

Designbericht für Prototyp

1. Allgemein

Im Softwaretechnik-Praktikum wird eine Webanwendung zur Begleitung des Übungsbetriebes erstellt. Um sich mit verschiedenen Konzepten der Architektur und Implementierung vertraut zu machen, wird eine Softwarestudie durchgeführt und die analysierten Konzepte in einem Prototyp umgesetzt. Diese Dokumentation verschafft einen Überblick über die verwendeten Designkonzepte sowohl in Textform, als auch in UML-Notation.

2. Produktübersicht

Der erstellte Prototyp orientiert sich stark an den Vorgaben aus Aufgabenblatt 2 und dem gegebenen Beispiel. Ziel ist es rollenbezogene Informationen über einen Webbrowser bereitzustellen. Dabei werden verschiedene Ansätze für die entgeltliche Anwendung festgelegt. In einem Vorstadium zum „Webinformationssystem“ werden folgende Dinge umgesetzt: der vom Benutzer eingegebene Benutzername und das Kennwort werden abgefragt, bei erfolgreicher Anmeldung wird eine neue Session geöffnet. Hier wird dem Nutzer in einem Navigationssystem die Möglichkeit gegeben zu verschiedenen Themen rollenbezogene Informationen zu erlangen. In Anlehnung an die Besprechung wird ein Link für einen Onlinechat für angemeldete Nutzer bereitgestellt, der Chat selber aber vorerst nicht realisiert. Dieser kann jedoch zu einem späteren Zeitpunkt noch umgesetzt und angefügt werden. Der Prototyp wird über verschiedene Klassen und Servlets realisiert, die in einem Container für Java-Servlet-Klassen, in diesem Fall Tomcat 5.0, abgelegt und bei Start des Prototyps daraus aufgerufen werden. Die angezeigten Daten werden aus Systemdateien gelesen oder vom Servlet selber erzeugt. Es ist demzufolge keine persistente Datenspeicherung notwendig.

Die Benutzer wurden in einer XML-Datei (demo-user.xml) angelegt. Aus dieser werden sie mit der Klasse „User“ ausgelesen und mit den vom Anwender eingegebenen Daten verglichen. Stimmen die Daten überein, so wird dem Anwender die für ihn festgelegte Rolle (oder auch Rollen) zugewiesen und ihm werden dann rollenspezifische Informationen zugänglich gemacht.

Die Navigation erfolgt über ein dynamisch erzeugtes Menü.

Um Missbrauch vorzubeugen, sind die unterschiedlichen Servlets (natürlich außer dem Login-Servlet) nicht einzeln aufrufbar. Sollte dies versucht werden, erscheint in einem Fenster „Sie sind nicht eingeloggt“ und der Link „jetzt einloggen“ zur Startseite. Des Weiteren ist die Datei mit den angelegten Benutzern nicht öffentlich zugänglich.

Da der Benutzer sich bei dem Server anmeldet und dieser ihm bestimmte Optionen offeriert, kann die gesamte Anwendung als Client-Server-System betrachtet werden.

3. Design-Entscheidungen

a) Grundsätzliche Design-Entscheidungen

Zur Umsetzung des Prototyps stehen unterschiedliche Ansätze und Konzepte zur Verfügung. Die Auswahl der Benutzer hat verschiedene Gründe, sie wurden entweder im Praktikum vorgegeben oder von den Implementierern der Übungsgruppe als günstiger bewertet. Bei der Begründung der verwendeten Konzepte wird jedoch auf diese Unterscheidung verzichtet.

Der Prototyp wird über Servlets verwirklicht, die in Java programmiert werden. Die Ausführung der Java-Programme und die Bereitstellung der Verbindung zum Client etc. erfolgt dabei durch einen sogenannten Servlet-Container. Diese Technologie hat den Vorteil das die Programme über http mit einem Client in Verbindung stehen, dessen Anfragen empfangen, verarbeiten und beantworten können, dabei aber für den Implementierer transparent und bearbeitbar sind. Die Implementierung wird durch den Zugriff auf das komplette Java API und viele Dritthersteller Java-Bibliotheken erleichtert. Außerdem können Servlets relativ plattformunabhängig eingesetzt werden.

Um auf Servlet-erstellte Daten zuzugreifen oder andere aus Systemdateien zu lesen, wird eine User.class verwendet. Diese liest unter Verwendung des Standards Sax die Daten aus und sendet sie an die jeweilige Java-Klasse. Der Einsatz von Sax wurde aus folgenden Gründen gewählt: es ist ressourcenschonend und schnell, wodurch insbesondere die Suche von kleinen Informationsteilen in großen XML-Dokumenten effektiv möglich ist. Außerdem ist eine flexible Manipulation durch Außenstehende, die insbesondere von DOM unterstützt wird, hier nicht notwendig. Das Model-View-Controller-Modell wird berücksichtigt, um Fachkonzept und Nutzerschnittstellen voneinander zu trennen. Dies geschieht durch verwenden der Servlets als Controller. Die View-Schicht wird durch ein Stylesheet realisiert. Durch diese Abgrenzung werden Layout-Änderungen wie Farbanpassungen, Schriftgröße, etc. auch ohne fundierte Java-Kenntnisse möglich. Die Model-Schicht wird durch die Java-Klassen symbolisiert, auf die durch die Servlets zugegriffen wird.

b) Layout und logischer Aufbau

Zu den grundsätzlichen Designentscheidungen gehört auch das Layout. Hier haben wir uns an dem gegebenen Beispiel orientiert.

Die Oberfläche ist wie folgt aufgegliedert:

Auf der Startseite befindet sich unter einer Überschrift „Übungsbetrieb“ und der Aufforderung „Registrierte Nutzer: Bitte geben sie Benutzernamen und Passwort ein:“ ein Fenster für die Interaktionen zum Login. Hier befinden sich zwei Eingabefelder mit ihrer Bezeichnung auf der linken Seite die untereinander angeordnet sind. Das obere mit der Bezeichnung „Login“ ist für die Eingabe des Benutzernamens und das untere für das Passwort. Außerdem befinden sich unterhalb der Eingabefenster zwei nebeneinanderliegende Button mit den Aufschriften „Anmelden“ und „Löschen“. Bei Betätigung des Löschen-Button werden die gerade in die Eingabefelder eingegebenen Daten gelöscht. Bei Benutzung des Anmelde-Button findet eine Überprüfung des Benutzernamens und des Passworts über Aufruf der user.class mit den in der demo-user.xml eingetragenen Benutzern statt. Konnte dabei keine Übereinstimmung gefunden werden erscheint der Hinweis „Die Eingabe war leider falsch, versuchen sie es erneut“ und ein Link „Neuer Versuch“ zur Startseite.

War der Login erfolgreich, konnte die Rolle des Nutzers identifiziert werden und die ihm jetzt zugänglichen Seiten enthalten für seine Rolle festgelegte Daten.

Im Prototyp stehen folgende Benutzer zur Verfügung:

Benutzername	Passwort	Rolle
schulze	ezluhcs	student
meier	reiem	student,tutor
admin	nimda	admin
lehmann	nnamhel	betreuer

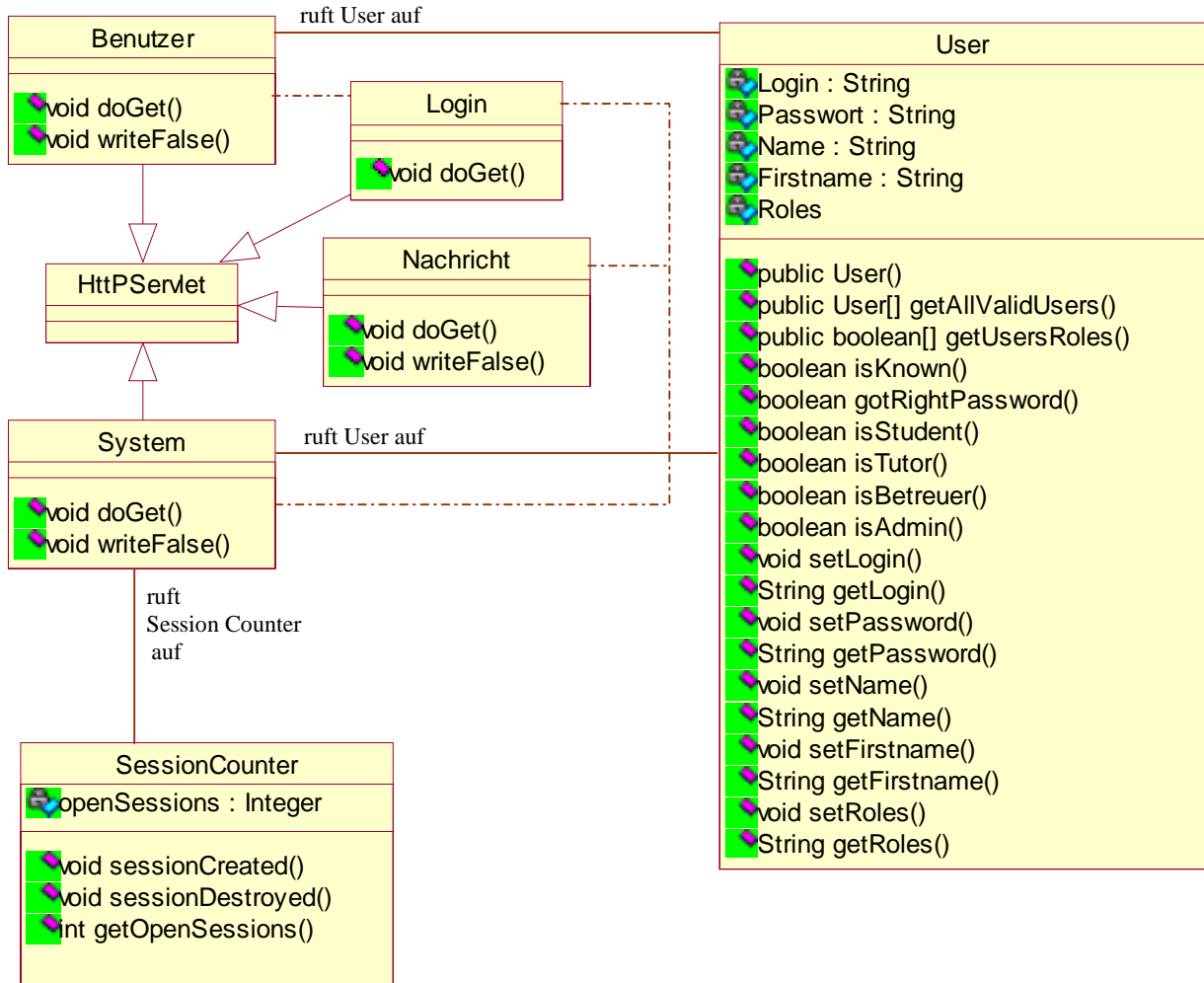
Der Grundaufbau der nächsten Seiten ist identisch. Sie bestehen aus einem Interaktionsfenster mit einer Linkleiste am linken Rand, in dem alle aufrufbaren Funktionen angegeben sind. Dabei ist die Ebene auf der man sich gerade befindet im Gegensatz zu den anderen nicht unterstrichen.

Nach einloggen befindet man sich auf der Benutzer-Ebene. Hier wird der Nutzer begrüßt und ihm sein Name und seine Rollen angezeigt. Unter Ebene ‚Benutzer‘ befindet sich in der Linkleiste die Ebene ‚System‘. In dieser sehen alle Nutzer die Uhrzeit. Der Student sieht außerdem seine Session ID und die Session Create Time. Der Tutor sieht außer der Uhrzeit noch die Session Create Time jedes Nutzers der gerade online ist. Die rollenspezifischen Angaben des Administrators bestehen aus der Anzahl der gerade angemeldeten Nutzer und Session Create Time für jeden Nutzer der derzeit online ist. Dem Betreuer sind diese Informationen ebenfalls zugänglich, aber zusätzlich wird ihm seine Remote Adress, also seine IP angezeigt.

Es ist möglich, das ein Nutzer mehreren Rollen angehört. Es werden ihm dann alle Informationen die den einzelnen Rollen zugänglich sind im Bereich ‚System‘ angezeigt. In der Ebene ‚Nachrichten‘ erscheint nur der Vermerk „Hier könnte ein Forum installiert werden.“, da der Chat vorerst nicht umgesetzt wird.

Unter den Links befindet sich noch ein Button mit der Aufschrift „Abmelden“. Bei Betätigen dieses Buttons wird die Session des Nutzers beendet und es erscheint die Startseite.

4. Klassenstruktur



----- von der Klasse „Login“ wird login und passwort an die Klassen „Benutzer“, „System“ und „Nachricht“ übergeben