

Ausbildung Schulprojekt zum

Fachinformatiker Anwendungsentwicklung

Thema:

Instant Messaging Dienst

Von

Lennart Sommerfeld

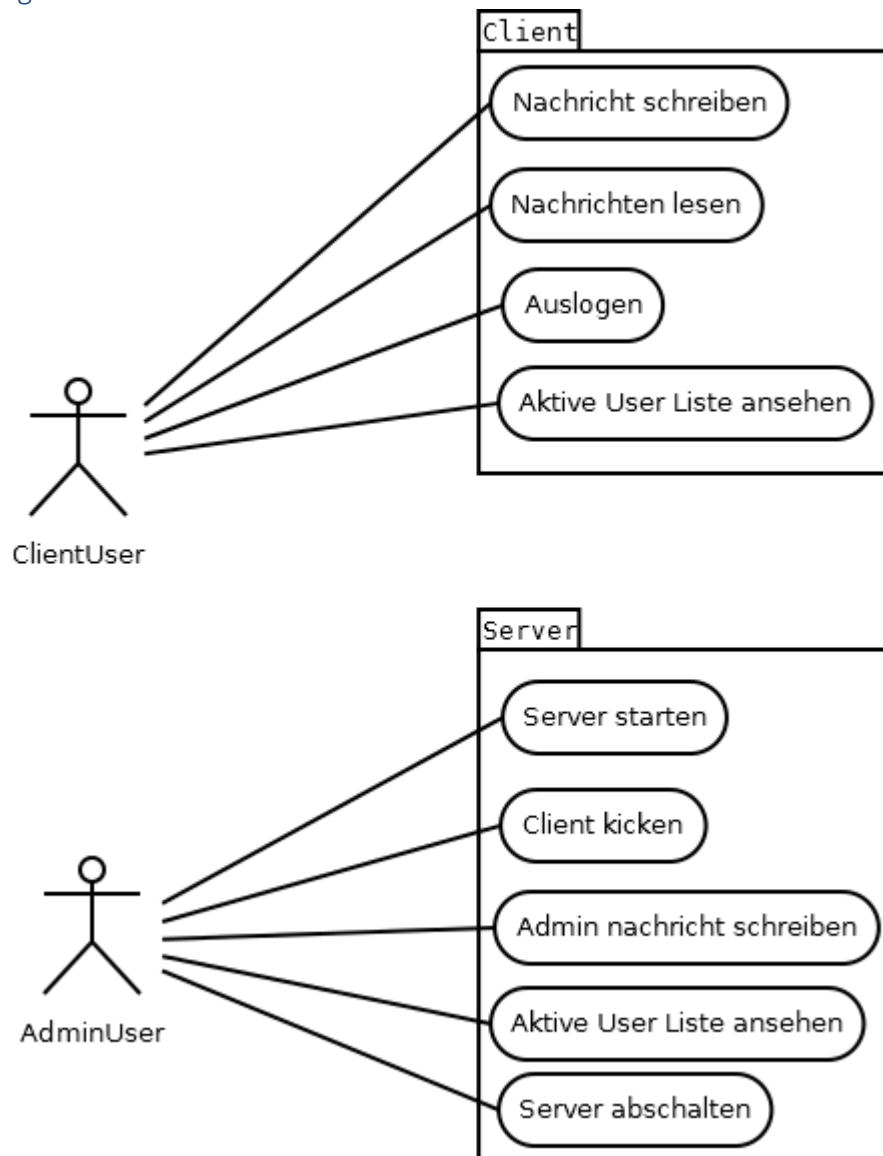
Inhaltsverzeichnis

<u>Das Projekt.....</u>	<u>3</u>
<u>Use-Case Diagramm.....</u>	<u>4</u>
<u>Der Server.....</u>	<u>5</u>
<u>Aufgaben des Servers.....</u>	<u>5</u>
<u>UML Diagramm.....</u>	<u>5</u>
<u>Server Klassen Beschreibung.....</u>	<u>6</u>
<u>Server.....</u>	<u>6</u>
<u>ConsoleListener.....</u>	<u>6</u>
<u>ClientConnectionListener.....</u>	<u>6</u>
<u>ClientSender.....</u>	<u>6</u>
<u>ClientData.....</u>	<u>6</u>
<u>Log.....</u>	<u>6</u>
<u>Der Client.....</u>	<u>7</u>
<u>Aufgaben des Clients.....</u>	<u>7</u>
<u>UML Diagramm.....</u>	<u>7</u>
<u>Client Klassen Beschreibung.....</u>	<u>7</u>
<u>Client.....</u>	<u>7</u>
<u>ClientMainGui.....</u>	<u>7</u>
<u>ClientMessengerGui.....</u>	<u>7</u>
<u>ClientLoginGui.....</u>	<u>8</u>
<u>Client.....</u>	<u>8</u>
<u>Das Nachrichtenprotokoll.....</u>	<u>9</u>
<u>senderID.....</u>	<u>9</u>
<u>date.....</u>	<u>9</u>
<u>messageTyp.....</u>	<u>9</u>
<u>messageText.....</u>	<u>9</u>

Das Projekt

Ziele dieses Projekts ist es einen Dienst zu entwickeln, der es ermöglicht zwischen mehreren Personen Nachrichten auszutauschen. Es wird einen Server geben, der die Nachrichten entgegennimmt und wieder an die Client austellt. Über die Clients können die User mit den anderen Teilnehmern chatten.

Use-Case Diagramm

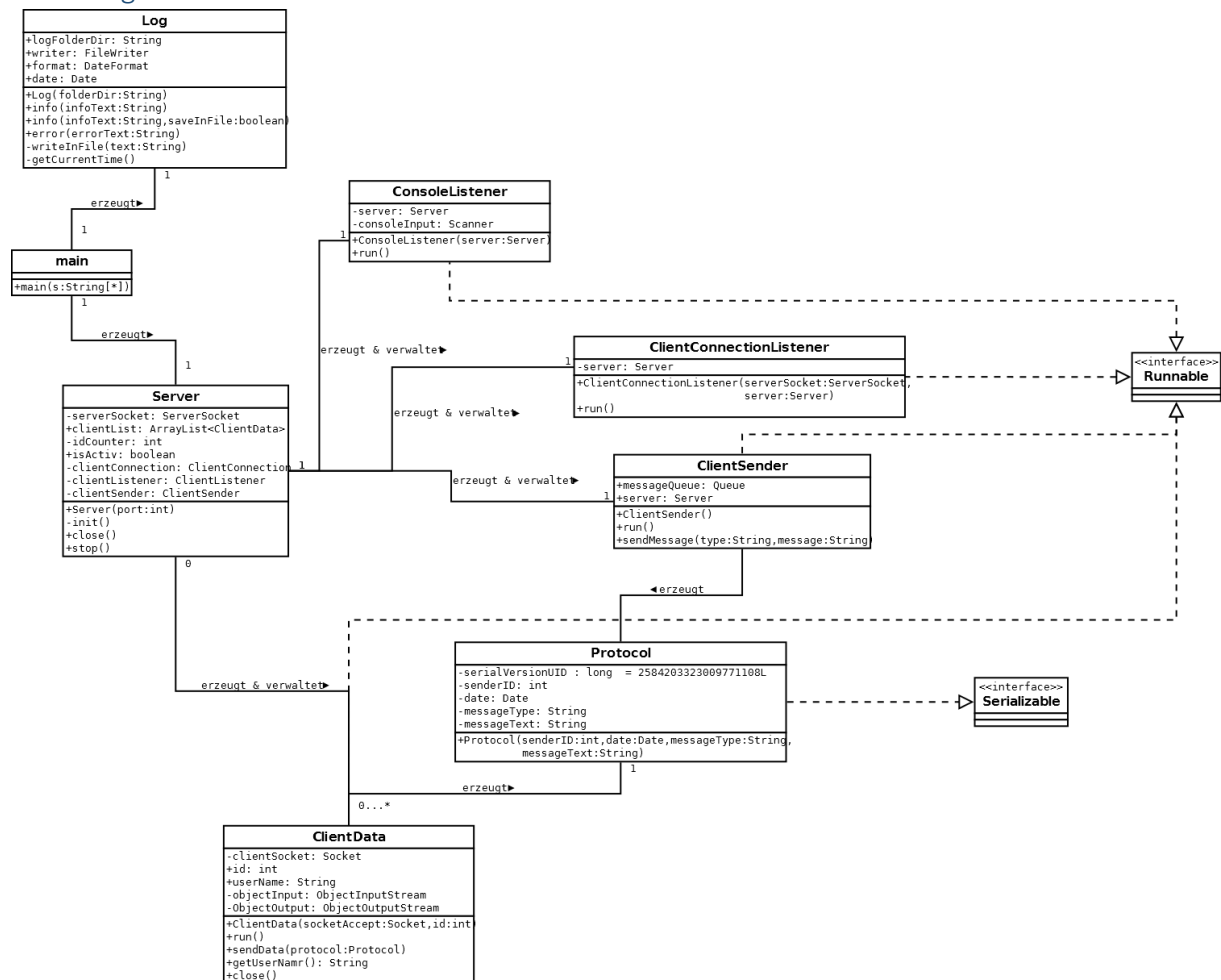


Der Server

Aufgaben des Servers

- Verwaltung von mehreren Clients
- Nachrichten an den Clients verteilen
- Überprüfen auf Online Status der Clients
- Log Datei mit schreiben

UML Diagramm



Server Klassen Beschreibung

Server

Aufgabe dieser Klasse ist es:

- Server Socket bereit zustellen
- Überprüfen ob sich Clients angemeldet haben
- Verbundene Clients in einer HashMap speichern

ConsoleListener

Reagiert auf Consoleneingaben des Admins und verwaltet diese

ClientConnectionListener

Diese Klasse überprüft neue eingehende Verbindungen und speichert diese in einer ArrayList

ClientSender

Diese Klasse verteilt die übergeben Nachricht an allen Clients.

Nachrichten werden in einer Queue gespeichert und nacheinander an alle Clients verschickt

ClientData

In dieser Klasse werden alle clientspezifische Daten gespeichert:

- Eindeutige ID
- Name des Users
- Socket
- In- und Output-Stream

Der eingehende Stream des Clients werden hier verwaltet.

Log

Diese Klasse schreibt Logdateien und gibt Texte auf der Console aus.

- ## Client Klassen Beschreibung

ClientLoginGui

Diese Klasse erzeugt ein Login-Formular. Folgende Daten können eingegeben werden:

- IP Adresse
- Port Nummer
- Username

Client

Erzeugt die Socketverbindung und händelt die Ein- und Ausgabe-Streams mit Hilfe der Klassen ClientOutput und ClientInput.

Das Nachrichtenprotokoll

Das Protokoll standardisiert den Datenaustausch zwischen dem Client und Server

senderID

Jedes Protokoll bekommt eine eindeutige ID des Senders

date

Datum, wann das Protokoll erstellt wurde

messageTyp

Gibt an, um was für einen Aktionstyp es sich handelt:

- message (einfache Nachricht)
- checkStatus (fordert den Status des Zieles an)
- logout (Verbindung schließen)
- login (Verbindungsaufbau)

messageText

Hier wird die Nachricht oder optionale Parametern übergeben

(getrennt durch eine „|“)