

Internet-Recherche

Studentenverwaltung (SOS)

SOS ist das Modul für die Studierendenverwaltung in der HIS-GX-Familie. Es ist integriert mit dem Prüfungsverwaltungsmodul POS, dem Zulassungsmodul ZUL und dem Lehrveranstaltungsmodul LSF. Es unterstützt alle Verwaltungsvorgänge der Studierendenverwaltung an deutschen Hochschulen und wird laufend an aktuelle Entwicklungen angepasst.

SOS ermöglicht:

1. eine zeitgemäße, rationelle und effektive Abwicklung von Massen- und Einzelvorgängen wie Einschreibung, Rückmeldung, Fachwechsel, Wiedervorlage, Beitragskontrolle, Exmatrikulation, Beurlaubung, Ausweis und Bescheinigungserstellung und Belegung
2. Registrierung von Vor-, Zwischen- und Abschlussprüfungen
3. Verwaltung von Chipkarten
4. Automatische Vergabe von E-Mail-Adressen, Login-Kennungen und Passwörtern für die Studierende
5. Zentrale Erfassung und Fortschreibung der von verschiedenen Stellen innerhalb und außerhalb der Hochschule benötigten Studenten- und Absolventendaten
6. Unterstützung von Studienkonten- und Studienguthaben-Modellen
7. Schnittstellen zu zentralen Nutzerverzeichnissen
8. Lieferung der Daten für die Amtliche Statistik (inkl. PL des Stat. Bundesamtes)
9. Webbasierte Selbstbedienungsfunktionen (QIS-SOS)

Weitere Information finden wir unter <http://www.his.de/Abt1/HISSOS>

Prüfungsverwaltung(POS)

POS ist das Modul für Prüfungsverwaltung in der HIS-GS-Familie, POS ist integriert mit dem Studentenverwaltungsmodul SOS, ISY und mit der Web-Anwendung für Lehre, Studium und Forschung LSF.

POS ermöglicht:

1. Prüfungsanmeldung
2. Kontrolle der Vorleistungen
3. Erfassung der Prüfungsergebnisse
4. Automatische Berechnung von Zwischen- und Gesamtnoten
5. Erstellung von Zeugnissen, Diploma Supplement
6. Erstellung von Bescheinigungen und Listen
7. Erstellung von Prüfungsstundenplänen
8. Verwaltung von Praktika und Praktikumsstellen
9. Aufbereitung und Auswertung der Prüfungsdaten
10. Lieferung der Prüfungsdaten an die amtliche Statistik
11. Erfassung und Fortschreibung der Prüfungsordnungsdaten, insbesondere auch mit Credit-Point-Systemen
12. Webbasierte Selbstbedienungsfunktionen (z.B. Anmeldung zu Prüfungen, Ergebnisübersichten, Notenverbuchung, Studienberatung) für Studierende

und Prüfer mit dem Zusatzmodul QIS-POS

Weitere Information finden wir unter <http://www.his.de/Abt1/HISSOS>

Prüfungsverwaltung FlexNow

FlexNow! ist ein konfigurierbares Prüfungsverwaltungssystem mit Selbstbedienungsfunktionen im Internet, das seit 1994 am Lehrstuhl Wirtschaftsinformatik, insbesondere Systementwicklung und Datenbankentwicklung, an der Universität Bamberg entwickelt wird. FlexNow! ist speziell für die Unterstützung studienbegleitender Prüfungssysteme (Credit-Point-Systeme) konzipiert, die besonders hohe Anforderungen an Flexibilität, zu verwaltes Mengengerüst und Lenkung des Prüfungsprozesses stellen. Daneben werden herkömmliche Blockprüfungssysteme unterstützt. Der fachliche Kern von FlexNow! sind die zu verwaltes Prüfungs- bzw. Studienordnungen, die unter Verwendung eines grafischen Werkzeugs modelliert werden. Aufbauend auf diesen Meta-Daten unterstützt FlexNow! den kompletten Prüfungsprozess: von der Erstellung des Prüfungsangebotes über die Prüfungsanmeldung via Internet, die Prüfungslogistik (Raum- und Zeitplanung), die Erfassung der Korrekturdaten durch die Prüfer bis hin zur Erstellung von Bescheiden und Zeugnissen.

Weitere Information finden wir unter

<http://www.uni-hamburg.de/fachbereicheeinrichtungen/pa2/flexnow.htm>

Lehrveranstaltungen FlexNow!

Lehrveranstaltungen werden in der FlexNow!-Modellhierarchie unmittelbar unterhalb der Fakultäten definiert, da sie im Gegensatz zu Fach- und Teilprüfungen nicht innerhalb der Studiengang- und Prüfungshierarchie, die durch Studien- und Prüfungsordnungen vorgegeben sind, strukturiert werden können. Der Zugriff auf die Lehrveranstaltungsdaten der Studenten (Anmeldung, Abmeldung,..) wird somit allen Sachbearbeitern eingeräumt, die für die Fakultät autorisiert sind, in der diese Lehrveranstaltung originär angeboten wird. Das schließt aber natürlich auch die Studenten aller anderen Fakultäten ein, sofern diese sich für diese Lehrveranstaltung angemeldet haben.

Prüfungsverwaltung (VirPa)

Das Virtuelle Prüfungsamt (VirPa) ist ein Webangebot des Prüfungsamtes an die Studierenden der Hochschule. Über diese Schnittstelle erhalten die Studierenden die Möglichkeit

1. Zum Abruf allgemeiner Informationen der Prüfungsämter
2. Zur Einsicht in Prüfungsergebnisse
3. Zur Anmeldung zu Prüfungen
4. Zur Abmeldung von Prüfungen
5. Zur Einsicht in Aushanglisten
6. Und zur Mitteilung einer Adressänderung.

Bei Entwicklung des Vitullen Prüfungsamtes wurde bewusst auf die Verwendung von Java und Javascript verzichtet. Das Virtulle Prüfungsamt ist in der Sprache PHP4 programmiert und stützt sich auf eine MySQL- Datenbank.

Weitere Information finden wir unter <http://www.campussource.de/software/virpa>

1.Glossar

- **Account:** ein Benutzerkonto (englisch user account) bezeichnet im IT-Bereich (IT = Informationstechnik) einen Zugang zu einem IT-System. Üblicherweise muss man sich vor dem Zugriff mit einem Benutzernamen und Passwort authentifizieren.
- **Apache:** bezeichnet einen HTTP-Server. Der Apache HTTP Server ist ein Produkt der Apache Software Foundation, und der meistverbreitete Webserver im Internet.
- **Client:** (engl. "Kunde") ist eine Anwendung, die in einem Netzwerk den Dienst eines Servers in Anspruch nimmt.
- **Ereignisbehandlung:** Ein Ereignis (Event) zeichnet sich dadurch aus, dass es auftreten kann aber nicht muss. Eventhandling ist somit die Behandlung eines solchen Events.
- **Framework** (engl. Rahmenwerk, Fachwerk) ist ein Begriff aus der Softwaretechnik und wird insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungs-Ansätzen verwendet.
- **HTML:** Die Hypertext Markup Language (HTML) ist ein Dokumentenformat zur Auszeichnung von Hypertext im World Wide Web und wurde 1989 von Tim Berners-Lee am CERN in Genf festgelegt. Sie basiert dabei auf der Metasprache SGML, die zur Definition von Auszeichnungssprachen verwendet wird. HTML ist also eine Auszeichnungssprache zur Beschreibung von Informationen in Hypertexten.
- **Http:** (Hypertext Transfer Protocol), kurz http, ist ein Datenaustausch-Protokoll zur Übertragung von Daten im World Wide Web. Primär wird es im Rahmen des World Wide Web zur Übertragung von Webseiten verwendet (Web-Browser greifen fast ausschließlich mit diesem Protokoll auf Web-Server zu).
- **Instanz:** (vom lateinischen *instans* = gegenwärtig) bezeichnet in der objektorientierten Programmierung) eine Ausprägung eines Objektes.
- **Jakarta Projekt:** ist ein Projekt der Apache Software Foundation. Es beinhaltet, entwickelt und unterstützt Anwendungen, die mit Hilfe von Java geschrieben werden. Es enthält viele Unterprojekte(z.B. Tomcat) Jakarta Homepage (<http://jakarta.apache.org/>)
- **Jakarta Tomcat:** Tomcat ist ein Servletcontainer, der im Jakarta-Projekt der Apache Software Foundation entwickelt wurde. Tomcat ist eine Implementierung der Servlet und der JavaServer Pages (JSP) Spezifikation von Sun Microsystems.
- **JavaBeans-Modell:** JavaBeans werden in der Softwareentwicklung als Container zur Datenübertragung verwendet. Java Beans sind Software-Komponenten. Ursprünglich waren Java-Beans für "visuelles Programmieren" gedacht, also ein Baustein-System, in dem man Programme aus Komponenten zusammenbauen kann, ohne selber programmieren zu müssen, ähnlich wie etwa bei Schaltkreis-Entwurfs-Systemen.
- **JSP:** Java Server Pages, abgekürzt JSP, ist eine von Sun Microsystems entwickelte Technologie, die im Wesentlichen zur einfachen dynamischen Erzeugung von HTML- und XML-Ausgaben eines Webservers dient.
- **MVC:** Model-View-Controller, kurz MVC. Beim MVC-Modell handelt es sich um ein Entwurfsmuster zur Trennung bestimmter Programmeigenschaften. Idee: Trennung des Programms in die drei Einheiten Datenmodell (Model), Präsentation (View) und Programmsteuerung (Controller). Ziel des Modells ist ein flexibles Programmdesign (Änderung oder Erweiterung einfacher, Wiederverwendbarkeit einzelner Komponenten) Reduzierung der Komplexität (sorgt für Übersicht/Ordnung bei großen Anwendungen) Rollenverteilung durch Trennung.

- **Open Source:** Der Ausdruck Open Source (engl.: open source = offene Quelle) steht einerseits dafür, dass der Quelltext eines Programms frei erhältlich ist.
- **Parser:** (englisch to parse = analysieren bzw. lateinisch pars = Teil -> deutsch "Zerteiler") ist ein Programm, das entscheidet, ob eine Eingabe zur Sprache einer bestimmten Grammatik gehört.
- **Resource Bundle:** (engl. Bundle = Bündel) Bündelung von verschiedenen Produkten, die sich gegenseitig ergänzen.
- **Server:** Der Begriff Server hat in der Informationstechnik zwei Bedeutungen: Zum einen wird ein Dienst-Programm als Server bezeichnet, welches - i. a. über ein Netzwerk - Ressourcen zur Verfügung stellt, zum anderen die Hardware, auf der dieses Dienst-Programm läuft.
- **Servlet:** Als Servlet bezeichnet man im Rahmen der Java 2 Plattform Enterprise Edition (J2EE) ein Java-Objekt, an das ein Webserver Anfragen seiner Clients delegiert, um die Antwort an den Client zu erzeugen.
- **Servlet-Architekturen:** Sie dienen zur einheitlichen Behandlung von Webservices. Servlets erstellen einen serverseitigen Code für Web-basierte Clients und machen sie damit nutzbar, indem sie Web-Entwicklern einen einfachen Mechanismus zur Erweiterung der Funktionalität eines Web-Servers zur Verfügung stellen.
- **Session:** in der Datenverarbeitung eine logische Verbindung zwischen zwei adressierbaren Einheiten in einem Leitungsnetz, um Daten auszutauschen. In der Client/Server-Technik ist es die zeitweise bestehende Verbindung eines Clients mit einem Server.
- **SGML:** (engl. Standard Generalized Markup Language) ist eine Metasprache, mit deren Hilfe man verschiedene Auszeichnungssprachen (engl. *markup languages*) für Dokumente definieren kann.
- **Struts:** Struts ist ein Open Source-Framework für die Präsentationsschicht von Java-Web-Anwendungen. Es ist eines der bekanntesten Jakarta-Projekte und beschleunigt die Entwicklung von Web- Applikationen wesentlich, indem es einen standardisierten Prozess zur Verarbeitung von HTTP-Anfragen vornimmt. Dabei bedient es sich standardisierter Technologien wie JavaServlets, JavaBeans, Resource Bundles und XML sowie verschiedener Jakarta Commons Pakete.
- **UML:** Unified Modeling Language, kurz UML, ist eine standardisierte Beschreibungssprache für Strukturen und Abläufe in objektorientierten Programmsystemen.
- **WAP:** Wireless Application Protocol, kurz WAP. Protokoll für die mobile Telekommunikation.
- **Web-Architekturen:** bestehen aus einem Web-Server und dem Web-Client und basieren auf Hypertext-Technik. Sie ermöglichen den Zugriff auf andere Internetdienste.
- **Webservice:** ist ein Dienst, der mit Hilfe von XML auf der Basis von Internet-Netzwerkprotokollen erbracht wird. Web Services sind nicht für menschliche Benutzer gedacht, sondern für Softwaresysteme, die automatisiert Daten austauschen und/oder Funktionen auf entfernten Rechnern aufrufen.
- **XML:** Extensible Markup Language, kurz XML, ist eine Programmiersprache zur Implementation von Netzwerkdiensten. Sie ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur. Dabei definiert XML die Regeln für den Aufbau solcher Dokumente.

2.Konzepte

- Ereignisbehandlung: (Eventhandling) wird von Java eingesetzt, damit Instanzen miteinander kommunizieren können. JavaBeans machen starken Gebrauch von dem Konzept, was primär daran liegt, dass es sich bei JavaBeans um visuelle Komponente-Web-Architekturen Beschreibungen handelt, die im Allgemeinen mit Events gekoppelt werden. Möchte ein Bean oder jedes andere Objekt von einem Event Kenntnis erlangen, muss es sich bei der Eventquelle mit einem sogenannten Listener registrieren. Die Quelle pflegt also einen Zuhörerstamm. Tritt nun ein bestimmtes Event ein, wird das Objekt alle entsprechenden Zuhörer davon mit Aufruf einer Methode in Kenntnis setzen. Das genau diese Methoden, den Zuhörern innerhalb des Listener zur Verfügung steht, ist es dem Zuhörer möglich individuell auf das Event zu reagieren und es ggf. auch zu ignorieren.
- Java-Beans-Modell:
Das JavaBeans Application Programming Interface ermöglicht es Komponenten Software in Java zu

implementieren. Komponenten sind selbstenthaltene, wiederverwendbare Softwareeinheiten die visuell als Komponenten zusammengesetzt werden können. So können dann neue Komponenten, Applets, Programme oder auch Servlets in visuellen Programmierwerkzeugen entstehen. (aus JavaBeans.pdf url ???).

- MVC-Konzept (Model -View -Control): Das MVC-Konzept erlaubt damit eine klare Schichtentrennung zwischen den Anwendungsdaten, den Sichten auf die Anwendungsdaten und der Benutzungsschnittstelle. Es besteht aus den drei Klassen Model, View und Controller.
 - Model-Objekte stellen die Verbindung zur fachlichen Logik her, deren Ausführung entweder auf dem Client oder auf dem Server stattfindet. Außerdem ist es für die Kernfunktionalität und das Anwendungsobjekt zuständig.
 - Controller-Objekte nehmen die Eingaben in Empfang und entscheiden, welche fachliche Logik ausgeführt werden soll.
 - View-Objekte schließlich sind für die Präsentation am Bildschirm zuständig.
- Servlet-Architekturen: Sie dienen zur einheitlichen Behandlung von Webservices. Servlets erstellen einen serverseitigen Code für Web-basierte Clients und machen sie damit nutzbar, indem sie Web-Entwicklern einen einfachen Mechanismus zur Erweiterung der Funktionalität eines Web-Servers zur Verfügung stellen.
- Struts: Struts ist ein Open Source-Framework für die Präsentationsschicht von Java-Web-Anwendungen. Es ist eines der bekanntesten Jakarta-Projekte und beschleunigt die Entwicklung von Web- Applikationen wesentlich, indem es einen standardisierten Prozess zur Verarbeitung von HTTP-Anfragen vornimmt. Dabei bedient es sich standardisierter Technologien wie JavaServlets, JavaBeans, Resource Bundles und XML sowie verschiedener Jakarta Commons Pakete.
- Web-Architekturen: bestehen aus 2 Komponenten: dem Web-Server und dem Web-Client und basieren auf Hypertext-Technik. Außerdem ermöglichen sie den Zugriff auf die anderen Internetdienste.

3. Beschreibung der Rahmenapplikation

3.1. Leistungsmerkmale

Der Übungsmanager ist ein Tool, welches die Abwicklung der Organisation einer Vorlesung erleichtert. Das Programm beruht dabei einer JavaServlet Technologie(Struts), wodurch das Programm plattformunabhängig ist.

Jede Instanz des Übungsmanagers ist dabei für eine Vorlesung zuständig. Es treten 5 verschiedenen Akteuergruppen auf, die den Übungsmanager nutzen können. Diese sind der Administrator, der Dozent, die Tutoren, unbekannte Nutzer und die Studenten. Für jede dieser Benutzergruppen stehen verschiedene Funktionen zur Verfügung. Der Administrator hat die Möglichkeiten der Konfiguration und Datensicherung für die Vorlesung. Die Erstellung der Übungsgruppen, die Veröffentlichung der Materialien(Aufgaben usw.) und die Moderation des Forums. Ein Tutor kann Lösungen runterladen und die Korrektur Hochladen. Die Studenten können sich, nachdem sie angemeldet sind in eine Übungsgruppe einschreiben, Aufgaben runterladen, Lösungen hochladen, den Punktestand abfragen, sowie sich für die Klausur einschreiben. Außerdem haben die Tutoren und die Studenten die Möglichkeit auf das Forum zuzugreifen. Desweiteren gibt es noch die Möglichkeit Daten von Usern zwischen einzelnen Instanzen herzuschieben, damit z.B. ein User der sich bei 2 Instanzen anmeldet nicht alle persönlichen Einstellungen zwei mal vornehmen muss.

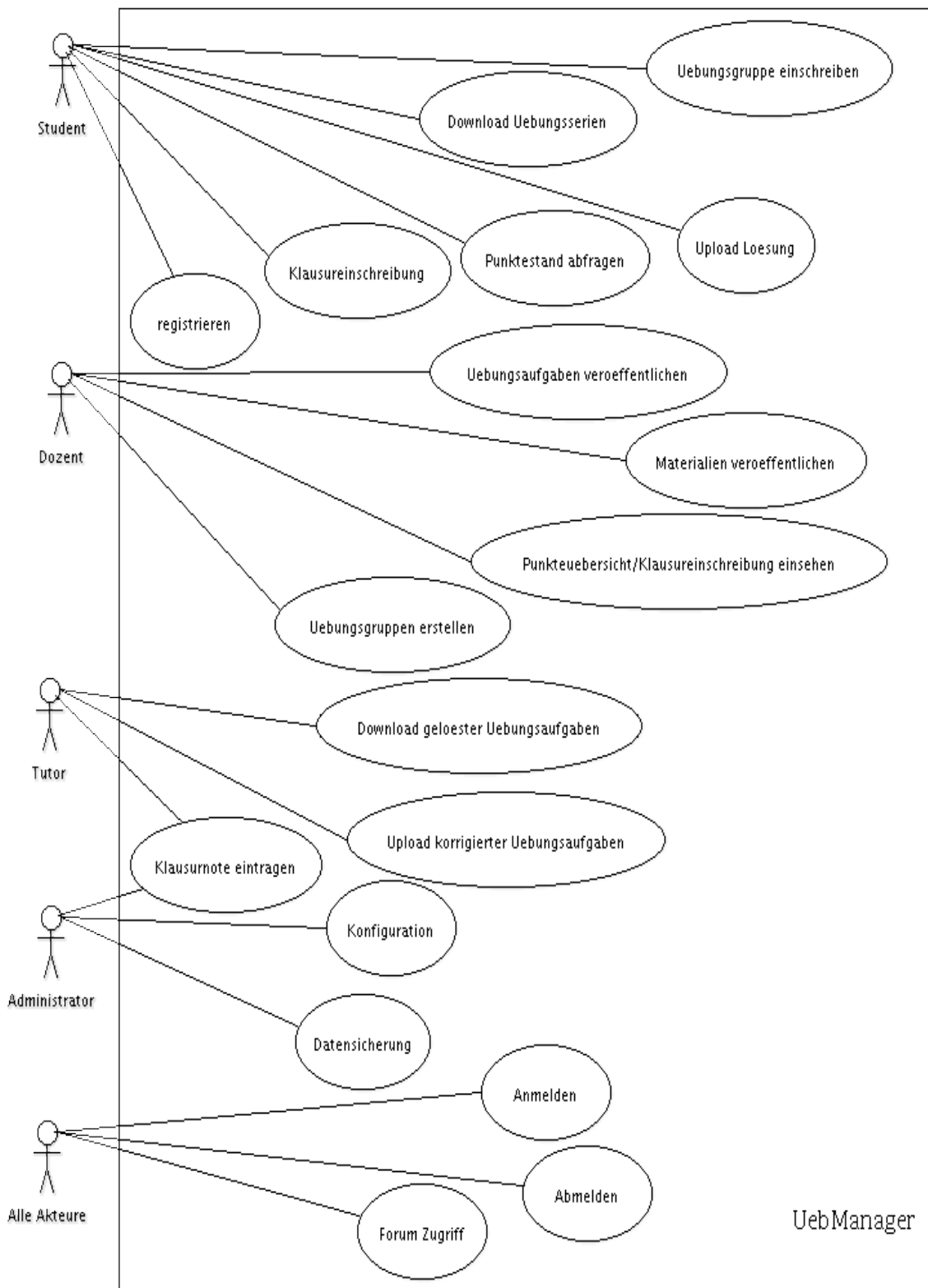
Es existiert auch noch ein Webservice, mit dem man feststellen kann wieviele Instanzen(also quasie Vorlesungen) gerade aktiv sind.

3.2. Beschreibung von Anwendungsfällen

Anwendungsfall:	Anmelden
Akteur:	Student, Dozent, Tutor, Administrator
Beschreibung:	Akteure können sich über ein Login mit Usernamen und Passwort anmelden
Ergebnis:	Zugang zum Passwortgeschützten Bereich

Anwendungsfall:	Abmelden
Akteur:	Student, Dozent, Tutor, Administrator

Beschreibung:	Akteure können den Passwortgeschützten Bereich verlassen
Ergebnis:	Passwortgeschützten Bereich verlassen
Anwendungsfall:	registrieren
Akteur:	Student
Beschreibung:	Akteur kann sich registrieren um ein Passwort zu erhalten
Ergebnis:	Benutzername mit zugehörigen Passwort
Anwendungsfall:	Übungsgruppe einschreiben
Akteur:	Student
Beschreibung:	Der Student trägt sich in eine freie Übungsgruppe ein
Ergebnis:	Student erhält eine Bestätigung in welche Gruppe er eingeschrieben ist oder eine Fehlermeldung
Anwendungsfall:	Download von Übungsserien
Akteur:	Student
Beschreibung:	der Student lädt sich zu lösende Übungsaufgaben herunter
Ergebnis:	der Student kann sich die Übungsaufgaben ansehen
Anwendungsfall:	Upload Lösung
Akteur:	Student
Beschreibung:	der Student lädt die Lösung seiner Übungsaufgaben hoch
Ergebnis:	die Lösung des Studenten ist für die Korrektur verfügbar
Anwendungsfall:	Klausuranmeldung
Akteur:	Student
Beschreibung:	Student kann sich mit seinen Daten zu einer verfügbaren Klausur anmelden
Ergebnis:	der Student erhält eine Bestätigung seiner Anmeldung oder eine Fehlermeldung.
Anwendungsfall:	Klausurnote eintragen
Aktuer:	Dozent, Tutor
Beschreibung:	Dozent oder Tutor können Klausurnoten eintragen(veröffentlichen)
Ergebnis:	Student kann seine Klausurnote sehen
Anwendungsfall:	Punktstand abfragen
Akteur:	Student
Beschreibung:	Student kann seine in den Übungsaufgaben erreichten Punkte einsehen
Ergebnis:	Auskunft über bisher erreichte Punkte
Anwendungsfall:	Übungsaufgaben veröffentlichen
Akteur:	Dozent
Beschreibung:	der Dozent veröffentlicht zu lösende Übungsaufgaben
Ergebnis:	Übungsaufgaben können von den Studenten heruntergeladen werden
Anwendungsfall:	Materialien veröffentlichen
Akteur:	Dozent, Tutor
Beschreibung:	der Dozent oder der Tutor können begleitende Materialien zu den Übungen veröffentlichen
Ergebnis:	Materialien stehen den Studenten zum download bereit
Anwendungsfall:	Übungsgruppen erstellen
Akteur:	Dozent, Tutor
Beschreibung:	der Dozent oder der Tutor erstellen die einzelnen Übungsgruppen
Ergebnis:	Übungsgruppen stehen zur Einschreibung für die Studenten bereit



Anwendungsfall:

Download gelöster Übungsaufgaben

Akteur:

Tutor

Beschreibung:

der Tutor lädt Übungsaufgaben der Studenten zum korrigieren herunter

Ergebnis:

der Tutor erhält gelöste Übungsaufgaben zum korrigieren

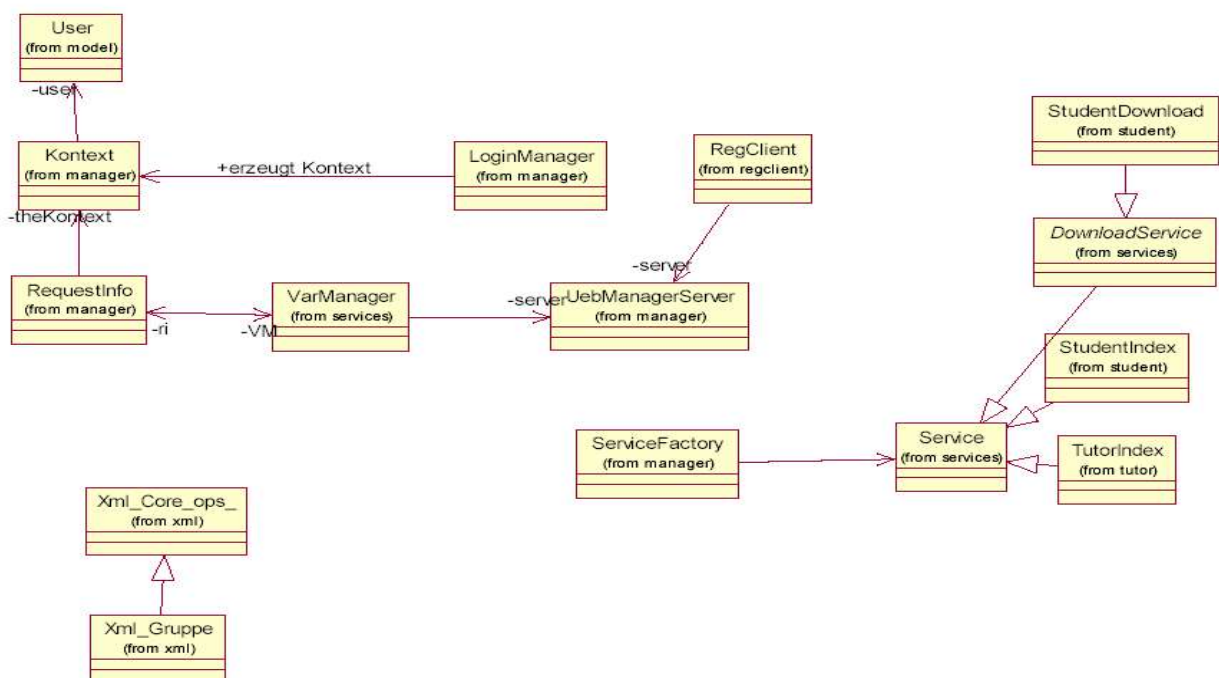
Anwendungsfall: Upload korrigierter Übungsaufgaben
 Akteur: Tutor
 Beschreibung: der Tutor lädt die korrigierten Übungsaufgaben sowie Punktvergabe hoch
 Ergebnis: der Student kann seine korrigierten Übungsaufgaben herunterladen

Anwendungsfall: Konfiguration
 Akteur: Administrator
 Beschreibung: der Administrator nimmt Einstellungen an der Applikation vor
 Ergebnis: lauffähige Applikation

Anwendungsfall: Forum Zugriff
 Akteur: Student, Dozent, Tutor, Administrator
 Beschreibung: je nach Berechtigung können Beiträge erstellt, editiert sowie gelesen werden
 Ergebnis: Fragen und Hilfen zum Übungsbetrieb

3.3. äußerlich sichtbarer Aspekte der inneren Logik der Applikation

Eine Instanz des UebManagerServer dient zur Verwaltung einer Veranstaltung. Zu jedem UebManagerServer gehört ein LoginManager, der prüft ob sich ein registrierter User anmelden möchte. Falls es einen korrekten Loginversuch gab, wird ein Kontext für den entsprechenden User erstellt. Dieser Kontext hält die Daten für eine Session. Der User hat dann die Möglichkeit verschiedene Services in Anspruch zu nehmen. Dabei stehen nur solche Services zur Verfügung, die der User entsprechend seiner Rolle (Student, Tutor, Dozent, Admin) zur Verfügung stehen. Die ServiceFactory hat die Aufgabe den richtigen Service eines Requestes an den UebManager zu liefern. Falls der verlangte Service mit einer Datei arbeitet, ist das Interface DownloadService zuständig, ansonsten wird der Service über Service abgehandelt. Für Services die mit XML-Dateien arbeiten dient Xml_Core_ops als Schnittstelle. Über diese Schnittstelle ist es z. B. möglich eine Gruppe zu bearbeiten (Mitglieder entfernen, hinzufügen usw.). Die Klasse User dient zum Austausch von Userinfos zwischen verschiedenen Instanzen des UebManager. Dadurch wird realisiert, daß es einem User möglich ist seine Accountdaten von einer Instanz zu einer andern Instanz zu übertragen. Desweiteren gibt es noch die Klasse RegClient, welches wohl dafür zuständig ist verschiedene Instanzen des UebManager zu ermitteln.



18.04.2005

tr-2

Florian Drechsler, Swen Schipp, Konstantin Thierbach, Martin Vogt, Maik Müller, Michael Kunze, Qingli
Liu, Lei Chang