

Table des matières

1 Contexte Général Du Projet	2
Introduction	3
1.1 Présentation de la société	3
1.1.1 Description de la société	3
1.1.2 Logo de la société	3
1.1.3 Informations de contact de la société	3
1.1.4 Services de la société	4
1.1.5 Organigramme fonctionnel de la société	5
1.2 Étude de l'existant	5
1.2.1 Description de l'existant	5
1.2.2 Critique de l'existant	8
1.2.3 Solution proposée	10
1.3 Présentation du projet	10
1.4 Langage et méthodologie de conception	11
1.5 Pourquoi Scrum	11
1.6 Choix des technologies	13
Conclusion	13
2 Planification et Architecture	14
Introduction	15
2.1 Analyse des besoins	15
2.1.1 Identification des acteurs	15
2.1.1.1 Diagramme de contexte statique	16
2.1.2 Les besoins fonctionnels	16
2.1.3 Les besoins non fonctionnels	20
2.2 Planning du traitement de cas d'utilisation	21
2.2.1 Priorités	21
2.2.2 Risques	21
2.3 Prototypage des interfaces	22
2.4 Pilotage du projet avec Scrum	24
2.4.1 Les outils Scrum	24
2.4.2 Équipe et rôles	24
2.4.3 Le backlog du produit	24
2.4.4 Diagramme de cas d'utilisation global	31
2.5 Planification des sprints	33
2.6 Environnement du travail	34
2.6.1 Environnement matériel	34

TABLE DES MATIÈRES

2.6.2	Environnement logiciel	35
2.7	Architecture de l'application	38
2.7.1	Architecture logique	38
2.7.2	Architecture physique	39
	Conclusion	39
3	Premier Sprint : Mise en Place des Fonctions Utilisateur et Sujets	40
	Introduction	41
3.1	Le premier sprint	41
3.1.1	Spécification fonctionnelle.....	53
3.1.1.1	Raffinement de cas d'utilisation du sprint 1.....	53
3.1.1.2	Diagramme de cas d'utilisation détaillé du sprint1 ...	53
3.1.1.3	Description textuelle de cas d'utilisation	55
3.1.2	La modélisation conceptuelle	62
3.1.2.1	Diagramme de classes.....	62
3.1.2.2	Diagramme de séquence détaillé	63
3.1.3	Réalisation	69
3.1.4	Tests	73
3.1.4.1	Tests unitaires	74
3.1.4.2	Tests avec Postman	75
	Conclusion	76
4	Deuxième Sprint : Gestion Pédagogique et Confidentialité	77
	Introduction	78
4.1	Le deuxième sprint	78
4.1.1	Spécifications fonctionnelles	93
4.1.1.1	Raffinement de cas d'utilisation du sprint 2.....	93
4.1.2	Diagramme de cas d'utilisation détaillé du Sprint 2	93
4.1.3	Description textuelle de cas d'utilisation.....	95
4.1.4	La modélisation conceptuelle	100
4.1.4.1	Diagramme de classes.....	100
4.1.4.2	Diagramme de séquence détaillé	100
4.1.5	Réalisation	107
4.1.6	Tests	113
4.1.6.1	Tests unitaires	113
4.1.6.2	Tests avec Postman	115
	Conclusion	115
5	Troisième Sprint : Mise en Place du Forum de Discussion et Planification des Réunions	116
	Introduction	117
5.1	Le troisième sprint	117
5.1.1	Spécification fonctionnelle	125
5.1.1.1	Raffinement de cas d'utilisation du sprint 3.....	125
5.1.1.2	Diagramme de cas d'utilisation détaillé du sprint3 ...	127
5.1.1.3	Description textuelle des cas d'utilisation	128
5.1.2	La modélisation conceptuelle	131

TABLE DES MATIÈRES

TABLE DES MATIÈRES

5.1.2.1	Diagramme de classes	131
5.1.2.2	Diagramme de séquence détaillé	133
5.1.3	Réalisation	137
5.1.4	Tests	140
5.1.4.1	Les tests unitaires	140
5.1.4.2	Tests avec Postman	142
Conclusion	142

Table des figures

1.1	Logo de la société LeaderPos	3
1.2	Organigramme fonctionnel de LeaderPos	5
1.3	Page d'accueil de la plateforme Coursera	5
1.4	Page d'accueil de la plateforme Udemy	6
1.5	Page d'accueil de la plateforme W3School	6
1.6	Processus Scrum	12
2.1	Diagramme de contexte statique.....	16
2.2	Page d'authentification	22
2.3	Interface d'accueil de visiteur	23
2.4	Interface de profil instructeur	23
2.5	Diagramme de cas d'utilisation global	32
2.6	Architecture en couches	39
2.7	Diagramme de paquetage	39
3.1	Diagramme de cas d'utilisation détaillé du premier sprint.....	54
3.2	Diagramme de classes du premier sprint	63
3.3	Diagramme de séquence du cas d'utilisation « S'authentifier »	65
3.4	Diagramme de séquence du cas d'utilisation « Ajouter un membre » ..	66
3.5	Diagramme de séquence du cas d'utilisation « Consulter profil »	67
3.6	Diagramme de séquence du cas d'utilisation « Consulter statistiques »	69
3.7	Interface d'authentification.....	70
3.8	Interface de gestion des membres	70
3.9	Interface de création d'un nouveau membre	71
3.10	Interface de confirmation de nouveau compte	71
3.11	Interface du tableau de bord du gestionnaire	72
3.12	Interface de gestion du profil du gestionnaire.....	72
3.13	Interface d'ajout d'un nouveau sujet	73
3.14	Interface de gestion des sujets	73
3.15	Test unitaire du service : création de sujet.....	74
3.16	Test unitaire du contrôleur : récupération des derniers sujets	74
3.17	Test Postman : création d'un sujet	75
4.1	Diagramme de cas d'utilisation détaillé du sprint 2	94
4.2	Diagramme de classes du second sprint	100
4.3	Diagramme de séquence du cas d'utilisation « Changer mot de passe »	101
4.4	Diagramme de séquence du cas d'utilisation « Ajouter un cours » ..	102
4.5	Diagramme de séquence du cas d'utilisation « S'inscrire à un cours ».	104
4.6	Diagramme de séquence du cas d'utilisation « Supprimer une leçon »	105

TABLE DES FIGURES

4.7	Diagramme de séquence du cas d'utilisation « Créer un quiz »	106
4.8	Interface de gestion de la confidentialité	107
4.9	Interface tableau de bord instructeur	108
4.10	Interface de gestion des cours	108
4.11	Interface d'ajout d'un cours	109
4.12	Interface d'ajout de quiz avec la sélection des questions déjà existantes	109
4.13	Interface d'ajout de question à un quiz	110
4.14	Interface de consultation des ressources pour les candidats.....	111
4.15	Interface de consultation des cours	112
4.16	Interface de consultation des leçons pour les candidats	112
4.17	Interface de passage de quiz pour candidat	113
4.18	Test unitaire du service : mise à jour d'un cours avec un identifiant invalide	114
4.19	Test unitaire du contrôleur : inscription à un cours - échec.....	114
4.20	Test Postman : modification du mot de passe	115
5.1	Diagramme de cas d'utilisation détaillé du Sprint 3 »	127
5.2	Diagramme de classes du troisième sprint	132
5.3	Diagramme de séquence du cas d'utilisation « Créer une discussion »	133
5.4	Diagramme de séquence du cas d'utilisation « Créer une réunion de planification ».....	135
5.5	Interface d'ajout d'un message.....	137
5.6	Interface de gestion des réponses de l'instructeur aux messages de ses candidats dans les forums de discussion	137
5.7	Interface de réponse du candidat dans le forum de discussion	138
5.8	Interface du calendrier de planification des réunions par l'instructeur .	138
5.9	Interface d'ajout d'une planification de réunion par l'instructeur	139
5.10	Email de notification pour une réunion planifiée	139
5.11	Interfaces de rejoindre une réunion	140
5.12	Interface de réunion instantanée pour l'instructeur	140
5.13	Test unitaire du service : création de réunion avec cours introuvable ..	141
5.14	Test unitaire du contrôleur : mise à jour d'une réponse.....	141
5.15	Test postman : mise à jour d'une réponse à un message	142

Liste des tableaux

1.1	Fiche technique de LeaderPos.....	3
1.2	Analyse comparative des plateformes : Udemy, Coursera et W3Schools	6
1.3	Tableau comparatif : avantages et inconvénients des plateformes d'apprentissage en ligne	8
2.1	Description des acteurs de la plateforme d'apprentissage	15
2.2	Exigences Fonctionnelles	17
2.3	Backlog de Produit	25
2.4	Planification des Sprints.....	33
2.5	Environnement Matériel.....	34
3.1	Backlog Du Premier Sprint	42
3.2	Description textuelle du cas d'utilisation « S'authentifier »	55
3.3	Description textuelle du cas d'utilisation « Ajouter Membre ».....	56
3.4	Description textuelle du cas d'utilisation « Modifier informations de profil ».....	58
3.5	Description textuelle du cas d'utilisation « Télécharger photo de profil »	59
3.6	Description textuelle du cas d'utilisation « Ajouter sujet »	60
3.7	Description textuelle du cas d'utilisation « Modifier sujet »	60
3.8	Description textuelle du cas d'utilisation « Supprimer sujet »	61
4.2	Description textuelle du cas d'utilisation « Modifier mot de passe » ..	95
4.3	Description textuelle du cas d'utilisation « Ajouter cours »	96
4.4	Description textuelle du cas d'utilisation « Modifier leçon »	97
4.5	Description textuelle du cas d'utilisation « Passer un quiz »	98
4.6	Description textuelle du cas d'utilisation « Ajouter un quiz avec sélection de questions ».....	99
5.2	Description textuelle du cas d'utilisation « Créer un planning de réunion »	128
5.3	Description textuelle du cas d'utilisation « Rejoindre une réunion » ..	129
5.4	Description textuelle du cas d'utilisation « Créer une discussion »	129
5.5	Description textuelle du cas d'utilisation « Ajouter une réponse »	130

Remerciement

Tout d'abord, nous remercions Allah le tout puissant de nous avoir donné le courage et la patience nécessaires pour mener ce travail à son terme.

Nous tenons à remercier tout particulièrement notre encadrante **Mme. Nahla Sassi**, pour l'aide compétente qu'elle nous a apportée, pour sa patience et son encouragement. Son oeil critique nous a été très précieux pour structurer le travail et améliorer la qualité des différentes sections.

Nous souhaitons également exprimer notre profonde gratitude à nos collègues pour leur soutien continu et leurs encouragements tout au long de ce projet. Leur aide précieuse et leurs précieux conseils ont grandement contribué à la réalisation de ce travail.

Nous adressons nos sincères remerciements à nos parents, pour leur amour, leur soutien indéfectible et leurs sacrifices constants. Leur encouragement nous a donné la force et la motivation nécessaires pour mener à bien ce projet.

Que les membres du jury trouvent ici l'expression de notre reconnaissance pour l'honneur qu'ils nous font en prenant le temps de lire et d'évaluer ce travail.

Pour finir, nous souhaitons remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Dédicace

À mes chers parents,

Votre amour, votre soutien et vos encouragements ont été mes piliers tout au long de ce parcours.

Cette réussite est le fruit de notre amour et de notre unité.

Je vous dédie ce travail avec toute ma gratitude et mon amour éternel.

À mes sœurs,

Nourhen et Amani, Merci pour votre amour, leur tendresse et leur soutien sans faille. Votre présence et vos encouragements ont été essentiels à la réalisation de ce travail. Cette réussite est le fruit de notre amour et de notre unité. Je vous dédie ce travail avec toute ma gratitude et mon amour éternel.

À mes amis,

Merci pour votre présence, vos encouragements et les moments de joie partagés.

Votre amitié a été un soutien précieux tout au long de ce chemin.

Ikram AYOUN

Dédicace

C'est avec un immense plaisir et une profonde reconnaissance que je dédie ce modeste travail

À mon père, pour son soutien inébranlable, ses conseils avisés et son encouragement constant tout au long de mes études. Que Dieu lui accorde santé et bonheur.

À ma mère, dont l'amour et la dévotion inépuisables ont toujours été une source d'inspiration. Que Dieu lui procure une bonne santé et une longue vie.

À mes sœurs, Sahar et Yasmine, et à mon frère, Wissem, pour leur amour, leur tendresse et leur soutien indéfectible. Votre présence et vos encouragements ont été essentiels à la réalisation de ce travail.

À tous ceux que j'aime, pour leur amour et leur soutien.

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

Syrine ELAYEB

Introduction générale

Dans un contexte de transformation permanente, l'éducation et l'apprentissage continuent de jouer un rôle crucial dans le développement individuel et collectif. Aujourd'hui, à l'arrivée des technologies numériques, l'accès à l'éducation est devenu plus accessible que jamais, transformant les méthodes traditionnelles et ouvrant de nouvelles perspectives pour l'apprentissage. En particulier, l'éducation en ligne a émergé comme une solution viable et innovante pour répondre aux besoins croissants de flexibilité et d'accessibilité.

C'est dans cette optique que notre projet s'inscrit avec la création de la plate-forme d'apprentissage en ligne EduCare. En collaboration avec l'Institut Supérieur des Études Technologiques (ISET) et sous la supervision de notre encadrante, nous avons entrepris de développer une solution éducative moderne et interactive. Ce rapport constitue le cadre de notre Projet de Fin d'Études (PFE), délimitant notre exploration autour d'une problématique centrale : comment concevoir une plateforme d'apprentissage en ligne efficace et adaptative, capable de répondre aux besoins diversifiés des apprenants du XXI^e siècle ? Conscients de l'importance académique et professionnelle de ce projet, nous sommes investis avec détermination dans ce projet.

Notre objectif est de fournir un outil qui non seulement facilite l'accès à l'éducation, mais aussi améliore la qualité de l'apprentissage grâce à des fonctionnalités personnalisées et interactives.

Nous avons choisi de diviser ce rapport sur cinq chapitres principaux, chacun explorant des aspects essentiels de notre projet. Nous commencerons par analyser le contexte général du projet, en examinant les tendances actuelles de l'apprentissage en ligne et en identifiant les besoins spécifiques de notre public cible. Ensuite, nous détaillerons notre méthodologie de planification et d'architecture, établissant les fondations d'EduCare. Les chapitres suivants seront consacrés aux réalisations et aux enseignements tirés de chaque sprint de développement. Du premier sprint, où les idées prennent forme, au troisième sprint, où la plateforme atteint son plein potentiel, nous exposerons nos succès, nos défis et nos solutions. En conclusion, nous évaluerons l'impact de notre projet et proposerons des perspectives d'amélioration pour l'avenir.

Chapitre 1

Contexte Général Du Projet

Plan

Introduction	3
1.1 Présentation de la société	3
1.1.1 Description de la société.....	3
1.1.2 Logo de la société	3
1.1.3 Informations de contact de la société.....	3
1.1.4 Services de la société.....	4
1.1.5 Organigramme fonctionnel de la société.....	5
1.2 Étude de l'existant.....	5
1.2.1 Description de l'existant	5
1.2.2 Critique de l'existant	8
1.2.3 Solution proposée	10
1.3 Présentation du projet.....	10
1.4 Langage et méthodologie de conception.....	11
1.5 Pourquoi Scrum	11
1.6 Choix des technologies	13
Conclusion	13

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. .

1.1 Présentation de la société

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.

1.1.1 Description de la société

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.

1.1.2 Logo de la société



FIGURE 1.1 – Logo de la société LeaderPos

1.1.3 Informations de contact de la société

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

1.1.4 Services de la société

— 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.

1.1.5 Organigramme fonctionnel de la société

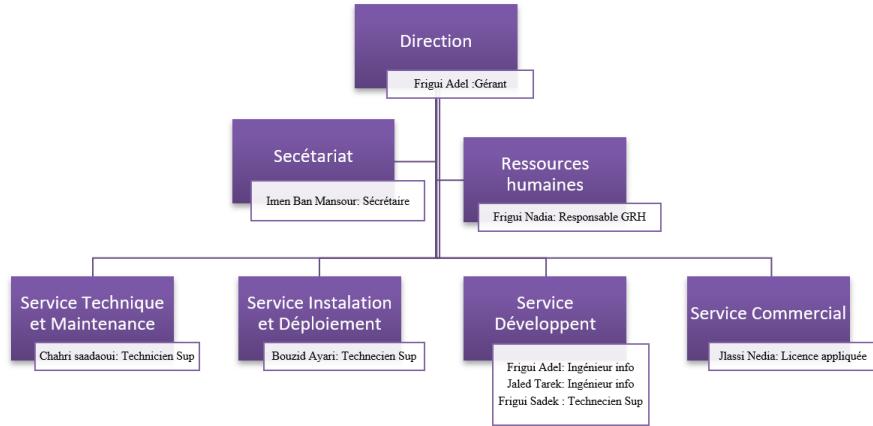


FIGURE 1.2 – Organigramme fonctionnel de LeaderPos

1.2 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.

1.2.1 Description de l'existant

- **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. *SHAH*

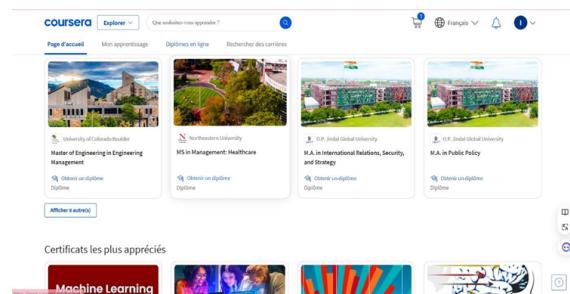


FIGURE 1.3 – Page d'accueil de la plateforme Coursera

- **Udemy** 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.
- WIKIPÉDIA*

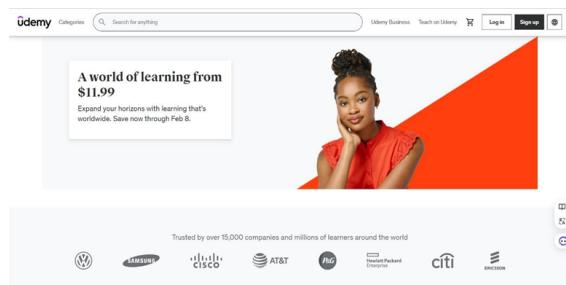


FIGURE 1.4 – Page d'accueil de la plateforme Udemy

- **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.
- CHATGPT*



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	✗	✗

Critères	Coursera	Udemy	W3Schools
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	✗
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.	✗

1.2.2 Critique 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. 	<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. 	<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.

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. 	<ul style="list-style-type: none"> - Aucune certification pour les cours gratuits. - Pas d'accréditation institutionnelle. - Coût élevé. - Difficulté à suivre plusieurs cours simultanément. - Cours obligatoires non disponibles. - Dates fixes pour les cours. 	<ul style="list-style-type: none"> - Impossible de suivre les cours complets gratuitement. - Besoin de discernement entre informations précises et obsolètes. - Manque de support direct. - 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.
--------------------------	---	---	--

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.
-------------------	---	---	---

1.2.3 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.

1.3 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.

1.4 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.

1.5 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 aiguille les membres de l'équipe, les repositionne dans la bonne direction et donne le tempo pour assurer la réussite du projet. » *AUBRY*

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.

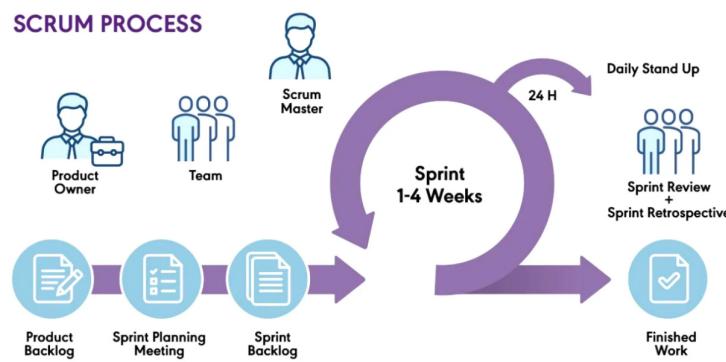


FIGURE 1.6 – Processus Scrum

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.

1.6 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.

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.

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.

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.

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

Plan

Introduction	15
2.1 Analyse des besoins	15
2.1.1 Identification des acteurs	15
2.1.1.1 Diagramme de contexte statique	16
2.1.2 Les besoins fonctionnels	16
2.1.3 Les besoins non fonctionnels	20
2.2 Planning du traitement de cas d'utilisation	21
2.2.1 Priorités	21
2.2.2 Risques	21
2.3 Prototypage des interfaces	22
2.4 Pilotage du projet avec Scrum	24
2.4.1 Les outils Scrum	24
2.4.2 Équipe et rôles	24
2.4.3 Le backlog du produit	24
2.4.4 Diagramme de cas d'utilisation global	31
2.5 Planification des sprints	33
2.6 Environnement du travail	34
2.6.1 Environnement matériel	34
2.6.2 Environnement logiciel	35
2.7 Architecture de l'application	38
2.7.1 Architecture logique	38
2.7.2 Architecture physique	39
Conclusion	39

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.

2.1 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 aisés de concevoir et de développer une solution répondant aux besoins de tous les intervenants.

2.1.1 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. » ROQUES et VALLÉ 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 :

TABLE 2.1 – Description des acteurs de la plateforme d'apprentissage

Acteur	Description
Visiteur	Un utilisateur non authentifié qui peut naviguer sur la plateforme pour explorer les sujets du cours, les cours disponibles. Ne peut pas accéder aux leçons, quiz ou autres fonctionnalités réservées aux utilisateurs authentifiés. Il a la possibilité de contacter le gestionnaire de la plateforme.
Gestionnaire	Un membre authentifié chargé de la gestion globale de la plateforme. Crée, modifie et supprime des sujets pour organiser les contenus pédagogiques. Il 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.

2.1.1.1 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.

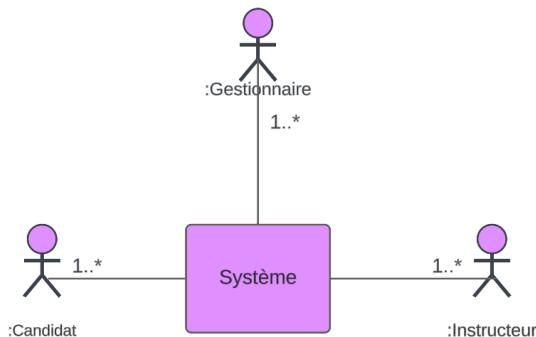


FIGURE 2.1 – Diagramme de contexte statique

2.1.2 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	<p>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> <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>Planifier des 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>
--	---

	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.
Candidat	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.
	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.
	Consulter les sujets : les candidats doivent pouvoir consulter les sujets disponibles sur la plateforme pour explorer les domaines d'apprentissage qui les intéressent.
	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.
	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.
	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.

	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.
	Consulter ou passer des quiz : les candidats doivent pouvoir consulter, passer des quiz associés aux cours. Pour évaluer leur compréhension et tester leurs connaissances

2.1.3 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.

2.2 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.

2.2.1 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/faible) afin d'organiser efficacement nos efforts de développement.

2.2.2 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.

2.3 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

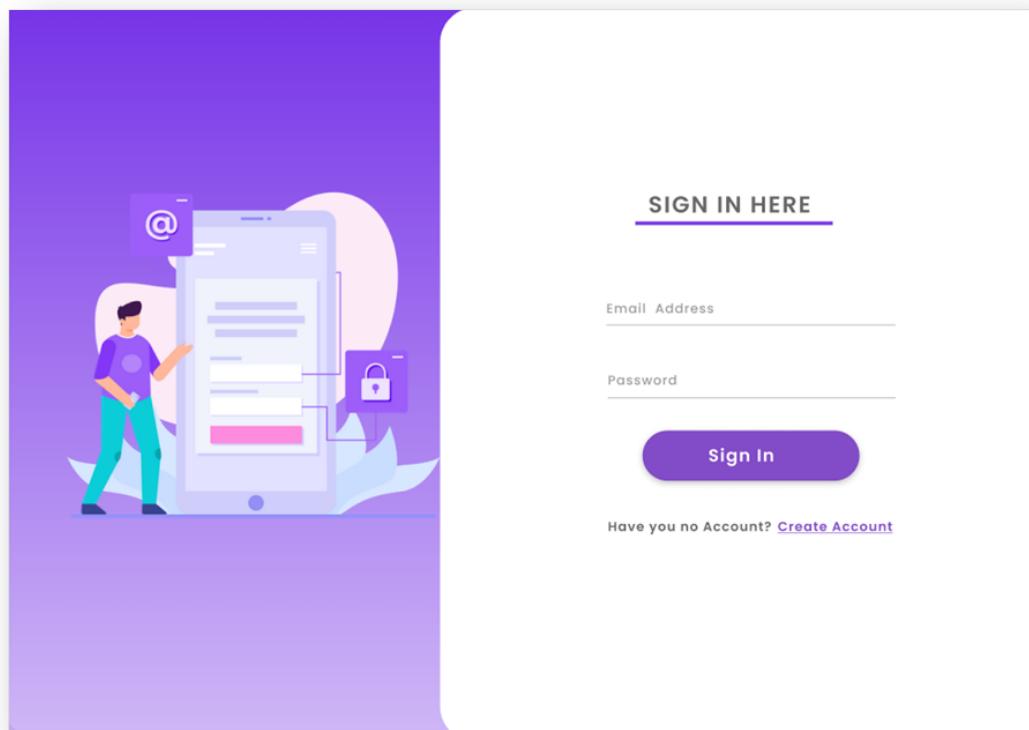


FIGURE 2.2 – Page d'authentification

The screenshot shows the visitor homepage of the eduCare platform. At the top, there is a navigation bar with the eduCare logo, a search bar, and buttons for 'Login' and 'Sign up for free'. Below the navigation bar, a large purple banner features the text 'Studying Online is now much easier' and a subtext: 'eduCare is an interesting platform that will teach you in more an interactive way'. A circular profile picture of a woman working on a laptop is displayed. To the left of the banner, there is a 'Contact Us' form with fields for Name, Email, and Message, and a 'Submit' button. To the right of the banner, there is an illustration of a smartphone displaying a 'CONTACT US' screen with a 24-hour support icon. Below the banner, there is a footer section with social media icons (Facebook, Instagram, LinkedIn, YouTube), a phone number (+2034 4040 3030), an email address (hello@gmail.com), and links to 'Home', 'About', 'Service', and 'Contact Us'.

FIGURE 2.3 – Interface d'accueil de visiteur

The screenshot shows the profile page for an instructor named John Doe. The page has a sidebar with a purple header containing the user's profile picture and name, and a list of navigation items: Dashboard, Schedule, Manage course, Create Quiz, Meet, Profile & Settings, and Forum. The main content area displays personal details (First name: John, Last name: Doe, Gender: Select Gender, Country: Select Country), contact information (Email: John@gmail.com, Phone number: +216 50 767 872, GitHub username: john_doe, LinkedIn username: john_doe04, Twitter username: john_doe04), and a biography section. The biography text reads: 'Jane Smith is a passionate and dedicated educator with a strong focus on web design. With 8 years of experience, she has become a specialist in guiding students through the intricacies of creating visually stunning and user-friendly websites.' At the bottom of the page is an 'Edit profile' button.

FIGURE 2.4 – Interface de profil instructeur

2.4 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.

2.4.1 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 répondues, 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.

2.4.2 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 ». 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.

2.4.3 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	S'inscrire	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.	2	Moyenne
2	S'authentifier	US002 En tant que membre, je veux pouvoir me connecter de manière sécurisée pour garantir la confidentialité de mes données .	2	Moyenne
3	Gestion des profils	US003 En tant que membre, je veux pouvoir consulter mon profil pour visualiser mes informations personnelles.	2	Moyenne
		US004 En tant que membre, je veux pouvoir modifier mes informations personnelles pour les mettre à jour.	1	Faible
		US005 En tant que membre, je veux pouvoir supprimer mon compte si nécessaire.	1	Faible
4	Gestion des sujets	US009 En tant que gestionnaire, je veux pouvoir ajouter un sujet.	2	Moyenne

		US010 En tant que gestionnaire, je veux pouvoir modifier un sujet.	1	Faible
		US011 En tant que gestionnaire, je veux pouvoir supprimer un sujet.	1	Faible
		US012 En tant que visiteur, je souhaite pouvoir consulter les différents sujets, rechercher des sujets et accéder aux cours associés à ces sujets.	1	Haute
		US013 En tant que candidat, je veux pouvoir consulter les sujets récents.	1	Haute
5	Gestion des comptes membre	US014 En tant que gestionnaire, je veux pouvoir ajouter un compte membre à la plateforme afin d'élargir la base d'utilisateurs.	2	Moyenne
		US015 En tant que gestionnaire, je veux pouvoir supprimer un compte membre de la plateforme si nécessaire.	1	Faible
		US016 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.	2	Moyenne
		US017 En tant que membre, je souhaite recevoir mes informations de connexion dans ma boîte mail.	2	Faible
6	Gestion des cours	US018 En tant qu'instructeur, je veux créer un nouveau cours assigné à un sujet pour offrir une structure claire aux candidats.	3	Moyenne
		US019 En tant qu'instructeur, je veux pouvoir modifier les détails d'un cours existant pour maintenir la pertinence et la qualité du contenu.	1	Faible
		US020 En tant qu'instructeur, je veux pouvoir supprimer un cours existant afin de gérer efficacement les ressources pédagogiques.	1	Faible

		US021 En tant qu'instructeur, je veux pouvoir suivre le progrès des candidats dans chaque cours pour évaluer leur performance et leur compréhension du contenu.	3	Faible
		US022 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.	3	Moyenne
		US023 En tant que candidat, je veux pouvoir consulter mes cours inscrits pour suivre ma progression et accéder facilement aux leçons.	3	Moyenne
		US024 En tant que visiteur, je veux pouvoir consulter les cours disponibles pour avoir un aperçu de l'offre de formation.	3	Moyenne
7	Gestion de la confidentialité	US006 En tant que membre, je veux pouvoir modifier mon mot de passe pour garantir la sécurité de mon compte.	1	Moyenne
		US007 En tant que membre, je veux pouvoir désactiver temporairement mon compte si besoin.	1	Faible
		US008 En tant que membre, je veux pouvoir supprimer définitivement mon compte pour effacer toutes mes données.	1	Faible
8	Gestion des Leçons	US025 En tant qu'instructeur, je veux créer une leçon pour un cours spécifié afin de structurer le contenu pédagogique.	3	Haute
		US026 En tant qu'instructeur, je veux pouvoir modifier les détails d'une leçon existante pour mettre à jour ou améliorer le contenu.	1	Faible
		US027 En tant qu'instructeur, je veux pouvoir supprimer une leçon existante afin de gérer efficacement le programme du cours.	1	Faible

		US028 En tant que candidat, je veux pouvoir accéder aux leçons des cours auxquels je suis inscrit afin de suivre le contenu et mon progress dans ma formation.	3	Moyenne
9	Gestion des ressources	US029 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.	2	Moyenne
		US030 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.	1	Faible
		US031 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.	1	Faible
		US032 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.	2	Moyenne
10	Gestion des quiz	US033 En tant qu'instructeur, je veux pouvoir créer un quiz interactif pour chaque cours afin d'ajouter des questions et des réponses à ce quiz pour évaluer les connaissances de mes candidats.	3	Haute
		US034 En tant qu'instructeur, je souhaite pouvoir modifier les détails d'un quiz déjà créé.	1	Faible
		US035 En tant qu'instructeur, je veux pouvoir supprimer un quiz qui n'est plus pertinent ou nécessaire.	1	Faible

		US036 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.	3	Faible
		US037 En tant qu'instructeur ou candidat, je veux pouvoir consulter la liste des quiz disponibles sur la plateforme.	3	Moyenne
		US038 En tant que candidat, je veux pouvoir participer aux quiz attribués à mes cours.	2	Haute
		US039 En tant que candidat, je souhaite pouvoir consulter mes propres résultats après avoir terminé un quiz.	2	Moyenne
11	Plannification des réunions	US040 En tant qu'instructeur, je souhaite planifier des réunions pour mes cours en spécifiant une date, une heure et éventuellement un sujet, et je veux que les candidats inscrits soient notifiés automatiquement par email lors de l'ajout d'une nouvelle réunion.	4	Haute
		US041 En tant qu'instructeur, je souhaite pouvoir modifier les détails d'une réunion déjà planifiée, tels que la date, l'heure ou le sujet, et je veux que les candidats inscrits soient informés des modifications par email.	2	Moyenne
		US042 En tant qu'instructeur, je veux pouvoir annuler une réunion planifiée pour mes cours, et je veux que les candidats inscrits soient informés de l'annulation par email.	2	Moyenne

		US043 En tant que candidat, je souhaite pouvoir consulter les réunions planifiées pour les cours auxquels je suis inscrit afin de rester informé des événements à venir, et je veux être notifié par email des nouvelles réunions ajoutées à ces cours, des modifications apportées aux réunions existantes ou des annulations de réunions.	3	Moyenne
		US044 En tant que candidat, je souhaite participer activement aux réunions en temps réel dans les cours auxquels je suis inscrit pour interagir avec l'instructeur et les autres candidats.	3	Haute
		US045 En tant que candidat, je veux être notifié par email des nouvelles réunions planifiées dans les cours auxquels je suis inscrit pour rester informé des événements à venir.	2	Moyenne
12	Gestion des forums de discussion	US046 En tant qu'instructeur, je souhaite pouvoir consulter les messages des candidats sur les forums de discussion de mes leçons afin de suivre les conversations et répondre aux messages pertinents pour offrir un soutien pédagogique. De plus, l'instructeur peut également répondre aux messages de ses candidats.	3	Haute
		US047 En tant qu'instructeur, je veux pouvoir éditer ou supprimer mes propres messages dans le forum de leçons pour corriger les erreurs ou retirer des informations obsolètes.	3	Moyenne

		US048 En tant que candidat, je souhaite avoir la possibilité de contribuer activement aux forums de discussion en ajoutant des messages sous forme de texte, de PDF, d'images, d'enregistrements ou de vidéos, en répondant aux messages existants, et en ayant la capacité de modifier ou supprimer mes propres réponses lorsque nécessaire pour maintenir la clarté et la pertinence des messages.	3	Haute
--	--	--	---	-------

2.4.4 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.

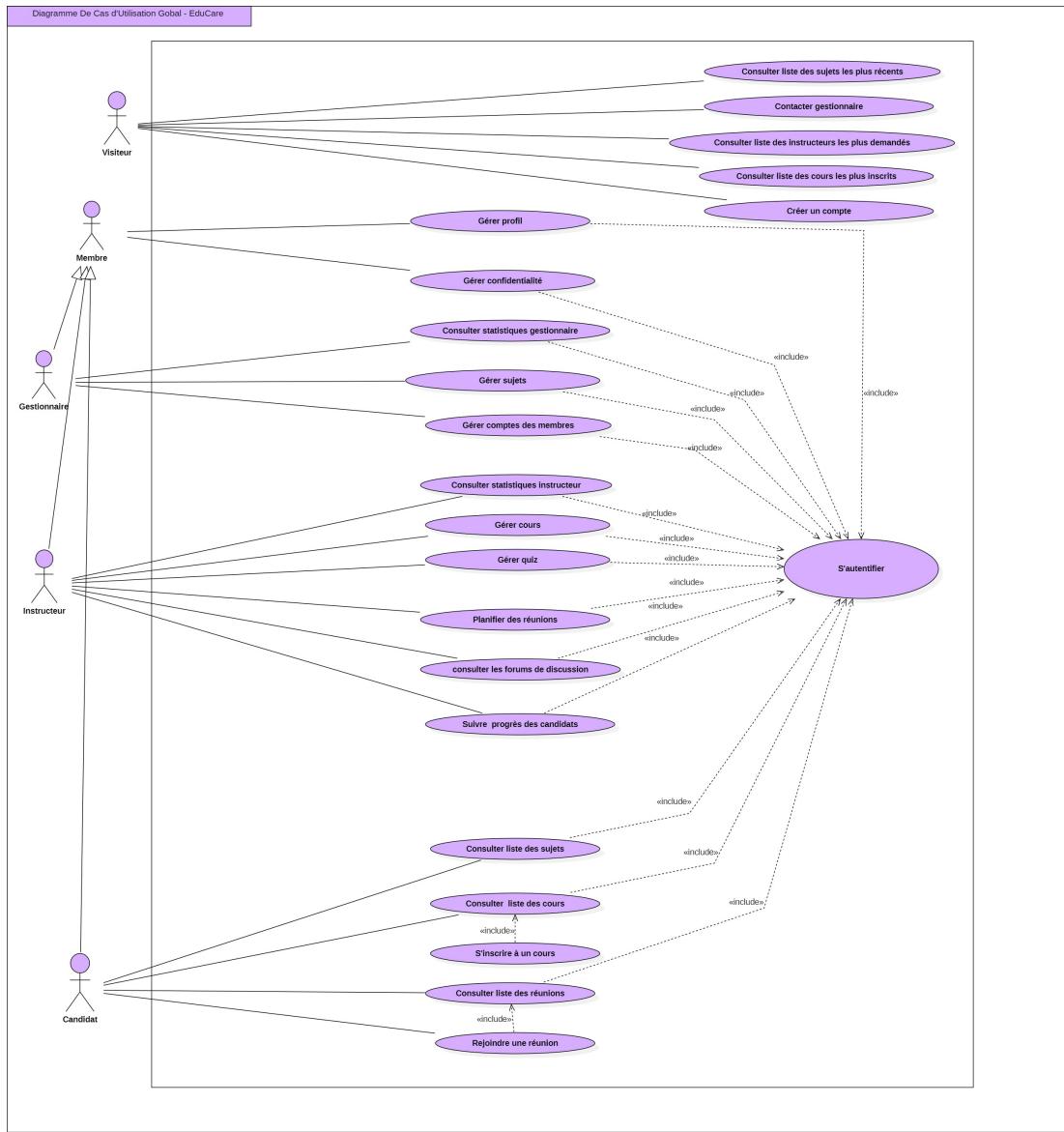


FIGURE 2.5 – Diagramme de cas d'utilisation global

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 visiteurs, une fois authentifiés, peuvent gérer leur profil. En outre, ils peuvent également gérer les paramètres de confidentialité de leur compte. 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, qui héritent également des fonctionnalités de base, peuvent consulter les statistiques, gérer les cours et les quiz, planifier des réunions, consulter les forums de discussion, et suivre les progrès des candidats. Les candidats, une fois authentifiés, peuvent consulter les sujets et les cours, s'inscrire à un cours, consulter la liste des réunions organisées par

les instructeurs, et participer à celles qui les intéressent. 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 Planification des sprints

La planification de sprint est une activité essentielle qui consiste à déterminer et à ordonner 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 – Planification des Sprints

ID	Story	Date Début	Date Fin	Nom Sprint
S1	Inscription (ID 1) – Authentification (ID 2) – Gestion des profils (ID 3), sujets (ID 4) – Gestion des comptes membres (ID 5).	20/02/2024	11/03/2024	Mise en Place des Fonctions Utilisateur et Sujets

ID	Story	Date Début	Date Fin	Nom Sprint
S2	Gestion des cours (ID 6) - Gestion de la confidentialité (ID 7) - Gestion des leçons (ID 8) - Gestion des ressources (ID 9) - Gestion des quiz (ID 10).	12/03/2024	02/05/2024	Gestion Pédagogique et Confidentialités
S3	Plannifications des réunions (ID 11) - Gestion des forum de discussion (ID 12).	03/05/2024	28/05/2024	Mise en Place du Forum de Discussion et Plannification des Réunions

2.6 Environnement du travail

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.

2.6.1 Environnement matériel

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

TABLE 2.5 – Environnement Matériel

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

2.6.2 Environnement logiciel

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

— 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. *SIMPLILEARN*

— 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. *WIKIPEDIA*



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 plate-forme 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. *WIKIPEDIA*

— 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. *WIKIPEDIA*



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. *JUNIT*

— 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.

— Système de Gestion de Base de Données



MySQL : MySQL est la base de données open source la plus populaire au monde. Elle se classe comme la deuxième base de données la plus populaire selon DB-Engines, juste derrière Oracle Database.

— 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).

— 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.).

— 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.

— Outil de Conception et de Prototypage (Figma)



Figma est un outil de conception web polyvalent qui permet la collaboration en temps réel pour créer des prototypes d'interfaces utilisateur interactives. *DESKTOP*

— Langages de programmation

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



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.



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.

— Protocole de Messagerie Simple (SMTP)



Le SMTP (Simple Mail Transfer Protocol) est un protocole standard de communication utilisé pour transférer des e-mails d'un serveur à un autre, jouant le rôle d'expéditeur lors de l'envoi de messages électroniques.

Il utilise le protocole TCP pour le transfert des données et ne prend pas en charge la récupération à distance des e-mails dans une boîte mail sur un serveur.

— Outil de Conférence en Ligne (Jitsi)



Jitsi : Jitsi est une plateforme de visioconférence en ligne open source qui permet aux utilisateurs de tenir des réunions virtuelles, des conférences et des discussions en temps réel.

Développée par une communauté de développeurs passionnés, Jitsi offre une alternative gratuite et sécurisée aux solutions de visioconférence propriétaires. Il prend en charge diverses fonctionnalités telles que la vidéo en direct, le partage d'écran, le chat texte et la collaboration en temps réel, offrant ainsi une expérience de communication riche et interactive pour les utilisateurs distants.

2.7 Architecture de l'application

2.7.1 Architecture logique

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 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.

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.

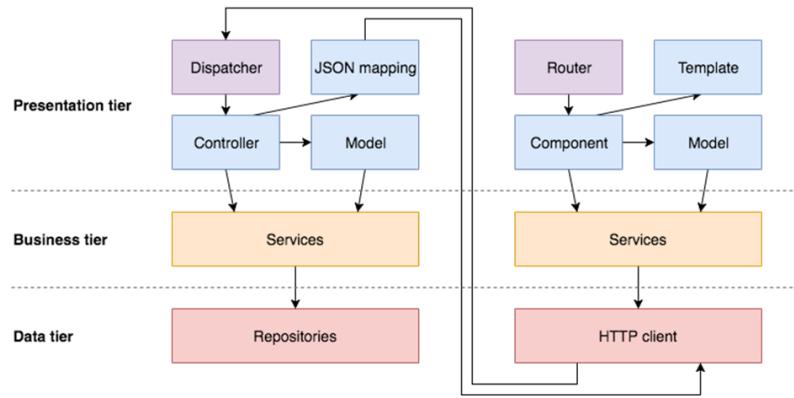


FIGURE 2.6 – Architecture en couches

2.7.2 Architecture physique

— Diagramme de paquetage

Le diagramme de paquetage est un outil essentiel pour visualiser et organiser la structure modulaire d'une application. Il offre une vue claire de l'organisation des différents composants logiciels et de leurs relations, ce qui facilite la compréhension des responsabilités de chaque module et améliore la communication au sein de l'équipe de développement. En examinant ce diagramme, on peut aisément identifier les liens entre les modules, les interfaces qu'ils présentent et les flux de données dans le système, ce qui favorise la création d'une architecture logicielle robuste et modulaire. Cette approche renforce la durabilité, la réutilisabilité et l'évolutivité de l'application.

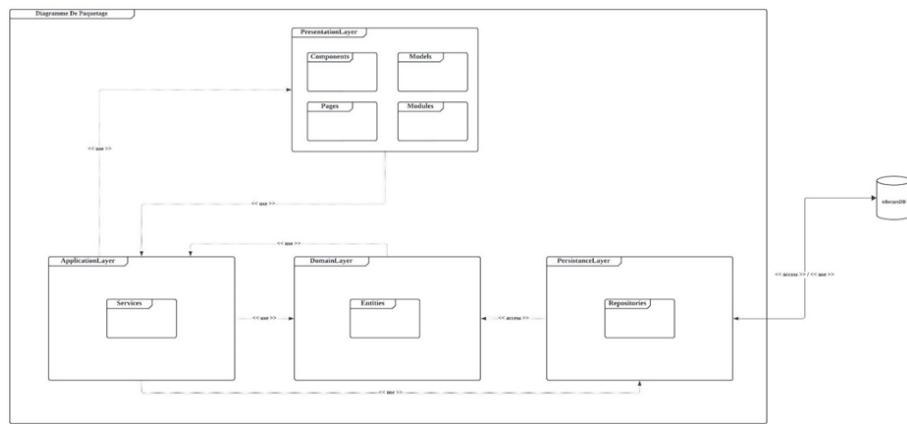


FIGURE 2.7 – Diagramme de paquetage

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

Premier Sprint : Mise en Place des Fonctions Utilisateur et Sujets

Plan

Introduction	41
3.1 Le premier sprint	41
3.1.1 Spécification fonctionnelle	53
3.1.1.1 Raffinement de cas d'utilisation du sprint 1	53
3.1.1.2 Diagramme de cas d'utilisation détaillé du sprint1	53
3.1.1.3 Description textuelle de cas d'utilisation.....	55
3.1.2 La modélisation conceptuelle.....	62
3.1.2.1 Diagramme de classes	62
3.1.2.2 Diagramme de séquence détaillé.....	63
3.1.3 Réalisation	69
3.1.4 Tests	73
3.1.4.1 Tests unitaires	74
3.1.4.2 Tests avec Postman	75
Conclusion	76

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 "Mise en Place des Fonctions Utilisateur et Sujets". Chaque User story suivra les quatre étapes du cycle Scrum : spécification fonctionnelle, conception, réalisation et tests.

3.1 Le premier sprint

Le sprint représente l'essence de Scrum, définissant une période de temps spécifique durant laquelle une version améliorée du produit sera développée. Les sprints au sein d'une release ont une durée constante et ne se chevauchent pas, ce qui signifie 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. La planification du sprint 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 les avons décomposées 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 user stories seront incluses. 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 l'user story	Tâches effectuées	Estimation
1	En tant que candidat, instructeur ou gestionnaire, 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) 	2 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. (Tests avec Postman) - Créer un formulaire d'enregistrement avec les champs suivants : prénom, nom, email, mot de passe. (Frontend) - Valider les informations saisies. (Frontend) 	2 jours

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. (Tests avec Postman)- 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)	1 jours
---	---	--	---------

4	<p>En tant que membre, je veux pouvoir modifier mon profil utilisateur.</p>	<ul style="list-style-type: none"> - Créer un 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. (Tests avec Postman) - 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 une fonctionnalité dans le formulaire de modification du profil qui permet à l'utilisateur de télécharger une image de profil. Cela doit inclure la possibilité de sélectionner une image depuis l'explorateur de fichiers de son appareil, afin de modifier l'image de profil directement dans le même formulaire. (Frontend). (Frontend) - Afficher un message d'erreur approprié en cas d'échec de la modification du profil. (Frontend) 	1 jour
---	---	---	--------

5	En tant que gestionnaire, 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. (Tests avec Postman) - Créer un formulaire d'ajout d'instructeur ou candidat avec les champs : nom, prénom, email, mot de passe et rôle. (Frontend) 	1 jours
6	En tant que gestionnaire, 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. (Tests avec Postman) - 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) 	1 jours

7	<p>En tant que gestionnaire, je veux pouvoir supprimer un compte d'instructeur ou de candidat existant depuis la liste des utilisateurs.</p>	<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 suppression. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Créer une interface utilisateur conviviale pour afficher la liste des comptes d'instructeurs et de candidats. (Frontend) - Associer une action de suppression permettant ainsi aux gestionnaires de supprimer un compte utilisateur en un clic. Le bouton de suppression sera activé seulement si le statut de l'utilisateur est inactif. Le bouton "Delete" sera cliquable uniquement si le statut de l'utilisateur est "Inactive". À ce moment-là, le gestionnaire pourra supprimer le compte utilisateur si nécessaire. (Frontend) 	1 jours
---	--	---	---------

8	En tant que gestionnaire, je veux pouvoir filtrer 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. (Tests avec Postman) - Afficher les résultats de la filtre dans l'interface utilisateur. (Frontend) 	1 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. (Tests avec Postman) - Créer une interface utilisateur conviviale pour afficher la liste des sujets de cours. (Frontend) 	1 jours

10	En tant que gestionnaire, 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. (Tests avec Postman) - Créer une interface utilisateur pour permettre aux gestionnaire 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 que gestionnaire, 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. (Tests avec Postman) - Associer une action de modification à chaque icône de modification de sujet de cours dans l'interface utilisateur de la liste. (Frontend) - Valider les modifications saisies avant de les envoyer au backend pour modification. (Frontend) 	1 jours

12	En tant que gestionnaire, 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. (Tests avec Postman) - 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) 	1 jours
13	En tant que gestionnaire, je veux pouvoir filtrer un sujet de cours existant.	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour permettre le filtrage de sujets de cours existants. (Backend) - Gérer les erreurs lors de l'opération de filtrage et renvoyer des messages d'erreur appropriés en cas de problème. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Ajouter un champ de filtrage dans l'interface utilisateur pour permettre au gestionnaire de saisir des mots-clés ou des critères de filtrage. (Frontend) 	1 jours
14	En tant que gestionnaire, je veux pouvoir déconnecter	<ul style="list-style-type: none"> - Implémentez une logique pour invalider la session ou supprimer le JWT Token associé au gestionnaire. (Frontend) - Ajoutez un bouton de déconnexion dans l'interface utilisateur, généralement dans le navbar ou dans un menu déroulant. (Frontend) 	0 jour

15	En tant que gestionnaire, je veux pouvoir consulter les statistiques suivantes : nombre de membres actifs, nombre de membres inactifs, nombre total de membres, nombre d'instructeurs et nombre de candidats.	<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. (Tests avec Postman) - 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) 	1 jours
16	En tant que visiteur, je veux pouvoir consulter les sujets les plus récents	<ul style="list-style-type: none"> -Créer une route API pour récupérer les sujets les plus récents. . (Backend) - Implémenter la logique nécessaire pour récupérer les sujets les plus récents depuis la base de données. (Backend) - Gérer les erreurs éventuelles lors de la récupération des sujets récents. (Backend) - Tester la route API pour s'assurer qu'elle retourne les résultats attendus.(Tests avec Postman) - Créer une interface utilisateur pour afficher les sujets les plus récents. (Frontend) <p>Intégrer la requête API pour récupérer les sujets les plus récents dans l'interface utilisateur(Frontend)</p> <p>Afficher les sujets de manière claire et conviviale pour les visiteurs(Frontend)</p>	1 jours

17	En tant que visiteur, je veux pouvoir consulter les instructeurs les plus actifs	<ul style="list-style-type: none">- Créer une route API pour récupérer les instructeurs les plus actifs. (Backend)- Implémenter la logique nécessaire pour récupérer les instructeurs les plus actifs depuis la base de données. (Backend)- Gérer les erreurs éventuelles lors de la récupération des instructeurs les plus actifs. (Backend)- Tester la route API pour s'assurer qu'elle retourne les résultats attendus. (Tests avec Postman)- Créer une interface utilisateur pour afficher les instructeurs les plus actifs. (Frontend)- Intégrer la requête API pour récupérer les instructeurs les plus actifs dans l'interface utilisateur. (Frontend)- Afficher les instructeurs de manière claire et conviviale pour les visiteurs. (Frontend)	1 jour
----	--	---	--------

18	<p>En tant que visiteur, je veux pouvoir Contacter gestionnaire par email</p>	<ul style="list-style-type: none"> - Créer une route API pour permettre l'envoi d'email au gestionnaire. (Backend) - Implémenter la logique nécessaire pour gérer l'envoi d'email au gestionnaire. (Backend) - Valider les données reçues côté backend avant d'envoyer l'email pour éviter les injections de code ou les abus. (Backend) - Configurer les paramètres de messagerie pour permettre l'envoi d'email depuis l'application. (Backend) - Ajouter un formulaire de contact dans l'interface utilisateur pour collecter les informations de l'utilisateur et le message qu'il souhaite envoyer au gestionnaire. (Frontend) - Valider les données du formulaire côté frontend pour s'assurer de leur conformité avant l'envoi. (Frontend) - Intégrer la requête API pour envoyer l'email au gestionnaire lors de la soumission du formulaire. (Frontend) - Afficher un message de confirmation à l'utilisateur une fois l'email envoyé avec succès. (Frontend) 	1 jour
----	---	--	--------

19	<p>En tant que membre, je veux recevoir un email de bienvenue contenant les informations de connexion requises dès que le gestionnaire crée mon compte</p>	<ul style="list-style-type: none"> - Créer une route API pour envoyer un email de bienvenue lors de la création de compte. (Backend) - Implémenter la logique pour envoyer un email contenant les informations de connexion lorsque le gestionnaire crée un nouveau compte. (Backend) - Valider les informations du compte avant d'envoyer l'email de bienvenue. (Backend) - Configurer les paramètres de messagerie pour permettre l'envoi d'email de bienvenue. (Backend) - Tester la fonctionnalité d'envoi d'email de bienvenue pour s'assurer qu'elle fonctionne correctement. (Tests avec Postman) 	1 jour
----	--	---	--------

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, réalisation et le test. Tout au long de ce sprint, nous respectons ces activités pour construire le plan de notre travail.

3.1.1 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.

3.1.1.1 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**

3.1.1.2 Diagramme de cas d'utilisation détaillé du sprint1

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 :

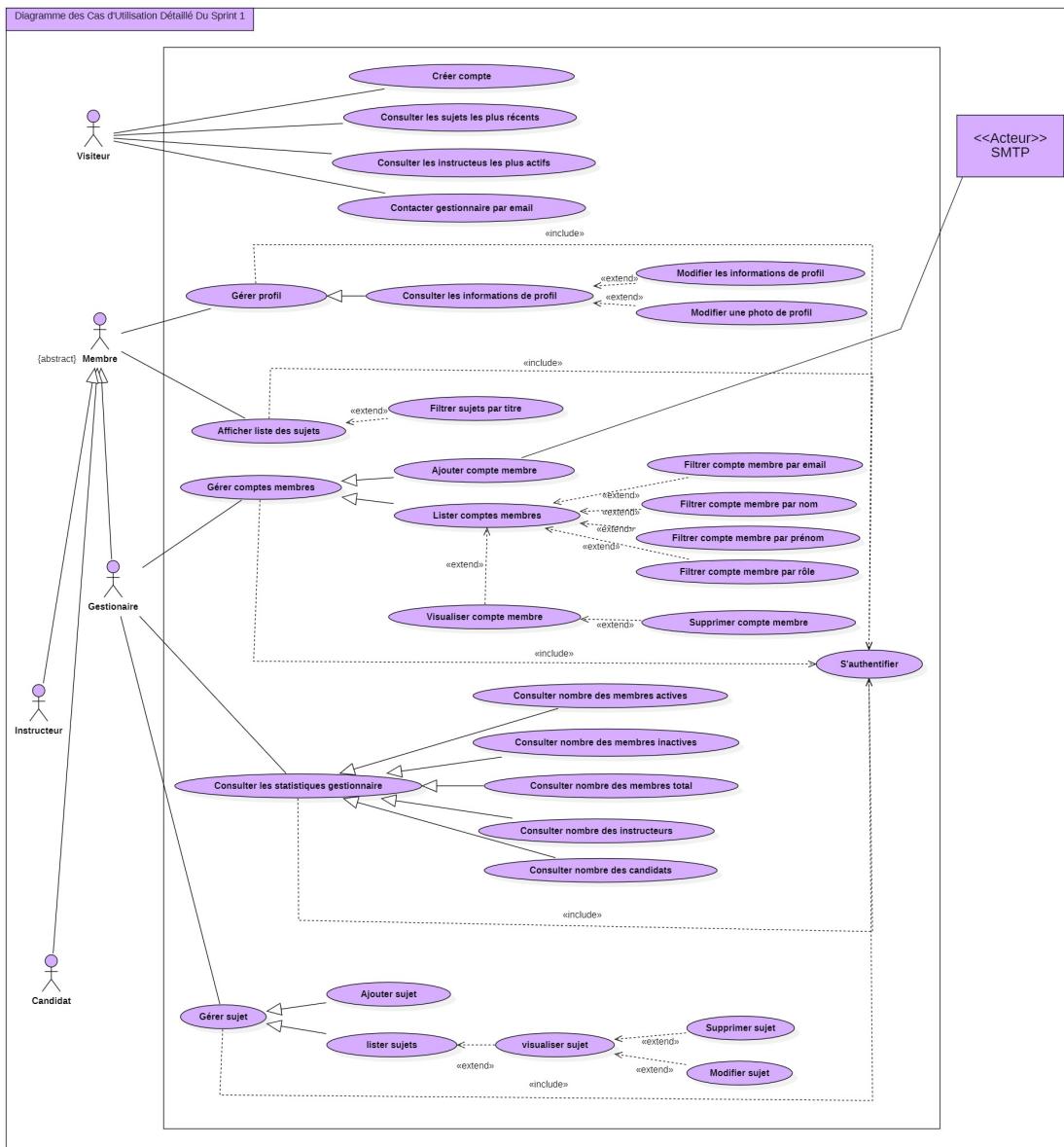


FIGURE 3.1 – Diagramme de cas d'utilisation détaillé du premier sprint

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. Lorsqu'il ajoute un utilisateur, un système de messagerie envoie un email de bienvenue contenant les informations de connexion. L'administrateur peut également attribuer un profil et gérer les 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.

3.1.1.3 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.
- **postcondition** : 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 « S'authentifier »

Cas d'utilisation	S'authentifier
Acteur principal	Membre
Précondition	Membre 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 membre
Scénario Nominal	1. Le membre accède à la page de connexion de l'application

	<ol style="list-style-type: none"> 2. Le système affiche le formulaire d'authentification 3. Le membre saisit son adresse e-mail et son mot de passe. 4. Le membre 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 membre. 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 membre. 9. L'accueil personnel de membre s'affiche, présentant ses fonctionnalités et informations spécifiques.
Scénario d'exception	<p>SE1. Champs vides. SE1 démarre au point 3 du scénario nominal.</p> <ol style="list-style-type: none"> 4. Le système affiche un message d'erreur si des champs requis sont laissés vides. <p>Le cas d'utilisation se termine en échec.</p> <p>SE2. Informations d'identification incorrectes. SE2 démarre au point 5 du scénario nominal.</p> <ol style="list-style-type: none"> 6. L'application indique une erreur si les informations d'identification fournies sont incorrectes. 7. Le membre est invité à corriger ses informations pour réessayer. <p>Le cas d'utilisation se termine en échec.</p>

Le tableau ci-dessous explique le processus d'ajout d'un membre par un gestionnaire, permettant à ce dernier de valider et d'effectuer l'ajout d'un nouveau membre dans le système.

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

Cas d'utilisation	Ajouter Membre
Acteur principal	Gestionnaire
Précondition	Le gestionnaire est authentifié et possède les autorisations nécessaires pour ajouter des membres.
Postcondition	Un nouveau membre est ajouté avec succès à l'application, avec le rôle spécifié.
Scénario Nominal	<ol style="list-style-type: none"> 1. Le gestionnaire accède à la fonctionnalité d'ajout de membre dans l'espace d'administration de la plateforme.

	<p>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.</p> <p>3. Le gestionnaire saisit les informations requises.</p> <p>4. Après avoir saisi les informations requises, le gestionnaire valide le formulaire.</p> <p>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.</p> <p>6. Si toutes les données saisies sont valides, le système enregistre les informations du membre dans la base de données.</p> <p>7. Le système affiche un message de confirmation indiquant que le membre a été ajouté avec succès.</p> <p>8. Une fois le membre ajouté, le système crée automatiquement un profil pour ce membre, en utilisant les informations fournies lors de l'ajout, telles que le nom, le prénom et le type d'utilisateur.</p> <p>9. La page de la liste des membres est rechargée, ajoutant ainsi le nouveau membre à la liste.</p> <p>10. Après l'ajout du nouveau membre, un e-mail contenant ses informations de connexion (nom d'utilisateur et mot de passe) est automatiquement envoyé au nouveau membre.</p>
Scénario d'exception	<p>SE1. Champs vides</p> <p>SE1 démarre au point 4 du scénario nominal.</p> <p>5. Le système détecte des champs obligatoires vides et affiche un message d'erreur indiquant que tous les champs doivent être remplis.</p>
Scénario alternatif	<p>SA1. Format d'e-mail incorrect ou mot de passe trop court</p> <p>SA1 démarre au point 5 du scénario nominal.</p> <p>6. Le système détecte ces erreurs lors de la vérification des données et affiche des messages d'erreur correspondants à chaque champ erroné.</p> <p>7. Le gestionnaire corrige les champs erronés.</p> <p>8. Une fois que les erreurs ont été corrigées, le gestionnaire soumet à nouveau le formulaire.</p> <p>9. Le système reprend le processus de vérification des données, comme il est décrit dans les étapes du scénario nominal.</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 3.4 – 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 et Sécurité" dans la barre latérale. 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, sa biographie, son emploi, 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	<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"> 6. Le système détecte que les nouvelles informations saisies sont incorrectes ou invalides. 7. Le système affiche un message d'erreur spécifique, indiquant la nature des erreurs ou des informations invalides.

	8. 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. Voici la correction :

TABLE 3.5 – Description textuelle du cas d'utilisation « Télécharger photo de profil »

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

	7. Le membre 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 d'ajouter un sujet de cours.

TABLE 3.6 – 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	<p>SA1 : Sujet déjà existant</p> <p>SA1 commence après l'étape 4 du scénario nominal.</p> <ol style="list-style-type: none"> 5. Le système détecte que le sujet que le gestionnaire tente d'ajouter existe déjà dans la base de données. 6. Le système affiche un message d'erreur indiquant que le sujet existe déjà.

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

TABLE 3.7 – 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. Le gestionnaire recherche et sélectionne le sujet qu'il souhaite modifier. 4. Le système affiche les détails du sujet sélectionné dans un formulaire modifiable. 5. Le gestionnaire 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 d'exception	<p>SE1 : Erreur lors de la modification SE1 survient après l'étape 7 du scénario nominal.</p> <ol style="list-style-type: none"> 8. Le système rencontre une erreur lors de la modification des données du sujet. 9. Le système affiche un message d'erreur indiquant que la modification a échoué en raison d'une erreur interne.

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

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

Cas d'utilisation	Supprimer sujet
Acteur principal	Gestionnaire
Précondition	Le Gestionnaire doit être authentifié et avoir les droits nécessaires pour accéder à la fonctionnalité de suppression d'un sujet.
Postcondition	Sujet Supprimé
Scénario Nominal	<ol style="list-style-type: none"> 1. Le gestionnaire accède à la fonctionnalité de suppression de sujet dans l'espace d'administration de la plateforme.

	<ol style="list-style-type: none"> 2. Le système affiche une liste des sujets existants dans un formulaire modifiable. 3. Le gestionnaire recherche et sélectionne le sujet qu'il souhaite supprimer. 4. Le système affiche les détails du sujet sélectionné. 5. Le gestionnaire confirme la suppression du sujet. 6. Le système supprime le sujet de la base de données. 7. Le système affiche un message de confirmation indiquant que le sujet a été supprimé avec succès.
Scénario d'exception	<p>SE1. Erreur lors de la suppression</p> <p>SE1 commence après l'étape 5 du scénario nominal.</p> <ol style="list-style-type: none"> 6. Le système rencontre une erreur lors de la suppression du sujet de la base de données. 7. Le système affiche un message d'erreur indiquant qu'une erreur s'est produite lors de la suppression du sujet.

3.1.2 La modélisation conceptuelle

3.1.2.1 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 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.

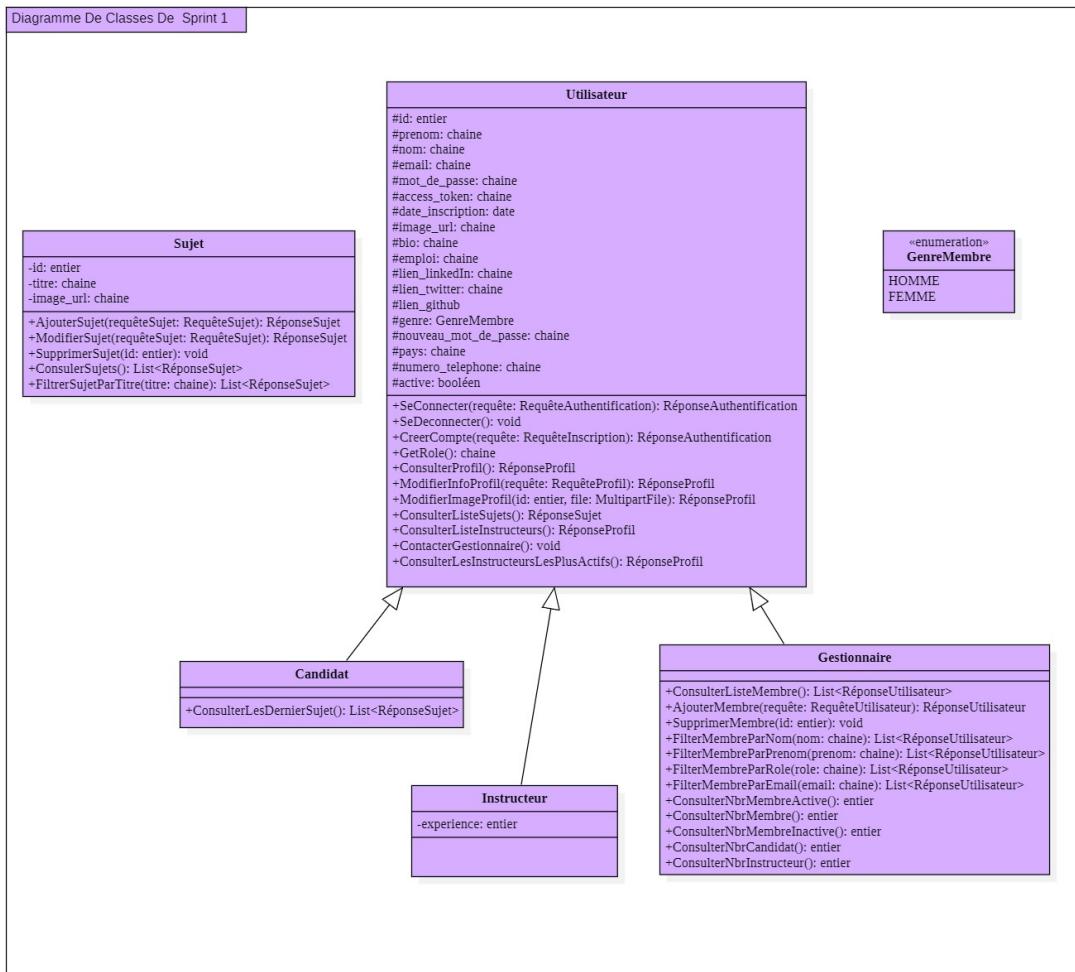


FIGURE 3.2 – Diagramme de classes du premier sprint

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

3.1.2.2 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".

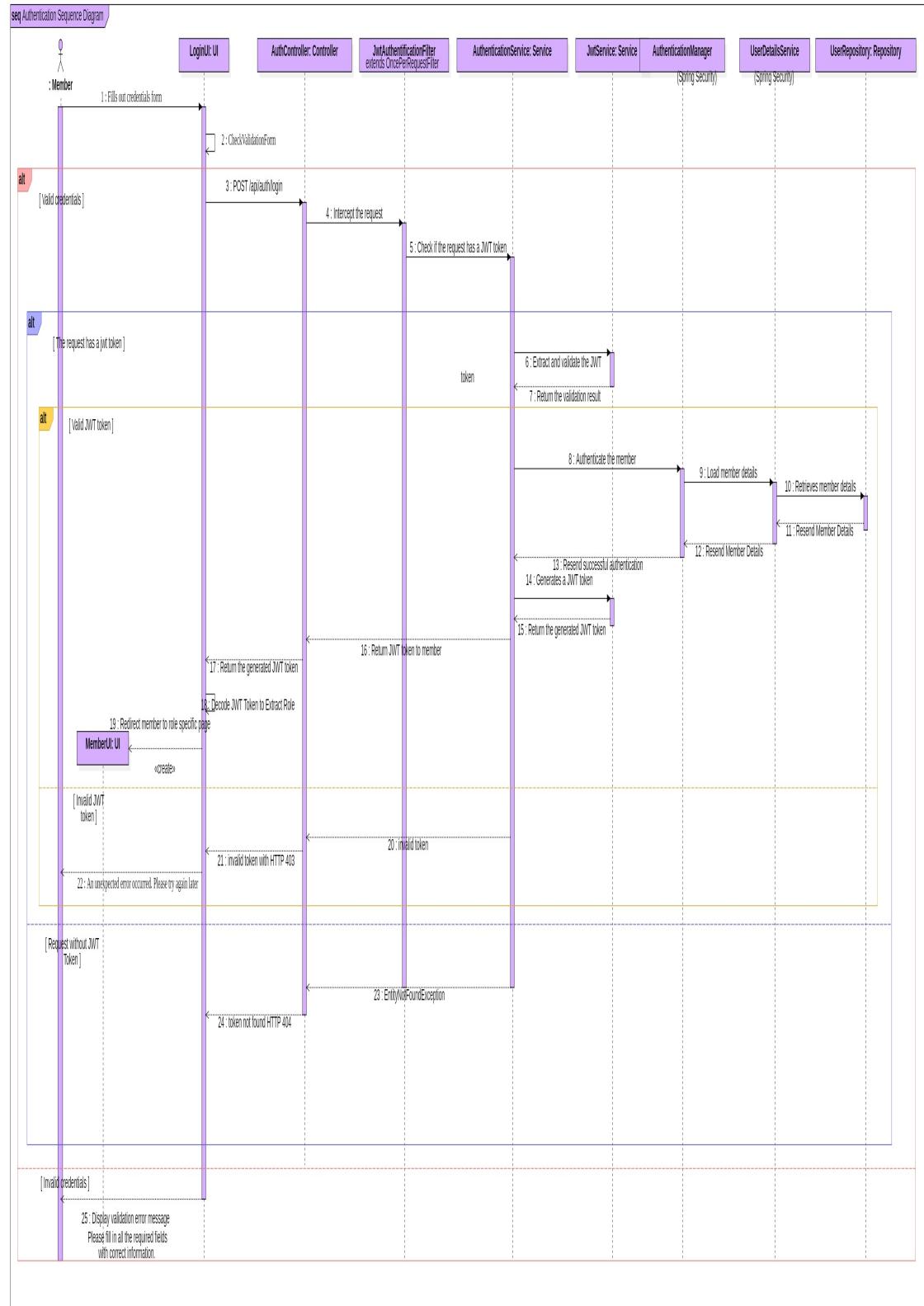


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation « S'authentifier »

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 »

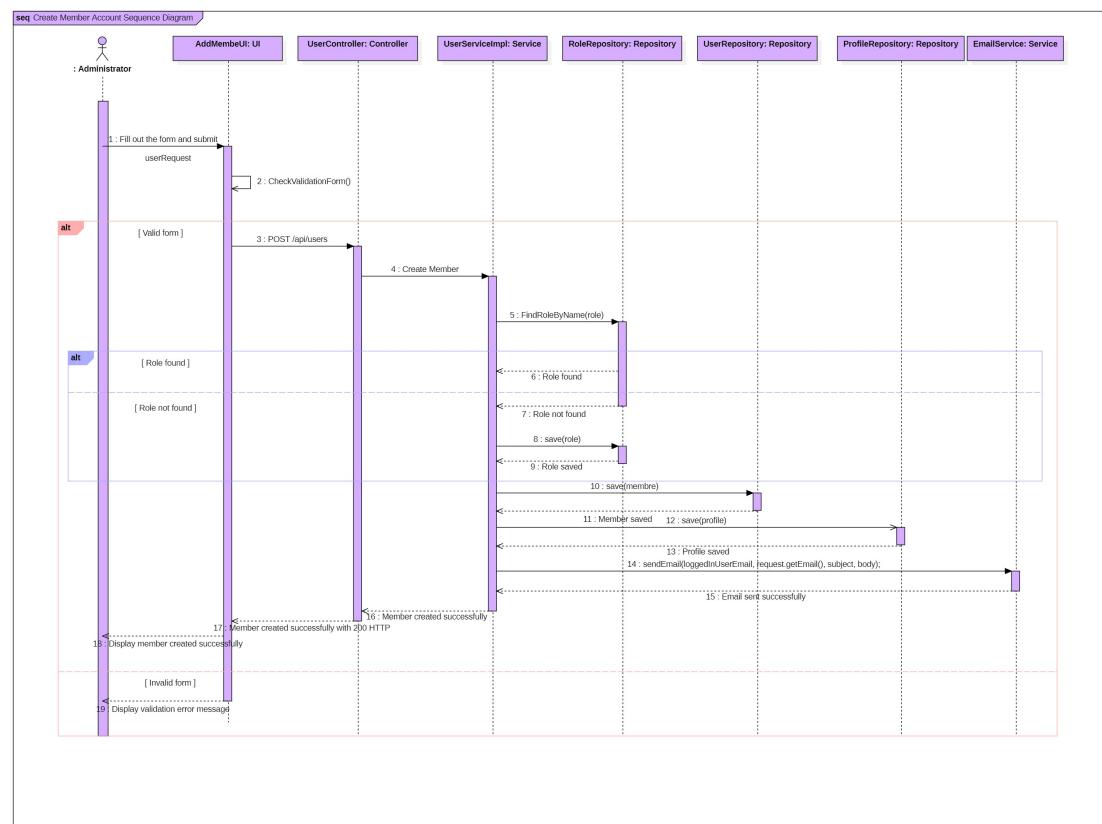


FIGURE 3.4 – Diagramme de séquence du cas d’utilisation « Ajouter un membre »

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 consultés du compte par le membre.

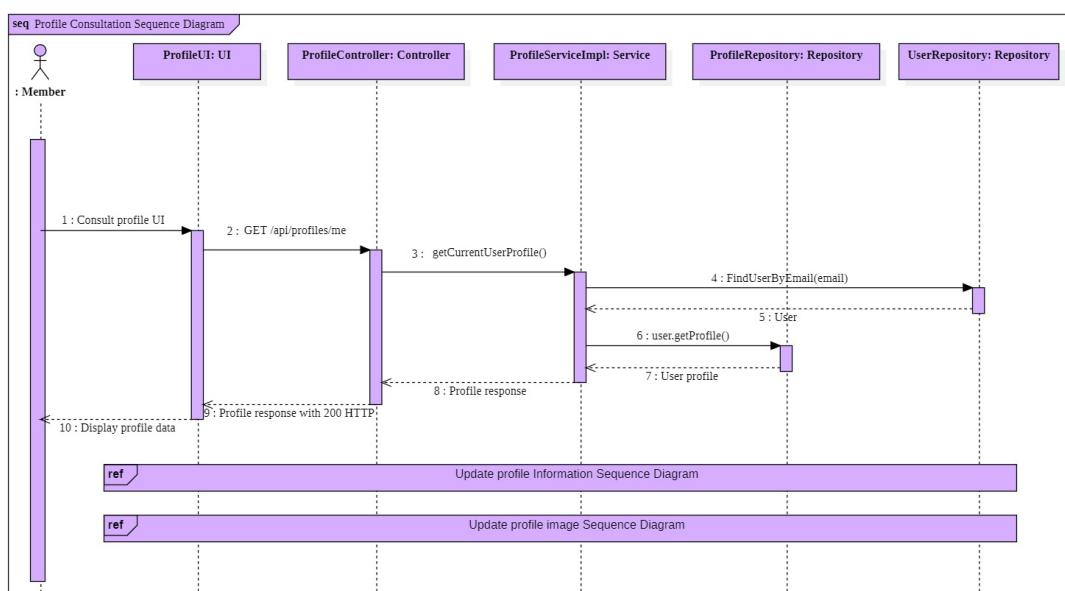


FIGURE 3.5 – Diagramme de séquence du cas d'utilisation « Consulter profil »

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 »**

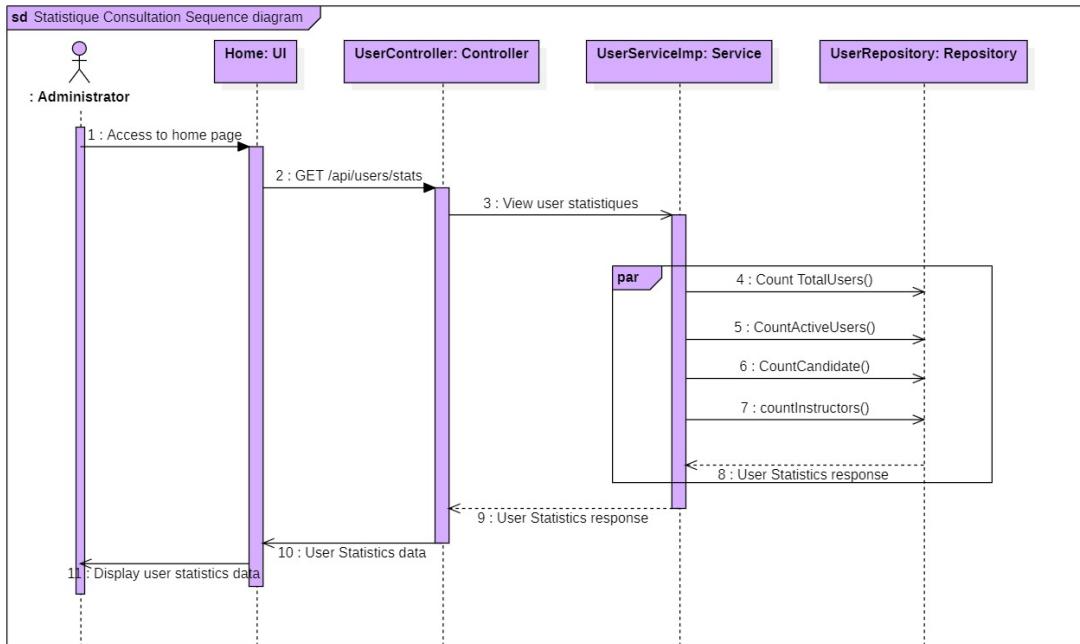


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation « Consulter statistiques »

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.

3.1.3 Réalisation

Après avoir terminé la phase de conception, nous entamons la réalisation de notre premier sprint. Les captures d'écran suivantes présentent les différentes interfaces réalisées pendant ce premier sprint, illustrant ainsi certains éléments clés de notre plateforme Educare.

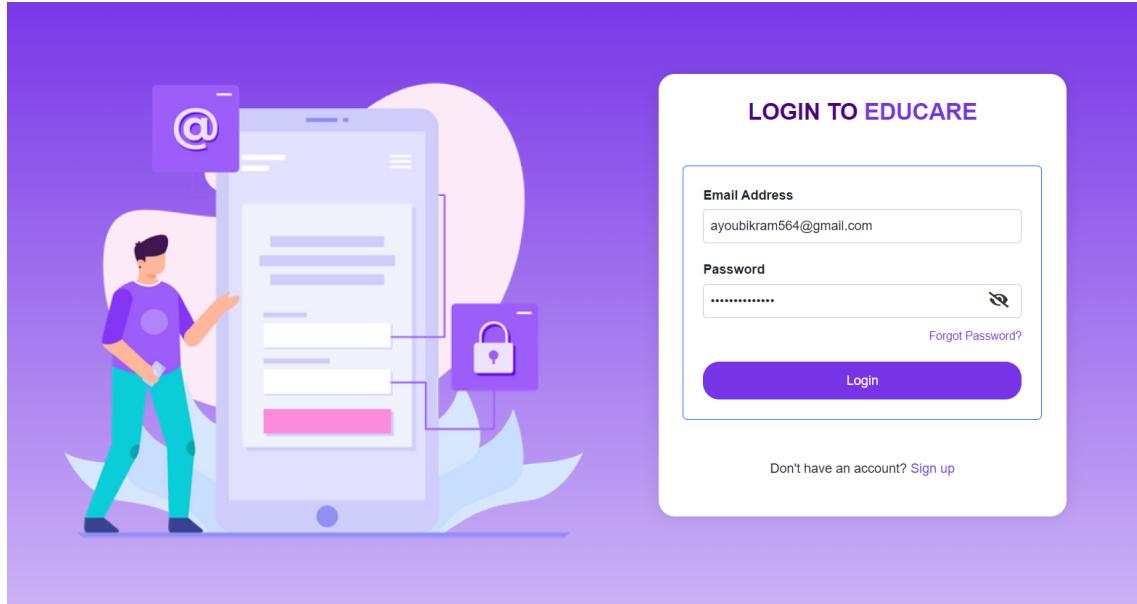


FIGURE 3.7 – Interface d'authentification

L'interface d'authentification, représentée ci-dessus, doit être consultée par tous les utilisateurs de notre application pour y accéder.

ID	Username	Email	Role	Status	User Since	Last Login Date	Actions
1	Syrine Ayeb	syrineayeb@gmail.com	Admin	●	June 6, 2024	June 6, 2024, 4:39 PM	<button>Delete</button>
2	Issaa Malloug	imalloug@gmail.com	Instructor	●	January 2, 2024	June 2, 2024, 10:00 AM	<button>Delete</button>
3	Nahla Sassi	nahla.oughattas@gmail.com	Instructor	●	March 4, 2024	June 4, 2024, 10:00 AM	<button>Delete</button>
4	Ikram Ayoub	ayoubikram564@gmail.com	Admin	●	March 21, 2024	June 21, 2024, 10:00 AM	<button>Delete</button>

FIGURE 3.8 – Interface de gestion des membres

L'interface de gestion des membres, comme illustrée dans la figure ci-dessus, permet de consulter, supprimer, filtrer et ajouter des utilisateurs de l'application. Elle offre une vue d'ensemble des membres existants et des options pour gérer efficacement leurs informations.

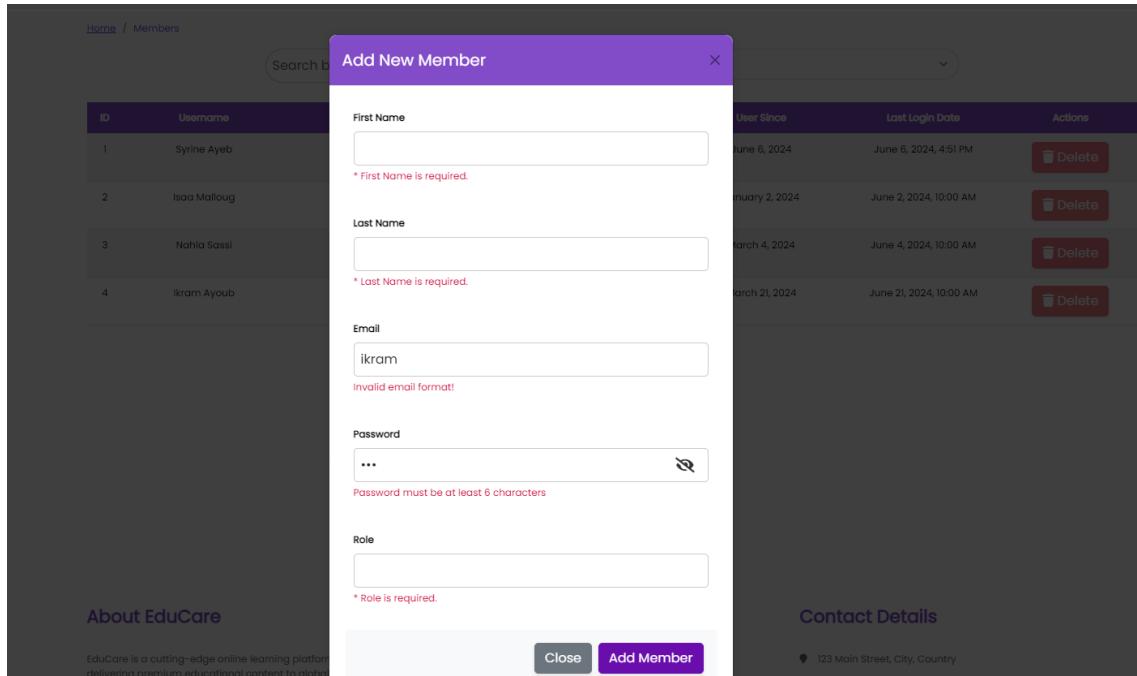


FIGURE 3.9 – Interface de création d'un nouveau membre

Cette interface permet l'ajout d'un nouvel utilisateur à l'application à travers un formulaire.

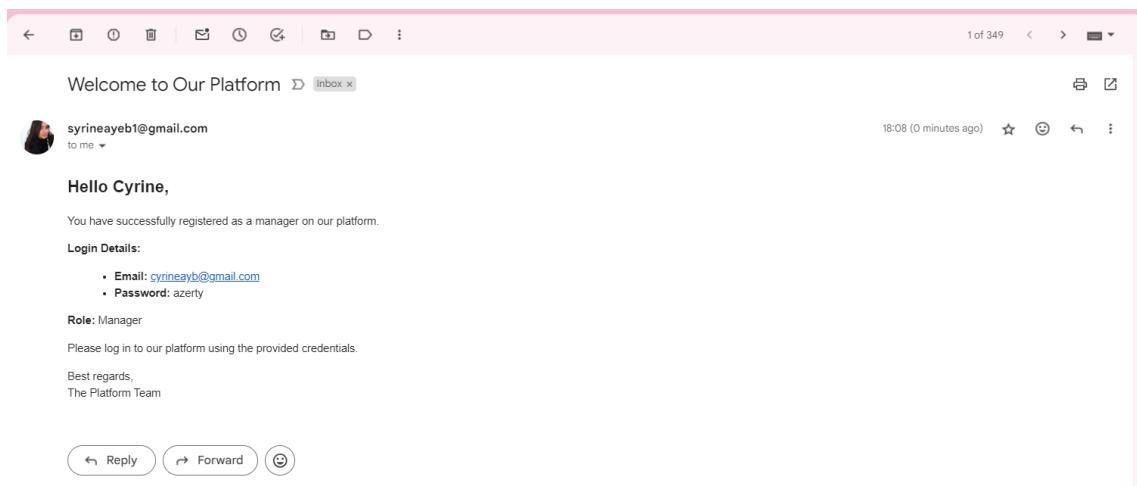


FIGURE 3.10 – Interface de confirmation de nouveau compte

L'interface, située ci-dessus, représente le message de confirmation automatiquement envoyé par email au nouveau membre lors de la création de son compte. Ce message contient les détails de son inscription, tels que son email et son mot de passe.

La prochaine

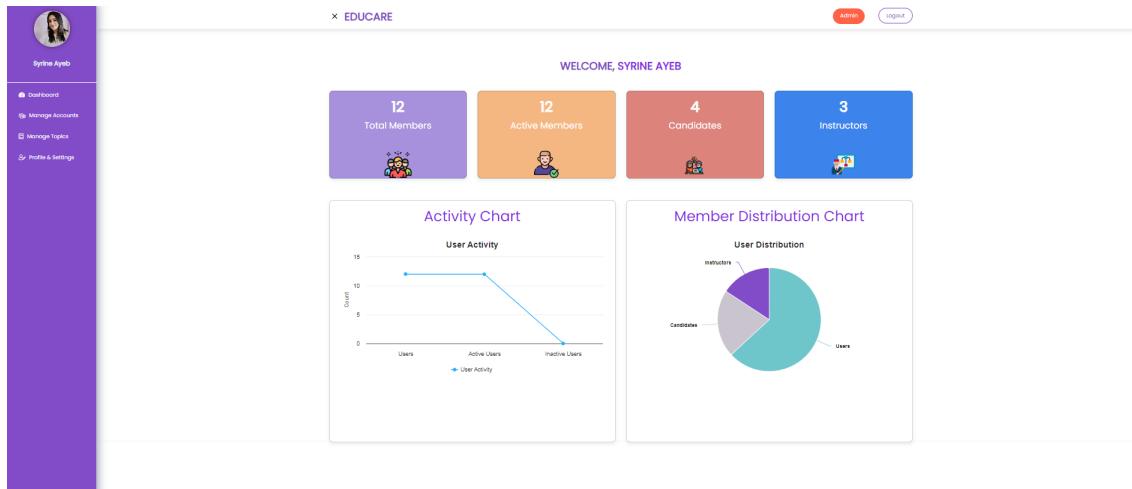


FIGURE 3.11 – Interface du tableau de bord du gestionnaire

Cette interface présente diverses statistiques sur les membres, notamment le nombre total de membres, le nombre de membres actifs et le nombre de membres inactifs. Elle offre une vue d'ensemble claire et concise de l'état actuel de la base d'utilisateurs de l'application.

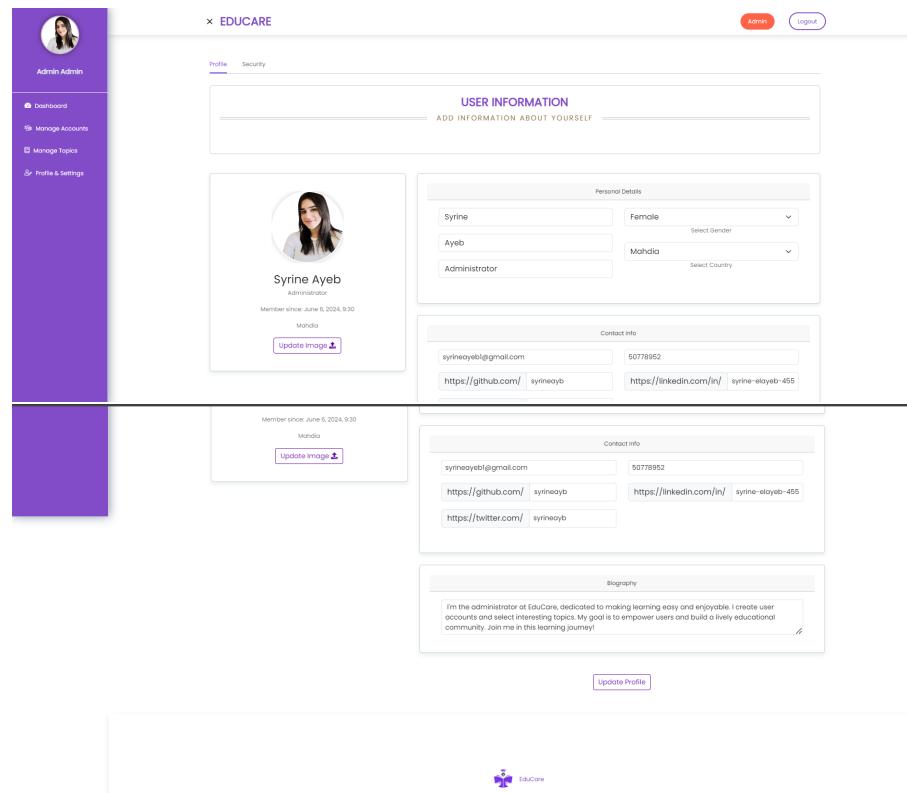


FIGURE 3.12 – Interface de gestion du profil du gestionnaire

Cette interface permet de consulter et de modifier les différentes informations du gestionnaire authentifié. Elle offre une vue détaillée de son profil, facilitant ainsi la

mise à jour et la gestion des données personnelles.

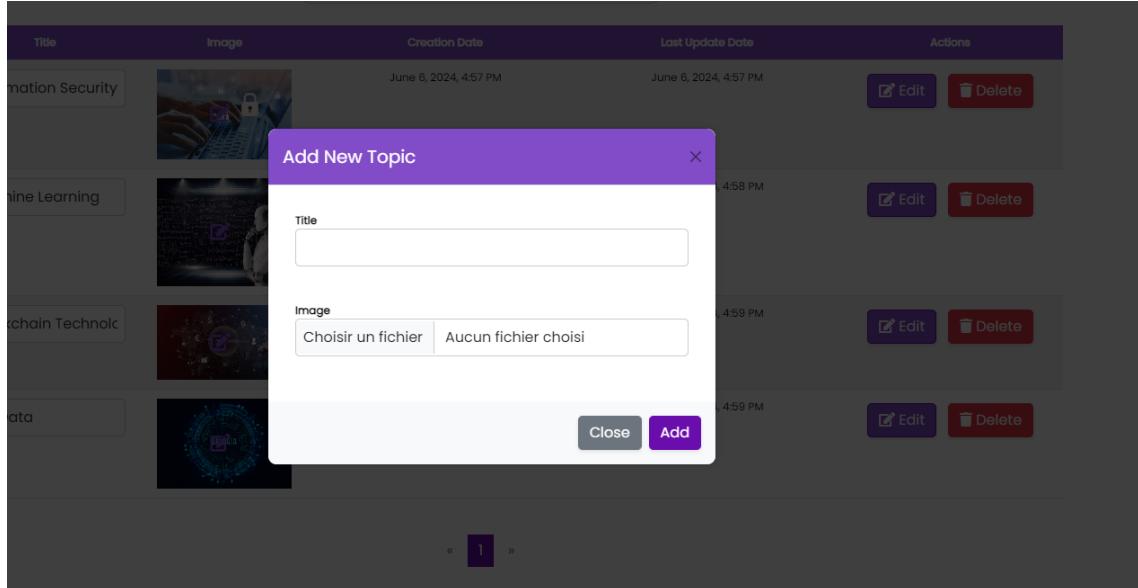


FIGURE 3.13 – Interface d'ajout d'un nouveau sujet

Cette interface offre la possibilité d'ajouter un nouveau sujet à l'application via un formulaire convivial.

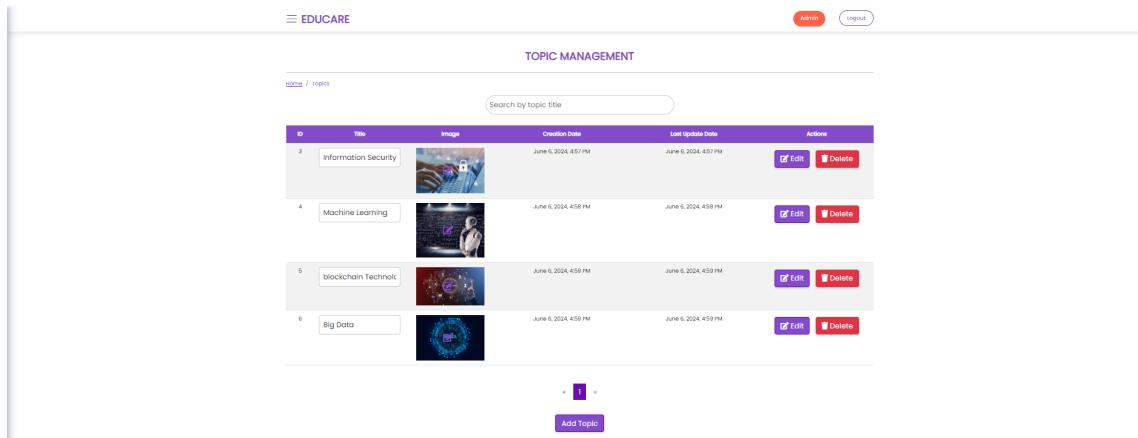


FIGURE 3.14 – Interface de gestion des sujets

Cette interface permet de consulter, modifier et supprimer les sujets de l'application. Elle offre une vue d'ensemble des sujets existants avec des options pour mettre à jour leurs informations ou les retirer de la liste.

3.1.4 Tests

Dans le cadre de notre processus de développement, nous accordons une importance capitale aux tests pour assurer la qualité et la fiabilité de notre application. Nous

avons adopté une approche complète en intégrant à la fois des tests unitaires et des tests avec Postman pour valider le comportement de nos API HTTP.

3.1.4.1 Tests unitaires

Les tests unitaires jouent un rôle essentiel en garantissant le bon fonctionnement de chaque composant logiciel dès les premières étapes du développement. Nous avons ainsi développé des tests unitaires pour nos services et nos contrôleurs. Les tests des services nous permettent de vérifier la logique métier et les opérations sur les données, assurant ainsi la robustesse de notre application. De même, les tests des contrôleurs s'assurent que les requêtes HTTP sont correctement gérées et que les réponses aux clients sont conformes aux spécifications.

— Test unitaire de service

```
if (true) {
    @Test
    void createTopicSuccessfullyCreatesTopic() {
        // Given
        TopicRequest request = new TopicRequest("Web development", null);
        Topic topic = new Topic("Web development");
        Topic savedTopic = new Topic("Web development");
        savedTopic.setId(1L);

        // Mock the calls
        when(topicMapper.toTopic(request)).thenReturn(topic);
        when(topicMapper.save(topic)).thenReturn(savedTopic);
        when(topicMapper.toTopic(savedTopic)).thenAnswer(new TopicResponse()
            .id(1L)
            .title("Web development")
            .topicId(null) // Set LocalDateTime.now() to null as we're not testing this field here
            .localDateTime(LocalDateTime.now())
            .localDateTime(LocalDateTime.now())
        );
        doNothing().when(validation).validate(request);

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

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

        // Verify interactions
        verify(validation).validate(request);
        verify(topicMapper).toTopic(request);
        verify(topicMapper).save(savedTopic);
        verify(topicMapper).toTopic(response.savedTopic);
    }
}
```

FIGURE 3.15 – Test unitaire du service : création de sujet

Ce test unitaire valide la méthode `createTopic` de `TopicServiceImpl` en simulant la création d'un nouveau sujet avec le titre "Web development". 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.

— Test unitaire de contrôleur

```
if (true) {
    @Test
    public void testLatestTopicsForOneElement() throws Exception {
        int peek = 1;
        int size = 1;
        List<TopicResponse> expectedTopicResponses = new ArrayList();
        expectedTopicResponses.add(new TopicResponse());
        expectedTopicResponses.add(new TopicResponse());
        expectedTopicResponses.add(new TopicResponse());
        expectedTopicResponses.add(new TopicResponse());
        content.assertEquals(expectedTopicResponses, topicService.latestTopics(1));
        assertEquals(1, topicService.latestTopics(1).size());
        assertEquals(1, topicService.latestTopics(1).get(0).getId());
        assertEquals("Web development", topicService.latestTopics(1).get(0).getTitle());
        assertEquals("Web development", topicService.latestTopics(1).get(0).getTopicId());
        assertEquals("2023-09-12T10:00:00", topicService.latestTopics(1).get(0).getLocalDateTime());
        assertEquals("2023-09-12T10:00:00", topicService.latestTopics(1).get(0).getLocalDateTime());
    }

    @Test
    public void testLatestTopicsForTwoElements() throws Exception {
        int peek = 2;
        int size = 2;
        List<TopicResponse> expectedTopicResponses = new ArrayList();
        expectedTopicResponses.add(new TopicResponse());
        expectedTopicResponses.add(new TopicResponse());
        content.assertEquals(expectedTopicResponses, topicService.latestTopics(2));
        assertEquals(2, topicService.latestTopics(2).size());
        assertEquals(1, topicService.latestTopics(2).get(0).getId());
        assertEquals("Web development", topicService.latestTopics(2).get(0).getTitle());
        assertEquals("Web development", topicService.latestTopics(2).get(0).getTopicId());
        assertEquals("2023-09-12T10:00:00", topicService.latestTopics(2).get(0).getLocalDateTime());
        assertEquals("2023-09-12T10:00:00", topicService.latestTopics(2).get(0).getLocalDateTime());
        assertEquals(2, topicService.latestTopics(2).get(1).getId());
        assertEquals("Java", topicService.latestTopics(2).get(1).getTitle());
        assertEquals("Java", topicService.latestTopics(2).get(1).getTopicId());
        assertEquals("2023-09-12T10:00:00", topicService.latestTopics(2).get(1).getLocalDateTime());
        assertEquals("2023-09-12T10:00:00", topicService.latestTopics(2).get(1).getLocalDateTime());
    }
}
```

FIGURE 3.16 – Test unitaire du contrôleur : récupération des derniers sujets

Ce test unitaire évalue le comportement de l'endpoint `latest-topics` du contrôleur `TopicController`. À l'aide de Mockito, il simule la réponse du service `TopicService` pour une demande de sujets les plus récents. Les données simulées comprennent une liste de deux `TopicResponse`. En simulant une requête GET vers l'endpoint avec des

paramètres de pagination définis, le test vérifie que la réponse renvoyée a un statut HTTP 200 (OK) et que le contenu correspond aux données simulées, incluant le nombre d'éléments dans la liste et le nombre total de pages.

3.1.4.2 Tests avec Postman

En parallèle, nous avons également effectué des tests avec Postman pour tester nos API HTTP dans des conditions proches de celles rencontrées en production. Ces tests nous ont permis de valider le comportement de notre application dans son ensemble, en assurant le bon fonctionnement des interactions entre le frontend et le backend.

The screenshot shows the Postman interface for a POST request to create a topic. The URL is `HTTP E-learning platform / topics management / create topic`. The method is set to `POST`, and the endpoint is `({{api_url}}:{{api_port}}/api/topics`. The `Body` tab is selected, showing a `form-data` structure with two fields: `title` (Text: "Web development") and `imageFile` (File: "web-development.jpg"). The response section shows a 202 Accepted status with 188 ms and 631 B. The JSON response body is displayed as:

```

1 {
2   "id": 1,
3   "title": "Web development",
4   "imageFile": "http://localhost:8080/api/topics/downloadTopicImage/web-development.jpg",
5   "createdAt": "2024-06-07T23:30:13.709302",
6   "updatedAt": "2024-06-07T23:30:13.709302"
7 }

```

FIGURE 3.17 – Test Postman : création d'un sujet

Après avoir effectué le test de création d'un sujet à l'aide de l'API "/api/topics" avec les données de formulaire contenant le titre et le fichier image, ainsi que le token d'authentification du gestionnaire, nous avons reçu une réponse avec le statut 202 Accepted, indiquant que la demande a été acceptée pour traitement. La réponse contient les détails du sujet créé, y compris son identifiant, son titre, le lien vers l'image et les dates et heures de création et de mise à jour.

Cette approche combinée de tests unitaires et de tests avec Postman nous offre une couverture complète, assurant que notre application fonctionne de manière fiable, respecte les normes établies et offre une expérience utilisateur optimale.

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**

Deuxième Sprint : Gestion Pédagogique et Confidentialité

Plan

Introduction	78
4.1 Le deuxième sprint	78
4.1.1 Spécifications fonctionnelles	93
4.1.1.1 Raffinement de cas d'utilisation du sprint 2	93
4.1.1.2 Diagramme de cas d'utilisation détaillé du Sprint 2.....	93
4.1.1.3 Description textuelle de cas d'utilisation	95
4.1.1.4 La modélisation conceptuelle.....	100
4.1.1.4.1 Diagramme de classes	100
4.1.1.4.2 Diagramme de séquence détaillé.....	100
4.1.1.5 Réalisation	107
4.1.1.6 Tests	113
4.1.1.6.1 Tests unitaires	113
4.1.1.6.2 Tests avec Postman	115
Conclusion	115

Introduction

Ce chapitre présente le travail effectué durant le deuxième sprint, couvrant les étapes d'analyse, de conception et de réalisation. Pendant ce sprint, nous nous concentrerons sur la gestion du contenu pédagogique et la confidentialité des données. L'analyse définira clairement les besoins pour gérer le contenu pédagogique et protéger les données des utilisateurs. La conception traduira ces besoins en solutions techniques, qui seront ensuite implémentées et testées pour assurer leur bon fonctionnement et leur intégration avec le système existant.

4.1 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. (Tests avec Postman) - 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) 	1 jour

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. (Tests avec Postman) - 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) 	1 jour
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. (Tests avec Postman) - 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) 	1 jour
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) 	2 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. (Tests avec Postman) - Afficher la liste des cours récupérés depuis l'API backend. (Frontend) 	2 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. (Tests avec Postman) - Créer un formulaire de modification de cours avec les détails pré-remplis. (Frontend) 	1 jour
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. (Tests avec Postman) - Afficher un message de confirmation après la suppression réussie. (Frontend) 	1 jour
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. (Tests avec Postman) - Afficher la liste des cours récupérés depuis l'API backend. (Frontend) 	1 jour

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. (Tests avec Postman) - Afficher une confirmation de l'inscription réussie dans l'interface utilisateur du candidat. (Frontend) 	1 jour
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. (Tests avec Postman) - Afficher la liste des candidats inscrits dans l'interface utilisateur de l'instructeur pour chaque cours. (Frontend) 	1 jour
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. (Tests avec Postman) - Afficher la liste des cours récupérés depuis l'API backend. (Frontend) 	1 jour

12	En tant qu'instructeur ou candidat, je veux filtrer les cours par titre et par nom instructeur	<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) 	1 jour
13	En tant que candidat, je veux voir le nombre de candidats inscrit dans un cours	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour consulter le nombre de candidats inscrit dans un cours (Backend) - Gérer les erreurs liées à la consultation et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Afficher une interface utilisateur permettant la consultation du nombre de candidats inscrit dans un cours. (Frontend) 	1 jour
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. (Tests avec Postman) - Créer une interface utilisateur permettant à l'instructeur de saisir les détails de la leçon et de l'assigner à un cours spécifique. (Frontend) 	2 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. (Tests avec Postman) - Afficher la liste des leçons récupérées depuis l'API backend dans l'interface utilisateur pour l'instructeur. (Frontend) 	1 jour
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. (Tests avec Postman) - Créer une interface utilisateur permettant à l'instructeur de modifier les détails de la leçon sélectionnée. (Frontend) 	1 jour
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) 	1 jour

18	En tant qu'instructeur, je veux ajouter des ressources associées à une leçon de mes cours (image, vidéo, PDF)	<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. (Tests avec Postman) - Créer une interface utilisateur permettant à l'instructeur de télécharger et d'ajouter des ressources à une leçon sélectionnée. (Frontend) 	2 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. (Tests avec Postman) - Afficher la liste des ressources récupérées depuis l'API backend dans l'interface utilisateur pour l'instructeur. (Frontend) 	2 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) 	1 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. (Tests avec Postman) - 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) 	1 jour
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. (Tests avec Postman) - 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) 	2 jours
23	En tant qu'instructeur, je veux lister les quiz assignés à mon cours	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer les quiz assignés 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. (Tests avec Postman) - Dans l'interface utilisateur, afficher les détails du quiz assigné à chaque cours dans la liste des cours de l'instructeur. (Frontend) 	1 jour

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. (Tests avec Postman) - 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) 	1 jour
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. (Tests avec Postman) - 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) 	1 jour
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) 	2 jours

26	En tant qu'instructeur, je veux ajouter des questions et leurs réponses à un quiz	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour ajouter des questions et leurs réponses à un quiz spécifique. (Backend) - Gérer les erreurs liées à l'ajout des questions et leurs réponses et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Dans l'interface utilisateur, fournir un formulaire permettant à l'instructeur de saisir les détails des questions et leurs réponses et de les assigner à un quiz spécifique. (Frontend) 	2 jours
27	En tant qu'instructeur, je veux lister les questions et les réponses assignées à un quiz	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer les questions et les réponses assignées à un quiz spécifique. (Backend) - Gérer les erreurs liées à la récupération des questions et des réponses et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Dans l'interface utilisateur, afficher les détails des questions et leurs réponses assignées à chaque quiz dans la liste des quiz de l'instructeur. (Frontend) 	2 jours

28	En tant qu'instructeur, je veux modifier les détails d'une question et ces réponses assignées à un quiz	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour la modification des détails d'une question et ces réponses assignées à un quiz spécifique. (Backend) - Gérer les erreurs liées à la modification des questions et ces réponses et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Dans l'interface utilisateur, fournir une option de modification pour les détails des questions et ces réponses, permettant à l'instructeur de mettre à jour les informations des questions et des réponses assignées à un quiz. (Frontend) 	1 jour
29	En tant qu'instructeur, je veux supprimer une question et ces réponses assignées à un quiz	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour la suppression d'une question et ces réponses assignées à un quiz spécifique. (Backend) - Gérer les erreurs liées à la suppression des questions et ces réponses et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Dans l'interface utilisateur, fournir une option de suppression pour chaque question et ces réponses, permettant à l'instructeur de supprimer une question et ces réponses assignées à un quiz. (Frontend) 	1 jour

30	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) 	2 jours
31	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. (Tests avec Postman) - Dans l'interface utilisateur, afficher le résultat du quiz du candidat après qu'il l'a terminé. (Frontend) 	1 jours
32	En tant que candidat, je veux consulter les réponses correctes après avoir passé un quiz	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer les réponses correctes d'un quiz après que le candidat l'a terminé. (Backend) - Gérer les erreurs liées à la récupération des réponses correctes et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Dans l'interface utilisateur, afficher les réponses correctes au candidat après qu'il a terminé le quiz. (Frontend) 	1 jours

33	En tant que candidat, je veux suivre ma progression dans mes cours, ressources et leçons	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour récupérer les informations de progression du candidat dans ses cours, ressources et leçons. (Backend) - Gérer les erreurs liées à la récupération des données de progression et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Dans l'interface utilisateur, fournir un tableau de bord affichant la progression du candidat dans ses cours, ressources et leçons. Utiliser des graphiques et des indicateurs de progression pour une visualisation claire. (Frontend) 	1 jours
34	En tant qu'instructeur, je souhaite consulter les statistiques de mes cours et de mes candidats, y compris le nombre de candidats inscrits à chacun de mes cours, le nombre total de candidats, le nombre total de cours que j'ai créés, ainsi que le progrès de mes candidats dans les cours et les quiz.	<p>Créer un endpoint API dédié pour récupérer les statistiques des cours et des candidats de l'instructeur, comprenant le nombre de candidats inscrits à chaque cours, le nombre total de candidats, le nombre total de cours de l'instructeur et les progrès des candidats dans les cours et les quiz. (Backend)</p> <p>Gérer les erreurs liées à la récupération des statistiques et renvoyer des réponses appropriées. (Backend)</p> <p>Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman)</p> <p>Dans l'interface utilisateur, fournir un tableau de bord affichant de manière claire et compréhensible les statistiques des cours et des candidats de l'instructeur. Utiliser des graphiques, des tableaux et d'autres visualisations appropriées pour rendre les données facilement accessibles. (Frontend)</p>	2 jours

35	En tant qu'instructeur, je veux voir la progression de mes candidats dans mes cours et leurs quiz	- Créer un endpoint API dédié pour récupérer les informations de progression des candidats dans les cours et les quiz de l'instructeur. (Backend) - Gérer les erreurs liées à la récupération des données de progression et renvoyer des réponses appropriées. (Backend) - Tester les endpoints avec Postman pour garantir leur bon fonctionnement. (Tests avec Postman) - Dans l'interface utilisateur, fournir un tableau de bord affichant la progression des candidats dans les cours et les quiz de l'instructeur. Utiliser des graphiques et des indicateurs de progression pour une visualisation claire. (Frontend)	1 jours
----	---	--	---------

36	En tant que candidate, je veux marquer les ressources comme terminées	<ul style="list-style-type: none"> - Créer un endpoint API dédié pour permettre aux candidates de marquer les resources comme terminées. (Backend) - Gérer les requêtes de marquage des ressources terminées côté serveur et renvoyer des réponses appropriées. (Backend) - Tester le bon fonctionnement de l'endpoint API pour garantir la fiabilité de la fonctionnalité. (Tests avec Postman) - Implémenter une fonctionnalité dans le code Angular permettant aux candidates de marquer les ressources comme étant terminées. (Frontend) - Gérer les interactions utilisateur pour permettre le marquage des ressources comme terminées. (Frontend) - Tester la fonctionnalité de marquage pour garantir son bon fonctionnement et une expérience utilisateur fluide. (Frontend) 	1 jours
37	En tant que candidat, je veux télécharger les ressources assignées à mes cours	<ul style="list-style-type: none"> - Implémenter une fonction de téléchargement dans le code Angular pour permettre aux candidats de télécharger les ressources assignées à leurs cours. (Frontend) - Gérer les erreurs liées au téléchargement des ressources et afficher des messages appropriés à l'utilisateur en cas de problème. (Frontend) - Tester la fonctionnalité de téléchargement pour garantir son bon fonctionnement et une expérience utilisateur fluide. (Frontend) 	2 jours

38	<p>En tant qu'instructeur, je souhaite créer un quiz et sélectionner ses questions à partir d'une liste de questions déjà existantes dans le même cours.</p>	<ul style="list-style-type: none"> - Mettre en place un point de terminaison API dédié pour Ajouter les quiz attribués à un cours spécifique et lister toutes les questions existantes dans les anciens quiz qui y sont incluses. (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. (Tests avec Postman) - 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) 	2 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, réalisation et le test.

4.1.1 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.

4.1.1.1 Raffinement de cas d'utilisation du sprint 2

Le diagramme de cas d'utilisation du sprint deux 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
- Gérée confidentialité

4.1.2 Diagramme de cas d'utilisation détaillé du Sprint 2

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

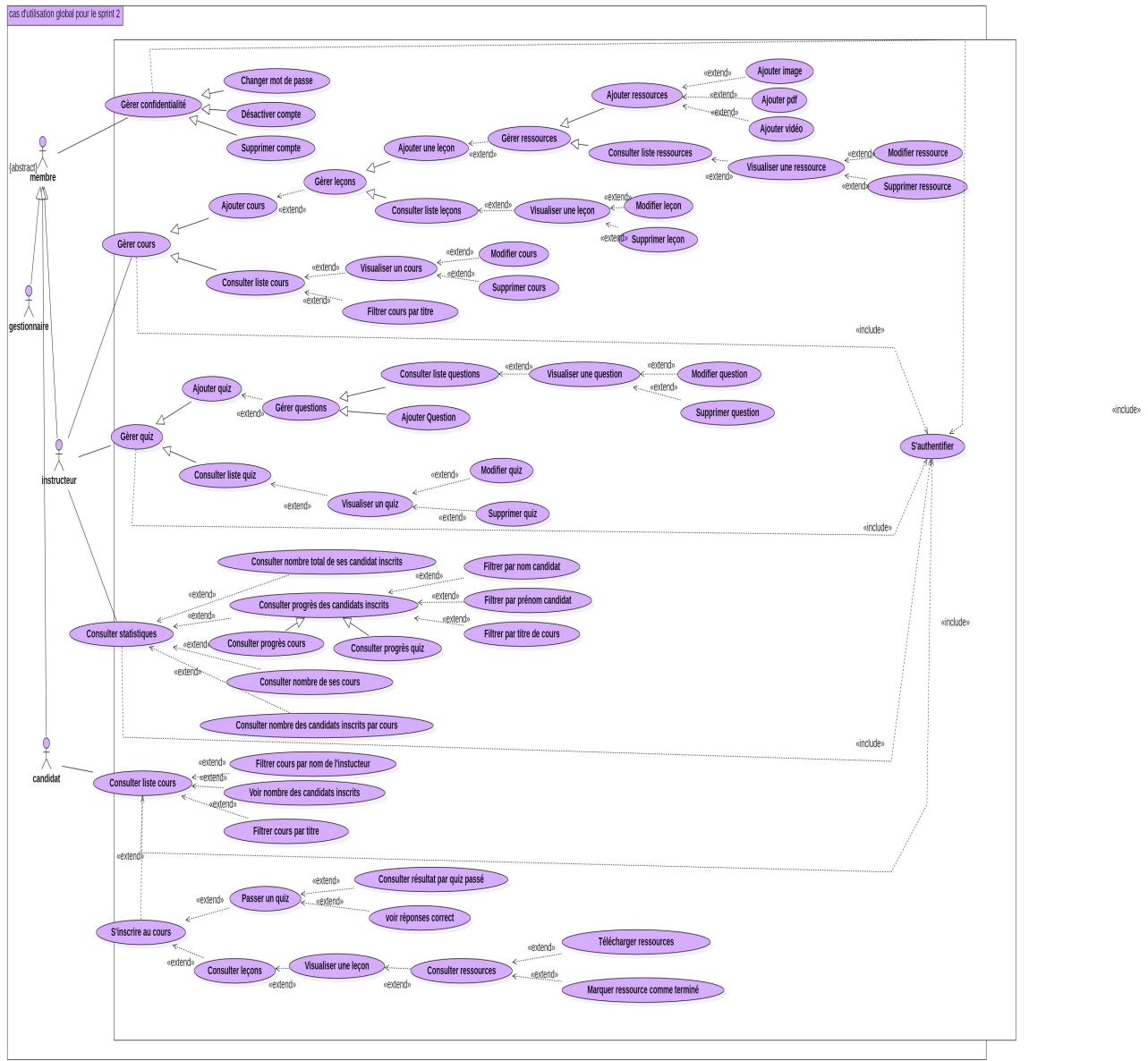


FIGURE 4.1 – Diagramme de cas d'utilisation détaillé du sprint 2

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 et des ressources à ces cours, consulter les statistiques et gérer les quiz. L'authentification est une étape préalable nécessaire avant toute action sur le système.

Les candidats inscrits à un cours peuvent passer les quiz associés une fois qu'ils y sont inscrits. Ils peuvent également consulter les leçons et les ressources fournies dans le cadre de ce cours, ainsi que marquer ces ressources comme terminées pour suivre leur progression. En outre, les candidats ont la possibilité de télécharger les

ressources pour un accès hors ligne ou ultérieur.

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.

4.1.3 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	<p>SA1 : Mot de passe actuel invalide.</p> <p>SA1 commence après l'étape 6 du scénario nominal.</p> <p>7. Le système détecte que le mot de passe actuel saisi par le membre est incorrect.</p> <p>8. Le système affiche un message d'erreur indiquant que le mot de passe actuel est invalide.</p>

	<p>9. Le membre est invité à saisir à nouveau son mot de passe actuel.</p> <p>SA2 : Le mot de passe et sa confirmation ne sont pas identiques.</p> <p>SA2. commence après l'étape 4 du scénario nominal.</p> <p>5. Le système détecte que le nouveau mot de passe saisi par le membre et sa confirmation ne correspondent pas.</p> <p>6. 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>7. Le membre est invité à saisir à nouveau le nouveau mot de passe et sa confirmation.</p> <p>SA3 : Champ vide.</p> <p>SA3. commence après l'étape 3 du scénario nominal.</p> <p>4. 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>5. Le système affiche un message d'erreur indiquant que tous les champs doivent être remplis.</p> <p>6. Le membre est invité à saisir les informations manquantes.</p>
--	---

Le tableau ci-dessous explique le processus d'ajout d'un cours par un instructeur.

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	<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.

	<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	<p>SA1 : Champs obligatoires manquants. SA1 commence après l'étape 6 du scénario nominal.</p> <p>7. Le système détecte que certains champs obligatoires du formulaire ne sont pas remplis.</p> <p>8. Le système affiche un message d'erreur indiquant que tous les champs obligatoires doivent être remplis.</p> <p>9. L'instructeur est invité à remplir les champs manquants.</p> <p>SA2 : Erreur lors de l'ajout du cours. SA2 commence après l'étape 7 du scénario nominal.</p> <p>8. Le système rencontre une erreur lors de l'ajout du cours à la base de données.</p> <p>9. Le système affiche un message d'erreur indiquant que l'ajout du cours a échoué.</p> <p>10. L'instructeur est invité à réessayer ultérieurement ou à contacter le support technique.</p>

Le tableau ci-dessous explique le processus de la modification d'une leçon par un instructeur

TABLE 4.4 – 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	leçon est modifiée avec succès.
Scénario Nominal	<p>1. L'instructeur accède à la page de gestion des cours.</p> <p>2. L'instructeur sélectionne le cours auquel la leçon à modifier est associée.</p> <p>3. Le système affiche une liste des leçons disponibles dans le cours sélectionné dans un tableau modifiable.</p> <p>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.</p> <p>5. Une fois les modifications effectuées, l'instructeur valide le formulaire.</p>

	<p>6. Le système enregistre les modifications de la leçon dans la base de données.</p> <p>7. Le système affiche un message de confirmation indiquant que la leçon a été modifiée avec succès.</p>
Scénario alternatif	<p>SA1 : Erreur lors de la modification de la leçon.</p> <p>SA1 commence après l'étape 7 du scénario nominal.</p> <p>8. Le système rencontre une erreur lors de la modification de la leçon dans la base de données.</p> <p>9. Le système affiche un message d'erreur indiquant que la modification de la leçon a échoué.</p> <p>10. L'instructeur est invité à réessayer ultérieurement.</p>

Le tableau ci-dessous explique le processus de passage d'un quiz par un candidat.

TABLE 4.5 – Description textuelle du cas d'utilisation « Passer un quiz »

Cas d'utilisation	Passer un quiz
Acteur principal	Candidats
Précondition	<p>Le candidat est authentifié sur la plateforme.</p> <p>Le candidat est inscrit à un cours comportant un quiz.</p>
Postcondition	Affichage des résultats attribués à ce quiz passé.
Scénario Nominal	<p>1. Le candidat accède à la liste des cours auxquels il est inscrit.</p> <p>2. Le candidat sélectionne le cours contenant le quiz qu'il souhaite passer.</p> <p>3. Le système affiche les détails du cours sélectionné, y compris les quiz disponibles.</p> <p>4. Le candidat identifie le quiz qu'il souhaite passer et sélectionne l'option "Passer le quiz".</p> <p>5. Le système affiche le quiz avec les questions et les options de réponses.</p> <p>6. Le candidat répond à chaque question en sélectionnant l'option appropriée.</p> <p>7. Une fois toutes les questions répondues, le candidat soumet ses réponses.</p> <p>8. Le système enregistre les réponses du candidat et évalue son score.</p> <p>9. Le système affiche les résultats du quiz, y compris le score obtenu par le candidat.</p> <p>10. Le candidat peut consulter ses réponses et voir les corrections si disponibles.</p>

Le tableau ci-dessous explique le processus d'ajout d'un quiz avec sélection de questions par un instructeur.

TABLE 4.6 – Description textuelle du cas d'utilisation « Ajouter un quiz avec sélection de questions »

Cas d'utilisation	Ajouter un quiz avec sélection de questions
Acteur principal	Instructeur
Précondition	Le cours auxquels le quiz sera ajouté existe déjà.
Postcondition	Le quiz est ajouté au cours sélectionné.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'instructeur accède à la section de gestion des cours. 2. L'instructeur sélectionne le cours auquel il souhaite ajouter un quiz. 3. Le système affiche les détails du cours sélectionné, y compris les options de gestion des quiz. 4. L'instructeur sélectionne l'option "Ajouter un quiz". 5. Le système affiche un formulaire permettant à l'instructeur de saisir les détails du quiz, tels que le titre, la durée et la note. 6. L'instructeur remplit les informations requises pour le quiz. 7. L'instructeur sélectionne les questions à inclure dans le quiz en ajoutant de nouvelles questions avec des options ou en sélectionnant des questions existantes dans d'autres quiz du même cours. 8. Une fois toutes les informations saisies, l'instructeur valide l'ajout du quiz. 9. Le système enregistre le nouveau quiz dans le cours sélectionné.
Scénario Alternatif	<p>SA1 : Le titre du quiz n'est pas unique</p> <p>SA1 démarre juste après l'étape 8 du scénario nominal, lorsque l'instructeur tente de valider l'ajout du quiz mais que le système détecte un problème avec le titre du quiz.</p> <ol style="list-style-type: none"> 9. Le système affiche un message d'erreur à l'instructeur, l'informant que le titre du quiz doit être unique. 10. L'instructeur est invité à fournir un nouveau titre pour le quiz. 11. L'instructeur modifie le titre du quiz pour le rendre unique. 12. L'instructeur valide à nouveau l'ajout du quiz avec le nouveau titre.

13. Le système enregistre le nouveau quiz dans le cours sélectionné.

4.1.4 La modélisation conceptuelle

4.1.4.1 Diagramme de classes

Suite au diagramme du sprint 1, le diagramme de classes suivant représente les nouvelles classes et entités impliquées dans le processus d'authentification et de gestion de la confidentialité, des cours, des leçons, des ressources, des quiz et des inscriptions pour le deuxième sprint. Ce diagramme permet de visualiser la structure et les relations entre ces classes, ce qui facilite la compréhension de la manière dont les différentes entités du système interagissent dans ce processus.

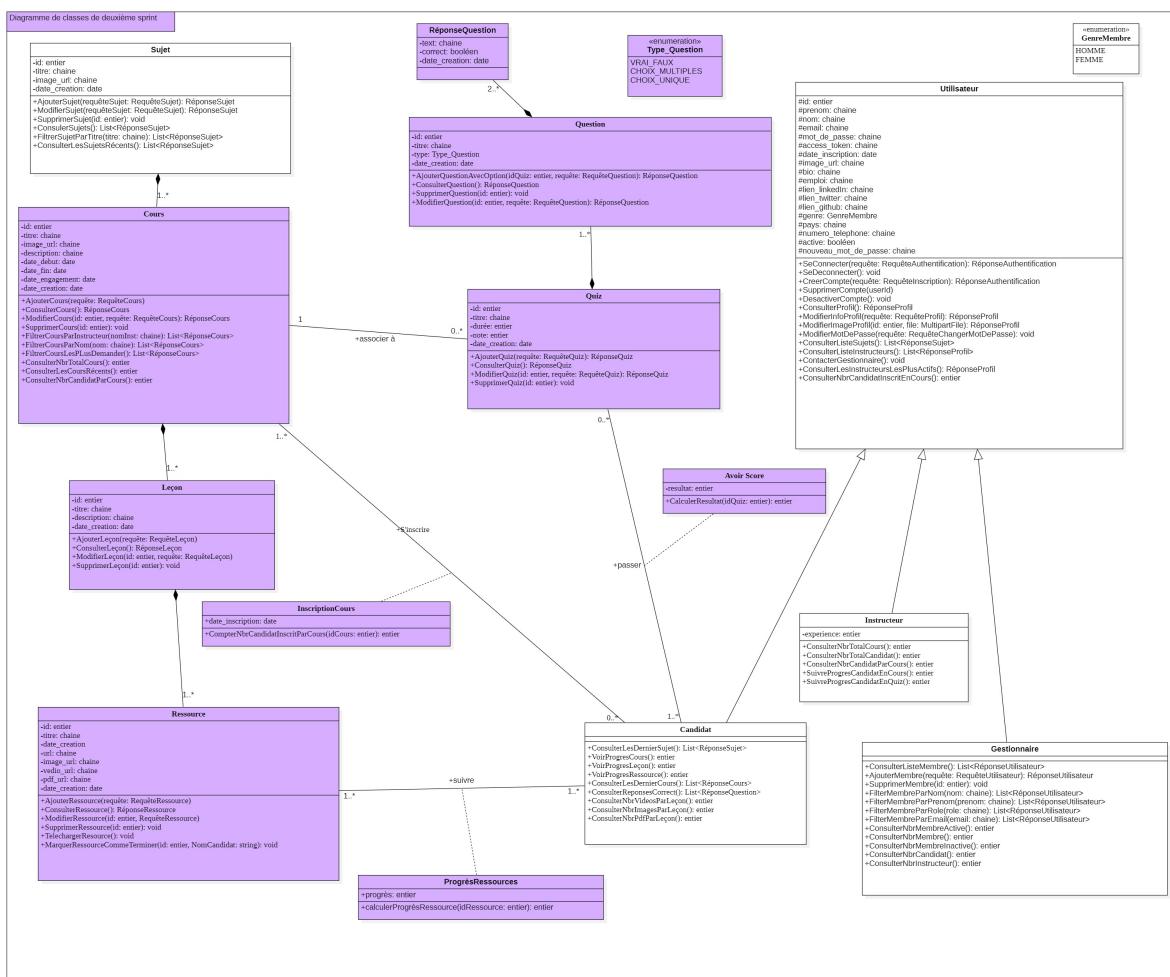


FIGURE 4.2 – Diagramme de classes du second sprint

4.1.4.2 Diagramme de séquence détaillé

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

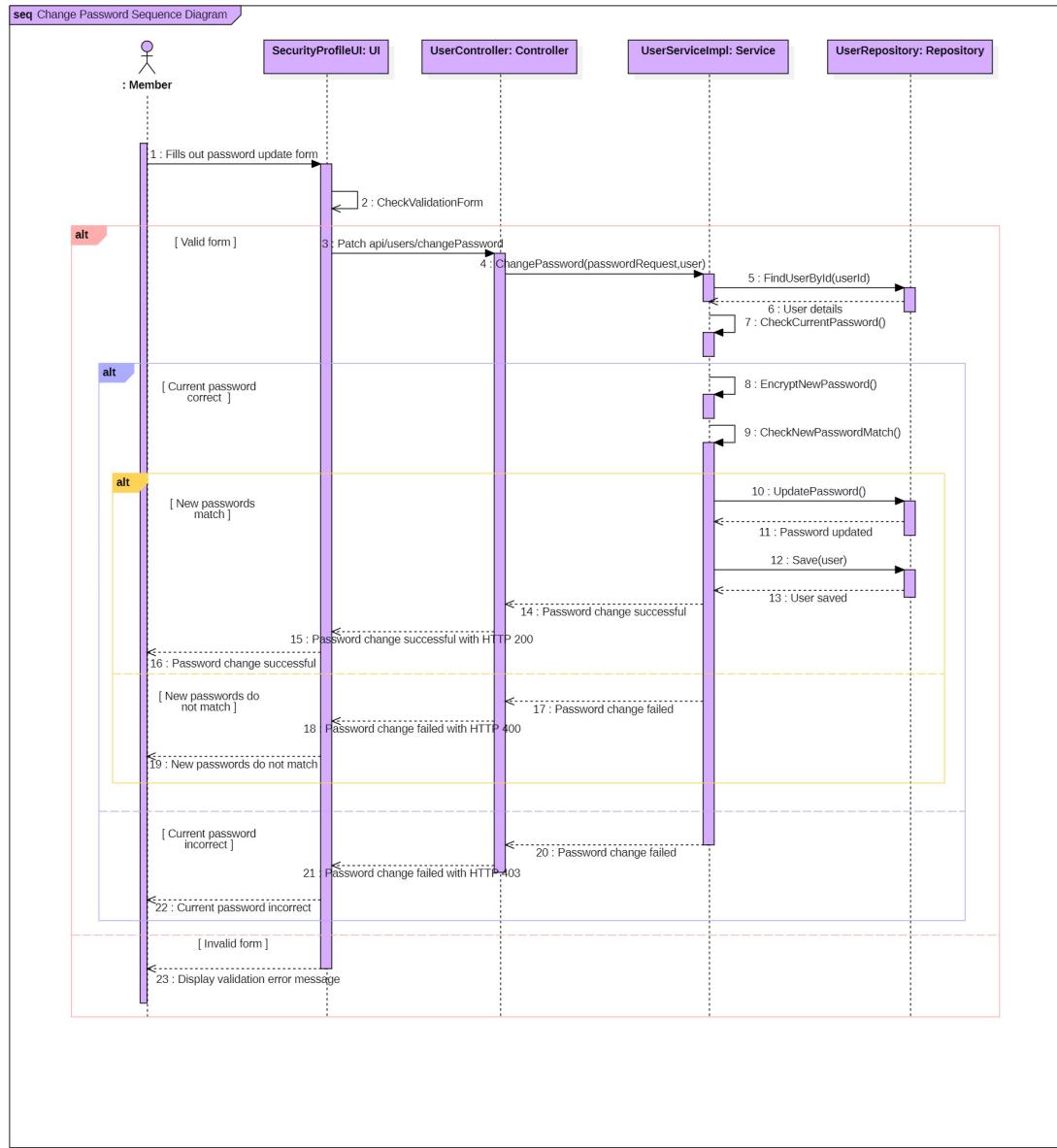


FIGURE 4.3 – Diagramme de séquence du cas d'utilisation « Changer mot de passe »

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 remplaçant 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'utilisateur. 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 »

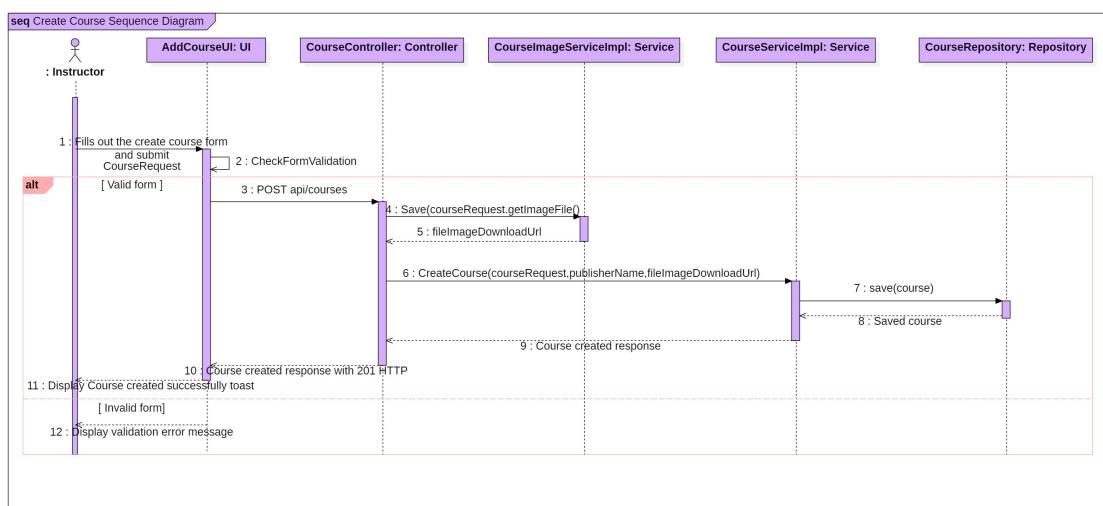


FIGURE 4.4 – Diagramme de séquence du cas d'utilisation « Ajouter un cours »

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 « S'inscrire à Un Cours »**

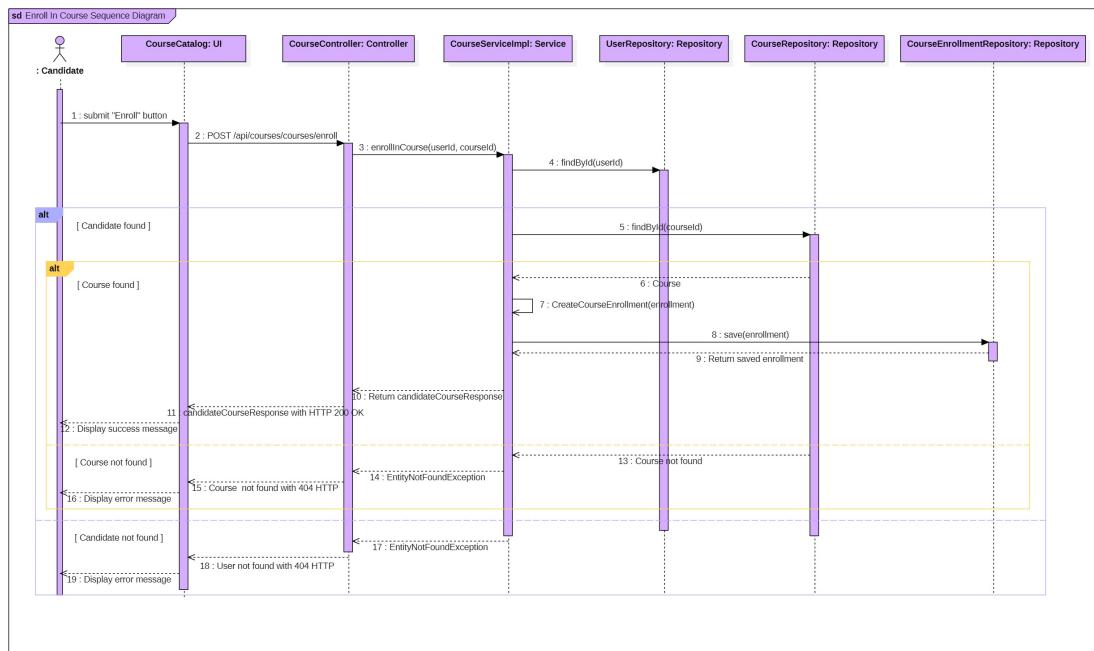


FIGURE 4.5 – Diagramme de séquence du cas d'utilisation « S'inscrire à un cours »

Le diagramme de séquence ci-dessus détaille le processus d'inscription à un cours par un candidat via la méthode `enrollInCourse`. Ce processus s'inscrit dans le cadre du système éducatif et est une étape importante dans le parcours de l'utilisateur pour accéder aux ressources pédagogiques.

Il fait suite à plusieurs actions préalables, notamment la création des cours par les instructeurs. Le processus de création de cours est défini dans le diagramme de séquence "Create Course Sequence Diagram". De plus, l'authentification du candidat, une étape essentielle pour garantir l'accès sécurisé aux fonctionnalités du système, est prise en charge par le système avant l'inscription au cours.

Le processus d'inscription à un cours commence lorsque le candidat interagit avec l'interface utilisateur de la page d'accueil ou du catalogue de cours. En cliquant sur le bouton "Enroll", une requête POST est envoyée au contrôleur de cours, qui déclenche ensuite l'exécution de la méthode `enrollInCourse` du service de cours.

Ce service commence par valider l'existence du candidat et du cours correspondant aux identifiants fournis. Pour garantir une inscription sécurisée, l'authentification du candidat est vérifiée au préalable. En cas de succès, une nouvelle inscription au cours est créée et sauvegardée dans le référentiel des inscriptions. Le candidat est alors ajouté à la liste des candidats inscrits au cours, et une réponse contenant les détails de l'inscription est renvoyée au contrôleur de cours.

Le contrôleur de cours retourne ensuite une réponse HTTP 200 OK à l'interface utilisateur, qui affiche un message de succès au candidat, confirmant ainsi son inscription au cours. En cas d'erreurs, telles que l'absence du candidat ou du cours, des réponses d'erreur appropriées sont renvoyées pour informer le candidat de la situation.

— Diagramme de séquence associé à « Supprimer une leçon »

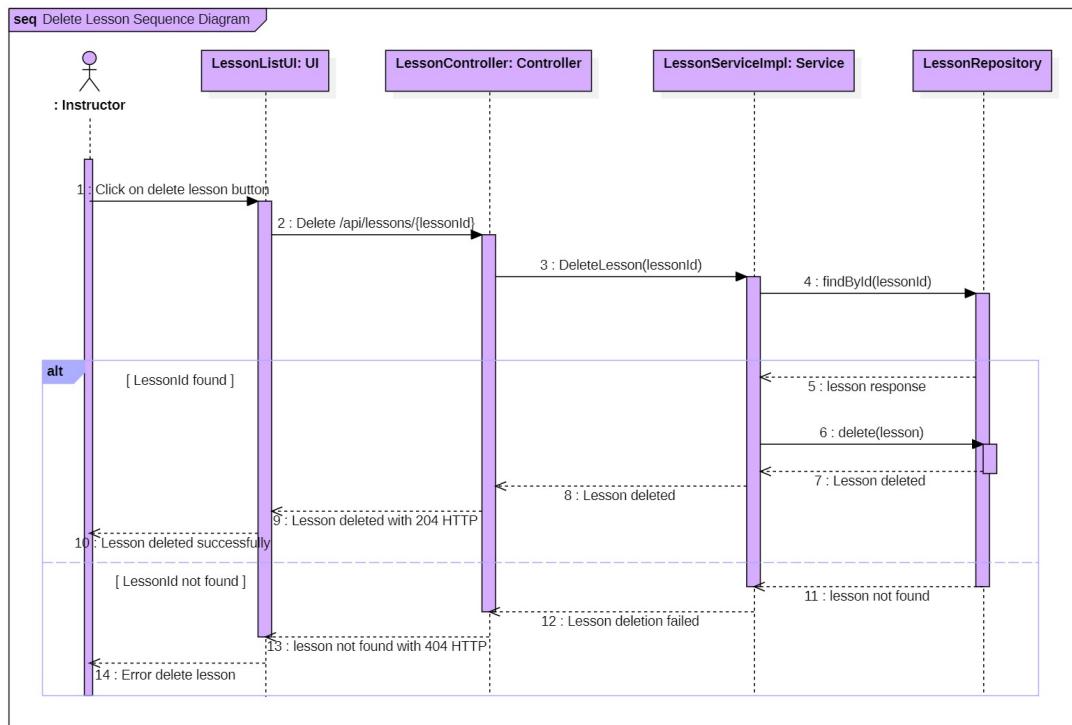


FIGURE 4.6 – Diagramme de séquence du cas d'utilisation « Supprimer une leçon »

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.

— Diagramme de séquence associé à « Crée un quiz »

— Diagramme de séquence associé à « Crée Un Planning D'Une Réunion »

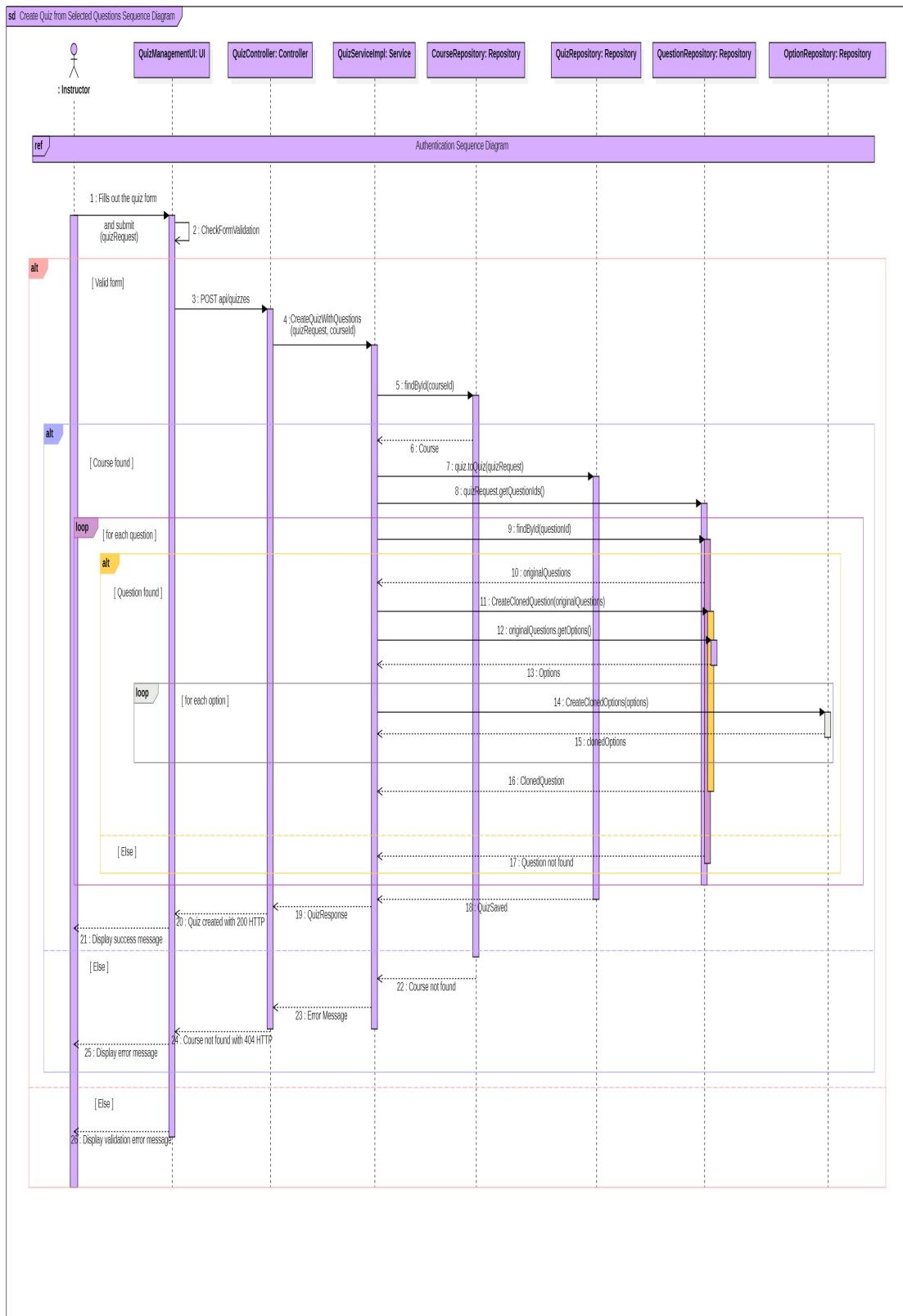


FIGURE 4.7 – Diagramme de séquence du cas d'utilisation « Créer un quiz »

Le diagramme de séquence présent décrit le processus de créer une *****

4.1.5 Réalisation

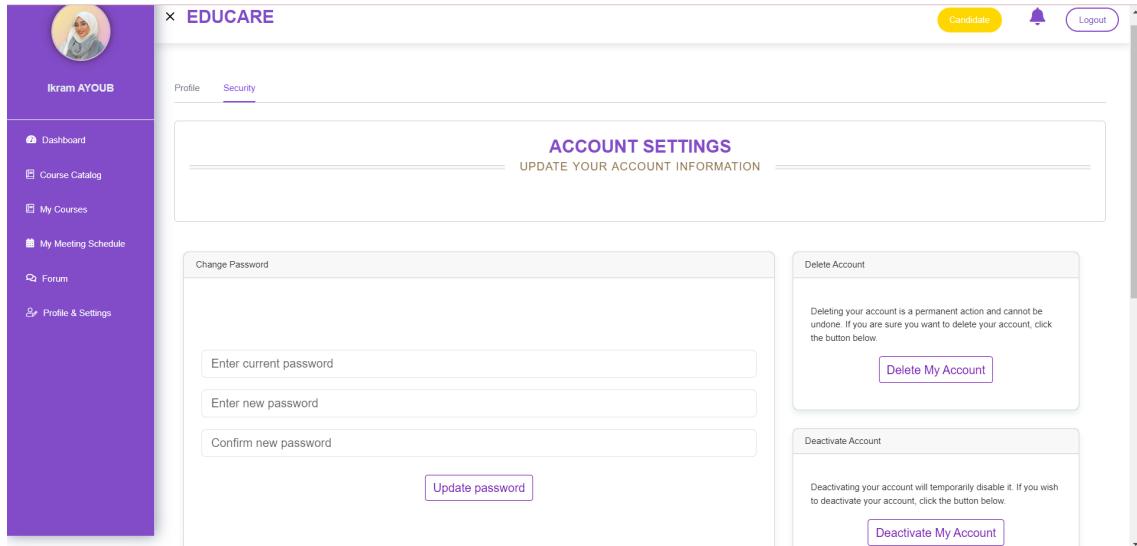


FIGURE 4.8 – Interface de gestion de la confidentialité

Cette interface offre la possibilité de gérer la confidentialité des utilisateurs en leur permettant de modifier leur mot de passe, de désactiver leur compte ou de le supprimer.

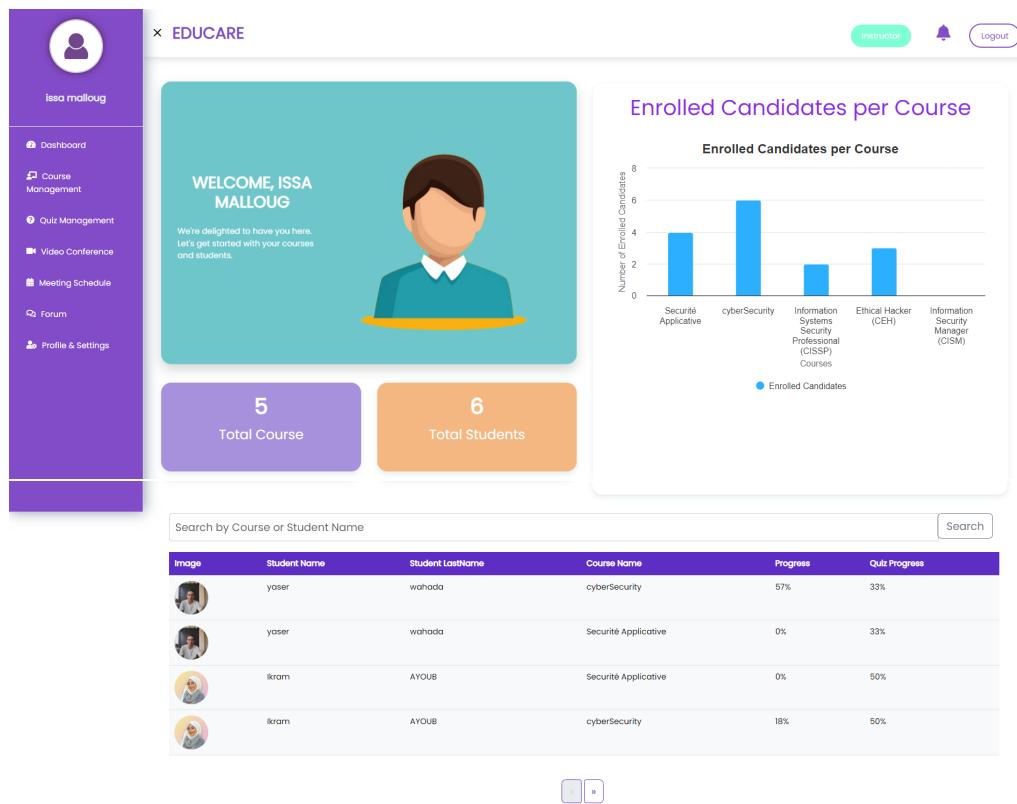


FIGURE 4.9 – Interface tableau de bord instructeur

Cette interface offre la possibilité de consulter les statistiques de mes cours et de mes candidats, ce qui me permet ensuite de suivre leur progression dans le cours ainsi que dans les activités passées.

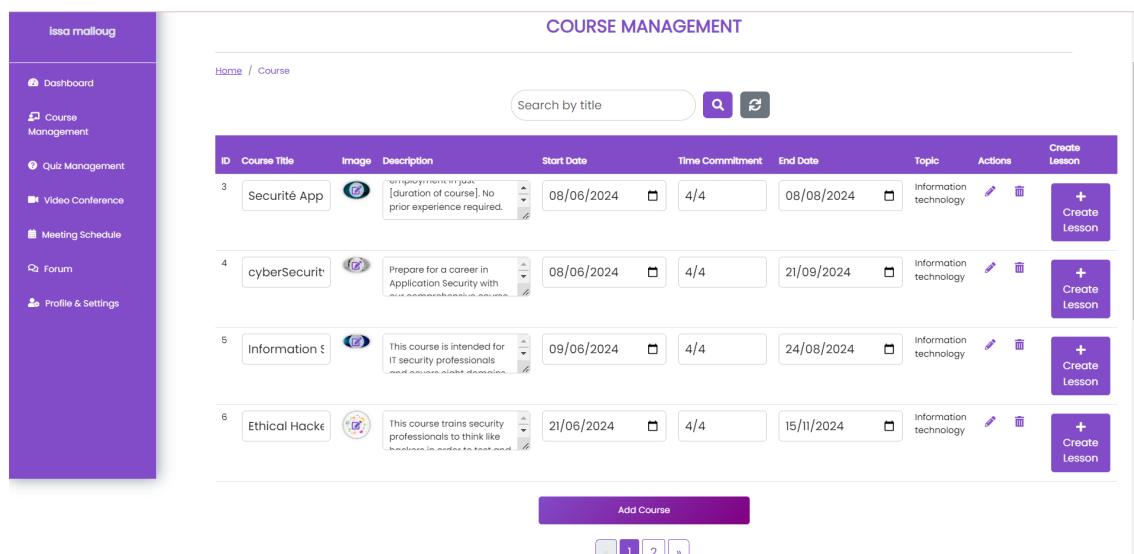


FIGURE 4.10 – Interface de gestion des cours

Cette interface offre la possibilité de gérer les cours en permettant de créer, suppri-

mer, modifier et consulter les cours existants.

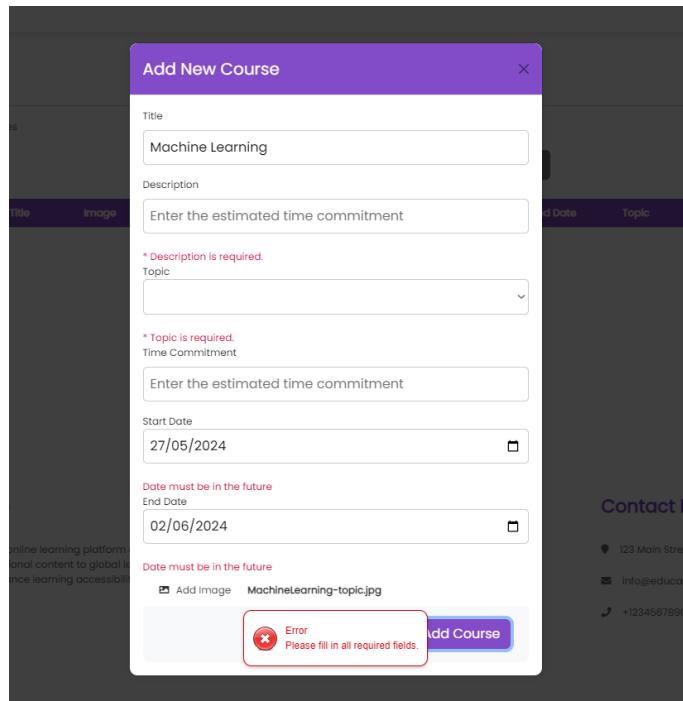


FIGURE 4.11 – Interface d’ajout d’un cours

Cette interface offre la possibilité d’ajouter un cours.

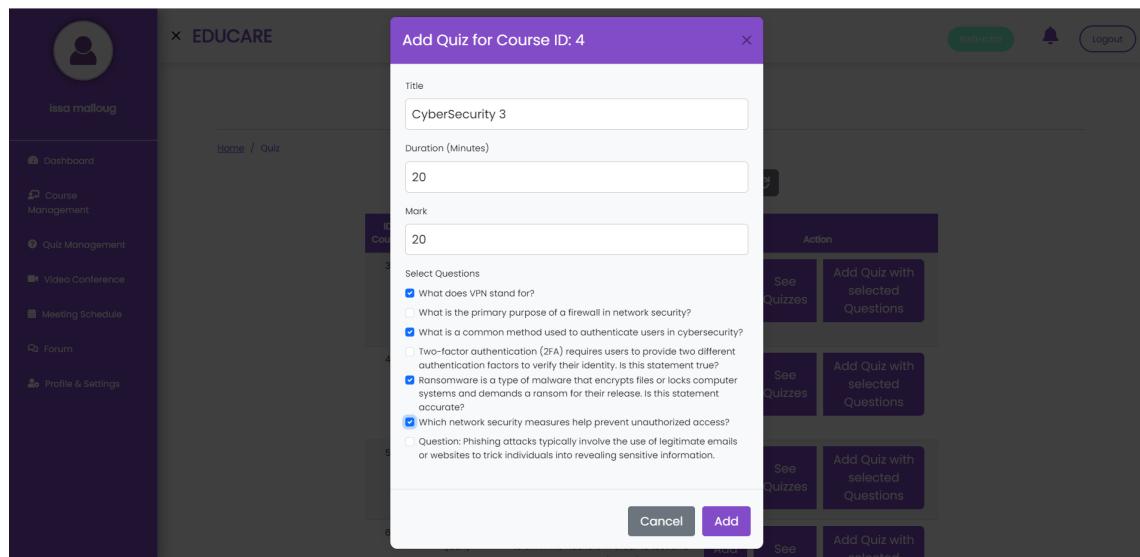


FIGURE 4.12 – Interface d’ajout de quiz avec la sélection des questions déjà existantes

Cette interface offre la possibilité d’ajouter Quiz avec la sélection des questions déjà existant.

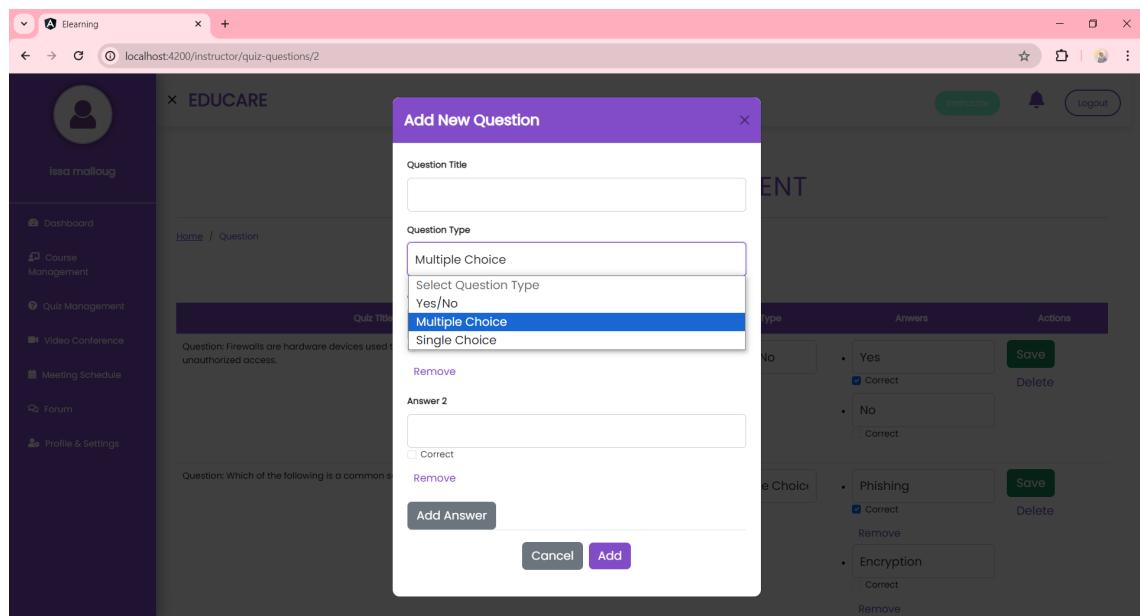


FIGURE 4.13 – Interface d'ajout de question à un quiz

Cette interface offre la possibilité d'ajouter une question à un quiz.

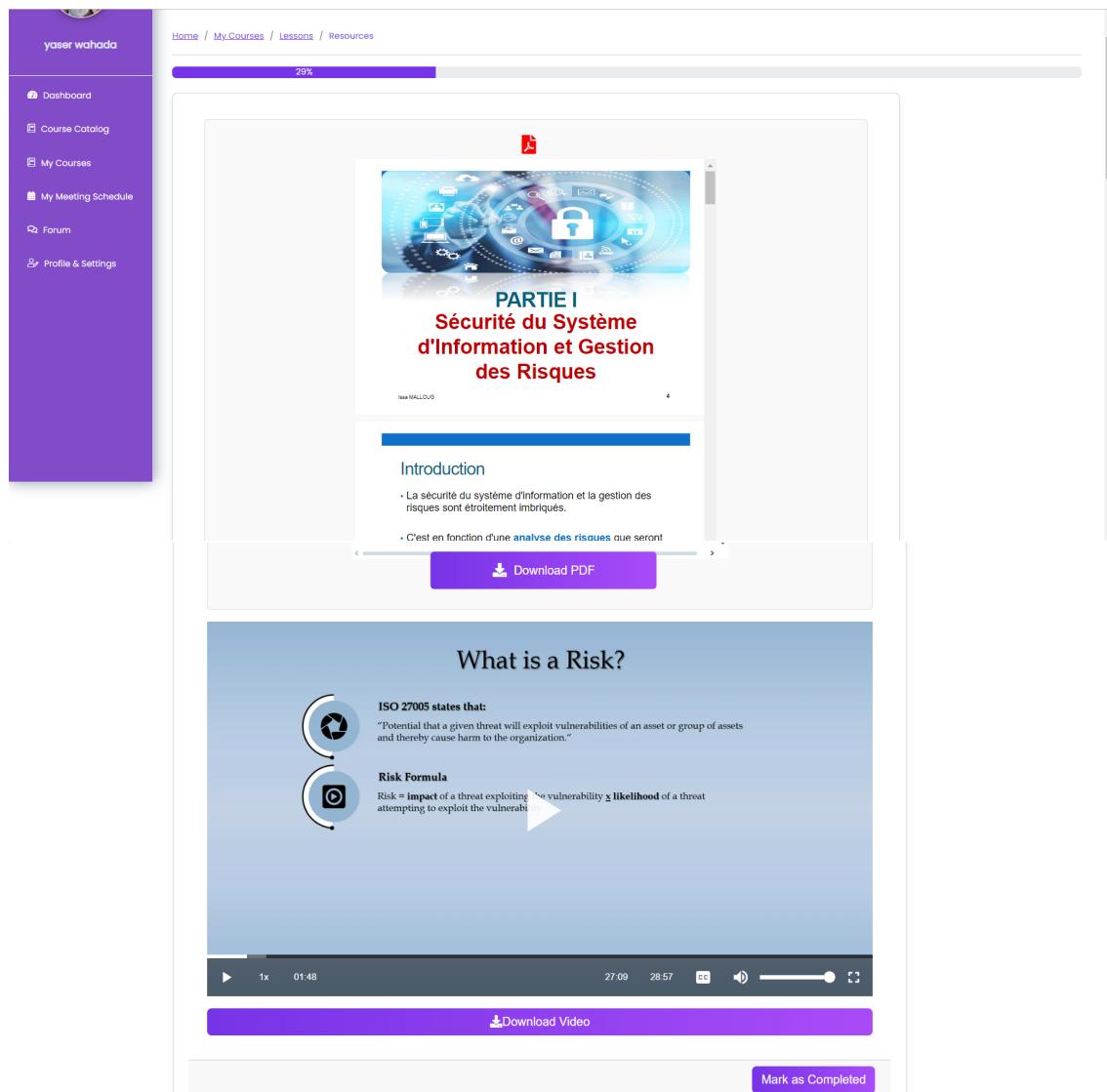


FIGURE 4.14 – Interface de consultation des ressources pour les candidats

Cette interface permet aux candidats de consulter leurs ressources, de les télécharger et de les marquer comme terminées pour suivre leur progression.

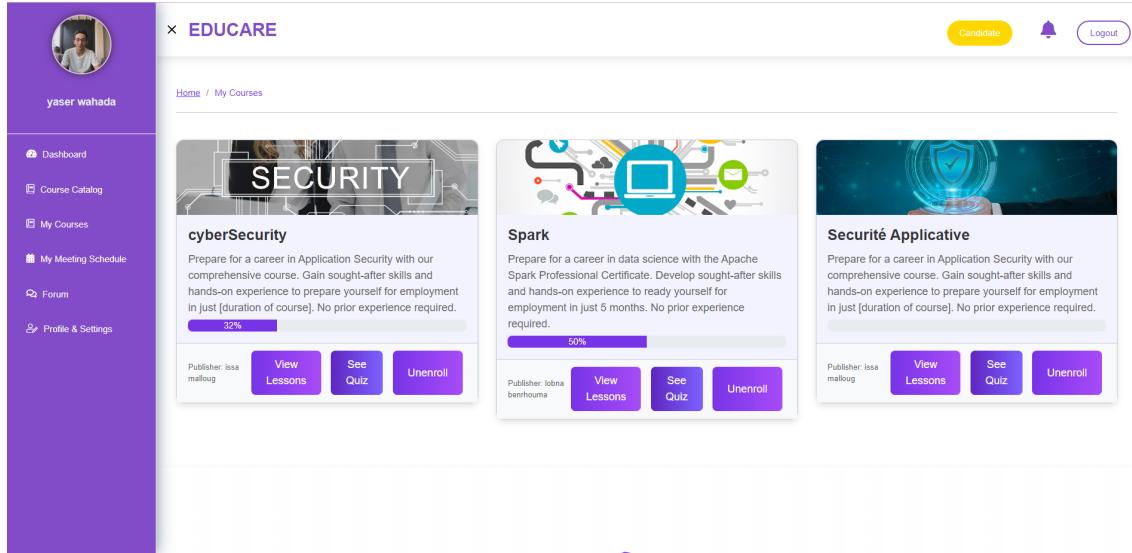


FIGURE 4.15 – Interface de consultation des cours

Cette interface permet aux candidats de consulter leurs cours et de suivre leur état d'avancement. À partir de cette interface, ils peuvent consulter leurs leçons, passer des quiz, et, s'ils le souhaitent, se désinscrire de leur cours.

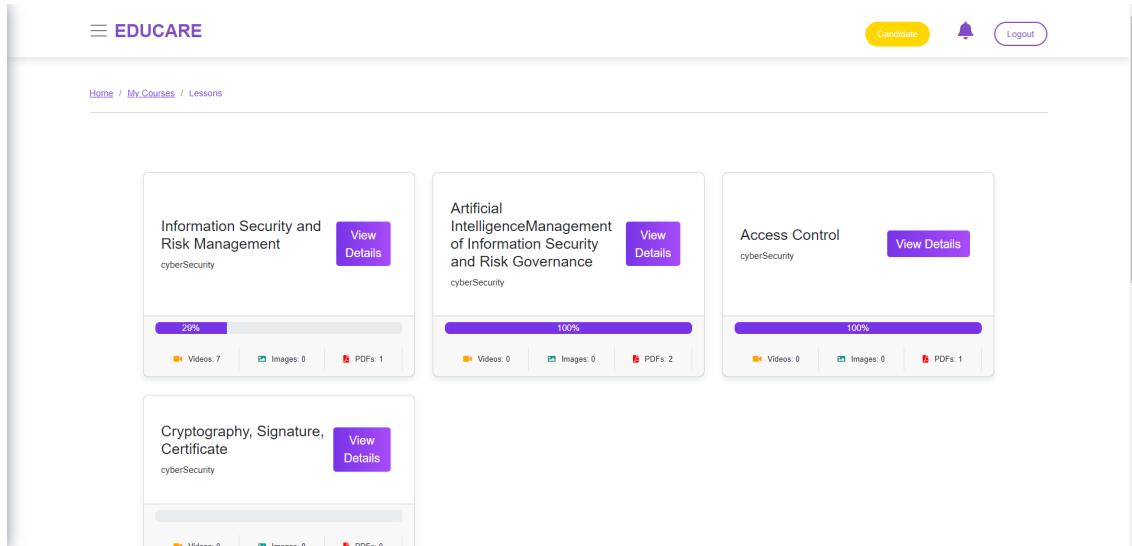


FIGURE 4.16 – Interface de consultation des leçons pour les candidats

Cette interface permet aux candidats de consulter ces leçons et de suivre leur état d'avancement de chaque leçon. À partir de cette interface, ils peuvent aussi consulter le nombre des ressources existant à une leçon.

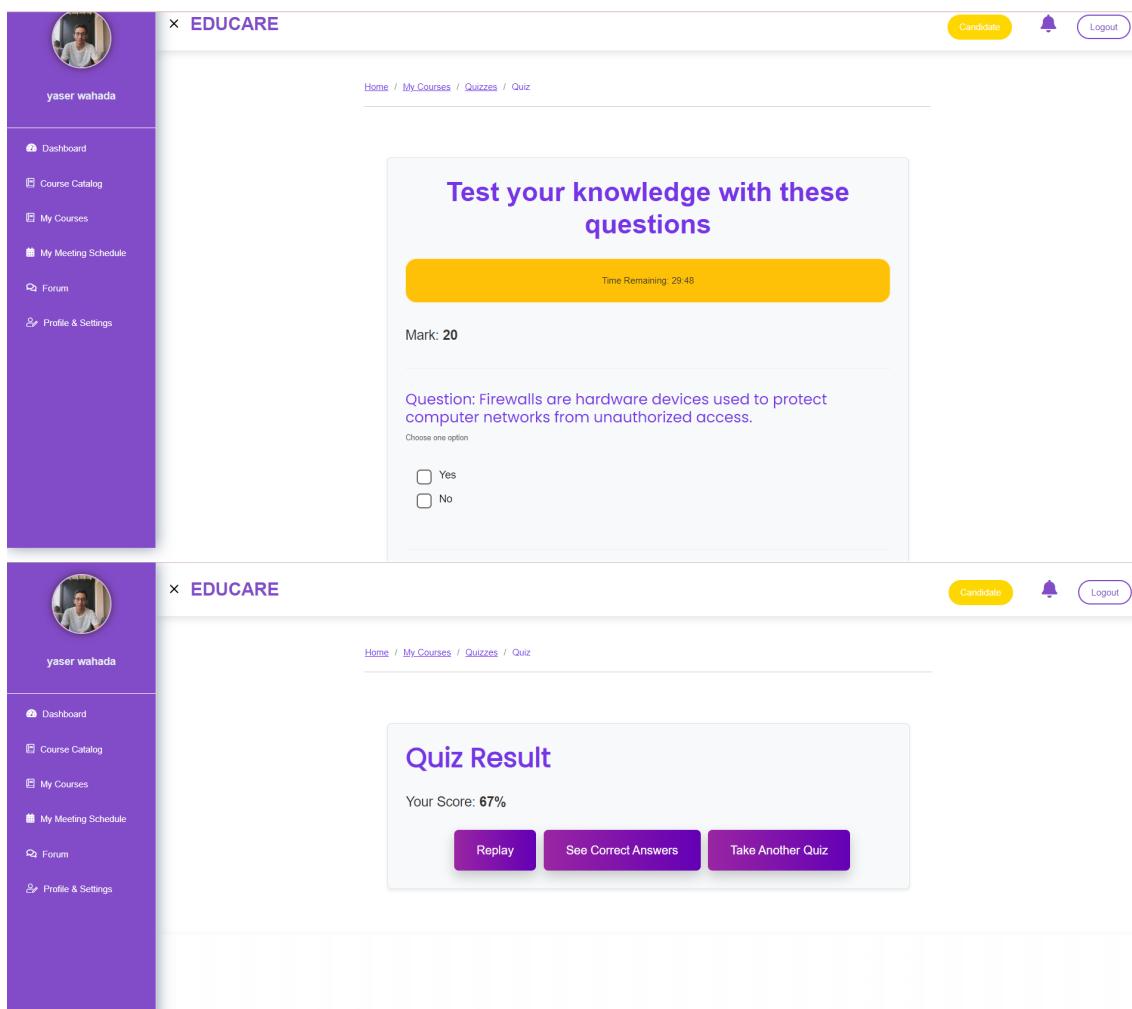


FIGURE 4.17 – Interface de passage de quiz pour candidat

Cette interface permet aux candidats de passer des quiz, de suivre leurs résultats et de voir les réponses correctes.

4.1.6 Tests

4.1.6.1 Tests unitaires

Dans le chapitre précédent du sprint 1, nous avons souligné l'importance cruciale des tests unitaires pour notre processus de développement. Ils ont été présentés comme des outils essentiels permettant de détecter les erreurs dès les premières étapes du développement. Dans cette section, nous poursuivons cette approche en détaillant les tests spécifiques que nous avons développés dans le cadre du sprint 2.

— Test Unitaire de Service

```

28
29
30  @Test
31  @DisplayName("Update Course - Invalid Course ID")
32  void updateCourse_InvalidCourseId_ThrowsEntityNotFoundException() {
33      // Mock data
34      Integer courseId = 1;
35      CourseRequest courseRequest = new CourseRequest();
36
37      // Mock behavior
38      when(courseRepository.findById(courseId)).thenReturn(Optional.empty());
39
40      // Method call and assertion
41      assertThrows(EntityNotFoundException.class, () -> {
42          courseService.updateCourse(courseId, courseRequest, imageUrl: null);
43      });
44
45      // Verify
46      verify(courseRepository, times(wantedNumberOfInvocations: 1)).findById(courseId);
47      verify(courseRepository, never()).save(any());
48      verify(courseMapper, never()).toCourseResponse(any());
49  }

```

FIGURE 4.18 – Test unitaire du service : mise à jour d'un cours avec un identifiant invalide

Le test illustré dans la figure ci-dessus simule le scénario où une tentative de mise à jour d'un cours est effectuée avec un identifiant de cours invalide. Dans ce cas, une exception de type EntityNotFoundException devrait être déclenchée, indiquant que le cours n'a pas été trouvé dans la base de données. Ce test garantit que notre système gère correctement les cas où des identifiants de cours invalides sont fournis lors de la mise à jour des cours.

— Test Unitaire de Contrôleur

```

38
39
40  @Test
41  @DisplayName("Test Enroll in Course Endpoint - Failure")
42  public void testEnrollInCourseEndpoint_Failure() throws Exception {
43      // Mocking service method behavior to throw an exception
44      when(courseService.enrollInCourse(any(Integer.class), any(Integer.class)))
45          .thenThrow(new EntityNotFoundException("Course not found"));
46
47      // Perform POST request to enroll in course
48      mockMvc.perform(post(urlTemplate: "/api/courses/courses/enroll"))
49          .param(name: "userId", ...values: "1")
50          .param(name: "courseId", ...values: "1")
51          .contentType(MediaType.APPLICATION_JSON)
52          .andExpect(status().isNotFound()); // Expecting 404 status code

```

The screenshot shows the Java code in a code editor. Below the code, there is a run configuration bar with the application name 'ElearningApplication' and the test name 'CourseControllerTest testEnrollInCourseEndpoint_Failure'. The status bar at the bottom indicates 'Tests passed: 1 of 1 test – 3 sec 312 ms' and the command line path 'C:\Program Files\Java\jdk-17\bin\java.exe'.

FIGURE 4.19 – Test unitaire du contrôleur : inscription à un cours - échec

Nous simulons le lancement d'une exception de type EntityNotFoundException lorsque la méthode enrollInCourse du service est appelée. Ensuite, nous envoyons une requête POST à l'endpoint d'inscription à un cours avec des identifiants d'utilisateur et de cours fictifs. Enfin, nous nous attendons à ce que le code d'état de la réponse soit 404 (Non trouvé), indiquant une tentative d'inscription échouée en raison de la non-existence du cours.

4.1.6.2 Tests avec Postman

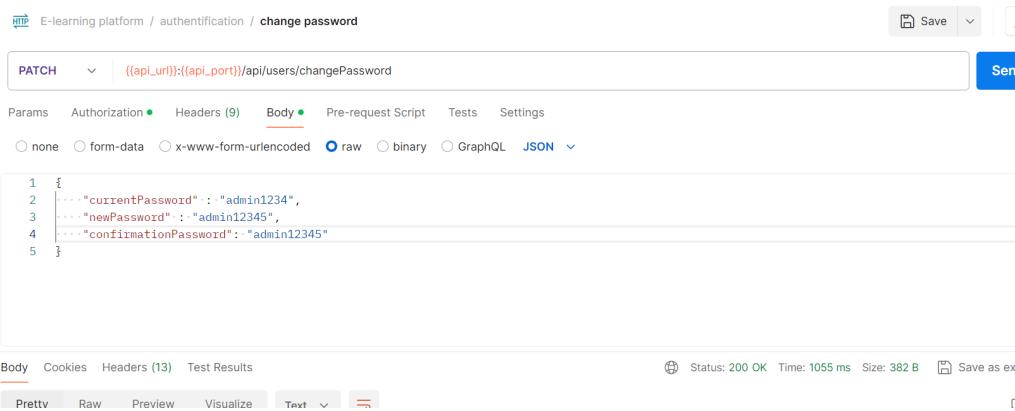


FIGURE 4.20 – Test Postman : modification du mot de passe

Ce test Postman montre l'exécution réussie de la modification du mot de passe utilisateur via l'API. L'endpoint utilisé "/api/users/changePassword", avec le corps de la requête contenant le mot de passe actuel, le nouveau mot de passe et sa confirmation. Le test inclut également une authentification via JWT. Le statut de la réponse est 200 OK.

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

Chapitre **5**

Troisième Sprint : Mise en Place du Forum de Discussion et Planification des Réunions

Plan

Introduction	117
5.1 Le troisième sprint	117
5.1.1 Spécification fonctionnelle	125
5.1.1.1 Raffinement de cas d'utilisation du sprint 3	125
5.1.1.2 Diagramme de cas d'utilisation détaillé du sprint3	127
5.1.1.3 Description textuelle des cas d'utilisation	128
5.1.2 La modélisation conceptuelle.....	131
5.1.2.1 Diagramme de classes	131
5.1.2.2 Diagramme de séquence détaillé.....	133
5.1.3 Réalisation.....	137
5.1.4 Tests	140
5.1.4.1 Les tests unitaires	140
5.1.4.2 Tests avec Postman.....	142
Conclusion	142

Introduction

Dans ce cinquième chapitre, nous suivrons les mêmes principes adoptés lors des deux premiers sprints de notre projet. Nous explorerons les étapes d'analyse, de conception, de réalisation et de test, mais cette fois-ci en nous focalisant sur les objectifs spécifiques du Troisième Sprint. Nous mettrons en avant les progrès accomplis dans la mise en place du forum de discussions et dans l'optimisation de la planification des réunions.

5.1 Le troisième sprint

Pour notre troisième sprint, nous continuons à suivre le même principe qui a fait ses preuves lors des deux sprints précédents. Après des discussions approfondies avec le Product Owner et une analyse de nos besoins actuels, nous avons décidé de nous concentrer sur la finalisation de la gestion du forum et des réunions.

Ainsi, notre objectif principal pour ce sprint est d'achever la mise en place de ces fonctionnalités essentielles, en suivant notre approche agile éprouvée. Voyons maintenant les histoires que nous avons sélectionnées pour ce sprint.

Id	Description de l'user story	Tâches effectuées	Estimation
1	En tant que candidat, je veux communiquer avec mes collègues et mes instructeurs via un forum de discussion attribué à chaque leçon.	<ul style="list-style-type: none"> - Mettre à jour la logique de création de leçon pour inclure la création automatique d'un forum de discussion associé. (Backend) - Implémenter les mécanismes d'attribution automatique du forum de discussion au candidat lorsqu'il consulte la leçon. (Backend) - Tester l'API pour vérifier la fonctionnalité de création automatique et d'accès au forum de discussion. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour afficher le forum de discussion lors de la consultation de la leçon. (Frontend) 	2 jours

2	<p>En tant que candidat, je souhaite pouvoir ajouter un message dans le forum de discussion en incluant le sujet du message et en pouvant attribuer différents types de contenus tels que du texte, des PDF, des vidéos, des enregistrements audio et des images.</p>	<ul style="list-style-type: none"> - Ajouter une route API dédiée permettant aux candidats d'ajouter un message dans le forum de discussion avec différents types de contenus, en incluant le sujet de la discussion. (Backend) - Tester l'API pour vérifier la fonctionnalité d'ajout de messages avec différents types de contenus. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour permettre aux candidats d'ajouter un message dans le forum, en incluant la saisie du sujet et du contenu du message, ainsi que la possibilité d'ajouter différents types de pièces jointes telles que des fichiers PDF, des images, des vidéos et des enregistrements audio. (Frontend) 	2 jours
3	<p>En tant que candidat, je souhaite pouvoir modifier un message existant dans le forum de discussion afin de mieux refléter son contenu.</p>	<ul style="list-style-type: none"> - Ajouter une route API dédiée permettant aux candidats de modifier un message existant dans le forum. (Backend) - Tester l'API avec Postman pour vérifier la fonctionnalité de modification du message. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour permettre aux candidats de modifier un message existant dans le forum de manière conviviale. (Frontend) 	1 jour
4	<p>En tant que candidat, je souhaite pouvoir supprimer mon message dans le forum si nécessaire.</p>	<p>Ajouter une route API dédiée permettant aux candidats de supprimer leur propre message dans le forum. (Backend)</p> <p>Tester l'API avec Postman pour vérifier la fonctionnalité de suppression de message. (Tests avec Postman)</p> <p>Mettre à jour l'interface utilisateur pour permettre aux candidats de supprimer leur propre message dans le forum de manière intuitive. (Frontend)</p>	1 jour

5	<p>En tant que candidat, je veux pouvoir consulter tous les messages créés par mes collègues dans la même leçon afin de suivre les échanges et les contributions de chacun. Je souhaite également avoir la possibilité de filtrer les messages par sujet pour trouver rapidement ceux qui m'intéressent le plus.</p>	<ul style="list-style-type: none"> - Mettre à jour la logique backend pour permettre aux candidats de consulter tous les messages créés par leurs collègues dans la même leçon. (Backend) - Tester l'API avec Postman pour vérifier la fonctionnalité de consultation des messages des candidats, y compris le filtrage par sujet. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour afficher de manière claire et conviviale tous les messages créés par les collègues dans la même leçon, en ajoutant un mécanisme de filtrage par sujet. (Frontend) 	2 jours
6	<p>En tant que candidat, je veux pouvoir consulter toutes les discussions créées par mes collègues dans la même leçon afin de suivre les échanges et les contributions de chacun.</p>	<ul style="list-style-type: none"> - Mettre à jour la logique backend pour permettre aux candidats de consulter toutes les discussions créées par leurs collègues dans la même leçon. (Backend) - Tester l'API avec Postman pour vérifier la fonctionnalité de consultation des discussions des collègues. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour afficher de manière claire et conviviale toutes les discussions créées par les collègues dans la même leçon. (Frontend) 	2 jours

7	<p>En tant qu'instructeur, je veux pouvoir consulter tous les messages créés par mes candidats inscrits dans mes cours afin de suivre leur progression et de faciliter la communication. Je souhaite également avoir la possibilité de filtrer les messages soit par leçon, soit par sujet, pour trouver rapidement ceux qui sont pertinents pour chaque partie du cours.</p>	<ul style="list-style-type: none"> - Mettre à jour la logique backend pour permettre aux instructeurs de consulter tous les messages créés par leurs candidats inscrits dans leurs cours, avec la possibilité de filtrer soit par leçon, soit par sujet. (Backend) - Tester l'API avec Postman pour vérifier la fonctionnalité de consultation des messages des candidats, y compris le filtrage par sujet et par leçon. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour permettre aux instructeurs de consulter de manière claire et conviviale tous les messages de leurs candidats, avec des options de filtrage par leçon ou par sujet. (Frontend) 	2 jours
7	<p>En tant que candidat ou instructeur, je souhaite pouvoir ajouter des réponses aux messages existants dans le forum de discussion, afin de participer activement aux échanges et de contribuer au contenu pédagogique.</p>	<ul style="list-style-type: none"> - Ajouter une fonctionnalité dans l'API backend permettant aux candidats et aux instructeurs d'ajouter des réponses aux messages existants. (Backend) - Tester l'API avec Postman pour vérifier la fonctionnalité d'ajout de réponses aux messages. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour permettre aux candidats et aux instructeurs d'ajouter des réponses aux messages de manière intuitive. (Frontend) 	2 jours

8	<p>En tant que candidat ou instructeur, je souhaite avoir la possibilité de modifier mes réponses aux messages existants dans le forum, afin de corriger d'éventuelles erreurs ou d'apporter des clarifications supplémentaires.</p>	<ul style="list-style-type: none"> - Ajouter une fonctionnalité dans l'API backend permettant aux candidats et aux instructeurs de modifier leurs réponses aux messages existants. (Backend) - Tester l'API avec Postman pour vérifier la fonctionnalité de modification des réponses aux messages. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour permettre aux candidats et aux instructeurs de modifier leurs réponses aux messages de manière intuitive. (Frontend) 	1 jour
9	<p>En tant que candidat ou instructeur, je veux avoir la possibilité de supprimer mes réponses aux discussions existantes dans le forum, afin de retirer des informations incorrectes ou obsolètes.</p>	<ul style="list-style-type: none"> - Ajouter une fonctionnalité dans l'API backend permettant aux candidats et aux instructeurs de supprimer leurs réponses aux discussions existantes. (Backend) - Tester l'API avec Postman pour vérifier la fonctionnalité de suppression des réponses aux discussions. (Tests avec Postman) - Mettre à jour l'interface utilisateur pour permettre aux candidats et aux instructeurs de supprimer leurs réponses aux discussions de manière intuitive. (Frontend) 	1 jour
11	<p>En tant qu'instructeur, je souhaite avoir la possibilité de récupérer facilement le lien de réunion à partir de l'API Jitsi Meet, afin de simplifier et d'accélérer le processus de planification des réunions.</p>	<ul style="list-style-type: none"> - Concevoir et développer une interface utilisateur conviviale permettant à l'instructeur de récupérer rapidement le lien de réunion. (Frontend) - Intégrer l'API Jitsi Meet dans l'application pour permettre la génération de liens de réunion. (Frontend) 	1 jour

12	<p>En tant qu'instructeur, je veux pouvoir insérer les détails d'une réunion (meet) pour mes cours. Je dois pouvoir spécifier le cours, ajouter un titre, indiquer la durée, insérer un lien de meet créé et fournir une description de la réunion. Si nécessaire, je peux également associer cette réunion à une leçon spécifique du cours. Une fois les détails ajoutés, un email avec les informations de la réunion sera envoyé à tous les candidats inscrits à ce cours.</p>	<ul style="list-style-type: none"> - Créer une route API pour gérer l'ajout de planning de réunion. (Backend) - Mettre en place la logique de gestion des plannings de réunion. (Backend) - Implémenter la fonctionnalité d'envoi d'email aux candidats inscrits au cours une fois le planning ajouté. (Backend) - Créer un formulaire permettant à l'instructeur de saisir les détails du planning (titre, durée, lien de meet, description, cours, leçon). (Frontend) - Afficher une confirmation à l'instructeur une fois que le planning a été ajouté avec succès. (Frontend) - Effectuer des tests d'intégration avec Postman pour vérifier la fonctionnalité d'ajout de planning de réunion. (Tests avec Postman) - Réaliser des tests unitaires pour valider la logique backend. 	2 jours
12	<p>En tant qu'instructeur, je souhaite pouvoir modifier les détails d'une réunion planifiée, tels que le titre, la durée, le lien de la réunion, la description et la leçon associée, afin de les mettre à jour en cas de changement. Une fois les détails modifiés, un email contenant les nouvelles informations doit être envoyé à tous les candidats inscrits à ce cours.</p>	<ul style="list-style-type: none"> - Créer une route API dédiée pour la modification des détails du planning de réunion. (Backend) - Implémenter la logique nécessaire pour mettre à jour les détails du planning. (Backend) - Configurer l'envoi d'email pour notifier les candidats inscrits une fois les détails du planning modifiés. (Backend) - Créer un formulaire de modification des détails du planning dans l'interface utilisateur. (Frontend) - Effectuer des tests d'intégration avec Postman pour vérifier la fonctionnalité de modification des détails du planning. (Tests avec Postman) - Réaliser des tests unitaires pour valider la logique backend de modification des détails du planning. 	1 jour

13	<p>En tant qu'instructeur, je veux avoir la possibilité de supprimer une réunion planifiée pour annuler une séance de cours. Lorsque la réunion est supprimée, un email d'annulation doit être envoyé à tous les candidats inscrits pour les informer de cette annulation.</p>	<ul style="list-style-type: none"> - Créer une route API dédiée pour la suppression d'une réunion planifiée. (Backend) - Implémenter la logique nécessaire pour supprimer les détails du planning. (Backend) - Configurer l'envoi d'email d'annulation pour notifier les candidats inscrits lorsque la réunion est supprimée. (Backend) - Ajout de l'option de suppression à chaque élément de la liste des réunions planifiées. (Frontend) - Effectuer des tests d'intégration pour vérifier le fonctionnement de l'API de suppression de réunion. (Tests avec Postman) 	1 jour
13	<p>En tant que candidat, je souhaite pouvoir consulter les détails des réunions planifiées afin de me tenir informé des séances de cours prévues.</p>	<ul style="list-style-type: none"> - Créer une route API dédiée pour récupérer les détails des réunions planifiées accessibles par les candidats. (Backend) - Implémenter la logique nécessaire pour récupérer les détails des réunions planifiées dans la base de données. (Backend) - Créer une interface utilisateur permettant aux candidats de consulter les détails des réunions planifiées. (Frontend) - Ajouter la fonctionnalité permettant aux candidats de rejoindre la réunion directement à partir de la liste des réunions planifiées. (Frontend) - Afficher les détails des réunions planifiées dans l'interface utilisateur, y compris le titre, la date, la durée et le lien de la réunion. (Frontend) 	2 jours

16	<p>En tant que candidat, je souhaite recevoir des notifications par e-mail lorsqu'un planning de réunion est ajouté, modifié ou annulé pour les cours auxquels je suis inscrit, afin de rester informé des changements dans mon emploi du temps.</p>	<ul style="list-style-type: none"> - Mettre en place la logique pour envoyer un e-mail aux candidats lors de l'ajout d'un nouveau planning de réunion. (Backend) - Configurer l'envoi d'e-mails pour notifier les candidats en cas de modification des détails d'un planning existant. (Backend) - Ajouter la fonctionnalité d'envoi d'e-mails pour informer les candidats de l'annulation d'une réunion planifiée. (Backend) - Réaliser des tests pour vérifier que les e-mails sont correctement envoyés aux candidats en cas d'ajout, de modification ou d'annulation d'un planning. (Tests avec Postman) - Mettre en place la gestion des autorisations pour garantir que seuls les candidats inscrits reçoivent les notifications par e-mail. (Backend) 	1 jour
15	<p>En tant que candidat, je souhaite recevoir des notifications par e-mail lorsqu'un planning de réunion est ajouté, modifié ou annulé pour les cours auxquels je suis inscrit, afin de rester informé des changements dans mon emploi du temps.</p>	<ul style="list-style-type: none"> - Mettre en place la logique pour envoyer un e-mail aux candidats lors de l'ajout d'un nouveau planning de réunion. (Backend) - Configurer l'envoi d'e-mails pour notifier les candidats en cas de modification des détails d'un planning existant. (Backend) - Ajouter la fonctionnalité d'envoi d'e-mails pour informer les candidats de l'annulation d'une réunion planifiée. (Backend) - Réaliser des tests pour vérifier que les e-mails sont correctement envoyés aux candidats en cas d'ajout, de modification ou d'annulation d'un planning. (Tests avec Postman) - Mettre en place la gestion des autorisations pour garantir que seuls les candidats inscrits reçoivent les notifications par e-mail. (Backend) 	1 jour

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, réalisation et le test.

5.1.1 Spécification fonctionnelle

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

5.1.1.1 Raffinement de cas d'utilisation du sprint 3

Le diagramme de cas d'utilisation du sprint 3 modélise les différentes fonctionnalités du sprint. Il présente trois acteurs héritant du membre, représentés par l'acteur membre, qui partagent des fonctionnalités communes telles que :

- **S'authentifier.**

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

5.1.1.2 Diagramme de cas d'utilisation détaillé du sprint3

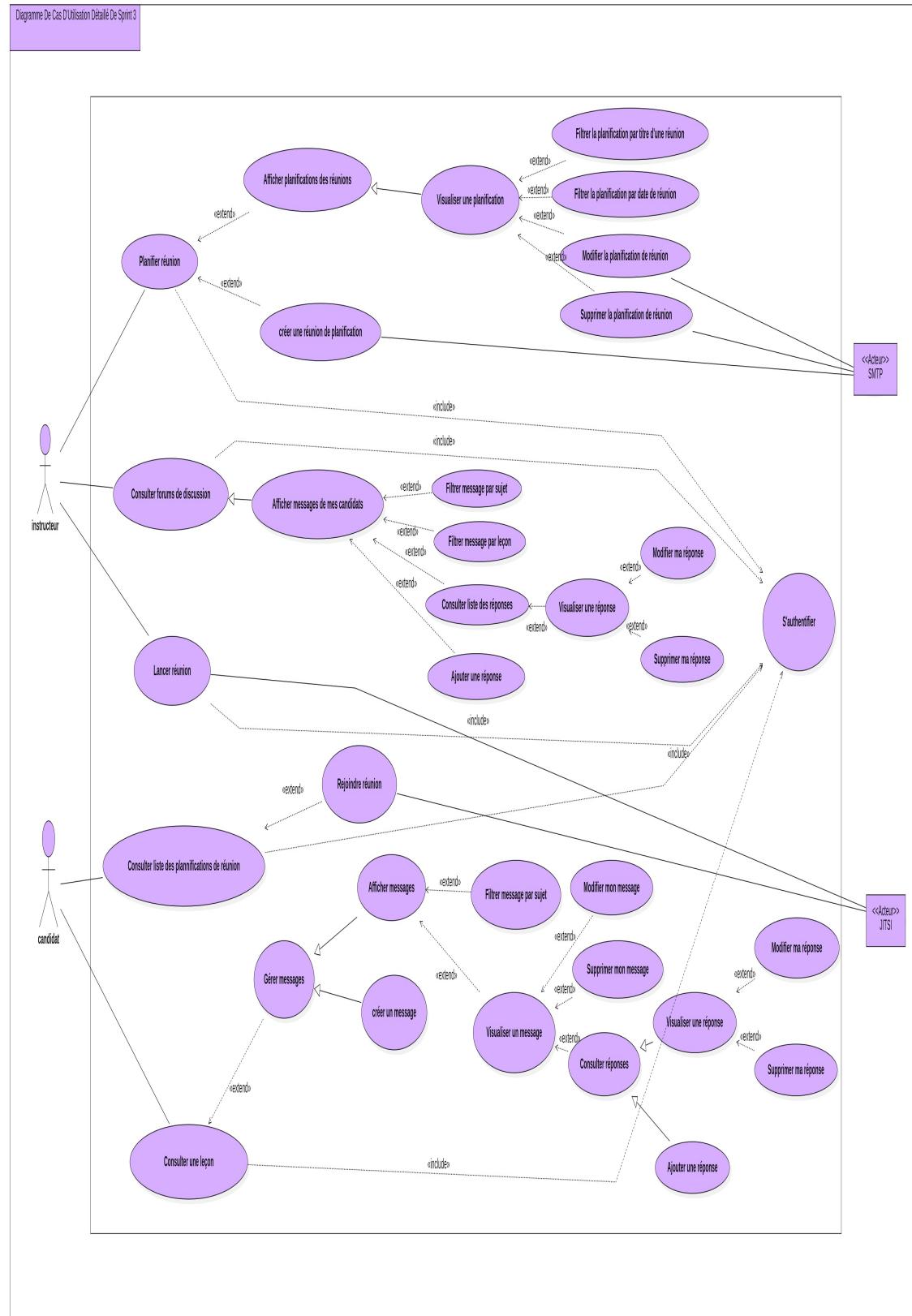


FIGURE 5.1 – Diagramme de cas d'utilisation détaillé du Sprint 3 »

5.1.1.3 Description textuelle des cas d'utilisation

Le tableau ci-dessous explique le processus de création d'un planning d'une réunion par un instructeur.

TABLE 5.2 – Description textuelle du cas d'utilisation « Créer un planning de réunion »

Cas d'utilisation	Créer un planning de réunion
Acteur principal	Instructeur
Précondition	L'instructeur est authentifié sur la plateforme.
Postcondition	Une nouvelle planification de réunion est ajoutée au cours sélectionné et un email personnalisé est envoyé aux candidats inscrits.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'instructeur accède à la page de planification des réunions. 2. Le système affiche un formulaire permettant à l'instructeur de saisir les détails de la réunion, tels que le titre de la réunion, la date de la réunion, la durée, et l'URL de la réunion. 3. L'instructeur remplit les informations requises pour la réunion. 4. L'instructeur sélectionne le cours associé à la réunion et peut également sélectionner la leçon associée. 5. Une fois toutes les informations saisies, l'instructeur valide l'ajout de la réunion. 6. Le système enregistre la nouvelle réunion dans le cours sélectionné. 7. Le système envoie un email personnalisé contenant tous les détails de la réunion planifiée aux candidats inscrits au cours. 8. Le système affiche un message de confirmation indiquant que la réunion a été ajoutée avec succès.
Scénario alternatif	<p>SA1. Erreur de validation du formulaire (champs mal renseignés ou date incorrecte). SA1 commence après l'étape 5 du scénario nominal.</p> <p>6. Le système affiche un message d'erreur indiquant les champs mal renseignés ou la date de réunion incorrecte.</p> <p>SA2. Champs vides. SA2 commence après l'étape 5 du scénario nominal.</p> <p>6. Le système affiche un message d'erreur demandant à l'instructeur de remplir tous les champs obligatoires.</p>

Le tableau ci-dessous explique le processus de consultation et de participation à une réunion planifiée par un candidat inscrit à un cours.

TABLE 5.3 – Description textuelle du cas d'utilisation « Rejoindre une réunion »

Cas d'utilisation	Rejoindre une Réunion
Acteur principal	Candidat
Précondition	Le candidat est authentifié sur la plateforme et inscrit à un cours ayant une réunion planifiée.
Postcondition	Le candidat rejoint avec succès la réunion planifiée.
Scénario Nominal	<ol style="list-style-type: none"> Le candidat accède à la liste des réunions planifiées pour ses cours. Le système affiche les détails des réunions prévues, y compris le titre, la date, et la durée de la réunion, ainsi qu'un bouton pour rejoindre la réunion. Le système redirige le candidat vers la réunion en utilisant Jitsi Meet, l'outil de réunion en ligne associé à la réunion sélectionnée.
Scénario alternatif	<p>SA1. Aucune réunion planifiée disponible. SA1 commence après l'étape 1 du scénario nominal.</p> <ol style="list-style-type: none"> Le système informe le candidat qu'aucune réunion planifiée n'est actuellement disponible pour ses cours. <p>SA2. Date de réunion passée SA2 commence après l'étape 2 du scénario nominal.</p> <ol style="list-style-type: none"> Le système affiche les détails de la réunion, mais désactive le bouton "Rejoindre la Réunion".

Le tableau ci-dessous explique le processus d'ajouter une discussion par un candidat.

TABLE 5.4 – Description textuelle du cas d'utilisation « Créer une discussion »

Cas d'utilisation	Créer une Discussion
Acteur principal	Candidat
Précondition	Le candidat est authentifié sur la plate-forme et inscrit à un cours.
Postcondition	Une nouvelle discussion est ajoutée avec succès dans le forum associé à la leçon sélectionnée par le candidat.
Scénario Nominal	<ol style="list-style-type: none"> Le candidat sélectionne une leçon spécifique dans la liste des leçons du cours auquel il est inscrit.

	<p>2. Le système affiche les ressources de la leçon sélectionnée, y compris le forum de discussion associé à cette leçon.</p> <p>3. Le candidat identifie un bouton "Add discussion" et le sélectionne.</p> <p>4. Le système affiche un formulaire d'ajout de nouvelle discussion qui inclut les champs obligatoires pour le sujet et le texte de la discussion, ainsi que des options pour joindre des pièces jointes.</p> <p>5. Le candidat saisit le sujet et le texte de la discussion dans les champs appropriés, tout en ayant la possibilité d'ajouter une seule pièce jointe, telle qu'une image, une vidéo, un fichier PDF ou un enregistrement audio, à l'aide des options fournies dans le formulaire.</p> <p>6. Une fois la saisie terminée, le candidat soumet le formulaire en cliquant sur le bouton "Submit".</p> <p>7. Le système enregistre la nouvelle discussion dans la base de données, associée au candidat en tant qu'auteur et au forum de la leçon spécifiée.</p> <p>8. Le système affiche un message de confirmation indiquant que la discussion a été ajoutée avec succès dans le forum de la leçon sélectionnée.</p>
Scénario alternatif	<p>SA1. Sujet ou contenu de la discussion vide.</p> <p>SA1 commence après l'étape 5 du scénario nominal.</p> <p>6. Le système affiche un message d'erreur demandant au candidat de remplir tous les champs obligatoires.</p> <p>7. Le candidat corrige les champs vides et soumet à nouveau le formulaire.</p>

Le tableau ci-dessous décrit le processus d'ajout de réponses aux discussions des candidats par l'instructeur.

TABLE 5.5 – Description textuelle du cas d'utilisation « Ajouter une réponse »

Cas d'utilisation	Ajouter une réponse
Acteur principal	Instructeur
Précondition	L'instructeur est authentifié sur la plateforme. Des discussions soumises par les candidats sont disponibles dans le forum associé aux leçons de l'instructeur.
Postcondition	La réponse de l'instructeur est enregistrée avec succès et associée à la discussion correspondante dans le forum.

Scénario Nominal	<ol style="list-style-type: none"> 1. L'instructeur accède à l'interface du forum. 2. Le système affiche les discussions soumises par les candidats dans le forum. 3. L'instructeur sélectionne une discussion spécifique à laquelle il souhaite répondre. 4. Le système affiche le contenu de la discussion sélectionnée, y compris les messages soumis par les candidats. 5. L'instructeur rédige sa réponse dans le champ de réponse prévu à cet effet. 6. Une fois la réponse rédigée, l'instructeur soumet sa réponse en cliquant sur le bouton "Submit". 7. Le système enregistre la nouvelle réponse dans la base de données, associée à la discussion sélectionnée. 8. Le système affiche un message de confirmation indiquant que la réponse a été ajoutée avec succès à la discussion.
Scénario alternatif	<p>SA1. Champ de réponse vide. SA1 commence après l'étape 5 du scénario nominal.</p> <ol style="list-style-type: none"> 6. Le système détecte que le champ de réponse est vide et affiche un message d'erreur. 7. L'instructeur remplit le champ de réponse avec du contenu valide et soumet à nouveau sa réponse. 8. Le système enregistre la nouvelle réponse dans la base de données, associée à la discussion sélectionnée. 9. Le système affiche un message de confirmation indiquant que la réponse a été ajoutée avec succès à la discussion.

5.1.2 La modélisation conceptuelle

5.1.2.1 Diagramme de classes

Suite au diagramme du sprint 2, le diagramme de classes suivant représente les nouvelles classes et entités impliquées dans le processus de gestion de planning et de forum du troisième sprint. Ce diagramme permet de visualiser la structure et les relations entre ces classes, ce qui facilite la compréhension de la manière dont les différentes entités du système interagissent dans ce processus.

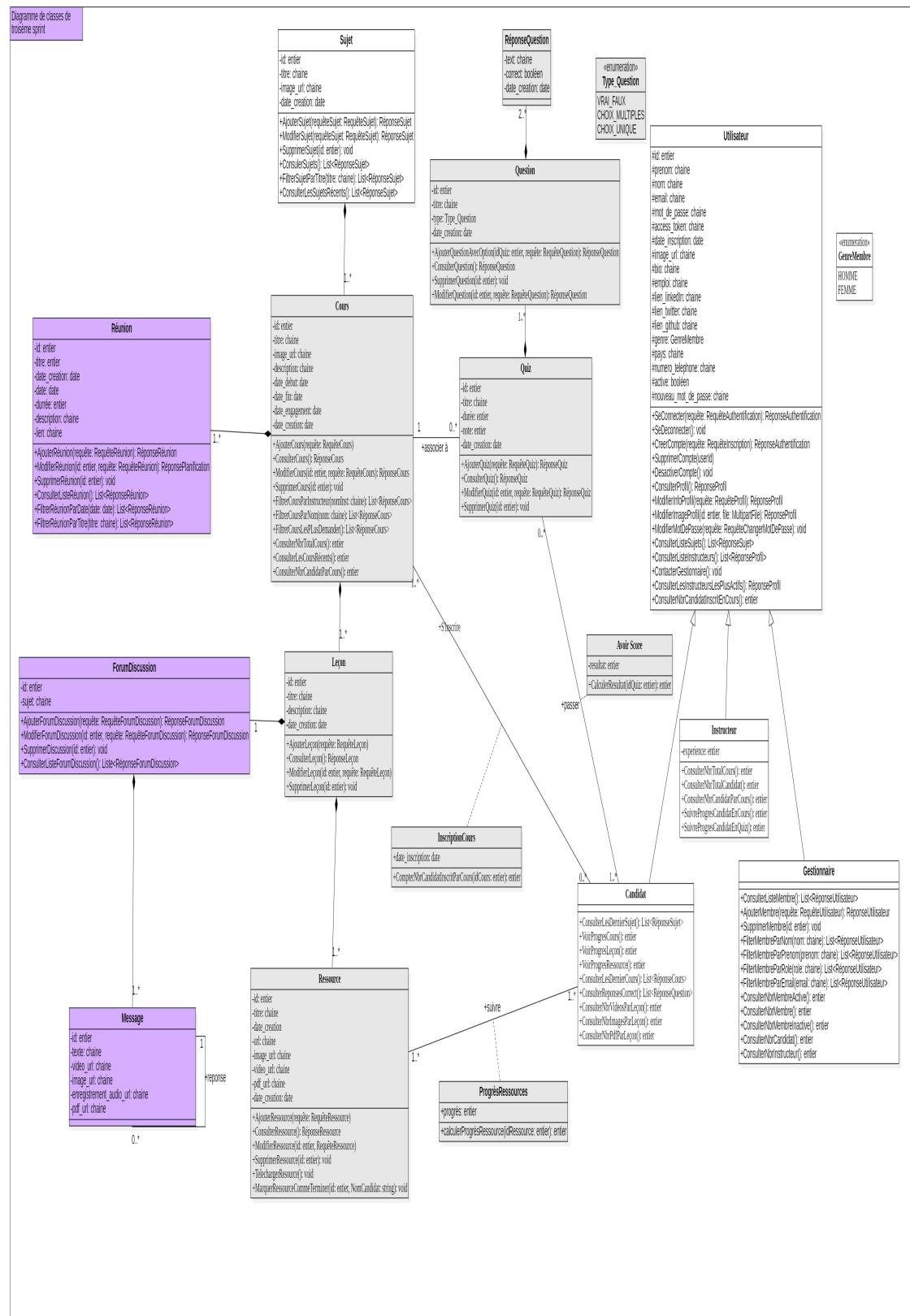


FIGURE 5.2 – Diagramme de classes du troisième sprint

5.1.2.2 Diagramme de séquence détaillé

— Diagramme de séquence associé à « Crée Une Discussion »

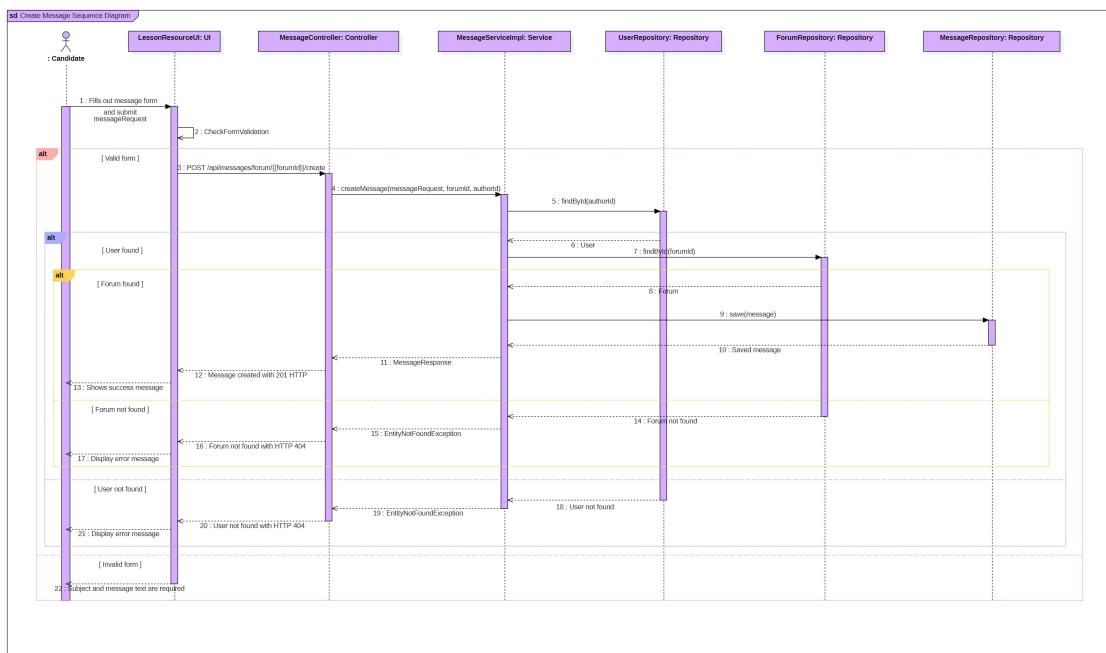


FIGURE 5.3 – Diagramme de séquence du cas d'utilisation « Crée une discussion »

Le diagramme de séquence ci-dessus illustre le processus de création d'un message par un candidat, tel que décrit dans le cas d'utilisation "Créer un message".

Le processus débute quand le candidat accède à l'interface utilisateur de la leçon concernée et déclenche le processus en cliquant sur le bouton "Add Message". Ensuite, il remplit le formulaire de message avec les informations requises, telles que le sujet, le contenu du message, et éventuellement des fichiers joints.

Une fois le formulaire rempli, le candidat soumet les données en cliquant sur "Submit". Le système effectue alors une vérification pour s'assurer que tous les champs obligatoires sont remplis et affiche un message d'erreur si ce n'est pas le cas.

Une fois que tous les champs sont correctement remplis, une requête HTTP POST est envoyée au ServiceController, avec l'identifiant du candidat récupéré lors de l'authentification. Le MessageController appelle ensuite la méthode createMessage de MessageServiceImpl, en transmettant les détails du message tels que l'identifiant du forum de discussion et de l'auteur.

La MessageServiceImpl utilise alors le UserRepository pour récupérer les informations sur l'auteur depuis la base de données, et le ForumRepository pour obtenir le forum associé à la leçon. Une fois toutes les données nécessaires récupérées, la demande de message est transformée en une entité Message, puis sauvegardée via le MessageRepository.

Une fois que le message est enregistré avec succès, il est converti en un objet MessageResponse et renvoyé au MessageController, qui répond à l'interface utilisateur de la leçon avec une réponse HTTP 201, indiquant que la création du message a été effectuée avec succès. Le nouveau message est alors affiché dans l'interface utilisateur, permettant au candidat de visualiser sa contribution.

- **Diagramme de séquence associé à « Crer Un Planning D'Une Runion »**

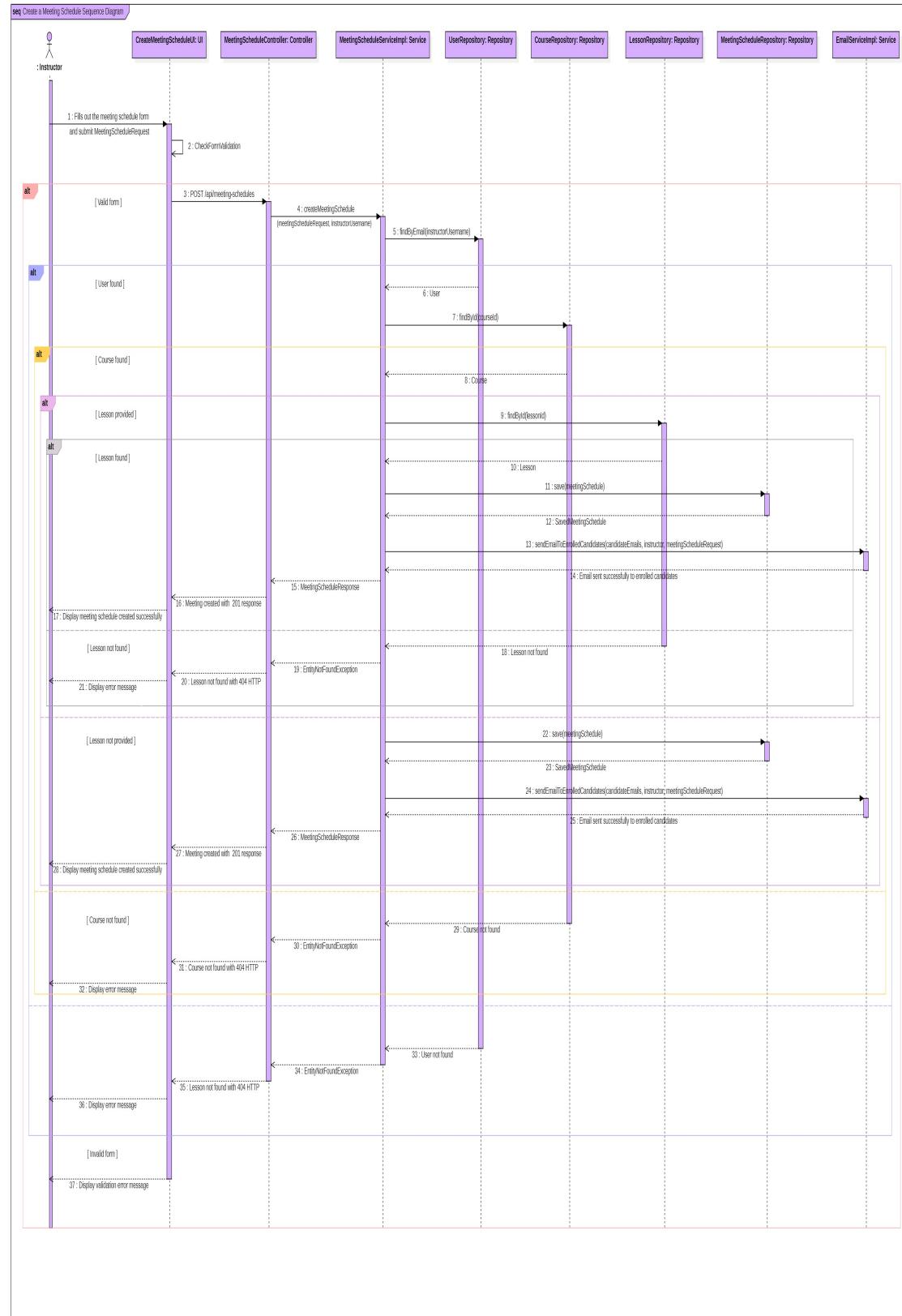


FIGURE 5.4 – Diagramme de séquence du cas d'utilisation « Créer une réunion de planification »

Le diagramme de séquence ci-dessus illustre le processus de création d'une planification de réunion dans le système.

Le processus débute lorsque l'instructeur accède à l'interface utilisateur dédiée à la création de planifications de réunion, nommée "Create Meeting Schedule UI". L'instructeur remplit un formulaire structuré comprenant des champs obligatoires tels que le titre de la réunion, sa durée prévue, sa description ainsi l'URL de la planification de la réunion pour faciliter l'accès des participants.

Une fois le formulaire rempli et tous les champs obligatoires validés, une requête HTTP POST est envoyée au MeetingPlanningController, chargé de gérer les demandes relatives aux planifications de réunion. Ce dernier extrait ensuite les détails de la planification de la réunion de la requête.

Le MeetingPlanningController appelle ensuite la méthode createMeetingSchedule de MeetingPlanningServiceImpl, située dans la couche de service, pour créer la planification de réunion. Cette méthode prend en charge les détails de la planification ainsi que l'identifiant de l'instructeur pour garantir la cohérence des données.

Dans MeetingPlanningServiceImpl, les données fournies sont méticuleusement traitées. L'instructeur est recherché dans la base de données via l'UserRepository, tandis que les informations facultatives sur le cours et la leçon associés à la planification de réunion sont récupérées depuis les CourseRepository et LessonRepository respectivement.

Une fois toutes les informations consolidées, un nouvel objet MeetingSchedule est créé avec ces détails et sauvegardé de manière sécurisée dans la base de données via le MeetingScheduleRepository.

Enfin, une réponse confirmant la réussite de la création de la planification de réunion est renvoyée au client, et la nouvelle planification est affichée dans l'interface utilisateur, permettant à l'instructeur de visualiser sa contribution.

5.1.3 Réalisation

Forum for Lesson: Information Security & Risk Management

Search
Reset

No discussions found.

Subject

Message Type

Attach PDF File

Choose File

risks-is.pdf

Message

I'm curious about the risk assessment procedures we employ and would like to delve deeper into this topic. I've prepared a document outlining some specific questions and areas of interest. Could you please review it and provide your insights?

Attach Record File

Click To Start Recording

Recorded Audio Preview:

0:00 / 0:31

Submit
Cancel

FIGURE 5.5 – Interface d'ajout d'un message

Cette interface permet aux candidats de créer des messages incluant du texte ainsi que des médias tels que des enregistrements audio ou vidéo, et des images. Cela favorise une communication complète et efficace entre les candidats et leur instructeur.

Isaa Molloug

- Dashboard
- Course Management
- Quiz Management
- Video Conference
- Meeting Schedule
- ✉ Forum
- 👤 Profile & Settings

DISCUSSION FORUM

Home / Discussion forum

Search
Reset

Inquiry about Risk Assessment Procedures

I'm curious about the risk assessment procedures we employ and would like to delve deeper into this topic. I've prepared a document outlining some specific questions and areas of interest. Could you please review it and provide your insights?

Hey Yaser, thanks for reaching out and sharing your questions. I've read through them, and they're great. We mostly rely on the FAIR framework for risk assessment, which works well for us. I'll look over your document carefully and get back to you soon.

By Isaa Molloug on Jun 8, 2024, 9:40:01 PM

Hey Yaser, thanks for reaching out and sharing your questions. I've read through them, and they're great. We mostly rely on the FAIR framework for risk assessment, which works well for us. I'll look over your document carefully and get back to you soon.

By Isaa Molloug on Jun 8, 2024, 9:40:01 PM

Update
Delete

Add Answer

FIGURE 5.6 – Interface de gestion des réponses de l'instructeur aux messages de ses candidats dans les forums de discussion

Cette interface permet à l'instructeur de consulter les messages envoyés par ses candidats dans les forums de discussion. Elle offre également la possibilité à l'instructeur de modifier ses réponses précédentes ou de les supprimer si nécessaire.

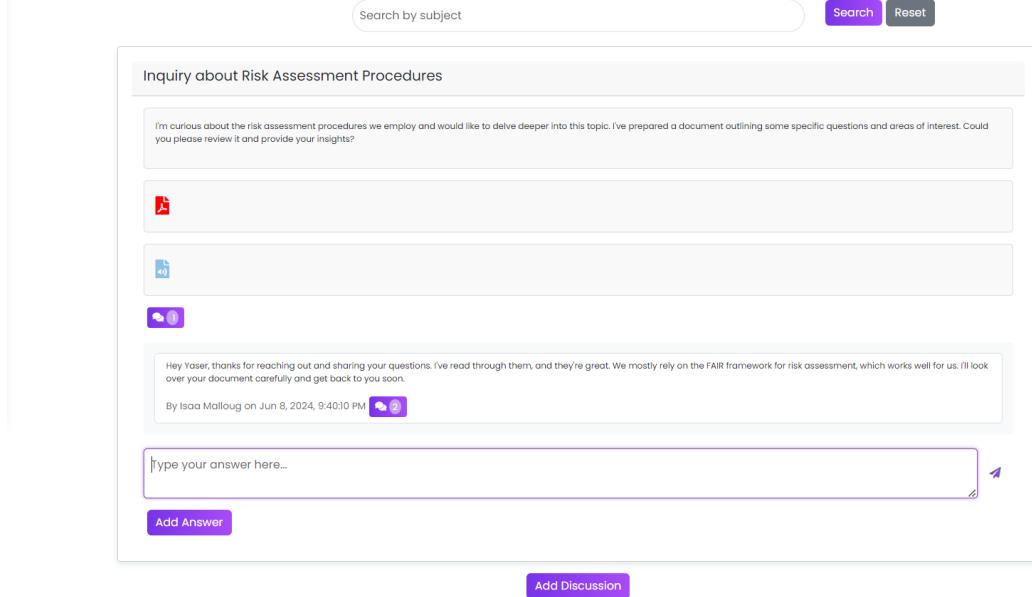


FIGURE 5.7 – Interface de réponse du candidat dans le forum de discussion

Cette interface offre au candidat la possibilité de rédiger sa réponse directement dans le fil de discussion, facilitant ainsi la communication et l'échange d'informations avec l'instructeur et les autres candidats.

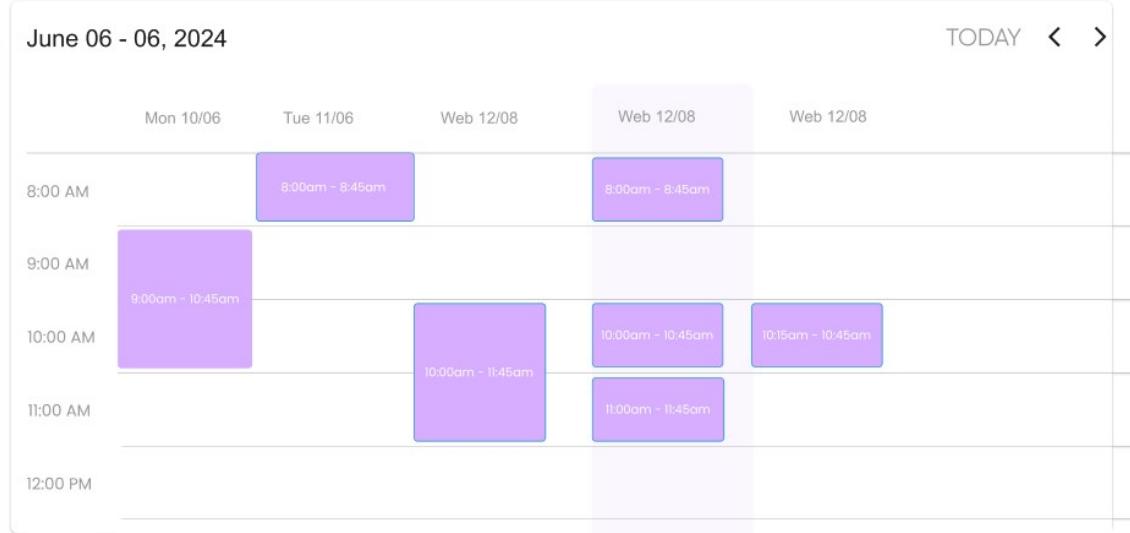


FIGURE 5.8 – Interface du calendrier de planification des réunions par l'instructeur

Cette interface présente un calendrier permettant à l'instructeur de choisir une date pour planifier une réunion. Elle offre une vue visuelle et pratique des dates disponibles, facilitant ainsi la planification et l'organisation des réunions.

CREATE MEETING SCHEDULE

Meeting Title:

Duration (in minutes):

Meeting URL: Launch Meeting

Meeting Description:

Select Course: Select a course

Select Lesson (optional): ▼

Create Meeting Schedule

◀ Back to Meeting Schedule List

FIGURE 5.9 – Interface d'ajout d'une planification de réunion par l'instructeur

Ce formulaire permet à l'instructeur de spécifier les détails de la réunion, tels que le titre, la durée, l'URL de la réunion, la description de la réunion, le cours associé, ainsi que la leçon (optionnelle). Une fois les informations saisies, l'instructeur peut soumettre le formulaire pour planifier la réunion.



FIGURE 5.10 – Email de notification pour une réunion planifiée

Cet exemple d'email de notification est envoyé aux candidats inscrits au cours pour les informer de la planification d'une réunion. L'email comprend des détails tels que le titre de la réunion, la date, l'heure et l'URL de la réunion, ainsi que toute autre

information pertinente. Il vise à informer les candidats de manière claire et concise sur les détails de la réunion à venir.

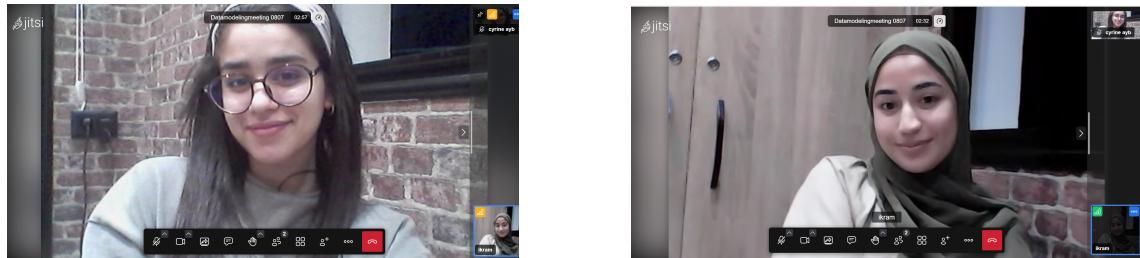


FIGURE 5.11 – Interfaces de rejoindre une réunion

Les interfaces permettent aux candidats de rejoindre une réunion planifiée. Ils offrent un accès à la salle de réunion où les participants peuvent interagir, partager des écrans et échanger des messages en direct.

A screenshot of the EduCare instant meeting interface. At the top, there's a navigation bar with 'EDUCARE' on the left, 'Instructor' and 'Logout' on the right. Below it is a large button labeled 'START A VIDEO CALL'. Underneath is a form field with placeholder text 'Enter a unique room name and click "Create Room". Share this room name with the other participants to join the call.' followed by a text input containing '123450987' and a 'Create Room' button. At the bottom, there are three sections: 'About EduCare' (describing EduCare as a cutting-edge online learning platform), 'Useful Links' (listing Home, Topics, Courses, Instructors, About Us, and Contact Us), and 'Contact Details' (listing address, email, and phone number).

FIGURE 5.12 – Interface de réunion instantanée pour l'instructeur

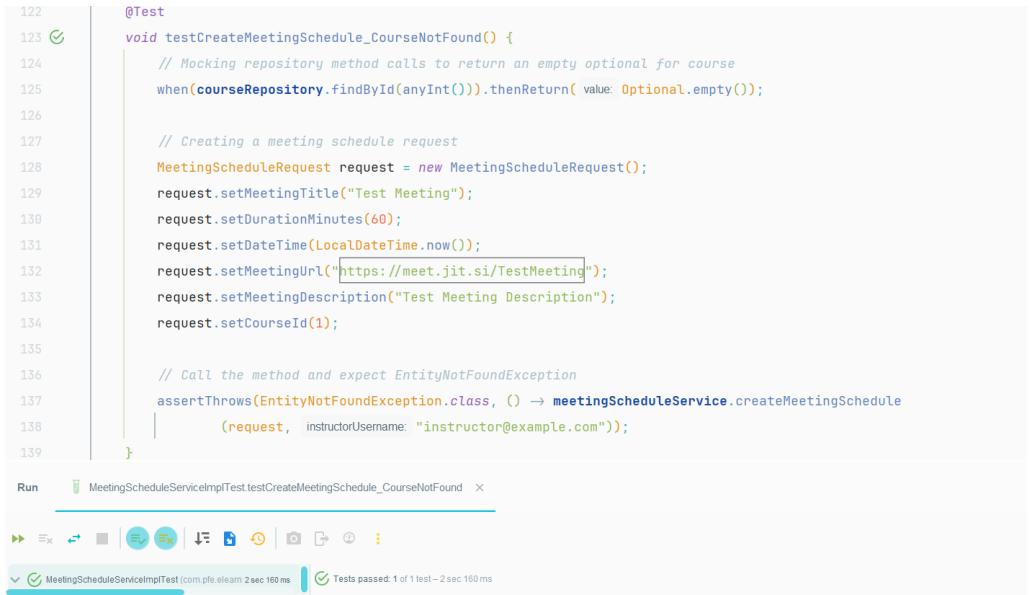
Cette interface permet à l'instructeur de lancer instantanément une réunion en saisissant le nom de la salle de réunion.

5.1.4 Tests

5.1.4.1 Les tests unitaires

Dans ce sprint, nous avons adopté la même méthode que lors des précédents pour tester les nouvelles fonctionnalités. Ces tests unitaires sont essentiels pour vérifier que les nouvelles fonctionnalités fonctionnent correctement.

— Test Unitaire de Service



```

122
123 @Test
124 void testCreateMeetingSchedule_CourseNotFound() {
125     // Mocking repository method calls to return an empty optional for course
126     when(courseRepository.findById(anyInt())).thenReturn(Optional.empty());
127
128     // Creating a meeting schedule request
129     MeetingScheduleRequest request = new MeetingScheduleRequest();
130     request.setMeetingTitle("Test Meeting");
131     request.setDurationMinutes(60);
132     request.setDateTime(LocalDateTime.now());
133     request.setMeetingUrl("https://meet.jit.si/TestMeeting");
134     request.setMeetingDescription("Test Meeting Description");
135     request.setCourseId(1);
136
137     // Call the method and expect EntityNotFoundException
138     assertThrows(EntityNotFoundException.class, () -> meetingScheduleService.createMeetingSchedule(
139         request, instructorUsername: "instructor@example.com"));
}

```

Run MeetingScheduleServiceImplTest.testCreateMeetingSchedule_CourseNotFound ×

MeetingScheduleServiceImplTest (com.pfe.eleam) 2 sec 160 ms Tests passed: 1 of 1 test – 2 sec 160 ms

FIGURE 5.13 – Test unitaire du service : création de réunion avec cours introuvable

Le test illustré dans la figure ci-dessus simule le scénario où une tentative de création de planning est effectuée avec un identifiant de cours invalide. Dans ce cas, une exception de type `CourseNotFoundException` devrait être déclenchée, indiquant que le cours spécifié n'a pas été trouvé dans la base de données. Ce test garantit que notre système gère correctement les cas où des identifiant de cours invalides sont fournis lors de la création de plannings.

— Test Unitaire de Contrôleur



```

47
48 @Test
49 void testUpdateAnswer() throws Exception {
50     // Mocking the necessary entities
51     AnswerRequest answerRequest = new AnswerRequest();
52     answerRequest.setMessageContent("Updated answer content");
53
54     AnswerResponse answerResponse = new AnswerResponse();
55     answerResponse.setMessageContent("Updated answer content");
56
57     // Mocking the service method call
58     when(answerService.updateAnswer(anyInt(), any())).thenReturn(answerResponse);
59
60     // Performing the request with MockMvc
61     mockMvc.perform(put("/api/answers/update/{answerId}", ...uriVariables: 1)
62         .contentType(MediaType.APPLICATION_JSON)
63         .content(objectMapper.writeValueAsString(answerRequest)))
64         .andExpect(status().isOk())
65         .andExpect(content().json(objectMapper.writeValueAsString(answerResponse)));
}

```

FIGURE 5.14 – Test unitaire du contrôleur : mise à jour d'une réponse

Le test illustré dans la figure ci-dessus vérifie que le contrôleur `AnswerController`

peut correctement traiter une requête de mise à jour d'une réponse. Ce test effectue une requête HTTP PUT à l'URL /api/answers/update/answerId avec des données de mise à jour et s'assure que :

- Le statut HTTP de la réponse est 200 (OK).
- Le contenu de la réponse JSON correspond aux données de mise à jour fournies.

Pour ce test, un objet AnswerRequest est créé et son contenu est défini à "Updated answer content". Un objet AnswerResponse simulé est également créé avec le même contenu. Le service AnswerService est mocké pour retourner cet AnswerResponse lorsque la méthode updateAnswer est appelée. Ce test garantit que le contrôleur traite correctement les requêtes de mise à jour et retourne les réponses appropriées.

5.1.4.2 Tests avec Postman

The screenshot shows the Postman interface with the following details:

- Header:** E-learning platform / answers management / update answer
- Method:** PUT
- URL:** http://localhost:8080/api/answers/update/7
- Body:** Raw JSON content:


```

1 {
2   "messageContent": "Hey Yaser, thanks for your message. We primarily use the FAIR framework for risk assessment, which helps us prioritize risks effectively. I'll review your document thoroughly and provide insights ASAP."
3 }
```
- Response Status:** 200 OK
- Response Time:** 72 ms
- Response Size:** 1.02 KB
- Response Content:** (JSON format)


```

1 {
2   "id": 7,
3   "messageId": 3,
4   "messageContent": "Hey Yaser, thanks for your message. We primarily use the FAIR framework for risk assessment, which helps us prioritize risks effectively. I'm curious about the risk assessment procedures we employ and would like to delve deeper into this topic. I've prepared a document on the subject for you to review.",
5   "message": "I'm curious about the risk assessment procedures we employ and would like to delve deeper into this topic. I've prepared a document on the subject for you to review.",
6   "authorId": 3,
7   "authorFullName": "Yaser Wahada",
8   "createdAt": "2024-06-09T00:07:17.789604",
9   "updatedAt": "2024-06-09T00:21:25.682718"
10 }
```

FIGURE 5.15 – Test postman : mise à jour d'une réponse à un message

Ce test Postman démontre avec succès la mise à jour d'une réponse à un message de forum via l'API. L'authentification JWT est utilisée pour la sécurité de la requête. Le statut de la réponse est vérifié pour être 200 OK, confirmant la réussite de la mise à jour. Les données retournées incluent des informations essentielles telles que l'ID de la réponse, l'ID du message associé, l'ID de l'auteur, le nom complet de l'auteur, ainsi que les dates de création et de mise à jour de la réponse.

Conclusion

En conclusion, ce troisième sprint a été une étape cruciale pour notre projet. Nous avons mis en place des forums de discussion dédiés à chaque leçon, favorisant ainsi la collaboration et l'échange. De plus, nous avons introduit une fonctionnalité de planification des réunions, ce qui a facilité la coordination et la communication au sein de l'équipe. Grâce à notre travail acharné et à notre collaboration, nous avons créé une plateforme répondant aux besoins spécifiques du projet, renforçant ainsi

la productivité et l'efficacité. Ce sprint marque la fin de notre développement, mais ouvre la voie à de nouvelles opportunités d'amélioration continue.

Conclusion générale et perspectives

Dans le cadre de ce projet de fin d'études, nous avons développé EduCare, une plateforme d'apprentissage en ligne conçue pour répondre aux besoins variés de l'éducation numérique. Notre approche agile, basée sur la méthodologie Scrum, nous a permis de structurer efficacement notre travail en trois sprints distincts, chacun apportant des améliorations significatives à la plateforme.

Le premier sprint a posé les bases de notre application en établissant les fonctionnalités utilisateur et les sujets principaux. Cette phase initiale nous a permis de définir clairement les objectifs de notre projet et de mettre en place une structure solide pour son développement ultérieur.

Le deuxième sprint a consolidé notre travail en améliorant la gestion pédagogique et la confidentialité de la plateforme. Nous avons affiné les spécifications, développé de nouveaux modules et effectué des tests rigoureux pour assurer la qualité et la sécurité de notre système.

Enfin, le troisième sprint a introduit des fonctionnalités interactives cruciales telles que le forum de discussions et la planification des réunions. Ces ajouts ont favorisé l'interaction et la collaboration entre les utilisateurs, renforçant ainsi l'expérience d'apprentissage sur la plateforme.

En résumé, chaque sprint a été essentiel pour faire progresser notre projet, en établissant les bases, en renforçant la structure et en ajoutant des fonctionnalités interactives. L'approche agile que nous avons adoptée s'est révélée être un choix judicieux, nous permettant de rester flexibles et réactifs tout au long du processus de développement.

Pour l'avenir, plusieurs perspectives d'amélioration peuvent être envisagées. Tout d'abord, la gestion des notifications pourrait être intégrée pour informer les utilisateurs des mises à jour importantes ou des nouvelles activités sur la plateforme. De plus, il serait bénéfique de donner aux gestionnaires un accès aux paramètres pour personnaliser les fonctionnalités selon les besoins spécifiques de leur organisation ou de leur cours.

En ce qui concerne les enseignants, leur permettre de paramétriser les fonctionnalités disponibles pour les apprenants pourrait être une évolution intéressante. Cela leur donnerait un plus grand contrôle sur le contenu et les activités proposées, favorisant ainsi une expérience d'enseignement plus personnalisée et adaptée.

Enfin, simplifier et résumer les fonctionnalités existantes pourrait améliorer l'accèsibilité et l'utilisabilité de la plateforme pour tous les utilisateurs, en particulier ceux qui sont moins familiers avec les technologies numériques.

En conclusion, notre expérience avec EduCare nous a permis de mieux comprendre les défis et les opportunités de l'éducation en ligne. Nous sommes enthousiastes à l'idée d'explorer ces perspectives d'amélioration et de contribuer ainsi à soutenir l'éducation numérique pour les apprenants du monde entier.

Webographie

- SHAH, D. « Coursera : Plateforme de formation en ligne ». Consulté le 27 janvier 2024. <https://www.classcentral.com/help/what-is-coursera>.
- WIKIPÉDIA. « Udemy : Plateforme de cours en ligne ». Consulté le 28 janvier 2024. <https://fr.wikipedia.org/wiki/Udemy>.
- CHATGPT. « W3Schools : Ressources de développement Web ». Accès le 28 janvier 2024. <https://www.w3schools.com/about/>.
- SIMPLILEARN. « IntelliJ : Environnement de développement intégré (IDE) ». Accès le 13 avril 2024. <https://www.simplilearn.com/what-is-intellij-idea-article>.
- WIKIPEDIA. « Angular : Framework JavaScript ». Accès le 13 avril 2024. [https://fr.wikipedia.org/wiki/Angular_\(framework\)](https://fr.wikipedia.org/wiki/Angular_(framework)).
- . « Spring Boot : Cadre de travail pour le développement d'applications Java ». Accès le 13 avril 2024. https://fr.wikipedia.org/wiki/Spring_Boot.
- . « Postman : Logiciel de développement d'API ». Accès le 14 avril 2024. [https://fr.wikipedia.org/wiki/Postman_\(logiciel\)](https://fr.wikipedia.org/wiki/Postman_(logiciel)).
- JUnit. « JUnit : Cadre d'automatisation des tests pour Java ». Accès le 14 avril 2024. <https://fr.wikipedia.org/wiki/JUnit>.
- DESKTOP, Noble. « Figma : Plateforme de conception et de prototypage collaboratif ». Accès le 15 avril 2024. <https://www.nobledesktop.com/learn/figma/what-is-figma>.

Bibliographie

- AUBRY, Claude. *L'inspiration du Scrum : Une Analogie avec le Rugby*. Dunod, 2010.
- ROQUES, Pascal et Franck VALLÉ. *Le Concept d'Acteur dans le Contexte de l'Interaction Système-Utilisateur*. Paris : Eyrolles, 2007.