TITELSEITE

Dokumentation zur schulischen Projektarbeit

**SagMa**

Anwendung zum Austauschen von Nachrichten

Abgabedatum : Frankfurt, den 18.05.2016

**Projektmitarbeiter:**

Tobias Stensbeck

Steven Pohl

Kevin Kahlsdorf

FIRMENLOGO

Inhaltsverzeichnis :

Mach ich wenn anderes soweit fertig

1. Einleitung
   1. Projektumfeld

* Das Projekt ist ein Teil der schulischen Ausbildung für Fachinformatiker Anwendungsentwicklung und ist daher für alle Azubis obligatorisch und wird im Auftrag des verantwortlichen Lehrers ausgeführt.
  1. Projektziel
* Bestand des Projektes ist es, den Austausch von Nachrichten zwischen verschiedenen Benutzern zu ermöglichen.
* Ein Benutzer soll sich anmelden können, um daraufhin mit anderen aktiven Benutzern Nachrichten austauschen zu können. Des Weiteren kann ein Benutzer andere Benutzer in eine Gruppe einladen und alle eingeladenen Benutzer können dann über einen Gruppenchat gleichzeitig miteinander kommunizieren.
  1. Projektbegründung
* Dieses Projekt beruht nicht auf einem wirtschaftlichen Interesse für keinen der Beteiligten.
* Der Sinn dieses Projekts ist die Vorbereitung auf die spätere Abschlussprüfung, in welcher ebenfalls alle Bestandteile, wie in diesem Projekt, gefordert werden.
* Die Projektmitarbeiter (hier: die Schüler) sollen so ein Gefühl für eine Projektarbeit bekommen und zur Kenntnis nehmen, welche Bereiche zu einer entsprechenden Aufgabe gehören.
  1. Projektschnittstellen
* Das Programm zum Austauschen von Nachrichten ist im derzeitigen Umfang nicht an andere Systeme gebunden. Es existieren also keine technischen Schnittstellen.
* Das Projekt wurde, nach Vorlage des Pflichtenhefts(siehe Anhang), durch Herrn Eick genehmigt und wird vor der Klasse FI4D präsentiert.

1. Projektplanung
   1. Projektphasen

* Die Umsetzung des Projekts erfolgte im Zeitraum vom 25.11.2015 - 18.05.2016.

Allerdings lag der Fokus der Arbeit am Projekt auf der Zeit in der Berufsschule und daher im Zeitraum vom 01.02.2016 – 26.02.2016 sowie vom 09.05.2016 – 11.05.2016.

* 1. Abweichungen vom Pflichtenheft
* Änderung der Produktumgebung:

Ein Zugriff auf eine MySQL bzw. Microsoft SQL Server Datenbank ist nicht notwendig, da keine Datenbank zum Speichern der Passwörter verwendet wird. Zum Projektstart war dies noch nicht ins Detail geklärt und es wurden zwei Möglichkeiten zum Abspeichern der Passwörter in Betracht gezogen. Die erste Möglichkeit war das Abspeichern in der Datenbank, die zweite das Abspeichern in einer Datei.

Gemeinsam wurde entschieden keine Datenbank zu verwenden, da der Aufwand nicht im Verhältnis zum Nutzen steht. Es gibt nur eine geringe Zahl an Benutzern und daher lohnt sich der Einsatz einer Datenbank nicht. Als Folge wurde die Umsetzung mit dem Abspeichern in einer Datei gewählt.

Eine Implementierung mit Hilfe einer Datenbank wäre trotzdem noch möglich, dazu müsste lediglich eine Klasse für den Datenbankzugriff entwickelt werden, welche das Interface AuthStore implementiert. Es handelte sich also nicht um eine kritische Entscheidung sondern es wurde die Dateivariante gewählt, da sie für den derzeitigen Verwendungszweck am Effizientesten war.

Wenn benötigt, wäre die Umstellung auf eine Datenbank zum Abspeichern möglich.

* 1. Ressourcenplanung
* Für die Umsetzung des Projektes wurden folgende Ressourcen verwendet :

1. Arbeitsplatz incl. PC, Tastatur, Maus und Internetzugang

2. Eclipse als Integrated Development Environment (IDE)

3. Java Runtime Enviroment 8 oder neuer

4. EclEmma als Code Coverage Tool

5. Git & GitHub für die Versionsverwaltung

* 1. Entwicklungsprozess
* Aufteilung der verschiedenen Aufgaben innerhalb des Projektteams, sowie ständige interne Kommunikation in Form von Gesprächen und der Nutzung von WhatsApp (Anlegen einer entsprechenden Gruppe).
* Aufteilung der Aufgaben wie folgt :

Erstellung der serverseitigen Anwendungen : Tobias Stensbeck.

Erstellung der Benutzeroberfläche(mit allen Funktionalitäten) : Steven Pohl.

Erstellung der Dokumentation, sowie des Authentifizierungsprozesses: Kevin Kahlsdorf.

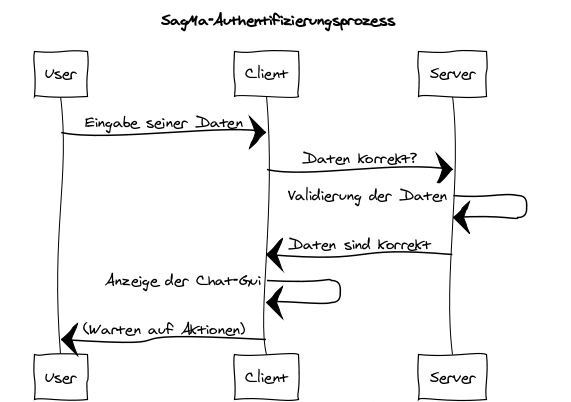
1. Entwurfsphase
   1. Zielplattform

* Die Auswahl der Zielplattform fand im Einklang unter den Projektmitarbeitern statt. Es wurde diskutiert, was benötigt wird bzw. ob etwas umsetzbar sei. Als Ergebnis wurde sich auf die Verwendung der Programmiersprache JAVA festgelegt, da diese alle geforderten Bedingungen erfüllt (Server, Client, GUI, etc.) und das Wissen der Projektmitarbeiter hier bereits sehr fortgeschritten ist.
  1. Entwurf der Benutzeroberfläche
* Das Wichtigste bei der Erstellung der GUI war die Übersichtlichkeit, wobei jedoch gleichzeitig ein angenehmes Umfeld geschaffen werden sollte. Die Funktionalitäten sollten leicht erkennbar und somit leicht bedienbar sein.

Die Bedienung soll ersichtlich und ohne Hinzunahme einer Bedienungsanleitung gewährleistet werden.

Um dies zu gewährleisten wurden gemeinsam einige Grundstrukturen festgelegt:

* + Eine Menüleiste am oberen Rand.
  + Ein Fenster zum Anzeigen der gesendeten und empfangenen Nachrichten am linken Rand gehalten, aber als größtes Feld.
  + Ein Fenster zur Eingabe von Nachrichten, sowie ein Button zum Abschicken unterhalb des Fensters mit den empfangenen Nachrichten.
  + Eine Übersicht der Chats an der rechten Seite
  + Eine Übersicht der angemeldeten Benutzer und Gruppen an der rechten Seite unterhalb des Fensters für die Übersicht der Chats.
  1. Geschäftslogik
* Bei Start der Anwendung wird der Benutzer aufgefordert einen Server, sowie Benutzername und Passwort einzugeben. Werden gültige Daten eingegeben, wird dem Benutzer die GUI (siehe 3.2) angezeigt und ihm stehen die Funktionalitäten zur Verfügung.
* Sequenzdiagramm, am Beispiel der Anmeldung (vereinfacht, keine Methodennamen zur besseren Verständlichkeit):



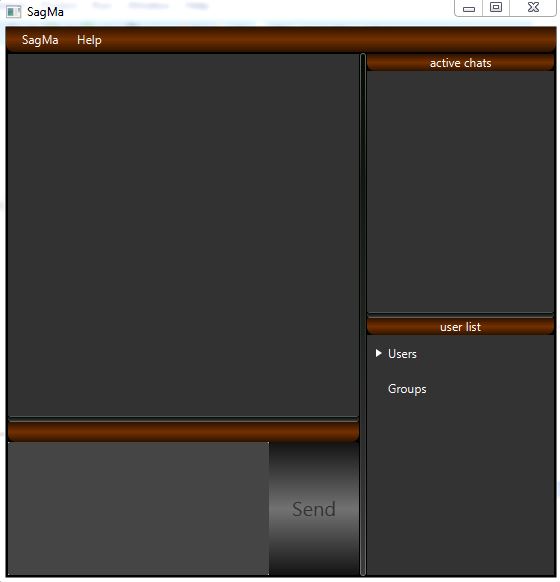
* Dies ist natürlich nur ein kleiner Ausschnitt, aber diese grundsätzliche Funktionsweise lässt sich auf weitere Funktionen übertragen. Der Benutzer gibt etwas ein oder wählt eine Funktion, der Client kommuniziert mit dem Server und führt die Aktion entsprechend aus.
  1. Maßnahmen zur Qualitätssicherung
* Um die Qualität des Produkts zu sichern wurden mehrere Testarten verwendet:

1. Erstellen von automatisierten Test mit dem Testframework JUnit.
2. Anwendertests mit den folgenden Testfällen :
3. Registrierung eines Benutzers
4. Anzeige anderer aktiver Benutzer
5. Versenden einer Nachricht an einen anderen Benutzer
6. Empfangen von Nachrichten
7. Erstellung von Gruppen
8. Hinzufügen von Gruppenmitgliedern
9. Schließen des Fensters -> Logout
10. Speichern der Registrierungsdaten in einer Datei
11. Authentifikation eines Benutzers (anmelden, nicht registrieren)
12. Implementierungsphase
    1. Implementierung der Benutzeroberfläche

* Bei der Implementierung der Benutzeroberfläche wurde eng am Entwurf festgehalten und sich auf die Funktionalität und Übersichtlichkeit fokussiert.

Außerdem kann der Benutzer die Breite der einzelnen Fenster seinen Wünschen anpassen.

Screenshot der Benutzeroberfläche nach erfolgtem Einloggen/Registrieren:



Menüleiste

Übersicht der aktiven Chats

Übersicht der angemeldeten User

Übersicht der existierenden Gruppen.

Feld zum Versenden von Nachrichten

* 1. Implementierung der Geschäftslogik

•

Beschreibung des Vorgehens bei der Umsetzung/Programmierung der entworfenen Anwendung.

•

Ggfs. interessante Funktionen/Algorithmen im Detail vorstellen, verwendete Entwurfsmuster zeigen.

•Quelltextbeispiele zeigen.

•Hinweis: Es wird nicht ein lauffähiges Programm bewertet, sondern die Projektdurchführung. Dennoch würde ich immer

Quelltextausschnitte zeigen, da sonst Zweifel an

der tatsächlichen Leistung des Prüflings aufkommen können.

1. Fazit

Als Abschluss kann man sagen, dass das Projekt vom Umfang und Komplexität her für den Zeitraum angemessen war. Alle Musskriterien aus dem Pflichtenheft wurden erfüllt. Bli bla blub.

Anhang

A1 : Pflichtenheft

SagMa - Pflichtenheft

1. Zielbestimmung

1.1 Musskriterien

Die Software soll den Austausch von Textnachrichten zwischen Individuen oder in Gruppen ermöglichen. Die Identifikation der Benutzer erfolgt über selbstgewählte Benutzernamen. Vor der Benutzung der Hauptfunktionalität muss sich ein Benutzer mit seinem Benutzernamen und Passwort identifizieren.

1.2 Wunschkriterien

Die Texteingabe und -darstellung kann so erweitert werden, dass einfache Formatierungen wie z. B. fetter oder kursiver Text und andere Textfarben möglich sind.

Das Versenden von Bildern und Dateien kann ermöglicht werden.

1.3 Abgrenzungskriterien

Der Chatverlauf soll nicht gespeichert werden.

Es soll nur ein Server verwendet werden. D. h. alle Benutzer, die miteinander kommunizieren wollen, müssen mit dem gleichen Server verbunden sein.

Eine Peer-to-Peer Kommunikation der Clients ist ebenfalls nicht vorgesehen.

2. Produktumgebung

Die Software wird in der Programmiersprache Java geschrieben.

Der Client benötigt ein Java Runtime Environment (JRE) in der Version 8 oder neuer.

Der Server benötigt ebenfalls eine JRE in der Version 8 oder neuer sowie Zugriff auf eine MySQL bzw. Microsoft SQL Server Datenbank.

3. Produktfunktionen

Man kann die IP-Adresse eines Servers angeben und sich mit Benutzername und Passwort anmelden.

Anschließend sieht man eine Liste mit vorhanden Gruppen und angemeldeten Benutzern. Wird eine Gruppe oder ein Benutzer ausgewählt, öffnet sich ein Chatfenster. Chatfenster von Gruppen zeigen alle Teilnehmer der Gruppe an.

4. Entwicklungsumgebung

Zur Entwicklung der Software wird Eclipse als Integrated Development Environment (IDE), JUnit als Testframework, EclEmma als Code Coverage Tool und Git & GitHub zur Versionsverwaltung verwendet.