

RIZZI Emma
ROUSSEAU Patrick
4 IR-SI A1

Chat System

Conception

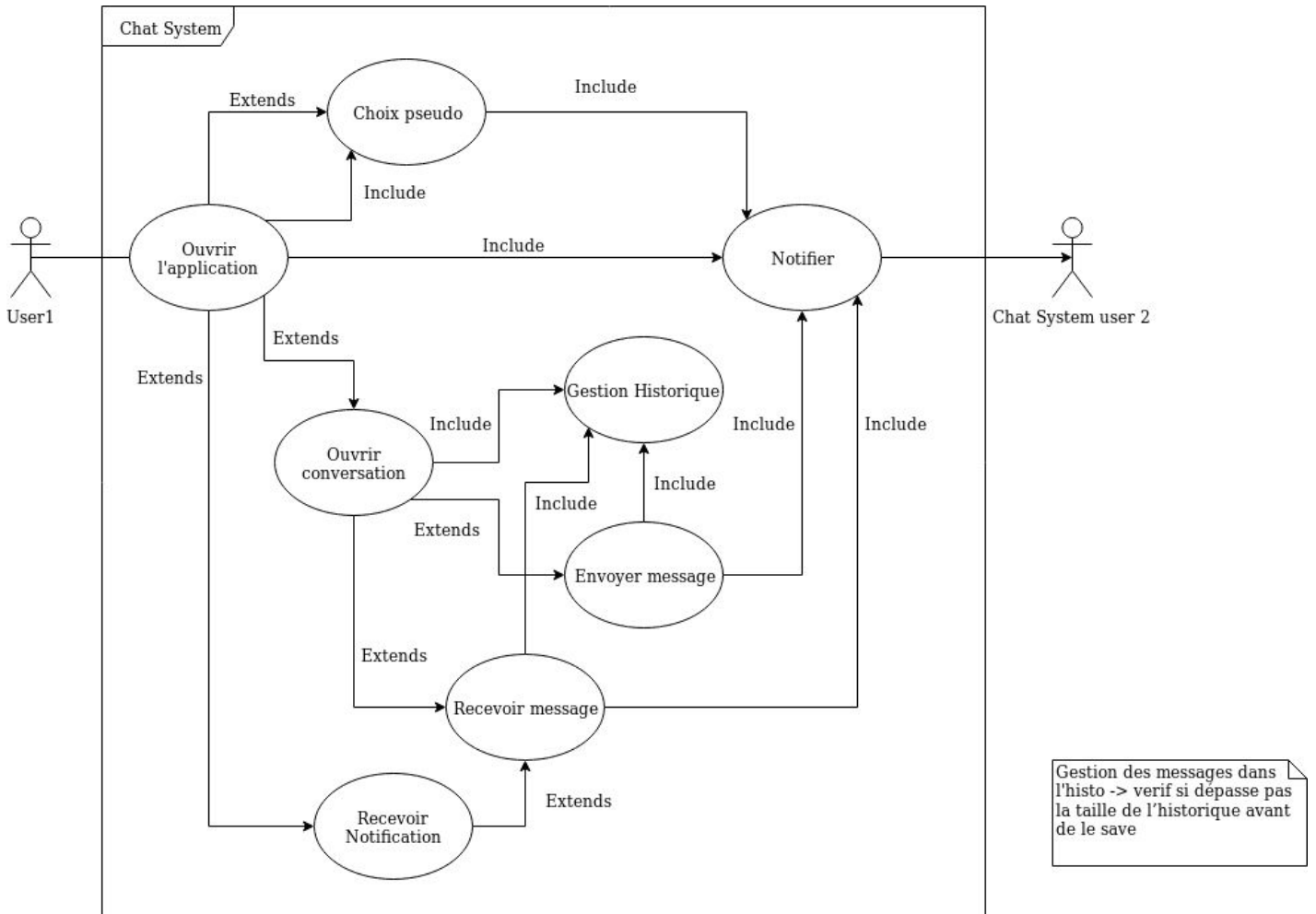
POO/COO

Sommaire

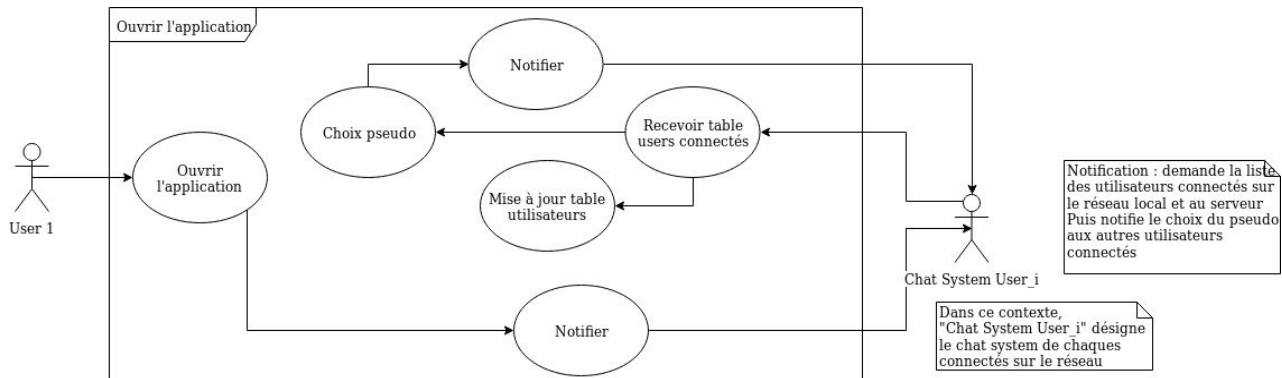
Diagrammes des cas d'utilisation :	3
Cas général : utilisation globale de l'application	3
Ouverture de l'application	4
Choix du pseudo	4
Ouverture d'une conversation	4
Envoi d'un message	5
Réception d'un message	5
Diagramme de séquence	6
Ouverture de l'application (dans le réseau local)	6
Ouverture de l'application (à l'extérieur du réseau local)	6
Envoi/réception d'un message	8
Diagramme de classe	8

1.Diagrammes des cas d'utilisation :

a. Cas général : utilisation globale de l'application



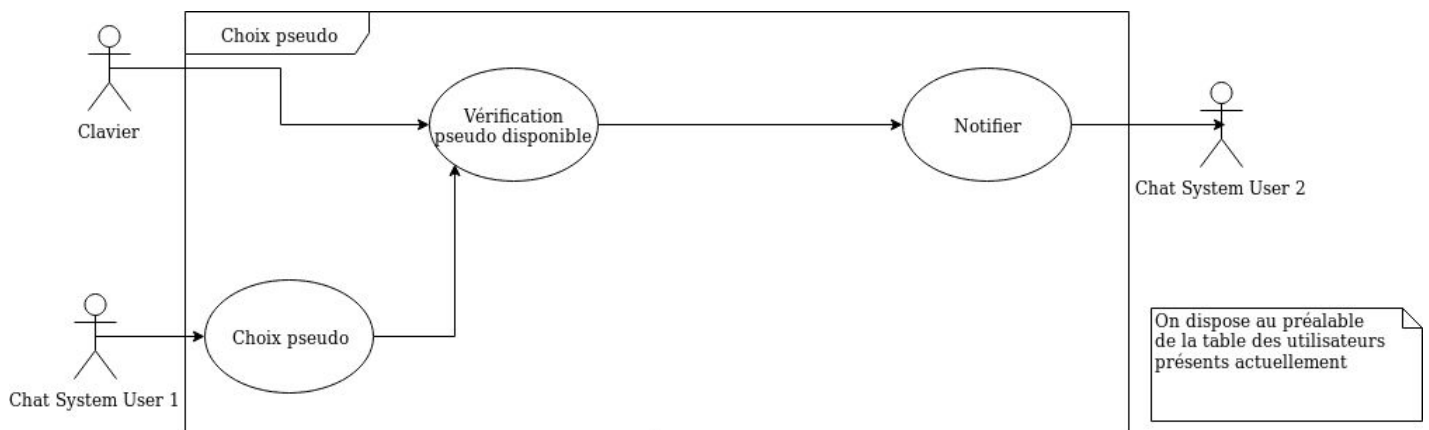
b. Ouverture de l'application



Ici lors de l'ouverture de l'application, celle-ci va demander à toutes les application connectés au réseau leurs coordonnées grâce à un broadcast sur le réseau local, ces derniers vont ensuite répondre et ainsi constituer une table d'utilisateurs sur l'application qui vient de se connecter. L'appli ouvrante va pouvoir demander son nouveau pseudo à l'utilisateur et notifier les autres application des coordonnées de ce nouvel utilisateur et de sa connexion. A la première ouverture de l'application, la table des utilisateurs est initialisée, avec pseudo, adresse IP et adresse MAC. L'IP nous permettra de communiquer avec un utilisateur, et mis à jour si elle change. L'identification se fait par l'adresse MAC, avec un utilisateur par machine.

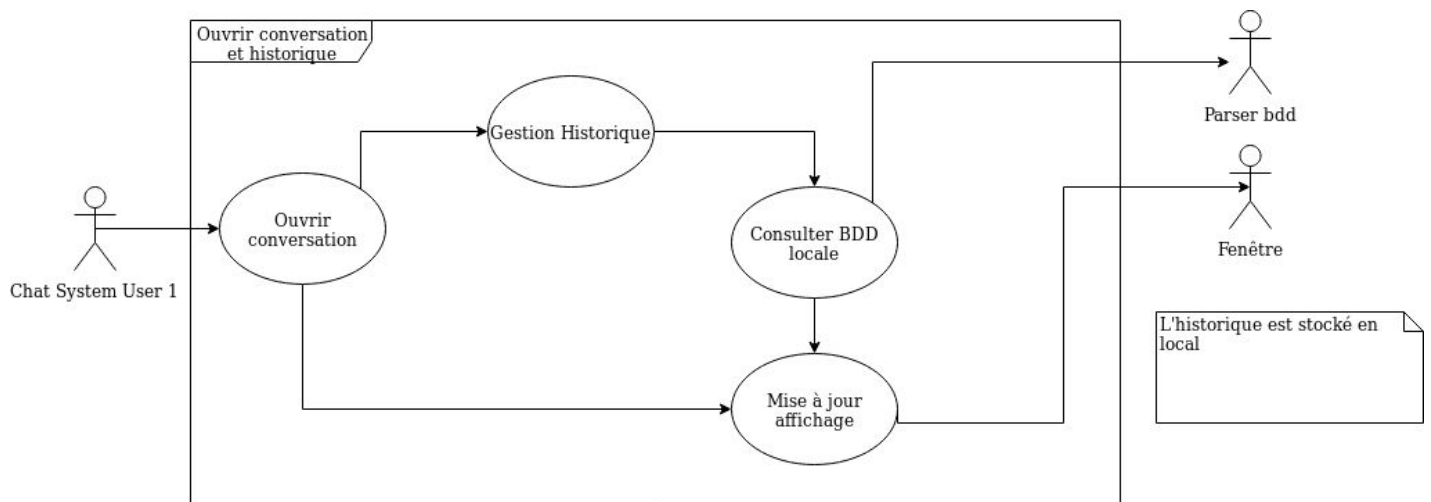
c. Choix du pseudo

ici on detail l'étape du choix du pseudo, avec l'étape de vérification de pseudo disponible, qui bloque l'utilisateur jusqu'à ce qu'il entre un pseudo accepté par l'application. La notification du nouveau pseudo met à jour la table utilisateur dans le serveur pour les utilisateurs extérieurs. Si l'application est déjà initialisée sur la machine, on vérifie si le pseudo est connu dans la table des utilisateurs. Si l'adresse MAC est celle de la machine utilisée, on se connecte et on peut accéder à l'historique local, sinon la connexion est refusée et il faut choisir une autre pseudo.



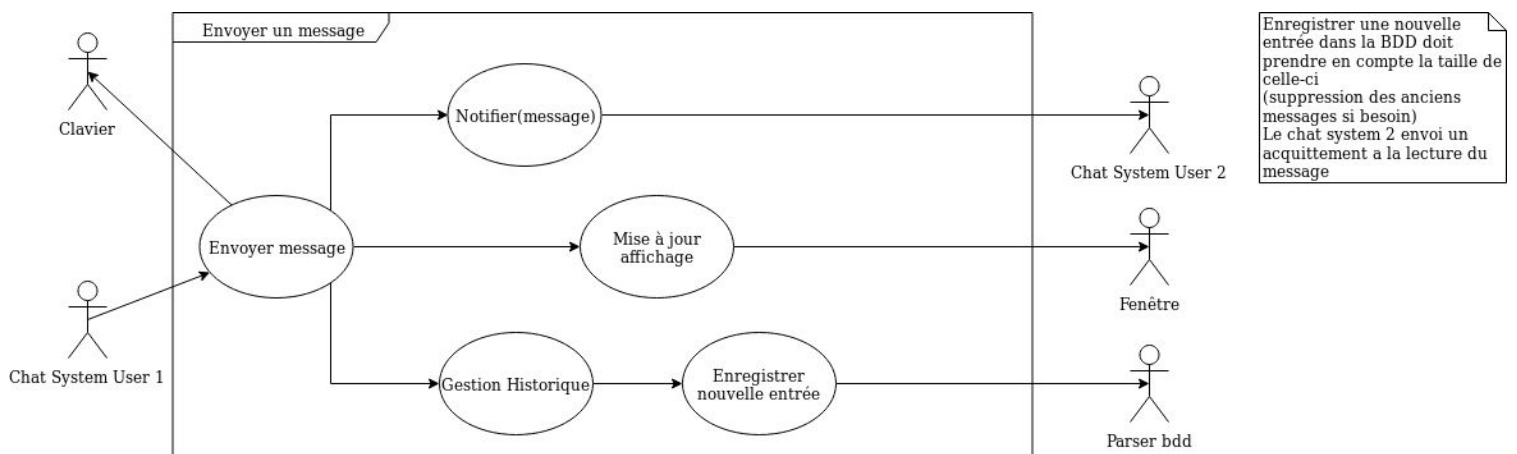
d. Ouverture d'une conversation

Lors de l'ouverture d'une conversation avec un autre utilisateur, on va charger tout l'historique des messages (sauvé en local) avec cet interlocuteur grâce au parser. Puis on déclenche une mise à jour de l'affichage pour rafraîchir la fenêtre avec son nouveau contenu.



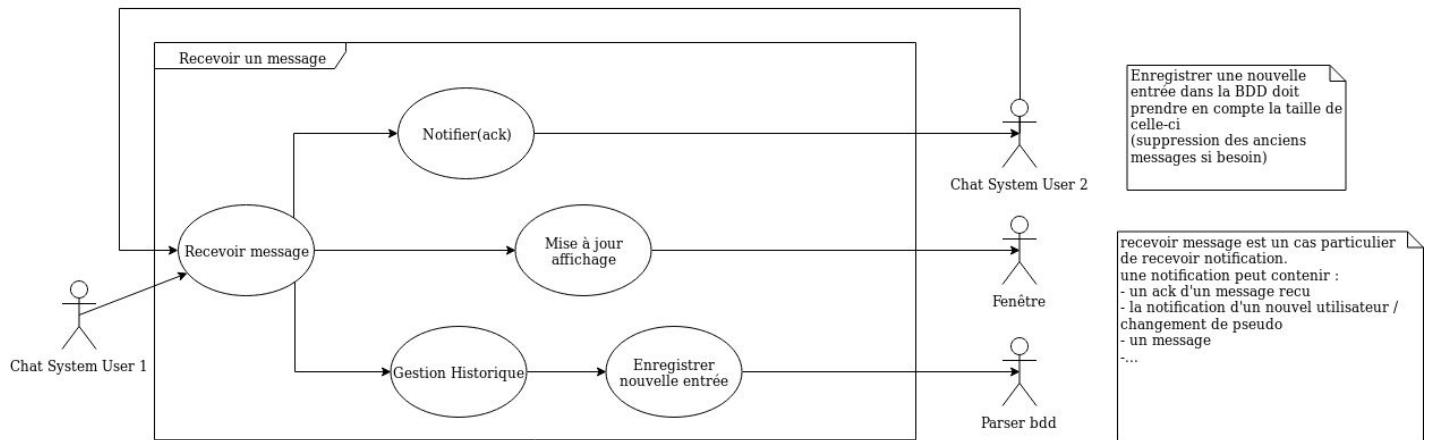
e. Envoi d'un message

Lors de l'envoi d'un message avec l'application, si c'est le premier message de l'échange, il y a un établissement de connexion TCP avec l'appli distante. Sinon dans les deux cas trois choses sont déclenchées, on sauve le message en local dans l'historique, on affiche ce message sur l'appli, et on envoie le message en TCP au l'appli distant (on récupère ces coordonnées dans la table utilisateur établie précédemment)



f. Réception d'un message

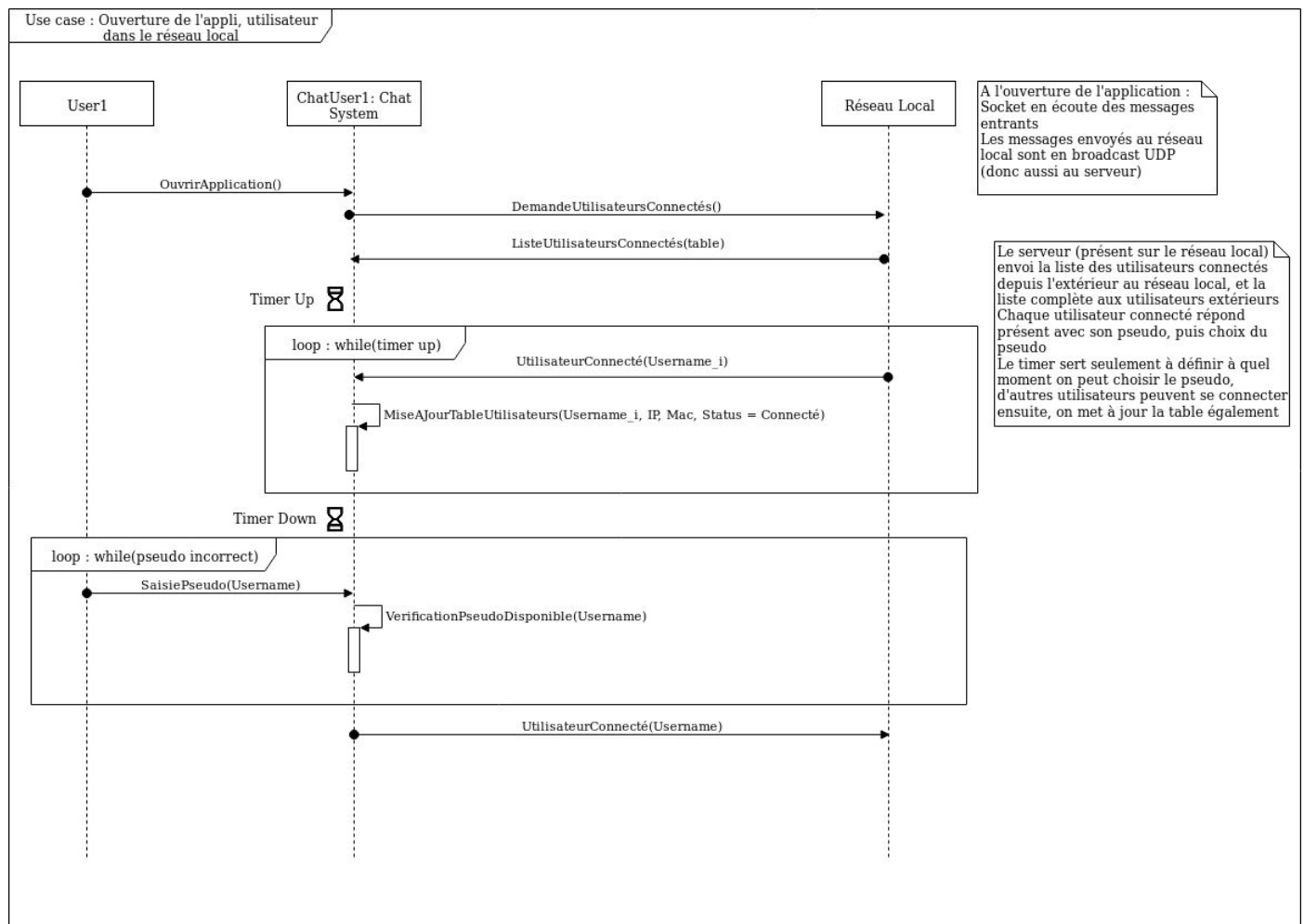
La réception a l'image de l'émission déclenche les trois mêmes processus, sauf que l'envoi est un acquittement de la lecture du message par l'utilisateur recevant, cet envoi s'effectue si l'utilisateur a la conversation ouverte ou lorsqu'il la consultera.



2. Diagramme de séquence

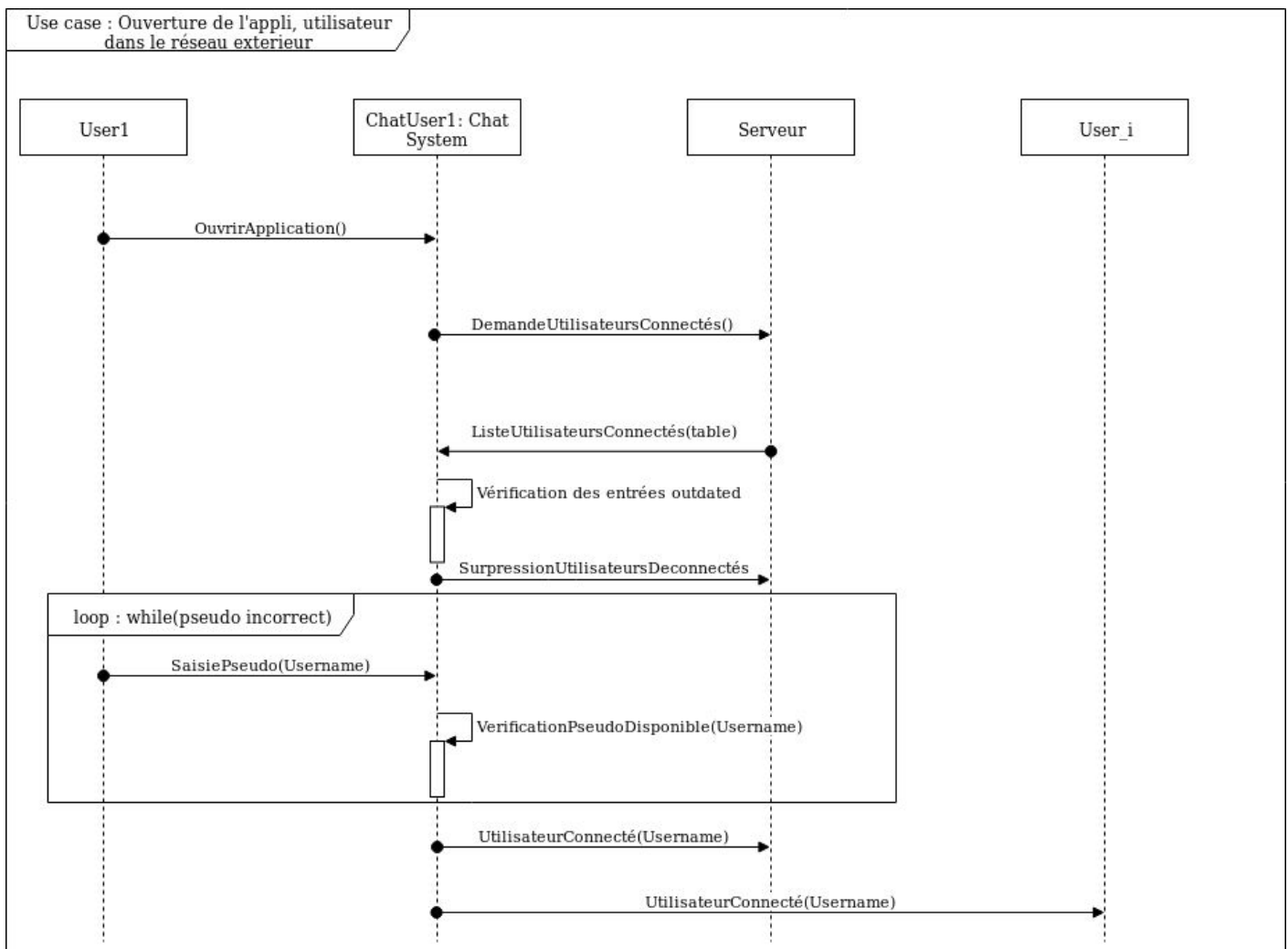
a. Ouverture de l'application (dans le réseau local)

On distingue le cas où l'utilisateur est présent sur le réseau local ou non. A l'ouverture depuis le réseau local, l'utilisateur signal sa connexion en broadcast sur le réseau (et au serveur également présent sur ce réseau), et chacun lui répond en unicast qu'il est présent, en indiquant son pseudo. Une fois le pseudo choisi, celui ci est signalé à tous les utilisateurs en unicast (réseau local ou non). La demande des utilisateurs sur le réseau local se fait en broadcast par UDP, on connaît l'adresse IP du réseau. Le serveur envoie la liste des utilisateurs en dehors du réseau, et chaque utilisateur connecté répond présent. On construit ainsi la liste des utilisateurs connectés avec pseudo, adresse IP, adresse MAC. Toutes les communications suivantes sont en TCP car on a les informations nécessaires pour établir la connexion.



b. Ouverture de l'application (à l'extérieur du réseau local)

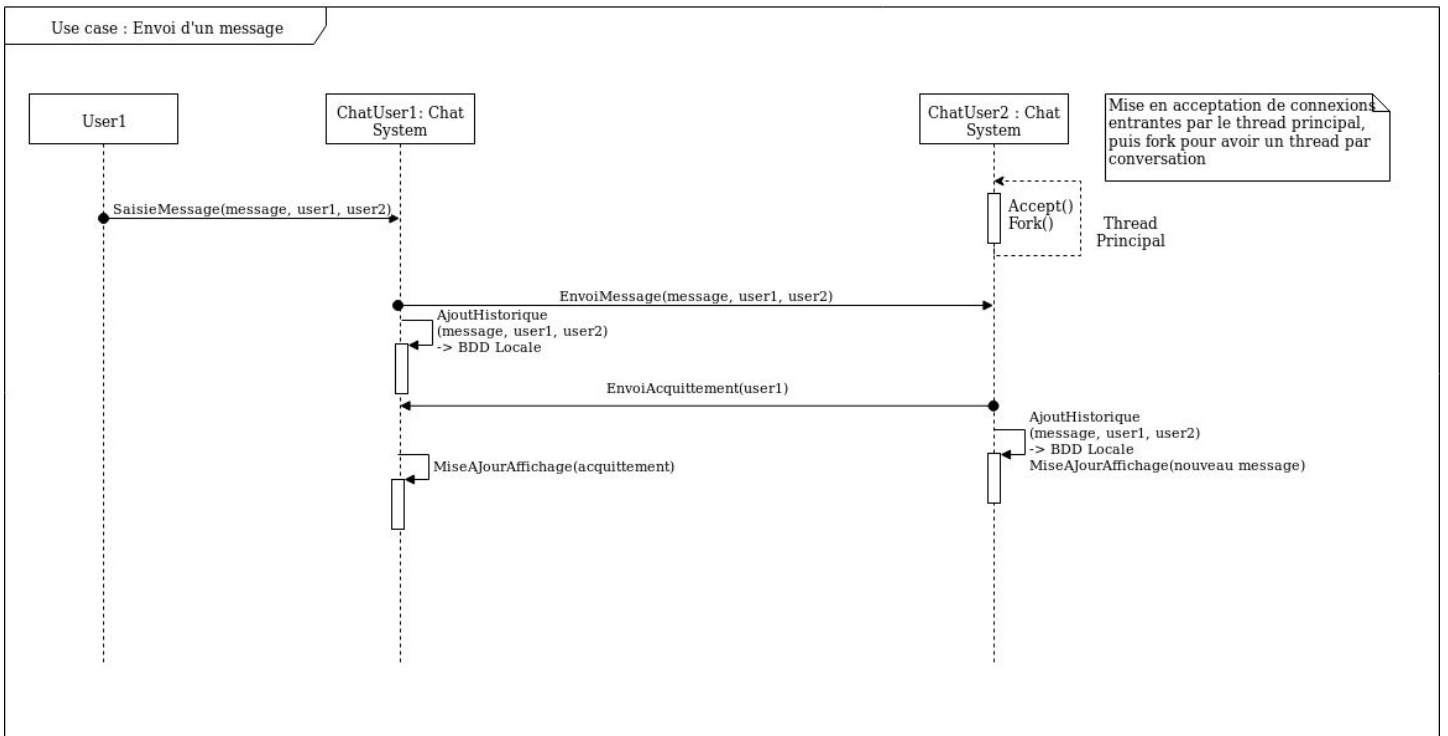
Depuis l'extérieur, on demande directement au serveur la liste des utilisateurs présents. Tous les messages sont envoyés par TCP, car on connaît l'adresse IP du serveur à l'avance, puis les adresses des autres utilisateurs sont récupérées par la liste des utilisateurs.



c. Envoi/réception d'un message

La connexion TCP est établie sur un port fixé à l'avance, et avec l'adresse IP connue dans la table des utilisateurs.

Prérequis :
Les deux utilisateurs sont connectés
User1 a ouvert la conversation avec
User2 (chargement de l'historique)



EnvoiMessage :
Premier message : ouvre une connexion TCP entre ChatUser1 et ChatUser2, maintenue jusqu'à un certains temps sans envoi de message, ou si trop de conversation sont ouvertes (fermeture de la plus inactive pour créer une nouvelle)

3. Diagramme de classe

Le chat system s'initialise par la classe Initialiseur au lancement de l'application, UtilisateurDAO permet d'obtenir la liste des utilisateurs par requête au serveur distant, et/ou Initialiseur broadcast au réseau local pour connaître les utilisateurs présents. Il gère ensuite le choix du pseudo et notifie les utilisateurs présents de la connexion d'un nouvel utilisateur.

La classe Listener représente un thread lancé après connexion de l'utilisateur. Il contient un serveur Socket qui permet d'accepter les connexions entrantes (conversation initiée par un autre utilisateur), il lance un nouveau thread par conversation : Gestionnaire Réseau qui correspond à une conversation avec un socket associé à chaque conversation. Le gestionnaire réseau permet de recevoir et envoyer des messages à un destinataire.

A la réception d'une notification, le gestionnaire réseau transmet le message à l'objet conversation qui traite le contenu si c'est un message (HistoriqueDAO) sinon au chat system si c'est un nouvel utilisateur connecté (car il contient la liste des utilisateurs connectés durant l'utilisation de l'application). Il notifie également le chat system par le design pattern des observer/notifier, pour qu'il mette à jour l'affichage en faisant appel au Gestionnaire d'affichage.

La conversation est l'objet manipulé par l'application lors de l'échange de message contenant les informations nécessaires au maintien de la conversation. Historique est l'objet qui sera stocké dans la base de données locale par l'application, afin de pouvoir consulter les anciens messages à la réouverture d'une conversation. HistoriqueDAO fait l'interface avec cette base locale.

