

# Chat App with JAVA FX



**SEBBAR ABDELLAH**

**E-mail : [SEBBAR\\_ABDELLAH@emsi-edu.ma](mailto:SEBBAR_ABDELLAH@emsi-edu.ma)**

# TABLE DES MATIERES

Introduction .....	2
Présentation des fonctionnalités .....	3
Conception de l'interface utilisateur .....	4
Implémentation technique .....	5

# INTRODUCTION

L'objectif de ce rapport est de présenter le développement d'une application de chat en utilisant JavaFX et les sockets. Cette application permet à plusieurs clients de se connecter à un serveur centralisé et de communiquer entre eux via des messages texte. Je vais aborder les différentes fonctionnalités de l'application, la conception de l'interface utilisateur et les aspects techniques de mise en œuvre.

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println(" In code we trust ");  
    }  
}
```

Cette application de chat présente les fonctionnalités suivantes :

Connexion au serveur :

L'application dispose d'une interface de connexion où les utilisateurs peuvent entrer leur nom d'utilisateur et leur mot de passe. Lorsqu'un utilisateur se connecte, il est enregistré par le serveur et peut commencer à envoyer et recevoir des messages.

Interface de chat :

Une fois connecté, l'utilisateur accède à une interface de chat conviviale. Cette interface affiche les messages échangés entre tous les utilisateurs connectés au serveur. L'utilisateur peut sélectionner le host et le port puis il peut envoyer des messages en les saisissant dans une zone de texte et en appuyant sur le bouton "Send". Les messages sont affichés dans l'interface du chat avec le nom de l'utilisateur qui les a envoyés.

Communication en temps réel :

L'application utilise des sockets pour établir des connexions réseau entre le serveur et les clients. Cette approche permet une communication en temps réel, où les messages sont envoyés instantanément à tous les clients connectés. Les messages reçus sont affichés automatiquement dans l'interface du chat sans nécessiter de rafraîchissement manuel.

## II. Conception de l'interface utilisateur :

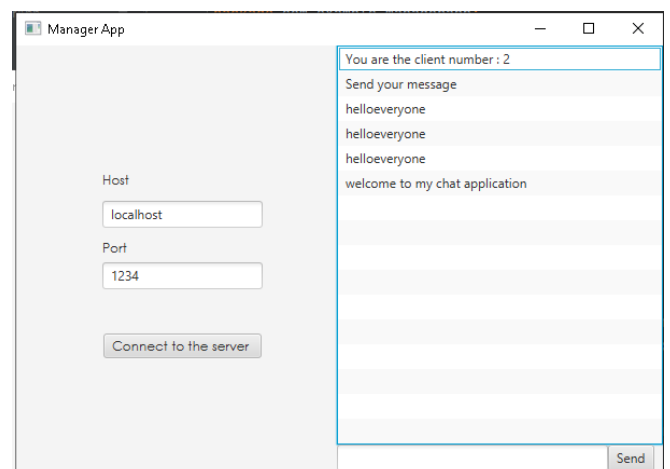
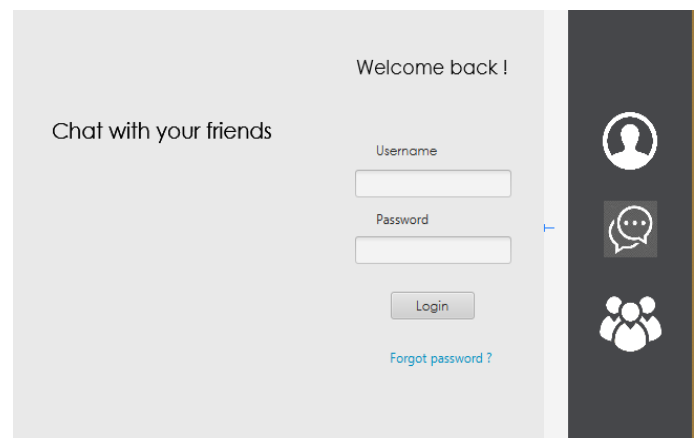
L'interface utilisateur de cette application de chat est conçue en utilisant JavaFX, qui offre une grande flexibilité pour la création d'interfaces graphiques interactives. Elle est composée de deux fenêtres distinctes :

Interface de connexion :

L'interface de connexion comprend des champs de texte pour entrer le nom d'utilisateur et le mot de passe, ainsi qu'un bouton "Login" pour établir la connexion avec le serveur. Cette interface est simple et facile à utiliser, permettant aux utilisateurs de se connecter rapidement.

Interface du chat :

L'interface du chat est divisée en deux sections principales : une zone de chat où les messages sont affichés et une zone de saisie où les utilisateurs peuvent écrire leurs messages. Les messages sont affichés avec le nom d'utilisateur correspondant, permettant une identification claire des participants à la conversation. La zone de chat est défilable pour faciliter la lecture des messages précédents.



### III. Implémentation technique :

L'application de chat est implémentée en utilisant le langage de programmation Java et la bibliothèque JavaFX pour la création de l'interface utilisateur. Voici les principales étapes d'implémentation :

Configuration du serveur :

Le serveur est configuré pour écouter les connexions entrantes des clients sur un port spécifié. Il utilise des threads pour gérer simultanément plusieurs clients connectés. Chaque client est enregistré avec un nom d'utilisateur unique et un socket de communication.

Connexion du client :

Lorsqu'un client souhaite se connecter, il fournit un nom d'utilisateur et un mot de passe. Le client établit un socket de communication avec le serveur en utilisant l'adresse IP et le port spécifiés. Une fois connecté, le client envoie son nom d'utilisateur au serveur pour l'enregistrement.

Communication client-serveur :

Le client et le serveur communiquent en utilisant des flux d'entrée/sortie sur les sockets. Lorsqu'un client envoie un message, il est transmis au serveur qui le diffuse à tous les autres clients connectés. Le serveur reçoit également les messages des autres clients et les transmet au client approprié.

