

Future Kids

Semesterarbeit - Werkzeug zur Unterstützung der Stundenplanerstellung

Marco Wettstein

2015-03-06

Inhaltsverzeichnis

1	Vorwort	4
2	Einleitung	5
2.1	Ausgangslage	5
2.2	Vorgehen	5
2.2.1	Anforderungsanalyse	5
2.2.2	Ist-Analyse	5
2.2.3	Konzept, Lösungsvarianten und Entscheid	6
2.2.4	Umsetzung des Prototyps	6
2.2.5	Testing und Abnahme	6
2.3	Planung	6
3	Anforderungsanalyse	7
3.1	Stakeholder-Analyse	7
3.1.1	AOZ	7
3.1.2	Pädagogischen Hochschule Thurgau (PHTG)	7
3.1.3	Administration der AOZ	7
3.1.4	Mentoren	7
3.1.5	Schüler	7
3.1.6	Lehrpersonen	7
3.2	Funktionale Anforderungen: User Stories	8
3.3	Nicht-Funktionale Anforderungen	11
3.3.1	NFR-001 - Definition of done	11
4	Ist-Analyse	13
4.1	Bestehende Lösung	13
4.1.1	Mentoren	13
4.1.2	Schüler	14
4.2	Probleme der bestehenden Lösung	16
4.2.1	Nicht-Funktionale Anforderungen (NFR)	16
4.3	Ergebnisse der Analyse	20
5	Konzept	21
5.1	Variante 1: Automatisierte Zuweisung	21
5.1.1	Das Stundenplanproblem	21
5.1.2	Verfügbare Lösungen zur Stundenplanerfestellung	21
5.1.3	Diskussion	21

5.2	Variante 2: Bestehende Lösung (manuelle Zuweisung) optimieren	22
5.2.1	Diskussion	22
5.3	Entscheidung	22
5.4	Feinkonzept: Bestehende Lösung (manuelle Zuweisung) optimieren	23
5.4.1	Anforderungen Sc-001 v-1 Sc-002 v-1, Sc-003 v-1, Sc-004 v-1	23
5.4.2	Anforderungen Sc-005 v-1, Sc-006 v-1, Sc-007 v-1, Sc-015 v-1	23
5.4.3	Anforderungen Sc-008 v-1, Sc-009 v-1	23
5.4.4	Anforderungen Sc-010 v-1	24
5.4.5	Anforderung Sc-011 v-1	24
5.4.6	Anforderung Sc-012 v-1	24
5.4.7	Anforderung Sc-013 v-1	25
5.4.8	Anforderung Sc-014 v-1	25
6	Design & Architektur	26
6.1	Erfassen der Zeitfenster eines Schülers / Mentors	26
6.2	Mentoren-Auswahl	26
6.2.1	Filter und Auswahl für den Vergleich	26
6.2.2	Layout	27
6.2.3	Vergleich der Zeitfenster als Kalenderansicht	28
6.2.4	Auswahl des Zeitpunkt des Treffens	28
6.2.5	Komponenten-Diagramm	30
6.3	Daten-Schemas	30
6.4	Technologie-Wahl	31
6.4.1	Client-Side-Komponenten	32
6.4.2	React	32
7	Umsetzung Prototyp	34
7.1	Verwendete Werkzeuge	34
7.1.1	Source-Code-Kontrolle	34
7.1.2	IDE	34
7.1.3	React	34
7.1.4	JSX und CJSX	34
7.2	Erste Iteration	35
7.3	Zweite Iteration	35
8	Testing	36
9	Fazit / Ausblick	37
A	Anhang	38
A.1	Quellcode	38
A.2	Quellenangaben	38

1 Vorwort

2 Einleitung

2.1 Ausgangslage

Future Kids ist ein gemeinnütziges Projekt, welches Primarschüler und -schülerinnen fördert, die bei der Bearbeitung ihrer schulischen Aufgaben zu Hause nur wenig Unterstützung erhalten. Future Kids stellt dabei jedem Kind einen Mentor oder eine Mentorin zur Seite, welches einmal pro Woche das Kind zu Hause unterstützt. In Freiwilligenarbeit wird eine Plattform für die Administrative Unterstützung von Future Kids entwickelt.

Die Plattform zum Projekt, welches Administratoren und Mentoren bei der Planung und Durchführung ihrer Aufgaben unterstützt wird unter anderem für das Planen und Zuweisen der Einsatzzeiten eingesetzt. Dies erfordert aber viel manuelle, aufwendige Arbeitsschritte.

In Zusammenarbeit mit der Administration, welche die Plattform nutzt, soll die Zuweisung der Mentoren zu den Schülern vereinfacht werden um den Administrativen Aufwand zu reduzieren. Dazu soll das bestehende Modul, welches für diese Planung zuständig ist, neu konzipiert werden.

2.2 Vorgehen

Die Arbeit wird in folgende Phasen unterteilt:

2.2.1 Anforderungsanalyse

In dieser Phase werden die Stakeholder und Rollen analysiert und die funktionalen, als auch die nicht-funktionalen Anforderungen von diesen Stakeholdern an die Arbeit eingeholt.

Die **funktionalen Anforderungen** werden in Anlehnung an SCRUM als User-Stories erfasst.¹

Zu jeder Story gehören:

(TODO:)

(TODO: add INVEST?)

(TODO: show story card template)

Nicht-Funktionale Anforderungen werden soweit wie möglich als "Definition of Done" erfasst, ebenfalls ein Mechanismus aus SCRUM. Hierbei werden vom Entwicklungsteam Kriterien erfasst, welche jede Story erfüllen müssen.

Ein Vorteil von *Definition of Done* ist, dass es eine Möglichkeit bietet, nicht-funktionale Rahmenbedingungen zu codieren, welche keinen direkten Business-Wert für das Produkt haben. Somit bleiben alle Stories immer "funktional".

2.2.2 Ist-Analyse

Die bestehende Anwendung, deren Schnittstellen und deren Daten werden in dieser Phase analysiert um zu überprüfen, wie sich das geplante Modul in die bestehende Lösung integrieren lässt.

(### Recherche)

(In der Recherchephase wird nach bereits existierenden, passenden Software-Lösungen, Algorithmen oder Darstellungen gesucht, um die die Anforderungen aus der Phase 2.2.1 zu erfüllen.)) TODO entfernen?

¹User Stories formulieren eine (in der Regel) funktionale Anforderung an ein Projekt aus der Sicht einer bestimmten Rolle und sind in einer Sprache erfasst, die sowohl der Auftraggeber als auch das Umsetzungsteam versteht. (et al.)

2.2.3 Konzept, Lösungsvarianten und Entscheid

In dieser Phase sollen mögliche Lösungsvarianten erarbeitet werden. Sofern es bereits passenden Lösungen auf dem Markt gibt, sollen diese gegebenenfalls gegen selbst zu erstellenden Varianten verglichen werden. Die optimale Lösungsvariante für die Umsetzung soll mittels eines geeigneten Verfahrens ermittelt werden.

2.2.4 Umsetzung des Prototyps

Die in der vorherigen Phase eruierte Lösung soll mit einer geeigneten Technologie umgesetzt werden und in die bestehende Plattform integriert werden

2.2.5 Testing und Abnahme

Die Lösung soll mit Hilfe automatisierter Tests und durch Abnahme durch die Anwender selbst verifiziert werden. Während die automatisierten Tests in der vorherigen Phase bereits umgesetzt werden, soll in dieser Phase ein Test- und Abnahme-Protokoll erstellt werden.

2.3 Planung

Der Zeitplan für die Umsetzung der einzelnen Phasen ist wie folgt:

Anforderungsanalyse Januar - Februar 2015

Ist-Analyse Anfang März 2015

Recherche Mitte März 2015

Konzept April 2015

Umsetzung Mai 2015

Testing und Abnahme Juni 2015

Letzer Abgabetermin 22.07.2015

3 Anforderungsanalyse

3.1 Stakeholder-Analyse

3.1.1 AOZ

Die AOZ ist eine Non-Profit-Organisation, "welche Asylsuchende und Flüchtlinge im Rahmen der Sozialhilfe und Unterbringung" unterstützt. Sie bietet Deutschkurse und weitere Hilfsmittel an um Personen in den Arbeitsmarkt und in die Gesellschaft zu integrieren. Sie ist eine selbständige Anstalt der Stadt Zürich.²

Die AOZ hat im Jahr 2010 das Projekt Future Kids ins Leben gerufen und ist Auftraggeberin der Arbeit.

3.1.2 Pädagogischen Hochschule Thurgau (PHTG)

Die Pädagogischen Hochschule Thurgau hat die Plattform lizenziert und wird sie Mitte des Jahres 2015 einführen. Die PHTG kann die geplanten Erweiterungen ebenfalls übernehmen, sie wird aber in der Anforderungsanalyse dieser Arbeit aus organisatorischen Gründen nicht berücksichtigt.

3.1.3 Administration der AOZ

Mitarbeiter der Administration der AOZ benutzen die Plattform für folgende Aktivitäten

- Erfassen und bearbeiten von Schülerprofilen
- Erfassen und bearbeiten von Mentoren
- Planung der Einsatzzeiten von Mentoren
- Zuweisung von Ersatzmentoren im Verhinderungsfall
- Kontrolle des Fortschrittes der Schüler (?) TODO

Frau Beren Tuna ist Mitarbeiterin von AOZ und administriert die Future Kids Plattform. Sie ist die primäre Ansprechperson, nimmt die User-Stories ab (siehe Kapitel 3.2) und vertritt die Administration der Future Kids Plattform.

3.1.4 Mentoren

Mentoren sind Studierende verschiedener Fachrichtungen und unterstützen die Schulkinder im "Future Kids"-Proramm zu Hause bei ihren schulischen Aufgaben.

Sie werden für ihren Einsatz finanziell oder insbesondere durch Erhalten von ECTS-Punkten³ vergütet. Mentoren nutzen die Plattform zum Erfassung von Protokollen, die sie nach jedem Hausbesuch erstellen.

3.1.5 Schüler

Schüler im "Future-Kids"-Projekt erhalten einmal pro Woche Unterstützung durch die Mentoren. Sie benutzen die Plattform nicht selbst, haben aber ein Profil auf der "Future Kids"-Plattform.

3.1.6 Lehrpersonen

Die Lehrpersonen der zu unterstützenden Schüler benutzen aktuell die Plattform nicht, in Zukunft sollen sie aber den Fortschritt der Schüler direkt auf der Plattform überprüfen können.

²Siehe Quellen ("AOZ Startseite"; "Über Die AOZ")

³European Credit Transfer System, siehe ("ECTS - Wikipedia")

3.2 Funktionale Anforderungen: User Stories

Um die funktionalen Anforderungen einzuholen wurde zusammen mit der Auftraggeberin User-Stories erarbeitet und priorisiert.

Diese Stories werden in der Umsetzungsphase anhand ihrer Prioritäten umgesetzt. Stories und Prioritäten wurden mit dem Kunden in einem Workshop ausgearbeitet. Die Werte für die Prioritäten sind *MUST*, *SHOULD* und *MAY* gemäss rfc2119.⁴

TODO: title description anders?

Sc-001 v-1	Als Administrator möchte ich die verfügbaren Zeitfenster eines Mentors erfassen
<i>Story</i>	Jeder Mentor gibt die Zeitfenster an, an welchen er Schüler betreuen kann. linebreak Ein Zeitfenster hat einen Wochentag, sowie eine Start- und Endzeit. linebreak Diese Zeitfenster sollen auf dem System erfasst werden können.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Erfassbare Zeitfenster: von 13:00 bis 19:30 <input type="checkbox"/> Es können beliebig viele Zeitfenster erfasst werden <input type="checkbox"/> Erfassbare Tage: Montag - Freitag <input type="checkbox"/> Erfasste Zeitfenster können sich nicht überlappen. <input type="checkbox"/> Auswahl kann persistiert werden.
<i>Priorität</i>	MUST

Sc-002 v-1	Als Administrator möchte ich die Zeitfenster eines Mentors sehen
<i>Story</i>	Es kann eingesehen werden, wann ein Mentor verfügbar ist.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Es werden alle erfassten Zeitfenster angezeigt
<i>Priorität</i>	MUST

Sc-003 v-1	Als Administrator möchte ich die Zeitfenster eines Schülers erfassen
<i>Story</i>	Jeder Schüler gibt an, an innerhalb welcher Zeitfenster er oder sie verfügbar ist. Ein Zeitfenster ist analog zur Story Sc-001 v-1 definiert. Ein Administrator soll diese Zeitfenster für einen Schüler speichern können.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Analog Story Sc-001 v-1
<i>Priorität</i>	MUST

Sc-004 v-1	Als Administrator möchte ich die Zeitfenster eines Schülers sehen
<i>Story</i>	Analog zu Sc-002 v-1 sollen die Zeitfenster eines Schülers eingesehen werden.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Analog Story Sc-002 v-1
<i>Priorität</i>	SHOULD

⁴<http://tools.ietf.org/search/rfc2119>

Sc-005 v-1	Als Administrator möchte ich einem Schüler einen primären Mentor zuweisen können
<i>Story</i>	Ein Administrator soll einen Mentor aus der Liste der verfügbaren Mentoren auswählen können und einem Schüler zuweisen können
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> User kann aus verschiedenen Mentoren auswählen <input type="checkbox"/> User kann einen davon selektieren <input type="checkbox"/> Der selektierte Mentor kann persistiert werden <input type="checkbox"/> Der selektierte Mentor erscheint auf dem Schüler Profil als zugewiesener Mentor
<i>Priorität</i>	MUST

Sc-006 v-1	Als Administrator möchte ich einem Schüler einen Ersatzmentor zuweisen können, welcher im Ersatzfall den Schüler betreut
<i>Story</i>	Im Ersatzfall wird der Mentor von einem anderen vertreten. Dieser Ersatzmentor kann einem Schüler zugewiesen werden
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> User kann aus verschiedenen Mentoren auswählen <input type="checkbox"/> User kann einen davon selektieren <input type="checkbox"/> Der selektierte Mentor kann persistiert werden <input type="checkbox"/> Der selektierte Mentor erscheint auf dem Schüler Profil als zugewiesener Ersatzmentor
<i>Priorität</i>	SHOULD

Sc-007 v-1	Als Administrator möchte ich bei einem Schüler vermerken, dass der Ersatzmentor aktiv ist
<i>Story</i>	Tritt der Ersatzfall ein, so soll bei dem Schüler vermerkt werden können, dass der Ersatzmentor aktiv ist.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> User kann den zugewiesenen Ersatzmentor als aktiv markieren <input type="checkbox"/> Auswahl wird persistiert
<i>Priorität</i>	MAY

Sc-008 v-1	Als Administrator möchte ich für einen Schüler nach Mentoren mit überschneidenden Zeitfenstern suchen können
<i>Story</i>	Für die Wahl eines Mentors kommen primär nur die in Frage, welche mit dem Schüler ein genügend grosses, sich überschneidendes Zeitfenster haben.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Es werden Mentoren gefunden, welche min 2h Überschneidung mit dem Schüler haben
<i>Priorität</i>	MAY

Sc-009 v-1	Als Administrator möchte ich für einen Schüler nach Mentoren mit nicht ganz überschneidenden Zeitfenstern suchen können
<i>Story</i>	Manchmal ist es sinnvoll, auch Mentoren zu finden, welche nur 1.5h Überschneidung mit dem Schüler haben, aber beispielsweise wegen anderen Kriterien für den Schüler passend wären. In diesem Falle würde der Administrator bei dem Mentor rückfragen.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Es werden Mentoren gefunden, welche min 1.5h Überschneidung mit dem Schüler haben
<i>Priorität</i>	MAY

Sc-010 v-1	Als Administrator möchte ich für einen Schüler nach Mentoren mit passendem Geschlecht suchen können.
<i>Story</i>	Für manche Schüler wird ein Mentor eines bestimmten Geschlechts bevorzugt. Es soll daher möglich sein, nach diesem Kriterium Mentoren zu finden.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Es können nach weiblichen oder männlichen Mentoren gesucht werden <input type="checkbox"/> Wenn kein Geschlecht ausgewählt ist, werden alle Mentoren gefunden.
<i>Priorität</i>	MAY

Sc-011 v-1	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche mittels ECTS-Punkten vergütet werden.
<i>Story</i>	Wie in 3.1.4 beschrieben, werden Mentoren durch Erhalt von ECTS-Punkten für ihren Einsatz vergütet. Dies ist ein Kriterium, dass bei der Auswahl eines Mentor eine Rolle spielen kann.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Wenn ECTS ausgewählt ist, werden nur Mentoren mit ECTS-Vergütung gefunden <input type="checkbox"/> Wenn Finanziell ausgewählt ist, werden nur Mentoren gefunden, die finanziell vergütet werden. <input type="checkbox"/> Wenn nichts davon ausgewählt ist, werden alle Mentoren gefunden.
<i>Priorität</i>	MAY

Sc-012 v-1	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche an einer bestimmten Schule eingesetzt werden.
<i>Story</i>	Der Einsatzort eines Mentors ist für die Wahl eines Mentoren ebenfalls von Bedeutung.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Ich kann nach Mentoren suchen, die an einer bestimmten Schule eingesetzt werden <input type="checkbox"/> Werden mehrere Einsatzorte angewählt, werden Mentoren aller Einsatzorte angezeigt <input type="checkbox"/> Wenn kein Einsatzort ausgewählt wird, werden alle Mentoren gefunden.
<i>Priorität</i>	SHOULD

Sc-013 v-1	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche eine bestimmte Anzahl Schüler betreuen
<i>Story</i>	Mentoren können kein, ein oder zwei Kinder betreuen (einmal als primärer Mentor, einmal als Ersatzmentor)
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Es können Mentoren gefunden werden, welche noch kein Kind betreuen <input type="checkbox"/> Es können Mentoren gefunden werden, welche nur ein Kind betreuen <input type="checkbox"/> Es können Mentoren gefunden werden, welche ein oder kein Kind betreuen <input type="checkbox"/> Es können Mentoren gefunden werden, welche bereits zwei Kinder betreuen
<i>Priorität</i>	MAY

Sc-014 v-1	Als Administrator möchte ich Mentoren nach Name suchen können.
<i>Story</i>	Die Administration möchte auch Mentoren nach ihrem Vor- und Nachnamen suchen können.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Bei Eingabe eines Namens sollen Mentoren mit passendem Vor- oder Nachnamen gefunden werden <input type="checkbox"/> Es sollen auch Mentoren gefunden werden, welche das Suchwort im Namen enthalten
<i>Priorität</i>	SHOULD

Sc-015 v-1	Als Administrator möchte ich den Zeitpunkt und den Wochentag des Treffens eines Mentors mit dem Schüler erfassen.
<i>Story</i>	Neben der Wahl eines Mentors (efsc-005) soll auch angegeben werden können, wann ein Mentor einen Schüler betreut.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Bei der Wahl eines Mentors kann ein Wochentag und ein Zeitpunkt ausgewählt werden <input type="checkbox"/> Wochentag und Zeitpunkt werden persistiert
<i>Priorität</i>	SHOULD

3.3 Nicht-Funktionale Anforderungen

Wie in Abschnitt 2.2.1 erläutert werden die Nicht-Funktionalen Anforderungen als "Definition of Done" erfasst.

Zu jeder Story müssen folgende Anforderungen erfüllt sein (Definition of done):

3.3.1 NFR-001 - Definition of done

Definition of Done	
<input type="checkbox"/> DD-0	Funktionalität wurde implementiert
<input type="checkbox"/> DD-1	Funktionalität wurde getestet (Akzeptanztests)
<input type="checkbox"/> DD-2	Funktionalität wurde auf eine Testumgebung geladen und die Integration überprüft
<input type="checkbox"/> DD-3	Umsetzung der Funktionalität ist eine Verbesserung gegenüber der bestehende Lösung und benötigt eine geringere Anzahl Klicks für die Benutzung
<input type="checkbox"/> DD-4	Funktionalität und Projekt kann unter einer Open Source Lizenz veröffentlicht werden und Umsetzung beachtet Lizenzen verwendeter Dritt-Software
<input type="checkbox"/> DD-5	Funktionalität wurde auf unterstützten Browsern überprüft (Firefox aktuell, Chrome aktuell, IE ab 9)
<input type="checkbox"/> DD-6	Gestaltung der Funktionalität entspricht der Corporate-ID von AOZ
<input type="checkbox"/> DD-7	Funktionalität und Deployment ist dokumentiert

4 Ist-Analyse


4.1 Bestehende Lösung

Die bestehende Lösung ist eine Webapplikation welche mittels dem Framework *Ruby on Rails*⁵ entwickelt wurde. Sie wurde von der Firma Panter AG und Freiwilligen entwickelt, der Quellcode ist auf Github verfügbar⁶ und unter der *GNU Affero General Public License*⁷ lizenziert.

Die Lösung kann angepasst und erweitert werden und es besteht die Möglichkeit, Änderungen auf einen Staging-Server zu stellen, welcher die AOZ zur Verifizierung nutzen kann.

4.1.1 Mentoren


Administrator/in
Schüler/in
Lehrperson
Mentor/in
SL/QUIMS-Verantwortliche/r
Erinnerung
Anleitungen & Dokumente
Schule



Mentor/in anzeigen

Aktionen
Bearbeiten
Zurück zur Liste
Stundenplan bearbeiten
Seitenanfang

Name Haller
Vorname Frederik
E-Mail mentor@example.com
Strasse, Nr. Keine Angabe
PLZ, Ort Keine Angabe
Geburtsdatum Keine Angabe
Telefon Keine Angabe
Hochschule Keine Angabe
Studienfach Keine Angabe
Ausbildungsstufe Keine Angabe
ÖV Keine Angabe
Einsatzort Keine Angabe
Einsatztag Keine Angabe
Kinder Keine Angabe
Kinder (Vertretung) Keine Angabe
Personalnummer Keine Angabe
Ects Nein
Startzeit Keine Angabe
Bekannte Keine Angabe
Absenzen / Verfügbarkeit
Bemerkungen Keine Angabe
Pendenzen Keine Angabe
Ersatzmentor Nein
Inaktiv gesetzt Nein



Zeiterfassung für den Monat

März
2015

31.03.2015	Coaching Pauschale	60 Minuten
------------	--------------------	------------

AOZ Zypressenstrasse 60 8040 Zürich Telefon: 044 415 65 91 futurekids@aoz.ch

Abbildung 1: Bearbeitung eines Mentors (Bestehende Lösung)

Administratoren können neue Mentoren erfassen und bestehende bearbeiten. Abbildung 1 zeigt die Profilseite eines Mentors. Es ist anzumerken, dass das Geschlecht eines Mentors nicht zugewiesen werden kann. Diese Funktion muss ergänzt werden um **Sc-010 v-1** zu erfüllen. Auch die Model-Klasse des Mentors oder der abgeleiteten `User`-Klasse enthält kein passendes Feld.


⁵<http://rubyonrails.org/>

⁶https://github.com/panterch/future_kids

⁷http://de.wikipedia.org/wiki/GNU_Affero_General_Public_License

Unter "Stundenplan bearbeiten" können die Zeitfenster bearbeitet werden, an denen ein Mentor verfügbar ist. Abbildung 2 zeigt, wie die Zeitfenster eingestellt werden können. Die Zeitfenster können in 30min Intervallen erfasst werden, von 13:00 bis 19:30, jeweils Montags bis Freitags. Es können beliebige Zeitfenster und -Kombinationen angeklickt werden. Dies erfüllt bereits **Sc-001 v-1** und **Sc-002 v-1** teilweise.

Listing 1 zeigt die Beziehungen einer Mentor-Entität. Zu sehen ist, dass im Datenmodell einem Mentor mehrere Schüler zugewiesen werden können.

Administrator/in Schüler/in Lehrperson Mentor/in SL/QUIMS-Verantwortliche/r Erinnerung Anleitungen & Dokumente Schule 

aoz

Aktionen
Stundenplandaten speichern

Stundenplan von Haller Frederik

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
18:30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
19:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Stundenplandaten speichern

Abbildung 2: Editieren der Zeitfenster eines Mentors (Bestehende Lösung)

```

1 class Mentor < User
2
3   has_many :kids
4   has_many :secondary_kids, :class_name => 'Kid',
5     :foreign_key => 'secondary_mentor_id'
6   has_many :journals
7   has_many :reminders
8   has_many :secondary_reminders, :class_name => 'Reminder',
9     :foreign_key => 'secondary_mentor_id'
10  has_many :schedules, :as => :person
11  belongs_to :primary_kids_school, :class_name => 'School'
12  belongs_to :primary_kids_admin, :class_name => 'Admin'


```

Listing 1: Mentor-Klasse in Ruby (mentor.rb)

4.1.2 Schüler

Schüler können ähnlich wie Mentoren bearbeitet werden. Abbildungen 3 und 4 zeigt die Profilseite eines Schülers. Im Gegensatz zu einem Mentoren kann ein Geschlecht zugewiesen werden.

Im unterem Teil des Bildschirms können die zugewiesenen Lehrer und Mentoren angesehen werden.

Administrator/in Schüler/in Lehrperson Mentor/in SL/QUIMS-Verantwortliche/r Erinnerung Anleitungen & Dokumente Schule 

aoz


Aktionen
 Stundenplandaten speichern

Stundenplan von Haller Frederik

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
18:30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
19:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Stundenplandaten speichern

Abbildung 3: Schüler Profil-Seite (Bestehende Lösung)

Administrator/in Schüler/in Lehrperson Mentor/in SL/QUIMS-Verantwortliche/r Erinnerung Anleitungen & Dokumente Schule 

aoz

Aktionen
 Stundenplandaten speichern

Stundenplan von Haller Frederik

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
18:30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
19:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Stundenplandaten speichern

Abbildung 4: Schüler Profil-Seite - Beziehungen (Bestehende Lösung)

Editiert man einen Schüler unter “Bearbeiten” können unter anderem Mentoren zugewiesen werden, sowie Wochentag und Zeitpunkt des Treffens des Mentoren mit dem Schüler. (Abbildung 5, Story **Sc-005 v-1**).

Treffen Wochentag

Treffen beginnt um

Lehrperson: Meckler Janine

Zweite Lehrperson

Mentor/in: Haller Frederik

Ersatz Mentor/in: Sabine Müller

☐ Stellvertretung aktiv

Coach

Abbildung 5: Zuweisen eines Mentoren zu einem Schüler (Bestehende Lösung)

Unter “Stundenplan bearbeiten” lassen sich analog zum Mentor die Zeitfenster eines Schülers einstellen (Abbildung 6). Dies erfüllt **Sc-003 v-1** und **Sc-004 v-1** teilweise. Dieser Bildschirm hat zudem eine weitere Funktion: Es können Mentoren ausgewählt werden, deren Zeitfenster mit denen des Schülers verglichen werden können. Eine direkte Wahl und Zuweisung eines Mentors aus diesem Bildschirm heraus ist nicht möglich.

4.2 Probleme der bestehenden Lösung

Bei der Analyse der bestehenden Lösung und im Workshop mit dem Auftraggeber sind weitere Probleme aufgetreten, welche zusätzliche, nicht-funktionale Anforderungen definieren.

4.2.1 Nicht-Funktionale Anforderungen (NFR)

Problemstellung

Die Komponente `\emph{Stundenplan}` eines Schülers erfüllt zwei Zwecke: Sie dient einerseits dazu, die Zeitfenster eines Schülers zu bearbeiten, andererseits dazu, die Zeitfenster von Schüler und Mentoren zu vergleichen um einen passenden Mentoren zu finden. Diese Doppelnutzung irritiert und kann zu Fehlmanipulation führen.

NFR-01	keine Doppelnutzung von Bedienelementen
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Jede Komponente hat genau eine primäre Aufgabe <input type="checkbox"/> Die Aufgabe einer Komponente ist klar ersichtlich
<i>Priorität</i>	normal

aoz

Aktionen
Auswahl bestätigen
Stundenplandaten speichern

Stundenplan

Mentorauswahl

Stundenplan von Hendrick Elaine

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18:30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Stundenplandaten speichern

Abbildung 6: Bearbeitung des Stundenplans eines Schülers (Bestehende Lösung)

aoz

Stundenplan

Mentorauswahl

Mentor Daten einblenden

Auswahl bestätigen

Auswahl aufheben

Mentor/innen ohne Schüler	Mentor/innen mit primärem Schüler zugewiesen	Mentor/innen mit sekundärem Schüler zugewiesen	Mentor/innen mit beiden Schülern zugewiesen	Ersatzmentor/innen
	Haller Frederik	Sabine Müller		

Abbildung 7: Hier können Mentoren ausgewählt werden, deren Zeitfenster mit einem Schüler verglichen werden (Bestehende Lösung)

Problemstellung

Wie auf Abbildung 6 zu sehen, befinden sich links vom Stundenplan die Schaltflächen. *Auswahl bestätigen* und *Stundenplandaten speichern*, letztere wird unterhalb des Stundenplan nochmals wiederholt und speichert die angewählten Zeitfenster (Checkbox). *Auswahl bestätigen* hat hier noch keine Funktion, die Bedeutung von *Auswahl bestätigen* zeigt sich erst, wenn man auf *Mentorauswahl* klickt (Abbildung 7) und dient dazu, hier die gewählten Mentoren für den Vergleich zu speichern; sie erscheinen danach im Stundenplan des Schülers.

NFR-02	Keine unnötigen oder nichtf-funktionale Bedienelemente
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Jedes Bedienelement auf jeder Komponente hat zu jeder Zeit eine Funktion
<i>Priorität</i>	normal

Problemstellung

Werden viele Mentoren für den Vergleich angewählt, erscheinen alle innerhalb des Stundenplans des Schülers, wie auf Abbildung 8 zu sehen. Der Stundenplan wird dadurch verzogen, was durch lange Namen noch verstärkt wird. Weiterhin ist nicht auf den ersten Blick zu sehen, welche der ausgewählten Mentoren überhaupt in Frage kommen, da nicht berücksichtigt wird, ob und wie lange die Zeitfenster überlappen.

NFR-03	Bedienung bleibt übersichtlich unabhängig von der Anzahl Mentoren
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Es soll möglich sein, die Mentorensuche mit beliebiger Anzahl Mentoren zu verwenden <input type="checkbox"/> Überlappungen sollen visuell gekennzeichnet sein <input type="checkbox"/> Es existieren Massnahmen zur Vorselektion von Mentoren zur Anzeige <input type="checkbox"/> Die Anzahl gleichzeitig dargestellter Mentoren soll sinnvoll limitiert werden
<i>Priorität</i>	normal

Problemstellung


Hat ein Administrator einen passenden Mentoren und einen passenden Termin gefunden, so können diese nicht direkt dem Schüler zugewiesen werden. Es muss zuerst auf das Schülerprofil gewechselt werden und wie in Abbildung \ref{screenshot_kid_bottom} zu sehen zugewiesen werden. Dies erfordert zu viele Klicks (TODO: anzahl) und es ist nötig, sich den Mentoren und den Termin zu merken.

NFR-04	Ausführen einer Aktion soll wenige Klicks benötigen
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Eine bestimmte Aktion soll weniger Klicks benötigen als die aktuelle Version
<i>Priorität</i>	normal

Problemstellung

Bei der Suche eines Mentors stehen dem Administrator nicht alle benötigten Informationen zur Verfügung. Die Liste der zur Auswahl stehenden Mentoren kann nicht nach bestimmten Kriterien wie Geschlecht oder Einsatzort gefiltert werden. Die Mentoren werden lediglich danach gruppiert, ob sie bereits einem oder zwei Schüler zugewiesen sind und ob sie als Ersatzmentor eingetragen sind. Dadurch ist es für einen Administrator nötig, zusätzliche Informationen zu einem Mentoren zu kennen oder auf der Profilseite des Mentors nachzuschlagen.

Hat ein Administrator einen passenden Mentor und Termin gefunden, so muss er diese Informationen sich merken, auf die Profilseite des Schülers zurückkehren und dort eintragen.



Aktionen

Auswahl bestätigen

Stundenplandaten speichern

Seitenanfang

Stundenplan von Hendrick Elaine

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
13:00	<input type="checkbox"/> Meier Franziska	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Sabine Müller	<input type="checkbox"/> Sabine Müller
13:30	<input type="checkbox"/> Meier Franziska	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Sabine Müller	<input type="checkbox"/> Sabine Müller
14:00	<input type="checkbox"/> Meier Franziska	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Sabine Müller	<input type="checkbox"/> Sabine Müller
14:30	<input type="checkbox"/> Meier Franziska	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Meier Franziska <input type="checkbox"/> Sabine Müller	<input type="checkbox"/> Sabine Müller
15:00	<input type="checkbox"/> Meier Franziska	<input type="checkbox"/>	<input type="checkbox"/> Meier Franziska Steiner Manuel	<input type="checkbox"/> Meier Franziska Sabine Müller	<input type="checkbox"/> Sabine Müller
15:30	<input type="checkbox"/> Meier Franziska	<input type="checkbox"/> Sabine Müller	<input type="checkbox"/> Julio Ricardo Montoya De La Rosa Ramírez Meier Franziska Steiner Manuel	<input type="checkbox"/> Haller Frederik Meier Franziska Sabine Müller	<input type="checkbox"/> Sabine Müller Steiner Manuel
16:00	<input checked="" type="checkbox"/> Haller Frederik Julio Ricardo Montoya De La Rosa Ramírez Meier Franziska	<input type="checkbox"/> Sabine Müller	<input type="checkbox"/> Julio Ricardo Montoya De La Rosa Ramírez Meier Franziska Steiner Manuel	<input type="checkbox"/> Haller Frederik Meier Franziska	<input checked="" type="checkbox"/> Steiner Manuel
16:30	<input checked="" type="checkbox"/> Haller Frederik Julio Ricardo Montoya De La Rosa Ramírez Meier Franziska	<input checked="" type="checkbox"/> Haller Frederik Julio Ricardo Montoya De La Rosa Ramírez Meier Franziska	<input checked="" type="checkbox"/> Julio Ricardo Montoya De La Rosa Ramírez Meier Franziska Sabine Müller	<input checked="" type="checkbox"/> Haller Frederik Julio Ricardo Montoya De La Rosa Ramírez Meier Franziska	<input checked="" type="checkbox"/> Julio Ricardo Montoya De La Rosa Ramírez Steiner Manuel

AOZ Zugesamtessen 80 8040 Zürich Telefon: 044 415 65 91 info@aoz.ch

Abbildung 8: Werden viele Mentoren für den Vergleich ausgewählt, leidet die Übersicht (Bestehende Lösung)

NFR-05	Ein Benutzer hat in einer Komponente alle Informationen für eine Entscheidung oder Aktion zur Verfügung
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Für eine Entscheidung oder Aktion in einer Komponente oder eines Bildschirms muss der Benutzer nicht zwischen Bildschirmen wechseln <input type="checkbox"/> Die dargestellten Informationen haben eine genügende Relevanz <input type="checkbox"/> Es werden keine irrelevanten Informationen dargestellt.
<i>Priorität</i>	normal

4.3 Ergebnisse der Analyse

TODO, evtl. doch weglassen

5 Konzept

Um die grundlegenden Anforderungen zu erfüllen kommen zwei prinzipielle Varianten in Frage:

5.1 Variante 1: Automatisierte Zuweisung

Ausgehend von den verfügbaren Zeitfenster von Mentoren und Schüler liesse sich die Zuweisung von Mentoren zu Schülern automatisieren. In dieser Variante soll ein geeignetes Verfahren zur Stundenplanautomatisierung gewählt werden, welches sich in die bestehende Lösung integrieren lässt. Dabei sollen sowohl fertige Lösungen (kommerziell oder nicht-kommerziell), als auch geeignete Frameworks, Bibliotheken oder Ansätze betrachtet werden.

5.1.1 Das Stundenplanproblem

Prinzipiell handelt es sich beim Problem der Stundenplanerstellung um ein NP-Vollständiges Problem,⁸ lässt sich also bei grossen Eingabegrößen (z.b. viele Schüler und Lehrer) nicht effizient lösen. Da alle NP-Vollständigen Probleme "gleich schwer" im Sinne der Komplexitätstheorie sind, ist es also gleich schwer wie das "Problem des Handlungsreisenden" oder das "Rucksackproblem". Entsprechend kommen zur Lösung (oder vielfach auch nur Optimierung) des Problems ähnliche Verfahren zur Anwendung.

Unter anderem können genetische Algorithmen für die Stundenplanerstellung verwendet werden.⁹ Bei der im Jahre 2007 durchgeführten "International Timetabling Competition 2007" gewannen jedoch Verfahren in der Art der "Lokalen Suche."¹⁰

In der Praxis müssen bei der Stundenplanerstellung viele Parameter und Randbedingungen beachtet werden, welche das Problem komplex machen können. Im vorliegenden Fall sind beispielsweise neben den Zeitfenstern von Schüler und Mentor auch die Einsatzorte der Mentoren zu berücksichtigen.

5.1.2 Verfügbare Lösungen zur Stundenplanerstellung

OptoPlanner ist ein Werkzeug für Constraint-Satisfaction-Probleme (CSP, Bedingungserfüllungsprobleme) (Siehe ("JBoss Optoplanner")) und kann prinzipiell auch für die Erstellung von Stundenplänen genutzt werden.

UniTime ist ein Quelloffenes System für die Erstellung von Stundenplänen insbesondere für Universitäten und war unter den Finalisten des "International Timetabling Competition 2007."¹¹

FET - Free Timetabling Software ist ein weiterer Quelloffenes Programm zur Stundenplanerstellung von Schulen und Universitäten¹²

Es existieren zahlreiche weitere Verfahren und Lösungen (sowohl kommerziell als auch nicht-kommerziell).

5.1.3 Diskussion

Diese Variante löst die Bedienprobleme aus 4.2, indem keine oder nur wenige Bedienelemente notwendig gemacht werden, um Mentoren den Schülern zuzuweisen.

Ebenfalls erfüllt sie wichtige Stories aus 3.2, macht Stories 9-14 allerdings überflüssig, da diese der Auswahl eines Mentors helfen.

⁸Vgl. (, and, Abschnitt 3.2.2). Je nach Rahmenbedingungen können Stundenplanprobleme auch in P liegen, vgl. ("The Complexity of Timetable Construction Problems," Abschnitt 3)

⁹Siehe (, and, Abschnitt 4)

¹⁰Vgl. ("International Timetabling Competition 2007")

¹¹Vgl. ("Unitime.org Constraint Solver Library (CPSolver 1.3)") und ("International Timetabling Competition 2007")

¹²Vgl. ("FET - Free Timetabling Software").

Vorteile:

- Kein oder nur minimales User-interface benötigt
- Dadurch gute Bedienbarkeit

Nachteile:

- Der Algorithmus muss sämtliche Randbedingungen berücksichtigen, da manuelle Anpassungen nur umständlich gemacht werden können
- Algorithmus muss entsprechend angepasst werden
- Integration eines Fremdsystem ist aufwendig (insb. falls fertige Lösung) und kann das Setup der Lösung erschweren.

5.2 Variante 2: Bestehende Lösung (manuelle Zuweisung) optimieren

Die bestehende Lösung erfüllt bereits die Stories 1-8 aus Abschnitt 3.2, jedoch ist die Bedienbarkeit durch die erwähnten Probleme in Kapitel 4.2 eingeschränkt. Diese können jedoch mit geeigneten Massnahmen gelöst werden. Insbesondere dienen Stories 9-14 als wichtige Hilfestellungen bei der Wahl eines Mentores, welche im Moment noch Fehlen.

In dieser Variante soll erarbeitet werden, mit welchen Konzepten die bestehende Lösung um sämtliche Stories aus 3.2 ergänzt werden kann und welche Massnahmen notwendig sind um die Probleme in 4.2 zu lösen.

5.2.1 Diskussion

Die bestehende Lösung erfüllt teilweise bereits die Anforderungen, Probleme liegen bei der Bedienbarkeit, der Übersicht und der Informationsdarstellung. Werden diese Probleme gezielt angegangen, kann die bestehende Lösung schrittweise verbessert werden.

Vorteile:

- Konzept der bestehenden Lösung wird beibehalten
- Genaue Anpassung an Kundenwunsch möglich
- Kein Fremdsystem notwendig
- Automatisierung zusätzlich immer noch möglich

Nachteile:

- Aufwendige Anpassungen der Bedienelemente notwendig

5.3 Entscheidung

Beide Varianten erfüllen alle Stories und können alle Probleme von Abschnitt 4.2 erfüllen. Nach Rücksprache mit dem Auftraggeber wurde Variante 2 gewählt.

Häufig spielen bei der Wahl eines Mentors für einen bestimmten Schüler weitere Umstände eine Rolle, die die Administratoren von AOZ berücksichtigen müssen. Diese Kriterien können persönliche Einschätzungen sein oder andere Kriterien, welche nicht auf der Plattform abgebildet werden. Manchmal werden auch Rückfragen mit Mentoren getätigt, wenn beispielsweise Zeitfenster nicht genau passen oder ein Ersatz gesucht werden muss. Daher wird vom Auftraggeber eine manuelle Zuweisung bevorzugt.

Die bestehende Lösung erfüllt teilweise bereits manche Anforderungen und soll daher gemäss Variante 2 optimiert werden. Die bestehende Lösung kann zukünftig immer noch um einen (teil-)automatisierten Prozess ergänzt werden.

5.4 Feinkonzept: Bestehende Lösung (manuelle Zuweisung) optimieren

5.4.1 Anforderungen Sc-001 v-1 Sc-002 v-1, Sc-003 v-1, Sc-004 v-1

Diese Stories umfassen das Erfassen und Anzeigen der Zeitfenster von Mentoren und Schüler. Sie werden bereits in der aktuellen Lösung erfüllt, allerdings existiert die erwähnte Doppel-Nutzung des Stundenplans wie in NFR-01: *Doppelnutzung von Bedienelementen* geschildert.

Massnahmen:

NFR-01: *Doppelnutzung von Bedienelementen*,

Bearbeitung der Zeitfenster eines Schülers soll von der Einsatzplanung des Mentors getrennt werden (visuell und funktional). Eine Möglichkeit wäre, das Bearbeiten der Zeitfenster des Schülers erst nach Aktivieren einer Schaltfläche (*Schüler-Zeitfenster bearbeiten*) zu ermöglichen. Damit ist klarer, welche Aktion gerade ausgeführt wird.

Das Setzen der Zeitfenster mit den Checkboxes kann beibehalten werden.

NFR-02: *Irritierende Bedienelemente*

- Doppelte und inaktive Schaltflächen entfernen
- Bedienelemente klar beschriften.

5.4.2 Anforderungen Sc-005 v-1, Sc-006 v-1, Sc-007 v-1, Sc-015 v-1

Die bisherige Lösung sieht für die Zuweisung eines Mentors ein Feld in der Editier-Maske eines Schülers vor, ebenso kann dort vermerkt werden, ob wer als Ersatzmentor vermerkt ist und ob dieser aktiv ist. Weiterhin kann ein Wochentag und Zeitpunkt des Treffens erfasst werden gemäss Anforderung Sc-015 v-1.

Massnahmen:

Diese Bedienelemente können weitgehend beibehalten werden.

Bei der Zuweisung von Mentor und Termin muss dieser aber bereits bekannt sein, d.h. ein Administrator muss bereits einen passenden Mentor und Termin gefunden haben, doch der Prozess des Findens eines Mentors ist in der bisherigen Lösung noch getrennt von der Zuweisung des Mentors; der Bildschirm, welcher die Mentorensuche ermöglicht, erlaubt es nicht, einen Mentor auch zuzuweisen. Dies widerspiegelt sich in den folgenden Anforderungen:

NFR-04: *Ausführen einer Aktion soll wenige Klicks benötigen*, sowie NFR-05: *Ein Benutzer hat in einer Komponente alle Informationen für eine Entscheidung oder Aktion zur Verfügung*

Es soll direkt aus der Komponente, welche das Finden eines Mentors erlaubt, heraus ein Mentor, sowie Termin ausgewählt und dem Schüler zugewiesen werden können. Dadurch muss sich ein Administrator nicht Mentor und Treffpunkt merken.

5.4.3 Anforderungen Sc-008 v-1, Sc-009 v-1

Diese Stories sollen einem Administrator ermöglichen, Mentoren zu finden, welche mit dem Schüler überschneidende Zeitfenster haben. Gemäss Gespräch mit dem Auftraggeber kann es aber auch sein, dass bei knappen Überschneidungen Rücksprache mit dem Mentor genommen werden kann.

Massnahmen

Diese Funktion ist in der aktuellen Lösung implementiert, allerdings leidet die Übersicht wenn viele Mentoren für den Vergleich ausgewählt werden, wie in Abbildung 8 zu sehen. Dazu wurde die folgende Anforderung erstellt:

NFR-03: *Bedienung bleibt übersichtlich unabhängig von der Anzahl Mentoren*

- Mentorennamen kürzen und kleiner darstellen
- Mit Farben oder Symbolen arbeiten pro Mentor
- Mentoren oder Zeitfenster, welche nicht in Frage kommen, ausblenden oder anders darstellen.
- Maximale Anzahl Mentoren limitieren

Weiterhin stehen neben den Zeitfenstern des Kindes und des Mentors keine weiteren Informationen zur Verfügung, es können lediglich Mentoren danach vorselektiert werden, ob sie bereits ein Kind betreuen oder nicht. Dazu ist aber ein Wechsel des Bildschirms nötig.

NFR-05: *Ein Benutzer hat in einer Komponente alle Informationen für eine Entscheidung oder Aktion zur Verfügung*

Gemäss den Stories **Sc-010 v-1**, **Sc-011 v-1**, **Sc-012 v-1**, **Sc-013 v-1** sollen Mentoren nach bestimmten Kriterien, wie Geschlecht, Name und Einsatzort ausgewählt werden können. Diese Kriterien können als Filter implementiert werden, welche passende Mentoren einblendet oder unpassende ausblendet.

Diese Filter sollen ohne Wechsel des Bildschirms benutzbar sein und jederzeit klar ersichtlich machen, welche Mentoren sichtbar sind.

Weiterhin kann es Sinn machen, Mentoren mit nicht passenden Zeitfenstern nicht gänzlich herauszufiltern, sondern auf grafische Weise zu visualisieren, ob und wie lange sich die Zeitfenster überschneiden. Es kann auch möglich sein, dass bei einem Mentoren noch keine Zeitfenster erfasst wurden. Eine Grafische Lösung macht hier daher Sinn und erfüllt weitgehend **NFR-05**, dass ein Administrator jederzeit alle nötigen Informationen zur Hand hat.

Die Kalenderartige Darstellung kann dabei beibehalten werden.

5.4.4 Anforderungen **Sc-010 v-1**

Das Geschlecht eines Mentors kann in der aktuellen Lösung nicht erfasst werden, das Geschlecht eines Schülers hingegen schon. D.h. die grundsätzlichen Mechanismen dazu existieren bereits.

Massnahmen:

- Es soll eine Datenmigration durchgeführt werden, welche allen Mentoren ein Feld "Geschlecht" hinzufügen.
- Es sollen Eingabemasken von Mentoren ergänzt werden, damit diese Information abgelegt werden kann
- Die Profilseite eines Mentors soll diese Information anzeigen
- Bei der Mentorensuche für einen Schüler sollen Mentoren nach dem Geschlecht gesucht oder gefiltert werden können
- Das Verhalten bei nicht zugewiesenem Geschlecht muss definiert werden.

5.4.5 Anforderung **Sc-011 v-1**

Das Feld, ob ein Mentor mit ECTS-Punkten vergütet wird oder nicht, ist bereits vorhanden

Massnahmen:

- Bei der Mentorensuche für einen Schüler sollen Mentoren nach diesem Kriterium gesucht oder gefiltert werden können.

5.4.6 Anforderung **Sc-012 v-1**

Der Einsatzort (d.h. eine Schule) eines Mentors nach Betrachtung des Datenmodells nicht direkt einem Mentor zugewiesen, sondern entspricht der Schule des Kindes, dem der Mentor zugewiesen ist.

Dadurch ist diese Information nur für Mentoren verfügbar, die aktuell ein Kind betreuen.

Massnahmen:

- Bei der Mentorensuche für einen Schüler sollen Mentoren nach diesem Einsatzort gesucht oder gefiltert werden können.

5.4.7 Anforderung Sc-013 v-1

Diese Funktion ist, wie oben erwähnt in der aktuellen Version bereits implementiert. (Siehe Abbildung 7). Allerdings muss dazu die Ansicht gewechselt werden.

Massnahmen:

Diese Funktionalität soll gleichartig wie die Suche nach passendem Geschlecht, Einsatzort oder anderen Kriterien umgesetzt werden.

5.4.8 Anforderung Sc-014 v-1

Die Suche nach Mentoren per Name für die Auswahl kann ebenfalls als Filter implementiert werden.

6 Design & Architektur

6.1 Erfassen der Zeitfenster eines Schülers / Mentors

Die bisherige Lösung verfügt bereits über eine Möglichkeit, die Zeitfenster eines Schülers und Mentors einzugeben, welche die Anforderungen des Kunden erfüllen. Allerdings wird beim Schüler dieser Bildschirm noch für die Auswahl der Mentoren zusätzlich genutzt (siehe Problem ??). Wie in Abbildung 9 gezeigt, werden daher die Funktionalitäten zur Mentorenauswahl von diesem Bildschirm entfernt. Ebenso werden unnütze Bedienelemente wie in ?? beanstandet entfernt.

Stundenplan von Hendrick Elaine

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17:30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18:30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Stundenplandaten speichern

Abbildung 9: Erfassen der Zeitfenster - bei der bisherigen Lösung können die markierten Bedienelemente entfernt werden

6.2 Mentoren-Auswahl

Für die Auswahl der Mentoren wird ein neuer Bildschirm erstellt. Er dient dazu, aus allen im System erfassten Mentoren zu filtern und die Zeitfenster der ausgewählten Mentoren mit denen des Schülers zu vergleichen. Dazu stehen diverse Filtermöglichkeiten zur Verfügung. Weiterhin soll es möglich sein, ein Treffen zwischen Mentor und Schüler zu persistieren (Wochentag und Zeitpunkt), gemäss **Sc-015 v-1**. Diese direkte Zuweisung reduziert die in ?? beanstandeten Anzahl Klicks.

6.2.1 Filter und Auswahl für den Vergleich

Die Auswahl der Mentoren geschieht in zwei Schritten:

1. Filter definieren die zur Auswahl stehenden Mentoren, diese sollen initial auf sinnvolle Werte gesetzt sein.
2. Aus den gefilterten Mentoren können bestimmte Mentoren für den Vergleich mit dem Schüler ausgewählt werden.

Durch dieses Zwei-Schritte Verfahren soll sichergestellt werden, dass die Übersicht sowohl bei vielen möglichen Mentoren, als auch im Fall, dass nur wenige passende Mentoren zur Verfügung stehen, nicht eingeschränkt ist.

(TODO: mock)

Die einzelnen Filter reduzieren die Mentoren für die Suche nach Kriterien, die jeweils mit UND verknüpft werden, d.h. Geschlecht=m und ECTS=true wird alle männlichen Mentoren anzeigen, welche ECTS Punkte bekommen:

6.2.1.1 Geschlecht Hat drei Zustände: nichts ausgewählt, m oder f.

Ist nichts ausgewählt, so wird nicht nach Geschlecht gefiltert

6.2.1.2 ECTS Hat drei Zustände: nichts ausgewählt, true oder false:

- ECTS=true: alle Mentoren, die ECTS-Punkte bekommen
- ECTS=false: alle Mentoren, die nicht ECTS-Punkte bekommen
- nichts ausgewählt: es wird nicht nach dem Kriterium gefiltert.

6.2.1.3 Einsatzort Zeigt alle Schulen an. Wird eine Schule ausgewählt, werden nur Mentoren angezeigt, die an dieser Schule aktiv sind.

6.2.1.4 Anzahl betreute Kinde Hat folgende Werte:

- 0-1: Zeigt alle Mentoren, die kein oder ein Kind betreuen
- 0: Zeigt alle Mentoren, die kein Kind betreuen.
- 1: Zeigt Mentoren, die genau ein Kind betreuen
- 2: Zeigt Mentoren, die bereits zwei Kinder betreuen
- unselektiert: Das Kriterium ist nicht aktiv.

6.2.1.5 Namensfilter Der Namensfilter wählt die Mentoren per Namen aus, welche angezeigt werden sollen. Die zur Auswahl stehenden Mentoren sollen dabei bereits die durch die anderen Filter eingeschränkten Mentoren sein.

- Einfache Variante: eine Checkbox per Name
- Komfortable Variante: React-Select,¹³ dabei können Mentoren per Drop-Down-Menu ausgewählt oder per Suchfeld gesucht werden.
- Die maximale Anzahl Mentoren soll dort visualisiert werden, damit man sieht, wieviele in Frage kommen.

6.2.2 Layout

TODO

Filter Links / oben / unten / rechts?

-> Stundenplan soll genug platz haben, evtl. floating filters oder collapsable

→ Optisch klar hervorheben, was zeitfenster des schülers und was zeitfenster des mentors ist

¹³Siehe ("Github - React-Select")

6.2.3 Vergleich der Zeitfenster als Kalenderansicht

Damit die Darstellung auch bei mehreren Mentoren übersichtlich bleibt, bietet sich eine kalenderartige Darstellung der Zeitfenster an, ähnlich wie sie bis jetzt bereits verwendet wird.

Abbildung 10 zeigt die geplante Darstellung des Stundenplan eines Schülers, wobei die Zeitfenster des Schülers grün hinterlegt sind. Es wurden drei Mentoren zum Vergleich ausgewählt und mit jeweils unterschiedlichen Farben dargestellt. Werden noch mehr Mentoren ausgewählt, so werden die Balken entsprechend kleiner. Es ist dabei auch möglich, dass die Beschriftungen auf den Balken nicht mehr lesbar sind. Daher wird unterhalb eine Legende eingeblendet. Die Farben können aus einer Palette oder mittels Farb-Rotation ausgewählt werden.

Die maximale Anzahl anzeigbarer Mentoren sollte allenfalls eingeschränkt werden, damit sich die Farben nicht wiederholen und die Balken eine vernünftige Breite beibehalten.

Stundenplan von Schwarz Sandra

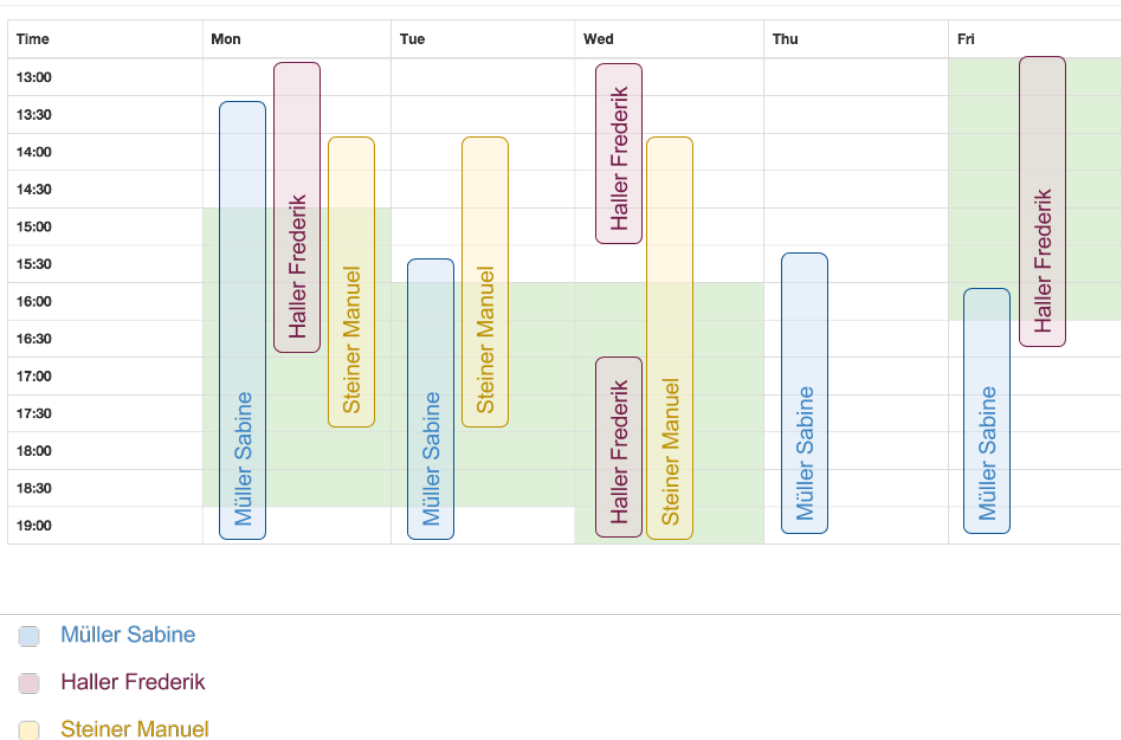


Abbildung 10: Bearbeitung eines Mentors (Mock)

In der Abbildung sind Zeitfenster von Mentoren, die sich nur knapp oder gar nicht mit dem des Schülers überschneiden, nicht speziell berücksichtigt (z.b. Müller Sabine am Donnerstag und Freitag). Es wäre denkbar, diese ein- und ausblendbar zu machen oder anders zu kennzeichnen (schmäler, andere Farbgebung, Transparenz).

6.2.4 Auswahl des Zeitpunkt des Treffens

Möchte ein Administrator den Zeitpunkt des Treffens auswählen, so kann auf eine Zelle des Stundenplans geklickt werden, wie in Abbildung 11 gezeigt. Dies ermöglicht es dem Benutzer, mit nur einem Klick einen Mentor inklusive Zeitpunkt des Treffens und Wochentag zuzuweisen und erfüllt somit **Sc-005 v-1** und **Sc-015 v-1**, und reduziert stark die Anzahl Klicks wie in **NFR-04** gefordert.

Stundenplan von Schwarz Sandra

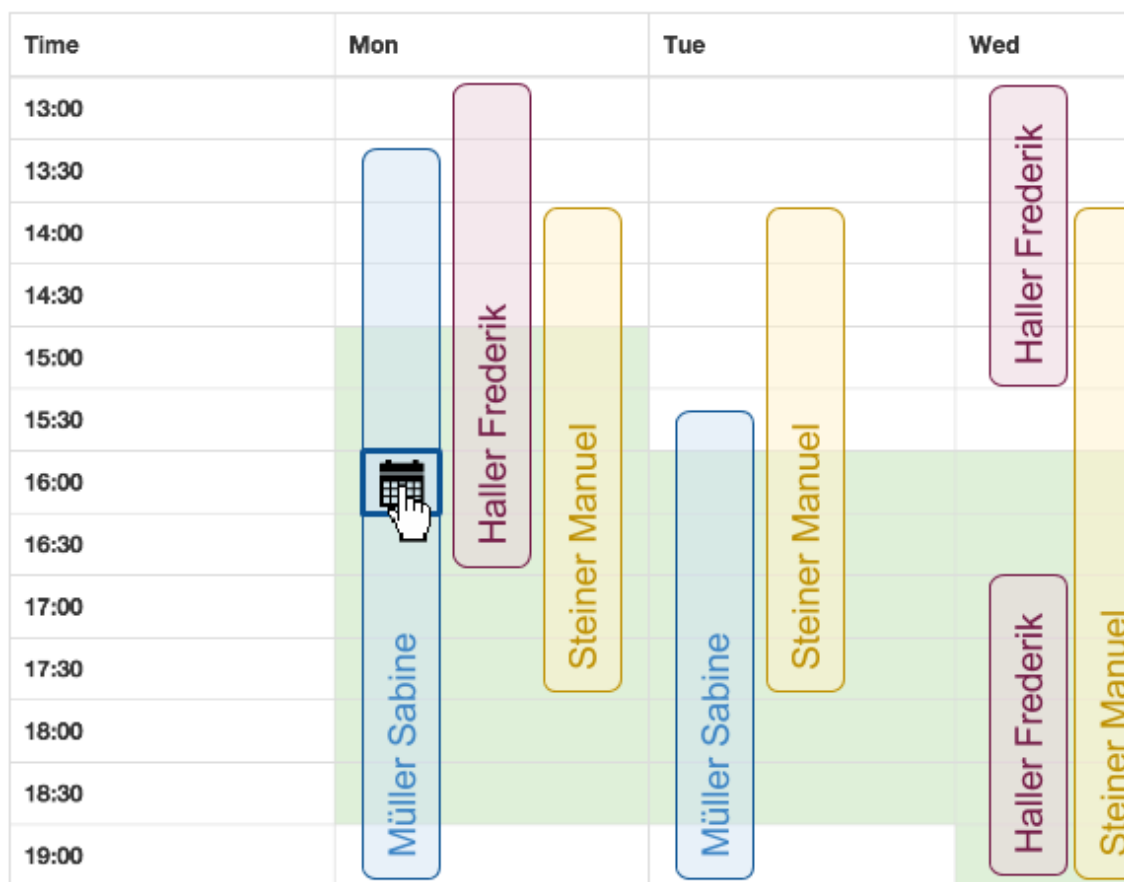


Abbildung 11: Auswahl eines Zeitpunkt des Treffens von Schüler und Mentor (Mock)

6.2.5 Komponenten-Diagramm

Aus den Mocks für die Mentoren-Auswahl lassen sich folgende GUI-Komponenten ableiten, wie in Abbildung 12 gezeigt.

Mentoren-Suche Diese Komponente ist die übergeordnete Komponente, mit der der Benutzer interagiert. Sie erhält von einem Server-Controller die nötigen Daten und sendet ggf. Daten, wie der gewählte Mentor und der Zeitpunkt des Treffens zurück an den Server.

Stundenplan Zeigt den Stundenplan/Zeitfenster des Schülers und der gewählten Mentoren und ermöglicht Auswahl eines Zeitfensters (vgl. 6.2.3).

Mentoren-Auswahl Wählt die Mentoren für die Anzeige auf dem Stundenplan aus

Mentoren-Filter Ermöglicht Filtern der Mentoren nach Kriterien (vgl. 6.2.1).

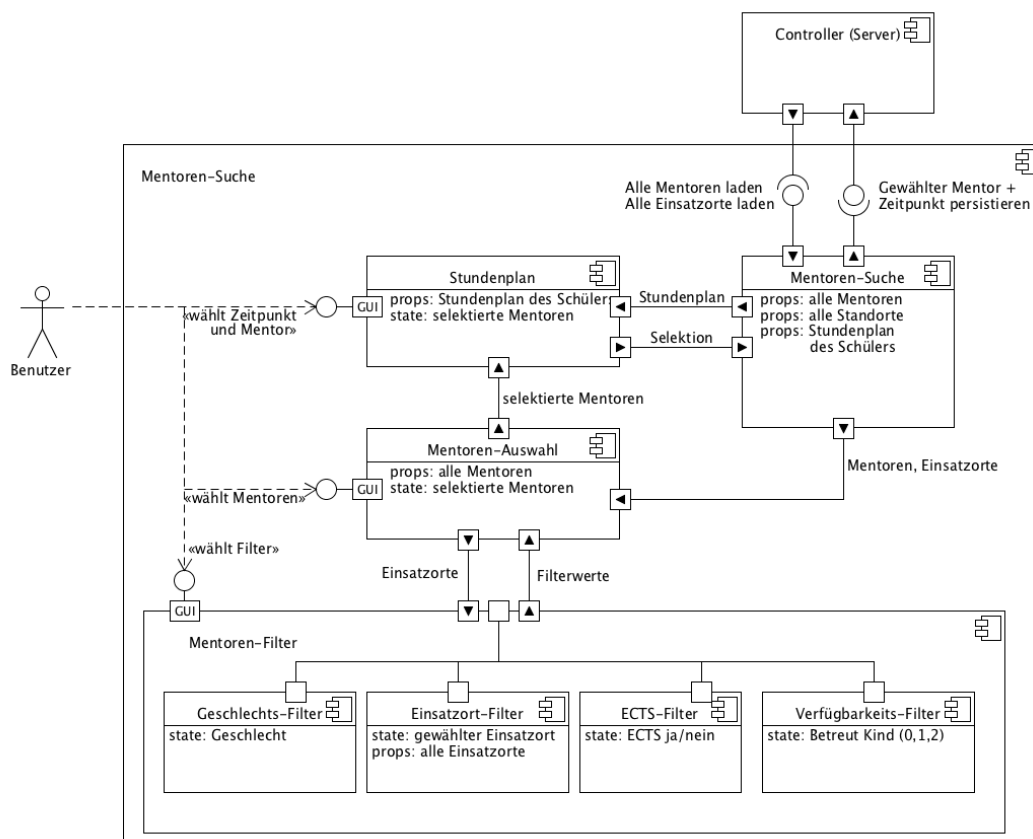


Abbildung 12: Komponenten-Diagramm der Mentoren-Suche

6.3 Daten-Schemas

TODO: uml

Die geplante Komponente muss folgende Daten kennen:

```

1
2 mentors =
3     type: [Mentor]
4

```

```

5 schools =
6   type: [School]
7
8 School =
9   id:
10    type: String
11   name:
12    type: String
13
14 Mentor =
15   id:
16    type: String
17   firstname:
18    type: String
19   lastname:
20    type: String
21   gender:
22    type: ["m", "f"]
23   school:
24    type: String # school-fk
25   kids:
26    type: [String] #Array of Kid-Ids
27   secondaryKids:
28    type: [String] #Array of Kid-Ids
29   ects:
30    type: Boolean
31   timetable:
32    type: Array
33    0: # monday
34       type: [String] # hh:mm
35    1: # tuesday
36       type: [String]
37    # usw.
38
39 Kid =
40   id:
41    type: String
42   firstname:
43    type: String
44   lastname:
45    type: String
46   timetable:
47    type: Array
48    0: # monday
49       type: [String] # Array of hh:mm
50    1: # tuesday
51       type: [String]
52    # usw.

```

6.4 Technologie-Wahl

Einige der Probleme in 4.2 sind der Tatsache geschuldet, dass Ruby on Rails historisch bedingt einen starken Fokus auf klassische, Ressourcen-getriebene *Request-Response*-Anwendungen hat. Diese Klasse von Webanwendungen betrachten jeden ihrer Seiten oder Bildschirme als adressierbare Resource und modellieren Änderungen an ihren Ressourcen mittels HTTP-Verben, wie GET, POST, PUT, DELETE. Dieses als *Representational State Transfer* oder kurz *REST* bezeichnete Paradigma wird in *Ruby on Rails* und damit auch in *Future Kids* konsequent eingesetzt.

Eine Schwierigkeit bei diesem Paradigma ist es, den Status oder englisch *State* einer Anwendung zu

modellieren. REST sagt uns, dass jede Adresse genau einen Seiteninhalt repräsentiert.¹⁴ Im Sinne eines *States* einer Webanwendung muss somit jedem *State* und somit jedem möglichen Bildschirminhalt eine Adresse zugewiesen werden können. Dieses Problem ist der *State*-losen Natur des HTTP-Protokolles geschuldet.

Als Alternative kann mittels *Session* der Status einer Anwendung an einen Besucher zugewiesen werden. Dieses Vorgehen verletzt das REST-Prinzip, bietet aber klare Vorteile in Fällen, bei denen der Status eines Besuchers geheim sein soll oder wo die Anzahl für die Abbildung des Status benötigten Parameter zu gross wird.

Trotz dieser Möglichkeiten bleibt das Problem des langen Weges zwischen einer Statusänderung durch einen Anwender, der Übertragung dieser Information an den Server, der Verarbeitung der Information durch den Server, das Erzeugen der Datenrepräsentation auf dem Server und letztendlich dem Herunterladen und Anzeigen dieser Daten. Dieser lange *Round-Trip* zwischen Client-Server-Client verstösst gegen ein Lokalitätsprinzip, bei welchem auf Daten, die häufig zusammen genutzt werden auch mit gleicher Zugriffszeit aufgerufen werden können. Sie müssen somit in der Speicherhierarchie auf der gleichen Schicht liegen und bei Client-Server-Anwendungen insbesondere auf dem gleichen Gerät.

Bei Webanwendungen ist beispielsweise der durch HTML beschriebene DOM-Baum im Client im Arbeitsspeicher gespeichert. Da eine *State*-Änderung häufig sichtbare Änderungen nach sich zieht, wird bei einer Änderung am *State* bei einer klassischen Webanwendung nun ein neuer Seiten-Request an den Server gesendet, welcher eine neue HTML-Seite mit der Veränderten Ansicht zurückliefert. Der Client verwirft seinen alten DOM-Baum im Arbeitsspeicher und zeichnet die geänderte Seite neu. Dies bringt Wartezeiten mit sich und erhöht die Komplexität der Webanwendung: *State*-Änderungen müssen mehrere Schichten der Architektur durchdringen.

Bei der bisherigen Anwendung *Future Kids* zeigt sich dies damit, dass die bisherige Lösung für die Mentorensuche einerseits rudimentär (durch die verhältnismässig hohe Komplexität des Problems) ausgefallen ist, andererseits viele Seiten-Reloads und Klicks benötigt - die Architektur bildet diesen Fall schlicht nicht gut genug ab.

6.4.1 Client-Side-Komponenten

Durch die Verwendung von Javascript können Client-Seitige Anwendung gebaut werden. Die Verwendung von Javascript aber lange Zeit eingeschränkt durch langsame Ausführung, beschränkter Browser-Support oder fehlende Frameworks, welche der sehr unterschiedlich nutzbaren Sprache Struktur gaben. Javascript-Frameworks erfreuen sich zur Zeit des Schreibens dieser Arbeit grosser Beliebtheit. Erst Backbone, dann Ember und Angular modellieren alle Teile einer Client-seitigen Anwendung (häufig als Model-View-Controller-Architektur), neuere Frameworks wie Meteor, Derby oder Volt decken gar Client- und Server-Schichten einer Anwendung ab.

6.4.2 React

Facebooks *React* im Gegenzug beschränkt sich auf das Modellieren von Client-seitigen, in sich abgeschlossenen Komponenten und verzichtet bewusst auf weitere Schichten. Es lässt sich somit sehr gut in bestehende Frameworks, wie Ruby on Rails integrieren und bietet sich daher für die Erweiterungen an *FutureKids* an. React ist im Projekt bereits für eine Komponente in Verwendung.

React ersetzt oder ergänzt die *View*-Schicht von Ruby on Rails, ist dabei selbst aber keine reine View-Schicht. Vielmehr werden einzelne Komponenten als *Mini-Anwendungen* realisiert, welche jeweils ihre eigenen Daten (*Properties*), Status (*State*) und Darstellung (in Form von HTML), sowie Kontrollfluss (als Javascript-Code) haben und lose miteinander interagieren. React bricht somit leicht mit dem MVC-Entwurfsmuster, wobei Daten (Model), Präsentation (View) und Kontrollfluss (Controller) getrennt sein sollen, zugunsten von Portabilität und einfacherer Entwicklung.

¹⁴Siehe Einleitung Wikipedia-Artikel ("Wikipedia - Representational State Transfer").

Die geforderte Lokalität von Status und Repräsentation wird hierbei folgendermassen umgesetzt: Die Darstellung ist an den Status und an die Daten gekoppelt, Ereignisse wie Klicks führen zu Änderungen am Status und somit zu einer Änderung der Darstellung - diesmal aber gänzlich innerhalb dieser Client-seitigen Komponente. Sie kann nach Aussen ebenfalls über Ereignisse kommunizieren und kann - in ein Webframework wie Rails eingebettet - auch REST-Schnittstellen aufrufen. In einem bestehenden Projekt müssen dabei nicht alle Views mit React ersetzt werden, vielmehr können einzelne Views punktuell ersetzt werden.

React bietet sich daher für den gegebenen Anwendungsfall sehr gut an; die geplanten GUI-Komponenten können somit als React-Komponenten umgesetzt werden.

7 Umsetzung Prototyp

7.1 Verwendete Werkzeuge

7.1.1 Source-Code-Kontrolle

Als Werkzeug für die Source-Code-Kontrolle wurde *Git* verwendet, wobei das Projekt bereits auf *Github* gehostet wird.

Es wurde ein Branch *feature/schedules* erstellt, ausgehend vom aktuellsten Entwicklungsstand. Die Arbeit wurde regelmässig in die Versionskontrolle übertragen und nach Github hochgeladen

7.1.2 IDE

Als Entwicklungsumgebung wurde RubyMine 7.1.2 verwendet, welche es erlaubt *Ruby on Rails*-Applikationen direkt auszuführen und zu debuggen. RubyMine vereinfacht auch das Ausführen von *Specs* oder *Tests*.

Zusätzlich wurde *Sublime Text 3* verwendet, da dieser aktuell für das verwendete *cjsx*-Format eine bessere Syntax-Unterstützung bietet.

7.1.3 React

React ist in *FutureKid* über das *gem*¹⁵ '*react-rails*' bereits integriert, welches die *react*-Laufzeitumgebung enthält, *jsx*-Dateien kompiliert und in die Asset-Pipeline¹⁶ von Ruby on Rails integriert, sowie eine Integration der Komponenten in die Rails-View-Schicht.¹⁷ Ferner können damit React-Komponenten serverseitig gerendert werden..¹⁸ Ergänzt wurde das *gem* '*sprockets-coffee-react*', welches den *CJSX*-Syntax unterstützt. Zu *JSX* und *CJSX* siehe nachfolgender Abschnitt.

7.1.4 JSX und CJSX

JSX erweitert Javascript um eine XML-artige Syntax, in welcher React-Komponenten ähnlich wie *HTML* erstellt werden können:

```
1  var dropdown =
2    <Dropdown>
3      A dropdown list
4      <Menu>
5        <MenuItem>Do Something</MenuItem>
6        <MenuItem>Do Something Fun!</MenuItem>
7        <MenuItem>Do Something Else</MenuItem>
8      </Menu>
9    </Dropdown>;
10
11 render(dropdown);
```

¹⁹

¹⁵Gems ("Edelsteine") sind Pakete oder Module für Ruby, welche jeweils eine bestimmte Aufgabe erfüllen. Ein gem kann andere gems als Abhängigkeiten benötigen.

¹⁶Zusatzdateien zu *HTML*-Seiten, wie *Scripts* und *Stylesheets* können in *Ruby on Rails* vorkompiliert und optimiert werden. Diesen Ablauf erledigt die *Asset Pipeline*

¹⁷Siehe Quelle ("Github - React-Rails")

¹⁸Serverseitiges Rendering (SSR) soll es unter anderem Suchmaschinen leichter machen, Web-Anwendung zu indizieren und zu durchsuchen, zudem kann die initiale Ladezeit von Webseiten reduziert werden. Für *FutureKids* ist dies nicht von Bedeutung.

¹⁹Beispiel angelehnt an: <https://facebook.github.io/jsx/>

CJSX ist eine Variante von *JSX*, welcher *CoffeeScript*, statt Javascript zugrunde liegt.

CoffeeScript ist ein *Dialekt* von JavaScript, wobei Klammern häufig weggelassen und stattdessen Blöcke mittels Einrücken codiert werden. Weiterhin bietet *CoffeeScript* eine Reihe an *Syntactic Sugar*, d.h. Syntaktische Erweiterungen, welche häufig benötigte Konstrukte vereinfachen.²⁰

CJSX vereint nun CoffeeScript mit der XML-Erweiterungen von JSX:

```
1 var dropdown =
2   <Dropdown>
3     { <h1>A dropdown list.</h1> if @state.showTitle }
4     <Menu>
5       {
6         for menu in menuItems
7           <MenuItem
8             id=menu.id
9             isActive=@state.activeMenu is menu.id
10            >{menu.label}</MenuItem>
11       }
12     </Menu>
13   </Dropdown>;
14
15 render(dropdown);
```

Da in *FutureKids* weitgehend *CoffeeScript* verwendet wird, ist deren Verwendung auch für React-Komponenten naheliegend. Als Alternative gibt es die vereinfachte JSX-CoffeeScript-Syntax, wobei die JSX-Subsyntax mittels ‘-Zeichen “escaped” wird. Allerdings entsteht dadurch eine Mischsyntax aus Javascript und CoffeeScript. Der Autor dieser Arbeit vertritt die Meinung dass die CJSX-Syntax expressiver ist, als die Mischform und reines JSX, da Schleifen und Kontrollstrukturen sehr einfach in den XML-Strukturen verwendet werden können (Siehe obiges Beispiel).

7.2 Erste Iteration

In einer ersten Iteration wurden die Anforderungen **Sc-001 v-1** - **Sc-015 v-1** gemäss Planung umgesetzt, dabei wurden **Sc-008 v-1** und **Sc-009 v-1** nur graphisch gelöst (wie im Abschnitt 5.4 beschrieben).

Die Anwendung wurde daraufhin auf die Test-Instanz (*Staging-Server*) geladen und durch den Kunden verifiziert.

Die Rückmeldung fiel sehr positiv aus. In einem weiteren Treffen wurden Anpassungen besprochen. Dabei wurden die folgenden Stories erstellt:

TODO: screenshots

7.3 Zweite Iteration

²⁰Nähers zu CoffeeScript auf <http://coffeescript.org/>

8 Testing

Akzeptanzkriterien Id vergeben und erfüllt / nicht erfüllt

■ oder automatisch testsuite

(Abnahmeprotokol)

9 Fazit / Ausblick

Statement vom Kunde

A Anhang

A.1 Quellcode

Quellcode der Arbeit:

(TODO)

A.2 Quellenangaben

Titelbild:

(TODO)

, and. "ESSENTIAL SCRUM - A Practical Guide to the Most Popular Agile Process."

, and. "Masterthesis - Die Anwendung Eines Genetischen Algorithmus Auf Das Timetabling Problem Einer Grundschule." <http://www.math.uni-magdeburg.de/~lange/papers/masterthesis.pdf>.

"AOZ Startseite." <https://www.stadt-zuerich.ch/content/aoz/de/index.html>.

"ECTS - Wikipedia." http://de.wikipedia.org/wiki/European_Credit_Transfer_System.

"FET - Free Timetabling Software." <http://laescu.ro/liviu/fet/>.

"Github - React-Rails." <https://github.com/reactjs/react-rails>.

"Github - React-Select." <https://github.com/JedWatson/react-select>.

"International Timetabling Competition 2007." <http://stackoverflow.com/a/8578980/1463534>.

"JBoss Optaplanner." <http://www.optaplanner.org/>.

"The Complexity of Timetable Construction Problems." <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.222&rep=rep1&type=pdf>.

"Unitime.org Constraint Solver Library (CPSolver 1.3)." <http://www.unitime.org/index.php?tab=1>.

"Über Die AOZ." <https://www.stadt-zuerich.ch/aoz/de/index/aoz.html>.

"Wikipedia - Representational State Transfer." https://de.wikipedia.org/wiki/Representational_State_Transfer.