

Rapport de projet : Projet Java/UML

Diagrammes du projet

Rédigé par :

Lejeune Aurélia - Lacroix Raphaël

- Version du 20 novembre 2022 -

Table des matières

1	Identification des Acteurs	2
1.1	Acteurs primaires	2
1.2	Acteurs secondaires	2
2	Diagramme de Cas d'Utilisations	3
3	Diagramme de Classe	4
3.1	Schéma	4
3.2	Détails et typages	4
4	Diagrammes de Séquence	6
4.1	Affichage	6
4.2	Interactions utilisateurs	7
4.3	Gestion des utilisateurs	8
5	Diagramme de Structure Composite	9
6	Diagramme de déploiement	9
7	Base de Données	10
7.1	Bases de données locales	10
7.2	Serveur de gestion des ID	10
8	Maquette GUI	11
8.1	Maquette fonctionnelle	11
8.2	Maquette Visuelle	12

1 Identification des Acteurs

1.1 Acteurs primaires

L'application doit être installée par un administrateur (**Admin**) qui est un premier utilisateur primaire. L'application sera ensuite utilisée par l'utilisateur lambda (**User**) pour des sessions de discussion.

1.2 Acteurs secondaires

Il n'y a pas d'utilisateurs secondaires du système. En effet, l'utilisateur et l'administrateur sont les seuls à interagir avec le système.

2 Diagramme de Cas d'Utilisations



FIGURE 1 – Diagramme des Cas d'utilisation de l'application de discussion instantanée

Sur ce diagramme nous avons représenté deux utilisateurs communiquant ensemble. Une première catégorie affichage montre les divers items affichés sur l'écran. La catégorie gestion d'utilisateur permet d'afficher toutes les actions propres à la configuration du compte de l'utilisateur : sa connexion, sa déconnexion et son changement de pseudo. Enfin l'utilisateur peut interagir avec l'application en envoyant des messages ou encore en mettant l'application en barre de tâches.

En bas est représenté la configuration que l'administrateur réalise pour mettre en place le chat sur les ordinateurs de chaque utilisateur voulant utilisé le chat.

3 Diagramme de Classe

3.1 Schéma

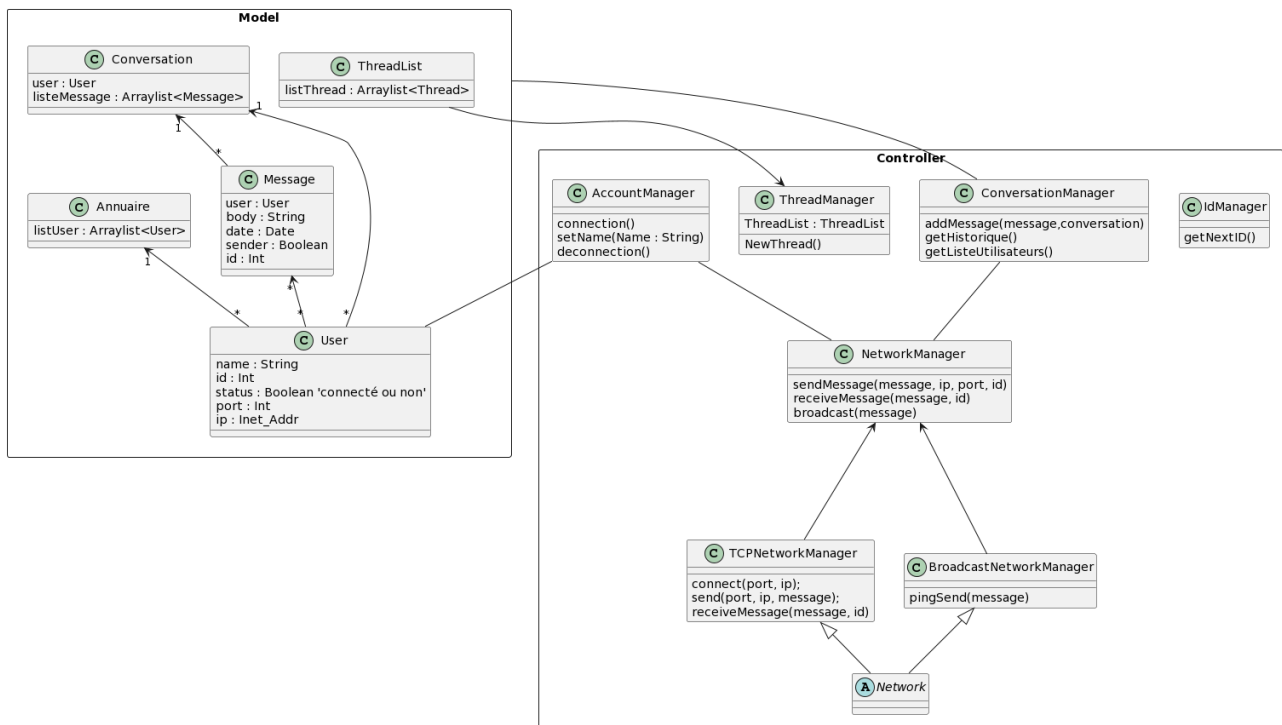


FIGURE 2 – Diagramme de Classe du Model et du Controller

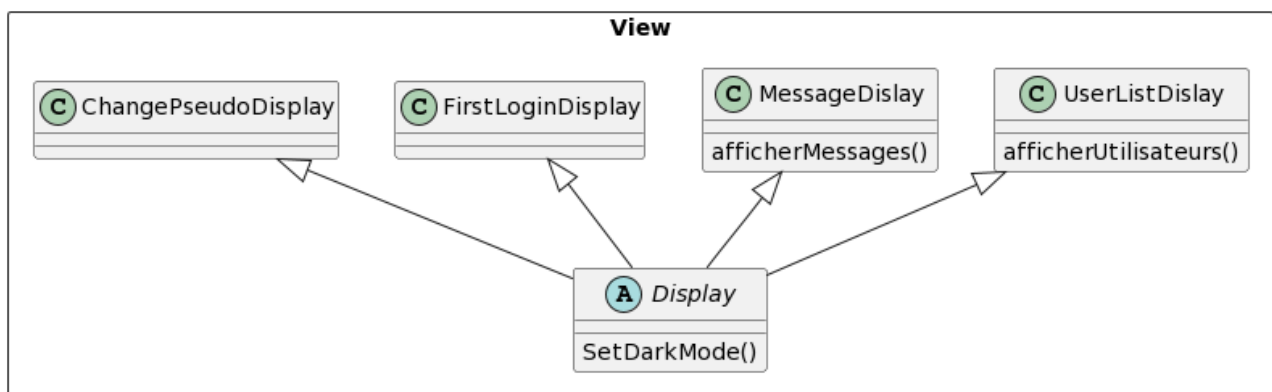


FIGURE 3 – Diagramme de la View

3.2 Détails et typages

Nous avons fait le choix pour notre chat, d'utiliser un serveur central qui s'occupera de gérer les ids déjà attribués aux utilisateurs. Ainsi lorsqu'un nouveau utilisateur se connecte pour la première fois au chat, le serveur central lui attribue un id disponible. Ce sera pour la suite son id de connexion qui servira d'identificateur pour les autres utilisateurs. Cet id est donc unique. (voir le scénario sur la diagramme de séquence n°3).

Chaque utilisateur possédera en local une base de donnée. Celle-ci contiendra diverses informations. Premièrement, la liste des utilisateurs connus avec leur statuts (1 pour connecté et 0 pour déconnecté), leur port de communication, leur adresse IP ainsi que leur pseudo actuel (celui-ci pouvant être changé). On y trouvera également les conversations, les messages.

A chaque nouvelle conversation, on crée un thread qui traite la conversation en cours, on aura donc une liste de thread en cours à retenir et un ThreadManager.

Enfin, nous avons fait le choix de créer une classe NetworkManager qui s'occupera de la gestion du réseau. Celle-ci fera appelle à TCP pour les messages mais aussi à UDP pour le broadcast du début (pour connaître les utilisateurs).

4 Diagrammes de Séquence

Nous avons réalisé les diagrammes de séquence de trois catégories de scénarios : l'affichage des messages et des utilisateurs, les interactions de l'utilisateur avec le chat et pour finir la connexion, déconnexion ainsi que le changement de pseudo.

4.1 Affichage

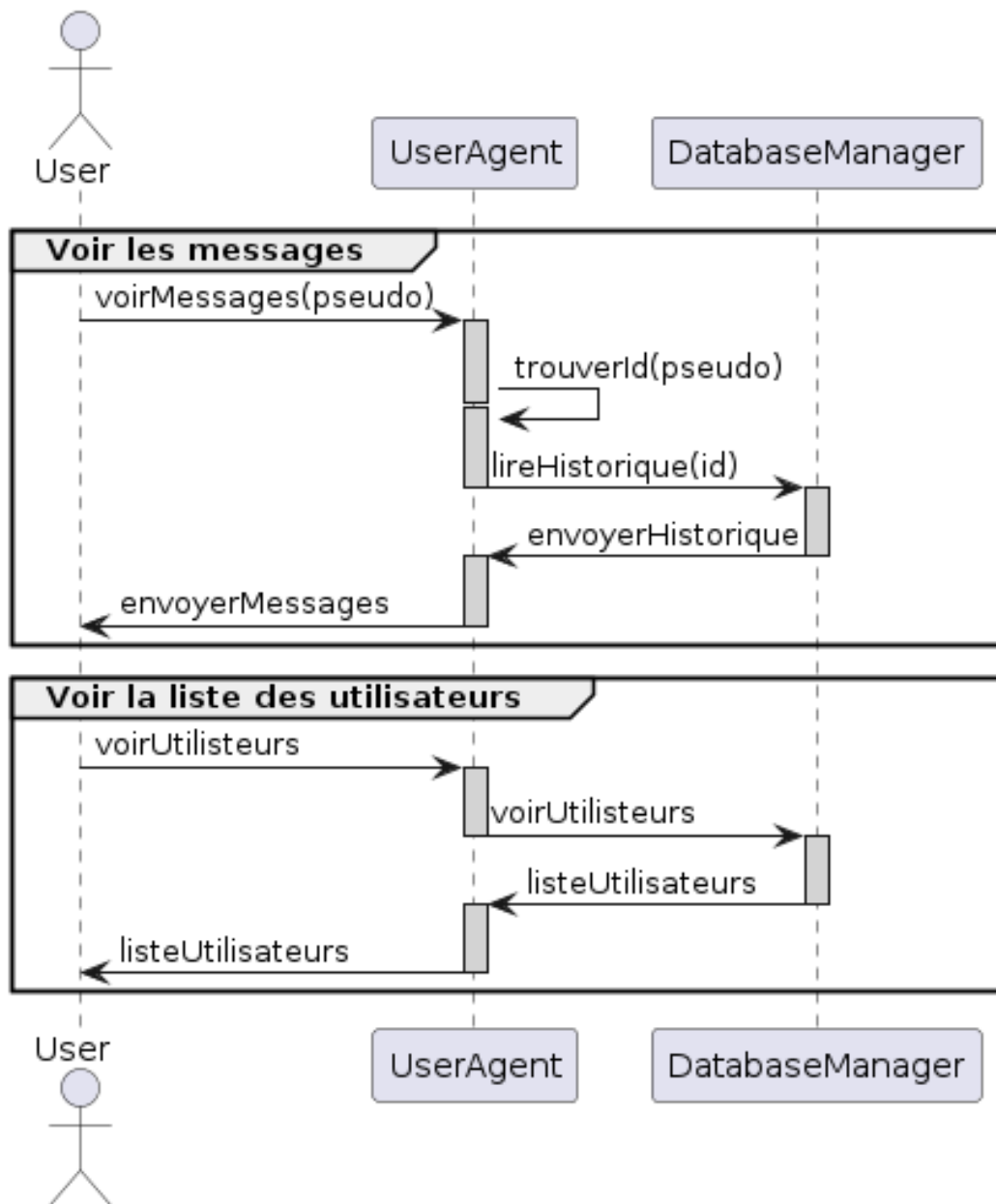


FIGURE 4 – Diagramme de Séquence de l'affichage

4.2 Interactions utilisateurs

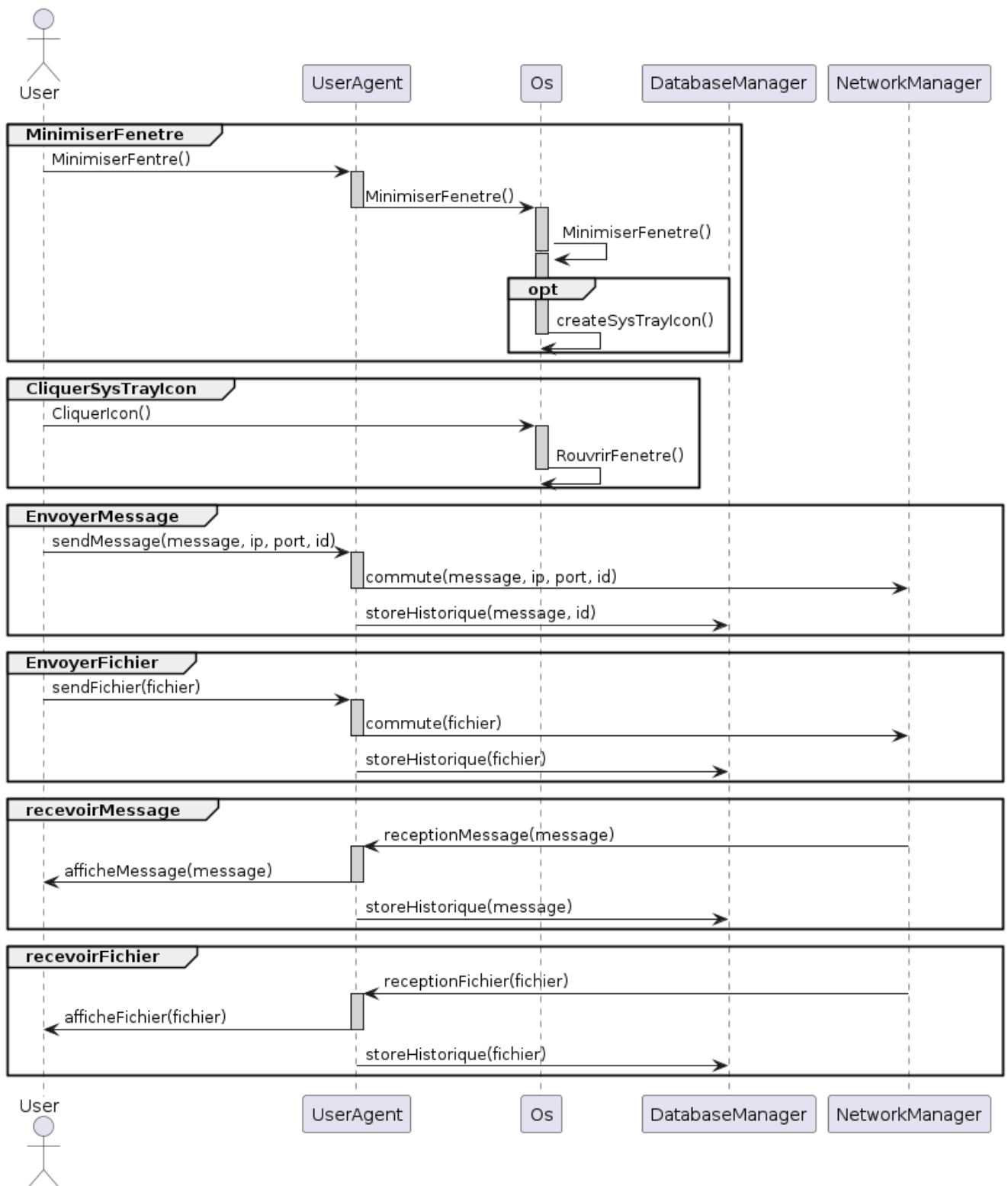


FIGURE 5 – Diagramme de Séquence des interactions utilisateur

4.3 Gestion des utilisateurs

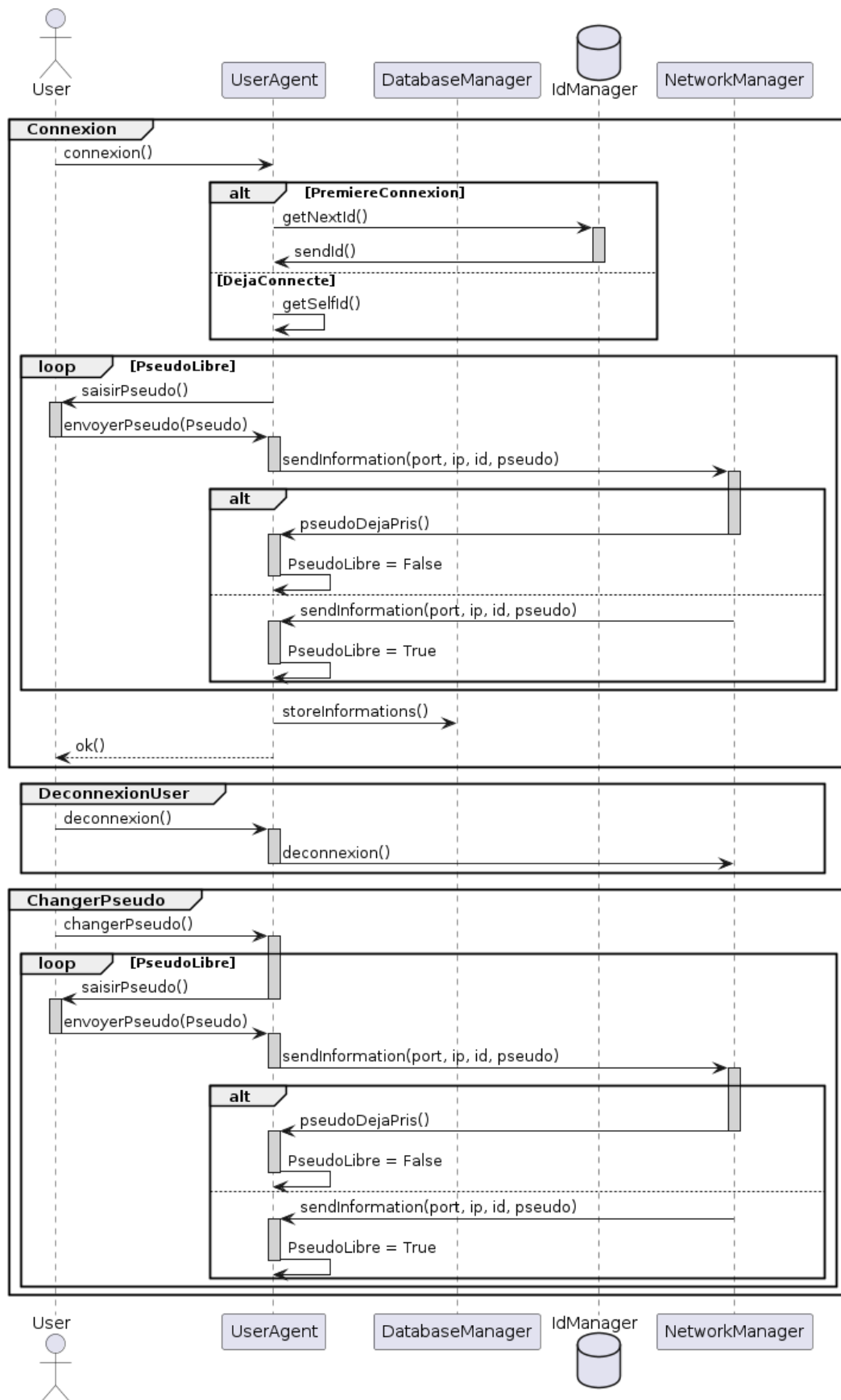


FIGURE 6 – Diagramme de séquence de la gestion des utilisateurs

5 Diagramme de Structure Composite

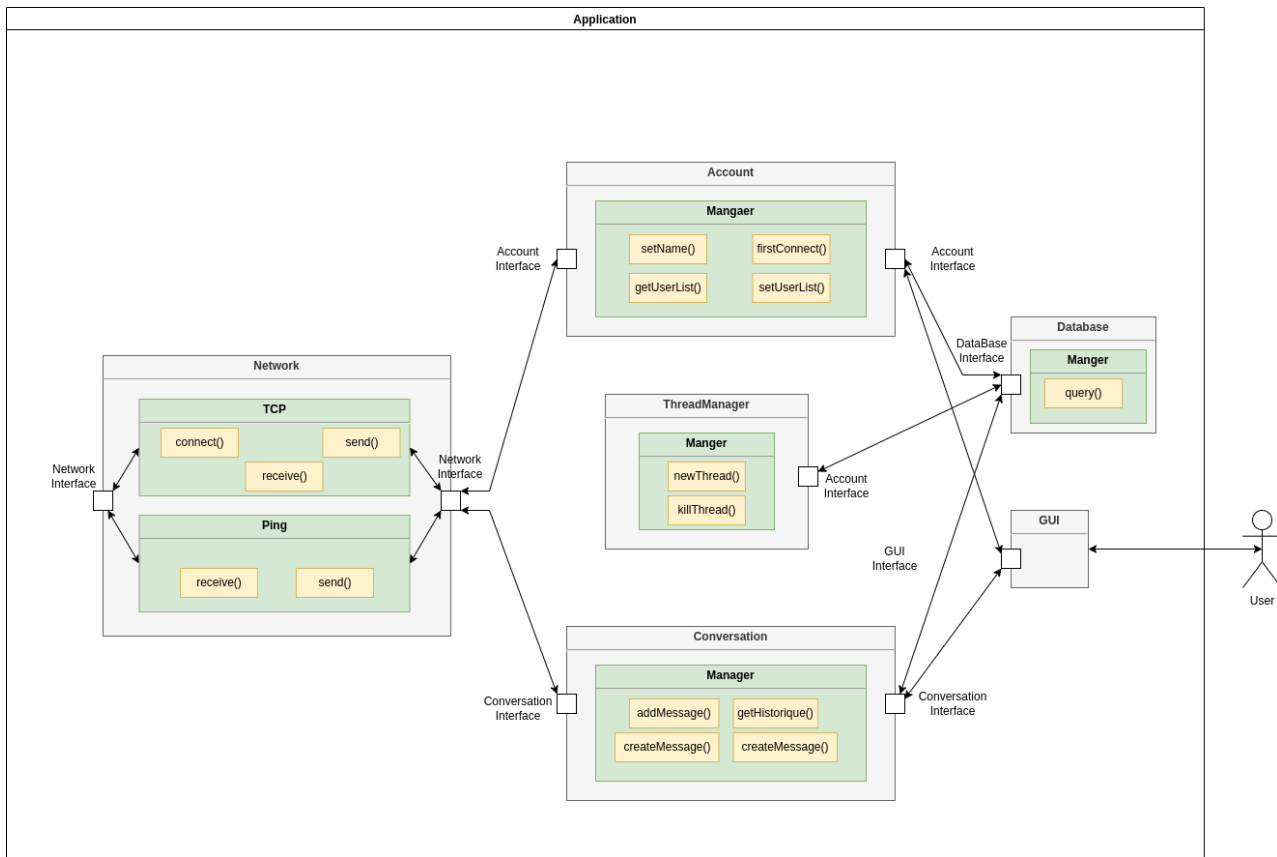


FIGURE 7 – Diagramme de Structure Composite

6 Diagramme de déploiement

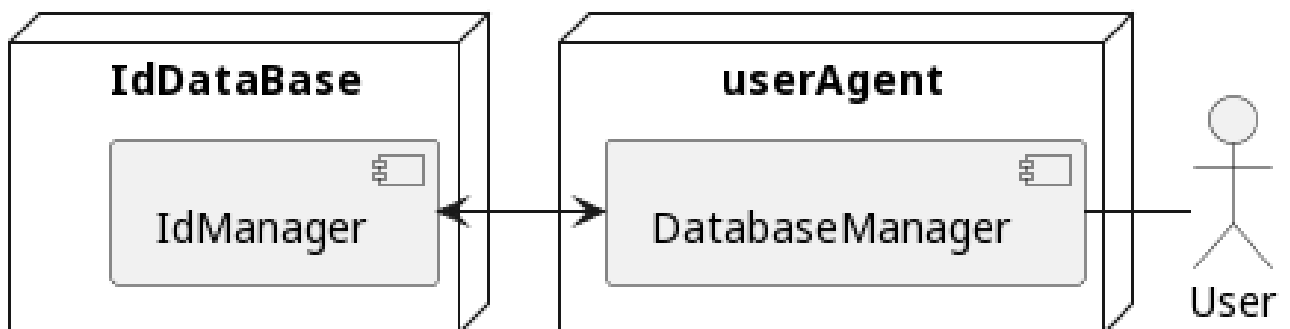


FIGURE 8 – Diagramme de Déploiement

7 Base de Données

7.1 Bases de données locales

En local l'application sera dotée de deux tables SQL, l'une pour les historiques de conversations et une autre pour stocker l'annuaire des utilisateurs.

id	username	ip	port	connected
...	...	...	...	...

FIGURE 9 – En-tête de la table de stockage des Conversations

idMessage	User	texte	sender	date/time	dfgh
...	...	...	...	...	...

FIGURE 10 – En-tête de la table de l'annuaire

7.2 Serveur de gestion des ID

Le serveur centralisé stockera la valeur de l'ID en cours. A chaque fois qu'un utilisateur souhaitera se connecter au chat pour la prochaine, il lui attribuera un id en lui donnant la valeur en cours puis incrémentera cette valeur de 1 pour le prochain utilisateur qui arrivera sur le chat. Ainsi, on s'assure que l'id est unique. Ajouter un id dans les caractéristiques de l'user permet de faire en sorte que si un utilisateur utilise un pseudo qui avait été pris précédemment par un autre utilisateur (par exemple john) , les autres utilisateurs le voient comme une nouvelle personne et non pas comme l'ancien john. Cela permet aussi de garantir que même si un user change de pseudo, les autres machines comprendront qu'il s'agit du même utilisateur.

8 Maquette GUI

8.1 Maquette fonctionnelle

Nous avons réalisé deux maquettes. La première est la maquette fonctionnelle. Elle montre les divers options qui seront présentes à l’affichage. La deuxième maquette montre le visuel que nous comptons faire pour le chat.

A la place des avatars, nous mettrons des avatars générés à partir du nom de l’utilisateur.

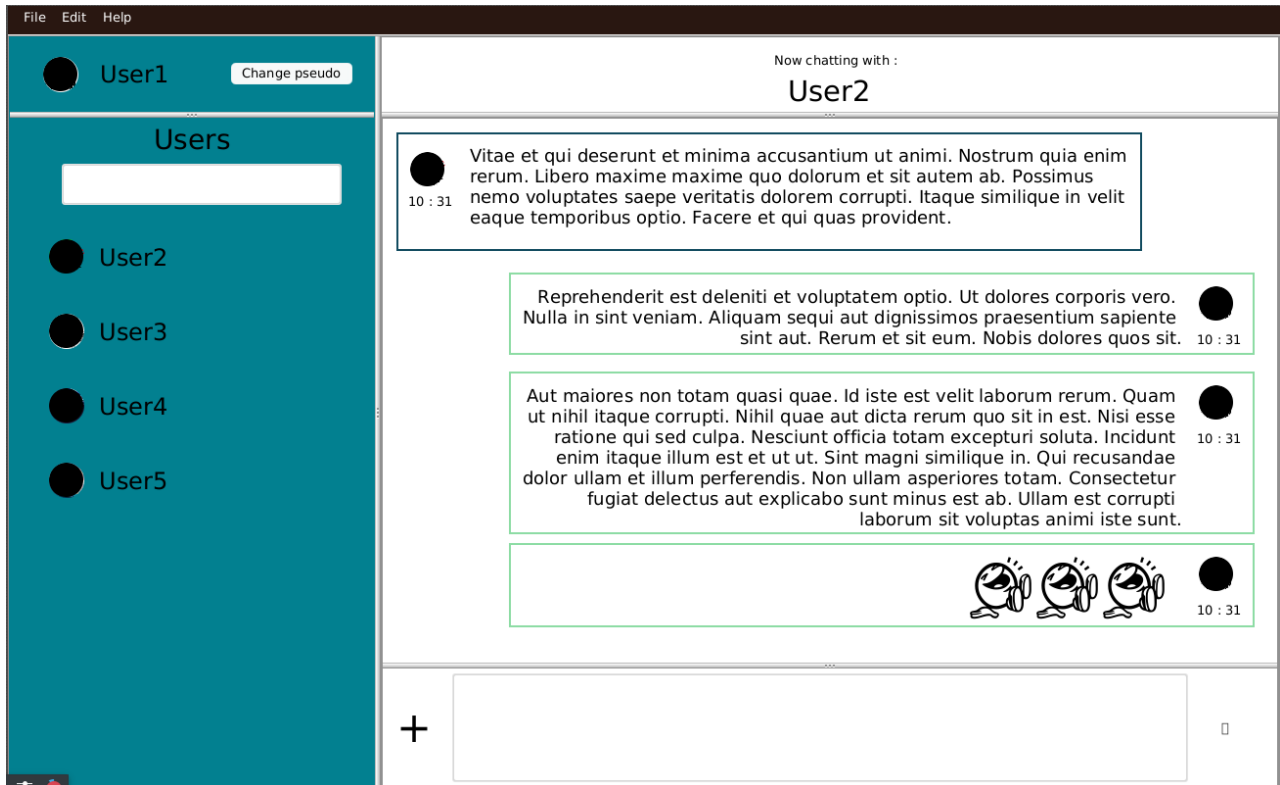


FIGURE 11 – Maquette fonctionnelle de l’interface principal

8.2 Maquette Visuelle



FIGURE 12 – Maquette visuelle de l'interface principal

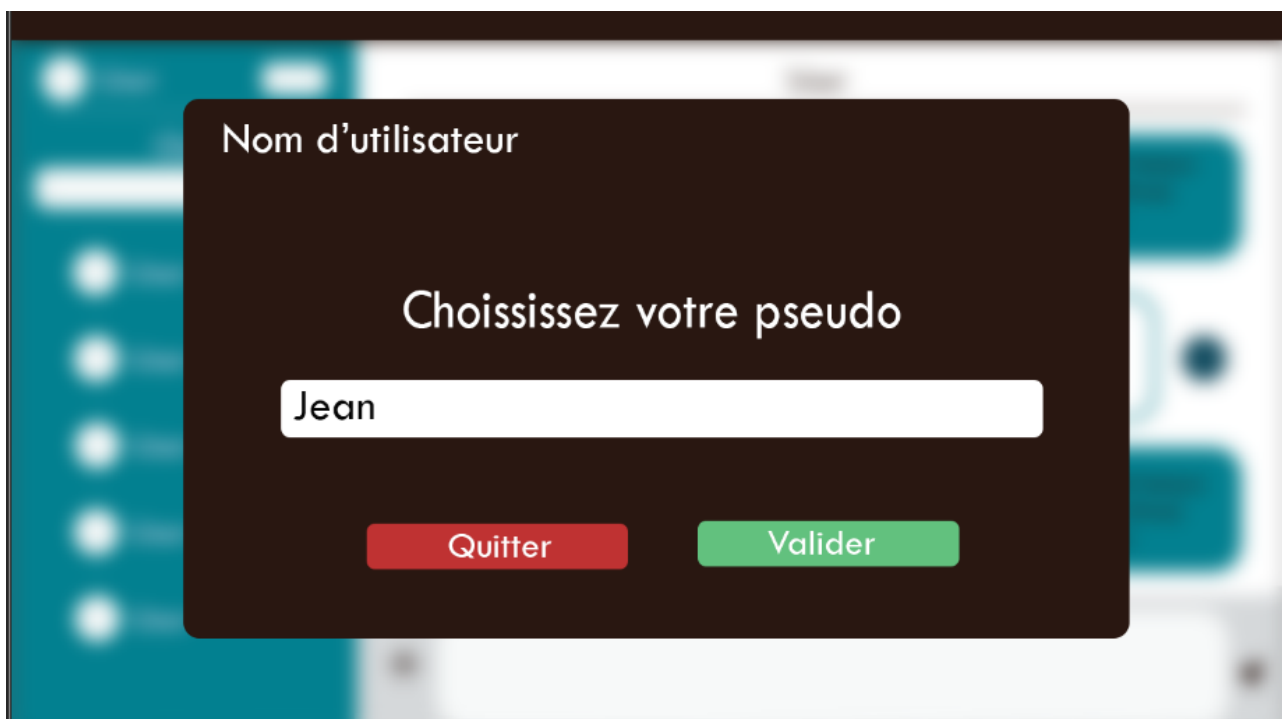


FIGURE 13 – Maquette visuelle de l'écran de connexion/changement de pseudo

INSA Toulouse
135, Avenue de Rangueil
31077 Toulouse Cedex 4 - France
www.insa-toulouse.fr



MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE