École Nationale Supérieure d'Informatique et d'Analyse des Systèmes

Rapport du projet Java EE

Filière Génie Logiciel

Sujet

Réalisation d'un réseau social orienté sciences

Réalisé par :
Ajabri Hiba
Douslimi Yassir
Eddaghal Mohammed
EL BIACHE Houda

Encadré par :
Mr. HAMLAOUI Mahmoud

Année universitaire 2020 - 2021

Remerciements

Ce projet est le fruit des conseils et critiques bienveillantes d'un grand nombre de personnes. Il nous est agréable de nous acquitter d'une dette de reconnaissance auprès de toutes ces personnes, dont l'intervention a favorisé son aboutissement.

Tout d'abord, nos sincères remerciements s'adressent à notre cher professeur, Monsieur HAMLAOUI Mahmoud. Ses compétences indéfectibles, ses directives clairvoyantes, ses conseils instructifs, de même que ses encouragements réconfortants nous ont toujours été singulièrement précieux, au cours de ce projet.

Toute notre considération et tout notre respect vont également à toute personne ayant contribué à l'élaboration de ce projet et à sa réussite. Il nous est aussi agréable de remercier notre chef de filière, Madame EL ASRI Bouchra, ainsi que les enseignants de notre filière qui ont veillé à notre formation en mettant à notre service leur savoir et leur érudition.

Résumé

Le présent document est l'aboutissement de notre travail dans le cadre du projet Java EE, dont l'enjeu principal est de mettre en pratique nos connaissances et compétences acquises en technologies web au cours notre formation en 2ème année Génie Logiciel au sein de l'école nationale supérieure d'informatique et d'analyse des systèmes.

Notre choix de sujet s'est porté vers la réalisation d'un **réseau social orienté sciences**.

Nous avons commencé par établir notre cahier de charge suite à une étude qui nous a permis de relever les besoins fonctionnels. Nous avons ensuite entamé la phase de la conception à travers des diagrammes UML qui nous ont permis de structurer la couche métier de l'application. Puis nous avons abordé la phase de la réalisation en développant de manière efficace les fonctionnalités en question et en effectuant les tests nécessaires, pour arriver finalement à faire le bilan du projet.

Ce rapport qui a pour but de décrire le déroulement de notre projet contient l'ensemble des éléments mis en jeu et permet de synthétiser le résultat du travail accompli.

Abstract

The current document is a report which summarizes the work done as part of our Java EE project which main aim was to put into practice the skills and the knowledge we acquired in terms of web development and software engineering during our 2nd year at ENSIAS.

Concerning the subject, we chose to make an attempt to create a **science-oriented social network** geared toward sharing knowledge and research.

It was first necessary to apply the fundamental phases in most, if not all, web and mobile engineering methodologies. Starting by the analysis, we established our specifications. Then came conception phase through UML diagrams which allowed us to structure the business layer of the application. Based on the previous steps, we developped a prototype according to the needs. In the following implementation phase, system unit components were built effectively with respect to quality, performance, baselines, libraries, and debugging issues, to finally be able to carry out the necessary tests.

This report, which purpose is to describe the progress of our project, contains all the elements involved and makes it possible to summarize the work accomplished, as well as its results.

Table des figures

Table des matières

Chapitre 1

Présentation du projet

1.1 Introduction

Nous avons opté pour "Ensialligence", comme nom de projet puisque cela nous a semblé adapté aux fonctionnalités proposées et étant donné qu'il illustre parfaitement le contexte général du projet.

1.2 Cahier de charges

1.2.1 Contexte et périmètre

Le projet dont il est question consiste à implémenter un site web sous forme de réseau social qui offre principalement la possibilité d'interagir avec une communauté regroupée par ses points d'intérêt communs, et ce notamment par la création de publications qui peuvent être partagées avec d'autres utilisateurs.

Les premières versions devraient se concentrer principalement sur les articles textuels avec des articles multimédias à venir plus tard.

Le projet devrait également permettre aux utilisateurs d'interagir les uns avec les autres en réagissant à leurs posts ou articles à travers des likes ou mentions "j'aime", en les commentant ou en les partageant, ainsi que de pouvoir envoyer des demandes d'amitié dans le but de discuter dans la section du chat avec leurs amis.

Il y aura également de nombreux éléments d'interface utilisateur qui amélioreront l'expérience d'apprentissage et de recherche de l'utilisateur.

Notre application vise en premier lieu les utilisateurs passionnés, qui ont soif de connaissances, de découverte et de partage.

1.2.2 Besoins

1.2.2.1 Exigences fonctionnelles

La distinction des besoins fonctionnels fait l'office d'une étape primordiale au début de chaque démarche de développement.

Son but est de veiller à développer un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : Quelles en sont les fonctions ?

Notre site web doit donc répondre aux exigences suivantes :

— Authentification ;

— Création de posts, publications, articles ;

— Réaction à ces derniers à travers des mentions "j'aime" ;
— Ajout de commentaires sur les publications ;

— Recherche d'utilisateurs ;

— Envoi d'invitations ou de demandes d'amitié afin d'ajouter d'autres utilisateurs en tant qu'amis ;

— Visite de profils ;

— Messagerie (chat) ;

1.2.2.2 Exigences non fonctionnelles

— Performance :

Le système doit répondre à toutes les exigences des utilisateurs de manière optimale

— Convivialité :

Les interfaces utilisateurs doivent être conviviales, simples, ergonomiques et adaptées à l'utilisateur.

— Sécurité :

L'application doit respecter surtout la confidentialité des données personnelles qui reste l'une des contraintes les plus importantes.

— Service de support :

L'application doit être capable de fournir un service de support après le déploiement chez les utilisateurs dans le cas de bugs ou d'éventuels problèmes rencontrés afin de faciliter l'utilisation de la solution.

1.3 Conclusion

Ce chapitre a été le point de départ pour l'élaboration du projet, dans la mesure où il décrivait son contexte général, en présentant les objectifs généraux à atteindre ainsi que la démarche et les étapes de sa mise en œuvre.
Le chapitre suivant sera réservé à l'analyse et la conception détaillées en expliquant les démarches du projet.

Chapitre 2

Analyse et conception

2.1 Introduction

Afin de bien maitriser le système, cette phase s'impose comme une étape indispensable. De même pour l'élaboration des spécifications techniques qui a pour objectif de proposer de meilleures solutions qui satisfont complétement nos besoins. Et ce à travers une étude UML en faisant appel à des diagramme élaborés par le biais de PlantText UML Editor. Ces diagrammes UML servent à bien structurer la couche métier de l'application, en rassemblant les classes d'analyse par fonctionnalité.

2.2 Méthodologie projet de cycle en V

Le cycle en V est un modèle d'organisation des activités d'un projet qui se caractérise par un flux d'activité descendant qui détaille le produit jusqu'à sa réalisation, et un flux ascendant, qui assemble le produit en vérifiant sa qualité.

Ce modèle est issu du modèle en cascade dont il reprend l'approche séquentielle et linéaire de phases.

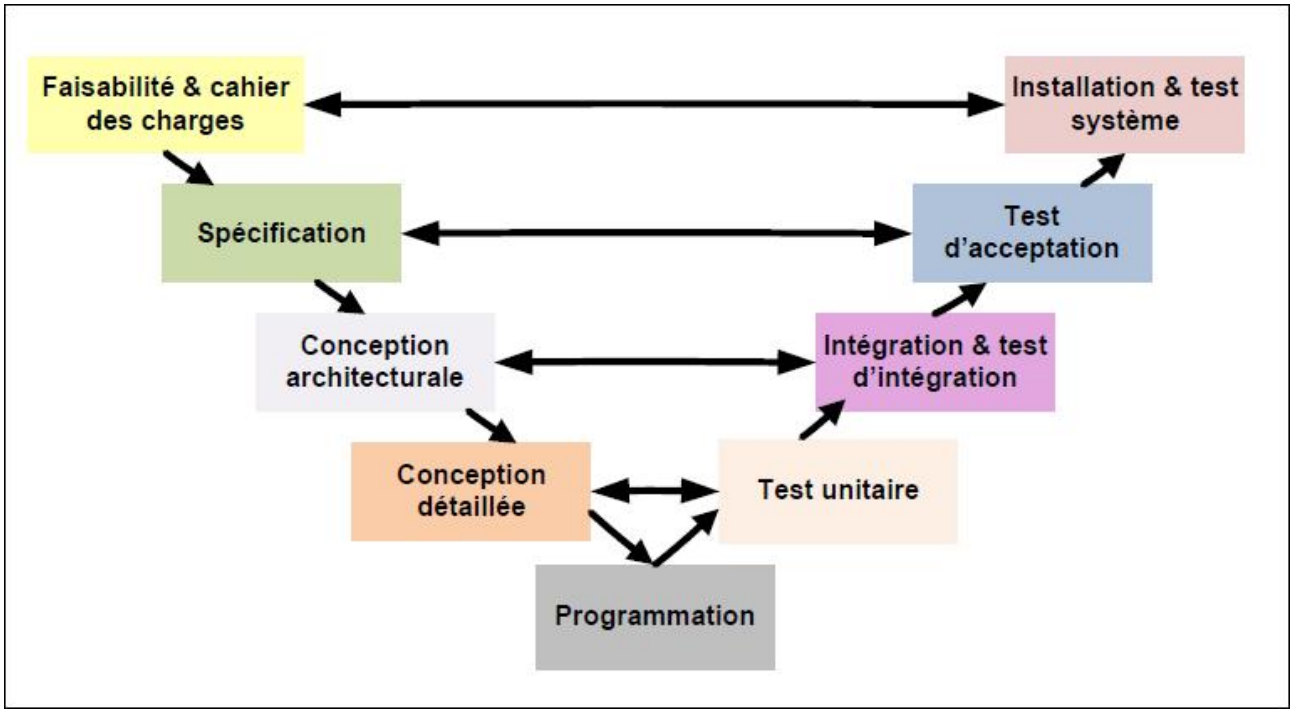

FIGURE 2.1 – Modèle de cycle en V

2.3 Spécification des exigences : cas d'utilisation

2.3.1 Identification des acteurs

Les acteurs qui interviennent dans notre application représentent toute personne y ayant accès. Le cœur du réseau social sera donc accessible de la part de tous les **utilisateurs** suite à leur authentification.

2.3.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation ou use case permet de structurer les besoins de l'utilisateur et les objectifs correspondants d'un système, ainsi qu'à identifier les fonctionnalités principales du système. Les use cases ne doivent en aucun cas décrire des solutions d'implémentation. Leur but est justement d'éviter de tomber dans la dérive d'une approche fonctionnelle.

Voici les cas d'utilisation de notre réseau social :

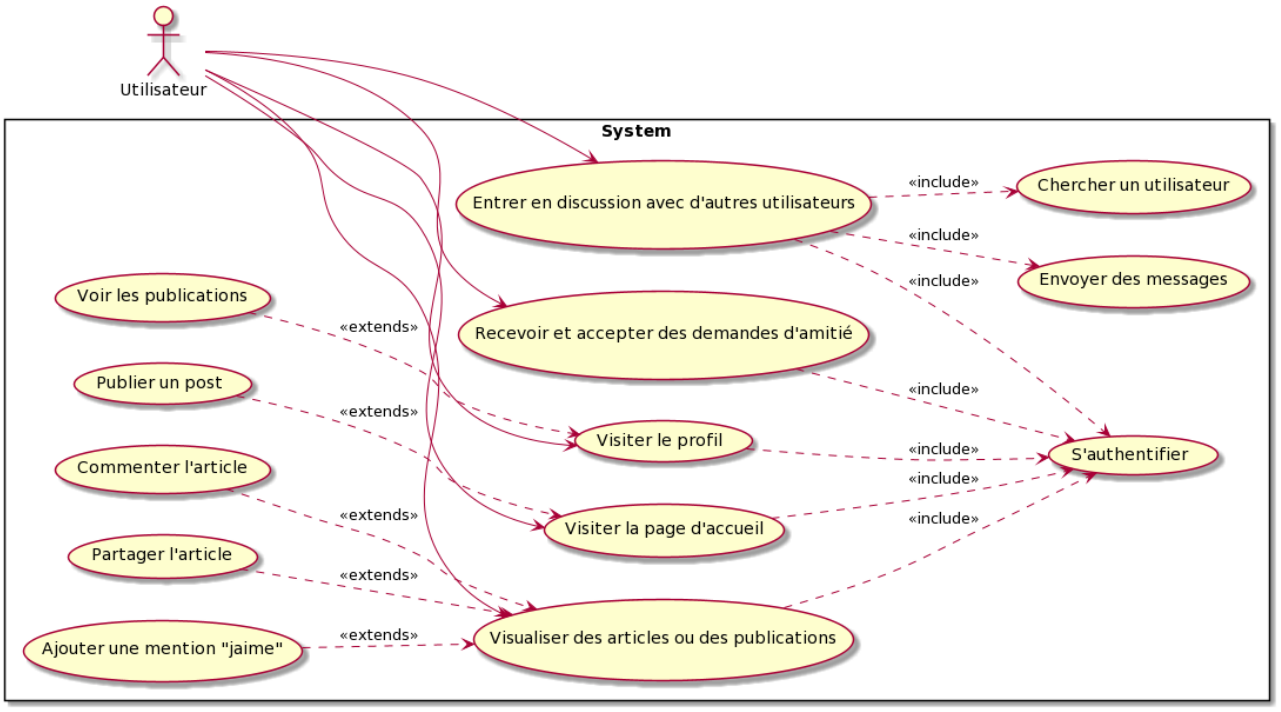

FIGURE 2.2 – Diagrammes de cas d'utilisation

2.4 Analyse du domaine : diagramme de classes

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes. Son intérêt est de modéliser les entités du système d'information, il permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont regroupées dans des classes.

Le diagramme met en évidence d'éventuelles relations entre ces classes, comme suit :

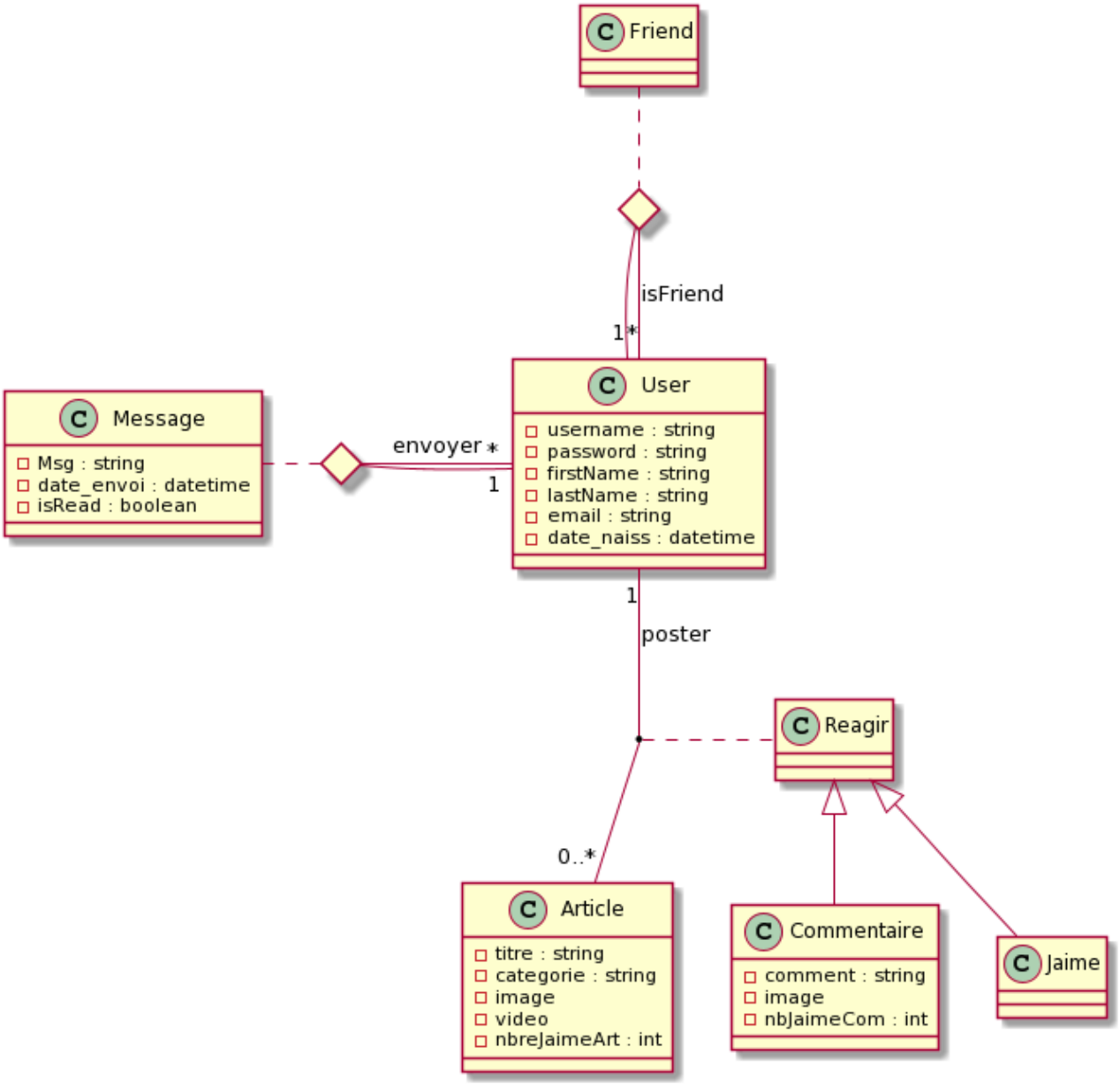

FIGURE 2.3 – Diagramme de classes

2.5 Conclusion

Dans ce chapitre, nous avons délimité le projet afin d'avoir une idée globale sur sa structure en abordant, notamment l'aspect fonctionnel de l'application, ainsi que les différents acteurs du système et leur cas d'utilisation, et enfin le diagramme de classes établies suite à l'étude des spécifications. Tous les ingrédients sont ainsi disponibles pour entamer efficacement la mise en œuvre du système.

Chapitre 3

Maquettes

3.1 Logo

FIGURE 3.1 – Logo de Ensialligence

3.2 Login

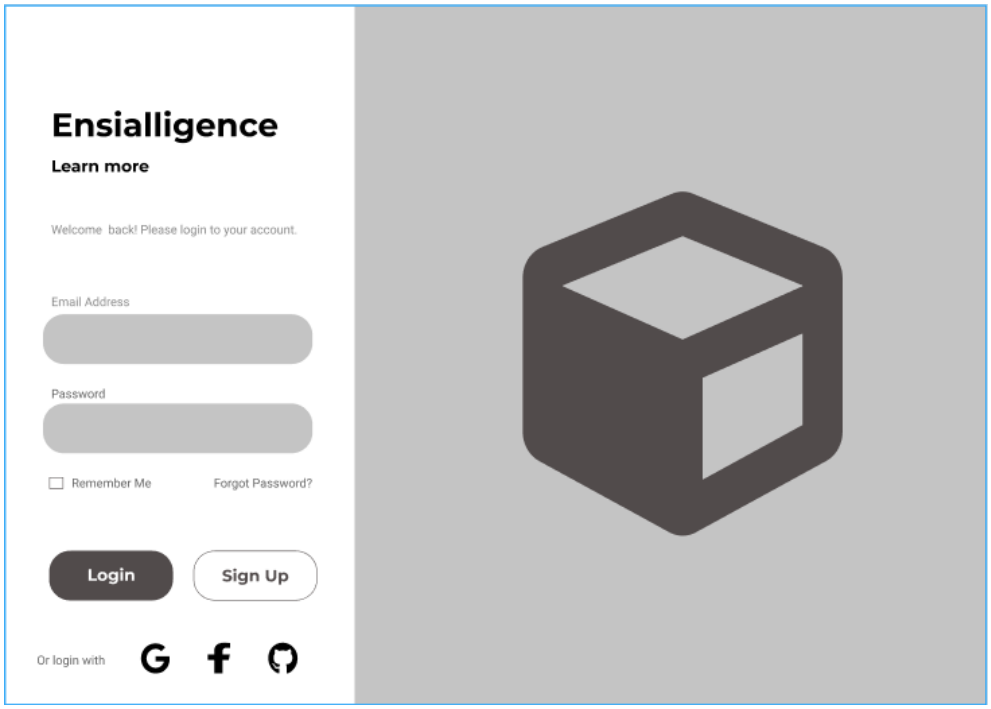

FIGURE 3.2 – Maquette Login

3.3 Home

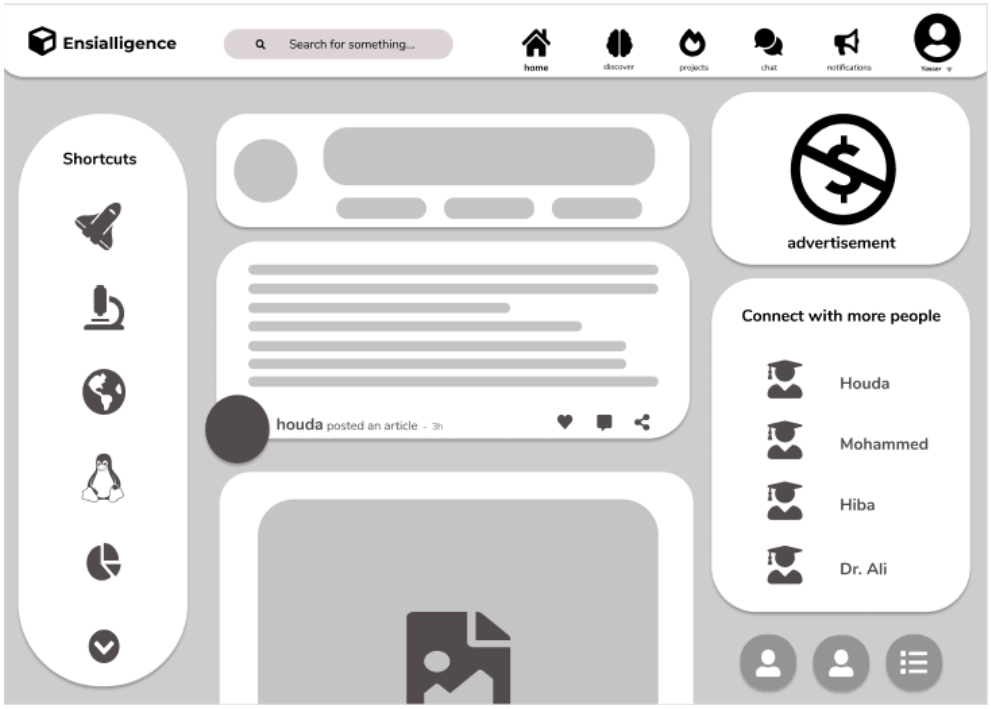

FIGURE 3.3 – Maquette Home

3.4 Chat

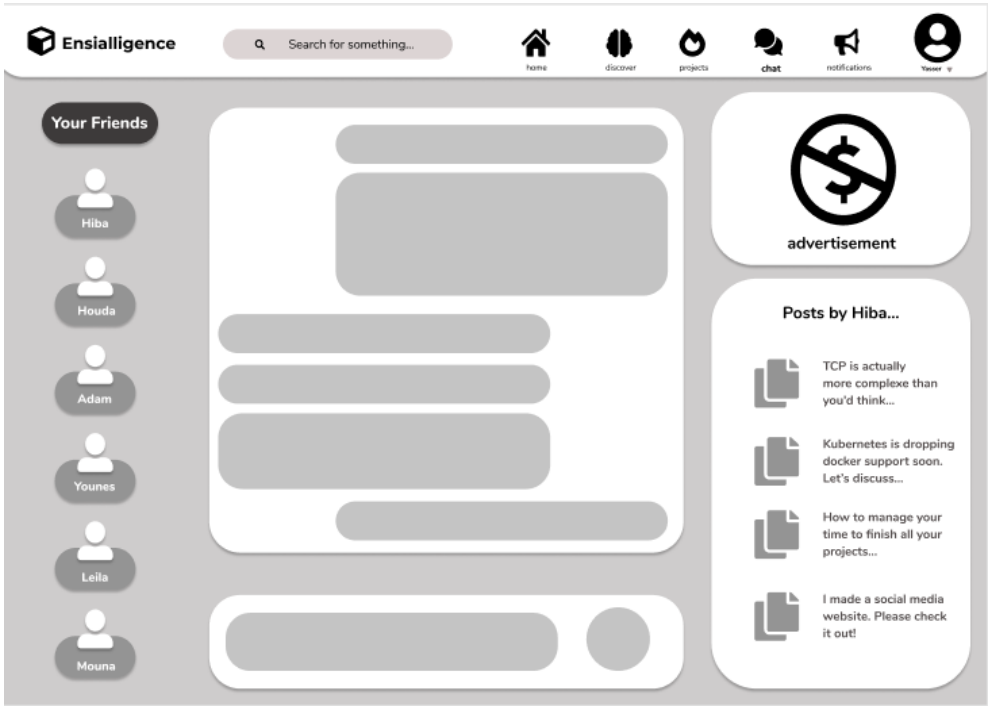

FIGURE 3.4 – Maquette Chat

Chapitre 4

Réalisation

4.1 Introduction

Après avoir mené les phases précédentes, passant par la phase de la spécification et d'analyse, suivies par les phases de la conception détaillée et de l'étude technique, l'étape suivante sera consacrée à la réalisation du projet.

4.2 Architecture du projet

4.2.1 Pattern MVC

Pour la réalisation de notre projet nous avons opté pour une architecture en modèle MVC (Modèle - vue - contrôleur). Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles de données, les vues (interface utilisateur) et les contrôleurs (logique de contrôle).

Un modèle (Model) : Élément qui contient les données ainsi que de la logique en rapport avec ces dernières : validation, lecture et enregistrement.

Une vue (View) : Partie visible d'une interface graphique. Elle contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle.

Un contrôleur (Controller) : Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue. Ce motif est utilisé par de nombreux Framework pour applications web tels que Ruby on Rails, Django, ASP.NET MVC, Spring, Struts, Symfony, Apache Tapestry ou Angular Js.

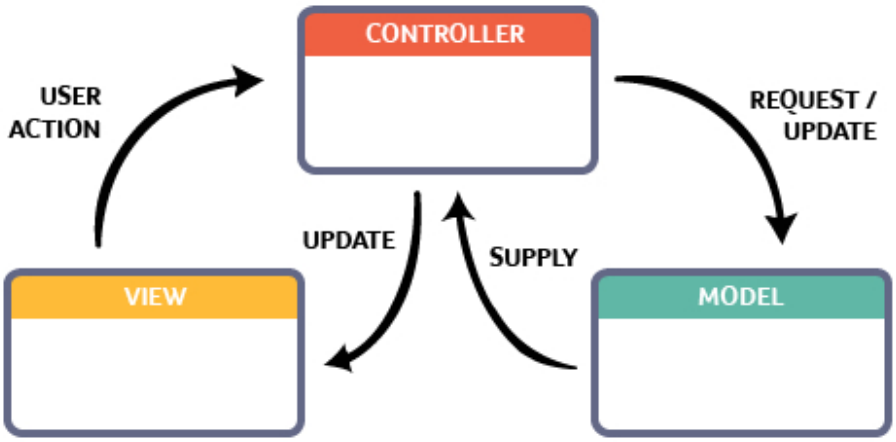

FIGURE 4.1 – Architecture MVC

Afin d'appliquer le paradigme cité ci-dessus et de regrouper les fonctions néccessaires, nous avons procédé au découpage de l'application en packages en séparant les différents composants de cette architecture, comme suit :

— **Model** : données (accès et mise à jour)

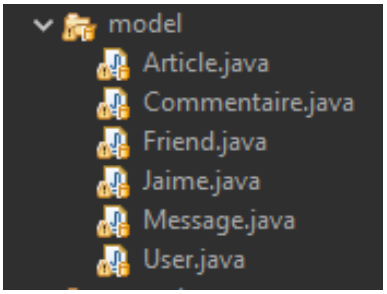

FIGURE 4.2 – Modèle

— **View** : interface utilisateur (entrées et sorties)

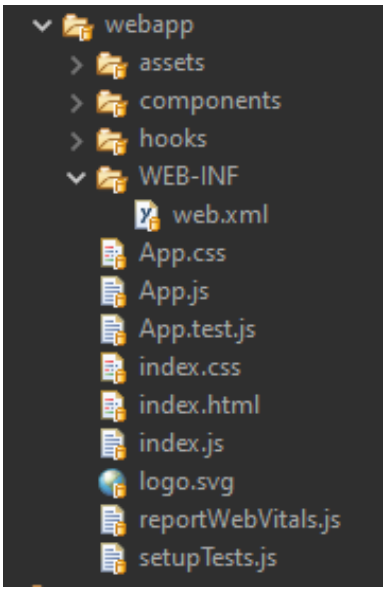

FIGURE 4.3 – Vue

— **Controller** : gestion des événements et synchronisation

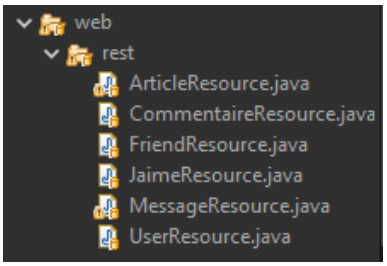

FIGURE 4.4 – Contrôleur

4.2.2 Partie métier

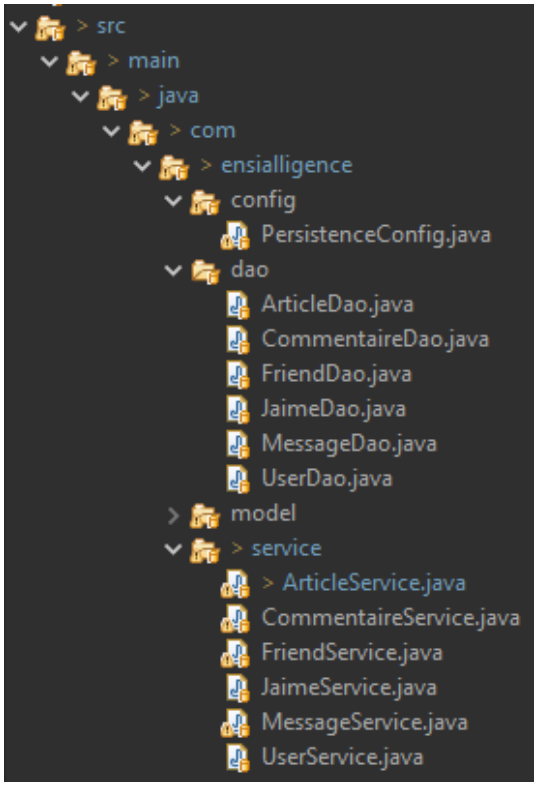

FIGURE 4.5 – Partie métier

4.3 Outils et technologies

Pour l'implémentation de l'architecture proposée dans les chapitres précédents, plusieurs outils ont étés utilisés, dont notamment :

4.3.1 Java EE

FIGURE 4.6 – Logo de Java EE

Java EE (Java Enterprise Edition) est un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

La plate-forme JEE présente une solution optimale pour développer des applications robustes, sécurisées et évolutives.

On en parle généralement pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution.

4.3.2 Maven

FIGURE 4.7 – Logo de Maven

Apache Maven est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier.

Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel.

4.3.3 MySQL

FIGURE 4.8 – Logo de MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Miscrosoft SQL server.

4.3.4 JDBC

JDBC (Java Database Connectivity) est une interface de programmation pour les programmes utilisant la plateforme Java. Elle permet d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC. Normalement, il s'agit d'une base de données relationnelle, et des pilotes JDBC sont disponibles pour tous les systèmes connus de bases de données relationnelles.

4.3.5 JAX-RS

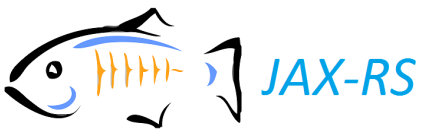

FIGURE 4.9 – Logo de JAX-RS

JAX-RS : Java API for RESTful Web Services est une interface de programmation Java permettant de créer des services Web avec une architecture REST.

4.3.6 Jersey

FIGURE 4.10 – Logo de Jersey

Jersey est un framework open-source écrit en langage Java permettant de développer des services web selon l'architecture REST suivant les spécifications de JAX-RS

4.3.7 ReactJS

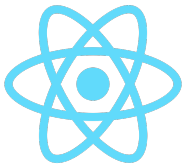

FIGURE 4.11 – Logo de ReactJS

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre dont le but principal est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page ou portion HTML à chaque changement d'état.

C'est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC.

Elle se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité

4.3.8 Eclipse

FIGURE 4.12 – Logo de Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting...

4.3.9 Apache Tomcat

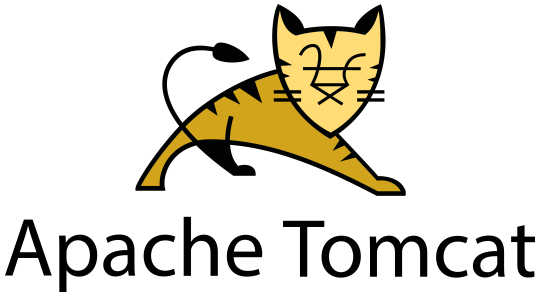

FIGURE 4.13 – Logo de Apache Tomcat

Apache Tomcat est un conteneur web libre de servlets et JSP. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

4.3.10 JUnit

FIGURE 4.14 – Logo de JUnit

JUnit est un framework de test unitaire pour le langage de programmation Java.

4.3.11 Docker

FIGURE 4.15 – Logo de Docker

Docker est un logiciel libre permettant de lancer des applications dans des conteneurs logiciels.

Selon la firme de recherche sur l'industrie 451 Research, « Docker est un outil qui peut empaqueter une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur ».

Il ne s'agit pas de virtualisation, mais de conteneurisation, une forme plus légère qui s'appuie sur certaines parties de la machine hôte pour son fonctionnement.

4.3.12 Git et Github

FIGURE 4.16 – Logos Git et Github

Git est un logiciel de gestion de versions (Version Control System) qui suit l'évolution des fichiers sources et garde les anciennes versions de chacun d'eux sans rien écraser. Cela permet de retrouver les différentes versions d'un fichier ou d'un lot de fichiers connexes et ainsi éviter des problèmes tel que "Qui a modifié le fichier ZaZa, tout fonctionnait hier et aujourd'hui, il y a des bugs !" Avec Git, vous retrouverez sans problème la version qui fonctionnait la veille.

GitHub est une plateforme de "codes" open-source. Pour faire court, c'est une sorte de réseaux social de développeurs. Vous pouvez donc utiliser les projets (à condition de citer vos sources !) et même participer à un projet qui vous intéresse. Sur cette plateforme, vous trouvez de nombreux projets connus comme JQuery par exemple...

4.4 Conclusion

Ce chapitre a été consacré à la description de la mise en œuvre du projet. Ainsi, nous avons présenté l'architecture du projet, ainsi que les différentes technologies et outils de réalisation mis en jeu.

Chapitre 5

Bilan du projet

5.1 Introduction

Ce chapitre va illustrer l'allure, ainsi que les interfaces représentant les différents modules de notre application à travers des captures d'écran.

5.2 Captures d'écran

5.2.1 Login Page

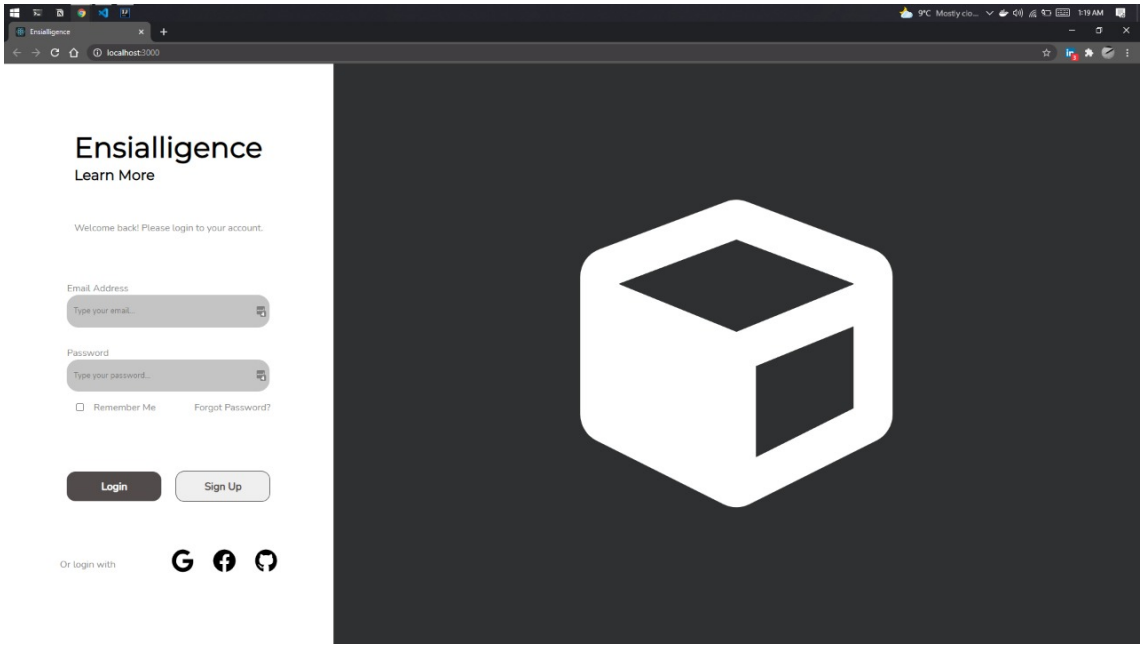

FIGURE 5.1 – Login Page

5.2.2 Home Page

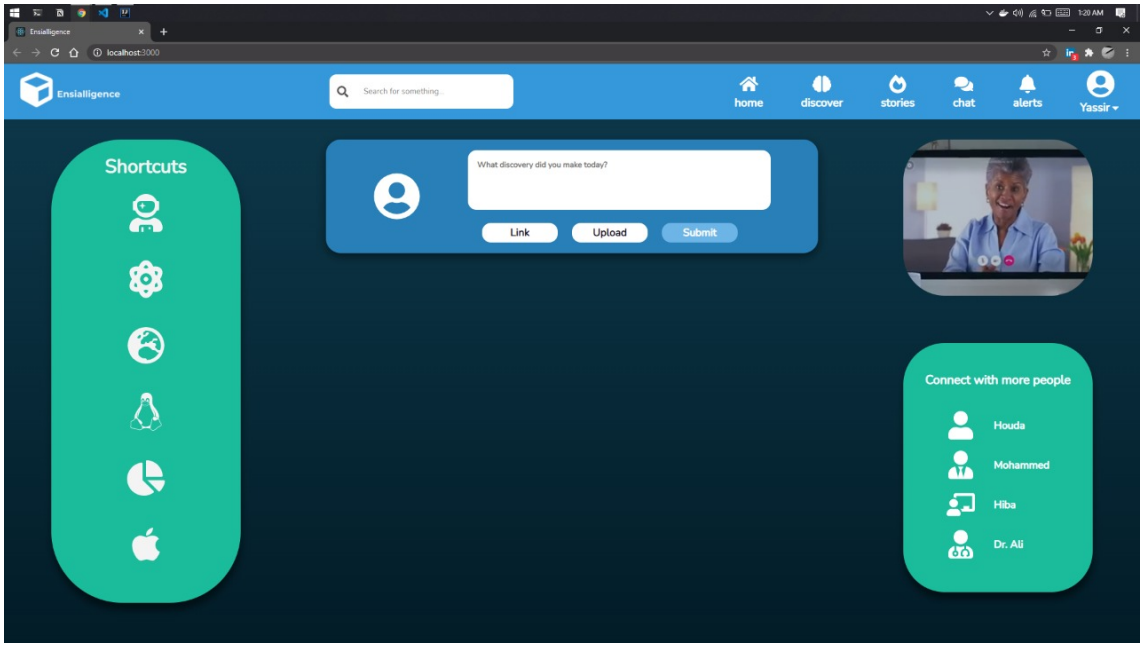

FIGURE 5.2 – Home Page

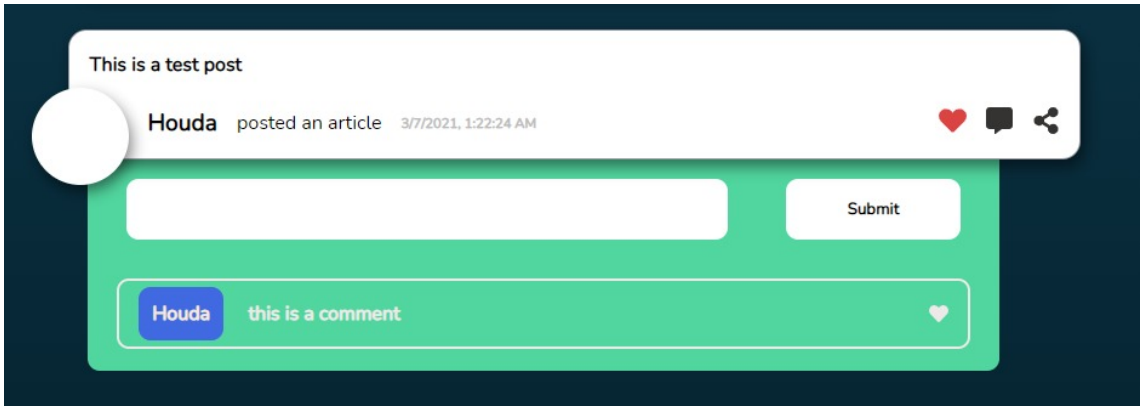

FIGURE 5.3 – Post + Mention "j'aime" + Commentaire

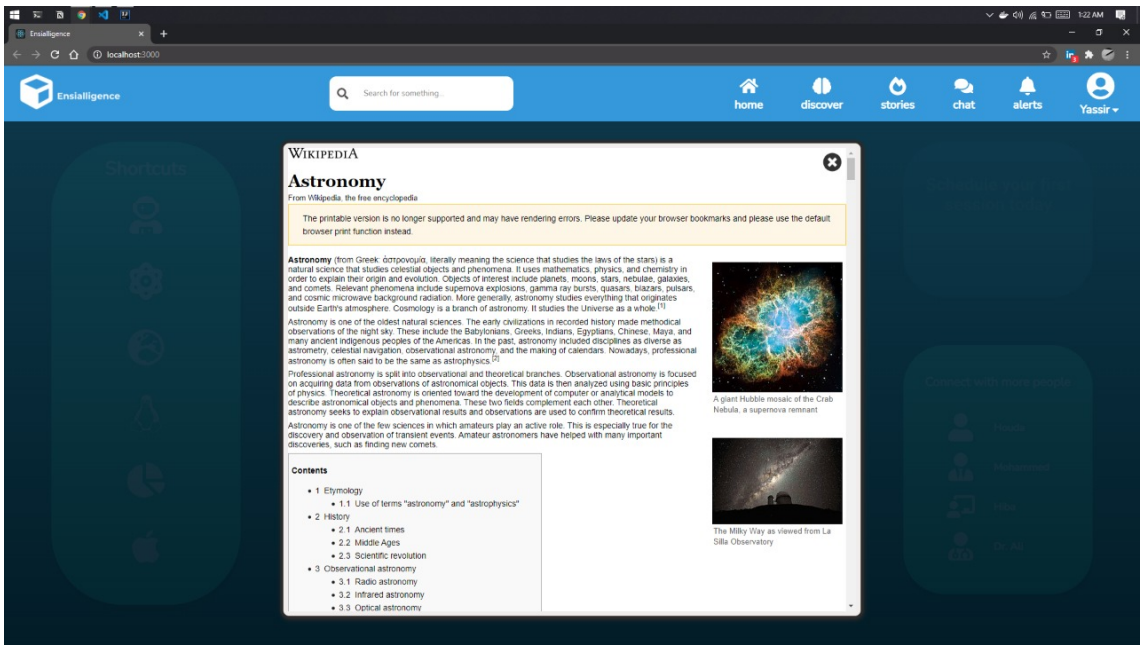

FIGURE 5.4 – Shortcuts

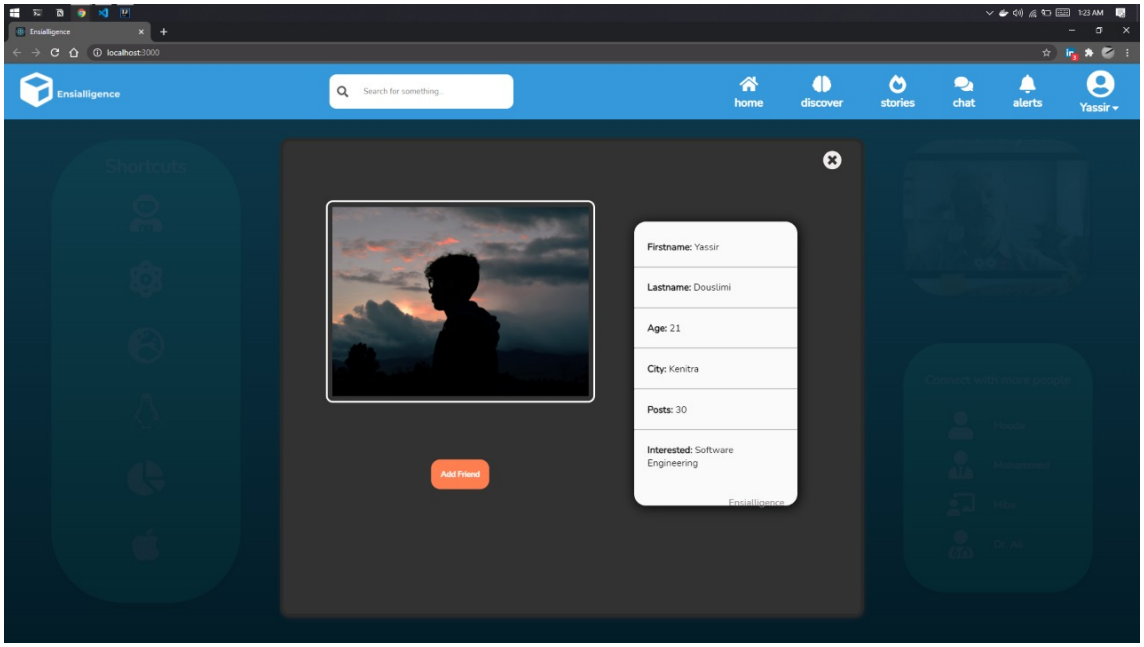

FIGURE 5.5 – Connect with more people

5.2.3 Discover

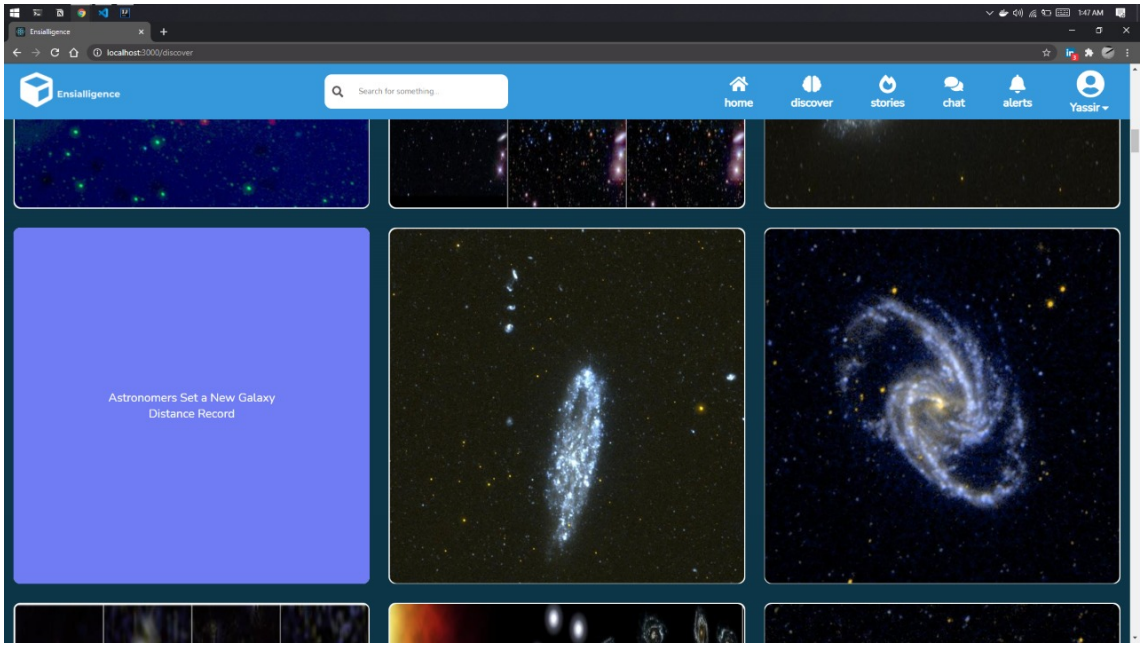

FIGURE 5.6 – Discover Page

5.2.4 Stories

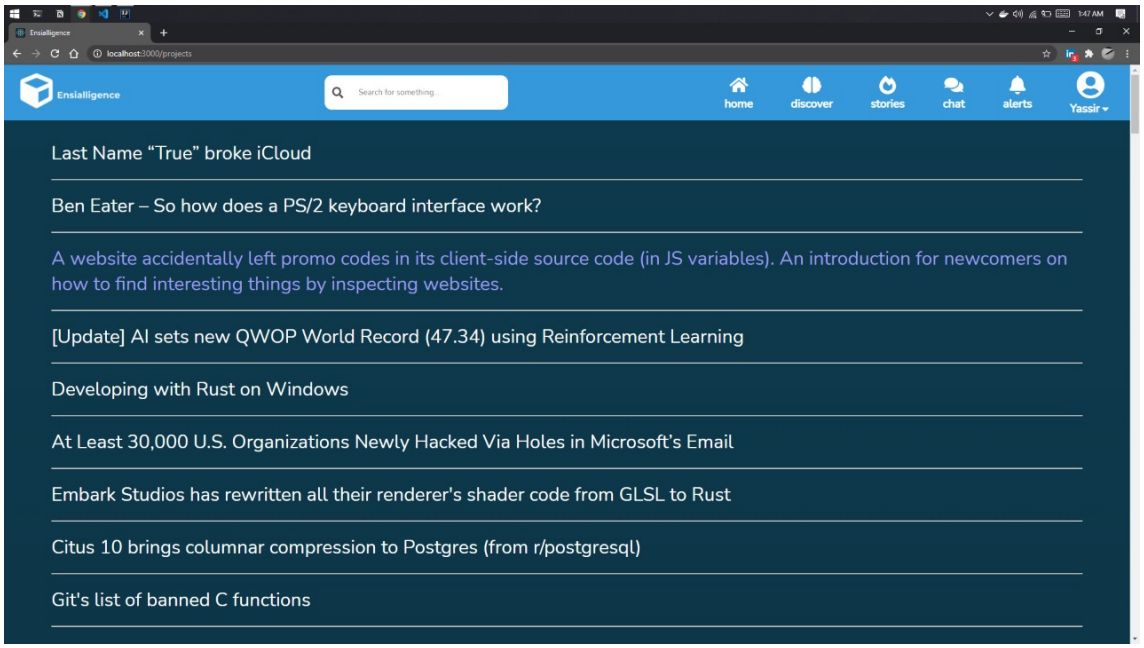

FIGURE 5.7 – Stories Page

5.2.5 Chat

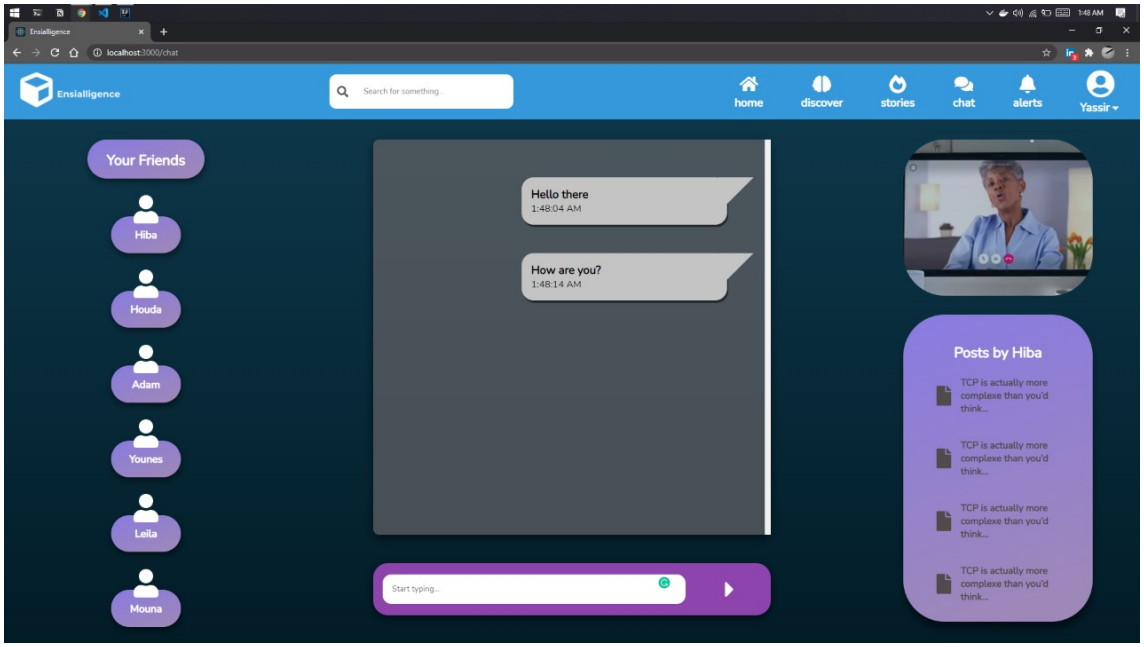

FIGURE 5.8 – Chat Page

5.2.6 Alerts

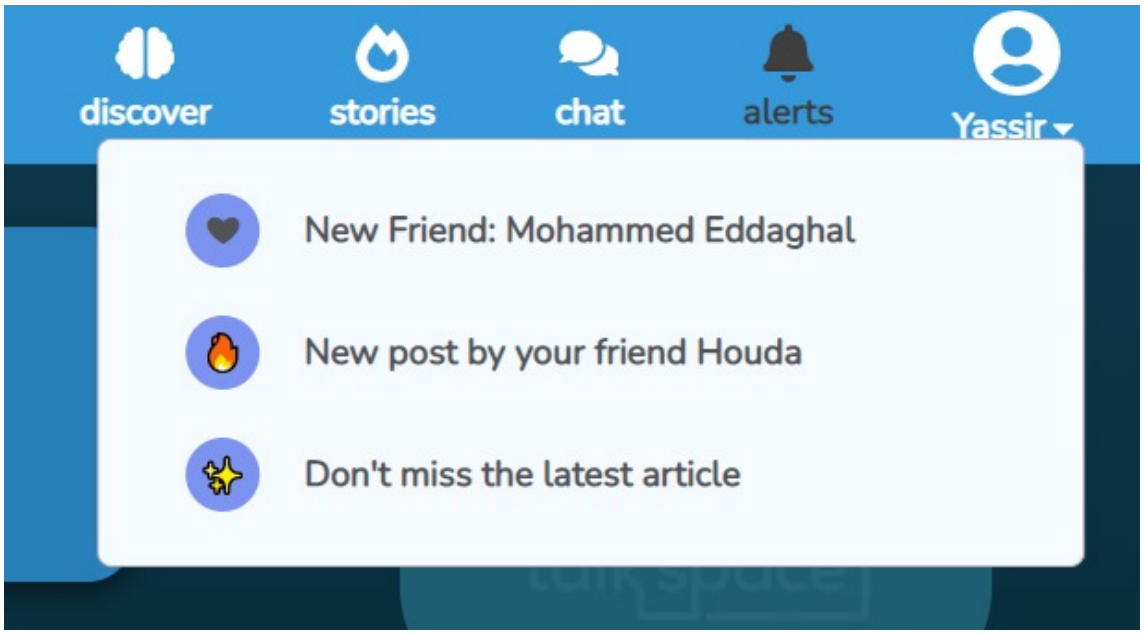

FIGURE 5.9 – Alerts

5.2.7 Profile

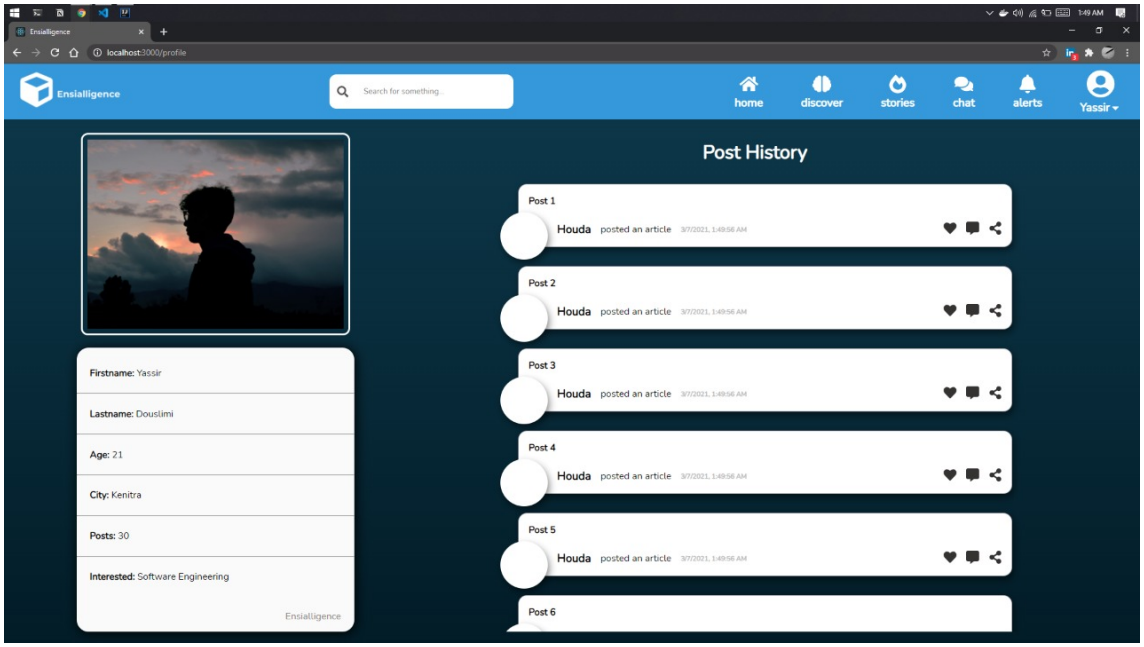

FIGURE 5.10 – Profile Page

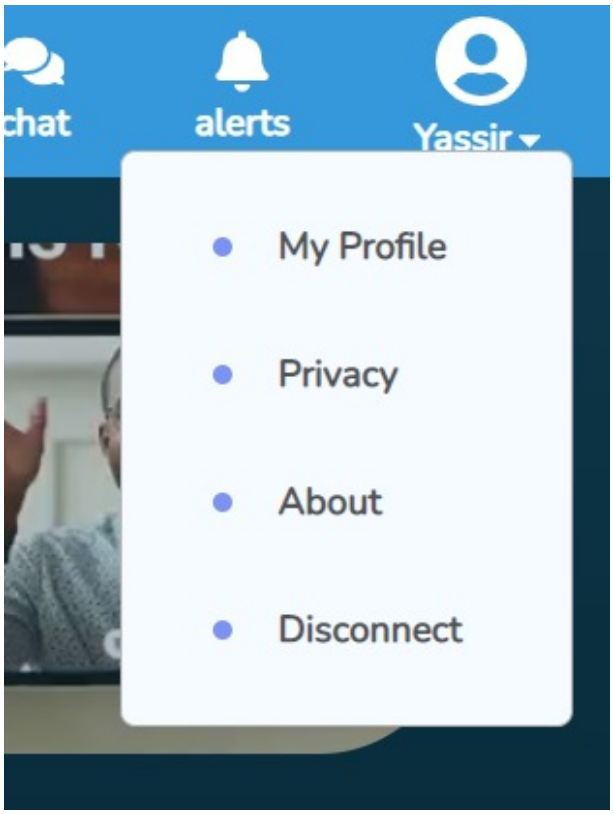

FIGURE 5.11 – Profile

Conclusion

En guise de conclusion, notre projet Java EE consistait à la réalisation d'un site web de réseau social.

Pour réaliser ce projet, nous avons commencé par une étude détaillée du contexte général avant de s'intéresser aux besoins et aux technologies existantes. La phase suivante était celle de la conception , suivi de la phase de réalisation du projet dans laquelle nous avons détaillé la mise en œuvre de la solution élaborée. Cette réalisation nous a permis d'affiner nos capacités d'analyse et de conception et de renforcer nos compétences en matière de technologies employées.

Or, notre réalisation est loin d'être parfaite, voire terminée. Une fois le travail achevé, une phase approfondie de tests pourrait démarrer, en commençant par les tests d'intégration en arrivant aux tests end to end. Des améliorations pourraient éventuellement être envisageables par la suite.

En somme, ce projet nous aura permis de nous familiariser avec les outils de Java EE, comme il nous aura permis d'améliorer notre aptitude à collaborer et à travailler en groupe, malgré les difficultés que présente la situation sanitaire actuelle.