

---

# CloneCademy

---

## Qualitätssicherungsdokument

Gruppe 12: Ilhan Simsiki <ilhan.simsiki@stud.tu-darmstadt.de>  
Leonhard Wiedmann <leonhard.wiedmann@stud.tu-darmstadt.de>  
Tobias Huber <tobias.huber@stud.tu-darmstadt.de>  
Claas Völcker <c.voelcker@stud.tu-darmstadt.de>

Teamleiter: Alexander Nagl <alexander.nagl@t-online.de>

Auftraggeber: iGEM-Team TU Darmstadt  
vertreten durch Thea Lotz <lotz@bio.tu-darmstadt.de>  
Fachbereich Biologie

Abgabedatum: 14.07.2017

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Bachelor-Praktikum SoSe 2017  
Fachbereich Informatik

---

---

## Contents

---

<b>1. Einleitung</b>	<b>4</b>
<b>2. Qualitätsziele</b>	<b>5</b>
2.1. Datensicherheit (Security) . . . . .	5
2.2. Bedienbarkeit . . . . .	6
2.3. Veränderbarkeit . . . . .	7
<b>A. Konkrete Ausführungen des QS-Maßnahmenkatalogs</b>	<b>10</b>
A.1. Ablaufplan der QS-Maßnahmen . . . . .	10
A.1.1. Wöchentlich . . . . .	10
A.1.2. Kalender . . . . .	10
A.2. Checklisten . . . . .	11
A.2.1. Checkliste Bedienbarkeit . . . . .	11
A.2.2. Checkliste Datensicherheit . . . . .	11
A.2.3. Checkliste Veränderbarkeit . . . . .	12
A.3. Informelle Beschreibung der Nutzungsrollen . . . . .	13
A.3.1. Nutzer*in . . . . .	13
A.3.2. Moderator*in . . . . .	13
A.3.3. Administrator*in . . . . .	13
A.4. Rahmen der Wikidokumentation . . . . .	13
A.5. 1. Nutzerstudie . . . . .	14
<b>B. Einleitung - Bericht und Anmerkung zum QS Prozess</b>	<b>17</b>
B.1. Änderungen des Prozesses während des Semesters . . . . .	17
<b>C. Nachweis über den QS Prozess pro Iteration</b>	<b>18</b>
<b>D. Userstories</b>	<b>20</b>
D.1. Iteration bis . . . . .	20
D.2. Iteration bis . . . . .	20
D.3. Iteration bis . . . . .	21
D.4. Iteration bis . . . . .	23
D.5. Iteration bis . . . . .	24
D.6. Iteration bis . . . . .	25
D.7. Iteration bis . . . . .	26
D.8. Iteration bis . . . . .	27
D.9. Iteration bis . . . . .	28
D.10. Iteration bis . . . . .	30
D.11. Iteration bis . . . . .	31
D.12. Iteration bis . . . . .	32
D.13. Iteration bis . . . . .	32

---

D.14.Iteration bis	32
D.15.Iteration bis	35
D.16.Iteration bis	36
<b>E. Protokolle</b>	<b>37</b>
<b>F. Abschließende Bemerkungen des Teams zum Projekt</b>	<b>38</b>

---

## Anmerkungen und Hinweise

---

Die Verfasser dieses Dokumentes verwenden als Maßnahme für die Gleichstellung aller Personen die sogenannte "gendergerechte Sprache". Als Zeichen der Inklusion aller Geschlechter wird der Stern (\*) oder eine Verlaufsform (*Nutzung* statt *Nutzer*) verwendet. Von dieser Regelung wird nur abgewichen, wenn entweder alle Mitglieder einer Gruppe sich ein und demselben Geschlecht zugehörig fühlen (z.B. im Entwicklerteam) oder die Vorgaben der Veranstalter\*innen keine Alternative zulassen (z.B. auf dem Deckblatt).

---

## 1 Einleitung

---

*CloneCademy* ist ein Projekt für das iGEM-Team der TU Darmstadt. Die *international Genetically Engineered Machine competition* (iGEM) ist ein internationaler Wettbewerb für Studierende auf dem Gebiet der synthetischen Biologie. Dieser wird seit 2003 von der iGEM-Foundation veranstaltet.

Im Rahmen des Wettbewerbs wird eine Online-Lernplattform für das iGEM-Team der TU Darmstadt erstellt. Das Ziel dieser Plattform ist es, durch interaktive Unterrichtseinheiten Prinzipien der Molekularbiologie und der synthetischen Biologie zu erlernen und sowohl eigene Lernfortschritte, als auch die anderer Teams begutachten zu können. Darüber hinaus soll es auch anderen Interessierten (z.B. andere iGEM-Teams, Lehrende an Universitäten und Schulen, etc.) möglich sein, eigene Inhalte einzupflegen und zur Verfügung zu stellen.

Als Kernfunktionalität bietet die Plattform Nutzer\*innen die Möglichkeit, sich zu registrieren und einen Account anzulegen. Registrierte Nutzer\*innen können bereitgestellte Aufgaben bearbeiten und Feedback zu ihren Antworten erhalten. Zusätzlich können Nutzer\*innen mit Moderationsrechten neue Aufgaben in das System einpflegen und bereits bestehende überarbeiten. Die bereitgestellten Aufgabentypen sind Multiple-Choice-, Drag-And-Drop- und Lückentextaufgaben. Außerdem wird im Rahmen des vereinbarten Ziels der *Veränderbarkeit* die Möglichkeit geschaffen, nach Ende des Projektes auch noch weitere Aufgabentypen in die Plattform einzubauen.

Um einen Überblick über ihren bisherigen Lernerfolg zu bekommen, haben Nutzer\*innen die Möglichkeit, Statistiken zu ihrer Nutzung der Plattform einzusehen. Den Auftraggeber\*innen geben diese Statistiken auch Einblick in die Nutzung und die Qualität ihrer Inhalte, sodass diese weiter verbessert werden können.

Das Ziel des Projektes ist eine voll funktionsfähige Webanwendung für die aktuellen Versionen der Browser *Firefox* und *Chrome* auf Desktop- und Laptoprechner bereitzustellen. Eine eventuelle Erweiterung der Plattform, um zum Beispiel auch mobile Endgeräte zu unterstützen, wird durch die Implementierung einer REST-Schnittstelle ermöglicht.

---

## 2 Qualitätsziele

---

### 2.1 Datensicherheit (Security)

---

Im Rahmen des Projekts CloneCademy wird eine Webanwendung entwickelt, auf welche über das Internet zugegriffen werden kann. Daher ist die Sicherung gegen unbefugten Zugriff und unautorisierte Änderung der Daten in diesem Projekt ein wichtiges Qualitätsziel. Da CloneCademy sowohl persönliche Daten der Nutzer\*innen als auch Metadaten über die Nutzung der Plattform und die Inhalte der einzelnen Lerneinheiten persistent speichert, ist es essentiell, dass Internetnutzer\*innen diese Daten nicht verändern oder einsehen können, solange sie dazu nicht die benötigten Rechte besitzen.

Die größte Bedrohung geht von bekannten Webangriffen und Fehlkonfigurationen im Backend einer Webseite aus<sup>1</sup>. Während dieses Projekts wird die Plattform daher gegen die wichtigsten Sicherheitslücken einer Webanwendung gesichert. Dies sind vor allem mögliche Schwachstellen in der Nutzungsoberfläche und der REST-Schnittstelle.

Die betrachteten Angriffsvektoren sind:

- Möglichkeiten zur Ausführung fremden Codes (Injection & Cross-Site Scripting)
- Fehlerhafte Zugriffskontrolle
- Nutzung von Komponenten mit bekannten Schwachstellen

Um die Sicherheit der Anwendung zu gewährleisten, wird ein mehrschrittiger Plan befolgt.

Ein Entwickler des Teams wurde zum Sicherheitsbeauftragten ernannt, der die korrekte Durchführung aller genannten Maßnahmen sicherstellt. Seine Aufgabe ist es, die Einhaltung aller Programmierrichtlinien durchzusetzen und Tests zur Validierung der Schutzziele durchzuführen. Die genaue Vorgehensweise wird im Folgenden detailliert ausgeführt.

Um klare Zugriffsbeschränkungen umsetzen zu können, wurden Nutzungsrollen für das Projekt definiert (Administrator\*in, Moderator\*in und Nutzer\*in)<sup>2</sup>. In jeder User Story wird festgehalten, ob und welche Zugriffsbeschränkungen einzelne Module der Webseite oder Daten besitzen. Um die Verwaltung der Rechte einzelner Nutzungsrollen zu ermöglichen, werden die zur Verfügung gestellten Authentifizierungswerkzeuge der genutzten Frameworks (Django, Django REST) verwendet. Die korrekte Implementierung der Rechteverwaltung wird im Rahmen der Security Reviews überprüft.

---

<sup>1</sup> Unsere Einschätzung relevanter Sicherheitslücken basiert auf den Empfehlungen des Open Web Application Security Project (OWASP)  
Link zum Download: <https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010%20-%202017%20RC1-English.pdf>

<sup>2</sup> Detaillierte Beschreibungen der Rollen finden sich im Anhang

---

Während der Implementierung halten alle Entwickler die Best Practices im Web Development ein. Als Referenz dafür werden die Handreichungen des Open Web Application Security Project (OWASP)<sup>3</sup> ausgeführt. In den wöchentlichen internen Code-Reviews wird die Qualität des Codes hinsichtlich dieser Richtlinien überprüft. Hierfür wird eine Checkliste<sup>4</sup> geführt, anhand der die Einhaltung dieser Richtlinien überprüft wird.

Um die Qualität des Codes im Python-basierten Backend automatisiert zu testen, wird das verbreitete Werkzeug *bandit*<sup>5</sup> verwendet. Als statisches Analyse-Werkzeug für das Frontend wird *ng2lint*<sup>6</sup> verwendet. Beide Programme unterstützen die Code-Review, indem die gelieferten Meldungen als Grundlage für eine detaillierte Analyse des Codes genutzt werden. Um eine dauerhaft hohe Qualität des Codes zu gewährleisten, werden die Analyse-Tools automatisiert bei einem Push auf dem Development-Branch des verwendeten Versionskontrollsystems Git verwendet. Alle gefundenen Schwachstellen werden vor der Präsentation der User Story zur Abnahme behoben. Dies ist die Aufgabe der entsprechenden Verantwortlichen für die User Story. Sollte dies nicht möglich sein, wird eine schriftliche Begründung verfasst und eine realistische Einschätzung über die Risiken der gefundenen Fehler an die Auftraggeber\*innen gemeldet. Diese entscheiden, ob die Risiken tragbar sind und die User Story trotz der Sicherheitslücke als erfolgreich gemeldet wird.

Um die tatsächliche Sicherheit des Endproduktes gegen externe Angriffe zu zeigen, wird ein automatisiertes Penetrations-Werkzeug verwendet. Das OWASP empfiehlt hierfür das *Zed Attack Proxy Project*<sup>7</sup>, welches eine Reihe bekannter Angriffe auf die Plattform ausführt und die gefundenen Schwachstellen meldet. Der Sicherheitsbeauftragte führt diese Tests spätestens eine Woche vor einem geplanten Release durch und meldet die Ergebnisse an die jeweiligen Entwickler des betroffenen Moduls, damit diese vor dem Release behoben werden können. Ist eine Behebung nicht möglich, werden die sicherheitsrelevanten Gefahren dieser Lücke zusammen mit möglichen weiteren Schritten zum Schutz der Anwendung im Betrieb an die Auftraggeber\*innen gemeldet.

---

## 2.2 Bedienbarkeit

---

CloneCademy ist eine Lernplattform und steht als solche einer sehr heterogenen Gruppe an Nutzer\*innen zur Verfügung. Daher muss die Plattform für alle Benutzer\*innen einfach und intuitiv bedienbar sein. Das heißt, dass eine Bedienung ohne vorherige Einarbeitung möglich ist und komplexe Aufgaben mit einer Erklärung versehen sind. Um dies zu gewährleisten, wird das Design und der Aufbau aller Bereiche der Webseite nach diesem Kriterium evaluiert.

Auch für dieses Ziel wurde ein Mitglied des Teams benannt, um Nutzerstudien durchzuführen und die Einhaltung aller Richtlinien durchzusetzen.

---

<sup>3</sup> [https://www.owasp.org/images/0/08/OWASP\\_SCP\\_Quick\\_Reference\\_Guide\\_v2.pdf](https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf)

<sup>4</sup> Siehe Anhang

<sup>5</sup> <https://wiki.openstack.org/wiki/Security/Projects/Bandit>

<sup>6</sup> <https://www.npmjs.com/package/ng2lint>

<sup>7</sup> [https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

---

Um das oben formulierte Ziel zu erreichen, verwenden wir als Grundlage für das Design der Seite das Material Design<sup>8</sup> für Angular2. Die Verwendung von standardisierten Komponenten sorgt dafür, dass die Webseite einen einheitlichen und übersichtlichen Aufbau hat. Um beispielsweise Knöpfe und andere Schaltflächen übersichtlicher zu gestalten, werden diese mit Icons<sup>9</sup> versehen. Bei der wöchentlichen internen Code-Review prüft der Beauftragte für dieses QS-Ziel anhand der Design-Checkliste<sup>10</sup> bei allen veränderten oder neuen Bereichen der Webseite, ob sie den Richtlinien eines einheitlichen Designs entsprechen. Sollte dies nicht der Fall sein, wird die User Story den Auftraggeber\*innen nicht präsentiert, sondern in der nächsten Iteration überarbeitet und erneut geprüft.

Da die Bedienbarkeit der Oberfläche nicht objektiv gemessen werden kann, werden zusätzlich zu den Designrichtlinien Nutzerstudien durchgeführt. In den Studien setzen sich die Proband\*innen ohne vorherige Erklärung mit der Plattform auseinander und werden darum gebeten, verschiedene Aufgaben zu erfüllen<sup>11</sup>. Während der Studie wird das Verhalten der Nutzer\*innen mit Hilfe eines Screen-Capture-Programms erfasst, um später zu überprüfen, ob die Aufgaben zielgerichtet gelöst werden konnten. Zusätzlich wird die persönliche Meinung der Proband\*innen nach der Studie mit einem Fragebogen<sup>12</sup> erfasst und ausgewertet.

Um abschließend ein, von einer Vielzahl unterschiedlicher Nutzer\*innen gut bedienbares, Produkt zu haben, werden die Nutzerstudien in mehreren Iterationen und mit unterschiedlichen Gruppen durchgeführt. Verbesserungsvorschläge aus den vorherigen Studien können so direkt wieder getestet werden. Die Nutzerstudien erfolgen mindestens eine Woche vor einem geplanten Release oder wenn die Auftraggeber\*innen eine Zwischenevaluation des Designs wünschen.

Nach jeder Nutzerstudie wird ein Maßnahmenkatalog mit den Problemen und Verbesserungsvorschlägen der Proband\*innen erstellt, welche ein dafür benannter Entwickler bis zum nächsten Treffen mit den Auftraggeber\*innen umsetzt. Diese können dann die Designänderungen überprüfen und erst durch die Abnahme der Designänderungen ist die Iteration der Studie abgeschlossen.

---

## 2.3 Veränderbarkeit

---

Für die Webanwendung CloneCademy ist es wichtig, dass sie nach Abschluss des Projekts noch veränderbar ist. Es muss für weitere Entwicklerteams möglich sein, sowohl neue Inhalte in die Datenbank einzupflegen, als auch den Quellcode der Webseite selbst erweitern und verändern zu können, da die Auftraggeber\*innen planen, die Webseite nach dem Abschluss des Bachelorprojektes eigenständig weiterzuentwickeln. Um diese Veränderbarkeit zu gewährleisten, wird

---

<sup>8</sup> <https://material.angular.io/>

<sup>9</sup> <https://material.io/icons/>

<sup>10</sup> siehe Anhang

<sup>11</sup> Ablauf siehe Anhang

<sup>12</sup> siehe Anhang



---

im Projekt auf folgende Punkte geachtet:

- hohe Qualität des Quellcode
- ausführliche Kommentare
- externe Dokumentation
- ausführliche Testabdeckung des Codes

Es wurde ein Verantwortlicher für das Qualitätsziel benannt, der Ansprechpartner für alle Rücksprachen oder Fragen ist und dafür sorgt, dass die Qualität der oben aufgeführten Bereiche, wie unten beschrieben, sichergestellt wird.

**Qualität des Quellcodes** Grundlegend werden während der Entwicklung der Software die Styleguides der verwendeten Programmiersprachen und Frameworks umgesetzt. Diese sind der *Angular Style Guide*<sup>13</sup> für das Frontend und *PEP 8*<sup>14</sup> für das Backend. Um die Einhaltung dieser beiden Richtlinien zu überprüfen, werden bei jedem Push auf dem Development-Branch statische Analyse-Tools verwendet, welche überprüfen, ob alle Richtlinien umgesetzt wurden.

Die verwendeten Tools sind *pep8*<sup>15</sup>, welches die Einhaltung der generellen Python Richtlinien überprüft, *django-lint*<sup>16</sup>, welches Framework spezifische Fehler markiert (z.B. Verwendung von veralteten Methoden) und *ng2lint*<sup>17</sup>, welches den Frontend-Code überprüft. Alle Fehler, die von diesen Tools gefunden wurden, werden bis zum nächsten internen Gruppentreffen von demjenigen Entwickler behoben, der für die User Story verantwortlich war, bevor der Code den Auftraggeber\*innen zur Abnahme präsentiert wird.

**Kommentare** Jede Methode und jede Klasse im Code erhält einen Kommentar. Zusätzlich werden unklare Stellen im Code (z.B. komplexe Steuermechanismen oder Abfragen) mit eigenen Erklärungen versehen. Alle Kommentare werden in Englisch verfasst.

Bei der Code-Review überprüft der Beauftragte anhand der Veränderbarkeits-Checkliste<sup>18</sup>, ob alle Klassen und Methoden kommentiert sind und ob die Kommentare das Verstehen des Codes unterstützen. Den Auftraggeber\*innen werden nur User Stories mit vollständig dokumentiertem Code zur Abnahme präsentiert.

**Wiki** Das Projekt verwendet zur klaren Trennung zwischen Back- und Frontend eine sogenannte REST-API. Diese stellt eine Schnittstelle dar, über welche die Daten des Backends abgerufen und gegebenenfalls verändert oder erweitert werden können. Diese wird deshalb gesondert dokumentiert, da eine korrekte Implementierung und Nutzung Grundlage für die Interaktion zwischen Front- und Backend ist.

---

<sup>13</sup> <https://angular.io/guide/styleguide>

<sup>14</sup> <https://www.python.org/dev/peps/pep-0008/>

<sup>15</sup> <https://pypi.python.org/pypi/pep8>

<sup>16</sup> <https://pypi.python.org/pypi/django-lint>

<sup>17</sup> <https://www.npmjs.com/package/ng2lint>

<sup>18</sup> siehe Anhang

---

Um die Schnittstelle zentral zu dokumentieren, wird ein Wiki geführt, in dem alle Funktionen der Schnittstelle zwischen Backend und Frontend definiert sind. Ist für eine User Story eine neue Funktion nötig, wird diese von allen Entwicklern zusammen entworfen und dann im Wiki dokumentiert.

Um sicherzustellen, dass die Dokumentation der Schnittstelle vollständig ist, überprüft der Beauftragte spätestens eine Woche vor Release, ob alle Funktionen aufgeführt sind und ob die Dokumentation den vorgegebenen Rahmen<sup>19</sup> erfüllt. Bevor das Wiki nicht vollständig ist, wird die Software nicht zum Release freigegeben.

**Testabdeckung** Bei Codeerweiterung muss bereits bestehender Code weiterhin funktionieren. Um dies zu verifizieren, wird eine umfangreiche Testabdeckung des Codes sichergestellt. Das heißt, dass eine vollständige Function Coverage und eine Statement Coverage von mindestens 80% erreicht wird. Eine Überprüfung der Abdeckung erfolgt im Backend durch das mitgelieferte Testframework des Django-Projekts. Im Frontend werden Tests mit Hilfe des Frameworks Selenium<sup>20</sup> durchgeführt.

Die Function Coverage wird vor jeder internen Code-Review sichergestellt. Die Tests der API Schnittstellen werden bei deren Definition geschrieben, damit sie den jeweiligen Entwicklern als Vorlage für eine korrekte Implementierung dienen können. Da eine umfassende Statement Coverage während des Projektes nicht zu jeder Zeit implementiert werden kann, ist diese zu jedem Release anzufertigen. Damit wird gezeigt, dass der Code stabil ist und die Tests können als Basis für die Implementierung der kommenden User Stories verwendet werden.

Eine User Story wird erst zur Abnahme präsentiert, wenn für alle Funktionen Tests vorliegen. Ein Release wird erst freigegeben, wenn eine Code Coverage von mindestens 80% erreicht ist. Die Überprüfung dieser Regeln erfolgt durch den, für das QS-Ziel zuständigen Entwickler.

---

<sup>19</sup> siehe Anhang

<sup>20</sup> <http://www.seleniumhq.org/>

---

## A Konkrete Ausführungen des QS-Maßnahmenkatalogs

---

---

### A.1 Ablaufplan der QS-Maßnahmen

---

---

#### A.1.1 Wöchentlich

---

Das Entwicklerteam trifft sich wöchentlich, um den aktuellen Stand des Projektes zu besprechen. Für die Qualitätssicherung werden dabei folgende Aufgaben durchgeführt:

- Sichten aller Reports der automatisierten Tools
- Überprüfen des aktuellen Designs
- Überprüfen der Checklisten
- Beschluss, welche User Stories zur Abnahme präsentiert werden

Alle hierbei gefundenen Mängel werden von den jeweiligen, für den Code im Rahmen einer User Story verantwortlichen Entwickler behoben. Eine User Story wird erst präsentiert, wenn der Code keine Mängel mehr aufweist.

---

#### A.1.2 Kalender

---

- Anfang August: geplanter erster Release auf dem Server des iGEM-Teams
- Anfang September: geplanter zweiter Release auf dem Server des iGEM-Teams
- Ende September: endgültiger Release der finalen Software auf dem Server des iGEM-Teams

Eine Woche vor jedem Release:

- Deadline der Nutzerstudie
- Security Test mit dem *ZED Attack Tool*
- Prüfung der Wiki auf Vollständigkeit

Alle hierbei gefundenen Mängel werden bis zum jeweiligen Release behoben. Zur Behebung wird ein dedizierter Entwickler benannt.

---

## A.2 Checklisten

---

---

### A.2.1 Checkliste Bedienbarkeit

---

Nach jeder User Story muss für die Elemente folgende Checkliste beachtet werden:

- |  |  |
|--|--|
| <input type="checkbox"/> Wurden nur Elemente des Material Designs verwendet?   | <input type="checkbox"/> Ist wichtiger Text von Nebeninformatio-<br>nen abgetrennt und deutlich sichtbar?                                      |
| <input type="checkbox"/> Sind Schaltflächen optisch von ihrem<br>Hintergrund abgehoben und im gleichen<br>Format gehalten, wie andere der selben<br>Art? | <input type="checkbox"/> Sind die Schaltflächen richtig beschriftet<br>und ist durch die Beschriftung erkennbar,<br>welche Funktion sie haben? |
| <input type="checkbox"/> Sind Buttons durch ihre Position direkt<br>ersichtlich?   | <input type="checkbox"/> Sind neue Elemente optisch gut unter-<br>scheidbar von vorhandenen Elementen?   |
|  | <input type="checkbox"/> Ist alles in der Mindestauflösung<br>(1024x768) erkennbar?  |

---

### A.2.2 Checkliste Datensicherheit

---

Bei der Codereview wird von folgendem Bedrohungsmodell ausgegangen:

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>• Angreifende: Es wird davon ausgegangen, dass böswillige Nutzer*innen der Plattform bei der Beantwortung der Fragen unerlaubte Methoden benutzen möchten, sowie dass Dritte die gespeicherte Nutzerdaten auslesen und verändern wollen.</li><li>• Angriffsoberfläche: Als Oberfläche wird die API und das Frontend betrachtet.</li></ul> | <ul style="list-style-type: none"><li>• Mögliche Angriffe: Es werden die gebräuchlichsten Angriffsformen wie zum Beispiel XSS, CSRF sowie Code- und SQL-Injection erwartet.</li><li>• Als grundlegende Sicherheitsmaßnahmen wird die korrekte Implementierung von Django, Django REST und Angular2 verwendet.</li></ul> |
|---|---|

Fragen zu jedem Codeabschnitt:

- |  |  |
|--|--|
| <input type="checkbox"/> HTML Input: Wird jeglicher Input korrekt nach Cross-Site-Scripts durchsucht, bevor er als sicher markiert wird? | <input type="checkbox"/> Ist jede Schnittstelle mit passender Authentifizierung und Autorisierung versehen, sowie gegebenenfalls auch mit einer Anfragenbegrenzung (Throttling)? |
| <input type="checkbox"/> Sind die Django-Features zum Schutz vor Cross Site Request Forgery (CSRF) aktiviert?                            | <input type="checkbox"/> Werden alle Nutzereingaben nach Fehlern gefiltert?  |

- 
- Sind alle Security Probleme, die bei der softwarebasierten Analyse aufgetreten sind, behoben, oder aus dem Kon-

text heraus als unkritisch betrachtet worden?

---

### A.2.3 Checkliste Veränderbarkeit

---

#### Code Qualität

- Gibt es Meldungen der Tools?
- Gibt es obsoleten Code?
- Gibt es unbenutzte Variablen?
- Wurden bereits vorhandene Funktionalitäten neu implementiert?
- Wurden Hilfsmethoden ausgelagert?
- Für das Frontend:
  - Sind alle POST Methoden richtig formatiert?
- Für das Backend:
  - Sind alle Schnittstellen wie abgesprochen implementiert?
    - Wenn ja: Sind diese begründet und dokumentiert?
  - Geben alle Views einen gültigen und passenden Status Code zurück?
  - Wurde die Datenbankenstruktur geändert?
    - Wenn ja: Sind diese begründet und dokumentiert?

Dies umfasst:

- Beschreibung der Funktion und Verwendung
- Autor
- Sind alle Methoden kommentiert?

Dies umfasst:

  - Beschreibung der Funktion und Verwendung
  - Autor
  - Eingabeparameter
  - Rückgabewert
- Gibt es weitere, schwer verständliche Stellen, die kommentiert werden sollten? (Anmerkungen der anderen Entwickler beachten)
- Sind die Kommentare verständlich für alle Entwickler?
- Ist die Dokumentation im Wiki vollständig?

#### Kommentare

- Sind alle Klassen kommentiert?

#### Tests

- Sind Tests aller Methoden vorhanden?
- Wie hoch ist die Statement Coverage?

---

## A.3 Informelle Beschreibung der Nutzungsrollen

---

Die konkreten Rechte jeder Nutzungsrolle ergeben sich aus den Beschreibungen der User Stories. Rollen sind hierarchisch strukturiert. übergeordnete Rollen erhalten auch immer alle Rechte der untergeordneten Rollen. Ein\*e Administrator\*in hat zum Beispiel alle Rechte, die auch normale Nutzer\*innen und die Moderator\*innen besitzen.

---

### A.3.1 Nutzer\*in

---

Nutzer\*innen besitzen ein Account, mit welchem sie sich gegenüber der Webseite authentifizieren können. Sie können Aufgaben lösen und Feedback erhalten. Außerdem haben sie Zugriff auf ihre eigene Nutzerseite, können die eigenen Daten ändern und ihre Statistik einsehen.

---

### A.3.2 Moderator\*in

---

Moderator\*innen können neue Lerninhalte auf der Webseite hochladen. Sie sind nur dazu berechtigt, Kurse zu verändern, die sie selbst angelegt haben. Bevor ein angelegter Kurs für alle Nutzer\*innen sichtbar wird, muss er durch einen Administrator überprüft werden.

Einzelnen Moderator\*innen können das Recht zur Freigabe von Kursen erhalten. Diese können dann ihre eigenen Kurse sofort, ohne erneute Überprüfung, freischalten.

---

### A.3.3 Administrator\*in

---

Administrator\*innen haben das Recht, anderen Nutzer\*innen zusätzliche Rechte (Moderation, Administration) zu geben und zu entziehen.

---

## A.4 Rahmen der Wikidokumentation

---

Für jede Funktion der Schnittstelle muss das Wiki folgende Informationen enthalten:

- |                                       |  |
|---------------------------------------|--|
| • Name der Schnittstelle              | • gültige Formatierung des JSON-Objekts (bei POST-Methoden)  |
| • Beschreibung der Funktionsweise     | • gültige Formatierung des JSON-Objekts in der Antwort       |
| • erlaubte Methoden (POST/GET/DELETE) | • Übersicht aller möglichen Antwortcodes und ihrer Bedeutung |
| • erwartete Header-Felder             |  |
| • erwartete Parameter aus der URL     |  |

---

## **A.5 1. Nutzerstudie**

---

Bislang wurde nur die erste Nutzerstudie ausgearbeitet. Weitere folgen bei der endgültigen Abgabe des Anhangs.

Die Aufgaben, welche die Proband\*innen erfüllen sollten, waren:

1. Registrieren Sie sich auf der Webseite mit einem neuen Account.
2. Loggen Sie sich ein.
3. Schließen Sie einen Kurs erfolgreich ab.
4. Erstellen Sie einen neuen Kurs.

## Clonecademy Nutzerstudie

Feedback

**\*Required**

**1. Wie fanden sie die Aufgaben? \***

*Mark only one oval.*

	1	2	3	4	5	
gut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schlecht

**2. Konnten sie sich auf der Seite zurecht finden? \***

*Mark only one oval.*

	1	2	3	4	5	
gut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schlecht

**3. Wie fanden sie die Menüführung? \***

*Mark only one oval.*

	1	2	3	4	5	
gut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schlecht

**4. Wie fanden sie die Anordnung der Elemente? \***

*Mark only one oval.*

	1	2	3	4	5	
gut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	schlecht

**5. Gab es etwas störendes auf der Seite?**

---

---

---

---

---



**6. Was hat ihnen gut gefallen?**

---

---

---

---

---

**7. Weitere Kritik**

---

---

---

---

---

---

Powered by  
 Google Forms

---

## B Einleitung - Bericht und Anmerkung zum QS Prozess

---

### B.1 Änderungen des Prozesses während des Semesters

---

Während des Semesters ist dem Team aufgefallen, dass der vorgestellte QS-Prozess nicht optimal ist. Die wichtigste Änderung ist, dass die automatisierte Code-Überprüfung im Backend durch das Tool *pylint* mit dem *pylint\_django* Plugin durchgeführt wurde. Dieses vereinigt die Funktionen von *pep8* und *django-lint*, ist einfacher konfigurierbar und zu verwenden. Das Team hat deshalb beschlossen, dieses Tool zu verwenden, obwohl im QS-Dokument ein anderes vorgeschlagen wurde.

---

## C Nachweis über den QS Prozess pro Iteration

---

Um dieses Dokument nicht unübersichtlich werden zu lassen, wurden nicht alle Outputs der automatisierten Tests angehängt. Wir haben stattdessen die Meldungen zum Stand der Abgabe gespeichert und sporadisch einzelne Reports angehängt, um den Verlauf der Softwareentwicklung deutlich zu machen.

Bei den Berichten über die Testabdeckung haben wir zu jeder Iteration die Übersichtsseite der Reports angehängt, und einen gesamten Report für jeden Release.

Um die Dokumentation der Webseite in der Wiki anschaulich zu dokumentieren, haben wir einen Auszug der Dokumentation für jeden Release angehängt und eine vollständige Version der Seite für die endgültige Übergabe an die Auftraggeber\*innen.

---

### Erster vollständiger QS Prozess: Iteration 6

---

---

**Iteration 7**

---

---

**Iteration 8**

---

---

**Iteration 9**

---

---

**Iteration 10**

---

---

**Iteration 11**

---

---

**Iteration 12**

---

---

**Iteration 13**

---

---

**Iteration 14**

---

---

**Iteration 15**

---

---

**Iteration 16**

---

---

---

**Iteration 17**

---

---

**Iteration 18**

---

---

**Iteration 19**

---

---

**Endgültiger Stand des Projektes und Übergabe**

---

---

## D Userstories

---

---

### D.1 Iteration bis

---

In dieser Iteration wurden grundlegende Dinge besprochen und die benötigten Frameworks aufgesetzt.

Es konnten noch keine Userstories umgesetzt werden.

---

### D.2 Iteration bis

---

ID	2
Benutzerrolle	Nutzer
Name	Authentifizierung
Beschreibung	Als Nutzer möchte ich mich auf der Seite als Nutzer einloggen können.
Akzeptanzkriterium	Das Authentifizierungsfenster wird korrekt angezeigt. Bei gültiger Eingabe wird der Nutzer auf die Kursübersicht weitergeleitet und ein Authentifizierungstoken wird generiert, der zur Authentifizierung weitergereicht wird. Bei ungültiger Eingabe wird der Nutzer darauf hingewiesen und nicht eingeloggt.
Story Points	2
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	1
Tatsächlicher Aufwand	3.5 h
Velocity	1.75 h/SP
Bemerkung	

---

ID	3
Benutzerrolle	Moderator
Name	Neuen Kurs anlegen
Beschreibung	Als Moderator möchte ich einen neuen Kurs anlegen können. Dazu gehe ich auf die Kursübersicht und wähle "Neuen Kurs anlegen" aus. Ein Editor erlaubt mir, Name und Kategorie des Kurses auszuwählen. Danach kann ich den Kurs speichern und er taucht auf der Übersichtsseite auf. Der Kurs ist erst einmal nur für Moderatoren sichtbar.
Akzeptanzkriterium	Der neu angelegte Kurs mit Name und Kategorie wird persistent gespeichert und ist für Moderatoren einsehbar.
Story Points	3
Entwickler	Claas Völcker & Leonhard Wiedmann
Umgesetzt in Iteration	1
Tatsächlicher Aufwand	3 h
Velocity	1.5 h/SP
Bemerkung	

### D.3 Iteration bis

ID	5
Benutzerrolle	Nutzer
Name	Kurs auswählen
Beschreibung	Als Nutzer möchte ich aus einer Kursübersicht einen Kurs auswählen und im Folgenden bearbeiten, also die beinhaltenden Fragen beantworten können. Nachdem ich den Kurs ausgewählt habe, soll er in meiner persönlichen Kursübersicht samt Fortschrittsangabe angezeigt werden.
Akzeptanzkriterium	Der ausgewählte Kurs kann daraufhin vollständig bearbeitet und zugleich jederzeit abgebrochen werden, wobei der Fortschritt gespeichert und in einer persönlichen Statistik angezeigt wird. Die Bearbeitung kann jederzeit fortgesetzt werden.
Story Points	7
Entwickler	Ilhan Simsiki
Umgesetzt in Iteration	3
Tatsächlicher Aufwand	11 h
Velocity	1.57 h/SP
Bemerkung	

ID	1
Benutzerrolle	Nutzer
Name	Nutzer erstellen
Beschreibung	Als Nutzer möchte ich auf der Seite einen eigenen Account anlegen können. Mit Ausfüllen des Anmeldeformulars mit den Pflichtfeldern: E-Mail-Adresse, Benutzername, Passwort. Und den Optionalen Feldern: Vorname, Nachname, Alter
Akzeptanzkriterium	Das Anmeldeformular wird korrekt dargestellt. Mit dem Klicken des "Registrieren" Buttons und Ausfüllen der Pflichtfelder wird der Account persistent in der Datenbank gespeichert.
Story Points	8
Entwickler	Tobias Huber
Umgesetzt in Iteration	5
Tatsächlicher Aufwand	20
Velocity	0.4 SP/h
Bemerkung	

ID	8
Benutzerrolle	Nutzer
Name	Frage beantworten
Beschreibung	Ein Benutzer möchte eine Frage beantworten. Dabei soll jede Art Fragen nur mit falsch oder richtig ausgewertet werden und dem Nutzer sein Ergebnis anzeigen. Falls der Benutzer Probleme bei einer Frage hat, gibt es ein Hinweisfeld "Hinweis erhalten".
Akzeptanzkriterium	Die Frage kann beantwortet werden und das Ergebnis wird in der Datenbank unter Statistik gespeichert.
Story Points	7 Story Points
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	4
Tatsächlicher Aufwand	14 h
Velocity	2 h/SP
Bemerkung	Es werden neue Daten in die Datenbank geschrieben, jedoch keine existente verändert oder gelöscht.

## D.4 Iteration bis

ID	6
Benutzerrolle	Nutzer
Name	Statistik einsehen
Beschreibung	Als Nutzer möchte ich einen Button "Meine Statistik" anklicken können. Nach Anklicken erscheint eine Seite, auf der ich meine bisherigen beantworteten Fragen sehen kann: Wie viele Fragen habe ich richtig beantwortet, wie viele sind falsch.
Akzeptanzkriterium	Die Statistik des Nutzers wird vollständig angezeigt.
Story Points	8 Story Points
Entwickler	Claas Völcker
Umgesetzt in Iteration	4
Tatsächlicher Aufwand	8 h
Velocity	1 h/SP
Bemerkung	Es werden keine Daten in der Datenbank geändert.

ID	7
Benutzerrolle	Moderator
Name	Multiple Choice Fragen einpflegen
Beschreibung	Als Moderator kann ich einem Kurs neue Fragen hinzufügen. Dazu wähle ich auf der Kursübersicht die Schaltfläche neue Frage anlegen aus. Ich kann daraufhin in einem Interface den Name der Frage, den Text der Frage und mögliche Antwortmöglichkeiten eingeben. In einem weiteren Fenster kann ich einen Text eingeben, der den Nutzenden bei erfolgreicher Bearbeitung angezeigt wird und einen Feedbacktext bei falscher Beantwortung.
Akzeptanzkriterium	Eine neu angelegte Frage wird persistent unter dem Kurs gespeichert und ist nach Freigabe für Nutzende bearbeitbar.
Story Points	8 Story Points
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	4
Tatsächlicher Aufwand	15 h
Velocity	1.875 h/SP
Bemerkung	Moderatoren können nur Fragen in ihren eigenen Kursen anlegen. Es werden neue Daten in die Datenbank geschrieben, aber keine existierenden verändert oder gelöscht.



## D.5 Iteration bis

ID	9
Benutzerrolle	Admin
Name	Nutzerliste einsehen
Beschreibung	Als Admin möchte ich mir eine Übersicht über die Vorhandenen Nutzer anzeigen lassen um damit auf die Ansicht für Details zu gelangen. Die hierfür benötigte Schlatfläche findet sich nur für Admins einsehbar auf dem Dashboard.
Akzeptanzkriterium	Eine Liste aller angemeldeten Nutzer*innen wird anch Klicken auf die entsprechende Schaltfläche angezeigt. Dies ist nur als Administrator*in möglich.
Story Points	3 Story Points
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	4
Tatsächlicher Aufwand	1:30 h
Velocity	0.5 h/SP
Bemerkung	Es werden keine Daten verändert, gelöscht oder erweitert. Eine Einsicht aller Nutzer*innen ist nur Administrator*innen möglich.

ID	10
Benutzerrolle	Admin
Name	Profil einsehen
Beschreibung	Als Admin der Seite möchte ich auf die Profile der Nutzer Einsicht haben. Ich möchte aus einer Liste von Nutzern einen Nutzer auswählen, anklicken und somit sein Profil sehen können.
Akzeptanzkriterium	Der Administrator kann jedes Profil von Nutzern einsehen, das Profil der Nutzer wird korrekt dem Admin angezeigt.
Story Points	4 Story Points
Entwickler	Ilhan Simsiki
Umgesetzt in Iteration	4
Tatsächlicher Aufwand	10 h
Velocity	2.5 h/SP
Bemerkung	Einsicht auf die Profile fremder Nutzer hat nur ein Admin. Es werden keine Daten geändert oder eingefügt.

ID	11
Benutzerrolle	Nutzer
Name	Moderationsrechte beantragen
Beschreibung	Als Nutzer möchte ich auf meiner Profilseite die Möglichkeit haben, Moderationsrechte zu beantragen. Ich klicke dazu auf dem Dashboard auf die Schaltfläche "Moderationsrechte beantragen", woraufhin mich die Webseite auffordert, eine Begründung einzugeben. Diese wird zusammen mit einem Link auf mein Profil an die Administartor*innen geschickt. Die Schaltfläche ist so lange nicht mehr auswählbar, bis ein*e Administrator*in mein Profil besucht und den Antrag angenommen oder abgelehnt hat.
Akzeptanzkriterium	Die Schaltfläche ist auswählbar, solange kein Antrag vorliegt. Die E-Mail wird korrekt mit allen oben genannten Informationen verschickt.
Story Points	8 Story Points
Entwickler	Claas Völcker
Umgesetzt in Iteration	8
Tatsächlicher Aufwand	5:30 h
Velocity	0.68 h/SP
Bemerkung	Es werden keine Daten geändert oder eingefügt.

## D.6 Iteration bis

Diese Iteration wird dafür reserviert, den aktuellen Status der Webseite soweit zu bekommen, dass ein Alpha-Release auf dem Server der Auftraggeber\*innen möglich ist. Dazu wird der Code von Back- und Frontend überarbeitet, Installskripte geschrieben und Integrationstests durchgeführt.

## D.7 Iteration bis

ID	15
Benutzerrolle	Nutzer (bei Registrierung)
Name	Sprachauswahl
Beschreibung	Bei der Registrierung wählt der Nutzer auch eine Sprache aus, in der alle Inhalte der Webseite angezeigt werden sollen. Danach bekommt er nur Kurse seiner gewählten Sprache angezeigt.
Akzeptanzkriterium	Die Sprachauswahl wird mit dem Nutzer gespeichert und auf der Webseite berücksichtigt. Der Nutzer erhält als Erleichterung seiner Wahl eine Auskunft darüber, wie viele Kurse in der jeweiligen Sprache vorhanden sind.
Story Points	6
Entwickler	Ilhan Simsiki
Umgesetzt in Iteration	7
Tatsächlicher Aufwand	8
Velocity	
Bemerkung	Aktuell werden nur Deutsch und Englisch unterstützt, aber die Möglichkeit zum Anlegen weiterer Sprachen wird im Backend berücksichtigt (Stichwort: Erweiterbarkeit)

## D.8 Iteration bis

ID	14
Benutzerrolle	Nutzer
Name	Kurs fortführen
Beschreibung	Als Nutzer möchte ich einen bereits begonnenen Kurs fortführen können. Dazu gehe ich im Menü der Kursübersicht auf die Schaltfläche "Kurs fortführen", um bei der ersten nicht beantworteten Frage fort zu fahren, oder wähle eine bereits beantwortete Frage aus, um an diesem Punkt mit der Kursbearbeitung fortzufahren.
Akzeptanzkriterium	Die Schaltfläche "Kurs fortführen" wird angezeigt, sobald eine Frage in einem Kurs beantwortet wurde. Es ist nicht möglich Fragen in einem Kurs zu überspringen.
Story Points	8
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	7
Tatsächlicher Aufwand	7h
Velocity	0,88 h/SP
Bemerkung	

ID	12
Benutzerrolle	Admin
Name	Moderationsrechte freischalten
Beschreibung	Als Admin möchte ich einer*inem beliebigen Nutzer*in meiner Wahl Moderatorenrechte gewähren, indem ich auf dem zugehörigen Profil die entsprechende Schaltfläche "Moderationsrechte freigeben" auswähle.
Akzeptanzkriterium	Dem*der entsprechenden Nutzer*in werden Moderationsrechte beim Klicken auf die Schaltfläche gewährt. Wenn ein Moderationsantrag vorliegt und abgelehnt wird, kann der*die entsprechende Nutzer*in erneut Rechte beantragen.
Story Points	7 Story Points
Entwickler	Tobias Huber
Umgesetzt in Iteration	8
Tatsächlicher Aufwand	7,5 h
Velocity	
Bemerkung	Eventuelle Sicherheitsfeatures, wie bestätigung durch erneute Passworteingabe, werden in eine andere US ausgelagert. In der Datenbank werden Moderationsrechte gespeichert, aber keine anderen Daten verändert, erweitert oder gelöscht.

## D.9 Iteration bis

ID	19
Benutzerrolle	Nutzer*in
Name	Nutzerdetail ändern: E-Mail
Beschreibung	Als Nutzer*in möchte ich auf meinem Profil einstellen können, welche E-Mailadresse mit meinem Account verknüpft ist und dementsprechend zur Kommunikation verwendet wird, indem ich im Abschnitt E-Mail das Textfeld neu ausfülle und auf "speichern" drücke.
Akzeptanzkriterium	Die Userstory wird akzeptiert, wenn der*die Nutzer*in seine E-Mailadresse ändern kann.
Story Points	3
Entwickler	Ilhan Siminski
Umgesetzt in Iteration	12
Tatsächlicher Aufwand	6,5 h
Velocity	
Bemerkung	

ID	20
Benutzerrolle	Nutzer*in
Name	Nutzerdetail ändern: Passwort
Beschreibung	Als Nutzer*in möchte ich auf meinem Profil unter dem Abschnitt "Passwort" mein eigenes Passwort neu festlegen können, indem ich das Textfeld neu ausfülle, mein neues Passwort noch einmal bestätige und auf "speichern" drücke.
Akzeptanzkriterium	Die Userstory wird akzeptiert, wenn der*die Nutzer*in sein*ihr Passwort ändern und sich mit diesem weiterhin auf der Website einloggen kann.
Story Points	4
Entwickler	Tobias Huber
Umgesetzt in Iteration	12
Tatsächlicher Aufwand	6 h
Velocity	
Bemerkung	Zwei Faktor Authentifizierung wird in einer anderen Userstory behandelt.

## D.10 Iteration bis

ID	4
Benutzerrolle	Moderator
Name	Kurs bearbeiten
Beschreibung	Als Moderator möchte ich meine eigenen Kurse bearbeiten können. Dafür soll in der Kursübersicht bei meinen Kursen ein Feld "Kurs bearbeiten" zum Anklicken sein. Nach Anklicken erscheint ein Bearbeitungsfenster, auf dem ich Titel, Zuordnung des Kurses und die einzelnen Module des Kurses bearbeiten kann. Wenn ich die Bearbeitung ohne zu Speichern abbreche, wird der Kurs wieder auf seinem ursprünglichen Zustand gesetzt.
Akzeptanzkriterium	Der bearbeitete Kurs wird korrekt in der Datenbank abgespeichert und auf der Seite aktualisiert. Moderatoren können nur ihre eigenen Kurs bearbeiten. Wird der Änderungsprozess abgebrochen ändern sich die Daten in der Datenbank nicht.
Story Points	8
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	12
Tatsächlicher Aufwand	8,25 h
Velocity	
Bemerkung	Verschoben in spätere Iteration

ID	17
Benutzerrolle	Nutzer*in
Name	Nutzerdetail ändern: Sprache
Beschreibung	Als Nutzer*in möchte ich auf meinem Profil einstellen können, ob ich Englisch oder Deutsch als allgemeine Sprache der Website bevorzuge, indem ich vorher einen Reiter "Einstellungen" öffne und dort im Abschnitt "Sprache" den Knopf mit der entsprechenden Sprache auswähle.
Akzeptanzkriterium	Die Userstory wird akzeptiert, wenn der*die Nutzer*in nach der Auswahl die Website auf der korrekten Sprache angezeigt bekommt, sofern Übersetzungen vorliegen.
Story Points	3
Entwickler	Ilhan Siminski
Umgesetzt in Iteration	10
Tatsächlicher Aufwand	5 h
Velocity	
Bemerkung	

ID	18
Benutzerrolle	Nutzer*in
Name	Nutzerdetail ändern: Name
Beschreibung	Als Nutzer*in möchte ich auf meinem Profil sowohl meinen eigenen (Vor- und Nach-) Namen, als auch meinen Username ändern können, indem ich vorher einen Reiter "Einstellungen" öffne und dort im Abschnitt "Namen" die Textfelder neu ausfülle und auf Speichern" drücke.
Akzeptanzkriterium	Die Userstory wird akzeptiert, wenn der Name in der Datenbank nach Eingeben des korrekten Passworts persistent gespeichert wird.
Story Points	3
Entwickler	Tobias Huber
Umgesetzt in Iteration	10
Tatsächlicher Aufwand	4
Velocity	
Bemerkung	Der Username kann nicht geändert werden, da er als eindeutige Authentifikation gegenüber anderen Nutzer*innen der Plattform dient.

## D.11 Iteration bis

ID	22
Benutzerrolle	Moderator
Name	Multiple Choice Bilder einpflegen
Beschreibung	Als Moderator möchte ich bei einer Multiple Choice Frage zur Frage und zum Feedback ein Bild anzeigen lassen. Dazu möchte ich als Moderator beim Fragen erstellen die Möglichkeit haben ein Bild für die Frage und eines für das Feedback hochzuladen.
Akzeptanzkriterium	Die Bilder werden hochgeladen und im hinteren Drittel des Fensters angezeigt. Das Feedbackbild wird erst angezeigt wenn die Frage richtig beantwortet ist.
Story Points	5
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	11
Tatsächlicher Aufwand	4,5
Velocity	1,1
Bemerkung	



ID	16
Benutzerrolle	Nutzer
Name	Kursfortschritt anzeigen (Übersicht)
Beschreibung	Als Nutzer möchte ich meinen Fortschritt in einem Kurs in der Kursübersicht einsehen können.
Akzeptanzkriterium	Die Schaltfläche zur Kurswahl wird prozentual zum Fortschritt eines Nutzers im jeweiligen Kurs eingefärbt. In der Kursübersicht wird die Anzahl der beantworteten Fragen und die Gesmatzahl an Fragen im Kurs angezeigt.
Story Points	5
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	7
Tatsächlicher Aufwand	5 h
Velocity	
Bemerkung	Ist noch nicht begonnen, nur formuliert.

#### D.12 Iteration bis

#### D.13 Iteration bis

#### D.14 Iteration bis

ID	23
Benutzerrolle	Moderator
Name	Einklappen von Komponenten ermöglichen
Beschreibung	In der Kursbearbeiten-Übersicht soll jede Komponente (Module und Fragen) ein und ausklappbar sein. Dazu soll neben der Überschrift der jeweiligen Komponente eine Schaltfläche vorhanden sein. Ein klick auf diese ermöglicht, abhängig vom aktuellen Zustand, das ein und ausklappen.
Akzeptanzkriterium	Die Schaltfläche ist für jede Komponente vorhanden und durch das klicken wird nur die betreffende Komponente animiert.
Story Points	8
Entwickler	Ilhan Siminski
Umgesetzt in Iteration	
Tatsächlicher Aufwand	
Velocity	
Bemerkung	

ID	24
Benutzerrolle	Moderator
Name	Info Fragetyp einpflegen
Beschreibung	Als Moderator möchte ich einen Informationstext anstelle einer Frage einfügen können. Dieser soll im Editor wie eine normale Frage anlegbar und bearbeitbar sein. Anstelle von möglichen Antworten kann ein längerer Text eingefügt werden.
Akzeptanzkriterium	Der Info-typ kann im Editor wie eine Frage hinzugefügt und bearbeitet werden und wird nach speichern persistent in der Datenbank angelegt.
Story Points	3
Entwickler	Claas Völcker
Umgesetzt in Iteration	13
Tatsächlicher Aufwand	2,5 h
Velocity	
Bemerkung	Nach Speichern des Kurses wird der Informationstext persistent in der Datenbank gespeichert.

ID	25
Benutzerrolle	Nutzer*in
Name	Info Fragetyp anzeigen
Beschreibung	Als Nutzer*in möchte ich beim Bearbeiten des Kurses Fragen vom Typ Information angezeigt bekommen.
Akzeptanzkriterium	Dem*der Nutzer*in werden Fragen vom Typ Information angezeigt und er kann sie bearbeiten, indem er auf die Schaltfläche "Fortfahren" klickt.
Story Points	5
Entwickler	Claas Völcker
Umgesetzt in Iteration	14
Tatsächlicher Aufwand	5,5 h
Velocity	
Bemerkung	

ID	26
Benutzerrolle	Nutzer*in
Name	Informationsseite
Beschreibung	Als Nutzer*in möchte ich Informationen über die Webseite und das zugrunde liegende iGem Projekt angezeigt bekommen, wenn ich im entsprechenden Reiter auf die Schaltfläche "About" klicke.
Akzeptanzkriterium	Die Informationsseite kann als HTML hinterlegt werden und wird korrekt dargestellt.
Story Points	2
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	13
Tatsächlicher Aufwand	2 h
Velocity	1 h / SP
Bemerkung	
ID	28
Benutzerrolle	Admin
Name	Statistik exportieren
Beschreibung	Als Admin möchte ich die Möglichkeit haben, die gesamte Nutzungsstatistik der Seite zu exportieren.
Akzeptanzkriterium	Es gibt eine Schaltfläche in der Statistik übersicht, über welche ich mir alle geloggten Einträge der Statistik als *.csv exportieren kann.
Story Points	4
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	13
Tatsächlicher Aufwand	3,75h
Velocity	0,94 h/SP
Bemerkung	

## D.15 Iteration bis

ID	27
Benutzerrolle	Nutzer
Name	Kurs Dashboard aufrufen
Beschreibung	In einer Frage möchte ich als Nutzer die Möglichkeit haben mir das eine Kursübersicht anzeigen zu lassen. Dieser enthält alle Fragen des Kurses, den aktuellen Fortschritt und Schaltflächen um richtig bearbeitete Fragen zu wiederholen.
Akzeptanzkriterium	Es gibt eine Schaltfläche in jeder Frage um das Dashboard zu öffnen. Das Dashboard enthält alle Fragen und den aktuellen Fortschritt.
Story Points	6
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	11
Tatsächlicher Aufwand	6,5h
Velocity	1,08 h/SP
Bemerkung	

ID	38
Benutzerrolle	Nutzer
Name	Dashboard ansicht strukturieren
Beschreibung	Wenn ein Nutzer auf die Seite /courses/ kommt, soll für den Nutzer ersichtlich sein, welche Kurse er zuletzt bearbeitet und noch nicht abgeschlossen hat oder falls er noch keine Kurse angefangen hat, wie er neue Kurse findet.
Akzeptanzkriterium	Sobald ein Nutzer die Seite /courses/ aufruft, werden ihm alle von ihm angefangenen Kurse angezeigt. Die Übersicht enthält für jeden Kurs die folgenden Informationen: Kurstitel, aktuelles Modul, Prozentsatz des Kurses, welcher bereits abgeschlossen wurde. Durch Auswählen einer Schaltfläche kann der Nutzer sofort mit der Bearbeitung des Kurses fortfahren.
Story Points	8
Entwickler	Claas Völcker
Umgesetzt in Iteration	15
Tatsächlicher Aufwand	9 h
Velocity	
Bemerkung	

---

## D.16 Iteration bis

---

**Noch geplant, nicht begonnen**

---

ID	16
Benutzerrolle	Nutzer
Name	Kursfortschritt anzeigen (Übersicht)
Beschreibung	Als Nutzer möchte ich meinen Fortschritt in einem Kurs in der Kursübersicht einsehen können.
Akzeptanzkriterium	Die Schaltfläche zur Kurswahl wird prozentual zum Fortschritt eines Nutzers im jeweiligen Kurs eingefärbt. In der Kursübersicht wird die Anzahl der beantworteten Fragen und die Gesamtzahl an Fragen im Kurs angezeigt.
Story Points	5
Entwickler	Leonhard Wiedmann
Umgesetzt in Iteration	7
Tatsächlicher Aufwand	5 h
Velocity	
Bemerkung	Ist noch nicht begonnen, nur formuliert.

---

## E Protokolle

---

# Protokol Auftraggebertreffen

## 12.05.2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas A. Völcker  
BP Cloncademy

### Technische Fragen

- Multi device Nutzung
  - nur Desktop Nutzung
- Multilingualität
  - Fragen werden Kursen in einer Sprache zugeordnet, um mehrere Sprachen zu ermöglichen, muss der Kurs zweimal angelegt werden.
  - Das Interface wird mit Option auf Mehrsprachigkeit (Deutsch/Englisch) angelegt
- Die Webseite wird mit Bootstrap erstellt
- Die BP-Gruppe erhält keinen Admin Zugang zum Server des iGEM Teams

### Inhaltliche Fragen

- Welche Kategorisierungen von Fragen gibt es?
  - Es gibt Kurse, die aus einer unterschiedlichen Menge an Fragen bestehen.
  - Die Kurse sind nach Schwierigkeit kategorisiert: Anfänger\*innen Schule/Studium - Expert\*innen
  - Jeder Kurs erhält optional Tags mit den Themen des Kurses (z.B.: Klonierungsverfahren)
- Kurse sind nach dem Anlegen nicht sofort sichtbar, sondern müssen erst freigeschaltet werden. Das erlaubt es, den Kurs weiter zu anzupassen, bevor Lernenden diesen bearbeiten.
- Alle Fragen bestehen aus mindestens einem Namen und gehören zu einem Kurs.
- Persistent gespeichert werden soll

- Der Fortschritt eines\*r Nutzer\*in im Kurs
- Die vollständig bearbeiteten Kurse
- Wie soll die Fragenseite aufgebaut sein?
  - Analog zu Codecademy: Links die Aufgabenstellung & weitere Informationen, Rechts das Bearbeitungsfenster
  - Beim Feedbackfenster wird Fragestellung & Antwortmöglichkeiten mit angezeigt
  - Ein Knopf zum Anzeigen eines Hinweises sollte eingebaut werden.
- Aufbau der Landing Page / des Dashboards
  - Wie bei Codecademy mit aktiven Kursen, Navigationsschaltflächen und weiteren Kursen
- Welche Nutzerrollen gibt es?
  - Identifiziert sind Admin (Zugriff auf alles), Moderator (stellt Kurse ein und gibt diese frei)
- Kein dynamisches Feedback, nur statische: Es war falsch, so ist der richtige Ansatz
- Kurse sollen nach Schwierigkeit und Kategorien filterbar sein
- Ob Nutzer\*innen am Ende eines Kurses eine Möglichkeit zum Feedback gegeben wird, wird noch diskutiert.

### Weiteres Vorgehen

- Ein PDF mit allen (vollständigen) Userstories (ToDo, Diese Iteration) immer bis Montag im Seafile hochladen

# Protokoll 19.05.2017

CloneCadamy



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Struktur der Kurse Beispiel:

- Grundlagen (Organisationsgruppe)
  - Einstieg I (? Frage)
  - Einstieg II (?)
  - PBS (Modul)
    - Translation (?)
- „Kurse“ sind öffentlich oder privat (private nicht einsehbar)
- Falls die Kurse privat sind, nur Einladung von Moderator
- Als Gruppe zusammen einen Kurs erstellen
  - mehrere Moderatoren können den Kurs bearbeiten
- Jede Organisation hat Moderator(en)
- Module mit Tags (variabel aber mit Vorgaben zur Auswahl)
  - Suchfunktion
- Module werden von Anfang bis zum Ende immer schwerer
- Fragen die für einen Kurs erstellt wurden, gehören dann auch nur zu diesem Kurs
  - Module die einem Kurs angehören können auch nur diesem Kurs angehören (keine Duplikate)
  - öffentliche Fragen sind immer öffentlich
- Nach jeder bearbeiteten Frage wird das Ergebnis in die Statistik eingetragen.
- Was haben Fragen gemeinsam?
  - Name, Inhalt, richtige Antwort, Feedback
  - Fragen Inhalt: Text mit oder ohne Bilder
- Feedback bekommt man am Ende vom Modul
- Man kann auch weiter wenn man eine Frage falsch beantwortet hat. Kann auch auf die falsch beantwortete Frage wieder zurück
- Bei falschen Antworten keine Hinweise und auch keine Lösung
- Dynamisches Feedback bei Fragen die sehr oft Falsch beantwortet werden(mit Default z.B. „FALSCH“ und Möglichkeit selber ein Feedback zu schreiben)



- 
- Statistik für die Fragen
    - wie oft wurde die Frage Richtig oder Falsch beantwortet? Bsp:
      - Frage XY - XX (zweimal Falsch)
      - Frage YZ - XXXO (dreimal Falsch einmal richtig)
        - vom Lehrer einsehbar

#### Qualitätsmerkmale

- Sicherheit (Security)
  - Die Daten von Angreifer schützen
  - Nutzer Daten von Angreifer schützen
  - kein unbefugter Zugang in Admin/Mod Bereichen
- Veränderbarkeit (Update = einpflegen der Inhalte)
- Bedienbarkeit
  - MUSS einfach sein
- (Verbrauchsverhalten)

# Protokoll Auftraggebertreffen

## 2. Juni 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas A. Völcker  
BP Clonecademy

### Technische Anmerkungen

- Server muss HTTPS sprechen
- Dockernutzer einrichten

### Inhaltliche Anmerkungen

- Es geht weiter, sobald ein Nutzer die Frage richtig beantwortet hat.
  - Fehlerhighlighting an Fragentypus anpassen
- Lückentext können wir als kommende Frage aufnehmen
  - Fragen sind *immer* in Modulen
- Modulstruktur der Fragen
  - Lerngruppen fliegen raus
  - Nur zwei große Übergruppen: "iGEM", "allg. Biotech"
  - privat vs. öffentlich fliegt raus, nur sichtbar und nicht sichtbar bleibt (Entwurf und Öffentlich)
  - Datenbankverknüpfungen per ForeignKey
  - Fragen sind *immer* in Modulen
- Es wird eine Schaltfläche gewünscht: Moderator werden
  - Schaltfläche klicken, Mail an Admins, Freischalten oder nicht
- iGEM oder Biotech und Sprache wählbar machen (keine einmalige Wahl)
  - Sprache ist nur am Anfang auswählbar, damit man nicht mit den angebotenen Kursen durcheinander kommt
- Moderator kann einen Kurs nicht selbst freischalten, sondern muss die Freischaltung beim Admin beantragen
  - Wenn Kurse einmal freigeschaltet werden, können sie geändert werden

- Wer kann Kurse später bearbeiten?
- Kurse brauchen ein Feld: Wer hat angelegt
- iGEM-Team berät sich zum Thema Moderations-Whitelist;

- Rechteentzug implementieren
- Lerntext
  - Infotext kommt wo, wird wie eingepflegt?
    - \* Wird im iGEM-Team diskutiert?

### Qualitätssicherung

- Wie ist Änderbarkeit genau definiert?
  - Codemodifizierung einfach, Erweiterungen coden einfach
  - Wird auf Codeebene gewährleistet
    - \* Kommentare sind wichtig
    - \* Übersichtlicher, wartbarer Code
    - \* Hooks für Erweiterungen: gutes Vererbungsmodell

### Weiteres Vorgehen

- Wir müssen genauer auf die konkrete Beschreibung in den User Stories achten
- Wir haben ein, zwei Kleinigkeiten vergessen
- Gestrichene Features verbleiben erst einmal im Code, damit sie später reingenommen werden können
- Das Wiki muss befüllt werden
  - Nutzerrechte werden im Development Wiki festgehalten
  - Dokumentation der API ist wichtig! Was wird übergeben auf welchen API Schnittstellen
- Code Refactoring vor allem im Backend

# Protokoll Auftraggebertreffen

## 9. Juni 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas A. Völcker  
BP Clonecademy

### Anmerkung

- Das Protokoll ist ein Gedächtnisprotokoll, da durch einen Festplattenabsturz das Original verloren gegangen ist.
- Eventuelle Unstimmigkeiten bitte so schnell wie möglich melden.

### User Stories

- Die präsentierten Userstories wurden abgenommen (siehe US PDF)
- "Adminrechte freischalten" (ID 12) wurde in die nächste Iteration verlängert

### Funktionalität

- Man kann Fragen überspringen und später beantworten
- Die Texte ändern sich nach Kurs und Modul, müssen also in beiden gespeichert werden

### Aufbau der ersten Übersichtsseite

- Auswahl der Sprache (Englisch/Deutsch), später nicht mehr änderbar
- Auswahl der Kursgruppe: iGEM, Allgemein

### Design

- Das grundlegende Design, dass von der BP Gruppe präsentiert wurde, ist als Basis gut
  - drei Farben, plus weiß und schwarz
  - Weiß, schwarz und grau sollten als Basisfarben verwendet werden, die anderen beiden als Akzente
  - (52/56/60), (196/6/51), (214/191/131)

### Nutzerrollen

Die Namen der Rollen sind vorläufige Arbeitsversionen

- Nutzer: Können Fragen sehen und beantworten, können ihr eignes Profil einsehen
- Moderator\*innen: Können Kurse neu anlegen und eigene Kurse ändern
- Trusted Moderator\*innen: Können Kurse anlegen, die nicht mehr explizit freigeschaltet werden müssen, können alle Kurse ändern
- Admins: Können Rechte vergeben und entziehen, können Kurse freischalten
- Server-Admins: Haben Zugriff auf den Server und das Django Backend

Die Ordnung über die Gruppen impliziert, dass Nutzer\*innen alle Rechte der kleineren Gruppe haben

- Zum Thema Accountlöschung/Blockierung tauscht sich das iGEM Team noch aus.

### Weiteres Vorgehen

- Die kommende Iteration streckt sich über zwei Wochen
- Ziel der Iteration ist es, den existierenden Code zu überarbeiten, und Integrationstests durchzuführen, damit ein erster Release auf dem Server möglich ist
- Dazu wird das Design umgesetzt und die Datenbank sauber überarbeitet
- Alle existierenden Schnittstellen und der Code sollte dokumentiert sein
- Der Code wird auf Modularität überprüft
- Am 16.06. findet ein eigenes Treffen statt, um technische Fragen zu klären
- Als Ziel wird eine Vorstellung der Webseite auf dem iGEM Treffen angestrebt

# Protokol Auftraggebertreffen

## 23. Juni 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas A. Völcker  
BP Clonecademy

### Aufgabe 1 Bericht der letzte Iteration

#### Aufgabe 1.1 Design

Wir präsentieren das neue Design  
Anmerkungen:

- Das Menü zur Nutzer Auswahl ein Suchfeld haben und kein reines Dropdown sein

#### Aufgabe 1.2 Weitere Anmerkungen

- Kurskategorien sollten bei den Kursen auswählbar sein
  - z.B. via eines Reiters über der Übersicht im Dashboard
- Bei der Registrierung sollte die Lernsprache auswählbar sein
- Der "Get hint" Knopf fehlt noch
  - Wird in einer eigenen User Story programmiert
- Der Kursfortschritt ist noch nicht ganz klar
  - Das muss auf der Kursübersicht einsehbar sein (z.B. via Fortschrittsbalken)

### Aufgabe 2 Vorschlag für neue User Stories:

- Nutzerdaten ändern
- Kurs fortführen
  - Knopf "Kurs starten" wird zu "Fortführen"
  - Man kann einen bereits begonnenen Kurs an jeder Stelle, an der man bereits war, wiederholt starten

- Moderatorenrechte vergeben
- Neue Fragen
- Sprachauswahl bei Registrierung
  - Klar machen, dass Englisch Primärsprache ist
  - Vorschlag: Anzahl der verfügbaren Kurse pro Sprache anzeigen

### Aufgabe 2.1 Ausgewählt

- Moderatorenrechte vergeben
- Kurs fortführen
- Sprachauswahl

### Aufgabe 3 Zielvereinbarung für die nächste Woche

- User Stories bearbeiten
- Refactoring erstmal abschließen
- Deployment ermöglichen

### Aufgabe 4 Weitere Planung

- Vor einem iGEM Teamtreffen können Nutzerstudien gemacht werden
  - Am Anfang eine kleine Gruppe mit viel Beobachtung
  - Auf Leos Laptop, um Screenshot zu ermöglichen

# Protokol Auftraggebertreffen

## 30. Juni 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas A. Völcker  
BP Clonecademy

### Aufgabe 1 Bericht der letzte Iteration

#### Aufgabe 1.1 User Stories

- Kursfortschritt anzeigen
  - abgenommen
- Kurs fortsetzen
  - abgenommen
  - Anmerkungen: In den Nutzerstudien wurde bemängelt, dass nicht alle Buttons klar platziert sind
  - Der start Course Button könnte die volle Kursübersicht ersetzen
- Sprache auswählen
  - abgenommen
  - die deutsche Lokalisierung fehlt noch zu Teilen

### Aufgabe 2 Design Anmerkungen

- Die Fragenbeantwortung dritteln
  - 1. Drittel: Dashboard
  - 2. Drittel: Fragefenster
  - 3. Drittel: Feedbackfenster (wenn kein Feedback vorhanden, leer)
    - \* Visuelles Feedback zur Frage (z.B. durch Symbole)
    - \* Möglichkeit, einen Feedbacktext für korrekte Antworten einpflegen
    - \* Feedbacktext sollte Markdown Support haben
- Fragentitel sind erwünscht, wenn keine eingegeben werden, sollte ein generischer aus dem Modulnamen generiert werden.

### Aufgabe 3 Vorschlag für neue User Stories

- Drag & Drop
  - Anschauliche Plasmidkarten werden ins Seafile hochgeladen
  - Drag&Drop Elemente sollten nicht zu viel Informationen über die Lösung verraten
  - Beim Klick auf ein Element sollten die Lösungsmöglichkeiten klar ersichtlich sein
  - Es könnte mehrere richtige Lösungen geben, es wird überlegt, wie das eingepflegt werden kann
  - Vorschlag zum Einpflegen:
    - \* Mögliche Antwortfelder auswählbar
    - \* Mögliche "Stückchen" auswählbar
    - \* Gültige Kombinationsmöglichkeiten angebe
- Markdown Support

### Aufgabe 4 Zielvereinbarung für die nächste Woche

- Es wird um eine realistische Einschätzung der Machbarkeit von Drag & Drop bis nächste Woche durch das BP Team gebeten
- BP Team diskutiert über die bessere Platzierung von Knöpfen
- Markdown Support wird angeschaut
- Überarbeitung des Designs der Frageseite ist erwünscht
- User Stories

### Aufgabe 5 Weitere Planung

- Stabilitätstest wären cool

# Protokol Auftraggebertreffen

## 14. Juli 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas A. Völcker  
BP Clonecademy

### Aufgabe 1 Bericht der letzte Iteration

- Leo präsentiert das Frontend
  - Multiple-Choice zu Dritteln macht wenig Sinn, zu viel Platz wird für Feedback verschwendet
  - Der genaue Aufbau der Frageseite muss mit jedem Fragetypus neu evaluiert werden
  - Die Bestandteile der Frage könnten noch deutlicher getrennt werden
  - Vorschlag: Auf der "Kurs anlegen" Seite kann man fertig bearbeitete Komponenten wieder einklappen, damit man nicht zu viel Platz verschwendet → wird eine User Story

### Aufgabe 1.1 User Stories

- "Moderationsrechte freischalten"
  - Feedback zur Seite ist noch nicht vorhanden
  - nicht abgenommen

### Aufgabe 2 Konkretisierte User Stories

- Wir beschränken uns auf Plasmidkarten
  - Es gibt eine kreisförmige Schablone, mit n Feldern
  - Diese wird vorher implementiert und dann beim Erstellen ausgewählt
- Zwischen 3 und 4 Elementen auf einen Kreis

- Welche Antwortbausteine gibt es?
  - Auch falsche, die gibt der Aufgabensteller ein
  - Es gibt einen Satz an vorgefertigten Elementen, die man beim Eintragen beschriften kann

### Aufgabe 3 Zielvereinbarung für die nächste Woche

- Das Refactoring wird beendet

### Aufgabe 3.1 Neue User Stories

- Drag And Drop (Plasmidkarten) (Moderator/Nutzer)
- Nutzerdetails ändern (Nutzer)
- Neue Kategorie hinzufügen (Administrator)
- Übersicht über das eigene Profil (Nutzer)

### Aufgabe 4 Weitere Planung

- Das iGEM Team plant einen Release Termin, der passt.
- Dafür werden die Server-Leute gebraucht.
- Es wird ein Termin um das erste August Wochenende angestrebt.

# Protokoll Auftraggebertreffen

## 21. Juli 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Leonhard Wiedmann  
BP Cloncademy

### Aufgabe 1 Bericht der letzte Iteration

- Die Userstories werden vorgezeigt
  - bei edit course sollen
  - us-15(Passwort und email ändern) wurde abgenommen

#### Aufgabe 1.1 User Stories

- "edit course"
  - bessere Fehlermeldungen wenn Felder leer
- "change password"
  - done
- "change email"
  - done

### Aufgabe 2 Konkretisierte User Stories

- Plasmidkarten wurden wieder verworfen
- Es wird zuerst die Multiple choice Frage überarbeitet

#### Aufgabe 2.1 "Multiple choice Fragen"

- Markdown prüfen ob man es erweitern kann (speziell Pfeile um Text)
- Bild zur Frage hinzufügen
- Bild zum Feedback hinzufügen
- Bild zu möglichen Antworten hinzufügen

### Aufgabe 3 Zielvereinbarung für die nächste Woche

- Das Design wird überarbeitet
- Die Multiple Choice Frage wird überarbeitet

#### Aufgabe 3.1 Design

- der ganze Bildschirm soll ausgefüllt sein
- den grauen Hintergrund entfernen
- hover Effekt der Buttons überarbeiten
- header kleiner machen
- Schattierungen in der Hauptbox entfernen

### Aufgabe 4 Weitere Planung

#### Aufgabe 4.1 "visibility"

- Ein Moderator kann alle invisible Kurse sehen
- ein Moderator kann auch Kurse invisible schalten
- der Admin kann auch Kurse löschen
- der Trusted Moderator hat die gleichen rechte wie ein Admin im Kurs Bereich

# Protokoll Auftraggebertreffen

## 28. Juli 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas Völcker  
BP Clonecademy

### Bericht der letzte Iteration

#### User Stories

- bei den Nutzerprofil Änderungen wird die Passwortbestätigung nicht richtig überprüft
- Kurs bearbeiten: die Fehlermeldung wird nicht korrekt angezeigt
- Markdown: angerissen, nicht fertig
- Bilder hochladen: abgenommen
  - kleine Anmerkung: die Zwischenüberschriften "Image for the question" und "Image for the result" ist redundant

#### Design

- gewählte Schriftart: Railway (von Google)
- finaler Farbcode aus Sample im Seafile
  - Achtung: weiß ist eigentlich ein Grauton, bitte beachten
- Logo im Seafile: full\_red\_blue\_Main

- die Akzentfarbe Blau wird den bisherigen Sandton ersetzt, die Buttons werden Grau, das BP Team setzt ein paar Kombinationsmöglichkeiten um

- in einem eigenen Treffen kann das Design dann weiter verfeinert werden

### Kommende Iteration

- kommende Woche soll das Deployment stattfinden
  - 7. August, 14:00 Uhr, im Schreibraum des iGEM Team, Claas schließt sich mit Markus kurz
- Design-Treffen: Dienstag, 14:00 Uhr, an der Biologie Seminarraum Botanik, bis dahin werden die Farben eingearbeitet
- Skalierung der hochgeladenen Bilder muss bedacht werden

### Weitere Planung

- YouTube Embedding wurde angesprochen
  - wurde vom BP Team als machbar eingestuft



# Protokoll Auftraggebertreffen

## 21. August 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas Völcker  
BP Clonecademy

### Bericht der letzte Iteration

#### User Stories

- bei den Nutzerprofil Änderungen wird die Passwortbestätigung nun korrekt überprüft
- Kurs bearbeiten: Bild hochladen funktioniert zur Zeit nur als .jpg außerdem wird das Bild nicht angezeigt

#### Deployment

- Das beste wäre, wenn die WebSite am Montag aus dem Internet erreichbar wäre.
- Wir merken an, dass wir keine Sicherheit garantieren können.
- Als größte Gefahr wird ein (D)DoS eingeschätzt, der möglicherweise nur die VM abschießt, und nicht die gesamte Serverhardware lahmlegt.

#### Wettbewerbsjury

- Beim iGem Wettbewerb werden Juror\*innen wenig Zeit haben sich die WebSite anzusehen.
- Es wird angestrebt eine leicht zugängliche Demo auch ohne Log-in bereit zu stellen.

### Kommende Iteration

- Bilder: Müssen bis Montag angezeigt werden
- Infofragetyp: Nur Fragetext, keine Antwortmöglichkeiten, nur auf weiter drücken
- Kurs bearbeiten Seite: soll "übersichtlicher" werden: Module und andere Elemente sollen einklappbar sein
- Introseite: statische HTML Seite, die Hardgecoded auf dem Server liegt.
- Fragen beantworten: Navigation zwischen Fragen in einem Kurs soll möglich sein.

### Weitere Planung

- 
- Es wird angestrebt bis Ende August alle relevanten Features einzubringen und nach der zweiten Septemberwoche definitiv keine weiteren Features zu implementieren.

# Protokoll Auftraggebertreffen

## 21. August 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas Völcker  
BP Cloncademy

### User Stories

- Info Fragetyp einpflegen - abgenommen
- Info Fragetyp anzeigen - abgenommen
  - Bild sollte 2/3 der Seite einnehmen, der Text kommt auf die rechte Seite (siehe Designänderungen).
- Static Pages - abgenommen
  - Der Knopf in der Kopfleiste wird direkt auf die "About" Seite verlinken, die anderen statischen Seiten werden in der Fußleiste verlinkt.
  - Es soll folgende Seiten geben "Privacy Policy", "Terms", "Contact", "About", "Impressum"
- Statistik exportieren - abgenommen
- Dashboard im Kurs anzeigen - abgenommen
  - Kann größer gemacht werden (1/3 der Seite)
  - Weitere US:
    - \* Knopf der aufs Kursübergreifende Dashboard führt ("Zurück" Icon)
- \* Daneben: Moderator Name mit Profil
- \* Daneben: Dashboard schließen
- \* Kurs, Kursbeschreibung, Modultitel, Modulbeschreibung als ausklappbares Feld, Fragetitel (mit Navigations-Knöpfen)
- \* Aktuelle Frage sollte auch hervorgehoben werden
- \* Dashboard Knopf kleiner
- Zusätzliche Funktionalität: YouTube Videos
  - YouTube Videos können anstelle des Bildes in einen InfoText eingebaut werden.
  - Am Ende soll es die Möglichkeit geben, zwei Fragetypen auszuwählen "InfoText (YouTube)" und "InfoText (Bild)"
- Antwortmöglichkeiten Bilder einpflegen - abgenommen
  - In der Fragenbeantwortung sollten die Antwortbilder größer sein

### Änderungen & Neue Features

Alle Änderungen finden sich detail handschriftlich im Seafle!

- Navbar (von links nach rechts)
  - About/information
  - Learn
  - Profile
- Fragen Anzeigen
  - Fragebeschreibung und Fragestellung in zwei verschiedenen Boxen anzeigen
  - Daraus resultiert, dass es Fragetitel, Fragentext und Fragestellung gibt
  - Kursbeschreibung soll hinzugefügt werden
- Kursbeschreibung muss eine Zeichenbeschränkung erhalten (144 Zeichen?)
- Fußleiste in einer Frage
  - a) Dashboard (umbenannt zu Overview)
  - b) Fragetitel
  - c) Vor-Zurück Pfeile (in der Mitte Fragennummer/Anzahl von Fragen)
- Fußleiste außerhalb
  - Static Page Buttons
  - Copyright & Siteinformation (Name)
  - Social Media Accounts der iGEM Gruppe
  - Farbe: Wie Header

- 
- Profile
    - Dropdown aus dem Knopf
    - Nutzerbild (default wäre z.B. stylisierter DNA Strang)
    - Punktzahl (siehe Achievements)
    - "View Profile" Schaltfläche
    - Logout
  - Statistiken
    - Vom aktuellen Standpunkt aus 7 Tage zurück: Wie viele Fragen wurden beantwortet
    - Kreisdiagramm Aufteilung auf Kategorien
      - \* Kurse müssen Farben haben
  - Quiz
    - Nur die mittlere Spalte (also Antwortmöglichkeiten)
    - Nur die Fragestellungen werden angezeigt, keine Texte und Beschreibungen
    - Auswertung wie gehabt
    - Keine Möglichkeit zur Navigation
    - Jeder Kurs hat optional ein Quiz (min. 10 Fragen) am Ende
  - Optional ein Bild anzeigen
  - Die Antwort einer einmal beantwortete Quizfrage kann nicht nochmal geändert werden, es sei denn man startet das Quiz von vorne
  - Vorschläge der BP Gruppe
    - Password recovery in Profile or Login
    - Moderatorenrechte entziehen
      - \* Die Kurse können dann vorläufig nur noch von Admins bearbeitet werden
    - Adminrechte vergeben oder entziehen
    - Kategorien bearbeiten für Admins
      - \* Kategorien löschen benötigt eine Warnung, da alle Kurse in dieser Kategorie mit gelöscht werden.
    - Übersicht über die Kurse auf der Frontpage
      - \* Kurstitel und -beschreibung
      - \* aktuelles Modul
      - \* Knopf zum Fortführen
      - \* Fortschrittsanzeige
    - In Sidenav Moderatorenkurse anzeigen

---

#### Kommende Iteration

---

- Footer Redesign
- Besprochene Änderungen und Redesign (Dashboard, Infotext)
- Password recovery im Login
- Moderatorenrechte entziehen
- Adminrechte vergeben oder entziehen
- Kategorien bearbeiten für Admins
- "Learn" Seite Übersicht
- Statistik

# Protokoll Auftraggebertreffen

## 1. September 2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Claas Völcker  
BP Clonecademy

### Bericht der letzte Iteration

#### User Stories

- US 30: abgenommen
  - Sterne sollten auch an anderen Stellen verwendet werden
- US 31: abgenommen
- US 36: abgenommen
- US 37: abgenommen
  - Statische Seite "Privacy Policy" sollte eventuell noch hinzugefügt werden
  - Trennstriche zwischen den statischen Links
  - externe Links einfarbig (nur weisse Silhouetten)
  - Instagramm fehlt
  - "Dashboard" Button wird in "Overview" umbenannt
- US 40: abgenommen
- US 41: abgenommen

### User Stories - in progress

- Modulbeschreibung sollte im Sidenav nicht auftauchen und auch nicht zeichenbegrenzt sein
- Module sollten im Sidenav ein- und ausklappbar sein
- Fragenanzahl in der "Kurs bearbeiten" Übersicht ist eine gute Idee

### Anmerkungen und Verbesserungen

- Custom Feedback wird im Popup Fenster angezeigt

### Kommende Iteration

- Nutzerstudie 06.09. um 13:00 Uhr
- Serverdeployment
- weitere Altlasten aufarbeiten
- Quiz

# Protokoll Auftraggebertreffen

## 08.09.2017



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Leonhard Wiedmann  
BP Clonecademy

### Bericht der letzte Iteration

#### User Stories

- US 42 abgenommen
  - alle buttons oben außer x entfernen und x rechts alignen
- US 43: abgenommen
- US 44: abgenommen
- US 46: abgenommen
  - User kann sich unsichtbar im Ranking setzten
- US 31: abgenommen
- US 34: abgenommen
- US 33: abgenommen
  - farbe ändern wird eigene userstory (wenn colorpicker funktioniert)

### Fragen abschluss

- nach dem Quiz kommt ein Fenster mit feedback. Hier steht eine liste mit den richtgen und Falsch beantworteten Fragen.
- Feedback nach Fragen beantworten gibt ein Popup mit Feedback

### Curse löschen/invisible

- in der Kursübersicht kommt ein Button für visible/invisible (offenes/geschlossenes Auge). Der Button von geschlossen zu offen kann nur von Moderatoren benutzt werden, wobei der von offen zu geschlossen auch vom responsible Mod verwendet werden kann.
- in der Kursübersicht kommt ein Button für Admins, wodurch man den Kurs löschen kann

---

## **F Abschließende Bemerkungen des Teams zum Projekt**

---