



Rapport de mini projet de la réalisation de l'application 'instant messaging'



Date:
14/01/2023

Filière :
Cycle d'ingénieurs en génie informatique

Module :
Réseaux informatiques

Responsable :
Prof. Fatima AMOUNASS

PLAN

I. Introduction

- A. Contexte et objectifs du projet
- B. Présentation de l'application de messagerie instantanée
- C. Description du fonctionnement (mode client/serveur, utilisation des pseudos)

II. Conception de l'application

- A. Choix de la technologie (Python, sockets, bases de données)
- B. Design de l'interface utilisateur
- C. Implémentation des fonctionnalités (envoi et réception de messages, gestion des utilisateurs, etc.)

III. Etapes de réalisation de l'application

- A. Utilisation d'une machine comme serveur et client
- B. Connexion de plusieurs clients
- C. Connexion de plusieurs machines à un serveur dans un réseau

IV. Conclusion

- A. Résumé des principales fonctionnalités de l'application
- B. Perspectives d'amélioration et de développement futur
- C. Conclusion générale sur le projet

INTRODUCTION :

A. CONTEXTE ET OBJECTIFS DU PROJET :

Le projet consiste à concevoir une application de messagerie instantanée ("Instant Messaging") pour permettre aux utilisateurs de dialoguer en temps réel sur le réseau. L'objectif de ce projet est de développer une application fonctionnelle, facile à utiliser et adaptée aux besoins des utilisateurs.

B. PRÉSENTATION DE L'APPLICATION DE MESSAGERIE INSTANTANÉE :

L'application de messagerie instantanée est un outil de "chat" qui permet à des internautes de discuter avec d'autres utilisateurs sur le réseau. Il fonctionne en mode client/serveur, avec un seul serveur et de multiples clients. Les utilisateurs peuvent envoyer et recevoir des messages en temps réel, et sont identifiés par un pseudo.

C. DESCRIPTION DU FONCTIONNEMENT

LE FONCTIONNEMENT DE L'APPLICATION EST BASÉ SUR LE MODÈLE CLIENT/SERVEUR, OÙ UN SEUL SERVEUR GÈRE LES CONNEXIONS ET LES ÉCHANGES DE MESSAGES ENTRE LES CLIENTS. LES UTILISATEURS SE CONNECTENT AU SERVEUR EN UTILISANT LEUR PSEUDO, ET PEUVENT ALORS ENVOYER ET RECEVOIR DES MESSAGES AVEC D'AUTRES UTILISATEURS CONNECTÉS. L'APPLICATION UTILISE LES SOCKETS POUR GÉRER LES CONNEXIONS RÉSEAU ET LES ÉCHANGES DE DONNÉES ENTRE LES CLIENTS ET LE SERVEUR.



La partie de conception de l'application comprend la sélection de la technologie (Python, sockets, bases de données), le design de l'interface utilisateur et l'implémentation des fonctionnalités



II. Conception de l'application

A. CHOIX DE LA TECHNOLOGIE :

Pour la conception de l'application, nous avons choisi d'utiliser Python en raison de sa popularité et de sa facilité d'utilisation pour la création d'applications réseau. Nous utiliserons également les sockets pour gérer les connexions réseau entre les clients et le serveur, ainsi que les bases de données pour stocker les informations relatives aux utilisateurs et aux messages.

B. DESIGN DE L'INTERFACE UTILISATEUR :

Le design de l'interface utilisateur de l'application a été conçu pour être simple, intuitif et facile à utiliser. Il comprend des fonctionnalités telles que l'envoi et la réception de messages, la gestion des utilisateurs et la présentation des conversations en cours. L'application prend en charge les fonctions de base de messagerie instantanée, telles que la possibilité de voir les utilisateurs en ligne, de créer des conversations de groupe et de partager des fichiers.

C. IMPLÉMENTATION DES FONCTIONNALITÉS :

Pour implémenter les fonctionnalités de l'application, nous utiliserons les sockets pour gérer les connexions réseau entre les clients et le serveur. Nous utiliserons également Visual Studio comme environnement de développement pour écrire le code de l'application.

Les sockets sont utilisés pour gérer les connexions réseau entre les clients et le serveur. Nous utiliserons les sockets TCP pour établir une connexion entre les clients et le serveur, et pour échanger les messages entre eux. Les sockets UDP peuvent également être utilisés pour les besoins de la mise à jour de l'état de la conversation en temps réel. Nous utiliserons également les thread pour gérer les connexions simultanées des clients.

Pour gérer les fonctionnalités de l'application, nous utiliserons Visual Studio pour écrire le code en python. Nous utiliserons les fonctions de la bibliothèque socket pour créer et gérer les connexions réseau, et les fonctions de la bibliothèque thread pour gérer les connexions simultanées des clients. Nous utiliserons également les fonctions de la bibliothèque de base de données pour stocker et récupérer les informations relatives aux utilisateurs et aux messages.

III. Etapes de réalisation de l'application

A. UTILISATION D'UNE MACHINE COMME SERVEUR ET CLIENT :

Une machine peut être utilisée à la fois comme serveur et client pour l'application de messagerie instantanée. En tant que serveur, la machine gère les connexions réseau entre les clients et gère les échanges de messages entre les utilisateurs. En tant que client, la machine peut se connecter au serveur et envoyer et recevoir des messages avec d'autres utilisateurs connectés. Cette configuration permet une flexibilité pour les utilisateurs qui souhaitent utiliser l'application à la fois comme un moyen de communiquer avec d'autres utilisateurs et comme un moyen de recevoir des messages.

B. CONNEXION DE PLUSIEURS CLIENTS

Pour connecter plusieurs clients à l'application de messagerie instantanée, chaque client doit d'abord se connecter au serveur en utilisant son adresse IP et un port spécifique. Le serveur utilise alors les sockets pour gérer les connexions réseau et les échanges de messages entre les clients. Les clients peuvent ensuite envoyer et recevoir des messages avec d'autres utilisateurs connectés, en utilisant leur pseudo pour s'identifier.

C. CONNEXION DE PLUSIEURS CLIENT À UN SERVEUR DANS UN RÉSEAU

Pour connecter plusieurs machines à un serveur dans un réseau, chaque machine doit d'abord avoir accès au réseau et connaître l'adresse IP du serveur. Les clients peuvent ensuite se connecter au serveur en utilisant l'adresse IP du serveur et un port spécifique. Le serveur utilise alors les sockets pour gérer les connexions réseau et les échanges de messages entre les clients. Les clients peuvent ensuite envoyer et recevoir des messages avec d'autres utilisateurs connectés, en utilisant leur pseudo pour s'identifier. Il est important de noter que les réglages de firewall et de routeurs pourraient nécessiter une configuration spécifique pour permettre la communication entre les différentes machines dans le réseau.



A. RÉSUMÉ DES PRINCIPALES FONCTIONNALITÉS DE L'APPLICATION :

L'application de messagerie instantanée a été conçue pour permettre aux utilisateurs de dialoguer en temps réel sur le réseau. Il fonctionne en mode client/serveur, avec un seul serveur et de multiples clients. Les utilisateurs peuvent envoyer et recevoir des messages en temps réel, et sont identifiés par un pseudo. Il permet également la gestion des utilisateurs (inscription, connexion, déconnexion), la gestion des conversations de groupe, la possibilité de voir les utilisateurs en ligne, la mise en place de notification pour les nouveaux messages, et la possibilité de partager des fichiers.

B. PERSPECTIVES D'AMÉLIORATION ET DE DÉVELOPPEMENT FUTUR :

Il y a plusieurs perspectives d'amélioration pour l'application, comme la possibilité d'ajouter une fonction de cryptage pour protéger les conversations, l'intégration avec d'autres plateformes de communication, et l'ajout de fonctionnalités supplémentaires telles que la vidéo et l'audio.

C. CONCLUSION GÉNÉRALE SUR LE PROJET :

Ce projet a permis de développer une application de messagerie instantanée fonctionnelle et adaptée aux besoins des utilisateurs. Les choix technologiques et les fonctionnalités implémentées ont été soigneusement sélectionnés pour garantir une expérience utilisateur optimale. Il y a des perspectives d'amélioration pour l'application pour continuer à la rendre plus adaptée aux besoins des utilisateurs et pour l'évolution des technologies.

