

# Application de clavardage

Poulin Lancelot et Aldebert Lucie  
4IR groupe B1



04/02/2020

# Sommaire

I) Déploiement du système.....	p 2
II) Manuel d'utilisation de l'application de clavardage.....	p 2
1) Connexion	
2) Gérer une session avec un autre utilisateur	
3) Envoyer un message	
4) Envoyer un document	
5) L'historique des messages	
6) Changer de pseudonyme	
7) Se déconnecter	
III) Conception et choix des technologies.....	p 6
IV) Procédures de test et de validation.....	p 8

DISCLAIMER : Le système développé considère qu'un utilisateur utilise un unique ordinateur et par conséquent est identifié à une unique adresse MAC.

## I) Déploiement du système

### 1) Web Service

Le fichier WAR est présent sur le GitHub dans "POO/WebServer/" sous le nom de "WebServer.jar" et doit être déployé sur un serveur Tomcat (testé sur la version 8.5) en local (le serveur du gei n'étant pas atteignable facilement depuis l'extérieur). Alternativement, le projet Eclipse IDE WebServer peut être ouvert puis lancé en configurant le serveur tomcat.

### 2) Application

Note: Si vous souhaitez déplacer le JAR, l'application utilise deux répertoires "Files" et "Histories" qui doivent se trouver dans le même répertoire.

Deux options :

- Utiliser l'application normalement (dans son but final) et donc sur plusieurs ordinateurs : dans le répertoire "POO/ProjetChatSystem/", lancer la commande "java -jar ProjectChatSystem\_final.jar".
- Utiliser l'application sur UN SEUL ordinateur, et donc pouvoir tester l'application avec deux fenêtres MAXIMUM : dans le répertoire "POO/ProjetChatSystem/" lancer la commande "java -jar ProjectChatSystem\_local.jar 5000 5001" pour la première fenêtre puis "java -jar ProjectChatSystem\_local.jar 5001 5000" pour la deuxième en faisant attention d'inverser le numéro de port entre les deux pour les faire concorder. Ce JAR enlève quelques conditions dans le code pour permettre le test entre deux applications sur un même ordinateur.

## II) Manuel d'utilisation de l'application de clavardage

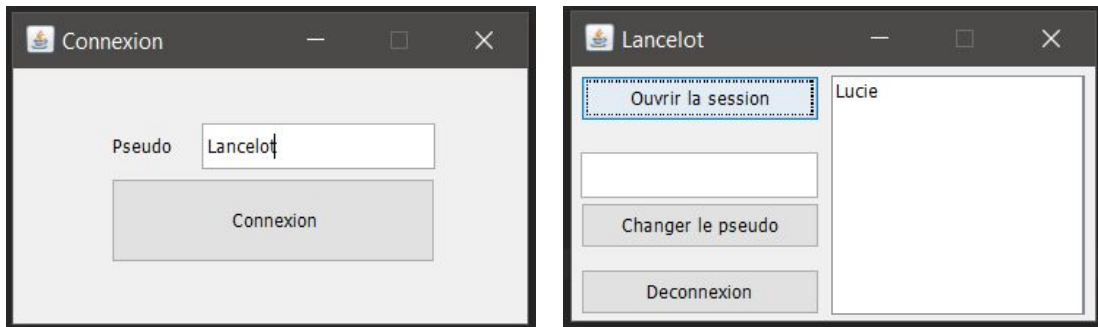
### 1) Connexion

Une fois l'application lancé, la fenêtre de connexion apparaît. Elle est composée de d'un champ de texte et d'un bouton labellisé "connexion". Chaque compte sur l'application est relié à un ordinateur.

Entrer un pseudonyme\* dans le champ de texte et appuyer sur le bouton. Si ce pseudonyme est disponible\*\* la fenêtre principale apparaît, vous êtes connecté sinon rentrez un autre pseudonyme.

\*disponible: signifie qu'aucun utilisateur de l'application est connecté avec ce pseudonyme

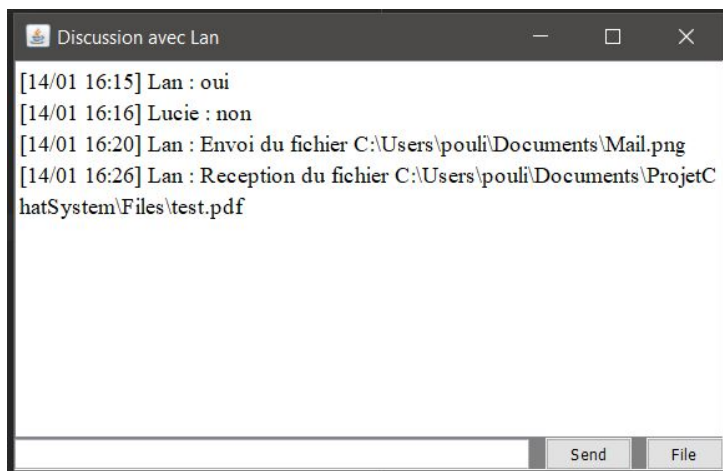
\*\*pseudonyme: nom sous lequel vous apparaîtrez auprès des autres utilisateurs



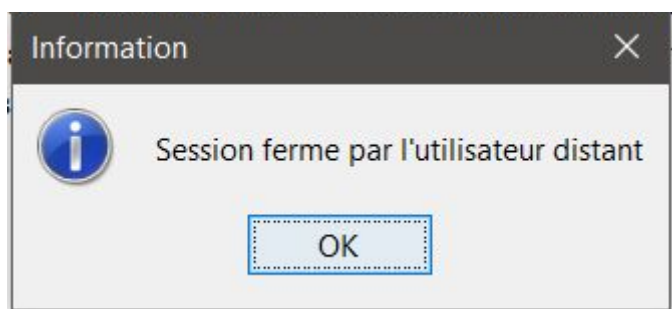
## 2) Gérer une session avec un autre utilisateur

Prés requis: être connecté voir 1) Connexion

Pour démarrer une session, sur la fenêtre principale sélectionner un utilisateur actif dans la liste des utilisateurs actifs puis appuyer sur le bouton labéliser “Ouvrir la session”. Pour réduire la session réduire la fenêtre et pour fermer la session fermer la fenêtre.



A la fermeture d'une session par l'un des utilisateurs, l'autre est averti et voit sa session se fermer aussi.



## 3) Envoyer un message

Prés requis: être connecté voir 1) Connexion et avoir démarrer une session voir 2) Gérer une session avec un autre utilisateur

Sur une fenêtre de session ouverte, rentrer le message dans le champs de texte en bas de la fenêtre et appuyer sur le bouton labélisé “Envoyer”. Le message est envoyé. Il apparaît dans l'historique de la conversation dans la zone de texte de la session ouverte.

Certain symboles ne peuvent pas être utilisés dans les messages. Les symboles interdits sont [ , ] et /.

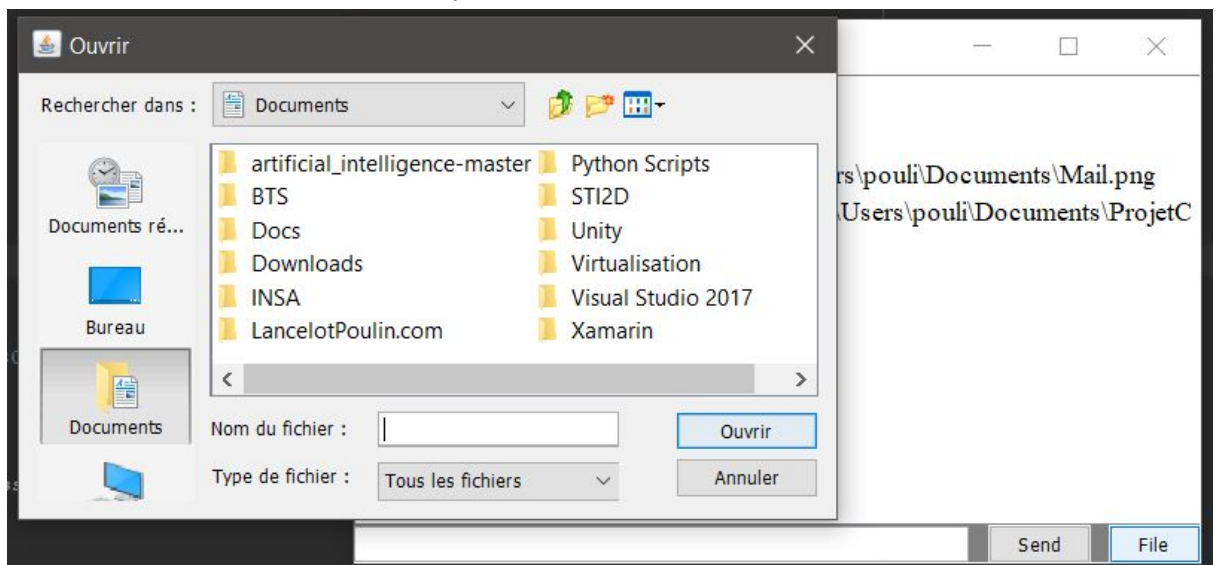


#### 4) Envoyer un document

Prés requis: être connecté voir 1) Connexion et avoir démarré une session voir 2) Gérer une session avec un autre utilisateur

Sur une fenêtre de session ouverte, appuyez sur le bouton labellisé “Fichier”. Une fenêtre s’ouvre, sélectionnez le fichier souhaité et appuyez sur le bouton “ok”. Le fichier est envoyé. Le fichier se trouve dans le répertoire /Files de l’application. Dans l’historique on peut voir apparaître le chemin du fichier que l’on envoie.

La taille maximale du fichier envoyé est 100Ko.



#### 5) L'historique des messages

Prés requis: être connecté voir 1) Connexion et avoir démarré une session voir 2) Gérer une session avec un autre utilisateur

L'historique est visible sur la fenêtre de session automatiquement, si aucun historique existe pour l'utilisateur (@MAC) distant, un message s'affiche l'indiquant.

Le format de l'historique est le suivant: [date][adresse mac] message|fichier

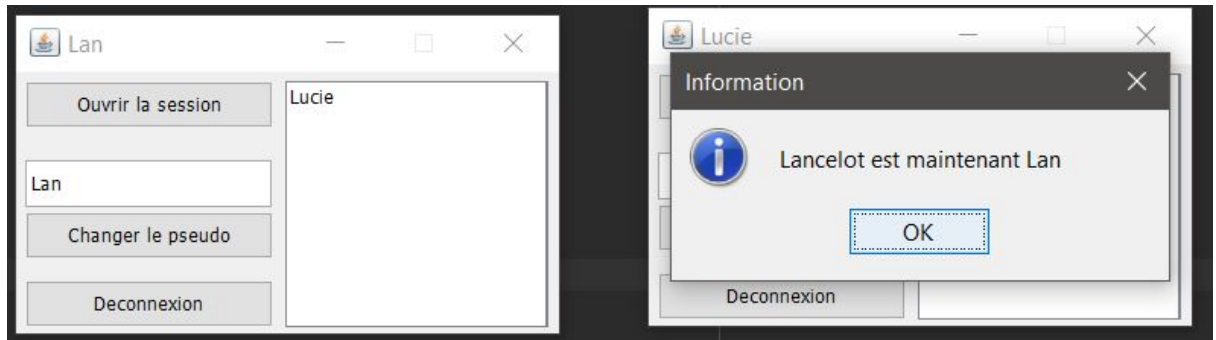
Voici l'historique des messages vu plus haut. (dans /Histories)



#### 6) Changer de pseudonyme

Prés requis: être connecté voir 1) Connexion

Sur la fenêtre principale, composé d'un premier champ de texte et d'un bouton labellisé "Changer de nom d'utilisateur". Pour changer de nom d'utilisateur: entrez le nouveau pseudonyme souhaité dans le champs de texte et appuyez sur le bouton "Changer de nom d'utilisateur". Si celui ci est disponible vous changez de nom d'utilisateur, sinon un message s'affiche.

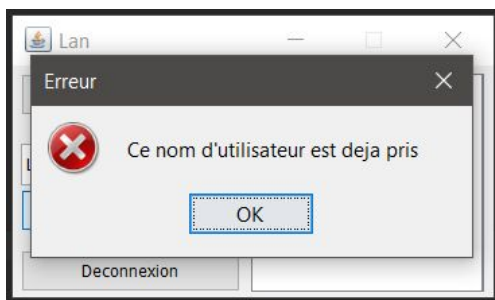


#### 7) Se déconnecter

Prés requis: être connecté voir 1) Connexion

Sur la fenêtre principale, apparaît un bouton labellisé "Déconnexion". Pour se déconnecter appuyez sur le bouton "Déconnexion". Vous êtes redirigé vers la fenêtre de connection.

La quasi totalité des erreurs (username, UDP, TCP, HTTP, sauvegarde historique, etc) sont gérées et affichées au niveau graphique par le billet de message pop-up :



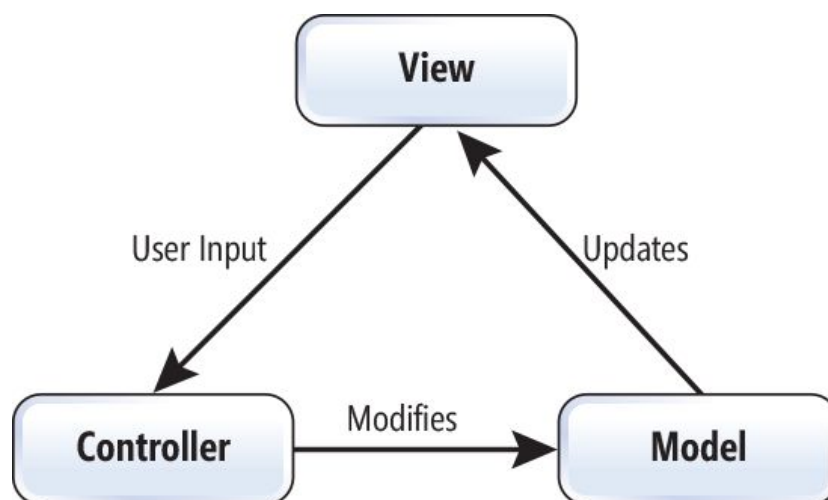
### III) Conception et choix des technologies

Nous avons décidé de développer l'application en suivant le pattern MVC (Model-View-Controller).

On retrouve les Views qui sont les parties graphiques du projet : page de connexion, page principale et page de session (chat, liste, boutons).

On a aussi les Controllers associés à chaque Views permettant la relation entre cette dernière et les modèles d'information en les avertissant des événements utilisateurs (comme un clic sur un bouton).

Ces modèles d'information (Models) contiennent le corps du programme en intégrant les fonctionnalités principales de l'application : la gestion des utilisateurs distants, la communication avec le réseau, la gestion des sessions de chat. Les Models affichent ensuite leurs informations en mettant à jour les parties graphiques qui leur sont associées. Cela forme une boucle.



Nous avons choisis d'utiliser Java Swing pour développer la partie graphique de l'application pour plus de facilité et ainsi obtenir un design simple.

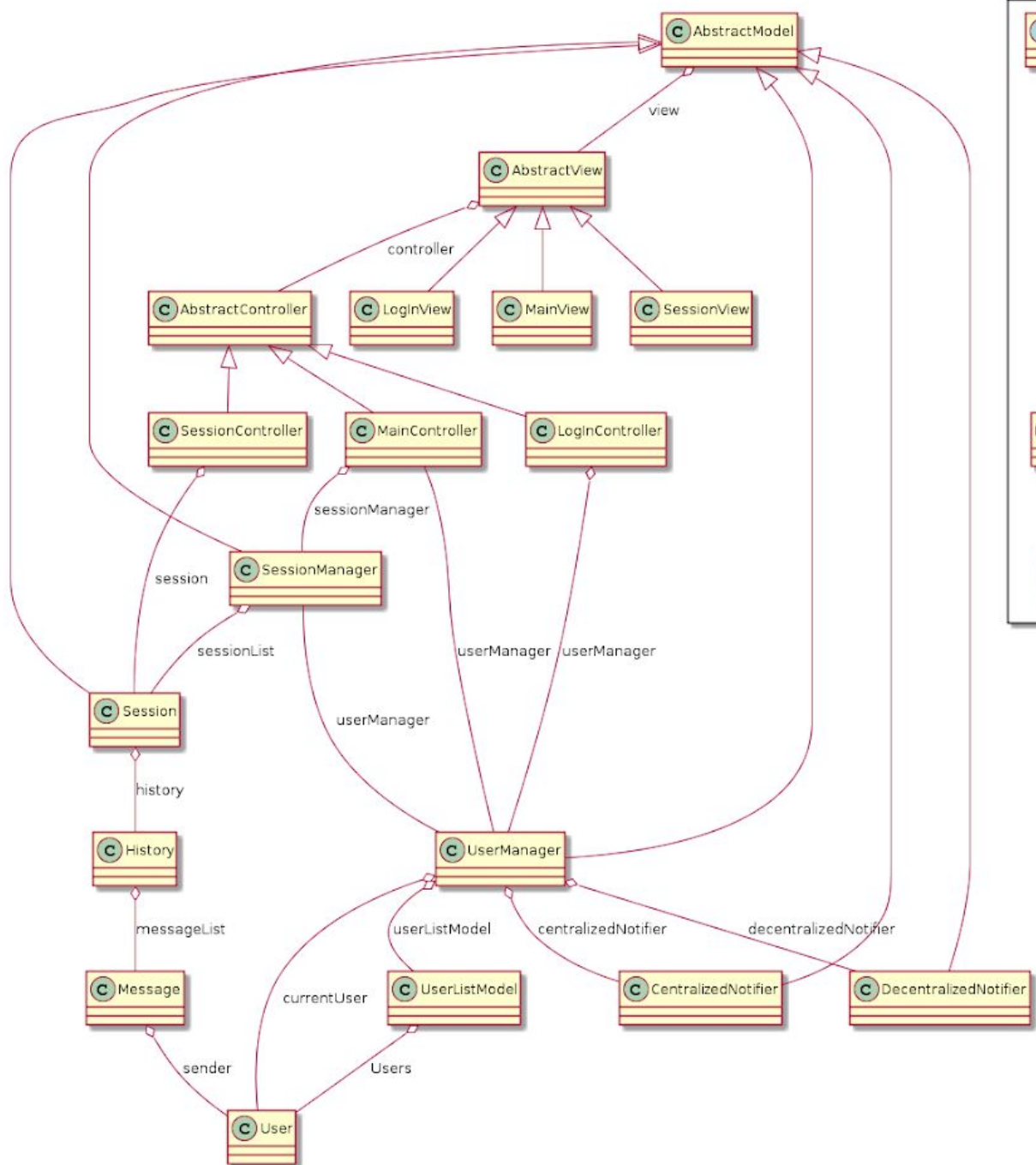
Pour les historiques entre utilisateurs nous avons décidé de les stocker en local dans des fichiers textes associés à une adresse MAC distant.

L'application utilise le protocole UDP en broadcast et unicast pour les notifications sur le réseau local à l'entreprise (décentralisé) : connexion, déconnexion, changement de nom d'utilisateur, réponses précisant l'adresse IP.

Pour les communications entre deux utilisateurs (soit deux IP), le protocole TCP est utilisé afin de fiabiliser les échanges. Ces derniers peuvent être des messages textes ou des fichiers.

Pour le côté centralisé, qui permet à un utilisateur extérieur à l'entreprise de se connecter à l'application, le Web-service est basé sur le modèle REST, utilisant un servlet Java où l'application effectue des requêtes Http GET indiquant si l'utilisateur se connecte, change de nom d'utilisateur, se déconnecte ou souhaite obtenir la liste des utilisateurs actifs avec leur IP.

Diagramme de classe de l'application :



Note :

DecentralizedNotifier → UDP

CentralizedNotifier → HTTP

SessionManager → TCP Server

Session → TCP Client



## IV Procédures de test et de validation

scénario	résultat	réussi/échoué
Se connecter avec un pseudonyme disponible	-connecté, affiche la fenêtre principale dont la liste des utilisateurs actifs -apparaît dans la liste des utilisateurs actifs des autres utilisateurs	réussi
Se connecter avec un pseudonyme non disponible	-non connecté -message d'erreur "ce nom d'utilisateur est déjà pris "	réussi
Changer de pseudonyme avec un disponible	-le pseudonyme est changé - le nouveau pseudonyme apparaît dans la liste des utilisateurs actifs des autres utilisateurs -un message prévient les autres utilisateurs connectés	réussi
Changer de pseudonyme avec un non disponible	-le pseudonyme reste inchangé -message d'erreur "ce nom d'utilisateur est déjà pris "	réussi
Ouvrir une session avec un utilisateur actif	- une fenêtre de session s'ouvre	réussi
Envoyer un fichier	-le fichier est envoyé et enregistré dans le dossier /Files de l'application sur l'ordinateur de celui qui le reçoit -le fichier apparaît dans l'historique de celui qui envoie et reçoit le "message" est horodaté	réussi
Envoyer un message	-le message est envoyé et reçu -le message apparaît dans l'historique de celui qui envoie et reçoit le message est horodaté	réussi

Fermer la fenêtre d'une session	-La fenêtre se ferme pour l'utilisateur -Son correspondant voit que la session à été fermé par l'utilisateur et sa session se ferme -l'historique de la conversation est sauvegardé des deux côtés	réussi
Ouvrir une session avec un utilisateur avec lequel on a déjà échangé	-la session s'ouvre des deux côtés - l'historique de la conversation est chargé des deux côtés	réussi
ouvrir une session alors que la session est déjà ouverte	-rien	réussi
Un utilisateur se déconnecte	-toutes ses sessions ouvertes se ferment des deux côtés -l'historique des messages est sauvegardé - son nom est retiré de la liste des utilisateurs actifs chez les utilisateurs connectés	réussi

On peut tester le Web service directement sur navigateur avec les paramètres GET:

- Connexion (la réponse contient "OK" ou "NOT OK" si le nom n'est pas déjà prit)

["http://localhost:8080/WebServer/request?Type=Connection&Username=Laaaaancelot&MacAddress=94-E9-79-54-8A-3F&IpAddress=192.168.1.11"](http://localhost:8080/WebServer/request?Type=Connection&Username=Laaaaancelot&MacAddress=94-E9-79-54-8A-3F&IpAddress=192.168.1.11)

- Refresh (retourne la liste des utilisateurs, pas de sérialisation d'objet Json) et change le pseudo si changé:

<http://localhost:8080/WebServer/request?Type=Refresh&Username=Lancelot&MacAddress=94-E9-79-54-8A-3F&IpAddress=192.168.1.11>

```
Lancelot | 94-E9-79-54-8A-3F | 192.168.1.11
Toto | 94-E9-79-54-0A-3F | 192.168.1.12
LeJ | 94-E9-79-54-0A-3D | 192.168.1.13
```

- Déconnexion

<http://localhost:8080/WebServer/request?Type=Connection&Username=Lancelot&MacAddress=94-E9-79-54-8A-3F&IpAddress=192.168.1.11>