Rapport de projet ChatSystem

Agathe BACONNIER, Diego LOPEZ, Vatosoa RAZAFINIARY

****

INTRODUCTION :

Ce projet a pour objectif de mettre en place un système de clavardage, permettant l'échange de messages entre des agents, suivant un cahier des charges bien défini. La réalisation du projet passe par quatre phases principales de développement :

- appropriation du cahier des charges du projet et modélisation UML du comportement du système incluant un diagramme de cas d'utilisation, des diagrammes de séquences et un diagramme de classe
- pilotage du projet sous Jira
- implémentation du code en Java sous forme d'un projet Maven
- intégration continue avec Jenkins (cette phase ne sera pas couverte dans ce rapport)

Elles sont interdépendantes et leurs contenus sont mis à jour continuellement en vue de rester en phase avec l'évolution du projet.
Nous retracerons dans ce rapport les grandes lignes de chacune de ces phases de développement, accompagné d'un manuel d'utilisation simplifié du ChatSystem développé.

I. Appropriation du cahier des charges et modélisation UML du ChatSystem

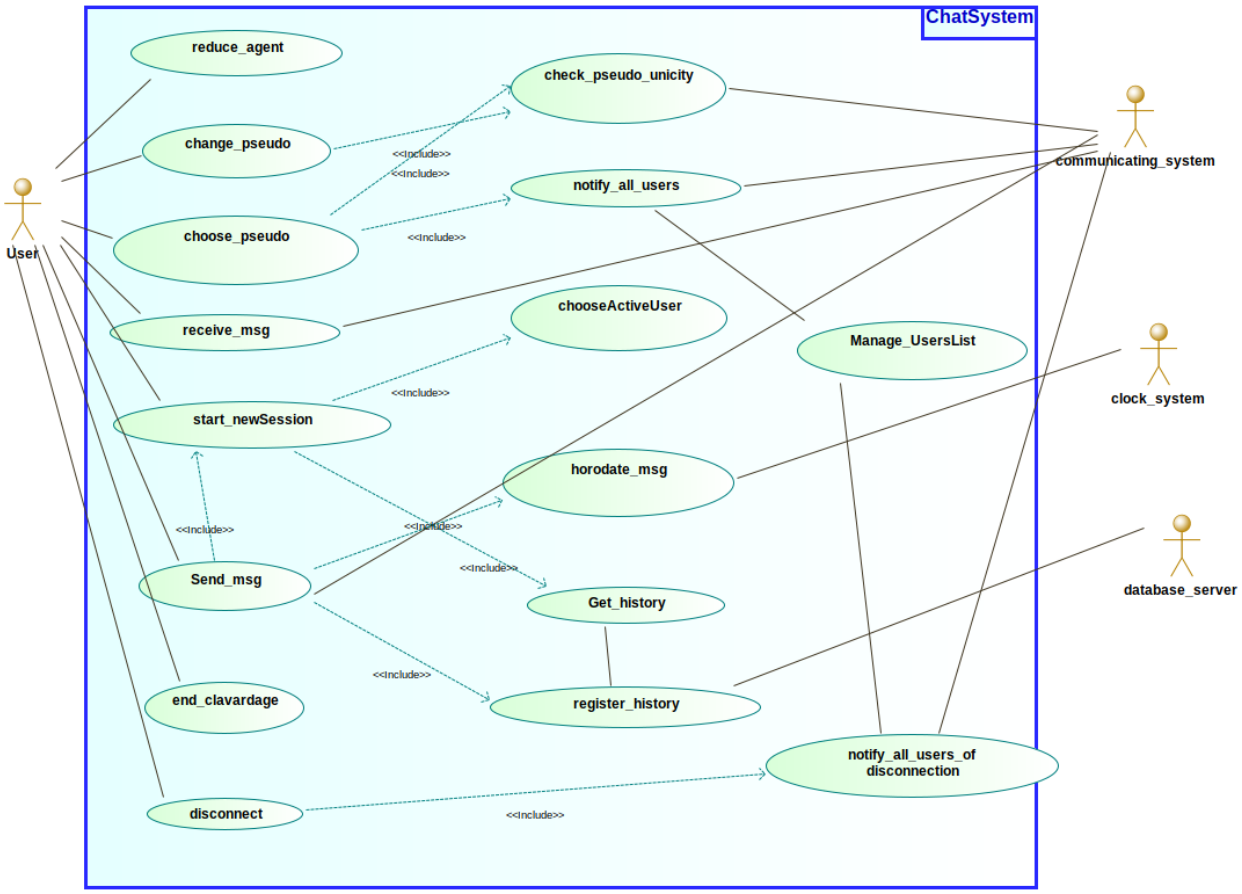
Cette partie est fondamentale pour le démarrage du projet car elle permet d'assurer une compréhension claire et commune du système au sein de l'équipe.

I.a) Récapitulatif des fonctions attendues :

Ce système permettra aux utilisateurs de communiquer en envoyant et en recevant des messages texte à l'aide d'appareils interconnectés.

- ➢ Chaque utilisateur utilise un nom d'utilisateur ou pseudo pour se connecter au système de chat. Deux utilisateurs ne peuvent pas utiliser le pseudo.
- ➢ Lorsqu'un utilisateur se connecte au système, la liste des autres utilisateurs connectés est mise à jour. Cette liste comprend les noms d'utilisateurs connectés.
- ➢ Lorsqu'un utilisateur change de pseudo, la liste des autres utilisateurs connectés est mise à jour. Seuls les utilisateurs connectés peuvent communiquer en utilisant les fonctions du système de chat.
- ➢ Un utilisateur peut mettre fin à une session de chat
- ➢ Lorsqu'un utilisateur se connecte ou se déconnecte, les autres utilisateurs doivent en être informés.
- ➢ Lorsqu'un utilisateur souhaite communiquer avec un autre utilisateur (envoyer un message), il doit sélectionner l'utilisateur distant dans la liste des utilisateurs connectés. Le message à envoyer doit être indiqué puis envoyé.
- ➢ Lorsque le système reçoit un message destiné à l'utilisateur local connecté, l'utilisateur doit en être informé (c'est-à-dire en affichant le message).

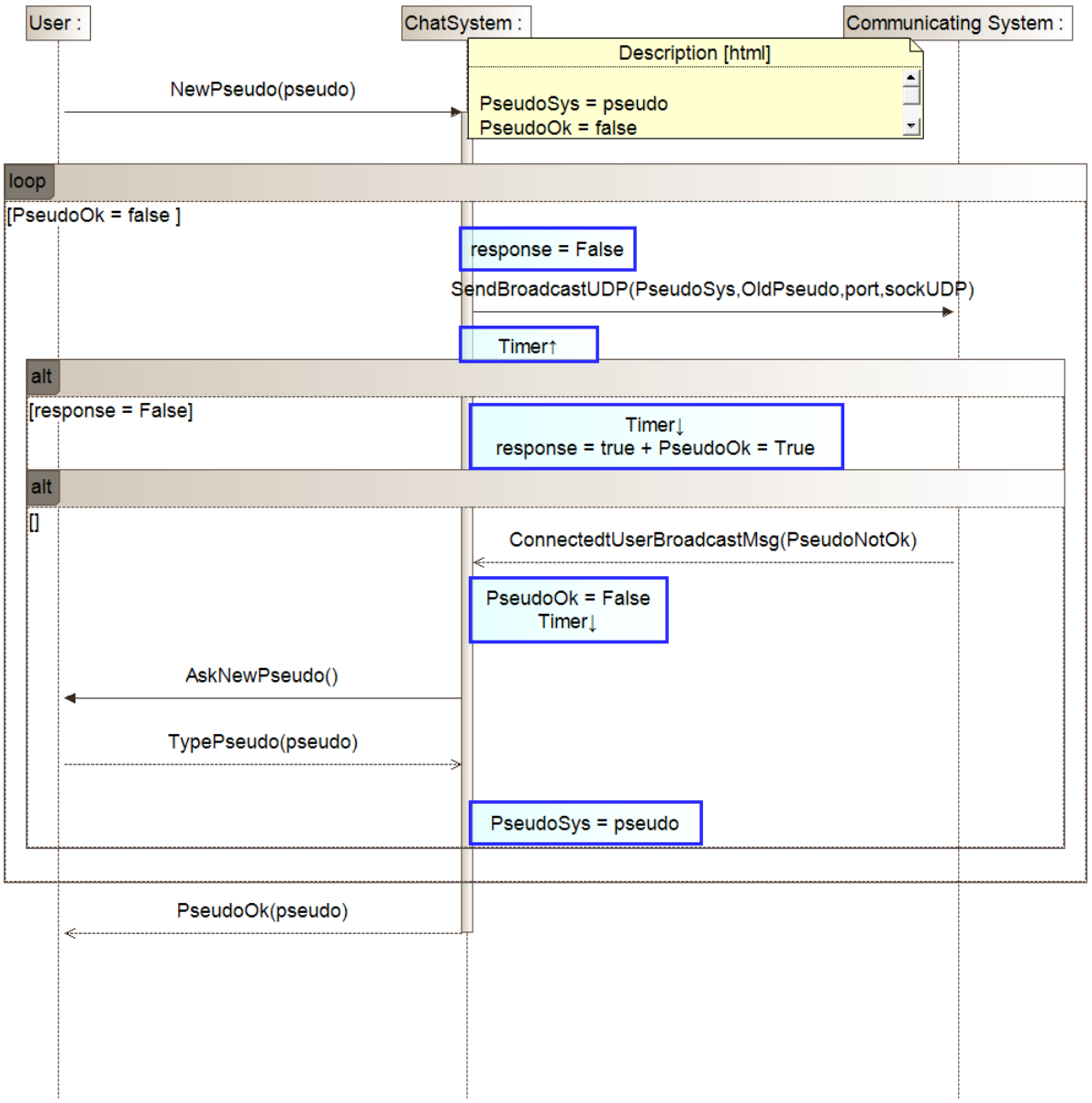
I.b) Diagramme de cas d'utilisation du ChatSystem :



I.c) Diagrammes de séquence associés à quelques cas d'utilisation :

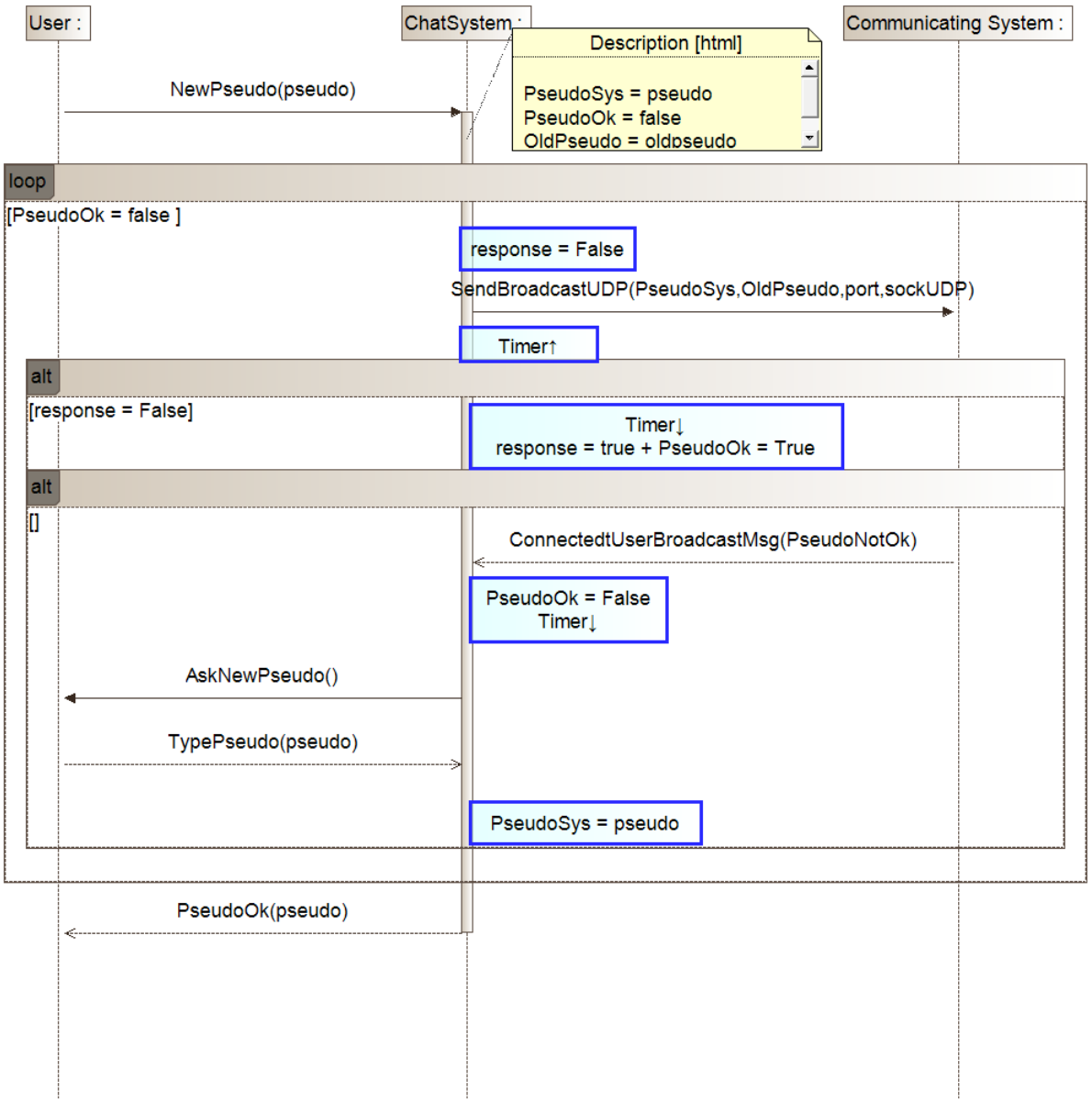
Nous avons fait l'exercice d'établir les diagrammes de séquences associés aux quelques cas d'utilisation suivants, dans le but :
- d'identifier les différents acteurs pouvant intervenir dans le fonctionnement du Chatsystem et la nature de leurs interactions
- d'élaborer les algorithmes qu'on pourrait mettre en place pour les réaliser

→ Cas d'utilisation 1 - ChoosePseudo() : permet à l'utilisateur de se connecter au chat

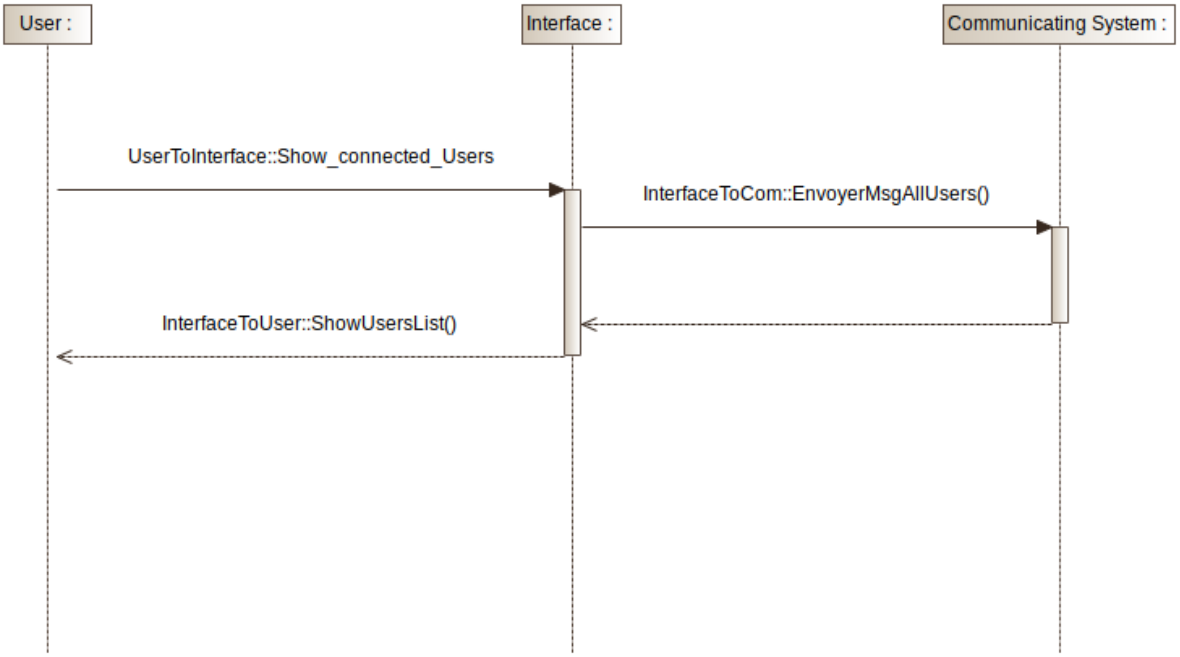


→ Cas d'utilisation 2 - ChangePseudo() : permet à l'utilisateur de changer son pseudo

RQ : en raison de la version de modelio utilisée, ne permettant pas d'afficher les champ du bloc "alt", le second bloc alt constitue le champ "else" du premier bloc alt

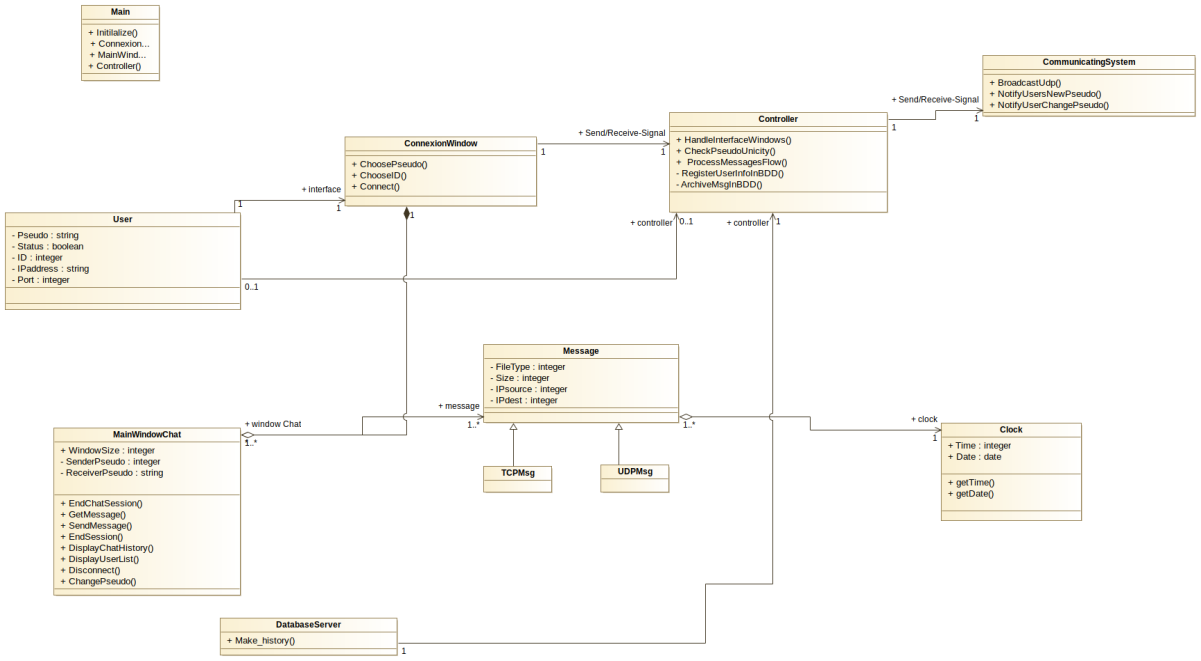


→ Cas d'utilisation 3 - ShowUserList() : permet à l'utilisateur d'afficher la liste des utilisateurs connectés



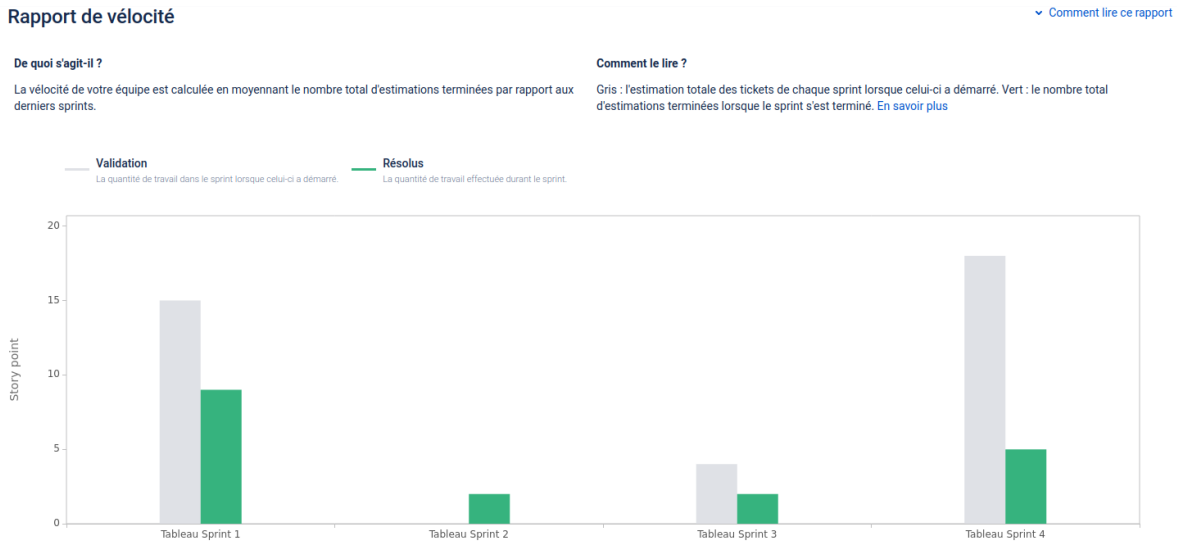
I.d) Diagramme de classe du ChatSystem :

Au démarrage du projet :



II. Pilotage du projet sous Jira :

Nous avons mis en place 5 tableaux de Sprint pour réaliser ce projet. Nous sommes actuellement en phase de finalisation du Sprint 5.
Un rapport de vélocité sur les quatre premiers Sprint est présenté ci-dessous. Nous remarquerons qu'il a été un réel défi de parvenir à finir toutes les tâches attribuées à chaque Sprint dans les temps impartis.



Sprint	Validation	Résolus
Tableau Sprint 1	15	9
Tableau Sprint 2	0	2
Tableau Sprint 3	4	2
Tableau Sprint 4	18	5

III. Implémentation proprement dite du ChatSystem :

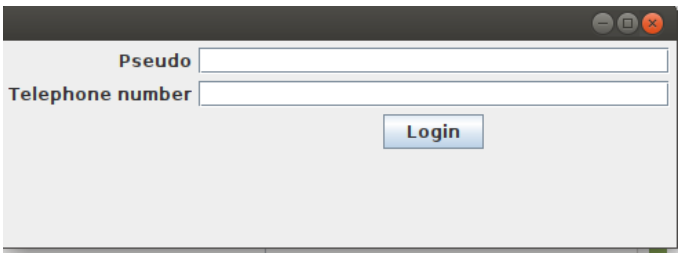
III.a) Architecture du système et choix technologiques :

Nous avons fait les choix suivants :
→ base de de données : base de données CENTRALE
Pour accéder à celle-ci il vous faudra avoir accès au VPN de l'INSA et vous connecter avec l'identifiant : tp_servlet_012 et mot de passe : Thi0zaes en saisissant les lignes de commandes suivantes :

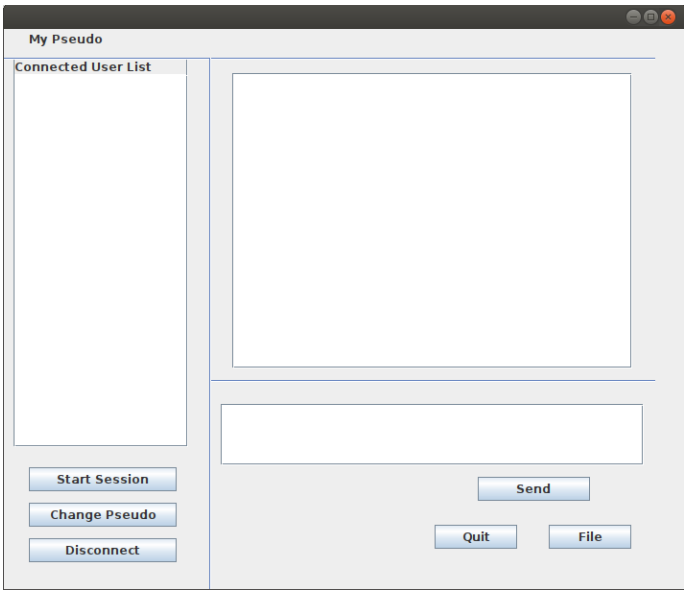
mysql -h srv-bdens -P 3306 -u tp_servlet_012 -p
Puis saisissez le mot de passe : Thi0zaes

→GUI : nous avons mis en place 2 fenêtres principales pour couvrir les fonctionnalités proposées par le ChatSystem :

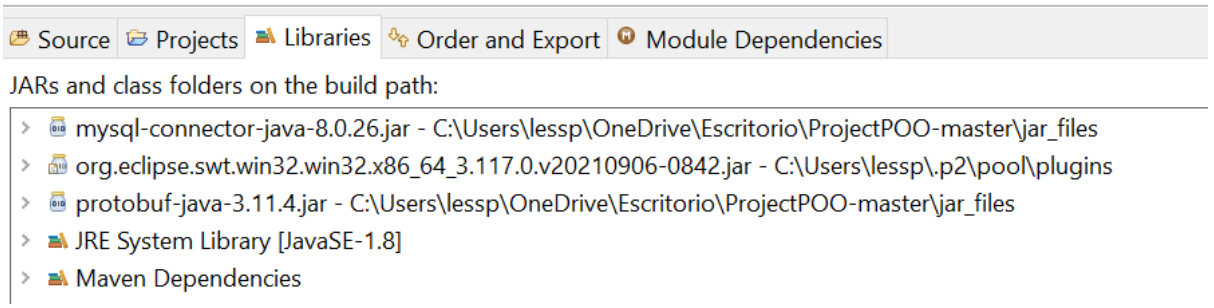
- la fenêtre de connexion, où l'utilisateur devra renseigner son pseudo et son numéro de téléphone qui lui servira de clé d'identification unique.



- la fenêtre principale de chat



→ libs utilisées :



→ identifiant unique pour les utilisateurs :

Le pseudo pouvant varier au cours d'une même session pour un même utilisateur connecté, nous avons introduit "le numéro de téléphone" comme clé d'identification unique d'un utilisateur.

III.b) Procédures d'évaluation et de test

Stade d'avancement actuel du projet : développement en backend et mise en place de tests pour la démonstration du fonctionnement correct de toutes les fonctions implémentées.
Des indications plus précises sont renseignées dans le readme : https://github.com/Diegostd/ProjectPOO/blob/diego/README.md

III.c) Manuel d'utilisation simplifié

Le manuel d'utilisation simplifié est accessible sur notre dépôt git, sur la branche diego.

CONCLUSION :

En guise de conclusion, nous pouvons dire que ce projet nous a permis d'endosser plusieurs casquettes : product owner et développeur tout en nous familiarisant à l'environnement de pilotage de projet Jira et aux logiciels de développement : Git, Modelio, EclipseJee. Bien que ces divers environnements de travail ont été bien conçus pour permettre aux membres de l'équipe un travail en parallèle, nous avons constaté que ça ne garantissait pas toujours une compréhension commune du problème ou encore une implication plus ou moins équitable de chaque participant dans le projet. Un effectif de participants adéquat aurait pu rendre le projet plus intéressant.