

# Recherchebericht – Gruppe Ue11

## Allgemeines

1. Einsatzumfeld: Universitäten, Fachhochschulen und alle weiteren Schulformen
2. Grundsätzliche Logik für prototypische Applikation:

Es handelt sich um ein typisches Verwaltungssystem, bei dem zu einer Menge von Objekten (Studenten) eine Menge von Daten (Noten) gespeichert werden. Ziel ist es, die Übersicht und Verwaltbarkeit zu verbessern. Dahinter steht recht einfach die strukturierte Speicherung der Daten in einer Datenbank, sowie die einfache Bedienung durch den Benutzer durch einen Browser o.ä. Nun kann die Software praktische Möglichkeiten bieten die Daten einzugeben, sie zu verändern oder sie strukturiert anzuzeigen. Als Erweiterung kann man sich vorstellen, unterschiedlichen Benutzern unterschiedliche Sichten auf den Datenraum zu bieten, und ihnen somit eine Möglichkeit bieten, einzelnen Benutzern auch private Daten zugänglich zu machen.

3. Themenspezifische Terminologie:

Administrator:	Benutzer mit Administrationsrecht
HTML:	Hyper text markup language, kurz html, ist die Beschreibungssprache für Dokumentinhalte im world wide web
Java-Applet:	(applet, Verniedlichung von application, engl. Anwendung) Wird ein in Java erstelltes Programm genannt. Durch den Java-Compiler wurde es in Bytecode übersetzt, der dann durch die JVM ausgeführt wird.
Java-Servlet:	Eine Möglichkeit, serverseitig Programme ausführen zu lassen. Damit kann z. B. auf Formulareingaben des Benutzers reagiert und die Antwort-Page generiert werden. Vereinfacht sind es Java-Applets, die vom Internet-Server ausgeführt werden. Da sie auch über die JVM laufen, sind sie Plattform-unabhängig. Das Java Servlet Development Kit ist Bestandteil der JDK ab Version 1.2.
JDK:	Java Development Kit, wird eine von Sun Microsystems bereitgestellte Entwicklungsumgebung für das Schreiben, Testen und Debuggen von Java-Applets genannt.
JVM:	Java Virtual Machine, ist das Programm, dass für die Ausführung von Java-Applets notwendig ist. Es ist für jede Plattform, jedes Betriebssystem und jeden Browser eine eigene Version notwendig. Die Umgebung bietet keinen Zugriff auf Routinen des Betriebssystems, weswegen auch von einer Sandbox gesprochen wird.

Übersicht über themenrelevante Applikationen:

### Klausurnotenverwaltungssystem

3.1 Quelle: [http://swt.wi-inf.uni-essen.de/~kws\\_java/](http://swt.wi-inf.uni-essen.de/~kws_java/)

3.2 Leistungsparameter:

3.2.1.1 Die Daten (Benutzer, Angaben zu den Studenten, Klausuren und Termine, Noten) werden in einer Datenbank abgelegt.

3.2.1.2 Die Verwaltung (Administration) der Benutzer und Studentendaten erfolgt über HTML Ausgabe.

3.2.1.3 Autorisierte Personen können Klausuren anlegen und dazu Termine freischalten. Auch die Notenvergabe erfolgt über HTML durch autorisierte Personen.

3.2.1.4 Die Studenten haben dann die Möglichkeit über HTML oder WML ihre Klausurergebnisse abzufragen (Auf Wunsch alle bisherigen oder alle neuen).

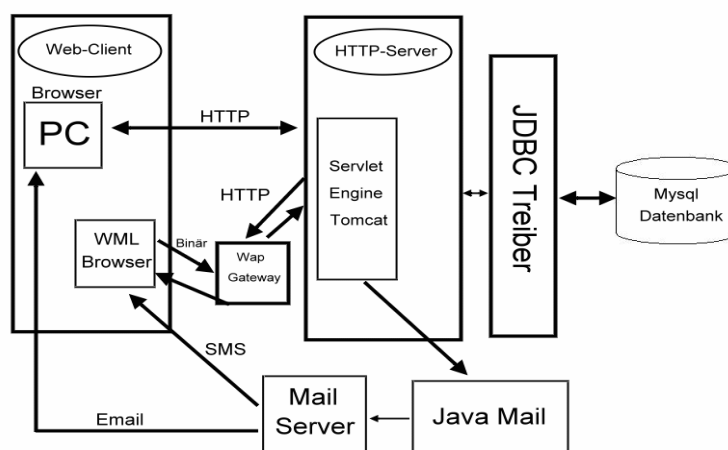
3.2.1.5 Für die Studenten besteht außerdem die Möglichkeit über ein HTML Interface das System so zu konfigurieren, dass sie bei Bekanntgabe der Noten per SMS und/oder e-Mail informiert werden.

3.3 Architektur:

Da die Notenverwaltung eine grundlegende Funktion des von uns geplanten Projektes darstellt, kann die ihr zu Grunde liegende Struktur als Vorlage für unser Projekt ohne Probleme benutzt werden.

Prinzipiell geht bei diesem Projekt um eine Client-Server-Anwendung, mit allen daraus resultierenden Vor-(und Nach-)teilen. Die Benutzer (mit unterschiedlichen Rollen) können einfach über die Benutzung eines Browsers auf einem Client-Rechner mit dem Server in Verbindung treten.

Dabei werden an das System gestellte Anfragen per http übermittelt und an die Servlet-Engine weitergeleitet. In Java wird dann der weitere Zugriff auf entweder die DB (JDBC) oder andere angeschlossene und benötigte Systeme, wie etwas des Mailserver (über JavaMail) zur Benachrichtigung bei bestimmten Ereignissen, geregelt. Die Schnittstelle zwischen der Servlet-Engine wird durch einen speziellen JDBC Treiber gebildet. Die Antworten des Systems gelangen über die Servlet Engine zurück zum Client zu der jeweiligen genutzten Schnittstelle. Der Client kann Informationen durch Email und/ oder SMS vom System erhalten. Die Servlet Engine kontaktiert mit dem Auftrag das Java Mail Framework. Dieses bereitet die zu Übertragenden Informationen konform zum systeminternen Mailserver auf. Daraufhin leitet er dann die Nachricht via Email oder SMS zu dem vom Client gewünschten Ziel weiter.



3.4 Einsatzgebiete: Universitäten, Fachhochschulen und alle weiteren Schulformen

## 2.2 Gruppenterminkalender

### 3.5 Konzeption:

Ziel ist die Vereinbarung und Verwaltung von Einzel- und Gruppenterminen über ein Web-basiertes Informationssystem. Die Funktionalität soll die Eingabe, Verwaltung und Abfrage sowohl über ein HTML-als auch über ein WML-Interface umfassen. Das System soll die Benutzer unterstützen indem es z.B. einen Termin für ein Gruppentreffen vorschlagen kann (indem es z.B. alle Termine der einer Gruppe zugehörigen Personen durchsucht und einen Termin findet, an dem keine Person einen anderen Termin hat). Außerdem soll das System einen Benutzer auf Wunsch per Email/SMS an einen Termin erinnern. Über ein HTML-Interface soll das System administrierbar sein.

### 3.6 Quelle: [http://swt.wi-inf.uni-essen.de/~gtk\\_java/](http://swt.wi-inf.uni-essen.de/~gtk_java/)

### 3.7 Leistungsparameter: keine angegeben

### 3.8 Architektur:

Das System basiert auf dem Client-Server-Modell. Auf der Clientseite sollen die Schnittstellen zu den Benutzern erstellt werden, die den Abruf der HTML- und WML-Service über den Browser ermöglichen sollen. Die Benutzer sollen die verschiedenen Gruppen frei auswählen, persönliche Daten und Termine auf den Servern speichern und eigene Termine verwalten können. Es sollen Kommunikationsmöglichkeiten zwischen allen Benutzern einer Gruppe bestehen. Die Serverseite ist in zwei Schichten unterteilt: Java-Servlet und Datenbank. Das Java-Servlet übernimmt die Anfragen vom Client (Benutzer) und greift mit Hilfe von JDBC Technologie auf die Datenbank zu. Die Ergebnisse sollen vom Java-Servlet in Form von HTML/WML wieder an den Client (Benutzer) geschickt werden.

### 3.9 Einsatzgebiete: zum Austausch von Informationen, die von mehreren Mitarbeitern eingesehen werden sollen, d.h. überall, wo webgestützte Teamarbeit erforderlich ist, z.B. Softwareentwicklung

## Genauere Beschreibung einer Applikation (Klausurnotenverwaltungssystem)

Das Online-Notenverwaltungssystem dient zur Verwaltung von Daten über studentische Leistungen über das Internet.

### 3.1 Verbale Beschreibung der Leistungsmerkmale

- 3.9.1 Anmeldung: nur Sachbearbeiter können einen neuen Studenten anlegen, danach kann sich jeder Nutzer mit Loginname und Passwort anmelden
- 3.9.2 Mit Hilfe von Formularen können Veranstaltungen / Prüfungen / Leistungen erstellt und gespeichert werden
- 3.9.3 nur Administratoren angegebener Nutzer mit Loginname und Passwort an kann neuen Benutzer anlegen und Benutzerpasswort ändern
  - 3.9.4 Student kann (nachdem er angelegt ist) seine Daten (PW, e-Mail, SMS, Benachrichtigungswunsch) selber verwalten.
  - 3.9.5 Ein autorisierter Benutzer kann Klausuren und dazugehörige Termine eingeben und dann Matrikelnummern und Noten eingeben. (evtl. Masseneingabe ermöglichen)
  - 3.9.6 Autorisierte Benutzer können Noten nachträglich ändern (zu jeder Note wird letzter Änderungsdatum und Person gespeichert)
  - 3.9.7 Änderungen und Eintragungen von Noten sind nur von autorisierten Benutzern erlaubt. Zu Beginn ist nur der berechtigt, der die Klausur angelegt hat.
  - 3.9.8 Studenten können alle aktuellen oder alle bisher erbrachten Leistungen abfragen
  - 3.9.9 Auf Wunsch benachrichtigt das System Studenten, wenn für diese neue Ergebnisse eingetragen wurden.

### 3.2 Beschreibung von Anwendungsfällen

Anwendungsfall: Student erstellen  
Akteur: Sachbearbeiter  
Beschreibung: Es wird ein Formular angezeigt, in dem alle für das Studium relevante Daten eingetragen werden (Name, Vorname, Mat.nr., Studienbeginn, Vordiplom bestanden/nicht bestanden, E-Mail, Handy...)  
Ergebnis: Student in Datenbank aufgenommen

Anwendungsfall: Student bearbeiten  
Akteur: Sachbearbeiter  
Beschreibung: Durch die Angabe von Mat.Nr. oder Name kann der gesuchte Datensatz gefunden werden. Danach werden alle Felder mit den aktuellen Daten und einer Änderungsoption angezeigt  
Ergebnis: Studentendaten geändert / bearbeitet

Anwendungsfall: Auflistung der aktuellen Noten  
Akteur: Student  
Beschreibung: Unter der Angabe eines Zeitraumes in Monaten können die Prüfungsergebnisse angezeigt werden, die in diesen Zeitraum (von dem aktuellen Datum ausgehend in die Vergangenheit gerechnet) fallen  
Ergebnis: Anzeigen der Noten im angegebenen Zeitraum

Anwendungsfall: Auflistung des gesamten Kontos  
Akteur: Student  
Beschreibung: Dem Studenten werden alle Leistungen angezeigt, die er während seines Studiums erbracht hat. Dabei soll ihm auch eine Übersicht über die erbrachten, sowie die noch fehlenden Kreditpunkte angezeigt werden.  
Ergebnis: Anzeigen der gesamten Leistungen

Anwendungsfall: Benutzer einrichten  
Akteur: Administrator  
Beschreibung: Es wird ein Formular zur Verfügung gestellt, auf dem der eindeutige Schlüssel, Name, UserID, Passwort und der Status des neuen Benutzers eingetragen wird.  
Ergebnis: Neuer Benutzer eingerichtet

Anwendungsfall: Studenten für Prüfung anmelden  
Akteur: Dozent  
Beschreibung: Zu der jeweiligen Veranstaltung wird ein Prüfungstermin angelegt unter dem alle Studenten, die sich für diesen Termin angemeldet haben hinterlegt werden.  
Ergebnis: Prüfungstermin angelegt

### 3.3 Aktivitätsdiagramm

