Eberhard Karls
Universität
Tübingen

Fakultät für Informatik

Webbasierte Verwaltungssoftware für einen Internet-Dienstleister

Eine LAMP-Implementierung

Diplomarbeit

Sven Dittmar, 30. Oktober 2000

Betreuer: Prof. Dr. Herbert Klaeren
Zweitkorrektor: Prof. Dr.-Ing. Wolfgang Straßer

Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit selbständig verfasst zu haben.

Es wurden keine anderen als die angebenen Quellen benutzt.

Danksagung

Das Korrekturlesen übernahmen Guido Burger und Daniela Gehle.

Diese Diplomarbeit wurde mit folgender Software (Open Source) erstellt:

Textsatz: LaTeX2e (Doc Class `scrbook`),
Editor: `vi`, Screenshots: `gimp`, Diagramme: `dia`

Inhaltsverzeichnis

1. Einführung

1.1. Motivation und Ziel

Der Verein BaWue-Net e.V. bietet seit 1994 als Internet-Dienstleister Zugang zum Internet und weitere Internet-Dienste, wie z.B. Homepages, Domainhosting, Email-Adressen, Mailinglisten, an.

Wer ein Nutzer des Bawue-Net e.V. werden möchte, kann sich online anmelden. Es gibt derzeit drei Nutzungsmodelle (IP, UUCP und Shell-Account). Das Zustandekommen eines solchen Vertrages erlaubt dann die Nutzung aller im Verein angebotenen Dienste. Details hierzu siehe Kapitel 4.1. Zusätzlich ist für Nutzer eine Mitgliedschaft im Verein möglich. Die Admininistratoren des Vereins sind besonders aktive Vereinsmitglieder, die sich ehrenamtlich um die Installation und Wartung der Vereinshardware und Nutzersupport kümmern.

Die Verwaltung der Nutzerdaten erfolgte bisher in einer proprietären Datenbank. Die finanzielle Verwaltung des Vereins geschah jedoch mit einem anderen Softwareprodukt. Die Folge waren häufige Synchronisationsprobleme und schlechte Erweiterbarkeit, da die Datenbankschnittstelle nur sehr unzureichend dokumentiert wurde. Außerdem war der Verein bisher dezentral organisiert - die einzelnen Einwahlpunkte waren autark - was zu weiteren Schwierigkeiten bei der Erfassung und Aktualisierung der Benutzerdaten führte.

Eine starke Veränderung des Internet-Marktes im Zeitraum 1998/99 (Preisverfall und sinkende Benutzerzahlen) erforderte eine Neuausrichtung und Zentralisierung des Vereins und seiner Einwahlstruktur. Zugleich zeigte sich eine Abnahme der ehrenamtlichen Helfer und Administratoren.

Zur Lösung dieser Probleme wurde zwingend ein neues Verwaltungskonzept und eine umfassendere Datenbankstruktur und gut dokumentierte Datenbankschnittstelle auf der Grundlage marktgängiger Softwareprodukte notwendig. Da es sich um einen im Internet tätigen Anbieter handelt, liegt eine Implementierung mittels Webunterstützung nahe.

Desweiteren wurde ein webbasiertes Frontend für die Nutzer des Vereins gewünscht, zur Kontrolle und Aktualisierung ihrer persönlichen Daten und Nutzung/Konfiguration der angebotenen Dienste.

Entstanden ist mit dieser Diplomarbeit eine Internet-Applikation, die mittels MySQL, PHP und Perl:

- die Verwaltung der Nutzerdaten des Vereins bewerkstelligt,
- die Administratoren bei Ihrer täglichen Arbeit unterstützt (Benutzerverwaltung, technische Konfiguration),
- dem Schatzmeister Erleichterung im Bereich der Finanzen verschafft (Erzeugung der Bankeinzugsdiskette, Verwaltung und Generierung von Rechnungen),
- den Nutzern des Vereins einen zeitgemäßen Zugang zu Ihren Daten und Emails bietet und
- dem Vorstand Übersicht und Kontrolle über die Technik, Benutzer und Finanzen gibt.

Hinweis: Da es sich um ein Gemeinschaftsprojekt mehrerer Personen handelt, stelle ich hier primär die von mir entwickelte Software dafür vor. Sollte es zum Verständnis aber notwendig sein, so werde ich auch externe, von anderen entwickelte Softwarekomponenten einflechten und entsprechend kennzeichnen.

1.2. Aufbau der Arbeit

Kapitel 2 erläutert die eingesetzen und auf Open Software basierenden Softwareprodukte.

Kapitel 3 gibt einen generellen Überblick zur Entwicklung von Web-Applikationen.

Kapitel 4 geht auf den Datenbank-Entwurf zu den in dieser Arbeit entworfenen Datenmodellen ein.

Kapitel 5 beschäftigt sich mit der Realisierung der Applikationen in den Bereichen Administration, Automatismen und Benutzer-Frontend.

Kapitel 6 bespricht softwaretechnische Aspekte (Projektverlauf, Verwaltung des Quellcodes und Benutzer-Umfragen zur Applikation).

Kapitel 7 schließt die Arbeit mit einer Schlussbemerkung und einem Ausblick.

Das Literaturverzeichnis ist zweigeteilt in Druckwerke und Online-Referenzen. Bei letzteren muss leider davon ausgegangen werden, dass einige davon nach Drucklegung nicht mehr unter der angegebenen URL ansprechbar sein könnten.

Der Anhang enthält ein Glossar, die Fragebögen der Umfrage, die alten MySQL-Lizenzbestimmungen, detaillierte Programm-Beispiele sowie das Gesamt-Datenmodell.

1.3. Darstellungsweisen und Abkürzungen

Da sich der Begriff „Web“ als Kurzform für das World Wide Web (WWW) eingebürgert hat, verwende ich auch diesen kürzeren Begriff.

URLs (Links) und Codefragmente werden wie folgt dargestellt:

```
http://www.bawue.de/
```

Abkürzungen

BSD Berkeley Software Distribution

CD Compact Disk

CGI Common Gateway Interface

CVS Concurrent Versions System

DAEMON Disk And Execution MONitor

DBI Database Interface

DENIC deutsches Netzwork Information Center

DES Data Encryption Standard

DHTML Dynamic Hypertext Markup Language

DVD Digital Versatile Disk

EMail Electronic Mail

FAQ Frequently Asked Questions

FTP File Transfer Protocol

GNU GNU is not Unix

GUI Graphical User Interface

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

ISO/OSI International Org. for Standardization/Open Systems Interconnection

IP Internet Protocol

JDBC Java Database Connectivity

LAMP Linux-Apache-MySQL-PHP/Perl

NCSA National Center for Supercomputing Applications

ODBC Open Database Connectivity

Perl Practical Extraction and Report Language

PGP Pretty Good Privacy

PHP PHP: Hypertext Preprocessor

RCS Revision Control System

RSA Rivest-Shamir-Adleman

RIPE Réseaux IP Européens

S-HTTP Secure Hypertext Transfer Protocol

SGML Standard Generalized Markup Language

SQL Structured Query Language

SSL Secure Socket Layer

TCP/IP Transmission Control Protocol/Internet Protocol

URL Uniform Resource Locator

URI Uniform Resource Identifier

WWW World Wide Web

XML eXtended Markup Language

2. Eingesetzte Software

Zum Einsatz kommt ein sogenanntes LAMP-System (Linux-Apache-MySQL-Perl/PHP), welches die Grundlage für alle Applikationen bildet.

Ein LAMP-System kommt der limitierten finanziellen Ausstattung des Vereins sehr entgegen, da keine Lizenzkosten entstehen und trotzdem ausgereifte und stabile Softwarekomponenten zum Einsatz kommen. Weitere Gründe für den Einsatz von Linux und dessen Komponenten sind die hohe Zuverlässigkeit, geringe Hardwareanforderungen, die gute Fernwartbarkeit, die verfügbare System-Dokumentation und das große Angebot an zusätzlicher freier Software.

Die technischen Randbedingungen des Vereins sind beschränkt: Es stehen zwei ältere Vereinsserver zur Verfügung, der eine davon dient als zentraler Diensteserver ohne Benutzerlogins (Netzwerkdienste, Datenbank, Emailempfang, Vereinshomepage), der andere hält Logins zum Emaillesen/-verschicken, Homepages etc. für die Benutzer und Vereinsmitglieder vor. Es musste also eine Software gefunden werden, die mit diesen Limitationen zurecht kommt.

2.1. Open Source

Das Schlagwort „Open Source“ ist derzeit in aller Munde, dabei ist die Idee dahinter eigentlich schon so alt wie die Softwareindustrie an sich. Der Begriff „Open Source“ ist ein Oberbegriff für Softwarelizenzen, die „Freiheit im Sinne der Wissenschaft verkörpern“ [SnoMue2000], darunter fallen zum Beispiel:

- Public Domain
- BSD-License
- GPL, GNU Public License (siehe Anhang C.1)

Vorzüge des Open Source-Modells:

- Möglichkeit, den Quellcode zu lesen oder sogar zu verändern
- Veränderungen wiederum weiterzugeben

Die hier entwickelte Software schließt sich dem Modell der GPL an, welches folgende Punkte beinhaltet:

1. freie Weiterverbreitung des Produktes
2. Quellcode ist eingeschlossen
3. Produkt basiert selbst auf freier Software

Die Open Source-Bewegung präsentiert sich auf einer eigenen Homepage im Web unter `http://www.opensource.org/`.

2.2. LAMP

Zum Einsatz kommt ein System, das unter dem Namen LAMP bekannt geworden ist. Die einzelnen Bestandteile Linux, Apache, MySQL, PHP und Perl werden im folgenden näher erläutert.

2.2.1. Betriebssystem Linux

Linux ist eine Open Source Variante von UNIX für PCs und andere Hardware-Plattformen, die zusätzlich zum Betriebssystemkern (Kernel) Werkzeuge und Befehle aus anderen Open Source-Projekten enthält (GNU, Berkeley UNIX).

Linux wurde 1991 von Linus Torvalds an der Universität von Helsinki in Finnland als Erweiterung des „kleinen“ Unix-Systems Minix von Andrew S. Tanenbaum entworfen. Der erste offizielle, stabile Kernel wurde 1994 im Internet freigegeben, derzeit (Herbst 2000) wird am Kernel 2.4 gearbeitet [linux.org].

Herausragende Eigenschaften sind Stabilität, Sicherheit, Fernwartbarkeit und die offene, bis ins Detail dokumentierte Architektur im Sinne von Open Source.

Linux wird über sogenannte Distributionen vertrieben. Der aktuelle Kernel ist zwar einfach über das Internet herunterzuladen, dies macht in der Regel alleine jedoch keinen Sinn [Siever1999]. Linux-Distributoren (SuSE, RedHat, etc.) vertreiben daher komplette Linux-Systeme auf Datenträgern (CD, DVD), die Kernel, Filesystem, Installationsroutinen, Standardwerkzeuge und Officeapplikationen beinhalten. Auch zugehöriger Support wird inzwischen kommerziell angeboten.

Linux läuft auf sehr vielen Hardware-Architekturen (INTEL, DEC Alpha, SUN Sparc, m68k). In dieser Diplomarbeit wird RedHat-Linux 5.2 auf der x86-Architektur von INTEL eingesetzt.

2.2.2. Webserver Apache

Der Apache Webserver ging aus dem NCSA httpd 1.3 hervor und wurde im April 1995 freigegeben. Laut Netcraft Studie war er schon am 4. April 1996 der populärste Webserver (`http://www.netcraft.com/survey`) und steht immer noch an der Spitze.

Der Apache Webserver ist ebenfalls der Open Source Bewegung zuzuordnen, hat eine eigene Apache-Lizenz und wird von der Apache Group (> 30 Personen) betreut, die

auch an anderen Webserver-Projekten arbeiten (XML, Java) [Apache].

Andere Open Source-Webserver sind

- Jigsaw, W3C Open Source Project (`http://www.w3.org/Jigsaw/`)
- Roxen Challenge Webserver, GPL (`http://www.roxen.com/`)
- thttpd - tiny/turbo/throttling HTTP server, (`http://www.acme.com/software/thttpd/`)
- Boa Webserver, GPL (`http://www.boa.org/`)
- Mathop (`http://mathop.diva.nl/`)
- AWKhttpd, GPLL (`http://awk.geht.net:81/`)

Ein Vergleich dieser und anderer Webserver findet sich zum Beispiel unter `http://www.acme.com/software/thttpd/benchmarks.html`

Der Apache Webserver wurde wegen seiner großen Anzahl an ausgereiften Erweiterungsmodulen und seiner großen Verbreitung ausgewählt.

2.2.3. OpenSSL

Da die Übertragung der Daten auf Basis von HTTP unverschlüsselt erfolgt, ist es bei sensiblen Daten notwendig, dass diese Daten auf dem Weg zwischen Server und Client verschlüsselt werden. Diese Funktion bietet zum Beispiel das Secure Socket Layer (SSL) von Netscape.

SSL sitzt über TCP/IP zwischen den Schichten vier und sieben des ISO/OSI Schichtenmodells und ist nicht nur zusammen mit HTTP einsetzbar (im Gegensatz zum Protokoll S-HTTP) sondern praktisch mit allen Anwendungsschichtprotokollen, die TCP verwenden (siehe Abbildung 2.1) [LHG2000].

Eine Implementierung von SSL auf Open Source Basis ist OpenSSL (`http://www.openssl.org`), welches auf SSLeay basiert, das von Eric A. Young und Tim J. Hudson entwickelt wurde. Open SSL ist ebenso wie der Apache Open Source.

Beispiel für den Ablauf eines Schlüsselaustauschs per SSL (der Client kann hier ein Browser sein, muss es jedoch nicht) [SaferNet1998]:

1. Client-Hello: Client sendet Nachricht an Server, wobei die vom Client verwendbaren Kryptoverfahren angeben werden.
2. Server-Hello: Server sendet das ausgesuchte Kryptoverfahren und sein Server-Zertifikat, welches den öffentlichen Schlüssel des Servers („Public Key“, PK)

enthält, an den Client.
Mit diesem PK überprüft der Client die Signatur des Server-Zertifikates, also die Echtheit des Servers.

3. Der Client schickt nun sein eigenes Zertifikat (enthält ebenfalls einen PK) und einen zufällig gewählten Sitzungsschlüssel („Session Key“) an den Server, der mit dem PK des Server verschlüsselt wurde, zurück.

4. Ab sofort wird mit diesem Sitzungsschlüssel eine sichere Kommunikation betrieben, wobei dieser nie unverschlüsselt durch das Internet transportiert wurde.

Ein anderes Prokotoll ist S-HTTP, das als Erweiterung zum Anwendungsschicht-Protokoll HTTP realisiert wurde. Es setzt keinen neuen Standard, sondern greift auf Bewährtes zurück (RSA, PGP, DES, etc.). Allerdings hat sich Netscapes SSL-Verfahren 1996 durchgesetzt [SSL].

Die in dieser Diplomarbeit entwickelten Applikationen verwenden aus Sicherheits- und Datenschutzgründen die SSL-Variante des Webservers Apache. Apache enthält in seiner Standard-Version aus politischen Gründen keine SSL-Implementierung, da in einigen Ländern Verschlüsselung unter das Waffenexportverbot und andere Beschränkungen fällt. Dank seiner stark modularen Bauweise gibt es aber fertige SSL-Programmmodule zur Nachinstallation.

Schicht		Beschreibung	Beispiel
7	Anwendung	Schnittstelle zu Programmen	HTTP/FTP
6	Darstellung	Umwandlung Datenformate	SSL
5	Kommunikation	Steuerung Schicht 4	
4	Transport	Verbindung zwischen Sender und Empfänger	TCP/UDP
3	Vermittlung	Routing	IP
2	Sicherung	Prüfsummen, Bitgruppierung	PPP
1	Bitübertragung	physikalische Datenübertragung	ISDN

Abbildung 2.1.: ISO/OSI Schichtenmodell mit Beispiel-Anwendungen

2.2.4. Datenbank MySQL

Die relationale Datenbank MySQL hat durch den Internetboom eine sehr starke Verbreitung erfahren. Das Grunddesign von MySQL basiert auf mSQL [mSQL] und hatte unter anderem die Zielsetzung, im Vergleich zu mSQL schneller zu sein und mehr Befehle anzubieten. MySQL implementiert einen großen Teil des SQL-92-Standards [SQL Reference]. Der Hersteller von MySQL ist die schwedische Firma T.c.X. DataKonsultAB, die erste Version stammt aus dem Jahre 1995 [MySQL].

In Bezug auf ANSI-SQL wurden einige Befehle (noch) nicht implementiert, wie Transaktionen, Locks, Trigger, referentielle Integrität, Stored Procedures oder Views, gleichzeitig aber weitergehende Befehle eingeführt (z.B. auto_increment, encrypt, vielfältige Datentypen, Umlaut-Unterstützung). Als Grund für das Fehlen dieser Eigenschaften werden hauptsächlich dadurch entstehende Geschwindigkeitseinbußen genannt.

Für große, umfangreiche Projekte scheidet MySQL daher eher aus (das Lesen von Datensätzen geht sehr schnell, Schreibvorgänge aber langsam). Die Stärke von MySQL liegt dagegen im breiten Angebot von Schnittstellen an Programmiersprachen wie Perl, PHP, Tcl, Python, C, C++ und auch ODBC bzw. JDBC, die allesamt von freien Entwicklern stammen.
Diese Angaben beziehen sich auf MySQL Versionen 3.22.x. Es kann damit gerechnet werden, dass künftige Versionen von MySQL mehr ANSI-SQL-Befehle verarbeiten können. Am Ende dieser Arbeit wurde eine Version von MySQL angekündigt, die Transaktionen beherrschen soll, weitere Informationen unter `http://www.maxsql.com/`.

Die Lizenzbestimmungen für MySQL haben sich im Laufe dieser Diplomarbeit grundlegend geändert. Seit August 2000 ist sie dem Open Source-Modell zuzuordnen. Das alte Lizenzmodell wird im Anhang C.2 besprochen.

Es werden folgende Plattformen unterstützt

- alle UNIX-Systeme
- Microsoft Win32 Systeme
- IBM OS/2

Das Kommandozeilenwerkzeug „mysql“ ist ein Client und verhält sich wie eine Kommandozeilen-Schnittstelle. Alte Kommandos werden in einer Historie gespeichert und sind so wiederverwendbar.

Weitere Tools wie „mysqladmin“ zur Verwaltung des Servers und „mysqldump“ zur Erzeugung eines Datenbankabbildes in ASCII-Form gedacht (zum Export der Daten beispielsweise für eine Datensicherung) sind ebenfalls sehr hilfreich.

2.2.5. PHP

PHP steht als rekursives Akronym für „PHP: Hypertext Preprecessor“.

Entwickelt wurde PHP 1994 ursprünglich von Rasmus Lerdorf (zuerst als PHP/FI Personal Home Page Tools/Form Interpreter), steht unter der Open Source Lizenz und wird derzeit in zwei Versionen angeboten [PHP]:

- PHP3, Version 3.0.14, aktuelle, stabile Programmversion

- PHP4 bzw. Zend, liegt derzeit noch im Beta-Stadium vor
  (Hier wurde der Kern optimiert, was große Auswirkungen auf die Performance hat).

Für die Weiterentwicklung steht ein ganzes Team von freien Entwicklern im Hintergrund. PHP wurde für die gängigen UNIX- und Microsoft-Windows-Systeme implementiert.

PHP ist eine sogenannte „HTML-embedded scripting language“, also eine in HTML eingebettete Skriptsprache, die vom Web-Server ausgewertet wird, bevor sie zum Web-Client (Browser) geschickt wird. Die Syntax lehnt sich an C, Java und Perl an [Medinets2000].

Eingeleitet wird PHP in HTML durch folgendes XML-konformes Konstrukt:

```
<?php ...?>
```

Auch möglich ist:

```
<?  ...?>
```

Ein kleines Programmbeispiel aus der PHP-Dokumentation „urldecode – Decodes URL-encoded string“ (siehe `http://www.php.net/manual/function.urldecode.php`):

```
Decodes any %## encoding in the given string.
The decoded string is returned.
Example 1. Urldecode()

<?php
  $a = split ('&', $querystring);
  $i = 0;
  while ($i < count ($a))
  {
    $b = split ('=', $a [$i]);
    echo 'Value for parameter ', htmlspecialchars (urldecode ($b [0])),
         ' is ', htmlspecialchars (urldecode ($b [1])), "<BR>";
    $i++;
  }
?>
```

2.2.6. Perl

Perl steht für „Practical Extraction and Report Language“ und verkörpert nach Ansicht des Autors alle Vorteile von C, sed, awk, und sh [WaSch1992].

Der Autor Larry Wall hat Perl als Programmiersprache konzipiert, die primär zur Manipulation von Text, Dateien und Prozessen dienen soll. Zuerst für UNIX-Systeme implementiert, gibt es Perl inzwischen auch für andere Betriebssysteme, wie z.B.

Microsoft Win32 und Macintosh.

Die Lizenz ist Open Source. Die erste öffentliche Version 1.000 stammt vom 18. Dezember 1987.

Die Einsatzgebiete von Perl reichen von anfangs System-Administration bis hin zur heutigen Anwendung als CGI (Common Gateway Interface) im Web, vor allem seit sehr viele freie Programmierer Module für allerlei Zwecke (DBI, CGI) entwickelt haben [Perl].

Zum Einsatz kommt Perl in der derzeit aktuellen Version 5.005_03.

2.3. Sonstiges

Da von Javascript Gebrauch gemacht wurde und für die Navigation ein bereits fertiges Softwareprodukt eingesetzt wurde, findet sich hier jeweils eine kurze Einführung.

2.3.1. Javascript

Javascript wurde 1995 von Netscape mit dem Navigator 2 eingeführt und ist eine HTML-embedded Skriptsprache (d.h. sie wird vom Browser ausgeführt und es besteht keine direkte Verbindung zum Server). Sie lehnt sich in ihrer Syntax an Java an.

Als Microsoft seinerseits eine Version von Javascript (genannt JScript) in ihren Internet-Explorer einbaute, begannen allerdings die bis heute andauernden Probleme.

Die Versionen von Netscape und Microsoft sind in ihrem zugrundeliegende Objektmodell unterschiedlich und daher ist der entsprechende Code zunächst einmal nicht kompatibel, was wiederum bedeutet, dass der Entwickler stets mehrere Versionen seines Javascript-Programmes schreiben und mit verschiedenen Browsern testen muss.

Um die Problematik von Javascript zu entschärfen, gibt es Standardisierungsbemühungen des W3C, die unter dem Stichwort ECMA auf [w3c] zu finden sind.

Ein Entwickler muss sich heutzutage also mit mindestens zwei verschiedenen Versionen von Javascript beschäftigen, wenn er dieses in seiner Applikation verwenden möchte. Ein Grund, ein Template für die Navigation zu verwenden, das mit den alltäglichen Widrigkeiten von Javascript und Browser zurecht kommt (siehe Kapitel 2.3.2).

Beliebte Anwendungsgebiete von Javascript [KueMi2000]:

- client-seitige Überprüfung von Einträgen in Formularen
- Bildvertauschungen (mittels mouse-over)
- dynamisches HTML (DHTML)
- Lauftexte

Aufgrund der Inkompatibilitäten von Netscape und Microsoft und der Tatsache, dass einige Browser Javascript überhaupt nicht unterstützen, Javascript oft im Zusammenhang mit sicherheitsrelevanten Problemen erwähnt wird und Benutzer Javascript daher in ihrem Browser abgeschaltet haben, ist eine Verwendung von Javascript im Einzelfall zu prüfen und sollte generell eher nicht angewandt werden. In dieser Diplomarbeit wird Javascript nur im `easy.bawue` und nicht im `my.bawue` eingesetzt.

Javascript	Netscape Navigator	Internet Explorer
1.0	2	
1.1	3	3
1.2	4 nicht ECMA-262 konform	4 erfüllt ECMA-262

Abbildung 2.2.: Übersicht Javascript Versionen

2.3.2. Navigation: Joust Outliner

Joust ist ein Web-Navigations-System, das Javascript (DHTML) und Cookies (für die Umschaltung des Darstellungs-Modus) verwendet. Es ist Freeware, der Quellcode ist verfügbar und entsprechend anpassbar. Die jüngste und hier eingesetzte Version ist die 2.4.1 vom 5. Juli 1999.

Das Joust-Paket besteht aus einer verhältnismäßig großen Hauptdatei (Größe ca. 30 KB), zwei weiteren Index-Dateien und verschiedenen, kleinen GIF-Bildern für die Navigation.
Joust bietet browser- und betriebssystem-spezifische Icons und verschiedene Darstellungs-Modi (Frame, Floating und Non-Frame).

Der Floating-Mode entspricht einem eigenen, kleinen Browser-Fenster vom Format einer Fernbedienung (Vorteil: volle Breite im Daten-Frame nutzbar). Non-Frame wird nur insofern unterstützt, als dass ein Link vorbereitet ist. Wollte man diese Funktion verwenden, müsste man diesen Bereich selber ausprogrammieren.

Das Navigations-System besteht aus 4 Teilen:

- Menu-Frame (Navigationleiste)
- Title-Frame (Titelbereich)
- Menu-Control-Frame (Umschalten zwischen verschiedenen Darstellungs-Modi, Expandieren, Zusammenklappen aller Menüpunkte, Links zu Hilfe und Startseite)
- Daten-Frame (enthält die Ausgabe der Applikation)

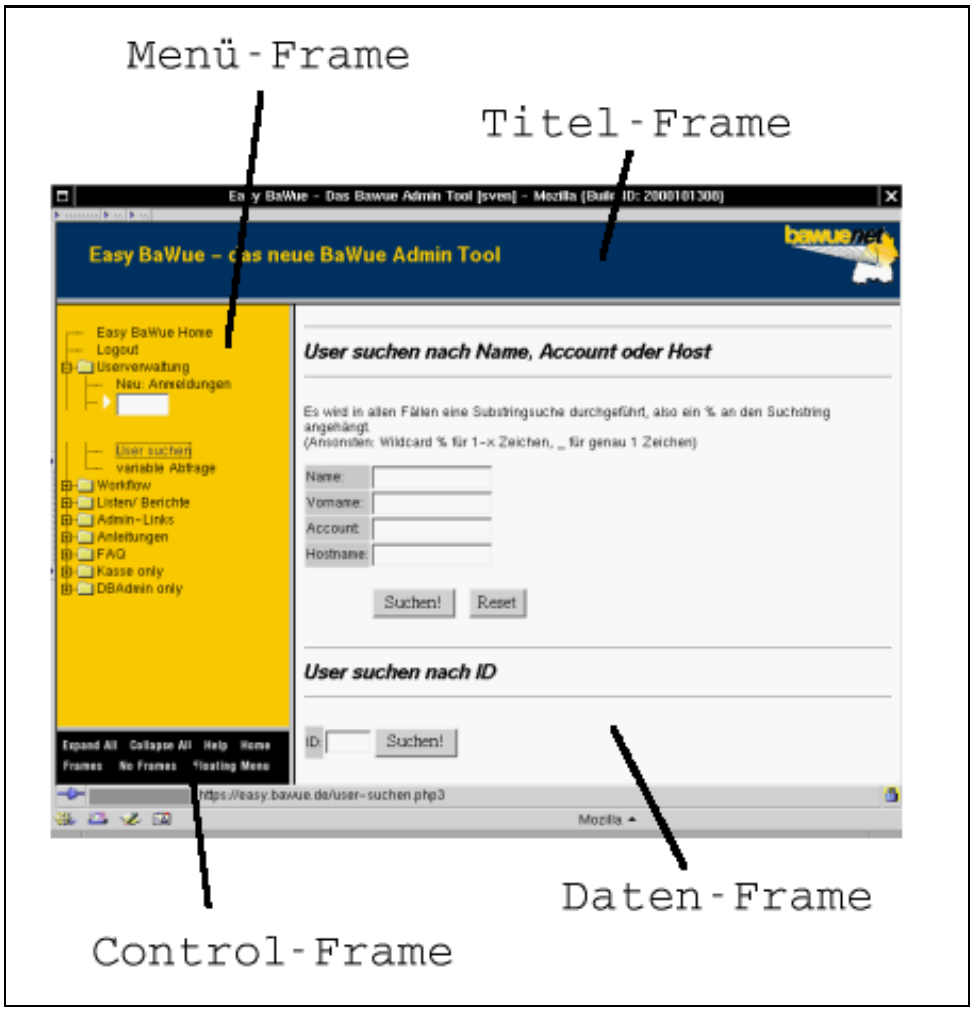

Abbildung 2.3.: Joust Outliner Struktur

Es besteht die Möglichkeit, durch Einbindung eines entsprechenden Javascript-Kommandos im Daten-Frame, alle Webseiten immer zusammen mit der Joust-Navigation zu laden. Die Folge ist: Es sind keine „inneren“ Bookmarks für den Benutzer verfügbar, d.h. Bookmarks, die auf Bereiche innerhalb der Webapplikation zeigen. Dadurch hat der Entwickler eine bessere Kontrolle bei der Veränderung oder Erweiterung der Gesamtapplikation, da die Nutzer dann keine veralteten Bookmarks verwenden [Joust].

In dieser Arbeit wurde der „Navigations-Zwang“ umgesetzt (siehe auch Antworten auf diesen Punkt in der Umfrage aus Kapitel 6.3).

3. Aspekte beim Entwurf einer Webapplikation

Meine Empfehlung für den generellen Aufbau beim Entwurf, Test und der Weiterentwicklung von Webapplikationen sind zwei getrennte Systeme:

- Test und
- Produktion

mit jeweils eigenem Webserver und eigener Datenbank, am besten auf unterschiedlichen Computern, was auch in diesem Projekt durchgeführt wurde. So ist ein von einander unabhängiger und störungsfreier Test von neuen Versionen der beteiligten Komponenten und eingesetzten Software möglich.

3.1. Das Protokoll HTTP und der Webserver

Für die Entwicklung einer Webapplikation muß besondere Sorgfalt auf die Auswahl des Webservers gelegt werden, da dieser die Kommunikation mit dem Webbrowser des späteren Benutzers steuert.

3.1.1. Das HTTP-Protokoll

Das Hypertext Transfer Protocol wurde 1990 mit folgender Zielsetzung bzw. Definition entworfen: „The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.“ [RFC2616] HTTP zählt zur Gruppe der „request/response“-Protokolle.

Die aktuelle Version des HTTP-Protokoll ist 1.1. Hier sind im Vergleich zu 1.0 insbesondere Verbesserungen in Bezug auf Proxying, Caching, persistente Verbindungen und virtuelle Hosts implementiert worden.

Ein Besonderheit des HTTP-Protokolls ist die Zustandslosigkeit („stateless“), d.h. nachdem ein Client eine Anfrage an einen Server geschickt hat, sendet der Server die Antwort und beendet die Verbindung. Eine weitere Anfrage desselben Clients sieht somit aus Sicht des Servers wie eine vollkommen neue Anfrage eines neuen Clients aus. Dies ist inbesondere bei der Entwicklung von Web-Applikationen zu berücksichtigen.

3.1.2. Webserver

Ein Webserver ist ein Programm, welches das HTTP-Protokoll „spricht“ und entsprechend mit Browser kommuniziert.

Bei der Auswahl eines Webservers für dynamische Inhalte sind folgende Merkmale zu berücksichtigen:

- Unterstützung moderner Protokolle (HTTP 1.1)
- Skalierbarkeit (z.B. durch Threads, sogenannte leichte Prozesse)
- möglichst kleiner, schneller Daemon
- Möglichkeit, CGI-Schnittstellen einzurichten (z.B. Perl)
- Möglichkeit, Module (z.B. PHP) in den http-Daemon zu integrieren
- ausführliche Dokumentation der CGI-Schnittstelle

Hierfür eignet sich der Apache Webserver in allen Punkten sehr gut.

3.1.3. Die Standard-Authentifizierung über HTTP

In HTTP 1.0 und 1.1 gibt es nur die sogenannte „Basic Authentication“ [RFC2716] als Sicherheitsmechanismus, die auch in allen Browsern implementiert ist.

Hierbei schickt der Webserver aufgrund seiner Konfiguration für eine bestimmte URI eine Aufforderung zur Login- und Kennworteingabe an den Browser. Dieser öffnet ein Fenster, in dem der Benutzer die Daten eingeben kann. Stimmt das Login, so schickt der Webserver die angeforderte Webseite [SaferNet1998].

Diese Art der „basic authentication“ hat folgende Nachteile [Ratsch2000]:

1. Da die Authentifizierung durch den Browser geschieht, kann man einen Benutzer nicht ohne weiteres durch die Applikation abmelden.
2. Der Verfall eines Logins nach einer bestimmten Zeit der Untätigkeit ist nicht zu erreichen.
3. Gruppen von Benutzern können mit diesem Modell nicht verwaltet werden, dies muss auf Applikationsebene passieren.
4. Der Login-Prozess kann nicht selber gestaltet werden, da der Browser hier immer nur eine modale Dialogbox sendet, insbesondere für neue Benutzer kann hier zum Beispiel keine Hilfestellung gegeben werden, da modale Dialogboxen die anderen Instanzen des Browser blockieren.

Desweiteren ist diese Art der Authentifizierung nicht sicher. Das Kennwort wird bei der Übertragung vom Client zum Server nicht verschlüsselt, es ist also prinzipiell möglich, dieses Kennwort abzuhören.

Weitergehenden Schutz bieten nur kryptografische Protokolle wie SSL und die Realisierung der Benutzer-Login-Verwaltung auf Applikationsebene (zum Beispiel über Sessions).

3.1.4. Andere Methoden der Authentifizierung

Sessions

Eine Session ist ein zeitlich begrenzter, autorisierter Zugang zu einer Internet-Applikation.

Technisch werden Sessions gerne folgendermaßen realisiert:

- Authentifizierung eines Benutzers gegen eine Datenbank.
- Nach Freigabe der Authentifizierung wird eine großer Schlüssel („Session-Key“) generiert und mit einem Verfallsdatum in einer Datenbank gespeichert.
- Der Webclient gibt bei jeder Aktion den Benutzernamen und den Session-Key an die Applikation weiter (meist wird dies über versteckte Felder in Formularen oder Cookies realisiert).
- Nach dem Verfall des Session-Keys fordert die Applikation erneut zur Authentifizierung auf.

Cookies

Cookies stellen Variablen dar, die - vom Webserver initiiert und wieder auslesbar - vom Browser lokal in einer Datei abgespeichert und als eine Art Gedächtnisstütze für Webapplikationen verwendet werden können. Es lassen sich allerdings nur begrenzte Informationen speichern und wieder auslesen.

Durch den Begriff „gläserner Surfer“ sind Cookies in Verruf gekommen, denn durch die Auswertung von Cookies lässt sich das Surfverhalten von Benutzern im Internet analysieren. Für den Programmierer stellt dieser Mechanismus allerdings eine praktische und einfache Variante zur Speicherung von Informationen beim Client dar.

Als alleinige Zugangskontrolle eignen sich Cookies nicht, in Verbindung mit anderen Mechanismen können sie aber einen erleichterten Zugang (z.B. durch Vorbelegung des Loginfeldes) erzeugen.

3.2. HTML und Browser

HTML ist die klassische Sprache zur Seitenerstellung im Web. Es ist eine Auszeichnungssprache, die von SGML abgeleitet wurde.

Die Schlüsselworte von HTML werden „Tags“ genannt, in spitze Klammern eingeschlossen und treten im Allgemeinen paarweise auf. Ein HTML-Dokument wird zum Beispiel eingeschlossen von folgendem Konstrukt:

```
<HTML> Dokumentinhalt </HTML>
```

Leider haben sich auch herstellerspezifische „Tags“ eingeschlichen und machen damit die Browser stellenweise inkompatibel zueinander, was es wiederum dem Entwickler schwerer macht, eine gemeinsame Programmbasis zu finden.

Heutzutage reicht es allerdings nicht mehr aus, nur HTML zu kennen, um Webseiten verstehen bzw. erstellen zu können, da immer häufiger „Mischungen“ mit anderen Sprachen in den HTML-Code einer Webseite integriert werden (Javascript, PHP).

Eine gute Anlaufstelle für Fragen zu HTML ist `http://www.w3.org/MarkUp/`.

Liste der aktuellen Browser

- Netscape Navigator 4
  `http://www.netscape.com/`
- Internet Explorer 5
  `http://www.microsoft.com/`
- Netscape 6 Mozilla
  `http://www.mozilla.org/`
- Opera 4
  `http://www.opera.com/`

Als Fazit für Web-Entwickler gilt: Verwendung minimaler „Tag“-Schnittmenge, ausgiebige Tests mit möglichst vielen, verschiedenen Browsern.

Version	besonderes Merkmal	Jahr
1	erste Version	unbekannt
2	Formulare	1995
3	Tabellen, Textausrichtung	1997
4	allg. Fehlerbereinigung, Internationalisierung Stylesheets	1998

Abbildung 3.1.: Übersicht HTML Versionen
Quelle: `http://www.teamone.de/selfhtml/tbaf.htm`

3.3. CGI - Schnittstelle des Web zu Programmen

Mittels Formularen können im Web Informationen von Benutzerseite aus eingegeben werden. Wie werden diese Informationen aber an eine Applikation übergeben?

Die Antwort lautet: Es gibt zwei Standardmethoden zur Übergabe von Parametern,

- GET, via Standard Input (sichtbar in der URL, praktisch für Debugging-Zwecke, aber auch leicht durch den Benutzer kompromittierbar, weil hier Parameter einfach manuell geändert werden können) und
- POST, via Umgebungsvariablen (unsichtbar in der URL).

Die Parameter werden also an die betroffene Applikation (z.B. in Perl oder PHP) übergeben, die diese weiterverarbeiten kann. Mit sogenannten versteckten (für den Benutzer unsichtbaren) Parametern („hidden fields“) können Werte über mehrere Webseiten hinweg „mitgeschleift“ werden.

Bindet man nun noch eine Datenbank (z.B. MySQL) an die Applikation an, so sind alle Voraussetzungen zur Entwicklung einer dynamischen Webapplikation geschaffen.

3.4. Nutzbarkeit des Systems, Benutzerführung

Die Nutzbarkeit („usability“) für den Benutzer sollte in jedem Fall wohlüberlegt sein, möglichst vor Beginn der Realisierung eines Projektes. Denn wenn der Benutzer des Systems keinen Nutzen aus der entwickelten Applikation ziehen kann bzw. deren Bedienbarkeit zu umständlich ist und nicht auf die Akzeptanz des Benutzers stößt, so hat sich diese selbst ad absurdum geführt.

Die Nutzbarkeit einer Webapplikation bezieht sich auf folgende Frage: Wie einfach ist es, das System

1. zu bedienen?
2. zu verstehen?

Dabei sollte berücksichtigt werden, dass

- HTML keine 100%tige Kontrolle über das Layout zulässt,
  (Das liegt einerseits an HTML selbst, das keine absolute Positionierung kennt, andererseits an den Browsern, die alle ihr eigenes Verständnis von HTML haben, was an vielen Stellen zu unterschiedlichen Darstellungen führen kann.)
- die Benutzer-Interaktion durch die Bandbreite beeinträchtigt werden kann,
  (Dies ist abhängig von der Anbindung des Benutzers - schnelles LAN oder langsame Modemverbindung? Vor allem sollten Entwickler darauf achten, ihr System auch mit einer langsamen Anbindung zu testen.)

- der Zugriff auf die Applikation auch durch externe Bookmarks bzw. direkte Links oder den Linklisten aus Suchmaschinen erschlichen werden kann, d.h. der Entwickler kann nicht davon ausgehen, dass ein Benutzer immer über die Startseite einsteigt,
  (Frames haben die praktische Eigenschaft, den Benutzer mit einem Bookmark durch eine große Website zu führen und vereinfachen auch die Erstellung von gleichartiger Navigation; fortgeschrittene Benutzer setzen aber gerne direkte Bookmarks, außerdem können die meisten Suchmaschinen nicht gut mit Frames umgehen.)

- Benutzer es nicht gewöhnt sind, Anleitungen zu Webapplikationen zu lesen wie zu herkömmlichen Programmen.
  (Besonders im Web entsteht dieses Problem, denn das eigentliche Programm ist der Browser, welcher immer gleich erscheint. Die Ebene der Webapplikation wird hierdurch verschleiert und von Entwicklern wird häufig angenommen, dass der künftige Benutzer schon damit zurecht kommen wird.)

Daher sollten folgende Punkte bei Web-Projektentwicklung eine Rolle spielen:

1. Frühes und direktes Miteinbeziehen der künftigen Benutzer,
   (Der Benutzer erhält hierbei das Gefühl, dass er an der Entwicklung beteiligt ist, die ja etwas schaffen soll, das er später einsetzen wird.)

2. frühe und regelmäßige Bewertung der Applikation,
   (Ein gutes Hilfsmittel, um die Veränderungen zu bewerten, später auszuwerten und zu dokumentieren.)

3. empirische Messungen der Nutzbarkeit auch schon im frühen Stadium,
   (Das Beobachten eines Benutzers bei der Anwendung, Erfahrungsaustausch zwischen Entwickler und Benutzer.)

4. iterative Entwicklung der Applikation.
   (Im Gegensatz zum „großen Wurf“, also ein fertiges Produkt zu übergeben, das der künftige Benutzer noch nie vorher gesehen hat.)

Natürlich sind absolute Messwerte hier mit Vorsicht zu genießen, da jeder Mensch Webapplikationen subjektiv anders bewertet (zum Beispiel durch Unterschiede aufgrund von Erfahrungen mit anderen Webapplikationen oder auch nur aufgrund einer Tageslaune). Trotzdem sollte man starke Tendenzen bei der Messung auswerten und in die Projektentwicklung zurückfließen lassen.

Eine benutzerfreundliche Web-Applikation hat folgende Merkmale:

1. Die Aufgabenstellung kann damit gut gelöst werden:

   - Stellt die Applikation eine korrekte Lösung der Aufgabenstellung dar?
   - Ist die Aufgabe effizient und effektiv gelöst? Muss der Benutzer unnötige Schritte durchführen? Wird nur Information angezeigt, die im Moment notwendig ist? Werden Voreinstellungen (zum Beispiel Datumsvorgaben) getroffen?

2. Sie ist beherrschbar:
   - Der Benutzer versteht sie und kann sie in seinen Funktionen vollständig bedienen.
   - Fehlermeldungen werden genügend lange und verständlich gezeigt.
   - Rückgängigmachen von Funktionen, die aus Versehen betätigt wurden (oder zumindest das Angebot „zurückzukommen“ zur gewünschten Funktion).
3. Sie reagiert entsprechend den Vorstellungen des Benutzers:
   - Funktionen befinden sich an den Stellen, an denen der Benutzer sie vermutet (Konsistenz) und tun auch das, was er von ihnen erwartet (Korrektheit).
   - Systemmeldungen erscheinen immer gleichartig und an der gleichen Stelle.
   - Konsistente Namensgebung der Funktionen und Links.
4. Sie ist maßgeschneidert („personalized“):
   Kulturelle Besonderheiten des Benutzers werden berücksichtigt (Sprache, Zeitangaben, Postleitzahlen, etc.), im Falle eines internationalen Publikums entsprechend verschiedene Angebote für die Benutzer.
5. Sie ist selbsterklärend:
   Eine Web-Applikation ist selbsterklärend, wenn ein neuer Benutzer das System aufgrund der eingebauten Hilfen kennen und verstehen lernen kann.

Aus Kapitel 4, [Ratsch2000].

3.5. Sicherheit

Der Bereich der Sicherheit kommt bei der Entwicklung von Webapplikationen oft zu kurz und wird häufig erst dann ein Thema, wenn schon ein Problem aufgetreten ist.

3.5.1. Vertrauen in die Internet-Dienste

Da es regelmäßig Berichte zu Fehlern in den Internetprotokollen sowie den beteiligten Softwareprodukten gibt, sollte bei der Entwicklung der Applikation davon ausgegangen werden, dass solche Fehler auch ausgenutzt werden können. Dementsprechend muss die eigene Software regelmäßig auf solche Bugs untersucht werden.

Beispiele für bekannte Sicherheitslücken: Speicherung von URLs mit relevanter Information in den Cache-Dateien von Proxy-Servern oder Spionage in lokalen Dateien mittels Ausnutzung von Sicherheitslücken in Microsofts Internet Explorer [MSbug].

Außerdem sollte sich eine Web-Applikation nicht einfach auf eingegebene Daten und deren Konsistenz verlassen, sondern diese soweit wie möglich validieren (weil z.B. Suchmaschinen auf Webformularen den „Submit“-Knopf auslösen und somit leere Anfragen erzeugen).

Ein Beispiel wäre ein Web-Formular mit einem dahintersteckenden Skript, welches die Eingabedaten direkt in eine Datenbank schreibt. Hier sollte die Abarbeitung des Skriptes, welches nicht vom eigenen Server kommt, eingeschränkt werden, da sonst externe, fremde Web-Formulare das eigene System überlasten oder gar lahmlegen könnten.

3.5.2. Einsatz von kryptografischen Protokollen

Es ist abzuwägen, ob die normale Standard-Authentifizierung über HTTP ausreicht (keine Verschlüsselung der Daten auf dem Transportweg, problematisch für sensitive Daten wie Kreditkarten-Informationen) oder evtl. zusätzlich kryptografische Protokolle (z.B. SSL) eingesetzt werden sollen.

Hierbei ist zu berücksichtigen, dass ein SSL-Zertifikat käuflich erworben werden muss und jeweils nur eine bestimmte Zeitspanne gültig ist.

Wird ein Zertifikat selbst erstellt (ohne eine „offizielle“ Zertifizierungsstelle), so wird der Benutzer von seinem Browser davon in Kenntnis gesetzt, dass er ein „unsicheres“ Zertifikat, dessen Vertrauenswürdigkeit nicht geprüft werden kann, verwendet.

3.5.3. Datenschutz im Internet

In Deutschland existiert bis dato kein einheitliches Internet-Datenschutzrecht, auf welches sich ein Internet-Dienstleister stützen könnte. [1]

Folgende vier Gesetze sind in diesem Zusammenhang relevant:

- Telekommunikationsgesetz (TKG) und Telekommunikationsdienstunternehmen-Datenschutzverordnung (TDSV)
- Teledienstedatenschutzgesetz (TDDSG)
- Mediendienstestaatsvertrag (MDStV)
- Bundesdatenschutzgesetz (BDSG)

Ohne im Detail auf die Einzelheiten dieser Gesetze eingehen zu wollen, so kann man zumindest folgende Grundsätze ableiten, die von einem Internet-Dienstleister eingehalten werden sollten:

- Personenbezogene Daten nur mit Einwilligung des Nutzers speichern und nicht an andere Parteien weitergeben,
- nur wirklich notwendige Daten speichern (Prinzip der Zweckbindung und Datenvermeidung),

[1] Diese Informationen stammen aus einem Vortrag von Dr. Ulrike Burscheidt, gehalten am 3. Mai 2000 im „Workshop: Online-Recht“ am Kompetenzzentrum für Multimedia und Telematik (KMMT) in Tübingen

- Nutzerdaten nach Ablauf der Mitgliedschaft löschen, bzw. nur so lange aufbewahren, bis alle Fragen geklärt sind (Beispiel Abrechnungsdaten),
- Nutzungsprofile nur unter Verwendung von Pseudonymen erstellen,
- Auskunftsrecht einhalten (unentgeltlich, elektronisch).

3.5.4. Typische Design-Fehler

Im Laufe dieser Diplomarbeit traten einige typische Fehler zu Tage, die hier kurz skizziert werden.

Voreinstellungen des Webservers

Die Konfiguration eines Webservers sollte vor Inbetriebnahme immer geprüft und nicht einfach so übernommen werden. Beispielsweise müssen die eingesetzten Module/Bibliotheken korrekt installiert sein (PHP, Perl, SSL, Basic Authentication), die Pfade zu den Konfigurationsdateien stimmen und auch gewünschte Seiteneffekte (z.B. Verzeichnisse der Benutzer über die Tilde ~ erreichbar) überprüft werden.

Benutzerrechte der Datenbank

Die Benutzer-Zugriffsrechte auf eine Datenbank sollten immer eingeschränkt sein, bei der Installation von MySQL ist beispielsweise darauf zu achten, dass der privilegierte Benutzer ein möglichst „gutes“ Kennwort erhält und anschließend der Datenbank-Daemon neu gestartet wird.

CGI-Skripte extern aufrufbar

Der Entwickler sollte sich Gedanken darüber machen, ob sein CGI-Programm auch von anderen Servern aus erreichbar ist und ob dies Sinn macht. (Im Falle eines Gästebuch-Systems ist dies sicher gewünscht, im Falle unserer Applikation jedoch nicht.)

Vertrauen in hidden-Fields

Die versteckten Felder in HTML-Formularen sind für einen versierten Benutzer natürlich nicht wirklich unsichtbar, daher sollten hier grundsätzlich keine sicherheitsrelevanten Informationen (beispielsweise Klartext-Kennworte) angegeben werden.

Dateirechte auf dem Server

Sollte der Webserver auch anderen Personen Zugang bieten (was im Falle des `my.bawue` aus Kapitel 5.3 der Fall ist), so muss genau auf die Dateirechte geachtet werden. Beispielsweise sollten Dateien, die das Kennwort zur Datenbank enthalten, nur für den Webserver und Administrator der Applikation lesbar sein.

Debug-Code als HTML-Kommentar

Oft wird bei der Entwicklung der HTML-Kommentar zur Ausgabe von Debugging-Informationen genutzt. Es sollte darauf geachtet werden, dass dieser in der fertiggestellten Applikation wieder entfernt wird bzw. zumindest keine kritischen Informationen mehr enthält.

4. Datenbankentwurf

Für die Applikation werden insgesamt vier Datenbanken eingesetzt, welche hier detailliert erläutert werden. Auf der letzten Seite dieser Diplomarbeit findet sich ein Gesamtüberblick über alle beteiligten Tabellen aller Datenbanken.

Beim Entwurf der Relationen wurde besonders darauf geachtet, das Speichern von redundanten Informationen zu vermeiden und die Datenbanken logisch und physikalisch voreinander zu trennen, um Sicherungskopien der einzelnen Datenbanken einfacher durchführen zu können.

Zudem wurden alle Datenbanken weitestgehend normalisiert [Date1990].

DB-Name	Kommentar
`bawue`	Speicherung der Benutzerdaten und Dienste, Verwaltung Finanzen (Rechnungen), technische Administration (z.B. Mailrouten, Domains)
`bawuelogs`	Logfiles, Statistik
`workflow`	Ticketverwaltung, Anleitungen
`FAQ`	Erstellung und Verwaltung der FAQ

Abbildung 4.1.: Grobübersicht Datenbanken

4.1. Die Datenbank `bawue`

Hier werden die Benutzerdaten, die angebotenen Dienste, die Rechnungen und die technischen Verwaltungsdaten gespeichert.

Folgende Dienste wurden abgebildet:

- IP-Nutzer: Nutzung der Einwahl über TCP/IP und Nutzung sämtlicher anderer Dienste, pauschal DM 27,- pro Monat
- UUCP-Nutzer: Einwahl über UUCP, pauschal DM 20,- im Monat
- Nur-Shell-Account: alle Dienste außer Einwahl, DM 16,- pro Monat
- Zweitnutzer (für Familien oder Personen, die in einem gemeinsamen Haushalt wohnen): weiteres Login auf Vereinsserver (z.B. für separate Mailboxen), kostenlos

- Domainhosting: Hosting einer DE-Domain, Kostenpunkt DM 58,- einmalig für das erste Jahr (inkl. Registrierung oder Umzug einer Domain), jedes weitere Jahr DM 29,-
- Rechnungsstellung: Der Benutzer möchte eine schriftliche Rechnungen bekommen, dieser Dienst ist kostenfrei und wird für den Schatzmeister benötigt.
- Mailingliste: Zuordnung Benutzer zu Mailingliste, dieser Dienst ist kostenlos
- Vereinsmitglied: Mitgliedschaft im Verein, jährlich, Kosten DM 40,-

Alle Dienste enthalten einen Mehrwertsteuerbetrag von 16%, mit Ausnahme der Vereinsmitgliedschaft.

Eine Erweiterung der angebotenen Dienste ist simpel: Ein neuer Eintrag in die Dienstetabelle genügt, um den Dienst nutzbar zu machen.

Die Rechnungen werden in vier Tabellen gespeichert. Jede Rechnung hat einen oder mehrere Rechnungspositionen und außerdem jeweils einen Typ und einen Status. Der Typ einer Rechnung kann sein:

- automatisch, wurde durch das Quartal-Rechnungsskript erzeugt,
- manuell, wurde von Hand durch den Schatzmeister angelegt,
- Guthaben, wurde ebenfalls manuell angelegt und dient zur Verrechnung überbezahlter Beträge.

Der Status einer Rechnung kann sein:

- offen
- bezahlt
- on hold
- storniert

Die technischen Verwaltungsdaten sind folgende:

- Mailrouten
- Mailinglisten
- Domains, Hostnamen
- Datenbank-Rechte
- Hilfesystem

Siehe Abbildungen 4.2, 4.3 und 4.4.

4.2. Die Datenbank bawuelogs

Hier werden alle Zugriffe auf die Webseiten von easy.bawue und my.bawue, sowie die bei den Seitenaufrufen ausgeführten SQL-Statements mitprotokolliert. Diese Zugriffs-Daten umfassen einen Zeitstempel, den Autor, die IP-Adresse und das SQL-Statement. Desweiteren werden hier die Statistik-Daten (Details siehe Kapitel 5.2.5) gespeichert.

Siehe Abbildung 4.5.

4.3. Die Datenbank workflow

Die Daten des Workflows (Tickets, Events, Teams, Anleitungen) werden hier verwaltet.

Die Anleitungen haben einen Typ (eigene Tabelle), einen suchbaren Titel und HTML-fähigen Inhalt. Die Versionierung wird über eine separate Anleitungs-ID bewerkstelligt, außerdem werden die Autoren und Zeitstempel gespeichert.

Der Workflow wird in seiner Funktionsweise in Kapitel 5.1.4 auf Seite 41 besprochen.

Siehe Abbildungen 4.6 und 4.7.

4.4. Die Datenbank FAQ

Speicherung der FAQ-Daten. Die Anleitungsdatenbank hat die identische Datenstruktur wie die der Anleitungen im Workflow.

Siehe Abbildung 4.6.

4.5. Erläuterung der Feldtypen

Es kamen folgende MySQL-Datentypen zum Einsatz; PRI bezeichnet den oder die Primärschlüssel der Tabelle [MySQL].

- `tinyint(x)`: A very small integer. The signed range is -128 to 127. The unsigned range is 0 to 255.
- `smallint(x)`: A small integer. The signed range is -32768 to 32767. The unsigned range is 0 to 65535.
- `int(x)`: A normal-size integer. The signed range is -2147483648 to 2147483647. The unsigned range is 0 to 4294967295.
- `double(x,y)`: A normal-size (double-precision) floating-point number. Cannot be unsigned. Allowable values are -1.7976931348623157E+308 to -2.2250738585072014E-308, 0 and 2.2250738585072014E-308 to 1.7976931348623157E+308.
- `char(M)`: A fixed-length string that is always right-padded with spaces to the specified length when stored. The range of M is 1 to 255 characters. Trailing spaces are removed when the value is retrieved. CHAR values are sorted and compared in case-insensitive fashion unless the BINARY keyword is given.
- `varchar(M)`: A variable-length string. Note: Trailing spaces are removed when the value is stored (this differs from the ANSI SQL specification). The range of M is 1 to 255 characters. VARCHAR values are sorted and compared in case-insensitive fashion unless the BINARY keyword is given.
- `text`: A BLOB or TEXT column with a maximum length of 65535 (2$\hat{1}$6 - 1) characters.
- `datetime`: A date and time combination. The supported range is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. MySQL displays DATETIME values in 'YYYY-MM-DD HH:MM:SS' format, but allows you to assign values to DATETIME columns using either strings or numbers.
- `timestamp(M)`: A timestamp. The range is '1970-01-01 00:00:00' to sometime in the year 2037. MySQL displays TIMESTAMP values in YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD or YYMMDD format, depending on whether M is 14 (or missing), 12, 8 or 6, but allows you to assign values to TIMESTAMP columns using either strings or numbers. A TIMESTAMP column is useful for recording the date and time of an INSERT or UPDATE operation because it is automatically set to the date and time of the most recent operation if you don't give it a value yourself. You can also set it to the current date and time by assigning it a NULL value.
- `year(M)`: A year. The allowable values are 1901 to 2155, and 0000. MySQL displays YEAR values in YYYY format, but allows you to assign values to YEAR columns using either strings or numbers.

Tabellenname	Feldname	Feldtyp	Kommentar
Benutzer	ID	int(5) PRI	eindeutige ID
	Account	varchar(20)	UNIX-Accountname
	Name	varchar(50)	
	Vorname	varchar(50)	
	ErstnutzerID	int(11)	Referenz-ID zu Hauptuser
	Passwort	varchar(50)	Initialkennwort im Klartext
	DESPasswort	varchar(15)	verschlüsseltes Kennwort
	DBRechte	varchar(50)	Zugriffrecht auf Datenbank
	Primaersite	varchar(50)	Name der Site
	TS	timestamp(14)	Zeitstempel Erstellung
	Anrede	varchar(10)	Anschriftenfeld
	Strasse	varchar(50)	Anschriftenfeld
	Organisation	varchar(80)	Anschriftenfeld
	PLZ	int(5)	Anschriftenfeld
	BankKonto	varchar(50)	Bankkonto
	BankBLZ	varchar(50)	Bankleitzahl
	BankOrt	varchar(50)	Ort der Bank
	Zahlungsweise	char(2)	Überweisung/Bankeinzug
	BankKonto	varchar(50)	Bankkonto Inhaber
	InhaberName	varchar(50)	
	BankKonto	varchar(50)	Bankkonto Inhaber
	InhaberVorname	varchar(50)	
Benutzer_Dienst	ID	int(11) PRI	eindeutige ID
	BenutzerID	int(5)	Referenz-ID zum User
	DienstID	int(11)	Referenz-ID zum Dienst
	Datum	datetime	Datum des Eintrages
	Verfall	datetime	Verfallsdatum des Dienstes
	StatusID	int(2)	Status des Dienstes
	TS	timestamp(14)	Zeitstempel der letzten Änderung
	Zusatzkosten	double(16,4)	eventuell anfallend
	LoginVerfall	datetime	Wann verfällt Login?
Dienst	ID	int(11) PRI	eindeutige ID
	Beschreibung	varchar(50)	Beschreibung im Klartext
	Abrechnungsweise	char(1)	Monat/Quartal/Jahr
	Kosten	double(16,4)	Preis
	MwstProzent	tinyint(4)	Steuersatz
Status	ID	int(2) PRI	eindeutige ID
	Beschreibung	varchar(50)	Benennung des Status
Historie	ID	int(11) PRI	eindeutige ID
	BenutzerID	int(5)	Referenz-ID zum User
	Datum	datetime	Datum des Eintrages
	Beschreibung	text	Eintrag
	Autor	varchar(30)	ID des Erstellers
Kontakt	BenutzerID	int(5) PRI	Referenz-ID zum User
	Typ	varchar(50) PRI	Wert aus Kontaktvorschlag
	Adresse	varchar(50) PRI	Datum des Kontakts
	Bemerkung	varchar(50)	Zusatzbemerkung
KontaktVorschlag	Sortierung	int(11) PRI	eindeutige ID
	Name	varchar(50)	Benennung des Kontaktes
	Bemerkung	varchar(255)	Erklärung

Abbildung 4.2.: Datenmodell bawue Benutzer

Tabellenname	Feldname	Feldtyp	Kommentar
Abrechnungsweise	Code	char(1) PRI	Einbuchstabiger Key
	Beschreibung	varchar(50)	Jahr/Quartal/Monat
Bank	ID	int(11) PRI	eindeutige ID
	name	varchar(90)	Name der Bank
	suchName	varchar(30)	Sortierungsfeld
	blz	varchar(10)	Bankleitzahl
	typ	smallint(6)	Banktyp (unbenutzt)
Rechnung	ID	int(11) PRI	eindeutige ID
	BenutzerID	int(5)	Referenz-ID zum Benutzer
	ErstellDatum	datetime	Datum der Erstellung
	LetzteStatusAenderung	datetime	letztes Änderungsdatum
	Status	char(1)	Status (siehe Status-Tabelle)
	Summe	double(16,4)	Gesamtbetrag der Rechnung
	Waehrung	char(3)	Währungseinheit
	Jahr	year(4)	Jahr der Rechnung
	Quartal	int(11)	Quartal der Rechnung
	Typ	int(2)	Rechnungstyp (siehe Typen-Tabelle)
	Kommentar	varchar(255)	Kommentar
	Mwst	double(16,2)	enthaltener MwSt-Betrag
RechnungsPosition	ID	int(11) PRI	eindeutige ID
	RechnungsID	int(11)	Referenz-ID zur Rechnung
	BenutzerID	varchar(50)	Referenz-ID zum Benutzer
	ErstellDatum	datetime	Datum der Erstellung
	Betrag	double(16,4)	Einzelbetrag
	Beschreibung	varchar(100)	Beschreibung
	Waehrung	char(3)	Währungseinheit
RechnungsStatus	ID	int(11) PRI	eindeutige ID
	Beschreibung	varchar(50)	Beschreibung
RechnungsTyp	ID	int(2) PRI	eindeutige ID
	Beschreibung	varchar(50)	Beschreibung
Zahlungsweise	Code	char(2) PRI	eindeutige ID
	Beschreibung	varchar(50)	Beschreibung

Abbildung 4.3.: Datenmodell bawue Finanzbereich

Tabellenname	Feldname	Feldtyp	Kommentar
DBRechte	id	tinyint(4) PRI	eindeutige ID
	Name	varchar(20)	Name der Gruppe
	Beschreibung	varchar(100)	Beschreibung im Klartext
Domains	ID	int(11) PRI	eindeutige ID
	BenutzerDienstID	int(11)	Referenz-ID auf Dienst-ID
	Name	varchar(100)	Name der Domain
Host	BenutzerID	int(5) PRI	Referenz-ID Benutzer
	HostName	varchar(50) PRI	Name des Hosts
	IPAdresse	varchar(50)	IP-Adresse
	PPPPeerAdresse	tinyint(1)	Peer-IP-Adresse
Hilfetexte	id	int(11) PRI	eindeutige ID
	titel	varchar(255)	Titel
	text	text	Hilfetext
MailRoute	ID	int(5) PRI	eindeutige ID
	VirtualAddress	varchar(100) PRI	zu routende Adresse
	RedirectAddress	varchar(100)	Zielroute
	BenutzerID	int(11)	Referenz-ID Benutzer
Mailinglisten	id	int(11) PRI	eindeutige ID
	BenutzerDienstID	int(11)	Referenz-ID Dienst
	Name	varchar(255)	Name der Liste
Site	Name	varchar(50) PRI	Sitename
	FQDN	varchar(50)	DNS-Name der Site
	Vorwahl	int(5)	Ortsvorwahl
	MX10	varchar(50)	DNS-Mailexchange
	MX20	varchar(50)	DNS-Mailexchange
Regeln	DienstID	int(11) PRI	Referenz-ID Dienst
	StatusID	int(11) PRI	Referenz-ID Status
	Service	char(20) PRI	der resultierende UNIX-Service
	Prioritaet	smallint(6)	Sortierung
	Verhalten	char(20)	Verhaltensfunktion
PLZ_Ort_Geo	PLZ	int(5) PRI	Postleitzahl
	name	varchar(30)	Ortsname

Abbildung 4.4.: Datenmodell bawue Administratives

Tabellenname	Feldname	Feldtyp	Kommentar
DBlog	id	int(11) PRI	eindeutige ID
	Datum	datetime	Zeitstempel
	Autor	varchar(8)	Loginname im Klartext
	IP	varchar(255)	IP-Adresse
	Query	text	SQL-Query an Datenbank
stats	id	int(11) PRI	eindeutige ID
	ts	timestamp(14)	Zeitstempel
	nameid	int(11)	Referenz-ID auf statsconfig
	anzahl	int(11)	Ergebnis der Query
statsconfig	id	int(11) PRI	eindeutige ID
	name	varchar(255)	Name im Klartext
	query	text	auszuführende SQL-Query

Abbildung 4.5.: Datenmodell bawuelogs

Tabellenname	Feldname	Feldtyp	Kommentar
instructiontypes	id	int(11) PRI	eindeutige ID
	name	varchar(30)	Name des Anleitungs-/ FAQ-Typs
	color	varchar(7)	Darstellungsfarbe
instructions	id	int(11) PRI	eindeutige ID
	title	varchar(255)	Titel der Anleitung/FAQ
	typeid	tinyint(4)	Art der Anleitung/FAQ (instructiontypes.id)
	text	text	Inhalt der Anleitung/FAQ
	url	varchar(255)	URL
	autorid_create	int(11)	ID des Erstellers
	date_created	timestamp(14)	Datum der Erstellung
	autorid_lastmod	int(11)	ID des letzten Änderers
	date_lastmod	timestamp(14)	Datum der letzten Änderung
	instid	int(11)	ID der Anleitung/FAQ (zur Versionierung)
	version	int(11)	Versionsnummer
	hits	int(11)	Anzahl der Aufrufe (bisher unbenutzt)

Abbildung 4.6.: Datenmodell Anleitungen im Workflow bzw. FAQ

Tabellenname	Feldname	Feldtyp	Kommentar
events	id	int(11) PRI	eindeutige ID
	description	varchar(255)	Beschreibung in Klartext
hilfetexte	id	int(11) PRI	eindeutige ID
	titel	varchar(255)	Titel der Hilfe
	text	text	Hilfetext
pendingevents	id	int(11) PRI	eindeutige ID
	eventid	int(11)	Referenz-ID zum entsprechenden Event
	timestamp	timestamp(14)	Zeitstempel der Erstellung
	useridtodo	int(11)	ID des Users, auf den sich das Ticket beziehen soll
	done	tinyint(4)	wenn > 0, dann Tickets schon generiert
	comment	varchar(255)	Freitext-Kommentar
team	id	tinyint(4) PRI	eindeutige ID
	name	varchar(50)	Name des Teams
	email	varchar(255)	Email-Adresse des Teams
tickets	id	int(11) PRI	eindeutige ID
	timestamp	timestamp(14)	Zeitstempel der Erstellung
	timestampTBD	timestamp(14)	Zeitstempel Soll-Fertigstellung
	description	varchar(255)	Beschreibung im Klartext
	instid	int(11)	ID zur Anleitung
	teamid	int(11)	ID zum Team
	timestampDONE	timestamp(14)	Zeitstempel der Fertigstellung
	useridDONE	int(11)	ID des Users, der Ticket geschlossen hat
	useridtodo	int(11)	ID des Users, auf den sich das Ticket bezieht
	actionURL	varchar(255)	URL
	cronintervall	int(11)	Intervall für Wiedervorlage Angabe in Minuten
	comment	varchar(255)	Freitext-Kommentar
tickettemplate	id	int(11) PRI	eindeutige ID
	eventid	int(11)	Referenz-ID zum Event
	cronid	int(11)	Zeitstrahl-ID
	instid	int(11)	Referenz-ID zu Anleitung
	teamid	int(11)	ID zum Team
	description	varchar(255)	Beschreibung im Klartext
	announceagain	int(11)	Intervall für Wiedervorlage
	actionURL	varchar(255)	URL

Abbildung 4.7.: Datenmodell Workflow (ohne Anleitungen)

5. Entwurf und Realisierung der Applikationen

Die Applikationen teilen sich in drei Bereiche auf: administrativer Bereich, Automatismen und Benutzer-Frontend.

Es gibt zwei Typen von Benutzern der Webapplikationen:

- den normalen Benutzer, der über das Benutzer-Frontend `my.bawue` zugreift und
- den privilegierten Benutzer (Administrator), der die Applikation aus dem administrativen Bereich nutzt (`easy.bawue`).

Die Datenbank befindet sich auf dem zentralen Vereinsrechner[1], dort ist auch das Administrator-Frontend (`easy.bawue`) platziert. Der normale Benutzer hat dort keinen Zugang, sondern meldet sich über das Frontend `my.bawue` auf dem Rechner[2] an.

Eine grafische Übersicht zum Systemaufbau findet sich in Abbildung 5.1.

Die Automatismen nehmen den Administratoren durch automatische Überprüfung und Aktionen Arbeit ab.

5.1. Administrations-Bereich

Der Zugang zum Administrations-Bereich ist nur via SSL möglich. Hierfür wurde ein eigenes Zertifikat erstellt (aus Kostengründen). Die Authentifizierung des Kennwortes erfolgt über die standardisierte HTTP-Authentication der eingegebenen Benutzerdaten gegen den Benutzernamen und das DES-Kennwort in der Datenbank.[3]

Die Datensicherung erfolgt täglich über den Betreuer der Hardware vor Ort. Sporadische Dumps der Produktions-Datenbank werden von den Datenbank-Administratoren durchgeführt, um das Testsystem aktuell zu halten.

Die Benutzerführung entwickelte und veränderte sich im Laufe der Zeit stark. Es fanden zwei Umfragen statt, deren Auswertung in Kapitel 6.3 eine Veränderung der Navigation bewirkte (siehe Anhang B.3). Außerdem wurden im Laufe der Zeit neue Funktionen

[1] Dual-Pentium 2x200 MHz, 256 MB RAM
[2] Pentium 133, 196 MB RAM
[3] Als Webserver kommt Apache 1.3.9 mit den Modulen PHP/3.0.15, modssl/2.4.10 und OpenSSL/0.9.4 zum Einsatz. Die eingesetzte Datenbank ist MySQL 3.23.8-alpha

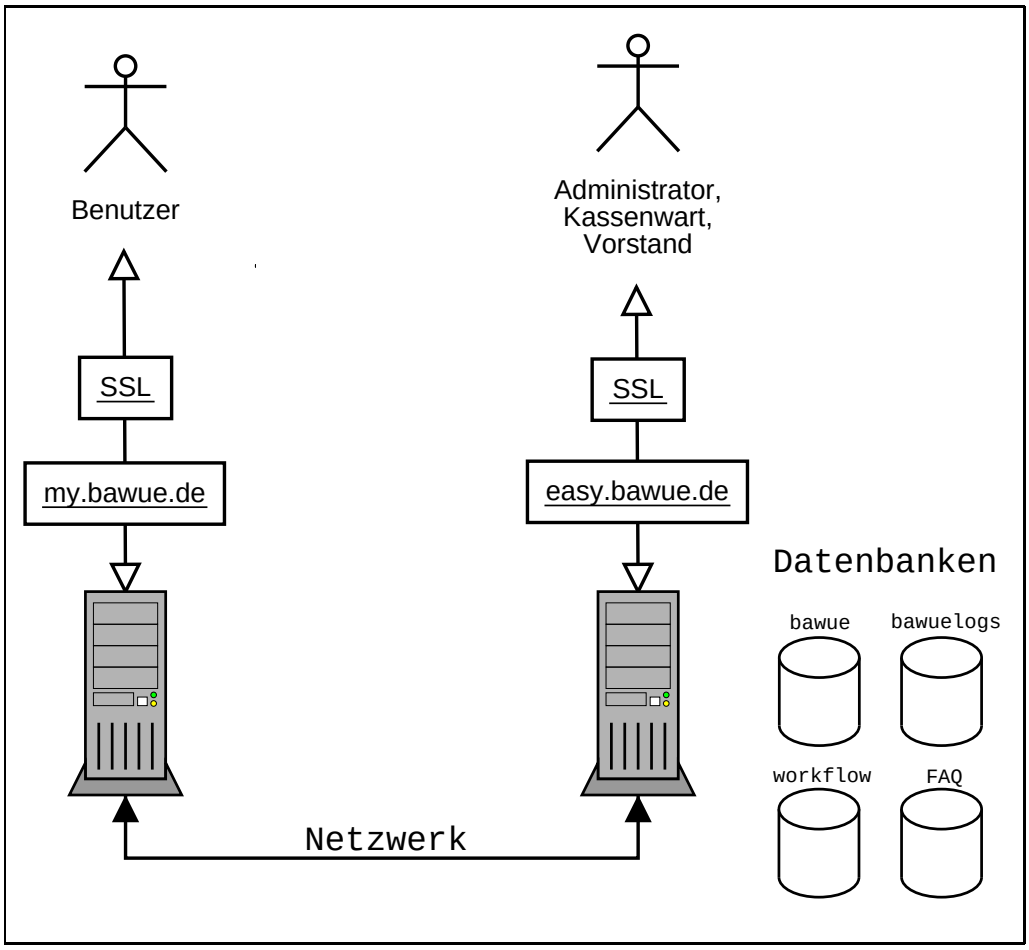

Abbildung 5.1.: Applikations- und Netzwerkstruktur

eingebaut (iterativer Entwurf; siehe spätere Einführung der Anleitungen, FAQ und des Workflow).

Es wurde großer Wert auf die Rückmeldung der Benutzer gelegt und deren Wünsche und Verbesserungsvorschläge soweit wie möglich realisiert. Zwei zeitlich getrennte, aber inhaltlich gleiche Umfragen gaben den Administratoren die Möglichkeit, an der Entwicklung der Applikation teilzuhaben.

5.1.1. Abgestuftes Benutzer-Modell

Es gibt verschiedene Stufen der administrativen Benutzer, die einheitlich über ein Feld in ihrem Datensatz der Tabelle `Benutzer` deklariert werden. Pro Benutzer ist nur ein Typus möglich, die Typen enthalten die Berechtigungen der vorherigen jeweils vollständig.

Es gilt: keinerlei Berechtigung $\subset$ readonly $\subset$ admin $\subset$ kasse $\subset$ dbadmin

Implementiert wird die Zugangsberechtigung via PHP am Anfang jeder HTML- bzw. PHP-Seite mit:

```
<?
  $REQUIRE[1] = "kasse";
  $REQUIRE[2] = "admin";
  $REQUIRE[3] = "dbadmin";
?>
```

In diesem Beispiel haben drei Gruppen Zugang zur Seite, Benutzer anderer Gruppen erhalten eine Fehlermeldung und können weder Seite noch Inhalt betrachten.

Die Auswertung der Rechte erfolgt in der Datei `bawue_db_authenticate.php3`, welche von allen PHP-Skripten am Anfang mitgeladen werden muss.

5.1.2. Kernapplikation easy.bawue

Idee

Zentrale Anlaufstelle zur Benutzer- und Systemverwaltung für alle Administratoren des Vereins, den Schatzmeister und die restlichen Mitglieder des Vorstandes.

Realisierung

Aufgrund des vorherigen Benutzer-Modells können Seiten der Applikation auf einfache Art und Weise entsprechenden Gruppen von Benutzern zugeordnet werden.

Der Kassenbereich ist beispielsweise nur für den Vorstand (incl. Schatzmeister) erreichbar. Die Datenbank-Admins können zusätzlich noch Logfiles betrachten. Alle restlichen Seiten sind für alle Nutzer des Admin-Systems zugänglich. Der Read-Only-Nutzer kann Daten abfragen, jedoch keine Änderungen dieser Daten speichern.

Außerdem wurde ein Hilfesystem mit Javascript-Fenster und Datenbank-Anbindung integriert.

Im Einzelnen wurden folgende Funktionen implementiert:

- Userverwaltung
  - Anzeigen neu angelegter Dienste für Benutzer
  - Suchen nach Benutzern (nach verschiedene Kriterien)
- Workflow
  - Anzeigen offener/geschlossener Tickets
  - Anzeigen der Tickettemplates
  - Anlegen von Einzeltickets
  - Anlegen eines PendingEvents
- Listen/Berichte

  - Datenbank-Statistiken
  - Übersicht und Suchen der Mailroute
  - vorbereitete Userlisten (nach Diensten)
  - vorbereitete Listen der Vereinsmitglieder (Telefon-, Unterschriftenliste, etc.)
- Admin-Links
  - diverse Links zu externen Tools (Statistiken, DB-Frontend, Mailman, etc.)
  - Formulare zur Eingabe von Daten in die Datenbank
- Anleitungen
  - Anzeigen einer Übersichtsliste
  - Suchfunktion
  - Erstellung Neueintrag
- FAQ
  - Anzeigen einer Übersichtsliste
  - Suchfunktion
  - Erstellung Neueintrag
- Kasse
  - Rechnungen (anzeigen, erzeugen, drucken)
  - Bankeinzug (erzeugen)
  - Kostenaufschläge (anzeigen, erzeugen)
  - Links zur Dokumentation
- DBAdmin
  - Datenbank-Logfile anzeigen
  - Datenbank-User anzeigen
  - User löschen
  - Dokumentation über den Versionstand

Programm-Beispiel

Auszug aus `user-suchen-ergebnis.php3`

```
...
<?
### wenn kein Ergebnis gefunden: Auto-Reload auf Suchmaske

if ($rows==0)
{
  echo "<meta http-equiv=\"Refresh\"
```

```
        content=\"10;url=user-suchen.php3\">"
;
}
?>

</HEAD>
<body bgcolor="#FFFFFF">

<H1><HR size=1>Suchergebnis<HR size=1></H1>

<?
### wenn kein Ergebnis gefunden: Mitteilung in roter Schrift

if ($rows==0)
{
  echo "<p><font color=red>keine Datens&auml;tze gefunden!
        Auto-Reload der Suchmaske in 10 Sekunden.</font></p>";
}

?>

<table bgcolor="#FFFFFF" cellspacing="1" cellpadding="1" border="0">
<tr>
  <td>Name:</td>
  <td><? echo $Name ?></td>
</tr>
<tr>
  <td>Vorname:</td>
  <td><? echo $Vorname ?></td>
</tr>
...
```

Screenshots

Siehe Abbildung 5.2.

5.1.3. Finanzen

Idee

Rechnungen, evtl. nachfolgende Mahnungen und quartalsweiser Bankeinzug der Beiträge können nun mit Hilfe der Datenbank erzeugt und verwaltet werden. Insbesondere kann der Schatzmeister aus dem System heraus auf Knopfdruck eine Bankeinzugsdiskette generieren und diese anschließend bei der Bank einreichen.

In einem früheren System wurden zwei voneinander getrennte Applikationen verwendet, was häufig zu „menschlichen“ Fehlern geführt hat, da die Daten manuell abgeglichen

Abbildung 5.2.: easy.bawue: Suchergebnis Benutzer

werden mussten.

Ein weiteres Merkmal ist die regelmäßige, automatische Konsistenzprüfung der Bankdaten, um Fehlbuchungen und somit unnötige Kosten zu vermeiden (siehe Kapitel 5.2)

Realisierung

Das zeitraubendste Problem stellte die Erzeugung einer korrekten Bankauszugsdatei `dtaus0.txt` dar. Mangelhafte Dokumentation dieses Datenträgeraustausch-Formates [dtaus] und die Nicht-Existenz eines Prüfprogrammes (inzwischen liegt dem Verein eines vor) führte zu mehrfachen Iterationen bis zum ersten erfolgreichen Bankeinzug. Frühere Bankeinzüge wurden über die vom damaligen Schatzmeister eingesetzte Buchhaltungssoftware erzeugt.

Ablauf der Bankeinzugsdisketten-Erzeugung:

1. Erzeugung der Rechnungen für das aktuelle Quartal
2. Datensätze für den Bankeinzug anzeigen (es werden alle offenen Rechnungen mit den Benutzern verknüpft, die Bankeinzug als Zahlungsweise angegeben haben)
3. Erzeugung der Datei `dtaus0.txt` und speichern auf einer Diskette

4. Ausdruck des Begleitzettels
5. Abgabe von Diskette und Begleitzettel in der Bankfiliale

Programm-Beispiel

Auszug aus `rechnungsliste-anzeigen.php3`

```
...
<table border=0 bgcolor="#FFFFFF">
<form action="rechnungsliste-anzeigen-ergebnis.php3">
<tr>
<td align="right">Rechnungstyp:</td>
<td>
        <SELECT NAME="select_rechnungstyp">
        <OPTION SELECTED VALUE="alle">alle</OPTION>
        <?
        $i=0;
        while($i<$rows_typ):
            $typname = mysql_result($result_typ,$i,"Beschreibung");
            $typid = mysql_result($result_typ,$i,"ID");
                echo '<OPTION VALUE = "';
                echo $typid;
                echo '">';
                echo $typname;
                echo "</OPTION>";
            $i++;
        endwhile;
        ?>
        </SELECT>
</td>
</tr>
...
```

Screenshots

Siehe Abbildung 5.3.

5.1.4. Workflow-Unterstützung

Idee

Ziel ist es, eine Umgebung zu schaffen, in der regelmäßige, sich wiederholende oder auch einmalige Abläufe („Todos“) für verschiedenen Personengruppen („Teams“) in Form von virtuellen Arbeitszetteln („Tickets“) erfasst und applikationsübergreifend verwaltet werden können.

Zum Beispiel müssen bei der Neuanmeldung einer Domain bestimmte Schritte in einer bestimmten Reihenfolge durchgeführt werden. Zuerst muss ein RIPE-Handle generiert, dann Nameserver-Einträge erstellt und als letztes die eigentliche Registrierung der

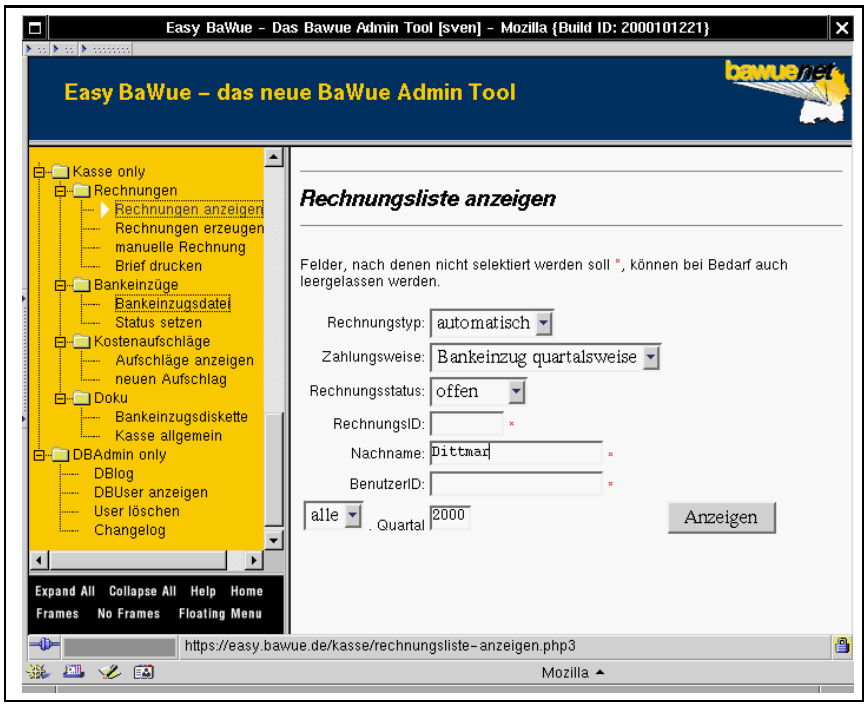

Abbildung 5.3.: easy.bawue: Suchmaske Rechnungen

Domain beim DENIC-Partner durchgeführt werden.

Auch der Antrag eines Benutzers auf Aufnahme als Mitglied in den Verein muss in einer bestimmten Reihenfolge innerhalb einer vordefinierten Zeitspanne (Benachrichtigung per Email, Ausdruck von Satzung und Begrüßungsschreiben) erledigt werden.

Um die Durchführung solcher Aufgaben zu vereinfachen und automatisieren, wurde folgende Web-Applikation entwickelt.

Realisierung

Aus einer virtuellen Beschreibung (`tickettemplates`) werden eine Menge realer Tickets generiert, sobald ein regelmäßig ablaufendes Skript einen neuen Eintrag in der Tabelle `pendingevents` vorfindet.

Die Reihenfolge der virtuellen Tickets wird auf einem Zeitstrahl (ohne konkrete Einheit) mittels einer `cronid` (ein Integer-Wert) realisiert. Je höher die `cronid`, desto weiter hinten auf dem Zeitstrahl liegt das Ticket.

Die realen Tickets erhalten einen Zeitstempel, bis zum dem das Ticket abgearbeitet sein muss, sonst wird das „Team“ von dieser Zeitüberschreitung in Kenntnis gesetzt. Dies geschieht per Email und durch farbliche Markierung und Umsortierung der Tickets im Web-Frontend.

Der Zeitstempel `timestampTBD` wird berechnet aus der Summe der `accounceagain`-Werte (in Minuten) aller vorherigen Tickets, also der aufsummierten Zeitspannen bis zur Wiedervorlage. Falls mehrere Tickets zu einer `cronid` existieren, so wird nur der jeweils größte Wert verwendet.

Programme, die einen Event auslösen, setzen einen Neueintrag in die Tabelle `pendingevents`. Ein Perlskript arbeitet die Einträge dort stündlich ab und erzeugt bei Neueinträgen die realen Tickets gemäß der Anleitung aus `tickettemplate` und markiert den `pendingevent` als „done“.

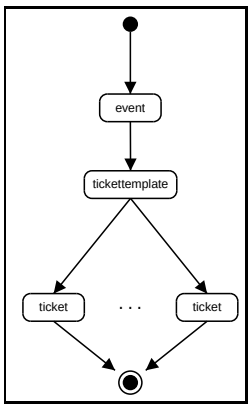

Abbildung 5.4.: Workflow: Aktivitätsdiagramm Events/Ticketerstellung

Genereller Ablauf:

1. Ein PendingEvent wird erzeugt, durch eine Applikation oder manuell im Web-frontend.
   `insert_event.pl`

2. Ein Cronjob findet neue Events und erzeugt via vordefiniertem Tickettemplate einen Satz realer Tickets daraus, anschließend wird der Status des PendingEvents auf „done“ gesetzt und eine Email an das Team geschickt, dass neue Tickets generiert wurden.
   `create_tickets.pl`

3. Nach Ablauf von `timestampTBD` werden Mails an das Team geschickt, die mitteilen, dass noch ein oder mehrere Tickets offen sind. Zusätzlich wird das Ticket im Web-Frontend farblich hervorgehoben und nach vorne sortiert.
   `check_for_open_tickets.pl` und `show_open_tickets.php`

4. Erledigung der Arbeitsaufträge, Auswahl und Bestätigung im Web-Frontend.
   `ticket-detail.php` und `ticket-close.php`

5. Übersicht aller Aufträge (offen, geschlossen) im Web-Frontend
   `show_open_tickets.php` und `show_closed_tickets.php`

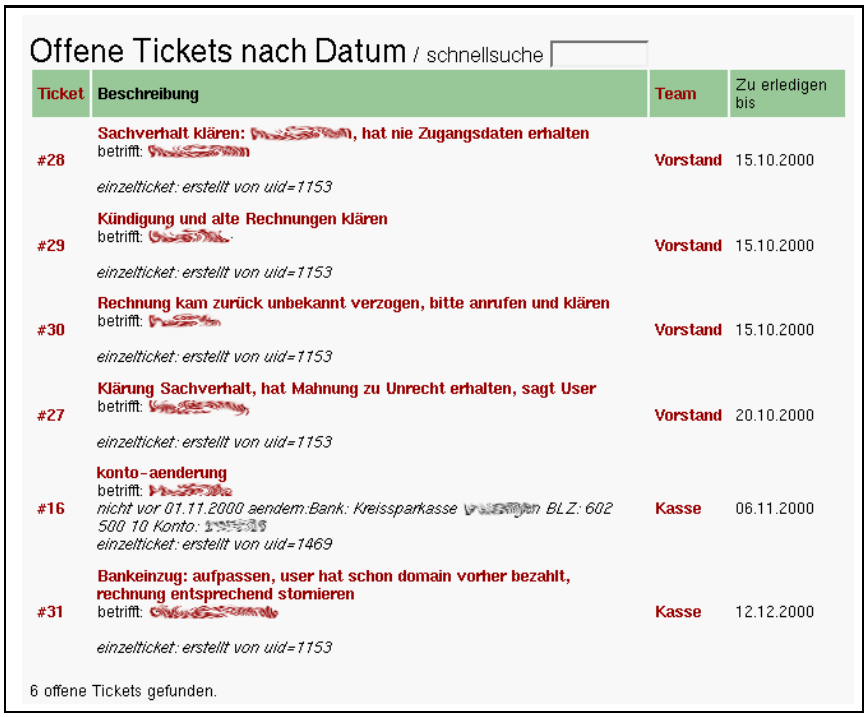

Abbildung 5.5.: Workflow: Ticket-Liste

Programm-Beispiel

Auszug aus `create_tickets.pl`

```
...
my $actionURL = $data2[7];
my $teamname  = $data2[9];
my $teamemail = $data2[10];

print "\t addtime=$addtime Minuten (cronint=$cronint, j=$j)\n";

$insert="INSERT INTO tickets (timestamp, timestampTBD, description,
         instid, teamid, useridtodo, actionURL, cronintervall, comment)
         VALUES(\"$tsnow\",DATE_ADD(\"$ts\", INTERVAL $addtime MINUTE),
         \"$desc\",\"$instid\", \"$teamid\", \"$useridtodo\",
         \"$actionURL\", \"$cronint\", \"$comment\")";

#
# INSERTs machen
#
$sth3 = $dbh -> query($insert)
        or die "Couldn't insert statement: " . $dbh->errstr;
$sth3 -> finish;
print "\tDONE: $insert\n";
```

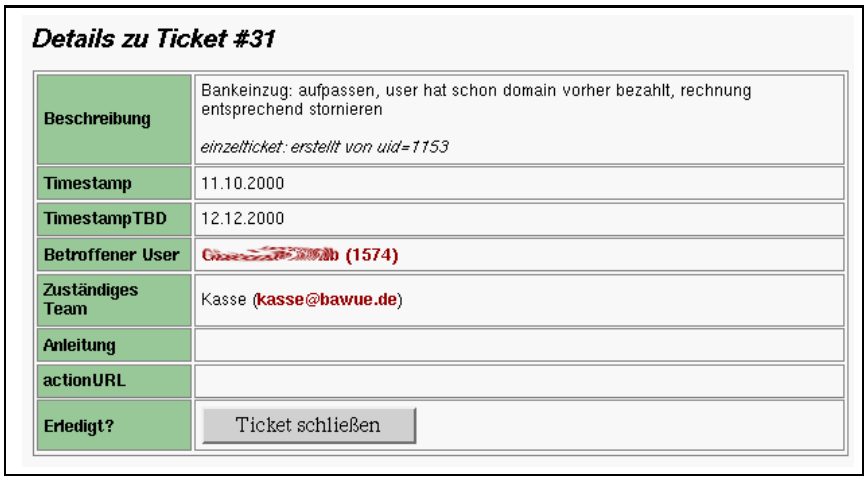

Abbildung 5.6.: Workflow: Ticket Detail-Ansicht

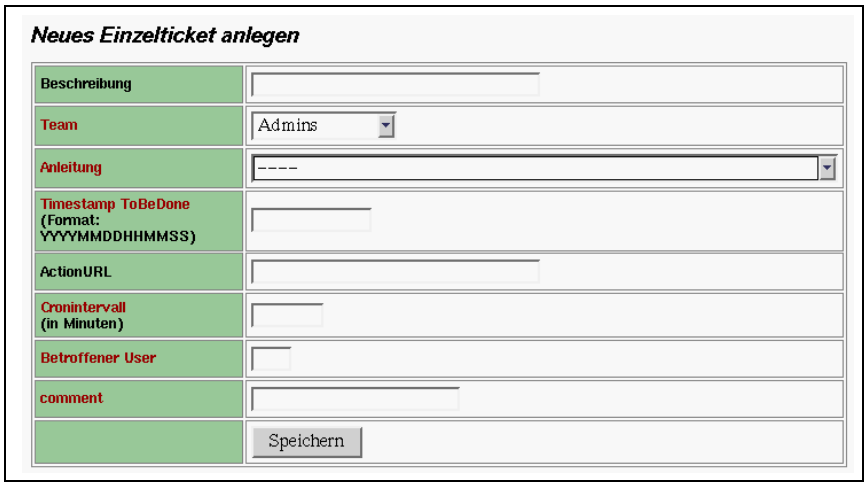

Abbildung 5.7.: Workflow: Ticket manuell anlegen

...

Screenshots

Siehe Abbildungen 5.5, 5.6 und 5.7. Aus datenschutzrechtlichen Gründen wurden die Namen verwischt.

5.1.5. Anleitungen/FAQ-Unterstützung

Idee

Es soll eine datenbankgestützte Applikation entworfen werden, die häufig auftauchende Fragen und Antworten der Administratoren („Anleitungen“) und Benutzer („FAQ“) einheitlich verwaltet.

Diese sollen einer Versionierung unterworfen sein und es soll möglich sein, HTML-Formatierungen innerhalb der Anleitung/FAQ vorzunehmen. Desweiteren soll nach Schlüsselworten gesucht werden können.

Realisierung

Das zugehörige Datenmodell findet sich in Kapitel 4.3 und Abbildung 4.6.
Da es in MySQL keine „Subselects“ gibt, muss hier ein kleiner Kunstgriff angewandt werden, um an die jeweils neueste Version einer Anleitung bzw. FAQ zu kommen: der SELECT liefert alle Versionen, in der Schleife werden jedoch nur die Versionen ausgegeben, die die höchste Versionsnummer haben (siehe Programm-Beispiel).

Die Anleitungs- und FAQ-Datenbank sind zwar getrennte, aber gleichartige Datenbanken und Programme, mit Ausnahme der HTML-Formatierung von Überschriften etc.

Programm-Beispiel

Auszug aus `anleitungs-liste.php3`

```
...
MYSQL_SELECT_DB("todo");

if ($keyword)
{
  $where = "instructions.typeid = instructiontypes.id AND
            instructions.title LIKE \"%$keyword%\"";
}
else
{
  $where = "instructions.typeid = instructiontypes.id";
}

$query_string = 'SELECT  * FROM instructions, instructiontypes
                 WHERE '.$where.'
                 ORDER BY typeid, instid, title, version DESC';
...
while($i<$rows):
  $lastinstid=$instid;
  $instid=mysql_result($result,$i,"instid");
  # nur ausgeben, wenn eine neue Anleitung kommt (nicht eine alte Version)
  if ($lastinstid != $instid)
...
```

Screenshots

Siehe Abbildungen 5.8 und 5.9. Aus datenschutzrechtlichen Gründen wurden die Namen verwischt.

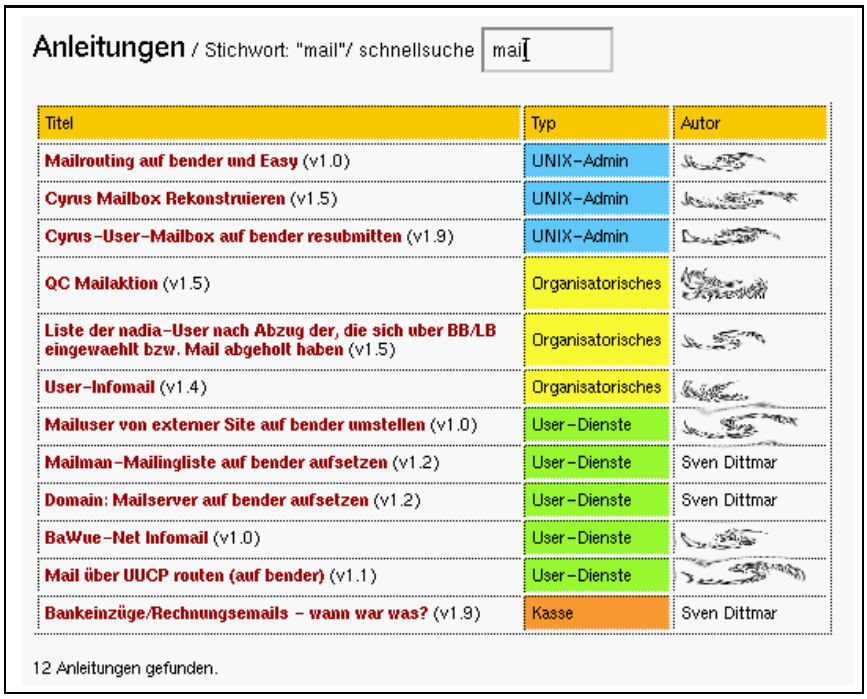

Abbildung 5.8.: Anleitungen/FAQ: Liste

5.2. Automatismen

Die Automatismen sind Perl-Programme, die via Cronjob regelmäßig auf dem Vereinsrechner ausgeführt werden.

Dies ist notwendig, weil MySQL keine „constraints“ (Einschränkungen aufgrund vorgegebener Bedingungen beim Einfügen von Datensätzen) kennt und das Web-Frontend nicht alle Eingaben schon bei der Erfassung überprüft. Außerdem sollen einige Programme selbstständig ohne Eingriff der Administratoren ablaufen (Erzeugung der Mailinglisten und der Statistiken).

5.2.1. Kontodatenprüfung

Montags um sechs Uhr werden die Kontodaten der Benutzer mit der internen Bankleitzahlen-Datenbank verglichen. Außerdem wird nach illegalen Zeichenketten gesucht (führende Nullen oder Nicht-Zahlzeichen) und Logikfragen („Bankeinzug als Zahlungsweise eingetragen, aber keine Bankdaten vorhanden“) überprüft. Dies war insbesondere nach dem Import der Daten aus der alten Vereinsdatenbank und für die Bankeinzugsdatei nötig.

Sollten Differenzen auftreten, so wird der Schatzmeister per Email davon in Kenntnis gesetzt. Da die quartalsweise Bankeinzugsdatei `dtaus0` aus dieser Datenbank generiert wird, ist diese Funktionalität sehr wichtig.

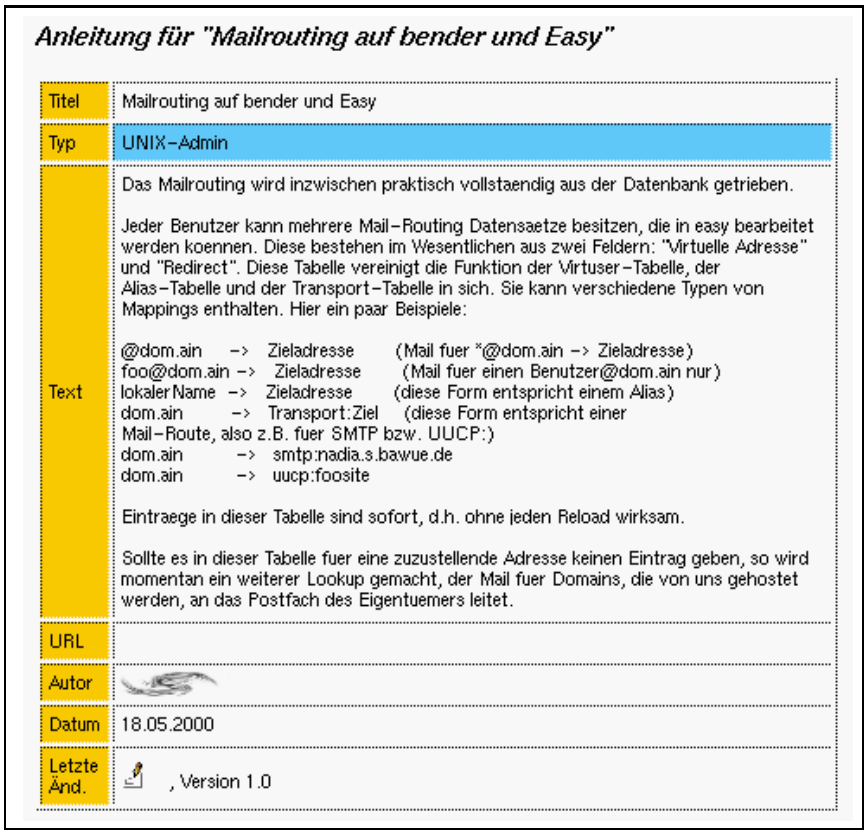

Abbildung 5.9.: Anleitungen/FAQ: Detailansicht

5.2.2. Zweitnutzertest

Sollte sich ein Nutzer vom Verein abmelden, so müssen auch seine Zweitnutzer gelöscht werden, die im Datenmodell eine entsprechende ErstnutzerID gesetzt haben, was vom Administrator übersehen werden könnte.

Es darf somit kein aktiver Zweitnutzer existieren, dessen Erstnutzer keine aktiven Dienste mehr hat. Dies wird montags und donnerstags um sechs Uhr geprüft und das Ergebnis per Email an die Datenbankadministratoren gemeldet.

5.2.3. PEmail

Jeder aktive Benutzer muss genau eine sogenannte primäre Email („PEmail“) besitzen, damit er für die Administratoren, den Vorstand und den Schatzmeister erreichbar ist und automatisch in Mailinglisten eingetragen werden kann.

Auf Basis eines Cronjobs wird hier ebenfalls montags und freitags jeweils um sechs Uhr morgens überprüft, ob jeder aktive Nutzer auch genau eine PEmail besitzt.

5.2.4. Kollision Account und Mailrouten

Das Mailsystem „Cyrus“ legt für neue Benutzer automatisch eine Mailbox mit deren Account-Namen an. Hierbei ist zu prüfen, dass eine von Hand angelegte Mailroute nicht mit der Mailbox und damit gewünschten Mailzustellung des Benutzers kollidiert.

Beispiel: Es gibt einen Account `andreas` und daher eine Mailbox `andreas`. Nun legt sich Administrator eine Mailroute `andreas@bawue.de --> meinaccount@bawue.de` an, was die Mailbox des Users `andreas` umleiten würde.

Diese Kollisionkontrolle geschieht zweimal täglich um 10 und 18 Uhr mit folgendem Select-Statement:

```
SELECT Account, VirtualAddress, RedirectAddress, Benutzer.ID AS id1,
       MailRoute.BenutzerID AS id2
FROM   Benutzer, MailRoute
WHERE  Benutzer.ID != MailRoute.BenutzerID
       AND ( Benutzer.Account = MailRoute.VirtualAddress
       OR CONCAT(Benutzer.Account,'@bawue.de')
          = MailRoute.VirtualAddress )
ORDER BY Account
```

5.2.5. Statistik

Es wurde eine Applikation und Datenbank entworfen, die SELECT-Ausdrücke speichert, diese via Perl-Skript ausliest und ausführt. Das Ergebnis stellt die Anzahl der Antworten auf den SELECT dar und wird separat mit einem Zeitstempel gespeichert („Gedächtnisfunktion“).

Zur Zeit werden folgende Zahlen täglich um sechs Uhr „gemerkt“:

- Anzahl der gehosteten Domains (mit Select-Statement als Beispiel)

  ```
  SELECT count(*)
  FROM   Benutzer, Benutzer_Dienst, Dienst, Status, Domains
  WHERE  Benutzer.ID = Benutzer_Dienst.BenutzerID
     AND Benutzer_Dienst.StatusID = Status.ID
     AND Benutzer_Dienst.DienstID = Dienst.ID
     AND Domains.BenutzerDienstID = Benutzer_Dienst.ID
     AND Benutzer_Dienst.Datum    > "0001"
     AND Benutzer_Dienst.Datum    < "9999-12-31"
     AND Status.Beschreibung      = "normal"
     AND Dienst.Beschreibung      = "Domainhosting"
  ```

- Anzahl der IP-Nutzer
- Anzahl der Zweitnutzer
- Anzahl der Shell-Account-Nutzer

- Anzahl der UUCP-Nutzer
- Verteilung der IP-Nutzer auf die Einwahlsites
- Anzahl der offenen Rechnungen (mit Select-Statement als Beispiel)

```
SELECT count(*)
FROM   Rechnung
WHERE  Rechnung.Status = '0'
```

- Anzahl der Mailrouten
- Anzahl der Vereinsmitglieder

5.2.6. Generator für Mailinglisten

Idee

Als Mailinglisten-Programm kommt Mailman in der Version 1.1 zum Einsatz [Mailman].

Mailman bietet ein Kommandozeilen-Programm `sync_members` an, das mit Textdateien gefüttert werden kann und aufgrund der dort angeführten Emailadressen die Mailingliste verwaltet, also neue Adressen hinzufügt und alte löscht.

Realisierung

Via Cronjob wird täglich um sechs Uhr ein Perlskript ausgeführt, das eine Abfrage an die Vereinsdatenbank startet und Vorname, Name und Emailadresse der aktiven Personen ausgibt.

Durch Umleitung dieser Daten in eine temporäre Datei kann das `sync_members` Programm damit gefüttert werden und löscht bzw. fügt User entsprechend hinzu.

Programm-Beispiel

Erzeugen der Daten für die Mailingliste `domainowner` aus der Datenbank:

```
SELECT DISTINCT Benutzer.Name, Benutzer.Vorname,Kontakt.Adresse
FROM            Benutzer, Benutzer_Dienst, Kontakt
WHERE           Benutzer.ID=Benutzer_Dienst.BenutzerID
            AND Benutzer.ID=Kontakt.BenutzerID
            AND Kontakt.Typ='PEmail'
            AND Benutzer_Dienst.StatusID=20
            AND Benutzer_Dienst.DienstID=5
ORDER BY        Benutzer.Name
```

Erzeugung der Liste „domainowner“:

```
erzeuge_mliste.pl domainowner > list.domain
```

Aufruf von `sync_member`:

```
/opt/mailman/bin/sync_members -w=no -a=no -f list.domain domainowner
```

5.3. Benutzer-Frontend my.bawue

Der Zugang zum Benutzer-Frontend ist nur via SSL möglich. Hierfür wurde - wiederum aus Kostengründen - ein eigenes Zertifikat erstellt.
Die Authentifizierung des Kennwortes erfolgt über die standardisierte HTTP-Authentication der eingegebenen Benutzerdaten gegen den Benutzernamen und das DES-Kennwort in der Datenbank.[4]

Die Benutzerführung lehnt sich an das Layout der Vereinshomepage `http://www.bawue.de/` an (andere Farben, um der Verwechslungsgefahr aus dem Weg zu gehen, aber gleiches Layout für die Navigationsführung). Eine kleine Gruppe freiwilliger Beta-Tester aus dem Kreise der Domainbesitzer des BaWue-Net e.V. hat das System im Mai 2000 getestet, ihre Rückmeldungen bezüglich des Mail-Frontends und Bugreports wurden gerne angenommen.

5.3.1. Mailreader

Idee

Da das Lesen und Schreiben von Emails einer der wichtigsten Punkte für die Benutzer ist, liegt es nahe, diesen Dienst auch in das Web-Frontend my.bawue zu integrieren.

Damit können beispielsweise Emails von einem Internet-Café aus beantwortet werden.

Realisierung

Zum Einsatz kommt ein Open Source Mailprogramm namens AeroMail (`http://the.cushman.net/reverb/aeromail/`), welches in PHP geschrieben, von Frank Berger installiert und angepasst (Authentifizierung über die Vereinsdatenbank, Erweiterung von Funktionen) wurde.

Screenshot

Siehe Abbildung 5.10.

5.3.2. Mailroute konfigurieren

Idee

Insbesondere für Domainbesitzer ist es interessant, eigene Mailrouten (beispielsweise quelle@eigene.domain.de –> ziel@andere.domain.de) erstellen und ändern zu können. Mailrouten können z.B. Mail-Aliase oder Forwards zu anderen Mailboxen im Internet sein.

Realisierung

Nach Prüfung der Benutzer-Eingabedaten (Prüfung auf Eingabe einer Empfängeradresse und Verwendung illegaler Zeichen) werden die Daten in die Datenbank geschrieben

[4] Als Webserver kommt Apache 1.3.9 mit den Modulen PHP/3.0.15, modssl/2.4.10 und OpenSSL/0.9.4 zum Einsatz. Die eingesetzte Datenbank ist MySQL 3.22.23b.

Abbildung 5.10.: my.bawue: AeroMail

(Aktualisierung, Neuerstellung oder Löschung einer Mailroute sind hierbei möglich).

Das Mailsystem „Postfix“ seinerseits befragt regelmäßig die Datenbank, es sind also keine weiteren Schritte notwendig, um eine Änderung dort zu aktivieren.

Programm-Beispiel

Auszug „Prüfung auf illegale Zeichen“ aus `query_mailroute.php3`

```
...
# negated charater class in regex
  if (ereg("[^\.0-9a-z_A-Z\-]", $virtadr))
  {
      $msg = $msg . '<li>Es wurden ung&uuml;ltige Zeichen verwendet!<br>';
      $msg = $msg . 'Erlaubt sind nur a-zA-z_0-9 . und -';
      $err++;
  }
...
```

Screenshot

Siehe Abbildung 5.11.

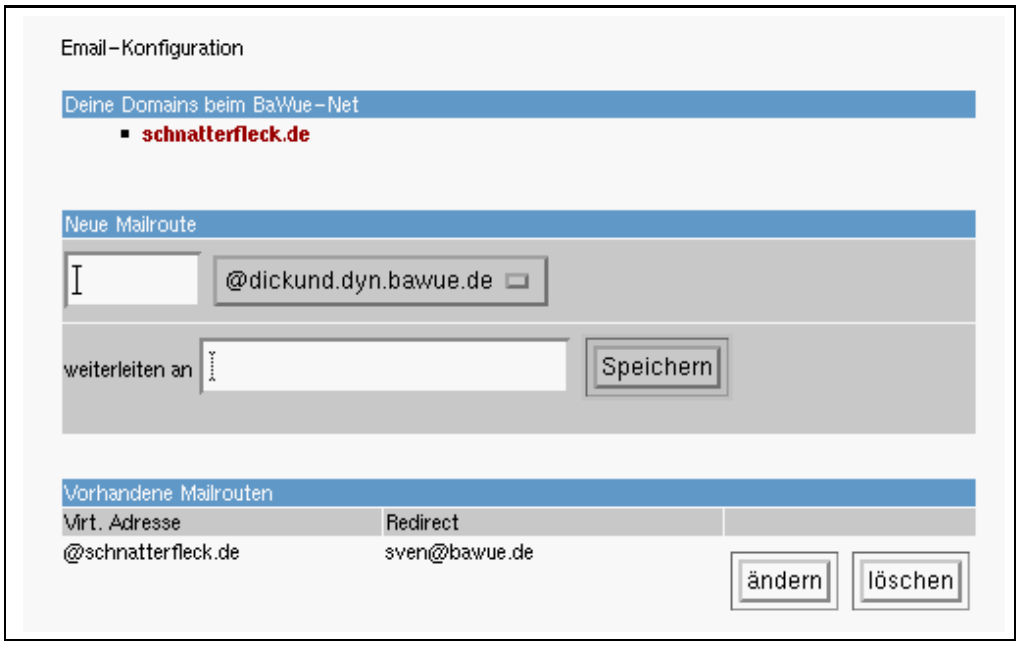

Abbildung 5.11.: my.bawue: neue Mailroute eintragen

5.3.3. Stammdaten verwalten

Idee

Schon lange gibt es im Verein die Idee, dass es praktisch wäre, wenn jeder Benutzer zumindest lesenden Zugriff auf seine im Verein gespeicherten, persönlichen Daten hätte, da dies Transparenz erhöht und dem Benutzer Kontrolle über die von ihm gespeicherten Daten gibt.

Realisierung

Die Realisierung ist recht einfach, da eine ähnliche Ausgabe der Stammdaten schon für das Administrator-Frontend erstellt wurde. Mit einigen kleinen Kürzungen (Historie, Kennwort) kann das angepasste PHP-Skript eingesetzt werden.

Programm-Beispiel

Auszug aus `daten.php3`

```
...
<td bgcolor="#FFFFFF"> </td>
<td bgcolor="#CCCCCC">Bankkonto</td>
<td bgcolor="#FFFFFF"> <? echo mysql_result($result,$i,"BankKonto"); ?> </td>
<td bgcolor="#CCCCCC">BLZ</td>
<td bgcolor="#FFFFFF"> <? echo mysql_result($result,$i,"BankBLZ"); ?> </td>
...
```

Screenshot

Siehe Abbildung 5.12.

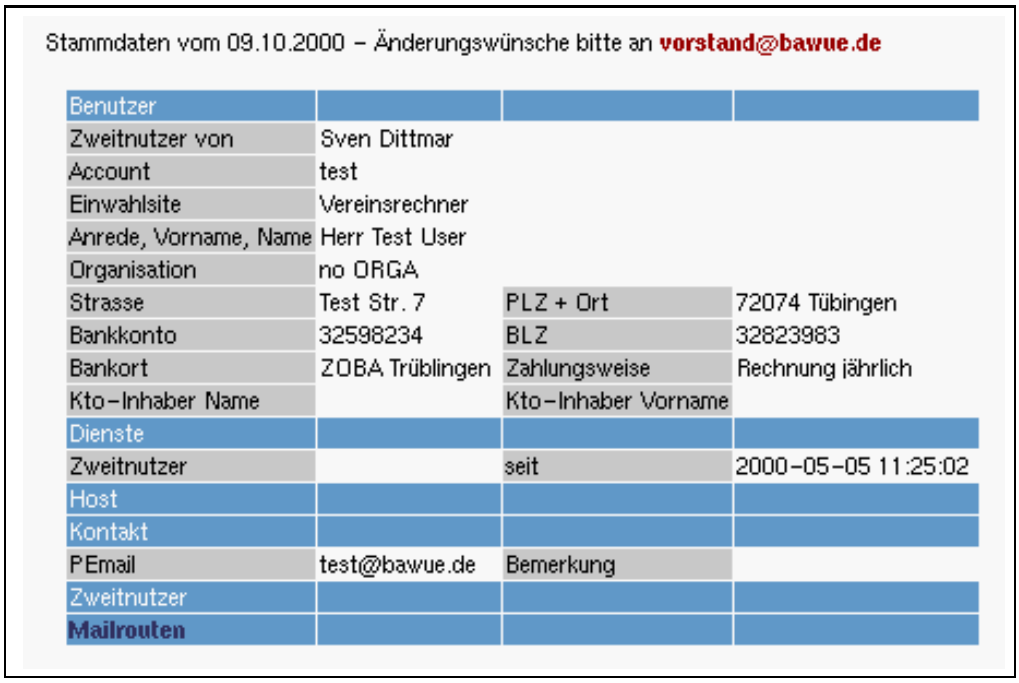

Abbildung 5.12.: my.bawue: Stammdaten

5.3.4. Kennwort ändern

Idee

Da das zentrale Kennwort eines Benutzers als DES-String in der Datenbank gespeichert ist, sollte es für den Benutzer auch eine Möglichkeit geben, dieses ändern zu können. Durch das Kommandozeilen-Programm `usermod` von Linux und einen Cronjob kann dieses Kennwort regelmäßig auf allen Servern synchronisiert werden.

Realisierung

Durch die PHP-Funktion `crypt` kann überprüft werden, ob das alte, eingegebene Kennwort korrekt ist, und anschließend wird das neue Kennwort DES-konform verschlüsselt und gespeichert.

Programm-Beispiel

Auszug aus `daten.php3`

```
...
$check=crypt($oldpasswd,"$dbpasswd");
if ($check != $dbpasswd)
{
   echo "Sorry, aber das alte Pa&szlig;wort war nicht richtig.<br>";
   $msg="Pa&szlig;wort <b>nicht</b> ge&auml;ndert!";
}
else
```

```
{
   echo "Altes Pa&szlig;wort richtig.<br>";
   if ($newpwd1 = $newpwd2 AND $newpwd1 != "" AND strlen($newpwd1)>3)
   {
      # altes und neues Kennwort okay, weitermachen!
      srand((double)microtime()*1000000);
      $salt=rand();
      # erzeuge neuen DES string mit $salt
      $newcrypt=crypt($newpwd1,$salt);
      #---- UPDATE des DESpasswort im Benutzerdatensatz ----
      $query_string = 'UPDATE Benutzer SET DESPasswort="'.$newcrypt.'"
                      WHERE ID="'.$userid.'"';
...
```

Screenshot

Siehe Abbildung 5.13.

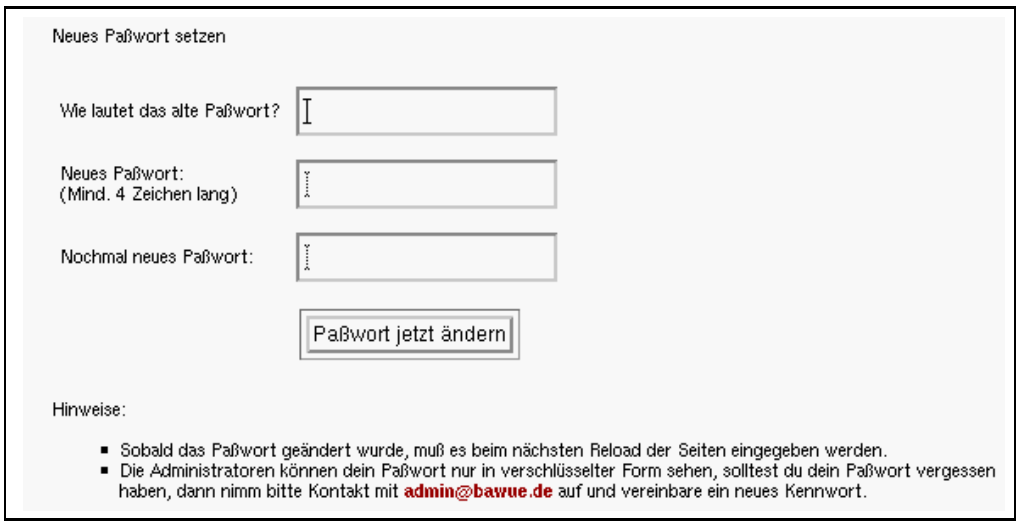

Abbildung 5.13.: my.bawue: Kennwort ändern

6. Softwaretechnische Aspekte

6.1. Projektverlauf, Zeitrahmen

Das Projekt startete im November 1999 mit einem Rohentwurf der neuen Datenbank basierend auf einer Analyse der alten Vereinsdatenbank. Die anschließende Migration der Altdaten in die neue Struktur ermöglichte Ende November die ersten Tests mit der neuen Software. Hierbei waren anfangs insbesondere die beiden Vereinsmitglieder Jörg Henne und Daniela Gehle involviert.

Ein internes Testsystem und eine erste Version für die Admins entstand schon gleich Anfang Dezember 1999. Dieses wurde dann unter Einbeziehung der Administratoren sukzessive weiterentwickelt.

Einen großen Teil der Zeit im Januar 2000 nahm die Korrektur der schlecht gepflegten und großteils fehlerhaften Daten aus der alten Vereinsdatenbank ein.

Anfang Mai fand der erste erfolgreiche Bankeinzug direkt aus dem neuen System statt.

Die Monate März und April waren geprägt von Bugfixing und vielen kleinen Erweiterungen im Bereich der Rechnungserzeugung, des Bankeinzugs und der Benutzerverwaltung.

Ab Mai wurde mit der Implementierung des Web-Frontends my.bawue begonnen. Nach einmonatiger Testphase mit einigen freiwilligen Beta-Testern wurde das System freigegeben und wird seither mit durchschnittlich 60 Anfragen pro Stunde (Stand September 2000) für Emails, zur Stammdatenprüfung und Mailrouten-Verwaltung genutzt. Die Installation und Betreuung des Mail-Frontends wurde von Vereinsmitglied Frank Berger übernommen.

Da inzwischen eine ausgereifte und stabile Version des Gesamtsystems erreicht wurde, kamen im Juli diverse Automatismen in Perl hinzu und es fand die erste Umfrage unter den Administratoren statt.

Parallel dazu begann der Entwurf und die Entwicklung des Workflow-Systems, welches im September freigeschaltet wurde und sich seitdem reger Benutzung erfreut. Das easy.bawue-System hatte im September 2000 durchschnittlich 18 Aufrufe pro Stunde.

Im September fand auch die zweite Befragung und eine abschließende Benotung des Gesamtsystems unter den Administratoren statt.

Das Projekt wurde Anfang Oktober 2000 vorübergehend eingefroren, um zu einer stabilen Zwischenversion zu kommen und diese Diplomarbeit fertig zu stellen.

Es stehen aber schon einige gewünschte Erweiterungen auf dem Plan, die nach Abschluss dieser Arbeit implementiert werden sollen. Diese werden in Kapitel 7 angesprochen.

Der zeitliche Projektverlauf ist in Abbildung 6.1 dargestellt.

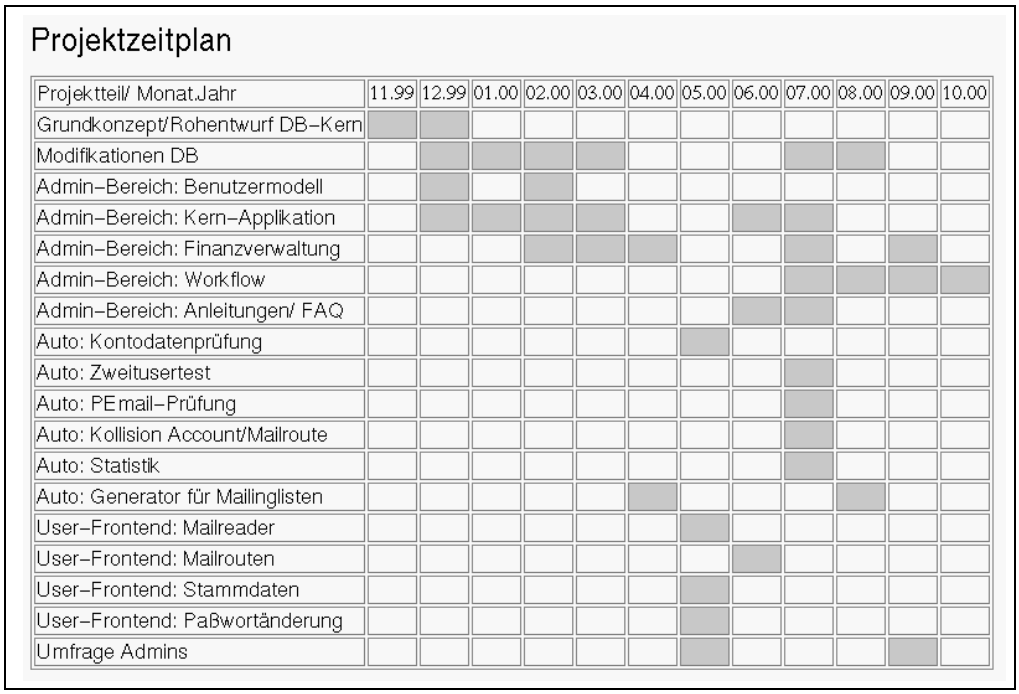

Abbildung 6.1.: zeitlicher Projektverlauf

6.2. Verwaltung des Quellcodes

6.2.1. RCS - Revision Control System

Da das erste Testsystem auf einem lokalen Server von nur zwei Personen erstellt wurde, kam RCS zum Einsatz.

Merkmale von RCS:

- Vergibt neue Versionsnummer für jede Änderung einer Datei.
- Speichert einen Kommentar zu jeder Version und den Autor der Änderungen in einer lokalen Datei („Repository“).
- Bietet die Möglichkeit, alte Versionen wiederherzustellen.
- Warnt den Benutzer, falls die Datei schon von jemand anderem bearbeitet wird.

- Mit „rcsdiff“ können Unterschiede von verschiedenen Versionen einer Datei ermittelt werden .

Folgender Ablauf gilt:

1. Exklusives Schreibrecht für die zu verändernde Datei holen („auschecken“):
   `co -l datei`

2. Datei kann nun bearbeitet werden:
   `vi datei`

3. Nach Erledigung und Speicherung der Datei muss diese wieder in RCS eingepflegt werden („einchecken“):
   `ci -u datei`

4. RCS fragt nun einen Kommentar vom Benutzer ab und legt eine neue Version in seinem „Repository“ an.

Alle vorhandenen Versionen eines Files kann man mittels `rlog datei` anschauen, eine ältere Version kann via `co -l -r1.1 datei` aus dem System geladen werden, in diesem Beispiel die Version 1.1.

Durch Einfügen der Markierung `$Id$` in den Quellcode wird das RCS System aufgefordert, diese Markierung entsprechend zu expandieren nach

```
$ Id: datei 1.01 2000/10/08 10:38:30 user Exp$
```

Dies ist sinnvoll, da anschließend alle relevanten Informationen direkt im Quellcode stehen.

6.2.2. CVS - Concurrent Versions System

Für die Zukunft, insbesondere wenn das System von zusätzlichen Personen verwaltet und weiterentwickelt wird, bietet sich CVS an, das einige Erweiterungen zu RCS bietet:

- CVS verwaltet zusammenhängende Projekte einfacher (rekursives Laden der Dateien aus dem „Repository“ eines externen Servers), ist also direkt netzwerkfähig.

- Der Sperrmechanismus („Locking“) entfällt, mehrere Personen können eine Datei gleichzeitig bearbeiten, welches die Teamfähigkeit erhöht.
  Dies wird durch einen „automatic merge“ bewerkstelligt, der aber nur funktioniert, wenn die Datei an unterschiedlichen Stellen verändert wurde, sonst ist ein manueller Eingriff nötig.

Aus [Ratsch2000], Kapitel 5.

6.3. Umfrage unter den Administratoren

6.3.1. Umfrage zum GUI

Der originale Fragebogen, welcher per Email an die 16 Administratoren verschickt wurde, findet sich im Anhang B. Die Zahlen in den eckigen Klammern kennzeichnen die Anzahl der gegebenen Antworten, die Umfrage wurde zweimal durchgeführt, das erste Mal im Mai 2000 (10 Rückläufer, erste Klammer) und dann im September 2000 (9 Rückläufer, zweite Klammer).

Die numerische Auswertung:

```
Frage 1:
  Stört dich das selbstgenerierte SSL-Zertifikat?
  [2] [0] Ja
  [8] [9] Nein

Frage 2:
  Ist der SSL-Zugang notwendig oder würde auch HTTP ausreichen?
  [10] [8] SSL notwendig
   [0] [1] SSL nicht notwendig

Frage 3:
  Ist die Struktur der Navigation gut und ausgewogen?
  [5] [7] Ja
  [0] [0] Nein
  [5] [1] Teils teils

Frage 4:
  Sind alle dir zugänglichen Teile von Easy-Bawue verständlich
  und gut auffindbar?
  [5] [7] Ja
  [1] [0] Nein
  [3] [2] Teils teils

Frage 5:
  Würdest du "lynx-Konformität" (simpler Browser, ohne Javascript)
  bevorzugen, wenn vorhanden?
  [3] [2] Ja
  [6] [7] Nein
  [1] [0] Teils teils

Frage 6:
  Würdest du gerne direkte Bookmarks setzen können, anstatt die
  Navigation zu verwenden?
  [3] [3] Ja
  [6] [3] Nein
```

```
  [1] [0] ist mir egal, brauche ich nicht
  [0] [3] ungültig

Frage 7:
  Ist die Antwortzeit (Datenbank, Aufbau der Seiten) ausreichend?
  [6] [6] Ja
  [1] [1] Nein
  [2] [2] Teils teils

Frage 8:
  Welches Feature ist dir am wichtigsten?
  [Mai]  User suchen, Workflow, "alles aus einer Hand"
  [Sept] Usersuche, Tickets, Workflow, Bankeinzug

Frage 9:
  Welches Feature erscheint dir am unwichtigsten?
  [Mai]  keine klaren Antworten
  [Sept] Logout

Frage 10:
  Gibt es etwas, das du vermisst und das nicht bereits geplant ist?
  [Mai]  Querlinks, Uebersichtsseiten; gleichzeitiges Bearbeiten von
         mehreren User in einem Fenster (cut+paste...); Speichern aller
         Eintraege ueber nur einen Knopf, anstatt mehrere, d.h. eine
         je Bereich; ein "Ticket-System" im Easy, vorzugsweise auf der
         Startseite; Eine Maske zum Selbstbasteln von Queries
         wäre vielleicht nett
  [Sept] IP-Verwaltung, Kopplung Userdatensicht und Benutzerrechte
         (Beispiel Kassendaten für Schatzmeister in den Userdaten
         sichtbar), getrenntes Kennwort für verschiedene Server

Platz für generelle Anmerkungen:
  [Mai]  - Ich habe erst 3* damit gearbeitet, werde aber bald mehr
           dazu schreiben koennen. Momentan Zeitmangel...
         - Schön wären evtl noch mehr technische Infos, was genau ein
           neues Feature macht. Nicht einfach, "das ändert das Mailrouting",
           sondern "die Änderung wird in /etc/postfix/virtual.cf bereits
           als gehashte Datei eingetragen etc..."
  [Sept] - es ist schwer easy zu erlernen ohne die prozesse
           (die arbeitsprinzipien, nicht die system-prozesse) zu kennen.
         - ist rundum klasse geworden und dank der ganzen neuen
           automatischen Perlskripte ne grosse Erleichterung!
           schönes Tool, mit dem sich gut arbeiten lässt.
         - Hervorragendes System, dank meiner Flatrate auch von hier
           aus ohne dauerndes "AufdieUhrgucken" bedienbar.
```

Auswertung der beiden Befragungen im Vergleich zueinander:

1. Das „selbstgenerierte“ Zertifikat stört niemanden, jedoch wäre ein „echtes“ Zertifikat besser. Hier gab es auch den Hinweis, dass eine Prüfsumme zur Zertifikatüberprüfung gut wäre.

2. Der Zugang allein über SSL ist wegen der Sensibilität der Daten unabdingbar.

3. Die Navigationsstruktur hat sich im Laufe der Zeit verbessert und ist mittlerweile klar und ausgewogen.

4. Die Auffindbarkeit und Verständlichkeit der Funktionen hat sich erhöht.

5. Der Wunsch nach Lynx-Konformität hat abgenommen.

6. Die Vorliebe nach „direkten“ Bookmarks ist ungebrochen, vielleicht sollte man den „Navigations-Zwang“ überdenken.

7. Die Antwortzeit ist unverändert gut.

8. Die wichtigste Funktion ist nach wie vor die Suchen- und Ändern-Funktion der Userdaten. Seitdem der Workflow mit seinen Tickets in Betrieb ist, wird auch er als wichtigste Funktion genannt.

9. Weitere Wünsche für die Zukunft, wie Verwaltung der IP-Adressen und der Nameserverdaten, gibt es ebenfalls.

Anmerkung: Die Rücklaufquote von durchschnittlich ca. 55% (10:6 bzw. 9:8) ist eher gering ausgefallen, lässt sich aber damit erklären, dass nicht alle eingetragenen Administratoren das Tool wirklich nutzen und aus diesem Grund keine Antworten gaben. Insofern machen die Ergebnisse trotzdem Sinn, da sie die aktiven Nutzer des System widerspiegeln.

6.3.2. Abschließende Befragung zum Gesamtsystem

Abschließend wurden Ende September 2000 alle 16 Administratoren noch zur einer Notenvergabe nach Schulnoten (1 - sehr gut bis 6 - mangelhaft) aufgefordert. Es gab diesmal 9 Rückläufer, von denen zwei jeweils eine Frage nicht beantworteten.

Folgende Noten wurden in der vergeben (Fragebogen siehe Anhang B):

frage	arith. mittel	min	max	anzahl d. antworten
1	2.6	2.0	4.5	9
2	1.9	1.0	3.0	9
3	1.4	1.0	2.0	9
4	2.1	1.0	3.0	9
5	2.5	1.0	3.0	8
6	2.3	1.0	4.0	9
7	2.0	1.0	4.0	9
8	2.0	1.0	3.0	9
9	1.8	1.0	2.0	8

Gesamtnote: 2.1 (gut)

Analyse:

- Am besten kommen die Zweckerfüllung (Frage 3, Durchschnittsnote 1.4), die Updates (Frage 9, Durchschnittsnote 1.8) und Benutzbarkeit (Frage 2, Durchschnittsnote 1.9) weg.
- Die Erlernbarkeit (Frage 1, Durchschnittsnote 2.6) kommt zusammen mit der Funktionsweise (Frage 5, Durchschnittsnote 2.5) gut weg. Der Ausreißer mit Note 4-5 bei der Erlernbarkeit bezieht sich sicher auf die Komplexität des Systems und daher die schwere Erlernbarkeit (insbesondere bei seltener Verwendung). Hier müssen weitere Hilfestellungen geleistet werden. In Form der integrierten Online-Hilfe und kleinen Workshops ist dies in der Zwischenzeit bereits geschehen. Besonders bei Neulingen ist darauf zu achten, dass diese gut und präzise eingelernt werden.
- Alle restlichen Fragen wurden im Schnitt mit „gut“ benotet, was auf den Erfolg und insbesondere die Verbesserung im Hinblick auf die alte Vereinsdatenbank hindeutet.
- Optimierungen könnten bei der Ladezeit und Darstellung von großen Tabellen im Browser vorgenommen werden.

7. Schlussbemerkungen und Ausblick

Mit dieser Diplomarbeit ist eine erfolgreiche Webapplikation geschaffen worden, die insbesondere den Administratoren des Vereins BaWue-Net e.V. bei ihrer täglichen Arbeit eine große Hilfe ist.

Folgende Erfahrungswerte haben sich herausgebildet:

- In einer neuen Version oder beim nächsten größeren Update der Applikation könnte man einen Wechsel der Datenbank anstreben. MySQL ist sehr performant, läßt allerdings keine „constraints“ zu, welche an einigen Stellen sehr hilfreich wären und einige Prüfprogramme obsolet machen würden.
- Die Mischung von HTML und PHP-Syntax in PHP-Dateien ist zwar praktisch und schnell für kleine Projekte, allerdings ist diese Lösung prinzipiell unsauber und sollte daher für große Projekte nur mit Bedacht gewählt werden.
- Bei künftigen Entwicklungen sollte das sprachliche Durcheinander von Deutsch und Englisch bei den Tabellen- und Feldbezeichnern vermieden werden.
- Noch mehr Zeit bei der Konzeptentwicklung würde es einfacher machen, Funktionen vor der Implementierung präzise zu planen. Bei diesem Projekt stand allerdings ein gewisser Zeitdruck im Hintergrund, weshalb einige der Funktionen „on-the-fly“ entworfen wurden.
- Die Befragung und Berücksichtigung der Meinung der Benutzer während der gesamten Entwicklungsphase hat zu positiven Effekten geführt, nur so konnte die Navigation verbessert und die neue Funktionalität des Workflow zum Leben erweckt werden.

Für die künftige Weiterentwicklung stehen bereits einige Funktionswünsche im Raum. Der gesamte DNS-Bereich des BaWue-Net wird beispielsweise noch manuell gepflegt und es wäre sinnvoll, diesen Bereich auch über `easy.bawue` abzudecken. Desweiteren wäre ein „echtes“ SSL-Zertifikat für die Systeme `easy.bawue` und `my.bawue` sinnvoll und sollte, sofern finanzierbar, erworben werden.

Literaturverzeichnis

[LHG2000] Anonymous: *Linux Hacker's Guide*; Markt & Technik Verlag, 2000. ISBN 3-8272-5622-4

[Date1990] Date, C.J.: *An introduction to Database Systems*; Addison-Wesley, 1990. ISBN 0-201-52878-9

[Friedl1997] Friedl, Jeffrey E.F.: *Mastering Regular Expressions*; O'Reilly, 1997. ISBN 1-56592-257-3

[Gilly1994] Gilly, Daniel and the staff of O'Reilly & Associates, Inc: *UNIX in a Nutshell*; O'Reilly & Associates, 1994. ISBN 1-56592-001-5

[Hunt2000] Hunt, Craig: *TCP/IP Netzwerk-Administration*; O'Reilly, 2000. ISBN 3-89721-110-6

[KueMi2000] Kühn, Christine; Mintert, Stefan: *JavaScript revisited*; ix Magazin für professionelle Informationstechnik, Ausgabe Oktober 2000. Seite 34ff.

[Medinets2000] Medinets, David: *PHP3 Programming Browser-Based Applications*; McGraw-Hill, 2000. ISBN 0-07-135342-9

[Ratsch2000] Ratschiller, Tobias; Gerken, Till: *Web Application Development with PHP 4.0*; New Riders, 2000. ISBN 0-7357-0997-1

[SaferNet1998] Schmeh, Klaus: *Safer Net: Kryptografie im Internet und Intranet*; dpunkt-Verlag, 1998. ISBN 3-932588-23-1

[SiSpainPat1999] Siever, Ellen; Spainhour, Stephen; Patwardhan, Nathan: *Perl in a Nutshell*; O'Reilly & Associates, 1999. ISBN 1-565-92286-7

[Siever1999] Sievers, Ellen: *Linux in a Nutshell*; O'Reilly, 1999. ISBN 3-89721-116-5

[SnoMue2000] Snoopy; Müller, Martin: *Open Source kurz & gut*; O'Reilly, 1999. ISBN 3-89721-222-6

[Tanenbaum1996] Tanenbaum, Andrew S.: *Computer Networks*; Prentice-Hall, 1996. ISBN 0-13-394248-1

[WaSch1992] Wall, Larry; Schwartz, Randal L.: *Programming Perl*; O'Reilly & Associates, 1992. ISBN 0-937175-64-1

[YaReKi1999] Yarger, Randy Jay; Reese, George & King, Tim: *MySQL & mSQL*; O'Reilly & Associates, 1999. ISBN 1-56592-434-7

Literaturverzeichnis

[Apache] Apache Webserver:
`http://www.apache.org/`

[dtaus] Datenträgeraustausch-Format:
`http://infodrom.north.de/dtaus/dtaus.html`

[Joust] The Outline Navigation System in JavaScript:
`http://www.alchemy-computing.co.uk/joust/`

[linux.org] Linux.org:
`http://www.linux.org/info/`

[Mailman] The GNU Mailing List Manager:
`http://www.list.org/`

[MSbug] Sicherheitslücke im MS Internet Explorer:
`http://www.heise.de/newsticker/data/pab-27.09.00-000/`

[mSQL] mSQL-Homepage:
`http://www.hughes.com.au/products/msql/`

[MySQL] MySQL-Homepage:
`http://www.mysql.com/`

[Perl] Perl Homepage:
`http://www.perl.com/`

[PHP] Hypertext Preprocessor:
`http://www.php.net/`

[RFC2616] RFC 2616 zu HTTP 1.1:
`ftp://ftp.isi.edu/in-notes/rfc2616.txt`

[RFC2716] RFC 2716 zu HTTP Basic Authentication:
`ftp://ftp.isi.edu/in-notes/rfc2617.txt`

[SQL Reference] SQL reference:
`http://www.contrib.andrew.cmu.edu/~shadow/sql.html`

[SSL] Netscape SSL:
`http://www.netscape.com/eng/ssl3/`

[w3c] Homepage des World Wide Web Consortiums:
`http://www.w3.org/`

Abbildungsverzeichnis

A. Glossar

Apache Ein ▷Webserver, der auf ▷Open Source beruht.

Bookmark Die im Browser gespeicherte ▷URL einer Webseite, die für einen schnellen, erneuten Zugriff in einer speziellen Liste abgelegt ist und vom User selbst verwaltetet werden kann.

Browser Ein Client-Programm für das „Stöbern“ (englisch: to browse) im Web. Versteht ▷HTML, meist auch ▷Javascript und kann ▷Java-Applets ausführen. Es gibt verschiedene Hersteller, z.B. Netscape, Microsoft und Opera.

Bug Ein Fehler in einem Computerprogramm.

Bugfixing Das Beheben eines ▷Bugs.

CGI Common Gateway Interface, Schnittstelle zwischen Webseiten und Programmen. Eine typische Funktion ist das Übertragen von Daten, die ein Benutzer im ▷Browser eingegeben hat, in eine Datenbank.

Client Ein Computer oder Programm, das die Dienste eines anderen Rechners (▷Servers) in Anspruch nimmt.

Cronjob Auf UNIX-Systemen können damit Zeitvorgaben für den automatischen Start von Programmen gemacht werden.

CVS Abkürzung für Concurrent Versions System. Erweitertes ▷RCS-System.

DAEMON Abkürzung für Disk And Execution MONitor. Ein Server-Programm, das nur auf Anfrage eines Clients aktiviert wird und ansonsten „schläft“.

DBI Abkürzung für DataBase Interface. Schnittstelle einer Programmiersprache, die Funktionen zum Zugriff auf Datenbanken bereitstellt.

DENIC Abkürzung für DEutsches Network Information Center. Organisation zur Verwaltung der ▷Domains für Deutschland (DE).

DES Abkürzung für Data Encryption Standard. Serienmäßiges Verschlüsselungsverfahren für Kennworte etc. auf UNIX-Betriebssystemen.

Domain Eine Domain ist eine eindeutige textuelle Beschreibung einer Adresse im Internet. Die Domainnamen sind hierarchisch von rechts nach links gegliedert. Der ganz rechte Teil nach dem Punkt heisst Top Level Domain, der davor Second Level Domain oder einfach Domain. Alle weiteren Namensteile links davon sind jeweils Sub-Domains.

Domainhosting Das Unterbringen einer ▷Domain auf einem dafür eingerichteten ▷Server.

DHTML Abkürzung für Dynamic HTML. Erweiterung von ▷HTML mittels ▷Javascript und Stylesheets (CSS).

Email Abkürzung für Electronic Mail. Ein Dienst des Internet, der für das Empfangen und Versenden von elektronischen Nachrichten zuständig ist.

FAQ Abkürzung für Frequently Asked Questions („häufig gestellte Fragen“). Liste, in der zu einem Thema häufig gestellte Fragen und deren Antworten aufgelistet sind.

Frame „Rahmen“, Bestandteil der ▷HTML-Definition. Mit Frames kann der Anzeigebereich des ▷Browsers in verschiedene, frei definierbare Segmente aufgeteilt werden. Jedes Segment kann eigene Inhalte enthalten.

Frontend Der Teil des Programmes, mit dem der Endbenutzer in Berührung kommt. Siehe auch ▷GUI.

FTP Abkürzung für File Transfer Protocol. Ein Dienst im Internet und Protokoll, das dazu dient, Dateien zwischen zwei Rechnern auszutauschen.

GUI Abkürzung für Graphical User Interface („grafische Benutzeroberfläche“). Ein grafisch (nicht textuell) gestaltetes ▷Frontend.

HTML Abkürzung für Hypertext Markup Language, die Seitenbeschreibungssprache des Web. Über HTML wird die logische Struktur eines Dokumentes festgelegt, nicht jedoch das genaue Layout. Für das Layout ist der ▷Browser zuständig. HTML basiert auf SMGL (Standard Generalized Markup Language), einer ISO-Norm zur Definition von strukturierten Datentypen.

HTTP Abkürzung für Hypertext Transfer Protocol. HTTP regelt die Kommunikation zwischen ▷Browser und ▷Webserver und wird im Web als Übertragungsprotokoll verwendet.

Icon Piktogramm, kleines Bild.

Internet Weltweiter Verbund von Computern, die alle über das ▷TCP/IP Protokoll kommunizieren. Das Internet bietet vielfältige Dienste (▷FTP, ▷Email, ▷Web) an.

Javascript Eine von der Firma Netscape entwickelte, in HTML eingebettete Programmiersprache. Lehnt sich in ihrer Syntax an ▷Java an.

Java Eine von der Firma SUN entwickelte objektorientierte Programmiersprache.

Java-Applet Ein auf den ▷Browser zugeschnittenes ▷Java-Programm.

Linux Ein ▷Open Source Betriebsystem, das eine Variante von UNIX darstellt.

LAMP Abkürzung für Linux-Apache-MySQL-PHP/Perl. LAMP steht für eine beliebte Serverkonfiguration, die vollständig aus ▷Open Source-Komponenten besteht.

mSQL Abkürzung für Mini SQL. Relationales Datenbanksystem von Hughes Technologies.

MySQL Relationales Datenbanksystem der Firma T.c.X. DataConsultAB. Das Lizenzmodell ist ▷Open Source.

Open Source Eine Softwarelizenz, die den Quellcode eines Programmes mitliefert und ihn zur freien Verfügung stellt.

Perl Abkürzung für Practical Extraction and Report Language. Eine ▷Open Source-Programmiersprache, die es für viele Plattformen gibt und gerne im Bereich ▷CGI eingesetzt wird.

PGP Abkürzung für Pretty Good Privacy. Ein asymmetrisches Verschlüsselungsverfahren, das insbesondere für Email eingesetzt wird.

PHP Abkürzung für PHP: Hypertext Processor. In ▷HTML eingebettete Skriptsprache. Das Lizenzmodell ist ▷Open Source.

Public Key Die englische Bezeichnung für den Öffentlichen Schlüssel in einem asymmetrischen Verschlüsselungsverfahren. Der geheime, persönliche Schlüssel wird „Private Key“ genannt.

RCS Abkürzung für Revision Control System. RCS ist ein Programm zur Verwaltung von Änderungen in Dateien.

RIPE Abkürzung für Réseaux IP Européens. Organisation zur Koordination des europäischen Internetverkehrs.

RSA Bezeichnung für ein von Rivest, Shamir und Adleman erfundenes asymmetrisches Kryptografieverfahren.

Server Ein Rechner, der Dienste für andere Rechner (▷Clients) bereitstellt.

SSL Abkürzung für Secure Socket Layer. Ein von der Firma Netscape entwickeltes Verschlüsselungsverfahren.

SQL Abkürzung für Structured Query Language. Eine standardisierte Abfragesprache für relationale Datenbanksysteme.

Tag Die englische Bezeichnung für „Markierung“ oder „Kennzeichnung“ eines bestimmten Wortes. Im Zusammenhang mit HTML ist ein einzelnes HTML-Kommando gemeint.

Template Die englische Bezeichnung für „Schablone“, im Computerbereich oft ein vorgefertigtes Programm.

TCP/IP Abkürzung für Transmission Control Protocol/Internet Protocol. Das Übertragungsprotokoll, das den Datenverkehr innerhalb des ▷Internet regelt.

Tool Englisch für: Werkzeug. Computerprogramm, das einem als „hilfreiches“ Werkzeug dient.

URL/URI Abkürzung für Uniform Resource Locator bzw. Identifier. Dabei handelt es sich um Zeichenketten, die eine Quelle im ▷Web identifizieren, zum Beispiel: Dokumente, Dateien, Dienste, ▷Email oder anderes.

Web Ein Dienst des ▷Internet, der auf Hypertext basiert und gerne für das Anbieten von multimedialen Inhalten (Text, Bild, Ton, Video) eingesetzt wird. Die Daten werden auf einem ▷Server gespeichert, die der Benutzer mit einem ▷Browser abrufen kann.

Web-Client siehe ▷Browser.

Webserver Ein ▷Server, der Webseiten bereitstellt.

Workflow Englisch für Arbeitsablauf. Im Sinne dieser Arbeit ist der Workflow ein Programm, das den Arbeitsablauf in bestimmen Bereichen unterstützt und erleichert.

WWW siehe ▷Web.

XML Abkürzung für eXtended Markup Language. Erweiterung des ▷HTML-Standards.

B. Fragebögen

B.1. Benotungsfragebogen

Noten-Fragebogen für die Nutzer des Admin-Tools. Verschickt am 30. September 2000 an die 16 admininistrativen Nutzer des Systems. [1]

```
Notenvergabe für Easy-BaWue. Schulnoten-System (1 = sehr gut, 6 = mangelhaft).

1) Erlernbarkeit
   Note:

2) Benutzbarkeit insgesamt
   Note:

3) Zweckerfüllung
   Note:

4) Selbsterklärend, Hilfesystem
   Note:

5) Funktionsweise, Auffindbarkeit von Funktionen
   Note:

6) Geschwindigkeit Applikation, Download
   Note:

7) Geschwindigkeit Browser-Darstellung
   Note:

8) Navigation
   Note:

9) Veränderungen, die im Laufe der Zeit eingebaut wurden
   Note:

Anmerkungen:
```

[1] Select-Statement für eine Liste der aktiven DB-User: `SELECT Kontakt.Adresse FROM Benutzer, Kontakt WHERE Benutzer.DBRechte != '' AND Benutzer.ID = Kontakt.BenutzerID AND Kontakt.Typ = 'PEmail'`

B.2. Umfragebogen

Fragebogen für die Nutzer des Admin-Tools. Verschickt am 19. Mai 2000 und am 30. September 2000 an 16 admininistrative Nutzer des Systems. [2]

```
Frage 1:
  Stört dich das selbstgenerierte SSL-Zertifikat?
  [ ] Ja
  [ ] Nein

Frage 2:
  Ist der SSL-Zugang notwendig oder würde auch HTTP ausreichen?
  [ ] SSL notwendig
      weil: _________________
  [ ] SSL nicht notwendig
      weil: _________________

Frage 3:
  Ist die Struktur der Navigation gut und ausgewogen?
  [ ] Ja
  [ ] Nein
  [ ] Teils teils
      Anmerkungen: ____________________________

Frage 4:
  Sind alle dir zugänglichen Teile von Easy-Bawue verständlich
  und gut auffindbar?
  [ ] Ja
  [ ] Nein
  [ ] Teils teils
      Anmerkungen: ____________________________

Frage 5:
  Würdest du "lynx-Konformität" (simpler Browser, ohne Javascript)
  bevorzugen, wenn vorhanden?
  [ ] Ja
  [ ] Nein

Frage 6:
  Würdest du gerne direkte Bookmarks setzen können, anstatt die
  Navigation zu verwenden?
```

[2] Auswertung und Berechnung der Durchschnittsnoten mittels folgendem Select-Statement:
`SELECT frage, round(avg(antwort),1) AS mittel, round(min(antwort),1) AS min, round(max(antwort),1) AS max, count(antwort) AS antworten FROM easyumfrage3 GROUP BY frage ORDER BY frage`
Die Gesamtnote berechnet sich wie folgt:
`SELECT round(avg(antwort),1) AS gesamtnote FROM easyumfrage3`

```
  [ ] Ja
  [ ] Nein

Frage 7:
  Ist die Antwortzeit (Datenbank, Aufbau der Seiten) ausreichend?
  [ ] Ja
  [ ] Nein
      Woran könnte das liegen? _________________
  [ ] Teils teils
      Anmerkungen: _____________________________

Frage 8:
  Welches Feature ist dir am wichtigsten?
  Antwort: ___________________________________

Frage 9:
  Welches Feature erscheint dir am unwichtigsten?
  Antwort: ___________________________________

Frage 10:
  Gibt es etwas, das du vermisst und das nicht bereits geplant ist?
  Antwort: ___________________________________

Platz für generelle Anmerkungen:

  ____________________________
```

B.3. Navigation vorher/nachher

Siehe Screenshots in Abbildung B.1 und B.2.

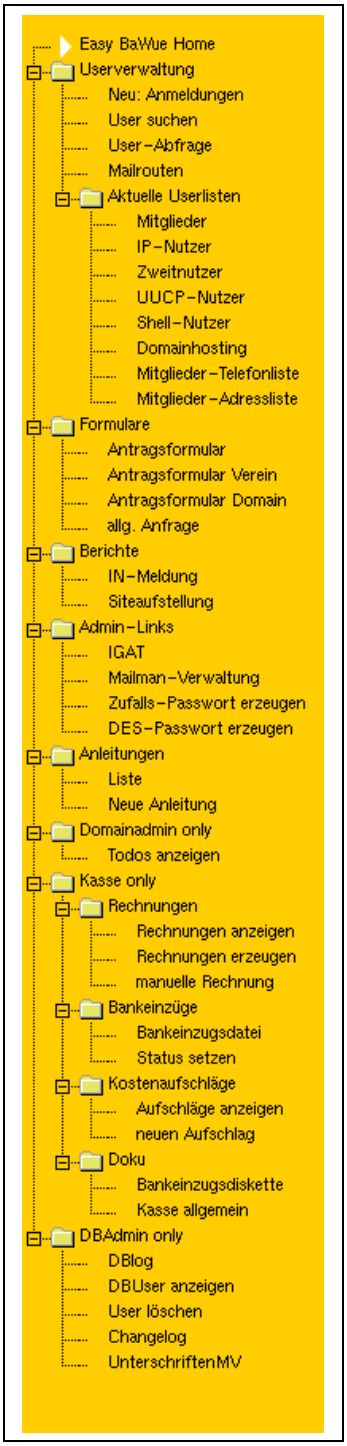

Abbildung B.1.: easy.bawue: Navigation vor Umfrage

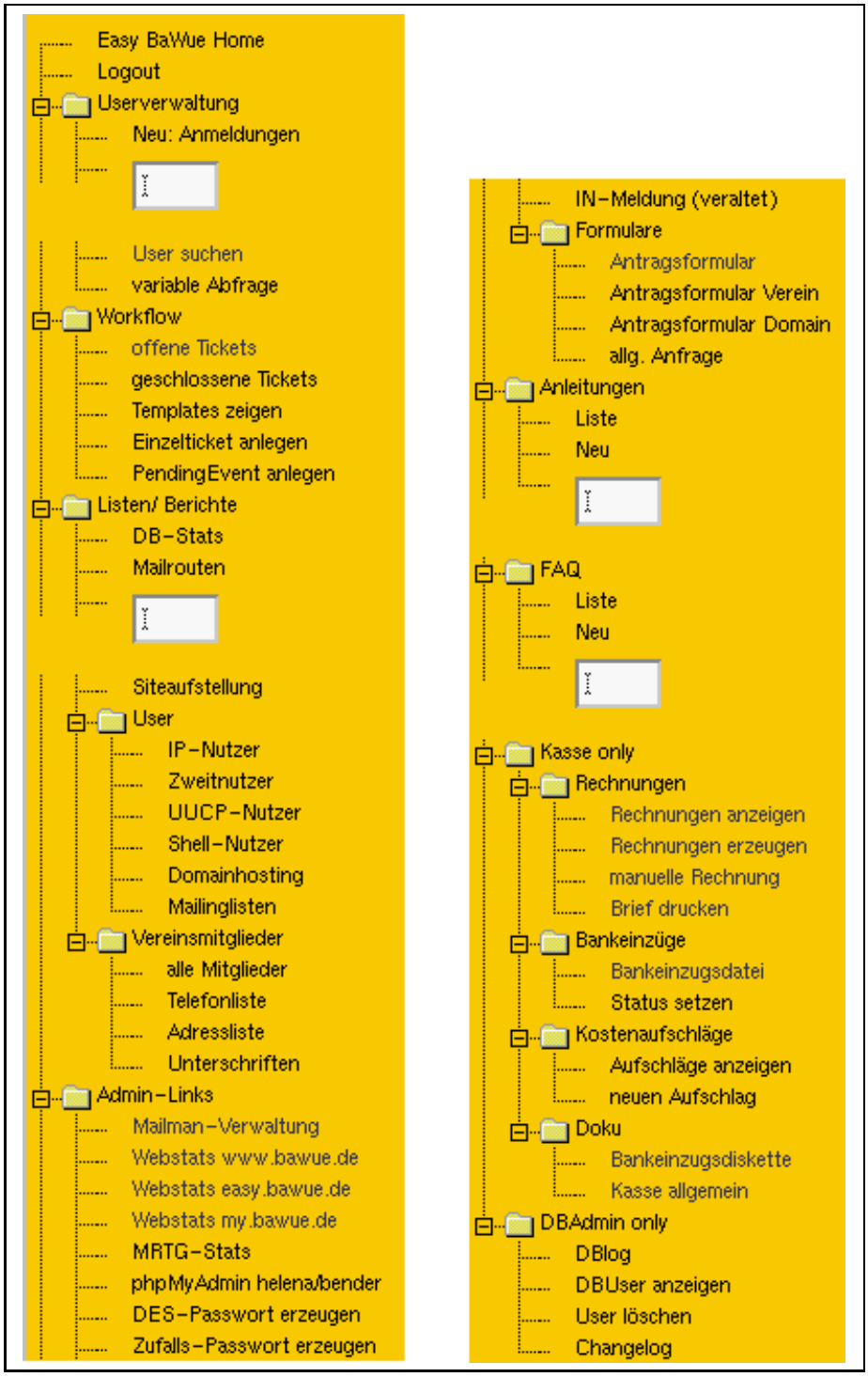

Abbildung B.2.: easy.bawue: Navigation nach Umfrage

C. Lizenzbestimmungen

C.1. GNU Public License

Ein Auszug aus der GNU Public License, der gesamte Text kann unter `http://www.gnu.org/copyleft/gpl.txt` abgerufen werden.

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble
...

                    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License.  The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language.  (Hereinafter, translation is included without limitation in
the term "modification".)  Each licensee is addressed as "you".

...

                            NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.  EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.  SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
```

INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

C.2. Alte MySQL Lizenz

Das Lizenzmodell war bisher recht komplex und wurde vom Hersteller wie folgt angeben: (nur auszugsweise, weitere Details siehe MySQL Homepage [MySQL])

The MySQL server license for non Microsoft operating systems.

- For normal internal use, MySQL generally costs nothing. You do not have to pay us if you do not want to.
- A license is required if:
  1. You sell the MySQL server directly or as a part of another product or service
  2. You charge for installing and maintaining a MySQL server at some client site
  3. You include MySQL in a distribution that is non redistributable and you charge for some part of that distribution
- You do not need a license to include client code in commercial programs.
- If you use MySQL in a commercial context such that you profit by its use, we ask that you further the development of MySQL by purchasing some level of support.

De facto sind nur in zwei Fällen Lizenzgebühren zu entrichten gewesen:

1. bei Verkauf des MySQL-Servers als Teil eines anderen Produktes
2. für die Variante für die Windows-Betriebssysteme von Microsoft

D. Ausführliche Programm-Beispiele

D.1. Admin-Bereich

D.1.1. bawue_db_connect.php3

Diese Datei wird via include in alle PHP-Skripte eingebettet und dient zum Verbindungsaufbau zur Datenbank. Diese PHP-Skript wurde von Daniela Gehle entworfen.

```
<?
#-----------------------------------------------------------------#
# bawue_db_connect.php3
# dg/12.12.1999
#
# Macht den Connect auf die BaWue-Net-DB
# und gibt ggf. eine Fehlermeldung aus
#-----------------------------------------------------------------#

$dbconn = mysql_connect("localhost","bawue","passwd")
             OR DIE ("<p><font color=red><blink>ERROR:</blink>
can't connect to database bawue on localhost -
                        please contact the admin of the site!</font>");

MYSQL_SELECT_DB("bawue");
?>
```

D.1.2. show_open_tickets.php

Wird im Workflow verwendet, um die offenen Tickets im Browser anzuzeigen.

```
<?
# ------------------------------------------------------------------
# $Id: show_open_tickets.php,v 1.10 2000/10/08 14:14:08 sven Exp $
#
# Funktion: Zeigt alle offenen Tickets des Workflow an
# ------------------------------------------------------------------

#----- Rechte fuer diese Seite -------
$REQUIRE[0] = "admin";
$REQUIRE[1] = "dbadmin";
$REQUIRE[2] = "kasse";

#---------- Authentisierung ----------

include("../include/bawue_db_authenticate.php3");
```

```
include("../include/bawue_db_connect.php3");

#---------- Funktionen ----------

function get_userinfo($uid)
{
  # hole autorname
  $query_string1 = 'SELECT Vorname,Name FROM Benutzer where id='.$uid;
  $result1 = mysql_db_query("bawue",$query_string1);
  $rows1 = mysql_num_rows($result1);
  $vorname=mysql_result($result1,0,"Vorname");
  $name=mysql_result($result1,0,"Name");
  $ret="$vorname $name";
  return ($ret);
}

#---------- In die Datenbank workflow schauen ----------

  $today=date("YmdHs");
  MYSQL_SELECT_DB("workflow");

  if ($keyword)
  {
    $where="WHERE  tickets.teamid = team.id
            AND timestampDONE = 0
            AND tickets.description LIKE \"%$keyword%\"";
  }
  else
  {
    $where=" WHERE tickets.teamid = team.id
             AND timestampDONE = 0";
  }

  $query_string = 'SELECT  * FROM tickets, team '.$where.' ORDER BY timestampTBD';

  $result = mysql_db_query("workflow",$query_string);
  $rows = mysql_num_rows($result);
  echo "<!-- rows=$rows -->\n";

?>

<HTML>
<HEAD>

<!--- HILFE Fenster vorbereiten -->
<script language="javascript">
function hilfe(keyword)
{
  this.kw = "keyword";
  var url='hilfe.php3?keyword=' + keyword;
  myWindow = window.open(url,"hilfefenster","resizable=yes,toolbar=0,location=0,
             directories=0,status=0,menubar=0,scrollbars=1,copyhistory=0,width=340,height=340");
 myWindow.location = url;
}
</script>
```

```
<!--------------------------------->

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

<TITLE>BaWue-Net Workflow offene Tickets</TITLE>
</HEAD>

<BODY BGCOLOR=#FFFFFF>

<font face="Verdana, Arial, Helvetica, Courier">
<form action="<? echo $PHP_SELF ?>">
<font size="+2">Offene Tickets nach Datum</font>
<? if ($keyword)
   {
      echo "/ Stichwort: \"$keyword\"";
   }
?>
/ schnellsuche <input type="text" name="keyword" size="10">
</form>
</font>


<table border="0" cellpadding="4" bgcolor="#FFFFFF">
<tr>
   <td bgcolor="#99CC99">
      <b><a HREF="#" onClick=hilfe("ticket")>Ticket</a></b>
   </td>
   <td bgcolor="#99CC99">
      <b>Beschreibung</b>
   </td>
   <td bgcolor="#99CC99">
      <b><a HREF="#" onClick=hilfe("team")>Team</a></b>
   </td>
   <td bgcolor="#99CC99">
      Zu erledigen bis
   </td>
</tr>

<?
$i       = 0;
$j       = 0;
$eventid = 0;

while($i<$rows):
  $id        = 0;
  $team      = 0;
  $instid    = 0;
  $useridtodo= 0;
  $id        = mysql_result($result,$i,"id");
  $desc      = mysql_result($result,$i,"description");
  $tsTBD     = mysql_result($result,$i,"timestampTBD");
  $tsTBD     = date($tsTBD);
  $cronint   = mysql_result($result,$i,"cronintervall");
  $team      = mysql_result($result,$i,"team.name");
  $teamemail = mysql_result($result,$i,"team.email");
  $instid    = mysql_result($result,$i,"instid");
```

```
  $comment   = mysql_result($result,$i,"comment");
  $useridtodo= mysql_result($result,$i,"useridtodo");

  if ($useridtodo)
  {
    $betroffeneruser = get_userinfo($useridtodo);
  }
  else
  {
    unset ($betroffeneruser);
  }

  echo "<tr>";
  echo "<td><a href=\"ticket-detail.php?id=$id\">#$id</a></td>\n<td>";
  echo "<a href=\"ticket-detail.php?id=$id\">$desc</a><br>";

  if ($betroffeneruser)
  {
     echo "betrifft: <a href=\"../user-aendern.php3?userid=$useridtodo\">";
     echo "$betroffeneruser</a><br>";
  }
  echo "<i>$comment</i></td>\n";
  echo "<td><a href=\"mailto:$teamemail\">$team</a></td>\n";

  if (substr($today,0,8) >= substr($tsTBD,0,8))
  {
    $farbe="red";
  }
  elseif (substr($today,0,8)+1 == substr($tsTBD,0,8))
  {
    $farbe="#FFcccc";
  }
  else
  {
    $farbe="white";
  }

  echo "<td bgcolor=\"$farbe\">".substr($tsTBD, 6, 2).".";
  echo substr($tsTBD, 4, 2).".".substr($tsTBD, 0, 4)."</td>";
  echo "</tr>\n";

  $i++;

endwhile;
?>

<!-- zeilentrenner -->
</table>

<P>
  <? echo $i ?> offene Tickets gefunden.
</P>
</BODY>
</HTML>
```

D.1.3. easystats.pl

Wird täglich via Cronjob aufgerufen und erzeugt statistische Einträge.

```
#!/usr/bin/perl
# $Id: easystats.pl,v 1.5 2000/10/08 14:21:02 sven Exp $
# lese aus statsconfig die selects und
# schreibe ergebnis zurück in stats

print "stats\n";

use Mysql;
$dbh = Mysql->connect('bender.bawue.de','bawuelogs','bawueadm','passwd');

$query="select * from statsconfig";

print "$query\n";

$sth = $dbh->query($query);

while(1)
{
  @out = $sth->fetchrow;
  if (@out[0] != "")
  {
    $dbh2 = Mysql->connect('bender.bawue.de','bawue','bawueadm','passwd');
    $query2=$out[2];
    $sth2 = $dbh2->query($query2);
    @out2 = $sth2->fetchrow;
    print "$out[0]) $out[1]: $out2[0] Einträge ";
    $ts=`date +%Y%m%d%H%M%S`; chop($ts);
    $insert="INSERT into stats (id, ts, nameid, anzahl) VALUES('','$ts',$out[0],$out2[0])";
    $dbh3 = Mysql->connect('bender.bawue.de','bawuelogs','bawueadm','passwd');
    $sth3 = $dbh3->query($insert);
    print "(insert done)\n";
  }
  else
  {
     exit;
  }
}
```

D.2. Automatismen

D.2.1. erzeuge_mliste.pl

```
#!/usr/bin/perl
# $Id: erzeuge_mliste.pl,v 1.4 2000/10/30 11:09:59 sven Exp $
# generiere liste nach schema: vorname nachname <PEmail>
# aus der easy-bawue-db fuer mailman-listen
# wenn kein Parameter übergeben, nimm *all*
# sonst Wert als Primaersite im Select verwenden
# Ausnahmen: ev und domainowner
# Donau = Iller (!)
```

```
if (!@ARGV[0])
{
  # SELECT nur auf alle aktiven user
  # print "select *alle*\n";
  $query="SELECT DISTINCT Kontakt.BenutzerID, Benutzer.Name,
                 Benutzer.Vorname,Kontakt.Adresse
          FROM Benutzer, Benutzer_Dienst, Dienst, Kontakt, Status
          WHERE Benutzer.ID=Benutzer_Dienst.BenutzerID AND
                Benutzer.ID=Kontakt.BenutzerID AND
                Benutzer_Dienst.StatusID = Status.ID AND
                Benutzer_Dienst.DienstID = Dienst.ID AND
                Benutzer_Dienst.Datum > '0001' AND
                Kontakt.Typ='PEmail' AND
                Benutzer_Dienst.Datum < '9999-12-31' AND
                Status.Beschreibung = 'normal'
          ORDER BY Benutzer.Name ";
}
else
{
  if (@ARGV[0] eq "ev")
  {
    # SELECT nur auf Vereinsmitglieder
    #print "ev \n";
    $query="SELECT Kontakt.BenutzerID, Benutzer.Name, Benutzer.Vorname,
                   Kontakt.Adresse
            FROM Benutzer, Benutzer_Dienst, Kontakt
            WHERE Benutzer.ID=Benutzer_Dienst.BenutzerID AND
                  Benutzer.ID=Kontakt.BenutzerID AND Kontakt.Typ='PEmail' AND
                  Benutzer_Dienst.StatusID=20 AND Benutzer_Dienst.DienstID=1
            ORDER BY Benutzer.Name";
  }
  elsif (@ARGV[0] eq "domainowner")
  {
    # SELECT nur auf Domainowner
    $query="SELECT DISTINCT Kontakt.BenutzerID, Benutzer.Name,
                   Benutzer.Vorname,Kontakt.Adresse
            FROM Benutzer, Benutzer_Dienst, Kontakt
            WHERE Benutzer.ID=Benutzer_Dienst.BenutzerID AND
                  Benutzer.ID=Kontakt.BenutzerID AND
                  Kontakt.Typ='PEmail' AND
                  Benutzer_Dienst.StatusID=20 AND
                  Benutzer_Dienst.DienstID=5
            ORDER BY Benutzer.Name";
  }
  else
  {
    # SELECT auf Übergabewert als Primaersite
    $query="SELECT DISTINCT Kontakt.BenutzerID, Benutzer.Name,
                   Benutzer.Vorname,Kontakt.Adresse
            FROM Benutzer, Benutzer_Dienst, Dienst, Kontakt, Status
            WHERE Benutzer.ID=Benutzer_Dienst.BenutzerID AND
                  Benutzer.ID=Kontakt.BenutzerID AND
                  Benutzer_Dienst.StatusID = Status.ID AND
                  Benutzer_Dienst.DienstID = Dienst.ID AND
                  Benutzer_Dienst.Datum > '0001' AND
                  Kontakt.Typ='PEmail' AND
```

```
                Benutzer_Dienst.Datum < '9999-12-31' AND
                Status.Beschreibung = 'normal' AND
                Benutzer.Primaersite='@ARGV[0]'
            ORDER BY Benutzer.Name ";
  }
}

use Mysql;
$dbh = Mysql->connect('localhost','db','user','passwort');
$sth = $dbh->query($query);

while(1)
{
  @out = $sth->fetchrow;
  if (@out[0] != "")
  {
     print "$out[1] $out[2] <$out[3]>\n";
  }
  else
  {
     exit;
  }
}
```

D.3. Benutzer-Frontend

D.3.1. i-header.html

Diese Datei wird via include in alle HTML/PHP-Seiten eingebettet und erzeugt den Titel und die Navigation einer Seite auf my.bawue.

```
<?
  # $Id: i-header.html,v 1.5 2000/10/08 14:28:43 sven Exp $
  # Titel, Navigation für my.bawue.de
  # ------------------------------------------------

  include("includes/bawue_db_authenticate.php3");
  # Anfang Code von Frank Berger
  require 'php3/functions.php3';
  init();
  check_auth($PHP_AUTH_USER,$PHP_AUTH_PW);
  // trigger damit AeroMail nicht nochmal nach U/P fraegt
  $extern_require = 'true';
  // pre-load der config fuer mail/global.inc
  require 'mail/themes/bawue.theme.inc';
  require 'mail/lang/de.lang.inc';
  require 'mail/config.inc';
  // init von imap... definiert bei erfolg u.a. $mailbox
  require 'mail/global.inc';
  # Ende Code von Frank Berger
?>

<html>
<head>
```

```
<meta href="http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<meta name="AUTHOR" content="Sven Dittmar">

<link rel=stylesheet type="text/css" href="includes/styles.css">

<!-- -----------Titel der Seite------------ -->
<title>myBaWue: <?echo $db_Vorname," ",$db_Name?></title>
<!-- -----------Ende---------------------- -->

</head>

<body link="#000000" vlink="#000000" alink="#FFFFFF" bgcolor="#FFFFFF">

<xxtable border="0" width="600" cellspacing="0" cellpadding="4">
<table border="0" width="100%" cellspacing="0" cellpadding="4">
  <tr>
    <td xwidth="100" bgcolor="#6699cc"><a href="https://my.bawue.de"><img
src="_img/bawue-logo-org-108n.gif" width="108" height="56"
        alt="BaWue-Net" border="0"></a></td>

    <td valign="top" align="left" bgcolor="#6699cc"><img
src="_img/trans1.gif" width="1" height="1" alt="." hspace="5" vspace="8"></td>

    <td width="490" bgcolor="#6699cc">



<!-- Anfang Code von Frank Berger -->
<p class="fett">
<!-- -----------Titel der Seite im gelben Balken----------- -->
<a href="/">myBaWue</a> f&uuml;r <?echo $db_Vorname," ",$db_Name?>
<!-- -----------Ende--------------------------------------- -->
</p>
<!-- Ende Code von Frank Berger -->

</td>
  </tr>
  <tr>
    <td width="100" bgcolor="#6699cc"><img src="_img/trans1.gif" width="1" height="1"
    alt="." hspace="10" vspace="1"></td>
    <td width="1" valign="top" align="left"></td>
    <td width="500"></td>
  </tr>
  <tr>
    <td width="100" bgcolor="#6699cc" valign="top" align="left" class="fett">
    <p class="fett">
    <a href="/hinweise.php3"><nobr>Aktuelle Hinweise</nobr></a><br>
    <a href="/logout.php3"><nobr>Logout</nobr></a>
    </p>
    <p class="fett">
    Pers&ouml;nliches<br>
    <img src="_img/black1.gif" width="100" height="2" alt="."><br

<!-- Anfang Code von Frank Berger -->
<!-- dynamic mailbox-check -->
<?
```

```
    if ($mailbox)
    {
      echo "<a href=\"/mail/\">Email</a> <font size=-1>($folder: ";
      echo "imap_num_msg($mailbox) .")</font><br>\n";
    }
?>
<!-- end dynamic mailbox-check -->
<!-- Ende Code von Frank Berger -->

    <a href="/mailkonfig.php3"><nobr>Email-Konfig</nobr></a>
    <a href="/daten.php3"><nobr>Stammdaten</nobr></a><br>
    <a href="/passwort.php3"><nobr>Pa&szlig;wort &auml;ndern</nobr></a>
    </p>
    <p class="fett">
    Sonstiges<br>
    <img src="_img/black1.gif" width="100" height="2" alt="."><br>
    <a href="http://www.bawue.de/" target="bawue"><nobr>Miniumfrage</nobr></a><br>
    <a href="http://www.bawue.de/faq/" target="bawue"><nobr>FAQ</nobr></a><br>
    <a href="http://www.bawue.de/main/news.html" target="bawue"><nobr>News</nobr></a><br>
    <a href="http://www.bawue.de/main/suche.html" target="bawue"><nobr>Suchen</nobr></a><br>
    <a href="http://www.bawue.de/main/kontakt.html" target="bawue"><nobr>Kontakt</nobr></a><br>
    </p>
    </td>
    <td width="1" valign="top" align="left"></td>


<td width="500" valign="top" align="left">
```

D.3.2. edit_mailroute.php3

Ändern einer vorhandenen Mailroute via my.bawue.

```
<?
# $Id: edit_mailroute.php3,v 1.3 2000/10/08 14:33:10 sven Exp $
# Funktion: Ändern einer vorhandenen Mailroute
#
#---------- Authentisierung ----------
  include("i-header.html");
?>

<!---------- Datenbank oeffnen ---------->
<? include("includes/bawue_db_connect.php3") ?>


<?
  echo "<h1>Domainverwaltung: Mailroute &auml;ndern</h1>";


  if (!$routeid)
  {
    echo "<p>Es wurde keine MailrouteID &uuml;bergeben!</p>";
  }
```

```
  else
  {
    echo "\n<!-- query = $query -->\n";
?>
<table border="0">
<tr>
<td>
  <form method="post" action="query_mailroute.php3">
  <input name="action" type="hidden" value="update">
  <input name="virtadr" size="20" type="text" value="<? echo $virtadr ?>"> ->
  <input name="redirect" size="20"  type="text" value="<? echo $redirect ?>">
  <input name="routeid" type="hidden" value="<? echo $routeid ?>">
</td>
<td>
  <input type="submit" value="Speichern">
  </form>
</td>
</tr>
<tr>
<td>
  <form method="post" action="query_mailroute.php3">
  <? echo "$virtadr -> $redirect"; ?>
  <input name="action" type="hidden" value="delete">
  <input name="routeid" type="hidden" value="<? echo $routeid ?>">
</td>
<td>
  <input type="submit" value="Löschen">
  </form>
</td>
</tr>
</table>
<?

  }

  echo "<p>Zur&uuml;ck zur ";
  echo "<a href=\"domainverwaltung.php3\">Domainverwaltung</a>.</p>";
  include("i-footer.html");

?>
```

E. Datenmodell easy.bawue

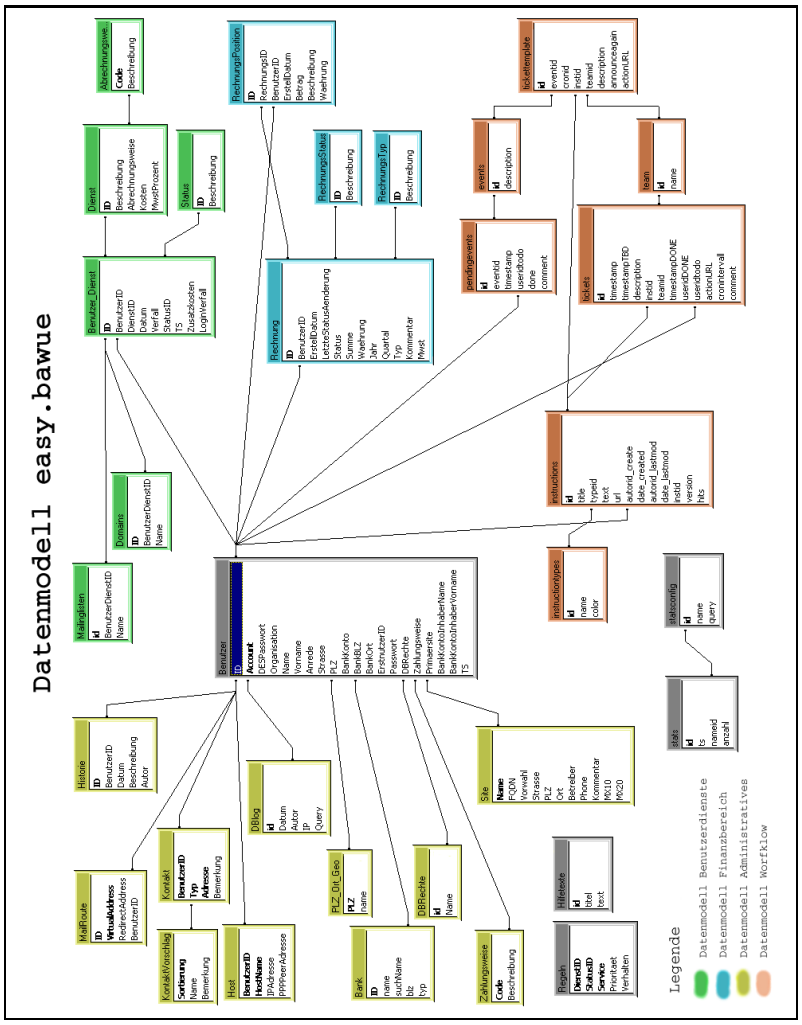

Abbildung E.1.: Datenmodell easy.bawue