



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Koray Kizilarlan

Entwicklung eines webbasierten  
Informationssystems zur Aufgabenverteilung

Koray Kizilarlan  
Entwicklung eines webbasierten  
Informationssystems zur Aufgabenverteilung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Cristoph Klauck  
Zweitgutachter : Prof. Dr. Olaf Zukunft

Abgegeben am 14. Juni 2012

**Koray Kizilarlan**

**Thema der Bachelorarbeit**

Entwicklung eines webbasierten Informationssystems zur Aufgabenverteilung

**Stichworte**

ASP.NET, Groupware, Ajax

**Kurzzusammenfassung**

Die Anzahl der Hochschulbesucher steigt ständig an. Dies hat zur Folge, dass der Verwaltungsaufwand für die Zuteilung von Veranstaltungen an Dozenten immer größer wird. In dieser Arbeit sollte daher ein webbasiertes Informationssystem entwickelt werden, das die Aufgabenverteilung bzw. Veranstaltungsplanung für Dozenten vereinfacht und somit dessen eigenen Verwaltungsaufwand und den der Hochschule reduziert.

**Koray Kizilarlan**

**Title of the paper**

Development of a web-based information system for the distribution of tasks

**Keywords**

ASP.NET, Groupware, Ajax

**Abstract**

The number of high school visitors increases continuously. As a consequence, the administrative burden for the allocation of events to teachers is constantly increasing. The intention of this paper is to develop a web-based information system, which is especially designed to support the administration of teachers and to provide an allocation of tasks.

## **Danksagung**

Zunächst möchte ich Prof. Dr. rer. nat. Christoph Klauck für das interessante Arbeitsthema, die kompetente Betreuung der Arbeit und die Begutachtung der Bachelorarbeit danken.

Herrn Prof. Dr. Olaf Zukunft danke ich für die Bereitschaft, diese Arbeit als Zweitgutachter zu lesen und zu bewerten.

Außerdem danke ich Herrn Prof. Dr. rer. nat. Stephan Pareigis für das Verständnis, das er mir entgegengebracht hat und Jessica Rauch für das Korrekturlesen.

Zu guter Letzt ein herzliches Dankeschön an meine Eltern, meine Schwestern und ebenso an meine Freunde, die für den nötigen Ausgleich gesorgt und mich auf diese Weise stets motiviert haben, nicht nur während der Bachelorarbeit sondern auch über die gesamte Studienzeit.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>9</b>
<b>Abbildungsverzeichnis</b>	<b>10</b>
<b>1. Einleitung</b>	<b>12</b>
1.1. Motivation . . . . .	12
1.2. Aufgabenstellung . . . . .	13
1.3. Aufbau der Arbeit . . . . .	13
<b>2. Analyse</b>	<b>14</b>
2.1. Aufgabenstellung . . . . .	14
2.1.1. Beispielszenario . . . . .	14
2.1.2. Schwerpunkte und Abgrenzungen . . . . .	15
2.2. Anwendungsfälle . . . . .	16
2.2.1. Anwendungsfall Nr. 1: Semesterliste einspielen . . . . .	16
2.2.2. Anwendungsfall Nr. 2: Lehrkraft anlegen . . . . .	16
2.2.3. Anwendungsfall Nr. 3: Veranstaltung erstellen . . . . .	17
2.2.4. Anwendungsfall Nr. 4: Veranstaltung löschen . . . . .	17
2.2.5. Anwendungsfall Nr. 5: Kompetenzen erfassen . . . . .	18
2.2.6. Anwendungsfall Nr. 6: Kompetenzen bearbeiten . . . . .	18
2.2.7. Anwendungsfall Nr. 7: Veranstaltung buchen . . . . .	19
2.2.8. Anwendungsfall Nr. 8: Veranstaltung stornieren durch eine Lehrkraft . . . . .	20
2.2.9. Anwendungsfall Nr. 9: Veranstaltung stornieren durch den Admin . . . . .	20
2.3. Groupware . . . . .	21
2.3.1. Systemeigenschaften . . . . .	23
2.3.1.1. Klassifizierung . . . . .	23
2.3.1.2. Zentrale Aspekte . . . . .	26
2.3.2. Architektur . . . . .	27
2.3.3. freie und kommerzielle Groupware . . . . .	27
2.4. Anforderungsmerkmale . . . . .	29
2.4.1. Funktionale Merkmale . . . . .	29
2.4.2. Qualitätsbezogene Merkmale . . . . .	29
2.4.3. Systembezogene Merkmale . . . . .	32

---

2.4.4. Prozessbezogene Merkmale . . . . .	32
2.5. Systemarchitektur . . . . .	33
2.5.1. Datenbanken . . . . .	33
2.5.1.1. Hierarchisches Datenbankmodell . . . . .	33
2.5.1.2. Netzwerkdatenbankmodell . . . . .	34
2.5.1.3. Relationales Datenbankmodell . . . . .	35
2.5.1.4. Objektorientiertes Datenbankmodell . . . . .	36
2.5.1.5. Objektrelationales Datenbankmodell . . . . .	37
2.5.2. Programmiersprachen und -systeme . . . . .	38
2.5.2.1. Perl . . . . .	38
2.5.2.2. Java . . . . .	38
2.5.2.3. PHP . . . . .	38
2.5.2.4. ASP.NET . . . . .	39
2.5.2.5. JSP . . . . .	39
2.5.2.6. HTML . . . . .	39
2.5.3. Wahl der Datenbank und Programmiersprache . . . . .	40
2.6. State-of-the-Art . . . . .	40
<b>3. Konzept</b>	<b>41</b>
3.1. Problemdefinition und Anforderungsanalyse . . . . .	41
3.1.1. Funktionale Anforderungen . . . . .	42
3.1.2. Qualitätsbezogene Anforderungen . . . . .	44
3.1.3. Systembezogene Anforderungen . . . . .	45
3.2. Zeitlicher Ablauf . . . . .	46
3.3. Entwurf . . . . .	51
3.3.1. Fachklassen . . . . .	51
3.3.2. fachliche Architektur . . . . .	53
3.3.3. technische Architektur . . . . .	54
3.4. Sicherheitskonzept . . . . .	58
3.4.1. Authentifizierung . . . . .	58
3.4.2. Autorisierung . . . . .	60
3.4.3. Netzwerk- und Datensicherheit . . . . .	61
<b>4. Realisierung</b>	<b>62</b>
4.1. Grundlegendes . . . . .	62
4.1.1. Rollenverteilung . . . . .	62
4.1.2. Navigierung . . . . .	63
4.1.3. Masterseite . . . . .	63
4.1.4. Sicherheit . . . . .	64
4.2. Implementierung des Portals . . . . .	65
4.2.1. Applikationsschicht . . . . .	65

---

4.2.1.1. Verzeichnisstruktur . . . . .	66
4.2.1.2. Konfiguration . . . . .	67
4.2.1.3. Datenbankkommunikation . . . . .	70
4.2.2. Persistenzschicht . . . . .	72
4.2.3. Präsentationsschicht . . . . .	73
<b>5. Bewertung der Lösung</b>	<b>74</b>
5.1. Umfangsmetrik . . . . .	74
5.2. Vergleich des Portals mit den Anforderungen . . . . .	75
5.2.1. Funktionale Anforderungen . . . . .	75
5.2.1.1. Benutzerverwaltung . . . . .	75
5.2.1.2. Erstellen und Verwalten von Veranstaltungen . . . . .	76
5.2.1.3. Buchung und Zuordnung von Veranstaltungen . . . . .	77
5.2.1.4. Speicherung der Daten in einem relationalen Datenbankver- waltungssystem . . . . .	78
5.2.1.5. Darstellung einer Veranstaltungs-Historie . . . . .	78
5.2.1.6. Benachrichtigungen per Email . . . . .	78
5.2.2. Qualitätsbezogene Anforderungen . . . . .	79
5.2.2.1. Richtigkeit . . . . .	79
5.2.2.2. Usability . . . . .	79
5.2.2.3. Angemessenheit . . . . .	80
5.2.2.4. Interoperabilität . . . . .	80
5.2.2.5. Performance . . . . .	80
5.2.2.6. Sicherheit . . . . .	81
5.2.2.7. Persistenz der Daten . . . . .	81
5.2.3. Systembezogene Merkmale . . . . .	81
5.3. Gewonnene Erkenntnisse . . . . .	82
<b>6. Schluss</b>	<b>83</b>
6.1. Zusammenfassung . . . . .	83
6.2. Ausblick und Kritik . . . . .	84
<b>Literaturverzeichnis</b>	<b>86</b>
<b>A. Die Datenbank des Portals</b>	<b>91</b>
<b>B. Konfigurationsdateien</b>	<b>96</b>
<b>C. Beispielcode</b>	<b>100</b>
<b>D. Handbuch</b>	<b>103</b>
D.1. Das Portal . . . . .	103

D.1.1. Design der Oberfläche . . . . .	103
D.1.2. Benutzerbereich . . . . .	104
D.1.2.1. Login . . . . .	104
D.1.2.2. Übersicht . . . . .	105
D.1.2.3. Profil-Daten . . . . .	107
D.1.2.4. Profil-Daten ändern . . . . .	107
D.1.2.5. Passwort ändern . . . . .	108
D.1.2.6. Auswahl der Kompetenzen . . . . .	108
D.1.2.7. Übersicht Veranstaltungswahl . . . . .	109
D.1.2.8. Übersicht Veranstaltungs-Historie . . . . .	109
D.1.3. Admin Bereich . . . . .	110
D.1.3.1. Startseite . . . . .	110
D.1.3.2. Importieren von Veranstaltungslisten . . . . .	111
D.1.3.3. Erstellen von Veranstaltungen . . . . .	112
D.1.3.4. Bilden von Kompetenzgruppen . . . . .	113
D.1.3.5. Einstellungen . . . . .	113
D.1.3.6. Benutzerübersicht . . . . .	116
D.1.3.7. Benutzerdetails ändern . . . . .	117
D.1.3.8. Registrieren von Benutzern . . . . .	118
D.1.3.9. Passwort Recovery . . . . .	119
D.1.3.10. Semesterliste einsehen . . . . .	119
D.1.4. Zeitverhalten . . . . .	121
D.1.4.1. Deadline-Modus . . . . .	121
D.1.4.2. Normal-Modus . . . . .	121
D.1.4.3. Zeitpunkte . . . . .	122

**Glossar****123**



# Tabellenverzeichnis

5.1. Umfangsmetriken . . . . . 74

# Abbildungsverzeichnis

2.1. Übersicht über die Abgrenzung des Arbeitsthemas . . . . .	15
2.2. Raum-Zeit-Matrix als Einteilung der Groupware . . . . .	22
2.3. 3K-Modell . . . . .	24
2.4. Übersicht Software-Qualitätsmerkmale . . . . .	30
2.5. Hierarchisches Datenmodell . . . . .	33
2.6. Netzwerkdatenmodell . . . . .	34
2.7. Relationales Datenmodell . . . . .	35
2.8. Objektorientiertes Datenmodell . . . . .	36
2.9. Objektrelationales Datenmodell . . . . .	37
3.1. Zeitlicher Verlauf des Dozentenplaners (Phase 0 bis Phase II) . . . . .	47
3.2. Zeitlicher Verlauf des Dozentenplaners (Phase III bis Phase IV) . . . . .	49
3.3. fachliche Architektur . . . . .	52
3.4. Zentrale Client/Server-Architektur . . . . .	54
3.5. ASP.NET-Architektur . . . . .	55
3.6. 3-Schichten-Einteilung (3-Tier-Architektur) . . . . .	56
3.7. Architektur des Membership-Systems . . . . .	57
3.8. Mögliche Authentifizierungsverfahren . . . . .	58
3.9. Formularauthentifizierung . . . . .	59
4.1. Verzeichnisstruktur . . . . .	66
4.2. Hierarchie der web.config . . . . .	67
4.3. Aufbau der XML-Konfigurationsdatei . . . . .	68
4.4. Ausführung einer Prozedur mit anschließender Datenbindung . . . . .	70
4.5. Benutzeroberfläche von SQL Server Management Studio Express . . . . .	73
D.1. Übersicht Portal . . . . .	104
D.2. Login-Bereich . . . . .	105
D.3. Übersicht der Startseite des Portals . . . . .	106
D.4. Profildaten . . . . .	107
D.5. Profildaten ändern . . . . .	107
D.6. Passwort ändern . . . . .	108
D.7. Kompetenzen wählen . . . . .	108

---

D.8. Veranstaltungen wählen . . . . .	109
D.9. Veranstaltungs-Historie . . . . .	109
D.10. Startseite des Admins . . . . .	110
D.11. Semesterliste importieren . . . . .	112
D.12. Erstellen von einer Veranstaltung . . . . .	112
D.13. Bildung von Kompetenzgruppen . . . . .	113
D.14. Einstellungen . . . . .	114
D.15. Benutzerübersicht . . . . .	116
D.16. Benutzer kontaktieren . . . . .	117
D.17. Benutzerdetails ändern . . . . .	117
D.18. Registrierung von Benutzern . . . . .	118
D.19. Passwort wiederherstellen . . . . .	119
D.20. Semesterliste einsehen . . . . .	120

# 1. Einleitung

## 1.1. Motivation

Als Student einer Hochschule bekommt man die Problematik der Veranstaltungswahl und den damit verbundenen eigenen Verwaltungsaufwand auf verschiedenste Weise mit. Bestimmte Fristen für die Anmeldungen wurden nicht eingehalten, Voraussetzungen für eine Veranstaltung sind nicht erfüllt, die zu belegende Veranstaltung ist überbucht, die Wochenstundenanzahl ist zu hoch oder auch eine ungünstige Veranstaltungsgestaltung in der Woche, die mit vielen Freistunden verbunden ist. Weitere Verwaltungsaufgaben wie das Einplanen des Praxissemesters und der Zeit zum Lernen und Vorbereiten für Veranstaltungen, Prüfungen und Praktikas sind dabei nicht unerheblich.

Genau wie den Studenten steht auch den lehrenden Professoren die Vergabe eines adequaten Wochenstundenplans zu, der ihnen die Planung und Durchführung der Lehrveranstaltungen geeignet gestaltet. Professoren haben einen gewissen SollStunden-Satz an Lehrveranstaltungen in der Woche, den sie einhalten sollten. Um dies zu gewährleisten, muss sichergestellt werden, dass genügend Veranstaltungen angeboten werden, die von den Hochschulprofessoren gehalten werden können. Man könnte diese Zuordnung auch anders formulieren, indem genügend Professoren vorhanden sein müssen, um eine gewisse Menge an Veranstaltungen anbieten zu können. Allerdings hat die Hochschule sich auch an eine Studienordnung zu halten, die es vorschreibt, dass bestimmte Veranstaltungen für eine Studienrichtung vorhanden sein müssen.

Es muss also eine Zuordnung zwischen Lehrenden und Veranstaltung in dem Maße erfolgen, dass so wenig Engpässe wie möglich auftreten. Das hier entwickelte System dient in erster Linie als Warnsystem. Es überprüft, ob sich Lehrende rechtzeitig für Veranstaltungen angemeldet haben und erinnert Professoren daran sich, rechtzeitig für diese einzutragen.

Mit steigender Anzahl von Hochschulbesuchern und Professoren, steigt die Verwaltungsarbeit und die Übersichtlichkeit sinkt bei dieser Problematik. Aus diesem Grund ist es wichtig, die Möglichkeiten der heutigen Technologien praxisorientiert auch in diesen Prozessen einzusetzen.

## 1.2. Aufgabenstellung

Für eine autonom arbeitende Gruppe ist eine Plattform zu erstellen, über die sich Personen Aufgaben zuteilen können. Die Plattform ist im World Wide Web (WWW) zugänglich, jedoch über eine Zugangskontrolle gesichert. Bei der Aufgabenzuordnung werden Regeln und Rahmenbedingungen überwacht. Werden diese nicht eingehalten, folgt eine Warnmeldung, durch die alle Betroffenen darüber informiert werden.

## 1.3. Aufbau der Arbeit

Im Kapitel Analyse wird die Aufgabenstellung anhand eines Beispielszenarios erläutert und das zu entwickelnde System auf seine Bestandskomponenten und die Art der Verknüpfungen zwischen ihnen analysiert. Außerdem werden Technologien und Hardware vorgestellt, die für die Lösung der gestellten Aufgabe relevant sind, und ausgewählt, welche von ihnen in dieser Arbeit verwendet werden.

Im Kapitel Konzept werden die Anforderungsmerkmale konkretisiert und der zu realisierende Entwurf vorgestellt.

In der Realisierung wird das Ergebnis der Implementierung präsentiert, wobei zusätzlich die Beleuchtung der einzelnen Schichten vorgenommen wird.

Die umgesetzten Anforderungen werden daraufhin in der Realisierung mit den geforderten verglichen.

Letztendlich werden im Kapitel Schluss die Ergebnisse der Arbeit vorgestellt und analysiert. Zudem wird auf Möglichkeiten der Weiterentwicklung des Systems eingegangen.

## 2. Analyse

Die Aufgabenstellung wird in diesem Kapitel durch ein Beispielszenario näher analysiert, die Schwerpunkte der Arbeit erläutert und Abgrenzungen des Themas werden festgelegt.

Die für das System grundlegenden Anwendungsfälle werden in diesem Kapitel beschrieben. Der Bereich Groupware wird näher betrachtet, da hiervon einige Aspekte in der Arbeit eine Rolle spielen. Dabei werden die Bereiche Systemeigenschaften, Architektur und bereits im Einsatz befindliche Groupware beleuchtet.

Die Anforderungsmerkmale für Groupware werden als Grundlage genommen, um die für die Aufgabe relevanten Bereiche zu analysieren.

Aufbauend auf diesem Kapitel wird auf die möglichen Systemarchitekturen eingegangen. Abschliessend wird der Stand der Kunst betrachtet.

### 2.1. Aufgabenstellung

#### 2.1.1. Beispielszenario

Als Beispielszenario für diese Arbeit wird die Veranstaltungsverteilung und somit die Stundenplangestaltung/verwaltung von Professoren herangezogen.

Professoren arbeiten autonom und haben die Aufgabe, Veranstaltungen wie Vorlesungen, Projekte, Übungen und Praktika zu realisieren und durchzuführen. Über die erstellte Plattform ist es den Professoren möglich, sich für favorisierte Veranstaltungen anzumelden und diese zu buchen. Sie bietet den Professoren zudem eine bessere Übersichtlichkeit bezüglich der in dem jeweiligen Semester angebotenen Veranstaltungen für Studierende.

Bei Mehrfachbuchungen von einzelnen Veranstaltungen haben die Professoren zudem die Möglichkeit, über diese Plattform untereinander zu kommunizieren.

Über bestimmte Vergaberegeln kann hierbei der Administrator (Admin) Einfluss auf die Art und Weise der Veranstaltungsverteilung ausüben. So endet z.B. die Verteilung nach Ablauf einer vom Admin festgelegten Frist, so dass gewährleistet wird, dass die angebotenen Veranstaltungen zu Semesterbeginn unter den Professoren zugeteilt sind. Zudem wird der Admin im Falle von Konflikten wie z.B. Mehrfachbuchungen über eine Email benachrichtigt

und kann diese bearbeiten beziehungsweise beheben.

Letztendlich wird jedem Professor über seine gebuchten Veranstaltungen eine Bestätigungsmail geschickt.

### 2.1.2. Schwerpunkte und Abgrenzungen

Ein Schwerpunkt dieser Arbeit sollte die Verteilung von Aufgaben an eine autonom arbeitende Gruppe darstellen. In diesem Fall stellen die Professoren einer Hochschule diese Gruppe dar und die Aufgabenverteilung die Zuordnung einzelner Veranstaltungen auf diese. Jedes Semester werden mehrere Studiengänge angeboten, die aus Modulen und Veranstaltungen bestehen (s. Abb. 2.1). Hierbei lehren Dozenten in Bereichen, die ihren eigenen Schwerpunkten und Fähigkeiten, d.h. ihren Kompetenzen entsprechen. Die Verteilung von Terminen und Räumen zu den jeweiligen Veranstaltungen ist jedoch nicht Thema dieser Abschlussarbeit, da dies nur in diesem speziellen Beispielszenario eine Rolle spielt. Für andere autonom arbeitende Gruppen könnten räumliche und terminbezogene Zuordnungen keine Relevanz darstellen und werden daher in dieser Arbeit nicht weiter berücksichtigt.

Die Plattform soll im WWW nur für eine bestimmte Gruppe von Nutzern zugänglich sein, daher besteht ein weiterer Teil dieser Arbeit in der Gewährleistung eines sicheren An- und Abmeldens in der erstellten Plattform, um diese Gruppe im WWW zu separieren. Außerdem trennt eine Rollenverwaltung die einzelnen Gruppenmitglieder voneinander, da bei der Aufgabenverteilung Regeln und Rahmenbedingungen überwacht werden sollen. In diesem Beispielfall existieren daher Professoren und der Admin, der für die Einhaltung und Erstellung der Regeln zuständig ist. Abbildung 2.1 gibt eine Übersicht über die Schwerpunkte dieser Arbeit. Hierbei sind die grau markierten Bereiche nicht Thema dieser Arbeit.

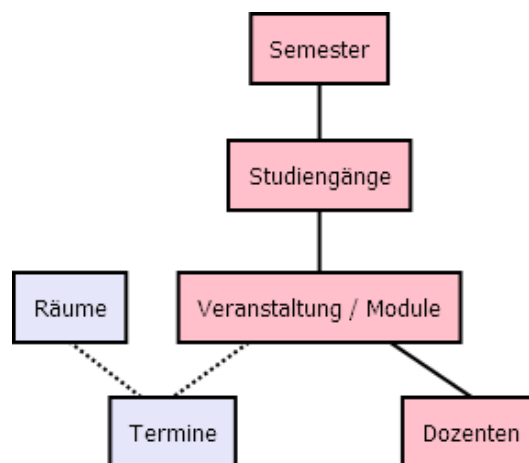


Abbildung 2.1.: Übersicht über die Abgrenzung des Arbeitsthemas

## 2.2. Anwendungsfälle

Um einen Überblick über das System zu bekommen, werden an dieser Stelle Anwendungsfälle definiert, die üblicherweise während der Laufzeit des Systems auftreten können. Alle Anwendungsfälle werden, soweit möglich, durch eine kurze Zusammenfassung, bestehende Vorbedingung und Nachbedingung, sowie einer kurzen Beschreibung des Szenarios und dessen Ausnahmen beschrieben.

Sie dienen später als Grundlage für die Analyse und sollen die Grundfunktionalität darstellen, die für die Verteilung nötig ist.

### 2.2.1. Anwendungsfall Nr. 1: Semesterliste einspielen

Dieser Anwendungsfall beschreibt den typischen Ablauf beim Einspielen einer neuen Semesterliste, die die anzubietenden Veranstaltungen eines Semesters enthält.

**Vorbedingung:**

Man ist als Admin eingeloggt. Die Liste ist eine .csv-Datei, welche als Feldtrenner Semikolons benutzt. Die Spaltenspezifikation entspricht den Vorgabebedingungen (Semester, Planungseinheit, Semestergruppe, Fach, Semesterwochenstunde (SWS), Lehrveranstaltungsstunde (LVS), Art, Fachbezeichnung).

Das betreffende Semester ist noch nicht eingespielt.

**Nachbedingung:**

Die Semesterliste wurde erfolgreich in die Datenbank eingespielt

**Beschreibung des Szenarios:**

Eine neue Liste der Veranstaltungen für das nächste Semester steht fest. Der Admin spielt diese Liste ein. Das System stellt diese Liste im betreffenden Zeitraum (entsprechendes Semester) der Auswahl der noch zu vergebenden Veranstaltungen dar.

**Ausnahmen:**

Veranstaltungen, die bereits in der Vergangenheit eingespielt wurden, werden nicht neu angelegt bzw. nicht überschrieben.

### 2.2.2. Anwendungsfall Nr. 2: Lehrkraft anlegen

Der Anwendungsfall beschreibt das Erstellen einer Lehrkraft, die zugleich als Benutzer der Plattform fungiert.



**Vorbedingung:**

Man ist als Admin eingeloggt. Eine neue Lehrkraft existiert noch nicht im System.

**Nachbedingung:**

Die Lehrkraft wurde erfolgreich in das Benutzerverzeichnis des Systems eingespielt.

**Beschreibung des Szenarios:**

Eine neue Lehrkraft wird in der Fachhochschule eingestellt bzw. angeworben. Der Admin legt den neuen Professor/Assistenten im System an, um diesen bei der Verteilung der Veranstaltungen mit einbeziehen zu können.

### 2.2.3. Anwendungsfall Nr. 3: Veranstaltung erstellen

Die Änderung der Prüfungsordnung erfordert gegebenenfalls das Nachtragen notwendiger Veranstaltungen. Daher beschreibt der Anwendungsfall das Erstellen einer Veranstaltung, welche nicht in der importierten Veranstaltungsliste (s. 2.2.1 Anwendungsfall Nr. 1.0) enthalten war.

**Vorbedingung:**

Man ist als Admin eingeloggt. Die Deadline für das betreffende Semester, in das die Veranstaltung einbezogen werden soll, ist noch nicht abgelaufen.

**Nachbedingung:**

Die Veranstaltung wurde erfolgreich in das System eingespielt.

**Beschreibung des Szenarios:**

Eine Änderung in der Prüfungsordnung erfordert das Nachtragen gewisser Veranstaltungen. Die Liste für das aktuelle Semester wurde schon eingespielt. Der Admin erstellt die neue Veranstaltung über die Webanwendung, so dass diese auch in der Liste der noch zu buchenden Veranstaltungen auftaucht.

### 2.2.4. Anwendungsfall Nr. 4: Veranstaltung löschen

Der Anwendungsfall beschreibt das Löschen einer Veranstaltung, welche in der Liste der noch zu buchenden Veranstaltungen enthalten ist.

**Vorbedingung:**

Man ist als Admin eingeloggt. Die Deadline ist noch nicht abgelaufen und die Veranstaltung wurde von keiner Lehrkraft gebucht.

**Nachbedingung:**

Die Veranstaltung ist nicht mehr im System vorhanden.

**Beschreibung des Szenarios:**

Ein Professor meldet sich für längere Zeit krank. Da keine andere Lehrkraft vorhanden ist, die diese Vorlesung halten kann, muss diese aus dem Lehrplan für das betreffende Semester gestrichen werden. Der Admin löscht die Veranstaltung über die Webanwendung, so dass diese aus der Liste der noch zu buchenden Veranstaltungen entfernt wird.

### 2.2.5. Anwendungsfall Nr. 5: Kompetenzen erfassen

Der Anwendungsfall beschreibt das Erfassen der Fähigkeiten bzw. Kompetenzen einer Lehrkraft.

**Vorbedingung:**

Man ist als Lehrkraft eingeloggt.

**Nachbedingung:**

Die Kompetenz wurde der Lehrkraft zugewiesen und entsprechende Veranstaltungen werden sichtbar.

**Beschreibung des Szenarios:**

Die Lehrkraft hat sich erstmals in der Webanwendung eingeloggt. Um eine Auflistung der zu unterrichtenden Veranstaltungen für die Lehrkraft zu erhalten, wird die Eingabe der jeweiligen Fähigkeiten bzw. Kompetenzen vom System benötigt. In einem Web-Formular wird die Lehrkraft aufgefordert, entsprechende Kompetenzen zu kennzeichnen.

### 2.2.6. Anwendungsfall Nr. 6: Kompetenzen bearbeiten

Der Anwendungsfall beschreibt das Erweitern bzw. Löschen von Kompetenzen einer Lehrkraft.

**Vorbedingung:**

Man ist als Lehrkraft eingeloggt. Im Falle einer Entfernung von Kompetenzen sollten keine Veranstaltungen gebucht worden sein, für die die entsprechenden Fähigkeiten erforderlich sind.

**Nachbedingung:**

Die Kompetenz wurde erfolgreich von der Lehrkraft entfernt oder hinzugefügt. Im Falle der

Erweiterung seiner Fähigkeiten, werden dem Lehrenden weitere entsprechende Veranstaltungen angezeigt und können nun gebucht werden.

Im Falle der Entfernung einer Fähigkeit, werden die entsprechenden Veranstaltungen dem Lehrenden nicht mehr angezeigt.

**Beschreibung des Szenarios:**

Eine Lehrkraft hat einen Fortbildungskurs besucht und damit seine Kompetenzen erweitert. In der Webanwendung kann der Lehrende nun seine erworbenen Fähigkeiten angeben, wodurch ihm bei der Wahl der zu buchenden Veranstaltungen ein größeres Spektrum zur Verfügung steht.

Wird hingegen der Inhalt einer Veranstaltung durch die Prüfungsordnung insoweit geändert, dass der Lehrende eine Fortbildung besuchen müsste, um die Veranstaltung halten zu können, wird die Möglichkeit gegeben, sich ebenfalls einer Kompetenz eigenständig zu entziehen.

**Ausnahmen:**

Eine neu erworbene Kompetenz ist für den Lehrplan irrelevant, wenn hierfür keine Veranstaltung auf dem Lehrplan laut Prüfungsordnung vorgesehen ist. Hierdurch würde sich das Spektrum der noch zu buchenden Veranstaltungen entsprechend nicht erweitern.

### 2.2.7. Anwendungsfall Nr. 7: Veranstaltung buchen

Der Anwendungsfall beschreibt das Buchen einer Veranstaltung aus der Liste der zur Verfügung stehenden Veranstaltungen.

**Vorbedingung:**

Man ist als Lehrkraft eingeloggt. Die Kompetenzen der Lehrkraft wurden bereits vom System erfasst.

Die Deadline ist noch nicht abgelaufen.

**Nachbedingung:**

Die Veranstaltung erhält den Status 'angefragt' und wird in der Liste der gebuchten Veranstaltungen dargestellt.

**Beschreibung des Szenarios:**

Ein Professor stellt sich für das kommende Semester seinen Stundenplan zusammen. Hierbei wählt er seine favorisierten Veranstaltungen aus dem ihm dargestellten Spektrum an Veranstaltungen aus.

### **2.2.8. Anwendungsfall Nr. 8: Veranstaltung stornieren durch eine Lehrkraft**

Der Anwendungsfall beschreibt das Stornieren einer bereits gebuchten Veranstaltung durch eine Lehrkraft.

**Vorbedingung:**

Man ist als Lehrkraft eingeloggt. Die Deadline ist noch nicht abgelaufen.

**Nachbedingung:**

Die Veranstaltung wurde erfolgreich aus der Liste der Lehrkraft entfernt, die die bereits gebuchten Veranstaltungen dieser Lehrkraft enthält.

**Beschreibung des Szenarios:**

Ein Professor stellt bei der Übersicht seines selbst erstellten Stundenplanes fest, dass er an einem Tag keine Veranstaltung halten kann. Er storniert diese Veranstaltung aus seinem gebuchten Stundenplan heraus. Die Veranstaltung taucht nun wieder in der Liste der noch zu buchenden Veranstaltungen auf.

### **2.2.9. Anwendungsfall Nr. 9: Veranstaltung stornieren durch den Admin**

Der Anwendungsfall beschreibt das Stornieren einer bereits gebuchten Veranstaltung durch den Admin.

**Vorbedingung:**

Man ist als Admin eingeloggt. Die Deadline ist bereits abgelaufen.

**Nachbedingung:**

Die Veranstaltung wurde erfolgreich aus der Liste, von der Lehrkraft bereits gebuchten Veranstaltungen, entfernt.

Die Lehrkraft wird darüber informiert.

**Beschreibung des Szenarios:**

Das System hat nach Ablauf der Deadline festgestellt, dass für bestimmte Veranstaltungen keine Anfragen vorhanden sind. Beim Admin läuft diese Nachricht auf. Dieser hat jetzt die Möglichkeit aus der Liste der gebuchten Veranstaltungen, Veranstaltungen wieder freizugeben. Um die Lehrkräfte herauszufiltern, die die noch nicht gebuchten Veranstaltungen übernehmen könnten, kann er vorab in der Kompetenzübersicht der Lehrkräfte die betreffenden Namen der Professoren einsehen.

## 2.3. Groupware

Einige Aspekte von Groupware finden Verwendung in dieser Arbeit und werden daher im Folgenden kurz erläutert.

Groupware setzt sich aus den Wörtern „group“ und „software“ zusammen.

*Eine zentrale Definition für den Begriff „Groupware“ stammt von Ellis, Gibbs und Rein. Danach ist Groupware ein „computer-basiertes System, das eine Gruppe von Personen in ihrem Aufgabengebiet oder Ziel unterstützt und eine Schnittstelle für eine geteilte Arbeitsumgebung bietet“.<sup>1</sup>*

Groupware versucht im Gegensatz zu anderen Softwareanwendungen, die Isolation der Benutzer voneinander zu reduzieren. Es ist ein Hauptmerkmal von Groupware, dem Benutzer ein Wissen über die Kollegen und deren Aktivitäten zu vermitteln. Dies ist auch unter dem Begriff Awareness bekannt, welcher in einem späterem Abschnitt (Kapitel 2.3.1.2) noch genauer erläutert wird. Groupware schafft bei einem Benutzer das Bewusstsein, dass er ein Teil einer Gruppe ist, während andere Mehrbenutzersysteme versuchen, den Benutzer vor anderen zu verstecken und zu schützen<sup>2</sup>.

Professoren sind ein Teil der Gruppe, die in der Lage ist, Veranstaltungen zu wählen. Wichtig ist auch hier, dass die Benutzer untereinander kommunizieren können und dass sie einen Überblick erhalten, wenn bestimmte von ihnen favorisierte Veranstaltungen auch von anderen Benutzern ausgewählt wurden. Sie sollen miteinander kommunizieren, um sich untereinander über die Veranstaltungsverteilung zu einigen.

Das zweite Merkmal von Groupware ist die Anpassbarkeit. Groupware muss nicht nur auf die Anforderungen der Gruppe als Ganzes, sondern auch auf die individuellen Anforderungen aller Gruppenmitglieder eingehen.

Damit Benutzer und Gruppen auf unterschiedliche Art und Weise die Groupware nutzen können, muss diese sehr generisch und/oder anpassbar sein.

Das Nutzen von Emails wäre das beste Beispiel für diesen Aspekt von Groupware. Auch auf der erstellten Plattform sollen die einzelnen Nutzer, hier die Professoren, untereinander über ein Nachrichtensystem kommunizieren. Zudem dient dies außerdem der Bestätigung von Veranstaltungswahlen oder der Warnmeldung, falls vorgegebene Regeln oder Rahmenbedingungen nicht eingehalten wurden.

*Eine Groupware ist ein Email- und Kommunikationssystem, das folgende grundlegende Inhalte bereitstellt: Adressen und Kontakte, Aufgaben, Email, gemeinsame Folder, Kalender sowie Notizen.<sup>3</sup>*

---

<sup>1</sup>vgl. (Ellis u. a., 1991, 40)

<sup>2</sup>vgl. (Lynch u. a., 1990, 160)

<sup>3</sup>vgl. C. Böttger (2008)

Mit dieser Definition als Grundlage, wird in der Computer-Fachpresse und der Softwareindustrie die Groupware mit bekannten Client/Server-Produkten wie beispielsweise Microsoft Outlook/Exchange gleichgesetzt.

Tatsächlich wird der Begriff Groupware durch den Forschungsbereich Computer Supported Cooperative Work (CSCW) aber breiter betrachtet. Zu Groupware zählen alle Lösungen, welche Gruppen bei der Zusammenarbeit unterstützen.

Bei der Unterstützung asynchroner Zusammenarbeit gehören neben der in der Definition genannten Merkmale auch Werkzeuge zur Verwaltung gemeinsamer Datenbestände (Informationsräume) mit Awareness-Funktionalität dazu.

Bei der Unterstützung synchroner Zusammenarbeit betrachtet man bei Groupware auch verschiedene Konferenzsysteme, Instant-Messaging-Anwendungen, Gruppendeditoren und Social Software.

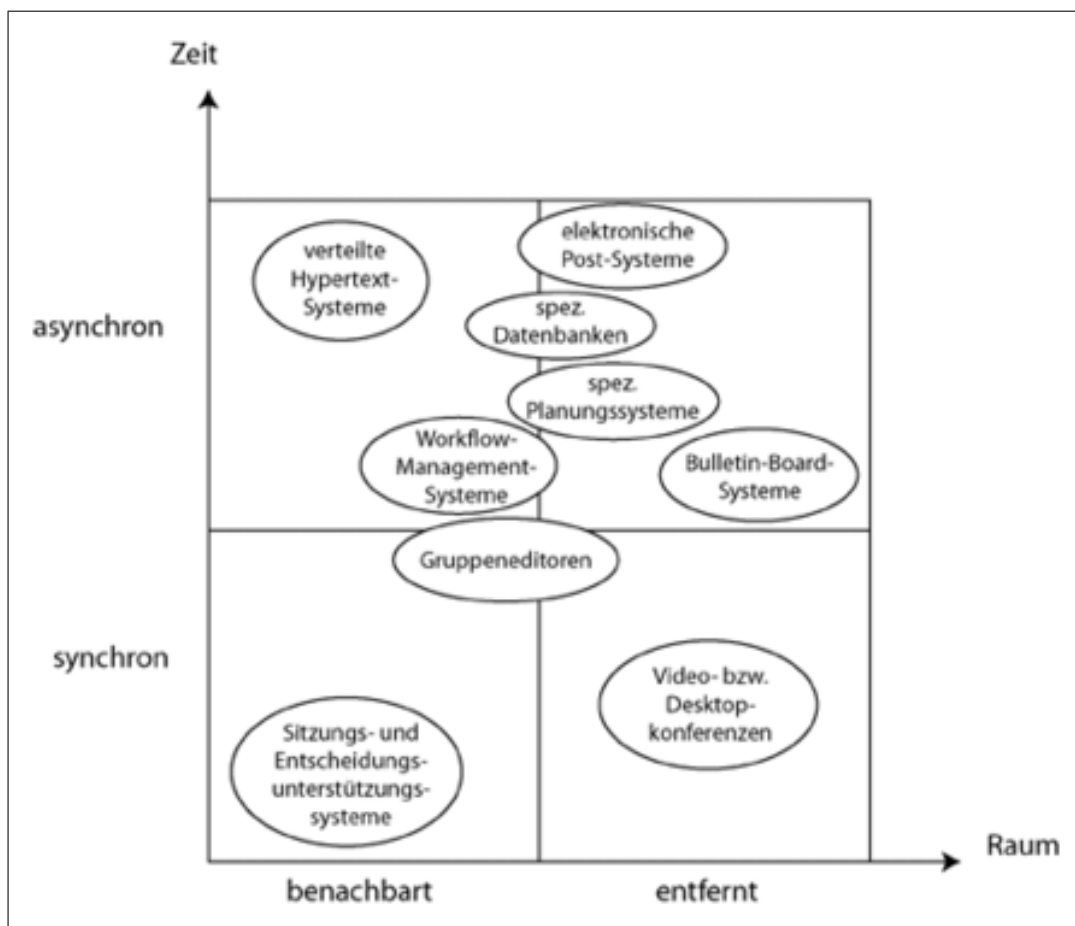


Abbildung 2.2.: Raum-Zeit-Matrix als Einteilung der Groupware

Die Abbildung 2.2<sup>4</sup> zeigt die Einteilung in Raum- und Zeitmerkmale, welche die am weitesten verbreitete Einteilung der Groupware ist<sup>5</sup>. Diese stellt eine Aufhebung der räumlichen und zeitlichen Barrieren dar und wird durch die Einteilung in rein synchrone und rein asynchrone Kooperationen aber auch durch die Ermöglichung beliebiger Synchronitäten erreicht.

Synchrone Kommunikation bedeutet, dass diese zur gleichen Zeit aber an unterschiedlichen Orten stattfindet wie beispielsweise bei einer Instant-Messaging-Anwendung.

Asynchrone Kommunikation bezieht sich auf den zeitlich unterschiedlichen Informationsaustausch wie es beispielsweise beim Email-Verkehr oder in der vorliegenden Arbeit beim Nachrichteninformationssystem zu sehen ist.

### 2.3.1. Systemeigenschaften

Ziel von Groupware ist die Unterstützung eines Gruppenprozesses. Hiermit ist die gemeinsame Erarbeitung eines Ergebnisses gemeint bzw. die Verarbeitung der Informationen eines Anfangszustandes in einen Endzustand. Der Anfangszustand wäre in dieser Arbeit, in der erstellten Plattform für Professoren, die Erstellung einer neuen Veranstaltungsliste zum jeweiligen Semester. Die Veranstaltungen sind dann durch die verschiedenen Gruppenmitglieder, die Professoren, wählbar und können in einem Gruppenprozess verteilt werden, der die Kommunikation der einzelnen Professoren untereinander beinhaltet. Nach Abschluss der Veranstaltungszuteilung wird dann zu einem späteren Zeitpunkt der Endzustand erreicht.

#### 2.3.1.1. Klassifizierung

Groupware lässt sich nach verschiedenen Faktoren klassifizieren<sup>6</sup>:

- 3K-Modell
- Ort
- Zeit
- Steuerung der Kommunikation
- Struktur
- Größe

---

<sup>4</sup>vgl. (Teufel u. a., 1995, 25)

<sup>5</sup>vgl. (Seufert und Back, 2000, 8 f)

<sup>6</sup>vgl. (Teufel u. a., 1995, 23 f)

**3K-Modell:**

Abbildung 2.3<sup>7</sup> veranschaulicht das 3K-Modell in Verbindung mit ausgewählten CSCW-Applikationen.

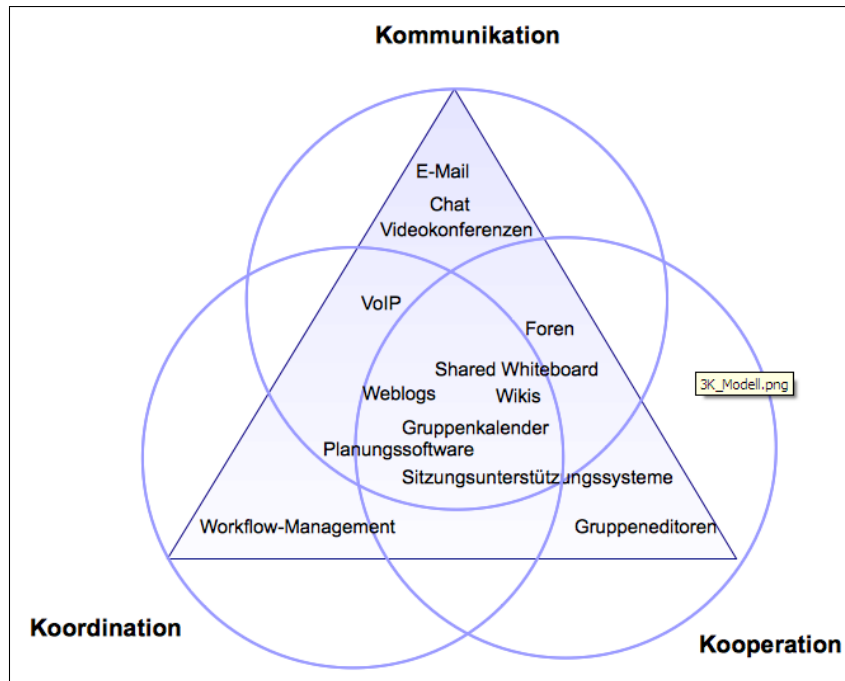


Abbildung 2.3.: 3K-Modell

- **Kommunikation:**  
In welchem Ausmaß kommunizieren die Teilnehmer miteinander?  
Die Kommunikation stellt die Grundlage der Kooperation und Koordination dar<sup>8</sup>.
- **Kooperation:**  
In welchem Ausmaß wird auf das Erreichen eines gemeinsamen Zieles hingearbeitet?
- **Koordination:**  
In welchem Ausmaß werden Ressourcen gemeinsam geplant?<sup>9</sup>

<sup>7</sup>vgl. (Teufel u. a., 1995, 27)

<sup>8</sup>vgl. (Seufert und Back, 2000, 42)

<sup>9</sup>vgl. (Pawlak u. a., 2007, 200 ff)



**Ort:**

Die Zusammenarbeit unterscheidet sich in der räumlichen Distanz. Lokal würde beispielsweise bedeuten, dass die Zusammenarbeit im selben Büro stattfindet und verteilt bedeutet eine Zusammenarbeit über verschiedene Städte. Durch die Nutzung einer webbasierten Lösung für die hier erstellte Plattform wird eine verteilte Zusammenarbeit begünstigt.

**Zeit:**

Die Zusammenarbeit unterscheidet sich in der zeitlichen Differenz. Eine gleichzeitige Zusammenarbeit (synchron) würde beispielsweise in einer Videokonferenz stattfinden, wohingegen eine zeitversetzte Zusammenarbeit (asynchron) über Email zustandekommen würde. Auf der Plattform wird eine asynchrone Zusammenarbeit unterstützt.

**Steuerung der Kommunikation:**

Bei der Initiierung einer Kommunikation unterscheidet man zwischen Hol- und Bringschuld (Pull- bzw. Pushvorgänge). Die Initiierung des Kommunikationsvorgangs durch den Sender bezeichnet man als Push-Vorgang. Dies wären beispielsweise ein Anruf bei Voice over IP. Als Pull-Vorgänge wird eine durch den Empfänger initiierte Kommunikation beschrieben. Werden Nachrichten von einer Webseite abgeholt, bezeichnet man dies als Pull-Vorgang. Die Kommunikation auf der Plattform wird über ein Nachrichtensystem abgewickelt und ist daher ein Pull-Vorgang.

**Struktur:**

Bei der Groupware wird zwischen einem strukturierten und unstrukturierten Ablaufplan unterschieden. Ein Strukturierter Ablaufplan wäre z.B. eine Tagesordnung oder eine Semesterliste, wohingegen ein unstrukturierter Ablaufplan Brainstorming wäre.

**Größe:**

Eine Gruppe von Benutzern, die im Team arbeiten, wird durch ihre Größe unterschieden. Hierbei unterscheidet man zwischen großen (beispielsweise einer Vorlesung) und kleinen Gruppe (beispielsweise ein Projekt). Übertragen auf die Aufgabenstellung wäre dies die Sichtweise der Lernenden. Bei den Lehrkräften wird diese Unterteilung über entsprechende Kompetenzen für Veranstaltungen bestimmt.

### 2.3.1.2. Zentrale Aspekte

Zentrale Aspekte einer Groupware sind:

- **Awareness:**  
Viele Systeme setzen eine oder mehrere Formen der Awareness (Gruppenerkenntnis) um. Die Formen der Awareness reichen von der Rückmeldung der Anwesenheit der Teilnehmer (Onlinestatus), der aktuellen Verfügbarkeit (z.B. Abwesenheitsnotiz bei Email) bis hin zur detaillierten Übermittlung bestimmter Handlungen innerhalb des gemeinsamen Handlungsbereiches (z.B. „Aufgabe A wird übernommen von ...“). Die Software überwacht und informiert über die Zusammensetzung der Gruppe deren Teilnahme und Handlungen.
- **WYSIWIS (What You See Is What I See):**  
WYSIWIS beschreibt die Fähigkeit der Benutzer, sehen zu können, wie der Stand der Dinge bei anderen Benutzern ist (Was für Arbeiten werden gemacht und wie lange dauern diese noch an? Welche Veranstaltungen wurden gebucht und von wem?). Ebenso sollten Informationen bei den Benutzern einheitlich dargestellt werden.
- **Informationsarchitektur:**  
Die Informationsarchitektur beschreibt die inhaltliche Struktur der Informationen und die Benutzerzugänge.
- **Synchronisation und Konsistenzhaltung:**  
Die Datenbasis soll in sich schlüssig bleiben, auch wenn mehrere Benutzer gleichzeitig darauf zugreifen. Falls dennoch ein Konflikt auftritt, sollte dieser durch das System aufgezeigt werden.
- **Floor-Control:**  
Das System ist in der Weise verwaltbar, dass festgelegt werden kann, welcher Benutzer auf welche Daten zugreifen darf (Verwaltung von Systemressourcen)
- **Session-Control:**  
Das System ist in der Weise verwaltbar, dass festgelegt werden kann, welcher Benutzer in welche Arbeitsgruppe beitreten und welche Rolle er annehmen darf. (Verwaltung von Benutzern)

### 2.3.2. Architektur

Groupware ist ein verteiltes System, das die gemeinsame Bearbeitung einer Menge von Daten oder Dokumenten ermöglicht. Sie setzt daher stets eines der folgenden Konzepte um:

- **Zentrale Architektur**  
Alle Benutzer (Clients) sind mit einem Server verbunden. Über den Server werden die Ressourcen zentral verwaltet, aufgeteilt und zur Verfügung gestellt. Anfragen durch den Client werden vom Server ausgewertet und entsprechende Daten werden zurückgesendet.
- **Peer-to-Peer-Architektur**  
In einem Peer-to-Peer-Netzwerk ist jeder verbundene Computer zu den anderen gleichberechtigt. Jeder Computer stellt den anderen Computern seine Ressourcen zur Verfügung. Da es keinen Netzwerkverwalter gibt, muss jeder Benutzer selbst bestimmen, welche Ressourcen er freigeben will.
- **Hybride Architektur**  
Die Peer-to-Peer-Architektur wird um einen zentralen Server ergänzt, der jedoch lediglich der Protokollierung dient.

In dieser Arbeit wird eine zentrale Architektur umgesetzt.

### 2.3.3. freie und kommerzielle Groupware

CSCW-Applikationen unterstützen dediziert spezifische Aspekte der Gruppenarbeit. Es wird zwischen folgenden Kategorien unterschieden<sup>10</sup>:

- Email/Messaging
- Calendaring/Scheduling
- Electronic-Meeting Systems
- Real-Time Data Conferencing
- Non Real-Time Conferencing
- Group Document Handling

---

<sup>10</sup>vgl. (Seufert und Back, 2000, 11 ff)

- Workflow-Produkte

Die CSCW-Einzelanwendungen werden in sogenannten CSCW-Suites zusammengefasst, wodurch eine einheitliche Arbeitsumgebung geschaffen wird. Diese erlaubt es Arbeitsgruppen, Informationen zwischen den einzelnen Anwendungen auszutauschen. Einarbeitungs- und Schulungskosten werden durch die Verwendung von Suites erheblich reduziert.

Im Internet werden CSCW-Suites in Form von Collaborative Web Suites angeboten, welche die Nutzung der kollaborativen Funktionalitäten über einen Web-Browser ermöglichen.<sup>11</sup>

Die Produkte unterscheiden sich nicht nur durch ihren Funktionsumfang, sondern auch in den Anforderungen an die Betriebssysteme. Einige Lösungen sind für mehrere Plattformen vorhanden und andere sind speziell auf ein bestimmtes Betriebssystem zugeschnitten.

Die Einbindung mobiler Endgeräte gewinnt immer mehr an Bedeutung und wird bei jedem Produkt unterschiedlich umgesetzt.

Beliebte Groupware- und Kollaborationslösungen<sup>12</sup>:

- Microsoft Exchange Server 2010<sup>13</sup> (65% Marktanteil<sup>14</sup>  
ist nur auf einem Windows Server 2008 lauffähig
- Zarafa Zarafa<sup>15</sup>  
ist eine freie Groupware-Server-Lösung für Linux-Server.
- IBM Lotus Notes<sup>16</sup>  
ist auf dem Apple, Windows und Linux lauffähig.

Zu Groupware-Server-Lösungen die nicht kommerziell vertrieben werden, zählen beispielsweise auch openGroupware<sup>17</sup> und phpGroupware<sup>18</sup>.

---

<sup>11</sup>vgl. (Seufert und Back, 2000, 11 ff)

<sup>12</sup>vgl. Jeschke (2012)

<sup>13</sup>vgl. Microsoft (2012f)

<sup>14</sup>vgl. Aupperle (2012)

<sup>15</sup>vgl. Zarafa (2012)

<sup>16</sup>vgl. IBM (2012)

<sup>17</sup>vgl. MDLink (2009)

<sup>18</sup>vgl. Kuykendall (2012)

## 2.4. Anforderungsmerkmale

Groupwaresysteme lassen sich in die Reihe der Informationssysteme einordnen.

*Die Websystem-Problemdefinition ist die zusammenfassende Beschreibung von intentionalen Anforderungen an die Entwicklung eines Websystems, die sich in funktionale, qualitative, plattformbezogene und prozessbezogene Anforderungen unterteilen.<sup>19</sup>*

Die Anforderungen lassen sich in funktionale, qualitative, systembezogene und prozessbezogene Merkmale klassifizieren.

### 2.4.1. Funktionale Merkmale

Aus der Zuordnung zu den Informationssystemen lassen sich folgende Merkmale übernehmen<sup>20</sup>:

- Speicherung, Aktualisierung und Archivierung umfangreicher Datenbestände auf Datenträgern oder Medien
- die Möglichkeit der Informationsabfrage durch flexible Anfragesprachen oder Reports

### 2.4.2. Qualitätsbezogene Merkmale

Software wird über sechs Qualitätsmerkmale nach ISO 9126<sup>21</sup> (DIN 66272)<sup>22</sup> bewertet:

- Funktionalität: Sind alle geforderten Funktionen implementiert und ausführbar?
  - Richtigkeit: Fähigkeit des Systems, die richtigen oder vereinbarten Ergebnisse oder Wirkungen mit der gewünschten Genauigkeit bereitzustellen.
  - Angemessenheit: Fähigkeit des Systems, für spezifizierte Aufgaben geeignete Funktionen bereitzustellen.
  - Interoperabilität: Fähigkeit des Systems, mit anderen Systemen über vorgegebene externe Schnittstellen zusammenzuarbeiten.

---

<sup>19</sup>vgl. (Dumke u. a., 2003, 60)

<sup>20</sup>vgl. (Dumke, 2001, 215)

<sup>21</sup>vgl. Picand (2011)

<sup>22</sup>vgl. (Dumke u. a., 2003, 65 ff)

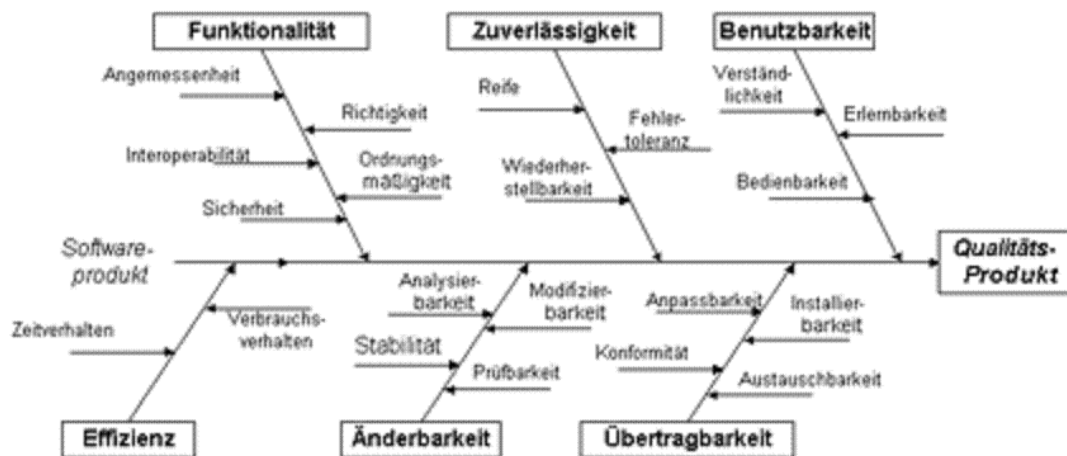


Abbildung 2.4.: Übersicht Software-Qualitätsmerkmale

- Ordnungsmäßigkeit: Fähigkeit des Systems anwendungsspezifische Normen, Vereinbarungen oder gesetzliche Bestimmungen und ähnliche Vorschriften einzuhalten.
- Sicherheit: Fähigkeit des Systems, die Daten und Programme so schützen, dass nicht autorisierte Personen oder Systeme diese nicht vorsätzlich oder versehentlich lesen oder manipulieren können.
- Zuverlässigkeit: Ist die Software permanent verfügbar und korrekt?
  - Reife: Das System ist, trotz Fehlern in der Software, in der Lage nicht zu versagen. Wenn eine Funktionalität einen Fehler hat, sollten andere Funktionalitäten trotzdem funktionieren.
  - Fehlertoleranz: Im Gegensatz zur Reife, die sich in Abgrenzung auf die Häufigkeit des Versagens bezieht, ist mit der Fehlertoleranz das Halten eines spezifizierten Leistungsniveaus (notwendigsten Funktionalitäten verfügbar) gemeint.
  - Wiederherstellbarkeit: Fähigkeit des Systems, bei einem Versagen (durch beispielsweise Blitzeinschlag) das Leistungsniveau und die betroffenen Daten wiederherzustellen. Berücksichtigt werden hierbei die dafür benötigte Zeit und der benötigte Aufwand.
- Benutzbarkeit: Ist die Software intuitiv zu benutzen und ist sie leicht zu erlernen?
  - Verständlichkeit: Kann der Benutzer das Konzept und die Anwendung ohne großen Aufwand verstehen?

- Bedienbarkeit: Wie hoch ist der Aufwand für den Benutzer, die Anwendung zu bedienen? Sind beispielsweise häufig benutzte Funktionen in zu tiefen Menüs?
- Erlernbarkeit: Wie hoch ist der Aufwand für den Benutzer, die Anwendung zu erlernen? (z.B. Bedienung, Ein-, Ausgabe)
- Effizienz: Wie effizient ist das zeitliche Verhalten bei Anfragen und Bearbeitungen? Wie hoch ist der Ressourcenverbrauch in Hinblick auf Speicherkapazität und Systemanforderungen?
  - Zeitverhalten: Wie ist die Antwort- und Verarbeitungszeit des Systems? Und wie hoch ist der Durchsatz bei der Funktionsausführung?
  - Verbrauchsverhalten: Welche Anzahl und über welche Dauer benötigt das System Betriebsmittel für die Erfüllung der Funktionen? (Ressourcenverbrauch, Festplattenzugriffe, CPU-Zeit usw.)
- Änderbarkeit: Wie hoch ist der Aufwand zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder Änderungen der Umgebung?
  - Analysierbarkeit: Wie hoch ist der Aufwand, um Mängel oder Ursachen von Versagen zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen?
  - Änderbarkeit: Wie hoch ist der Aufwand zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder zur Änderung der Umgebung?
  - Stabilität: Wie hoch ist die Wahrscheinlichkeit, dass bei einer durchgeführten Änderung im System unerwartete Fehlerwirkungen (Seiteneffekte) auftreten?
  - Testbarkeit: Wie hoch ist der Aufwand, der zur Prüfung der modifizierten Software notwendig ist?
- Übertragbarkeit: Ist die Software auch auf anderen Systemen (Hard- und Software) einsetzbar?
  - Anpassbarkeit: Die Fähigkeit der Software in verschiedenen spezifizierten Umgebungen laufen zu können.
  - Installierbarkeit: Wie hoch ist der Aufwand, das System in einer anderen Umgebung zu installieren?
  - Konformität: Grad, in dem die Software Normen oder Vereinbarungen zur Übertragbarkeit erfüllt.
  - Austauschbarkeit: Ist es möglich diese Software anstelle einer spezifizierten Anderen in der Umgebung jener Software zu verwenden? Und wie hoch ist der dafür notwendige Aufwand?

Die Merkmale, die für das System umzusetzen sind, werden in Kapitel 3.1.2 beschrieben.

### 2.4.3. Systembezogene Merkmale

- Hardware: Workstations, PC
- Betriebssysteme und Netze: Unix, Linux, Windows, Mac OS, WWW
- Technologie: Client/Server oder n-Tier
- Programmiersprachen und -systeme: Perl, Python, Java, PHP, JSP, ASP.NET, HTML, XML, SQML, SQL
- Datenbank-Managementsysteme: ORACLE, SYBASE, DB2, PostgreSQL, MySQL

### 2.4.4. Prozessbezogene Merkmale

Bei den prozessbezogenen Merkmalen geht es um die Anforderungen zum Entwicklungsverlauf und den eingesetzten Ressourcen (Kostenverhältnisse werden abgeschätzt)<sup>23</sup>. Da es sich um eine außerbetriebliche Arbeit handelt, wird auf diesen Punkt im späteren Teil nicht weiter eingegangen.

Es können hauptsächlich die Anforderungen an Informationssysteme übernommen werden<sup>24</sup>:

- Meist werden sequentielle Modelle (z.B. Wasserfall oder VModelle) angewendet. Bei webbasierten Systemen werden auch zyklische Modelle angewendet.
- Die Datenmodellierung wird in Abstimmung mit dem künftigen Nutzer realisiert.
- Kosten- und Aufwandsschätzverfahren sind im Allgemeinen sehr gut anwendbar.

---

<sup>23</sup>vgl. (Dumke u. a., 2003, 69 ff)

<sup>24</sup>vgl. (Dumke, 2001, 216)



## 2.5. Systemarchitektur

In diesem Abschnitt werden die für die Arbeit in Frage kommenden Datenbanken und Programmiersprachen vorgestellt.

### 2.5.1. Datenbanken

Eines der Grundprinzipien für Informationssysteme ist die Speicherung von Informationen. Ein Datenbankmodell ist die theoretische Grundlage für eine Datenbank und bestimmt, auf welche Art und Weise die Daten in der Datenbank gespeichert und bearbeitet werden können.

Im folgenden werden die bekannten Datenbankmodelle vorgestellt.

#### 2.5.1.1. Hierarchisches Datenbankmodell

Ein hierarchisches Datenbankmodell ist das älteste Datenbankmodell und kommt vorwiegend bei Informationsmanagementsystemen (IMS) zum Einsatz.

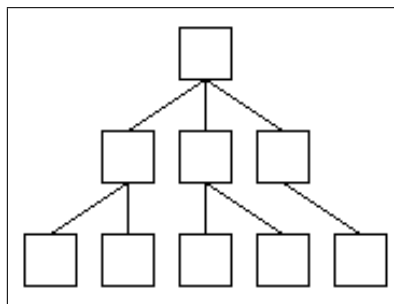


Abbildung 2.5.: Hierarchisches Datenmodell

Die Daten werden als hierarchisch strukturierte Datensätze in einer Baumform betrachtet. Dieses Konzept eignet sich für 1:n-Beziehungen, bei denen sich aus einem Oberbegriff viele Unterbegriffe ableiten lassen.

Ein Zugriff auf einen Datensatz benötigt die Pfadkenntnis und kann nur über den Suchschlüssel des Objektes auf oberster Ebene erfolgen.

**Vorteile:**

- einfaches Modell
- schnelles Suchen und schnelle Sortierung möglich

**Nachteile:**

- Verknüpfungen zwischen verschiedenen Bäumen oder über mehrere Ebenen innerhalb eines Baumes sind nicht möglich
- veraltet und durch andere Datenbankmodelle abgelöst worden
- Redundanz in Dateneinträgen, da diese unter Umständen in verschiedenen hierarchischen Zusammenhängen steht
- Schwierig im Nachhinein die Struktur der Datenbank wesentlich zu verändern

**2.5.1.2. Netzwerkdatenbankmodell**

Das Netzwerkdatenbankmodell ist eine Verallgemeinerung des hierarchischen Datenbankmodells und kann eine beliebige Anzahl übergeordneter Datensätze aufweisen (n:m-Beziehungen).

Jeder Datenknoten hat unmittelbaren Zugang zu jedem anderen Knoten.

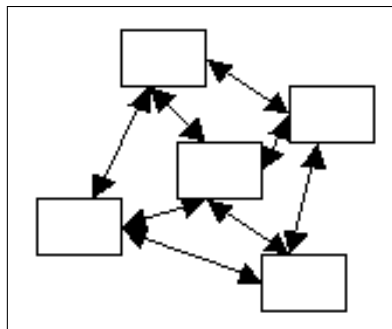


Abbildung 2.6.: Netzwerkdatenmodell

Das Konzept des Netzwerkdatenbankmodells eignet sich beispielsweise für die Modellierung von Stücklisten, da ein Bauteil aus mehreren untergeordneten Teilen besteht und in mehrere übergeordnete Baugruppen eingehen kann.

Das Datenbankmodell wird hauptsächlich von Großrechnern eingesetzt.

Der Zugriff auf einen Datensatz benötigt die Pfadkenntnis und kann über beliebige Datensätze erfolgen.

**Vorteil:**

- kein Knoten muss mehrfach vorhanden sein, da Erreichbarkeit über Verkettung geschaffen wird
- leistungsfähiger als relationale Modelle

**Nachteil:**

- Struktur wird ziemlich schnell undurchsichtig

**2.5.1.3. Relationales Datenbankmodell**

Beim relationalen Datenbankmodell stehen als Strukturelemente ausschließlich Relationen, die sich durch Tabellen darstellen lassen, zur Verfügung.

P-ID	Name	Vorname	Ort	Ort-ID	Fläche	Art	...
1	Müller	Hans	Oberdorf	5689	45869	Dorf	...
2	Meier	Jakob	Hinterwil	4758	23864	Weiler	...
3	Keiser	Josef	Unterdorf	2145	86541	Stadtteil	...
...	...	...	...	...	...	...	...

Abbildung 2.7.: Relationales Datenmodell

Zur Modellierung einer relationalen Datenbank wird das Entity-Relationship-Modell verwendet.

Jeder Entitätstyp und Beziehungstyp wird in ein eigenes Relationenschema abgebildet.

Das Schema legt fest, in welcher Beziehung die Daten zueinander stehen und wie diese in der Datenbank gespeichert werden.

Hierbei stehen die Zeilen für die Datensätze und die Spalten für die Datenfelder.

Der Zugriff auf Datensätze wird über die Feldinhalte ermöglicht.

**Vorteil:**

- Die Struktur der Datenbank kann geändert werden, ohne dass deswegen Anwendungen geändert werden müssen, die sich auf die ältere Struktur beziehen
- höherer Handhabungskomfort
- SQL-Fähig

**Nachteil:**

- Anforderungen an Rechen- und Zugriffsgeschwindigkeit steigen, da ein über Datenwerter definierter Zugriff die Durchsuchung aller Tabellen erforderlich machen könnte.

#### 2.5.1.4. Objektorientiertes Datenbankmodell

Im Unterschied zur relationalen Datenbank werden Daten hier als Objekte im Sinne der Objektorientierung verwaltet.

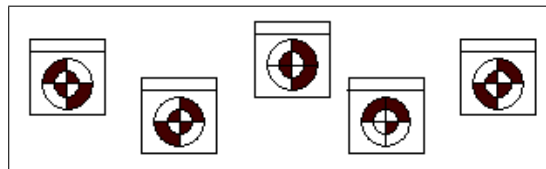


Abbildung 2.8.: Objektorientiertes Datenmodell

Ziel ist es, die Struktur und das Verhalten komplexer Objekte in der Datenbank abzubilden. Die Datenbank speichert die Objekte, ist selbst aber nicht objektorientiert.

##### **Aufgaben sind:**

- Verwaltung komplexer Objekte
- Sicherstellung der Objektidentität
- Kapselung der Objekte nach dem Konzept der objektorientierten Programmierung
- Objekte sind einer Objektklasse zugeordnet
- Objektklassen sind in einer Klassenhierarchie angeordnet
- Durch späte Bindung werden bei vererbten Objekten die überladenen Methoden verwendet
- Das objektorientierte Datenbankmanagementsystem (ODBMS) muss eine Turing-vollständige Manipulationssprache (DML) bereitstellen

##### **Vorteile**

- durch Entwicklung in einer objektorientierten Programmiersprache mit einem relationalen Datenbanksystem werden objektrelationale Abbildungen (Softwarekomponenten) verwendet, die zwischen der Datenbank und der Software vermitteln müssen. Durch die Verwendung eines Objektdatenbanksystems wird diese Vermittlung überflüssig, da die Anwendung direkt mit der Datenbank kommunizieren kann
- Zusammensetzen von Datensätzen mittels Joins entfällt, da die Objekte über die gespeicherten Beziehungen abgefragt werden können.

- Dem Datenbanksystem sind die semantischen Zusammenhänge zwischen den Objekten bekannt und weiß somit, welche Daten zusammengehören. Dies vereinfacht den Zugriff auf Daten.
- Problem der Objektidentität wird vom ODBMS übernommen.

### Nachteile

- komplizierte Implementierung
- geringe Verbreitung
- Schnittstellen teilweise noch nicht vorbereitet
- Performanceprobleme bei Zugriffspfaden zu Objekten (Vererbung und Assoziation)

#### 2.5.1.5. Objektrelationales Datenbankmodell

Das Objektrelationale Datenbankmodell stellt das Bindeglied zwischen relationalen Datenbanken und objektorientierten Datenbanken dar.





ID	XY	DF	ER
56		XXX	
45		YYY	
...	...	...	...

Abbildung 2.9.: Objektrelationales Datenmodell

Einige Konzepte des objektorientierten Datenbankmodells wurden von Herstellern relationaler Datenbanken übernommen und in spätere Versionen integriert. Einige Regeln, die beispielsweise übernommen wurden, sind:

- Speicherung komplexer Objekte
- Vergabe von eindeutigen Objektidentitäten
- Kapselung von Daten
- Typen und Klassen
- Vererbung von Struktur und Verhalten
- Anwendung gleicher Methodennamen auf Objekte unterschiedlicher Klassen

## 2.5.2. Programmiersprachen und -systeme

Wie im Abschnitt 2.4.3 gezeigt, können verschiedene Programmiersprachen und -systeme für die Umsetzung von Groupwaresystemen verwendet werden. Folgend wird auf die Programmiersprachen Perl, Java, PHP, JSP, ASP.NET, und HTML näher eingegangen.

### 2.5.2.1. Perl

Perl ist eine Interpretersprache, die in vielen Aspekten an C angelehnt ist, aber auch UNIX-Funktionen (sed und awk) und andere Sprachen integriert. Ein geschriebenes Programm wird erst direkt vor der Ausführung interpretiert. Perl stellt neben den wichtigen Programmierbefehlen wie Schleifen, Verzweigungen, usw. auch leistungsstarke Funktionen für die Manipulation textähnlicher Daten dar. Programme lassen sich mit Perl leicht entwickeln, da es keinen aufwendigen Compilerlauf kennt, sondern einfach das Skript einliest und ausführt bzw. bei Fehlern im Skript vor Ausführung zurückweist.

Die Programmiersprache wird gerne für komplexe Anwendungen eingesetzt, bei denen große Textdatenmengen verarbeitet werden müssen, da Perl beim Suchen und Verarbeiten von Texten eine hohe Geschwindigkeit aufweist.<sup>25</sup>

### 2.5.2.2. Java

Java zählt zu den objektorientierten und imperativen Programmiersprachen. Hierbei ist es wichtig, strikt zwischen der Programmiersprache Java und der Skriptsprache JavaScript zu trennen. Letztere benutzt zwar ähnliche Syntax, wird aber hauptsächlich in HTML-Dokumente eingebettet und bietet zudem einen wesentlich geringeren Befehlsumfang.

Was Java im Gegensatz zu anderen Programmiersprachen besonders macht ist, dass die erzeugten Programme plattformunabhängig funktionieren.<sup>26</sup>

### 2.5.2.3. PHP

PHP (rekursives Akronym für PHP: Hypertext Preprocessor) ist eine interpretierte, serverseitige Skriptsprache, welche speziell für die Webprogrammierung geeignet ist und in HTML eingebettet werden kann.<sup>27</sup>

Die Programmiersprache ist weit verbreitet, was unter anderem auch auf die freie Verfügbarkeit für den allgemeinen Gebrauch zurückzuführen ist.

---

<sup>25</sup>vgl. [Corff \(1999\)](#)

<sup>26</sup>vgl. [Lappöhn \(2010\)](#)

<sup>27</sup>vgl. [PHP.net \(2012\)](#)

#### 2.5.2.4. ASP.NET

ASP (Active Server Pages) ist eine Technologie von Microsoft, die es Webprogrammierern erlaubt, dynamische Webseiten zu schreiben.<sup>28</sup>

Wie bei PHP wird der Programmcode meist in HTML-Dateien eingebettet. Die Anwendungen können in jeder Programmiersprache geschrieben werden, die mit der CLR (Common Language Runtime) kompatibel sind. Hierunter fallen beispielsweise Visual Basic und C#.

#### 2.5.2.5. JSP

JSP (Java Server Pages) ist die Antwort von SUN auf Microsofts ASP.NET. Es handelt sich dabei um serverseitige Programmiersprachen, d.h. der Webbrowser bekommt letztendlich nur reinen HTML-Code geliefert.<sup>29</sup>

Wie bei PHP und ASP kann hier der Java-Code direkt in HTML eingebettet werden.

Java Server Pages werden beim ersten Aufruf kompiliert und in Servlets umgewandelt. Durch mehrfache Verwendung einer JSP-Komponente ergibt sich ein Performance-Gewinn, da diese nur einmal kompiliert wird. Erst wenn der Webserver neu gestartet wird, muss die Komponente neu geladen werden.

Für die Ausführung von JSP wird ein Application-Server benötigt, der die JSP-Erweiterungen unterstützt. Meistens fällt die Wahl hierbei auf den Tomcat-Server von Apache.

#### 2.5.2.6. HTML

HTML ist keine Programmiersprache, sondern eine Anwendung von SGML (Standard Generalized Markup Language). Dies ist ein System zur Definition von Dokumentenbeschreibungssprachen. HTML-Dokumente können mit jedem beliebigen Textverarbeitungsprogramm angesehen und auch erstellt werden.<sup>30</sup> Eine der grössten Vorteile von HTML ist die Plattformunabhängigkeit. HTML-Seiten lassen sich auf jedem Betriebssystem, für das ein Webbrowser verfügbar ist, anzeigen.

---

<sup>28</sup>vgl. [Falz \(2012\)](#)

<sup>29</sup>vgl. [Welk \(2012\)](#)

<sup>30</sup>vgl. [HTML \(2012\)](#)

### 2.5.3. Wahl der Datenbank und Programmiersprache

Im Rahmen dieser Arbeit habe ich mich für die Verwendung einer relationalen Datenbank und ASP.NET als Technologie mit C# als Programmiersprache entschieden.

Die Wahl fiel auf diese Komponenten, da die Datenbank SQL-fähig und leicht erweiterbar ist und die Struktur geändert werden kann, ohne Änderungen an einer bestehenden Anwendung vornehmen zu müssen. Zudem konnten im Rahmen des Studiums erste Erfahrungswerte in Bezug auf diese Datenbank vermittelt werden.

Das Erstellen einer Benutzerverwaltung wird in ASP.NET durch Steuerelemente vereinfacht. Da die Voraussetzungen für die Entwicklung mit C# unter ASP.NET durch eine bereits installierte Entwicklungsumgebung gegeben war und der dadurch verbundene Einarbeitungsfaktor gering gehalten wurde, fiel die Wahl auf diese Programmiersprache.

## 2.6. State-of-the-Art

Wie bereits im Abschnitt 2.3.3 beschrieben, existieren viele verschiedene Groupware-Lösungen.

Das Ziel der CSCW-Forschung ist das Zusammenarbeiten von Menschen durch eine informationstechnologische Unterstützung effizienter und flexibler, aber auch humaner und sozialer zu gestalten. Ein zentraler Ausgangspunkt der Forschung richtet sich dabei nach den Aufgaben und an die Personen, die diese Aufgaben durchführen und wie diese mit anderen zusammen arbeiten.

Die CSCW-Forschung unterliegt einem ständigen Wechsel. Das Aufkommen neuer Technologien und neuer Arbeits- und Gruppenkonstellationen erfordert ein ständiges Anpassen der CSCW-Systeme an die gegebenen Bedingungen.<sup>31</sup>

Die Trends der Zukunft von Groupware liegen im Bereich Cloud-Computing (siehe Glossar) und beim Einsatz mobiler Endgeräte.<sup>32</sup>

Das Cloud-Computing ermöglicht Benutzern ihre Dateien von verschiedenen Rechnern aus abzurufen, ohne dabei die Daten übertragen bzw. sich durch verschiedene Versionen ihrer Arbeiten durcharbeiten zu müssen. Hierdurch sind die Benutzer nicht mehr räumlich eingeschränkt und können jederzeit von irgendeinem Rechner der über einen Browser und Internet verfügt, ihre Arbeiten vom letzten Stand fortführen.<sup>33</sup>

Die Einbindung mobiler Endgeräte betrifft die Synchronisierung und das Weiterleiten (Push-mail) von Emails, Kontakten, Terminen und Aufgaben.

---

<sup>31</sup>vgl. (Seufert und Back, 2000, 6 f)

<sup>32</sup>vgl. PresseBox (2011)

<sup>33</sup>vgl. Ahluwalia (2012)



## 3. Konzept

Die in der Analyse erstellten Anwendungsfälle und Anforderungen bilden die Grundlage für das Konzept und werden in diesem Kapitel durch einen Systementwurf näher beschrieben. Abschliessend wird der Sicherheitsentwurf vorgestellt.

Im Text ist des Öfteren von Kompetenzen die Rede. Eine Kompetenz zu haben bedeutet, dass ein Benutzer fähig für ein bestimmtes Aufgabengebiet (Fach) ist. Hierbei bezieht sich das Aufgabengebiet auf ein eindeutiges Fach, welches sich aus einem Fachkürzel und einer Fachbezeichnung zusammensetzt. Kompetenzen implizieren im diesem Sinne keine hierarchischen Vererbungen.

### 3.1. Problemdefinition und Anforderungsanalyse

Ziel ist es, eine im WWW zugängliche Plattform zu erstellen, über die es Benutzern ermöglicht wird, Veranstaltungen zu buchen. Der Zugang zur Plattform soll durch eine Zugangskontrolle gesichert werden.

Bei der Zuordnung zu Veranstaltungen sollen Warnungen beim Benutzer angezeigt werden, sobald gewisse Rahmenbedingungen (z.B. Überschreiten der maximalen LVS) überschritten werden.

Durch einen Admin sollen neue Veranstaltungen erstellt und Benutzer eingespielt und erstellt werden können. Dieser soll im Falle eines Konfliktes (z.B. Überbuchung einer Veranstaltung) eine manuelle Zuordnung vornehmen können.

Die in der Analyse für die Groupware erstellten Anforderungsmerkmale werden in diesem Abschnitt näher erläutert.

### 3.1.1. Funktionale Anforderungen

Die wichtigsten funktionalen Anforderungen an das System stellen folgende dar:

- **Benutzerverwaltung**

Es handelt sich bei der Gruppe von Benutzern um eine autonome Einheit. Alle Benutzer, die auf dem System tätig sein werden, sind dem Admin bekannt.

Der Admin soll als einziger die Möglichkeit haben, neue Benutzer anzulegen.

Personen die als Benutzer auf dieser Plattform registriert werden möchten, sollen sich beim Admin über eine Email anmelden oder werden bei Vorliegen ihrer Daten automatisch von ihm registriert und erhalten eine Einladung per Email.

Beim Erstellen eines Benutzers, kann der Admin die Daten der jeweiligen Person eingeben. Hierzu zählen die persönlichen Daten wie z.B. der Name, die maximale Anzahl der LVS und die Rolle, die er ihm System übernehmen wird.

Durch die Registrierung eines Benutzers werden dem neuen Benutzer die Zugangsdaten durch eine Email zugestellt. Dieser soll dann die Möglichkeit haben, sein Passwort den Regeln entsprechend ändern zu können.

Tritt der Fall ein, dass der Benutzer sein Passwort vergessen hat, sollte er dies nur über den Admin wiedererlangen können.

- **Erstellen und Verwaltung von Veranstaltungen**

Das webbasierte Informationssystem soll dem Admin ermöglichen, alle für eine Veranstaltung benötigten Informationen über eine einheitliche Eingabemaske zu erfassen. Es soll ebenso möglich sein neben der Erfassung einer einzelnen Veranstaltung eine Liste von Veranstaltungen einspielen zu können. Hierbei muss gewährleistet sein, dass auch alle Informationen aufnehmbar sind, die in den aktuell gültigen Normen gefordert werden.

Beim Erstellen einer Veranstaltung soll dem Admin die Möglichkeit gegeben werden, aus bereits erfassten Veranstaltungen Informationen für eine neu zu erstellende Veranstaltung wieder verwenden zu können.

Die Erstellung von Veranstaltungen basiert auf SWS und soll während der Speicherung auf LVS umgerechnet werden. Jede Vorlesung hat standardmäßig eine SWS wohingegen Praktikas und Übungen standardmäßig drei SWS haben sollen.

Das Bereitstellen (aktivieren) von Veranstaltungen sollte separat ausgelöst werden können, um die Informationen den Benutzern erst dann sichtbar zu machen können, wenn es nötig bzw. gewünscht ist.

- **Buchung und Zuordnung von Veranstaltungen**

Die Benutzer des Systems sollen die Möglichkeit haben, Veranstaltungen buchen zu können. Hierfür soll zum Einen gewährleistet sein, dass auch nur jene Buchungen

für Veranstaltungen getätigt werden können, die deren Kompetenzen entsprechen und zum Anderen die Möglichkeit gegeben werden, auch Veranstaltungen aus kommenden Semestern vorzumerken. Alle gebuchten bzw. angefragten Veranstaltungen sollen dem Benutzer in einer Übersicht dargestellt werden.

Zudem sollen die wählbaren Veranstaltungen erst dann dargestellt werden, wenn die Lehrenden hierzu eine Kompetenz angegeben haben. Dies hat den Vorteil, dass jeder Professor eine übersichtliche, auf ihn zugeschnittene Untermenge der Veranstaltungen eines Semesters vorliegen hat.

Durch das Buchen einer Veranstaltung soll der persönliche Workload (siehe Glossar) des jeweiligen Professors addiert werden und im Falle einer Überschreitung, eine Warnung an den Benutzer erstellt werden.

Dem Administrator hingegen soll es möglich sein, existente Veranstaltungen an Professoren zuordnen zu können, die eine entsprechende Kompetenz aufweisen. Zudem soll der Admin bei der Zuordnung eines Benutzers zu einer Veranstaltung auch den Überblick bei der allgemeinen Auslastung sämtlicher Veranstaltungen behalten können.

- **Speicherung der Daten in einem relationalen Datenbankverwaltungssystem**

Die Daten aus dem System sollen in einer Datenbank gespeichert werden und jederzeit abrufbar sein.

Das System soll nur einen Teil des Stundenplaners darstellen und aus diesem Grunde auch erlauben, bestimmte Daten aus der Datenbank über Schnittstellen zu extrahieren.

- **Darstellung einer Veranstaltungs-Historie**

Semester, bei denen bereits alle Veranstaltungen zugeteilt wurden, sollen weiterhin einsehbar sein. Es soll ersichtlich sein, wem welche Veranstaltung zugesprochen wurde. Dies sollte Semesterübergreifend geregelt werden können.

- **Benachrichtigungen per Email**

Verschiedene Aktionen im System sollen durch einen Email-Versand unterstützt werden, hierzu zählen die Registrierung von neuen Benutzern, das Versenden von neuen Passwörtern und der Hinweis auf eine bald endende Anmeldefrist bzw. das Ende der Anmeldefrist.

Weitere Benachrichtigungen, welche die Kommunikation zwischen den Benutzern ermöglicht, muss nicht zwingend über Email geschehen, sollte aber gewährleistet sein.

### 3.1.2. Qualitätsbezogene Anforderungen

Wie in Abschnitt 2.4.2 erläutert, gibt es verschiedene Merkmale von Softwarequalität. Hier wird nun speziell auf die aus der Aufgabenstellung hervorgehenden Qualitätsanforderungen eingegangen.

- **Richtigkeit**  
Es soll sichergestellt werden, dass eingeloggte Benutzer nur den Bereich einsehen können, der ihnen zugewiesen wurde. Hierbei ist eine Unterscheidung zwischen Admin und Professoren vorzunehmen.  
Zudem ist wichtig, dass alle Veranstaltungen, die den gewählten Kompetenzen entsprechen, in der Auswahl der zu buchenden Veranstaltungen aufgelistet werden.
- **Sicherheit**  
Das Websystem soll vor unerlaubtem Zugriff geschützt werden. Hierfür ist ein kennwortgeschützter Login-Bereich erforderlich.
- **Verständlichkeit, Bedienbarkeit, Erlernbarkeit**  
Das Websystem sollte intuitiv bedienbar sein und den Benutzer einen Leitfaden für die Benutzung mitgeben. Unbekannte Arbeitsschritte sollten durch angegebene Hilfestellungen leicht zu erlernen sein.  
Mehrsprachigkeit ist nicht erforderlich. Der Text im System soll in der englischen Sprache verfasst werden.
- **Angemessenheit**  
Das System soll kein Stundenplaner darstellen, sondern ein Course/Teacher-Scheduling (Zuordnung Veranstaltung/Lehrende). Dabei soll das System nicht selbst die Zuordnung übernehmen, sondern unterstützend bei der manuellen Zuordnung wirken, so dass bestimmte Regeln eingehalten werden können. Zu diesen Regeln zählen unter anderem das Einhalten des persönlichen Workloads und das Buchen von Veranstaltungen die bereits vorgemerkt sind bzw. das Buchen von Veranstaltungen zu denen man überhaupt die Kompetenz besitzt.
- **Interoperabilität**  
Es soll die Möglichkeit bestehen, aus einer bereits bestehenden Semesterliste die Veranstaltungen und die dazu eingetragenen Professoren zu exportieren.

Weitere Anforderungen wurden nicht genannt bzw. abgesprochen. Die Erfüllung der verbleibenden Qualitätsmerkmale ist natürlich dennoch anzustreben, da diese mit steigender Anzahl der Erfüllung die Güte des Systems verbessern würden.

### 3.1.3. Systembezogene Anforderungen

Die systembezogenen Merkmale wurden im Rahmen dieser Arbeit definiert:

- Programmiersprachen und -systeme:  
C # ASP v2.0.50727 (Microsoft Visual Studio 2005), HTML, SQL
- Datenbank Managementsystem:  
Microsoft SQL Server 2005 (Express Edition)
- Betriebssysteme und Netze:  
Microsoft Windows XP Professional (Server), Client beliebig
- Hardware:  
Workstation, PC
- Technologie:  
3-Tier

## 3.2. Zeitlicher Ablauf

Durch das Nutzen einer Deadline, kann sich das System in zwei verschiedenen Zuständen befinden:

- Zustand ausserhalb der Deadline
- Zustand während der Deadline

Die Deadline ist die Zeitspanne in der die Verteilung der Veranstaltungen vorgenommen wird. Sie wird eingeleitet durch eine Deadline-Frist und endet durch das Bestätigen der gesamten Veranstaltungszuteilung. Beides wird durch den Admin festgelegt.

Neben der Deadline-Frist existiert eine Warning-Frist, welche für das Versenden von Emails vorgesehen ist. Die Warning-Frist liegt zeitlich vor der Deadline-Frist und wird ebenfalls vom Admin festgelegt.

Die Deadline-Frist sollte vom Admin direkt nach Abschluss der Veranstaltungszuteilung für das nächste Semester neu gesetzt werden, was bis zur Warning-Frist möglich ist. Dies sollte jedoch rechtzeitig erfolgen, um den Benutzern den Ablauf der nächsten Frist rechtzeitig anzuzeigen.

Während der Deadline ist es den Benutzern nicht möglich, Änderungen an ihren bestehenden Anfragen vorzunehmen.

Die Zustände werden über die aktuelle Serverzeit und einen Statuscode geregelt. Das Setzen der Deadline-Frist, das Erreichen der Warning-Frist, das Erreichen der Deadline-Frist und das Bestätigen der Veranstaltungszuteilung bewirken hierbei die Änderung eines Statuscodes. Dadurch wird sichergestellt, dass der Programmcode für das Erreichen der Warning- und Deadline-Frist nur einmalig ausgeführt wird.

Die Abbildungen [3.1](#) und [3.2](#) veranschaulichen das Zeitverhalten in einem Sequenzdiagramm, welches in sich in mehrere Phasen unterteilt.

Der Phasenzyklus wiederholt sich dabei jedes Semester von neuem.

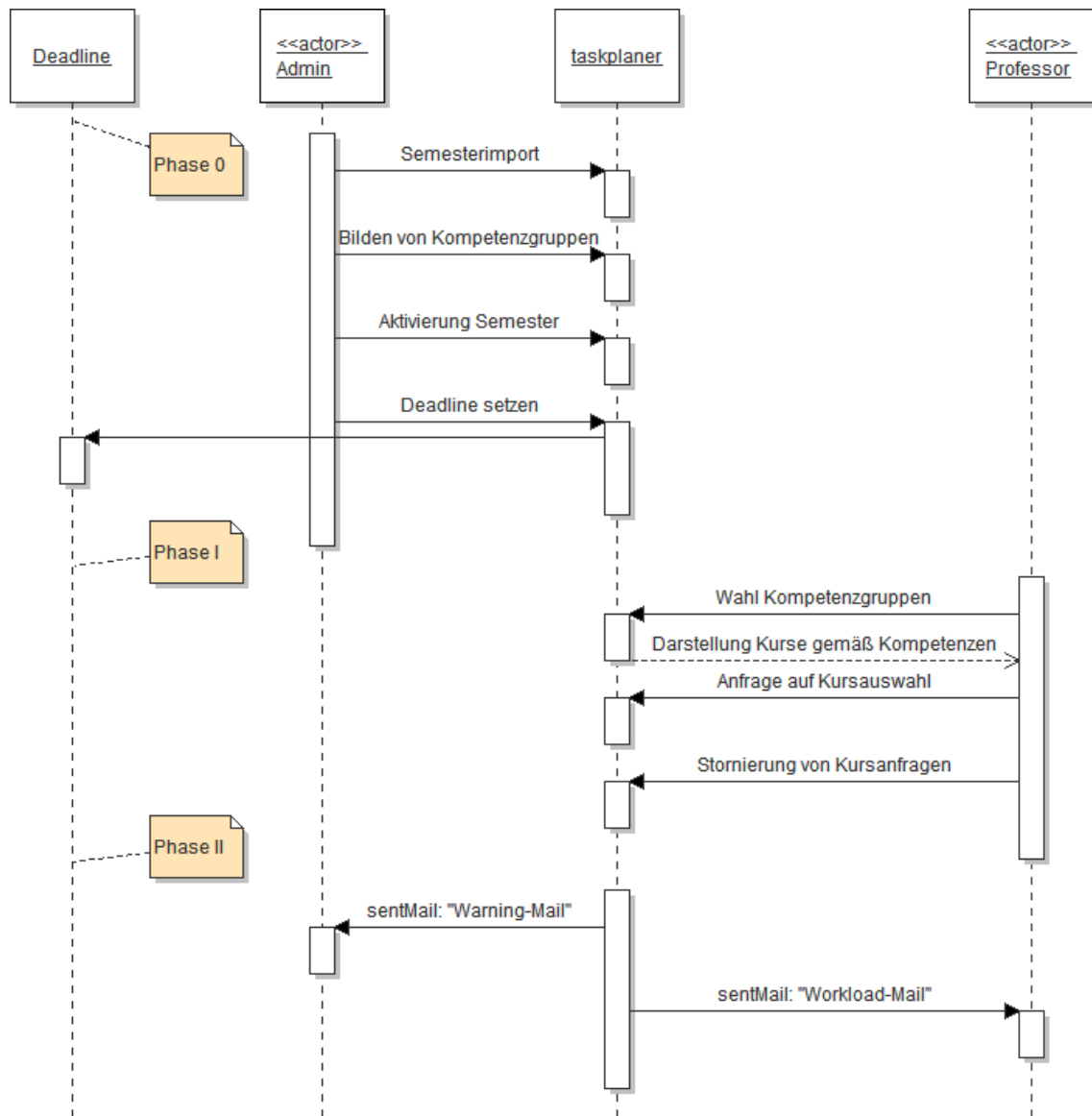


Abbildung 3.1.: Zeitlicher Verlauf des Dozentenplaners (Phase 0 bis Phase II)

Das Einspielen von Semesterlisten oder auch das manuelle Eingeben von Veranstaltungen geschieht durch den Admin.

Sobald Veranstaltungen eingespielt werden, die dem System noch unbekannt sind, wird hierfür noch keine Zugehörigkeit zu einer Kompetenzgruppe existieren. Diese wird durch den Admin vorgenommen.

Neu eingespielte Veranstaltungen werden für die Benutzer der Plattform erst durch Aktivierung des entsprechenden Semesters sichtbar.

Durch das Festlegen der Deadline und der dazugehörigen Warning-Frist, wird die erste Phase eingeleitet.

### **Phase I**

Die Benutzer der Plattform wählen Kompetenzgruppen aus, die ihren Fähigkeiten entsprechen. Hierdurch wird eine Auswahl an verfügbaren Veranstaltungen dargestellt, aus denen sie Veranstaltungen für die kommenden Semester anfragen können.

Solange sich das System im Zustand außerhalb der Deadline befindet, können bereits angefragte Veranstaltungen auch storniert werden.

### **Phase II**

Die zweite Phase wird durch das Erreichen der Warning-Frist eingeleitet.

Der Admin wird durch eine Email (*Warning-Mail*) über den aktuellen Stand der Veranstaltungen und deren Anfragen informiert.

Die Benutzer werden durch eine Email (*Workload-Mail*) über den aktuellen Buchungsstand der gewählten Veranstaltungen informiert. Des Weiteren wird ihnen Auskunft über den aktuellen Workload erteilt.



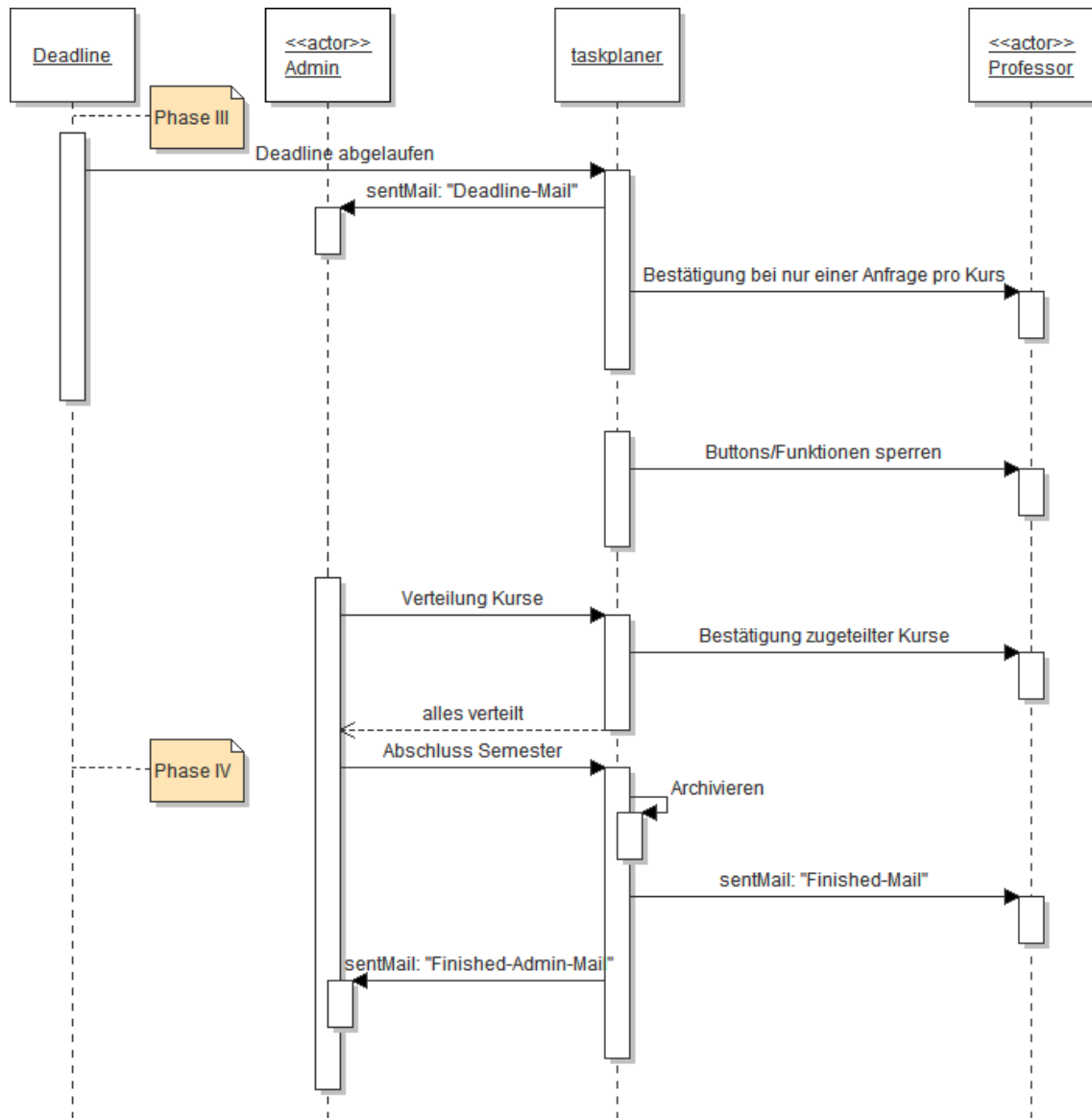


Abbildung 3.2.: Zeitlicher Verlauf des Dozentenplaners (Phase III bis Phase IV)

**Phase III**

Die dritte Phase (Zustand während der Deadline) wird durch das Erreichen der Deadline-Frist eingeleitet.

Der Admin wird per Email (*Deadline-Mail*) über den Ablauf der Deadline-Frist informiert und die Veranstaltungen, welche lediglich von einem Dozenten angefragt wurden, automatisch bestätigt.

Veranstaltungen, die gar nicht oder mehrfach gebucht wurden, werden nun durch den Admin zugeteilt. Zudem erfolgt die Sperrung bestimmter Funktionen und Buttons wie beispielsweise die Stornierung von Veranstaltungen.

**Phase IV**

Durch den vollständigen Abschluss der Veranstaltungsverteilung für das kommende Semester wird Phase IV eingeleitet.

Hierüber wird der Admin wie auch die Benutzer per EMail (*Finished-Admin-Mail* und *Finished-Mail*) informiert. Durch Abschluss des Semesters durch den Admin wird die Archivierung des aktuellen Semesters vorgenommen und Veranstaltungen des kommenden Semesters ins System eingespielt.

Auf eine detaillierte Beschreibung des Zeitverhaltens wird im Anhand D (Handbuch - Kapitel D.1.4) näher eingegangen.

### 3.3. Entwurf

Dieser Abschnitt befasst sich mit der Anwendungslogik. Durch das Definieren von Fachklassen werden Beziehungen untereinander aufgezeigt und sowohl die fachliche als auch die technische Architektur vorgestellt.

#### 3.3.1. Fachklassen

Anhand der im vorigen Kapitel (Kapitel 2.0) erwähnten Anwendungsfällen und des zu untersuchenden Beispielszenarios, werden nun die wichtigsten Klassen und deren Beziehungen untereinander aufgezeigt.

**Benutzer** (user) Der Benutzer repräsentiert einen Professor, Gastprofessor oder auch den Admin, welcher über eine eindeutige UserID identifiziert werden kann.

Der Benutzer kann Veranstaltungen, die seinen angegebenen Kompetenzen entsprechen, wählen und auch wieder abwählen. Der Admin, welcher einen Professor mit erweiterten Rechten repräsentiert, kann darüber hinaus Veranstaltungen erstellen, ändern und löschen.

**Benutzerdetails** (user\_details) Die Benutzerdetails beinhalten persönliche Angaben des jeweiligen Benutzers wie den vollständigen Namen und die Anzahl der maximalen LVS.

**Fach** (Fach) Ein Fach stellt eine eindeutige Zusammensetzung aus Fachkuerzel und Fachbezeichnung dar.

Jedes Fach kann durch den Admin einer bestimmten Gruppe Gruppe von Fächern zugeordnet werden, wodurch zusammengehörige Inhalte wie z.B. Analysis und Analysis Übungen kombiniert werden.

**Kompetenz** (Kompetenz) Eine Kompetenz entspricht bestimmten Fähigkeiten eines Professors, die zur Unterrichtung eines Fachs benötigt werden. Dem Professor werden nun Fachgruppen angezeigt, aus dem dieser nun seine favorisierten Veranstaltungen wählen kann.

**Veranstaltung** (Kurs) Eine Veranstaltung setzt sich aus einem Fach mit den dazugehörigen SWS zusammen und kann vom Benutzer ausgewählt werden. Hierdurch ändert sich der Veranstaltungsstatus von frei auf angefragt bzw. gebucht. Des Weiteren ist die Veranstaltung einer Veranstaltungsart und einer Prüfungsart zugeordnet.

**Semester** (Semesterliste) Die Semesterliste stellt die Gesamtheit aller Veranstaltungen

eines Semesters dar. Hierbei setzt sich eine Veranstaltung aus der Anzahl der LVS und der damit verbundenen Gruppenanzahl, in die sich beispielsweise ein Praktikum teilen kann, zusammen. Zudem ist die Veranstaltung einem Semester, einer Planungseinheit (z.B. *D-Inf*), einer Semestergruppe (z.B. *B-A11*) und einem Studiengang (z.B. *Bachelor*) zugeordnet.

**Nachrichten** (Messages) Auf dem Portal befindet sich ein integriertes Nachrichtensystem. Die Benutzer können über die Details einer Veranstaltung eine Liste der Benutzer abrufen, welche sich für die Veranstaltung vorgemerkt haben und Kontakt zu ihnen aufnehmen. Über das Nachrichtensystem kann aber auch direkt ein Benutzer aus der Gesamtliste aller Benutzer ausgewählt werden, um diesen zu kontaktieren. Beide Varianten ermöglichen eine optionale EMail-Benachrichtigung an den Empfänger, welche ein Hinweis über neu eingetroffene Emails liefern.

Stellt man die eben beschriebenen fachlichen Klassen in Beziehung zueinander, ergibt sich ein Klassendiagramm, welches in Abbildung 3.3 dargestellt ist.

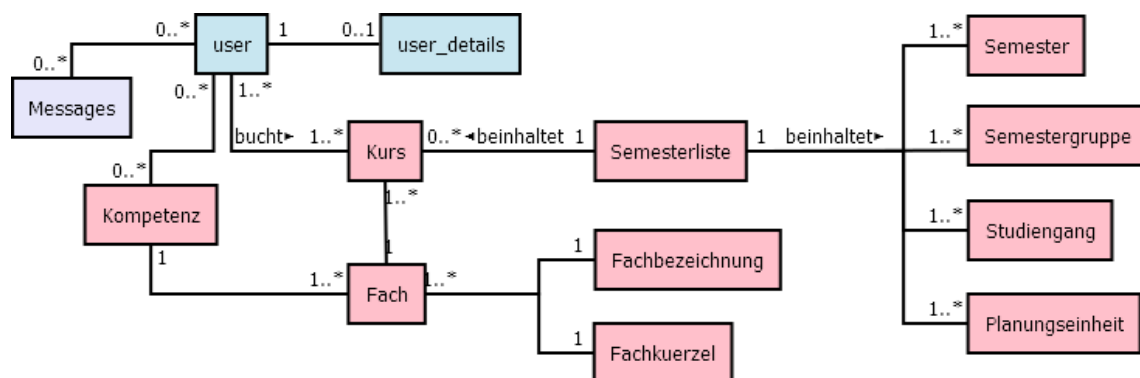


Abbildung 3.3.: fachliche Architektur

### 3.3.2. fachliche Architektur

Bei der Betrachtung der Fachklassen in Abbildung 3.3 lassen sich drei Aufgabenfelder festlegen.

Das erste Aufgabenfeld stellt die Verwaltung der Benutzer dar. Die Benutzer- und Rollenverwaltung wird zum größten Teil vom sogenannten Benutzerverwaltungssystem Membership von ASP.NET übernommen. Zu diesem Bereich zählt auch die Rolle des Benutzers, also ob es sich um einen Professor, Gastprofessor oder den Admin handelt. Ergänzende Informationen zum Benutzer werden in der Klasse `User_Details` festgehalten. Genauere Details zum Aufbau der Membership Datenbank lassen sich aus dem Anhang A (Die Datenbank des Portals) und aus dem Folgekapitel 3.3.3 entnehmen.

Das zweite Aufgabenfeld beschäftigt sich mit der Kommunikation zwischen den einzelnen Benutzern bzw. Benutzergruppen. Dies wird zum einen über die Benutzernachrichten (Messages) realisiert und zum anderen durch den Versand von Emails über den Administrator bzw. über das System unterstützt.

Das dritte Aufgabenfeld sind die Veranstaltungen. Jede Veranstaltung hat als Inhalt ein Fach, welches sich aus einem Fachkuerzel und einer Fachbezeichnung zusammensetzt. Zusammen mit der Bezeichnung eines Studienganges, Semesters, Semestergruppe und einer Planungseinheit bildet eine Veranstaltung einen Teil der Semesterliste. Die gewählte Kompetenz legt fest, welche Veranstaltungen dargestellt und somit für den Benutzer wählbar sind.

### 3.3.3. technische Architektur

Wie in Abbildung 3.4 zu sehen ist, sind alle Benutzer (Clients) mit einer zentralen Verwaltung (Server) verbunden, über die die gesamte Zusammenarbeit abgewickelt wird.

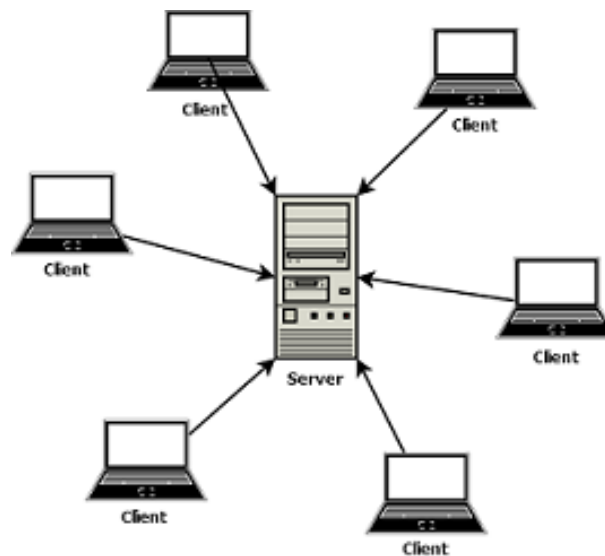


Abbildung 3.4.: Zentrale Client/Server-Architektur

Der Client baut die Kommunikation zu dem ASP.NET-fähigem Webserver über eine HTTP-Anfrage auf.

Das Kommunikationsprotokoll ist das Standardprotokoll für die Übermittlung von Webseiten und wird benutzt, um Webseiten vom Webserver anzufordern und Antworten zurück an den Webbrowser zu senden. HTTP-Nachrichten werden über den Port 80 zwischen Webserver und Webbrowser ausgetauscht.

ASP.NET ist eine Erweiterung für den Internet Information Service (IIS) und ist als eine Internet Server Application Programming Interface-Erweiterung (ISAPI) mit dem Namen `aspnet_isapi.dll` implementiert. Neben der ISAPI-Erweiterung gibt es einen ASP.NET-Arbeitsprozess (engl. Worker Process) mit dem Namen `aspnet_wp.exe`, in welchem die Webseiten kompiliert werden und ablaufen.

In Abbildung<sup>1</sup> 3.5 wird die ASP.NET-Architektur, die man auch ASP.NET Page Framework nennt, veranschaulicht.

Neben einem HTTP-Server verfügt der IIS 5.1 auch über einen SMTP-Server welcher für das Versenden von E-Mails benutzt werden kann.

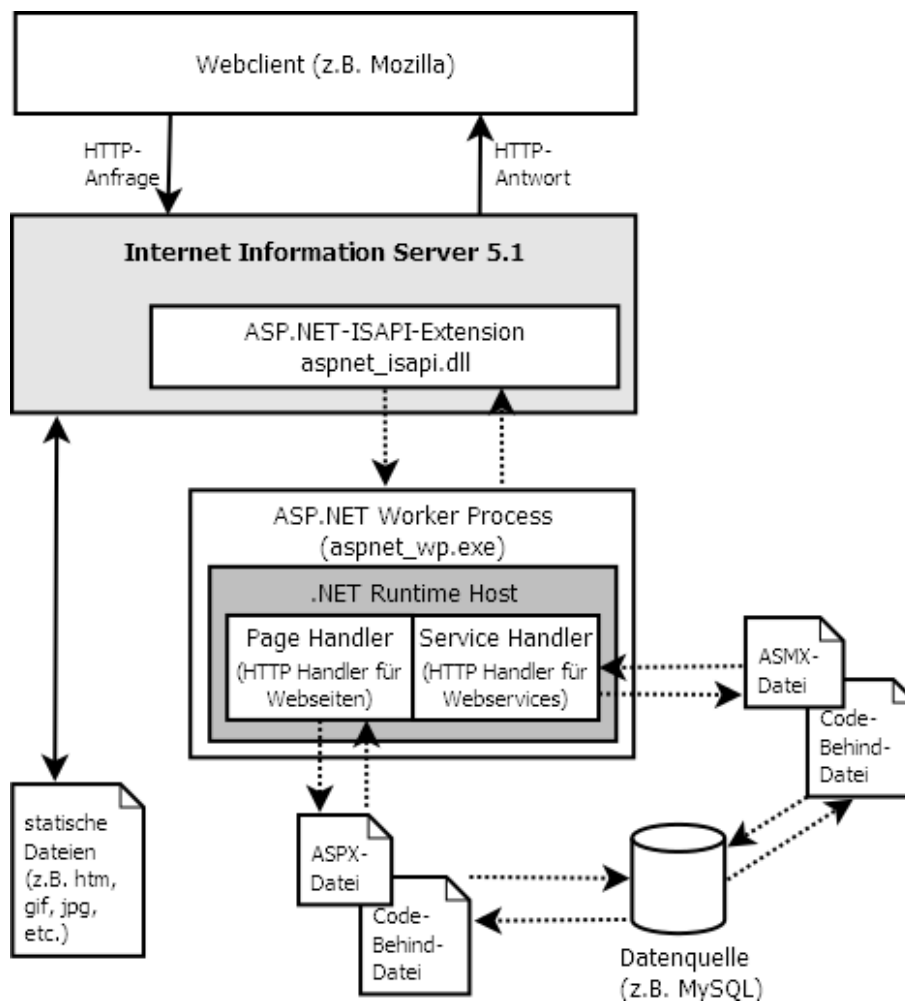


Abbildung 3.5.: ASP.NET-Architektur

<sup>1</sup>vgl. (Schwichtenberg, 2006, 985)

Die Ausführung von ASP.NET-Seiten und die Erzeugung unterscheiden sich von gängigen Webserverprogrammierungen insofern, dass sie neben HTML-Code auch spezielle Tags mit dem Attribut `runat=Server` beinhalten. Diese Tags werden als Serversteuerelemente bezeichnet und werden nicht 1:1 an den Clients übertragen. Es entscheidet sich erst zur Laufzeit auf der Serverseite, wie diese in browserspezifische Tags umgesetzt werden. Dieser Vorgang der Umsetzung wird als Adaptive Rendering bezeichnet.<sup>2</sup>

Die Architektur der Anwendung wird als 3-Schichten-Architektur geplant. Abbildung 3.6 zeigt den Aufbau der Architektur und die Einordnung der Komponenten in die entsprechenden Schichten.

Die Kommunikation zwischen der Geschäftslogik und der Oberfläche soll über Methodenaufrufe in der Geschäftslogik geschehen. Sie lädt und speichert die Daten in einer Datenbank, die neben denen für die Veranstaltungen auch die Benutzerdaten enthält, über welche die Authentifizierung gesteuert wird.

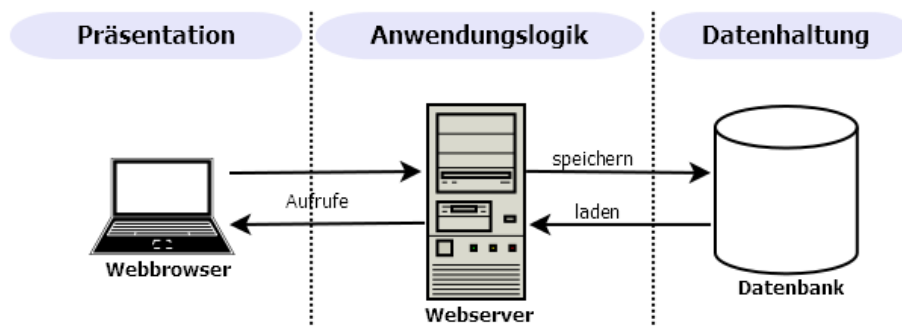


Abbildung 3.6.: 3-Schichten-Einteilung (3-Tier-Architektur)

Zur Speicherung der Benutzerdaten bietet ASP.NET ein Membership-Provider an. In ASP.NET 2.0 wird unter anderem ein Mitgliedschaftssystemprovider für Microsoft SQL Server (`SqlMembershipProvider`) mitgeliefert. Dieser wird über die `web.config`-Dateien konfiguriert.

<sup>2</sup>vgl. (Schwichtenberg, 2006, 196 ff)



Wie in Abbildung 3.7 zu sehen ist, stehen für die Provider Datenbanken im Hintergrund. Über das ASP.NET SQL-Server-Registrierungstool (aspnet\_reqsql.exe) kann man den Membership-Provider in eine bereits existierende Datenbank integrieren.

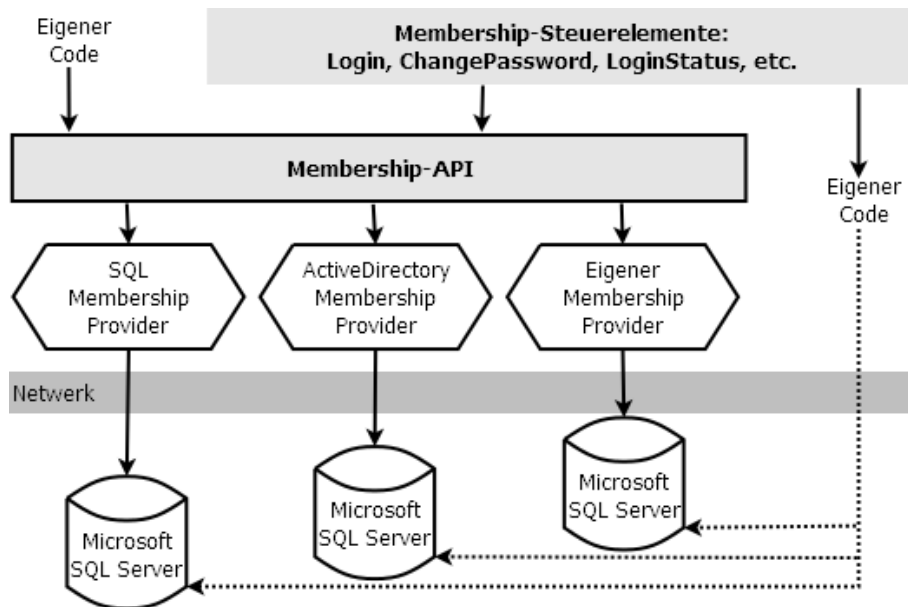


Abbildung 3.7.: Architektur des Membership-Systems

## 3.4. Sicherheitskonzept

### 3.4.1. Authentifizierung

Bei Authentifizierungsprozess wird überprüft, ob die betreffende Person tatsächlich die Person ist, die sie vorgibt zu sein. Dieser Beweis wird in der Regel durch eine Kombination aus Benutzername und Kennwort erbracht.

Die Authentifizierung in ASP.NET ergibt sich aus dem Zusammenspiel der ASP.NET Authentifizierungseinstellungen und den IIS Authentifizierungseinstellungen<sup>3</sup>. In Abbildung 3.8 werden verschiedene Möglichkeiten der Authentifizierungsverfahren aufgezeigt.

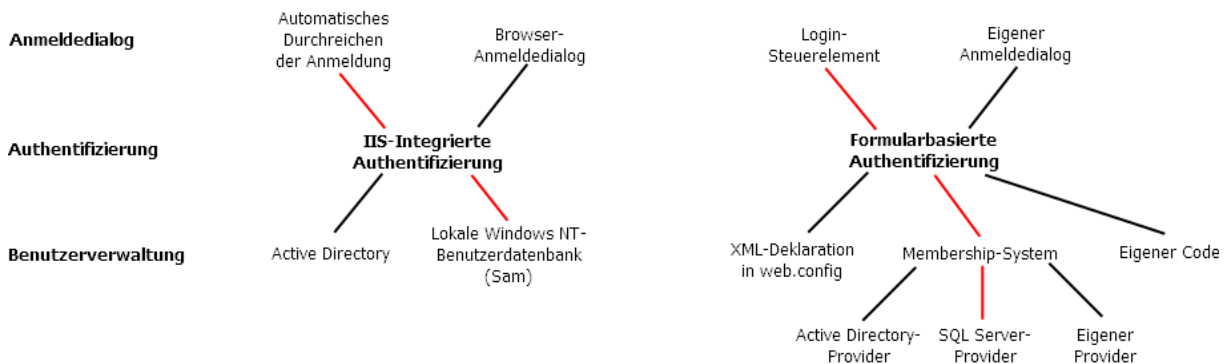


Abbildung 3.8.: Mögliche Authentifizierungsverfahren

Im Rahmen dieser Arbeit wird eine formularbasierte Authentifizierung verwendet. Der Vorteil in dieser Methode liegt darin, dass ASP.NET für dieses Verfahren bereits Klassen, Sicherheitssteuerelemente und eine einfache Möglichkeit zur Konfiguration bereitstellt. Es müssen nur geringfügige Erweiterungen an der Implementierung vorgenommen werden, um es für die Aufgabenstellung angemessen anzupassen.

<sup>3</sup>vgl. (Schwichtenberg, 2006, 596 ff)

In Abbildung 3.9 wird der Sicherheitsdatenfluss der Formularauthentifizierung grafisch dargestellt. Hierbei werden die Anmeldeinformationen von einem Benutzer direkt an die Anwendung weitergereicht, ohne dabei eine Authentizitätsprüfung beim IIS zu durchlaufen.

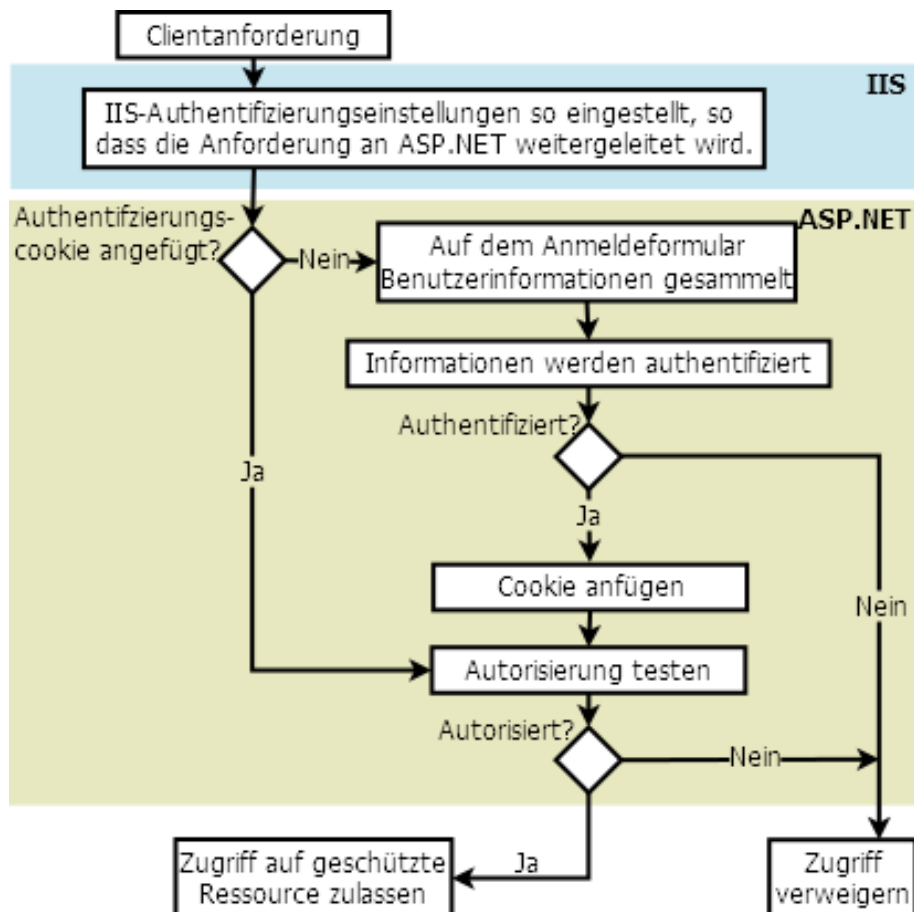


Abbildung 3.9.: Formularauthentifizierung

### 3.4.2. Autorisierung

Autorisierung bezeichnet den Vorgang, bei dem ermittelt wird, was der Benutzer machen darf bzw. worauf er zugreifen darf.

Vorraussetzung für die Zugriffskontrolle ist die vorab erwähnte Authentifizierung. Erst durch die Authentifizierung ist es möglich, eine Benutzerrolle zu identifizieren und die damit verbundenen Zugriffsrechte zu erteilen.

ASP.NET unterscheidet beim Zugriff auf eine angegebene Ressource zwischen zwei Verfahren:

- **Dateiautorisierung**

Bei der Dateiautorisierung wird über eine Zugriffssteuerungsliste der ASPX-Datei überprüft, ob ein Benutzer über Zugriffsrechte verfügt.

- **URL-Autorisierung**

Bei der URL-Autorisierung werden Benutzern und Rollen URLs zugeordnet. Hierdurch kann bestimmten Benutzern oder Rollen gezielt Zugriff auf beliebige Teile einer Anwendung gegeben werden.

In dieser Arbeit kommt die URL-Autorisierung zum Einsatz.

ASPX-Seiten die beispielsweise nur für die Rolle des Admin zugänglich sein dürfen, befinden sich in einem Unterordner, in welchen dies über eine `web.config`-Datei geregelt wird. Der Vorteil bei dieser Variante liegt neben der Übersichtlichkeit in der Zugriffsberechtigung mehrerer Dateien, die alleine über den darüberliegenden Unterordner zentral festgelegt werden können.

### 3.4.3. Netzwerk- und Datensicherheit

Ein wichtiges Merkmal von Netzwerksicherheit ist die Integrität, zu der das Verfahren zum Schutz der Dateien vor Manipulation zählt.

Insbesondere zum Schutz vor böswilligen Benutzereingaben muss sichergestellt werden, dass die Eingabedaten keinen schädlichen Inhalt aufweisen, der die Anwendung gefährden könnte.

Folgende Angriffe könnten durch böswillige Benutzereingaben ausgelöst werden:

- **Script-Injection**

Bei der Script-Injection handelt es sich um den Versuch, ein ausführbares Script an die Anwendung zu senden. Die Intention des Angreifers liegt hierbei darin, einen anderen Benutzer dieses Script ausführen zu lassen.

- **SQL-Injection**

Bei einem SQL-Injection-Angriff wird versucht die Datenbank der Anwendung zu gefährden. Hierbei werden SQL-Befehle erstellt, die zusätzlich zu den eigentlich gedachten SQL-Befehlen ausgeführt werden.

Über Validierungsmaßnahmen werden die Benutzereingaben zum größten Teil schon vorgeschrieben.

Da es dennoch vorkommen könnte, dass man verschiedene Varianten vergessen hat, gibt es die Möglichkeit, die Benutzereingaben durch die sogenannte `HtmlEncode()`-Methode zu kodieren. Hierbei wird HTML in seine Textdarstellung konvertiert und verhindert so die Ausführung des Markup im Browser. Dies kann somit Script-Injection-Angriffe vermeiden.

Um SQL-Injection-Angriffen vorzubeugen, sollte man die Gestaltung von SQL-Abfragen nicht durch Zeichenverkettungen vornehmen, da diese das Einschleusen von speziellen Zeichenketten ermöglichen. Stattdessen ist es ratsam parametrisierte Abfragen und/oder gespeicherte Prozeduren verwenden.

Um den Benutzernamen und das Kennwort für die SQL Serverinstanz nicht im Klartext abzuspeichern, bietet es sich an, die Verbindung mittels integrierter Sicherheit herzustellen. Die Verbindungszeichenfolge sollte auch nicht direkt auf einer Seite abgespeichert werden, sondern eher aus der `web.config`-Datei bezogen werden können.

## 4. Realisierung

Dieses Kapitel beschreibt die Implementierung der Anwendung anhand der 3-Schichten-Einteilung.

Eingeleitet wird dies über eine Beschreibung der grundlegenden Techniken, die bei der Lösung eingesetzt wurden.

### 4.1. Grundlegendes

Von den grundlegenden Techniken werden in diesem Abschnitt die Rollenverteilung von Benutzern, die Navigierung auf der Webseite, das Verwenden von Masterseiten und die Verwendung von Sicherheitssteuerelementen vorgestellt.

#### 4.1.1. Rollenverteilung

Die Rollenverwaltung dient zur Verwaltung der Autorisierung<sup>1</sup>.

Hierüber wird bestimmt, auf welche Ressourcen der jeweils zugeordnete Benutzer zugreifen kann. Der Ressourcenzugriff wird über Regeln in der ASP-Anwendungskonfiguration bestimmt, indem den entsprechenden Dateien bzw. Ordnern der Zugriff durch bestimmte Benutzer gewährt wird.

Für eine bessere Übersichtlichkeit wird in der Anwendung zwischen den Ordnern 'memberpages' und 'adminpages' unterschieden, die entweder durch alle Benutzer oder nur durch den Admin eingesehen werden können. Der Login-Bereich und die Startseite sind in der Ordnerstruktur eine Ebene höher angesiedelt, so dass diese durch die Zugriffsregeln der Unterordner nicht betroffen sind. Auf der Startseite wird der Zugriff über Templates gelöst, welche eine Unterscheidung zwischen den Rollen vornimmt.

#### **Admin**

Benutzer der Gruppe Admin haben uneingeschränkten Zugriff auf die Anwendung.

---

<sup>1</sup>vgl. [Microsoft \(2012e\)](#)

**Professor**

Benutzer der Gruppe Professoren verfügen über einen eingeschränkten Zugriff auf die Anwendung, hierbei wird ihnen der Ordner *adminpages* nicht angezeigt, da sie nicht autorisiert sind, diesen zu sehen.

**staff**

Dieser Bereich ist für Gastprofessoren bzw. technische Mitarbeiter gedacht. Derzeit besitzt diese Gruppe von Benutzern die gleichen Rechte wie Professoren. Anpassungen müssten an dieser Stelle noch vorgenommen werden.

### 4.1.2. Navigierung

Um eine Webnavigation verwenden zu können, wird ein Verfahren benötigt, um das Layout der Seiten in der Webseite beschreiben zu können. Dieses wird in ASP durch eine Siteübersicht<sup>2</sup> (siehe Glossar) realisiert, welche als XML-Datei in das Projekt eingebunden werden muss (Details siehe Kapitel 4.2.1.2). Diese Datei enthält die Seitentitel, die entsprechenden URLs und optional auch eine Rollenzuteilung, die durch das Rollenmodell unterstützt wird. Das Treeview-Steuerelement (siehe Glossar) wird als reduzierbares Navigationsmenü genutzt. Das Sitemappath-Steuerelement (siehe Glossar) wird als Navigationsanzeige verwendet, welches den Benutzern den Pfad bis zurück zur Startseite anzeigt.

### 4.1.3. Masterseite

Masterseiten erleichtern die einheitliche Gestaltung einer Website. Eine Masterseite hat ein vordefiniertes Layout, welches statischen Text, HTML-Elemente und Serversteuerelemente enthalten kann<sup>3</sup>. Sobald eine Seite vom Benutzer angefordert wird, wird die geforderte Inhaltsseite mit der dazugehörigen Masterseite zusammengeführt. Hierbei besitzt die Masterseite `ContentPlaceHolder`-Steuerelemente, welche mit den zugehörigen `Content`-Steuerelementen der Inhaltsseite vereint werden.

Layoutänderungen können somit zentral auf der Masterseite vorgenommen werden und betreffen alle Seiten, die diese nutzen. Die zuvor erwähnten Navigations-Steuerelemente (Treeview & Sitemappath) sind beispielsweise auf der Masterseite definiert worden.

---

<sup>2</sup>vgl. [Microsoft \(2012d\)](#)

<sup>3</sup>vgl. [Microsoft \(2012b\)](#)

#### 4.1.4. Sicherheit

Das Portal ist durch einen Login-Bereich geschützt, zu dem nur authentifizierte Benutzer Zugriff haben. Dies ist wichtig, da der Zugriff auf das Portal über das WWW möglich sein soll und dies nur einer separierten Gruppe von Benutzern vorbehalten ist.

*ASP.NET-Webserver-Steuererelemente sind Objekte auf ASP.NET-Webseiten, die beim Anfordern der Seite ausgeführt werden und Markup an einen Browser ausgeben.<sup>4</sup>*

Durch die Nutzung des ASP.NET-Anmeldesteuerelementes wird der Zugriff über die Authentifizierung, Mitgliedschaft und Rollen gesteuert. Der Vorteil bei der Nutzung von Steuererelementen liegt dabei darin, dass nur wenig oder gar kein zusätzlicher Code geschrieben werden muss.

Das Anmeldesteuerelement unterteilt sich in fünf Unterelemente:

- CreateUserWizard-Steuererelement
- Login-Steuererelement
- PasswordRecovery-Steuererelement
- ChangePassword-Steuererelement
- Register-Steuererelement

Das CreateUserWizard-Steuererelement umfasst Informationen von einem Benutzer und erstellt dann im ASP.NET-Mitgliedschaftssystem ein Konto für diesen.

Um die Anzahl und Art der vorgegebenen Textfelder bei einem Steuererelement erweitern zu können, muss das Steuererelement in eine bearbeitbare Vorlage konvertiert werden.

Das PasswordRecovery- und Register-Steuererelement befindet sich im Normalfall immer zugänglich neben dem Login-Steuererelement, um dem Benutzer die Möglichkeit zu geben, sein vergessenes Passwort erneut zuschicken zu lassen oder sich auf der Website zu registrieren. Da das Anlegen von neuen Benutzern wie auch das Anfordern eines neuen Passwortes in diesem Portal aber nur dem Admin vorbehalten ist, wurde die Vorlage des Steuererelementes entsprechend angepasst. Durch Angabe des Benutzernamens beim Admin kann dem jeweiligen Benutzer ein neues Passwort per Email zugestellt werden.

Das ChangePassword-Steuererelement ist jedem authentifizierten Benutzer zugänglich und ermöglicht eine Änderung des aktuellen Passwortes durch die Eingabe des vorherigen.

---

<sup>4</sup>vgl. [Microsoft \(2012c\)](#)



## 4.2. Implementierung des Portals

In diesem Abschnitt wird auf die Umsetzung der Implementierung des Portals eingegangen. Um den Rahmen dieser Arbeit nicht zu sprengen, wird nur auf einige Punkte eingegangen. Dieses Kapitel soll helfen, einen Eindruck von den prinzipiellen Funktionalitäten und Vorgehensweisen bei der Programmierung des Portals zu erhalten.

Das entwickelte System unterstützt eine asynchrone Zusammenarbeit über eine räumliche Distanz. Auch zentrale Aspekte von Groupware wurden bei der Implementierung berücksichtigt. Der Awareness-Aspekt, der sich in der Liste der wählbaren Veranstaltungen wiederfindet, wird durch das Anzeigen der Personen realisiert, die bereits Anfragen auf einen Kurs gestellt haben. Auch die in der Vergangenheit bereits erfolgten Veranstaltungszuteilungen können über jeden Benutzer eingesehen werden. Der WYSIWIS-Aspekt ist ebenfalls in der Veranstaltungliste wiederzufinden. Hier können alle Benutzer mit gleichen Kompetenzen dieselben Kurse und bereits bestehende Anfragen hierauf einsehen.

### 4.2.1. Applikationsschicht

Für die Implementierung des Websystems kam die Programmiersprache C# in ASP.NET unter Verwendung von Microsoft Visual Studio 2005 zum Einsatz.

ASP.NET ist eine serverseitige Technologie von Microsoft, die es ermöglicht, Webanwendungen und Webseiten zu erstellen. Dabei besteht die Option, zwischen verschiedenen Programmiersprachen wie beispielsweise JSharp (J#), VB.NET oder auch CSharp (C#) auszuwählen. Die Wahl fiel auf C#, weil diese der Syntax von Java nahekommt und mir damit nicht unbekannt war.

Um ASP.NET für die Erstellung von Webanwendungen nutzen zu können, wird ein Webserver benötigt. Hierfür kam der Microsoft Internet Information Server (IIS) 5.1 zum Einsatz (siehe auch Abbildung 3.5).

In den folgenden Abschnitten wird neben der Verzeichnisstruktur und der Konfiguration der Webanwendung auch auf die Datenbankkommunikation und einige Sicherheitsmechanismen eingegangen.

#### 4.2.1.1. Verzeichnisstruktur

Für ein effizientes Arbeiten in der Entwicklungsumgebung ist eine gut strukturierte Gestaltung der Dateien unerlässlich.

ASP.NET bietet bei der Entwicklung von Webanwendung eine bestimmte Verzeichnisstruktur zur Ablage der Dateien an. In Abbildung 4.1 wird die verwendete Struktur dargestellt.

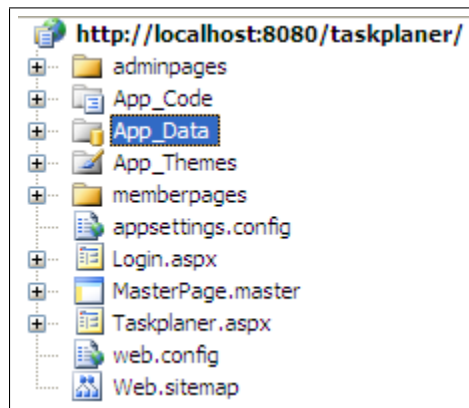


Abbildung 4.1.: Verzeichnisstruktur

**App\_Code** enthält den Quellcode für Klassen, die in der Anwendung kompiliert werden.

**App\_Data** enthält die Anwendungsdateien (Email-Vorlagen und hochgeladene csv-Dateien)

**App\_Themes** enthält die Dateien, die das Aussehen der Webseite und Steuerelementen definieren (.css-Dateien und Bilder)

**memberpages** enthält die Seiten für die Benutzer mit der Rolle Professor

**adminpages** enthält die Seiten für die Benutzer mit der Rolle Admin

Neben den dargestellten Verzeichnissen existieren noch weitere Verzeichnisse, wie beispielsweise *App\_GlobalResources* (wird bei der Lokalisierung verwendet), welche in dieser Arbeit nicht zum Einsatz kommen.

#### 4.2.1.2. Konfiguration

Webseiten werden mit einer Hierarchie aus XML-Konfigurationsdateien konfiguriert. Diese Hierarchiestruktur wird in Abbildung 4.2 dargestellt.

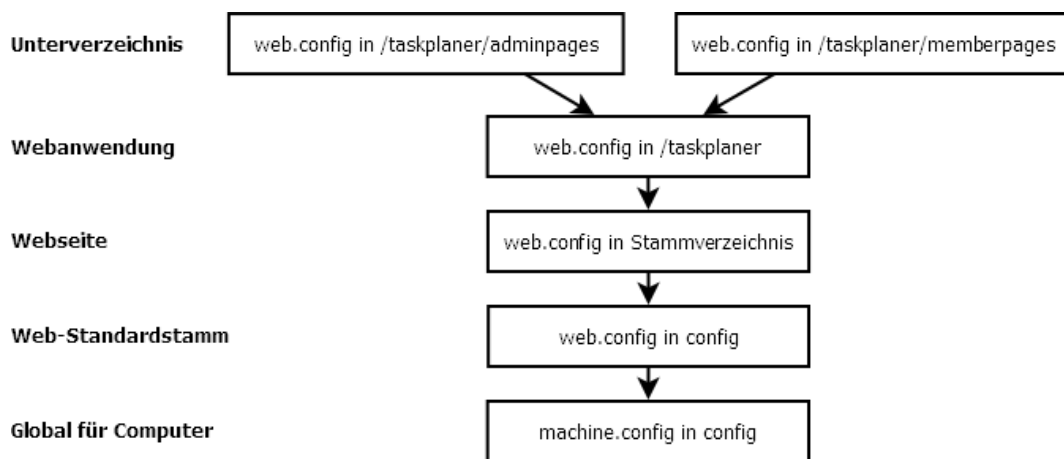


Abbildung 4.2.: Hierarchie der web.config

Die erste Konfigurationsdatei ist die Datei `machine.config`. Die Einstellungen gelten global für den Computer. Einige Dateien der `machine.config` können von den Einstellungen der `web.config` überschrieben werden. Jene die nicht überschrieben werden können, gehören zum .NET-Framework und sind daher geschützt.

Die `web.config`-Stammdatei befindet sich im selben Verzeichnis wie die `machine.config` und enthält Einstellungen für den Webserver.

Die `web.config` im Stammverzeichnis der Webseite enthält für die Webseite spezifische Einstellungen, um diese von anderen Webseiten trennen zu können.

In der Webanwendung befindet sich eine `web.config`-Datei, die weitere Einstellungen enthalten kann und die vorigen überschreibt. Auf diese wird im nächsten Abschnitt näher eingegangen.

Letztendlich kann auch jedes Unterverzeichnis einer Webanwendung optional eine `web.config`-Datei enthalten, in der nur eine Untermenge der Einstellungen gültig ist. Über diese werden speziell die Berechtigungen für die jeweiligen Benutzerrollen gesetzt. Im vorigen Abschnitt konnte man diese in der Verzeichnisstruktur unter den Namen `adminpages` und `memberpages` sehen.

Auf Ebene der Webanwendung besitzt die XML-Konfigurationsdatei die Hauptelemente `appSettings`, `configSections`, `connectionStrings`, `system.net` und `system.web` (siehe Abbildung 4.3).

```
<?xml version="1.0"?>
<configuration>
  <appSettings configSource="appsettings.config"/>
  <configSections>
    ...
  </configSections>
  <connectionStrings>
    <remove name="LocalSqlServer"/>
    <add name="LocalSqlServer" connectionString="Data Source=PC180444111268\SQLEXPRESS;Initial Catalog=TaskPlanner;Integrated Security=True" providerName="System.Data.SqlClient"/>
    <add name="TaskPlannerConnectionString" connectionString="Data Source=PC180444111268\SQLEXPRESS;Initial Catalog=TaskPlanner;Integrated Security=SSPI" providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <system.web>
    ...
    <globalization uiCulture="en-US" culture="en-US"/>
    <authorization>
      <allow roles="admin" />
      <allow roles="staff" />
      <allow roles="professor" />
      <deny users="*" />
      <deny users="*" />
    </authorization>
    <authentication mode="Forms">
      <forms name="taskplanner" defaultUrl="Taskplanner.aspx" protection="All" slidingExpiration="true" timeout="30" loginUrl="login.aspx"/>
    </authentication>
    <sessionState mode="InProc" cookieless="UseCookies" timeout="20" />
    ...
    <customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.htm">
      <error statusCode="403" redirect="NoAccess.htm"/>
      <error statusCode="404" redirect="FileNotFound.htm"/>
    </customErrors>
    <membership defaultProvider="SqlProvider" userIsOnlineTimeWindow="20">
      <providers>
        <clear />
        <add name="SqlProvider" type="System.Web.Security.SqlMembershipProvider" connectionStringName="LocalSqlServer" enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="false" passwordFormat="Hashed" requiresUniqueEmail="true" applicationName="/" />
      </providers>
    </membership>
    <siteMap defaultProvider="XmlSiteMapProvider" enabled="true">
      <providers>
        <add name="XmlSiteMapProvider" description="Default SiteMap provider." type="System.Web.XmlSiteMapProvider" siteMapFile="Web.sitemap" securityTrimmingEnabled="true"/>
      </providers>
    </siteMap>
  </system.web>
  <system.net>
    <mailSettings>
      <smtp deliveryMethod="Network" from="dozentenplaner@yahoo.de">
        <network host="smtp.mail.yahoo.de" port="25" password="xx" userName="xx" />
      </smtp>
    </mailSettings>
  </system.net>
</configuration>
```

Abbildung 4.3.: Aufbau der XML-Konfigurationsdatei

Innerhalb des `appSettings`-Konfigurationselements können beliebige Variablen in Form von Key/Value-Paaren hinterlegt werden, die dort zur Laufzeit gelagert und abgerufen werden können. Durch das Schreiben von Wertepaaren in die `web.config`

würde ein Neustart der Anwendung verursacht werden. Um dies zu umgehen, ist innerhalb des `appSettings`-Konfigurationselements ein Verweis auf eine externe `appsettings.config`-Datei erstellt worden, was einen schreibenden Zugriff auf die `web.config`-Datei verhindert.

Im Rahmen dieser Arbeit wurden in der Datei unter anderem die Zeitwerte für die Deadlines, das aktuelle Semester und weitere Einstellungen hinterlegt.

In dem `configSections`- und `system.web`-Konfigurationselement wurden Einstellungen vorgenommen, um AJAX für die Webanwendung zu aktivieren<sup>5</sup>. Diese werden in der Webanwendung verwendet, um das Neuladen einer kompletten Seite zu verhindern.

Das `connectionStrings`-Konfigurationselement beinhaltet die Verbindungszeichenfolge, die zum Verbinden mit der Datenbank benötigt wird.

Neben den benötigten AJAX-Einstellungen enthält das `system.web`-Konfigurationselement weitere Einstellungen. Hierbei werden neben den Spracheinstellungen (`globalization`) auch Autorisierungs-, Authentifizierungs- und Sessionmangementeinstellungen definiert und die Möglichkeit für die Erstellung von `customErrors` (Fehlerbehandlung für HTTP-Statuscodes) gegeben. Auch Einstellungen für den Membership-Provider und den XmlSiteMapProvider können hier vorgenommen werden. Letzterer gibt einen Verweis auf die `Web.sitemap`, aus welcher dann die Seitenübersichtsstrukturen generiert werden.

Im letzten Hauptelement mit der Bezeichnung `system.net` wurden die Einstellungen für das Versenden von Emails vorgenommen.

Eine vollständige Darstellung der `appsettings.config`- und `Web.sitemap`-Datei sowie den `web.config`-Dateien aus der Webanwendung und den Unterordnern ist im Anhang B (Konfigurationsdateien) zu finden.

---

<sup>5</sup>vgl. [Microsoft \(2012a\)](#)

### 4.2.1.3. Datenbankkommunikation

Die Kommunikation mit der Datenbank wurde über den .NET Framework-Datenanbieter (`System.Data.SqlClient`-Namespace) realisiert. Dieser unterstützt die Entwicklung von datengesteuerten Anwendungen.

ASP.NET unterstützt zwei Arten von Datenbindungen:

- **Direkte Datenbindung:**

Die Daten werden direkt über das Attribut `DataSource` an ein Datensteuerlement übergeben.

- **Indirekte Datenbindung mit Datenquellensteuerelementen:**

Die Daten werden über ein Datenquellensteuerelement (sogenanntes `SqlDataSource`) zur Verfügung gestellt. Dabei wird über einen Verweis einem Datensteuerelement der Zugriff ermöglicht.

Im Rahmen dieser Arbeit wurde überwiegend eine direkte Datenbindung implementiert. Diese wird in Abbildung 4.4 anhand eines Beispielscodes erläutert.

```
SqlConnection con;
SqlDataAdapter da;
DataSet ds = new DataSet();
try{
    con = new SqlConnection(ConfigurationManager.ConnectionStrings["TaskPlannerConnectionString"].ConnectionString);
    SqlCommand cmd = new SqlCommand("mem_semester_bindSemesterListe", con);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("@PageIndex", SqlDbType.Int).Value = profListe.CurrentPageIndex;
    ...
    cmd.Connection = con;
    da = new SqlDataAdapter(cmd);
    da.Fill(ds);
    con.Open();
    cmd.ExecuteNonQuery();
    ...
    DataView dv = new DataView(ds.Tables[0]);
    profListe.DataSource = dv;
    profListe.DataBind();
    ...
} catch (Exception e){
    e.Message.ToString();
} finally{
    if (con.State == ConnectionState.Open){
        con.Close();
    }
}
```

Abbildung 4.4.: Ausführung einer Prozedur mit anschließender Datenbindung

Über das `Connection`-Objekt wird eine Datenbankverbindung zum SQL-Server hergestellt.

Dies geschieht durch eine Instanziierung der Klasse `SqlConnection`, wobei die Verbindungszeichenfolge für die Datenbank als Konstruktor übergeben wird. Die Verbindungszeichenfolge ist dabei, wie im vorherigen Kapitel erwähnt, in der `web.config`-Datei hinterlegt und wird durch ein Zugriff über die `System.Configuration.ConfigurationManager`-Klasse bereitgestellt.

Die Verbindung zur Datenbank wird durch die Methoden `Open()` und `Close()` hergestellt und wieder beendet.

Die im Beispielcode angezeigte Vorgehensweise hat das Ziel, die aus einer Prozedur resultierenden Datensätze in einer Tabelle (`profListe`) zu speichern. Bei dieser Tabelle handelt es sich um ein Datensteuerelement (`DataGrid`-Objekt), welches es ermöglicht, die gebundenen Daten darzustellen, zu sortieren und zu editieren.

Die vollständige Implementierung der Prozedur ist im Anhang A (Die Datenbank des Portals) aufgeführt. Das Codebeispiel und der Aufbau des dazugehörigen `DataGrids` kann in Anhang C (Beispielcode) eingesehen werden.

Für den Aufruf der Prozedur wird zuerst eine Instanziierung der Klasse `SqlCommand` vorgenommen. Diese erhält als Parameter für den Konstruktor den Prozedurnamen und die vorab instanziierte Verbindung. Da es auch die Möglichkeit gibt, eine Transact-SQL-Anweisung zu erstellen, muss über den `CommandType` festgelegt werden, dass es sich um eine Prozedur handelt. Nachdem die benötigten Parameter für die Prozedur übergeben wurden, wird das `Connection`-Objekt an das instanziierte `SqlCommand`-Objekt über die Eigenschaft `Connection` gebunden.

Die Daten können unter anderem über ein `DataSet` zum Datenverbraucher gelangen. Dieses benötigt zur Beschaffung der Daten ein `DataAdapter`-Objekt. Hierfür wird die Instanziierung der `SqlDataAdapter`-Klasse vorgenommen, wobei das `SqlCommand`-Objekt an den Konstruktor übergeben wird. Nachdem über die Methode `Fill()` das `DataSet` dem `SqlDataAdapter` übergeben wurde, kann die Ausführung der Prozedur beginnen, um das `DataSet` zu füllen.

Da das `DataGrid` auch seitenweises Blättern durch die einzelnen Ergebnisse ermöglicht und nicht bei jeder Seitenänderung oder -sortierung sämtliche Seiten geladen werden sollen, wird jeweils nur die aktuelle Seite geladen. Dies wird durch eine Selektion über den Seitenindex und die Seitengröße in der Prozedur bei der Abfragemenge realisiert. Allerdings wird im Falle einer gewünschten Sortierung nur innerhalb der dargestellten Datensätze sortiert und nicht Seitenübergreifend. Zu diesem Zweck kommt die Klasse `DataGridView` zum Einsatz, an welche das `DataSet` übergeben wird. Mit Hilfe dieser Datensicht, kann man dynamische Sichten auf einzelne Datentabellen definieren, wobei das Sortieren von Datensätzen möglich ist.

Zum Binden der Daten an das `DataGrid`-Objekt wird nun über dessen Eigenschaft `DataSource` das `GridView`-Objekt als Datenquelle festgelegt und über die Methode `DataBind()` gebunden.

### 4.2.2. Persistenzschicht

Im Rahmen der Implementierung wurde das fachliche Konzept aus Kapitel 3.2.2. in einen relationalen Datenbankentwurf überführt (Datenbankentwurf aus dem Anhang A zu entnehmen) und auf dem Microsoft SQL-Server 2005 umgesetzt.

Abbildung 4.5 zeigt die Benutzeroberfläche am Beispiel einer zu bearbeitenden Stored Procedure. Wie der Abbildung zu entnehmen ist, sind neben den grundlegenden Tabellen einige Sichten und Stored Procedures definiert worden.

Sichten geben einen bestimmten Blickwinkel auf eine Reihe von verknüpften Tabellen und haben einen hohen Wiederverwendungswert bei oft verwendeten Teilabfragen.

Die Namesgebung der Stored Procedures wird mit einer bestimmten Struktur geführt. Hierbei sind die mit `aspnet_` beginnenden Stored Procedures durch die automatisch generierte Benutzerverwaltung entstanden. Bei allen weiteren Stored Procedures setzt sich der Name über die Verzeichnisstruktur zusammen. Dabei bedeutet beispielsweise `mem_semester_bookEvent`, dass die Stored Procedure aus dem Bereich für Professoren (`mem_` für memberpages), der Seite Semester (`semester_` für Semester.aspx) und der Methode, die der Namensgebung (`bookEvent`) entspricht, heraus aufgerufen wurde.

Der größte Teil der Aufrufe an die Persistenzschicht entsteht durch das Ausführen (call bzw. execute) von Stored Procedures. Hierdurch ist es der Applikationsschicht möglich, über einen einzigen Aufruf einen Ablauf von Anweisungen auszuführen. Dies vermindert zum einen die Netzwerklast, da die Befehle auf den Datenbankserver ausgelagert werden, und führt zum anderen zu einer schnelleren Ausführung bzgl. vergleichbarer Einzelanweisungen. Weitere Vorteile von dieser Variante sind, dass eine bereits ausgeführte Stored Procedure die Ergebnisse im Cache speichert und bei einer erneuten Abfrage derselbigen schneller die Ergebnisse zur Verfügung stehen. Auch bei Veränderungen am Datenbankdesign müssen nur die Prozeduren am Server geändert werden und nicht die dazugehörigen Aufrufe in der Anwendung.



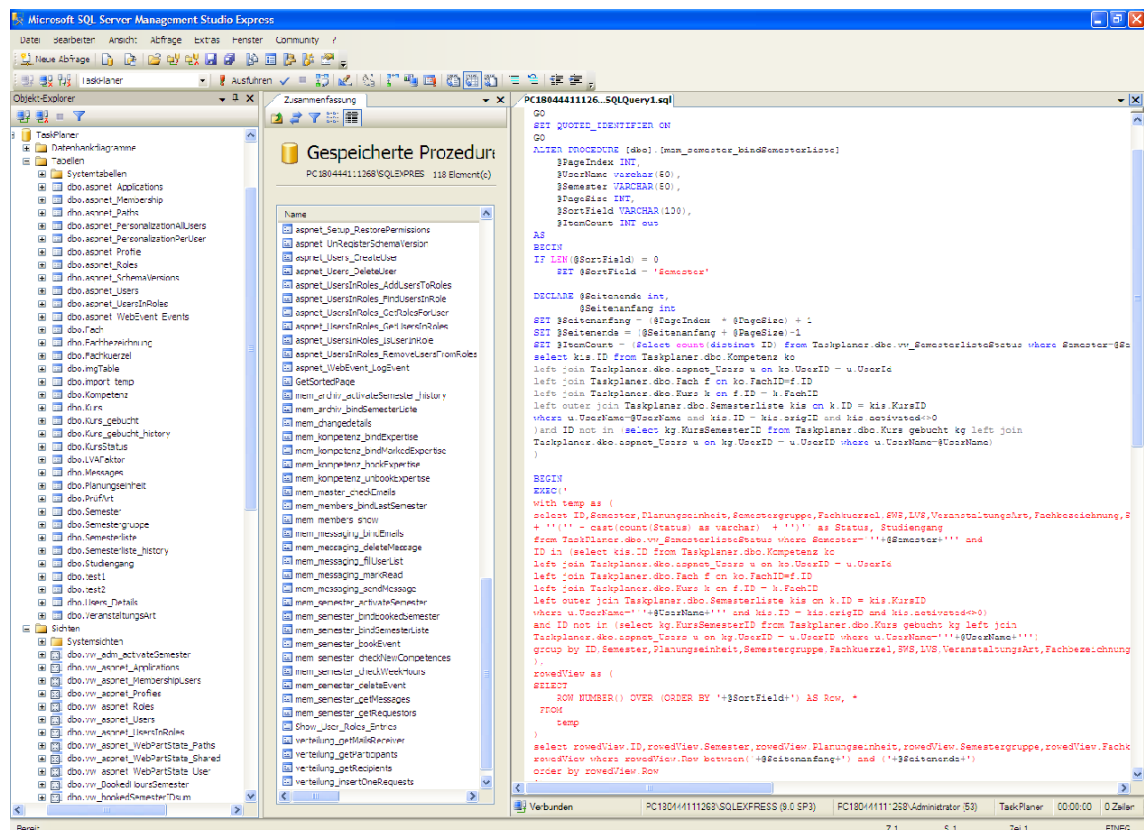


Abbildung 4.5.: Benutzeroberfläche von SQL Server Management Studio Express

### 4.2.3. Präsentationsschicht

Die Benutzeroberfläche wird für das System durch einen beliebigen Browser dargestellt. Auch durch die Nutzung eines mobilen Endgerätes können viele Teile der Oberfläche dargestellt werden. Allerdings werden durch die geringe Auflösung einige Bereiche nicht mehr visualisiert, was aber über die Auswahl der Spalten teilweise kompensiert werden kann. In Kapitel 3.3.3 wurde bereits darauf eingegangen, wie der Browser die Daten vom Server geliefert bekommt und im Anhang D (Handbuch) wurde detailliert eine Beschreibung der einzelnen Komponenten vorgenommen und wird daher an dieser Stelle nicht weiter erwähnt.

## 5. Bewertung der Lösung

In diesem Kapitel wird zur Beurteilung des umgesetzten Systems anfangs eine Messung des Quellcodes mittels einer Umfangsmetrik vorgenommen, um sich einen Überblick über die Grösse der Anwendung verschaffen zu können.

Um abschätzen zu können, inwieweit die Umsetzungen den Vorgaben entsprechen, werden die im Konzept festgelegten Anforderungen mit dem Ist-Zustand der Realisierung verglichen. Abschließend werden die aus der Umsetzung gewonnenen Erkenntnisse dargelegt.

### 5.1. Umfangsmetrik

Über das Tool DPack<sup>1</sup>, welches in die Entwicklungsumgebung (Visual Studio 2005) integriert werden kann, kann man sich eine Statistik über das aktuelle Projekt erstellen lassen.

Hierbei werden die Anzahl der Dateien und die Gesamtanzahl der Zeilen gezählt. Letztere werden durch eine Aufteilung in Codezeilen, Kommentarzeilen und leere Zeilen untergliedert. Die Tabelle 5.1 zeigt die ermittelten Werte.

<b>Metrik</b>	<b>Anzahl</b>
Dateien	26
Gesamtzeilen	6739
Codezeilen	5567
Kommentarzeilen	846
Leere Zeilen	321

Tabelle 5.1.: Umfangsmetriken

---

<sup>1</sup>vgl. [USysWare \(2011\)](#)

## 5.2. Vergleich des Portals mit den Anforderungen

Die im Konzept ermittelten Anforderungsmerkmale wurden nicht vollständig ausimplementiert. In den folgenden Abschnitten wird hinsichtlich der funktionalen Anforderungen gezeigt, inwieweit diese fertiggestellt worden sind.

Die qualitativen und systembezogenen Anforderungen werden abschließend betrachtet.

### 5.2.1. Funktionale Anforderungen

Die in der Analyse für Informationssysteme erarbeiteten funktionalen Anforderungen wurden in dem Konzept teilweise konkreter formuliert bzw. durch für die Aufgabenstellung relevante Teile erweitert.

In diesem Abschnitt werden die aus dem Konzept erwähnten Anforderungen mit den dazugehörigen Anwendungsfällen als Vergleich zum realisierten Ist-Zustand herangezogen.

#### 5.2.1.1. Benutzerverwaltung

Die Benutzerverwaltung ist zu etwa 80% ausimplementiert und deckt den Anwendungsfall Nummer 2 ab.

Es ist dem Admin möglich, neue Benutzer wie einen Professor, Admin oder Assistenten anzulegen. Allerdings müsste die Rechtevergabe beim Assistenten noch weiter ausgebaut werden, da dieser momentan die gleichen Rechte wie ein Professor besitzt.

Die Informationen zu einem Benutzer sind derzeit rudimentär und decken die persönlichen Daten wie Vor- und Nachname, die Emailadresse und den Benutzernamen ab. Dies könnte bei Bedarf entsprechend erweitert werden.

Die Verlagerung der PasswordRecovery-Funktion auf den Adminbereich birgt eine kleine Gefahr. Wenn lediglich ein Admin auf dem System registriert ist und dieser sein Passwort vergisst, kann er dies nur über den Systemadministrator wiedererlangen. Dieser müsste hierfür direkt in der Benutzerverwaltungsoberfläche des Servers das Passwort zurücksetzen.

### 5.2.1.2. Erstellen und Verwalten von Veranstaltungen

Das Erstellen und Verwalten von Veranstaltungen umfasst die vollständige Bearbeitung der Veranstaltung von Seiten des Admins und schließt die Anwendungsfälle Nummer 1, 3 und 4 ein.

Das Einspielen von Semesterlisten (Anwendungsfall Nummer 1) ist durch die Vorlage einer .csv-Datei entstanden. Bei dieser waren Veranstaltungen, die zusammengehörig sind, wie beispielsweise Analysis (AA) und Analysis Übungen (AAÜ), auf zwei Zeilen verteilt.

Mit dieser Basis wurde die Importroutine implementiert. Hierbei werden zusammengehörige Veranstaltungen als eigenständig betrachtet und müssen durch den Admin in sogenannte Fachgruppen zusammengefasst werden. So würden die beiden Veranstaltungen AA und AAÜ mit einer Zuordnung zur Fachgruppe AA in einer Übersichtsliste auftauchen, sobald als Kompetenz die Fachgruppe AA ausgewählt wurde. Hierbei würden alle Veranstaltungen aus der Fachgruppe und aus allen Semestern dargestellt werden.

Die Liste der neu eingespielten Semesterliste wird erst durch eine Aktivierung auf der Einstellungsseite für alle sichtbar, die sich mit der entsprechenden Kompetenz eingetragen haben. Diese Veranstaltungen sind bis zu diesem Zeitpunkt nur für den Admin in der Semesterübersicht farblich markiert sichtbar.

Neben der Variante, Veranstaltungen als Liste einzuspielen, gibt es auch die Möglichkeit, Veranstaltungen einzeln zu erstellen. Hierfür wird unter anderem aus den Komponenten bereits existenter Veranstaltungen eine Vorauswahl bereitgestellt, aus dem der Admin die Komponenten für eine neue Veranstaltung zusammenstellen kann. Unter Komponenten einer Veranstaltung versteht man beispielsweise eine Planungseinheit (*unit*), eine Semestergruppe (*group*), ein Fachkürzel (*course*) oder eine Fachbezeichnung (*course name*). Auch hier muss die Veranstaltung erst über die Einstellungsseite aktiviert werden, damit diese für die Benutzer sichtbar wird. Diese Variante deckt den Anwendungsfall Nummer 3 ab.

Beide Varianten wurden komplett realisiert.

Das Löschen von Veranstaltungen (Anwendungsfall Nummer 4) kann vom Admin über die Semesterübersichtsseite vorgenommen werden. Um eine bereits gebuchte Veranstaltung zu löschen, müssen erst die Buchungen entfernt werden. Hierdurch werden die betroffenen Benutzer über eine rot markierte Buchung mit dem Status *failed* informiert. Sobald keine Buchung auf der Veranstaltung existiert, kann die Veranstaltung aus der Liste gelöscht werden.

Das Löschen von Veranstaltungen wurde vollständig implementiert.

### 5.2.1.3. Buchung und Zuordnung von Veranstaltungen

Das Buchen und die Zuordnung von Veranstaltungen beschäftigt sich mit der Zuordnung zu einer Veranstaltung durch das Auswählen über den Professor selbst (Anwendungsfall Nummer 7) und durch eine manuelle Zuordnung über den Admin.

Bei der manuellen Zuordnung über den Admin kann eine bereits gebuchte Veranstaltung einem anderen Benutzer zugesprochen werden. Dies würde das Stornieren einer bereits gebuchten Veranstaltung beschreiben (Anwendungsfall Nummer 9).

Das Stornieren einer bereits gebuchten Veranstaltung durch den Buchenden selbst (Anwendungsfall Nummer 8) wird in diesem Abschnitt ebenfalls beschrieben.

Das Wählen und Abwählen von Kompetenzen (Anwendungsfall Nummer 5 und 6) bildet die Grundlage zum Buchen für Veranstaltungen und wird daher auch vorgestellt.

Um eine Auswahl der Veranstaltungen zu erhalten, für die man als Benutzer durch seine Fähigkeiten geeignet ist, muss vorab aus den Fachgruppen die sogenannte Kompetenz ausgewählt werden. In der daraus resultierenden Veranstaltungsauswahl werden alle dazugehörigen Fachgruppen dargestellt. Durch die Wahl einer Fachgruppe werden semesterübergreifend die Veranstaltungen dargestellt, die in der Fachgruppe enthalten sind. Auf der Startseite wird der Benutzer darüber informiert, sobald neue Fachgruppen hinzugekommen sein sollten.

Das Löschen einer Kompetenz verlangt, dass hierzu keine Veranstaltung der dazugehörigen Fachgruppe gebucht wurde. Andernfalls muss die Veranstaltung zuvor storniert werden.

Das Wählen und Abwählen der Kompetenzen wurde zu 100% umgesetzt.

Das Buchen einer Veranstaltung durch einen Professor wurde vollständig umgesetzt.

Hierbei gilt allerdings wie bereits erwähnt, dass zusammengehörige Veranstaltungen einzeln betrachtet werden und entsprechend auch einzeln gebucht werden müssen.

Die Buchung einer Veranstaltung selbst gilt bis zum Ablauf der Deadline als angefragt (*requested*) und der Status ändert sich erst auf gebucht (*booked*), sobald lediglich eine Anfrage existiert. Ist dies nicht der Fall, kann der Admin den Status dieser Veranstaltung über eine manuelle Zuordnung auf *assigned* bzw. *failed* setzen.

Das Stornieren einer Veranstaltung über den Buchenden selbst wurde zu 100% ausimplementiert.

Hierbei ist es dem Professor möglich, bis zum Ablauf der Deadline die Veranstaltung jederzeit wieder abzuwählen. Sobald die Deadline allerdings erreicht ist, ist dies nur noch über den Admin möglich.

Auch das Stornieren einer Veranstaltung über den Admin wurde zu 100% ausimplementiert. Hierbei wird zwischen dem Status *requested* (vor Ablauf der Deadline) und dem Status

*booked* (nach Ablauf der Deadline) unterschieden. Vor Ablauf der Deadline kann der Admin die Veranstaltung entweder einem anderen Benutzer zuweisen oder wieder auf *unrequested* stellen. Nach Ablauf der Deadline erscheinen Veranstaltungen im Status *booked*, können jedoch in dieser Zeitspanne durch den Admin wieder freigegeben werden.

#### **5.2.1.4. Speicherung der Daten in einem relationalen Datenbankverwaltungssystem**

Die Daten werden über die Geschäftslogik an Stored Procedures in der Datenbank weitergeleitet und dort in die entsprechenden Tabellen eingefügt. Das Generieren von Schnittstellen für den Datenexport würde relativ schnell machbar sein, da hierfür nur entsprechende Sichten auf die Tabellen erstellt werden müssten.

Die Speicherung der Daten selbst wurde zu 100% implementiert, jedoch die Schnittstellen für den Datenexport noch nicht. Daher ist dieser Bereich zu etwa 90% fertiggestellt.

#### **5.2.1.5. Darstellung einer Veranstaltungs-Historie**

Die Veranstaltungs-Historie ist für jedes Semester und jeden Benutzer verfügbar und wurde im System vollständig implementiert.

#### **5.2.1.6. Benachrichtigungen per Email**

Der Bereich der Kommunikation wurde zu etwa 65% fertiggestellt.

Die Anforderungen, zu welchen die Registrierung von Benutzern, das Versenden von neuen Passwörtern und der Hinweis auf eine bald endende Anmeldefrist bzw. das Ende der Anmeldefrist zählen, wurden erfüllt.

Die Kommunikation zwischen den einzelnen Benutzern wurde über ein internes Nachrichtensystem ermöglicht, welches optional eine Email-Benachrichtigung enthält. Derzeit ist die Emailbenachrichtigung nur vom Versender zu beeinflussen. Es würde aber auch Sinn machen, eine Profileinstellung für die Benutzer zu erstellen, in der über eine Checkbox gesagt wird, ob man Email-Benachrichtigungen erhalten möchte. Dies müsste noch in das System implementiert werden.

Auch die Benachrichtigung an Benutzer, dass eine angefragte Veranstaltung abgelehnt bzw. gelöscht wurde, wird bisher nur über eine farbliche Markierung und entsprechendem Statuscode (*failed*) im Bereich der angefragten Veranstaltungen signalisiert. In diesem Fall wäre es sinnvoll, den betreffenden Benutzern eine Nachricht bzw. Email zu senden, um sie über die Änderung zu informieren.

## 5.2.2. Qualitätsbezogene Anforderungen

Dieser Abschnitt zeigt inwieweit die aus dem Konzept festgelegten Qualitätsmerkmale umgesetzt bzw. erfüllt wurden.

### 5.2.2.1. Richtigkeit

Eine Unterscheidung zwischen den Rollen Professor und Admin wurde hinsichtlich der Zugriffsberechtigungen vorgenommen. Hierbei kann ein als Professor eingeloggter Benutzer den Adminbereich nicht einsehen.

Durch das sogenannte URL-Tweaking (siehe Glossar) werden als Professor eingeloggte Benutzer erneut aufgefordert Benutzerdaten einzugeben, um sich als Admin zu registrieren. Entsprechen die eingegebenen Daten nicht dem eines Admins, wird der Zugang zum Adminbereich verweigert.

Die Darstellung von Veranstaltungen wird durch die entsprechend angegebenen Kompetenzen beeinflusst. Hierbei werden Kompetenzen über die durch den Admin erstellten Fachgruppen identifiziert.

### 5.2.2.2. Usability

Die Usability unterteilt sich nach dem im Konzept angegebenen Aspekten der Verständlichkeit des Websystems, der Einfachheit seiner Bedienung und der Erlernbarkeit der Websystem-Handhabung.

Die Verständlichkeit des Websystems wird durch eine geringe Pfadtiefe und eine begleitende Navigationsstruktur unterstützt.

Ein zentraler Hilfetext gibt Aufschluss über die einzelnen Bereiche. Auf der Startseite wird eine Übersicht der wichtigsten Wahlmöglichkeiten gegeben, bei denen eine kurze Erklärung der Funktion beigefügt ist.

Um die Einfachheit der Bedienung zu unterstützen, wurden die Inhalte auf die nötigsten Funktionalitäten beschränkt. Die Unterteilung der Seite in drei Bereiche, bei denen zwei hauptsächlich zu Übersichtszwecken gestaltet wurden, ermöglicht dem Benutzer, das Augenmerk auf die Inhaltsseite zu richten.

Die Sitemap des Websystems spiegelt den Aufbau und die Logik der Strukturierung wieder

und unterstützt damit auch die Erlernbarkeit der Handhabung. Die auf der Startseite angegebenen Hilfsschritte sind in einer Reihenfolge gegliedert, in der der Benutzer vorgehen sollte, um möglichst wenig Schritte zur Erledigung der Gesamtaufgabe machen zu müssen.

#### **5.2.2.3. Angemessenheit**

Durch einen Benutzer angefragte Veranstaltungen werden ihm in einer Tabelle zur Übersicht aufgelistet. Sobald die Summe der LVS der angefragten Veranstaltungen dem persönlichen Workload gleichkommt bzw. überschritten wird, wird vom System eine Warnung an den Benutzer dargestellt. Dieser dient nur als Hinweis und hindert den Benutzer nicht daran, weitere Veranstaltungen anzufragen.

Das Buchen von Veranstaltungen, zu denen der Benutzer keine Kompetenz angegeben hat, ist nicht möglich.

#### **5.2.2.4. Interoperabilität**

Aktuell gibt es keine Implementierung im System, die das Exportieren einer Semesterliste mit den dazugehörigen Veranstaltungen und Professoren unterstützt. Durch eine Exportfunktion im Bereich der Veranstaltungshistorie könnte dies realisiert werden. Es könnte extern über eine Sicht auf die entsprechenden Tabellen im DBMS ein Export vorgenommen werden.

#### **5.2.2.5. Performance**

Durch Lasttests mit JMeter hat sich herausgestellt, dass schon bei einer geringen Anzahl an Verbindungen ein Time-out erzeugt wird.

Beim IIS 5.1, welches mit Windows XP Professional geliefert wird, gibt es einige Einschränkungen. Eine der Einschränkungen ist, dass nur eine einzige Webseite unterstützt wird, was für diese Arbeit keine negativen Auswirkung hat. Eine weitere Einschränkung, die Lasttests nicht aussagekräftig macht, stellt die Unterstützung von nur zehn gleichzeitigen Verbindungen dar. Jede weitere Verbindung wird von XP abgelehnt.

Die Ladezeiten der einzelnen Seiten sind durch die Benutzung von ImageButtons beim ersten Aufruf verhältnismäßig lang. Sobald diese einmal geladen wurden, verkürzt sich die Ladezeit jedoch.



#### **5.2.2.6. Sicherheit**

Der Schutz vor unberechtigten Zugriff auf das Portal wird über eine Formularauthentifizierung sichergestellt. Hierbei können Benutzer nur über den Admin ins Portal eingeladen werden, um die erforderlichen Benutzerdaten zu erhalten.

Die Autorisierung stellt sicher, dass Benutzer nur Portalbereiche einsehen können, für die sie befugt sind. Diese wird durch den Admin in der Mitgliederverwaltung vorgenommen und in einer ASP.NET internen Datenbank (SQL Membership) verwaltet.

Auf zusätzliche Sicherungen wie SSL (Secure Sockets Layer), VPN (Virtual Private Network) oder auch eine zusätzliche Kennwortabfrage über den IIS wurde verzichtet. Diese müssten bei Bedarf nachimplementiert bzw. eingestellt werden.

#### **5.2.2.7. Persistenz der Daten**

Das System speichert alle Anfragen auf Veranstaltungen des aktuellen Semesters. Hierbei verfügt eine Veranstaltung über einen Buchungsstatus. Wie auch bei der Semesterliste, wird bei gebuchten Veranstaltungen eine Historie geführt. Diese ist allerdings nur für das aktuelle Semester verfügbar.

Ein regelmäßiges Backup des Systems sollte durch den Systemadministrator veranlasst werden, um die Persistenz zu gewährleisten.

### **5.2.3. Systembezogene Merkmale**

Als Programmiersprache wurde C# unter ASP.NET verwendet. Die Anfragen an die Datenbank sind mit SQL realisiert wurden. Dem Benutzer werden dynamisch erstellte HTML-Seiten präsentiert.

Als DBMS kam der Microsoft SQL Server 2005 zum Einsatz. Hierbei wurden überwiegend Stored-Procedures verwendet, um eine geringere Netzwerklast zu erlangen.

Das System arbeitet auf dem IIS 5.1 unter Windows Microsoft XP Professional.

Als Technologie wurde die Client/Server-Architektur verwendet.

### 5.3. Gewonnene Erkenntnisse

Die Erstellung des erarbeiteten Systems verfolgt im Wesentlichen das Ziel, eine Plattform für Professoren zu erstellen. Hierbei stellt die Realisierung eines Class/Teacher-Scheduling den Schwerpunkt dar. Das Ergebnis dieser Zuordnung soll später ein Teilbereich für die Erstellung eines Stundenplans darstellen.

Bei der Erstellung des Systems sollte das Zusammenspiel zwischen Räumen, Terminen und Veranstaltungen nicht weiter betrachtet werden. Hierdurch wird zum einen die Komplexität des Systems gering gehalten, aber zum anderen wird auch die Möglichkeit reduziert, auf zusätzliche Verteilungsstrategien einzugehen.

Es bleibt nur die Variante der Verfügbarkeiten und Wunschanfragen der Lehrkräfte, Priorisierungen des Admins und die Einigung der Lehrkräfte untereinander zu berücksichtigen.

Das bestehende System zeigt, dass die wesentlichen Kriterien für die Verteilung von Veranstaltungen zum größten Teil umgesetzt wurden.

Es ist in der Lage, die Wunschanfragen von Lehrkräften zu berücksichtigen, die durch das Buchen einer Veranstaltung visualisiert werden. Auch die Priorisierung des Admins, durch die eigenmächtige Zuteilung der Veranstaltungen, ist gegeben. Die Lehrkräfte einer Fachgruppe können bei der Auswahl einer Veranstaltung, bereits angefragende Lehrkräfte erkennen und diese ggf. auch kontaktieren.

Die Möglichkeit, sich als Lehrkraft beim System weniger bzw. gar nicht Verfügbar zu stellen, ist derzeit nicht ausreichend gegeben, kann aber über den Admin beispielsweise durch Minimierung des maximalen Workloads erreicht werden.

Um eine optimale Entscheidungsgrundlage für die Einsetzbarkeit von bestimmten Technologien treffen zu können, ist die Entwicklung eines rudimentären Systems sinnvoll. Hierdurch lassen sich eventuelle Schwachstellen der Technologie bereits in einer frühen Phase der Entwicklung erkennen und man kann frühzeitig nach alternativen Lösungen suchen.

Die Nutzung des IIS 5.1 ist als Testumgebung ausreichend, aber für den Einsatz auf einer Anwendung mit mehr als zehn gleichzeitig eingeloggtten Benutzern nicht brauchbar. Es müsste eine höhere Version des IIS installiert werden, um den höheren Ansprüchen gerecht werden zu können.

In Hinblick auf die Erweiterbarkeit des Systems durch Funktionalitäten, ist die Gliederung der Verzeichnisstruktur und die dokumentierte Umsetzung fördernd.

# 6. Schluss

## 6.1. Zusammenfassung

Ziel dieser Arbeit war es, ein webbasiertes Informationssystem zu schaffen, welches eine Aufgabenverteilung zwischen Personengruppen ermöglicht bzw. unterstützt. Hierbei lag der Augenmerk auf der Problemerkennung und -anzeige durch das entwickelte System, jedoch nicht zwangsläufig auf der Problemfindung.

Dazu wurde zunächst die allgemeine Aufgabenstellung anhand eines Beispielszenarios definiert und eingegrenzt. Erstellte Anwendungsfälle haben einen ersten Überblick über die zu bearbeitende Aufgabe gegeben.

Eine Groupware hat ähnliche Ziele wie das zu entwickelnde System und wurde deswegen zur Analyse herangezogen. Hierbei wurden die Systemeigenschaften und Architektur von Groupware analysiert und ein Überblick über bereits bekannte Groupwaresysteme gegeben. Die Anforderungsmerkmale für Websysteme wurden herausgearbeitet und bilden die Grundlage für die im Anschluss untersuchten Systemarchitekturen im Bereich Datenbanken und Programmiersprachen.

Abschließend wurde der Stand der Kunst in Bezug auf existente Groupware-Systeme dargestellt.

Im Konzept wurden die allgemeinen Anforderungsmerkmale konkretisiert.

Ein Entwurf mit fachlicher und technischer Architektur wurde festgelegt sowie das verwendete Sicherheitskonzept.

In der Realisierung wurden einige grundlegende Eigenschaften des Systems vorgestellt sowie das Portal an sich.

Bei der Bewertung der Lösung hat sich gezeigt, dass ein Großteil der Anforderungen umgesetzt wurde. Es erfüllt die wichtigsten funktionalen Anforderungen und lässt sich leicht durch zusätzliche Funktionen erweitern.

Eine Übersicht über den Fortschritt der funktionalen Anforderungen kann man sich in Kapitel 5.2.1 verschaffen. Insbesondere die Kommunikation zwischen den Benutzern weist noch

einige Lücken auf.

Auch die Funktionalität, einen Export über die Anwendung selbst bereitzustellen, steht noch aus.

Bei der Gestaltung des administrativen Bereichs für die Zuordnung von Benutzern zu Aufgaben wurde eine übersichtliche und funktionale Gestaltung erarbeitet (Kapitel D.1.3.10 und Kapitel D.1.3.6 geben einen Überblick).

Im Ganzen lässt sich sagen, dass das implementierte System als hilfreiche Anwendung für den geforderten Bereich zu sehen ist und dabei viel Verwaltungsarbeit, Zeit und Mühe sparen kann.

## 6.2. Ausblick und Kritik

Der Funktionsumfang des entwickelten Systems bietet zum jetzigen Zeitpunkt eine Verwaltungsmöglichkeit für Benutzer und Veranstaltungszuteilungen.

Einige Merkmale, die aus den Anforderungen ermittelt wurden, wie die Kommunikation, fehlen teilweise noch. Eine Implementierung dieser Komponenten kann noch vorgenommen werden, da diese leicht ins Konzept zu integrieren sind. Zu diesen Merkmalen zählen neben den Benachrichtigungen an den Admin (z.B. wegen Mehrfachbuchungen) auch die Benachrichtigungen an Professoren, die beispielsweise erfolgreich eine Veranstaltung zugesprochen bekommen haben.

Ebenso wären Benachrichtigungen an Benutzer, ausgelöst durch beispielsweise eine Stornierung einer Veranstaltung seitens des Admins, eine gute Erweiterung.

Weitere Groupware-Aspekte wie den Status eines Benutzers sehen zu können und die Möglichkeit direkt mit diesen in Kontakt (z.B. über Instant Messaging) treten zu können, wären eine sinnvolle Erweiterung und würden die Koordination unter den Benutzergruppen erhöhen.

Als Ergänzung zu der derzeit implementierten Lösung, dass Benutzer einer gleichen Kompetenzgruppe gemeinsam über Veranstaltungen diskutieren können, wäre eine Möglichkeit dies als Forum umzuwandeln. Dies würde die Übersichtlichkeit bezüglich der Kommunikation fördern.

Durch den Verzicht auf die Mehrsprachigkeit der einzelnen Textbausteine haben sich unerwartet Hürden aufgetan, die teilweise mühsam zu umgehen waren. Insbesondere bei der Option eine .csv-Datei einspielen zu können, gab es keine Möglichkeit den dazugehörigen Button-Text anzupassen. Dies wird derzeit durch die Nutzung von verschiedenen

Anzeigeebenen über CSS gelöst. Hierfür muss unter anderem auch bei jedem Laden der Seite geprüft werden, ob das entsprechende Element auf der Seite ist, damit dieses dann ersetzt werden kann.

# Literaturverzeichnis

- [Ahluwalia 2012] AHLUWALIA, S.: *Work together in Group on Documents From Anywhere in world on Cloud*. 2012. – URL <http://www.cloudcomputingtechie.com/work-group-together-document-anywhere-world/>. – (Zugriffsdatum: 03.04.2012)
- [Aupperle 2012] AUPPERLE, A.: *Miele vertraut auf Microsoft-Technologie*. 2012. – URL <http://www.it-director.de/nc/home/newsdetails/article/die-weichen-gestellt.html>. – (Zugriffsdatum: 03.04.2012)
- [C. Böttger 2008] C. BÖTTGER, A. R.: Neu gruppiert - Collaboration-Lösungen für KMU. In: *iX Magazin für professionelle Informationstechnik*, S. 99-105 (2008)
- [Christensen 2006] CHRISTENSEN, R. P.: *ComboBox in a DataGrid*. 2006. – URL <http://www.codeproject.com/KB/grid/RenderDataGridComboBoxColumn.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Corff 1999] CORFF, O.: *Einführung in die Perl Programmierung*. 1999. – URL <http://userpage.fu-berlin.de/corff/perl/perlkurs.html>. – (Zugriffsdatum: 03.04.2012)
- [css4you 2009] CSS4YOU: *CSS 4 You*. 2009. – URL <http://www.css4you.de/>. – (Zugriffsdatum: 29.02.2012)
- [Dierker und Sander 1998] DIERKER, M. ; SANDER, M.: *Lotus Notes 4.6 und Domino - Integration von Groupware und Internet*. Addison-Wesley, 1998. – ISBN 3-8273-1314-7
- [Dumke 2001] DUMKE, R.: *Software Engineering*. Vieweg-Verlag, 2001. – ISBN 9783528253554
- [Dumke u. a. 2003] DUMKE, R. ; LOTHER, M. ; WILLE, C. ; ZBROG, F.: *Web Engineering*. Person Studium, 2003. – ISBN 3-8273-7080-9
- [Education 2001] EDUCATION, CAS: *Campus Management System*. 2001. – URL <http://www.cas-education.de/fuer-hochschulen/cas-campus/funktionen/>. – (Zugriffsdatum: 29.02.2012)

- [Ellis u. a. 1991] ELLIS, A. ; GIBBS, S. J. ; REIN, G.: Groupware: some issues and experiences. In: *Communications of the ACM* 34(1), S. 39-58 (1991)
- [Falz 2012] FALZ, S.: *active Server Pages: FAQ*. 2012. – URL <http://www.aspfaq.de/index.asp>. – (Zugriffsdatum: 03.04.2012)
- [HTML 2012] HTML: *Was ist HTML?* 2012. – URL <http://www.htmlbasis.de/websprachen/html/html-kurs/was-ist-html.html>. – (Zugriffsdatum: 03.04.2012)
- [IBM 2012] IBM: *IBM Lotus - Kommunikationsanwendungen und Collaboration Software*. 2012. – URL <http://www-01.ibm.com/software/de/lotus/>. – (Zugriffsdatum: 25.04.2012)
- [Jeschke 2012] JESCHKE, M.: *Die beliebtesten Groupware-Lösungen*. 2012. – URL [http://www.tecchannel.de/kommunikation/e-mail/2029031/die\\_bliebtesten\\_groupware\\_loesungen/](http://www.tecchannel.de/kommunikation/e-mail/2029031/die_bliebtesten_groupware_loesungen/). – (Zugriffsdatum: 25.04.2012)
- [Kühnel 2006] KÜHNEL, A.: *Visual C# 2005*. 3. Auflage. Galileo Press, 2006. – ISBN 3-89842-586-X
- [Klinke 2011] KLINKE, M.: *Groupware in Kleinunternehmen: Eine empirische Untersuchung*. 2011. – URL <http://papercc.de/22919-groupware-in-kleinunternehmen-9783836614504#!/pages/1>. – (Zugriffsdatum: 29.02.2012)
- [Kuykendall 2012] KUYKENDALL, D.: *phpGroupware - Man Vs WebApp*. 2012. – URL <http://www.manvswebapp.com/phpgroupware>. – (Zugriffsdatum: 25.04.2012)
- [Lappöhn 2010] LAPPÖHN, J.: *IT-Basiswissen: Was ist Java*. 2010. – URL <http://suite101.de/article/it-basiswissen-was-ist-java-a70953>. – (Zugriffsdatum: 03.04.2012)
- [Lynch u. a. 1990] LYNCH, J. ; SNYDER, M. ; VOGEL, R. ; MCHENRY, K.: *The Arizona Analyst Information System: Supporting Collaborative Research on International Technological Trends*. North-Holland Publ., 1990. – ISBN 0-444-88760-1
- [MDLink 2009] MDLINK: *OpenGroupware.org*. 2009. – URL <http://www.opengroupware.org/>. – (Zugriffsdatum: 25.04.2012)
- [Microsoft 2010a] MICROSOFT: *appSettings-Element (allgemeines Einstellungsschema)*. 2010. – URL <http://msdn.microsoft.com/de-de/library/ms228154%28VS.80%29.aspx>. – (Zugriffsdatum: 29.02.2012)

- [Microsoft 2010b] MICROSOFT: *ASP.NET Themes and Skins*. 2010. – URL <http://msdn.microsoft.com/en-us/library/ykzx33wh.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010c] MICROSOFT: *Erstellen einer Website mit Mitgliedschaft und Benutzeranmeldung*. 2010. – URL [http://msdn.microsoft.com/de-de/library/879kf95c\(vs.80\).aspx](http://msdn.microsoft.com/de-de/library/879kf95c(vs.80).aspx). – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010d] MICROSOFT: *Exemplarische Vorgehensweise: Filtern von Siteübersichtsknoten auf Basis von Sicherheitsrollen*. 2010. – URL <http://msdn.microsoft.com/de-de/library/ms178429.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010e] MICROSOFT: *How to: Reference ASP.NET Master Page Content*. 2010. – URL <http://msdn.microsoft.com/en-us/library/xxwa0ff0.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010f] MICROSOFT: *How to test Windows 2000 IIS SMTP services manually*. 2010. – URL <http://support.microsoft.com/kb/286421/de>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010g] MICROSOFT: *How to use Telnet to Test SMTP Communication*. 2010. – URL <http://technet.microsoft.com/en-us/library/bb123686%28EXCHG.80%29.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010h] MICROSOFT: *Implementieren rollensbasierter Sicherheit mithilfe formularbasierter Authentifizierung in einer ASP.NET-Anwendung mit C#*. 2010. – URL <http://support.microsoft.com/kb/311495>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010i] MICROSOFT: *Programmgesteuertes Verwenden von ASP.NET-Masterseiten*. 2010. – URL <http://msdn.microsoft.com/de-de/library/c8y19k6h.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010j] MICROSOFT: *Set up Database using SQL Server 2005*. 2010. – URL <http://forums.asp.net/t/980214.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010k] MICROSOFT: *Speichern von Daten im Anwendungsbereich mit ASP.NET und Visual C #*. 2010. – URL <http://support.microsoft.com/kb/311515/de>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010l] MICROSOFT: *Upload and import CSV-File*. 2010. – URL <http://forums.asp.net/t/1208600.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Microsoft 2010m] MICROSOFT: *Verwenden von SQL Server Express mit ASP.NET*. 2010. – URL <http://msdn.microsoft.com/de-de/library/ms247257.aspx>. – (Zugriffsdatum: 29.02.2012)



- [Microsoft 2012a] MICROSOFT: *ASP.NET AJAX > Configuring ASP.NET AJAX*. 2012. – URL <http://www.asp.net/AJAX/Documentation/Live/ConfiguringASPNETAJAX.aspx>. – (Zugriffsdatum: 30.03.2012)
- [Microsoft 2012b] MICROSOFT: *ASP.NET Masterseiten*. 2012. – URL <http://msdn.microsoft.com/de-de/library/wtxbf3hh.aspx>. – (Zugriffsdatum: 03.04.2012)
- [Microsoft 2012c] MICROSOFT: *ASP.NET-Webserver-Steuererelemente*. 2012. – URL [http://msdn.microsoft.com/de-de/library/bs302eat\(v=vs.80\).aspx](http://msdn.microsoft.com/de-de/library/bs302eat(v=vs.80).aspx). – (Zugriffsdatum: 03.04.2012)
- [Microsoft 2012d] MICROSOFT: *Filtern von Siteübersichtsknoten auf der Basis von Sicherheitsrollen*. 2012. – URL <http://msdn.microsoft.com/de-de/library/ms178429.aspx>. – (Zugriffsdatum: 03.04.2012)
- [Microsoft 2012e] MICROSOFT: *Grundlegende Informationen über die Rollenverwaltung*. 2012. – URL [http://msdn.microsoft.com/de-de/library/5k850zwb\(v=vs.80\).aspx](http://msdn.microsoft.com/de-de/library/5k850zwb(v=vs.80).aspx). – (Zugriffsdatum: 03.04.2012)
- [Microsoft 2012f] MICROSOFT: *Microsoft Exchange Server 2010*. 2012. – URL <http://www.microsoft.com/exchange/2010/de/de/>. – (Zugriffsdatum: 25.04.2012)
- [Microsoft 2012g] MICROSOFT: *Sichern des Datenzugriffs*. 2012. – URL <http://msdn.microsoft.com/de-de/library/ms178375%28v=vs.80%29.aspx>. – (Zugriffsdatum: 06.05.2012)
- [Mitchell 2010] MITCHELL, S.: *Creating User Accounts: The official Microsoft ASP.NET Site*. 2010. – URL <http://www.asp.net/security/tutorials/creating-user-accounts-cs>. – (Zugriffsdatum: 29.02.2012)
- [Pawlak u. a. 2007] PAWLAK, A. ; SANDKUHL, K. ; CHOLEWA, W. ; INDRUSIAK, L.: *Coordination of Collaborative Engineering - State of the Art and Future Challenges*. Gesellschaft für Informatik (GI), 2007. – ISBN 978-3-88579-214-7
- [Pellegrin 2010] PELLEGRIN: *Violet UML Editor*. 2010. – URL <http://alexdp.free.fr/violetumleditor/page.php>. – (Zugriffsdatum: 29.02.2012)
- [PHP.net 2012] PHP.NET: *PHP: Was ist PHP*. 2012. – URL <http://php.net/manual/de/intro-what-is.php>. – (Zugriffsdatum: 03.04.2012)
- [Picand 2011] PICAND, Yann: *ISO/IEC 9126*. 2011. – URL [http://dictionary.sensagent.com/ISO/IEC\\_9126/de-de/](http://dictionary.sensagent.com/ISO/IEC_9126/de-de/). – (Zugriffsdatum: 02.04.2012)

- [PresseBox 2011] PRESSEBOX: *Die Zukunft von Groupware und Kommunikation*. 2011. – URL <http://www.pressebox.de/pressemeldungen/reddoxx-gmbh-rottweil/boxid/455744>. – (Zugriffsdatum: 03.04.2012)
- [Robillard 2004] ROBILLARD, M.: *Erstellen von benutzerdefinierten Spalten für das Datagrid-Steurelement von ASP.NET*. 2004. – URL <http://msdn.microsoft.com/de-de/library/aa479316.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Schwichtenberg 2006] SCHWICHTENBERG, H.: *Microsoft ASP.NET 2.0 mit Visual Basic 2005 - Das Entwicklerbuch*. Microsoft Press Deutschland, 2006. – ISBN 978-3860635445
- [Seufert und Back 2000] SEUFERT, A. ; BACK, A.: Computer Supported Cooperative Work (CSCW) - State-of-the-Art und zukünftige Herausforderungen. In: *HMD:Praxis der Wirtschaftsinformatik 213*, S.5-22 (2000)
- [Teufel u. a. 1995] TEUFEL, S. ; SAUTER, C. ; MÜHLHERR, T. ; BAUKNECHT, K.: *Computerunterstützung für die Gruppenarbeit*. Addison-Wesley, 1995. – ISBN 3-89319-878-4
- [USysWare 2011] USYSWARE, Inc: *DPack*. 2011. – URL <http://www.usysware.com/dpack/Default.aspx>. – (Zugriffsdatum: 29.02.2012)
- [Vogel 2006] VOGEL, J.: *Computergestützte Gruppenarbeit*. 2006. – URL <http://pi4.informatik.uni-mannheim.de/pi4.data/content/courses//2006-ss/csw/>. – (Zugriffsdatum: 29.02.2012)
- [Welk 2012] WELK, M.: *jsp-develop.de: Basics*. 2012. – URL <http://www.jsp-develop.de/basics/>. – (Zugriffsdatum: 03.04.2012)
- [Zarafa 2012] ZARAFÄ: *Zarafa*. 2012. – URL <http://www.zarafa.com/content/community>. – (Zugriffsdatum: 25.04.2012)

## **A. Die Datenbank des Portals**

# Die Membership-Datenbank

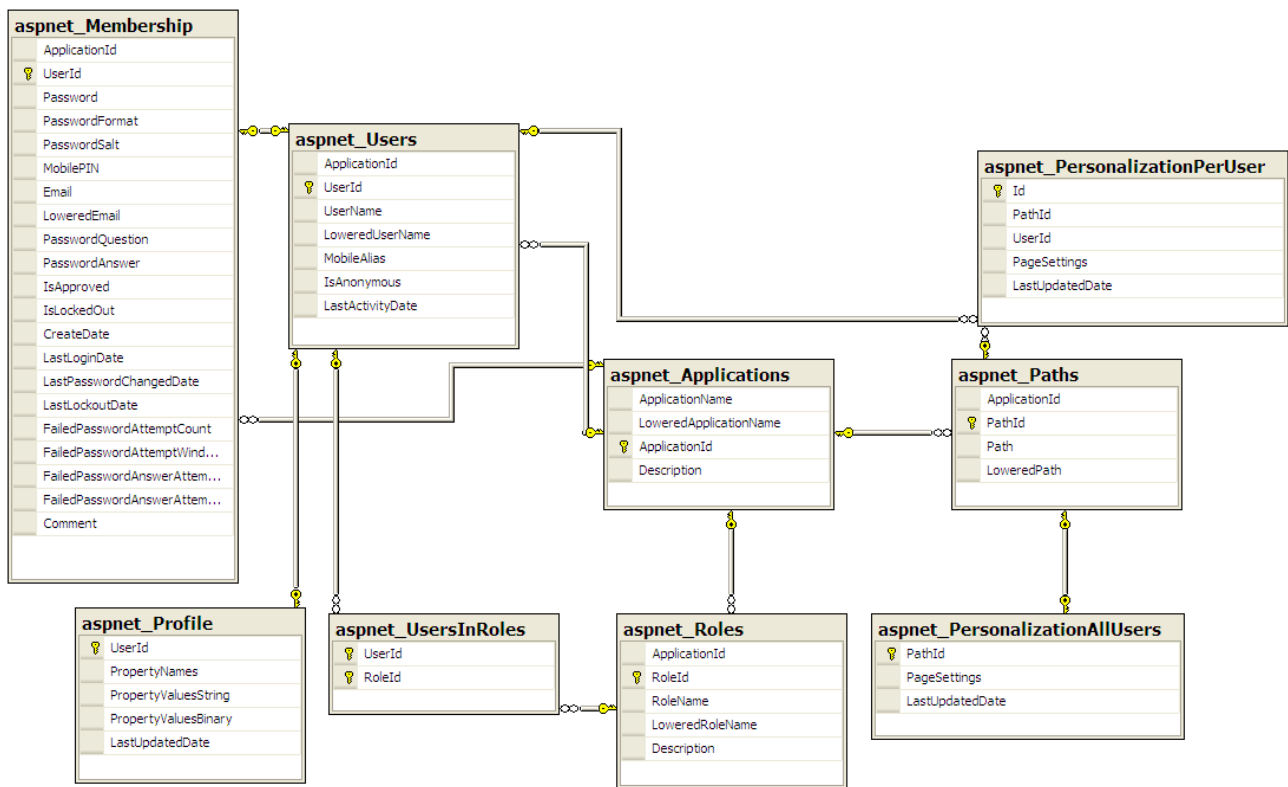


Abbildung 1: Membership-Datenbank

# Die Portal-Datenbank

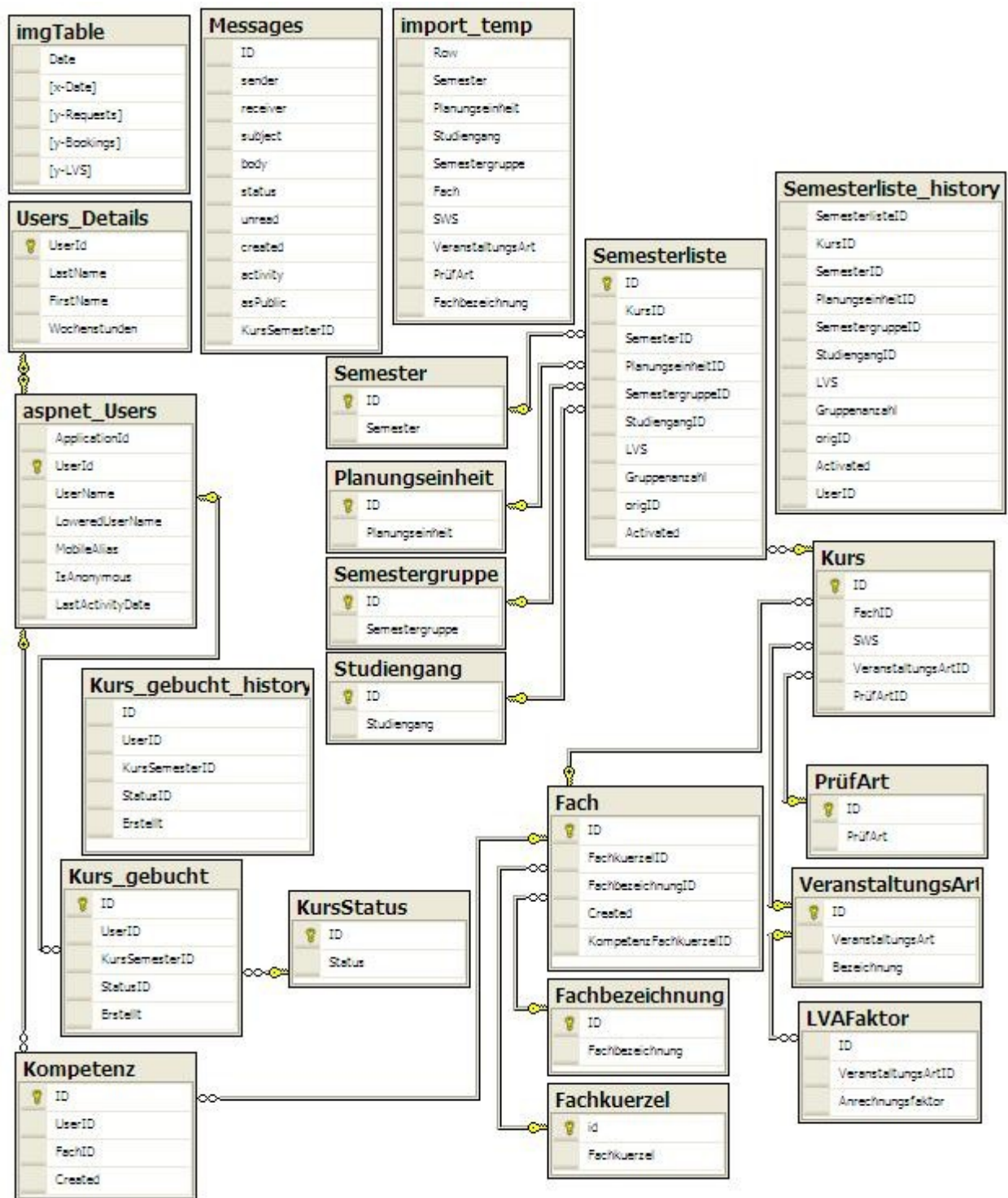


Abbildung 2: Portal-Datenbank

# Stored Procedure

/\*\*\*\*\*\* Objekt: StoredProcedure [dbo].[mem\_semester\_bindSemesterListe] Skriptdatum: 02/23/2012 13:14:11 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[mem\_semester\_bindSemesterListe]

@PageIndex INT,

@UserName varchar(50),

@Semester VARCHAR(50),

@PageSize INT,

@SortField VARCHAR(100),

@ItemCount INT out

AS

BEGIN

IF LEN(@SortField) = 0

SET @SortField = 'Semester'

DECLARE @Seitenende int,

@Seitenanfang int

SET @Seitenanfang = (@PageIndex \* @PageSize) + 1

SET @Seitenende = (@Seitenanfang + @PageSize) - 1

SET @ItemCount = (select count(distinct ID) from Taskplaner.dbo.vw\_SemesterlisteStatus where Semester=@Semester and ID in(

select kis.ID from Taskplaner.dbo.Kompetenz ko

left join Taskplaner.dbo.aspnet\_Users u on ko.UserID = u.UserId

left join Taskplaner.dbo.Fach f on ko.FachID=f.ID

left join Taskplaner.dbo.Kurs k on f.ID = k.FachID

left outer join Taskplaner.dbo.Semesterliste kis on k.ID = kis.KursID

where u.UserName=@UserName and kis.ID = kis.origID and kis.activated<>0

)and ID not in (select kg.KursSemesterID from Taskplaner.dbo.Kurs\_gebucht kg left join

Taskplaner.dbo.aspnet\_Users u on kg.UserID = u.UserID where u.UserName=@UserName)

)

BEGIN

EXEC('

with temp as (

select ID,Semester,Planungseinheit,Semestergruppe,Fachkuerzel,SWS,LVS,VeranstaltungsArt,Fachbezeichnung,Status

+ "(" + cast(count(Status) as varchar) + ")" as Status, Studiengang

from TaskPlaner.dbo.vw\_SemesterlisteStatus where Semester="@Semester" and

ID in (select kis.ID from Taskplaner.dbo.Kompetenz ko

left join Taskplaner.dbo.aspnet\_Users u on ko.UserID = u.UserId

left join Taskplaner.dbo.Fach f on ko.FachID=f.ID

left join Taskplaner.dbo.Kurs k on f.ID = k.FachID

left outer join Taskplaner.dbo.Semesterliste kis on k.ID = kis.KursID

where u.UserName="@UserName" and kis.ID = kis.origID and kis.activated<>0)

and ID not in (select kg.KursSemesterID from Taskplaner.dbo.Kurs\_gebucht kg left join

Taskplaner.dbo.aspnet\_Users u on kg.UserID = u.UserID where u.UserName="@UserName")

group by ID,Semester,Planungseinheit,Semestergruppe,Fachkuerzel,SWS,LVS,VeranstaltungsArt,Fachbezeichnung,Status,Studiengang

),

rowedView as (

SELECT

ROW\_NUMBER() OVER (ORDER BY '+@SortField+' AS Row, \*

FROM

temp

)

select

rowedView.ID,rowedView.Semester,rowedView.Planungseinheit,rowedView.Semestergruppe,rowedView.Fachkuerzel,rowedView.SWS,rowedView

.LVS,rowedView.VeranstaltungsArt,rowedView.Fachbezeichnung,rowedView.Status,rowedView.Studiengang from

rowedView where rowedView.Row between('+@Seitenanfang+') and ('+@Seitenende+')

order by rowedView.Row

)

END

END

GO

# Trigger

```
USE [TaskPlaner]
GO
/***** Objekt: Trigger [dbo].[addExpertise] Skriptdatum: 02/23/2012 17:40:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE trigger [dbo].[addExpertise]
on [dbo].[Fach]
for update
as
begin
    if(update(KompetenzFachkuerzelID))
    BEGIN
        DECLARE @oldID int
        DECLARE @newID int
        SET @oldID = (select KompetenzFachkuerzelID from deleted)
        SET @newID = (select KompetenzFachkuerzelID from inserted)
        if(@oldID IS NULL)
        BEGIN
            DECLARE @FachID int
            SET @FachID = (select ID from inserted)
            insert into Taskplaner dbo.Kompetenz
            select distinct UserID,@FachID,getDate() from Kompetenz where FachID
            in (select ID from Fach where KompetenzFachkuerzelID=@newID)
            and FachID <> @FachID
        END
    END
END
```

## **B. Konfigurationsdateien**



## Web.sitemap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap>
  <siteMapNode title="Home" description="Home" url="~/Taskplaner.aspx" roles="professor,admin,staff">
    <siteMapNode title="personal" description="themselves concerning" roles="professor,admin,staff">
      <siteMapNode title="mails" description="check emails" url="~/memberpages/Messaging.aspx"/>
      <siteMapNode title="profile data" description="change personal data" url="~/memberpages/Members.aspx">
        <siteMapNode title="change password" description="change password" url="~/memberpages/ChangePassword.aspx" />
        <siteMapNode title="change profile" description="change profile data" url="~/memberpages/ChangeDetails.aspx" />
      </siteMapNode>
      <siteMapNode title="expertise" description="alter areas of expertise" url="~/memberpages/Kompetenz.aspx"/>
    </siteMapNode>
    <siteMapNode title="semester events" description="list of semester events" roles="professor,admin,staff">
      <siteMapNode title="actual events" description="events for the actual and following semesters" url="~/memberpages/Semester.aspx"
        role="professor,staff"/>
      <siteMapNode title="event history" description="past semesters" url="~/memberpages/Archiv.aspx"/>
    </siteMapNode>
    <siteMapNode title="faq" description="frequently asked question" url="~/memberpages/Help.aspx" roles="professor,admin,staff"/>
    <siteMapNode title="administration" description="account data" roles="admin" url="~/adminpages/Admin_Overview.aspx">
      <siteMapNode title="import semester list" description="import new semester list" url="~/adminpages/ImportSemester.aspx" />
      <siteMapNode title="create course" description="create a new course" url="~/adminpages/CreateCourse.aspx" />
      <siteMapNode title="bind expertise" description="bind expertise to course" url="~/adminpages/Expertise.aspx" />
    </siteMapNode>
    <siteMapNode title="view semester list" description="view booked status" url="~/adminpages/Semester.aspx"/>
    <siteMapNode title="members" description="member data" url="~/adminpages/Members.aspx">
      <siteMapNode title="register members" description="register member" url="~/adminpages/Register.aspx" />
      <siteMapNode title="change memberdetails" description="change memberdetails" url="~/adminpages/ChangeMembers.aspx" />
      <siteMapNode title="lost password" description="generate new user password" url="~/adminpages>PasswordRecovery.aspx" />
    </siteMapNode>
    <siteMapNode title="preferences" description="preferences" url="~/adminpages/Timeline.aspx"/>
    <siteMapNode title="help" description="description of sections" url="~/adminpages/help.aspx"/>
  </siteMapNode>
</siteMapNode>
</siteMap>
```

## appsettings.config

```
<appSettings>
  <add key="Deadline" value="634710816000000000" />
  <add key="Deadline2" value="634686624000000000" />
  <add key="Status" value="2" />
  <add key="lastSemester" value="SS 2010" />
  <add key="Semester" value="SS 2010" />
  <add key="minimumPercentage" value="15" />
  <add key="maximumPercentage" value="100" />
  <add key="WarningMail" value="False" />
  <add key="DeadlineMail" value="False" />
  <add key="FinishedMail" value="False" />
  <add key="FinishedAdminMail" value="False" />
  <add key="WorkloadMail" value="False" />
  <add key="courseSelect" value="False" />
  <add key="courseUnselect" value="False" />
  <add key="expertiseSelect" value="False" />
  <add key="expertiseUnselect" value="False" />
  <add key="WorkloadMailSendDay" value="7" />
</appSettings>
```

## web.config (Webanwendung)

```
<?xml version="1.0"?>
<configuration>
  <configSections>
    <sectionGroup name="system.web.extensions" type="System.Web.Configuration.SystemWebExtensionsSectionGroup, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35">
      <sectionGroup name="scripting" type="System.Web.Configuration.ScriptingSectionGroup, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35">
        <section name="scriptResourceHandler"
          type="System.Web.Configuration.ScriptingScriptResourceHandlerSection, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" requirePermission="false" allowDefinition="MachineToApplication"/>
        <sectionGroup name="webServices" type="System.Web.Configuration.ScriptingWebServicesSectionGroup, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35">
          <section name="jsonSerialization" type="System.Web.Configuration.ScriptingJsonSerializationSection, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" requirePermission="false" allowDefinition="Everywhere" />
          <section name="profileService" type="System.Web.Configuration.ScriptingProfileServiceSection, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" requirePermission="false" allowDefinition="MachineToApplication" />
          <section name="authenticationService" type="System.Web.Configuration.ScriptingAuthenticationServiceSection, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" requirePermission="false" allowDefinition="MachineToApplication" />
        </sectionGroup>
      </sectionGroup>
    </sectionGroup>
  </configSections>
  <appSettings configSource="appsettings.config"/>
  <connectionStrings>
    <remove name="LocalSQLServer"/>
    <add name="LocalSQLServer" connectionString="Data Source=PC180444111268\SQLEXPRESS;Initial Catalog=TaskPlanner;Integrated Security=True" providerName="System.Data.SqlClient"/>
    <add name="TaskPlannerConnectionString" connectionString="Data Source=PC180444111268\SQLEXPRESS;Initial Catalog=TaskPlanner;Integrated Security=SSPI" providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <system.web>
    <httpHandlers>
      <remove verb="*" path="*.asmx"/>
      <add verb="*" path="*.asmx" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
      <add verb="*" path="*_AppService.axd" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
      <add verb="GET,HEAD" path="ScriptResource.axd" type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" validate="false"/>
    </httpHandlers>
    <httpModules>
      <add name="ScriptModule" type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
    </httpModules>
    <sessionState mode="InProc" cookieless="UseCookies" timeout="20" />
    <authentication mode="Forms">
      <forms name="taskplanner" defaultUrl="Taskplanner.aspx" protection="All" slidingExpiration="true" timeout="20" loginUrl="login.aspx"/>
    </authentication>
    <authorization>
      <allow roles="admin" />
      <allow roles="staff" />
      <allow roles="professor" />
      <deny users="?" />
      <deny users="*" />
    </authorization>
    <globalization uiCulture="en-US" culture="en-US"/>
    <roleManager enabled="true"/>
    <compilation debug="true">
      <assemblies>
        <add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
        <add assembly="System.Web.Abstractions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
        <add assembly="System.Core, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Data.Linq, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Windows.Forms, Version=2.0.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Data.Services.Client, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Data.Services.Design, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Data.Entity, Version=3.5.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Design, Version=2.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A"/>
      </assemblies>
    </compilation>
    <siteMap defaultProvider="XmlSiteMapProvider" enabled="true">
      <providers>
        <add name="XmlSiteMapProvider" description="Default SiteMap provider." type="System.Web.XmlSiteMapProvider" siteMapFile="Web.sitemap" securityTrimmingEnabled="true"/>
      </providers>
    </siteMap>
  </system.web>
</configuration>
```

```

</siteMap>
<customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.htm">
  <error statusCode="403" redirect="NoAccess.htm"/>
  <error statusCode="404" redirect="FileNotFound.htm"/>
</customErrors>
<membership defaultProvider="SqlProvider" userIsOnlineTimeWindow="20">
  <providers>
    <clear/>
    <add name="SqlProvider" type="System.Web.Security.SqlMembershipProvider" connectionStringName="LocalSqlServer"
enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="false" passwordFormat="Hashed"
requiresUniqueEmail="true" applicationName="/" />
  </providers>
</membership>
</system.web>
<system.net>
  <mailSettings>
    <smtp deliveryMethod="Network" from="dozentenplaner@yahoo.de">
      <network host="smtp.mail.yahoo.de" port="25" password="xxx" userName="xxx"/>
    </smtp>
  </mailSettings>
</system.net>
</configuration>

```

## web.config (Unterverzeichnis: adminpages)

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <authorization>
      <deny roles="professor" />
      <deny roles="staff" />
    </authorization>
  </system.web>
</configuration>

```

## web.config (Unterverzeichnis: memberpages)

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
</configuration>

```

## **C. Beispielcode**

## Datagrid "profListe" für die Darstellung der verfügbaren Veranstaltungen

```
<asp:DataGrid ID="profListe" runat="server" AutoGenerateColumns="False" AllowPaging="True"
    AllowCustomPaging="true" AllowSorting="True" OnSortCommand="profListe_SortCommand"
    OnDeleteCommand="profListe_DeleteCommand" OnPageIndexChanged="profListe_PageIndexChanged"
    PageSize="20" OnItemCommand="profListe_Command" OnItemDataBound="profListe_MarkLine"
    OnItemCreated="profListe_ItemCreated">
    <AlternatingItemStyle BackColor="#E0E0E0"></AlternatingItemStyle>
    <HeaderStyle BackColor="Gray" CssClass="headerText" Font-Bold="true" HorizontalAlign="center">
    </HeaderStyle>
    <Columns>
        <asp:BoundColumn DataField="ID" Visible="False"></asp:BoundColumn>
        <asp:BoundColumn DataField="Semester" ReadOnly="True" SortExpression="Semester" HeaderText="Semester">
        </asp:BoundColumn>
        <asp:BoundColumn DataField="Planungseinheit" ReadOnly="True" SortExpression="Planungseinheit"
            HeaderText="unit">
            <HeaderStyle Width="40px" />
        </asp:BoundColumn>
        <asp:BoundColumn DataField="Semestergruppe" ReadOnly="True" SortExpression="Semestergruppe"
            HeaderText="group"></asp:BoundColumn>
        <asp:BoundColumn DataField="Fachkuerzel" ReadOnly="True" SortExpression="Fachkuerzel"
            HeaderText="course"></asp:BoundColumn>
        <asp:BoundColumn DataField="SWS" ReadOnly="True" SortExpression="SWS" HeaderText="SWS">
        </asp:BoundColumn>
        <asp:BoundColumn DataField="LVS" ReadOnly="True" SortExpression="LVS" HeaderText="LVS">
        </asp:BoundColumn>
        <asp:BoundColumn DataField="Studiengang" ReadOnly="True" SortExpression="Studiengang"
            HeaderText="course of studies"></asp:BoundColumn>
        <asp:BoundColumn DataField="VeranstaltungsArt" ReadOnly="True" SortExpression="VeranstaltungsArt"
            HeaderText="type"></asp:BoundColumn>
        <asp:BoundColumn DataField="Fachbezeichnung" ReadOnly="True" SortExpression="Fachbezeichnung"
            HeaderText="course name"></asp:BoundColumn>
        <asp:TemplateColumn HeaderText="status" SortExpression="status">
            <ItemTemplate>
                <asp:Label ID="Label1" runat="server" Text='<%=# DataBinder.Eval(Container, "DataItem.Status") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateColumn>
        <asp:TemplateColumn>
            <ItemTemplate>
                <asp:ImageButton ID='bookButton' ImageUrl='~/App_Themes/teach.gif' runat='server'
                    CausesValidation='false' CommandName='Delete' ToolTip='book course' />
            </ItemTemplate>
        </asp:TemplateColumn>
        <asp:TemplateColumn>
            <ItemTemplate>
                <asp:ImageButton ID="viewRequests" ImageUrl="~/App_Themes/view.gif" runat="server"
                    CausesValidation="false" CommandName="viewRequesters" ToolTip="view Requesters" />
            </ItemTemplate>
        </asp:TemplateColumn>
    </Columns>
    <PagerStyle BackColor="Gray" Mode="NumericPages"></PagerStyle>
</asp:DataGrid>
```

## Methode zum Füllen des Datagrids „profListe“

```
///<summary>
/// Aufbau Datagrid für verfügbare Kurse
///</summary>
///<param name="currentPage">ausgewählte Seite</param>
public void bindData(int currentPage)
{
    if (noCourse != 0)
    {
        try
        {
            int numRows = 0;
            profListe.PageSize = Convert.ToInt32(this.DropDownEntrys.SelectedValue);
            con = new SqlConnection(ConfigurationManager.ConnectionStrings["TaskPlanerConnectionString"].ConnectionString);
            SqlCommand cmd = new SqlCommand("mem_semester_bindSemesterListe", con);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.Add("@PageIndex", SqlDbType.Int).Value = profListe.CurrentPageIndex;
            cmd.Parameters.Add("@UserName", SqlDbType.NVarChar).Value = User.Identity.Name.ToString();
            cmd.Parameters.Add("@Semester", SqlDbType.NVarChar).Value =
                Context.Server.HtmlDecode((string)DropDownList1.SelectedItem.Value.ToString());
            cmd.Parameters.Add("@PageSize", SqlDbType.Int).Value = profListe.PageSize;
            cmd.Parameters.Add("@SortField", SqlDbType.NVarChar).Value = (string)ViewState["SortField"].ToString() +
                (string)ViewState["SortOrder"].ToString();//SortField;
            cmd.Parameters.Add("@ItemCount", SqlDbType.Int).Direction = ParameterDirection.Output;
            cmd.Connection = con;
            da = new SqlDataAdapter(cmd);
            da.Fill(ds);
            con.Open();
            cmd.ExecuteNonQuery();
            numRows = (int)cmd.Parameters["@ItemCount"].Value;
            profListe.VirtualItemCount = numRows;
            profListe.Columns[1].Visible = this.cbSemester.Checked; //Semester
            profListe.Columns[2].Visible = this.cbUnit.Checked; //Unit
            profListe.Columns[3].Visible = this.cbGroup.Checked; //group
            profListe.Columns[4].Visible = this.cbCourse.Checked; //course
            profListe.Columns[5].Visible = this.cbSWS.Checked; //sws
            profListe.Columns[6].Visible = this.cbLVS.Checked; //lvs
            profListe.Columns[7].Visible = this.cbCourseOfStudy.Checked; //course of study
            profListe.Columns[8].Visible = this.cbType.Checked; //type
            profListe.Columns[9].Visible = this.cbCourseName.Checked; //course name
            DataView dv = new DataView(ds.Tables[0]);
            profListe.DataSource = dv;
            profListe.DataBind();
            if (numRows > 0)
            {
                lblRows.Text = numRows.ToString() + " Events";
                profListe.Visible = true;
                profListe_header.Visible = true;
                lblRows.Visible = true;
            }
            else
            {
                showColumns.Visible = false;
                profListe.Visible = false;
                profListe_header.Visible = false;
                lblRows.Text = "<br>There are no other events available that match your area of expertise!";
            }
        }
        catch (Exception e)
        {
            e.Message.ToString();
        }
        finally
        {
            if (con.State == ConnectionState.Open)
            {
                con.Close();
            }
        }
    }
    else
    {
        this.profListe_header.Text = "Temporary no events available!";
    }
    UpdatePanel1.Update();
}
```

# D. Handbuch

## D.1. Das Portal

In diesem Abschnitt werden das Design der Oberfläche (Abbildung [D.1](#)) und die Funktionalitäten der Inhaltsseiten des Portals im Einzelnen vorgestellt.

### D.1.1. Design der Oberfläche

Das Portal besteht aus drei Bereichen:

- Kopfzeile
- vertikale Navigationsleiste
- Inhaltsbereich

Die Kopfzeile enthält den Login-Status, die Benachrichtigungen über neue Nachrichten und die Überschrift des aktuellen Inhaltes.

Die vertikale Navigationsleiste beinhaltet die Treeview des Portals, über das der Benutzer navigieren kann.<sup>1</sup>

Die Inhaltsseite wird über eine Masterseite generiert. Beim Laden einer Inhaltsseite müssen die Kopfzeile und die vertikale Navigationsleiste nicht immer neu geladen werden. Um dies zu erreichen, wird auf den entsprechenden Inhaltsseiten ein partielles Update veranlasst.

Für die Präsentation der Inhaltsseiten werden überwiegend `Datagrids`<sup>2</sup> (siehe Glossar) verwendet, wobei die Struktur und Farbgebung bei allen vorkommenden Seiten weitgehend einheitlich gehalten werden.

---

<sup>1</sup>vgl. ([Schwichtenberg, 2006](#), 405 ff)

<sup>2</sup>vgl. ([Schwichtenberg, 2006](#), 445 ff)

The screenshot shows the 'Semesterlist' portal interface. At the top, it says 'Semesterlist' and 'You are logged in. Welcome, administrator'. Below this is a navigation menu on the left with categories like 'personal', 'semester events', and 'administration'. The main content area displays a table for the semester 'SS 2010' with columns for 'unit', 'group', 'course', 'SWS', 'LVS', 'GC', 'type', 'able', 'requests', and 'staff'. The table lists various courses such as 'D-inf B-AI2 DBP', 'U-inf TI6 WPP1', and 'D-inf B-AI2 RMP'. At the bottom of the table, it indicates '19 events' and provides a search filter for 'Semester'.

unit	group	course	SWS	LVS	GC	type	able	requests	staff
D-inf	B-AI2	DBP	1	3.00	3	Prak	5	stored	eee
D-inf	TI6	WPP1	1	1.00	1	Pro	4	stored	ddc
U-inf	U-III	GIP	1	3.00	3	Prak	5	stored	ddc
D-inf	B-AI2	RMP	3	3.00	1	SeU	5	stored	cuc
D-inf	TI6	WP1	3	3.00	1	SeU	4	stored	cuc
D-inf	B-AI4	WPP1	1	3.00	3	Prak	4	stored	bob
D-inf	B-AI1	G	3	3.00	1	ScU	5	stored	bob
U-inf	U-AI2	RMFP	1	3.00	3	Prak	5	stored	bob
D-inf	B-AI3	AD	3	3.00	1	SeU	7	stored	administrator
D-inf	B-AI4	WP1	3	3.00	1	SeU	4	stored	administrator
D-inf	B-AI3	ADP	1	3.00	3	Prak	7	stored	administrator
D-inf	B-AI2	PRP2	1	3.00	3	Prak	5	0	
U-inf	U-AI2	PR2	3	3.00	1	SeU	5	0	
D-inf	B-AI2	DB	3	3.00	1	SeU	5	0	
D-inf	TI6	WP2	3	3.00	1	SeU	4	0	
D-inf	B-AI1	G-U	1	3.00	3	Prak	5	3	
D-inf	B-TI1	GT	3	3.00	1	ScU	3	0	
U-inf	TI6	WPP2	1	1.00	1	Pro	4	2	
D-inf	A,5	PO	3	3.00	1	Pro	5	2	

Abbildung D.1.: Übersicht Portal

## D.1.2. Benutzerbereich

Jeder Benutzer des Portals kann diesen Bereich einsehen wie auch der Admin da dieser als Professor mit administrativen Rechten angesehen wird.

### D.1.2.1. Login

Zu Beginn erscheint wie bei jedem Portal der Login-Bereich (Abbildung D.2). Hier können sich Benutzer mit bereits bestehendem Account anmelden, oder neue Benutzer den Admin per Email kontaktieren, da dieser ausschließlich neue Benutzer registrieren kann. Über den Login-Button gelangt man schließlich auf die Startseite des Portals.



admin for authorization requests!'." data-bbox="146 128 871 254"/>

Login for the taskplanner

Username:

Passwort:

Contact [admin](#) for authorization requests!

Abbildung D.2.: Login-Bereich

### D.1.2.2. Übersicht

Die Startseite des Portals (Abbildung D.3) gibt eine kleine Übersicht über die laufenden Aktivitäten.

Hier wird der Benutzer willkommen geheißen und kurz erläutert, wozu der Taskplaner dienen soll. Zudem werden Einstiegslinks genannt, die direkt zur Kompetenzerfassung, Veranstaltungsanfrage sowie zur Informationsseite führen. Der Link für den administrativen Bereich (*Administrate semesters*) wird ausschließlich auf der Startseite des Admins angezeigt, alle weiteren Benutzer können diesen nicht einsehen. Am unteren Rand der Seite, im rot umrahmten Textbereich, wird die verbleibende Zeit bis zum Ablauf der Deadline angezeigt, um den Benutzer hierüber auf dem Laufenden zu halten.

Home

### Welcome to the taskplaner!

The taskplaner manages the allocation of your course selections for the upcoming semester. By defining your areas of expertise, the corresponding selectable courses are narrowed down from the comprehensive range of courses. Out of the result set you can choose your preferred courses.

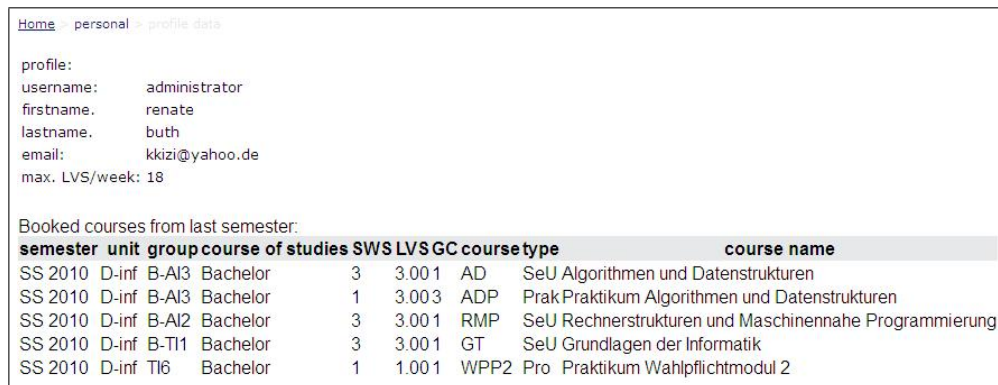
- Please supply your [areas of expertise](#)
- [Request](#) courses for the upcoming semester
- Read [FAQ](#) for detailed informations
- [Administrate](#) semesters

The registration deadline SS 2012 will end on: Thursday, January 26, 2012  
Still 9 days,0 hours,11 minutes,58 seconds until expiration of the deadline!

Abbildung D.3.: Übersicht der Startseite des Portals

### D.1.2.3. Profil-Daten

Die Profildaten (Abbildung D.4) umfassen persönliche Angaben des Benutzers wie den Vor- und Nachnamen sowie den Benutzernamen für das Portal, die Email-Adresse und die maximalen LVS pro Woche. Zudem wird eine Liste der Veranstaltungen angezeigt, die im letzten Semester erfolgreich gebucht wurden.



Home > personal > profile data

profile:  
 username: administrator  
 firstname: reate  
 lastname: both  
 email: kkizi@yahoo.de  
 max. LVS/week: 18

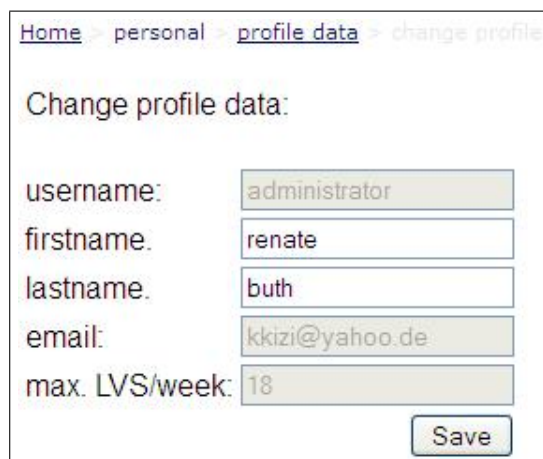
Booked courses from last semester:

semester	unit	group	course of studies	SWS	LVS	GC	course type	course name
SS 2010	D-inf	B-AI3	Bachelor	3	3.001	AD		SeU Algorithmen und Datenstrukturen
SS 2010	D-inf	B-AI3	Bachelor	1	3.003	ADP		PrakPraktikum Algorithmen und Datenstrukturen
SS 2010	D-inf	B-AI2	Bachelor	3	3.001	RMP		SeU Rechnerstrukturen und Maschinennahe Programmierung
SS 2010	D-inf	B-TI1	Bachelor	3	3.001	GT		SeU Grundlagen der Informatik
SS 2010	D-inf	TI6	Bachelor	1	1.001	WPP2		Pro Praktikum Wahlpflichtmodul 2

Abbildung D.4.: Profildaten

### D.1.2.4. Profil-Daten ändern

In diesem Bereich kann der Benutzer die Änderungen seiner persönlichen Daten (Vor- und Nachname) vornehmen (Abbildung D.5).



Home > personal > profile data > change profile

Change profile data:

username:

firstname:

lastname:

email:

max. LVS/week:

Abbildung D.5.: Profildaten ändern

### D.1.2.5. Passwort ändern

In diesem Bereich wird dem Benutzer die Möglichkeit gegeben, sein Passwort zu ändern (Abbildung D.6). Die Stärke des Passwortes wird dabei durch Validierungssteuerelemente sichergestellt. Durch Eingabe des vorherigen Passwortes kann nun ein neues durch wiederholtes Niederschreiben generiert werden.

Change Password:

old password:

new password:

confirm new password:

Abbildung D.6.: Passwort ändern

### D.1.2.6. Auswahl der Kompetenzen

Der Benutzer wird gebeten, seine fachlichen Fähigkeiten anzugeben, um ihm eine Auswahl der von ihm wählbaren Veranstaltungen anzeigen zu können. Dadurch wird der Benutzer nicht mit Veranstaltungen anderer Fachbereiche konfrontiert und erhält eine strukturierte Übersicht an Veranstaltungen, die auf seine Kompetenzen zugeschnitten sind (Abbildung D.7).

Select courses that meet your area of expertise:

course	course name	Created	
DB	Datenbanken	05.01.2012	
GI	Grundlagen der Informatik	05.01.2012	
LB	Logik und Berechenbarkeit	06.01.2012	
PO	Projektmodul	05.01.2012	
PR2	Programmieren 2	05.01.2012	
RMP	Rechnerstrukturen und Maschinennahe Programmierung	05.01.2012	
TH1	Theoretische Informatik 1	06.01.2012	
VS	Verteilte Systeme	06.01.2012	
WP1	Wahlpflichtfach 1	05.01.2012	
WP1	Wahlpflichtmodul 1	05.01.2012	
WP2	Wahlpflichtmodul 2	05.01.2012	

1

10 areas of expertise

Your area of expertise:

course	course name	
AD	Algorithmen und Datenstrukturen	

1 areas of expertise

Abbildung D.7.: Kompetenzen wählen

### D.1.2.7. Übersicht Veranstaltungswahl

Hat der Benutzer seine Kompetenzen angegeben, wird ihm eine Liste der von ihm wählbaren Veranstaltungen aus der aktuellen Semesterliste angezeigt (Abbildung D.8). Hierbei werden die Komponenten der Veranstaltungen, wie beispielsweise die Planungseinheit (*unit*) oder auch Fachbezeichnung (*course name*), im Einzelnen aufgelistet.

Home > semester events > actual events

SS 2010

Choose for you from the eligible courses:

Semester	unit	group	course	SWS	LVS	GC	type	course name	status
SS 2010	D-inf	B-AI3	ADP	1	3.00		Prak	Praktikum Algorithmen und Datenstrukturen	
SS 2010	D-inf	B-AI3	AD	3	3.00		SeU	Algorithmen und Datenstrukturen	

1

2 Events

Abbildung D.8.: Veranstaltungen wählen

### D.1.2.8. Übersicht Veranstaltungs-Historie

Bereits erfolgreich zugewiesene Veranstaltungen aus vergangenen Semestern können in dieser Historie (Abbildung D.9) nachgeschaut werden.

SS 2010

Semester	unit	group	study	course	SWS	LVS	GC	type	course name	username
SS 2010	D-inf	TI6	Bachelor	WPP2	1	1.00	1	Pro	Praktikum Wahlpflichtmodul 2	bbb
SS 2010	D-inf	TI6	Bachelor	WPP1	1	1.00	1	Pro	Praktikum Wahlpflichtmodul 1	ddd
SS 2010	D-inf	B-AI4	Bachelor	WPP1	1	3.00	3	Prak	Praktikum Wahlpflichtfach 1	bbb
SS 2010	D-inf	TI6	Bachelor	WP2	3	3.00	1	SeU	Wahlpflichtmodul 2	ccc
SS 2010	D-inf	B-AI4	Bachelor	WP1	3	3.00	1	SeU	Wahlpflichtfach 1	administrator
SS 2010	D-inf	TI6	Bachelor	WP1	3	3.00	1	SeU	Wahlpflichtmodul 1	ccc
SS 2010	D-inf	B-AI2	Bachelor	RMPP	1	3.00	3	Prak	Praktikum Rechnerstrukturen und Maschinenn. Programm.	bbb
SS 2010	D-inf	B-AI2	Bachelor	RMP	3	3.00	1	SeU	Rechnerstrukturen und Maschinennahe Programmierung	ccc
SS 2010	D-inf	B-AI2	Bachelor	PRP2	1	3.00	3	Prak	Praktikum Programmieren 2	eee
SS 2010	D-inf	B-AI2	Bachelor	PR2	3	3.00	1	SeU	Programmieren 2	ddd
SS 2010	D-inf	AI5	Bachelor	PO	8	8.00	1	Pro	Projektmodul	ccc
SS 2010	D-inf	B-TI1	Bachelor	GTP	1	3.00	3	Prak	Praktikum Grundlagen der Informatik	ddd
SS 2010	D-inf	B-TI1	Bachelor	GT	3	3.00	1	SeU	Grundlagen der Informatik	eee
SS 2010	D-inf	B-AI1	Bachelor	GIÜ	1	3.00	3	Prak	Übung Grundlagen der Informatik	eee
SS 2010	D-inf	B-AI1	Bachelor	GI	3	3.00	1	SeU	Grundlagen der Informatik	bbb
SS 2010	D-inf	B-AI2	Bachelor	DBP	1	3.00	3	Prak	Praktikum Datenbanken	eee
SS 2010	D-inf	B-AI2	Bachelor	DB	3	3.00	1	SeU	Datenbanken	ccc
SS 2010	D-inf	B-AI3	Bachelor	ADP	1	3.00	3	Prak	Praktikum Algorithmen und Datenstrukturen	administrator
SS 2010	D-inf	B-AI3	Bachelor	AD	3	3.00	1	SeU	Algorithmen und Datenstrukturen	administrator

Abbildung D.9.: Veranstaltungs-Historie

### D.1.3. Admin Bereich

Admins können zusätzlich zum Professoren-Bereich auch den administrativen Bereich einsehen, da der Admin in diesem System als Professor mit erweiterten Aufgaben gilt. So ist es auch dem Admin möglich, Veranstaltungen aus der aktuellen Semesterliste zu wählen.

#### D.1.3.1. Startseite

Die Administration-Startseite (Abbildung D.10) stellt drei Bereiche für eine bessere Übersicht dar. Hierzu zählen der zeitliche Verlauf des aktuellen Semesters, Einstiegslinks und eine Zusammenfassung über Aktivitäten die noch geklärt werden sollten.

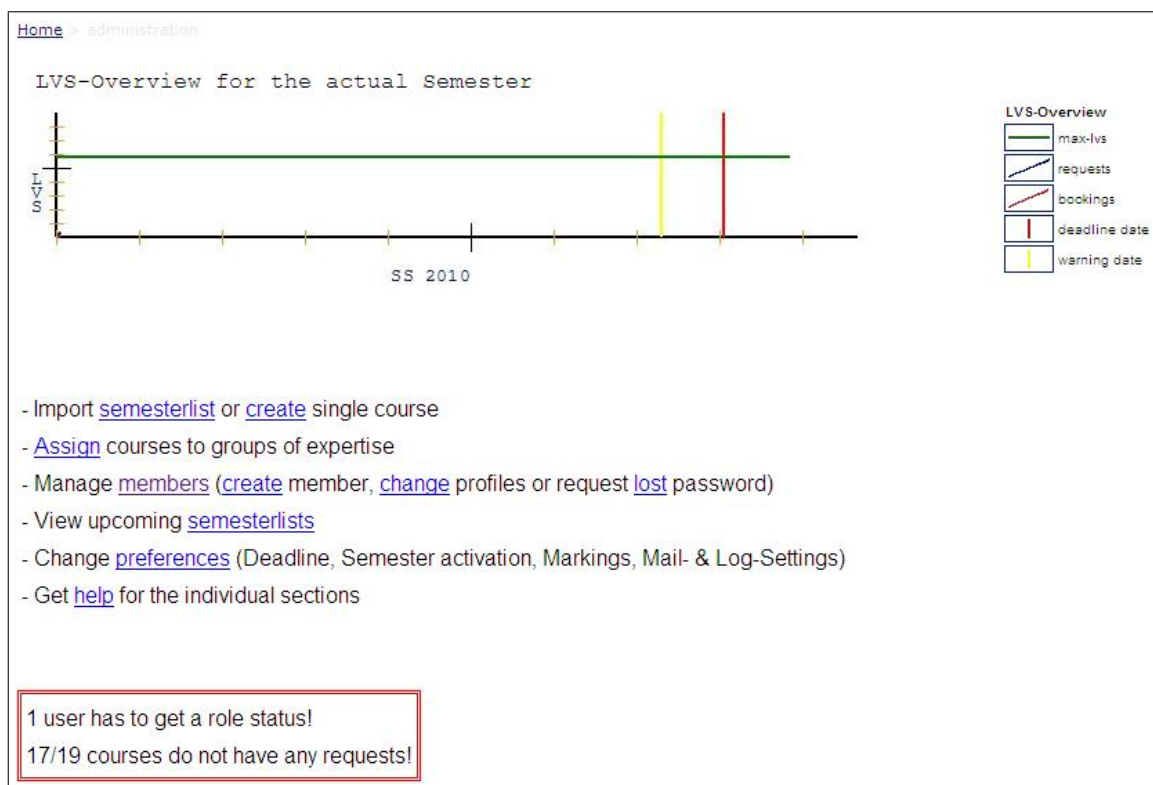


Abbildung D.10.: Startseite des Admins

Die Graphik stellt den zeitlichen Verlauf des aktuellen Semesters dar. Hierbei ist die x-Achse die Zeitachse, die das Semester in 180 Tage unterteilt und die y-Achse die Gesamtheit der

LVS des aktuellen Semesters.

Der grüne horizontale Graph stellt die Summe der LVS aller Veranstaltungen des aktuellen Semesters dar, die rote vertikale Linie den Zeitpunkt der Deadline und die gelbe vertikale Linie den entsprechenden Warnzeitpunkt.

Der blaue Graph zeigt die Summe der angefragten LVS, der rote hingegen stellt die tatsächlich gebuchten Veranstaltungen dar und wird im Normalfall erst nach Ablauf der Deadline, sobald der Admin die Veranstaltungen zuteilt, ein Wert annehmen.

In Ausnahmefällen kann der Admin eine Veranstaltung auch schon vor Ablauf der Deadline zuteilen. Allerdings wird hierdurch anderen Mitbewerbern die Möglichkeit genommen, die Veranstaltung ebenfalls anfragen zu können. Der Graph wird über Daten aus einer SQL-Tabelle gezeichnet. Statt diese Berechnung anhand eines cron-jobs oder DTS-Packets täglich ausführen zu lassen, wird der Graph beim Aufrufen der Seite aktualisiert. Dies ermöglicht die Erfassung der Anfragen des aktuellen Tages miteinzuberechnen.

Der zweite Teil der Übersicht stellt Einstiegslinks für den Admin zur Verfügung, welche die Navigation durch das Portal erleichtern sollen.

Im unteren rotumrandeten Bereich werden dem Admin Meldungen für das aktuelle Semester angezeigt. Mögliche Meldungen wären z.B. fehlende Rollenzuweisungen, fehlende maximal LVS Angabe oder auch die Anzahl der Veranstaltungen, für die es keine Kompetenzen gibt und die Anzahl der Veranstaltungen, für die noch keine Anfragen existieren.

### **D.1.3.2. Importieren von Veranstaltungslisten**

Beim Importieren von Veranstaltungslisten (Abbildung [D.11](#)) ist die Spezifikation zu beachten, die auf der 'help'-Seite der Plattform näher erläutert wird.

Sofern es sich bei der Datei um eine .csv- bzw .xls-Datei handelt, wird diese im ersten Schritt auf dem Server hochgeladen.

Im zweiten Schritt wird die Datei auf der Seite dargestellt. Wenn sich jedoch Fehler in der Spezifikation befinden, werden diese aufgelistet. Sind die Fehler nun behoben oder im Idealfall gar nicht erst aufgetreten, erscheint der Button zum Speichern in der Datenbank. Sobald die Datei erfolgreich gespeichert wurde müssen die Daten noch aktiviert werden, damit diese in der Liste für die Professoren sichtbar werden. Wenn neue Veranstaltungskombinationen entstanden sind, muss noch eine Kompetenzgruppenzugehörigkeit erstellt werden. Dies wird in den folgenden Abschnitten näher erläutert.

Home > administration > import semester list

Import semesterlist:

Filename: 'testing.csv' (443 kb) successfully uploaded! (Time elapsed: 04 ms)

Correct the red marked Errorlist at the bottom of the page!

semester	unit	course of studies	group	course	SWS	type of course	type of test	course name
WT 2013	D-inf	Bachelor	B-AI3	TS	5d	SeU	P	Theoretische Informatik Praktikum
aha	D-inf	Bachelor	B-AI3	TS	0	Prak	P	Theoretische Informatik Praktikum
aha	D-inf	Bachelr	B-AI3	TS	0	Prak	P	Theoretische Informatik Praktikum
WS ddd	D-inf	Bachelor	B-AI3	TS	0	Pra	P	Theoretische Informatik Praktikum
WS 1111	D-inf	Bachelor	B-AI3	TS	0	Pra	P	Theoretische Informatik Praktikum
WS 2111	D-inf	Bachelor	B-AI3	TS	0	Pra	P	Theoretische Informatik Praktikum

row:1 col:1 **WT 2013** --> allowed values: ([SS|WS][0-9+]{1,4} with number > 2000 ... e.g. 'WS 2013')

row:1 col:6 **5d** --> allowed values: (Integer)

row:2 col:1 **aha** --> allowed values: ([SS|WS][0-9+]{1,4} with number > 2000 ... e.g. 'WS 2013')

row:3 col:1 **aha** --> allowed values: ([SS|WS][0-9+]{1,4} with number > 2000 ... e.g. 'WS 2013')

row:3 col:3 **Bachelr** --> allowed values: (Bachelor,Master,Dualer Bachelor)

row:4 col:1 **WS ddd** --> allowed values: ([SS|WS][0-9+]{1,4} with number > 2000 ... e.g. 'WS 2013')

row:4 col:7 **Pra** --> allowed values: (SeU,Prak,Pro,Sem,Üb,KgPro,StAr,Bath,Math,Dar,Vorl)

row:5 col:1 **WS 1111** --> allowed values: ([SS|WS][0-9+]{1,4} with number > 2000 ... e.g. 'WS 2013')

row:5 col:7 **Pra** --> allowed values: (SeU,Prak,Pro,Sem,Üb,KgPro,StAr,Bath,Math,Dar,Vorl)

row:6 col:7 **Pra** --> allowed values: (SeU,Prak,Pro,Sem,Üb,KgPro,StAr,Bath,Math,Dar,Vorl)

Abbildung D.11.: Semesterliste importieren

### D.1.3.3. Erstellen von Veranstaltungen

Das Erstellen von einzelnen Veranstaltungen (Abbildung D.12) verhält sich ähnlich des Veranstaltungsimports. Aus bereits vorhandenen Daten werden für die einzelnen Felder Dropdown-Listen erstellt, über die ausgewählt werden können.

Home > administration > import semester list > create course

Create course:

semester	unit	course of studies	group	course	sws	type of Course	type of Test	course name
WS 2012	D-inf	*****	*****	*****	*****	*****	*****	*****
WS 2012	D-inf	Master Bachelor Dualer Bachelor						

Create new values:

value	field	action
	semester	transfer

Abbildung D.12.: Erstellen von einer Veranstaltung

Um Daten einfügen zu können, die noch nicht in der Datenbank gespeichert wurden, wird über ein angefügtes Element die Möglichkeit gegeben, den Wert manuell setzen zu können. Hierbei werden nur jene Felder berücksichtigt, die variabel sind. Sobald bei



allen Feldern eine Wertzuweisung stattgefunden hat, wird ein Button zum Erstellen der Veranstaltung angezeigt.

#### D.1.3.4. Bilden von Kompetenzgruppen

Das Erstellen von Kompetenzgruppen (Abbildung D.13), ermöglicht die Zuordnung von zusammengehörigen Veranstaltungen zu einer Kompetenz.

Hierdurch wird die Menge der wählbaren Kompetenzen auf die Kompetenzgruppen eingeschränkt. Durch Auswahl einer Kompetenz wird nun eine Veranstaltung und das dazugehörige Praktikum in der Liste der verfügbaren Veranstaltungen dargestellt.

Home > administration > import semester list > bind expertise

Assign each course to an expertise-group:

<u>course</u>	<u>course name</u>	<u>expertise</u>	
GT	Grundlagen der Informatik	GI <input type="button" value="v"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>
AD	Algorithmen und Datenstrukturen	AD	
ADP	Praktikum Algorithmen und Datenstrukturen	AD	
DB	Datenbanken	DB	
DBP	Praktikum Datenbanken	DB	
GI	Grundlagen der Informatik	GI	
GIÜ	Übung Grundlagen der Informatik	GI	
GTP	Praktikum Grundlagen der Informatik	GI	
LB	Logik und Berechenbarkeit	LB	
LBP	Praktikum Logik und Berechenbarkeit	LB	

[1](#) [2](#) [3](#)

25 courses

Abbildung D.13.: Bildung von Kompetenzgruppen

#### D.1.3.5. Einstellungen

Die Einstellungen (Abbildung D.14) für das laufende System betreffen das Festlegen der Deadline, das Aktivieren von Veranstaltungen, das Setzen von Markierungen und Mail- sowie Logging-Einstellungen.

Home - administration - preferences

### set new Deadline

< January 2012 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

The **Warning-Date** will activate the mail delivery ('warning mail' & 'workload mail'). The date will be set  weeks and  days before the **Deadline-Date**.

**Note:** You cannot change the Deadline-Date between the Warning-Date and the finish of the running semester.

Deadline is changeable!

---

### activate upcoming Semester

WS 2011    2 Entries   

---

### Percentage markings

minimum:     maximum:    

Workload in [Members-Section](#) will be marked as followed:  
 0 to 5% will be marked **yellow**  
 5 to 85% will be marked **green**  
 85% and more will be marked **red**

---

### Mail settings

warning mail:   
 Summary of course status will be sent via email to all admins, if warning-date is reached.

deadline mail:   
 Every admin will get an email-notification, that the actual semester-deadline has been reached!

finished admin mail:   
 Every admin will get a notification that the assignment has been finished. Log-files will be attached.

finished mail:   
 Every member, who takes part in the actual semester will get an email-notification if the assignment has been finished!

workload mail:   
 Every member, who has a role-status and request-percentage is marked yellow, will get an email-notification at warning-date to request more courses. ( now)

---

### Log settings

The log-files will be send as an attachment to the admins, when the assignment has been finished ('finished admin mail'). Its recommended to leave the loggings checked, to make retracing easier.

course settings	expertise settings
request course: <input type="checkbox"/> Log every request on a course	select expertise: <input type="checkbox"/> Log every select on an area of expertise
delete request: <input type="checkbox"/> Log every deletion of a request on a course	deselect expertise: <input type="checkbox"/> Log every deselect of an area of expertise

Abbildung D.14.: Einstellungen

Über den angezeigten Kalender kann der Admin ein neues Datum für die Deadlinefestlegung der Veranstaltungswahl des aktuellen Semesters auswählen. Im rechten Bereich neben dem Kalender stellt das rot markierte Feld die Deadline dar und ein gelb markiertes Datum stellt das Warndatum dar. Bevor die Deadline gesetzt wird, kann über zwei Dropdown-Listen der Zeitraum angegeben werden, zu welchem das Warndatum relativ zum Deadline-Datum festgelegt werden soll.

Importierte Veranstaltungen und Semesterlisten, müssen aktiviert werden, um entgültig in der Auswahl der Semesterliste für Professoren zu erscheinen. Im deaktivierten Zustand sind diese lediglich in der allgemeinen Liste des Admins sichtbar. Allerdings nur dann sobald vorher schon aktivierte Veranstaltungen für das betreffende Semester aktiviert wurden.

Das Setzen von Markierungen betrifft den Benutzerverwaltungsbereich des Admins. Hier existieren zwei Spalten, die für eine prozentuale Darstellung der Anfragen und Buchungen des betreffenden Benutzers zuständig sind.

Um während der Veranstaltungszuteilung eine bessere Übersicht zu gewährleisten, können farbliche Markierung vorgenommen werden. Hierbei sind die Werte zwischen Minimum und Maximum als optimaler Wert anzusehen und somit als grün gekennzeichnet. Gelb werden jene markiert, die sich zwischen 0 und dem Minimum befinden. Alle Werte, die das Maximum der Auslastung überschreiten, werden farblich rot markiert.

Die Email-Einstellungen sind für die gängigen Emails an die Professoren und die Admins zuständig.

### D.1.3.6. Benutzerübersicht

In der Benutzerübersicht (Abbildung D.15) sind alle registrierten Benutzer des Systems aufgelistet.

Home administration > members

user	email	role	competence	requester	securepw	max LVS	SS 2010(r)	SS 2010(b)			
administrator	kkizi@yahoo.de	admin	ok	ok	ok	18	22.22 %	0.00 %			
bbb	bbb@bbb.de	professor	ok	ok	failed	18	0.00 %	0.00 %			
ccc	ccc@ccc.de	professor	ok	ok	ok	18	0.00 %	0.00 %			
ddd	ddd@ddd.de	professor	ok	ok	ok	18	0.00 %	0.00 %			
eee	eee@ee.de	professor	ok	ok	ok	18	0.00 %	0.00 %			
fff	fff@fff.de	professor	failed	failed	ok	18	0.00 %	0.00 %			
ggg	ggg@gggg.de	professor	failed	failed	ok	16	0.00 %	0.00 %			
hhh	hh@hh.de	professor	failed	failed	ok	18	0.00 %	0.00 %			
iii	ii@ii.de	professor	failed	failed	ok	18	0.00 %	0.00 %			
jjj	jjj@jjj.de	*no role*	failed	failed	ok	18	0.00	0.00			

1 2 3

28 user

no role\*  
 admin  
 professor

entries: 10  UserName  Email  role  competence  requester  securepw  created  activity  maxLVS  requested  booked

Abbildung D.15.: Benutzerübersicht

In diesem Menü kann eine Rollenzuweisung an registrierte Benutzer vorgenommen werden. Zudem ist außerdem die Möglichkeit gegeben, einem Benutzer bereits bestehende Rolle wieder zu entziehen. Dies würde dazu führen, dass der jeweilige Benutzer nicht mehr in der Lage ist, sich im System anzumelden. Auch das komplette Löschen eines Benutzers ist hier möglich, jedoch ist die Variante des Rollenentzugs vorzuziehen, da hierdurch gespeicherte Daten des Benutzers nicht verloren gehen. Sollte dieser Nutzer zu einem späteren Zeitpunkt erneut registriert werden, muss kein neues Profil erstellt werden. Das Löschen des eigenen Accounts und das Entziehen der eigenen Rolle sind für den Admin nicht möglich.

Die im vorherigen Abschnitt erwähnten Markierungen der prozentualen Darstellung der Anfragen und Buchungen für das aktuelle Semester, wird in diesem Bereich genutzt.

Die Spalten 'competence' und 'requester' geben Auskunft darüber, ob der Benutzer sich schon Fähigkeiten zugewiesen hat und ob er Veranstaltungen aus dem aktuellen Semester gebucht hat.

Die Sicherheit des Passwortes wird über die Anzahl der Anmeldeversuche festgelegt. Bei mindestens drei falschen Passwort-Eingaben, wird dieser Bereich rot markiert. Falls das letzte Anmeldedatum dem Datum der Erstellung des Accounts gleichen sollte, werden die Datumsangaben blau markiert, um einen Hinweis zu geben, dass der Benutzer das Passwort noch nicht geändert hat.

Die Anzahl der Spalten und die Anzahl der Einträge pro Seite können hier für eine bessere

Übersicht eingestellt werden. Die Zeilen können zusätzlich über die blau markierten Spaltenüberschriften sortiert werden.

Das Kontaktieren eines Benutzers (Abbildung D.16) wird über diese Maske ermöglicht.

Abbildung D.16.: Benutzer kontaktieren

Die Eingabe eines individuellen Betreffs und Textkörpers ist hierüber möglich.

### D.1.3.7. Benutzerdetails ändern

Dieser Bereich zeigt dem Admin eine Übersicht über die Profildaten der Benutzer (Abbildung D.17).

Home > administration > members > change memberdetails

<a href="#">username</a>	<a href="#">firstname</a>	<a href="#">lastname</a>	<a href="#">weekly hours</a>	<a href="#">email</a>	
administrator	renate	buth	18	kkizi@yahoo.de	
bbb	bbbj	bbb	18	bbb@bbb.de	
ccc	ccc	ccc	18	ccc@cccc.de	
ddd	ddd	ddd	18	ddd@ddd.de	
eee	eee	eee	18	eee@ee.de	
fff	fff	fff	18	fff@fff.de	
ggg	gggg	gg	16	ggg@ggggg.de	
hhh	hhh	hhh	18	hh@hh.de	
iii	iii	iii	18	ii@ii.de	
jjj	jjj	jjj	18	jjj@jjj.de	

1 2 3  
28 user

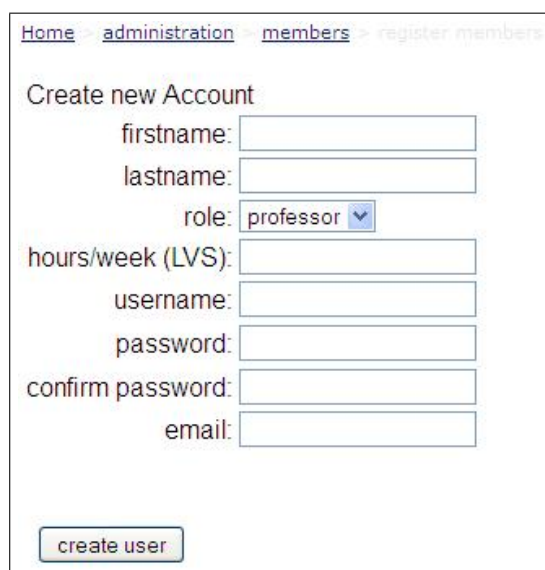
Abbildung D.17.: Benutzerdetails ändern

Die Ansicht wird seitenweise dargestellt und lässt sich über die blau markierten Spalten sortieren.

Neben den Namensfeldern, die ein Benutzer auch selbst bearbeiten kann, kann hier die wöchentliche Höchstlast an LVS und die Emailadresse geändert werden.

### D.1.3.8. Registrieren von Benutzern

Das Registrieren von Benutzern (Abbildung D.18), kann ausschließlich vom Admin vorgenommen werden.



The screenshot shows a web interface for creating a new account. At the top, there is a breadcrumb trail: Home > administration > members > register members. Below this, the heading 'Create new Account' is displayed. The form contains several input fields: 'firstname:', 'lastname:', 'role:' (with a dropdown menu currently showing 'professor'), 'hours/week (LVS):', 'username:', 'password:', 'confirm password:', and 'email:'. At the bottom of the form is a button labeled 'create user'.

Abbildung D.18.: Registrierung von Benutzern

Das Initial-Passwort wird vom Admin selbst bestimmt. Sobald der Benutzer erstellt wird, sendet das System eine Einladung an seine angegebene Emailadresse. Diese enthält eine Aufforderung an den Benutzer, sich anzumelden und sein zugewiesenes Passwort zu ändern.

### D.1.3.9. Passwort Recovery

Für den Fall, dass ein Benutzer sein Passwort vergessen haben sollte, gibt es normalerweise eine Passwort-Recovery Funktion (Abbildung D.19), die es dem Benutzer ermöglicht, über eine Geheimfrage ein neues Passwort anzufordern.

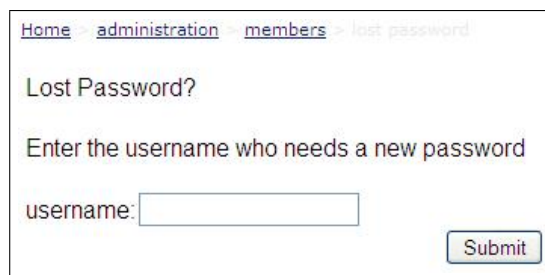


Abbildung D.19.: Passwort wiederherstellen

Diese Funktion ist in diesem System nicht gegeben. Wenn ein Benutzer sein Passwort vergessen haben sollte, kann er dies nur über den zuständigen Admin zurücksetzen lassen. Hiefür muss der Admin den Usernamen des Benutzers eingeben, so dass dieser eine Email mit einem neu generierten Passwort zugeschickt bekommt.

### D.1.3.10. Semesterliste einsehen

In diesem Bereich (Abbildung D.20) werden die Veranstaltungen des aktuellen Semesters dargestellt und im Regelfall nach Ablauf der Deadline zugeteilt.

Über die Dropdown-Liste können zudem Veranstaltungen der kommenden Semester eingesehen werden.

Die Anzahl der Spalten und die Anzahl der Einträge pro Seite können für eine passendere Übersicht eingestellt werden.

Man kann zusätzlich über die blau markierten Spaltenüberschriften die Zeilen sortieren.

Über den Button 'assign a member' kann entweder die ganze Veranstaltung oder auch nur eine Teilsumme der LVS-Stunden der Veranstaltung an einem in der Dropdown-Liste angezeigten Professor übertragen werden. Wird nur eine Teilsumme der Veranstaltung an den ausgewählten Benutzer übertragen, wird die Veranstaltung automatisch gesplittet. Die andere Teilsumme der Veranstaltung erscheint in einer farblich hinterlegten Zeile in der Veranstaltungsauswahl und kann nun einem anderen Professor zugeordnet werden.

Die Auswahl in der Dropdown-Liste für die Benutzer, enthält nur für die Veranstaltung fähige Professoren. Diese haben alle eine Kennzeichnung vor ihrem Kürzel, welches Aussagen über den aktuellen Buchungsverlauf des Professors geben. Man erkennt hier die Summen

Home > administration > view semester list

SS 2011

unit	group	course	SWS	LVS	GC	type	course name	able	requests	staff					
D-inf	B-A11	GI	3	3.00	1	SeU	Grundlagen der Informatik	5	0						
D-inf	B-A11	GIÜ	1	1.00	3	Prak	Übung Grundlagen der Informatik	5	0						
D-inf	B-A12	PRP2	1	1.00	3	Prak	Praktikum Programmieren 2	5	0						
D-inf	B-A12	PR2	3	3.00	1	SeU	Programmieren 2	5	0						
D-inf	B-A13	AD	3	3.00	1	SeU	Algorithmen und Datenstrukturen	5	0						
D-inf	B-A13	ADP	1	1.00	3	Prak	Praktikum Algorithmen und Datenstrukturen	5	0						
D-inf	A15	PO	8	8.00	1	Pro	Projektmodul	5	0						
D-inf	B-A12	RMP	3	3.00	1	SeU	Rechnerstrukturen und Maschinennahe Programmierung	5	0						
D-inf	B-A12	RMPP	1	1.00	3	Prak	Praktikum Rechnerstrukturen und Maschinenn. Programm.	5	0						
D-inf	B-T11	GT	3	3.00	1	SeU	Grundlagen der Informatik	3	0						

1 2

15 events

entries: 10 semester  unit  group  course  svs  lvs  gc  course of studies  type  course name  able  requests  staff

Abbildung D.20.: Semesterliste einsehen

der noch angefragten Veranstaltungen, der bereits gescheiterten Wahlen, der erfolgreich gebuchten Veranstaltungen und die maximale LVS-Last des Professors für das ausgewählte Semester.

Es ist möglich eine Veranstaltung komplett aus dem Veranstaltungsangebot zu löschen oder auch die Gruppengröße einer Veranstaltung neu festzulegen. Die Gruppengröße steht standardmäßig auf 1, bei Übungen/Praktikas auf 3. Eine Änderung dieser Werte wirkt sich automatisch auf die LVS aus, die je nach Gruppengröße variiert.

Aus der Tabelle lässt sich für jede Veranstaltung die Summe der fähigen Professoren und die Summe der Anfragen ablesen. Details zu beiden können aufgerufen werden und geben unter anderem Aufschluss über den Zeitpunkt der Buchungsanfrage.

In dem Punkt Detailsview können bereits zugeordnete Veranstaltungen wieder freigegeben werden und stehen somit wieder zur Veranstaltungswahl durch die Professoren zur Verfügung.



### **D.1.4. Zeitverhalten**

Durch das Nutzen einer Deadline existieren zwei Zustände, zwischen denen das System unterscheidet, die Zeitspanne, in der sich das System im Deadline-Modus befindet und die Zeitspanne, in der sich das System im Normal-Modus befindet.

Neben diesen Zeitspannen existieren zwei Zeitpunkte, an denen das System jeweils einmalige Handlungen vornimmt. Diese wären das Erreichen der Deadline und der davor liegende Warnzeitpunkt.

#### **D.1.4.1. Deadline-Modus**

Der Deadline-Modus beginnt nach Ablauf der Deadline-Frist und endet mit dem Abschluss der Veranstaltungsverteilung durch den Admin.

Die Deadline wird durch den Admin jedes Semester manuell gesetzt, wobei hier gleichzeitig ein Warnzeitpunkt angegeben wird, an welchem die Beteiligten durch entsprechende Emails über den Status der aktuell angefragten Veranstaltungen vor Ablauf der Deadline informiert werden. Das Setzen der Deadline-Frist ist in der Zeit zwischen dem Warnzeitpunkt und Abschluss des Semesters nicht möglich.

Im Deadline-Modus sind bestimmte Funktionalitäten des Systems eingeschränkt, so ist es einem Professor nicht mehr möglich Veranstaltungen zu wählen oder abzuwählen.

#### **D.1.4.2. Normal-Modus**

Der Normal-Modus beschreibt den Zustand zwischen dem Abschluss eines Semesters durch den Admin und dem Erreichen der nächsten Deadline. In diesem Modus befindet sich das System den überwiegenden Teil der Zeit. Hier können wie gewohnt Anfragen auf Veranstaltungen getätigt werden.

### D.1.4.3. Zeitpunkte

Die Realisierung der Zeitpunkte wird über eine mitgeführte Statusvariable erreicht. Das System lässt sich indirekt in vier Zustände versetzen:

- A - Normal-Modus  
Das System befindet sich im Normal-Modus und der Statuswert ist 0.
  
- B - Warnzeitpunkt  
Die Systemzeit hat den Zeitwert des Warnzeitpunktes überschritten und der Statuswert beträgt 0.  
Aktionen werden ausgeführt und der Statuswert wird auf 1 gesetzt.
  
- C - Deadline  
Die Systemzeit hat den Zeitwert der Deadline überschritten und der Statuswert beträgt 1.  
Aktionen werden ausgeführt und der Statuswert wird auf 2 gesetzt.
  
- D - Deadline-Modus  
Durch Erreichen der Deadline befindet sich das System im Deadline-Modus.  
Durch Abschluss des Semesters über den Admin, wird der Statuswert auf 3 gesetzt.  
Wird eine neue Deadline-Frist festgesetzt, wird der Statuswert wieder auf 0 zurückgesetzt.

Die Prüfung der Systemzeit und der Statusvariablen findet beim Betreten der Startseite eines Benutzers statt.

# Glossar

**Cloud-Computing** Cloud-Computing beschreibt das Prinzip IT-Ressourcen nur dann bereitzustellen und abzurechnen, wenn sie wirklich gebraucht werden. Hierbei werden Rechenkraft und Speicher, Plattformen inkl. Programmierumgebung und Software auf Abruf in einer sogenannten Cloud dargestellt.

**Datagrid** Ein Datagrid ist ein ASP.NET-Steuerelement, welches Elemente aus einer Datenquelle in einer Tabelle anzeigt. Das Editieren und Sortieren dieser Elemente wird unterstützt.

**IIS** Internet Information Service (kurz IIS) ist eine von Microsoft entwickelte Webserver-Applikation

**LVS** Über Lehrveranstaltungsstunden (kurz LVS) wird der Umfang der Lehrverpflichtung angegeben. Errechnet werden diese über die SWS mit einem Faktor. Der Faktor ist abhängig von der Veranstaltungsart.

**Pushmail** Pushmail ist ein Verfahren welches eingehende Emails direkt an ein mobiles Endgerät, welches mit dem Mailserver verbunden ist, weiterreicht. Man erhält umgehend Information über eingehende Emails, da kein manuelles Abrufen mehr notwendig ist.

**Sitemappath** Auch bekannt als *Breadcrumb*, ermöglicht einem Benutzer leichter über eine Website zu navigieren, indem diesem der aktuellen Pfad auf dem er sich auf der Website befindet, angezeigt wird

**Siteübersicht** Eine Siteübersicht wird zur Darstellung einer logischen Struktur einer Website verwendet

**SSL** Secure Sockets Layer (kurz SSL) ist ein Netzwerkprotokoll zur sicheren Übertragung von Daten

**SWS** Eine Semesterwochenstunde (kurz SWS) wird an Hochschulen benutzt, um den Zeitaufwand für eine Lehrveranstaltung anzugeben. Eine SWS bedeutet das die Veranstaltung 45 Minuten in der Woche gelehrt wird.

**Synchronisierung** Synchronisierung bezeichnet das zeitliche aufeinander Abstimmen von Vorgängen.

**Treeview** Treeview ist ein graphisches Steuerelement, welches Informationen in einer hierarchischen Baumstruktur anzeigt. Die einzelnen Knoten sind jeweils reduzierbar.

**Url-Tweaking** URL-Tweaking bezeichnet einen Angriff auf eine Webseite, bei dem über die manuelle Manipulation der URL versucht wird, Zugang zu einem geschützten Bereich zu bekommen

**VPN** Über ein Virtual Private Network (kurz VPN) kann ein Arbeitsplatz mit beispielsweise einem Firmennetzwerk verbunden werden. Das auf dem Arbeitsplatz laufende Hilfsprogramm (VPN-Client) sorgt dabei für eine sichere Übertragung aller Datenpakete.

**Workload** Workload beschreibt das wöchentliche Arbeitspensum

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 14. Juni 2012

Ort, Datum

Unterschrift