

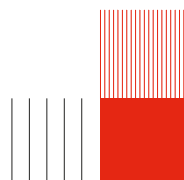
RAPPORT DE CONCEPTION

PROJET « MIAOU CHAT » :
APPLICATION DE CLAVARDAGE

Marie Mecaliff
Thomas Cayla Ginestet

4IR B

2022/2023

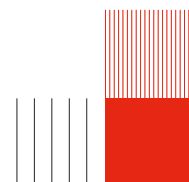


RAPPORT DE CONCEPTION

**PROJET « MIAOU CHAT » :
APPLICATION DE CLAVARDAGE**

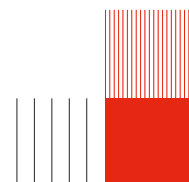
Marie Mecaliff
Thomas Cayla Ginestet

4IR B
2022/2023



Sommaire

Introduction	5
I. Conception et diagramme.....	6
1. Diagramme de cas d'utilisation.....	6
2. Diagramme de classes	7
3. Diagrammes de séquence	8
a) Connexion d'un utilisateur à l'application.....	8
b) Envoi de messages entre utilisateurs dans une session de clavardage.....	9
c) Déconnexion d'un utilisateur à l'application	10
4. Schéma de conception de la base de données	11
5. Interface graphique.....	12
II. Procédures de tests, d'installation et de déploiement.....	13
III. Manuel d'utilisation simplifié	14



Introduction

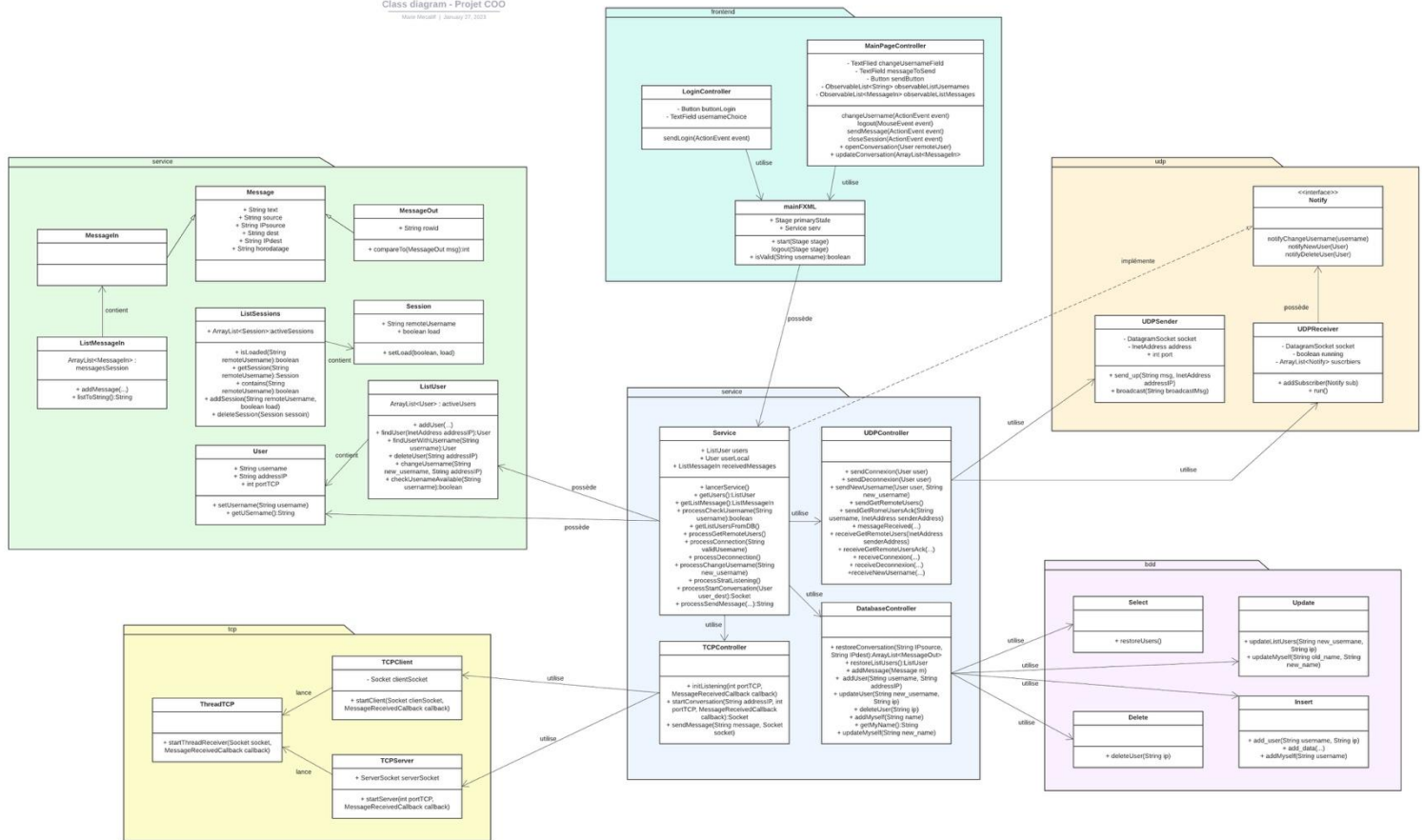
Ce rapport présente les résultats de production et les choix de conception faits afin de mener à bien le projet de Programmation et Conception Orientées Objet de la quatrième année Informatique et Réseaux de l'INSA Toulouse.

L'objectif de ce projet était de développer une application de clavardage, permettant à plusieurs utilisateurs situés sur un même réseau local de communiquer par envoi de messages. Ainsi, les fonctionnalités suivantes sont contenues dans le fonctionnement de notre application :

- une visibilité constante sur la liste des utilisateurs actuellement connectés ;
- le choix d'un pseudonyme au moment de la connexion, sachant que pour être accepté comme valide pour l'application, ce pseudonyme doit être unique vis-à-vis des autres utilisateurs connectés et ne comporter aucun espace ou caractère spécial ;
- le changement de pseudonyme à n'importe quel moment, tant qu'il respecte les conditions citées précédemment, et la communication immédiate du nouveau pseudonyme à tous les autres utilisateurs connectés ;
- l'ouverture d'une session de chat avec un des utilisateurs connectés, et la récupération de l'historique des messages échangés précédemment avec l'utilisateur sélectionné à l'ouverture de la conversation ;
- l'affichage de l'horodatage de chaque message échangé ;
- enfin, les messages échangés sont conservés dans une base de donnée centralisée, ainsi que la liste des utilisateurs connectés et les informations sur l'utilisateur local.

2. Diagramme de classes

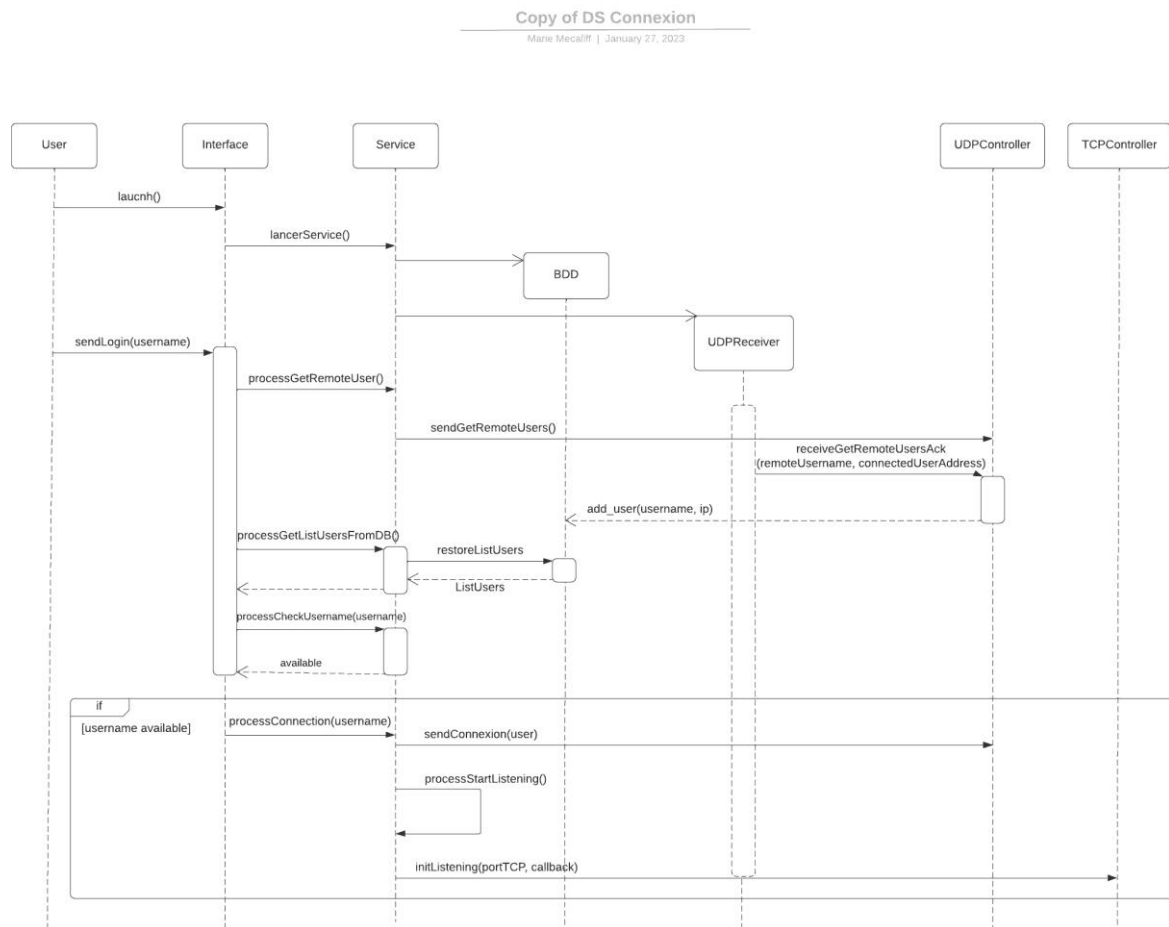
Class diagram - Projet COO
Mickaël BOUTIN | January 27, 2023



Ce diagramme de classe présente une version simplifiée de l'architecture du système, où seules les classes principales et leurs méthodes les plus importantes sont décrites, afin de faciliter la compréhension et la lisibilité. L'architecture de notre projet est basée sur une organisation en packages indépendants les uns des autres, répartis par fonctionnalités principales et organisés en couches. La couche réseau de la partie communication est répartie en deux packages différents selon le protocole : *tcp* et *udp*. Les éléments principaux du modèle sont réunis dans le package *data* et les classes de type Controller sont réunies dans le package *service*. Enfin, la base de données et de l'interface possède chacune leur package spécifique, respectivement *bdd* et *frontend*.

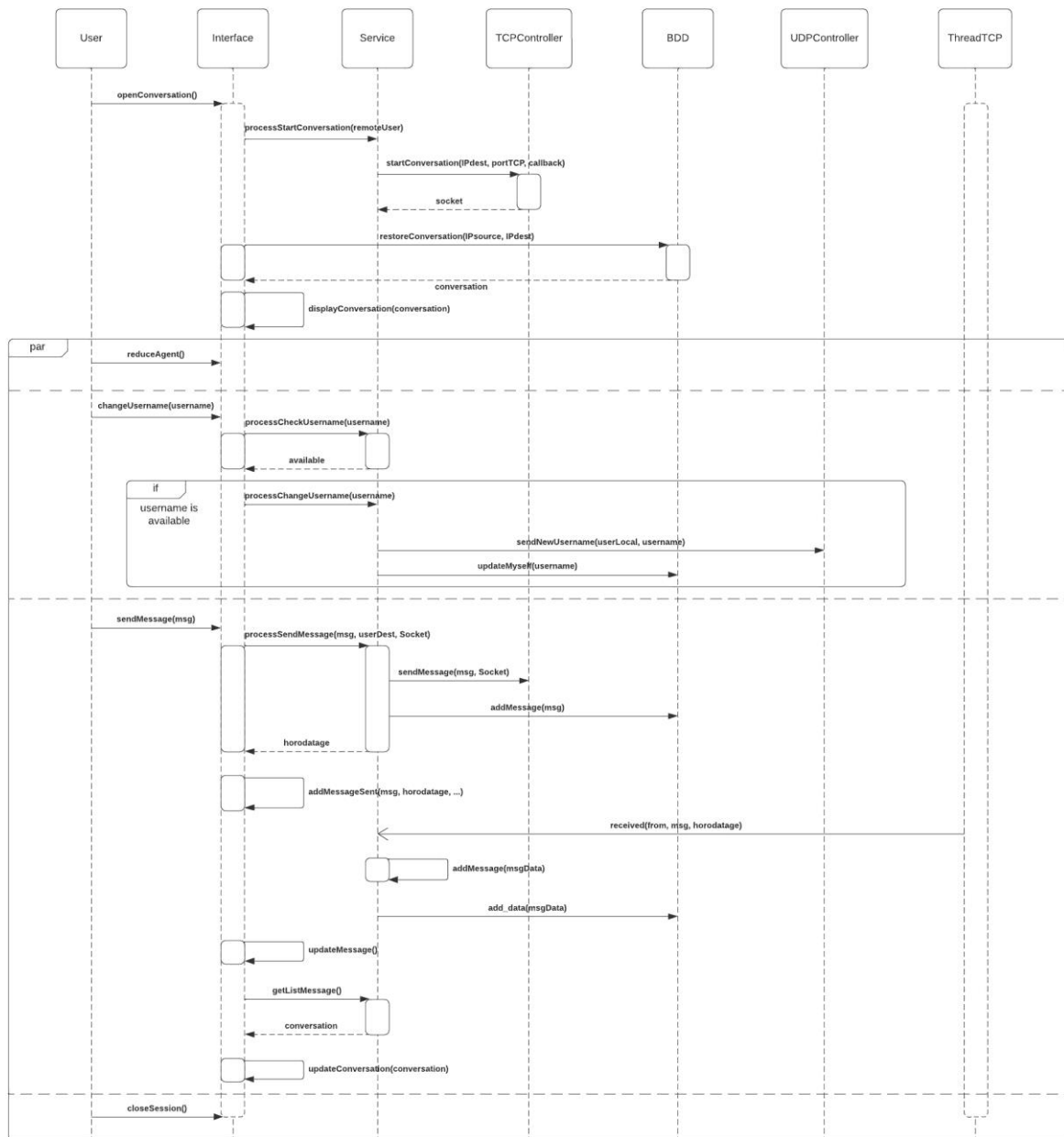
3. Diagrammes de séquence

a) Connexion d'un utilisateur à l'application

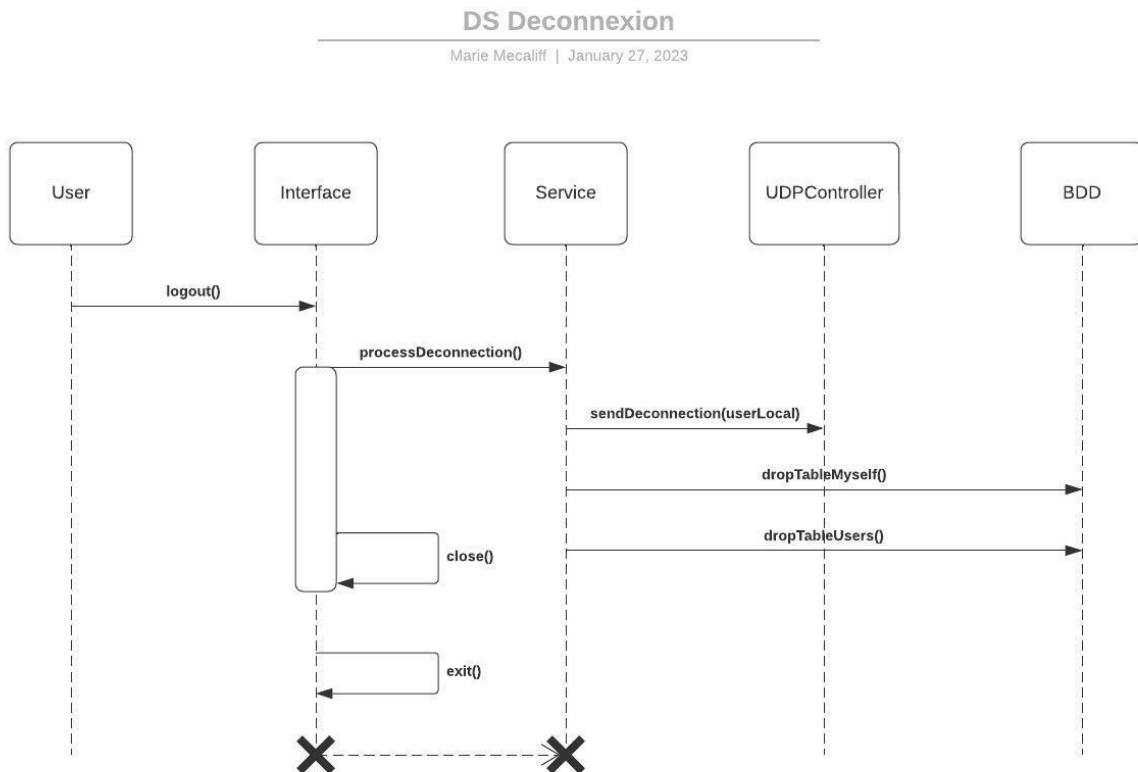


Du côté des utilisateurs distants, à la réception d'un message UDP de type « Connexion », le nouvel utilisateur est ajouté à la liste des utilisateurs connectés.

b) Envoi de messages entre utilisateurs dans une session de clavardage



c) Déconnexion d'un utilisateur à l'application



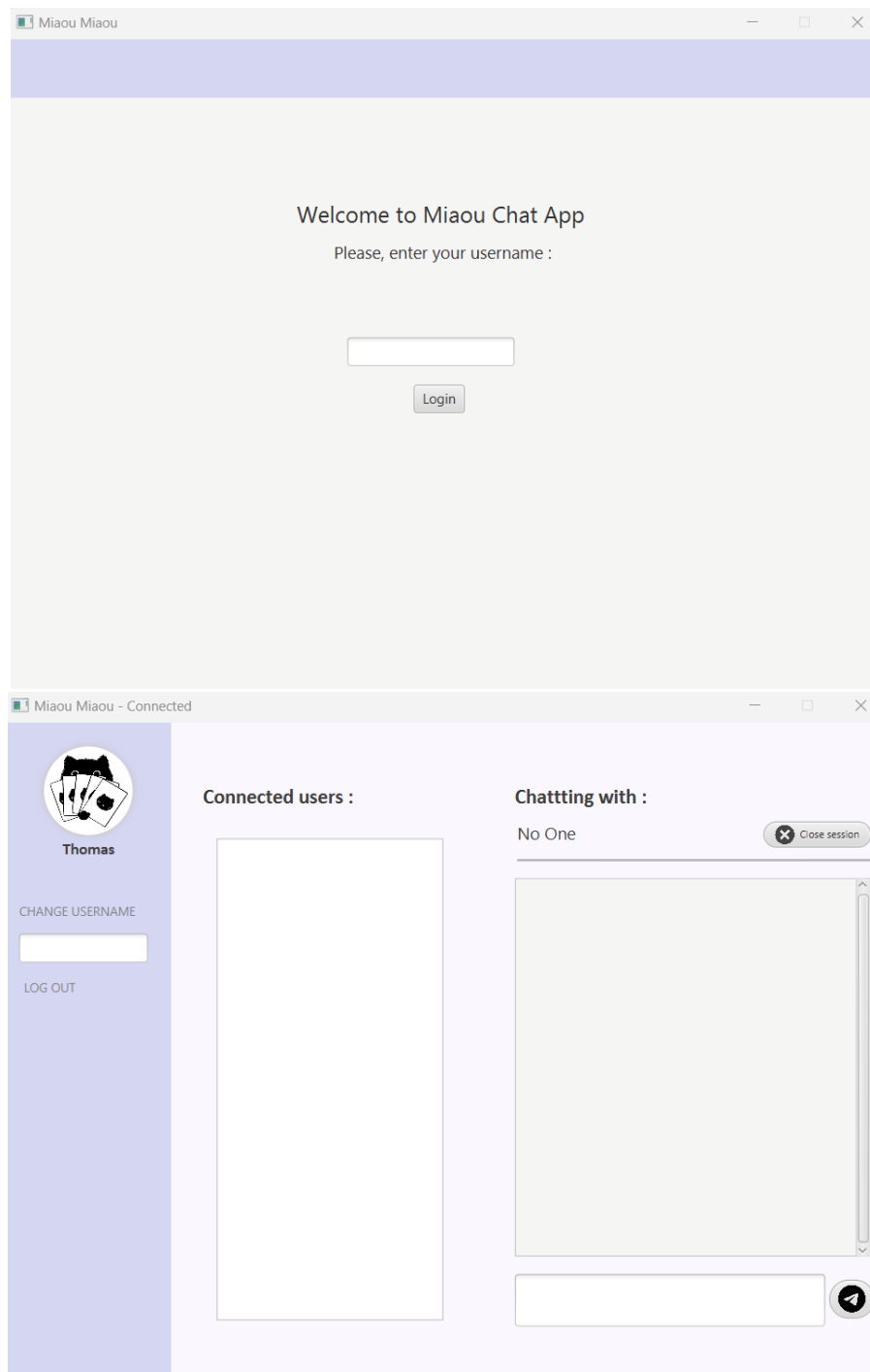
Du côté des utilisateurs distants, à la réception d'un message UDP de type « Déconnexion », le nouvel utilisateur est supprimé de la liste des utilisateurs connectés. De plus, si une session de chat avec cet utilisateur est ouverte sur l'interface, celle-ci est automatiquement fermée.

4. Schéma de conception de la base de données



Comme cela était indiqué dans le cahier des charges, nous avons construit une base de données décentralisée pour permettre et garantir à chaque hôte une indépendance quant à la gestion de ses données. Nous avons construit notre base de données avec trois tables indépendantes. Messagedb permet de conserver l'historique de tous les messages reçus et envoyés par l'hôte. La table ListUsers permet d'avoir dynamiquement les informations nécessaires afin de contacter les autres utilisateurs connectés, et échanger des messages avec eux. Enfin, la table Myself nous permet d'avoir un moyen simple et efficace de conserver le nom d'utilisateur de l'utilisateur local.

5. Interface graphique



Nous avons choisi de mettre en place l'interface à l'aide de JavaFx, par sa simplicité d'utilisation et de conception avec SceneBuilder, pour les nombreuses fonctionnalités d'affichage qu'il propose et pour ses visuels relativement plus modernes.

II. Procédures de tests, d'installation et de déploiement

L'intégration continue a été mise en place à travers Jenkins. Si tout se passe bien au niveau de la compilation, la conception de la plupart nos tests unitaires ne nous permettent pas de les automatiser.

En effet, seuls les tests `ListUserTest` et `DatabaseTest` peuvent être automatisés. La classe de test `UDPTTest` nous a permis de tester nos méthodes du package *udp* au début de la conception de notre application. Pour les tests de `TCPTest` et `ServiceTest`, deux ordinateurs sont nécessaires. Pour chaque paire de test, un ordinateur serveur doit lancer le tests « server » en premier tandis que l'ordinateur client lance le test « client » correspondant en suivant. Ces tests nécessitent également de renseigner en haut des fichiers les adresses IP des ordinateurs, comme indiqué et expliqué sur les fichiers en question. Tous les tests unitaires ont été testés avec succès avant de lancer le développement de l'interface.

Pour ce qui est des procédures de déploiement, nous avons utilisé Maven pour mettre en place les dépendances dans le fichier `pom.xml`. Nous avons ensuite créé le fichier `-jar` exécutable en utilisant les commandes Maven `clean` et `package`. Cependant, nous avons rencontré un problème de reconnaissance de certains modules `javafx` par notre fichier `-jar` que nous ne sommes pas parvenus à résoudre. Ce problème empêche une installation directe en téléchargeant uniquement le fichier `-jar` exécutable puis en l'exécutant sur un terminal. La solution que nous avons mise en place est ainsi de créer un fichier compressé contenant notre exécutable `-jar` ainsi que la bibliothèque de la version de `javafx` utilisée pour notre interface. Pour lancer notre application, il faut ainsi au préalable télécharger ce fichier compressé (lien google drive mis en place) puis le décompresser et enfin, exécuter le fichier `-jar` avec la commande suivante :

```
java -module-path ./lib --add-modules javafx.controls,javafx.fxml -jar projetMiaou.jar
```

III. Manuel d'utilisation simplifié

Lors de l'arrivée sur la page de connexion à l'application de Chat, rentrer un pseudonyme et appuyer sur le bouton login pour confirmer. Si le pseudonyme choisi est invalide ou s'il est déjà pris par un autre utilisateur déjà connecté, un message d'erreur s'affichera. Veuillez alors choisir un autre login, conforme.

Sur la page principale du chat, vous trouverez sur le panneau de gauche votre photo de profil, et en dessous le pseudonyme choisit. Celui-ci se mettra automatiquement à jour si vous décidez de le changer. Pour ce faire, veuillez rentrer votre nouveau pseudonyme dans le champ prévu à cet effet intitulé « Change Username » et appuyez sur entrer. Ce pseudonyme doit toujours être unique et remplir les conditions imposées sur les caractères utilisables.

En dessous du champ pour changer votre pseudonyme, vous trouverez un bouton « Logout » sur lequel cliquer si jamais vous souhaitez vous déconnecter. Un message de confirmation de déconnexion s'affichera alors, il vous faudra appuyer sur « Cancel » si vous souhaitez revenir en arrière.

Au milieu de la page principale, vous trouverez la liste des utilisateurs connectés, qui se met automatique à jour en cas de nouvel connexion, de déconnexion ou de changement de pseudonyme par un de ces utilisateurs. Pour démarrer une session de clavardage avec l'un d'eux, il vous faudra cliquer sur son pseudonyme dans la liste.

Vous trouverez à gauche la fenêtre de chat. Le pseudonyme de la personne avec laquelle vous êtes en train d'échanger s'affiche au-dessus et se met à jour si celui-ci vient à en changer. Pour envoyer un message, veuillez taper le contenu du message que vous souhaitez envoyer dans le champ prévu à cet effet en bas de la fenêtre, et appuyez sur le bouton « Send » à droite du champ ou bien sur la touche entrée de votre clavier pour déclencher l'envoi. Les messages que vous aurez envoyés s'afficheront à droite de la fenêtre et ceux envoyés par l'utilisateur avec qui vous conversez s'afficheront sur la gauche. Si tous les messages ne s'affichent pas dans la fenêtre, vous pouvez les faire défiler à l'aide de la souris pour afficher les plus anciens. Enfin, pour fermer une session de chat, vous pouvez simplement cliquer sur le bouton « Close session » situé en haut à droite.