

Kris TOURE
Irvyn COURSIMAULT

Rapport

Rapport :

Ce rapport présente une suite de programmes développés en Java et JavaFX pour implémenter un système de serveur-client. Ces programmes comprennent un serveur, un client console, et une application de chat graphique.

1. Server

Le programme du serveur (Server.java) crée un serveur socket qui accepte les connexions des clients. Il utilise un pool d'exécution pour gérer les connexions des clients de manière concurrente. Le serveur permet la diffusion de messages aux clients connectés, la gestion des connexions et déconnexions, ainsi que la gestion des commandes utilisateur.

2. ConnectionHandler

La classe ConnectionHandler gère la communication avec un client particulier. Elle est responsable de recevoir les messages du client, de traiter les commandes utilisateur, et de gérer les interactions liées au suivi d'utilisateurs. Chaque instance de ConnectionHandler est exécutée dans un thread séparé.

Limitations et Axes d'Amélioration

Malheureusement, en raison de contraintes de temps, certaines fonctionnalités n'ont pas pu être entièrement implémentées. Notamment, les commandes telles que /like, /unlike, /delete n'ont pas été développées. Cependant, le principe de fonctionnement de ces commandes aurait été similaire aux autres commandes implémentées, et leur absence ne compromet pas la structure globale de l'application.

Stockage des Messages dans une Base de Données

L'une des fonctionnalités que nous aurions aimé implémenter est le stockage des messages dans une base de données. Cela aurait permis de conserver l'historique des messages même lorsque les clients se déconnectent et de permettre aux nouveaux clients de voir les messages existants à leur arrivée.

Le stockage des messages dans une base de données aurait offert une solution plus robuste pour la gestion des données persistantes, permettant ainsi une expérience utilisateur plus complète.

Conclusion

Les programmes développés offrent un système de chat fonctionnel avec une interface graphique. Ils démontrent l'utilisation de sockets pour la communication entre clients et serveur, la gestion concurrente des connexions, la sérialisation JSON, et l'implémentation d'une interface utilisateur en JavaFX.