

Deckblatt

Abgabe der Gruppe 15

(Homepage der Ueb15: <http://pcai003.informatik.uni-leipzig.de/~ueb15>)

Aufgabe 2

„Präsentation des Prototyps der Softwarestudie Lastenheft und Glossar“

Gruppenmitglieder und aktuelle Rollenverteilung :

Name	Funktion	Email
Frank Jühling	Projektleiter	mai01dca@studserv.uni-leipzig.de
Markus Scholz	technischer Assistent Verantwortlicher für Recherche	mai01dnn@studserv.uni-leipzig.de
Frank Nowak	Verantwortlicher für Design	mai01chu@studserv.uni-leipzig.de
Philipp Metz	Verantwortlicher für Tests	Metz.Philipp@t-online.de
Stefan Nordt	Verantwortlicher für Implementation	mai01chs@studserv.uni-leipzig.de
Stefan Schmidt	Verantwortlicher für Dokumentation	derstift@web.de

Sie finden die folgenden Dateien im Verzeichnis Submit am Sonntag, dem 11.05.03 :

Dateiname	Teilaufgabe	Bearbeitet durch
Aufgabe2.pdf (dieses Dokument)	Designbeschreibung	Markus Scholz, Frank Jühling
PrototypUeb15_bin.jar	Prototyp Klassen und alle benötigten XML Dateien (inklusive web.xml und Stylesheet).	Markus Scholz, Frank Jühling
PrototypUeb15_src.jar	Prototyp Quellen und javadoc Output (HTML).	Markus Scholz, Frank Jühling

Anmerkung : Die Aufgabenstellung wurde so geteilt das der Rest des Teams die letzte Abgabe (am 4.5) gelöst hat, so dass Frank und Markus sich komplett auf den Prototyp konzentrieren konnten. Die anderen Mitglieder des Teams können sich aber jederzeit einen Überblick über den Quellcode (CVS) verschaffen und sich somit auch entsprechend in die Materie einarbeiten.

Zusammenstellung dieser Ausarbeitung durch Markus Scholz.

Seiten : 11 (inklusive Deckblatt)

1. Allgemeines

Charakterisierung

Zur Erfüllung der Aufgabe 2 galt es neben Lastenheft, Glossar und diesem Dokument einen ersten Prototyp zu erstellen. Dies sollte dazu dienen sich mit den grundlegenden Konzepten der Java-Servlet Technologie vertraut zu machen und sich in den Umgang mit XML Dateien in Java einzuarbeiten. Die sehr beschränkte Funktionalität des Servlets sollte dabei folgende Aufgaben erfüllen können :

- Abfrage von Benutzername und Kennwort zum Einloggen eines Benutzers, der in der XMLDatei "demo-users.xml" definiert wurde
- Eröffnung einer Session für den angemeldeten Benutzer.
- Ausgabe rollenbezogener Informationen durch Navigation des angemeldeten Benutzers:
Benutzername [Rollen: Student, Tutor, Betreuer, Admin]
Rollen [Rollen: Student, Tutor, Betreuer, Admin]
Uhrzeit [Rollen: Student, Tutor, Betreuer, Admin]
Remote Address [Rolle: Betreuer]
Session ID [Rolle: Student]
Session Create Time [Rolle: Student]
- Anzeige Anzahl der derzeit angemeldeten Nutzer [Rolle: Betreuer, Admin] und dessen Session Create Time [Rollen: Betreuer, Tutor, Admin]
- Zur Navigation sollte ein dynamisch erzeugtes Menü verwendet werden
- Nachrichtenversand an angemeldete Nutzer
Hier wurde eine kleine Änderung vorgenommen. Da es uns sinnvoller erschien die Nachrichten nicht nur versenden zu können, wenn der User online ist, sondern auch sonst (eine Art Mailbox), wurde der Code entsprechend modifiziert.
- Design der ausgegebenen HTML-Seiten sollte über ein CSS festgelegt werden

Der Prototyp sollte außerdem unter der, in der Aufgabe, angegebenen Adresse zur Überprüfung zur Verfügung stehen.

Alle diese Aufgaben wurden mit dem derzeitigen Prototyp realisiert.

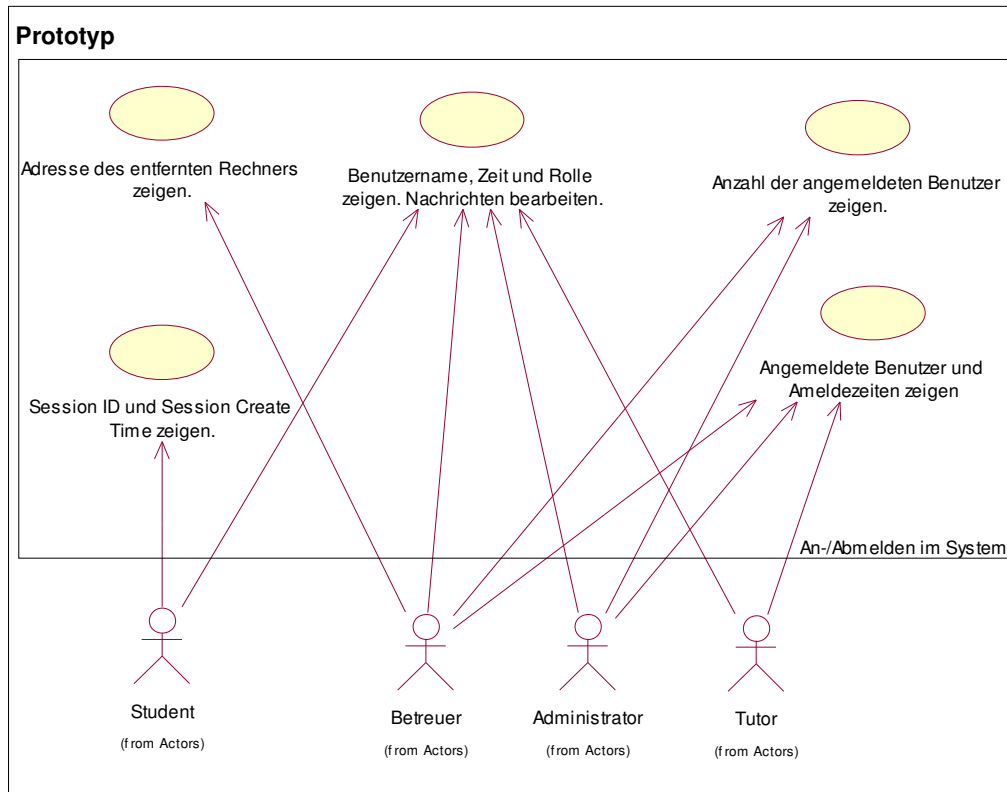
Demo - URL : <http://pcai003.informatik.uni-leipzig.de:8015/ueb15/servlet/demo.login>

Voraussetzungen des System und Produktumgebung

Grundvoraussetzung ist ein Apache Tomcat Server 4+ oder ein anderer Java-Servlet fähiger Webserver und eine für den Server entsprechend konfigurierte web.xml Datei (ist, für den Tomcat, in der PrototypUeb15_bin.jar enthalten). Weiterhin sollte das Sun Java SDK mindestens ab Version 1.3.1 installiert sein, zusätzlich sollte der Xerces Java Parser vorliegen. Das Betriebssystem spielt keine Rolle. Aufgrund der geforderten Software sollte es sich daher mindestens um einen Rechner der Pentium II - Klasse (oder äquivalente Modelle anderer CPU Hersteller) oder höher handeln. Vor dem Starten des Programms ist es nötig die Quellen neu zu übersetzen und die Pfade der einzelnen Dateien (XML, CSS) in der XMLServlet.java anzupassen.

2. Produktübersicht

Umweltdiagramm des Systems :



Produktfunktionen :

Anwendungsfall: Anmeldung
Akteure : Student, Tutor, Betreuer, Administrator
Beschreibung: Anmeldung zu einer Sitzung durch Angabe von Loginname, Passwort und Rolle.
Ergebnis: Zugang zu der Account-Seite und (nicht sichtbar) Erstellung einer neuen SessionID sowie Eintragen des Users in die `logged.in.xml`.
Ist der Nutzer eingeloggt sind weitere rollenspezifische Aktionen möglich.
Auf der Startseite erscheinen außerdem Rolle der Users und aktuelle Zeit.
Voraussetzung: Es existiert ein Benutzer mit dieser Username/Passwort Kombination, der die angegebene Rolle hat, in der Datei `demo-users.xml`.

Anwendungsfall: Abmeldung
Akteure : Student, Tutor, Betreuer, Administrator
Beschreibung: Abmeldung von einer Sitzung
Ergebnis: Verlassen des Systems. Schließen der, bei der Anmeldung angelegten, Session. Löschen der SessionID und Austragen des Nutzers aus der `logged.in.xml`.
Voraussetzung: Nutzer ist im System angemeldet.

Anwendungsfall: Nachrichten lesen
Akteure : Student, Tutor, Betreuer, Administrator
Beschreibung: Lesen der Nachrichten die dem Nutzer zugegangen sind.
Ergebnis: Nachrichten werden auf dem Bildschirm angezeigt. Nachrichten werden aus der `messages.xml` gelesen und mit Absender und Versandzeit dargestellt.
Voraussetzung: Nutzer ist im System angemeldet.

Anwendungsfall: Menüpunkt System wird gewählt
Akteure : Student
Beschreibung: Menüpunkt „System“ bietet rollenspezifische technische Informationen.
Ergebnis: Session ID und Session Create Time dieses Benutzers werden angezeigt.
Voraussetzung: Nutzer ist im System angemeldet.

Anwendungsfall: Menüpunkt System wird gewählt
Akteure : Tutor
Beschreibung: Menüpunkt „System“ bietet rollenspezifische technische Informationen.
Ergebnis: Angemeldete Benutzer, mit Anmeldezeit werden angezeigt.
Voraussetzung: Nutzer ist im System angemeldet.

Anwendungsfall: Menüpunkt System wird gewählt
Akteure : Administrator
Beschreibung: Menüpunkt „System“ bietet rollenspezifische technische Informationen.
Ergebnis: Namen, Anzahl und Anmeldezeiten der angemeldeten Benutzer werden gezeigt.
Voraussetzung: Nutzer ist im System angemeldet.

Anwendungsfall: Menüpunkt System wird gewählt
Akteure : Betreuer
Beschreibung: Menüpunkt „System“ bietet rollenspezifische technische Informationen.
Ergebnis: Namen, Anzahl und Anmeldezeiten der angemeldeten Benutzer werden gezeigt.
Zusätzlich wird die Adresse des entfernten Rechners des Betreuers angezeigt.
Voraussetzung: Nutzer ist im System angemeldet.

Anwendungsfall: Nachrichten löschen
Akteure : Student, Tutor, Betreuer, Administrator
Beschreibung: Alle Nachrichten eines Benutzers werden gelöscht.
Ergebnis: Alle Nachrichten eines Users werden aus der Datei `messages.xml` entfernt.
Eine Bestätigungsmeldung erscheint auf dem Bildschirm.
Voraussetzung: Nutzer ist im System angemeldet.

Anwendungsfall: Nachrichten schreiben
Akteure : Student, Tutor, Betreuer, Administrator
Beschreibung: Eine Nachricht wird an einen ausgewählten User geschickt.
Ergebnis: Die eingegebene Nachricht wird in die Datei `messages.xml` eingetragen, der Absender und die Zeit werden dort ebenfalls vermerkt. Eine Bestätigungsmeldung erscheint auf dem Bildschirm.
Voraussetzung: Nutzer ist im System angemeldet.

3. Grundsätzliche Designentscheidungen

Da uns nahe gelegt wurde XML für die Abspeicherung von Daten zu nutzen, setzt der Prototyp bei seiner Datenablage komplett auf dieses Format. Um XML und Java miteinander zu verbinden gibt es viele verschiedene Bibliotheken (SAX, Xerces, etc.). Unser Prototyp benutzt zum größten Teil die in Java 1.4.X bereits integrierten SAX Funktionen um mit XML zu arbeiten. Lediglich zum Anlegen neuer Dokument wir das Xerces Paket benötigt. Weiterhin galt zu überlegen wie das Management der einzelnen Rollen aussehen sollte. Wir haben uns (siehe auch die Klassendiagramme in 4) für eine Teilung in einzelne Klassen je nach Rolle entschieden, wobei es eine Rollengrundklasse gibt von der die anderen Rollen erben. Da diese Basis-Klasse aber noch keinerlei XML- oder Servlet-Methoden besitzt müssen diese 2 Klassen die Obersten sein. Es ergibt sich folgende Hierarchie :

XML-Klasse ← Basis-Servlet ← Rollenklasse mit den grundlegenden Methoden ← andere Rollen

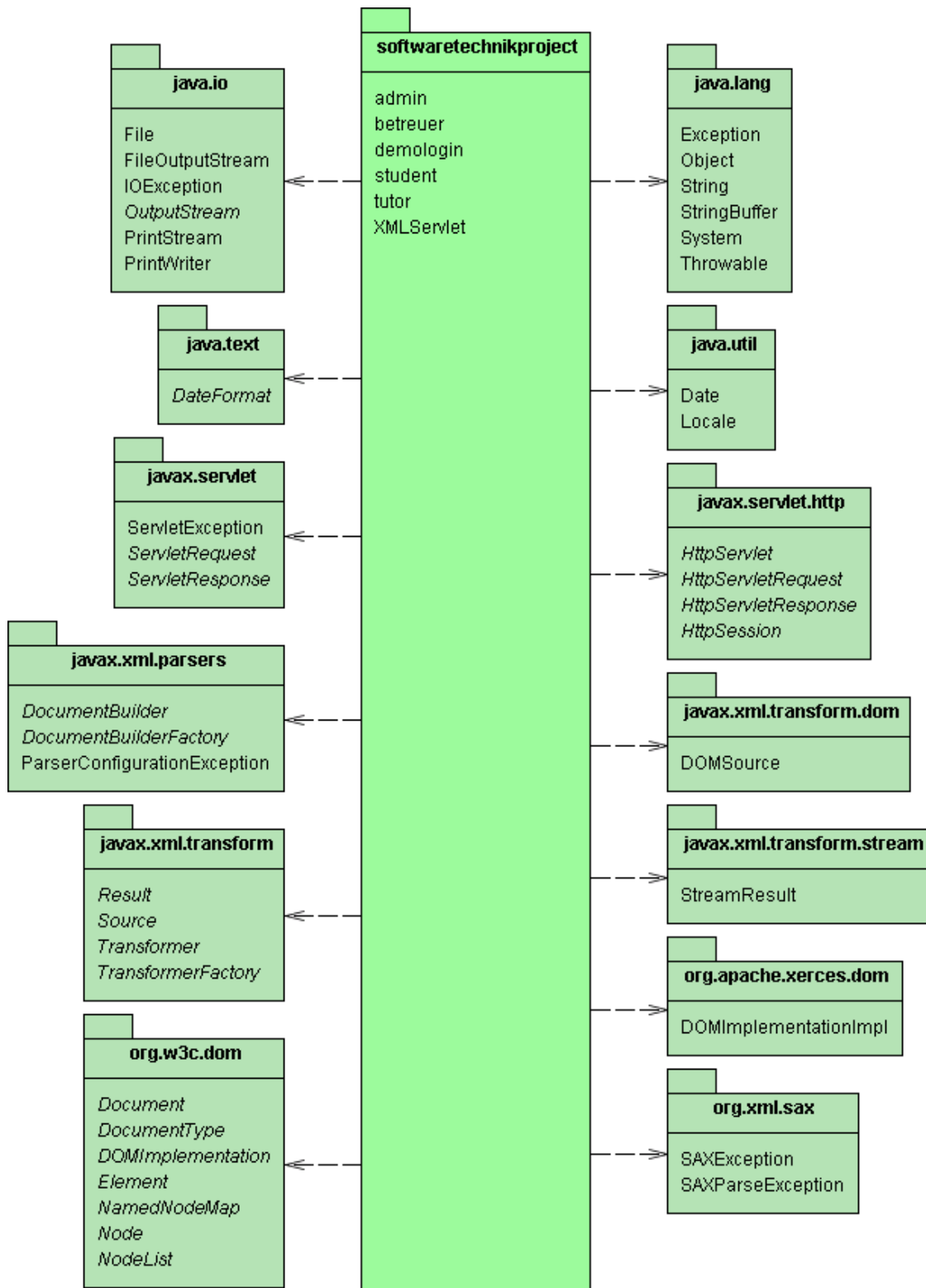
Weiterhin gibt es für jede (datenabhängige-) Funktion eine XML Datei : Für die Nachrichten, eingeloggte Benutzer und, wie vorgeschrieben, für die Anmeldung. Da das XML Format ein sehr variables, gut zu bearbeitendes Format ist. Sind Erweiterungen , z.B. das Loggen der IP Adresse beim Nachrichtenversenden, kein Problem. Man würde lediglich in den entsprechenden Funktionen die XML Node um diese Childnode (z.B. „IP“) erweitern müssen und beim Laden darauf achten das diese Childnode auch ausgelesen wird.

Damit in den Javasourcecode nicht auch noch die ganzen HTML Layoutdetails einfließen müssen, wurde die Idee solche Dinge in einem Stylesheet zu realisieren, sehr gut angenommen und wird auch später definitiv im Endprodukt zu finden sein. Denn es erspart nicht nur den Entwicklern viel mehr Schreibaarbeit sondern ermöglicht auch den Anwendern ein, zumindest layouttechnisch gesehen, individuelles Aussehen des Systems.

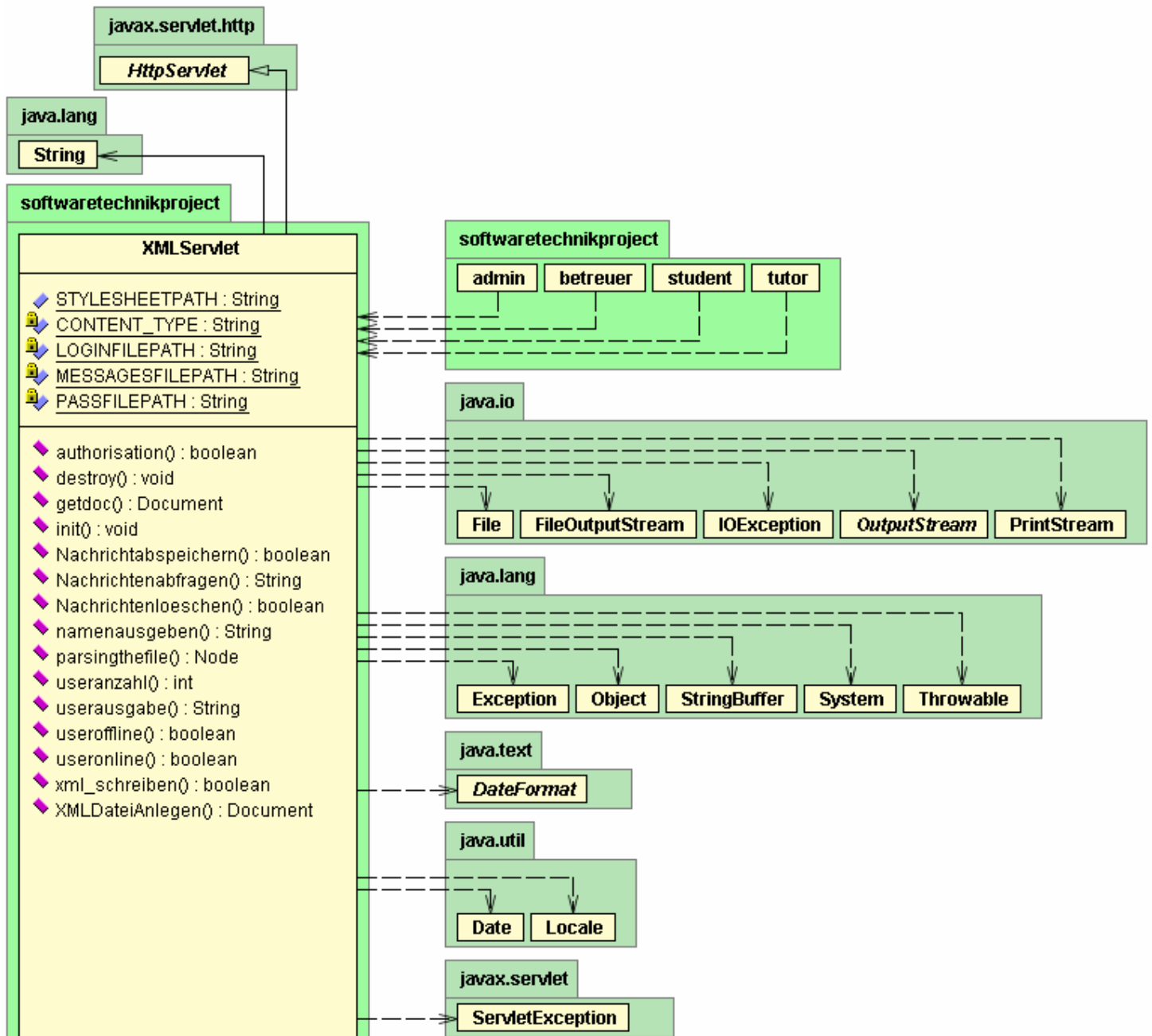
Es wurden außerdem noch viele kleine Entscheidungen bezüglich des Designs getroffen.

4. Paket- und Klassenstruktur

Struktur des Pakets „softwaretechnikproject“ :



Die einzelnen Klassen des Prototyps :



...
...

...
...

