

# **COO/POO - Projet clavardage**

**Rapport final**

DARTHENUCQ Sacha, DE CADOINE DE GABRIAC Elie, STARCK Paul (4IR-A2)

le 27/01/2023

<b>I. Introduction</b>	<b>3</b>
<b>II. Conception du projet</b>	<b>3</b>
II.1. Diagramme des cas d'utilisation	3
II.2. Diagramme des classes	3
II.3. Diagrammes de séquence	5
II.4. Implémentation de la base de données	9
<b>III. Architecture du système et choix technologiques</b>	<b>9</b>
<b>IV. Procédures d'évaluations et de tests</b>	<b>10</b>
<b>V. Procédure d'installation et de déploiement</b>	<b>10</b>
<b>VI. Manuel d'utilisation simplifié</b>	<b>10</b>

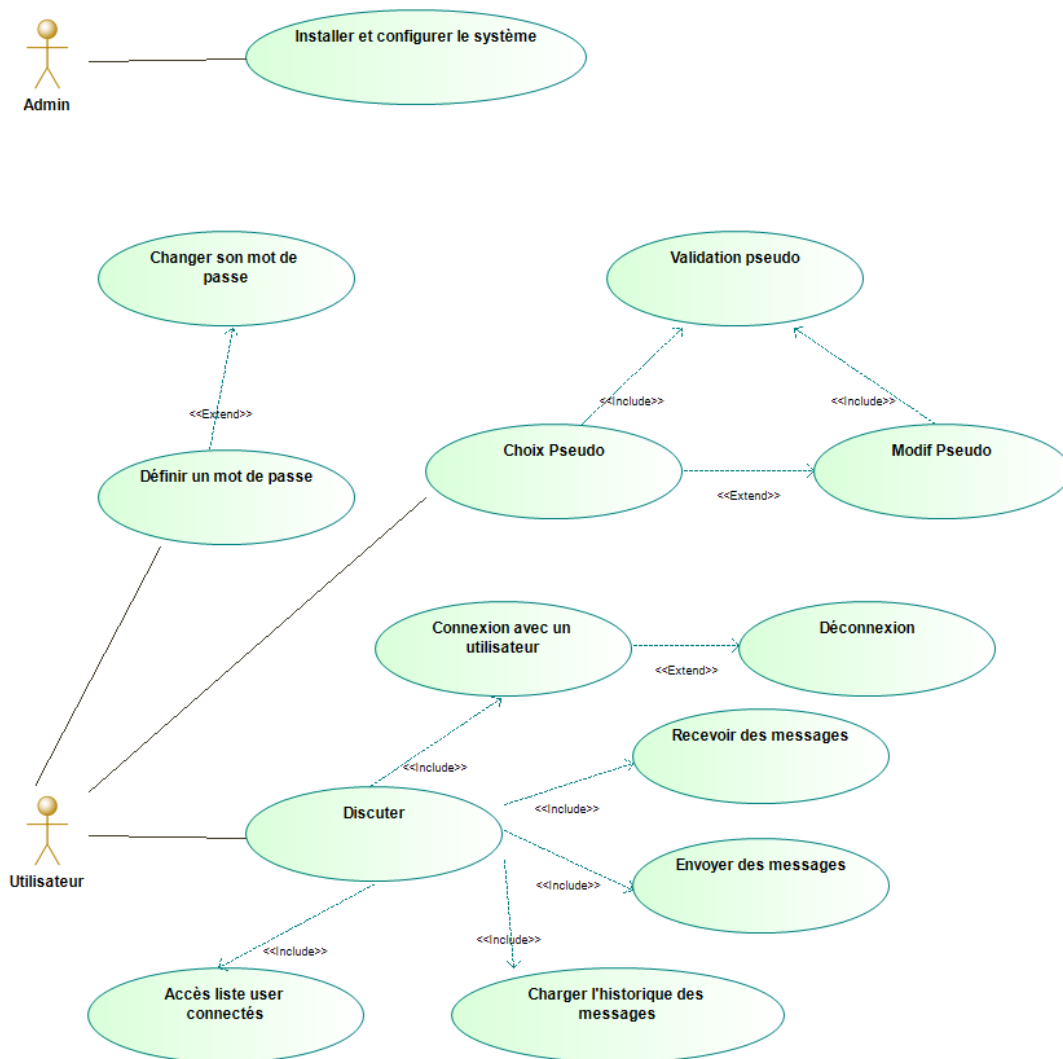
Version	Date	Modifications
V1	10/11/2022	Création et remplissage du document à l'issue de la conception
V2	10/01/2023	Mise à jour du document à l'issue de l'implémentation du projet

# I. Introduction

Le Client souhaitait un service d'échanges de messages textuels sur un réseau local, de façon décentralisée et avec une installation et une configuration simplifiée. Le détail des contraintes est fourni dans le Cahier des Charges

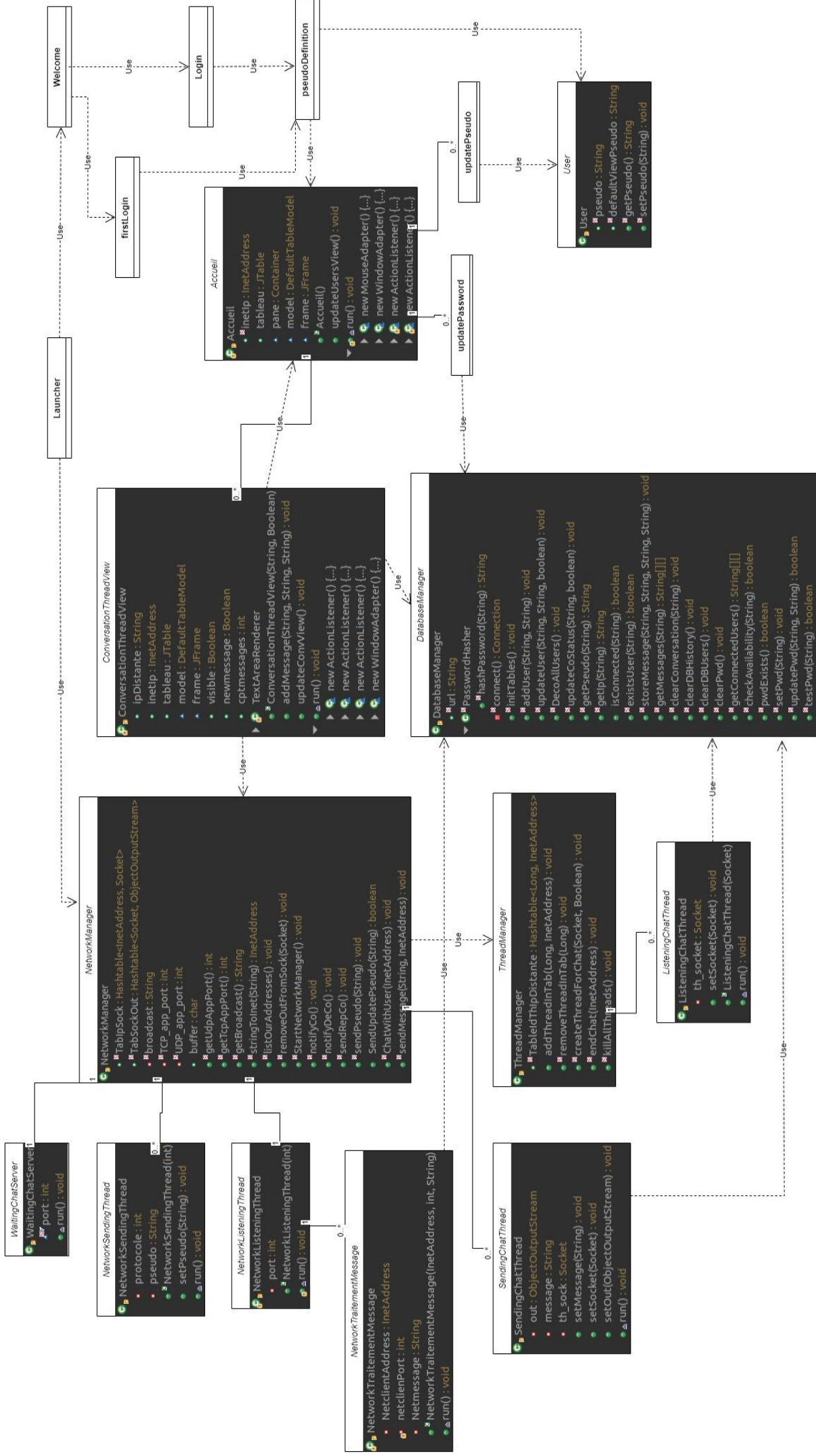
## II. Conception du projet

### II.1. Diagramme des cas d'utilisation



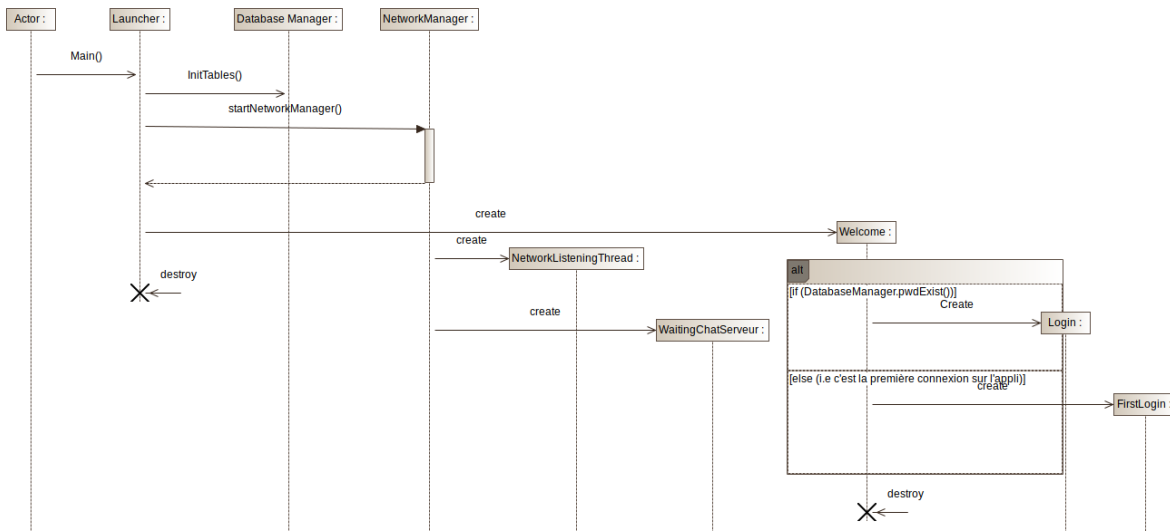
### II.2. Diagramme des classes

cf. page suivante

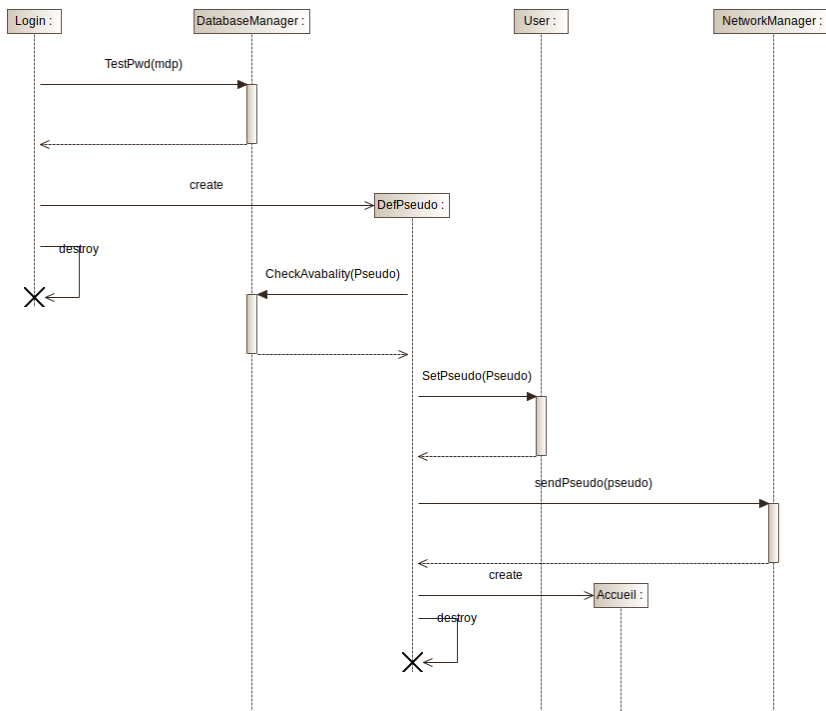


## II.3. Diagrammes de séquence

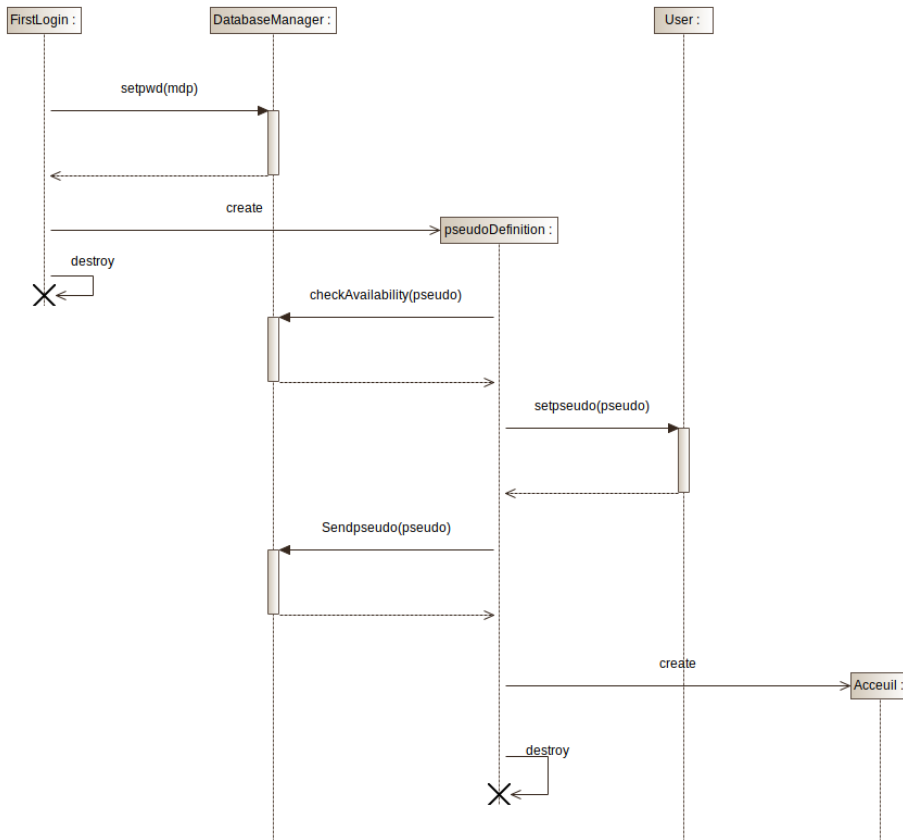
- Lancement de l'application



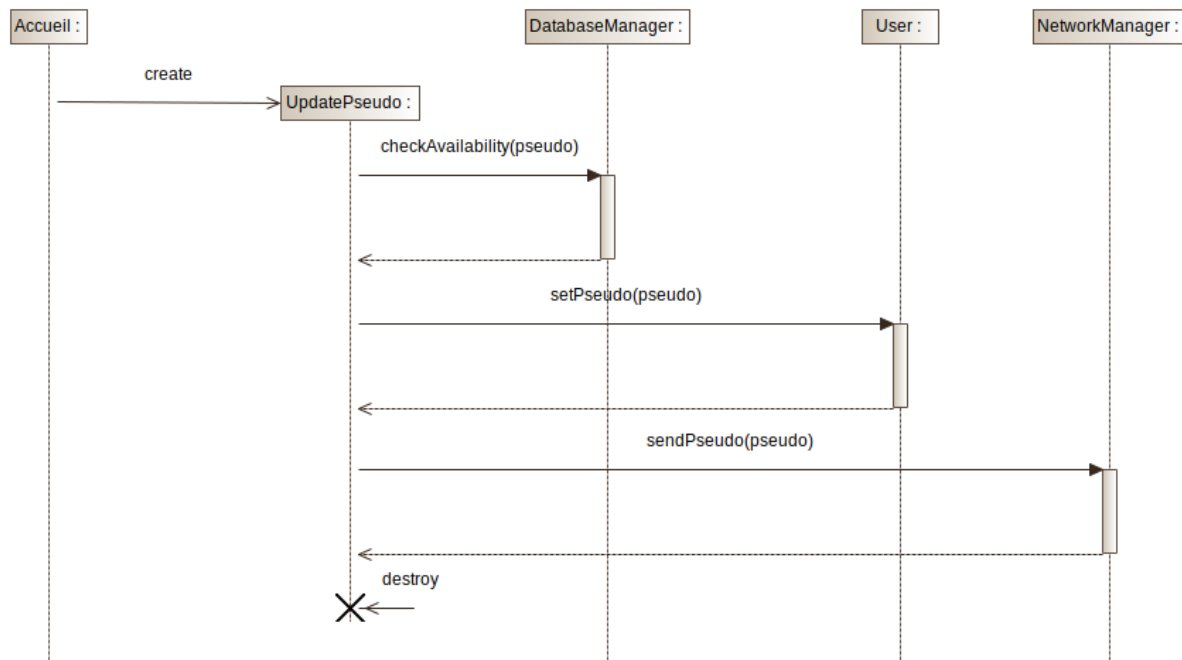
- Connexion (login)



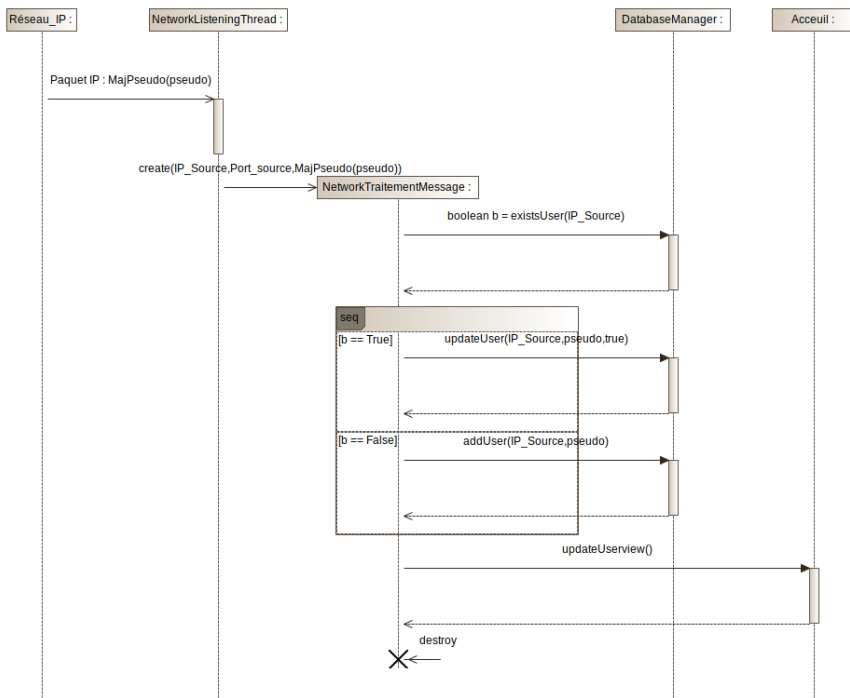
- Première connexion (first login)



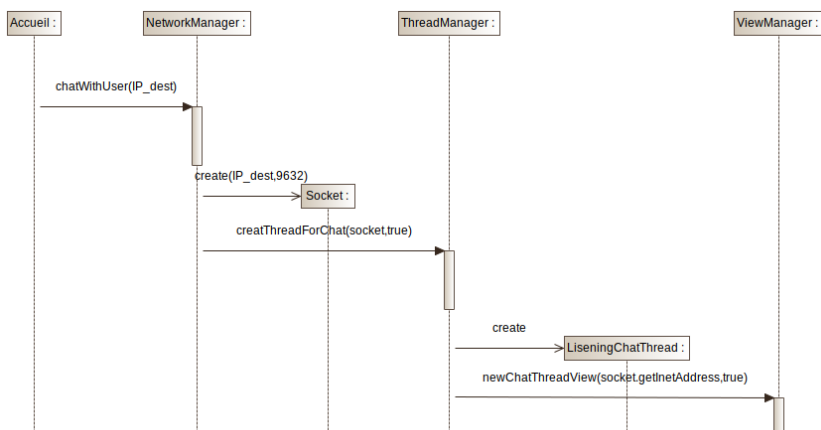
- Mise à jour du pseudo de l'utilisateur



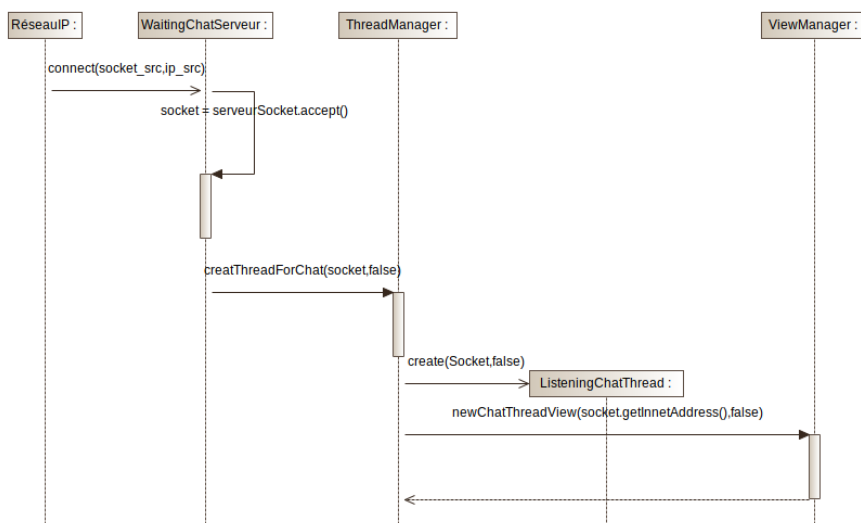
- Mise à jour du pseudo d'un autre utilisateur



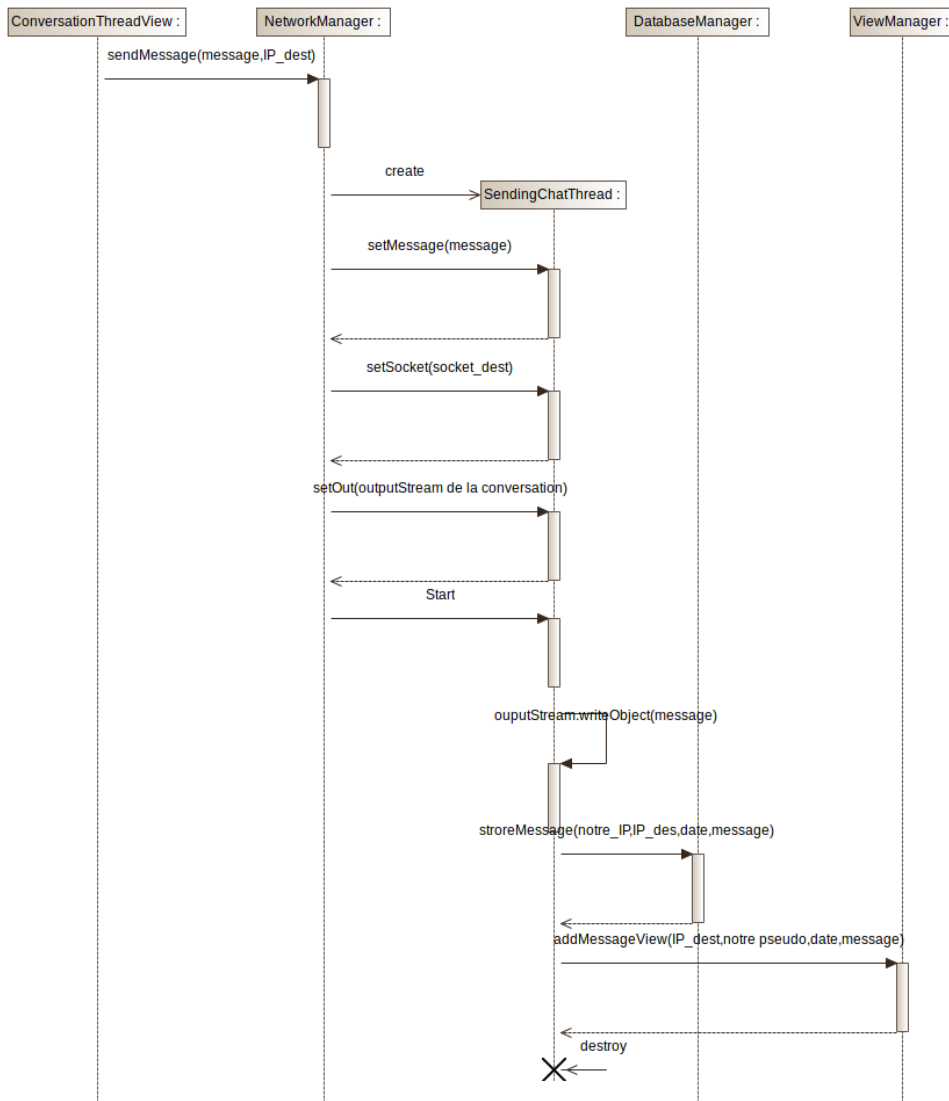
- Lancement d'une conversation par l'utilisateur



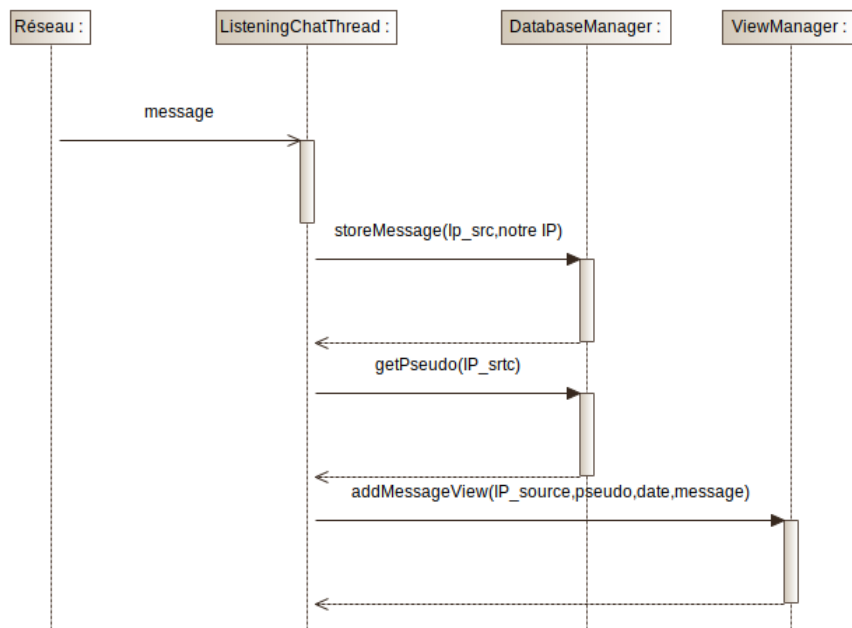
- Lancement d'une conversation par un autre utilisateur



- Envoi d'un message par l'utilisateur

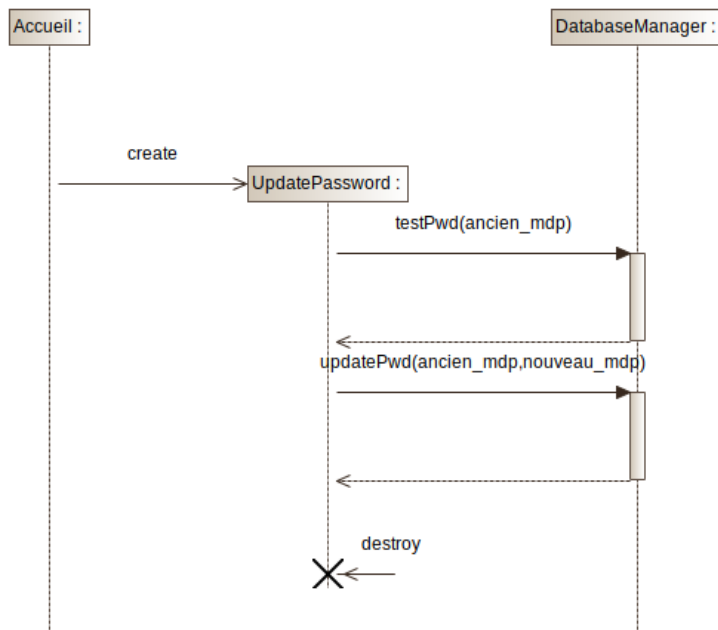


- Réception d'un message par l'utilisateur





- Mise à jour du mot de passe



## II.4. Implémentation de la base de données

La base de données est composée de 3 tables. La base de données est créée par l'application si elle n'existe pas au lancement.

La première table, *history*, contient l'historique des conversations ; elle stocke les messages un par un. La table est composée des champs suivants : #id (clé primaire), #from\_ip, #to\_ip, #date, #content. Ces champs permettent de conserver les conversations avec les autres utilisateurs même s'ils changent de pseudo.

La seconde table, *users*, stocke les informations concernant les autres utilisateurs de l'application. Elle est composée des champs suivants : #ip (clé primaire), #pseudo, #connected.

La dernière table, *pwd*, sert au stockage du mot de passe. Elle est composée des champs #id (clé primaire) et #password. Pour des questions de sécurité, le mot de passe est hashé.

## III. Architecture du système et choix technologiques

Nous avons choisi d'utiliser *Java Swing* pour l'interface graphique du projet car c'est la technologie que nous maîtrisons le plus.

Pour accéder à la base de données, nous avons utilisé SQLite ainsi que le module *sqlite-jdbc-3.7.2*. Ce module est inclus dans l'application, il n'est pas nécessaire de le télécharger séparément. La base de données est créée automatiquement à l'installation de l'application sur la machine hôte. En cas de suppression manuelle par l'utilisateur, elle sera régénérée automatiquement au prochain lancement de l'application.

Conformément au cahier des charges, l'application est décentralisée pour le stockage des messages (ainsi que le mot de passe de connexion). Chaque machine où l'application est installée possède sa propre base de données ; celle-ci n'est ni importée depuis une source externe ni exportée.

Port TCP : 9632 , sert pour les discussions par message.

Port UDP : 9500 , sert pour tous les protocoles de l'application en broadcast sur le réseau (notification que l'on est connecté / déconnecté et définition / changement de pseudo)

## IV. Procédures d'évaluations et de tests

Pour vérifier notre programme nous avons décidé d'effectuer une batterie de tests:

1. Nous avons entré le même pseudo qu'un autre utilisateur, et cela nous a envoyé un message d'erreur.
2. Nous avons entré le mauvais mot de passe et cela nous a envoyé un message d'erreur.
3. Nous envoyons et recevons correctement les messages et ceux-ci sont bien stockés dans la base de données et affichés sur l'application.
4. Plusieurs utilisateurs envoient des messages à la même personne, et celle-ci les reçoit bien.
5. Quand un utilisateur se connecte ou se déconnecte, il apparaît et disparaît bien de la liste à gauche chez les autres utilisateurs.
6. Lors de la déconnexion d'un utilisateur, les conversations ouvertes qu'il avait se ferment pour les autres utilisateurs.
7. L'historique reste persistant après une déconnexion puis reconnexion.
8. Quand un utilisateur change son pseudo, celui-ci est actualisé pour les autres utilisateurs, à la fois dans la liste de gauche et dans les conversations ouvertes.
9. Les messages envoyés à d'autres utilisateurs connectés sont bien reçus, et dans le cas où la conversation n'était pas ouverte, une notification s'affiche pour le destinataire.
10. Lors d'un changement de mot de passe, c'est bien le nouveau mot de passe qui est attendu lors d'une reconnexion à l'application.

## V. Procédure d'installation et de déploiement

Le fichier *README.md* contient l'essentiel des informations nécessaires à l'installation et au déploiement de l'application.

Pour installer l'application sur une machine (Windows, Linux, Android, Mac), il suffit de procéder de la manière suivante :

- Créer un dossier pour installer l'application
- Copier l'archive *ChatProject.jar* dans ce dossier
- Créer un script (sur le bureau par exemple) qui lance l'application
  - exemple sous Linux : copiez le code suivant dans un fichier *.sh*  

```
java -jar emplacement_de_l_archive/ChatProject.jar
```

À noter que l'application va créer une base de données *chatdb.db* dans le même répertoire que celui de l'archive jar.

L'application est alors prête et l'utilisateur peut communiquer avec toutes les autres machines sur le réseau local (sous réserve d'installation de l'application sur les autres machines).

## VI. Manuel d'utilisation simplifié

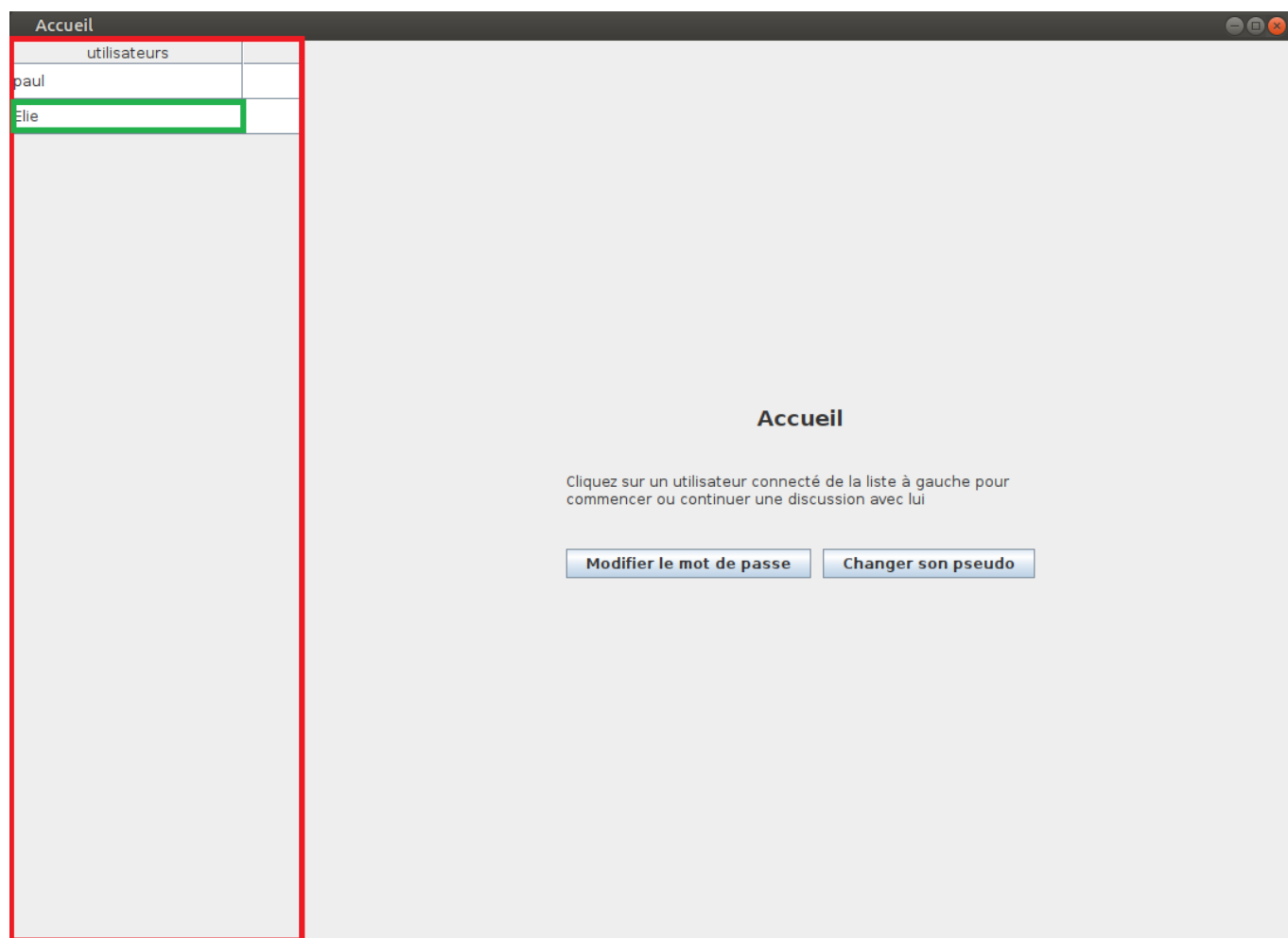
Pour lancer l'application, exécutez le script pointant vers l'archive de l'application (cf. section précédente).

À l'ouverture vous avez deux choix :

1. *Connexion* → permet de se connecter en insérant son mot de passe. Si c'est la première connexion, il faudra définir son mot de passe d'accès à l'application.
2. *Réinitialiser* → permet de réinitialiser l'application (mot de passe y compris) et supprime les conversations enregistrées, la création d'un nouveau mot de passe vous sera demandée

Une fois connecté, il faut choisir son pseudo. Un pseudo peut être refusé si celui-ci est déjà utilisé par un autre utilisateur connecté de l'application.

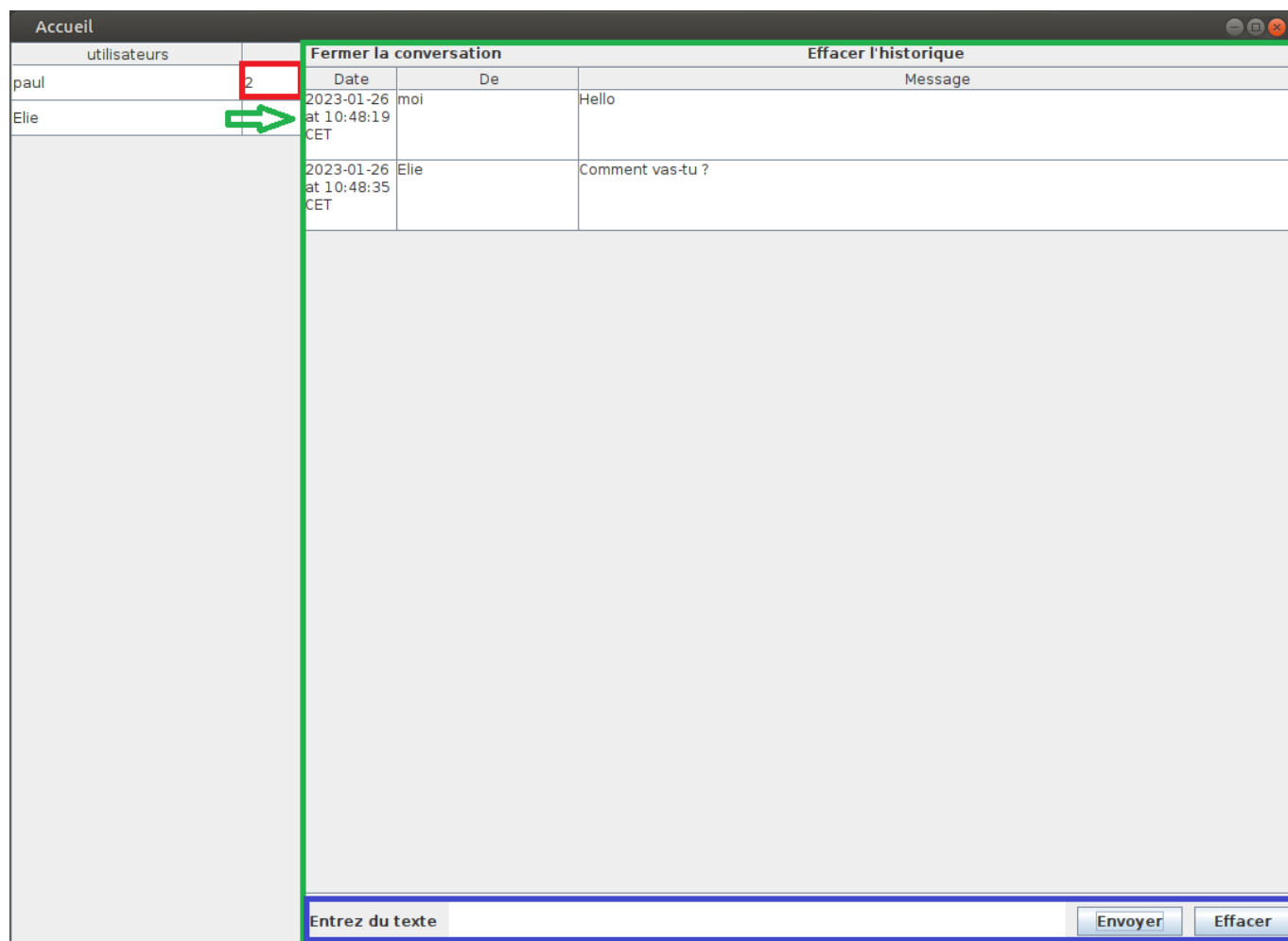
Une fois le pseudo défini, la page d'accueil s'ouvre.



Sur le panneau latéral gauche (en rouge sur la capture d'écran) se trouve la liste des utilisateurs connectés.

Il suffit de cliquer sur l'un d'eux pour accéder à la conversation avec celui-ci et envoyer des messages.

Il est aussi possible de changer son pseudo / modifier son mot de passe via les boutons prévus à cet effet.



Une fois que nous avons cliqué sur un utilisateur, la conversation avec celui-ci s'affiche (cadre vert). Il est possible d'envoyer des messages via la zone de texte dans le cadre bleu et en cliquant ensuite sur le bouton *Envoyer*.

En rouge ici vous pouvez voir le nombre de messages non lus qu'un utilisateur a envoyé. Ici paul a envoyé 2 messages qui n'ont pas encore été lus.

Il est possible de switcher entre plusieurs conversations en cliquant sur leur pseudo dans le panneau latéral gauche.

Pour revenir à l'écran d'accueil cliquez simplement sur *Fermer la conversation* en haut du cadre bleu.

Si vous souhaitez supprimer l'historique des messages avec un utilisateur, il suffit de cliquer sur *Effacer l'historique* en haut du cadre bleu.

Pour quitter l'application il suffit de fermer la fenêtre principale via la croix rouge en haut à droite de l'application.