INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Report INSA Chat

AGBETI-MESSAN Komlan Epe-Nsin Merveilles
Telefo Tankeu Smith Walter

Année 2022 – 2023
Promotion n°57
4IR

Summary

I. **Introduction**

II. **Needs Analysis**

III. **Design**

- A. Proposed architectures
- B. Chosen architecture
- C. Data model
- D. User Interface Design

IV. **Development**

- A. Description of the developed features
- B. Description of the tools and technologies used
- C. Testing Report
- D. Description of encountered difficulties and solutions

V. **Deployment**

- A. Description of the deployment procedure
- B. Description of maintenance and support
- C. Installation

VI. **Appendices**

- A. User Manual
  - a. Introduction
  - b. Getting Started
  - c. Features
- B. UML Diagrams
  - a. Use case diagram
  - b. Sequence diagram
    - i. Connection
    - ii. Send a message
    - iii. Disconnection
  - c. Composite diagram
  - d. Class diagram

VII. **Conclusion**

I. Introduction

This report covers the development of a communication system for businesses. The system uses an agent to facilitate the exchange of messages and images between users. It is intended for organizations representing a group of people who share a common interest and collaborate together. The report covers the project's needs analysis, design, development, deployment, and conclusion. It also includes future prospects and recommendations.

II. Needs Analysis

The system described is a communication system for businesses that uses an agent to facilitate the exchange of messages and images, between users. It is intended for organizations . The system is used as a service for users and administrators responsible for managing and deploying it. It is capable of functioning in any region of the world with a network connection, and can be deployed quickly without disrupting other existing systems. The system will first be demonstrated in simulated conditions to prove its operational capabilities, then will be put into use based on customer demand. Finally, the system could be removed when it is no longer needed. It is deployed without restrictions on the operating system and offers basic features accessible through several user interfaces .

III. Design

A. Proposed architectures

By thinking just about a chat system for an organization we could come out with 3 possible architecture:

- Client Server
- A Peer to Peer architecture
- A Three-Tier architecture

B. Chosen architecture

But according to the client's needs we must get rid of the Client Server architecture . Now we have to choose between the Peer to Peer architecture and the Client Server architecture . In fact the best one would be the Three-Tier architecture but we didn't choose it because it is more complicated to handle and we didn't have enough time to do it so we chose the Peer to Peer architecture.

C. Data model

The data model used in this system is a distributed architecture where each user has its own internal database. The data is organized in one table, in each user's database we have the table userMessages which is used to store the messages between the users. This table has 5 fields :

- **idMessage**: used to number the messages
- **idChater** : used to know who sent or received the message
- **type :** used to know if the message is received(type = 0) or sent (type = 1)

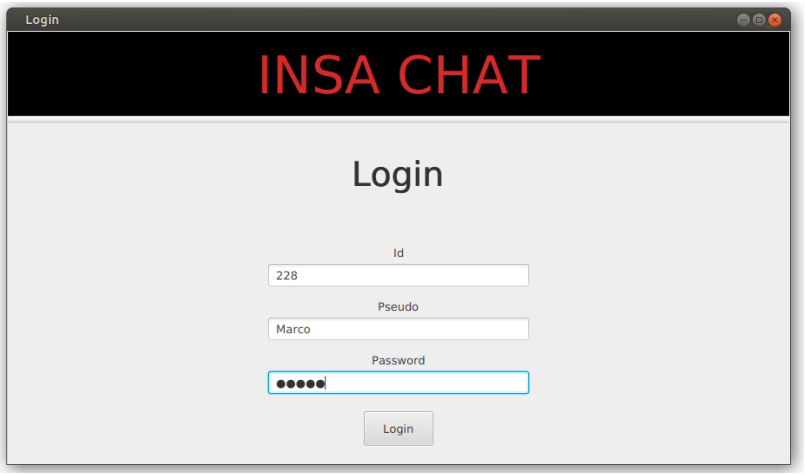
- **message :** content the text message
- **time :** used to know at what time the message was sent or received

D. User Interface Design

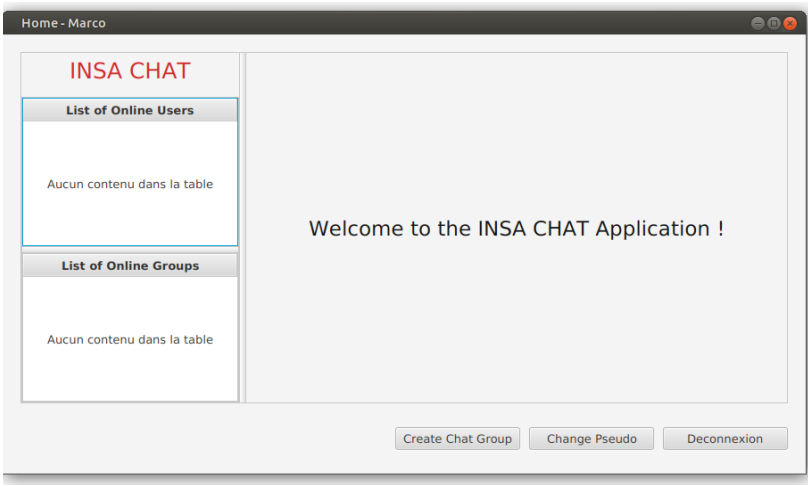
- **ICON** : The User Interface can be accessed by clicking on the icon located on your desktop if you're in **Windows**

or by executing the **./InsaChat** if you're using **Linux**.

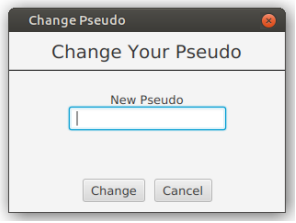
- **LOGIN** : Before you can use the User Interface, you'll need to log in with your **id**, choose a **pseudo** and a **password** on the login page. (The **id** and the **password** are chosen on the first login)



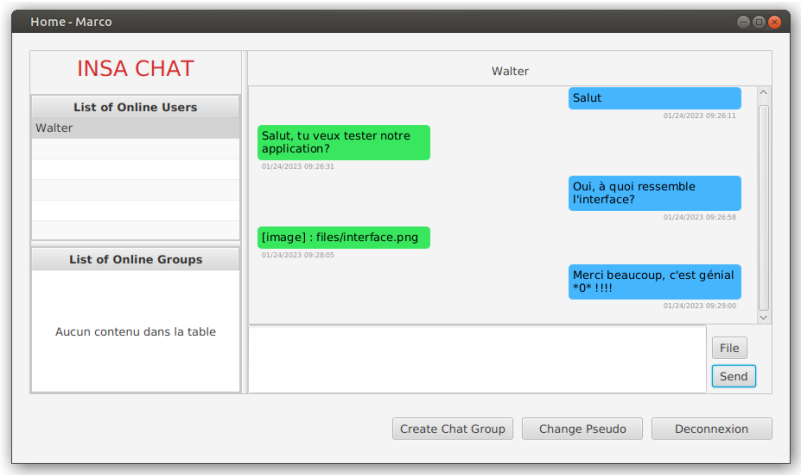
- **HOME** : After logging in, you'll be taken to the homepage, which is the main hub for using the User Interface.



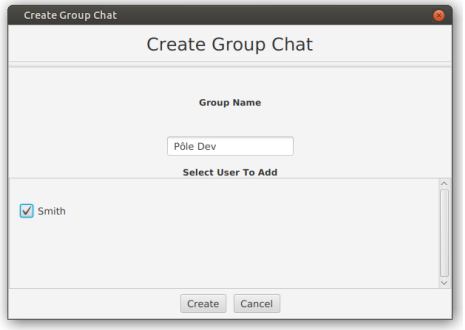
- **CHANGE PSEUDO** : The 'Change Pseudo' feature allows you to personalize your chat experience by updating your name when clicking on the **Change Pseudo** button.



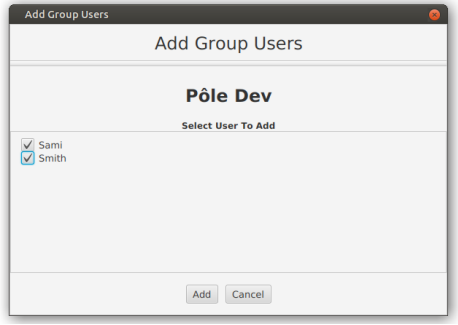
- **NORMAL MESSAGES** : When we have a new member in the System we can click on his name in the left panel to start a communication. The communication Interface looks like this. We can share **Texts** and **Images**.



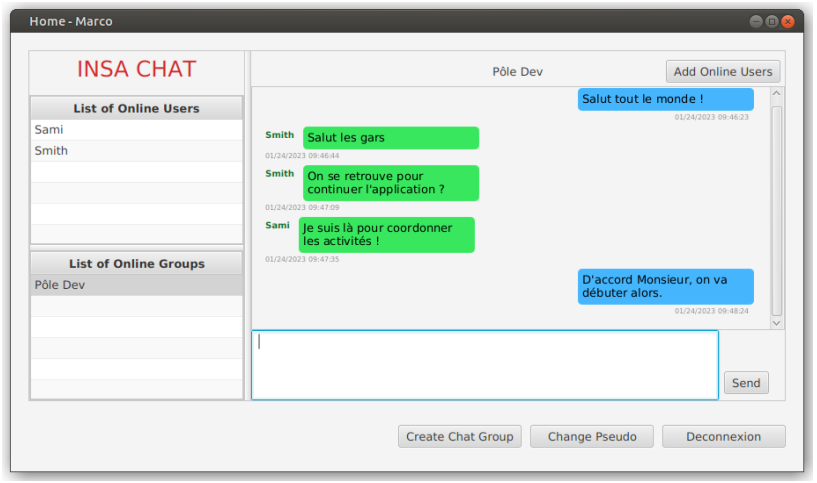
- **CREATE GROUP** : The '**Create Chat Group**' button allows you to create a new chat group.

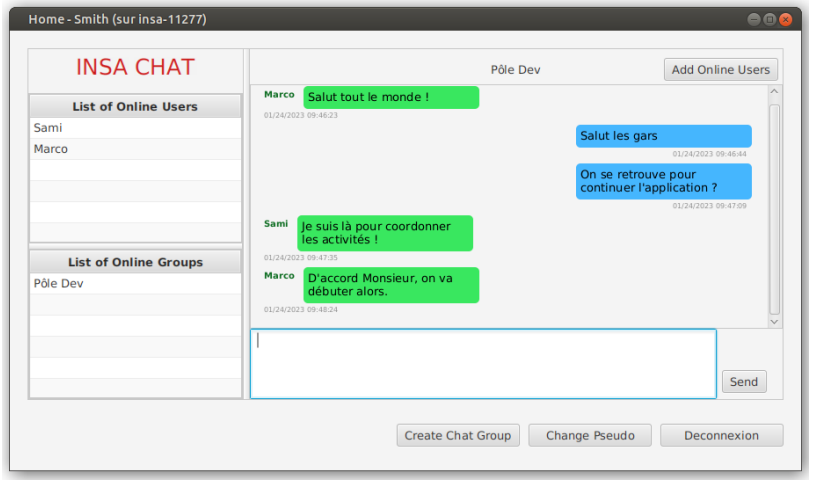


- **ADD USER** : The '**Add Online User**' button allows you to invite other users to join the current chat group.

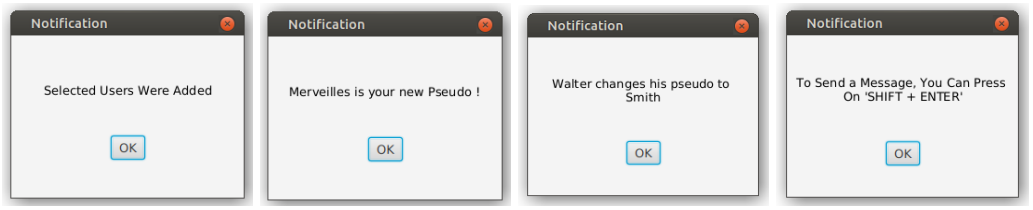


- **GROUP MESSAGES** : We can send messages to the Group. The communication Interface looks like this. We can share **Texts**.

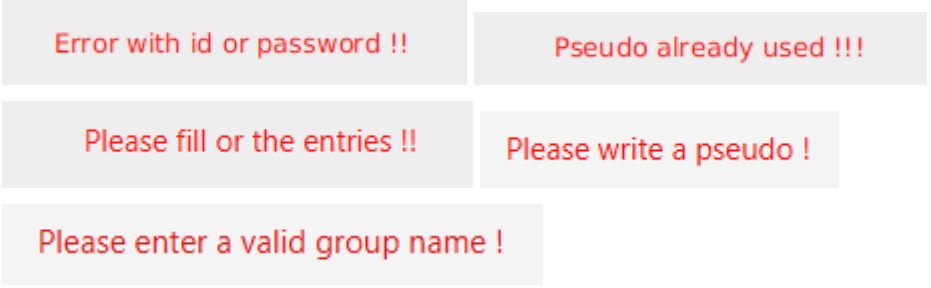




- **NOTIFICATIONS** : The User Interface has a built-in notification system that alerts you to important updates and events.



- **ERROR MESSAGES** : The User Interface has a personalized error system that provides clear and actionable error messages, making it easy to understand and fix the problem.



IV. Development

A. Description of the developed features

- **Authentication and Pseudo management** : They are managed by the UDP socket . This socket manages all the technical messages like *newUser, newPseudo, Disconnecting, newGroup, JoinTheChat.*
- **Instant messaging** : It is managed by a TCP server which creates a client after receiving a new chat request by another user. Once the TCP Client is created it handles a single conversation .
- **History management** : When two users start a chat session their previous messages are loaded from their respective databases .
- **Send of images** : When a user sends a file to another one, the file is stored in a folder named "**files"** and the name of the image is stored in the database.
- **User Interface** :The User Interface was developed using JavaFX. To have an overall view please refer to **section D** of **III. Design**.
- **Database** : The database was developed using **sqlite**. To have an overall view of the database organization please refer to **section C** of **III. Design**.

B. Description of the tools and technologies used

- **Java** : Java is a popular, object-oriented programming language that is known for its "write once, run anywhere" capability due to Java Virtual Machine (JVM). It is commonly used for building enterprise-level applications and Android mobile apps.
- **SQLite** : it is a lightweight, open-source relational database management system that is often used for embedded systems and local data storage. It is written in C and does not require a separate server process or system to run, making it easy to integrate into a variety of applications.
- **JavaFX** : JavaFX is a Java-based software platform for creating and delivering desktop applications, as well as rich internet applications (RIAs) that can run across a wide variety of devices. It is used for building media-rich client applications and it is included in the Java SE 8 and later. It provides a rich set of UI controls and a modern graphics stack for creating visually appealing and responsive applications.
- **Socket** : A socket is an endpoint for sending or receiving data across a computer network. It is a combination of an IP address and a port number and it provides a common interface for network communication in many operating systems. Sockets can be used for building. network protocols, creating networked applications and for direct communication between nodes in a network. It can be used with many different protocols like TCP,UDP and others
- **TLS / SSL** : They are cryptographic protocols that provide secure communication over a network . The use of TLS and SSl was implicit in the use of TCP and UDP sockets .
- **Jira** : Jira is a software development project management tool that allows teams to track progress and manage tasks. It is popular among Agile teams and includes features such as bug tracking, task management and custom workflows.

- **Jenkins** : Jenkins is an open-source automation server that automates the building, testing, and deployment of software using plugins and pipeline as code, it is widely used for Continuous integration and Continuous Delivery (CI/CD).
- **Maven** : It is a Java build tool that manages project structure, dependencies, build, and test. It uses an XML file for easy management of the build process and allows for easy integration with development tools and frameworks.
- **Eclipse** : It is an open-source integrated development environment (IDE) widely used for Java development. It provides a wide range of tools for development such as code editing, debugging, and profiling.
- **Visual Studio code** : It is a free, open-source code editor developed by Microsoft. It is a lightweight but powerful source code editor that runs on Windows, Linux, and macOS. It supports multiple languages, and has built-in debugging, version control and Git integration.
- **Github** : It is a web-based platform for version control and collaboration that uses Git. It allows developers to host and review code, manage projects, and build software.

V. Deployment

A. Description of the deployment procedure

Pre-deployment checks: Before deploying the app, we checked some configurations and ensured that Jenkins and all necessary plugins are installed and configured.

Building the app: Jenkins is used to automate the build process of our app. It pulls the source code from GitHub when a push is made, and compiles it. Here the Runnable Jar is created.

Deployment configuration: We configured Launch4j to build the .exe for windows.

Deployment: Once the app is built, we created the installable exe with InnoSetup for windows. We gave executable permission and created the .sh to be executed also in Linux.

Post-deployment checks: After the app is deployed, we make manual tests to check that the app is accessible, that all necessary services are running, and that there are no errors or issues.

B. Installation

To install the system, you will first need to install the agent on the workstations of the people who will be interacting. To do this, follow the instructions below:
Run the installation file on each workstation where the agent is to be installed.
Follow the on-screen instructions to install the agent.
Once installation is complete, configure the agent settings on each workstation as needed.

C. Usage

Once installation is complete, the system is ready to be used. Users can begin using the agent to exchange messages and images with each other.

VI. Appendices

A. User Manual

a. Introduction

Welcome to the User Manual for the distributed, interactive, multi-user, real-time chat system. This system is designed to increase the efficiency of communication within teams and groups in your company. With this system, users can coordinate through text messages and images sharing. The system is dynamic and adapts to changes in user connections in real-time.

b. Getting Started

1. To start using the system, you will need to log in using your company credentials.
2. Once you are logged in, you will be taken to the main interface of the system. Here, you will see a list of online users and the list of groups you have been invited to . You can join a group by clicking on it.
3. Once you have joined a group, you will be able to send and receive messages with other users in the group.
4. The system also allows you to create new groups by clicking on the "Create Group" button. You can invite other users to join the group by selecting their pseudo.
5. The system also allows you to change your pseudo whenever you want all the users will be notified of your change.

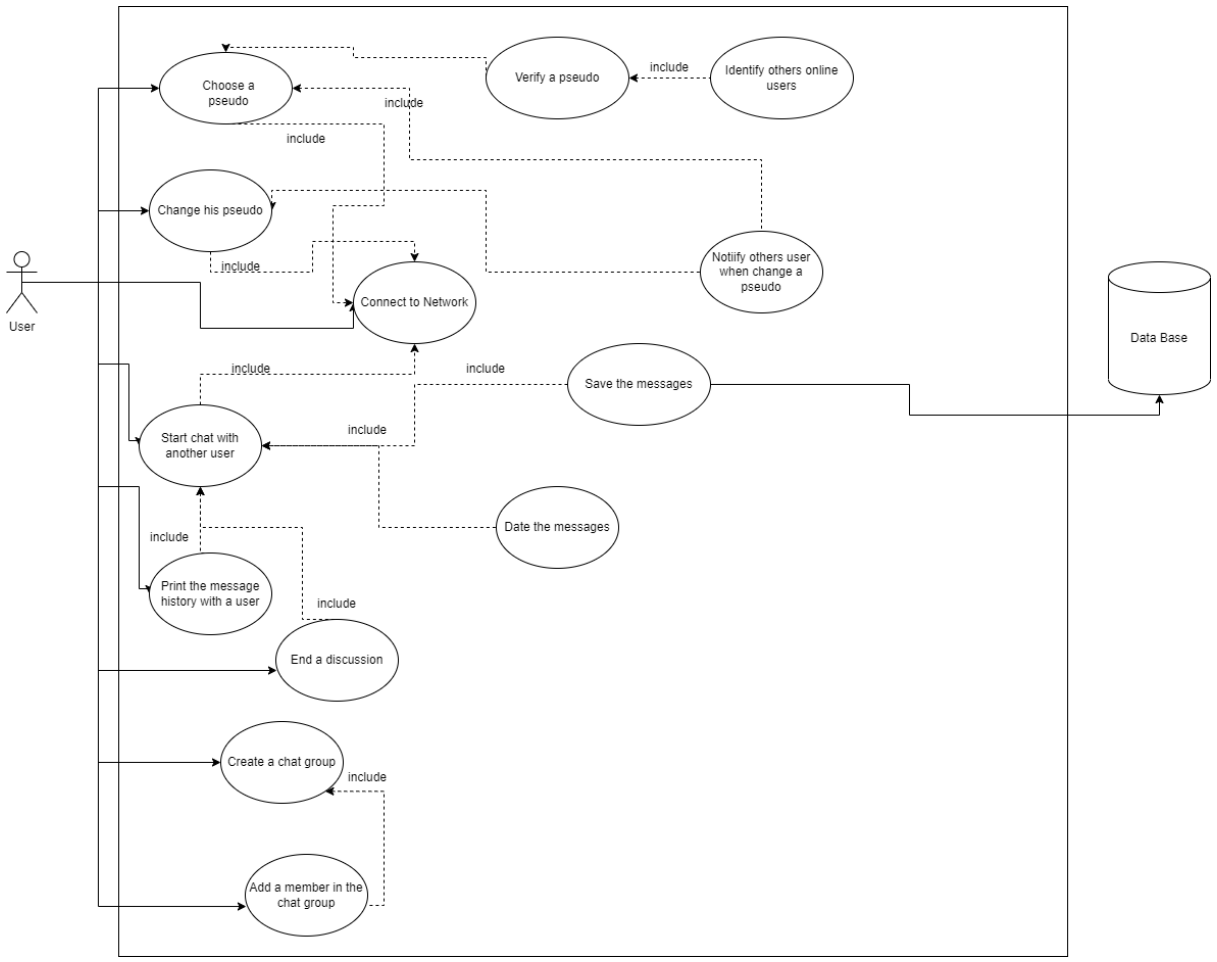
If you need to leave the system, you can do so by clicking on the "Disconnexion" button**.**

c. Features

Refers to the **Part IV - A**

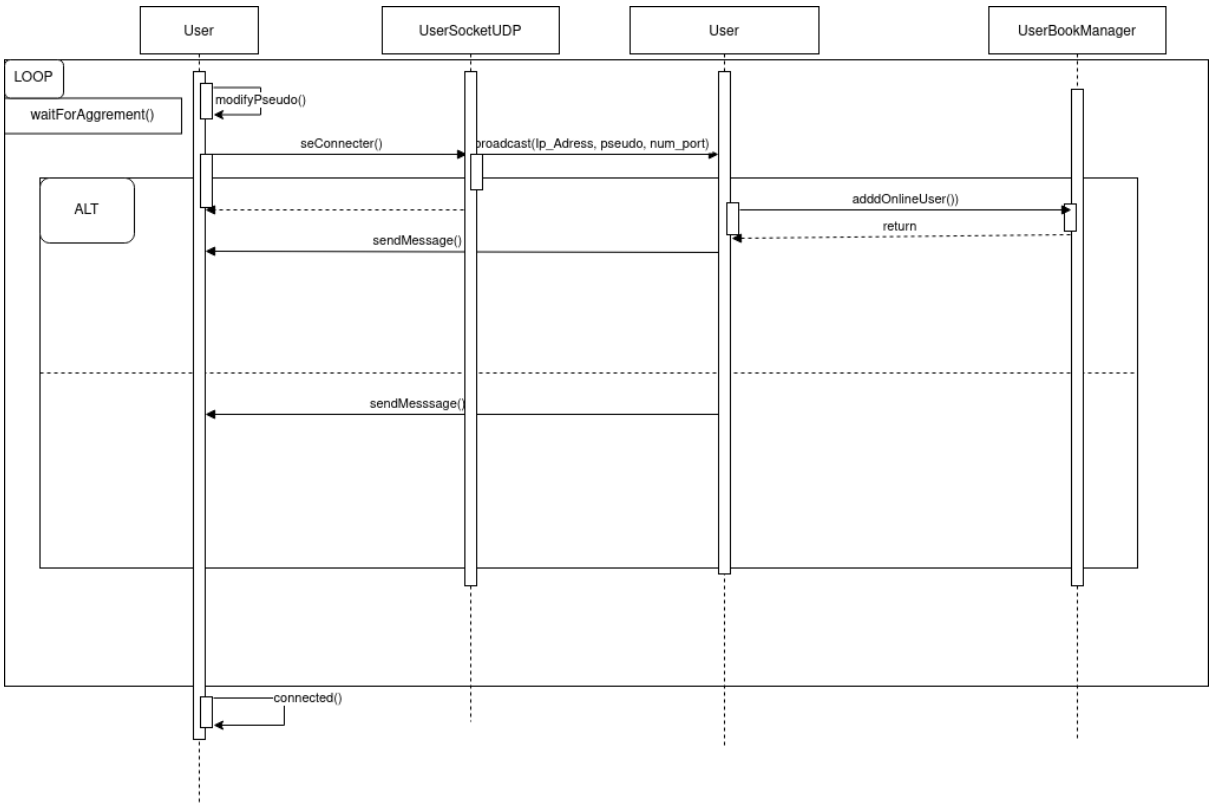
B. UML Diagrams

a. Use case diagram

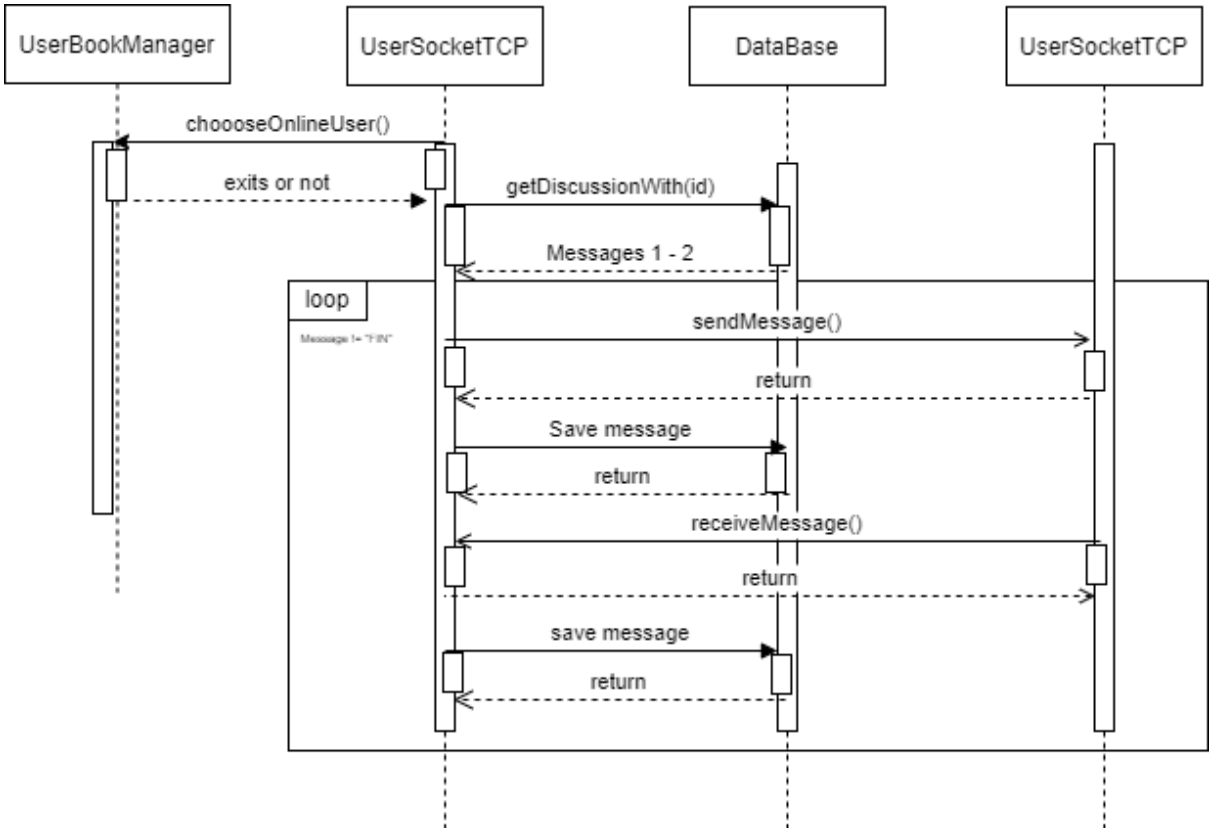


b. Sequence diagram

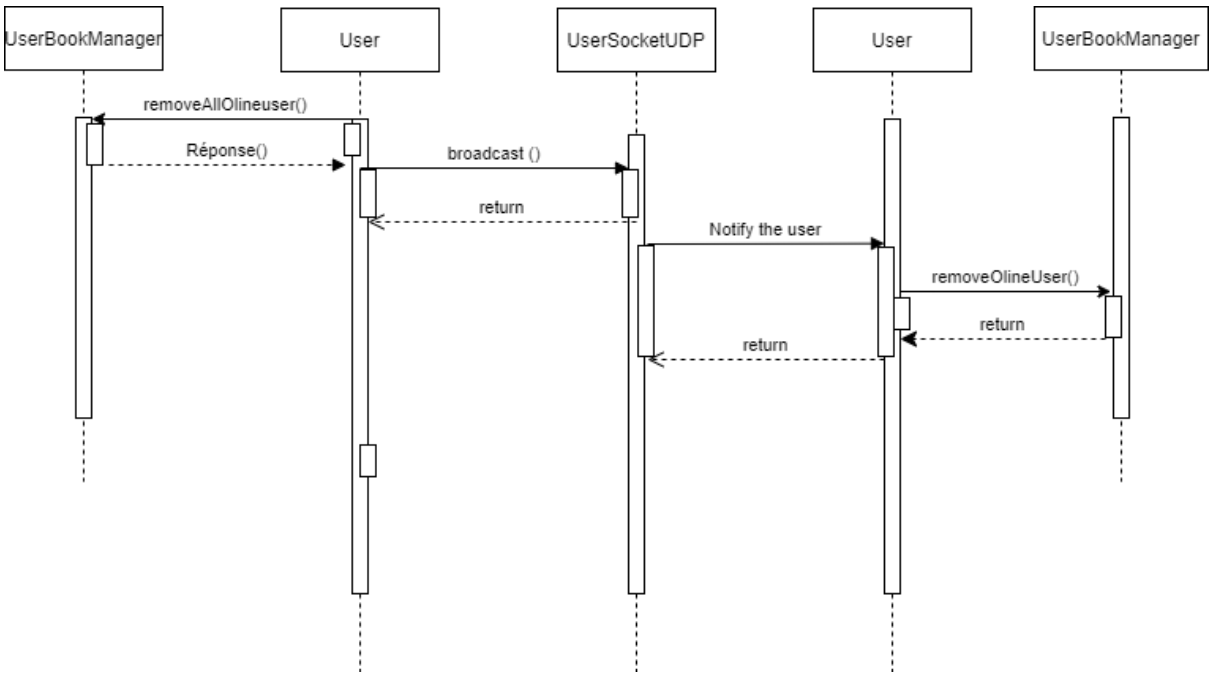
i. Connection



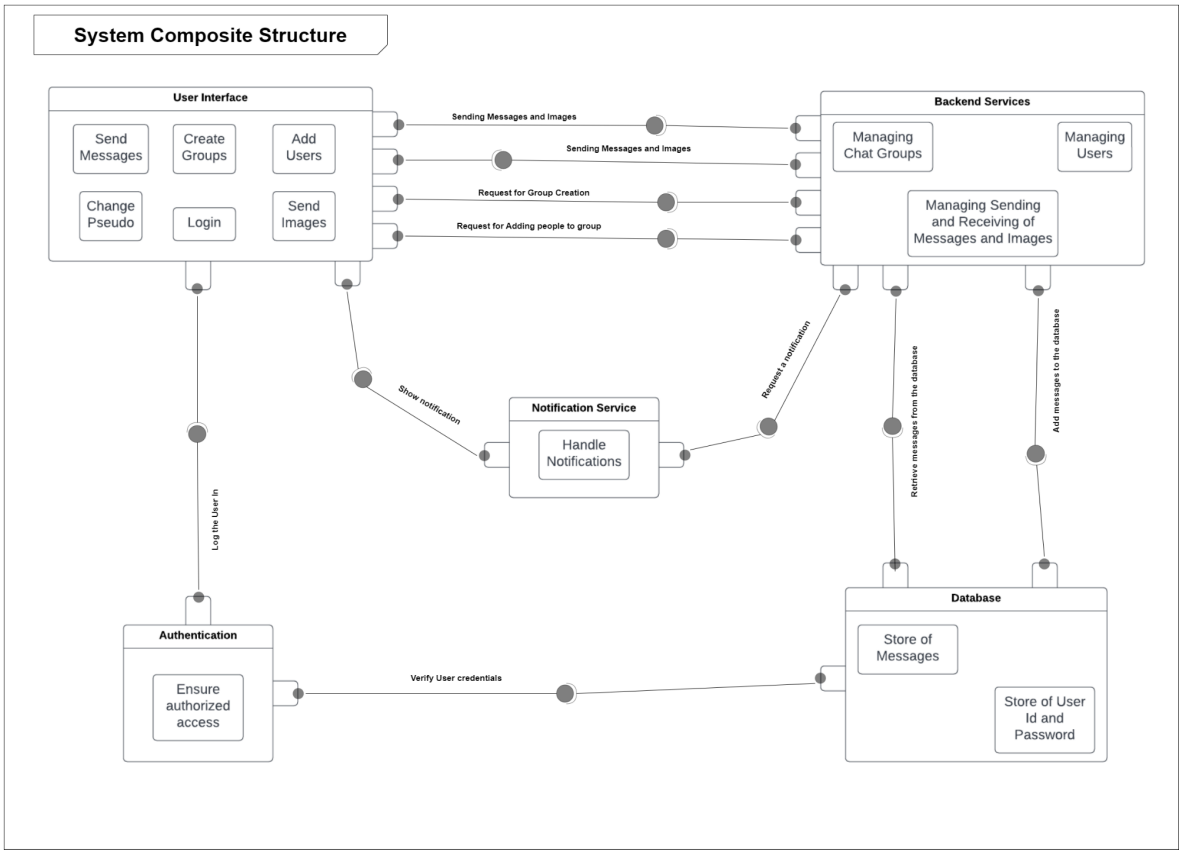
ii. Send a message



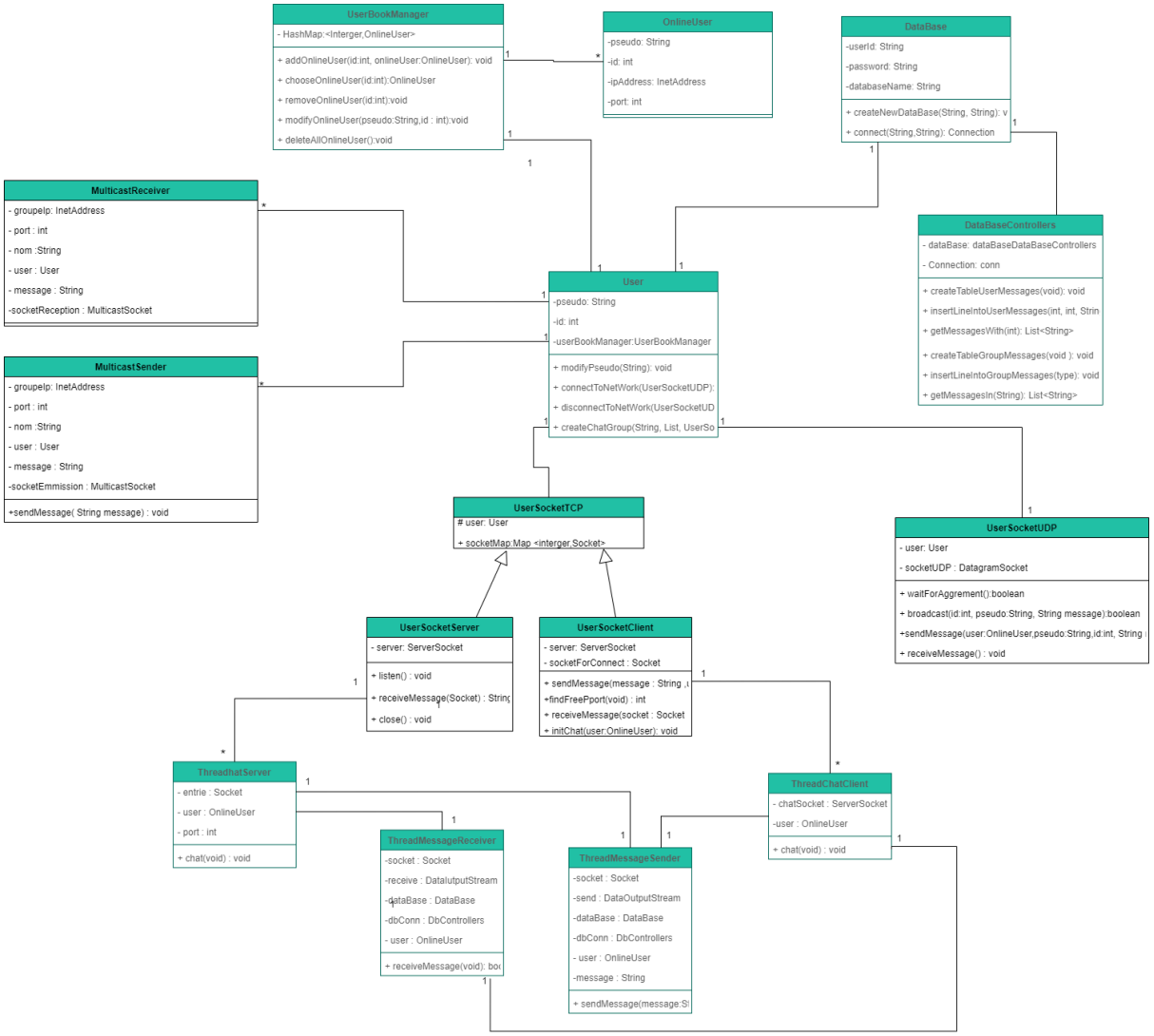
iii. Disconnection



c. Composite diagram



d. Class diagram



VII. Conclusion

The distributed, interactive, multi-user, real-time chat system is a powerful tool for increasing the efficiency of communication within your company. With its ability to send and receive messages, images, group creation and leave features, this system is sure to meet the needs of any team or group.

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

Université Fédérale Toulouse Midi-Pyrénées
Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE