Chat-Server

Inhalt

[Aufgabenbeschreibung 1](#_Toc93311447)

[Benutzerhandbuch 1](#_Toc93311448)

[Troubleshooting von Fehlermeldungen 2](#_Toc93311449)

[Klassendiagramm 2](#_Toc93311450)

[Dynamischer Aufbau 2](#_Toc93311451)

# Aufgabenbeschreibung

Für Entwicklung verteilter Anwendungen in Java bei Herr Manuel Duque-Anton haben Lena Morawietz und Manuel Gentes ein Chat-Programm entwickelt.

Dabei kann ein Nutzer anderen aktiven Nutzern schreiben und von diesen Nachrichten erhalten solange beide mit dem Server der Hochschule über VPN verbunden sind.

Die Funktionalitäten wären:

* Anmelden (Eingabe eines Nutzernamens und Passworts) und Registrieren
* Auswahl verschiedener Personen zum Chatten
* Speichern der Nachrichten in einer Datei (Pro Nutzer zu Nutzer/Chat eine Datei) nach jeder Nachricht

# Benutzerhandbuch

Für den Admin:

1. Mit dem VPN der Hochschule verbinden
2. In Putty mit der IP 10.0.3.26 verbinden
3. Mit den Nutzerdaten Student1 und mdaws21 einloggen
4. Den Befehl ausführen: java -jar server.jar

Für den Nutzer:

1. Mit dem VPN der Hochschule verbinden
2. Das Programm starten
3. Den Namen und das Passwort in die vorgegeben Felder schreiben
   1. Als neuer Benutzer: Auf Registrieren klicken. Beim nächsten Start kann man diese als Login benutzen.
   2. Als Benutzer: Auf Login klicken.
4. Das Chatfenster öffnet sich.
5. Auf der linken Seite kann man jetzt alle aktiven Nutzer sehen und einen Chatpartner auswählen
6. Auf der rechten Seite kann man unten die Nachrichten eingeben und absenden. Darüber werden dann die eigenen und die Nachrichten des Chatpartners angezeigt.

## Troubleshooting von Fehlermeldungen

|  |  |
| --- | --- |
| “Es kann keine Verbindung hergestellt werden.” | Bitte überprüfe, ob du mit dem VPN der Hochschule verbunden bist und ob der Server gestartet ist. |
|  |  |
|  |  |
|  |  |

# Klassendiagramm

# Dynamischer Aufbau

Verbindung über TCP

CLIENTS:

Main-Thread im Client (Verbinden, Einloggen/Registrieren, Nutzer auswählen -> neuer Chat-Thread)

Jeder Client ein Thread pro Chat

SERVER:

Main-Thread im Server (Verbindungsthreads erstellen)

Verbindungsthreads (Verbindung zu Clients, Austausch der Nachrichten zwischen Clients)

KeepAlive-Threads und Fehlermeldungen in Clients und Server