

Designbericht für ISIS

Version	Datum	Kommentar
Version 1.0	16.06.2003	umgesetzt wurde die Oberfläche und Ein-/Ausgang für gesonderte Benutzer. Es ist ein Überblick über (fast) alle Menüpunkte des Endproduktes und die Organisation dieser.
Version 2.0	23.06.2003	es wurde hauptsächlich das Auslesen der User aus der XML-Datei und die Up-/Download Funktion für Vorlesungs- und Übungsbegleitende Materialien realisiert
Version 3.0	30.06.2003	Umstrukturierung des Programms und erweitern der Funktionen des Seminarleiters, Neuanmelden der Studenten
Version 4.0	07.07.2003	Umsetzung aller wichtigen Funktionen um ISIS sinnvoll nutzen zu können

1. Allgemein

Bei ISIS (Internet-Studenten-Informationssystem) handelt es sich um ein webgestütztes Informationssystem, welches Studenten und Mitarbeiter einer Universität eine Begleitung zum Übungsbetrieb bietet. Da es sich hier um eine Praktikumsarbeit an der Universität Leipzig handelt werden die Versionen bis zur endgültigen Fertigstellung sich an der Fakultät Informatik der Uni Leipzig orientieren.

Der Seminarleiter als Hauptverantwortlicher für den Übungsbetrieb erhält die Möglichkeit Übungsgruppen und dazugehörige Materialien zu verwalten. Außerdem kann er Mitarbeiter anlegen, die über dieses System Übungsaufgaben korrigieren und studentische Leistungen registrieren. Studenten wird die Anmeldung zu Übungen und Klausuren gegeben, die Abrechnung der Übungsserien erleichtert und ein Überblick über die erbrachten Leistungen gegeben.

Im nachfolgenden werden die Konzepte des endgültigen Produktes und der jeweiligen Versionen dargelegt. Dabei soll ein einfacher Überblick über die Funktionalität und die dazu nötige Implementierung gegeben werden.

2. Produktübersicht

Das ISISystem ist eine reine Internet Anwendung. Ziel ist es den verschiedenen Nutzern die Möglichkeit zur Informationsweitergabe und Informationserlangung, sowie zur Kommunikation zu geben. Nutzer sind Studenten, Seminarleiter, Dozenten, Kontrolleure, Gäste und Administratoren.

Der Administrator hat eine gesonderte Ansicht auf das System. Wie sein Rollenname bereits deutlich macht ist er für die Installation, Konfiguration, Wartung und Deinstallation der Applikation verantwortlich. Dazu gehört auch das Anlegen der Seminarleiter und Dozenten mit der Zuweisung ihrer Passwörter. Die anderen 5 Benutzergruppen greifen über eine Webschnittstelle auf das System zu. Gäste haben nur Zugriff auf die allgemeinen Links auf der Startseite. Die Gruppen Student, Seminarleiter, Dozent und Kontrolleur besitzen einen Zugangsnamen und ein Passwort. Damit wird ihnen ermöglicht auf speziell für sie ausgewählte Anwendungen zuzugreifen.

Die Anwendungen sind inhaltlich und optisch in verschiedene Bereiche aufgeteilt. Die Navigation zwischen diesen erfolgt über ein dynamisch erzeugtes Menü. So kann der Seminarleiter unter anderem seine Kontrolleure anlegen, Übungsgruppen konfigurieren, Übungsmaterial, Übungsaufgaben und Musterlösungen zum Download bereitstellen und von anderen eingestellte Informationen erhalten. Dozenten können Einschreibungen für Klausuren organisieren, ihre Vorlesungsdaten eingeben und Vorlesungsmaterial einstellen. Der Student kann Übungsserien herunterladen, sie lösen und die bearbeiteten Aufgaben über das System abgeben. Nach Korrektur seiner Abgaben erhält er diese mit Kommentaren und einer Bewertung zurück. Der Kontrolleur wiederum wird in die Lage versetzt auf einfachem Wege die abgegebenen Aufgaben herunterzuladen und nach Kontrolle die kommentierten Dateien und die Punkte dem Studenten wieder zugänglich zu machen.

Die dabei anfallenden Daten der Übungsserien werden in Form von PDF's in zentralen oder falls es Studenten betrifft in persönlichen Ordnern abgelegt. Die während der Bearbeitung entstehenden Informationen wie Übungsgruppen mit Zeiten; Vorlesungen mit Zeiten; Benutzer mit Name, Passwort, Rolle und zugehörigen Übungsgruppen; Punktelisten; etc. werden in XML-Dateien gespeichert. ISIS ist über Java-Klassen und Servlets implementiert, deshalb besteht für den Anwender die Notwendigkeit eines Servlet-unterstützenden Webservers. Eine weitergehende Voraussetzung ist das Vorhandensein eines Servlet-Containers auf dem System auf dem die Applikation installiert werden soll.

Version 1.0

Bei der ersten Stufe der Umsetzung wurde der Rahmen für die spätere Implementierung abgesteckt. Dabei wurde weitgehend das entgeltliche Design festgelegt, die Mehrheit der Menüpunkte wurde in die Oberfläche integriert und dabei im Quelltext schon die Mehrheit der zur Implementierung notwendigen Klassen als eine Art Skelett angelegt.

Die Benutzergruppen Gast, Student, Seminarleiter, Dozent und Kontrolleur haben bereits die Möglichkeit zumindest einen Überblick über die endgültige Applikation zu erlangen.

Hierzu wurden, da das Auslesen und Schreiben der XML erst für die nächste Version vorgesehen ist, einige Benutzer statisch angelegt.

Jeder der Punkte im Menü, welches sich auf der linken Seite befindet, ist anwählbar. Später wird in dem Feld rechts die innere Funktion aufgerufen. Momentan wird eine kurze Beschreibung über die spätere Verwendung gegeben.

Version 2.0

In der ersten Version wurden die Benutzer nur statisch angelegt. Um später Benutzer auch aktiv nach Installation des Programms Benutzer anlegen (z.B. Studenten) und ihre Daten verifizieren zu können, werden diese in XML-Dateien gespeichert. In dieser Version wurde vorbereitend darauf bereits das Auslesen der Benutzer aus der XML-Datei realisiert. Hierzu wurde wie geplant SAX verwendet.

Bei den Daten innerhalb der Datei handelt es sich um die selbe Benutzerkonfiguration wie in Version 1.0.

Um eine Strukturierung der Implementierung beizubehalten, wird versucht die einzelnen Nutzer logisch nacheinander zu bearbeiten. Durch die gegebene Wichtigkeit und die Vielfalt der Aktionen beginnen wir mit dem Seminarleiter. Da es verschiedene sehr ähnliche Methoden zwischen unterschiedlichen Nutzern gibt, wird die Bearbeitung dieser vorgezogen. So ist der Quelltext der Methode „public void uploadTaskList()“ des Seminarleiters zum hochladen der Übungsaufgaben größten Teil identisch mit der „public void uploadScripts()“ des Dozenten zum Skripten hochladen.

Durch umsetzen von Methoden in den Klassen Seminarleader, Dozent, AdvancedMenu ergeben sich nun erstmals direkte Anwendungen der Applikation. Für Seminarleiter und Dozent sind die Upload Funktionen realisiert wurden. Gesondert kann der Seminarleiter bereits Kontrolleure anlegen. Allen Benutzergruppen ist es jetzt möglich die Materialien herunterzuladen. Das Hoch- und Runterladen der Übungen für Studenten und Kontrolleure gehört allerdings nicht dazu.

Version3.0

Bei der Umsetzung einiger XML-Funktionen wurde die Notwendigkeit des Umbaus der Struktur festgestellt. Diese wird in Abschnitt 4 erklärt.

Im Zuge dessen wurde auf die Umsetzung der „freigeben“-Funktion an allen Stellen verzichtet.

Es wurden für weitere Anwendungen (Neuanmeldung der Studenten, Anlegen der Gruppen, Anmelden an Gruppen) die Oberflächen programmiert.

Der Aufruf „Anlegen von Gruppen“ wurde als Funktion implementiert und steht nun dem Seminarleiter zur Verfügung.

Version4.0

Nach einem erneuten strukturellen Umbau besteht das Programm jetzt aus inhaltlich aufgegliederten Klassenpaketen. Dabei wurde unterteilt in Servlets, XML-relevante Dateien und Klassen, Hauptklassen und Anzeige-Klassen.

In dieser Version wurde versucht eine möglichst anwendungsfähige Version zu erstellen. Dazu mussten einige vorgesehene Anwendungen verworfen werden. Die nun durchführbaren Aktionen sind in Abschnitt 3 beschrieben.

3. Design-Entscheidungen

Im folgenden werden die für die Implementierung wichtigen Entscheidungen genauer beleuchtet. Es soll eine kurze, verständliche Zusammenfassung über die verwendeten Konzepte und zu programmierenden Klassen gegeben werden. Auf Grund der leichteren Verständlichkeit und da die Applikation noch nicht fertig gestellt ist, kann sie nicht vollständig sein und es können Abweichungen zum endgültigen Produkt auftreten. Die Abschnitte über die einzelnen Versionen beziehen sich hauptsächlich auf die Neuerungen im Programm. Anschließend wird der Aufbau des Designs erläutert, wie er sich den einzelnen Nutzern darstellt.

a) Grundsätzliche Design-Entscheidungen

ISIS wird über Klassen und Servlets realisiert, die Java-programmiert sind. Diese sind in einem sogenannten Servlet-Container abgelegt. In unserem Falle wird Tomcat 4.1.24 verwendet. Die Ausführung der Java-Programme und die Bereitstellung der Verbindung zum Client etc. erfolgt dabei über diesen Container. Diese Technologie hat den Vorteil das eine Verbindung mit einem http-Client besteht. Er kann Anfragen der Applikation empfangen, verarbeiten und beantworten, wobei die Applikation selber für den Implementierer transparent und bearbeitbar bleibt. Die Programmierung wird durch den Zugriff auf das komplette Java API und externe Bibliotheken Dritter erleichtert. Ein großer Vorteil von Servlets besteht in der plattformunabhängigen Einsetzbarkeit.

Alle Klassen, Servlets und Stylesheets werden in einer Sprache implementiert und kommentiert. Die Verwendung einer einzigen Sprache erhöht die Lesbarkeit und Verständlichkeit des Quelltextes. Die Wahl der Sprache fand unter zwei Gesichtspunkten statt: zum einen sollte der Quelltext möglichst international verständlich sein. Zum anderen wollen wir bei der Implementierung übliche Programmaufrufe und bereits in Java-Bibliotheken implementierte Methoden verwenden. Um beim Einsatz dieser nicht die Implementierungsregel der Benutzung einer einzigen Sprache zu brechen, wird das System, genauso wie die verwendeten Aufrufe und Methoden, in englisch implementiert und kommentiert. Der Hauptteil der Kommentierung kann mit javadoc extrahiert werden.

Eine weitere Regel für gute Implementierung ist die Berücksichtigung des Model-View-Controller-Modells. Dies ist besonders wichtig um Fachkonzept und Benutzerschnittstellen voneinander zu trennen. In der ISIS Applikation wird die View-Schicht über Stylesheets realisiert. Durch die so entstehende Abgrenzung werden Layout-Änderungen wie Schriftgröße, Farbe, etc. auch ohne fundierte Java-Kenntnisse möglich. Als Controller fungieren die Servlets. Die Model-Schicht wird durch die Java-Klassen symbolisiert. Der Zugriff auf diese erfolgt mit Hilfe der Servlets.

Wie bereit in Abschnitt zwei erwähnt, müssen bei der Verwendung des Systems Daten angelegt und abgefragt werden. Für diese Art von Daten werden XML-Dateien verwendet. In ihnen werden z.B. die Nutzer mit ihren Passwörtern und Rollen festgehalten, die Konfiguration der einzelnen Übungsgruppen mit Seminarleiter, Kontrolleur und eingeschriebenen Studenten, Klausuren mit eingeschriebenen Studenten sowie die Punktelisten der Übungen und Klausuren.

Um diese Daten in die jeweilige Datei zu schreiben wird DOM verwendet. Der Vorteil liegt hier darin, das durch DOM das gesamte Dokument eingelesen und dabei eine Baumstruktur gebildet wird. Durch die Strukturierung ist es sehr einfach Informationen an einer bestimmten Stelle einzufügen.

Der Aufruf zum Schreiben in eine XML-Datei erfolgt hauptsächlich in den Menü-Klassen. Z.B. werden durch die Methode createGroups die für eine Übungsgruppe erforderlichen Daten wie Zeit, Raum, max. Studentenzahl, etc in die XML-Datei GroupList geschrieben.

Zum Auslesen der Daten verwenden wir SAX. Dieses wurde aus verschiedenen Gründen gewählt: die Datei wird nicht wie bei DOM erst vollständig in den Hauptspeicher geladen, sondern Zeile für Zeile durchgearbeitet. Bei Auffinden der gefragten Information wird die Bearbeitung der Datei abgebrochen und eine vorher definierte Aktion ausgeführt. Sollte die Information im Dokument nicht enthalten sein wird am Ende eine für diesen Fall bestimmte Aktion gestartet. Durch diese

Vorgehensweise ist SAX besondersst ressourcenschonend und bei der Suche von Informationsteilen in großen XML-Dokumenten überaus effektiv. Ein Beispiel zu SAX ist das Überprüfen der Login-Eingaben „login“ und „password“ durch Aufruf der Klasse user und der User.xml.

Für die Sicherheit des Systems ist es wichtig den Aufruf einzelner Klassen ohne Login zu unterbinden. Sollte dies versucht werden, erscheint ein Fenster mit der Bemerkung „Sie sind nicht eingeloggt“. Des weitem sind weder die PDF noch die XML Dateien frei zugänglich.

b) Layout und logischer Aufbau

Das ISISystem wird von einem Administrator für eine Lehrveranstaltung installiert. Danach werden von ihm die Seminarleiter und der Dozent mit Name und Passwort angelegt. Zur weiteren Konfiguration gehört das Festlegen des Namens der Lehrveranstaltung sowie Angabe von Ort und Zeit der Vorlesung. Somit ist das System fertig zur Nutzung.

Jeder Nutzer erreicht nun die Applikation über einen http-Client. Zu Beginn befindet er sich auf einer allgemeinen Login-Seite. Diese beinhaltet zwei Login-Bereiche, zwischen diesen einen Link für Neuanmeldungen von Studenten und unterhalb Links zu Universitätsseiten. Geplant sind Links zu dem allgemeinen Stundenplan, dem AkademischenJahr und dem Vorlesungsverzeichnis.

Der obere Login-Bereich ist für Studenten und studentische Kontrolleure, der untere für Dozenten und Seminarleiter. Durch Eingabe eines Loginnamens und Passwortes gelangt der Nutzer auf die Startseite in der sich Links das Navigationsmenü befindet und rechts die Aktionen gemäß seinen Rechten ermöglicht werden. In der Kopfleiste befindet sich ggf. die Möglichkeit die Rolle zu wechseln. Innerhalb des Menüs werden jedem Benutzer individuelle, für seine Rolle festgelegte Anwendungen zur Verfügung gestellt. (Diese können im einzelnen im Pflichtenheft nachgelesen werden.) Zu besseren Verständnis hier mal die Startseite des Seminarleiters.

Guten Tag!

Sie sind angemeldet beim ISIS für das Fach Softwaretechnik

Allgemein	Sie sind im Moment als Seminarleiter eingeloggt Sie haben nun die Möglichkeit mehrere Aktionen durchzuführen. Wählen sie die gewünschte Aktion im linken Menu !
Termine Forum Daten ändern	
Skripta	
herunterladen	
Übungsaufgaben	
einstellen löschen Statistik	
Übungsmaterial	
einstellen löschen	
Musterlösungen	
einstellen löschen freigeben	
Übungsgruppen	
anlegen löschen Gruppenliste	
Kontrolleure	
hinzufügen entfernen	
ISIS Sitzung	
beenden	

c) Version 1.0

Wie bereit erwähnt bietet die Version 1.0 einen Ausblick auf die meisten Anwendungsbereiche. Zur vorläufigen Benutzung wurden in einer XML-Datei die Nutzer statisch mit Name, Passwort und Rolle angegeben. Die Startseite besteht wie bereits im allgemeinen Teil beschrieben aus zwei Login-Bereichen und einem Link für die Neuansmeldung der Studenten. Diese Option ist allerdings noch nicht umgesetzt. Nach Eingabe von Name und Passwort eines Benutzers bietet sich folgendes Bild: Am oberen Fensterrand befindet sich die Begrüßung „Guten Tag! Sie sind angemeldet beim ISIS für das Fach Softwaretechnik „. Darunter befindet sich eine Statuszeile in der die aktuelle Rolle und gegebenenfalls noch weiter verfügbare Rollen angegeben sind. Die Version 1.0 erlaubt es bereit zwischen verschiedenen Rollen zu wechseln.

Darunter befindet sich das ein optisch durch die Hintergrundfarbe abgehobenes Fenster in welchem links eine Menüleiste und rechts die Aktion dargestellt ist. Nach dem Login ist rechts ein Standardtext zu sehen“ Sie sind im Moment als <Rolle> eingeloggt. Sie haben nun die Möglichkeit mehrere Aktionen durchzuführen. Wählen sie die gewünschte Aktion im linken Menu !“ Hier eine Auflistung der im Momentanen individuellen Menü ansteuerbaren Unterpunkte:

(Student, Seminarleiter, Dozent, Kontrolleur.)

Allgemein: Termine, Forum, Daten ändern

Skripte: hochladen, löschen, freigeben, herunterladen

Übungsaufgaben: einstellen, löschen, freigeben, Statistik

Übungsaufgaben: Aufgabenstellung, hochladen, heruntergeladen, Statistik

Übungsmaterial: einstellen, löschen, herunterladen

Gelöste Aufgaben: Aufgabenstellung, heruntergeladen, hochladen, Statistik

Musterlösungen: einstellen, löschen, freigeben, herunterladen

Übungsgruppen: anlegen löschen Gruppenliste

Übungsgruppe: anmelden, ändern

Klausur: eintragen, ändern

Klausuren: einschreiben, abmelden

Klausurergebnisse: ansetzen, ändern

Kontrolleure: hinzufügen, entfernen

ISIS Sitzung: beenden

Zu jedem dieser Punkte wird eine kurze Erklärung angegeben.

d) Version 2.0

Gegenüber der vorherigen Version sind jetzt einzelnen Anwendungen umgesetzt. Wie bereits erwähnt wurde bei der Implementierung der Klassen beim Seminarleiter begonnen.

Für diesen bieten sich jetzt nach dem Login folgende Möglichkeiten: Es wurde der Menüpunkt „Übungsmaterial“ mit einstellen und löschen, und „Skripte heruntergeladen“ komplett umgesetzt. In den Punkten „Übungsaufgaben“ und „Musterlösungen“ sind jeweils einstellen und löschen verfügbar. Außerdem ist es dem Seminarleiter jetzt möglich die Daten für eine studentische Hilfskraft in dem Menüpunkt „Kontrolleure anlegen“ in eine Eingabemaske einzuschreiben oder einen am System angemeldetem Seminarleiter als Kontrolleur anzugeben. Allgemein wurden alle Funktionen „herunterladen“ oder „hochladen“(Außer für die persönlichen Dateien von Studenten und Kontrolleuren) für die jeweiligen Nutzer zugänglich gemacht.

e) Version 3.0

In dieser Version ist zumindest die optische Logik der Übungsanmeldung beendet. Die Studenten können sich anmelden, der Seminarleiter legt seine Kontrolleure und die Übungsgruppen an. Danach der Student sich in einer Übungsgruppe einschreiben. Die Funktionalität sieht etwas anders aus. Es können sich die Seminarleiter anmelden und Kontrolleure anlegen. Neu in dieser Version ist die Möglichkeit des Seminarleiters seine Gruppen anzulegen und ihnen einen Kontrolleur zuzuweisen. Die Möglichkeit des Studenten der Neuansmeldung und Einschreibung in eine Gruppe existiert nur optisch, nicht funktionell. Dies soll in der nächsten Version dann ergänzt werden.

f) Version 4.0

Wie bereits erwähnt wurden einzelne Aktionen gestrichen, um das Programm sinnvoll lauffähig zu bekommen.

Hier erst mal eine Auflistung der möglichen Aktivitäten:

Studenten können sich neu anmelden (und es wird automatisch ein neuer Ordner zu diesem anlegt). Sie können ihre Gruppenanmeldung ändern; Skripte, Übungsaufgaben, Übungsmaterial und Musterlösungen herunterladen; ihre eigenen Lösungen uploaden und die kontrollierten downloaden. Sie können sich die Statistik zu den Punkten ansehen.

Seminarleiter können Übungsaufgaben, Übungsmaterial und Musterlösungen hochladen; Skripte herunterladen; Übungsgruppen mit Kontrolleur konfigurieren oder löschen, Kontrolleure anlegen oder löschen und Gruppenliste anzeigen lassen.

Kontrolleure können Skripte, Übungsaufgaben und Musterlösungen herunterladen; die bearbeiteten Aufgaben der Studenten downloaden und die korrigierten mit Punkten wieder uploaden.

Dozenten können Skripte und Übungsaufgaben hochladen oder löschen.

Übungsgruppen:

- der Seminarleiter meldet sich mit den ihm vom Administrator zugewiesenen Daten an
- In dem Menüpunkt „Kontrolleure anlegen“ kann er über die Eingabemaske Daten für eine studentische Hilfskraft eingeben (Name, Email-Adresse, Passwort, etc.) oder am System angemeldete Seminarleiter als Kontrolleure anlegen
- Mit dem Menüpunkt „Übungsgruppen anlegen“ können der Ort, die Zeit und die maximale Studentenzahl für jede von ihm geleitete Übungsgruppe angelegt werden. Außerdem wird hier jeder Gruppe ein Kontrolleur zugeteilt
- über den Menüpunkt „Übungsgruppe anmelden“ erlangt der angemeldete Student die Möglichkeit sich in die von ihm gewählte Gruppe einzutragen, solange diese nicht voll belegt ist
- seine Einschreibung wird in einer XML-Datei registriert
- es ist dem Studenten möglich zu späterer Zeit über das gleiche System die Übungsgruppe zu wechseln

Übungsaufgaben

- im System hat der Seminarleiter die Möglichkeit Kontrolleure anzulegen und ihnen jeweils Übungsgruppen zuzuteilen
- es werden neben Materialien zu den Seminaren auch die Übungsaufgabe (und die Musterlösungen) als PDF-Dokumente über das System veröffentlicht. Dies kann durch die Seminarleiter oder den Dozenten geschehen.
- Jeder angemeldete Nutzer, und so auch die Studenten, können nun die Übungsaufgaben herunterladen
- nachdem der Student die Aufgaben gelöst hat, kann er die Lösungen als PDF-Dokumente über die Funktion „Übungsaufgaben hochladen“ in seinen persönlichen Ordner transferieren
- der Kontrolleur erhält den Zugriff auf die Ordner der ihm zugeteilten Übungsgruppe und den darin eingeschriebenen Studenten. Er kann über „Übungsaufgaben herunterlade“ und eine spezielle Ordnerauswahl die Lösungen der Studenten herunterladen.
- nachdem er die Abgaben korrigiert und mit Kommentaren versehen hat kann er über eine Upload-Funktion diese mit Eingabe der erreichten Punkte in ein dafür vorgesehenes Fenster abschieken. Ist dies erfolgt hat der Kontrolleur keine Möglichkeit mehr die Punkte des Studenten zu ändern
- der Student hat nun über den Ordner und einen Menüpunkt die Möglichkeit die seine korrigierten Aufgaben herunterzuladen. Außerdem kann er sich über „Statistik“ seine aktuellen Punkte und die aller bisherigen Übungsreihen ansehen.
- Sollte es Unstimmigkeiten in der Bewertung geben, haben nur der Dozent und der Seminarleiter die Option die Punkte zu ändern

Es waren weitere Nutzungen vorgesehen, die noch ergänzt werden können, wie etwa die Benutzung eines Forums.

4. Paket- und Klassenstruktur

Hier befinden sich die Diagramme zur Verdeutlichung der Zusammenhänge der zu implementierenden Klassen.

Die Klasse User fungiert als Oberklasse der einzelnen Nutzer Seminarleiter, Kontrolleur, Studenten und Dozent. Für den Benutzer Administrator ist keine Klasse vorgesehen, da er nicht über das System direkt, sondern über eine Konsole agiert.

Die Klasse HttpFunktions dient dazu eine geforderte URL zu erzeugen. Dazu wird der Standart des Systems mit dem notwendigen Anhang versehen. Dies ist wichtig, da unterschiedliche Systeme unterschiedliche URL-Parameter haben und somit keine statische Angabe dieser möglich ist.

Es kann während der Implementierungsphase noch zu Änderungen der Klassen oder Methoden kommen. Aber das Grundgerüst wird identisch sein.

Version 1.0

Die Methoden in den Klassen Student, Seminarleader, Dozent und Controller werde zur Zeit nur als Dummys benutzt.

Durch die Unterteilung des Menüs in die Klassen Menu, AdvancedMenu, ControllerMenu, DozentMenu, SeminarleaderMenu, StudentMenu kann für jede Benutzerrolle ein individuelles Menü generiert werden. Die Klasse HttpFunktions ist vollständig implementiert und erfüllt genau die vorgesehenen Eigenschaften. (Übergabe einer URL)

Version 2.0

Zum Verständnis der Zusammenarbeit der Klassen befindet sich am Ende ein Sequenzdiagramm

Es wurden hauptsächlich die neue Klassen für die Arbeit mit XML erstellt. Diese waren zum Zeitpunkt des Erstellens des Designberichtes noch nicht zum Einbinden verfügbar.

Außerdem wurde die Klasse SessionCounter ergänzt.

In den Klassen Seminarleader, Dozent und AdvancedMenu wurden einzelne Methoden umgesetzt.

Version 4.0

Die neue Struktur der Klassen ist wie folgt:

Zur besseren Übersichtlichkeit wurden die Klassen in Pakete eingeteilt.

1. isis: User, Group, SessionCounter, Task - alles was zu den Daten gehört
- 1.2 -.display: Menu, MenuStudent, ..., HttpFunction - alles was zur Anzeige von ISIS wichtig ist
2. servlet: ServletStudent, ..., NewMember - alle Servlets
3. xml: XMLReaderLogin, ParserGroup, ParserUser, Serializer - alles was zu xml gehört

Das Servlet Login ruft eines der fünf Servlets ServletStudent, ... , NewMember auf. Außerdem legt es ein Objekt User an und übergibt dieses an die Session, die während der gesamte Zeit über besteht. Abhängig von der Rolle kann dann zwischen den Servlets gewechselt werden. Die wichtigsten Klassen sind die Klassen User und Group. Hier drin gibt es statische Methoden um z.B. alle User, nur die Seminarleiter oder die Gruppen zu laden. Zum lesen der XML-Dateien werden die Klassen XMLReaderLogin, ParserGroup und ParserUser genutzt, zum Schreiben die Klasse Serializer.

Spezielle Datenobjekte (User-, Groupinstanzen) können gespeichert (save()) und gelöscht (remove()) werden. Diese Klassen- und Objektmethoden finden hauptsächlich Anwendung in den Methoden der Klassen MenuStudent, MenuDozent,

Diese vier Menu-Klassen erben von der Klasse Menu Methoden zum anzeigen allgemeiner Informationen wie z.B. Termine, Downloads, usw. In der Klasse Menu stehen außerdem allgemeine Methoden zum Schreiben einer Html-Seite bereit, wie z.B. writeStart(), writeEnd(), was die Leserlichkeit der Servlets wesentlich erhöht.



