# BACHELORARBEIT 2

## Handbuch zur Erstellung eines Geoinformationssystems basierend auf Joomla! und der Google Maps-API

durchgeführt am Bachelorstudiengang Informationstechnik und System–Management Fachhochschule Salzburg GmbH

> vorgelegt von: Klaus Höpfl



Studiengangsleiter: Betreuer: FH-Prof. DI Dr. Gerhard Jöchtl Dipl.-Ing.(FH) Markus Eisenmann

Salzburg, Mai 2010

## Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich die vorliegende Bachelorarbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und alle aus ungedruckten Quellen, gedruckter Literatur oder aus dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte gemäß den Richtlinien wissenschaftlicher Arbeiten zitiert bzw. mit genauer Quellenangabe kenntlich gemacht habe(n). Diese Arbeit wurde in gleicher oder ähnlicher Form weder im In- noch im Ausland in irgendeiner Form als Prüfungsarbeit vorgelegt und stimmt mit der durch die BegutachterInnen beurteilten Arbeit überein.

Ort, Datum

Name und Unterschrift der/s Studierenden

## Danksagung

Ich bedanke mich bei allen, die mich in meinem Studium und bei der Erstellung dieser Arbeit unterstützt haben. Ohne deren Hilfe wäre diese Arbeit nicht entstanden.

#### Vielen Dank!

Im Besonderen gilt mein aufrichtigster Dank meiner Freundin *Juliane Mondry*. Sie hatte wirklich tiefstes Verständnis für die letzten Jahre aufgebracht. Zu jeder Zeit hatte sie meine Launen und Stimmungen sehr tapfer ertragen und mir stets geholfen und mich unterstützt, damit ich meine Konzentration auf das Studium und diese Arbeit lenken konnte.

#### Vielen lieben und herzlichen Dank, Juliane!

Mein tiefster Dank gilt meinen *Eltern*, die mich all die Jahre unterstützt haben. Ohne Euch hätte ich es nie so weit geschafft. Das Studium wäre ohne Euch nicht möglich gewesen.

#### Vielen lieben und herzlichen Dank, für Euer grenzenloses Vertrauen und Eure Geduld!

Ich bedanke mich bei meinen beiden Kollegen Anton Hennecke und Clemens Langegger des Studiengangs Informationssystem und System-Management (ITS) der Fachhochschule Salzburg. Zusammen mit ihnen wurde der Grundstock dieser Arbeit gelegt. Beide hatten immer ein offenes Ohr für Probleme und Fragen.

#### Freundlichsten Dank, Anton und Clemens!

Dem gesamten *Team des Studiengangs* ITS möchte ich Dank und meine Anerkennung für das vermittelte Wissen aussprechen. Für die beispiellose Bereitschaft, jederzeit für Fragen und Probleme aller Art, Hilfestellungen und Diskussionen, Zeit und ein offenes Ohr gefunden zu haben.

#### Vielen lieben Dank!

Mein respektvoller Dank gilt *Herrn Univ.-Doz. FH-Prof. Dr. Karl Entacher*, für die Möglichkeit, diese sehr interessante und herausfordernde Arbeit zu erstellen. Meinen persönlichen Dank möchte ich für die Chance und gewonnene Erfahrung aussprechen, innerhalb des Erstellungszeitraumes dieser Arbeit, die theoretischen und praktischen Inhalte an einem Gymnasium in Salzburg zu unterrichten. Seine Tür stand zu jeder Zeit für mich offen. Er versteht es, mit Wort, Witz und Charme, seine Studenten zu motivieren!

#### Vielen lieben Dank, Herr Entacher!

Ein großes Dankeschön möchte ich *Herrn Dipl.-Ing.(FH) Markus Eisenmann* aussprechen. Für die Betreuung und langwierigen Korrekturen dieser Arbeit. Besonders für Verbesserungsvorschläge, sachdienlichen Hinweise und hilfreichen Tipps zur Gestaltung und Abrundung dieser Bachelor-Arbeit.

#### Besten Dank, Herr Eisenmann!

## Anmerkung

Aus Gründen der besseren Lesbarkeit werden manche Begriffe nur in ihrer männlichen Form verwendet. Selbstverständlich sind die weiblichen Formen im Sinne einer impliziten Schreibweise mitgedacht.

#### Kurzzusammenfassung

Diese Bachelorarbeit beschreibt ein Konzept zur Erstellung und Bedienung eines Geoinformationssystems. Die Arbeit ist in vier Bereiche gegliedert. Zunächst werden die theoretischen Grundlagen zu geocodierten Daten, Content Management Systemen und Web-Technologien verdeutlicht. Weiterhin erläutern Anwendungsprogramme und Funktionen die Erstellung eines Geoinformationssystems. Das dritte Kapitel zeigt die Vorgehensweise zur Einführung des Geoinformationssystems an einer Schule auf. Der Anhang dieses Handbuchs stellt eine Anleitung zur Installation einer Web-basierten Geo-Plattform dar.

### Abstract

This bachelor thesis deals with the concept to establish and handle a geographical information system. This thesis is divided into four chapters. First of all the basic information are explained to geographically coded data, content management systems and web technology. Furthermore programs and functions are explaining the generation of a geographically based system. Moreover the course of action is illustrated to adopt the system in the form of given lessons at a grammar school. The appendixes of this thesis are providing an instructions manual to set up a web based geographical platform.

## Inhaltsverzeichnis

1	Einleitung									
2	Gru	ndlager	1	2						
	2.1	1 Geocodierte Daten								
	2.2	2 Google Maps								
	2.3	Conte	nt Management System							
	2.4	Webse	rver	10						
	2.5	Web-7	Technologien	11						
3	Anw	Anwendungsprogramme 19								
	3.1	Ubunt	u	19						
	3.2	Jooml	a!	23						
	3.3	Web-A	Applikation	27						
		3.3.1	Überblick	27						
		3.3.2	Erste Schritte	28						
	3.4	Projek	tbezogene Funktionen	30						
		3.4.1	Marker hinzufügen	30						
		3.4.2	Marker anzeigen	33						
		3.4.3	Routen hinzufügen	37						
		3.4.4	Routen anzeigen	39						
		3.4.5	Modul Kategorien auswählen	42						
		3.4.6	Modul Kategorien bearbeiten	44						
		3.4.7	GPX-Routen	47						
	3.5	Portal	pilität	49						
		3.5.1	Projektdateien	49						
		3.5.2	Gesamte Webseite	50						
4	Lehi	Lehrveranstaltung								
	4.1	Erste	Unterrichtseinheit	52						
	4.2	Zweite	e Unterrichtseinheit	52						
	4.3	Dritte	Unterrichtseinheit	53						
	4.4	Vierte	Unterrichtseinheit	53						
5	Zusammenfassung und Ausblick									
6	Abkürzungsverzeichnis									
7	Abbildungsverzeichnis									
8	Literaturverzeichnis									

9 Anhang				
	9.1	XAMPP	66	
	9.2	Ubuntu Server 8.04 LTS	70	
	9.3	Joomla! 1.5	86	
10	Que	Icode Geoinformationssystem	97	
	10.1	googleMapFiles/APIkey	97	
	10.2	$googleMapFiles/db\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\$	97	
	10.3	$googleMapFiles/getMarkers \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	97	
	10.4	$googleMapFiles/setMarkers \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	104	
	10.5	$googleMapFiles/getRoutes\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\$	127	
	10.6	$googleMapFiles/setRoutes\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\ .\$	136	
	10.7	$googleMapFiles/getGPX file \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	143	
	10.8	$googleMapFiles/getKMLfile \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	177	
	10.9	$modules/mod\_edu \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	179	
	10.10	$Mod_edu_cat$	182	

## 1 Einleitung

Das Schlagwort "WEB2.0" kursiert seit geraumer Zeit in den Medien. Zu unscharf sind die Definitionen und auf vielfältigste Weise wird "WEB2.0" interpretiert. Die Motivation wurde geweckt, die wahre Bedeutung zu erfahren. Mit den Begriffen, Aussagen und den Technologien vertraut zu werden. Das umgangssprachlich bezeichnete "Mitmach-Web" und Online-Plattformen kennen zu lernen, die dem *World Wide Web* eine neue Dynamik verleihen sollen. Zu ergründen, wo der Unterschied zwischen dem "alten" und dem "neuen" Internet liegt. Die Bedeutung für den Einzelnen, den Web-Anwender, Designer und dem Programmierer.

Obwohl wir uns an die Gegenwärtigkeit des Internets gewöhnt haben, schreitet die Entwicklung rasch voran. Das Web wird interaktiver, dynamischer und benutzerfreundlicher. Grund genug, aktuelle Designtrends, Techniken, sowie einschlägige Werkzeuge zur Entwicklung und Gestaltung näher unter die Lupe zu nehmen. Werkzeuge verwenden, welche Online-Plattformen frei zur Verfügung stellen, damit Programmierer diese nutzen und erweitern können. Vorhandene Daten kostenfrei einsetzen und neue Ideen verwirklichen ist eine Herausforderung des "WEB2.0".

Diese Bachelorarbeit geht einen Schritt weiter. Es werden nicht nur die vorhandenen Technologien und Werkzeuge genutzt, mit deren Hilfe ein Geoinformationssystems erstellt wird, sondern diese Arbeit soll zudem ein Handbuch darstellen, um die einzelnen Schritte nachvollziehen zu können. Die Zielsetzung dieser Arbeit lag auf der Erstellung einer informativen Web-basierten Plattform zur Präsentation von Bildungsstätten.

Es werden die notwendigen Grundlagen zum besseren Verständnis der verwendeten Technologien erläutert. Des Weiteren wird anhand von Beispielen die Installation eines Webservers, der Einsatz eines Inhaltverwaltungssystems und die dazu notwendigen Funktionen und Module aufgezeigt. Dem Open-Source Gedanken, freies Wissen und kostenfreie Programme zu verwenden, darauf aufzubauen und wiederum offen zu legen, wurde mit diesem Handbuch nachgekommen.

Darüber hinaus lag der größte Teil der Motivation darin, sich theoretisches Wissen und praktische Fähigkeiten anzueignen, um es in Form einer Lehrveranstaltung an Schüler zu vermitteln.

### 2 Grundlagen

Für das Verständnis dieser Arbeit, der Erstellung eines Geoinformationssystems, ist es notwendig, die Grundkenntnisse der verwendeten Techniken und Anwendungsprogramme und die Gründe für ihren Einsatz zu vermitteln.

#### 2.1 Geocodierte Daten

Geodaten sind in der Regel in digitaler Form vorliegende Daten mit direktem oder indirektem Bezug zu einem bestimmten geographischen Gebiet oder Standort. Geodaten sind als Bestandteil eines Informationssystems zu sehen. Für die Beschreibung können zum Beispiel Distanzen und Winkel verwendet werden. In der Regel wird auf Koordinaten übergegangen und die Wahl eines Bezugssystems und der Metrik festgelegt. Für die geometrische Darstellung werden Vektor- oder Rasterdaten verwendet. Sie dienen zur geographischen Ortsbestimmung auf der Erdoberfläche. Ähnlich der Ebene eines Koordinatensystems ist die Oberfläche der Erde mit einem gedachten Gradnetz definiert. Das Bezugssystem stellt der Äquator der Erde und der historische Nullmeridian dar, welcher durch das Flamsteed House in Greenwich verläuft [42]. Das Gradnetz weißt 180 Breitenkreise parallel zur Äquatorebene auf, deren Durchmesser sich nach Norden und Süden zu den Polen verjüngt. Der Äquator stellt die 0°-Marke dar, von dieser aus beginnend die Breitenkreise nach Norden und Süden um jeweils 90°zunehmen. Die 360 Längenkreise stehen senkrecht zu den Breitenkreisen und verlaufen alle vom Nord- zum Südpol der Erde. Vom Nullmeridian aus erfolgt die Einteilung nach Westen und Osten in jeweils 180°-Schritten. Die Lage einer Stadt, einer Straße oder eines Wohnhauses kann mit der Angabe der Längen- und Breitengradwerte exakt bestimmt werden.

Die International Organization for Standardization (ISO) definiert in der Norm ISO 19113 allgemeine Qualitätsparameter für den Einsatz von Geoinformationssystemen [36]. Zu den allgemeinen Qualitätsparametern zählen die Vollständigkeit einzelner Attribute oder Beziehungen zu Objekten, die logische Konsistenz und Widerspruchsfreiheit der konzeptionellen und physikalischen Datenstruktur, sowie eine durchschnittliche Positions- und thematische Genauigkeit und Korrektheit der Attribute.

Google stellt einen Internet dienst zur Visualisierung von Geodaten bereit, der als Ergebnis eine Karte im Raster datenformat<sup>1</sup> liefert.

 $<sup>^1\</sup>mathrm{Bilddateien}$  im PNG-Bildformat

#### 2.2 Google Maps

Google Maps ist ein von Google bereitgestellter kostenloser Dienst, der die Möglichkeit bietet nach Orte, Straßen und verschiedensten Objekten zu suchen und auf einer Karte darzustellen [27]. Google Maps stellt darüber hinaus zusätzliche Funktionen eines Routenplaners dar. In Abbildung 1 wird eine typische Benutzeroberfläche der Anwendung in einem Browserfenster gezeigt.



Abbildung 1: Google Maps Benutzeroberfläche

Dieses Beispiel zeigt, dass die Kartenanwendung diverse Funktionen zur Navigation und Bearbeitung des Kartenmaterials zur Verfügung stellt. Die Schaltflächen in der rechten oberen Ecke ermöglichen ein Wechseln zwischen Karten-, Satelliten- und Geländedarstellung. Bei der Satellitendarstellung wird das Kartenmaterial durch Satelliten- oder Luftaufnahmen ersetzt. Die Geländedarstellung *Gelände* lässt einen Überblick mit Angaben von Höhenmeterlinien erkennen, wie nachfolgende Abbildung 2 zeigt.



Abbildung 2: Google Maps Geländedarstellung

In der linken oberen Ecke befindet sich die Navigationsschaltfläche zum Bewegen der Karte. Zusätzlich besteht die Möglichkeit, die angrenzende Umgebung der aktuellen Kartenansicht durch Klicken mit der linken Maustaste, sowie dem Ziehen der Maus in die gewünschte Richtung anzuzeigen. Am linken Rand befindet sich außerdem das Zoom-Werkzeug. Mithilfe dieses Tools kann der aktuelle Kartenausschnitt vergrößert oder verkleinert werden, wobei das Betätigen des Mausrades dieselbe Funktion erfüllt. Die Miniaturkarte - zu sehen in der rechten unteren Ecke stellt ein zusätzliches Navigationselement dar. Sie visualisiert eine kleine Übersicht über den aktuellen Kartenausschnitt. Des Weiteren ist es möglich ein zusätzliches Eingabefeld hinzuzufügen, wodurch in der Karte nach Orte, Straßen oder Objekten gesucht werden kann. Die Darstellung der Suchergebnisse erfolgt über Markierungen auf der Karte.

#### Google Maps-API

Ein Application Programming Interface (API) ist eine Schnittstelle zur Anwendungsprogrammierung. Diese ermöglicht anderen Anwendungen eine Anbindung an das Softwaresystem. Dies geschieht im Falle von Google mit Hilfe der Scriptsprache JavaScript. Ein Application Programming Interface (API) kann grundsätzlich mit einem Objekt verglichen werden, über dessen Methoden und Attribute Programmierer auf externe Daten Zugriff haben. Generell erleichtern Programmierschnittstellen komplexe Datenbestände, ohne aufwändig selbsterstellte Methoden abzurufen. Die Offenheit der Daten liefert den Webentwicklern neue technologische Möglichkeiten. So lässt sich der vorhandene Kartendienst von Google nutzen, wie im Kapitel 3 Anwendungsprogramme, unter 3.4 Projektbezogene Funktionen, erläutert wird. Die Funktionsbibliothek dieses APIs ist umfangreich, wobei deren gezielter Einsatz eine strukturierte Entwicklerdokumentation voraussetzt. Eine Dokumentation wird von Google angeboten [31].

#### Funktionsumfang

Diese beinhaltet Referenzen zu den Funktionen und beschreibt deren Verwendung. Um das Interface zu nutzen, ist ein Schlüssel notwendig. Der API-Key ist kostenlos bei Google erhältlich. Dieser ermöglicht die Kommunikation zwischen der eigenen Webseite und dem Google Maps-Server. Sobald der Schlüssel eingebunden ist, kann das API ohne Einschränkungen verwendet werden.

Eine weitere Funktion bietet das Hinzufügen und Bearbeiten von Markierungspunkten, sogenannten Markern. Diese können auf verschiedene Weise auf dem Kartenmaterial platziert werden. Eine Möglichkeit ist beispielsweise durch einfaches Klicken auf einen bestimmten Punkt auf der Karte. Durch die Eingabe eines Ortes oder einer Straße kann ebenfalls eine Markierung gesetzt werden. In beiden Fällen werden dabei die Koordinaten des zu platzierenden Markers ermittelt und an die Anwendung weitergeleitet, woraufhin eine Markierung an diesem Punkt dargestellt wird.

Die Verwendung von Geodaten in Form von Koordinaten stellt ebenfalls eine Möglichkeit zur Platzierung solcher Markierungen dar. Dazu müssen die Koordinaten im Dezimalformat vorliegen. Zum Beispiel ist die Gradangabe 47°43' 28" in die Dezimalschreibweise 47,70777778 umzurechnen. Zusätzlich bietet das Maps-API die Möglichkeit Informationen zu den dargestellten Markierungen anzuzeigen. Diese können ebenfalls individuell angepasst werden und durch verschiedene Ereignisse, wie zum Beispiel ein Klick auf einen Marker, ein- oder ausgeblendet werden.

Neben der Handhabung von Markierungen, können mithilfe des Maps-API auch Routen zwischen zwei oder mehreren Punkten dargestellt werden. Dies setzt voraus, dass ein Start- und Zielpunkt gegeben sind. Die Verbindung der beiden Punkte errechnet die Funktion und stellt die Route anhand der kürzesten Strecke auf der Karte dar, wie nachfolgende Abbildung 3 beispielhaft zeigt.

Eine Funktion zum Auslesen von *Keyhole Markup Language* (KML) Dateien ist ebenfalls vorhanden. KML ist eine Auszeichnungssprache um Geodaten zu beschreiben. Diese wird hauptsächlich von Google selbst verwendet, beispielsweise für die Anwendung Google Earth<sup>2</sup>. Bei KML-Dateien handelt es sich um ein komprimiertes Archiv. Diese können geokodierte Informationen in Form von Koordinaten beinhalten und einzelne Punkte oder ganze Routen beschreiben. Außerdem können zusätzliche Informationen, wie etwa Farbe, Linienstärke oder auch ein beschreibender Text zu den Punkten und Routen darin abgespeichert werden. Ein Beispiel einer KML-Fahrrad-Route um den Gaisberg bei Salzburg ist in Abbildung 3 dargestellt.



Abbildung 3: Geländekartenausschnitt KML-Route Gaisberg

Eine Alternative zu Google Maps ist OpenStreetMap [47]. Diese Anwendung ist eine freie Software mit Kartenmaterial. Der Quellcode zum Programm kann von der offiziellen Webseite bezogen werden [46]. Da zu dieser Software weder eine Entwicklerdokumentation noch ein API vorhanden ist, wurde in dieser Arbeit auf die Verwendung von Google Maps zurückgegriffen. Im folgenden Abschnitt werden einige Grundlagen zu Content Management Systemen beschrieben.

<sup>&</sup>lt;sup>2</sup>http://earth.google.com/intl/de/

#### 2.3 Content Management System

Allgemein betrachtet wird unter dem Begriff eines *Content Management System* (CMS) eine Software verstanden, die zur formatierten Darstellung von Inhalten dient. Zu den Inhalten zählen nicht nur Artikel, sondern auch Medientypen wie Bilder, Video und Audio. Die einheitliche Verwaltung und Aufbereitung der Daten, zur strukturierten Visualisierung und Bearbeitung der Inhalte, ist die zentrale Aufgabe eines CMS.

#### Kriterien

Ein CMS zu verwenden, unterstützt vor allem die Benutzer, welche keine Datenbank- oder sonstige Informatik-Kenntnisse besitzen. Weder für die Installation, noch für einen produktiven Einsatz sind tiefgehende oder umfangreiche Erfahrungen in Programmiersprachen erforderlich.

Entwickler schätzen demgegenüber den Einsatz dieser Systeme, insbesondere im Hinblick auf eine beträchtliche Zeitersparnis. Langwierige Entwicklungsprozesse zum Aufbau einer gegliederten Menüstruktur entfallen. Ein seitenübergreifender Suchalgorithmus ist vorhanden. Ebenso eine sichere Zugangsverwaltung und eine übersichtliche Organisation des Mitgliederbereiches.

Mittels benutzerfreundlicher Installationsroutinen, zusammen mit informativen Hilfetexten, werden angehende Autoren weitestgehend durch ein CMS unterstützt. Es stellt sich demnach noch die Frage, von welcher Kategorie ein *Content Management System* ausgewählt werden sollte.

#### Kategorien

Alle verfügbaren Content Management Systeme lassen sich wie folgt klassifizieren:

- Content Management System (CMS)
- Web Content Management System (WCMS)
- Gebührenpflichtige CMS, WCMS
- Open Source CMS, WCMS

Insbesondere in Medienunternehmen des Buch-, Zeitungs-, Zeitschriftenverlages werden Inhaltsverwaltungssysteme eingesetzt. Diese Form eines CMS dient lediglich der firmeninternen Erstellung, Verwaltung, Speicherung sowie Archivierung von Textmaterial. Die Software ist speziell auf diesen Anwendungsbereich zugeschnitten und optimiert. Veröffentlichungen finden meistens nur in ausgedruckter Form statt. Ein Beispiel für die Kategorie CMS ist Atex<sup>3</sup>.

Im Gegensatz dazu dienen web-basierte CMS, um die erstellten Medieninhalte online zu publizieren, sowie von Anfang an eine Präsentation im Internet zu schaffen.

Zu den WCMS zählen überdies die sogenannten Weblog-Systeme, zu deren bekanntesten Vertretern WordPress<sup>4</sup> zählt. Das Wesentliche beschränkt sich auf die Veröffentlichung privater Inhalte.

<sup>&</sup>lt;sup>3</sup>http://www.atex.com/

<sup>&</sup>lt;sup>4</sup>http://wordpress.org/

Mehr oder weniger komplexe Shop-Systeme, beispielsweise Amazon, bei welchem eine zusätzliche Komponente des Online-Bezahl-Systems hinzukommt. Dokumentenverwaltungssysteme, wie zum Beispiel SharePoint, sind ebenfalls den WCMS zuzuordnen. Über eine Internetverbindung können auf einem SharePoint-Server liegende Dateien gleichzeitig von mehreren Personen bearbeitet und eingesehen werden. Beispiele für die Kategorie WCMS sind ACCURA<sup>5</sup>, Drupal<sup>6</sup>, Typo3<sup>7</sup>, Joomla!<sup>8</sup>, WordPress und SharePoint<sup>9</sup>.

Ein wichtiges Kriterium bei der Auswahl stellt das Lizenzmodell, beziehungsweise der Kostenfaktor dar. Es gilt eine Entscheidung für den Einsatz eines gebührenpflichtigen oder kostenlos erhältlichen Inhaltverwaltungssystem zu treffen.

#### Auswahl

Welche Kategorie der CMS den Anforderungen des geplanten Anwendungsumfeldes gerecht wird, muss zuerst abgeklärt werden.

Erfüllt ein kostenpflichtiges CMS allen Ansprüchen des Einsatzzweckes, bietet es nachfolgende Vorteile: Die Installation und Einrichtung des gesamten Systems erübrigt sich. Wartung und Pflege des Onlineauftrittes entfallen. Einzig die Bereitstellung von Inhalten ist mit Eigenleistung und Mehrkosten verbunden. Zusätzlich kann ein Rechtsanspruch bestehen, sollten sich Sicherheitslücken in dem erworbenen System befinden und zum eigenen Schaden verwendet worden sein.

Der Nachteil von fehlerhaftem Programmcode besteht bei einem kostenlosen CMS nicht. Je bekannter und von umso mehr Personen genutzt, desto eher werden schadhafte Komponenten erkannt, sowie Fehlfunktionen korrigiert. Zusätzliche Kosten fallen nicht an. Ermöglicht wird dies durch quelloffene Software (Open Source) und einer großen Entwicklergemeinde. Unentgeltliche Hilfestellungen bei Fragen und Problemen werden meist in Foren des Open Source CMS angeboten.

Ein weiterer Punkt, welcher vor der Anschaffung eines CMS bedacht werden sollte, ist eine einfache Erweiterbarkeit. Zum Beispiel kann das vorhandene System im Bedarfsfall mit einer Onlineshop-Komponente ergänzt, Börseninformationen hinzugefügt oder der Internetauftritt mit einer einfachen Umfrage ausgestattet werden.

Klarheit darüber schafft die CMS-Matrix [52]. Auf dieser werden alle Stärken und Schwächen zu einem bestimmten Produkt aufgezeigt. Angefangen von relevanten Funktionen zur Sicherheit, Benutzerfreundlichkeit, über geschwindigkeitsbezogene Aspekte, bis hin zu bereits implementierten Zusatzmodulen. Besonders interessant ist der direkte, übersichtliche Vergleich der verschiedenen

<sup>&</sup>lt;sup>5</sup>http://www.yoursitecms.com/

<sup>&</sup>lt;sup>6</sup>http://www.drupal.de/

<sup>&</sup>lt;sup>7</sup>http://typo3.org/

<sup>&</sup>lt;sup>8</sup>http://www.joomla.de/

<sup>&</sup>lt;sup>9</sup>http://office.microsoft.com/de-de/sharepointserver/default.aspx

	Joomla! 1.5.10 🛛 🗙	<u>TYPO3 4.3.3</u> 🔀	TYPOlight 2.6.3 🔀	WordPress 2.2.1 🔀
Last Updated	1/11/2009	4/20/2010	8/20/2009	7/25/2007
System Requirements	Joomla! 1.5.10	TYPO3 4.3.3	TYPOlight 2.6.3	WordPress 2.2.1
Application Server	CGI	Apache	CGI	Apache
Approximate Cost	Free	Free	Free	Free
Database	MySQL	MySQL	MySQL	MySQL
License	Open Source	Open Source	Open Source	Open Source
Operating System	Platform Independent	Platform Independen	t Platform Independent	Platform Independent
Programming Language	PHP	PHP	PHP	PHP
Root Access	No	No	No	No
Shell Access	No	No	No	No
Web Server	Apache	Apache	Any	Apache

Content Management System. Nachfolgende Abbildung 4 stellt einen Auszug der Vergleichsmöglichkeiten mit der CMS-Matrix dar.

Abbildung 4: CMS-Vergleich mit CMS-Matrix

Das Funktionsprinzip ist bei sämtlichen Content Management Systemen identisch und spielt in Bezug auf die Entscheidung für ein bestimmtes Softwareprodukt keine Rolle.

#### Funktionsweise

Wird beispielsweise eine Seite über den Browser angefordert, erfolgt dies von einem Computer, auch als Client bezeichnet. Ein Server im Internet nimmt die Anfrage des Clients entgegen und leitet diese an den Serverdienst, den Web-Server, weiter. Webserverdienste sind beispielsweise der Apache-Server, von der Organisation Apache Software Foundation und der *Internet Information Service* (IIS) der Firma Microsoft.

Ein Webserver empfängt die Daten und startet das CMS zur weiteren Verarbeitung. Die Anfrage kann hinsichtlich der Berechtigung des Nutzers überprüft werden. Besitzt der User die erforderlichen Rechte, wird die Anforderung verarbeitet und eine Antwort von den einzelnen Modulen des CMS zusammengestellt. CMS-Komponenten holen die Inhalte aus einer Datenbank und setzen diese in eine vorgegebene HTML-Vorlage ein. Erweitert wird die Anforderung mit Menübereichen, Umfragen oder gegebenenfalls Werbung. Das resultierende Ergebnis wird vom CMS als HTML-Code an den Webserver weitergeleitet und an den Client zurückgesendet. Somit bedient diese Form einer Web-Applikation die von einem Kunden angeforderten Inhalte und stellt sie in seinem Browserfenster dar.

#### 2.4 Webserver

Der Begriff Webserver wird zumeist auf eine physische Hardware bezogen. Als Server wird ein Computer bezeichnet, welcher anderen Rechnern Dateien zur Verfügung stellt. Hierbei kann es sich um jede Form von Dokumenten oder Musik-, Bild- und Video-Dateien handeln. Die Möglichkeit des Anbietens<sup>10</sup> übernimmt und steuert jedoch eine Software. Dieser Software-Dienst ist der tatsächliche Server. Als Webserver wird diejenige Server-Software bezeichnet, welche Dateien und Inhalte über das Internet oder Intranet zur Verfügung stellt. Dies können statische *HTML*-Seiten, Textdokumente, aber auch dynamische Datenbank-basierte Webseiten sein.

Zum Austausch der Informationen zwischen einem Client und einem Webserver kommt neben dem *File Transfer Protocol* (FTP) hauptsächlich das *Hypertext Transfer Protocol* (HTTP) zum Einsatz [60], [61]. Durch das HTTP-Protokoll wird bei einer Benutzeranfrage einer Webseite eine *Transmission Control Protocol* (TCP)-Verbindung mit dem Webserver aufgebaut. Dies geschieht zumeist über eine *Uniform Resource Locator* (URL), zum Beispiel http://www.beispiel.net/ inhalt.html. Der Server nimmt die Anfrage entgegen und bearbeitet diese. Das Ergebnis der Ressource /inhalt.html wird an den Client zurückgesendet und im Browser dargestellt. Die TCP-Verbindung wird wieder beendet.

Die gängigste Webserver-Software für HTTP ist der auf Open Source basierende Apache HTTP Server [2]. Im Kapitel 3.1 Ubuntu wird genauer auf das Softwarepaket Apache eingegangen.

 $<sup>^{10} \</sup>ddot{\mathrm{U}}\mathrm{b}\mathrm{e}\mathrm{rsetzung}$ aus dem Englischen to serve: dienen, bedienen, anbieten

#### 2.5 Web-Technologien

Tim Berners-Lee [37], der Erfinder des *World Wide Web*, verfolgte am Europäischen Kernforschungszentrum (CERN) eine Idee. Sein Ziel war es, anhand digitaler Verweise für ein effizienteres Auffinden von Dokumenten zu sorgen. Das Problem, an dessen Lösung sich Tim Berners-Lee im Jahre 1989 versuchte, war bereits bekannt. Universitäten und wissenschaftliche Einrichtungen suchten seit Jahren nach Technologien, um Ordnung in dem globalen Chaos des Informationsaustausches zu erlangen. Das heißt, die verschiedensten lokal entstandenen Hypertext-Systeme miteinander zu verknüpfen und zu vereinheitlichen. Dabei wollte Berners-Lee noch einen Schritt weiter gehen. Die Inhalte sollten frei und weltweit verfügbar sein, außerdem jedem Interessenten die Möglichkeit bieten, an dieser neuen Technologie teilzunehmen. Anhand einer einfachen Beschreibungssprache, eindeutigen Adressenmustern zum Auffinden von Dokumenten, sowie der Entwicklung eines Übertragungsprotokolls, wurde der Informationsaustausch und die Datenverknüpfung für alle Anwender vereinheitlicht.

#### HTML

Die von Berners-Lee anfänglich einfach konstruierte Seitenbeschreibungssprache Hypertext Mark-Up Language (HTML) wurde im Laufe der Jahre weiterentwickelt und liegt zum Erstellungszeitpunkt dieser Arbeit in der Version 4.01 vor. Die Version HTML 5 befindet sich in der Entwicklung. Es wurde eine Sprache für das World Wide Web geschaffen, welche Texte, Bilder und Hyperlinks in Webbrowsern darstellt. HTML stellt zusätzliche Angaben bereit. Dazu zählen etwa der Seitentitel, Angaben über den Autor, Schlagwörter für Suchmaschinen und layoutspezifische Verweise.

HTML ist seither die Basissprache für das Strukturieren von beispielsweise Überschriften, Absätzen, Listen und Tabellen bei Webseiten. Des Weiteren ist HTML ein definierter Standard des *W3C Consortium* [56], dem Standardisierungsgremium für Webtechnologien. Listing 1 zeigt ein HTML-Beispiel zum Anzeigen eines Titels, einer Überschrift und Text.

```
<html>
<html>
<html>
<html>
<title> Der Titel der Webseite </title>
</head>
<body>
<h1>Eine Überschrift</h1>
Und der entsprechende Text dazu...
</body>
</html>
```

Listing 1: Seitenbeschreibungssprache HTML

#### $\mathbf{CSS}$

*Cascading Style Sheets* (CSS) ist eine Sprache, um einzelne Textbausteine oder ein gesamtes Dokument zu gestalten. Mit der Positionierung einzelner Bereiche lassen sich Navigations-, Inhaltsund Informationsspalten von Webseiten entwerfen. CSS schafft eine übersichtlichere Strukturierung von HTML-Code, indem es die optische Gestaltung von der Hypertext-Auszeichnungssprache trennt. CSS ist folglich ein in HTML integrierbarer Mechanismus zum Layouten und Formatieren von Webseiteninhalten [57]. Listing 2 erweitert das Beispiel aus Listing 1 um eine eingebettete CSS-Formatangabe in einem HTML-Dokument.

```
<html>
1
       <head>
2
       <title> Der Titel der Webseite </title>
3
       <style type="text/css">
4
       h1 {
\mathbf{5}
            color: green;
6
        }
7
        span.rotertext {
8
            color: red;
9
        }
10
       </style>
11
       </head>
12
       <body>
13
       <h1>Eine Überschrift</h1>
14
       <span class="rotertext">Und der entsprechende Text dazu...</span>
15
       </body>
16
  </html>
17
```

Listing 2: HTML-Dokument mit CSS-Definitionen

In den Zeilen 4 bis 11 aus Listing 2 werden die CSS-Layoutoptionen für die Überschrift und den Text festgelegt. Dadurch wird die Überschrift in der Farbe grün, der Text in der Farbe rot dargestellt.

#### $\mathbf{XML}$

Die rein auf Text basierte Auszeichnungssprache *Extensible Markup Language* (XML) wurde eingeführt, um hierarchisch gegliederte und komplexe Datensätze in einer einheitlichen Form zwischen verschiedenen Anwendungen auszutauschen. XML-Datensätze können unabhängig von verwendeten Programmiersprachen einheitlich zusammengesetzt, übertragen, empfangen, ausgelesen und verarbeitet werden.

Ein Beispiel einer XML-Datei zeigt das nachfolgende Listing 3.

```
<?xml version="1.0" encoding="utf-8" ?>
<install type="module" version="1.0.0">
<name>Who is Online</name>
```

Listing 3: XML-Datei des Joomla!-Modul "Who's Online"

#### Semantisches Web

Die Grundideen von XML brachte die neue Erweiterung des *World Wide Web* hervor. Unter dem Begriff des *Semantischen Web* ist die Möglichkeit zu verstehen, Inhalte so zu strukturieren, dass diese für Menschen und Maschinen gleichartig zu interpretieren und analysieren sind. Ein nach semantischen Regeln ablaufender Suchalgorithmus findet Beziehungen zu einem Namen, wie zum Beispiel geschäftliche Kontakte oder Adressen von Schulen. Dabei werden diese namensbezogenen Daten nicht mittels Durchsuchen von Schlüsselwörtern in Textblöcken "erraten", sondern können gemäß ihrer vorher definierten Bedeutung interpretiert und zugeordnet werden. Ermöglicht wird dies durch zusätzliche semantische Attribute im Quellcode von Webinhalten, Mikroformate genannt. Mit deren Hilfe suchen nicht mehr Anwender auf mehreren Webseiten nach in Wohnortnähe befindlichen Bildungsstätten, sondern eine Online-Plattform liefert eine Liste aller Schulen in einem bestimmten Umkreis.

Ein mögliches Mikroformat einer Fachhochschule stellt das Listing 4 dar.

```
<address class="vcard">
<span class="org">FH-Salzburg</span><br>
<span class="adrgory">Fachhochschule</span><br>
<span class="adr">
<span class="adr">
<span class="street-address">Urstein Süd 1</span><br>
<span class="country-name">AT</span>-
<span class="postal-code">5412</span>
<span class="locality">Puch/Salzburg</span><br>
</span>
<span class="tel_cell">+43-(0)50-2211-0</span><br>
```

```
<a class="url" href="http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">http://www.fh-salzburg.ac.at/">
```

Listing 4: Mikroformat-Beispiel für die FH-Salzburg

#### XHTML

Parallel zu HTML entstand aus den Vorteilen von XML und den Mikroformaten heraus die *Ex*tensible Hypertext Markup Language (XHTML), eine erweiterbare und Text-basierte Auszeichnungssprache zur Strukturierung von semantischen Inhalten. Die Verwendung von XHTML-Dokumenten erlaubt die Einbettung anderer XML-basierender Dokumente. XHTML ist XMLgerechtes HTML. Auf die Unterschiede zwischen HTML und XHTML näher einzugehen, würde den Rahmen dieser Arbeit sprengen<sup>11</sup>.

Das Listing 5 zeigt einen kurzen (X)HTML-Quellcode der Abweichungen beider standardisierten Auszeichnungssprachen [56].

```
<?xml version = "1.0" encoding = "ISO - 8859 - 1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHIML 1.0 Transitional//EN" "xhtml1-
   transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
       <meta http-equiv="content-type" content="text/html; charset=ISO
           -8859-1" />
       <title>Wenn HTML zu XHTML wird</title>
   </head>
   <body>
       <h1><a name="start" id="start">Wenn HTML zu XHTML wird</a></h1>
       Dann mü ssen alle Elemente mit Inhalt ein End-Tag haben.
           >
       Leere Elemente<br />mkuuml; ssen einen Schrkauml; gstrich am Ende
            haben .
       <hr noshade="noshade" />
       Leere Attribute erhalten ihren eigenen Namen als Wert
           zugewiesen .
       <a href="#start">Verweise zu Ankern</a> springen zum
           Zielelement
        aufgrund des id-Attributs, nicht das name-Attributs.
```

<sup>&</sup>lt;sup>11</sup>http://de.selfhtml.org/html/xhtml/unterschiede.htm

Listing 5: Unterschied HTML zu XHTML [41]

*HTML 5* soll die nebeneinander existierenden beschreibenden Sprachen XML, XHTML und *HTML 4.01* ersetzen und zu einem gemeinsamen Standard vereinen [59].

#### PHP

Hypertext Preprocessor (PHP) ist eine Server-seitige Skriptsprache. Sie findet Verwendung in der Programmierung von Webseiten und Webanwendungen. Die Seitenbeschreibungssprache HTMList nur für die Präsentation von statischen Inhalten geeignet. Die einzigen dynamischen Elemente waren sogenannte Hyperlinks<sup>12</sup>, mit denen sich andere statische Seiten aufrufen ließen. Erweiterungen wurden vorgenommen, um dieser Einschränkung zu entgehen. Flash<sup>13</sup> und Java-Applets<sup>14</sup> ergänzten HTML-Seiten Client-seitig um interaktive Funktionen. Die nahezu vollständige Programmiersprache *PHP* wurde entwickelt, um dem fehlenden Funktionsumfang auf der Server-Seite gerecht zu werden. Nahezu vollständig, da *PHP* auf *HTML* angewiesen ist.

Die Ausgabe eines *PHP*-Scripts wird Server-seitig interpretiert und in den *HTML*-Code eingebunden. Das Ergebnis wird an den Client zurückgesendet und durch den Client-Browser dargestellt. Dynamisch bedeutet im Zusammenhang mit *PHP*, dass erst zum Zugriffszeitpunkt eines Clients der Inhalt der Webseite erstellt wird. Listing 6 zeigt einen in *HTML* eingebundenen *PHP*-Quellcode.

```
<html>
<head>
<title> PHP in HTML </title>
</head>
<body>
Dies ist ein statischer Text.
<?php
echo ("Dieser Text wird erst bei Anforderung dieses HTML-
Dokuments durch den PHP-Interpreter eingefügt.");
?>
</body>
</html>
```

Listing 6: Eingebetteter PHP-Code in einem HTML-Dokument

<sup>&</sup>lt;sup>12</sup>Hyperlink: Verknüpfung, Verbindung, Verweis, Link

<sup>&</sup>lt;sup>13</sup>http://www.adobe.com/products/flashplayer/

<sup>&</sup>lt;sup>14</sup>http://java.sun.com/applets/

Heutzutage bilden Datenbanken die Hauptkomponente jeder Web-basierten Applikation. Im Gegensatz zu *HTML* bietet *PHP* die Werkzeuge zur Interaktion mit *Datenbankmanagementsystemen* (DBMS) [48]. Listing 7 zeigt ein Beispiel, mittels *PHP* auf Inhalte einer Datenbanktabelle zuzugreifen.

```
<html>
    <head>
        <title> Einfache Datenbankabfrage </title>
    </head>
    <body>
    <?php
    $link = mysql connect("datenbankserver", "benutzer", "passwort") or
       die ("Keine Verbindung möglich!");
        mysql select db("datenbankname") or die("Auswahl der Datenbank
            fehlgeschlagen");
        $query = "SELECT COUNT(*) FROM benutzer";
        $result = mysql query($query) or die (mysql error());
        $anzahl = mysql fetch assoc($result);
        foreach($anzahl as $key => $wert) {
            echo("Anzahl registrierte Benutzer: ".$wert."\n"); // Ausgabe
                der Tabelleninhalte
        }
        mysql close($link);
    ?>
    </body>
</html>
```

Listing 7: Datenbankzugriff mit PHP

#### JavaScript

Eine weitere Sprache ist JavaScript. Diese Programmiersprache läuft im Browser ab und muss keine besonderen Server-Voraussetzungen erfüllen. Mit Hilfe von JavaScript gelang es, die eingeschränkten Möglichkeiten von HTML zu erweitern und zu optimieren. Zum Beispiel können bereits Client-seitig Anmeldeformulare dahingehend überprüft werden, ob ein Benutzer alle Angaben der Formularfelder ausgefüllt hat. Eine Übertragung der eingegebenen Daten an den Webserver findet somit erst statt, wenn das Formular vollständig ausgefüllt ist. Die korrekte Eingabe einer E-Mail-Adresse in einem Formular wird zum Beispiel Client-seitig mit JavaScript validiert, wie durch das Listing 8 implementiert.

```
<html>
<head>
<title>E-Mail-Formular Validierung</title>
<script type="text/javascript">
```

```
function checkForm(){
                 var strFehler = '';
                 if (!validEmail(document.forms[0].email.value)) {
                     strFehler += "Dies ist keine gültige E-Mail-Adresse!\n
                         ";
                 }
                 if (strFehler.length > 0) {
                     alert ("Festgestellte Probleme: \n\ + \ strFehler);
                     return (false);
                 }
             }
             function validEmail(email){
                 var strReg = " ([a-zA-Z0-9])+@(([a-zA-Z0-9])+))
                    +([a-zA-Z0-9]{2,4})+";
                 var regex = new \operatorname{RegExp}(\operatorname{strReg});
                 return (regex.test(email));
             }
        </\operatorname{script}>
    </head>
    <body>
        <form action='emailtest.php' onsubmit='return checkForm()'>
            <h1>Newsletter-Anmeldung</h1>
            Email-Adresse <input name='email'/>
            <input type='submit' value='Absenden'/>
        </form>
    </body>
</html>
```

Listing 8: E-Mail-Formular Validierung mit JavaScript

Aus der Scriptsprache JavaScript [62] hat sich das Konzept AJAX entwickelt.

## AJAX

Asynchronous JavaScript and XML (AJAX) ermöglicht bestimmte Inhaltsbereiche einer Webseite asynchron zu aktualisieren. Angeforderte Seiteninformationen werden nachgeladen und nach Anfrage des Nutzers in die Seitenstruktur eingebettet. Der Vorteil gegenüber dem synchronen Seitenaufbau ist, dass vom Webserver nicht die gesamte Webseite neu generiert und übertragen wird, sondern nur die angeforderten Daten übermittelt und dargestellt werden. AJAX reagiert auf Interaktionen vom Benutzer, kann Anfragen an den Server stellen, dessen Antworten in der Seitendarstellung entsprechend anpassen und auf erneute Eingaben reagieren. Dies alles geschieht im "Hintergrund". Ein einfaches Beispiel dieser Client-Server-Prozesse demonstriert die Suchmaschine Google, welche noch nicht vollständig eingegebene Suchbegriffe vorschlägt und zu vervollständigen versucht. Die AJAX-Architektur basiert auf der Kombination von:

- (X)HTML für standardkonforme Seitenstrukturen
- XML zum Nachladen, Austausch und Abändern von lokal und extern gespeicherten Daten
- Asynchrone Datenübermittlung zwischen Client- und Server-Seite.

Listing 9 stellt ein kurzes Beispiel von AJAX-Funktionen dar.

```
1 <script language="JavaScript">
  var request = new XMLHttpRequest();
2
   request.open("GET", "file.xml", true);
3
   request.onreadystatechange = function() {
4
     if (request.readyState == 4) {
5
       alert(request.responseText);
6
     }
7
  }
8
  request.send(null);
9
10 </ script>
```

Listing 9: Inhalte einer Datei mit AJAX laden

Mithilfe des nativen Browserobjektes XMLHttpRequest ist es möglich, die in Zeile 3 angegebene Datei file.xml zu laden. Der Parameter true gibt eine asynchrone Übertragung vor. Die Scriptausführung läuft weiter und die HTTP-Abfrage wird im Hintergrund ausgeführt. Das Ergebnis onreadystatechange wird immer ausgelöst, wenn sich der Zustand des XMLHttpRequest-Objekts ändert. Dies geschieht zum Beispiel bei der Initialisierung und beim Verbindungsaufbau zur Datei. Die Eigenschaft readyState == 4 in Zeile 5 gibt an, dass die alert-Anweisung erst ausgeführt werden soll, wenn die Resource file.xml komplett geladen ist. Ein Dialog-Fenster der alert-Methode gibt den Inhalt der Datei im Browser aus.

### 3 Anwendungsprogramme

In diesem Kapitel wird auf die notwendigen Komponenten zur Erstellung des Geoinformationssystems eingegangen. Das Entscheidungskriterium der Auswahl eines Webserver-Betriebssystems wird erläutert und die benötigten Webserver-Dienste aufgezeigt. Die Grundlage des Informationssystems bildet ein *Content Management System*. Mit Hilfe von Programmbeispielen wird die Verwendung eines CMS zusammen mit den projektbezogenen Funktionen veranschaulicht. Die Funktionen Marker hinzufügen/anzeigen, Routen hinzufügen/anzeigen, sowie die GPX-Routen basieren auf der API von Google Maps.

#### 3.1 Ubuntu

Die Webseite *Distro Watch*<sup>15</sup> hat sich der Aufgabe verschrieben, einen Überblick über Linux-Distributionen zu schaffen. Angehenden Benutzern des Betriebssystems Linux werden die verschiedensten Distributionen vorgestellt. Dabei erwähnt *Distro Watch* nicht nur die Vorteile, sondern geht ebenso auf Nachteile der Distributionen ein. Des Weiteren wird eine Liste der zehn führenden Linux-Distributionen geführt.

Im Rahmen dieser Arbeit wurde Ubuntu, das auf dem ebenfalls freien Betriebssystem Debian basiert, für die Installation eines Webservers ausgewählt [17]. Ubuntu wird von DistroWatch als eine leicht benutzbare Distributionen geführt, welches durchaus mit proprietären Systemen mithalten kann. Betriebssystemversionen von Ubuntu haben feststehende Veröffentlichungstermine und Unterstützungsdauer. Hinter dem Projekt Ubuntu steht die Firma Canonical Ltd<sup>16</sup>. Diese stellt auf der Webseite für die Ubuntu-Server Edition die genauen Zeiträume der von Distro-Watch erwähnten feststehenden Termine und Dauer der Unterstützung dar [11]. Canonical Ltd stellt laut eigenen Angaben alle zwei Jahre eine Long Term Support (LTS) Veröffentlichung zur Verfügung. Diese speziellen LTS-Distributionen werden fünf Jahre lang mit Updates unterstützt. Dies schließt sowohl Updates zur Sicherheit, als auch für die Stabilität und Robustheit der jeweiligen LTS-Version mit ein. Nach Ablauf der fünf Jahre verspricht Canonical Ltd einen nahtlosen Übergang für die nächste Ubuntu LTS-Version [11]. Abbildung 5 zeigt den Lebenszyklus und Supportdauer von Ubuntu.

Ein weiteres Entscheidungskriterium für das Server-Betriebssystem *Ubuntu 8.04 LTS* sind die umfangreichen Dokumentationen. Diese werden sowohl von Benutzern [21], als auch von *Canonical Ltd* bereitgestellt [13]. Für jede *Ubuntu*-Distribution wird eine eigene klar strukturierte Übersicht der verfügbaren Installationspakete (Packages) geführt [20].

<sup>&</sup>lt;sup>15</sup>http://distrowatch.com/

<sup>&</sup>lt;sup>16</sup>http://www.canonical.com/



Abbildung 5: Lebenszyklen und Supportdauer für Ubuntu Server Editionen [11]

#### SSH

Zum einen bezeichnet Secure Shell (SSH) ein Netzwerkprotokoll, zum anderen entsprechende Programme, welche auf sichere Art und Weise eine Netzwerkverbindung mit einem entfernten Computer herstellen. SSH ermöglicht, sich an dem Ubuntu-Webservers anzumelden und die Kommandozeile des Servers auf einen lokalen Rechner darzustellen. SSH bietet gegenüber einer Telecommunication Network (Telnet)-Verbindung eine verschlüsselte Verbindung. Über diese sichere SSH-Verbindung kann der Webserver administriert werden. Es können beispielsweise Konfigurationsdateien bearbeitet oder Updates der Packages durchgeführt werden, ohne vor Ort am Server zu sein. Für Ubuntu wird das Package OpenSSH Server bereitgestellt [14]. Die Installationsroutine von Ubuntu bietet die Möglichkeit, einen Server mit SSH aufzusetzen (siehe Anhang Ubuntu Server 8.04 LTS > Softwareauswahl). Darüber hinaus sind sowohl die Desktop-, als auch die Server-Distributionen von Ubuntu kostenlos erhältlich [18].

Im Anhang dieser Arbeit, unter dem Kapitel *Ubuntu Server 8.04 LTS*, werden die einzelnen Installations- und Konfigurationsschritte für den Betrieb eines Webservers aufgezeigt.

#### LAMP

Das Akronym LAMP und steht für die Programmkombination *Linux, Apache, MySQL* und *PHP* [5]. Die Softwarekomponenten (*Apache, MySQL* und *PHP*) werden benötigt, um dynamische Web-Seiten und -Anwendungen zu entwickeln. Sie bilden die Kernsoftware, die ein Web-Server benötigt.

Ein *Linux Apache MySQL PHP* (LAMP)-System ist nicht von einem Linux-System abhängig. Das Betriebssystem *Windows* eignet sich ebenfalls für die Aufgabe eines Webservers und wird als *Windows Apache MySQL PHP* (WAMP) bezeichnet. Bei dem Akronym XAMP steht der Buchstabe "X" für eine betriebssystemunabhängige Bezeichnung.

Die Architektur Apache, MySQL, sowie PHP werden auf dem Server installiert. Die genaue Vorgehensweise ist im Anhang unter dem Kapitel Ubuntu Server 8.04 LTS verdeutlicht. Als lokaler Test-Webserver eignet sich (Betriebssystem) Apache MySQL PHP PERL (XAMPP), siehe Anhang Kapitel XAMPP.

#### Apache

Der Begriff "Webserver" wird oft im Zusammenhang mit einer reinen Hardware, einem Computer, verstanden. Streng genommen ist ein "Webserver" jedoch ein Dienst, siehe Kapitel 2.4 Webserver. Programmmodule sorgen für die Kommunikation zwischen Browser und der Webserver-Software. *Apache* [2] ist ein bekanntes Webserver-Softwareprodukt und hält laut *Netcraft* [43] einen derzeitigen Marktanteil von über 50% aller eingesetzter Webserver-Dienste, wie Abbildung 6. Der *Apache HTTP* Server-Dienst ist eine frei verfügbare Software [2].



Abbildung 6: Marktanteil Apache [43]

Auf einem *Apache*-Webserver können mehrere Webangebot laufen. *Apache* unterstützt das namensbasierte virtuelle Hosting. Somit ist es möglich, auf einem Computer eine Vielzahl von Webseiten zu betreiben. Jedes Webseitenangebot ist unter seiner vorgesehenen URL erreichbar [1].

#### MySQL

Das relationale Datenbankverwaltungssystem MySQL ist als Open-Source-Software verfügbar. Structured Query Language (SQL) ist eine strukturierte Abfragesprache von Tabelleninhalten einer Datenbank (DB). Ein DB System setzt sich aus einer Verwaltungssoftware, dem Datenbankmanagementsystem (DBMS) und den zu verwaltenden Daten der DB zusammen.

Bei Datenspeicherungen von Web-Services wird MySQL häufig eingesetzt. Mehrere Tabellen können in einer DB angelegt werden. Wiederum mehrere Datenbanken können von einem Datenbank-Management-System verwaltet werden. Somit kann eine Vielzahl von Webseiten auf die MySQL-Datenbank zugreifen. Die maximale Tabellengröße wird im Prinzip nur durch das Betriebssystem limitiert [49].

Das *Ubuntu*-Betriebssystem bildet neben dem *Apache*-Webserver, der DB-*MySQL* und der Scriptsprache *PHP* die Grundlage dynamischer Webauftritte, wie zum Beispiel dem CMS *Joomla!*.

#### 3.2 Joomla!

Joomla! ist ein web-basiertes Open-Source CMS und steht unter der GNU Public Licence, eine Lizenz speziell für freie Software [25]. Das CMS ist voll dynamisch und wurde in PHP 5 entwickelt. Als Datenbank wird *MySQL* verwendet, wobei sich zusätzliche Datenbankschnittstellen bereits in Entwicklung befinden. Ursprünglich ist *Joomla!* aus dem ebenfalls web-basierten Open-Source CMS *Mambo* entstanden. Da ein Großteil der Mambo Entwickler das Projekt verließen, diese das System jedoch weiterentwickeln wollten, entstand aus dem Quellcode der damaligen Mambo Version *Joomla!* 1.0. Für die aktuelle Version 1.5 des CMS wurde die Code-Basis umgeschrieben. Dadurch erreichten die Entwickler eine erhebliche Steigerung der Leistung, Sicherheit, Flexibilität und Erweiterbarkeit. Vor allem zeichnet sich das *Content Management System* durch seine hohe Flexibilität und Erweiterbarkeit aus. Aufgrund der Größe der vorhandenen Community sind bereits viele ausgereifte Erweiterungen für nahezu alle erdenklichen Einsatzbereiche vorhanden. Zusätzlich geht die Entwicklung beziehungsweise die Überarbeitung und Verbesserung der Erweiterungen dementsprechend schnell vonstatten. Außerdem ist *Joomla!* modular und logisch aufgebaut. Dies verkürzt die Einarbeitungsphase und erhöht die Übersichtlichkeit trotz des großen Funktionsumfangs [23].

#### Installation und Verwendung

Für die Installation von *Joomla!* ist es notwendig die zur Verfügung gestellten Dateien in einen beliebigen Ordner auf den Webserver zu kopieren. Danach muss dieser Ordner im Browser geöffnet werden, worauf das Einrichten des Systems beginnt. Dabei wird der Vorgang Schritt für Schritt erklärt. Es muss lediglich auf die korrekte Eingabe der erforderlichen Daten, wie zum Beispiel für die Datenbankanbindung, geachtet werden. Ist das Setup abgeschlossen erscheint eine Aufforderung aus Sicherheitsgründen den Ordner *Installation* vom Server zu löschen.

Daraufhin kann Joomla! verwendet werden. Bei der Verwendung des Systems fällt die grundlegende Aufteilung in Front- und Back-End auf. Das sogenannte Front-End beschreibt die Benutzeransicht der Webseite so, wie diese im Browser angezeigt wird, welche prinzipiell den öffentlichen Bereich des Systems darstellt. Das Back-End ist der administrative Bereich des CMS, welcher nur für berechtige Personen zugänglich ist. In diesem Bereich können alle administrativen Tätigkeiten, wie zum Beispiel die Benutzerverwaltung oder das Erstellen eines Artikels erfolgen.

#### Alternative Content Management Systeme

Einige Alternativen zu Joomla! sind beispielsweise die ebenfalls freien Content Management Systeme TYPO3, TYPOlight und WordPress. Aufgrund der folgenden Gründe wurde für diese Arbeit auf die Verwendung von Joomla! gesetzt. TYPO3 weist zwar eine mindestens gleich hohe Flexibilität wie Joomla! auf, ist jedoch für den Einsatzbereich dieser Arbeit viel zu umfangreich. Da das CMS keinen logischen Aufbau, wie zum Beispiel Joomla!, aufweist, ist aufgrund der hohen Komplexität des Systems eine viel längere Einarbeitungszeit nötig. Außerdem besitzt TYPO3 sehr hohe Hardware-Anforderungen. Dies ist auf den großen Funktionsumfang, sowie die Vielschichtigkeit zurückzuführen [52, 23]. TYPOlight hat nichts, wie der Name vermuten lässt, mit TYPO3 gemeinsam, sondern ist eine komplette Eigenentwicklung. Im Vergleich zu *Joomla!* weist dieses CMS eine geringere Flexibilität und Erweiterbarkeit auf. Zusätzlich wird die Standardversion mit relativ wenigen Funktionen ausgeliefert. Da diverse Funktionen nachinstalliert werden müssen, erhöht dies den Mehraufwand bezüglich Installation und Wartung [52]. WordPress weist ebenfalls eine niedrigere Flexibilität und Erweiterbarkeit als *Joomla!* auf. Außerdem ist dieses System eher auf die Verwaltung von Weblogs oder Nachrichtenseiten ausgelegt.

#### Template

Um das von Joomla! verwendete Front-End zu individualisieren, stellt das Content Management System die Verwendung sogenannter Templates zur Verfügung. Diese sind Vorlagen, die es ermöglichen das Aussehen des Front-Ends je nach Bedarf individuell anzupassen. Da bereits viele fertige und kostenlose Templates für Joomla! zur Verfügung stehen, können diese ohne Probleme verwendet und auch angepasst werden. Außerdem besteht die Möglichkeit komplette Vorlagen selbst zu implementieren.

Bevor ein eigenes Template erstellt wird, ist es notwendig den späteren Einsatzbereich und Verwendungszweck der Webseite zu beachten. Außerdem muss entschieden werden, welche *Joomla!* Module notwendig sind und wo sich diese jeweils auf der Webseite befinden sollen. Danach kann die Grundstruktur des Templates mithilfe von sogenannten Division-Elementen - kurz DIV - implementiert werden. Durch die Verwendung dieser Elemente erfolgt eine Aufteilung des Templates in verschiedene Bereiche. Der Einsatz von DIV-Elementen geschieht folgendermaßen:

Listing 10: Verwendung von Division-Elementen

Das Code-Segment in Listing 10 definiert den Bereich, in dem später beim Aufruf der Webseite die Inhalte, wie zum Beispiel Artikel oder Ähnliches, zur Verfügung stehen sollen. Mit demselben Verfahren werden sämtliche Bereiche, wie zum Beispiel Menü, Kopf oder Fuß definiert. Um dem so erstellten Konstrukt das gewünschte Aussehen zu verleihen ist CSS notwendig. Dabei werden in einer CSS-Datei die nötigen Parameter, beispielsweise Breite, Höhe, Position oder Hintergrund gewählt. Im Listing 11 ist die Formatierung eines DIV-Elements anhand des in Code-Beispiel 10 erstellten Bereichs dargestellt.

```
div#content {
    background-color: #FFFFF; // Hintergrundfarbe Weiss
    -moz-border-radius: 5px; // Runde Ecken
    overflow: hidden; // Überlauf verstecken
    min-height: 580px; // Mindesthöhe
}
```

#### Listing 11: Formatieren eines Division-Elements

Alle anderen Bereiche werden nach Bedarf auf dieselbe Weise formatiert. Nachdem das Template vollständig formatiert ist, muss die Platzierung der *Joomla!* Module definiert werden. Dazu gibt es vordefinierte Code-Segmente, welche *Joomla!* automatisch erkennt und durch das jeweilige Modul ersetzt. Ein Beispiel für die Verwendung dieser Code-Segmente stellt Code-Beispiel 12 dar.

Listing 12: Einbinden der Joomla!-Module

Nachdem das Template fertiggestellt ist, muss dieses noch in den dafür vorgesehen Ordner in der Joomla! Dateistruktur kopiert und anschließend aktiviert werden.

#### Module/Plugins

Obwohl Joomla! in der Standardausführung schon sehr viele Funktionen mit sich bringt, haben die Entwickler großen Wert darauf gelegt das Content Management System so einfach wie möglich erweitern zu können. Dies wird unter anderem durch den modularen Aufbau von Joomla! erreicht. So besteht die Möglichkeit einzelne Module und Plugins zu aktivieren oder zu deaktivieren beziehungsweise zu installieren oder zu deinstallieren. Außerdem ist die Erstellung eigener Module und Plugins möglich, wodurch eine hohe Bandbreite an Zusatzfunktionen für das CMS erreicht wird. Im Folgenden werden die beiden zusätzlich verwendeten Erweiterungen Sourcerer<sup>17</sup> und JoomlaFCK<sup>18</sup> behandelt.

<sup>&</sup>lt;sup>17</sup>http://extensions.joomla.org/extensions/edition/custom-code-in-content/ 5051

<sup>&</sup>lt;sup>18</sup>http://extensions.joomla.org/extensions/edition/editors/90

Da Joomla! normalerweise die Anwendung von Quellcode im Inhalt, zum Beispiel in Artikel, verbietet, wurde das Plugin Sourcerer verwendet. Diese Erweiterung erlaubt es Quellcode, wie PHP oder JavaScript, im Inhalt von Joomla! auszuführen. Dies wird benötigt um beispielsweise externe PHP-Scripts einzubinden. Durch die Verwendung folgenden Codes in Listing 13, kann ein Ausführen externer Scripts erfolgen.

Listing 13: Einbinden von Quellcode in den Inhalt von Joomla!

Dabei ist zu beachten, dass die Möglichkeit des Ausführens von Quellcode im Inhalt ein beträchtliches Sicherheitsrisiko darstellen kann, da alle Personen mit Berechtigungen zum Erstellen oder Bearbeiten von Artikeln, schädlichen Quellcode ausführen könnten. Um dieses Sicherheitsproblem zu beheben, wurde diese Funktion mittels eines kleinen CSS Code-Segments, zu sehen in Listing 14, behoben.

```
table.contentpaneopen tr td span.hasTip a img {
    display: none;
}
```

Listing 14: Entfernen der Editier-Funktion

Dabei wurde die Schaltfläche zur Aktivierung der Editier-Funktion ausgeblendet und ist somit im Front-End nicht mehr zugänglich.

Das Plugin JoomlaFCK ist ein What You See Is What You Get-Editor, kurz WYSIWYG-Editor. Dieser dient als Ersatz für den mitgelieferten Joomla!-Editor. JoomlaFCK wurde aufgrund der erhöhten Benutzerfreundlichkeit verwendet, da dieser eine Benutzeroberfläche besitzt, die der Oberfläche von Microsoft Office sehr ähnelt. Dadurch ist nahezu keine zusätzliche Einarbeitungszeit seitens des Benutzers notwendig.

Im Anhang dieser Arbeit, unter dem Kapitel *Joomla!* 1.5, werden die einzelnen Installationsund Konfigurationsschritte für den Betrieb eines Webservers aufgezeigt.

#### 3.3 Web-Applikation

Der Begriff Web-Applikation (Web-Anwendung) lässt sich anhand von Joomla! erläutern. Joomla! ist eine Anwendung auf einem im Internet als Server fungierenden Rechner. Wird das Internet als Transportmedium zum Datenaustausch zwischen zwei Computern hinzugezogen, ist dies eine Web-Applikation. Genau genommen ist jedes CMS ein Webserver. Komponenten und Module eines CMS agieren als Anwendungen im Web. Ein weiteres Beispiel einer Web-Applikation ist die Google Maps-API, welches in dieser Arbeit verwendet wird.

#### Google Maps-API

Dieser Abschnitt zeigt einen Überblick der Entstehung der *Google Maps-API*, sowie deren rechtlichen Rahmenbedingungen auf. Weiterhin werden erste Schritte zur Verwendung der API aufgezeigt.

#### 3.3.1 Überblick

Die Firma Google stellte eine Web-Applikation zur Verfügung, welche Satellitenbilder und Landkarten über ein Interface bereitstellt. *MapQuest* [39] schaffte vor Google die Voraussetzungen, Wegbeschreibungen online abzurufen. Am 08.02.2005 war auf der Internetseite von Heise Online folgender Artikel zu lesen [4]:

"Ohne offizielle Mitteilung hat heute der Suchmaschinenbetreiber Google in den USA einen neuen Dienst gestartet: 'Google Maps' bietet eine stufenlos zoombare Karte der gesamten USA, einen Routenplaner und eine Suchmöglichkeit nach Geschäften in der Umgebung. Der Dienst stützt sich offenbar auf Kartenmaterial des Unternehmens TeleAtlas."

Zu finden war der Dienst unter der Labor-Adresse von Google<sup>19</sup>. Neue Entwicklungen werden zunächst auf diesem Wege der Öffentlichkeit vorgestellt. Die Suche nach Stadtplänen war zu diesem Zeitpunkt keine neue Erfindung. Dennoch bewerkstelligte Google, sich von den Konkurrenten abzuheben, indem auf einer Karte Orte markiert werden konnten [35]. Mit der Veröffentlichung der Google Maps-API, gelang schließlich der Durchbruch zum bekannten Kartensuchdienst. Externen Programmierern wurde die Erstellung auf Google Maps basierenden Anwendungen ermöglicht, da Google den benötigten Quellcode online stellte. MapQuest hingegen verhinderte die Verwendung der eigenen Daten und hat auf Grund dessen seine Marktherrschaft schnell gegen Google, Yahoo! Maps [63] und Microsoft Virtual Earth [40] verloren.

Inwiefern Google eine kostenlose Nutzung des Kartenmaterials und der hauseigenen API weiterhin gestattet, ist fraglich.

<sup>&</sup>lt;sup>19</sup>http://labs.google.com-http://www.googlelabs.com/

Einen Nutzen zieht, neben Microsoft und Yahoo, ebenso Google mit der unentgeltlichen Leistung. Eine Registrierung wird vorausgesetzt, um den eindeutigen Schlüssel zu erhalten. Im Falle von Google erfolgt dies über das Google-Mail-Konto. Dadurch, dass die E-Mail Adresse zusammen mit der URL der Webseite dem Unternehmen bekannt ist, können Daten gesammelt werden. Beispielsweise über das Nutzungsverhalten, Verwendung der API, sowie Informationen über die Besucher der Webseite. Google speichert, analysiert und verwertet diese [26]. Google sichert sich auf diesem Wege vor Missbrauch des eigenen Application Programming Interface ab.

Nutzungsbedingungen für Google Maps, rechtliche Hinweise und Richtlinien zum Bildmaterial führt Google in einem Unterpunkt der Allgemeinen Geschäftsbedingungen an [33].

#### 3.3.2 Erste Schritte

Die Dokumentation [32] der Google Maps-API Konzepte ist ausführlich und eignet sich zum Entwickeln erster Kartenfunktionen. Gleichwohl sei darauf hingewiesen, dass die Dokumentation keine vollständige Funktionsreferenz wiederspiegelt [30]. Daher empfiehlt es sich, bei den angebotenen Code-Beispielen, den Quelltext nach bisher nicht erfassten Methoden zu durchsuchen [29], [28]. Im Besitz eines API-Schlüssels und einem funktionsfähigen Webserver, steht dem ersten Schritt zur selbstimplementierten Karte nichts mehr im Wege, wie dieses Code-Beispiel 15.

```
<html xmlns="http://www.w3.org/1999/xhtml">
1
       <head>
2
з
        . . .
       <script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=</pre>
4
           API KEY&sensor=true" type="text/javascript"></script>
       <script type="text/javascript">
\mathbf{5}
            function initialize() {
6
                 if (GBrowserIsCompatible()) {
7
                var map = new GMap2(document.getElementById("map canvas"));
8
                map.setCenter(new GLatLng(37.4419, -122.1419), 13);
9
                }
10
            }
11
       </\operatorname{script}>
12
       </head>
13
       <body onload="initialize()" onunload="GUnload()">
14
       <div id="map canvas" style="width: 500px; height: 300px"></div>
15
       </body>
16
  </html>
17
```

Listing 15: Google Map einbinden

In der fünften Zeile des Code-Beispiels 15 ist der persönliche API-Key einzufügen. Die Verwendung der neuesten Google API Version deutet v=2 an. Mit sensor=true würde eine weitere Kontrollfunktion hinzukommen, um sich den eigenen aktuellen Standort anzeigen zu lassen.

Allerdings blendet Google diese Möglichkeit der Standortbestimmung aus. Von Zeile acht bis dreizehn beginnt die Initialisierung der Karte. Zuerst findet eine Überprüfung zur Browserkompatibilität statt. Insbesondere bei Webbrowsern mit deaktivierter JavaScript-Einstellung ist eine Darstellung der Karte nicht möglich. Hingegen treten bei allen bekannten Browsertypen keine Schwierigkeiten auf. Zeile zehn instanziiert die Klasse GMap2 mit der Angabe des Containers, üblicherweise ein DIV-Element, in welchem die Karte innerhalb der HTML-Struktur eingebunden ist. Die Funktion *setCenter* veranlasst die Anzeige eines Kartenausschnitts mit den eingegebenen Koordinaten und der gewünschten Zoomstufe. Der BODY-Tag lädt die JavaScript-Funktion *initialize()* mit der Zusatzangabe, beim Beenden der Browsersitzung den verwendeten Arbeitsspeicher des Computers wieder freizugeben. Der angesprochene DIV-Container findet sich in der nächsten Zeile wieder. Nach der Definition der Kartenhöhe und -breite wird die Google Map dargestellt.

Es ist empfehlenswert, zuerst die angebotenen Beispiele durchzugehen, bevor sich der umfangreichen und teilweise sehr komplexen Demogalerie gewidmet wird. Nützliche Informationen finden sich unter den Artikeln: von ersten Anleitungen, bis hin zu geokodierten Adressen. Sogar auf Probleme weist Google mit Lösungsvorschlägen hin.

Im Anhang dieser Arbeit, unter dem Kapitel *Joomla!* 1.5 ab Seite 86, werden die einzelnen Installations- und Konfigurationsschritte für dieses CMS aufgezeigt.
### 3.4 Projektbezogene Funktionen

Bei der Erstellung dieser Arbeit wurde von Anfang an besonderer Wert darauf gelegt, den erstellten Funktionsumfang aus einem Verzeichnis *googleMapFiles* heraus zu laden. Auf der einen Seite bleibt durch diese Maßnahme die Übersicht gewahrt. Andererseits bietet diese Vorgehensweise die elegante Möglichkeit, kurzerhand einen anderen Webserver als Testplattform einzusetzen, da nur das *googleMapFiles*-Verzeichnis kopiert werden muss.

Nachfolgend werden die zwei grundlegenden Funktionsbereiche der Marker beschrieben. Als Zusatz zur Arbeit werden die Bereiche von Routen erwähnt. Anschließend folgen Module, welche im Funktionsumfang von *Joomla!* benötigt werden.

### 3.4.1 Marker hinzufügen

Das Setzen eines Markers, einer Stecknadel auf der Karte, übernimmt *setMarkers*. Beim Funktionsaufruf lädt *Joomla!* das Script *setMarkers/index.php*. Dieses Script veranlasst den Aufbau der Google Map, initialisiert die notwendigen Kartenfunktionen, sowie das Anzeigen eines Formulars, zur Eingabe der Marker relevanten Informationen.

Zuerst wird zur Auswahl einer Bilddatei abgefragt, zu welcher ein Marker gesetzt werden soll. Ein *HTML*-Formular in der *uploadForm.php* nimmt den angegebenen Dateipfad entgegen und speichert die Datei in ein temporäres Verzeichnis auf dem Webserver. Das Formular ist durch Klicken und Ziehen mit der Maus bewegbar. Die dazu verwendete *JavaScript*-Funktion wurde von *SelfHTML* [53] entnommen. Aus der *set\_functions.php* überprüft die Methode *checkIfImageIsUploaded()*, ob es sich um eine Datei im Bildformat handelt. Andernfalls wird das Eingabeformular zurückgesetzt und eine entsprechende Meldung angezeigt.

Als nächstes findet eine Überprüfung der zulässigen Formate statt. Im Rahmen dieser Arbeit werden nur die Bildtypen Joint Photographic Experts Group (JPEG), Joint Photographic Group (JPG), Graphics Interchange Format (GIF) und Portable Network Graphics (PNG) berücksichtigt, da diese am häufigsten zur Illustrierung im Webumfeld genutzt werden. Bei dem Versuch, etwa eine Datei vom Typ Tagged Image File Format (TIFF) anzugeben, wird im Eingabeformular auf diese Einschränkung hingewiesen.

War die Dateiüberprüfung erfolgreich, wartet checkIfImageIsUploaded(), bis das Hochladen auf den Server abgeschlossen ist und verschiebt jene in das Imageverzeichnis von getMarkers. Die Abfrage getNewImageFileName() gibt die nächsthöhere Identifikationsnummer aus der Datenbank zurück. Zusätzlich mit dem Präfix geo\_ erfolgt ein Umbenennen der verschobenen Datei, beispielsweise in geo\_42.jpg. Mit dieser Maßnahme wird ein versehentliches Überschreiben einer vorhandenen Bilddatei verhindert. Parallel dazu erzeugt das Script ein sogenanntes Thumbnail. Ein kleineres Vorschaubild, mit einer maximalen Abmessung der Bilddimensionen von 200 Pixeln, siehe Abschnitt getMarkers.

Die Teilfunktion *readExif.php* durchsucht das Bild nach geocodierten Merkmalen. Beim Auffinden von Längen- und Breitengradangaben innerhalb der Bilddatei wird der Benutzer nicht zum Setzen eines Markers aufgefordert. Anschließend erwartet das *HTML*-Formular die Eingabe eines bezeichnenden Namens und den bildzugehörigen Quellenverweis in Form einer URL. Des Weiteren muss der Benutzer die Auswahl einer treffenden Kategorie tätigen, sowie eine aussagekräftige Beschreibung eingeben. Alle benannten Eingabefelder sind Pflichtfelder. Das Script lädt zusätzlich ein Vorschaubild des hochgeladenen Bildes. Sind alle Eingaben erfasst, weist das Script auf die Platzierung des Markers im Bild, auf die bildzugehörige Position, hin. Mit dem erwähnten Unterschied, dass dieser Schritt entfällt, falls geocodierte Längen- und Breitenangaben gefunden wurden. Nach Betätigen des Speicherbuttons, hinterlegt die Funktion *uploadForm.php* die erfassten Informationen in der Datenbank. Abbildung 7 zeigt ein Beispiel anhand eines ausgefüllten Formulars.

	Mühlbach	Huttich Einen neuen Märker set. Seewalchen	Neu zen (	ahm Sommerholz Fischhof click to drag)	$\otimes$
		Das Bild wurde erfolgrei Bitte geben Sie hier die Informat	ch h ione	ochgeladen! n zu diesem Bild ein.	P senschv
	Name:	Fachhochschule Salzburg			
	Quelle:	http://www.fh-salzburg.ac.at/			
	Kategorie:	Fachhochschule -			
ssi	Beschreibun	ıg:			Loibi
ieze	Informati Salzburg	onstechnik studieren in			Mondse
EG	Techniker	Innen und IT-ExpertInnen			27
	Berufsgru	ppen mit den größten	-		2
E	Bevor Sie rotfarbenen	speichern, bitte noch den Marker auf die bildzugehörige			Vinkl
	Posicion au			Dieses Bild löschen?	
.5	Speichern				Ischwar
	Nieder	alm Taxach-Rif	1. m <sup>2</sup>	See.	Brunn

Abbildung 7: Formular zur Eingabe von Informationen einer Bildungsstätte

Im Fall, dass die Eingabe an dieser Stelle ohne zu speichern unterbrochen wurde, erscheint die veranschaulichte Eingabemaske erneut, beim wiederholten Aufruf von *setMarkers*. Auf diese Weise wird gewährleistet, dass sich maximal ein noch nicht in der Datenbank erfasstes Bild auf dem Server befindet. Eine Verschwendung von Speicherressourcen ist somit ausgeschlossen. Es steht dem Benutzer frei, jenes Bild zu löschen und durch ein anderes zu ersetzen. Mit Hilfe von *Joomla!* ist eine Abfrageroutine angegliedert, wodurch es nur Administratoren erlaubt ist, die Funktion *setMarkers* einzusehen, beziehungsweise auszuführen.



Abbildung 8 stellt ein *Unified Modeling Language* (UML)-Diagramm der grundlegenden *set-Markers*-Funktionen dar.

Abbildung 8: UML-Diagramm setMarkers

### 3.4.2 Marker anzeigen

In der Datenbank hinterlegte Informationen auf der Karte anhand von Markern darzustellen, ist die Hauptaufgabe der Funktion *setMarkers*. Diese Funktion ist im Gegensatz zu *getMarkers* ohne weitere Einschränkungen oder spezifischen Rechten benutzbar.

Beim Funktionsaufruf lädt Joomla! die Datei getMarkers/index.php. Die Karte, mitsamt den Steuerelementen, initialisiert das Script map\_functions.php. Außerdem wird eine Methode retrieveMarkers() initialisiert, welche gleichzeitig das Script retrieveMarkers.php mit einbezieht. Die in der Datenbank gespeicherten Informationen zu Längen-, Breitengradwerte, Namen, Quelle, Beschreibung, Icon- sowie Bilddateiname übermittelt das PHP-Script retrieveMarkers.php als XML-formatierten Datenstring an die JavaScript-Methode retrieveMarkers() [54].

Jeder einzelne Datensatz wird *HTML*-formatiert aufbereitet und dient der Darstellung in dem von Google implementierten Informationsfensters eines Markers. Die Funktion *createMarker* vollendet die Zusammenstellung aller Angaben. Die *API*-Methode *addListener* stellt daraufhin jeden einzelnen Marker mit Positionsangaben, *HTML*-Datenstring, sowie einem Icon auf der Google Karte dar. Listing 16 zeigt in Auszügen die Funktionszeilen von *createMarker*.

```
function createMarker(point, html, iconImage){
    ...
    var marker = new GMarker(point);
    ...
    GEvent.addListener(marker, 'click', function(){
    var markerHTML = html;
    marker.openInfoWindowHtml(markerHTML);
    });
    return marker;
}
```

Listing 16: Funktion createMarker

Damit bei einer zu erwartenden hohen Anzahl von Datensätzen das zu übertragende Datenvolumen vom Server zum Client möglichst reduziert wird, sind die Vorschaubilder (geo\_1.jpg) in den Datenfluss eingebunden, wie im Listing 17 dargestellt. Es zeigt ein Beispiel des vom Web-Server zum Client gesendeten XML-Datenstrings.

```
<markers>
<marker lat="47.7241909030962" lng="13.0867874622345" name="Fachhochschule
Salzburg" text="Informationstechnik studieren in Salzburg<br&gt;&
lt;br>TechnikerInnen und IT-ExpertInnen zählen nicht nur in
Österreich zu den Berufsgruppen mit den größten Zukunftschancen..."
link="http://www.fh-salzburg.ac.at/" image="geo_1.jpg" iconname="
Fachhochschule"/>
<marker lat="47.8070443561521" lng="13.0236053466797" name="Gymnasium"
text="Text zum Gymnasium" link="http://www.gymnasium.at/" image="geo_2
.jpg" iconname="Gymnasium"/>
<marker lat="47.7804048805519" lng="12.9947662353516" name="Realschule"
text="Nur als Beispiel" link="http://www.realschule.at/" image="geo_3.
jpg" iconname="Realschule"/>
....
</markers>
```

Listing 17: Beispiel gesendeter XML-Daten der Funktion retrieveMarkers.php

Abbildung 9 stellt das Ergebnis der aus Listing 17 erstellter Bildungsstätten-Marker auf der Karte dar.



Abbildung 9: Ausgabebeispiel der Funktion getMarkers

Die Ausgabe der Informationen zu einem bestimmten Marker werden in Abbildung 10 dargestellt. In diesem Beispiel wurde der Name der Bildungsstätte als Überschrift gewählt. Informationen werden auf der rechten Hälfte der "Sprechblase" angezeigt, zusammen mit einem Hyperlink Bildungsstätte. Die linke Hälfte stellt verschiedene Ansichten der Bildungseinrichtung dar.



Abbildung 10: Informationen zur FH-Salzburg



Abbildung 11 stellt ein UML-Diagramm der grundlegenden getMarkers-Funktionen dar.

Abbildung 11: UML-Diagramm getMarkers

### 3.4.3 Routen hinzufügen

Mit der Funktion setRoutes wurde die Möglichkeit zum Hochladen von Routen in Form von KML- beziehungsweise *GPS Exchange Format* (GPX)-Dateien implementiert. Dazu muss im dafür vorgesehenen Formular eine Datei mit Geodaten im KML- oder GPX-Format ausgewählt und durch zusätzliche Daten wie Titel, Quelle und Beschreibung ergänzt werden. Das Formular wird als verschiebbares Fenster angezeigt. Um die Beweglichkeit des Fensters zu ermöglichen wurde eine JavaScript-Funktion von SelfHTML, einer angesehenen Dokumentation für Webprogrammierung, übernommen [53]. Nach dem Absenden des Formulars wird überprüft ob alle erforderlichen Daten korrekt eingegeben wurden. Ist dies nicht der Fall erscheint eine Aufforderung zur Komplettierung der Zusatzinformationen. Sind alle Daten vorhanden, wird die Route durch folgenden Quellcode 18 auf dem Server gespeichert:

...
move\_uploaded\_file(\$route['tmp\_name'], \$target\_path)
...

Listing 18: Hochladen der Route

Ist die hochgeladene Route im KML-Format, wird diese mit dem Namen *trackXX.kml* auf dem Webserver gespeichert, wobei *XX* jeweils eine Zahl darstellt, die automatisch von der Funktion *move\_uploaded\_file* erhöht wird. Wenn die hochgeladene Datei im GPX-Format vorliegt, wird diese durch die implementierte Funktion *convertGPX* umgewandelt, da Google Maps ausschließlich KML-Dateien darstellen kann. Durch den in Code-Beispiel 19 dargestellten Quellcode werden die Wegpunkte aus der GPX-Datei ausgelesen und in die neu erstellte Datei geschrieben. Alle anderen enthaltenen Daten, können ignoriert werden, da diese statisch erzeugt werden.

```
...
// lese GPX Wegpunkte
foreach ($gpx->trk->trkseg->trkpt as $key => $waypoint) {
    // write waypoints in $kml variable
    $kml .=" ".$waypoint["lon"].",".$waypoint["lat"].",0\n";
}
...
```

#### Listing 19: Wegpunkte in KML-Datei schreiben

Die Idee für die Funktion zur Umwandlung von GPX-Dateien stammt ursprünglich aus einem Script zur Konvertierung von Geocaching GPX-Files [34, 58]. Da der Anwendungsbereich dieses Scripts nicht für diese Arbeit geeignet war, wurde basierend auf diesem eine komplett neue Funktion implementiert. Nach der Konvertierung erhält die Datei den bereits erwähnten Namen *trackXX.kml*. Gleichzeitig wird die GPX-Datei wieder vom Server entfernt, da sie nicht mehr notwendig ist. Wurde die Route hochgeladen, extrahiert die Funktion *getStartCoords*, dargestellt in Code-Beispiel 20, die Startkoordinaten der Route.

```
...
// search and assign values of "coordinates" to variable
foreach( $kml as $val )
{
     $element = $val->getElementsByTagName("coordinates");
     $data = $element->item(0)->nodeValue;
}
// separate coordinates at new line
$coords = explode("\n", ltrim($data));
// separate start coordinate
$start
...
```

Listing 20: Extrahieren der Startkoordinaten

Zusätzlich werden die gesamten erfassten Daten wie Startkoordinaten, Titel, Quelle, Beschreibung und Dateiname durch den in Listing 21 beschriebenen Quellcode in die dafür vorgesehene Datenbank eingetragen.

```
...
// write the data into the database
$query = "INSERT INTO 'db_routes' ('lat', 'lng', 'name', 'text', 'link', '
route') VALUES ('".$coords[1]."', '".$coords[0]."', '".
htmlspecialchars($title)."', '".htmlspecialchars($description)."', '".
htmlspecialchars($source)."', '".$name."')";
mysql_query($query);
...
```

Listing 21: Eintragen der Daten in die Datenbank

Dabei werden Sonderzeichen, wie zum Beispiel Hochkommas oder Anführungszeichen, herausgefiltert um Darstellungsfehler zu verhindern. Die Daten werden in folgender Form in die Datenbank eingefügt, dargestellt in Abbildung 12.

	id_geo n		name	text	link	route
1	×	1	Gaisbergrunde	Eine wunderschoene tolle Runde	www.bikemap.at	track1.kml
1	$\mathbf{X}$	2	USA - Chicago	Verschieden Routen in Chicago/USA	www.google.com	track2.kml
1	$\mathbf{X}$	3	Salzburg - Venedig	Route von Salzburg nach Venedig!	www.url.com	track3.kml

Abbildung 12: Tabelleneinträge verschiedener Routen

Wurden alle Daten in der Datenbank abgelegt, kann eine neue Datei hochgeladen, beziehungsweise die neu hinzugefügte Route durch die Funktion *getRoutes* auf der Karte angezeigt werden.

### 3.4.4 Routen anzeigen

Die Funktion getRoutes wurde entwickelt, um alle am Server und in der Datenbank vorhandenen Routen auf der Karte darzustellen. Dabei werden die Routen als farbige Linien mit einem Marker am Startpunkt dargestellt. Um eine Darstellung zu ermöglichen müssen die erforderlichen Daten aus der Datenbank gelesen und mithilfe der Google Maps-API auf das Kartenmaterial übertragen werden. Der Marker am Startpunkt jeder Route enthält alle Daten die mithilfe von setRoutes in der Datenbank abgelegt wurden. Diese Informationen erscheinen durch einen einfachen Klick auf den Marker anhand eines Informationsfensters. Damit die Datensätze aus der Datenbank übertragen werden können, müssen diese, wie in Listing 22 dargestellt, ausgelesen werden.

```
...
// mysql query
$query = mysql_query("SELECT lat, lng, name, text, link, route FROM
db_routes");
...
```

Listing 22: Tabelleninhalte aus Datenbank auslesen

Es handelt sich dabei nahezu um dieselbe Funktion die bei *getMarkers* zur Anwendung kam. Um diese für *getRoutes* verwenden zu können, waren leichte Modifikationen erforderlich. Sobald die Daten aus der Datenbank ausgelesen wurden, kommt die Google Maps-API zum Einsatz. Die dazu verwendeten Funktionen sind grundlegend dieselben wie bei *getMarkers*. Diese mussten wiederum leicht abgeändert beziehungsweise erweitert werden. Mithilfe der JavaScript-Funktion *createRoute* aus Listing 23, wird die von der Datenbank ausgelesene KML-Datei in Google Maps geladen.

```
...
// get routes from database with absolute URL
function createRoute(path) {
    // assign the absolute URL of the KML-File to a variable
    var route = new GGeoXml(http://www.some-domain.at/googleMapFiles/
    getRoutes/routes/+ path);
    return route;
}...
```

### Listing 23: KML-Datei laden

Anhand des Listings wird dargestellt, dass die absolute URL der KML-Datei Verwendung findet. Dies ist unbedingt erforderlich, da die Google Maps-API KML-Dateien nur dann lesen und einbinden kann, wenn die absolute URL zur Verfügung steht. Darauf wird die geladene Route sowie der dazugehörige Marker, mit den aus der Datenbank erhaltenen Zusatzinformationen, auf der Karte platziert. Listing 24 zeigt die hierfür notwendige Funktion.

```
...
// get routes from database with absolute URL
function createRoute(path) {
    // assign the absolute URL of the KML-File to a variable
    var route = new GGeoXml(http://itsgeoweb.meyr.info/googleMapFiles/
    getRoutes/routes/+ path);
    return route;
}...
```

Listing 24: Platzieren der Route und des Markers

Danach können alle auf diese Weise in das Kartenmaterial eingebundenen Routen und Markierungen betrachtet werden. Außerdem besteht die Möglichkeit einzelne Routen durch einen Klick auf *Download Route* im Informationsfenster in Form einer KML-Datei herunterzuladen. Mithilfe der in diesem Kapitel beschriebenen Funktionen konnte das Ziel der verschiedenen Darstellungsmöglichkeiten von Markierungen und Routen erreicht werden (Abbildung 13).



Abbildung 13: Verschieden Routen in Chicago (USA)

Der dazugehörige XML-Datenstring, welcher durch die Funktion *retrieveRoutes.php* erzeugt wird, ist in Listing 25 dargestellt.

```
<routes>
<route name="Gaisbergrunde" text="Eine wunderschoene tolle Runde" link="
www.bikemap.at" route="track1.kml"/>
<route name="USA - Chicago" text="Verschieden Routen in Chicago/USA" link=
"www.google.com" route="track2.kml"/>
<route name="Salzburg - Venedig" text="Route von Salzburg nach Venedig! "
link="www.url.com" route="track3.kml"/>
</routes>
```

Listing 25: Beispiel gesendeter XML-Daten der Funktion retrieveRoutes.php

### 3.4.5 Modul Kategorien auswählen

Als Modul definiert *Joomla!* einen kleinen dynamischen Bereich einer Webseite. Beispiele für Module sind die Suchfunktion oder das Formular zur Anmeldung als registrierter Benutzer des CMS. *Joomla!*-Module lassen sich innerhalb der Seitenstruktur frei positionieren. Sie können als kleine *PHP*-Zusatzprogramme verstanden werden, welche vorhandenen Inhalt neu aufbereiten, laden oder verändern.

Die Auswahl zur Anzeige der Bildungsstätten kann mit diesem Modul präzisiert werden. Es werden zum Beispiel bei der Kategorie "Fachhochschule" nur diese Bildungseinrichtungen auf der Karte angezeigt. Alle anderen Schulen (Gymnasien, Berufsschulen, usw.) werden ausgeblendet.

Die Klassenfunktion *getCategories* der *PHP*-Datei *helper.php* baut eine Verbindung zur Datenbank auf, dargestellt in Listing 26.

```
public function getCategories() {
    db = \&JFactory :: getDBO();
    // categories in database are unique - but just to get sure - distinct
        query
    $query = "SELECT id cat, iconname FROM db category ORDER BY iconname
       ASC":
    $db->setQuery($query);
    if (! $db->query()) {
        echo "<script type='text/javascript'>
        alert('".$db->getErrorMsg()."');
        </\operatorname{script}>";
        return false;
    else 
        $rows = $db->loadObjectList();
        return $rows;
        }
}
```

Listing 26: Auslesen der Kategorien aus der Datenbank

Aus der Tabelle *db\_category* werden alle vorhandenen "Iconnamen" alphabetisch sortiert zusammen mit ihrer *Identitätsnummer* (ID) ausgelesen. Diese dienen sowohl als Bezeichner der einzelnen Kategorien als auch als Dateiname des zu ladenden Kategorie-Icons (siehe Kapitel 3.4.2, Abbildung 9). Die Funktion *getCategories* übergibt den Tabelleninhalt als Array an das *PHP*-Script *mod\_edu.php*. Wie in Listing 27 aufgezeigt, werden die einzelnen Kategorien aus dem Array zu einem *HTML*-String zusammengesetzt. Dieser String bildet für jeden Kategorienamen einen Hyperlink mit der ID in der *Uniform Resource Identifier* (URI).

```
foreach ($test as $row) {
    echo("<a href=\"".$__uri."&facilities=".$row->id_cat."\">".$row->
        iconname."</a><br/>br />");
}
```

Listing 27: Ausgabe der Kategorien als Hyperlink

Die ID dient zur Identifizierung einer eindeutigen Kategorie. Die vorher erwähnte Funktion *retrieveMarkers.php* (Kapitel 3.4.2) liest Server-seitig die übertragene ID aus dem Hyperlink aus und gibt nur die relevanten Kategorien an den Client-Browser zur Darstellung auf der Karte zurück, wie in Listing 28 dargestellt.

```
function retrieveMarkers(){
...
var topic_url = ("googleMapFiles/getMarkers/retrieveMarkers.php" + "/?
facilities=" + get_url_parameter('facilities'));
...
}
```

Listing 28: Auszug aus dem Script map\_functions.js von Kapitel 3.4.2

Eine mögliche Ausgabe des Moduls verschiedener Kategorien von Bildungsstätten auf der Google-Karte der Webseite stellt die Abbildung 14 dar.



Abbildung 14: Modul zur Auswahl bestimmter Bildungsstätten

### 3.4.6 Modul Kategorien bearbeiten

Mithilfe dieses Modules ist ein Administrator der Webseite in der Lage, vorhandene Kategorien von Bildungsstätten zu löschen. Wird zum Beispiel die Bildungsstätten-Kategorie "Realschule" entfernt, werden folglich keine "Realschulen"-Marker auf der Karte angezeigt. Die in der Datenbank hinterlegten Informationen zu einer gelöschten Kategorie sollten ebenso wie bisher gespeichterte Bilder zu den einzelnen Schulen erhalten bleiben.

Um ein versehentliches Aktivieren der Löschfunktion zu verhindern, wurde eine Sichheitsabfrage (Listing 29) in der Scriptsprache JavaScript innerhalb der PHP-Datei mod\_edu\_cat.php integriert.

```
<script language="JavaScript">
  function delConfirm(nr, name){
     if (confirm("Soll " + name + " wirklich gelöscht werden?")) {
        document.location = "<?php echo $__uri;?>&deletecat=" + nr;
     }
   }
  </script>
```

Listing 29: Sicherheitsabfrage vor dem Löschen einer Kategorie

Wird die Sicherheitsabfrage des Löschvorgangs bestätigt, übergibt ein *HTML*-Formular die ID der zu löschenden Kategorie an das *PHP*-Script *mod\_edu\_cat.php*. Die Klassenfunktion *delCategory*, dargestellt in Listing 30, sendet die Löschanweisung der Kategorie-ID an die Datenbank.

```
public function delCategory($int) {
    $db = &JFactory::getDBO();
    // categories in database are unique - but just to get sure - distinct
    query
    $query = "DELETE FROM db_category WHERE id_cat = `".$int."`";
    $db->setQuery($query);
    if (!$db->query()) {
        echo "<script type='text/javascript'>
        alert(`".$db->getErrorMsg()."`);
        </script>";
        return false;
    }
    else {
        return true;
    }
}
```

Listing 30: Klassenfunktion delCategory zum Löschen einer Kategorie

Dieses Modul übernimmt nicht nur das Entfernen vorhandener Kategorien, sondern es können ebenfalls neue Bildungsstätten-Kategorien angelegt werden. Der Administrator gibt den Namen einer neuen Schulkategorie ein. Die Klassenfunktion *checkCategories* überprüft, ob dieser Kategoriename bereits in der Datenbank vorhanden ist. Ist die neue Kategorie noch nicht vorhanden wird diese angelegt. Listing 31 stellt Statusmeldungen dar, welche über mögliche Fehler bei der Eingabe eines neuen Kategorienamens informieren.

```
// write new category to database
if (isset($ POST["submit"])) {
    if (($_POST["category"] != "")) {
       // check if category is already in database
       if (( edu cat::checkCategories($ POST["category"])) == "") {
           // category does not exist - add new category
           __edu_cat::setCategory($_POST["category"]);
           echo("<span class=\"edu cat ok\"><br />Die Kategorie <b>\"".
               $ POST["category"]."\"</b> wurde hinzugefügt.<br /></span>
               ");
       } else {
           // category exists
           echo("<span class=\"edu_cat_error\"><br />Die Kategorie <b>\""
               .$_POST["category"]."\"</b>
               /></span>");
       }
   } else {
       // button pressed without input
       echo("<span class = \"edu_cat_error \">Sie haben keine Kategorie
           eingegeben!</span>");
   }
}
```

Listing 31: Neuen Kategorienamen hinzufügen

Ein entsprechendes Kategorie-Icon muss als nächster Schritt im Modul-Formular ausgewählt werden, um dieses beispielsweise zu einer Kategorie "Universität" hinzuzufügen. Anderenfalls werden neu angelegte Marker zu "Universität" nicht angezeigt. Des Weiteren besteht die Möglichkeit, ein vorhandenes Icon durch ein neues oder anderes zu ersetzen. Abbildung 15 zeigt das Modul zum Bearbeiten von Bildungsstätten.

Bildungsstätt	en be	arbeiten										
Nachfolgende Kategorien existieren:												
Kategorie Icon Aktion												
Fachhochschule												
Gymnasium	GYM 7	Löschen										
Realschule	RS	Löschen										
Universität		Löschen										
Neue Kategorie anlegen: Kategorie anlegen												
Kategorieicon auswählen (PNG Format 32px * 34px): Durchsuchen												
und zu dieser Ka Bitte auswählen	und zu dieser Kategorie hinzufügen: Bitte auswählen 🔻											
Kategorieicon h	Kategorieicon hochladen											

Abbildung 15: Modul zum Bearbeiten der Kategorien Bildungsstätten

### 3.4.7 GPX-Routen

Zusätzlich zum Umfang dieser Arbeit wurde eine erweiterte Form von KML hinzugenommen, das Global Positioning System Exchange Format (GPX). Mit Hilfe von GPX können weitere Informationen einer Route gespeichert werden. Abbildung 16 stellt am Beispiel einer Fahrradtour die Angaben zur Höhe und Geschwindigkeit dar. Zu Testzwecken wurden die Funktionen des GPX Viewer [3] zum Projekt dieser Arbeit mit eingebunden und in diesem Abschnitt kurz erwähnt. GPX Viewer soll Möglichkeiten aufzeigen und Interesse an einer Fortsetzung dieser Arbeit wecken.



Abbildung 16: Höhen-, Steigungs- und Geschwindigkeitsangaben einer GXP-Datei

Die Ausgabe der dieser *GPS Exchange Format*-Datei basiert auf der vorher erwähnten Funktion *Routen anzeigen*. Jedoch lassen sich mit den Daten aus dem *GPS Exchange Format*-File nachfolgende Diagramme auf der Webseite darstellen. Die Abbildung 17 stellt die überwundene Höhe auf die gefahrene Strecke dar. Die gemessenen Geschwindigkeiten in Abhängigkeit zu einem Streckenzeitpunkt werden in Abbildung 18 dargestellt. Die prozentuale Steigung eines Streckenabschnittes gibt Abbildung19 wieder.







Abbildung 18: Geschwindigkeitsdiagramm GXP-Datei



Abbildung 19: Steigungsdiagramm $G\!X\!P\text{-}\mathsf{Datei}$ 

### 3.5 Portabilität

Der Begriff Portabilität bezeichnet in der Informatik die Fähigkeit einer Software verstanden, plattformunabhängig eingesetzt zu werden. Des Weiteren spielt der geschätzte Arbeitsaufwand eine Rolle. Das Softwareprodukt auf eine Weise zu modifizieren, damit es auf anderen Systemen lauffähig ist.

Von Beginn dieser Arbeit an wurde auf die Portabilität der einzelnen Softwarekomponenten geachtet.

Im Folgenden werden zwei Möglichkeiten aufgezeigt. Im Rahmen dieser Arbeit beschränkt sich die Portierung von einem *Joomla!*- auf ein weiteres *Joomla!*-System. Dabei ist darauf zu achten, dass beide Installationen zumindestens *Joomla!*-Version 1.5 aufweisen. Werden andere CMS eingesetzt sind Anpassungen im Template notwendig. Außerdem muss ein anderes CMS die Ausführung von PHP-Code im Inhalt erlauben.

### 3.5.1 Projektdateien

Die erforderlichen Dateien des Gesamtprojekts befinden sich im entsprechenden Webverzeichnis auf dem Webserver, aus welchem heraus die *Joomla!*-Installation geladen wird:

- googleMapFiles
- modules/mod\_edu
- modules/mod\_edu\_cat
- templates/geoweb

Diese Ordnerstruktur ist mitsamt den beinhaltenden Unterordnern zu übernehmen und in die entsprechenden Verzeichnisse des Zielsystems einzufügen. Beispielsweise muss sich der Template-Ordner geoweb auf einer neuen Joomla!-Webseite ebenfalls wieder im Templateverzeichnis template/ befinden. Mit dem Kopieren der weiteren Ordnern ist äquivalent zu verfahren.

Mit dem Datenbank-Management-Werkzeug phpMyAdmin sind die beiden Tabellen  $db\_category$ und  $db\_icon$  zu exportieren. Die von phpMyAdmin erstellten Datenbankdateien  $db\_category.sql$ und  $db\_icon.sql$  sind mittels phpMyAdmin in die Zieldatenbankstruktur zu importieren.

Im Anhang dieser Arbeit, Kapitel *Joomla!*, findet sich die weitere Vorgehensweise. Die Einstellungen der neuen *Joomla!*-Webseite, auf welcher das Geo-Projekt genutzt werden möchte, sind den aufgelisteten Schritten entsprechend anzupassen.

### 3.5.2 Gesamte Webseite

Weitaus einfacher gestaltet sich ein "Umzug" der gesamten vorhandenen *Joomla!*-Webseite, auf welcher das Geo-Projekt bereits integriert ist.

Hierzu ist die vollständige Verzeichnisstruktur des *Joomla!*-Verzeichnisses in den dafür vorgesehenen Web-Ordner des Zielservers zu kopieren. Mittels *phpMyAdmin* werden alle vorhandenen Tabellen exportiert, sowie ebenfalls mit dem Werkzeug *phpMyAdmin* des neuen Webservers importiert. Die Abbildung 20 stellt die notwendigen Einstellungen zur Erstellung eines Abbildes der gesamten *Joomla!* dar.



Abbildung 20: Datenbank mit phpMyAdmin exportieren

Im Root-Verzeichnis der Ziel-Installation von *Joomla!* befindet sich die PHP-Datei *configuration.php*. Die im Listing 32 aufgeführten Zeilen sind in der *configuration.php*-Datei entsprechend anzupassen.

```
<?php
class JConfig {
    // Auszug der configuration.php
    var log_path = '/joomlaverzeichnis/logs'; // Angaben des Webhosters
       beachten!
    var $tmp path = '/joomlaverzeichnis/tmp'; // Angaben des Webhosters
       beachten!
    var $host = 'xxx'; // Servername der Datenbank
    var $user = 'xxx'; // Username Datenbank
    var $db = 'xxx'; // Name der Datenbank
    var $password = 'xxx'; // Datenbankpasswort
    var $dbprefix = 'edu_';
    var $mailfrom = 'xxx@yyy.zz'; // E-Mail des Administrators
    var $smtphost = 'localhost';
    // Auszug der configuration.php
}
?>
```

Listing 32: Notwendige Anpassungen configuration.php-Datei

Die Anforderungen an ein portierbares Projekt für das *Content Management System Joomla!* wurden durch die aufgezeigten Schritte für die Projektdateien erfüllt. Darüberhinaus wurden die einzelnen Anweisungen der Portierung der gesamten *Joomla!*-Projekt Webseite aufgezeigt.

# 4 Lehrveranstaltung

Eine Gruppe von Schülern des Bundesgymnasiums in Salzburg [24] nimmt an einem Wettbewerb "Wissenschaft ruft Schule - Schule ruft Wissenschaft" des Forschungsprogrammes von Sparkling Science teil [38]. Dieses Handbuch zur Erstellung eines Geoinformationssystems soll eine Grundlage für ein erfolgreiches eigenes Projekt der Schüler darstellen. Dabei die wichtigsten Grundbegriffe klären und Verständnis in Bezug auf die benötigten Anwendungsprogramme schaffen.

Die theoretischen und praktischen Inhalte dieser Arbeit wurden den Schülern im Rahmen einer Lehrveranstaltung zu vier Einheiten vermittelt. Auf welche Weise im Unterricht vorgegangen wurde, sollen die nachfolgenden Unterpunkte kurz erläutern

# 4.1 Erste Unterrichtseinheit

Zu Beginn der ersten Einheit wurde das Thema des Projektes angesprochen und näher erörtert. Es galt das Ziel und die Anforderungen, die sowohl an das Projekt, als auch an diese Arbeit gestellt werden, gemeinsam einzugrenzen und zu diskutieren.

Daraufhin wurde der Begriff Webserver erläutert. Die Begriffsdefinition begann mit den verschiedenen Arten von Webservern. Anhand eines Beispiels vom Aufruf einer Webseite bis hin zur Darstellung auf dem Client-Browser. Ein konkretes projektbezogenes Beispiel informierte die Schüler anhand eines skizzierten Tafelbildes über den Ablauf der Kommunikationen zwischen einem Webserver, dem Client und dem API-Anbieter Google Maps.

Von Vorteil war, dass jedem Schüler im Unterrichtsraum ein Computer zur Verfügung stand. Erste Erfahrungen mit der Installation der lokalen Webserver-Software XAMPP konnten somit gewonnen werden. Der Installationsprozess ist auf einem Beamer unterstützend Schritt für Schritt demonstriert worden. Nach der Installation lieferte XAMPP eine erste Beispielseite an die Browser der Arbeitsplatzrechner.

Vor dem Aufsetzen von *Joomla!* folgte eine Erklärung über CMS. Auf welche Weise ein CMS von einem Client angeforderte Inhalte zusammenstellt und diese als Webseite an den Client-Browser zurückschickt.

## 4.2 Zweite Unterrichtseinheit

Die zweite Einheit begann mit einer kurzen Wiederholung der letzten Unterrichtsstunde. Anhand der *Joomla!*-Startseite wurden die verschiedenen Bereiche, wie Komponenten, Module und Plugins zum besseren Verständnis der Schüler erklärt.

Nach dem Anmeldevorgang der Schüler als Administrator der Webseite, folgte eine Einweisung in den Aufbau des *Joomla!*-Backends. Es wurde auf die wichtigen Konfigurationspunkte zur Webseite, dem System und die Servereinstellungen der Administratorebene eingegangen. Das nachträgliche Ändern der Webseitenbezeichnung gehörten ebenso zum Einweisungsumfang, wie die Funktion der Benutzerregistrierung und relevante Einstellungen der Serverattribute. Joomla! stellt eine Vielzahl von informativen Beispielinhalten bereit. Diese Inhalte galt es zu löschen, da sie für das Geoinformationsprojekt belanglos sind. Daraufhin wurden nicht benötigte Module deaktiviert.

Anhand eines Beitrages wurde der Zweck des zusätzlichen Plugins "Sourcerer" veranschaulicht. Von einem Netzlaufwerk konnten die Schüler die erforderlichen Verzeichnisse für das Geo-Projekt und ein vorgefertigtes Template in die entsprechenden Ordner des Webservers kopieren.

# 4.3 Dritte Unterrichtseinheit

Diese Einheit begann mit der Eingabe der Zugangsdaten zur Datenbankverbindung. Die Datenbanktabellen wurden mit dem Datenbank-Werkzeug *phpMyAdmin* eingefügt. Daraufhin folgte eine Erklärung zu *phpMyAdmin* und der *Joomla!*-Datenbankstruktur. Die Schüler erhielten eine Überblick vom Aufbau der *Joomla!*-Datenbanktabellen.

Die im Rahmen dieser Arbeit erstellten Module haben die Schüler installiert. Als nächstes wurden die Modulpositionen eingerichtet, sowie die Beiträge *getMarkers* und *setMarkers* erstellt. Die Menüstruktur wurde erklärt und für das Projekt entsprechend angepasst.

Zwischenzeitlich wurde auf Probleme eingegangen, welche bei den Schüler beim Versuch der Installation am eigenen Rechner von *Joomla!* und *XAMMP* auftraten.

Die Schüler wurden in dieser Unterrichtseinheit bei Konfigurationsproblemen und Anzeigefehlern gleichermaßen unterstützt, wie in den vorherigen Einheiten. Ein wichtiges Unterrichtsziel war es, dass alle Schüler zu jeder Zeit auf demselben Stand sind.

## 4.4 Vierte Unterrichtseinheit

Die Lehrinhalte der letzten drei Einheiten bildeten den Anfang dieser Stunde. Anschließend wurde die globale Beitragskonfiguration angepasst und erläutert.

Die in *Joomla!* integrierte Baumstruktur zur Verwaltung der Bereiche, Kategorien und Beiträge vermittelte ein mit Beispielen versehenes Tafelbild.

Abschließend verdeutlichte eine Demonstration den Schülern die Funktionsweise und Handhabung der Module im Frontend der Webseite. Dies beinhaltete ebenfalls das Erstellen neuer Kategorien, den zugehörigen Icons, sowie der Anmeldung als registrierter Benutzer, um einen neuen Marker setzen zu können.

Das Ziel, den Lehrinhalt und die Installation des Geoinformationssystems, wurde in den vier Unterrichtseinheiten erreicht. Zur Klärung von Fragen, welche nach dieser vierten Einheit auftreten könnten und zur Heranführung der Schüler an den Sparkling Science Wettbewerb, wird eine fünfte Unterrichtseinheit angestrebt. Ein gemeinsamer und unterstützender Erstellvorgang der ersten Bildungsstätten dient zur Abrundung der gesamten Lehrveranstaltung.

## 5 Zusammenfassung und Ausblick

Die geforderten theoretischen und praktischen Ziele dieser Arbeit wurden vom Autor erreicht.

Der theoretische Bestandteil liegt in Form dieser Arbeit vor, der Erstellung eines Handbuchs für ein Geoinformationssystem. Im Kapitel 2 wurden die Grundlagen zu geocodierten Daten, Google Maps, Content Management Systemen und Webservern erläutert. Die verwendeten Web-Technologien wurden jeweils mit Beispielen unterlegt. Des Weiteren enthält der Anhang dieser Arbeit Anleitungen zur Installation eines Webservers, des CMS Joomla! und ein Quellcodeverzeichnis der erstellten Funktionen.

Der praktische Bestandteil gliedert sich in zwei Bereiche. Zum einen die Erstellung und Umsetzung eines Geoinformationssystems basierend auf Joomla! und der Google Maps-API. Zum anderen die Installation eines Webserver-Betriebssystems zur gleichzeitigen Präsentation mehrerer Geoinformationssysteme. Die Portabilität der im Kapitel 3.4 Projektbezogene Funktionen wurde zusätzlich um die Portierung des gesamten CMS erweitert.

In dieser Arbeit wurde das Geo-Projekt anhand von Bildungsstätten erläutert. Die Funktionen wurden vom Autor jedoch dahingehend programmiert, das Gesamtprojekt gleichermaßen für andere Themen zu nutzen, zum Beispiel der Präsentation historischer Bäume oder einer Informationsplattform aller ärztlichen und medizinischen Einrichtungen einer Stadt. Diesbezüglich wäre eine Vermarktung des Geoinformationssystems denkbar und wird seitens des Autors, mit weiteren Ideen und Entwicklungen, angestrebt.

Darüber hinaus wurden die theoretischen und praktischen Themengebiete, wie im Kapitel 4 Lehrveranstaltung dargelegt, unterrichtet. Eine zusätzliche Unterrichtseinheit soll den Schülern die Verwendung des Geoinformationssystems erleichtern und Anregungen zur Weiterentwicklung und Pflege des Geoinformationssystems bieten.

Das Themengebiet der Darstellung geocodierter Daten bietet weiterhin Möglichkeiten im Bezug auf Gestaltung und Umsetzung diverser Informationen, um Anwendern eine hilfreiche und unterstützende Web-basierte Plattformen bereitzustellen.

# 6 Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
ΑΡΙ	Application Programming Interface
CMS	Content Management System
CSS	Cascading Style Sheets
DB	Datenbank
DBMS	Datenbank management system
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GPX	GPS Exchange Format
HTML	Hypertext Mark-Up Language
НТТР	Hypertext Transfer Protocol
ID	Identitätsnummer
IIS	Internet Information Service
ISO	International Organization for Standardization
ITS	Informationssystem und System-Management
JPEG	Joint Photographic Experts Group
JPG	Joint Photographic Group
KML	Keyhole Markup Language
LAMP	Linux Apache MySQL PHP
LTS	Long Term Support
PHP	Hypertext Preprocessor
PNG	Portable Network Graphics
SQL	Structured Query Language
SSH	Secure Shell
ТСР	Transmission Control Protocol
Telnet	Telecommunication Network
TIFF	Tagged Image File Format
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
WAMP	Windows Apache MySQL PHP
WYSIWYG	What You See Is What You Get
ХАМР	(Betriebssystem) Apache MySQL PHP
ХАМРР	(Betriebssystem) Apache MySQL PHP PERL

# 7 Abbildungsverzeichnis

# Abbildungsverzeichnis

1	Google Maps Benutzeroberfläche	3
2	Google Maps Geländedarstellung	4
3	Geländekartenausschnitt <i>KML</i> -Route Gaisberg	6
4	CMS-Vergleich mit CMS-Matrix	9
5	Lebenszyklen und Supportdauer für Ubuntu Server Editionen [11]	20
6	Marktanteil Apache [43]	21
7	Formular zur Eingabe von Informationen einer Bildungsstätte	31
8	UML-Diagramm setMarkers	32
9	Ausgabebeispiel der Funktion getMarkers	34
10	Informationen zur FH-Salzburg	35
11	UML-Diagramm getMarkers	36
12	Tabelleneinträge verschiedener Routen	38
13	Verschieden Routen in Chicago (USA)	40
14	Modul zur Auswahl bestimmter Bildungsstätten	43
15	Modul zum Bearbeiten der Kategorien Bildungsstätten	46
16	Höhen-, Steigungs- und Geschwindigkeitsangaben einer $GXP$ -Datei $\ldots$	47
17	Höhendiagramm GXP-Datei	48
18	Geschwindigkeitsdiagramm GXP-Datei	48
19	Steigungsdiagramm $GXP$ -Datei	48
20	Datenbank mit <i>phpMyAdmin</i> exportieren	50
21	Installationsoberfläche von XAMPP	67
22	Abschließender Installationsschritt von XAMPP	68
23	XAMPP Control Panel Application	68
24	Joomla! 1.5.15 Dateistruktur	86
25	Webauftritte auf einem Webserver	87
26	Aktivierte Module	89
27	Globale Beitragskonfiguration	92
28	Beitrag getMarkers erstellen	92
29	Einstellungen Benutzermenü	94
30	Ausschnitt aktivierter Module	95
31	Kategorie-Icon Fachhochschule	96

# Listings

Seitenbeschreibungssprache HTML
HTML-Dokument mit CSS-Definitionen 12
XML-Datei des Joomla!-Modul "Who's Online"
Mikroformat-Beispiel für die FH-Salzburg 13
Unterschied $HTML$ zu $XHTML$ [41]
Eingebetteter <i>PHP</i> -Code in einem <i>HTML</i> -Dokument
Datenbankzugriff mit <i>PHP</i> 16
E-Mail-Formular Validierung mit <i>JavaScript</i>
Inhalte einer Datei mit $AJAX$ laden $\ldots$ 18
Verwendung von Division-Elementen
Formatieren eines Division-Elements
Einbinden der <i>Joomla!</i> -Module
Einbinden von Quellcode in den Inhalt von <i>Joomla!</i>
Entfernen der Editier-Funktion
Google Map einbinden
Funktion createMarker
Beispiel gesendeter XML-Daten der Funktion retrieveMarkers.php
Hochladen der Route
Wegpunkte in KML-Datei schreiben
Extrahieren der Startkoordinaten
Eintragen der Daten in die Datenbank
Tabelleninhalte aus Datenbank auslesen    39
KML-Datei laden
Platzieren der Route und des Markers 40
Beispiel gesendeter XML-Daten der Funktion retrieveRoutes.php 41
Auslesen der Kategorien aus der Datenbank
Ausgabe der Kategorien als Hyperlink 43
Auszug aus dem Script $map\_functions.js$ von Kapitel 3.4.2
Sicherheitsabfrage vor dem Löschen einer Kategorie
Klassenfunktion <i>delCategory</i> zum Löschen einer Kategorie
Neuen Kategorienamen hinzufügen 45
Notwendige Anpassungen <i>configuration.php</i> -Datei
Root-Passwort setzen
Administrationsrechte zur Konfiguration des Web-Servers
Statische IP-Adresse zuweisen
Angepasster Inhalt der Datei <i>interfaces</i>
Netzwerkinterface neu starten
hosts-Datei zur Bearbeitung öffnen
Angepasster Inhalt der Datei <i>hosts</i>

40	Hostnamen setzen
41	Hostnamen übernehmen
42	Hostname - erste Kontrolle
43	Hostname - zweite Kontrolle
44	sources.list bearbeiten
45	Paketquellen anpassen
46	Nach neuen Updates suchen
47	Updates installieren
48	Web-Server Packages installieren
49	DNS-Server installieren
50	DNS-Server stoppen
51	DNS-Server stoppen
52	Angepasster Inhalt der Datei <i>bind9</i>
53	Benötigte Verzeichnisse anlegen und Berechtigungen setzen
54	Datei syslogd bearbeiten
55	Angepasster Inhalt der Datei syslogd
56	DNS-Server starten
57	MySQL installieren
58	<i>my.cnf</i> -Datei zur Bearbeitung öffnen
59	Zeile <i>localhost</i> auskommentieren
60	MySQL neu starten
61	Mail-Server Postfix installieren    78
62	Post fix neu starten
63	Konfiguration von <i>Postfix</i>
64	Zertifikate für $TLS$ -Verschlüsselung
65	Hostnamen an <i>Postfix</i> mitteilen
66	Postfix konfigurieren
67	Datei main.cf bearbeiten 80
68	Angepasster Inhalt der Datei <i>main.cf</i>
69	Authentifizierung saslauthd
70	Datei saslauthd bearbeiten
71	Angepasster Inhalt der Datei saslauthd
72	Postfix-Benutzer einrichten
73	Postfix und sasl neu starten
74	Courier-IMAP/Courier-POP3 installieren
75	Mailversand an Benutzer
76	Apache installieren
77	Datei dir.conf bearbeiten
78	Angepasster Inhalt der Datei dir.conf
79	Apache-Module aktivieren

80	Apache neu starten
81	Webalizer installieren
82	System-Uhr synchroniseren
83	Google Maps-API Key einfügen
84	Datenbankverbindung einrichten 90
85	Tabellenstruktur für Tabelle $db\_category$
86	Tabellenstruktur für Tabelle $db_geoweb$
87	Beitragsinhalt <i>setMarkers</i>
88	Google Map-API Key
89	Datenbankzugang konfigurieren
90	Verbindung zur Datenbank aufbauen
91	getMarkers index.php
92	getMarkers Kartenfunktionen
93	getMarkers XML-Daten auslesen
94	setMarkers index.php
95	setMarkers Kartenfunktionen 105
96	setMarkers Overlay für Eingabemaske 107
97	setMarkers Eingabemaske
98	setMarkers EXIF-Daten auslesen 109
99	setMarkers Uplaod-Formular 110
100	setMarkers Funktionen Upload-Formular
101	getRoutes index.php
102	getRoutes Kartenfunktionen
103	getRoutes XML-Daten auslesen
104	getRoutes GPX-Dateien konvertieren
105	setRoutes index.php
106	setRoutes Kartenfunktionen
107	setRoutes Overlay für Eingabemaske 139
108	setRoutes Eingabemaske
109	setRoutes Uplaod-Formular
110	getGPXfile index.php
111	getGPXfile index.php
112	getKMLfile index.php
113	$mod\_edu \ index.html$
114	$mod_edu$ Modul Auswahl Bildungsstätten
115	$mod\_edu$ Modul Klassenfunktionen
116	$mod\_edu$ Modulbeschreibung
117	$mod\_edu\_cat index.html$
118	$mod\_edu\_cat$ Modul Auswahl Bildungsstätten
119	mod_edu_cat Modul Klassenfunktionen

120	$mod\_edu\_$	cat Modulbeschreibung																		188
-----	--------------	-----------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

### 8 Literaturverzeichnis

- [1] Apache Software Fd.: Unterstützung namensbasierter virtueller Hosts Apache HTTP Server.
   http://httpd.apache.org/docs/2.1/de/vhosts/name-based.html (09.05.2010).
- [2] Apache Software Fd.: Welcome! The Apache HTTP Server Project. http://httpd.apache.org/ (18.02.2010).
- [3] Berkemeier J.: GPX Viewer. http://www.j-berkemeier.de/GPXViewer/ (10.05.2010).
- [4] Bleich, H.: Google Maps in den USA gestartet. http://www.heise.de/newsticker/meldung/Google-Maps-in-den-USA-gestartet-133243.html (05.05.2010).
- [5] Canonical Ltd.: ApacheMySQLPHP. https://help.ubuntu.com/community/ApacheMySQLPHP (08.05.2010).
- [6] Canonical Ltd.: Configuration. https://help.ubuntu.com/8.10/serverguide/C/dns-configuration.html (15.02.2010).
- [7] Canonical Ltd.: Courier. https://help.ubuntu.com/community/Courier (25.02.2010).
- [8] Canonical Ltd.: DNS-Server Bind/Erweiterte Konfiguration. http://wiki.ubuntuusers.de/DNS-Server\_Bind/Erweiterte\_Konfiguration (08.05.2010).
- [9] Canonical Ltd.: HTTPD Apache2 Web Server. https://help.ubuntu.com/8.04/serverguide/C/httpd.html (23.02.2010).
- [10] Canonical Ltd.: IsoBrennen. https://help.ubuntu.com/community/IsoBrennen (10.02.2010).
- [11] Canonical Ltd.: Maintenance policy and life-cycle: Ubuntu Server Edition. http://www.ubuntu.com/products/whatisubuntu/serveredition/benefits/ lifecycle (07.05.2010).
- [12] Canonical Ltd.: MYSQL5FromSource. https://help.ubuntu.com/community/MYSQL5FromSource (16.02.2010).
- [13] Canonical Ltd.: Official Ubuntu Documentation. https://help.ubuntu.com/ (25.02.2010).
- [14] Canonical Ltd.: OpenSSH Server. https://help.ubuntu.com/7.04/server/C/openssh-server.html (08.05.2010).

- [15] Canonical Ltd.: Postfix. https://help.ubuntu.com/community/Postfix (25.02.2010).
- [16] Canonical Ltd.: Time Synchronisation with NTP. https://help.ubuntu.com/8.04/serverguide/C/NTP.html (12.02.2010).
- [17] Canonical Ltd.: Ubuntu and Debian / Ubuntu. http://www.ubuntu.com/community/ubuntustory/Debian (08.05.2010).
- [18] Canonical Ltd.: Ubuntu Home Page. http://www.ubuntu.com/ (09.02.2010).
- [19] Canonical Ltd.: Ubuntu Manpage: sudo\_root How to run administrative commands. http://manpages.ubuntu.com/manpages/hardy/man8/sudo\_root.8.html (12.02.2010).
- [20] Canonical Ltd.: Ubuntu Packages Search. http://packages.ubuntu.com/ (11.02.2010).
- [21] Canonical Ltd.: UserDocumentation. https://help.ubuntu.com/community (25.02.2010).
- [22] Canonical Ltd.: webalize package in Ubuntu. https://launchpad.net/ubuntu/+source/webalizer (18.03.2010).
- [23] Ebersbach, A., Glaser, M., & Kubani, R.: Joomla! 1.5 Das umfassende Handbuch. http://openbook.galileocomputing.de/joomla15/ (11.02.2010).
- [24] Fleischhacker, K.: BundesGymnasiumZaunergasse. http://www.bgzaunergasse.at/ (12.05.2010).
- [25] Free Software Foundation, Inc.: The Free Software Definition GNU Project Free Software Foundation (FSF). http://www.gnu.org/philosophy/free-sw.html (06.05.2010).
- [26] Google Inc.: Datenschutz-Center. http://www.google.com/privacy\_faq.html (05.05.2010).
- [27] Google Inc.: Google Maps. http://maps.google.com/ (02.02.2010).
- [28] Google Inc.: Google Maps-API Beispiele. http://code.google.com/intl/de/apis/maps/documentation/examples/index. html (05.05.2010).
- [29] Google Inc.: Google Maps-API Demogalerie. http://code.google.com/intl/de/apis/maps/documentation/demogallery.html (05.05.2010).

- [30] Google Inc.: Google Maps-API Referenz. http://code.google.com/intl/de/apis/maps/documentation/reference.html (05.05.2010).
- [31] Google Inc.: Google Maps-API-Konzepte. http://code.google.com/intl/de-DE/apis/maps/documentation/ (04.05.2010).
- [32] Google Inc.: Google Maps-API-Konzepte. http://code.google.com/intl/de-DE/apis/maps/documentation/ (05.05.2010).
- [33] Google, Inc.: Nutzungsbedingungen für Google Maps. http://www.google.com/intl/de/help/terms\_maps.html (05.05.2010).
- [34] Groundspeak, Inc.: Geocaching The Official Global GPS Cache Hunt Site. http://www.geocaching.com/ (04.02.2010).
- [35] Hwang, D.: Happy Turkey Day. http://googlemapsapi.blogspot.com/2005/11/happy-turkey-day.html (05.05.2010).
- [36] ISO: ISO International Organization for Standardization. http://www.iso.org/iso/home.html (14.05.2010).
- [37] Krüger, A.: Tim Berners-Lee. http://www.heise.de/tp/r4/artikel/16/16446/1.html (22.04.2010).
- [38] Loibl, C.: Sparkling Science. http://www.sparklingscience.at/ (12.05.2010).
- [39] MapQuest, Inc.: MapQuest Maps Driving Directions Map. http://www.mapquest.com/ (05.05.2010).
- [40] Microsoft Corp.: Bing Maps. http://www.bing.com/maps/ (05.05.2010).
- [41] Münz, S.: Wenn HTML zu XHTML wird. http://de.selfhtml.org/html/xhtml/anzeige/beispiel.xhtml (10.05.2010).
- [42] National Maritime Museum : History of the Royal Observatory : Our history in Greenwich : About us : NMM. http://www.nmm.ac.uk/about/history/royal-observatory/ (14.05.2010).
- [43] Netcraft Ltd.: Web Server Survey Archives. http://news.netcraft.com/archives/web\_server\_survey.html (09.05.2010).
- [44] Olson, P.: PHP: Funktionsreferenz. http://php.net/manual/de/funcref.php (11.05.2010).
- [45] Olson, P.: PHP: Klassen und Objekte (PHP 5). http://de3.php.net/manual/de/language.oop5.php (11.05.2010).

- [46] OpenStreetMap und Mitwirkende, CC-BY-SA: Getting The Source OpenStreetMap Wiki. http://wiki.openstreetmap.org/wiki/Getting\_The\_Source (04.05.2010).
- [47] OpenStreetMap und Mitwirkende, CC-BY-SA: *OpenStreetMap Die freie Wiki-Weltkarte*. http://www.openstreetmap.org/ (04.05.2010).
- [48] Oracle Corp.: MySQL :: MySQL 5.1 Referenzhandbuch :: 1.4 Was ist MySQL? http://dev.mysql.com/doc/refman/5.1/de/what-is.html (11.05.2010).
- [49] Oracle Corp.: MySQL :: MySQL 5.1 Referenzhandbuch :: 1.4.4 Wie groß können MySQL-Tabellen sein? http://dev.mysql.com/doc/refman/5.1/de/table-size.html (08.05.2010).
- [50] Pritchard, S.: Linux Hardware Compatibility HOWTO. http://www.tldp.org/HOWTO/Hardware-HOWTO/ (10.02.2010).
- [51] Seidler, K.: apache friends xampp. http://www.apachefriends.org/ (27.04.2010).
- [52] Smith J.: The CMS Matrix cmsmatrix.org The Content Management Comparison Tool. http://www.cmsmatrix.org/ (04.02.2010).
- [53] Thoma, D.: JavaScript: Drag and Drop. http://aktuell.de.selfhtml.org/artikel/javascript/draganddrop/ (27.04.2010).
- [54] Thoma, D.: Namespace GXmlHttp. http://code.google.com/intl/de/apis/maps/documentation/reference.html (27.04.2010).
- [55] Timme, F.: The Perfect Server Ubuntu Hardy Heron (Ubuntu 8.04 LTS Server). http://www.howtoforge.com/perfect-server-ubuntu8.04-lts (10.02.2010).
- [56] W3C: About W3C. http://www.w3.org/Consortium/ (27.04.2010).
- [57] W3C: Cascading Style Sheets Home Page. http://www.w3.org/Style/CSS/ (04.02.2010).
- [58] W3C: GPX nach KML Konverter « Stupid Coder's Random Stuff. http://wp1173987.wp090.webpack.hosteurope.de/wordpress/?page\_id=661 (20.04.2010).
- [59] W3C: *HTML5*. http://dev.w3.org/html5/spec/Overview.html (20.04.2010).
- [60] W3C: *HTTP Hypertext Transfer Protocol Overview*. http://www.w3.org/Protocols/ (12.05.2010).

- [61] W3C: RFC 959: File Transfer Protocol. http://www.w3.org/Protocols/rfc959/ (12.05.2010).
- [62] Wenz, C.: JavaScript und AJAX Das umfassende Handbuch. http://openbook.galileocomputing.de/javascript\_ajax/ (08.05.2010).
- [63] Yahoo! Inc.: Yahoo! Maps. http://maps.yahoo.com/ (05.05.2010).
# 9 Anhang

In diesem Abschnitt sind die einzelnen Installations- und Einrichtungsschritte der verschiedenen Softwarepakete aufgezeigt:

- XAMPP
- Ubuntu Server 8.04 LTS
- Joomla!

# 9.1 XAMPP

Eine neu erstellte Webseite sollte in jedem Falle getestet werden. Es ist von Vorteil, sich eine lokale Testplattform einzurichten. Auf diese Weise ist es möglich, alle Funktionalitäten und das Layout auf Fehler hin zu überprüfen, bevor die Webseite "online" geht.

Als Testumgebung empfiehlt sich XAMPP [51]. Dieses lokale Webserver-Komplettpaket lässt sich auf den Betriebssystemen  $Linux^{20}$ ,  $Windows^{21}$ ,  $Mac \ OS \ X^{22}$ , sowie  $Solaris^{23}$  einsetzen und ist kostenlos erhältlich unter http://www.apachefriends.org/de/xampp.html.

# XAMPP-Installation

Zum Erstellungszeitpunkt dieser Arbeit lag das Installationspaket von XAMPP in der Version 1.7.3 für das Betriebssystem *Windows* vor. Die Datei *xampp-win32-1.7.3.exe* starten und als *Destination Folder* die Laufwerkskennung *C:*\, wie aus Abbildung 21 ersichtlich, eingeben und die Schaltfläche *Install*.

<sup>&</sup>lt;sup>20</sup>http://de.wikipedia.org/wiki/Linux

<sup>&</sup>lt;sup>21</sup>http://www.microsoft.com/

<sup>&</sup>lt;sup>22</sup>http://www.apple.com/de/macosx/

<sup>&</sup>lt;sup>23</sup>http://de.sun.com/practice/software/solaris/



Abbildung 21: Installationsoberfläche von XAMPP

Es wird empfohlen, den vorgeschlagenen *Destination folder* der Installationsroutine beizubehalten. Aufgrund von Einschränkungen bezüglich Lese- und Schreibrechten unter dem Betriebssystem *Windows* ist als Installationspfad  $C:\Programme$  keineswegs zu empfehlen.

Die weiteren Frage sind wie folgt zu beantworten und mit der Return-Taste zu bestätigen:

Should I add shortcuts to the startmenu/desktop? (y/n):  $\triangleright y$ 

Should I locate the XAMPP paths correctly? Should I proceed?  $(y/x=exit \ setup): \triangleright y$ 

Should I make a portable XAMPP without drive letters? Your choice? (y/n):  $\triangleright n$ 

Den nächsten Schritt durch Drücken der *Return*-Taste, ebenso wie die Abfrage der *timezone* bestätigen. Abbildung 22 zeigt den abschließenden Installationsschritt.



Abbildung 22: Abschließender Installationsschritt von XAMPP

Dieser wird mit der Eingabe des Zeichens x, gefolgt von der Return-Taste, beendet.

Mittels Doppelklick auf das Desktop-Icon XAMPP Control Panel kann die Oberfläche zur Steuerung des lokalen Webservers (siehe Abbildung 23) gestartet werden.

XAMPP Control Panel Sh (Apache Friends Edition) Set	e <u>l</u> l up				
Modules Port-C	heck				
Svc Apache Start Admin Expl	ore				
Svc MySql Start Admin SC	M				
Svc FileZilla Start Admin <u>R</u> efr	esh				
Svc Mercury Start Admin	lp				
Svc Tomcat Start Admin	it				
XAMPP Control Panel Version 2.5.8 (2009-07-28) XAMPP for Windows Version 1.7.3 Windows 6.1 Build 7600 Platform 2 Current Directory: C:\xampp Status Check OK					

Abbildung 23: XAMPP Control Panel Application

Abschließend jeweils auf die beiden Schaltflächen Start in den Zeilen Apache, sowie MySQL mit der linken Maustaste klicken.

Der lokale Webserver XAMPP ist jetzt vollständig konfiguriert und einsatzbereit.

Das Verzeichnis, aus welchem heraus die Webseiten geladen werden, befindet sich unter  $C:\xamphtdocs$ .

Die Web-basierte Administrationsoberfläche von XAMPP ist unter der URL http://localhost/ xampp/ erreichbar. Durch einen Klick mit der linken Maustaste auf die Schaltfläche Admin der Zeile Apache (siehe Abbildung 23) erzielt denselben Effekt.

Das ebenfalls Web-basierte Werkzeug *phpMyAdmin* zur Administration der lokalen *MySQL*-Datenbank ist über die Schaltfläche *Admin* der Zeile *MySql* des *XAMPP Control Panels* startbar. Alternativ über die URL http://localhost/phpmyadmin/ im Browser aufrufbar.

# 9.2 Ubuntu Server 8.04 LTS

#### Ubuntu Server - Installation als Webserver

Voraussetzung zur Installation des Betriebssystem *Ubuntu* [18] ist ein Computer, welcher als Webserver fungieren soll. Auf der eingebauten Festplatte sollten sich keine wichtigen oder benötigten Dateien mehr befinden. Die Kapazität des Festplattenlaufwerkes, sowie des Arbeitsspeichers, sollte die jeweilige Mindestgröße von einem Gigabyte betragen. Genaue Hardwarevoraussetzungen lassen sich im *Linux Hardware Compatibility HOWTO* [50] in Erfahrung bringen.

### Ubuntu 8.04 LTS

Das Serverbetriebssystem Ubuntu 8.04 LTS steht auf der Webseite http://www.ubuntu.com/ GetUbuntu/download als 32-bit und 64-bit Version zum Download unter den alternativen Optionen Alternative download options, including Ubuntu installer for Windows bereit. Mit Hilfe eines Brennprogrammes ist die heruntergeladene Datei, die im ISO-Format [10] vorliegt, auf einen CD-Rohling zu brennen. Eine Installationshilfe für einen Ubuntu-Webserver [55] ist verfügbar, wurde jedoch für die Zwecke dieser Arbeit verändert und der Netzwerkstruktur der Fachhochschule Salzburg angepasst.

### Vorbereitungen zur Installation

Die im vorherigen Schritt erstellte Installations-CD einlegen und von dieser CD im Ladevorgang des Rechners starten. Startet der Computer nicht die Installationsroutine der CD, ist im sogenannten *BIOS* die Boot-Reihenfolge zu verändern, um von der eingelegten Server-CD laden zu können.

Auf welche Weise das *BIOS* aufgerufen werden kann, ist abhängig vom Hersteller des Rechners und der verwendeten *BIOS*-Version. Das Handbuch des Computerherstellers oder eine Recherche auf dessen Webseite gibt zu dieser Frage Aufschluss.

### Installation Ubuntu 8.04 LTS

Startet der Server von CD, so ist als erstes die Sprache auszuwählen, in welcher das Installationsmenü angezeigt werden soll. Sodann wird der oberste Menüpunkt angewählt, welcher, für den Fall, dass *Deutsch* als Installationssprache gewünscht wurde, *Ubuntu Server installieren* lautet.

Wiederum wird nach einer Sprachauswahl gefragt, mit deren Hilfe durch die Betriebssysteminstallation geführt wird, gefolgt von der Angabe des Landes und dem Tastaturlayout.

Bei der Frage nach dem Hostnamen des Servers wurde in dieser Arbeit maps gewählt.

Der weitere Installationsschritt ist auf die Aufgabe angepasst, die im Rahmen dieser Arbeit erstellten Projekte und Webseiten zu verwalten und im mittels einer Internetverbindung Benutzern zur Verfügung zu stellen. Aus diesem Grund wurde auf eine manuelle Einteilung der Festplatte verzichtet. Es wurde die *Geführt - gesamte Festplatte verwenden* Option gewählt.

Der Warnhinweis, dass alle bisherigen Daten auf der Festplatte unwiderruflich gelöscht werden,

wird bestätigt. Die vorgeschlagene Partitionierung mit Ja beantwortet.

Das Serverbetriebssystem Ubuntu 8.04 LTS wird auf die Festplatte installiert.

### Benutzerdaten

Der Name für den Benutzer [19] des Serversystems ist einzugeben. Als Passwort ist hinsichtlich der Sicherheit eine Kombination aus Zeichen, Ziffern und Sonderzeichen bestehende Zeichenkette, sowie im nachfolgenden Schritt erneut einzutragen.

# HTTP Proxy

Der Server soll nicht als Proxy fungieren. Daher wird dieses Eingabefeld leer gelassen.

### Softwareauswahl

Das Auswahlmenü bietet in diesem Schritt die Installation als *LAMP Server* an. *LAMP* ist ein Acronym für *Linux-Apache-MySQL-PHP*. Dennoch wird die Option *OpenSSH Server* ausgewählt und mit *Weiter* bestätigt. *SSH* bietet die Möglichkeit, den Server von jedem beliebigen Standort über eine verschlüsselte Verbindung zu administrieren.

# Root-Konto

Rechte für Änderungen am System behält die Benutzerverwaltung von Linux-Systemen dem Benutzernamen *root* [19] vor. Damit die nachfolgenden Eingriffe in das System durchgeführt werden können, muss das *Root-Konto* aktiviert und mit einem Passwort versehen werden. Hierzu den Befehl

sudo passwd root

Listing 33: Root-Passwort setzen

eingeben.

# Root-Rechte

Ab diesem Zeitpunkt ist es jederzeit möglich, mit der Eingabe von

sudo su

Listing 34: Administrationsrechte zur Konfiguration des Web-Servers

Administratorrechte [19] auf dem Server zu erlangen, um Änderungen am System vorzunehmen.

### Netzwerkkonfiguration

Einem Server, dies schließt einen Web-Server mit ein, sollte stets eine statische IP-Adresse zugeordnet werden. Mit der Eingabe von

```
nano /etc/network/interfaces
```

Listing 35: Statische IP-Adresse zuweisen

lassen sich die Netzwerkeinstellungen bearbeiten. Im nachfolgenden Beispiel sind address, netmask, network, broadcast und gateway an das eigene Netzwerk anzupassen.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
# Configuration for maps.fh-salzburg.ac.at
auto eth0
iface eth0 inet static
    address 193.170.119.150
    netmask 255.255.255.128
    network 193.170.119.128
    broadcast 193.170.119.255
    gateway 193.170.119.129
```

Listing 36: Angepasster Inhalt der Datei interfaces

Anschließend das Netzwerk neu starten mit dem Befehl:

```
/etc/init.d/networking restart
```

Listing 37: Netzwerkinterface neu starten

Die hosts-Datei editieren:

nano / etc / hosts

Listing 38: hosts-Datei zur Bearbeitung öffnen

Die folgenden Änderungen vornehmen:

```
127.0.0.1 localhost.localdomain localhost
192.168.0.100 maps.fh-salzburg.ac.at maps
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
```

ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

Listing 39: Angepasster Inhalt der Datei hosts

Nach dem Editieren der hosts-Datei sind die beiden Befehlszeilen einzugeben:

echo maps.fh-salzburg.ac.at > /etc/hostname

Listing 40: Hostnamen setzen

#### und

/ etc/init.d/hostname.sh start

Listing 41: Hostnamen übernehmen

Wurden bis jetzt alle Einstellungen und Änderungen korrekt vorgenommen, lautet das Resultat bei Eingabe von

hostname

Listing 42: Hostname - erste Kontrolle

und

hostname -f

Listing 43: Hostname - zweite Kontrolle

jeweils maps.fh-salzburg.ac.at.

# Ubuntu Installation aktualisieren

Bevor für die Ubuntu-Version eine Aktualisierung durchgeführt werden kann, ist die *sources.list* anzupassen:

nano / etc/apt/sources.list

Listing 44: *sources.list* bearbeiten

Der Inhalt der *sources.list* ist auf diese Weise anzupassen:

#

# deb cdrom:[Ubuntu-Server 8.04 \_Hardy Heron\_ - Release i386 (20080423.2)
]/ hardy main restricted

#deb cdrom:[Ubuntu-Server 8.04 Hardy Heron - Release i386 (20080423.2)]/ hardy main restricted # See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to # newer versions of the distribution. deb http://de.archive.ubuntu.com/ubuntu/ hardy main restricted deb-src http://de.archive.ubuntu.com/ubuntu/ hardy main restricted ## Major bug fix updates produced after the final release of the ### distribution. deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted ## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu ## team, and may not be under a free licence. Please satisfy yourself as to ## your rights to use the software. Also, please note that software in ## universe WILL NOT receive any review or updates from the Ubuntu security ## team. deb http://de.archive.ubuntu.com/ubuntu/ hardy universe deb-src http://de.archive.ubuntu.com/ubuntu/ hardy universe deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe ## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu  $\#\!\!\#$  team, and may not be under a free licence. Please satisfy yourself as to ## your rights to use the software. Also, please note that software in ## multiverse WILL NOT receive any review or updates from the Ubuntu ## security team. deb http://de.archive.ubuntu.com/ubuntu/ hardy multiverse deb-src http://de.archive.ubuntu.com/ubuntu/ hardy multiverse deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse ## Uncomment the following two lines to add software from the 'backports' ## repository. ## N.B. software from this repository may not have been tested as  $\#\!\#$  extensively as that contained in the main release, although it includes ## newer versions of some applications which may provide useful features. ### Also, please note that software in backports WILL NOT receive any review

## or updates from the Ubuntu security team.

- # deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-backports main
   restricted universe multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository. This software is not part of Ubuntu, but is
## offered by Canonical and the respective vendors as a service to Ubuntu
## users.

```
\# \text{ deb } \text{http://archive.canonical.com/ubuntu hardy partner}
```

# deb-src http://archive.canonical.com/ubuntu hardy partner

deb http://security.ubuntu.com/ubuntu hardy-security main restricted deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted deb http://security.ubuntu.com/ubuntu hardy-security universe deb-src http://security.ubuntu.com/ubuntu hardy-security universe deb http://security.ubuntu.com/ubuntu hardy-security multiverse deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse

Listing 45: Paketquellen anpassen

Nach den Änderungen ist die Paketdatenbank auf den neuesten Stand zu bringen:

apt-get update

Listing 46: Nach neuen Updates suchen

Die Updates installieren, falls verfügbar:

apt-get upgrade

Listing 47: Updates installieren

### Software-Pakete

Der nachfolgende Befehl installiert benötigte Pakete [20], welche zum Betrieb eines Webservers benötigt werden.

apt-get install binutils cpp fetchmail flex gcc libarchive-zip-perl libc6dev libcompress-zlib-perl libdb4.3-dev libpcre3 libpopt-dev lynx m4 make ncftp nmap openssl perl perl-modules unzip zip zlib1g-dev autoconf automake1.9 libtool bison autotools-dev g++ build-essential

# **DNS Server**

Für das DNS-Server Installationspaket [6] eingeben:

apt-get install bind9

Listing 49: DNS-Server installieren

Den DNS-Server-Dienst stoppen, um die empfohlenen Einstellungen zu treffen [8].

/ etc/init.d/bind9 stop

Listing 50: DNS-Server stoppen

Die Datei *bind9* öffnen:

nano / etc / default / bind9

Listing 51: DNS-Server stoppen

Den Inhalt abändern:

OPTIONS="-u bind -t /var/lib/named" \# Set RESOLVCONF=no to not run resolvconf RESOLVCONF=yes

Listing 52: Angepasster Inhalt der Datei bind9

### Benötigte Verzeichnisse

Diese Eingabe erstellt für DNS-Dienst erforderliche Ordner, verschiebt das *config*-Verzeichnis und beugt Fehlermeldungen vor, falls *bind* aktualisiert wird. Desweiteren werden Berechtigungen an den relevanten Verzeichnissen vorgenommen.

```
mkdir -p /var/lib/named/etc
mkdir /var/lib/named/dev
mkdir -p /var/lib/named/var/cache/bind
mkdir -p /var/lib/named/var/run/bind/run
mv /etc/bind /var/lib/named/etc
ln -s /var/lib/named/etc/bind /etc/bind
mknod /var/lib/named/dev/null c 1 3
mknod /var/lib/named/dev/random c 1 8
chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random
chown -R bind:bind /var/lib/named/var/*
```

chown -R bind: bind /var/lib/named/etc/bind

Listing 53: Benötigte Verzeichnisse anlegen und Berechtigungen setzen

Wichtige Mitteilungen sollten von den System Logs weiterhin erfasst werden. Dazu die Datei

nano / etc/default/syslogd

Listing 54: Datei syslogd bearbeiten

editieren und anpassen:

SYSLOGD="-a /var/lib/named/dev/log"

Listing 55: Angepasster Inhalt der Datei syslogd

Den Logging-Daemon, sowie den DNS-Server neu starten:

/etc/init.d/sysklogd restart
/etc/init.d/bind9 start

Listing 56: DNS-Server starten

### MySQL

Das Paket MySQL [12] installieren:

apt-get install mysql-server mysql-client libmysqlclient15-dev

Listing 57: MySQL installieren

Ein Passwort für den MySQL-Root-Benutzer eingeben.

Damit MySQL nicht nur auf *localhost* reagiert, ist in der *my.cnf* 

nano / etc/mysql/my.cnf

Listing 58: my.cnf-Datei zur Bearbeitung öffnen

die Zeile bind-address = 127.0.0.1 auszukommentieren:

 $\forall \# bind-address = 127.0.0.1$ 

Listing 59: Zeile *localhost* auskommentieren

und MySQL neu starten:

/etc/init.d/mysql restart

Listing 60: MySQL neu starten

### Postfix

Um den Mail-Server [15] zu installieren:

```
apt-get install postfix libsasl2-2 sasl2-bin libsasl2-modules procmail
```

Listing 61: Mail-Server Postfix installieren

Für General type of mail configuration, sowie System mail name jeweils maps.fh-salzburg.ac.at eingeben.

dpkg-reconfigure postfix

### Listing 62: Postfix neu starten

Die daraufhin gestellten Fragen wie folgt beantworten:

*General type of mail configuration: >Internet Site* 

 $System mail name: \triangleright maps.fh-salzburg.ac.at$ 

Root and postmaster mail recipient:  $\triangleright$  [blank]

 $Other \ destinations \ to \ accept \ mail \ for \ (blank \ for \ none): \triangleright maps. fh-salzburg. ac. at, \ localhost. fh-salzburg. ac. at, \ localhost. localhost \ localhost$ 

Force synchronous updates on mail queue?  $\triangleright$  No

Local networks:  $\triangleright 127.0.0.0/8$ 

Use procmail for local delivery?  $\triangleright$  Yes

Mailbox size limit (bytes):  $\triangleright 0$ 

Local address extension character:  $\triangleright$  +

Internet protocols to use:  $\triangleright$  all

*Postfix* konfigurieren mit:

```
postconf -e 'smtpd_sasl_local_domain ='
postconf -e 'smtpd_sasl_auth_enable = yes'
postconf -e 'smtpd_sasl_security_options = noanonymous'
postconf -e 'broken_sasl_auth_clients = yes'
```

```
postconf -e 'smtpd_sasl_authenticated_header = yes'
postconf -e 'smtpd_recipient_restrictions = permit_sasl_authenticated,
    permit_mynetworks,reject_unauth_destination'
postconf -e 'inet_interfaces = all'
echo 'pwcheck_method: saslauthd' >> /etc/postfix/sasl/smtpd.conf
echo 'mech_list: plain login' >> /etc/postfix/sasl/smtpd.conf
```

### Listing 63: Konfiguration von Postfix

Die Zertifikate für die TLS-Verschlüsselung erstellen:

```
mkdir /etc/postfix/ssl
cd /etc/postfix/ssl/
openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
chmod 600 smtpd.key
openssl req -new -key smtpd.key -out smtpd.csr
openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out smtpd.
crt
openssl rsa -in smtpd.key -out smtpd.key.unencrypted
mv -f smtpd.key.unencrypted smtpd.key
openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem
-days 3650
```

Listing 64: Zertifikate für TLS-Verschlüsselung

Und Postfix für TLS [15] konfigurieren:

postconf - e 'myhostname = maps.fh-salzburg.ac.at'

Listing 65: Hostnamen an *Postfix* mitteilen

Weitere Einstellungen treffen [15]:

```
postconf -e 'smtpd_tls_auth_only = no'
postconf -e 'smtp_use_tls = yes'
postconf -e 'smtpd_use_tls = yes'
postconf -e 'smtpd_tls_note_starttls_offer = yes'
postconf -e 'smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key'
postconf -e 'smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt'
postconf -e 'smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem'
postconf -e 'smtpd_tls_loglevel = 1'
postconf -e 'smtpd_tls_received_header = yes'
postconf -e 'smtpd_tls_received_header = yes'
```

```
postconf -e 'tls_random_source = dev:/dev/urandom'
```

Listing 66: Postfix konfigurieren

Die Datei main.cf von Postfix

```
nano / etc / postfix / main.cf
```

### Listing 67: Datei main.cf bearbeiten

sollte folgenden Inhalt aufweisen:

```
\# See /usr/share/postfix/main.cf.dist for a commented, more complete
   version
# Debian specific: Specifying a file name will cause the first
\# line of that file to be used as the name. The Debian default
\# is / \text{etc}/\text{mailname}.
\#myorigin = /etc/mailname
smtpd banner = $myhostname ESMTP $mail name (Ubuntu)
biff = no
\# appending .domain is the MUA's job.
append dot mydomain = no
# Uncomment the next line to generate "delayed mail" warnings
\#delay warning time = 4h
readme directory = no
\# TLS parameters
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd tls key file = /etc/postfix/ssl/smtpd.key
smtpd use tls = yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp tls session cache database = btree:{data directory}/smtp scache
\# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
\# information on enabling SSL in the smtp client.
myhostname = maps.fh-salzburg.ac.at
alias maps = hash:/etc/aliases
alias database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = maps.fh-salzburg.ac.at, localhost.fh-salzburg.ac.at,
   localhost.localdomain, localhost
```

```
relayhost =
mynetworks = 127.0.0/8
mailbox_command = procmail -a "$EXTENSION"
mailbox size limit = 0
recipient delimiter = +
inet interfaces = all
inet protocols = all
smtpd sasl local domain =
smtpd sasl auth enable = yes
smtpd sasl security options = noanonymous
broken sasl auth clients = yes
smtpd sasl authenticated header = yes
smtpd recipient restrictions = permit sasl authenticated, permit mynetworks
   , reject unauth destination
smtpd tls auth only = no
smtp use tls = yes
smtp_tls_note_starttls_offer = yes
smtpd tls CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd tls received header = yes
smtpd tls session cache timeout = 3600 \,\mathrm{s}
tls\_random\_source = dev:/dev/urandom
```

Listing 68: Angepasster Inhalt der Datei main.cf

Den Authentifizierungsordner *saslauthd* einrichten [15]:

mkdir -p /var/spool/postfix/var/run/saslauthd

Listing 69: Authentifizierung saslauthd

saslauthd Datei öffen und Konfiguration anpassen:

nano / etc/default/saslauthd

Listing 70: Datei saslauthd bearbeiten

```
#
# Settings for saslauthd daemon
# Please read /usr/share/doc/sasl2-bin/README.Debian for details.
#
# Should saslauthd run automatically on startup? (default: no)
START=yes
```

# Description of this saslauthd instance. Recommended.

```
\# (suggestion: SASL Authentication Daemon)
DESC="SASL Authentication Daemon"
# Short name of this saslauthd instance. Strongly recommended.
# (suggestion: saslauthd)
NAME="saslauthd"
\# Which authentication mechanisms should saslauthd use? (default: pam)
#
\# Available options in this Debian package:
# getpwent --- use the getpwent() library function
\# kerberos5 --- use Kerberos 5
\# pam
            --- use PAM
# rimap
            --- use a remote IMAP server
\# shadow
            -- use the local shadow password file
# sasldb
            -- use the local sasldb database file
# ldap
            --- use LDAP (configuration is in /etc/saslauthd.conf)
#
\# Only one option may be used at a time. See the saslauthd man page
\# for more information.
#
\# Example: MECHANISMS="pam"
MECHANISMS="pam"
# Additional options for this mechanism. (default: none)
\# See the saslauthd man page for information about mech-specific options.
MECH OPTIONS=""
\# How many saslauthd processes should we run? (default: 5)
\# A value of 0 will fork a new process for each connection.
THREADS=5
# Other options (default: -c -m /var/run/saslauthd)
# Note: You MUST specify the -m option or saslauthd won't run!
#
# See /usr/share/doc/sasl2-bin/README.Debian for Debian-specific
   information.
\# See the saslauthd man page for general information about these options.
#
# Example for postfix users: "-c -m /var/spool/postfix/var/run/saslauthd"
#OPTIONS="-c -m /var/run/saslauthd"
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

Listing 71: Angepasster Inhalt der Datei saslauthd

Der Postfix-Benutzer muss zur Gruppe sasl hinzugefügt werden.

adduser postfix sasl

Listing 72: Postfix-Benutzer einrichten

Anschließend *Postfix* und *sasl* neu starten:

/etc/init.d/postfix restart
/etc/init.d/saslauthd start

Listing 73: Postfix und sasl neu starten

### Courier-IMAP/Courier-POP3

Zum Mailversand mit den Paketen Courier-IMAP/Courier-POP3 [15]:

apt-get install courier-authdaemon courier-base courier-imap courier-imapssl courier-pop courier-pop-ssl courier-ssl gamin libgamin0 libglib2 .0-0

Listing 74: Courier-IMAP/Courier-POP3 installieren

Die gestellten Fragen beantworten:

Create directories for web-based administration?  $\triangleright$  No

SSL certificate required  $\triangleright Ok$ 

E-Mails an das Maildir der jeweiligen Benutzers schicken:

postconf -e 'home\_mailbox = Maildir/'
postconf -e 'mailbox\_command ='
/etc/init.d/postfix restart

Listing 75: Mailversand an Benutzer

### Apache PHP5

Die Apache-Webserversoftware [9] installieren:

- apt-get install apache2 apache2-doc apache2-mpm-prefork apache2-utils libexpat1 ssl-cert
- apt-get install libapache2-mod-php5 libapache2-mod-ruby php5 php5-common php5-curl php5-dev php5-gd php5-idn php-pear php5-imagick php5-imap php5-mcrypt php5-memcache php5-mhash php5-ming php5-mysql php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl

Listing 76: Apache installieren

Als nächsten Schritt die Datei dir.conf öffnen:

nano /etc/apache2/mods-available/dir.conf

Listing 77: Datei dir.conf bearbeiten

und wie folgt umändern:

### Listing 78: Angepasster Inhalt der Datei dir.conf

Jetzt werden alle *Apache*-Module aktiviert:

a2enmod ssl a2enmod rewrite a2enmod suexec a2enmod include

## Listing 79: Apache-Module aktivieren

Apache neu starten:

```
/etc/init.d/apache2 force-reload
```

Listing 80: Apache neu starten

## Webalizer

Das Paket Webalizer [22] installieren:

apt-get install webalizer

Listing 81: Webalizer installieren

# System-Uhr

Für eine synchronisierte System-Uhr [16]:

apt-get install ntp ntpdate

Listing 82: System-Uhr synchroniseren

Der Ubuntu 8.04 LTS-Webserver ist jetzt vollständig konfiguriert und einsatzbereit.

# 9.3 Joomla! 1.5

#### Installation des Content Management Systems Joomla! auf einem Webserver

Vorausgesetzt, zur Durchführung nachfolgender Installationsschritte, sind eine vorhandende Internetverbindung, eine FTP-Software, sowie die Benutzernamen und Passwörter für den FTP-Zugang und der Datenbank. Diese Informationen stellt der jeweilige Webhoster zur Verfügung. Wird XAMPP als Webserver lokal eingesetzt, ist nur eine Verbindung zum Internet notwendig.

#### Joomla! Installationspaket

Das Gesamtpaket von Joomla! ist als komprimierte Datei in der aktuellsten Version unter http://www.joomla.de/download-joomla.html erhältlich. Es kann zwischen zwei Gesamtpaketen gewählt werden. Eine Version liegt in englischer Sprache vor. In der zweiten Version liegen die Installationsroutinen, Menüpunkte, Hilfetexte und alle weiteren Text-basierten Inhalte in deutscher Sprache vor. Der englische Zusatz "Stable" im Dateinamen bezeichnet ein vom Entwicklerteam für den produktiven Einsatz getestetes, stabiles freigegebenes Installationspaket. Die nachfolgenden Installationsschritte werden mit der zum Zeitpunkt dieser Arbeit aktuellen Version von Joomla! 1.5.15 Stable erläutert.

Die Datei *Joomla\_1.5.15-Stable-Full\_Package\_German.zip* in einen beliebigen lokalen Ordner des eigenen Arbeitscomputers herunterladen. Abbildung 24 zeigt den Inhalt des *Joomla!* Pakets nach dem Extrahieren. Zum Entpacken der Datei wird ein Komprimierungsprogramm benötigt, beispielsweise der kostenlose WinRAR, erhältlich unter der URL http://www.winrar.de.



Abbildung 24: Joomla! 1.5.15 Dateistruktur

Die in Abbildung 24 zu sehenden Ordner und Dateien werden für eine erfolgreiche Installation auf einem Webserver benötigt und in den dafür vorgesehenen Verzeichnispfad kopiert. Dieser Pfad, aus welchem heraus die spätere Webseite geladen wird, ist vom Provider des Webdienstes vorgegeben. Bei Verwendung des lokalen Webservers XAMPP und den meisten Webhostern lautet der Webordner *htdocs*. Unter der Serverversion von Ubuntu 8.04 LTS /var/www/. Selten werden hierfür die Ordnerbezeichnungen *htdoc* oder *html* verwendet.

Es empfiehlt sich, im Hinblick zusätzlicher Webauftritte, aussagekräftige Projektordner zu erstellen, aus welchen heraus die jeweiligen CMS geladen werden. Für diese Arbeit wurde der Ordnername *edu* verwendet. Abbildung 25 zeigt Beispiele anhand des Webverzeichnispfades *htdocs* von XAMPP.



Abbildung 25: Webauftritte auf einem Webserver

Der gesamte Inhalt des Jommla! Pakets aus Abbildung 25 wird in den Ordner *htdocs/edu/* kopiert. Die Installationsroutine des CMS startet durch den Aufruf von http://www.ihredomain. tld/edu/. Im Fall von XAMPP lautet die URL im Browser http://localhost/edu.

### Joomla! Installation

Hilfreiche und unterstützende Hinweise begleiten durch die sieben notwendigen Installationsabschnitte.

Im ersten Schritt wird die vorgeschlagene Sprache Deutsch durch das Klicken auf den Button namens *Weiter* bestätigt.

Zur Installationsprüfung müssen im Browser Java-Script und Cookies aktiviert sein. *Joomla!* setzt die Anweisung *Fehler anzeigen* standardmäßig. Diese Option kann später deaktiviert werden.

Schritt 3 beinhaltet die Lizenz und sollte vom zukünftigen Administrator der Webseite durchgelesen werden.

Zur Einrichtung der Datenbankanbindung im vierten Schritt sind die zugewiesenen Daten des Webhosters einzugeben. Für XAMPP lautet der Servername *localhost*, der Benutzername *root*, das Passwortfeld bleibt leer, der Datenbankname lautet *edu*. Unter den erweiterten Einstellungen ist ein Tabellenpräfix *edu\_* einzutragen. Die FTP-Konfiguration im fünften Schritt ist nur bei Verwendung eines eigenen lokalen Webservers mit dem Betriebssystem Linux notwendig. Die Installationshinweise erläutern die genaue Vorgehensweise.

Hinweis: Selbstverständlich kann für die Datenbank ein root-Passwort vergeben werden. XAMPP

ist allerdings nicht für den produktiven Einsatz eines Webservers entwickelt worden, sondern vielmehr als "Testumgebung" eigener Projekte. Desweiteren hat im Normalfall nur der Entwickler Zugang zur eigenen "Testumgebung". Aus diesem Grund wird auf die Vergabe eines *root*-Passwortes verzichtet.

Der Name der Webseite ist im sechsten Schritt anzugeben, kann jedoch nachträglich jederzeit wieder geändert werden. Eine gültige E-Mailadresse ist mit einem Passwort einzugeben und zu bestätigen. Empfehlenswert ist die Installation der Beispieldateien.

Im Abschlussschritt sieben muss das Verzeichnis *installation* gelöscht werden. *Joomla!* verweigert ansonsten das Anzeigen der Webseite, sowie das Einloggen in das Backend, welches durch das Klicken auf den *Admin* Button zu erreichen ist.

### Webseite

Ab jetzt ist die Webseite des CMS unter der URL http://localhost/edu, beziehungsweise http://www.ihredomain.tld/edu/ erreichbar.

# Das Backend von Joomla!

Die Konfigurationsumgebung von Joomla! ist jederzeit über den Zusatz /administrator der URL erreichbar. Für XAMPP lautet die URL http://localhost/edu/administrator. Als "Backend" wird die Konfigurationsebene für die Administratoren bezeichnet. Benutzer hingegen sehen nur das "Frontend", die eigentliche Webseite.

# Administratorebene

Zur Navigation innerhalb des Backend von *Joomla!* ist nur auf die CMS-eigenen Steuerungsmodule zurückzugreifen. Die browserspezifischen Menüpunkte, insbesondere die Schaltflächen für *Zurück* oder *Vorwärts*, sollten, um Datenverlust vorzubeugen, niemals verwendet werden!

Unter der Konfiguration lassen sich grundlegende Einstellungen der Webseite, des Systems, sowie des Servers vornehmen. Weitere, als die nachfolgend erwähnten Einstellungsschritte, sind auf jeden Fall mit Vorsicht und Bedacht vorzunehmen!

Die Listenlänge der Site-Einstellungen wird auf den Wert 100 gesetzt. Diese Maßnahme erleichtert das spätere Auffinden und Bearbeiten der Beitrags-, Komponenten-, Modul- und Plugin-Listen. Unter dem Punkt System die Benutzerregistrierung und die Aktivierung neuer Konten ausschalten. Der Menüpunkt Server bleibt unberührt bis auf eine Einstellung unberührt. Lediglich die erwähnte Option *Fehler anzeigen* kann hier unter *Fehler berichten* von *Standard* auf *Nirgendwo* oder bei Bedarf auf *Maximum* gestellt werden. Ein Klick auf *Speichern* übernimmt die getroffenen Einstellungen.

# Sicherheitsrelevante Hinweise

Die Anmeldenamen und das Passwörter sämtlicher Benutzer des CMS Joomla! lassen sich unter  $Site \triangleright Benutzer$  ändern. Wird XAMPP nicht als Testumgebung eingesetzt, ist es ratsam, den

### Benutzernamen admin zu ersetzen!

Nur der/die Administrator(en) sollte(n) die Anmeldeinformationen zum Backend der Webseite und vor allem die Zugangsdaten des FTP-Programmes bekannt sein. Mutwillige Benutzer können anderenfalls an sensible Zugangsdaten gelangen.

# Artikel, Banner löschen

Über den Menüpunkt Inhalt  $\triangleright$  Beiträge werden alle vorhandenen Beitragsartikel angezeigt. Sämtliche Artikel werden markiert und in den Papierkorb verschoben. Das endgültige Löschen der Beispielinhalte erfolgt im Hauptmenü Inhalte  $\triangleright$  Papierkorb: Beiträge und muss zweimal bestätigt werden. Jetzt lassen sich sämtliche Kategorien und Bereiche über den Papierkorb löschen. Mit den vorinstallierten Bannern wird auf die gleiche Weise unter Komponenten  $\triangleright$  Banner  $\triangleright$  Banner verfahren.

# Menüeinträge löschen

Unter  $Men \ddot{u}s \triangleright Hauptmen \ddot{u}$ , sowie  $Men \ddot{u}s \triangleright Men \ddot{u}$  oben, alle Einträge, mit Ausnahme der Startseite löschen.

# Module deaktivieren

Es sind hierzu alle, bis auf die in Abbildung 26 ersichtlichen Module, unter dem Menüpunkt *Erweiterungen*  $\triangleright$  *Module*, zu deaktivieren, indem einmal auf jedes entsprechende grüne Häkchen der Spalte Aktiviert geklickt wird.

Name	Aktiviert	Reihenfolge	Zugriffsebene
Hauptmenü	✓	1	Öffentlich
Menü oben	✓	1	Öffentlich
Fußzeile	✓	2	Öffentlich
Benutzermenü	∠	∞ 4	Registriert
Anmeldung	×	▲ 8	Öffentlich

Abbildung 26: Aktivierte Module

# **Plugin Sourcerer**

Dieses Plugin ist unter der Adresse http://www.nonumber.nl/extensions/sourcerer?tab= download in der aktuellen Version 2.4.3 erhältlich. Die Paketdatei wird im Backend von *Joomla!* über *Erweiterungen* ▷ *Installieren/Deinstallieren* hochgeladen. *Sourcerer* erlaubt das Einbinden von PHP-Programmcode in Beiträgen und wird zur Darstellung der Google Karte auf der Webseite benötigt.

### Google Maps-API Key

Zur Darstellung einer Karte und Verwendung der Google Maps-API wird ein Schlüssel und ein Google Konto vorausgesetzt. Google bietet den kostenlosen API Key unter der Adresse http://code.google.com/intl/de-DE/apis/maps/signup.html an. Die URL einer lokalen Webseite mit XAMPP lautet: http://localhost. Ansonsten ist die Webadresse in der Form http://www.yourdomain.tld anzugeben. Den API-Schlüssel generieren lassen durch nachfolgende Anmeldung im eigenen Google Konto. Ist kein Konto bei Google vorhanden, wird zur kostenlosen Registrierung aufgefordert. Die Datei googleMapFiles/APIkey/googleAPIkey.php mit einem Editor öffnen und den generierten Google Maps-API Key eintragen, siehe Listing 83.

```
<?php
$google_api_key = "IHR_GOOGLE_API_KEY";
?>
```

Listing 83: Google Maps-API Key einfügen

# Datenbankverbindung einrichten

Datenbankserver, Benutzer, Passwort, sowie der Datenbankname sind in die dafür vorgefertigte Datei googleMapFiles/db/config.inc.php einzutragen. Die erforderlichen Daten werden vom Webhoster bereitgestellt. Für den lokalen Webserver XAMPP lautet der Inhalt wie in Listing 84 angegeben.

```
<?php

$dbserver = "localhost";

$dbuser = "root";

$dbpassword = "";

$dbname = "edu";

?>
```

Listing 84: Datenbankverbindung einrichten

# Datenbanktabellen hinzufügen

Webhoster bieten einen Link oder die erforderlichen Zugangsdaten zur Administration der Datenbank. Meistens kommt die Verwaltungssoftware *phpMyAdmin* zum Einsatz. Bei XAMPP lässt sich diese vorinstallierte Software über den Link http://localhost/phpmyadmin aufrufen.

Im Auswahlmenü von phpMyAdmin wird der Datenbankname edu ausgewählt. Mit einem Klick auf den Reiter SQL wird der SQL-Befehl aus Listing 85 für die Tabelle  $db_category$  eingefügt.

```
CREATE TABLE IF NOT EXISTS 'db_category' (
 'id_cat' int(11) NOT NULL AUTO_INCREMENT,
 'iconname' varchar(20) NOT NULL,
 PRIMARY KEY ('id_cat')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=0;
```

Listing 85: Tabellenstruktur für Tabelle *db\_category* 

Anschließend mit dem Button OK bestätigen und für die Tabelle  $db\_geoweb$ , siehe Listing 86 wiederholen.

```
CREATE TABLE IF NOT EXISTS 'db_geoweb' (
    'id_geo' int(11) NOT NULL AUTO_INCREMENT,
    'id_cat' int(11) DEFAULT NULL,
    'lat' double NOT NULL,
    'lat' double NOT NULL,
    'name' text,
    'text' longtext,
    'link' text,
    'link' text,
    'image' varchar(20) DEFAULT NULL,
    PRIMARY KEY ('id_geo'),
    KEY 'FK_IST' ('id_cat')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=0;
```

Listing 86: Tabellenstruktur für Tabelle db geoweb

### Bereiche

Ein neuer Bereich maps wird im Hauptmenüpunkt  $Inhalt \triangleright Bereiche$  mit dem Button Neu erstellt. Als Titel wird maps eingetragen und anschließend gespeichert.

### Kategorie

Der Kategoriename edu wird als Titel unter  $Inhalt \triangleright Kategorie$  nach dem Anklicken den Neu-Buttons verwendet und gespeichert. Da erst ein Bereich existiert, hat das CMS die Kategorie edu automatisch dem Bereich maps zugeordnet.

#### Konfiguration Beiträge

Das Konfigurationsmenüs der Beiträge findet sich im Hauptmenü unter  $Inhalt \triangleright Beiträge$  und *Einstellungen.* Aus Abbildung 27 kann die erforderliche Konfiguration entnommen werden.

Nicht zugängliche Links anzeigen Beitragstitel anzeigen Verlinkte Titel Einleitungstext anzeigen	<ul> <li>Nein</li> <li>Ja</li> <li>Nein</li> <li>Ja</li> <li>Nein</li> <li>Ja</li> <li>Verbergen</li> <li>Ansteinen</li> </ul>	Icons PDF-Icon Drucken-Icon E-Mail-Icon	<ul> <li>Verbergen</li> <li>Anzeigen</li> <li>Verbergen</li> <li>Anzeigen</li> <li>Verbergen</li> <li>Anzeigen</li> <li>Verbergen</li> <li>Anzeigen</li> </ul>	
Bereichsname	Verbergen      Anzeigen	Zugriffe Für jeden Feedeintrag	Verbergen Anzeigen     Finleitungstext Gesamter Text	
Verlinkbarer Bereichstitel Kategorietitel Verlinkbarer Kategorietitel	<ul> <li>Nein <sup>O</sup> Ja</li> <li>Verbergen <sup>O</sup> Anzeigen</li> <li>Nein <sup>O</sup> Ja</li> </ul>	anzeigen Filteroptionen		
Autor Erstellungsdatum und -zeit	<ul> <li>Verbergen</li> <li>Anzeigen</li> <li>Verbergen</li> <li>Anzeigen</li> </ul>	Filtergruppen	Offentiches Frontend - Registriert - Autor - Editor - Publisher - Offentliches Backend - Manager	
Navigation anzeigen	<ul> <li>Verbergen</li> <li>Anzeigen</li> <li>Verbergen</li> <li>Anzeigen</li> </ul>	Filterverfahren	Administrator     Super Administrator     Super Administrator     Super Administrator     Restitutiste (Standard)      Positivitiste      Kein HTML	
Beitragsbewertung/Umfrage	<ul> <li>Verbergen</li> <li>Anzeigen</li> <li>Verbergen</li> <li>Anzeigen</li> </ul>	Elemente filtern Attribute filtern		

Abbildung 27: Globale Beitragskonfiguration

# Beiträge erstellen

Im Hauptmenü:  $Inhalt \triangleright Beiträge$  auswählen und durch Klicken auf die Schaltfläche Neu einen Beitrag mit dem Titel getMarkers erstellen. Im Auswahlmenü den Bereich maps wählen und als Kategorie edu. Als Inhalt ist folgender Text einzutragen, wie aus Abbildung 28 ersichtlich:

Titel Alias Bereich	getMarkers getmarkers maps	Freigegeben Startseite Kategorie	<ul> <li>○ Nein <ul> <li>● Ja</li> <li>● Nein <ul> <li>● Ja</li> <li>■ edu</li> <li>■</li> </ul> </li> </ul></li></ul>
B <i>I</i> i≡ i≡ - ⊘	Ŭ AB€   ≣ ≣ ≣   Format   ≇ ≇   ♥) №   ∞ ∯ Ů 死   ■   ×₂ ײ   Ω	▼ Absatz ▼	
{source php<br {/source	e} include("googleMapFiles/getMarkers/inc re}	dex.php"); ?>	

Abbildung 28: Beitrag getMarkers erstellen

Nach derselben Vorgehensweise wird ein weiterer Beitrag mit dem Titel *setMarkers* und nachfolgendem Inhalt erstellt:

```
{source}
    <?php include("googleMapFiles/setMarkers/index.php"); ?>
{/source}
```

Listing 87: Beitragsinhalt setMarkers

## Startseite

Joomla! erwartet zumindest einen Beitrag, welcher beim Aufruf der Webseite als Inhalt auf der Startseite dargestellt wird. Hierzu im Hauptmenü von Joomla! Menüs  $\triangleright$  Hauptmenü wählen und den Neu-Button anklicken. Auf Beiträge  $\triangleright$  Beitrag  $\triangleright$  Layout: Beitrag (Joomla!-Standard) klicken. Für den Titel Bildungsstätten eingeben, sowie unter Beitrag wählen auf getMarkers klicken und speichern.

Die *Checkbox* links neben *Bildungsstätten* anwählen und im oberen Menü auf *Standard* klicken. Somit ist der Beitrag *getMarkers* die aktuelle Startseite. Der vorherige Eintrag namens *Startseite* wird gelöscht.

### Beitragszugriffe

Mit dem Beitrag setMarkers lassen sich auf der Google Karte der Webseite neue Marker platzieren. Dies sollte nur den registrierten Benutzern und/oder den Administratoren vorbehalten sein, um unseriöse Marker-Einträge zu vermeiden. Im Hauptmenü Inhalt  $\triangleright$  Beiträge kann die jeweilige Zugriffsebene der einzelnen Beiträge bestimmt werden. Durch Klicken auf Öffentlich wechselt die Zugriffsebene auf Registriert für gleichnamige registrierte Benutzer des CMS. Ein erneuter Klick verändert die Zugriffsebene auf Spezial. Somit ist dieser Beitrag nur für Administratoren, welche sich über das Frontend eingeloggt haben, sichtbar.

Die Zugriffsebene des Beitrags *setMarkers* wird aus Sicherheitsgründen auf *Spezial* gesetzt. Somit wird gewährleistet, dass dieser Beitrag auch durch Manipulation der URL nicht erreichbar ist.

#### Menüeinträge löschen

Im Hauptmenü  $Menüs \triangleright Menüs$  werden die jeweiligen Menüpunkte Schlüsselkonzepte, Beispielseiten, sowie Weiteres Menü gelöscht. Die Sicherheitsabfrage bezüglich der enthaltenen Untermenüpunkte wird ebenfalls durch einen Klick auf den Löschen-Button bestätigt.

# Benutzermenü anpassen

Zum Bearbeiten des Benutzermenüs im Hauptmenü  $Menüs \triangleright Benutzermenü$  wählen. Die beiden Menüeinträge Beitrag einreichen, sowie Weblink einreichen werden jeweils durch einen Klick auf den grünen Haken der Tabellenspalte Freigeben deaktiviert, wie Abbildung 29 zeigt.

Menüeintrag	Standard	Freigegeben	Reihenfolge= 🗞	Zugriffsebene
Ihre Details		¥	▼ 1	Registriert
Beitrag einreichen		0	▲ ▼ 2	Spezial
Weblink einreichen		0	▲ ▼ 3	Spezial
Abmelden		¥	<b>4</b>	Registriert

# Abbildung 29: Einstellungen Benutzermenü

# Projektspezifischen Ordner und Dateien

Der Ordner googleMapFiles ist in das Hauptverzeichnis der Joomla! Installation zu kopieren.

Die beiden Verzeichnisse *mod\_edu* und *mod\_edu\_cat* müssen sich im Unterordner *modules* befinden.

Der Templateordner geoweb wird in das Joomla!-Verzeichnis templates kopiert.

### Template

Im Hauptmenü: Erweiterungen > Templates das GeoWeb Template als Standard auswählen.

### Module installieren

Die erforderlichen Module *EDU Google Maps* und *EDU Google Maps Category* lassen sich unter *Erweiterungen*  $\triangleright$  *Module*, sowie einem Klick auf *Neu* nachinstallieren. Als Titel jeweils den entsprechenden Namen *EDU Google Maps* oder *EDU Google Maps Caterory* wählen und speichern.

### Menüeinträge

Im Hauptmenü unter  $Menü \triangleright Menü$  oben einen neuen Menüpunkt auf der Webseite erstellen. Den Button Neu und sodann Beiträge  $\triangleright$  Beitrag  $\triangleright$  Layout: Beitrag (Joomla!-Standard) anklicken. Als Titel Bildungsstätten eingeben und unter Basisparameter den Beitrag getMarkers wählen und speichern.

Für den nächsten Menüpunkt lautet der Titel Marker setzen, als Beitrag wird setMarkers gewählt und gespeichert. Danach für setMarkers in der Spalte Zugriffsebene von Öffentlich auf Spezial wechseln.

# Modulpositionen einrichten

Im Hauptmenü Erweiterungen  $\triangleright$  Module auswählen und das Modul Anmeldung anklicken. Bei der Position right wählen. Bei der Menüzuweisung wird unter der Menüauswahl bei Menüs die Option Alle markiert und gespeichert. Dem Modul EDU Google Maps, sowie dem Modul EDU Google

Maps Category ebenfalls die Position right zuweisen. EDU Google Maps erhält in der Menüzuweisung die Option Alle, EDU Google Maps Category wird in der Menüauswahl der Eintrag Marker setzen zugewiesen. Der Titel von EDU Google Maps wird in Auswahl Bildungsstätten, der Titel von EDU Google Maps Category in Kategorie bearbeiten umbenannt. Die jeweiligen Änderungen bei beiden Modulen separat speichern. Die Zugriffsebene von EDU Google Maps Category wird auf Spezial gestellt. Anschließend wird im Modul Benutzermenü unter Parameter, Erweiterte Parameter der Modul-Klassen-Suffix namens \_menu heraus gelöscht und gespeichert.

### Module Hauptmenü

Im Hauptmenü Erweiterungen  $\triangleright$  Module das Modul Hauptmenü mittels Klick in der Spalte Aktiviert deaktivieren.

Abbildung 30 zeigt eine Übersicht der aktivierten Module, deren Zugriffsebene, Position, sowie die Einstellungen der Seiten.

Name	Aktiviert	Reihenfolge	Zugriffsebene	Position	Seiten	Тур
Benutzermenü	¥	<b>v</b> 0	Registriert	right	Alle	mod_mainmenu
Anmeldung	×		Öffentlich	right	Alle	mod_login
Auswahl Bildungsstätten	✓	A V 0	Öffentlich	right	Alle	mod_edu
Kategorie bearbeiten	✓	۵	Spezial	right	Verschiedene	mod_edu_cat
Menü oben	<b>1</b>	1	Öffentlich	user3	Alle	mod_mainmenu
Fußzeile	$\checkmark$	2	Öffentlich	footer	Alle	mod_footer
Hauptmenü	٥	<b>v</b> 0	Öffentlich	right	Alle	mod_mainmenu

Abbildung 30: Ausschnitt aktivierter Module

#### Webseite

Die Template-Einstellungen, Menüstruktur, Modulpositionen, sowie das Anzeigen der Karte von Google sind jetzt fertig eingerichtet und lässt sich unter der URL http://localhost/edu einsehen.

# Marker-Icon

Mit Hilfe eines Bildbearbeitungsprogrammes sind für jede zukünftig geplante Kategorie Icons zu erstellen. Die Icons müssen im Bildformat PNG mit den maximalen Abmessungen von 32 Pixeln in der Breite und 34 Pixeln in der Höhe abgespeichert werden. Ein Beispiel für ein auf eine Kategorie *Fachhochschule* bezogenes Icon ist in Abbildung 31 zu sehen.



Abbildung 31: Kategorie-Icon Fachhochschule

# Anmeldung Frontend

Die Anmeldung erfolgt auf der eigentlichen Webseite mit den Zugangsdaten des Administrators. Im Menü erscheint der Menüpunkt *Marker setzen*.

### Kategorie anlegen

Durch Anklicken des Menüpunktes *Marker setzen* erscheint in der rechten Spalte das Modul namens *Kategorie bearbeiten*. Eine neue Kategorie, beispielsweise *Fachhochschule* eingeben und den Button *Kategorie anlegen* klicken.

### Kategorie-Icon

Das erstellte Kategorie-Icon Fachhochschule wird unter Kategorie bearbeiten durch Klicken auf den Durchsuchen-Button angegeben. Im Auswahlmenü die Kategorie Fachhochschule auswählen und den Button Kategorie hochladen betätigen. Für die angelegte Kategorie Fachhochschule, sowie dem zugehörigen Icon Fachhochschule, lassen sich jetzt beliebig viele Fachhochschulen auf der Google Karte platzieren.

#### Marker setzen

Um einen neuen Marker auf der Webseite setzen zu können, muss ein Bild der Bildungsstätte vorhanden sein und mit der Maske *Einen neuen Marker setzen* hochgeladen werden. Sodann sind die erforderlichen Eingaben, wie Name, Link zur Homepage, passende Kategorie, sowie eine treffende Beschreibung zur Bildungsstätte anzugeben. Ein roter Marker wird auf der Google Karte angezeigt, falls das hochgeladene Bild der Bildungsstätte keine geocodierten Koordinaten aufweist. Dieser Marker ist auf der bildzugehörigen Position zu platzieren. Ein Klick auf den *Speichern*-Button übernimmt die Eingaben oder weißt gegebenenfalls auf Eingabefehler hin.

### Marker anzeigen

Den oder die bereits erstellten Marker zu den jeweiligen Kategorien der Bildungsstätten werden mit den entsprechenden Kategorie-Icons durch einen Klick auf *Bildungsstätten* im Menü auf der Google Karte angezeigt.

# 10 Quellcode Geoinformationssystem

# 10.1 googleMapFiles/APIkey

```
<?php
$google_api_key = "Ihr Google Api Key"; // Der API-KEY ist auf die URL
einer Webseite codiert!
?>
```

Listing 88: Google Map-API Key

# 10.2 googleMapFiles/db

```
<?php

$dbserver = "localhost";

$dbuser = "root";

$dbpassword = "";

$dbname = "edu";

?>
```

Listing 89: Datenbankzugang konfigurieren

```
<?PHP
include ("config.inc.php");
$db_connection = mysql_connect($dbserver, $dbuser, $dbpassword) or die ("
    Database: Connection failed!");
$db_name = mysql_select_db($dbname, $db_connection) or die ("Database:
    DBName failed!");
?>
```

Listing 90: Verbindung zur Datenbank aufbauen

# 10.3 googleMapFiles/getMarkers

```
<?php
defined('_JEXEC') or die('Restricted access');
include ("googleMapFiles/APIkey/googleAPIkey.php");
?>
<script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=<?php
echo $google_api_key; ?>" type="text/javascript">
</script>
</script>
<style type="text/css">
@import url("http://www.google.com/uds/css/gsearch.css");
@import url("http://www.google.com/uds/solutions/localsearch/
gmlocalsearch.css");
```

```
/* needed for the localsearch */
</style>
<script src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0" type="</pre>
   text/javascript">
   // needed for the localsearch
</script>
<script src="http://www.google.com/uds/solutions/localsearch/gmlocalsearch</pre>
   .js" type="text/javascript">
    // needed for the localsearch
</script>
<script src="googleMapFiles/getMarkers/map functions.js" type="text/</pre>
   javascript">
</script>
<!--[if IE]>
   <style type="text/css">
    v\:*{ behavior:url(#default#VML);}
    </style>
<![endif]-->
<body class="loading">
   <div id="map">
        Map loading ...
        < noscript >
            Turn on JavaScript for Google Maps
        </noscript>
    </div>
    <div id="status">
        GEO-Daten werden geladen!
        </div>
</body>
```

```
Listing 91: getMarkers index.php
```

```
var centerLatitude = 47.800000; // Geocoords for Salzburg / Austria
var centerLongitude = 13.10000; // Geocoords for Salzburg / Austria
//var centerLatitude = google.loader.ClientLocation.latitude;
//var centerLongitude = google.loader.ClientLocation.longitude;
var startZoom = 11;
var map;
function umlaut(string){
    string = string.replace(/"a/g, "ae");
    string = string.replace(/"o/g, "oe");
    string = string.replace(/"u/g, "ue");
    string = string.replace(/"A/g, "Ae");
```

```
string = string.replace(/ "O/g, "Oe");
    string = string.replace(/"U/g, "Ue");
    string = string.replace(/"s/g, "ss");
    string = string.replace(/ /g, "-");
    return string;
}
// get values of URL parameter
function get url parameter(name) {
    name = name.replace (/[\backslash []/, "\backslash \backslash [").replace (/[\backslash ]]/, "\backslash \backslash ]");
    var regexS = "[\\?&]" + name + "=([^&#]*)";
    var regex = new RegExp(regexS);
    var results = regex.exec(window.location.href);
    if (results == null)
        return "";
    else
        return results [1];
}
function changeBodyClass(from, to){
    document.body.className = document.body.className.replace(from, to);
    return false;
}
function init(){
    // display Google Maps only, if the browser is able to handle GMap API
    if (GBrowserIsCompatible()) {
        // create object map
        map = new GMap2(document.getElementById("map"));
        // function retrieveMarkers gets all markers stored in the xml
            file
        retrieveMarkers();
        // centers the map to the given lat, lng values and zooms it to
            the given value
        map.setCenter(new GLatLng(centerLatitude, centerLongitude),
            startZoom);
        // adding Google's localsearch
        map.addControl(new google.maps.LocalSearch(), new GControlPosition
            (G ANCHOR BOTTOM RIGHT, new GSize(130, 20));
        // adding the minimap
        map.addControl(new GOverviewMapControl());
        // the most recent MapControl
        var customUI = map.getDefaultUI();
        customUI.maptypes.hybrid = true;
        map.setUI(customUI);
```

```
else {
        alert ("Leider ist Ihr Browser mit der Google-Maps API nicht
           kompatibel!");
    }
   changeBodyClass('loading', 'standby');
}
// function to create marker(s) with lat, lng and the html
// tags for the information window...
function createMarker(point, html, iconImage){
   // change body.class to loading data
   changeBodyClass('standby', 'loading');
    if (iconImage) {
        var icon = new GIcon();
        // icon to display - if none load default google marker
        icon.image = "googleMapFiles/getMarkers/icons/" + umlaut(iconImage
           .toLowerCase()) + ".png";
        icon.shadow = "googleMapFiles/getMarkers/icons/shadow50.png";
        icon.iconSize = new GSize(32, 35); // icon size
        icon.iconAnchor = new GPoint(16, 35); // position of marker
        icon.infoWindowAnchor = new GPoint(32, 0); // position of html
           window
        var marker = new GMarker(point, icon); // create object marker
    }
    else {
        var marker = new GMarker(point);
    }
    // change body.class to data loaded (standby)
   changeBodyClass('loading', 'standby');
   GEvent.addListener(marker, 'click', function(){
        var markerHTML = html;
        marker.openInfoWindowHtml(markerHTML);
    });
    return marker;
}
// function to get the geo data from the xml file
function retrieveMarkers(){
   // fabric methode to create the XmlHttpRequest object (AJAX)
   var request = GXmlHttp.create();
   // get the specific facilities only
   var topic_url = ("googleMapFiles/getMarkers/retrieveMarkers.php" + "/?
       facilities=" + get url parameter('facilities'));
    // specify where to get the xml data from
```

```
request.open('GET', topic url, true);
// retrieve all xml data
request.onreadystatechange = function(){
   // as long as there is data to load (readyState == 4)
    if (request.readyState == 4) {
       // load the data from the xml file
       var xmlDoc = request.responseXML;
        var markers = xmlDoc.documentElement.getElementsByTagName("
           marker");
        for (var i = 0; i < markers.length; i++) {
            var lng = markers[i].getAttribute("lng");
            var lat = markers[i].getAttribute("lat");
           // for Internet Explorer to avoid errors, if one values is
                null
            if (lng && lat) {
               var latlng = new GLatLng(parseFloat(lat), parseFloat(
                   lng));
                var html = "";
               html += "<div id = \"marker";
               html += "
                   \operatorname{cellspacing} = \langle "0 \rangle " > ";
               html += "";
               html += "<h1>" + markers [i].getAttribute("name") + "</
                   h1>":
               html += "";
               html += "";
               html += "<div id = \"bubbleimage \">";
                /*
                * loading thubnails instead of full size image
                * using GD2
                */
               html += "<a href=\"googleMapFiles/getMarkers/images/"
                   + markers [i]. getAttribute ("image") + "\" target = \"
                   blank ">";
               html += "<img class = \"image \" src = \"googleMapFiles/"
                   getMarkers/images/thumbs/th " + markers[i].
                   getAttribute ("image") + "\" align=\"middle\"
                   hspace = \langle "10 \rangle " > ";
               html += "<br />Bild vergroessern";
               html += "</a>";
               html += "</div>";
               html += "";
               html += "";
               html += "<div class = \"infotext \">";
               html += "";
                //html += "<b>Beschreibung</b>:";
```
```
html += markers[i].getAttribute("text");
                   html += "";
                   html += "</div>";
                   html += "";
                   html += "";
                   html += "";
                   html += "<b>Quelle </b>: ";
                   html += "<a href=\"" + markers[i].getAttribute("link")
                       + "\" target=\" blank\">";
                   html += markers[i].getAttribute("link");
                   html += "</a>";
                   html += "";
                   html += "";
                   html += "";
                   html += "</div>";
                   var iconImage = markers[i].getAttribute("iconname");
                   // create the marker(s) on the map including the
                      information window(s)
                   var point = new GLatLng(lat, lng);
                   map.addOverlay(createMarker(point, html, iconImage));
               }
           }
       }
   }
   request.send(null);
}
// to avoid writting the onload() function in the <body> tag
// it's better to start the functions within this script...
window.onload = init;
// unloading map and free memory
window.onunload = GUnload;
```

Listing 92: getMarkers Kartenfunktionen

```
<?php
include ("../db/db_connect.php");
/*
 * the following line are required to produce a valid XML data stream
 * from the database of the server to the client in order to display
 * the markers on google map...
 *
 */
header('Content-Type: text/xml; charset=UTF-8');
```

```
// mysql query to set utf8
$query = mysql_query("set names 'utf8'");
// if there's an selection in URL – the query differs
// removing underscores and replace them with spaces
if ($ GET["facilities"] != "") {
    // retrieve data from database
    // if "Fachhochschule" is selected only these category is displayed
    $query = mysql query("SELECT geo.lat, geo.lng, geo.name, geo.text, geo
        .link, geo.image, cat.iconname from db geoweb geo
                         INNER JOIN db category cat
                         WHERE geo.id cat = cat.id cat
                         AND cat.id_cat = '".$_GET["facilities"]."'");
} else {
    // if no named category is found display everything
    $query = mysql_query("SELECT geo.lat, geo.lng, geo.name, geo.text, geo
        .link, geo.image, cat.iconname from db geoweb geo
                         INNER JOIN db category cat
                        WHERE geo.id cat = cat.id cat");
}
// get amount of fields from selected database table
$fields = mysql num fields($query);
// XML data opening tag
markers = "<markers>";
while ($value = mysql fetch array($query)) {
    $markers .= "<marker ";</pre>
    /*
     * for each field of the database table
     * get the field name and the value
     * this even sends the id of the table
     * but this can be ignored, because map functions.js
     * doesn't ask for the id to be displayed!
     */
    for (\$i = 0; \$i < \$fields; \$i++) {
        $markers .= "".mysql field name($query, $i)."=\"".addslashes(
            $value[$i])."\" ";
    }
    markers := "/>";
}
// XML data closing tag
```

```
$markers .= "</markers>";
// now the markers string is requested by map_functions.js
echo $markers;
?>
```

Listing 93: getMarkers XML-Daten auslesen

## $10.4 \ google MapFiles/set Markers$

```
<?php
defined('_JEXEC') or die('Restricted access');
include ("googleMapFiles/APIkey/googleAPIkey.php");
echo ("<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">");
?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
   \operatorname{com}:\operatorname{vml}">
    <head>
         <\!\!\text{meta http-equiv}=\!\!"\operatorname{content-type"}"\operatorname{content}=\!"\operatorname{text}/\operatorname{html}; \operatorname{charset}=\!\operatorname{utf}-8"
            \langle \rangle
         <title>Google Maps-API: JavaScript-Beispiel</title>
         <script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=</pre>
             <?php echo $google api key; ?>" type="text/javascript">
         </script>
         <style type="text/css">
              @import url("http://www.google.com/uds/css/gsearch.css");
              @import url("http://www.google.com/uds/solutions/localsearch/
                 gmlocalsearch.css");
             /* needed for the localsearch */
         </style>
         <script src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0"</pre>
             type="text/javascript">
             // needed for the localsearch
         </script>
         <script src="http://www.google.com/uds/solutions/localsearch/</pre>
             gmlocalsearch.js" type="text/javascript">
             // needed for the localsearch
         </script>
         <!--<script src="googleMapFiles/setMarkers/contextmenu.js" type="</pre>
             text/javascript">
         </script>>>>
         <script type="text/javascript" language="JavaScript">
              function deleteImage(name){
                  if (confirm("Soll dieses Bild gelöscht werden?")) {
                       document.location = ("<?php echo $ SERVER['REQUEST URI
                           ']; ?>\&deleteImage=" + name);
```

Listing 94: setMarkers index.php

```
var centerLatitude = 47.800000; // Geocoords for Salzburg / Austria
var centerLongitude = 13.10000; // Geocoords for Salzburg / Austria
//var centerLatitude = google.loader.ClientLocation.latitude;
//var centerLongitude = google.loader.ClientLocation.longitude;
var startZoom = 11;
var map;
function changeBodyClass(from, to){
   document.body.className = document.body.className.replace(from, to);
    return false;
}
function init(){
    // display Google Maps only, if the browser is able to handle GMap API
    if (GBrowserIsCompatible()) {
       // Dem Div-Container mit der ID 'map' die Karte zuweisen
       map = new GMap2(document.getElementById("map"));
        // centers the map to the given lat, lng values and zooms it to
           the given value
       map.setCenter(new GLatLng(centerLatitude, centerLongitude),
           startZoom);
       // adding Google's localsearch
       map.addControl(new google.maps.LocalSearch(), new GControlPosition
           (G ANCHOR BOTTOM RIGHT, new GSize(130, 20)));
        // adding the minimap
       map.addControl(new GOverviewMapControl());
        // the most recent MapControl
        var customUI = map.getDefaultUI();
        customUI.maptypes.hybrid = true;
        map.setUI(customUI);
```

}

```
// contextmenu to set markers if no GPS data is found in image
       //map.addControl(new ContextMenu(map));
       // check if image is uploaded and show up marker
       var askToSetMarker = document.getElementById("nogpsData");
       // hide dragable marker as long as image is not uploaded
       if (askToSetMarker != null) {
           // if image is uploaded
           // place marker on the left side of the dragbox
            marker = new GMarker(new GLatLng(centerLatitude,
               centerLongitude), {
                draggable: true
           });
           // if marker is dragged, submit lat and lng to page in order
               to be stored in database
           GEvent.addListener(marker, "dragend", function(){
                document.getElementById('lat').value = marker.getPoint().
                   lat();
                document.getElementById('lng').value = marker.getPoint().
                   lng();
            });
            // open info window to guide the user
            GEvent.addListener(marker, "dragend", function(){
                marker.openInfoWindowHtml("<div style=\"width: 300px; text
                   -align: justify;\">Bitte platzieren Sie den Marker auf
                    die bildrelevante Position.<br />Zoomen Sie
                   weiter in die Karte hinein, um den Marker genau
                   positionieren zu können.<br /><br />Wenn Sie den
                   Marker platziert haben, klicken Sie bitte auf den \"
                   Speichern \" Button. </div>");
            });
            // show this dragable marker on the map
           map.addOverlay(marker);
       }
   }
   else {
        alert ("Leider ist Ihr Browser mit der Google-Maps API nicht
           kompatibel!");
   }
   changeBodyClass('loading', 'standby');
function showAddress(address){
   if (geocoder) {
        geocoder.getLatLng(address, function(point){
```

```
if (!point) {
                alert(address + " not found");
            }
            else {
                map.setCenter(point, 13);
                var marker = new GMarker(point);
                map.addOverlay(marker);
                marker.openInfoWindowHtml(address);
            }
        });
   }
}
function initAll(){
    draginit(); // drag & drop overlay
    init(); // google maps
}
// to avoid writting the onload() function in the <body> tag
// it's better to start the functions within this script...
window.onload = initAll;
// unloading map and free memory
window.onunload = GUnload;
```

Listing 95: setMarkers Kartenfunktionen

```
<script type="text/javascript" src="googleMapFiles/setMarkers/drag.js">
</script>
<div id="overlay">
   <div onmousedown="dragstart(document.getElementById('overlay'))" id="</pre>
       innerHead">
       </div>
   <div onclick="minimize(document.getElementById('innerWrapper'))" id="</pre>
       innerButton">
       <img src="googleMapFiles/setMarkers/images/button.png" width="30px</pre>
          ">
   </div>
   <div id="innerWrapper">
       <div id="innerCont">
           <?php
           require ("googleMapFiles/setMarkers/uploadForm.php");
           ?>
       </div>
   </div>
```

</div>

Listing 96: setMarkers Overlay für Eingabemaske

```
// http://aktuell.de.selfhtml.org/artikel/javascript/draganddrop/
//Das Objekt, das gerade bewegt wird.
var dragobjekt = null;
// Position, an der das Objekt angeklickt wurde.
var dragx = 0;
var dragy = 0;
// Mausposition
var posx = 0;
var posy = 0;
var tmpHeight;
var flag = false;
function draginit(){
    // Initialisierung der Ueberwachung der Events
    document.onmousemove = drag;
    document.onmouseup = dragstop;
}
function dragstart(element){
    //Wird aufgerufen, wenn ein Objekt bewegt werden soll.
    dragobjekt = element;
    dragx = posx - dragobjekt.offsetLeft;
    dragy = posy - dragobjekt.offsetTop;
}
function dragstop(){
    //Wird aufgerufen, wenn ein Objekt nicht mehr bewegt werden soll.
    dragobjekt = null;
}
function drag(ereignis){
    //Wird aufgerufen, wenn die Maus bewegt wird und bewegt bei Bedarf das
        Objekt.
    posx = document.all ? window.event.clientX : ereignis.pageX;
    posy = document.all ? window.event.clientY : ereignis.pageY;
    if (dragobjekt != null) {
        dragobjekt.style.left = (posx - dragx) + "px";
        dragobjekt.style.top = (posy - dragy) + "px";
    }
```

```
}
function minimize(element){
    if (flag == false) {
        tmpHeight = element.offsetHeight;
        //Effect.BlindUp(element, { duration: 3.0 });
        element.style.height = "0px";
        flag = true;
    } else if (flag == true) {
        element.style.height = tmpHeight + "px";
        flag = false;
    }
}
```

Listing 97: *setMarkers* Eingabemaske

```
<?php
function checkGPSData($image) { // check if there is gps data in an image
    $exif = @exif_read_data($image, 'GPS');
    if (\$exif = true) {
        return true;
    } else {
        return false;
    }
}
function readGPSData($image) {
    exif = @exif_read_data($image, 0, true);
    if (\$exif = true) {
        $latitude = $exif['GPS']['GPSLatitude'];
        $longitude = $exif['GPS']['GPSLongitude'];
        // transform latitude from degrees, minutes, seconds form to
           dezimal form
        for (\$i = 0; \$i < 3; \$i++) {
            $latExplode[$i] = explode("/", $latitude[$i]); // split string
                 at "/"
```

}

```
}
                                             for (\$i = 0; \$i < 3; \$i++) {
                                                                   $latTmp = $latExplode[$i][0] / $latExplode[$i][1];
                                                                    (0) +=  (0) += 
                                             }
                                             // transform longitude from degrees, minutes, seconds form to
                                                                dezimal form
                                             for (\$i = 0; \$i < 3; \$i++) {
                                                                    $lngExplode[$i] = explode("/", $longitude[$i]); // split
                                                                                       string at "/"
                                             }
                                             for (\$i = 0; \$i < 3; \$i++) {
                                                                   $latTmp = $lngExplode[$i][0] / $lngExplode[$i][1];
                                                                    coord[1] +=  torm{main for a latence of the second statence of t
                                             }
                                             return $coord;
                       } else {
                                             return 0;
                      }
?>
```

Listing 98: setMarkers EXIF-Daten auslesen

```
<?php
include ("googleMapFiles/db/db connect.php");
require ("set functions.php");
require ("readExif.php");
?>
<?php
if (! empty($ GET['deleteImage'])) {
    $nameOfImageTODelete = $_GET[ 'deleteImage '];
    if (is file("googleMapFiles/getMarkers/images/".$nameOfImageTODelete))
        {
        unlink("googleMapFiles/getMarkers/images/".$nameOfImageTODelete);
        unlink("googleMapFiles/getMarkers/images/thumbs/th ".
           $nameOfImageTODelete);
    }
}
```

```
// function call to upload an image from an user
$checkIfImageIsUploaded = checkIfImageIsUploaded();
// check if a "new" image already exits on server
$checkIfAnImageAlreadyExists = checkIfAnImageAlreadyExists();
if (($checkIfAnImageAlreadyExists == false) AND ($checkIfImageIsUploaded
   != "true")) {
   // no new image exists and no image is uploaded
   // therefore load following <form>
?>
<form method="POST" action="<?php echo (htmlspecialchars($ SERVER['
   REQUEST_URI'])); ?>" enctype="multipart/form-data">
   <table class="uploadForm" border="0">
       < {
m tr} >
           Bitte wÄchlen Sie ein Bild aus, zu welchem Sie einen
                   Marker setzen wollen.
           < {
m tr} >
           <input type="file" name="imageFromClient" size="55"><input
                    type="hidden" name="sendImage" value="1">
           <\!/{
m tr}>
       < {
m tr} >
           <input type="submit" name="submit" value="Bild hochladen">
           < {
m tr} >
           </form>
<?php
}// if (($checkIfAnImageAlreadyExists == false) AND (
   $checkIfImageIsUploaded == false))
if ($checkIfAnImageAlreadyExists == true) {
    /*
    * an image already exists on the server!
```

?>

```
* - if EXIF data found in image
     * - go on with this image
     * - or delete this image in order to upload a new one
        - get latitude, longitude from image

    ask user for information

          - name
          - source
           - link
           - ...
          - store information in database
     * - if no EXIF data found in image
     * - go on with this image
     * - or delete this image in order to upload a new one
        - ask to place marker (to get latitude, longitude)
        - ask user for information
          – name
           - source
          - link
           - ...
           - store information in database
     */
    echo("Das Bild wurde erfolgreich hochgeladen!<br/>br />Bitte geben Sie
       hier die Informationen zu diesem Bild ein.<br /><br />");
    // get filename of the last uploaded image
    $getFileNameOfUploadedImage = getFileNameOfUploadedImage();
    // check for GPS data
    if (checkGPSData($getFileNameOfUploadedImage) == true) {
        // get the filename of new image
        $getFileNameOfUploadedImage = getFileNameOfUploadedImage();
        // get latitude, longitude from uploaded image
        $getLatLng = readGPSData($getFileNameOfUploadedImage);
        // form for GPS data
<form action="<?php echo $_SERVER['REQUEST_URI']; ?>" method="POST"
   enctype="multipart/form-data">
   <\!\mathrm{tr}\!>
            Name:
            <input type="text" name="title" size="30" value="<?php
                   echo $ POST['title']; ?>"<?php if ($ POST['gpsData']</pre>
                   == 1) { echo("disabled = \"true \""); } ?>/>
```

```
<td rowspan="6">
        <?php
        $imgToDisplay = explode("googleMapFiles/getMarkers/images/
            ", $getFileNameOfUploadedImage);
        echo("<img src=\"googleMapFiles/getMarkers/images/thumbs/
           th_".simgToDisplay[1]."\" hspace=\"10\">");
        ?>
    <\!/{
m tr}>
<\!\mathrm{tr}\!>
    Quelle:
    <input type="text" name="source" size="30" value="<?php
            echo $_POST['source']; ?>"<?php if ($_POST['gpsData']</pre>
           == 1) { echo("disabled = \langle "true \rangle ""); } ?>/>
    <\!/{
m tr}>
< {
m tr} >
    Kategorie:
    <?php
        if (\operatorname{POST}['gpsData'] = 1) {
            echo("<select name=\"category\" disabled=\"true\">");
        } else {
            echo("<select name=\"category\">");
        }
        // get categories from database
        $query = mysql_query("SELECT iconname FROM db category
           ORDER BY iconname ASC");
        while ($value = mysql_fetch_array($query)) {
            echo "<option value=\"".$value['iconname']."\"";</pre>
            if ($ POST['category'] == $value['iconname']) {
                 echo " selected \"";
            }
            echo ">" . $value [ 'iconname ']. "</option>\n";
        }
        echo("</select>");
        ?>
    <\!\mathrm{tr}\!>
```

```
<td colspan="2">
         
    <\!/{
m tr}>
<\!\mathrm{tr}\!>
    <td colspan="2">
        Beschreibung:
    <\!/\,t\,r\!>
<td colspan="2">
        <?php
        if ($_POST['gpsData'] == 1) {
             echo("<textarea name=\"description\" cols = \"38\" rows
                = \"5\" disabled = \"true\">");
        } else {
            echo("<textarea name=\"description\" cols = \"38\" rows
                = \parallel "5 \parallel "> ");
        }
        echo $ POST['description'];
        echo("</textarea>");
        ?>
    <\!/{
m tr}>
<td colspan="2">
        <?php
        if ($ POST['gpsData'] != 1) {
            // disable submit button after information was given
                by user
            echo("<input type=\"submit\" value=\"Speichern\"/>");
        }
        ?>
        <input type="hidden" id="gpsData" name="gpsData" value="1"
           >
    <?php
        /*
         * if user uploaded the wrong image
         * or if another image is still on the server due to
         * connection problems or user cancelled the upload
         *
         */
        if ($ POST['gpsData'] != 1) {
```

```
echo("<a href=\"javascript:deleteImage('".</pre>
                        $imgToDisplay[1]."')\">Dieses Bild lö schen?
                        a>");
                 }
                ?>
            </form>
<?php
// if submit button was clicked (GPS data found)
if (\operatorname{POST}['gpsData'] = 1) {
    // check if all needed information is available
    if (( empty($ POST['title']))) {
        echo("<div class=\"infoMissing\">Bitte geben Sie einen
            aussagekr\tilde{A}\squareftigen Namen f\tilde{A}\frac{1}{4}r das Bild an!</div>");
        checkIfInformationWasGivenGPS();
    } elseif (( empty($ POST['source']))) {
        echo("<div class=\"infoMissing\">Bitte geben Sie eine URL zu dem
            Bild an! < /div > ");
        checkIfInformationWasGivenGPS();
    } elseif (( empty($_POST['category']))) {
        echo("<div class=\"infoMissing\">Bitte wÃ\capathen Sie die passende
            Kategorie zum dem Bild aus!</div>");
        checkIfInformationWasGivenGPS();
    } elseif (( empty($_POST[ 'description ']))) {
        echo("<div class=\"infoMissing\">Bitte geben Sie eine Beschreibung
             fü r das Bild an!</div>");
        checkIfInformationWasGivenGPS();
    else 
        // hide "submit button" and "delete link" in form
        echo("<input type=\"id=\"gpsData\" name=\"gpsData\" value
            = | "0 | / | / | ;
        /*
         * updating database with given information
         */
        $filename = explode("googleMapFiles/getMarkers/images/",
            $getFileNameOfUploadedImage);
        $ category = $ POST['category'];
        // getting id for category from database
        query = ("SELECT id cat FROM db category WHERE iconname = \"".
            scategory." \setminus "");
        result = mysql query(squery);
        $id_cat = mysql_fetch_row($result);
```

```
insert into db = ("INSERT INTO db geoweb (id cat, lat, lng, name, ))
            text, link, image) VALUES ('".$id cat[0]."', '".$getLatLng
           [0]."', '".$getLatLng[1]."', '".htmlspecialchars($_POST['title
           '])."', '".htmlspecialchars($ POST['description'])."', '".
           htmlspecialchars($ POST['source'])."', '".$filename[1]."')");
       mysql_query($insert_into_db);
       echo("<span class = \"success \">Ihre Informationen wurden
           gespeichert!</span>br /> br />");
       echo("Um einen neuen Marker zu setzen, klicken Sie bitte ");
       echo("<a href=".$_SERVER['REQUEST URI'].">hier</a>");
       echo("<br />=);
       echo ("Wenn Sie keinen neuen Marker setzen wollen, klicken Sie
           bitte ");
       // get the url for "home" of the webserver
        $urlHome = explode("?", ($_SERVER['REQUEST_URI']));
        echo("<a href=".surlHome[0].">hier</a>");
        return true;
   // else - check if needed information was given by user
// if ($_POST['gpsData'] == 1)
} /* if (checkGPSData($getFileNameOfUploadedImage) == true) */ else {
   /*
    * no GPS data was found in image
    * place marker to get latitude and longitude
    */
   // form no GPS data was found
?>
<script src="googleMapFiles/setMarkers/setMarker.js" type="text/javascript"
   ">
</\operatorname{script}>
<form action="<?php echo $ SERVER['REQUEST URI']; ?>" method="POST"
   enctype="multipart/form-data">
   Name:
           <input type="text" name="title" size="30" value="<?php
                   echo $ POST['title ']; ?>"<?php if ($ POST['nogpsData']</pre>
                    = 1) { echo("disabled=\"true\""); } ?>/>
           <td rowspan="6">
                <?php
```

```
$imgToDisplay = explode("googleMapFiles/getMarkers/images/
           ", $getFileNameOfUploadedImage);
        echo("<img src=\"googleMapFiles/getMarkers/images/thumbs/
           th ".simgToDisplay[1]."\" hspace=\"10\">");
        ?>
    <\!/{
m tr}>
<\!\mathrm{tr}\!>
    Quelle:
    <input type="text" name="source" size="30" value="<?php
           echo $ POST['source']; ?>"<?php if ($ POST['nogpsData'
           ] = 1) \{ echo("disabled = \langle "true \rangle ""); \} ?>/>
    <\!\!/\,t\,r\!>
< {
m tr} >
    Kategorie:
    <?php
        if ($ POST['nogpsData'] == 1) {
            echo("<select name=\"category\" disabled=\"true\">");
        } else {
            echo("<select name=\"category\">");
        }
        // get categories from database
        $query = mysql query("SELECT iconname FROM db category
           ORDER BY iconname ASC");
        while ($value = mysql_fetch_array($query)) {
            echo "<option value=\"".$value['iconname']."\"";</pre>
            if ($ POST['category'] == $value['iconname']) {
                echo " selected \"";
            }
            echo ">" . $value [ 'iconname ']. "</option >\n";
        }
        echo("</select>");
        ?>
    <td colspan="2">
```

```
<td colspan="2">
        Beschreibung:
    <\!\!/\,t\,r\!>
< \mathrm{tr} >
    <td colspan="2">
        <?php
        if ($ POST['nogpsData'] == 1) {
            echo("<textarea name=\"description\" cols = \"38\" rows
                = \"5\" disabled = \"true\">");
        } else {
            echo("<textarea name=\"description\" cols = \"38\" rows
                = \parallel "5 \mid " > ");
        }
        echo($_POST['description']);
        echo("</textarea>");
        ?>
    <\!/{
m tr}>
<td colspan="2">
        <?php
        if ($_POST['nogpsData'] != 1) {
            echo("<div style=\"width: 300px; text-align: justify
                ;\">");
            echo ("Bevor Sie speichern, bitte noch den rotfarbenen
                Marker auf die bildzugehö rige Position auf
                der Karte platzieren!");
            echo("<br/>");
            echo("</div>");
            // disable submit button after information was given
                by user
            echo("<input type=\"submit\" value=\"Speichern\"/>");
        }
        ?>
        <input type="hidden" id="nogpsData" name="nogpsData" value
           ="1"/>
    <?php
        /*
         * if user uploaded the wrong image
         * or if another image is still on the server due to
         * connection problems or user cancelled the upload
```

```
*
                                      */
                                     if ($ POST['nogpsData'] != 1) {
                                              echo("<a href=\"javascript:deleteImage('".</pre>
                                                     $imgToDisplay[1]."')\">Dieses Bild lö schen?
                                                     a>");
                                    }
                                    ?>
                           <input id="lat" name="lat" type="hidden" value="<?php echo $_POST['lat
                  ']; ?>" /><input id="lng" name="lng" type="hidden" value="<?php
                echo $ POST['lng']; ?>" />
</form>
<?php
// if submit button was clicked (no GPS data found)
if (\operatorname{POST}[\operatorname{'nogpsData'}] = 1) {
         // check if all needed information is available
         if (( empty($ POST['title']))) {
                  echo("<div class=\"infoMissing\">Bitte geben Sie einen aussagekr&
                          auml; ftigen Namen fü r das Bild an!</div>");
                  checkIfInformationWasGivenNoGPS();
         } elseif (( empty($ POST['source']))) {
                  echo("<div class=\"infoMissing\">Bitte geben Sie eine URL zu dem
                          Bild an! < /div > ");
                  checkIfInformationWasGivenNoGPS();
         } elseif (( empty($ POST['category']))) {
                  echo("<div class=\"infoMissing\">Bitte w&auml; hlen Sie die
                          passende Kategorie zum dem Bild aus!</div>");
                  checkIfInformationWasGivenNoGPS();
         } elseif (( empty($ POST['description']))) {
                  echo("<div class=\"infoMissing\">Bitte geben Sie eine Beschreibung
                            fü r das Bild an!</div>");
                  checkIfInformationWasGivenNoGPS();
         } elseif (( empty($_POST['lat'])) && ( empty($_POST['lng']))) {
                  echo("<div class=\"infoMissing\">Bitte platzieren Sie den
                          rotenfarbenen Marker erneut < br />auf die bildzugeh&ouml; rige
                          Position auf der Karte, <br />damit die genaue Position &uuml;
                          bernommen werden kann! < /div > ");
                  checkIfInformationWasGivenNoGPS Marker();
         else 
                  // hide "submit button" and "delete link" in form
                  echo("<input type=\"hidden\" id=\"nogpsData\" name=\"nogpsData\" id=\"nogpsData\" name=\"nogpsData\" id=\"nogpsData\" id=\"
                          value = \langle "0 \rangle "/> ");
                  /*
```

```
* updating database with given information
         */
        $filename = explode("googleMapFiles/getMarkers/images/",
           $getFileNameOfUploadedImage);
        $ category = $ POST['category'];
        // getting id for category from database
        query = ("SELECT id cat FROM db category WHERE iconname = \"".
            scategory." \setminus "");
        result = mysql_query(squery);
        $id cat = mysql fetch row($result);
        insert into db = ("INSERT INTO db geoweb (id cat, lat, lng, name,
            text, link, image) VALUES ('".$id cat[0]."', '".$ POST['lat'
            ]."', '".$_POST['lng']."', '".htmlspecialchars($_POST['title']
           ])."', '".htmlspecialchars($ POST['description'])."', '".
           htmlspecialchars($ POST['source'])."', '".$filename[1]."')");
        mysql query($insert into db);
        echo("<span class = \"success \">Ihre Informationen wurden
            gespeichert!</span><br /><br />");
        echo("Um einen neuen Marker zu setzen, klicken Sie bitte ");
        echo("<a href=".$ SERVER['REQUEST URI'].">hier</a>");
        echo("<br /><br />");
        echo ("Wenn Sie keinen neuen Marker setzen wollen, klicken Sie
           bitte ");
        // get the url for "home" of the webserver
        $urlHome = explode("?", ($_SERVER['REQUEST_URI']));
        echo("<a href=".surlHome[0].">hier</a>");
        return true;
    }// else - check if needed information was given by user
// if ($ POST['nogpsData'] == 1)
}// else { no GPS data was found in image
}// if ($checkIfAnImageAlreadyExists == true)
```

```
?>
```

Listing 99: setMarkers Uplaod-Formular

<?php
/\*\*
 \* This file contains all needed functions to set markers
 \*/
// Upload maximum will be 5MB per file
\$MAXIMUM\_FILESIZE = 5 \* 1024 \* 1024;</pre>

```
/*
* returns the highest id geo from db geoweb
* used to rename an uploaded image file with
* filename: id geo+1.image
*/
function getNewImageFileName() {
    $query = mysql query("SELECT MAX(id geo) FROM db geoweb");
    value = mysql result(squery, 0);
   return (\$value + 1);
}
/*
* checks if there is already a new image on the server
 * might be due to the following reasons:
 * - user uploaded a new image and cancelled to fill out the form
 * - user lost connection to server
 * if file exits on server
 * - ask user to go on with this file
 * - offer possibility to delete the existing file
 * remember - only jpg, gif, png are valid image formats
 * ("jpeg" is renamed to "jpg")
*/
function checkIfAnImageAlreadyExists() {
   $getNewImageFileName = getNewImageFileName();
    if ((is file("googleMapFiles/getMarkers/images/geo ".
       $getNewImageFileName.".jpg")) || (is file("googleMapFiles/
       getMarkers/images/geo ".$getNewImageFileName.".gif")) || (is file(
       "googleMapFiles/getMarkers/images/geo".$getNewImageFileName.".png
       "))) {
       // a "new" image exists
       return true;
   } else {
       // no new image exists
       return false;
   }
}
/*
* returns filename of the last uploaded image
*/
function getFileNameOfUploadedImage() {
   $getNewImageFileName = getNewImageFileName();
```

```
if (is file("googleMapFiles/getMarkers/images/geo ".
       $getNewImageFileName.".jpg")) {
        return ("googleMapFiles/getMarkers/images/geo".
           $getNewImageFileName.".jpg");
    }
    if (is_file("googleMapFiles/getMarkers/images/geo_".
       $getNewImageFileName.".gif")) {
        return ("googleMapFiles/getMarkers/images/geo".
           $getNewImageFileName.".gif");
    }
    if (is file("googleMapFiles/getMarkers/images/geo ".
       $getNewImageFileName.".png")) {
        return ("googleMapFiles/getMarkers/images/geo".
           $getNewImageFileName.".png");
   }
}
/*
* checks if image is uploaded
 * checks if the uploaded file is an image
 * generates thumbnails of image
*/
function checkIfImageIsUploaded() {
   // if submit button was clicked
    if (\$ POST['sendImage'] == 1) {
        // get file type of uploaded image
        if (eregi('image/', $ FILES['imageFromClient']['type'])) {
            // allow only jpeg, jpg, png and gif as image files
            if ($ FILES['imageFromClient']['type'] == "image/jpeg" ||
               $_FILES['imageFromClient']['type'] == "image/jpg" ||
               $ FILES['imageFromClient']['type'] == "image/png" ||
               $_FILES['imageFromClient']['type'] == "image/gif") {
                // if image is being uploaded (not modified on server)
                // tmp name of image will be changed by function
                   getNewImageFileName()
                if (is_uploaded_file($_FILES['imageFromClient']['tmp_name'
                   ])) {
                    //remove spaces and other unwanted chars
                    safeFilename = preg_replace(array("/\s+/", "/[^-\.w
                       ]+/"), array("_", ""), trim($_FILES['
                       imageFromClient ' [ 'name ' ] ) );
                    // get file extension of uploaded image
                    $safeFileExtentsion = $ FILES['imageFromClient']['type
                        '];
                    safeFileExtentsion = explode('image/',
                        $safeFileExtentsion);
```

```
// file extension jpg is more common than jepg
if ($safeFileExtentsion[1] == "jpeg") {
    safeFileExtentsion[1] = "jpg";
}
/*
 * overwrite an existing image with the same file name
 * on the one hand: this prevents that the server may
    be
 * flooded with unused images ...
 * on the other hand: there won't be images with the
    same
 * filename in the image directory
 */
move uploaded file(($ FILES['imageFromClient']['
   tmp name']), "googleMapFiles/getMarkers/images/".
   $safeFilename);
// create a new image name for the uploaded file
$newImageName = getNewImageFileName();
// rename new image - prefix "geo_": other images won'
   t be overwritten by new uploaded images if
   filenames are equal!
rename ("googleMapFiles/getMarkers/images/".
   $safeFilename, "googleMapFiles/getMarkers/images/
   geo ".$newImageName.".".$safeFileExtentsion[1]);
/*
 * generating a thumbnail from the image in order to:
 * - increase performance by loading all markers
    faster
 * - decreasing traffic by showing only thumbnails
    instead of full size images
 */
$imageFile = ("googleMapFiles/getMarkers/images/geo".")
   $newImageName.".".$safeFileExtentsion[1]);
$imageSize = getimagesize($imageFile); // GD2 lib
imageWidth = imageSize[0];
imageHeight = imageSize[1];
imageType = imageSize[2];
switch ($imageType) {
    case 1: // GIF
        $image = imagecreatefromgif($imageFile);
        break;
    case 2: // JPEG
        $image = imagecreatefromjpeg($imageFile);
        break;
    case 3: // PNG
        $image = imagecreatefrompng($imageFile);
```

```
break;
    default:
        die ('Dieses Bildformat ist nicht zugelassen!')
           :
}
// maximum width and height of thumb in markers bubble
maxThumbWidth = 200;
maxThumbHeight = 200;
$thumbWidth = $imageWidth;
t = t = t = t 
// scaling width and height
if ($thumbWidth > $maxThumbWidth) {
    $factor = $maxThumbWidth / $thumbWidth;
    $thumbWidth *= $factor;
    $thumbHeight *= $factor;
}
if ($thumbHeight > $maxThumbHeight) {
    $factor = $maxThumbHeight / $thumbHeight;
    $thumbWidth *= $factor;
    $thumbHeight *= $factor;
}
// generating thumb
$thumb = imagecreatetruecolor($thumbWidth,
   $thumbHeight);
/*
 * copies a rectangular portion of one image to
    another image,
 * smoothly interpolating pixel values so that, in
    particular,
 * reducing the size of an image still retains a great
     deal of clarity.
 * (http://php.net/manual/de/function.
    imagecopyresampled.php)
 */
imagecopyresampled ($thumb, $image, 0, 0, 0, 0,
   $thumbWidth, $thumbHeight, $imageWidth,
   $imageHeight);
// saving thumb as file
switch ($imageType) {
    // 1 = GIF, 2 = JPG, 3 = PNG
    case 1: // GIF
        thumbFile = ("googleMapFiles/getMarkers/")
           images/thumbs/th_geo_".$newImageName.".gif
           ");
        break;
    case 2: // JPEG
```

}

```
$thumbFile = ("googleMapFiles/getMarkers/")
                                 images/thumbs/th_geo_".$newImageName.".jpg
                                 ");
                              break;
                          case 3: // PNG
                              $thumbFile = ("googleMapFiles/getMarkers/
                                 images/thumbs/th geo ".$newImageName.".png
                                 ");
                              break;
                          default:
                             // cant't happen again!
                              die('...');
                     }
                     imagejpeg($thumb, $thumbFile);
                     // free memory of created thumb
                     imagedestroy($thumb);
                 }//is_uploaded_file
                 /*
                  * check if image was uploaded successfully
                  * true => image uploaded and renamed successfully
                  * false => image not uploaded and not renamed
                      successfully
                  */
                 if (is file("googleMapFiles/getMarkers/images/geo ".
                     $newImageName.".".$safeFileExtentsion[1])) {
                     // clear file from cache for better performance
                     clearstatcache();
                     return ("true");
                 else 
                     return ("Serverfehler: Bilddatei konnte nicht
                         gespeichert werden!");
                 }
             } /* image[type] */ else {
                 return ("Als Bildformate sind nur JPEG, JPG, PNG und GIF
                     erlaubt!");
             }
         } /* eregi('image/', */ else {
             return ("Die hochgeladene Datei ist kein Bild!");
         }
    }/* submit */ else {
         // no picture was subbitted yet
         return ("false");
    }
// GPS: if title, source, category, ... is missing \Rightarrow error warning
```

```
function checkIfInformationWasGivenGPS() {
    echo("<form action=\"".$ SERVER['REQUEST URI']."\" method=\"POST\"
        enctype = \mbox{"multipart / form-data }">");
    echo("<input type=\"hidden\" name=\"title\" value=\"".\"POST['title'].
        " \setminus " > ");
    echo("<\!input type=\!\!\!\!|"hidden \!\!|" name=\!\!\!|"source \!\!|" value=\!\!|"".\$_POST['source']
        ]. "\"/>");
    echo("<input type=\"hidden\" name=\"category\" value=\"".$ POST['
        category']."\langle "/>");
    echo("<input type=\"hidden\" name=\"description\" value=\"".$ POST['
        description ']. "\"/>");
    echo("<input type=\"hidden\" id=\"lat\" name=\"lat\" value=\"".\"overline(").
        'lat ']. "\"/>");
    echo ("<input type=\"id=\"lng \" name= \"lng \" value= \". $ POST[
        (\ln g'). "\"/>");
    // GPS: show submit button and delete link in form
    echo("<input type=\"id=\"gpsData\" name=\"gpsData\" value
        = | "0 | | / > ");
    echo("<\!input type=\!\!\backslash"submit\!\backslash" value=\!\!\backslash"Fehlende Information hinzuf\tilde{A}^{1}_{4}gen
        || / > " );
    echo("</form>");
}
// No GPS: if title, source, category, \dots is missing => error warning
function checkIfInformationWasGivenNoGPS() {
    echo("<form action=\"".$ SERVER['REQUEST URI']."\" method=\"POST\"
        enctype = \langle "multipart / form - data \rangle ">");
    echo("<input type=\"hidden\" name=\"title\" value=\"".$ POST['title'].
        " \setminus " > ");
    echo("<input type=\"hidden\" name=\"source\" value=\"".$_POST['source']
        ]. "\"/>");
```

echo("<input type=\"hidden\" id=\"lat\" name=\"lat\" value=\"".POST['lat']."\"/>");

echo ("<input type=\"hidden \" id=\"lng \" name=\"lng \" value=\"".POST['lng']."\"/>");

// No GPS: show submit button and delete link in form

echo("<input type=\"hidden\" id=\"nogpsData\" name=\"nogpsData\" value =\"0\"/>");

echo("</form>");

```
// No GPS: if title, source, category, ... is missing \Rightarrow error warning
function checkIfInformationWasGivenNoGPS Marker() {
    echo("<form action=\"".$ SERVER['REQUEST URI']."\" method=\"POST\"
       enctype = \mbox{"multipart / form-data }">");
    echo("<\!input type=\"hidden" name=\"title" value=\"".\$_POST['title'].
       " \setminus " > ");
    echo ("<input type=\"hidden\" name=\"source\" value=\"".POST ; source '
       ]. "\"/>");
    echo("<input type=\"hidden\" name=\"category\" value=\"".$ POST['
       category']."\langle "/>");
    echo("<input type=\"hidden" name=\"description" value=\".$_POST[']
       description ']. "\"/>");
    'lat']. "\"/>");
    echo ("<input type=\"id=\"lng \" name= \"lng \" value= \". $ POST[
       'lng']."\"/>");
    // No GPS: show submit button and delete link in form
    echo("<input type=\"hidden\" id=\"nogpsData\" name=\"nogpsData\" value
       = |"0|"/>");
    echo("<\!input type=\"submit", value=\"Zur\tilde{A}_4^1ck um Marker zu setzen", "/>")
       ;
    echo("</form>");
}
?>
```

Listing 100: setMarkers Funktionen Upload-Formular

## 10.5 googleMapFiles/getRoutes

```
<?php
defined(' JEXEC') or die('Restricted access');
include ("googleMapFiles/APIkey/googleAPIkey.php");
echo("<!DOCTYPE html PUBLIC \"-//W3C//DID XHTML 1.0 Strict//EN\" \"http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>");
?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
   \operatorname{com}: \operatorname{vml}">
    <head>
        <title>Google Maps-API: JavaScript-Beispiel</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"</pre>
            >
        <meta name="robots" content="index, follow">
        <meta name="keywords" content="">
        <meta title="">
        <meta name="description" content="">
```

```
<link rel="stylesheet" type="text/css" href="css/main.css">
        <script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=</pre>
            <?php_echo_$google_api_key; ?>"_type="text/javascript">
        </script>
        <style type="text/css">
            @import url("http://www.google.com/uds/css/gsearch.css");
            @import url("http://www.google.com/uds/solutions/localsearch/
                gmlocalsearch.css");
            /* needed for the localsearch */
        </style>
        <script src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0"</pre>
           type="text/javascript">
            // needed for the localsearch
        </script>
        <script src="http://www.google.com/uds/solutions/localsearch/</pre>
            gmlocalsearch.js" type="text/javascript">
            // needed for the localsearch
        </script>
        <script src="googleMapFiles/getRoutes/map_functions.js" type="text"
           /javascript">
        </script>
    </head>
    <body>
        <div id="map" style="width: 100%; height: 600px">
        </div>
    </body>
</html>
```

Listing 101: getRoutes index.php

```
var centerLatitude = google.loader.ClientLocation.latitude;
var centerLongitude = google.loader.ClientLocation.longitude;
var startZoom = 7;
var map;
function changeBodyClass(from, to){
    document.body.className = document.body.className.replace(from, to);
    return false;
}
function init(){
    // display Google Maps only, if the browser is able to handle GMap API
    if (GBrowserIsCompatible()) {
        // create object map
        map = new GMap2(document.getElementById("map"));
}
```

```
// function retrieveMarkers gets all markers stored in the xml
           file
        retrieveRoutes();
        // centers the map to the given lat, lng values and zooms it to
           the
        // given value
       map.setCenter(new GLatLng(centerLatitude, centerLongitude),
           startZoom);
        // adding Google's localsearch
       map.addControl(new google.maps.LocalSearch(), new GControlPosition
           (G ANCHOR BOTTOM RIGHT, new GSize(130, 20)));
        // adding the minimap
       map.addControl(new GOverviewMapControl());
        // the most recent MapControl
        var customUI = map.getDefaultUI();
        customUI.maptypes.hybrid = true;
       map.setUI(customUI);
   }
   changeBodyClass('loading', 'standby');
}
/*
 * function createMarker(point, html){ var marker = new GMarker(point);
* GEvent.addListener(marker, 'click', function(){ var markerHTML = html;
 * marker.openInfoWindowHtml(markerHTML); }); return marker; }
*/
// function to get the geo data from the xml file
function retrieveRoutes() {
   // fabric methode to create the XmlHttpRequest object (AJAX)
   var request = GXmlHttp.create();
   // specify where to get the xml data from
    request.open('GET', 'googleMapFiles/getRoutes/retrieveRoutes.php',
       true);
   // retrieve all xml data
    request.onreadystatechange = function(){
        // as long as there is data to load (readyState == 4)
        if (request.readyState == 4) {
            // load the data from the xml file
            var xmlDoc = request.responseXML;
            var routes = xmlDoc.documentElement.getElementsByTagName("
               route");
            for (var i = 0; i < routes.length; i++) {
                map.addOverlay(retrieveRoute(routes[i].getAttribute("route
                   ")));
```

```
}
    }
    request.send(null);
}
// get routes from database with absolute URL
function retrieveRoute(path){
    var route = new GGeoXml("http://itsgeoweb.meyr.info/googleMapFiles/
       getRoutes/routes/" +
    path);
    return route;
}
// to avoid writting the onload() function in the <body> tag
// it's better to start the functions within this script...
window.onload = init;
// unloading map and free memory
window.onunload = GUnload;
```

Listing 102: getRoutes Kartenfunktionen

```
<?php
include ("../db/db connect.php");
/*
* the following line are required to produce a valid XML data stream
 * from the database of the server to the client in order to display
 * the markers on google map...
 *
 */
header('Content-Type: text/xml; charset=UTF-8');
// mysql query
$query = mysql_query("SELECT name, text, link, route FROM db_routes");
// get amount of fields from selected database table
$fields = mysql_num_fields($query);
// XML data opening tag
routes = "<routes>";
while ($value = mysql fetch array($query)) {
    $routes .= "<route ";</pre>
    /*
     * for each field of the database table
```

```
* get the field name and the value
     * this even sends the id of the table
     * but this can be ignored, because map functions.js
     * doesn't ask for the id to be displayed!
     */
    for (\$i = 0; \$i < \$fields; \$i++) {
        $routes .= "".mysql field name($query, $i)."=\"".addslashes($value)
            [$i])."\" ";
    ł
    sin (1 - 1) = "/>";
}
// XML data closing tag
sroutes := "</routes>";
// now the markers string is requested by map functions.js
echo $routes;
?>
```

Listing 103: getRoutes XML-Daten auslesen

```
<?php
```

```
//\,http://\,wp1173987.wp090.webpack.hosteurope.de/wordpress/?page_id{=}661
```

```
/\,/ GPX to KML converter
```

```
11
```

- // This script reads a Geocaching.com GPX file and creates a Google Maps
  KLM file , containing waypoints
- // for each cache, including some cache information and log entries. It also shows photos of any cache
- // for which they are available.
- // First of all: Disable any caching, set the MIME type and disable the time limit. This script can take some time...

```
header("Cache-Control: no-cache, must-revalidate");
//header("Expires: Sat, 26 Jul 1997 05:00:00 GMT");
//header("Content-type: application/vnd.google-earth.kml+xml");
set_time_limit(0);
```

- if ((\$gpxPath = /\*\$\_GET["gpx"]\*/"/GeoWeb/googleMapFiles/getRoutes/routes/ Salzburg\_nach\_Venedig.gpx") == "") die; \$cachedKMLPath = \$\_SERVER["DOCUMENT\_ROOT"].\$gpxPath.".kml";

```
if (file_exists($cachedKMLPath))
{
```

```
$gpxDate = filemtime($_SERVER["DOCUMENT_ROOT"].$gpxPath);
```

```
$kmlDate = filemtime($cachedKMLPath);
    if ($kmlDate >= $gpxDate)
    {
        $kmlFile = fopen($cachedKMLPath, "r");
        $kml = fread($kmlFile, filesize($cachedKMLPath));
        fclose($kmlFile);
        echo $kml;
        die;
    }
}
// These arrays define which icons will be use for the various cache
   containers, types and log events
eventTypeIcons = array(
    "Found it"
                         \Rightarrow "event found.png",
    "Didn't find it" => "event notfound.png",
    "Write note"
                         \implies "event_note.png");
cacheTypeIcons = array(
    "Traditional Cache" => "type traditional.gif",
    "Multi-cache"
                         => "type multi.gif",
                         => "type mystery.gif",
    "Unknown Cache"
                        => "type_virtual.gif",
    "Virtual Cache"
    "Webcam Cache"
                         \Rightarrow "type webcam.gif",
    "Earthcache"
                         \Rightarrow "type earth.gif",
    "Letterbox Hybrid" => "type letterbox.gif");
cacheSizeIcons = array(
    "Micro"
                         \Rightarrow "size micro.gif",
    "Small"
                         \implies "size small.gif",
    "Regular"
                         => "size regular.gif",
                         \implies "size large.gif",
    "Large"
    "Other"
                         \Rightarrow "size other.gif",
    "Virtual"
                         => "size_virtual.gif",
                         \implies "size not chosen.gif");
    "Not chosen"
// This function creates a thumbnail image if it doesn't yet exist
function createThumbnail($sourcePath, $width, $height)
{
    $thumbPath = $sourcePath.".".$width."x".$height.".thumb.jpg";
    if (file exists($thumbPath))
        return;
    list(sourceWidth, sourceHeight) = getimagesize(sourcePath);
    if ($height == 0)
    {
        $scale = $width / $sourceWidth;
        $newWidth = $sourceWidth * $scale;
```

```
$newHeight = $sourceHeight * $scale;
    }
    else if (\$width == 0)
    {
        $scale = $height / $sourceHeight;
        $newWidth = $sourceWidth * $scale;
        $newHeight = $sourceHeight * $scale;
    }
    else
    {
        \operatorname{snewWidth} = \operatorname{width};
        newHeight = height;
    }
    destinationImage = imagecreatetruecolor(snewWidth, snewHeight);
    $sourceImage = imagecreatefromjpeg($sourcePath);
    imagecopyresampled ($destinationImage, $sourceImage, 0, 0, 0, 0,
       $newWidth, $newHeight, $sourceWidth, $sourceHeight);
    imagejpeg($destinationImage, $thumbPath, 90);
}
// This function takes a date and time string of the format used in the
   GPX files and converts it to German notation
function reformatTime($time)
{
    year = substr(stime, 0, 4);
    month = substr(stime, 5, 2);
    day = substr(stime, 8, 2);
    return $day.".".$month.".".$year;
}
// Try to load and parse the specified GPX file
if (($gpx = simplexml load file($ SERVER["DOCUMENT ROOT"].$gpxPath)) ===
   FALSE)
    die;
// Figure out the absolute path to the icons and the photos
$url = parse url($ SERVER["SERVER NAME"].$ SERVER["REQUEST URI"]);
$iconPath = "http://".dirname($url["path"])."/icons/";
// Make a list of available cache photos
/*$photoPath = dirname($ SERVER["DOCUMENT ROOT"].$gpxPath)."/photos/";
$externalPhotoPath = "http://".dirname($ SERVER NAME"].$gpxPath)
   ."/photos/";
photos = array();
dir = dir(photoPath);
while (false !== (\$entry = \$dir -> read()))
```

```
if (!is dir($photoPath."/".$entry))
                   if (substr($entry, -strlen("thumb.jpg")) != "thumb.jpg")
                             \frac{1}{2}  sphotos [] =   entry;
$dir->close();*/
// Write the KML header, including icon styles for found and not found
        caches
kml = "<?xml version = \"1.0\" encoding = \"UTF-8\"?>";
kml := "<kml xmlns=\"http://www.opengis.net/kml/2.2\" xmlns:gx=\"http://www.opengis.net/kml/2.2\" xmlns:gx=\"http://www.opengis.net/kml/2\" xmlns:gx=\"http://www.opengis.net/kml/2\" xmlns:gx=\"http://www.opengis.net/kml/2\" xmlns:gx=\"http://www.opengis.net/kml/2\" xmlns]
       www.google.com/kml/ext/2.2\" xmlns:kml=\"http://www.opengis.net/kml
       /2.2 \" \text{xmlns: atom} = \" \text{http:} / \www.w3. \text{org} / 2005 / \Atom \">";
kml := "<Document>";
$kml .=
                       "<name>Geocaches</name>";
$kml .=
                       "<Style id=\"sn treasure found\">";
$kml .=
                        "<IconStyle>";
kml =
                            < Icon > ";
$kml .=
                             "<href>http://www.stupidcoder.de/wiki/extensions/geocaching/
       icons/treasure found.png</href>";
kml =
                          "</Icon>";
$kml .=
                           "< scale > 1.0 < / scale >";
kml .=
                         "<hotSpot x=\"0.5\" y=\"0.5\" xunits=\"fraction \" yunits=\"
        fraction \langle "/>";
                       "</IconStyle>";
kml =
$kml .=
                       "</Style>";
$kml .=
                       "<Style id=\"sn treasure notfound\">";
kml =
                         \| < I \operatorname{conStyle} > \|;
ml = 
                           \ll Icon > ;
                             "<href>http://www.stupidcoder.de/wiki/extensions/geocaching/
$kml .=
       icons/treasure notfound.png</href>";
                            "</Icon>";
kml =
                            < scale > 1.0 < / scale > ";
kml =
\ .=
                            "<hotSpot x=\"0.5\" y=\"0.5\" xunits=\"fraction \" yunits=\"
        fraction \langle "/>";
ml = 
                        "</IconStyle>";
                       "</Style>";
kml .=
// Iterate over each waypoint in the GPX file, which each represents one
        geocache
foreach ($gpx->wpt as $key => $waypoint)
{
         // Create a little cache information box with information about the
                 cache...
         $cache = $waypoint->children("http://www.groundspeak.com/cache/1/0")->
                 cache->children("http://www.groundspeak.com/cache/1/0");
```

```
description = "img src=\"".$iconPath.
   cacheTypeIcons[(string)cache->type]." << td valign= "middle"
   "> von ". $cache->owner." (".reformatTime($waypoint->time).")
   description = "
   = "50% ">Grö & szlig; e: < img src = \"". $iconPath. $cacheSizeIcons
   [(string)$cache->container]."\">Schwierigkeit:<
   img src=\"".$iconPath."difficulty_".$cache->difficulty.".png\"></
   td > ";
description = "<a href=\"".$waypoint->url."\"
   target = || blank || > Mehr Informationen ... < / a>
   :<img src=\"". $iconPath." difficulty ". $cache->terrain.".
   png = // td > // tr > // table = ;
// ... any photos of this cache...
cachePhotos = array();
foreach ($photos as $key => $photo)
    if (substr($photo, 0, strlen($waypoint->name)) == (string)
       $waypoint->name)
       cachePhotos[] = 
if (count(\$cachePhotos) > 0)
    description = "<hr>";
foreach ($cachePhotos as $key => $photo)
{
   createThumbnail($photoPath.$photo, 0, 40);
    $description .= "<a href=\"".$externalPhotoPath.$photo."\" target
       = \" blank \" > < img src = \" " .  $externalPhotoPath . $photo . " . 0 x40 .
       thumb.jpg\" border=\"0\"></a>&nbsp;";
}
// ... all the logs from this cache...
if (count(scache->logs->children("http://www.groundspeak.com/cache/1/0))
   ")->log) > 0)
   description = '<hr>';
foreach ($cache->logs->children("http://www.groundspeak.com/cache/1/0"
   ) \rightarrow \log as key \implies cacheLog
{
    $logInfo = $cacheLog->children("http://www.groundspeak.com/cache
       /1/0");
    $description .= "<b>".reformatTime($logInfo->date).":</b> ";
    $description .= "<img src=\"".$iconPath.$eventTypeIcons[(string)
       \log Info \rightarrow type ]. " " = ";
    $description .= " ".$logInfo->text."";
}
// ... and finally put it into a KML placemark
```

```
kml := "<Placemark>";
    $kml .= "<name><![CDATA[".$cache->name." (".$waypoint->name.")]]></</pre>
       name>";
    $kml .= "<styleUrl>#sn treasure found</styleUrl>";
    $kml .= "<description><![CDATA[".$description."]]></description>";
    kml = "<Point>";
    $kml .= "<coordinates>".$waypoint["lon"].",".$waypoint["lat"].",0
       coordinates > ";
    kml := "</Point>";
    kml := "</Placemark>";
}
// Close the KML document and send it to the client
kml := "</Document>";
kml := "</kml>";
echo $kml;
// Save a copy of the KML file for caching purposes
$kmlFile = fopen($cachedKMLPath, "w");
fwrite($kmlFile, $kml);
fclose($kmlFile);
?>
```

Listing 104: getRoutes GPX-Dateien konvertieren

## $10.6 \ google MapFiles/set Routes$

```
<?php
defined(' JEXEC') or die('Restricted access');
include ("googleMapFiles/APIkey/googleAPIkey.php");
echo("<!DOCTYPE html PUBLIC \"-//W3C//DID XHTML 1.0 Strict//EN\" \"http://
            www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">");
?>
< html \ xmlns = "http://www.w3.org/1999/xhtml" \ xmlns:v = "urn:schemas-microsoft-indication of the second seco
            \operatorname{com}: \operatorname{vml}">
               <head>
                             <\!\!meta http-equiv="content-type" content="text/html; charset=utf-8"
                                          \langle \rangle
                             <title>Google Maps-API: JavaScript-Beispiel</title>
                             <script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=</pre>
                                            <?php echo $google api key; ?>" type="text/javascript">
                             </script>
                             <style type="text/css">
                                              @import url("http://www.google.com/uds/css/gsearch.css");
                                              @import url("http://www.google.com/uds/solutions/localsearch/
                                                          gmlocalsearch.css");
```

```
/* needed for the localsearch */
        </style>
        <script src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0"</pre>
            type="text/javascript">
            // needed for the localsearch
        </script>
        <script src="http://www.google.com/uds/solutions/localsearch/</pre>
            gmlocalsearch.js" type="text/javascript">
            // needed for the localsearch
        </script>
        <script src="googleMapFiles/setRoutes/contextmenu.js" type="text/</pre>
            javascript">
        </script>
        <script src="googleMapFiles/setRoutes/map functions.js" type="text</pre>
            /javascript">
        </script>
    </head>
    <body>
        <?php include ("overlay.php"); ?>
        <div id="map">
        </div>
    </body>
</html>
```



```
var centerLatitude = 47.000; //google.loader.ClientLocation.latitude;
var centerLongitude = 10.000; //google.loader.ClientLocation.longitude;
var startZoom = 7;
var map;
function init(){
    // display Google Maps only, if the browser is able to handle GMap API
    if (GBrowserIsCompatible()) {
        // Dem Div-Container mit der ID 'map' die Karte zuweisen
        map = new GMap2(document.getElementById("map"));
        //\ {\rm centers} the map to the given lat, lng values and zooms it to
           the given value
        map.setCenter(new GLatLng(centerLatitude, centerLongitude),
           startZoom);
        // adding Google's localsearch
        map.addControl(new google.maps.LocalSearch(), new GControlPosition
           (G ANCHOR BOTTOM RIGHT, new GSize(130, 20)));
        // adding the minimap
        map.addControl(new GOverviewMapControl());
        // the most recent MapControl
```
```
var customUI = map.getDefaultUI();
        customUI.maptypes.hybrid = true;
        map.setUI(customUI);
        // contextmenu
        map.addControl(new ContextMenu(map));
        marker = new GMarker(new GLatLng(centerLatitude, centerLongitude),
            ł
            draggable: true
        });
        GEvent.addListener(marker, "dragend", function(){
            document.getElementById('lat').value = marker.getPoint().lat()
                ;
            document.getElementById('lng').value = marker.getPoint().lng()
                ;
        });
        map.addOverlay(marker);
    }
    else {
        alert ("Leider ist Ihr Browser mit der Google-Maps API nicht
           kompatibel!");
    }
}
function showAddress(address){
    if (geocoder) {
        geocoder.getLatLng(address, function(point){
            if (!point) {
                alert(address + " not found");
            }
            else {
                map.setCenter(point, 13);
                var marker = new GMarker(point);
                map.addOverlay(marker);
                marker.openInfoWindowHtml(address);
            }
        });
    }
}
function initAll() {
```

```
draginit(); // drag & drop overlay
init(); // google maps
}
// to avoid writting the onload() function in the <body> tag
// it's better to start the functions within this script...
window.onload = initAll;
// unloading map and free memory
```

```
window.onunload = GUnload;
```

Listing 106: setRoutes Kartenfunktionen

```
<script type="text/javascript" src="googleMapFiles/setRoutes/drag.js">
   script >
<div id="overlay">
    <div onmousedown="dragstart(document.getElementById('overlay'))" id="</pre>
       innerHead">
        (click to drag)
    </div>
    <div onclick="minimize(document.getElementById('innerWrapper'))" id="</pre>
       innerButton">
        <img src="googleMapFiles/setRoutes/images/button.png" width="30px"</pre>
           >
    </div>
    <div id="innerWrapper">
        <div id="innerCont">
        <?php
        require ("googleMapFiles/setRoutes/uploadForm.php");
        ?>
        </div>
    </div>
</div>
```

Listing 107: setRoutes Overlay für Eingabemaske

```
// http://aktuell.de.selfhtml.org/artikel/javascript/draganddrop/
//Das Objekt, das gerade bewegt wird.
var dragobjekt = null;
// Position, an der das Objekt angeklickt wurde.
var dragx = 0;
var dragy = 0;
// Mausposition
var posx = 0;
```

```
var posy = 0;
var tmpHeight;
var flag = false;
function draginit(){
    // Initialisierung der Ueberwachung der Events
    document.onmousemove = drag;
    document.onmouseup = dragstop;
}
function dragstart(element){
    //Wird aufgerufen, wenn ein Objekt bewegt werden soll.
    dragobjekt = element;
    dragx = posx - dragobjekt.offsetLeft;
    dragy = posy - dragobjekt.offsetTop;
}
function dragstop(){
    //Wird aufgerufen, wenn ein Objekt nicht mehr bewegt werden soll.
    dragobjekt = null;
}
function drag(ereignis){
    //Wird aufgerufen, wenn die Maus bewegt wird und bewegt bei Bedarf das
        Objekt.
    posx = document.all ? window.event.clientX : ereignis.pageX;
    posy = document.all ? window.event.clientY : ereignis.pageY;
    if (dragobjekt != null) {
        dragobjekt.style.left = (posx - dragx) + "px";
        dragobjekt.style.top = (posy - dragy) + "px";
    }
}
function minimize(element){
    if (flag = false) {
        tmpHeight = element.offsetHeight;
        //Effect.BlindUp(element, { duration: 3.0 });
        element.style.height = "0px";
        flag = true;
    else if (flag == true) {
        element.style.height = tmpHeight + "px";
        flag = false;
```

}

}

Listing 108: *setRoutes* Eingabemaske

```
<?php
defined('_JEXEC') or die('Restricted access');
require_once("googleMapFiles/db/db_connect.php");
?>
<span>Route hochladen</span>
<br/>
<br/>
<form action="<?php echo $_SERVER['REQUEST_URI']; ?>" method="POST"
   enctype="multipart/form-data">
    <table class="uploadForm" border="0">
        <\!{
m tr}\!>
            Route:
            <input type="file" name="route">
            <\!\!/\,t\,r\!>
        <\!\mathrm{tr}\!>
            Name:
            <\!\! input type="text" name="title" value="<\!\!?php echo \g_POST['
                     title ']; ?>">
            <\!/{
m tr}>
        < {
m tr} >
            Quelle:
            <\!\! input type = \!\!"text" name = \!\!"source" value = \!\!"<\!\!?php echo \$_POST
                     ['source']; ?>">
            <\!\!/\,t\,r\!>
        < {
m tr} >
             
            <\!/{
m tr}>
```

```
Beschreibung:
            < \mathrm{tr} >
            <td colspan="2">
                <textarea name="description" cols="38" rows="5"><?php echo
                    $ POST['description']; ?></textarea>
            <\!\mathrm{tr}>
            <td colspan="2">
                <\!\! input type = "submit" value = "Hochladen" >
            <\!/{
m tr}>
   </form>
<?php
// assign variables
$route = $_FILES['route'];
$title = $_POST['title'];
$source = $ POST['source'];
$description = $_POST['description'];
if ($route != "") {
    if (substr($route['name'], -3) = strtolower("kml")){
        // assign auto incremented names for uploaded tracks
        $query = mysql_query("SELECT max(id_geo) FROM db_routes");
        $value = mysql fetch array($query);
        num = value[0] + 1;
        name = "track". num.". kml";
        $target_path = "googleMapFiles/getRoutes/routes/".$name;
        if(move_uploaded_file($route['tmp_name'], $target_path)){
            echo "geht";
            $query = mysql query("INSERT INTO 'db routes' ('name', 'text',
                'link', 'route') VALUES
            ('".$title."', '".$description."', '".$source."', '".$name."')
               ");
        } else {
            echo "geht nicht";
```

```
} else {
     echo "<br/>Die Datei muss vom Typ KML sein!<br/>";
} else {
     echo "Bitte wählen Sie eine Route aus!";
}
/*
$query = mysql_query("SELECT id_geo FROM db_routes");
while($value = mysql_fetch_array($query)){
     echo $value['id_geo']."<br/>br>";
}
*/
```

Listing 109: setRoutes Uplaod-Formular

## 10.7 googleMapFiles/getGPXfile

```
<?php
include ("googleMapFiles/APIkey/googleAPIkey.php");
echo ("<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://
   www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">");
?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
   \operatorname{com}: \operatorname{vml}">
    <head>
        <title>Google Maps: GPX File</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
           >
        <link rel="stylesheet" type="text/css" href="css/main.css">
        <script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=</pre>
            ABQIAAAAVTrbgScTINO22y4XJhxP2BSDrGc-
            bfsgZsP0yPdbh6Jblet42hR5n3VCfJKTyBNnKKK3eX qX0updQ" type="text
            /javascript">
        </script>
        <style type="text/css">
            @import url("http://www.google.com/uds/css/gsearch.css");
            @import url("http://www.google.com/uds/solutions/localsearch/
                gmlocalsearch.css");
            /* needed for the localsearch */
        </style>
```

```
<script src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0"</pre>
           type="text/javascript">
            // needed for the localsearch
       </\operatorname{script}>
       <script src="http://www.google.com/uds/solutions/localsearch/</pre>
           gmlocalsearch.js" type="text/javascript">
            // needed for the localsearch
       </script>
       <script type="text/javascript"><?php echo ("var key = \"".
           google api key." \setminus ";"); ?></script>
       <script type="text/javascript" src="googleMapFiles/getGPXfile/gpx/
           GM Utils/GPX2GM. js ">
            // http://www.j-berkemeier.de/Touren/GPXViewer.html
       </script>
   </head>
   <body>
       <table border="0" cellpadding="0" cellspacing="10">
            <div id="map" class="gpxview:googleMapFiles/getGPXfile</pre>
                       /gpx/Beispiel1.gpx" style="width: 500px; height
                       :500px">
                    </div>
                <\!div id="map_hp" style="width:500px; height:300px">
                    </div>
                <\!\mathrm{tr}\!>
                <div id="map_vp" style="width:500px; height:300px">
                    </div>
                <div id="map_sp" style="width:500px; height:300px">
                    </div>
                <\!/\,t\,r>
       </body>
</html>
```



```
// Darstellung von GPS-Daten aus einer GPX-Datei in Google Maps
// Version 4.1
// 27. 11. 2009 JA<sup>1</sup>/<sub>4</sub>rgen Berkemeier
// www.j-berkemeier.de
if (typeof (GPXVIEW_Debuginfo) == "undefined") var GPXVIEW_Debuginfo = false;
document.write('<script src="http://maps.google.com/maps?file=api&amp;v=2&
   amp; key='+key+'" type="text/javascript"><\/script>');
function makeMap(ID) {
  var lmCntrl = (typeof(LargeMapControll)!="undefined") ? LargeMapControll
      : false;
  var oMCntrl = (typeof(OverviewMapControl)!="undefined") ?
     OverviewMapControl : false;
  var legende = (typeof(Legende)!="undefined") ? Legende : true;
  var legende_trk = (typeof(Legende_trk)!="undefined") ? Legende_trk :
     true;
  var legende_rte = (typeof(Legende_rte)!="undefined") ? Legende_rte :
     true;
  var legende wpt = (typeof(Legende wpt)!="undefined") ? Legende wpt :
     true;
  var t verbinden = (typeof(Tracks verbinden)!="undefined") ?
     Tracks verbinden : false;
  var rdSpeed = (typeof(readSpeed)!="undefined") ? readSpeed : true;
  var tOver = (typeof(TrackOver)!="undefined") ? TrackOver : true;
  var twidth = 2.0;
  var rwidth = 2.0;
  var widtho = 3.0;
  var topac = 0.8;
  var ropac = 0.8;
  var tcols = new Array ("#ff0000", "#00ff00", "#0000ff", "#aaaa00", "#ff00ff",
     "#00ffff","#000000");
  "#008080","#808080");
  var colo = "\#000000";
  var icons = \{
    scenic: { image: "scenic.png", iconSize: [21.0,31.0],
              shadow: "shadow.png", shadowSize: [52.0, 29.0],
              iconAnchor: [5.0,30.0], infoWindowAnchor: [10.0,5.0] },
    marker: { image: "marker.gif", iconSize: [11.0, 11.0], iconAnchor: [5.0, 5.0]
        }
  }
  var makeIcon = function(SymName) {
```

```
var icon = new GIcon();
   if (icons [SymName]) {
    if (icons [SymName].image) icon.image = GPX2GM Path + icons [SymName].
       image;
    if (icons [SymName].iconSize) icon.iconSize = new GSize(icons [SymName
        ]. iconSize [0], icons [SymName]. iconSize [1]);
    if (icons [SymName].shadow) icon.shadow = GPX2GM Path + icons [SymName
        . shadow;
    if (icons [SymName].shadowSize) icon.shadowSize = new GSize(icons [
       SymName].shadowSize[0], icons[SymName].shadowSize[1]);
    if (icons [SymName].iconAnchor) icon.iconAnchor = new GPoint (icons [
       SymName].iconAnchor[0],icons[SymName].iconAnchor[1]);
    if (icons [SymName].infoWindowAnchor)
      icon.infoWindowAnchor = new GPoint(icons[SymName].infoWindowAnchor
          [0], icons [SymName]. infoWindowAnchor [1]);
  }
  return icon;
} // makeIcon
JB GM Info(ID, "makeMap", false);
 var polylineEncoder = new PolylineEncoder();
var dieses = this;
var id = ID;
var id hp = ID+" hp";
var id sp = ID+" sp";
var id vp = ID+" vp";
var load = false
var latmin=1000, latmax=-1000, lonmin=1000, lonmax=-1000;
var zoom = 1;
var osm mapnik map, osm tah map, osm cycle map;
var fname, maptype;
var tracks, waypoints, routes, tracklens, routlens, alledaten;
var routeNames, trackNames;
var GPX2GM Path="";
var scr = document.getElementsByTagName("script");
var hp_xtext = "Strecke in km";
var vp xtext = "Strecke in km";
var sp xtext = "Strecke in km";
var hp ytext ="H<br />&ouml;<br />h<br />e<br />&nbsp;<br />in<br />&
   nbsp; < br />m";
var sp ytext ="Stg.<br />&nbsp;<br />in<br />&nbsp;<br />%";
var vp ytext ="V<br />&nbsp;<br />i<br />knbsp;<br />km/h";
for (var i=0;i<scr.length;i++) if (scr[i].src && scr[i].src.length) {
  var path = scr[i].src;
  var pos = path.search("GPX2GM.js");
  if (pos!=-1) {
```

```
GPX2GM Path = path.substring (0, pos);
      break;
   }
 }
 var icon = makeIcon("scenic") ;
 var markericon = makeIcon("marker") ;
 var movemarker;
 var markerinfo = document.createElement("div");
 markerinfo.style.position = "absolute";
 markerinfo.style.visibility = "hidden";
 markerinfo.style.border = "1px solid black";
 markerinfo.style.backgroundColor = "white";
 var div = document.getElementById(id);
 var w = div.offsetWidth;
 var h = div.offsetHeight;
 var MapHead = document.createElement("div");
 MapHead.id = "map_head"+id;
 MapHead.style.margin = 0;
 MapHead.style.padding = 0;
// MapHead.style.fontSize = "0.8em";
// MapHead.style.lineHeight = "1.5em";
 MapHead.appendChild(document.createTextNode(": "));
 var mapdiv = document.createElement("div");
 mapdiv.id = "map "+id;
 mapdiv.style.width = w+"px";
  while(div.hasChildNodes()) div.removeChild(div.firstChild);
  if (!legende) MapHead.style.display="none";
  div.appendChild(MapHead);
  div.appendChild(mapdiv);
  if (legende) mapdiv.style.height = h-mapdiv.offsetTop+MapHead.offsetTop+
     "px";
               mapdiv.style.height = h+"px";
  else
  var map = new GMap2(document.getElementById("map "+id));
  if(lmCntrl) map.addControl(new GLargeMapControl());
                       map.addControl(new GSmallMapControl());
  else
  if (oMCntrl) map.addControl(new GOverviewMapControl());
 map.addMapType(G PHYSICAL MAP);
 var copyrightCollection = new GCopyrightCollection ('\&copy; 2009 <a href
     ="http://www.openstreetmap.org/">OpenStreetMap</a>');
  copyrightCollection.addCopyright(new GCopyright(1, new GLatLngBounds(new
     GLatLng(-90,-180), new GLatLng(90,180)), 0, '(<a rel="license" href="
     http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>););
 var tilelayers mapnik = new Array();
  tilelayers_mapnik [0] = new GTileLayer(copyrightCollection, 0, 18);
  tilelayers mapnik [0]. get Tile Url = function(a, z) \{ return "http://tile.
     openstreetmap.org/" + z + "/" + a.x + "/" + a.y + ".png"; };
```

```
tilelayers mapnik[0].isPng = function() { return true; };
  tilelayers mapnik [0]. getOpacity = function () { return 1.0; };
 osm_mapnik_map = new GMapType(tilelayers_mapnik,new GMercatorProjection
     (19), "OSM Mapnik", { urlArg: 'mapnik', linkColor: '#000000' });
 map.addMapType(osm mapnik map);
 var tilelayers_tah = new Array();
  tilelayers tah[0] = new GTileLayer(copyrightCollection, 0, 17);
  tilelayers_tah [0].getTileUrl = function(a, z) { return "http://tah.
     openstreetmap.org/Tiles/tile/" + z + "/" + a.x + "/" + a.y + ".png";
      };
  tilelayers tah[0].isPng = function() { return true; };
  tilelayers tah[0].getOpacity = function() { return 1.0; };
 osm tah map = new GMapType(tilelayers tah, new GMercatorProjection(19), "
     OSM T&H", { urlArg: 'tah', linkColor: '#000000' });
 map.addMapType(osm tah map);
  copyrightCollection = new GCopyrightCollection ('© 2009 <a href=""".
     http://www.opencyclemap.org/">OpenCycleMap</a> <a href="http://www.
     openstreetmap.org/">OpenStreetMap</a>');
  copyrightCollection.addCopyright(new GCopyright(1, new GLatLngBounds(new
     GLatLng(-90,-180), new GLatLng(90,180)), 0, '(<a rel="license" href="
     http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>););
 var tilelayers_cycle = new Array();
  tilelayers cycle [0] = new GTileLayer(copyrightCollection, 0, 17);
  tilelayers cycle [0]. getTileUrl = function(a, z) { return "http://andy.}
     sandbox.cloudmade.com/tiles/cycle/" + z + "/" + a.x + "/" + a.y + ".
     png"; };
  tilelayers cycle[0].isPng = function() { return true; };
  tilelayers cycle[0].getOpacity = function() { return 1.0; };
 osm cycle map = new GMapType(tilelayers cycle, new GMercatorProjection
     (19), "OSM Cycle", { urlArg: 'cycle', linkColor: '#000000'});
 map.addMapType(osm cycle map);
 map.addControl(new GMenuMapTypeControl());
// map.addControl(new GMapTypeControl());
// map.addControl(new GHierarchicalMapTypeControl());
// map.addMapType(G_SATELLITE_3D_MAP); // benï;<sup>1</sup>/<sub>2</sub>tigt Plugin
 map.addControl(new GScaleControl());
 map.enableScrollWheelZoom();
 map.getPane(G MAP FLOAT PANE).appendChild(markerinfo);
// GEvent.addListener(map, "maptypechanged", function() { dieses.rescale() });
 var hp = document.getElementById(id hp);
 var hp diag;
  if(hp) {
    hp diag = new plot(id hp, "\mathbf{x}", "h");
    if (hp.className & hp.className.search("no_x")!=-1) hp_xtext="";
    JB GM Info(id, "Hï; <sup>1</sup>/<sub>2</sub>henprofil, ID: "+id hp, false);
  }
```

```
var sp = document.getElementById(id sp);
var sp diag;
if(sp) 
  sp diag = new plot(id sp, "x", "s");
  if (sp.className \&\& sp.className.search("no x")!=-1) sp xtext="";
 JB_GM_Info(id, "Steigungsplot, ID: "+id_sp, false);
}
var vp = document.getElementById(id vp);
var vp diag;
if(vp) {
  vp diag = new plot(id vp, "x", "v");
  if (vp.className && vp.className.search("no x")!=-1) vp xtext="";
 JB GM Info(id, "Geschwindigkeitsprofil, ID: "+id vp, false);
}
this.Spur = function(fn, mpt) {
  JB_GM_Info(id, "Spur, Filename: "+fn, false);
  if (mpt="osm mapnik map") maptype = osm mapnik map;
  else if (mpt="osm_tah_map") maptype = osm_tah_map;
  else if (mpt="osm cycle map") maptype = osm cycle map;
  else maptype = mpt;
  if (fname!=fn) {
    fname = fn;
    JB GM Info(id, "Lade Datei "+fname, false);
    GDownloadUrl(fname, function(data, responseCode) {
      if (responseCode != 200 && responseCode != 0 ) {
         JB GM Info(id, "Beim Oeffnen der Datei "+fname+" ist der Fehler
             "+responseCode+" aufgetreten!", true);
        return;
      }
      dieses.parseGPX(data) ;
      dieses.setMapHead();
      zoom = map.getBoundsZoomLevel(new GLatLngBounds(new GLatLng(latmin
          , lonmin), new GLatLng(latmax, lonmax))); // sw, ne
      dieses.rescale();
      map.setMapType(maptype);
      dieses.checkBoxes()
    });
  }
  else {
    dieses.checkBoxes()
  }
} // Spur
var chkwpt, chktrk, chkrt;
this.setMapHead = function() 
  JB GM Info(id, "setMapHead", false);
```

}

```
var name = fname.replace(/.+//,"");
 MapHead.innerHTML = name+": ";
  if (waypoints.length) {
    if (waypoints.length==1) var texte=new Array ("Wegpunkt"+String.
       fromCharCode(160));
    else if (waypoints.length>1) var texte=new Array ("Wegpunkte"+String.
       fromCharCode(160));
   chkwpt = new JB CheckBoxGroup(MapHead.id,texte,ID+" wpt",["black"],
       legende wpt, dieses.checkBoxes);
 }
  if (tracks.length) {
    var texte=new Array()
    if (tracks.length==1)
      texte[0] = "Track ("+Number(tracklens[0].toPrecision(10).toString
         (10) + "km) "+String.fromCharCode (160);
    else if(tracks.length>1) {
      texte[0] = "Tracks ("+Number(tracklens.sum().toPrecision(10).
         toString(10) + "km) "+String.fromCharCode(160);
      for (var i=0; i < tracks.length; i++) texte [i+1] = trackNames [i]+" ("+
         Number(tracklens[i].toPrecision(10).toString(10))+"km)";
    }
    chktrk = new JB_CheckBoxGroup(MapHead.id, texte, ID+"_trk", tcols,
       legende trk, dieses.checkBoxes);
  }
  if(routes.length) {
    var texte=new Array()
    if (routes.length==1)
      texte[0] = "Route ("+Number(routlens[0].toPrecision(10).toString)]
         (10) + "km) "+String.fromCharCode (160);
    else if (routes.length >1) {
      texte[0] = "Routen ("+Number(routlens.sum().toPrecision(10).
         toString(10) + "km) "+String.fromCharCode(160);
      for (var i=0; i < routes . length; i++) texte [i+1] = routeNames [i]+" ("+
         Number(routlens[i].toPrecision(10).toString(10))+"km)";
    }
    chkrt = new JB CheckBoxGroup(MapHead.id, texte, ID+" rt", rcols,
       legende rte, dieses.checkBoxes);
 }
} // setMapHead
this.checkBoxes = function(obj,ele) {
 var what="";
  if (chkwpt && chkwpt.status [0]) what += "w";
  if (chktrk && chktrk.status [0]) what += "t";
  if (chkrt && chkrt.status [0]) what += "r";
  dieses.show(what);
```

```
this.parseGPX = function(data) {
 JB GM Info(id, "parseGPX", false);
 var entf = new this. Entfernung();
  tracks = new Array();
  alledaten = new Array();
  trackNames = new Array();
  tracklens = new Array();
 load = false;
 latmin = 1000; latmax = -1000; lonmin = 1000; lonmax = -1000;
 var xml = GXml. parse(data);
 var trk = xml.documentElement.getElementsByTagName("trk"); // Tracks
 JB GM Info(id, trk.length +"Tracks gefunden", false);
  for (var k=0;k<trk.length;k++) {
    var trkseg = trk[k].getElementsByTagName("trkseg"); //
       Trackssegmente
    if (trkseg.length == 0) trkseg = [trk[k]];
    var name = trk[k].getElementsByTagName("name");
    if (name.length && name[0].firstChild && name[0].firstChild.length)
      trackNames[k] = name[0]. firstChild.data;
    else
      trackNames[k] = "Track "+k;
   JB_GM_Info(id, trkseg.length+"Tracksegmente in Track "+k+" gefunden",
       false);
    for (var \ j=0; j < trkseg.length; j++) {
      var trkpts = trkseg[j].getElementsByTagName("trkpt"); //
         Trackpunkte
      var trkptslen = trkpts.length;
      var track = new Array();
      var daten = new Array();
      var x0 = 0;
      if (t verbinden & k>0) x0 = tracklens.sum();
      var tracklen = 0;
      var hflag=hp || sp, tflag=vp, vflag=vp, h, t, v;
     JB GM Info(id, trkptslen+" Trackpunkte in Tracksegment "+j+" in
         Track "+k+" gefunden", false);
      for (var i=0;i<trkptslen;i++) { // Trackdaten erfassen
        var lat = parseFloat(trkpts[i].getAttribute("lat"));
        var lon = parseFloat(trkpts[i].getAttribute("lon"));
        track.push(new GLatLng(lat, lon));
        if (lat < latmin) latmin=lat; if (lat > latmax) latmax=lat;
        if(lon<lonmin) lonmin=lon; if(lon>lonmax) lonmax=lon;
        if (hflag && trkpts [i].getElementsByTagName("ele").length &&
           trkpts [i].getElementsByTagName("ele")[0].hasChildNodes())
          h = parseFloat(trkpts[i].getElementsByTagName("ele")[0].
              firstChild.data);
        else {
```

}

```
hflag = false;
      h = -1e6;
    }
    if (tflag && trkpts [i].getElementsByTagName("time").length &&
        trkpts[i].getElementsByTagName("time")[0].hasChildNodes())
      t = JB_utc2msec(trkpts[i].getElementsByTagName("time")[0].
          firstChild.data);
    else {
      tflag = false;
      t = -1;
    }
    if (vflag && rdSpeed && trkpts [i].getElementsByTagName("speed").
        length && trkpts[i].getElementsByTagName("speed")[0].
        hasChildNodes())
      v = parseFloat(trkpts[i].getElementsByTagName("speed")[0].
          firstChild.data);
    else {
      v = -1;
      vflag = false;
    }
    if (i==0) entf.init(lat, lon);
    var dx = entf.rechne(lat, lon);
    tracklen += dx;
    daten.push(\{lat:lat, lon:lon, x:tracklen+x0, t:t, h:h, dx:dx, v:v\});
  }
  if(hflag) {
    daten = JB_smoth(daten, "x", "h", "hs", trkptslen/50);
    daten = JB diff(daten, "\mathbf{x}", "\mathbf{hs}", "\mathbf{s}", 0.1);
    daten = JB smoth(daten, "x", "s", "s", trkptslen/50);
  }
  else {
    hp = false;
    sp = false;
  }
  if (tflag && !vflag) {
    daten = JB_smoth(daten, "t", "x", "xs", trkptslen/200);
    daten = JB diff(daten, "t", "xs", "v", 3600000);
    daten = JB_smoth(daten, "x", "v", "v", trkptslen/200);
  }
  else if (!tflag && !vflag) {
    vp = false;
  }
if(hflag) JB_GM_Info(id, "Hï¿<sup>1</sup>/<sub>2</sub>hendaten gefunden", false); else
   JB GM Info(id, "Keine Hï; <sup>1</sup>/<sub>2</sub>hendaten gefunden", false);
```

```
153
```

```
if (tflag) JB GM Info(id, "Zeitdaten gefunden", false); else JB GM Info
     (id, "Keine Zeitdaten gefunden", false);
  if (vflag) JB_GM_Info(id, "Geschwindigkeitsdaten gefunden", false);
     else JB GM Info(id, "Keine Geschwindigkeitsdate gefunden", false);
  alledaten.push(daten);
  tracks.push(track);
  tracklens.push(Math.round(tracklen*10)/10);
}
var rte = xml.documentElement.getElementsByTagName("rte"); // Routen
JB GM Info(id, rte.length +" Routen gefunden", false);
routes = new Array();
routeNames = new Array();
routlens = new Array();
for (var \ j=0; j< rte. length; j++) {
  var rtepts = rte[j].getElementsByTagName("rtept");
  JB_GM_Info(id, rtepts.length +" Zwischenziele gefunden", false);
  var route = new Array();
  var routlen = 0;
  var name = rte[j].getElementsByTagName("name");
  if (name.length && name[0].firstChild && name[0].firstChild.length)
    routeNames [j] = name [0]. firstChild.data;
  else
    routeNames[j] = "Route "+j;
  for (var i=0;i<rtepts.length;i++) { // Zwischenziele
    var lat = parseFloat(rtepts[i].getAttribute("lat"));
    var lon = parseFloat(rtepts[i].getAttribute("lon"));
    if (i==0) entf.init(lat, lon);
    routlen += entf.rechne(lat,lon);
    if (lat < latmin) latmin=lat; if (lat > latmax) latmax=lat;
    if(lon<lonmin) lonmin=lon; if(lon>lonmax) lonmax=lon;
    route.push(new GLatLng(lat,lon));
    var ext = rtepts[i].getElementsByTagName("extensions");
    if (ext.length) {
      var rpts = JB GetElementsByTagNameNS(ext[0], "gpxx", "rpt"); //
          Routenpunkte
      JB_GM_Info(id, rpts.length +" Routenpunkte (Garmin) gefunden",
          false);
      for (var k=0; k< rpts.length; k++) {
        var lat = parseFloat(rpts[k].getAttribute("lat"));
        var lon = parseFloat(rpts[k].getAttribute("lon"));
        routlen += entf.rechne(lat,lon);
        if (lat < latmin) latmin=lat; else if (lat > latmax) latmax=lat;
        if (lon<lonmin) lonmin=lon; else if (lon>lonmax) lonmax=lon;
        route.push(new GLatLng(lat,lon));
      }
```

```
}
    routes.push(route);
    routlens.push(Math.round(routlen*10)/10);
 }
 var wpts = xml.documentElement.getElementsByTagName("wpt"); //
     Waypoints
 JB GM Info(id, wpts.length +" Wegpunkte gefunden", false);
 waypoints = new Array();
  for (var i=0;i<wpts.length;i++) { // Wegpunktdaten
    var wpt = wpts [i];
   var lat = parseFloat(wpt.getAttribute("lat"));
    var lon = parseFloat(wpt.getAttribute("lon"));
    if (lat < latmin) latmin=lat; if (lat > latmax) latmax=lat;
    if (lon<lonmin) lonmin=lon; if (lon>lonmax) lonmax=lon;
    var waypoint = new Object();
    waypoint.lat = lat;
    waypoint.lon = lon;
    waypoint.name = "";
    waypoint.cmt = "";
    waypoint.desc = "";
    var name = wpt.getElementsByTagName("name");
    var cmt = wpt.getElementsByTagName("cmt");
    var desc = wpt.getElementsByTagName("desc");
    if (name.length && name[0].firstChild && name[0].firstChild.length)
      waypoint.name = name[0].firstChild.data;
    if (cmt.length && cmt[0].firstChild && cmt[0].firstChild.length)
      waypoint.cmt = cmt[0].firstChild.data;
    if (desc.length && desc[0].firstChild && desc[0].firstChild.length)
      waypoint.desc = desc[0].firstChild.data;
    waypoints.push(waypoint);
 }
 load = true;
} // parseGPX
this.showWpts = function() {
  if (load) {
    for (var i=0;i<waypoints.length;i++) {</pre>
      var waypoint = waypoints[i];
      if (checkImageName(waypoint.name)) {
        map.addOverlay(createImgMarker(waypoint));
      }
      else if (waypoint.name.length || waypoint.cmt.length)
        map.addOverlay(createTxtMarker(waypoint));
      else
        map.addOverlay(new GMarker(new GLatLng(waypoint.lat, waypoint.lon
           )));
```

```
}
} // showWpts
this.showTracks = function() {
  if (load) {
    if (alledaten.length>1) {
      for (var i=0; i< alled aten.length; i++) {
        var daten = alledaten [i];
        if (daten.length && chktrk.status[i+1]) {
          if (hp) hp diag.scale(daten);
          if (sp) sp diag.scale(daten);
          if (vp) vp diag.scale(daten);
        }
      }
    }
    else if(alledaten.length==1) {
      var daten = alledaten [0];
      if(daten.length) {
        if (hp) hp diag.scale(daten);
        if(sp) sp_diag.scale(daten);
        if (vp) vp diag.scale(daten);
      }
    }
    if (hp) hp diag.frame (50,35, hp xtext, hp ytext);
    if (sp) sp diag.frame (50,35, sp xtext, sp ytext);
    if (vp) vp_diag.frame(50,35,vp_xtext,vp_ytext);
    if (tracks.length>1) {
      for (var i=0;i<tracks.length;i++) if (chktrk.status[i+1]) {</pre>
        this.makePolyline(map, tracks[i], tcols[i%tcols.length], twidth,
            topac, widtho, colo, trackNames [i]+"<br />"+Number(tracklens [i
            ]. to Precision (10). to String (10))+"km");
        if (alledaten [i].length) {
          if (hp) hp diag.plot(alledaten[i], tcols[i%tcols.length]);
          if(sp) sp_diag.plot(alledaten[i],tcols[i%tcols.length]);
          if (vp) vp diag.plot(alledaten[i],tcols[i%tcols.length]);
        }
      }
      if(t verbinden) {
        var d t = new Array();
        for (var i=0; i < alledaten . length; i++) if (chktrk.status [i+1]) d t =
             d t.concat(alledaten[i]);
        if (d t.length) {
          if (hp) hp diag.markeron(d t, dieses.markerstart, dieses.
              markerstop, dieses.markermove, "Linie");
          if (sp) sp diag.markeron(d t, dieses.markerstart, dieses.
              markerstop, dieses.markermove, "Linie");
```

```
if (vp) vp diag.markeron(d t, dieses.markerstart, dieses.
              markerstop, dieses.markermove, "Linie");
        }
      }
    }
    else if(tracks.length==1) {
      if (chktrk.status [0]) {
        this.makePolyline(map, tracks [0], tcols [0], twidth, topac, widtho,
            colo, trackNames[0] + "<br />"+Number(tracklens[0].toPrecision
            (10).toString(10))+"km");
        if (alledaten [0].length) {
          if(hp) {
            hp diag.plot(alledaten[0], tcols[0]);
            hp diag.markeron(alledaten[0], dieses.markerstart, dieses.
                markerstop, dieses.markermove, "Linie");
          }
          if(sp) {
            sp diag.plot(alledaten[0], tcols[0]);
            sp_diag.markeron(alledaten[0], dieses.markerstart, dieses.
                markerstop, dieses.markermove, "Linie");
          }
          if(vp) {
            vp diag.plot(alledaten[0], tcols[0]);
            vp diag.markeron(alledaten[0], dieses.markerstart, dieses.
                markerstop, dieses.markermove, "Linie");
          }
        }
      }
    }
  }
} // showTracks
this.showRoutes = function() {
  if (load) {
    if (routes.length >1) {
      for (var i=0;i<routes.length;i++) if (chkrt.status[i+1])
        this.makePolyline(map,routes[i],rcols[i%tcols.length],rwidth,
            ropac, widtho, colo, routeNames [i]+"<br />"+Number(routlens [i].
            toPrecision(10) . toString(10)) + "km");
    }
    else if (routes.length==1) {
      if (chkrt.status [0])
        this.makePolyline(map,routes[0],rcols[0],rwidth,ropac,widtho,
            colo, routeNames[0] + " < br /> "+Number(routlens [0]. to Precision
            (10).toString(10))+"km");
    }
```

```
} // showRoutes
this.show = function(what) \{
     map.clearOverlays() ;
     if(hp) hp diag.clear();
     if (sp) sp diag.clear();
     if(vp) vp_diag.clear();
     if (what.search("w") != -1) dieses.showWpts();
     if (\text{what.search}("t") != -1) dieses.showTracks();
     if (\text{what.search}("r") != -1) dieses.showRoutes();
\} // show
this.rescale = function() {
     if (load) {
          map.setCenter(new GLatLng((latmax+latmin)/2, (lonmax+lonmin)/2), zoom
                   );
     }
} // rescale
this.Entfernung = function() {
     var ls, bs;
     this.init = function(b, l) {
          ls = l;
          bs = b;
     }
     this.rechne = function(b, 1) {
          var e = new \ GLatLng(bs, ls).distanceFrom(new \ GLatLng(b, l));
          1s = 1;
          bs = b;
          return e/1000;
     }
} // Entfernung
this.showtrackmarker = function() {
     JB GM Info(id, "showtrackmarker", false);
     movemarker = new GMarker(tracks [0][0], markericon);
     map.addOverlay(movemarker);
     markerinfo.style.visibility = "visible";
} // showtrackmarker
this.hidetrackmarker = function() {
    map.removeOverlay(movemarker);
     markerinfo.style.visibility = "hidden";
} // hidetrackmarker
this.settrackmarker = function(a) {
     map.removeOverlay(movemarker);
     movemarker = new GMarker(new GLatLng(a.lat,a.lon), markericon);
     map.addOverlay(movemarker);
     var point=map.getCurrentMapType().getProjection().fromLatLngToPixel().getProjection().fromLatLngToPixel().getProjection().fromLatLngToPixel().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection().getProjection(
             map.fromDivPixelToLatLng(new GPoint(0,0),true),map.getZoom());
```

```
var offset=map.getCurrentMapType().getProjection().fromLatLngToPixel(
     movemarker.getPoint(),map.getZoom());
 var anchor=movemarker.getIcon().iconAnchor;
  var width=movemarker.getIcon().iconSize.width;
 var height=markerinfo.clientHeight;
  var pos = new GControlPosition (G_ANCHOR_TOP_LEFT, new GSize (offset.x -
      point.x - anchor.x + width, offset.y - point.y - anchor.y - height)
     );
  pos.apply(markerinfo);
} // movetrackmarker
this.markerstart = function() {
 JB GM Info(id, "markerstart", false);
  dieses.showtrackmarker();
  if (hp) hp diag.showmarker("Linie");
  if(vp) vp diag.showmarker("Linie");
  if(sp) sp diag.showmarker("Linie");
} // markerstart
this.markerstop = function() {
  dieses.hidetrackmarker();
 JB GM Info(id, "markerstop", false);
  if (hp) hp diag.hidemarker();
  if(vp) vp_diag.hidemarker();
  if(sp) sp diag.hidemarker();
} // markerstop
this.markermove = function(p,a) {
  var info = "Strecke:  "+a.x.toFixed(1)+"km";
  if(hp) {
    info += "<br />H&ouml; he:&nbsp; "+Math.round(a.h)+"m";
    hp diag.setmarker(a, "Linie");
  }
  if(vp) {
    info += "<br />Geschw.:&nbsp; "+Math.round(a.v)+"km/h";
    vp diag.setmarker(a,"Linie");
  }
  if(sp) {
    info += "<br />Stg.:  "+Math.round(a.s)+"%";
    sp diag.setmarker(a, "Linie");
 }
  markerinfo.innerHTML = info;
  dieses.settrackmarker(a);
} // markermove
var createImgMarker = function(waypoint) {
  var marker = new GMarker(new GLatLng(waypoint.lat, waypoint.lon), icon);
 GEvent.addListener(marker, "click", function() {
    var bild = new Image();
    bild.onload = function() {
```

```
marker.openInfoWindowHtml("<img src='"+waypoint.name+"'\/><br\/>"+
            waypoint.cmt);
      };
      bild.src = waypoint.name;
    });
    return marker;
  } // createImgMarker
  var createTxtMarker = function(waypoint) {
    var marker = new GMarker(new GLatLng(waypoint.lat, waypoint.lon),{title
        :waypoint.name});
    GEvent.addListener(marker, "click", function() {
      marker.openInfoWindowHtml("<strong>"+waypoint.name+"<\/strong><br/>br\/>
          "+waypoint.cmt);
    });
    return marker;
  } // createTxtMarker
  var checkImageName = function(url) {
    var ext = url.substr(url.lastIndexOf(".")+1).toLowerCase();
    return (ext="jpg" || ext="jpeg" || ext="png" || ext="gif") ;
  } // checkImageName
  this.makePolyline = function (map, line, col, width, opac, widtho, colo, info) {
    // var PL = new GPolyline(line, col, width, opac);
    var PL = polylineEncoder.dpEncodeToGPolyline(line, col, width, opac);
    if(tOver) {
      GEvent.addListener(PL, "mouseover", function(latlng) {
        JB GM Info("", "Line over, "+this.getVertexCount()+" Punkte", false)
        this.setStrokeStyle({color:colo,weight:widtho});
      });
      GEvent.addListener(PL, "mouseout", function() {
        JB GM Info("","Line out", false);
        this.setStrokeStyle({color:col,weight:width});
      });
      GEvent.addListener(PL, "click", function(overlay, latlng) {
        JB GM Info("", "Line click "+overlay+" "+latlng, false);
        map.openInfoWindow(overlay, info);
      });
    }
    map.addOverlay(PL);
  } // makePolyline
} // makeMap
function JB GM Info(id, Infotext, alertflag) {
  if (GPXVIEW Debuginfo) GLog.write ("Map "+id+": "+Infotext);
  if (alertflag) alert (Infotext);
} // GM Info
```

```
function JB GetElementsByTagNameNS(ele, namespace, name) {
 var alltags = ele.getElementsByTagName("*");
 var tagname = namespace.toLowerCase()+":"+name.toLowerCase()
 var tags = new Array();
  for (var i=0; i<alltags.length; i++) if (alltags[i].nodeName.toLowerCase()=
     tagname) tags.push(alltags[i]);
 return tags;
} // JB GetElementsByTagNameNS(ele, namespace, name)
function JB CheckBoxGroup(id, Texte, Label, Farbe, def stat, clickFunc) {
  var dieses = this;
 var nbx = Texte.length;
  this.nboxen = nbx;
  this.status = new Array(nbx); for (var i=0;i<nbx;i++) this.status[i] =
     def stat ;
 var ele;
 var box=document.createElement("div");
 box.style.position = "absolute";
 box.style.display = "inline";
 box.style.height = "1.4em";
 box.style.overflow = "hidden";
 box.style.backgroundColor = "";
 box.style.zIndex = 1000;
 box.style.margin = "0";
 box.style.padding = "0";
 box.onmouseover = function() 
    this.style.height = "";
    this.style.overflow = "";
    this.style.backgroundColor = "white";
    this.style.paddingRight = "0.3em";
    this.style.paddingBottom = "0.2em";
  };
 box.onmouseout = function() 
    this.style.height = "1.4em";
    this.style.overflow = "hidden";
    this.style.backgroundColor = "";
    this.style.paddingRight = "";
    this.style.paddingBottom = "";
  };
  for (var i=0; i < nbx; i++) {
    ele = document.createElement("input");
    ele.type = "checkbox";
    ele.id = Label + i;
    ele.nr = i;
    if (i==0) ele.onclick = function () {
```

```
var l = nbx;
      var n = Label;
      var status = this.checked;
      dieses.status [this.nr] = status;
      for (var \ j=1; j<l; j++) {
        document.getElementById(n+j).checked=status;
        dieses.status[j] = status;
      }
      clickFunc(dieses, this);
    };
    else
              ele.onclick = function() {
      var l = nbx;
      var n = Label;
      var status = false;
      for (var \ j=1; j<l; j++) status \mid = document.getElementById(n+j).checked;
      document.getElementById (n+"0").checked = status;
      dieses.status [0] = status=true;
      dieses.status[this.nr] = this.checked;
      clickFunc(dieses, this);
    };
    box.appendChild(ele);
    ele.checked = def stat;
    ele=document.createElement("span");
    if (i=0 \&\& nbx==1) ele.style.color=Farbe [0];
    else if(i) ele.style.color=Farbe[(i-1)%Farbe.length];
    ele.appendChild(document.createTextNode(Texte[i]));
    box.appendChild(ele);
    if (i<Texte.length-1) box.appendChild (document.createElement("br"));
  }
  ele=document.getElementById(id);
  ele.appendChild(box);
  var spn=document.createElement("span"); // Platzhalter
  spn.appendChild(document.createTextNode(" X "+Texte[0]+" "));
  spn.style.visibility="hidden";
  ele.appendChild(spn);
} // JB CheckBoxGroup
function addEvent (obj, type, fn) {
   if (obj.addEventListener) {
      obj.addEventListener(type, fn, false);
   } else if (obj.attachEvent) {
      obj.attachEvent('on' + type, function () {
         return fn.call(obj, window.event);
      });
   }
} // addEvent
```

```
function JB utc2msec(utcdate) {
    var jahr = utcdate.substr(0,4);
    var monat = utcdate.substr(5,2)*1-1;
    var tag = utcdate.substr(8,2);
    var stunde = utcdate.substr(11,2);
    var minute = utcdate.substr(14,2);
    var sekunde = utcdate.substr(17,2);
    return Date.UTC(jahr, monat, tag, stunde, minute, sekunde);
} // utc2msec
function JB smoth(a,x,y,ys,range) {
    var l=a.length, l1=l-1;
    var t = new Array(1); for (var i=0; i<l; i++) \{ t[i] = \{\}; t[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i][ys]=a[i
              ]; for (var o in a[i]) t[i][o] = a[i][o]; }
     if (l < 2*range) return t;
    var rr, fak, faksum, sum, d;
    var r = Math. floor(range/2);
     for (var i=0; i<l; i++) {
           if(i < r) rr = i;
           else if (i > (l1-r)) rr=l1-i;
           else rr=r;
           if(rr>0) {
               sum = faksum = 0;
                d = Math.max(Math.abs(a[i][x]-a[i-rr][x]), Math.abs(a[i][x]-a[i+rr][x]))
                         |));
                if(d==0) d=1;
                for (var j=i-rr; j <=i+rr; j++) {
                     fak = (d-Math.abs(a[j][x]-a[i][x]))*rr/d;
                           fak = 1+rr-Math.abs(j-i);
                     sum += a[j][y] * fak;
                     faksum += fak;
                }
                t[i][ys] = sum/faksum;
          }
    }
    return t;
} // smoth
function JB diff(a,x,y,d,fak) {
    var l=a.length, l1=l-1;
     if (1 < 3) { for (var \ i=0; i<1; i++) a [i][d] = 0; return a; }
    var dx, dy;
    dx = a[1][x] - a[0][x] ;
    dy = a[1][y] - a[0][y];
    if (dx = 0) a [0] [d] = 0;
     else
                                a[0][d] = fak * dy/dx
     for (var i=1; i<l1; i++) {
          dx = a[i+1][x]-a[i-1][x] ;
```

```
dy = a[i+1][y]-a[i-1][y] ;
    if (dx = = 0) a[i][d] = a[i-1][d];
              a[i][d] = fak * dy/dx
    else
  }
  dx = a[11-1][x]-a[11][x] ;
  dy = a [ 11 - 1 ] [y] - a [ 11 ] [y] ;
  if (dx = = 0) a [11] [d] = a [11 - 1] [d];
  else
            a[l1][d] = fak * dy/dx
  return a;
} // diff
Array.prototype.sum=function() {
for (var s=0, i=0; i < this.length; i++)s += this [i];
return s;
} // sum
// gra
// Version vom 28.5.09
// Jï¿1/2rgen Berkemeier
// www.j-berkemeier.de
function gra(ID) {
var feld=document.getElementById(ID);
var dv=document.createElement("div");
 var cont=feld.appendChild(dv);
 var buf=document.createElement("div");
 this.w=parseInt(feld.offsetWidth-1);
 this.h=parseInt(feld.offsetHeight-1);
 var maxbuf=1;
 var bufsize = 1;
 var sp = document.createElement("div");
var col = "#000000";
sp.style.position="absolute";
sp.style.width="1px";
sp.style.height="1px";
sp.style.overflow="hidden";
sp.style.left="0px";
sp.style.top="0px";
sp.style.backgroundColor=col;
 this.setbuf=function(siz) {
  bufsize=maxbuf=Math.max(1,siz);
 }
 this.flush=function() {
  cont.appendChild(buf);
  buf=document.createElement("div");
  bufsize=maxbuf;
 }
```

```
this.punkt=function(x,y,c) {
  if (x<0 \mid \mid y<0 \mid \mid x>this.w \mid \mid y>this.h) return;
  if (c!=col) { col=c; sp. style.backgroundColor=col; }
  var pkt = sp.cloneNode(true);
  pkt.style.left=Math.round(x)+"px";
  pkt.style.top=Math.round(this.h-y)+"px";
  buf.appendChild(pkt);
  bufsize --; if (!bufsize) this.flush();
 } // punkt
 this.ver linie=function(x,y1,y2,c) {
  if (x < 0 | | x > this.w) return;
  if (y_1 < 0 \\ y_2 < 0) || (y_1 > this.h \\ y_2 > this.h) ) return;
  if (c!=col) { col=c; sp. style.backgroundColor=col; }
  y1=Math.max(0,Math.min(this.h,y1));
  y2=Math.max(0,Math.min(this.h,y2));
  var vl = sp.cloneNode(true);
  vl.style.left=Math.round(x)+"px";
  vl. style. top=Math. round (this.h-Math. \max(y1, y2))+"px";
  vl. style. height=Math.round (Math.abs(y2-y1)+1)+"px";
  buf.appendChild(vl);
  bufsize --; if (!bufsize) this.flush();
 } // ver linie
 this.hor_linie=function(x1,x2,y,c) {
  if (y < 0 || y > this.h) return;
  if ( (x1<0&&x2<0) || (x1>this.w&&x2>this.w) ) return;
  if (c!=col) { col=c; sp. style.backgroundColor=col; }
  x1=Math.max(0,Math.min(this.w,x1));
  x2=Math.max(0,Math.min(this.w,x2));
  var hl = sp.cloneNode(true);
  hl.style.left=Math.round (Math.min(x1, x2))+"px";
  hl.style.top=Math.round(this.h-y)+"px";
  hl. style. width=Math. round (Math. abs(x2-x1)+1)+"px";
  buf.appendChild(hl);
  bufsize --; if (!bufsize) this.flush();
 } // hor linie
 this.linie=function(xs, ys, xe, ye, c) {
// var flag=(maxbuf==1);
// if (flag) maxbuf=1000;
  xs=Math.round(xs); xe=Math.round(xe);
  ys=Math.round(ys); ye=Math.round(ye);
  var dx=xe-xs;
  var dy=ye-ys;
  if (dx==0 && dy==0) this.punkt(xs, ys, c)
  else if (dx==0) this.ver linie(xs, ys, ye, c);
  else if (dy==0) this.hor linie(xs, xe, ys, c);
  else {
```

```
var adx=Math.abs(dx);
       var ady=Math.abs(dy);
       var d=Math.min(adx,ady);
       dx = dx/d;
       dy=dy/d;
        if (adx=ady) {
          for (\text{var } x=xs, y=ys, i=0; i \le d; x=dx, y=dy, i++) this.punkt(x, y, c);
       }
        else if (adx<ady) {</pre>
          var dd=dy/Math.abs(dy);
           this.ver linie(xs, ys, ys+dy/2-dd, c);
           for (var x=xs+dx, y=ys+dy/2, i=1; i < d; x=dx, y=dy, i++) this.ver linie (x, y, y) = 0
                   y+dy-dd,c) ;
           this.ver linie (xe, ye-(dy+dd)/2, ye, c);
       }
        else {
          var dd=dx/Math.abs(dx);
           this.hor linie (xs, xs+dx/2-dd, ys, c);
            for (var x=xs+dx/2, y=ys+dy, i=1; i < d; x=dx, y=dy, i++) this.hor_linie(x, x+dy, i++) this.hor_lini
                   dx-dd, y, c;
           this.hor linie (xe-(dx+dd)/2, xe, ye, c);
       }
     }
// if (flag) {maxbuf=1; this.flush();}
  } // linie
   this.text=function(x,y,size,color,text,align) {
     var align h = "m";
     var align v = "m";
     if(align && align.length) {
       align h = align.substr(0,1);
       if (\text{align.length} > 1) align v = \text{align.substr}(1,1);
     }
     var pkt = document.createElement("div");
     pkt.style.position = "absolute";
     pkt.style.fontSize = size+"px";
     pkt.style.color = color;
     pkt.style.textAlign = "center";
     pkt.innerHTML = text;
     cont.appendChild(pkt);
     switch(align h) {
       case "l": default: pkt.style.left = Math.round(x) + "px"; break;
       case "m": pkt.style.left = Math.round(x) - pkt.offsetWidth/2 + "px";
                 break
       case "r": pkt.style.left = Math.round(x) - pkt.offsetWidth + "px";
                 break
```

```
switch(align v) {
   case "o": default: pkt.style.top = Math.round(this.h-y) + "px"; break;
   case "m": pkt.style.top = Math.round(this.h-y) - pkt.offsetHeight/2 + "
      px"; break;
   case "u": pkt.style.top = Math.round(this.h-y) - pkt.offsetHeight + "px
      "; break;
  }
 } // text
 this.del=function() {
  feld.removeChild(cont);
  delete cont;
  delete buf;
  var dv=document.createElement("div");
  cont=feld.appendChild(dv);
  buf=document.createElement("div");
  bufsize=maxbuf;
} // del
} // gra
// plot
// Version vom 29. 10. 2009
// Jï¿1/2rgen Berkemeier
// www.j-berkemeier.de
var plot = function(id, xstr, ystr) {
  var JB_log10 = function(x) \{ return Math.log(x)/Math.LN10; \}
  var JB toString = function(n) { return Number(n.toPrecision(15)).
     \operatorname{toString}(10); \}
  var JB makediv = function (parentnode, id, x0, y0, width, height) {
    var ele = document.createElement("div");
    ele.id = id;
    ele.style.position = "absolute";
    ele.style.left = x0 + "px";
    ele.style.top = y0 + "px";
    ele.style.width = width + "px";
    ele.style.height = height + "px";
    parentnode.appendChild(ele);
    return ele;
  }
  var plotid = id+"plot";
  var plotidm = id+"plotm";
  var idxlabel = id+"xlabel";
  var idylabel = id+"ylabel";
  var xobj = xstr?xstr:"x";
  var yobj = ystr?ystr:"y";
  JB GM Info(id, xobj+" "+yobj, false);
  var xmin=0, xmax=0, ymin=0, ymax=0;
```

```
var xfak=0, yfak=0;
var dx, dy, fx, fy;
var gr = null;
var xlabel = null;
var ylabel = null;
var feld = document.getElementById(id) ;
var w = parseInt(feld.offsetWidth-1);
var h = parseInt(feld.offsetHeight-1);
var ifeld = document.createElement("div");
var marker;
ifeld.style.position = "absolute";
ifeld.style.width = w + "px";
ifeld.style.height = h + "px";
feld.appendChild(ifeld);
this.scale = function(a) \{
  if (xmin=xmax) {
    \operatorname{xmax} = \operatorname{xmin} = \operatorname{a}[0][\operatorname{xobj}];
    ymax = ymin = a[0][yobj];
  }
  for (var i=0; i < a. length; i++)
    var t = a[i];
    if(t[xobj] < xmin) xmin = t[xobj];
    if(t[xobj]>xmax) xmax = t[xobj];
    if(t[yobj] < ymin) ymin = t[yobj];
    if(t[yobj]>ymax) ymax = t[yobj];
  }
  JB GM Info(id, xobj+": "+xmin+" ... "+xmax+"; "+yobj+": "+ymin+" ... "+
     ymax, false);
  if (xmax = xmin) \{ xmin = 0.5; xmax + 0.5; \}
  dx = xmax - xmin;
  fx = Math.pow(10, Math.floor(JB log10(dx))-1);
  xmin = Math. floor (xmin/fx) * fx;
  xmax = Math.ceil(xmax/fx)*fx;
  if (ymax=ymin) { ymin -= 0.5; ymax += 0.5; }
  dy = ymax - ymin;
  fy = Math.pow(10, Math.floor(JB log10(dy))-1);
  ymin = Math. floor (ymin/fy) * fy;
  ymax = Math.ceil(ymax/fy)*fy;
} // plot.scale
this.clear = function() {
  ifeld.innerHTML = "";
  xmax = xmin = ymax = ymin = xfak = yfak = 0;
} // plot.clear
this.frame = function(x0, y0, xl, yl) {
  ifeld.innerHTML = "";
```

```
JB makediv(ifeld, plotid, x0,0,w-x0,h-y0);
  if (xl.length) JB makediv(ifeld, idxlabel, x0, h-y0, w-x0, y0);
  JB makediv(ifeld, idylabel, 0, 0, x0, h-y0);
  JB makediv(ifeld, plotidm, x0,0,w-x0,h-y0);
  if(xl.length) xlabel=new gra(idxlabel);
  ylabel=new gra(idylabel);
  gr=new gra(plotid);
  gr.setbuf(1000);
  xfak = gr.w/(xmax-xmin);
  yfak = gr.h/(ymax-ymin);
  if (xl.length) xlabel.text(xlabel.w/2, 0, 16, "black", xl, "mu");
  ylabel.text(7, ylabel.h/2, 16, "black", yl, "mm");
  var tx = 100 * dx/gr.w;
  var ty = gr.h < 250 ? 50*dy/gr.h : 100*dy/gr.h;
  var tx10 = Math.pow(10, Math.floor(JB log10(tx)));
  tx = Math.round(tx/tx10);
  var ty10 = Math.pow(10, Math.floor(JB_log10(ty)));
  ty = Math.round(ty/ty10);
  tx = Number(String(tx).replace(/3/,"2").replace(/[4567]/,"5").replace
     (/[89]/,"10"));
  ty = Number(String(ty).replace(/3/,"2").replace(/[4567]/,"5").replace
     (/[89]/,"10"));
  tx = tx10;
  ty *= ty10;
  var mxmin = Math.ceil(xmin/tx) * tx;
  var mymin = Math.ceil(ymin/ty) * ty;
  for (var x=mxmin; x < =xmax; x + =tx) {
    gr.linie((x-xmin)*xfak,0,(x-xmin)*xfak,gr.h,"gray");
    if (xl.length) xlabel.text((x-xmin)*xfak, xlabel.h,14, "black",
       JB toString(x), "mo");
  }
  for (var y=mymin; y<=ymax; y+=ty) {</pre>
    gr.linie(0,(y-ymin)*yfak,gr.w,(y-ymin)*yfak,"gray");
    ylabel.text(ylabel.w,(y-ymin)*yfak,14,"black",JB toString(y),"rm");
  }
  var rahmen=new gra(plotid);
                                                    0, "black");
  rahmen.linie (
                       0.
                                 0, rahmen.w,
  rahmen.linie (rahmen.w,
                                 0, rahmen.w, rahmen.h, "black");
  rahmen.linie (rahmen.w, rahmen.h,
                                          0, rahmen.h, "black");
  rahmen.linie (
                       0, rahmen.h,
                                                    0, "black");
                                         0,
} // plot.frame
this.plot = function(a, col) {
  for (var \ i=0; i<a.length-1; i++)
    gr.linie(
     (a[i] xobj] - xmin) * xfak,
     (a[i][yobj]-ymin)*yfak,
```

```
(a[i+1][xobj]-xmin)*xfak,
     (a[i+1][yobj]-ymin)*yfak,
     col);
  gr.flush();
} // plot.plot)
this.showmarker = function(markertype) {
 var pele = document.getElementById(plotid);
 marker = document.createElement("div");
 marker.style.position = "absolute";
  marker.style.display = "none";
  if (markertype="Punkt") {
    marker.style.fontSize = "32px";
   var txt=document.createTextNode(String.fromCharCode(8226)) ; //
       Kreis als Zeichen: • oder • evtl auch 8729
   marker.appendChild(txt);
 }
  else {
   marker.style.top = "0";
   marker.style.height = gr.h + "px";
   marker.style.width = "1px";
   marker.style.backgroundColor = "black";
 }
 pele.appendChild(marker);
} // plot.showmarker
this.hidemarker = function() {
 marker.style.display = "none";
} // plot.hidemarker
this.setmarker = function(a, markertype) {
 marker.style.display = "";
  if (markertype="Punkt") {
    marker.style.left = Math.round((a[xobj]-xmin)*xfak) - marker.
       offsetWidth/2 + "px";
   marker.style.top = Math.round(gr.h - (a[yobj]-ymin)*yfak) - marker.
       offsetHeight/2 + "px";
 }
  else {
   marker.style.left = Math.round((a[xobj]-xmin)*xfak) + "px";
 }
} // plot.setmarker
this.markeron = function(a, callback over, callback out, callback move,
   markertype) {
 var dieses = this;
 var posx=0;
 var mele = document.getElementById(plotidm);
 mele.onmouseover = function(e) {
    if (!e) e = window.event;
```

```
e.cancelBubble = true;
    if (e.stopPropagation) e.stopPropagation();
    mele.onmousemove = function(e) {
      if (!e) e = window.event;
      e.cancelBubble = true;
      if (e.stopPropagation) e.stopPropagation();
      posx = e.layerX ? e.layerX : e.offsetX;
      var x = posx/xfak+xmin;
      var al = a.length;
      var p, pi;
      if (x \le a[0] [xobj]) pi=0;
      else if (x \ge a[al-1][xobj]) pi=al-1;
      else {
        p = al/2;
        pi = Math. floor(p);
        var dp = Math.ceil(p/2);
        do {
          var apx = a[pi][xobj];
          if(x < apx) \{ p -= dp; if(p < 0) p = 0; \}
          else if (x>apx) \{ p \models dp; if (p>al-1) p=al-1; \}
          else break;
          pi = Math. floor(p);
          dp = dp/2;
         while(dp>=0.5) ; 
      }
                     dieses.setmarker(a[pi],markertype);
      if (callback move && typeof (callback move) == "function")
          callback move(pi, a[pi]);
      return false;
    }
    if (callback over && typeof (callback over) == "function") callback over
        ();
    return false;
  }
  mele.onmouseout = function(e) {
    if (!e) e = window.event;
      dieses.hidemarker();
    mele.onmousemove = null;
    if (callback out && typeof (callback out) == "function") callback out();
    return false;
  }
} // plot.markeron
this.markeroff = function() {
  var ele = document.getElementById(plotid);
  ele.onmousemove = null;
  ele.onmouseout = null;
```

```
} // plot.markeroff
} // plot
// PolylineEncoder.js copyright Mark McClure April/May 2007
// V 2.1
         July 2007
// http://facstaff.unca.edu/mcmcclur/GoogleMaps/EncodePolyline/
   PolylineEncoderClass.html
// Geringfï¿ <sup>1</sup>/<sub>2</sub>gig angepasst von JB
var PolylineEncoder = function() {
  this.numLevels = 18;
  this.zoomFactor = 2;
  this.verySmall = 0.00001;
  this.forceEndpoints = true;
  this.zoomLevelBreaks = new Array(this.numLevels);
  for (var i = 0; i < this.numLevels; i++) {
    this.zoomLevelBreaks[i] = this.verySmall*Math.pow(this.zoomFactor,
        this.numLevels-i-1);
  }
}
PolylineEncoder.prototype.dpEncode = function(points) {
  var absMaxDist = 0;
  var stack = [];
  var dists = new Array(points.length);
  var maxDist, maxLoc, temp, first, last, current;
  var i, encodedPoints, encodedLevels;
  var segmentLength;
  if (points.length > 2) {
    stack.push([0, points.length -1]);
    while (stack.length > 0) {
      current = stack.pop();
      maxDist = 0;
      segmentLength = Math.pow(points[current[1]].lat()-points[current
          [0].lat(),2) +
        Math.pow(points [current [1]].lng()-points [current [0]].lng(),2);
      for (i = current[0]+1; i < current[1]; i++) {
        temp = this.distance(points[i],
          points [current [0]], points [current [1]],
          segmentLength);
        if (temp > maxDist) {
          maxDist = temp;
          \max Loc = i;
          if (maxDist > absMaxDist) {
            absMaxDist = maxDist;
          }
        }
```

```
if (maxDist > this.verySmall) {
        dists[maxLoc] = maxDist;
        stack.push([current[0], maxLoc]);
        stack.push([maxLoc, current[1]]);
      }
    }
  }
  encodedPoints = this.createEncodings(points, dists);
  encodedLevels = this.encodeLevels(points, dists, absMaxDist);
  return {
    encodedPoints: encodedPoints,
    encodedLevels: encodedLevels,
    encodedPointsLiteral: encodedPoints.replace(/\backslash/g,"/\rangle/")
 }
}
PolylineEncoder.prototype.dpEncodeToJSON = function(points,
  color , weight , opacity ) {
  var result;
  if (!opacity) {
    opacity = 0.9;
  }
  if (!weight) {
    weight = 3;
  }
  if(!color) {
    color = "#0000 ff";
  }
  result = this.dpEncode(points);
  return {
    color: color,
    weight: weight,
    opacity: opacity,
    points: result.encodedPoints,
    levels: result.encodedLevels,
    numLevels: this.numLevels,
    zoomFactor: this.zoomFactor
  }
}
PolylineEncoder.prototype.dpEncodeToGPolyline = function (points,
  color, weight, opacity) {
  if(!opacity) {
    opacity = 0.9;
  }
  if (!weight) {
    weight = 3;
  }
```

```
if(!color) {
    color = "#0000 ff";
 }
 return new GPolyline.fromEncoded(
    this.dpEncodeToJSON(points, color, weight, opacity));
}
PolylineEncoder.prototype.distance = function (p0, p1, p2, segLength) {
 var u, out;
  if(p1.lat() = p2.lat() \& p1.lng() = p2.lng()) 
    out = Math. sqrt (Math. pow(p2. lat ()-p0. lat (), 2) + Math. pow(p2. lng ()-p0.
       lng(), 2));
 }
  else {
    u = ((p0.lat()-p1.lat())*(p2.lat()-p1.lat())+(p0.lng()-p1.lng())*(p2.lat()-p1.lng()))
       \ln g() - p1 \cdot \ln g()) / 
      segLength;
    if(u \le 0) {
      out = Math. sqrt(Math. pow(p0. lat() - p1. lat(), 2) + Math. pow(p0. lng())
         - p1.lng(),2));
    }
    if (u >= 1) {
      out = Math.sqrt(Math.pow(p0.lat() - p2.lat(), 2) + Math.pow(p0.lng())
         - p2.lng(),2));
    }
    if(0 < u \& u < 1) 
      out = Math. sqrt(Math. pow(p0.lat()-p1.lat()-u*(p2.lat()-p1.lat()), 2))
         +
        Math.pow(p0.lng()-p1.lng()-u*(p2.lng()-p1.lng()),2));
   }
  }
 return out;
}
PolylineEncoder.prototype.createEncodings = function(points, dists) {
 var i, dlat, dlng;
 var plat = 0;
 var plng = 0;
 var encoded points = "";
  for (i = 0; i < points.length; i++)
    if (dists[i] = undefined || i = 0 || i = points.length -1) 
      var point = points [i];
      var lat = point.lat();
      var lng = point.lng();
      var late 5 = Math. floor(lat * 1e5);
      var lnge5 = Math. floor (lng * 1e5);
      dlat = late5 - plat;
```
```
dlng = lnge5 - plng;
      plat = late5;
      plng = lnge5;
      encoded points += this.encodeSignedNumber(dlat) +
        this.encodeSignedNumber(dlng);
   }
  }
  return encoded points;
}
PolylineEncoder.prototype.computeLevel = function(dd) {
 var lev=0;
  if (dd > this.verySmall) {
    while(dd < this.zoomLevelBreaks[lev]) {</pre>
      lev++;
    }
  }
 return lev;
}
PolylineEncoder.prototype.encodeLevels = function (points, dists,
   absMaxDist) {
 var i;
 var encoded_levels = "";
  if(this.forceEndpoints) {
    encoded levels += this.encodeNumber(this.numLevels-1)
 } else {
    encoded levels += this.encodeNumber(
      this.numLevels-this.computeLevel(absMaxDist)-1)
  }
  for (i=1; i < points.length -1; i++)
    if(dists[i] != undefined) {
      encoded levels += this.encodeNumber(
        this.numLevels-this.computeLevel(dists[i])-1);
   }
  }
  if(this.forceEndpoints) {
    encoded_levels += this.encodeNumber(this.numLevels-1)
 } else {
    encoded levels += this.encodeNumber(
      this.numLevels-this.computeLevel(absMaxDist)-1)
 }
 return encoded levels;
}
PolylineEncoder.prototype.encodeNumber = function (num) {
 var encodeString = "";
 var nextValue, finalValue;
  while (num \ge 0x20) {
```

```
nextValue = (0x20 | (num \& 0x1f)) + 63;
    encodeString += (String.fromCharCode(nextValue));
    num >>= 5;
  }
  finalValue = num + 63;
  encodeString += (String.fromCharCode(finalValue));
  return encodeString;
}
PolylineEncoder.prototype.encodeSignedNumber = function (num) {
  var sgn num = num \ll 1;
  if (num < 0) {
    sgn_num = (sgn_num);
  }
  return(this.encodeNumber(sgn num));
}
PolylineEncoder.latLng = function(y, x) {
        this.y = y;
        this.x = x;
}
PolylineEncoder.latLng.prototype.lat = function() {
        return this.y;
}
PolylineEncoder.latLng.prototype.lng = function() {
        return this.x;
}
PolylineEncoder.pointsToLatLngs = function(points) {
        var i, latLngs;
        latLngs = new Array(0);
        for(i=0; i<points.length; i++) {</pre>
                latLngs.push(new PolylineEncoder.latLng(points[i][0],
                    points [i][1]));
        }
        return latLngs;
}
PolylineEncoder.pointsToGLatLngs = function(points) {
        var i, gLatLngs;
        gLatLngs = new Array(0);
        for(i=0; i<points.length; i++) {</pre>
                gLatLngs.push(new GLatLng(points[i][0], points[i][1]));
        }
        return gLatLngs;
} // PolylineEncoder
addEvent(window, "load", function() {
  if (document.getElementsByTagName && GBrowserIsCompatible()) {
    var Map Nr=0;
```

```
var chkTyp = function(typString) {
  if (typString="Karte") return G NORMAL MAP ;
  if(typString="Satellit") return G_SATELLITE_MAP ;
  if (typString="Hybrid") return G HYBRID MAP ;
  if (typString="Oberflaeche") return G PHYSICAL MAP ;
  if(typString="OSM" || typString="OSM_Mapnik") return "
     osm mapnik map";
  if (typString="OSM TaH") return "osm tah map";
  if(typString="OSM_Cycle") return "osm_cycle_map";
  return G SATELLITE MAP;
 // chkTyp 
var divs = document.getElementsByTagName("div");
var typ = G SATELLITE MAP;
for (var i=0; i < divs. length; i++) {
  var div = divs[i];
  if (div.className) {
    var Klasse = div.className;
    var CN = Klasse.toLowerCase().indexOf("gpxview");
    if (CN>-1) {
      if(div.id) var Id = div.id;
      else {
        var Id = \operatorname{"map"} + (\operatorname{Map}_Nr++);
        \operatorname{div.id} = \operatorname{Id};
      }
      var GPX = Klasse.substring(CN).split()[0];
      GPX = GPX. split(":") ;
      if (GPX. length == 3) {
         typ = chkTyp(GPX[2]);
      }
      if (GPX[1].length) {
        window ["Karte "+Id] = new makeMap(Id);
        window ["Karte_"+Id]. Spur (GPX[1], typ);
      }
    }
  }
}
var buttons = document.getElementsByTagName("button");
for (var i=0;i<buttons.length;i++) {</pre>
  var button = buttons[i];
  if(button.className) {
    var Klasse = button.className;
    var CN = Klasse.toLowerCase().indexOf("gpxview");
    if (CN > -1) {
      var cmd = Klasse.substring(CN).split()[0];
      cmd = cmd. split(":");
      if (\text{cmd.length} > 2) {
```

```
var Id = \operatorname{cmd}[1];
             switch(cmd[2]) {
               case "skaliere":
                  (function() {
                    var mapid = "Karte "+Id;
                    button.onclick = function(){window[mapid].rescale()};
                  })();
                  break;
               case "lade":
                  if (\text{cmd.length} > 3) {
                    if (\text{cmd.length} > 4) typ = \text{chkTyp}(\text{cmd}[4]);
                    else typ = G SATELLITE MAP;
                    (function() {
                      var fn = cmd [3];
                      var mapid = "Karte "+Id;
                      var tp = typ;
                      button.onclick = function() {window [mapid].Spur(fn,tp)
                          };
                    })();
                  }
                  break;
                default:
                  break;
             }
          }
        }
      }
    }
    addEvent(window, "unload", GUnload);
  }
  else JB GM Info("", "Ihr Browser unterstï; ½tzt nicht die benï; ½tigten
     Methoden!",true);
});
```



## 10.8 googleMapFiles/getKMLfile

```
<?php
include ("googleMapFiles/APIkey/googleAPIkey.php");
echo("<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">");
?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
com:vml">
```

```
<head>
    <title>Google Maps-API: JavaScript-Beispiel</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"</pre>
       >
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=</pre>
       <?php echo $google api key; ?>" type="text/javascript">
    </script>
    <style type="text/css">
        @import url("http://www.google.com/uds/css/gsearch.css");
        @import url("http://www.google.com/uds/solutions/localsearch/
            gmlocalsearch.css");
        /* needed for the localsearch */
    </style>
    <script src="http://www.google.com/uds/api?file=uds.js&amp;v=1.0"</pre>
       type="text/javascript">
        // needed for the localsearch
    </script>
    <script src="http://www.google.com/uds/solutions/localsearch/</pre>
       gmlocalsearch.js" type="text/javascript">
        // needed for the localsearch
    </script>
    <script type="text/javascript">
        var map;
        var geoXml;
        var toggleState = 1;
        function init(){
            if (GBrowserIsCompatible()) {
                // GGeoXML needs an url tag starting with http://
                    11111
                geoXml = new GGeoXml("http://itsgeoweb.meyr.info/
                    googleMapFiles/getKMLfile/kml/Gaisbergrunde.kml");
                //geoXml = new GGeoXml("http://localhost/edu/
                    googleMapFiles/getKMLfile/kml/Gaisbergrunde.kml");
                map = new GMap2(document.getElementById("map canvas"))
                map.setCenter(new GLatLng(47.8, 13.2), 11);
                map.setUIToDefault();
                map.addOverlay(geoXml);
            }
        }
        function toggleMyKml() {
            if (toggleState = 1) {
```

```
map.removeOverlay(geoXml);
                     toggleState = 0;
                 }
                 else {
                     map.addOverlay(geoXml);
                     toggleState = 1;
                 }
             }
            // to avoid writting the onload() function in the <body> tag
             // it's better to start the functions within this script...
            window.onload = init;
            // unloading map and free memory
            window.onunload = GUnload;
        </\operatorname{script}>
    </head>
    < body >
        <div id="map_canvas" style="width: 100%; height: 600px">
        </div>
    </body>
</html>
```

Listing 112: getKMLfile index.php

## 10.9 $modules/mod_edu$

```
<\!\!html\!\!>\!\!\!dody \ bgcolor="\#\!FFFFF"><\!\!/body><\!\!/html>
```

```
Listing 113: mod_edu index.html
```

```
<?php
/**
 * mod_edu.php
 */
// no direct access
defined('_JEXEC') or die('Restricted access');
// Include the functions only once
require_once (dirname(__FILE__).DS. 'helper.php');
// get the current topic
$__facilities = $_GET["facilities"];
// get the current uri</pre>
```

```
 uri =  SERVER ["REQUEST URI"];
// removing the topic if already set
$remove_facilities = explode("&facilities=".$__facilities, $__uri);
 uri =  remove facilities [0]; 
test = \__edu::getCategories();
echo("<div class=\"mod edu\">");
foreach ($test as $row) {
    echo("<a href=\"".$ uri."&facilities=".$row->id cat."\">".$row->id cat."\">".
       iconname. "</a><br/>br />");
}
echo("<hr>");
// display all facilities
echo("<span class = ||edu||><a href = ||".
   anzeigen </a></span>");
echo("</div>");
?>
```



```
<?php
/**
* mod_edu.php (helper)
*/
// no direct access
defined(' JEXEC') or die('Restricted access');
// get access to database in order to list categories
class __edu {
    // get categories
    public function getCategories() {
        db = \&JFactory :: getDBO();
        // categories in database are unique - but just to get sure -
            distinct query
        $query = "SELECT id cat, iconname FROM db category ORDER BY
            iconname ASC";
        $db->setQuery($query);
        if (!$db->query()) {
            echo "<script type='text/javascript'>
             alert('".$db->getErrorMsg()."');
             </\operatorname{script}>";
            return false;
```

```
} else {
    $rows = $db->loadObjectList();
    return $rows;
    }
}
```

Listing 115:  $mod\_edu$  Modul Klassenfunktionen

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.0.0">
    <name>
        EDU Google Maps
    </name>
    < author >
        Hoepfl Klaus
    </author>
    <creationDate>
        April 2010
    </creationDate>
    <copyright>
        Copyright (C) Klaus Hoepfl. All rights reserved.
    </copyright>
    <license>
        http://www.gnu.org/licenses/gpl-2.0.html GNU/GPL
    </license>
    < authorEmail>
    </authorEmail>
    <authorUrl>
        www.fh-salzburg.ac.at
    </authorUrl>
    <version>
        1.0.0
    < version >
    < description >
        Selectable categories within the edu project
    </description>
    <files>
        <filename module="mod edu">
            mod edu.php
        </filename>
        <filename>
            index.html
        </filename>
    </files>
    < params >
```

</params> </install>

Listing 116: mod\_edu Modulbeschreibung

## 10.10 modules/mod\_edu\_cat

```
<html><body bgcolor="#FFFFF"></body></html>
```

```
Listing 117: mod_edu_cat index.html
```

```
<?php
/**
* mod_edu_cat.php
*/
// no direct access
defined('_JEXEC') or die('Restricted access');
function umlaut($string) {
   \theta = array("ae", "oe", "ue", "Ae", "Oe", "Ue", "ss", ");
   how = array("ae", "oe", "ue", "Ae", "Oe", "Ue", "ss", "-");
   return (str_replace($what, $how, $string));
}
// get the current uri
// Include the functions only once
// get access to database in order to list categories
require_once (dirname(__FILE__).DS. 'helper.php');
// script to confirm deletion of category
?>
<script language="JavaScript">
   function delConfirm(nr, name){
       if (confirm("Soll " + name + " wirklich gelÃ{scht werden?"})) {
           document.location = "<?php echo $__uri;?>&deletecat=" + nr;
       }
   }
</script>
```

```
<?php
// begin module mod edu cat
echo ("<div class = \"mod edu cat \">");
// delete confirmed category
if (! empty($ GET["deletecat"])) {
   $temp = __edu_cat::getIconname($_GET["deletecat"]);
   __edu_cat::delCategory($_GET["deletecat"]);
   echo ("<span class=\"edu cat ok\">Die Kategorie <b>\"". tep. "\"<br/>/"</br/>
      wurde gel\tilde{A} (scht.</span>);
}
// remove "&deletecat=" from uri
$data = explode(("&deletecat=".$ GET["deletecat"]), $ uri);
 uri = data[0]; 
// list all existing categories
echo("Nachfolgende Kategorien existieren:");
echo("<form id=\"edu cat del\" name=\"edu cat\" method=\"get\" action=\"".
   $__uri."\">");
// display categories within a table
echo ("<table class=|"edu cat|">");
echo("");
//echo("<b>ID</b>");
echo("Kategorie</b>");
echo("<b>Icon</b>");
echo(");
// get all categorries
test = edu_cat::getCategories();
foreach ($test as $row) {
   // providing links for each category of educational facilities
   echo("");
   //echo("".$row->id cat."");
   echo ("<td class = \"iconname \">". $row -> iconname . "</td>");
   echo("<img src=\"googleMapFiles/getMarkers/
      icons/".strtolower(umlaut($row->iconname)).".png\">");
   // delete
   echo("><a href=\"javascript:delConfirm('".$row->id cat."', '".$row
      ->iconname . " ');\">Löschen</a>
}
echo("");
echo("</form>");
// adding a new category
echo("<hr>");
?>
<form id="edu cat" name="edu cat" method="post" action="<?php echo $ uri;
   ?>">
   Neue Kategorie anlegen:
   <br/>
```

```
<input class="edu cat" type="text" name="category" size="40" maxlength</pre>
       ="40" value="" />
    <br/>
    <input class="edu cat submit" name="submit" type="submit" value="
       Kategorie anlegen" />
</form>
<?php
// write new category to database
if (isset($_POST["submit"])) {
    if (($ POST["category"] != "")) {
        // check if category is already in database
        if ((__edu_cat::checkCategories($_POST["category"])) == "") {
            // category does not exist - add new category
            edu cat::setCategory($ POST["category"]);
            echo("<span class=\"edu cat ok\"><br />Die Kategorie <b>\"".
                $ POST["category"]."\"</b> wurde hinzugef\tilde{A}_{\frac{1}{4}}gt.<br/>br /></span
               >");
        } else {
            // category exists
            echo("<span class=\"edu cat error\"><br />Die Kategorie <b>\""
                .$ POST["category"]."\"</b>
                /></span>");
        }
    } else {
        // button pressed without input
        echo ("<span class=\"edu cat error\">Sie haben keine Kategorie
            eingegeben!</span>");
    }
}
echo("<hr>");
// form to upload an icon to an existing category
?>
<form id="edu cat img" name="edu_cat_img" method="post" action="<?php echo</pre>
    $ uri; ?>" enctype="multipart/form-data">
    Kategorieicon auswählen
    <br/>
    (PNG Format 32px * 34px):
    <br/>
    <input class="edu cat" type="file " name="imageFromClient" size="22" />
    <br/>
    <br/>
    und zu dieser Kategorie hinzuf\tilde{A}_{4}^{1}gen:
    <br/>
    <\!\!{\tt select name} = \!\!"edu\_cat\_icon" class = \!"edu\_cat\_icon"\!>
        <option value="">Bitte auswÃ\cuthen</option>
        <?php
```

```
// dropdown list
        foreach ($test as $row) {
             // providing links for each category of educational facilities
            echo("<option value=\"".$row->id cat."\">".$row->iconname."</
                option>");
        }
        ?>
    </select>
    <br/>
    <br/>
    <input class="edu cat submit" name="submit2" type="submit" value="
        Kategorieicon hochladen" />
</form>
<?php
// if a new icon is added to an existing category
if (isset($ POST["submit2"])) {
    if ((! empty($_POST["edu_cat_icon"])) && (! empty($_FILES["
       imageFromClient"]["tmp name"])) && ($ FILES["imageFromClient"]["
        type"] = "image/png")) {
        // get iconname of edu cat icon
        $iconname = strtolower(umlaut(__edu_cat::getIconname($_POST["
            edu cat icon"]).".png"));
        // move
        move uploaded file(($ FILES['imageFromClient']['tmp name']), "
            googleMapFiles/getMarkers/icons/".$iconname);
        echo("<span class=\"edu cat ok\"><br/>br />Das neue Icon wurde zur
            Kategorie <b>/"".( edu cat::getIconname($ POST["edu cat icon"
            ]))."\"</b> hinzugef\tilde{A}_4^1gt.<br /></span>");
        // get filename of imageFromClient
    } else {
        // no iconname and no file
        echo("<span class=\"edu cat error\"><br />Bitte wAghlen Sie ein
            Icon f\tilde{A}^{\frac{1}{4}}r den Marker im PNG Bildformat und eine zugeh\tilde{A}¶rige
            Kategorie.<br /></span>");
    }
    if (( empty($ POST["edu cat icon"])) && (! empty($ FILES["
       imageFromClient" | ["tmp name" ] ) )) {
        echo("<span class=\"edu cat error\"><br />Bitte wAghlen Sie ein
            Icon f\tilde{A}_{4}^{1}r den Marker im PNG Bildformat.<br/>br /></span>");
    }
    if ((! empty($ POST["edu cat icon"])) && ( empty($ FILES["
       imageFromClient" | [ "tmp name" | ) ) ) {
        echo("<span class = \"edu cat error \"><br/>br />Bitte wA\"alpha hlen Sie eine
            zugeh \tilde{A}  [rige Kategorie. < br /> </ span>");
    }
```

echo("</div>"); ?>

Listing 118: mod\_edu\_cat Modul Auswahl Bildungsstätten

```
<?php
/**
* mod_edu_cat.php (helper)
*/
// no direct access
defined(' JEXEC') or die('Restricted access');
// get access to database in order to list categories
class edu cat {
    // get categories
    public function getCategories() {
        db = \&JFactory :: getDBO();
        // categories in database are unique - but just to get sure -
           distinct query
        $query = "SELECT id cat, iconname FROM db category ORDER BY
           iconname ASC";
        $db->setQuery($query);
        if (!$db->query()) {
            echo "<script type='text/javascript'>
             alert('".$db->getErrorMsg()."');
             </\operatorname{script}>";
            return false;
        else 
            $rows = $db->loadObjectList();
            return $rows;
        }
    }
    // set category
    public function setCategory($string) {
        db = \&JFactory :: getDBO();
        // categories in database are unique - but just to get sure -
            distinct query
        $query = "INSERT INTO db_category (iconname) VALUES ('".$string."
            ')";
        $db->setQuery($query);
        if (!$db->query()) {
            echo "<script type='text/javascript'>
             alert('".$db->getErrorMsg()."');
```

```
</\operatorname{script}>";
        return false;
    else 
        return true;
    }
}
// delete category
public function delCategory($int) {
    db = \&JFactory :: getDBO();
    // categories in database are unique – but just to get sure –
        distinct query
    $query = "DELETE FROM db category WHERE id cat = '".$int."';
    $db->setQuery($query);
    if (! $db->query()) {
        echo "<script type='text/javascript'>
         alert('".$db->getErrorMsg()."');
         </\operatorname{script}>";
        return false;
    } else {
        return true;
    }
}
// check categories if a new entered category exists
public function checkCategories($string) {
    db = dJFactory :: getDBO();
    // categories in database are unique - but just to get sure -
        distinct query
    $query = "SELECT id cat FROM db category WHERE iconname LIKE '".
        $string."'";
    $db->setQuery($query);
    if (!$db->query()) {
        echo "<script type='text/javascript'>
         alert('".$db->getErrorMsg()."');
         </\operatorname{script}>";
        return false;
    } else {
        // category exitst
        $result = $db->loadResult();
        return $result;
    }
}
// get iconname from id cat
public function getIconname($int) {
```

```
db = \&JFactory :: getDBO();
        // categories in database are unique - but just to get sure -
            distinct query
        $query = "SELECT iconname FROM db category WHERE id cat = '".$int.
            ",";
        $db->setQuery($query);
        if (! %db->query()) {
            echo "<script type='text/javascript'>
             alert('".$db->getErrorMsg()."');
             </\operatorname{script}>";
            return false;
        } else {
            // category exitst
             $result = $db->loadResult();
            return $result;
        }
   }
}
```

Listing 119:  $mod\_edu\_cat$  Modul Klassenfunktionen

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.0.0">
    <name>
        EDU Google Maps Category
    </name>
    <\!\!\mathrm{author}\!>
        Hoepfl Klaus
    </author>
    < creationDate >
        April 2010
    </creationDate>
    <copyright>
        Copyright (C) Klaus Hoepfl. All rights reserved.
    </copyright>
    <license>
        http://www.gnu.org/licenses/gpl-2.0.html GNU/GPL
    </license>
    <authorEmail>
    </authorEmail>
    <authorUrl>
        www.fh-salzburg.ac.at
    </authorUrl>
    <version>
        1.0.0
    </version>
```

```
<description>
Adding new category to edu project
</description>
<files>
<filename module="mod_edu_cat">
mod_edu_cat.php
</filename>
<filename>
index.html
</filename>
</filename>
</files>
<params>
</params>
</install>
```

Listing 120:  $mod\_edu\_cat$  Modulbeschreibung