Pflichtenheft

11.11.2011

Projekt: HWR-Chat-Programm

Autoren: Christian Gebauer, Sebastian Große, Jonathan Wiens, Benjamin Pfeiffer, Nico Smeenk

Modul: Software-Engineering I

Studiengang: Informatik 2010

Fachbereich: Fachbereich 2, Duales Studium Wirtschaft • Technik, HWR Berlin

1 Vorwort

Dieses Pflichtenheft ist eine Vereinbarung zwischen der Dozentin Frau Prof. Dr. Monett Diaz als Auftraggeberin und der Gruppe 1 des Kurses "Software-Engineering I" aus dem Studiengang "Informatik 2010". Die Vereinbarung enthält alle Anforderungen an ein zu erstellendes Chat-Programm für die Hochschule für Wirtschaft und Recht Berlin. Das Dokument dient zur Strukturierung und genaueren Einschätzung des zeitlichen Aufwandes des Projektes und der Priorisierung der einzelnen Module. Somit lässt sich auch einschätzen, ob und mit welchen Kosten das Projekt durchführbar ist.

2 Einleitung

Das Chat-Programm soll den Studenten, aber auch den Dozenten die Möglichkeit bieten, sich auf einer für die HWR-Berlin einheitlichen Plattform schnell und unkompliziert zu kontaktieren und sich über studienrelevante Themen auszutauschen. Dabei können sich beispielsweise Studenten mit ihren Kommilitonen oder auch mit den Dozenten bei offenen Fragen oder Problemen in Verbindung setzen und um Rat oder Unterstützung fragen. So kann eine neue Plattform entstehen, die einerseits organisatorische Angelegenheiten, aber auch den Austausch lernspezifischer Inhalte optimiert und erleichtert. Mit dem Programm präsentiert sich die HWR-Berlin als aufgeschlossene Bildungseinrichtung, die sich modernen Trends nicht verschließt. Sie entwickelt sich stetig weiter, um den Studenten und Dozenten bestmögliche Lern- und Lehrbedingungen bieten zu können.

Inhaltsverzeichnis

3 Produkteinsatz

3.1 Anwendungsbereiche

Dieser Chat soll im täglichen Hochschulbetrieb der HWR-Berlin für folgende Punkte eingesetzt werden:

- Organisatorisches und Ankündigungen
- erleichterte Kommunikation zwischen Studenten
- erleichterte Kontaktaufnahme zu Dozenten und anderen Mitarbeitern

Mithilfe dieser Punkte sollen bessere Lern- und Lehrbedingungen an der Hochschule geschaffen werden.

3.2 Zielgruppen

Folgende Zielgruppen soll das Chatprogramm ansprechen:

- Studenten der HWR-Berlin
- Dozenten der HWR-Berlin
- sonstige Mitarbeiter der HWR-Berlin

3.3 Betriebsbedingungen

Folgende Bedingungen sind für den Betrieb des Chats relevant:

- Der Chat-Server soll auf einem Server der HWR-Berlin betrieben werden, zu dem die User des Chats eine Zugriffsmöglichkeit benötigen.
- Die User benötigen ein gültiges Nutzerprofil, welches durch einen Administrator angelegt wird.
- Ein Administrator wird benötigt, der sich um Probleme (z.B.: Vandalismus, Missbrauch) im Chat kümmert und diese beseitigt.

4 Zielbestimmung

Für das zu entwickelnde Chatprogramm sind mehrere Anforderungen vom Auftraggeber in Zusammenarbeit mit den Entwicklern definiert und priorisiert worden.

4.1 Kundenanforderungen

Diese Anforderungen sind direkt vom Auftraggeber vorgegeben worden.

4.1.1 Muss-Kriterien

Diese Leistungen sind für das Projekt existenziell und müssen daher erfüllt werden.

#1 Eins zu eins Chat
Es soll möglich sein mit einer Person privat zu schreiben.
Priorität: 3

#2 Gruppenchat
Es soll möglich sein in einer Gruppe zu chatten.
Priorität: 3

#3 Kontaktliste
Jeder User hat eine Kontaktliste, in der ihm bekannte Personen aufgelistet sind.
Priorität: 3

#4 Dateitransfer
Es soll einem User möglich sein, einer anderen Person eine Datei zu schicken.
Priorität: 3

4.1.2 Soll-Kriterien

Diese Anforderungen sollten implementiert.

#5 Statusanzeige
Jeder User hat eine Statusanzeige. (online/offline)
Priorität: 2

#6 Logging
Ein User hat die Möglichkeit, eine Konversation zu loggen.
Priorität: 2

4.1.3 Kann-Kriterien

Hier werden Anforderungen gelistet, die zusätzlich implementiert werden sollen, wenn noch Reserven an Zeit und Budget vorhanden sind.

#7 Profilseite
Jeder User hat eine Profilseite, auf der seine persönlichen Informationen hinterlegt sind.
Priorität: 1

4.2 Systemanforderungen

Aus den Kundenanforderungen ergeben sich ergänzend folgende Systemanforderungen.

4.2.1 Muss-Kriterien

Diese Leistungen sind für das Projekt existenziell und müssen daher erfüllt werden.

#8 Benutzer erstellen
Nur der Administrator kann Nutzerprofile erstellen.
Priorität: 3

#9 Benutzer einloggen
Es soll die Möglichkeit geben, dass sich ein bereits registrierter Benutzer in das System einloggen kann. **Falls das nicht möglich ist, erhält er eine Fehlermeldung.**
Priorität: 3

#10 Kontakt hinzufügen
Es soll die Möglichkeit geben einen Kontakt hinzuzufügen. Dieser bekommt dann eine Mitteilung und kann darauf eine Bestätigung oder Ablehnung senden. Personen, die eine Anfrage noch nicht akzeptiert haben, werden in der Kontaktliste als Offline aufgeführt.
Priorität: 3

4.2.2 Soll-Kriterien

Diese Anforderungen sollten implementiert.

#11 Gruppe erstellen
Jeder Nutzer soll eine Gruppe erstellen können. Dazu muss nur der Name angegeben werden. Teilnehmer können neue Teilnehmer einladen, siehe #12.
Priorität: 2

#12 in Gruppe einladen
Ein Teilnehmer einer Gruppe soll andere Teilnehmer in die Gruppe einladen können.
Priorität: 2

4.2.3 Kann-Kriterien

Hier werden Anforderungen gelistet, die zusätzlich implementiert werden sollen, wenn noch Reserven an Zeit und Budget vorhanden sind.

#13 Optionen
Es soll eine Möglichkeit geben benutzerdefinierte Einstellungen in einer Konfigurationsdatei zu speichern. Hier könnnen die Serveradressen stehen, das Design verändert werden und auch das Logging-Verhalten angegeben werden.
Priorität: 1

#14 Nutzerdaten bearbeiten
Jeder User soll die Möglichkeit haben, seine eigenen Nutzerdaten auf seiner Profilseite zu bearbeiten.
Priorität: 1

5 Produktübersicht

Der HWR-Chat ist ein Chatprogramm, das übliche Chatfunktionen wie persönlicher Chat, Gruppen-Chat, Dateiübertragung sowie Strukturierung von Kontakten in einer Liste u.a. bietet.
Der Chat ist nicht öffentlich zugänglich, sondern explizit für Studenten, Dozenten und andere Mitarbeiter der HWR-Berlin vorgesehen. Aus diesem Grund ist das Anlegen eines neuen Profils nur durch einen Administrator möglich.

5.1 Systemarchitektur

Das folgende UseCase-Diagramm beschreibt die grundlegende System-Architektur des HWR-Chats (siehe Abb. 1).

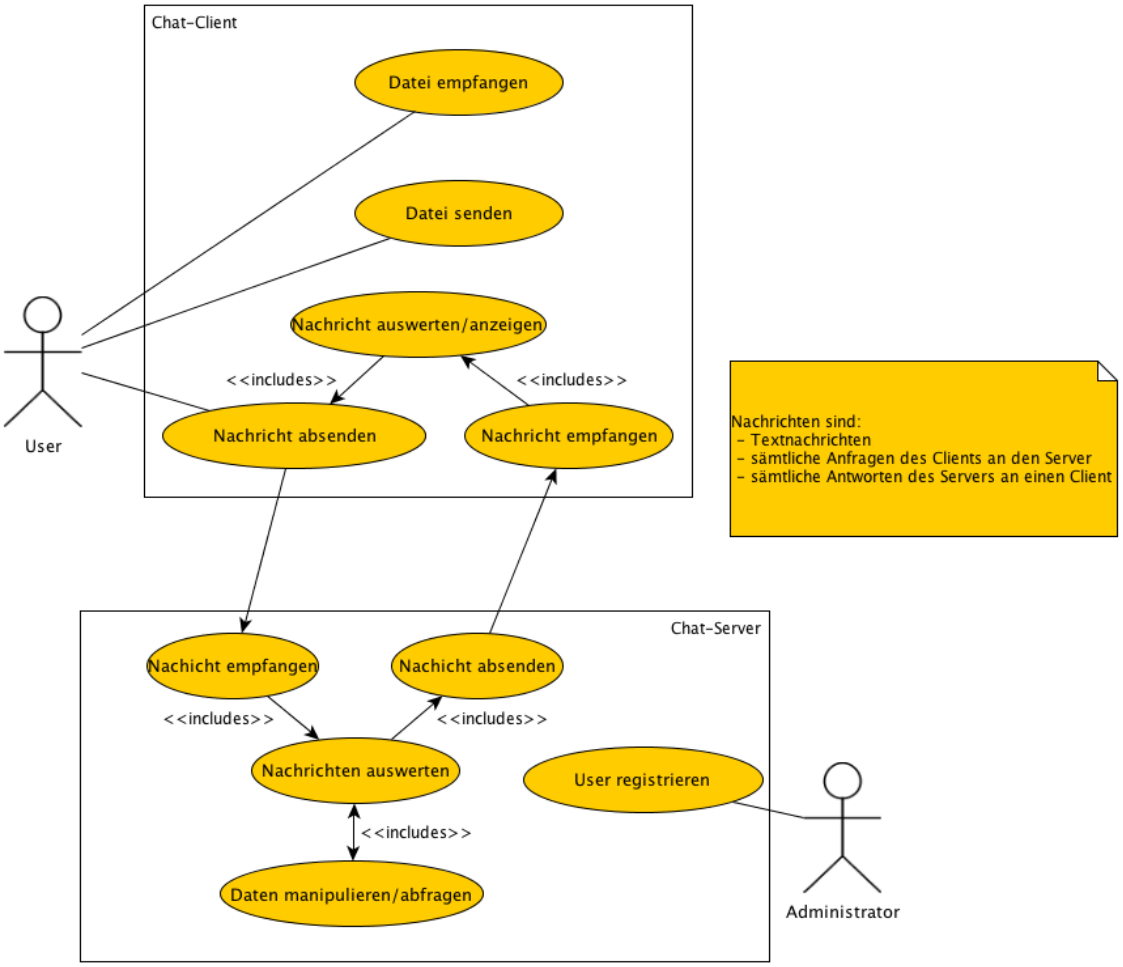

Abb. 1: Die Systeme Chat-Client und Chat-Server beschreiben die UseCases für User und Administratoren

Sämtliche Kommunikation zwischen dem Server und dem Client erfolgt in Form von Nachrichten. Dabei kann es sich um folgende Typen handeln:

- Auf Seite des Clients:
  - Login Anfrage
  - Freunde Suche
  - Freunde Einladung
  - Gruppen Einladung
  - Textnachricht
  - Dateiversand Anfrage
  - Statusänderung
- Auf Seite des Servers:
  - Login Bestätigung
  - Freunde Suche Ergebnis
  - Freunde Einladung (Weiterleitung von Client)
  - Gruppen Einladung (Weiterleitung von Client)
  - Textnachricht (Weiterleitung von Client)
  - Dateiversand Anfrage (Weiterleitung von Client)
  - UpdateMessage (ein User ändert beispielsweise seinen Status, Freunde müssen darüber in Kenntnis gesetzt werden)

Die an einen Kontakt gerichteten Nachrichten werden vom Server interpretiert, verarbeitet und an den adressierten Client weitergeleitet. Die übrigen Nachrichten, die an den Server gerichtet sind, verarbeitet dieser direkt weiter.
Nur die Übertragung von Dateien erfolgt Peer-to-Peer zwischen zwei Clients, wobei die nötigen Verbindungsdaten auch über den Server ausgetauscht werden. Bei beiden Systemen (Client und Server) existiert eine Auswertungslogik die entsprechend reagiert.

6 Produktfunktionen bzw. Projektumsetzung

Hier werden die Funktionen und die Zusammensetzung des Programmes im Detail beschrieben.

6.1 Datenbank

- Zunächst sollen alle User in einer Tabelle festgehalten werden.
  Diese sollen neben einer eindeutigen ID alle im unten dargestellten ER-Diagramm aufgeführten Attribute enthalten. Die Benutzer-ID und der Nickname sind eindeutig.
  Eine Besonderheit stellen die Profilbilder dar. Diese sollen auf dem Server abgelegt und nur die Pfade in der Datenbank gespeichert werden (siehe Abb. 2).

- Um eine Kontaktbeziehung darzustellen braucht man noch eine "kennt"- Tabelle. Diese soll lediglich zwei IDs der Benutzer enthalten. User kennt User. Dabei ist zu beachten, dass nur wenn die Benutzer IDs in beide Richtungen eingetragen sind, die Nutzer sich auch beide kennen. Wird nur eine einseitige Beziehung geführt handelt es sich um eine intiale Kontaktanfrage, wobei der Gegenüber noch nicht angenommen hat.

- Für die Gruppeneinteilung muss eine Tabelle "Gruppe" erstellt werden, die den Namen und eine ID als Primär-Schlüssel enthält.
  Es soll eine Relation "ist in Gruppe" existieren, bei der eine User-ID und eine Gruppen-ID gespeichert werden. Beide zusammen ergeben den Primärschlüssel (siehe Abb. 2)

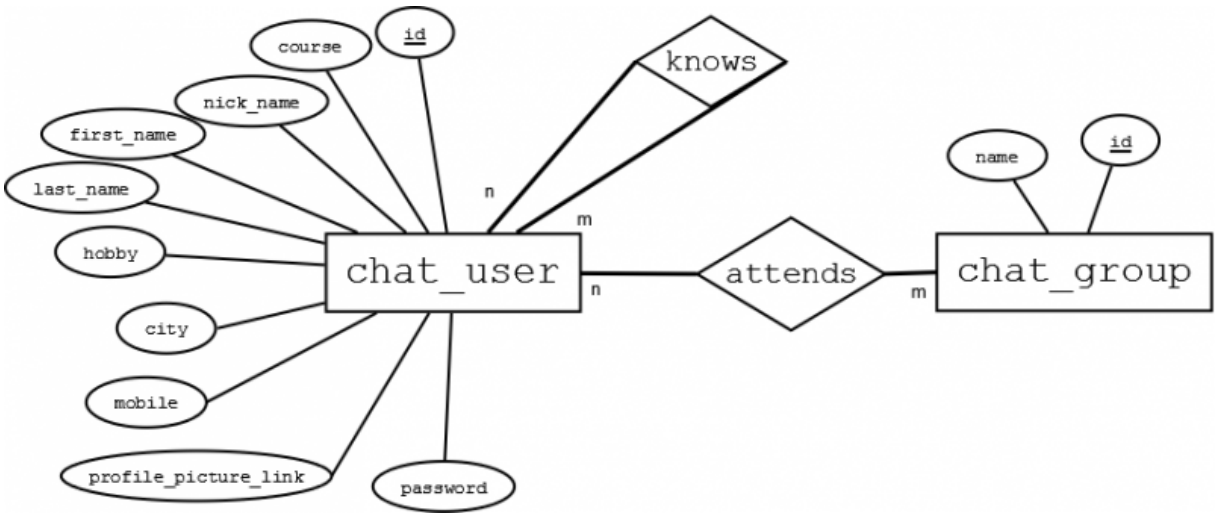

Abb. 2: Eine Beschreibung der Tabellen „chat_user“ und „chat_group“ für das Datenbankmodell mit Beziehungen und Attributen.

6.2 Login

Das Zustandsdiagramm auf Abb. 3 beschreibt den Vorgang des Login. Dabei sendet der Client eine Login-Anfrage an den Server, der diese entweder zurückweist oder anerkennt.

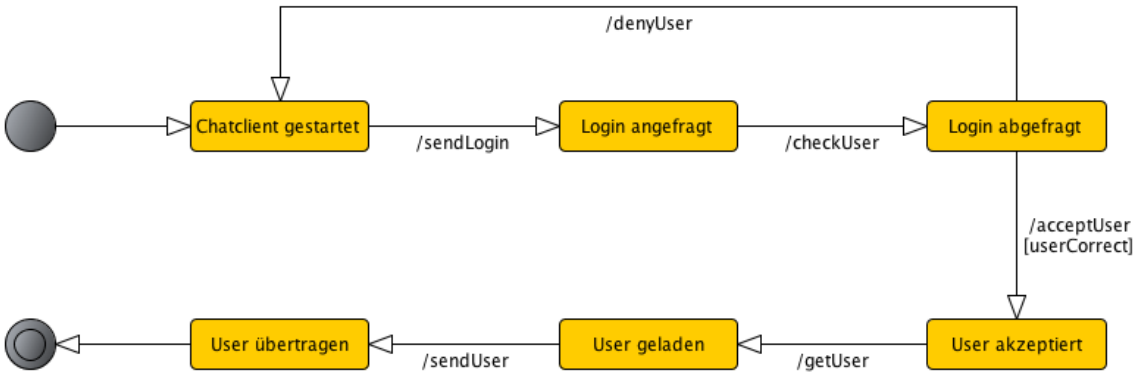

Abb. 3: Ein Zustandsdiagramm zur Beschreibung des Login-Vorgangs. Als Endzustand wurde der User übertragen, bzw. der User ist eingeloggt.

6.3 Gekapseltes Nachrichtenmodell

Um die Daten bequem vom Client zum Server senden zu können, soll ein gekapseltes Modell verwendet werden. So können einfach AMessage Objekte versendet werden, um den Rest kann sich jede Funktion selber kümmern. Ist kein Empfänger angegeben, so ist die Nachricht für den Server. Ansonsten leitet der Server diese weiter und führt gegebenenfalls ergänzende Arbeiten aus.
Das bedeutet, dass neben Textnachrichten auch Kontaktanfragen, Gruppenerstellungen, Gruppeneinladungen und Adressänderungen Nachrichten sind (Siehe Abb. 4).

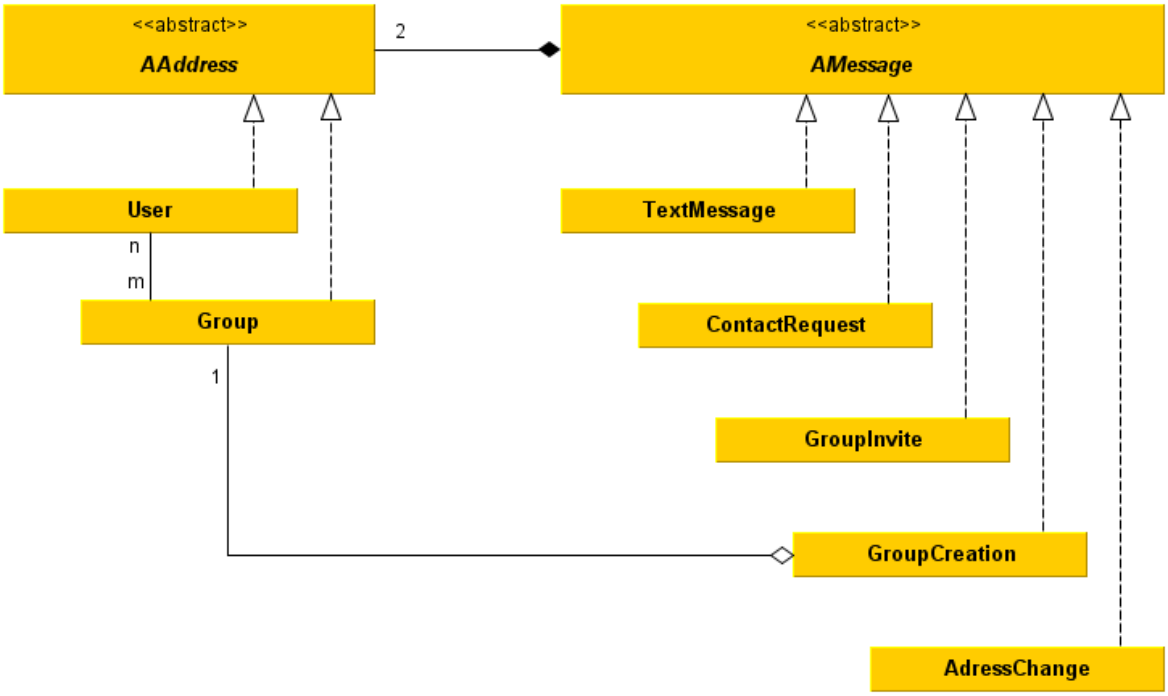

Abb. 4: Das gekapselte Nachrichtenmodell als Klassendiagramm. Jede Nachricht hat zwei Attribute vom Typ AAdress, diese können Benutzer oder Gruppen sein. Dabei dient ein Attribut zur Kennzeichnung des Senders, das andere für den Empfänger. Die AMessage-Typen können bei Bedarf um Weitere ergänzt werden.

6.4 Versenden einer Nachricht

Bei Versenden einer Nachricht, sind mindestens zwei Akteure, bzw. zwei User, von Nöten. Der Server übernimmt die Rolle des Vermittlers (siehe Abb. 5).

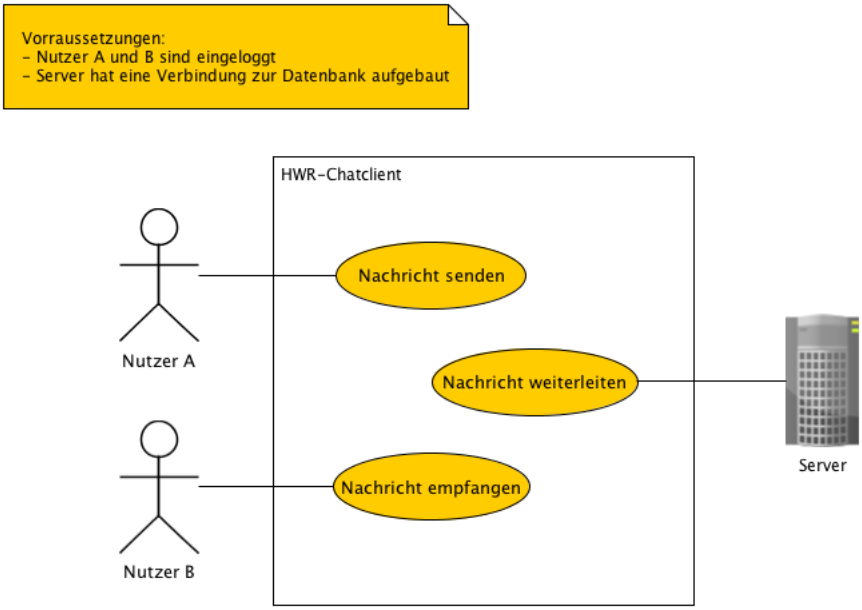

Abb. 5: Ein UseCase Diagramm, das das Senden und Empfangen einer Nachricht zeigt.

6.5 Nachrichtenweiterleitung vom Server

In der Abb. 6 wird beschrieben, wie der Server eine Nachricht (in diesem Spezialfall eine Textnachricht), die ein Client versendet hat, an den Empfänger weiterleiten soll.

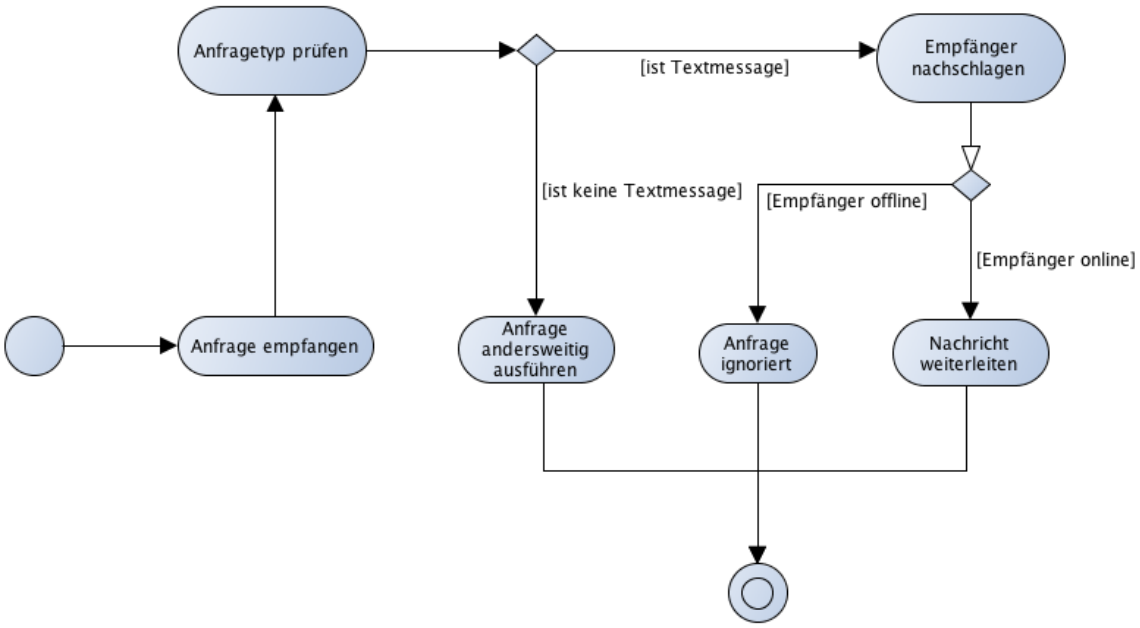

Abb. 6: Ein Flussdiagramm, dass das Ankommen und Weiterleitung einer Nachricht für den Server veranschaulicht.

6.6 Datenerhaltungskonzept: Server Data Access Object

Die Anmeldung der User soll am Server erfolgen. Dabei lädt der Nutzer sein Profil sowie seine ihm bekannten Kontakte und Gruppen. Diese Daten müssen daher für den Server vorgehalten werden, damit dieser sie bei Bedarf speichern, bearbeiten oder auch löschen kann. Um auch bei einem möglichen Absturz oder beim Herunterfahren konsistente Daten zu gewährleisten, soll hierzu eine Datenbank genutzt werden.

Für die konkrete Implementierung bedeutet dies, dass es zwei unterschiedliche Datenmodelle gibt. Zum Einen die Datenbank selbst (und deren Schnittstelle um Daten auszulesen / zu speichern) und andererseits ein internes Datenmodell des Servers. Beim initialen Start wird die Datenbasis des Servers mit der Datenbank synchronisiert, was im konkreten Fall bedeutet, dass alle Nutzer und Gruppen geladen werden. Werden nun lesende Anfragen gestellt, wird nur die ServerDatenStruktur benutzt, die Datenbank bleibt aus Perfomancegründen unangetastet. Erst wenn Daten verändert werden, muss ein Abgleich der Datenbank mit dem Server erfolgen, damit die Daten konsistent bleiben.

Die Datenerhaltung auf dem Server erfolgt durch einen Liste von Benutzerobjekten, die wiederum Referenzen auf andere Gruppen/User Objekte besitzen. Diese Objektorientiertheit hat den Vorteil, das bei Änderung eines Users, die User die auf diesen User referenziert worden sind, unmittelbar die aktuellen Daten bereithält. Der Server muss bei Änderungen nur dafür Sorge tragen, dass alle Freunde, die diesen Nutzer kennen, diese Änderung auch mitbekommen.
Die einzigen Daten, die nicht mit der Datenbank synchronisiert werden, sind die Statuseinträge, da diese lediglich live existieren.

6.7 Config-Manager

Bestimmte Einstellungen müssen in Konfigurationsdateien gespeichert werden, damit bei Änderung das Programm nicht neu kompiliert werden muss. Das betrifft unter anderem die Adresse des Servers, die Adresse der Datenbank oder das ausgewählte Design des Clients. Wenn eine solche Datei noch nicht existiert, soll diese mit vorgegebenen Standardeinstellungen automatisch erstellt werden. Es soll ein System für beide Konfigurationsdateien (Server, Client) entwickelt und standardisiert werden. Die Konfigurationsdateien sollen "server.conf" und "client.conf" heißen und mit einem Texteditor veränderbar sein.

6.8 Server-Kommandos

Direkt am Server soll die Eingabe verschiedener Kommandos möglich sein, um so beispielsweise einen neuen Benutzer zu registrieren.

Der Server soll eine Konsole besitzen. Folgende Kommandos soll der Server verarbeiten können:

Befehl	**Beschreibung**
register <Name> <Mail>	Registriert einen neuen Benutzer mit angegebenem Namen und Mail Adresse
listUsers	Listet alle User auf
listGroups	Listet alle Gruppen auf
resetPassword <ID>	Setzt ein neues Passwort
removeUser <ID>	Entfernt den Nutzer mit der angegebenen ID
removeGroup <ID>	Entfernt den Nutzer mit der angegebenen ID

Bei Bedarf sollen ergänzend auch verschiedene Informationen auf der Server-Konsole angezeigt werden können.

7 Produktdaten

In diesem Abschnitt werden die Daten aufgelistet, welche für den Betrieb des Programmes langfristig gespeichert werden müssen.

- Auf dem Server werden die persönlichen Nutzerdaten (Profilinformationen) des Users, ...
- ... seine Kontakte und Gruppen, ...
- ... und die Gruppen des Chats selbst gespeichert.
- Logdateien einer Konversation kann der User lokal auf seinem Client-System speichern.
- Nachrichten an Kontakte, die zum Zeitpunkt der Übertragung nicht online sind, werden auf dem Server zwischengespeichert, bis die Übertragung abgeschlossen werden kann.

8 Produktleistungen

Hier werden gewünschte Leistungskriterien in Bezug auf die Geschwindigkeit des Chats (Reaktionszeiten) festgelegt.

- Die Übertragung einer Textnachricht darf nicht länger als 10 Sekunden dauern.
- Für das Abrufen von Profilinformationen dürfen nicht mehr als 10 Sekunden benötigt werden.
- Der Onlinestatus der Kontakte wird in einer Periode von weniger als 30 Sekunden aktualisiert.

9 Qualitätsanforderungen

- Während einer Dateiübertragung soll das Senden von Nachrichten weiterhin möglich sein.
- Nachrichten dürfen nicht verloren gehen, auch wenn während der Übertragung die Verbindung zum Empfänger unterbrochen ist.

10 Benutzungsoberfläche (GUI)

Die Benutzeroberfläche des Clients soll die folgenden Punkte beinhalten:

- Die Oberfläche bietet eine Eingabemaske zur Anmeldung am Chat.
- Nach erfolgreicher Anmeldung steht dem User das Hauptfenster zur Verfügung, welches die Liste seiner bekannten Kontakte (inklusive Onlinestatus) und Gruppen beinhaltet.
- Sofern implementiert, kann der User von jedem seiner Kontakte und auch von sich selbst eine Seite mit den zugehörigen Profilinformationen (Profilseite) aufrufen.
- Zum Versenden von Nachrichten an einen Kontakt oder einer Gruppe, steht dem User ein entsprechendes Fenster zur Verfügung. Dieses beinhaltet ein Eingabefeld zum Schreiben der Nachricht, sowie ein Feld zur Anzeige der zuletzt ausgetauschten Nachrichten.
- In diesem Fenster findet sich auch der Button für die Dateiübertragung. Buttons und Felder für weitere Funktionalitäten können bei Bedarf ergänzt werden.

11 Nichtfunktionale Anforderungen

Anforderungen die nicht die Funktionalität der Software betreffen, werden hier aufgeführt.

- Die grafische Oberfläche soll sich am Designkonzept der HWR-Berlin orientieren.
- Die Nutzung oder Anpassung des Logos der HWR-Berlin für den Chat ist rechtlich zu prüfen.
- Gespeicherte persönliche und sensible Daten sind vor unbefugtem Zugriff aus datenschutzrechtlichen Gründen zu schützen.
- Für die Passwörter gelten besondere Sicherheitsanforderungen: Diese dürfen nur verschlüsselt gespeichert werden.
- Die Oberfläche soll ohne weitere Anleitung von jedem intuitiv bedienbar sein.
- Der Chat-Client soll für alle gängigen Betriebssysteme zur Verfügung gestellt werden.

12 Technische Produktumgebung

Hier werden die technischen Voraussetzungen, welche für den Betrieb der Software benötigt werden, aufgeführt.

12.1 Software

12.1.1 Server

- Der Chat-Server soll auf verschiedenen Serverumgebungen, gegebenenfalls nach Anpassung, betrieben werden können.
- Die Serverumgebung muss die zur Betreibung des Chats benötigten Sicherheitsanforderungen erfüllen (Firewall, sichere Konfiguration der Serverumgebung und bestenfalls Anti-Viren-Software).
- Der Server muss die nötigen Ports für die Clients freigeben, so dass diese auf den Chat-Server zugreifen können.
- Es wird eine MySQL5-Datenbank benötigt.
- Auf dem Server muss eine aktuelle Java-Version verfügbar sein.

12.1.2 Client

- Der Chat-Client soll auf allen gängigen Betriebssystemen (Windows, Mac OS, Linux) lauffähig sein.
- Auf dem Client-Rechner muss eine aktuelle Java-Umgebung vorhanden sein.
- Auf Seiten des Clients müssen die für den Chat benötigten Ports freigeschalten und für den Chat-Client zur Verfügung stehen.

12.2 Hardware

12.2.1 Server

- Die Serversoftware muss auf einem Netzwerkserver betrieben werden. Dieser soll hohe Zugriffszahlen verarbeiten können.
- Der Chat-Server soll ressourcensparend betrieben werden (Prozessorlast, Speicherbedarf).

12.2.2 Client

- Der Clientrechner muss netzwerkfähig sein, um sich mit dem Server verbinden zu können.

12.3 Orgware

12.3.1 Server

- Zur Verwaltung und Betreuung des Servers ist ein Administrator oder ein Administratorenteam erforderlich.

12.4 Produktschnittstellen

12.4.1 Server

- Derzeit sind keinerlei Schnittstellen vorgesehen. Bei Bedarf können jedoch Schnittstellen zu den Diensten der HWR-Berlin (z.B.: iPool, Moodle) nachgerüstet werden.

13 Spezielle Anforderungen an die Entwicklungsumgebung

Die hier genannten Aspekte werden für die qualitative Entwicklung der Software benötigt.

13.1 Software

- Die Software "Eclipse" zur Programmierung mit Java wird benötigt.
- Eine aktuelle Java-Developer-Version (jdk) wird zur Kompilierung von Server und Client benötigt.

13.1.1 Test-Server

- Auf dem Server muss eine aktuelle Java-Version verfügbar sein.
- Der Server muss die nötigen Ports für die Clients freigeben, so dass diese auf den Chat-Server zugreifen können.
- Es wird eine MySQL5-Datenbank benötigt.

13.1.2 Test-Client

- Um den Chat-Client zu testen, benötigt man um die gewünschte Plattformunabhängigkeit zu gewährleisten, mehrere Rechner mit jeweils einem der gängigen Betriebssysteme.
- Auf dem Client-Rechner muss eine aktuelle Java-Umgebung verfügbar sein.
- Auf Seiten des Clients müssen die für den Chat benötigten Ports freigegeben werden und für den Chat-Client zur Verfügung stehen.

13.2 Hardware

13.2.1 Test-Server

- Die Serversoftware muss auf einem netzwerkfähigen Rechner betrieben werden.
- Der Chat-Server soll ressourcensparend betrieben werden (Prozessorlast, Speicherbedarf).

13.2.2 Test-Client

- Der Clientrechner muss netzwerkfähig sein, um sich mit dem Server verbinden zu können.

14 Gliederung in Teilprodukte

Das Chat-Programm kann in folgende Teilprodukte unterteilt werden:

- Client
- Server (+ Datenbank)

14.1 Client

Der Client wird an den User ausgeliefert. Über diesen kann sich der User mit dem Server verbinden, relevante Daten abfragen und sich mit anderen Clients in Verbindung setzen.

14.2 Server

Der Server wird an die HWR-Berlin geliefert und von dieser in ihrem Rechenzentrum betrieben. Der Server stellt mit Hilfe der angebundenen Datenbank Informationen für die Clients zur Verfügung und steuert die Kommunikation zwischen ihnen.

15 Glossar

Abstraktion: Eine Abstraktion bezeichnet meist den induktiven Denkprozess des Weglassens von Einzelheiten und des Überführens auf etwas Allgemeineres oder Einfacheres.

Administrator: Ein (System-)Administrator verwaltet Computersysteme auf der Basis von umfassenden Zugriffsrechten auf das System. Er ist zuständig für die Planung, Installation, Konfiguration und Pflege der IT-Infrastruktur eines Unternehmens oder einer Organisation.

Attribut: Ein Attribut ist eine Eigenschaft oder ein Merkmal, Kennzeichen, Informationsdetail etc., das einem konkreten Objekt zugeordnet ist.

Button: Ein Button ist ein Bedienelement in einer grafischen Oberfläche, oft als Knopf oder Schaltfläche bezeichnet.

Chat: Als Chat bezeichnet man die elektronische Kommunikation in Echtzeit, die im Netzwerk oder über das Internet stattfindet.

Client: Ein Client ist ein Programm, das einen Server kontaktiert, um dessen Dienstleistung zu nutzen.

Eclipse: Eclipse ist ein quelltextoffenes Programmierwerkzeug zur Entwicklung von Software verschiedenster Art. Dabei wird Eclipse als integrierte Entwicklungsumgebung für die Programmiersprache Java genutzt.

Firewall: Eine Firewall ist eine Software, die dazu dient, den Netzwerkzugriff, basierend auf Absender- oder Zieladresse und genutzten Diensten, zu beschränken.

GUI: Das GUI ist die Grafische Benutzeroberfläche (Graphical User Interface) des Programmes. Es bildet die Schnittstelle zum Menschen, also zwischen Client und User, und zwischen Server und Administratoren

Java: Java ist eine objektorientierte Programmiersprache, welche weitestgehend plattformunabhängig ist. Das heißt sie ist ohne weitere Anpassungen auf verschiedenen Rechnerarchitekturen und Betriebssystemen lauffähig.

JDK: Das Java Development Kit (JDK) ist eines der von Java-Entwicklern meistgenutzten Java-SDKs (Software Development Kits).

Kapselung: Als (Daten-)Kapselung bezeichnet man in der Programmierung das Verbergen von Daten oder Informationen vor dem Zugriff von außen. Der direkte Zugriff auf die interne Datenstruktur wird unterbunden und erfolgt stattdessen über definierte Schnittstellen.

Klasse: Unter einer Klasse versteht man in der objektorientierten Programmierung ein abstraktes Modell bzw. einen Bauplan für eine Reihe von Objekten.

Kommilitone: Kommilitone ist die übliche Bezeichnung von Studenten für ihre Studienkollegen.

Kompilierung: Unter Kompilierung versteht man die Übersetzung des Quelltextes eines Computerprogramms in ein semantisch äquivalentes Programm in der Zielsprache.

Konfigurationsdatei: Eine Konfigurationsdatei ist eine Datei, in der bestimmte Einstellungen (die Konfiguration) von Computerprogrammen oder Hardwarebestandteilen gespeichert sind.

Konsistenz: In der Informatik sind Daten konsistent, wenn Widerspruchsfreiheit innerhalb einer Datenbank gewährleistet ist.

Konsole: Eine Konsole ist ein Benutzerendgerät zur Anzeige und Eingabe von Daten.

Logdatei: Als Logdatei wird ein automatisch geführtes Protokoll bestimmter Aktionen bezeichnet. Im Falle des Chat Programmes ist das Protokoll der ausgetauschten Nachrichten gemeint.

MySQL: MySQL ist ein relationales Datenbankverwaltungssystem, das als Open-Source-Software sowie als kommerzielle Enterpriseversion für verschiedene Betriebssysteme verfügbar ist.

Objekt: Ein Objekt bezeichnet in der objektorientierten Programmierung (OOP) ein Exemplar eines bestimmten Datentyps oder einer Klasse.

Objektreferenz: Eine Objektreferenz ist eine Variable, die mit einer Klasse als Typangabe deklariert wird. Eine solche Objektreferenz kann einen Verweis auf die Instanz einer passenden Klasse aufnehmen.

Parameter: Parameter sind Variablen, über die ein Computerprogramm oder Unterprogramm, für einen Aufruf gültig, auf bestimmte Werte "eingestellt" werden kann.

Peer-to-Peer: Unter Peer-to-Peer versteht man eine direkte Rechner zu Rechner Verbindung. Beide Rechner sind gleichberechtigt, ein Server wird nicht benötigt.

Port: Der Port ist der Teil einer Netzwerkadresse, der bestimmt welchen Daten welches Netzwerkprotokoll zugewiesen wird.

Primärschlüssel: Ein Primärschlüssel dient in einer Datenbank zur eindeutigen Identifizierung von individuellen Daten.

Profilseite: Als Profilseite bezeichnet man eine Seite, auf der einem angemeldeten Nutzer persönliche Informationen zur Verfügung gestellt werden.

Relation: Eine Relation ist eine bestimmte Beziehung zwischen mindestens zwei unterschiedlichen Daten.

Schnittstelle: Eine Schnittstelle (auch Interface) dient zur Kommunikation zwischen Modulen eines Systems oder zwischen dem Menschen und Modulen eines Systems. Schnittstellen müssen immer exakt definiert sein.

SDK: Ein Software Development Kit (SDK) ist eine Sammlung von Werkzeugen und Anwendungen, um eine Software zu erstellen.

Server: Ein Server ist ein Programm, das mit einem oder mehreren Clients kommuniziert, und ihnen damit Zugang zu einem Dienst verschafft.

UML: Die Unified Modeling Language (UML) ist eine graphische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software.