

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 585 – Télématicque

Devoir 3

Rapport de groupe

Travail présenté à
Bessam Abdulrazak

Par

Soumar Alrubaie, 15 161 815
Claudia Bergeron, 15 173 205
Naomi Lépine, 15 034 669
Arnaud Ricaud, 17 132 853
Francis Vachon, 17 022 240
Aziz Zouaoui, 14 185 008

Équipe E

Table des matières

Introduction	3
Description du contenu du zip	3
src	3
exécutables	3
La procédure d'exécution des programmes	3
Serveur	4
Client	4
Description sommaire du système solution	4
Document de conception (après la mise à jour)	5
Diagrammes de classe	5
Diagramme de classe « Serveur »	6
Diagramme de classe « Client »	7

Introduction

Après avoir fait l'analyse de notre système lors du Tp2, nous avons commencé la conception et la programmation. Le choix du langage fut décidé par la majorité des membres de l'équipe qui était le langage «Java».

Ce tp a apporté certaines modifications à l'architecture pensée lors de la conception initiale (Tp2).

Description du contenu du zip

Le zip soumis pour le Tp3 contient trois dossiers principaux:

- **src**

Ce dossier contient alors les "Package" suivants:

- *Client*: contient toutes les données concernant un client ainsi que les interfaces utilisées dans notre système.
- *Serveur*: contient l'architecture TCP et UDP ainsi que l'authentification client.
- *Structures*: contient, tout simplement, les définitions des classes utilisées par le serveur et le client.

Ces "Packages" contiennent le code source de notre application non compilée.

- **exécutables**

Ce dossier contient des fichiers à exécuter afin de profiter d'une expérience de messagerie instantanée

Un dernier fichier appelé "*noms.txt*" vient compléter le zip, ce dernier servant à fournir le nom des participants.

Le client (package Client) et le serveur (package Serveur) sont gérés comme une seul et même programme. Ils partagent les classes communes sous "Structures" (Message, Utilisateur, Salle, etc...).

La procédure d'exécution des programmes

Avant l'extraction du zip, il faudra s'assurer d'une installation préalable de Java sur la machine d'exécution, puis naviguer jusqu'au dossier "exécutable" cité plus haut, puis ouvrir une fenêtre de ligne de commande a cet emplacement précis (deux fois: pour le client et le serveur).

L'exécution se fera en deux étapes:

1. *Serveur*

Écrire la commande “java -jar serveur.jar” et ce pour démarrer un serveur.

2. *Client*

Écrire la commande “java -jar client.jar” et ce pour démarrer un client et son interface graphique.

Le but ultime de ce Tp3, étant de faire communiquer plusieurs clients en même temps, par conséquent il vous faudra ouvrir exécuter plusieurs clients et ce sur le même ordinateur que le serveur, car ce dernier communique au travers du localhost.

Les informations d'authentification se trouvent directement au niveau du serveur (dans le programme), puisque les données utilisateur sont importées lors du lancement de celui-ci. Vous aurez la possibilité de rajouter un “utilisateur” et utiliser ce dernier pour effectuer du clavardage.

Description sommaire du système solution

Afin de pouvoir utiliser l'application, un usager devra s'authentifier au préalable, l'application fera le nécessaire pour vérifier l'authenticité du *nom d'utilisateur* et le *mot de passe* de ce dernier. Si ce dernier ne dispose de compte dans notre application, il pourra éventuellement s'y inscrire et ainsi rejoindre l'application.

Cette application de messagerie utilise des interfaces faites avec la librairie graphique pour le langage de programmation Java¹ communément appelée “**Swing**”, par conséquent le client et le serveur sont écrit dans le même langage de programmation. ces derniers utilisent des sockets pour communiquer à distance.

Une fois la connexion établie entre le(s) client(s) et le serveur, les utilisateurs pourront:

- Se joindre à une salle de discussion
- Créer une nouvelle salle de discussion
- Consulter et modifier leur profil utilisateur
- S'échanger des messages
- Supprimer des messages qui leurs sont propre
- Quitter une salle de discussion
- Changer leurs propres informations
- Se déconnecter

¹ [https://fr.wikipedia.org/wiki/Swing_\(Java\)](https://fr.wikipedia.org/wiki/Swing_(Java))

Document de conception (après la mise à jour)

Diagrammes de classe

Plusieurs modifications ont été faites sur notre conception du tp2, concernant principalement les diagrammes de classe, ainsi que la création du diagramme « Client ».

Le diagramme «Serveur» :

Le protocole UDP est utilisé pour l'échange des messages à l'aide de la classe «SocketUDP». La méthode « run() » explique le déroulement des actions dans un socket UDP, lorsqu'on reçoit un paquet, la méthode «inPacket.getData()» nous permet d'extraire son contenu. on a ajouté une méthode « treatNewMessage(Message msg)» qui permet de recevoir un message dans une salle, ensuite le passer à la salle en question. Afin de renvoyer la réponse au client sur le port auquel il attend une réponse.

La classe «SocketTCP» contient les méthodes permettant de gérer les utilisateurs et les salles (Authentifier un utilisateur, créer une salle, se désabonner etc ...).

Pour les autres classes, nous avons gardé la même structure, en ajoutons des accesseurs et mutateurs. De plus, on a ajouté des méthodes qui nous facilitent le travail comme « estVide() » et «isSubscribed» dans la classe salle.

Le diagramme « Client »:

Ce diagramme a été ajouté pour décrire le comportement d'un client dans notre application. On trouve dans cette classe toutes les méthodes concernant un client (ajouter un message, inscription dans une salle, obtenir la liste des utilisateurs dans le serveur etc...).

Diagramme de classe « Serveur »

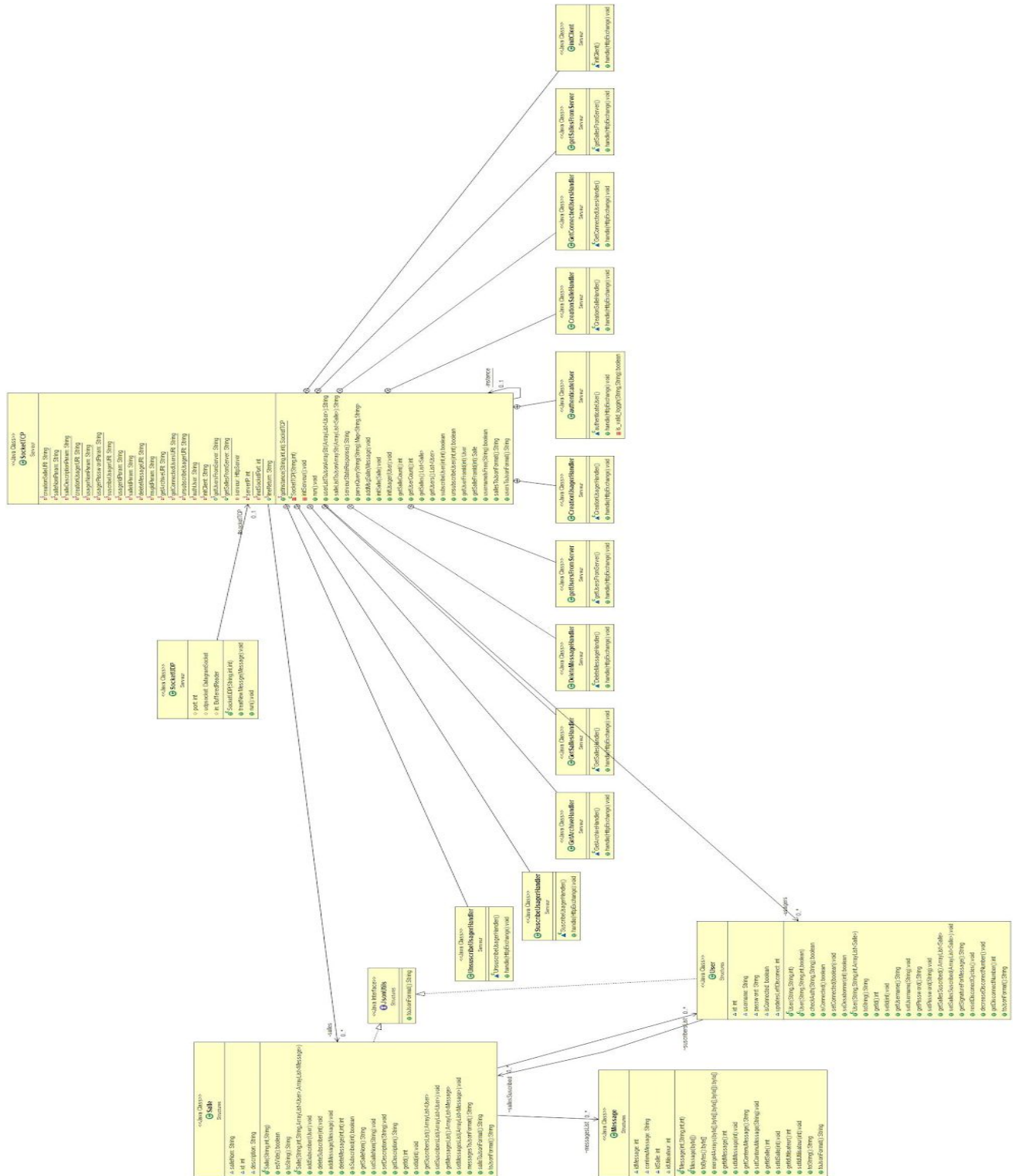


Diagramme de classe « Client »

