. - Interface conviviale. - Contenu régulièrement mis à jour. - Possibilité d'apprendre à son propre rythme. - Grande communauté en ligne.

A (+) / I (-)	Coursera	Udemy	W3Schools
Inconvénients (-)	<ul style="list-style-type: none"> - Cours non flexibles. - Modération du forum. - Certains cours accessibles uniquement à des dates précises. - Impossible de revenir en arrière après la fin du cours. - Nécessité de lire des articles ou livres prescrits. - Attente du début du cours pour s'inscrire. - Aucune certification pour les cours gratuits. 	<ul style="list-style-type: none"> - Pas d'accréditation institutionnelle. - Coût élevé. - Difficulté à suivre plusieurs cours simultanément. - Cours obligatoires non disponibles. - Dates fixes pour les cours. - Impossible de suivre les cours complets gratuitement. - Besoin de discernement entre informations précises et obsolètes. - Manque de support direct. 	<ul style="list-style-type: none"> - Manque d'accréditation officielle ou de reconnaissance institutionnelle. - Qualité variable du contenu. - Absence de fonctionnalités avancées comme des projets pratiques ou des exercices interactifs. - Limitations dans la couverture de certains sujets ou technologies plus avancées. - Publicité ou promotion de produits tiers sur le site. - Nécessité de discernement entre informations précises et obsolètes. - Manque de support direct ou d'interaction avec des instructeurs ou des experts.

A (+) / I (-)	Coursera	Udemy	W3Schools
Idéal pour	<ul style="list-style-type: none"> - Chercheurs de cours sérieux dans des universités renommées. - Amateurs de cours et délais plus traditionnels. - Obtention de certificats d'établissements accrédités. 	<ul style="list-style-type: none"> - Chercheurs d'options budgétaires pour acquérir de nouvelles compétences à leur propre rythme. - Apprenants cherchant à recevoir des recommandations de cours personnalisées en fonction de leur emploi actuel. 	<ul style="list-style-type: none"> - Apprenants à la recherche de contenu gratuit et accessible. - Amateurs de tutoriels et ressources éducatives sur les technologies web. - Apprenants cherchant à apprendre à leur propre rythme.

1.3.2 Solution proposée

À la suite de l'analyse des limitations mentionnées dans la section précédente, nous avons élaboré une solution novatrice pour répondre aux besoins de gestion de plateforme éducative.

- **Apport sur le plan technique :** Notre solution repose sur le développement d'une application web intuitive, conçue pour simplifier le processus d'apprentissage tout en offrant une expérience utilisateur optimale. En utilisant les dernières technologies disponibles, nous avons mis en œuvre des fonctionnalités avancées pour garantir la fiabilité, la sécurité et la performance de la plateforme. Cela comprend notamment une architecture robuste, une navigation fluide et des mécanismes de sauvegarde et de récupération des données pour assurer une expérience utilisateur sans faille.
- **Apport sur le plan fonctionnel :** Notre solution vise à enrichir l'expérience d'apprentissage en offrant une gamme complète de fonctionnalités interactives tant pour les candidats que pour les instructeurs. Les candidats bénéficieront d'une interface conviviale leur permettant de naviguer facilement à travers les cours, de participer à des activités pratiques et d'interagir avec leurs pairs et leurs instructeurs. De leur côté, les instructeurs disposeront d'outils puissants pour créer, organiser et gérer efficacement leurs cours, tout en facilitant la

communication et la collaboration au sein de la communauté éducative.

1.4 Présentation du projet

Notre application, conçue pour la gestion des plateformes éducatives, est accessible via un navigateur internet standard et se divise en trois parties distinctes pour répondre aux besoins des utilisateurs.

La première partie de l'application est spécifiquement dédiée à la gestion des cours, offrant aux instructeurs la possibilité de créer de nouveaux cours, de publier du contenu pédagogique et de suivre la progression des candidats.

La seconde partie est réservée aux candidats, leur offrant une interface conviviale pour accéder aux cours, participer aux activités et interagir avec les instructeurs et les autres membres de la communauté éducative.

Quant à la troisième partie, elle est conçue pour les gestionnaires, leur proposant une interface dédiée où ils peuvent superviser les sujets des cours ainsi que d'autres aspects administratifs de manière efficace.

1.5 Langage et méthodologie de conception

La méthodologie, étant une démarche organisée et rationnelle visant à atteindre un résultat, se décline en plusieurs approches selon les besoins du projet. Parmi celles-ci, nous retrouvons le modèle en cascade, souvent utilisé pour les projets simples avec des besoins clairs et bien définis dès le départ, le modèle en Y, adapté au développement des applications mobiles, ainsi que le processus unifié et les méthodologies agiles (Scrum extrême programming), reconnues pour leur flexibilité et leur utilisation dans les grands projets. Pour garantir le bon déroulement des différentes phases de notre projet, nous avons choisi Scrum comme méthodologie de gestion de projet agile. Cette décision s'appuie sur les avantages de Scrum, notamment sa capacité à favoriser une collaboration efficace et une adaptation continue aux besoins changeants du projet. Bien que Scrum soit largement utilisé dans le développement logiciel, il peut également être appliqué à d'autres domaines. Cependant, il se concentre principalement sur la gestion des tâches et des équipes dans le cadre d'un projet, plutôt que sur la conception proprement dite. Bien que Scrum puisse être utilisé pour organiser le processus de conception, il ne fournit pas de directives spécifiques sur la manière de concevoir quelque chose. Après avoir sélectionné la méthodologie, il était nécessaire de choisir un langage de modélisation

unifié pour concevoir notre système. Nous avons opté pour UML en raison de ses nombreux atouts, tels que sa standardisation et la diversité des diagrammes qu'il propose. UML offre un moyen efficace pour schématiser des systèmes complexes de manière simplifiée et normalisée, grâce à un format graphique et textuel.

Il convient de noter que UML n'est ni un processus ni une approche spécifique, d'où la nécessité de choisir une méthodologie de conception et de développement qui puisse s'intégrer harmonieusement avec ce langage de modélisation.

1.6 Pourquoi Scrum

« Scrum signifie mêlée au rugby. Scrum utilise les valeurs et l'esprit du rugby et les adapte aux projets de développement. Comme le pack lors d'un ballon porté au rugby, l'équipe chargée du développement travaille de façon collective, soudée vers un objectif précis. Comme un demi de mêlée, le Scrum Master aiguillonne les membres de l'équipe, les repositionne dans la bonne direction et donne le tempo pour assurer la réussite du projet. » [1] Deux des signataires du Manifeste Agile, Ken Schwaber et Jeff Sutherland, ont développé Scrum au début des années 1990. Il fait partie de la catégorie des méthodes itératives et incrémentales et se base sur les principes et les valeurs de l'agile. Les spécialistes de Scrum, y compris ses fondateurs, le décrivent généralement comme un cadre ou un responsable de processus axé sur la gestion de projet, qui peut incorporer diverses méthodes ou pratiques d'ingénierie. Bien que la nature de Scrum puisse être difficile à définir, sa mise en place est relativement simple et peut être résumée de manière concise, comme illustré dans la Figure 6. Le principe fondamental de Scrum consiste à :

- **D'abord, identifier le maximum de fonctionnalités à réaliser afin de créer le backlog du produit.**
- **Ensuite, établir les priorités des fonctionnalités et sélectionner celles qui seront réalisées dans chaque étape.**
- **Ensuite, concentrer progressivement l'équipe sur toutes les fonctionnalités à accomplir, dans des étapes appelées Sprints.**
- **Chaque Sprint se termine toujours par la livraison d'un produit partiel fonctionnel appelé incrément.**

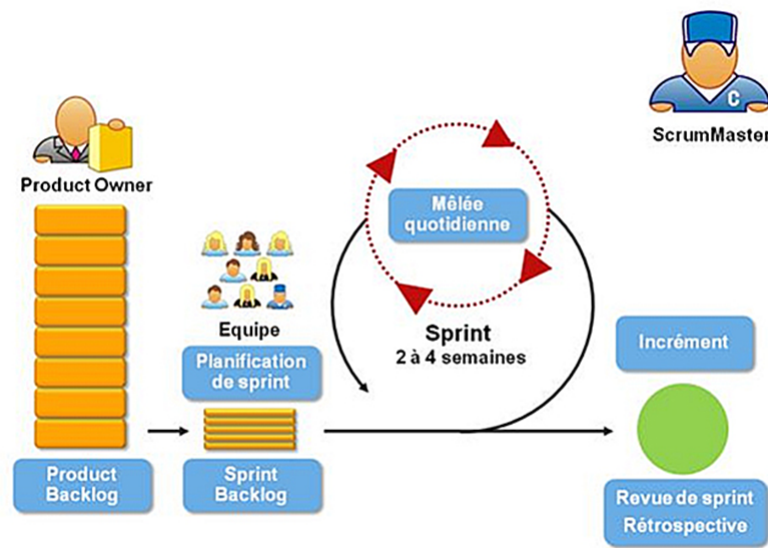


FIGURE 1.6 – Processus Scrum

Nous avons opté pour Scrum comme méthodologie de pilotage pour notre projet en raison des avantages qu'il présente. Son résumé est le suivant :

- **Une plus grande flexibilité et réactivité.**
- **Une grande capacité d'adaptation au changement grâce à des itérations courtes.**
- **Et ce qui compte le plus, c'est que Scrum combine les aspects théoriques et pratiques et se rapproche énormément de la réalité.**

Vu que Scrum ne couvrant que les aspects de gestion de projet, et pour compléter le vide laissé en matière de pratiques de développement, nous avons pris la décision de coupler Scrum avec une autre méthodologie agile qui est l'extrême programming et qui couvre les bonnes pratiques d'ingénierie logicielle notamment le développement dirigé par le test, qui sera détaillé dans les chapitres qui suivent, et la programmation en binôme, etc.

1.7 Choix des technologies

Nous avons sélectionné avec soin les technologies adaptées à notre projet afin de garantir son développement efficace et sa maintenance future. Notre choix s'est porté sur Angular pour l'interface utilisateur, Spring Boot pour la logique métier côté serveur, et MySQL comme système de gestion de base de données. Angular est un framework d'application Web monopage, basé sur TypeScript et dirigé par

l'équipe Angular de Google. Son écosystème dynamique rassemble plus de 1,7 million de développeurs, facilitant ainsi son évolution et sa pérennité. [5] Spring Boot, quant à lui, est un framework Java open source, conçu pour simplifier le développement d'applications Spring autonomes de qualité production. Sa capacité à minimiser les problèmes de configuration et son adoption généralisée en font un choix fiable pour notre logique métier côté serveur. [6] Enfin, MySQL a été retenu comme système de gestion de base de données en raison de sa compatibilité, de sa fiabilité et de sa stabilité. Son modèle relationnel s'aligne parfaitement avec la structure de nos données, facilitant ainsi leur gestion et leur organisation logique. [7] L'intégration de Spring Boot et Angular représente une approche stratégique pour notre projet d'apprentissage en ligne. Cette combinaison nous permettra de développer une plateforme fonctionnelle et immersive, offrant une expérience utilisateur optimale tout en garantissant une évolutivité et une maintenabilité optimales. En choisissant MySQL comme base de données relationnelle, nous assurons une intégration cohérente avec notre architecture technologique, renforçant ainsi la cohérence, la fiabilité et la performance de notre solution d'apprentissage en ligne.

1.8 Conclusion

Ce chapitre introductif a établi les fondements de notre projet d'application d'apprentissage en ligne. À travers la présentation de l'organisme d'accueil, nous avons défini le cadre de développement de notre plateforme. En examinant les solutions existantes dans le domaine de l'e-learning, nous avons identifié les objectifs clés que notre application vise à réaliser. De plus, en adoptant la méthodologie SCRUM, nous nous sommes engagés à suivre une approche agile pour la gestion du projet. Cette phase initiale nous a permis de poser les bases solides nécessaires à une analyse approfondie des besoins et à la spécification des fonctionnalités essentielles de notre application.

Chapitre 2

Planification et Architecture

2.1 Introduction

Dans le chapitre précédent, nous avons choisi d'utiliser la méthode Scrum pour créer notre futur système. Cette méthode se divise en trois phases, la première étant la planification et l'architecture (parfois appelée sprint 0). Cette phase est cruciale car elle affecte directement la réussite des sprints, en particulier le premier. Pendant cette période, nous avons travaillé pour clarifier la vision du produit, définir les rôles des utilisateurs, et identifier les principales fonctionnalités pour créer notre liste des tâches initiales. Nous avons également commencé à planifier les sprints. Ce chapitre vise à définir les besoins fonctionnels et non fonctionnels, les acteurs du projet, et l'architecture technique de notre application. Nous créerons également un diagramme global de cas d'utilisation, et présenterons les outils et technologies que nous utiliserons pour développer le projet.

2.2 Analyse des besoins

Pour une compréhension approfondie de l'application, il est essentiel de commencer par identifier les acteurs et définir les besoins fonctionnels et non fonctionnels. De plus, il est crucial de mettre en lumière les parties prenantes qui interagissent avec le système. Cette étape préliminaire garantit l'alignement de toutes les parties prenantes sur les objectifs et les exigences de l'application. En comprenant le contexte d'utilisation de l'application, il devient plus aisé de concevoir et de développer une solution répondant aux besoins de tous les intervenants.

2.2.1 Identification des acteurs

« Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. » [ref] Tous simplement un acteur est une entité physique (personne) ou abstraite (logiciel) capable d'utiliser le système afin de répondre à un besoin bien défini. Dans notre projet, il existe quatre acteurs cités ci-dessous :

Acteur	Description
Visiteur	Un utilisateur non authentifié qui peut naviguer sur la plate-forme pour explorer les sujets du cours, les cours disponibles. Ne peut pas accéder aux leçons, quiz ou autres fonctionnalités réservées aux utilisateurs authentifiés comme il a la possibilité de contacter le gestionnaire de la plateforme.
Gestionnaire	Un membre authentifié chargé de la gestion globale de la plate-forme. Crée, modifie et supprime des sujets pour organiser les contenus pédagogiques. Peut mettre à jour son propre profil et consulter les statistiques d'utilisation de la plateforme.
Instructeur	Un membre authentifié a la possibilité de gérer son profil et de créer, modifier et supprimer des cours. Il peut également gérer les leçons, les ressources et les quiz, y compris les questions et les réponses qui y sont liées. De plus, il peut organiser des réunions en ligne, suivre l'avancement des étudiants pour chaque cours et chaque quiz associé à ce cours, et discuter avec eux via un forum spécifique à chaque leçon.
Candidat	Un membre authentifié peut gérer son profil, explorer les sujets disponibles, accéder aux cours auxquels il est inscrit, participer à des réunions avec les enseignants, et consulter ou passer des quiz liés à ces cours. Après chaque cours et chaque quiz, il a également la possibilité d'évaluer sa compréhension et de discuter avec ses enseignants via un forum dédié à chaque leçon.

TABLE 2.1 – Description Des acteurs De La Plateforme D'Apprentissage

2.2.1.1 Diagramme de contexte statique

Ce diagramme d'UML permet simplement de montrer la relation des différents acteurs avec le système. Il spécifie le nombre d'instances de chaque acteur relié au système à un moment donné.

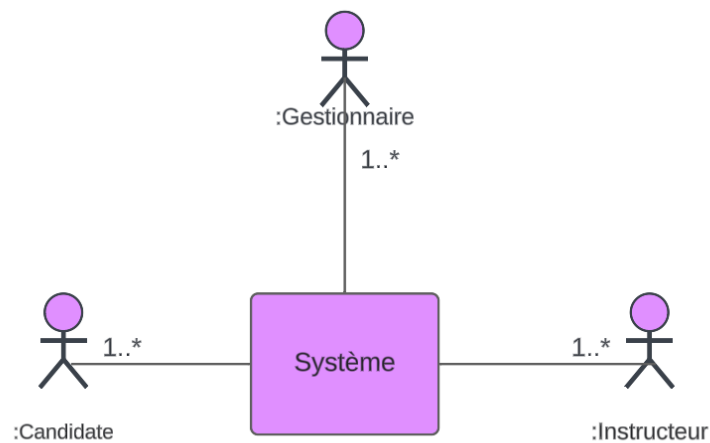


FIGURE 2.1 – Diagramme De Contexte Statique

2.2.2 Les besoins fonctionnels

Dans cette section, nous présentons les besoins fonctionnels des différents acteurs impliqués dans notre application de plateforme d'apprentissage en ligne.

TABLE 2.2 – Exigences Fonctionnelles

Acteur	Exigences Fonctionnelles
Gestionnaire	S'authentifier : Le gestionnaire doit pouvoir s'identifier sur la plateforme en utilisant des informations d'identification spécifiques. Cette démarche vise à assurer que seules les personnes autorisées ont accès aux fonctionnalités d'administration.

	<p>Gérer les comptes des membres : Le gestionnaire de la plateforme est chargé de gérer les comptes des instructeurs et des candidats. Son rôle principal est d'ajouter de nouveaux comptes selon les besoins et de supprimer ceux qui ne sont plus nécessaires. Lors de la création d'un nouveau compte, un e-mail est automatiquement envoyé à l'adresse associée, fournissant les informations de connexion requises. Cette procédure assure une gestion sécurisée et efficace des comptes, permettant à chaque utilisateur d'accéder facilement à la plateforme. De plus, le gestionnaire peut modifier l'état d'un compte de "actif" à "inactif" si l'utilisateur n'a pas utilisé son compte depuis plus d'un mois. Cette fonctionnalité garantit la sécurité en désactivant les comptes inutilisés, tout en conservant la possibilité de les réactiver au besoin.</p>
	<p>Gérer les sujets : cette fonctionnalité permet au gestionnaire de gérer les sujets couverts par la plateforme, y compris l'ajout, la modification ou la suppression de sujets.</p>
	<p>Consulter les statistiques de la plateforme : Le gestionnaire doit avoir la possibilité de consulter des données statistiques et analytiques sur l'utilisation de la plateforme, telles que le nombre d'utilisateurs actifs, etc.</p>
	<p>Gestion de profil : le gestionnaire peut également gérer son propre profil sur la plateforme, ce qui lui permet de mettre à jour ses propres informations personnelles, telles que son nom, son adresse électronique. Cela lui offre un contrôle total sur son compte administrateur.</p>
Instructeur	<p>S'authentifier : Les instructeurs doivent pouvoir accéder à leur compte de manière sécurisée en utilisant des informations d'identification appropriées, telles qu'une adresse e-mail et un mot de passe.</p>
	<p>Gérer le profil : les instructeurs doivent pouvoir consulter et mettre à jour leur profil personnel sur la plateforme.</p>

	<p>Gérer les Cours : Les instructeurs peuvent créer de nouveaux cours avec des détails tels que le titre, la description, l'image, ainsi que les dates de début, de fin et d'engagement. Ils peuvent également modifier ou supprimer les cours existants.</p>
	<p>Gérer les Leçons : Les instructeurs peuvent créer des leçons individuelles pour chaque cours. Ils ont la possibilité d'organiser les leçons dans un ordre logique et de les modifier à tout moment.</p>
	<p>Gérer des Ressources : Les instructeurs peuvent associer des ressources spécifiques à chaque cours et à chaque leçon. Ces ressources comprennent du contenu pédagogique tel que du texte, des vidéos, des images, des fichiers PDF, etc. Les instructeurs peuvent ajouter, modifier ou supprimer des ressources en fonction des besoins d'enseignement et d'apprentissage.</p>
	<p>Gérer des quiz : Les instructeurs ont la capacité de concevoir des quiz visant à évaluer les connaissances des candidats. Ils peuvent assigner ces quiz à des cours spécifiques afin d'évaluer la compréhension des étudiants. De plus, ils ont la possibilité de créer de nouveaux quiz en attribuant des questions déjà existantes dans d'autres quiz de ce même cours.</p>
	<p>Gérer des Questions et ces réponses possibles : Dans le cadre de la création de quiz, les instructeurs peuvent ajouter et gérer des questions individuelles. Ils peuvent spécifier le type de question (choix multiples, un choix, vrai/faux, etc.) et fournir les réponses possibles. Ils peuvent modifier les questions et les réponses au besoin.</p>
	<p>Évaluer les candidats : les instructeurs doivent avoir accès à une liste complète des notes attribuées à chaque candidat pour chaque quiz ou évaluation.</p>

	<p>Gérer réunions : les instructeurs ont la capacité d'organiser des réunions virtuelles avec les candidats grâce à des outils de visioconférence intégrés. Ils peuvent créer des réunions, planifier des sessions à des moments précis, et modifier les détails des réunions prévues si nécessaire. De plus, les instructeurs peuvent annuler des réunions si besoin. Ils ont également la possibilité de consulter la liste des réunions qu'ils ont programmées, assurant ainsi une gestion efficace des interactions en ligne avec les candidats.</p>
	<p>Créer et gérer des forums de discussion : Les instructeurs ont la possibilité de créer automatiquement un forum de discussion pour chaque leçon, mais ils peuvent également initier des discussions et envoyer des messages aux candidats. Ces messages peuvent contenir divers formats de contenu tels que des fichiers PDF, des enregistrements audio, des images, des vidéos, et bien sûr, du texte. Cela permet aux instructeurs d'enrichir les discussions avec une variété de supports multimédias, offrant ainsi une expérience d'apprentissage plus dynamique et engageante pour les participants.</p>
Candidat	<p>S'authentifier : les candidats doivent pouvoir accéder à leur compte de manière sécurisée en utilisant des informations d'identification appropriées, telles qu'un nom d'utilisateur et un mot de passe.</p>
	<p>Gérer leur profil : les candidats doivent avoir la possibilité de consulter et de mettre à jour leur profil personnel sur la plateforme. Cela inclut la modification d'informations telles que la photo de profil, les coordonnées et les préférences de compte.</p>
	<p>Consulter les sujets : les candidats doivent pouvoir consulter les sujets disponibles sur la plateforme pour explorer les domaines d'apprentissage qui les intéressent.</p>

	<p>Consulter les Caractéristiques du Cours : Elle permet aux candidats d’avoir un aperçu détaillé des cours auxquels ils sont inscrits avant de commencer leur apprentissage. Elle présente des informations telles que le nombre de candidats inscrits au cours, l’identité de l’instructeur qui l’a créé, ainsi que les dates de création et de dernière modification du cours. De plus, elle donne un aperçu du nombre de leçons et de ressources disponibles, fournissant ainsi aux candidats une vision complète et claire du contenu et de la structure du cours.</p>
	<p>Accéder aux cours : les candidats doivent avoir accès aux cours auxquels ils sont inscrits, avec la possibilité de consulter le contenu des leçons et ces ressources.</p>
	<p>Participer aux réunions : les candidats doivent avoir la possibilité de participer à des réunions virtuelles organisées par les instructeurs pour discuter du contenu du cours, poser des questions et interagir avec d’autres candidats.</p>
	<p>consulter emploi : les candidats peuvent consulter leur emploi du temps pour connaître les détails de leurs réunions planifiées, et ils peuvent également recevoir des e-mails contenant des informations pertinentes concernant leurs prochaines réunions, offrant ainsi une expérience d’apprentissage bien organisée et fluide.</p>
	<p>Consulter ou passer des quiz : les candidats doivent pouvoir consulter, passer des quiz associés aux cours. Pour évaluer leur compréhension et tester leurs connaissances</p>

2.2.3 Les besoins non fonctionnels

Après avoir défini clairement les besoins fonctionnels, il est essentiel de considérer les besoins non fonctionnels tout au long du processus de développement de la plateforme. Ces exigences portent sur les éléments perceptibles par l’utilisateur, mais qui ne sont pas directement liés au fonctionnement du système. Leurs fonctions

sont cruciales pour assurer la qualité globale du système et garantir une expérience utilisateur optimale. Par conséquent, plusieurs éléments essentiels sont mis en avant afin de satisfaire ces besoins non fonctionnels.

- **Sécurité** : Le système garantit la confidentialité des données en utilisant des mécanismes d'authentification basés sur des jetons (token-based authentication) via Spring Boot Security. Les informations sensibles sont cryptées à l'aide de JSON Web Tokens (JWT), garantissant la protection contre les accès non autorisés. En appliquant le Principe de Responsabilité Unique (SRP), chaque composant de sécurité est conçu pour gérer une seule responsabilité.
- **Ergonomie des interfaces** : L'ergonomie des interfaces est primordiale pour assurer une expérience utilisateur fluide et intuitive. Le Principe Ouvert-Fermé (OCP) est adopté pour garantir que les interfaces utilisateur sont évolutives sans altérer leur fonctionnement existant. En concevant des composants d'interface ouverts à l'extension, mais fermés à la modification, il est possible d'introduire de nouvelles fonctionnalités sans perturber l'expérience utilisateur déjà établie.
- **Performance** : Le Principe d'Inversion de Dépendance (DIP) peut être utilisé pour améliorer la performance de l'application en réduisant les dépendances directes entre les modules. En minimisant les dépendances, il est plus facile de mettre en cache les données et d'optimiser le code, ce qui contribue à une meilleure réactivité de l'application et à des temps de chargement plus rapides.
- **Fiabilité** : En respectant le Principe SRP et en appliquant des tests unitaires, la fiabilité de l'application est renforcée. En isolant chaque fonctionnalité dans des composants logiciels distincts, les risques d'erreurs et de défaillances sont réduits, garantissant ainsi une expérience utilisateur plus fiable.
- **Maintenance** : Les principes SOLID, notamment le SRP et le DIP, facilitent la maintenance de l'application en favorisant une conception modulaire et flexible. En réduisant les dépendances et en isolant les responsabilités, le code devient plus facile à comprendre, à modifier et à mettre à jour, ce qui réduit les temps d'arrêt et simplifie la maintenance continue de l'application.
- **Portabilité responsive** : L'application est conçue pour s'adapter et fonctionner de manière optimale sur différents navigateurs et appareils. Cela assure une expérience utilisateur cohérente et satisfaisante, quel que soit le dispositif utilisé pour accéder à l'application.

- **Réutilisabilité** : L'architecture du projet favorise la réutilisabilité du code en utilisant des modules indépendants, des services de domaine génériques et des patterns de conception appropriés. Cela réduit les efforts de développement et améliore l'efficacité globale du processus de développement logiciel.

2.3 Planning du traitement de cas d'utilisation

À la suite de l'identification de tous les cas d'utilisation de notre application e-learning, nous les classifions en fonction de deux éléments fondamentaux : la priorité et les risques. Cette approche a pour objectif de structurer notre travail de développement de manière efficace et de réduire au minimum les obstacles éventuels.

2.3.1 Priorités

Dans notre application e-learning, chaque cas d'utilisation est évalué en fonction de son impact sur la qualité globale du système et sur l'expérience utilisateur. La planification des priorités est cruciale pour diriger nos ressources vers les fonctionnalités clés et garantir une expérience utilisateur optimale. Par exemple, la fonctionnalité de création de cours peut être considérée comme plus critique que la modification de profil, car elle apporte une valeur directe à l'application. Ainsi, nous avons identifié trois niveaux de priorité - élevé, moyen et faible - afin d'organiser efficacement nos efforts de développement.

2.3.2 Risques

Pour gérer notre projet d'apprentissage en ligne, nous mettons l'accent sur l'identification et la gestion des risques. Il est essentiel de planifier ces risques afin d'anticiper et de réduire les difficultés qui pourraient entraver la réussite de notre projet. Les principales menaces auxquelles nous pourrions faire face sont liées à la complexité de l'application et aux contraintes particulières du domaine de l'éducation en ligne. Par exemple, des difficultés de performance ou de sécurité pourraient mettre en péril la réussite du déploiement de notre projet. En mettant en place des stratégies visant à réduire ces risques dès le début du projet, nous sommes davantage préparés à relever les défis potentiels et à garantir la progression efficace de notre application e-learning.

2.4 Prototypage des interfaces

Le prototypage des interfaces revêt une importance capitale dans le développement web. Grâce à des outils spécialisés tels que Figma, nous avons pu élaborer des maquettes interactives, offrant ainsi une représentation concrète de notre application. Ces prototypes ne servent pas seulement à évaluer la satisfaction du client, mais ils facilitent également la communication au sein de l'équipe de conception. L'analyse de ces prototypes a engendré des discussions constructives entre les membres de l'équipe, permettant une amélioration précise des besoins et des attentes du projet. Grâce à cette approche itérative, nous nous assurons de comprendre en profondeur les exigences du projet et de créer une application web qui y répond parfaitement.

Les figures ci-dessous présentent un aperçu de quelques-unes des interfaces que nous avons conçues à l'aide de l'outil Figma

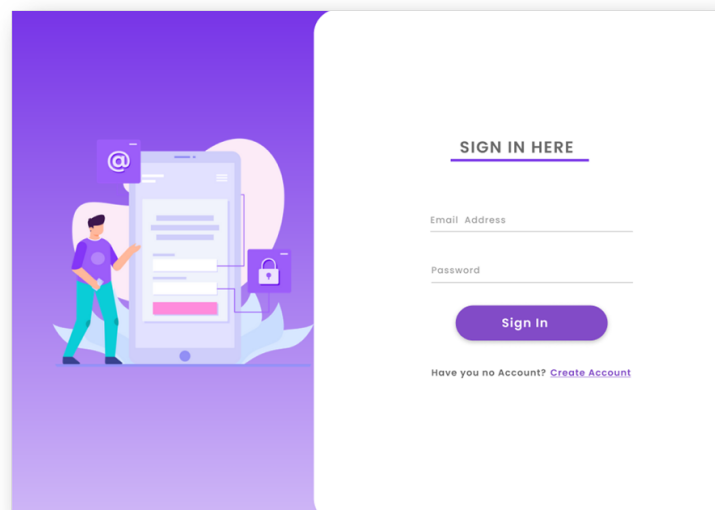


FIGURE 2.2 – Page De Connexion

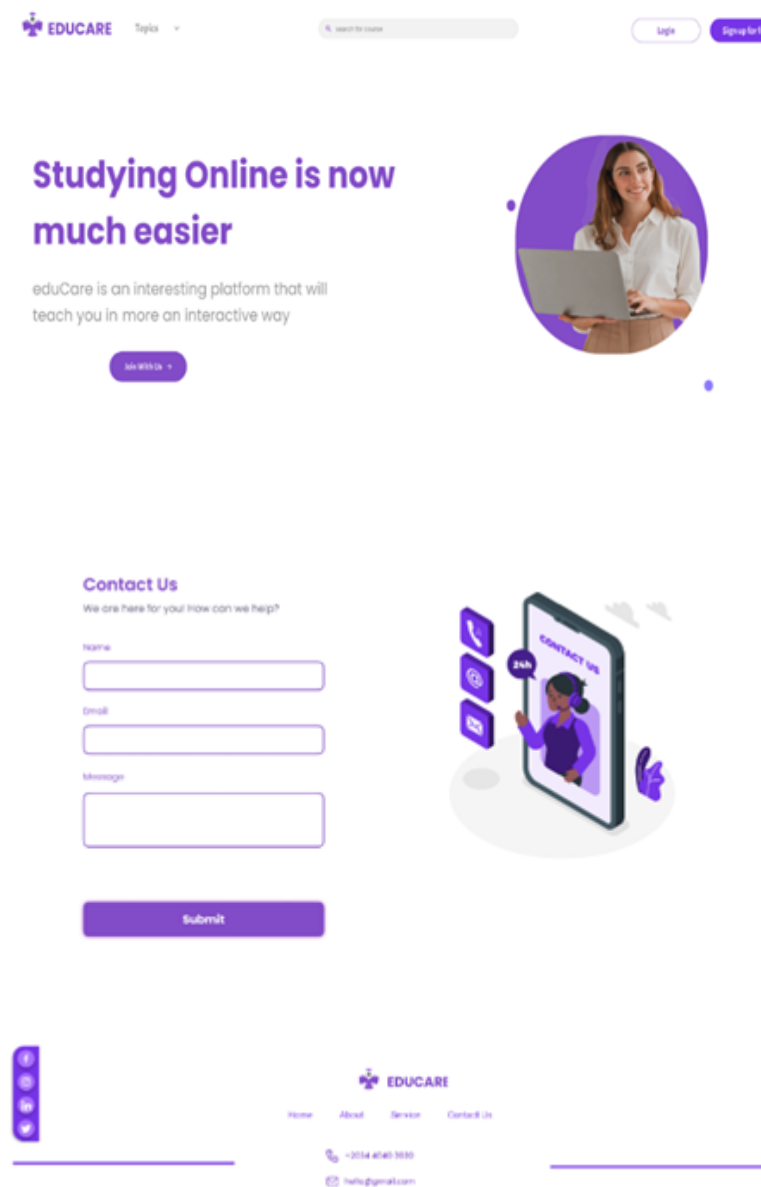


FIGURE 2.3 – Interface D’Accueil De Visiteur

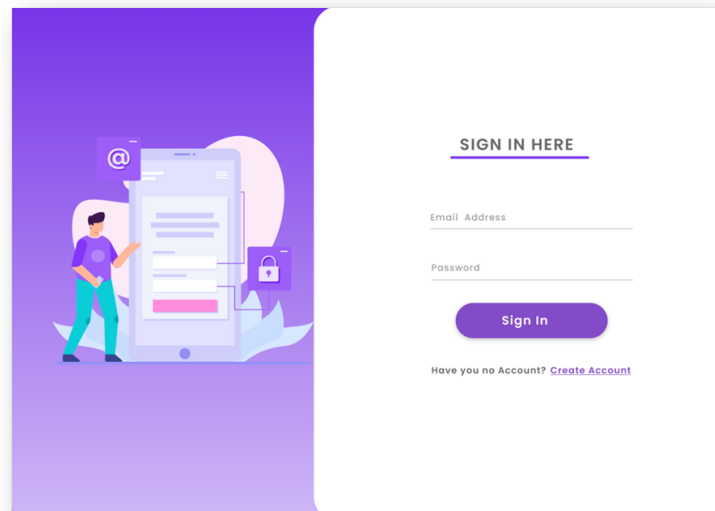


FIGURE 2.4 – Interface De Profil Instructeur

2.5 Pilotage du projet avec Scrum

Scrum est un cadre de travail itératif et incrémental qui favorise la collaboration, la transparence et l'adaptation continue. De ce fait, l'un des rôles clés dans Scrum est celui de l'équipe qui répond constamment aux besoins demandés afin d'atteindre le meilleur résultat possible.

2.5.1 Les outils Scrum

Pour le pilotage de leurs projets Scrum, les membres de l'équipe font recours à plusieurs techniques. Une de ces techniques, qui est la plus répandue, consiste à créer des fiches (post It) et de les coller sur un mur ou sur un tableau visible pour tous les membres de l'équipe. Une autre technique consiste à utiliser un fichier Excel contenant toutes les informations nécessaires pour les sprints, les user story leurs estimations, etc. Ce fichier devra être partagé en lecture et en écriture (pour que tous les membres de l'équipe puissent le modifier à tout moment). Par conséquent, plusieurs outils sont apparus en offrant la possibilité de suivre la priorité, la traçabilité et la gestion de tout le travail associé. Parmi les outils existants, nous avons choisi d'utiliser Trello.

2.5.2 Équipe et rôles

« L'équipe a un rôle capital dans Scrum : elle est constituée avec le but d'optimiser la flexibilité et la productivité ; pour cela, elle s'organise elle-même et doit avoir toutes les compétences nécessaires au développement du produit. Elle est investie avec le pouvoir et l'autorité pour faire ce qu'elle a à faire ». [ref] Bref, Scrum définit trois rôles qui sont :

Le Product Owner (le propriétaire du produit) : Représente les besoins métier, définit les objectifs du projet, priorise les fonctionnalités à développer et maximise la valeur du produit en maintenant le Product Backlog.

Le Scrum Master (le directeur de produit) : Facilite l'équipe Scrum en veillant à ce qu'elle suive les principes et les pratiques Scrum, en éliminant les obstacles et en favorisant l'amélioration continue. Il assure le bon déroulement des réunions Scrum.

Le Scrum Team (l'équipe de Scrum) : Réalise les fonctionnalités du Product Backlog, possède toutes les compétences nécessaires, et collabore pour concevoir, développer, tester et livrer des fonctionnalités de qualité. Dans notre projet, Monsieur Tarek Jalled assumera le rôle de Product Owner, tandis que Madame Nahla Sassi sera le Scrum Master. Les développeurs responsables de la mise en œuvre des fonctionnalités seront Syrine Elayeb et Ikram Ayoub.

2.5.3 Le backlog du produit

Pour formaliser les besoins de l'application, nous avons élaboré le Backlog Produit. Ce dernier constitue une liste hiérarchisée de macro-fonctionnalités, nommées User Stories, qui orienteront le développement de l'application. Chaque élément est soigneusement classé par ordre de priorité, déterminant ainsi la séquence de leur mise en œuvre. Le Backlog Produit représente un outil central dans la méthodologie Scrum, permettant de planifier, de prioriser et de suivre le travail. Il présente les éléments suivants :

- **ID** : Identifiant unique attribué à chaque User Story.
- **Thème** : Catégorie ou thème auquel cette User Story est reliée.
- **User Story** : Les User Stories représentent les spécifications du projet sous forme d'histoires utilisateurs. Elles décrivent les interactions attendues de l'utilisateur avec le système, fournissant une compréhension détaillée des fonctionnalités à développer et des résultats attendus.

- **Estimation** : Cette valeur indique le temps estimé nécessaire pour compléter chaque User Story, exprimé en jours.
- **Priorité** : Indication du niveau d'importance de chaque User Story par rapport aux autres éléments du backlog. Cette priorité guide la séquence de développement, en accordant la priorité aux fonctionnalités les plus cruciales en premier lieu.

En identifiant les besoins définis dans le cahier des charges, nous avons pu élaborer le backlog de notre produit, présenté dans le tableau ci-dessous :

TABLE 2.3 – Backlog de Produit

ID	Thème	User Story	Estimation	Priorité
1	Inscription et Accès à la Plateforme	US001 En tant que visiteur, je veux m'inscrire à la plateforme pour devenir un membre de plateforme et accéder à l'ensemble des fonctionnalités.	5	Haute
2	Connexion sécurisée pour les membres	US002 En tant que membre, je veux pouvoir me connecter de manière sécurisée pour garantir la confidentialité de mes données.	4	Haute
3	Gestion des comptes profils	US003 En tant que membre, je veux pouvoir consulter mon profil pour visualiser mes informations personnelles.	3	Haute
		US004 En tant que membre, je veux pouvoir modifier mes informations personnelles pour les mettre à jour.	2	Moyenne
		US005 En tant que membre, je veux pouvoir supprimer mon compte si nécessaire.	2	Haute
4	Gestion de la confidentialité	US006 En tant que membre, je veux pouvoir modifier mon mot de passe pour garantir la sécurité de mon compte.	2	Haute

ID	Thème	User Story	Estimation	Priorité
		US007 En tant que membre, je veux pouvoir désactiver temporairement mon compte si besoin.	2	Moyenne
		US008 En tant que membre, je veux pouvoir supprimer définitivement mon compte pour effacer toutes mes données.	2	Moyenne
5	Gestion des sujets	US011 En tant que gestionnaire, je veux pouvoir ajouter un sujet.	4	Haute
		US012 En tant que gestionnaire, je veux pouvoir modifier un sujet.	3	Haute
		US013 En tant que gestionnaire, je veux pouvoir supprimer un sujet.	3	Haute
		US014 En tant que visiteur, je souhaite pouvoir consulter les différents sujets, rechercher des sujets et accéder aux cours associés à ces sujets.	3	Haute
		US015 En tant que candidat, je veux pouvoir consulter les sujets récents.	3	Haute
6	Gestion des comptes	US015 En tant que gestionnaire, je veux pouvoir ajouter un compte membre à la plateforme afin d'élargir la base d'utilisateurs.	4	Haute
		US016 En tant que gestionnaire, je veux pouvoir supprimer un compte membre de la plateforme si nécessaire.	4	Moyenne

ID	Thème	User Story	Estimation	Priorité
		US017 En tant que gestionnaire, je veux pouvoir consulter les détails d'un compte membre de la plateforme afin d'accéder aux informations nécessaires à la gestion des utilisateurs.	4	Moyenne
7	Gestion des cours	US016 En tant qu'instructeur, je veux créer un nouveau cours assigné à un sujet pour offrir une structure claire aux candidats.	5	Haute
		US017 En tant qu'instructeur, je veux pouvoir modifier les détails d'un cours existant pour maintenir la pertinence et la qualité du contenu.	4	Haute
		US018 En tant qu'instructeur, je veux pouvoir supprimer un cours existant afin de gérer efficacement les ressources pédagogiques.	3	Haute
		US039 En tant qu'instructeur, je veux pouvoir suivre le progrès des candidats dans chaque cours pour évaluer leur performance et leur compréhension du contenu.	4	Haute
		US021 En tant que candidat, je veux pouvoir consulter les cours récemment ajoutés, examiner les détails des cours (nombre de candidats inscrits, nom de l'instructeur, etc.) et m'inscrire à un cours pour sélectionner ceux les plus adaptés à mes besoins et intérêts.	4	Haute

ID	Thème	User Story	Estimation	Priorité
		US022 En tant que candidat, je veux pouvoir consulter mes cours inscrits pour suivre ma progression et accéder facilement aux leçons.	3	Haute
		US020 En tant que visiteur, je veux pouvoir consulter les cours disponibles pour avoir un aperçu de l'offre de formation.	4	Haute
8	Gestion des Leçons	US021 En tant qu'instructeur, je veux créer une leçon pour un cours spécifié afin de structurer le contenu pédagogique.	5	Haute
		US022 En tant qu'instructeur, je veux pouvoir modifier les détails d'une leçon existante pour mettre à jour ou améliorer le contenu.	4	Haute
		US023 En tant qu'instructeur, je veux pouvoir supprimer une leçon existante afin de gérer efficacement le programme du cours.	3	Haute
		US024 En tant que candidat, je veux pouvoir accéder aux leçons des cours auxquels je suis inscrit afin de suivre le contenu et progresser dans ma formation.	2	Haute
9	Gestion des ressources	US025 En tant qu'instructeur, je veux pouvoir ajouter des ressources, telles que des documents PDF et des vidéos , à chaque leçon afin d'enrichir le contenu pédagogique.	5	Haute

ID	Thème	User Story	Estimation	Priorité
		US026 En tant qu'instructeur, je veux pouvoir modifier les détails des ressources existantes pour une leçon spécifiée afin de mettre à jour ou corriger le contenu.	4	Haute
		US027 En tant qu'instructeur, je veux pouvoir supprimer des ressources existantes pour une leçon spécifiée afin de maintenir la pertinence et la qualité du matériel pédagogique.	3	Haute
		US028 En tant que candidat, je veux pouvoir accéder facilement aux ressources ajoutées par l'instructeur pour chaque leçon afin d'approfondir ma compréhension et mieux préparer mes études.	3	Haute
10	Gestion des quiz	US034 En tant qu'instructeur, je veux pouvoir créer un quiz interactif pour chaque cours pour évaluer les connaissances des candidats.	5	Haute
		US035 En tant qu'instructeur, je souhaite pouvoir modifier les détails d'un quiz déjà créé.	4	Haute
		US036 En tant qu'instructeur, je veux pouvoir supprimer un quiz qui n'est plus pertinent ou nécessaire.	3	Haute
		US038 En tant qu'instructeur, je veux pouvoir suivre le progrès des candidats pour chaque quiz passé afin de mieux évaluer leur performance et leur compréhension du contenu.	4	Haute

ID	Thème	User Story	Estimation	Priorité
		US039 En tant que membre, je veux pouvoir consulter la liste des quiz disponibles sur la plateforme.	3	Moyenne
		US040 En tant que candidat, je veux pouvoir participer aux quiz attribués à mes cours.	4	Haute
		US041 En tant que candidat, je souhaite pouvoir consulter mes propres résultats après avoir terminé un quiz.	3	Moyenne
		US042 En tant que candidat, je veux être notifié lorsqu'un nouveau quiz est attribué à une de mes cours.	4	Moyenne
12	Gestion des meetings	US044 En tant qu'instructeur, je souhaite pouvoir planifier des réunions à une date et une heure spécifique.	3	Haute
		US045 En tant qu'instructeur, je veux pouvoir modifier les détails d'une réunion déjà planifiée.	2	Haute
		US046 En tant qu'instructeur, je souhaite pouvoir annuler une réunion si nécessaire.	2	Haute
		US047 En tant qu'instructeur, je veux pouvoir consulter la liste des réunions que j'ai planifiées.	2	Haute
		US048 En tant que candidat, je veux pouvoir accéder aux réunions planifiées par mes instructeurs.	3	Haute

ID	Thème	User Story	Estimation	Priorité
		US049 En tant que candidat, je souhaite pouvoir participer activement aux réunions en temps réel.	4	Haute
		US050 En tant que candidat, je veux être notifié lorsqu'une nouvelle réunion est planifiée pour un de mes cours.	2	Moyenne
13	Gestion des forums de discussion	US054 En tant qu'instructeur, je veux pouvoir consulter les forums de discussion de mes leçons créés par les candidats et répondre aux discussions.	5	Haute
		US055 En tant qu'instructeur, je veux pouvoir éditer ou supprimer mes propres messages dans le forum de leçons.	3	Haute
		US056 En tant que candidat, je veux pouvoir créer un nouveau sujet de discussion, ajouter un message sous forme de texte et/ou PDF et/ou image et/ou enregistrement et/ou vidéo, répondre aux discussions existantes, et modifier ou supprimer mes propres réponses.	5	Haute

2.5.4 Diagramme de cas d'utilisation global

Dans cette section, nous présentons les besoins de notre système de manière formelle en utilisant le langage de modélisation UML et ses diagrammes de cas d'utilisation. L'utilisation d'UML présente plusieurs avantages pour notre application web. Tout d'abord, cela permet d'offrir une documentation synthétique qui rassemble efficacement les parties prenantes autour des besoins du projet. De plus, UML offre une modélisation indépendante des langages de programmation et des environnements

technologiques, facilitant ainsi la communication entre les développeurs et les autres membres de l'équipe.

En utilisant UML, nous pouvons exprimer de manière claire les aspects statiques et dynamiques de notre système d'information. Les différents types de diagrammes UML, comme les diagrammes de cas d'utilisation, de classes, séquence, etc., nous permettent de représenter de manière exhaustive les fonctionnalités du système ainsi que les interactions entre les différents composants.

Dans cette partie de notre rapport, nous mettons en avant la présentation de cas d'utilisation, qui sont essentiels pour définir et clarifier les fonctionnalités de notre projet. Les cas d'utilisation offrent une vue globale des interactions entre les utilisateurs et le système, mettant en lumière les actions que chaque acteur peut entreprendre. En nous concentrant sur les cas d'utilisation, nous identifions de manière précise et organisée les besoins fonctionnels du projet, facilitant ainsi la compréhension des exigences par toutes les parties prenantes impliquées.

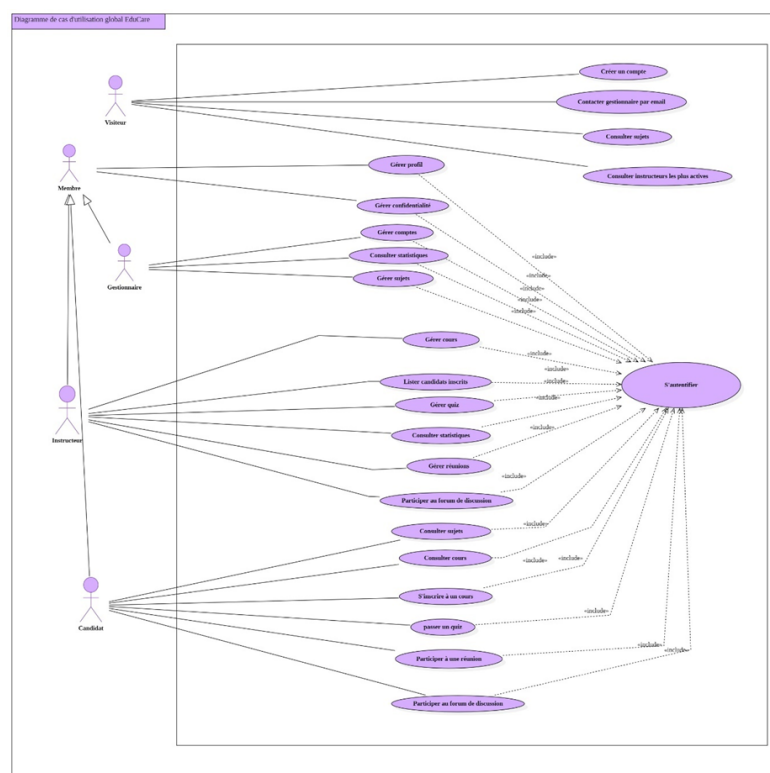


FIGURE 2.5 – Diagramme De cas D'Utilisation Global

Le diagramme de cas d'utilisation de notre plateforme d'apprentissage en ligne présente les différentes actions disponibles pour chaque type d'utilisateur, en tenant compte de l'authentification requise. Les utilisateurs non authentifiés, tels que les

visiteurs, peuvent explorer le contenu de la plateforme sans créer de compte. Les membres, une fois authentifiés, peuvent gérer leur profil. Les gestionnaires, qui héritent des fonctionnalités des membres, ont en plus la capacité de gérer les comptes membre, les statistiques et les sujets. Les instructeurs, également héritiers des fonctionnalités de base, peuvent créer et gérer des cours, organiser des réunions et animer des forums de discussion. Les candidats, une fois authentifiés, peuvent consulter les sujets, les cours et les leçons, ainsi que participer aux réunions organisées par les instructeurs. Cette représentation illustre de manière concise les actions disponibles pour chaque acteur, en mettant en évidence le lien entre les différents niveaux d'accès et les fonctionnalités offertes par la plateforme.

2.5.5 Architecture de l'application

Notre projet est conçu selon une architecture rigoureuse basée sur un modèle à trois couches, enrichie par les principes du Domain-Driven Design (DDD)¹. Cette approche garantit une répartition distincte des responsabilités et encourage la flexibilité, la croissance et la maintenance de notre application. Voici un aperçu détaillé de chaque couche, intégrant les concepts du DDD :

- **La couche de présentation** : Elle constitue le principal point d'interaction entre notre application et ses utilisateurs. Cette couche vise à fournir une interface utilisateur intuitive et conviviale, permettant une interaction efficace avec les fonctionnalités de l'application. Elle inclut des éléments tels que les pages web, les interfaces mobiles, et d'autres composants visuels, offrant ainsi une expérience utilisateur immersive.
- **La couche d'accès aux données** : Cette couche assure la gestion des interactions avec la source de données de notre application. Les éléments et services de cette couche collectent et stockent les données dans la base de données ou tout autre système de stockage utilisé. En intégrant les concepts du DDD, nous veillons à une modélisation précise des entités métier, garantissant ainsi une manipulation optimale des données.
- **La couche logique métier** : Elle constitue le cœur fonctionnel de notre application, renforcée par les principes du DDD. Cette couche contient l'essentiel

1. Le Domain-Driven Design (DDD) est une approche de conception logicielle qui met l'accent sur la modélisation d'un domaine d'application en collaboration étroite avec des experts du domaine. Il aide à créer un modèle de domaine riche et significatif qui reflète avec précision les processus, règles et structures du domaine métier.

de la logique métier et les règles régissant le comportement global de l'application. Elle assure le traitement des données, l'exécution de calculs complexes, et coordonne les interactions entre la couche de présentation et la couche d'accès aux données. Grâce à cette approche, nous assurons la cohérence et la fiabilité des opérations, tout en préservant une distinction claire entre les différents domaines fonctionnels.

En adoptant cette architecture à trois couches, enrichie par les principes du DDD, nous parvenons à distinguer clairement les responsabilités entre l'interface utilisateur, la manipulation des données et les règles métier. Cette approche favorise la flexibilité, l'évolutivité et la durabilité de l'application, tout en facilitant la réutilisation du code et en améliorant les performances et la flexibilité globale du système. Ce schéma illustre la séparation des préoccupations et la répartition des responsabilités entre les différentes couches de notre application.

2.5.5.1 Diagramme de paquetage

Le diagramme de paquetage est un outil puissant pour visualiser et organiser la structure modulaire d'une application. Il permet de représenter les différents composants logiciels et leurs relations, offrant ainsi une vue d'ensemble claire de l'organisation de l'application. Ce diagramme est crucial afin de saisir la répartition des responsabilités entre les divers modules et de faciliter la communication au sein de l'équipe de développement. En observant le schéma de paquetage, il est facile de repérer les liens entre les modules, les interfaces présentées et les flux de données à travers le système. Cela favorise la création d'une architecture logicielle solide et modulaire, ce qui favorise la durabilité, la réutilisabilité et l'évolutivité de l'application.

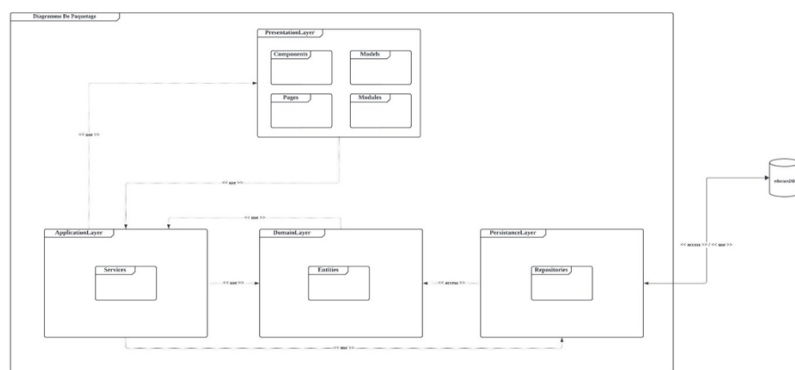


FIGURE 2.6 – Diagramme de paquetage

2.5.6 Planification des sprints

La planification de sprint est une activité essentielle qui consiste à déterminer et à ordonnancer les tâches du projet et à estimer leur charge de travail. Dans notre projet, la planification de sprint se déroule en quatre étapes principales, menées en parallèle avec notre processus de développement, afin de garantir son bon fonctionnement. Les objectifs de cette planification sont les suivants :

- **Une description du user story** : Chaque user story décrit une fonctionnalité ou une exigence du point de vue de l'utilisateur final. Elle comprend généralement un titre, une description de la fonctionnalité, les critères d'acceptation, et parfois des notes supplémentaires pour clarifier les détails.
- **Évaluer la faisabilité des objectifs** : Nous évaluons si les objectifs définis pour le sprint peuvent être atteints dans les délais prévus ou s'ils nécessitent des ajustements. Cette évaluation prend en compte les ressources disponibles, les compétences de l'équipe, et les éventuels obstacles ou risques qui pourraient survenir.
- **Suivre et communiquer l'avancement** : Nous mettons en place des mécanismes pour suivre et communiquer régulièrement l'avancement du projet. Cela inclut l'utilisation d'outils tels que des tableaux de bord, Trello ou des réunions d'avancement, permettant une visibilité constante sur l'état du projet et facilitant la prise de décision en temps réel.
- **Établir le calendrier du sprint** : Nous définissons la durée approximative du sprint, qui dans notre cas sera d'environ 4 mois. Ce calendrier inclut des jalons clés, des points de revue intermédiaires, et des phases de test, assurant une progression structurée vers les objectifs finaux.

Les différentes étapes du projet sont organisées dans un tableau de planification, tel que présenté dans le tableau ci-dessous :

TABLE 2.4 – Backlog de Produit

ID	Story	Date Début	Date Fin	Nom Sprint
S1	Inscription (ID 1) – Authentification (ID 2) – Gestion des profils (ID 3), sujets (ID 5) – Gestion des comptes (ID 6).	21/02/2024	20/03/2024	Sprint 1
S2	Gestion des cours (ID 7) - Gestion des leçons (ID 8) - Gestion des ressources (ID 9) - Gestion des quiz (ID 10) - Gestion de la confidentialité (ID 4).	21/03/2024	31/04/2024	Sprint 2
S3	Gestion des réunions (ID 12) - Gestion des forums de discussion (ID 13).	01/05/2024	30/05/2024	Sprint 3

2.6 Conclusion

Dans ce chapitre nous avons préparé notre plan de travail. Nous avons capturé les besoins fonctionnels de notre application, les rôles des membres, par la suite nous avons préparé l'architecture logique ainsi que le plan de sprints de notre projet.

Chapitre 3

Etude et réalisation du premier Sprint

3.1 Introduction

Après avoir identifié l'ensemble des besoins fonctionnels spécifiques à notre projet, nous avons entrepris de découper le projet afin de planifier efficacement les différentes phases de travail. Ce chapitre se concentre sur le premier sprint de notre projet intitulé "Authentification et Gestion compte d'utilisateur, Gestion sujet, Gestion profil". Chaque User story suivra les quatre étapes du cycle Scrum : spécification fonctionnelle, conception, implémentation et tests.

3.2 Le premier sprint

Le sprint représente l'essence de Scrum, délimitant un laps de temps pendant lequel une version améliorée du produit sera créée. Les sprints dans une release ont une durée constante et ne se chevauchent pas, signifiant qu'un sprint ne peut débuter tant que le précédent n'est pas terminé. Avant d'entamer un sprint, l'équipe Scrum doit définir clairement son objectif, formulé en termes métier pour une compréhension aisée par tous les membres. Cette définition répond à la question fondamentale : "Pourquoi réalisons-nous ce sprint?". Après discussion entre le Product Owner et l'équipe Scrum, nous avons fixé l'objectif suivant pour notre sprint initial Une fois l'objectif du sprint défini, nous sélectionnons les user stories pertinentes dans notre backlog pour composer le backlog du sprint, comme résumé dans le Tableau 5. Le sprint planning constitue une étape cruciale où nous planifions minutieusement les tâches à réaliser. Lors d'une réunion d'équipe de quatre heures, nous avons discuté en détail des objectifs et des fonctionnalités à atteindre, tout en tenant compte des besoins des utilisateurs et des contraintes du projet. Nous avons examiné les différentes user stories liées à l'authentification et à la gestion des comptes, à la gestion des sujets, et à la gestion des profils, et nous les décomposons en tâches spécifiques. Chaque membre de l'équipe s'est vu attribuer des responsabilités claires et des délais précis pour l'accomplissement de ces tâches. De plus, nous avons estimé le temps nécessaire pour chaque fonctionnalité afin d'organiser au mieux notre charge de travail. Après avoir établi l'objectif de notre sprint, nous devons maintenant déterminer quelles histoires seront incluses dans ce dernier. En d'autres termes, nous devons sélectionner les user stories pertinentes de notre backlog du produit pour les inclure dans le backlog du sprint. Le Tableau suivant récapitule ainsi le backlog de notre premier sprint.

TABLE 3.1 – Backlog Du Premier Sprint

Id	Description de la user story	Tâches effectuées	Estimation
1	En tant que candidat, instructeur ou administrateur, je veux pouvoir m'authentifier pour accéder à mon compte.	<ul style="list-style-type: none"> - Implémenter l'authentification JWT. (Backend) - Gérer les erreurs de connexion. (Backend) - Testez les endpoints avec Postman. (Backend) - Implémenter la déconnexion. (Backend) - Gérer les sessions utilisateur. (Backend) - Créer un formulaire de connexion. (Frontend) - Valider les données d'entrée. (Frontend) - Créer une page de redirection après l'authentification réussie en fonction du rôle de l'utilisateur. (Frontend) 	6 jours
2	En tant que candidat ou instructeur, je veux pouvoir m'enregistrer pour créer un compte.	<ul style="list-style-type: none"> - Gérer les erreurs lors de l'inscription. (Backend) - Enregistrer le nouvel utilisateur. (Backend) - Générer un token JWT après l'inscription. (Backend) - Tester les endpoints avec Postman. (Backend) - Créer un formulaire d'enregistrement avec les champs suivants : prénom, nom, email, mot de passe. (Frontend) - Valider les informations saisies. (Frontend) 	3 jours

Id	Description de la user story	Tâches effectuées	Estimation
3	En tant que membre, je veux pouvoir consulter mon profil utilisateur.	<ul style="list-style-type: none">- Créer un endpoint API dédié pour récupérer les informations du profil utilisateur. (Backend)- Mettre en place une logique pour récupérer les données du profil à partir de la base de données. (Backend)- Gérer les erreurs lors de la récupération des informations du profil. (Backend)- Effectuer des tests complets des endpoints avec Postman pour assurer leur bon fonctionnement et leur robustesse. (Backend)- Créer une interface utilisateur conviviale pour afficher la liste des informations du profil utilisateur. (Frontend)- Afficher les informations du profil utilisateur dans des champs éditables. (Frontend)	6 jours

Id	Description de la user story	Tâches effectuées	Estimation
4	En tant que membre, je veux pouvoir modifier mon profil utilisateur.	<ul style="list-style-type: none"> - Créer un autre endpoint API dédié pour modifier les informations du profil utilisateur. (Backend) - Mettre en place une logique pour mettre à jour les données du profil dans la base de données. (Backend) - Gérer les erreurs lors de l'opération de modification du profil. (Backend) - Effectuer des tests complets des endpoints avec Postman pour assurer leur bon fonctionnement et leur robustesse. (Backend) - Créer une interface utilisateur conviviale pour modifier des informations du profil utilisateur, y compris nom, prénom, URL LinkedIn, URL GitHub, email, bio et téléchargement d'une image de profil. (Frontend) - Ajouter la fonctionnalité de téléchargement d'images de profil, en permettant à l'utilisateur de sélectionner une image depuis l'explorateur de fichiers de son appareil. (Frontend) - Intégrer la fonctionnalité d'upload d'image de profil dans le formulaire de modification du profil, permettant à l'utilisateur de télécharger et de modifier son image de profil en un seul endroit. (Frontend) - Afficher un message d'erreur approprié en cas d'échec de la modification du profil. (Frontend) 	6 jours

Id	Description de la user story	Tâches effectuées	Estimation
5	En tant qu'administrateur, je veux pouvoir ajouter un compte candidat ou instructeur.	<ul style="list-style-type: none"> - Valider les informations saisies, y compris nom, prénom, email, mot de passe et rôle. (Backend) - Enregistrer le nouvel instructeur ou candidat. (Backend) - Gérer les erreurs lors de l'ajout. (Backend) - Tester les endpoints avec Postman. (Backend) - Créer un formulaire d'ajout d'instructeur ou candidat avec les champs : nom, prénom, email, mot de passe et rôle. (Frontend) 	2 jours
6	En tant qu'administrateur, je veux pouvoir lister les comptes d'instructeur ou de candidat existant.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer la liste des comptes d'instructeurs et de candidats existants. (Backend) - Gérer les erreurs lors de la récupération de la liste des comptes utilisateurs. (Backend) - Créer une interface utilisateur conviviale pour afficher la liste des comptes d'instructeurs et de candidats, en utilisant par exemple un tableau. (Frontend) - Afficher les détails de chaque compte utilisateur, y compris le nom, prénom, email et rôle. (Frontend) - Tester l'affichage de la liste pour vérifier son bon fonctionnement. (Frontend) 	2 jours

Id	Description de la user story	Tâches effectuées	Estimation
7	En tant qu'administrateur, je veux pouvoir supprimer un compte d'instructeur ou de candidat existant depuis la liste des utilisateurs.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer la liste des comptes d'instructeurs et de candidats existants. (Backend) - Créer une logique de traitement côté backend pour la suppression d'un compte utilisateur existant. (Backend) - Gérer les erreurs lors de la récupération de la liste des comptes utilisateurs ou lors des opérations de modification et de suppression. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Créer une interface utilisateur conviviale pour afficher la liste des comptes d'instructeurs et de candidats. (Frontend) - Ajouter des icônes associées à chaque compte utilisateur dans l'interface utilisateur de la liste. (Frontend) - Associer des actions de modification et de suppression à ces icônes, permettant ainsi aux administrateurs de modifier ou de supprimer un compte utilisateur en un clic. (Frontend) - Valider les modifications saisies avant de les envoyer au backend pour suppression. (Frontend) 	4 jours

Id	Description de la user story	Tâches effectuées	Estimation
8	En tant qu'administrateur, je veux pouvoir chercher un compte d'instructeur ou de candidat existant par nom, email, rôle.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour rechercher un compte d'instructeur ou de candidat existant par nom. (Backend) - Créer un endpoint API dédié pour rechercher un compte d'instructeur ou de candidat existant par email. (Backend) - Créer un endpoint API dédié pour rechercher un compte d'instructeur ou de candidat existant par rôle. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Afficher les résultats de la recherche dans l'interface utilisateur. (Frontend) 	2 jours
9	En tant que membre, je veux pouvoir consulter la liste des sujets de cours.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer la liste des sujets de cours existants. (Backend) - Gérer les erreurs lors de la récupération de la liste des sujets de cours. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Créer une interface utilisateur conviviale pour afficher la liste des sujets de cours. (Frontend) 	2 jours

Id	Description de la user story	Tâches effectuées	Estimation
10	En tant qu'administrateur, je veux pouvoir ajouter un sujet de cours.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour ajouter un nouveau sujet de cours. (Backend) - Valider les informations saisies pour l'ajout du sujet de cours. (Backend) - Gérer les erreurs lors de l'ajout du sujet de cours, telles que les doublons ou les données invalides. (Backend) - Tester le endpoint avec Postman pour garantir son bon fonctionnement. (Backend) - Créer une interface utilisateur pour permettre aux administrateurs d'ajouter un nouveau sujet de cours. (Frontend) - Valider les informations saisies lors de l'ajout du sujet de cours. (Frontend) 	2 jours
11	En tant qu'administrateur, je veux pouvoir modifier un sujet de cours existant depuis la liste des sujets de cours.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour la modification d'un sujet de cours existant. (Backend) - Gérer les erreurs lors des opérations de modification. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Valider les modifications saisies avant de les envoyer au backend pour modification ou suppression. (Frontend) 	3 jours

Id	Description de la user story	Tâches effectuées	Estimation
12	En tant qu'administrateur, je veux pouvoir supprimer un sujet de cours existant depuis la liste des sujets de cours.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour la suppression des sujets de cours existants. (Backend) - Gérer les erreurs lors de la suppression d'un sujet de cours existant. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Associer une action de suppression à chaque icône de suppression de sujet de cours dans l'interface utilisateur de la liste. (Frontend) - Demander une confirmation avant de supprimer un sujet de cours. (Frontend) 	2 jours
13	En tant qu'administrateur, je veux pouvoir chercher un sujet de cours existant.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour permettre la recherche de sujets de cours existants. (Backend) - Gérer les erreurs lors de l'opération de recherche et renvoyer des messages d'erreur appropriés en cas de problème. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Ajouter un champ de recherche dans l'interface utilisateur pour permettre à l'administrateur de saisir des mots-clés ou des critères de recherche. (Frontend) 	2 jours

Id	Description de la user story	Tâches effectuées	Estimation
14	En tant qu'administrateur, je veux pouvoir déconnecter	<ul style="list-style-type: none">- Implémentez une logique pour invalider la session ou supprimer le JWT Token associé à l'administrateur. (Frontend)- Ajoutez un bouton de déconnexion dans votre interface utilisateur, généralement dans le navbar ou dans un menu déroulant. (Frontend)	1 jour
15	En tant qu'administrateur, je veux pouvoir consulter les statistiques	<ul style="list-style-type: none">- Créer un endpoint API dédié pour récupérer les données statistiques. (Backend)- Gérer les erreurs lors de la récupération des statistiques et renvoyer des messages d'erreur appropriés en cas de problème. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Créer une interface utilisateur conviviale pour afficher les statistiques. (Frontend)- Afficher les différentes données statistiques de manière claire et compréhensible, en utilisant des graphiques, des tableaux ou d'autres visualisations appropriées. (Frontend)	2 jours

Passons maintenant au vif de notre sujet les activités et le cycle de développement. Dans un sprint nous pouvons dégager quatre activités principales qui sont la spécification fonctionnelle, la conception, le codage et le test. Tout au long de ce sprint, nous respectons ces activités pour construire le plan de notre travail.

3.2.1 Spécification fonctionnelle

La spécification fonctionnelle dans notre cas se traduit par le diagramme de cas d'utilisation d'UML et la description textuelle de ces derniers.

3.2.1.1 Raffinement de cas d'utilisation du sprint 1

Le diagramme de cas d'utilisation du sprint 1 modélise les différentes fonctionnalités du sprint. Il présente trois acteurs héritant de l'utilisateur qui ont présenté par l'acteur membre, qui partagent des fonctionnalités communes telles que :

- **S'authentifier**
- **Gérer le profil**

3.2.2 Diagramme de cas d'utilisation détaillé du Sprint 1

Dans cette partie, nous nous intéressons à réaliser une description textuelle détaillée de cas d'utilisation mentionnés dans le sprint 1. En fait, le diagramme de cas d'utilisation détaillé offre une représentation graphique approfondie des fonctionnalités et des interactions entre les acteurs et le système. Il permet de décrire en détail les principales fonctionnalités nécessaires aux utilisateurs du système. La figure suivante présente le diagramme de cas d'utilisation détaillé de ce premier sprint :

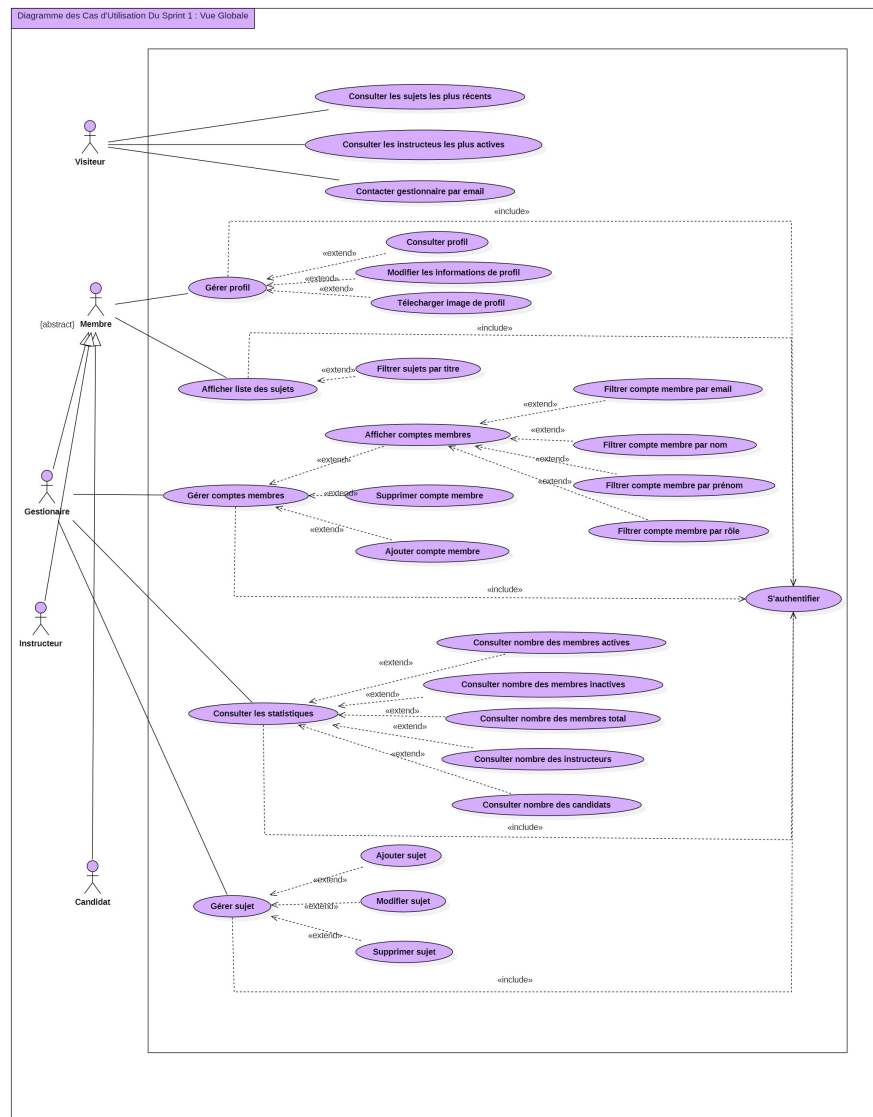


FIGURE 3.1 – Diagramme De cas D'Utilisation Du Premier Sprint

Ce diagramme de cas d'utilisation détaillé montre que tous les membres peuvent modifier leurs profils après s'être authentifiés. Seul l'administrateur a le privilège de créer de nouveaux utilisateurs et de leur attribuer un profil et de gérer des sujets des cours. L'authentification est une étape préalable nécessaire avant toute action sur le système. En utilisant le diagramme de cas d'utilisation détaillé du Sprint 1, nous avons pu fournir une représentation visuelle complète des fonctionnalités attendues et des interactions entre les acteurs et le système. Cela nous a permis de mieux comprendre les besoins des utilisateurs et de garantir le développement d'un système répondant à leurs attentes spécifiques.

3.2.3 Description textuelle de cas d'utilisation

Il est recommandé de rédiger une description textuelle de chaque cas d'utilisation afin de les détailler pour bien comprendre tous les processus. Une description textuelle classique se compose par

Partie 1 : Identification

- **Titre** : Nom du cas d'utilisation
- **Acteurs** : Descriptions des acteurs principaux et secondaires

Partie 2 : Description des scénarios.

- **Prés condition** : Elles décrivent l'état du système avant que le cas d'utilisation puisse être déclenché.
- **Les scénarios** :
 1. **Scénario nominal** : Correspond à un déroulement normal d'un cas d'utilisation.
 2. **Scénarios alternatifs** : Variantes du scénario normal.
 3. **Scénarios d'exceptions** : Décrivent ce qui se passe lors d'une erreur.
- **Post condition** : Elles décrivent l'état du système après l'issue de chaque scénario.

Partie 3 : Exigence non fonctionnelle

- Exigences non fonctionnelles spécifiques au cas d'utilisation.

La partie 3 peut être omise, mais si elle est présente, elle permet de préciser des spécifications non fonctionnelles (fréquence, fiabilité, type d'interface homme-. Machine...). La description détaillée de chaque cas d'utilisation clarifie les processus et facilite l'identification des parties redondantes. En examinant attentivement les processus, on peut repérer les sections répétitives et les décomposer en cas d'utilisation plus spécifiques, favorisant ainsi la modularité du système.

TABLE 3.2 – Description textuelle du cas d'utilisation « Authentification »

Cas d'utilisation	Authentification
Acteur principal	Gestionnaire, candidats et instructeurs
Précondition	Utilisateur non connecté. L'adresse électronique et le mot de passe doivent être précis et valides.

Postcondition	Acteur authentifié et accès à l'accueil personnel de l'utilisateur
Scénario Nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page de connexion de l'application 2. Le système affiche le formulaire d'authentification 3. L'utilisateur saisit son adresse e-mail et son mot de passe. 4. L'utilisateur soumet le formulaire. 5. L'application valide les informations d'identification. 6. Si les informations sont valides, un JWT est généré, contenant les détails de l'utilisateur. 7. Le JWT est stocké localement et envoyé dans l'en-tête des requêtes pour prouver l'identité. 8. À chaque demande, l'application vérifie la validité du JWT et autorise l'accès à l'accueil personnel de l'utilisateur. 9. L'accueil personnel de l'utilisateur s'affiche, présentant ses fonctionnalités et informations spécifiques.
Scénario d'exception (A1)	A1. Champs vide : Un message d'erreur est affiché si des champs requis sont vides.
Scénario d'exception (A2)	A2. Informations de connexion incorrectes : L'application indique une erreur si les informations d'identification sont incorrectes. L'utilisateur est invité à corriger les informations et peut réessayer.

Le tableau ci-dessous explique le processus de suppression d'un utilisateur par un administrateur permettant à ce dernier de valider et d'effectuer la suppression d'un utilisateur existant dans le système.

TABLE 3.3 – Description textuelle du cas d'utilisation « Ajouter Utilisateur »

Cas d'utilisation	Ajouter Utilisateur
Acteur principal	Gestionnaire

Précondition	Gestionnaire est authentifié et possède les autorisations nécessaires pour ajouter des membres.
Postcondition	Nouveau membre est ajouté avec succès à l'application, avec le rôle spécifié.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à la fonctionnalité d'ajout de membre dans l'espace d'administration du la plateforme. 2. Le système affiche un formulaire pour saisir les informations du membre à ajouter, telles que le nom, le prénom, l'adresse e-mail, le type de membre (instructeur, candidat, autre administrateur), mot de passe, etc. 3. L'administrateur saisit les informations requises. 4. Après avoir saisi les informations requises, l'administrateur valide le formulaire. 5. Le système vérifie les données saisies et les valide. Il peut s'agir de vérifier si l'adresse e-mail est unique dans le système ou si les champs obligatoires sont remplis. 6. Si toutes les données saisies sont valides, le système enregistre les informations de membre dans la base de données. 7. Le système affiche un message de confirmation indiquant que le membre a été ajouté avec succès. 8. Une fois le membre ajouté, le système crée automatiquement un profil pour cet membre, en utilisant les informations fournies lors de l'ajout, telles que le nom, le prénom et le type d'utilisateur. 9. La page de la liste des membres est rechargée en ajoutant le membre ajouté.
Scénario alternatif (SA1)	<p>SA1 : L'administrateur laisse un ou plusieurs champs obligatoires vides et tente de valider le formulaire</p> <p>SA1 commence au point 4 du scénario nominal.</p>

		1. Le système détecte des champs obligatoires vides et affiche un message d'erreur indiquant que tous les champs doivent être remplis.
Scénario (SA2)	alternatif	<p>SA2 : L'administrateur saisit les informations de membre, y compris l'adresse e-mail, mais commet une erreur en saisissant une adresse e-mail avec un format incorrect (par exemple, sans "@") ou le nom est inférieur à 2 caractères</p> <p>SA2 commence au point 5 du scénario nominal.</p> <ol style="list-style-type: none"> 1. Le système détecte ces erreurs lors de la vérification des données. 2. Le système affiche des messages d'erreur correspondants à chaque champ erroné. 3. L'administrateur corrige les champs erronés. 4. Une fois que les erreurs ont été corrigées, l'administrateur soumet à nouveau le formulaire. 5. Le système reprend le processus de vérification des données, comme il est décrit dans les étapes du scénario nominal.

Le tableau ci-dessous explique le processus de suppression d'un utilisateur par un administrateur permettant à ce dernier de valider et d'effectuer la suppression d'un utilisateur existant dans le système.

TABLE 3.4 – Description textuelle du cas d'utilisation « Supprimer un membre »

Cas d'utilisation	Supprimer un membre
Acteur principal	Gestionnaire
Précondition	Gestionnaire doit être authentifié et avoir les droits nécessaires pour accéder à la fonctionnalité de supprimer des utilisateurs. L'utilisateur que l'administrateur souhaite modifier doit déjà être enregistré dans le système.
Postcondition	Utilisateur supprimé
Scénario Nominal	1. L'administrateur accède à la page de liste des utilisateurs.

	<p>2. L'administrateur recherche le membre qu'il souhaite supprimer.</p> <p>3. Le système affiche les informations actuelles du membre sélectionné dans un formulaire modifiable avec un bouton qui permet la suppression.</p> <p>4. L'administrateur clique sur le bouton de suppression.</p> <p>5. Avant de procéder à la suppression, le système affiche une demande de confirmation supplémentaire pour éviter les suppressions accidentelles ou non intentionnelles, puisque la suppression est une action irréversible et que toutes les données associées à cet utilisateur seront définitivement perdues.</p> <p>6. Une fois la suppression confirmée, le système effectue la suppression du membre dans la base de données.</p> <p>7. Toutes les informations liées au membre, telles que son profil et ses données associées, sont supprimées de manière permanente.</p> <p>8. Le système affiche un message indiquant que l'utilisateur a été supprimé avec succès.</p> <p>9. La page de la liste des utilisateurs est rechargée en enlevant l'utilisateur supprimé.</p>
--	---

Le tableau suivant explique comment le membre peut gérer ses informations du profil. En fait, le membre authentifié peut accéder et gérer certaines informations spécifiques de son profil.

TABLE 3.5 – Description textuelle du cas d'utilisation « Modifier informations de profil »

Cas d'utilisation	Modifier informations de profil
Acteur principal	Gestionnaire, candidats et instructeurs
Précondition	Le membre est connecté à son compte sur la plateforme.
Postcondition	Informations modifiées

Scénario Nominal	<ol style="list-style-type: none"> 1. Le membre visualise le sidebar de sa page d'accueil. 2. Le membre identifie et sélectionne l'option "profil" dans le sidebar. 3. Le système redirige le membre vers la section de gestion de son profil en affichant les informations personnelles actuelles du membre dans un formulaire modifiable. 4. Le membre modifie les informations nécessaires dans le formulaire (nom, prénom, adresse e-mail, etc.). 5. Une fois les modifications effectuées, le membre valide le formulaire de modification. 6. Le système vérifie les données modifiées et les enregistre dans la base de données. 7. Le système affiche un message de confirmation indiquant que les modifications ont été enregistrées avec succès. 8. La page du profil du membre est rechargée de façon synchronisée avec chaque champ modifié. 9. Si le membre a modifié son adresse e-mail, un nouveau JWT est généré et stocké dans le stockage local du navigateur. 10. Le nouveau JWT est envoyé dans l'en-tête de la requête pour prouver l'identité du membre lors des prochaines interactions avec la plateforme.
Scénario alternatif (SA1)	<p>SA1 : Les nouvelles informations saisies sont incorrectes ou invalides.</p> <p>SA1 commence au point 5 du scénario nominal.</p> <ol style="list-style-type: none"> 1. Le système détecte que les nouvelles informations saisies sont incorrectes ou invalides. 2. Le système affiche un message d'erreur spécifique, indiquant la nature des erreurs ou des informations invalides.

	3. Le membre est invité à corriger les champs concernés dans le formulaire. Une fois les informations valides, le système procède aux étapes du scénario nominal.
--	---

Le tableau suivant explique comment Le membre peut télécharger sa photo de profil.

TABLE 3.6 – Description textuelle du cas d'utilisation « Télécharger photo de profil »

Cas d'utilisation	Télécharger photo de profil
Acteur principal	Gestionnaire, candidats et instructeurs
Précondition	L'utilisateur est connecté à son compte sur la plateforme. L'utilisateur accède à la section de gestion de son profil où se trouve l'option pour mettre à jour l'image de profil.
Postcondition	L'image de profil de l'utilisateur est mise à jour avec la nouvelle image téléchargée.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la section de gestion de son profil où se trouve l'option pour mettre à jour l'image de profil. 2. L'utilisateur sélectionne l'option pour télécharger une nouvelle image. 3. Le système affiche une fenêtre ou une interface permettant à l'utilisateur de choisir un fichier image depuis son appareil. 4. L'utilisateur sélectionne le fichier image à télécharger. 5. Une fois le fichier sélectionné, l'utilisateur confirme son choix. 6. L'utilisateur clique sur le bouton pour confirmer la mise à jour. 7. La page profil de l'utilisateur est rechargée en affichant la photo du profil.
Scénario alternatif (SA1)	SA1 : Erreur lors du téléchargement de l'image

	<p>SA1 commence juste après l'étape 4 du scénario nominal.</p> <ol style="list-style-type: none"> 1. Le système détecte une erreur lors du téléchargement ou du traitement de l'image. 2. Le système affiche un message d'erreur spécifique indiquant la nature du problème. 3. L'utilisateur est invité à réessayer le processus de téléchargement. Le processus se poursuit selon les étapes du scénario nominal après que l'erreur a été résolue.
--	---

Le tableau suivant explique la procédure de déconnexion de l'utilisateur ou le système supprime le JWT.

TABLE 3.7 – Description Textuelle Du Cas D'Utilisation « Déconnexion »

Cas d'utilisation	Déconnexion
Acteur principal	Gestionnaire, candidats et instructeurs
Prés condition	Le membre est connecté.
Post condition	Le membre est déconnecté et redirigé vers la page de connexion.
Scénario Nominal	<ol style="list-style-type: none"> 1. Le membre souhaite se déconnecter de l'application. 2. Le membre clique sur le bouton de déconnexion qui est placé dans le navbar. 3. Le système effectue les actions appropriées pour supprimer le JWT Token stocké dans le local Storage du navigateur. 4. Le membre est redirigé vers la page de connexion.

Le tableau suivant explique la procédure d'ajouter un sujet de cours.

TABLE 3.8 – Description textuelle du cas d'utilisation « Ajouter sujet »

Cas d'utilisation	Ajouter sujet
Acteur principal	Gestionnaire
Préconditions	L'administrateur doit être authentifié et avoir les droits nécessaires pour accéder à la fonctionnalité d'ajouter un sujet.
Postconditions	Sujet ajouté
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à la fonctionnalité d'ajout de sujet dans l'interface d'administration. 2. Le système affiche un formulaire pour saisir les détails du nouveau sujet, tels que le titre, l'image, etc. 3. L'administrateur saisit les informations requises pour le sujet. 4. Une fois les informations saisies, l'administrateur valide le formulaire. 5. Le système vérifie les données saisies et les enregistre dans la base de données. 6. Le système affiche un message de confirmation indiquant que le sujet a été ajouté avec succès. 7. La page est rechargée en affichant le nouveau sujet ajouté.
Scénario alternatif (SA1)	<p>SA1 : Sujet déjà existant</p> <p>SA1 commence après l'étape 4 du scénario nominal.</p> <ol style="list-style-type: none"> 1. Le système détecte que le sujet que l'administrateur tente d'ajouter existe déjà dans la base de données. 2. Le système affiche un message d'erreur indiquant que le sujet existe déjà. 3. L'administrateur peut soit modifier le sujet existant, soit choisir un autre titre pour le nouveau sujet.

Le tableau suivant explique la procédure Modifier un sujet de cours.

TABLE 3.9 – Description textuelle du cas d'utilisation « Modifier sujet »

Cas d'utilisation	Modifier sujet
Acteur principal	Gestionnaire
Préconditions	Le Gestionnaire doit être authentifié et avoir les droits nécessaires pour accéder à la fonctionnalité de modifier un sujet.
Postconditions	Sujet modifié
Scénario Nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à la fonctionnalité de modification de sujet dans l'espace d'administration de la plateforme. 2. Le système affiche une liste des sujets existants dans un formulaire modifiable. 3. L'administrateur recherche et sélectionne le sujet qu'il souhaite modifier. 4. Le système affiche les détails du sujet sélectionné dans un formulaire modifiable. 5. L'administrateur effectue les modifications nécessaires sur le sujet (titre, image, etc.). 6. Une fois les modifications terminées, l'administrateur valide le formulaire de modification. 7. Le système vérifie les données modifiées et les enregistre dans la base de données. 8. Le système affiche un message de confirmation indiquant que les modifications ont été enregistrées avec succès. 9. La page est rechargée en affichant les nouvelles modifications.
Scénario alternatif (SA1)	<p>SA1 : Sujet introuvable</p> <p>SA1 commence après l'étape 3 du scénario nominal.</p> <ol style="list-style-type: none"> 1. Lorsque l'administrateur tente de modifier un sujet, le système ne parvient pas à trouver le sujet correspondant dans la base de données.

		2. Le système affiche un message d'erreur indiquant que le sujet est introuvable. 3. L'administrateur peut réessayer en vérifiant le titre du sujet ou en contactant le support technique.
Scénario (SA2)	alternatif	SA2 : Sujet déjà existant SA2 commence après l'étape 5 du scénario nominal. 1. Le système détecte que le sujet que l'administrateur tente d'ajouter existe déjà dans la base de données. 2. Le système affiche un message d'erreur indiquant que le sujet existe déjà. 3. L'administrateur peut soit modifier le sujet existant, soit choisir un autre titre pour le nouveau sujet.

Le tableau suivant explique la procédure supprimer un sujet de cours.

TABLE 3.10 – Description textuelle du cas d'utilisation « Supprimer sujet »

[HTML]EFEFEF d'utilisation	Cas	Supprimer sujet
Acteur principal		Gestionnaire
Précondition		Le Gestionnaire doit être authentifié et avoir les droits nécessaires pour accéder à la fonctionnalité de supprimer un sujet.
Postcondition		Sujet Supprimé
Scénario Nominal		1. L'administrateur accède à la fonctionnalité de suppression de sujet dans l'espace d'administration de la plateforme. 2. Le système affiche une liste des sujets existants dans un formulaire modifiable. 3. L'administrateur recherche et sélectionne le sujet qu'il souhaite modifier.

		<p>4. Le système affiche les détails du sujet sélectionné dans un formulaire modifiable.</p> <p>5. L'administrateur effectue les modifications nécessaires sur le sujet (titre, image, etc.).</p> <p>6. Une fois les modifications terminées, l'administrateur valide le formulaire de modification.</p> <p>7. Le système vérifie les données modifiées et les enregistre dans la base de données.</p> <p>8. Le système affiche un message de confirmation indiquant que les modifications ont été enregistrées avec succès.</p> <p>9. La page est rechargée en affichant les nouvelles modifications.</p>
Scénario (SA1)	alternatif	<p>SA1 : Sujet introuvable</p> <p>SA1 commence après l'étape 3 du scénario nominal.</p> <p>1. Lorsque l'administrateur tente de modifier un sujet, le système ne parvient pas à trouver le sujet correspondant dans la base de données.</p> <p>2. Le système affiche un message d'erreur indiquant que le sujet est introuvable.</p> <p>3. L'administrateur peut réessayer en vérifiant le titre du sujet ou en contactant le support technique.</p>
Scénario (SA2)	alternatif	<p>SA2 : Sujet déjà existant</p> <p>SA2 commence après l'étape 5 du scénario nominal.</p> <p>1. Le système détecte que le sujet que l'administrateur tente de modifier existe déjà dans la base de données.</p> <p>2. Le système affiche un message d'erreur indiquant que le sujet existe déjà.</p>

3. L'administrateur peut soit modifier le sujet existant, soit choisir un autre titre pour le nouveau sujet.

3.2.4 La modélisation conceptuelle

3.2.4.1 Diagramme de classes

Dans le contexte de cette partie, le diagramme de classes est utilisé pour représenter les différentes classes et entités impliquées dans le processus de l'authentification et la gestion des comptes, profils, sujets. Le diagramme de classes permet de visualiser la structure et les relations entre ces classes, ce qui peut aider à comprendre comment les différentes entités du système interagissent dans ce processus.

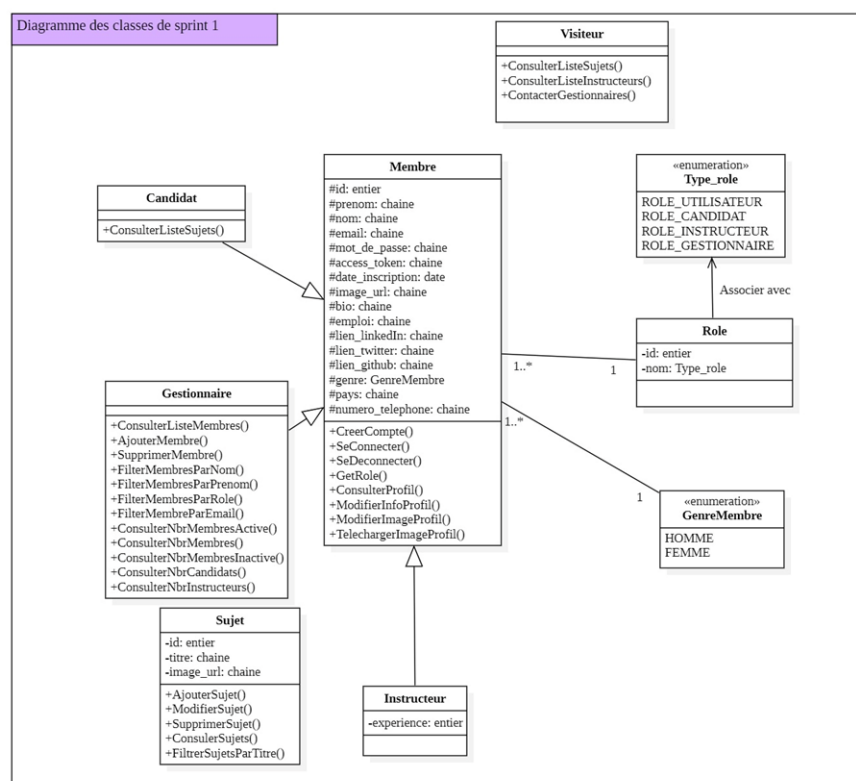


FIGURE 3.2 – Diagramme De Classes Du Premier Sprint

Ce diagramme de classes peut compléter la description textuelle en fournissant une représentation visuelle de la structure du système et des relations entre les classes impliquées dans le processus décrit. Cela facilite la compréhension du fonctionnement du système et peut aider à identifier les points d'interaction entre les différentes entités

3.2.4.2 Diagramme de séquence détaillé

Le diagramme de séquence est un outil de modélisation utilisé pour représenter l'interaction entre les objets d'un système en mettant en évidence la séquence chronologique des messages échangés entre eux. Cela permet de comprendre le flux de contrôle et de communication au sein du système. Pour plus de simplicité, nous positionnons les acteurs principaux du côté gauche de la figure, tandis que les acteurs secondaires éventuels se trouvent du côté droit du système. L'objectif est de décrire le déroulement des actions entre les acteurs ou les objets impliqués.

— Diagramme de séquence associé à « Authentification »

Le diagramme de séquence illustré dans la figure suivante représente le processus d'authentification de membre dans le cas d'utilisation "S'authentifier".

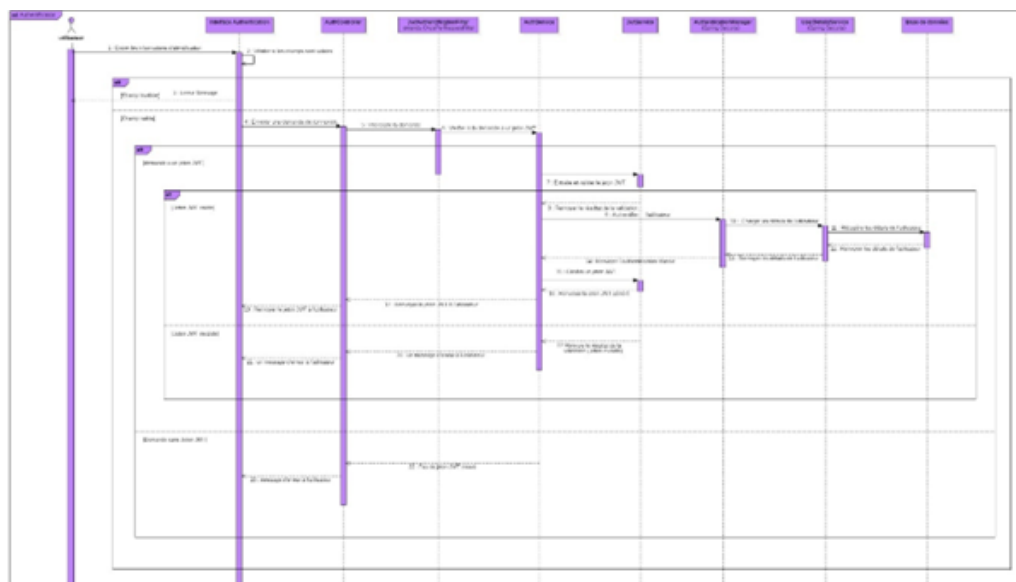


FIGURE 3.3 – Diagramme De Séquence Du Cas D'Utilisation « S'authentifier »

Après avoir ouvert la page de connexion de l'application, le visiteur est confronté à un formulaire d'authentification. Il saisit alors ses identifiants (adresse e-mail et mot de passe) et les soumet. Le système procède à la validation de ces informations, probablement en les confrontant à une base de données via Spring Boot Security. Si les identifiants sont corrects, l'application émet un jeton d'authentification (JWT) contenant des détails sur le membre, tels que son identifiant et son rôle. Ce jeton est stocké localement dans le navigateur et envoyé dans l'en-tête de chaque requête future pour attester l'identité du membre. À chaque requête, l'application vérifie la validité du JWT pour s'assurer de son authenticité et de sa non-expiration. Si le JWT est valide, le membre est redirigé vers son espace personnel où il peut accéder

aux fonctionnalités et informations propres à son rôle. Une fois ce processus achevé, le membre est authentifié et peut utiliser l'application conformément à ses droits d'accès. Si des données incorrectes sont fournies lors de la connexion, un message d'erreur s'affiche, invitant le membre à corriger ses informations et à réessayer.

— Diagramme de séquence associé à « Ajouter Membre »

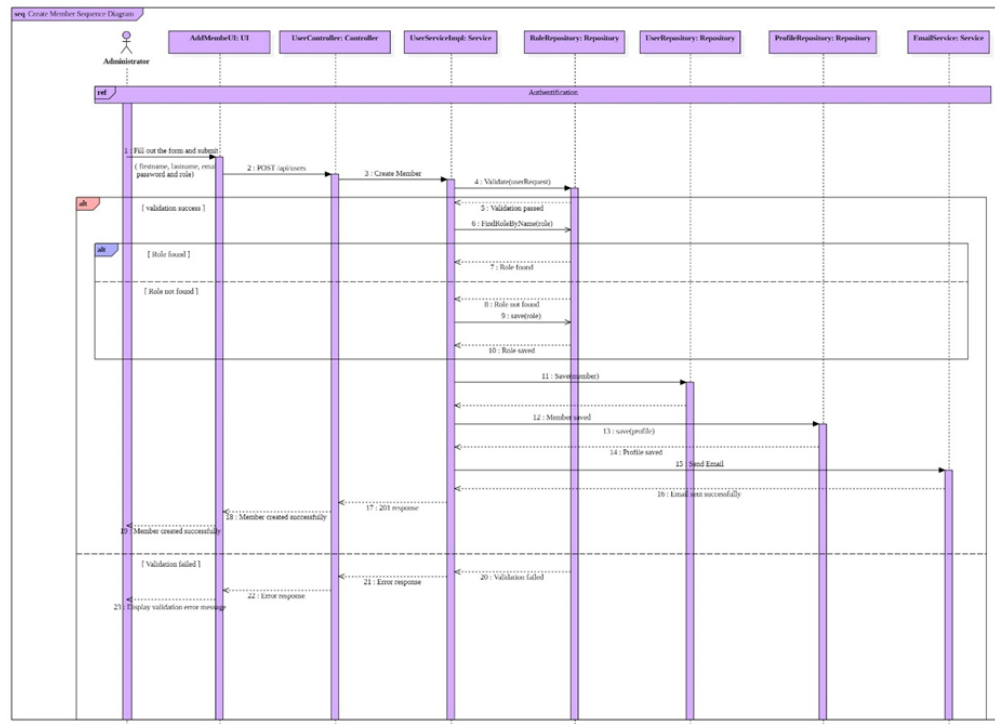


FIGURE 3.4 – Diagramme De Séquence Du Cas D'Utilisation « Ajouter Un Membre »

Le diagramme de séquence ci-dessus détaille le processus d'ajout d'un membre dans le système, mettant en lumière les différentes étapes et les responsabilités de chaque composant. L'interaction débute avec l'acteur gestionnaire, qui utilise l'interface d'ajout de membre pour fournir les informations requises. Cette interaction initiale est représentée par la transition de l'acteur gestionnaire vers l'interface d'ajout de membre. Ensuite, une requête d'ajout de membre est envoyée du composant de l'interface utilisateur, représenté par l'interface AddMember, vers le UserController. Ce dernier prend en charge la réception de la requête et la transmission des données au UserServiceImpl pour traitement. Le UserServiceImpl, en tant que composant central de la logique métier, est responsable de plusieurs étapes cruciales du processus. Il commence par valider les données fournies pour le nouveau membre en faisant appel au RoleRepository. Cette validation est essentielle pour garantir l'intégrité et

la cohérence des données du système. Si la validation des données est réussie, le `UserServiceImpl` procède à la recherche du rôle associé au nouveau membre dans le `RoleRepository`. Si le rôle est trouvé, l'utilisateur est enregistré dans le `UserRepository`, et son profil associé est sauvegardé dans le `ProfileRepository`. Une fois l'enregistrement de l'utilisateur et de son profil effectué avec succès, le composant `EmailService` est sollicité pour envoyer un email de confirmation au nouveau membre. L'`EmailService` joue ainsi un rôle crucial dans l'interaction système en assurant la communication avec les utilisateurs. Après l'envoi réussi de l'email de confirmation, le `UserServiceImpl` envoie une réponse HTTP 201 Created au `UserController` pour indiquer que le processus d'ajout de membre s'est déroulé avec succès. Enfin, une notification ou un message de confirmation peut être affiché à l'administrateur pour l'informer que l'ajout du membre a été réussi.

— Diagramme de séquence associé à « Consulter Profil »

Le diagramme de séquence suivant présente le processus de changement des consulter du compte par le membre.

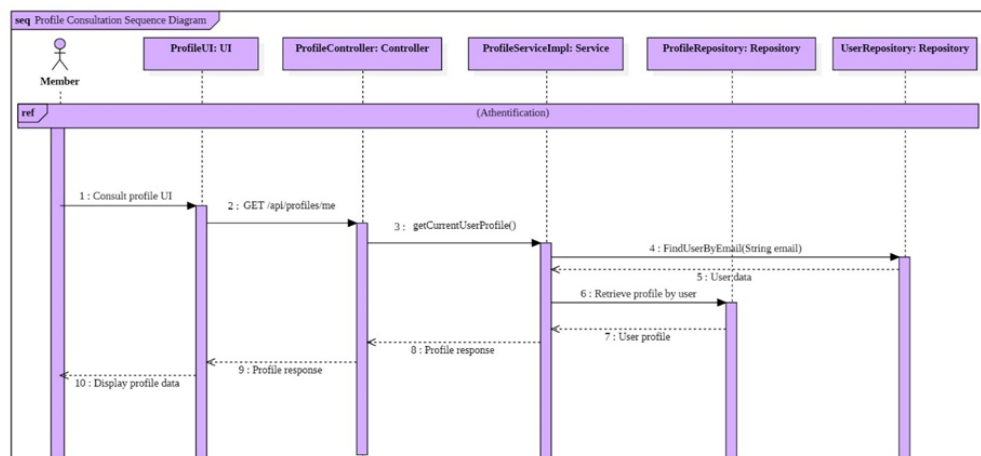


FIGURE 3.5 – Diagramme De Séquence Du Cas D'Utilisation « Consulter Profil »

Ce diagramme de séquence illustre le flux d'interaction lorsqu'un membre interagit avec l'interface utilisateur de profil (Profile UI) pour consulter son propre profil. Initialement, le membre interagit avec l'interface utilisateur de profil, déclenchant ainsi une demande de consultation de profil. Cette demande est transmise au `ProfileController`, qui agit comme un point d'entrée pour les requêtes liées au profil. Le `ProfileController`, à son tour, invoque le `ProfileServiceImpl` pour obtenir le profil actuel du membre.

Dans le `ProfileServiceImpl`, la méthode `getCurrentUserProfile` est appelée pour récupérer les détails du profil de membre actuellement authentifié. Pour cela, le

service utilise le UserRepository pour trouver le membre correspondant dans la base de données en fonction de son adresse e-mail. Une fois le membre trouvé, le ProfileServiceImpl fait appel au ProfileRepository pour récupérer le profil associé à cet utilisateur.

Une fois le profil récupéré avec succès, il est renvoyé au ProfileController, qui le transmet ensuite à l'interface utilisateur de profil.

— **Diagramme de séquence associé à « Modifier informations du profil »**

Après une authentification réussie, le membre accède à la page d'accueil et clique sur le bouton " View profile". Un formulaire apparaît permettant au membre de modifier les données existantes. Ensuite, le membre clique sur le bouton 'update' pour envoyer les données saisies au contrôleur. Le contrôleur vérifie la validité des nouvelles informations et effectue la mise dans la base de données.

— **Diagramme de séquence associé à « Consulter Statistiques »**

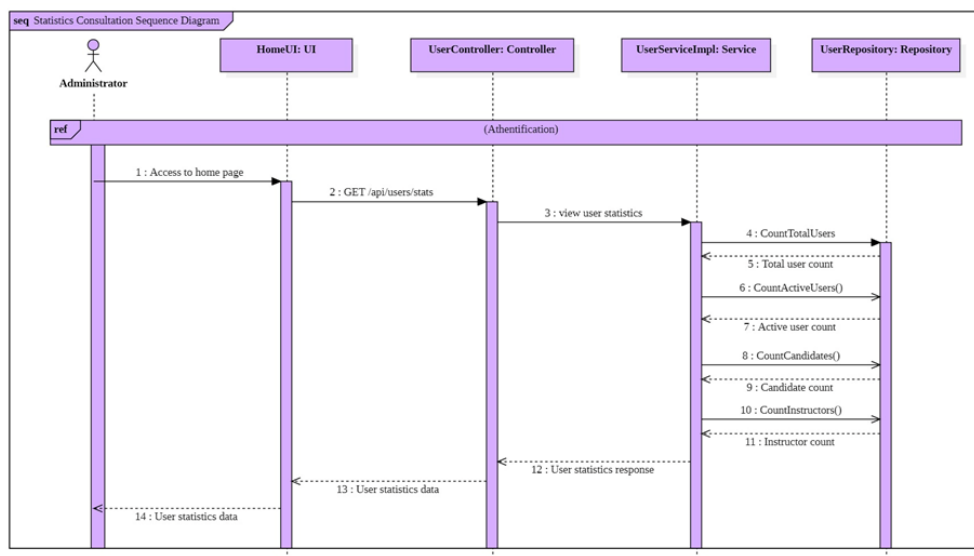


FIGURE 3.6 – Diagramme De Séquence Du Cas D'Utilisation « Consulter Statistiques »

Dans ce diagramme de séquence, nous observons le processus d'accès aux statistiques des membres par un gestionnaire lorsqu'il accède à sa page d'accueil. L'interaction commence lorsque le gestionnaire accède à sa page d'accueil via l'interface utilisateur, représentée par la classe "UI". Une fois sur sa page d'accueil, l'option pour consulter les statistiques des membres est automatiquement chargée et affichée. Cette action déclenche alors l'envoi automatique d'une requête depuis l'interface utilisateur

vers le contrôleur utilisateur ("UserController") pour récupérer les statistiques des membres. Le "UserController" reçoit cette demande et appelle immédiatement le service utilisateur ("UserService") pour récupérer les statistiques nécessaires. Le service utilisateur interagit avec le référentiel utilisateur ("UserRepository") pour accéder aux données des membres. Une fois que les statistiques des membres sont extraites avec succès de la base de données, elles sont renvoyées par le service utilisateur au "UserController". Le "UserController" transmet ensuite les statistiques des membres à l'interface utilisateur pour qu'elles soient affichées sur la page d'accueil du gestionnaire. Ainsi, lorsque le gestionnaire accède à sa page d'accueil, les statistiques des membres sont automatiquement chargées et présentées pour sa consultation.

3.2.5 Codage

Dans le développement de notre application e-learning, la phase de codage joue un rôle essentiel, car elle permet de mettre en œuvre les spécifications fonctionnelles définies lors des étapes précédentes. Le déroulement de cette étape se fait de manière itérée et méthodique, en suivant les bonnes pratiques de développement logiciel.

3.2.6 Réalisation

Après avoir terminé la phase de conception, nous commençons la réalisation de notre premier sprint. Les captures d'écran suivantes présentent les différentes interfaces réalisées pendant le premier sprint. Ceci permet d'illustrer certains éléments clés de notre plateforme educare La figure suivante représente l'interface de l'authentification qui doit être consultée par tous les utilisateurs de notre application afin d'accéder à notre application :

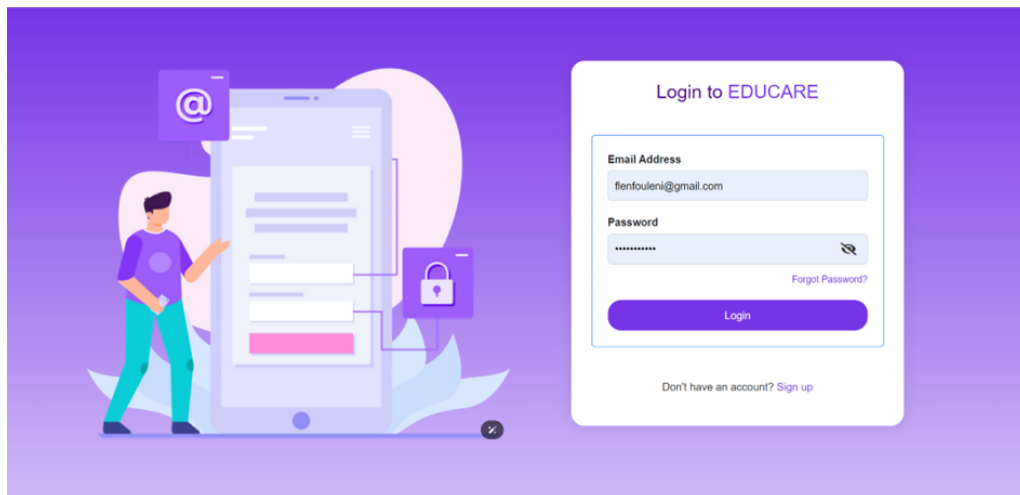


FIGURE 3.7 – L'interface d'authentification

L'interface d'authentification comprend les champs d'adresse e-mail et de mot de passe permettant de se connecter.

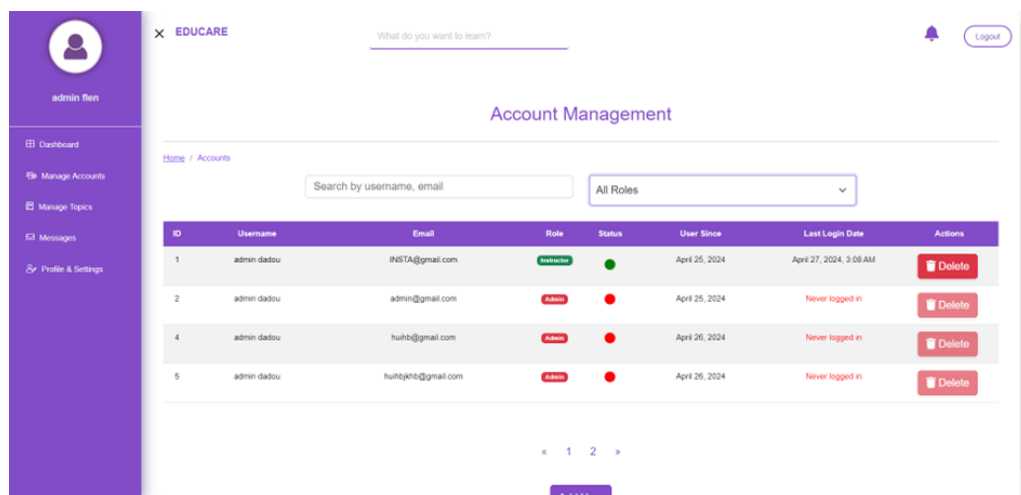


FIGURE 3.8 – L'interface d'ajout d'utilisateurs

L'interface d'ajout d'utilisateurs, présentée dans la figure suivante, est réservée exclusivement à l'administrateur.

3.2.7 Tests

Les tests jouent un rôle crucial dans l'assurance de la qualité des logiciels. Ils permettent de vérifier que le système ou le composant répond à ses spécifications et identifient les différences entre les résultats spécifiés et les résultats obtenus. Dans notre approche de développement, nous intégrons les tests unitaires et les tests

d'acceptance, notamment le Test-Driven Development (TDD) et l'Acceptance Test-Driven Development (ATDD) , pour garantir plusieurs avantages.

3.2.8 Les tests unitaires

Les tests unitaires impliquent la rédaction des tests avant même de les programmer, ce qui permet aux développeurs d'assurer le bon fonctionnement de chaque élément logiciel. Cette méthode contribue à rendre le code plus clair et garantit sa solidité lors de futures manipulations. En ce qui concerne notre projet d'apprentissage en ligne, nous avons décidé de mettre en œuvre des tests unitaires afin de vérifier les fonctionnalités essentielles du premier sprint, telles que l'ajout, la modification et la suppression des sujets, des cours et des leçons. Ainsi, nous avons employé JUnit, une bibliothèque de tests unitaires Java couramment employée dans l'environnement Spring Boot. Cette méthode nous a donné l'opportunité de profiter de la solidité de Spring Boot dans le backend, tout en bénéficiant de sa facilité d'utilisation et de sa compatibilité.

Nous avons choisi de tester les mappers, les services et les contrôleurs car ces éléments sont essentiels à notre application. Les tests des mappers garantissent que la transformation des données est adéquate, assurant ainsi la cohérence entre les différentes couches de l'application. Les tests des services vérifient la logique métier et les opérations sur les données, tandis que les tests des contrôleurs assurent que les requêtes HTTP sont gérées de manière adéquate et que les réponses aux clients répondent aux attentes. En étudiant ces éléments, nous veillons à ce que notre application fonctionne de manière fiable, respecte les normes et offre une expérience utilisateur fluide.

— Test unitaire de Mapper

Les tests de Mapper assurent que la conversion des données entre les objets de domaine et les objets de transfert de données (DTO) est effectuée correctement.

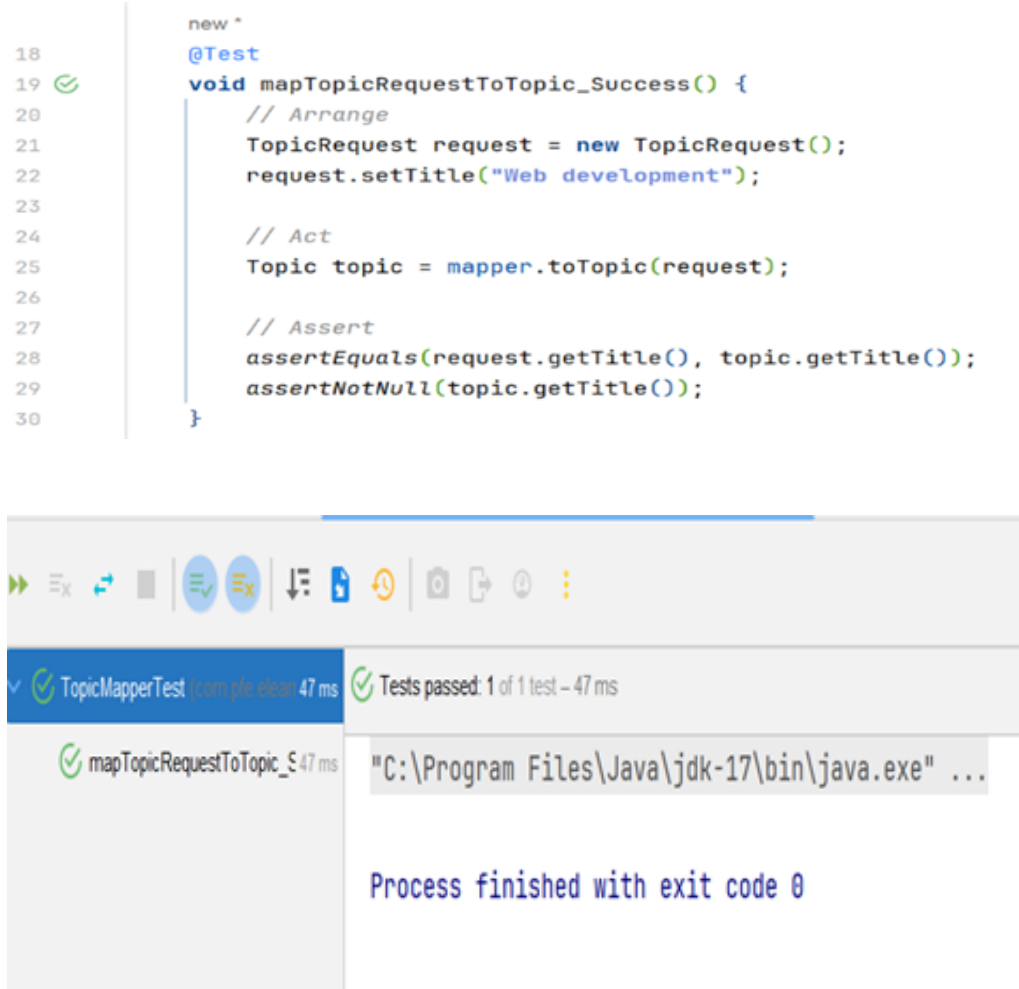


FIGURE 3.9 – Résultat réussi du mapping de la requête de sujet vers la réponse de sujet

Ces captures d'écran montrent le succès du test `mapTopicRequestToTopic_success()`. *Cet est validation*

— Test unitaire de service

Les tests des services vérifient la logique métier et les opérations sur les données, garantissant que les fonctionnalités du système répondent aux exigences spécifiées.

```

@Test
void createTopic_SuccessfullyCreatesTopic() {
    // Given
    TopicRequest request = new TopicRequest( title: "Web development", imageFile: null);
    Topic topic = new Topic( title: "Web development");
    Topic savedTopic = new Topic( title: "Web development");
    savedTopic.setId(1);

    // Mock the calls
    when(topicMapper.toTopic(request)).thenReturn(topic);
    when(topicRepository.save(topic)).thenReturn(savedTopic);
    when(topicMapper.toTopicResponse(savedTopic)).thenReturn(new TopicResponse(
        id: 1,
        title: "Web development",
        imageFile: null, // Set imageFile to null as we're not testing this field here
        LocalDateTime.now(),
        LocalDateTime.now()
    ));
    doNothing().when(validator).validate(request);

    // When
    TopicResponse response = topicService.createTopic(request, imageUrl: null);

    // Then
    assertNotNull(response);
    assertEquals(request.getTitle(), response.getTitle());
    assertEquals(savedTopic.getId(), response.getId());

    // Verify interactions
    verify(validator).validate(request);
    verify(topicMapper).toTopic(request);
    verify(topicRepository).save(topic);
    verify(topicMapper).toTopicResponse(savedTopic);
}

```

FIGURE 3.10 – Test unitaire de création réussie d’un sujet

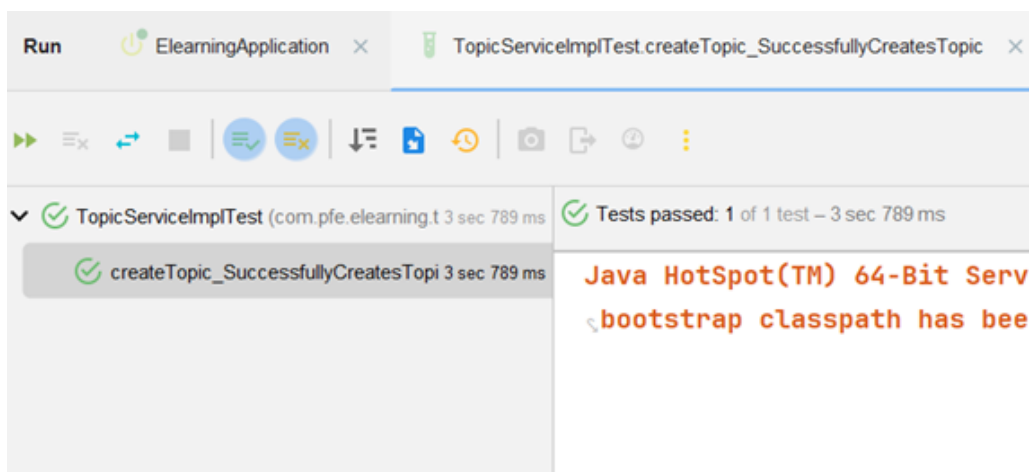


FIGURE 3.11 – Validation de test unitaire de création d’un sujet

Ce test unitaire valide la méthode `createTopic` de `TopicServiceImpl` en simulant la création d'un nouveau sujet avec le titre "Web development". Les dépendances sont simulées pour vérifier que la méthode appelle correctement le mapper pour convertir la demande de sujet, sauvegarde le sujet avec le repository, et retourne une réponse contenant les détails du sujet créé. En cas de succès, la méthode doit renvoyer une réponse non nulle avec l'identifiant et le titre du sujet créé correspondant à la demande.

3.3 Conclusion

Le premier sprint a été essentiel pour établir les fondations de notre projet. Nous avons défini les spécifications fonctionnelles, réalisé une modélisation conceptuelle approfondie, initié le codage des premiers modules et conduit des tests préliminaires. Ces activités ont permis de clarifier les exigences du projet, de structurer notre travail de manière efficace et de poser les bases pour les développements futurs. Les réalisations de ce sprint ont créé une base solide pour aborder les sprints suivants avec confiance et rigueur.

Chapitre 4

Etude et réalisation du deuxième Sprint

4.1 Introduction

Ce chapitre sera consacré pour présenter le travail durant la deuxième sprint, en passant par les étape d'analyse, de conception et de réalisation.

4.2 Le deuxième sprint

En partant sur le même principe que le sprint précédent, nous commençons par définir le but de notre deuxième sprint. À la suite d'une conversation entre le Product Owner et l'équipe Scrum, nous avons décidé le but suivant : « terminer la partie qui concerne la gestion de confidentialité, la gestion des cours, gestion de la Leçon, gestion des ressources, gestion des quiz ». Une fois, nous avons défini le but de notre sprint, il est temps de décider quelles histoires inclure dans ce dernier. Le Tableau suivant résume donc le backlog de notre deuxième sprint :

Id	Description de l'user story	Tâches effectuées	Estimation
1	En tant que membre, je veux changer mon mot de passe	<ul style="list-style-type: none">- Créer un endpoint API dédié pour la modification du mot de passe. (Backend)- Gérer les erreurs liées à la modification du mot de passe et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Créer une interface utilisateur pour permettre aux membres de saisir leur ancien mot de passe et leur nouveau mot de passe. (Frontend)- Gérer les réponses du backend et afficher des messages d'erreur ou de succès en conséquence. (Frontend)	4 jours

2	En tant que membre, je veux désactiver mon compte	<ul style="list-style-type: none">- Créer un endpoint API dédié pour désactiver le compte. (Backend)- Gérer les erreurs liées à la désactivation du compte et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Créer une interface utilisateur avec un bouton "Désactiver le compte". (Frontend)- Afficher un message de confirmation et rediriger le membre vers la page de connexion après la désactivation réussie du compte. (Frontend)	3 jours
3	En tant que membre, je veux supprimer mon compte	<ul style="list-style-type: none">- Créer un endpoint API dédié pour supprimer le compte. (Backend)- Gérer les erreurs liées à la suppression du compte et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Créer une interface utilisateur avec un bouton "Supprimer le compte". (Frontend)- Afficher un message de confirmation et rediriger le membre vers la page de connexion après la suppression réussie du compte. (Frontend)	4 jours

4	En tant qu'instructeur, je veux créer un nouveau cours assigné à un sujet	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour créer un cours. (Backend) - Gérer les erreurs liées à la création du cours et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Créer une interface utilisateur avec un formulaire de création de cours. (Frontend) 	6 jours
5	En tant qu'instructeur, je veux consulter la liste de mes cours	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour consulter la liste des cours. (Backend) - Gérer les erreurs liées à la récupération de la liste des cours et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Afficher la liste des cours récupérés depuis l'API backend. (Frontend) 	3 jours
6	En tant qu'instructeur, je veux pouvoir modifier les détails de mes cours existants	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour modifier les détails des cours. (Backend) - Gérer les erreurs liées à la modification du cours et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend) - Créer un formulaire de modification de cours avec les détails pré-remplis. (Frontend) 	6 jours

7	En tant qu'instructeur, je veux pouvoir supprimer mes cours existants	<ul style="list-style-type: none">- Créer un endpoint API dédié pour supprimer un cours. (Backend)- Gérer les erreurs liées à la suppression du cours et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Afficher un message de confirmation après la suppression réussie. (Frontend)	2 jours
8	En tant que membre, je veux consulter tous les cours existants	<ul style="list-style-type: none">- Créer un endpoint API dédié pour consulter tous les cours. (Backend)- Gérer les erreurs liées à la récupération de la liste des cours et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Afficher la liste des cours récupérés depuis l'API backend. (Frontend)	4 jours

9	En tant que candidat, je veux m'inscrire dans un cours	<ul style="list-style-type: none">- Créer une interface utilisateur permettant au candidat de rechercher et de sélectionner un cours auquel s'inscrire. (Frontend)- Implémenter un endpoint API dédié pour gérer l'inscription d'un candidat à un cours. (Backend)- Gérer les erreurs liées au processus d'inscription et renvoyer des réponses appropriées. (Backend)- Tester l'inscription à un cours avec différentes situations pour garantir son bon fonctionnement. (Backend)- Afficher une confirmation de l'inscription réussie dans l'interface utilisateur du candidat. (Frontend)	5 jours
10	En tant qu'instructeur, je veux voir la liste des candidats inscrits à mon cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour récupérer la liste des candidats inscrits à un cours spécifique. (Backend)- Gérer les erreurs liées à la récupération de la liste des candidats et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Afficher la liste des candidats inscrits dans l'interface utilisateur de l'instructeur pour chaque cours. (Frontend)	3 jours

11	En tant que candidat, je veux consulter mes cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour consulter tous les cours auxquels chaque candidat est inscrit. (Backend)- Gérer les erreurs liées à la récupération de la liste des cours et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Afficher la liste des cours récupérés depuis l'API backend. (Frontend)	4 jours
12	En tant qu'instructeur ou candidat, je veux filtrer les cours par nom et par popularité	<ul style="list-style-type: none">- Créer un endpoint API dédié pour filtrer les cours par nom et par popularité. (Backend)- Gérer les erreurs liées au filtrage des cours et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Afficher une interface utilisateur avec des options de filtrage par nom et par popularité, permettant aux membres de rechercher des cours spécifiques ou les cours les plus demandés. (Frontend)	2 jours

13	En tant que candidat, je veux filtrer les cours par nom de l'instructeur	<ul style="list-style-type: none">- Créer un endpoint API dédié pour filtrer les cours par nom de l'instructeur. (Backend)- Gérer les erreurs liées au filtrage des cours et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Afficher une interface utilisateur permettant de saisir le nom de l'instructeur et d'appliquer le filtre aux cours affichés. (Frontend)	3 jours
14	En tant qu'instructeur, je veux ajouter une leçon assignée à un cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour ajouter une leçon à un cours. (Backend)- Gérer les erreurs liées à l'ajout de la leçon et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Backend)- Créer une interface utilisateur permettant à l'instructeur de saisir les détails de la leçon et de l'assigner à un cours spécifique. (Frontend)	4 jours

15	En tant qu'instructeur, je veux consulter la liste des leçons associées à mon cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour récupérer la liste des leçons associées à un cours donné. (Backend)- Gérer les erreurs liées à la récupération de la liste des leçons et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Afficher la liste des leçons récupérées depuis l'API backend dans l'interface utilisateur pour l'instructeur. (Frontend)	3 jours
16	En tant qu'instructeur, je veux modifier les détails d'une leçon existante	<ul style="list-style-type: none">- Créer un endpoint API dédié pour la modification des détails d'une leçon existante. (Backend)- Gérer les erreurs liées à la modification de la leçon et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Créer une interface utilisateur permettant à l'instructeur de modifier les détails de la leçon sélectionnée. (Frontend)	3 jours

17	En tant qu'instructeur, je veux supprimer une leçon existante	<ul style="list-style-type: none">- Créer un endpoint API dédié pour la suppression d'une leçon existante. (Backend)- Gérer les erreurs liées à la suppression de la leçon et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Dans l'interface utilisateur, fournir une action de suppression pour chaque leçon permettant à l'instructeur de supprimer une leçon spécifique. (Frontend)	2 jours
18	En tant qu'instructeur, je veux ajouter des ressources associées à une leçon de mes cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour ajouter des ressources à une leçon spécifique. (Backend)- Gérer les erreurs liées à l'ajout des ressources et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Créer une interface utilisateur permettant à l'instructeur de télécharger et d'ajouter des ressources à une leçon sélectionnée. (Frontend)	3 jours

19	En tant qu'instructeur, je veux consulter les ressources associées à une leçon de mes cours	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer les ressources associées à une leçon spécifique. (Backend) - Gérer les erreurs liées à la récupération des ressources et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend) - Afficher la liste des ressources récupérées depuis l'API backend dans l'interface utilisateur pour l'instructeur. (Frontend) 	3 jours
20	En tant qu'instructeur, je veux modifier les ressources associées à une leçon de mes cours	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour la modification des ressources associées à une leçon spécifique. (Backend) - Gérer les erreurs liées à la modification des ressources et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend) - Dans l'interface utilisateur, fournir une action de modification pour chaque ressource permettant à l'instructeur de mettre à jour les ressources associées à une leçon sélectionnée. (Frontend) 	3 jours

21	En tant qu'instructeur, je veux supprimer les ressources associées à une leçon de mes cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour la suppression des ressources associées à une leçon spécifique. (Backend)- Gérer les erreurs liées à la suppression des ressources et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Dans l'interface utilisateur, fournir une option de suppression pour chaque ressource, permettant à l'instructeur de supprimer les ressources associées à une leçon sélectionnée. (Frontend)	3 jours
22	En tant qu'instructeur, je veux ajouter un quiz assigné à mon cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour ajouter un quiz à un cours spécifique. (Backend)- Gérer les erreurs liées à l'ajout du quiz et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Dans l'interface utilisateur, fournir un formulaire permettant à l'instructeur de saisir les détails du quiz et de l'assigner à un cours spécifique. (Frontend)	4 jours

23	En tant qu'instructeur, je veux consulter le quiz assigné à mon cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour récupérer le quiz assigné à un cours spécifique. (Backend)- Gérer les erreurs liées à la récupération du quiz et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Dans l'interface utilisateur, afficher les détails du quiz assigné à chaque cours dans la liste des cours de l'instructeur. (Frontend)	3 jours
24	En tant qu'instructeur, je veux modifier les détails d'un quiz assigné à mon cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour la modification des détails d'un quiz assigné à un cours spécifique. (Backend)- Gérer les erreurs liées à la modification du quiz et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Dans l'interface utilisateur, fournir une option de modification pour les détails du quiz, permettant à l'instructeur de mettre à jour les informations du quiz assigné à son cours. (Frontend)	3 jours

25	En tant qu'instructeur, je veux supprimer un quiz assigné à mon cours	<ul style="list-style-type: none">- Créer un endpoint API dédié pour la suppression d'un quiz assigné à un cours spécifique. (Backend)- Gérer les erreurs liées à la suppression du quiz et renvoyer des réponses appropriées. (Backend)- Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend)- Dans l'interface utilisateur, fournir une option de suppression pour chaque quiz, permettant à l'instructeur de supprimer un quiz assigné à son cours. (Frontend)	2 jours
26	En tant que candidat, je veux passer un quiz assigné à mon cours	<ul style="list-style-type: none">- Créer une interface utilisateur permettant au candidat de passer le quiz assigné à son cours. (Frontend)- Assurer que les fonctionnalités de validation et de soumission du quiz sont implémentées correctement côté frontend.- Tester l'interaction de passage de quiz avec différents scénarios d'utilisation pour garantir une expérience utilisateur fluide. (Frontend)	4 jours

27	En tant que candidat, je veux consulter le résultat de mon quiz	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer le résultat du quiz d'un candidat. (Backend) - Gérer les erreurs liées à la récupération du résultat du quiz et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement (Backend) - Dans l'interface utilisateur, afficher le résultat du quiz du candidat après qu'il l'a terminé. (Frontend) 	3 jours
----	---	---	---------

En partant sur le même principe que le sprint précédent, Passons maintenant au vif de notre sujet : les activités et le cycle de développement. Dans un sprint, nous pouvons dégager quatre activités principales qui sont la spécification fonctionnelle, la conception, le codage et le test.

4.2.1 Spécifications fonctionnelles

La spécification fonctionnelle dans notre cas se traduit par le diagramme de cas d'utilisation d'UML et la description textuelle de ces derniers.

4.2.1.1 1. Raffinement de cas d'utilisation du sprint 2

Le diagramme de cas d'utilisation du sprint 2 modélise les différentes fonctionnalités du sprint. Il présente trois acteurs héritant du membre qui ont présenté par l'acteur membre, qui partagent des fonctionnalités communes telles que :

- **S'authentifier**
- **Géré confidentialité**

4.2.1.2 Diagramme de cas d'utilisation détaillé du Sprint 2

La Figure suivante présente le diagramme de cas d'utilisation détaillé de ce deuxième sprint :

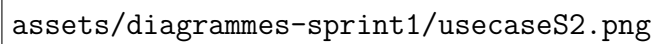
The image is a large, empty rectangular box with a thin black border. Inside the box, the text 'assets/diagrammes-sprint1/usecaseS2.png' is centered horizontally and vertically. This text likely represents a missing or placeholder image for the UML Use Case diagram.

FIGURE 4.1 – : Diagramme De Cas D’Utilisation Détaillé du Sprint 2

Le diagramme de cas d’utilisation détaillé illustre les fonctionnalités accessibles à chaque utilisateur du système. Tous les utilisateurs ont la possibilité de modifier leur mot de passe, de désactiver ou de supprimer leur compte après s’être authentifiés. Seuls les instructeurs ont le privilège de créer de nouveaux cours, et par conséquent, ils peuvent également ajouter des leçons, des ressources et des quiz à ces cours. L’authentification est une étape préalable nécessaire avant toute action sur le système. De plus, les candidats peuvent passer un quiz un fois inscrit à un cours comportant ce quiz. En utilisant ce diagramme, nous avons pu visualiser de manière exhaustive les fonctionnalités attendues et les interactions entre les utilisateurs et le système. Cela nous a permis de mieux comprendre les besoins des utilisateurs et de garantir le développement d’un système répondant à leurs attentes spécifiques.

4.2.1.3 Description textuelle de cas d'utilisation

En partant sur le même principe que le sprint précédent. Le tableau suivant explique comment l'utilisateur peut modifier son mot de passe.

TABLE 4.2 – Description textuelle du cas d'utilisation « Modifier mot de passe »

Cas d'utilisation	Modifier mot de passe
Acteur principal	Gestionnaire, candidats et instructeurs
Pré-condition	Le membre est connecté à son compte sur la plateforme.
Post-condition	Mot de passe modifié
Scénario Nominal	<p>Le membre accède à la section de gestion de son compte.</p> <p>Le membre sélectionne l'option "Modifier mot de passe".</p> <p>Le système affiche un formulaire permettant au membre de saisir son ancien mot de passe ainsi que le nouveau mot de passe souhaité avec la confirmation.</p> <p>Le membre saisit son ancien mot de passe et entre le nouveau mot de passe.</p> <p>Une fois les champs remplis, le membre valide le formulaire.</p> <p>Le système vérifie que le mot de passe actuel correspond à celui enregistré dans la base de données. Si la correspondance est confirmée, le nouveau mot de passe est enregistré et associé au compte du membre.</p> <p>Le système affiche un message de confirmation indiquant que le mot de passe a été modifié avec succès.</p>
Scénario alternatif SA1	<p>Mot de passe actuel invalide.</p> <p>SA1 commence après l'étape 6 du scénario nominal.</p> <p>Le système détecte que le mot de passe actuel saisi par le membre est incorrect.</p>

	<p>Le système affiche un message d’erreur indiquant que le mot de passe actuel est invalide.</p> <p>Le membre est invité à saisir à nouveau son mot de passe actuel.</p>
Scénario alternatif SA2	<p>Le mot de passe et sa confirmation ne sont pas identiques.</p> <p>SA2 commence après l’étape 4 du scénario nominal.</p> <p>Le système détecte que le nouveau mot de passe saisi par le membre et sa confirmation ne correspondent pas.</p> <p>Le système affiche un message d’erreur indiquant que la confirmation du mot de passe ne correspond pas au nouveau mot de passe saisi.</p> <p>Le membre est invité à saisir à nouveau le nouveau mot de passe et sa confirmation.</p>
Scénario alternatif SA3	<p>Champ vide.</p> <p>SA3 commence après l’étape 3 du scénario nominal.</p> <p>Le système détecte que le membre n’a pas saisi de valeur pour le mot de passe actuel, le nouveau mot de passe ou sa confirmation.</p> <p>Le système affiche un message d’erreur indiquant que tous les champs doivent être remplis.</p> <p>Le membre est invité à saisir les informations manquantes.</p>

TABLE 4.3 – Description textuelle du cas d’utilisation « Ajouter cours »

Cas d’utilisation	Ajouter cours
Acteur principal	Instructeurs
Pré-condition	L’instructeur est authentifié sur la plateforme.
Post-condition	Le cours est ajouté avec succès.
Scénario Nominal	1. L’instructeur accède à la section de gestion de ses cours.

	<p>2. L'instructeur sélectionne l'option "Ajouter cours".</p> <p>3. Le système affiche un formulaire permettant à l'instructeur de saisir les détails du nouveau cours, tels que le titre, la description, la durée, etc.</p> <p>4. L'instructeur remplit les champs du formulaire avec les informations nécessaires.</p> <p>5. Une fois les champs remplis, l'instructeur valide le formulaire.</p> <p>6. Le système vérifie que tous les champs obligatoires sont remplis.</p> <p>7. Si toutes les informations sont valides, le nouveau cours est ajouté à la base de données.</p> <p>8. Le système affiche un message de confirmation indiquant que le cours a été ajouté avec succès.</p>
Scénario alternatif SA1	<p>Champs obligatoires manquants.</p> <p>SA1 commence après l'étape 6 du scénario nominal.</p> <p>Le système détecte que certains champs obligatoires du formulaire ne sont pas remplis.</p> <p>Le système affiche un message d'erreur indiquant que tous les champs obligatoires doivent être remplis.</p> <p>L'instructeur est invité à remplir les champs manquants.</p>
Scénario alternatif SA2	<p>Erreur lors de l'ajout du cours.</p> <p>SA2 commence après l'étape 7 du scénario nominal.</p> <p>Le système rencontre une erreur lors de l'ajout du cours à la base de données.</p> <p>Le système affiche un message d'erreur indiquant que l'ajout du cours a échoué.</p> <p>L'instructeur est invité à réessayer ultérieurement ou à contacter le support technique.</p>

TABLE 4.4 – Description textuelle du cas d'utilisation « Ajouter cours »

Cas d'utilisation	Ajouter cours
Acteur principal	Instructeurs
Pré-condition	L'instructeur est authentifié sur la plateforme.
Post-condition	Le cours est ajouté avec succès.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'instructeur accède à la section de gestion de ses cours. 2. L'instructeur sélectionne l'option "Ajouter cours". 3. Le système affiche un formulaire permettant à l'instructeur de saisir les détails du nouveau cours, tels que le titre, la description, la durée, etc. 4. L'instructeur remplit les champs du formulaire avec les informations nécessaires. 5. Une fois les champs remplis, l'instructeur valide le formulaire. 6. Le système vérifie que tous les champs obligatoires sont remplis. 7. Si toutes les informations sont valides, le nouveau cours est ajouté à la base de données. 8. Le système affiche un message de confirmation indiquant que le cours a été ajouté avec succès.
Scénario alternatif SA1	<p>Champs obligatoires manquants.</p> <p>SA1 commence après l'étape 6 du scénario nominal.</p> <p>Le système détecte que certains champs obligatoires du formulaire ne sont pas remplis.</p> <p>Le système affiche un message d'erreur indiquant que tous les champs obligatoires doivent être remplis.</p> <p>L'instructeur est invité à remplir les champs manquants.</p>
Scénario alternatif SA2	<p>Erreur lors de l'ajout du cours.</p> <p>SA2 commence après l'étape 7 du scénario nominal.</p>

	<p>Le système rencontre une erreur lors de l'ajout du cours à la base de données.</p> <p>Le système affiche un message d'erreur indiquant que l'ajout du cours a échoué.</p> <p>L'instructeur est invité à réessayer ultérieurement ou à contacter le support technique.</p>
--	--

TABLE 4.5 – Description textuelle du cas d'utilisation « Modifier leçon »

Cas d'utilisation	Modifier leçon
Acteur principal	Instructeurs
Précondition	L'instructeur est authentifié sur la plateforme.
Postcondition	Le cours est ajouté avec succès.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'instructeur accède à la page de gestion des cours. 2. L'instructeur sélectionne le cours auquel la leçon à modifier est associée. 3. Le système affiche une liste des leçons disponibles dans le cours sélectionné dans un tableau modifiable. 4. L'instructeur choisit la leçon qu'il souhaite modifier et entre les détails de la leçon selon ses besoins, puis clique sur le bouton "Modifier" à côté de cette leçon. 5. Une fois les modifications effectuées, l'instructeur valide le formulaire. 6. Le système enregistre les modifications de la leçon dans la base de données. 7. Le système affiche un message de confirmation indiquant que la leçon a été modifiée avec succès.
Scénario alternatif SA1	<p>Erreur lors de la modification de la leçon.</p> <p>SA1 commence après l'étape 7 du scénario nominal.</p> <p>Le système rencontre une erreur lors de la modification de la leçon dans la base de données.</p>

	<p>Le système affiche un message d'erreur indiquant que la modification de la leçon a échoué.</p> <p>L'instructeur est invité à réessayer ultérieurement ou à contacter le support technique.</p>
--	---

TABLE 4.6 – Description textuelle du cas d'utilisation « Passer un quiz »

Cas d'utilisation	Passer un quiz
Acteur principal	Candidats
Précondition	Le candidat est authentifié sur la plateforme.
Postcondition	Lesson est modifiée avec succès.
Scénario Nominal	<ol style="list-style-type: none"> 1. Le candidat accède à la liste des cours auxquels il est inscrit. 2. Le candidat sélectionne le cours contenant le quiz qu'il souhaite passer. 3. Le système affiche les détails du cours sélectionné, y compris les quiz disponibles. 4. Le candidat identifie le quiz qu'il souhaite passer et sélectionne l'option "Passer le quiz". 5. Le système affiche le quiz avec les questions et les options de réponses. 6. Le candidat répond à chaque question en sélectionnant l'option appropriée. 7. Une fois toutes les questions répondues, le candidat soumet ses réponses. 8. Le système enregistre les réponses du candidat et évalue son score. 9. Le système affiche les résultats du quiz, y compris le score obtenu par le candidat. 10. Le candidat peut consulter ses réponses et voir les corrections si disponibles.

4.2.2 La modélisation conceptuelle

4.2.2.1 Diagramme de classes

Dans le contexte de cette partie, le diagramme de classes est utilisé pour représenter les différentes classes et entités impliquées dans le processus de l'authentification et la gestion des confidentialité, cours, Lesson, ressources, quiz, inscription. Le diagramme de classes permet de visualiser la structure et les relations entre ces classes, ce qui peut aider à comprendre comment les différentes entités du système interagissent dans ce processus



FIGURE 4.2 – : Diagramme De Classe Du Second Sprint

4.2.2.2 Diagramme de séquence détaillé

— Diagramme de séquence associé à « Modifier Mot De Passe »

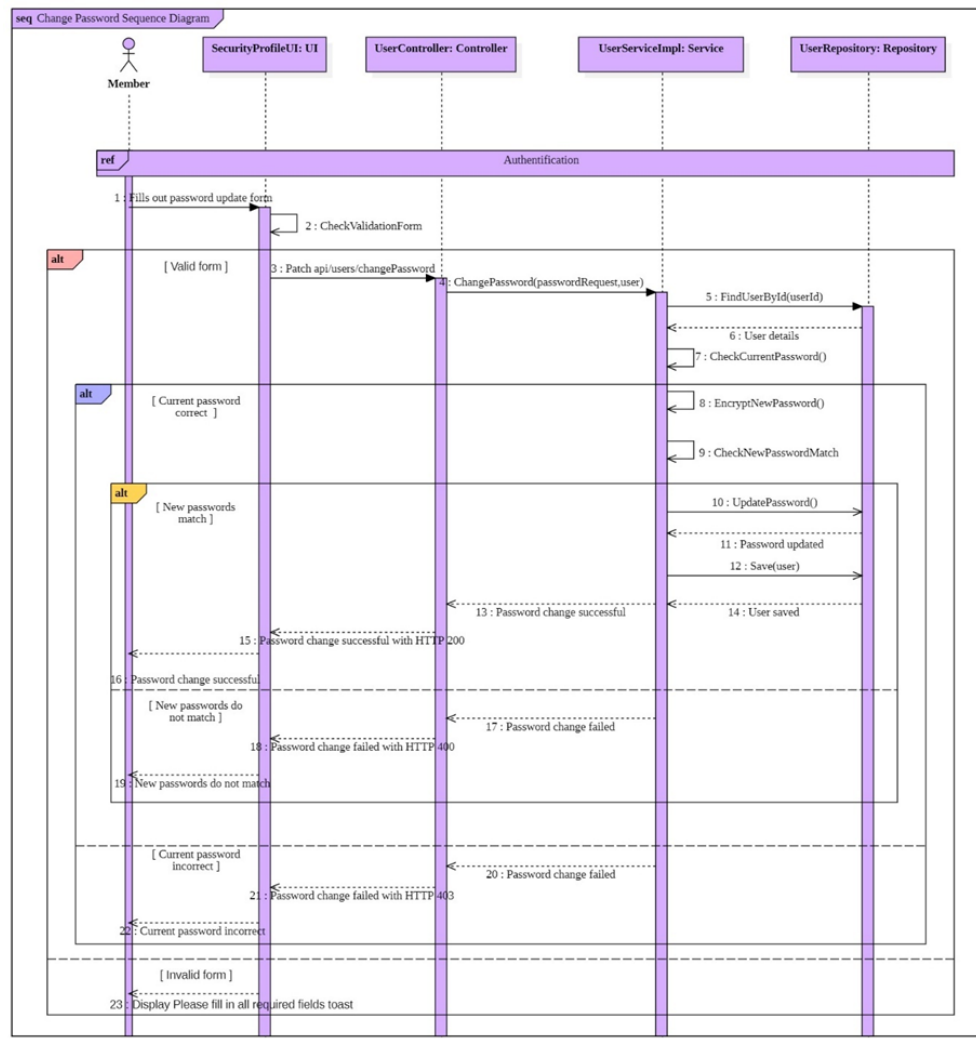


FIGURE 4.3 – :Diagramme De Séquence Du Cas D’Utilisation « Changer Mot De Passe »

Le diagramme de séquence ci-dessus illustre le processus de changement de mot de passe d’un membre à travers l’interface utilisateur de profil sécurité. L’interaction commence lorsque le membre interagit avec l’interface SecurityProfileUI en remplissant le formulaire de mise à jour du mot de passe. Avant d’envoyer une requête PATCH pour modifier le mot de passe de membre via l’API REST `"/api/users/changePassword"`, une validation du formulaire est effectuée dans l’interface utilisateur pour s’assurer que tous les champs requis sont remplis avec des informations correctes. Si des champs sont manquants ou invalides, un message d’erreur est affiché, demandant au membre de corriger les erreurs. Une fois que tous les champs du formulaire sont validés, l’interface utilisateur envoie la requête PATCH au contrôleur UserController pour modifier le mot de passe de l’utilisa-

teur. Le contrôleur UserController déclenche ensuite la méthode changePassword de la classe UserServiceImpl pour effectuer le changement de mot de passe. La classe UserServiceImpl vérifie d'abord si le membre est connecté en utilisant la classe Principal. Ensuite, elle vérifie si le mot de passe actuel est correct en comparant le mot de passe fourni avec celui stocké dans la base de données via la classe UserRepository. Si le mot de passe actuel est correct, la classe UserServiceImpl vérifie si les deux nouveaux mots de passe fournis correspondent. Si c'est le cas, la classe UserServiceImpl procède à l'encryption du nouveau mot de passe et met à jour le mot de passe de l'utilisateur via la méthode updatePassword de la classe UserRepository. Une fois que le mot de passe est mis à jour avec succès, la classe UserServiceImpl enregistre les modifications de l'utilisateur dans la base de données en utilisant la méthode save de la classe UserRepository. Enfin, un message "Password change successful" est renvoyé au contrôleur UserController, qui notifie ensuite l'interface utilisateur de la réussite de l'opération. En outre, le processus de vérification si les mots de passe correspondent et de déclenchement d'erreurs correspondantes est représenté dans le diagramme par des branches alternatives. Si les mots de passe ne correspondent pas, une IllegalStateException est levée et un message "Passwords do not match" est renvoyé au contrôleur UserController. Ce dernier transmet alors ce message à l'interface utilisateur pour informer le membre que le changement de mot de passe a échoué.

— Diagramme de séquence associé à « Ajouter Un Cours »

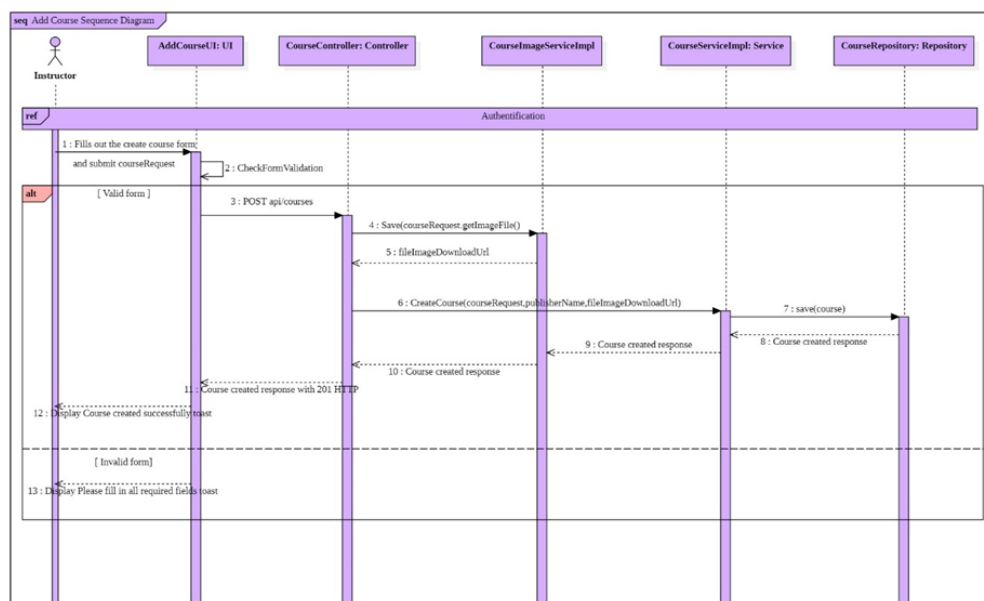


FIGURE 4.4 – : Diagramme De Séquence Du Cas D'Utilisation « Ajouter Un Cours »

Ce diagramme de séquence illustre le processus de création d'un cours par un instructeur dans l'interface utilisateur `AddCourseUI`. Le processus commence lorsque l'instructeur initialise la création d'un cours dans l'interface utilisateur. Dès le début de la création, l'interface utilisateur effectue une validation des champs obligatoires du formulaire pour s'assurer que toutes les informations requises sont fournies correctement. Si des champs obligatoires sont manquants ou invalides, un message d'erreur est affiché à l'instructeur, lui demandant de corriger les erreurs avant de pouvoir soumettre le formulaire. Une fois que tous les champs obligatoires sont remplis correctement, l'instructeur soumet le formulaire de création de cours. L'interface utilisateur envoie ensuite une requête au contrôleur de cours, `CourseController`, pour créer le cours. Le contrôleur de cours commence par extraire le nom d'utilisateur de l'instructeur à partir des détails de l'utilisateur authentifié. Ensuite, il vérifie si une image est fournie avec la requête. Si tel est le cas, le contrôleur utilise le service de fichiers pour sauvegarder l'image téléchargée et récupère l'URL de téléchargement de l'image. Ensuite, le contrôleur envoie une demande de création de cours au service de cours, `CourseServiceImpl`. Avant de procéder à la sauvegarde du cours, une vérification est effectuée pour s'assurer que tous les champs obligatoires sont remplis correctement. Si la validation réussit, le service de cours utilise le répertoire de cours, `CourseRepository`, pour sauvegarder les informations du cours dans la base de données. Si une image est fournie avec le cours, le service de cours utilise également le service d'images de cours, `CourseImageServiceImpl`, pour sauvegarder l'image associée au cours. Une fois la sauvegarde réussie, le service de cours retourne une réponse contenant les détails du cours créé au contrôleur de cours. Enfin, le contrôleur de cours renvoie une réponse HTTP contenant les détails du cours créé à l'interface utilisateur, qui affiche un message de succès indiquant que le cours a été créé avec succès.

— Diagramme de séquence associé à « Supprimer Une Leçon »

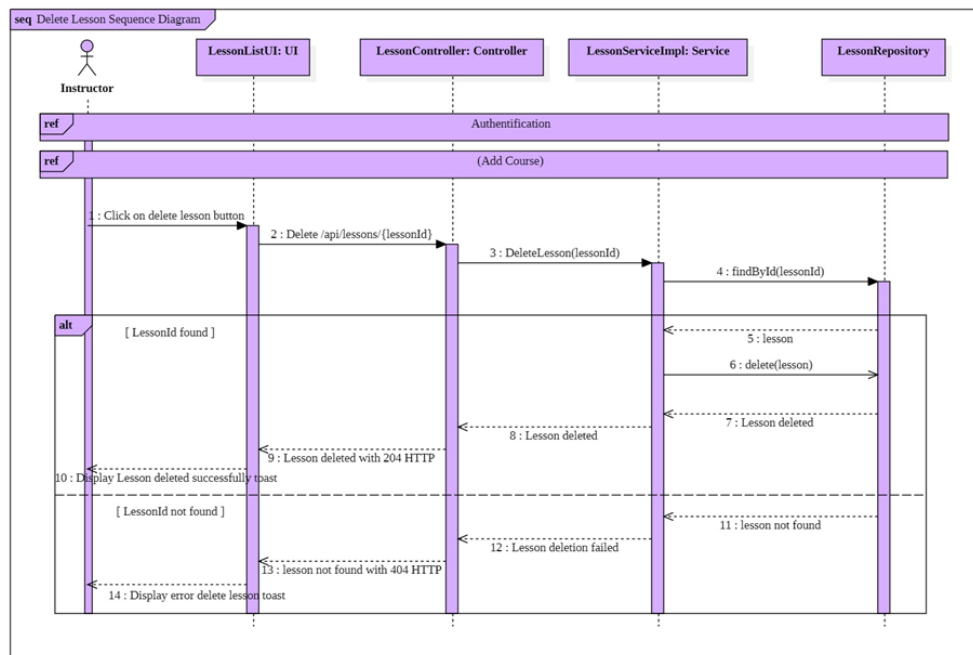


FIGURE 4.5 – :Diagramme De Séquence Du Cas D’Utilisation « Supprimer Une Leçon »

Le diagramme de séquence présent décrit le processus de suppression d’une leçon dans l’application, initié par un instructeur à travers l’interface LessonListUI. Lorsque l’instructeur clique sur le bouton de suppression associé à une leçon spécifique, une requête DELETE est envoyée au contrôleur de leçon (LessonController), contenant l’identifiant de la leçon à supprimer. Cette requête est alors traitée par le service de leçon (LessonServiceImpl), qui cherche la leçon correspondante dans le référentiel de données (LessonRepository). Si la leçon est trouvée, le service de leçon informe le contrôleur de leçon de la réussite de la suppression, et celui-ci renvoie un code de statut HTTP 204 (No Content) à l’interface utilisateur. Cette réponse est reçue par LessonListUI, qui affiche un message de succès indiquant que la leçon a été supprimée avec succès. En revanche, si la leçon n’est pas trouvée dans le référentiel, le service de leçon informe le contrôleur de leçon de l’échec de la suppression en raison de l’absence de la leçon. Le contrôleur répond alors avec un code de statut HTTP 404 (Not Found), indiquant que la ressource demandée n’a pas été trouvée. Cette réponse est reçue par LessonListUI, qui affiche un message d’erreur informant l’instructeur que la leçon n’a pas pu être trouvée pour la suppression.

4.2.3 Tests

```

112 @Test
113 void shouldReturnEmptyListOfLessonResponsesWhenCourseHasNullListOfLessons() {
114     // Create a Course object with null list of lessons
115     Course course = new Course();
116     course.setId(1);
117     course.setTitle("Test Course");
118     course.setDescription("This is a test course");
119     course.setCreatedAt(LocalDate.now());
120     course.setStartDate(LocalDate.now());
121     course.setEndDate(LocalDate.now().plusWeeks(2));
122     course.setFree(true);
123     course.setTimeCommitment("2 weeks");
124     User publisher = new User();
125     publisher.setFirstname("John");
126     publisher.setLastname("Doe");
127     course.setPublisher(publisher);
128     course.setLessons(null); // Set the list of lessons to null
129
130     // Call the method to map lesson responses
131     List<LessonResponse> lessonResponses = courseMapper.mapLessonResponses(course.getLessons());
132
133     // Assertions
134     assertNotNull(lessonResponses);
135     assertTrue(lessonResponses.isEmpty()); // Ensure that an empty list of lesson responses is returned
136 }

```

FIGURE 4.6 – :Test Unitaire Pour Une Liste Vide Des Leçons

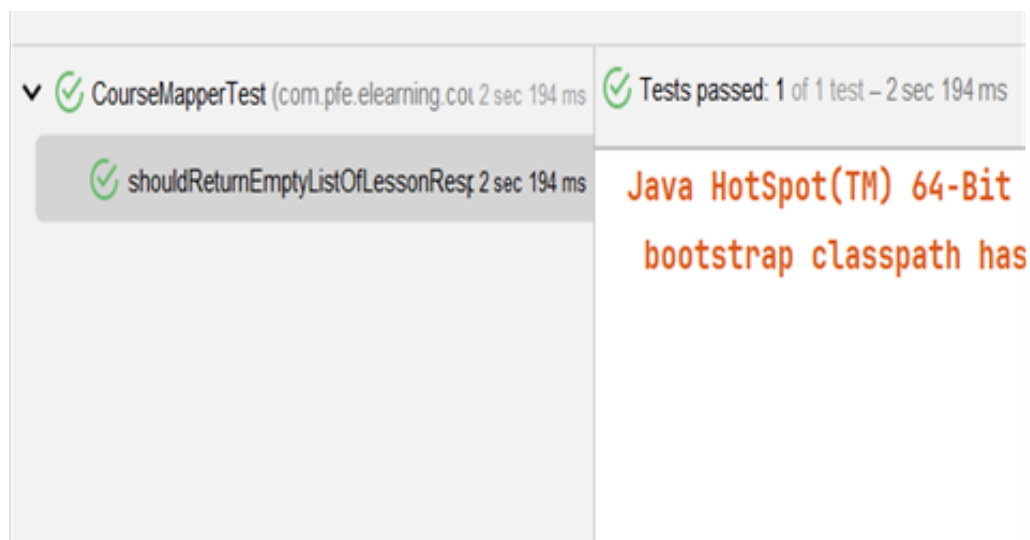


FIGURE 4.7 – :Résultat Réussi Pour Une Liste Vide Des Leçons

La figure de test unitaire présente le scénario où la liste des leçons d'un cours est nulle, et elle vérifie le comportement de la méthode 'mapLessonResponses' dans cette situation. La figure de résultat montre le résultat du test unitaire exécuté avec succès. Elle confirme que la méthode 'mapLessonResponses' a correctement renvoyé une liste vide de réponses de leçon, conformément aux attentes du test. Cela démontre que la méthode fonctionne comme prévu dans ce cas de figure spécifique.

4.3 Conclusion :

Le deuxième sprint a consolidé les bases du projet en précisant les spécifications, en affinant la modélisation, en développant des modules supplémentaires et en réa-

lisant des tests rigoureux. Ces efforts ont renforcé la structure et assuré la qualité du système, nous préparant efficacement pour le troisième sprint.

Chapitre 5

La phase closure

5.1 Introduction

Ce chapitre représente la fin de notre projet en examinant l'étape finale. Nous aborderons l'environnement de développement, tant sur le plan matériel que logiciel, qui a joué un rôle essentiel dans la mise en place de notre solution. Par la suite, nous soulignerons les langages de programmation employés.

5.2 Environnement de développement

La notion d'environnement de développement fait référence à l'ensemble des outils et du langage employés pour mettre en place une solution informatique. Au départ, nous abordons le cadre matériel.

5.2.1 Environnement matériel

Les caractéristiques de la machine utilisée pour la réalisation de notre projet sont évoquées dans le tableau suivant :

TABLE 5.1 – Environnement Matériel

Ordinateur	Propriétaire	Caractéristiques
Ordinateur 1	Ikram AYOUB	Processeur : 11e génération Intel(R) Core(TM) i5-1135G7 @ 2,40 GHz (2,42 GHz) Mémoire RAM : 8,00 Go Disque dur : 256Go SSD Système d'exploitation : Windows 10 64 bits
Ordinateur 2	Syrine AYEB	Processeur : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz Mémoire RAM : 8,00 Go Disque dur : 452Go SSD Système d'exploitation : Windows 11 64 bits

5.2.2 Environnement logiciel

Dans cette section, nous présentons les outils logiciels et les langages de programmation qui ont contribué au développement de notre solution.

5.2.2.1 Environnement de Développement Intégré (EDI)

IntelliJ IDEA : IntelliJ IDEA est un environnement de développement intégré (IDE) très populaire développé par JetBrains. Il est conçu principalement pour Java, mais prend également en charge un large éventail de langages de programmation, notamment Kotlin, Groovy, Scala, etc. Cet outil de développement logiciel est très bénéfique pour les programmeurs car il leur permet de créer, éditer, déboguer et gérer efficacement le code dans diverses applications.

5.2.2.2 Bibliothèque

Angular : Angular est un framework d'application Web d'une seule page basé sur TypeScript, gratuit et open source, exécuté sur Node.js. Il est dirigé par l'équipe Angular de Google et par une communauté d'individus et d'entreprises. Angular est une réécriture complète de la même équipe qui a construit AngularJS. L'écosystème

angulaire se compose d'un groupe diversifié de plus de 1,7 million de développeurs, auteurs de bibliothèques et créateurs de contenu.

Spring Boot : Spring Boot est un framework Java open source utilisé pour programmer des applications Spring autonomes de qualité production avec un minimum d'effort. Spring Boot est une extension de convention sur configuration pour la plateforme Spring Java destinée à aider à minimiser les problèmes de configuration lors de la création d'applications basées sur Spring. La plupart des applications peuvent être préconfigurées à l'aide de la « vision avisée » de l'équipe Spring sur la meilleure configuration et utilisation de la plate-forme Spring et des bibliothèques tierces. Il est largement utilisé pour créer des microservices, des applications Web et d'autres projets basés sur Java en raison de sa facilité d'utilisation et de sa robustesse. [6]

5.2.2.3 Outils d'Évaluation d'API

Postman : Postman est une application qui permet de tester des API web. Le logiciel a été créé en 2012 par Abhinav Asthana, Ankit Sobti et Abhijit Kane à Bangalore, en Inde afin de résoudre le problème de partage des tests API. À l'origine, il a été développé comme un plugin pour Google Chrome, puis un client riche et enfin un client léger. Il est désormais utilisé par plus de 500 000 entreprises dans le monde. Le développeur, Postman Inc., originaire d'Inde, a son siège social à San Francisco.

JUnit : JUnit est un framework de test unitaire pour le langage de programmation Java. Il fournit une structure pour écrire des tests automatisés afin de vérifier le bon fonctionnement des différentes parties de votre code. Les tests unitaires sont essentiels pour assurer la qualité du logiciel en identifiant et en corrigeant les erreurs dès le début du processus de développement.

5.2.2.4 Outils de Développement de Serveur Local

XAMPP : XAMPP est l'environnement de développement PHP le plus largement utilisé. Il s'agit d'une distribution Apache entièrement gratuite et facile à installer contenant MySQL, PHP et Perl. Le paquetage open source XAMPP a été mis au point pour être incroyablement facile à installer et à utiliser. [11]

5.2.2.5 Système de Gestion de Base de Données

MySQL : MySQL est un système de gestion de base de données relationnelle open-source. Il utilise le langage SQL pour organiser, modifier et extraire des don-

nées. MySQL travaille avec un système d'exploitation pour mettre en œuvre une base de données dans le stockage de l'ordinateur, gérer les utilisateurs, permettre l'accès réseau, et assurer l'intégrité des données et les sauvegardes. [7]

5.2.2.6 Plateforme de Gestion de Code Source

GitHub : GitHub représente une plateforme de développement logiciel qui repose sur Git, offrant des outils de gestion de version, de collaboration et d'hébergement de code source. GitHub est une plateforme où les développeurs peuvent héberger leurs projets, suivre les modifications du code, travailler en collaboration avec d'autres développeurs, et gérer les problèmes et les demandes de tirage (pull requests). GitHub est fréquemment employé pour la création de logiciels open source et les projets en collaboration.

5.2.2.7 Plateforme de Gestion de Projet

Trello : Trello est une plateforme de gestion de projet en ligne qui utilise un système de tableau visuel pour organiser et prioriser les tâches. Les utilisateurs peuvent créer des tableaux pour différents projets, puis ajouter des listes de tâches à chaque tableau. Chaque tâche est représentée par une carte, que les utilisateurs peuvent déplacer entre les listes pour indiquer son statut (à faire, en cours, terminée, etc.). Trello permet également aux utilisateurs de collaborer en équipe, d'assigner des tâches, de fixer des dates d'échéance et de joindre des documents à chaque carte.

5.2.2.8 Outils de Modélisation UML

StarUML : StarUML est une plateforme de modélisation UML qui propose une interface conviviale pour créer une variété de diagrammes de conception, notamment des diagrammes de classes, de séquence, d'activité, et bien d'autres. Son utilisation facilite grandement la visualisation et la communication des concepts de conception tout au long du processus de développement logiciel. En permettant aux développeurs et aux concepteurs de collaborer efficacement, StarUML favorise la création de modèles visuels clairs et facilement compréhensibles.

5.2.3 Langages de programmation

Nous avons utilisé les langages de programmation suivants pour développer notre application.

5.2.3.1 Java

Java : Java, créé par Sun Microsystems en 1995, est un langage de programmation polyvalent et orienté objet. Il est largement utilisé pour développer la logique métier et les composants back-end des applications. Grâce à sa robustesse, sa portabilité et sa grande communauté de développeurs, Java est un choix populaire pour les applications d'entreprise. Il est également au cœur de nombreux services et applications essentiels du monde numérique, ainsi que de produits et services innovants.

5.2.3.2 TypeScript

TypeScript : TypeScript est un langage de programmation développé par Microsoft qui étend les fonctionnalités de JavaScript en ajoutant un typage statique. Il est utilisé pour le développement des interfaces utilisateur dynamiques et réactives avec Angular. TypeScript permet de détecter et de corriger les erreurs de manière proactive, ce qui contribue à la robustesse et à la fiabilité de notre application front-end.

5.3 Conclusion

Au cours de ce chapitre, nous avons exposé l'environnement de développement matériel et logiciel ainsi que les langages de programmation employés dans notre projet. La mise en place et le développement de notre solution informatique ont été essentiels pour ces éléments.

Conclusion générale

Ce mémoire présente le bilan du travail effectué durant la période de mon stage de Fin d'Etudes au sein de la Digital Factory de l'Office Chérifien des Phosphates. Le but de mon stage est de contribuer à la conception ainsi que le développement d'une solution moderne pour le gestionnement et le suivi complet des anomalies dans les mines du groupe [OCP](#).

La naissance de ce projet est due au besoin de la centralisation des données ainsi minimisation le temps de réponse pour résoudre les anomalies lors de l'extraction de phosphates. En effet, la plateforme Prospektor présente une résolution à cette problématique en se basant sur une vision architecturale qui répond aux besoins de modularité ainsi que la scalabilité.

Durant l'ensemble des activités menées dans le cadre de ce projet, nous avons passé par l'ensemble des phases d'un projet logiciel ainsi que nous avons utilisé plusieurs technologies notamment Spring Boot comme framework d'implémentation de [API](#) Backend et Android Native & ReactJs comme framework d'implémentation de la logique Frontend. Le projet s'est déroulé en adoptant une méthodologie agile basée essentiellement sur Scrum comme processus de développement afin de maximiser la livraison de valeur dans un minimum de temps.

Durant ce travail, on a pu réaliser plusieurs fonctionnalités, à savoir le suivi complet des anomalies dans les sites, leurs attachements et les commentaires des utilisateurs sur ces anomalies. Toutefois, il reste plusieurs fonctionnalités à réaliser, à savoir stocker les données localement en cas d'absence de connexion internet, envoyer des notifications par [SMS](#) ainsi les analytiques et le moteur de recherche qui doivent être livrées les mois qui suivent.

[table]xcolor [HTML]BD9FE1

Webographie

- [2] CHATGPT. *W3Schools*. Accès le 28 janvier 2024. URL : <https://www.w3schools.com/about/>.
- [3] D. SHAH. *Coursera*. Consulté le 27 janvier 2024. URL : <https://www.classcentral.com/help/what-is-coursera>.
- [4] WIKIPÉDIA. *Udemy*. Consulté le 28 janvier 2024. URL : <https://fr.wikipedia.org/wiki/Udemy>.

Bibliographie

- [1] Claude AUBRY. *L'inspiration du Scrum : Une Analogie avec le Rugby*. Dunod, 2010.