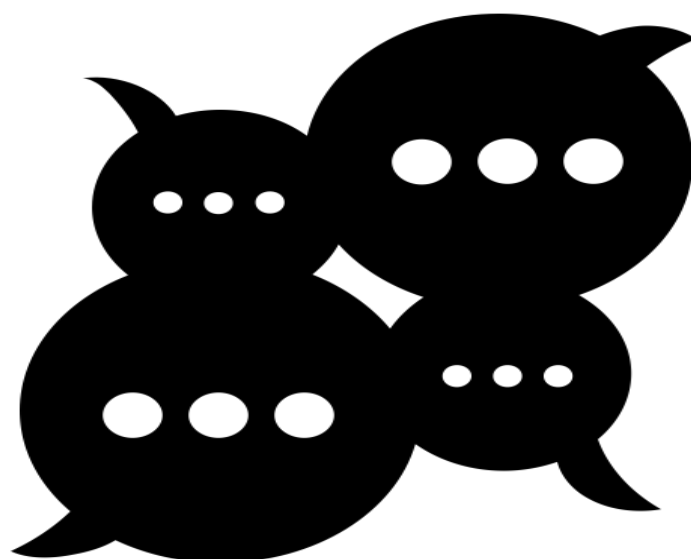


RAPPORT COO



CHAT'APP

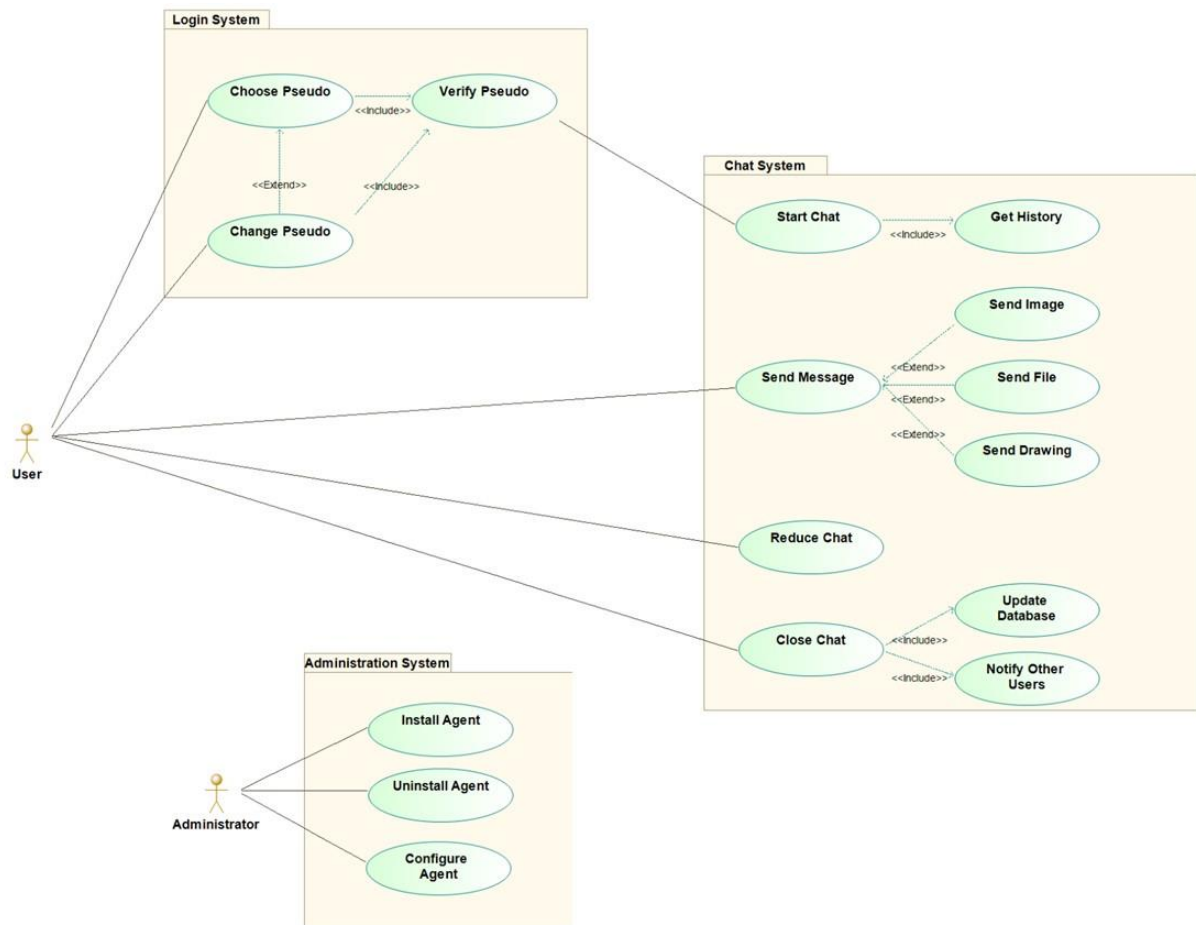
Réalisées par Vincent BAURES, Arnaud BUISSON et Alicia CALMET

Sommaire

1) Diagramme de cas	1
2) Diagrammes de Séquences	2
a- Création d'un utilisateur.....	2
b- Authentification.....	3
c- Choix du pseudo	4
d- Démarrage d'un chat.....	5
e- Envoie message	6
f- Déconnexion.....	7
3) Diagramme de classe.....	9

1) Diagramme de cas

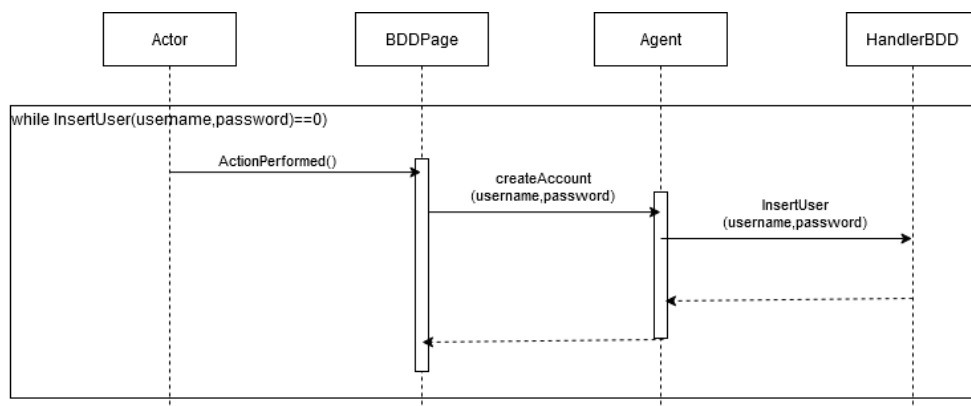
Notre diagramme de cas présente les actions qui doivent être réalisables dans notre application.



2) Diagrammes de Séquences

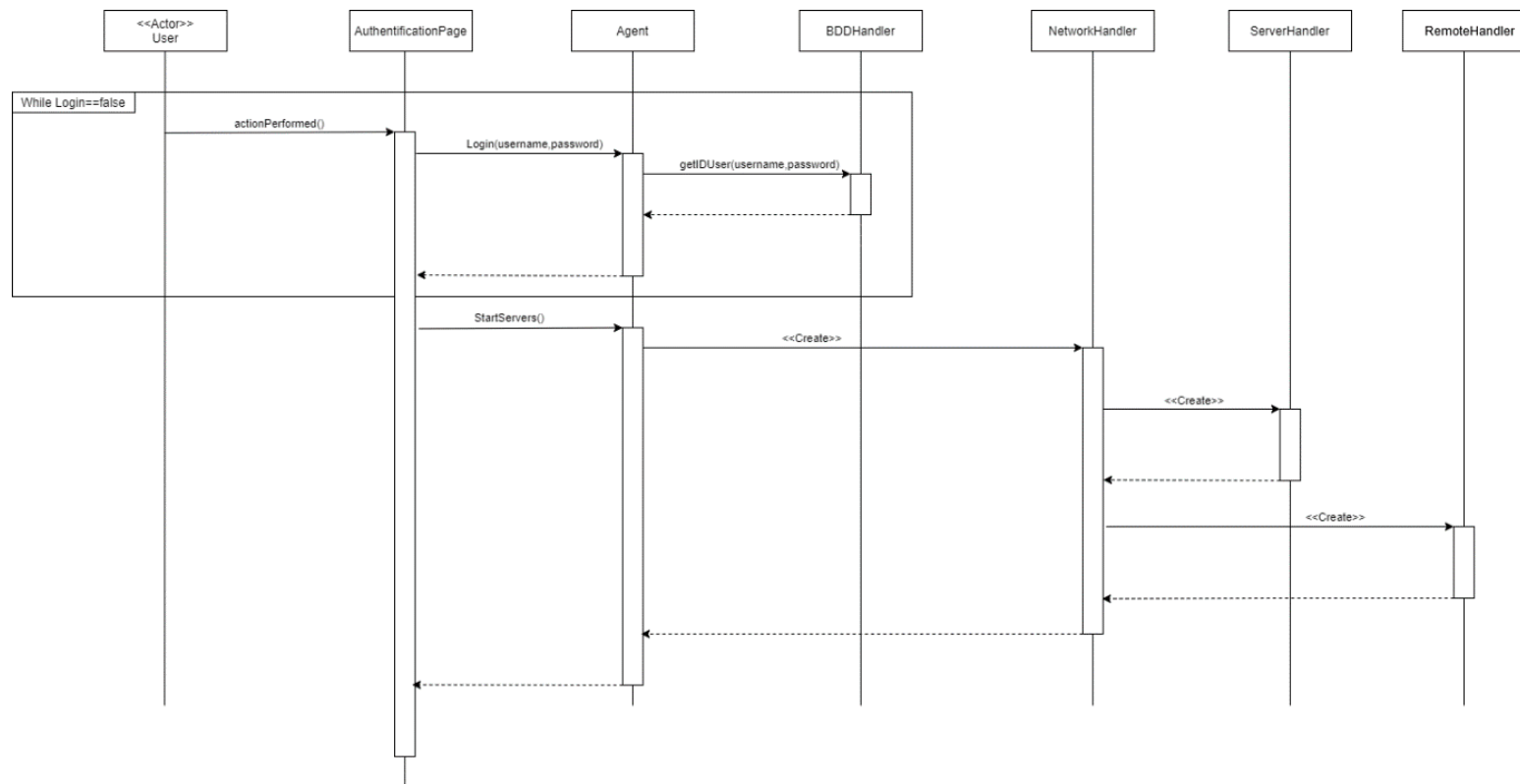
a- Création d'un utilisateur

Nous avons décidé de ne pas laisser la création des utilisateurs à l'administrateur mais de donner à tout un chacun la possibilité de se créer un compte. Pour se faire, l'utilisateur va appuyer sur un bouton qui lui permet d'accéder à la page d'inscription via la page d'authentification. Une fois cette page ouverte, il pourra rentrer un username et un password. Ce username et ce password seront envoyés à l'agent qui demandera au HandlerBDD de les ajouter à la base de données. Le HandlerBDD retournera le nombre de lignes ajoutées à la base de données. En effet, si le username et le password existent déjà ils ne seront pas ajoutés et l'utilisateur devra trouver de nouveau identifiant.



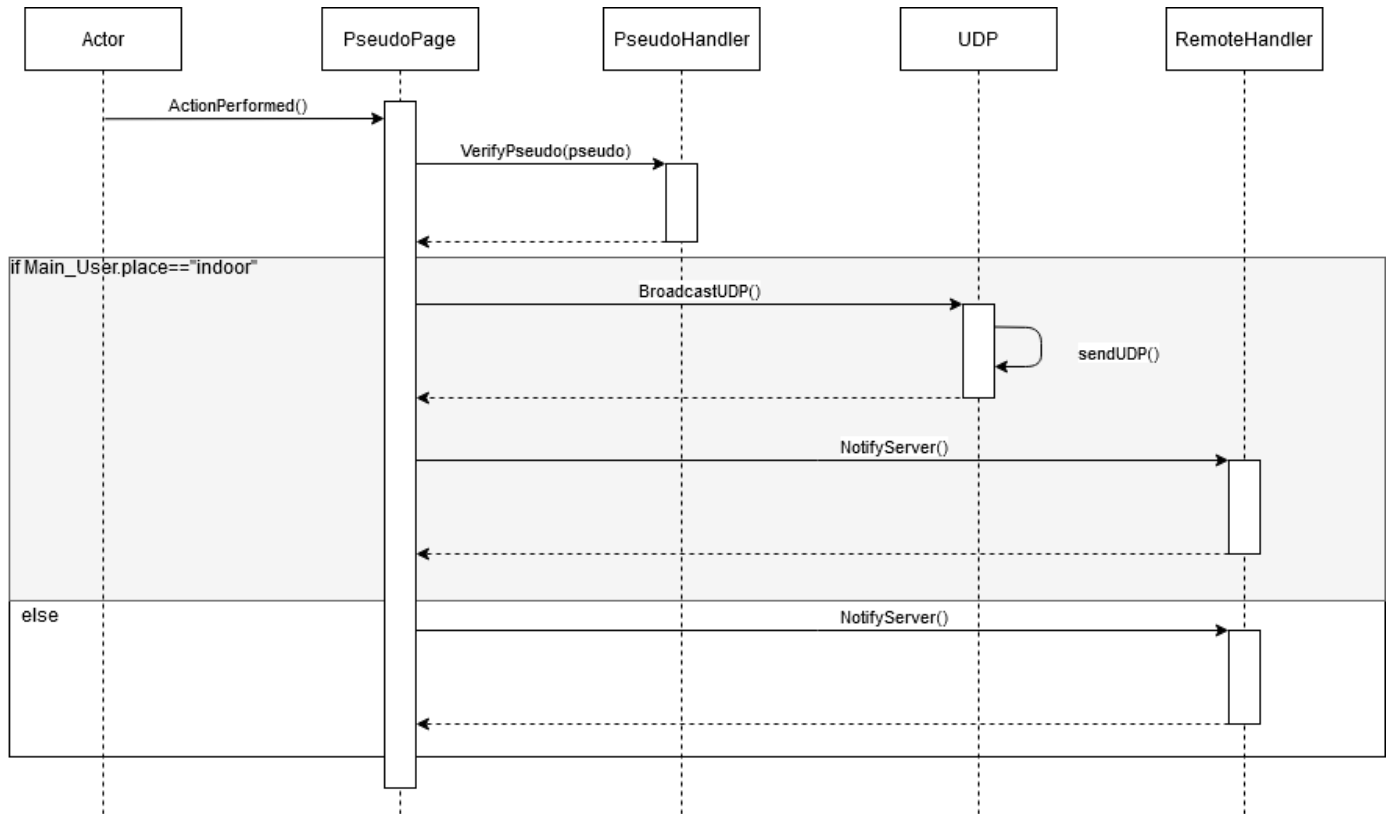
b- Authentification

Si l'utilisateur a déjà un compte ou s'il vient de s'en créer un, il pourra se connecter via l'interface de connexion. C'est alors que le pseudo et le mot de passe entrés dans la fenêtre de connexion seront envoyés à l'agent qui demandera au HandlerBDD de vérifier leur existence dans la base de données. Le cas échéant, toutes les classes permettant l'accès au réseau seront créées via le NetworkHandler préalablement créé par l'Agent.



c- Choix du pseudo

L'utilisateur pourra choisir de changer son pseudo à chaque instant. Pour cela, il accèdera à une page où il pourra entrer son nouveau pseudo. Le PseudoHandler vérifiera alors que ce pseudo est unique en le comparant au pseudo des autres utilisateurs connectés qu'il possède dans sa liste. Si le pseudo est unique en fonction de la place dans le réseau l'application notifiera les utilisateurs locaux via broadcast et/ou le server de présence et fermera la page de choix du pseudo.



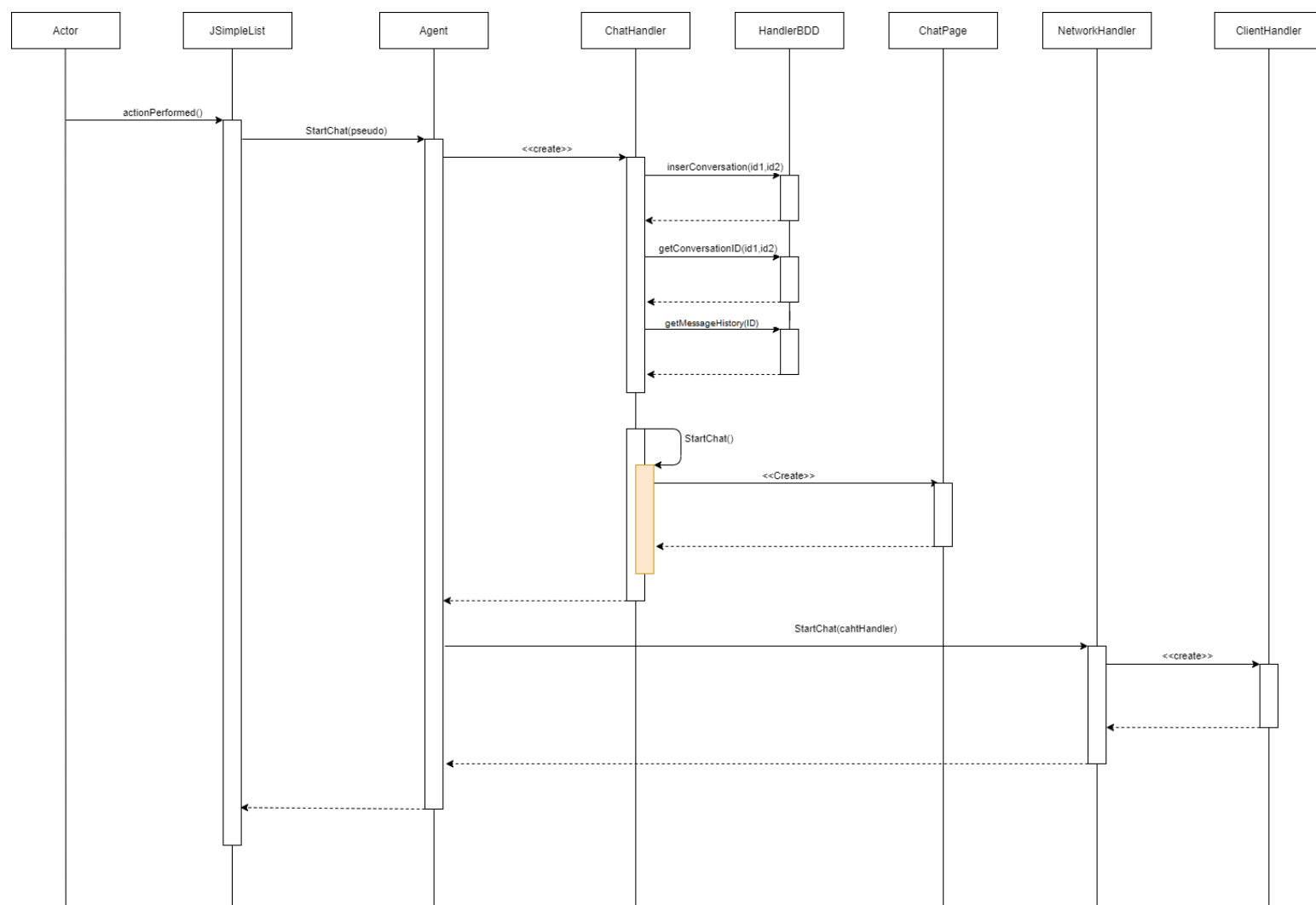
d- Démarrage d'un chat

Afin de démarrer un chat, l'utilisateur sélectionnera une des personnes connectées au sein d'une JSimpleList. Une fois cette personne choisie, il sera demandé à l'agent de créer un ChatHandler pour gérer l'échange de messages.

Une fois le ChatHandler créé, celui-ci créera via au HandlerBDD une nouvelle conversation dans la base de données avant de récupérer les anciens messages des conversations passées avec cette personne.

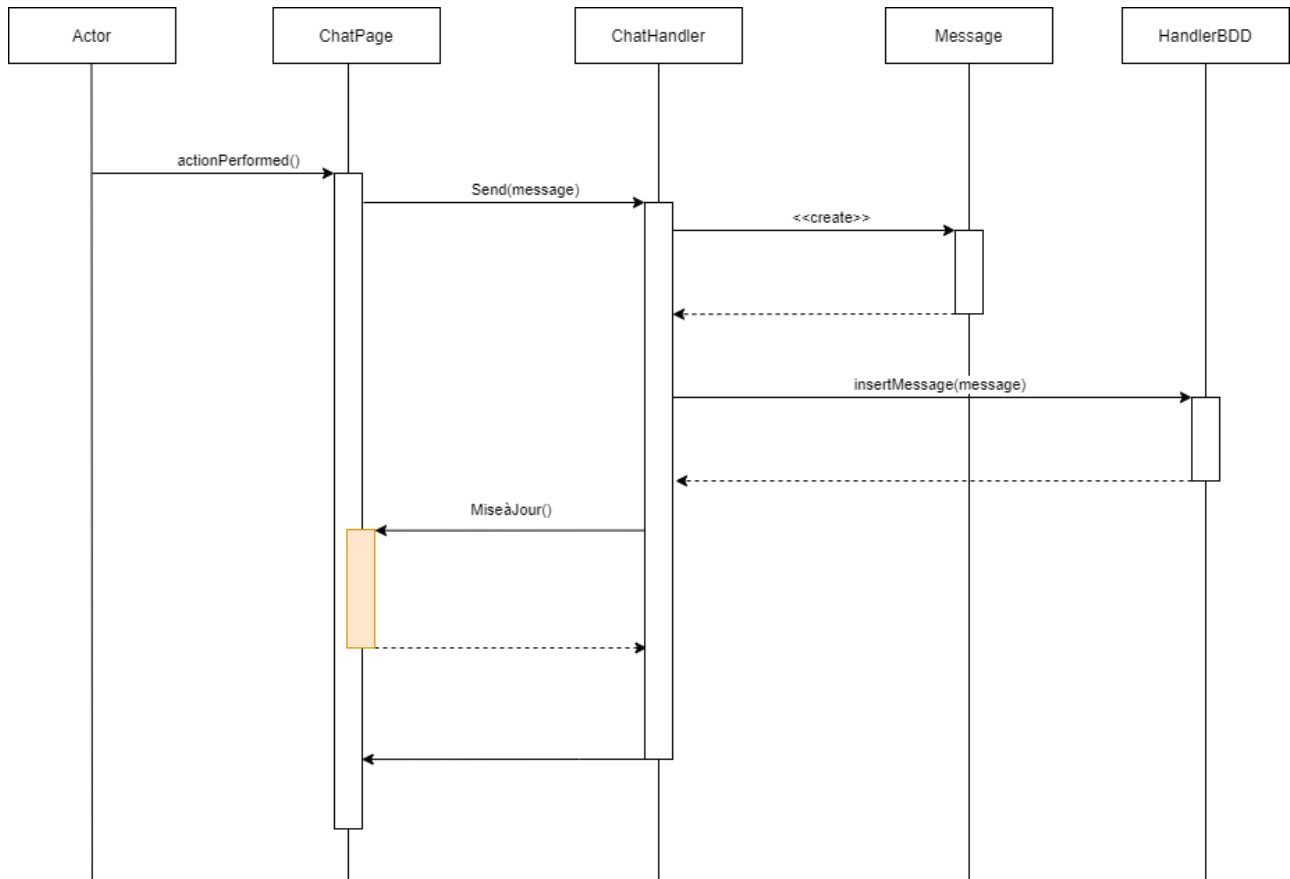
La page de communication sera ensuite affichée afin que l'utilisateur puisse envoyer des messages.

L'agent demandera également au NetworkHandler de créer un ClientHandler afin de se connecter au server TCP de la personne avec laquelle nous voulons communiquer.



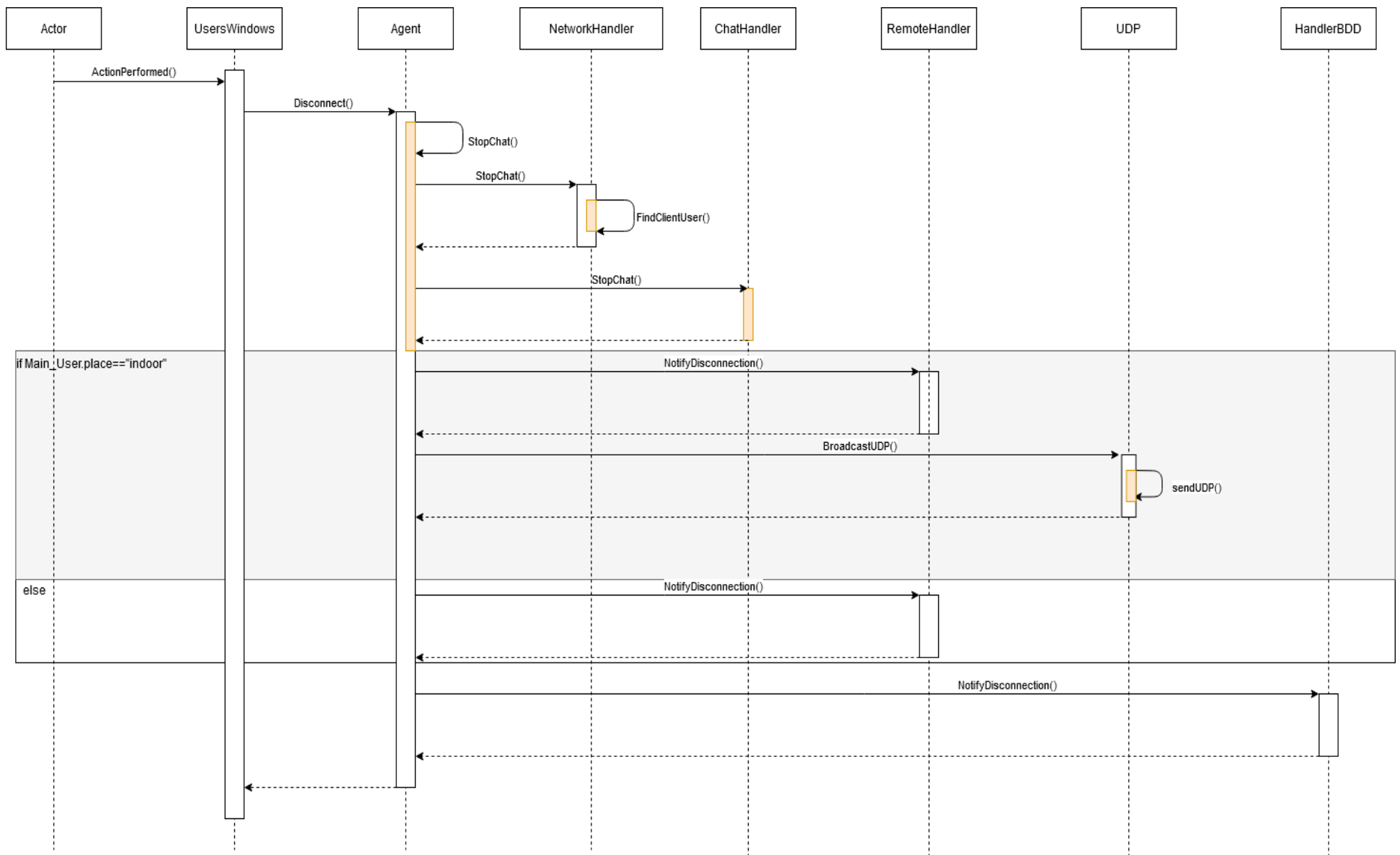
e- Envoie message

Lorsque l'utilisateur envoie un message un processus assez simple s'enclenche. Le message est créé par le ChatHandler grâce aux données rentrées par l'utilisateur dans l'interface de communication. Ce message est ensuite envoyé par le ChatHandler à son destinataire via le réseau puis enregistré dans la base de données grâce au HandlerBDD. Enfin, l'interface de communication est mise à jour pour afficher le message envoyé.

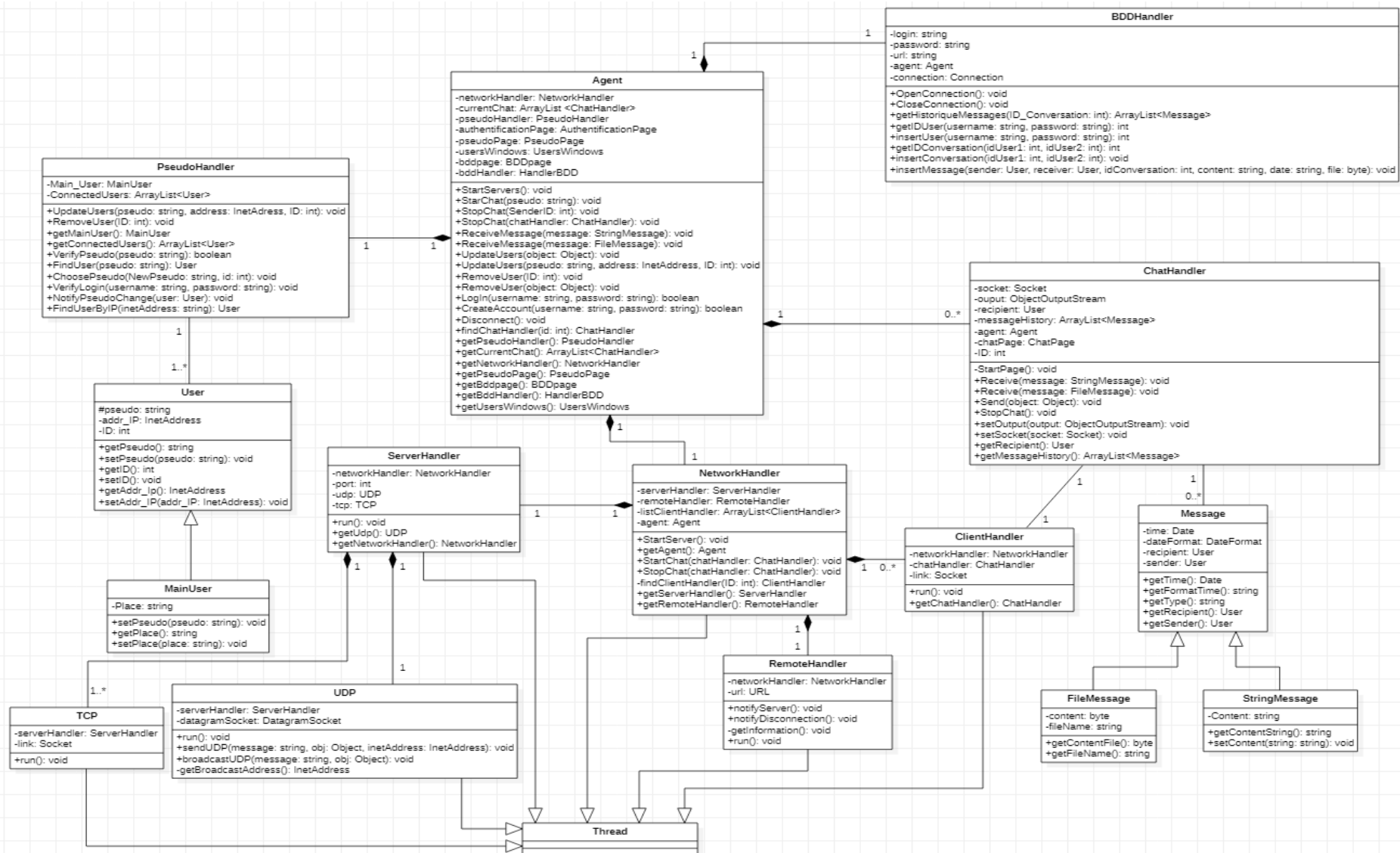


f- Déconnexion

Lorsque l'utilisateur décide de se déconnecter, une procédure d'arrêt de tous les chats encore en cours se met en route. Les utilisateurs sont donc prévenus de la fin du chat si nous étions encore en train de communiquer et toutes les connexions réseaux avec les autres utilisateurs sont interrompus. En dernier, un signal de broadcast sur le réseau local et/ou vers le serveur de présence est émis pour prévenir les autres utilisateurs de notre déconnexion.



3) Diagramme de classe



4) Diagramme base de données

