

**Département informatique Génie informatique (GI)**

**Année universitaire 2022/2023**

**Projet de fin d’études Sous le thème :**

*Développement de réseau social pour les étudiants de ESTF*

**Encadré par : Mr. Mehdi Tmimi et Mr. IBRIZ Abdelali**

**Réalisé par : Mohammed mesbahi aouam**

Sommaire

[Sommaire 2](#_Toc132142109)

[Remerciement 5](#_Toc132142110)

[Table d’abréviation 6](#_Toc132142111)

[Introduction 7](#_Toc132142112)

[Chapitre 1 : Cadre et contexte du projet 9](#_Toc132142113)

[1.1. Introduction 9](#_Toc132142114)

[1.2. Contexte du projet 9](#_Toc132142115)

[1.3. Présentation de l’organisme d’accueil : 9](#_Toc132142116)

[Chapitre 2 : Présentation et cahier des charges 11](#_Toc132142117)

[2.1. Introduction 11](#_Toc132142118)

[2.2. Présentation du projet 11](#_Toc132142119)

[2.3. Objectifs et périmètre de projet 12](#_Toc132142120)

[2.4. Public ciblé : 12](#_Toc132142121)

[2.5. Contenu de l’application 13](#_Toc132142122)

[2.5.1 Etape d’inscription 13](#_Toc132142123)

[2.5.2 Étape d’authentification : 13](#_Toc132142124)

[2.5.3 Espace Publications : 13](#_Toc132142125)

[2.5.4 Espace Notifications 13](#_Toc132142126)

[2.5.5 Espace Messagerie 13](#_Toc132142127)

[2.5.6 Espace Compte 13](#_Toc132142128)

[2.5.7 Espace étudiant 14](#_Toc132142129)

[2.6. Charte graphique : exigences et limites 14](#_Toc132142130)

[2.7. Arborescence de l’application 14](#_Toc132142131)

[2.8. Planification du projet 16](#_Toc132142132)

[2.8.1 Diagramme de GANTT 16](#_Toc132142133)

[Chapitre 3 : Technologies et outils utilisés 17](#_Toc132142134)

[3.1. Introduction 17](#_Toc132142135)

[3.2. Technologies implémentées 17](#_Toc132142136)

[3.3.1. Node js 17](#_Toc132142137)

[3.3.2. Angular 18](#_Toc132142138)

[3.3.3. MongoDB 18](#_Toc132142139)

[3.3.4. Express 18](#_Toc132142140)

[3.3.5. Socket.IO 19](#_Toc132142141)

[3.3.6. TypeScript 19](#_Toc132142142)

[3.3.7. CSS 19](#_Toc132142143)

[3.3.8. HTML 20](#_Toc132142144)

[3.3.9. Bootstrap 20](#_Toc132142145)

[3.1. Outils d’implémentation 20](#_Toc132142146)

[3.3.1. Visual Studio Code 20](#_Toc132142147)

[3.3.2. Postman 21](#_Toc132142148)

[3.3.3. MongoDB 21](#_Toc132142149)

[3.3.4. Angular DevTools 22](#_Toc132142150)

[4.1. Chapitre 4 : Architecture et conception 23](#_Toc132142151)

[4.1. Introduction 23](#_Toc132142152)

[4.2. Architecture de l’application 23](#_Toc132142153)

[4.2.1. Architecture 3-tiers (architecture applicative de l’application) 23](#_Toc132142154)

[Couche présentation (premier niveau) : 23](#_Toc132142155)

[Couche métier / business (deuxième niveau) : 24](#_Toc132142156)

[Couche accès aux données (troisième niveau) : 24](#_Toc132142157)

[4.2.2. MVC 2 (Modèle-Vue-Contrôleur 2) 24](#_Toc132142158)

[4.3. Conception de l’application 26](#_Toc132142159)

[4.3.1 Pourquoi UML ? 26](#_Toc132142160)

[4.3.2 Règles de gestion 27](#_Toc132142161)

[4.3.3 Diagrammes des cas d’utilisation 28](#_Toc132142162)

[Gestion globale de l’application 28](#_Toc132142163)

[4.3.4 Diagramme de classes 30](#_Toc132142164)

[4.3.5 Le Diagramme entité association : 31](#_Toc132142165)

[5.1. Chapitre 5 : Présentation de l’application 36](#_Toc132142166)

[5.1. Introduction 36](#_Toc132142167)

[5.2. L’accueil de l’application 36](#_Toc132142168)

[5.3. Etape d’inscription 37](#_Toc132142169)

[5.4. Etape d’authentification 38](#_Toc132142170)

[5.5. Espace utilisateur 39](#_Toc132142171)

[5.5.1 Page d'actualité 39](#_Toc132142172)

[5.5.2 Nouvelle publication 40](#_Toc132142173)

[5.5.3 Mes messages 41](#_Toc132142174)

[5.5.4 Mon Profile 42](#_Toc132142175)

[5.6. Espace administrateur 43](#_Toc132142176)

[5.6.1 Page utilisateurs 43](#_Toc132142177)

[5.6.2 Page nouveaux utilisateurs 44](#_Toc132142178)

[5.6.3 Page Publications 45](#_Toc132142179)

[5.6.4 Page nouveaux publications 46](#_Toc132142180)

[Conclusion 48](#_Toc132142181)

[Bibliographie 49](#_Toc132142182)

[Table de figures 50](#_Toc132142183)

[Table de matières 51](#_Toc132142184)

Remerciement

En préambule à ce projet fin d’étude nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces deux années d’étude.

Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l’élaboration de ce projet ainsi qu’à la réussite de cette formidable formation universitaire.

Ces remerciements vont tout d’abord à Monsieur le professeur Mehdi Tmimi en tant qu’un encadrant de ce projet, s’est toujours montré à l’écoute, très disponible tout au long de la réalisation de ce projet, ainsi pour l’inspiration, l’aide et le temps qu’il a bien voulu nous consacrer et sans qui ce projet n’aurait jamais vu le jour.

Nous tenant à remercier sincèrement l’honorable membre de jury, qui ont acceptés de nous accorder son précieux temps pour juger ce travail.

Enfin, nous remercions tout le corps professoral et administratif de l’École Supérieure de Technologie Fès surtout notre professeur Monsieur Mehdi Tmimi pour la richesse et la qualité de son enseignement et qui déploie de grands efforts pour assurer à leurs étudiants une formation actualisée.

Merci à tous et à toutes.

Table d’abréviation

**NoSQL** :"non seulement SQL", (« Not Only Sql»).

**CLI** : Interface en ligne de commande, (« Command Line Interface »).

**IDE** : Environnement de développement intégré (« Integrated Development Environment.»).

**MVC** : Modèle-Vue-Contrôleur, (« Model-View-Controller »)

**UML** : langage de modélisation unifié (« Unified Modeling Language »).

**HTML** : Hypertext Markup Language.

**CSS** : feuilles de style en cascade, (« Cascading Style Sheets »).

**DOM** : Document Object Model.

**ERD**: Entity-Relationship diagram

Introduction

Le projet de fin d’études est souvent considéré comme un élément clé de parcours universitaire dans lequel les étudiants mettre en pratique les connaissances et les compétences qu’elles sont acquises au cours de leur formation universitaire. En travaillant sur un projet de recherche ou une expérience pratique, les étudiants peuvent appliquer les théories et les concepts qu’ils ont appris dans leur domaine d’études et ainsi mieux comprendre leur pertinence dans un contexte concret.

De plus, le projet de fin d’études est souvent un projet de grande envergure qui nécessite une planification minutieuse et une gestion rigoureuse du temps. En travaillant sur tel projet, les étudiants peuvent développer des compétences importantes telles que la gestion de projet, la résolution de problème et la prise de décision.

Dans ce cadre en réalisant un réseau social pour les étudiants de notre école, l’école supérieur de technologie de Fès, en utilisant deux technologies clés du développement des applications web modernes et courant qui permettent aux développeurs de construire des applications web riches et performantes.

Cette application considère comme un outil de communication en ligne qui permet aux étudiants de communiquer facilement et efficacement pour partager leurs idées, leurs compétences et leurs opinions entre eux.

Afin de comprendre la démarche que nous avons utilisée pour mener ce projet à son terme, notre rapport ce structure de la façon suivante :

En premier lieu en va aborder le contexte général de l’école qui a parrainé notre projet. Après, on passera à la définition générale de l’aspect de notre travail. Puis dans une seconde partie, nous présentent les outils et les technologies que nous avons utilisées pour réaliser ce travail. Ensuit dans une troisième partie on va aborder la tache technique avant que nous décrivions une démonstration de fonctionnement du projet.

Chapitre 1 : Cadre et contexte du projet

1. Introduction

Ce premier chapitre s’intéressera à la présentation du cadre du projet en tant qu’un travail réalisé lors du projet de fin d’études, toute en présentant notre école où nous effectuions ce travail.

1. Contexte du projet

Notre projet réalisé à l’occasion de mettre en pratique nos connaissances et nos compétences qu’ont prises durant notre parcours universitaire à la forme d’un projet de fin d’études, dont le sujet le développement d’un réseau social pour les étudiants de l’école supérieure de technologie de Fès.

1. Présentation de l’organisme d’accueil :

L'école supérieure de technologie de Fès (EST de Fès) est une institution d'enseignement supérieur marocaine située dans la ville de Fès. Elle a été créée en 1992 et est rattachée à l'Université Sidi Mohamed Ben Abdellah.

L'EST de Fès propose des formations de niveau bac+2 (DUT) et bac+3 (Licence professionnelle) dans différents domaines techniques tels que l'informatique, l'électronique, le génie civil, la mécanique, le génie industriel, le textile, le tourisme, le commerce et la gestion.

L'EST de Fès a pour mission de former des techniciens supérieurs capables de répondre aux besoins du marché du travail national et international, ainsi que de mener des projets de recherche appliquée en partenariat avec les entreprises et les institutions publiques.

L'EST de Fès est réputé pour la qualité de ses formations et la proximité de son enseignement avec les besoins du marché de l'emploi. Elle compte également un corps enseignant compétent et expérimenté, ainsi que des infrastructures modernes et bien équipées pour l'enseignement et la recherche.

L'École Supérieure de Technologie de Fès (ESTF) propose différents services pour accompagner les étudiants dans leur parcours universitaire et leur offrir les meilleures conditions pour réussir leur formation. Voici quelques-uns des services proposés par l'ESTF :

Service de la scolarité : ce service est en charge de la gestion administrative des étudiants. Il s'occupe notamment de l'inscription, de la délivrance des diplômes et des relevés de notes, de la gestion des absences et des retards, etc.

Service de l'orientation et de l'insertion professionnelle : ce service aide les étudiants à choisir leur parcours de formation en fonction de leurs centres d'intérêt et de leurs compétences. Il les accompagne également dans leur recherche de stage et d'emploi en organisant des forums, des rencontres avec les entreprises, des simulations d'entretien d'embauche, etc.

Bibliothèque : l'ESTF dispose d'une bibliothèque bien fournie en ouvrages spécialisés dans les domaines techniques et technologiques, ainsi qu'en revues scientifiques et techniques. Les étudiants peuvent y accéder librement pour effectuer des recherches ou emprunter des livres.

Infirmerie : l'ESTF dispose d'une infirmerie pour assurer les premiers soins en cas de malaise ou de blessure. Un infirmier est disponible pour assister les étudiants en cas de besoin.

Services informatiques : l'ESTF dispose de salles informatiques équipées de matériel moderne et de logiciels spécialisés pour les différentes filières de formation. Les étudiants peuvent également accéder à internet depuis ces salles ou depuis les bornes wifi disponibles dans les espaces communs. Activités parascolaires : l'ESTF encourage la pratique des activités sportives et culturelles en proposant des activités parascolaires telles que le football, le basketball, le volleyball, la musique, la danse, etc.

Chapitre 2 : Présentation et cahier des charges

1. Introduction

Ce chapitre sera réservé à l’étude fonctionnelle de l’application à savoir les différents aspects fonctionnels de notre projet.

On commencera notre chapitre par révéler les différentes motivations de la création de l’application puis on présentera les aspects fonctionnels de ce projet, c’est-à-dire le but du travail. Ensuite on passera à la population intéressée par le réseau social et en terminera par déclarer l’ensemble des aspects fonctionnels de l’application et les exigences du choix de la charte graphique

1. Présentation du projet

De plus en plus, les réseaux sociaux ont connu une croissance exponentielle et ils sont devenus un moyen incontournable de communication et d'interaction sociale. D’où l’idée de créer un réseau social qui répond aux besoins de ses utilisateurs en leur offrant une plateforme sécurisée et conviviale pour se connecter, partager des idées et échanger des informations.

Ce réseau social a été créé dans le but de favoriser les échanges entre les étudiants de différents niveaux et de faciliter la communication entre les membres de notre communauté.

Grâce à ce réseau social, vous pourrez facilement vous connecter avec d'autres étudiants, partager des informations utiles, poser des questions, organiser des événements et vous tenir informé de tout ce qui se passe dans notre école. Vous pourrez également créer des conversations individuelles ou en groupe, partager des publications qui peuvent être des photos et des vidéos ou un texte et recevoir des notifications sur les nouveautés.

Notre réseau social est facile à utiliser et accessible à tous les étudiants de l'école. Il est également sécurisé et protégé, ce qui signifie que seuls les étudiants de l’ESTF peuvent y accéder.

1. Objectifs et périmètre de projet

Le projet opte à la communication grâce à un réseau social qui permet la connexion entres les étudiants, le partage des informations utiles, l'organisation des événements et la création d’un environnement d’étude et de motivation.

* Tout étudiant approuvé par l’administrateur peut :
* Ajouter ses données personnelles dans son profil et les modifier.
* Suivre ses amis pour recevoir leurs nouvelles.
* Réagir ou commenter sur les postes des autres étudiants pour partager son opinion ou son idée.
* Recevoir des notifications sur les postes publiés ou les nouveaux commentaires et réactions.
* Ajouter, supprimer ou modifier une publication qui peut contenir du texte, des images ou des vidéos.
* Se communiquer dans des conversations individuelles ou collectives pour le partage des informations.
* L'étudiant a le droit de s’inscrire dans l’application par son compte académique et par ses informations personnelles et ne sera fonctionnelles qu’après être approuver par l’administrateur. Ce dernier a le droit total d’ajouter ou de supprimer ou suspendre un étudiant en cas de violation des limites.

1. Public ciblé :

L'application concerne les étudiants de l’école supérieure de technologie de Fès de toutes les filières et de tous les niveaux.

1. Contenu de l’application

L'application doit offrir une espace propre aux étudiants grâce aux publications qui seront vu que par les étudiants de l’ESTF après avoir passé par une phase d’authentification en donnant la possibilité d’inscription pour les nouveaux étudiants.

1. Etape d’inscription

Tout étudiant volant profiter de l’application doit passer par la phase d’inscription en fournissant certaines informations personnelles c’est à dire le nom, le compte académique -Email-, et son image.

1. Étape d’authentification :

Afin d’accéder à l’application et utiliser ses différentes fonctionnalités (messagerie, notifications, publications...), tout utilisateur doit s’identifier par un compte académique et un mot passe.

1. Espace Publications :

Tout membre de l’application possède une espace où il peut dans une première partie d’ajouter des publications qui peut être de texte, des images ou des vidéos. Et dans une deuxième partie de voir les postes des autres membres classés selon leurs dates en donnant la possibilité de réagir sur une publication, d’ajouter un commentaire, répondre ou réagir à un commentaire déjà existant.

1. Espace Notifications

Pour que les membres de la communauté étudiante peuvent être à jour dans l’application ils peuvent recevoir toutes les nouvelles sur les publications partagés par les autres étudiants, les réactions ou les commentaires sur ses publications et aussi les réactions ou les réponses des autres étudiants sur ses commentaires dans les publications des autres membres.

1. Espace Messagerie

Un membre peut consulter ses messages envoyés et reçus. Et il a également la possibilité de répondre à ses messages, de voir l’émetteur de message et de distinguer ceux qui n’ont pas encore lus.

1. Espace Compte

Un membre a des droits sur ses informations personnelles. Il peut donc modifier à tout moment ses données telles que son pseudo, son mot de passe… et aussi son image personnelle (avatar).

1. Espace étudiant

• L'étudiant doit être en mesure de publier des publications sur la plateforme pour partager ses expériences et ses activités avec ses camarades.

• Les étudiants peuvent recevoir des notifications pour leurs messages et être en mesure de répondre à ces notifications pour interagir avec leurs camarades.

• L'étudiant doit être en mesure de gérer son profil, y compris des informations telles que son nom, sa photo de profil et son statut académique.

• Les étudiants peuvent également gérer leur liste de suivi, en ajoutant ou en supprimant des étudiants pour suivre leurs activités sur la plateforme.

Espace administrateur

• L'administrateur est capable de gérer l'ensemble des comptes des étudiants inscrits sur la plateforme, y compris les nouveaux inscrits et les utilisateurs existants.

• L'administrateur peut également gérer les publications publiées par les étudiants pour maintenir un environnement sain et positif sur la plateforme.

1. Charte graphique : exigences et limites

L'application doit être flexible et simple à utiliser pour gager la fidélité d’utilisateur, ce dernier doit être capable de naviguer entre les pages avec une clique.

Les différentes couleurs et choix typographiques doivent permettent à un utilisateur de repérer les différentes fonctionnalités qui s’offrent à lui.

Pour toutes questions ou informations, l’utilisateur doit pouvoir contacter l’administrateur général.

1. Arborescence de l’application

Parmi les étapes de la conception d’une application web, il y a l’arborescence de l’application, qui est une étape très importante, car elle conditionne les modes d’accès à l’information et la navigation sur l’application, elle est donc importante pour l’ergonomie d’une application web.

L’arborescence de notre application part de la page d’accueil, puis elle est structurée en quatre rubriques : « Accueil », « Notifications », et « Message », et « Profil », et chacune de ces rubriques contient des sous rubriques et des pages.

Le schéma suivant illustre l’arborescence de notre application :

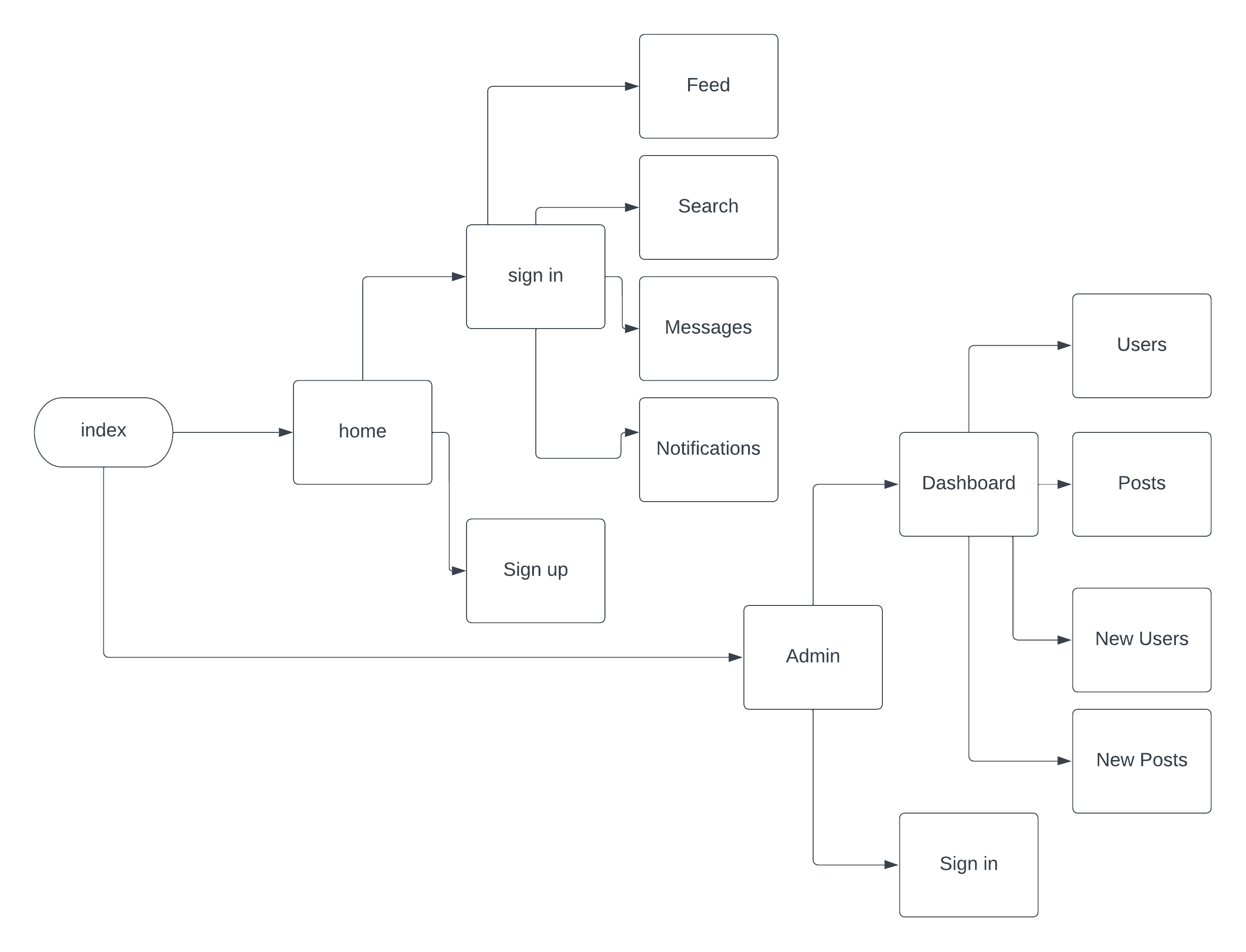


Figure 1Arborescence de l'application

1. Planification du projet

La planification du projet est l’activité qui consiste à déterminer et à ordonnancer les tâches du projet, à estimer leurs charges et à déterminer les profils nécessaires à leur réalisation. L’objectif principale de la planification (le planning) est suivi et communiquer l’avancement du projet, et déterminer si Les objectifs des tâches sont réalisés ou dépassés

1. Diagramme de GANTT

Le diagramme de GANTT est la technique et la représentation graphique permettant de renseigner et situer dans le temps les tâches et ressources du projet. Voici le diagramme de GANTT que nous avons suivi :

Figure 2 Diagramme de GANTT

**Conclusion**

Il est important de prendre le temps de réfléchir et de planifier ces étapes afin de maximiser les chances de réussite du projet. Une présentation bien conçue et un cahier des charges complet permettront de définir les rôles et responsabilités de chacun, de gérer efficacement les ressources et de garantir la satisfaction du client ou des parties prenantes.

En somme, une présentation réussie et un cahier des charges solide sont des éléments clés pour le succès de tout projet, et doivent être considérés avec sérieux et rigueur dès le début du processus.

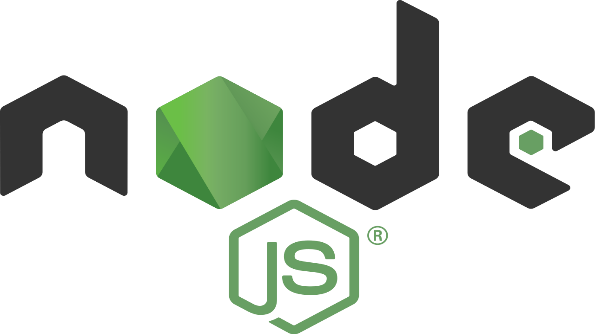
Chapitre 3 : Technologies et outils utilisés

1. Introduction

Dans ce chapitre on abordera plutôt l’aspect technique implémenté par notre application, à l’inverse du deuxième chapitre consacré à l’aspect fonctionnel ; on commencera par présenter notre choix de technologies, pourquoi ? Et pour quel but ? Puis on présentera l’ensemble des technologies et outils utilisés depuis la conception jusqu’à la réalisation.

Du fait de la puissance de la technologie java, on a opté pour les mêmes raisons d’implémenter notre application selon l’architecture JEE, en offrant une abstraction et séparation des différentes couches de l’application.

1. Technologies implémentées
2. Node js

**Node**.**js :** est une plateforme de développement qui permet d'exécuter du code JavaScript côté serveur. Node.js utilise un modèle de programmation événementielle, ce qui signifie qu'il est capable de gérer de nombreux événements simultanément de manière efficace, rendant Node.js particulièrement adapté aux applications en temps réel telles que les chats, les jeux et les réseaux sociaux

1. Angular

**Angular** : un Framework de développement d'applications web basé sur TypeScript. Il est utilisé pour construire l'interface utilisateur (UI) de votre application web. Angular offre une structure solide pour l'application et permet de créer des applications web complexes en utilisant des composants, des services, des directives et des pipes

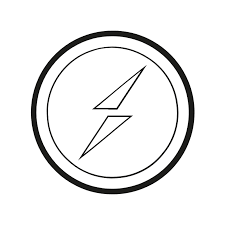
1. MongoDB

**MongoDB** est une base de données NoSQL (Not Only SQL) qui stocke les données sous forme de documents JSON. MongoDB offre une grande flexibilité pour stocker et organiser les données en évitant les contraintes liées aux relations et schémas rigides. Elle est particulièrement bien adaptée aux projets de développement web où la structure de données évolue constamment

1. Express

**Express** est un Framework Node.js utilisé pour construire des applications web basées sur Node.js. Express fournit un ensemble d'outils pour la création de serveurs HTTP, la gestion des requêtes, la définition des routes et la gestion des erreurs. Express est également très extensible grâce à la disponibilité de nombreux modules complémentaires

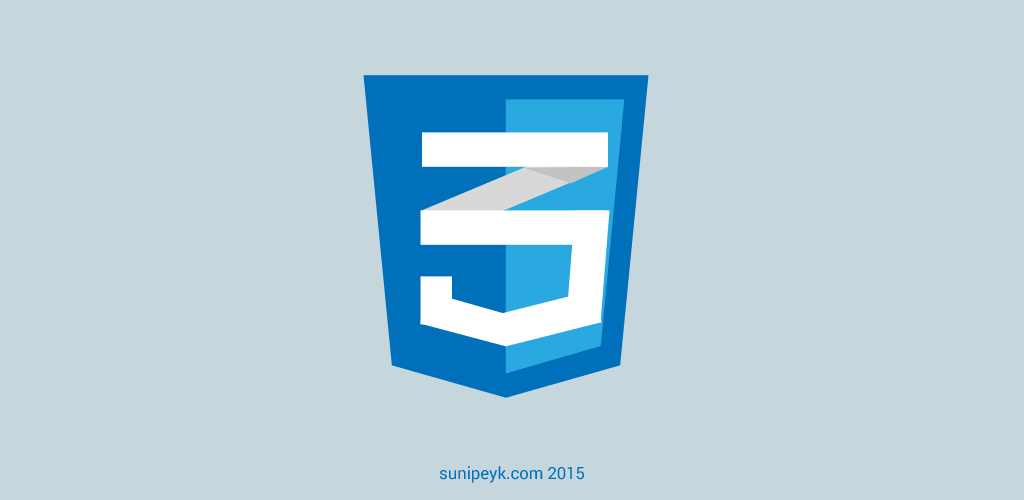
1. Socket.IO

**Socket.IO** est une bibliothèque JavaScript qui permet une communication en temps réel entre le client et le serveur. Elle est souvent utilisée dans des applications de chat, de jeux en ligne, de tableaux de bord en temps réel et de réseaux sociaux. Socket.IO fournit une abstraction sur le protocole WebSocket, ce qui rend la mise en œuvre de la communication en temps réel plus facile et plus fiable

1. TypeScript

**TypeScript** est un langage de programmation open-source développé par Microsoft qui offre des fonctionnalités supplémentaires à JavaScript telles que la vérification statique des types, les interfaces, les classes, les modules et autres fonctionnalités de programmation orientée objet. Il a été créé pour résoudre certains des problèmes courants de JavaScript en fournissant une meilleure vérification des erreurs de typage et une aide à la saisie plus efficace, ce qui peut aider à améliorer la qualité et la fiabilité du code

1. CSS

**CSS** (Cascading Style Sheets) est un langage de feuilles de style utilisé pour décrire la présentation d'un document écrit en HTML, XML ou autre langage de balisage. CSS permet de définir des règles de mise en forme et de style pour les éléments HTML, tels que la taille et la couleur des textes, les marges, les bordures, les arrière-plans, les polices de caractères, les effets de transition,

1. HTML

**HTML** (signifie HyperText Markup Language.) : C'est un langage de balisage utilisé pour créer et structurer du contenu sur le web. HTML est la base de la plupart des sites web et est utilisé pour définir la structure des pages web. HTML se compose d'une série de balises et d'attributs qui définissent les différents éléments d'une page web. Par exemple, la balise <html> est utilisée pour définir le début et la fin d'un document HTML, tandis que la balise <head> est utilisée pour définir les métadonnées du document, telles que le titre, l'auteur.

1. Bootstrap

******Bootstrap** est un Framework open-source de développement web qui permet de créer rapidement des sites web adaptatifs (responsive) et esthétiquement attrayants. Il utilise du HTML, du CSS et du JavaScript pour faciliter la conception de pages web et simplifier la création de mises en page, de formulaires, de boutons, de menus, de carrousels, de modèles de grille, etc. Bootstrap fournit également une bibliothèque de composants prêts à l'emploi et personnalisables, ainsi que des fonctionnalités JavaScript telles que des modales, des onglets, des popovers, des tooltips, etc. qui peuvent être facilement ajoutés aux pages web.

1. Outils d’implémentation
2. Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source gratuit et open-source développé par Microsoft. Il a été lancé en 2015 et est rapidement devenu l'un des éditeurs de code les plus populaires auprès des développeurs. VS Code est conçu pour être utilisé avec différents langages de programmation tels que JavaScript, TypeScript, Python, Java, C++, C#, Ruby, PHP, et bien d'autres. Il est disponible pour les systèmes d'exploitation Windows, MacOs et Linux. Il propose de nombreuses fonctionnalités pour faciliter la création, l'édition et le débogage de code. Il dispose d'une interface utilisateur personnalisable, d'un système d'extensions permettant d'ajouter des fonctionnalités supplémentaires pour répondre aux besoins spécifiques des développeurs.

1. Postman

**Postman** est un outil de collaboration pour les développeurs d'API qui permet de tester, de documenter et de partager des API plus facilement. Il est disponible en version gratuite et en version payante avec des fonctionnalités supplémentaires. Avec Postman, les développeurs peuvent créer des requêtes HTTP pour tester les API, analyser les réponses et déboguer les erreurs. Il propose une interface utilisateur intuitive pour faciliter la création de requêtes HTTP personnalisées, l'ajout d'en-têtes et de paramètres de requête, la gestion de cookies, la validation des données de réponse et bien plus encore. Postman offre également une fonctionnalité de partage de documentation d'API pour aider les développeurs à communiquer plus facilement sur leur API avec leur équipe et leur client. Les développeurs peuvent créer une documentation complète d'API avec des exemples de requêtes, des descriptions de chaque méthode et des réponses

1. MongoDB

MongoDB Atlas est un service de base de données cloud qui permet aux utilisateurs de créer, gérer et mettre à l'échelle facilement des bases de données MongoDB. Il est proposé par MongoDB, Inc. et fournit une plate-forme de base de données en tant que service (DBaaS) entièrement gérée. Atlas offre un certain nombre de fonctionnalités, notamment la mise à l'échelle automatique, la sauvegarde et la récupération, le patching automatisé, des clusters globaux pour les déploiements multi-régions et des fonctionnalités de sécurité intégrées. Avec Atlas, les utilisateurs peuvent rapidement créer de nouvelles instances de base de données, ajuster les ressources en fonction de leurs besoins et se connecter facilement à leurs bases de données depuis n'importe où dans le monde. Il offre également une interface conviviale basée sur le Web et une API puissante pour la gestion des bases de données de manière programmatique

1. Angular DevTools

**Angular DevTools** étend Chrome DevTools en ajoutant des fonctionnalités de débogage et de profilage spécifiques à Angular. Nous pouvons utiliser Angular DevTools pour comprendre la structure de notre application et prévisualiser l’état de la directive et des instances de composants. Pour obtenir des informations sur l’exécution de l’application, nous pouvons utiliser l’onglet profileur, qui vous montre les cycles de détection des changements individuels, ce qui les a déclenchés et combien de temps Angular y a passé.

**Conclusion**

Ce chapitre a discuté des techniques et des outils utilisés dans divers domaines tels que les technologies de l'information, la santé, l'ingénierie, l'éducation et les affaires. La technologie évolue rapidement, offrant de nouvelles opportunités pour améliorer les processus et les résultats dans de nombreuses industries

.

1. Chapitre 4 : Architecture et conception
2. Introduction

Dans ce chapitre, on discutera de la conception, on parlera d'abord de l’architecture de l’application (l’architecture trois tiers puis du modèle MVC 2) ; ensuite de l'UML qui a été la base essentielle de notre travail puisqu’il nous a donné la chance de se servir de trois diagrammes : diagramme de cas d'utilisation et diagrammes de classes diagrammes entité association.

1. Architecture de l’application
2. Architecture 3-tiers (architecture applicative de l’application)

*L'architecture 3-tiers* (de l'anglais tiers signifiant étages ou niveaux) est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches, étages, niveaux ou strates dont le rôle est clairement défini :

* La présentation des données : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
* Le traitement métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
* L'accès aux données persistantes : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Les trois couches de l’architecture 3-tiers sont :

Couche présentation (premier niveau) :

Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'Interface Homme Machine.

En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en HTML pour être exploitée par un navigateur web ou en WML pour être utilisée par un téléphone portable.

Couche métier / business (deuxième niveau) :

Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation. Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche.

La couche métier offre des services applicatifs et métier à la couche présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés.

Couche accès aux données (troisième niveau) :

Elle consiste en la partie gérant l'accès aux gisements de données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme.

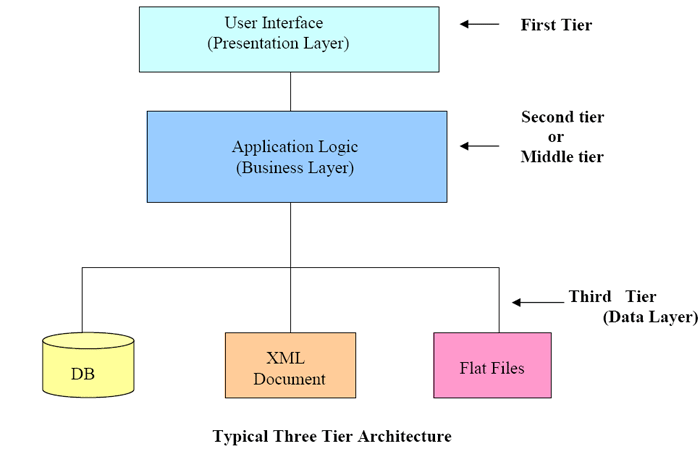


Figure 3 : Architecture 3-tiers

Dans notre projet, pour la couche présentation, on a utilisé le Framework Struts 2, et pour la couche accès aux données, ont utilisé le Framework Hibernate.

1. MVC 2 (Modèle-Vue-Contrôleur 2)

Une architecture MVC cherche à séparer trois choses : le Modèle, les Vues et les Contrôleurs. Les contrôleurs permettent de répondre aux actions de l'utilisateur. Chaque contrôle est associé à une vue : cette dernière permet de présenter l'information retournée à l'utilisateur. Bien entendu, l'information renvoyée est dépendante des actions d'entrées de l'utilisateur (capturées par les contrôleurs, nous venons de le dire). Les liens (les traitements) sont réalisés par le modèle (la logique métier).

Dans le but d'améliorer ce point, certains préfèrent implémenter une architecture MVC 2. Dans une telle architecture, il n'existe plus qu'un seul et unique contrôleur réceptionnant toutes les requêtes clientes.

Le contrôleur unique devient le point d’entrée exclusif de l’application. Il devient alors très aisé de centraliser la gestion des accès, des droits, des statistiques ou de toute autre fonctionnalité transverse.

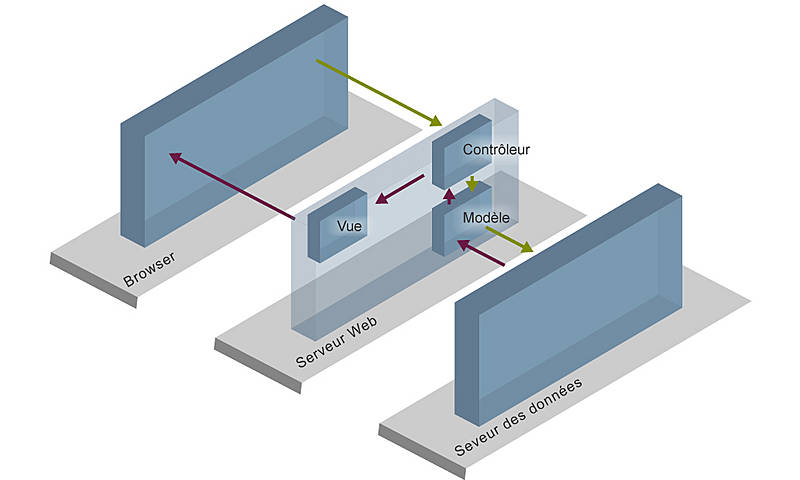


Figure 4 : Schéma détaillé du model MVC 2

1. Conception de l’application

Afin de concevoir notre jeu, nous avons opté pour UML comme langage de conception et de modélisation.

1. Pourquoi UML ?

UML est apparu au cours des années 90 et tient sa force de la fusion des méthodes de BOOCH, de la méthode OMT et OOSE qui préexistaient et dont il a su tirer le meilleur. En pratique, il s’agit, comme pour tout formalisme, d’un ensemble de diagrammes permettant dès lors que l’on en comprend la signification, de décrire simplement un ensemble de données, d’actions ou de traitements, permettant ainsi la création d’une analyse complète pour réaliser un projet informatique (une application web dans notre cas).

**UML :**

* Un langage de Modélisation (standard) ;
* Basé sur une approche solide Objectory ;
* Un support important de stratégies et d’heuristiques ;
* Adapté au développement Objet ;
* Adapté au développement Itératif ;
* Adaptable au processus d’une équipe ;
* Supporté par des outils.

**UML & POO :**

* UML utilise une approche objet des logiciels ;
* Les entités métier sont identifiés comme des objets ;
* Les modules opérationnels sont aussi vus comme des objets ;
* La construction se fait avec une approche Yoyo (Top-down, Botton-Up).

**Adaptable :**

* UML n’impose pas de méthode de développement. C’est un standard de communication contenant un certain nombre d’artefacts utiles ;
* Il est donc possible d’intégrer l’utilisation d’UML à un processus de développement existant.

Pour toutes ces raisons nous avons choisi UML comme langage de conception lors de la réalisation de notre projet de fin d’étude.

La norme UML actuelle comporte 13 diagrammes. Cependant seuls 2diagrammes nous seront réellement utiles : Le diagramme des cas d’Utilisation, et le diagramme de classes.

Le diagramme de classes est le plus utile, c’est un formalisme pour représenter les concepts usuels de l’orienté objet. Le diagramme des cas d’utilisation permet de décrire les besoins de l’utilisateur final de l’application.

1. Règles de gestion

• Les étudiants peuvent s'inscrire à l'application via un formulaire d'inscription.

• L'administrateur est responsable de la confirmation de l'inscription des étudiants, ou de la rejeter.

• L'administrateur peut ajouter ou supprimer des comptes d'étudiants.

L'administrateur peut supprimer les publications d'étudiants

• Après confirmation et identification, chaque étudiant peut envoyer des publications, des messages et recevoir des notifications.

• Un étudiant peut suivre un autre étudiant.

• Un étudiant peut avoir un seul profil.

• Seul l'étudiant peut modifier son profil (ajout, suppression ou modification des informations).

• Seul l'étudiant peut supprimer ses propres publications.

• Les étudiants peuvent envoyer des messages uniquement à leurs amis ou à des groupes créés par l'administrateur.

• Seul l'étudiant peut recevoir des notifications.

• Un étudiant peut délivrer un ou plusieurs publications, ainsi que la suppression d'un post est permise.

• Un post concerne un seul étudiant, ce dernier peut le modifier, le supprimer ou le consulter à tout moment.

• Un post contient un ou plusieurs contenus (texte, image).

• Chaque contenu correspond à un seul post.

• Les étudiants peuvent modifier leur mot de passe à tout moment.

1. Diagrammes des cas d’utilisation

Les cas d’utilisation (use cases) décrivent sous forme d’actions et de réactions, le comportement d’un système du point de vue d’un utilisateur. Ils sont utiles lors de l’élaboration du cahier des charges ou du document de spécifications des besoins du logiciel. Ce diagramme comprend les acteurs, le système et les cas d’utilisation.

Gestion globale de l’application

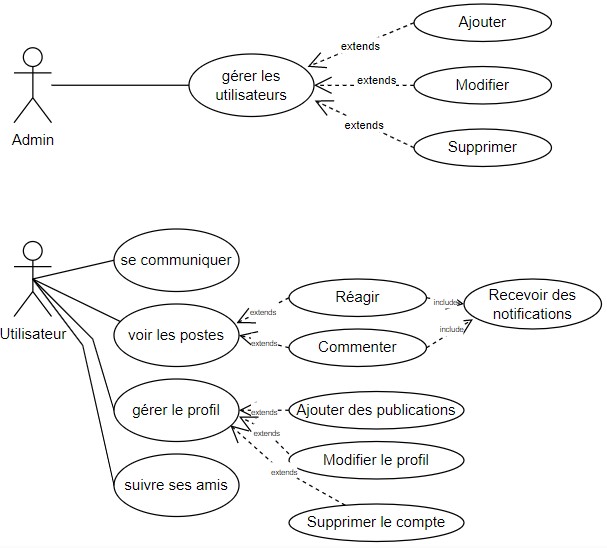
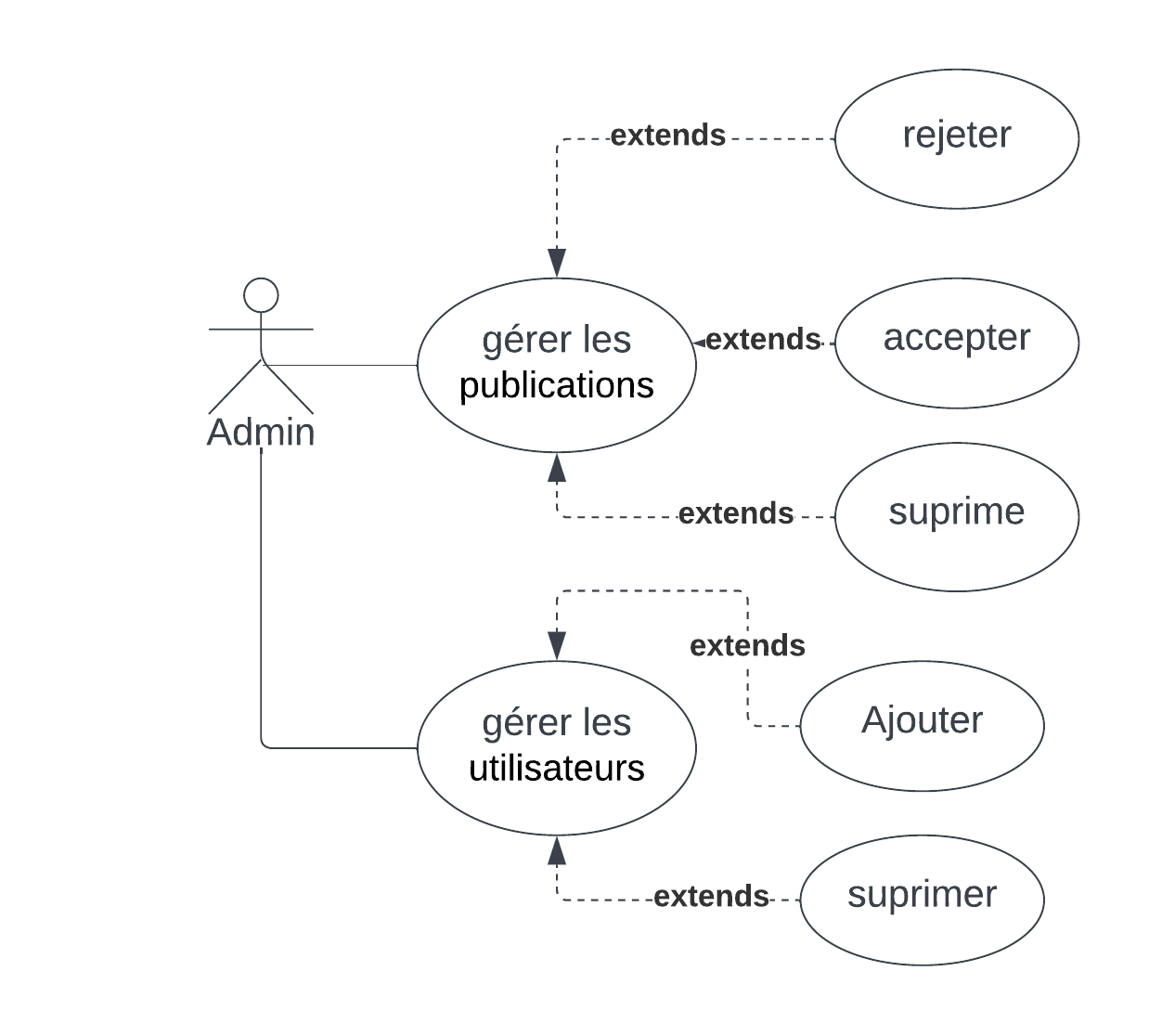


Figure 5Diagramme de cas d'utilisation de la gestion globale de l'application

Le diagramme ci-dessus décrit toutes les tâches principales à accomplir : les utilisateurs peuvent ajouter, modifier ou supprimer leurs propres publications, ainsi que commenter ou réagir aux publications des autres utilisateurs. Ils peuvent également recevoir des notifications pour les nouvelles publications, les nouvelles réactions ou les commentaires, et envoyer ou recevoir des messages individuels ou collectifs. De plus, les utilisateurs peuvent modifier ou supprimer leur profil à tout moment. Enfin, les tâches de l'administrateur sont également présentées dans le diagramme.

**Gestion des publications :**

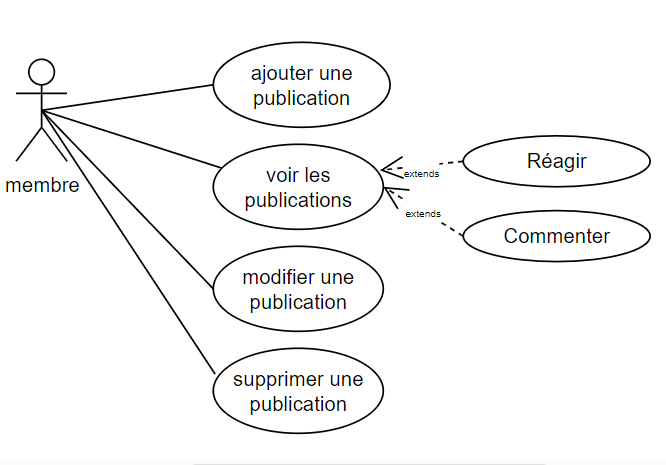
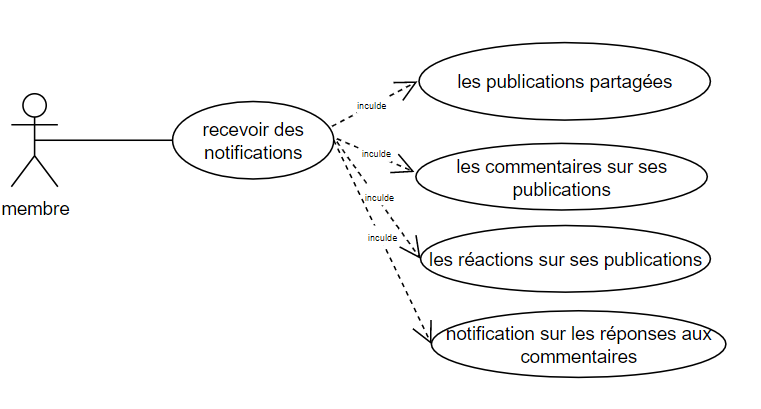


Figure 6 Diagramme des cas d'utilisation de la gestion des publications

Les membres peuvent ajouter, modifier ou supprimer des publications, voir les publications d'autres membres, commenter, répondre à un commentaire ou réagir à une publication.

**Gestion des Notifications :**



*Figure 7 Diagramme de cas d'utilisation de la gestion des notifications*

Recevoir des notifications sur les publications partagées, les réactions ou les commentaires sur ses publications, ainsi que les réactions ou les réponses des autres étudiants sur ses commentaires dans les publications des membres.

**Gestion des Messages :**

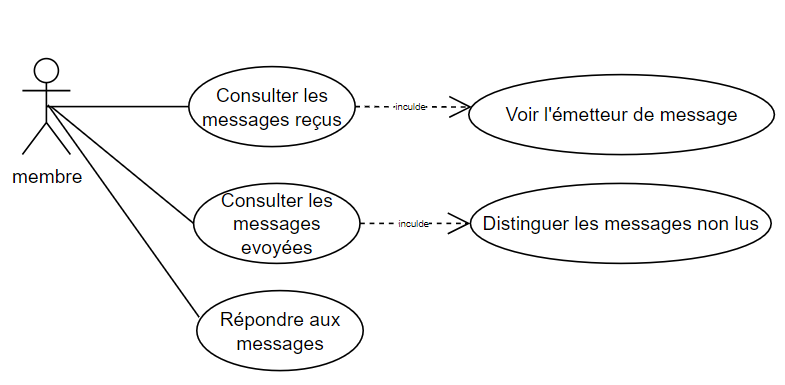


Figure 8Diagramme de cas d'utilisation de la gestion des messages

Un membre peut consulter les messages envoyés et reçus, répondre aux messages, voir l'émetteur du message et distinguer les non-messages.

1. Diagramme de classes

Le diagramme de classes d’UML saisit la structure statique d'un système en montrant les objets dans le système, les relations entre les objets, les attributs et les opérations qui caractérisent chaque classe d'objets. C'est le plus important des modèles d’UML.

Il est totalement indispensable quel que soit la méthode utilisée, généralement le diagramme de classes défini la structure interne et les liens entre les différentes classes du système mis en œuvre.

Nous examinerons plus en détails ce diagramme dans le schéma suivant :



Figure 9 Diagramme de classe

1. Le Diagramme entité association :

Le modèle Entité-Association (E-A) est un modèle conceptuel de données utilisé en génie logiciel pour décrire les entités et les relations qui existent entre elles dans un domaine donné. Le modèle E-A repose sur la représentation des entités et de leurs caractéristiques (ou attributs), ainsi que sur la définition des relations entre ces entités.

Le modèle E-A est basé sur deux concepts clés :

Les entités : ce sont des objets ou des concepts du monde réel que nous représenterons dans notre modèle. Les entités sont caractérisées par des attributs, qui représentent les caractéristiques ou les propriétés de ces entités.

Les relations : ce sont les liens qui existent entre les entités du modèle. Les relations peuvent être de différents types : un-à-un, un-à-plusieurs ou plusieurs-à-plusieurs.

Le modèle E-A est souvent utilisé pour décrire des schémas de base de données relationnelles. Pour ce faire, les entités sont représentées sous forme de tables relationnelles, et les relations sont représentées sous forme de clés étrangères qui reposent sur les tables entre elles.

Le modèle E-A est souvent utilisé pour décrire des systèmes d'information complexes. Il permet de clarifier la structure de données et les relations entre les données, facilitant ainsi le développement de logiciels qui utilisent ces bases de données. Il est également utile pour la maintenance et l'évolution de ces bases de données au fil du temps.

En résumé, le modèle Entité-Association est un modèle conceptuel de données qui permet de décrire les entités et les relations qui existent entre elles dans un domaine donné. Il est souvent utilisé pour décrire des schémas de base de données relationnelles et pour clarifier la structure de données dans des systèmes d'information complexes. Nous examinerons plus en détails ce diagramme dans le schéma suivant :

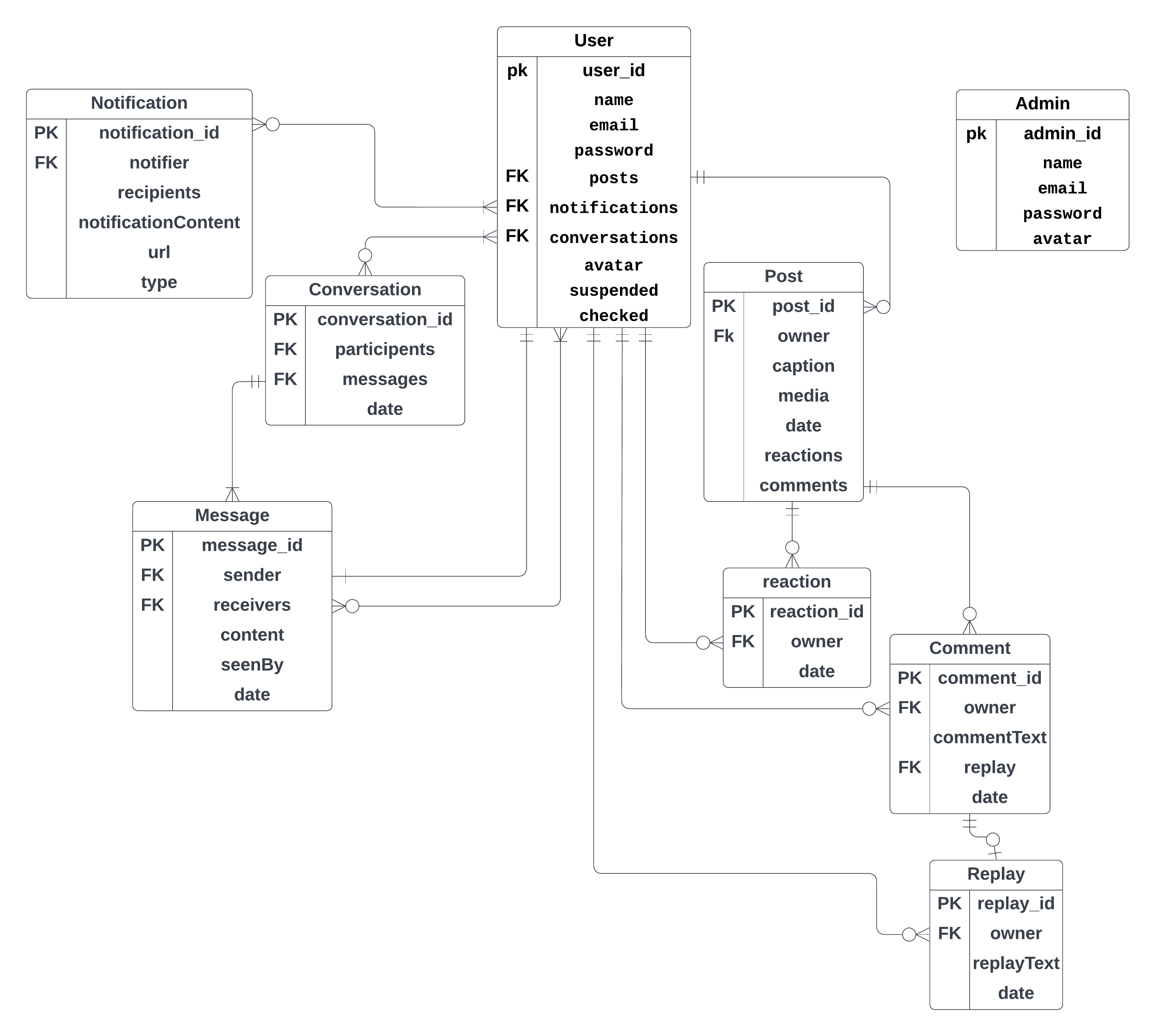


Figure 10 Diagramme d'**entité association**

**L'entité Utilisateur**

L' entité Utilisateur représente les utilisateurs de notre application. Elle contient les attributs suivants:

* nom: une chaîne de caractères qui représente le nom de l'utilisateur.
* email: une chaîne de caractères qui représente l'adresse e-mail de l'utilisateur. Elle doit être unique.
* Mot\_de\_passe: une chaîne de caractères qui représente le mot de passe de l'utilisateur. Il doit être haché en utilisant un algorithme sécurisé.
* avatar: une chaîne de caractères qui représente l'URL de la photo de profil de l'utilisateur.
* publications: un tableau d'objets Post qui représentent les publications de l'utilisateur sur la plateforme.
* conversations: un tableau d'objets Conversation qui représentent les conversations de l'utilisateur avec d'autres utilisateurs sur la plateforme.
* notifications: un tableau d'objets Notification qui représentent les notifications reçues par l'utilisateur.
* abonnements: un tableau d'objets Utilisateur qui représentent les utilisateurs que l'utilisateur suit.
* abonnés: un tableau d'objets Utilisateur qui représentent les utilisateurs qui suivent l'utilisateur.
* suspendu: un booléen qui indique si le compte de l'utilisateur a été suspendu ou non. Il est par défaut à false.
* vérifié: un booléen qui indique si le compte de l'utilisateur a été vérifié par les administrateurs ou non. Il est par défaut à false.

**L'entité Publication**

Publication représente une publication créée par un utilisateur. Elle possède les propriétés suivantes :

* Propriétaire : l'ID de l'utilisateur qui a créé la publication
* Légende : la légende de la publication (facultatif)
* Média : l'URL de tout média attaché à la publication (facultatif)
* Réactions : un tableau d'objets Réaction associés à la publication
* Commentaires : un tableau d'objets Commentaire associés à la publication
* Date : la date et l'heure de création de la publication

**L' entité Réaction**

Réaction représente une réaction à une publication. Elle possède les propriétés suivantes :

* Propriétaire : l'ID de l'utilisateur qui a créé la réaction
* Date : la date et l'heure de création de la réaction

**L' entité Commentaire**

Commentaire représente un commentaire sur une publication. Elle possède les propriétés suivantes :

* Propriétaire : l'ID de l'utilisateur qui a créé le commentaire
* texteCommentaire : le texte du commentaire
* Date : la date et l'heure de création du commentaire
* Réponses : un tableau d'objets Réponse associés au commentaire (facultatif)

**L' entité Réponse**

Réponse représente une réponse à un commentaire. Elle possède les propriétés suivantes :

* Propriétaire : l'ID de l'utilisateur qui a créé la réponse
* Texte Réponse : le texte de la réponse
* Date : la date et l'heure de création de la répons

**L' entité Message**

Message représente un message envoyé entre utilisateurs. Elle possède les propriétés suivantes :

* Expéditeur : l'ID de l'utilisateur qui a envoyé le message
* Destinataires : un tableau d'objets Utilisateur qui ont reçu le message
* Contenu : le texte du message
* Vu Par : un tableau d'objets Utilisateur qui ont vu le message
* Date : la date et l'heure d'envoi du message

**L' entité Conversation**

Conversation représente une conversation entre utilisateurs. Elle possède les propriétés suivantes :

* Participants : un tableau d'objets Utilisateur qui participent à la conversation
* Messages : un tableau d'objets Message associés à la conversation

**L' entité Notification**

Notification représente une notification envoyée à un utilisateur. Elle possède les propriétés suivantes :

* Émetteur : l'ID de l'utilisateur qui a créé la notification
* Destinataires : un tableau d'objets Utilisateur qui ont reçu la notification
* Type : le type de notification (par exemple "j'aime", "commentaire", "nouvel abonné")
* Contenu : le contenu de la notification
* Date : la date et l'heure de création de la notification"

1. Chapitre 5 : Présentation de l’application
2. Introduction

Ce chapitre vise l'étude et la description détaillée de notre application, celui-là est divisé en cinq espaces: inscription, connexion, newsfeed, messages, profile, et administrateur. Pour mieux éclaircir les choses, cette présentation sera accompagnée avec une suite de captures d'écran. Nous essayerons de donner le maximum important tandis que le reste sera disponible dans l'annexe.

1. L’accueil de l’application

Notre application de réseau social est conçue pour être conviviale et intuitive, offrant une expérience sans interruption pour les étudiants et les administrateurs. Lorsque l'application est lancée, les utilisateurs sont présentés à une page d'accueil qui est à la fois visuellement attrayante et fonctionnelle. La page d'accueil comprend le nom de l'université et le logo de notre école, qui sert de représentation visuelle de notre institution.

Pour assurer un accès sans interruption et une navigation facile, la page d'accueil comprend également un groupe de trois boutons sur le côté droit de l'écran. Les deux premiers boutons permettent aux étudiants d'accéder à la page de connexion, où ils peuvent se connecter en utilisant leurs identifiants uniques et accéder à leur fil d'actualité personnel, leurs messages et leur profil. Le troisième bouton permet aux nouveaux utilisateurs de rejoindre la plateforme en entrant leur nom, leur adresse e-mail, leur avatar et leur mot de passe. De plus, le troisième bouton permet aux administrateurs de se connecter à la plateforme, où ils peuvent gérer les utilisateurs de l'application.

Dans l'ensemble, la page d'accueil de notre application de réseau social a été conçue en gardant à l'esprit la simplicité et la fonctionnalité, offrant aux utilisateurs une expérience sans interruption qui leur permet de se connecter, de communiquer et de collaborer facilement avec d'autres utilisateurs sur la plateforme.

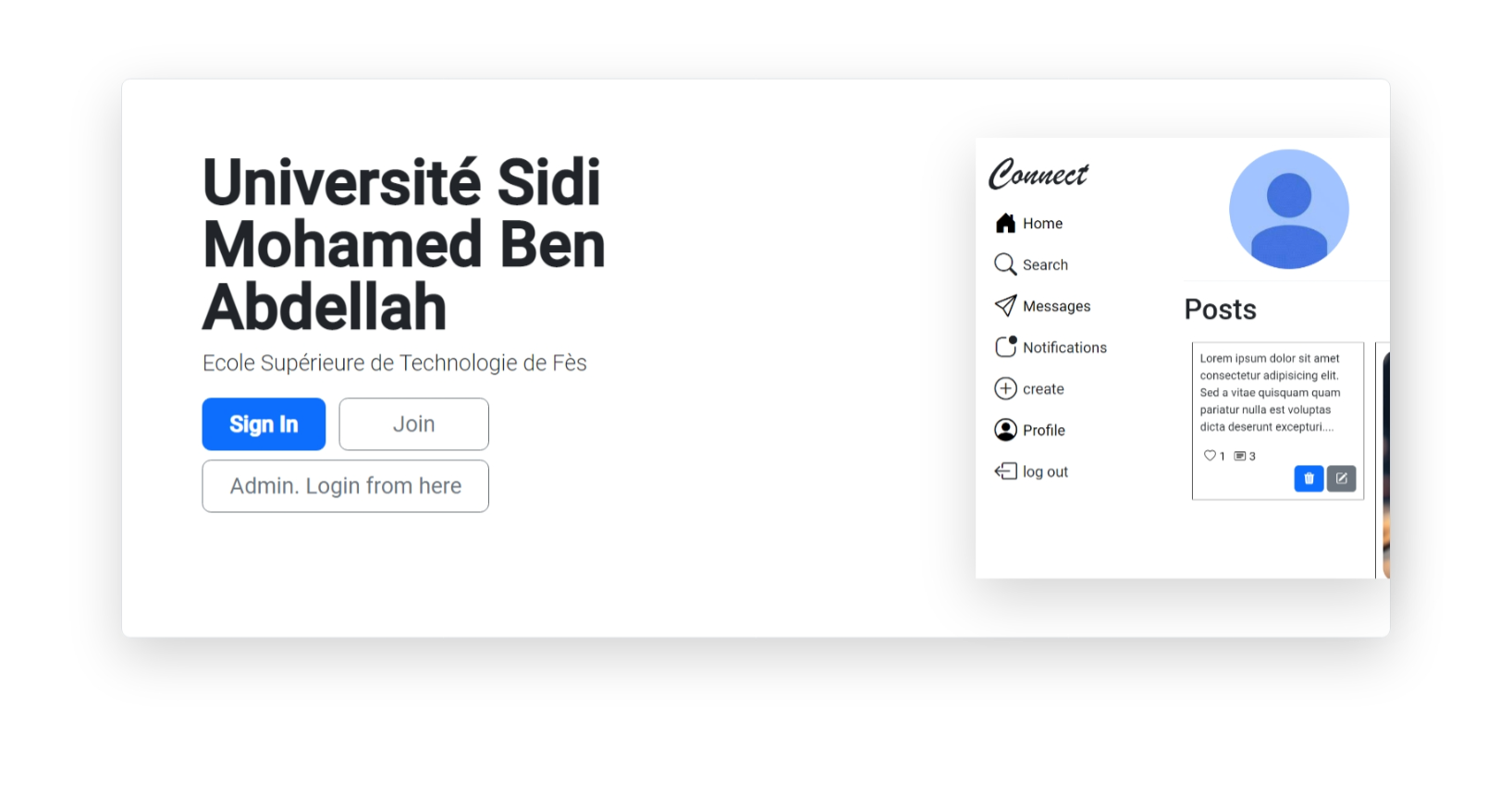


Figure 11 Page d'acceuil de l'application

1. Etape d’inscription

Pour créer un compte sur notre application de médias sociaux, les utilisateurs doivent remplir un formulaire d'inscription en trois étapes.

Étape 1: Informations générales Cette étape comprend les champs obligatoires suivants: nom, avatar, adresse e-mail et mot de passe. Pour garantir l'unicité des informations saisies, nous vérifions si l'adresse e-mail saisie n'a pas déjà été utilisée pour créer un compte. En outre, un mot de passe fort est requis pour garantir la sécurité du compte de l'utilisateur.

Étape 3:l'utilisateur doit accepter les conditions d'utilisation et cliquer sur le bouton "S'inscrire".

Après avoir terminé l'inscription, l'utilisateur sera redirigé vers la page de connexion et recevra un e-mail de confirmation. L'administrateur examinera ensuite la demande d'inscription et approuvera le compte si tout est en ordre. Une fois approuvé, l'utilisateur pourra se connecter et profiter de toutes les fonctionnalités de notre application.

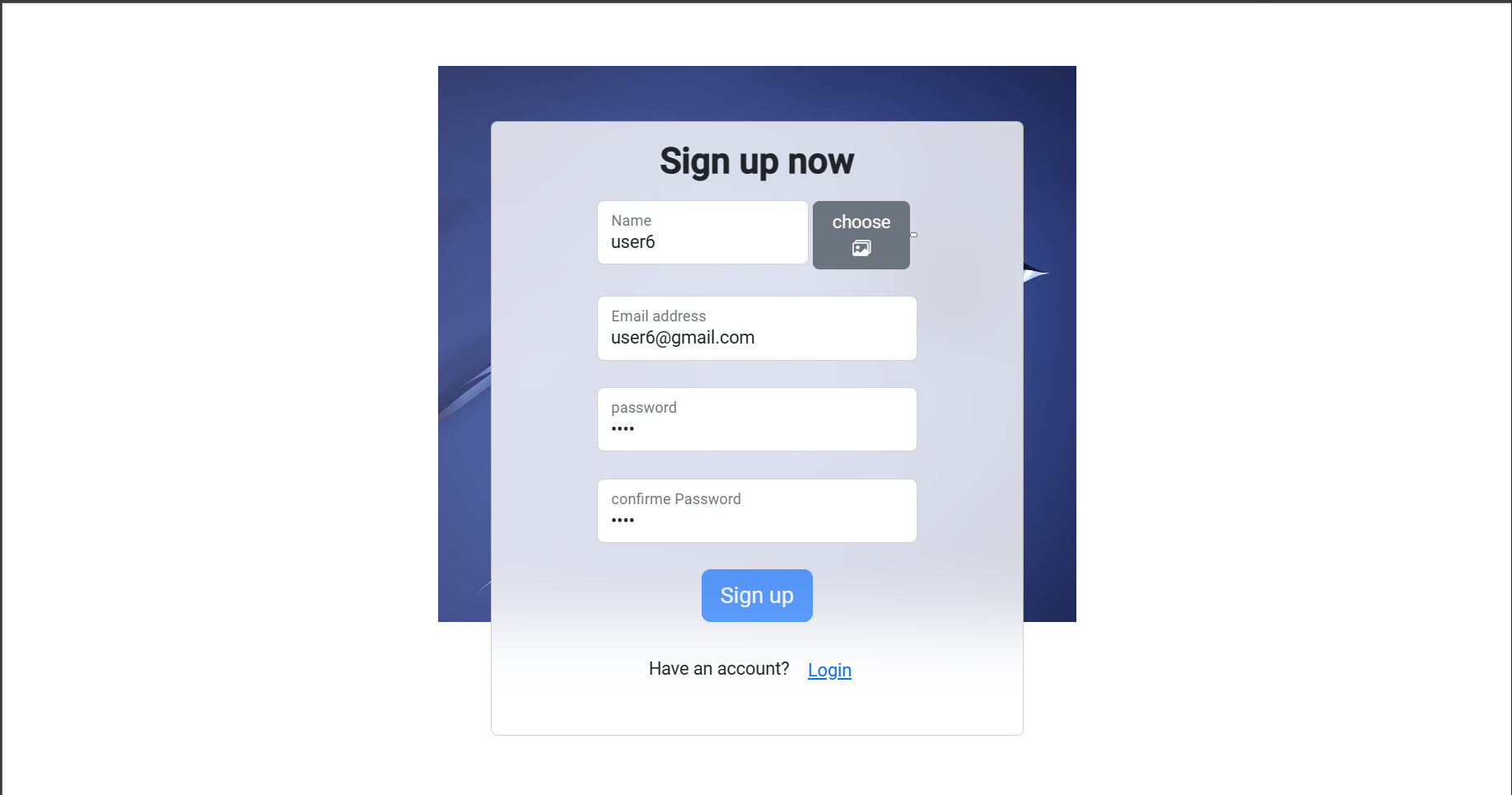


Figure 12Page d'inscription des étudiants

1. Etape d’authentification

L'authentification de l'utilisateur est essentielle pour assurer la sécurité des données et prévenir les abus. L'interface d'authentification sera presque la même pour tous les utilisateurs, qu'il s'agisse d'étudiants ou d'administrateurs. Lorsqu'un utilisateur entre son nom d'utilisateur et son mot de passe, plusieurs vérifications sont effectuées pour garantir que l'accès n'est autorisé qu'aux utilisateurs légitimes. Si le nom d'utilisateur ou le mot de passe est incorrect, un message d'erreur apparaîtra pour informer l'utilisateur de l'erreur et l'inviter à réessayer. Si les informations sont correctes mais que l'inscription n'a pas encore été confirmée par l'administrateur, un message apparaîtra pour informer l'utilisateur que l'accès à son compte est en attente de confirmation. De plus, si l'administrateur a suspendu leur compte, ils ne pourront pas accéder à la plateforme tant que l'administrateur ne l'aura pas réactivé.

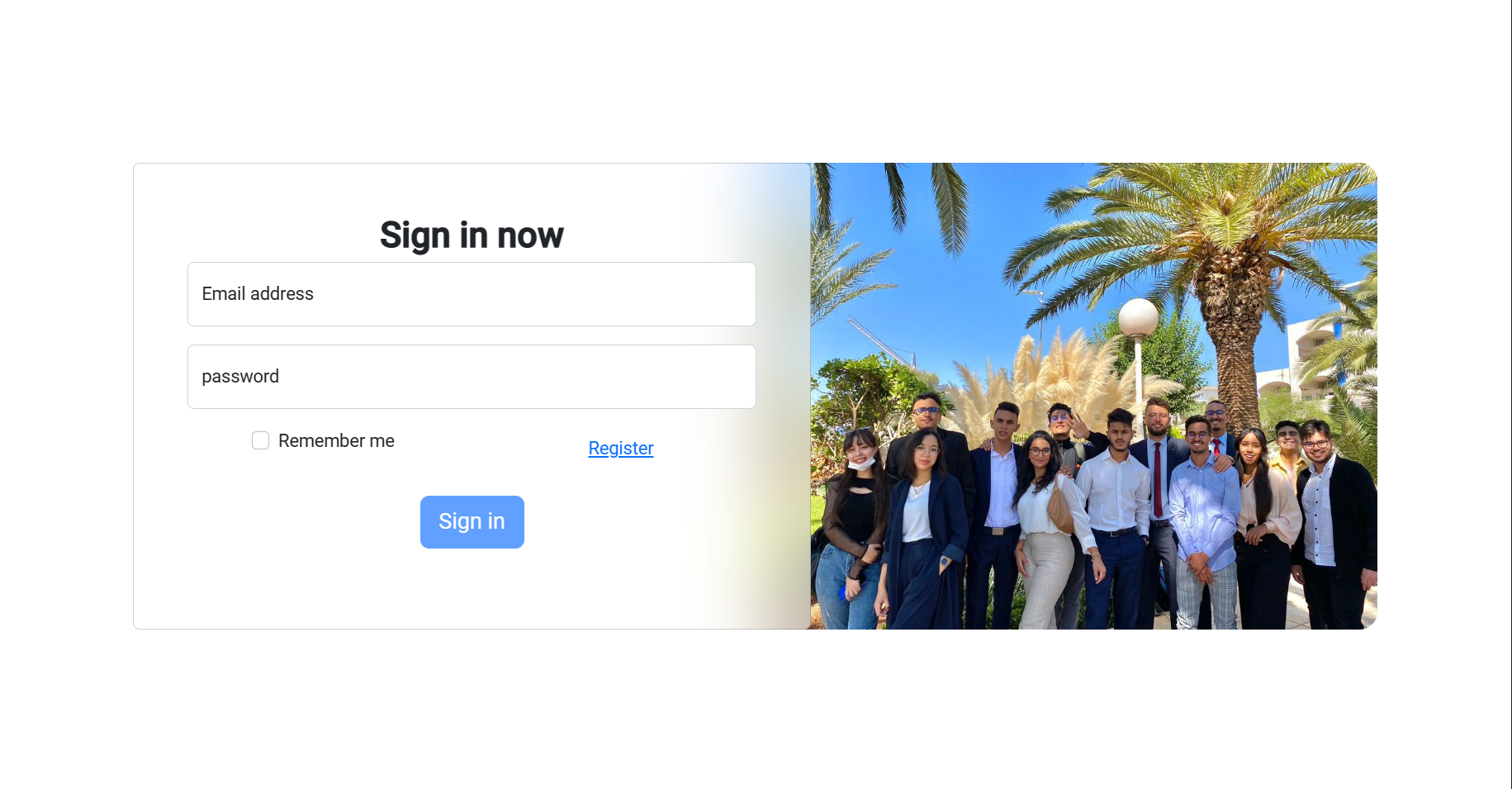


Figure 13Page d'authentification des étudiants

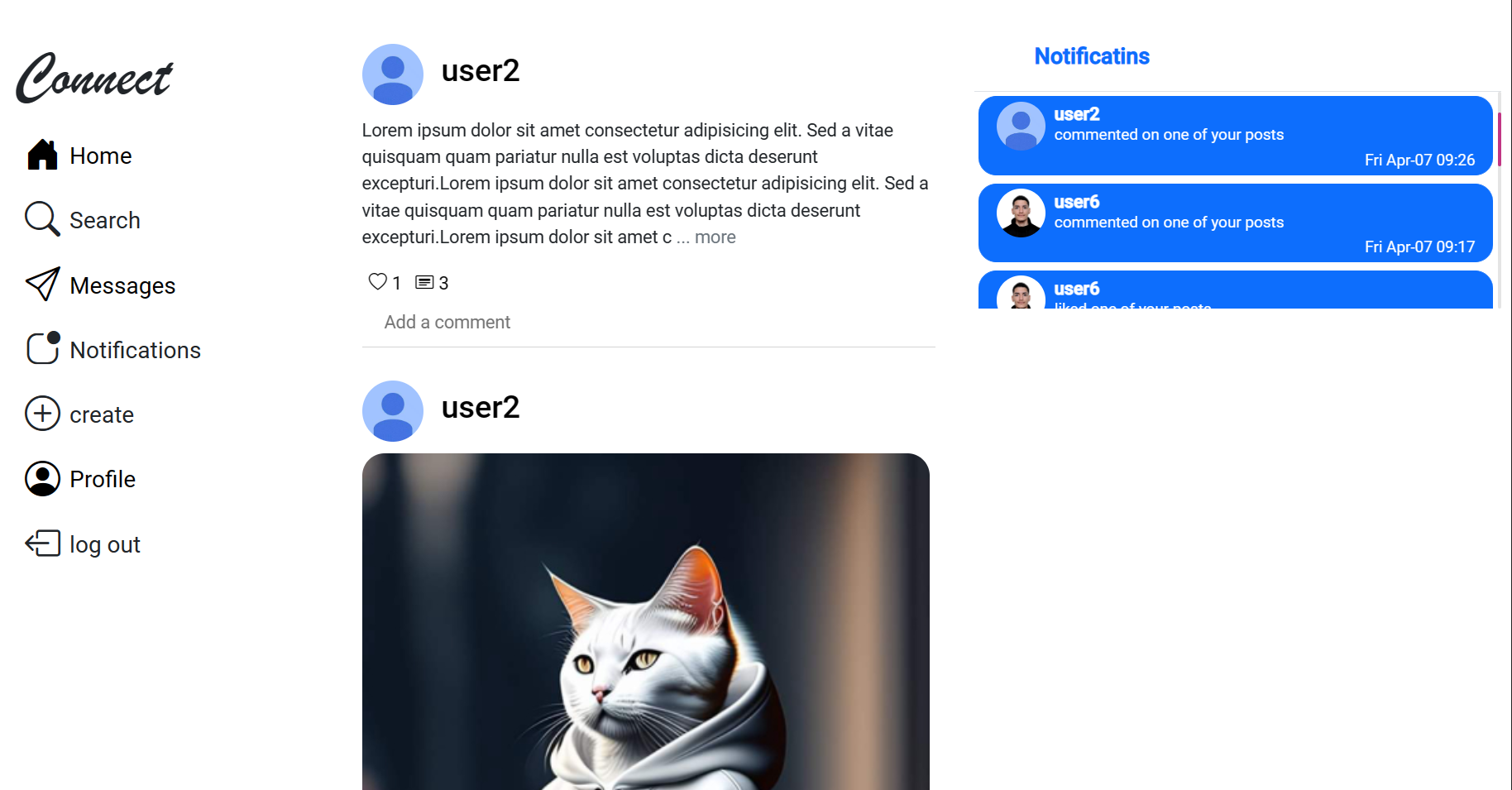
1. Espace utilisateur
2. Page d'actualité

La section "Fil d'actualités" de notre application de médias sociaux est le centre d'activité principal des utilisateurs, où tous les messages des utilisateurs sont répertoriés du plus récent au plus ancien. La mise en page est conçue pour être intuitive et conviviale, avec une barre de navigation claire et accessible sur le côté gauche de l'écran. Cette barre de navigation comprend des liens vers les fonctionnalités "Fil", "Recherche", "Messages","Crée" et "Déconnexion", permettant aux utilisateurs d'accéder rapidement et facilement aux fonctionnalités dont ils ont besoin.

Au milieu de l'écran, l'utilisateur est présenté avec une liste dynamique et engageante de messages d'autres utilisateurs. Cette liste comprend une variété de types de contenu, y compris du texte, des images, ou les deux, et permet aux utilisateurs d'interagir entre eux de diverses façons. Les utilisateurs peuvent aimer, commenter et partager des messages, ainsi que suivre d'autres utilisateurs pour voir leurs derniers Publications.

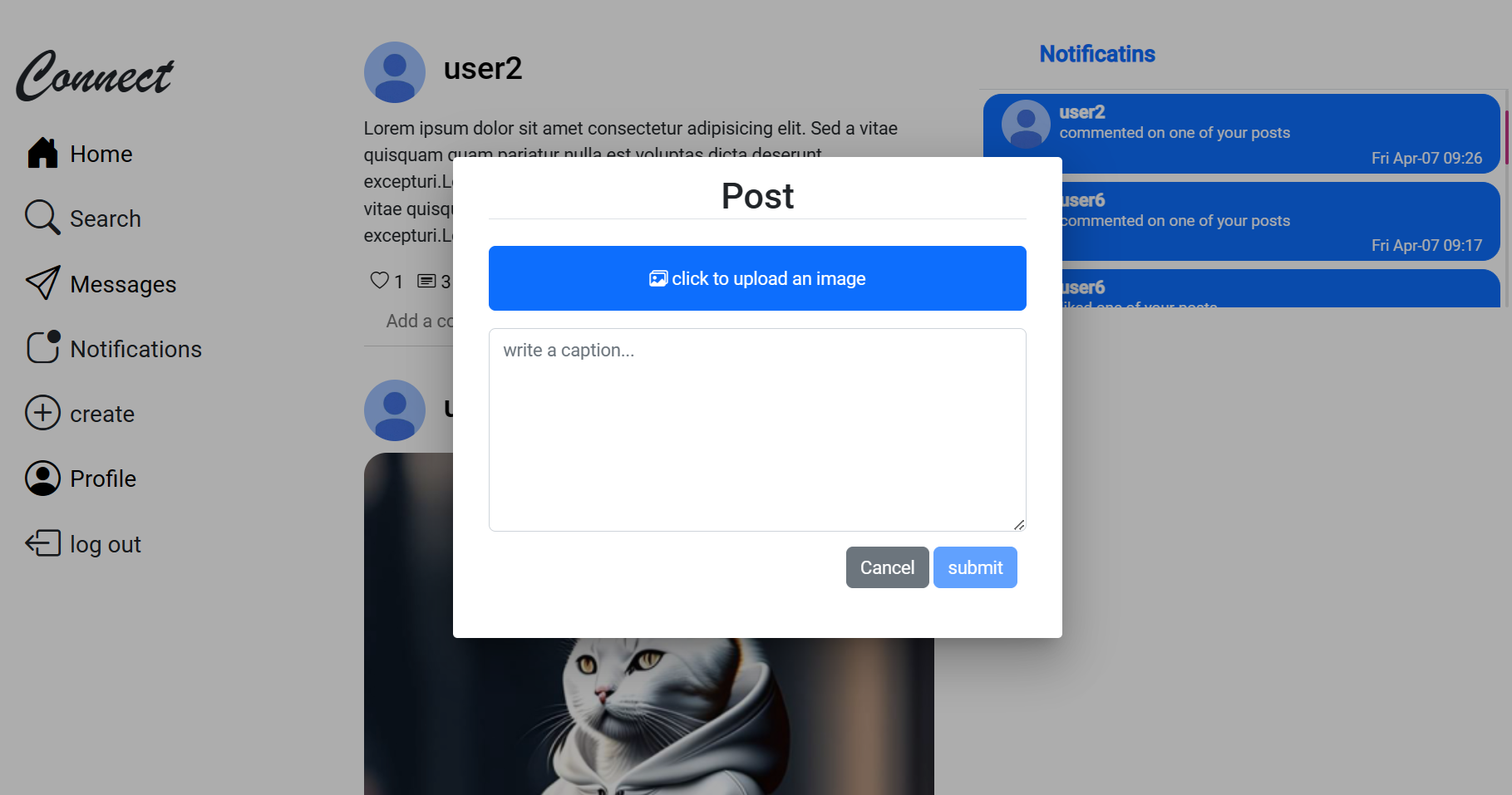
Sur le côté droit de l'écran, les utilisateurs peuvent trouver un petit conteneur étiqueté "notifications". Ce conteneur offre un moyen rapide et facile de suivre les derniers j'aime, commentaires et nouveaux messages d'utilisateurs qu'ils suivent. Cette fonctionnalité est conçue pour aider les utilisateurs à rester à jour sur les derniers événements de leur réseau social et encourager une plus grande participation et engagement dans la communauté.

Dans l'ensemble, la section "Fil d'actualités" de notre application de médias sociaux est conçue pour être une plateforme engageante et conviviale pour l'interaction sociale et la création de contenu. Avec sa navigation intuitive, sa liste dynamique de messages et sa fonctionnalité de notification pratique, Nous croyons que notre application offrira aux utilisateurs une expérience de média social vraiment unique et agréable.



1. Nouvelle publication

Lorsque l'utilisateur clique sur "Créer" dans la barre de navigation, une fenêtre contextuelle apparaît, offrant deux champs d'entrée pour sélectionner une image et écrire une légende ou un texte. Deux boutons sont situés en bas de la fenêtre contextuelle, l'un pour soumettre la nouvelle publication et l'autre pour annuler.



1. Mes messages

Lorsque l'utilisateur navigue vers la section "messages" de notre application, il voit une vue divisée en deux colonnes : la colonne de gauche affiche une liste de ses conversations avec d'autres utilisateurs ainsi qu'une barre de recherche pour trouver un ami ou un utilisateur, et la colonne de droite affiche le contenu de la conversation sélectionnée.

Dans la colonne de gauche, l'utilisateur peut voir une liste de toutes ses conversations avec d'autres utilisateurs, triées par date, avec les plus récentes apparaissant en haut. Chaque conversation est représentée par une photo de profil et le nom d'utilisateur de l'autre utilisateur, ainsi qu'un aperçu du dernier message échangé. Si la conversation a des messages non lus, un petit point bleu est affiché à côté du nom d'utilisateur pour indiquer qu'il y a des nouveaux messages.

Lorsque l'utilisateur clique sur une conversation, la colonne de droite affiche l'intégralité de la conversation, en commençant par le message le plus récent en bas et en remontant vers les messages plus anciens. Chaque message est affiché avec la photo de profil et le nom d'utilisateur de l'expéditeur, ainsi que le contenu du message et l'heure à laquelle il a été envoyé. L'utilisateur peut également envoyer un nouveau message en tapant dans le champ de saisie de texte en bas de la colonne de droite et en appuyant sur le bouton d'envoi.

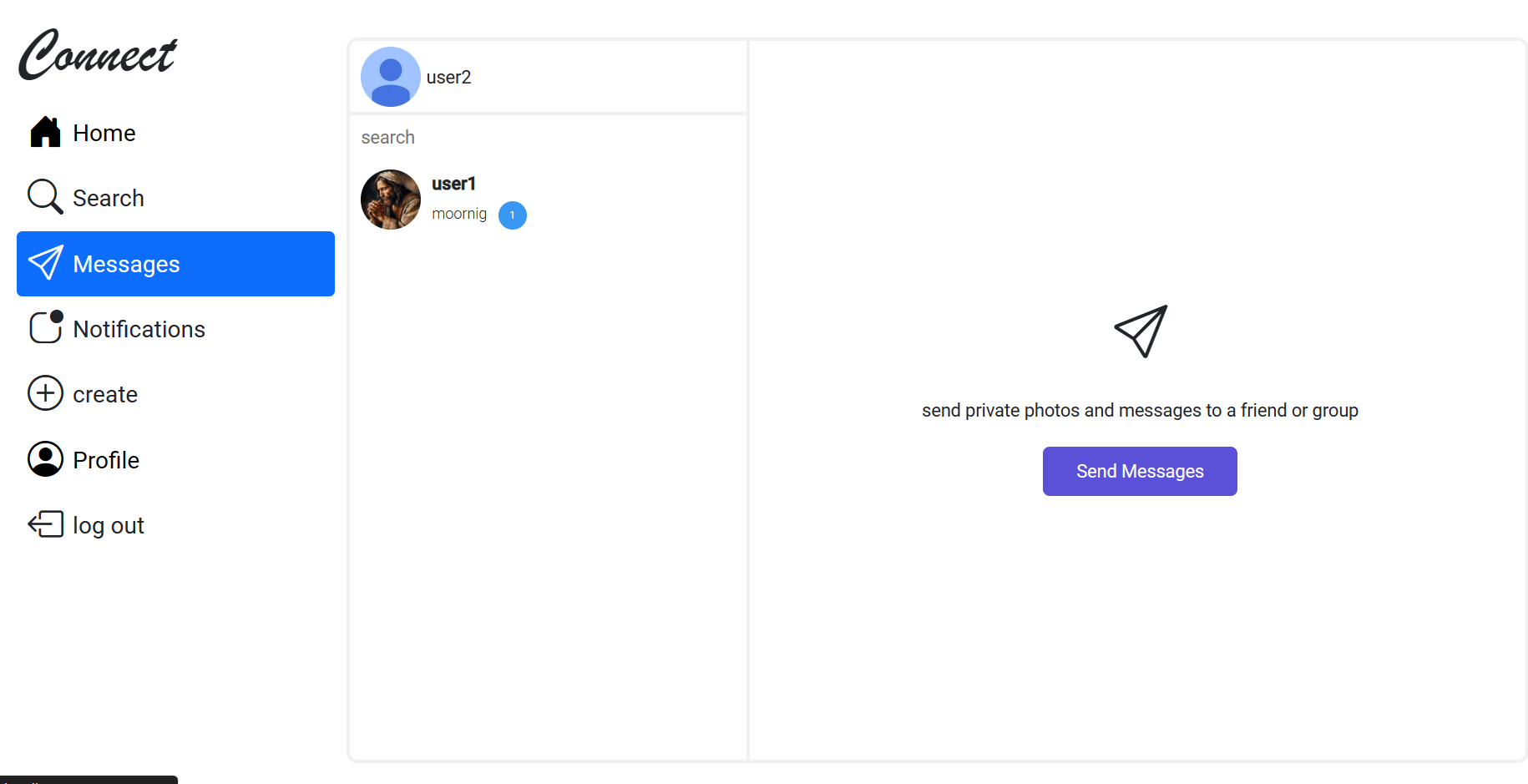
Dans l'ensemble, la vue est claire et facile à naviguer, permettant à l'utilisateur de facilement suivre ses conversations et rester en contact avec d'autres utilisateurs.

Figure 14Page "Messages"

1. Mon Profile

La page "Mon Profile" est divisée en deux rangées, similaire à la page de profil dans Instagram. La rangée supérieure est à son tour divisée en deux colonnes : la colonne de droite affiche l'image de profil tandis que celle de gauche est divisée en deux rangées. La première rangée affiche le nom de l'utilisateur et un bouton qui lui permet de modifier son profil. Sous cette rangée, l'utilisateur peut voir le nombre de ses publications, ainsi que le nombre d'utilisateurs qu'il suit et le nombre de ses abonnés.

La ligne inférieure affiche les publications de l'utilisateur, et chaque publication contient le nombre de mentions "J'aime" et de commentaires, ainsi que deux boutons, un pour supprimer et un pour éditer la publication. Dans l'ensemble, la vue est claire et facile à naviguer, permettant à l'utilisateur de visualiser facilement ses publications et d'interagir avec elles.

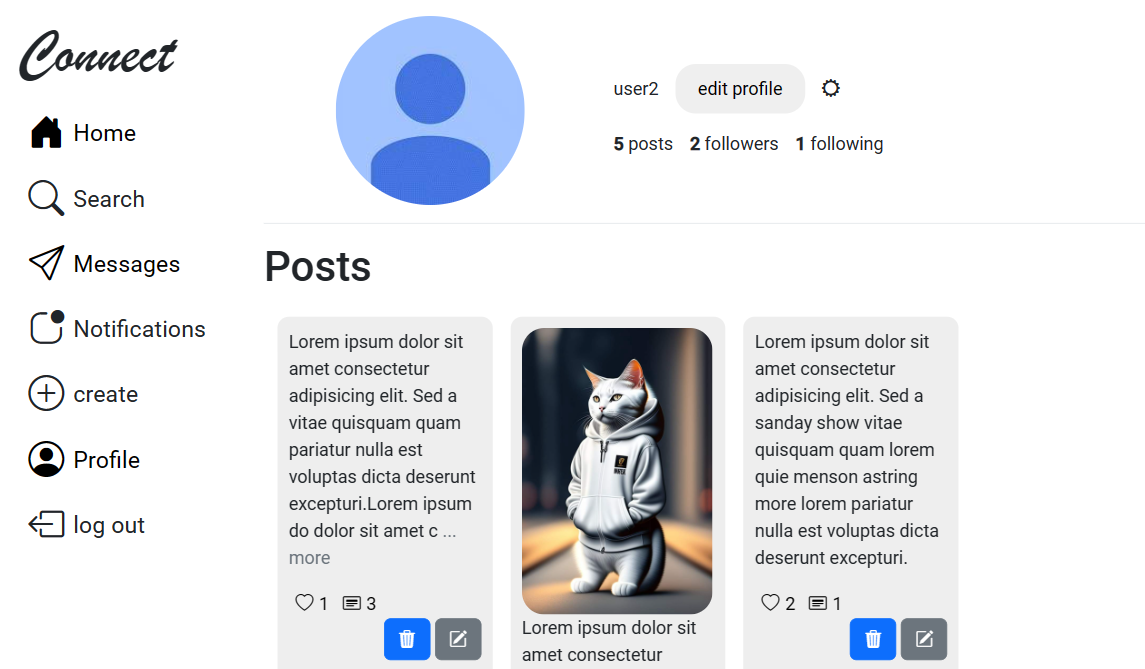


Figure 15Page Profile

1. Espace administrateur
2. Page utilisateurs

Lorsque l'administrateur se connecte avec succès, il sera redirigé vers la page "tableau de bord". La vue est divisée en deux colonnes. La première colonne est une barre de navigation qui contient des liens vers "utilisateurs", "publications", "nouveaux utilisateurs", "nouvelles publications" et un bouton "se déconnecter". La deuxième colonne contient une barre de recherche et une liste d'utilisateurs.

Chaque utilisateur de la liste a deux boutons à côté de son nom-adresse e-mail, l'un pour suspendre l'utilisateur et l'autre pour le supprimer. La barre de recherche permet à l'administrateur de rechercher un utilisateur spécifique en entrant son nom ou son adresse e-mail. Cela facilite la gestion de la base de données des utilisateurs par l'administrateur et la prise de mesures contre tout utilisateur malveillant ou inapproprié.

Dans l'ensemble, la vue offre une interface propre et facile à utiliser pour l'administrateur pour gérer les utilisateurs de la plateforme. La barre de navigation permet une navigation facile entre les différentes sections, et la barre de recherche simplifie le processus de recherche d'utilisateurs spécifiques. Les boutons à côté du nom de chaque utilisateur permettent à l'administrateur de prendre rapidement et facilement les mesures nécessaires, telles que la suspension ou la suppression d'un utilisateur.



1. Page nouveaux utilisateurs

Lorsque l'administrateur sélectionne le lien "new users" dans la barre de navigation, il est redirigé vers la page "dashboard/new-users". La page est divisée en deux colonnes. La colonne de gauche contient la barre de navigation, tandis que la colonne de droite affiche une liste de nouveaux utilisateurs qui se sont inscrits sur la plateforme. Chaque nouvel utilisateur de la liste est représenté par son avatar, son nom et son adresse e-mail. En dessous de chaque utilisateur, il y a deux boutons : "refuser" et "accepter". L'administrateur peut cliquer sur le bouton "refuser" pour refuser l'inscription de l'utilisateur et sur le bouton "accepter" pour approuver son inscription et l'ajouter à la plateforme. La vue fournit un moyen simple et efficace pour l'administrateur de gérer les nouveaux utilisateurs sur la plateforme. Les boutons sous chaque nouvel utilisateur permettent à l'administrateur de prendre rapidement et facilement les actions nécessaires, telles que l'acceptation ou le rejet d'une inscription d'utilisateur. La barre de navigation permet une navigation facile entre les différentes sections, telles que "utilisateurs" et "publications". Dans l'ensemble, la section "nouveaux utilisateurs" fournit un outil essentiel à l'administrateur pour gérer la base de données des utilisateurs et assurer la sécurité et la qualité de la plateforme.

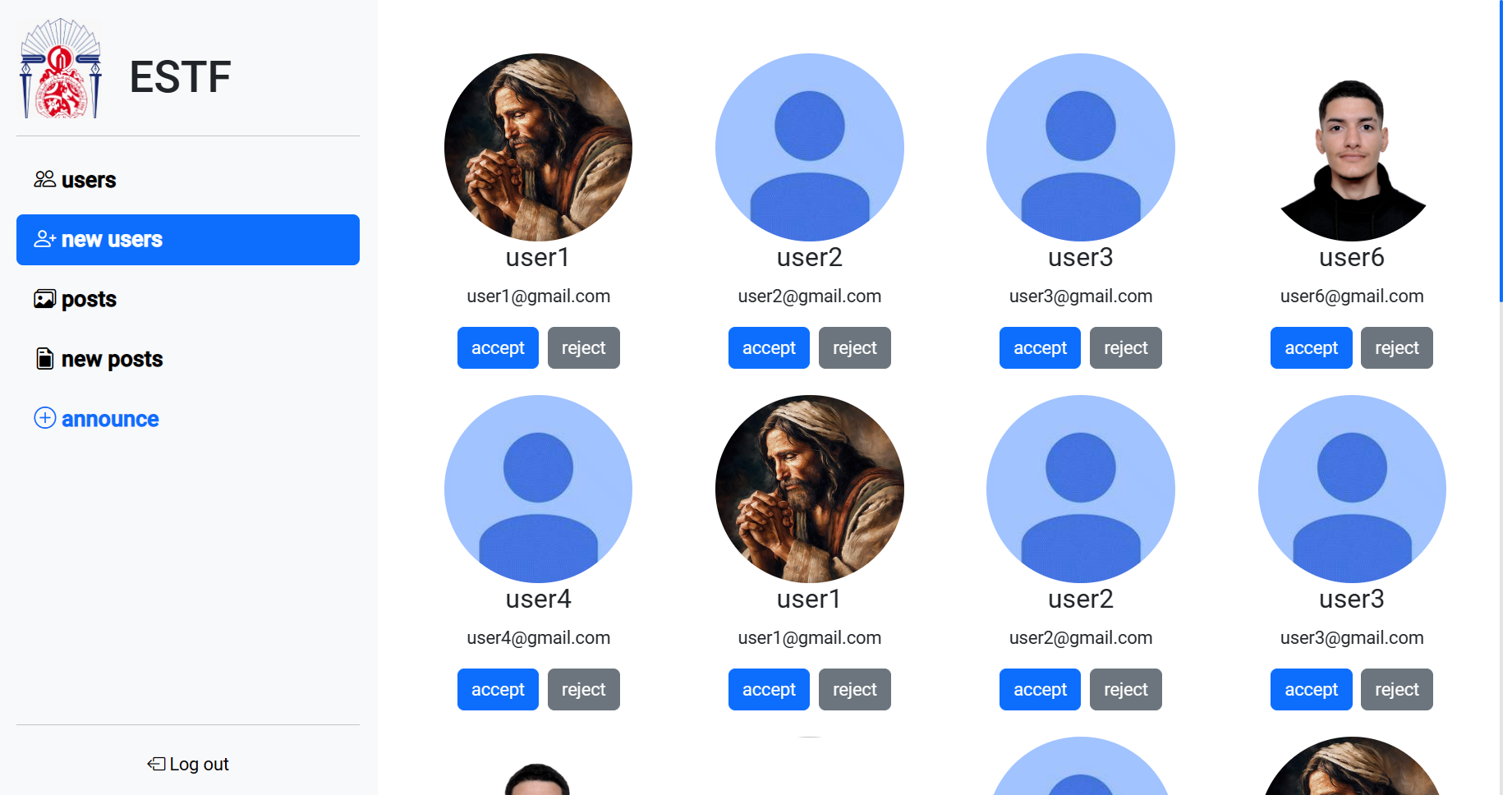


Figure 16Page nouveaux utilisateurs

1. Page Publications

Lorsque l'administrateur sélectionne le lien "posts" dans la barre de navigation, il est redirigé vers la page "dashboard/posts" qui est divisée en deux colonnes. La colonne de gauche contient la barre de navigation, tandis que la colonne de droite affiche une liste de posts. Chaque post de la liste affiche le nom de l'auteur, la date de création et le nombre de likes et de commentaires qu'il a reçus. En bas de chaque post, il y a un bouton de suppression qui permet à l'administrateur de supprimer le post de la plateforme s'il contient un contenu inapproprié ou offensant. La vue offre une interface facile à utiliser pour que l'administrateur puisse gérer les posts sur la plateforme. Le bouton de suppression à côté de chaque post simplifie le processus de suppression de tout post qui viole les politiques de la plateforme.

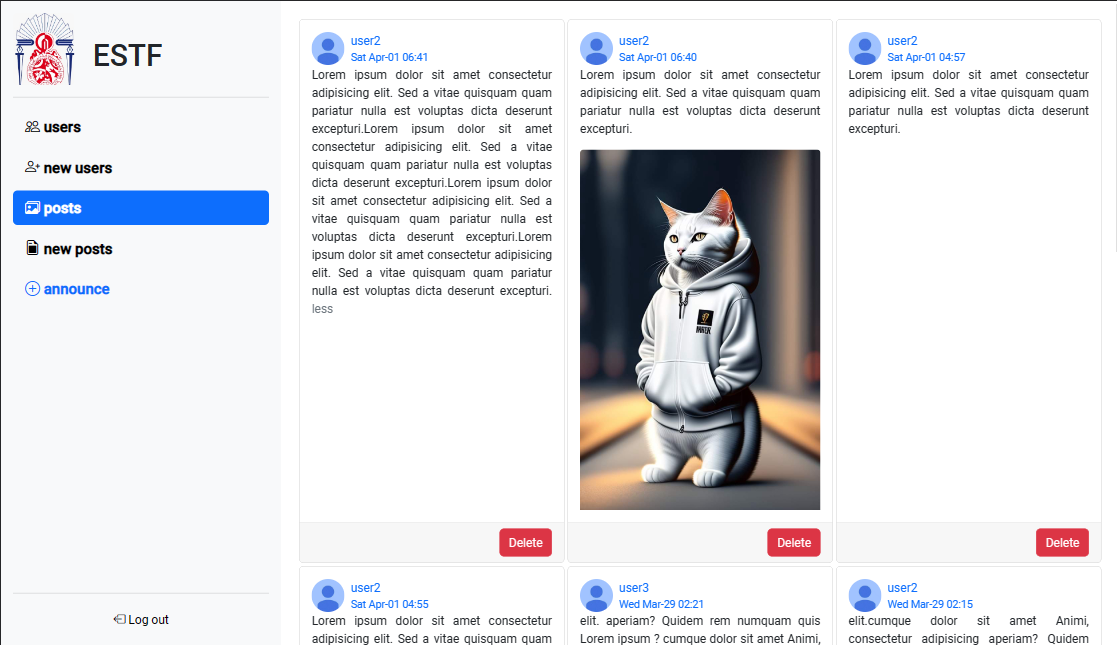


Figure 17page Publications

1. Page nouveaux publications

Lorsque l'administrateur clique sur le lien "new posts" dans la barre de navigation, il est dirigé vers la page "dashboard /new-posts", qui est divisée en deux colonnes.

La colonne de gauche contient la barre de navigation, tandis que la colonne de droite affiche une liste des nouveaux messages. Chaque message de la liste affiche des informations telles que l'auteur, la date de création et le contenu (texte ou image). En dessous de chaque publication, il y a deux boutons : "rejeter" et "accepter".

L'administrateur peut cliquer sur le bouton "rejeter" pour refuser la publication et l'empêcher d'être publiée sur la plateforme, ou cliquer sur le bouton "accepter" pour approuver la publication et l'ajouter à la plateforme.

La vue fournit une interface facile à utiliser permettant à l'administrateur de gérer les nouveaux messages sur la plate-forme. Les boutons "rejeter" et "accepter" sous chaque publication permettent à l'administrateur de prendre rapidement et facilement les mesures nécessaires. La barre de navigation permet une navigation facile entre les différentes sections, telles que "utilisateurs" et "messages".

Dans l'ensemble, la section "new posts" fournit un outil essentiel à l'administrateur pour garantir la qualité du contenu de la plateforme et empêcher la publication de messages inappropriés ou offensants.

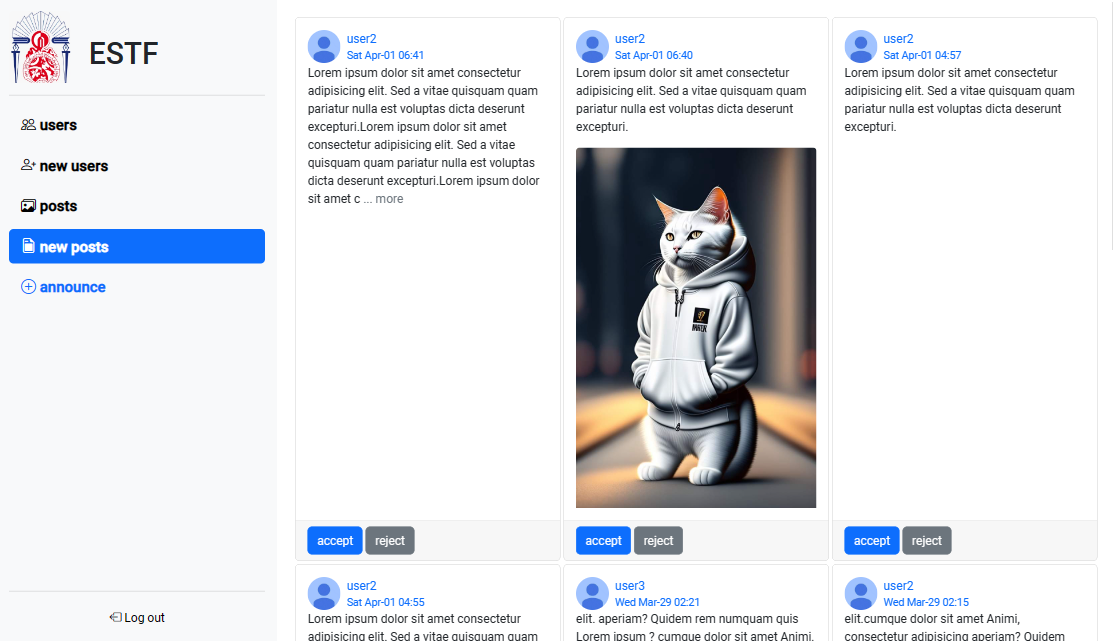


Figure 18 Page nouveaux Publications

Conclusion

En conclusion, le développement d'une application de médias sociaux pour les étudiants a été un voyage passionnant et stimulant. En tant qu'étudiants nous-mêmes, nous reconnaissons l'importance d'une plate-forme qui favorise la communication et la collaboration entre pairs. Tout au long du processus de développement, nous avons concentré nos efforts sur la création d'une application intuitive et conviviale qui permet aux étudiants de se connecter les uns avec les autres, de partager des idées et de rester informés des événements et des actualités.

Nous croyons que notre application a le potentiel de faire un impact positif sur la communauté étudiante et au-delà. En fournissant un espace sûr et inclusif pour que les étudiants puissent interagir, nous espérons promouvoir l'apprentissage, la créativité et la socialisation. De plus, les fonctionnalités de l'application, telles que la messagerie en temps réel, peuvent aider les étudiants à améliorer leurs compétences en communication et en travail d'équipe, qui sont essentielles pour réussir dans le monde du travail moderne.

En regardant vers l'avenir, nous sommes enthousiastes à l'idée de continuer à améliorer notre application et d'explorer de nouvelles façons d'améliorer l'expérience utilisateur. Nous sommes confiants que notre projet peut servir de catalyseur pour un changement positif et l'innovation dans le domaine des médias sociaux. Nous sommes reconnaissants pour l'opportunité de travailler sur ce projet et nous sommes fiers des résultats que nous avons obtenus.

Bibliographie

[Tutoriel officiel de](http://java.sun.com/j2ee/1.4/docs/tutorial/doc/) Angular (en anglais), [en ligne]. Adresse URL: [Angular - Getting started with Angular](https://angular.io/start)

[Tutoriel officiel de](http://java.sun.com/j2ee/1.4/docs/tutorial/doc/) Mongoose (en anglais), [en ligne]. Adresse URL: [Mongoose v7.0.3: Schemas (mongoosejs.com)](https://mongoosejs.com/docs/guide.html)

[Tutoriel officiel de](http://java.sun.com/j2ee/1.4/docs/tutorial/doc/) mongoDB (en anglais), [en ligne]. Adresse URL: [MongoDB CRUD Operations — MongoDB Manual](https://www.mongodb.com/docs/manual/crud/)

Table de figures

[Figure 1Arborescence de l'application 13](#_Toc132141150)

[Figure 2 Diagramme de GANTT 14](https://d.docs.live.net/0882d3a55d8b5090/folders/Documents/rapport-pfe.docx#_Toc132141151)

[Figure 3 : Architecture 3-tiers 22](#_Toc132141152)

[Figure 4 : Schéma détaillé du model MVC 2 23](#_Toc132141153)

[Figure 5Diagramme de cas d'utilisation de la gestion globale de l'application 26](#_Toc132141154)

[Figure 6 Diagramme des cas d'utilisation de la gestion des publications 27](#_Toc132141155)

[*Figure 7 Diagramme de cas d'utilisation de la gestion des notifications* 27](#_Toc132141156)

[Figure 8Diagramme de cas d'utilisation de la gestion des messages 28](#_Toc132141157)

[Figure 9 Diagramme de classe 29](#_Toc132141158)

[Figure 10 Diagramme d'**entité association** 31](https://d.docs.live.net/0882d3a55d8b5090/folders/Documents/rapport-pfe.docx#_Toc132141159)

[Figure 11 Page d'acceuil de l'application 35](#_Toc132141160)

[Figure 12Page d'inscription des étudiants 36](#_Toc132141161)

[Figure 13Page d'authentification des étudiants 37](#_Toc132141162)

[Figure 14Page "Messages" 40](#_Toc132141163)

[Figure 15Page Profile 41](#_Toc132141164)

[Figure 16Page nouveaux utilisateurs 43](#_Toc132141165)

[Figure 17page Publications 44](#_Toc132141166)

[Figure 18 Page nouveaux Publications 45](#_Toc132141167)

Table de matières

Table of Contents

[Sommaire 2](#_Toc132142033)

[Remerciement 3](#_Toc132142034)

[Table d’abréviation 4](#_Toc132142035)

[Introduction 5](#_Toc132142036)

[Chapitre 1 : Cadre et contexte du projet 7](#_Toc132142037)

[1.1. Introduction 7](#_Toc132142038)

[1.2. Contexte du projet 7](#_Toc132142039)

[1.3. Présentation de l’organisme d’accueil : 7](#_Toc132142040)

[Chapitre 2 : Présentation et cahier des charges 9](#_Toc132142041)

[2.1. Introduction 9](#_Toc132142042)

[2.2. Présentation du projet 9](#_Toc132142043)

[2.3. Objectifs et périmètre de projet 10](#_Toc132142044)

[2.4. Public ciblé : 10](#_Toc132142045)

[2.5. Contenu de l’application 11](#_Toc132142046)

[2.5.1 Etape d’inscription 11](#_Toc132142047)

[2.5.2 Étape d’authentification : 11](#_Toc132142048)

[2.5.3 Espace Publications : 11](#_Toc132142049)

[2.5.4 Espace Notifications 11](#_Toc132142050)

[2.5.5 Espace Messagerie 11](#_Toc132142051)

[2.5.6 Espace Compte 11](#_Toc132142052)

[2.5.7 Espace étudiant 12](#_Toc132142053)

[2.6. Charte graphique : exigences et limites 12](#_Toc132142054)

[2.7. Arborescence de l’application 12](#_Toc132142055)

[2.8. Planification du projet 14](#_Toc132142056)

[2.8.1 Diagramme de GANTT 14](#_Toc132142057)

[Chapitre 3 : Technologies et outils utilisés 15](#_Toc132142058)

[3.1. Introduction 15](#_Toc132142059)

[3.2. Technologies implémentées 15](#_Toc132142060)

[3.3.1. Node js 15](#_Toc132142061)

[3.3.2. Angular 16](#_Toc132142062)

[3.3.3. MongoDB 16](#_Toc132142063)

[3.3.4. Express 16](#_Toc132142064)

[3.3.5. Socket.IO 17](#_Toc132142065)

[3.3.6. TypeScript 17](#_Toc132142066)

[3.3.7. CSS 17](#_Toc132142067)

[3.3.8. HTML 18](#_Toc132142068)

[3.3.9. Bootstrap 18](#_Toc132142069)

[3.1. Outils d’implémentation 18](#_Toc132142070)

[3.3.1. Visual Studio Code 18](#_Toc132142071)

[3.3.2. Postman 19](#_Toc132142072)

[3.3.3. MongoDB 19](#_Toc132142073)

[3.3.4. Angular DevTools 20](#_Toc132142074)

[4.1. Chapitre 4 : Architecture et conception 21](#_Toc132142075)

[4.1. Introduction 21](#_Toc132142076)

[4.2. Architecture de l’application 21](#_Toc132142077)

[4.2.1. Architecture 3-tiers (architecture applicative de l’application) 21](#_Toc132142078)

[Couche présentation (premier niveau) : 21](#_Toc132142079)

[Couche métier / business (deuxième niveau) : 22](#_Toc132142080)

[Couche accès aux données (troisième niveau) : 22](#_Toc132142081)

[4.2.2. MVC 2 (Modèle-Vue-Contrôleur 2) 22](#_Toc132142082)

[4.3. Conception de l’application 24](#_Toc132142083)

[4.3.1 Pourquoi UML ? 24](#_Toc132142084)

[4.3.2 Règles de gestion 25](#_Toc132142085)

[4.3.3 Diagrammes des cas d’utilisation 26](#_Toc132142086)

[Gestion globale de l’application 26](#_Toc132142087)

[4.3.4 Diagramme de classes 28](#_Toc132142088)

[4.3.5 Le Diagramme entité association : 29](#_Toc132142089)

[5.1. Chapitre 5 : Présentation de l’application 34](#_Toc132142090)

[5.1. Introduction 34](#_Toc132142091)

[5.2. L’accueil de l’application 34](#_Toc132142092)

[5.3. Etape d’inscription 35](#_Toc132142093)

[5.4. Etape d’authentification 36](#_Toc132142094)

[5.5. Espace utilisateur 37](#_Toc132142095)

[5.5.1 Page d'actualité 37](#_Toc132142096)

[5.5.2 Nouvelle publication 38](#_Toc132142097)

[5.5.3 Mes messages 39](#_Toc132142098)

[5.5.4 Mon Profile 40](#_Toc132142099)

[5.6. Espace administrateur 41](#_Toc132142100)

[5.6.1 Page utilisateurs 41](#_Toc132142101)

[5.6.2 Page nouveaux utilisateurs 42](#_Toc132142102)

[5.6.3 Page Publications 43](#_Toc132142103)

[5.6.4 Page nouveaux publications 44](#_Toc132142104)

[Conclusion 46](#_Toc132142105)

[Bibliographie 47](#_Toc132142106)

[Table de figures 48](#_Toc132142107)

[Table de matières 49](#_Toc132142108)