

Table des matières

1	Premier Sprint : Mise en Place des Fonctions Utilisateur et Sujets	2
	Introduction.....	3
1.1	Le premier sprint	3
1.1.1	Spécification fonctionnelle.....	15
1.1.1.1	Raffinement de cas d'utilisation du sprint 1.....	15
1.1.1.2	Diagramme de cas d'utilisation détaillé du sprint1 ...	15
1.1.1.3	Description textuelle de cas d'utilisation	17
1.1.2	La modélisation conceptuelle	24
1.1.2.1	Diagramme de classes.....	24
1.1.2.2	Diagramme de séquence détaillé	25
1.1.3	Réalisation	30
1.1.4	Tests	34
1.1.4.1	Tests unitaires	34
1.1.4.2	Tests avec Postman	35
	Conclusion	36

Table des figures

1.1	Diagramme de cas d'utilisation détaillé du premier sprint	16
1.2	Diagramme de classes du premier sprint	25
1.3	Diagramme de séquence du cas d'utilisation « S'authentifier »	26
1.4	Diagramme de séquence du cas d'utilisation « Ajouter un membre » .	27
1.5	Diagramme de séquence du cas d'utilisation « Consulter profil »	28
1.6	Diagramme de séquence du cas d'utilisation « Consulter statistiques »	29
1.7	Interface d'authentification.....	30
1.8	Interface de gestion des membres	31
1.9	Interface de création d'un nouveau membre	31
1.10	Interface de confirmation de nouveau compte	32
1.11	Interface du tableau de bord du gestionnaire	32
1.12	Interface de gestion du profil du gestionnaire.....	33
1.13	Interface d'ajout d'un nouveau sujet	33
1.14	Interface de gestion des sujets	34
1.15	Test unitaire du service : création de sujet.....	35
1.16	Test unitaire du contrôleur : récupération des derniers sujets	35
1.17	Test Postman : création d'un sujet	36

Liste des tableaux

1.1	Backlog Du Premier Sprint	4
1.2	Description textuelle du cas d'utilisation « S'authentifier »	17
1.3	Description textuelle du cas d'utilisation « Ajouter Membre ».....	18
1.4	Description textuelle du cas d'utilisation « Modifier informations de profil »	20
1.5	Description textuelle du cas d'utilisation « Télécharger photo de profil »	21
1.6	Description textuelle du cas d'utilisation « Ajouter sujet »	22
1.7	Description textuelle du cas d'utilisation « Modifier sujet »	22
1.8	Description textuelle du cas d'utilisation « Supprimer sujet »	23

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 œil 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 AYOUB

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 plateforme 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

Premier Sprint : Mise en Place des Fonctions Utilisateur et Sujets

Plan

Introduction.....	3
1.1 Le premier sprint	3
1.1.1 Spécification fonctionnelle	15
1.1.1.1 Raffinement de cas d'utilisation du sprint 1	15
1.1.1.2 Diagramme de cas d'utilisation détaillé du sprint1	15
1.1.1.3 Description textuelle de cas d'utilisation.....	17
1.1.2 La modélisation conceptuelle.....	24
1.1.2.1 Diagramme de classes	24
1.1.2.2 Diagramme de séquence détaillé.....	25
1.1.3 Réalisation.....	30
1.1.4 Tests	34
1.1.4.1 Tests unitaires	34
1.1.4.2 Tests avec Postman.....	35
Conclusion	36

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.

1.1 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 1.1 – 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	En tant que membre, je veux pouvoir modifier mon profil utilisateur.	<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).- 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	En tant que gestionnaire, je veux pouvoir supprimer un compte d'instructeur ou de candidat existant depuis la liste des utilisateurs.	<ul style="list-style-type: none">- Créer un endpoint API dédié pour récupérer la liste des comptes d'instructeurs et de candidats existants. (Backend)- Créer une logique de traitement côté backend pour la suppression d'un compte utilisateur existant. (Backend)- Gérer les erreurs lors de la récupération de la liste des comptes utilisateurs ou lors des opérations de 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 filtrage 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	En tant que vis-iteur, je veux pouvoir Contacter gestionnaire par email	<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	En tant que membre, je veux recevoir un email de bienvenue contenant les informations de connexion requises dès que le gestionnaire crée mon compte	<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.

1.1.1 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.

1.1.1.1 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**

1.1.1.2 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 :

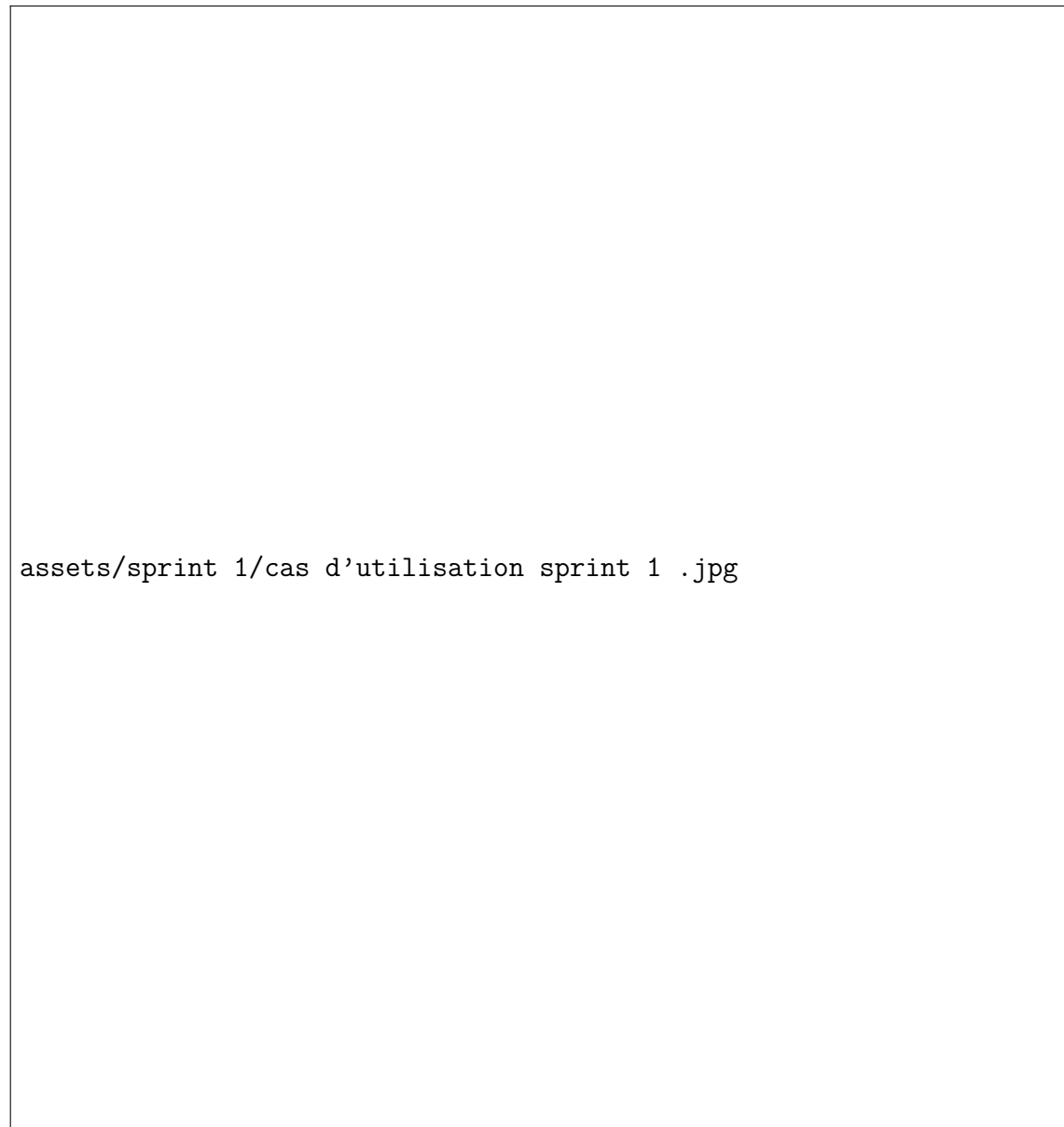


FIGURE 1.1 – Diagramme de cas d'utilisation détaillé du premier sprint

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.

1.1.1.3 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 1.2 – 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

	<p>2. Le système affiche le formulaire d'authentification</p> <p>3. Le membre saisit son adresse e-mail et son mot de passe.</p> <p>4. Le membre soumet le formulaire.</p> <p>5. L'application valide les informations d'identification.</p> <p>6. Si les informations sont valides, un JWT est généré, contenant les détails de membre.</p> <p>7. Le JWT est stocké localement et envoyé dans l'en-tête des requêtes pour prouver l'identité.</p> <p>8. À chaque demande, l'application vérifie la validité du JWT et autorise l'accès à l'accueil personnel de membre.</p> <p>9. L'accueil personnel de membre s'affiche, présentant ses fonctionnalités et informations spécifiques.</p>
Scénario d'exception	<p>SE1. Champs vides. SE1 démarre au point 3 du scénario nominal.</p> <p>4. Le système affiche un message d'erreur si des champs requis sont laissés vides.</p> <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> <p>6. L'application indique une erreur si les informations d'identification fournies sont incorrectes.</p> <p>7. Le membre est invité à corriger ses informations pour réessayer.</p> <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 1.3 – 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	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 1.4 – 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 1.5 – 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 1.6 – 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 1.7 – 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</p> <p>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 1.8 – 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.

	<ul 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 SE1 commence après l'étape 5 du scénario nominal.</p> <ul 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.

1.1.2 La modélisation conceptuelle

1.1.2.1 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 1.2 – 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

1.1.2.2 Diagramme de séquence détaillé

Le diagramme de séquence modélise les interactions entre les objets d'un système, illustrant la séquence chronologique des messages échangés. Cela clarifie le flux de contrôle et de communication. Les acteurs principaux sont positionnés à gauche, les secondaires à droite. L'objectif est de décrire les actions entre les acteurs ou 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".

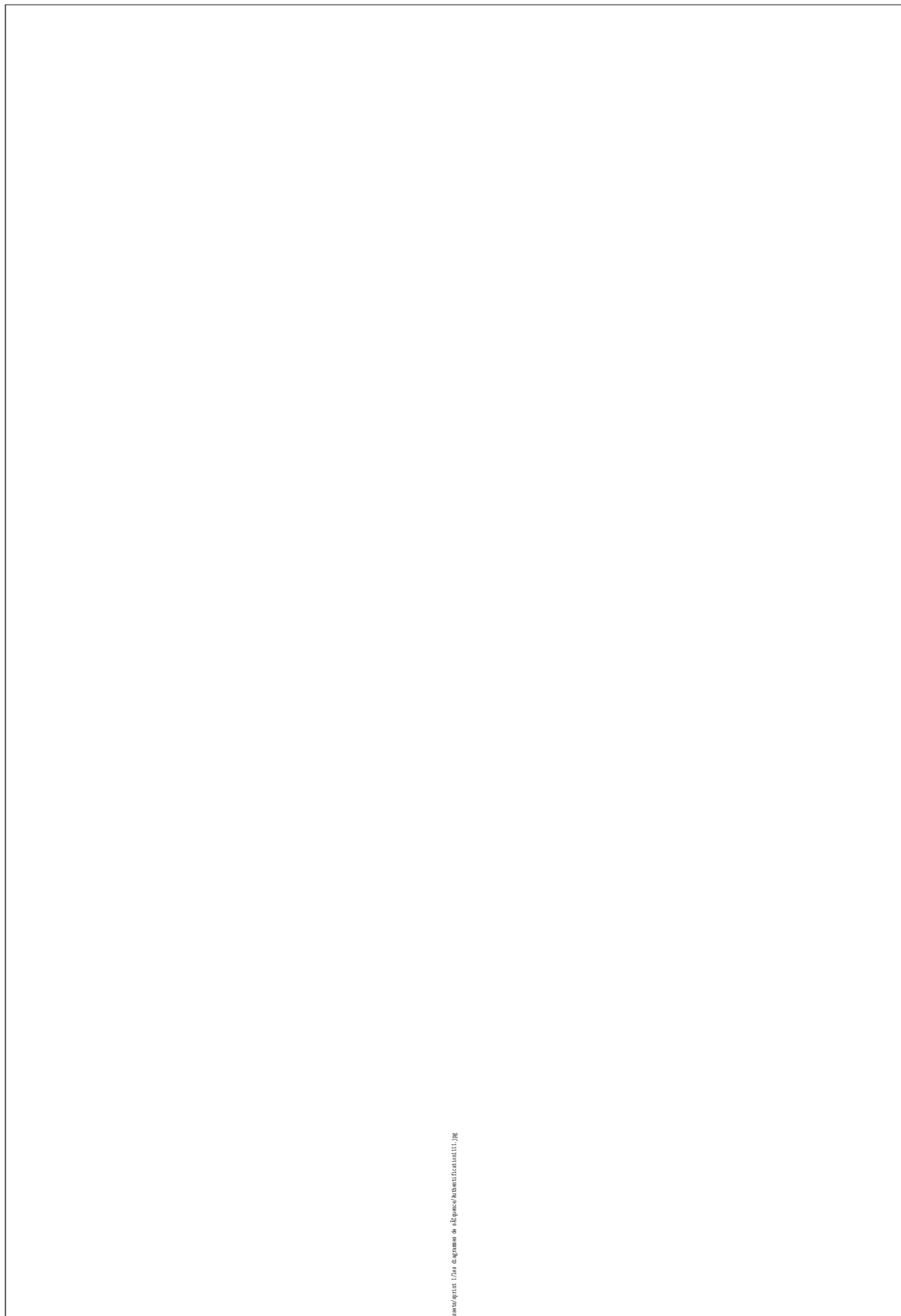


FIGURE 1.3 – Diagramme de séquence du cas d'utilisation « S'authentifier »

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 »

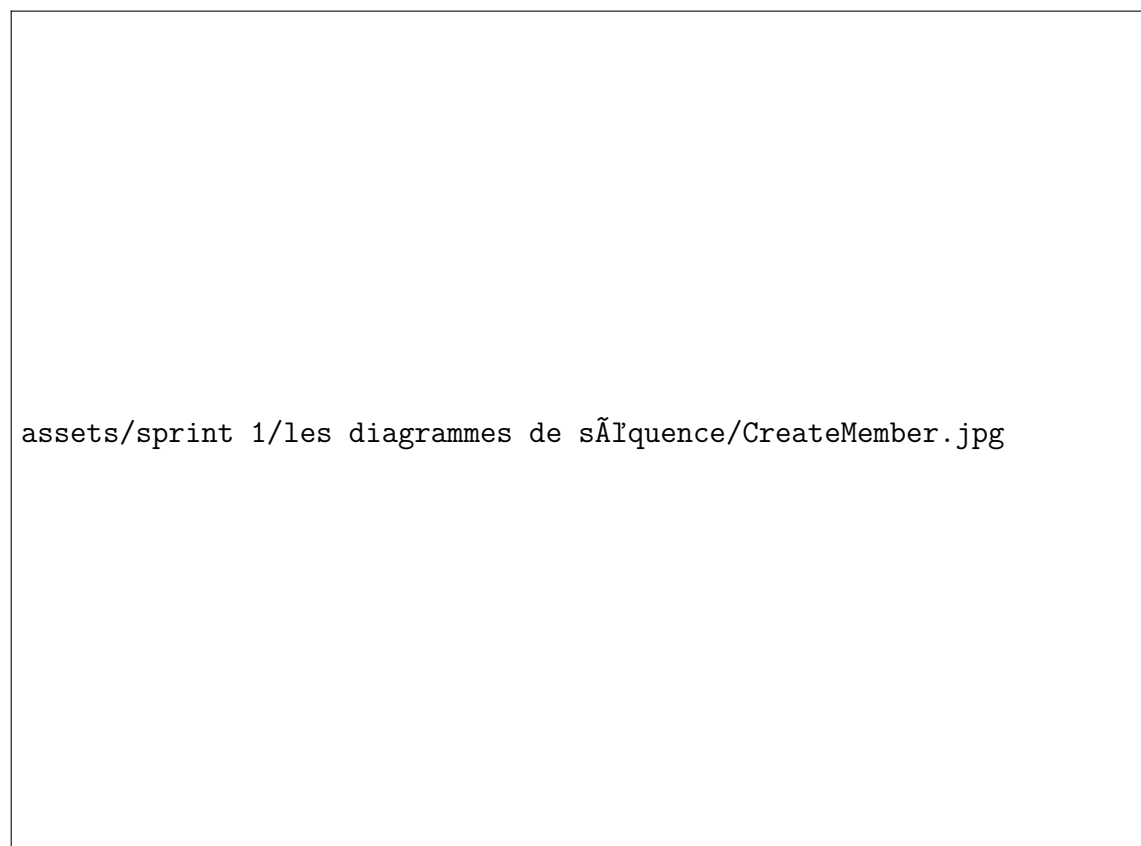


FIGURE 1.4 – Diagramme de séquence du cas d'utilisation « Ajouter un membre »

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.

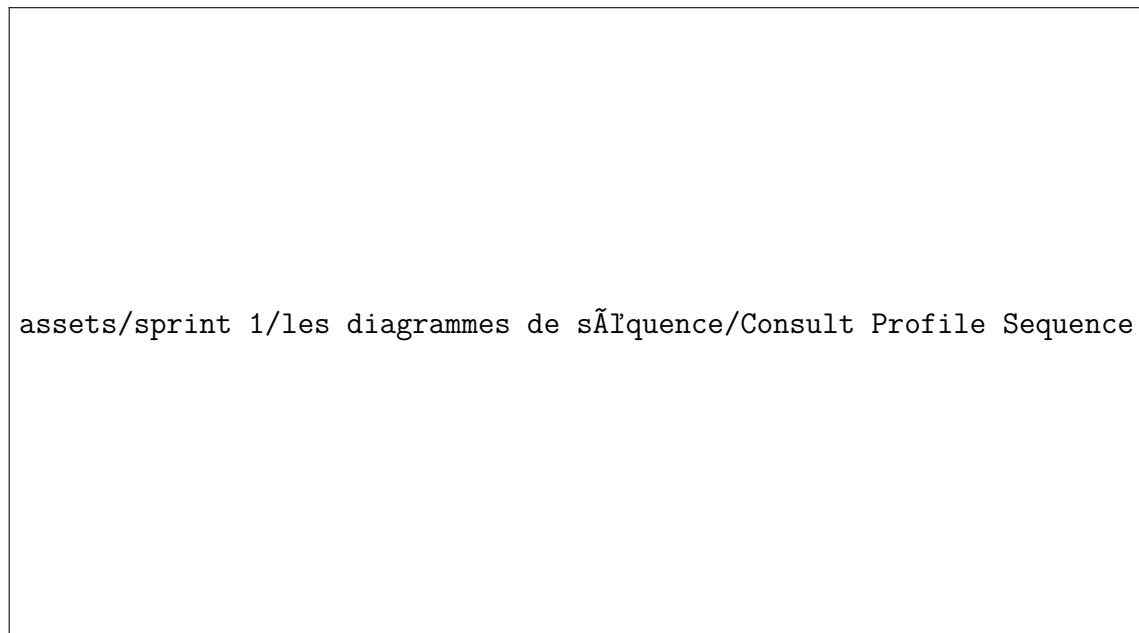


FIGURE 1.5 – Diagramme de séquence du cas d'utilisation « Consulter profil »

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. Le membre peut alors modifier sa photo de profil ou ses informations personnelles.

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

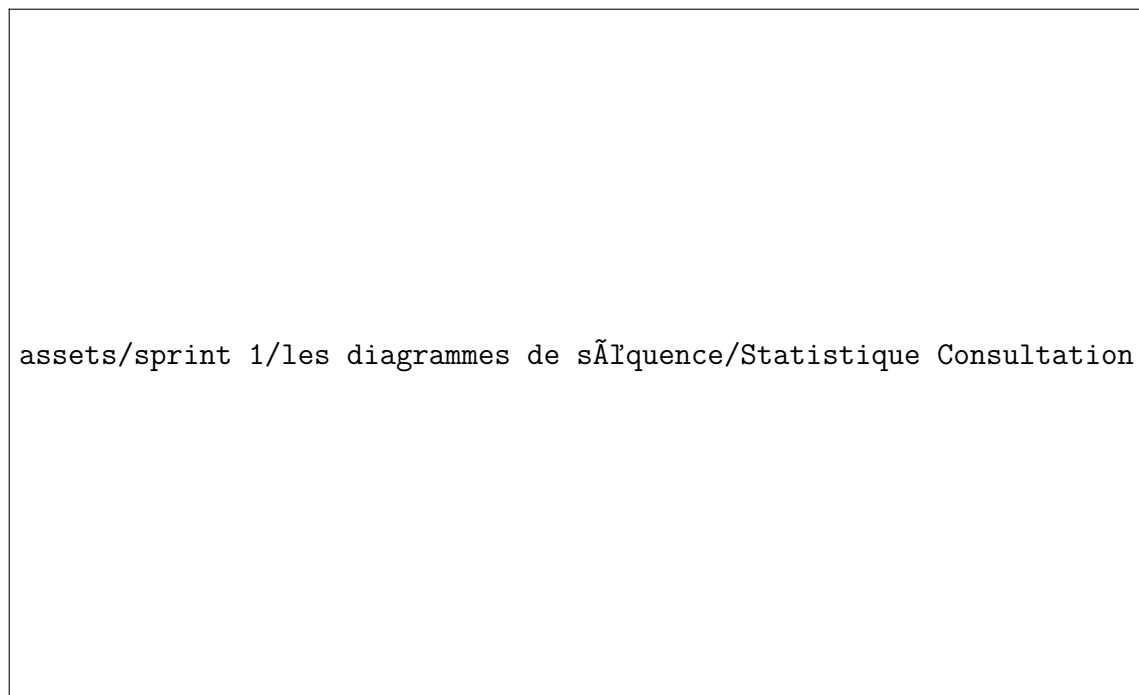


FIGURE 1.6 – Diagramme de séquence du cas d'utilisation « Consulter statistiques »

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.

1.1.3 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.

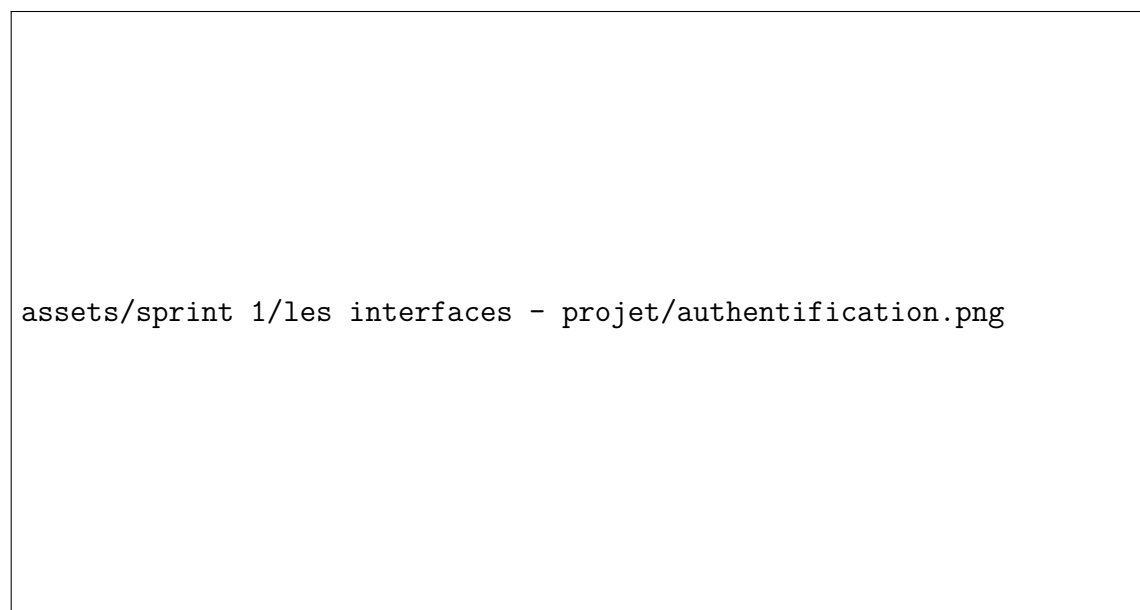


FIGURE 1.7 – Interface d'authentification

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

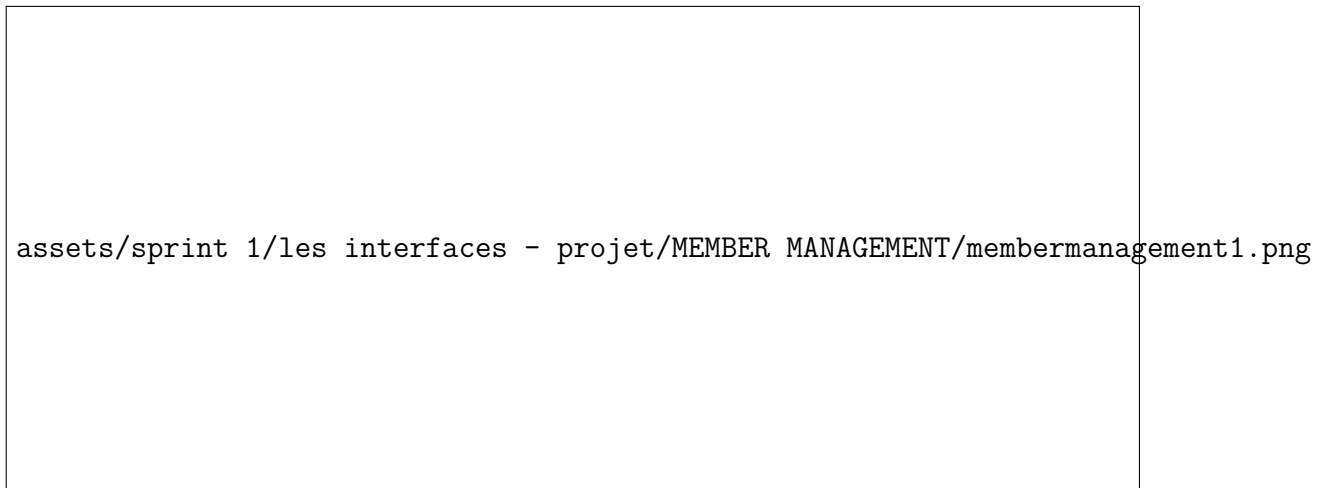


FIGURE 1.8 – Interface de gestion des membres

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 1.9 – Interface de création d'un nouveau membre

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



FIGURE 1.10 – 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

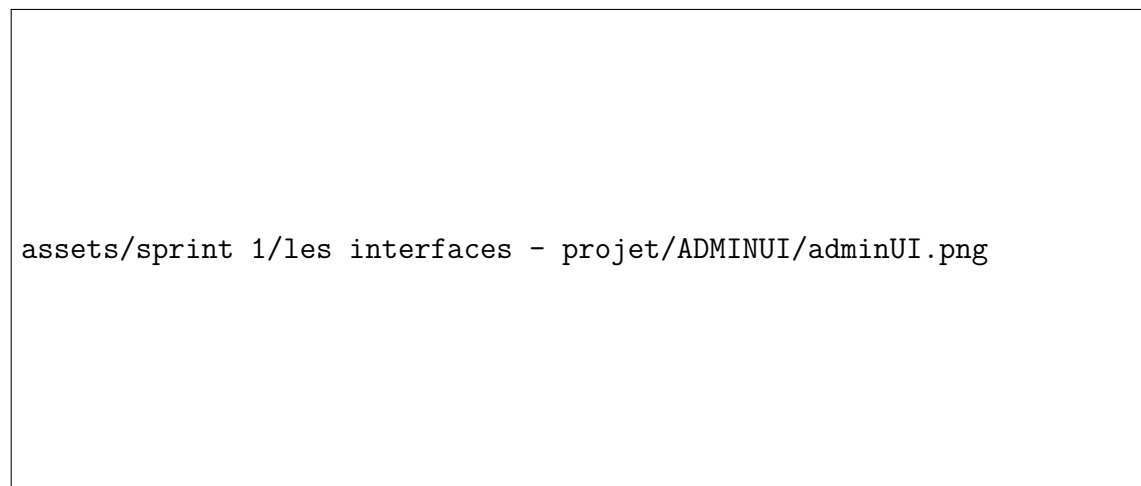


FIGURE 1.11 – Interface du tableau de bord du gestionnaire

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.

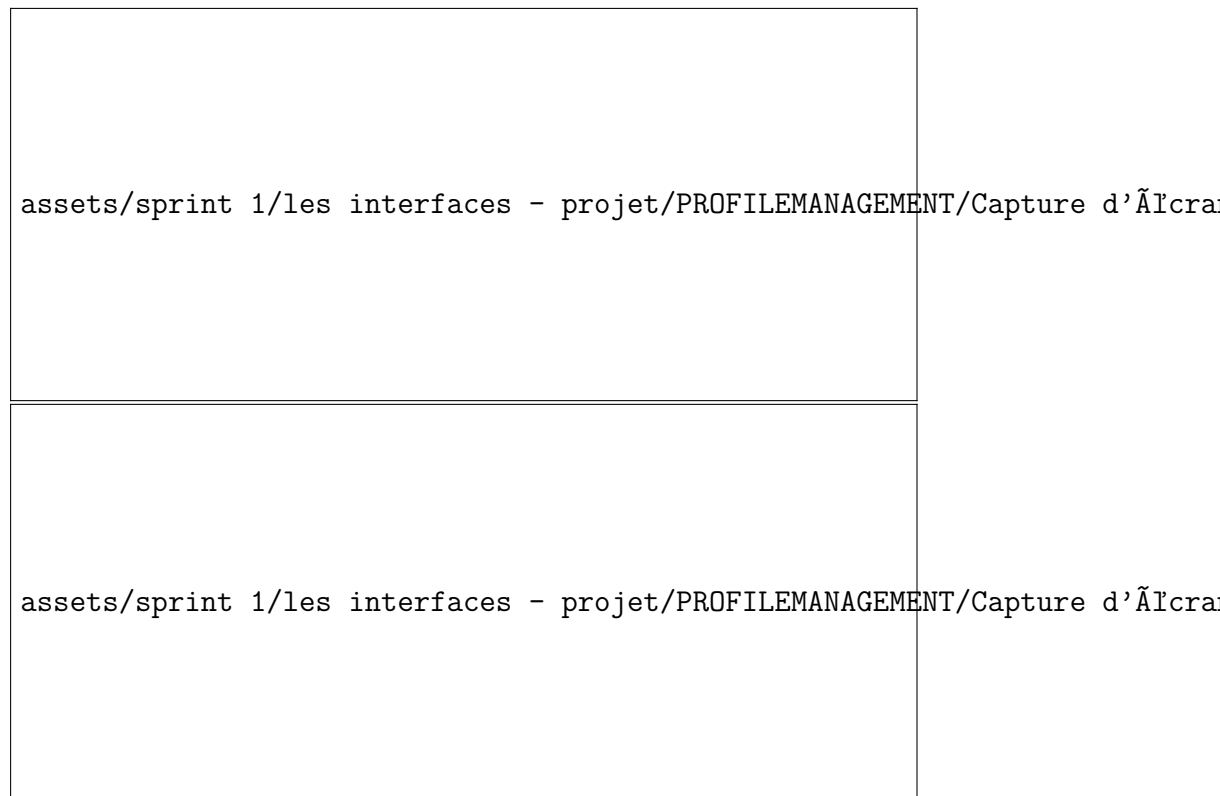


FIGURE 1.12 – Interface de gestion du profil du gestionnaire

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 1.13 – Interface d'ajout d'un nouveau sujet

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

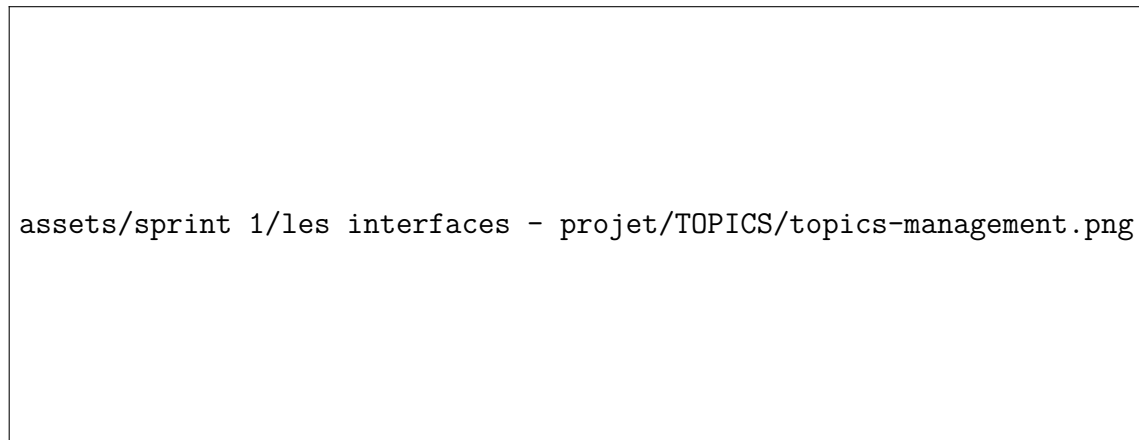


FIGURE 1.14 – Interface de gestion des sujets

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.

1.1.4 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.

1.1.4.1 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

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

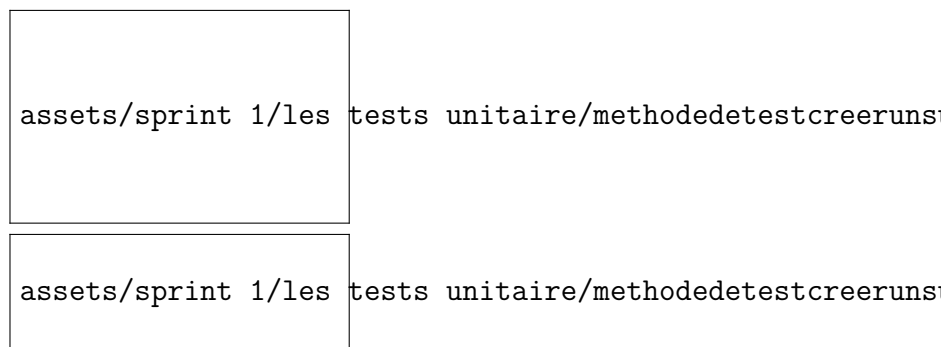


FIGURE 1.15 – Test unitaire du service : création de sujet

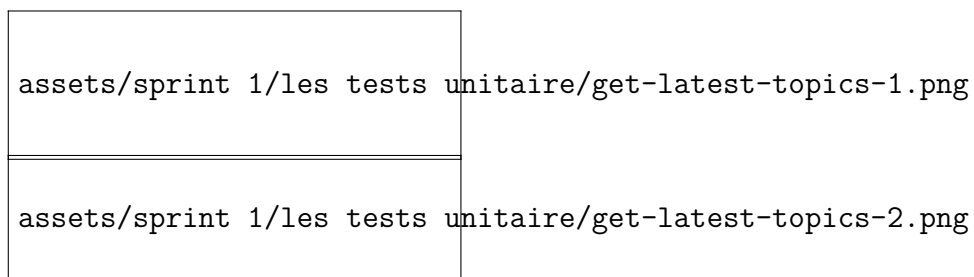


FIGURE 1.16 – Test unitaire du contrôleur : récupération des derniers sujets

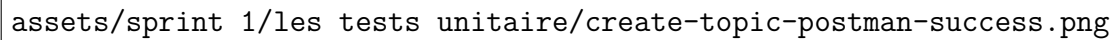
Ce test unitaire évalue le comportement de l'endpoint latest-topics du contrôleur TopicController. À l'aide de Mockito, il simule la réponse du service TopicService pour une demande de sujets les plus récents. Les données simulées comprennent une liste de deux TopicResponse. En simulant une requête GET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

1.1.4.2 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.

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.



```
assets/sprint 1/les tests unitaire/create-topic-postman-success.png
```

FIGURE 1.17 – Test Postman : création d’un sujet

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.

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 paramétrer 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'accessibilité 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.

