Aufgabe 2 – Chat-Anwendung  
(Client-Server-Architektur)

# Verbindungsaufbau (Handshake)

Die Verbindung zwischen Server und Client wird über TCP-Sockets realisiert.

Beim Handshake werden Server- und Client-Version abgeglichen, um die Kompatibilität beider Anwendungen sicherzustellen.

### Erfolgsfall (kompatibel)

1. Client: Protocol: *<Protocol-Version>* *Beispiel: Chat-0.0.1*
2. Server: ACCESS GRANTED

### Fehlschlag (ggf. inkompatibel)

1. Client: Protocol: *<Protocol-Version>*
2. Server: ACCESS DENIED – Supported Protocol: *<Protocol-Version>*

# Autorisierung

Bevor ein Client am Chat teilnehmen kann, muss er zuerst einen Nutzernamen wählen. Hierzu dient der Befehl: /login <Nutzername>  
**Ohne Autorisierung kann der Client weder Nachrichten an andere Chatteilnehmer senden, noch Nachrichten empfangen.**

# Protokoll

Befehle, die der Client an den Server sendet, müssen mit einem „/“ beginnen.  
Der Befehl und die Parameterliste, sowie die Parameter innerhalb der Liste, werden durch Leerzeichen voneinander getrennt.

# Unterstützte Befehle

|  |  |  |
| --- | --- | --- |
| **Befehl** | **Parameter** | **Beschreibung** |
| /msg | <Nachricht> | Sendet <Nachricht> an alle angemeldeten Nutzer |
| /help | - | Gibt alle verfügbaren Befehle aus |
| /users | - | Gibt eine Liste aller angemeldeten Nutzer aus |
| /list | - | -siehe users- |
| /rename | <Gewünschter Nutzername> | Ändert den eigenen Nutzernamen im Chat |
| /w | <Nutzername> | Sendet eine Privatnachricht an <Nutzername> |
| /poke | <Nutzername> | “Stupst” <Nutzername> an |
| /logout | - | Meldet den Client vom Chat ab und beendet die Verbindung |

# Serverantworten

|  |  |
| --- | --- |
| **Clientbefehl** | **Serverantwort** |
| /login <Nutzername> | /LOGINSUCCESS <Nutzername> |
| /rename <Nutzername> | /RENAMESUCCESS <Nutzername> |
| /users  /list | /USERLIST<\n><Liste von Benutzernamen, getrennt durch „\n“> |