

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 2
Nom, prénom : REICHHART Damien		N° candidat : 02149931394
Épreuve ponctuelle <input type="checkbox"/>	Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : 28/04/2025
Organisation support de la réalisation professionnelle Du code exécuté en client lourd sur un client.		
Intitulé de la réalisation professionnelle Atelier de professionnalisation 4		
Période de réalisation : Septembre 2024 – Janvier 2025		Lieu : Cci Campus Alsace Strasbourg
Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		
Compétences travaillées <ul style="list-style-type: none"> X Concevoir et développer une solution applicative X Assurer la maintenance corrective ou évolutive d'une solution applicative X Gérer les données 		
Conditions de réalisation¹ (ressources fournies, résultats attendus) <p>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.</p> <p>Objectifs principaux :</p> <ul style="list-style-type: none"> - 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 		
Description des ressources documentaires, matérielles et logicielles utilisées²		
Ressources documentaires : <ul style="list-style-type: none"> • Cahier des charges et spécifications fonctionnelles fournies 		
Ressources matérielles : <ul style="list-style-type: none"> • Postes de développement : Ordinateurs avec spécifications minimales : <ul style="list-style-type: none"> - Processeur : Dual-core 2GHz ou supérieur - Mémoire : 4Go RAM minimum (8Go recommandé) - Espace disque : 500Mo minimum d'espace libre - Systèmes d'exploitation supportés : Windows 10/11, macOS 10.15+, ou Linux 		

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

Ressources logicielles :

- **Environnement de développement intégré (EDI) :** Visual Studio Code, PhpStorm
- **Langages de programmation :**
 - Java 21 (JDK 21) : Langage principal pour le client et le serveur
 - SQL : Pour les opérations de base de données
- **Frameworks et bibliothèques :**
- **Côté serveur :**
 - Spring Boot 3.2.4 : Framework principal du serveur
 - Spring WebSocket : Gestion des communications en temps réel
 - Spring Messaging : Traitement des messages STOMP
 - Log4j 2.24.3 : Journalisation des événements serveur
- **Côté client :**
 - JavaFX 21.0.6 : Interface utilisateur graphique
 - Spring WebSocket Client : Connectivité WebSocket
 - STOMP Client : Communication structurée avec le serveur
 - Log4j 2.24.3 : Journalisation des événements client
 - PostgreSQL JDBC 42.7.5 : Connexion à la base de données locale
- **Module commun :**
 - Jackson 2.16.2 : Sérialisation/désérialisation JSON
 - Hibernate Validator 8.0.1 : Validation des données
 - Jakarta Validation API 3.0.2 : API de validation
- **Outils de build et déploiement :**
 - Maven 3.13.0 : Gestion des dépendances et automatisation des builds
 - JUnit 5.11.4 : Tests unitaires
 - Mockito 5.14.2 : Framework de mock pour tests
 - JaCoCo 0.8.12 : Analyse de couverture de code
- **Docker : Conteneurisation du serveur**
 - JPackage : Création d'installateurs natifs pour le client
 - Make : Automatisation via Makefile
- **Base de données :**
 - PostgreSQL : Système de gestion de base de données relationnelle
- **Environnement d'exécution :**
 - JVM 21 : Machine virtuelle Java
 - Docker : Conteneurisation pour le déploiement serveur
 - Windows et Linux : Plateformes supportées avec installateurs natifs
- **Sécurité et communication :**
 - WebSockets : Protocole de communication bidirectionnelle
 - STOMP : Protocole de messagerie orienté texte
 - JSON : Format d'échange de données

Modalités d'accès aux productions³ et à leur documentation⁴

Accès aux productions accessible sur gitlab à l'url suivante : <https://gitlab.com/DamienReichhart/CCI-BTS-SIO-23-25-Atelier-Professionnalisation-4>

La documentation du projet est également accessible à cette url : <https://damien-reichhart.fr/>

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS

SESSION 2024

**ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)**

Épreuve E5 - Conception et développement d'applications (option SLAM)

³ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

Table des matières

Table des matières	3
Contexte	5
Objectifs principaux	5
Description fonctionnelle	5
Gestion des utilisateurs.....	5
Types de conversations	6
Messagerie.....	6
Administration.....	6
Gestion de projet	7
Formalisme des commits.....	7
Gestion des versions du code	7
Workflow de développement	7
Normes de codage	8
Conventions de nommage	8
Formatage et style	8
Accolades	8
Structure de l'application	9
Documentation technique.....	11
Diagrammes de classe	11
Chat Common	11
Chat Server.....	12
Chat Client.....	13
Diagramme de contexte statique	19
Diagramme de package	19
Diagramme de cas d'utilisation.....	20
Diagramme de déploiement	21
Diagramme d'activité pour envoyer un message	22
Diagramme de représentation du workflow CI CD	23
Diagramme de séquence pour quitter un chat	24
Diagramme de séquence pour initialiser une connexion.....	24
Diagramme de séquence pour partager un fichier.....	25
Diagramme de séquence pour rejoindre un chat	25
Diagramme de séquence pour supprimer un message.....	26
Diagramme de séquence pour pin un message.....	26

Diagramme de séquence pour envoyer un message textuel	27
Captures d'écran de l'application.....	28
Connexion	28
Création de compte.....	29
Page d'accueil	30
Créer un canal.....	30
Rejoindre un groupe	31
Editer son profil.....	31
Gérer les rôles d'un canal.....	32

Contexte

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, membres d'une organisation ou particuliers 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
- 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
- Modération du contenu

2. Gestion des utilisateurs

- Attribution et révocation de rôles

Gestion de projet

Le projet a été mené selon la méthodologie Waterfall, ou cascade en français.

Dans ce cadre, l'ensemble des tâches et des étapes du projet ont été planifiées en amont, puis structurées sous forme de tickets et de jalons (Milestones).

Chaque ticket s'est vu attribuer une date buttoir correspondante à laque la fonctionnalité devait être mise en place. Cela permettait un suivi clair de l'avancement du projet

Formalisme des commits

Pour ce projet, un formalisme des commits et des tickets a été mis en place afin de garantir un suivi de projet de qualité.

Pour cela, le formalisme suivant a été défini avec les préfixes suivant :

[FIX]	Définit une correction dans un message de commit
[FEATURE]	Définit que le ticket représente une fonctionnalité ou que le commit représente l'ajout de Fonctionnalité
[RELEASE]	Définit que le commit représente une release
[DOC]	Définit que le ticket ou le commit représente un ajout ou une modification de documentation
[HOTFIX]	Définit que le ticket ou le commit a réalisé une correction de bug rapide et urgente sur une des branches de production (preprod / main)

De plus les messages devaient être clair et structurés.

Gestion des versions du code

La gestion des branches suit un modèle adapté à la méthodologie en cascade :

- main : Branche principale, contient la version stable et validée du projet
- dev : Branche de développement intégrant progressivement les fonctionnalités
- feature/* : Branches temporaires pour le développement des fonctionnalités
- release/* : Branches de préparation des versions à déployer
- hotfix/* : Branches pour les corrections urgentes

Workflow de développement

1. Création d'un ticket détaillant la tâche à réaliser
2. Attribution du ticket
3. Création d'une branche dédiée à partir de dev
4. Développement avec commits réguliers
5. Tests unitaires locaux
6. Merge dans dev après validation
7. Tests sur dev
8. Création d'une branche de release pour regrouper les fonctionnalités développées
9. Tests système et validation sur la branche de release
10. Merge dans main après validation complète

Normes de codage

Conventions de nommage

Package :

- Noms en minuscules
- Notation en arborescence (com.ap4.client.controllers)
- Organisation par fonctionnalité ou couche technique

Classes :

- Notation PascalCase (première lettre de chaque mot en majuscule)
- Noms significatifs reflétant la responsabilité de la classe
- Exemples : UserController, ChatService, MessageDAO

Interfaces :

- Préfixe "I" pour les interfaces (ex: IUserService)
- Notation PascalCase Décrivent un comportement ou un contrat

Méthodes :

- Notation camelCase (première lettre en minuscule, puis majuscule pour chaque nouveau mot)
- Verbes d'action décrivant ce que fait la méthode
- Exemples : getUserById(), sendMessage(), createNewChat()

Variables:

- Notation camelCase
- Noms descriptifs évitant les abréviations obscures
- Exemples : userName, messageContent, isConnected

Constantes :

- Notation SNAKE_CASE en majuscules Mots séparés par des underscores
- Exemples : MAX_MESSAGE_LENGTH, DEFAULT_PORT

Fichiers FXML :

- Notation camelCase Décrivent l'écran ou le composant qu'ils représentent
- Exemples : login.fxml, chatListCell.fxml

Formatage et style

- Indentation et espacement
- Indentation de 4 espaces (pas de tabulations)
- Limite de 120 caractères par ligne
- Espacement autour des opérateurs (x + y plutôt que x+y)
- Espacement après les virgules (method(a, b, c) plutôt que method(a,b,c))

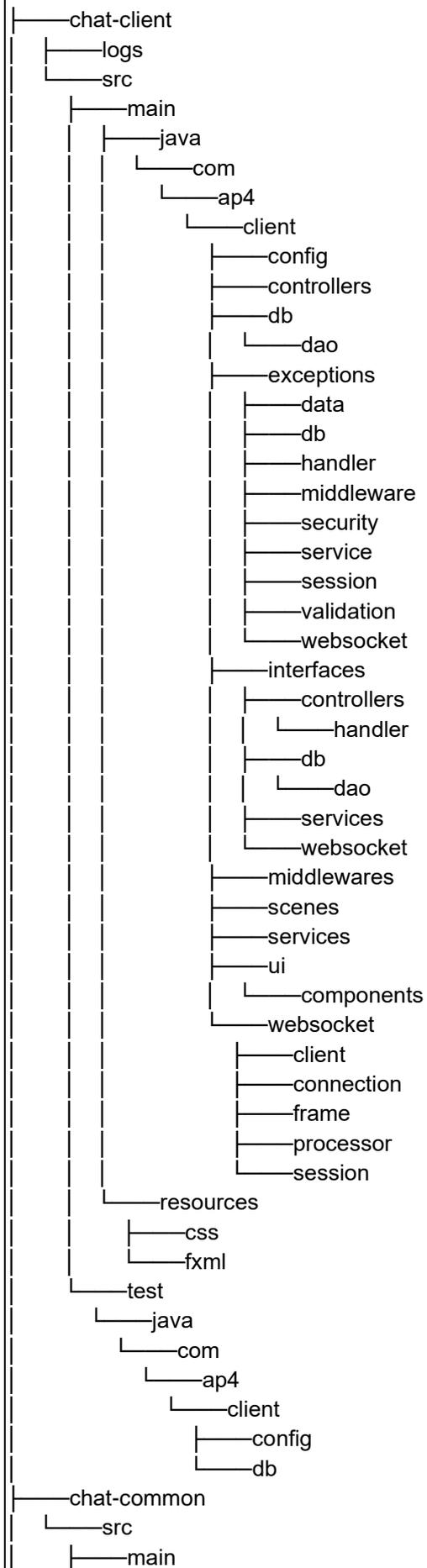
Accolades

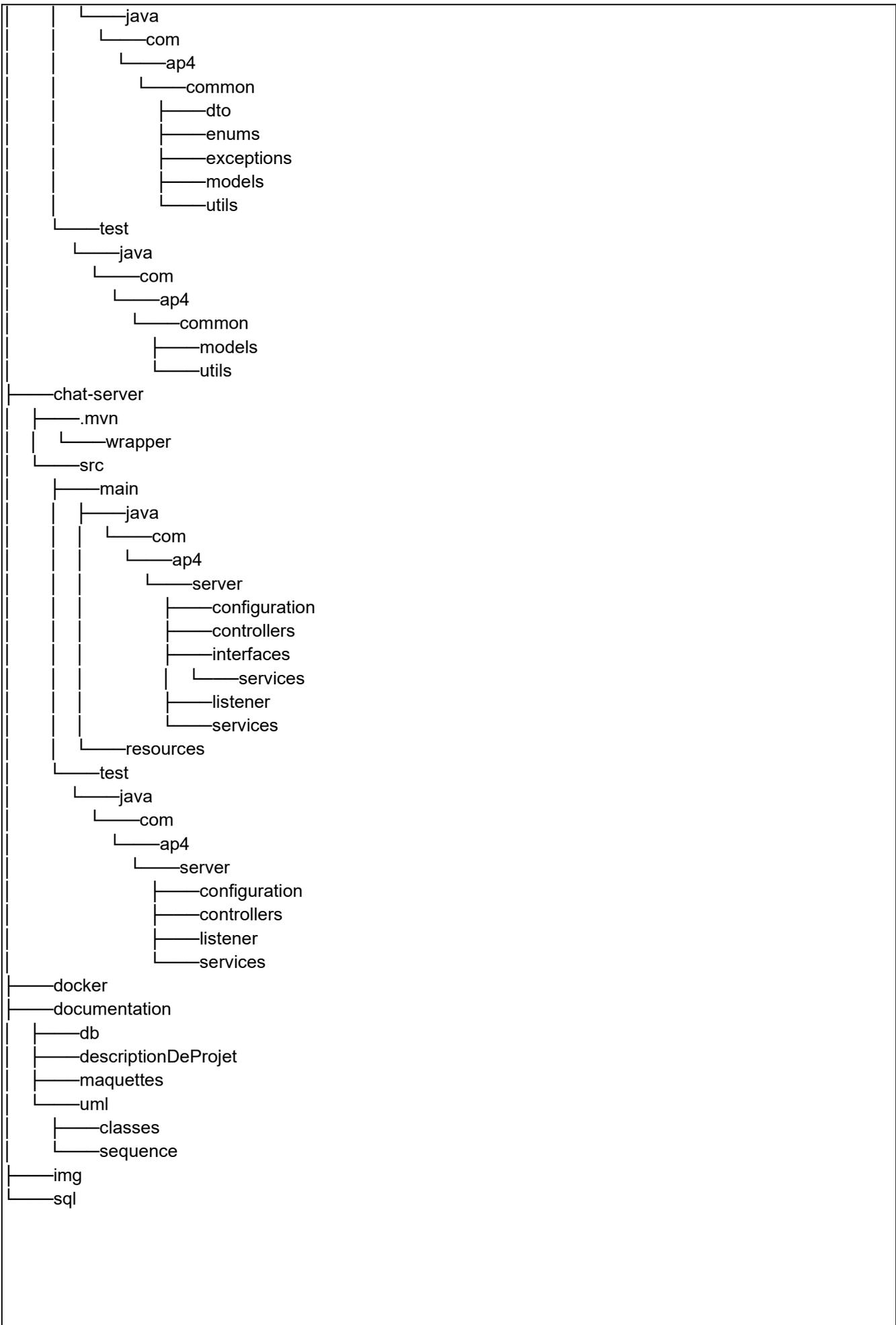
Style Java standard :

```
if (condition) {  
    // Code  
} else {  
    // Code  
}
```

Structure de l'application

C:.



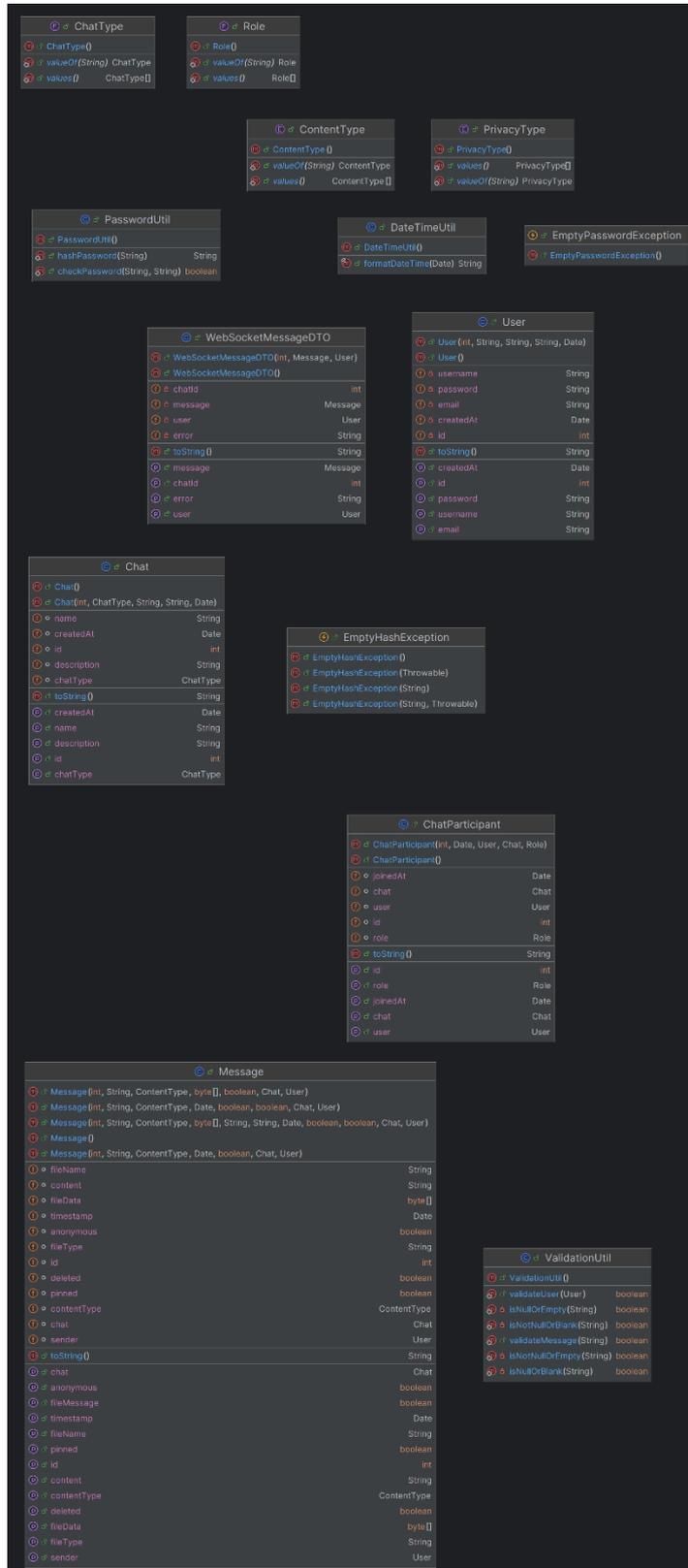


Documentation technique

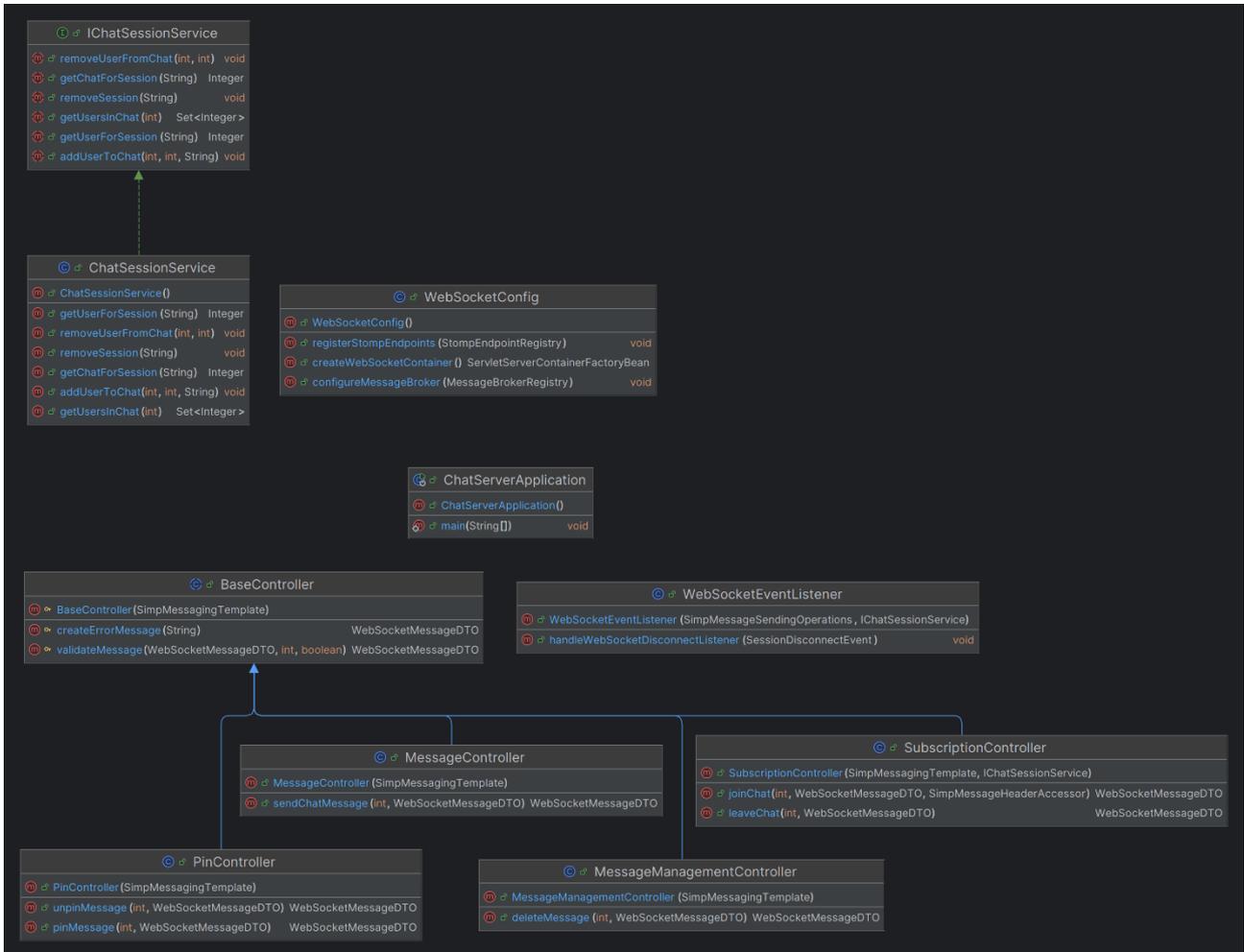
Diagrammes de classe

Voici les différents diagrammes de classe de l'application :

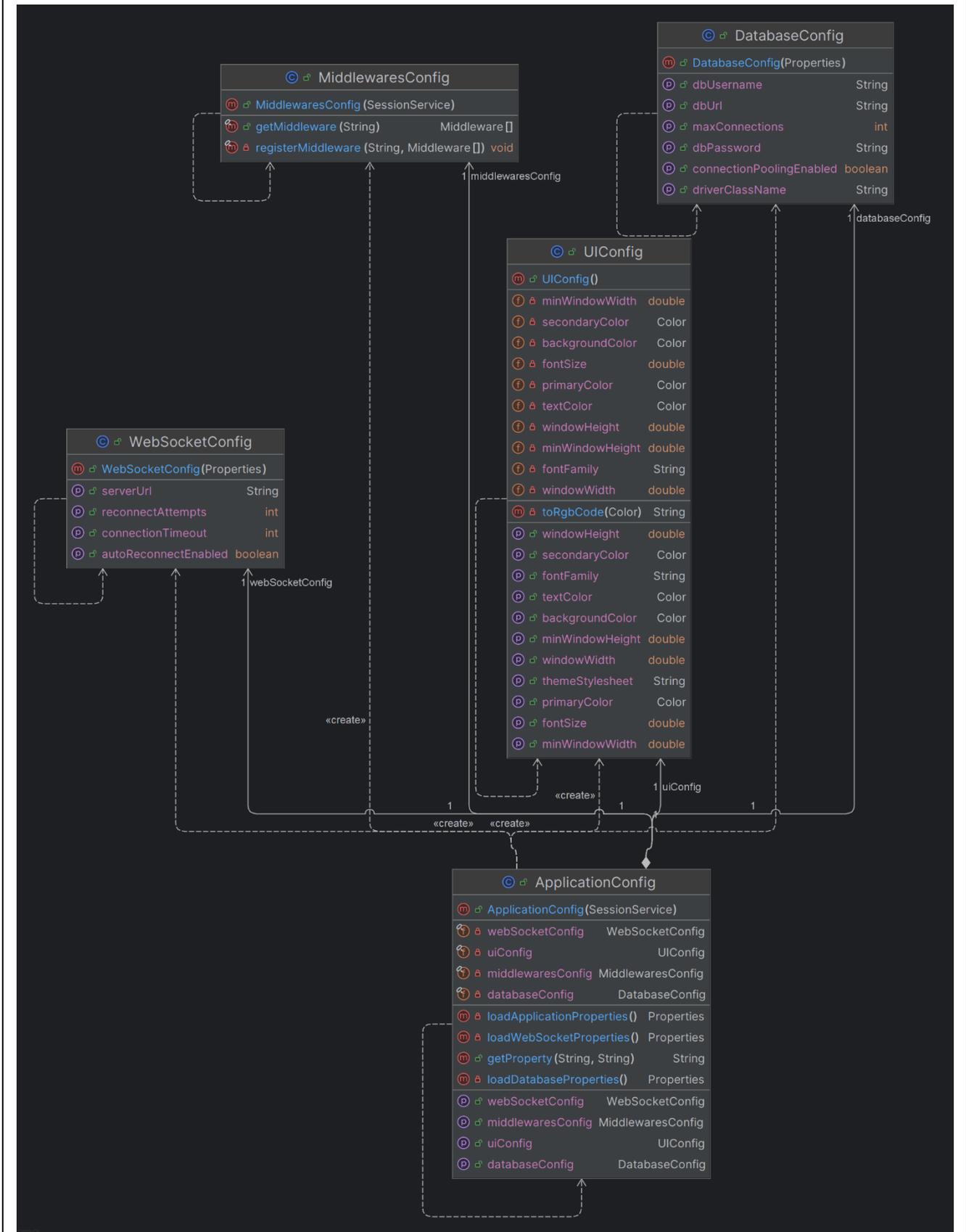
Chat Common

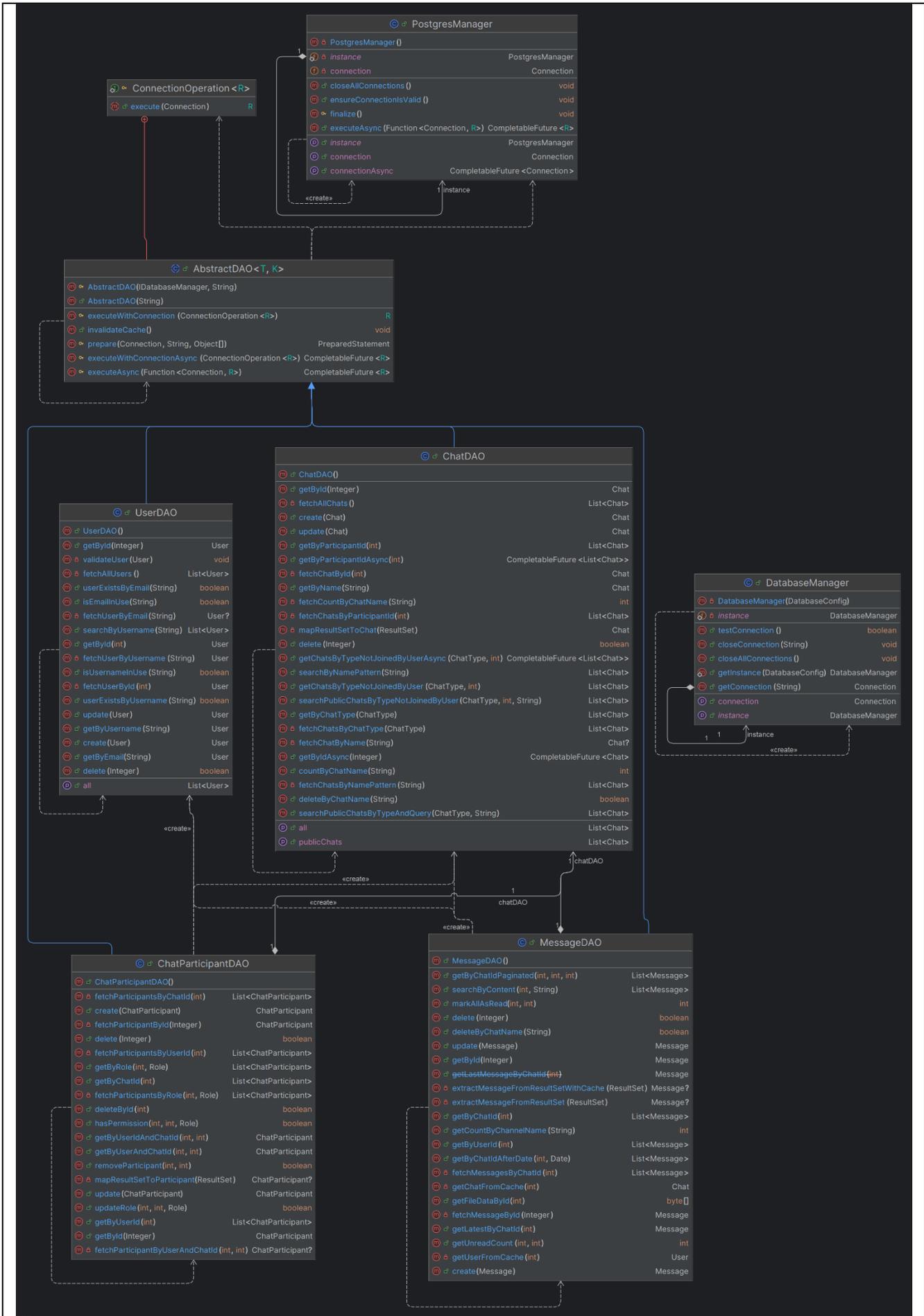


Chat Server



Chat Client





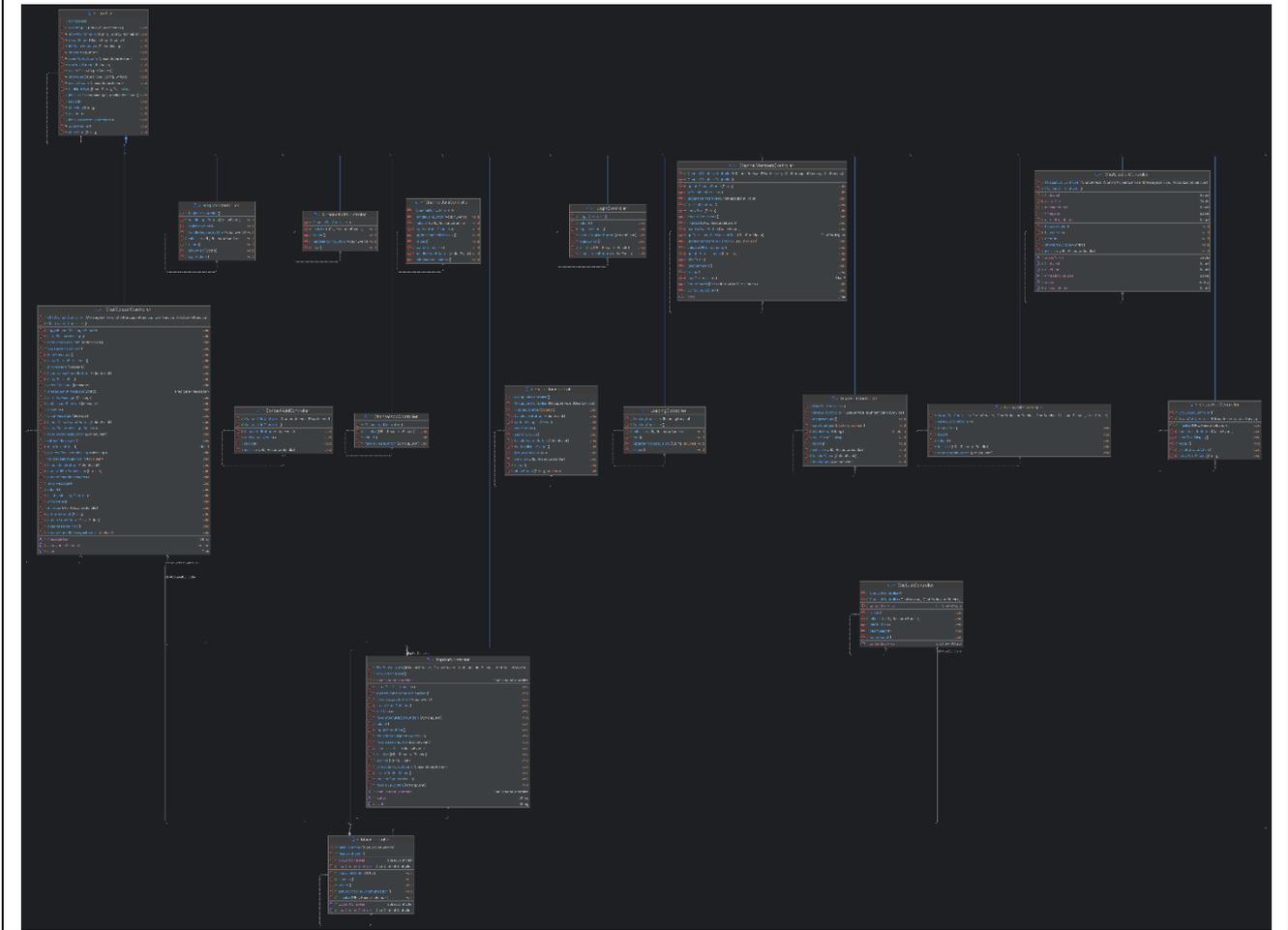
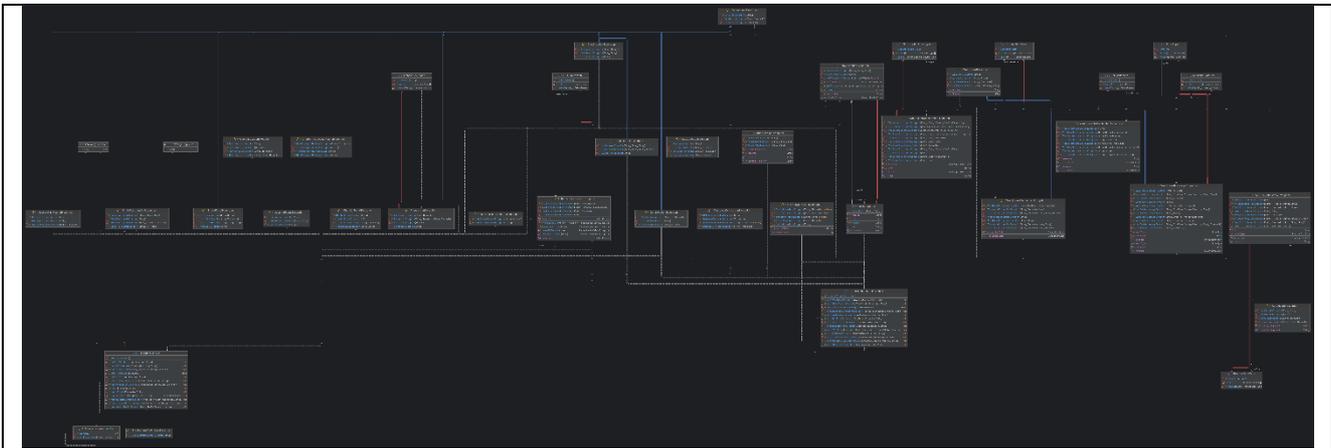


Diagramme de contexte statique

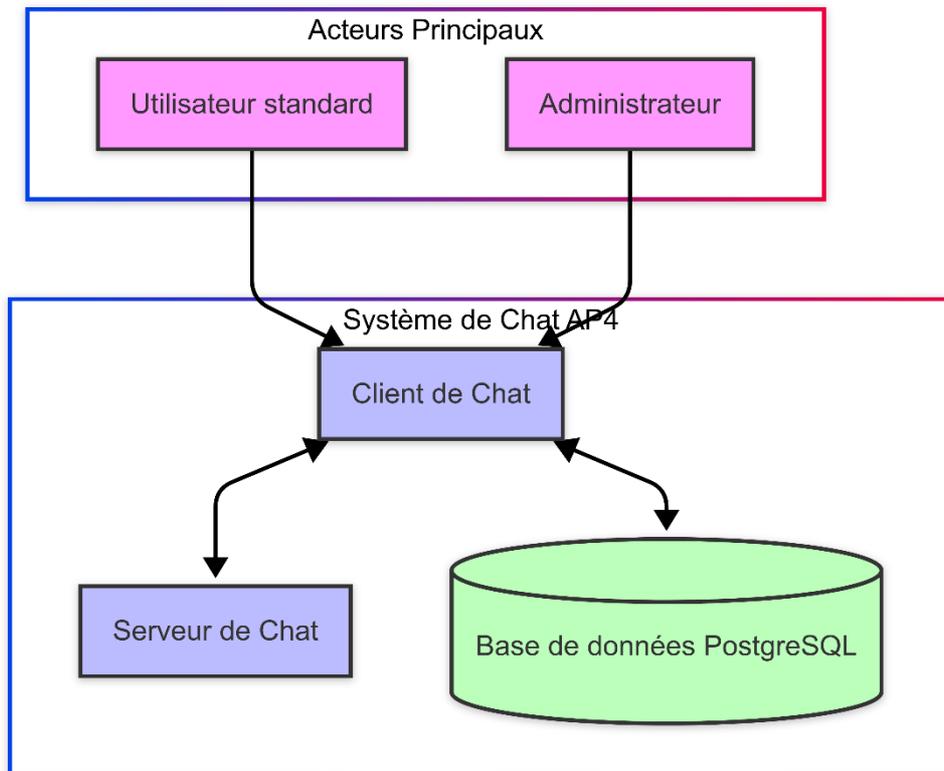


Diagramme de package

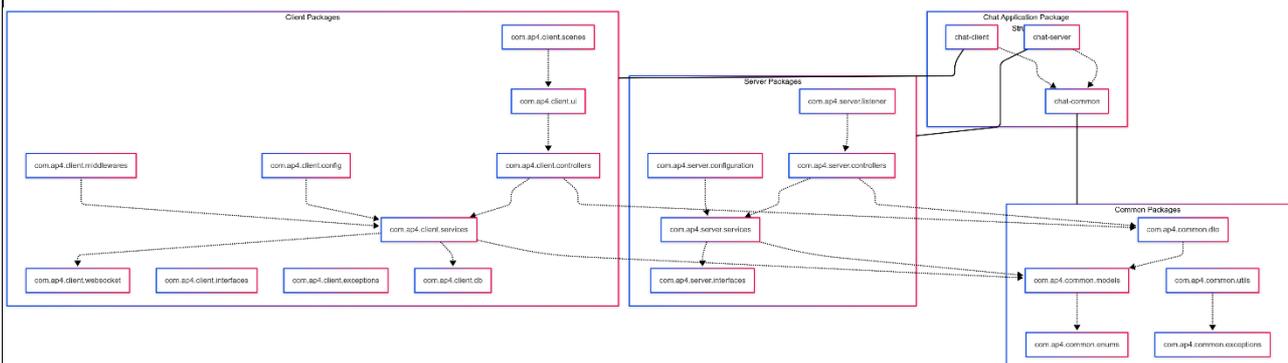


Diagramme de cas d'utilisation

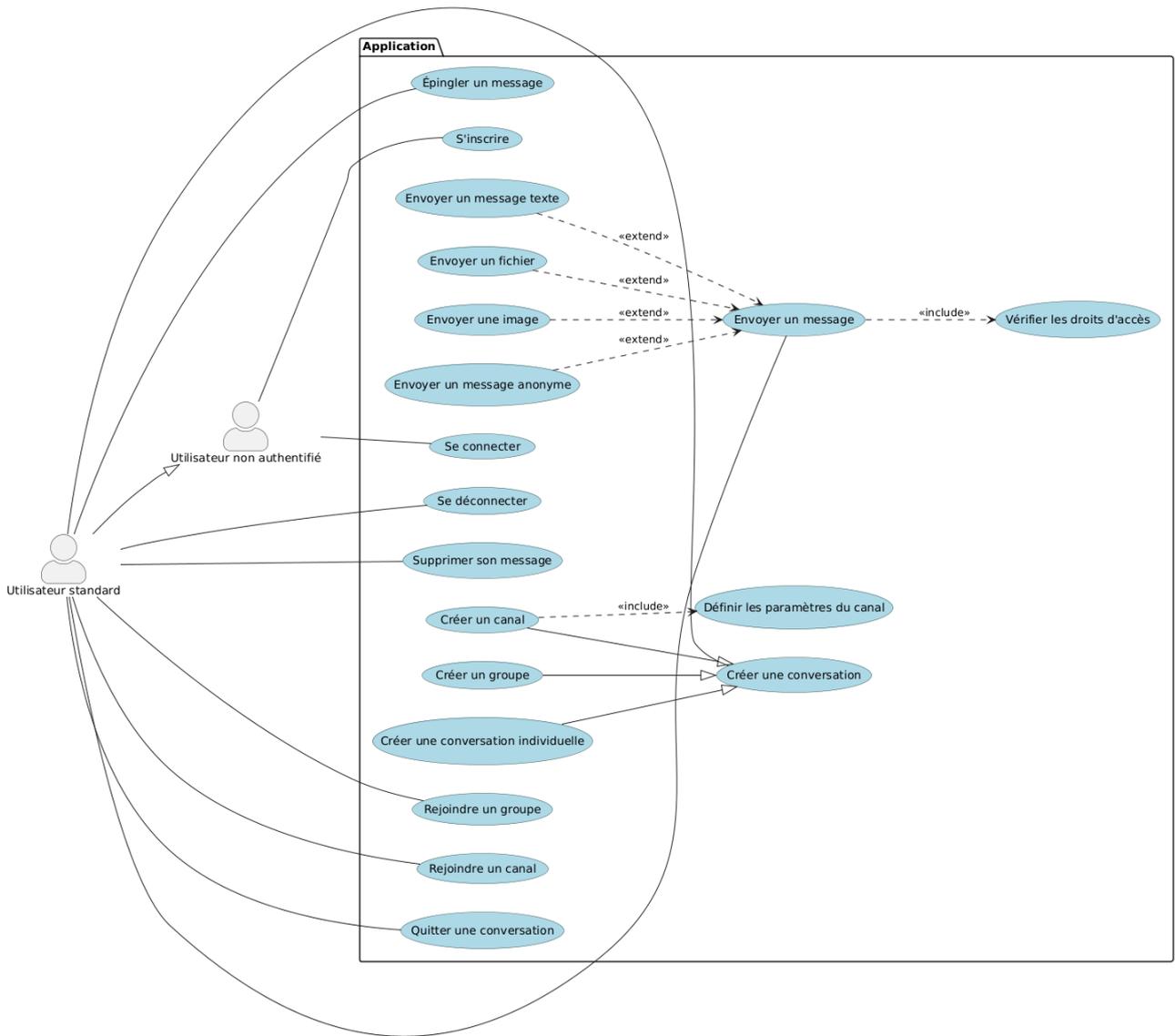


Diagramme de deployment

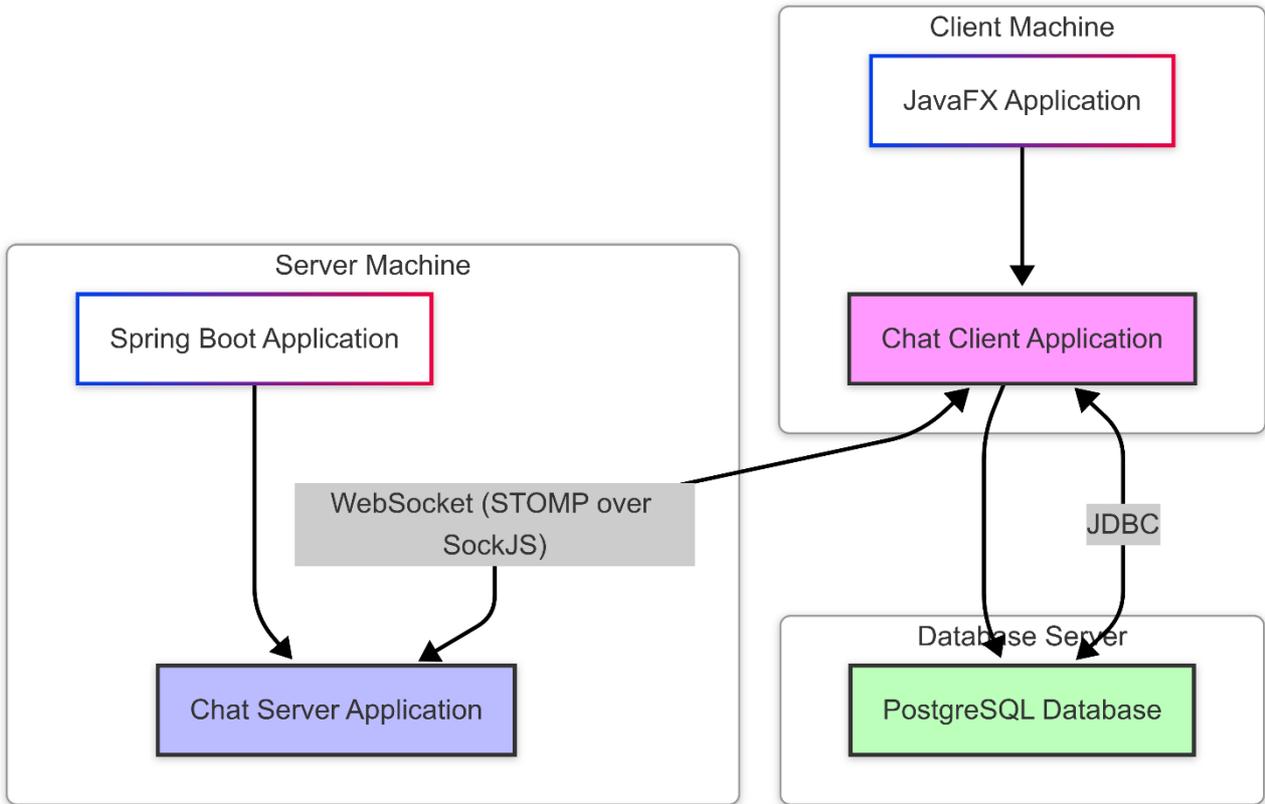


Diagramme d'activité pour envoyer un message

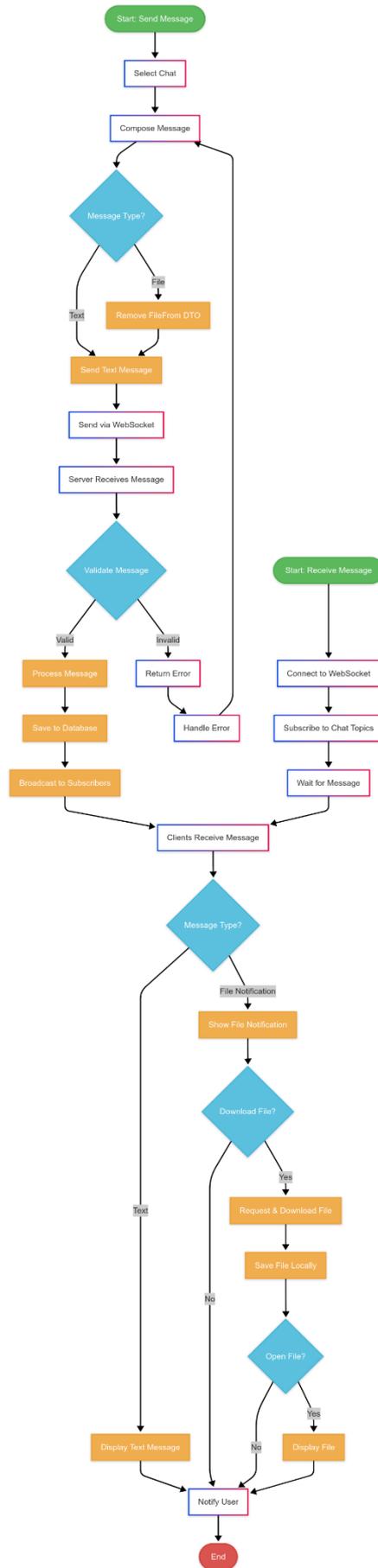


Diagramme de représentation du workflow CI CD

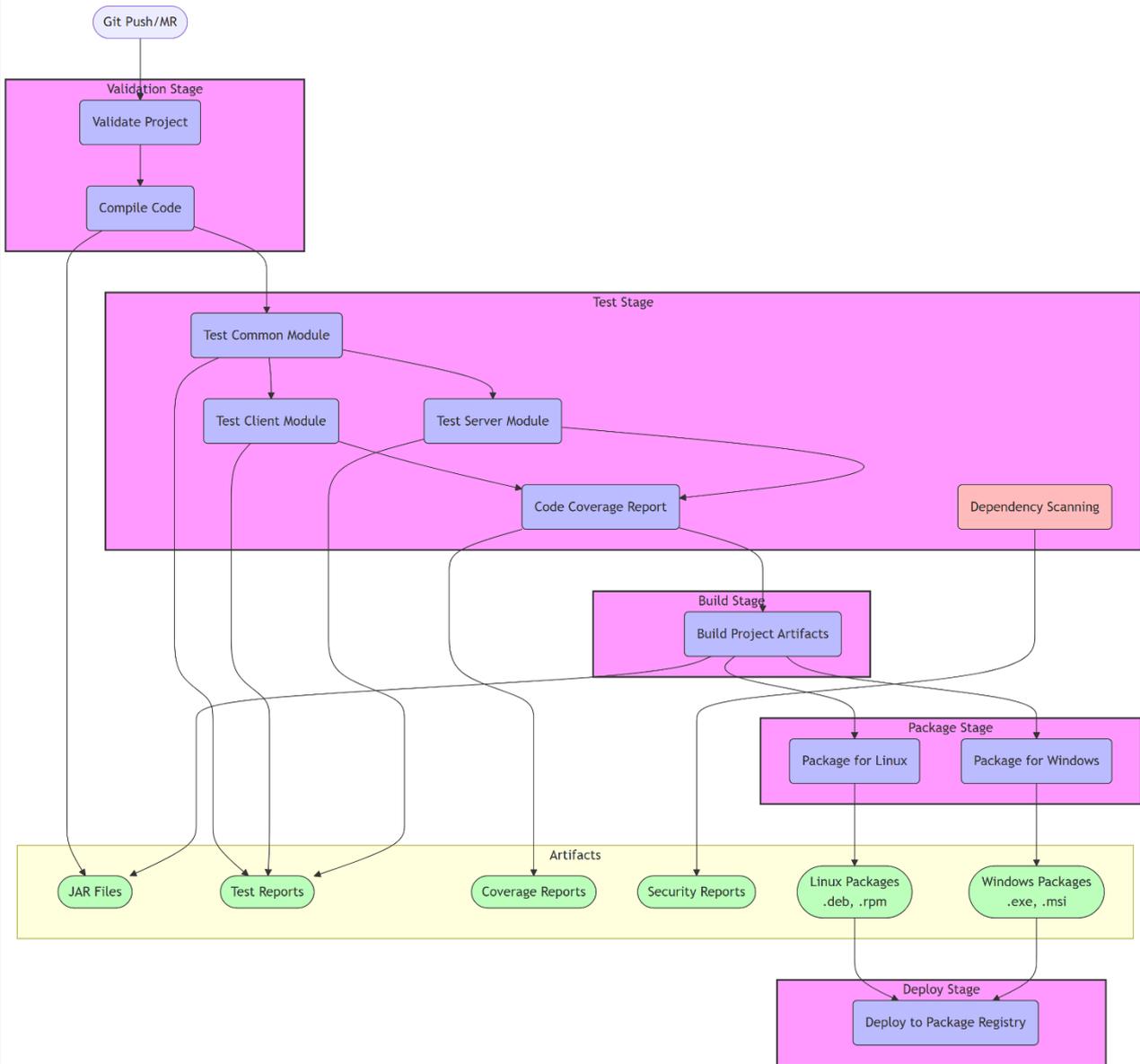


Diagramme de séquence pour partager un fichier

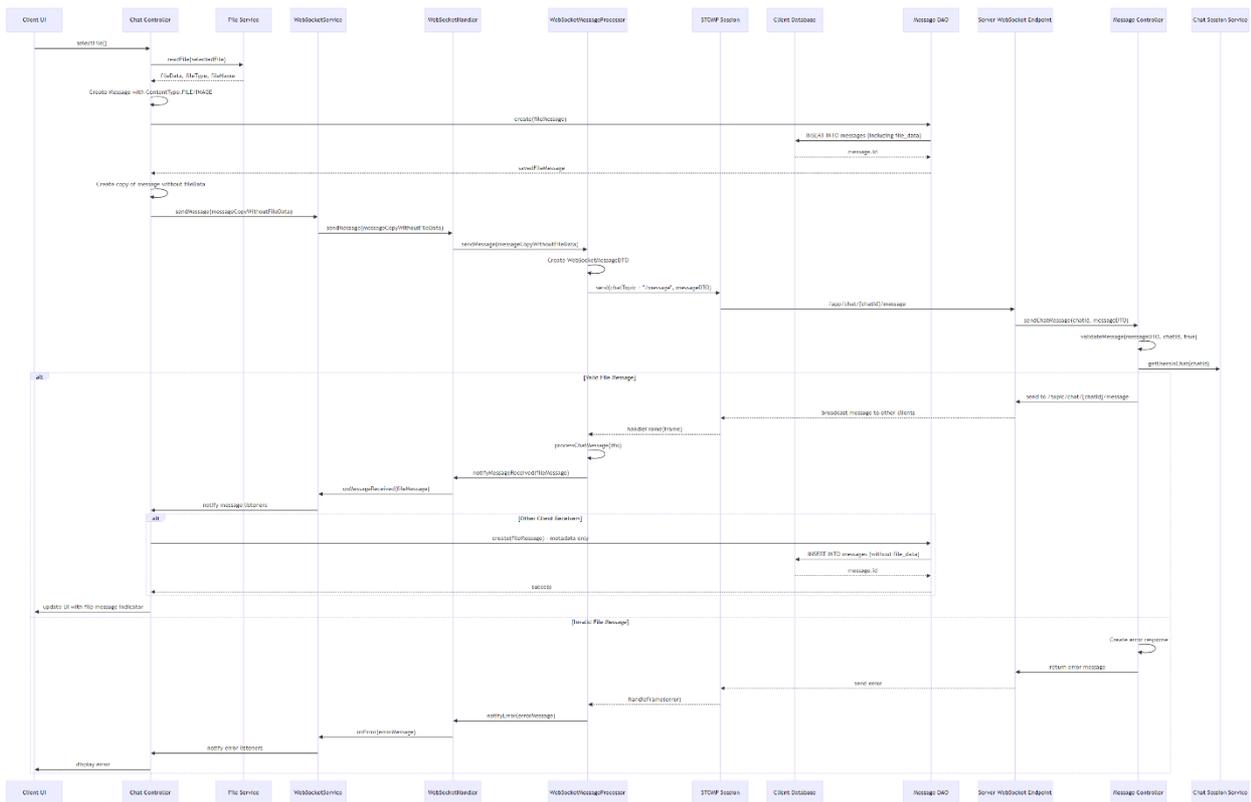


Diagramme de séquence pour rejoindre un chat

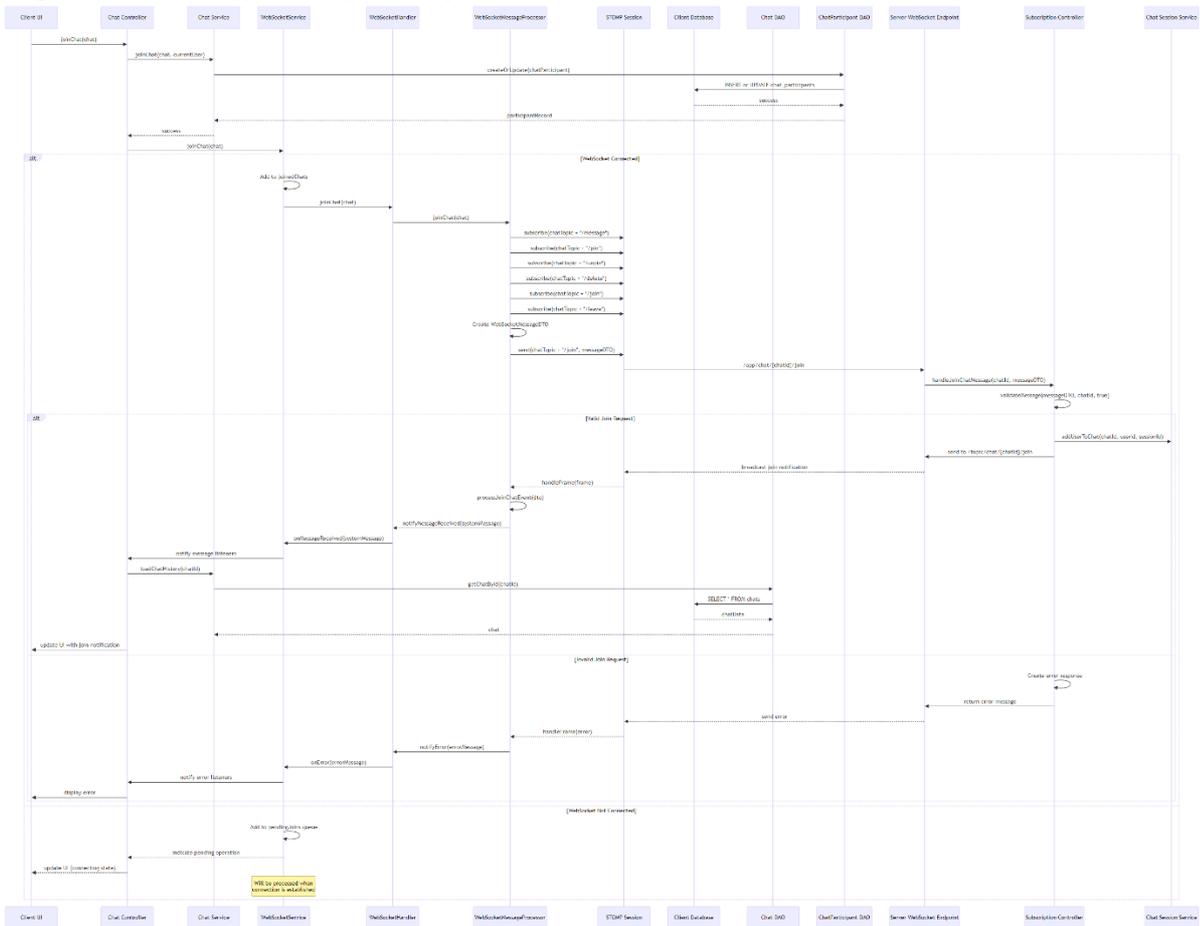


Diagramme de séquence pour supprimer un message

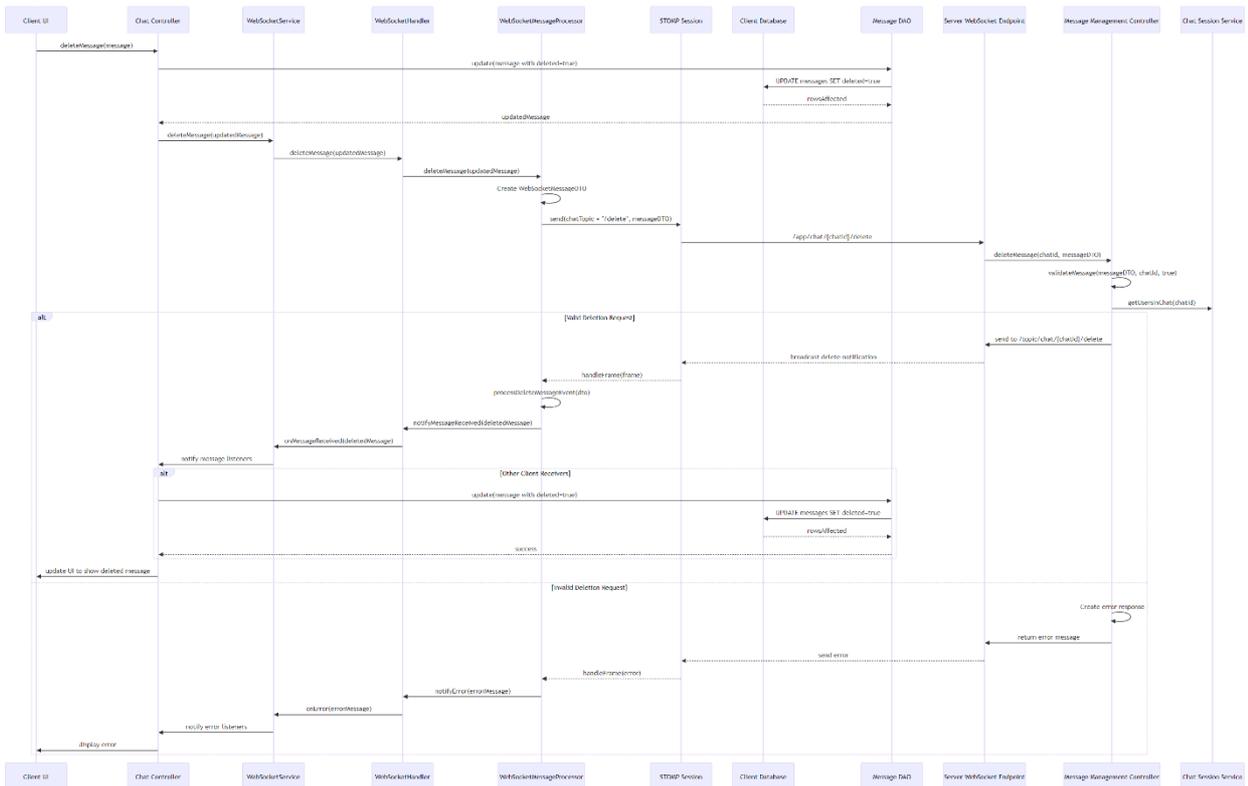


Diagramme de séquence pour pin un message

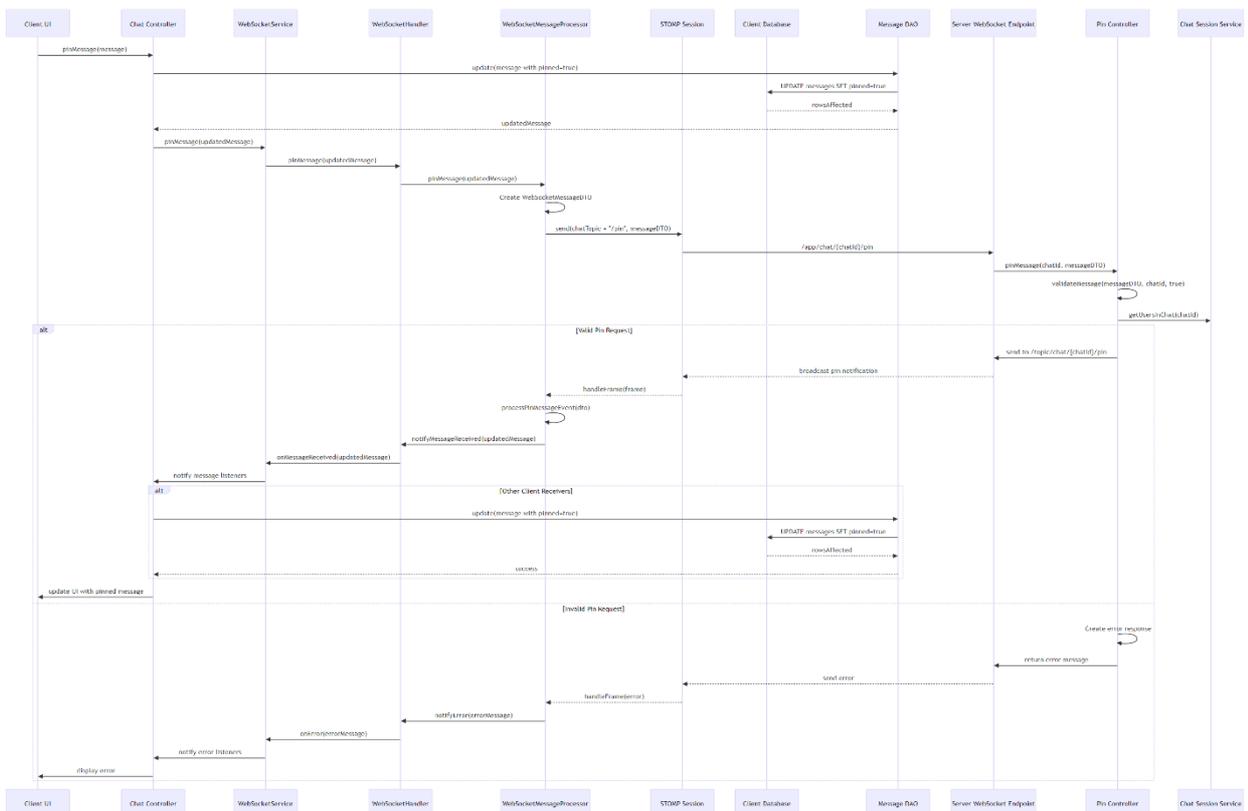
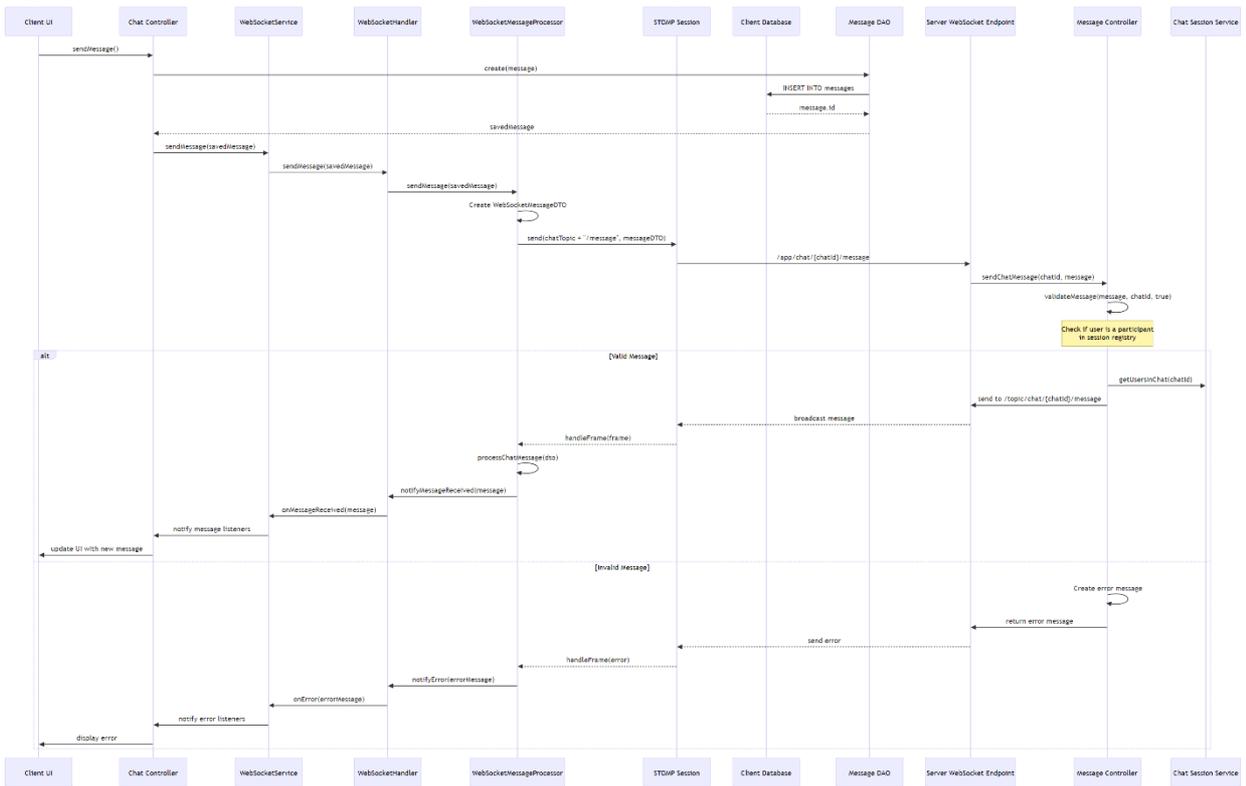


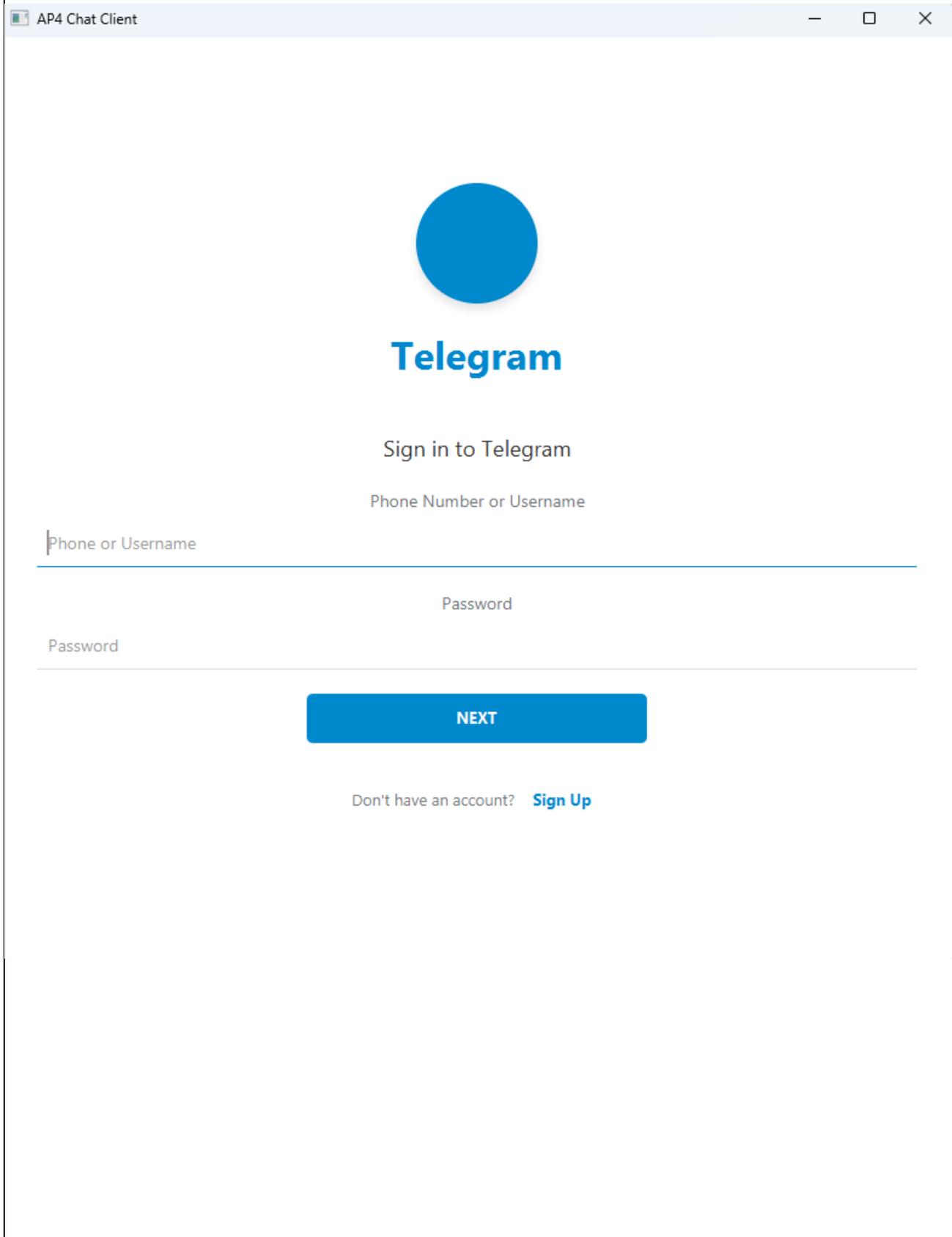
Diagramme de séquence pour envoyer un message textuel



Captures d'écran de l'application

Ci-dessous, vous pourrez trouver des captures d'écran correspondantes a l'application :

Connexion



The screenshot shows a window titled "AP4 Chat Client" with standard window controls (minimize, maximize, close). The main content is the Telegram login interface. At the top center is a blue circular logo. Below it is the word "Telegram" in a bold, blue, sans-serif font. Underneath is the text "Sign in to Telegram" followed by "Phone Number or Username". There are two input fields: the first is labeled "Phone or Username" and the second is labeled "Password". Below the input fields is a prominent blue button with the text "NEXT" in white. At the bottom, there is a link that says "Don't have an account? Sign Up".



Telegram

Sign up for Telegram

Username

Enter your username

Email

Enter your email

Password

Enter your password

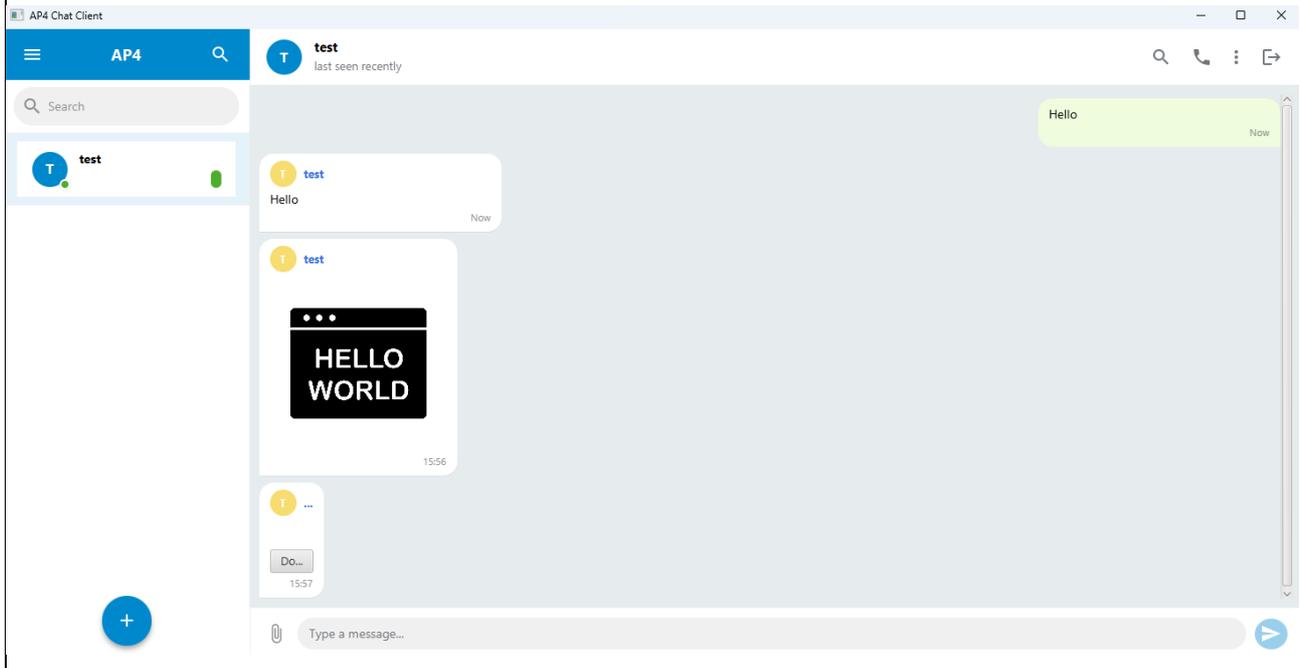
Confirm Password

Confirm your password

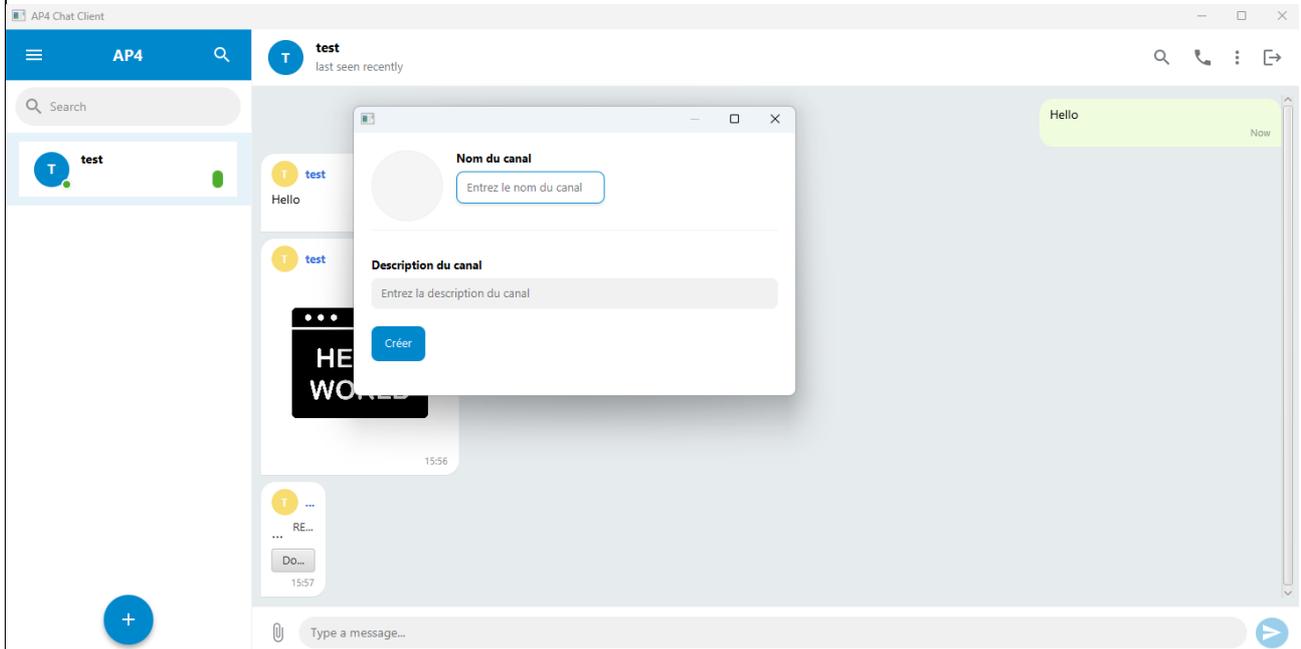
SIGN UP

Already have an account? [Sign In](#)

Page d'accueil



Créer un canal



Rejoindre un groupe

Rejoindre un groupe

Groupes disponibles

dadaz -

groupe01 -

Editer son profil

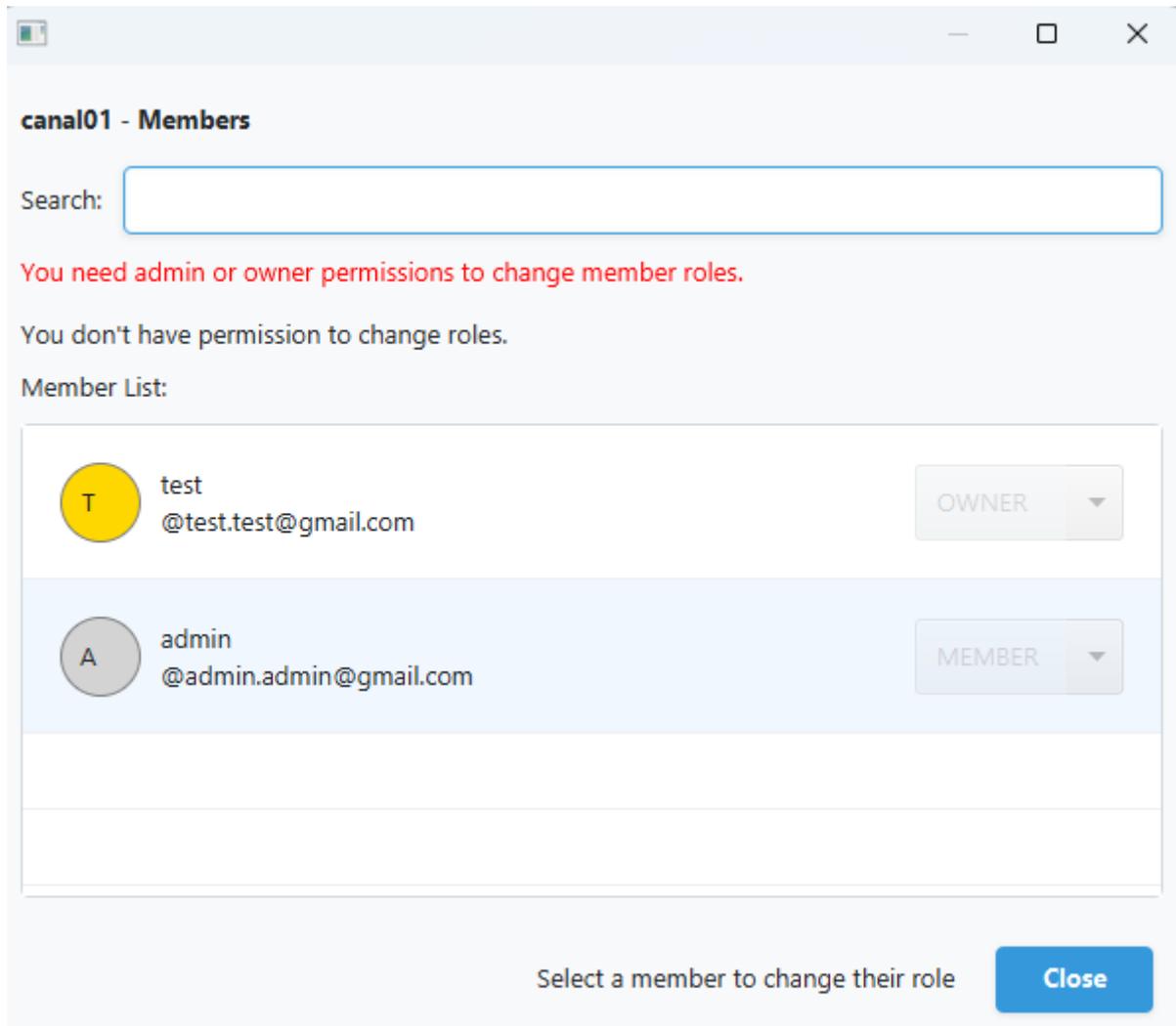
Modifier votre profil

 **Nom d'utilisateur**
admin

Email

Changer le mot de passe

Gérer les rôles d'un canal



canal01 - Members

Search:

You need admin or owner permissions to change member roles.

You don't have permission to change roles.

Member List:

	test @test.test@gmail.com	OWNER
	admin @admin.admin@gmail.com	MEMBER

Select a member to change their role [Close](#)