

# Webbasiertes relationales Datenbanksystem für die Vereinigung Kanu-Guide Schweiz

Seminararbeit

Eingereicht bei: Prof. Dr. Andreas Meier  
Betreuer: Darius Zumstein

Information System Research Group  
Departement für Informatik  
Wirtschafts- und Sozialwissenschaftliche Fakultät  
Universität Freiburg, Schweiz

Verfasser: Christoph Kuert  
Turnhallenstrasse 10  
4900 Langenthal  
christoph.kuert@unifr.ch  
05-201-553

Winter 2009/2010

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>3</b>
<b>Tabellenverzeichnis</b>	<b>3</b>
<b>Abkürzungsverzeichnis</b>	<b>3</b>
<b>1. Einleitung</b>	<b>4</b>
<i>1.1 Problemstellung</i>	<i>4</i>
<i>1.2 Zielsetzung</i>	<i>4</i>
<i>1.3 Vorgehensweise</i>	<i>4</i>
<b>2. Die Vereinigung Kanu-Guide Schweiz</b>	<b>5</b>
<b>3. Entwicklung der Datenbank</b>	<b>6</b>
<i>3.1 Ausgangslage</i>	<i>6</i>
<i>3.2 Anforderungskatalog</i>	<i>6</i>
<i>3.3 Datenanalyse</i>	<i>10</i>
<i>3.4 Entitäten-Beziehungsmodell</i>	<i>11</i>
<i>3.5 Relationales Datenbankschema</i>	<i>13</i>
<b>4. Realisierung des Systems</b>	<b>15</b>
<i>4.1 Umsetzung mittels MySQL</i>	<i>15</i>
<i>4.2 Datenbankbenutzer</i>	<i>18</i>
<i>4.3 Benutzerschnittstelle</i>	<i>19</i>
<i>4.3.1 Herstellen der Verbindung zur Datenbank</i>	<i>19</i>
<i>4.3.2 Datenbankabfrage mit PHP</i>	<i>19</i>
<i>4.3.3 Erstellen einer Prüfung</i>	<i>19</i>
<i>4.3.5 Anmelden für die Prüfung</i>	<i>21</i>
<i>4.3.6 Prüfungsergebnisse eintragen</i>	<i>21</i>
<i>4.3.7 Ausgebildete Kanu-Guides anzeigen</i>	<i>22</i>
<b>5. Schlusswort</b>	<b>24</b>
<b>Literaturverzeichnis</b>	<b>25</b>
<b>Anhang</b>	<b>26</b>

## Abbildungsverzeichnis

Abbildung 2-1:	Logo des Schweizerischen Kanuverbands	5
Abbildung 2-2:	Logo der Swiss Outdoor Association	5
Abbildung 3.1-1:	Screenshot der bestehenden Webseite	6
Abbildung 3.2-1:	Screenshot des Backends	8
Abbildung 3.2-2:	UML-Aktivitätsdiagramm: Prüfungsanmeldung	9
Abbildung 3.4-1:	Entitäten-Beziehungsmodell	11
Abbildung 4.1-1:	Hostpoint ControlPanel: Datenbank erstellen	15
Abbildung 4.2-1:	Hostpoint ControlPanel: Datenbankbenutzer	18
Abbildung 4.3-1:	Screenshot: Prüfungserstellung	20
Abbildung 4.3-2:	Screenshot: Kanu-Guide erfassen	20
Abbildung 4.3-3:	Screenshot: Prüfung auswählen	21
Abbildung 4.3-4:	Screenshot: Prüfungsergebnisse eintragen	22
Abbildung 4.3-5:	Screenshot: ausgebildete Kanu-Guides	23

## Tabellenverzeichnis

Tabelle 3.2-1:	Der Aufbau des Frontend der Webseite	7
Tabelle 4.2-1:	SQL-Befehle	18

## Abkürzungsverzeichnis

CPR	<b>C</b> ardiopulmonary <b>r</b> esuscitation, Herz-Lungen-Wiederbelebung
CSV	<b>C</b> omma- <b>S</b> eparated <b>V</b> alues
HTML	<b>H</b> ypertext <b>M</b> arkup <b>L</b> anguage
HTTP	<b>H</b> ypertext <b>T</b> ransfer <b>P</b> rotocol
KGID	<b>K</b> anu- <b>G</b> uide <b>I</b> dentifikationsnummer
LAMP	<b>L</b> inux, <b>A</b> pache, <b>M</b> ySQL, <b>P</b> erl oder <b>P</b> HP
PDF	<b>P</b> ortable <b>D</b> ocument <b>F</b> ormat
PHP	<b>P</b> HP: <b>H</b> ypertext <b>P</b> reprocessor, eine Skriptsprache
SKV	<b>S</b> chweizerischer <b>K</b> anuverband
SQL	<b>S</b> tructured <b>Q</b> uery <b>L</b> anguage
SOA	<b>S</b> wiss <b>O</b> utdoor <b>A</b> ssociation
UML	<b>U</b> nified <b>M</b> odeling <b>L</b> anguage
VKGS	<b>V</b> ereinigung <b>K</b> anu- <b>G</b> uide <b>S</b> chweiz

# 1. Einleitung

## 1.1 Problemstellung

Die Vereinigung Kanu-Guide Schweiz ist verantwortlich für die Ausbildung der Schweizer Kanu-Guides.

Um die Sicherheit im touristischen Bereich des Kanusports zu fördern, wurden Ausbildungsrichtlinien erarbeitet. Kanu-Guides aus der ganzen Schweiz sind nun aufgefordert diese Prüfungen abzulegen. Die Kanu-Guides müssen sich entsprechend für eine Prüfung anmelden können. Auch sollen die Kanu-Guides online aufgeführt werden, damit sich Interessierte über die ausgebildeten Kanu-Guides informieren können.

Die Vereinigung Kanu-Guide Schweiz hat sich zum Ziel gesetzt, dies mittels einer Webseite zu tun.

Im Winter 2008 wurde die erste Version des Systems implementiert. In der ersten Phase der Nutzung wurde festgestellt, dass man noch mehr Ansprüche an das Online-System hat.

## 1.2 Zielsetzung

Am Ende der Seminararbeit soll eine solide webbasierte Datenbank zur Verfügung stehen. Die Administratoren sollen die Inhalte der Webseite, wie auch die Prüfungen administrieren können. Dies soll mittels einer webbasierten Oberfläche geschehen. Es soll aber auch möglich sein, die Kanu-Guides, welche sich für eine Prüfung angemeldet haben, auf einem lokalen Computer in einem Tabellenkalkulationsprogramm weiter zu verarbeiten.

Des Weiteren soll mit dieser Arbeit eine übersichtliche Dokumentation über das vorliegende System resultieren, welche zukünftige Weiterentwicklungen vereinfachen sollen.

## 1.3 Vorgehensweise

In einem ersten Schritt muss abgeklärt werden, was genau mit dem System bezweckt werden soll. Aus diversen Gesprächen wird ein Anforderungskatalog erarbeitet, dieser dient der Datenmodellierung. Weiter entsteht Schritt für Schritt ein relationales Datenbankschema.

In einem zweiten Schritt wird die MySQL-Datenbank erstellt.

Schlussendlich wird der HTML- und PHP-Code entwickelt, welcher mit der Datenbank arbeitet.

## 2. Die Vereinigung Kanu-Guide Schweiz



Abb. 2-1: Logo SKV



Abb. 2-2: SOA

Die Vereinigung Kanu-Guide Schweiz (VKGS) wurde gemeinsam vom Schweizerischen Kanuverband (SKV) und von der Swiss Outdoor Association (SOA) ins Leben gerufen.

Vor der Gründung der VKGS fehlte in der Schweiz im Bereich des touristischen Kanusports mit Erwachsenen klare Richtlinien in der Ausbildung zum Kanu-Guide. Im Gegensatz dazu besteht seitens des Bundesamtes für Sport mit dem Projekt Jugend & Sport ein klares Ausbildungskonzept im Vereinsbereich.

Seit 2008 werden mehrmals pro Jahr Prüfungen abgenommen für die Stufen Guide 1 (Flachwasser/See) und Guide 2 (bis Wildwasser II) in den Bootskategorien Kajak oder Canadier.

Nebst dem Erlassen von Richtlinien im Bereich der Kanu-Guide-Ausbildung fällt unter anderem auch die Organisation der Prüfungen in die Zuständigkeit der Vereinigung.

Die Administration wird von der Geschäftsstelle des Schweizerischen Kanuverbandes in Zürich getätigt.

## 3. Entwicklung der Datenbank

### 3.1 Ausgangslage

Bereits im Winter 2008 wurde eine erste, einfache Version des Kanu-Guide Systems im Netz zur Verfügung gestellt. Dies hat den damaligen Anforderungen mit dem Erfassen der Anmeldungen für die Prüfungen und dem Veröffentlichen der geprüften Kanu-Guides Rechnung getragen. Mit dem aktiven Nutzen des Systems entstanden weitere Bedürfnisse. In erster Linie wollte man die eingegebenen Daten der Kanu-Guides auch Online anpassen können. Diese Funktionen waren zu Beginn nicht gefordert und somit auch nicht implementiert.

Da das System erst zwei Saisons im Einsatz ist, sind noch nicht viele Einträge in der jetzigen Datenbank erfasst. Der Übertrag in die neue Datenbank kann somit ohne grossen Aufwand manuell erfolgen.

Die Webseite ist in einem schlichten, übersichtlichen Design gehalten und ist auf Französisch wie auf Deutsch verfügbar. Die Vereinigung Kanu-Guide Schweiz verfügt noch über kein Logo, aus diesem Grund wurde als Titel eine einfache Schrift gewählt.

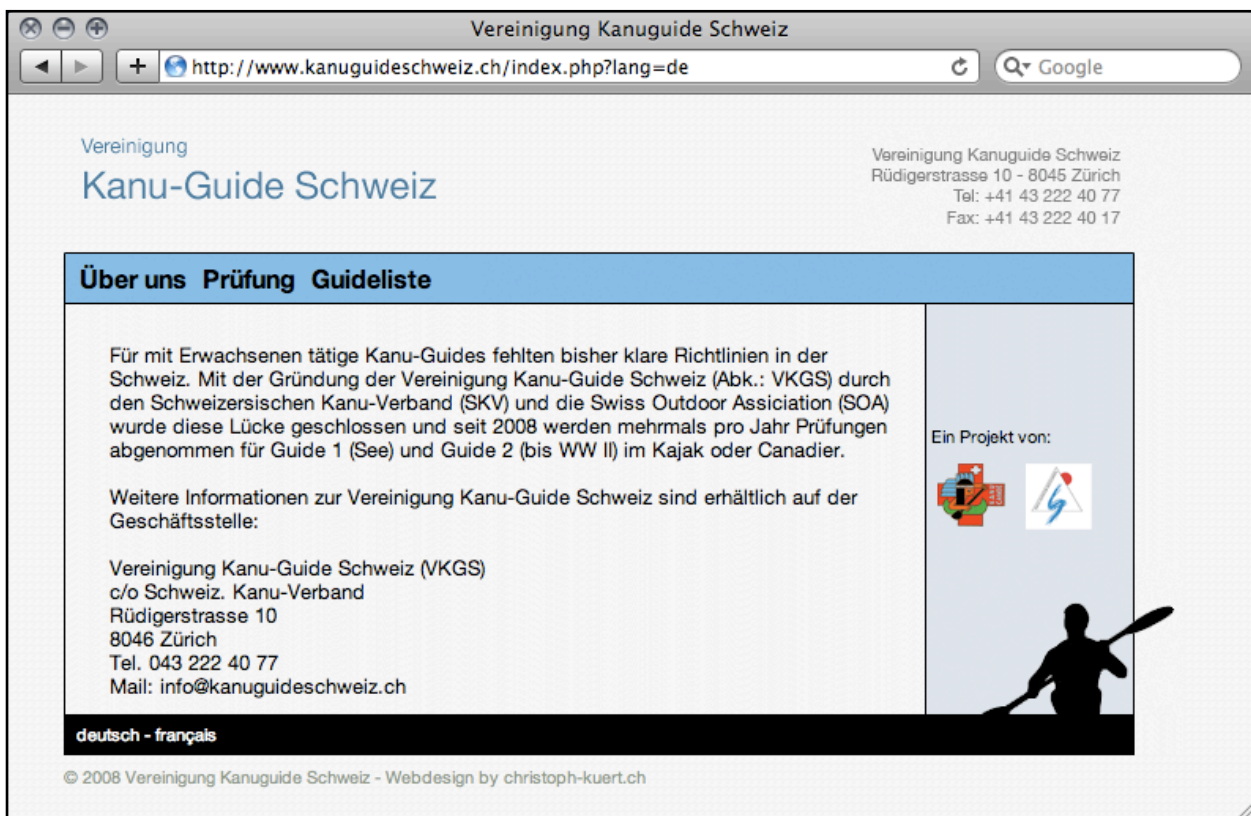


Abb. 3.1-1: Screenshot der bestehenden Webseite

### 3.2 Anforderungskatalog

Um die Anforderungen an die Datenbank und das ganze System festzulegen, wurden diverse Gespräche mit den zukünftigen Benutzern geführt. Insbesondere war dies, Reto Wyss, er ist der Geschäftsführer des SKV und wird die Administration des Systems übernehmen.

[Kemper, Eickler 2006] erwähnen den Einsatz eines Datenbankverwaltungssystems als Motivation unter anderem die Redundanz und Inkonsistenz der Daten. Durch den Einsatz

eines Datenbankverwaltungssystem kann sichergestellt werden, dass sämtliche Personen immer die aktuellen Daten zur Verfügung haben. Bei der VKGS geht es soweit, dass die Datenbank via Internet erreichbar ist und potentiellen Kunden jederzeit Informationen über die ausgebildeten Kanu-Guides erhalten können.

Im vorliegenden Anforderungskatalog wurden die bereits vorhandenen, wie auch die neu entstandenen Anforderungen zusammengeführt.

Die Anforderungen wurden in zwei Teile gegliedert, einerseits das Frontend und andererseits das Backend. Im Frontend wird definiert, was ein Besucher auf der Webseite alles ansehen kann und welche Funktionen ihm zur Verfügung stehen werden. Im Backend wird der gesamte administrative und redaktionelle Bereich untergebracht sein.

### **Frontend:**

- Die Webseite soll zweisprachig zur Verfügung stehen (Französisch und Deutsch).
- Auf der Webseite sollen sich Interessierte über die Vereinigung Kanu-Guide Schweiz informieren können.
- Die Liste der ausgebildeten Kanu-Guides soll für alle Besucher der Webseite abrufbar sein. Folgende Angaben sollen veröffentlicht werden: Nachname, Vorname, Ort, Prüfungsdatum, Prüfungsart.
- Angehende Kanu-Guides sollen sich für die ausgeschriebenen Prüfungen anmelden können.
- Es soll lediglich möglich sein, sich für Prüfungen anzumelden, welche noch nicht ausgebucht sind und die Anmeldefrist noch nicht abgelaufen ist.
- Ein Kanu-Guide muss folgende Informationen über sich angeben, wenn er sich für eine Prüfung anmelden will: Nachname, Vorname, Adresse, PLZ, Ort, Geburtsdatum, E-mail, Telefon und Organisation.
- Falls ein Kanu-Guide bereits für eine Prüfung angemeldet hat, kann er sich für weitere Prüfungen mittels Angabe seiner Kanu-Guide Nummer anmelden.
- Um an einer Prüfung teilnehmen zu können, muss der Kanu-Guide über ein gültiges CPR-Brevet verfügen. Sofern es gewünscht wird, kann ein CPR-Grund- oder Fortbildungskurs mit der Kanu-Guide Prüfung kombiniert werden.
- Die Webseite soll folgenden Aufbau haben:

Tab. 3.2-1: Der Aufbau des Frontend der Webseite

Hauptmenu	Untermenu	Bemerkung
Über uns	-	Die VKGS wird vorgestellt.
Prüfungen		Allgemeine Informationen werden publiziert.
	Rahmenbedingungen	Die Rahmenbedingungen werden publiziert.
	Daten	Die zukünftigen Prüfungen sollen angezeigt werden. Inkl. einem Link zur Anmeldung, sowie falls vorhanden, ein prüfungsspezifisches Dokument.
	Unterlagen	Es stehen weitere allgemeine Informationen zu den Prüfungen zur Verfügung.
	Anmeldung	Die Anmeldung für die zukünftigen Prüfungen erfolgt hier.
Guideliste	-	Die ausgebildeten Kanu-Guides werden in der entsprechenden Kategorie aufgelistet.

**Backend:**

- Um Zugriff auf das Backend des Systems zu erhalten, muss sich der Administrator mittels Benutzernamen und Passwort authentifizieren.
- Der Administrator muss die Inhalte der Webseite bearbeiten können. Diese Daten sind unabhängig von den Prüfungen und Kanu-Guides.
- Der Administrator soll Unterlagen (z.B. ein PDF-Dokument) publizieren können.
- Der Administrator muss Prüfungen erstellen können. Folgende Angaben zur Prüfung sollen definiert werden: Prüfungsart, Prüfungsdatum, Anmeldefrist, maximale Teilnehmerzahl. Optional soll es möglich sein, der Prüfung ein Dokument mit zusätzlichen Informationen zur Prüfung anzuhängen.
- Der Administrator soll sehen können, wer sich für eine Prüfung angemeldet hat. Des Weiteren soll die Möglichkeit bestehen, die Anmeldungen lokal auf dem Computer zu speichern und in einem Tabellenkalkulationsprogramm weiterzuverarbeiten.
- Es gibt vier verschiedene Prüfungsarten. Jeweils die Stufe 1 und 2 in den beiden Bootstypen Kajak und Canadier.
- Da es bei den Prüfungen eine Teilnehmerbeschränkung gibt, werden die Anmeldungen in der Eingangsreihenfolge beachtet - first come, first served. Sobald die Teilnehmerzahl erreicht ist, sind keine Anmeldungen mehr möglich.
- Der Administrator soll die Kanu-Guides verwalten können. Er soll neue Guides erfassen, Bestehende verändern oder löschen können.
- Der Administrator soll nach erfolgreicher Prüfung die Kanu-Guides in der Guideliste des Frontend veröffentlichen können.

Die graphische Gestaltung des Backend ist wie das Frontend gehalten [Abbildung 3.2-1]. Der Administrator wird nach erfolgreicher Authentifizierung begrüsst und hat die Möglichkeit, mit den Daten zu arbeiten. Die Funktionen sind in die drei Kategorien Inhalte, Prüfung und Guides aufgeteilt.



Abb. 3.2-1: Screenshot des Backends

Um den Ablauf einer Prüfungsanmeldung besser zu verstehen, wurden die verschiedenen Schritte in [Abbildung 3.2-2] mittels eines UML-Aktivitätsdiagramm visualisiert. Speziell zu erwähnen ist, dass zwischen dem Schritt des E-Mail-Versandes und dem Erfassen der Prüfungsergebnisse, die Kanu-Guide Prüfung noch stattfinden muss.

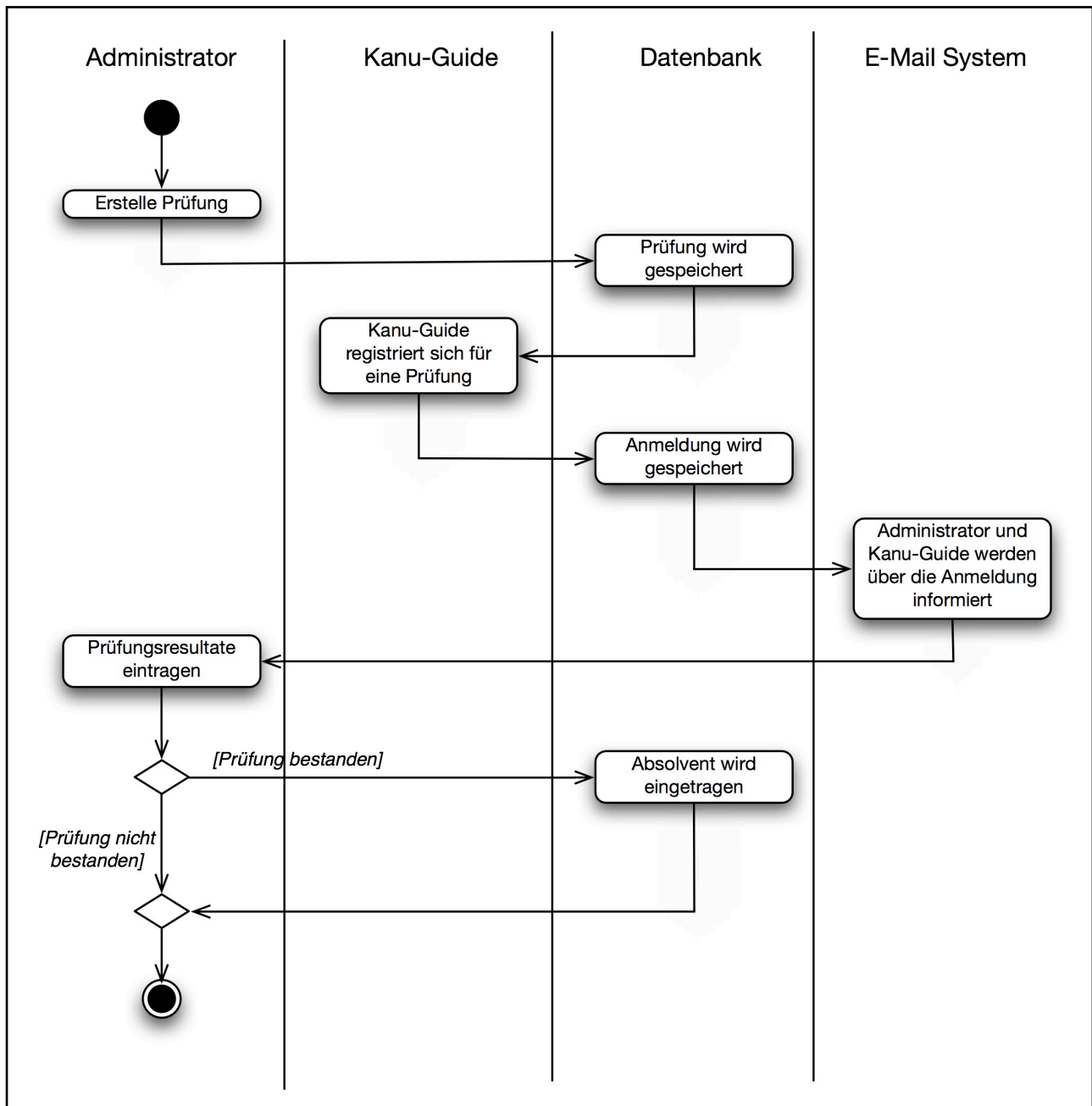


Abb. 3.2-2: UML-Aktivitätsdiagramm: Prüfungsanmeldung

### 3.3 Datenanalyse

Bei der Datenanalyse geht es darum, die Erkenntnisse aus dem Anforderungskatalog zu strukturieren. Die Datenanalyse bildet die Grundlage des Entitäten-Beziehungsmodells (Abschnitt 3.4). Dies sagt bereits viel über die Datenanalyse aus. Es sollen die Entitäten, ihre Attribute und auch die Beziehungen untereinander erarbeitet werden.

- **Webseiteninhalte:**

Da die Webseite zweisprachig geführt wird, soll jeder Datensatz über eine Übersetzung verfügen.

Auf einer Seite sollen allgemeine Unterlagen zu den Prüfungsdisziplinen publiziert werden.

- **Kanu-Guides:**

Bei jedem Kanu-Guide werden folgende Angaben gespeichert: Vorname, Nachname, Adresse, PLZ, Ort, Geburtsdatum, E-Mail, Telefon, Organisation und Sprache. Die Sprache wird aufgrund der gewählten Sprache der Webseite vorgeschlagen. Mögliche Sprachen sind Deutsch oder Französisch.

- **Prüfungen:**

Eine Prüfung besteht aus folgenden Angaben: Art der Prüfung, Prüfungsdatum, Anmeldefrist, Teilnehmerzahl. Optional ist es möglich eine Datei mit weitergehenden Informationen mit der Prüfung zu verknüpfen.

- **Anmeldungen:**

Ein Kanu-Guide meldet sich für eine Prüfung an, indem er unter anderem seine aktuelle CPR-Ausbildung angibt.

Die Anmeldung enthält eine Prüfungsnummer, eine Kanu-Guidenummer, das Anmeldedatum, sowie den CPR-Ausbildungsstand. Folgende drei Varianten der CPR-Ausbildung sind möglich: „besucht den Grundkurs“, „besucht den Fortbildungskurs“ und „Ausbildung bereits absolviert“.

- **Unterlagen:**

Es können Dokumente zur Verfügung gestellt werden. Sie können auf der Seite der allgemeinen Informationen im Frontend angezeigt oder direkt mit einer Prüfung verknüpft werden.

### 3.4 Entitäten-Beziehungsmodell

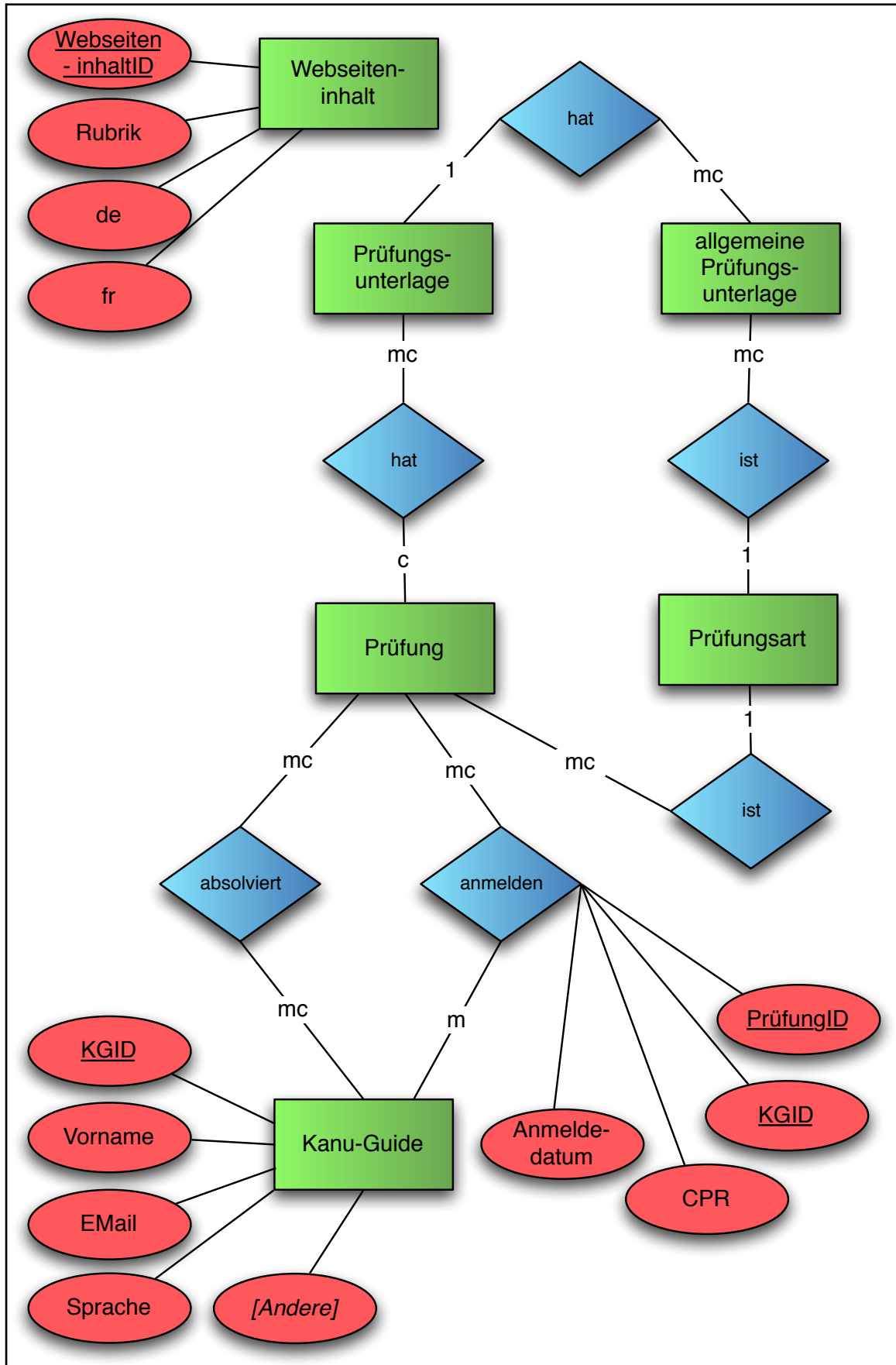


Abb. 3.4-1: Entitäten-Beziehungsmodell

Es folgen nun einige grundlegende Erläuterungen zum Entitäten-Beziehungsmodell [Abbildung 3.4-1] vom KGS-System.

- Die grünen Rechtecke sind Entitäten und werden in der Datenbank als Tabellen dargestellt.
- Die blauen Rhomben sind Beziehungen zwischen den Entitäten und müssen nach den Regeln in Abschnitt 4.5 verarbeitet werden.
- Als rote Ellipsen sind die Attribute der Entitäten und Beziehungen abgebildet. Um das Modell jedoch nicht zu überfüllen, wurden zur Illustration nur einige Attribute angehängt.

Es gibt vier Arten von Beziehungen [Meier 2007]:

- **1** heisst einfache Assoziation: es gibt „genau einen“ Tupel aus Entität A zu jedem Tupel aus Entität B
- **c** heisst konditionelle Assoziation: es gibt „keinen oder einen“ Tupel aus Entität A zu jedem Tupel aus Entität B
- **m** heisst mehrfache Assoziation: es gibt „einen oder mehrere“ Tupel aus Entität A zu jedem Tupel aus Entität B
- **mc** heisst mehrfach-konditionelle Assoziation: es gibt „keinen, einen oder mehrere“ Tupel aus Entität A zu jedem Tupel aus Entität B

Wenn diese vier Beziehungsarten auf das ER-Modell [Abbildung 3.4-1] angewendet werden, liest es sich folgendermassen:

- Ein Kanu-Guide kann sich für eine oder mehrere Prüfungen anmelden; eine Prüfung kann keinen, einen oder mehrere Kanu-Guides aufnehmen.
- Ein Kanu-Guide kann keine, eine oder mehrere Prüfungen bestehen; eine Prüfung kann von keinem, einem oder mehreren Kanu-Guides bestanden werden.
- Eine Prüfung hat eine Prüfungsart; eine Prüfungsart kann von keiner, einer oder mehreren Prüfungen verwendet werden.
- Eine Prüfung hat eine oder keine Prüfungsunterlage; eine Prüfungsunterlage kann von keiner, einer oder mehreren Prüfungen verwendet werden.
- Eine allgemeine Prüfungsunterlage hat eine Prüfungsunterlage; eine Prüfungsunterlage kann von keiner, einer oder mehreren allgemeinen Prüfungsunterlagen verwendet werden.
- Eine allgemeine Prüfungsunterlage hat eine Prüfungsart; eine Prüfungsart kann von keiner, einer oder mehreren allgemeinen Prüfungsunterlagen verwendet werden.

Der Spezialfall: Webseiteninhalt

Diese Entität ist losgelöst vom Modell. Sie dient lediglich der Speicherung der Textinhalte des Frontendes. Obwohl sie keine Beziehung zum restlichen Modell besitzt, ist sie zur Vollständigkeit des Systems essentiell.

### 3.5 Relationales Datenbankschema

Im letzten Schritt wird nun aus dem Entitäten-Beziehungsmodell das relationale Datenbankschema erarbeitet. Die Entitäten und Beziehungen werden nach den Regeln aus [Meier 2007] in Tabellen überführt. Grundsätzlich gilt, dass jede Entität in eine eigene Tabelle überführt werden muss, wobei bei den Beziehungen entschieden werden kann. Hier werden nun auch die Attribute, wie sie verwendet werden, aufgeführt. Es geht in diesem Schritt darum, die Datenbankobjekte formal zu beschreiben.

Weiter gilt es die Normalformen zu beachten. Es gibt sechs Stufen von Normalformen, wobei die nächsthöhere Form jeweils die Kriterien der vorhergehenden Normalform integriert. Die Normalformen dienen in erster Linie der Vermeidung von Redundanzen und können die Leistungsfähigkeit steigern.

Oft genügt es den Anforderungen, wenn sich die Tabellen in der dritten Normalform befinden. Es werden folgende Charakteristiken beachtet [Meier 2007]:

- 1. Normalform: Es kommen keine Wiederholungsgruppen vor, alle Werte sind atomar.
- 2. Normalform: Nichtschlüsselmerkmale sind voll vom Schlüssel abhängig.
- 3. Normalform: Es bestehen keine transitiven Abhängigkeiten.

In der Tabelle „Kanuguide“ wird jedoch die dritte Normalform verletzt. Denn wenn man die Postleitzahl kennt, kann man den Ort ermitteln. Diese Auslagerung macht aber hier keinen Sinn und kann somit vernachlässigt werden.

Folgende Angaben dienen der Leserlichkeit: Tabellen werden **fett** benannt, Schlüssel werden unterstrichen, Fremdschlüssel werden *kursiv* gedruckt.

#### Webseiteninhalt

- WebseiteninhaltID
- Rubrik
- de
- fr

#### Prüfung

- PrüfungID
- *PrüfungsartID*
- *PrüfungsunterlageID*
- Datum
- Anmeldefrist
- Teilnehmerzahl

#### Prüfungsart

- PrüfungsartID
- Beschreibung

#### Prüfungsunterlage

- PrüfungsunterlageID
- Unterlage
- Unterlagename
- Sprache

### **allgemeinePrüfungsunterlage**

- allgemeinePrüfungsunterlageID
- *PrüfungsunterlageID*
- *PrüfungsartID*
- Sprache

### **Kanuguide**

- KanuguideID
- Vorname
- Nachname
- Adresse
- PLZ
- Ort
- Geburtsdatum
- EMail
- Telefon
- Organisation
- Sprache

### **Anmeldung**

- PrüfungID
- KanuguideID
- CPR
- Anmeldedatum

### **Absolvent**

- PrüfungID
- KanuguideID

## 4. Realisierung des Systems

### 4.1 Umsetzung mittels MySQL

Warum sich heute viele Entwickler auf die Datenbank MySQL setzen, hat verschiedene Gründe. Im Internet hat sich das LAMP (Linux Betriebssystem, Apache Webserver, MySQL Datenbank und Perl respektive PHP Skriptsprache) etabliert. Alle diese Produkte sind Open Source und können gratis verwendet werden. Der Hostingpartner von KGS, die Hostpoint AG, stellt diese Konfiguration zur Verfügung.

Das von Hostpoint entwickelte ControlPanel dient der Verwaltung der einzelnen Komponenten der Webseite, z.B. die E-Mail Adressen aber auch die MySQL-Datenbanken.

Um die Datenbankinhalte zu administrieren, steht die Applikation phpMyAdmin bereit. Wie der Name vermuten lässt, handelt es sich um eine Applikation, welche mit PHP realisiert wurde und eben der Administration von MySQL Datenbanken dient. Sie bietet ein grafisches Interface, es besteht aber auch die Möglichkeit SQL-Befehle auszuführen.

Als Engine wird im ganzen System InnoDB gewählt aus dem einfachen Grund, dass InnoDB im Moment die einzige Speicherart ist, welche Fremdschlüsselbeziehungen prüfen kann. In [Tahaghoghi, Williams 2007] wird darauf verwiesen, dass das Überprüfen der Fremdschlüsselbeziehungen an die Applikation übergeben werden soll, wenn man nicht InnoDB verwendet.

Im folgenden Teil geht es um den MySQL-Code, welcher benötigt wird um die relationale Datenbank von Abschnitt 3.5 zu erzeugen.

Als erstes muss die Datenbank erzeugt werden:

```
mysql> CREATE DATABASE kgs2009;
```

Abb. 4.1-1: Hostpoint ControlPanel: Datenbank erstellen

In [Abbildung 4.1-1] wird gezeigt, dass im ControlPanel von Hostpoint keine SQL-Befehle möglich sind, um die Datenbank zu erzeugen. Mittels HTML-Formular wird der Datenbankname übergeben und im Hintergrund erstellt. Der Datenbankname setzt sich aus dem Hostpoint Benutzername und einem selbstwählbaren Teil zusammen. Der Hostpoint Benutzername wurde unkenntlich gemacht.

Hingegen ist es im phpMyAdmin möglich, die folgenden SQL-Befehle auszuführen, um die Tabellen zu erzeugen.

```
CREATE TABLE `Webseiteninhalt` (
  `WebseiteninhaltID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `Rubrik` VARCHAR( 20 ) NOT NULL ,
```

```

`de` TEXT NOT NULL ,
`fr` TEXT NOT NULL
) ENGINE = INNODB;

```

Die Tabelle Webseiteninhalt besitzt vier Attribute. Das Attribut WebseiteninhaltID ist der Primärschlüssel, es handelt sich um ein Integer, welcher mittels AUTO\_INCREMENT automatisch um eins erhöht wird. Diese Möglichkeit welche MySQL bietet, ermöglicht es, auf eine einfache Weise einen Primärschlüssel eindeutig zu definieren. Das Attribut Rubrik besitzt 20 Charaktere. Die letzten beiden Attribute de und fr sind vom Typ Text.

Bei den folgenden Tabellen werden jeweils lediglich die Neuerungen erklärt.

```

CREATE TABLE `Pruefungsart` (
  `PruefungsartID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `Beschreibung` VARCHAR( 50 ) NOT NULL
) ENGINE = INNODB;

```

```

CREATE TABLE `Pruefungsunterlage` (
  `PruefungsunterlageID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `Unterlage` VARCHAR( 150 ) NOT NULL ,
  `Unterlagenname` VARCHAR( 150 ) NOT NULL ,
  `Sprache` VARCHAR( 2 ) NOT NULL
) ENGINE = INNODB;

```

```

CREATE TABLE `Pruefung` (
  `PruefungID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `PruefungsartID` INT NOT NULL ,
  `PruefungsunterlageID` INT NOT NULL ,
  `Datum` DATE NOT NULL ,
  `Anmeldefrist` DATE NOT NULL ,
  `Teilnehmerzahl` INT NOT NULL ,
  FOREIGN KEY ( `PruefungsartID` ) REFERENCES `Pruefungsart` ( `PruefungsartID` ) ,
  FOREIGN KEY ( `PruefungsunterlageID` ) REFERENCES `Pruefungsunterlage`
( `PruefungsunterlageID` )
) ENGINE = INNODB;

```

Bei der Tabelle Pruefung werden erstmals Datumswerte gesetzt. Dies ermögliche das Verarbeiten von Datumswerten direkt in der SQL-Abfrage.

Des Weiteren sind hier auch die ersten Fremdschlüssel definiert. Zu beachten ist, dass die Reihenfolge der Erstellung eine Rolle spielt. Es ist logischerweise nicht möglich auf eine Tabelle zu referenzieren welche noch nicht erstellt ist.

```

CREATE TABLE `allgemeinePruefungsunterlage` (
  `allgemeinePruefungsunterlageID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `PruefungsunterlageID` INT NOT NULL ,
  `PruefungsartID` INT NOT NULL ,
  `Sprache` VARCHAR( 2 ) NOT NULL ,
  FOREIGN KEY ( `PruefungsartID` ) REFERENCES `Pruefungsart` ( `PruefungsartID` ) ,
  FOREIGN KEY ( `PruefungsunterlageID` ) REFERENCES `Pruefungsunterlage`
( `PruefungsunterlageID` )
) ENGINE = INNODB

```

```

CREATE TABLE `Kanuguide` (
  `KanuguideID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `Vorname` VARCHAR( 100 ) NOT NULL ,
  `Nachname` VARCHAR( 100 ) NOT NULL ,
  `Adresse` VARCHAR( 100 ) NOT NULL ,
  `PLZ` INT( 4 ) NOT NULL ,
  `Ort` VARCHAR( 100 ) NOT NULL ,
  `Geburtsdatum` DATE NOT NULL ,

```

```

`EMail` VARCHAR( 100 ) NOT NULL ,
`Telefon` VARCHAR( 50 ) NOT NULL ,
`Organisation` VARCHAR( 100 ) NOT NULL ,
`Sprache` VARCHAR( 2 ) NOT NULL
) ENGINE = INNODB;

CREATE TABLE `Anmeldung` (
`PruefungID` INT NOT NULL ,
`KanuguideID` INT NOT NULL ,
`CPR` VARCHAR( 100 ) NOT NULL ,
`Anmeldedatum` TIMESTAMP ON UPDATE CURRENT_TIMESTAMP NOT NULL ,
PRIMARY KEY ( `PruefungID` , `KanuguideID` ) ,
FOREIGN KEY ( `PruefungID` ) REFERENCES `Pruefung` ( `PruefungID` ) ,
FOREIGN KEY ( `KanuguideID` ) REFERENCES `Kanuguide` ( `KanuguideID` )
) ENGINE = INNODB;

```

In der Tabelle Anmeldung wie auch in Absolvent ist eine Variante der Definition von Primärschlüsseln. Dies vereinfacht es, sobald mehrere Attribute als Primärschlüssel dienen. Um das aktuelle Anmeldedatum einfach erfassen zu können, wird die Funktion ON UPDATE verwendet. Sie fügt automatisch die aktuelle Zeit ein.

```

CREATE TABLE `Absolvent` (
`PruefungID` INT NOT NULL ,
`KanuguideID` INT NOT NULL ,
PRIMARY KEY ( `PruefungID` , `KanuguideID` ) ,
FOREIGN KEY ( `PruefungID` ) REFERENCES `Pruefung` ( `PruefungID` ) ,
FOREIGN KEY ( `KanuguideID` ) REFERENCES `Kanuguide` ( `KanuguideID` )
) ENGINE = INNODB;

```

Die konkreten Inhalte der Datenbank werden lediglich via dem Webbasierten System eingefügt. Der entsprechende Programmiercode ist in Abschnitt 4.3 näher erläutert.

## 4.2 Datenbankbenutzer

MySQL erlaubt es verschiedene Benutzer zu generieren, welche unterschiedliche Rechte besitzen.

Die sicherste MySQL-Datenbank ist laut [Tabaghoghi, Williams 2007] diejenige ohne Benutzer und ohne jegliche Privilegien. Die flexibelste hingegen ist diejenige, in welcher sich alle als Systemadministrator anmelden.

Um die Sicherheit zu erhöhen, wurden insgesamt drei verschiedene Benutzer erstellt. Das Ziel ist, dass die Nutzer lediglich diejenigen Privilegien erhalten, um ihre Aufgabe zu erfüllen.

Der erste *kgsA* (Admin) wird für die Administration der Datenbank verwendet. Ihm stehen alle Rechte offen.

Für das Backend wurde der Benutzer *kgsB* erstellt. Im Backend muss es nicht möglich sein die Struktur der Datenbank zu verändern, lediglich dessen Inhalte.

Im Frontend wird der Benutzer *kgsF* verwendet, dieser muss ausschliesslich Inhalte auswählen können, um sie den Besuchern präsentieren zu können. Es ist aber trotzdem möglich, sich als Kanu-Guide zu registrieren. Für die Anmeldung wird der Benutzer *kgsB* verwendet. Somit kann der eigentliche Zweck der KGS-Webseite erfüllt werden. Die Abbildung 4.2-1 stellt die verschiedenen Datenbankbenutzer, wie sie im Controlpanel von Hostpoint konfiguriert werden können, dar. In der Tabelle 4.2-1 werden die verschiedenen SQL-Befehle kurz erläutert.

Tab. 4.2-1: SQL-Befehle

SQL-Befehl	Bedeutung
SELECT	wählt aus der Datenbank aus
INSERT	fügt in die Datenbank ein
UPDATE	ändert Tupel in der Datenbank
DELETE	löscht aus der Datenbank
CREATE	erzeugt eine Datenbank oder Tabelle
DROP	löscht eine Datenbank oder Tabelle
ALTER	ändert eine Datenbank oder Tabelle




Dieser Datenbank sind die folgenden Datenbankbenutzer zugewiesen:							
Datenbankbenutzer	SELECT	INSERT	UPDATE	DELETE	CREATE	DROP	ALTER
 _kgsA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
 _kgsB	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 _kgsF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Abb. 4.2-1: Hostpoint ControlPanel: Datenbankbenutzer

### 4.3 Benutzerschnittstelle

Die MySQL Datenbank wurde in [Abschnitt 4.1] erstellt. Doch die Datenbank alleine bringt noch keinen Nutzen. Sie bietet lediglich die Grundlage um den Anforderungen [Abschnitt 3.2] gerecht zu werden.

Um das Kernstück der Datenbank wird mittels HTML und PHP nun eine Benutzerschnittstelle erstellt welches Dateneingaben und -manipulationen zulässt. Des Weiteren werden die Daten so aufbereitet, dass sie Nutzergerecht angezeigt werden können.

Zum Erfassen der einzelnen Angaben werden HTML Formulare verwendet.

Formulare dienen der Dateneingabe und -korrektur. Sie haben typischerweise die Form einer Bildschirmmaske mit einem Feld für jeden Datenwert eines Datensatzes einer Daten- oder Abfragetabelle [Zehnder 1998, S. 186].

Die eingegebenen Werte werden mit einer HTTP-Request-Methode an die jeweiligen PHP-Elemente übergeben. Die PHP-Elemente verarbeiten die Werte und geben eine entsprechende Statusmeldung zurück. Im Anhang werden die wichtigsten Elemente erklärt.

#### 4.3.1 Herstellen der Verbindung zur Datenbank

```
$conn = @mysql_connect($server, $user, $pass);
mysql_select_db = ($dbase, $conn);
```

Mittels `@mysql_connect` wird die Verbindung zur Datenbank hergestellt. Es werden drei Parameter übergeben. `$server` ist die Adresse des Servers, `$user` ist der Benutzer und `$pass` ist das Passwort. Dies wird in der Variable `$conn` gespeichert welche ein erstes Mal bei der Auswahl der Datenbank mit `mysql_select_db` verwendet wird.

Bei jeder Datenbankabfrage muss diese Verbindung zu Stande kommen. Aus diesem Grund ist sie in einer separaten Datei gespeichert welche von den anderen Programmteilen verwendet werden kann.

Es gibt zwei Versionen dieser Datei. Eine für das Frontend und eines für das Backend, jeweils mit dem entsprechenden Benutzer (gemäss Abschnitt 4.2).

#### 4.3.2 Datenbankabfrage mit PHP

```
$SQL = "SELECT * FROM Pruefung";
$result = mysql_query($SQL, $conn);
while ($row = mysql_fetch_array($result)) {
    // Durchlaufen jedes Tuppels
}
```

Der obige Code gibt sämtliche Tuppel der Tabelle „Pruefung“ aus.

Für eine Datenbankabfrage sind grundsätzlich zwei PHP-Funktionen nötig. Einerseits `mysql_query` welche die eigentliche Anfrage ausführt. Sie Benötigt zwei Argumente, ein SQL-Statement und den Verbindungsaufbau zur Datenbank. In einem zweiten Schritt wird die Funktion `mysql_fetch_array` aufgerufen welche als Argument die Funktion `mysql_query` hat. `mysql_fetch_array` liefert den aktuellen Tuppel als Array zurück und `FALSE` wenn kein weiterer vorhanden ist. Mit der `while`-Schleife kann so jeder einzelne Tuppel verarbeitet werden.

#### 4.3.3 Erstellen einer Prüfung

In der [Abbildung 4.3-1] wird das HTML-Formular gezeigt welches für das Erstellen einer Prüfung notwendig ist. Das Darstellen der Prüfungsart wie auch der Unterlage führt zu ei-

ner Datenbankabfrage. Die Daten und die Teilnehmerzahl werden vom Administrator eingegeben.

#### 4.3.4 Kanu-Guide erfassen

Um einen neuen Kanu-Guide in der Datenbank erfassen zu können werden die persönlichen Angaben wie sie in [Abbildung 4.3-2] ersichtlich sind eingegeben. Die Auswahl der Sprache erfolgt mittels Radio-Buttons.

Fax: +41 43 222 40 17

### Inhalte Prüfung Guides

**Prüfungsart**

**Datum**

**Anmeldefrist**

**Teilnehmerzahl**

**Unterlage**

Ein Projekt von:

Christoph Kuert abmelden

Abb. 4.3-1: Screenshot: Prüfungserstellung

Fax: +41 43 222 40 17

### Inhalte Prüfung Guides

**Nachname**

**Vorname**

**Adresse**

**PLZ/Ort**

**Geburtsdatum**

**E-Mail**

**Telefon**

**Organisation**

**Sprache**  Deutsch  Französisch

Ein Projekt von:

Christoph Kuert abmelden

Abb. 4.3-2: Screenshot: Kanu-Guide erfassen

### 4.3.5 Anmelden für die Prüfung

Der Ablauf für die Anmeldung zu einer Prüfung wird in [Abbildung 3.2-1] als UML-Aktivitätsdiagramm erläutert.

Nachdem die Prüfung in der Datenbank erfasst wurde, kann sich ein angehender Kanu-Guide für eine Prüfung registrieren. Um sich für eine Prüfung anmelden zu können, darf die maximale Teilnehmerzahl noch nicht erreicht und die Anmeldefrist darf noch nicht verstrichen sein. Die Prüfungen welche diese zwei Kriterien erfüllen, werden als HTML-Select-Box angezeigt [Abbildung 4.3-3].

Abb. 4.3-3: Screenshot: Prüfung auswählen

Der angehende Kanu-Guide füllt die selben Felder wie der Administrator aus. Wenn die Anmeldung erfolgreich war, wird automatisch eine Bestätigung per E-Mail versandt. Ein E-Mail erhält der Kanu-Guide, das zweite die Geschäftsstelle des SKV.

Der Eintrag in die Datenbank ruft zwei unabhängige PHP-Elemente auf. Einerseits werden die persönlichen Angaben des Kanu-Guides erfasst und andererseits wird die Anmeldung eingetragen.

### 4.3.6 Prüfungsergebnisse eintragen

Nachdem die Prüfung stattgefunden hat, hat der Administrator die Möglichkeit, die Kanu-Guides in die Guideliste aufzunehmen. Er wählt dazu die Anmeldungen für die entsprechende Prüfung aus.

In der [Abbildung 4.3-4] ist ersichtlich wer die Prüfung bestanden hat und wer nicht. Durch einen einfachen klick auf „hat bestanden“ wird der Kanu-Guide in die Liste der Absolventen aufgenommen.

Hier hat der Administrator zusätzlich die Möglichkeit die Daten als CSV-Datei auf seinen eigenen Computer zu speichern und dort weiter zu verarbeiten (*Tabelle speichern*). Der Vorteil einer CSV-Datei ist, dass sie von allen Tabellenkalkulationsprogrammen gelesen werden kann.

Fax: +41 43 222 40 17

### Inhalte Prüfung Guides

**10.10.2009 Kanuguide 1: Canadier - Anmeldungen: 5**

GuidelD	Name	Vorname	Organisation	CPR	bestanden?
1	Kuert	Christoph	Uni Fribourg	absolviert	bestehen lassen
4	Müller	Fritz	MüllerMilch	absolviert	bestehen lassen
5	Müller	Paul	Kanu Müller	absolviert	hat bestanden
15	Muster	Thomas	Kanu Muster	absolviert	hat bestanden
16	Muster	Alfred	Kanu Muster	absolviert	bestehen lassen

Tabelle speichern

Ein Projekt von:





Abb. 4.3-4: Screenshot: Prüfungsergebnisse eintragen

#### 4.3.7 Ausgebildete Kanu-Guides anzeigen

Das Ausgeben der einzelnen Kanu-Guides nach Kategorie tönt erst einfach, doch durch die Flexibilität des Systems sind vier der acht Tabellen involviert. Es braucht die Prüfungsarten, die Prüfungen, die Kanu-Guides und die Absolventen.

Die [Abbildung 4.3-5] zeigt das selbe Bild wie [Abbildung 4.3-4] plus noch einen Kanu-Guide in der Kategorie *Kajak 1*.

Tel: +41 43 222 40 17  
Fax: +41 43 222 40 17

## Über uns Prüfung Guideliste

<b>Kanuguide 1: Kajak</b>			
Nachname	Vorname	Ort	Prüfungsdatum
Wyss	Reto	Luzern	12.12.2009



<b>Kanuguide 1: Canadier</b>			
Nachname	Vorname	Ort	Prüfungsdatum
Müller	Paul	Müllerigen	10.10.2009
Muster	Thomas	Muster	10.10.2009

**Kanuguide 2: Kajak**  
In dieser Kategorie gibt es noch keine Guides.

**Kanuguide 2: Canadier**  
In dieser Kategorie gibt es noch keine Guides.

Ein Projekt von:

deutsch - français

Abb. 4.3-5: Screenshot: ausgebildete Kanu-Guides

## 5. Schlusswort

Das Ziel dieser Arbeit war es, den gestiegenen Anforderungen seitens der Vereinigung Kanu-Guide Schweiz gerecht zu werden. Weiter sollte eine übersichtliche Dokumentation über das realisierte System resultieren welches eine zukünftige Weiterentwicklung vereinfacht.

Das überarbeitete System wird im März 2010 in Betrieb genommen. Die bisherige Zusammenarbeit mit Reto Wyss habe ich sehr geschätzt.

Es hat sich gezeigt, dass webbasierte Systeme einen grossen Programmieraufwand beinhalten. Sämtliche Manipulationen an der Datenbank müssen erstellt werden. Des Weiteren muss auch die gesamte graphische Benutzeroberfläche erstellt werden.

Ein Einsatz von einem Content Management System könnte bei einer sorgfältigen Implementierung Vorteile bringen. Viele grundlegende Funktionen sind bereits vorhanden. Weiter haben Open Source Projekte wie z.B. Drupal unzählige Module um die Funktionalität den eigenen Bedürfnissen anzupassen.

Um zusätzliche Flexibilität im Entwicklungsprozess zu gewinnen, liessen sich unter Umständen Agilen Methoden einsetzen.

Die konkrete Entwicklung sollte unbedingt auf dem Rechner erfolgen, auf welchem das System später auch eingesetzt wird. Es hat sich herausgestellt, dass nicht immer alle Einstellungen identisch sind und zu mühsamen Nachbesserungen am Quellcode führen kann.

Das System kann unter <http://kgs.christoph-kuert.ch> getestet werden. Um Zugang zum Backend zu erhalten, kann ein Benutzername und Passwort beim Autoren angefragt werden.

## 6. Literaturverzeichnis

[Meier 2007]

Meier Andreas: *Relationale und postrelationale Datenbanken*, 6. Auflage, Springer-Verlag, Berlin-Heidelberg, 2007.

[Miles, Hamilton 2006]

Miles Russ, Hamilton Kim: *Learning UML 2.0*, 1. Auflage, O'Reilly Media Inc., Sebastopol USA, 2006.

[Kemper, Eickler 2006]

Kemper Alfons, Eickler André : *Datenbanksysteme – Eine Einführung*, 6. Auflage, Oldenbourg Wissenschaftsverlag GmbH, München, 2006.

[MySQL 2009]

*MySQL 5.1 Referenzhandbuch*, <http://dev.mysql.com/doc/refman/5.1/de/index.html>, 2009, Zugriff am 04.09.2009

[PHP 2009]

*PHP-Handbuch - Manual*, <http://www.php.net/manual/de/index.php>, 2009, Zugriff am: 04.09.2009

[Tabaghoghi, Williams 2007]

Tabaghoghi Seyed M.M., Williams Hugh E.: *Learning MySQL*, 1. Auflage, O'Reilly Media Inc., Sebastopol USA, 2007.

[Zehnder 1998]

Zehnder Carl August : *Informationssysteme und Datenbanken*, 6. Auflage, vdf Hochschulverlag AG an der ETH, Zürich und B.G. Teubner, Stuttgart, 1998.

## Anhang

Die wichtigsten PHP-Elemente wie sie im Abschnitt 4.3 im Hintergrund ausgeführt werden, sind hier aufgeführt und erläutert.

### • Hochladen von Unterlagen

```
function unterlagenAdd($conn){

    $dokumentname = mysql_real_escape_string($_POST['dokumentname']);

    // Beginn des Dateiupload

    // Prüfen ob es keinen Fehler gibt und ob die Datei übergeben wurde

    if (!$_FILES["Unterlagen"]["name"] == ""){
        if ($_FILES["Unterlagen"]["error"] > 0) {
            $fileMessage = "Fehler {$_FILES["Unterlagen"]["error"]} beim Upload der Datei
            {$_FILES["Unterlagen"]["name"]}";
        }
        else {

            // Prüfen ob die Datei bereits existiert und falls nein, die Datei auf den Server laden

            if (file_exists("../exams.folder/dokumente/" . $_FILES["Unterlagen"]["name"])) {
                $fileMessage = "Die Datei {$_FILES["Unterlagen"]["name"]} existiert
                bereits.";
            }
            else {
                move_uploaded_file($_FILES["Unterlagen"]["tmp_name"],
                "../exams.folder/dokumente/" . $_FILES["Unterlagen"]["name"]);
                $fileMessage = "Upload der Datei {$_FILES["Unterlagen"]["name"]}
                erfolgreich.";
            }
        }
    }
    //Ende des Dateiupload

    // Die Angaben in die Datenbank eintragen

    $SQL = "      INSERT INTO  Pruefungsunterlage
                (      Unterlagenname,
                Unterlage,
                Sprache)
                VALUES      (      ".$dokumentname.", '
                ".$_FILES['Unterlagen']['name']."',
                ".$_POST['sprache']."");

    $result = mysql_query($SQL, $conn);

    // Wenn die Variable $result keinen Wert hat, dann ist ein Fehler aufgetreten. Entsprechende
    // Meldung ausgeben.
    if(!$result){
        $dbMessage = "Ein Fehler ist aufgetreten, versuchen Sie es erneut oder
        kontaktieren sie webmaster-at-kanuguideschweiz.ch";
    }
    else{
```

```

        $dbMessage = "Die Unterlage: {$dokumentname} wurde erfolgreich
        hinzugef&uuml;gt.";
    }
    return $dbMessage."<br />".$fileMessage;
}

```

Die PHP-Funktion *unterlagenAdd* transferiert eine Datei auf den Server und speichert die dazugehörigen Angaben in der Datenbank. Text welcher nach zwei „/“ folgt, ist Kommentar und gehört nicht zum eigentlichen Programm.

Mittels *mysql\_real\_escape\_string* werden spezielle Zeichen maskiert welche in der SQL-Abfrage zu einer Fehlermeldung führen können. Alle manuell eingegebenen Werte werden so überprüft.

Mit einer so genannten superglobalen Variable werden die Daten weiterübermittelt. Sie sind vom Datentyp Array. Der Name des Dokumentes wurde mit Hilfe der HTTP-Request-Methode POST übermittelt und kann mittels *\$\_POST['dokumentname']* wieder ausgegeben werden.

Die Angaben zum Dokument werden im Array *\$\_FILES ["Unterlagen"]* gespeichert. Mit der Funktion *file\_exists* prüft ob das Dokument bereits vorhanden ist. Mit *move\_uploaded\_file* wird das Dokument definitiv auf dem Server gespeichert. Ein vorhandenes, gleichnamiges Dokument würde ohne Warnung überschrieben.

Es folgt nun die SQL-Anfrage welche in der Variable *\$SQL* gespeichert wird. Die SQL-Anfrage wird zusammen mit dem Verbindungsaufbau *\$conn* von der zentralen Funktion *mysql\_query* verwendet. In der Datenbank wird eine Referenz zum Dokument gespeichert für eine spätere Wiederverwendung.

Zum Schluss wird mittels *return* eine Nachricht zum Resultat des Datenbankeintrages und zum Status des Dokumenten-Übertrages zurückgegeben.

## • Erstellen der Prüfung

```

function addExam($conn) {

    $Teilnehmerzahl = mysql_real_escape_string($_POST['Teilnehmerzahl']);
    $dasJahr = mysql_real_escape_string($_POST['dasJahr']);
    $derMonat = mysql_real_escape_string($_POST['derMonat']);
    $derTag = mysql_real_escape_string($_POST['derTag']);
    $fristJahr = mysql_real_escape_string($_POST['fristJahr']);
    $fristMonat = mysql_real_escape_string($_POST['fristMonat']);
    $fristTag = mysql_real_escape_string($_POST['fristTag']);

    //Datum umwandeln in String YYYY-MM-DD

    $Datum = $dasJahr."-".$derMonat."-".$derTag;
    $Anmeldefrist = $fristJahr."-".$fristMonat."-".$fristTag;

    // Die Angaben in die Datenbank eintragen

    $SQL = "        INSERT INTO Pruefung
                (        PruefungsartID,
                        Datum,
                        Anmeldefrist,
                        Teilnehmerzahl,
                        PruefungsunterlageID)
                VALUES
                (        "".$_POST['PruefungsartID'].",

```

```

        ".$Datum.",
        ".$Anmeldefrist.",
        ".$Teilnehmerzahl.",
        ".$_POST['PruefungsunterlagenID'].");

$result = mysql_query($SQL, $conn);

if(!$result){
    $dbMessage = "Ein Fehler ist aufgetreten, versuchen Sie es erneut oder
    kontaktieren sie webmaster-at-kanuguideschweiz.ch";
}
else{
    $dbMessage = "Die Pr&uuml;fung ".$_POST['Bezeichnung']." am
    ".$_POST["derTag"].".".$_POST["derMonat"]."."
    ".$_POST["dasJahr"]." wurde erfolgreich hinzugef&uuml;gt.";
}
return $dbMessage;
}
}

```

Beim erstellen einer Prüfung gibt es mehrere Felder die vom Administrator manuell ausgefüllt werden müssen. Damit MySQL mit den Angaben korrekt rechnen kann, werden die beiden Daten, Teilnehmerzahl und Anmeldefrist, entsprechend formatiert.

### • Anmelden für die Prüfung

Der Ablauf für die Anmeldung zu einer Prüfung wird in [Abbildung 3.2-1] als UML Aktivitätsdiagramm erläutert.

```

// Abfrage für alle Prüfungen für welche die Anmeldefrist noch nicht abgelaufen ist
$SQL = "
    SELECT      PruefungID,
               DATE_FORMAT(Pruefung.Datum,'%d.%m.%Y') as DatumFormatiert,
               Teilnehmerzahl,
               Beschreibung
    FROM        Pruefung,
               Pruefungsart
    WHERE       Anmeldefrist > CURRENT_DATE
    AND         Pruefung.PruefungsartID = Pruefungsart.PruefungsartID
    ORDER BY   Datum ASC";

$result = mysql_query($SQL, $conn);
$anz_rows = mysql_num_rows($result);
if ($anz_rows > 0) {
    echo "<option value=\"0\">
        - Pr&uuml;fung w&auml;hlen -
    </option>";
    while ($row = mysql_fetch_array($result)) {
        $SQL_Anmeldungen = "
            SELECT *
            FROM      Anmeldung
            WHERE     PruefungID = '{$row["PruefungID"]}";

        $result_Anmeldungen = mysql_query($SQL_Anmeldungen, $conn);
        $anz_Anmeldungen = mysql_num_rows($result_Anmeldungen);
        // Bei jeder Prüfung wird geprüft ob die Teilnehmerzahl erreicht ist oder nicht.
        if ($row["Teilnehmerzahl"] > $anz_Anmeldungen) {
            print "
                <option value=\"{$row["PruefungID"]}\">

```

```

        {$row["DatumFormatiert"]}: {$row["Beschreibung"]}
    </option>";
    // Es wird gezählt wie viele Prüfungen aufgeführt werden
    $i = $i + 1;
}

// Falls der Zähler $i kleiner als 1 ist sind alle Prüfungen ausgebucht
if ($i < "1") {
    echo "    <option value=\"0\">
        - Pr&uuml;fungen ausgebucht -
    </option>";
}
}
}
else {
    echo "
    <option value=\"0\">
        - keine Pr&uuml;fung in der Datenbank -
    </option>";
}
}

```

Nachdem die Prüfung in der Datenbank erfasst wurde, kann sich ein angehender Kanu-Guide für eine Prüfung registrieren. Die Prüfung darf die maximale Teilnehmerzahl noch nicht erreicht haben und die Anmeldefrist darf noch nicht verstrichen sein.

In der SQL-Abfrage werden nur diejenigen Einträge ausgewählt noch in der Anmeldefrist sind (*WHERE Anmeldefrist > CURRENT\_DATE*) Das Prüfungsdatum wird mit *DATE\_FORMAT* in die uns gewohnte Schreibweise TT.MM.YYYY formatiert. *mysql\_num\_rows(\$result)* berechnet wie viele Zeilen den gewünschten Kriterien entsprechen. Falls dieser Wert nicht grösser als 0 ist, ist es nicht möglich sich für eine Prüfung anzumelden. Für jede Prüfung wird anschliessend berechnet wie viele Anmeldungen bereits vorhanden sind. Falls alle ausgebucht wären, würde dies entsprechend angezeigt.

Die eigentliche Ausgabe erfolgt über *mysql\_fetch\_array()*. Die Resultate werden mittels einer while Schleife durchlaufen und als Array ausgegeben. Aus dem Code resultiert eine HTML-Select-Box wie sie in [Abbildung 4.3-3] zu sehen ist um eine Prüfung auszuwählen.

Nachdem der Kanu-Guide seine persönlichen Angaben gemacht hat folgt der eigentliche Eintrag in die Datenbank. Es beinhaltet zwei unabhängige Teile. Das Erfassen der persönlichen Angaben und die Anmeldung für die Prüfung.

```

function newGuide($conn){

    // Eingegebene Daten maskieren, damit sie ohne Probleme in die Datenbank übertragen
    // werden können

    $Nachname = mysql_real_escape_string($_POST['Nachname']);
    $Vorname = mysql_real_escape_string($_POST['Vorname']);
    $Adresse = mysql_real_escape_string($_POST['Adresse']);
    $PLZ = mysql_real_escape_string($_POST['PLZ']);
    $Ort = mysql_real_escape_string($_POST['Ort']);
    $gebTag = mysql_real_escape_string($_POST['gebTag']);
    $gebMonat = mysql_real_escape_string($_POST['gebMonat']);
    $gebJahr = mysql_real_escape_string($_POST['gebJahr']);
    $EMail = mysql_real_escape_string($_POST['EMail']);
    $Telefon = mysql_real_escape_string($_POST['Telefon']);
    $Organisation = mysql_real_escape_string($_POST['Organisation']);
}

```

```

//Datum umwandeln in String YYYY-MM-DD

$gebDatum = $_POST["gebJahr"]."-".$_POST["gebMonat"]."-".$_POST["gebTag"];

// Die Angaben in die Datenbank eintragen

$SQL = "  INSERT INTO Kanuguide
          (    Vorname,
              Nachname,
              Adresse,
              PLZ,
              Ort,
              Geburtsdatum,
              EMail,
              Telefon,
              Organisation,
              Sprache)
          VALUES
          (    ".$Vorname.",
              ".$Nachname.",
              ".$Adresse.",
              ".$PLZ.",
              ".$Ort.",
              ".$gebDatum.",
              ".$EMail.",
              ".$Telefon.",
              ".$Organisation.",
              ".$_POST["sprache"].");

$result = mysql_query($SQL, $conn);
if(!$result){
    $dbMessage = "Ein Fehler ist aufgetreten, versuchen Sie es erneut oder kontaktieren
                  Sie webmaster-at-kanuguideschweiz.ch";
}
else{
    $dbMessage = "Registrierung erfolgreich: ".$Nachname."&nbsp;".
                  $Vorname."&nbsp;von&nbsp;".$Organisation;
}
return $dbMessage;
}

```

Die Registrierung der persönlichen Daten des Kanu-Guides enthält keine neuen Code-Elemente.

```

function examRegistration($conn){

    // Falls sich ein neuer Guide registriert hat er noch keine KanuguideID und diese muss aus
    // der DB ausgelesen werden.

    if ($_POST['KanuguideID'] == "") {
        $SQL_KanuguideID = "
            SELECT    KanuguideID
            FROM      Kanuguide
            WHERE     Vorname = '{$_POST['Vorname']}'
            AND       Nachname = '{$_POST['Nachname']}'
            AND       Organisation = '{$_POST['Organisation']}'";

        $result_KanuguideID = mysql_query($SQL_KanuguideID, $conn);
    }
}

```

```

        while ($row = mysql_fetch_array($result_KanuguidID)) {
            $KanuguidID = $row["KanuguidID"];
        }
    }
    else {
        $KanuguidID = $_POST['KanuguidID'];
    }

    $SQL = "  INSERT INTO      Anmeldung
            (KanuguidID,
             PruefungID,
             CPR)

            VALUES

            ('.$KanuguidID.',
             '$_POST['PruefungID'].',
             '$_POST['CPR'].')";

    $result = mysql_query($SQL, $conn);
    if(!$result){
        $dbMessage = "P&uuml;fungsameldung fehlgeschlagen.";
    }
    else{
        $dbMessage = "P&uuml;fungsameldung erfolgreich.";
    }
    return $dbMessage;
}

```

Falls sich der Kanu-Guide neu anmeldet, ist seine KanuguidID noch nicht bekannt. Da jedoch die Funktion *newGuide()* zeitlich vor *examRegistration()* durchgeführt wird, kann man mit Vorname, Name und Organisation die KanuguidID aus der Datenbank ausgeben. Anschliessend wird die Anmeldung gespeichert.

Die Benachrichtigung per E-Mail hat nicht direkt etwas mit der Datenbank zu tun, es wird verzichtet auf diesen Teil konkret einzugehen.

```

function examPassed($conn){

    $SQL = "  INSERT INTO      Absolvent
            (KanuguidID,
             PruefungID)

            VALUES

            ('.$_GET['KanuguidID'].',
             '$_GET['PruefungID'].')";

    $result = mysql_query($SQL, $conn);
    if(!$result){
        $dbMessage = "Ein Fehler ist aufgetreten, versuchen Sie es erneut oder kontaktieren
                     sie webmaster-at-kanuguideschweiz.ch";
    }
    else{
        $dbMessage = "Der neue Guide wurde erfolgreich hinzugef&uuml;gt.";
    }
    return $dbMessage;
}

```

Das Eintragen der Prüfungsergebnisse durch *examPassed()* erfordert dann lediglich noch zwei Parameter, die KanuguidID und die PrüfungID.

### • Ausgebildete Kanu-Guides Anzeigen

Das Ausgeben der einzelnen Kanu-Guides nach Kategorie tnt einfach, es sind aber zwei Datenbankabfragen ntig und insgesamt vier Tabellen involviert.

```

$SQL_Art = "SELECT * FROM Pruefungsart";
$result_Art = mysql_query($SQL_Art, $conn);
while ($row_Art = mysql_fetch_array($result_Art)) {

    echo "<b>{$row_Art["Beschreibung"]} </b>";

    $SQL = "    SELECT      Nachname,
                    Vorname,
                    Ort,
                    DATE_FORMAT(Pruefung.Datum,'%d.%m.%Y') as Datum
            FROM      Pruefung,
                    Pruefungsart,
                    Absolvent,
                    Kanuguide
            WHERE     Pruefungsart.PruefungsartID = {$row_Art["PruefungsartID"]}
            AND      Pruefung.PruefungsartID = Pruefungsart.PruefungsartID
            AND      Pruefung.PruefungID = Absolvent.PruefungID
            AND      Kanuguide.KanuguideID = Absolvent.KanuguideID
            ORDER BY Nachname ASC ";

    $result = mysql_query($SQL, $conn);
    $anz_rows = mysql_num_rows($result);

    echo "<table border=\"0\" width=\"400\" cellspacing=\"5\" cellpadding=\"5\">";
    if ($anz_rows > 0) {
        if ($akt_lang == "fr") {
            print "
                <tr>
                    <td>nom</td>
                    <td>preacute;nom</td>
                    <td>lieu</td>
                    <td>date de l'examen</td>
                </tr>";
        }
        else {
            print "
                <tr>
                    <td>Nachname</td>
                    <td>Vorname</td>
                    <td>Ort</td>
                    <td>Pruuml;fungsdatum</td>
                </tr>";
        }
        while ($row = mysql_fetch_array($result)) {
            print "
                <tr>
                    <td>{$row["Nachname"]}</td>
                    <td>{$row["Vorname"]}</td>
                    <td>{$row["Ort"]}</td>
                    <td>{$row["Datum"]}</td>
                </tr>";
        }
    }
    else {

```

```
    if ($akt_lang == "fr") {
        echo "<tr><td>Il n'y a pas de guides dans cette categorie.</td></tr>";
    }
    else {
        echo "<tr><td>In dieser Kategorie gibt es noch keine Guides.</td></tr>";
    }
}
echo "</table><br /><br />";
}
```

Um die Kanu-Guides nach Kategorie anzuzeigen werden zwei separate Schleifen durchlaufen. Als erstes werden die verschiedenen Prüfungsarten durchlaufen. Für jede Art werden dann die benötigten Angaben aus den Tabellen Pruefung, Pruefungsart, Absolvent und Kanuguide zusammengeführt. In diesem Teil ist auch noch die Variable *\$akt\_lang* sichtbar. Sie definiert in welcher Sprache die Webseite dem Besucher präsentiert wird.