

Deckblatt

Recherchebericht der Gruppe 15 zum Thema: Übungsbetrieb
(Homepage der Ueb15: <http://pcai003.informatik.uni-leipzig.de/~ueb15>)

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

Seiten (inklusive Deckblatt) : 11

Zusammenstellung der einzelnen Ausarbeitungen durch Markus Scholz.

Bearbeitet von Frank Jühling und Markus Scholz

Teil 1 : Allgemeines

Es soll ein webgestütztes Informationssystem zur Begleitung des Übungsbetriebes an einer Hochschule erstellt werden. Dazu sollen alle Studenten und Veranstaltungen erfasst werden, die für die Gestaltung des Übungsbetriebes relevant sind. Das System soll die Studenten auf dem aktuellem Stand der Übungsserien halten, dazu soll ein öffentlicher Web-Bereich mit up- und download-Bereichen eingerichtet werden. Außerdem sollen auch Tutoren und Seminarleiter gesonderte Zugriffe erhalten.

Zur Nutzung des System ist es nötig, dass sich die Studenten registrieren, d.h. der Student hat übermittelt dem System seine Daten (Name, Vorname, Matrikel, usw.) und kann sich dann in das System einloggen. Ist dies erfolgt, so stehen ihm die folgenden Optionen zur Verfügung. Der Student kann sich in Übungsgruppen einschreiben, zu Klausuren anmelden, seine Lösungen uploaden und seine Punktestände sowie Klausurergebnisse abfragen. Außerdem kann er Material zu den Übungen/Vorlesungen herunterladen, Kontakt zum Seminarleiter herstellen, seinen Übungskalender betrachten und ändern sowie entsprechende Diskussionsforen benutzen.

Tutoren, die vom Seminarleiter an das System angemeldet werden müssen, haben die Möglichkeit, gelöste und hochgeladene Lösungen der Studenten zu kontrollieren und Punktelisten ihrer Übungsgruppen abzufragen, zu erstellen und zu ändern. Weiterhin sind sie verantwortlich für die Foren ihrer Übungsgruppen.

Seminarleiter werden vom Administrator an das System angemeldet. Sie besitzen die gleichen Rechte wie Tutoren, können aber zusätzlich alle Übungspunktelisten abschließen sowie Klausurpunktelisten erstellen und ändern, außerdem sind sie neben dem Administrator die einzigen Benutzer, die die Anmeldedaten aller registrierter Studenten einsehen und ausdrucken können. Die Seminarleiter verwalten außerdem den Veranstaltungskalender.

Glossarentwurf :

Administrator:

Einzelne Person die alle Rechte über das System besitzt, es einrichtet und im laufenden Betrieb verwaltet. Der Administrator meldet Seminarleiter im System an. Kann aber außerdem die Studentendaten ändern.

Tutoren:

Sind den Seminarleitern unterstellt und kontrollieren die hochgeladenen Übungen der Studenten.

Seminarleiter:

Sind hauptverantwortlich für den Übungsbetrieb, organisieren und kontrollieren diesen.

webgestütztes Informationssystem:

Stellt eine Schnittstelle zwischen Studenten, Tutoren und Seminarleitern dar. Alle beteiligten Personen können in ihrer Rolle über das Internet auf diese Schnittstelle zugreifen.

Übungsbetrieb:

Zu Vorlesungen werden begleitende Übungen angeboten. Hier werden Übungsaufgaben gestellt, die dann von den beteiligten Studenten gelöst werden müssen.

Übungsgruppe:

Eine Gruppe von Studenten, welche gemeinsam ein Seminar besuchen.

Seminar:

Versammlung einer Übungsgruppe zur Besprechung von Vorlesungsthemen und Übungsaufgaben.

..
..

Teil 2 : Übersicht über themenrelevante Applikationen

(1) **„Web-basierte Übungen für große Vorlesungen“
von Frank Lübeck und Max Neunhöffer, Aachen**

Quelle: <http://www.fachgruppe-computeralgebra.de/CAR/CAR31/node11.html>

Laufendes System : <http://www.math.rwth-aachen.de/LAI12002>

Zusammenfassung

Einsatzgebiet : Zum Management des Übungsbetrieb für größere Studentengruppen an Fachhochschulen.

Leistungsparameter : Das System ist in der Lage Studentengruppen von mehr als 1000 Studenten ohne Probleme zu verwalten.

Und bietet folgende Leistungen:

- Anmeldung zu den Übungen; danach sind für alle individuellen Services Matrikelnummer und Passwort anzugeben.
- Änderung des Passwortes.
- Abholung von individuellen Übungsblättern; diese werden auf Wunsch als (HTML-)Webseite oder als PDF-Datei (zum Ausdrucken) generiert.
- Abgabe der Multiple-Choice Aufgaben; die Lösungen werden in die HTML-Version der Übungsblätter direkt eingetragen und dann per Mausklick abgeschickt. Beim erneuten Aufrufen des Blattes wird die bisherige Lösung angezeigt. Bis zum Abgabeschluss kann diese geändert und neu abgeschickt werden. Nach Abgabeschluss wird die Bewertung angezeigt und eine Änderung ist nicht mehr möglich.
- Eingabemöglichkeiten für Punkte in schriftlichen Aufgaben und Klausuren durch die Tutoren.
- Ergebnisanzeige für Hausaufgabenpunkte und Klausuren.

Architektur : Das Produkt besteht aus einem Server-Programm, das für die Studenten- und Übungsblatt-verwaltung zuständig ist. Zusätzlich existieren verschiedene Hilfsprogramme und ein Skript, das aus den Webseiten heraus aufgerufen wird und über das Datennetz mit der Server-Software kommuniziert. Zur Umsetzung des Projekts wurde hier die einfache und flexible Programmiersprache Python verwendet. Aufgrund seiner plattformunabhängigen Architektur kann das Produkt sowohl auf Unix basierten Systemen als auch auf Windows Systemen zum Einsatz kommen.

Bearbeitet von Frank Jühling und Markus Scholz

(2) **"Plattform 2000" der FU Hagen** (LVU Hagen, Lernraum Virtuelle Universität)

Quelle : <http://www.fernuni-hagen.de/LVU/public/plattform2003.html>

Laufendes System : <https://vu.fernuni-hagen.de> (usr: gast / pwd: gast)

Einsatzgebiet : Virtuelle Lehre an der FernUni, d.h. das System stellt dem angemeldeten Studenten eine komplette Studien- und Lernumgebung zur Verfügung und gibt den entsprechenden Betreuern die Werkzeuge an die Hand um die Studenten so gut wie möglich zu verwalten.

Leistungsparameter : Leistungsspektrum des LVU Hagen
(<http://www.fernuni-hagen.de/LVU/public/LVUBroschuere.pdf>)

Für Studenten finden sich Funktionen zu folgenden Punkten :	Für Betreuer finden sich Funktionen zu folgenden Punkten :
Persönliche Daten und Arbeitsumgebung Übungen News und Kommunikation Lehrveranstaltung Studierenden durch das ICE (Internet Communication Environment) Download	Kurskonfiguration Benutzerverwaltung Betreuung

Architektur : Das System basiert auf einer Client/Server- Architektur, die sich auf gängige Standards stützt. Es kommen Java-Techniken (Skripts, Servlets, Applets, usw.) sowie professionelle DBS zum Einsatz. Außerdem werden mehrere Server für Kommunikations- und Kooperationsfunktionen genutzt. Grundlegend lässt sich das System in 2 Teile aufspalten: ein sicheren Teil für funktionale Bereiche, u.a. in denen die Authentifizierung des Nutzers erforderlich ist und einen unsicheren Teil der z.B. öffentliche Informationen bereithält.

Anmerkung : Das System der FU Hagen ist ein außerordentlich komplexes und sehr leistungsfähiges System (zur Zeit werden ca. 26.000 Studenten erfolgreich verwaltet). Die oben genannten Daten sind deshalb nur als ein sehr grober Überblick über die Funktionalität des System zu sehen.

(3) **System zum Übungsbetrieb der Vorlesung Betriebssysteme von Prof. K. Irmscher**

Quelle : http://www.informatik.uni-leipzig.de/rnvs/lehre/praktikum_bs/index.html

Einsatzgebiet : Das System wurde für den Übungsbetrieb der Vorlesung Betriebssysteme an der Universität Leipzig konzipiert und wird auch dort eingesetzt.

Leistungsparameter : siehe „Teil 3 : Genauere Beschreibung einer Applikation (1)“

Architektur : web-basiertes System auf HTML- und CGI- Grundlage

Bearbeitet von Frank Jühling und Markus Scholz

(4) **Prototyp eines System zum Übungsbetrieb von Dipl.-Inf. Th. Riechert**

Quelle: <http://pcai003.informatik.uni-leipzig.de/rieichert/prototyp>

Einsatzgebiet : System zum Verwalten des Übungsbetriebes an Hochschulen

Leistungsparameter : siehe „**Teil 4 : Genauere Beschreibung einer Applikation (2)**“

Architektur : web-basiertes System auf HTML- und Java- Grundlage

Bearbeitet durch Stefan Nordt und Frank Nowak

Teil 3 : Genauere Beschreibung einer Applikation (1)

3.1

L := Link (→ ist ein Verweis auf genauere Beschreibung)

Übungssystem zur Vorlesung BetriebsSysteme:

Quelle : http://www.informatik.uni-leipzig.de/rnvs/lehre/praktikum_bs/index.html

[Startseite]

Service für Studenten:

- L: Angebot (Vorlesungen des Semesters) → (1)
- L: Registrierung → (2)
- L: Anmeldung für Sitzung → (3)

Service für Kontrolleure und Betreuer:

- L: Anmeldung K.
- L: Anmeldung Mitarbeiter (Betreuer)

(1) Auflistung der Lehrveranstaltungen und Download dazu

L: Startseite

(2) L: angebotene Vorlesung

L: Anmeldung für Sitzung →(3)

Möglichkeit der Dateneingabe (mit Erläuterung) und anschließende Registrierung
(Mat.-Nr., Studservkennung, Passwort, Nachname, Vorname, Studientyp)

Reset-taste

L: Startseite

(3) Eingabe Mat.-Nr. , Passwort und weiter zur Sitzungsumgebung →(A)

L: Startseite

(A) [Hauptseite]

Allgemeiner Service

L: Angebot →(1)

L: Aktuelles (Infos durch Übungsleiter)

Spezieller Service

Passwort ändern →(a)

Einschreibung für Übung →(b)

Verwaltung der Übungsaufgaben

Aufgabendownload → (c)

Lösung upload → (d)

Status → (e)

Sitzungsabmeldungstaste

(a) Daten Eingabe: Mat.-Nr. , Altes Passwort. Neues Passwort, Wiederholung
Update und Reset Taste

L: Hauptseite

Bearbeitet durch Stefan Nordt und Frank Nowak

(b) Auflistung der bereits eingeschriebenen Vorlesungen

Auflistung der möglichen Vorlesungen

L: Hauptseite

(c) Übersicht über die Übungsblätter von eingeschriebenen Vorlesungen

L: Hauptseite

(d) Auflistung der angemeldeten Vorlesungen

L: Übungsaufgaben

L: Hauptseite

(e) Auflistung der Übungsserien zu den angemeldeten Vorlesungen mit Angabe über Upload-Datum sowie erreichter Punktzahl

L: Hauptseite

3.2

Anwendungsfall: Vorlesungsübersicht

[evtl. Akteur]: Alle

Beschreibung: Akteur möchte Informationen über Lehrveranstaltungen des Lehrstuhls

Ergebnis: Anzeige aller angebotenen Vorlesungen

Anwendungsfall: Registrierung

[evtl. Akteur]: Studenten der Übungsveranstaltung

Beschreibung: Student registriert sich im System durch Angabe seiner Daten (siehe 3.1)

Ergebnis: Erhalt eines Accounts

Anwendungsfall: Anmeldung

[evtl. Akteur]: registrierter Student / Mitarbeiter / Kontrolleur

Beschreibung: Anmeldung zu einer Sitzung durch Angabe von Matrikelnummer/Login und Passwort

Ergebnis: Zugang zu der Account-Seite, die weitere spezifische Aktionen ermöglicht

Anwendungsfall: Abrufen aktueller Informationen

[evtl. Akteur]: registrierter Student

Beschreibung: Student sucht Informationen zu aktuellen Veranstaltungen

Ergebnis: Informationserhalt

Anwendungsfall: Passwortänderung

[evtl. Akteur]: registrierter Student (/ Mitarbeiter / Kontrolleur)

Beschreibung: Änderung des Passwortes, durch wiederholte Eingabe des neuen Passwortes

Ergebnis: geändertes Passwort

Anwendungsfall: Einschreibung in Übung

[evtl. Akteur]: registrierter Student

Beschreibung: Student schreibt sich in angebotene Übungen ein

Ergebnis: Eintrag in Liste für Übung

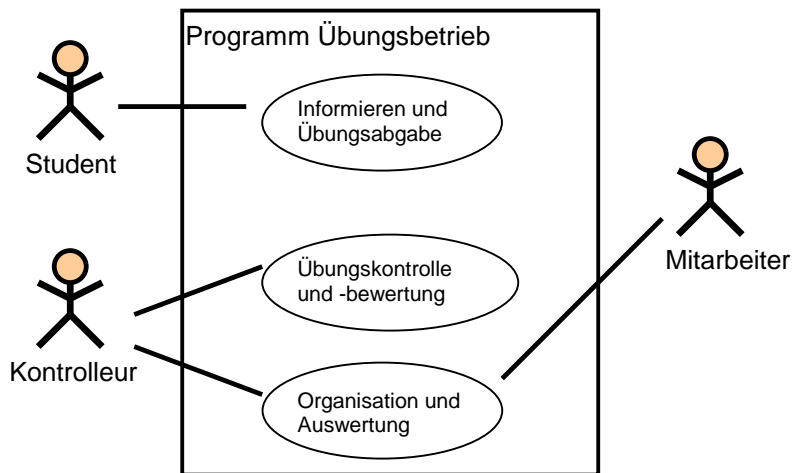
Bearbeitet durch Stefan Nordt und Frank Nowak

Anwendungsfall: Übungsaufgaben - Download
[evtl. Akteur]: registrierter Student
Beschreibung: Anklicken der gewünschten Serie
Ergebnis: Download

Anwendungsfall: Lösung uploaden
[evtl. Akteur]: registrierter Student
Beschreibung: Aussuchen der Datei in der die Lösung enthalten ist und upload
Ergebnis: Erscheinen des UploadDatums im Statusbereich

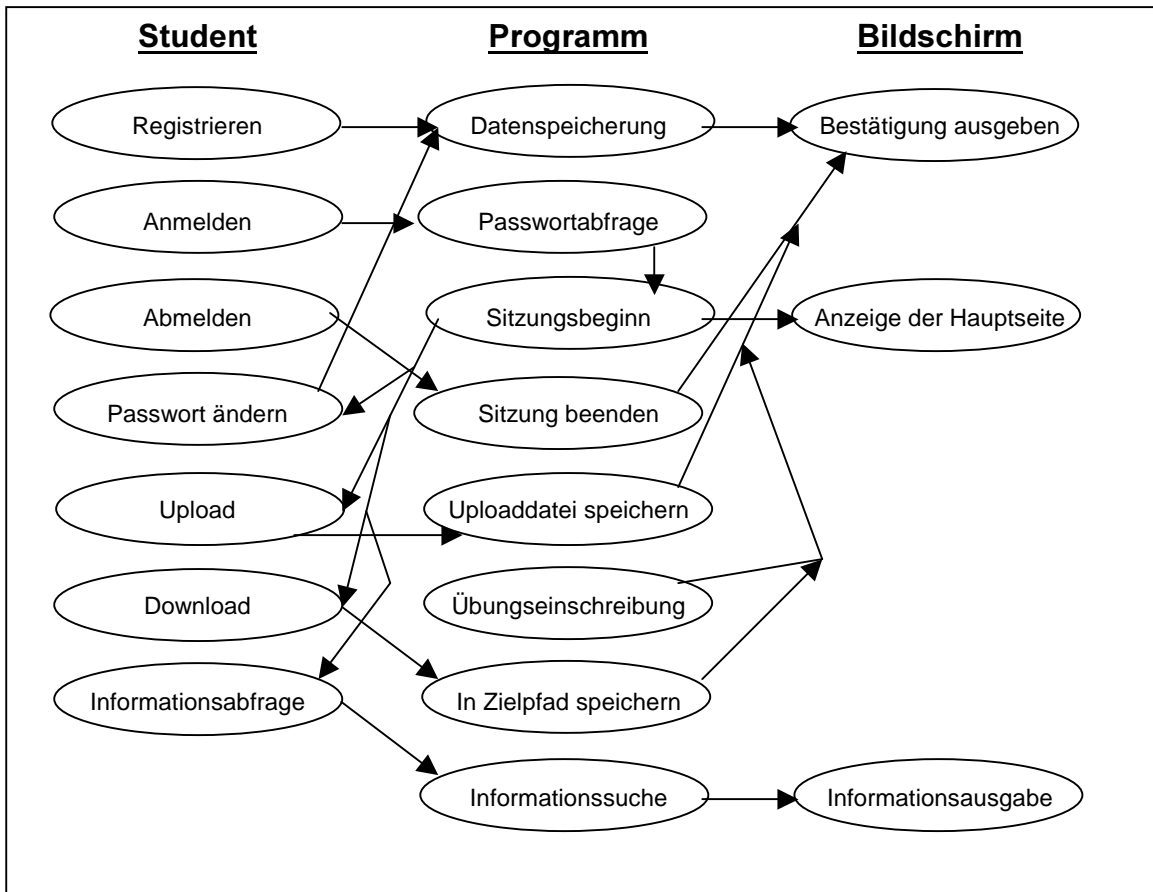
Anwendungsfall: Status abfragen
[evtl. Akteur]: registrierter Student
Beschreibung: Anzeige der spezifischen Daten des Studenten
Ergebnis: Anzeige der aktuellen Daten

3.3 grobes Geschäftsprozessdiagramm auf der obersten Ebene



Bearbeitet durch Stefan Nordt und Frank Nowak

Aktivitätsdiagramm



Teil 4 : Genauere Beschreibung einer Applikation (2)

Applikation: „Übungen zu Vorlesung Softwaretechnik“ (Thomas Riechert)

Quelle: <http://pcai003.informatik.uni-leipzig.de/rieichert/prototyp>

4.1

Die Prototyp-Applikation zum Übungsbetrieb 2003/2004 gibt Aufschluss über die eigentlich Applikation, die später folgen soll. Noch ohne Datenbank arbeitend zeigt sich auf den mit HTML generiert Seiten zum Übungsbetrieb, was die Zielstellung der Applikation sein soll und wird. Zum einen gibt es Öffentliche Seiten, die man ohne weitere Umschweife erreichen kann, zum anderen gibt es einen gesicherten Bereich, der durch persönlichen Login zu erreichen ist.

Auf den öffentlichen Seiten kann sich für den Übungsbetrieb einschreiben, sich das Vorlesungsscript ansehen und downloaden, genauso wie das dazu gehörige Übungsmaterial. Ein Kalender ist ebenfalls ansteuerbar, wie auch die öffentlichen Einträge des Diskussionsforum (*reviewed*).

Auf den Seiten, die man nach dem Login erreicht, wird nach Student, Tutor und Seminarleiter unterschieden. Der Student kann mit seinem Login sich seine aktuellen Punktestand zu den Übungen anschauen, sowie sich für die Klausur anmelden (nach bestandenem Übungsschein). Erreichbar ist ebenfalls ein persönlicher Übungskalender und das Diskussionsforum. Über ein Kontaktformular ist der Seminarleiter zu erreichen.

Die Seite des Tutors ist folgendermaßen inhaltlich strukturiert: Es erlaubt dem Tutor die Punktelisten der Übungsaufgaben und Klausuren zu erstellen und zu verwalten. Es erfolgt die Eingabe der Punkte nach Korrektur. Ebenfalls angegeben werden Name, Matrikelnummer und Aufgabe. Der Tutor ist ebenfalls für die Administration des Diskussionsforums zuständig.

Der Seminarleiter kann ebenfalls auf seinen Seiten Punkteliste der Übungen und Klausuren erstellen und verwalten, aber nur er kann diese Listen auch abschließen, damit keine nachträglichen Änderungen vorgenommen werden können. Dem Seminarleiter obliegt nicht nur, wie beim Tutor, die Administration des Diskussionsforums, sondern auch die des Kalenders, der Vorlesungsscriptseite und Übungsseite.

4.2

Anwendungsfall	Vorlesungsscript downloaden
[evtl. Akteur]	[Seminarleiter, Student, Tutor]
Beschreibung	Seminarleiter lädt Script auf vorbestimmte Seite hoch (mit Hilfe von ftp) schreibt einen dazugehörigen Eintrag in Quellcode (bei HTML-Applikation)
Ergebnis	Tutor und Student können sich Script downloaden

Anwendungsfall	Übungsmaterialien downloaden
[evtl. Akteur]	[Seminarleiter, Student, Tutor]
Beschreibung	Seminarleiter lädt Daten auf vorbestimmte Seite hoch (beispielsweise mit Hilfe von ftp) schreibt einen dazugehörigen Eintrag in Quellcode (bei HTML-Applikation)
Ergebnis	Tutor und Student können sich Material downloaden
Anwendungsfall	Kalender
[evtl. Akteur]	[Seminarleiter, Student, Tutor]
Beschreibung	Seminarleiter lädt Kalender, in welcher form auch immer (pdf, jpg, doc,...) auf vorbestimmte Seite hoch (mit Hilfe von ftp) schreibt einen dazugehörigen Eintrag in Quellcode (bei HTML-Applikation)
Ergebnis	Tutor und Student können sich Kalender ansehen

Bearbeitet durch Stefan Schmidt und Philipp Metz.

Anwendungsfall	Einschreibung
[evtl. Akteur]	[Student]
Beschreibung	Student geht auf Seite von Einschreibung und trägt seine persönlichen Daten ein (Name, Vorname, Matrikelnummer, Seminargruppenwunsch, E-Mail) Eventuell wird dem Studenten dann automatisch ein Bestätigungsmail zu
Ergebnis	Student ist eine der Seminargruppen eingeschrieben

Anwendungsfall	Diskussionsforum
[evtl. Akteur]	[Seminarleiter, Student, Tutor]
Beschreibung	Seminarleiter und Tutor betreuen Diskussionsforum Das Forum selber unterliegt einer Programmiermaske, in die Einträge eingeschrieben werden können und die automatisch ausgegeben werden. Einträge, die den vermerkt privat haben werden automatisch erkannt und für den öffentlichen Bereich herausgefiltert
Ergebnis	Einträgen können gelesen und geschrieben werden

Anwendungsfall	Punktlisten der Übungen und Klausur
[evtl. Akteur]	[Seminarleiter, Student, Tutor]
Beschreibung	Seminarleiter und Tutor tragen nach Korrektur der Aufgaben die Punkte in die dazugehörigen Listen. Ebenfalls Matrikelnummer und Name. Der Seminarleiter schließt die Listen ab, damit keine nachträgliche Veränderrungen vorgenommen werden können.
Ergebnis	Der Student kann dann seine Punkte aktuell einsehen

Anwendungsfall	Klausureinschreibung
[evtl. Akteur]	[Student]
Beschreibung	Der Student trägt sich in eine vorbereitete Maske ein. Dort gibt er seine persönlich Angaben ein (Matrikelnummer, Name, Vorname, Fachsemester,..) Eventuell erfolgt abermals der Versand einer E-Mail, zur Bestätigung seiner Einschreibung
Ergebnis	Student ist für Klausur eingeschrieben