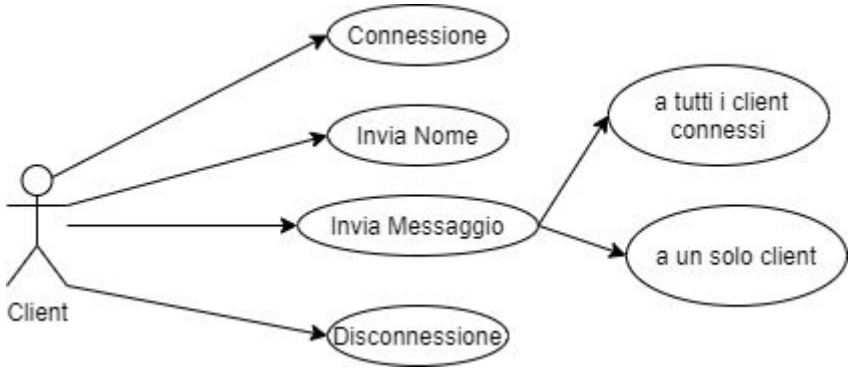
Specifiche dei requisiti

Descrizione della soluzione

Per la creazione di una Chat tra due o più client ho utilizzato i socket che permettono la comunicazione tra due processi diversi, inoltre è stato necessario l'utilizzo dei thread per la gestione di più client contemporaneamente.

Casi d'uso Client



Connessione

Descrizione: Il client creerà la connessione con il server, quindi è necessario conoscere l'ip e la porta dove il server è in ascolto.
Considerazioni aggiuntive: Una volta connesso il client dovrà gestire i due canali di comunicazione, uno verso il server e uno dal server, quindi uno di output e uno di input. Siccome i due canali devono essere gestiti secondo un ordine preciso e anche contemporaneamente i dati dal server li gestirà un thread e i dati che il client invia verranno gestiti da vari eventi (Actionlistener).
Risposta dal server: Il server quindi una volta connesso invierà ai client un messaggio di conferma della connessione

Invia nome

Descrizione: Il client utilizza un evento per l'invio del messaggio contenente il nome

Invia messaggio a tutti

Descrizione: il client utilizza un evento per l'invio del messaggio che verrà indirizzato ad un client connesso

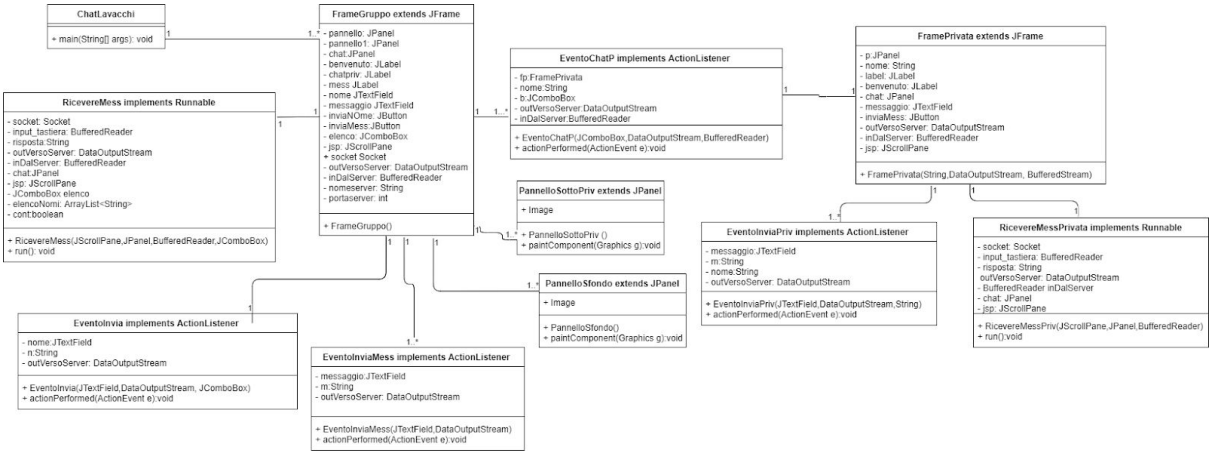
Invia messaggio a un solo client

Descrizione: il client dopo aver selezionato l'altro client con cui vuole comunicare invia il messaggio attraverso un evento

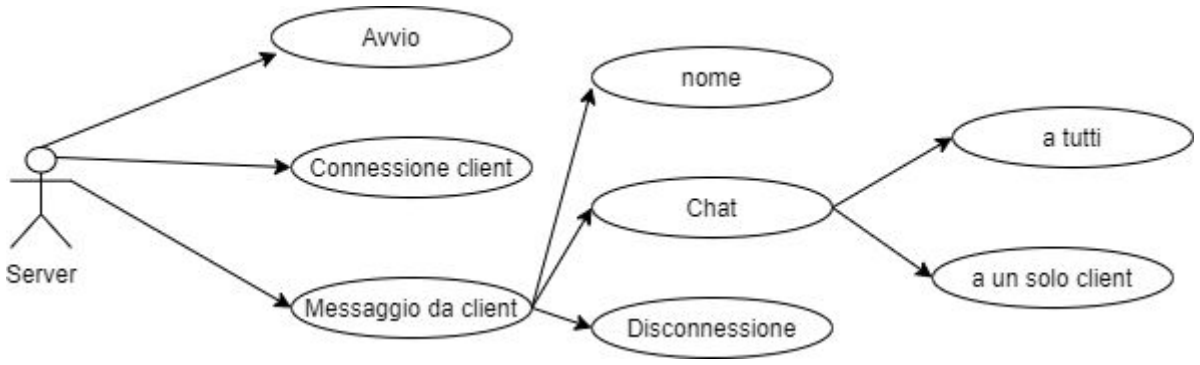
Disconnessione

Descrizione: il client per la disconnessione manderà un messaggio di addio
Risposta del server: Il server quindi confermerà l'uscita e la comunicherà ad ogni client connesso

Diagramma classi Client (guardare Javadoc per la spiegazione)



Casi d'uso Server



Avvio

Descrizione: Il server aprirà il socket e si metterà in attesa di un client

Connessione client

Descrizione:una volta che il client si è connesso gli verrà assegnato un threadserver che si occuperà dell'invio dei messaggi.

Messaggio da Client (Nome)

Descrizione: il client invia il proprio nome al server e quest'ultimo invierà un messaggio di tipo "nome" in cui comunica la connessione del nuovo client

Messaggio da Client (chat a tutti)

Descrizione: il client invia il messaggio al server e quest'ultimo invierà un messaggio di tipo "tutti" in cui rinvia agli altri client il messaggio e chi è il mittente.

Messaggio da Client (chat a un solo client)

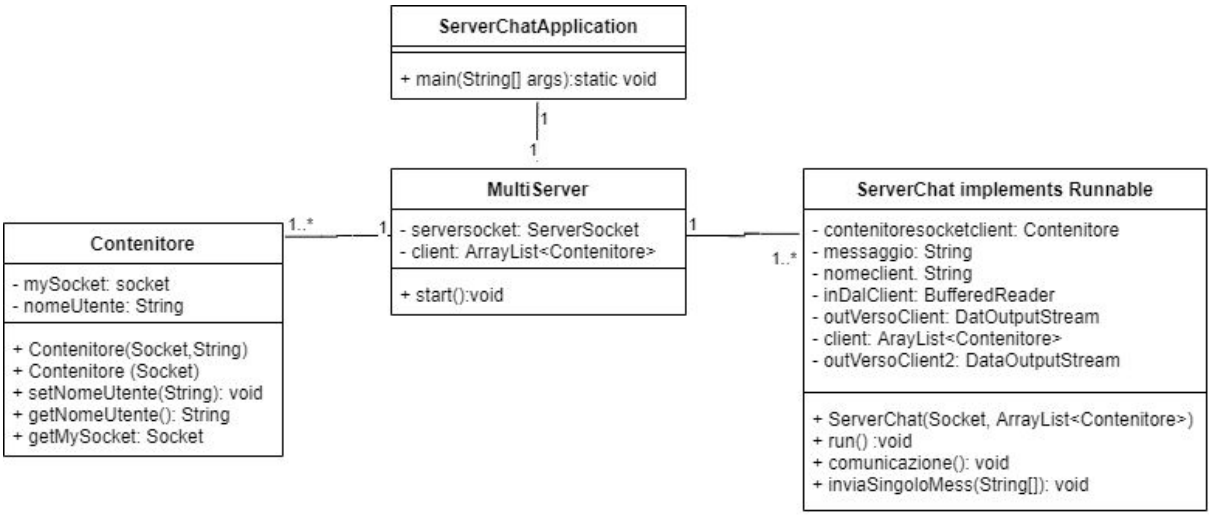
Descrizione: il client invia il messaggio al server e quest'ultimo invierà un messaggio di tipo "Chatpriv" in cui rinvia al client selezionato il messaggio.

Messaggio da Client (disconnessione)

Descrizione: il client invia il messaggio al server in cui scrive "ADDIO" e quest'ultimo invierà un messaggio di tipo "Disc" in cui avverte gli altri client connessi che un client si è disconnesso, comunicando anche il nome.

Lavacchi Ginevra 5BIA
15/11/2020

Diagramma classi Server (guardare Javadoc per la spiegazione)



Messaggi scambiati tra Server e Client

Messaggio	Mittente	Contenuto	Formato
nome Client	Client	nome del client	“nome”
messaggio1	Client	messaggio a tutti	“tutti:::messaggio”
messaggio2	Client	messaggio a un solo	“messaggio”
disconnessione	Client	messaggio a tutti	“addio”
connessione	Server	client connesso	“nome:::nomeclient:::si è connesso”
avvisi	Server	Avvisi	“avviso:::messaggio”
disconnessione	Server	client disconnesso	“avvisoDisco:::”+nomeclient+ “:::si è disconnesso”