

Projet POO

Louis Chauvet & Pierre Marsaa

Conception et diagrammes réalisés	2
Diagramme des cas d'utilisation	2
Diagramme de séquence d'une connexion	3
Diagramme de séquence d'envoi d'un message	4
Diagramme des classes	4
Architecture du système et choix technologies	5
Base de données	5
Autre librairie (Google collection)	5
GUI	5
Procédures d'évaluation de tests	6
Explication fonctionnement Github Actions	6
Explication fonctionnement Github	6
Explication fonctionnement Jira	6
Procédure d'installation et déploiement	7
Manuel d'utilisation	7
La connexion	7
Se renommer	8
Fonctionnement des profils	8
Communiquer	8

Conception et diagrammes réalisés

Diagramme des cas d'utilisation

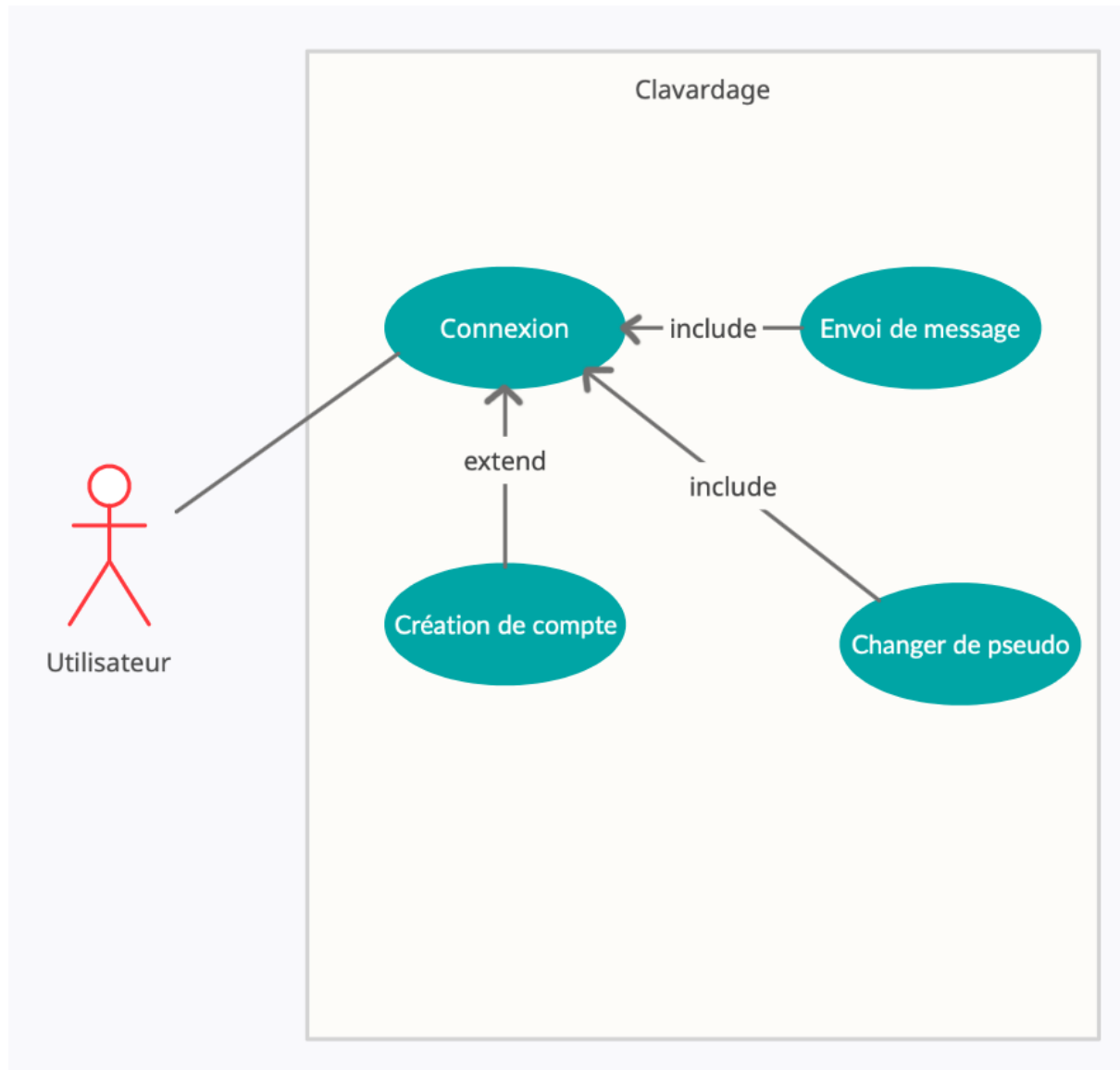


Diagramme de séquence d'une connexion

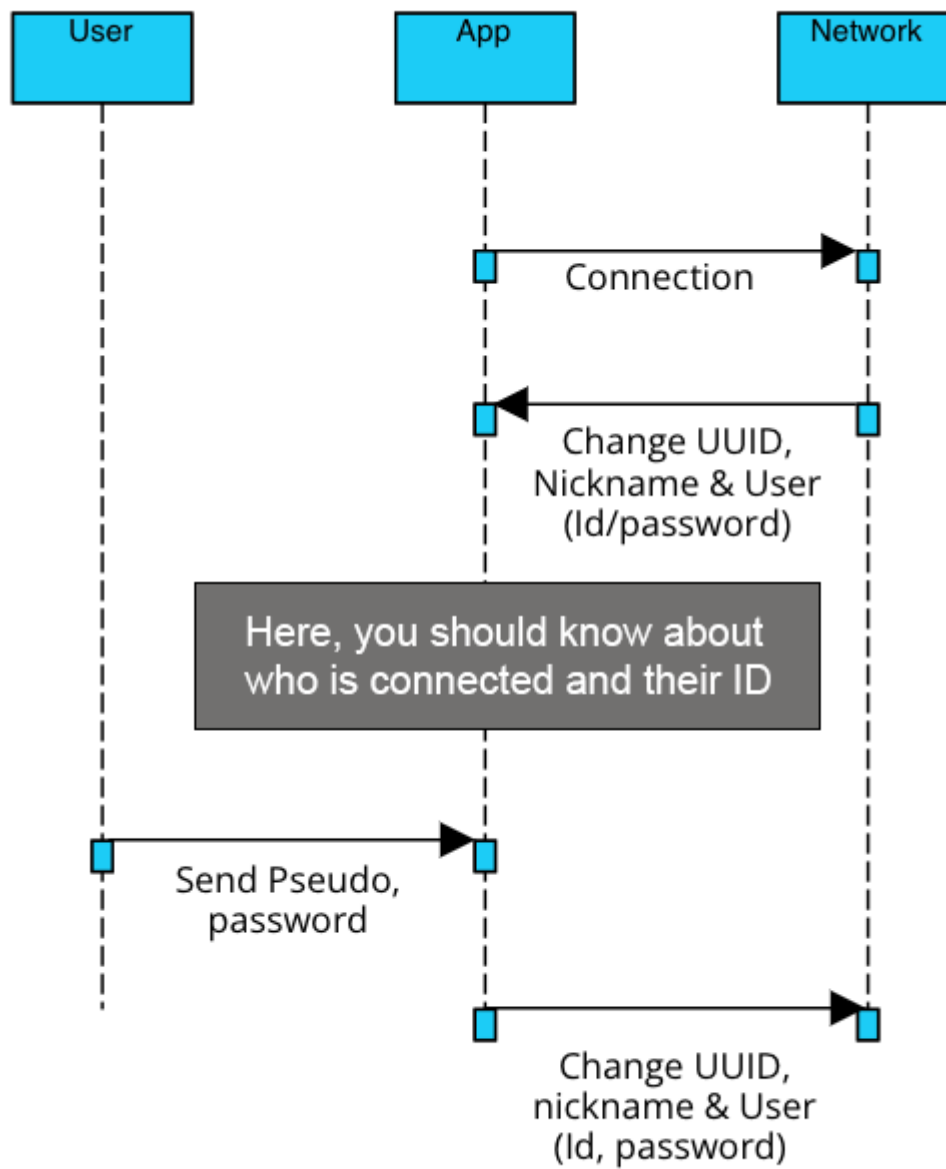


Diagramme de séquence d'envoi d'un message

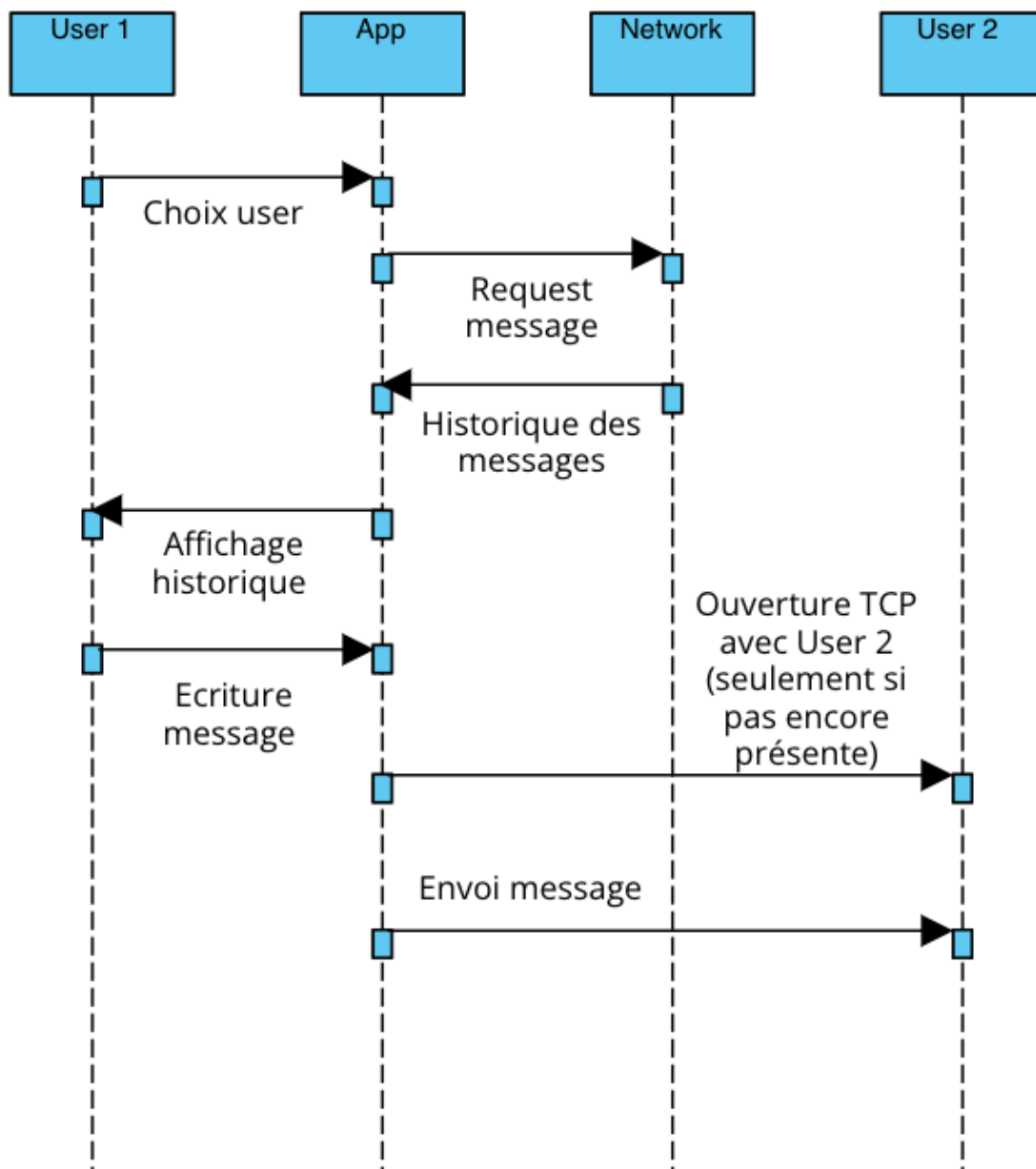


Diagramme des classes

Voir le fichier joint (classe.svg dans le github)

Architecture du système et choix technologies

Base de données

La base de données est entièrement décentralisée, chaque instance de l'application utilise une base de données SQLite. Pour assurer la cohérence entre les instances, il y a plusieurs éléments de synchronisation:

- Partage des identifiants/mots de passe au démarrage
- Récupération de l'historique des messages

Pour éviter les doublons, chaque utilisateur et chaque message est identifié par un UUID unique.

Autre librairie (Google collection)

Pour la gestion de la base de données et des données de connexion nous avons utilisé des hashmap symétriques, pour ne pas réinventer la roue nous avons utilisé la librairie GoogleCollection.

GUI

Pour le GUI, le choix a été d'utiliser SWING car vu en cours et assez complet. Au final, même si assez vieux et un peu compliqué à utiliser pour certains points (le responsive en l'occurrence). Nous avons décidé de refaire à notre sauce certains composants, car Swing était peu adapté à des moments (La liste des utilisateurs et son affichage par exemple), mais dans l'ensemble, l'outil reste suffisant.

Nous avons donc organisé notre projet en 2 panels principaux (gauche et droite) qui contiennent eux-mêmes un grand nombre de panels, mais dont le but final est d'afficher les utilisateurs (gauche) et les messages (droite).

Les choses les plus intéressantes à travailler (mais aussi les plus compliquées) étaient liées au responsive (redimensionnement de la fenêtre). Après avoir commencé avec du positionnement en absolu, nous avons finalement tout changé vers du GridBagLayout qui est semblable à une grille avec des cases pour mettre les éléments. De cette manière, nous avons réussi à bien organiser toute la page. Le second élément à travailler était l'affichage des messages, beaucoup de choses ont été essayé, BoxLayout, Absolute, GridBagLayout, même de l'html et css mais tout était trop limité ou trop complexe à utiliser étant donné que les messages peuvent être variable en longueur et en largeur. Nous sommes donc partis sur un StyledDocument qui est semblable à une seule et même zone de texte avec des styles personnalisables pour chaque texte. Même si les options de personnalisation ne sont pas très présentes, c'est la solution la plus propre que nous avons trouvée pour faire du responsive en ajoutant des éléments de différentes tailles au fur et à mesure.

La dernière chose que nous avons dû travailler plus que prévu était la liste des utilisateurs, nous voulions que les cases utilisateurs aient un certain design (photo, nom, heure, voyant de présence), et ceci n'était possible qu'avec des Panel. Malheureusement, tous les outils

intégrés de Swing gèrent les listes avec des Label seulement. Nous avons donc dû re-coder un gestionnaire d'élément semblable à une liste avec un listener pour pouvoir cliquer sur seulement un utilisateur et savoir qu'il est sélectionné.

Procédures d'évaluation de tests

Pour la procédure d'évaluation de tests, nous n'avons fait qu'un test JUnit, il est là pour tester la partie UDP. Tout le reste a été testé en condition réelle (plusieurs ordinateurs / machine virtuelle) avec envoi de message, connexion, déconnexion, changement de pseudo et arrivée d'un nouvel utilisateur.

Le test JUnit est utilisé pour valider le fonctionnement des éléments du réseau UDP, qui nécessitent la création manuelle de paquet réseau. La procédure de test est assez simple, on crée un paquet puis on le lit, si les données sont identiques on considère que le paquet est correctement créé et lu.

Explication fonctionnement Github Actions

Pour faciliter le déploiement nous avons créé des actions de compilation (vérification que le code en ligne peut être compilée à chaque commit) et de création de release à chaque tag. C'est donc facile de proposer un lien pour effectuer les mises à niveau chez les utilisateurs.

Explication fonctionnement Github

Nous avons travaillé sur une même branche tout au long projet. L'objectif au départ était de garder la branche "master" assez propre (c'est-à-dire un code qui compile et qui fonctionne) et d'utiliser des branches secondaires pour développer. Finalement, notre manière de travailler (grosses journées de travail non-stop, ensemble) ne se prêtait pas à cette organisation. Nous travaillions en direct tous les deux et enregistrions énormément de code d'un bloc.

Explication fonctionnement Jira

Nos backlogs et prédictions des tâches par rapport à Jira étaient assez variables, certaines tâches ont pris beaucoup plus de temps que prévu et d'autres ont pris moins de temps (comme dans beaucoup de projets).

Notre projet a été fait en 2 temps.

Dans un premier temps, nous avons décidé de développer le front et le back totalement séparément, cela représente notre premier sprint de 2 semaines, et puis, nous avons eu quelques soucis avec des points comme le responsive ou encore de l'affichage et donc nous avons changé de direction. Cela explique pourquoi, dans les sprints suivants, il y a donc eu des modifications importantes.

La partie qui au final nous a pris le plus de temps, et qui a le plus occupé les derniers sprint est la partie de la liaison front/back. Le fait d'essayer de les lier faisait remonter des problèmes qui n'apparaissaient pas au départ et donc pour certaines choses, nous avons dû, là aussi, tout reprendre.

Au niveau des estimations pour le temps des tâches, nous avons beaucoup utilisé l'expérience que nous avons pour essayer d'avoir une bonne estimation du temps que les tâches allaient prendre. Mais comme dit plus haut, les quelques problèmes inattendus dus au langage et à SWING nous ont fait perdre beaucoup de temps.

Pour l'organisation des sprints, tout s'est fait naturellement, nous avons privilégié les tâches primaires d'abord puis nous avons continué au fur et à mesure avec les tâches essentielles mais moins importantes.

Conclusion du développement

Notre méthode de travail qui consistait en de très grosses journées de développement est certes efficace car beaucoup de travail était effectué d'un coup mais rend le suivi du projet plus compliqué. Pour nos futurs projets, nous travaillerons de manières plus régulières afin d'être mieux organisés.

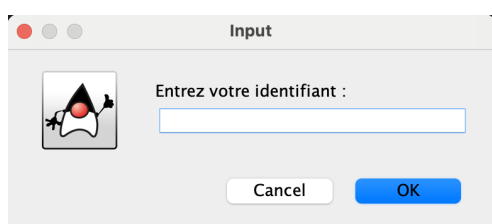
Procédure d'installation et déploiement

L'installation est assez simple, il suffit d'exécuter le fichier jar avec java 16. Il suffit ensuite de suivre le manuel d'utilisation.

Manuel d'utilisation

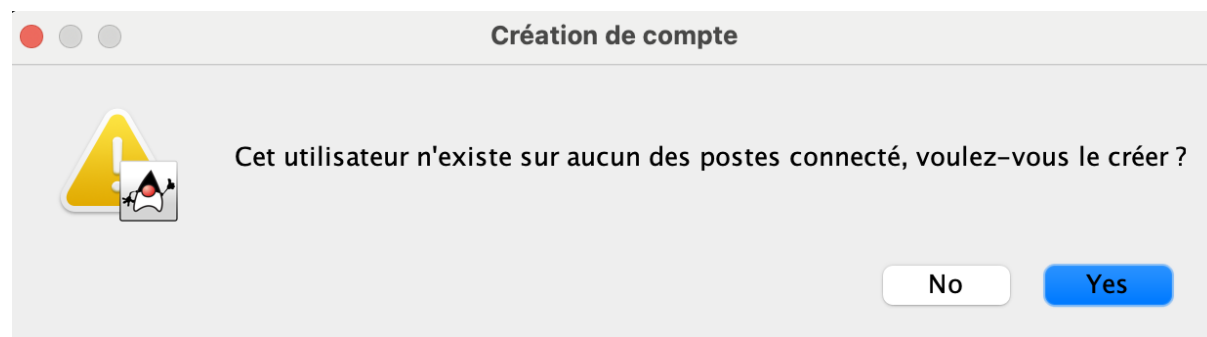
La connexion

Au lancement de l'application, deux fenêtres apparaîtront, une en fond, qui est l'application, et une popup, qui vous demandera de vous identifier. Voici à quoi ressemble cette dernière :



Vous pouvez donc entrer un pseudo, après avoir appuyé sur "OK", deux options s'offrent à vous.

Soit votre pseudo est nouveau, auquel cas nous vous proposons de créer un compte. Si vous cliquez sur "Yes", il vous sera alors demandé d'entrer un mot de passe pour ce compte.



Soit votre pseudo existe déjà, vous passez directement à l'étape "entrer le mot de passe". Si vous entrez le bon mot de passe, vous êtes connecté, sinon, vous pouvez recommencer cette procédure.

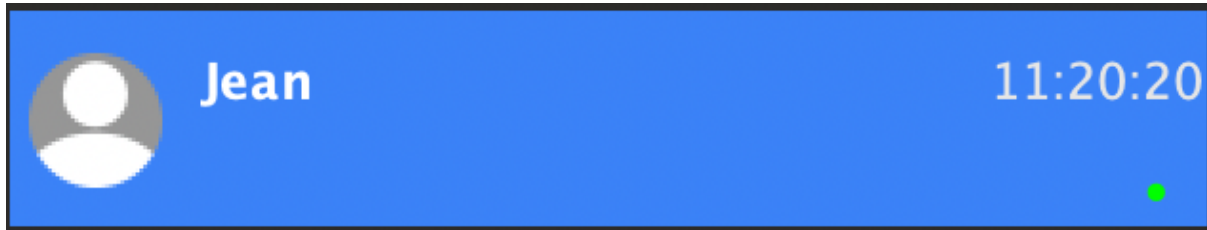
Se renommer

Vous avez la possibilité de vous renommer dans l'application, pour se faire, il vous suffit de cliquer sur la roue crantée qui se trouve en haut de la fenêtre.



Fonctionnement des profils

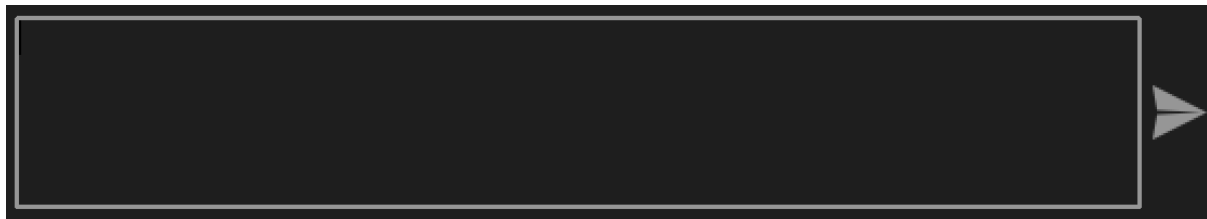
Vous pouvez trouver plusieurs informations sur la case de présentation d'un utilisateur sur la gauche du logiciel. En voici un exemple :



Vous pouvez voir ici la photo de profil de Jean (non personnalisable), son nom, l'heure du dernier message et enfin, un voyant lumineux, ce voyant indique si Jean est connecté ou non, si c'est vert, il est connecté, si c'est rouge, il est déconnecté.

Communiquer

Pour envoyer un message, vous devez écrire dans la zone de texte en bas à droite.



Une fois que le message est écrit, vous pouvez cliquer sur le symbole d'envoi à droite et le message s'enverra.

Une fois le message envoyé, il s'affiche de cette manière, en bleu si c'est vous qui l'avez envoyé, et en gris si c'est l'autre personne. L'heure et la date d'envoi sont aussi indiquées.

