Dokumentation MKN 151

Hangman

David Spörri

2015

Inhalt

[Tiers 2](#_Toc431133508)

[Tier 1 – Presentation 2](#_Toc431133509)

[LESS 2](#_Toc431133510)

[jQuery und jQuery UI 2](#_Toc431133511)

[Bootstrap 2](#_Toc431133512)

[FontAwesome 3](#_Toc431133513)

[Die View- und Formularkomponenten des Zend Framework 2 3](#_Toc431133514)

[Tier 2 – Webserver 3](#_Toc431133515)

[Aufbereitung der Daten für den ersten Tier 3](#_Toc431133516)

[Ansteuerung des dritten Tiers 3](#_Toc431133517)

[Validierung von Formulardaten 3](#_Toc431133518)

[Authentifizierung 4](#_Toc431133519)

[Spiellogik 4](#_Toc431133520)

[Tier 3 – Application Server 5](#_Toc431133521)

[Automatische Erstellung bzw. Aktualisierung der Datenbank anhand erstellter Entities. 5](#_Toc431133522)

[Einfache und Komplexere Möglichkeiten, Daten aus der Datenbank zu laden 5](#_Toc431133523)

[Paginator 5](#_Toc431133524)

[Sicherheit 5](#_Toc431133525)

[Weitere Datentypen 5](#_Toc431133526)

[Beziehungen zwischen Entities 5](#_Toc431133527)

[Einfache Integration in Zend Framework 2 6](#_Toc431133528)

[Tier 4 – Datenbank 6](#_Toc431133529)

[Konzeptionelles Datenmodell 6](#_Toc431133530)

[Relationales Datenmodell 7](#_Toc431133531)

[Anmerkungen 7](#_Toc431133532)

[Austauschbarkeit der Tiers 7](#_Toc431133533)

[Tier 1 – Presentation 7](#_Toc431133534)

[Tier 2 – Webserver 7](#_Toc431133535)

[Tier 3 – Application Server 7](#_Toc431133536)

[Tier 4 – Datenbank 8](#_Toc431133537)

[GUI-Prototypen 8](#_Toc431133538)

[Spielinterface 8](#_Toc431133539)

[Wörteradministration 9](#_Toc431133540)

[Testfälle 9](#_Toc431133541)

[Testprotokoll 12](#_Toc431133542)

[Abnahmeprotokoll 13](#_Toc431133543)

# Vorwort, Allgemeines

In dieser Dokumentation möchte ich festhalten, welche Werkzeuge wie verwendet wurden, um die Hangman-Applikation für den MKN 151 umzusetzen.

Die Hangman-Applikation wurde mit PHP auf Basis des Zend Framework 2 (kurz ZF2), einem der weitverbreitetsten PHP-Frameworks umgesetzt. Als Versionierungssystem wurde Git verwendet, und der Code wurde auf Github (<https://github.com/spoorey/hangman>) gehostet bzw. gesichert.

Um Framework, ORM etc. zu installieren und aktuell zu halten wurde Composer (<https://getcomposer.org/>), der am weitesten verbreitete Dependency Manager für PHP (ähnlich wie z.Bsp. NPM) verwendet.

Bei den einzelnen Tiers werden die im jeweiligen Tier verwendeten Tools, Vorgehen usw. genauer erklärt.

# Tiers

## Tier 1 – Presentation

Auf diesem Tier werden sämtliche Daten, die von den anderen Tiers aufbereitet bzw. gespeichert werden, dargestellt.

Dazu werden folgende Technologien verwendet.

* HTML5
* Javascript
* CSS

Um die Umsetzung des Projektes zu erleichtern werden folgende Libraries verwendet:

* LESS (<http://lesscss.org/>)
* jQuery (<https://jquery.org/>)
* jQuery UI (<http://jqueryui.com/>)
* Bootstrap (<http://getbootstrap.com/>)
* FontAwesome (<http://fontawesome.io/>)
* View- und Formular-Komponenten von Zend Framework 2 (<http://framework.zend.com/>)

LESS vereinfacht die Erstellung und Wartung von CSS stark. Im Gegensatz zu normalem CSS bietet es unter anderem Variablen, Verschachtelung und einfache Funktionen.  
Eine Ansammlung von LESS-files ist im Ordner /assets/ zu finden.

jQuery und jQuery UI bieten nützliche zusätzliche Funktionen gegenüber Standard-Javascript, welche vor allem bei der Darstellung nützlich sind. So wurden sowohl die Animationen in der Wörterliste (public/js/application/words.js) sowie die Asynchrone Aktualisierung des Wortes, falschen Buchstaben etc. im Spiel (public/js/application/game.js) mit jQuery UI, bzw. jQuery umgesetzt.

Bei der Spielansicht wird die Ansicht ebenfalls mit Javascript mit Daten abgefüllt, sowie die Eingabefelder und –Buttons für den zu ratenden Buchstaben generiert. Ebenfalls erfolgt beim Eingabefeld bereits auf Clientseite eine Eingabeüberprüfung.

Bootstrap bietet ein gutes Grundgerüst für CSS, das man einfach erweitern kann. Die meisten HTML-Komponenten in der Applikation wurden mit Bootstrap-Klassen gestylt.

Ausserdem werden von Bootstrap verwendeten LESS-Variabeln auch im eigenen LESS-Code verwendet, um eine einheitliche Farbgebung zu gewährleisten, etwa für die Untertitel der Wörterliste.

FontAwesome bietet eine breite Liste an Icons an. Da diese wie eine Schriftart verwendet werden können, lassen sich diese auch gut grösser oder in anderen Farben Darstellen (eine Funktion für die in der Applikation leider keine Anwendung gefunden wurde).

Sämtliche in der Anwendung verwendeten Icons stammen von FontAwesome.

Die View- und Formularkomponenten des Zend Framework 2 werden verwendet, um die Daten abzufüllen, und die Darstellung von sich wiederholenden Elementen zu erleichtern.

Ausserdem bietet das ZF2 auch eine sehr starke Formularkomponente. Damit können auf einfache Weise Formulare erstellt und validiert werden (mehr dazu im Tier 2).

Eine weitere Funktion sind die ViewHelper. Ein ViewHelper wird etwa verwendet, Zeitintervalle in Text umzuwandeln („Vor XY Stunden“).

Konfiguriert wird dieser in module/Application/Module.php auf Zeile 75:

'timeAgo' => **function**() {  
 $helper = **new** TimeAgoHelper();  
 **return** $helper;  
},

Der Code des ViewHelpers ist in der Datei module/Application/src/Application/View/Helper/TimeAgoHelper.php zu finden.

Von Zend Framework 2 wird unter anderem ein ViewHelper zum render von Formularfeldern bereitgestellt. Verwendet wird dieser unter anderem in der Datei module/Application/view/application/word/edit.phtml.

## Tier 2 – Webserver

Für den Webserver wird das bereits erwähnte Zend Framework 2 verwendet.

Dieses bietet –neben einer klaren Softwarearchitektur auf MVC-Basis- eine Fülle von Erweiterungen an, ohne diese eine Umsetzung eines solches Projekts mit PHP eine grössere Herausforderung wären.

Folgende Aufgaben werden auf dem zweiten Tier bearbeitet:

### Aufbereitung der Daten für den ersten Tier

Verschiedene Beispiele dafür finden sich in den Controllern im Ordner module/Application/src/Application/Controller.

### Ansteuerung des dritten Tiers

Ein Beispiel hierfür ist die Sortierung und Suche der Wörter in der Datei module/Application/src/Application/Controller/WordController.php ab Zeile 47.

Weitere Informationen folgen im Kapitel zum dritten Tier.

### Validierung von Formulardaten

Wie bereits erwähnt bietet Zend Framework eine starke Komponente, die sowohl das Erstellen als auch das Validieren von Formularen stark erleichtert.

Die Verwendung dieser kann mit dem Beispiel des Formulars zum Registrieren eines neuen Benutzers gut erklärt werden.

In module/Application/src/Application/Controller/UserController.php auf Zeile 69 wird dem Formular zum Registrieren ein InputFilter zugewiesen, welcher in module/Application/src/Application/InputFilter/UserInputFilter.php definiert ist.

Anschliessend verwendet das ZF2 diesen, um zu prüfen, ob sämtliche Eingaben gültig sind. Gleichzeitig werden ungültige Zeichen entfernt.

Sollten ungültige Eingaben getätigt worden sein, wird unter dem jeweiligen Formularfeld eine Fehlermeldung angezeigt.

Sollten die Validatoren von Zend Framework 2 nicht ausreichen, können auch Benutzerdefinierte Validatoren erstellt werden. Ein solcher findet sich in module/Application/src/Application/Validator/AllowedCharacters.php.

Wie der Name schon erahnen lässt, stellt dieser sicher, dass für ein Eingabefeld nur eine bestimmte Auswahl von Zeichen verwendet werden können.

Verwendet wird er im Inputfilter für die neuen Wörter in module/Application/src/Application/InputFilter/WordInputFilter.php auf Zeile 43.

### Authentifizierung

Zend Framework 2 bietet auch eine einfach umsetzbare Möglichkeit, eine sichere Authentifizierung einzurichten.

Aufgrund des dritten Tiers, musste der Authentifikationsadapter leicht angepasst werden. Der Adapter findet sich unter module/Application/src/Application/Authentication/Adapter.php. Dieser Adpater prüft Passwort und Benutzername und wird mithilfe einer factory (module/Application/src/Application/Service/Factory/AuthenticationFactory.php) dem Authentifikationsservice zugewiesen.

Die entsprechende Konfiguration findet sich in module/Application/config/module.config.php auf den Zeilen 122 und 125.

Die Passwörter werden mithilfe von BCrypt verschlüsselt.

In module/Application/Module.php auf Zeile 93 wird geprüft, ob die angeforderte Seite für den derzeit angemeldeten Usertyp verfübar ist.

Auf die Verwendung einer ACL wurde aufgrund der geringen Anzahl der zu schützenden Seiten verzichtet.

### Spiellogik

Die gesamte Spiellogik wird in module/Application/src/Application/Controller/GameController.php vor allem in der Methode updateAction umgesetzt.

Dadurch wird sichergestellt, dass die Daten, die dem User angezeigt werden, immer mit den Daten aus der Datenbank übereinstimmen, und das zu ratende Wort nie an den Client übertragen wird.

Die genannte updateAction generiert JSON, welches vom bereits erwähnten Javascript ausgewertet werden kann. Dieses JSON wird asynchron aufgerufen, wodurch während des Spielens nie ein erneutes Laden der Seite nötig ist.

## Tier 3 – Application Server

Für den dritten Tier werden der ORM von Doctrine 2 (<http://www.doctrine-project.org/projects/orm.html>) sowie einige Erweiterungen (<https://github.com/beberlei/doctrineextensions>) verwendet.

Doctrine 2 bietet eine Vielzahl an Funktionen. Die für dieses Projekt am wichtigsten bzw. meist verwendeten sind:

### Automatische Erstellung bzw. Aktualisierung der Datenbank anhand erstellter Entities.

In den Entites können durch Annotationen verschiedene Eigenschaften für Tabelle und Spalten festgelegt werden. Die Entities sind unter module/Application/src/Application/Entity zu finden.

Per Skript-Aufruf in der Konsole kann die Datenbank automatisch erstellt bzw. Aktualisiert werden (siehe /README.md)

### Einfache und Komplexere Möglichkeiten, Daten aus der Datenbank zu laden

Ein Beispiel für eine Einfache Abfrage ist in module/Application/src/Application/Controller/GameController.php auf Zeile 43 zu finden, wo sämtliche noch nicht abgeschlossene Spiele des derzeit angemeldeten Users sortiert nach Startzeit geladen werden.

Komplexere Abfragen können mittels eines QueryBuilders erstellt werden. Dieser wird zum Beispiel verwendet um in module/Application/src/Application/Controller/WordController.php ab Zeile 47 die Wörter anhand der Angegebenen Such- und Sortier-Informationen zu laden.

Bei der Wörterliste findet auch eine weitere Funktion von Doctrine 2 verwendung:

### Paginator

Um nicht alle Wörter aus der Datenbank laden zu müssen, wird in module/Application/src/Application/Controller/WordController.php ab Zeile 72 ein Paginator eingerichtet, und die gewünschte Seite geladen.

### Sicherheit

Doctrine 2 übernimmt Encoding, Escaping usw. um Sicherheitslücken wie etwa SQL-Injection zu verhindern. Dadurch wird dem Entwickler viel Programmierarbeit abgenommen, da gerade PHP auf diese Angriffe anfällig ist.

### Weitere Datentypen

Mit Doctrine 2 können zum Beispiel auch Arrays in einer MySQL Datenbank gespeichert werden.

Je nach Datenbank werden diese in ein passendes Format konvertiert. Bei MySQL ist dies etwa longtext.

Ein Beispiel ist das Attribut „guessedLetters“ der Entity „Game“ in module/Application/src/Application/Entity/Game.php ab Zeile 65.

### Beziehungen zwischen Entities

Doctrine 2 erleichtert das Arbeiten mit Beziehungen. So hat zum Beispiel das „User“ Entity eine Methode getGames() (und die „Game“ entity eine Methode getUser()), um auf die Spiele des Users zugreifen zu können zu finden in module/Application/src/Application/Entity/User.php auf Zeile 136. Man beachte auch die Annotation des games-Attributes der User-Entity.

### Einfache Integration in Zend Framework 2

Für Doctrine 2 ist eine Integrationslibrary für Zend Framework 2 verfügbar (<https://github.com/doctrine/DoctrineORMModule>).

Erfasst wurde das Integrationsmodul in config/application.config.php auf Zeile 11.

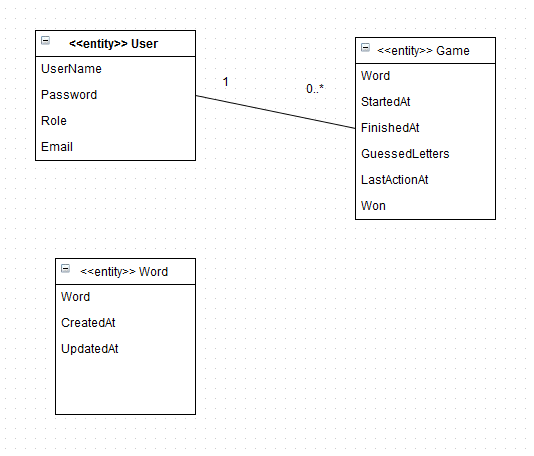
Anschliessend wird er Objectmanager, wenn ein Controller das entsprechende Interface implementiert, in module/Application/config/module.config.php auf Zeile 138 zugewiesen.

Die Konfiguration ist (nachdem die Applikation gemäss den Angaben in der Readme instaliert wurde) config/autoload/local.php hinterlegt.

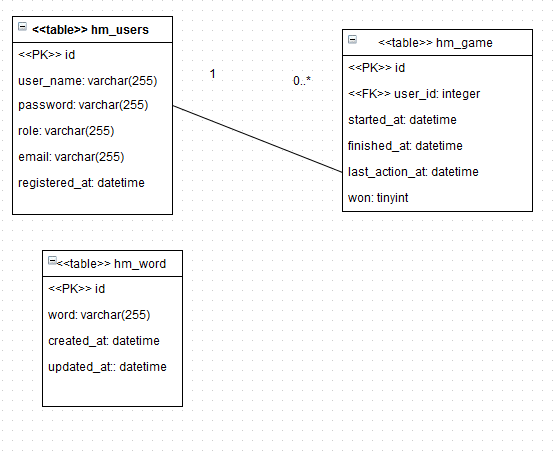
## Tier 4 – Datenbank

Als Datenbank wird MySQL verwendet.

### Konzeptionelles Datenmodell



### Relationales Datenmodell



### Anmerkungen

* Als prefix für die Tabellennamen wurde „hm\_“ gewählt
* Das Wort wird als String gespeichert, da es ansonsten zu Verwirrung beim Benutzer führen kann, wenn das Wort geändert wird, während das Spiel noch im Gang ist. Auch könnten bereits als verloren oder gewonnene Spiele so ungültig gespeichert sein.

## Austauschbarkeit der Tiers

### Tier 1 – Presentation

Ein austauschen dieses Tiers würde Zwangsweise auch zu Anpassungen im zweiten Tier führen.

Es wäre zum Beispiel möglich, den zweiten Tier als API zu gestalten, und beim ersten Tier mit Angular JS zu arbeiten.

### Tier 2 – Webserver

Dieser Tier ist der wohl am schwersten austauschbare. Eine Änderung hier würde zwangsweise auch zu Anpassungen im ersten Tier bzw. eine gänzliche Neugestaltung dessen notwendig machen. Ausserdem müsste je nach verwendeter Programmiersprache auch der dritte Tier ausgewechselt werden.

### Tier 3 – Application Server

Eine Ersetzung von Doctrine durch ein anderes oder ein selbst erstelltes ORM oder ähnliches Tool wäre ohne Anpassungen (ausserhalb der im vorherigen Kapitel erwähnten Konfigurationen) möglich, wenn dieses auf die gleiche Weise angesprochen werden kann, wie Doctrine. Da dies jedoch unwahrscheinlich ist, müssten sämtliche Doctrine-Aufrufe an den neuen dritten Tier angepasst werden.

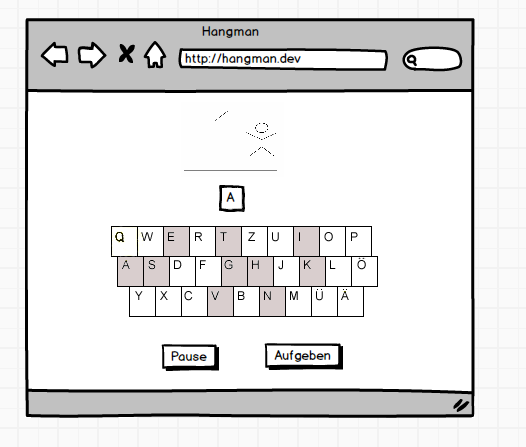
### Tier 4 – Datenbank

Ein anderer Datenbank-Typ kann in der Konfiguration von Doctrine hinterlegt werden. Sofern dieser von Doctrine unterstützt wird (<http://doctrine-dbal.readthedocs.org/en/latest/reference/configuration.html#driver>) und die RAND() Funktion unterstützt.

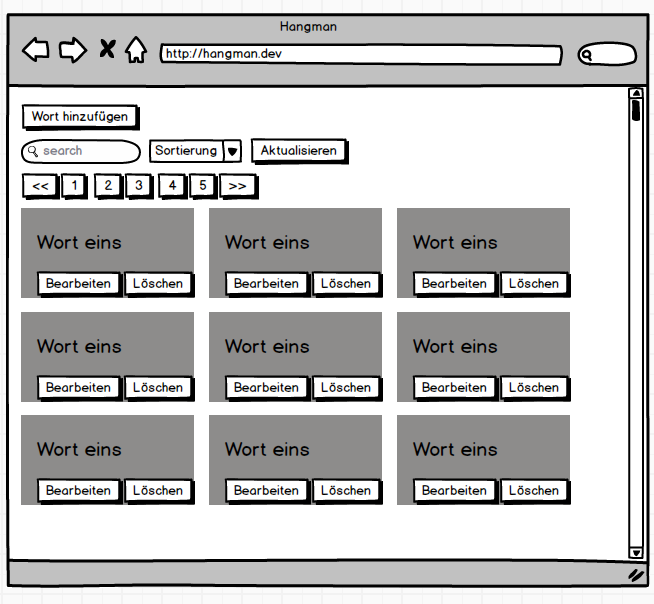
Sollte die RAND() Funktion nicht unterstützt werden, muss eine andere Möglichkeit zum Laden eines Zufälligen Wortes gefunden werden.

# GUI-Prototypen

## Spielinterface



## Wörteradministration



# Testfälle

**Webseite:** [**http://www.hangman.hayloft.it**](http://www.hangman.hayloft.it)

**Webapplikation:** Hangman

**Voraussetzungen:**Google Chrome oder Firefox

**Test des Logins und der Registration**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | Beschreibung | Eingabe | Ausgabe |
| 1 | Solange man nicht angemeldet ist, wird nur die Startseite angezeigt, mit einem Hinweise, sich einzuloggen oder sich zu registrieren. | Webseite ohne eingeloggt zu sein aufrufen | Startseite mit erwähnten hinweisen. |
| 2 | Bei falschen Eingaben im Registrierungsformular werden Fehlermeldungen angezeigt, und das Formular muss korrigiert werden. | * Zu kurzer oder langer Benutzername * Kein Benutzername * Eine ungültige E-Mail Adresse * Passwörter, die kürzer als 6 Zeichen sind * Passwörter, die nicht übereinstimmen | Formular mit Fehlermeldungen |
| 4 | Es dürfen sich nicht zwei User mit dem gleichen Benutzernamen registrieren | * Benutzername, der bereits verwendet wird | Formular mit Fehlermeldung |
| 5 | Bei korrekter Eingabe wird man zum Login weitergeleitet. | * Kein E-Mail * Oder ein gültiges E-Mail * Valider Benutzername und Passwörter | Formular zum Login |
| 6 | Login schlägt bei falsch-eingaben fehl | * Ungültiger Benutzername * Oder ein ungültiges Passwort | Formular zum Login, mit entsprechender Fehlermeldung |
| 7 | Bei korrekten Login Daten wird man zum Spiel weiteregeleitet | * Korrekte Anmeldedaten | * Ansicht der noch offenen Spiele |
| 8 | Normale User sehen keine Buttons zum Bearbeiten oder Löschen von Wörtern | Wörterliste, eingeloggt als normaler User | Keine Löschen oder Bearbeiten buttons |
| 9 | Administratoren sehen die Löschen und Bearbeiten Buttons | Wörterliste, eingeloggt als normaler User | Die Buttons werden angezeigt |
| 10 | Nicht eingeloggte User haben nur Zugriff auf Übersicht, Login, Registration und Wörterliste | URL zum Bearbeiten von Wörtern oder zum Spiel | Das Login-Formular wird angezeigt |
| 11 | Normale User können keine Wörter bearbeiten oder Löschen | URL zum Bearbeiten oder Löschen von Wörtern eingeben | Das Login-Formular wird angezeigt |

**Test der Wörterliste**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | Beschreibung | Eingabe | Ausgabe |
| 12 | Die Wörterliste wird angezeigt | Wörterliste aufrufen | Wörterliste |
| 13 | Die Wörterliste lässt sich durchsuchen und sortieren | Sortierung, Suchbegriff, Sortier-Richtung | Sortierte Wörterliste, ggf. gefiltert |
| 14 | Beim Hinzufügen von Wörtern wird die Eingabe auf zu kurze oder ungültige Wörter geprüft | Formular zum Hinzufügen eines Wortes aufrufen, eingeloggt als Administrator  Danach   * Wort mit nicht zugelassenen Zeichen * Wort, das kürzer als 6 Zeichen ist | Formular mit entsprechender Fehlermeldung |
| 15 | Wörter lassen sich hinzufügen | Formular zum Hinzufügen eines Wortes aufrufen, eingeloggt als Administrator  Danach ein gültiges Wort | Wort wird gespeichert; Weiterleitung zur Wörterliste |
| 16 | Wörter lassen sich bearbeiten | Formular zum Bearbeiten eines Wortes aufrufen, eingeloggt als Administrator  Danach ein gültiges Wort | Wort wird gespeichert; Weiterleitung zur Wörterliste |
| 16 | Wörter lassen sich Löschen | Löschen Knopf eines Wortes anklicken | Wort wird gelöscht |

**Test des Spiels**

|  |  |  |  |
| --- | --- | --- | --- |
| 17 | Bereits gestartete Spiele werden angezeigt | Spiel in der Navigationsleiste anklicken | Liste mit bereits gestarteten Spielen & links um diese fortzusetzen |
| 18 | Es lässt sich ein neues Spiel starten | „Neues Spiel“ anklicken | Ein neues Spiel wird gestartet |
| 19 | Es lässt sich ein bereits bestehendes Spiel fortsetzen | „Weiterspielen“ anklicken | Das Spiel wird fortgesetzt |
| 20 | Die Anzeige des Spielinterfaces ist korrekt | Spiel starten | Das Spielinterface wird korrekt geladen und angezigt |
| 21 | Korrekt geratene Buchstaben werden angezeigt | Korrekten Buchstaben anklicken oder eingeben | Der Korrekte Buchstaben wird grün hinterlegt angezeigt |
| 22 | Falsch geratene Buchstaben | Falschen Buchstaben anklicken oder eingeben | Der Falsche Buchstabe wird rot hinterlegt angezeigt |
| 23 | Wenn alle Buchstaben erraten wurden, ist das Spiel gewonnen | Alle korrekten Buchstaben | Mitteilung mit Links; Spiel wird nicht mehr in Liste der gestarteten Spiele angezeigt |
| 24 | Nach 7 falschen Eingaben ist das Spiel verloren | 7 falsche Buchstaben | Animation des Bildes, Nachricht mit Links, das Wort; Spiel wird nicht mehr in Liste der gestarteten Spiele angezeigt |
| 25 | Ungültige Eingaben werden abgefangen | Ein ungültiges Zeichen im Eingabefeld für den Buchstaben | Fehlermeldung, Zeichen wird nicht geraten |
| 26 | Ungültige Eingabe über die Javascript Konsole | guessLetter(‚=‘); in Javascript konsole (oder ein anderes Ungültiges Zeichen) | Zeichen wird nicht geraten |
| 27 | Automatische Weiterleitung zum Spiel, wenn es gestartet wurde | Spiel starten, dann auf „Spiel“ in der Navigation klicken | Spielinterface |
| 28 | Das Spiel lässt sich pausieren | „Spiel pausieren“ anklicken | Spiel wird pausiert, erscheint aber noch in der Liste offener Spiele |
| 29 | Das Spiel lässt sich aufgeben | „Spiel aufgeben“ anklicken | Spiel wird als verloren markiert und erscheint nicht mehr in der Liste offener Spiele |

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Resultat** | **Bemerkung** | **Unterschrift** |
| 1 | Erfolgreich |  | d.spoerri |
| 2 | Erfolgreich |  | d.spoerri |
| 4 | Erfolgreich |  | d.spoerri |
| 5 | Erfolgreich |  | d.spoerri |
| 6 | Erfolgreich |  | d.spoerri |
| 7 | Erfolgreich |  | d.spoerri |
| 8 | Erfolgreich |  | d.spoerri |
| 9 | Erfolgreich |  | d.spoerri |
| 10 | Erfolgreich |  | d.spoerri |
| 11 | Erfolgreich |  | d.spoerri |
| 17 | Erfolgreich |  | d.spoerri |
| 18 | Erfolgreich |  | d.spoerri |
| 19 | Erfolgreich |  | d.spoerri |
| 20 | Erfolgreich |  | d.spoerri |
| 21 | Erfolgreich |  | d.spoerri |
| 22 | Erfolgreich |  | d.spoerri |
| 23 | Erfolgreich |  | d.spoerri |
| 24 | Erfolgreich |  | d.spoerri |
| 25 | Erfolgreich |  | d.spoerri |
| 26 | Erfolgreich |  | d.spoerri |
| 27 | Erfolgreich |  | d.spoerri |
| 28 | Erfolgreich |  | d.spoerri |
| 29 | Erfolgreich |  | d.spoerri |

## Abnahmeprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| Auftraggeber: | BBBaden Andreas Blaser | Auftragnehmer: | David Spörri |
| Wiesenstrasse 32 | Tannenweg 6 |
| 5400 Baden | 5702 Niederlen |
| 056 222 02 06 | David.spoerri@stud.bbbaden.ch |
| Projektbezeichnung | Modulprüfung 151 Datenbanken in Webauftritt einbinden | | |
| Projektbeginn | 09.06.2015 | Projektende | 27.09.2015 |
| Datum | 27.09.2015 | Ort | Baden |
| Vertragsgrundlage | Kein Pflichtenheft vorhanden | | |
| Garantiebeginn | 28.09.2015 | Garantiedauer | 2 Wochen |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Beschreibung** | **+** | **-** |
| 1 | Solange man nicht angemeldet ist, wird nur die Startseite angezeigt, mit einem Hinweise, sich einzuloggen oder sich zu registrieren. | x |  |
| 2 | Bei falschen Eingaben im Registrierungsformular werden Fehlermeldungen angezeigt, und das Formular muss korrigiert werden. | x |  |
| 4 | Es dürfen sich nicht zwei User mit dem gleichen Benutzernamen registrieren | x |  |
| 5 | Bei korrekter Eingabe wird man zum Login weitergeleitet. | x |  |
| 6 | Login schlägt bei falsch-eingaben fehl | x |  |
| 7 | Bei korrekten Login Daten wird man zum Spiel weiteregeleitet | x |  |
| 8 | Normale User sehen keine Buttons zum Bearbeiten oder Löschen von Wörtern | x |  |
| 9 | Administratoren sehen die Löschen und Bearbeiten Buttons | x |  |
| 10 | Nicht eingeloggte User haben nur Zugriff auf Übersicht, Login, Registration und Wörterliste | x |  |
| 11 | Normale User können keine Wörter bearbeiten oder Löschen | x |  |
| 12 | Die Wörterliste wird angezeigt | x |  |
| 13 | Die Wörterliste lässt sich durchsuchen und sortieren | x |  |
| 14 | Beim Hinzufügen von Wörtern wird die Eingabe auf zu kurze oder ungültige Wörter geprüft | x |  |
| 15 | Wörter lassen sich hinzufügen | x |  |
| 16 | Wörter lassen sich bearbeiten | x |  |
| 16 | Wörter lassen sich Löschen | x |  |
| 17 | Bereits gestartete Spiele werden angezeigt | x |  |
| 18 | Es lässt sich ein neues Spiel starten | x |  |
| 19 | Es lässt sich ein bereits bestehendes Spiel fortsetzen | x |  |
| 20 | Die Anzeige des Spielinterfaces ist korrekt | x |  |
| 21 | Korrekt geratene Buchstaben werden angezeigt | x |  |
| 22 | Falsch geratene Buchstaben | x |  |
| 23 | Wenn alle Buchstaben erraten wurden, ist das Spiel gewonnen | x |  |
| 24 | Nach 7 falschen Eingaben ist das Spiel verloren | x |  |
| 25 | Ungültige Eingaben werden abgefangen | x |  |
| 26 | Ungültige Eingabe über die Javascript Konsole | x |  |
| 27 | Automatische Weiterleitung zum Spiel, wenn es gestartet wurde | x |  |
| 28 | Das Spiel lässt sich pausieren | x |  |
| 29 | Das Spiel lässt sich aufgeben | x |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Mängel | | | |
| Nr.: | Punkt: | Beschreibung: | Frist: |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Bemerkungen: |  | | |
| Vorbehalt.: | Die Tests wurden in Google Chrome und Firefox ausgeführt. Für andere Browser besteht keine vollständige Funktionsgarantie | | |
| Unterschrift / Datum Auftraggeber |  | Unterschrift / Datum  Auftragnehmer | David Spörri  27.09.2015 |