

Esiea Paris 2020.

Architecture Système.

Projet Chat-Room

Taleb Amine.

20/04/2020.

Versions du document	Date	Rédacteur	Modifications Réalisées
1.0	20/05/20	Taleb	Création

I. Introduction.

I.1 Objet :

Ce document décrit l'architecture système du projet.

- Choix de l'architecture (MVC, MVP ...).
- Diagramme des modules.
- Diagrammes détaillés des modules
- Diagramme UML.
- Conception suivant les principes SOLID.
- Exécution & Principe d'utilisation

I.2 Glossaire :

I.3 Choix de l'architecture :

Nous avons opté pour une architecture de type MVC (modèle vue contrôleur), qui se prête bien pour ce type de projet, où l'on distingue clairement la séparation de l'IHM correspondant à la vue qu'auront les clients et du contrôleur/modèle qui auront les tâches spécifiques liés aux traitements des données échangés et la gestion des connections entre les client-serveur.

Ce modèle de conception permettra de bien séparé les blocs de développement afin d'avoir une meilleure souplesse, évolutivité et maintenance du projet.

Pour nous, l'approche MVC pour ce projet apporte de réels avantages, avec une conception claire car les données sont clairement séparées de la vue et du contrôleur.

Aucun traitement ne sont réalisés dans la vue (excepté celles liées aux graphiques).

La partie modèle représente plusieurs types de données et règles :

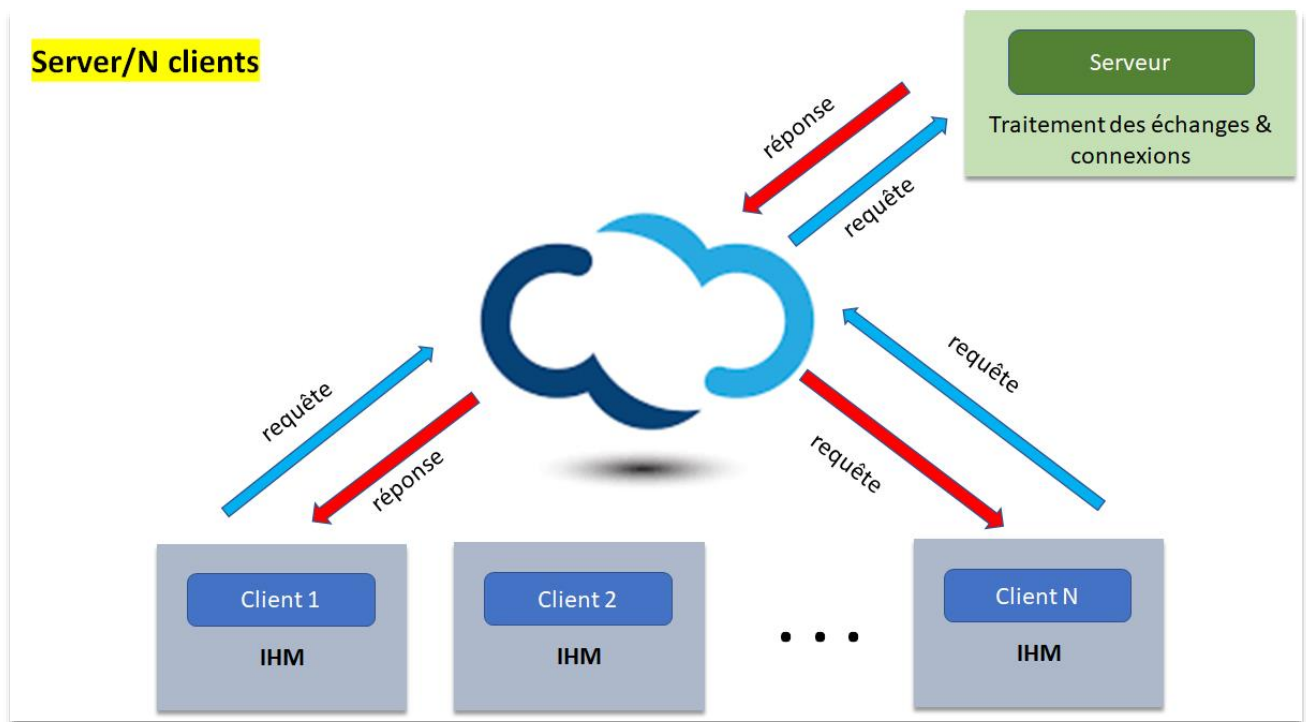
- Les caractéristiques de l'utilisateur, login, password ...
- Les données correspondant aux messages échangés.
- Le stockage de données.
- Les règles d'échanges entre tous les clients ou mode privé.

Le contrôleur interprétera les demandes des clients et appellera les parties du modèle et de la vue pour répondre à ces demandes.

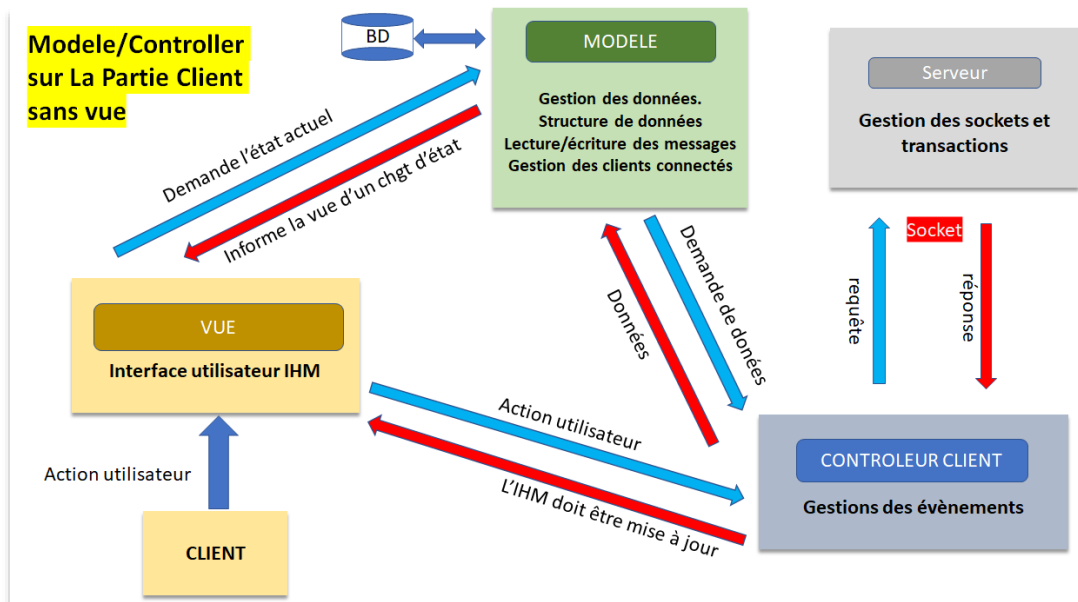
Ainsi, lorsqu'un client envoie un message, le contrôleur n'effectue aucun traitement, il intercepte les requêtes et détermine quelles vues et quels modèles aura la charge du traitement.

1.4 Diagramme des modules :

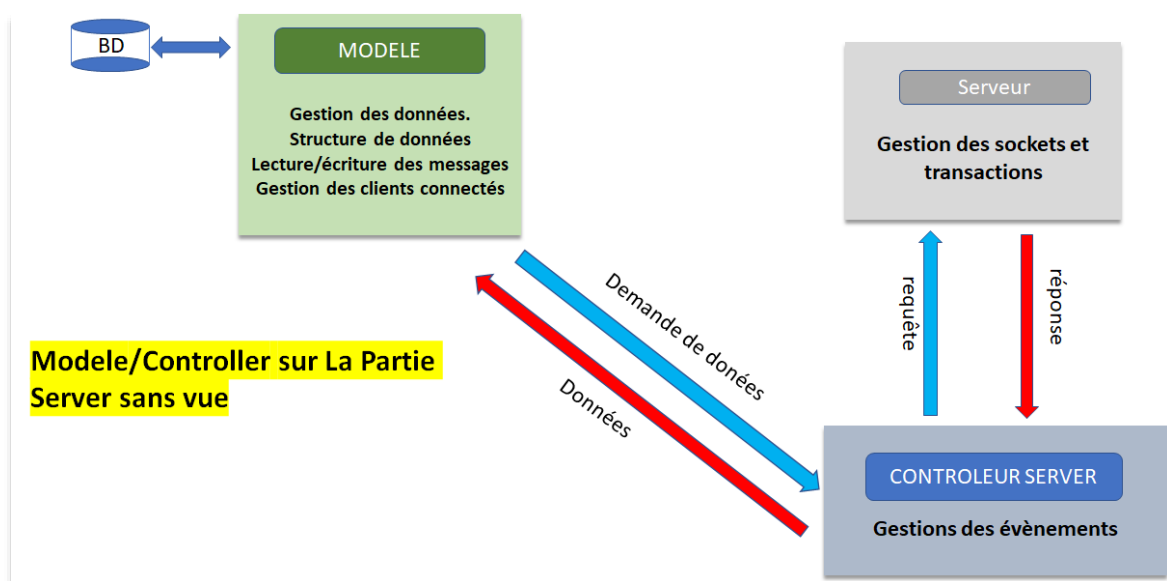
Le diagramme ci-dessous présente les modules haut niveau du projet avec N clients



Le diagramme haut niveau ci-dessous présente Les divers modules de la partie client en mode MVC.



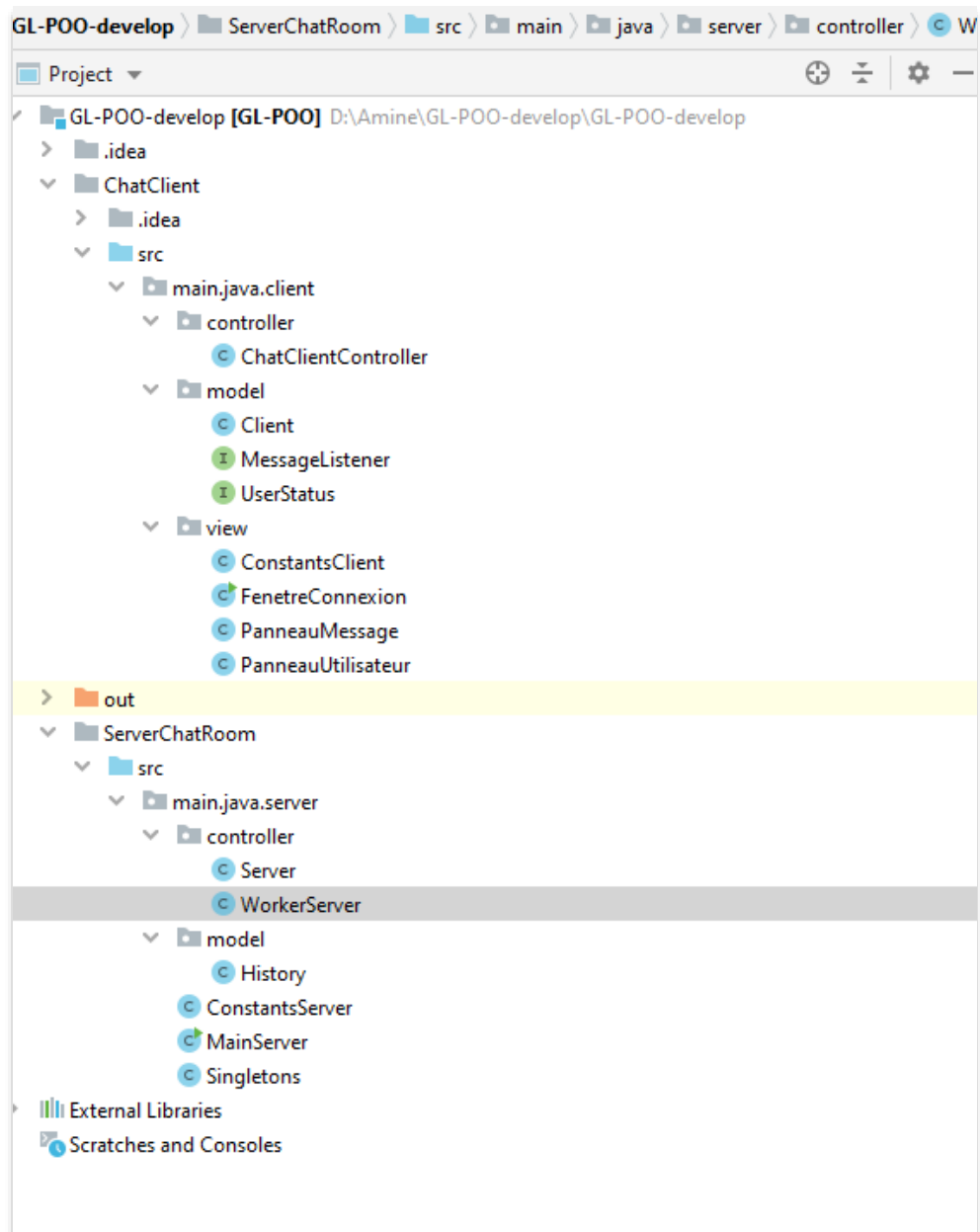
Le diagramme haut niveau ci-dessous présente Les divers modules de la partie serveur en mode MVC.



I.5 Diagramme UML :

Les diagrammes UML des classes sont dans le fichier associé UML.pdf qui présente les divers classes et méthodes du projet.

I.6 Au niveau source code :



On retrouve les blocs et les classes liés à l'architecture MVC.

I.6 Exécution et Principe d'utilisation :

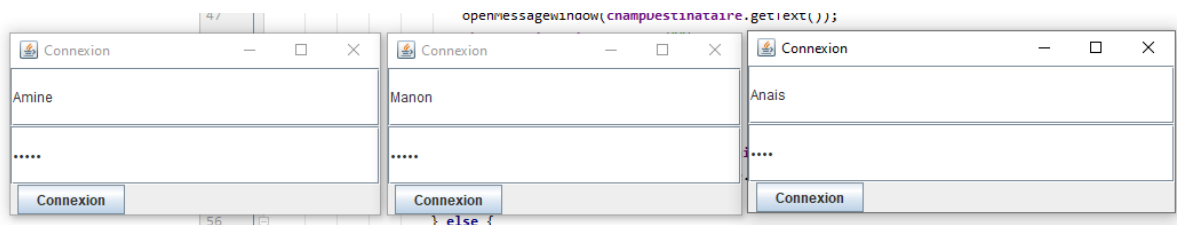
Lorsque le serveur est lancé sur la machine Host en réseau local.

Une trace de vérification apparaît indiquant que le serveur est prêt

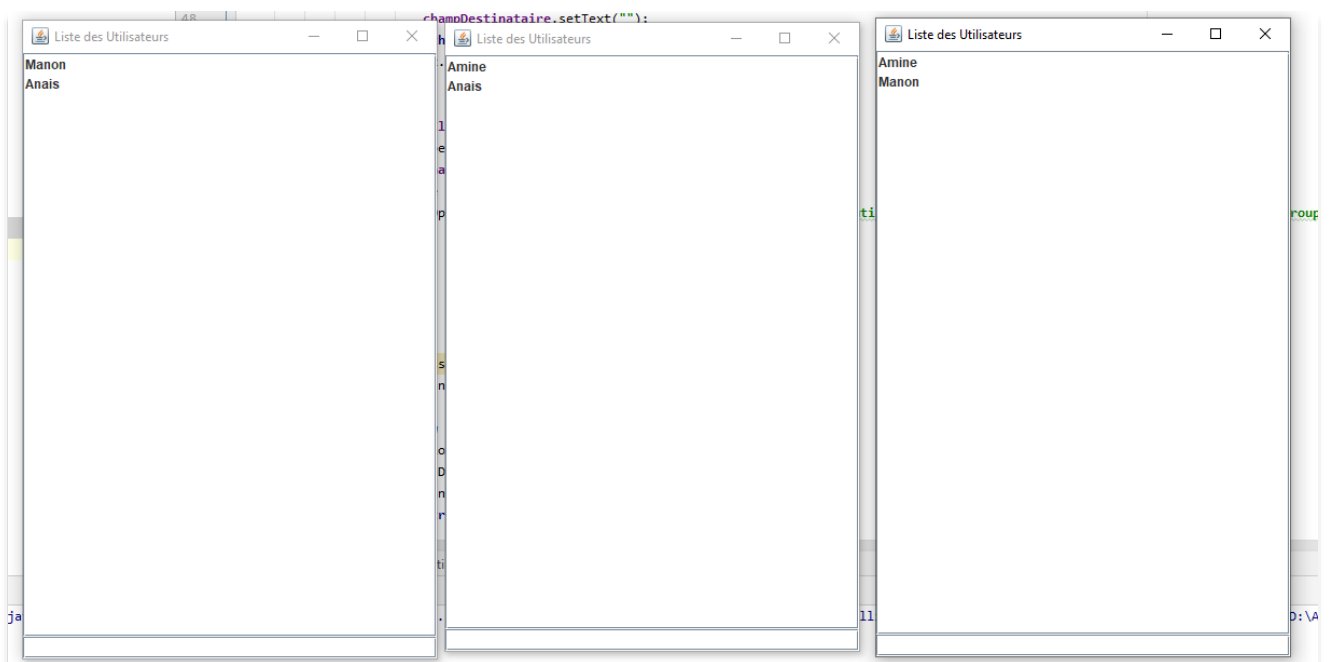
```
"C:\Program Files\Java\jdk-11.0.6\bin\java.exe" "-javaagent:D:\Program Files\JetBrains\IntelliJ"
About to accept client connection...
Accepted connection from Socket[addr=/127.0.0.1,port=30570,localport=8818]
```

Voici les étapes pour mener un ou plusieurs ChatRoom en parallèle.

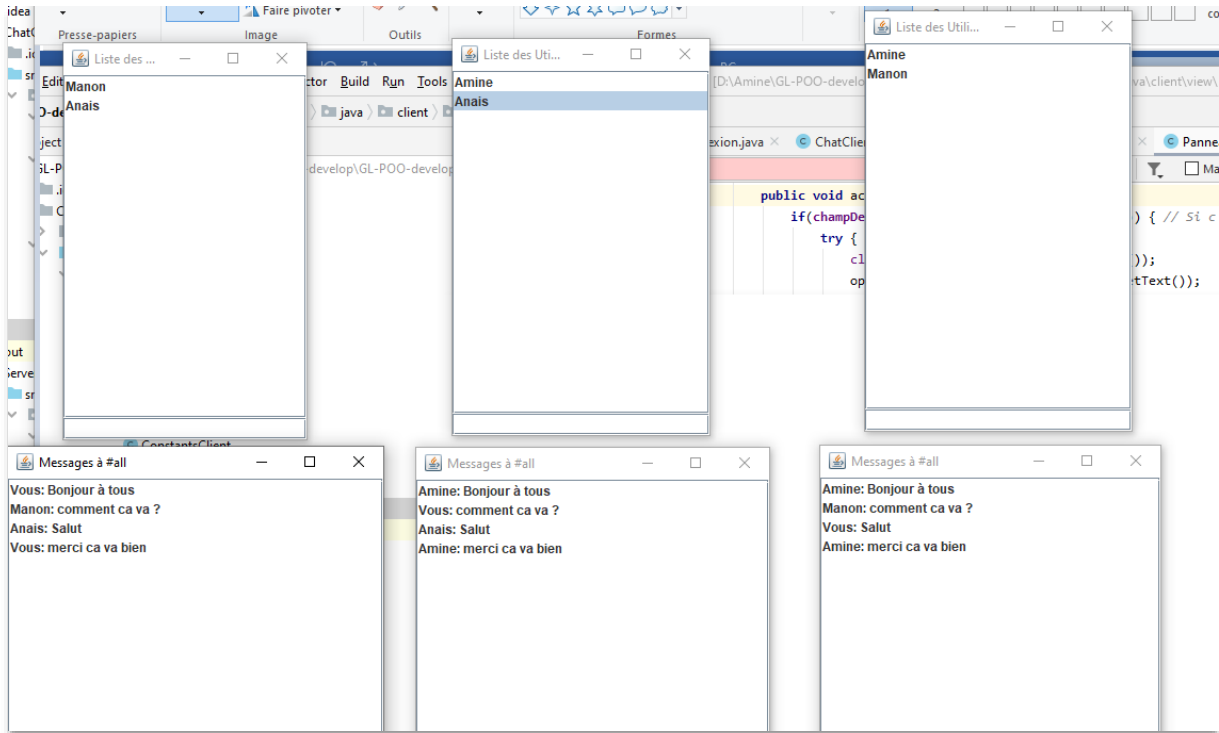
Par exemple Initialisation de 3 clients :



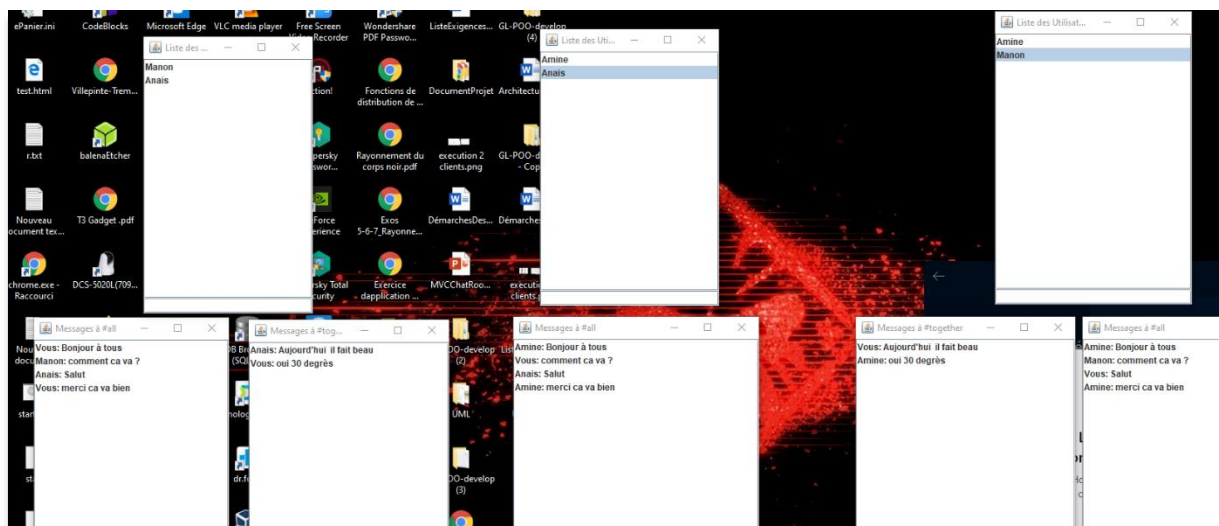
La validation des connexions va créer une fenêtre pour chaque client qui leur permettra de savoir qui est connecté au ChatRoom



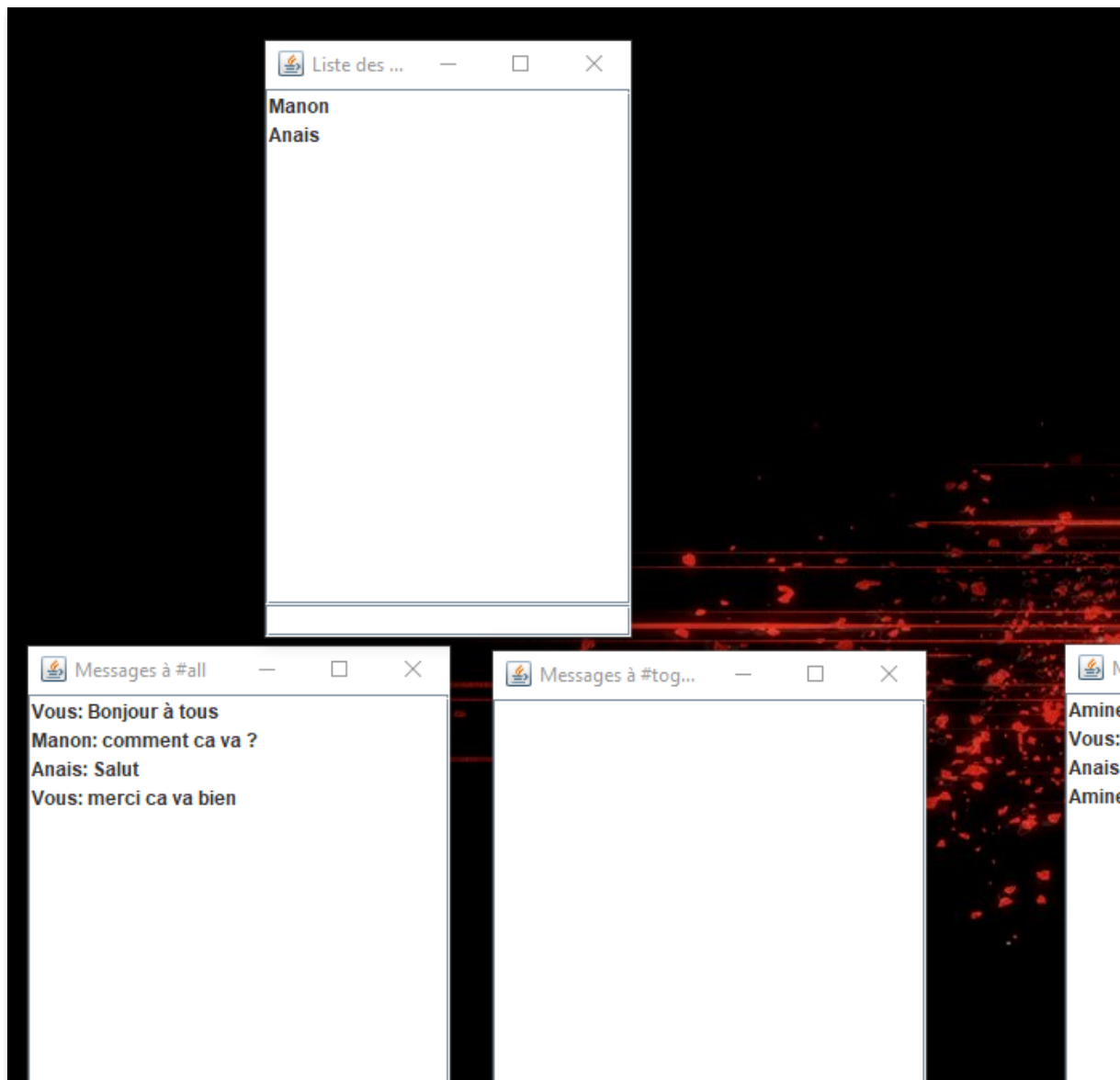
Création d'un groupe de discussion entre les 3 personnes et un autre entre 2 personnes. Le premier s'appelle **all**, pour qu'un client puisse y accéder il faut qu'il tape **#all**, il y a aura une fenêtre ayant pour titre « Messages à **#all** » par client pour ce Chat room.



Avec maintenant un second un second Chatroom entre 2 personnes
Second groupe appelé **#together**



La suppression des messages d'un client sur un chat particulier s'effectue avec l'appel #delete



Lorsqu'un client se déconnecte en quittant l'application on obtient la mise à jour des clients connectés.

