

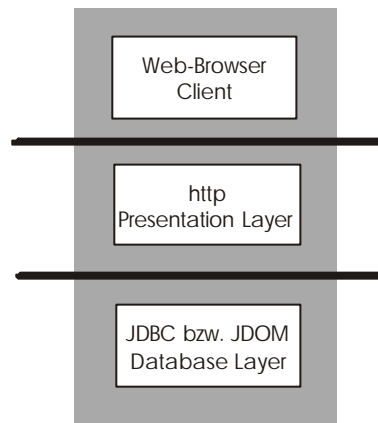
Recherchebericht

- Aufgabe 1 -

1. Allgemeines

Aufgrund der stetigen Verbreitung und der steigenden bis überwältigenden Akzeptanz des Internets und seiner Möglichkeiten unter den Studenten der Universitäten, steigt auch die Nachfrage nach internetbasierten Lösungen für den Studienalltag. Außerdem bietet die persistente Datenhaltung in Datenbanken und die Publikationsmöglichkeit Internet den Mitarbeitern der Universität und den studentischen Hilfskräften eine erhebliche Steigerung der Effizienz ihrer Arbeit durch die Abnahme von Routinetätigkeiten.

Die geplante Software ist im Groben wie folgt aufgebaut :



Themenspezifische Terminologie

Bei der von uns geplanten Software handelt es sich im weitesten Sinne um eine Lernplattform. Als **working definition** schlage ich folgendes vor : Eine Lernplattform stellt die Softwaretechnische Infrastruktur zur elektronischen Unterstützung von Lern- und Lehrprozessen bereit. Wobei sich Lehrprozesse in unserem Projekt vor allem auf publikatorische Prozesse beschränken. Zusätzlich übernimmt die Software die Verwaltung des Seminarbetriebes.

Die wichtigsten Anforderungen an unsere Lernplattform kann man aus den Rechercheergebnissen ableiten und wie folgt stichpunktartig zusammenfassen.

1. Funktionale Aspekte :

- Verwaltung der Studenten (Belegung der Übungsgruppen, Punkvergabe)
- Publikation von Lehr- bzw. Aufgabenmaterial
- Benutzerverwaltung mit Rechtevergabe

2. Nicht-Funktionale Aspekte :

- Kompatibilität zu Standardwebbrowser
- Benutzerfreundlichkeit

Für unser Projekt geforderte Funktionalitäten :

1. Studenten sollen sich über das Internet in Übungsgruppen einschreiben können
2. Bereits eingeschriebene Studenten sollen ihren Punktestand erfragen können
3. Studenten sollen Aufgaben per Internet uploaden können
4. Allen Studenten soll ein Studenten-Account eingerichtet werden können
5. Mitarbeiter und SHK sollen Listendateien mit Kontrollergebnissen uploaden können
6. Bereits angemeldete Benutzer sollen sich einloggen können
7. Kontrolle und Auskunft über aktuellen Stand der Punktevergabe

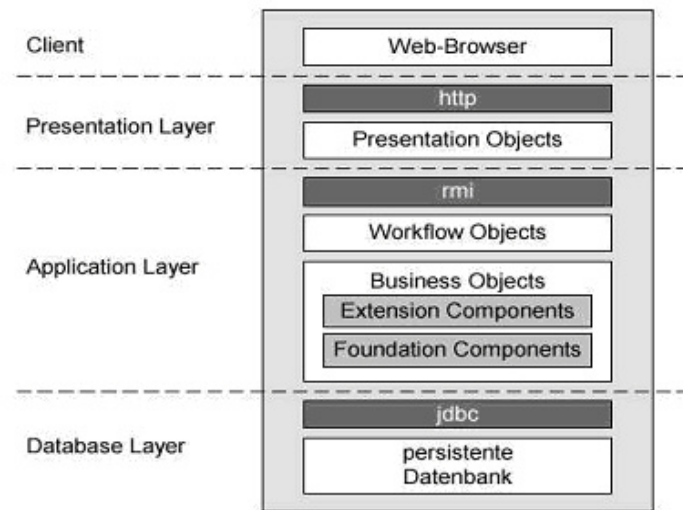
2. Übersicht über themenrelevante Software

Webbasierte Informationssysteme (aus dem riesigen Fundus hier nur eine kleine Auswahl)

- **EXAM** :Einsatzgebiete E-Learning, Prüfungen, Umfragen, Architektur Client/Server Quelle <http://myexams.net>

Und nun speziell im Lehrbetrieb genutzte webbasierte Informationssysteme :

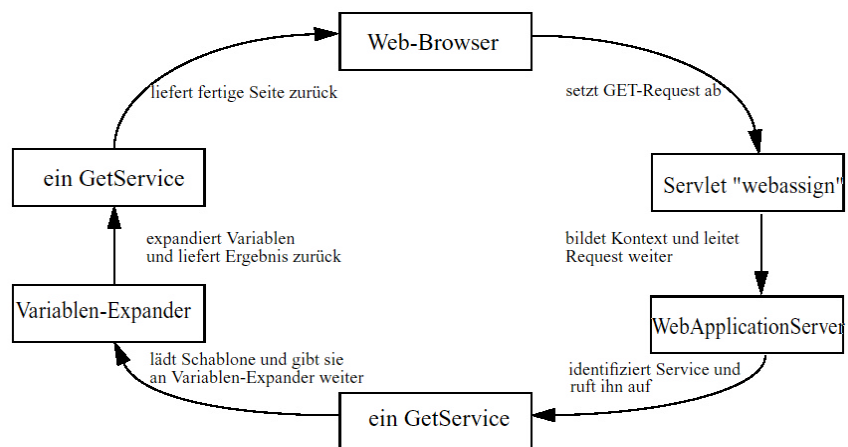
- **S.U.S.H.I.** == Universal Student Help Interface Einsatzgebiet ist vor allem die Organisation der Kursregistratur, Einteilung in Arbeitsgruppen und Bereitstellung von Informationen. Bei Sushi handelt es sich um ein Open Source Projekt der technischen Universität Berlin. Zum Einsatz gekommen ist hier eine auf JavaServlets aufsetzende Technologie, mit einer Datenbank im Hintergrund.
Quelle <http://fugu.sourceforge.net>
- **SOVA** heißt für meine Begriffe „Seminar Organisation für Verteilte Anwendungen“. Dabei handelt es sich um ein "Produkt" aus dem Hause RNVS.uni-leipzig.de. Die Architektur ist ähnlich, allerdings wird hier alles über XML geregelt. Das einzige mir bekannte Einsatzgebiet ist die Seminar- und Übungsseriesverwaltung des Faches Betriebssysteme an der Universität Leipzig.
- **WebAssign** realisiert einen umfassenden universitären Übungsbetrieb. Da WebAssign eine webbasierte Software ist, genügt ein Webbrowser, um Übungen zu erstellen, zu bearbeiten, zu korrigieren und zu administrieren. Allerdings ist es mehr zum Erstellen von Fragestellungen, deren Publikation im Netz und darauffolgender Korrektur der Lösungen von den Studenten gedacht. Dies impliziert aber entweder eine erzwungenen Grammatik für die Eingaben oder aber Multiple Choice Antworten. Dies könnte in einigen Fächern durchaus unbefriedigend bei entsprechender Komplexität des zu testenden Lehrinhaltes sein. Datenhaltung erfolgt über eine MySQL Datenbank.
- **OpenUSS** ist ebenfalls ein webbasiertes datenbanknutzendes Informationssystem für die Lehre. Es verfügt prinzipiell über eine Vierschichtenarchitektur. Einer Client-, Presentation-, Applikation und einer DatabaseLayer(JDBC). Sämtlich funktionellen Applikationen sind in Extension- oder Foundation Components unterteilt. Dadurch ist einfache Erweiterung eines bereits bestehenden Systems um neue Funktionalitäten durch Hinzunahme einer weiteren Komponente möglich.



Architektur von OpenUSS
 Quelle : <http://openuss.sourceforge.net/openuss/>

3. Genauere Beschreibung einer Applikation (WebAssign)

Grundsätzliche Arbeitsweise von WebAssign:



Funktionsweise von Diensten und Variablen
 Quelle: <http://www.campussource.de/dl/pub/webassign>

Grundsätzlich werden alle Funktionen, die WebAssign anbietet über das zentrale WebAssignServlet ausgeführt.

Anwendungsfall:	Veranstalter oder Kurs erzeugen/löschen
Akteur:	Administrator
Beschreibung:	Erzeugen neuer Einträge in HIS Datenbank und Anmeldung eventueller CORBA Lösungsserver
Ergebnis:	Veranstalter oder Kurs am Informationssystem angeschlossen

Anwendungsfall:	Aufgabenstellung erzeugen
Akteur:	Mitarbeiter der Universität (Betreuer)
Beschreibung:	Aufgaben bzw. Teilaufgaben generieren/analysieren Lösungen, Hinweise, Korrekturseiten oder Lösungsquittungen erzeugen
Ergebnis:	Neue Aufgaben, Lösungen oder Hinweise stehen über WebBrowser allen Usern zur Verfügung

Anwendungsfall:	Betreuer ändern / löschen
Akteur:	Mi Mitarbeiter der Universität (Betreuer)
Beschreibung:	Änderung der Kurszuordnung des Betreuers (weiteren Kurs hinzufügen / einen Kurs abgeben)
Ergebnis:	Eintrag des zugehörigen Betreuers in der Datenbank wurde verändert

Anwendungsfall:	Einsendung gelöster Aufgaben
Akteur:	Student
Beschreibung:	HTML-Datei hochladen als Lösung speichern je nach Kurseinstellung dem CORBA Lösungsserver überreichen und gegebenen Falls Quittung über Lösung ausstellen
Ergebnis:	Quittung über Upload und gegebenenfalls Bewertung des CORBA-Lösungsservers an Studenten schicken

Anwendungsfall:	eine E-Mail an den WebAssign Administrator schicken (z.B. zwecks Problemen mit WebAssign)
Akteur:	Betreuer, Student
Beschreibung:	eine Mail wird direkt an den WebAssign Administrator geschickt
Ergebnis:	es wurde eine Mail an den Administrator verschickt

Anwendungsfall:	Klausuranmeldung online
Akteur:	Student
Beschreibung:	Hochladen einer HTML Datei mit den Daten zur Klausuranmeldung (Klausur, Matrikelnummer) Speichern des neuen Datensatzes in DBS
Ergebnis:	der Student erhält eine Bestätigung oder Fehlermeldung seiner Klausuranmeldung Siehe nachfolgende Sequenzdiagramme.

Die folgenden zwei Sequenzdiagramme beschreiben die Anmeldung eines Studenten ans System und einen Lösungsupload mit Korrektur:

