ZAKARIA WAKACH

**2023**

**jUIN**



**CHAT APP**

CLIENT/SERVER JAVA APP

**Sommaire**

1. **Inroduction**
2. **Etapes**
3. **Explication**
4. **Execution**

# INTRODUCTION

C'est avec grand plaisir que nous vous présentons ce rapport final concernant notre projet de fin d'année portant sur le développement d'une application de chat en utilisant le langage Java. Nous

tenons tout d'abord à vous exprimer notre profonde gratitude pour votre guidance, votre soutien et votre expertise tout au long de ce projet. Votre engagement et votre dévouement envers notre réussite académique ont été inestimables, et nous sommes sincèrement reconnaissants de vous avoir eu comme professeur.

L'objectif principal de notre projet était de concevoir et de développer une application de chat interactive et conviviale en utilisant les principes de programmation objet et les fonctionnalités

offertes par le langage Java. Cette application permet aux utilisateurs de communiquer en temps réel, d'échanger des messages et de partager des fichiers, tout en offrant une interface utilisateur intuitive et attrayante.

Dans ce rapport, nous présenterons en détail les différentes phases de développement de notre application de chat. Nous aborderons la conception de l'architecture logicielle, l'implémentation des

fonctionnalités clés, la gestion de la sécurité des données, ainsi que les défis techniques auxquels nous avons été confrontés et les solutions que nous avons mises en œuvre. Nous partagerons également nos réflexions sur les améliorations potentielles et les perspectives d'avenir de cette application.

Nous tenons à souligner que notre succès dans la réalisation de ce projet n'aurait pas été possible sans votre expertise et votre encadrement. Votre soutien continu, vos conseils avisés et votre disponibilité ont joué un rôle essentiel dans notre apprentissage et notre progression. Vos encouragements constants ont renforcé notre confiance et notre motivation tout au long du processus de développement.

Nous tenons également à remercier nos camarades de classe et nos proches qui nous ont soutenus tout au long de ce projet, ainsi que notre institution pour nous avoir offert l'opportunité de mettre en pratique nos compétences et nos connaissances acquises au cours de notre formation.

Nous espérons que ce rapport témoignera de notre engagement, de notre diligence et de notre passion pour le développement d'applications logicielles de qualité. Nous sommes fiers de vous présenter les résultats de notre travail et nous espérons que vous les trouverez à la hauteur de vos attentes.

Encore une fois, nous vous remercions sincèrement pour votre précieuse contribution à notre apprentissage et à notre développement en tant que futurs professionnels de l'informatique. Nous

sommes honorés d'avoir eu la chance d'être vos étudiants et nous espérons que vous continuerez à inspirer et à guider de nombreux autres étudiants dans leurs projets futurs.

# LES ETAPES

## 1 LE SERVEUR

Un serveur local avec Java est une application qui permet à d'autres programmes

(clients) de se connecter et de communiquer avec lui sur la même machine. Il écoute les demandes de connexion des clients, traite les requêtes et renvoie les réponses. Il peut être utilisé pour des applications réseau telles que le partage de fichiers, les jeux en ligne ou la communication entre différentes parties d'une application.

## 2 Le Logique du connection a le serveur est d envoyer des message

Pour se connecter à un serveur en utilisant le nom d'hôte et le port, le processus peut être décrit en :

1.Le client crée une instance de connexion en spécifiant le nom d'hôte et le port du serveur.

2.Le client établit une connexion réseau avec le serveur en utilisant le nom d'hôte et le port spécifiés.

3.Le serveur accepte la demande de connexion du client et établit un canal de communication bidirectionnel.

4.Le client et le serveur peuvent maintenant échanger des données à travers cette connexion établie.

# LES ETAPES

Le processus d'envoi et de réception de messages entre deux clients dans un serveur :

1.Le client A se connecte au serveur et envoie un message contenant le contenu à transmettre au client B.

2.Le serveur reçoit le message du client A et l'identifie comme étant destiné au client B.

3.Le serveur vérifie si le client B est actuellement connecté et disponible.

4.Si le client B est connecté, le serveur transmet le message du client A au client B.

5.Le client B reçoit le message du serveur et l'affiche à l'écran.

6.Le processus se répète lorsque le client B envoie une réponse au client A, en inversant les rôles.

## 3 Les intefaces graphique

Les interfaces graphiques jouent un rôle crucial dans une application de chat en offrant une expérience utilisateur conviviale et interactive. Tout d'abord, elles

permettent aux utilisateurs de se connecter et de s'authentifier, en fournissant un moyen intuitif d'entrer leurs identifiants. Ensuite, elles affichent la liste des contacts disponibles, permettant aux utilisateurs de sélectionner facilement

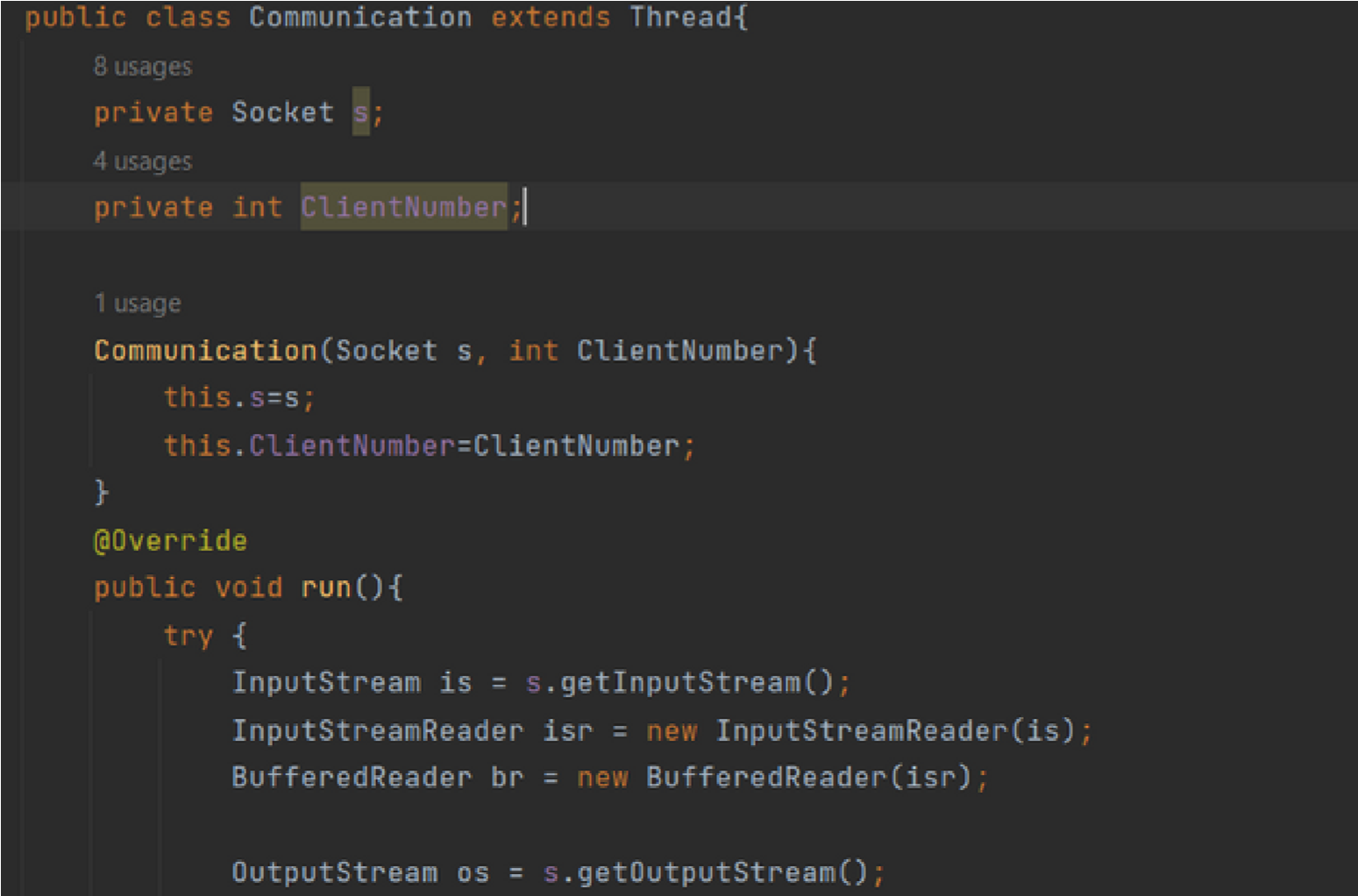
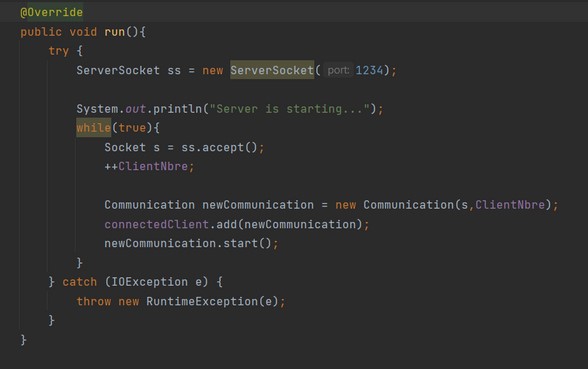
avec qui ils souhaitent discuter. Les interfaces graphiques facilitent également

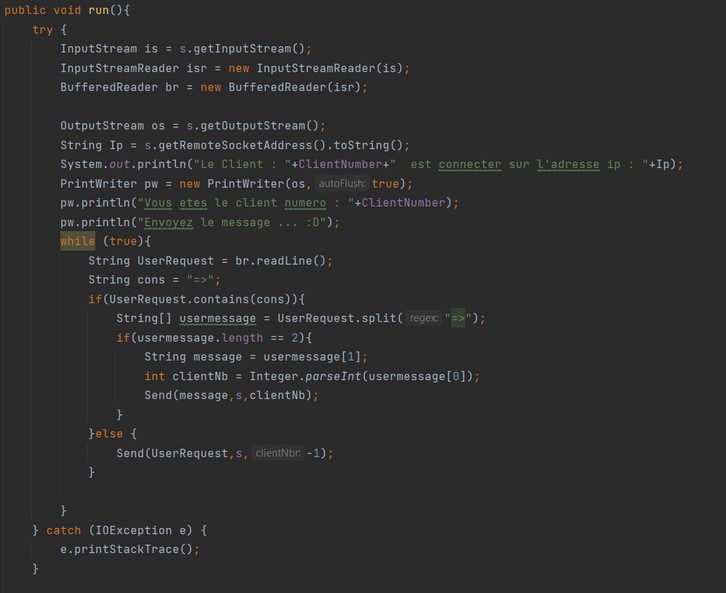
la visualisation des conversations en affichant les messages échangés dans une

disposition claire et lisible. Elles offrent des fonctionnalités telles que l'envoi de

messages, l'ajout d'émoticônes, le partage de fichiers et même la possibilité de créer des groupes de discussion. Les interfaces graphiques permettent aux utilisateurs de personnaliser leur expérience en choisissant des thèmes, des

couleurs et des paramètres de notification

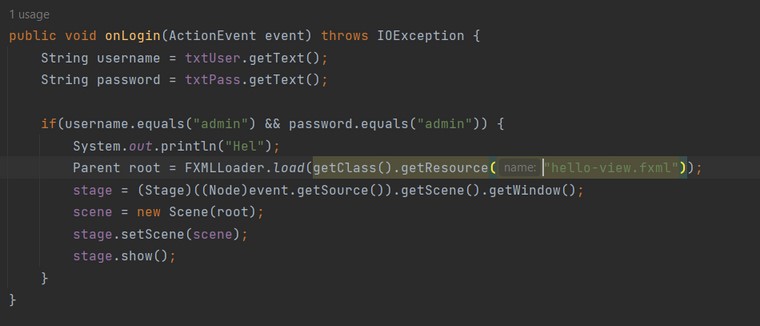






Ce code Java représente un serveur de chat local. Il écoute les connexions entrantes sur le port 1234 et permet aux clients de communiquer entre eux.

Chaque client est identifié par un numéro et peut envoyer des messages aux autres clients connectés. Le serveur gère la communication entre les clients en relayant les messages reçus à tous les autres clients. Il affiche également l'adresse IP des clients connectés.



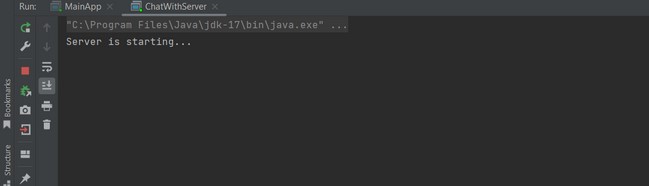
cette méthode permet de vérifier les identifiants de connexion et de passer à une nouvelle vue (hello-view.fxml) si les identifiants sont valides.

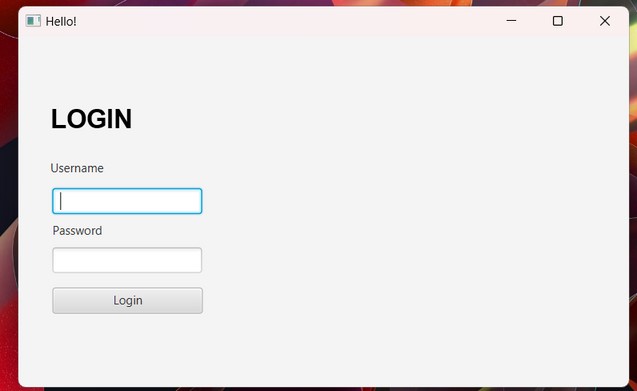


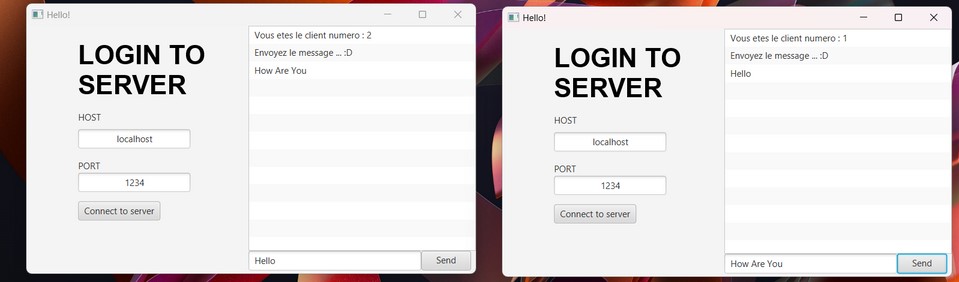
Cette méthode établit une connexion avec un serveur en utilisant le nom d'hôte et le port spécifiés, récupère les flux d'entrée et de sortie pour communiquer avec le serveur, et lit les réponses du serveur en continu en les ajoutant à une liste dans l'interface graphique.

Cette méthode est appelée lorsqu'un événement de soumission est déclenché, probablement en cliquant sur un bouton. Elle récupère le texte saisi dans un champ de texte et l'envoie au serveur en utilisant un objet PrintWriter.

**Execution**







**Conclusion**

En conclusion, ce projet de développement d'une application de chat avec Java a été une étape

importante dans notre parcours d'apprentissage.

Nous sommes reconnaissants des connaissances acquises, des compétences développées et des

défis relevés tout au long de ce processus. Nous

## espérons que cette application de chat pourra être utilisée et appréciée par de nombreux utilisateurs,

et nous sommes impatients de continuer à explorer et à développer nos compétences en développement logiciel à l'avenir.