

Système de clavardage temps réel

NEDELLEC Maël - CROS Alexandre

Déploiement et mise en place de l'application	1
Utilisation par un client	2
Connexion :	2
Choix du pseudo :	3
Pseudo déjà utilisé :	3
Fenêtre principale :	3
Ouverture de session :	4
Envoi de message :	4
Réception de message :	5
Fermeture de session :	5
Tests et validation	5
Test de connexion avec un login/password erroné :	5
Test de choix d'un pseudo déjà connecté :	6
Test d'ouverture de session et restauration d'historique :	6
Test d'envoi/réception d'un message :	7
Test de fermeture de session :	7
Test de déconnexion :	7
Choix techniques	7
Unicité du pseudo :	7
Gestion de l'historique :	8
Base de données :	8
Améliorations possibles	8

Déploiement et mise en place de l'application

Pour compiler le projet en une archive JAR, il est nécessaire d'installer Ant avec la commande suivante sous Linux : `sudo apt install ant`.

Il faut ensuite exécuter le script de compilation :

```
ant -buildfile ../coo/ChatSystemProject2/build.xml.
```

Pour lancer l'application, il faut maintenant exécuter l'archive avec la commande :

```
java -jar ../coo/ChatSystemProject2/dist/ChatSystemProject2.jar.
```

Utilisation par un client

Connexion :

Une fois l'application lancée, l'utilisateur voit une page de connexion :



La base de donnée est propre à chaque machine et est déjà peuplée de quelques valeurs afin de tester le système.

Nous n'avons implémenté aucune solution de sécurité, ainsi tous les login existant sont inscrits dans la base.

A ce stade là il existe donc trois login différents :

Login	Password
alex	alex
mael	mael

cou	cou
-----	-----

Choix du pseudo :

Une fois l'utilisateur connecté, il doit choisir un pseudo de son choix :



please choose a nickname
Be careful it will identify you on the network

pseudo :

Alexandre Cros - Nedellec Maël © 2019-2020

Pseudo déjà utilisé :

Si l'utilisateur choisi un pseudo déjà existant et connecté au sein du réseau, alors la fenêtre suivante apparaît :



Input

?

Please enter your new pseudo

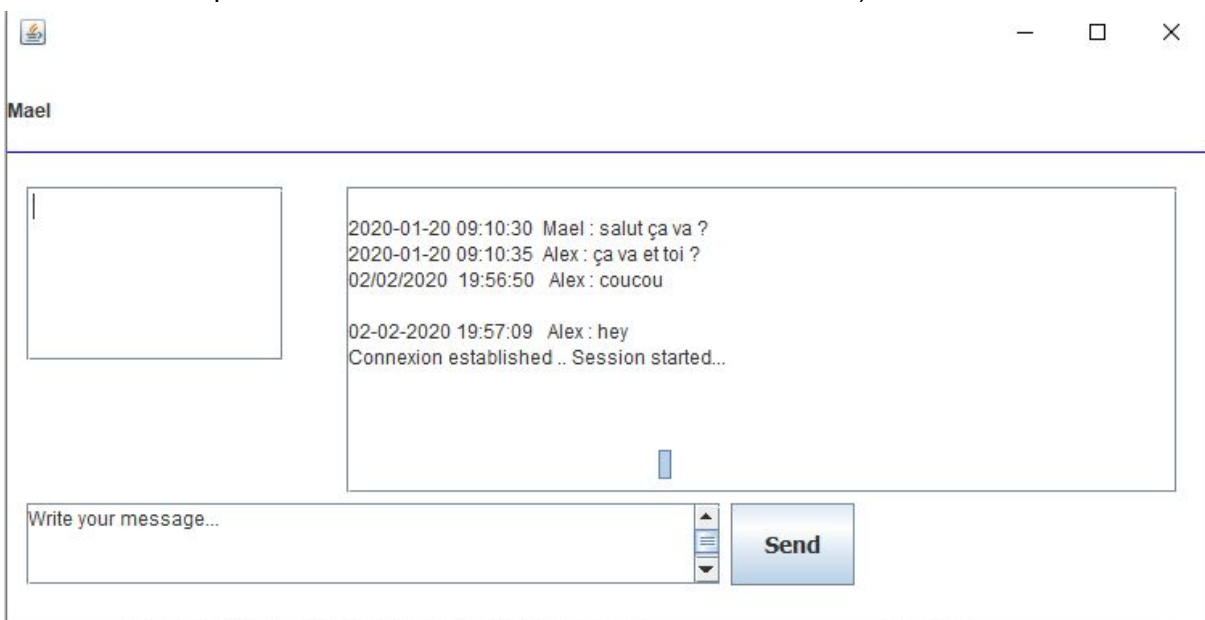
Fenêtre principale :

Si le pseudo est unique, alors l'utilisateur accède ensuite à la fenêtre principale de l'activation où il peut lire son pseudo et voir les utilisateurs connectés sur le réseaux :



Ouverture de session :

Pour ouvrir une nouvelle session avec un utilisateur connecté, l'utilisateur principal doit double-cliquer sur le destinataire de son choix, puis la fenêtre suivante s'ouvrira (si il existe un historique entre les deux utilisateurs, alors il est restauré) :

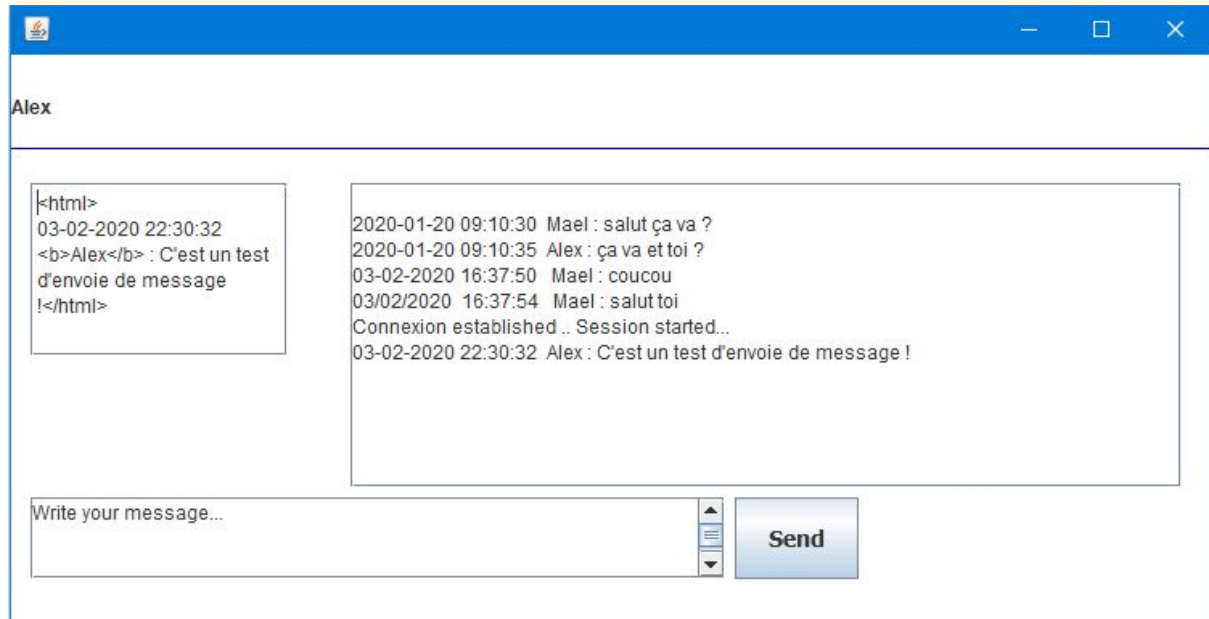


Envoi de message :

Pour un envoyer un message, il suffit à l'utilisateur d'écrire son contenu dans la case dédiée sur la session voulue, puis de l'envoyer. Il est également possible d'envoyer des fichiers ou format *pdf* ou *png*.

Réception de message :

Si l'utilisateur reçoit un message, une fenêtre de session s'ouvre et affiche l'historique (si il existe) et le nouveau message :



Fermeture de session :

Pour fermer une session, il suffit à l'utilisateur de cliquer sur le bouton de fermeture de fenêtre.

Dans cette version, un bug implique que fermer la session crée une exception et stoppe la totalité de l'application.

Tests et validation

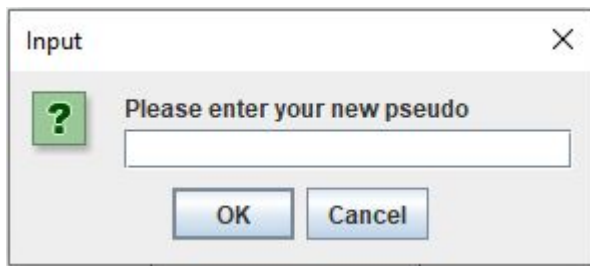
Pour les tests, "Alex" sera connecté sur la machine A, et "Mael" sur la machine B.

Test de connexion avec un login/password erroné :



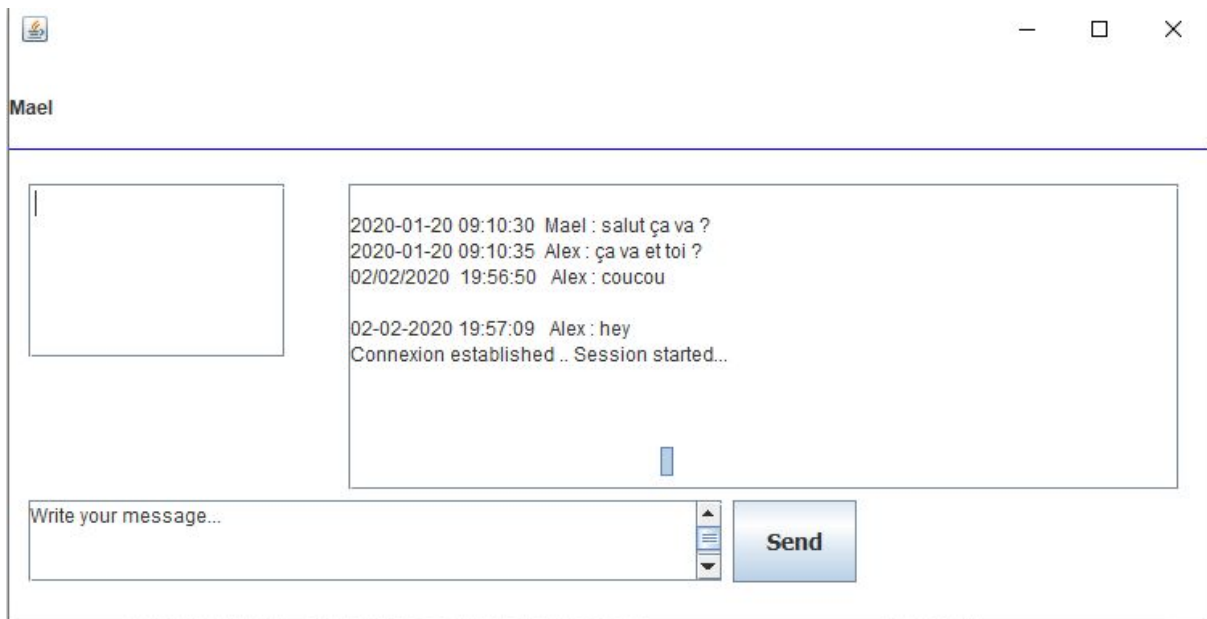
Avec le login "alex" et password "mael", un message d'erreur s'affiche puisque la combinaison n'existe pas.

Test de choix d'un pseudo déjà connecté :



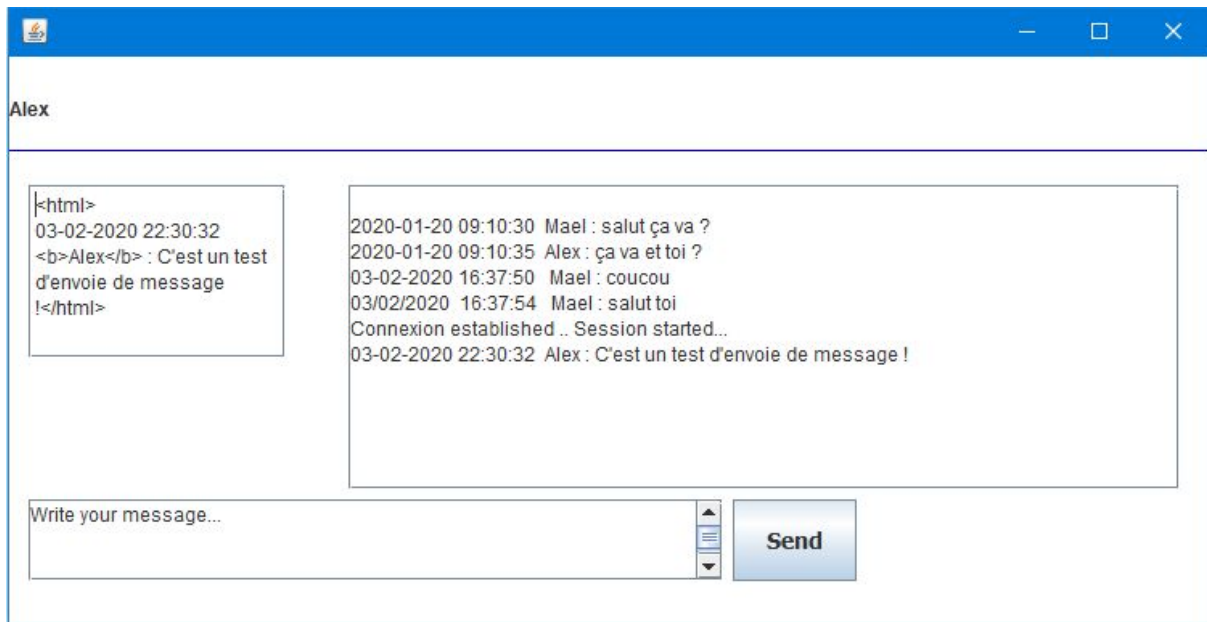
Si “Mael” est déjà en ligne, alors le choix du pseudo “Mael” impliquera l’ouverture de cette fenêtre bloquante.

Test d’ouverture de session et restauration d’historique :



En double-cliquant sur un utilisateur, la session s’ouvre, l’historique correspondant est bien restauré.

Test d'envoi/réception d'un message :



Le message envoyé depuis le machine A ("Alex") est bien reçu sur la machine B ("Mael").

Test de fermeture de session :

```
java.net.SocketException: Connection reset
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at sun.nio.cs.StreamDecoder.readBytes(Unknown Source)
    at sun.nio.cs.StreamDecoder.implRead(Unknown Source)
    at sun.nio.cs.StreamDecoder.read(Unknown Source)
    at java.io.InputStreamReader.read(Unknown Source)
    at java.io.BufferedReader.fill(Unknown Source)
    at java.io.BufferedReader.readLine(Unknown Source)
    at java.io.BufferedReader.readLine(Unknown Source)
    at chatsystemproject.ServerThread.run(ServerThread.java:75)
    at java.lang.Thread.run(Unknown Source)
```

Lorsque l'on ferme une session entre A et B sur la machine B, elle se ferme sans soucis. En revanche sur la machine A, la session se ferme mais l'erreur ci-dessus est retournée.

Test de déconnexion :

La déconnexion n'est pas complètement implémentée et donc non fonctionnelle. Lorsqu'un utilisateur se déconnecte les autres ne reçoivent pas la mise à jour de leur liste.

Choix techniques

Unicité du pseudo :

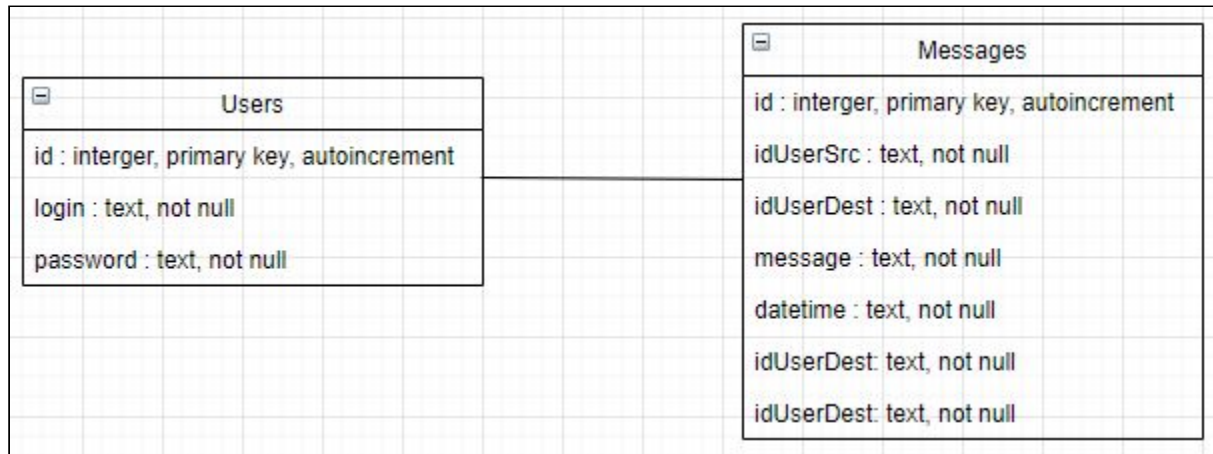
Afin de gérer l'unicité du pseudo, chaque nouvel utilisateur broadcast son pseudo. Ainsi si un autre utilisateur possède le même pseudo, il lui retourne un message d'alerte, ce qui oblige l'obligé à choisir un nouveau pseudo.

Gestion de l'historique :

Afin que l'historique puisse être géré peu importe le pseudo choisi par les utilisateurs, l'identifiant utilisé est le login des utilisateurs. Ainsi, à la connexion, l'utilisateur broadcast, en plus de son pseudo, son login, ce qui permet aux autres de l'enregistrer, et de restaurer l'historique si besoin est.

Base de données :

La gestion des données est implémentée par une base de donnée SQLite.
La base de données est composée ainsi :



Améliorations possibles

Certaines améliorations sont à effectuer pour correspondre totalement au cahier des charges :

- Le système ne fonctionne que sur un réseau local
- On ne peut pas rajouter un login/password autrement qu'en dur dans la BDD
- Le niveau de sécurité est très faible quant à la base de données
- On ne peut pas choisir l'état des utilisateurs (hors ligne, indisponible, etc)
- Un bug créé une erreur lorsque l'on ferme une fenêtre de session
- La déconnexion n'est pas complètement implémentée