

SOFTWARETECHNIKPRAKTIKUM

GRUPPE: SWP14-PRUEF

Entwurfsbeschreibung

Verfasser:
Kevin Schramm, Mirko
Schulze, Michael Rosenthal

Abgabe der Lösung zur
Endabnahme
19.05.2014

Inhaltsverzeichnis

1 Einleitung

Das vorliegende Dokument beschäftigt sich mit der Realisierung eines Prüfungsverwaltungssystems für die veterinärmedizinische Fakultät der Universität Leipzig. Es soll einen Einblick in die Funktionsweise, den Aufbau und die geplante Umsetzung der Software geben. Dabei wird versucht die relevanten Entscheidungen zu begründen und zu erläutern, welche Überlegungen das Team im Hinblick auf den Entwurf angestellt hat. So soll dem an der Einarbeitung Interessierten die Möglichkeit gegeben werden, die Zusammenhänge hinreichend gut zu verstehen.

2 Produktübersicht

Das System wird als Webanwendung realisiert, um eine ständige Verfügbarkeit für die zahlreichen Nutzer zu gewährleisten. Da bei einer Prüfungsverwaltungssoftware selbstredend ein hohes Maß an Sicherheit und Datenschutz garantiert sein muss, ist eine zuverlässige Authentifizierungsfunktion unentbehrlich. Bei der Authentifizierung wird dem Benutzer schließlich eine Rolle zugewiesen. Diese Rollen sind Prüfling und Prüfungsamt. Nach dem Login gelangt der Benutzer auf eine rollenspezifische Oberfläche, von welcher er die eigentliche Funktionalität anwenden kann. Ein Prüfling hat dabei die geringste Berechtigung. Er darf lediglich die Ergebnisse seiner eigenen MC-Prüfungen betrachten. Hingegen haben die Nutzer der Rolle Prüfungsamt sämtliche Rechte inne. Ihnen allein ist es möglich, eine CSV-Datei als Prüfungsergebnis zu importieren. Alle weiteren Prüfungsergebnisse aus dem System dürfen sie einsehen und bearbeiten. Ferner wird es die Möglichkeit geben, zur Prüfung Kommentare zu verfassen. Als detaillierte Funktion steht außerdem die sogenannte „optionale Nichtwertung“ zur Verfügung. Diese bewirkt, dass einzelne Teile der Klausur bei Unterschreiten eines bestimmten Leistungsgrades durch alle Prüfungsteilnehmer nicht zur Bewertung herangezogen werden. Dazu sind umfassende Daten nötig, die aus den CSV-Dateien ausgewertet werden müssen. Deshalb ist ebenfalls zumindest eine einfache Statistik-Funktion vorgesehen.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

Nachdem im Recherchebericht die Vor- und Nachteile der einzelnen Optionen ausführlich erläutert wurden haben wir uns für PHP als grundlegende Programmiersprache entschieden. Als Framework wird das objektorientierte Framework Zend 2 (im folgenden „Zend“ genannt) genutzt. Damit einhergehend wurde das Projekt dem MVC-Pattern entsprechend entworfen. Somit existieren für jede Rolle entsprechende Controller die auf die Unterschiedlichen Nutzereingaben reagieren. Außerdem wurde entsprechend dem Stil von Zend für jedes Event ein eigener View angelegt. Des weiteren existieren, für den Nutzer nicht sichtbare, Models in denen Datenbankzugriffe und weitere Programmlogik organisiert wird.

Die Benutzeroberfläche wird als HTML-basierte Website dargestellt. Das Layout wird durch CSS und Javascript definiert.

Die Authentifizierung erfolgt über den universitätseigenen Dienst Shibboleth. Zur Authentifizierung werden Benutzername und Passwort an das URZ übermittelt. Als Antwort kommt lediglich „ja“ oder „nein“, im Sinne von „Ist ein gültiger Student“. Zusätzlich erhalten wir die E-Mail Adresse um dem Nutzer seine entsprechende Rolle zuzuordnen zu können. So wird die Anonymität der Benutzerdaten gewährleistet.

Alle Daten werden in einer MySQL-Datenbank gespeichert.

Die Datenbankbindung erfolgt über den Objekt-relationalen Mapper Eloquent. Dieser sorgt dafür, dass die Daten aus den Datenbanktabellen in die benötigten Models gefüllt werden bzw. Änderungen in der Datenbank gespeichert werden.

4 Struktur- und Entwurfsprinzipien einzelner Pakete

Jede Grundfunktionalität (Prüfungsamt, Student) erhält einen eigenen Zend-Controller angelegt. Diese können Daten aus den jeweiligen Models (die mit den Datenbanktabellen korrespondieren) aggregieren, anzeigen und verändern.

4.1 Authentifizierung

Nach der erfolgreichen Authentifizierung über Shibboleth wird dem User die ihm entsprechende Rolle zugewiesen. Falls noch nicht vorhanden, wird dazu außerdem in der Datenbanktabelle `user` ein Eintrag angelegt. Dabei erhält er eine neue Identifikationsnummer für interne Zwecke (die in anderen Tabellen referenziert werden kann).

Die Authentifizierung erfolgt über das Modul `Simplesamlphp`.

4.2 Prüfungsdateien einladen

Die Funktionen dieses Pakets stehen nur für Nutzer der Rolle "Prüfungsamt" zur Verfügung.

Eingelesene CSV-Dateien werden entsprechend des vorgegebenen Schemas ausgewertet. In der Datei sind die Antworten für jeden Teilnehmer der Prüfung und jede Frage aufgelistet. Für jede Prüfung wird ein Eintrag in `exam` angelegt, für jede Frage ein Eintrag in `exam_question`. In letzterem kann festgehalten in welchen Fragenkomplex(`subject`) die Frage gewertet wird. Jede einzelne Antwort jedes Studenten erhält einen Eintrag in `exam_question_result`. Dort wird (ggf. nach Nutzereingabe) festgehalten, ob die Antwort korrekt ist, ob die Frage für diesen Nutzer gewertet wird und ob die Antwort für den Teilnehmer freigegeben ist.

4.3 Mapping zwischen Matrikelnummern und E-Mail-Adressen erstellen

Der universitätsinterne Shibboleth-Server stellt zu Nutzern das einzige personenbezogene Merkmal E-Mailadresse zur Verfügung. In den vorliegenden Prüfungsdaten werden allerdings die Studenten über ihre Matrikelnummer identifiziert. Um diese Daten zu verknüpfen hat das "Prüfungsamt" die Möglichkeit solche Mappings anzulegen. Diese werden in der `student` festgehalten, in der auch alle anderen Studenten mit Prüfungsergebnissen stehen.

4.4 Prüfungen eines Studenten betrachten

Für jeden Studenten ist es für Nutzer der Rolle "Prüfungsamt" sowie dem Studenten selbst möglich, die Ergebnisse aller von ihm abgelegten Prüfungen zu betrachten. Dazu werden alle als freigegeben markierten Antworten aus `exam_question_result` über die `exam_question` mit dem jeweiligen `exam` referenziert und ausgegeben.

4.5 Externe Anrechnungen eintragen

Für jeden Studenten können über eine Maske weitere Ergebnisse eingetragen werden (z.B. für an anderen Unis abgelegten Prüfungsleistungen). Diese werden in der Tabelle `external_result` festgehalten. Dort werden die Daten in der Form *Erreichte Punktzahl* und *Erreichbare Punktzahl* gespeichert.

4.6 Noten berechnen

Die Noten der Studenten werden auf Grundlage der vorliegenden Daten automatisch berechnet und angezeigt.

5 Datenmodell

Die Daten werden in einer MySQL-Datenbank gemäß dieses, schon beschriebenen, Schmas gespeichert.

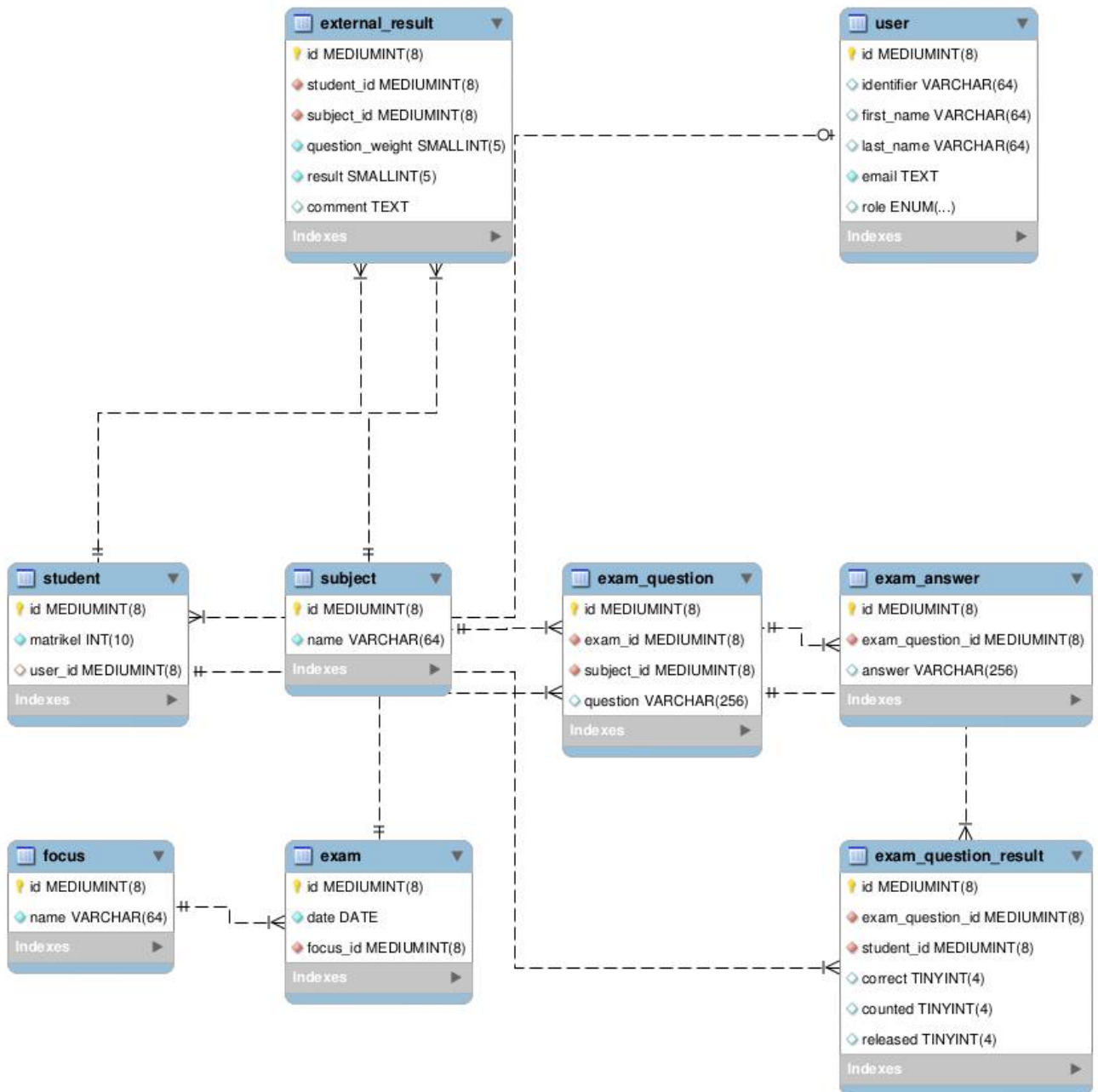


Abbildung 1: Datenmodell

6 Testkonzept

Im Sinne des bereits vorhandenen umfassenden Testkonzeptes wurden die Tests direkt in die Implementierung mit eingebunden. Auf erster Ebene wurden die Komponenten und Methoden vom jeweiligen Implementierenden in eigener Verantwortung getestet. Dafür wurden mithilfe des Testframeworks von Zend Testkomponenten erstellt. Anschließend wurden die Komponenten insgesamt einem Integrationstest unterzogen. Ferner wurde geprüft, ob die Gesamtfunktionalität der Softwarestudie vollständig und korrekt umgesetzt wurde. Dabei wurden auch verschiedene Testdurchläufe ausgeführt und die entstandenen Ergebnisse mit den gewünschten Ergebnissen verglichen.

Auf der letzten Testebene musste schließlich auch die Lauffähigkeit der Software auf dem Server geprüft werden. Auch die Weboberfläche als Benutzerschnittstelle, wurde wiederholt auf ihre Funktionen hin getestet.

7 Glossar

7.1 Authentifizierung

Authentifizierung ist der Nachweis einer behaupteten Eigenschaft einer Entität. Im Kontext einer Webanwendung bezeichnet die Authentifizierung den Nachweis, berechtigter Benutzer eines Benutzerkontos zu sein. Eine bestehende Zugangsberechtigung eines Benutzers wird bei erfolgreicher Authentifizierung anerkannt und erlaubt dem Benutzer weitere Aktionen. Die abschließende Bestätigung einer Authentifizierung wird als Autorisierung bezeichnet.

7.2 Benutzerkonto

Ein Benutzerkonto ist eine Zugangsberechtigung zu einem zugangsbeschränkten Server. Ein Benutzer muss sich üblicherweise authentifizieren, um autorisierten Zugriff auf Daten, die der Server zur Verfügung stellt, zu erhalten.

7.3 CSV-Datei

Das Dateiformat CSV steht für Comma-separated values oder Character-separated values und beschreibt den Aufbau einer Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten. Die Dateinamenserweiterung lautet „.csv“. In CSV-Dateien können Tabellen abgebildet werden.

7.4 Zend Framework 2

Das Zend Framework 2 ist ein komponenten-orientiertes Framework für PHP 5. Klassen und Pakete können unabhängig voneinander und auch in Kombination mit den Lösungen anderer Hersteller genutzt werden. Hersteller ist das Unternehmen Zend Technologies.

7.5 Eloquent

Eloquent ist ein Object-relational mapper, d.h. es bildet Model-Klassen auf Datenbanktabellen und umgekehrt ab, ohne dass man sich um einfache Datenbankverbindungen und -abfragen kümmern muss. <https://github.com/illuminate/database>

7.6 simpleSAMLphp

simpleSAMLphp ist eine PHP-Implementierung des Shibboleth-Authentifizierungsprotokolls. <https://simplesamlphp.org>