

HdM Vorlesungswahl 2.0

Innovation Project 1 (EDV Nr. 38934)

Marius Heil (27509)
HdM Stuttgart
10.01.2014
mh218@hdm-stuttgart.de

Bisherige Situation

Es existiert momentan eine Vielzahl von Anlaufstellen für die Vorlesungswahl, die der Student verstreut über die HdM Webseite finden kann.

Die wichtigsten sind: Studieninhalte¹, das Vorlesungsverzeichnis², Vorlesung auswählen³, die Dozentenpläne⁴, der Persönliche Stundenplan⁵, den Studiumsverlauf⁶ und das Stundenplan Tool⁷.

Jede dieser Ansichten bietet Funktionalitäten die in anderen Ansichten wiederum fehlen. Selbst die Detailansichten sind oft unterschiedlich und bieten einen anderen Umfang an Informationen. Oft ist es daher notwendig eine Kombination aller dieser Seiten zu verwenden um den Vorlesungsplan für das nächste Semester zu erstellen. Eines der vielen Probleme ist beispielsweise, dass es nur aus bestimmten Ansichten heraus möglich ist eine Vorlesung in den persönlichen Stundenplan zu übernehmen. Eine Veranstaltung muss dann meist in einer weiteren Ansicht erneut gesucht werden um sie hinzuzufügen.

Die Suchoptionen sind sehr eingeschränkt und wurden im Laufe der Jahre immer um weitere Seiten erweitert um den Ansprüchen gerecht zu werden. Das zuletzt aufgeführte Stundenplan-Tool entstand als Semesterprojekt auf die Anfrage hin, dass es möglich sein sollte alle Vorlesungen anzeigen zu lassen die in einer Freistunde stattfinden. Dieses Tool wiederum ist auf genau diese Aufgabe beschränkt und kann nicht für andere Suchanfragen verwendet werden.

Dieses Dilemma wollte ich verbessern und einen Ansatz entwickeln, der eine sehr einfache Bedienung ermöglicht und dabei ein Vielfaches der bisherigen Funktionalität zur Verfügung stellt.

¹ <http://www.hdm-stuttgart.de/studenten/stundenplan/studieninhalte/studiengang>

² <http://www.hdm-stuttgart.de/studenten/stundenplan/vorlesungsverzeichnis/vorlesungsverzeichnis>

³ https://www.hdm-stuttgart.de/studenten/stundenplan/pers_stundenplan/stundenplanfunktionen/stundenplan_bearbeiten/SemesterAnzeigen?sgid=350013&Semester=1

⁴ <https://www.hdm-stuttgart.de/intranet/dozenten/dozentenplaene/stundenplanDozent?mitid=165&vorname=Fridtjof&nachname=Toenniessen&prof=1&titel=Dr.>

⁵ https://www.hdm-stuttgart.de/studenten/stundenplan/pers_stundenplan/stundenplanfunktionen/wochenansicht

⁶ https://www.hdm-stuttgart.de/studenten/stundenplan/pers_stundenplan/stundenplanfunktionen/stundenplan_bearbeiten/studiengang?alles=1&sgang=350013

⁷ https://www.hdm-stuttgart.de/studenten/stundenplan/pers_stundenplan/stundenplanfunktionen/freie_stunde

Grundlegende Gedanken

Um diese Probleme zu lösen, bedarf es also einer Seite, die all diese Funktionen in einer übersichtlichen Weise bereitstellt. Diese Lösung soll intuitiv zu benutzen sein und eine Vielzahl an unterschiedlichen Suchanfragen ermöglichen. Suchkriterien müssen kombinierbar sein und eine gezielte Eingrenzung der Vorlesungen ermöglichen.

Mögliche Suchanfragen sind beispielsweise:

- Alle Vorlesungen aus den Studiengängen CSM und EMM, die im nächsten Semester an einem Montag stattfinden.
- Alle Vorlesungen aus meinem eigenen Studiengang, die 6 ECTS oder mehr geben.
- Alle Vorlesungen die mein Lieblingsprofessor dieses Semester im Studiengang MI7 anbietet.

All dies sind Suchanfragen, die dem Usecase eines typischen Studenten zu Beginn eines Semesters entsprechen. Die Ergebnisse dieser Suchen sind bisher jedoch einzig und allein durch manuelles Durchsuchen mehrerer Seiten herauszufinden.

Grafischer Entwurf

Unter Einbeziehung aller Bedingungen entstand der grafische Entwurf, der in Abbildung 1 betrachtet werden kann. Der Entwurf besteht aus vier Teilen. Im obersten Teil der Webseite können Filter hinzugefügt und entfernt werden, die sich auf die Suchergebnisse auswirken. Direkt darunter kann der aktuelle Persönliche Stundenplan eingesehen werden. Unter diesem wiederum befindet sich eine Tabelle mit allen Vorlesungen die den Filterkriterien entsprechen. Die Detailansicht am unteren Ende der Seite rundet das Konzept ab.

Vorlesungsfilter
Zurücksetzen

Zeit

ECTS

Von: 2

Bis: 4

Professor

Goik +

Kiefer -

Gerlicher -

einschließen

Vorlesungsname

IT Systeme

+

Filter hinzufügen

Persönlicher Stundenplan
Kalender Export

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
08:15 - 09:45		Embedded Systems/W (137) -		Multimedia Codecs/W (136) -		
10:00 - 11:30	Object Recognition in Image and Video Data/W Medien/W (047) (136) -	Seminar Interaktive Adaptive User Interfaces/W (135) -		Multimedia Codecs/W (136) -	Seminar Interaktive Medien / W Raum: 136 -	
11:45 - 13:15	Object Recognition in Image and Video Data/W (136) -					
13:15 - 14:15						
14:15 - 15:45						
16:00 - 17:30			Management von IT- Projekten/P (116) -			
17:45 - 19:15		Interaktive Medien/W (013) -	Management von IT- Projekten/P (133) -			
19:30 - 21:00						

Edv Nr	Vorlesungsname	SWS	ECTS	Dozent	Raum	Prüfungsleistung	
320454	Grundlagen der Webseitengestaltung	2	4	Marius Heil	P1	MP	+
320001	Lustiges Flanieren durch die Umwelt	2	2	Ansgar Gerlicher	128	K	-
220454	Backe Backe Kuchen	2	6	Martin Goik	110	ST	+
123456	Erstellung eines Projektes	1	12	Peter Pan	011	PA	+

Zu dieser Veranstaltung sind Skripte verfügbar. [Skripte zu dieser Veranstaltung](#)

Dozent(en): Christiane Reinsch. [Dozenteninfo einblenden](#)

EDV-Nr: 38655

Studiengänge: Computer Science and Media (Master), Prüfungsleistung im Modul Wahlpflicht:
[Mobile Kommunikation in Semester 1 2 3](#)

Abbildung 1: Entwurf der neuen Vorlesungswahl

Architektur

Während der Planungsphase wurde schnell klar, dass ich die Anwendung als Single Page Anwendung gestalten wollte. Der wichtigste Vorteil ist hierbei die Geschwindigkeit mit der die Seite bedient werden kann. Des Weiteren erleichtert man dem Benutzer auf diese Weise eine komplizierte Navigation durch viele Fenster und Tabs. Auch der Implementierungsaufwand konnte durch diese Architektur geringer gehalten werden. Die Serverseitige Programmiersprache ist Python. Dies wurde durch die existierende ZOPE Installation vorgegeben. Da ZOPE nur sehr schlecht dokumentiert ist und es im Internet sehr wenige Foren zu der Thematik gibt, habe ich beschlossen, die Serverseitige Programmierung auf ein Minimum zu reduzieren.

Ein Großteil der Funktionalität wurde mit JavaScript / jQuery und AJAX Requests realisiert. Die Serverseitige Programmierung erfolgte hauptsächlich in der ZOPE eigenen DTML Templatesprache, da diese für meine Zwecke ausreichend war und schnellere Ergebnisse als Python versprach.

Es wurden zudem einige Teile aus jQuery UI verwendet. Hierzu gehören die von mir angepassten Autocomplete Textfelder, das Multiselect Popup welches zum Wählen der Tabellenspalten verwendet wird, sowie die Tooltips. Die Vorlesungstabelle wurde mit jqGrid realisiert, welches eine mächtige Bibliothek zur Darstellung von Tabellen jeder Art ist.

Es wurde versucht den Code so einfach und strukturiert wie möglich zu halten, damit auch eine langfristige Wartung der Seite möglich wird. Der Quellcode ist sehr umfangreich kommentiert worden.

Grundlegende Verzeichnisstruktur

Der Code in den folgenden Dateien beinhaltet einen Großteil der Implementierung. Dieser Code ist, sofern nicht explizit anders gekennzeichnet, von mir eigenständig programmiert worden und beinhaltet keine Arbeit anderer.

/www/css/vorlesungswahl.css: Beinhaltet einen Großteil der notwendigen Styles die zur Darstellung der Seite notwendig sind.

/www/js/jquery.mh.vwMultiSelect.js: Ein selbst implementiertes jQuery Plugin welches einen MultiSelect Filter realisiert.

/www/js/vorlesungswahl.js: Ein Großteil der selbst implementierten JavaScript Funktionalität.

/www/index.html: Das Grundgerüst der Seite, sowie einige Templates die dynamisch eingesetzt werden.

Der Serverseitige Code besteht aus Zope-Template Dateien und SQL Queries von denen ich viele selbst verfasst oder auf Basis von bereits vorhandenem Code erstellt bzw. angepasst habe. Diese befinden sich im ***/zope/*** Ordner. Die DTML Templates haben die Dateiendung .html während die SQL Abfragen mit .sql enden. Viele dieser Abfragen und Templates sind leider außerhalb der ZOPE Plattform nicht ausführbar.

Das komplexeste Query mit ca. 120 Zeilen liegt in der Datei ***/zope/lectures/sql_lectures_all.sql***. Das Query befragt eine Vielzahl von Tabellen und baut daraus eine Liste an Vorlesungen, die direkt auf den aktuell angemeldeten Studenten zugeschnitten sind. Dies war notwendig, da sich Modulnamen, Termine, Dozenten, Pflichtvorlesungen und einige anderen Parameter je nach Studiengang ändern können. Das Query braucht im Schnitt 5 Sekunden um vom HdM Webserver beantwortet zu werden, diese Zeit kann je nach Auslastung schwanken.

Testumgebung

Die einfachste Möglichkeit das Projekt zu testen ist, das Projekt direkt vom HdM Webserver unter dieser Weiterleitungs-URL abzurufen:

http://mariusheil.de/upload/hdm_vorlesungswahl.php

Den vollen Funktionsumfang hat die Seite mit Google Chrome in der neusten Version. Auch in anderen Browsern sollte der Funktionsumfang weitgehend gleich sein. Am ausführlichsten wurde die Seite jedoch mit Chrome getestet.

Beim Öffnen der Seite erwartet der HdM Webserver Benutzername und Passwort um den geschützten Bereich zu betreten. Falls man bereits bei der HdM Webseite angemeldet ist, benötigt man diese Angaben nicht.

Es wurde mir mitgeteilt, dass es auch mit den Login Daten von Professoren oder Mitarbeitern möglich ist einen Persönlichen Stundenplan zu erstellen. Falls es hierbei jedoch unerwartet zu Problemen kommen sollte, muss die Seite mit den Logindaten eines Studenten getestet werden.

Features und Highlights

Das Projekt hat inzwischen einen Stand erreicht bei dem es unpraktisch wäre, detailliert auf die Programmierung einzugehen. Ich will im Folgenden lediglich den Fokus auf einige interessante Bereiche im Code lenken und mich ansonsten auf eine Auflistung der implementierten Funktionen beschränken. Für ein detailliertes Verständnis der Funktionen will ich auf den gut kommentierten Quellcode verweisen. Die Angabe von Codezeilen bezieht sich hierbei immer auf die Datei `/js/vorlesungswahl.js` in welcher ein Großteil der Funktionalität implementiert ist.

Grundlegende Features



Alle Daten werden zu anfangs geladen und dann per Ajax aktualisiert, sodass ein effizientes Benutzen der Seite möglich ist. Die Ladezeit hängt hierbei hauptsächlich von der Zeit ab, die der HdM Webserver zum Beantworten der SQL Anfragen benötigt. Diese Zeit kann durch Cachen der Abfragen deutlich verringert werden. Beim Start der Anwendung sind bereits der eigene Studiengang und das aktuelle Semester ausgewählt, um dem Benutzer den Einstieg zu erleichtern.

Das Design der Seite ist an das aktuelle Design des HdM Webauftritts angelehnt. Die Seite skaliert sehr gut auf verschiedenen Bildschirmgrößen. Sie ist jedoch nicht für mobile Endgeräte ausgelegt, da durch die Vielzahl an Funktionen eine stark abgeänderte Bedienung notwendig wäre.

Die implementierten Filter erlauben es den Studenten eine Vielzahl von Kriterien festzulegen nach welchen sie dann ihre Vorlesungen auswählen können. Die Bedienung sollte hierbei so einfach wie möglich gehalten werden.



Filter sind generisch aufgebaut. Hierdurch können bei Bedarf problemlos weitere Filter hinzugefügt werden. Das User Interface eines Filters wird aus einem benutzerdefinierten HTML Template zur Laufzeit gebaut. Es muss hierzu lediglich ein HTML Script-Tag mit dem Namen des Filters versehen werden. *(Code ab Zeile 66)*

Die gesamte Datenbasis kommt vom Server. Dies beinhaltet selbst die Tage und Vorlesungszeiten. Falls in Zukunft Änderungen an der Datenbank vorgenommen werden, so passt sich die Seite dynamisch an. (Code ab Zeile 12)

```
"sprachen": [
  {
    "id": "1",
    "name": "Deutsch"
  },
  {
    "id": "2",
    "name": "Deutsch und Eng
```

```
1 BEGIN:VCALENDAR
2 VERSION:2.0
3 PRODID:-//HdM by Marius
4
5 BEGIN:VEVENT
6 UID:timetable-event_1_6_
7 LOCATION:Raum: 220, 231
```

Es wurde ein einfacher Generator für *.ics Dateien implementiert, welcher es ermöglicht einzelne Vorlesungen oder aber auch den kompletten Persönlichen Stundenplan zu exportieren, um diesen dann in Outlook oder andere Kalenderapplikationen zu importieren. (Code ab Zeile 1670)

Diese Funktion ist eventuell nicht für ältere Browser verfügbar, da sie einige Funktionen verwendet, die erst in sehr aktuellen Browsern verfügbar ist.

Die Spalten in der Vorlesungsübersicht können je nach Wunsch und Interesse ausgewählt werden. Per Default sind nur die wichtigsten Spalten sichtbar. Auf Wunsch können jedoch weitere Informationen in die Tabelle eingefügt werden.



Zeitgleich stattfindende Blöcke sind der Übersichtlichkeit halber voneinander mit einer gestrichelten Linie getrennt. Auf diese Weise kann der Student sofort eine Doppelbelegung eines Zeitblocks erkennen.

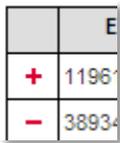
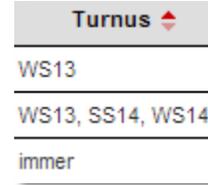
Die Spalten können zur besseren Übersicht sortiert werden, per Drag & Drop verschoben sowie vergrößert und verkleinert werden. Bei Größenänderung des Browserfensters passen sich die Inhalte dynamisch an. Die Sortierung berücksichtigt auch die unterschiedlichen Datentypen der einzelnen Spalten.

VS	Dozent	
	Toenniessen	Do: 11
	Toenniessen	Do: 11
	Toenniessen	Di: 16:



Konsequente Zeitblöcke werden textuell intelligent zusammengefasst um eine kompakte Darstellung in einer einzelnen Zelle zu ermöglichen. (Code ab Zeile 1766)

Der Turnus wird korrekt anhand einer Bitmaske aus der Datenbank geparkt und passt sich an das aktuell stattfindende Semester an. Diese Informationen sind über die HdM Webseite bisher nicht verfügbar und werden aktuell lediglich von diesem Tool unterstützt. (Code ab Zeile 1794)



Vorlesungen können mithilfe des Buttons am Zeilenanfang direkt aus der Vorlesungstabelle in den Stundenplan übernommen werden. Dies ist ebenfalls durch den entsprechenden Link in der Detailansicht möglich.



Die Detailansicht zeigt eine Vielzahl an Informationen und kann auch zur besseren Organisation in einem neuen Fenster geöffnet werden. Die Detailansicht kann entweder durch Auswahl in der Vorlesungsliste geöffnet werden, oder aber durch Auswahl einer Vorlesung aus dem Persönlichen Stundenplan.

Die Zeiten einer Vorlesung werden im Persönlichen Stundenplan hinterlegt sobald man über eine Vorlesung fährt. Hierdurch sieht man auf einen Blick, ob man zu diesem Zeitblock bereits eine andere Vorlesung belegt hat.



Eine Vielzahl an Tooltips vereinfacht die Benutzung der Seite und erklärt die Funktionsweise. Zur einfacheren Planung wird außerdem unterhalb des Persönlichen Stundenplans die aktuelle ECTS Summe ermittelt und angezeigt.

Filter

Der wichtigste Aspekt der neuen Vorlesungswahl sind die Filter. Und von diesen gibt es eine ganze Reihe. Eine Filterung ist somit möglich nach: Zeit, Dozent, Studiengang, Fachsemester, Wahl/Pflicht, Freitext, Turnus, ECTS und SWS. All diese Filter können beliebig kombiniert werden und ermöglichen so eine exakte Eingrenzung der Suche.



Durch einen Klick auf Filter hinzufügen erscheint ein Popup Menü, welches dem Benutzer erlaubt weitere Filter hinzuzufügen. Die Filter besitzen einen Titel und eine kurze Erklärung ihrer Funktionsweise.

Jeder Filter kann nur einmal hinzugefügt werden. Sobald alle Filter hinzugefügt wurden, verschwindet dieser Buttons.

Bei vielen Filtern ist sowohl eine Ein- wie auch Ausschließliche Filterung möglich. Auf diese Weise kann der Student entscheiden, ob er beispielsweise alle Vorlesungen eines Studienganges sucht oder bestimmte Studiengänge aus dem Suchergebnis entfernen möchte.

Im Titel des Filters wird die Anzahl der Vorlesungen angezeigt, die nach Anwendung des Filters noch gefunden werden.

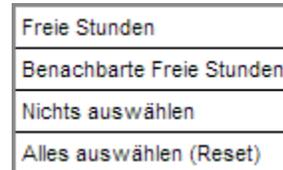


Je nach Inhalt werden Textfelder mit Autovervollständigung oder Listen genutzt, um die Filterkriterien darzustellen. Autovervollständigung eignet sich sehr gut, falls es viele Auswahlmöglichkeiten gibt.



Der Zeitfilter gehört zu den umfangreichsten Filtern. Man kann mit ihm exakt filtern, zu welcher Zeit die gesuchte Vorlesung stattfinden soll. Die einzelnen Zeiten können aktiviert oder deaktiviert werden. Zusätzlich kann man über die erste Spalte ganze Zeitblöcke ein- und ausschalten und über den Tabellenheader bestimmte Tage selektieren. Zur einfacheren Auswahl wird im Persönlichen Stundenplan angezeigt, über welchem Block sich der Mauszeiger momentan befindet.

Der Zeitfilter unterstützt ebenfalls eine Reihe von Voreinstellungen, die über den Schraubenschlüssel ausgewählt werden. Sehr praktisch sind hierbei die Einstellungen „Freie Stunden“ und „Benachbarte Freie Stunden“. Mit diesen ist es möglich Vorlesungen zu finden, die zeitlich nicht mit den bisher gewählten Vorlesungen kollidieren.



Die meisten Filter erlauben eine Mehrfachauswahl. Auf diese Weise können zum Beispiel die Vorlesungen einer Reihe von Professoren auf einmal gefunden werden.

Backend

Die Daten kommen gut strukturiert im JSON Format über eine REST Schnittstelle wo sie mithilfe von einigen sehr großen SQL Abfragen und der ZOPE eigenen DTML Template Engine erstellt werden. Die Gliederung wurde bewusst sehr einfach und universell gehalten. Auf diese Weise können die Daten auch von Folgeprojekten sehr gut weiterverwendet werden.

Es gibt insgesamt 4 URLs, die JSON Daten liefern:

- **Timetable:** Liefert den Persönlichen Stundenplan mit all den Vorlesungen des aktuell eingeloggten Studenten.
- **Lectures:** Gibt ein JSON zurück, welches alle Vorlesungen beinhaltet, die an der HdM angeboten werden. Zu jeder Vorlesung werden Dozenten, Räume, Prüfungsleistungen, ECTS, Termine, Studiengänge, etc. ausgegeben. Die darunterliegende SQL Abfrage erstreckt sich über 120 Zeilen. Die zurückgegebenen Informationen sind auf den Studiengang des eingeloggten Studenten zugeschnitten.
- **FilterInfo:** Das zurückgegebene JSON beinhaltet alle Informationen die für die Filterung und Darstellung der Daten notwendig sind. Hierzu gehört eine Liste der Dozenten inklusive deren IDs, eine Liste der Wochentage, Prüfungsleistungen, Studiengänge sowie die ID des aktuell laufenden Semesters.
- **PersonInfo:** Liefert Daten zur eingeloggten Person.

Außer diesen URLs gibt es noch einige REST Schnittstellen welche das Hinzufügen und Entfernen von Vorlesungen zum eigenen Stundenplan erlauben. Diese wurden jedoch von der aktuellen Webseite übernommen und nicht selbst implementiert. Die Vorlesungsdetails werden über ein angepasstes ZOPE Template direkt als HTML ausgeliefert und in einen Container geladen.

Schwierigkeiten und Lessons learned

- Die HdM Datenbankstruktur ist nirgendwo dokumentiert und musste durch analysieren vorhandener SQL Statements erst erarbeitet werden. Diese Struktur ist über die letzten 15 Jahre gewachsen und ist inzwischen altersbedingt sehr komplex. Es gibt zudem viele Bedingungen und Besonderheiten, die erst durch ein tiefes Verständnis der SPO herausgefunden werden können. Da ich das Frontend so weit wie möglich von dieser Komplexität abkapseln wollte, habe ich die Datenstruktur sehr vereinfacht und optimiert, bevor ich sie per JSON an das Frontend ausgeliefert habe.
- Da ich zu Beginn des Projektes noch nie mit jQuery oder ZOPE gearbeitet hatte, musste ich mich in diese Thematiken erst einarbeiten.
- Auf den HdM Servern ist eine Entwicklung aufgrund von technischen Einschränkungen und eingeschränkten Rechten nur sehr schwer möglich. Eine lokale Entwicklungsumgebung war jedoch auch nur bedingt möglich, da ich an das HdM Backend gebunden war. Durch das Deaktivieren des Cross-Site-Scripting Schutzes war es mir möglich sehr effizient lokal zu entwickeln.
- Die erste Version der SQL Abfrage, welche die Vorlesungen ausgibt, hat 40 Sekunden gedauert und musste stark optimiert werden. In Zusammenarbeit mit Herrn Schulz, dem Verantwortlichen für die Datenbankstruktur der HdM, gelang es uns nach mehrmaligen Treffen eine optimierte SQL Anfrage zu schreiben, welche die grob 1600 Vorlesungen in ca. 4-5 Sekunden ausgeben kann.

Ausblick

Die Vorlesungswahl soll allen HdM Studenten für die Planung des kommenden Semesters zur Verfügung stehen. Ich habe zur Integration in den HdM Webauftritt bereits Termine mit den Verantwortlichen Personen arrangiert.

Denkbar wäre auch eine nahtlose Integration in den Persönlichen Stundenplan. Hierfür müsste die Ladezeit der Seite jedoch noch deutlich reduziert werden und es müssten Caching Mechanismen eingebaut werden.

Es wurde auch bereits eine stark modifizierte Form der Seite vorgeschlagen, die dann den Stundenplanern zur Verfügung stehen könnte und deren Arbeit erheblich vereinfachen könnte.

