

Konzeption und Realisierung eines Bookmarkverwaltungssystems für eine Stadtbücherei auf der Basis von PHP und MySQL

Diplomarbeit

im Fach

Internet und Intranet

Studiengang Öffentliche Bibliotheken

der

Fachhochschule Stuttgart –

Hochschule der Medien

Klaus Köhler

Erstprüfer: Prof. Dr. Wolf-Fritz Riekert

Zweitprüfer: Johannes Auer

Bearbeitungszeitraum: 15. Juli 2002 bis 15. Oktober 2002

Stuttgart, Oktober 2002

Kurzfassung

Die hier vorliegende Arbeit umfasst die Konzeption und Realisierung eines web-basierten Bookmarkverwaltungssystems für die Stadtbücherei in Stuttgart. In diesem speziell für die Stadtbücherei Stuttgart erstellten System können die Lektoren selbstständig ihre fachbezogenen Links verwalten. Sie benötigen dafür keine speziellen Kenntnisse in HTML. Nur der sichere Umgang mit einem Internetbrowser ist erforderlich.

Die Lektoren loggen sich im Intranet der Stadtbücherei passwortgeschützt in das Verwaltungssystem ein. Sie haben dann Zugriff auf die ihnen vom Administrator freigegebenen Lektoratsgebiete. In einem extra Administrationsbereich befindet sich die Benutzerverwaltung. Ebenso gibt es dort die Möglichkeit, neue Bereiche einzurichten, und die Möglichkeit, Benutzerrechte für die jeweiligen Lektoratsgebiete zuzuweisen.

Realisiert wurde das Projekt auf einem LAMP-System (Linux - Apache - MySQL - PHP). Auf das plattformunabhängige System kann mittels Webbrowser zugegriffen werden.

Schlagwörter

Bookmark, Datenbankentwicklung, Intranet, LAMP, Verwaltungssystem

Abstract

This thesis covers the conception and realization of a webbased bookmark-administration-system. It was developed for the public library in Stuttgart.

This system specially developed for the city library enables the lectors to manage the links on their own. In addition it is not necessary that the lectors learn how to do HTML-coding. The lectors log into the system of the libraries intranet with a password. They have access to the topics which have been released by the administrator. Furthermore, there is a special administrative area. This is, where the administration of the users take place as well as the creation of new topics. In addition to that, the administrator can grant the rights for certain topics to the lectors.

The project was realised on a LAMP system (Linux - Apache - MySQL - PHP). The platform-independant system can be accessed from within a webbrowser.

Key Words

administration system, bookmark, database development, intranet, LAMP

Vorwort und Danksagung

Angeregt zu dieser Arbeit wurde ich durch zwei Umstände: Zum einen habe ich im Wahlpflichtfach „Internet und Intranet“ einen Kurs zum Thema Webdatenbanken besucht. Dort wurden verschiedene Möglichkeiten gezeigt, wie Datenbanken mittels Skriptsprache in Webseiten eingebunden werden können und was man damit alles machen kann. Dieses Thema hat mich so fasziniert, dass ich in diesem Bereich meine Diplomarbeit schreiben wollte.

Zum anderen war ich während meines Praxissemesters in der Stadtbücherei Stuttgart für das Projekt „Pflege der Webseite des Ateliers Moderne Technik“ eingeteilt. Auf diese Weise bin ich mit der EDV-Abteilung in Kontakt gekommen und habe erfahren, dass sie auf der Suche nach einer Datenbanklösung für die Linksammlungen der Ateliers ist. So entstand die Idee zu dieser Diplomarbeit.

Mein Dank gilt all denen, die mir in der Zeit, in der diese Arbeit entstanden ist, mit Rat und Tat zur Seite gestanden haben.

Mein besonderer Dank gilt Herrn Prof. Dr. Riekert und Herrn Auer für die Annahme und Betreuung der Diplomarbeit — mit genau dem richtigen Maß an Freiheit, Anregungen und Kritik, das ein gutes Arbeiten ermöglicht hat.

Stuttgart, im Oktober 2002

Klaus Köhler

Inhaltsverzeichnis

| | |
|---|-----------|
| Kurzfassung | 2 |
| Abstract | 3 |
| Vorwort und Danksagung | 4 |
| Inhaltsverzeichnis | 5 |
| Abbildungsverzeichnis | 8 |
| Tabellenverzeichnis | 9 |
| Abkürzungsverzeichnis | 10 |
| 1 Überblick | 11 |
| 1.1 Ausgangspunkt und Zielsetzung | 12 |
| 1.2 Inhalt dieser Arbeit | 12 |
| 2 Stand der Technik | 13 |
| 2.1 Beispielprojekte | 14 |
| 2.1.1 SuUB Bremen | 14 |
| 2.1.2 CoOL | 14 |
| 2.1.3 Zusammenschlüsse | 14 |
| 2.2 Ein BVS für die Stadtbücherei Stuttgart | 15 |
| 3 Aufgabenstellung | 17 |
| 3.1 Allgemeine Anforderungen an ein Bookmarkverwaltungssystem | 17 |
| 3.1.1 Was für Soft- und Hardware benötigt man für ein BVS? | 18 |
| 3.1.2 Abgrenzung | 18 |
| 3.2 Vorgaben der Stadtbücherei | 19 |
| 3.2.1 Spezielle Anforderungen an die Funktionalität von Seiten der Stadtbücherei | 19 |
| 3.2.2 Technische Voraussetzungen | 20 |

| | | |
|----------|---|-----------|
| 4 | Konzeption | 21 |
| 4.1 | Die Benutzergruppen des BVS | 21 |
| 4.1.1 | Lektoren | 21 |
| 4.1.2 | Administratoren | 22 |
| 4.1.3 | Homepagebesucher | 22 |
| 4.2 | Sicherheitsaspekte | 22 |
| 4.2.1 | Intranet | 22 |
| 4.2.2 | Persönliche Anmeldung | 23 |
| 4.2.3 | Homepagebesucher | 23 |
| 4.3 | Software | 23 |
| 4.3.1 | Die Klientseite | 23 |
| 4.3.2 | Die Serverseite | 24 |
| 4.3.3 | Benutzungsoberfläche | 24 |
| 4.4 | Fensteraufteilung | 24 |
| 4.4.1 | Konzeption der Seitenstruktur | 25 |
| 4.5 | Konzeption der Datenbankstruktur | 26 |
| 5 | System aus Anwendersicht | 27 |
| 5.1 | Sicht des Lektors | 27 |
| 5.1.1 | Anmeldung | 28 |
| 5.1.2 | Seitenübersicht | 28 |
| 5.1.3 | Bearbeitungsmodus | 30 |
| 5.2 | Sicht des Administrators | 32 |
| 5.2.1 | Last-Modified Liste | 32 |
| 5.2.2 | Benutzerverwaltung | 33 |
| 5.2.3 | Rechtevergabe | 34 |
| 5.2.4 | Verwaltung der Lektoratsgebiete | 35 |
| 5.2.5 | Verwaltung der Seiten | 35 |
| 5.3 | Sicht des Benutzers (Homepagebesuchers) | 37 |
| 6 | Realisierung | 39 |
| 6.1 | Datenbankstruktur | 39 |
| 6.2 | Programmierung | 42 |
| 6.2.1 | bvs.php | 43 |
| 6.2.2 | ausgabe.php | 49 |
| 6.3 | Zur Entwicklung verwendete Software | 50 |
| 7 | Resümee und Ausblick | 52 |
| 7.1 | Nutzen für die Stadtbücherei | 52 |
| 7.1.1 | Vorteile für die Lektoren | 53 |

| | |
|--|-----------|
| 7.1.2 Vorteile für die EDV-Abteilung | 53 |
| 7.2 Erweiterungsmöglichkeiten | 53 |
| 7.3 Ausblick | 53 |
| Literaturverzeichnis | 55 |
| Beilage | 56 |
| Quellcode | 56 |
| Ehrenwörtliche Erklärung | 57 |

Abbildungsverzeichnis

| | | |
|------|--|----|
| 4.1 | Konzeption der Seitenstruktur | 25 |
| 4.2 | Entitäten-Beziehungs-Diagramm | 26 |
| 5.1 | Eingabemaske für die Anmeldung | 28 |
| 5.2 | Die Seitenübersicht | 29 |
| 5.3 | Der Bearbeitungsmodus | 30 |
| 5.4 | Der Bearbeitungsmodus — Darstellung mit dem textorientierten Web- browser w3m | 31 |
| 5.5 | Änderungsformular der Linkverwaltung | 31 |
| 5.6 | Last-Modified Liste | 32 |
| 5.7 | Modul: Benutzerverwaltung | 33 |
| 5.8 | Änderungsformular der Benutzerverwaltung | 33 |
| 5.9 | Zuteilungsverwaltung der Lektoratsgebiete | 34 |
| 5.10 | Zuteilungsverwaltung für einen Benutzer | 34 |
| 5.11 | Wahl des Seitenlayouts | 35 |
| 5.12 | Eingabeformular der Seitenverwaltung | 36 |
| 5.13 | Beispielseite ohne Unterüberschriften | 37 |
| 5.14 | Beispielseite mit Unterüberschriften | 38 |
| 6.1 | Relationales Datenbankmodell | 42 |

Tabellenverzeichnis

| | | |
|-----|---|----|
| 6.1 | Felder der Tabelle „users“ | 40 |
| 6.2 | Felder der Tabelle „gebiete“ | 40 |
| 6.3 | Felder der Tabelle „rechte“ | 40 |
| 6.4 | Felder der Tabelle „seiten“ | 41 |
| 6.5 | Felder der Tabelle „topics“ | 41 |
| 6.6 | Felder der Tabelle „links“ | 42 |
| 6.7 | Inhalte des assoziativen Arrays „SESSION“ | 44 |
| 6.8 | Beispielausgabe der Funktion tc() | 46 |
| 6.9 | Module und ihre Funktion (bvs.php) | 49 |

Abkürzungsverzeichnis

| | |
|-------|------------------------------------|
| BVS | Bookmarkverwaltungssystem |
| CoOL | Catalogue of Object Links |
| CSS | Cascading Style-Sheets |
| DB | Datenbank |
| DNS | Domain Name Service |
| HTML | Hypertext Markup Language |
| IP | Internet Protocol |
| IT | Information Technology |
| PHP | PHP Hypertext Preprocessor |
| RSN | Resource Discovery Network |
| SQL | Structured Query Language |
| StaBü | Stadtbücherei |
| SuUB | Staats- und Universitätsbibliothek |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| URN | Uniform Resource Name |
| UB | Universitätsbibliothek |
| ULF | Ulmer Fachinformation |
| WWW | World Wide Web |

Kapitel 1

Überblick

Bookmarksammlungen sind ein weit verbreitetes Phänomen im Internet. Fast jede Institution oder auch private Homepage bietet eine Liste mit weiterführenden oder themenverwandten Links an. Je nach Größe der Institution kann diese Liste lang sein und ein breites Themenspektrum umfassen. Eine Pflege solcher Linksammlungen kann auf unterschiedlichste Weise erfolgen. Technisch lassen sie sich in zwei Typen unterscheiden:

1. statische Linksammlungen,
2. dynamisch erzeugte Linksammlungen.

Die statischen Linksammlungen sind meist von Hand erstellte HTML-Seiten. Der Bearbeiter dieser Seiten arbeitet entweder direkt im HTML-Quellcode oder mit einem HTML-Editor, der aus den Eingaben automatisch den Quellcode erzeugt. Beim Typ der dynamisch generierten Linksammlungen wird der HTML-Code beim Aufruf der Seite on the fly generiert. Die Links werden aus einer Datenbank oder einer Datei gelesen und mittels einer Skriptsprache in HTML-Code umgewandelt. Folgende Vorteile lassen sich erkennen, wenn der zweite Typ benutzt wird:

- Die Bearbeitung kann in einer komfortablen Maske geschehen und kann ohne HTML-Kenntnisse des Bearbeiters bedient werden.
- Durch die Aufgabenteilung kann eine Spezialisierung erfolgen z.B. in den Bereichen Inhalt, Design und Programmierung.
- Durch die Trennung können evtl. Fehler leichter gefunden werden.
- Änderungen des Layouts bzw. Anpassungen an das Cooperate-Design der Institution sind leichter durchzuführen.

1.1 Ausgangspunkt und Zielsetzung

Bisher war die Erstellung einer Linkseite für die Stadtbücherei immer mit einem Doppelaufwand verbunden. Wenn der Lektor¹ einen passenden Link gefunden hatte, kopierte er ihn per „Copy and Paste“ in eine Word-Datei und schickte diese Datei dann an die EDV-Abteilung. Dort wurde mittels „Copy and Paste“ aus den Daten der Word-Datei eine HTML-Seite erstellt. Danach konnte der Lektor die Seite begutachten und Änderungswünsche bestellen. Die Änderungen laufen nach dem gleichen Schema ab. Diese kommen wieder per Word-Dokument in der EDV-Abteilung an. Die EDV-Abteilung darf dann im HTML-Code nach den zu ändernden Links suchen und diese per Hand abändern.

Das Bookmarkverwaltungssystem soll hier eine Erleichterung schaffen. Erstens soll die Doppelbearbeitung vermieden werden und zweitens soll die Dateneingabe und -pflege über eine Eingabemaske vereinfacht werden.

1.2 Inhalt dieser Arbeit

In Kapitel 2 wird gezeigt, welche Projekte und Lösungen es bereits in dem Bereich Bookmarksammlungen gibt und warum die Stadtbücherei noch ein weiteres Bookmarkverwaltungssystem (BVS) benötigt.

Das folgende Kapitel 3 definiert die Anforderungen an ein BVS im Allgemeinen und die von der Stadtbücherei vorgegebenen speziellen Wünsche.

In Kapitel 4 wird die Konzeption des Systems erarbeitet. Kapitel 5 stellt das realisierte System aus Sicht der verschiedenen Anwendergruppen dar. Kapitel 6 beschreibt die technische Realisierung des Systems.

Abschließend bietet Kapitel 7 eine Zusammenfassung der Ergebnisse und des erzielten Nutzens und liefert einen Ausblick auf denkbare zukünftige Erweiterungsmöglichkeiten.

¹Der einfachen Lesbarkeit halber verwende ich hauptsächlich die maskuline Form (z.B. Lektor, Benutzer, Administrator usw.). Die weibliche Form ist hierbei selbstverständlich unausgesprochen stets mitgemeint.

Kapitel 2

Stand der Technik

Beim Surfen durch das Web trifft man einige Linksammlungen an, die auf ein Verwaltungssystem im Hintergrund schließen lassen, angefangen bei Portalen, über gepflegte Web-Verzeichnisse von Suchmaschinen, bis hin zu wissenschaftlichen und kommerziellen Seiten. Die Idee, ein Bookmarkverwaltungssystem zu entwickeln, ist also nicht unbedingt neu. Dennoch scheint jeder Webmaster seine eigenen Vorstellungen zu verwirklichen. Mögliche Gründe dafür können sein:

- mannigfaltige Kombinationen von unterschiedlicher Hard- und Software,
- die Anforderungen vom Management und die unterschiedlichen Bedürfnisse der Webseiten-Besucher hinsichtlich Inhalt und Design,
- volle Ressourcen-Nutzung des Servers und Hardwaretuning in puncto Schnelligkeit,
- Anbindung an vorhandene Systeme,
- die unterschiedliche Funktionsweise des Datenbanksystems,
- die unterschiedlichen Anforderungen an die Datenbankstruktur,
- die Frage, ob zusätzliche Funktionen benötigt werden
- und wie sich die Linksammlung in die Internetpräsentation (Layout, Technik) eingliedern muss.

Im ersten Abschnitt werden Projekte vorgestellt und die daran interessanten Aspekte umrissen. Im zweiten Abschnitt wird herausgearbeitet, warum die Stadtbücherei ein weiteres BVS benötigt.

2.1 Beispielprojekte

An dieser Stelle möchte ich kurz ein paar Projekte vorstellen, die jeweils interessante Aspekte enthalten. Da es im Web zu vielen Themen Linksammlungen gibt, habe ich mich auf Systeme im Umfeld von Bibliotheken beschränkt.

2.1.1 SuUB Bremen

Die Staats- und Universitätsbibliothek in Bremen besitzt mit ihrer elektronischen Bibliothek „E-LIB“ eine sehr umfangreiche Linksammlung zu vielen Wissensgebieten. Interessant ist an dem dahinter stehenden Verwaltungssystem die exotische, aber dennoch flexible Art der Dateneingabe in die Datenbank:

Die Fachreferenten schicken ihre gesammelten Links per E-Mail an ein automatisiertes Auswertungsskript auf dem Server. Dort wird der Mailinhalt verarbeitet und entsprechend in die Datenbank integriert. Wenige Sekunden später steht der Link dann für die Homepagebesucher zur Verfügung.

Die Links sind übrigens Startpunkte eines Internet-Crawlers. Dieser indexiert die wichtigsten Seiten des Angebots und ermöglicht dann auch eine inhaltliche Suche.[11]

2.1.2 CoOL

Die Linksammlung der Universitätsbibliothek Braunschweig ist eine Sammlung von Bookmarks sowie ein Orientierungssystem für die Bibliothek. Das System wird CoOL (Catalog of Object Links) genannt und auch die Art der Dateneingabe ist „voll cool“:

Das Programm übernimmt die Bookmark-Sammlungen aus den Browsern der Lektoren und überträgt diese in die dahinter stehende Allegro-Datenbank. Nachträgliche Korrekturen und Ergänzungen können über das Datenbankprogramm vorgenommen werden.[15]

2.1.3 Zusammenschlüsse

BINE

Eines der ersten Projekte im Bereich Linksammlungen war das Projekt BINE (Bibliothek + Internet = Navigation + Erschließung) von der Stadtbibliothek Bremen in Zusammenarbeit mit der Universität Bremen, Forschungsgruppe Telekommunikation.[13] Im Zeitraum vom 1. Oktober 1996 bis zum verlängerten Ende am 31. Juni 1998 hat das Projekt erwiesen, dass „*die themenspezifische bewertende Selektion von Internetressourcen eine wichtige Dienstleistung der Bibliotheken in ihrer Aufgabe der Informationsvermittlung darstellt*“. [12]

ULF

ULF (Ulmer Fachinformation) ist ein Kooperationsprojekt der Ulmer Bibliotheken. So haben sich die UB, die FH-Bibliothek und die Stadtbibliothek eine Liste von gemeinsam vereinbarten 30 Fachgebieten je nach sinnvoller Zuständigkeit aufgeteilt, um wichtige Internetadressen anzubieten.

Die Hard- und Software wird von der Universitätsbibliothek in Ulm im Rahmen ihrer IT- und WWW-Infrastruktur zur Verfügung gestellt.[16]

Resource Discovery Network

Ein Beispiel für einen Zusammenschluss im Ausland soll hier das RDN (Resource Discovery Network) in England sein. Das RDN ist ein Zusammenschluss von über 60 wissenschaftlichen Einrichtungen wie z.B. das Natural History Museum und die British Library. Die RDS-Teilnehmer sammeln Seiten, die sie sorgfältig auswählen und mit einer Beschreibung versehen indexieren. Thematisch müssen die Links in das Umfeld Lernen, Lehren und Forschung passen.[10]

Projekt der Bertelsmann Stiftung

Mit dem Projekt „Deutsche Internetbibliothek“ möchte die Bertelsmann Stiftung ein Portal des Internets mit qualitativ hochwertigen Top-Links gründen. Neben Internetressourcen soll auch der Zugriff auf andere Bibliotheksmedien möglich sein.

An dem Projekt, das in dem Zeitraum vom 1. Januar 2002 bis zum 31.12.2004 läuft, nehmen 65 Bibliotheken unterschiedlicher Größenordnung teil. [2]

2.2 Ein BVS für die Stadtbücherei Stuttgart

Warum möchte die Stadtbücherei Stuttgart eine weitere Linksammlung anlegen?

Die Stadtbücherei möchte eine Linksammlung unter dem Motto „effektives Lernen“ bereitstellen. Die Themen sind dabei der Konzeption der Lernateliers (=Lektoratsgebiete) angegliedert.

Dabei soll es nicht das Ziel sein, viele Links zu einem Thema zu sammeln, sondern die geeignetsten.

Und wozu benötigt die Stadtbücherei dazu ein weiteres BVS?

Im Rahmen des Open Content Modells der Stadtbücherei Stuttgart (= „Freier Zugriff auf freie Information“) soll den Lektoren die Verwaltung der Links so einfach

wie möglich gestaltet werden.[14] Dabei soll die Gleichstellung der Medien berücksichtigt werden, die besagt, dass beispielsweise ein Buch ebenso viel wert ist wie eine Internet-Ressource.

Die Stadtbücherei Stuttgart war unter den ersten öffentlichen Bibliotheken, die ihren Homepagebesuchern eine Linksammlung anbot. Während andere Projekte längst wieder einschliefen, hat sich das Konzept der Stadtbücherei Stuttgart bewährt und ist bis jetzt am Laufen. Das BVS erweitert das Konzept und trägt dazu bei, dass die Lektoren volle Verantwortung für ihre Seiten haben, aber dadurch auch selbständiger arbeiten können.

Die Lektoren haben ein umfangreiches Wissen in ihren Fachgebieten und können so optimal ihr gesamtes Know-how in die Linksammlungen mit einbringen. Dazu benötigt es ein funktionsfähiges, schlagkräftiges System, das in die Hausstruktur passt — aber dennoch einfach, schnell und logisch zu bedienen ist.

Kapitel 3

Aufgabenstellung

Das BVS soll ein einfaches, handhabbares, effektives, wirtschaftliches System zum Erfassen und Bearbeiten von Links werden. In den folgenden Abschnitten werden die Anforderungen an ein solches System aufgestellt.

Abschnitt 3.1 benennt die allgemeinen Anforderungen an ein BVS sowie die benötigte Hard- und Software. Weiter werden dann im Abschnitt 3.2 die speziellen Vorgaben und Anforderungen von Seiten der Stadtbücherei erläutert.

3.1 Allgemeine Anforderungen an ein Bookmarkverwaltungssystem

Bei der Auswahl eines BVS sollte man folgende Kriterien beachten:

- Hat das BVS eine ansprechende, einfach zu bedienende Oberfläche?
- Gibt es die Möglichkeit, Datensätze neu zu erstellen, zu bearbeiten oder zu löschen?
- Sind für die Bearbeiter HTML-Kenntnisse notwendig?
- Gibt es ein Konzept zur Regelung der Nutzerrechte für die Lektoratsgebiete?
- Wie ist die Abfolge im Workflow?
- Unterstützung zu einheitlichem Inhalt
- Wird die Kooperation, Kommunikation und Koordination zwischen den verschiedenen Gebieten unterstützt?
- Werden die Möglichkeiten einer Redaktion und Aktualisierung unterstützt?
- Kann das System je nach Bedarf die mediale Bandbreite ausnutzen?

- Stehen eine Recherchemöglichkeit und ein Archiv zur Verfügung?

Viele dieser Fragen werden sich aufgrund des zur Verfügung stehenden Personals klären lassen. Auch die Geldmittel sind sicher ausschlaggebende Kriterien.

3.1.1 Was für Soft- und Hardware benötigt man für ein BVS?

Das Zusammenspiel von Hard- und Software ist ein wichtiger Faktor, um ein BVS betreiben zu können. Um ein BVS zu erstellen, benötigt man folgende Hard- und Software:

- einen Server mit einem funktionsfähigen HTTP-Server,
- einen Datenbank-Server — bestenfalls auf demselben Server wie der HTTP-Server,
- eine Skriptsprache, welche die ausgelesenen Datensätze in eine angenehm lesbare HTML-Seite verwandelt und
- eine Klient-Software, mit der man möglichst einfach und komfortabel die Datensätze in eine Datenbank schreiben kann.

Ein weiterer, nicht unerheblicher Faktor außerhalb der Systemkriterien ist die Kostenfrage.

3.1.2 Abgrenzung

Laut der Webseite „ContentManager“¹ ist ein Content Management System (CMS) ein „*Softwaresystem für das Administrieren von Webinhalten mit Unterstützung des Erstellungsprozesses basierend auf der Trennung von Inhalten und Struktur*“.[4]

So gesehen ist ein BVS auch eine Art CMS. Die Konzeption und Datenbankstruktur eines BVS ist aber im Gegensatz zu einem CMS auf die Verwaltung einer Linksammlung ausgerichtet.

Bei der Auswahl eines CMS sollte man folgende Kriterien bedenken:

- Ziele, Ansprüche,
- benötigte Funktionalitäten,
- technische Grundlagen,
- Design,

¹URL <http://www.contentmanager.de/> — Letzter Zugriff: 05.09.2002

- Anzahl der geplanten User und Server.[4]

Das gilt auch für ein BVS, da ein BVS ja nur eine Art spezialisiertes CMS ist. Der hauptsächlichste Unterschied liegt in der benötigten Funktionalität und der Datenbankstruktur.

3.2 Vorgaben der Stadtbücherei

Ziel soll es sein, ein einfach handhabbares, effektives, wirtschaftliches System zum Erfassen und Bearbeiten von Links zu erstellen. Dabei sollte die Ausgabe nicht von dem bisherigen Layout abweichen. Zu berücksichtigen sind hier die Anforderungen an die Funktionalität seitens der Stadtbücherei (1. Unterabschnitt), die technischen Voraussetzungen (2. Unterabschnitt) sowie die Beispielseiten (3. Unterabschnitt), die es in dem BVS zu verwirklichen gilt.

3.2.1 Spezielle Anforderungen an die Funktionalität von Seiten der Stadtbücherei

Von Seiten der Stadtbücherei gab es einige Wünsche und Anforderungen an das werdende System:

- Drei Benutzergruppen: Lektoren, Administratoren, Homepagebesucher.
- Das System sollte möglichst intuitiv bedienbar sein.
- Die Funktionen sollten sich von selbst erklären und nur geringen Einführungsbedarf benötigen.
- Das Eintragen der Daten sollte möglichst einfach durchführbar sein.
- Pflege der Daten sollte dem Prinzip von geringem Aufwand und bestmöglicher Funktionalität folgen.
- Die Lektoren sollen nur die Seiten ihres Verantwortungsbereiches bearbeiten dürfen.
- Die Ausgabe sollte nahtlos in das bisherige Webseitenkonzept eingliederbar sein.
- Das System soll so viele Voreinstellungen wie möglich selbst treffen.
- Der Administrator soll eine genaue Aufstellung darüber erhalten, wann von wem zuletzt welche Seite bearbeitet wurde.
- Das ausgegebene Ergebnis soll den bisherigen Seiten gleichen.

- Der Webseitenbesucher soll von der internen Umstellung nichts merken.

Während der Entwicklung des Programms kamen noch einige Ideen hinzu. Diese konnten leider noch nicht realisiert werden und sind im letzten Kapitel im Abschnitt „Erweiterungsmöglichkeiten“ aufgeführt.

3.2.2 Technische Voraussetzungen

Die technischen Voraussetzungen der Stadtbücherei richten sich vor allem an die bereits vorhandene Hard- und Software. Daraus ergibt sich:

- Auf der Klientseite ist der Netscape Navigator in der Version 4.73 sowie Microsofts Internetexplorer in der Version 5.0 installiert.
- Das BVS sollte browserunabhängig² programmiert sein und auch bei evtl. Softwareupdates funktionieren.
- Auf dem mit Linux betriebenen Server läuft der Webserver Apache in der Version 1.3.x. Die Skriptsprache PHP ist durch ein Modul in den Webserver eingebunden. PHP liegt in der Version 4.x vor. MySQL liegt auf dem Server in der Version 3.23.x vor.

Die Serverseite ist komplett mit „Open-Source“-Software ausgestattet. Es entstehen nur Kosten für die Hardware sowie für den Personalaufwand, um die Hard- und Software zu betreuen.

²Für ein Extrembeispiel der Browserunabhängigkeit beachten Sie bitte die Textbrowser-Darstellung in Abbildung 5.4 auf Seite 31

Kapitel 4

Konzeption

In diesem Kapitel wird die Konzeption des Systems nach den Anforderungen in Kapitel 3 erstellt.

Dabei werden zunächst im ersten Abschnitt die Benutzergruppen beschrieben, die später mit dem System umgehen sollen. Notwendige Sicherheitsaspekte werden im anschließenden Abschnitt 4.2 behandelt. In Abschnitt 4.3 wird die Softwarezusammenstellung erörtert. Mit der Konzeption der Datenbankstruktur und einem Entwurf der Seitenstruktur schließt das Kapitel.

4.1 Die Benutzergruppen des BVS

In diesem Abschnitt wird die Konzeption der drei Benutzergruppen erstellt. Anregung hierfür war die Diplomarbeit von David Prüm, der ein Dokumentenverwaltungssystem entwickelt hat.[9]

4.1.1 Lektoren

Die Hauptaufgabe der Lektoren ist das Eingeben, Bearbeiten und Löschen von Links. Sie füllen die Datenbank mit Inhalten, welche dann später vom Homepagebesucher angesehen werden können. Weiter soll es für die Lektoren möglich sein, innerhalb der Seiten die Unterüberschriften verwalten zu können.

Die Eingabeformulare für die Unterüberschriften sollten ähnlich aufgebaut sein wie die der Links. Die Eingabeformulare sollten einfach und übersichtlich gestaltet sein. Die gewünschte Funktion sollte leicht auffindbar sein und mit möglichst wenigen Mausklicks zum Ziel führen. Weiter sollten die Funktionen intuitiv bedienbar und selbsterklärend sein. Um nicht zu verwirren, soll das BVS ein dezentes, schlichtes Design bekommen.

Jeder Vorgang soll sofort in der Datenbank gespeichert werden und sofort zum Abruf zur Verfügung stehen.

Nach jedem Bearbeitungsschritt soll die Möglichkeit der Vorschau gegeben werden.

Der Lektor bekommt eine Aufstellung über die ihm zugeordneten Lektoratsgebiete mit einer Information darüber, wann die darin enthaltenen Seiten das letzte Mal geändert wurden. Diese Liste soll als Startseite nach dem Login erscheinen. Über die Liste soll man direkt in den Bearbeitungsmodus der einzelnen Seiten kommen.

4.1.2 Administratoren

Dem Administrator müssen alle Möglichkeiten des Lektors gegeben sein. Er soll auf alle Lektoratsgebiete Zugriff bekommen, um bei eventuellen Problemen der Lektoren hilfegebend eingreifen zu können.

Darüber hinaus stehen ihm noch eine ganze Reihe weiterer Funktionen zur Verfügung. Dem Administrator soll die Verwaltung der Benutzer und der Lektoratsgebiete unterstellt sein. Weiter soll es dem Administrator vorbehalten sein, neue Seiten in ein Lektoratsgebiet einzubinden und das Layout der Seiten festzulegen. Für das Einrichten der Seiten sind HTML-Kenntnisse erforderlich.

4.1.3 Homepagebesucher

Der Homepagebesucher soll lediglich die Ausgabe der Links zu sehen bekommen. Er darf keinen Zugang zu den Verwaltungsmodulen erhalten.

4.2 Sicherheitsaspekte

Um zu gewährleisten, dass die Links nur von bevollmächtigten Leuten bearbeitet werden, muss beim Aufruf des Systems zunächst eine Überprüfung erfolgen.

4.2.1 Intranet

Die Gesamtheit der Lektoren hat ihren Arbeitsplatz innerhalb des Intranets der Stadtbücherei. Als erste Sicherheitsstufe soll somit der Zugriff von außerhalb des Intranets gesperrt werden. Dies kann durch eine webserverspezifische Datei geschehen, welche die Zugriffe innerhalb eines Verzeichnisastes regelt.

Im Falle des Apache-Webservers heißt diese Datei „.htaccess“. Sie lässt nur Webanfragen durch, die der vorgegebenen IP-Adresse oder dem vorgegebenen DNS-Namen entsprechen.

4.2.2 Persönliche Anmeldung

Um mit dem System arbeiten zu können, soll sich der Benutzer authentifizieren müssen. Dies soll mittels Benutzername und Passwort geschehen.

Dies hat drei Gründe:

- Aus den eingegebenen Daten wird ermittelt, ob der Benutzer Administratorenrechte besitzt.
- Der Benutzername taucht in der „Zuletzt-Bearbeitet“-Liste auf. Durch die Authentifizierung kann sichergestellt werden, dass es sich tatsächlich um den Bearbeiter handelt. Schließlich ist er für seine Eingaben verantwortlich.
- Letztlich wird noch verhindert, dass ein Bibliotheksbesucher, der zufällig an einen bibliothekarischen Arbeitsplatz gelangt, Unsinn mit dem System treiben kann.

Ohne Anmeldung soll ein Arbeiten mit dem System nicht möglich sein.

4.2.3 Homepagebesucher

Die Ausgaberroutine soll von dem Verwaltungsteil des BVS völlig abgekoppelt sein. Auch wird für die Ausgabe ein gesonderter Datenbankbenutzer erstellt, der nur Lesezugriff auf die benötigten Tabellen innerhalb der Datenbank bekommt.

Somit kann eine ausnutzbare, nicht vorhersehbare Schwachstelle im Programmcode keinen Schaden in der Datenbank anrichten.

4.3 Software

4.3.1 Die Klientseite

Grundsätzlich wird bei der Programmierung darauf geachtet, dass das Programm plattformunabhängig funktionieren kann.

Zur Eingabeerleichterung soll Javaskript eingesetzt werden, sodass man dies als Browservoraussetzung ansehen kann. Bei Browsern mit ausgeschaltetem Javaskript können diese Funktionen nicht genutzt werden — das Programm ist jedoch trotzdem funktionsfähig.

Um ein optimales Ausgabeergebnis zu erzielen, muss der Browser Cascading Style-Sheets (CSS) unterstützen — dies gilt ebenso für die Vorschaufunktion.

Diese Anforderungen sind an den Arbeitsplätzen in der Stadtbücherei erfüllt.

4.3.2 Die Serverseite

Auf der Serverseite wird ein lauffähiger Webserver mit integriertem PHP-Modul und einem MySQL-Datenbankserver erwartet.

4.3.3 Benutzungsoberfläche

Die Benutzungsoberfläche soll nach dem Motto „Einfach, aber funktional“ erstellt werden. Überlegungen dabei sind, eine möglichst intuitive Nutzung zu ermöglichen und mit möglichst wenigen Mausklicks das Ziel zu erreichen.

Dadurch soll sich die Chance der Akzeptanz bei den Lektoren verbessern und kein Frust durch umständliche Bedienung aufkommen.

4.4 Fensteraufteilung

Die Aufteilung der Elemente auf dem Bildschirm soll möglichst einfach, logisch und übersichtlich gestaltet sein.

Der Benutzer sollte zu jeder Zeit wissen, auf welcher Seite, in welchem Lektoratsgebiet er sich gerade befindet. Dies soll durch einen weißen Streifen unterhalb des Logos angezeigt werden.

Auf der linken Seite sollen die Optionen angezeigt werden. Dabei steht auf jeden Fall der „Neu-anmelden“-Button immer an oberster Stelle. Dieser Button soll dem Benutzer zu jeder Zeit zur Verfügung stehen, um ihn vom System abzumelden und an den Anfangspunkt (Anmeldebildschirm) zurückzuführen. Je nach Anmeldestatus stehen dem Benutzer weitere Optionen zur Verfügung (siehe auch Abbildung 4.1). Für diesen Bereich soll etwa ein Fünftel der Fläche unterhalb des Logos verwendet werden.

Der restliche Teil rechts von dem Optionsstreifen soll als Interaktionsfläche für die jeweiligen ausgewählten Optionen benutzt werden, beispielsweise für den Bearbeitungsmodus und die Änderungsformulare.

Realisierte Beispielseiten siehe Kapitel 5 ab Seite 27.

4.4.1 Konzeption der Seitenstruktur

Die vereinfachte Sitemap in Abbildung 4.1 zeigt, welche Optionen für die jeweiligen Nutzergruppen an welcher Stelle zur Verfügung stehen sollen. Während für den Lektor ohne besondere Rechte nur die Seitenübersicht und der Bearbeitungsmodus zugänglich sind, stehen dem Administrator weitere Optionen zur Verfügung.

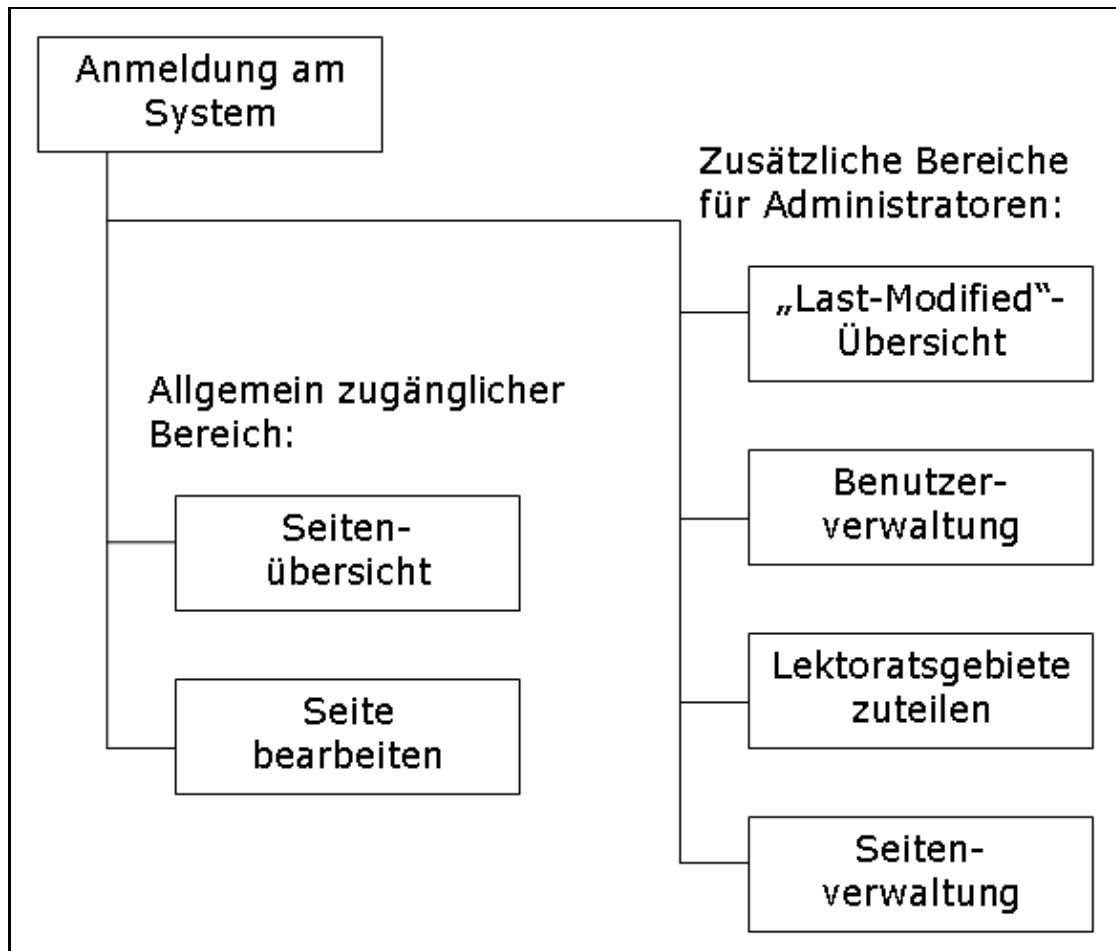


Abbildung 4.1: Konzeption der Seitenstruktur

4.5 Konzeption der Datenbankstruktur

Die Datenbank muss drei hauptsächliche Aufgaben erfüllen:

1. die Verwaltung der Benutzer,
2. die Verwaltung der Daten (Lektoratsgebiete, Seiten, Unterüberschriften, Links),
3. die Regelung der Rechte zwischen den Benutzern und den Daten.

In der Abbildung 4.2 wird in Form eines Entitäten-Beziehungs-Diagramms gezeigt, wie ich mir die Datenbankstruktur vorstelle und in welcher Beziehung die einzelnen Entitäten zueinander stehen.

Eine Entität (z.B. Benutzer) repräsentiert ein Objekt, das mehrere Eigenschaften beherbergen kann (z.B. die gesamten Daten des Benutzers).

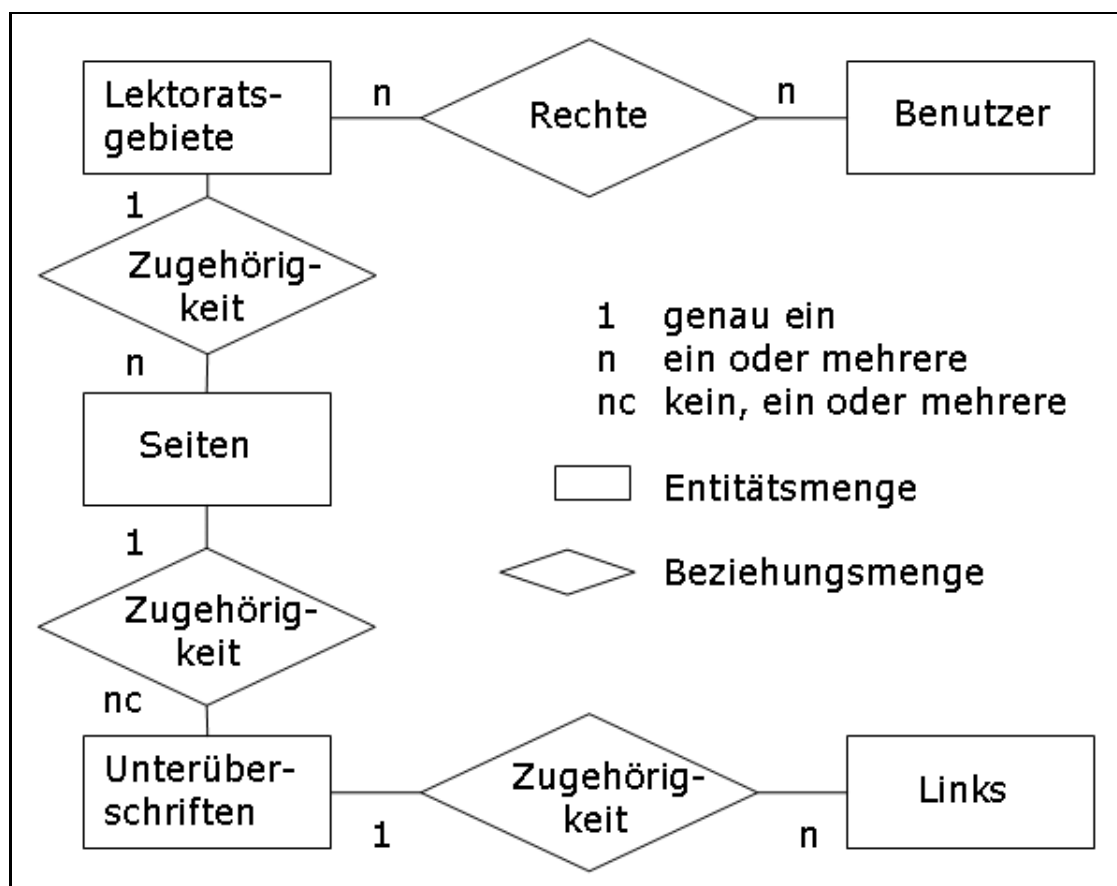


Abbildung 4.2: Entitäten-Beziehungs-Diagramm

Kapitel 5

System aus Anwendersicht

Zwei unterschiedliche Benutzergruppen treffen auf dem BVS zusammen:

- Lektoren, die Links verwalten, und
- Homepagebesucher, die nur die Ausgabe der Links sehen wollen.

Die Lektoren unterscheiden sich dann nochmals in zwei Gruppen:

- Administratoren und
- Lektoren.

Lektoren verwalten die Links in den ihnen zugeordneten Fachgebieten. Die Administratoren verwalten darüber hinaus die Fachgebiete, die Benutzer sowie die Zuordnung der Fachreferenten zu ihren Fachgebieten. Des Weiteren legen die Administratoren auch das Layout für die Seiten fest. Ein Administrator ist also ein Lektor mit mehr Rechten. Deshalb werden im Folgenden zunächst die Möglichkeiten des Lektors beschrieben. Beim Administrator werden dann nur noch die zusätzlichen Funktionen beschrieben.

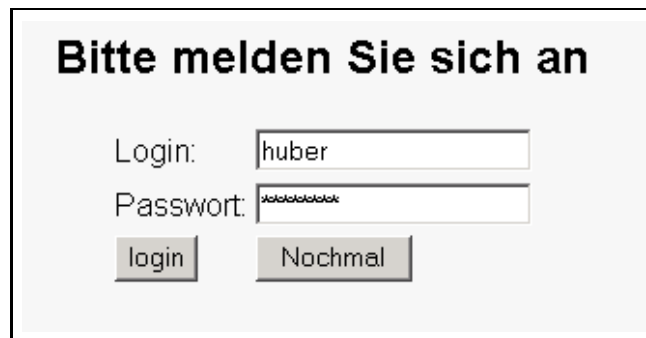
In diesem Kapitel wird das System aus Benutzersicht vorgestellt. Im ersten Abschnitt werden die Bereiche des Lektors vorgestellt. Im folgenden Abschnitt werden die weiteren Möglichkeiten des Lektors gezeigt. Der letzte Abschnitt zeigt das System aus Sicht der Homepagebesucher der Stadtbücherei.

5.1 Sicht des Lektors

Jeder Benutzer des BVS hat mindestens die Möglichkeiten eines Lektors. Diese sind, die Links der vom Administrator zugewiesenen Lektoratsgebiete zu verwalten. In diesem Abschnitt werden also die grundlegenden Optionen eines Lektors beschrieben.

5.1.1 Anmeldung

Bevor man mit dem BVS arbeiten kann, muss man sich anmelden. Das geschieht, indem man Benutzernamen und Passwort in die entsprechenden Felder eintippt und dies durch Betätigen der Schaltfläche „login“ bestätigt (Abbildung 5.1).



The image shows a login form with the following elements:


- Title: **Bitte melden Sie sich an**
- Label: Login: followed by an input field containing the text "huber".
- Label: Passwort: followed by an input field containing masked characters "*****".
- Buttons: "login" and "Nochmal".

Abbildung 5.1: Eingabemaske für die Anmeldung

5.1.2 Seitenübersicht

Nach erfolgreicher Anmeldung wird dem Lektor eine Aufstellung über seine Lektoratsgebiete sowie über die dazugehörigen Seiten gezeigt (Abbildung 5.2).

Diese Seite kann jederzeit über die Schaltfläche „Seitenübersicht“ aufgerufen werden. Weiter kann über diese Seite abgelesen werden, wann die einzelnen Seiten zuletzt bearbeitet wurden. In der Auflistung gibt es nun die Möglichkeit, eine bestimmte Seite zu bearbeiten oder mittels der Vorschaufunktion den bisherigen Stand anzusehen.



bvs **Bookmarkverwaltungssystem**
der Stadtbücherei Stuttgart

Lektoratsgebiet: keines ausgewählt -- Seite: keine ausgewählt

Neu anmelden

Optionen:

Seitenübersicht

Seite bearbeiten

Ihre Lektoratsgebiete und Seiten

Lektoratsgebiet **Moderne Technik**:

| Seite | zuletzt geändert am: | Optionen |
|---------------------|----------------------------|---|
| Mathematik | 2002-09-20 | [bearbeiten] [Vorschau] |
| Museen | 2002-09-21 | [bearbeiten] [Vorschau] |
| Naturwissenschaften | 2002-09-20 | [bearbeiten] [Vorschau] |
| Technik | 2002-09-20 | [bearbeiten] [Vorschau] |

Abbildung 5.2: Die Seitenübersicht

5.1.3 Bearbeitungsmodus

Über die Schaltfläche „Seite bearbeiten“ gelangt man in den Bearbeitungsmodus für die aktuell ausgewählte Seite. Die Anzeige für das aktuell ausgewählte Lektoratsgebiet sowie die aktuell ausgewählte Seite befindet sich in der Statuszeile unter dem Logo. Die Schaltfläche „Neu anmelden“ bewirkt, dass die momentane Bearbeitungssitzung beendet wird und das Anmeldeformular angezeigt wird. Um weiter zu arbeiten muss die Anmeldeprozedur wiederholt werden.

Im Bearbeitungsmodus (Abbildung 5.3 sowie Abbildung 5.4) kann der Lektor Unterüberschriften und Links neu anlegen, bearbeiten oder löschen.

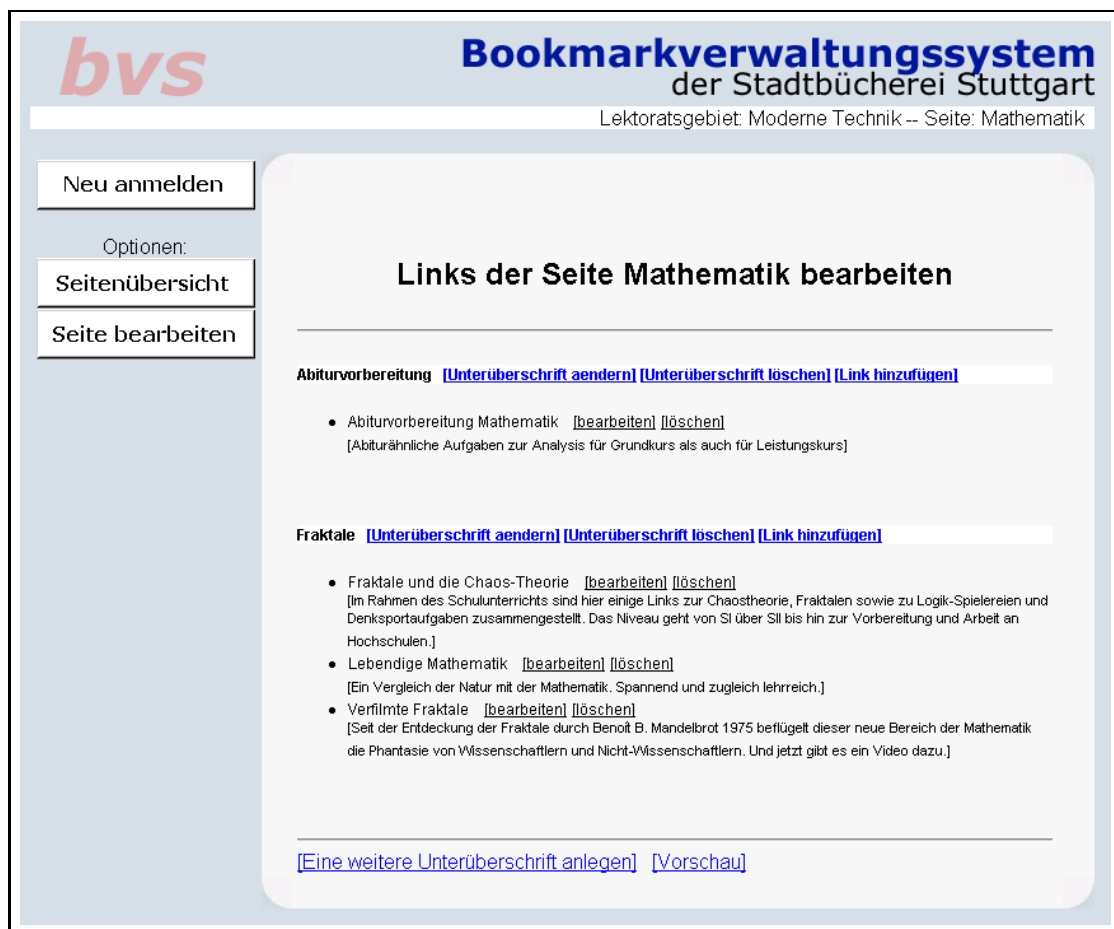


Abbildung 5.3: Der Bearbeitungsmodus

Bei einer Unterüberschrift kann nur auf die Bezeichnung Einfluss genommen werden. Die Bezeichnungen werden bei der Ausgabe alphabetisch sortiert. Bei der Erfassung der Links stehen drei Felder zur Verfügung:

- die Überschrift — eine kurze prägnante Bezeichnung des Links. Sie wird bei der Ausgabe als Linktext über die URL gelegt;
- die URL — sie liegt als Link unter der Überschrift;

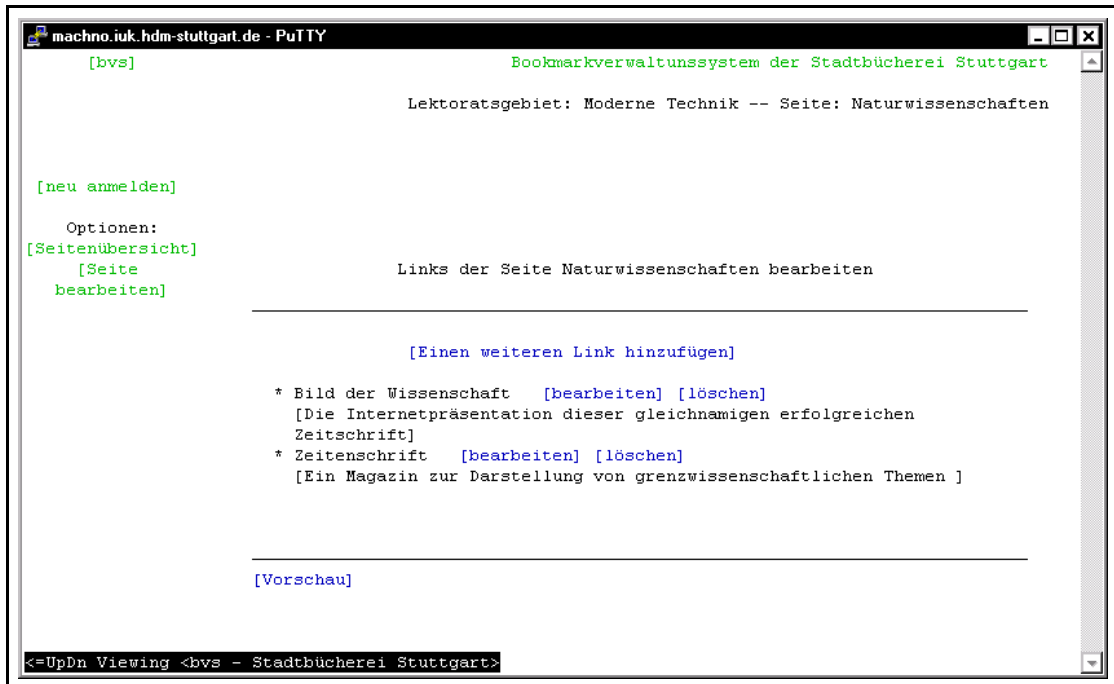


Abbildung 5.4: Der Bearbeitungsmodus — Darstellung mit dem textorientierten Webbrowser w3m

- die Beschreibung — eine kurze Beschreibung des Links. Sie wird in eckigen Klammern unter der Überschrift ausgegeben.

Die gleichen Felder stehen natürlich auch in der Änderungsfunktion (Abbildung 5.5). Die Links werden bei der Ausgabe alphabetisch sortiert.

Link bearbeiten

Url (Adresse) des Links:

Überschrift des Links:

Kurzbeschreibung:

Im Rahmen des Schulunterrichts sind hier einige Links zur Chaostheorie, Fraktalen sowie zu Logik-Spielereien und Denksportaufgaben zusammengestellt. Das Niveau geht von SI über SII bis hin zur Vorbereitung und Arbeit an

Abbildung 5.5: Änderungsformular der Linkverwaltung

5.2 Sicht des Administrators

Wie bereits erwähnt, stehen dem Administrator die gesamten Funktionen des Lektors zur Verfügung. Somit hat er auch Zugriff auf alle Lektoratsgebiete und deren Seiten.

5.2.1 Last-Modified Liste

Nach dem erfolgreichen Anmeldevorgang bekommt der Administrator eine Aufstellung über alle Lektoratsgebiete und die darin enthaltenen Seiten (Abbildung 5.6). In der Aufstellung sind Angaben dazu, wann die Seite zuletzt von einem Lektor sowie von einem Administrator bearbeitet wurde. Die damit verbundenen Namen sind mit der jeweiligen E-Mailadresse hinterlegt. Diese Liste ist jederzeit über die Schaltfläche „Last Modified“ erreichbar.

bvs **Bookmarkverwaltungssystem**
der Stadtbücherei Stuttgart

Lektoratsgebiet: Moderne Technik -- Seite: keine ausgewählt!

Neu anmelden

Verwaltung:
Benutzer
verwalten

Lektoratsgebiete
zuteilen

Lektoratsgebiete
verwalten

Seiten
verwalten

Neue Auswahl:
Lektoratsgebiet
wählen

Seite wählen

Kontrolle:
Last Modified

Optionen:
Seitenübersicht

Seite bearbeiten

Last-Modified Liste

Lektoratsgebiet **Fremde Länder:**

| Seite | Geändert am: | von User: | Geändert am: | von Admin: |
|------------|--------------|-----------------------|--------------|------------------------|
| Amerika | 2002-09-22 | Huber | 2002-09-22 | Köhler |
| Australien | 2002-09-22 | Huber | 2002-09-22 | Köhler |
| China | 2002-09-22 | Huber | 2002-09-22 | Köhler |

Lektoratsgebiet **Moderne Technik:**

| Seite | Geändert am: | von User: | Geändert am: | von Admin: |
|---------------------|--------------|-----------------------|--------------|------------------------|
| Mathematik | 2002-09-20 | Huber | 2002-09-18 | Maier |
| Museen | 2002-09-21 | Huber | 2002-09-18 | Maier |
| Naturwissenschaften | 2002-09-20 | Huber | 2002-09-20 | Köhler |
| Technik | 2002-09-20 | Huber | 2002-09-20 | Köhler |

Lektoratsgebiet **Sport:**
Noch keine Seiten in diesem Gebiet.

Abbildung 5.6: Last-Modified Liste

5.2.2 Benutzerverwaltung

Im Funktionsmodul Benutzerverwaltung (Abbildung 5.7) können Benutzer neu angelegt, bearbeitet oder gelöscht werden. Der Loginname ist in Kombination mit dem Passwort für den Anmeldevorgang wichtig. In der Ausgabe kommt der Bearbeitername mit als Link hinterlegter E-Mailadresse zum Tragen. Durch die Admin-Variable wird festgelegt, ob der Benutzer das erweiterte Menü zu sehen bekommt und ob er Zugriff auf alle Lektoratsgebiete hat (Abbildung 5.8).



Abbildung 5.7: Modul: Benutzerverwaltung

The screenshot shows the 'Benutzerdaten ändern' (Change User Data) form. The form contains the following fields and controls:

- Loginname:** Input field with value 'huber'.
- Passwort:** Input field.
- Bearbeitername für die Ausgabeseite:** Input field with value 'Huber'.
- Emailadresse für die Ausgabeseite:** Input field with value 'huber@stuttgart.de'.
- Administrator:** Dropdown menu with options 'Nein' (selected) and 'Ja'.
- Buttons:** 'Ändern' and 'Abgabe zurücksetzen'.

Abbildung 5.8: Änderungsformular der Benutzerverwaltung

5.2.3 Rechtevergabe

In dem Menüpunkt „Rechtevergabe“ (Abbildung 5.9) können einem Benutzer ohne Administratorenrechte ein oder mehrere Lektoratsgebiete zur Bearbeitung zugewiesen werden. Nach der Wahl des Benutzers über den Link „Rechte bearbeiten“ wird durch Folgen der jeweiligen Links entweder ein Recht vergeben oder das Recht entzogen. Ob ein Bearbeitungsrecht für eine Seite vorliegt, ist an dem vorangestellten „X“ (Abbildung 5.10) erkennbar.



Abbildung 5.9: Zuteilungsverwaltung der Lektoratsgebiete

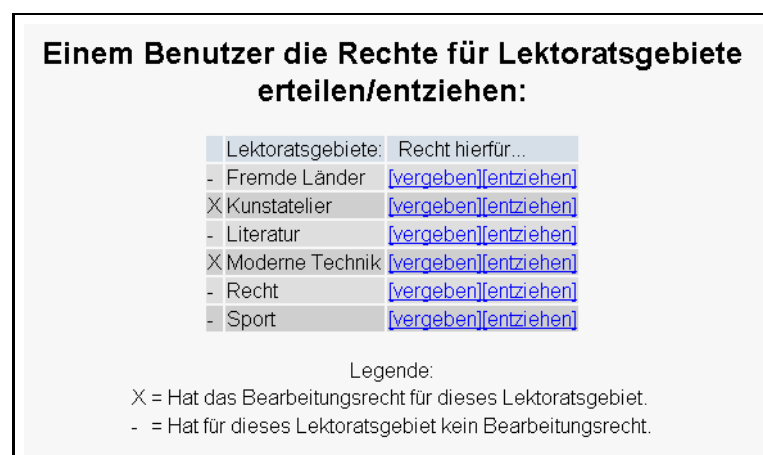


Abbildung 5.10: Zuteilungsverwaltung für einen Benutzer

5.2.4 Verwaltung der Lektoratsgebiete

Im Modul „Lektoratsgebiete verwalten“ hat man die Möglichkeit, Lektoratsgebiete neu zu erstellen, zu ändern oder zu löschen. Beim Lektoratsgebiet kann man die Bezeichnung wählen, was keinen Einfluss auf die Ausgabe hat. Sie ist nur programmintern für die Rechtevergabe und die Gruppierung der einzelnen Seiten notwendig.

5.2.5 Verwaltung der Seiten

Im Menüpunkt „Seiten verwalten“ erscheint dann eine Aufstellung der Seiten des aktuell ausgewählten Lektoratsgebietes. Diese Seiten können nun gelöscht oder bearbeitet werden. Weiter gibt es die Möglichkeit eine weitere Seite hinzuzufügen (Abbildung 5.12). Nach der Speicherung der Daten muss das Layout für die Seite gewählt werden (Abbildung 5.11). Dabei ist es möglich, entweder eine Seite mit oder ohne Unterüberschriften zu erstellen. Mit einem Mausklick auf die entsprechende Abbildung wird das Layout in die Datenbank geschrieben.

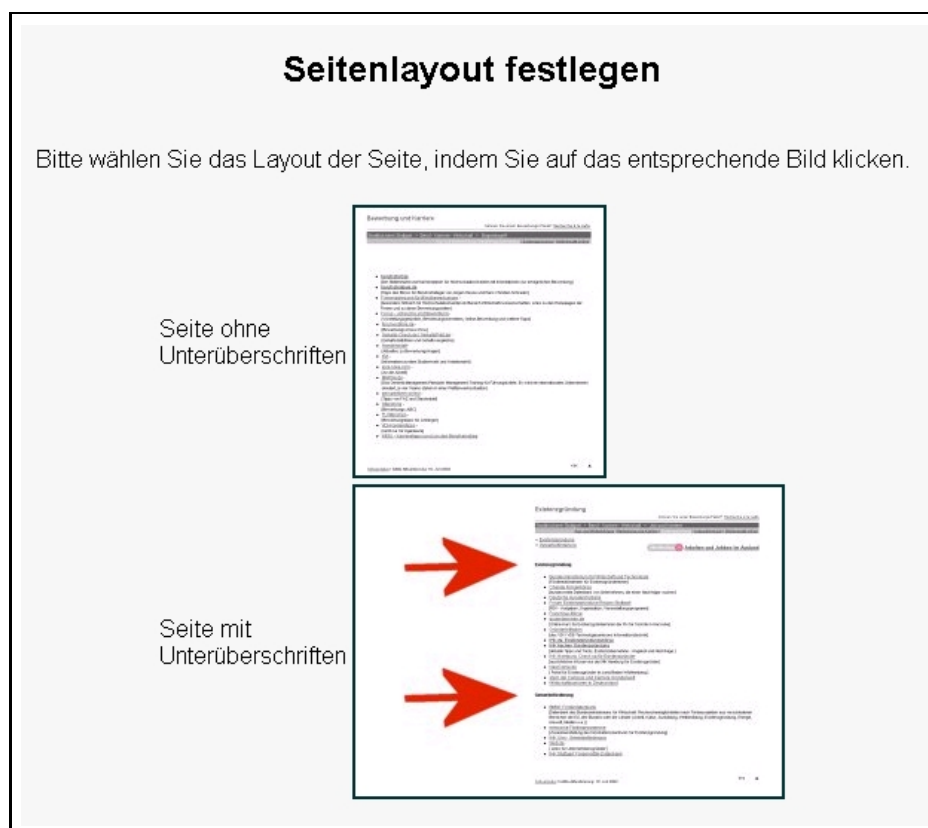


Abbildung 5.11: Wahl des Seitenlayouts

Daten zur Erfassung einer neuen Seite

Seitenüberschrift: [\[Hilfe\]](#)

Header-Bereich: [\[Hilfe\]](#)

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="Author" content="Stadtbücherei Stuttgart, Evi Schönleber">
<meta name="Description" content="">
<title>Stadtbücherei Stuttgart: Beruf - Karriere - Wirtschaft: Seitentitel</title>

<link rel="stylesheet" href="files_stabue/linklist.css" type="text/css">

<script language="JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt
(appVersion)==4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight;
```

Werbung: [\[Hilfe\]](#)

```
<div align="right"><span class="linkklein"> </span><span
class="linkklein">Kennen Sie unser Bewerbungspaket? <a
href="http://www.stuttgart.de/stadtbuecherei/infoanbieter/recherche_details.htm#m" target="_top">
  Recherche à la carte</a></span> </div>
```

Navigation: [\[Hilfe\]](#)

```
<div align="left"> <a href="http://www.stuttgart.de/stadtbuecherei/" target="_self"
class="stabilink">
  Stadtbücherei Stuttgart > </a><a
href="http://www.stuttgart.de/stadtbuecherei/bkw" _self class="stabilink"> Beruf
- Karriere - Wirtschaft </a><span class="stabilink">>
```

Medientipp: [\[Hilfe\]](#)

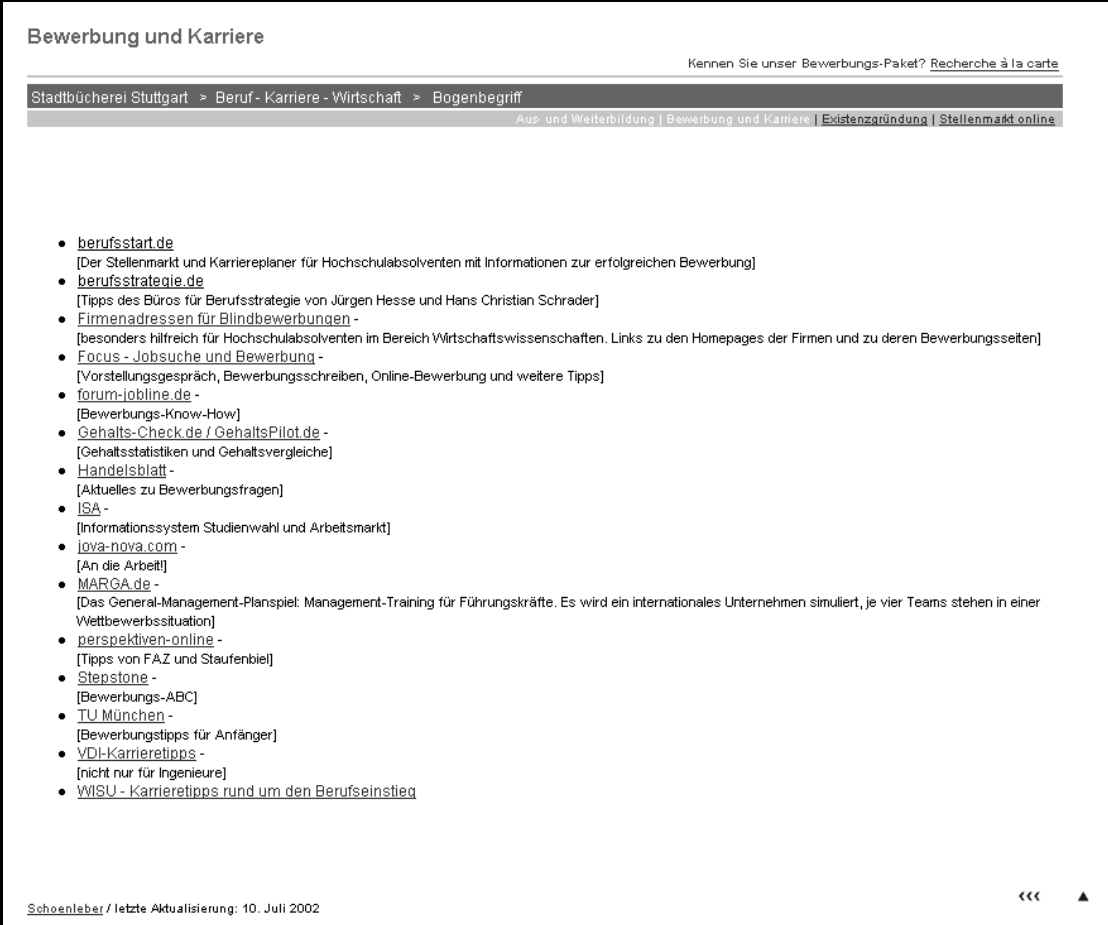
```
<td>
<table border="0" cellspacing="0" cellpadding="0" align="right">
<tr>
  <td height="15"> <a href="javascript:;"
onMouseOut="MM_swapImgRestore()" onMouseOver="MM_swapImage
('Image1','pics_stabue/medientipp_c.gif',1)" onClick="MM_openBrWindow
('http://141.31.128.162/wwwopac/woload.asp?SQL=from%20MEDIEN%20ME,%
20SUCHSYST%20SUN%20where%20ME.MEDIENNR%20=%
20SUN.MEDIENNR%20and%20(%20(SUN.WERTSUCH%20like%20%27hef%
```

Abbildung 5.12: Eingabeformular der Seitenverwaltung

5.3 Sicht des Benutzers (Homepagebesuchers)

Der Benutzer, der die Webseite der Stadtbücherei aufruft, kommt mit dem eigentlichen Bookmarkverwaltungssystem nicht in Berührung. Ihm steht lediglich die Ausgabefunktion zur Verfügung. Die Ausgabe unterscheidet sich nicht von den früheren Seiten der Stadtbücherei.

Abbildung 5.13 zeigt eine Ausgabeseite ohne Unterüberschriften. In Abbildung 5.14 wird eine Seite mit Unterüberschriften gezeigt.



The screenshot shows a web page with the following content:

Bewerbung und Karriere Kennen Sie unser Bewerbungs-Paket? Recherche à la carte

Stadtbücherei Stuttgart > Beruf - Karriere - Wirtschaft > Bogenbegriff

Aus- und Weiterbildung | Bewerbung und Karriere | Existenzgründung | Stellenmarkt online

- [berufsstart.de](#)
[Der Stellenmarkt und Karriereplaner für Hochschulabsolventen mit Informationen zur erfolgreichen Bewerbung]
- [berufsstrategie.de](#)
[Tipps des Büros für Berufsstrategie von Jürgen Hesse und Hans Christian Schrader]
- [Firmenadressen für Blindbewerbungen](#) -
[besonders hilfreich für Hochschulabsolventen im Bereich Wirtschaftswissenschaften. Links zu den Homepages der Firmen und zu deren Bewerbungsseiten]
- [Focus - Jobsuche und Bewerbung](#) -
[Vorstellungsgespräch, Bewerbungsschreiben, Online-Bewerbung und weitere Tipps]
- [forum-jobline.de](#) -
[Bewerbungs-Know-How]
- [Gehalts-Check.de / GehaltsPilot.de](#) -
[Gehaltsstatistiken und Gehaltsvergleiche]
- [Handelsblatt](#) -
[Aktuelles zu Bewerbungsfragen]
- [ISA](#) -
[Informationssystem Studienwahl und Arbeitsmarkt]
- [lova-nova.com](#) -
[An die Arbeit!]
- [MARGA.de](#) -
[Das General-Management-Planspiel: Management-Training für Führungskräfte. Es wird ein internationales Unternehmen simuliert, je vier Teams stehen in einer Wettbewerbssituation]
- [perspektiven-online](#) -
[Tipps von FAZ und Stautenbiel]
- [Stepstone](#) -
[Bewerbungs-ABC]
- [TU München](#) -
[Bewerbungstipps für Anfänger]
- [VDI-Karrieretipps](#) -
[nicht nur für Ingenieure]
- [WISU - Karrieretipps rund um den Berufseinstieg](#)

Schoenleber / letzte Aktualisierung: 10. Juli 2002 <<< ▲

Abbildung 5.13: Beispielseite ohne Unterüberschriften

Existenzgründung

Kennen Sie unser Bewerbungs-Paket? [Recherche à la carte](#)

Stadtbücherei Stuttgart > Beruf- Karriere - Wirtschaft > Job und Karriere

Aus- und Weiterbildung | Bewerbung und Karriere | Existenzgründung | Jobsuchformular | Stellenmarkt online

▼ Existenzgründung
▼ Gewerbeförderung

Medientipp **Arbeiten und Jobben im Ausland**

Existenzgründung

- [Bundesministerium für Wirtschaft und Technologie](#)
[Fördermaßnahmen für ExistenzgründerInnen]
- [Change Firmenbörse](#)
[bundesweite Datenbank von Unternehmen, die einen Nachfolger suchen]
- [Deutsche Ausgleichsbank](#)
- [Forum Existenzgründung Region Stuttgart](#)
[FEX - Aufgaben, Organisation, Veranstaltungsprogramm]
- [Franchise-Börse](#)
- [gruenderinnen.de](#)
[Online-Kurs für Existenzgründerinnen der FH für Technik in Karlsruhe]
- [Gründerleitfaden](#)
[des VDI / VDE-Technologiezentrums Informationstechnik]
- [IHK.de: Existenzgründungsbörse](#)
- [IHK Aachen: Existenzgründung](#)
[aktuelle Tipps und Tricks, Existenzübernahme - Angebot und Nachfrage]
- [IHK Hamburg: Check up für Existenzgründer](#)
[ausführlicher Infoservice der IHK Hamburg für Existenzgründer]
- [NewCome.de](#)
[Portal für Existenzgründer im Land Baden-Württemberg]
- [stern.de/ Campus und Karriere Gründerwelt](#)
- [Wirtschaftsjunioren in Deutschland](#)

Gewerbeförderung

- [BMWi: Förderdatenbank](#)
[Datenbank des Bundesministeriums für Wirtschaft. Recherchemöglichkeiten nach Förderprojekten aus verschiedenen Bereichen der EU, des Bundes oder der Länder (Arbeit, Kultur, Ausbildung, Weiterbildung, Existenzgründung, Energie, Umwelt, Medien u.a.)]
- [newcome Förderprogramme](#)
[Zusammenstellung des Informationszentrums für Existenzgründung]
- [IHK Ulm - Gewerbeförderung](#)
- [Web.de](#)
[Links für Unternehmensgründer]
- [IHK Stuttgart: Fördermittel-Datenbank](#)

Schoenleber / letzte Aktualisierung: 10. Juli 2002

◀◀◀ ▲

Abbildung 5.14: Beispielseite mit Unterüberschriften

Kapitel 6

Realisierung

Den Anforderungen in Kapitel 3 und der Konzeption in Kapitel 4 folgt hier die Umsetzung in das webbasierte Bookmarkverwaltungssystem. Die Ausführungen sind teilweise sehr technisch gehalten und setzen ein gewisses Grundwissen in der Programmierung, vor allem mit der Skriptsprache PHP voraus. Dieses Kapitel kann übersprungen werden, ohne dass dabei der Anschluss verloren wird.

In Abschnitt 6.1 wird die Datenbankstruktur aufgezeigt und anhand eines relationalen Datenbankmodells verdeutlicht. Abschnitt 6.2 beschreibt die Implementierung in PHP-Code. Die zur Entwicklung verwendete Software wird in Abschnitt 6.3 vorgestellt.

6.1 Datenbankstruktur

Abbildung 4.2 zeigt die Datenbankstruktur als Entitäten-Beziehungs-Diagramm. Daraus ergeben sich die im Folgenden aufgelisteten Tabellen. Der Datentyp und die Beschreibung der einzelnen Tabellenspalten sind den Tabellen 6.1 bis 6.5 zu entnehmen.

Wie die Tabellen untereinander verknüpft werden, ist in Abbildung 6.1 auf Seite 42 in einem relationalen Datenbankmodell dargestellt.

Tabelle „users“: In dieser Tabelle werden die Benutzer des BVS verwaltet. Das Feld ‚admin‘ bestimmt hierbei, ob es sich bei dem Benutzer um einen Administrator handelt. Ein Administrator hat Zugriff auf alle Funktionen und Lektoratsgebiete des BVS.

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|-----------------|-----------------|---------------------------------|------------------|
| uid | integer | eindeutige Nummer des Benutzers | Primärschlüssel |
| login | varchar(255) | Loginname des Benutzers | |

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|-----------------|-------------------|--|------------------|
| password | varchar(255) | verschlüsseltes Passwort des Benutzers | |
| realname | varchar(255) | richtiger Name des Benutzers | |
| admin | enum('Ja','Nein') | Administratoren-Rechte? Ja oder Nein. | |

Tabelle 6.1: Felder der Tabelle „users“

Tabelle „gebiete“: Diese Tabelle umfasst die unterschiedlichen Lektoratsgebiete.

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|-----------------|-----------------|---|------------------|
| gid | integer | eindeutige Nummer des Lektoratsgebietes | Primärschlüssel |
| beschr | varchar(255) | Bezeichnung des Lektoratsgebietes | |

Tabelle 6.2: Felder der Tabelle „gebiete“

Tabelle „rechte“: Hier werden die Zuordnungen der Benutzer aus der Tabelle „users“ zu den Lektoratsgebieten aus der Tabelle „gebiete“ geregelt.

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|-----------------|-----------------|---|------------------------------|
| rid | integer | eindeutige Nummer des vergebenen Rechts | Primärschlüssel |
| gid | integer | Nummer des Lektoratsgebietes | Fremdschlüssel aus „gebiete“ |
| uid | integer | Nummer des Benutzers | Fremdschlüssel aus „users“ |

Tabelle 6.3: Felder der Tabelle „rechte“

Tabelle „seiten“: Diese Tabelle enthält die Daten der einzelnen Seiten. Die Seiten sind genau einem Lektoratsgebiet der Tabelle „gebiete“ unterstellt.

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|-----------------|-----------------|------------------------------|------------------------------|
| sid | integer | eindeutige Nummer der Seite | Primärschlüssel |
| gebiet | integer | Nummer des Lektoratsgebietes | Fremdschlüssel aus „gebiete“ |
| head | text | HTML-Kopfdaten der Seite | |
| seitenueberschr | varchar(255) | Bezeichnung der Seite | |

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|--------------------------|-----------------|--|-------------------------------|
| streifen- werbung | text | HTML-Daten der Werbung | |
| navi- ueber- blick | text | HTML-Daten der Seitennavi- gation | |
| medien- tipp | text | HTML-Daten des Medientipps | |
| lm_user | date | Datum der letzten Bearbei- tung durch den User | |
| mb_user | integer | Nummer des Users, der die Sei- te zuletzt bearbeitet hat. | Fremdschlüssel aus „users“ |
| lm_admin | date | Datum der letzten Bearbei- tung durch den Admin | |
| mb_admin | integer | Nummer des Admins, der die Seite zuletzt bearbeitet hat. | Fremdschlüssel aus „users“ |

Tabelle 6.4: Felder der Tabelle „seiten“

Tabelle „topics“: In dieser Tabelle sind die Unterüberschriften der Seiten gespeichert. Jede Seite hat mindestens eine Unterüberschrift, da die Zuordnungen der Links über diese Tabelle abgewickelt werden.

Soll nun eine Seite ohne Unterüberschrift auf dem Bildschirm dargestellt werden, wird in die Beschreibung der String ‚kein Topic‘ eingetragen. Unterüberschriften mit diesem Eintrag werden bei der Ausgabe übergangen.

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|-----------------|-----------------|---|-----------------------------|
| tid | integer | eindeutige Nummer der Unterüberschrift | Primärschlüssel |
| beschr | varchar(255) | Bezeichnung der Un- terüberschrift | |
| seite | integer | Nummer der Seite | Fremdschlüssel aus „seiten“ |

Tabelle 6.5: Felder der Tabelle „topics“

Tabelle „links“: Diese Tabelle erfasst die Links. Jeder Link ist genau einer Unterüberschrift zugeordnet.

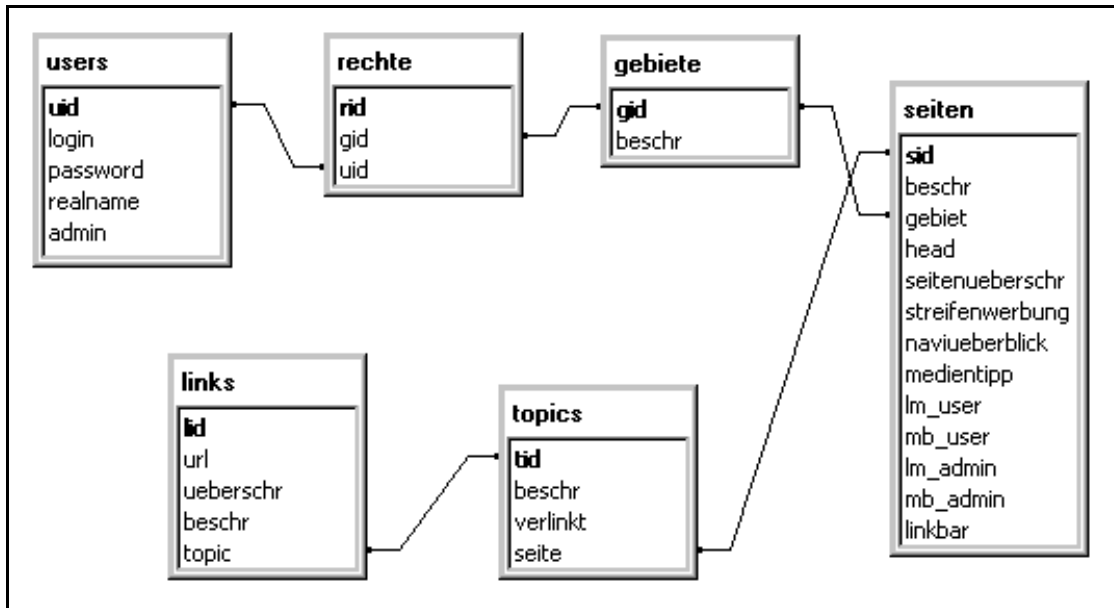


Abbildung 6.1: Relationales Datenbankmodell

| <i>Feldname</i> | <i>Datentyp</i> | <i>Inhalt</i> | <i>Schlüssel</i> |
|-----------------|-----------------|-----------------------------|-----------------------------|
| lid | integer | eindeutige Nummer des Links | Primärschlüssel |
| url | varchar(255) | URL des Links | |
| ueberschr | varchar(255) | Überschrift des Links | |
| beschr | text | Beschreibung des Links | |
| topic | integer | Nummer der Unterüberschrift | Fremdschlüssel aus „topics“ |

Tabelle 6.6: Felder der Tabelle „links“

Die Tabellen mit ihren ganzen Spalten und ihre Beziehungen untereinander sind nochmals deutlich in Abbildung 6.1 in einem relationalen Datenbankmodell dargestellt.

6.2 Programmierung

Ich habe das System der Funktion nach in zwei Dateien aufgeteilt. Die eine Datei (bvs.php) enthält den Quellcode für den Verwaltungsteil des BVS, die andere (ausgabe.php) ist lediglich für die Ausgabe zuständig. Die Dateien sind in der Verzeichnisstruktur voneinander getrennt, da die Eingabedatei Zugangsbeschränkungen hat. Die Ausgabedatei soll von überall her aufrufbar sein.

Die folgende Beschreibung der Quellcodeauszüge folgt dem Ablauf des Programmcodes.

6.2.1 bvs.php

Das Programm `bvs.php` ist der Verwaltungsteil des BVS.

Für die Vorschaufunktion wird auf die Datei `ausgabe.php` zurückgegriffen.

Session-Management

Seit PHP4 ist das Session-Management ein fester Bestandteil des PHP-Kerns. Eine Session beginnt mit dem Betreten der HTML-Seite. Sie endet, falls der User nichts mehr macht, nach einem vorher bestimmten Timeout — oder explizit durch das Abmelden des Users. In einer Session kann man Variable festlegen, welche in weiteren Sessions Gültigkeit besitzen und ihre Werte entsprechend beibehalten. Die Variableninhalte der Sessions werden in einer Datei auf dem Server gespeichert. Dem Webbrowser wird lediglich eine Sessionvariable in Form eines Cookies übergeben. Sollte der Surfer Cookies deaktiviert haben, übergibt PHP automatisch die Sessionvariable per URL der nächsten Seite (z.B. `http://www.server.tld/bvs.php?PHPSESSID=ffb45da68a32bc6bf1b6de158f0b2b9e`).

Die Session wird mit der Funktion `session_start()` begonnen. PHP sucht dann im HTTP-Header (bei aktivierten Cookies) oder in dem Uniform Resource Identifier (URI) (bei deaktivierten Cookies) nach einer Sessionvariable. Findet PHP keine Variable, wird eine neue Session begonnen. Ansonsten wird die vorherige Session fortgesetzt. Mit dem folgenden Befehl wird dann eine zuvor gespeicherte Variable eingelesen bzw. neu initialisiert:

```
session_register('SESSION');
```

Die Variable wird in dem assoziativen Array „SESSION“ abgespeichert. Das hat den Vorteil, dass nur eine Anmeldung vorgenommen werden muss, aber alle darin gespeicherten Variablen zur Verfügung stehen. Die Eigenschaft „Assoziativ“ hat die praktische Bedeutung, dass die Werte in dem Array nicht über eine fortlaufende Nummer angesprochen werden, sondern über Schlüsselworte. Der Inhalt des Arrays „SESSION“ kann der Tabelle 6.7 entnommen werden.

| <i>Index</i> | <i>Inhalt</i> | <i>Beispiel</i> |
|--------------|--|-----------------|
| admin | Hat der aktuelle Benutzer Administratorenrechte? | Ja oder Nein |
| gid | Die ID des aktuellen Lektoratsgebietes | 14 |

| <i>Index</i> | <i>Inhalt</i> | <i>Beispiel</i> |
|-----------------|---|-----------------|
| gebietebschr | Die Bezeichnung des aktuellen Lektoratsgebietes | Moderne Technik |
| login | Der Loginname des aktuellen Benutzers | schmidt |
| realname | Der bürgerliche Name des aktuellen Benutzers | Ewald Schmidt |
| sid | Die ID der aktuellen Seite | 7 |
| seitenueberschr | Die Bezeichnung der aktuellen Seite | Museen |

Tabelle 6.7: Inhalte des assoziativen Arrays „SESSION“

Die Anmeldung des Arrays läuft folgendermaßen ab:

```

$errors='';
if ($aktion=='neu_anmelden') {
    if (!session_unregister('SESSION')) {
        echo "Fehler beim Abmelden der Variablen SESSION! <br />
            Bitte melden Sie sich neu an.";
    }
    if (!session_destroy()) {
        echo "Fehler beim Beenden der Sitzung! <br />
            Bitte melden Sie sich neu an.";
    }
    header("Location: $PHP_SELF");
}

```

Login und Benutzerdaten prüfen

Wenn durch das vorhergehende HTML-Formular die Variable `$login` gesetzt wurde, wird die Variable `$login` und `$password` mit den Datensätzen in der Datenbank verglichen. Beim Anlegen eines neuen Benutzers wird das Passwort verschlüsselt in der Datenbank abgelegt. Bei der Abfrage in der Datenbank wird deshalb das Passwort mit der `PASSWORD()`-Funktion von MySQL erst verschlüsselt und dann mit den Werten in der Datenbank verglichen. Die MySQL-Abfrage lautet somit:

```

$query="SELECT *
    FROM users
    WHERE \"$username\" = login
    AND password = PASSWORD(\"$userpass\")";

```

Wenn bei dem Vergleich ein Datensatz Übereinstimmung gefunden hat, werden die weiteren Datensatzinhalte in den Array SESSION gelesen.

Sollte keine Übereinstimmung gefunden werden, wird die Login-Prozedur so lange wiederholt, bis es bei den Feldern Benutzername und Passwort zu Übereinstimmungen mit der Eingabe kommt - oder die Prozedur vom Benutzer abgebrochen wird.

Funktionen

Eine Funktion ist ein Teil des Quellcodes, der eine bestimmte Befehlsfolge ausführt. Sie wird nur einmal programmiert und kann an mehreren Stellen im Quellcode benutzt werden.

Eine dieser Funktionen heißt tc (=Table Color). Diese Funktion steuert die Farbwahl in den generierten HTML-Tabellen. Die Tabellenzeilen bekommen im Wechsel verschiedene Farben zur besseren Überschaubarkeit (beispielsweise die Tabelle der Lektoratsgebietsauflistung in Abbildung 5.2 auf Seite 29).

Der Funktion muss ein Zähler übergeben werden. Ist dieser Zähler eine gerade Zahl, gibt die Funktion einen anderen HTML-Farbwert zurück als bei einer ungeraden Zahl. Der Zähler ergibt sich aus einer Bearbeitungsschleife (z.B. beim Auslesen von Datensätzen).

```
function tc($zahl) {
$gerade_tabfarbe = "#CCCCCC";
$ungerade_tabfarbe = "#DDDDDD";
if ( ($zahl/2) != floor($zahl/2)) {
    return $gerade_tabfarbe;
} else {
    return $ungerade_tabfarbe;
}
}
```

Die Funktion wird dann innerhalb des HTML-Tags <td> (Tabeldata) in einer Tabellenumgebung eingebunden:

```
echo "<td bgcolor=\"".tc($i).\">";
```

wobei \$i eine Zählervariable innerhalb einer Schleife darstellt. Die Ausgabe dieser Zeile, nachdem sie von PHP abgearbeitet und in HTML-Code umgewandelt wurde, sieht dann folgendermaßen aus:

| Zählervariable \$i | Ausgabe |
|--------------------|------------------------|
| 0 | <td bgcolor="#CCCCCC"> |
| 1 | <td bgcolor="#DDDDDD"> |

```

2          | <td bgcolor="#CCCCCC">
usw. ...  |

```

Tabelle 6.8: Beispielausgabe der Funktion tc()

Auswahlmenüs für Lektoren und Administratoren

Nach dem Einloggen ins System erscheint am linken Rand ein Auswahlmenü. Dieses Menü hat unterschiedlich viele Optionen je nach dem, ob der angemeldete Benutzer Administratorenrechte hat oder nicht. (Siehe Abbildung 5.2 auf der Seite 29 für die Optionen der Lektoren und Abbildung 5.6 auf Seite 32 für Optionen der Administratoren.) Die Möglichkeit, sich unter einem anderen Namen anzumelden, sollte jedem Benutzer unabhängig vom Einlog-Status zur Verfügung stehen. Hier ein Codebeispiel für den „Anmelde“-Button:

```

echo "<p>";
echo"<a href=\"\$PHP_SELF?aktion=neu_anmelden\">
    <img border=\"0\" alt=\"[neu anmelden]\"
        src=\"bt_neu_anmeld.gif\" /></a><br />";
echo "</p>";

```

Im folgenden Codeausschnitt wird geprüft, ob der Benutzer Administratorstatus hat. Falls ja, werden die Buttons für die Optionen des Administrators freigegeben.

```

if ($SESSION["admin"]=="Ja") {
    [HTML-Code für die Buttons der Benutzer mit Administrationsrechten]
}

```

Die generellen Funktionen sollen jedem freigeschaltet werden, der einen gültigen Login besitzt:

```

if ($SESSION["login"]<>'') {
    [HTML-Code für die Buttons aller eingeloggten Benutzer]
}

```

Die Funktionsmodule

Funktionsmodule sind Codeblöcke, die eine bestimmte Funktion (z.B. Anmeldebildschirm anzeigen, Anmeldung prüfen, ...) ausführen. Die Variable `$aktion` legt fest, welches Modul abgearbeitet wird. Dabei sind die einzelnen Module nach einem ähnlichen Schema aufgebaut. Nach der Ausgabe der Modulüberschrift

```

echo "<h2>Modul-Überschrift</h2>";

```

erfolgt meistens eine Datenbankabfrage. Die SQL-Datenbankabfrage wird zunächst in der Variablen `$query` gebildet. Die Variable kann man während der Programm-entwicklung zusätzlich zu der Datenbankabfrage auf dem Bildschirm ausgeben. So kann man eventuelle Fehlerquellen leicht entdecken und korrigieren.

```
$query = "SELECT *
        FROM seiten
        WHERE gebiet = $gebiet
        ORDER BY seitenueberschr";
```

Die Abfrage erfolgt dann mit folgender PHP-Befehlsabfolge:

```
mysql_connect("localhost", "bvs", "11bvs2");
mysql_select_db(bvs);
$seitenresult = mysql_query($seitenquery);
```

Danach wird überprüft, ob die Abfrage erfolgreich war:

```
if ($seitenresult != 0) {
    $seitennum = mysql_num_rows($seitenresult);
}
```

In diesem Beispiel wird bei erfolgreicher Abfrage die Anzahl der zurückgegebenen Datensätze ermittelt.

Tabelle 6.9 enthält eine Auflistung aller in der Datei `bvs.php` verwendeten Module.

| <i>Modulname</i> | <i>Funktion</i> |
|--------------------------------------|---|
| (leer) | Anmeldebildschirm wird ausgegeben. |
| <code>neu_anmelden</code> | Datenbankabfrage der Anmeldeprozedur |
| <code>hilfe</code> | Das Hilfethema der Variablen <code>\$hilfethema</code> wird ausgegeben. |
| <code>gebiet_last_modified</code> | Last-Modified Liste für den Administrator |
| <code>list_my_pages</code> | Auflistung der Lektoratsgebiete und Seiten des Lektors |
| <code>gebiet_browsen</code> | Verwaltungsübersicht über die Lektoratsgebiete |
| <code>gebiet_hinzufuegen</code> | Eingabemaske um ein neues Lektoratsgebiet hinzuzufügen. |
| <code>gebiet_hinzufuegen_doit</code> | Datenbankeintrag des neuen Lektoratsgebietes |
| <code>gebiet_bearbeiten</code> | Änderungsmaske |
| <code>gebiet_bearbeiten_doit</code> | Update in der Datenbank |

| <i>Modulname</i> | <i>Funktion</i> |
|-----------------------------|---|
| gebiet_loeschen | Sicherheitsabfrage, ob das ausgewählte Lektoratsgebiet wirklich gelöscht werden soll. |
| gebiet_loe- schen_doit | Löschen des Datensatzes |
| user_browsen | Verwaltungsübersicht über die Benutzer |
| user_hinzufuegen | Eingabemaske, um einen neuen Benutzer hinzuzufügen. |
| user_hinzu- fuegen_doit | Neuer Eintrag in die Datenbank |
| user_aendern | Änderungsmaske |
| user_aendern_doit | Update der Datenbank |
| user_loeschen | Sicherheitsabfrage |
| user_loeschen_doit | Löschen des Datensatzes |
| rechte_browsen | Benutzer auswählen, um Lektoratsgebiete zuzuteilen. |
| recht_bearbeiten | Zuteilung der Lektoratsgebiete |
| recht_vergeben_doit | Update der Datenbank |
| recht_entzieh- en_doit | Update der Datenbank |
| seite_browsen | Verwaltungsübersicht über die Seiten |
| seite_hinzufuegen | Eingabemaske für neue Seiten |
| seite_hinzu- fuegen_doit | Abfrage des Seitenlayouts |
| seite_layout_doit | Eintrag in die Datenbank |
| seite_aendern | Änderungsmaske |
| seite_aendern_doit | Update der Datenbank |
| seite_loeschen | Sicherheitsabfrage |
| seite_loeschen_doit | Löschen des Datensatzes |
| gebiet_waehlen | Auswahl eines anderen Lektoratsgebietes |
| seite_auswaehlen | Auswahl einer anderen Seite |
| tls_view | Bearbeitungsmodus |
| topic_hinzufuegen | Eingabemaske für neue Unterüberschriften |
| topic_hinzu- fuegen_doit | Eintrag eines neuen Datensatzes |
| topic_bearbeiten | Änderungsformular für Unterüberschriften |
| topic_bearbei- ten_doit | Update der Datenbank |
| topic_loeschen | Sicherheitsabfrage |
| topic_loeschen_doit | Löschen des Datensatzes |

| <i>Modulname</i> | <i>Funktion</i> |
|-----------------------|---------------------------------|
| link_hinzufuegen | Eingabemaske für neue Links |
| link_hinzufuegen_doit | Eintrag eines neuen Datensatzes |
| link_bearbeiten | Änderungsformular für Links |
| link_bearbeiten_doit | Update der Datenbank |
| link_loeschen | Sicherheitsabfrage |
| link_loeschen_doit | Löschen des Datensatzes |

Tabelle 6.9: Module und ihre Funktion (bvs.php)

Abschluss

Es wird der Rest des HTML-Codes ausgegeben. Die noch offenen Tabellen und der HTML-Körper werden geschlossen. Danach ist die Ausgabe beendet.

Auf der Benutzerseite im Browser wartet nun das Programm auf eine Aktion des Benutzers. Auf der Serverseite tickt der Timeout-Zähler. Wenn innerhalb der festgelegten Zeit keine Reaktion des Benutzers erfolgt, wird die Sessionvariable ungültig, und der Benutzer muss sich erneut am System anmelden.

6.2.2 ausgabe.php

Das Programm `ausgabe.php` hat zwei Funktionen:

- Ausgabe für die Homepagebesucher der Stadtbücherei,
- Vorschaufunktion in einem Javaskript-Fenster für die Lektoren.

Zunächst wird überprüft, ob über die URI eine gültige Seiten-ID übergeben wurde. `ausgabe.php` erwartet die Übergabe der Variablen `$sid` mit einer gültigen Seiten-ID. Ohne Seiten-ID beendet sich das Programm mit einem Hinweistext. Die Seiten-ID einer Seite kann der Administrator über den Button „Seiten verwalten“ erfahren. War die Überprüfung der Seiten-ID erfolgreich, werden die Daten der Seite sowie die zusätzlich benötigten HTML-Daten eingelesen. Diese Daten werden in einer vorgegebenen Reihenfolge ausgegeben, sodass dabei ein funktionierender HTML-Code herauskommt. Die Ausgabereihenfolge ist im Programm festgelegt und kann nicht über ein Formular geändert werden.

Weiter werden die Unterüberschriften und die dazugehörigen Links eingelesen. Wurde vom Administrator bestimmt, dass eine Seite keine Unterüberschrift haben soll, wird intern die Bezeichnung „kein Topic“ vergeben. Kommt diese Bezeichnung in einer Seite vor, wird die Unterüberschrift nicht ausgegeben.

```

if (mysql_result($result_topics,$i,"beschr")!="kein Topic") {
    echo "<a name=\"";
    echo mysql_result($result_topics,$i,"tid");
    echo "\">";
    echo mysql_result($result_topics,$i,"beschr");
    echo "</a>";
}

```

Die Eintragung „kein Topic“ ist dennoch notwendig, um die Links zu gruppieren — denn diese werden anhand der Unterüberschriften-ID (`$tid`) ausgelesen und alphabetisch sortiert ausgegeben.

```

SELECT *
FROM links
WHERE topic=$tid
ORDER BY ueberschr

```

Am Ende des Programms wird überprüft, ob die Variable `$anzeige` gesetzt ist. Falls ja, wurde das Programm innerhalb von `bvs.php` als Vorschaufunktion im separaten Fenster aufgerufen. Es wird ein Button angezeigt mit der Aufschrift „Fenster schließen“. Dieser Button ist mit Javaskript hinterlegt, der das Fenster schließt.

```

if (isset($anzeige)) {
    echo "<tr><td colspan=\"2\" align=center>
        <br />
        <br />
        <center>
            <a href=\"#\" onclick=\"javascript:self.close();\">
                <img src=\"pics_bvs/bt_vorschau_schliessen.gif\"
                    alt=\"[Fenster schliessen]\" />
            </a>
        </center>
    </td></tr>";
}

```

6.3 Zur Entwicklung verwendete Software

Es wurde folgende Software zur Entwicklung des Systems benutzt:

Textverarbeitung: *gvim Version 6.1*¹ — Dieser Texteditor bietet neben zahlrei-

¹Siehe auch: <http://www.vim.org/about.php> — Letzter Zugriff 27.08.2002

chen Funktionen ein Syntax-Highlighting für PHP und HTML, mit dem das Bearbeiten des Quellcodes erheblich erleichtert wird.

Datenbankverwaltung: *phpMyAdmin* *Versin 2.2.3*² — Mit diesem Tool kann man die MySQL-Datenbank komfortabel mittels einer PHP-Weboberfläche administrieren.

Grafikbearbeitung: *gimp* *Version 1.2*³ — Ein Grafikprogramm, mit dem ich die Grafiken für die Tabellen sowie die Buttons und die Überschrift erstellt habe.

Browser: Zum Testen des Programms habe ich die gängigsten, javafähigen Browser benutzt:

- Netscape Communicator Version 4.8, 6.02, 7.0 RC1,
- Microsoft Internet Explorer 6.0,
- Mozilla 1.02 (Open-Source Projekt),
- Opera 5.0,
- w3m *Version 0.3*⁴ (Textorientierter Webbrowser, Open-Source Projekt, nicht javafähig) — Beispielausgabe siehe Abbildung 5.4 auf Seite 31.

Server: Auf dem mit Linux betriebenen Server läuft der Webserver Apache⁵ in der Version 1.3.x. Die Skriptsprache PHP⁶ ist durch ein Modul in den Webserver eingebunden. PHP liegt in der Version 4.x vor. MySQL⁷ liegt auf dem Server in der Version 3.23.48 vor.

²Siehe auch: <http://www.phpmyadmin.net> — Letzter Zugriff: 28.08.2002

³Siehe auch: http://www.gimp.org/the_gimp_about.html — Letzter Zugriff: 28.08.2002

⁴Siehe auch: <http://w3m.sourceforge.net/> — Letzter Zugriff: 19.08.2002

⁵Siehe auch: <http://www.apache.org/> — Letzter Zugriff: 28.08.2002

⁶Siehe auch: <http://www.php.net/> — Letzter Zugriff: 28.08.2002

⁷Siehe auch: <http://www.mysql.com/> — Letzter Zugriff: 28.08.2002

Kapitel 7

Resümee und Ausblick

In der vorliegenden Arbeit wurde die Konzeption und Realisierung eines webbasierten Bookmarkverwaltungssystems erarbeitet. Dieses System ist speziell nach den Anforderungen der Stadtbücherei Stuttgart entwickelt worden. Anpassungen an andere Institutionen bedürfen minimaler Änderungen. In einem Ergänzungsschritt ist es leicht möglich, Layoutanpassungen für den Administrator zu erlauben.

Das BVS wird über HTML-Formulare per Webbrowser bedient. Die Formulare werden serverseitig mittels PHP generiert. Über eine Anmeldung wird der Benutzerstatus überprüft, der den Lektoren die entsprechenden Lektoratsgebiete freigibt.

Der Administrator hat einen besonderen Benutzerstatus. Er hat zusätzlich zu den Möglichkeiten eines Lektors die Benutzerverwaltung und das Anlegen der Lektoratsgebiete sowie der zugehörigen Seiten zu machen. Die Lektoren können innerhalb der ihnen zugewiesenen Seiten die Unterüberschriften sowie die Links anlegen, ändern oder löschen. Diese Aktionen ziehen entsprechende Änderungen in der MySQL-Datenbank nach sich.

7.1 Nutzen für die Stadtbücherei

Die Stadtbücherei vermeidet durch das BVS erhebliche Doppelarbeit. Dadurch werden Personalressourcen frei und Kosten eingespart.

Die Linkeingabe wurde bisher über zwei Stellen abgewickelt. Das ist in Zukunft nicht mehr notwendig, da nun die Lektoren die Linkeingabe selbständig durchführen können. Die Eingabe wird sofort in die Datenbank eingetragen und ist über die Ausgabefunktion abrufbar. Nachträgliche Korrekturen sind natürlich ebenfalls einfach und schnell durchführbar. Durch mehr Selbständigkeit und Kompetenzgewinn wird eine Stärkung der Mitarbeiter erreicht, die sich auch in einer höheren Arbeitszufriedenheit ausdrückt.

Die Stadtbücherei verfolgt die Idee der Gleichstellung der Medien. Das BVS trägt dazu bei, dass die Links einen gleich hohen Stellenwert bekommen können wie die

übrigen Medien in der Stadtbücherei. Das wird vor allem durch die schnelle, einfache Bedienung und die Kontrollfunktionen erreicht.

7.1.1 Vorteile für die Lektoren

Für die Stadtbücherei ist es ein großes Anliegen, dass nach jedem Bearbeitungsschritt eine Vorschaufunktion zur Verfügung steht. Auf diese Art kann der Lektor bei der Eingabe sehen, was der Websurfer nachher zu Gesicht bekommt. Diese Funktion ermöglicht es auch, eventuelle Änderungen sofort durchzuführen. Das erleichtert die Arbeit erheblich und legt die Verantwortung der Seiteninhalte auf die Lektoren. Durch die gewonnene Selbständigkeit wird bei den Lektoren das Kompetenzgefühl erweitert. Daraus resultiert auch eine höhere Arbeitszufriedenheit.

7.1.2 Vorteile für die EDV-Abteilung

Die EDV-Abteilung hat künftig einen besseren Überblick darüber, wann die Links zuletzt überprüft worden sind bzw. wann zuletzt ein Update der Linkseite vorgenommen wurde. Weiter wird auch hier die EDV-Abteilung entlastet, da die Doppelarbeit der Übertragung der Links in den HTML-Code wegfällt.

7.2 Erweiterungsmöglichkeiten

Interessant wäre der Einbau eines Linkcheckers, der automatisiert täglich die Links auf Erreichbarkeit hin überprüft. Wenn ein Link beispielsweise drei Tage hintereinander nicht erreichbar ist, kann dem Lektor eine automatisch generierte E-Mail mit einem Überprüfungshinweis zugesandt werden.

Eine weitere denkbare Erweiterung wäre ein Contentchecker, der hin und wieder die eingetragene Seite aufsucht und Schlüsselwörter der Linkbeschreibung mit dem Seiteninhalt abgleicht. So könnte überprüft werden, ob sich vom Inhalt her Wesentliches an der Seite geändert hat.

7.3 Ausblick

Denkbar wäre auch eine Aufteilung der Lektoratsgebiete auf mehrere Bibliotheken. Durch die plattformunabhängige, webbasierte Programmierung ist es möglich, ohne große Änderungen am Programm eine fehlerfreie Benutzung von anderen Standorten aus zu gewährleisten. Lediglich die Authentifizierung sowie die Anmeldung an das System müssten überarbeitet werden.

Denkbar wäre auch, im Sinne des „Open Source“-Gedankens, das System anderen Bibliotheken zur Verfügung zu stellen.

Literaturverzeichnis

- [1] AUER, Johannes. *Lernen mit freier Information und freier Software. Positionspapier*. Unveröffentlicht
- [2] Bertelsmann Stiftung: *Projektinformation Deutsche Internetbibliothek*. – URL <http://www.bertelsmann-stiftung.de/documents/Projektinformation.pdf> — Letzter Zugriff: 24.09.2002
- [3] CRAMER, Florian. *Vom freien Gebrauch von Nullen und Einsen : Open Content und Freie Software*. URL http://www.stuttgart.de/stadtbuecherei/druck/oc/cramer_opencontent.htm — Letzter Zugriff: 25.09.2002. Mai 2002
- [4] Feig und Partner: *Glossar : Content Management System*. – URL http://www.contentmanager.de/ressourcen/glossar_10_content_management_system.html — Letzter Zugriff: 05.09.2002
- [5] KRAUSE, Jörg: *PHP 4 – die Referenz : [aktuell zu PHP 4.0.5]*. München [u.a.] : Hanser-Verlag, 2001. – 619 S. – ISBN 3-446-21687-1
- [6] KRAUSE, Jörg: *PHP 4 : Grundlagen und Profiwissen*. 2., überarb. Aufl. München [u.a.] : Hanser-Verlag, 2001. – 1125 S. – ISBN 3-446-21847-5
- [7] LERDORF, Rasmus: *PHP : Kurz und gut*. Köln ; Paris [u.a.] : O'Reilly Verlag, Mai 2000. – 112 S. – ISBN 3-89721-225-0
- [8] NIEDERST, Jeniffer: *HTML : Kurz und gut*. 2. Aufl. Köln ; Paris [u.a.] : O'Reilly Verlag, Mai 2002. – 104 S. – ISBN 3-89721-243-9
- [9] PRÜM, David: *Entwicklung eines webbasierten Dokumentenarchivs zur Verwaltung und Veröffentlichung von Lehrmaterialien auf einem LAMP-System*. Stuttgart, Fachhochschule Stuttgart—Hochschule für Bibliotheks- und Informationswesen, Diplomarbeit, November 2000. – 45 S
- [10] Resource Discovery Network Centre: *[Portaleingangseite]*. – URL <http://rdnet.ac.uk/> — Letzter Zugriff: 27.09.2002

-
- [11] Staats- und Universitätsbibliothek Bremen: *Elektronische Bibliothek*. – URL <http://elib.suub.uni-bremen.de/FramesetElib.html> — Letzter Zugriff: 23.09.2002
- [12] Stadtbibliothek Bremen: *Internet in Öffentlichen Bibliotheken. Strukturierungsbedarf und -möglichkeit beim Online-Zugang zu Information und Wissen. 1996-1998*. – URL http://www.stadtbibliothek-bremen.de/portrait/portrait_bl.php3?portrait=projekte — Letzter Zugriff: 27.09.2002
- [13] Stadtbibliothek Bremen: *Projekt BINE*. – URL <http://www.stadtbibliothek-bremen.de/bine/start.html> — Letzter Zugriff: 26.09.2002
- [14] Stadtbücherei Stuttgart: *Open Content in der Stadtbücherei Stuttgart*. – URL <http://www.stuttgart.de/stadtbuecherei/druck/opencontent.htm> — Letzter Zugriff: 23.09.2002
- [15] Universitätsbibliothek Braunschweig: *What is CoOl?*. – URL <http://www.biblio.tu-bs.de/CoOL/whatcool.htm> — Letzter Zugriff: 27.09.2002
- [16] Universitätsbibliothek Ulm: *ULF - Ulmer Fachinformation*. – URL http://www.bibliothek.uni-ulm.de/services/ulf/01ulf_systema.html — Letzter Zugriff: 23.09.2002

Beilage

Quellcode

Der Quellcode des BVS (bvs.php und ausgabe.php) sowie die Grafiken sind neben den SQL-Datendefinitionen mit einigen Beispieldatensätzen auf dem beigefügten Datenträger enthalten.

Bitte beachten Sie die readme-Datei auf dem Datenträger.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Stuttgart, 11. Oktober 2002

Klaus Köhler