

Rapport Projet COO-POO

Sirine CHAOUCH - Wissal ELKOURDI - Léonie GOUTTE

4 IR B2

Année scolaire 2022-2023

Table des matières

I. Choix de conception	3
1. Diagramme de séquence de la connexion	3
2. Diagramme de séquence de la déconnexion	4
3. Diagramme de séquence des échanges	5
4. Diagramme de classe	6
II. Architecture du système et choix technologiques (bases de données, GUI, toutes autres libs utilisées)	6
III. Procédures d'évaluation et de tests	7
IV. Procédure d'installation et de déploiement (commandes, configs nécessaires, script, etc.)	7
V. Manuel d'utilisation simplifié	7
Annexes	11

I. Choix de conception

1. Diagramme de séquence de la connexion

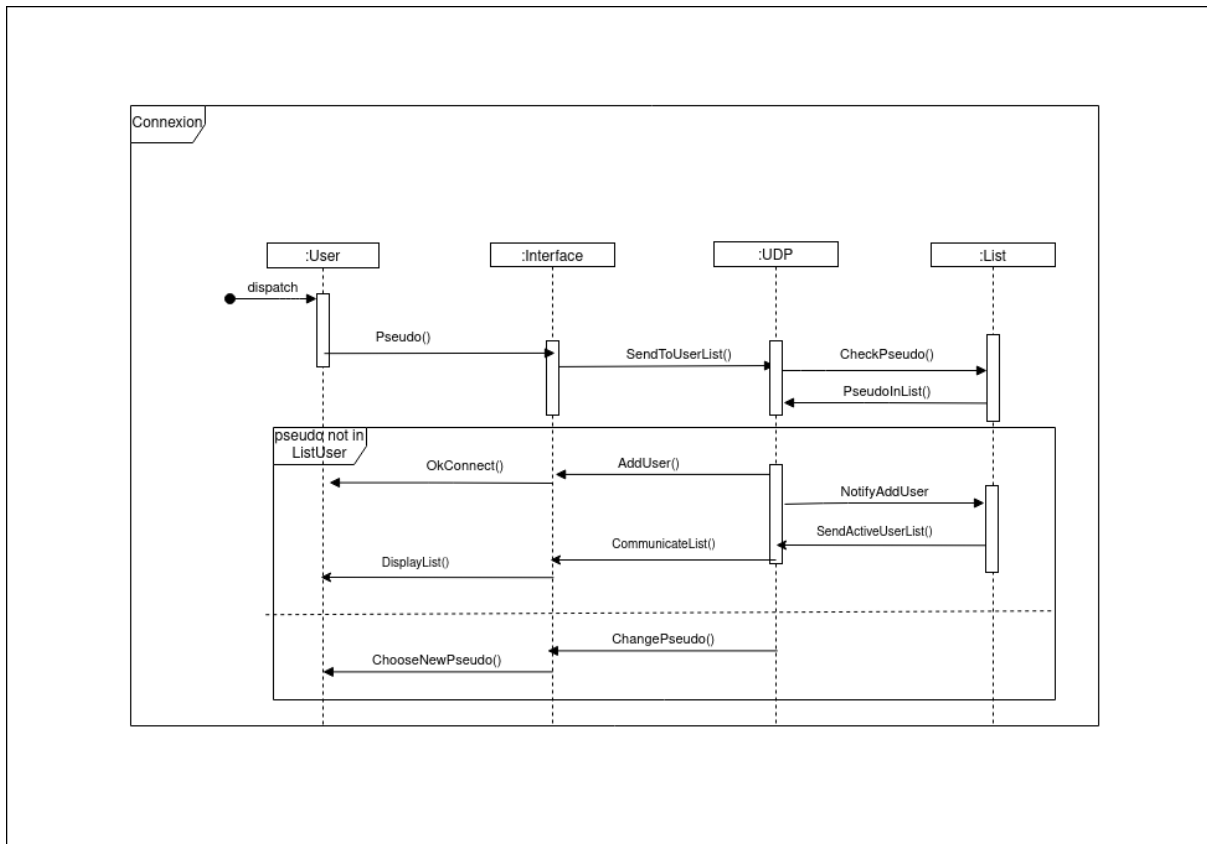


Figure 1 : diagramme de séquence de connexion

L'utilisateur choisit un pseudo, qui est communiqué à UDP par l'interface. UDP le compare ensuite à la liste des utilisateurs connectés. S'il n'est pas dans la liste, l'utilisateur garde ce pseudo et peut se connecter et on l'ajoute dans la liste des utilisateurs connectés. Sa propre liste est également mise à jour. Si ce nom est déjà dans la liste, il devra changer son pseudo.

2. Diagramme de séquence de la déconnexion

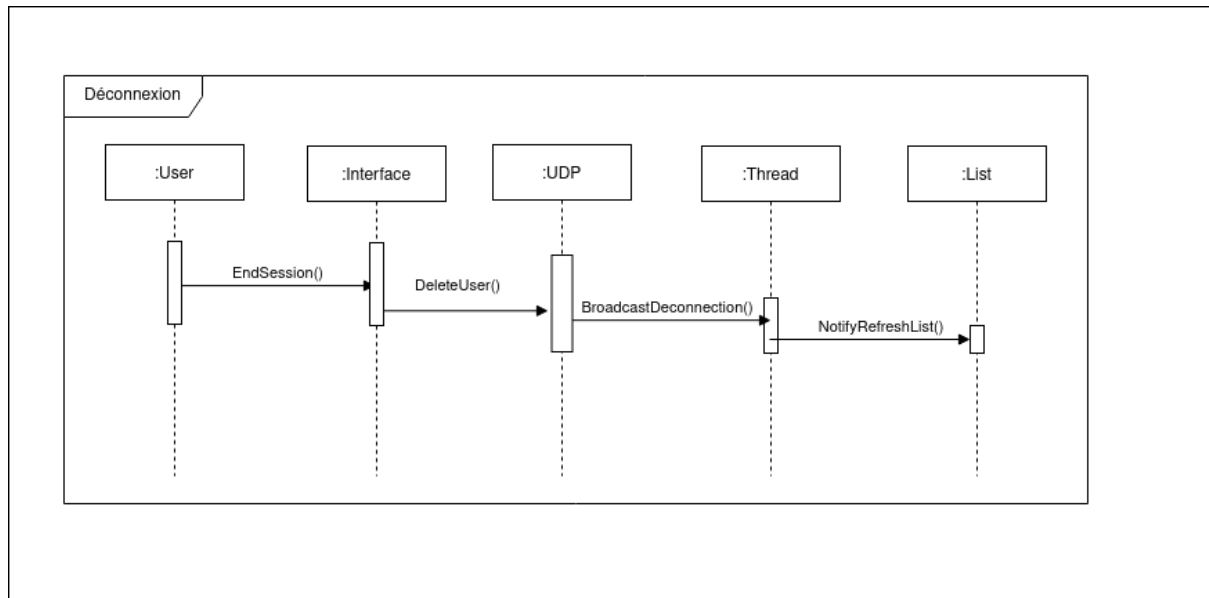


Figure 2 : diagramme de séquence de déconnexion

Dès qu'un utilisateur choisit de se déconnecter par le biais de l'interface, un broadcast de déconnexion est envoyé et on le supprime de toutes les listes des utilisateurs connectés, puisque chaque utilisateur à sa propre liste dans sa base de données.

3. Diagramme de séquence des échanges

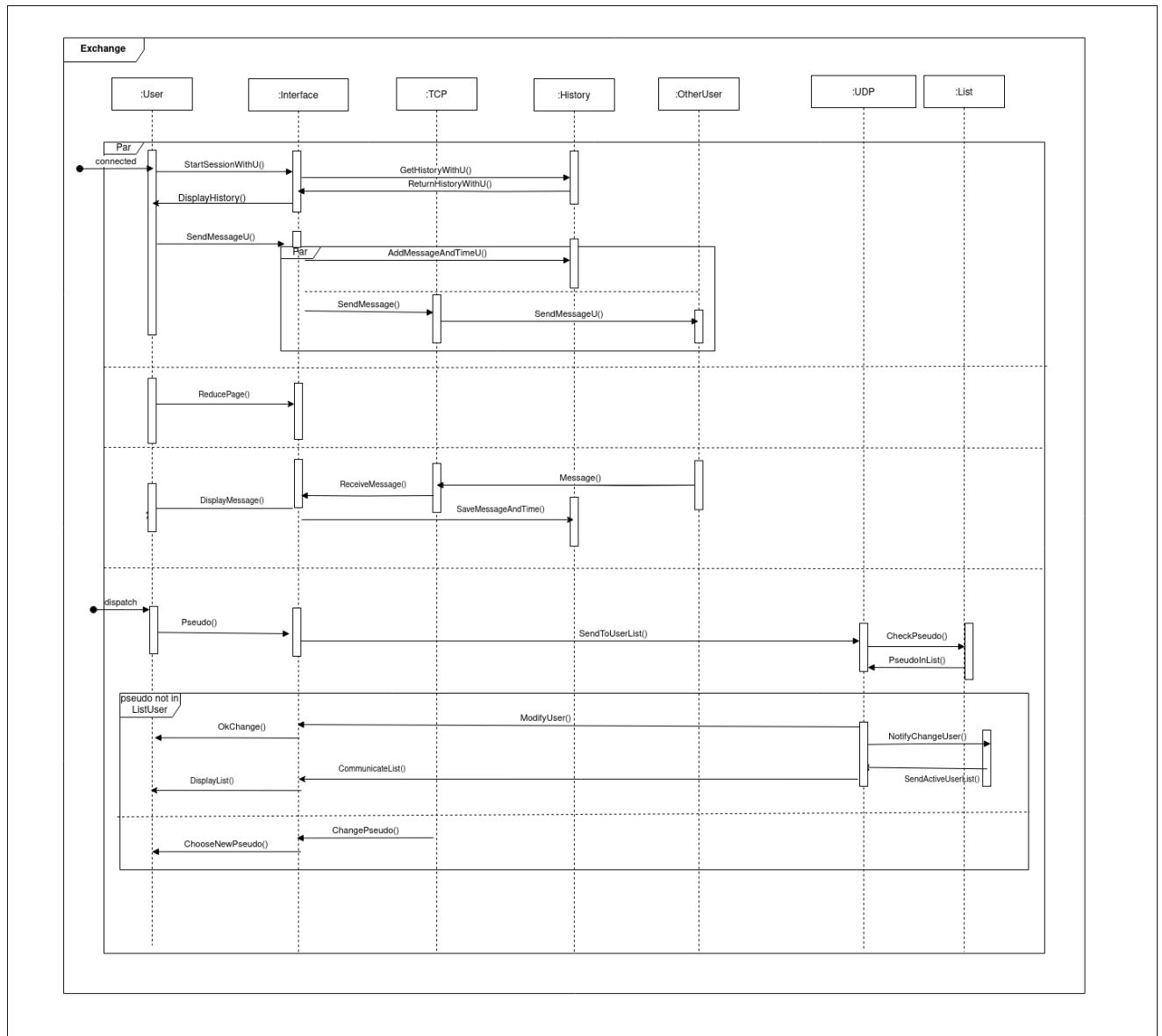


Figure 3 : diagramme de séquence d'envoi et de réception de message, de changement de pseudo et de réduction de la fenêtre

Lorsqu'un utilisateur choisit d'ouvrir une session avec un autre utilisateur, l'interface demande à sa BDD de lui retourner l'historique des messages pour que l'utilisateur les voit dès le début de la session. Lorsque l'utilisateur envoie un message, l'interface le communique à la BDD afin de l'ajouter dans l'historique ainsi que son heure d'envoi. En même temps, le message est envoyé à l'autre utilisateur grâce à TCP.

Pour réduire la page de l'application il suffit simplement d'appuyer sur le bouton réduire l'onglet en haut à droite pour que l'interface se charge de diminuer l'application.

Lorsque un autre utilisateur envoie un message par TCP, l'interface de notre utilisateur le récupère et l'affiche, puis il est ajouté à la base de données locale en sauvegardant l'heure de réception.

Pour le changement de pseudo, l'utilisateur entre le nouveau pseudo sur l'interface, qui en utilisant UDP interroge la liste des utilisateurs connectés pour vérifier que le nouveau pseudo n'est pas déjà utilisé. Dans ce cas, UDP prévient l'interface ainsi que les autres utilisateurs que le changement de pseudo a eu lieu, le modifie dans la liste des utilisateurs connectés (la sienne et celle des autres utilisateurs) qui est ensuite affichée. Si le pseudo est déjà présent dans la liste de l'utilisateur, l'interface prévient l'utilisateur qu'il doit en choisir un autre.

Vous trouverez le diagramme en version plus lisible dans l'annexe pages 13 et 14.

4. Diagramme de classe

Nous avons choisi de faire apparaître les différents packages sur le diagramme de classe, et de ne pas faire une représentation exhaustive de toutes les classes pour une meilleure compréhension. Ainsi, nous représentons une seule classe controller pour simplifier.

Vous trouverez le diagramme dans l'annexe pages 15.

II. Architecture du système et choix technologiques (bases de de données, GUI, toutes autres libs utilisées)

Nous avons choisi d'implémenter une base de données décentralisée : chaque utilisateur possède sa propre base de données contenant la liste des utilisateurs qui se sont déjà connectés en même temps que lui (et s'ils le sont actuellement), ainsi que tous les messages déjà échangés par cet utilisateur.

En ce qui concerne le GUI, nous avons implémenté une page de connexion simple où l'utilisateur peut choisir son nom d'utilisateur sur l'application. Le nom d'utilisateur est vérifié afin qu'il ne contienne pas de symbole non autorisé et soit de la bonne taille.

Une fois connecté, l'utilisateur arrive sur le menu, où il peut voir la liste des utilisateurs connectés à gauche de l'écran. Il peut ouvrir une session de chat avec un utilisateur en cliquant sur son nom dans la liste. Un nouvel onglet de chat s'ouvre après le clic. Le menu contient également des boutons de changement de pseudo et de déconnexion. L'onglet correspondant au menu ne peut pas être fermé par l'utilisateur.

Lorsque l'utilisateur a commencé une session de chat, il peut envoyer des messages en appuyant sur le bouton send. Il peut également faire défiler tous les messages reçus et envoyés à l'autre utilisateur. La déconnexion et le changement de pseudo ne sont pas accessibles sur la fenêtre de chat, il faut simplement retourner sur l'onglet menu (sans fermer les sessions de chat en cours).

Lorsqu'un utilisateur se déconnecte alors que nous avons une session de chat ouverte avec lui, l'onglet se ferme et son pseudo disparaît de la liste des utilisateurs connectés.

III. Procédures d'évaluation et de tests

Nous avons réalisé plusieurs tests unitaires pour s'assurer qu'une méthode particulière, une fonction, une classe, ... etc., effectue un ensemble de tâches spécifiques, correctement. Vous pouvez ainsi trouver dans la partie Test l'ensemble des classes testées (Handler, la base de donnée, quelques classes controllers etc ...).

IV. Procédure d'installation et de déploiement (commandes, configs nécessaires, script, etc.)

Pour installer et déployer notre application, il est simplement nécessaire d'avoir un SDK sur son ordinateur. Nous avons ajouté des plugins et dépendances Maven qui permettent de déployer l'application sans installer javafx.

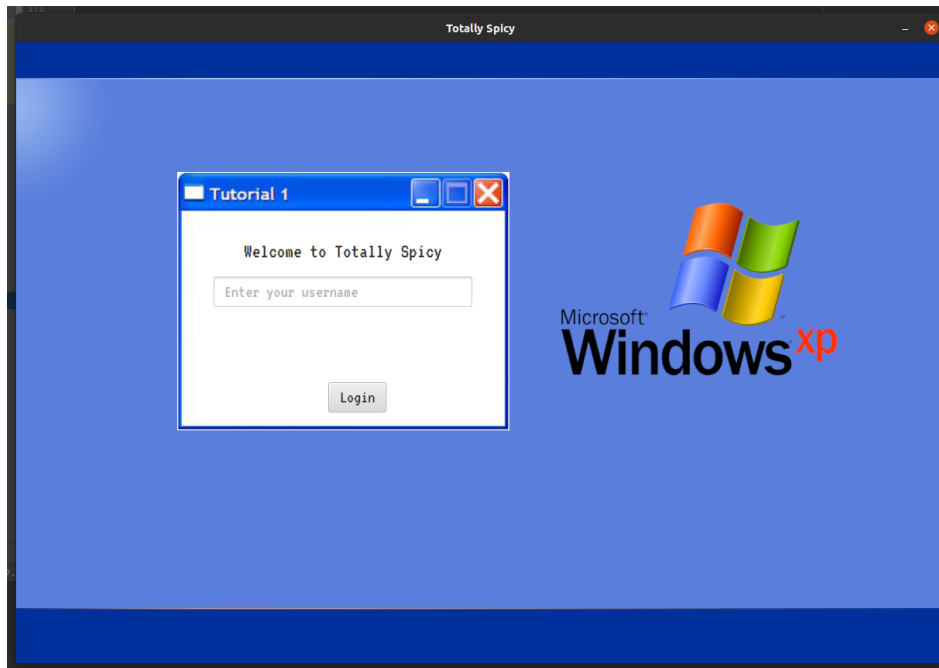
Pour l'installer, copier les commandes ci-dessous (dans un terminal ouvert dans le dossier contenant le fichier .jar)

```
mkdir sqlite  
java -jar <nom_du_fichier>
```

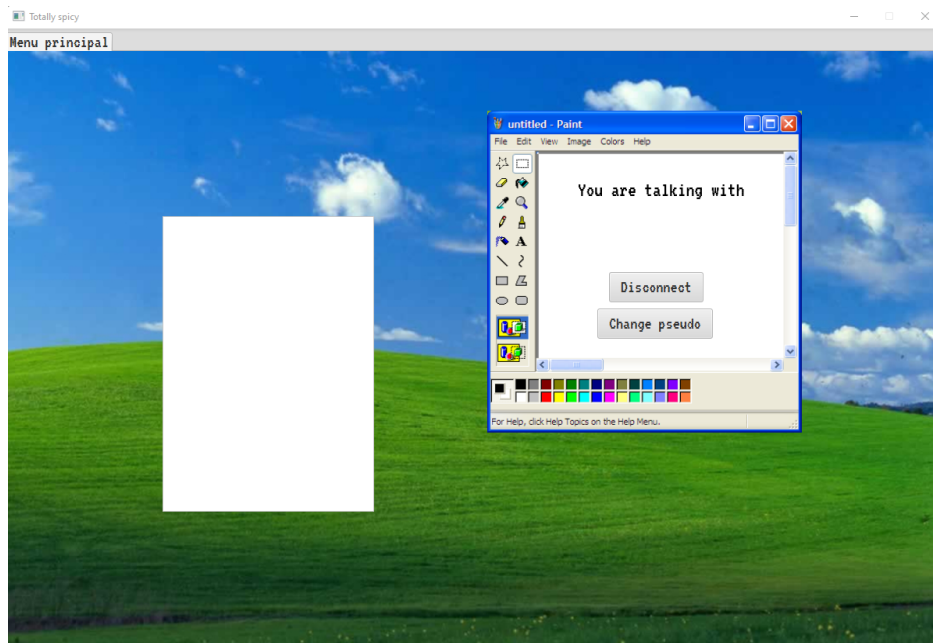
V. Manuel d'utilisation simplifié

Ceci est notre page de connexion. On demande à l'utilisateur de rentrer son pseudo et on vérifie si :

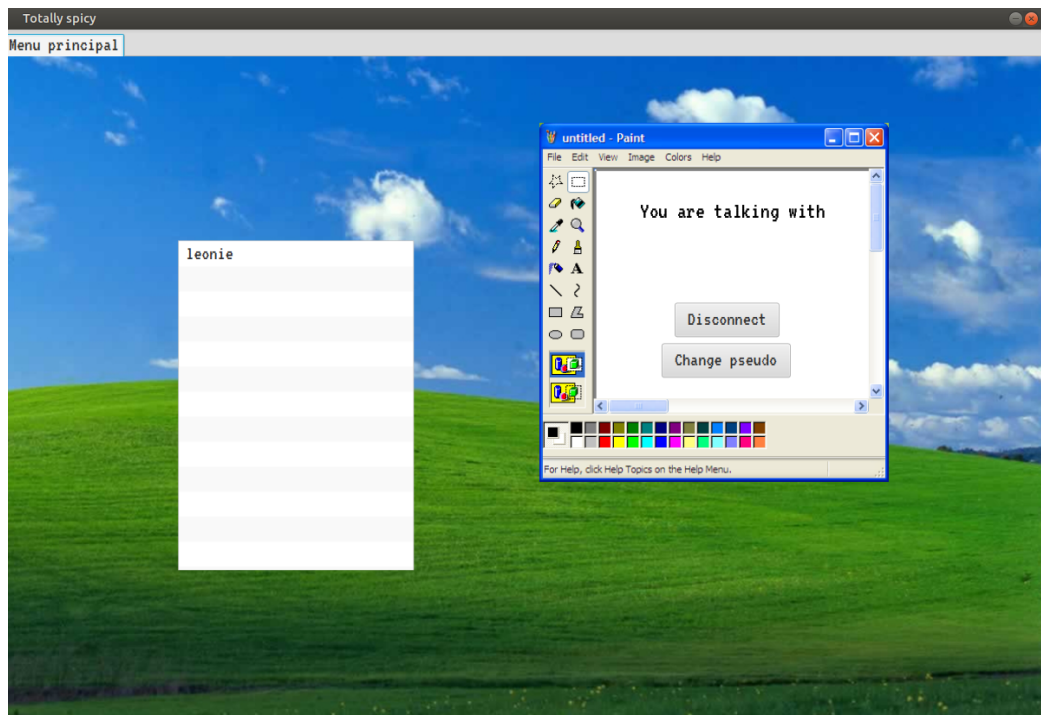
1. le pseudo ne contient pas de caractères spéciaux , si c'est le cas, on lui affiche un message lui disant de choisir un autre pseudo.
2. le pseudo est déjà utilisé, dans ce cas on prévient l'utilisateur avec un message en lui disant que le pseudo est déjà utilisé et qu'il doit en choisir un autre



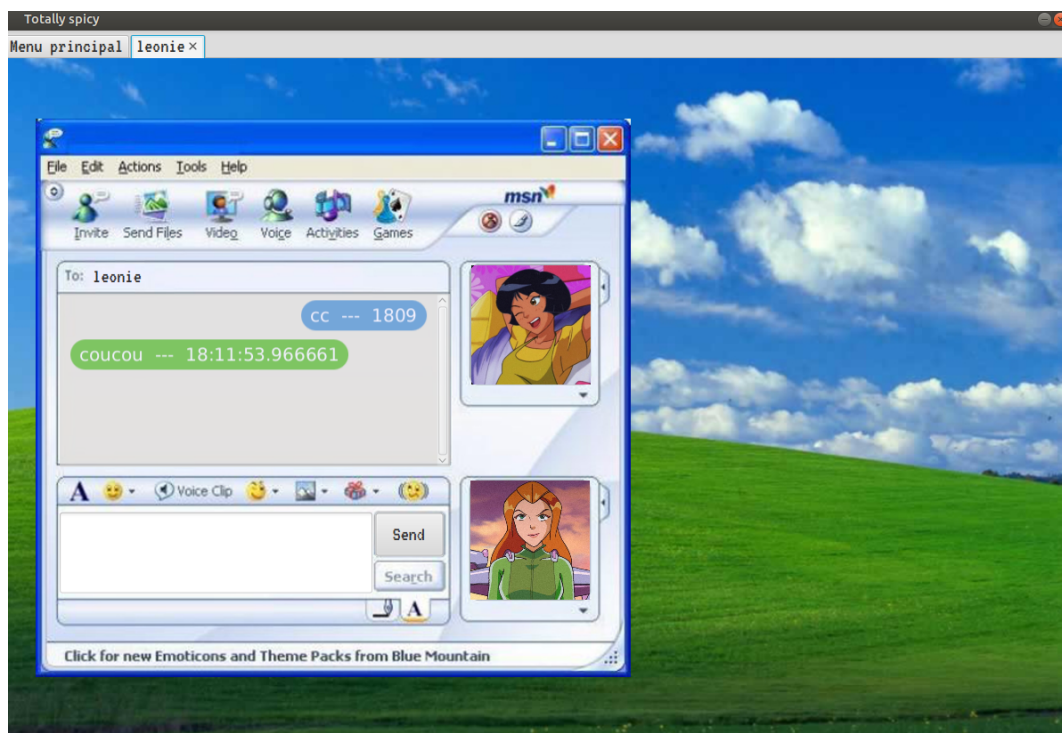
Une fois l'utilisateur est connecté une page d'accueil s'affiche :



Sur le rectangle à gauche, on affiche la liste des utilisateurs connectés.
Supposons qu'un user sous le nom de "leonie" est connecté:



Ensuite, on peut entamer une discussion avec “leonie” en cliquant sur son pseudo sur notre liste d'utilisateurs et on obtient la page suivante :



Le message à droite en bleu est le message envoyé par l'utilisateur tandis que le message situé à gauche est celui reçu par l'utilisateur.

Enfin, si jamais l'utilisateur veut se déconnecter, il n'a qu'à cliquer sur le bouton "disconnect" du menu principal, il sera donc redirigé vers la page de connexion, et toute session ouverte avec lui du point de vue des autres utilisateurs sera fermée.
Idem pour le cas du changement de pseudo.

Si l'utilisateur appuie sur la croix en haut de la fenêtre à n'importe quel moment, le processus sera fermé et l'utilisateur sera déconnecté. Il disparaîtra donc des listes des autres utilisateurs.

Annexes

Diagramme de séquence - connexion

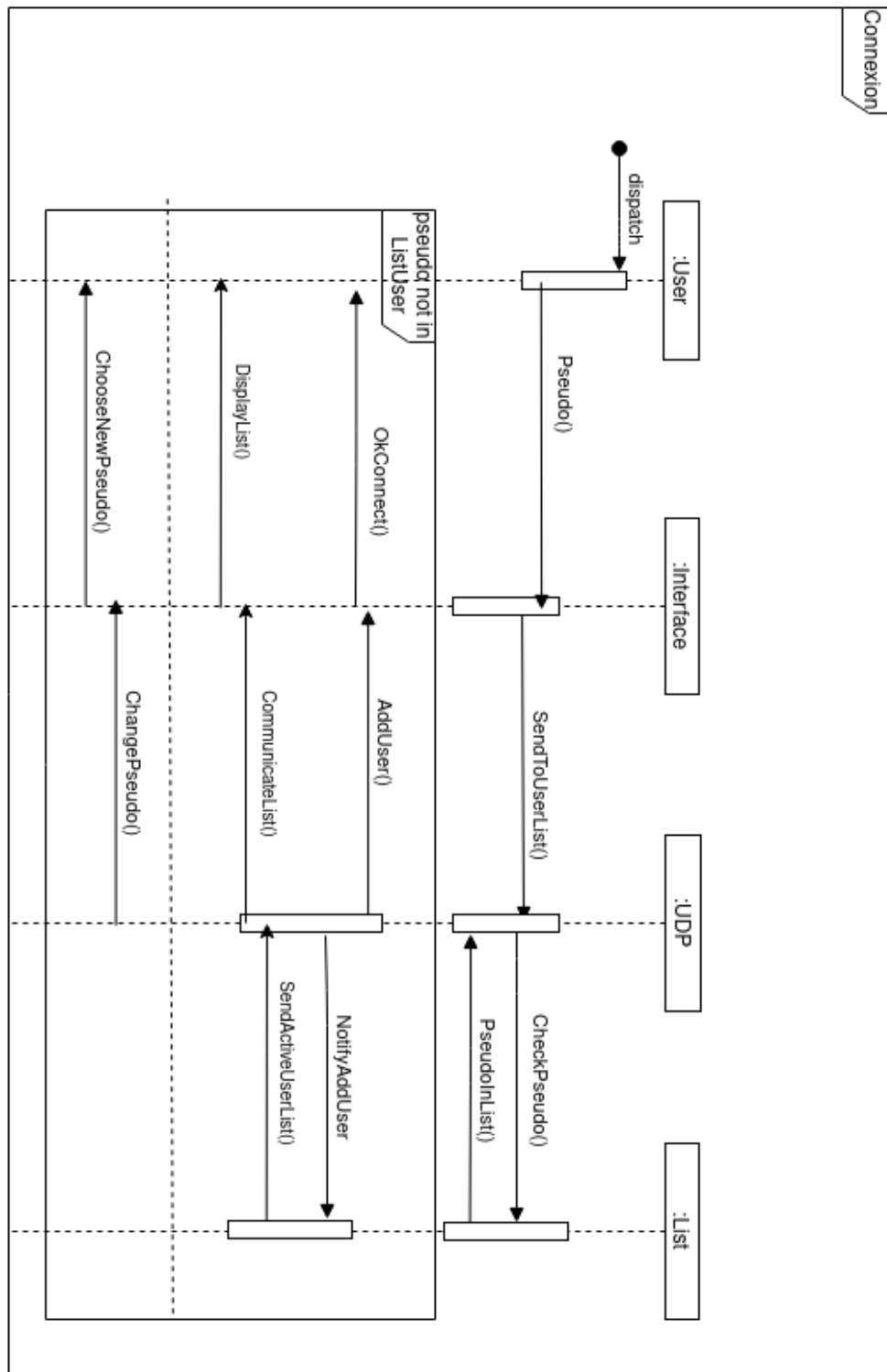


Diagramme de séquence - déconnexion

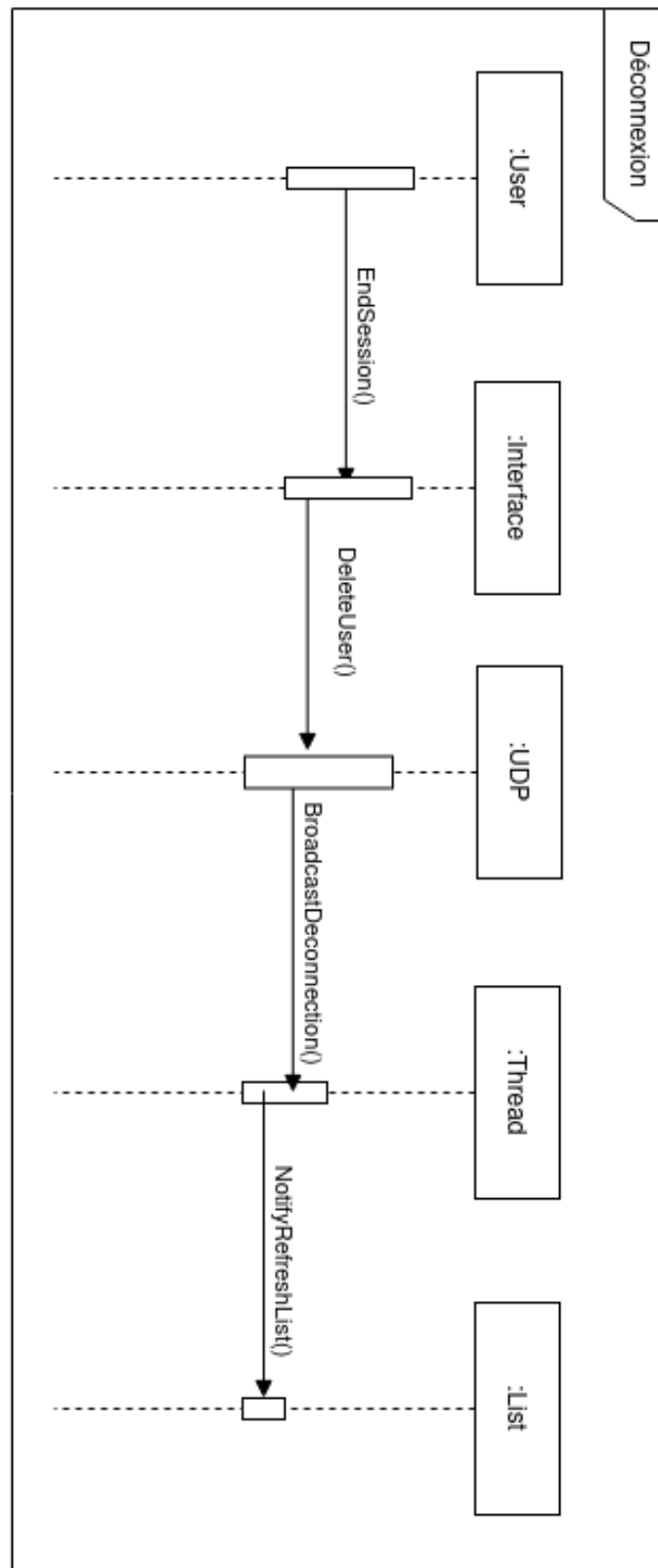


Diagramme de séquence - échanges - partie 1 : envoi de messages et réduction de la page

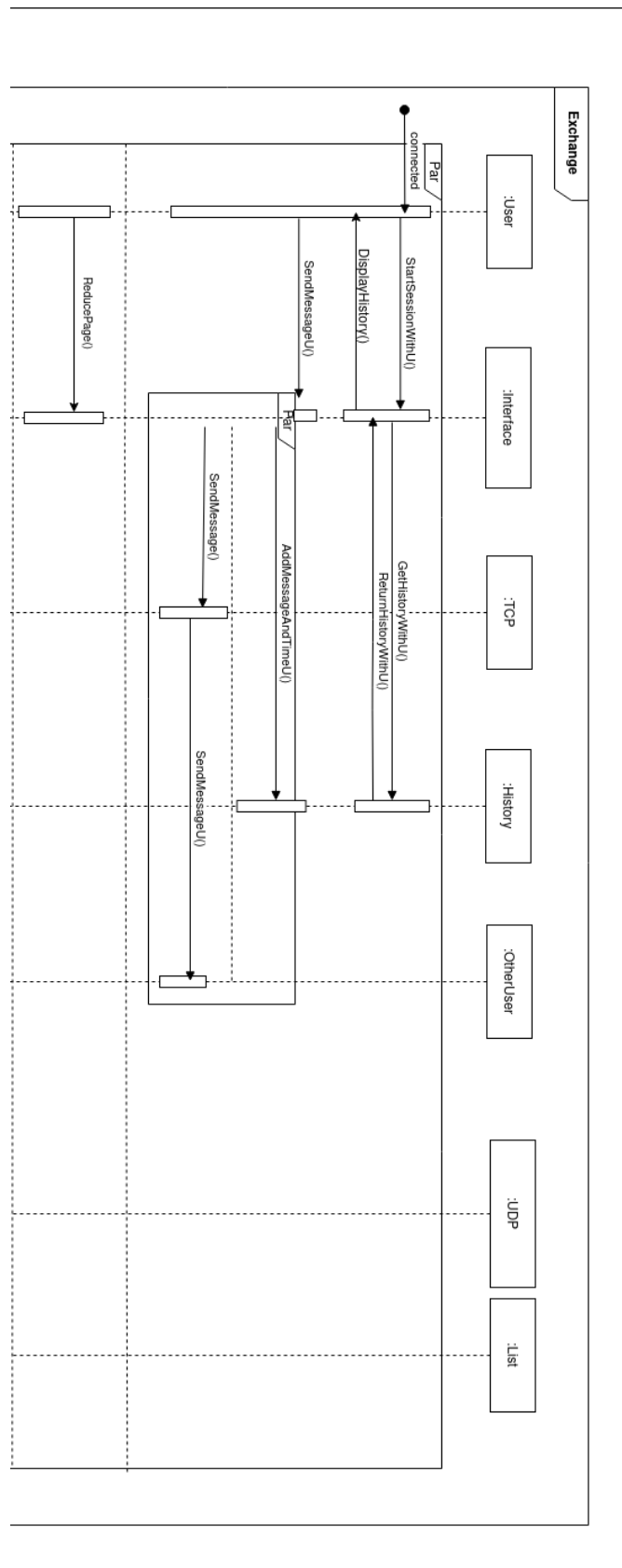


Diagramme de séquence - échanges - partie 2 : réception de messages et changement de pseudo

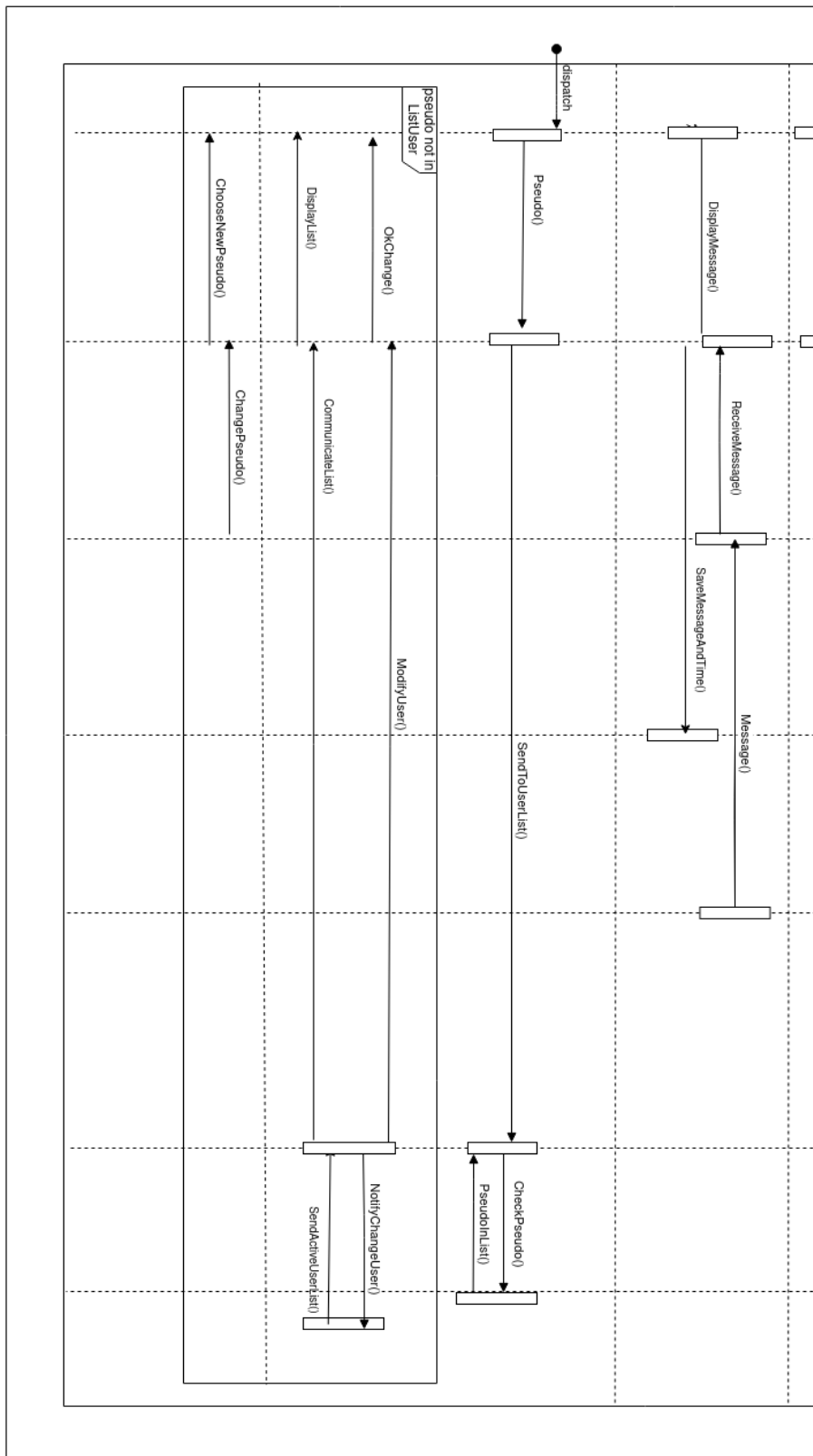


Diagramme de classes

