Verantwortlicher: Matthias Lehmann

Gruppe: UE12
Datum: 11.05.03

Aufgabenserie 2 – Designbeschreibung Prototyp

1.Allgemeines

Der zu erstellende Prototyp soll ein Portal für Studenten, Betreuer, Tutoren und Administratoren sein, die sich mit einem Passwort und Benutzernamen dort einloggen.
Die unterschiedlichen Nutzer haben allerdings nicht alle die gleichen Informationsrechte, womit je nach Nutzer entsprechend dynamische Seiteninhalte generiert werden müssen. Zum Lösen dieser Aufgabe soll von der Verwendung von JavaServerPages (jsp) abgesehen werden und sich nur auf das Konzept der JavaServlets konzentriert werden.
Unser Prototyp besteht aus 15 Dateien, wovon 10 Servlets sind und 5 Hilfsklassen zur Bearbeitung.

2. Produktübersicht

Wir müssen uns momentan nur für einen, der hier aufgelisteten und erläuterten, XML-Parser entscheiden. Entscheidung Siehe 3.

SAX (Simple API for XML)
SAX ist eine Programm-Schnittstelle (Application Programmers Interface API) für die Verarbeitung einer Klasse von XML-Dokumenten, also einer XML-Applikation, mit Hilfe einer objekt-orientierten Programmiersprache. SAX liefert ein XML-Element nach dem anderen in einem Eingabestrom und eignet sich daher auch für sehr große XML-Files.

DOM (Document Object Model)
DOM ist ein Objektmodell, es beschreibt die in einem Dokument einer bestimmten XML-Anwendung enthaltenen Elemente als Objekte. DOM liefert eine komplette Baumstruktur aller Objekte eines XML-Dokuments und eignet sich daher nicht für extrem große XML-Files.

3. Grundsätzliche Design-Entscheidungen

XML

Wir haben uns hier für einen DOMParser, da es mit ihm sehr leicht möglich ist, aufgrund der Eigenschaft das er das XML-Dokument wie einen Baum behandelt, das traversieren über die einzelnen Knoten leicht zu handhaben. Außerdem wird die demo-users.xml nicht sehr groß, womit es keine Probleme mit dem DOMParser geben dürfte.
Nach erfolgreichen Tests haben wir dann die Validierung des Parsers ausgeschaltet um die Leistung zu steigern, was allerdings bei der geringen Dateigröße der demo-users.xml keine spürbaren Effekte brachte.

Zwischenspeicherung

Die Benutzer werden in einer speziellen Liste (selbsterstellt) verarbeitet, die es ermöglicht die geforderten Informationen der Benutzer nach Belieben einzusehen bzw. zu verändern. Somit ist man flexibler als mit einem Array oder einem Vektor.
Sitzungsinformationen werden ebenfalls in einer speziellen Liste gehalten um besser auf entsprechende Anforderungen zu reagieren.

Dynamische Navigation

Um die Rollenbezogene Navigation zu gewährleisten wurde für jeden Bereich (persönliche Daten, technische Daten, rollenbezogener Bereich) ein einzelnes Servlet konzipiert, welches anhand der Rollenfunktion des Nutzers, die Ausgabe individuell erstellt.

4. Paket- und Klassenstruktur

!Alle fettgedruckten Wörter sind Klassennamen!