Maxym Charpentier
Jonathan Jory

2A2B

Rapport
SAE: R3.03

Messagerie Instantanée

2022-2023

I/ Fonctionnalité implémentées:

a) Sur Terminal

- Le client se connecte au serveur en précisant l'adresse IP ou le nom, ainsi que son nom d'utilisateur. (Si son nom d'utilisateur est déjà pris un nouveau nom lui est demandé)

- Il voit ensuite la liste des différents salons de discussions et peut rejoindre en faisant /join nomSalon

- Les personnes dans le même salon peuvent envoyer et recevoir des messages contenant l'heure, le nom de l'expéditeur et le message.

- L'utilisateur peut se déconnecter d'un salon en envoyant /quit au serveur. Il peut ensuite se reconnecter à un autre salon, ou se déconnecter en saisissant /quit.

- /nbuser: renvoie le nombre d'utilisateurs connectés au salon.

- /uptime: Connaitre depuis combien de temps le serveur est lancé.

- /users :Avoir la liste des personnes connectés.

- @userName :Envoyé un message privé à userName.

- /help: affiche la liste des commandes disponnible.

b) Sur Interface Graphique

- Le client se connecte en localhost au serveur en précisant son nom d'utilisateur. (Si son nom d'utilisateur est déjà pris un nouveau nom lui est demandé)

- Il voit ensuite la liste des différents salons de discussions et peut rejoindre en faisant /join nomSalon ou en cliquant sur l'un des salons disponibles.

- Les personnes dans le même salon peuvent envoyer et recevoir des messages contenant l'heure, le nom de l'expéditeur et le message.

- L'utilisateur peut se déconnecter du serveur en envoyant /quit au serveur. Il peut aussi quitter directement en cliquant sur le bouton quitter.

- Pour changer de salon, il lui suffit de cliquer sur le nouveau salon ou il veut se rendre

- Lorsque l'utilisateur entre dans un salon il voit les messages envoyés dans le salon depuis son lancement.

- /nbuser: renvoie le nombre d'utilisateurs connectés au salon.

- /uptime: Connaître depuis combien de temps le salon actuelle est lancé.

- /users :Avoir la liste des personnes connectés.

- @userName :Envoyé un message privé à userName.

- /help: affiche la liste des commandes disponible.

II/ Différents choix effectuées:

a) Sur Terminal

Au niveau des choix techniques , pour la partie terminal on a choisit d'avoir un dictionnaire avec en clef le nom du client et en valeur la session ce qui nous permet d'avoir facilement accès et très rapidement à la liste de tout nos clients. Pour par exemple lorsqu'on demande le nom d'un client vérifier qu'il n'existe pas déjà ou encore lorsqu'on envoie un message privé. Avoir accès à la session permet avec juste le nom d'un client de pouvoir lui envoyé un message. De plus nous avons un autre dictionnaire avec en clef le nom du salon et en valeurs la liste des membres de ce salon. Ce qui fait qu'on a facilement accès a la liste des salons disponible et aussi rapidement accès a la liste des membres dans un salon.

a) Sur ihm

Au niveau des choix techniques , pour la partie terminal IHM. On est tout d'abord partie sur les même choix que précédemment sauf que nous avons ajouter la possibilité de sauvegarder les messages dans un salon. C'est pourquoi pour éviter les répétitions nous avons crée une classe Salon contenant la liste des membres, des messages, et d'autres informations comme le nom et le temps depuis la création.
De plus pour les messages nous avons décidé de rajouter une entête contenant certaine informations pour que quand le client reçoit le message il sache si le message contient un message simple à afficher ou la liste des salons ou encore le message si il peut quitter le serveur etc.

III/ Les Apports de cette SAE:

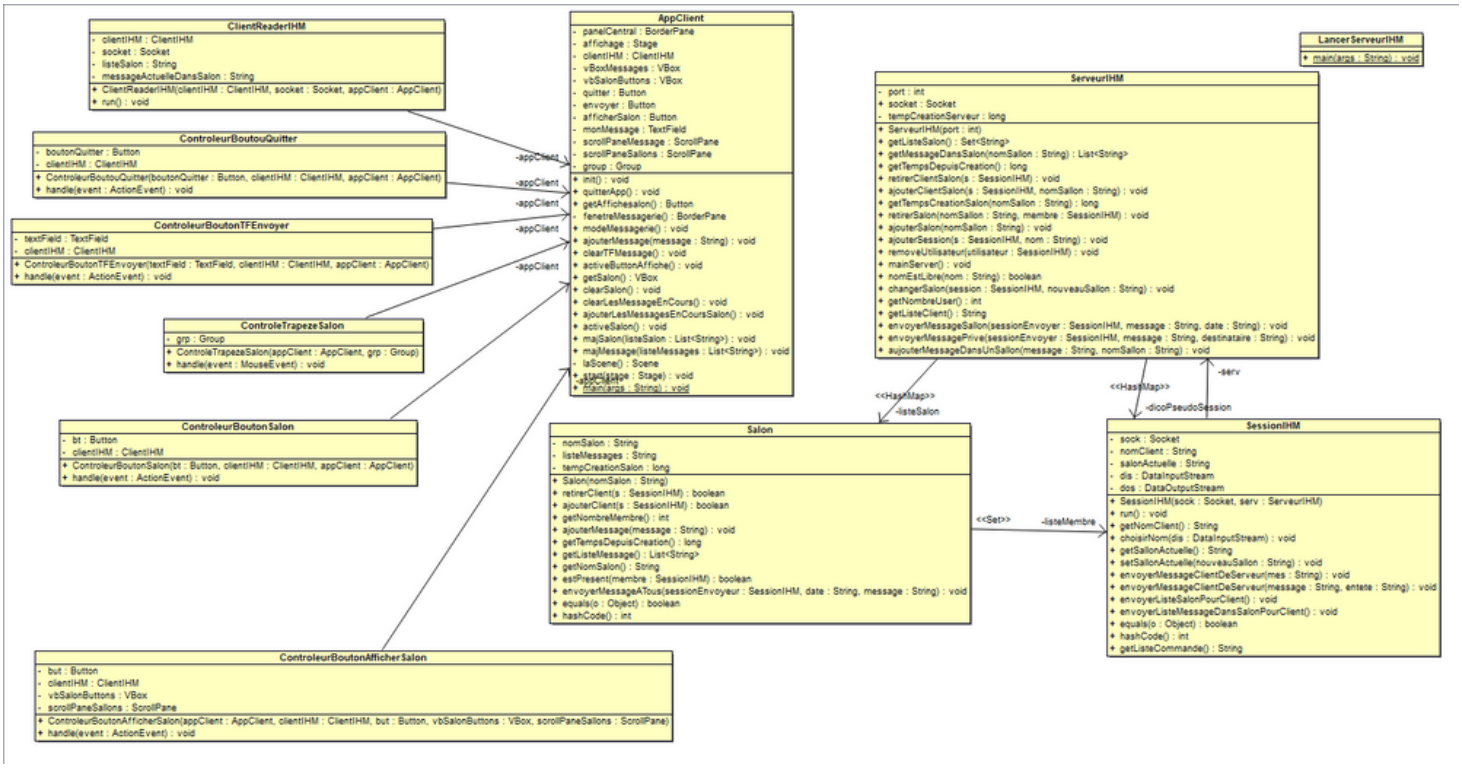
- Cela nous à permis de mettre en pratique les concepts de base de la programmation client-serveur en utilisant les notions vues en TD et TP et comment ces deux parties peuvent fonctionner ensemble pour réaliser une application complète tout en utilisant des protocoles de communication réseau pour que le client puisse se connecter au serveur et échanger des messages.

- Le serveur devait être capable de gérer les connexions utilisateur et les flux de messages en temps réel. Cela signifie qu'il devait être capable de traiter les messages envoyés par les utilisateurs tout en acceptant de nouvelles connexions et en gérant les utilisateurs déconnectés. Nous avons donc utilisé de la programmation en temps réel ainsi que de la gestion des threads. De plus avec le fait que le serveur doit être capable de traiter plusieurs messages simultanément, il est donc nécessaire d'utiliser la programmation parallèle .

- Pour finir, le fait de concevoir une interface graphique nous a permis de revoir l'utilisation de JAVAFX que nous n'avons pas utilisé depuis la dernière semaine du S2.

IV/ Diagramme de Classe:

IHM



Terminal