

Entwicklung einer Lern-Feedback-Plattform für Vorlesungen

Lecture Monitoring - LeMon

Studienarbeit

Semester 5 + 6

Studiengang *Informatik*

Vertiefungsrichtung *Angewandte Informatik*

an der
Dualen Hochschule Baden-Württemberg / Stuttgart

von

Ephraim Petry & Nick Herrmannsdörfer

06. Juni 2014

Bearbeitungszeitraum

Matrikelnummer Ephraim Petry, Kurs

Matrikelnummer Nick Herrmannsdörfer, Kurs

Betreuer der Praxisphase

zwei Semester

2767400, TINF11D

1655361, TINF11D

Frau Dr. Barbara Dörsam

Name: Ephraim Petry & Nick Herrmannsdörfer
Matrikel-Nr Ephraim Petry: 2767400
Matrikel-Nr Nick Herrmannsdörfer: 1655361
Kurs: TINF11D

Titel der Arbeit:

Entwicklung einer Lern-Feedback-Plattform für Vorlesungen:
Lecture Monitoring - LeMon

Erklärung

Ich versichere hiermit, dass ich das vorliegende Dokument selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Stuttgart, 06. Juni 2014

EphraimPetry&NickHerrmannsdörfer

Zusammenfassung

Im Rahmen dieser Studienarbeit sollte ein System erstellt werden, mit dem der Lernfortschritt und das Verständnis der Studenten an der Hochschule der Medien (HDM) für die in der Vorlesung vermittelten Inhalte abgeprüft werden sollte. Dieses System sollte Lecture Monitoring (LeMon) heißen.

Das System sollte dabei den kompletten Prozess von der Erstellung einzelner Übungsblätter durch den Dozenten, das Management der Fragen für die Übungsblätter, sowie das Freischalten der Übungsblätter zur entsprechenden Zeit beinhalten. Desweiteren sollte der Dozent den Studenten auf einfache Art und Weise das elektronische Übungsblatt zugänglich machen. Hierzu sollte bevorzugt ein QR-Code eingesetzt werden, den die Studenten mit ihrem Smartphone und einer entsprechenden App sofort einlesen und zum Übungsblatt gelangen konnten.

Am Ende sollte der Dozent auf einfache Weise die Möglichkeit haben, die Ergebnisse wenige Sekunden nach Ablauf der Zeit anschaulich angezeigt zu bekommen und diese mit den Studenten durchsprechen.

Zur Umsetzung dieses Projektes sollte eng mit einer Gruppe mit einem ähnlichen Projekt zusammengearbeitet werden, die jedoch eine etwas andere Art von Übungsblättern für den Einsatz von Aufgaben innerhalb oder auch außerhalb der Vorlesung erstellen sollte. Soweit möglich sollten dabei Synergien genutzt und eine gemeinsame Oberfläche für beide Projekte erstellt werden.

Vorwort

Dies ist die Studienarbeit von Ephraim Petry und Nick Herrmansdörfer. Sie ist bestandteil des dritten Studienjahres des Bachelor-Studiums an der Duale Hochschule Baden Württemberg (DHBW) Stuttgart.

Diese Studienarbeit sollte nicht nur den Zweck der Absolvierung einer Studienarbeit erfüllen. Ein besonderes Augenmerk lag auch darauf, dass bestehende oberflächliche Wissen in bestimmten Technologien in als praktisches Projekt umzusetzen und zu vertiefen.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Aufgabenstellung	7
1.2	Zielsetzung	8
2	Anforderungsanalyse	9
2.1	Use-Cases	9
2.1.1	UC1: Frage mit Musterlösungen anlegen	9
2.1.2	UC2: Kategorie anlegen	11
2.1.3	UC3: Vorlesung anlegen	12
2.1.4	UC4: Arbeitsblatt anlegen	13
2.1.5	UC5: Antworten abschicken	15
2.1.6	UC6: Auswertung des Übungsblattes	17
2.2	Allgemeine Anforderungen	19
2.3	Technologische Anforderungen	19
2.4	Eingesetzte Technologien	19
3	Vorüberlegungen	21
3.1	Datenmodell	21
3.2	Funktionsumfang	24
3.3	Benutzeroberfläche	24
4	Implementierung	25
4.1	Struktur des Projekts	25
4.2	Admin-Bereich	25
4.2.1	Struktur der Implementierung	25

4.2.2	Eingesetzte Libraries und Frameworks	26
4.2.3	Besonderheiten der Implementierung	26
4.3	LeMon-Bereich	27
4.3.1	Struktur der Implementierung	27
4.3.2	Eingesetzte Libraries und Frameworks	28
5	Fazit und Ausblick	29
6	Anhang	30
	Glossar	I
	Abbildungsverzeichnis	II
	Tabellenverzeichnis	III
	Quellcodeverzeichnis	IV

1 Einleitung

1.1 Aufgabenstellung

Im Rahmen dieser Studienarbeit sollte ein System erstellt werden, mit dem der Lernfortschritt und das Verständnis der Studenten an der Hochschule der Medien (HDM) für die in der Vorlesung vermittelten Inhalte abgeprüft werden sollte. Dieses System sollte Lecture Monitoring (LeMon) heißen. Dabei sollte ein Dozent eine Art Fragebogen für Studenten vorbereiten, der in den ersten Minuten der Vorlesung von den Studenten durchgearbeitet werden sollte. Die Studentenantworten sollten soweit möglich automatisiert ausgewertet oder zumindest anschaulich in einer Übersicht dargestellt werden, die anschließend am Beamer gezeigt und durch die Lehrkraft besprochen werden sollte. In diesem Schritt sollten sich dann Unklarheiten der Studenten herauskristalisieren. Der Dozent sollte dabei merken, welche Themen einer Wiederholung gedürfen und welche nicht.

Das System sollte dabei den kompletten Prozess von der Erstellung einzelner Übungsblätter durch den Dozenten, das Management der Fragen für die Übungsblätter, sowie das Freischalten der Übungsblätter zur entsprechenden Zeit beinhalten. Desweiteren sollte der Dozent den Studenten auf einfache Art und Weise das elektronische Übungsblatt zugänglich machen. Hierzu sollte bevorzugt ein QR-Code eingesetzt werden, den die Studenten mit ihrem Smartphone und einer entsprechenden App sofort einlesen und zum Übungsblatt gelangen konnten.

Am Ende sollte der Dozent auf einfache Weise die Möglichkeit haben, die Ergebnisse wenige Sekunden nach Ablauf der Zeit anschaulich angezeigt zu bekommen und diese mit den Studenten durchsprechen.

Zur Umsetzung dieses Projektes sollte eng mit einer Gruppe mit einem ähnlichen Projekt zusammengearbeitet werden, die jedoch eine etwas andere Art von Übungsblättern für den Einsatz von Aufgaben innerhalb oder auch außerhalb der Vorlesung erstellen sollte. Soweit möglich sollten dabei Synergien genutzt und eine gemeinsame Oberfläche für beide Projekte erstellt werden.

1.2 Zielsetzung

Diese Studienarbeit sollte nicht nur zur „Absolvierung“ der Studienarbeit dienen, sondern auch den Studenten einen praktischen Nutzen durch das Erlernen und Vertiefen von Wissen im Bereich moderner Technologien bieten. Zusätzlich war ein klares Ziel dieser Studienarbeit, dass das Endprodukt nicht nur irgend einen theoretischen Nutzen hat oder ein Proof of Concept sein sollte, sondern anschließend praktische Verwendung findet.

2 Anforderungsanalyse

2.1 Use-Cases

2.1.1 UC1: Frage mit Musterlösungen anlegen

UC Nr.: UC1	
Name	Frage mit Musterlösungen anlegen
Ziel im Kontext	Dozent legt eine neue Frage mit ihrer Musterlösung an
Akteure	Dozent (D.)
Trigger	D. klickt auf „Frage anlegen“
Essenzielle Schritte	<ol style="list-style-type: none"> 1. D. befindet sich im Admin Bereich 2. D. klickt auf „Frage anlegen“ 3. D. sieht Web Interface zum Frage erstellen 4. D. wählt den Fragentyp aus einer Auswahlmöglichkeit aus 5. D. wählt die Kategorie für die Frage aus einer Auswahlmöglichkeit aus 6. D. gibt Frage in das dazu passende Feld ein 7. D. gibt die zur Frage passende/n Musterlösung/en in das/die passende/n Feld/er ein 8. D. klickt abschließend auf „Abschicken“
Erweiterungen	4a. D. gibt an wie viele Antworten für Auflistaufgaben gefordert werden

Zugehöriger Use Case	UC1
Beteiligte Tabellen	Question, Type, Category, Solution
Vorbedingung	Alle benötigten Daten wurden eingegeben
Ablauf	<ol style="list-style-type: none"> 1. Es wurden schon beim Aufrufen des „Frage erstellen“ Formular, die Kategorien und Typen mit ihren IDs geladen 2. Beim abschicken des Formulars wird ein neuer Eintrag in die Tabelle „Question“ erstellt mit einer automatisch hochzählenden ID (qID), der Kategorie (cID) und Typen (tID) ID, dem jetzigen Datum mit Uhrzeit (creationDate und lastEditedDate), und der eigentlichen Frage (text). 3. Die dazu gehörigen Antworten werden in die „Solution“ Tabelle gespeichert mit einer automatisch hochzählenden unique ID (sID), der dazu gehörigen Fragen ID (qID), der Antwort selbst (text) und ob es sich um eine korrekte oder falsche Antwort handelt (correct) [wichtig für Multiple Choice]
Erweiterungen	

2.1.2 UC2: Kategorie anlegen

UC Nr.: UC2	
Name	Kategorie anlegen
Ziel im Kontext	Dozent legt eine neue Kategorie an
Akteure	Dozent (D.)
Trigger	D. klickt auf „Kategorie anlegen“
Essenzielle Schritte	<ol style="list-style-type: none"> 1. D. befindet sich im Admin Bereich 2. D. klickt auf „Kategorie anlegen“ 3. D. sieht Web Interface zum Kategorien erstellen 4. D. gibt den Namen der Kategorie in das dazu passende Feld ein 5. D. klickt abschließend auf den „Plus“-Button
Erweiterungen	

Zugehöriger Use Case	UC2
Beteiligte Tabellen	Category
Vorbedingung	Alle benötigten Daten wurden eingegeben
Ablauf	Beim Abschicken des Formulars wird in der Tabelle „Category“ ein neuer Eintrag erstellt mit automatisch hochzählender ID (cID) und dem Namen der Kategorie (cName)
Erweiterungen	

2.1.3 UC3: Vorlesung anlegen

UC Nr.: UC3	
Name	Vorlesung anlegen
Ziel im Kontext	Dozent legt eine neue Vorlesung an
Akteure	Dozent (D.)
Trigger	D. klickt auf „Vorlesung anlegen“
Essenzielle Schritte	<ol style="list-style-type: none"> 1. D. befindet sich im Admin Bereich 2. D. klickt auf „Vorlesung anlegen“ 3. D. sieht Web Interface zum Vorlesungen anlegen 4. D. gibt den Namen der Vorlesung in das dazu passende Feld ein 5. D. klickt abschließend auf den „Plus“-Button
Erweiterungen	

Zugehöriger Use Case	UC3
Beteiligte Tabellen	Lecture
Vorbedingung	Alle benötigten Daten wurden eingegeben
Ablauf	Beim Abschicken des Formulars wird in der Tabelle „Lecture“ ein neuer Eintrag erstellt mit automatisch hochzählender ID (cID) und dem Namen der Vorlesung (IName)
Erweiterungen	

2.1.4 UC4: Arbeitsblatt anlegen

UC Nr.: UC4	
Name	Arbeitsblatt anlegen
Ziel im Kontext	Dozent legt ein neues Arbeitsblatt an
Akteure	Dozent (D.)
Trigger	D. klickt auf „Arbeitsblatt anlegen“
Essenzielle Schritte	<ol style="list-style-type: none"> 1. D. befindet sich im Admin Bereich 2. D. klickt auf „Arbeitsblatt anlegen“ 3. D. sieht Web Interface zum Erstellen eines Arbeitsblattes 4. D. gibt den Namen des Arbeitsblattes in das dazu passende Feld ein 5. D. erhält Übersicht aller Fragen mit Checkbox zum Auswählen mit Filtermöglichkeit 6. D. sieht eine extra Tabelle mit den bisher ausgewählten Fragen 7. D. klickt nach Hinzufügen aller gewünschten Fragen auf „Arbeitsblatt anlegen“, wenn alle gewünschten Fragen hinzugefügt wurden
Erweiterungen	6a. D. kann die Reihenfolge der Fragen verändern oder sie wieder entfernen

Zugehöriger Use Case	UC4
Beteiligte Tabellen	ExcerciseSheet, SheetQuestions
Vorbedingung	Alle benötigten Daten wurden eingegeben
Ablauf	<ol style="list-style-type: none"> 1. Beim Abschicken des Formulars mit dem Button „Arbeitsblatt anlegen“ wird in der Tabelle „ExcerciseSheet“ ein neuer Eintrag erstellt mit automatisch hochzählender unique ID (esID), dem Namen des Arbeitsblattes (esName) und dem Zeitpunkt der Erstellung (esDateTime) 2. In der Tabelle SheetQuestions wird für jede, dem Arbeitsblatt hinzugefügte, Frage ein Eintrag mit der esID und der ID der hinzugefügten Frage (qlID) eingefügt.
Erweiterungen	Einstellen, wie lange Arbeitsblatt valid ist -> theoretisch auch über Button „aktiv machen“ und „sperren“ möglich.

2.1.5 UC5: Antworten abschicken

UC Nr.: UC5	
Name	Antworten abschicken
Ziel im Kontext	Studenten schicken Antworten ab
Akteure	Student (S)
Trigger	S. klickt auf „Antworten abschicken“
Essenzielle Schritte	<ol style="list-style-type: none">1. S. befindet sich im Studenten-Bereich2. S. klickt auf „Antworten abschicken“3. S. sieht Bestätigungsdialog4. S. klickt auf „OK“5. S. sieht Bestätigungsdialog
Erweiterungen	4a. S. klickt auf „Abbrechen“ und schickt somit seine Antworten noch nicht ab

Zugehöriger Use Case	UC5
Beteiligte Tabellen	Questions, StudentAnswer, Solution, StudentText
Vorbedingung	Alle benötigten Daten wurden eingegeben
Ablauf	<ol style="list-style-type: none"> 1. Bei jeder gegebenen Antwort wird überprüft, ob es sich um eine Multiple-Choice-Frage oder um eine Freitext-Frage handelte (anhand der cID aus der Questions-Tabelle) 2. Abhängig von Fragetyp <ol style="list-style-type: none"> a) Es handelt sich um eine Multiple-Choice-Antwort: <ol style="list-style-type: none"> i. Es wird in der Tabelle StudentAnswer ein Eintrag mit einer fortlaufenden ID (saID), der ID des Arbeitsblattes (esID), der VorlesungsID (IID), der FragenID (qID), der AntwortID (sID in der Spalte text), der SessionID seiner aktuellen Session und dem aktuellen Datum (dateTime) hinzugefügt b) Es handelt sich um eine Aufzähl-Antwort: <ol style="list-style-type: none"> i. Es wird in der Tabelle StudentAnswer ein Eintrag mit einer fortlaufenden ID (saID), der ID des Arbeitsblattes (esID), der VorlesungsID (IID), der FragenID (qID), dem Antworttext (text), der SessionID seiner aktuellen Session und dem aktuellen Datum (dateTime) hinzugefügt c) Es handelt sich um eine Freitext-Antwort <ol style="list-style-type: none"> i. In die Tabelle StudentText wird ein Eintrag mit einer fortlaufenden ID (stID), dem Text der Antwort, der ArbeitsblattID (esID), der VorlesungsID (IID), der FragenID (qID) und dem aktuellen Datum (dateTime) hinzugefügt.
Erweiterungen	

2.1.6 UC6: Auswertung des Übungsblattes

UC Nr.: UC6	
Name	Auswertung des Arbeitsblattes
Ziel im Kontext	Dozent wertet Arbeitsblatt aus
Akteure	Dozent (D)
Trigger	D. klickt auf „Auswertung“
Essenzielle Schritte	<ol style="list-style-type: none"> 1. D. befindet sich im Admin-Bereich 2. D. klickt auf „Auswertung“ 3. D. sieht Übersicht der verschiedenen Vorlesungen und Arbeitsblättern
Erweiterungen	

Zugehöriger Use Case	UC6
Beteiligte Tabellen	SheetQuestions, Question, StudentAnswer, StudentText
Vorbedingung	
Ablauf	<ol style="list-style-type: none"> 1. Zu jeder der Vorlesung und dem ausgewählten Arbeitsblatt werden die zugehörigen Fragen abgerufen. 2. Abhängig von Fragetyp <ol style="list-style-type: none"> a) Es handelt sich um eine Multiple-Choice-Antwort: <ol style="list-style-type: none"> i. Aus der Tabelle „studentAnswer“ werden nacheinander alle zur in der Vorlesung (IID) und dem Arbeitsblatt zugehörigen (esID) Frage (qID) die Antworten abgerufen und prozentual dargestellt, wie viele korrekt, falsch und nicht beantwortet wurden. b) Es handelt sich um eine Freitext-Antwort <ol style="list-style-type: none"> i. Aus der Tabelle studentText werden zehn zufällige Antworten ausgewählt und nacheinander präsentiert.
Erweiterungen	Anzahl der ausgegebenen Freitext-Antworten variabel machen

2.2 Allgemeine Anforderungen

Die an das Projekt gestellten Anforderungen gliedern sich in zwei Gebiete: Technologische Anforderungen und allgemeine Anforderungen, die an die Applikation gestellt werden. Soweit möglich sollte bei der Datenhaltung Synergie-Effekte mit dem Projekt „CCKE“ benutzt werden. Ebenso sollte der Bereich für Lehrbeauftragte für die Erstellung und Verwaltung und Präsentation der Ergebnisse weit möglichst für beide Projekte einheitlich gehalten werden.

2.3 Technologische Anforderungen

Die zu entwickelnde Applikation sollte auf Basis von Webtechnologien aufgebaut werden. Desweiteren sollte das Backend für Lehrbeauftragte sowie die Auswertung der Ergebnisse an einem Notebook bedient werden können. Die Oberfläche für die Auswertung der Antworten sollte dabei so optimiert sein, dass die Ergebnisse ohne größeren Aufwand mittels einem im Raum vorhandenen Beamer präsentiert werden können.

Die Oberfläche für die Eingabe der Antworten der Studenten sollte dabei für mobile Geräte optimiert werden, was insbesondere Smartphones und Tablets beinhalten sollte. Die Optimierung für mobile Geräte sollte die Bedienbarkeit für verbreitete mobile Plattformen sicherstellen.

Die gesamte Datenhaltung der Applikation sollte mit Hilfe einer frei verfügbaren Datenbank realisiert werden.

2.4 Eingesetzte Technologien

Bei der Auswahl der Technologien wurde insbesondere Wert darauf gelegt, dass die Grundlagen für diese den Studenten bereits bekannt sind. Allerdings sollten auch neue Aspekte unter Zuhilfenahme moderner Ansätze und Frameworks erlernt und

so die Grundlage für eine spätere Erweiterung des Projekts durch nachfolgende Gruppen sichergestellt werden.

Nachfolgend eine Auflistung der eingesetzten Technologien:

- HTML
- CSS
- JavaScript
- AJAX
- PHP
- SQL

3 Vorüberlegungen

Zu Beginn der Studienarbeit mussten einige Vorüberlegungen getätigt werden:

- Es musste ein Datenmodell erstellt werden, welches die Struktur der Datenbank beschreibt.
- Es mussten alle zu diesem Zeitpunkt bereits bekannten Aufgaben festgehalten werden, um sich einen Überblick über den Funktionsumfang und Arbeitsaufwand zu verschaffen.

Aus den so erhaltenen Aufgaben und den Funktionalitäten, die das Projekt am Ende bieten sollte, sollte ein Konzept zum Design einer passenden Oberfläche erstellt werden.

3.1 Datenmodell

Das Datenmodell wurde früh entwickelt und sollte als Grundlage dienen, um die Datenstruktur der Datenbank implementieren. Dabei ist es wichtig Zeit in das Datenmodell zu investieren, da das Datenmodell als fundamentale Grundlage dient und große Änderungen daran später zeitaufwändige Folgen haben würden. Dabei wurde eine relationale Datenbank gewählt und versucht, das Datenmodell nach den Prinzipien der Theorie der relationalen Datenbanken aufzubauen. Dementsprechend wurde die Tabellenstruktur in dritter Normalform aufgebaut.

Im Verlauf der Studienarbeit mussten dennoch Änderungen am Datenmodell vorgenommen werden, jedoch waren diese nur kleine Änderungen, wie z.B. das Hinzufügen einer Spalte innerhalb einer Tabelle. Dies lässt den Schluss zu, dass die zu

Beginn investierte Zeit in das Datenmodell und die Gespräche und Diskussionen zu Beginn hierzu sinnvoll investierte Zeit waren.

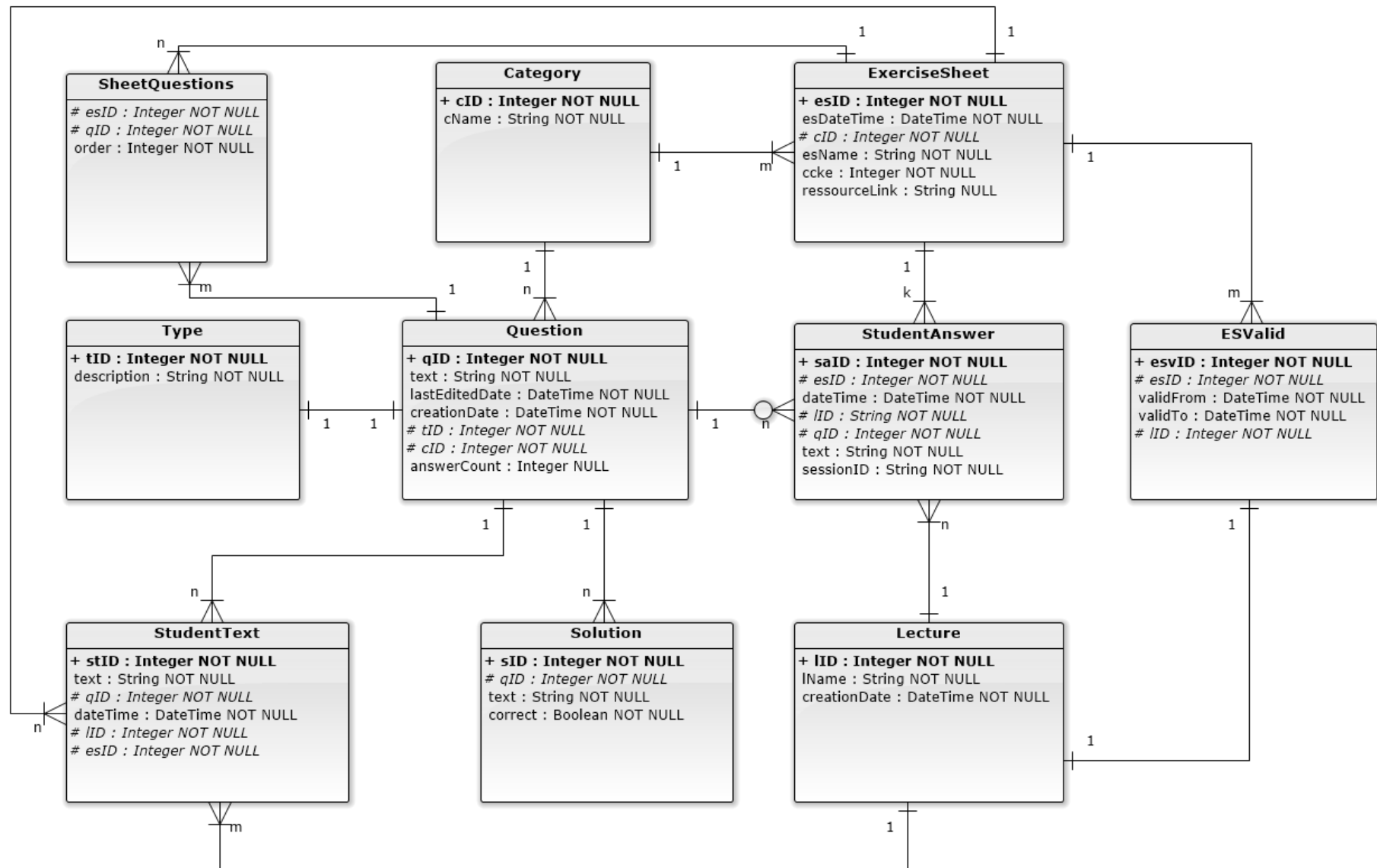


Abbildung 3.1: Datenmodell LeMon + CCKE

3.2 Funktionsumfang

Der letztendliche Funktionsumfang war zu Beginn des Projektes zwar grob klar, wie in Projekten in Unternehmen war es jedoch auch so, dass sich der Kunde gewisse Dinge später anders vorgestellt hat oder gar neue Wünsche für Funktionen äußerte. Darauf muss man dynamisch agieren und neue Dinge während der Projektphase hinzufügen oder bestehende gegebenenfalls zu überarbeiten. Für die wichtigsten Funktionalitäten des Projektes, wurden Use-Cases aufgestellt.

3.3 Benutzeroberfläche

Anhand der gesammelten Anforderungen hinsichtlich Funktionalitäten, wurden einige Oberflächen schon weitestgehend definiert. Als Beispiel sei hier die Funktion 'Frage erstellen' genannt, bei der gewisse Eingabefelder vorgegeben waren um den Fragentyp und -text festzulegen. Die letztendlichen Anordnungen wurden dann in direkten Gesprächen mit dem Kunden geklärt.

4 Implementierung

4.1 Struktur des Projekts

Die Struktur des Projekts auf Datei-Ebene im Ordner *Management* ist folgender Maßen aufgebaut:

- Im Unterordner *Admin* liegen alle Dateien für den Admin-Bereich
- Im Unterordner *Frameworks* liegen alle Frameworks die im Projekt verwendet werden
- Im Unterordner *Images* liegen alle Bilder und Icons die im Projekt verwendet werden
- Im Unterordner *User* liegen alle Dateien des User-Bereiches
- Im Ordner *Management* selbst liegt die *generalConfig.php* Datei, die globale Variablen anlegt welche Pfade oder Informationen über die Datenbank definiert

4.2 Admin-Bereich

Der Admin-Bereich ist die Oberfläche über die der Dozent die Möglichkeit hat neue Fragen, Kategorien, Vorlesungen und Arbeitsblätter zu erstellen.

4.2.1 Struktur der Implementierung

Die Struktur des Admin Bereichs auf Datei-Ebene im Ordner *Admin* ist folgender Maßen aufgebaut:

- Im Unterordner *adminStyles* liegen alle CSS Dateien für den Admin-Bereich
- Im Unterordner *js* liegen alle JavaScript Dateien und Bibliotheken für den Admin-Bereich
- Im Ordner *Admin* selbst liegen alle PHP Dateien der einzelnen Unterseiten des Admin-Bereichs

Die Struktur des Admin Bereichs auf Code-Ebene kann unter dem Begriff der „prozeduralen Programmierung“ zusammengefasst werden. Wo dies sinnvoll war, wurden dabei Funktionen geschrieben die bestimmte Aufgaben abdecken. Durch eine starke Verzahnung von PHP und Hypertext Markup Language (HTML) Code und um die Übersichtlichkeit zu gewährleisten, wurde der HTML Code größtenteils mit in den PHP Code integriert.

4.2.2 Eingesetzte Libraries und Frameworks

Es werden mehrere Frameworks verwendet um die Funktionalität des Admin-Bereichs zu erhalten.

- Mit dem Framework *adoDB*¹ können Verbindungen mit Datenbank erstellt und gemanaged. Außerdem stellt es Funktionen für Standard Query Language (SQL) Querys zur Verfügung.
- Die bekannte Bibliothek *jQuery*² wird im Admin-Bereich zur Sicherstellung der Crossbrowser-Funktionalität eingesetzt.
- Die Bibliothek *PHPQRCode*³ wird dazu verwendet um QR-Codes durch PHP zu generieren.
- Die Bibliothek *pChart*⁴ wird zum Erstellen von Auswertungsdiagrammen verwendet.

4.2.3 Besonderheiten der Implementierung

Im Folgenden sollen einige Besonderheiten der Implementierung dargestellt und erläutert werden.

¹ AdoDB-Projektseite auf Sourceforge: <http://adodb.sourceforge.net>

² JQuery-Projektseite: <http://jquery.com>

³ PHPQRCode-Projektseite auf Sourceforge: <http://phpqrcode.sourceforge.net>

⁴ pChart-Projektseite auf Sourceforge: <http://pchart.sourceforge.net>

- Damit die Webseite nicht bei jedem Seitenwechsel in der Navigationsleiste neu geladen werden muss, wurde zur Navigation Ajax verwendet. Dadurch wird immer nur der Seitenbereich mit neuem Content aktualisiert, der Rest der Seite muss deswegen nicht unnötig nach geladen werden.
- Der QR-Code wird mit der PHPQRCode Bibliothek generiert. Dabei ist jedoch eine Besonderheit zu beachten: Da der QR-Code nicht unnötig zwischengespeichert werden sollte, musste ein besonderes Konstrukt verwendet werden. Dabei wird durch eine separate PHP Datei der QR-Code erzeugt und in dieser Datei gleichzeitig der Content-Type auf *image/png* gesetzt. Anschließend wird diese PHP Datei mit dem generierten Bild in den Image-Tag einer anderen Seite eingebettet.
- Zur Auswertung von Aufzählaufgaben wird versucht eine möglichst genaue Korrektheitsüberprüfung durchzuführen. Dazu werden die gegebenen Antworten der Studenten in Kleinbuchstaben konvertiert, Umlaute durch (ae, ue, oe, ss) ersetzt und bestimmte Sonderzeichen entfernt. Danach werden diese Antworten mit der nach den selben Methoden konvertierten Musterlösung verglichen.
- Nach Abschluss des Auswertungsvorganges werden die zu einer Frage gehörenden Studentenantworten in einem Diagramm, welches mit der Bibliothek pChart erstellt wird, dargestellt. In diesem Diagramm sind die Antworten grafisch dargestellt.

4.3 LeMon-Bereich

Der LeMon-Bereich begrenzt sich auf das Ausfüllen der für LeMon angelegte Arbeitsblätter.

4.3.1 Struktur der Implementierung

Um die Implementierung einfach und übersichtlich zu halten, wurde nach dem Stil dem Konzept der objektorientierten Programmierung vorgegangen.

Folgendermaßen wurde dabei die Ordnerstruktur aufgebaut:

- Im Unterordner *css* liegen alle CSS Dateien für den LeMon-Bereich
- Im Unterordner *image* liegen alle Bilder und Icons, die im LeMon-Bereich verwendet werden

- Im Unterordner *include* liegt zum einen eine PHP Datei, welche die Datenbankverbindung sicherstellt und zum anderen eine PHP Datei die zentral alle verwendeten Klassen aus dem extra Unterordner *Classes* einbindet.
- Im Unterordner *js* liegen alle JavaScript Dateien und Bibliotheken für den LeMon-Bereich
- Im Ordner *LeMon* selbst liegt die Arbeitsblatt PHP Datei, welche das Layout des Arbeitsblattes festlegt.

4.3.2 Eingesetzte Libraries und Frameworks

Es werden mehrere Frameworks verwendet um die Funktionalität des LeMon-Bereichs zu erhalten.

- Mit dem Framework Bootstrap¹, welches „*das bekannteste Framework zum Entwickeln von responsive mobile first Web-Projekten ist*“ [1], wird der LeMon-Bereich auf gängigen Mobilgeräten angepasst dargestellt.
- Die bekannte Bibliothek jQuery² wird im Admin-Bereich zur Sicherstellung der Crossbrowser-Funktionalität eingesetzt.

¹ Bootstrap-Projektseite: <http://getbootstrap.com>

² JQuery-Projektseite: <http://jquery.com>

5 Fazit und Ausblick

6 Anhang

This is makeindex, version 2.15 [TeX Live 2014/dev] (kpathsea + Thai support). Scanning style file ./StudienarbeitLeMon.ist.....done (29 attributes redefined, 0 ignored). Scanning input file StudienarbeitLeMon.gls...done (0 entries accepted, 0 rejected). Scanning input file StudienarbeitLeMon.glo....done (29 entries accepted, 0 rejected). Overall 2 files read (29 entries accepted, 0 rejected). Sorting entries....done (139 comparisons). Generating output file StudienarbeitLeMon.ind....done (39 lines written, 0 warnings). Output written in StudienarbeitLeMon.ind. Transcript written in StudienarbeitLeMon.gls.

Abbildungsverzeichnis

3.1	Datenmodell LeMon + CCKE	23
-----	------------------------------------	----

Tabellenverzeichnis

Quellcodeverzeichnis

Literaturverzeichnis

- [1] MARC OTTO & JACOB THORNTON: *Bootstrap Website*
<http://getbootstrap.com>. März 2014