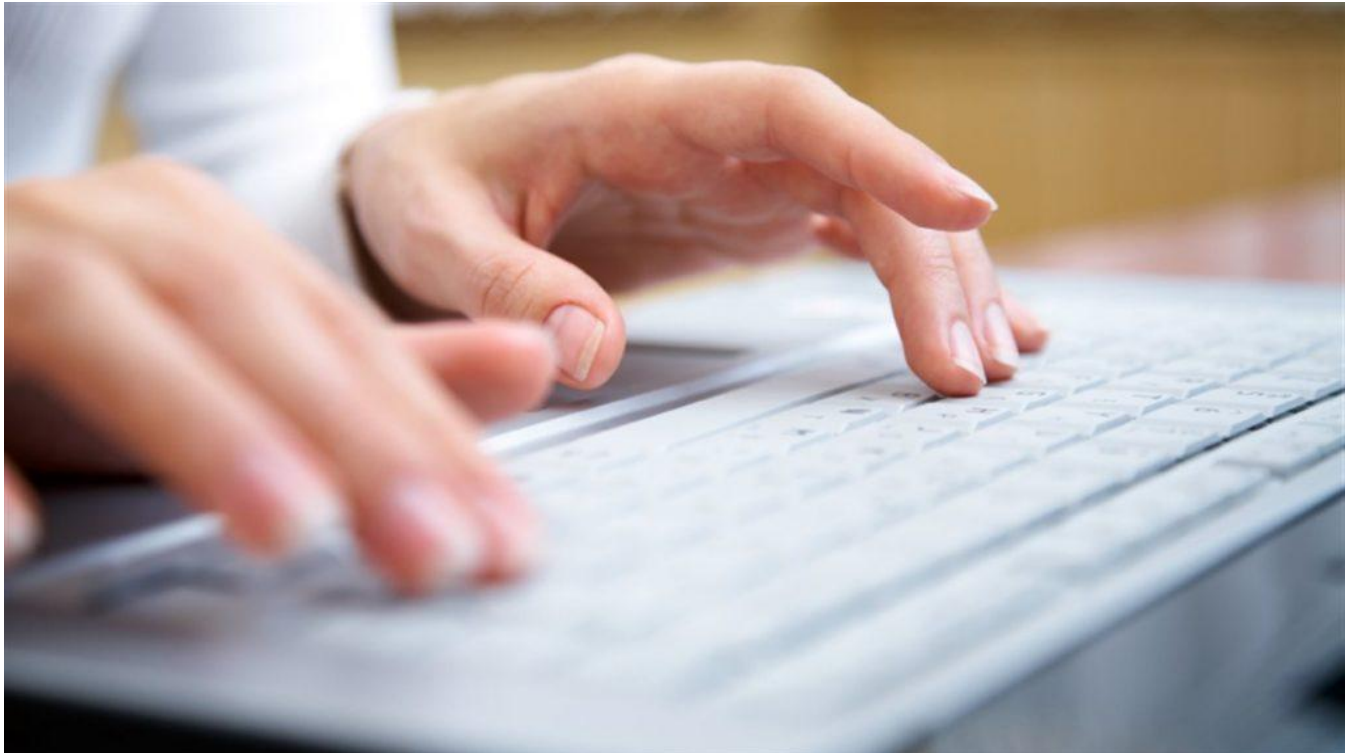


Compte rendu : Projet Java “Clavardage”

Rayane HACHEMI, Paul DUFFAUT



Sommaire

- 1) Procédure d'installation et de déploiement
- 2) Manuel d'utilisation simplifié
- 3) Conception et diagrammes réalisés
- 4) Architecture du système et choix technologiques (bases de de données, GUI, toutes autres libs utilisées)
- 5) Procédures d'évaluation et de tests

Procédure d'installation et de déploiement

Téléchargement du logiciel :

Clonez le git à cette adresse pour télécharger les fichiers du logiciel :

<https://github.com/rayanehmi/ProjetJava>

Génération d'un exécutable .jar

Utilisez Maven pour compiler le logiciel avec la commande :

maven build

L'exécutable créé se trouve alors à l'emplacement *Clavardage/target/* sous le nom suivant :

pr-clavardage-0.0.1-SNAPSHOT-jar-with-dependencies.jar

Execution du logiciel

Une fois l'exécutable créé, lancez-le avec la commande :

java -jar pr-clavardage-0.0.1-SNAPSHOT-jar-with-dependencies.jar

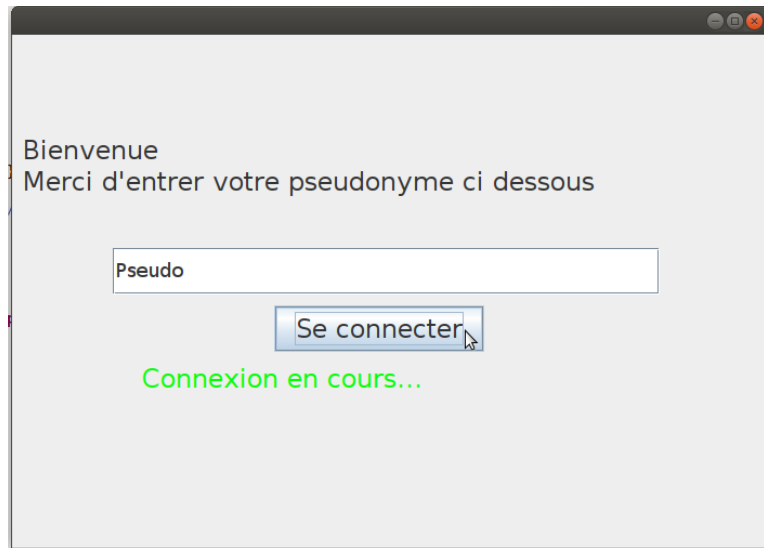
Le fichier "Clavardage.jar" dans le dossier racine est une version précompilée et exécutable du projet.

Manuel simple d'utilisation

Dans cette section, nous allons vous détailler les différentes étapes de fonctionnement de notre application.

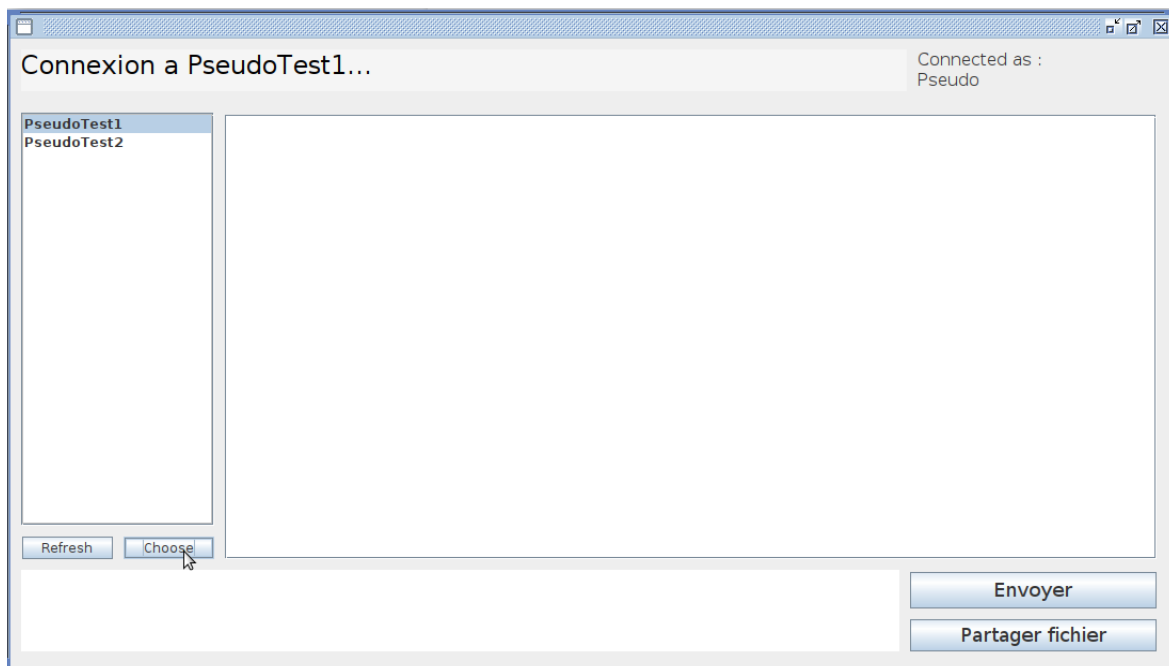
Etape 1 : se connecter

Entrez votre pseudonyme dans la case prévue à cet effet et cliquez sur le bouton "Se connecter". Si le pseudo n'est pas disponible, un message d'erreur s'affichera et vous devrez changer de pseudo.



Etape 2 : choisir un interlocuteur et communiquer

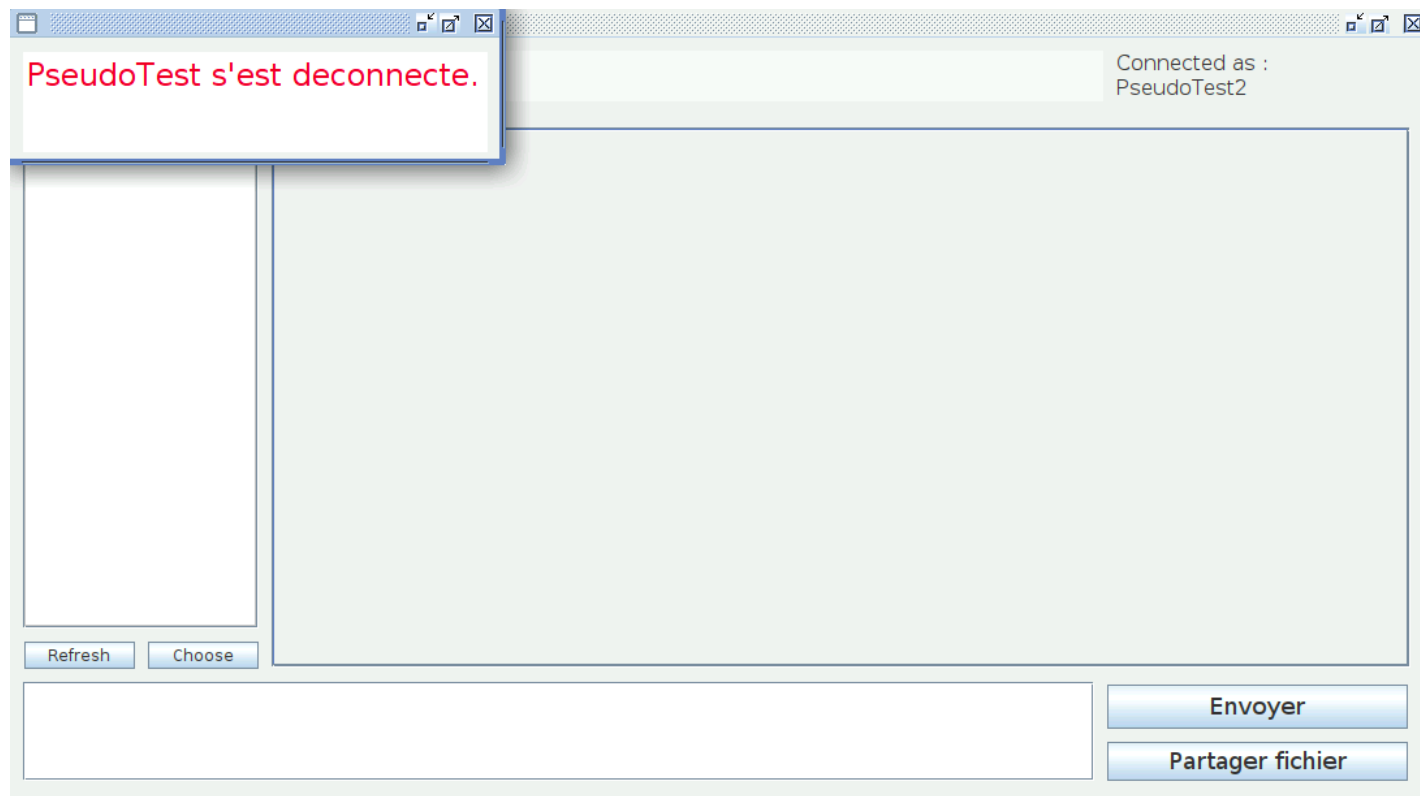
La liste des personnes en ligne s'affiche à gauche de l'interface. Sélectionnez le pseudo de votre interlocuteur et cliquez sur le bouton "Choose". Vous pouvez rafraîchir cette liste avec le bouton "Refresh", sinon un refresh automatique s'effectue toutes les 10 secondes.



Vous pouvez à présent clavarder en tapant votre message dans la zone du bas et en appuyant sur "Envoyer".

Etape 3 : mettre fin à la session de clavardage

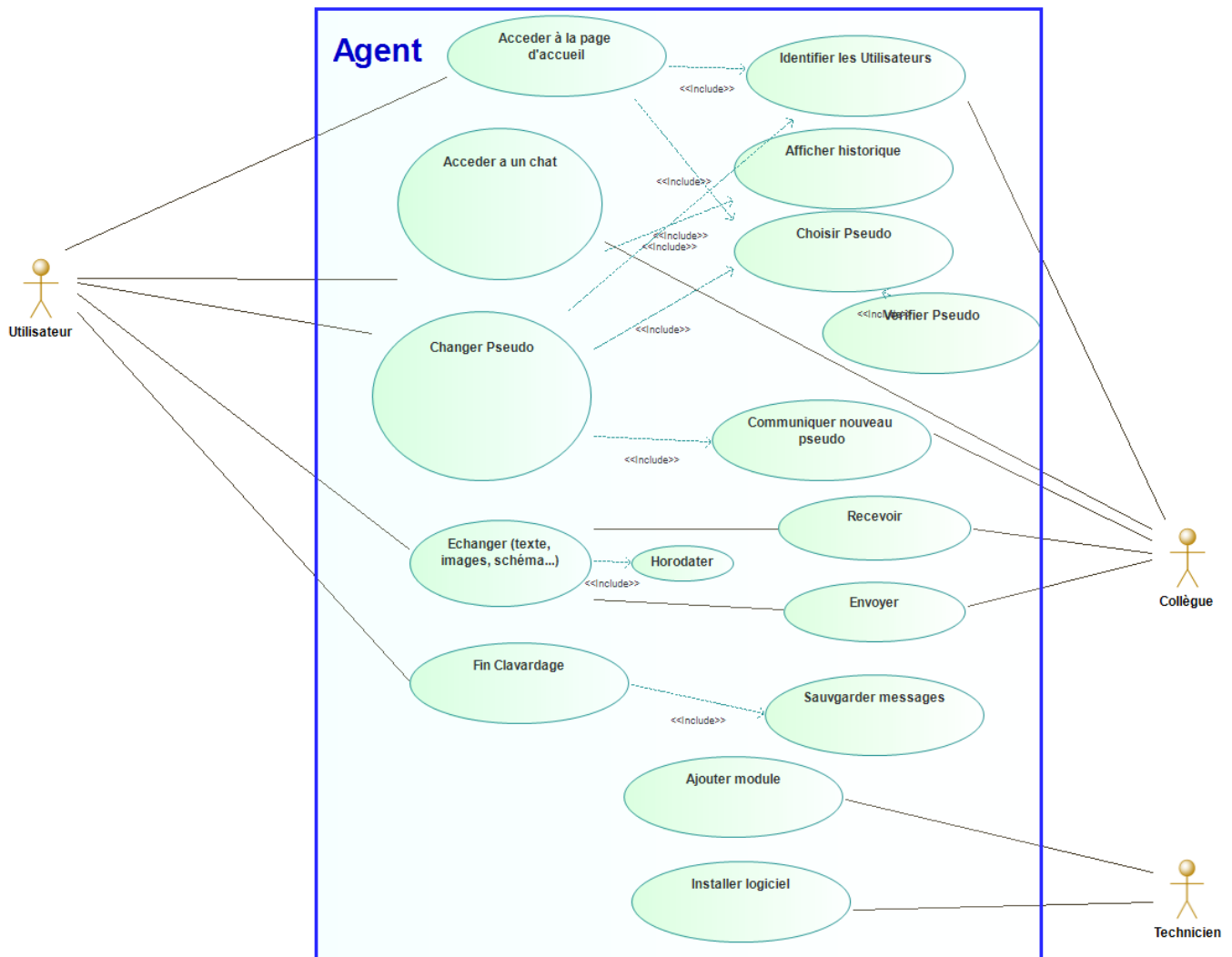
Lorsque vous voulez mettre fin à votre session de clavardage, il suffit d'appuyer sur la croix en haut à droite de votre écran. Celle-ci va clore les serveurs connectés et envoyer un message à tous les autres utilisateurs. Ainsi, lors de votre déconnexion, l'ensemble des utilisateur connectés vont recevoir un message les notifiant de votre déconnexion.



Conception et diagrammes créés

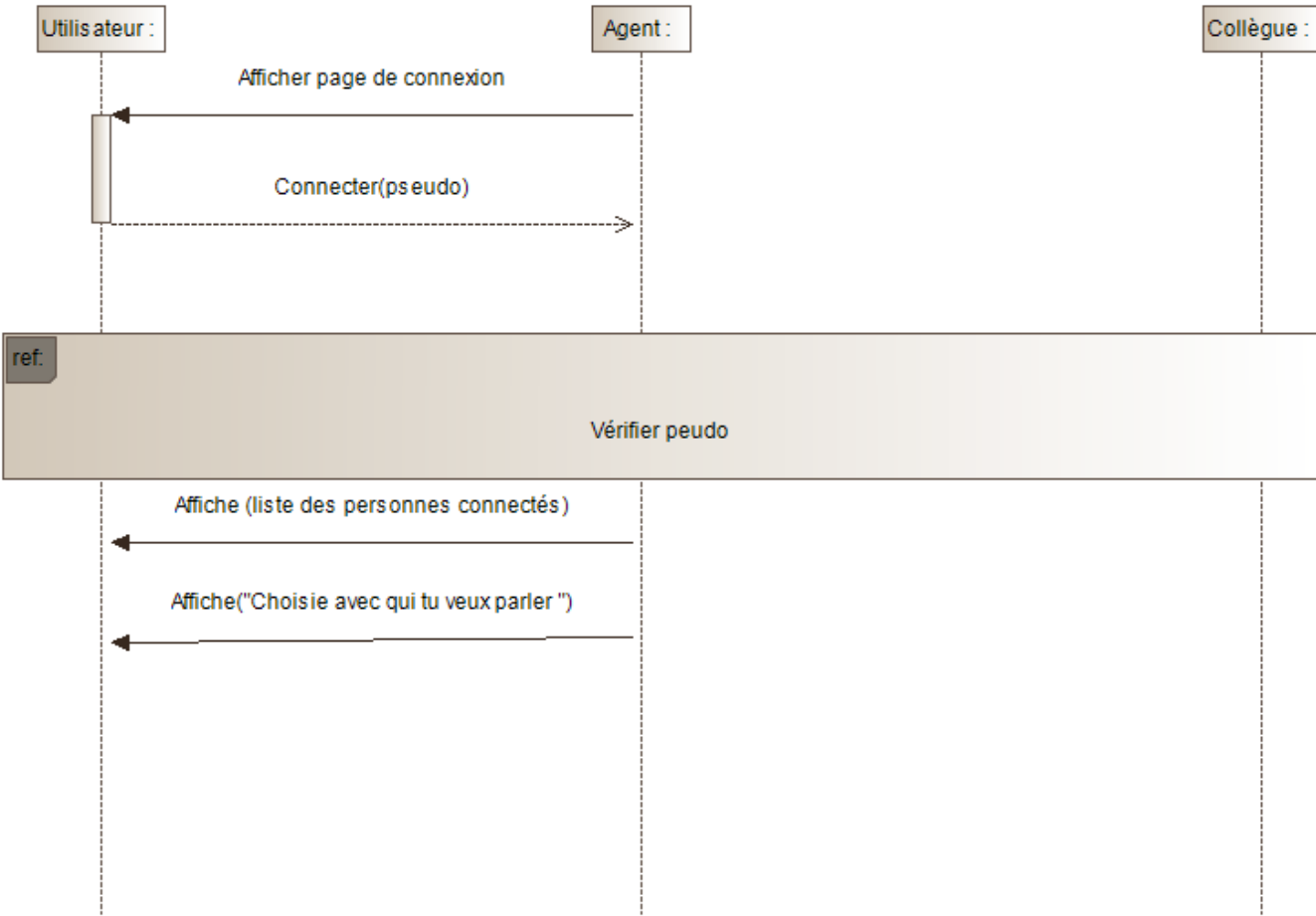
Dans cette partie, nous allons vous présenter les différents diagrammes que nous avons conçus, qui permettent de schématiser la conception de notre application. Pour cela, nous avons créé 1 diagramme de cas, 3 diagrammes de séquences, 1 diagramme de séquences à boîte blanche et 1 diagramme de classe.

Diagramme de cas d'utilisation :

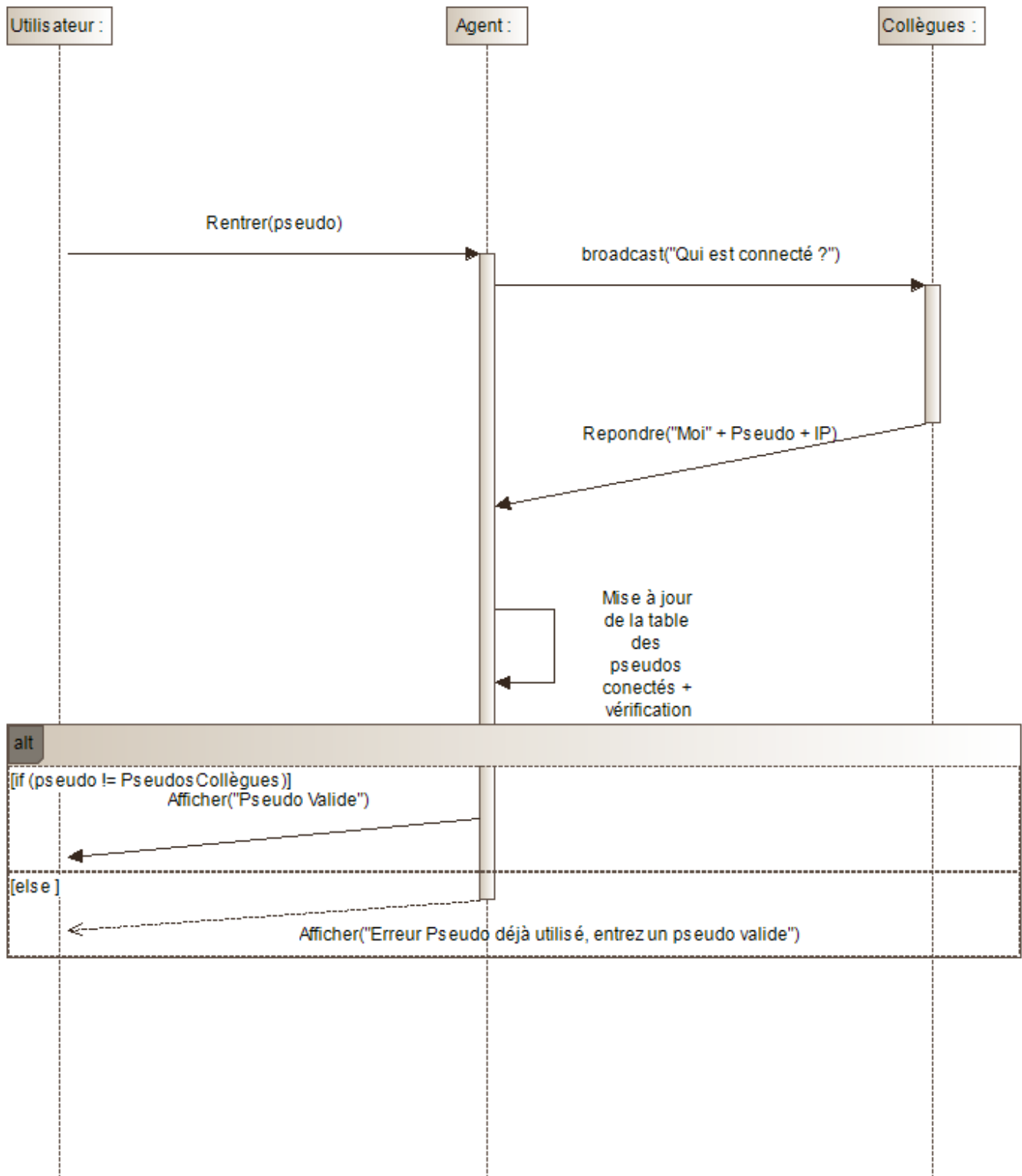


Diagrammes de séquences :

Se connecter



Vérifier pseudo



Conversation

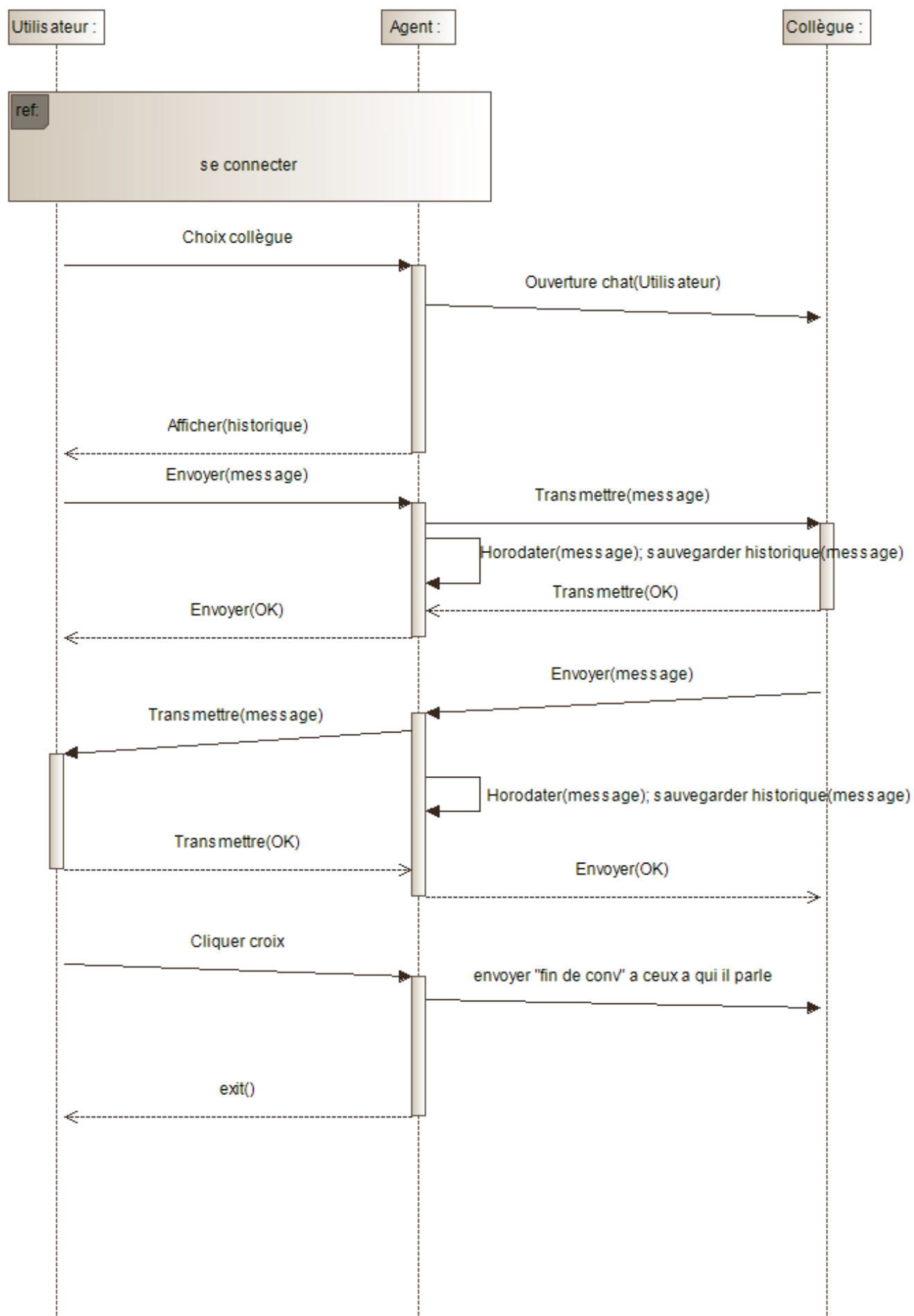
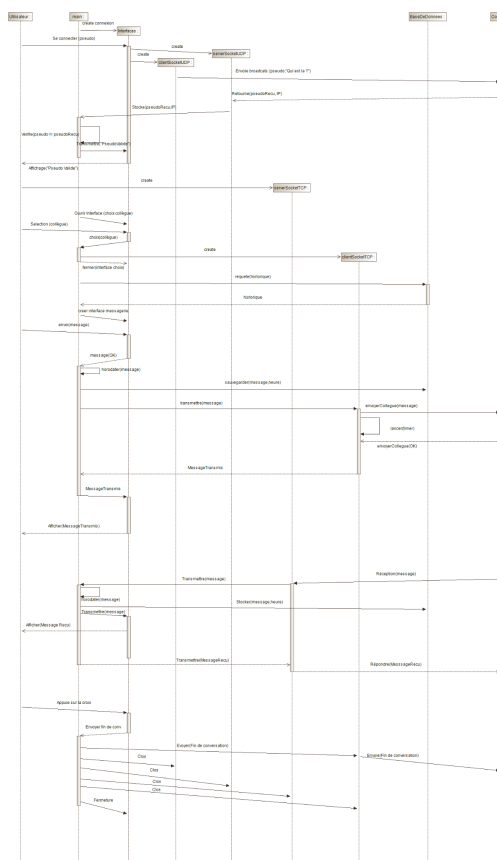
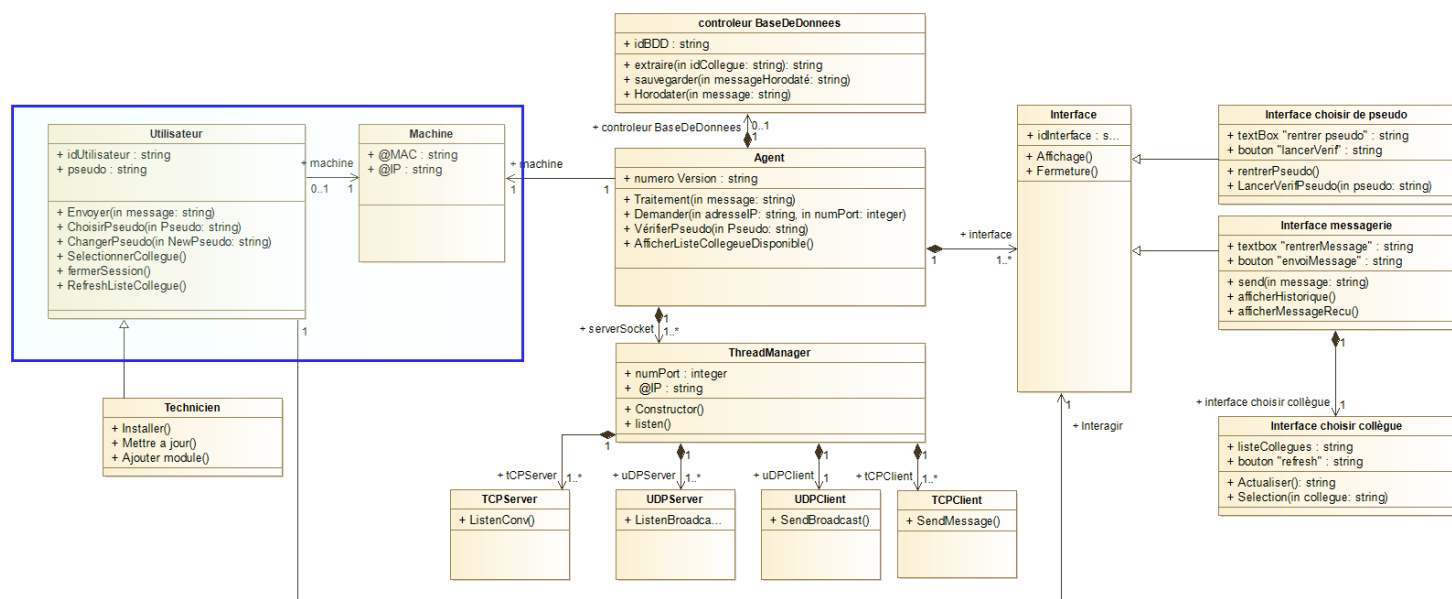


Diagramme de séquences à boîte blanche



(cliquez pour suivre le lien)

Diagramme de classes



Architecture du système et choix technologiques

Project Management

Nous nous sommes tournés vers Maven pour notre *project management*. Maven permet une intégration de plugins ainsi qu'une gestion de projet particulièrement simple.

Build

Nous utilisons le Apache Maven Assembly Plugin pour construire un unique exécutable contenant toutes les dépendances et librairies de notre projet.

Intégration continue

Nous avons installé Jenkins pour supporter l'intégration continue de notre application. Jenkins est réglé pour scruter les données du git toutes les minutes et pour build l'application en cas de nouveau commit.

The screenshot shows the Jenkins web interface for a project named 'Module Clavardage'. The top navigation bar includes the Jenkins logo, a search bar, and user information (Administrateur). The left sidebar contains navigation links: Up, État (selected), Modifications, Répertoire de travail, Lancer un build, Configurer, and Supprimer Module. The main content area displays the project name, full project name, and a list of recent artifacts. Below the artifacts, there is a section for 'Liens permanents' (Permanent Links) listing various build links.

Jenkins recherche ? 2 1 Administrateur se déconnecter

Tableau de bord > Clavardage 2.0 > Clavardage >

Up

État

Modifications

Répertoire de travail

Lancer un build

Configurer

Supprimer Module

Historique des builds tendance

Filter builds...

#15 27 janv. 2022 18:14

#14 27 janv. 2022 18:10

Atom feed des builds

Atom feed des échecs

Module Clavardage

Full project name: Clavardage 2.0/pr:pr-clavardage

Ajouter une description

Espace de travail

Derniers artefacts obtenus avec succès

pr-clavardage-0.0.1-SNAPSHOT-jar-with-dependencies.jar	6,32 MB	view
pr-clavardage-0.0.1-SNAPSHOT.jar	31,69 KB	view
pr-clavardage-0.0.1-SNAPSHOT.pom	1,77 KB	view

Changements récents

Liens permanents

- Dernier build (#15), il y a 49 s
- Dernier build stable (#15), il y a 49 s
- Dernier build avec succès (#15), il y a 49 s
- Dernier build en échec (#14), il y a 4 mn 49 s
- Dernier build non réussi (#14), il y a 4 mn 49 s
- Last completed build (#15), il y a 49 s

Partie Network (package “com”)

Nous nous sommes servis du module java.net pour gérer la partie réseau du projet. Nous déléguons beaucoup de tâches à des threads parallèles, notamment les tâches d'écoutes du réseau et de communication TCP.

Partie GUI (package “gui”)

Nous nous servons exclusivement de Java Swing pour nos interfaces graphiques. Lors de l'intégration, nous avons utilisé le module WindowBuilder d'Eclipse afin de pouvoir créer facilement nos GUI.

Il existe trois Frames :

- **WelcomePage** permet de choisir son pseudo et de se connecter
- **MainWindow** permet de communiquer avec les autres utilisateurs
- **Popup** s'affiche lorsqu'un utilisateur ferme sa session

Partie Base de données (package “db”)

Nous avons choisi d'utiliser la base de données de l'INSA mise à disposition des étudiants pour la réalisation de notre projet. Le nom de la base de données est : tp_servlet_014. Il existe deux tables :

- une table User qui contient les pseudos des utilisateurs
- une table Message qui contient le pseudo d'envoi, le pseudo destinataire, l'heure et la date d'envoi et le contenu du message

Procédures d'évaluation et de tests

Voici une liste non exhaustive des scénarios-tests que nous avons mené, des plus basiques aux plus complexes :

- Se connecter au réseau local avec un pseudo libre
- Se connecter au réseau local avec un pseudo déjà utilisé par quelqu'un d'autre
- Choisir un interlocuteur et lui envoyer un message
- Choisir un interlocuteur et lui envoyer un message puis changer d'interlocuteur
- Communiquer avec plusieurs interlocuteurs différents
- Recevoir un message d'un interlocuteur que l'on a pas explicitement choisi
- Choisir un interlocuteur avec qui on a déjà parlé afin d'afficher l'historique des messages

Notes :

- Il est impossible de tester la communication sur un seul ordinateur. Il faut au moins deux ordinateurs sur un même réseau local.
- L'application est actuellement réglée sur deux types d'adresses locales : 192.168.x.x ainsi que 10.1.x.x (INSA).

Liste de fonctionnalités à implémenter

1) Vérification de caractères interdits

Nous n'avons pas encore implémenté de vérification de caractères interdits lors du choix du pseudo et de l'envoi de message. De ce fait, notre démonstrateur est actuellement vulnérable aux attaques par injection SQL.

2) Changement d'interlocuteur en cliquant sur un autre onglet

Actuellement, il faut forcément cliquer sur le bouton "Choose" pour changer d'interlocuteur. Sur le long terme, il faudrait que le changement d'onglet change aussi l'interlocuteur choisi.

3) Partage de fichiers

Bien que le bouton soit déjà implémenté, le partage de fichiers n'est pas encore effectif.