

Designbeschreibung für Prototyp

demo.login des Übungsverwaltungsprogramm

Allgemeines:

Der Prototyp soll mit Hilfe der Servlet / Java / XML / Html / Stylesheet Technologie eine Session über einen Webbrowser für eine Person ermöglichen, welche sich mit Name und Passwort anmeldet und dann verschiedenen Funktionen und Informationen abrufen kann.

Benutzt wird dafür ein Apache Tomcat Server als Servlet-Container.

Der Prototyp wird über einen Link auf unsere Gruppenseite oder über Port: 9011/servlet/demo.login aufgerufen.

Cookies sind unbedingt erforderlich.

Produktübersicht:

Beim aufrufen des Servlets kommt ein Loginbildschirm der nach Passwort und Name fragt. dies macht login.java

Die Eingabedaten werden durch GetPass.java mit der demo-users.xml verglichen und User.java berechnet dann das weitere vorgehen → Passwort Falsch → kein Zugriff → neu einloggen.

Passwort richtig → abrufen der Benutzerspezifischen Informationen und Servletinfos und aufrufen eines Willkommensbildschirm.

HTMLout.java gibt dabei den Html code für den Browser aus je nachdem was gerade gefordert ist.

Durch ein Menü kann der User nun Benutzer- oder Systeminformationen ausgeben lassen wobei nur diese welche per Aufgabenstellung definiert sind angezeigt werden.

So sieht z.B. nur der Admin und Betreuer die Anzahl aller eingeloggten User.

Alle Daten werden in einer Tabelle angezeigt welche sich ggf. dynamisch ändert.

Alle Nutzer können einen Chat benutzen welcher die aktuellen Texte speichert diese aber auch wieder löscht. Dieser Chat ist wie ein allgemein bekannter Onlinechat gestaltet:

Man sieht ein Fenster in dem alle Nachrichten mit Nutzernamen ablaufen und man hat ein Eingabefeld (in dem der eigene Name schon steht) um eigene Nachrichten zu senden welche dann von allen anderen gelesen werden können.

Beim ausloggen werden entsprechende Attribute zurückgesetzt (#User) und das Servlet entfernt.

Danach hat der Nutzer die Möglichkeit sich neu anzumelden oder zu Homepage zu gehen.

Weiters siehe nächste Seite

Grundsätzliche Design-Entscheidungen

Äußeres:

Um das Erscheinungsbild des Designs im Webbrowser schnell ändern zu können benutzen wir Stylesheets ausgelagert in eine .css Datei welche u. a. das Hintergrundbild oder div. Schriftformationen beinhaltet.

Da sich um ein universitäres Verwaltungsprogramm handeln soll verzichten wir auf zu viele Farben und Bildchen und setzen lieber auf Übersichtlichkeit und Funktionalität.

Dazu benutzen wir u. a. dynamische Tabellen und Menüs welche auch bei anderen Auflösungen benutzt werden können.

Das bewegen zwischen verschiedenen Seiten oder Abrufen der Informationen soll über eindeutige Buttons geschehen welche zusammenhängend dargestellt sind z.B. in einer Menüführung links am Bildschirm und nicht über wild verstreute links, um Übersichtlichkeit und Funktionalität zu wahren.

Einfache Benutzerspezifische Informationen sollen im selben Fenster / Tabelle dargestellt werden andere Sachen wie der Chat wird extra dargestellt da dieser etwas komplexer ist.

Login / out Menü ist so gestaltet das Fehler minimiert werden und der Benutzer eindeutig weiß was zu machen ist. Ach sollte man nicht einfach auf einer Seite feststecken und evtl. nur über den Browser wechseln können, deswegen wird auf jeder Seite ein entsprechendes Menü oder ein Button zur Hauptseite gesetzt. Somit ist die einwandfreie Seitenführung gewährleistet.

Internes:

Die Klassenstruktur ist so gehalten das wichtige Funktionen gleicher Art auch in einer Klasse bleiben aber nicht zuviel oder zu wenig Servlets / Klassen gebraucht werden.

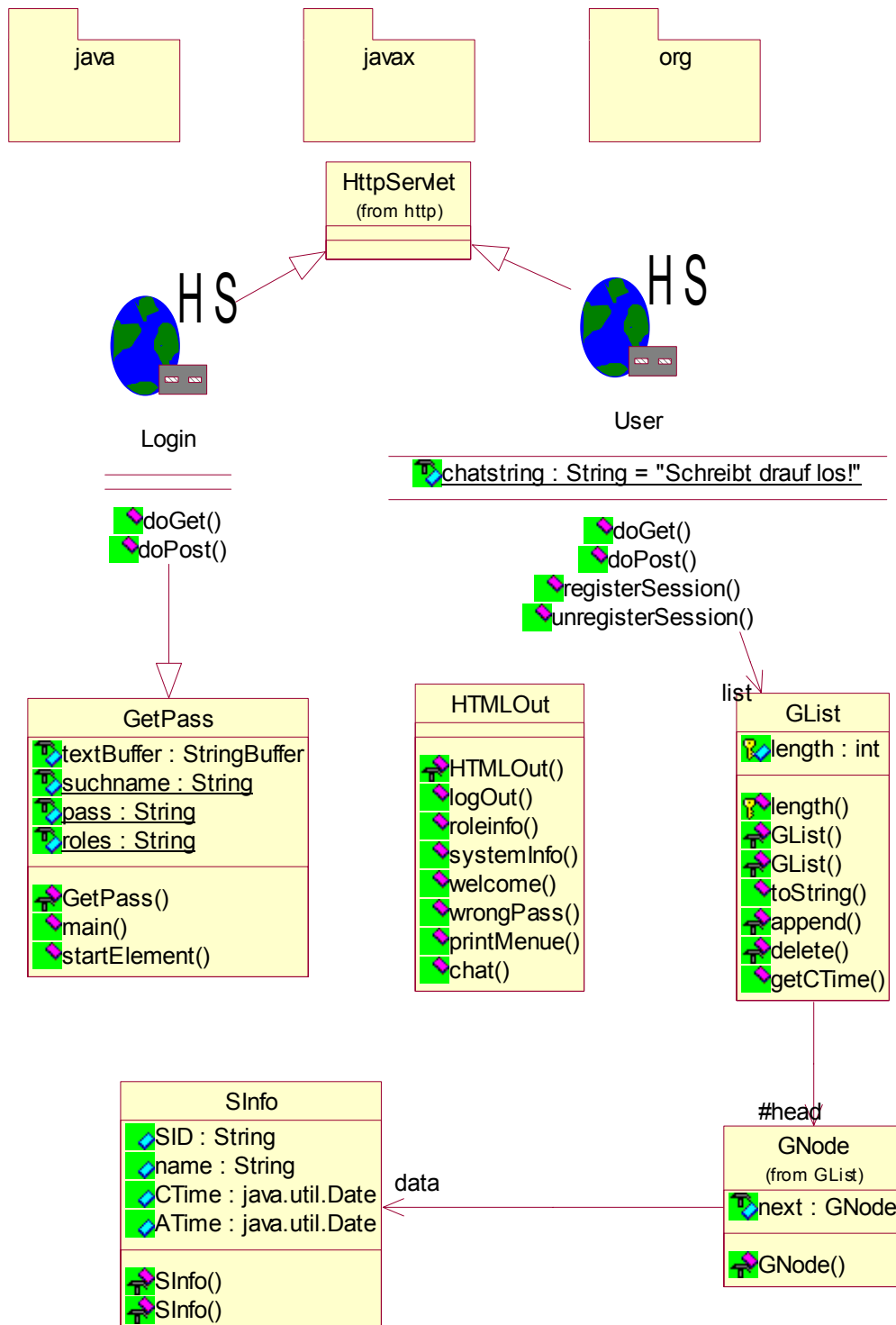
Sicher könnte man den Prototyp in einer einzigen Klasse bauen diese wäre aber sehr unübersichtlich. Baut man aber zu viele gehen die einzelnen Zusammenhänge zwischen Attributen und Methoden verschiedener Klassen evtl. verloren.

Wir haben uns deswegen auf 6 Klassen beschränkt.

Wie oben schon beschrieben z.B. eine Klasse die sich um die HTML Ausgabe kümmert eine die Passwörter überprüft usw.

Die .java Dateien sind gut strukturiert und dokumentiert was spätere Änderungen vereinfacht. Extra Informationen die von mehreren Users gebraucht werden sind in verketteten Listen gespeichert.

Paket und Klassenstruktur



Extraklassen für die Informationen der User create Time und login Informationen