

# Future Kids

Semesterarbeit - Werkzeug zur Unterstützung der Stundenplanerstellung

Marco Wettstein

2015-07-22

### Zusammenfassung

Die Asyl Organisation Zürich (AOZ) betreibt mit *Future Kids* ein gemeinnütziges Projekt, welches Primarschüler und -schülerinnen fördert, die bei der Bearbeitung ihrer schulischen Aufgaben zu Hause nur wenig Unterstützung erhalten. Jedem Schüler wird dabei von *Future Kids* ein Mentor oder eine Mentorin zur Seite gestellt, welche(r) den Schüler wöchentlich einmal bei den Lernaufgaben unterstützt. Der Administration der AOZ steht für die Einsatzplanung der Mentoren und weitere administrative Aufgaben eine gleichnamige Webanwendung *FutureKids* zur Verfügung, welche durch Freiwillige weiterentwickelt wird und auf dem Framework *Ruby on Rails* basiert.

Diese Anwendung erlaubt es den Administratoren unter anderem jedem Schüler einen Haupt- und einen Ersatzmentor zuzuweisen. Um einen passenden Mentoren zu finden muss die Administration allerdings Zeitfenster von Schüler und Mentoren, Geschlecht, Einsatzort und weitere Kriterien berücksichtigen. Diese Aufgabe deckte die Anwendung allerdings nur unzureichend und rudimentär ab, was zu erheblichen administrativen Mehraufwänden führte.

Das Ziel dieser Arbeit war es, die Mängel der bisherigen Lösung in diesem Bereich aufzudecken und das Modul für das Finden und Zuweisen eines Mentors neu zu konzipieren und zu implementieren. Dabei wurde zusammen mit dem Kunden die Anforderungen als User-Stories erarbeitet und insbesondere mit Anforderungen an die Bedienbarkeit, Komfort und Übersicht ergänzt.

Schrittweise wurde aus den Anforderungen Lösungskonzepte für die bestehenden Probleme erarbeitet, implementiert und mit den Anforderungen gegengeprüft (automatisiert und manuell), sowie durch den Kunden verifiziert. Dabei fand ein iteratives, an SCRUM-angelegtes Vorgehen statt, wobei Implementierung und Abnahme durch den Kunden mehrmals stattfanden.

Die bisherige, auf einfachen HTML-Formularen basierende Lösung wurde so durch eine im Javascript-Framework *React* entwickelte moderne und responsive Client-Komponente ersetzt und in die Anwendung integriert.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
1.1	Ausgangslage . . . . .	6
1.2	Vorgehen . . . . .	6
1.2.1	Anforderungsanalyse . . . . .	6
1.2.2	Ist-Analyse . . . . .	6
1.2.3	Konzept, Lösungsvarianten und Entscheid . . . . .	7
1.2.4	Umsetzung des Prototyps . . . . .	7
1.2.5	Testing und Abnahme . . . . .	7
1.3	Planung . . . . .	7
<b>2</b>	<b>Anforderungsanalyse</b>	<b>8</b>
2.1	Stakeholder-Analyse . . . . .	8
2.1.1	Asyl Organisation Zürich (AOZ) . . . . .	8
2.1.2	Pädagogischen Hochschule Thurgau (PHTG) . . . . .	8
2.1.3	Administration der AOZ . . . . .	8
2.1.4	Mentoren . . . . .	8
2.1.5	Schüler . . . . .	8
2.1.6	Lehrpersonen . . . . .	8
2.2	Funktionale Anforderungen: User Stories . . . . .	9
2.3	Nicht-Funktionale Anforderungen . . . . .	13
2.3.1	Definition of done . . . . .	13

<b>3 Ist-Analyse</b>	<b>14</b>
3.1 Bestehende Lösung . . . . .	14
3.1.1 Mentoren . . . . .	14
3.1.2 Schüler . . . . .	15
3.2 Probleme der bestehenden Lösung . . . . .	19
3.2.1 Nicht-Funktionale Anforderungen (NFR) . . . . .	19
<b>4 Konzept</b>	<b>22</b>
4.1 Variante 1: Automatisierte Zuweisung . . . . .	22
4.1.1 Das Stundenplanproblem . . . . .	22
4.1.2 Verfügbare Lösungen zur Stundenplanerfestellung . . . . .	22
4.1.3 Diskussion . . . . .	22
4.2 Variante 2: Bestehende Lösung (manuelle Zuweisung) optimieren . . . . .	23
4.2.1 Diskussion . . . . .	23
4.3 Entscheidung . . . . .	23
4.4 Feinkonzept: Bestehende Lösung (manuelle Zuweisung) optimieren . . . . .	24
4.4.1 Anforderungen SC-001 v1 SC-002 v1, SC-003 v1, SC-004 v1 . . . . .	24
4.4.2 Anforderungen SC-005 v1, SC-006 v1, SC-007 v1, SC-015 v1 . . . . .	25
4.4.3 Anforderungen SC-008 v1, SC-009 v1 . . . . .	25
4.4.4 Anforderungen SC-010 v1 . . . . .	26
4.4.5 Anforderung SC-011 v1 . . . . .	26
4.4.6 Anforderung SC-012 v1 . . . . .	26
4.4.7 Anforderung SC-013 v1 . . . . .	26
4.4.8 Anforderung SC-014 v1 . . . . .	26
<b>5 Design &amp; Architektur</b>	<b>27</b>
5.1 Mentoren-Auswahl . . . . .	27
5.1.1 Filter und Auswahl für den Vergleich . . . . .	27
5.1.2 Vergleich der Zeitfenster als Kalenderansicht . . . . .	29
5.1.3 Auswahl des Zeitpunkt des Treffens . . . . .	29
5.1.4 Komponenten-Diagramm . . . . .	29
5.2 Daten-Schemas . . . . .	32
5.3 Technologie-Wahl . . . . .	33
5.3.1 Client-Side-Komponenten . . . . .	33
5.3.2 React . . . . .	33

<b>6 Umsetzung Prototyp</b>	<b>35</b>
6.1 Verwendete Werkzeuge . . . . .	35
6.1.1 Source-Code-Kontrolle . . . . .	35
6.1.2 IDE . . . . .	35
6.1.3 React . . . . .	35
6.1.4 JSX und CJSX . . . . .	35
6.2 Erste Iteration . . . . .	37
6.2.1 Implementierung . . . . .	37
6.2.2 Verifizierung . . . . .	44
6.2.3 Verifizierung der NFR . . . . .	45
6.2.4 Definition of Done: . . . . .	45
6.3 Zweite Iteration . . . . .	46
6.3.1 Neue Anforderungen . . . . .	46
6.3.2 Umsetzung und Verifizierung . . . . .	47
6.3.3 Verifizierung der NFR (Ergänzungen) . . . . .	47
6.3.4 Definition of Done . . . . .	47
<b>7 Testing</b>	<b>51</b>
7.1 User-Stories . . . . .	51
7.2 Nicht-Funktionale Anforderungen . . . . .	52
7.3 Definition of Done . . . . .	52
7.4 Kundenfeedback . . . . .	52
<b>8 Rückblick</b>	<b>53</b>
8.1 Schwierigkeiten . . . . .	53
<b>9 Ausblick</b>	<b>54</b>
<b>A Anhang</b>	<b>55</b>
A.1 Quellcode . . . . .	55
Quellenangaben . . . . .	55

## Abbildungsverzeichnis

1 Bearbeitung eines Mentors (Bestehende Lösung) . . . . .	14
2 Editieren der Zeitfenster eines Mentors (Bestehende Lösung) . . . . .	15
3 Schüler Profil-Seite (Bestehende Lösung) . . . . .	16
4 Schüler Profil-Seite - Beziehungen (Bestehende Lösung) . . . . .	17
5 Zuweisen eines Mentoren zu einem Schüler (Bestehende Lösung) . . . . .	17
6 Bearbeitung des Stundenplans eines Schülers (Bestehende Lösung) . . . . .	18

7	Hier können Mentoren ausgewählt werden, deren Zeitfenster mit einem Schüler verglichen werden (Bestehende Lösung) . . . . .	18
8	Werden viele Mentoren für den Vergleich ausgewählt, leidet die Übersicht (Bestehende Lösung) . . . . .	20
9	Erfassen der Zeitfenster - bei der bisherigen Lösung können die markierten Bedienelemente entfernt werden . . . . .	24
10	Als einfache HTML-Komponente umgesetzter Entwurf für die Mentorensuche und der Filter. Die Zeitfenster des Schülers sind grün hinterlegt. . . . .	28
11	Konzipierte Darstellung für den Vergleich der Zeitfenster von Mentoren und des Schülers. Die Darstellung lehnt sich an bekannte Kalender-Anwendungen an. Verschiedene Farben der Mentoren sollen dabei die Orientierung erleichtern . . . . .	30
12	Auswahl eines Zeitpunkt des Treffens von Schüler und Mentor (Mock) . . . . .	30
13	Komponenten-Diagramm der Mentoren-Suche. Die Mentoren-Suche ist dabei die zentrale Komponente. Sie erhält Daten vom serverseitigem Controller und leitet sie an ihre Unter-Komponente weiter. Diese wiederum kommunizieren mit ihren übergeordneten Komponenten mit Ereignissen. Die jeweils in eine Richtung gerichteten Pfeile sollen dabei diesen Datenfluss visualisieren. . . . .	31
14	Ein neuer Menüpunkt öffnet die neue Mentorensuche . . . . .	37
15	Die neue Mentorensuche weist jedem Mentor eine Farbe zu (dynamisch abhängig von der Anzahl mentoren). Sie wurde als <i>React</i> -Komponente implementiert. . . . .	38
16	Bei einem Klick auf einen Zeitfenster kann der Mentor dem Schüler direkt zugewiesen werden. Es sind nur diejenigen Zeitfenster klickbar, welche sowohl von Schüler als auch von Mentor belegt sind. . . . .	39
17	Struktur des Bildschirms für die Mentorensuche, ausgehend vom aufgerufenen <i>Kids-Controller</i> , dargestellt als "Querschnitt" durch die Anwendung in Paket-Struktur. Die einzelnen Pakete zeigen dabei eine Schicht oder Komponente der Anwendung . . . . .	40
18	Innere Struktur der umgesetzten Mentoren-Suche-Komponente. Gegenüber dem Konzept wurde der Aufbau etwas vereinfacht. . . . .	41
19	Filter-Komponente mit Mentorenauswahl. Die Mentorenauswahl wurde mit <i>react-select</i> umgesetzt und ermöglicht eine Namenssuche . . . . .	42
20	Ausgabe der Testsuite von <i>RubyMine</i> nach der ersten Iteration . . . . .	44
21	Die Auswahl für den Mentoren zur Anzahl der betreuten Kinder, wurde angepasst. Zudem wurde das zuvor nicht sichtbare Formular, welches der Seite zu Grunde liegt, sichtbar gemacht, da es nützliche Informationen offenbart. Zudem kann dadurch der Mentor manuell geändert werden. . . . .	48
22	Sind schon beide Mentoren zugewiesen, erscheint nun eine Fehlermeldung . . . . .	49
23	Ausgabe der Testsuite von <i>RubyMine</i> nach der zweiten Iteration . . . . .	50

## Listings

1	Mentor-Klasse in Ruby (mentor.rb) . . . . .	15
2	Benötigte Daten für die Komponente . . . . .	32
3	Beispielcode in JSX . . . . .	35
4	Beispielcode in CJSX . . . . .	36
5	Hinzugefügte Methode auf dem KidsController. Sie liefert die Daten für die Komponente aus . . . . .	40
6	Beispiel für das herausfiltern mit nicht-passendem Geschlecht . . . . .	42
7	Diese Funktion weist jedem Mentor eine Farbe zu . . . . .	43

# 1 Einleitung

## 1.1 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, welche(r) 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.

## 1.2 Vorgehen

Die Arbeit wird in folgende Phasen unterteilt:

### 1.2.1 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.<sup>1</sup>

Zu jeder Story gehören:

<b>SC-Nummer vVersion</b>	Titel, eigentliche Story
<i>Beschreibung</i>	Beschreibung
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Akzeptanzkriterium 1 <input type="checkbox"/> Akzeptanzkriterium 2
<i>Priorität</i>	PRIORITÄT

**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".

### 1.2.2 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.

---

<sup>1</sup>User 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. ("Kenneth S. Rubin - ESSENTIAL SCRUM - A Practical Guide to the Most Popular Agile Process")

### 1.2.3 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.

### 1.2.4 Umsetzung des Prototyps

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

### 1.2.5 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.

Gegebenenfalls wird nach der Überprüfung durch den Anwender eine weitere Umsetzungsphase und anschließende Testphase hinzugefügt (als weitere Scrum-Iteration).

## 1.3 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

## 2 Anforderungsanalyse

### 2.1 Stakeholder-Analyse

#### 2.1.1 Asyl Organisation Zürich (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.<sup>2</sup>

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

#### 2.1.2 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.

#### 2.1.3 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
- Protokollierung

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 2.2) und vertritt die Administration der Future Kids Plattform.

#### 2.1.4 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-Punkten<sup>3</sup> vergütet. Mentoren nutzen die Plattform zum Erfassung von Protokollen, die sie nach jedem Hausbesuch erstellen.

#### 2.1.5 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.

#### 2.1.6 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.

---

<sup>2</sup>Siehe Quellen ("AOZ Startseite"; "Über Die AOZ")

<sup>3</sup>European Credit Transfer System, siehe ("ECTS - Wikipedia")



## 2.2 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.<sup>4</sup>

<b>SC-001 v1</b>	Als Administrator möchte ich die verfügbaren Zeitfenster eines Mentors erfassen
<i>Beschreibung</i>	Jeder Mentor gibt die Zeitfenster an, an welchen er Schüler betreuen kann. Ein Zeitfenster hat einen Wochentag, sowie eine Start- und Endzeit. 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

<b>SC-002 v1</b>	Als Administrator möchte ich die Zeitfenster eines Mentors sehen
<i>Beschreibung</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

<b>SC-003 v1</b>	Als Administrator möchte ich die Zeitfenster eines Schülers erfassen
<i>Beschreibung</i>	Jeder Schüler gibt an, an innerhalb welcher Zeitfenster er oder sie verfügbar ist. Ein Zeitfenster ist analog zur Story <b>SC-001 v1</b> definiert. Ein Administrator soll diese Zeitfenster für einen Schüler speichern können.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Analog Story <b>SC-001 v1</b>
<i>Priorität</i>	MUST

<b>SC-004 v1</b>	Als Administrator möchte ich die Zeitfenster eines Schülers sehen
<i>Beschreibung</i>	Analog zu <b>SC-002 v1</b> sollen die Zeitfenster eines Schülers eingesehen werden.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Analog Story <b>SC-002 v1</b>
<i>Priorität</i>	SHOULD

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

<b>SC-005 v1</b>	Als Administrator möchte ich einem Schüler einen primären Mentor zuweisen können
<i>Beschreibung</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

<b>SC-006 v1</b>	Als Administrator möchte ich einem Schüler einen Ersatzmentor zuweisen können, welcher im Ersatzfall den Schüler betreut
<i>Beschreibung</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

<b>SC-007 v1</b>	Als Administrator möchte ich bei einem Schüler vermerken, dass der Ersatzmentor aktiv ist
<i>Beschreibung</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

<b>SC-008 v1</b>	Als Administrator möchte ich für einen Schüler nach Mentoren mit überschneidenden Zeitfenstern suchen können
<i>Beschreibung</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 <input type="checkbox"/> Der Benutzer erkennt, zu an welchem Wochentag und zu welchem Zeitpunkt die Überschneidung ist
<i>Priorität</i>	MAY

<b>SC-009 v1</b>	Als Administrator möchte ich für einen Schüler nach Mentoren mit nicht ganz überschneidenden Zeitfenstern suchen können
<i>Beschreibung</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

<b>SC-010 v1</b>	Als Administrator möchte ich für einen Schüler nach Mentoren mit passendem Geschlecht suchen können.
<i>Beschreibung</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

<b>SC-011 v1</b>	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche mittels ECTS-Punkten vergütet werden.
<i>Beschreibung</i>	Wie in 2.1.4 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

<b>SC-012 v1</b>	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche an einer bestimmten Schule eingesetzt werden.
<i>Beschreibung</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

<b>SC-013 v1</b>	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche eine bestimmte Anzahl Schüler betreuen
<i>Beschreibung</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

<b>SC-014 v1</b>	Als Administrator möchte ich Mentoren nach Name suchen können.
<i>Beschreibung</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

<b>SC-015 v1</b>	Als Administrator möchte ich den Zeitpunkt und den Wochentag des Treffens eines Mentors mit dem Schüler erfassen.
<i>Beschreibung</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

## 2.3 Nicht-Funktionale Anforderungen

Wie in Abschnitt 1.2.1 erläutert werden die Nicht-Funktionalen Anforderungen als "Definition of Done" erfasst.

### 2.3.1 Definition of done

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

## 3 Ist-Analyse


### 3.1 Bestehende Lösung

Die bestehende Lösung ist eine Webapplikation welche mittels dem Framework *Ruby on Rails*<sup>5</sup> entwickelt wurde. Sie wurde von der Firma Panter AG und Freiwilligen entwickelt, der Quellcode ist auf Github verfügbar<sup>6</sup> und unter der *GNU Affero General Public License*<sup>7</sup> 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.

#### 3.1.1 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](mailto: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](mailto: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 v1 zu erfüllen. Auch die Model-Klasse des Mentors oder der abgeleiteten `User`-Klasse enthält kein passendes Feld.


<sup>5</sup><http://rubyonrails.org/>

<sup>6</sup>[https://github.com/panterch/future\\_kids](https://github.com/panterch/future_kids)


<sup>7</sup>[http://de.wikipedia.org/wiki/GNU\\_Affero\\_General\\_Public\\_License](http://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 v1 und SC-002 v1 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 

---



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)

### 3.1.2 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

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)

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 v1).

Unter “Stundenplan bearbeiten” lassen sich analog zum Mentor die Zeitfenster eines Schülers einstellen (Abbildung 6). Dies erfüllt SC-003 v1 und SC-004 v1 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.



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

# aoz

**Aktionen**

- Bearbeiten
- Zurück zur Liste
- Neues Lernjournal
- Neues Gespräch
- Stundenplan bearbeiten
- Seitenanfang

**Lernjournale** 0

**Gespräche** 0

**Überfachliche Lernziele** Keine Angabe  
**Treffen Wochentag** Keine Angabe  
**Treffen beginnt um** Keine Angabe  
**Lehrperson** [Meckler Janine](#)  
**Zweite Lehrperson** Keine Angabe  
**Mentor/in** [Haller Frederik](#)  
**Ersatz Mentor/in** [Sabine Müller](#)  
**Stellvertretung aktiv** Nein  
**Coach** Keine Angabe

## Projektinterne Informationen

**Eintritt** Keine Angabe  
**Austritt** Keine Angabe  
**Grund für Ausstieg** Keine Angabe  
**Letztes Telefoncoaching** Keine Angabe  
**Auffälligkeiten** Keine Angabe  
**Kritikalität der Auffälligkeiten** Keine Angabe  
**Pendenzen** Keine Angabe  
**Inaktiv gesetzt** Nein

## Beziehungen

Datum	Person	Rolle	Schüler/in
Freitag, 20. Februar 2015, 21:03 Uhr	<a href="#">Meckler Janine</a>	zugeordnet	als Lehrperson <a href="#">Hendrick Elaine</a>
Freitag, 20. Februar 2015, 21:03 Uhr	<a href="#">Meckler Janine</a>	zugeordnet	als Ersteller/in <a href="#">Hendrick Elaine</a>
Sonntag, 15. März 2015, 21:26 Uhr	<a href="#">Haller Frederik</a>	zugeordnet	als Mentor/in <a href="#">Hendrick Elaine</a>
Sonntag, 15. März 2015, 21:26 Uhr	<a href="#">Sabine Müller</a>	zugeordnet	als Ersatz Mentor/in <a href="#">Hendrick Elaine</a>

## Lernjournale

[Neues Lernjournal](#)

## Gesprächsdokumentationen

[Neue Gesprächsdokumentation](#)

AOZ Zypressenstrasse 60 8040 Zürich Telefon: 044 415 65 91 [futurekids@aoz.ch](mailto:futurekids@aoz.ch)

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

**Treffen Wochentag**

**Treffen beginnt um**

**Lehrperson**

**Zweite Lehrperson**

**Mentor/in**

**Ersatz Mentor/in**

☐ Stellvertretung aktiv

**Coach**

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

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)

## 3.2 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.

### 3.2.1 Nicht-Funktionale Anforderungen (NFR)

#### Problemstellung

Die Komponente *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.

<b>NFR-01</b>	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

#### 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.

<b>NFR-02</b>	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.

<b>NFR-03</b>	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



**Aktionen**  
 Auswahl bestätigen  
 Stundenplandaten speichern  
 Seitenanfang

### Stundenplan von Hendrick Elaine

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

AOZ Zugesamtessen 80 8040 Zürich Telefon: 044 415 65 91 [info@aoz.ch](mailto:info@aoz.ch)

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

**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 und es ist nötig, sich den Mentoren und den Termin zu merken.

<b>NFR-04</b>	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.

<b>NFR-05</b>	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 Konzept

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

### 4.1 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.

#### 4.1.1 Das Stundenplanproblem

Prinzipiell handelt es sich beim Problem der Stundenplanerstellung um ein NP-Vollständiges Problem,<sup>8</sup> 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.<sup>9</sup> Bei der im Jahre 2007 durchgeführten "International Timetabling Competition 2007" gewannen jedoch Verfahren in der Art der "Lokalen Suche."<sup>10</sup>

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.

#### 4.1.2 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."<sup>11</sup>

**FET - Free Timetabling Software** ist ein weiterer Quelloffenes Programm zur Stundenplanerstellung von Schulen und Universitäten<sup>12</sup>

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

#### 4.1.3 Diskussion

Diese Variante löst die Bedienprobleme aus 3.2, indem keine oder nur wenige Bedienelemente notwendig gemacht werden, um Mentoren den Schülern zuzuweisen.

---

<sup>8</sup>Vgl. ("Julia Lange - Masterthesis - Die Anwendung Eines Genetischen Algorithmus Auf Das Timetabling Problem Einer Grundschule," Abschnitt 3.2.2). Je nach Rahmenbedingungen können Stundenplanprobleme auch in P liegen, vgl. ("The Complexity of Timetable Construction Problems," Abschnitt 3)

<sup>9</sup>Siehe ("Julia Lange - Masterthesis - Die Anwendung Eines Genetischen Algorithmus Auf Das Timetabling Problem Einer Grundschule," Abschnitt 4)

<sup>10</sup>Vgl. ("International Timetabling Competition 2007")

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

<sup>12</sup>Vgl. ("FET - Free Timetabling Software").

Ebenfalls erfüllt sie wichtige Stories aus 2.2, macht Stories 9-14 allerdings überflüssig, da diese der Auswahl eines Mentors helfen.

**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.

## 4.2 Variante 2: Bestehende Lösung (manuelle Zuweisung) optimieren

Die bestehende Lösung erfüllt bereits die Stories 1-8 aus Abschnitt 2.2, jedoch ist die Bedienbarkeit durch die erwähnten Probleme in Kapitel 3.2 eingeschränkt. Diese können jedoch mit geeigneten Massnahmen gelöst werden. Insbesondere dienen Stories 9-14 als wichtige Hilfestellungen bei der Wahl eines Mentors, welche im Moment noch Fehlen.

In dieser Variante soll erarbeitet werden, mit welchen Konzepten die bestehende Lösung um sämtliche Stories aus 2.2 ergänzt werden kann und welche Massnahmen notwendig sind um die Probleme in 3.2 zu lösen.

### 4.2.1 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

## 4.3 Entscheidung

Beide Varianten erfüllen alle Stories und können alle Probleme von Abschnitt 3.2 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.

## 4.4 Feinkonzept: Bestehende Lösung (manuelle Zuweisung) optimieren

### 4.4.1 Anforderungen SC-001 v1 SC-002 v1, SC-003 v1, SC-004 v1

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, siehe Abbildung 9
- Bedienelemente klar beschriften.

**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



#### 4.4.2 Anforderungen SC-005 v1, SC-006 v1, SC-007 v1, SC-015 v1

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 v1.

##### 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.

#### 4.4.3 Anforderungen SC-008 v1, SC-009 v1

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 v1, SC-011 v1, SC-012 v1, SC-013 v1 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.

#### 4.4.4 Anforderungen SC-010 v1

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.

#### 4.4.5 Anforderung SC-011 v1

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.

#### 4.4.6 Anforderung SC-012 v1

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.

#### 4.4.7 Anforderung SC-013 v1

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.

#### 4.4.8 Anforderung SC-014 v1

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

## 5 Design & Architektur

### 5.1 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 v1**. Diese direkte Zuweisung reduziert die in **NFR-04** beanstandeten Anzahl Klicks.

#### 5.1.1 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. Abbildung 10 zeigt einen Entwurf der möglichen Lösung.

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:

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

Ist nichts ausgewählt, so wird nicht nach Geschlecht gefiltert

**5.1.1.2 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.

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

**5.1.1.4 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.

Verfügbarkeit: 
 Geschlecht: 
 Einsatzort: 
 ECTS:

**Gefilterte mentoren:** Hans Muster, Iris Gleichen, Rainer Zufall,

Mentoren für den Vergleich:

× Hans Muster
× Iris Gleichen
× Rainer Zufall

Time	Mon	Tue	Wed	Thu	Fri
13:00					
13:30					
14:00					
14:30					
15:00					
15:30					
16:00	Hans Muster Iris Gleichen	Hans Muster	Hans Muster Iris Gleichen		
16:30	Hans Muster Iris Gleichen	Hans Muster	Hans Muster Iris Gleichen		
17:00	Hans Muster Iris Gleichen	Hans Muster	Hans Muster Iris Gleichen		Iris Gleichen
17:30	Hans Muster Iris Gleichen	Hans Muster	Hans Muster Iris Gleichen		Iris Gleichen
18:00	Hans Muster Iris Gleichen	Hans Muster	Hans Muster Iris Gleichen		Iris Gleichen
18:30	Hans Muster Iris Gleichen	Hans Muster	Hans Muster Iris Gleichen		Iris Gleichen

Abbildung 10: Als einfache HTML-Komponente umgesetzter Entwurf für die Mentorensuche und der Filter. Die Zeitfenster des Schülers sind grün hinterlegt.

**5.1.1.5 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,<sup>13</sup> 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.

## 5.1.2 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 11 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.

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).

## 5.1.3 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 12 gezeigt. Dies ermöglicht es dem Benutzer, mit nur einem Klick einen Mentor inklusive Zeitpunkt des Treffens und Wochentag zuzuweisen und erfüllt somit SC-005 v1 und SC-015 v1, und reduziert stark die Anzahl Klicks wie in NFR-04 gefordert.

## 5.1.4 Komponenten-Diagramm

Aus den Mocks für die Mentoren-Auswahl lassen sich folgende GUI-Komponenten ableiten, wie in Abbildung 13 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. 5.1.2).

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

**Mentoren-Filter** Ermöglicht Filtern der Mentoren nach Kriterien (vgl. 5.1.1).

---

<sup>13</sup>Siehe ("Github - React-Select")

Stundenplan von Schwarz Sandra

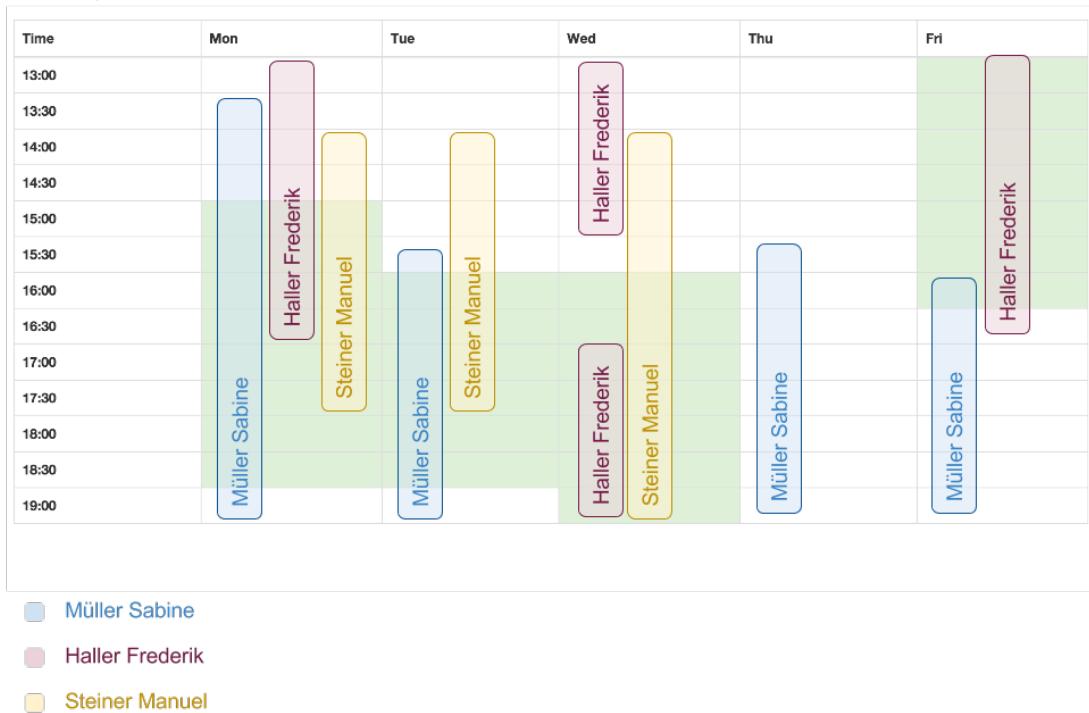


Abbildung 11: Konzipierte Darstellung für den Vergleich der Zeitfenster von Mentoren und des Schülers. Die Darstellung lehnt sich an bekannte Kalender-Anwendungen an. Verschiedene Farben der Mentoren sollen dabei die Orientierung erleichtern

Stundenplan von Schwarz Sandra

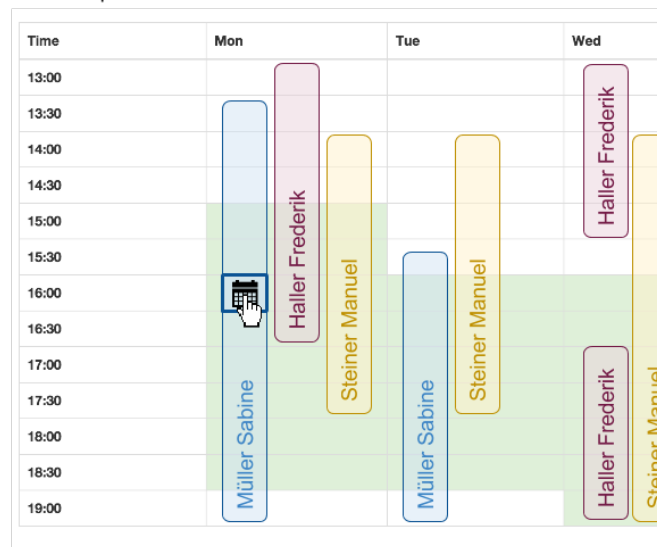


Abbildung 12: Auswahl eines Zeitpunkt des Treffens von Schüler und Mentor (Mock)

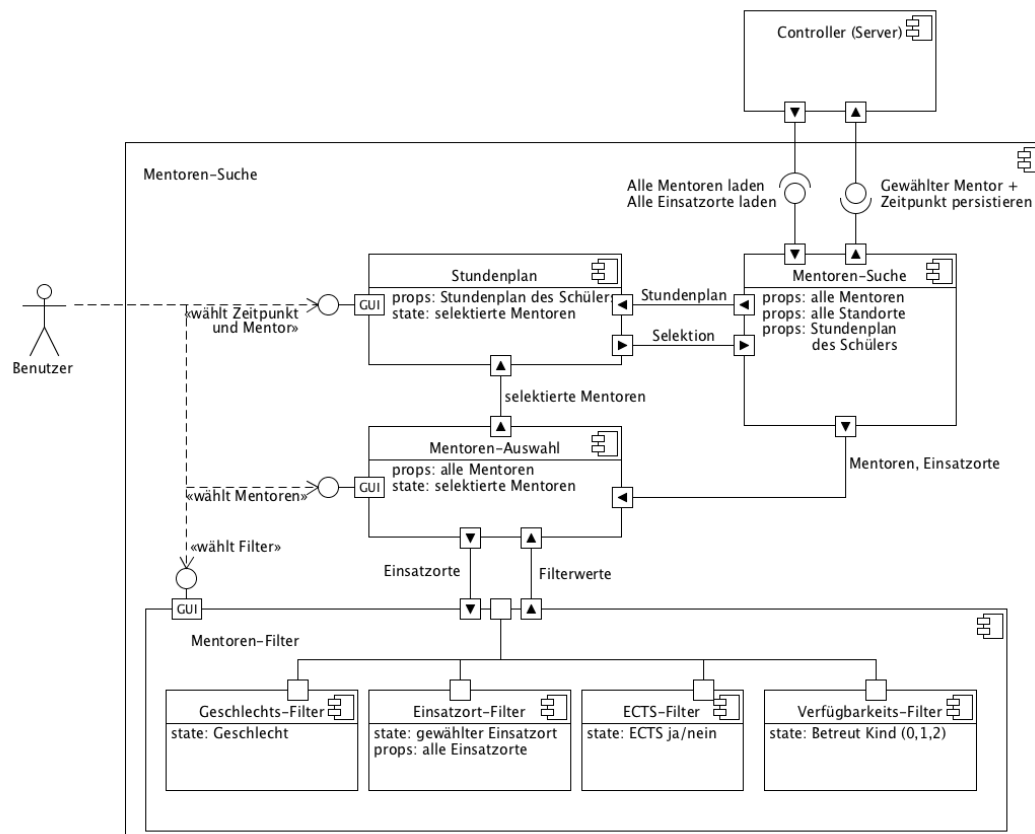


Abbildung 13: Komponenten-Diagramm der Mentoren-Suche. Die Mentoren-Suche ist dabei die zentrale Komponente. Sie erhält Daten vom serverseitigem Controller und leitet sie an ihre Unter-Komponente weiter. Diese wiederum kommunizieren mit ihren übergeordneten Komponenten mit Ereignissen. Die jeweils in eine Richtung gerichteten Pfeile sollen dabei diesen Datenfluss visualisieren.

## 5.2 Daten-Schemas

Die geplante Komponente muss folgende Daten kennen (Darstellung in Anlehnung an ein CoffeeScript-Objekt):

```
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   prename:
18    type: String
19   name:
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   prename:
43    type: String
44   name:
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.
```

Listing 2: Benötigte Daten für die Komponente



### 5.3 Technologie-Wahl

Einige der Probleme in 3.2 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.<sup>14</sup> 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.

#### 5.3.1 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.

#### 5.3.2 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

<sup>14</sup>Siehe Einleitung Wikipedia-Artikel ("Wikipedia - Representational State Transfer").

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 eigene 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.

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.

## 6 Umsetzung Prototyp

### 6.1 Verwendete Werkzeuge

#### 6.1.1 Source-Code-Kontrolle

Als Werkzeug für die Source-Code-Kontrolle kam *Git* zur Verwendung, da das Projekt bereits auf *Github* gehostet wird.

Ein neuer *Branch feature/schedules* protokollierte die Arbeit auf Github, dieser wurde ausgehend vom aktuellsten Entwicklungsstand erstellt. Der Fortschritt der Arbeit wurde regelmässig auf Github geladen und somit gesichert.

#### 6.1.2 IDE

Als Entwicklungsumgebung kam RubyMine 7.1.2 zur Verwendung, 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.

#### 6.1.3 React

React ist in *FutureKid* über das *gem*<sup>15</sup> *'react-rails'* bereits integriert, welches die *react*-Laufzeitumgebung enthält, *jsx*-Dateien kompiliert und in die Asset-Pipeline<sup>16</sup> von Ruby on Rails integriert, sowie eine Integration der Komponenten in die Rails-View-Schicht.<sup>17</sup> Ferner können damit React-Komponenten serverseitig gerendert werden..<sup>18</sup> Ergänzt wurde das *gem* *'sprockets-coffee-react'*, welches den CJSX-Syntax unterstützt. Zu JSX und CJSX siehe nachfolgender Abschnitt.

#### 6.1.4 JSX und CJSX

JSX erweitert Javascript um eine XML-artige Syntax, in welcher React-Komponenten ähnlich wie HTML erstellt werden können<sup>19</sup>:

```
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);
```

Listing 3: Beispielcode in JSX

---

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

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

<sup>17</sup>Siehe Quelle ("Github - React-Rails")

<sup>18</sup>Serverseitiges 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.

<sup>19</sup>Beispiel 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.<sup>20</sup>

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);
```

Listing 4: Beispielcode in CJSX

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 Author 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).

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

## 6.2 Erste Iteration

In einer ersten Iteration wurden die Anforderungen **SC-001 v1** - **SC-015 v1** gemäss Planung umgesetzt, dabei wurden **SC-008 v1** und **SC-009 v1** nur graphisch gelöst (wie im Abschnitt 4.4 beschrieben).

### 6.2.1 Implementierung

Auf der Profilseite eines Schülers wurde eine neue Seite erstellt, die Mentorensuche (Abbildung 14). Diese besteht aus einem versteckten Formular, welches die zu definierenden Werte wie Hauptmentor, Ersatzmentor, Wochentag des Treffens und Zeitpunkt des Treffens enthält. Weiterhin enthält sie die konzipierte *React*-Komponente, welche die es ermöglicht, Mentoren per Filterkriterien zu finden und mit den Zeitfenstern des Schülers zu vergleichen (Abbildung 15).

The image shows a web interface for a student profile named 'Hodler Rolf'. On the left is a sidebar with a list of actions: 'Bearbeiten', 'Zurück zur Liste', 'Neues Lernjournal', 'Neues Gespräch', 'Stundenplan bearbeiten', 'Mentor finden' (highlighted), and 'Seitenanfang'. Below the actions are two buttons: 'Lernjournale' with a counter '0' and 'Gespräche' with a counter '0'. The main content area is titled 'Hodler Rolf' and contains a form with the following fields:

Name	Hodler
Vorname	Rolf
Geschlecht	♀
Geburtsdatum	Keine Angabe
Klassenstufe	Keine Angabe
Erstsprache	Keine Angabe
Name der/s Erziehungsberechtigten	Keine Angabe
Strasse, Nr.	Keine Angabe
PLZ, Ort	Keine Angabe
Telefonnummer	Keine Angabe
Eltern brauchen bei Gesprächen unbedingt einen Dolmetscher	Nein
Weitere Bemerkungen	Keine Angabe
Schule	Primarschule <a href="#">Hinterpfufikon</a>

Below the form is the section header 'Allgemeine Informationen'.

Abbildung 14: Ein neuer Menüpunkt öffnet die neue Mentorensuche

Wählt ein Benutzer einen Mentor, erhält er einen Bestätigungs-Dialog (Abbildung 16). Wird dieser bestätigt, werden die gewählten Werte ins Formular übertragen und das Formular abgeschickt (als POST-, eigentlich PUT-Request). Zwar wäre auch ein AJAX-Aufruf möglich gewesen, d.h. ein asynchroner Aufruf an den Webserver, welcher die Seite nicht neu lädt. Allerdings macht es Sinn, nach gesetztem Wert die Profilseite des Schülers anzuzeigen. Ein Formular entspricht genau diesem Ablauf.

Diagramm 17 fasst diesen Aufbau zusammen. Die Komponente *KidMentorSchedules* entspricht dabei dem konzipierten Aufbau (Diagram 13), jedoch wurde die *Mentoren-Filter*-Komponente direkt der Hauptkomponente (*Mentoren-Suche*) unterstellt, da dies die Kommunikation zwischen den Komponenten vereinfacht (vgl. Diagramm 18). Die Haupt-Komponente trägt dabei sowohl den *State* aller gewählter Filter, sowie den *State* der Mentoren-Auswahl für die Darstellung.

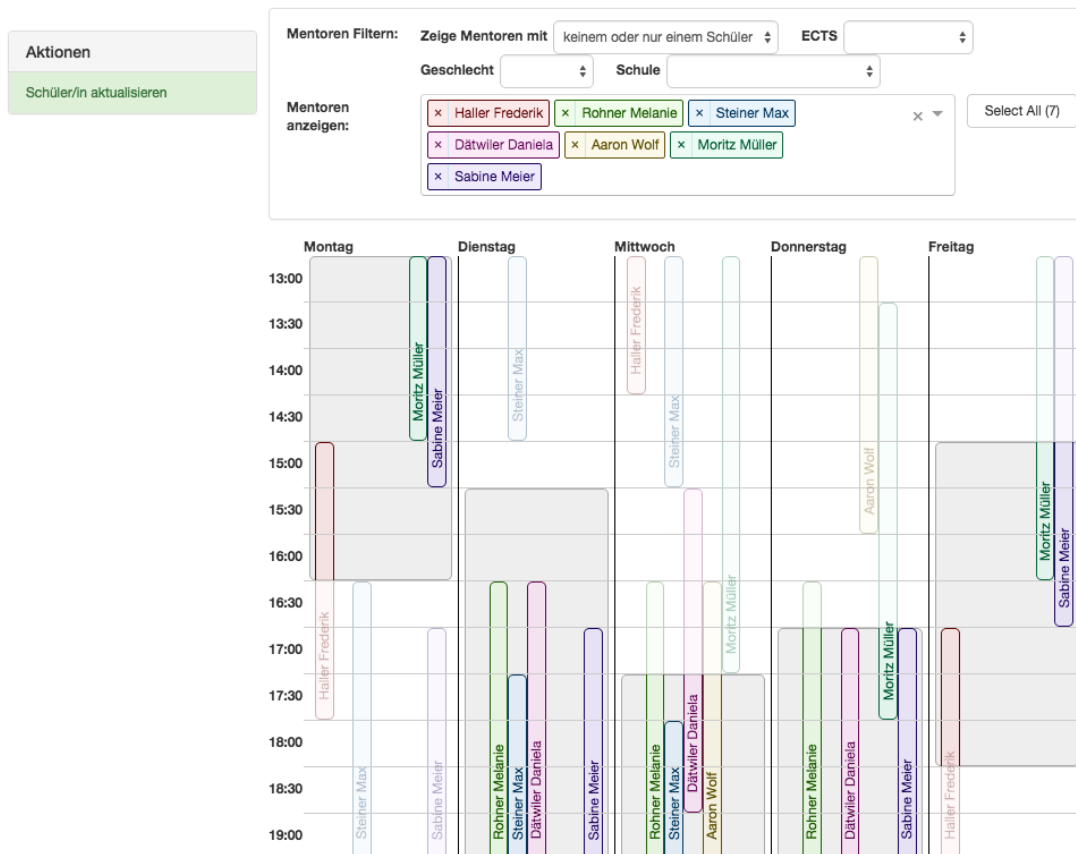


Abbildung 15: Die neue Mentorensuche weist jedem Mentor eine Farbe zu (dynamisch abhängig von der Anzahl mentoren). Sie wurde als *React*-Komponente implementiert.

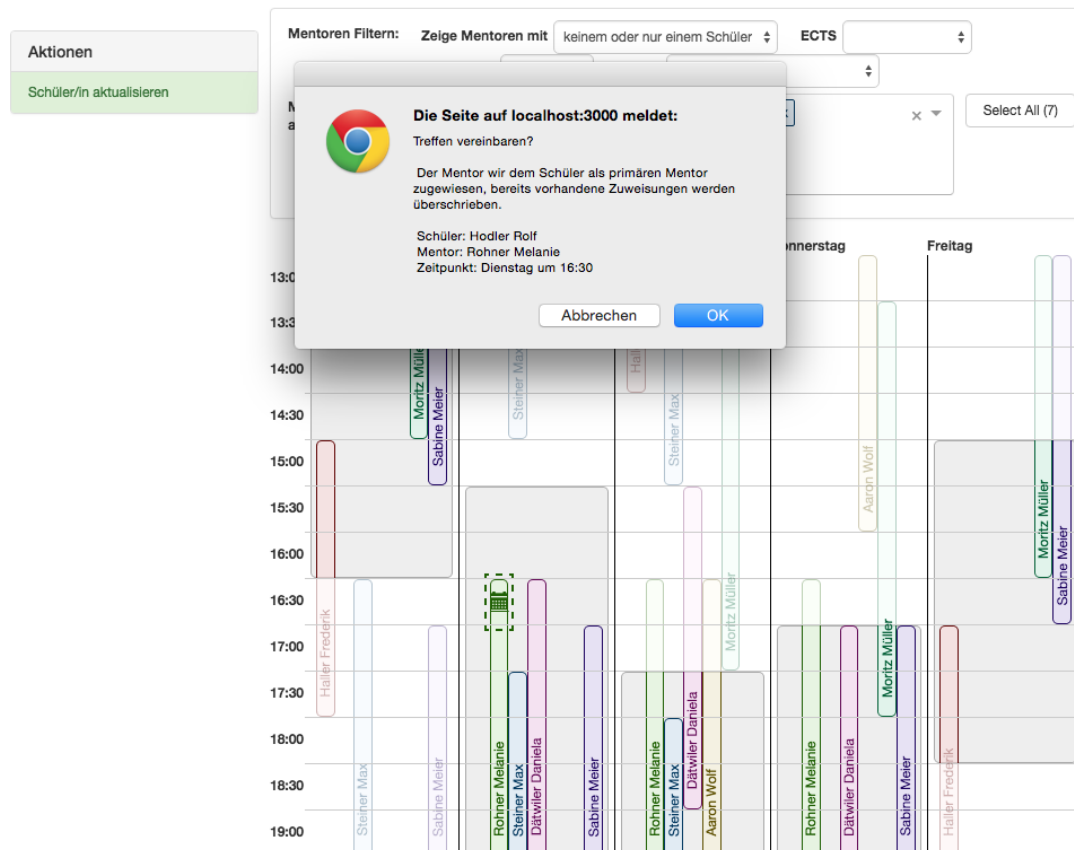


Abbildung 16: Bei einem Klick auf einen Zeitfenster kann der Mentor dem Schüler direkt zugewiesen werden. Es sind nur diejenigen Zeitfenster klickbar, welche sowohl von Schüler als auch von Mentor belegt sind.

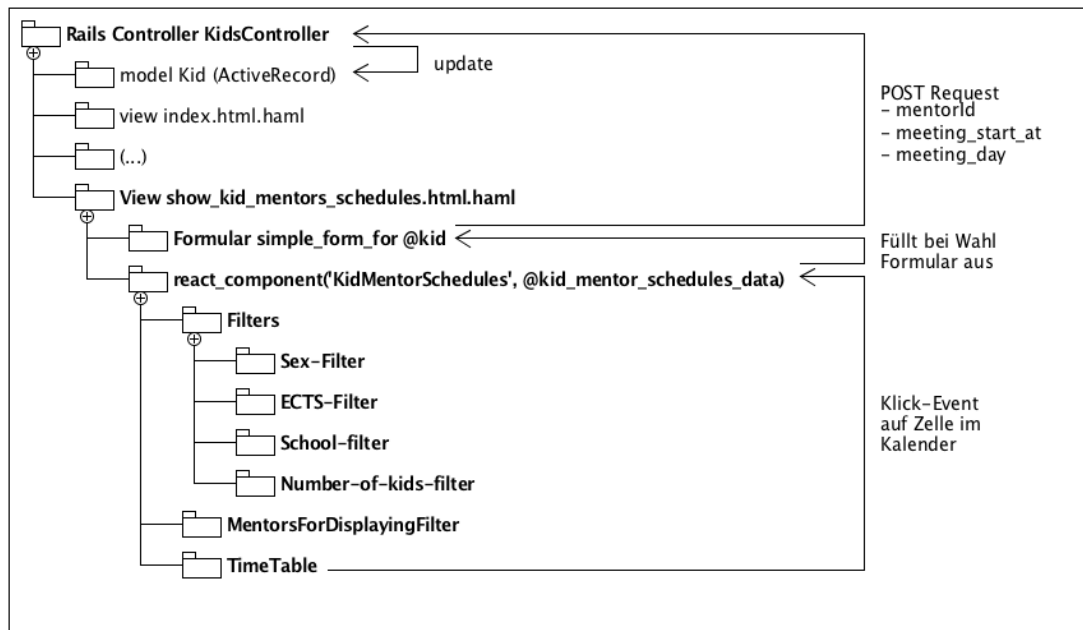


Abbildung 17: Struktur des Bildschirms für die Mentorensuche, ausgehend vom aufgerufenen *Kids-Controller*, dargestellt als "Querschnitt" durch die Anwendung in Paket-Struktur. Die einzelnen Pakete zeigen dabei eine Schicht oder Komponente der Anwendung

**6.2.1.1 Anpassungen am Controller** Der *KidsController*, welcher der Komponenten zu Grunde liegt, wurde um eine Methode *show\_kid\_mentors\_schedules* ergänzt (siehe Listing 5), welcher alle benötigten Daten ausliefert. Da diese Daten in der gerenderte HTML-Seite mitgeliefert werden und somit zur Ladezeit der Seite beiträgt, wurden die *Model*-Daten des Kindes und der Mentoren stark reduziert, damit nur die nötigen Daten gesendet werden.

```

1 def show_kid_mentors_schedules
2   @kid_mentor_schedules_data = Jbuilder.new do |json|
3     json.mentors do
4       Mentor.active.each do |mentor|
5         json.set! mentor.id do
6           json.id mentor.id
7           json.prenome mentor.prenome
8           json.name mentor.name
9           json.sex mentor.sex
10          json.ects mentor.ects
11          json.kids mentor.kids, :id
12          json.secondary_kids mentor.secondary_kids, :id
13          json.primary_kids_school do
14            if not mentor.primary_kids_school.nil?
15              json.id mentor.primary_kids_school.id
16            end
17          end
18          json.schedules create_schedules_nested_set mentor.schedules
19        end
20      end
21    end
22    json.kid do
23      json.id @kid.id
24      json.prenome @kid.prenome
25      json.name @kid.name

```



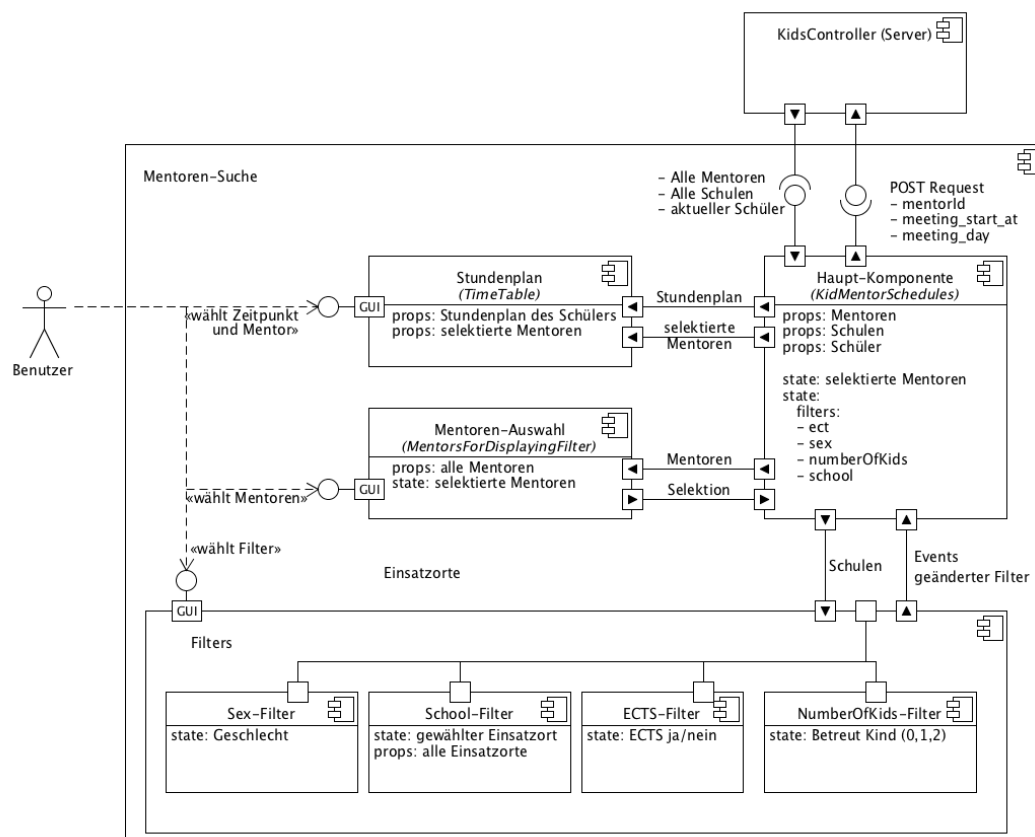


Abbildung 18: Innere Struktur der umgesetzten Mentoren-Suche-Komponente. Gegenüber dem Konzept wurde der Aufbau etwas vereinfacht.

```

26     json.mentor_id @kid.mentor_id
27     json.secondary_mentor_id @kid.secondary_mentor_id
28     json.schedules create_schedules_nested_set @kid.schedules
29   end
30   json.schools School.all, :id, :display_name
31   end.attributes!
32 end

```

Listing 5: Hinzugefügte Methode auf dem KidsController. Sie liefert die Daten für die Komponente aus

**6.2.1.2 Komponente *MentorsForDisplayingFilter*** Die Mentoren, welche im Stundenplan angezeigt werden, können über diese Komponente *MentorsForDisplayingFilter* ausgewählt werden (Siehe Abbildung 19). Die Komponente wurde mittels *react-select* umgesetzt.<sup>21</sup>

**6.2.1.3 Komponente *Filters*** Die Hauptkomponente hält den *State* aller Filter, welche in einer Komponente *Filters* zusammengefasst sind. Sie meldet Änderungen ihrer Filter über *Events* von den untergeordneten Filter-Komponenten an die Hauptkomponente. Listing 6 zeigt ein Beispiel, wie die Komponente alle nicht passenden Filter entfernt, sofern der Filter gesetzt ist.

```

1  getFilteredMentors: ->
2    mentors = @props.mentors
3    filteredMentors = _.clone @props.mentors
4    index = 0
5    total = _.size @props.mentors
6    for id, mentor of filteredMentors
7      mentor.colors = @getColorsOfMentor total, index
8      index++
9      if @state.filters?.sex?
10        delete filteredMentors[id] if mentor.sex isnt @state.filters?.
            sex
11      (...)
12    return filteredMentors

```

Listing 6: Beispiel für das herausfiltern mit nicht-passendem Geschlecht

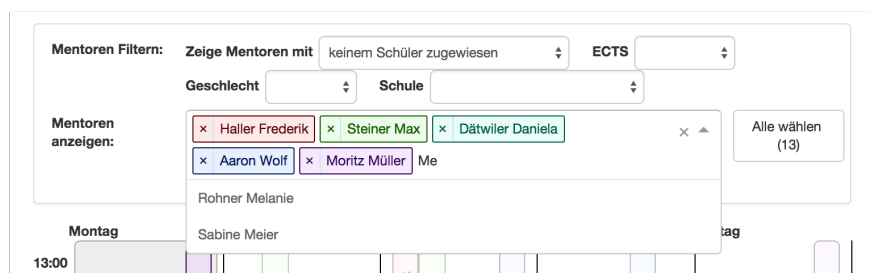


Abbildung 19: Filter-Komponente mit Mentorenauswahl. Die Mentorenauswahl wurde mit *react-select* umgesetzt und ermöglicht eine Namenssuche

**6.2.1.4 Färben der Mentoren** Jeder Mentor soll eine Farbe erhalten, damit leichter erkennbar ist, welche Spalte zu welchem Mentor gehört. Im HSL-Farbraum kann sehr leicht über den *Hue*-Wert, der Winkel im Farbkreis, rotiert werden, bei gleichbleibender Helligkeit (*lightness*) und Sättigung (*saturation*).

<sup>21</sup> Siehe ("Github - React-Select")

Möchte man beispielsweise jeder Index-Zahl von 0 bis 4 eine Farbe zuweisen, kann man den Farbkreis in 5 Segmente unterteilen: 0°, 72°, 144°, 216°, 288° und jeder Zahl eine dieser Farben zuweisen:

- 0 → 0°
- 1 → 72°
- 2 → 144°
- 3 → 216°
- 4 → 288°

Allgemeine Formel:

$$H = index \cdot \frac{360^\circ}{totalNumber}$$

Erhöht man die Anzahl der Segmente, so sind benachbarte Farben jedoch sehr ähnlich und damit auch benachbarte Index-Zahlen. Dies kann man umgehen, indem man jeweils eine Farbe "überspringt":

- 0 → 0°
- 1 → 144°
- 2 → 288°
- 3 → 72°
- 4 → 216°

Listing 7 zeigt die Umsetzung dieser Farbzusweisung. Die Fallunterscheidung von gerader und ungerader Totalanzahl ist nötig, da man beim Multiplizieren mit 2 in einem Restklassenring mit gerader Zahl nur die Hälfte der Elemente erreicht.

```
1 getColorsOfMentor: (total, index) ->
2   indexShifted = (index) ->
3     totalEven = total % 2 is 0
4     offset = if totalEven and index/total >= 0.5 then 1 else 0
5     (index*2+offset) % total
6
7   angle = 360 / total
8   hue = angle * indexShifted index
9   background: "hsla({hue}, 70%, 90%, 0.5)"
10  text: "hsl({hue}, 90%, 20%)"
```

Listing 7: Diese Funktion weist jedem Mentor eine Farbe zu

### 6.2.2 Verifizierung

Die einzelnen Anforderungen wurden durch Akzeptanztests mittels *Capbara-Specs* getrieben erstellt. Abbildung 20 zeigt das Testprotokoll (aus *RubyMine*) der Akzeptanztests.

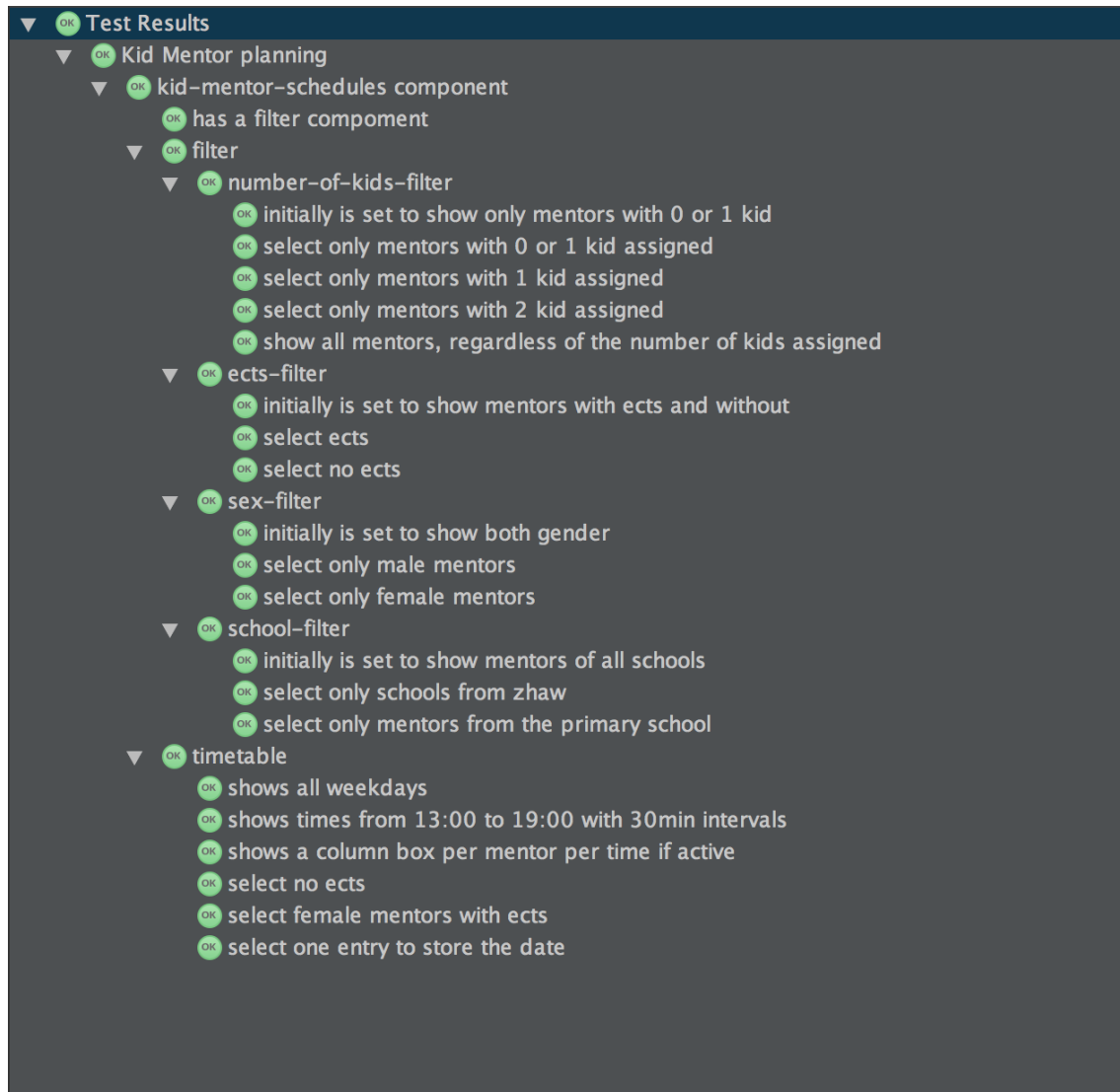


Abbildung 20: Ausgabe der Testsuite von *RubyMine* nach der ersten Iteration

Die Anforderungen, welche die Anwendung vor der Anpassung bereits erfüllten, wurden allerdings nicht erneut maschinell überprüft.

- **SC-001 v1** ✓ durch die bisherige Lösung erfüllt
- **SC-002 v1** ✓ durch die bisherige Lösung erfüllt
- **SC-003 v1** ✓ durch die bisherige Lösung erfüllt
- **SC-004 v1** ✓ durch die bisherige Lösung erfüllt
- **SC-005 v1** ✓ durch die bisherige Lösung und zudem durch die neue Mentoren-Suche erfüllt, welche im Test **timetable - select one entry to store the date** getestet wurde.
- **SC-006 v1** teilweise durch bisherige Lösung erfüllt, aus der Mentorensuche aus können aber nicht direkt Ersatzmentoren zugewiesen werden.
- **SC-007 v1** ✓ durch die bisherige Profilseite des Schülers erfüllt

- **SC-008 v1** ✓ grafisch über die neue Mentoren-Suche umgesetzt
- **SC-009 v1** ✓ grafisch über die neue Mentoren-Suche umgesetzt
- **SC-010 v1** ✓ als Filter umgesetzt, im Test **filters - sex-filter** getestet
- **SC-011 v1** ✓ als Filter umgesetzt, im Test **filters - ects-filter** getestet
- **SC-012 v1** ✓ als Filter umgesetzt, im Test **filters - school-filter** getestet
- **SC-013 v1** ✓ als Filter umgesetzt, im Test **filter - number-of-kids-filter** getestet
- **SC-014 v1** ✓ in dieser Iteration noch nicht maschinell getestet
- **SC-015 v1** ✓ im Test **timetable - select one entry to store the date** verifiziert

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

### 6.2.3 Verifizierung der NFR

Die Nicht-Funktionalen Anforderungen wurden manuell verifiziert:

- **NFR-01** ✓ Erfüllt durch die Trennung von Mentorensuche und Schülerstundenplan
- **NFR-02** ✓ Es sind keine doppelten oder inaktive Schaltflächen mehr vorhanden in den bearbeiteten Komponenten
- **NFR-03** teilweise erfüllt, bei vielen Mentoren sind allerdings die Spalten zu klein
- **NFR-04** ✓ ein gefundener Mentor kann mit zwei Klicks zugewiesen werden (einer für die Bestätigung)
- **NFR-05** teilweise erfüllt durch die Filter, jedoch fehlt der Name des aktuellen Kindes und bereits zugewiesener Mentoren

### 6.2.4 Definition of Done:

- **Funktionalität wurde implementiert** ✓
- **Funktionalität wurde getestet** Teilweise, es fehlen noch wenige Tests (Anzahl-Schüler-Filter)
- **Funktionalität wurde auf eine Testumgebung geladen und die Integration überprüft** ✓
- **Funktionalität und Projekt kann unter einer Open Source Lizenz veröffentlicht werden und Umsetzung beachtet Lizenzen verwendeter Dritt-Software** ✓ Die Verwendeten Libraries (React, React-Select, gems) wurden überprüft
- **Funktionalität wurde auf unterstützten Browsern überprüft (Firefox aktuell, Chrome aktuell, IE ab 10)** ✓
- **Gestaltung der Funktionalität entspricht der Corporate-ID von AOZ** ✓, auf Staging erscheint allerdings ein anderes Logo
- **Funktionalität und Deployment ist dokumentiert** ✓, Deployment benötigt keine Änderung

### 6.3 Zweite Iteration

Die erste Rückmeldung fiel sehr positiv aus. In einem weiteren Treffen wurden Anpassungen besprochen. Grösstes Problem war der Filter für die Anzahl der betreuten Kinder, gemäss **SC-013 v1**. In der Praxis ist es wichtiger zu unterscheiden, ob ein Mentor als Haupt- oder primärer Mentor ein Kind betreut oder nur Ersatzmentor ist. Zudem soll beim Zuweisen eines Mentors zu einem Kind dieser als primärer oder als Ersatzmentor zugewiesen werden, je nach dem, ob der Schüler schon ein Mentor hat oder nicht.

Zudem wurden gewisse Beschriftungen bemängelt, die angepasst wurden.

Weiterhin wurde die Anzahl an Mentoren für den Vergleich limitiert auf 10 Mentoren, sodass die Balken nicht zu klein werden.

#### 6.3.1 Neue Anforderungen

Die folgenden Stories wurden angepasst:

<b>SC-013 v2</b>	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche als primärer Mentor oder Ersatzmentor im Einsatz sind
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Es können Mentoren gefunden werden, welche noch kein Kind zugewiesen haben <input type="checkbox"/> Es können Mentoren gefunden werden, welche nur als primäre Mentoren aktiv sind <input type="checkbox"/> Es können Mentoren gefunden werden, welche nur als Ersatzmentoren aktiv sind <input type="checkbox"/> Es können Mentoren gefunden werden, sowohl als primärer Mentor als auch als Ersatzmentor aktiv sind
<i>Priorität</i>	MAY

<b>SC-010 v2</b>	Als Administrator möchte ich für einen Schüler nach Mentoren mit passendem Geschlecht suchen können.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Neu: Die Optionen sind mit 'männlich' und 'weiblich' beschriftet
<i>Priorität</i>	MAY

<b>SC-011 v2</b>	Als Administrator möchte ich für einen Schüler nach Mentoren suchen können, welche mittels ECTS-Punkten vergütet werden.
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Neu: Die Optionen sind mit 'ECTS' und 'Kein ECTS' beschriftet
<i>Priorität</i>	MAY

<b>SC-008 v2</b>	Als Administrator möchte ich für einen Schüler nach Mentoren mit überschneidenden Zeitfenstern suchen können
<i>Akzeptanzkriterien:</i>	<input type="checkbox"/> Neu: Es werden maximal 10 Mentoren für den Vergleich angezeigt
<i>Priorität</i>	MAY

Diese Story wurde als Massnahme für die Reduktion der nötigen Klicks implizit umgesetzt, aber noch nicht als Story erfasst. Da diese Anforderung nun etwas angepasst wurde, soll sie hier eine eigene Story erhalten:

<b>SC-016 v1</b>	Als Administrator möchte ich in der Mentoren-Suche ein Mentor direkt einem Kind zuweisen können
<b>Akzeptanzkriterien:</b>	<input type="checkbox"/> Die Wahl muss bestätigt werden <input type="checkbox"/> Der Mentor wird als Hauptmentor zugewiesen, wenn das Schüler noch keinen Mentor hat <input type="checkbox"/> Der Mentor wird als Ersatzmentor zugewiesen, wenn der Schüler bereits einen Mentor hat <input type="checkbox"/> Es erscheint eine Meldung, wenn sowohl primärer Mentor, als auch Ersatzmentor zugewiesen sind <input type="checkbox"/> Angewählter Wochentag wird persistiert <input type="checkbox"/> Angewählter Zeitpunkt wird persistiert
<b>Priorität</b>	MAY

### 6.3.2 Umsetzung und Verifizierung

Anwendung und Tests wurden angepasst und erweitert, das resultierende Testprotokoll aus *RubyMine* ist in Abbildung 23 ersichtlich.

- **SC-013 v2** ✓ im Test **filter - mentors-display-filter** verifiziert
- **SC-010 v2**, **SC-011 v2** Die Anpassung der Beschriftungen wurden nicht maschinell überprüft
- **SC-008 v2** ✓ in den angepassten Tests **timetable - shows maximum 10 mentors** verifiziert
- **SC-016 v1** ✓ in den angepassten Tests **timetable - selection of mentors** verifiziert

### 6.3.3 Verifizierung der NFR (Ergänzungen)

- **NFR-03** ✓ nun ebenfalls erfüllt durch das Limit von 10 maximal angezeigter Mentoren
- **NFR-05** ✓ erfüllt durch das Formular das jetzt angezeigt, sowie durch die Anzeige des Namens des Kindes.

### 6.3.4 Definition of Done

- **Funktionalität wurde implementiert** ✓
- **Funktionalität wurde getestet** ✓ die neuen Funktionalitäten wurden getestet, es wurden fehlende Tests nachgereicht.
- **Funktionalität wurde auf eine Testumgebung geladen und die Integration überprüft** ✓
- **Funktionalität und Projekt kann unter einer Open Source Lizenz veröffentlicht werden und Umsetzung beachtet Lizenzen verwendeter Dritt-Software** ✓
- **Funktionalität wurde auf unterstützten Browsern überprüft (Firefox aktuell, Chrome aktuell, IE ab 10)** ✓
- **Gestaltung der Funktionalität entspricht der Corporate-ID von AOZ** ✓
- **Funktionalität und Deployment ist dokumentiert** ✓

**Aktionen**

Schüler/in aktualisieren

Seitenanfang

## Hodler Rolf

**Mentor/in**

Rohner Melanie

**Ersatz Mentor/in**

**Treffen Wochentag**

Dienstag

**Treffen beginnt um**

2000-01-01 16:30:00 UTC

**Mentoren Filtern:** Zeige Mentoren mit keinem primären Schüler zugewiesen nur primärem Schüler zugewiesen nur sekundärem Schüler primärem und sekundärem Schüler ECTS 5

**Geschlecht** männlich

**Mentoren anzeigen:** x Haller Frederik x Steiner Max x Dätwiler Daniela x Moritz Müller x Sabine Meier Select All (5)

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
13:00					
13:30					
14:00					
14:30					
15:00					
15:30					
16:00					
16:30					
17:00					
17:30					
18:00					
18:30					
19:00					

Abbildung 21: Die Auswahl für den Mentoren zur Anzahl der betreuten Kinder, wurde angepasst. Zudem wurde das zuvor nicht sichtbare Formular, welches der Seite zu Grunde liegt, sichtbar gemacht, da es nützliche Informationen offenbart. Zudem kann dadurch der Mentor manuell geändert werden.



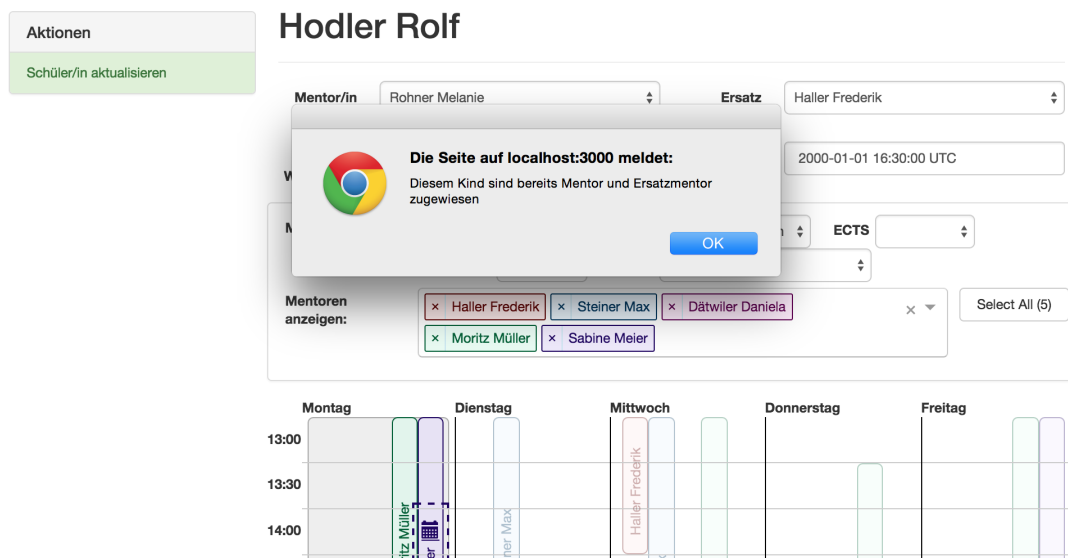


Abbildung 22: Sind schon beide Mentoren zugewiesen, erscheint nun eine Fehlermeldung

Abbildung 23: Ausgabe der Testsuite von *RubyMine* nach der zweiten Iteration

## 7 Testing

Die umgesetzte Arbeit wurde automatisiert mittels *Specs* und durch den Auftraggeber verifiziert. Nach den beiden im vorherigen Abschnitt erläuterten Iterationen ergibt sich folgende Zusammenfassung:

### 7.1 User-Stories

Story	Implementiert	automatisiert getestet	durch Kunden verifiziert	Beschreibung
SC-001 v1	✓	×	✓	durch die bisherige Lösung erfüllt
SC-002 v1	✓	×	✓	durch die bisherige Lösung erfüllt
SC-003 v1	✓	×	✓	durch die bisherige Lösung erfüllt
SC-004 v1	✓	×	✓	durch die bisherige Lösung erfüllt
SC-005 v1	✓	✓	✓	durch die bisherige Lösung erfüllt zudem durch die neue Mentoren-Suche erfüllt (siehe SC-016 v1)
SC-006 v1	✓	✓	✓	durch die bisherige Lösung erfüllt zudem durch die neue Mentoren-Suche erfüllt (siehe SC-016 v1)
SC-007 v1	✓	×	✓	durch die bisherige Profilseite des Schülers erfüllt
SC-008 v2	✓	Teilweise	✓	grafisch über die neue Mentoren-Suche umgesetzt
SC-009 v1	✓	Teilweise	✓	grafisch über die neue Mentoren-Suche umgesetzt
SC-010 v2	✓	✓	✓	als Filter umgesetzt
SC-011 v2	✓	✓	✓	als Filter umgesetzt
SC-012 v1	✓	✓	✓	als Filter umgesetzt
SC-013 v2	✓	✓	✓	als Filter umgesetzt
SC-014 v1	✓	✓	✓	als Filter umgesetzt
SC-015 v1	✓	✓	✓	durch bisherige Lösung erfüllt zudem über die neue Mentoren-Suche (siehe SC-016 v1)
SC-016 v1	✓	✓	✓	Direktzuweisung von Mentoren zu Schülern aus der Mentoren-Suche inkl. Wochentag und Zeitpunkt

## 7.2 Nicht-Funktionale Anforderungen

NFR	automatisiert getestet	durch Kunden verifiziert	Beschreibung
NFR-01	×	✓	Erfüllt durch die Trennung von Mentorensuche und Schülerstundenplan
NFR-02	×	✓	Es sind keine doppelten oder inaktive Schaltflächen mehr vorhanden in den bearbeiteten Komponenten
NFR-03	×	✓	Es werden maximal 10 Mentoren angezeigt
NFR-04	×	✓	ein gefundener Mentor kann mit zwei Klicks zugewiesen werden (einer für die Bestätigung)
NFR-05	×	✓	Alle nötigen Informationen werden angezeigt.

## 7.3 Definition of Done

- Funktionalität wurde implementiert ✓
- Funktionalität wurde getestet ✓
- Funktionalität wurde auf eine Testumgebung geladen und die Integration überprüft ✓
- Funktionalität und Projekt kann unter einer Open Source Lizenz veröffentlicht werden und Umsetzung beachtet Lizenzen verwendeter Dritt-Software ✓
- Funktionalität wurde auf unterstützten Browsern überprüft (Firefox aktuell, Chrome aktuell, IE ab 10) ✓
- Gestaltung der Funktionalität entspricht der Corporate-ID von AOZ ✓ ,
- Funktionalität und Deployment ist dokumentiert ✓

## 7.4 Kundenfeedback

### Feedback: Einführung Kalenderfunktion auf der Onlineplattform von Future Kids

Die Zusammenarbeit mit Marco war sehr angenehm und produktiv. Die gemeinsame Ideenentwicklung hat zur erfolgreichen Umsetzung der Kalenderfunktion geführt. Bei einem ersten Treffen konnte von Seiten Future Kids die Bedürfnisse der neuen Funktion angebracht werden. Durch gezieltes Fragen machte sich Marco ein genaues Bild der gewünschten Möglichkeiten der neuen Kalenderfunktion und bot konkrete Vorschläge zu deren Umsetzung.

Die Umsetzung entsprach ganz der besprochenen Idee. Kleine Anpassungswünsche wurden sofort umgesetzt. Die neue Kalenderfunktion ist sehr übersichtlich und benutzerfreundlich gestaltet. Sie entspricht den Nutzungsanforderungen von Future Kids. Durch die neue Funktion wird die Arbeit im Zusammenhang mit dem bilden von Mentor/in-Schüler/in Paaren stark erleichtert, wodurch viel Zeit eingespart werden kann. Mit dem Ergebnis sind wir sehr zufrieden.

- Livia Kernen, AOZ "Future Kids"

## 8 Rückblick

Mit stetiger Sicht auf die funktionalen und nicht-funktionalen Anforderungen konnte ich die bestehende Anwendung Schritt für Schritt verbessern. Aus den Anforderungen wurden Tests, mit der Schrittweisen Erfüllung der Tests entstand die Anwendung.

Die Arbeit mit React erwies sich dabei als sehr intuitiv, sofern man begriffen hat, welche (Unter-)Komponente einen *State besitzen* soll und wie *Events* durch die einzelnen Komponenten kommuniziert werden. Vorteilhaft ist dabei das sehr beschränkte und dadurch Simple Schnittstellendesign der React-Komponenten: Es gibt nicht viele Möglichkeiten etwas umzusetzen; ein falscher Aufbau der Komponenten oder falsche Platzierung des *States* macht sich sehr schnell durch fehlerhaftes Verhalten bemerkbar. Durch meine vielen Erfahrungen im Bereich Frontend-Entwicklung mit HTML und Javascript kam ich sehr schnell mit React zurecht, was auch ein Verdient der Einfachheit des Frameworks ist.

Auch die Integration von React in Ruby on Rails erwies sich als einfach; allerdings ist die Komponente in sich abgeschlossen und benötigt keine Live-Daten, d.h. Daten, die sich während der Benutzung der Komponente ändern können. Dadurch sind die Schnittstellen zwischen React und Rails sehr einfach.

### 8.1 Schwierigkeiten

Anforderungen in Form von Stories müssen klar notiert und mit Akzeptanzkriterien versehen werden, um sie einerseits klar kommunizieren und um sie danach auch verifizieren zu können in Form von Tests. Ein grosser Teil der Arbeit bestand jedoch nicht daraus, neue Funktionalitäten hinzuzufügen, sondern die vorhanden zu verbessern. Um trotzdem die Arbeit mit sinnvoller Granularität in einzelne Stories zu unterteilen, betrachte ich die gewünschten Funktionalitäten so, als wären sie noch gar nicht umgesetzt.

Dies ergab nun zwar gut differenzierbare Stories, die Akzeptanzkriterien für diese Stories zu erfassen, war allerdings nicht leicht, da auch die bestehende Lösung viele der Stories bereits erfüllte. Die Anforderungen an die Bedienbarkeit hatte ich daher in Form von Nicht-Funktionalen Anforderungen notiert. Dadurch waren sie allerdings schwierig zu verifizieren in Form von Tests. Rückblickend wäre es daher besser gewesen, diese als zusätzliche Akzeptanzkriterien den Stories hinzuzufügen.

Die hohen Anforderungen an die Bedienbarkeit ergaben zudem einen verhältnismässig komplexen Aufbau der Komponente, was das Testen zusätzlich erschwerte. In einem solchen Fall muss die hohe Komplexität gegen die damit verbundenen komplexeren Tests abgewogen werden. Ich entschied mich trotzdem dazu, mein erarbeitetes Konzept wie geplant umzusetzen und nahm dafür einige etwas weniger genaue Tests in Kauf, da ich der Meinung bin, im vorliegenden Fall überwiegt der Nutzen dem möglichen Risiko für eine Fehlfunktion oder der Weiterentwicklung.

## 9 Ausblick

Zum Zeitpunkt des Schreibens dieser Arbeit steht der Administration die aufwändige Aufgabe des “Matchings”, d.h. des Findens passender Mentor- und Ersatzmentor-Paare für die Schüler des kommenden Semesters bevor. Die entwickelte Komponente ist für diese Aufgabe essentiell und die Verbesserungen sollten sich gemäss Administration in einer signifikanten Zetersparnis niederschlagen. Da alle wichtigen Anforderungen umgesetzt und verifiziert wurden, steht einer Live-Schaltung auf die Produktiv-Umgebung nichts im Wege. Diese soll Anfangs August 2015 stattfinden.

Der Kunde hat zudem weitere mögliche Erweiterungen bereits erwähnt:

- Falls ein Mentor bereits ein anderes Kind betreut, wird eine Meldung angezeigt, falls der Mentor gewählt wird.
- Es soll möglich sein, nur einen einzelnen Wochentag anzuzeigen

Diese wurden als *Issues* auf dem *Github-Repository* von FutureKids<sup>22</sup> erfasst. Da dieses öffentlich ist, kann jeder einen *PullRequest* einsenden um diese Anforderungen umzusetzen. FutureKids hofft auf weitere Freiwillige für diese Arbeit.

---

<sup>22</sup>Siehe (“Github FutureKids”)

## A Anhang

### A.1 Quellcode

Quellcode der Arbeit:

[https://github.com/panterch/future\\_kids/](https://github.com/panterch/future_kids/)

### Quellenangaben

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

“ECTS - Wikipedia.” [http://de.wikipedia.org/wiki/European\\_Credit\\_Transfer\\_System](http://de.wikipedia.org/wiki/European_Credit_Transfer_System).

“FET - Free Timetabling Software.” <http://lalescu.ro/liviu/fet/>.

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

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

“Github FutureKids.” [https://github.com/panterch/future\\_kids/](https://github.com/panterch/future_kids/).

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

“JBoss Optaplanner.” <http://www.optaplanner.org/>.

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

“Kenneth S. Rubin - ESSENTIAL SCRUM - A Practical Guide to the Most Popular Agile Process.”

“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](https://de.wikipedia.org/wiki/Representational_State_Transfer).