



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**
Membre de **HONORIS UNITED UNIVERSITIES**

Projet de Fin d'Année

Application Web de Réseau Social EMSI

Réalisé par : **Mouhcine Latiris & Iliass Loutfi**

Encadré par : **Imane Daoudi**

Année universitaire : 2024 - 2025

Remerciements

Nous tenons tout d'abord à exprimer notre plus profonde gratitude à notre encadrante de Projet de Fin d'Année, Madame Imane Daoudi, pour son accompagnement exceptionnel tout au long de ce travail. Sa disponibilité constante, son écoute attentive et ses conseils éclairés ont été d'une aide précieuse à chaque étape du projet. Elle a su nous guider avec rigueur et bienveillance, nous poussant à dépasser nos limites tout en respectant nos idées. Son expertise en développement web et en encadrement de projets nous a permis de mener cette expérience à bien dans des conditions optimales.

Nous remercions également notre professeur de Python, dont les cours clairs et structurés ont posé les bases de notre logique de programmation et ont grandement contribué à la réussite technique de notre projet. Ses explications détaillées et sa pédagogie active nous ont permis de développer des compétences solides en algorithmique et en structuration du code, compétences que nous avons appliquées directement dans le développement de notre plateforme.

Nos remerciements s'adressent aussi à notre enseignant de recherche scientifique, qui nous a initiés à une démarche méthodologique rigoureuse. Grâce à lui, nous avons su adopter une approche scientifique dans la conception et l'analyse de notre travail, en nous basant sur des recherches documentées, des comparaisons de solutions existantes et une structuration cohérente de nos idées.

Nous tenons également à remercier chaleureusement l'ensemble du corps enseignant de l'École Marocaine des Sciences de l'Ingénieur (EMSI), pour la qualité de l'enseignement reçu tout au long de notre formation. EMSI a été bien plus qu'un établissement académique pour nous ; elle a été un véritable cadre d'épanouissement personnel et professionnel, un lieu d'apprentissage, d'innovation, et de partage. Le programme pluridisciplinaire proposé par l'école, l'encadrement pédagogique attentif, ainsi que les opportunités de projets concrets ont été des éléments clés dans notre développement en tant que futurs ingénieurs.

Enfin, nous remercions nos camarades de promotion pour leur soutien moral, leurs échanges enrichissants, et leur aide précieuse au fil des semaines. C'est grâce à cette solidarité et cette dynamique collective que nous avons pu relever les défis de ce projet avec détermination et enthousiasme.

Introduction Générale

Dans un monde de plus en plus connecté, les établissements d'enseignement supérieur doivent s'adapter aux nouveaux modes de communication et de collaboration. L'intégration du numérique dans la vie académique est devenue essentielle pour renforcer l'interaction entre les différents acteurs : étudiants, enseignants et personnel administratif. C'est dans cette optique que le présent projet a vu le jour : concevoir et développer une application web de réseau social interne dédiée à l'EMSI.

Ce réseau social vise à centraliser les échanges, améliorer la diffusion des informations, et favoriser une communauté académique active et engagée. Il s'inscrit dans une démarche pédagogique moderne, en proposant des outils adaptés au contexte éducatif tout en garantissant une accessibilité, une sécurité et une ergonomie optimales.

Ce rapport retrace les différentes étapes de réalisation de ce projet, depuis l'analyse des besoins jusqu'aux tests finaux, en passant par la modélisation UML, le choix des technologies, et le développement de l'application.

Résumé

Ce document présente le développement d'une application web de réseau social interne pour l'EMSI, conçue dans le cadre du Projet de Fin d'Année. L'objectif principal est de proposer une plateforme centralisée permettant aux étudiants, enseignants et administrateurs de communiquer, partager des ressources, et interagir de manière sécurisée et fluide.

Le rapport commence par une analyse du contexte et des besoins, suivie d'une modélisation fonctionnelle à l'aide de diagrammes UML. La solution technique repose sur une architecture MVC, avec un frontend développé en ReactJS et un backend en Django REST Framework, connecté à une base de données MySQL.

La phase de test a validé la robustesse et la pertinence de la solution. En conclusion, ce projet démontre la capacité à mener un développement logiciel complet en équipe, selon une approche agile, tout en répondant à un besoin réel du milieu académique.

Table des matières

Remerciements	1
Introduction Générale	2
Résumé	3
1 Contexte général du projet	5
1.1 Introduction	5
1.2 Contexte du projet	5
1.3 Périmètre du projet	6
1.4 Problématique	7
1.5 Objectifs du projet	7
1.6 Benchmark	8
1.7 Méthodologie de gestion de projet	9
1.8 Diagramme de Gantt	10
1.9 Conclusion	10
2 Analyse et Conception	11
2.1 Introduction	11
2.2 Besoins fonctionnels détaillés	11
2.3 Besoins non fonctionnels	13
2.4 Diagramme de cas d'utilisation	15
2.5 Diagramme de classes	16
2.6 Conclusion	16
3 Réalisation et Tests	17
3.1 Introduction	17
3.2 Architecture de la solution	17
3.3 Technologies utilisées	18
3.4 Interfaces principales développées	19
3.5 Méthodologie de tests	19
3.6 Résultats des tests	19
3.7 Conclusion	20
4 Conclusion et Perspectives	21
Bibliographie	23

Chapitre 1

Contexte général du projet

1.1 Introduction

L'avènement des technologies numériques a profondément transformé les modes d'interaction entre individus, bouleversant ainsi les dynamiques de communication au sein des organisations de manière irréversible. Le secteur de l'enseignement supérieur, loin d'être une exception, se trouve au cœur de cette mutation digitale. Les établissements cherchent désormais activement à intégrer et à exploiter judicieusement ces technologies pour non seulement améliorer l'expérience globale des étudiants, mais aussi pour rehausser la qualité pédagogique dispensée et optimiser l'efficacité de la gestion administrative quotidienne. C'est dans ce contexte dynamique et évolutif que l'École Marocaine des Sciences de l'Ingénieur (EMSI) manifeste sa volonté d'initier une modernisation significative de ses outils de communication interne. Cette ambition se traduit par la proposition de développer une plateforme numérique centralisée et performante, spécifiquement dédiée aux besoins de communication et de collaboration de ses étudiants, de son corps enseignant et de son personnel administratif.

Le développement d'un réseau social académique interne ne constitue pas simplement une mise à niveau technologique, mais représente une initiative stratégique et innovante. L'objectif principal est de créer un écosystème numérique unifié visant à centraliser l'ensemble des échanges d'informations pertinentes, qu'elles soient académiques, administratives ou associatives. En parallèle, cette plateforme ambitionne d'encourager activement la collaboration interdisciplinaire entre étudiants de différentes filières et niveaux, et de fluidifier considérablement la circulation de l'information essentielle au bon fonctionnement de l'établissement. En s'efforçant de répondre de manière ciblée aux besoins spécifiques inhérents à l'environnement éducatif de l'EMSI – tels que le partage de supports de cours, les annonces officielles, les discussions thématiques ou la coordination de projets de groupe – ce projet aspire à renforcer substantiellement la cohésion de la communauté éducative. Il vise également à intégrer les meilleures pratiques observées dans le domaine du web collaboratif, tout en les adaptant au contexte particulier d'une école d'ingénieurs.

1.2 Contexte du projet

Actuellement, la communication au sein de la communauté EMSI repose en grande partie sur une mosaïque d'outils de communication grand public, tels que les applications de messagerie instantanée comme WhatsApp ou Telegram, ainsi que les traditionnels

courriers électroniques. Bien que ces outils offrent une certaine praticité pour des échanges ponctuels, leur utilisation présente des inconvénients majeurs dans un cadre institutionnel. Ils ne sont ni centralisés, ni spécifiquement structurés pour les besoins académiques, ce qui engendre inévitablement une dispersion significative de l'information importante. Cette fragmentation conduit fréquemment à des pertes de suivi dans les conversations, à une difficulté accrue pour retrouver des informations antérieures, et à une faible visibilité globale des contenus à caractère pédagogique ou administratif essentiels. De surcroît, l'absence criante d'une plateforme unique et dédiée limite considérablement la possibilité de construire un environnement numérique de travail et d'apprentissage véritablement intégré, cohérent et efficace pour l'ensemble de la communauté.

Face à ce constat et aux limites inhérentes aux solutions actuelles, il est apparu clairement nécessaire et stratégique d'entreprendre le développement d'une solution interne, entièrement maîtrisée et propre à l'EMSI. Cette solution sur mesure devra impérativement répondre aux exigences spécifiques de l'institution en termes de sécurité, de confidentialité et de modération, tout en étant conçue pour favoriser activement une dynamique collaborative riche et constructive entre les différentes composantes de l'école : les étudiants, les enseignants, les différents clubs et associations étudiantes, ainsi que les services administratifs. Ce projet s'inscrit par conséquent dans une démarche globale et ambitieuse de transformation numérique de l'EMSI, une transformation qui dépasse le simple aspect technologique pour toucher également à la culture organisationnelle et aux méthodes de travail collaboratif.

1.3 Périmètre du projet

Le périmètre défini pour ce projet est exhaustif et couvre l'intégralité du cycle de développement logiciel, depuis la phase initiale d'analyse approfondie des besoins jusqu'à la mise en ligne effective d'une plateforme web fonctionnelle, responsive (adaptée à tous les écrans) et hautement sécurisée. L'application développée doit garantir une accessibilité optimale via l'ensemble des navigateurs web modernes et offrir une expérience utilisateur cohérente sur différents types d'appareils (ordinateurs, tablettes, smartphones). Le périmètre fonctionnel inclut notamment, mais sans s'y limiter, les capacités suivantes :

- **Système d'authentification robuste et sécurisé** : Intégrant une gestion fine des rôles (étudiants, enseignants, administrateurs), ce système permettra une personnalisation poussée de l'expérience utilisateur en fonction du profil et assurera un contrôle d'accès rigoureux aux différentes fonctionnalités et données de la plateforme, potentiellement couplé à une validation via l'email institutionnel.
- **Interface de publication intuitive et polyvalente** : Les utilisateurs disposeront d'une interface conviviale pour partager divers types de contenus : textes, images, liens, documents pédagogiques (PDF, présentations, etc.), et autres supports multimédias pertinents pour le contexte académique.
- **Module d'interaction sociale enrichi** : Pour favoriser l'engagement et le dialogue, la plateforme intégrera un module interactif permettant les commentaires sous les publications, les mentions d'autres utilisateurs (@pseudo), les réactions rapides (likes, emojis), et potentiellement le partage de publications au sein de la plateforme, afin de promouvoir activement l'échange d'idées et l'interactivité constructive entre les membres.
- **Gestion avancée des groupes** : La plateforme offrira la possibilité de créer et de gérer des groupes de discussion, qui pourront être configurés comme privés (accès

sur invitation ou validation) ou publics (ouverts à tous les membres). Ces groupes pourront être dédiés à des matières spécifiques, des classes, des projets de groupe, des clubs étudiants ou d'autres centres d'intérêt, offrant ainsi des espaces ciblés et organisés pour la discussion et la collaboration.

- **Tableau de bord administratif complet** : Un back-office dédié aux administrateurs permettra une gestion centralisée de la plateforme, incluant la modération proactive et réactive des contenus publiés, la gestion des comptes utilisateurs (création, modification, suppression, gestion des rôles), le suivi et le traitement efficace des signalements de contenus inappropriés, ainsi que l'accès à des statistiques d'utilisation pour analyser l'engagement et l'activité sur la plateforme.
- **Système de notifications intelligent et configurable** : Les utilisateurs seront informés en temps réel (via notifications push si une application mobile est envisagée, ou a minima par des notifications internes à la plateforme et/ou par e-mail) des événements importants les concernant : nouvelles publications dans leurs groupes, réponses à leurs commentaires, mentions, demandes d'adhésion à un groupe, annonces importantes, etc. Les utilisateurs pourront potentiellement configurer leurs préférences de notification.

1.4 Problématique

Dans le contexte spécifique d'un environnement académique tel que l'EMSI, où la qualité et la fluidité des échanges d'informations, ainsi que la capacité à collaborer efficacement, jouent un rôle absolument crucial dans le succès du parcours de formation des étudiants et dans l'efficacité pédagogique, l'absence actuelle d'un espace numérique unifié et dédié constitue un frein notable. Cette lacune peut nuire à la continuité pédagogique entre les cours, complexifier la diffusion d'informations essentielles et potentiellement affaiblir la cohésion institutionnelle et le sentiment d'appartenance. Dès lors, la question fondamentale et centrale que ce projet cherche à adresser peut être formulée comme suit :

Comment concevoir, développer et déployer une plateforme sociale numérique qui soit non seulement techniquement robuste et fonctionnelle, mais surtout parfaitement adaptée aux besoins spécifiques et aux usages particuliers d'un établissement d'enseignement supérieur comme l'EMSI ? Comment garantir simultanément une expérience utilisateur exceptionnellement fluide et intuitive pour tous les profils, une sécurité renforcée et sans compromis des données et des échanges, tout en apportant une réelle et mesurable valeur ajoutée sur le plan pédagogique et collaboratif ?

1.5 Objectifs du projet

Les objectifs spécifiques fixés pour ce projet s'inscrivent résolument dans une triple logique : celle de l'innovation pédagogique par l'intégration du numérique, celle de l'amélioration tangible des services offerts à la communauté EMSI, et enfin celle de la création et du renforcement d'une identité numérique propre et positive pour l'école. De manière plus concrète et opérationnelle, le projet vise à atteindre les buts suivants :

- **Concevoir une interface utilisateur exemplaire** : L'accent sera mis sur l'ergonomie, l'intuitivité et l'accessibilité. L'interface devra être facile à prendre en main et agréable à utiliser pour tous les profils d'utilisateurs, qu'ils soient des natifs du

numérique ou moins familiarisés avec ce type d'outils collaboratifs, en respectant les standards d'accessibilité web (WCAG).

- **Offrir un cadre numérique sécurisé et fiable** : La plateforme devra garantir un haut niveau de sécurité, en respectant scrupuleusement les bonnes pratiques en matière de protection des données personnelles (conformité RGPD si applicable) et de confidentialité des échanges. Cela implique une gestion rigoureuse des permissions d'accès, basée sur les rôles définis (étudiant, enseignant, administrateur), et la mise en œuvre de mesures de sécurité techniques appropriées (HTTPS, protection contre les attaques courantes, etc.).
- **Centraliser et structurer les échanges académiques** : La plateforme doit devenir le point névralgique pour le partage de supports de cours, la diffusion des annonces pédagogiques importantes, l'organisation de discussions thématiques liées aux enseignements ou aux projets, le tout dans un espace unique, bien organisé et facilement navigable pour retrouver l'information recherchée.
- **Dynamiser la communauté et renforcer l'appartenance** : Au-delà de l'aspect purement fonctionnel, la plateforme doit encourager activement l'interaction, la participation volontaire et le développement d'un fort sentiment d'appartenance à la communauté EMSI. Ceci sera réalisé à travers une plateforme conviviale, dotée de fonctionnalités sociales engageantes, conçue pour être évolutive afin d'intégrer de futurs besoins, et profondément ancrée dans la culture et les valeurs de l'école.

1.6 Benchmark

Afin de nourrir notre réflexion et de concevoir une solution à la fois innovante et efficace, une phase d'analyse comparative (benchmark) de plusieurs plateformes collaboratives et sociales existantes a été menée. L'objectif n'était pas de copier ces plateformes, mais plutôt d'identifier les fonctionnalités les plus pertinentes, les mécanismes d'interaction ayant fait leurs preuves, et les bonnes pratiques en termes d'expérience utilisateur, afin de les adapter judicieusement au contexte spécifique de notre projet. Les plateformes étudiées incluent notamment :

- **Facebook** : Analysé principalement pour ses mécanismes d'interaction sociale très populaires et intuitifs (système de likes, partages viraux, commentaires imbriqués), sa fonctionnalité de création et de gestion de groupes thématiques très utilisée, et l'expérience utilisateur globalement fluide et engageante qu'il propose, malgré ses controverses.
- **Slack** : Étudié pour sa structure organisationnelle basée sur des canaux de discussion thématiques, qui s'avère idéale pour segmenter les conversations par matière, par classe, par projet ou par club. Son intégration transparente des notifications en temps réel et sa capacité à s'interfacer avec des outils externes ont également retenu notre attention.
- **Microsoft Teams** : Considéré comme une référence dans le milieu éducatif et professionnel, Teams propose un environnement de travail collaboratif très structuré, intégrant nativement des fonctionnalités avancées telles que la visioconférence, le partage et la co-édition de fichiers, la gestion des devoirs et des notes, offrant un modèle intéressant pour un usage académique intensif.

Il est crucial de souligner que notre objectif final n'est en aucun cas de répliquer ces plateformes commerciales complexes. Il s'agit plutôt de s'en inspirer sélectivement, en extrayant les éléments conceptuels et fonctionnels les plus adaptés aux besoins spéci-

fiques et au contexte culturel de l’EMSI. Notre solution se distinguera par une interface volontairement simplifiée, épurée de toute publicité intrusive, et mettra un accent particulièrement fort sur les fonctionnalités à valeur ajoutée pédagogique et sur les mécanismes de modération institutionnelle nécessaires pour garantir un environnement d’échange sûr et respectueux.

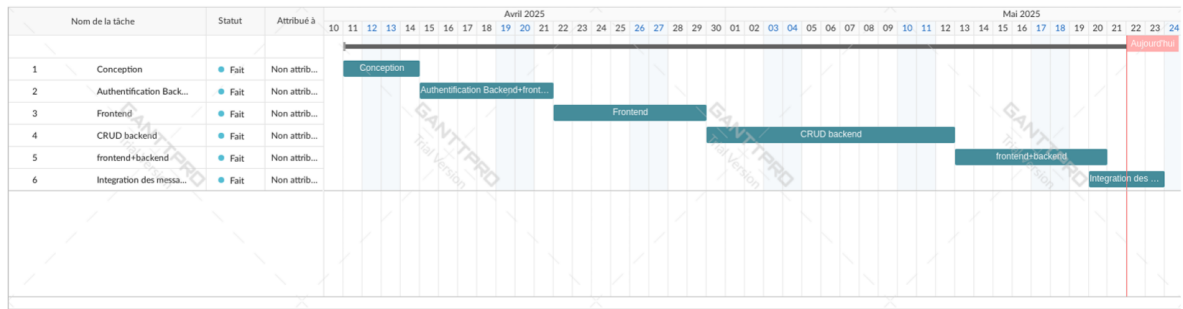
1.7 Méthodologie de gestion de projet

Pour mener à bien ce projet dans les délais impartis tout en garantissant une flexibilité suffisante pour s’adapter aux découvertes et aux retours en cours de route, nous avons opté pour l’adoption d’une méthodologie de gestion de projet Agile. Plus précisément, nous avons choisi de nous appuyer sur le cadre méthodologique **Scrum**, largement reconnu dans l’industrie logicielle pour sa flexibilité, son approche itérative et incrémentale, et sa capacité éprouvée à intégrer les retours des parties prenantes de manière continue tout au long du développement. Cette approche nous a permis de structurer notre travail en cycles de développement courts et réguliers, appelés **sprints** (généralement d’une durée d’une à deux semaines), ponctués par des rituels et des artefacts spécifiques favorisant la collaboration et la transparence. Les éléments clés de notre mise en œuvre de Scrum incluent :

- **Backlog de produit et planification de sprint** : Une définition claire et priorisée des fonctionnalités et des tâches à accomplir (le backlog) a été établie et affinée au début de chaque sprint lors de la réunion de planification, permettant à l’équipe de s’engager sur un objectif réalisable pour le cycle à venir.
- **Réunions de suivi quotidiennes (Daily Scrums)** : De courtes réunions quotidiennes ont permis à l’équipe de synchroniser ses efforts, d’évaluer l’état d’avancement par rapport à l’objectif du sprint, d’identifier rapidement les éventuels obstacles ou points de blocage, et de réajuster les priorités si nécessaire pour maintenir le cap.
- **Revue de sprint et Rétrospective** : À la fin de chaque sprint, une démonstration du travail accompli (incrément potentiellement livrable) a été présentée aux parties prenantes (notamment notre encadrante) lors de la revue de sprint, afin de recueillir leurs feedbacks. Une rétrospective de sprint a ensuite permis à l’équipe de réfléchir sur son propre processus de travail et d’identifier des axes d’amélioration pour le sprint suivant.
- **Tests et validation continus** : Des phases de test et de validation ont été intégrées de manière progressive à la fin de chaque sprint, voire en continu, pour assurer la qualité intrinsèque du code produit et vérifier la conformité fonctionnelle par rapport aux exigences définies, réduisant ainsi les risques d’intégration tardive.
- **Documentation agile** : Une documentation pertinente et suffisante a été produite de manière continue tout au long du développement (commentaires dans le code, etc.), dans le but de faciliter la compréhension, la maintenance future et la potentielle reprise du projet par d’autres équipes.

L’adoption de cette méthode Agile nous a conféré plusieurs avantages notables. Elle nous a permis de maintenir un rythme de développement soutenu et régulier, tout en nous assurant une grande réactivité face aux imprévus techniques inévitables ou aux précieuses suggestions d’amélioration formulées au fil de l’eau par notre encadrante, nos enseignants et les premiers utilisateurs tests impliqués dans le processus.

1.8 Diagramme de Gantt



1.9 Conclusion

Cette première phase cruciale du projet, englobant l'analyse du contexte, la définition de la problématique, la fixation des objectifs, l'étude comparative et le choix de la méthodologie, a permis d'établir une fondation extrêmement solide et cohérente pour la suite des opérations. Nous disposons désormais d'une compréhension claire et partagée des enjeux métiers et techniques, d'une problématique précisément circonscrite, et d'objectifs SMART (Spécifiques, Mesurables, Atteignables, Réalistes, Temporellement définis) qui sont réalistes et solidement cadrés dans le temps imparti pour le projet. Cette base rigoureuse est essentielle pour aborder sereinement les phases ultérieures de conception détaillée, de développement et de test.

Chapitre 2

Analyse et Conception

2.1 Introduction

La phase d'analyse et de conception constitue une étape absolument fondamentale et structurante dans tout processus de développement logiciel rigoureux. C'est durant cette phase que les besoins fonctionnels et non fonctionnels, initialement exprimés de manière souvent informelle par les parties prenantes (utilisateurs finaux, commanditaires, experts métier), sont méticuleusement traduits en spécifications techniques précises, claires, non ambiguës, structurées et directement exploitables par l'équipe de développement. L'objectif principal de cette étape est double : d'une part, formaliser de manière exhaustive les exigences fonctionnelles (ce que le système doit faire) et non fonctionnelles (comment il doit le faire – performance, sécurité, utilisabilité, etc.) de la future application ; d'autre part, anticiper au maximum les contraintes techniques potentielles (environnement de déploiement, interopérabilité, etc.) et modéliser les différents scénarios d'utilisation envisagés pour garantir que la solution réponde adéquatement aux attentes.

Afin d'assurer une modélisation à la fois rigoureuse, cohérente et universellement comprise de l'architecture du système envisagé, nous avons fait le choix stratégique d'adopter le langage de modélisation standardisé **UML (Unified Modeling Language)**. Ce langage graphique offre un cadre normé et éprouvé pour représenter visuellement les différents aspects du système : les interactions entre les acteurs (utilisateurs ou systèmes externes) et le système lui-même (cas d'utilisation), les processus métiers et les flux d'informations (diagrammes d'activité, de séquence), la structure statique des entités logicielles et leurs relations (diagrammes de classes), ainsi que l'organisation physique des composants (diagrammes de déploiement). L'utilisation d'UML facilite grandement la compréhension mutuelle des exigences et de la conception proposée, améliore la communication au sein de l'équipe de développement et avec les autres parties prenantes, et sert de référence tout au long du cycle de vie du projet.

2.2 Besoins fonctionnels détaillés

Les besoins fonctionnels décrivent précisément les actions spécifiques que le système logiciel doit être capable d'accomplir pour répondre aux attentes de ses utilisateurs et atteindre les objectifs fixés. Ils traduisent directement les fonctionnalités attendues de la plateforme de réseau social EMSI, en lien étroit avec les objectifs généraux définis lors de l'analyse initiale du contexte et de la problématique. Voici une présentation détaillée des

principales exigences fonctionnelles identifiées et spécifiées pour notre application :

- **Gestion complète de l'authentification et des comptes utilisateurs :**
 - **Inscription sécurisée avec validation par email :** Chaque nouvel utilisateur (étudiant, enseignant) doit pouvoir créer un compte personnel en fournissant des informations de base (nom, prénom, email institutionnel @emsi.ma, mot de passe). Le processus doit inclure une étape de validation de l'adresse email via un lien unique envoyé par courrier électronique pour confirmer l'identité et activer le compte.
 - **Connexion sécurisée :** Les utilisateurs enregistrés doivent pouvoir s'authentifier de manière sécurisée en utilisant leur adresse email institutionnelle et leur mot de passe. Des mécanismes de protection contre les attaques par force brute (limitation des tentatives, captcha) doivent être envisagés.
 - **Récupération de mot de passe :** Un processus sécurisé de réinitialisation de mot de passe doit être disponible pour les utilisateurs ayant oublié leurs identifiants.
 - **Gestion de profil :** Chaque utilisateur doit pouvoir consulter et modifier les informations de son profil (photo, biographie succincte, centres d'intérêt, etc.) et potentiellement gérer ses paramètres de confidentialité et de notification.
- **Interface dynamique et personnalisée selon le profil utilisateur :** L'interface utilisateur (UI) et l'expérience utilisateur (UX) doivent s'adapter dynamiquement en fonction du rôle attribué à l'utilisateur connecté (étudiant, enseignant, administrateur). Cette adaptation concernera principalement :
 - L'accès différencié aux fonctionnalités (par exemple, seuls les administrateurs accèdent au tableau de bord de gestion).
 - La présentation de la navigation et des menus.
 - Les droits spécifiques de création, modification, suppression ou modération de contenus.
- **Fil d'actualité centralisé et intelligent :** L'utilisateur connecté doit accéder à un fil d'actualités principal regroupant les publications récentes provenant de ses connexions, des groupes auxquels il appartient, et des annonces officielles de l'administration ou des enseignants. Ce fil devra offrir :
 - Un tri chronologique par défaut (publications les plus récentes en premier).
 - Des options de filtrage et/ou de tri alternatives, par exemple par importance (prioriser les publications des administrateurs ou enseignants), par pertinence (basée sur les interactions passées ou les groupes suivis), ou par type de contenu.
 - Un chargement potentiellement infini (infinite scroll) pour une navigation fluide.
- **Fonctionnalités de publication et de partage de contenu :**
 - **Création de publications :** Les utilisateurs (selon leurs droits) doivent pouvoir créer de nouvelles publications contenant du texte, des images (avec prévisualisation et potentiellement galerie), des liens (avec aperçu automatique si possible), et joindre des documents (PDF, Word, etc.).
 - **Édition et suppression :** Les auteurs de publications doivent pouvoir modifier ou supprimer leurs propres contenus.
 - **Interaction avec les publications :** Les utilisateurs doivent pouvoir interagir avec les publications via des actions telles que "liker" (ou autres réactions prédéfinies), commenter (avec support de texte riche simple et mentions), et potentiellement enregistrer ou partager une publication au sein de la plateforme.
- **Système de gestion de groupes avancé :**

- **Création et découverte de groupes** : La plateforme doit permettre la création de groupes de discussion thématiques, qui peuvent être publics (visibles et accessibles à tous les membres de l’EMSI) ou privés (visibles uniquement par les membres, accès sur invitation ou approbation). Un annuaire ou une fonction de recherche doit permettre de découvrir les groupes existants.
- **Administration de groupe** : Chaque groupe doit avoir un ou plusieurs administrateurs locaux (généralement l’enseignant pour un groupe de matière, ou des étudiants élus pour un club) capables de gérer les membres (invitations, approbations, exclusions), de modérer les publications et les commentaires au sein du groupe, et de personnaliser les paramètres du groupe (nom, description, image de couverture, règles).
- **Fil d’actualité de groupe** : Chaque groupe disposera de son propre fil d’actualité, affichant uniquement les publications spécifiques à ce groupe.
- **Mécanisme de signalement de contenu et modération** :
 - **Signalement par les utilisateurs** : Pour maintenir un environnement respectueux et conforme aux règles de l’établissement, les utilisateurs doivent avoir la possibilité de signaler facilement un contenu (publication ou commentaire) jugé inapproprié (discours haineux, harcèlement, spam, violation de droits d’auteur, etc.) en spécifiant la raison.
 - **Traitement des signalements par les administrateurs** : Les administrateurs de la plateforme (et potentiellement les administrateurs de groupe pour les contenus de leur groupe) seront notifiés des signalements. Ils devront disposer d’une interface dédiée pour examiner les contenus signalés, prendre une décision (supprimer le contenu, avertir l’auteur, ignorer le signalement) et potentiellement communiquer avec le signaleur ou l’auteur. Un traitement rapide et équitable des signalements est attendu.
- **Fonction de recherche globale** : Une barre de recherche facilement accessible doit permettre aux utilisateurs de rechercher des contenus (publications), des personnes (autres utilisateurs) ou des groupes au sein de la plateforme.

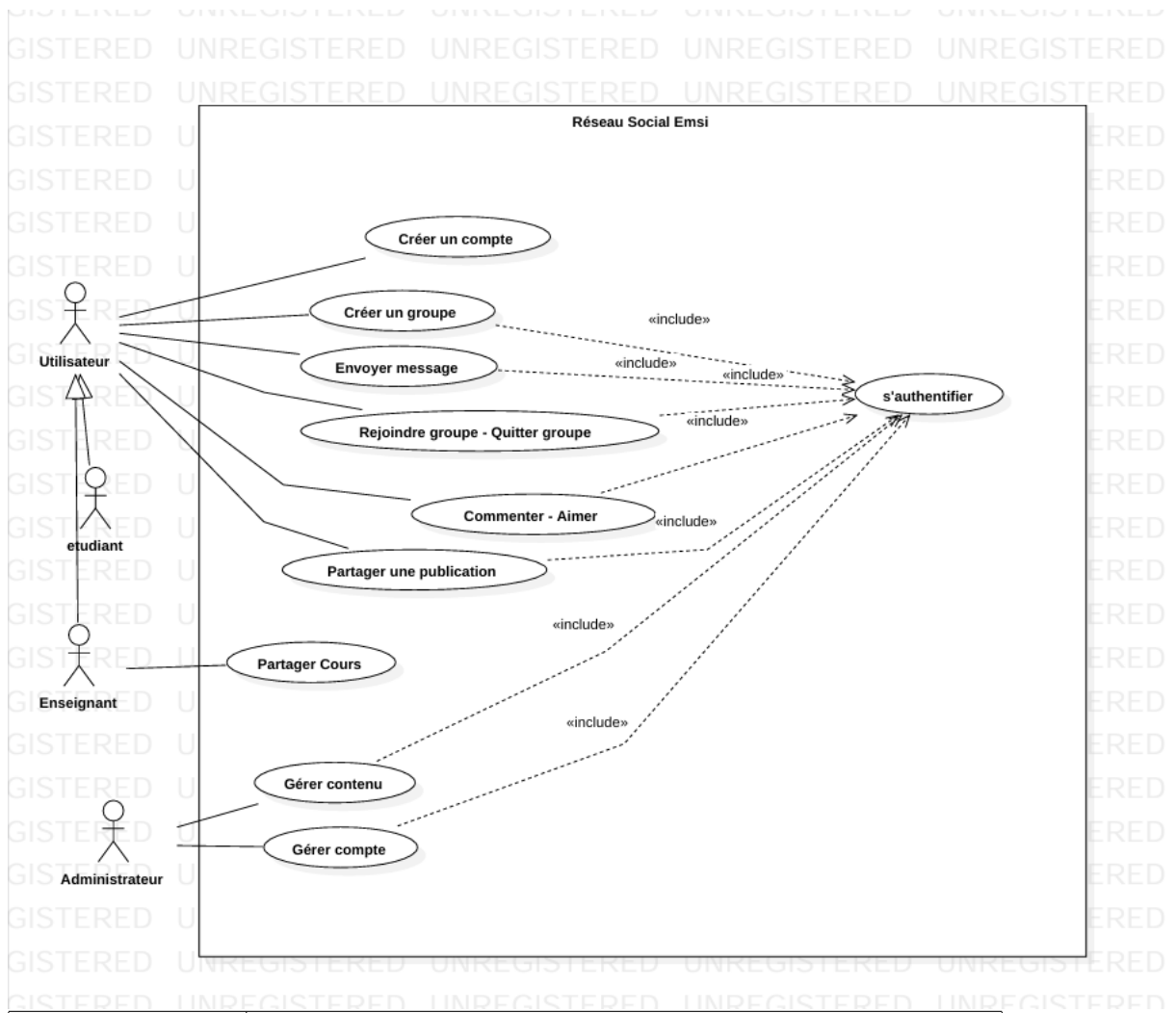
2.3 Besoins non fonctionnels

Au-delà des fonctionnalités spécifiques, les besoins non fonctionnels définissent les critères de qualité essentiels qui caractérisent le fonctionnement global du système et l’expérience qu’il procure. Ils sont cruciaux pour garantir l’adoption, la satisfaction des utilisateurs et la pérennité de l’application. Ils concernent des aspects tels que la performance, la sécurité, la fiabilité, la maintenabilité, l’utilisabilité et l’accessibilité. Voici les principaux besoins non fonctionnels retenus pour ce projet :

- **Sécurité Robuste** : La protection des données personnelles et des échanges est une priorité absolue. La plateforme devra implémenter des mesures de sécurité à plusieurs niveaux :
 - Utilisation systématique du protocole HTTPS pour chiffrer toutes les communications entre le client et le serveur.
 - Protection contre les vulnérabilités web courantes (OWASP Top 10), telles que les injections SQL, le Cross-Site Scripting (XSS), le Cross-Site Request Forgery (CSRF).
 - Stockage sécurisé des mots de passe (hachage fort avec salage).
 - Gestion fine et rigoureuse des permissions basée sur les rôles.

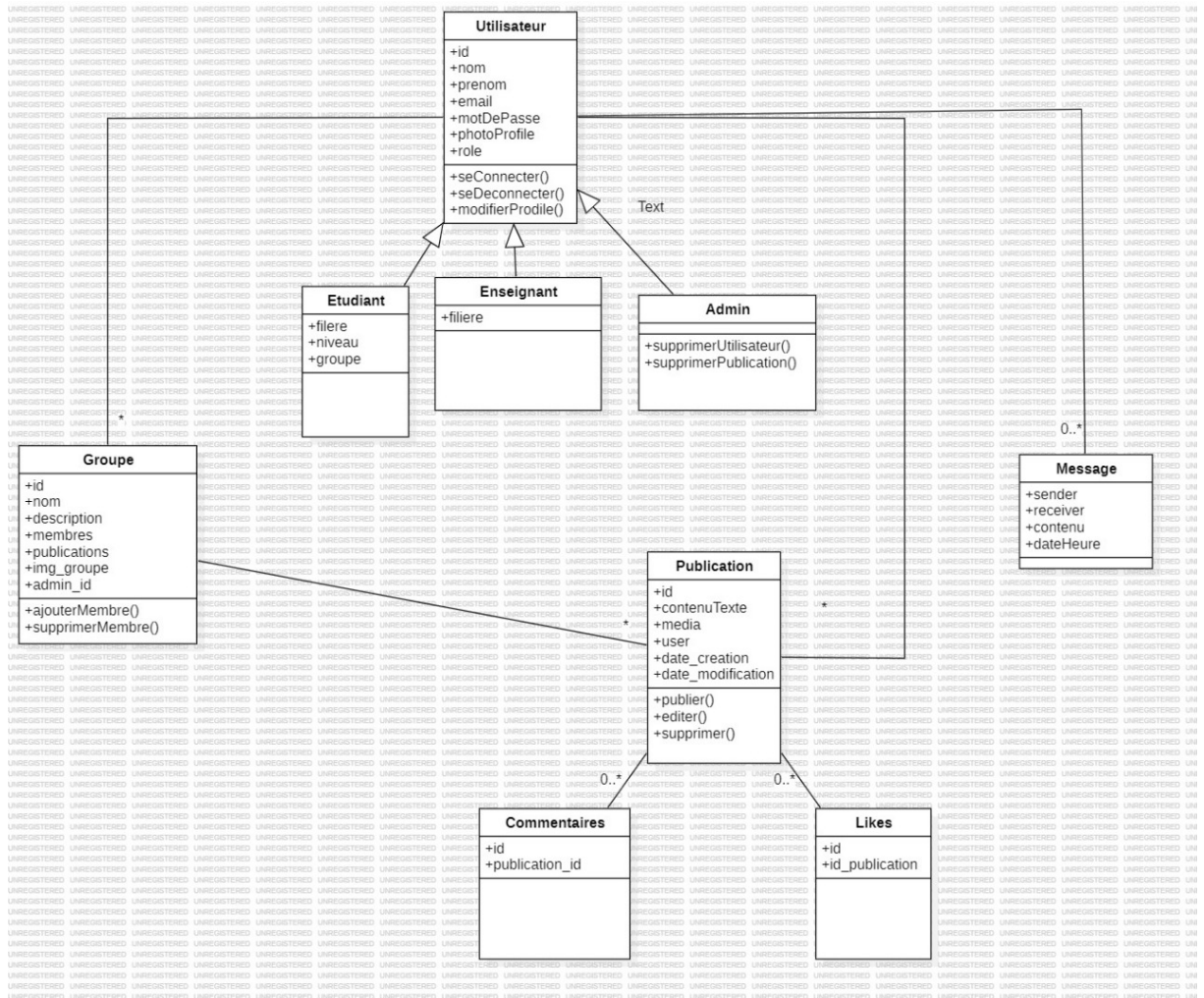
- Politique de confidentialité claire et accessible.
- **Performance et Réactivité** : L'application doit être rapide et réactive pour offrir une expérience utilisateur fluide.
 - Temps de chargement des pages principales inférieur à 3 secondes dans des conditions réseau standard.
 - Réponse rapide aux interactions utilisateur (clics, soumission de formulaires, chargement de commentaires).
 - Optimisation des requêtes base de données et mise en cache appropriée pour les données fréquemment accédées.
 - Capacité à gérer un nombre croissant d'utilisateurs et de contenus sans dégradation significative des performances (Scalabilité).
- **Fiabilité et Disponibilité** : Le système doit être stable et disponible pour les utilisateurs pendant les heures d'utilisation prévues.
 - Taux de disponibilité cible élevé (par exemple, 99.5
 - Gestion appropriée des erreurs et affichage de messages clairs pour l'utilisateur en cas de problème.
 - Mécanismes de sauvegarde régulière des données pour prévenir toute perte.
- **Utilisabilité et Expérience Utilisateur (UX)** : L'interface doit être intuitive, facile à apprendre et agréable à utiliser.
 - Navigation claire et cohérente à travers l'application.
 - Design épuré, moderne et respectant la charte graphique de l'EMSI.
 - Minimisation du nombre de clics pour accomplir les tâches courantes.
 - Feedback visuel immédiat lors des interactions.
- **Accessibilité** : L'application doit être utilisable par le plus grand nombre, y compris les personnes en situation de handicap.
 - Respect des principales directives d'accessibilité web (WCAG 2.1 niveau AA autant que possible).
 - Compatibilité avec les lecteurs d'écran et la navigation au clavier.
 - Contrastes de couleurs suffisants, textes alternatifs pour les images.
- **Maintenabilité et Évolutivité** : Le code source doit être bien organisé, documenté et facile à maintenir et à faire évoluer.
 - Respect des conventions de codage et des bonnes pratiques de développement.
 - Architecture modulaire et faiblement couplée (MVC).
 - Utilisation de technologies et de frameworks maintenus et disposant d'une bonne communauté.
 - Présence de tests automatisés (unitaires, intégration) pour faciliter les régressions.
- **Compatibilité Navigateurs et Appareils** : L'application web doit fonctionner correctement sur les dernières versions des principaux navigateurs (Chrome, Firefox, Safari, Edge) et s'adapter de manière responsive aux différentes tailles d'écran (ordinateurs de bureau, tablettes, smartphones).

2.4 Diagramme de cas d'utilisation



Acteur	Utilisateur
Précondition	L'utilisateur est déjà inscrit dans la base
Description	L'utilisateur entre ses identifiants, le système les vérifie
Postcondition	Accès au tableau de bord si succès

2.5 Diagramme de classes



2.6 Conclusion

Cette phase approfondie d'analyse et de conception a été absolument déterminante pour poser les fondations techniques et fonctionnelles du projet. Grâce à la formalisation rigoureuse des besoins fonctionnels et non fonctionnels, et à la modélisation précise de l'architecture à l'aide d'UML, nous avons pu structurer le développement futur autour d'une vision claire, partagée et cohérente. L'architecture choisie, basée sur le pattern MVC et des technologies modernes, est conçue pour être modulaire, facilitant ainsi les développements parallèles et l'intégration continue. Elle est également pensée pour être évolutive, afin de pouvoir intégrer de nouvelles fonctionnalités ou s'adapter à une charge croissante dans le futur. Cette étape nous a permis de réduire les ambiguïtés, d'anticiper les défis techniques et de fournir une base solide et documentée pour la phase de réalisation qui va suivre.

Chapitre 3

Réalisation et Tests

3.1 Introduction

La phase de réalisation et de tests représente le cœur opérationnel du projet, l'étape où les plans et les spécifications définis lors de l'analyse et de la conception prennent vie sous la forme d'un produit logiciel concret et fonctionnel. C'est durant cette phase intensive que les lignes de code sont écrites, les interfaces utilisateur sont construites, les bases de données sont peuplées et les différentes composantes du système sont assemblées et testées. Ce processus de développement s'est appuyé sur une combinaison d'outils de développement modernes, une architecture logicielle modulaire et éprouvée, ainsi qu'une organisation rigoureuse du code source et du travail d'équipe. L'objectif constant a été de garantir non seulement la fonctionnalité de l'application, mais aussi sa robustesse (capacité à gérer les erreurs), sa maintenabilité (facilité de correction et d'évolution) et son évolutivité (capacité à s'adapter aux changements futurs et à la montée en charge). Parallèlement, nous avons adopté une approche résolument centrée sur l'utilisateur final, accordant une attention toute particulière à l'ergonomie de l'interface, à la fluidité de la navigation entre les écrans, et à la réactivité globale du système face aux actions de l'utilisateur.

3.2 Architecture de la solution

L'architecture logicielle de notre plateforme de réseau social EMSI repose sur le paradigme architectural bien établi et largement adopté **MVC (Modèle – Vue – Contrôleur)**. Ce pattern architectural préconise une séparation claire des responsabilités au sein de l'application, ce qui améliore considérablement la modularité, la testabilité et la maintenabilité du code. Les trois composantes principales de cette architecture sont :

- **Modèle (Model)** : Cette couche est responsable de la logique métier de l'application et de la gestion des données. Elle encapsule les règles de gestion, interagit avec la source de données (la base de données dans notre cas) pour lire, écrire, mettre à jour et supprimer des informations, et assure la cohérence et l'intégrité des données. Dans notre implémentation, le Modèle est principalement réalisé à l'aide du système ORM (Object-Relational Mapping) intégré à Django. Cet ORM permet de manipuler les données de la base de données MySQL sous forme d'objets Python, simplifiant ainsi grandement les interactions avec la base et garantissant une certaine indépendance vis-à-vis du système de gestion de base de données

sous-jacent.

- **Vue (View)** : Cette couche correspond à l'interface utilisateur (UI) avec laquelle l'utilisateur final interagit directement. Elle est responsable de la présentation des données issues du Modèle à l'utilisateur et de la capture des actions de l'utilisateur (clics, saisies, etc.). Dans notre projet, la Vue a été développée en utilisant la bibliothèque JavaScript **ReactJS**. ReactJS est un choix moderne et populaire pour la construction d'interfaces utilisateur interactives et dynamiques. Il permet de créer des composants d'interface réutilisables, gère efficacement la mise à jour de l'affichage grâce à son DOM virtuel, et facilite la création d'applications monopages (Single Page Applications - SPA) offrant une navigation fluide et rapide sans rechargement complet de la page.
- **Contrôleur (Controller)** : Situé principalement dans la couche backend de l'application, le Contrôleur agit comme un intermédiaire entre le Modèle et la Vue. Il reçoit les requêtes HTTP provenant du frontend (initiées par les actions de l'utilisateur dans la Vue), interprète ces requêtes, interagit avec le Modèle pour récupérer ou modifier les données nécessaires, applique la logique métier appropriée, et finalement sélectionne et renvoie la réponse adéquate (souvent sous forme de données JSON) au frontend pour qu'elle soit affichée par la Vue. Cette couche a été développée en utilisant **Django REST Framework (DRF)**, une extension puissante de Django qui facilite grandement la construction d'APIs web RESTful robustes, sécurisées et conformes aux standards. DRF fournit des outils pour la sérialisation des données, la gestion des permissions, l'authentification, le routage des requêtes, etc.

Cette architecture MVC, combinée à une communication via API REST entre le frontend ReactJS et le backend Django, offre une séparation nette entre la présentation et la logique métier, favorisant ainsi la collaboration entre développeurs frontend et backend, et permettant une évolution indépendante des deux parties.

3.3 Technologies utilisées

Le développement de notre solution a mobilisé une stack technologique moderne, choisie pour sa stabilité, sa documentation riche, et sa forte communauté de soutien. Voici les outils principaux :

- **Frontend : ReactJS**, combiné avec Bootstrap pour une mise en page responsive. Des composants personnalisés ont été créés pour les publications, les commentaires, les groupes, etc., dans une logique de réutilisabilité.
- **Backend : Django REST Framework**, avec une structuration claire des endpoints REST, une gestion des permissions basée sur les rôles, et un système d'authentification sécurisé via **JWT (JSON Web Tokens)**.
- **Base de données : MySQL**, pour la gestion relationnelle des données. Le modèle de données a été conçu pour optimiser les requêtes les plus fréquentes (par exemple, les publications récentes d'un groupe).
- **Outils complémentaires** :
 - **GitHub** : gestion du versioning, suivi des commits, branches de développement, pull requests.
 - **Postman** : test et documentation des endpoints API.
 - **Render & Vercel** : déploiement respectif du backend et du frontend.

3.4 Interfaces principales développées

Un effort particulier a été mené sur le design des interfaces afin de rendre l'expérience utilisateur intuitive et agréable. Les principales interfaces mises en œuvre sont :

- **Page d'accueil** : regroupe les publications récentes, mises à jour dynamiquement. Les utilisateurs peuvent interagir (liker, commenter, enregistrer).
- **Système de notifications** : intégré en haut de page, il alerte l'utilisateur des interactions importantes (nouveau commentaire, publication dans un groupe, mention...).
- **Tableau de bord administrateur** : permet de gérer les utilisateurs, signalements, groupes, et contenus. Il offre des filtres, des statistiques, et une interface de modération.
- **Profil utilisateur** : chaque utilisateur peut modifier son profil, voir ses publications, et configurer ses préférences.

3.5 Méthodologie de tests

Nous avons mis en place une stratégie de test progressive et exhaustive, pour garantir la fiabilité et la stabilité de la solution :

- **Tests unitaires** : chaque fonction critique du backend a été testée isolément, notamment les endpoints liés à l'authentification, la création de publications, et la gestion des groupes.
- **Tests d'intégration** : des scénarios complets (ex : "authentifier un utilisateur, publier un contenu, commenter, signaler") ont été simulés pour tester l'interaction entre frontend, backend et base de données.
- **Tests utilisateurs** : une version bêta a été partagée avec un panel d'étudiants de l'EMSI, qui ont effectué des tests en situation réelle. Leurs retours ont été très utiles pour identifier les points de friction dans l'UX/UI.

3.6 Résultats des tests

Les tests ont permis de valider la stabilité de l'application et d'apporter des améliorations avant la mise en ligne finale. Voici les principaux constats :

- **95% des cas de test** ont été validés dès le premier cycle, ce qui témoigne de la robustesse initiale du code.
- **Bugs corrigés pendant les tests** :
 - Problèmes d'affichage de certaines images selon le format (.webp non supporté sur certains navigateurs)
 - Gestion incorrecte des erreurs de saisie dans les formulaires d'inscription et de création de publication
 - Latence dans la mise à jour des notifications (corrigée avec un système de WebSocket)
- **Performance** : le temps de chargement moyen sur réseau mobile a été jugé satisfaisant (moins de 2 secondes pour les pages principales).

3.7 Conclusion

La phase de réalisation a permis de concrétiser l'ensemble des éléments définis lors de l'analyse, en mettant l'accent sur la qualité du code, la modularité, et l'expérience utilisateur. Grâce à la rigueur appliquée dans l'architecture et aux nombreux tests réalisés, nous avons pu livrer une application stable, performante, et adaptée aux besoins de la communauté EMSI. Cette base logicielle offre également un excellent socle pour de futures évolutions (ex : intégration d'un chatbot, messagerie directe, ou espace de dépôt de devoirs).

Chapitre 4

Conclusion et Perspectives

Ce projet a permis de concrétiser nos acquis en développement web, architecture logicielle et gestion agile. Nous avons mis en pratique toutes les étapes d'un projet logiciel réel, de la conception à la mise en ligne.

Axes d'amélioration :

- Développement d'une application mobile native (React Native)
- Ajout d'un système de visioconférence intégré
- Implémentation d'algorithmes d'IA pour la détection automatique de contenus inappropriés

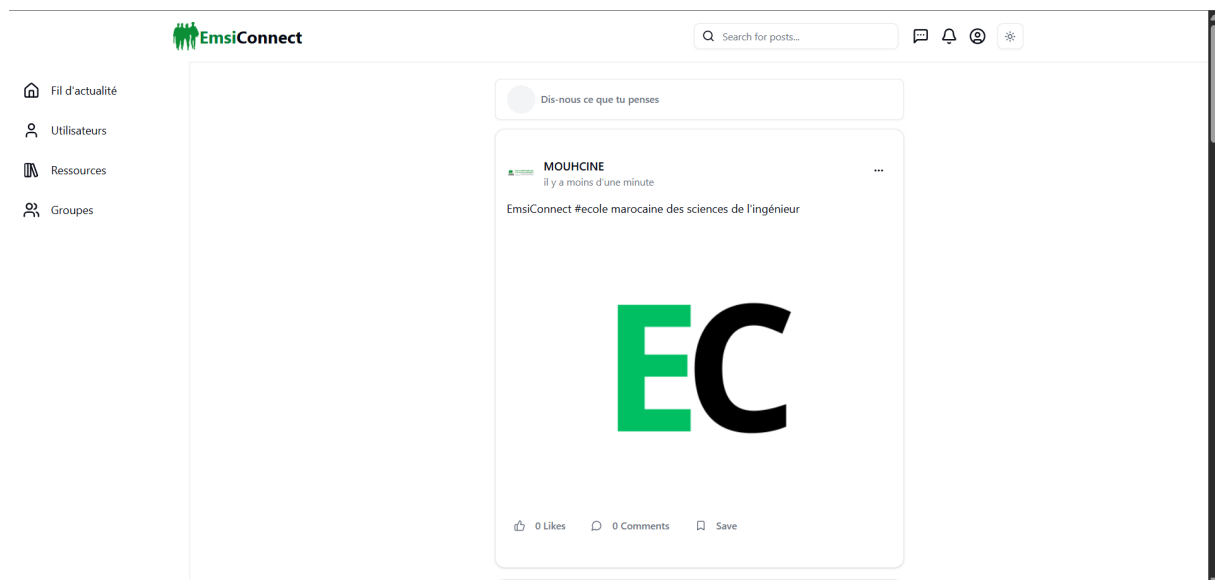


FIGURE 4.1 – Interface d'accueil de l'application

Bibliographie

Django Software Foundation (2023). Django Documentation. Disponible sur : <https://docs.djangoproject.com>

Meta Platforms, Inc. (2023). React : A JavaScript library for building user interfaces. Disponible sur : <https://reactjs.org>

Christie, T. et al. (2023). Django REST Framework Documentation. Disponible sur : <https://www.django-rest-framework.org>

Bootstrap Team (2023). Bootstrap Documentation. Disponible sur : <https://getbootstrap.com/docs>

Oracle Corporation (2023). MySQL Documentation. Disponible sur : <https://dev.mysql.com/doc>

BroCode (2022). React JS Full Course for Beginners. YouTube. Disponible sur : <https://www.youtube.com/c/BroCodez>

The Net Ninja (2022). Django React Tutorial Series. YouTube. Disponible sur : <https://www.youtube.com/c/TheNetNinja>

Stack Overflow (2023). Stack Overflow. Disponible sur : <https://stackoverflow.com>

Medium (2023). Medium. Disponible sur : <https://medium.com>

Schwaber, K. Sutherland, J. (2020). The Scrum Guide. Disponible sur : <https://scrumguides.org>

Object Management Group (2017). UML Specification. Disponible sur : <https://www.omg.org/spec/UML>

Jones, M., Bradley, J., Sakimura, N. (2015). JSON Web Token (JWT) – RFC 7519. Disponible sur : <https://datatracker.ietf.org/doc/html/rfc7519>