CloneCademy

Qualitätssicherungsdokument

Gruppe:	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 Thea Lotz <lotz@bio.tu-darmstadt.de> Fachbereich Biologie
Abgabedatum:	xx.xx.xxxx

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Bachelor-Praktikum SoSe 2017
Fachbereich Informatik

Inhaltsverzeichnis

Anmerkungen und Hinweise

Die Verfasser dieses Dokumentes verwenden als Maßnahme für die Gleichstellung aller Geschlechter die sogenannte ”gendergerechte Sprache”. Wir weichen nur von dieser Regelung ab, wenn entweder alle Mitglieder einer Gruppe sich ein und demselben Geschlecht zugehörig fühlen oder die Vorgaben der Veranstalter*innen keine Alternative zulassen (z.B. beim 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 sowie 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.

Das Ziel des Projektes ist eine voll funktionsfähige Webanwendung für Rechner. Eine mobile Nutzung der Seite ist nicht Teil des Bachelor Praktikums, eine spätere Implementierung einer plattformübergreifenden Lösung (z.B. einer App) ist jedoch durch Nutzung der REST-Schnittstelle möglich.

Der minimale geplante Funktionsumfang umfasst die Möglichkeit einen Nutzungsaccount auf der Seite anzulegen und eingestellte Kurse zu bearbeiten. Außerdem wird es dedizierte Moderator*innen geben, die Inhalte auf der Webseite hochladen können.

Zusätzlich sind mehrere Erweiterungen geplant. Allem voran soll es möglich sein, unterschiedliche Fragetypen zu nutzen. Als minimales Ziel wird hierfür die Möglichkeit gegeben, Multiple Choice Fragen zu stellen. Weitere Typen werden im laufe des Projektes mit den Auftraggeber*innen bsprochen und beschlossen.

Da auch die Interaktion verschiedener Nutzer*innen und der Lernfortschritt erklärte Ziele der Plattform sind, werden zusätzliche Erweiterungen indiesen Gebeten geplant. Dazu zählen ausführliche Statistiken zu Lernfortschritt und die Möglichkeit, diesen mit anderen Nutzer*innen der Plattform zu teilen und zu vergleichen.

2 Qualitätsziele

2.1 Datensicherheit (Security)

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

Die größte Bedrohung geht von bekannten Webangriffen und Misskonfigurationen im Backend einer Webseite aus. Während dieses Projekts wird darauf geachtet, dass wir die Plattform gegen die wichtigsten Sicherheitslücken in einer Webanwendung sichern. Da eine umfassende Sicherung gegen alle möglichen Angriffsvektoren und Schwachstellen den Projektumfang weit übersteigen würde, befassen wir uns im Rahmen dieses Projektes vor allem mit der Sicherheit der Nutzeroberfläche und der Schnittstelle.

Die betrachtenten Angriffsvektoren[1] 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, folgen wir einem mehrschrittigen Plan.

Um die Durchführung der folgenden Maßnahmen sicherzustellen wurde ein Entwickler des Teams zum Sicherheitsbeauftragten ernannt. Seine Aufgabe ist es, auf die Einhaltung aller Programmierrichtlinien zu achten und Tests zur Validierung aller Schutzziele durchzuführen. Die genaue Vorgehensweise wird im Folgenden detailliert ausgeführt.

Während der ersten Gespräche mit den Auftraggeber*innen wurden Nutzerrollen definiert (Admin, Moderator*in und Nutzer*in). In jeder User Story wird festgehalten, ob und welche Zugriffsbeschränkungen einzelne Module der Webseite oder Daten besitzen. Um die Einhaltung der Nutzer*innenrollen zu ermöglichen, werden die zur Verfügung gestellten Authentifizierungswerkzeuge der genutzten Frameworks (Django, Django REST, Angular2) verwendet.

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

[1] Paraphrasiert nach der verbreiteten Liste 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

(OWASP)[2] verwendet. In den wöchentlichen internen Code-Reviews wird die Qualität des Codes hinsichtlich dieser Richtlinien überprüft.

Um die Qualität des Codes im Python-basierten Backend automatisiert zu testen, wird das verbreitete Werkzeug *bandit*[3] verwendet. Als statisches Analyse-Werkzeug für das Frontend wird *ng2lint*[4] verwendet. Beide Programme unterstützen die Code-Review, indem die gelieferten Meldungen als Grundlage für eine detaillierte Analyse des Codes genutzt werden. Zusätzlich wird eine Sicherheits-Checkliste[5] geführt, die die Best Practices zur sicheren Webentwicklung und das Überprüfen aller Reports der verwendeten Werkzeuge enthält. Um eine dauerhaft hohe Qualität des Codes zu gewährleisten, werden die Analyse-Tools automatisiert bei einem Push auf dem Development Branch des Git Repositories verwendet. Alle gefundenen Schwachstellen werden vor dem finalen Commit auf dem stable Branch zur Abnahme der User Story behoben. Wenn dies nicht möglich ist, wird eine 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*, welches eine Reihe bekannter Angriffe auf die Plattform ausführt und die gefundenen Schwachstellen meldet. Der Sicherheitsbeauftragte führt diese Tests monatlich durch und meldet die Ergebnisse an die jeweiligen Entwickler des betroffenen Moduls, damit diese innerhalb des nächsten Monats behoben werden können. Ist eine Behebung nicht möglich, werden die Konsequenzen dieser Lücke zusammen mit möglichen weiteren Schritten zum Schutz der Anwendung im Betrieb an die Auftraggeber*innen gemeldet.

Da ein Fokus auf Sicherheit zwar wichtig ist, aber Korrekturen erst vorgenommen werden können, sobald das Kernprodukt nicht mehr verändert wird, werden alle vorgestellten Maßnahmen ab Juli durchgeführt.

2.2 Bedienbarkeit (Entwurf)

CloneCademy muss für alle Benutzer*innen einfach und intuitiv bedienbar sein. Um dies zu gewährleisten, wird das Design aller Bereiche der Webseite nach diesem Kriterium angepasst. Die grundlegende Bedienung der Webseite wird ohne Einführung erlernbar sein und komplizierte Elemente werden mit einer ausreichenden Erklarung versehen. Die Lernplatform wird vor allem von Schülern*innen und Studenten*innen verwendet werden. Deshalb werden diese Gruppen beim Design schwerpunktmäßig beachtet.

Auch für dieses Ziel wurde ein Mitglied des Teams benannt, um die Nutzungsstudien durchzuführen und auf die Einhaltung aller Richtlinien zu achten.

[2] https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf
[3] https://wiki.openstack.org/wiki/Security/Projects/Bandit
[4] https://www.npmjs.com/package/ng2lint
[5] siehe Anhang

Um das oben formulierte Ziel zu erreichen, verwenden wir als Grundlage für das Design der Seite das Material Design [6] für Angular2. Um Knöpfe und andere Schaltflächen übersichtlicher zu machen, werden diese mit Icons[7] versehen. Um ein einheitliches Design nach den Wünschen der Auftrageber*innen zu gewährleisten, wird ein Designdokument[8] mit bespielhaftem Aufbau einzelner Seiten und Elemente der Webseite angefertigt.

Da die Bedienbarkeit der Oberfläche nicht objektiv gemessen werden kann, werden zuzüglich zu den Designrichtlinien Nutzer*innenstudien[9] durchgeführt. In den Studien werden die Proband*innen ohne vorherige Erklärung mit der Platform konfrontiert und aufgefordert, verschiedene Aufgaben zu erfüllen. Durch Aufzeichnen des Bildschirms während der Studie wird das Verhalten der Nutzer*innen erfasst um es später zu reflektieren und durch einen Fragebogen nach der Studie wird die persönliche Meinung der Proband*innen zusätzlich erfasst.

Um zum Schluss ein bedienbares Produkt zu haben, werden die Nutzer*innenstudien in mehreren Iterationen und mit unterschiedlichen Gruppen durchgeführt, um Verbessungsvorschläge aus den vorherigen Studien direkt wieder zu testen. Die Hinweise und Erkenntnisse aus den Studien wird verwendet, um das Designdokument zu überarbeiten und zu präzisieren.

2.3 Veränderbarkeit (Entwurf)

Für die Webanwendung CloneCademy ist es wichtig, dass die Anwendung im Nachhinein noch veränderbar ist. Es muss möglich sein, neue Inhalte in die Datenbank einzupflegen und auch den Quellcode der Webseite selbst ohne große Mühe erweitern zu können.

Um dieses Ziel zu erfüllen gehen wir wie folgt vor. Auf folgende Punkte wird im Projekt geachtet:

- Quellcode
- Kommentare
- Refactoring
- Wiki
- Codeanalyse Tools

Es wird ein Beauftragter für das Qualitätsziel gewählt, der für alle Rücksprachen oder Fehler der Ansprechpartner ist und dafür sorgt das die oben genannten Punkte wie unten beschrieben durchgesetzt werden.

Quellcode Beim implementieren der Userstories wird darauf geachtet das ein einheitliches von Anfang an bestimmtes Muster durchgezogen wird. Klassennamen, Funktionen, Variablen, Klammersetzung etc. Zum Beispiel: Klassennamen beginnen immer mit einem Großbuchstaben und

[6] hrefhttps://material.angular.io/https://material.angular.io/
[7] https://material.io/icons/
[8] siehe Anhang
[9] siehe Anhang

bei mehreren Wörtern als Klassenname, werden diese zusammen geschrieben und jedes weitere Wort beginnt wieder mit einem Großbuchstaben. Variablen beginnen immer mit einem Kleinbuchstaben und bestehen im besten Fall aus nur einem Wort. Funktionen beginnen wie Variablen mit einem Kleinbuchstaben, bei mehreren Worten wird wie beim Klassennamen jedes weitere Wort mit einem Großbuchstaben angefangen, das erste Wort aber mit einem Kleinbuchstaben.

Kommentare Der Code wird durchgehend in englisch kommentiert. Dabei wird beachtet das eine korrekte Sprache benutzt wird. Der beauftragte für das Qualitätsziel ist dafür zuständig bei abgeschlossenen Userstories als Abschluss über den Code zu schauen. Bei unzureichenden Kommentaren wird der Entwickler vom beauftragten kontaktiert und darauf hingewiesen diesem nachzugehen.

Refactoring Da bei einem Agilen Projekt es dazu kommen kann, dass der Code sehr schnell unübersichtlich werde kann. Wird vor Abnahme der Userstory ein refactoring vom jeweiligen Entwickler durchgeführt. Dies ist unerlässlich und in diesem Prozess werden unbenutzte Codefragmente entfernt und Quellcode Optimierungen durchgeführt.

Wiki Nebenbei wird eine Wiki regelmäßig aktualisiert, wo alle Schnittstellen zwischen Backend und Frontend definiert sind. Das Wiki kann von allen Gruppenmitgliedern editiert werden. Bei großen Änderungen und neuen Einträgen wird der Beauftragte für das Qualitätsziels kontaktiert, dieser schaut sich die Änderungen an und führt gegeben falls Optimierungen durch.

Codeanalyse Tools Für die Codeanalyse wird für das Backend und Frontend jeweils die dazugehörigen statischen Codeanalyse Tools verwendet:

- ng2lint für Angular2.
- django-lint für Django.

A Anhang

(Am Ende des Projekts nachzureichen)