



SEP – Sommersemester 2020

Moodle light: Online Plattform für Übungsqizze

Lastenheft

Christian Bachmaier, Armin Größlinger

## 1 Motivation

Die Bearbeitung von Übungsaufgaben ergänzend zu einer Vorlesung ist für Studierende essentiell, um den Stoff zu verstehen und einzuüben, z.B. in Hinblick auf eine Klausur. Als Ersatz oder zur Ergänzung solcher Übungen (an Universitäten oder ähnlich strukturierten, d.h. kursorientierten Bildungseinrichtungen) soll ein webbasiertes System erstellt werden, bei dem zeitslotorientiert (z.B. wöchentlich) online Übungsaufgaben (= *Quizze*) für Studierende zur Bearbeitung bereitgestellt werden können. Diese bearbeiten die Aufgaben dann fristgerecht ähnlich wie die traditionellen Übungsblätter, jedoch online. Die Bewertung erfolgt nach Ende der Bearbeitungsfrist halb-automatisch nach den Punktevorgaben der Aufgabensteller, welche aber anschließend Bewertungsänderungen durchführen oder Kommentare einfügen können. Nach Freigabe der Ergebnisse können die Studierenden die Bewertungen einsehen. Die erreichten Punkte können z.B. als Rückmeldung des aktuellen Leistungsstandes, Motivation, Zusatzpunkte oder Zulassungsvoraussetzung für die Klausur eingesetzt werden. Ob weiterhin eine klassische Präsenzübung abgehalten wird, z.B. zur Besprechung dieser oder anderer Aufgaben, lässt das System vollkommen unberührt.

## 2 Aufgabenstellung

Ziel des Software-Engineering Praktikums im Sommersemester 2020 ist es, eine Webapplikation zur Abhaltung von Übungsaufgaben organisiert in Quizze online zu erstellen. Um die Anwendung einfach betreiben und verwalten zu können, sollte sie auf eigenem Webspaces ausgeführt werden können. Die Anwendung sollte für einen Betreiber leicht einzurichten sein. Außerdem sollte dieser die Möglichkeit haben, gewisse Konfigurationen der Anwendung vornehmen zu können, wie zum Beispiel das Festlegen des Logos, des Namens der betreibenden Einrichtung, etc. Der oder die Betreiber fungieren als Administratoren des Systems, z.B. technische Verwaltungsmitarbeiter einer Universität. Weiteren Zugang zum System haben nur registrierte Nutzer, d.h. Dozenten und Studierende, nach entsprechender Authentifizierung.

Die zu erstellende Web-Anwendung soll intuitiv zu bedienen sein, geringe Anwendungshürden aufweisen, zugleich aber fortgeschrittenen Benutzern eine Vielzahl von erweiterten Funktionen mit einfacher Bedienung bieten.

Prominentestes Beispiel für ein ähnliches und bereits existierendes System ist das u.A. an der Universität Passau eingesetzte *Moodle*<sup>1</sup>. Aber z.B. auch *Google Classroom*<sup>2</sup> oder *Schoology*<sup>3</sup> beinhalten die Möglichkeit für online Übungsaufgabe, um nur einige wenige der vielen existierenden zu nennen. Diese gehen jedoch durchgehend als komplette interaktive Lernplattformen/Lernmanagement Systems (LMS) deutlich über die hier geforderte Funktionalitäten hinaus.

Die Anwendung muss mit Java und Jakarta Faces (JF) entwickelt werden. Auf Benutzerseite kann ein moderner Web-Browser vorausgesetzt werden.

## 3 Produkteinsatz

Zur Zielgruppe der zu entwickelnden Anwendung gehören als Betreiber verschiedenste Arten von universitätsähnlichen Bildungseinrichtungen. Eine Instanz des System wird einrichtungsintern für eine geschlossene Benutzergruppe betrieben.

Ziel ist kein universelles Lernmanagementsystem für Bildungseinrichtungen, mit Funktionalitäten wie z.B. Vorlesungsverzeichnis, Kursanmeldung, Bereitstellung von Studienordnungen, Verteilung von Vorlesungsunterlagen oder Onlinekurse. Allerdings wäre denkbar, das entstehende Produkt in Folgeprojekten dahingehend zu erweitern.

## 4 Produktfunktionen

### 4.1 Begriffsdefinitionen

Es gibt mehrere Gruppen von Benutzern (= Akteure) des Systems:

---

<sup>1</sup><https://moodle.uni-passau.de/>, <https://moodle.de/>

<sup>2</sup><https://classroom.google.com/>

<sup>3</sup><https://www.schoology.com/>

**Administratoren** Sind (meist als Mitarbeiter der Verwaltung) für die Konfiguration des System verantwortlich. Können beispielsweise das Logo oder den Namen der Webapplikation festlegen aber auch technische Einstellungen vornehmen. Wichtigste weitere Aufgabe eines Administrators ist die Benutzerverwaltung. Ein Administrator hat jedoch allumfassende Rechte im System, auch an Kursen und Quizze.

**Dozenten** Können (meist wochenbasierte) Quizze anlegen, Teilnahmelisten von eigenen Kursen und Quizzen einsehen, editieren und von Studierenden erbrachte Leistungen darin verbuchen.

**Studierende** Sind angemeldete Nutzer, die an Kursen und Quizzen nach Authentifizierung über ein Einmalpasswort oder direkte Einladung teilnehmen können.

**Anonyme Benutzer** Haben die Möglichkeit sich im System zu registrieren nach Überprüfung der Validität der E-Mail-Adresse, die je nach systemweiter Konfiguration zur Einrichtung gehören muss.

## 4.2 Minimale Anforderungen

Im Folgenden ist mit Erstellen, wenn nicht anders explizit erwähnt, gleichzeitig auch nachträgliches Editieren und Löschen gemeint.

**Kurserstellung** Dozenten können Kurse mit einem aussagekräftigen Namen und mit Start und Endzeiten (z.B. Vorlesungszeit eines Semesters) erstellen. Bel. registrierte Nutzer können von Ihnen dann als Dozenten innerhalb dieses Kurses hinzugefügt werden. Alle diese Personen haben dann volle Rechte in diesem Kurs. Bei der Erstellung oder bei nachträglicher Konfiguration des Kurses kann ein Einmalpasswort festgelegt werden, welches den Zugang zum Kurs für Teilnehmer/Studierende regelt. Auch können Studierende direkt zum Kurs eingeladen/angemeldet werden. Beispielsweise kann dieses der Dozent in der zugehörigen Vorlesung verkünden.

Optional kann ein fester Zeitraum angegeben werden, nur in welchem eine An- oder Abmeldung zum Kurs möglich ist.

**Quizerstellung** Dozenten eines Kurses können Quizze erstellen. Ein Quiz ist eine Sammlung von nummerierten Aufgaben, die nach einem Startzeitpunkt und einer Deadline von zum Kurs angemeldeten Studierenden bearbeitet werden können. Typischerweise werden Quizze wöchentlich abgehalten.

**Aufgabenerstellung** Innerhalb eines Quiz kann ein Dozent dessen Aufgaben erstellen. Es gibt jeweils eine Aufgabenbeschreibung und einen Antworttyp, der mindestens Multiple-Choice, Auswahl aus einer Liste/Drop-Down-Box, Antwort als Zahl und Freitextantwort umfasst. Für jede Antwort können maximal zu erreichende Punkte angegeben werden. Bei Multiple-Choice können pro Antwortmöglichkeit Punkte vergeben werden (auch negativ, wobei eine Aufgabe insgesamt aber keine negativen Punkte liefern soll). Mögliche Antworten in einer Liste/Drop-Down-Box

können auch mehrere richtige (wie falsche) Antworten enthalten aber eine richtige auszuwählen gibt die zu erreichenden Punkte. Optionale weitere Antworttypen sind z.B. Kurzantwort in Textform, Antwort wahr/falsch oder Lückentexte.

Beachten Sie, sowohl bei der Aufgabenbeschreibung als auch bei den möglichen Antworten von Multiple-Choice soll einfacher Text und/oder ein Bild (z.B. für Formeln), aber auch zu rendernder HTML-Code ggf. inkl. Bildern angegeben werden können.

Aufgaben können jederzeit bearbeitet oder gelöscht werden.

**Quizteilnahme** Ein Teilnehmer eines Kurses kann ein Quiz innerhalb dessen Bearbeitungszeitrahmen absolvieren. Eine nachträgliche Abänderung der vorausgefüllten/schon gegebenen Antworten ist jederzeit möglich, soweit die Bearbeitungsdeadline noch nicht verstrichen ist. Es zählen stets die zuletzt gegebene Antworten. Falls ein Teilnehmer ein Quiz oder Aufgabe innerhalb eines Quizzes nicht beantwortet, werden alle bzw. die nicht beantworteten Aufgaben mit 0 Punkte bewertet. Nach Verstreichen der Abgabefrist, kann eine Lösung zwar eingesehen, aber nicht mehr verändert werden.

**Bewertung** Nach Abgabefristende kann ein Kursdozent eine Abgabe für ein Quiz (nach-)bewerten. Multiple-Choice, Listenauswahl und Zahlantworten sind dabei vom System schon mit Punkten vorausgefüllt, weil es bei diesen Antworttypen nur wahr oder falsch gibt. Für Freitextantworten ist dies nicht mgl., allenfalls mit 0 Punkten wenn keine Antwort gegeben wurde oder mit voller Punktezahl, wenn glücklicherweise genau der Text aus der eingerichteten Musterlösung getroffen wird. Zu jeder Frage kann der Dozent einen Bewertungskommentar abgeben. Alle Bewertungen werden gesammelt für alle Teilnehmer erst nach einer expliziten Freigabe einsehbar.

Als Komfortfunktion werden von Teilnehmern überhaupt nicht bearbeitete Quizze den Kursdozenten zwar angezeigt zur Bewertung, aber speziell markiert. I.d.R. wird dieser diese "Nicht-Abgaben" dann gar nicht durchsehen, weil sowieso alles mit 0 Punkten bewertet wird.

**Bewertungseinsicht** Die Bewertung einer Abgabe ist nach Freigabe durch einen Dozenten für einen Teilnehmer einsehbar. Es ist dabei ersichtlich, welcher Dozent die Bewertung erstellt hat.

**Anmeldungen** Studierende können sich an Kursen anmelden, wenn Sie das ggf. notwendige Einmalpasswort dazu kennen oder vom Dozenten angemeldet wurden. Eine Abmeldung vom Kurs ist jederzeit möglich, wobei aber jegliche erreichte Punkte verfallen.

Bei einem vom Kursdozent angegebenen fixen Anmeldezeitraum ist obiges nur zeitlich innerhalb dessen möglich.

**Kursübersicht/-Suche** Eigene Kurse (als Dozent als Ersteller oder als Teilnehmer) werden den Benutzern in einer Tabelle übersichtlich dargestellt. Desweiteren

kann aber nach allen Kursen gesucht werden oder aus einer großen Liste ausgewählt werden.

**Quizübersicht** Innerhalb eines Kurses, bei dem man angemeldet ist gibt es eine Übersicht aller bereits veröffentlichten Quizze.

**Registrierung** Personen, insb. Studierende, können sich selbst am System registrieren mit ihrer Matrikelnummer, Namen und einer gültigen E-Mail-Adresse. Diese muss je nach Systemeinstellung einem gewissen Suffix genügen, um nur Mitglieder der eigenen Bildungseinrichtung zuzulassen. Es muss verifiziert werden, dass die E-Mail-Adresse tatsächlich der registrierenden Person zugeordnet ist.

Die Anmeldung zu einem konkreten Kurs erfolgt erst in einem zweiten/weiteren Schritt.

**Dozenten** Die Administratoren können Benutzern Dozentenrechte verleihen, die das Anlegen von Kursen ermöglichen. Als Kursdozent muss man nicht notwendigerweise über (globale) Dozentenrechte verfügen.

**Quizübersicht/-statistik** Jeder Teilnehmer eines Kurses kann sich eine Übersichtstabelle anzeigen lassen, um zu sehen wieviele von wievielen möglichen Punkten pro Quiz (und pro dessen Aufgabennummer) erreicht worden sind. Ein Dozent hat ebenfalls eine Übersicht, aber hier aufgeschlüsselt nach allen Teilnehmern und den Quizzen eines Kurses. Die Einträge der Tabellen sind mit den konkreten Lösungen/Abgaben entsprechend per Hyperlink verknüpft. Pro Quiz ist jeweils der zeitliche Beginn und das Ende der Bearbeitung durch einen Teilnehmer ersichtlich.

**Customizing** Die Anwendung sollte die Möglichkeit bieten die jeweilige betreibende Einrichtung zu präsentieren (Name, Logo, Farbschema, Kontaktmöglichkeiten, etc.) bzw. es an ein bereits etabliertes Look & Feel anzupassen.

**Benutzerverwaltung** Im System gibt es i.W. drei verschiedene Benutzerrollen: Administratoren, Dozenten und registrierte Nutzer. Der Admin kann alle Benutzer und die damit verbundenen Rollen verwalten: Anlegen neuer Benutzer, vollständiges Bearbeiten und Löschen existierender Benutzer, Zuordnen/Entziehen von Admin- und Dozentenrechten und Verwalten von Kursanmeldungen. Eine Funktion zum Suchen nach bestimmten Benutzern erleichtert für einen Admin die Verwaltung.

**Login** Ein Benutzer kann sich in ein Benutzerkonto einloggen und ist damit angemeldet. Hierfür ist eine Authentifizierung notwendig.

**Kontoverwaltung** Ein angemeldeter Benutzer kann sein Benutzerkonto editieren und eigenmächtig löschen. Letzteres bedingt aber einen Verlust von sämtlichen Punkten in ggf. noch laufenden Kursen.

**Abmeldung** Ein angemeldeter Benutzer kann sich abmelden.

**Online-Hilfe** Der Benutzer muss zu jedem Zeitpunkt schnellen Zugriff auf die für die aktuelle Seite relevante Online-Hilfe haben.

## 5 Nicht-Funktionale Anforderungen

Folgend die Produktleistungen und Qualitätsanforderungen.

### 5.1 Usability

1. Einfache und intuitive Bedienbarkeit des Systems.
2. Die Benutzeroberfläche soll sich an allgemein geläufigen Bedienkonzepten und den damit verbunden Funktionen orientieren; alle Tabellen sollen z.B. nach den dargestellten Spalten sortierbar sein; Tabellen, die eine gewisse Größe überschreiten, sollen, um kurze Seitenladezeiten zu ermöglichen, auf mehrere Seiten aufgeteilt werden (Pagination).
3. Häufig wiederkehrende Aufgaben sind durch das System möglichst benutzerfreundlich zu unterstützen; häufig genutzte Funktionen sind möglichst einfach zugänglich zu machen.
4. Die Seiten der Applikation sind übersichtlich und einfach verständlich zu gestalten.
5. Daten sollen nicht nur leicht auffindbar und gut lesbar sein, sondern auch leicht einzugeben.
6. Bei Fehleingaben in ein HTML-Formular und der darauf folgenden Korrektur sollen die zuvor eingetragenen Felder nicht erneut einzugeben sein, sondern schon vorbesetzt sein. Außerdem sollte die Überprüfung der Eingaben nicht nach der ersten fehlerhaften Eingabe abbrechen, sondern alle Eingaben überprüfen und eine akkumulierte Fehlermeldung an den Benutzer zurückgeben.
7. Das System soll mit steigender Last skalieren.

### 5.2 Datensicherheit

1. Alle im System erfassten Daten sind persistent in einer Datenbank abzulegen; die Konsistenz der Daten ist sicherzustellen (Mehrbenutzerbetrieb!). Speziell, wenn Änderungen über mehrere Datenbanktabellen hinweg vorgenommen werden, sind Transaktionen zu nutzen.
2. Für die persistente Speicherung der Daten soll eine Datenbank (PostgreSQL) verwendet werden, die auf den Infosun-Rechnern zur Verfügung gestellt wird. Als Referenzplattform für die Implementierung dienen die Rechner im CIP-Pool.

3. Beim Löschen von Daten sind die Abhängigkeiten zwischen Daten einzelner Tabellen zu berücksichtigen; hat das Löschen eines Datensatzes das Löschen anderer Datensätze zur Folge, muss der Benutzer vorher deutlich darauf hingewiesen werden!

### 5.3 Datenschutz

1. Es muss sichergestellt werden, dass durch das System zu keinem Zeitpunkt sensible Daten für unberechtigte Dritte zugänglich sind.
2. Es sollen möglichst wenige technische Informationen über das System nach außen gegeben werden.
3. Alle personenbezogenen Daten, wie z.B. Login-Daten, sind sensibel und dürfen daher nur per SSL-Verbindung übertragen werden.
4. Passwörter dürfen nicht im Klartext gespeichert werden.
5. Die Nutzerdaten sind so zu speichern, dass kein unautorisierter Zugriff durch Dritte oder durch andere Nutzer im System stattfinden kann.
6. Das System darf nur im Rahmen der oben genannten oder vorgesehenen Funktionalitäten verändert werden.
7. Änderungen durch Manipulationen mit bekannten Angriffsmethoden wie SQL-Injection oder Cross-Site-Scripting müssen ausgeschlossen werden. Es müssen außerdem Maßnahmen ergriffen werden, um die Sitzungen der einzelnen Nutzer zu schützen (Session-Hijacking).

### 5.4 Internationalisierbarkeit

1. Für die Texte auf der Website ist die Zeichenkodierung UTF-8 zu wählen.
2. Die Sprache des Systems kann Deutsch oder Englisch sein; eine mehrsprachige Implementierung ist optional.

### 5.5 Weiterentwickelbarkeit

Das zu erstellende System soll flexibel gegenüber zukünftigen Erweiterungen sein. Eine einfache und kostengünstige Weiterentwickelbarkeit des Systems ist sicherzustellen.

### 5.6 Installation

Es soll eine komfortable Installation für Systembetreiber bereitgestellt werden. Die Installation sollte einfach und schnell sein und automatisch das entsprechende Datenbank-Setup vornehmen.

## 6 Ergänzungen

Die Benutzung des Systems sollte mit allen gängigen Web-Browsern möglich sein. Wir raten allerdings davon ab, für jeden Browsertyp unterschiedlichen HTML-Code zu generieren. Der HTML-Code soll logisches Markup darstellen und nicht dazu missbraucht werden, eine bestimmte graphische Darstellung zu erzwingen. Deswegen dürfen Features wie Frames nicht und explizites Javascript nur nach Absprache eingesetzt werden. Der HTML-Code muss HTML-konform sein und z.B. durch <http://validator.w3.org/> validierbar sein. Die Verwendung von Cascading Stylesheets (CSS) wird dringend angeraten.

Die Sessionverwaltung darf die Verwendung von Cookies nicht erzwingen.

Das System muss ein Log über alle Fehler führen, um das Debugging und den Betrieb der Anwendung zu vereinfachen. Achten Sie darauf, dass die Fehlerbeschreibungen detailliert genug sind, um auf einen Fehler bzw. dessen Ursache schließen zu können.

## 7 Leistungsnachweis

Mit der funktionalen Umsetzung o.g. Musskriterien kann die Note 1,0 erreicht werden. Wunschkriterien sind (wie der Name sagt) wünschenswert, aber deren Umsetzung nicht zwingend notwendig für das Erreichen der vollen Punktzahl.