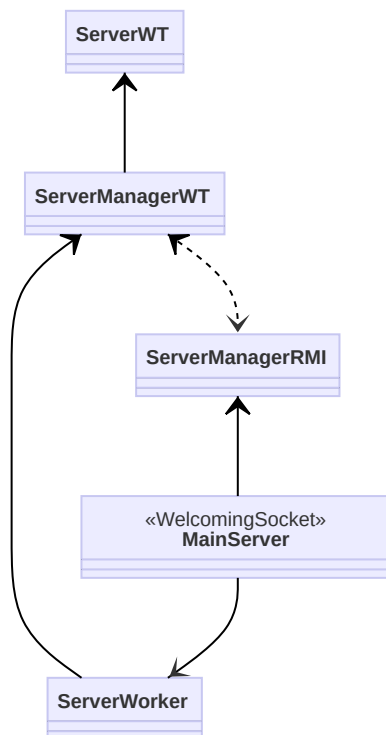


1. **Struttura classi**
2. **Ipotesi**
3. **Scelte**

1. Struttura classi

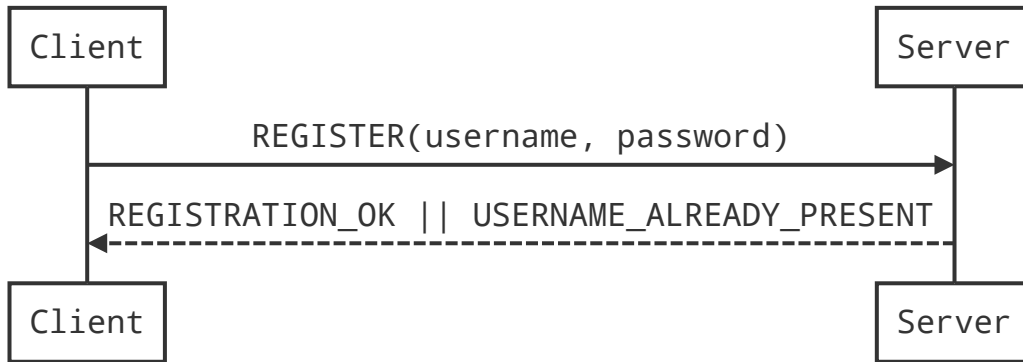


uso `ConcurrentHashMap` per le registrazioni in quanto tutte le operazioni sono garantite essere *thread-safe*.

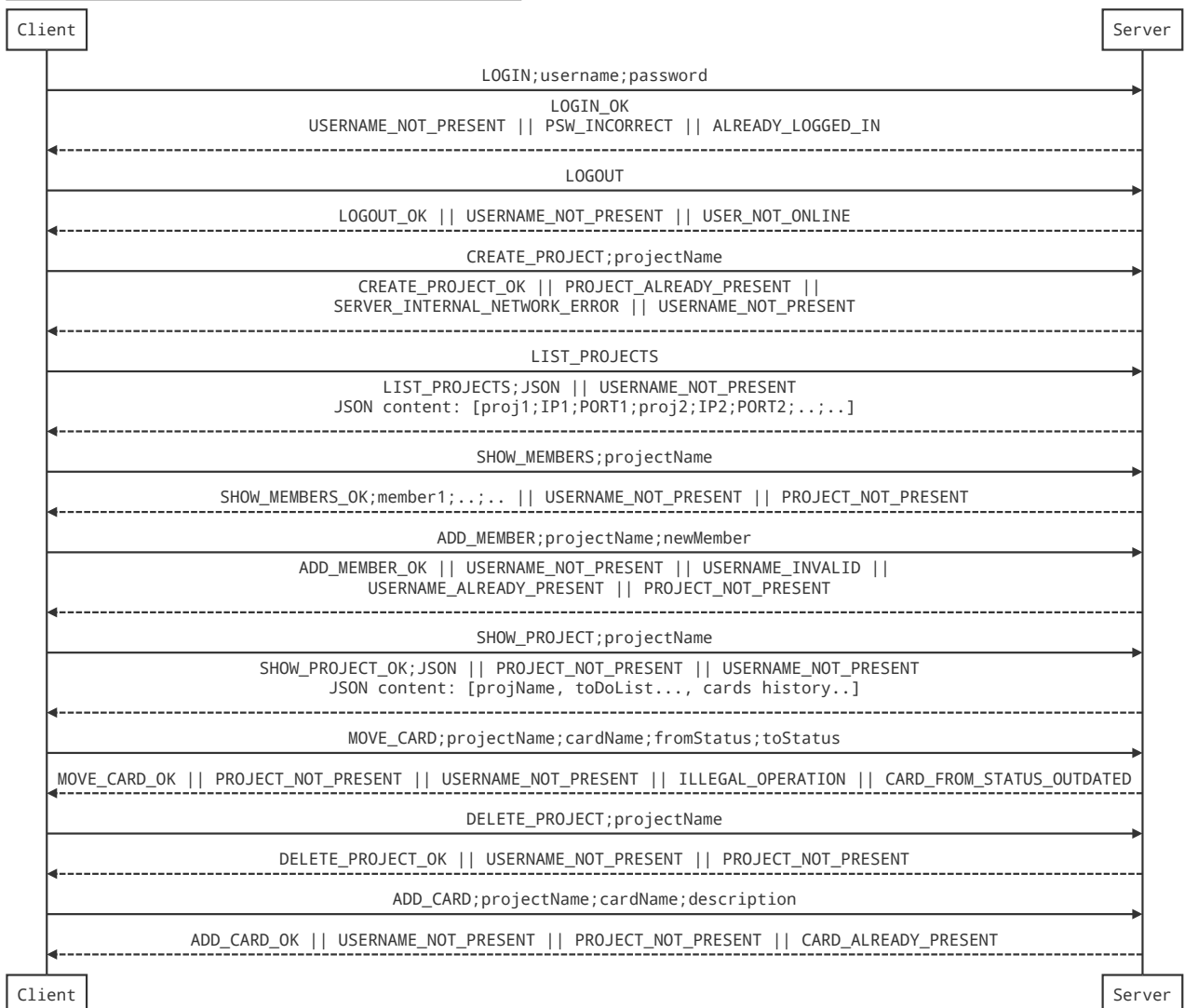
Non vi sono sovrapposizioni fra inserimenti e rimozioni in quanto un utente non può registrarsi ed effettuare il login contemporaneamente, pertanto reputo tale scelta la più efficiente garantendo l'accesso concorrente alla struttura dati.

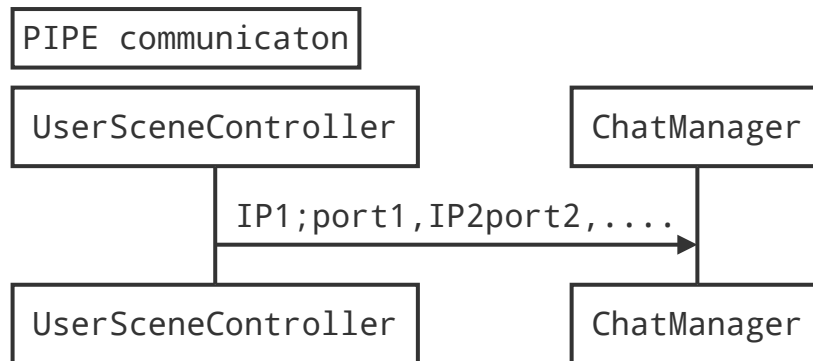
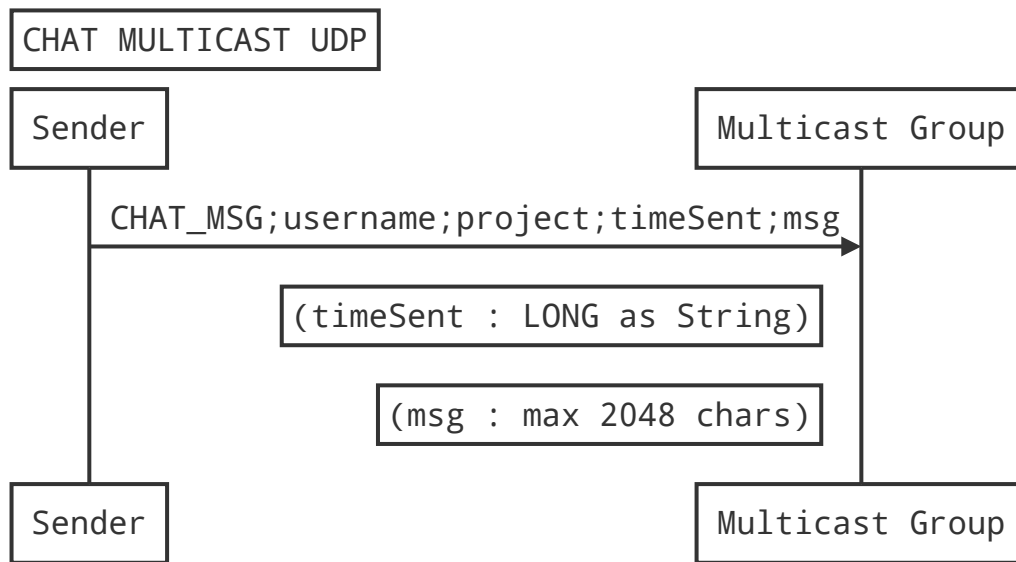
Di conseguenza non ho bisogno di gestire la concorrenza server-side riguardo l'RMI.

COMUNICAZIONE CLIENT-SERVER - RMI



COMUNICAZIONE CLIENT-SERVER: Protocollo, operazioni





2. Ipotesi

- un utente può loggarsi su una sola connessione (no login multipli)
- stateful
- limite superiore numero progetti dato dal numero di IP multicast locali
- scalabilità multithread + multiplexed

3. Scelte

quando la finestra della GUI del client viene chiusa viene mandata una richiesta di **EXIT** che informa il server di chiudere quella connessione TCP in quanto non verranno effettuati altri login.

Se il client aveva effettuato l'accesso in aggiunta viene inviato una richiesta di **LOGOUT**.

Comunicazione TCP stateful, lo stato di utente loggato è registrato durante la comunicazione, infatti non si ha bisogno di passare nuovamente il proprio username.

..

? é importante che threadChatManager e ClientWT usino dbHandler che non è sincronizzato

- Java DB Developer's Guide

Multi-Connection

From an application, using multiple connections to a Derby database and issuing requests against those connections on multiple threads.

«If thread

A does database work that is not transactionally related to thread B, assign them to different Connections»

Per l'aggiornamento automatico della Chat:

- passo al ChatManager una `ObservableList<ChatMsgObservable>` che al momento dell'istanziazione di un'istanza di `DbHandler` setta la variabile statica `currentChatMsgList` di quest'ultima classe (passando la ObservableList di cui sopra).