

Studienarbeit



*Entwicklung eines E-Learning-Systems zur Integration
in das Portal der Berufsakademie Mannheim*

David Koch
Michael Schlembach

Kurs: TIT01APE
Betreuer: Prof. Poller

Ehrenwörtliche Erklärung

Hiermit versichern wir ehrenwörtlich diese Studienarbeit und die dazu gehörige Entwicklung selbständig und nur mit den angegebenen Hilfsmitteln angefertigt zu haben.

Mannheim, den 01.06.2004

Ort, Datum

David Koch

Mannheim, den 01.06.2004

Ort, Datum

Michael Schlembach

Abstract

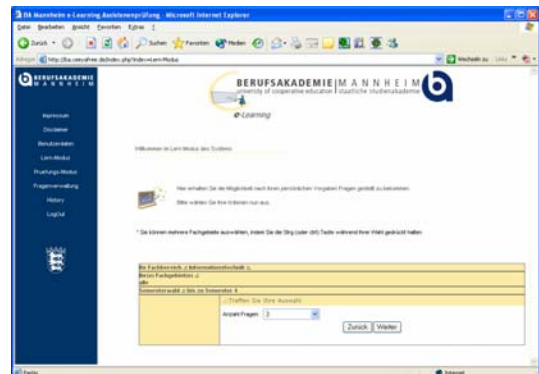


Abbildung 1: Screenshot des e-Learning Systems

Die Studienarbeit im 5. und 6. Semesters der Vertiefungsrichtung Projekt Engineering beinhaltet die Erstellung eines Konzeptes für ein e-Learning System und deren Implementierung.

Das entwickelte e-Learning System soll im Rahmen der Neuentwicklung des Internetportals der Berufsakademie Mannheim in den internen Bereich integriert werden.

Die Umsetzung erfolgte auf Basis einer MySQL Datenbank und PHP.

Das Konzept des Tools sieht eine Überprüfung der Fragen durch den Studenten vor.

Eine besondere Fokussierung bei der Umsetzung ist die so genannte „History-Funktion“. Sie ermöglicht es dem Studenten eine Übersicht über seine durchgeführten Tests zu erhalten und kann dadurch seine Fortschritte genau verfolgen.

Inhaltsverzeichnis

Ehrenwörtliche Erklärung	2
Abstract	3
Inhaltsverzeichnis	4
Einleitung	6
Aufgabenstellung und Annahmen	7
Genutzte Systeme	8
Die MySQL Datenbank	9
Die Skriptsprache PHP	10
Die Funktionalität von PHP	11
Zugriff mit PHP auf MySQL Datenbank	13
Schema des Datenbankzugriffs	13
Der erste Kontakt	13
Auswahl der Datenbank	14
Anfrage an MySQL	14
Schreiben mit SQL	15
Schließen der Verbindung	16
Funktionalitäten des e-Learning Tool	17
Verwaltungsfunktionalität	17
Benutzerverwaltung	17
Login-Funktionalität des Systems	17
Benutzergruppen	18
Fragen / Antwortverwaltung	20
Ergebnisverwaltung	20
Durchführung der Lern-Modi	21
Lern Modus	21
Prüfungsmodus	21
Umsetzung	22

Datenbank Schemata	22
Datenbankschema ohne Relationen	25
Datenbankschema mit Relationen	25
Datenbankverbindung und Abfragen	26
History Funktion	28
Fragenkatalog für Lern-Modi	30
Anzeigen der Fachbereiche und Fachgebiete	30
Randomizer bei Fragenauswahl	30
Frage darstellen	30
Überprüfung der Frage	30
Anhang	30
Abbildungsverzeichnis	30
Quellen	30
Unterstützende Professoren	30

Einleitung

Die Studienarbeit im 5. und 6. Semesters der Vertiefungsrichtung Projekt Engineering beinhaltet die Erstellung eines Konzeptes für ein e-Learning System und deren Implementierung.

Das entwickelte e-Learning System soll im Rahmen der Neuentwicklung des Internetportals der Berufsakademie Mannheim in den internen Bereich integriert werden.

In dieser Studienarbeit gehen wir im Einzelnen auf die genutzten Technologien, die Funktionalität des e-Learning Systems und exemplarisch auf die Implementierung und Datenbank Schemata ein.

Das Konzept des Tools sieht eine Überprüfung der Fragen durch den Studenten vor, wodurch eine möglichst hohe Flexibilität betreffend der Fragestellung gewährleistet ist.

Eine besondere Fokussierung bei der Umsetzung ist die so genannte „History-Funktion“. Sie ermöglicht es dem Studenten eine Übersicht über seine durchgeführten Tests zu erhalten und kann dadurch seine Fortschritte genau verfolgen und sein weiteres Lernen optimaler ausrichten.

Ein weiterer Fokus ist der so genannte „Prüfungs-Modus“. Dieser Modus erlaubt es dem Studenten eine Assistentenprüfung zu simulieren.

Aufgabenstellung und Annahmen

Die Aufgabenstellung des Themas für die Studienarbeit war die Erstellung eines Konzeptes für ein e-Learning System und dessen Entwicklung.

Das entwickelte e-Learning System soll im Rahmen der Neuentwicklung des Internetportals der Berufsakademie Mannheim in den internen Bereich integriert werden.

Für unsere Konzeptionierung und Entwicklung gehen wir von dem Vorhandensein der benötigten Ressourcen aus. Sie wurden uns entsprechend im 5. Semester zugesagt.

Aufgrund der Länge mancher Codezeilen haben wir uns das Recht vorbehalten Codezeilen über mehrere Zeilen zu verteilen.

Genutzte Systeme

Die Umsetzung des e-Learning Systems beruht auf einer MySQL Datenbank und einer Internetseite, auf einem Apache Server. Für den Datenaustausch und die Kommunikation zwischen Internetseite und der Datenbank kommt PHP zum Einsatz.

Die Installation, Konfiguration und Integration dieser Komponenten ist eine komplexe Aufgabe. Damit der Aufwand für diese Arbeit in einem annehmbaren Rahmen bleibt, wurde die kostenlose Distribution XAMPP eingesetzt.

XAMPP ist eine Distribution mit bereits vorkonfiguriertem Apache, MySQL, PHP und Perl.



Abbildung 2: Logo der XAMPP Distribution

XAMPP 1.4.2 beinhaltet die für das e-Learning System wichtigen Komponenten:

- Apache 2.0.48
- MySQL 4.0.17
- PHP 4.3.4
- phpMyAdmin 2.5.5 pl1

Die MySQL Datenbank

MySQL ist eine echte Multi-User, Multi-Treaded SQL Datenbank und wird von allen großen Providern oder auch Suchmaschinenbetreibern eingesetzt. MySQL ist eine Client/Server Implementierung, die aus einem Server-Dämon `mysqld` und vielen Client Programmen, sowie Bibliotheken für PERL, PHP/3, PHP/4 sowie ASP besteht.

SQL ist eine standardisierte Datenbanksprache, die das Speichern, Updaten und den Zugriff auf Informationen erleichtert. Beispielsweise kann man Produktinformationen eines Kunden auf einem WWW-Server speichern und abrufen. MySQL ist äußerst schnell und flexibel genug, um sogar Bilder und Log-Dateien darin abzulegen. In der Praxis ist MySQL sehr viel schneller, als z.B. ORACLE oder INFORMIX.

Die wichtigsten Eigenschaften von MySQL sind Geschwindigkeit, Stabilität und einfache Bedienbarkeit.

MySQL wurde ursprünglich entwickelt, weil auf TCX (dem Server der Entwickler) ein SQL Server benötigt wurde, der sehr große Datenbanken handeln konnte, und zwar um eine Größenordnung schneller, als die Datenbankhersteller damals liefern konnten. MySQL ist nun seit 1996 produktiv im Einsatz und erfreut sich seitdem wachsender Beliebtheit.

Die Skriptsprache PHP

PHP ist die Abkürzung für "PHP: Hypertext Preprocessor", eine weit verbreitete Open Source Skriptsprache speziell für Webentwicklungen. Das besondere Merkmal von PHP ist die einfache Einbindung in HTML. Seine Syntax erinnert an C, Java und Perl und es ist einfach zu erlernen. Das Hauptziel dieser Sprache ist es, Webentwicklern die Möglichkeit zu geben, schnell dynamisch generierte Webseiten zu erzeugen.

Folgend ein kurzes Beispiel für die Integration von PHP in den HTML Code:

```
<html>
  <head>
    <title>Beispiel</title>
  </head>
  <body>
    <?php
      echo "Hallo, ich bin ein PHP-Skript!";
    ?>
  </body>
</html>
```

Abbildung 3: HTML-Code als Beispiel für eine Einbindung von PHP

Am obigen Beispiel ist leicht zu erkennen, dass der PHP-Code zwischen speziellen Anfangs- und Schlusstags steht, mit ihnen ist es möglich auf leichte Art zwischen HTML- und dem PHP-Code zu wechseln, PHP besitzt also einen großen Vorteil gegenüber anderen Sprachen wie Perl oder C: Es ist kein Programm mit vielen Anweisungen zur Ausgabe von HTML nötig.

Was PHP von clientseitigen Sprachen wie Javaskript unterscheidet, ist dass der Code auf dem Server ausgeführt wird. Sollte das Skript wie das obige auf einem Server ausgeführt werden, würde der Besucher nur das Ergebnis empfangen, ohne die Möglichkeit herauszufinden, wie der zugrunde liegende Code aussieht.

Die Funktionalität von PHP

PHP ist hauptsächlich auf serverseitige Skripte fokussiert, es können also z.B. Formulardaten gesammelt, dynamische Inhalte für Websites generiert oder Cookies gesendet und empfangen werden.

Es gibt drei Hauptgebiete, in denen PHP Skripte genutzt werden.

- Serverseitige Skripte. Dies ist die Hauptfunktionalität von PHP. Es werden dafür drei Dinge benötigt: Der PHP Parser (CGI oder Server-Modul), ein Webserver und ein Webbrowser. Des Weiteren muss der Webserver mit der PHP Distribution verbunden sein.
- Skripte auf der Kommandozeile. Um ein PHP Skript ohne Server zu schreiben, benötigt man den PHP Parser. Diese Art der Verwendung ist für regelmäßig auszuführende Skripte mittels `cron` (unter Linux) oder dem `Task Scheduler` (unter Windows) ideal geeignet. Diese Skripte können auch für einfache Aufgaben, so zum Beispiel zur Verarbeitung von Text verwendet werden.
- Schreiben clientseitiger GUI Applikationen. PHP ist wahrscheinlich nicht die optimale Sprache um GUI Applikationen zu schreiben, aber dennoch ist dies durchaus mit PPH-GTK möglich, wobei eben erwähntes PlugIn nicht in der Hauptdistribution enthalten ist.

PHP kann auf allen gängigen Betriebssystemen verwendet werden, inkl. Linux, vielen Unix-Varianten (inkl. HP-UX, Solaris und OpenBSD), Microsoft Windows, Mac OS X, RISC OS, und anderen. PHP unterstützt auch die meisten der heute gebräuchlichen Webserver. Dies umfasst Apache, Microsoft Internet Information Server, Personal Web Server, Netscape und iPlanet Server, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd, und viele andere. Für den Großteil der Server bietet PHP ein eigenes Modul, für die anderen, welche den CGI Standard unterstützen, kann PHP als CGI Prozess arbeiten.

Der große Vorteil von PHP wurde im vorangegangenen Text mit der Möglichkeit der Integration in den HTML-Code beschrieben. Dies bedeutet jedoch nicht, dass dies eine Beschränkung von PHP ist.

Es ist über dies möglich dynamisch Bilder, PDF Dateien und Flash Animationen (mittels `libswf` und `Ming`) zu generieren.

Ein anderer großer Vorteil von PHP ist die Unterstützung einer großen Anzahl von Datenbanken. Die folgenden Datenbanken werden zurzeit von PHP unterstützt:

Adabas D	Ingres	Oracle (OCI7 und OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (nur Lesezugriff)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	Unix dbm	

Zusätzlich zu dieser Anzahl von Datenbanken unterstützt PHP ODBC, den Open Database Connection Standard, mit welchem es sich zu jeder anderen, diesen Weltstandard unterstützenden, Datenbank verbinden kann.

PHP unterstützt auch die Kommunikation mit anderen Services und Protokollen, z.B. LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (auf Windows)

PHP verfügt über äußerst hilfreiche Textverarbeitungsfunktionen, von den regulären Ausdrücken (POSIX erweitert oder Perl) bis zum Parsen von XML Dokumenten. Für den Zugriff und das Parsen von XML Dokumenten unterstützt es die Standards SAX und DOM.

Zugriff mit PHP auf MySQL Datenbank

Im folgenden Abschnitt wird auf die die Verwendung von PHP für Datenbankzugriffe eingegangen.

Schema des Datenbankzugriffs

Das Schema eines Datenbankzugriffes folgt generell den gleichen Schritten, dies gilt auch für MySQL mit PHP:

1. Verbindung zum MySQL-Server herstellen
2. Datenbank auswählen
3. SQL-Query abschicken
4. Ergebnis der Query verarbeiten
5. Verbindung beenden

Der erste Kontakt

Bevor Abfragen an den MySQL-Server gestellt werden, wird zuerst eine Verbindung aufgebaut. Bevor diese Verbindung aufgebaut werden kann, sind folgende Daten notwendig: der Name und Port des Rechners, auf dem der Datenbankserver läuft, eine Benutzerkennung und das dazugehörige Passwort. War die Verbindung erfolgreich, erhält man von PHP einen so genannten „*link identifier*“, ähnlich dem Filehandle, den man bei Operationen auf der Datenbank mit übergeben muss.

```
<?  
$link = mysql_pconnect("host:port","username","password");  
?>
```

Auswahl der Datenbank

Der nächste Schritt ist die Auswahl einer Datenbank, da ein MySQL-Server natürlich viele Datenbanken gleichzeitig beherbergen kann. Nach der Wahl der Datenbank erfolgen alle folgenden Queries auf diese Datenbank, und zwar solange, bis eine neue Datenbank ausgewählt wird. Diese Queries sind an das Handle geknüpft.

Durch dieses Prinzip können andere User auf andere Datenbanken des MySQL Servers zugreifen.

```
<?  
mysql_select_db("database", $link);  
?>
```

Anfrage an MySQL

Ist eine Verbindung hergestellt und eine Datenbank ausgewählt, kann eine Datenbankabfrage gesendet werden. Dazu formuliert man zuerst eine SQL-Abfrage und schickt diese dann ab. Als Ergebnis erhält man einen so genannten „*result identifier*“ zurück, über den dann im folgenden die Werte des Ergebnisses abgefragt werden können:

```
<?  
$query = "select name, vorname from tabelle where id > 10";  
$result = mysql_query($query, $link);  
?>
```

Konnte die Abfrage nicht ausgeführt werden, enthält \$result einen negativen Wert.

War die Abfrage erfolgreich, können nun über den „*result identifier*“ die zurück gelieferten Werte ausgewertet werden. SELECT-Abfragen liefern Tabellen zurück, die nun in einer Schleife Zeile für Zeile abgerufen werden können.

Dazu sollte man zuerst feststellen, wie viele Zeilen das Ergebnis enthält und dann in einer Schleife jede Zeile einzeln abrufen. Am einfachsten ist es jedoch, die komplette Zeile als Array abzufragen. Der Index der Array-Felder entspricht den Spaltennamen in der SELECT-Abfrage:

```
<?
$num = mysql_numrows($result);
for ($i = 0; $i < $num; $i++) {
    $row = mysql_fetch_array($result);
    echo "Name : $row[name], Vorname: $row[vorname]<br>";
}
?>
```

Schreiben mit SQL

INSERT fügt neue Zeilen in eine bestehende Tabelle ein. Die INSERT ... VALUES-Form des Statements fügt Zeilen basierend explizit angegebene Werte ein. Die INSERT ... SELECT-Form fügt Zeilen ein, die aus einer oder mehreren anderen Tabellen ausgewählt wurden. Die INSERT ... VALUES-Form mit mehrfachen Wertelisten wird ab MySQL-Version 3.22.5 unterstützt. Die `spalten_name = expression`-Syntax wird ab MySQL-Version 3.22.10 unterstützt.

Wenn keine Spaltenliste für INSERT ... VALUES oder INSERT ... SELECT angegeben ist, müssen für alle Spalten Werte in der VALUES()-Liste oder vom SELECT bereit stehen.

Jede Spalte, die nicht explizit in einer Werteliste angegeben wird, wird auf ihren Vorgabewert gesetzt. Wenn beispielsweise eine Spaltenliste angegeben ist, die nicht alle Tabellenspalten nennt, werden unbenannte Spalten auf ihre Vorgabewerte gesetzt.

UPDATE aktualisiert Spalten in bestehenden Tabellenzeilen mit neuen Werten. Die SET-Klausel gibt an, welche Spalten geändert werden sollen und welche Werte ihnen zugewiesen werden. Die WHERE-Klausel legt - falls angegeben - fest, welche Zeilen aktualisiert werden sollen. Ansonsten werden alle Zeilen aktualisiert. Wenn die ORDER BY-Klausel angegeben ist, werden die Zeilen in der angegebenen Reihenfolge aktualisiert.

Schließen der Verbindung

Schließlich wird die Verbindung zum Datenbankserver geschlossen. Nur bei sehr vielen Zugriffen pro Minuten könnte es sich lohnen, die Verbindung offen zu lassen. Wenn Sie einen Datenbankserver über das Netzwerk nutzen, sollte man sich darüber im Klaren sein, dass MySQL auf ca. 100 gleichzeitige, offene Verbindungen begrenzt ist. Wenn mehr simultane Verbindungen gewünscht sind, muss MySQL dementsprechend neu kompiliert werden. Der hierfür notwendige Quellcode ist außerdem leider nur unter LINUX bzw. UNIX verfügbar.

Funktionalitäten des e-Learning Tool

Im Folgenden werden die Funktionalitäten des entwickelten Tools beschrieben.

Verwaltungsfunktionalität

Benutzerverwaltung

Die Benutzerverwaltung beruht auf einer Datenbank, welche zusätzlich zu den Standardmerkmalen des Benutzers, wie Name und Login Merkmale, wie Typ (also Administrator, Dozent, Student) und die Freigabe des Benutzers speichert. Die Freigabe oder Löschung eines Benutzers kann nur durch einen Administrator erfolgen.

Login-Funktionalität des Systems

Vor der Benutzung des e-Learning Systems muss sich jeder Benutzer in das System einloggen. Im Vorfeld zum ersten Login muss folglich eine Registrierung und ggf. eine Freischaltung erfolgen (siehe Abschnitt „Benutzergruppen“). Erst nach dem erfolgreichen Login erfolgt der Zugang zum System.

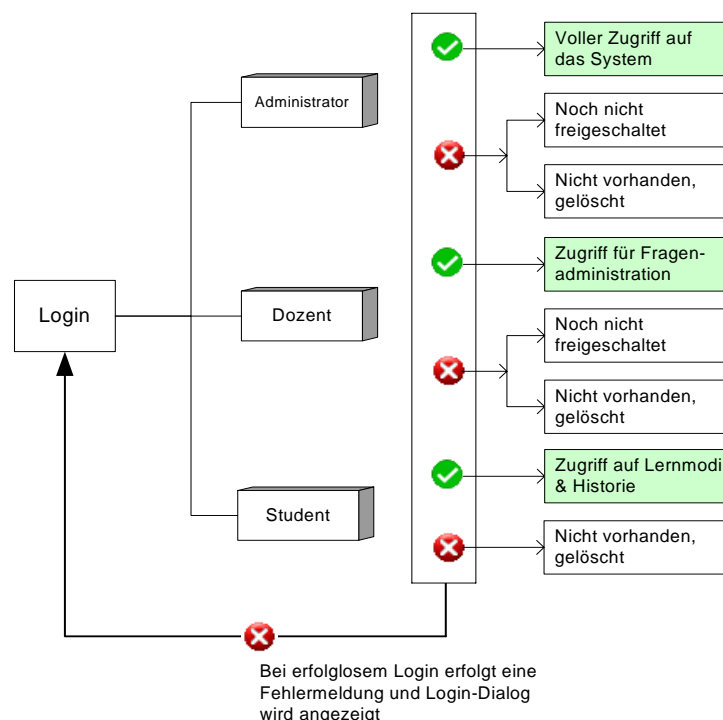


Abbildung 4: Ablaufdiagramm des Logins

Die Nutzung des Systems erfolgt nur durch eine Registrierung und einen Login, um zu gewährleisten, dass nur befugte Personen Änderungen vornehmen können und um die Möglichkeit für eine Historie-Funktion zu gewährleisten.

Benutzergruppen

Das Konzept dieses e-Learning Systems sieht drei unterschiedliche Benutzergruppen vor. Folgend werden diese kurz erklärt. Eine vollständige Übersicht folgt am Ende dieses Abschnitts.

1. Student

Diese Benutzergruppe kann das System zum Lernen nutzen und hat die Möglichkeit die persönlichen Daten zu ändern.

2. Dozent / Professor

- Diese Benutzergruppe kann Fragen mit Antworten einpflegen, bearbeiten und löschen aber auch
- Fachgebiete hinzufügen.

Zur Wahrung der Integrität des Systems muss jeder Benutzer dieser Gruppe vor dem ersten Zugriff auf das System von einem Administrator freigeschaltet werden. Zuzüglich zu dieser Funktionalität beinhaltet die Benutzergruppe Dozent die Rechte der Benutzergruppe Student

3. Administrator

Diese Benutzergruppe verfügt über die gleichen Rechte, wie die Benutzergruppe Dozent, zusätzlich hat diese jedoch die Möglichkeit

- Benutzer einzupflegen (Dozenten und Studenten werden automatisch freigeschaltet) und zu löschen,
- Fachgebiete und Fachbereiche zu erstellen und zu löschen,
- Dozenten freizuschalten.



Abbildung 5: Benutzerverwaltung für Administratoren

Übersicht der Rechte aller Benutzergruppen

Funktionalität	Zugriffsrechte für		
	Administrator	Dozent	Student
Lern Modi nutzen	✓	✓	✓
Frage einpflegen	✓	✓	✗
Frage entfernen	✓	✗	✗
Fachbereich hinzufügen	✓	✗	✗
Fachbereich entfernen	✓	✗	✗
Fachgebiet hinzufügen	✓	✓	✗
Fachgebiet entfernen	✓	✗	✗
Benutzer freischalten	✓	✗	✗
Benutzer hinzufügen	✓	✗	✗
Benutzer entfernen	✓	✗	✗

Fragen / Antwortverwaltung

Wie unter dem Abschnitt „Benutzergruppen“ bereits erwähnt, haben nur die Benutzergruppen „Dozent“ und „Administrator“ die Möglichkeit Fragen bzw. Antworten in das System einzupflegen.

Das Einpflegen erfolgt nach der Wahl von Fachbereich, Fachgebiet und Semester. Darauf folgend hat der Dozent die Möglichkeit zwischen folgenden „Frage-Typen“ zu wählen:

1. Frage und Antwort in Textform
2. Frage in Textform, Antwort in Textform, erweitert mit Bild (Beispiel: Physik)
3. Frage in Textform, Antwort besteht aus einer vorgegebenen Rangfolge von Stichwörtern (Beispiel: OSI-Modell)
4. Frage / Antwort in Multiple-Choice Form

Nach der Wahl des Fragetyps fügt der Dozent seine Frage und die Antwort in das System ein.

Die Benutzergruppen „Dozent“ und „Administrator“ verfügen zusätzlich über die Funktion sich die Vorhandenen Fragen anzuschauen und ggf. zu löschen.

Ergebnisverwaltung

In der Ergebnisverwaltung werden die Ergebnisse der Studenten gespeichert, es wird den Studenten dadurch ermöglicht ihren Lernfortschritt genau dokumentiert zu betrachten.

Durchführung der Lern-Modi

Der Student hat die Möglichkeit zwischen zwei Lern-Modi zu wählen:

1. manueller Lern Modi, kurz Lern-Modus
2. Prüfungssimulation

Prinzipiell laufen beide Modi identisch ab:

Es wird eine Frage und die Möglichkeit zur Eingabe der eigenen Antwort angezeigt, nach der Eingabe erfolgt die Auflösung.

Dieses System korrigiert nicht eigenständig, da unterschiedliche Formulierungen und Wortwahlen die Auswertung verkomplizieren.

Die Lösung wird also nach der Eingabe der Antwort angezeigt und der Geprüfte korrigiert bzw. bestimmt die Lösung als korrekt oder falsch eigenständig. Nach der Korrektur folgt die nächste Frage.

Nach der Beantwortung der gewählten Anzahl von Fragen wird dem Geprüften eine Gesamtübersicht seiner Ergebnisse angezeigt.

Lern Modus

Bei diesem Lernmodus kann der Student - nach Angabe seines Fachbereichs - je nach Wunsch das Fachgebiet und die Obergrenze der Semesterzahlen und die Anzahl der Fragen vorgeben.

Es ist ihm dadurch möglich je nach Wunsch und Fähigkeiten die Fragen zu optimieren.

Prüfungsmodus

Bei dem Prüfungsmodus handelt es sich um eine Simulation einer Assistentenprüfung, d.h. die Obergrenze der Semesterzahlen ist vier (die Assistentenprüfung erfolgt nach dem 4. Semester) und alle Fachgebiete werden abgefragt.

Die Wahl der Fragen erfolgt nach dem Zufallsprinzip.

Umsetzung

In diesem Abschnitt beschreiben wir exemplarisch die Umsetzung der beschriebenen Funktionalität des e-Learning Tools.

Datenbank Schemata

Hier soll ein Einblick in die verschiedenen Tabellen und deren interne Zusammenhänge geboten werden.

Wie bereits erwähnt erfolgt nur ein grober Einblick allerdings keine detaillierte Erklärung der dargelegten Sachverhalte da dies den Rahmen dieser Arbeit sprengen würde.

Folgend das reine Datenbankmodell mit seinen Attributen:

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
id	TINYINT(4)	PK	NN				
Frage_id	TINYINT(4)		NN				
Antwort	VARCHAR(255)		NN				
richtig	TINYINT(4)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	id					

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
id	TINYINT(4)	PK	NN				
Frage_id	TINYINT(4)		NN				
Antwort	VARCHAR(255)		NN				
Antwort_Nummer	VARCHAR(255)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	id					

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
id	TINYINT(4)	PK	NN				
Username	VARCHAR(50)		NN				
Passwort	VARCHAR(50)		NN				
Vorname	VARCHAR(50)		NN				
Name	VARCHAR(50)		NN				
Email	VARCHAR(30)		NN				
Typ	VARCHAR(15)		NN				
freigeschaltet	TINYINT(5)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	id					
Username	Index	Username					

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
id	TINYINT(5)	PK	NN				
Fachbereich	VARCHAR(50)		NN				
sichtbar	TINYINT(4)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	id					
Fachbereich	Index	Fachbereich					

Abbildung 6: Datenbankmodell mit seinen Attributen

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
id	TINYINT(5)	PK	NN				
Fachgebiet	VARCHAR(255)		NN				
Fachbereich	VARCHAR(255)		NN				
sichtbar	TINYINT(4)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	id					
Fachgebiet	Index	Fachgebiet					

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
Frage	VARCHAR(255)	PK	NN				
FrageTyp	VARCHAR(15)	PK	NN				
Antwort	VARCHAR(255)		NN				
Antwort_als_Bild	BLOB		NN				
Id	TINYINT(4)	PK	NN				
Semester	SMALLINT(2)		NN				
Fachgebiet	VARCHAR(255)		NN				
Fachbereich	VARCHAR(25)	PK	NN				
Professor	VARCHAR(25)	PK	NN				
sichtbar	TINYINT(5)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	Frage FrageTyp Id Fachbereich Professor					

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
id	TINYINT(5)	PK	NN				
Typ	LONGTEXT		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	id					

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
history_id	TINYINT(5)	PK	NN				
Username	VARCHAR(255)	PK	NN				
Datum	DATETIME	PK	NN		0000-00-00 00:00:00		
Anzahl_Fragen	SMALLINT(6)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	history_id Username Datum					

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
history_id	VARCHAR(255)		NN				
Frage	VARCHAR(255)		NN				
correct	VARCHAR(255)		NN				

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
id	INTEGER(5)	PK	NN				
Name	VARCHAR(50)		NN				
Berechtigung	VARCHAR(50)		NN				
IndexName	IndexType	Columns					
PRIMARY	PRIMARY	id					

Abbildung 7: Datenbankmodell mit seinen Attributen

Datenbankschema ohne Relationen

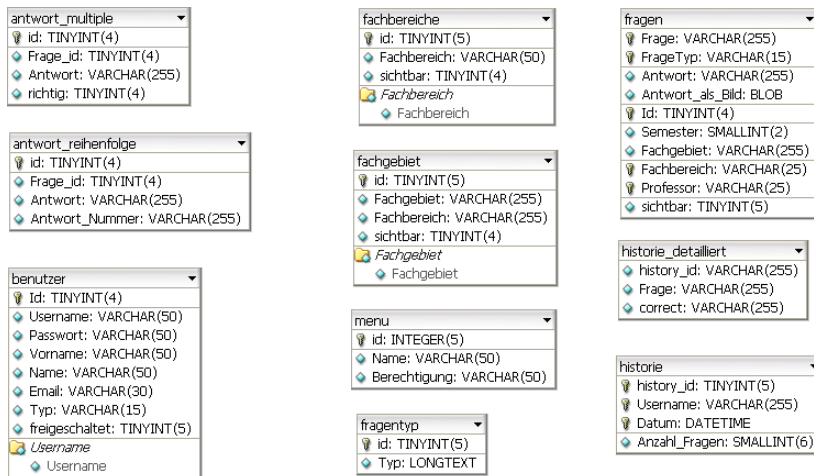


Abbildung 8: Datenbankschema ohne Relationen

Datenbankschema mit Relationen

Folgend das Datenbankschema mit allen Relationen, aufgrund der Funktionalität finden sich nicht alle in der Grafik dargestellten Relationen in der Datenbank. Die nicht vorhandenen Relationen wurden in PHP durch Join-Anweisungen realisiert.

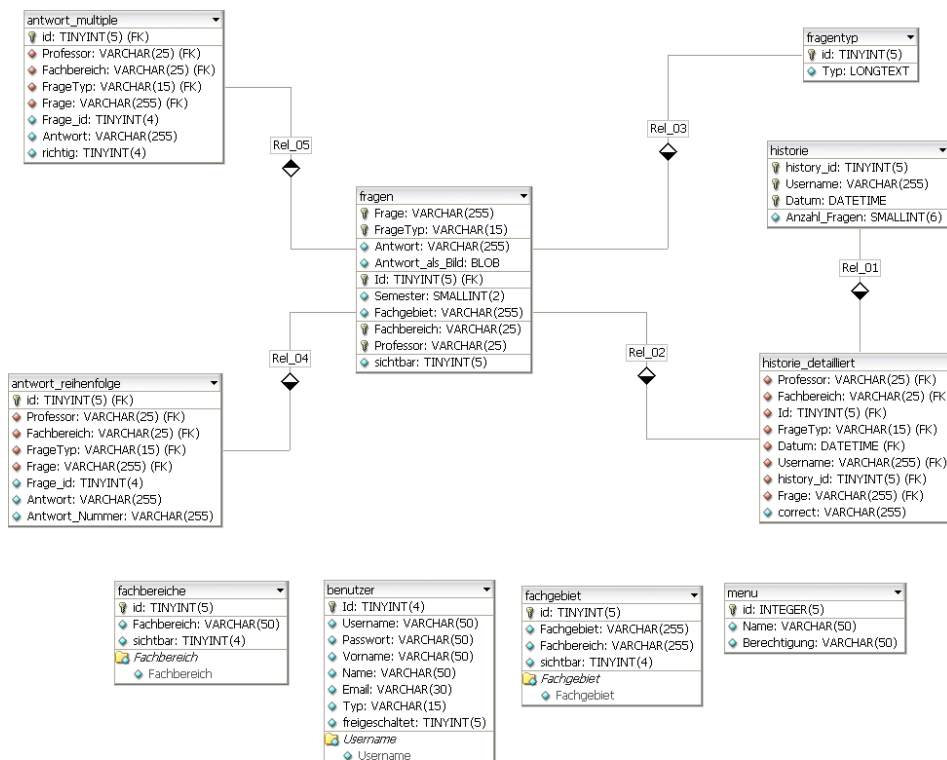


Abbildung 9: Datenbankschema mit allen Relationen

Datenbankverbindung und Abfragen

Die für die Datenbankverbindung (siehe Abschnitt „Zugriff mit PHP auf MySQL Datenbank“) notwendigen Daten sind in der Datei „config.inc“ enthalten. Diese wird in alle PHP Skripte eingebunden.

```
<?
// Datenbank Verbindungsdaten
//
//
//Host der Datenbank
$db_host="localhost";
//MySQL Benutzername
$db_user="root";
//MySQL Passwort
$db_pass="";
//Datenbankname
$databse="usr_web2_4";
//Anlegen einer Connection zu MySQL
$db_conn = mysql_connect($db_host, $db_user, $db_pass);
//Auswahl einer Datenbank
$db = mysql_select_db($databse,$db_conn);
//PHP Anzeigeeinstellungen der Fehlerverarbeitung
error_reporting(E_ERROR);
?>
```

Abbildung 10: Inhalt der Datei "config.inc"

Die Datenbankabfragen folgen diesem Schema:

```
$question_query="Select * from fragen where Id=".$quest_num."";
$question_result=mysql_query($question_query,$db_conn)
or die("Fehler15:".mysql_error());
```

Abbildung 11: DB-Abfrage nach Fragen für den Lern-Modus

Wobei diese Funktion im Falle eines Fehlers folgende Meldung erzeugt
„Fehler15:“, gefolgt von dem von MySQL erzeugten Fehlercode.

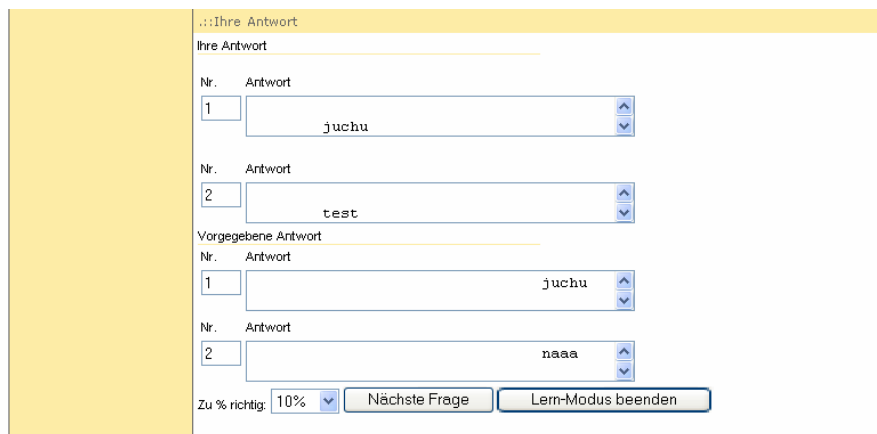
Mit folgendem Code wird jede Ergebniszeile von der Datenbankabfrage nach dem in der while-Schleife stehenden Schema abgearbeitet.

```
<?
while($answer_rows1 = mysql_fetch_array($answer_result1))
{
$answer_num2="answer1".$answer_rows["Antwort_Nummer"];
$answer_num3="1".$answer_rows["Antwort_Nummer"];
?>
<table>
<tr>
<td>Nr.</td>
<td>Antwort</td>
</tr>
<tr>
<td valign="top">
<input type="text" name="<? echo $answer_num3;?>" size="2" value="<?
echo $answer_rows1["Antwort_Nummer"];?>" >
</td>
<td><textarea name="<? echo $answer_num2;?>" rows="2"
cols="40" wrap="soft">
<? echo $answer_rows1["Antwort"];?></textarea>
</td>
</tr>
</table>
<?
}
?>
```

Abbildung 12: DB-Abfrage für die Ergebnisübersicht

Die gelb markierten Stellen sind direkte Ausgaben der Werte der Datenbank.

Die oben beschriebene Funktionalität würde folgende Ausgabe liefern:



Nr.	Antwort
1	juchu
2	test

Vorgegebene Antwort

Nr.	Antwort
1	juchu
2	naaa

Zu % richtig: 10%

Abbildung 13: Ausgabe einer Ergebnisübersicht

History Funktion

Jedem Nutzer des E-Learning Tools steht eine History zur Verfügung, welche dem Nutzer eine Übersicht über seine durchgeführten Tests ermöglicht.

Umgesetzt wurde dies wie folgt:

Es wird eine Abfrage aller history-Daten des eingeloggten Benutzers durchgeführt. Für jeden Eintrag in der history wird eine Tabelle angelegt und die jeweils dazugehörigen detaillierten Angaben abgefragt.

Diese angelegte Tabelle enthält:

- das Datum der Prüfung (direkt aus der Datenbank)
- die Anzahl der Fragen bei der jeweiligen Prüfung (direkt aus der Datenbank)
- die in der Prüfung enthaltenen Fragen. Die geprüften Fragen sind hierbei nicht für Historie-Zwecke gespeichert, sondern beziehen sich auf die dazugehörige ID. (welche in der Tabelle „history_detailliert“ eingetragen ist.)
- Die jeweilige Wertung zu jeder Fragen und eine prozentuale Übersicht über die richtig beantworteten Fragen, also dem Anteil der positiv durchgeführten Tests.

```
while($row_history_detail= mysql_fetch_array($history_result_details))
{
$query_history_detail_frage="SELECT Frage,Fachgebiet FROM `fragen` where
Id='".$row_history_detail["Frage"]."';
$history_result_details_frage=mysql_query($query_history_detail_frage,$db_conn);
$history_detail_frage=mysql_fetch_array($history_result_details_frage);
echo " <table width='100%'>
<tr>
<td width='10%'></td>
<td width='54%'>".$history_detail_frage["Frage"]."</td>
<td>".$history_detail_frage["Fachgebiet"]."</td>
<td width='15%'>zu <b>".$row_history_detail["correct"]."</b>korrekt</td>
</tr>
</table>";
$sact_num=$sact_num+$row_history_detail["correct"];
$sact_fachgebiet[$sact_num2]=$history_detail_frage["Fachgebiet"];
}
```

Abbildung 14: Generieren der History-Liste eines Benutzers

Zusätzlich werden die Ergebnisse in den jeweiligen Fachgebieten dargestellt, indem alle Fragen eines Fachgebietes abgefragt werden, ihre Ergebnisse addiert und durch die Anzahl dieser Fragen dividiert.

Die Ausgabe erfolgt dann folgendermaßen:



.::History vom 2004-06-08 14:34:42			
Anzahl Fragen:			
2			
Erhaltene Fragen:			
HURRRRRAAAAA	Datenbanken	zu 10% korrekt	
Wer bin ich?	Datenbanken	zu 80% korrekt	
Ergebnisse:			
Datenbanken:		45% korrekt	

Abbildung 15: Ausgabe einer History

Um die Historie-Funktion zu gewährleisten werden Fragen, Fachgebiete oder gar Fachbereiche nicht vollkommen gelöscht (dies würde eine Lücke in der Historie bedeuten), sondern sie werden nur „ausgeblendet“, also weder in den jeweiligen Auswahlen angezeigt noch in den Frage-Modi eingesetzt.

Fragenkatalog für Lern-Modi

Anzeigen der Fachbereiche und Fachgebiete

Die vorhandenen und sichtbaren Fachbereiche werden aus der Datenbank ausgelesen und dem Benutzer zur Auswahl angezeigt.

Abhängig von dessen Wahl werden die entsprechenden Fachgebiete (wenn sie sichtbar sind, Erläuterung siehe Abschnitt „History-Funktion“) angezeigt und der Benutzer kann auch hier wählen.

Vor dem Durchführen übergibt der Student seine Entscheidungskriterien bezüglich Fachbereich, Fachgebiet und Obergrenze der Semester (siehe auch Abschnitt „Lernmodus“).

In dem folgenden Beispiel wird dies noch einmal verdeutlicht. Außerdem wird in dem Beispiel die Ids der Fragen bei den entsprechenden Fachbereich und Fachgebiet in ein Array geschrieben.

```
if(${fachgebiet} == "alle")
{
    $num_fragen=1;
    $query3 = "Select * from fachgebiet where Fachbereich='${fachbereich}' and sichtbar=0";
    $result3= mysql_query($query3, $db_conn) or die("Fehler4".mysql_error());
    while($row3 = mysql_fetch_array($result3))
    {
        $query4 = "Select id from fragen where Fachbereich='${fachbereich}' and Fachgebiet='${row3[Fachgebiet]}.' and Semester <= '$semester' and sichtbar=0";
        $result4= mysql_query($query4, $db_conn) or die ("Fehler5".mysql_error());
        while($row4 = mysql_fetch_array($result4))
        {
            $available_questions[$num_fragen]=$row4["id"];
            $num_fragen=$num_fragen+1;
        }
        $num_fragen=$num_fragen+1;
    }
}
```

Randomizer bei Fragenauswahl

Die aktuelle Fragennummer wird mit der angegebenen bzw. gewählten Anzahl von Fragen verglichen.

Wenn die Zahl größer ist, wird die Historie des Benutzers angezeigt, ist sie kleiner die nächste Frage, ist sie noch nicht vorhanden, hat der Test soeben erst begonnen und die erste Frage wird erst erstellt.

In dem folgenden Beispiel wird per Randomizer die folgende Frage ausgewählt:

In einer ersten Überprüfung wird geprüft, ob die *"Anzahl aller schon mit dieser Spezifikation beantworteten fragen"* gleich der maximalen Anzahl verfügbarer Fragen ist.

Wenn dies nicht der Fall ist, wird eine Zufallszahl zwischen 1 und der maximalen Anzahl verfügbarer Fragen gewählt und überprüft, ob diese Frage bereits geprüft wurde. Dies geschieht, indem überprüft wird, ob die Frage ID in dem Array der bereits beantworteten Fragen enthalten ist.

Fällt diese zweite Überprüfung negativ aus, wird die Frage dargestellt. Bei einem positiven Ergebnis wird eine neue Frage via Randomizer gewählt.

Fällt die erste Überprüfung positiv aus, wird weiterhin geprüft, ob mehr als ein Fragedurchlauf mit den zuvor gewählten Spezifikationen erfolgt ist und stellt die Frage dar.

```
function get_question
    |({$num_avail_questions,$num_chosen_questions,$avail_questions,$done_questions,$num_done_questions,$quest_at_the_moment,$fragenzahl,$fachbereich,$fachgebiet,$semester,$history_id)
    {
        include("config.inc");
        $random_number = rand(1,$num_avail_questions);
        $act_question=$avail_questions[$random_number];
        if(array_search($act_question,$done_questions) == false)
        {
            echo "Ihre neue Frage: ".$avail_questions[$random_number]."<br><br>";
            show_question($avail_questions[$random_number],$fragenzahl,$quest_at_the_moment,$fachbereich,$fachgebiet,$semester,$history_id);
        }
        else
        {
            get_question($num_avail_questions,$num_chosen_questions,$avail_questions,$done_questions,$num_done_questions);
        }
    }
    return $act_question;
}
```

Frage darstellen

Die Frage mit der ermittelten Id (siehe Abschnitt „Randomizer bei Fragensauswahl“) wird in der Datenbank abgefragt.

Zur Übersicht für den Benutzer wird bei jeder Frage die Anzahl der bereits in dieser Prüfung beantworteter Fragen und die Anzahl ausgewählter Fragen angezeigt.

Die Frage wird entsprechend des Frage-Templates angezeigt. Abhängig von dem Antworttyp wird eines der folgenden Antwort-Templates ausgewählt:

- | | |
|--------------|--|
| Antworttyp=1 | Es wird eine Textarea dargestellt |
| Antworttyp=2 | Es wird eine Textarea eingeblendet und ein Hinweis das die Antwort ein Bild zur genaueren Erläuterung enthält. |
| Antworttyp=3 | Es wird abgefragt wie viele Antworten für diese Frage hinterlegt sind und deren Reihenfolge wird erstellt.
Abhängig von der ermittelten Anzahl werden Textfelder und Nummernboxen angezeigt
Es werden alle für diese Frage zur Auswahl stehenden |
| Antworttyp=4 | Antworten ausgewählt.
Abhängig von der Anzahl dieser werden die Antworten und dazugehörige Checkboxen erstellt und angezeigt. |

Zur Verdeutlichung folgt ein Beispiel:

```
if($answer_id=="4")
{
include("config.inc");
$answer_query="Select * from antwort_multiple where Frage_id=".$quest_num."";
$answer_result=mysql_query($answer_query,$db_conn) or die("Fehler8:".mysql_error());
?>
<br>
Bei dieser Frage müssen Sie die richtige Antwort(en) wählen.
<br>
<table>
<tr>
<td>Richtig?</td>
<td>Antwort</td>
</tr>
</table>
<?
while($answer_rows = mysql_fetch_array($answer_result))
{
$answer_num=$answer_rows["id"];
?>
<tr>
<td>
<input type="checkbox" name="seite[]" value="<? echo $answer_num;?>" size="2">
</td>
<td>
<? echo $answer_rows["Antwort"]; ?>
</td>
</tr>
<?
}
?>
</table>
<?
}
?>
?>
```

Überprüfung der Frage

Nach dem beantworten der Frage und durch das drücken von „weiter“ wird die Antwort angezeigt.

Es wird die Ergebniskorrektur angeboten und es muss die Korrektheit der Antwort in 10 %Schritten angegeben werden.

Diese Ergebnisse werden dann in die Datenbank eintragen.

```
function get_questions_allready_done($valid_user,$num_avail_questions,$avail_questions)
{
include("config.inc");
$query1 = "Select hd.Frage from historie AS h,historie_detailliert AS hd where h.Username='".$valid_user.'" and h.history_id = hd.history_id";
$result1 = mysql_query($query1,$db_conn) or die("Fehler3:".mysql_error());
while($row1 = mysql_fetch_array($result1))
{
$num_done_questions=0;
for($n=0;$n<$num_avail_questions;$n++)
{
if($row1["Frage"]=="$avail_questions[$n])
{
$done_questions[$num_done_questions] = $row1["Frage"];
}
}
}
return $done_questions;
}
```

Anhang

Übersicht über die Komponenten von XAMPP

Datei (Verzeichnis)	Bedeutung
\xampp\apache\logs	Logverzeichnis für Apache und PHP.
\xampp\cgi-bin\	Das CGI-BIN-Verzeichnis.
\xampp\apache\conf\httpd.conf	Die zentrale Konfigurations-Datei für den Apache.
\xampp\apache\bin\php.ini	Die zentrale Konfigurations-Datei für PHP mit MOD_PHP
\xampp\htdocs\	Das zentrale Dokumentenverzeichnis für HTML, PHP, CGI ..
\xampp\install\	Notwendig für das erste Setup ...
\xampp\mysql\	Der Datenbankserver MySQL
\xampp\perl\	Das Perl-Verzeichnis.
\xampp\php\	php.exe + dlls + pear usw.
\xampp\phpmyadmin\config.inc.php	Die Konfigurations-Datei für phpMyAdmin.
\xampp\tmp	Temp. Ordner für PHP-Uploads, Sessions usw.
\xampp\moddav	MOD-DAV Beispielordner.

Abbildungsverzeichnis

ABBILDUNG 1: SCREENSHOT DES E-LEARNING SYSTEMS	3
ABBILDUNG 2: LOGO DER XAMPP DISTRIBUTION	8
ABBILDUNG 3: HTML-CODE ALS BEISPIEL FÜR EINE EINBINDUNG VON PHP	10
ABBILDUNG 5: BENUTZERVERWALTUNG FÜR ADMINISTRATOREN	19
ABBILDUNG 6: DATENBANKMODELL MIT SEINEN ATTRIBUTEN	23
ABBILDUNG 7: DATENBANKMODELL MIT SEINEN ATTRIBUTEN	24
ABBILDUNG 8: DATENBANKSCHEMA OHNE RELATIONEN	24
ABBILDUNG 9: DATENBANKSCHEMA MIT ALLEN RELATIONEN	24
ABBILDUNG 10: INHALT DER DATEI CONFIG.INC	25
ABBILDUNG 11: DB-ABFRAGE NACH FRAGEN FÜR DEN LERNMODUS	25
ABBILDUNG 12: DB-ABFRAGE FÜR DIE ERGEBNISÜBERSICHT	26
ABBILDUNG 13: AUSGABE EINER ERGEBNISÜBERSICHT	27
ABBILDUNG 14: GENERIEREN DER HISTORYLIST EINES BENUTZERS	27
ABBILDUNG 15: AUSGABE EINER HISTORY	29

Quellen

<http://www.apachefriends.org/xampp.html>

<http://www.php.net/manual/de/index.php>

<http://www.mysql.com/documentation/mysql/bychapter/>

<http://www.schattenbaum.net/php/>

<http://selfphp.teamone.de/>

<http://www.teamone.de/selfhtml>

<http://www.phpfreaks.com/>

<http://www.mysqlfreaks.com/>

Unterstützende Professoren

Vielen Dank an dieser Stelle an die folgenden Professoren für die Bereitschaft uns bei unserer Studienarbeit mit Rede und Antwort zu unterstützen. Insbesondere durch das liefern von Beispiel-Fragen für den Fragenkatalog.

Herr Prof. Schmidt

Herr Prof. Colgen