Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

InIT Institut für Angewandte Informations-technologie

Lernaufgabe

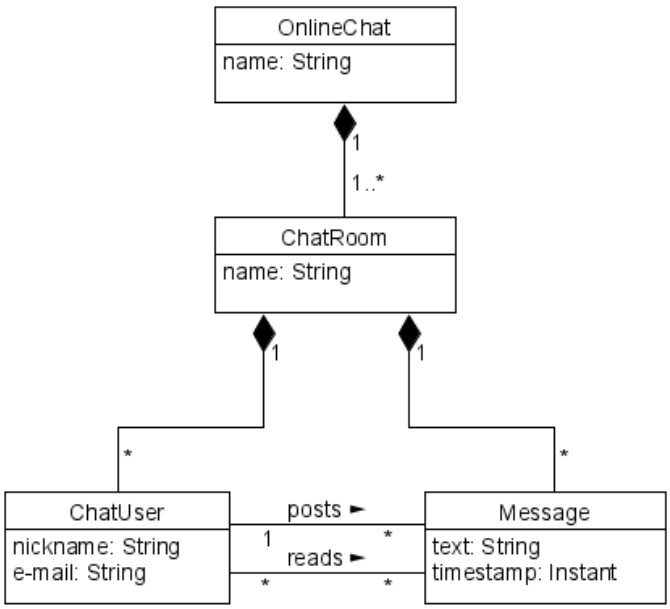
V1 – Verteilte Systeme

Lernziel

Sie sind in der Lage, eine einfache verteilte Applikation mit gängigen Design Patterns zu entwerfen.

Einleitung

Es soll eine Online-Chat-Applikation entworfen werden. Das folgende Domänenmodell dient als Basis für den Entwurf.



Chat (von englisch to chat «plaudern, sich unterhalten») oder Online-Chat bezeichnet die elektronische Kommunikation mittels geschriebenem Text in Echtzeit, meist über das Internet.

Ein Chat-Benutzer schreibt (engl. posted) eine Meldung in den Chat, der sofort bei allen angemeldeten Chat-Benutzern angezeigt und gelesen wird. Chat-Benutzer können sich in privaten Chat-Räumen austauschen, um nur mit bestimmten Chat-Benutzern chatten zu können.

Die wichtigsten funktionalen Anforderungen sind mit den folgenden Use Cases kurz beschrieben.

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

InIT Institut für Angewandte Informations-technologie

UC01 – Join a Chat

Ein Chat-Benutzer meldet sich im Chat mit einem Kurznamen (engl. nickname), optional mit E-Mail-Adresse und mit Angabe eines Chat-Raumes. Ohne Angabe eines Chat-Raumes ist der Chat-Benutzer automatisch im öffentlichen Chat-Raum mit dem Namen «Public»

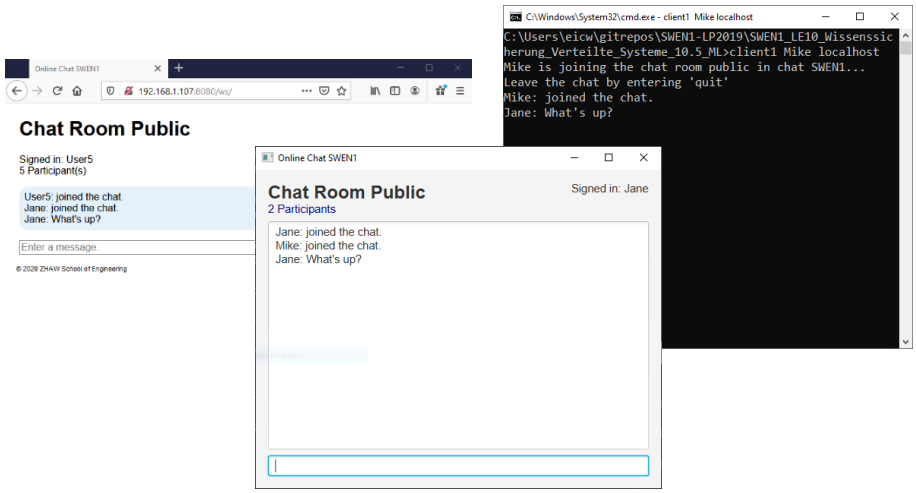
UC02 – Post a Message

Ein Chat-Benutzer schreibt eine Meldung in den Chat. Alle im Chat-Raum aktiven Chat-Benutzer können die Meldung sofort lesen.

UC03 – Leave a Chat

Ein Chat-Benutzer meldet sich wieder aus dem Chat ab und verlässt den Chat-Raum.

Zur Illustration der Use Cases ist hier ein Chat mit verschiedenen Chat-Clients abgebildet (Web, JavaFX, CLI).



Hier noch ein paar wichtige Qualitätsanforderungen und Randbedingungen, die den Lösungsraum einschränken.

- Die Applikation soll als verteiltes System auf verschiedenen Clients (PC, Tablet, Mobile) funktionieren.
- Die Applikation soll vollständig in Java implementiert werden.
- Für die Kommunikation zwischen den verschiedenen verteilten Komponenten (Client, Server) soll Java RMI eingesetzt werden (vorerst nur im Intranet einer Firma nutzbar).
- Die Applikation soll so entworfen werden, dass sie einfach für verschiedene Technologien im UI (Web, JavaFX, CLI etc.), der Kommunikation (Java RMI, REST, WebSocket etc.) und der Persistenz (DB, File, Memory etc.) verwendbar bzw. erweiterbar ist.

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

InIT Institut für Angewandte Informations-technologie

Aufgabe V1.1: Logische Architektur

Erstellen Sie eine erste grobe logische Architektur für den Online-Chat. Gehen Sie dabei davon aus, dass als Middleware Java RMI verwendet wird für die Kommunikation zwischen Client und Server. Für die Clients werden JavaFX und CLI verwendet.

Vorgehen

1. Überlegen Sie sich, was für einen Architekturstil für verteilte Systeme angewendet werden könnte.
2. Entwerfen Sie grob die Aufteilung der Applikation in Schichten (engl. Layers), Komponenten und Schnittstellen für den gewählten Architekturstil.
3. Modellieren Sie detaillierter die ein- und ausgehenden Schnittstellen der Domänenschicht (Abhängigkeiten) zu Clients (mit Java RMI) und der Persistenz (z.B. In-Memory DB). Clients und Persistenz müssen intern nicht detailliert werden, sondern nur deren Schnittstellen zur Domänenschicht.
4. Dokumentieren Sie Ihre logische Architektur, sodass die Schichtung, eine erste Aufteilung in Komponenten und deren Abhängigkeit zwischen den Schichten diskutiert werden kann.
5. Verifizieren Sie, ob alle Anforderungen und Einflussfaktoren für die Softwarearchitektur mit Ihrer Lösung erfüllt werden.

Hinweise, Tipps

- Wählen Sie eine Schichtung und Verteilung für einen Thin Client, wo die Domänenschicht (engl. domain layer) ausschliesslich auf dem Server und auf dem Client nur die Präsentationsschicht (engl. presentation layer) verteilt sind.

Ergebnis

Eine logische Architektur für den Online-Chat dargestellt mit einem UML-Paketdiagramm.

Zeit: 20’

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

InIT Institut für Angewandte Informations-technologie

Aufgabe V1.2: Design UC01 – Join a Chat

Erstellen Sie ein Design für die Realisierung der Systemoperation «join(nickname: String, email: String = «Unknown», chatroom: String = “Public“)», die das Hauptszenarios des Use Cases realisiert.

Vorgehen

1. Erstellen Sie ein erstes Design-Klassendiagramm (DCD) abgeleitet aus dem Domänen-modell für den Online-Chat.
2. Entwerfen Sie die Systemoperation «join(…)» und skizzieren Sie den Ablauf mit einem Sequenzdiagramm.
3. Aktualisieren Sie das DCD, so dass statisches und dynamisches Modell konsistent sind.

Hinweise, Tipps

- Keine

Ergebnis

Ein DCD und ein Sequenzdiagramm für die entworfene Systemoperation.

Zeit: 15’

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

InIT Institut für Angewandte Informations-technologie

Aufgabe V1.3: Design UC02 – Post a Message

Erstellen Sie ein Design für die Realisierung der Systemoperation «post(nickname: String, chatroom: String = “Public“, message: String)», die das Hauptszenarios des Use Cases realisiert.

Vorgehen

1. Entwerfen Sie die Systemoperation «post(…)» und skizzieren Sie den Ablauf mit einem Kommunikationsdiagramm.
2. Aktualisieren Sie das DCD, so dass statisches und dynamisches Modell konsistent sind.

Hinweise, Tipps

- Keine

Ergebnis

Ein aktualisiertes DCD und ein Kommunikationsdiagramm für die entworfene Systemoperation.

Zeit: 15’