Cahier des Charges : Réseau Social en Architecture Microservices

---

1. Présentation du Projet et Contexte

1.1 Contexte

L'objectif est de créer un réseau social performant en utilisant une architecture microservices, garantissant scalabilité et résilience. Inspiré de plateformes existantes comme Facebook, ce projet vise à offrir une expérience utilisateur fluide et interactive pour un public académique, dans le cadre d'un projet universitaire.

1.2 Objectifs

- Concevoir et développer un réseau social fonctionnel et optimisé.
- Utiliser une architecture microservices avec **Spring Boot, Spring Security, Spring Cloud** pour le backend et **React, TypeScript, HTML, CSS** pour le frontend.
- Implémenter un système de cache performant (**Redis**).
- Assurer une sécurité optimale des utilisateurs.
- Permettre un système d'authentification avancé avec confirmation par email et récupération de mot de passe.

---

2. Analyse et Expression des Besoins

2.1 Fonctionnalités Principales (User Stories)

2.1.1. Service d'Authentification

- En tant qu'utilisateur, je veux pouvoir **m'inscrire** en utilisant mon email et un mot de passe.
- En tant qu'utilisateur, je veux **recevoir un email de confirmation** pour activer mon compte.
- En tant qu'utilisateur, je veux pouvoir **récupérer mon mot de passe oublié**.
- En tant qu'utilisateur, je veux **me connecter** à mon compte via un (email ou username) et un mot de passe.
- En tant qu'utilisateur, je veux **me déconnecter** de mon compte.

2.1.2. Service Utilisateur

- En tant qu'utilisateur, je veux **mettre à jour mes informations personnelles** (nom, photo, bio, etc.).
- En tant qu'utilisateur, je veux **rechercher, suivre et envoyer des demandes d'amitié** à d'autres utilisateurs.

- En tant qu'utilisateur, je veux **recevoir et gérer les demandes d'amitié** (accepter ou refuser).

2.1.3. Service de Publications (Publication Service)

- En tant qu'utilisateur, je veux **créer** une publication contenant texte, images ou vidéos.
- En tant qu'utilisateur, je veux **modifier ou supprimer** mes propres publications.
- En tant qu'utilisateur, je veux **consulter les détails** d'une publication (contenu, auteur, date).

2.1.4. Service de Fil d'Actualité (Feed Service)

- En tant qu'utilisateur, je veux **voir un flux** de publications pertinent, incluant :
    - les publications **publiques** d'autres utilisateurs,
    - les publications de mes **amis** ou des utilisateurs que je suis.
- En tant qu'utilisateur, je veux **pouvoir filtrer** ou **trier** mon fil d'actualité selon certains critères (optionnel).

2.1.5. Service de Commentaires (Comment Service)

- En tant qu'utilisateur, je veux **ajouter un commentaire** à une publication.
- En tant qu'utilisateur, je veux **modifier ou supprimer** mes propres commentaires.
- En tant qu'utilisateur, je veux **voir tous les commentaires** associés à une publication.
- En tant qu'utilisateur, je veux **répondre à un commentaire** et voir les réponses sous forme de threads.

2.1.6. Service de Likes (Like Service)

- En tant qu'utilisateur, je veux **liker ou unliker** une publication.
- En tant qu'utilisateur, je veux **visualiser le nombre** de likes et l'**identité** des utilisateurs qui ont liké une publication.
- En tant qu'utilisateur, je veux **voir une liste détaillée des utilisateurs** ayant liké une publication.

2.1.7. Service de Notifications

- En tant qu'utilisateur, je veux **recevoir des notifications en temps réel** pour les interactions importantes (likes, commentaires, messages, demandes d'amitié).
- En tant qu'utilisateur, je veux **être notifié lorsqu'une demande d'amitié m'est envoyée** et pouvoir y répondre directement.

2.1.8. Service de Messagerie (Message Service)

- En tant qu'utilisateur, je veux **envoyer et recevoir des messages privés** avec d'autres utilisateurs.
- En tant qu'utilisateur, je veux **consulter l'historique** de mes conversations privées.
- En tant qu'utilisateur, je veux **avoir une interface de messagerie intuitive** similaire à WhatsApp, avec des discussions instantanées.

3. Analyse Fonctionnelle

3.1 Technologies Utilisées

Backend :

- **Spring Boot** (microservices)
- **Spring Security** (gestion des authentifications et autorisations)
- **Spring Cloud** (gestion des microservices et de la communication)
- **Base de données PostgreSQL** (par défaut) avec **Redis** pour la gestion du cache.
- **Kafka** pour la communication asynchrone (au besoin).
- **gRPC** (principalement pour les communications directes inter-services, plus orientées hautes performances).

Frontend :

- **React** (TypeScript, HTML, CSS)
- **Bootstrap ou TailwindCSS** pour le design
- **Consommation des API REST**
- **Conception d'interfaces séparées** :
    - Une **page de profil** affichant les informations personnelles et publications de l'utilisateur.
    - Une **page dédiée au fil d'actualité**.
    - Une **page de notifications** pour voir toutes les interactions reçues.
    - Une **page de messagerie** permettant des discussions privées en temps réel.

Infrastructure :

- **Docker et Kubernetes** pour le déploiement et la scalabilité (Kubernetes est considéré comme un élément essentiel).
- **CI/CD avec GitHub Actions ou GitLab CI.**

---

4. MVP (Produit Minimum Viable)

Le MVP inclut les fonctionnalités suivantes pour une version initiale :

- Authentification et confirmation d'inscription via email.
- Récupération du mot de passe oublié.
- Gestion de profil utilisateur (modification d'informations, recherche d'utilisateurs, demandes d'amitié).
- **Service de Publications** pour créer, modifier et supprimer un post.
- **Service de Feed** pour afficher les posts publics et ceux des amis/abonnements.
- **Service de Commentaires** avec possibilité de répondre aux commentaires.

- **Service de Likes** avec affichage détaillé des utilisateurs ayant aimé une publication.
- Messagerie privée avec interface de discussion instantanée et notifications en temps réel.

---

5. Fonctionnalités Optionnelles et évolutions futures

- **Base de données MongoDB** : pour le stockage flexible des publications et messages.
- **IA de recommandation** : Amélioration du fil d'actualité avec un algorithme de personnalisation.
- **Notifications push et WebSockets** : pour une meilleure interaction en temps réel.

---

6. Conclusion

Cher professeur,

Ce cahier des charges détaille notre projet de réseau social académique basé sur une architecture microservices. Il met en avant les fonctionnalités clés ainsi que la structure modulaire du projet. Nous avons conçu une interface intuitive et une expérience utilisateur optimisée. Nous restons disponibles pour toute recommandation et ajustement supplémentaire.

Nous vous remercions pour votre attention et votre accompagnement dans ce projet.