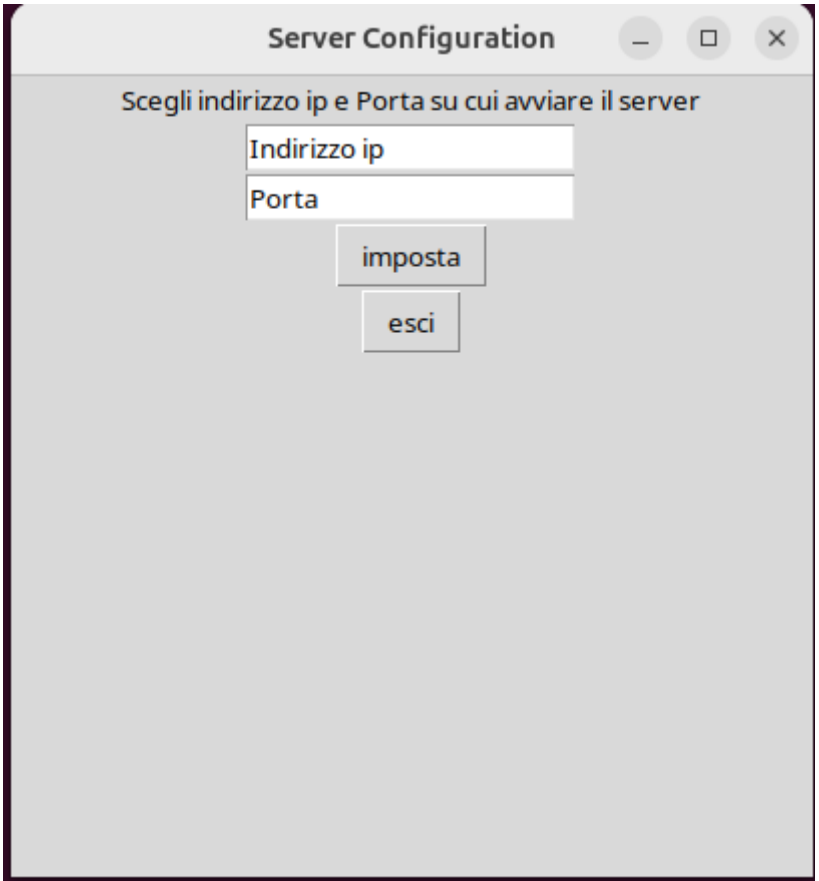
ChatSystem
ChatSystem permette di implementare un sistema di chat client-server. Utilizza “socket programming” e il linguaggio “python” sia per la gestione del server sia per la gestione dei client.
Chat system fornisce inoltre un interfaccia grafica sia per il controllo del server sia per l’utilizzo del client, tramite la libreria tkinter.

Requisiti & Dipendenze
Per utilizzare chat system bisogna aver installato sul proprio computer python 3, possibilmente alla versione 3.12.3 e la libreria tkinter.

Autore:
Paolo De Mori
Numero di Matricola: 0001071000

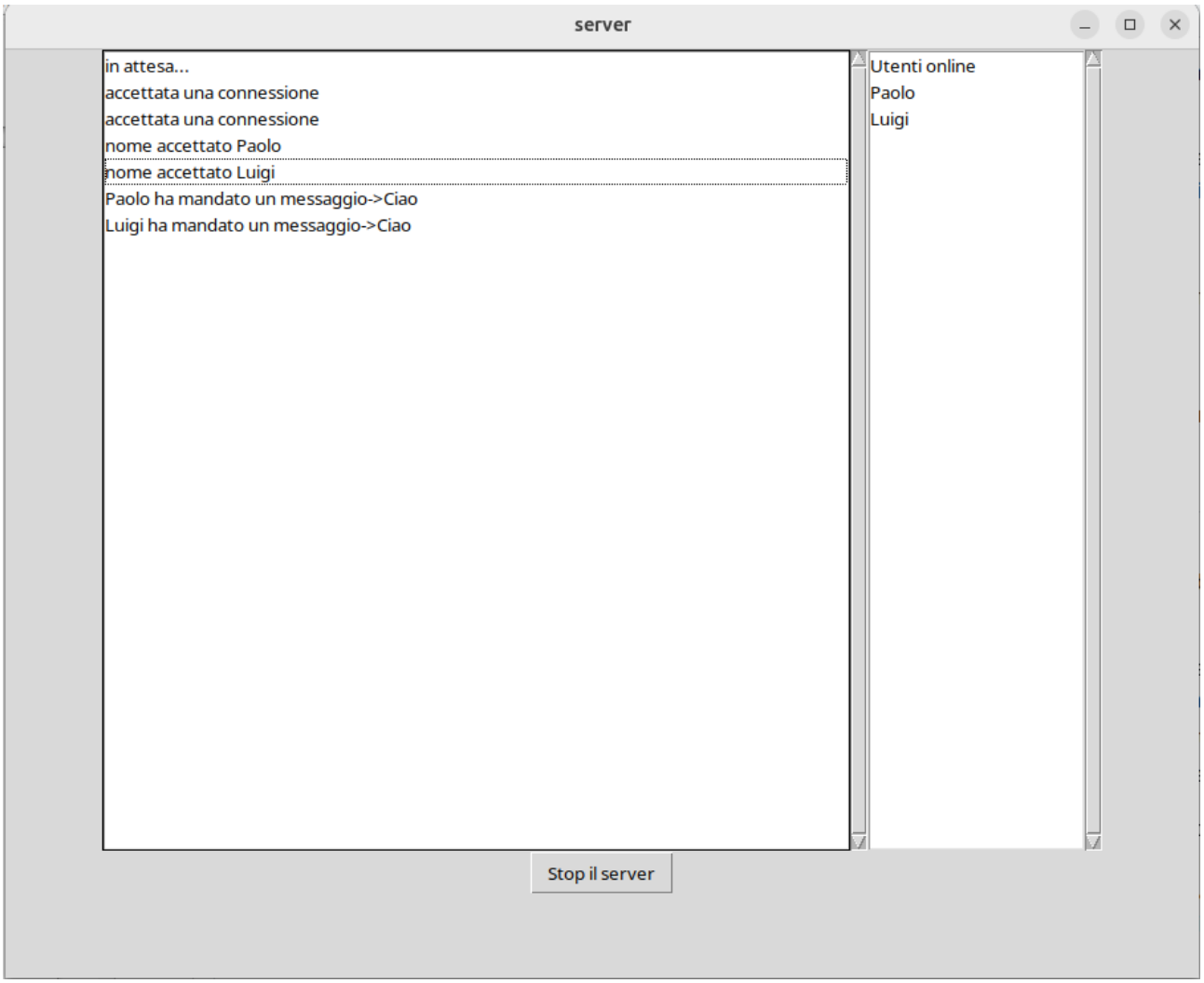
Guida a ChatSystem
Server:
Avvio:
Mantenendo tutti i file nella stessa cartella
- Su Linux/Mac dare i permessi di esecuzione al file “server.sh” ed eseguirlo
- Su Windows eseguire il file “server.bat”

GUI:
Impostare il Server:
- Una volta eseguito il file di avvio, compare una finestra che vi permetterà di decidere indirizzo ipv4 e porta su cui mettere in ascolto il server. Dopo aver inserito indirizzo e porta basta premere il pulsante “imposta” per avviare il server. Inoltre è presente un tasto “esci” per chiudere la configurazione
- In caso di inserimento di indirizzo o porta errati comparirà un messaggio di errore



Visualizzare stato Server:

- Dopo aver premuto il tasto “imposta”, vi si presenterà davanti una schermata rettangolare divisa in due parti:
    - A sinistra una schermata su cui appariranno messaggi che rappresentano gli avvenimenti e gli errori che si presentano sul server
    - A destra apparirà una schermata più piccola su cui sono presenti gli utenti connessi in quel momento
- Infine in basso è presente il pulsante “Stop Server” che se clickato disconnetterà tutti i client, chiuderà il server e interromperà l’applicazione



Client:

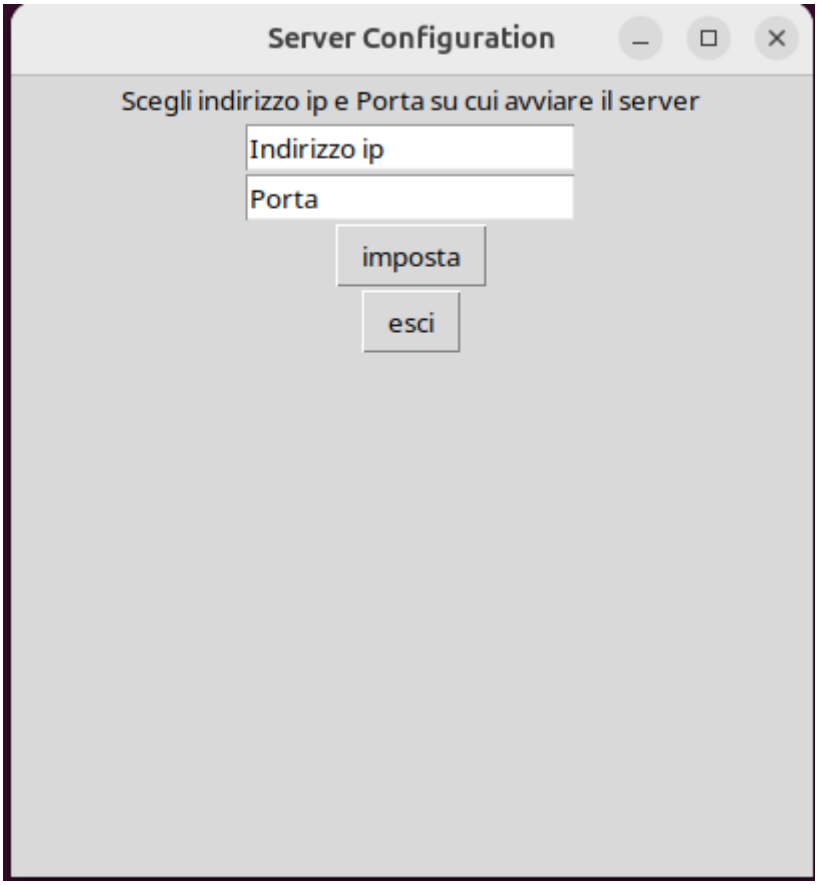
Impostare il Server:

- Su Linux/Mac dare i permessi di esecuzione al file “client.sh” ed eseguirlo
- Su Windows eseguire il file “client.bat”

GUI:

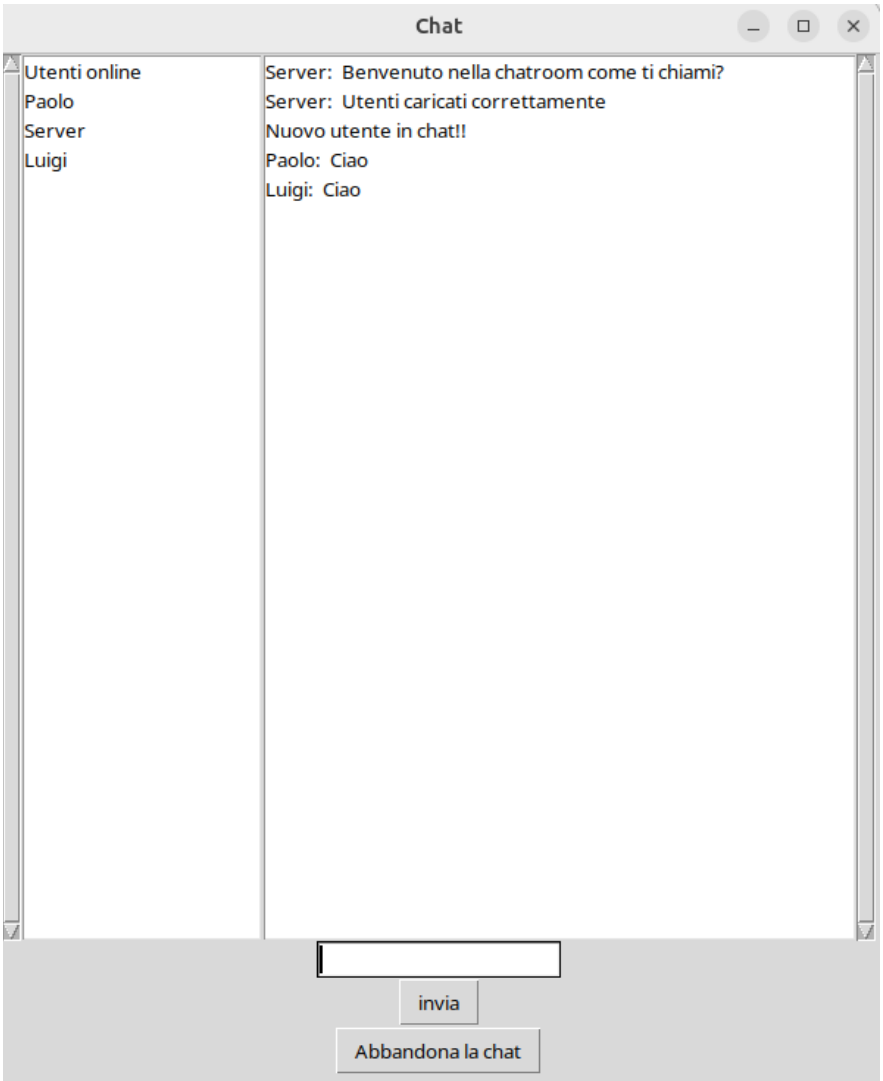
Impostare il Client:

- Una volta eseguito il file di avvio, compare una finestra che vi permetterà di decidere indirizzo ipv4 e porta su cui connettere il client. Dopo aver inserito indirizzo e porta basta premere il pulsante “imposta” per avviare il client. Inoltre è presente un tasto “esci” per chiudere la configurazione
- In caso di inserimento di indirizzo o porta errati comparirà un messaggio di errore



Utilizzare il Client:

- Dopo aver clickato il tasto “imposta” se l’indirizzo è stato inserito correttamente, apparirà una schermata rettangolare divisa in 3 parti:
  - A sinistra si potranno vedere in tempo reale gli utenti connessi al server
  - A destra verranno mandati i messaggi e le comunicazioni del server man mano che verranno inviati
  - In basso è possibile sia comporre e inviare i messaggi sia abbandonare la chatroom e contestualmente chiudere l’applicazione



Struttura dell’applicazione:

L’applicazione è composta oltre che dagli script d’avvio da 5 moduli python:

1. checkers.py: Che contiene funzioni per verificare l’indirizzo inserito in fase di configurazione e una funzione per il controllo della perdita di connessione tra 2 socket nella fase di invio di un messaggio.
2. guihelpers.py che contiene delle funzioni utili per l’interfaccia grafica
3. socketinit.py che si occupa della configurazione tramite interfaccia grafica e bind/connessione dei socket
4. client.py che contiene al suo interno il codice che gestisce l’interfaccia grafica e il funzionamento dei client
5. server.py che contiene al suo interno il codice che gestisce l’interfaccia grafica e il funzionamento del server

Funzionamento Server:

Lo script server.sh/bat si occupa di avviare il modulo server.py che per prima cosa utilizza il modulo socketinit.py per creare e mettere in ascolto il socket.
Successivamente si occupa di dichiarare le strutture dati per la memorizzazione dei socket e dei loro nomi, di avviare l’interfaccia grafica e di avviare un thread che si occupa della gestione delle richieste di connessione col server.
Il suddetto thread accetta le connessioni coi client, si assicura di aver stabilito correttamente la connessione e successivamente avvia un thread per ogni client che si dovrà occupare di loro(invio e ricezione di messaggi) per tutta la durata della connessione.
Il server inoltre può inviare ai client dei messaggi specifici quali

- __join__: che comunica ai client la partecipazione di un nuovo client alla chatroom
- __quits__:che comunica al client l’abbandono di un client
- __quit__:che comunica al client l’imminente chiusura della connessione

Il server in qualsiasi momento può decidere di interrompere la connessione con tutti i client, tramite la funzione “closeall” che invia a tutti i client online il messaggio “__quit__” e successivamente interrompe la connessione.

Funzionamento Client:

Lo script client.sh/bat si occupa di avviare il modulo client.py che per prima cosa utilizza il modulo socketinit.py per creare e connetere al server il socket.
Successivamente viene avviato un thread che si occupa di ricevere e mostrare all’utente i messaggi ricevuti e di gestire i 3 messaggi specifici (__join__,__quits__,__quit__) del server. Inoltre in caso di ricezione del messaggio __quit__ invierà al server il messaggio __quit2__ che confermerà l’avvenuta ricezione dell’imminente disconnessione.
L’invio dei messaggi e invece gestito dalla funzione sendmessage, che viene eseguita ad ogni pressione del pulsante “invia”.