

Rapport de ChatSystème

Réalisé par:

Zhang yu

Zeng yongjia

Groupe B _ 4 IR

Sommaire

1. Choix de technologie	
a. UDP/TCP	3
b. Base de Donnée	3
c. Swing	4
2. Processus de test	4
3. Souligner	5



Choix de technologie

a.UDP/TCP

1. UDP pour la Découverte des Contacts :

- **Légèreté et Vitesse** : L'en-tête UDP est moins volumineux que celui de TCP, ce qui rend les transmissions plus rapides.
- **Non Connexion** : UDP ne crée pas de connexion, cela nous convient à découvrir les contacts.

Dans notre projet, en ajoutant un identifiant spécifique au début du message (comme DECOUVERTE_ et REPONSE_), notre système peut reconnaître et traiter différents types de messages.

2. TCP pour l'Envoi des Messages :

- **Fiabilité** : TCP garantit que les données envoyées arrivent sans erreur et dans l'ordre.
- **Connexion Orientée** : Le caractère orienté connexion de TCP signifie qu'une connexion doit être établie entre l'expéditeur et le destinataire avant l'échange de données. et cela c'est nous besoin les données ne sont pas envoyées tant que la connexion n'est pas établie.

b.SQLite

Le choix de ne pas utiliser MySQL mais plutôt SQLite repose principalement sur la manière de stocker les informations. MySQL nécessite un serveur dédié et demande une maintenance régulière, ce qui n'est pas adapté pour notre projet en binôme. Par contre, SQLite ne requiert pas de serveur spécifique pour fonctionner, et nous pouvons implémenter les triggers ainsi que les opérations d'écriture avec du code. Ainsi, bien que les besoins des utilisateurs de notre projet ne soient peut-être pas très étendus, SQLite reste suffisant.



c.Swing

La conception de l'interface utilisateur de notre application repose sur la bibliothèque Swing, fournissant une plateforme pour le développement d'interfaces graphiques en Java. Ce choix est justifié par la nécessité d'une interface conviviale et interactive, assurant une expérience utilisateur optimale. Swing offre une variété de composants clés, tels que JLabel, JTextField et JButton, qui facilitent la création d'écrans de connexion, de pages de chat et de fenêtres de messagerie.

Processus de test

Pour tester les fonctionnalités que nous avons réalisées. Nous avons simulé des utilisateurs et établi une relation d'amitié entre eux, puis évalué la mise à jour du pseudonyme, ainsi que les processus de connexion et de déconnexion par vérification du nom et de l'état d'utilisation. Les tests concernant l'envoi et la réception de messages, ainsi que la mise à jour de la base de données des messages, ont été testés par vérification de l'intégrité de la base de données de réception.



Souligner

Code Structure->

Toutes les classes suivent le modèle MVC, en particulier dans Protocole (dédié à la transmission réseau). MessageListener est utilisé pour que la vue puisse être mise à jour en temps réel lorsque TCPReceiver reçoit des messages.

Synchronisation->

la classe MessageListener, la classe base de données

dans notre code, la synchronisation est utilisée principalement sur la MessageListener, dans la Base De Données , la TCPReceiver et la Messagerie pour assurer le timing de la base de données et de la view avec “ReentrantLock” et “Synchronized”.



