

Applicazione di Chat Client-Server

1. Introduzione:

L'applicazione di chat client-server è un sistema di messaggistica in rete progettato per facilitare la comunicazione in tempo reale tra più client su una rete. Questo documento tecnico fornisce una panoramica dell'architettura del sistema, delle funzionalità e dei componenti principali.

2. Architettura del Sistema:

Il software segue un modello architetturale client-server. È composta da due componenti principali:

- **Server:** Il componente server gestisce le connessioni dei client, la distribuzione dei messaggi e il coordinamento generale del sistema.
- **Client:** Il componente client fornisce l'interfaccia utente per inviare e ricevere messaggi, oltre a gestire le connessioni al server.

3. Componente Server:

Il componente server è responsabile delle seguenti attività:

- **Accettazione delle Connessioni dei Client:** Il server ascolta le connessioni in arrivo dai client utilizzando un socket. Quando un client si connette, il server stabilisce un canale di comunicazione con il client.
- **Gestione delle Richieste dei Client:** All'atto della connessione, a ciascun client viene assegnato un thread dedicato per gestire la comunicazione. Il server resta in ascolto dei messaggi dai client e li diffonde a tutti i client connessi.
- **Diffusione dei Messaggi:** Quando un client invia un messaggio, il server lo diffonde a tutti gli altri client connessi. Ciò garantisce che tutti i client ricevano aggiornamenti in tempo reale della conversazione.
- **Gestione delle Disconnessioni:** Il server rileva quando un client si disconnette in modo imprevisto e lo rimuove dall'elenco dei client attivi. Notifica inoltre agli altri client l'evento di disconnessione.

4. Componente Client:

Il componente client è responsabile delle seguenti attività:

- **Connessione al Server:** Il client stabilisce una connessione al server fornendo l'indirizzo host del server, il numero di porta e un nome utente univoco.

- **Invio dei Messaggi:** Gli utenti possono digitare messaggi nell'interfaccia del client e inviarli al server. Il client inoltra quindi il messaggio al server per la distribuzione agli altri client.
- **Ricezione dei Messaggi:** Il client resta in ascolto dei messaggi in arrivo dal server. Al ricevimento di un messaggio, lo visualizza nell'interfaccia utente per consentire all'utente di visualizzarlo.
- **Gestione dell'Interazione Utente:** Il client fornisce pulsanti per connettersi al server, inviare messaggi e chiudere l'applicazione. Gestisce anche eventi come la chiusura della finestra dell'applicazione.

5. Gestione degli Errori:

Il sistema include meccanismi di gestione degli errori per garantire robustezza e affidabilità. Principali scenari di errore e relativi meccanismi di gestione includono:

- **Connessione Persa:** Sia il componente client che quello server rilevano e gestiscono situazioni in cui la connessione tra loro viene persa in modo imprevisto. Puliscono le risorse, notificano gli altri componenti e forniscono feedback agli utenti.
- **Input Non Validi:** Il sistema convalida gli input degli utenti come l'indirizzo host, il numero di porta e il nome utente per evitare errori durante l'instaurazione della connessione.

5. Linee guida per l'esecuzione

-Server:

Il server va eseguito attraverso il comando "python Server.py --host host-address --port port-number" e può essere terminato attraverso il Keyboard Interrupt ALT+C da terminale.

-Client

Il client non richiede argomenti da linea di comando, può essere eseguito attraverso il comando python Client.py, il quale rende visibile la GUI dove viene predisposto il necessario per stabilire la connessione con il server.