

# **Pflichtenheft**

## **SEDA-WI-Proj-B**

### **SF GmbH**

Denis Hamann      Matr.-Nr. 1684873

Anna Kupfer      Matr.-Nr. 141515

Hannes Stadler      Matr.-Nr. ?

Christian Hindelang      Matr.-Nr. ?

Otto-Friedrich-Universität Bamberg

Version: v0.1 - 29. Oktober 2012

## Inhaltsverzeichnis

<b>1</b>	<b>Preamble</b>	<b>2</b>
<b>2</b>	<b>Preamble</b>	<b>3</b>
<b>3</b>	<b>Preamble</b>	<b>4</b>
<b>4</b>	<b>Produktumgebung</b>	<b>5</b>
4.1	Software . . . . .	5
4.2	Hardware . . . . .	5
4.3	Orgware . . . . .	6
4.4	Produkt-Schnittstellen . . . . .	6
<b>5</b>	<b>Funktionen</b>	<b>7</b>
5.1	Prozesse mit Dateneingabe . . . . .	7
5.2	Listen . . . . .	11
5.3	Berichte . . . . .	12
<b>6</b>	<b>Daten</b>	<b>14</b>
<b>7</b>	<b>Leistungen</b>	<b>16</b>

## Abbildungsverzeichnis

1	SERM der Daten-Architektur . . . . .	15
---	--------------------------------------	----

# 1 Preamble

We'd like to address the reader of this documentation before going into the details of this documentation that the project itself encountered several challenges which couldn't all be solved in a satisfying way. These lead to a program with a limited functionality and a documentation with some major gaps. For reasons and the progress itself which lead to this situation please refer to chapter 8.

## 2 Preamble

We'd like to address the reader of this documentation before going into the details of this documentation that the project itself encountered several challenges which couldn't all be solved in a satisfying way. These lead to a program with a limited functionality and a documentation with some major gaps. For reasons and the progress itself which lead to this situation please refer to chapter 8.

### 3 Preamble

We'd like to address the reader of this documentation before going into the details of this documentation that the project itself encountered several challenges which couldn't all be solved in a satisfying way. These lead to a program with a limited functionality and a documentation with some major gaps. For reasons and the progress itself which lead to this situation please refer to chapter 8.

## 4 Produktumgebung

Im folgenden wird die Umgebung beschrieben, in der die spätere Software eingesetzt werden wird.

### 4.1 Software

Die Software beschreibt die vorzuhaltende Software-Umgebung um eine problemlose Ausführung des Programms sicherzustellen.

Als Betriebssystem kommt Windows 7 (Professional) (x64 - 64bit) zum Einsatz. Dieses hat die neusten Updates vorzuweisen und eine gängige aktuelle Antiviren Lösung installiert zu haben.

Als Laufzeitumgebung wird auf die Java Virtual Machine Version 7 von Oracle gesetzt. Hierzu ist das entsprechende Java Runtime Environment in Version 7 vorzuhalten. Es muss sichergestellt werden, dass die entsprechenden Klassenpfade, sowie Systempfade gesetzt sind, sodass der prozess java als auch javaw im CLI bekannt ist. Darüber hinaus müssen .jar-Dateien mit der JRE verknüpft sein um einen Start der Anwendung per Doppelklick zu gewährleisten. Entsprechende Einstellungen sind notfalls in der Registry vorzunehmen.

```
[HKEY_CLASSES_ROOT\.jar]
```

```
@="jarfile"
```

```
[HKEY_CLASSES_ROOT\jarfile]
```

```
@="Executable Jar File"
```

```
[HKEY_CLASSES_ROOT\jarfile\shell]
```

```
[HKEY_CLASSES_ROOT\jarfile\shell\open]
```

```
[HKEY_CLASSES_ROOT\jarfile\shell\open\command]
```

```
@="\"C:\\Program Files\\Java\\jre7\\bin\\javaw.exe\" -jar \"%1\" %*"
```

Datenbank: PostgreSQL 9.2.1

Shell: Windows Shell (explorer.exe) Fensterprogramm, kein CLI Programm

### 4.2 Hardware

CPU: Intel x86/x65 Architektur - >1Ghz >=1GB DDR2RAM HDD: >100MB Maus & Tastatur Netzwerkanschluss (für DB Anbindung)

### 4.3 Orgware

Eine entsprechende Netzwerkinfrastruktur muss vorhanden sein um eine Verbindung zur Datenbank herstellen zu können. Die Datenbank (Postgresql) muss auf einem entsprechenden Server installiert sein und als Dienst laufen. Es muss sicher gestellt werden, dass entsprechende Router und Firewalls konfiguriert sind und eine Verbindung in beide Richtungen ermöglichen. Es wird davon ausgegangen, dass die Hausverwaltung, sowie die Dozenten entsprechende Informationen zur Räumen und Lehrveranstaltungen korrekt und zeitnah in das System einpflegen um eine sinnvolle Nutzung zu ermöglichen.

### 4.4 Produkt-Schnittstellen

## 5 Funktionen

### 5.1 Prozesse mit Dateneingabe

/F01/ **Prozess mit Dateneingabe:** Benutzer-Authentifizierung

**Akteur:** Alle

**Beschreibung:** Alle Benutzer des Systems haben die Möglichkeit sich mittels einer Login-Funktion am System anzumelden. Hierdurch werden Sie einer der möglichen Benutzergruppen zugeordnet. Nicht angemeldete Benutzer gelten als Mitglieder der Benutzergruppe SStudenten".

**Eingabevalidierung:** Nutzer müssen eine gültige Nutzerkennung mit dazu passendem Passwort eingeben. Diese werden von der Verwaltung festgelegt. Wünschenswert wäre, dass Studenten sich auch am System anmelden können (siehe /FW61/) und hierzu ihre normale Hochschulkennung und Passwort verwenden könnten. Im Falle von falschen Zugangsdaten erhält der Benutzer eine Meldung, die ihn über den Fehlversuch informiert. Es gibt nur eine Meldung, sodass für ihn nicht ersichtlich ist, ob das Passwort falsch war oder er schon eine ungültige Nutzerkennung eingegeben hat.

/F10/ **Prozess mit Dateneingabe:** Verwaltung von Raumdaten

**Akteur:** Verwaltung

**Beschreibung:** Mitglieder der Verwaltung können Räume erstellen, bearbeiten und löschen.

**Eingabevalidierung:** Es können beliebige Räume erstellt, gelöscht oder bearbeitet werden. Bei der Erstellung darf die Raumnummer (siehe Datenobjekt /D10/) nicht schon im System vorhanden sein (die systeminterne Raum ID ist für Nutzer nicht selbst wählbar, so wie das auch im Folgenden nie der Fall sein wird), anderenfalls wird der Nutzer mit einer Fehlermeldung darauf hingewiesen. Die übrigen Attribute eines Raums (/D10/) werden sofern möglich nach Logik und passender Größe überprüft und bei fehlerhaften Eingaben wird ein Bearbeitungs- oder Erstellvorgang abgebrochen, sowie der Nutzer darauf hingewiesen. So kann eine Kapazität beispielsweise nur eine ganze positive Zahl sein.



---

/F20/ **Prozess mit Dateneingabe:** Verwalten von Lehrveranstaltungen

**Akteur:** Dozenten

**Beschreibung:** Dozenten können Lehrveranstaltungen, die sie oder andere Mitarbeiter ihres Lehrstuhls in einem Semester halten eintragen, bearbeiten und löschen. (Diese sind nur für angehörige ihres Lehrstuhls sichtbar und bearbeitbar. Für Studenten werden sie erst sichtbar und verwendbar, wenn für sie ein Raum gebucht wurde und dies bestätigt wurde.)

**Eingabevalidierung:** Es können beliebige Lehrveranstaltungen erstellt, gelöscht oder bearbeitet werden. Doppelerstellungen sind möglich, falls dies in speziellen Fällen von Dozenten benötigt wird (durch die systeminterne ID ist ein fehlerfreier Systembetrieb gewährleistet). Die übrigen Attribute einer Lehrveranstaltung (/D50/) werden sofern möglich nach Logik und passender Größe überprüft und bei fehlerhaften Eingaben wird ein Bearbeitungs- oder Erstellvorgang abgebrochen, sowie der Nutzer darauf hingewiesen.

/FW21/ **Prozess mit Dateneingabe:** Verwalten von Live-Ticker Meldungen

**Akteur:** Dozenten und Verwaltung

**Beschreibung:** Dozenten und Mitglieder der Verwaltung können Meldungen erstellen, die Studenten im Live-Ticker zu sehen bekommen. Es lässt sich ein Zeitraum eingeben, in dem die Meldung sichtbar ist. Wünschenswert wäre hierbei, dass Studenten ein Profil haben und eingeloggten Studenten nur relevante Meldungen (zu einer Lehrveranstaltung, einem Raum oder einem Lehrstuhl) angezeigt werden.

**Eingabevalidierung:** Es sollten beliebige Live-Ticker Meldungen erstellt, gelöscht oder bearbeitet werden können. Die übrigen Attribute einer Ticker-Meldung (/DW80/) werden sofern möglich nach Logik und passender Größe überprüft und bei fehlerhaften Eingaben wird ein Bearbeitungs- oder Erstellvorgang abgebrochen, sowie der Nutzer darauf hingewiesen.

**/F30/ Prozess mit Dateneingabe:** Buchen von Räumen**Akteur:** Dozenten

**Beschreibung:** Dozenten können ihre Lehrveranstaltung für ein Semester zu einer bestimmten Zeit an einem bestimmten Werktag in einem Raum eintragen. Eine Zeiteinheit beträgt dabei 2 Stunden, kann immer nur voll belegt werden und die Zählung der Einheiten geht von 8 Uhr morgens bis 20 Uhr abends (womit ein Tag aus 6 Einheiten besteht). Buchungen müssen seitens der Verwaltung bestätigt werden (siehe /F40/).

**Eingabevalidierung:** Es sind Buchungen für bereits eingetragene Lehrveranstaltungen möglich. Eine Buchung umfasst genau eine Zeiteinheit zu einem Raum an einem Tag. Es sind beliebig viele Buchungen für eine Lehrveranstaltung möglich. Eine Buchung darf sich nicht mit einer bereits von der Verwaltung bestätigten Buchung im gleichen Raum zur gleichen Zeit am gleichen Tag überschneiden, sonst wird der Vorgang abgebrochen und dem Nutzer ein entsprechender Fehler gezeigt. Die übrigen Attribute einer Buchung (/D60/) werden sofern sie vom Dozenten anzugeben sind und möglich sind nach Logik und passender Größe überprüft und bei fehlerhaften Eingaben wird ein Bearbeitungs- oder Erstellvorgang abgebrochen, sowie der Dozent darauf hingewiesen.

**/F40/ Prozess mit Dateneingabe:** Verwaltung von Raumbellegungen inklusive Konfliktlösung**Akteur:** Verwaltung

**Beschreibung:** Mitglieder der Verwaltung können die Belegungswünsche zu Räumen seitens der Dozenten bestätigen, abweisen oder umlegen. Im Falle eines Konflikts, bzw. einer Ablehnung können sie dem Dozenten explizite Vorschläge machen (und hierzu Erläuterungen hinterlegen), die noch vor den vom System generierten automatischen Vorschlägen dem Dozenten angezeigt werden.

**Eingabevalidierung:** Es ist nur das zuweisen eines Status, wie bei /DW80/ spezifiziert, möglich. Dieser ist vom System vorgegeben so dass Falscheingaben nicht möglich sind. Im Falle einer Ablehnung kann zusätzlich eine Nachricht übermittelt werden, die bezüglich Größe validiert wird. Im Fehlerfall wird der Nutzer darauf hingewiesen und der Vorgang abgebrochen.

- /F50/ **Prozess mit Dateneingabe:** Verwaltung von Nutzerdaten  
**Akteur:** Verwaltung  
**Beschreibung:** Mitglieder der Verwaltung können Benutzer (Dozenten und Verwaltungsmitglieder) in das System einpflegen, sie bearbeiten oder löschen.  
**Eingabevalidierung:** Das Eintragen und Bearbeiten umfasst die in /D30/ spezifizierten Attribute (außer dem letzten Login und dem Salt, welche automatisch vom System generiert werden und wie gehabt der systeminternen ID). E-Mail und Benutzerkennung dürfen nicht schon im System vorhanden sein, anderenfalls erscheint eine entsprechende Fehlermeldung und der Vorgang wird abgebrochen. Das Passwort wird beim Erstellen und Bearbeiten direkt zur Laufzeit zusammen mit einem benutzerspezifischen zufälligen Wert (dem SSalt") in einen SHA512-Hash umgewandelt und nur als solcher persistent gespeichert. Es muss mindestens 8 Zeichen umfassen. Alle Attribute werden sofern möglich nach Logik und passender Größe überprüft und bei fehlerhaften Eingaben wird ein Bearbeitungs- oder Erstellvorgang abgebrochen, sowie der Nutzer darauf hingewiesen.
- /F51/ **Prozess mit Dateneingabe:** Verwaltung von Lehrstühlen  
**Akteur:** Verwaltung  
**Beschreibung:** Mitglieder der Verwaltung können Lehrstühle in das System einpflegen, sie bearbeiten oder löschen.  
**Eingabevalidierung:** Das Eintragen und Bearbeiten umfasst die in /D20/ spezifizierten Attribute. Der Lehrstuhlname darf nicht bereits im System vorhanden sein, anderenfalls erscheint eine entsprechende Fehlermeldung und der Vorgang wird abgebrochen. Alle Attribute werden sofern möglich nach Logik und passender Größe überprüft und bei fehlerhaften Eingaben wird ein Bearbeitungs- oder Erstellvorgang abgebrochen, sowie der Nutzer darauf hingewiesen.
- /F60/ **Prozess mit Dateneingabe:** Belegung von Lehrveranstaltungen  
**Akteur:** Studenten  
**Beschreibung:** Studenten können sich für Lehrveranstaltungen einschreiben (diese Einschreibungen werden nicht persistent im System gespeichert). Überschneidungen (Belegung zur gleichen Zeit) ist möglich, das System generiert aber einen Warnhinweis.  
**Eingabevalidierung:** Das System bietet nur die Möglichkeit eine öffentliche Veranstaltung als hinzuzufügen oder diese wieder zu entfernen. Eine vom Benutzer erstellte Eingabe und somit eine Validierung ist nicht von Nöten. Wünschenswert wäre die persistente Speicherung dieser Zuordnungen wie sie durch die Funktion /FW61/ möglich wäre und im Datenobjekt /DW70/ spezifiziert ist.

/FW61/ **Prozess mit Dateneingabe:** Profil-Verwaltung (inkl. Speichermöglichkeiten von Belegungen)

**Akteur:** Studenten

**Beschreibung:** Studenten sollten die Möglichkeit bekommen, ein Profil von sich anzulegen, welches persistent gespeichert wird und mit welchem ihre Belegungen dauerhaft verknüpft werden.

**Eingabevalidierung:** Die Profil-Bearbeitung umfasst die Attribute E-Mail, Passwort, Vor- und Nachname des Datenobjekts /D30/ für Studenten. Das Passwort wird dabei wie in /F50/ beschrieben gehasht und muss mindestens 8 Zeichen umfassen. Die E-Mail Adresse darf nicht bereits im System vorhanden sein, anderenfalls erscheint eine entsprechende Fehlermeldung und der Vorgang wird abgebrochen. Alle betreffenden Attribute werden sofern möglich nach Logik und passender Größe überprüft und bei fehlerhaften Eingaben wird ein Bearbeitungs- oder Erstellvorgang abgebrochen, sowie der Nutzer darauf hingewiesen.

## 5.2 Listen

/F70/ **Liste:** Lehrveranstaltungen eines Lehrstuhls

**Akteur:** Dozenten

**Beschreibung:** Dozenten können sich im System eine Liste von allen Lehrveranstaltungen, die ihrem Lehrstuhl zugeordnet sind anzeigen lassen. Hierbei steht ein Filter zur Verfügung, der nur bestimmte Semester oder bestimmte Dozenten anzeigt.

**Enthaltene Daten:** Name der Veranstaltung, Dozent, Semester, Tag(e), Zeit(en), gebuchter Raum, Freigabestatus, *wünschenswert eingeschriebene Studenten*

/F80/ **Liste:** Buchungen und Buchungswünsche

**Akteur:** Verwaltung

**Beschreibung:** Mitglieder der Verwaltung können sich im System eine Liste mit allen gebuchten Räumen, sowie noch nicht freigegebenen oder abgelehnten gebuchten Räumen anzeigen lassen. Hierbei steht ein Filter zur Verfügung, der es erlaubt nach Semester, Freigabestatus, Tag, Raum oder Lehrstuhl zu filtern. Noch nicht bearbeitete Raumbuchungen (weder freigegeben noch abgelehnt) werden an erster Stelle angezeigt.

**Enthaltene Daten:** Name der Veranstaltung, Dozent, Lehrstuhl, Semester, Tag(e), Zeit(en), gebuchter Raum, Freigabestatus, ggf. Verweis auf überschneidende Buchungen, *wünschenswert eingeschriebene Studenten*

- /F90/     **Liste:** Nutzer  
          **Akteur:** Verwaltung  
          **Beschreibung:** Mitglieder der Verwaltung können sich im System eine Liste mit allen Benutzern des Systems anzeigen lassen. Es steht ein Filter zur Verfügung, der es erlaubt nach Vor- oder Nachnamen sowie der Nutzerzugehörigkeit zu filtern.  
          **Enthaltene Daten:** Nutzerkennung, Vorname, Nachname, Nutzerzugehörigkeit (Lehrstuhl, Verwaltung oder ggf. auch Student), Zahl der Lehrveranstaltungen im aktuellen Semester, Datum des letzten Logins
- /F100/    **Liste:** Lehrveranstaltungen  
          **Akteur:** Studenten (und andere)  
          **Beschreibung:** Studenten (aber auch anderen Nutzern) können sich im System eine Liste aller durch die Verwaltung freigegebenen Lehrveranstaltungen anzeigen lassen. Es steht ein Filter zur Verfügung, der es erlaubt nach Semester, Lehrstuhl, Dozent, Name der Veranstaltung, Tag, Zeit und Raum zu filtern.  
          **Enthaltene Daten:** Name der Veranstaltung, Dozent, Lehrstuhl, Semester, Tag(e), Zeit(en), gebuchter Raum, *wünschenswert eingeschriebene Studenten*
- /F110/    **Liste:** Live-Ticker Meldungen  
          **Akteur:** Studenten (sowie wünschenswerterweise Verwaltung und Dozenten mit Bearbeitungsfunktionen)  
          **Beschreibung:** Studenten (aber auch anderen Nutzern) bekommen im System automatisch Live-Ticker Meldungen angezeigt, die die nächsten (falls möglich zu ihren Stundenplänen passenden) Lehrveranstaltungen anzeigen, sowie wünschenswerterweise auch von Dozenten oder Verwaltung zu diesen erstellte Meldungen.  
          **Enthaltene Daten:** Lehrveranstaltungsname, Raum, Zeit oder Meldungstexte

## 5.3 Berichte

- /F120/    **Bericht:** Raumbellegung  
          **Akteur:** Verwaltung (und wünschenswerterweise andere Nutzer)  
          **Beschreibung:** Mitglieder der Verwaltung (und wünschenswerterweise auch andere Nutzer) können sich eine Übersicht zur Belegung eines bestimmten Raumes in einem Semester anzeigen lassen, welche als Ansicht im System aufrufbar und als PDF exportierbar bzw. druckbar ist.

/F130/    **Bericht:** Semesterbelegung (Stundenplan)

**Akteur:** Studenten

**Beschreibung:** Studenten können sich eine Übersicht zu ihren Belegungen in einem Semester anzeigen lassen. Überschneidungen werden hierbei gesondert hervorgehoben. Die Übersicht (die einem Stundenplan gleich kommt) als Ansicht im System aufrufbar und als PDF exportierbar bzw. druckbar.

## 6 Daten

/D10/ **Datentyp:** Räume

**Attribute:** Raum ID (*systemintern*), Raumnummer, Gebäude, Stockwerk, Sitzplätze, PC-Plätze, Beamer, Visualizer, Overheads, Tafeln, Whiteboards

/D20/ **Datentyp:** Lehrstühle

**Attribute:** Lehrstuhl ID (*systemintern*), Lehrstuhlname, Lehrstuhlinhaber, Fakultät, (Haupt-)Gebäude, (Haupt-)Stockwerk

/D30/ **Datentyp:** Benutzer

**Attribute:** Benutzer ID (*systemintern*), Benutzerkennung, Passwort (Hash), Salt, E-Mail, Benutzerzugehörigkeit (Verwaltung, betreffender Lehrstuhl, wünschenswerterweise auch Student), Vorname, Nachname, letzter Login

/D50/ **Datentyp:** Lehrveranstaltungen

**Attribute:** Veranstaltungs ID (*systemintern*), Benutzer ID (Dozent), Veranstaltungskurzbezeichnung, Veranstaltungsname, Semester, Benötigte SWS, Art (Vorlesung|Übung|Tutorium), Freigabe durch Dozent, Beschreibung (Tag, Zeiteinheiten, etc. wird über "Raumbelegung (/D60/) ermittelt, wo für jede Zeiteinheit ein Eintrag erstellt wird und Veranstaltungen mehrere Einträge pro Semester buchen können.)

/D60/ **Datentyp:** Raumbelegungen (aller Freigabestatus-Arten)

**Attribute:** Belegungs ID (*systemintern*), Veranstaltungs ID (Lehrveranstaltung), Raum ID, Semester, Tag, Zeiteinheit, Freigabestatus (unbearbeitet|freigegeben|abgelehnt|gegenvorschlag), Freigabenachricht (Falls ein Vorschlag abgelehnt wurde und dies nun ein reservierter Vorschlag des Status "gegenvorschlagist), Kommentar

/DW70/ **Datentyp:** Studentenbelegungen

**Attribute:** Studenten-Belegungs ID (*systemintern*), Benutzer ID (Student), Belegungs ID (freigegebene Lehrveranstaltung)

/DW80/ **Datentyp:** Ticker-Nachricht

**Attribute:** Meldungs ID (*systemintern*), Meldungstext, Start-Datum, End-Datum, exklusiv für Lehrstuhl ID ('null' wenn für alle), exklusiv für Veranstaltungs ID ('null' wenn für alle), exklusiv für Raum ID ('null' wenn für alle)

In Structured-Entity-Relationship-Modell (SERM) wird der Zusammenhang der

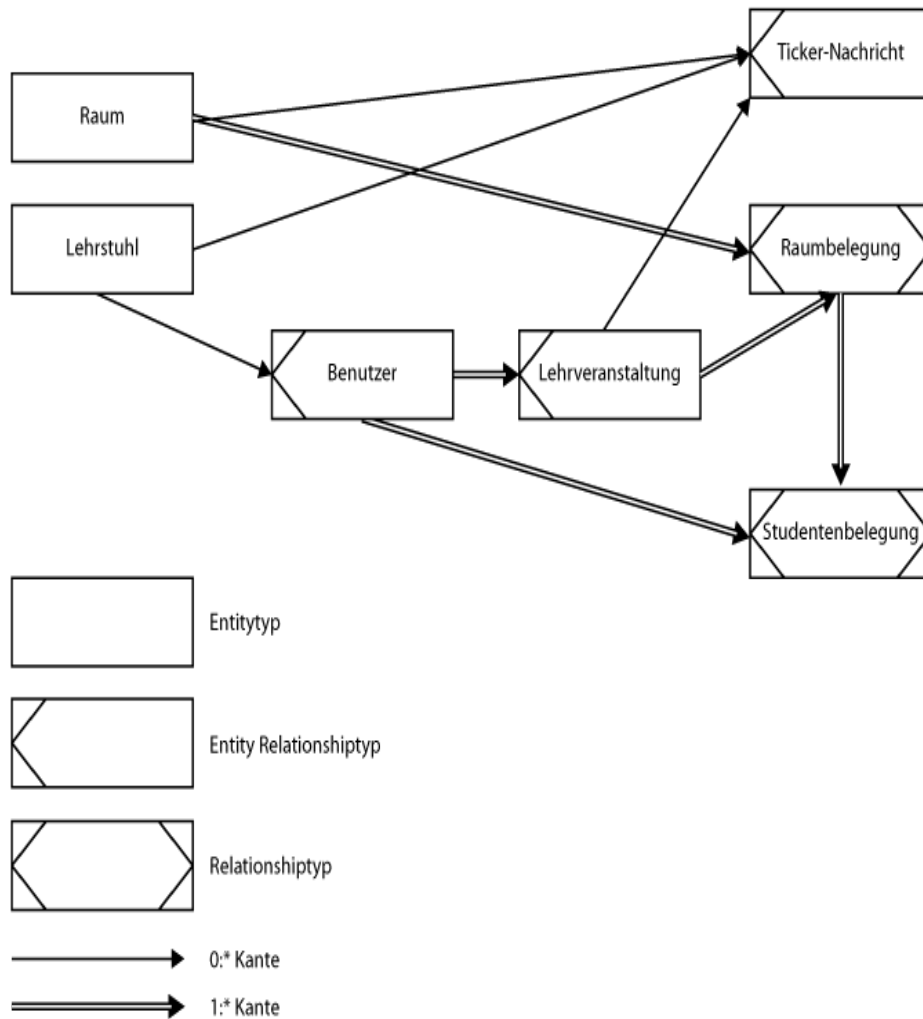


Abbildung 1: SERM der Daten-Architektur

oben spezifizierten persistent zu speichernden Daten verdeutlicht.

Wie zu sehen ist, existieren sowohl Räume als auch Lehrstühle als eigenständige Einheiten (Entity). Benutzer können einem Lehrstuhl zugeordnet werden, womit sie der Klasse "Dozententsprechen. Sie können aber auch der Klasse "Verwaltungs- oder SSStudenten" angehören und somit keinem Lehrstuhl zugeordnet werden. Lehrveranstaltungen müssen einem Benutzer zugeordnet werden, der (allein an der Grafik nicht erkennbar) ein der Klasse "Dozent" angehören muss. Eine Raumbelegung ist keine Einheit sondern eine Zuordnung (Relationship) und muss einer Lehrveranstaltung und einem Raum zugeordnet werden können. Die Zuordnung Studentenbelegung muss einem Benutzer (der Klasse "SSStudent") und einer Raumbelegung zugeordnet werden können. Ticker-Nachrichten können einem Raum, einer Lehrveranstaltung oder einem Lehrstuhl zugeordnet werden, nichts davon ist aber zwingend.



## 7 Leistungen

- /L10/ Alle Daten, die im vorherigen Abschnitt aufgelistet sind, müssen, sofern sie realisiert werden, persistent mittels einer SQL-Datenbank (erreichbar und verwaltet durch einen PostgreSQL-Server) gespeichert werden.
- /L20/ Bei allen Prozess mit Dateneingabe-Funktionen erhalten Nutzer aussagekräftige Fehlermeldungen, sollte ihr Prozess nicht ordnungsgemäß durchgeführt werden können.
- /L30/ Die Realisierung erfolgt als Java-Anwendung, so dass ein möglicher Betrieb auf allen gängigen PCs der Universität sichergestellt ist.
- /L40/ Die Programmarchitektur richtet sich nach dem MVC-Modell, wobei eine saubere untergliederung zwischen den Schichten gewährleistet wird, sowie ein hohes Maß an Modularität erreicht wird. Dieses erlaubt es auch verhältnismäßig einfach Erweiterungen zu einem späteren Zeitpunkt zu erstellen.
- Die erste Version der Anwendung soll aus folgenden Modulen bestehen:
- CoreGUI:** Hier wird die Start-Routine sowie der Startbildschirm der Anwendung realisiert. Die Funktion /F01/ und /F60/ werden von diesem Modul umgesetzt.
- Verwaltung:** Hier wird die Anzeige für Verwaltungsmitglieder umgesetzt, die die Funktion /F10/, wünschenswerterweise /FW21/, /F40/, /F50/, /F51/, /F80/ und /F90/ realisiert.
- Dozenten:** Hier wird die Anzeige für Dozenten umgesetzt, die die Funktion /F20/, wünschenswerterweise /FW21/, /F30/ und /F70/ realisiert.
- Stundenplan:** Hier wird die Anzeige für den Stundenplan von Studenten umgesetzt, die die Funktion /F130/ realisiert.
- Raumplan:** Hier wird die Anzeige für Mitglieder der Verwaltung (und ggf. andere) von Räumen und deren Belegung umgesetzt, die die Funktion /F120/ realisiert.
- Studentenprofil (wünschenswert):** Hier wird die Anzeige für Studenten umgesetzt, die die Funktion /FW61/ realisiert.
- /L50/ Es wird eine Nutzerauthentifizierung realisiert, die es gewährleistet, dass nur berechtigte Nutzer Änderungen durchführen. Um Missbrauch ausschließen zu können, sollte allerdings eine Middleware (die großteils die Model- und Controller-Schicht abdeckt) zwischen Datenbank und Client (der hauptsächlich die View-Schicht umsetzt) realisiert werden. In diesem Fall sollte die Kommunikation zwischen Client und Middleware durch SSL (mithilfe von javax.net.ssl.\* und javax.rmi.ssl.\*) abgesichert werden.

/L60/      Zeitliche Spezifikationen (Abgleich mit Christian)