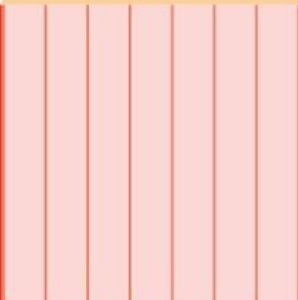
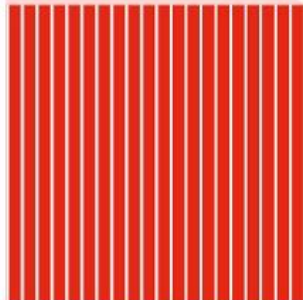
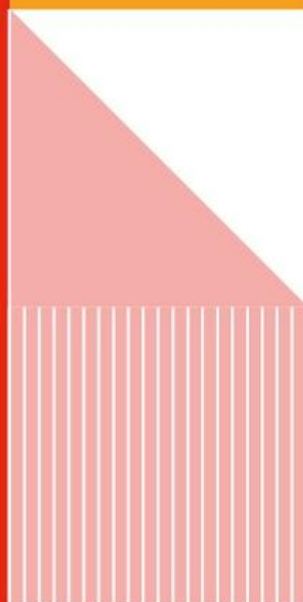


## Rapport du projet de création d'une application de clavardage

Réalisé par :

MAILLARD Etienne

SLAOUI Othmane



<b>1) Introduction :</b>	<b>2</b>
<b>2) Conception et diagrammes réalisés:</b>	<b>3</b>
<b>3) Architecture du système et choix technologiques:</b>	<b>10</b>
<b>4) Procédures d'évaluation et de tests:</b>	<b>10</b>
<b>5) Procédure d'installation et de déploiement:</b>	<b>13</b>
<b>6) Manuel d'utilisation simplifié :</b>	<b>14</b>
<b>7) Conclusion :</b>	<b>17</b>

# 1)Introduction :

Ce rapport présente les différentes étapes de création d'une application de clavardage en Java. L'objectif de cette application est de permettre aux différents utilisateurs présents sur un même réseau local de pouvoir échanger des messages avec les personnes connectées en ayant chacun un pseudo. Elle a été conçue pour répondre aux besoins des différents membres d'une entreprise afin de faciliter la communication entre ces derniers au sein du réseau.

Au cours de ce rapport, nous décrivons les différentes étapes de développement, depuis la conception jusqu'à l'utilisation finale. Nous présenterons les diagrammes UML réalisés lors de la conception, ainsi que les choix technologiques effectués pour la réalisation de l'application. Nous décrivons également les tests utilisés pour vérifier le bon fonctionnement de l'application. Enfin, nous présenterons la procédure d'installation et de déploiement ainsi qu'un manuel d'utilisation simplifié pour aider les utilisateurs à utiliser l'application.

## 2) Conception et diagrammes réalisés:

Afin de définir les besoins et les fonctionnalités de l'application, nous avons réalisé plusieurs diagrammes UML. Les principaux diagrammes utilisés lors de la conception de l'application sont :

- Diagrammes de cas d'utilisation : qui ont permis de décrire les actions réalisables par les utilisateurs de l'application.
- Diagramme de classe : qui a permis de définir la structure des classes et des interfaces de l'application.
- Diagrammes de séquence : qui ont permis de décrire les interactions entre les classes et les objets de l'application.

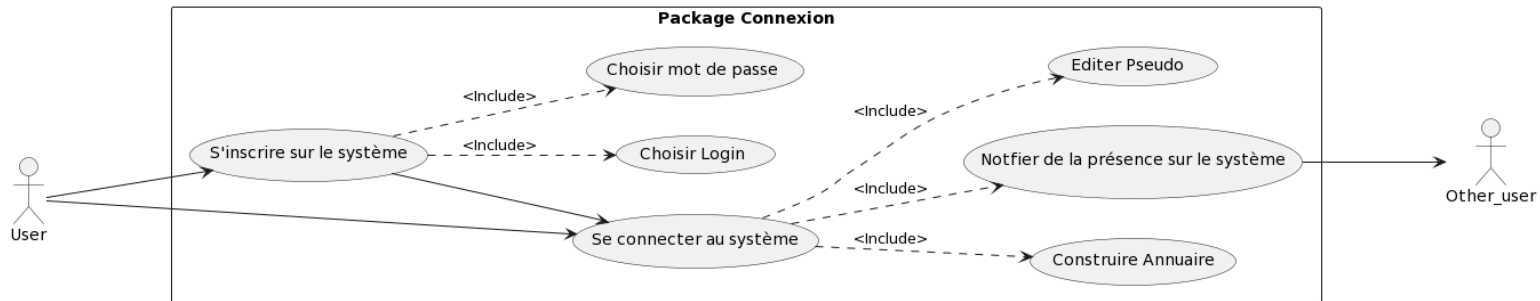
Nous avons commencé par l'identification des acteurs :

### Acteurs principaux :

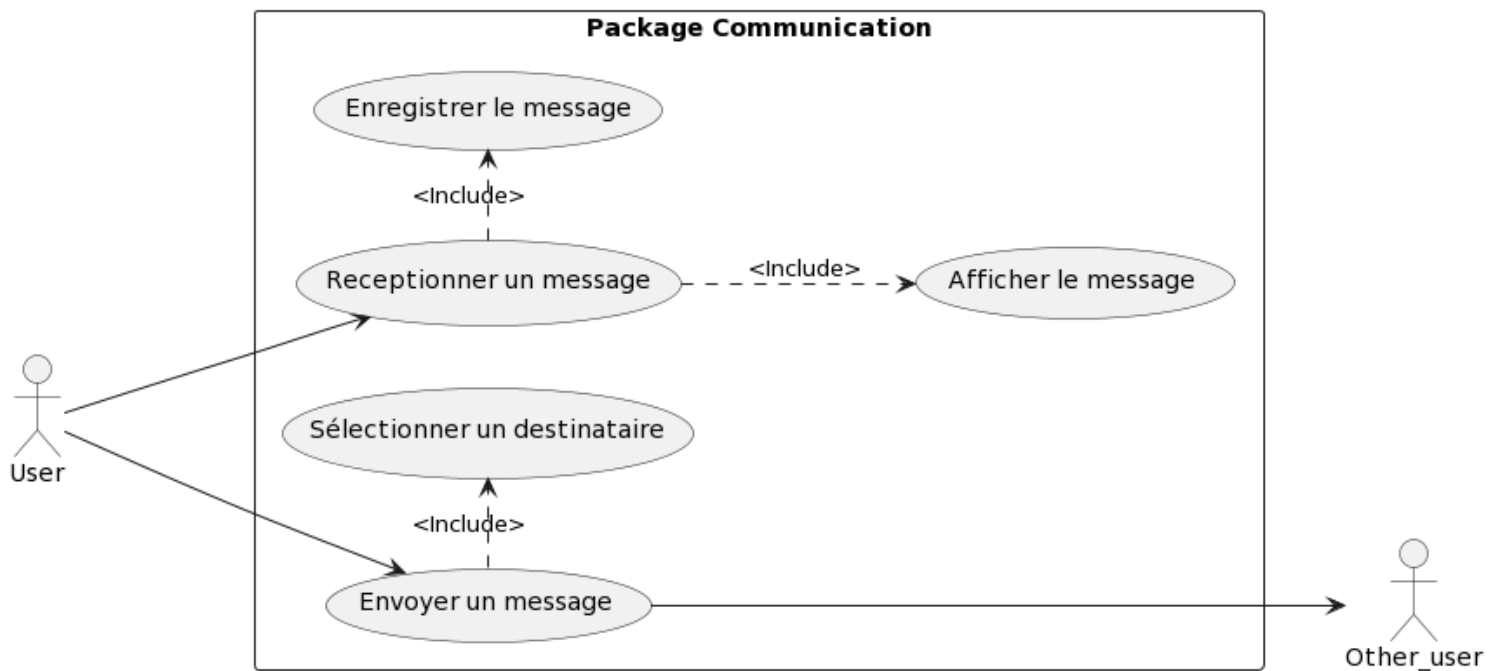
Lors de l'analyse du cahier des charges, plusieurs éléments nous ont semblé être d'une grande importance. Le premier élément à avoir été identifié est l'utilisateur en lui-même. En effet l'utilisateur possède certains attributs nécessaires au bon fonctionnement du système, d'identifiant ou encore l'adresse IP sont des éléments qui nous seront nécessaires lors de la réalisation du projet. Un autre acteur important de cette application, sont les messages en eux-mêmes. Les messages nécessitent d'être horodatés et doivent enregistrer quel utilisateur a envoyé le message mais aussi qui l'a reçu. Les deux derniers acteurs principaux sont la session chat ainsi que l'annuaire. La session chat représente une conversation active entre deux utilisateurs. Chaque session chat est un thread dédié à la conversation et possède son propre port dédié pour la communication. L'annuaire est lui assez explicite. Il représente la liste des différents utilisateurs connectés sur l'application. Il fonctionne grâce à des communications de type UDP.

## Diagrammes de cas d'utilisation :

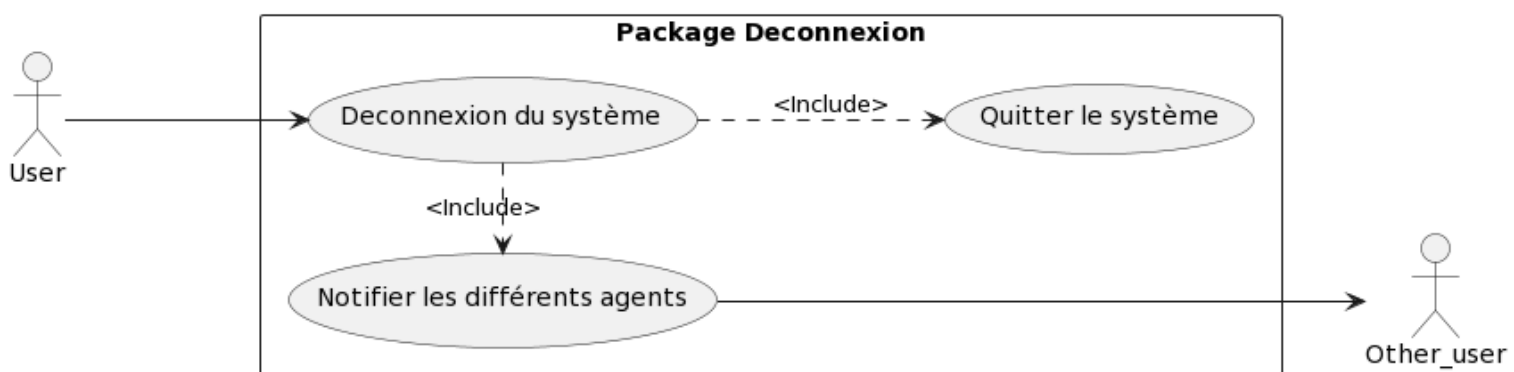
### a) Package connexion



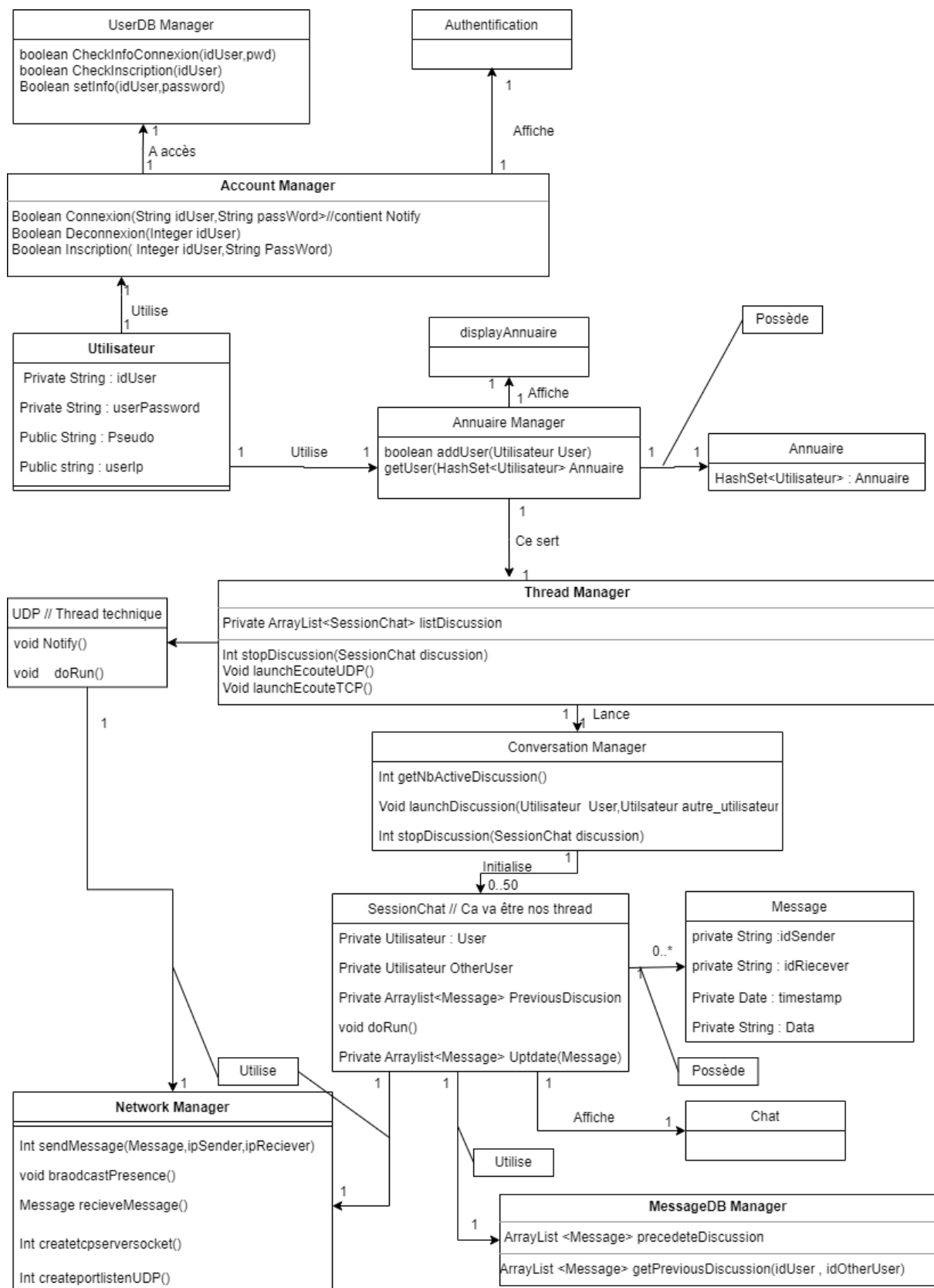
### b) Package communication



### c) Package déconnexion

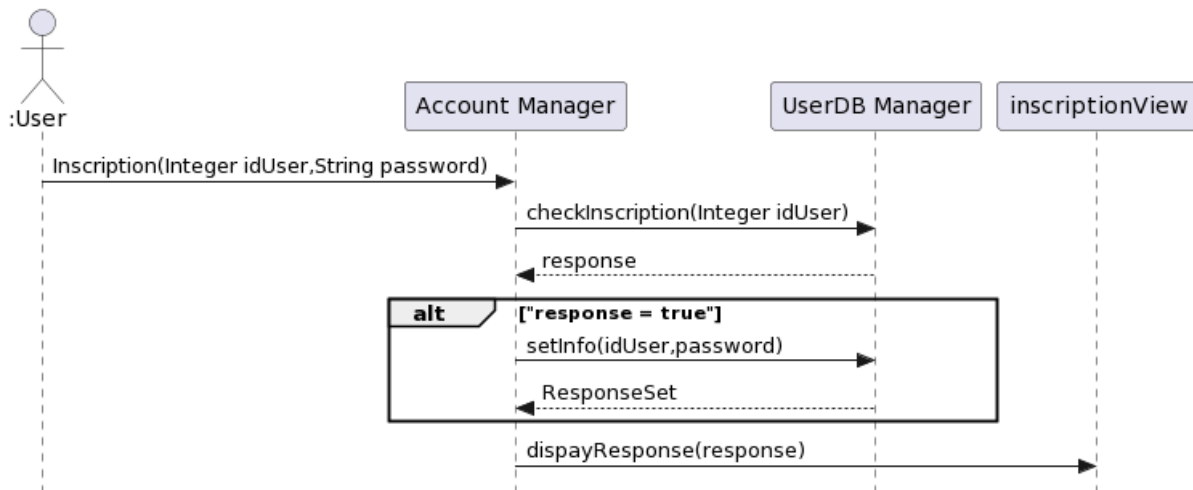


## Diagramme de classe :

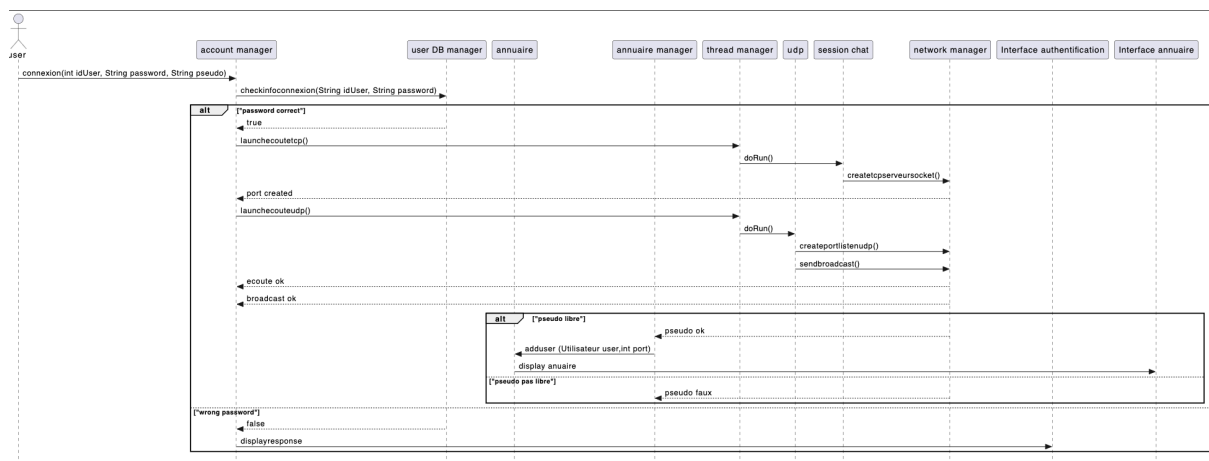


## Diagrammes de séquence :

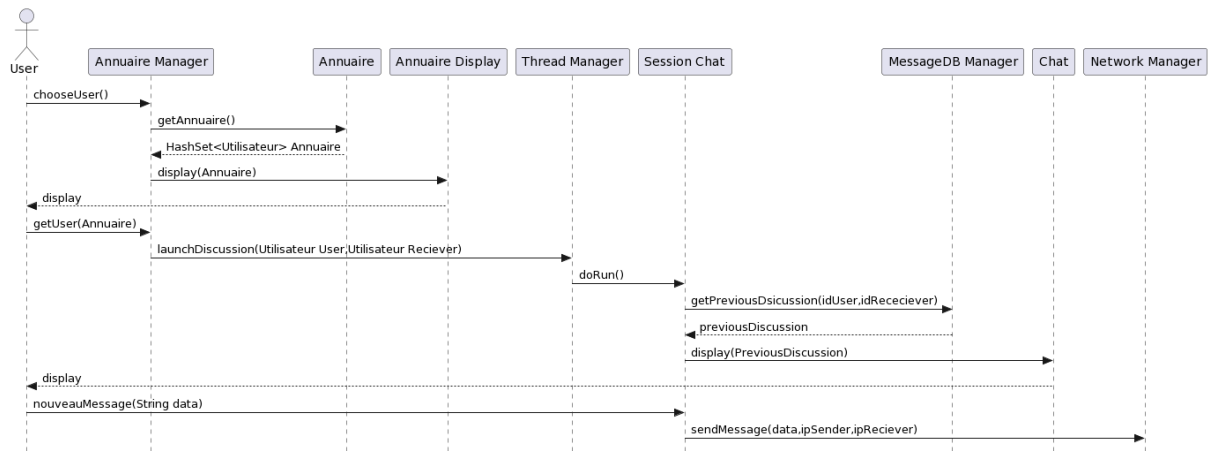
### a) Inscription



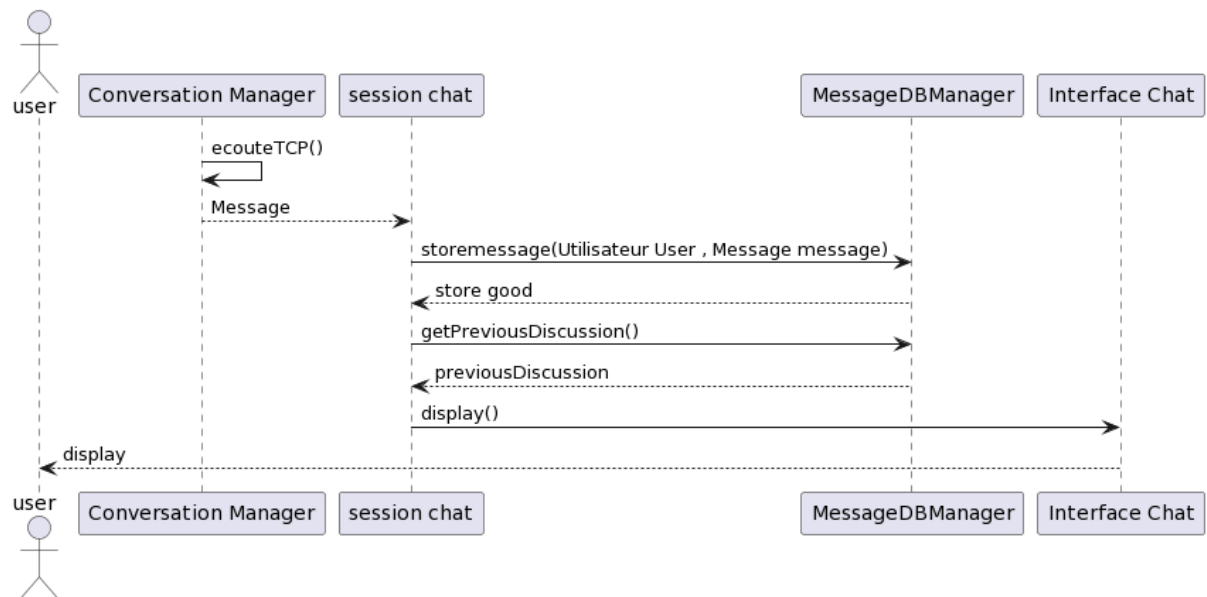
### b) Connexion



### c) Envoie d'un message

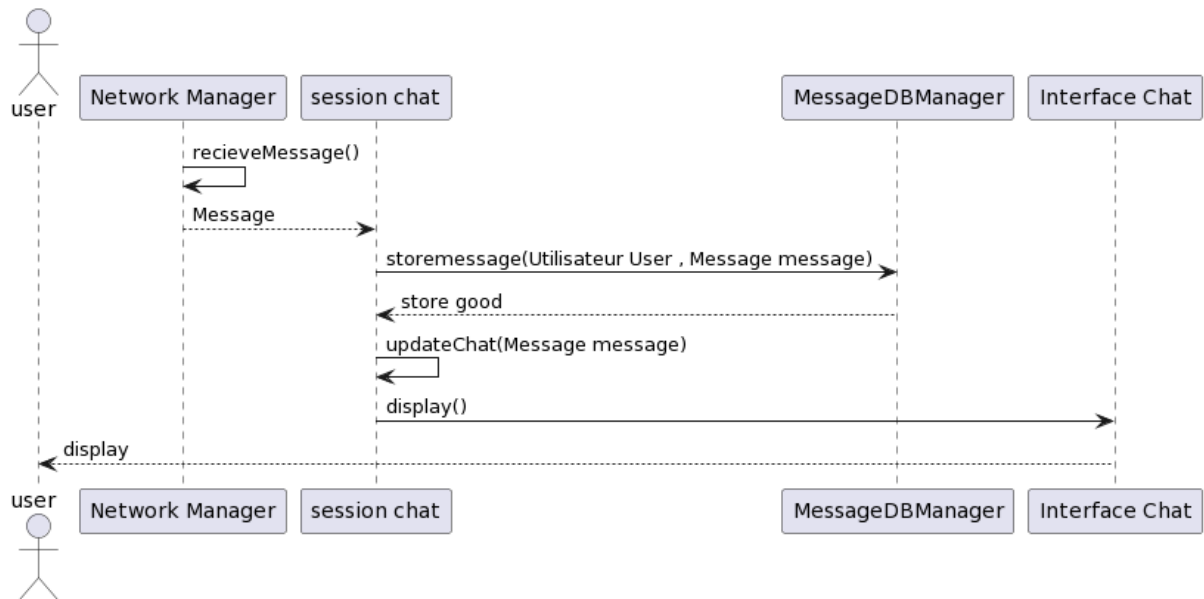


### d) Réception d'un message hors conversation

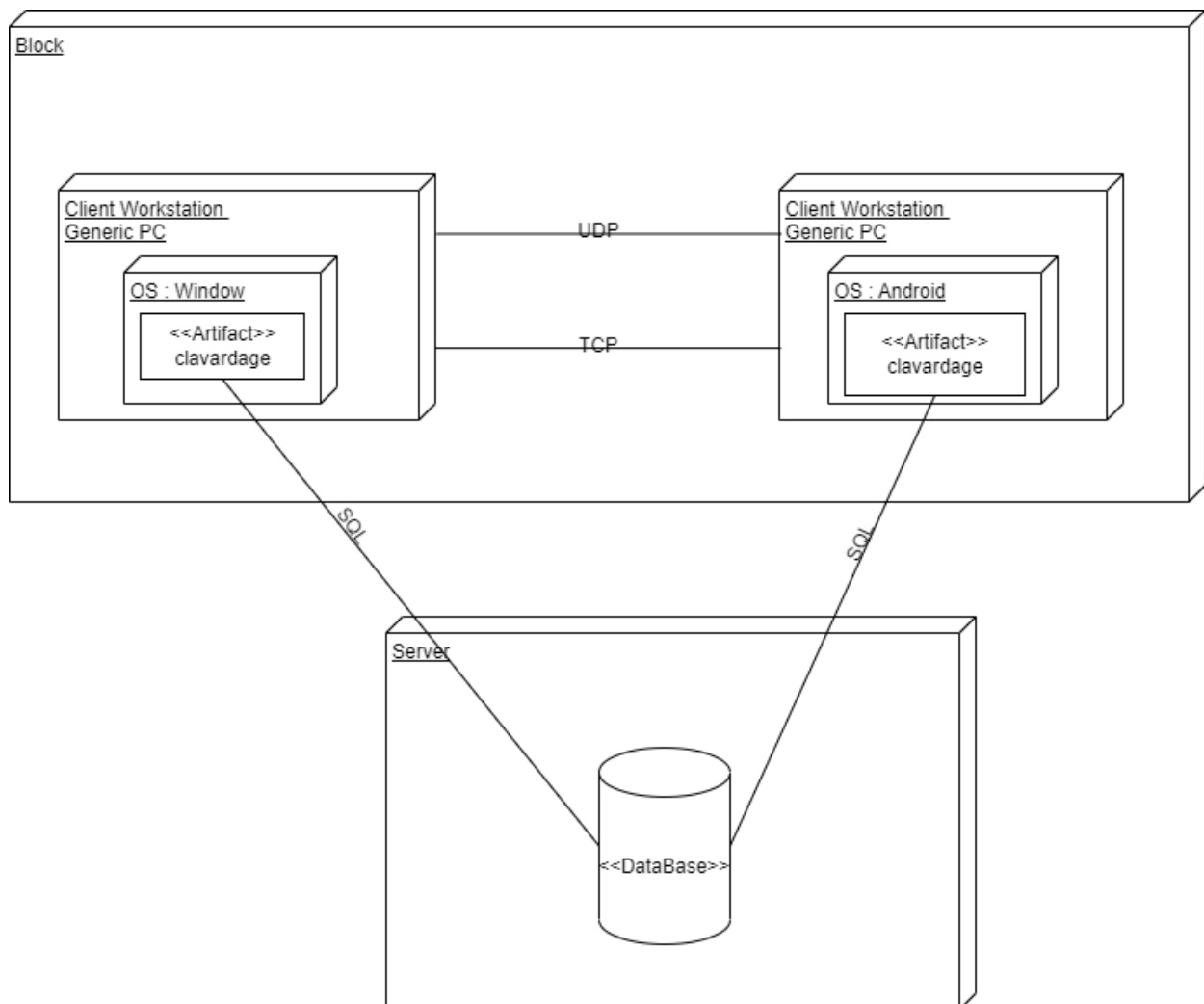




## e) Réception d'un message à l'intérieur d'une conversation



## Diagramme de déploiement :



### 3)Architecture du système et choix technologiques:

L'application a été développée en utilisant l'architecture MVC "Model View Controller". Cette architecture permet de séparer les différentes couches de l'application (couche présentation, couche application, couche données) pour en faciliter la maintenance et l'évolution.

En ce qui concerne le remplissage de l'annuaire, nous avons utilisé UDP, contrairement aux échanges de messages et à l'établissement des discussion qui se fait en TCP pour assurer la fiabilité et la qualité de service.

On a choisi d'utiliser l'ORM hibernate pour gérer la persistance des messages et des utilisateurs. En ce qui concerne la base de données, on utilise MySQL 5.7 et nous avons choisi l'option de la base de données centralisée.

Pour l'interface utilisateur, nous avons choisi d'utiliser JavaFX. Cette librairie permet de créer des interfaces graphiques modernes et interactives. En plus d'être facile à implémenter.

Les choix technologiques effectués ont été motivés par leur performance, leur facilité d'utilisation et leur compatibilité avec les besoins de l'application.

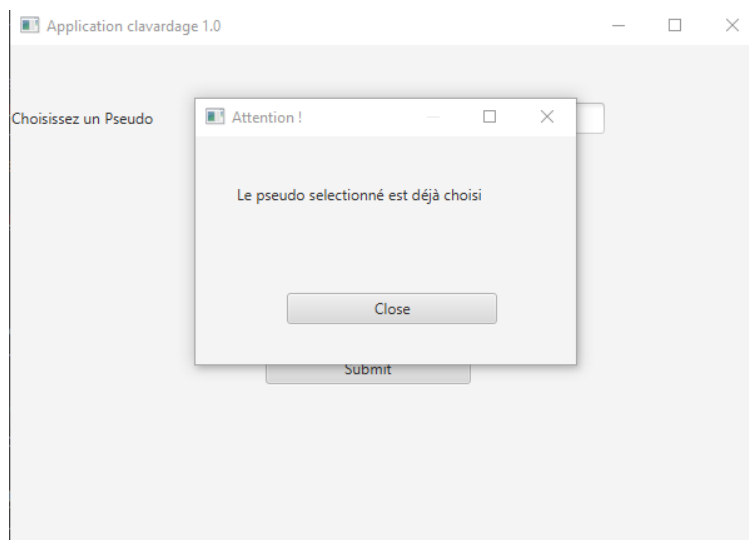
### 4)Procédures d'évaluation et de tests:

Nos tests peuvent se présenter en deux majeures parties. La première partie représentant l'ensemble de nos tests effectués au cours du développement de l'application assurant la vérification des fonctions ajoutées au code. Cela comprend notamment les tests de Broadcast, tests de communication en TCP, tests de l'affichage interface, tests de la création et de la modification de la base de données, etc. Et ce, jusqu'à l'obtention d'une application testable composée de l'ensemble de ses sous-fonctions testées au fur et à mesure. Au départ nous avons effectué les tests à l'aide de prints, ensuite une fois que l'interface a été développée nous avons pu tester hors du terminal. La seconde partie des tests constitue ainsi l'ensemble

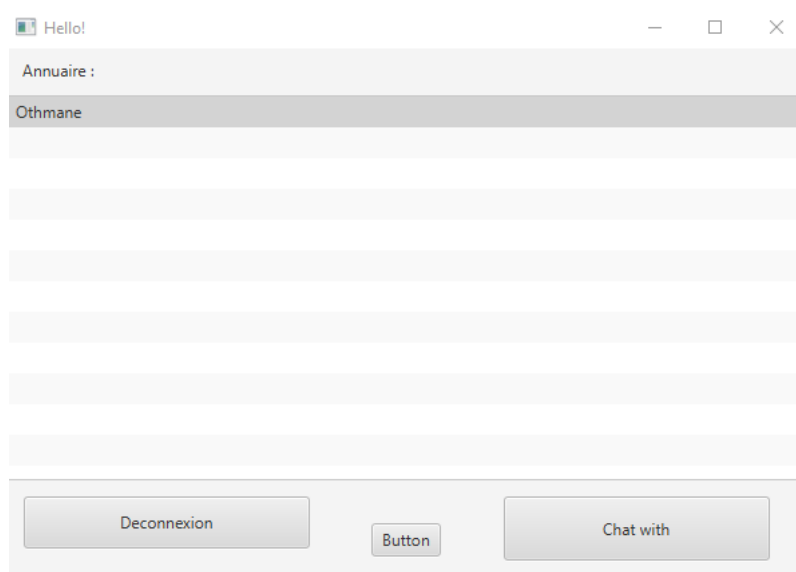
des tests de l'application entre plusieurs ordinateurs. Cette deuxième partie nous a permis d'améliorer le projet mais aussi de vérifier le respect du cahier des charges. Ainsi, cette partie du rapport se présente comme la checklist de l'ensemble des fonctionnalités du ChatBox pour une communication entre 2 ordinateurs ou plus.

Les résultats des tests ont été satisfaisants. Cependant, nous avons rencontré des bugs lors de nos tests, ils ont été corrigés et les tests ont été répétés pour vérifier leur correction.

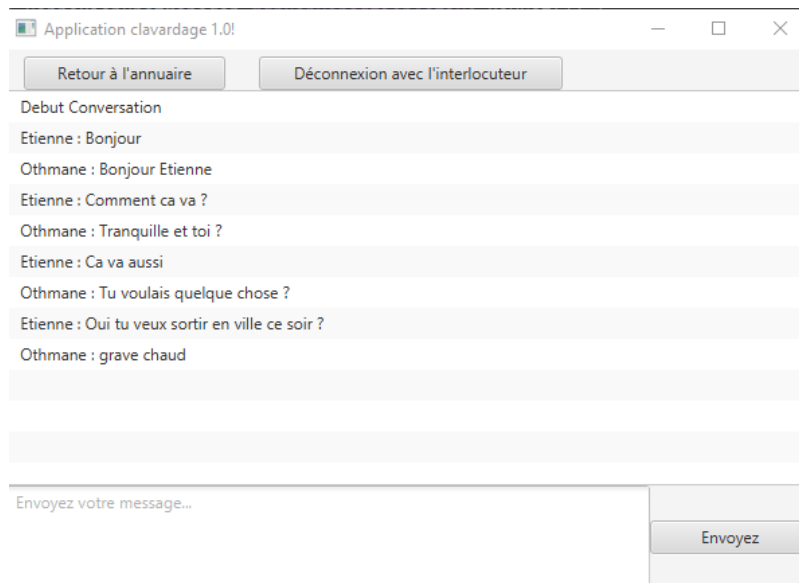
- Test de l'unicité du pseudo à la première connexion :



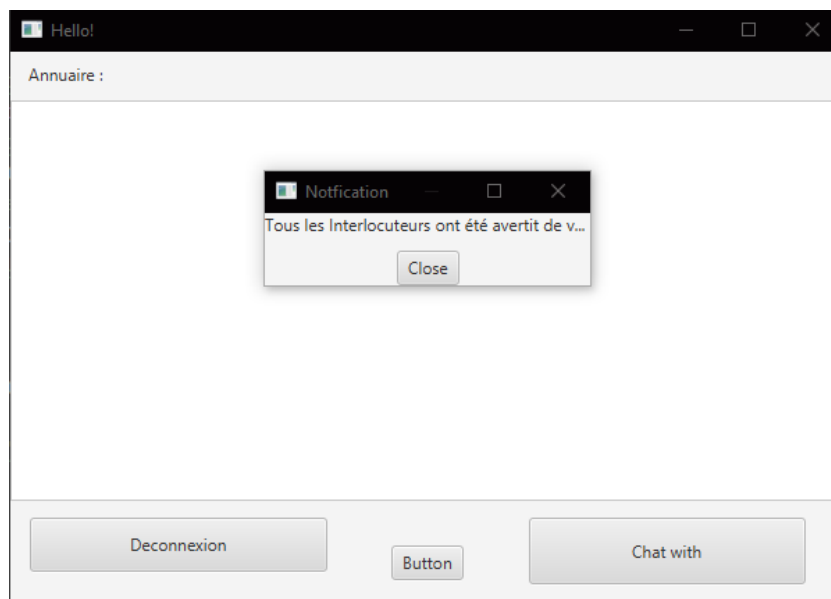
- Test du broadcast : on voit que lors de l'arrivée d'un utilisateur, les autres sont notifiés de sa connexion "Ici l'utilisateur "othmane" est arrivé ":



- Test d'envoi, de réception et d'affichage des messages :



- Test de la déconnexion :



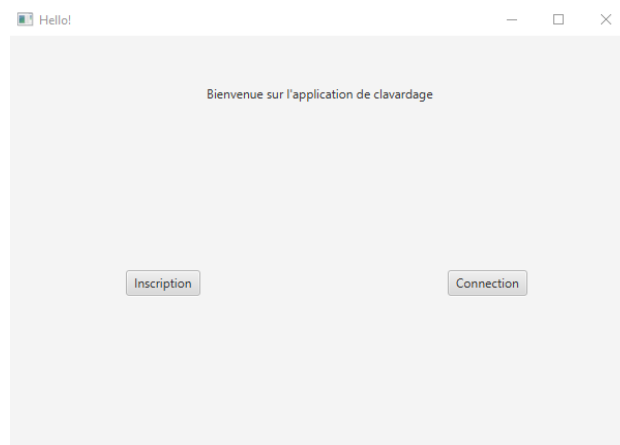
## 5) Procédure d'installation et de déploiement:

Pour installer et déployer l'application, il est nécessaire de suivre les étapes suivantes :

- Prérequis :
  - Version du JDK nécessaire pour faire fonctionner l'application : jdk-19.0.1 téléchargeable ici : <https://www.oracle.com/java/technologies/javase/jdk19-archive-downloads.html>
  - Version Hibernate : 6.1.5
  - Version Jakarta Persistence : 2.2
  - Version Serveur MySQL : 5.7.40
  - Version Connecteurs MySQL : 8.0.28
  - docs Hibernate/Jakarta Persistence : <https://hibernate.org/orm/documentation/6.1/>
  - docs MySQL : <https://dev.mysql.com/doc/>
  - Version JavaFX : 19 API Documentation : [https://openjfx.io/javadoc/19/Getting Started](https://openjfx.io/javadoc/19/Getting%20Started) : <https://openjfx.io/openjfx-docs/docs> : <https://www.oracle.com/java/technologies/javase/javafx-docs.html>
- Déploiement :
  - Télécharger le jdk-19 directement sur le site d'oracle ou depuis votre IDE. Comme le projet est un projet Maven, au moment de Build et du lancement, les dépendances se téléchargeront d'elles-mêmes si vous souhaitez modifier et travailler sur le projet.
  - Pour l'utilisation de l'application veuillez suivre le fichier texte README dans lequel se trouvent toutes les informations relatives à l'application ainsi que le mode d'emploi détaillé

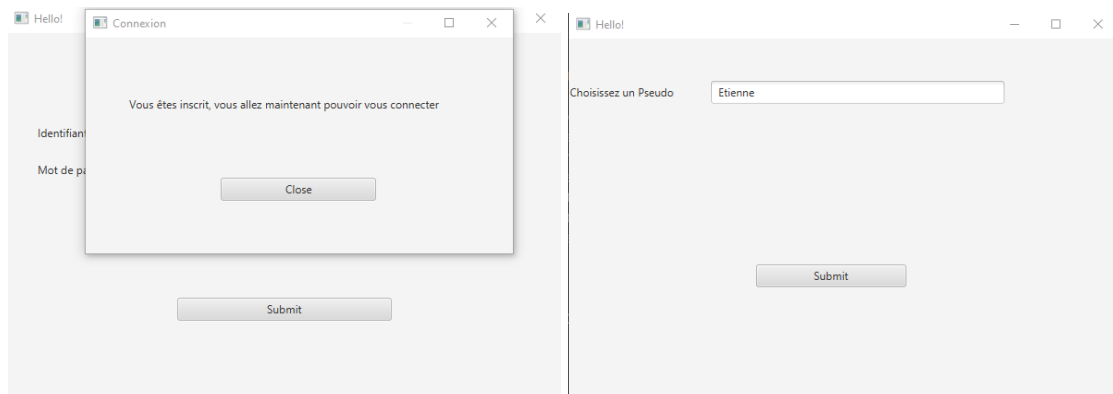
## 6) Manuel d'utilisation simplifié :

Une fois l'application ouverte, vous serez accueillis par notre interface de connexion ou d'inscription.

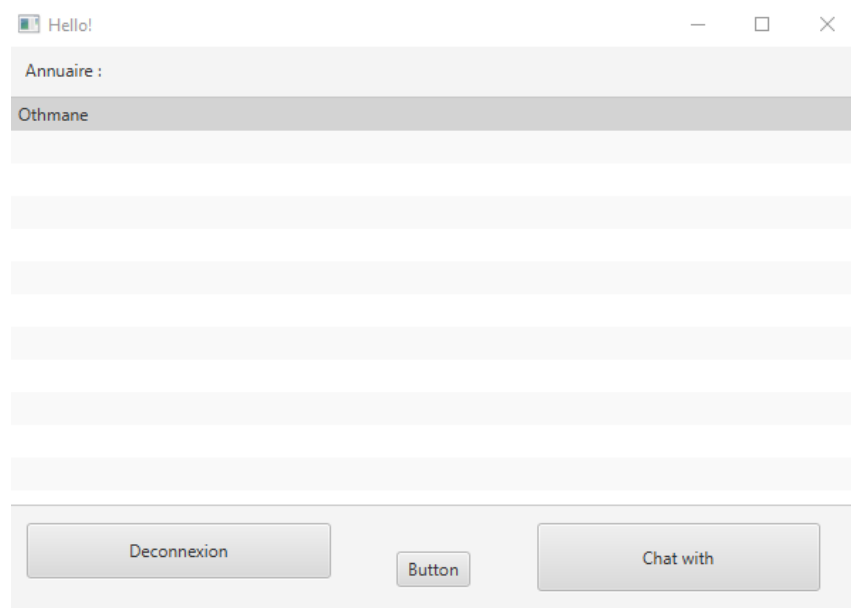


Pour commencer, vous devez d'abord vous inscrire en entrant votre adresse mail et un mot de passe unique.

Si l'adresse mail n'a jamais été utilisée par personne pour s'inscrire auparavant, un message de confirmation d'inscription apparaîtra et vous serez redirigés vers la page de connexion où vous devrez entrer vos identifiants et choisir votre pseudo par la suite.

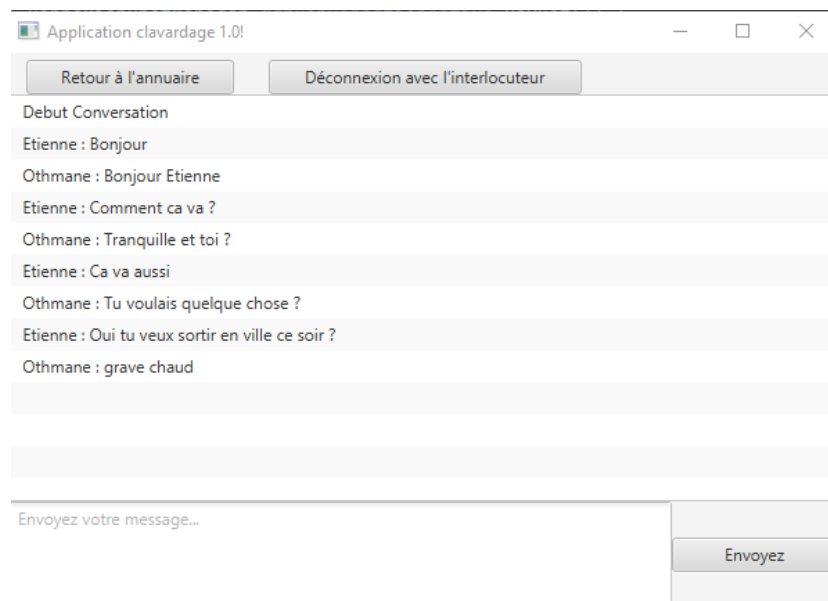


Si le pseudo n'appartient à aucun utilisateur connecté, vous serez redirigé vers la page où se trouve l'annuaire et où vous pourrez commencer à discuter. Ici on voit qu'il n'y a que "othmane" qui est connecté.

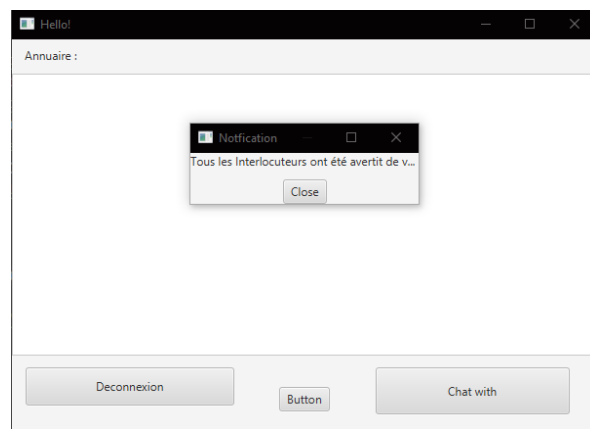
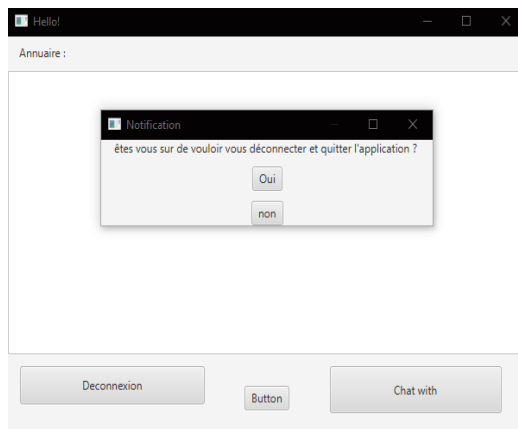


Sinon, vous aurez un message d'avertissement vous demandant de changer votre pseudo car celui que vous avez choisi est déjà pris.

Une fois avoir choisi l'interlocuteur, ici "othmane", vous serez redirigé vers une page où apparaîtront vos anciens messages s'il y'en a avec le bouton envoyer et une zone de texte pour écrire vos messages. Il y a également un bouton pour revenir à l'annuaire pour changer de personne.



Si vous souhaitez vous déconnecter, il suffit de revenir à l'annuaire et de cliquer sur "Deconnexion", une confirmation vous sera demandée et ensuite une notification sera envoyée aux autres utilisateurs.





## 7) Conclusion :

La réalisation de cette application de clavardage en binôme a été un défi passionnant qui nous a permis de mettre en pratique nos connaissances en programmation Java.

La conception et les diagrammes réalisés ont été un préalable important pour la compréhension de l'architecture du système.

Les choix technologiques tels que l'utilisation d'une base de données MySQL et d'une interface utilisateur graphique nous ont permis de développer une application efficace et conviviale.

Le manuel d'utilisation simplifié a été élaboré pour faciliter la prise en main de l'application pour les utilisateurs.

Nous sommes fiers de ce que nous avons accompli et espérons que cette application ait satisfait le cahier des charges qui nous a été remis en début de semestre.