

# Projet Clavardage

## Tp POO

Redigé par :  
Le Nezet Kilian

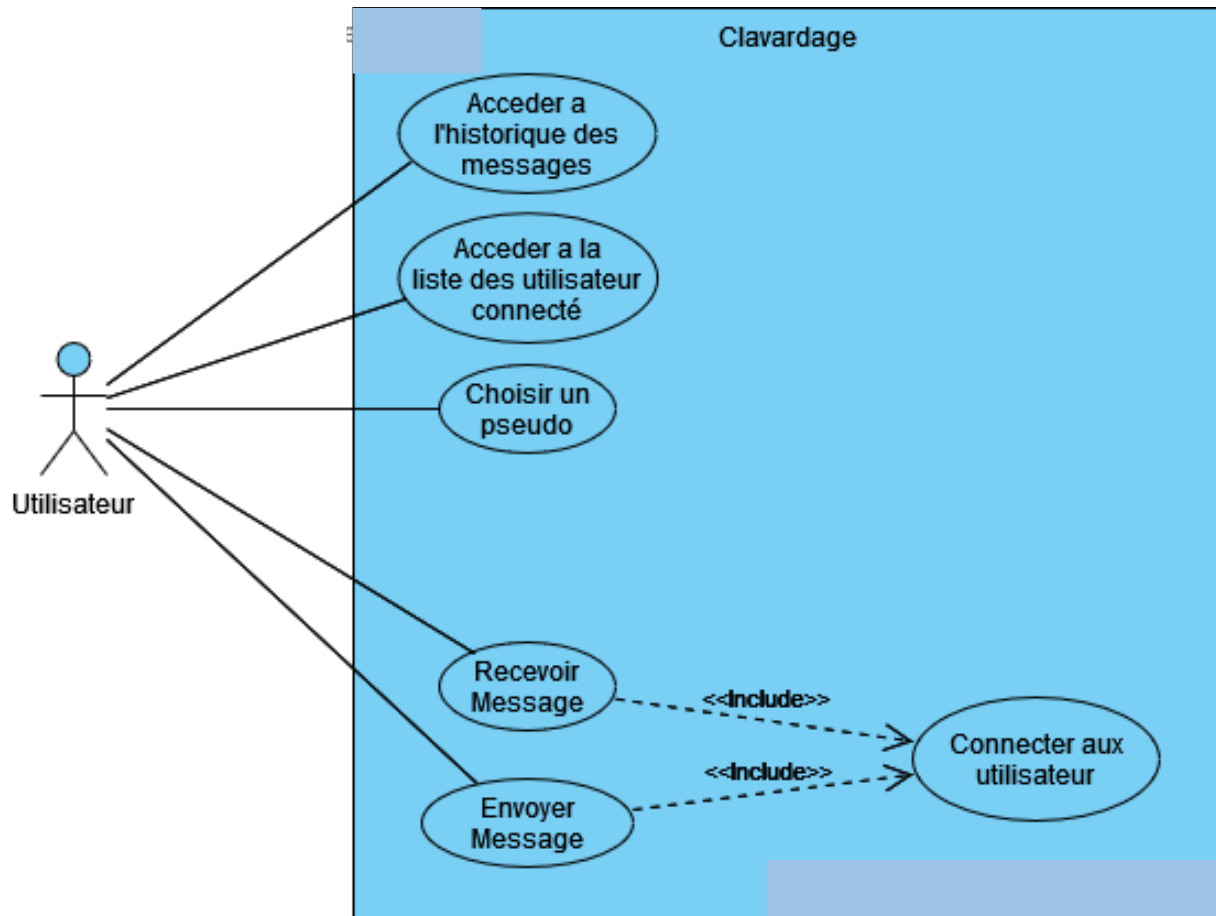
- Vendredi 28 janvier 2022 -

# Table des matières

<b>UML .....</b>	<b>3</b>
<b>Diagramme d'état :.....</b>	<b>3</b>
<b>Diagramme de Classe :.....</b>	<b>4</b>
Les trois classes principales :.....	4
Les trois Packages :.....	4
<b>Diagramme de séquence : .....</b>	<b>5</b>
<b>Choix technologique et de développements : .....</b>	<b>6</b>
<b>Base de données :.....</b>	<b>6</b>
<b>GUI : .....</b>	<b>6</b>
<b>Procédure de test :.....</b>	<b>6</b>
<b>GitHub Actions :.....</b>	<b>6</b>
<b>Jira :.....</b>	<b>6</b>
<b>Manuel d'utilisation : .....</b>	<b>7</b>
<b>Installation :.....</b>	<b>7</b>
<b>Connection et discussion : .....</b>	<b>7</b>

# UML

## Diagramme d'état :



## Diagramme de Classe :

Vous trouverez les diagrammes des classes complet sur le GitHub.  
Le Système est composé de 3 classes principales et de 3 packages.

Les trois classes principales :

La première classe principale est le Main servant juste à appeler la deuxième classe principale « App ». Cette dernière va lancer les différentes classes servant à la communication et la classe servant au lancement du GUI se nommant « ClavardeurWindow »

Les trois Packages :

Le premier package « TCP\_Message » sert à la communication en TCP pour gérer les messages entre les différents utilisateurs. On va utiliser une classe « messageTCP » pour faire un objet message.

La classe « ServerSideMessage » va gérer la connexion aux autres utilisateurs sur le réseau et va créer une connexion avec la classe « ClientSide » pour chaque utilisateur.

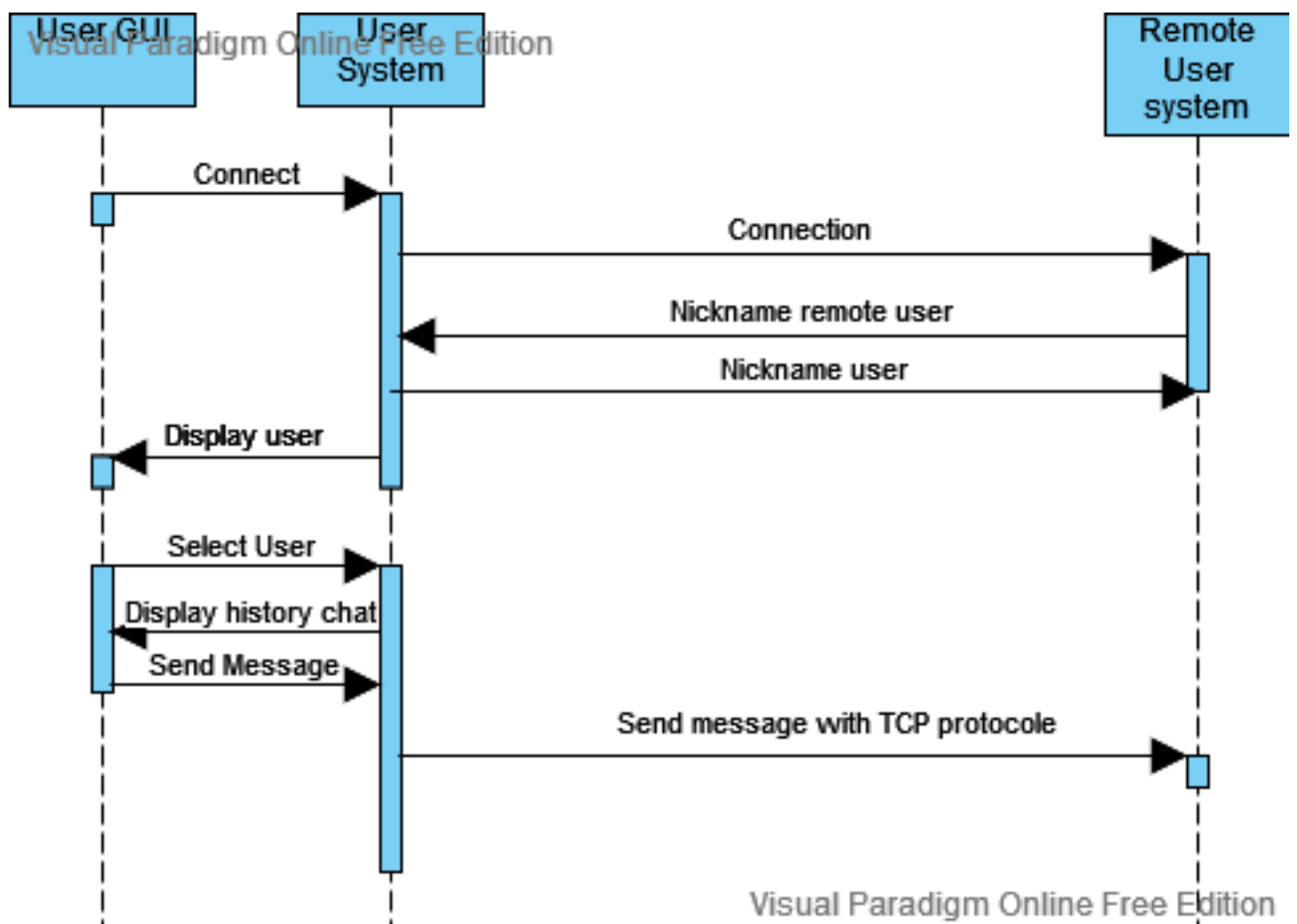
Le deuxième Package « Communication » sert à la communication en UDP pour gérer les connexions et la gestion des pseudonyme entre les différents utilisateurs. Il existe 4 classes qui hérite de la classe « TypeOfMessage ».

Chaque héritage correspond à une action à effectuer.

La classe « CommunicationManager » va gérer l'envoi et la réception des messages entre les différents clients

Enfin le troisième package « BDD » sert à la création et la gestion de base de données, cette BDD va sauvegarder l'historique de toutes les communications et va permettre de l'afficher sur l'interface graphique.

## Diagramme de séquence :



## Choix technologique et de développements :

### Base de données :

La base de données utilisé est locale, l'application utilise SQLite. La base données stock l'historique des messages avec le destinataire, l'émetteur et la date d'envois.

### GUI :

L'interface graphique a été créer grâce a l'éditeur de GUI intégré dans IntelliJ qui repose sur SWING.

Pour l'affiche des utilisateurs connecté l'application utilise une JList pour que l'affiche se mette à jour automatiquement.

Pour l'affichage de l'historique des messages l'application utilise un JTextArea qui vas se mettre à jour toute les 0,5 seconde.

### Procédure de test :

Tous les tests ont été fais en condition réel grâce a un autre ordinateur sur un réseau local.

### GitHub Actions :

Lien : [https://github.com/NailikLN/Clavardage\\_POO](https://github.com/NailikLN/Clavardage_POO)

Pour le déploiement en continue un workflow GitHub a été crée afin de publier une release sur GitHub dès que le mot clé [release] est insérée dans le message de push.

### Jira :

Lien : <https://lenezet.atlassian.net/jira/software/projects/CLAV/boards/1>

Pour faire un suivi de projet j'ai utilisé Jira et créer un tableau et des tickets représentant chaque étape de projet. Etant donnée que j'ai fais le projets seul je n'ai pas noter de sprint sur Jira. Cependant j'ai fait l'équivalent de 3 sprints de deux semaines. Un pour l'UDP, 1 pour le TCP et 1 pour le GUI.

# Manuel d'utilisation :

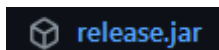
## Installation :

Pour installer le logiciel il faut vous rendre sur la page GitHub suivante :  
[https://github.com/NailikLN/Clavardage\\_POO](https://github.com/NailikLN/Clavardage_POO)

Ensuite cliquer sur la dernière Release présentée à droite de l'écran :



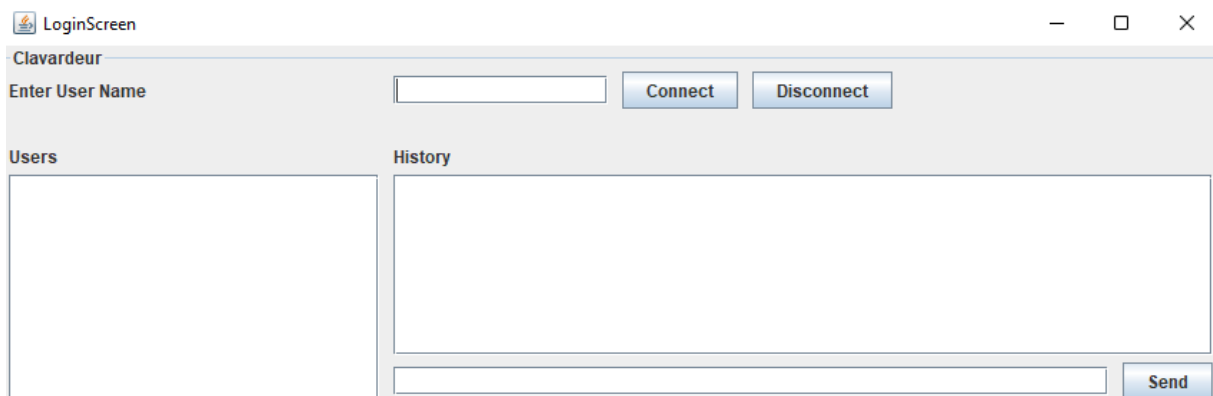
Ensuite sélectionné le fichier release.jar :



Téléchargez-le et exécutez-le (attention vous devez avoir Java 17 installé)

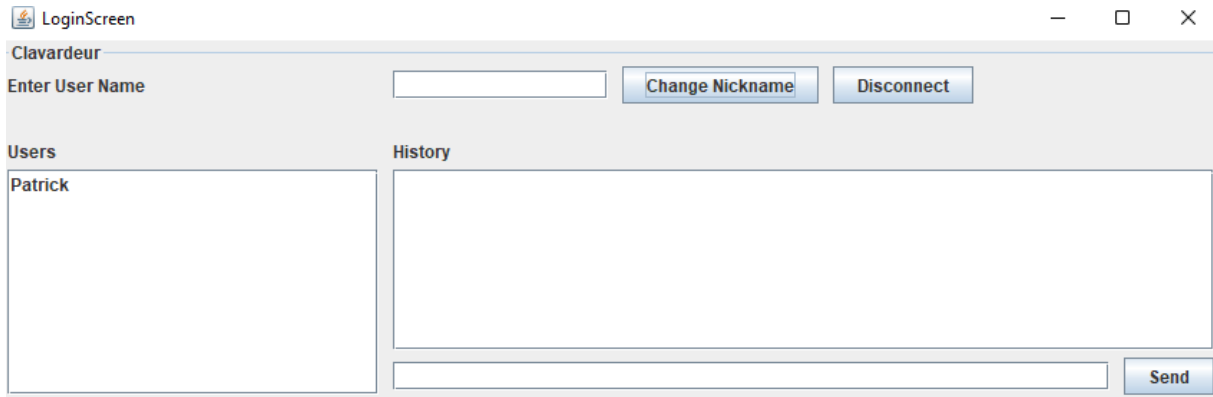
## Connection et discussion :

Lorsque la fenêtre s'affiche vous devriez avoir un écran similaire a ceci :

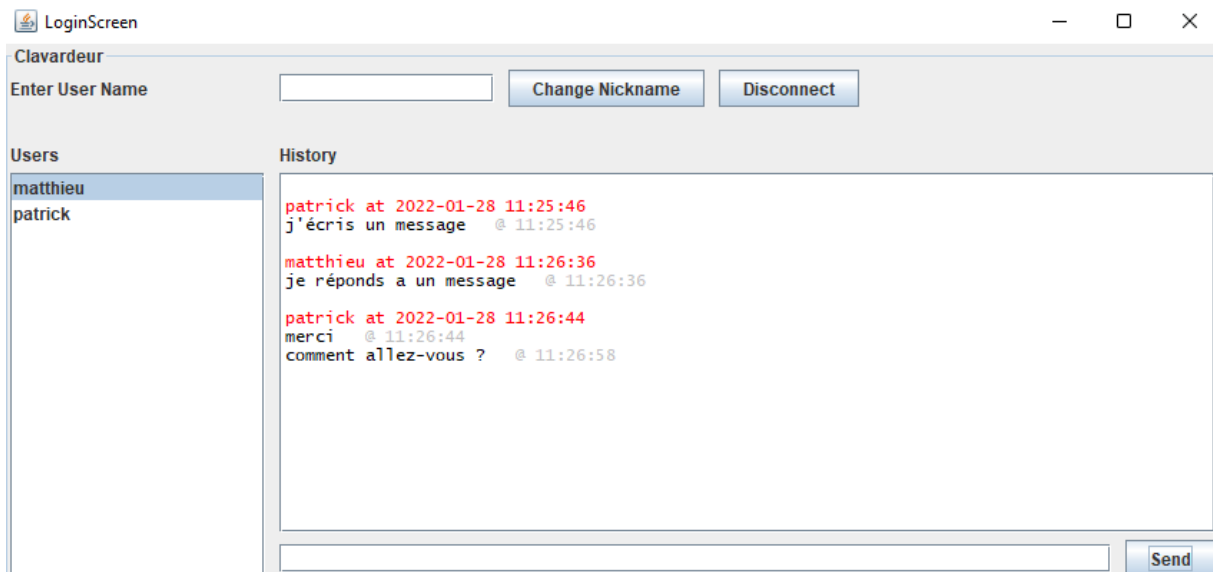


Entrez votre pseudo dans la barre de texte tout en haut et cliquez sur Connect

Le bouton devrait se changer en « Change Nickname » et votre pseudo devrais s'afficher à gauche comme ceci :



Quand une autre personne se connecte son pseudo va s'afficher a gauche. Si vous cliquez dessus l'historique vas s'afficher a droite.



Pour envoyer un message sélectionné le destinataire sur la liste de gauche et écrivez votre message dans la barre de texte en bas a droite et cliquez « send »

Quand vous avez finis cliquez sur « Disconnect » ou fermez l'application.