|  |  |
| --- | --- |
| HWR-Chat-Programm  Pflichtenheft | |
| **Autoren:** | Christian Mrosk, Alexander Salzer, Tobias Schlufter, Philipp Thomas, Felix Wiederschein |
| **Modul:** | IT2101 – Software Engineering I |
| **Fachbereich:** | Technik Fachbereich 2, Duales Studium Wirtschaft • Technik, HWR Berlin |
| **Studiengang:** | Angewandte Informatik |
| **Studienjahrgang:** | 2010 |
| **Datum:** |  |

**Vorwort**

**Im Modul** T2101 – Software Engineering I im 3. Semester bekam der Informatik-Kurs die Aufgabe von Frau Prof. Dr. Dagmar Monett Díaz gestellt, ein Chatprogramm für die HWR-Berlin zu erstellen. Dazu wurde der Kurs in 5 Gruppen á 5 Studenten aufgeteilt. Jede Gruppe sollte ein eigenes Chatprogramm programmieren. Zusätzlich sollten ein Pflichtenheft, eine Homepage und eine Präsentation zum Projekt erstellt werden.

Die Gruppe 2 besteht aus Christian Mrosk, Alexander Salzer, Tobias Schlufter, Philipp Thomas & Felix Wiederschein. Im diesem Pflichtenheft werden die **Systemarchitektur, die genauen Spezifikationen der Systemanforderungen**, die Systemmodelle sowie die Systementwicklung des HWR Instant Messenger (HWRIM) erläutert.

**Inhaltsverzeichnis**

[Vorwort I](#_Toc307319966)

[Inhaltsverzeichnis II](#_Toc307319967)

[Abkürzungsverzeichnis III](#_Toc307319968)

[Glossar III](#_Toc307319969)

[Abbildungsverzeichnis III](#_Toc307319970)

[Tabellenverzeichnis III](#_Toc307319971)

[Einleitung 1](#_Toc307319972)

[Definition der Benutzeranforderungen 1](#_Toc307319973)

[Systemarchitektur 2](#_Toc307319974)

[Spezifikation der Systemanforderungen 3](#_Toc307319975)

[Systemmodelle 5](#_Toc307319976)

[Systementwicklung 7](#_Toc307319977)

[Anhänge 7](#_Toc307319978)

[Index 7](#_Toc307319979)

**Abkürzungsverzeichnis**

**HWRIM - HWR Instant Messenger**

Server  **-** Programm, das zur Verwaltung von Anfragen dient.

Client  **-** Der Teil des Programms, den der Benutzer sieht.

RMI **- R**emote **M**ethod **I**nvocation

Beschreibt einen Methodenaufruf zu einem nicht lokalen Programm. Der Aufruf erfolgt aber durch das **RMI** wie ein lokaler Aufruf.

**Glossar**

**Abbildungsverzeichnis**

[Abbildung 1: Aktivitätsdiagramm Registrierung 5](#_Toc307320418)

[Abbildung 2: Use-Case-Diagramm Login 6](#_Toc307320419)

[Abbildung 3: Sequenzdiagramm Login 7](#_Toc307320420)

[Abbildung 4: Zustandsdiagramm Nachricht versenden 8](#_Toc307320421)

**Tabellenverzeichnis**

**Einleitung**

**Der HWR-Chat soll die Kommunikation zwischen den Studenten erleichtern. Besonderer Vorteil ist, dass die Studenten nicht wie in anderen Netzwerken ihre Kommilitonen einzeln suchen müssen. Dieses Programm bietet einen gesammelten Überblick über alle wichtigen Kontakte des Studiums an. So kann man sich mit den Studenten des gleichen Studiengangs über die Hausaufgaben oder Erfahrungen austauschen und mit Studenten anderer Fachrichtungen fächerübergreifende Diskussionen führen.**

**Jeder Student kann nach der Registrierung und Anmeldung am Server alle gewünschten Kommilitonen zu seinen Kontakten hinzufügen. Diese werden in einer Kontaktliste übersichtlich dargestellt. Mit einem Doppelklick auf einen Kontakt öffnet sich das Chatfenster in dem Nachrichten verschickt werden können. Der Nachrichtentransfer zu jedem Kontakt wird in einem Verlauf gespeichert und ist jederzeit einsehbar. Es ist möglich Dateien zu versenden. Auf der Profilseite kann jeder Student die Angaben zu seiner Person einsehen.**

**Mit diesen Funktionen erleichtert die HWR ihren Studenten die Kommunikation und Absprachen untereinander. Durch den Dateitransfer können auch gemeinsam bearbeitete Dokumente verschickt werden. Somit erleichtert der HWR-Chat dem Studenten das Hochschulleben.**

**Definition der Benutzeranforderungen**

**Muss-Kriterien:**

1. **Nachrichtentransfer zwischen 2 Personen**
2. **Kontaktliste als Hauptfenster mit folgenden Eigenschaften:**
   1. **Liste die alle hinzugefügten Kontakte anzeigt.**
   2. **Ein Doppelklick auf einen Kontakt soll den Nachrichtentransfer mit ihm ermöglichen.**
   3. **Menüleiste mit Zugriff auf die Optionen des Programms**
3. **Der Verlauf soll abgespeichert werden.**
4. **Parallel zum Nachrichtentransfer sollen auch Dateien versendet werden.**
5. **Profilseite, in der alle eigenen Angaben eingesehen werden können.**
6. **Eine Anzeige im Chatfenster, ob das Gegenüber gerade schreibt.**
7. **Benutzer müssen sich vor erster Benutzung registrieren**

**Soll-Kriterien:**

1. **Gruppierung der Kontakte in der Kontaktliste nach Studienrichtung**
2. **Anzeige, ob ein Kontakt online oder offline ist**
3. **Strukturiertes Chatfenster:**
   1. **Verschiedene Farben von Sender und Empfänger**
   2. **Zeitstempel vor den Nachrichten**
   3. **Absenden des Textes mit der Entertaste**

**Kann-Kriterien:**

1. **Benachrichtigung, dass ein Kontakt online gekommen ist**

**Systemarchitektur**

Das System soll sich in die Hauptbestandteile Server und Client aufteilen. Der Server besitzt zwei Hauptfunktionen. Zum einen das Bearbeiten von Anfragen und zum anderen das Verwalten von Benutzerdaten. Der Client ist die Schnittstelle zwischen Benutzer und System.

Die Verwaltung der Daten auf dem Server geschieht in einer XML-Datei. Darin ist für jeden registrierten Benutzer ein Datensatz angelegt, der von ihm beliebig bearbeitet werden kann.

Für jeden angemeldeten Benutzer erstellt der Server einen neuen Thread, damit mehrere Anfragen parallel bearbeitet werden können. Damit die Daten an die korrekte Zielperson übertragen werden, führt der Server eine Liste mit allen angemeldeten Nutzern. Aus dieser Liste wird der gewünschte Kontakt herausgesucht und die Nachrichten werden übertragen. Die Datenübertragung geschieht per Remote Method Invocation (RMI).

Der Benutzer arbeitet nur mit dem Client. In einem Menü können Einstellungen zu den persönlichen Angaben vorgenommen werden. Eine Kontaktliste zeigt alle befreundeten Kontakte sortiert nach Studienrichtung an. Mit einem Doppelklick öffnet sich ein Nachrichtenfenster mit dem ausgewählten Kontakt. Im unteren Teil des Fensters kann die Nachricht eingegeben und per Tastendruck versendet werden. Der angeschriebene Kontakt wird mit einer Einblendung aufmerksam gemacht, dass er angeschrieben wurde. Jede gesendete Nachricht wird im Verlauf gespeichert. So lassen sich jederzeit frühere Kommunikationen noch einmal nachvollziehen.

Jeder Nutzer kann neben einfachen Textnachrichten auch Dateien versenden. Ein entsprechendes Feld im Nachrichtenfenster öffnet ein Menü, in dem die Datei ausgewählt wird. Der Empfänger der Datei kann diese Datei annehmen und auf seinem Rechner abspeichern.

**Spezifikation der Systemanforderungen**

**Muss-Kriterien:**

1. **Das Nachrichtenfenster soll einen Dialogbereich für den Empfang und einen Schreibbereich für das Versenden von Nachrichten besitzen. Darunter soll sich eine Schaltfläche befinden, mit der sich die Nachrichten versenden lassen. Die Enter-Taste besitzt die gleiche Funktionalität.**

**Nach dem Absenden der Nachricht soll sie im Dialogbereich, mit einem Zeitstempel versehen, angezeigt werden.**

1. **Nach dem Start des Programms öffnet sich die Kontaktliste. Sie ist das zentrale Fenster des Programms. Die Kontaktliste selbst ist nach Studienrichtungen und anderen wichtigen Bereichen (z.B. externe Dozenten) sortiert.**

**Um das Nachrichtenfenster mit einem anderen Kontakt zu öffnen, muss doppelt auf den Kontakt geklickt werden. Es öffnet sich das oben beschriebene Nachrichtenfenster. In der Menüleiste *Datei* sind folgende Einträge vorhanden:**

* + **Login – Meldet den Benutzer an.**
  + **Logout – Meldet den Benutzer ab.**
  + **Registrierung – siehe Punkt 5**
  + **Profilseite – Zeigt die Angaben des Benutzers an.**
  + **Kontakt hinzufügen – Öffnet ein Menü, in dem nach anderen Benutzern gesucht werden kann.**
  + **Beenden – Beendet das Programm.**

1. **Die geschriebenen Nachrichten werden während der Übertragung in einem Verlauf gespeichert. Im Nachrichtenfenster besteht die Option „*Verlauf anzeigen*“. Ist diese aktiviert, so steht der gesamte Verlauf im Dialogbereich zur Verfügung.**
2. **Das Nachrichtenfenster besitzt die Schaltfläche „*Datei versenden*“. Klickt der Benutzer darauf, so öffnet sich ein Dateidialog, in dem der Benutzer die gewünschte Datei aussuchen kann. Die Datei wird anschließend versendet. Der Empfänger der Datei sieht in seinem Nachrichtenfenster eine Benachrichtigung und kann die Datei empfangen. Für ihn öffnet sich ebenfalls ein Dateidialog, in der er den Speicherort der Datei wählt. Danach wird die Datei übertragen.**
3. **Auf der Profilseite kann der Benutzer seine bei der Registrierung hinterlegten Daten einsehen. Folgende Daten können angesehen und geändert werden:**
   * **Vorname**
   * **Nachname**
   * **Nickname**
   * **Matrikelnummer**
   * **Passwort**
   * **Studiengang**
   * **Jahrgang**
   * **E-Mail Adresse**
4. **Zwischen Dialog- und Eingabebereich zeigt eine Einblendung an, ob der Kommunikationspartner gerade schreibt.**
5. **Bevor das Programm genutzt werden kann, muss sich der Benutzer registrieren. Dazu wählt er im Menü des Kontaktlistenfensters den Eintrag „*Registrierung*“. Es öffnet sich das Registrierungsfenster. Dort muss der Nutzer alle Angaben machen, die oben bei der Profilseite genannt wurden. Sind alle Eingaben korrekt und ist der Nutzername nicht bereits in der Nutzerdatenbank vorhanden, wird für den Benutzer ein neuer Eintrag angelegt. Die Registrierung ist nun erfolgreich abgeschlossen und der Benutzer kann nun mit seinen Kontakten kommunizieren.**

**Soll-Kriterien:**

1. **Die Kontakte des Benutzers werden automatisch nach Studienrichtung und anderen wichtigen Bereichen (z.B. externe Dozenten) sortiert.**
2. **Ist ein befreundeter Kontakt online, wird er in der seinem Studiengang zugehörigen Gruppe angezeigt. Ist er nicht online, wird er in der Kontaktliste auch nicht angezeigt.**
3. **Im Dialogbereich werden die geschriebenen Zeilen von Sender und Empfänger in verschiedenen Farben angezeigt. Zusätzlich besitzt jede Nachricht auch einen Zeitstempel. Das Absenden einer Nachricht geschieht mit der Eingabetaste.**

**Kann-Kriterien:**

1. **Kommt ein befreundeter Kontakt online, wird dies durch ein PopUp-Fenster signalisiert.**

**Systemmodelle**

**Aktivitätsdiagramm: Registrierung**

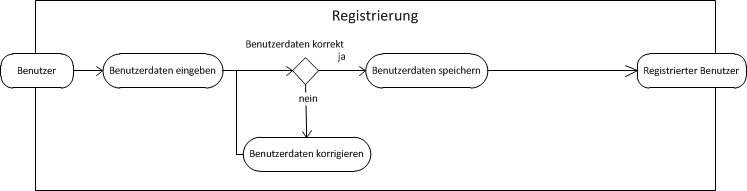
****

Abbildung 1: Aktivitätsdiagramm Registrierung

**Die Abbildung 1 ist ein Aktivitätsdiagramm, welches die Registrierung beim HWRIM zeigt. Der Benutzer startet das Programm und klickt in der Menüleiste *Datei* die Funktion *Registrierung* an. Im sich öffnenden Fenster gibt der Benutzer nun seine Benutzerdaten ein. Wenn alle Benutzerdaten korrekt eingegeben sind, werden die neuen Benutzerdaten auf dem Server gespeichert und der Benutzer ist nun registriert. Falls die Benutzerdaten Fehler aufweisen, wie zum Beispiel die Wahl eines bereits genutzten Nicknames, wird eine Fehlermeldung ausgegeben. Daraufhin wird der Benutzer aufgefordert seine Benutzerdaten zu korrigieren.**

**Use Case: Login**

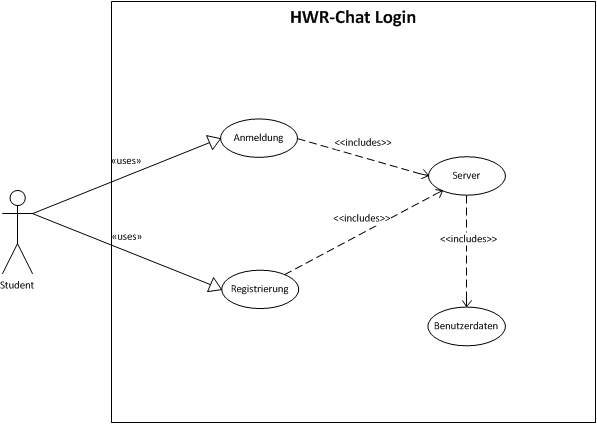
****

Abbildung 2: Use-Case-Diagramm Login

**Das Use-Case-Diagramm aus Abbildung 2 stellt die Login Funktion des HWRIM da. Die Studenten müssen sich vor Nutzung des Chatprogramms anmelden beziehungsweise vor der erstmaligen Benutzung registrieren. Bei der *Anmeldung* werden die eingegebenen *Benutzerdaten* zum *Server* geschickt und mit den vorhandenen *Benutzerdaten* in der Liste aus Usern überprüft und verglichen. Falls die eingegebenen *Benutzerdaten* nicht korrekt sind, wird eine Fehlermeldung an den Anwender gesendet. Bei der *Registrierung* werden die neuen *Benutzerdaten* auch erst vom *Server* geprüft. Da zum Beispiel jeder Nickname nur einmal vorhanden sein darf. Wenn alle *Benutzerdaten* korrekt sind, wird die User-Liste um einen neuen User, mit den neuen *Benutzerdaten*, erweitert.**

**Durch die** *include-Assoziation* **wird beim Starten der *Anmeldung* oder *Registrierung* auch der Server gestartet. Dieser wiederrum lädt die *Benutzerdaten*.**

**Sequenzdiagramm: Login**

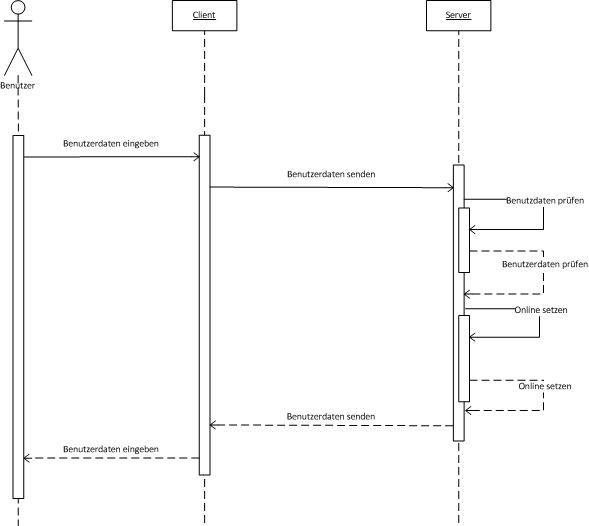


Abbildung 3: Sequenzdiagramm Login

**Die Abbildung 3 zeigt das Login nochmals als Sequenzdiagramm. Das Sequenzdiagramm beinhaltet als Akteur den *Benutzer* und die zwei Klassen *Client* und *Server*. Am Anfang gibt der Benutzer seine Benutzerdaten ein. Diese werden vom *Client* zum *Server* gesendet. Der *Server* überprüft nun die Benutzerdaten. Falls diese korrekt sind, wird der Benutzer vom *Server* online gesetzt. Nun besteht für den Benutzer die Möglichkeit mit seinen Freunden aus seiner Kontaktliste zu kommunizieren.**

**Zustandsdiagramm: Nachricht versenden**

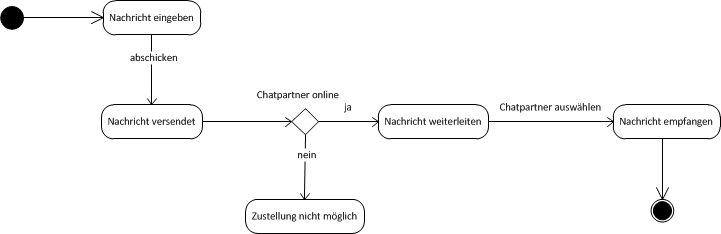


Abbildung 4: Zustandsdiagramm Nachricht versenden

**Das Zustandsdiagramm des HWRIM zeigt wie eine Nachricht im Programm vom Absender zum Empfänger gesendet wird. Dabei ist die Nachricht das Objekt des Zustandsdiagramms. Der Absender gibt im Schreibbereich eine Nachricht ein. Diese wird mit einem Klick auf die Enter-Taste oder die *Senden* Schaltfläche abgeschickt. Dabei schaut der Server in seiner Liste nach, ob der gewünschte Empfänger auch online ist. Ist dies der Fall wird die Nachricht an den Chatpartner weitergeleitet und die Nachricht konnte erfolgreich empfangen werden. Falls der gewünschte Empfänger nicht online ist, ist eine Zustellung der Nachricht nicht möglich und die Nachricht wird gelöscht.**

**Systementwicklung**

**Das Programm benötigt das .NET-Framework Version 4. Die weiteren Systemvoraussetzungen sind moderat und werden von jedem gängigen Bürorechner erfüllt.**

**Das Programm ist als Hintergrundanwendung ausgerichtet und dementsprechend hardwareschonend auf einen einzigen Prozessor ausgelegt. Steigende Hardwareanforderungen sind bei weiteren Versionen nicht zu erwarten.**

**Anhänge**

**Index**