

# Cahier des Charges

---

**Auteurs:** Équipe AP4

**Date de création:** 03/11/2024

**Dernière mise à jour:** 03/11/2024

## Sommaire

1. [Contexte et objectifs](#)
2. [Description fonctionnelle](#)
3. [Contraintes techniques](#)
4. [Architecture du système](#)
5. [Interfaces](#)
6. [Performances et sécurité](#)
7. [Livrables](#)
8. [Évolutions futures](#)

## Contexte et objectifs

### Contexte du projet

Le projet AP4 s'inscrit dans le cadre d'un atelier professionnel pour le BTS SIO, visant à développer une application de messagerie instantanée complète. Cette application doit répondre aux besoins de communication interne d'une organisation, en offrant diverses fonctionnalités de messagerie et de partage de fichiers.

### Objectifs principaux

- Concevoir et développer une application de chat fonctionnelle avec architecture client-serveur
- Mettre en œuvre une communication en temps réel entre utilisateurs
- Implémenter différents types de conversations (individuelles, groupes, canaux)
- Permettre le partage de fichiers et la gestion des messages
- Assurer la sécurité des données et des communications
- Appliquer les bonnes pratiques de développement logiciel

### Public cible

- Utilisateurs standards : Employés ou membres d'une organisation utilisant l'application pour communiquer

## Description fonctionnelle

### Gestion des utilisateurs

1. **Création de compte**
  - Interface d'inscription avec validation des données
  - Stockage sécurisé des informations utilisateur

- Validation de l'unicité du nom d'utilisateur et de l'email

## 2. Authentification

- Connexion sécurisée avec nom d'utilisateur et mot de passe
- Gestion des sessions utilisateur
- Déconnexion et gestion de la persistance

## Types de conversations

### 1. Conversations individuelles

- Messagerie privée entre deux utilisateurs
- Historique des messages persistant

### 2. Groupes de discussion

- Création de groupes avec nom et description
- Ajout et suppression de participants
- Gestion des rôles (utilisateur, administrateur)

### 3. Canaux thématiques

- Création de canaux publics ou privés
- Gestion avancée des membres et permissions
- Organisation thématique des discussions

## Messagerie

### 1. Types de messages

- Messages texte avec formatage minimal
- Envoi et affichage d'images
- Partage de fichiers divers (documents, archives, etc.)
- Messages anonymes (auteur masqué)

### 2. Fonctionnalités avancées

- Épinglage de messages importants
- Suppression de messages
- Historique des messages

## Administration

### 1. Gestion des canaux

- Création et suppression de canaux
- Configuration des paramètres de canaux

### 2. Gestion des utilisateurs

- Attribution et révocation de rôles

# Contraintes techniques

## Technologies imposées

- **Langage de programmation** : Java 21
- **Interface utilisateur** : JavaFX 21.0.6
- **Stockage de données** : PostgreSQL 42.7.5
- **Gestion de projets** : Maven 3.6+
- **Communication réseau** : WebSocket/STOMP
- **Serveur** : Spring Boot 3.2.4
- **Logging** : Log4j 2.24.3
- **Sérialisation** : Jackson 2.16.2

## Compatibilité

- **Systèmes d'exploitation** : Windows 10/11, macOS, Linux
- **Configuration minimale** : 4 Go RAM, processeur dual-core 2 GHz

## Contraintes techniques supplémentaires

- Respect des standards de codage Java
- Architecture modulaire et maintenable
- Tests unitaires et d'intégration
- Documentation du code et des APIs

# Architecture du système

## Modules principaux

### 1. chat-common

- Modèles de données partagés
- DTOs pour la communication client-serveur
- Utilitaires communs

### 2. chat-client

- Interface utilisateur JavaFX
- Gestionnaire de connexion WebSocket
- Stockage des données
- Logique de présentation

### 3. chat-server

- Gestion des connexions WebSocket
- API de messagerie
- Services métier
- Dispatching des messages

## Flux de données

- Authentification via protocole sécurisé
- Communication en temps réel via WebSocket
- Transfert de messages et fichiers via STOMP
- Notification des événements (nouveaux messages, changements d'état, etc.)

## Interfaces

### Interface utilisateur

#### 1. Écran de connexion

- Formulaire d'authentification
- Option d'inscription
- Interface de récupération de mot de passe

#### 2. Interface principale

- Liste des conversations à gauche
- Zone de messages au centre
- Informations sur la conversation à droite
- Barre de navigation en haut

#### 3. Conversations

- Affichage des messages avec informations de l'expéditeur
- Zone de saisie de message
- Boutons d'action (joindre fichier, envoyer, etc.)
- Indicateurs d'état (messages non lus, utilisateurs en train d'écrire)

#### 4. Administration

- Interface de gestion des membres
- Paramètres de conversation
- Outils de modération

### APIs

#### 1. API WebSocket

- Endpoints pour l'envoi et la réception de messages
- Gestion des abonnements aux conversations
- Notifications d'événements système

#### 2. API de gestion

- Création et configuration des conversations
- Gestion des utilisateurs et des droits
- Administration des canaux

## Performances et sécurité

### Objectifs de performance

- Temps de réponse pour l'envoi de messages < 500ms
- Support de plusieurs dizaines d'utilisateurs simultanés
- Transfert de fichiers jusqu'à 10 Mo

## Sécurité

### 1. Authentification

- Stockage sécurisé des mots de passe
- Validation des entrées utilisateur

### 2. Communication

- Validation des droits d'accès pour chaque action
- Protection contre les injections SQL
- Vérification des types de fichiers pour les téléchargements

### 3. Protection des données

- Validation des autorisations pour l'accès aux messages
- Journalisation des actions sensibles

## Livrables

### Applications

- Client de chat exécutable (JAR avec dépendances)
- Serveur de chat déployable
- Scripts d'initialisation de la base de données

### Documentation

- Manuel d'installation et de configuration
- Guide utilisateur
- Documentation technique (architecture, API, etc.)
- Rapport de tests

### Code source

- Dépôt GitLab avec historique des commits
- Documentation du code source (Javadoc)
- Fichiers de build Maven

## Évolutions futures

### Fonctionnalités supplémentaires envisagées

- Support de la visioconférence
- Intégration avec des services externes (calendrier, email)
- Client mobile (Android, iOS)
- Système de rappels et notifications avancées

- Chiffrement de bout en bout pour les messages

## Améliorations techniques

- Optimisation des performances pour de très grands volumes de messages
- Mise en place d'une architecture distribuée pour le passage à l'échelle
- Intégration continue et déploiement automatique