INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Estelle CHARPENTIER
Célia DESPAUX

Rapport de projet POO

Système de clavardage distribué interactif multi-utilisateurs temps réel

Janvier 2023 - promotion 57

Sommaire

Introduction

Dans le cadre de la 4ème année IR, nous avons dû réaliser un logiciel de communication décentralisé. Ce projet s'appelle Clavardage et a été conçu et implémenté par Estelle CHARPENTIER et Célia DESPAUX. Il permet à des utilisateurs présents sur un même réseau d'échanger des messages et il inclut des mécanismes de gestion de l'historique des messages et de découverte des utilisateurs du réseau.

Le projet a commencé en novembre 2022 et s'est terminé en janvier 2023. Durant cette période nous avons pu rendre un projet fonctionnel et qui répond à une partie des contraintes du cahier des charges. Pour la gestion du projet nous avons utilisé la méthode AGILE, et nous avons aussi utilisé des outils d'automatisations.

Dans ce rapport, nous décrirons les méthodes d'installation et de déploiement ainsi que le manuel d'utilisation. Nous présenterons aussi les diagrammes de conception, l'architecture choisie du système, les procédures de tests utilisées et finalement les procédures de développement automatisé utilisées.

1. Manuel d’utilisation

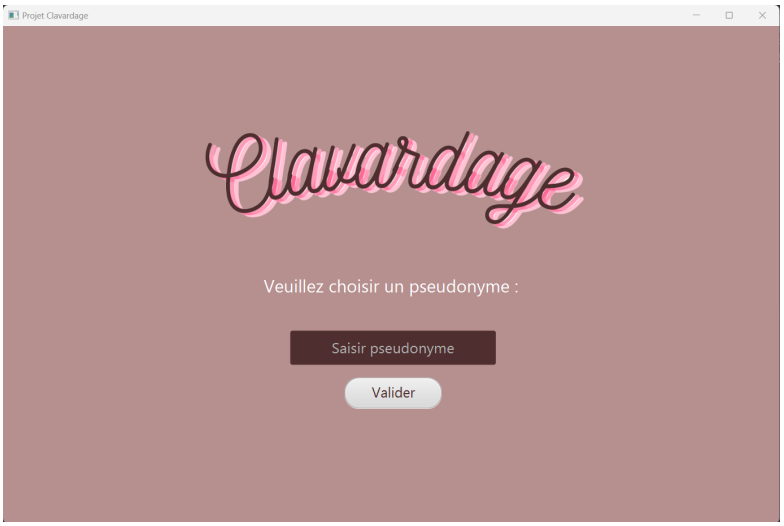
Ce manuel explique le fonctionnement de notre logiciel et les différentes fonctionnalités qui sont proposées. La première partie concerne l'installation, qui peut être effectuée sur Linux, Windows ou bien Mac.

a. Installation

Nous n’avons pas pu faire de fichier exécutable à cause de difficultés rencontrées avec JavaFX. Pour installer notre projet, il vous suffit de télécharger l'intégralité du projet. D’ouvrir les fichier User (Model), DB_locale_manager (Manager) et Main (default package). Il faut changer quelques valeurs dans ces deux premiers fichiers (voir commentaire sur le code), puis lancer l’application en effectuant un run sur le Main.

b. Utilisation

La première page qui s’affiche en ouvrant notre projet est la page de connexion. Sur cette page vous devez entrer un ID et un mot de passe pour pouvoir accéder à notre messagerie. Cet ID et ce mot de passe se trouvent dans la classe DB_locale_manager et correspond à un user. Une fois ces données entrées, il faut appuyer sur Connexion.



Ensuite une nouvelle page s'affiche, c'est la page de choix de pseudo. Il faut choisir un pseudo qui ne contient pas de caractères spéciaux et qui fait moins de 30 caractères. Ce pseudo doit aussi être unique par rapport au pseudo des autres utilisateurs. Une fois le pseudo choisi, il faut appuyer sur Valider.

Une fois connecté, vous arrivez sur la page principale de notre système de clavardage. C'est sur cette page que vous accédez à la plupart des fonctionnalités. Tout d'abord en haut à gauche, vous avez votre pseudo et un bouton en dessous pour changer de pseudo à tout moment. Pensez à rafraîchir le pseudo après l'avoir changé. Ensuite, sur la barre du haut apparaissent tous les contacts présents dans votre annuaire. Quand une personne se connecte, elle apparaîtra dans cette barre et fera partie de vos contacts. Si votre ami est connecté, il apparaîtra en clair, alors que s'il n'est pas connecté en ce moment,

il apparaîtra en foncé. Si vous cliquez sur un de ces utilisateurs, votre historique de discussions apparaîtra au milieu de l'écran. Ensuite, pour envoyer un message à cette personne, il suffit d'écrire le message dans la barre en bas et puis d'appuyer sur la flèche à droite. Vous verrez votre message s'afficher dans la conversation. Finalement, si vous voulez vous déconnecter, il vous suffit de cliquer sur le bouton Déconnexion.

Lorsque vous avez appuyé sur le bouton Déconnexion, cette page est affichée ce qui veut dire que vous avez bien été déconnecté.

2. Conception

Tous nos diagrammes UML sont disponibles dans le fichier UML version finale. Dans cette partie nous allons seulement présenter les diagrammes qui nous paraissent les plus cohérents pour la compréhension du projet.

a. Diagramme des cas d'utilisation

La partie COO qui à commencé avant l'implémentation du projet à été indispensable pour notre bonne compréhension du projet et nous questionner sur l'architecture de notre système. Après avoir lu le cahier des charges, nous nous sommes rendu compte qu'il était très complet. Nous avons donc dû faire des choix quant aux fonctionnalités qu'on voulait implémenter. Nous avons choisi une implémentation simple qui regroupe les fonctions que nous jugions les plus importantes dans le cadre de ce projet. Le premier diagramme réalisé et le diagramme des cas d'utilisation car c'est celui qui est le plus facile pour retranscrire les spécifications du cahier des charges.

b. Diagramme de classes

Ensuite, nous avons réalisé le diagramme de classes. Il nous a permis de mieux structurer nos fonctions dans des classes bien définies et de visualiser les échanges entre ces classes. Notre projet est structuré en 4 packages : le package Transport qui contient les classes TCP et UDP notamment, le package Model qui regroupe nos deux modèles qui sont User et Message, le package Manager qui correspond à tous les manager qui gèrent notre projet et finalement le package View qui englobe toutes les views. Ce diagramme à bien sur évoluer tout au long du projet, nous vous présentons ici la version finale.

c. Diagramme de séquence

La réalisation des diagrammes de séquence nous a permis de bien visualiser les différentes étapes comme la connexion, l'envoie des messages… ainsi que les différentes fonctions à implémenter. Pour ne pas encombrer le rapport nous ne présentons pas les diagrammes de séquences ici mais vous pouvez les retrouvez dans le fichier UML Version finale.

3. Architecture du système et choix technologiques

a. Environnement de développement

Ce projet a été développé sur Intellij et Eclipse. Nous avons utilisé la version 11 de Java, qui est compatible avec les ordinateurs de l'INSA. Le projet réalisé est un projet Maven.

b. Implémentation de la base de données

Nous avons décidé d'implémenter qu'une seule base de données locale pour notre projet. Cette base de données contient trois tables qui correspondent à la table des utilisateurs, la table annuaire et la table discussion. Cette base de données est bien sûr décentralisée pour répondre aux exigences du cahier des charges. Elle est développée avec SQL. Elle est composée de 3 tables que vous pouvez voir ci dessous :

Table Utilisateur:

id	pseudo	mdp	ip_adr	port_nb	connecte
1	Celia	motdepasse	172.29.71.138	6000	1

Table Annuaire :

id_ami	pseudo_ami	connecte	ip_ami
4	moi	1	172.29.89.85

Table Discussion:

id_user	date	message	recu
4	27-01-2023 18:52:42	Coucou	0
4	27-01-2023 19:04:49	Salut !	1

Si vous souhaitez plus de renseignements sur notre base de données, vous pouvez les trouver dans le fichier UML version finale.

c. Implémentation de l'interface

L'interface à été réalisée avec JavaFX car nous avons trouvé que c'était un moyen efficace de pouvoir réaliser l'interface que nous avions prévu lors de la conception. La gestion de notre interface est réalisée grâce au package View qui regroupe les différentes vues de notre projet.

d. Architecture de la communication

La communication, quant à elle, est gérée principalement par le package Transport. Pour les broadcast, nous utilisons le protocole UDP, par exemple pour dire qu'on est connecté, pour demander les pseudo, pour dire qu'on est déconnecté… En ce qui concerne les messages échangés entre les utilisateurs, ils sont transmis avec le protocole TCP, ce qui permet de garantir le transfert de ces messages de façon fiable.

Nous avons aussi décidé de définir un format particulier pour les messages, qui nous permet d'avoir les informations importantes dans un message. C'est-à-dire le contenu du message mais aussi l'utilisateur qui a envoyé le message ainsi que le type du message par exemple si c'est un message d'une discussion ou un utilisateur qui vient de se connecter.

4. Procédures de tests

Les tests ont été réalisés principalement dans les salles de TP de l'INSA car il fallait bien sûr que les deux utilisateurs soient connectés sur le même réseau pour simuler un réseau d'entreprise. Quand nous voulions travailler chez nous et donc tester hors INSA, nous avons utilisé l'application ZeroTier qui permet de créer un réseau avec des adresses privées utilisables au sein de ce réseau.

a. Tests de la communication

Pour l'échange de messages TCP et UDP, nous avons mis en place des tests au fur et à mesure des fonctions créées pour envoyer et recevoir des messages dans le package Transport du projet. Les tests sont dans un package à part qui s'appelle TestMessages.

b. Tests de la base de données

Les tests de la database ont été réalisés en local avec des petits scénarios pour tester l'ajout ou la suppression de certains éléments par exemple. Nous pouvons visualiser les évolutions de la database facilement grâce à un database viewer.

5. Processus de développement automatisé

a. Git

Durant tout le projet nous avons utilisé GitHub. L'avantage de GitHub est de pouvoir travailler en collaboration sur un même projet, de pouvoir revenir à des versions précédentes si jamais nous rencontrons des erreurs. De plus nous avons relier Git à Intellij pour pouvoir effectuer des commit et des push très facilement. Le dépôt contient une branche main où nous avons un regroupement de notre travail et puis deux branches personnelles où nous travaillons chacune dessus, que nous regroupions sur le main lorsque des avancées ont été faites.

b. Maven

Nous avons utilisé Maven pour la gestion du projet et principalement la gestion des dépendances. Cela nous à permis d'implémenter notre projet sur Linux, Windows et Mac à la fois sans problèmes de compatibilité.

c. Jira et la méthode AGILE

Nous avons utilisé la méthode AGILE pour la gestion de notre projet, associé a Jira pour rédiger et accéder a nos sprints. Etant donné que c'était la première fois que nous utilisions la méthode AGILE et un logiciel comme Jira, nous avons eu du mal à évaluer nos tickets et nos sprint, et nous nous sommes très souvent retrouvées à la fin du sprint sans avoir finis tous les tickets prévus.

Conclusion

Pour conclure, nous avons conçu et implémenté une application de clavardage distribué multi-utilisateurs temps réel.

Avec notre application, un utilisateur peut ainsi :

- Se connecter en rentrant son Id et son mot de passe
- Choisir un pseudo non utilisé
- Changer ce pseudo à tout instant et en informer automatiquement les autres utilisateurs tout en conservant ses conversations passées
- Accéder à l'historique des conversations, même si l'interlocuteur n'est pas en ligne actuellement
- Voir le nom des utilisateurs connectés actuellement
- Envoyer et recevoir des messages textuels en temps réel
- Se déconnecter

Finalement, l'application que nous avons développée est aujourd'hui utilisable pour tchatter sur un même réseau local. Dans le futur et pour mieux respecter le cahier des charges, nous pouvons ajouter les fonctionnalités suivantes : pouvoir échanger des données non textuelles, et pouvoir communiquer sur un réseau distant.

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