

Esiea Paris 2020.

Diagrammes UML
&
Diagrammes de séquences
Projet Chat-Room

Taleb Amine.

10/06/2020.

Versions du document	Date	Rédacteur	Modifications Réalisées
1.0	15/06/20	Taleb	Création
1.1	18/06/20	Taleb	Update classe ChatClient
1.2	20/06/20	Taleb	Update Modèle MVC

I. Introduction.

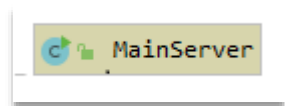
I.1 Objet :

Ce document décrit les divers diagrammes UML du projet.

I.2 Diagramme des classes high-level côté serveur.

Classe MainServer:

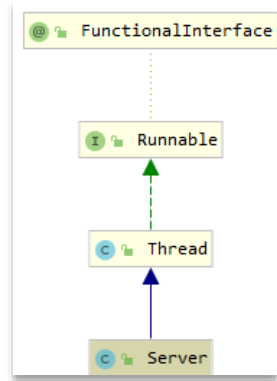
Classe d'entrée contenant la méthode main pour le lancement du serveur



Classe Server:

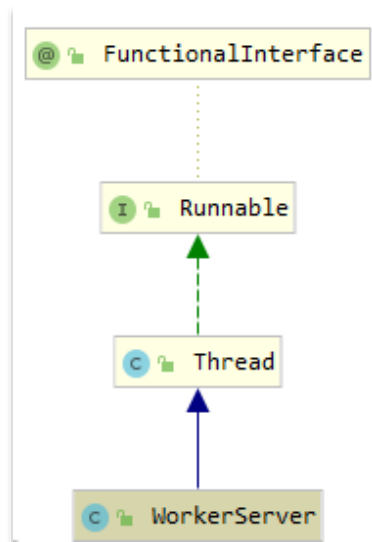
Cette classe qui étend la classe thread, permet d'être à l'écoute des sockets du réseau local

Lorsqu'un socket arrive en provenance d'un client, un thread de type WorkerThread sera créé afin de traiter la demande.




Classe WorkerServer:

Classe qui étend la classe `java.lang.Thread` est la classe centrale du `Server` permettant de gérer les diverses actions en fonction des requêtes issus des clients.

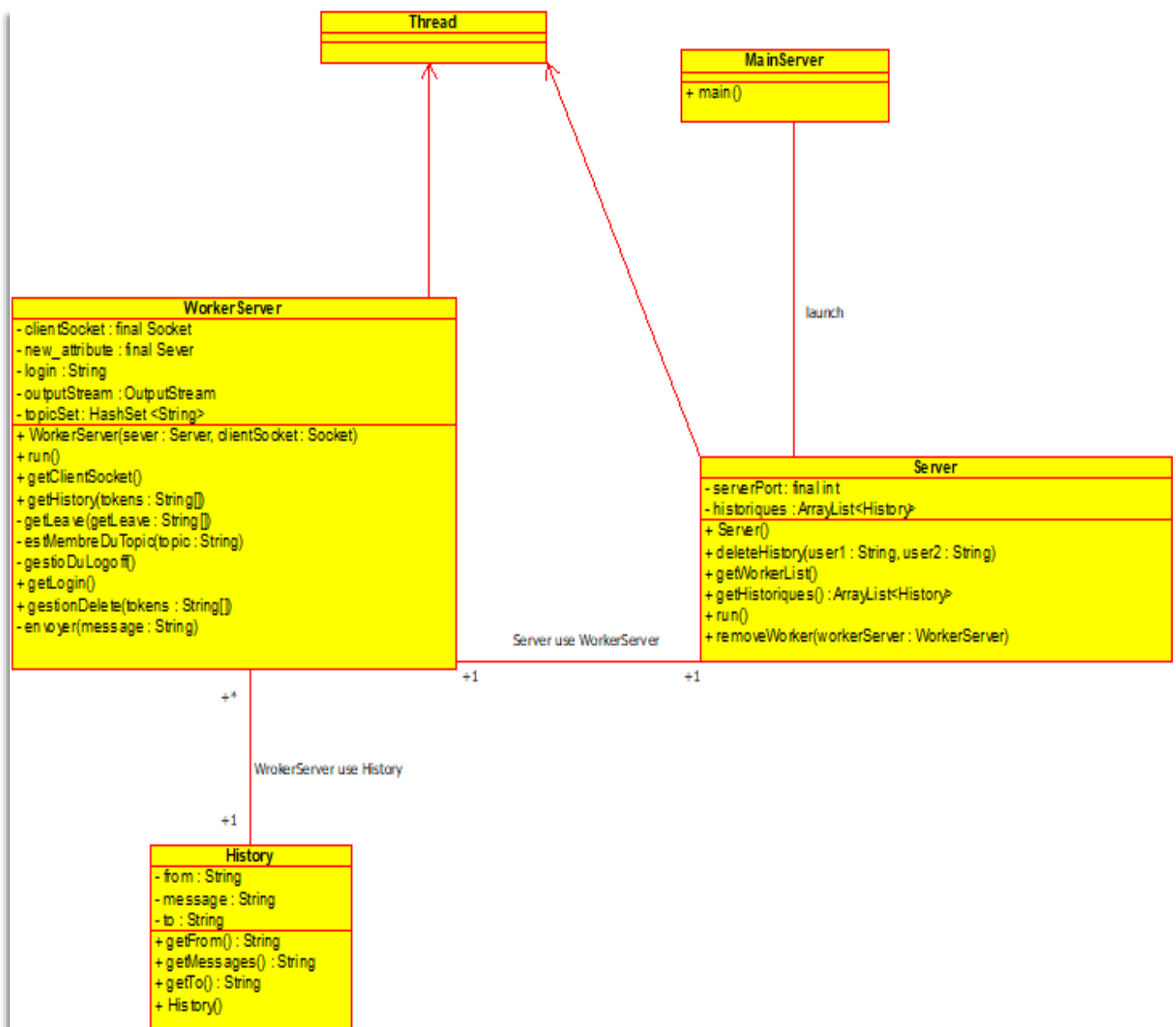


Classe History :

Classe permettant de stocker et de récupérer les messages afin de réaliser l'historique des échanges d'un client.

 History

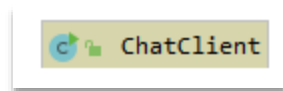
1.3 Diagramme UML détaillé côté server.



I.4 Diagramme des classes high-level côté client.

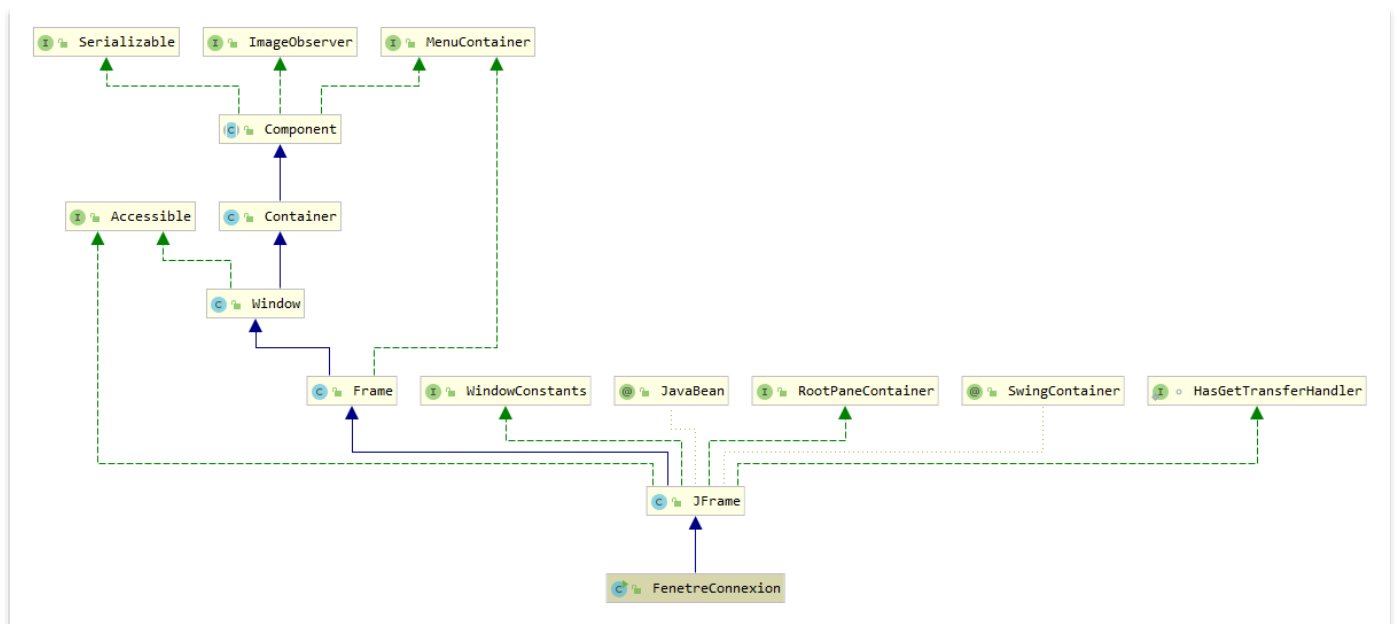
Classe ChatClient :

Classe d'entrée contenant la méthode main pour le lancement du client.



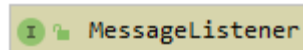
Classe FenetreConnexion :

Classe graphique permettant la gestion du login password de connexion client.



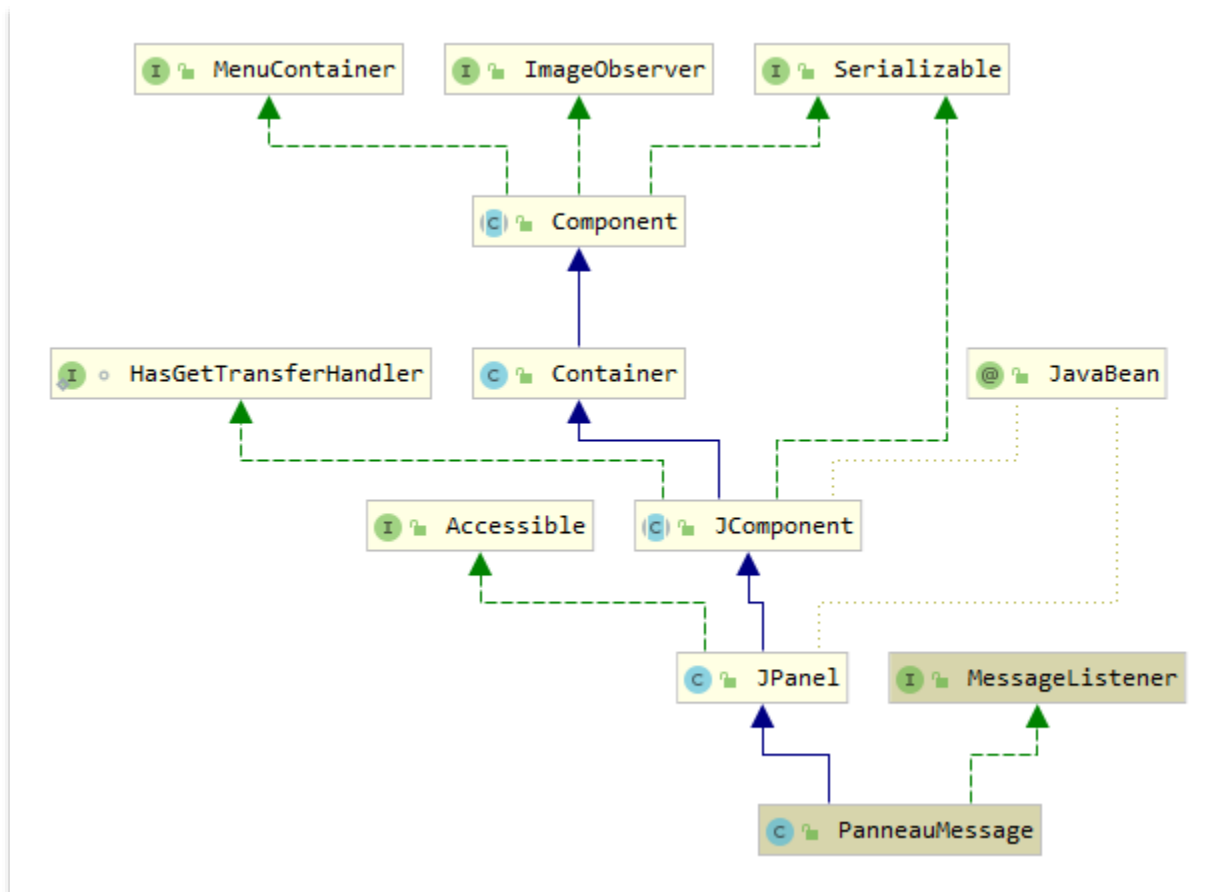
Classe *MessageListener*:

Classe d'interface *MessageListener* permettant de créer un écouteur sur la réception de messages.



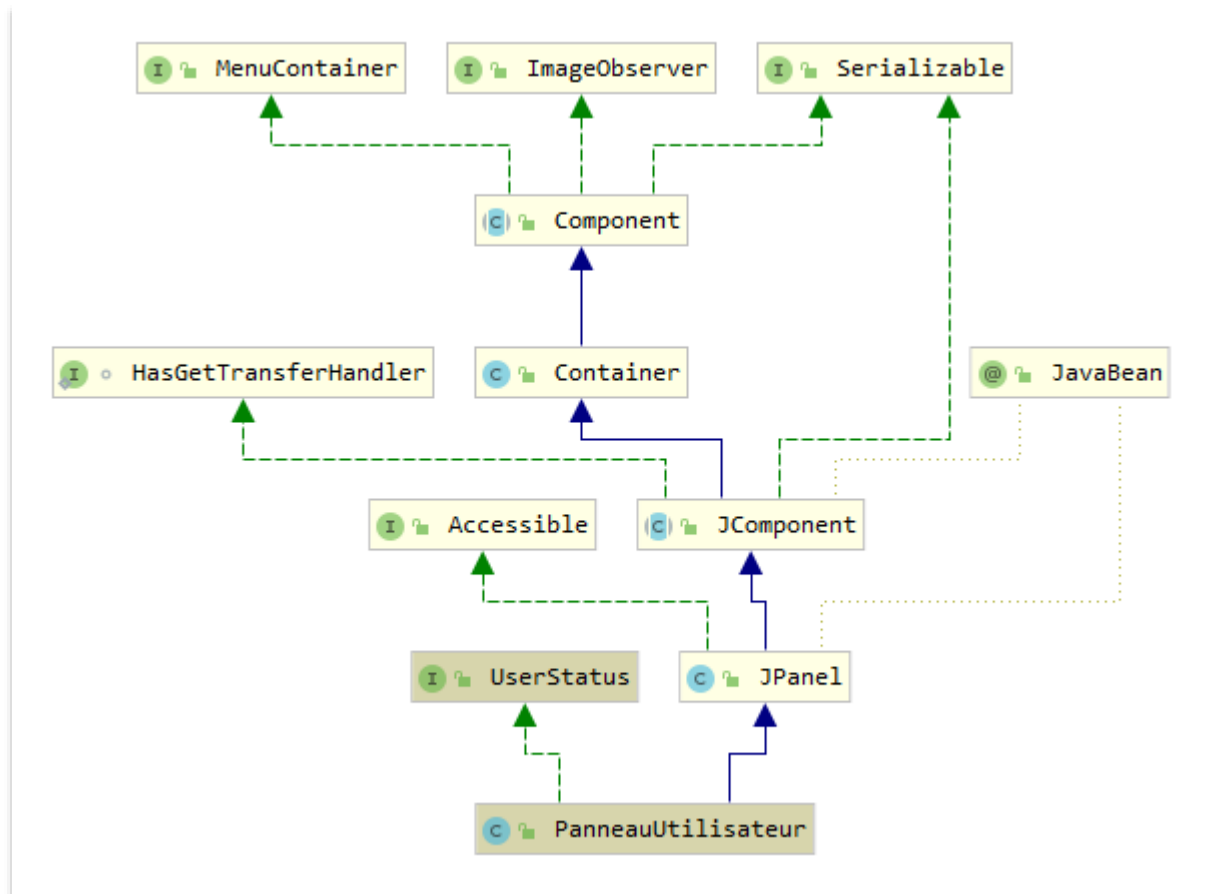
Classe *PanneauMessage*:

Classe permettant l'affichage des messages clients.



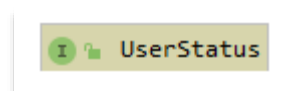
Classe PanneauUtilisateur:

Classe permettant l'affichage des messages clients.

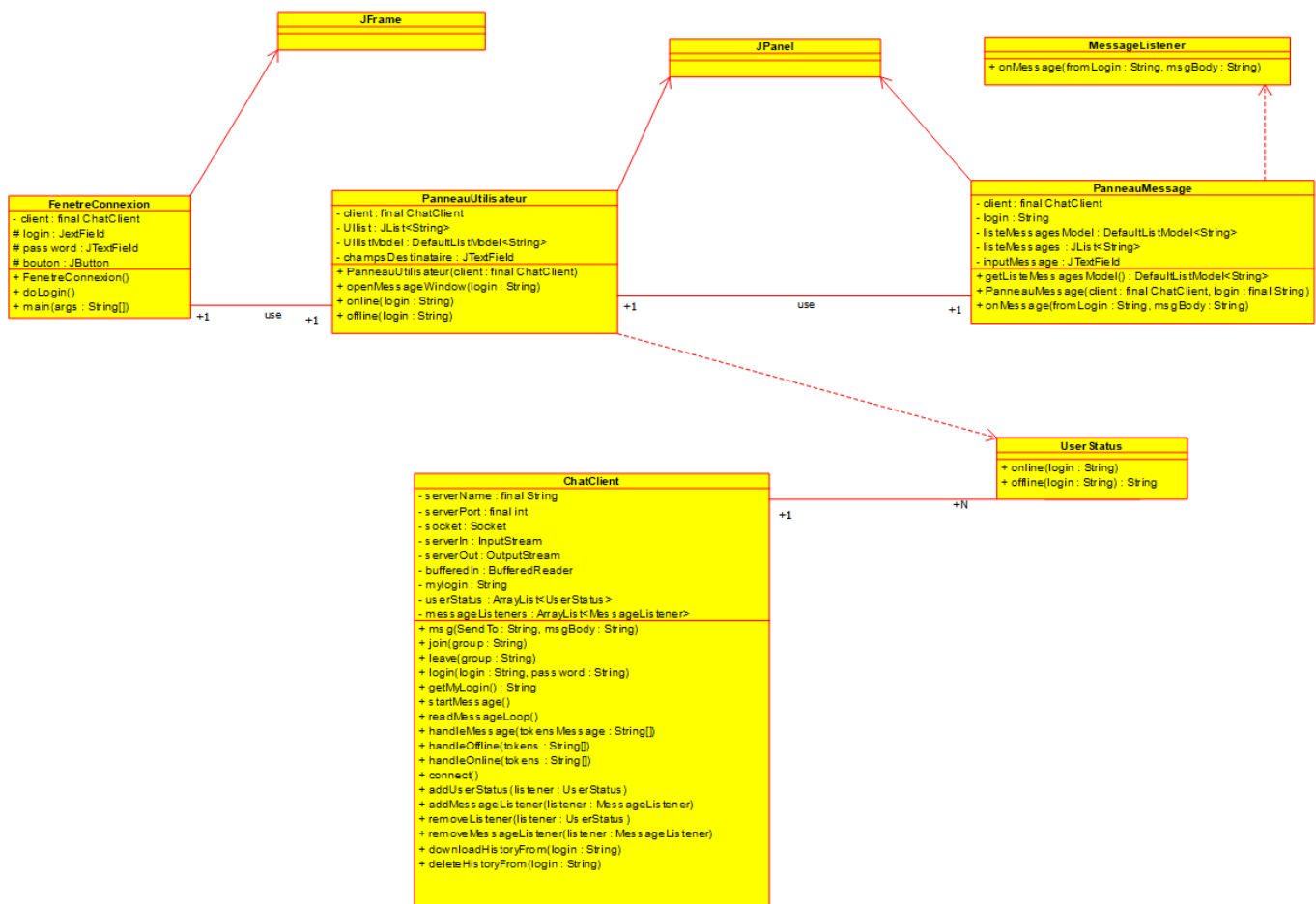


Classe UserStatus:

Classe d'interface permettant de savoir si un client est connecté ou non.



1.5 Diagramme UML détaillé côté server.



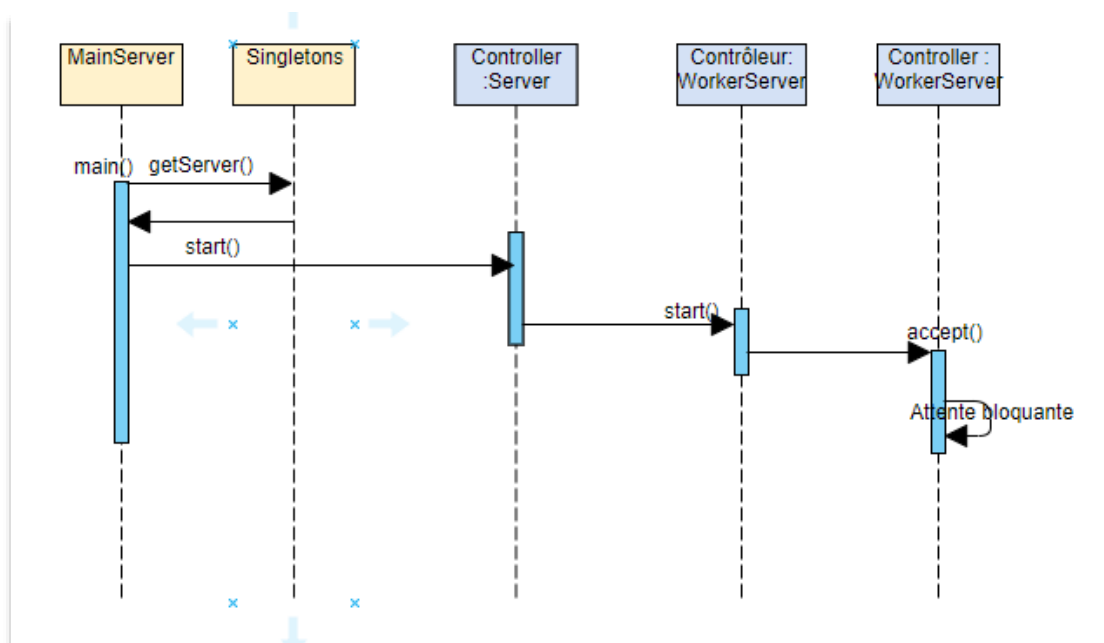
II Diagramme de séquences

Cette partie indique les principaux diagrammes de séquences du projet

II.1 Initialisation du Server

Une unique instance du serveur est créée grâce à l'utilisation d'un singleton.

L'initialisation du Serveur s'arrête jusqu'à la création d'un WorkerServer qui sera en attente une attente bloquante de socket (méthode accept). Il y a un WorkerServer par groupe de discussion



II.2 Lancement du client

Le lancement du client a lieu par initialisation de l'objet FenetreConnection qui permet via une interface graphique de saisir le login password du client. Lorsque la connexion est effective, il y a création du panneau utilisateur correspondant à la liste des clients connectés.

