

# Gesamtdokumentation

Thema: Erstellen von Lernspielen für Kinder der 1. Klasse der  
Grundschule

Fach: Web-Development II

Dozent: Prof. Dr. Jürgen Heym

Teammitglieder: August Kraft, Jan Mothes, Christian Pöhlmann, Philipp Marek

# Inhaltverzeichnis

Inhaltverzeichnis.....	2
1 Spezifikation.....	3
1.1 Einleitung .....	3
1.2 Webanwendung .....	3
1.2.1 Anwendergruppe: Schüler .....	3
1.2.2 Anwendergruppe: Eltern .....	3
1.2.3 Anwendergruppe: Lehrer .....	3
2 Projektmanagementdokumentation .....	4
2.1 Ist-Zustand .....	4
2.2 Soll-Analyse .....	4
2.3 Rahmenbedingungen .....	4
2.4 Aufgabenstellung.....	4
2.5 Gewichtung von PHP-Frameworks .....	4
2.5.1 Kriterien zum Vergleich von Frameworks .....	4
2.5.2 Gewichtung der Kriterien .....	5
2.6 Beschreibung der PHP-Frameworks .....	5
2.6.1 Yii .....	5
2.6.2 Zend Framework 2 .....	6
2.6.3 Sonstiges .....	7
2.6.3 Symfony 3 .....	7
2.6.4 Laravel 5.....	8
2.6.5 Nette.....	8
2.6.6 PHPixie.....	8
2.7 Gegenüberstellung der PHP-Frameworks .....	9
2.8 Bewertung von Projektmanagement-Methoden .....	10
2.9 Beschreibung der Projektmanagementmethoden.....	11
2.9.1 Kanban .....	11
2.9.2 Scrum.....	12
2.9.3 Feature-Driven-Development (FDD) .....	13
2.9.4 Extreme Programming (XP) .....	14
2.10 Gegenüberstellung der Projektmanagementmethoden .....	16
2.11 Software-Spezifikation .....	17
2.12 Frameworks .....	17
2.12.1 Bootstrap .....	17
2.12.2 Semantic UI.....	17
2.12.3 Vergleich .....	18
2.13 Ablauf .....	19
2.13 Programm.....	22
2.13.1 Hauptmenü.....	22
2.13.2 Nutzer .....	23
2.13.3 Statistik .....	23
2.14 Testspezifikation .....	24
2.14.1 Testfälle.....	24
2.14.2 Testfälle und deren Ergebnisse .....	27
3 Testprotokolle .....	28
4 Adminhandbuch .....	29
5 Benutzerhandbuch.....	30
6 Quellen.....	31

# 1 Spezifikation

## 1.1 Einleitung

Die Webanwendung hat als Ziel, Schüler der Jahrgangsstufe 1 spielerisch schulisches Wissen ihres Schuljahres zu vermitteln und Eltern und Lehrern einen Einblick in die Fortschritte und Fähigkeiten ihrer Kinder/Schüler zu geben.

## 1.2 Webanwendung

Es soll für die Anwendergruppen „Schüler“, „Eltern“ und „Lehrer“ verschiedene Ansichten geben.

### 1.2.1 Anwendergruppe: Schüler

Für Schüler soll es möglich sein, sich anhand ihres Namens, ohne Passwort, einzuloggen. Nach dem Login muss den Schülern eine Übersicht mit allen verfügbaren Spielen in einer kindgerechten Darstellung angeboten werden. Aus dieser heraus sollen die Kinder dann entsprechend Spiele auswählen können. Die Spiele sind in drei steigende Schwierigkeiten unterteilt werden, die die Kinder nacheinander durcharbeiten. Ihr Fortschritt soll ihnen Anhand einer Art Fortschrittsleiste angezeigt werden.

### 1.2.2 Anwendergruppe: Eltern

Eltern loggen sich mit Hilfe des Nutzernamens ihres Kindes und eines zusätzlichen Passwortes ein. Die Eltern können sich die Statistik ihres eigenen Kindes ansehen und von da aus in die entsprechende Aufgabe springen, um das Ergebnis nachzuvollziehen.

### 1.2.3 Anwendergruppe: Lehrer

Die Gruppe Lehrer zählt zusätzlich noch als Administratorengruppe. Für sie muss es möglich sein, alle Nutzer zu verwalten, also Nutzer zu erstellen, zu löschen, zu verändern und anzuzeigen. Außerdem können sie die Statistiken aller Schüler einsehen und von dort aus in die einzelnen Aufgaben springen.

## 2 Projektmanagementdokumentation

Im Folgenden wird das Projekt und das Management des Projektes aufgezeigt. Zuerst der Ist-Zustand des vorhandenen Projektes und eine Soll-Analyse des zukünftigen Programmes. Danach eine kurze Erklärung der Rahmenbedingungen und die eigentliche Aufgabenstellung. Nach dem grundsätzlichen Punkten kommen jeweils die Auswahl eines PHP-Frameworks, einer Projektmanagementmethodik und eines HTML/JavaScript/CSS-Frameworks. Nach der Auswahl der verschiedenen Tools und Methoden kommt der Ablauf und der Aufbau des Programmes und eine Testspezifikation.

### 2.1 Ist-Zustand

Im Moment ist kein vergleichbares Projekt vorhanden, das Programm ist also im aktuellen Kontext (Web-Development 2, Gruppe 17) eine Neuentwicklung.

### 2.2 Soll-Analyse

Im Zuge des Projektes soll eine Webseite mit funktionsfähigen Spielen entwickelt werden. Zu diesen Spielen sollen eine Anmeldung und eine Übersicht vorhanden sein.

### 2.3 Rahmenbedingungen

Die Zielgruppe dieser Spiele sind Kinder der ersten Klasse. Die Webseite soll über eine Übersicht mit mehreren Spielen verfügen. Diese Spiele sollen einen leicht ansteigenden Schwierigkeitsgrad haben. Die Webseite soll über eine rudimentäre Anmeldung verfügen, bei der sich die Kinder ohne Passwort anmelden können. Die Zielgruppe der Statistik sind vor allem Lehrer aber auch Eltern. Sie soll Informationen über den Spielfortschritt der Kinder enthalten.

### 2.4 Aufgabenstellung

Die Aufgabe in diesem Projekt ist vor allem eine lauffähige Sammlung von Spielen, die über eine Weboberfläche aufrufbar sind, zu implementieren. Diese Aufgaben sollen Mathematikaufgaben für Kinder der 1. Klasse der Grundschule sein. Dabei ist eine mögliche Aufgabe die Addition und die Subtraktion von Zahlenraum bis 10, bis 20 und bis 100.

### 2.5 Gewichtung von PHP-Frameworks

Im Folgenden werden zur Beurteilung einiger PHP-Frameworks die verwendeten Kriterien und deren Gewichtung erläutert.

#### 2.5.1 Kriterien zum Vergleich von Frameworks

Um Frameworks bewerten zu können, müssen zuerst Kriterien gefunden werden, nach denen diese auch verglichen werden können. Auch sollte man eine mögliche unterschiedliche Gewichtung verschiedener Kriterien berücksichtigen. Folgende Kriterien werden von uns verwendet:<sup>1</sup>

**Aktualität:** Wird das Framework aktiv gepflegt und entwickelt? Gut ist wenn das Framework erst vor kurzem und in regelmäßigen Abständen erweitert/ausgebessert wurde. Schlecht ist wenn das Framework entweder nur unregelmäßig oder vor langer Zeit bearbeitet wurde.

---

<sup>1</sup> <http://www.thewebhatesme.com/allgemein/10-auswahlkriterien-fur-php-frameworks/>

**Verbreitung:** Ruf des Frameworks bzw. Referenzen? Ein gutes Framework ist in diesem Fall ein Framework das von bekannten Unternehmen und/oder vielen Personen verwendet wird.

**Dokumentation:** Wie ausführlich ist die Dokumentation? Gibt es Anleitungen oder Tutorials? Im besten Falle hat das Framework von den Entwicklern selbst ein digitales, übersichtliches Tutorial mit Code-Beispielen.

**Community:** Wie schnell bekommt man hochwertige Hilfe auf Fragen? Gut ist, wenn eine große Community mit vielen Fragen/Antworten und einer schnellen Antwortzeit hinter dem Framework steht. Als Bewertungsgrundlage dienen entweder <http://www.stackoverflow.com>, oder das entsprechende eigene Forum.

**Einarbeitungszeit:** Wie aufwändig ist es sich in das Framework einzuarbeiten? Ein gutes Framework hat eine geringe Einarbeitungszeit. Als Basis dienen unterschiedliche Nutzeraussagen aus verschiedenen Foren.

**Features:** Welche Features werden bereitgestellt? Lassen sich externe Komponente hinzufügen, um die fehlenden Features abzudecken? Gut ist es wenn das Framework viele Features hat. Als Bewertungsbasis wird [https://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_frameworks](https://en.wikipedia.org/wiki/Comparison_of_web_frameworks) verwendet.

**Preis:** Kostet das Framework etwas oder nicht? Das Produkt muss kostenlos sein.

**PHP5:** Unterstützt das Framework PHP5? Das Framework muss PHP5 unterstützen.

## 2.5.2 Gewichtung der Kriterien

Es werden für diverse Frameworks alle Kriterien angeschaut, Schwerpunkt liegt jedoch auf Dokumentation und Community. Als Ausschlusskriterien zählen PHP5 und Preis. Es ist uns also wichtig, dass das Framework gut dokumentiert ist und eine aktiver Community hinter sich hat, es ist aber obligatorisch, dass es PHP5 unterstützt und kostenlos ist. Da für uns die Dokumentation sehr wichtig ist wird sie mit einer Gewichtung von 4 versehen, der höchsten Gewichtung der benutzten Kriterien. Fast genauso wichtig ist für uns die Community die hinter dem Framework steht, weswegen dieses Kriterium eine Gewichtung von 3 erhält. Die Kriterien Aktualität, Verbreitung, Einarbeitungszeit und Features sind in etwa gleich wichtig, weswegen sie eine Gewichtung von 2 erhalten. Die verbleibenden Kriterien Preis und PHP5 sind Ausschlusskriterien und bekommen daher nur eine Gewichtung von 1.

## 2.6 Beschreibung der PHP-Frameworks

Im Folgenden werden insgesamt sechs verschiedene Frameworks analysiert. Dabei werden besonders die Vor- und Nachteile betrachtet.

### 2.6.1 Yii

Yii wurde am 1. Januar 2008 von Qiang Xue ins Leben gerufen. Seitdem hat sich Yii zu einem schnellen, sicheren und professionellen Framework entwickelt, dessen Fokus auf die Erstellung von Web 2.0 Applikationen liegt.<sup>2</sup>

#### 2.6.1.1 Vorteile

Yii ist weit verbreitet. Das zeigen die 5 Millionen Treffer wenn man bei Google nach "Yii" sucht. Dazu kommen die 900 Stellenangebote, die sich im Forum der Yii-Webseite angesammelt haben<sup>3</sup> und den zahlreichen "Yii-powered" Webseiten, die das Framework benutzen.<sup>4</sup> Um sich Kenntnisse über Yii anzueignen stehen einem der ausführliche

---

<sup>2</sup> <http://www.yiiframework.com/about>

<sup>3</sup> <http://www.yiiframework.com/forum/index.php?/forum/15-job-opportunities/>

<sup>4</sup> <http://www.yiiframework.com/forum/index.php/forum/14-yii-powered-applications/>

<sup>5</sup> <http://yii.poweredsites.org/>

Guide auf der Webseite mit Codebeispielen und der Möglichkeit zu kommentieren,<sup>6</sup> Bücher (sowohl in Printform als auch als Ebook) und kostenpflichtige Videokurse zur Verfügung.<sup>7</sup> Das eigene Forum ist aktiv und es gibt Beiträge in mehr als zwei Dutzend Sprachen.<sup>8</sup> Die Mitglieder sind freundlich und hilfsbereit und man bekommt recht schnell eine Antwort auf seine Fragen.<sup>9</sup> Yii setzt ausschließlich auf PHP5 und lässt sich mit Erweiterungen (Extensions) beliebig erweitern. Derzeit gibt über 2000 Erweiterungen.<sup>10</sup> Auch lässt sich Yii mit anderen Komponenten verknüpfen, um das Framework so verwenden zu können, wie gewünscht.<sup>11</sup>

### **2.6.1.2 Nachteile**

Das Selbststudium kann recht teuer werden, die Videokurse liegen preislich über 50€, die Bücher zwischen 30€ und 50€, die jeweils nur einen Anwendungsbereich von Yii abdecken.

### **2.6.1.3 Sonstiges**

Die neueste Version ist 2.0.7 vom 14. Februar 2016 und ist eine vollständige Überarbeitung von Yii auf Basis von PHP 5.4.0.<sup>12</sup> Yii ist Open Source (BSD)<sup>13</sup> und wird von wenigen Entwicklern gepflegt.<sup>14</sup> Das Verhältnis zwischen gemeldeten und gelösten Bugs liegt bei 2,465/2,188.<sup>15</sup> Yii ist nach dem MVC Prinzip entwickelt.<sup>16</sup> Yii wird unter anderem von Stay.com und Fusonic GmbH verwendet.

## **2.6.2 Zend Framework 2**

Zend Framework 2 (nachfolgend Zend 2) entstand aus Zend Framework 1 (nachfolgend Zend 1), welches ein häufig benutztes Framework ist (15 Mio. Downloads). Google gibt bei der Suche nach "Zend 2" 1,4 Mio. Ergebnisse zurück. Zend 2 ist 100% objektorientiert.<sup>17</sup>

### **2.6.2.1 Vorteile**

Das Framework wird vom Entwickler getestet (mittels PHPUnit). Der Übergang von Zend 1 ist gegeben, dazu läuft zusätzlich Zend 1 Code in der Zend 2 Engine.<sup>18</sup> Die Standardbibliothek bietet viele Funktionen u.a. Datenbankenabstraktion, Eingabemasken mit HTML5 Unterstützung (Rendering, Überprüfung) und Komponenten für Authentifizierung und Autorisierung. Insgesamt beinhaltet Zend 2 die am meisten gebrauchten Webservices, deren Verwendung die Entwicklungszeit erheblich verkürzt, und die ausgiebig getestet wurden. Zend 2 ist mit Hilfe der eigenen lebhaften Community entstanden, deren Mitglieder auf verschiedene Wege erreichbar sind (Mail, IRC), um bei Fragen bereitzustehen.<sup>19</sup> Zum Selbststudium stehen einem ein Tutorial, ein Guide und die API Dokumentation zur Verfügung. Dazu gibt es diverse Bücher im Preisbereich von 40€ oder mehr. Man kann sich kostenpflichtig sein Fachwissen nachweisen mittels Zertifikate verschiedener Stufen (ab ca. 1000 Euro). Zend 2 fordert keine Erlernung einer neuen Sprache wie YAML.<sup>20</sup>

---

<sup>6</sup> <http://www.yiiframework.com/doc/guide>

<sup>7</sup> <http://www.yiiframework.com/doc>

<sup>8</sup> <http://www.yiiframework.com/forum/>

<sup>9</sup> <https://web.archive.org/web/20120618111441/http://blog.mbischof.de/auswahlkriterien-fur-php-frameworks>

<sup>10</sup> <http://www.yiiframework.com/extensions/>

<sup>11</sup> <http://www.yiiframework.com/search/?q=integrating&type=wiki>

<sup>12</sup> <http://www.yiiframework.com/download/>

<sup>13</sup> <http://www.yiiframework.com/license/>

<sup>14</sup> <https://github.com/yiisoft>

<sup>15</sup> <https://github.com/yiisoft/yii/issues>

<sup>16</sup> <http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>

<sup>17</sup> <http://framework.zend.com/about/>

<sup>18</sup> <http://framework.zend.com/about/faq/>

<sup>19</sup> <http://framework.zend.com/about/>

<sup>20</sup> <http://www.sheldmandu.com/php/php-mvc-frameworks/yii-vs-zend-vs-code-igniter-compared>

### 2.6.2.2 Nachteile<sup>21</sup>

Das Framework ist recht wortreich, was zu verhältnismäßig viel Code für simple Sachen führt. Es kann langsam sein, weil Klassen auf viele Dateien verteilt sind. Man kann dem jedoch entgegenwirken, wenn man Extraarbeit mit Opcode-Caching betreibt. Leider wird Wissen im Bereich Enterprise-Application Architektur vorausgesetzt bzw. ohne diese ist die Lernkurve recht steil.

### 2.6.3 Sonstiges

Die neueste Version (2.4.9) erschien am 23.11.2015<sup>22</sup> und ist Open Source. Zend 2 basiert auf PHP 5.3 und ist eine überarbeitete Version von Zend 1; u.a. wurde die MVC Schicht vollständig geändert mit Fokus auf Ereignissteuerung.<sup>23</sup> Zend steht in Partnerschaft mit diversen Unternehmen (z.B. Google, Microsoft), die Komponente, Features und Interfaces beigesteuert haben.<sup>24</sup> Bezüglich Bugfixes kann man auf der Github-Projektseite einsehen, dass zu den 1.632 gelösten 425 offene Issues stehen.<sup>25</sup> Zend wird von folgenden Unternehmen verwendet: BBC, Cisco WebEx, Centroy<sup>26</sup>

### 2.6.3 Symfony 3<sup>27</sup>

Symfony<sup>28</sup> ist ebenfalls ein bekanntes und benutztes Framework, welches sogar manchmal die Grundlage für andere Frameworks bildet (z.B. Laravel)<sup>29</sup>. Releases erscheinen bis heute regelmäßig (ca. 3 pro Monat), aktuelle Commits können auf Github<sup>30</sup> verfolgt werden. Symfony 3 wird gesponsert von 'SenioLabs' und von einer Einzelperson betreut und verwaltet. Die meisten Features werden über Githubs Pull-Request-System von beliebigen Entwicklern eingereicht. Symfony wird in größeren Projekten eingesetzt, z.B: Drupal, Spotify, BlaBlaCar, YouPorn.<sup>31</sup> Die Community kommuniziert stark über Stackoverflow, wo 40000 Fragen mit dem Tag 'Symfony' oder 'symfony2' (500 'symfony3') existieren. Es existiert eine ausführliche Dokumentation sowie eine kostenlose Tutorialsammlung namens 'The Symfony Cookbook'<sup>32</sup>.

Symfony steht unter der freien MIT Lizenz, und unterstützt PHP5<sup>33</sup>.

#### 2.6.3.1 Vorteile

Symfony ist als stabiles, robustes, aber vor allem sehr modulares Framework bekannt. Es ist aus Komponenten zusammengesetzt, welche ausgetauscht werden können. Das macht das Framework sehr flexibel und anpassbar.

#### 2.6.3.2 Nachteile

Nicht unbedingt simple oder ausgelegt für Rapid Development. Anpassbarkeit macht einiges an Konfiguration nötig. Die Einarbeitungszeit ist nicht besonders niedrig bzw. angemessen für den Funktionsumfang.

---

<sup>21</sup><http://www.sheldmandu.com/php/php-mvc-frameworks/yii-vs-zend-vs-code-igniter-compared>

<sup>22</sup><http://framework.zend.com/downloads/latest#ZF2>

<sup>23</sup><http://framework.zend.com/about/faq/>

<sup>24</sup><http://framework.zend.com/about/>

<sup>25</sup><https://github.com/zendframework/zf2/issues>

<sup>26</sup>[https://www.itcentralstation.com/products/comparisons/yii-framework\\_vs\\_zend-framework](https://www.itcentralstation.com/products/comparisons/yii-framework_vs_zend-framework)

<sup>27</sup><https://symfony.com/>

<sup>28</sup><https://symfony.com/>

<sup>29</sup><http://zenofcoding.com/2015/11/16/the-great-php-mvc-framework-showdown-of-2016-cakephp-3-vs-symfony-2-vs-laravel-5-vs-zend-2/>

<sup>30</sup><https://github.com/symfony/symfony>

<sup>31</sup><http://stfalcon.com/en/blog/post/largest-websites-built-on-symfony2>

<sup>32</sup><http://symfony.com/doc/current/cookbook/index.html>

<sup>33</sup> <http://symfony.com/blog/symfony-3-0-the-roadmap>

## 2.6.4 Laravel 5

Laravel<sup>34</sup> ist ein relativ junges, auf Symfony basierendes, leicht zugängliches Framework. Seit Anfang 2012 erscheinen regelmäßig Releases (aktuell ca. 2 pro Monat), aktuelle Commits sind auf Github<sup>35</sup>. Der erste LTS-Release war im Juni 2015. Das Projekt wird von einem einzelnen Entwickler verwaltet und gepflegt, Features werden über Githubs Pull-Request-System von beliebigen Entwicklern eingereicht. Laravel wird bisher noch nicht in großen Unternehmen eingesetzt, es ist eher beliebt bei kleineren Unternehmen und Privatpersonen. Es existiert eine Dokumentation, welche die Installation und Einrichtung, Architektur und Einsatz der Features abdeckt. Es existiert außerdem ein kostenpflichtiges Angebot für professionelle Videotutorial-Serien. Auf Stackoverflow lassen sich 35000 Fragen zum Tag 'Laravel' bzw. 6000 Fragen zu 'Laravel 5' finden.

### 2.6.4.1 Vorteile/Nachteile

Laravel 5 ist geeignet für Rapid Development<sup>36</sup>, relativ offen für Erweiterungen, bietet allerdings etwas weniger Struktur als andere Frameworks.<sup>37</sup> Es bietet gute Unterstützung für Unit Tests. Die Einarbeitungszeit ist gering für den sehr vollständigen Funktionsumfang. Zu den nicht selbstverständlichen Features gehören z.B. Billing, Caching, Facades und SSH.

## 2.6.5 Nette

Nettes letztes Release ist aktuell. Es kann einige Referenzen vorweisen<sup>38</sup>, und scheint auch laut Umfragen recht beliebt zu sein<sup>39</sup>. Eine ausführliche Dokumentation inklusive Code-Beispiel und GUI-Screenshots ist vorhanden. Es existiert außerdem ein offizielles, aktives Forum (Antwortzeiten 10-30 Minuten). Auf Stackoverflow ist Nette kaum präsent.

Nette Framework steht unter der freien BSD Lizenz.

### 2.6.5.1 Vorteile

Nette ist modular aufgebaut und lässt sich gut erweitern. Es bietet eine einfache Lernkurve und kann mit anderen Frameworks (z.B.) Zend kombiniert werden und auch über Add-Ons erweitert werden. Die Template-Engine ist leicht zu benutzen und ein guter PGP-Debugger sowie grafische Darstellung von Fehlern sind vorhanden.

### 2.6.5.2 Nachteile

Nette bietet kein ORM (Object-Relational-Mapping).

## 2.6.6 PHPixie

PHPixie<sup>40</sup> ist ein kleines, einfaches Framework, welches 2012 zum schnellen einmaligen Erstellen von Webseiten gegründet wurde. Es hat einen recht hohen Bekanntheitsgrad.<sup>41</sup> Besonderen Wert wird auf den schnellen Einstieg gelegt, es existiert eine Schnellanleitung, 30 Minuten Tutorials mit Screenshots, Codebeispielen und ausführlicher Beschreibung.<sup>42</sup> PHPixie steht unter der freien BSD Lizenz.

---

<sup>34</sup><https://laravel.com/>

<sup>35</sup><https://github.com/laravel/laravel>

<sup>36</sup> <https://www.quora.com/Which-should-I-learn-next-Symfony-2-or-Laravel>

<sup>37</sup><http://zenofcoding.com/2015/11/16/the-great-php-mvc-framework-showdown-of-2016-cakephp-3-vs-symfony-2-vs-laravel-5-vs-zend-2/>

<sup>38</sup><https://nette.org/>

<sup>39</sup><http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>

<sup>40</sup><https://phpixie.com/>

<sup>41</sup><http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>

<sup>42</sup><http://phpixie.jeromepasquier.com/>



### 2.6.6.1 Vorteile:

Das Framework bietet schnelle Geschwindigkeit und geringe Ladezeiten bei einer modularen und flexiblen Architektur. Es ist klein und sehr einfach zu erlernen. Für ein Nicht-Full-Stack-Framework besitzt es außerdem recht viele Features, die oft eher in größeren Frameworks zu finden sind wie z.B.: ORM, Caching, Templating, Authentication, Validation und Routing

### 2.6.6.1 Nachteile:

Da es sich nicht um ein Full-Stack-Framework handelt, sind die Funktionen beschränkt. Außerdem wird das „Single-Responsibility-Prinzip“ nicht eingehalten, was zu schlechter Kapselung und Objekten mit übermäßig viel Funktionalität führt. Es müssen außerdem verschiedene Namenskonventionen (z.B. für Klassennamen) eingehalten werden.

## 2.7 Gegenüberstellung der PHP-Frameworks

Im Folgenden werden die in „2.6 Beschreibung der PHP-Frameworks“ nach den Kriterien in „2.5 Gewichtung von PHP-Frameworks“ verglichen und gegeneinander aufgewogen.

Für die Kriterien Dokumentation und Features wird eine Range von 1 bis 2 gewählt. Es kann entweder eine gute Dokumentation vorhanden sein oder eben nicht. Dasselbe gilt für Features, entweder es gibt eine gute Auswahl an Features oder nicht.

Framework	Zend 2	Yii 2	Symfony 3	Laravel 5	Nette	PHPixie	Gewichtung	Range
Aktualität	2	2	3	3	3	1	2	1-3
Verbreitung	2	3	3	2	2	1	2	1-3
Dokumentation	1	2	2	1	2	2	4	1-2
Community	2	3	2	2	2	1	3	1-3
Einarbeitungszeit	1	2	2	3	2	3	2	1-3
Features	2	2	2	2	2	1	2	1-2
Preis	1	1	1	1	1	1	1	0-1
PHP 5	1	1	1	1	1	1	1	0-1
Gesamt	24	35	34	30	32	23		

Für alle Frameworks gilt: Freie Lizenz

Berechnungsgrundlage:

Gesamt = (Aktualität \* Gewichtung + Verbreitung \* Gewichtung + ... ) \* Preis \* PHP5

Beste Frameworks nach der Bewertung:

Platz	Framework
1	Yii 2
2	Symfony 3
3	Nette
4	Laravel 5
5	Zend 2
6	PHPixie

Das beste Framework ist nach dem gewichteten Vergleich „Yii 2“ knapp gefolgt von „Symfony 3“. Damit werden wir dieses für das Projekt als PHP-Framework verwenden.

## 2.8 Bewertung von Projektmanagement-Methoden

Wir betrachten vier Projektmanagement-Methoden mit dem Hauptaugenmerk auf Ablauf und Charakteristik. Diese werden am Ende auf der Basis der von uns gewählten Kriterien verglichen und bewertet. Die analysierten agilen Methoden sind aus einer vorhandenen Auflistung entnommen<sup>43</sup>. Die Methoden wurden genommen, da über sie bereits rudimentäres Wissen vorhanden ist. Folgende Kriterien werden von uns verwendet<sup>44 45</sup>

**Projektphase:** Phasen-spezifisch oder ebenfalls für das komplette Projekt geeignet?

Die Methode sollte sich für alle Phasen der Entwicklung anwenden lassen.

**Projektgröße:** Für welche Teamgröße und Projektdauer ist es geeignet?

Unwichtig für uns da wir ein kleines Team mit einem kleinen Projekt sind.

**Grad der Planung:** Darf das Datum, bzw. Budget, fest oder offen sein?

Wir haben kein Budget. Eine zu feste Strukturierung mit feste Daten, sowie eine vollständig fehlende Ordnung ist nicht gewünscht, weshalb wir nach einem Mittelmaß suchen.

**Änderungen während Iterationen:** Dürfen Änderungen während einer Iteration durchgeführt werden oder nur zwischen diesen? (z.B. Aufgaben-Zuteilung; Aufgaben-Aufwandseinschätzung, Neue Aufgaben in die Iteration aufnehmen, Priorität von Aufgaben)

Wir wollen flexibel in der Ausführung der Iterationen sein.

**Flexibilität:** Wie gut/flexibel lässt sich die Methode auf die eigenen Umstände anpassen? Lässt sich die Methodik schrittweise oder in Teilen einführen? (z.B. Länge der Iteration, Dauer der Meetings anpassen)

Es soll alles so flexibel wie möglich sein.

**Grad der Eigenverantwortung:** Wie hoch ist sie und wer trägt sie?

Alle Mitglieder sollen gleichermaßen verantwortlich sein.

**Einarbeitungsaufwand:** Wie schwer/aufwendig ist es, die Methodik einzuführen?

Da wir für dieses Projekt eine Managementmethode sofort anwenden, sollte diese so wenige Hürden haben wie möglich.

**Ein-Projekt- oder Mehr-Projektumgebungen:** Ermöglicht es das Managen mehrerer Projekte parallel? (z.B. eine Abteilung → zwei Projekte, Teammitglieder bearbeiten Aufgaben aus beiden Projekten)

Für uns unwichtig, da wir nur an einem Projekt arbeiten.

Wir haben uns unter den populärsten Projektmanagementmethoden die besseren herausgesucht.

---

<sup>43</sup>[https://de.wikipedia.org/wiki/Agile\\_Softwareentwicklung](https://de.wikipedia.org/wiki/Agile_Softwareentwicklung)

<sup>44</sup> <http://www.gpm-infocenter.de/PMMethoden/EinfuehrungMethodenauswahl>

<sup>45</sup> <http://www.heise.de/developer/artikel/Kriterien-fuer-eine-Entscheidung-fuer-Scrum-oder-Kanban-1071172.html>

## 2.9 Beschreibung der Projektmanagementmethoden

Im Folgenden werden vier verschiedene Projektmanagementmethoden beschrieben und gegeneinander aufgewogen.

### 2.9.1 Kanban

Der Begriff Kanban kommt aus dem Japanischem und bedeutet in etwa „Signalkarte“. Dieses Projektmanagement-Framework beinhaltet das „Pull-Prinzip“, d.h. ein Arbeiter erteilt Anfragen an andere (z.B. Nachschub anfordern) statt einer Person, die über dem ganzen steht und das Projekt überwacht. Dies soll dem Produktionsfluss dienen. Kanban legt einem wenige Vorlagen auf, weshalb es flexibel und anpassbar ist, da der Schwerpunkt auf einem optimalen Fluss (Flow) liegen soll.

#### 2.9.1.1 Kanban Ablauf

Aufgaben, Features, User-Stories und andere Formen von Anforderungen werden als Tickets (Work-Item) aufgefasst. Jedes Ticket durchläuft verschiedene Prozessschritte bis dieses fertig ist; dabei wird die benötigte Zeit gemessen. Pro Schritt dürfen sich nur eine maximale Anzahl an Tickets gleichzeitig befinden. „Staut“ es sich, so zeigt es, dass der Flow nicht optimal ist. Es müssen dann für den entsprechenden Abschnitt mehr Ressourcen hinzugefügt oder die Prozessschritte überarbeitet werden. All das soll als selbstregulierendes System den Produktionsfluss optimal halten.

#### 2.9.1.2 Kanban-Charakteristik

##### **Kaizen:**

Kaizen ist aus dem Japanischen (Kai = Veränderung, Zen = zum Besseren) und repräsentiert den KVP-Anteil (Kontinuierlicher Verbesserungsprozess) von Kanban.<sup>46</sup> Dieser beinhaltet u.a. auch die Treffen und Operations Reviews.

##### **Treffen:**

Diese finden täglich und in der Regel morgens statt. Dabei wird mittels des Kanban-Boards vorgetragen, wie weit die einzelnen Aufgaben seit dem letzten Meeting vorangeschritten sind und wo Probleme liegen, sofern vorhanden. Meetings sollten nicht länger als 15 Minuten dauern; bei größeren Diskussionen müssen diese zu einem anderen Zeitpunkt verschoben werden.

##### **Operations Reviews:**

Eine Form des Meetings, die im Gegensatz zu den täglichen Treffen unregelmäßig stattfinden. Sie dient der stetigen Verbesserung, indem alte Daten betrachtet werden (z.B. Fehlerrate, Durchlaufzeiten, Anzahl geblockter Tasks) und anhand dieser entschieden wird, was zu verbessern ist. Bei diesem Meeting treffen sich alle Mitglieder der Organisation. Operations Reviews sind nicht mit Retrospektiven aus anderen agilen Methoden zu verwechseln, die subjektiv auf die letzte Iteration(en) zurückblicken. Bei Kanban geht es um Zahlen und Fakten, die als Indikator gelten statt der Eindrücke der Mitarbeiter.

##### **Kanban-Board:**

Ein Kanban-Board ist eine simple Tafel, auf der die Prozessschritte als Spalten dargestellt werden. Die Aufgaben (Tasks) werden als Magnete, Post-its oder andere geeigneten Gegenstände spaltenweise verschoben. Dabei ist zu beachten, dass nur die erlaubte Anzahl an Tasks pro Spalte enthalten sein dürfen. Die Grenze wird üblicherweise oberhalb der Spalte eingetragen, so dass man auf einen Blick den Status der Prozesse sieht, Bottlenecks erkennt

---

<sup>46</sup> <https://de.wikipedia.org/wiki/Kaizen>

und sieht, wie gut der Produktionsfluss "fließt".

### **2.9.1.3 Kanban-Zusammenfassung**

#### **Projektphase:**

Kanban lässt sich für alle Arten von Aufgaben anwenden.

#### **Projektgröße:**

Kanban ist sehr flexibel. Es kann für ein einzelnes Team verwendet werden, aber auch für das ganze Portfolio (mehrere Wertschöpfungsketten) eines Unternehmens. Dementsprechend ist es auch für langfristigen Einsatz geeignet, es kann jedoch auch für ein einzelnes Projekt angewendet werden.

#### **Grad der Planung:**

Kanban erlaubt die Prioritätensetzung von Tickets mittels der Ticketklassen.

#### **Änderungen während Iterationen:**

Neue Anforderungen können zu jedem Zeitpunkt an das Team gegeben werden, falls Kapazitäten frei sind.

#### **Flexibilität:**

Iterationen haben keine feste Länge. Das ist gut für Einsatzgebiete, in denen man mit vielen Unterbrechungen bzw. Notfällen rechnen muss (z.B. Wartung). Treffen finden täglich zur gleichen Uhrzeit mit fixer Länge statt, andere Besprechungen sind flexibel.

#### **Grad der Eigenverantwortung:**

Hohe Eigenverantwortung wegen des Pull-Prinzips, Mitglieder und Teams holen sich ihre Aufgaben selbst.

#### **Einarbeitungsaufwand:**

Kanban setzt weder viel Vorwissen voraus, noch braucht man spezielles Equipment. Ein Whiteboard und Kärtchen reichen bereits aus. Jedoch kann man auch fertige Software dafür einsetzen was mit finanziellem Aufwand und Einarbeitung ins Programm verbunden ist.

#### **Ein-Projekt- oder Mehr-Projektumgebungen:**

Es ist möglich an mehreren Projekten gleichzeitig zu arbeiten. Die Tickets können mit verschiedenen Farben hervorgehoben werden, um diese dann entsprechend zuzuordnen bzw. man kann pro Projekt eigene Zeilen/Zeilengruppen auf dem Board bilden.

## **2.9.2 Scrum**

Scrum ist eine Projektmanagementmethodik zur agilen Softwareentwicklung. Es zeichnet sich besonders dadurch aus, dass es nur wenige Regeln kennt und daher einfach zu erlernen ist.

### **2.9.2.1 Projektmitglieder**

Personen die direkt am Prozess beteiligt sind können in drei Rollen eingeteilt werden:

- Product Owner: Stellt die fachlichen Anforderungen und priorisiert diese
- Scrum: Managt den Prozess und beseitigt gegebenenfalls Hindernisse
- Team: Entwickelt das Produkt

Als Beobachter und Berater gibt es zudem noch die sogenannten Stakeholder.

### **2.9.2.2 Ablauf**

- Anforderungen werden in den Product Backlog eingepflegt, priorisiert und ggf. erweitert
- Monatlich wird ein Arbeitspaket aus dem Product Backlog entnommen und umgesetzt (inkl. Test und Dokumentation)
- Das Arbeitspaket (Increment) wird während einer Iteration (Sprint) nicht modifiziert
- Das Arbeitspaket wird in kleinere Arbeitspakete (Tasks) eingeteilt, einem Bearbeiter zugeteilt, bekommt einen Restaufwand der täglich aktualisiert wird und wird in den Sprint Backlog geschoben
- Während des Sprints darf der Bearbeiter keinen Störungen ausgesetzt sein
- Täglich muss ein 15-Minütiges Daily Meeting abgehalten, um herauszufinden was die anderen Teammitglieder bearbeiten, was man selbst als nächstes bearbeiten muss und wo es Probleme gibt
- Am Ende eines Sprints wird die voll funktionsfähige Funktionalität dem Product Owner und den Stakeholdern live am System gezeigt
- Entsprechende Feedbacks fließen in den nächsten Sprint und das Sprint Planing Meeting mit ein
- Eine neue Iteration beginnt

### **2.9.3 Feature-Driven-Development (FDD)**

FDD ist eine in der Praxis entstandene Methodik, die versucht, agile Prinzipien in klassischen Unternehmenskulturen zu verwirklichen. Sie ist insbesondere für größere Teams geeignet.

#### **2.9.3.1 Prozess**

Der FDD-Prozess lässt sich in 2 Phasen aufteilen: Die erste Phase beinhaltet die Erstellung eines Modells, einer Feature-Liste und eines Plans (geplante Fertigstellung der Features).

Die zweite Phase ist die iterative, agile Phase, in der in einem, maximal zweiwöchigem, Zyklus Features designt und implementiert werden. Am Ende einer Iteration muss etwas stehen, was die Hände des verantwortlichen Teams verlassen kann.

#### **2.9.3.2 Teamorganisation**

Die Rolle des klassischen Team-Managers wird in FDD beibehalten. Der Manager weist jedem Feature Teammitgliedern zu und bildet so sogenannte "Feature-Teams". Solch ein Team besteht nur temporär solange, bis das Feature fertiggestellt wurde. Ein Teammitglied kann mehreren Feature-Teams zur gleichen Zeit angehören.

#### **2.9.3.3 Zusammenfassung**

##### **Projektphase:**

FDD beinhaltet keine Prozessschritte für die allgemeine grobe Spezifikation.

##### **Projektgröße:**

FDD skaliert problemlos auch mit einem sehr großen Team, inklusive zusätzlich vorhandenen Spezialisten oder Spezial-Teams.

##### **Grad der Planung:**

In der Planungsphase werden die Fertigstellungstermine der Features festgelegt. Es sind nur geringe Änderungen am Plan vorgesehen. Verspätungen weisen auf Probleme hin.

##### **Änderungen während Iteration:**

Änderungen sollten im Rahmen des jeweils dafür vorgesehenen Prozess-Schrittes ausgeführt werden, nicht außerhalb.

##### **Flexibilität:**

Der FDD-Prozess ist relativ spezifisch. Für jeden Prozessschritt gibt es eine Empfehlung, wie viel Zeit für diesen Schritt eingeplant werden sollte.

### **Grad der Eigenverantwortung:**

Die Eigenverantwortung der Entwickler ist abhängig vom Management-Stil des Team-Managers. Dies kann je nach Stil zu geringer Eigenverantwortung führen, wenn der Manager auf einer direkten Leitung besteht. In jedem Fall ist der Manager für sein Team verantwortlich.

### **Einarbeitungsaufwand:**

FDD ist sehr einfach auf klassische Unternehmensstrukturen abbildbar und damit in einem Unternehmen sehr einfach und konfliktfrei einführbar. In einer Struktur, die einem Unternehmen nicht ähnelt, ist jedoch viel Aufwand erforderlich.

### **Ein-Projekt- oder Mehr-Projektumgebungen:**

FDD ist besonders geeignet für hierarchisch organisierten Projektstrukturen, in denen mehrere Unterprojekte auf das Erfüllen einer gemeinsamen Aufgabe hinarbeiten. Diese hierarchischen Strukturen können relativ frei gestaltet werden. Die jeweiligen Projektleiter können sich dabei einigen, wie die vorhandenen Mitarbeiter für Features eingeteilt werden sollen.

## **2.9.4 Extreme Programming (XP)**

Ein bekannter agiler Softwareprozess ist Extrem Programming. Dabei werden die, sich während der Entwicklung ändernden, Kundenwünsche stets berücksichtigt. Der Softwareprozess wird immer wieder in kurzen Zyklen durchlaufen. Nur die im Iterationsschritt festgelegten Anforderungen werden implementiert. Diese Methode beruht auf der Erfahrung, dass der Kunde alle vollständigen Anforderungen seiner Software zu Projektbeginn noch nicht genau kennt.

### **2.9.4.1 Projektphase**

Der Kunde hat den gleichen Projektstand wie der Entwickler. Jede Komponente besitzt einen Modultest, um Fehler so früh wie möglich zu erkennen und zu beseitigen. Auf einen ehrlichen Umgang mit dem Kunden wird Wert gelegt, um Fehler mit Mitarbeitern und Kunden zu minimieren.

Die Projektphase gliedert sich in folgende Einheiten, die aufeinander aufbauen:

- Sekundentakt: Durch Pair-Programming wird die gegenseitige Kontrolle erhöht, und Fehler vermieden.
- Minutentakt: UNIT-Tests werden vor dem Code geschrieben, so dass der Code ständig überprüft wird
- Stundentakt: Entwickelte Komponenten werden sofort im Gesamtsystem integriert, und somit Fehler schneller gefunden.
- Tagestakt: tägliches Treffen des Entwicklungsteams um den Projektfortschritt zu reflektieren, und Fehlentwicklungen zu vermeiden.
- Wochentakt: Wöchentlicher Kundenkontakt, um die aktuell lauffähige Version auszuliefern, und um neue Änderungen zu besprechen.
- Monatstakt: dem Kunden wird regelmäßig eine neue Release ausgeliefert.

Die kleineren Einheiten sind immer mehrfach in der größeren enthalten. Solange eine Iteration fehlerhafte Tests liefert wird diese in mehrere kleiner Iterationsphasen aufgeteilt und in die nächste Iteration verschoben. Auch während der Iteration können Änderungen vom Kunden durchgeführt werden.

Um erfolgreiche Software entwickeln zu können müssen die Werte Kommunikation, Einfachheit, Rückmeldung, Mut und Respekt eingehalten werden. Das Vorgehen wird im Team ständig hinterfragt, um Prinzipien wie z.B. Selbstsicherheit und Qualität zu gewährleisten. Dadurch ist lauffähige Software zu jedem Zeitpunkt gewährleistet, bei welcher jedoch anfängliche Fehlschläge mit einkalkuliert werden müssen. Zum Einsatz kommen Praktiken wie z.B. Pair-Programming, bei welchem durch das Vier-Augen-Prinzip Fehler sofort gefunden werden, oder Refactoring, bei dem ständig Architektur-, Design- und Code-Verbesserungen vorgenommen werden.

### **2.9.4.2 Vorteile**

- Schnelle Entwicklung einer funktionierenden Software
- Nur eingesetzte Funktionen werden entwickelt
- Softwareentwicklung lässt sich mit den Kunden vereinbaren, und bietet schnelle, bewertbare Ergebnisse
- Mehr Kommunikation im Projekt durch kurze Zyklen
- Kunde wird stärker in den Entwicklungsprozess eingebunden, und hat dadurch ein besseres Verständnis
- Kosten bleiben bei großen Software-Projekten verhältnismäßig gering
- Hohe Qualität, Erweiterbarkeit und Nachhaltigkeit bei der Entwicklung
- XP hat eine positive Auswirkung auf den Teamgeist und der Motivation der Entwickler so wie ein Erfolgserlebnis bei der Auslieferung einer neuen Version

### **2.9.4.3 Nachteile**

- Schlecht bei kleinen Softwareprojekten
- Nur, bei Beginn definierte, Anforderungen können mit dem geringsten Aufwand umgesetzt werden
- Hohe Anforderung an die Entwickler, da die Software in jedem Zustand erweiterbar sein sollte
- Kunde hat nicht immer ausreichend Zeit, um sich regelmäßig mit den Entwicklern zu treffen
- Die Test-Entwicklung kann schnell zum Endprodukt werden bzw. ausgelassen werden
- Der Kunde muss Vertrauen in den Programmierer haben

### **2.9.4.4 Zusammenfassung**

#### **Projektphase:**

Das Projekt wird in kleine Teilschritte zerlegt, die einzelne Funktionsblöcke enthalten.

#### **Projektgröße:**

Gut geeignet für kleine bis mittelgroße Teams (<10-12 Mitglieder).

#### **Grad der Planung:**

Durch das Mitwachsen der Anforderungen verläuft die Kostenkurve gleichmäßig linear.

#### **Änderungen während Iterationen:**

XP ist auf schnelle Anforderungen ausgelegt weswegen Änderungen immer möglich sein müssen.

#### **Flexibilität:**

XP muss in einem Zug eingeführt werden, die Dauer der Iteration hängt vom Kunden ab.

#### **Grad der Eigenverantwortung:**

Die Eigenverantwortung liegt von der Funktionalität beim Kunden, und die Zuverlässigkeit und Umsetzung beim Entwickler.

#### **Einarbeitungsaufwand:**

XP erfordert Softwareentwickler mit besonderen Fähigkeiten, da alle Teammitglieder Entwickler sind.

#### **Ein-Projekt- oder Mehr-Projektumgebungen:**

Der Projektmanager kann an mehreren Projekten arbeiten.

## 2.10 Gegenüberstellung der Projektmanagementmethoden

Im Folgenden werden die Projektmanagementmethoden, die wir ausgewählt haben, gegeneinander verglichen. Für die Kriterien Projektphase und Mehrprojektumgebung wird ein Wertebereich von 0 bis 1 gewählt, da es entweder geht oder nicht. Für uns waren die Kriterien Eigenverantwortung und Einarbeitungsaufwand die wichtigsten Kriterien, mit einer Gewichtung von 3 und einem Wertebereich von 2 bis 4. Um zu verhindern, dass ein unwichtiges Kriterium ein schlecht bewertetes Framework zu stark verbessert, ist der Wertebereich der Kriterien mit Gewichtung „eins“ höher. Für das Kriterium Flexibilität wird der Wertebereich von 1 bis 2 gewählt, da eine Methodik entweder flexibel sein kann oder nicht. Alle anderen Kriterien haben einen Wertebereich von 1 bis 3, jeweils ob diese das Kriterien gut, sehr gut oder ausgezeichnet erfüllen.

	Scrum	Kanban	XP	FDD	Gewichtung	Range
<b>Projektphase</b>	1	1	0	0	2	0-1
<b>Projektgröße</b>	1	3	1	3	1	1-3
<b>Grad der Planung</b>	3	2	2	1	2	1-3
<b>Änderung während Iteration</b>	0	1	1	0	2	0-1
<b>Flexibilität</b>	1	2	1	1	2	1-2
<b>Grad Eigenverantwortung</b>	2	3	3	1	3	1-3
<b>Einarbeitungsaufwand</b>	2	2	3	1	3	1-3
<b>Mehrprojektumgebung</b>	1	1	0	1	1	0-1
<b>Gesamt</b>	<b>24</b>	<b>31</b>	<b>27</b>	<b>14</b>		

Berechnungsgrundlage:

Gesamt = Projektphase \* Gewichtung + Projektgröße \* Gewichtung + ...

Beste Projektmanagementmethodik:

Platz	Framework
<b>1</b>	Kanban
<b>2</b>	XP
<b>3</b>	Scrum
<b>4</b>	FDD

Die beste Projektmanagementmethodik nach dem gewichteten Vergleich ist Kanban gefolgt von Extreme Programming. Aufgrund der vom Kunden gestellten Anforderungen wird allerdings dennoch Scrum gewählt.



## 2.11 Software-Spezifikation

Zu Beginn dieser Spezifikation werden wir zwei Frameworks für die Endanwendung analysieren und vergleichen, um uns für eine der beiden zu entscheiden. Anschließend erklären wir mit UML-2 Diagrammen den Ablauf unseres Programms, welche Funktionen in der GUI aufgerufen werden können und welche Informationen wir verwalten.

## 2.12 Frameworks

### 2.12.1 Bootstrap

#### 2.12.1.1 Einleitung

Bootstrap wurde von Twitter entwickelt und war dafür gedacht, die eigene aufwändige Oberflächenentwicklung zu erleichtern. Ursprünglich als internes Werkzeug gedacht, wurde das Potential von Bootstrap später erkannt und 2011 als Open-Source-Projekt veröffentlicht.

#### 2.12.1.2 Eigenschaften

Bootstrap lässt sich über eine spezielle „Customize“-Option verändern, um für das eigene Projekt nur die benötigten Komponenten anzuwählen und diverse Werte (z.B. Schriftgröße, -farbe, Breite der Gridspalten) an die eigenen Bedürfnisse anzupassen. Letzteres wird dadurch erleichtert, dass der LESS CSS-Compiler verwendet wird, der Style-Variablen ermöglicht.

#### 2.12.1.3 Vorteile

- Beinhaltet wiederverwendbare Komponenten wie z.B. Buttons oder Navigationselemente
- Beinhaltet jQuery-Plugins für Menüs, Tooltips, etc.
- Arbeitet mit dem 940 Grid System (12 Spalten)
- Weitreichende Kompatibilität mit Browsern
- Kann mit zahlreichen Erweiterungen ausgebaut werden (z.B. für Wordpress oder Facebook Applikationen)
- Verhilft sehr schnell zu gut aussehenden Ergebnissen

#### 2.12.1.4 Nachteile

- Keine Klassen für mehrschichtige Menüs vorhanden
- Die oben genannten jQuery-Plugins sind sehr simpel.
- Bei zu viel Spielerei können Schönheitsfehler entstehen (z.B. Formulare verschoben)
- Das Grid System ist relativ groß was sich negativ auf die Ladezeit der Webseite auswirkt

### 2.12.2 Semantic UI

#### 2.12.2.1 Einleitung

Semantic UI ist ein sehr modernes Framework, und wurde im September 2013 von der Semantic Organisation veröffentlicht. Trotz seines jungen Alters ist es relativ beliebt und aktiv auf Github.

#### 2.12.2.2 Eigenschaften

Das Besondere an diesem Framework ist, dass für alle Gestaltungselemente semantische CSS-Klassen verwendet werden. Als Stylesheet-Sprache wird LESS verwendet. Das Framework besitzt ein 16-spaltiges Layout-Raster, und bietet einige Gestaltungselemente, wie zum Beispiel Sterne-Bewertung, welche andere Front-End-Frameworks nicht zur Verfügung stellen. HTML-Klassen werden aus einer intuitiv erfassbaren, sinnvollen Wortfolge gebildet.

### 2.12.2.3 Vorteile

- Layoutmuster können vom Code abgeleitet werden (gutes Design)
- Die Klassen sind einfach zu verstehen
- Lesbar und verständlich
- Keine starre Syntax und damit sehr flexibel erweiterbar
- 800 Forks auf Github
- Sehr viele gutaussehende Default-Themes

### 2.12.2.4 Nachteile

- Hoher Schreibaufwand
- Wird noch nicht so lange in der Praxis eingesetzt
- Responsive Design für mobile Geräte noch recht aufwändig
- Nicht besonders kompatibel mit älteren Browsern (z.B. Internet Explorer)

### 2.12.3 Vergleich

Relevante Kriterien für den Einsatz in diesem Projekt:

#### **Browser-Kompatibilität:**

Bootstrap ist mit mehr Browsern kompatibel als Semantic UI. Dieses Kriterium ist sehr wichtig, da die Zielgruppe "Grundschulkinder" nicht durch Kompatibilitätsprobleme vor Hürden bei der Benutzung gestellt werden soll.

#### **Einarbeitungsaufwand:**

Bootstrap ist für Prototyping sehr gut geeignet, da es den geringeren Einarbeitungsaufwand hat. Dieses Kriterium ist wichtig, da für die Implementierung wenig Zeit verfügbar ist.

#### **Erstellung komplexer GUIs:**

Semantic-UI kann programmtechnisch besser angepasst werden und verfügt über eine Integration für AngularJS. Dieses Kriterium ist wichtig, da die Spiele eine relativ komplexe GUI erfordern.

#### **Community:**

Bootstrap besitzt aufgrund seiner größeren Verbreitung eine größere Community. Dieses Kriterium ist wichtig, da so auftretende Probleme mit höherer Wahrscheinlichkeit sehr schnell gelöst werden können.

## **2.13 Ablauf**

### **2.13.1 Registrierung**

Die Registrierung der Nutzer (Schüler) und Eltern darf nur vom Leiter (Lehrer) durchgeführt werden. Der Nutzernamen ist eindeutig und besteht aus Vor- und Nachnamen des Nutzers und dem Schuljahr (z.B. maxmustermann16 im Schuljahr 2016).

### **2.13.2 Login/Anmeldung**

Nachdem ein Nutzer registriert wurde, kann er sich mit seinem Benutzernamen anmelden und aus den Aufgabengruppen und -typen eine auswählen. Im Falle eines Leiters oder eines Eltern kann die Statistikansicht aufgerufen werden.

### **2.13.3 Spielübersicht**

Der Nutzer sieht eine Übersicht aller vorhandenen Spiele. Hierbei werden die abgeschlossenen und angefangenen Spiele visuell hervorgehoben. Für die abgeschlossenen Spiele erhält der Spieler eine Belohnung.

### **2.13.4 Aufgabeansicht**

Der Spieler sieht die aktuelle Aufgabe des gewählten Spiels, inklusive Beschreibung, Aufgabenstellung und Hilfestellung. Zusätzlich hat er eine Fortschrittsanzeige im aktuell gewählten Spiel (z.B. Fortschritt: 3 von 10). Außerdem hat er jederzeit die Möglichkeit auf die Spielübersicht zurück zu kehren.

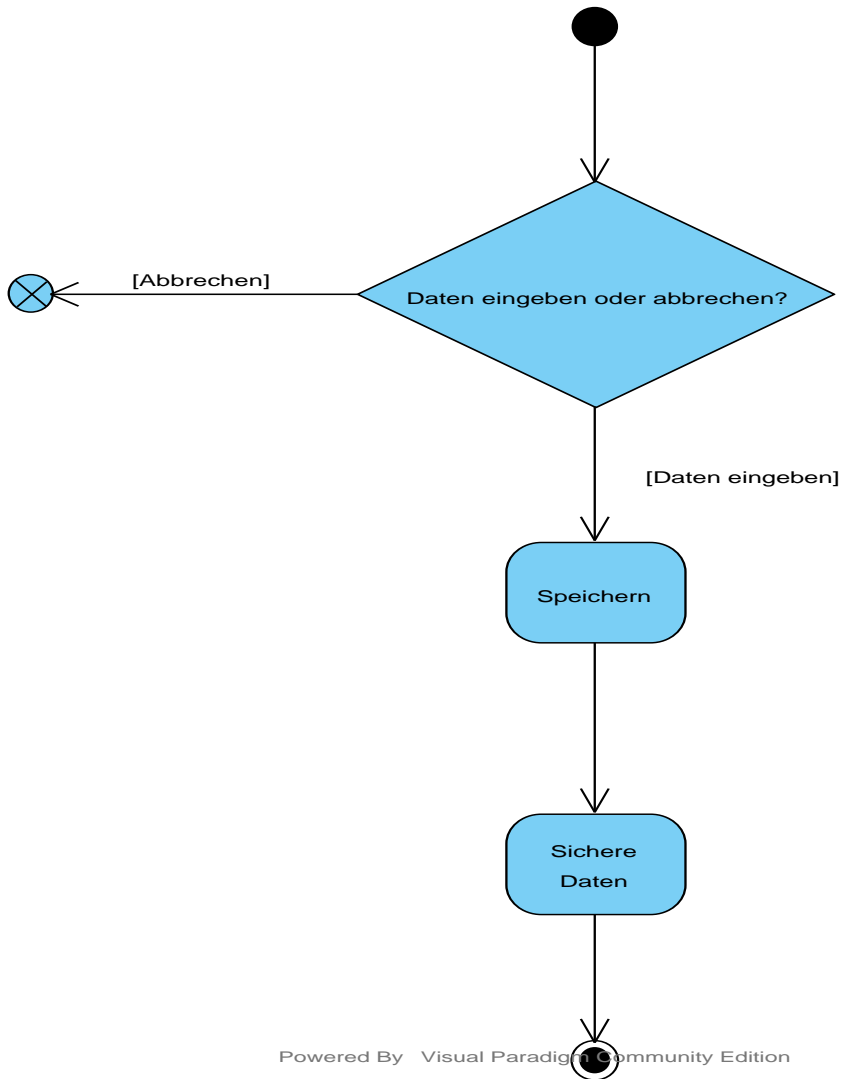
### **2.13.5 Statistikübersicht**

Der Leiter hat Zugriff auf eine Statistikübersicht. Dort sieht er spaltenweise für jeden Schüler dessen Fortschritt pro Spiel. Beim Klick auf die Spalte kommt man zu der detaillierten Übersicht des angewählten Nutzers.

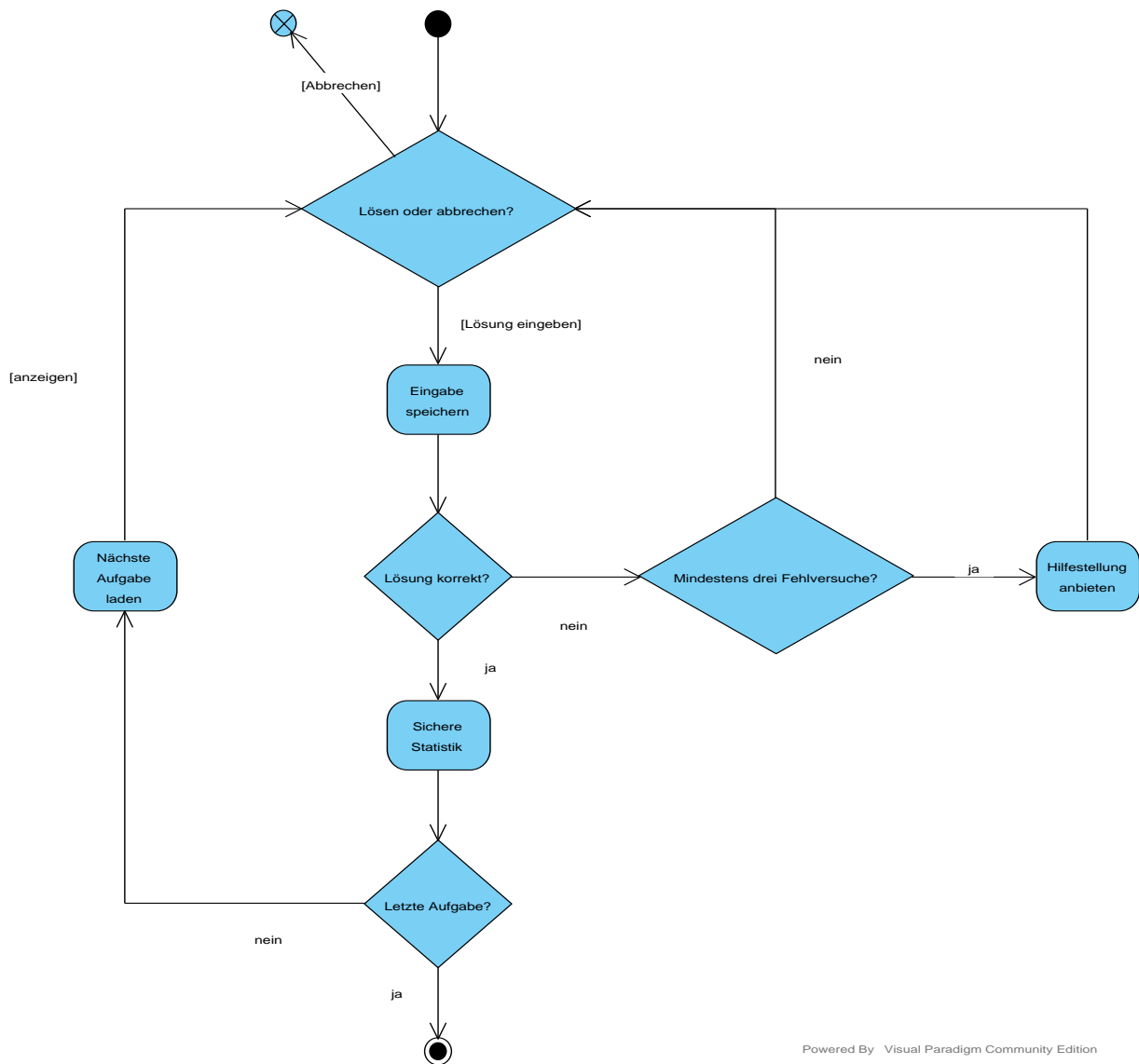
### **2.13.6 Benutzerdefinierte Statistikansicht**

In dieser wird pro Aufgabe die benötigte Zeit und Anzahl der Versuche angezeigt. Von hier kann man die entsprechenden Aufgaben hervorholen.

### 2.13.7 Aktivitätsdiagramme



Das Anlegen von Nutzern ist einfach. Der Leiter gibt Daten ein und betätigt den Button zum Speichern. Daraufhin werden die Daten z.B. in eine Datenbank gesichert.

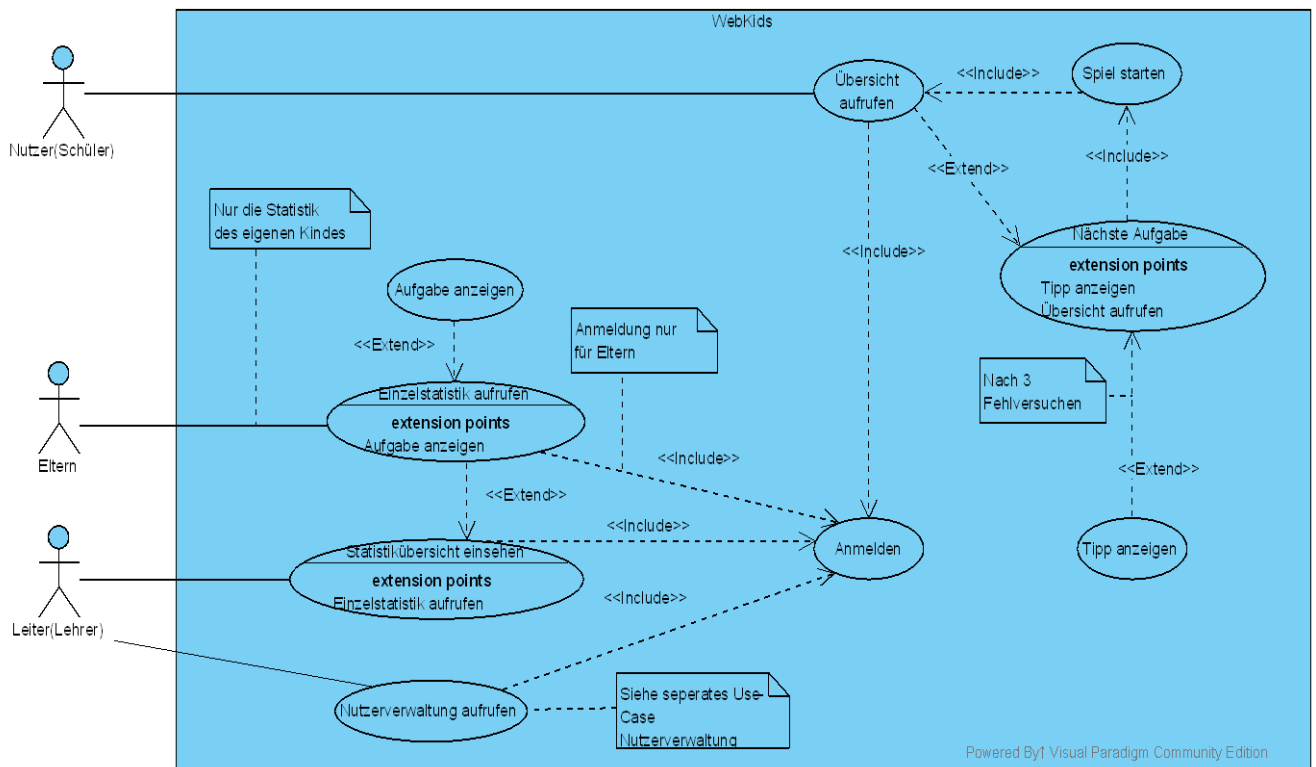


Powered By Visual Paradigm Community Edition

Sobald der Nutzer ein Spiel ausgewählt hat, werden ihm eine Reihe von Aufgaben gestellt. Jede Eingabe wird gespeichert und danach mit der richtigen Lösung verglichen. Sobald die korrekte Eingabe getätigt wurde, werden die entsprechenden Statistiken (z.B. benötigte Zeit) gespeichert und die nächste Aufgabe angezeigt, sofern es noch welche gibt, ansonsten ist das Spiel zu Ende. Werden wiederholt falsche Eingaben getätigt, wird dem Nutzer mit Tipps geholfen, um ein Feststecken des Nutzers zu verhindern.

## 2.13 Programm

### 2.13.1 Hauptmenü

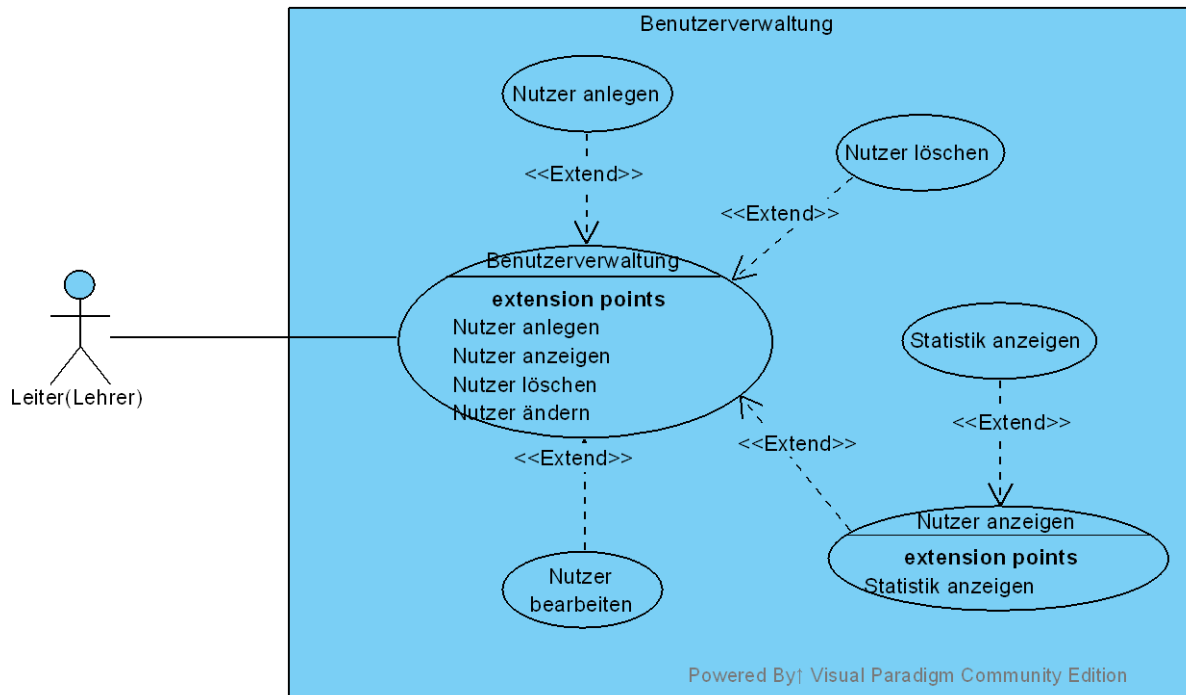


Nachdem man sich angemeldet hat kann man, je nach Berechtigung, folgende Aktionen tätigen:

- Spielübersicht anzeigen und spielen
- Statistikübersicht und/oder einzelne Statistikansichten anzeigen
- Nutzerverwaltung aufrufen

## 2.13.2 Nutzer

Die Nutzerverwaltung besitzt die Möglichkeit die Nutzerliste einzusehen und zu bearbeiten.



### 2.13.2.1 Zu pflegende Eigenschaften

Bei der Erstellung der Nutzer gibt es folgende Eigenschaften:

- Name
- Passwort für Elternzugriff
- Generierte Aufgaben
- Statistik

### 2.13.3 Statistik

Bei der Benutzung der Spiele werden folgende Eigenschaften festgehalten:

- Zeit bis zur Eingabe der richtigen Lösung
- Anzahl Fehlversuche
- Anzahl gelöster Aufgaben pro Spiel
- Nutzer

## 2.14 Testspezifikation

Projektname:	WebKids
Projektversion:	1.0
erstellt am:	
Testname:	Akzeptanztest01
Testebene:	
Testergebnis freigegeben am:	

Bearbeitungshinweis:

In der Testspezifikation werden alle Testfälle mit Ergebnissen beschrieben. Das Testprotokoll befindet sich am Ende.

Diese Spezifikation ist auf folgender Vorlage aufgebaut:

<https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0ahUKEwjpvffvgPXMAhVJ7BQKHbOCrGQFghaMAY&url=ftp%3A%2F%2Fwww.inf.fh-dortmund.de%2Fpub%2Fcontributors%2Fschlichtherle%2FLiteratur%2FVortr%25E4ge%2520SS%25202002%2F08%2520-%2520Testspezifikation.doc&usg=AFQjCNF2y7Yagwpf8siSCj7ojjbgPqHt7w&sig2=KWVHbz7i6hYr1DHEUSIU1g>

### 2.14.1 Testfälle

#### 2.14.1.1 Login

##### 2.14.1.1.1 Kind-Login

Was: Kind-Login (kindgerecht, intuitiv, visuell ansprechend)

Wer: Schüler-User

Wann: Nach Implementierung der Anmeldung

Voraussetzungen: Kind-Testaccount ist erstellt

Wie: Tester versucht sich einzuloggen (Tester entspricht Kind; Accountname: Child0)

Wie oft: Ein Mal

##### 2.14.1.1.2 Eltern-Login

Was: Eltern-Login (intuitiv, visuell ansprechend)

Wer: Eltern-User

Wann: Nach Implementierung der Anmeldung

Voraussetzungen: Kind-Testaccount ist erstellt -> Elternpasswort ist vergeben

Wie: Tester versucht sich einzuloggen (Tester entspricht Eltern; Accountname: Child0, Password: Parent0)

Wie oft: Ein Mal

##### 2.14.1.1.3 Lehrer-Login

Was: Lehrer-Login (intuitiv, visuell ansprechend)

Wer: Lehrer-User

Wann: Nach Implementierung der Anmeldung

Voraussetzung: Lehreraccount ist erstellt

Wie: Tester versucht sich einzuloggen (Tester entspricht Lehrer; Accountname: Teacher0, Password: root)

Wie oft: Ein Mal



## **2.14.1.2 Navigation**

### **2.14.1.2.1 Navigation zur Aufgabe**

Was: Navigation zu Aufgabe (kindgerecht, intuitiv, visuell ansprechend)

Wer: Schüler-User

Wann: Nach Implementierung der Übersicht

Voraussetzung: Testaccount ist erstellt -> Login ist erfolgt (Accountname: Child0)

Wie: Tester versucht zu einer bestimmten Aufgabe zu gelangen

Wie oft: Drei Mal

### **2.14.1.2.2 Navigation in Aufgabe**

Was: Navigation in der Aufgabe (Zurück/Hilfe) (kindgerecht, intuitiv, konsistent, visuell ansprechend)

Wer: Schüler-User

Wann: Nach Implementierung aller Aufgaben

Voraussetzung: Testaccount ist erstellt -> Login ist erfolgt (Accountname: Child0) -> Tester ist in Aufgabe angelangt

Wie: Tester versucht innerhalb einer Aufgabe zu navigieren, Tester versucht zurück in die Übersicht gelangen, Tester öffnet/schließt Hilfe

Wie oft: Ein Mal pro Aufgabentyp

### **2.14.1.2.3 Navigation allgemein**

Was: Navigation allgemein (konsistent, verständlich, Aufgaben differenzierbar)

Wer: Schüler-User

Wann: Nach Implementierung der Übersicht

Voraussetzung: Testaccount ist erstellt -> Login ist erfolgt (Accountname: Child0)

Wie:

- Tester öffnet alle Aufgaben einmal (inkl. Hilfe) und versucht, sich einen Überblick zu verschaffen.
- Anschließend versucht der Tester nur anhand der Übersicht aufzuzählen, welche Aufgabentypen er sich gemerkt hat und welcher Eintrag in der Navigation zu welcher Aufgabe gehört.

Wie oft: Ein Mal

## **2.14.1.3 Aufgaben**

### **2.14.1.3.1 Aufgabe lösen**

Was: Aufgabe lösen (kindgerecht, intuitiv, konsistent, verständlich, visuell ansprechend)

Wer: Schüler-User

Wann: Nach Implementierung aller Aufgaben

Voraussetzung: Testaccount ist erstellt -> Login ist erfolgt (Accountname: Child0) -> Tester ist in Aufgabe gelangt

Wie: Tester versucht die Aufgabe zu lösen, Fehlversuche werden getestet, erfolgreiche Aufgabe wird getestet, Hilfe wird getestet

Wie oft: Ein Mal pro Aufgabentyp: Ein Mal erfolgreich, mindestens vier Mal mit Fehlversuchen

## **2.14.1.4 Statistik**

### **2.14.1.4.1 Lehrer: Navigation in Statistik**

Was: Navigation in der Statistik (für Lehrer intuitiv, konsistent, verständlich)

Wer: Lehrer-User

Wann: Nach Implementierung der Statistik und aller Aufgaben

Voraussetzung: Lehreraccount wurde erstellt, mehrere Schüleraccounts sind vorhanden, Testdaten sind vorhanden (Accountname Schüler: Child0, Child1, Child2; Accountname Lehrer: Teacher0, Password: root)

Wie: Tester navigiert durch die Übersicht, Tester wählt einzelne Schüler und deren Statistik aus, Tester springt in eine Aufgabe

Wie oft: Ein Mal

#### **2.14.1.4.2 Eltern: Navigation in Statistik**

Was: Navigation in der Statistik (für Eltern, intuitiv, konsistent, verständlich, visuell ansprechend)

Wer: Eltern-User

Wann: Nach Implementierung der Statistik und aller Aufgaben

Voraussetzung: Elternaccount wurde erstellt, entsprechender Schüleraccount ist vorhanden, Testdaten sind vorhanden (Accountname Schüler: Child0, Child1; Accountname Eltern: Child0, Child1, Password: Parent0, Parent1)

Wie: Tester navigiert durch die Statistik des Schülers, Tester springt in Aufgabe

Wie oft: Ein Mal für Child0, ein Mal für Child1

#### **2.14.1.5 Verwaltung**

##### **2.14.1.5.1 Navigation in Benutzerverwaltung**

Was: Navigation in der Benutzerverwaltung (für Lehrer, intuitiv, konsistent)

Wer: Lehrer-User

Wann: Nach Implementierung der Benutzerverwaltung

Voraussetzung: Mindestens ein Account pro Accounttyp ist vorhanden (Eltern, Lehrer, Schüler; Accountname

Lehrer: Teacher0, Password: root; Accountname Eltern: Child0, Password: Parent0; Accountname Schüler: Child0)

Wie: Tester navigiert durch die Benutzerverwaltung, Tester springt aus der Benutzerverwaltung in die Statistik eines Benutzers

Wie oft: Ein Mal

##### **2.14.1.5.2 Nutzerverwaltung**

Was: Nutzerverwaltung (für Lehrer, intuitiv, konsistent)

Wer: Lehrer-User

Wann: Nach Implementierung der Benutzerverwaltung

Voraussetzung: Lehreraccount vorhanden (Accountname: Teacher0, Password: root)

Wie: Tester legt jeweils einen Account eines Accounttyps an, Tester sieht sich die Accounts an, Tester bearbeitet die Accounts, Tester löscht die Accounts

Wie oft: Ein Mal

## 2.14.2 Testfälle und deren Ergebnisse

Nr.	Testfall	Eingabe /Parameter	Erwartung	Ergebnis
1	Nutzerverwaltung	Schüleraccount: Child0, Elternaccount: Child0, Parent0, Lehreraccount: Teacher0, root, Jeweils leere Accountnamen und Passwörter	Accounts können angezeigt werden, Löschen funktioniert; Auf leere Felder wird hingewiesen	
2	Navigation in der Benutzerverwaltung	Anklicken der Links	Keine fehlerhaften Verlinkungen	
3	Kind-Login	Accountname: Child0; Leerer Accountname, Nicht vorhandener Accountname	Weiterleitung auf Übersicht; Benachrichtigung dass Account nicht vorhanden; Auf leere Eingabe wird hingewiesen	
4	Eltern-Login	Accountname: Child0, Password: Parent0; Leerer Accountname, Nicht vorhandener Accountname	Weiterleitung auf Statistik; Benachrichtigung dass Account nicht vorhanden; Auf leere Eingabe wird hingewiesen	
5	Lehrer-Login	Accountname: Teacher0, Password: root; Leerer Accountname, Nicht vorhandener Accountname	Weiterleitung auf Statistikübersicht; Benachrichtigung dass Account nicht vorhanden; Auf leere Eingabe wird hingewiesen	
6	Navigation zur Aufgabe	Anklicken von Links	Keine fehlerhaften Verlinkungen	
7	Navigation in Aufgabe	Anklicken von Links	Keine fehlerhaften Verlinkungen	
8	Navigation allgemein	Anklicken von Links	Keine fehlerhaften Verlinkungen	
9	Aufgabe lösen	Anklicken von Buttons, Eingabe von Zahlen	Überprüfung erfolgreich, Berechnung korrekt, Ergebnis wird angezeigt (erfolgreich/erfolglos)	
10	Eltern: Navigation in der Statistik	Anklicken von Links	Keine fehlerhaften Verlinkungen	
11	Lehrer: Navigation in der Statistik	Anklicken von Links	Keine fehlerhaften Verlinkungen	

### **3 Testprotokolle**

# 4 Adminhandbuch

# 5 Benutzerhandbuch

# 6 Quellen

## Kanban

<http://www.heise.de/developer/artikel/Kriterien-fuer-eine-Entscheidung-fuer-Scrum-oder-Kanban-1071172.html>  
<https://de.wikipedia.org/wiki/Kanban-Tafel>  
[http://de.wikipedia.org/wiki/Kanban\\_%28Softwareentwicklung%29](http://de.wikipedia.org/wiki/Kanban_%28Softwareentwicklung%29)  
[https://www.it-agile.de/fileadmin/docs/veroeffentlichungen/Artikel\\_OS\\_02.10\\_Interview.pdf](https://www.it-agile.de/fileadmin/docs/veroeffentlichungen/Artikel_OS_02.10_Interview.pdf)  
<https://www.it-agile.de/wissen/methoden/kanban/fragen-zu-kanban/>

## Scrum

<http://de.wikipedia.org/wiki/Scrum>  
[http://scrum-master.de/Was\\_ist\\_Scrum/Scrum\\_auf\\_einer\\_Seite\\_erklaert](http://scrum-master.de/Was_ist_Scrum/Scrum_auf_einer_Seite_erklaert)  
<https://www.crisp.se/file-uploads/Kanban-vs-Scrum.pdf>  
<http://www.heise.de/developer/artikel/Kriterien-fuer-eine-Entscheidung-fuer-Scrum-oder-Kanban-1071172.html>  
<https://www.scrumalliance.org/community/articles/2014/july/scrum-vs-kanban>

## FDD

<https://www.youtube.com/playlist?list=PLS7DQ8YkIVTpXyiWb2zl5F5rtpHhhPfsJ>  
[https://en.wikipedia.org/wiki/Feature-driven\\_development](https://en.wikipedia.org/wiki/Feature-driven_development)  
<https://dzone.com/articles/introduction-feature-driven>

## XP

<https://blog.seibert-media.net/blog/2005/05/01/extreme-programming-vorgehensmodell-zur-software-entwicklung-bei-seibertmedia/>  
<https://www.it-agile.de/wissen/methoden/extreme-programming/>  
<http://www.computerwoche.de/a/extreme-programming,2352505,2>  
<http://www.computerwoche.de/a/extreme-programming,2352505,3>  
[https://de.wikipedia.org/wiki/Extreme\\_Programming#Flexibilit.C3.A4tsgrad\\_vs.\\_Steifheit](https://de.wikipedia.org/wiki/Extreme_Programming#Flexibilit.C3.A4tsgrad_vs._Steifheit)  
<https://www.st.cs.uni-saarland.de/edu/lehrer/xp.pdf>

## Bootstrap

<http://getbootstrap.com/>  
<http://tutorialbox.de/2013/01/bootstrap-10-gute-gruende-fuer-das-open-source-framework-von-twitter/>  
[https://de.wikipedia.org/wiki/Bootstrap\\_%28Framework%29](https://de.wikipedia.org/wiki/Bootstrap_%28Framework%29)  
<https://www.feenders.de/ratgeber/experten/646-reagierendes-design-webseiten-zukunftssicher-erstellen.html>  
<http://webresourcesdepot.com/20-beautiful-resources-that-complement-twitter-bootstrap/>

## Semantic UI

<http://semantic-ui.com/>  
<http://nodecode.de/open-source-css-frameworks>  
<https://blog.concept2go.de/semantic-ui/>  
<https://www.quora.com/Do-you-prefer-Semantic-UI-or-Bootstrap>  
[http://www.slant.co/topics/150/compare/~foundation\\_vs\\_bootstrap\\_vs\\_semantic-ui](http://www.slant.co/topics/150/compare/~foundation_vs_bootstrap_vs_semantic-ui)  
<http://stackshare.io/stackups/bootstrap-vs-foundation-vs-semantic-ui>