

Abstimmung Projektvision

Yannik Voelker

28. Januar 2015

Inhaltsverzeichnis

1	Abstimmung der Projektvision	2
1.1	Projektvision	2
1.2	Voraussetzungen	2
1.3	Designübersicht und Funktionalität	2
1.4	Arbeitspakete	6
1.4.1	Vorprojekt	6
1.4.2	Hauptprojekt	7
1.5	Qualitätssicherung	9
1.6	Glossar	9

1 Abstimmung der Projektvision

1.1 Projektvision

Ziel des Projektes die die Erstellung eines Kalenders für die Studierenden der Universität Leipzig an der Fakultät für Mathematik und Informatik. Dieser soll die Termine für individuell gewählte Module mit einzelnen Veranstaltungen - also Vorlesungen, Seminare und Übungen - enthalten.

Die Auswahl der Kurse soll per Online-Anwendung erfolgen. In dieser kann eine Vorauswahl für ein „Standard“-Semester¹ ausgewählt werden und/ oder konkrete Module hinzu- oder abgewählt werden. Weiterhin soll die Auswahl konkreter Übungsgruppen zu Modulen möglich sein.

Weiterhin soll den Nutzern die Möglichkeit eingeräumt werden, für jeden Eintrag ein vom Vorlesungsbeginn und -ende abweichendes Start- und Enddatum festzulegen.

Nach abgeschlossener Auswahl der Lehrveranstaltungen wird der erstellte Kalender als Vorschau auf der Website angezeigt und kann im iCal-Format heruntergeladen werden.

1.2 Voraussetzungen

Für den Betrieb des Programms wird ein aus dem Netzwerk erreichbarer Linux-Host mit lauffähiger Python3-Installation und mindestens einem offenen Port vorausgesetzt. Für die Nutzung wird lediglich ein Webbrowser benötigt. Da auf neue Webstandards verzichtet wird sind hier keine Einschränkungen zu erwarten.

1.3 Designübersicht und Funktionalität

Das Programm für die personalisierten Stundenpläne soll über ein Webinterface gesteuert werden. Die Eingabe erfolgt über zwei Dropdown-Listen. Die erste Liste zeigt alle Studiengänge und die Zweite listet die Anzahl der entsprechenden Semester für diesen Studiengang auf. Das heißt, dass die Liste der Semester von der Eingabe des Studiengangs

¹Mathematik oder Informatik; B.Sc. oder M.Sc.

abhängig ist und somit nur die maximale Anzahl an Semestern für diesen Studiengang anzeigt und nicht die generell maximal mögliche Anzahl aller Studiengänge.

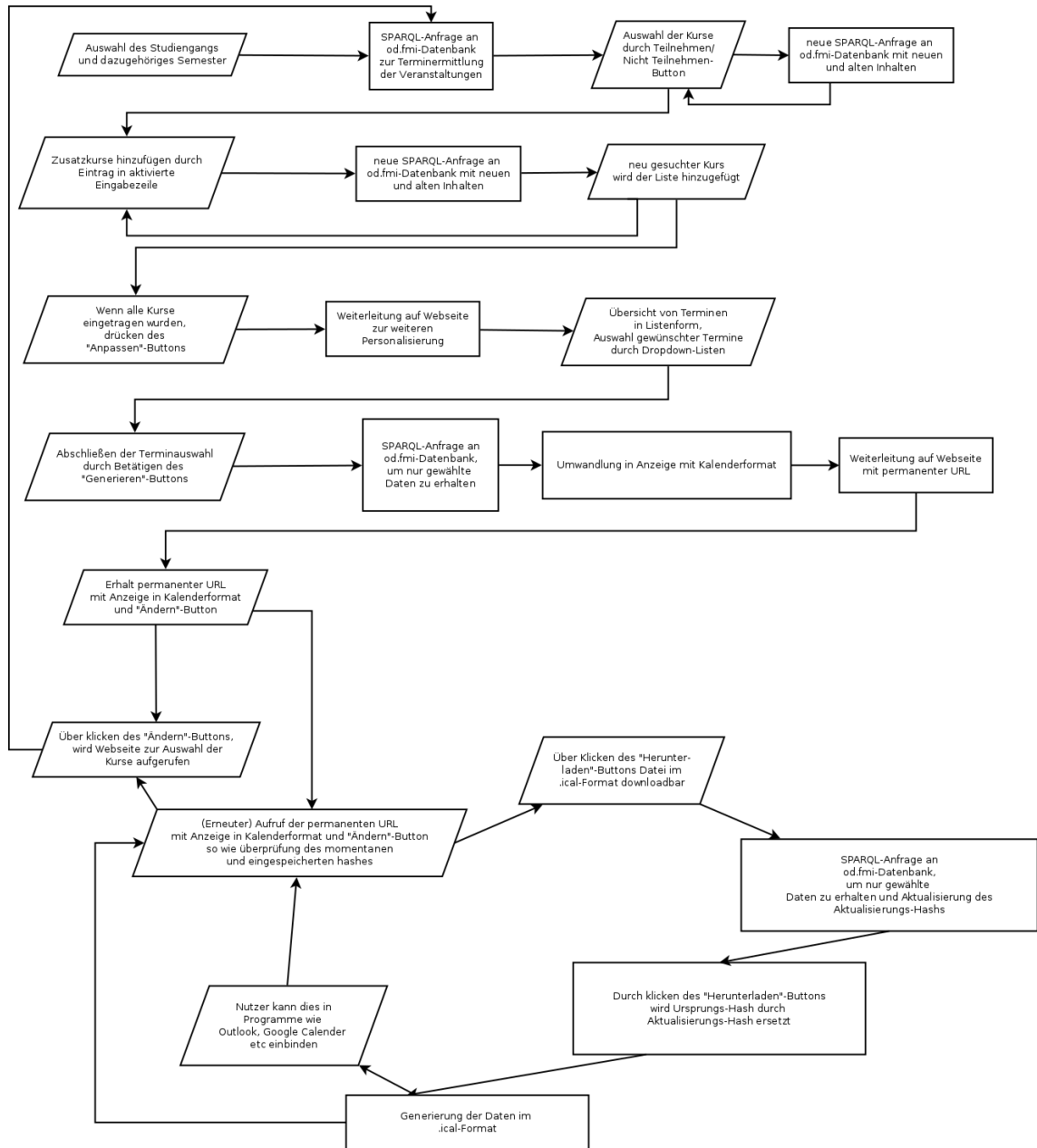
Sobald der Nutzer die Eingabe seines Studienganges und seines Semesters vornimmt, wird eine erste SPARQL-Anfrage an die od.fmi-Datenbank geschickt, um sämtliche Termine für Vorlesungen und Seminare des ausgewählten Studienganges und Semesters zu ermitteln. Mit dem Eintreffen der abgefragten Daten wird der Nutzer zunächst auf eine neue Website geführt, auf der sämtliche Kurse seines Semesters angezeigt werden. Der Nutzer hat hier die Möglichkeit nach seinem Belieben Kurse abzuwählen, indem er hinter den Kursnamen den Button „Nicht teilnehmen“ auswählt.

Zusätzlich dazu hat er über einen Radiobutton am Ende der Liste die Möglichkeit andere Kurse der Liste hinzuzufügen. Hierfür wird dann eine anfangs inaktive Eingabezeile aktiviert, in welcher der Nutzer den Namen des Kurses eingeben kann. Mit dieser Eingabe wird dann eine erneute Anfrage gestartet, welche sich aus der vorigen und der jetzt neu hinzugekommenen Eingabe zusammensetzt. Mit dem Eintreffen der abgefragten Daten wird die Seite aktualisiert und der neue Kurs am Ende der Liste hinzugefügt. Dieser Vorgang kann dann beliebig oft wiederholt werden. Sobald er sich für eine Kurszusammensetzung entschieden hat, wird er über einen „Anpassen“-Button auf die nächste Website geführt.

Hier werden die Termine aller Vorlesungen und Seminare der ausgewählten Kurse in einer Liste angezeigt. Jede Zeile dieser Liste setzt sich aus den Terminen für die Vorlesungen und denen für die dazugehörigen Seminare zusammen. Die einzelnen Termine werden hier auch als Dropdown-Liste angezeigt werden, um die Auswahl und Übersicht der Termine zu vereinfachen. Hier hat der Nutzer dann die Möglichkeit seine Termine nach seinen Wünschen zu legen und dabei auch andere Termine abzuwählen. Sobald er seine Entscheidungen getroffen hat, wird über den Button „Generieren“ eine SPARQL-Anfrage an die OD-Datenbank ausgelöst.

Nachdem die Anfrage bearbeitet wurde und ein Ergebnis vorliegt, wird aus den angefragten Daten eine iCalendar-Datei erstellt. Diese Kalenderdaten wird dann in einer neu erzeugten, permanenten URL in einer Vorschau als Kalender dargestellt, um dem Nutzer eine Übersicht über seine gewählten Kurse geben zu können. Die URL-Query-String sind zusätzlich auch noch alle ausgewählten Kurse enthalten, sodass auf der Serverseite nichts gespeichert werden muss. Zusätzlich dazu soll eine Eingabe über einen Button erfolgen, welcher die angezeigten Daten im Format .ical herunterlädt und auf dem Computer speichert. Diese Datei kann dann in die verschiedenen Kalenderprogramme wie Outlook, Thunderbird, iCalendar von Apple, Google Calendar usw. eingebunden werden.

Um die Möglichkeit zu realisieren den erstellten Kalender auf Aktualisierungen zu prüfen wird beim initialen Erstellen ein Hash der Daten zusammen mit der ID des Kalenders (entspricht dem Permalink) in einer Datenbank hinterlegt. Beim erneuten Aufruf des Links wird nach Abfrage an od.fmi erneut ein Hash erzeugt und mit dem Hinterlegten verglichen. Sobald der Nutzer die aktuelle Version als iCal herunterlädt wird der alte Hash durch den Neuen ersetzt.



1.4 Arbeitspakete

1.4.1 Vorprojekt

Im Vorprojekt soll eine Anwendung erstellt werden, bei welcher der Benutzer ein Studiengangsemester auswählt und eine visualisierte Stundenplantabelle angezeigt bekommt. Dies soll uns helfen, uns in die Methodik der SPARQL-Abfrage via eines Programmes einzuarbeiten. Desweiteren wollen wir dies visualisieren, was das Verständnis von RDF vertieft und uns hilft, damit umzugehen.

Paket 1:

Muss-Ziel:

- Erstellen der SPARQL-Abfrage für dieses Semester

Zuerst müssen wir die SPARQL-Abfragen programmieren, so dass beim Auswählen eines Semesters das richtige zurückgegeben wird.

Aufwand: 15%

Paket 2:

Muss-Ziel:

- Implementieren einer Visualisierung der Ergebnisse der Abfrage
- Einbau von allen geplanten Seiten als dummy

Kann-Ziel:

- Anzeige der Übungszeiten

Nachdem wir die Daten bekommen, müssen wir sie noch visuell darstellen. Dies wird in diesem Arbeitspaket realisiert.

Aufwand: 10%

1.4.2 Hauptprojekt

Paket 3:

Muss-Ziel:

- Erweitern der SPARQL-Abfrage auf eine flexible Auswahl

Diese Erweiterung bezieht sowohl das Auswählen konkreter Übungen sowie eine Flexibilität zwecks Vorlesungen, welche für ein anderes Semester empfohlen sind, mit ein. Nach dieser Erweiterung haben wir den Bereich der Informationsbeschaffung komplettiert und können uns vollständig auf den Bereich der Darstellung dieser Informationen konzentrieren. Aufwand: 25%

Paket 4:

Muss-Ziele:

- Implementieren des iCalendar-Formats
- Download des iCalenders
- Überarbeiten der Visualisierung durch einbinden des iCalendar-Formats

Wir wollen ein iCalendar-Format haben, da dieses mit allen Kalenderanwendungen kompatibel ist. Somit ist es wichtig, dieses auch zu implementieren. Nachdem es implementiert ist, können wir es nutzen, um unsere Visualisierung mit Hilfe dieses Format zu überarbeiten.

Aufwand: 15%

Paket 5:

Muss-Ziel:

- Nutzer-IDs hinzufügen
- Enddatum für Veranstaltungen anpassbar machen
- Personalisierten Kalender assoziiert mit Nutzer-ID speichern
- Personalisierten Kalender unter Angabe einer Nutzer-ID abrufen

Der Nutzer soll die Möglichkeit erhalten, bei Vorlesungen, die aus irgendeinem Grunde vor offiziellem Semesterende enden, das tatsächliche Ende einzutragen. Die Nutzer-ID wird eingeführt, um zum einen überlange Anfrage-Strings zu vermeiden, und zum anderen im nächsten Arbeitspaket den Kalender genau einem Nutzer zuordnen zu können.

Aufwand: 25%

Paket 6:

Muss-Ziele:

- Zuverlässige Hashmethode für Anfrage-Ergebnisse implementieren
- Hash des zuletzt heruntergeladenen Kalenders assoziiert mit Nutzer-ID speichern
- Nutzer bei geändertem Hash benachrichtigen und zu erneutem Download auffordern

Der Nutzer soll darauf hingewiesen werden wenn sich Daten im OD-Backend verändert haben die ihn betreffen, damit er sich einen aktualisierten Kallender herunterladen kann.

Aufwand: 10%

Paket 6: Für Nachfolgeprojekte

Kann-Ziele:

- Einbinden von almaweb
- Erweitern auf eine App

Das od.fini-Portal hat nur Veranstaltungen der Informatischen und Mathematischen Fakultät gespeichert. Somit wäre eine Erweiterung für alle Studenten wünschenswert. Dafür benötigt man einen Endpunkt im almaweb, von welchem man die Informationen beziehen kann. Aktuell ist dieser aber nicht vorhanden, weshalb wir dies nicht tun können.

Eine App wäre sicher wünschenswert für manche Leute, könnte man so browserfrei auf den Kalender zugreifen und die Informationen liegen bei dem Benutzer und nicht zentral. Da dies jedoch Kompatibilitätsprobleme mit z.B. Betriebssystemen bringen kann, haben wir uns für eine Webanwendung entschieden, welche von allen abrufbar ist, egal, welchen Browser oder welches Betriebssystem sie nutzen.

Aufwand: 50%

1.5 Qualitätssicherung

Siehe zeitgleich abgegebenes Dokument zur Qualitätssicherung.

1.6 Glossar

Dropdown-Liste Eine Dropdown-Liste ist ein grafisches Bedienelement, welches beim Aktivieren dem Nutzer mehrere Auswahlmöglichkeiten in einer Liste bietet

Flask ein auf Python basierendes Web-Framework

Hash eine Prüfsumme, die dazu dient, Datensätze auf Gleichheit zu prüfen

iCalendar ein Datenformat zum Austausch von Kalenderinhalten

Python Programmiersprache

Radiobutton Ein Radiobutton ist ein Bedienfeld einer grafischen Oberfläche, welche von einem Nutzer eine Auswahl verlangt

RDF Tripel von Daten, welche Beziehungen zwischen repräsentieren

SPARQL ist eine graph-basierte Abfragesprache für RDF

URL (auch Uniform Resource Locator) identifiziert und lokalisiert eine Website oder auch andere Ressourcen in einem Computernetzwerk