Hochschule der Medien
Studiengang Medieninformatik - SS 2006

Datenbankgestütztes Publizieren

G.R.U.N.Z Bandportal

Teilnehmer:
Daniel Glück (15398),
Marcus Himmel (13364),
Daniel Genchev (14474)

Betreuer:
Prof. Martin Goik

Inhaltsverzeichnis

1 Einführung

1.1 Projektbeschreibung

GRUNZ steht für Gemeinnütziger Rock und Nachwuchs Zusammenschluss. Der GRUNZ ist eine Musikerinitiative, die sich um die Bandförderung im Großraum Ludwigsburg kümmert. Das Web-Angebot ist bisher recht eingeschränkt und nicht konkret auf Bands ausgerichtet. Das soll sich ändern. Vor allem sollen die Bands den Content liefern und somit für eine lebendige, dynamische Website sorgen.

Die Band soll im Mittelpunkt stehen und ihr werden gewisse Services zur Verfügung gestellt

- Bandprofil
- News
- Konzerte (RSS Feed)

Die Band soll diese Services auch auf ihrer eigenen Website einbinden können.

1.2 Anmerkungen zur Projektbeschreibung

Die Projektbeschreibung unter 1.1 konnten wir soweit umsetzen. Allerdings haben wir das externe Einbinden der Services in die eigene Website der Band aus Zeitgründen nicht mehr realisiert.

1.3 Projektziele

An dieser stellen möchten wir anmerken das für uns bei diesem Projekt im Vordergrund stand, unsere Erfahrungen mit PHP und MySQL in einem „richtigen“ Projekt zu vertiefen. Wir haben auch bewusst nicht auf ein Framework oder CMS aufgesetzt. Wir wollten im Projekt unseren eigenen Lösungsweg gehen und können nun auch abwägen, ob es besser gewesen wäre komplett auf ein Framework bzw. CMS aufzusetzen.

1.4 Aufgabenteilung

Die Aufgabenteilung kann wie folgt charakterisiert werden

- Daniel Genchev: Entwurf und Umsetzung der Architektur, Programmieren von Modulen
- Daniel Glück: Projektverantwortlicher, Programmieren von Modulen
- Marcus Himmel: Design, Layout, CSS, Admin Bereich

Die Aufgabenteilung hat sich im Laufe des Projekts so ergeben. Wir haben diese nicht von Anfang an strikt so vereinbart.

1.5 Testen des Projekts

Wir haben das Projekt unter https://applic.mi.hdm-stuttgart.de/band abgelegt. Dort kann es getestet werden. Die Administration ist unter https://applic.mi.hdm-stuttgart.de/band/admin zu erreichen. Diese ist natürlich im Echtbetrieb passwortgeschützt (durch .htaccess, was auf dem applic Server leider nicht funktioniert).

Die Feeds funktionieren auf dem applic Server nicht da die Pear Library XML_Tree fehlt. Diese kann bei entsprechenden Rechten mit dem Befehl ***pear install XML_Tree*** nach installiert werden.

2 Architektur

2.1 Eingesetzte Technologien

Die eingesetzten Technologien haben sich einerseits durch den Webserver ergeben auf dem das Bandportal später laufen wird, andererseits durch unsere Projektziele. Der Webserver bietet eine MySQL Datenbank und PHP 5 an. Für die Umsetzung des Portals haben wir demnach mit diesen Technologien gearbeitet.

2.2 Datenbankmodell

Nachfolgend werden wir die vom Bandportal benutzten Tabellen näher erläutern. Im Mittelpunkt des Datenbankmodells steht die Tabelle **user**.

Das ER-Diagramm für die nachfolgenden Tabellen ist im Anhang zu finden.

- **user:** Verwaltung der Profildaten aller registrierten Benutzer wie z.B. Username, Passwort, Vorname, Nachname, Profilbild etc. Des Weiteren wird in dieser Tabelle die Sichtbarkeit der Kontaktdaten gesteuert, d.h. ob andere Benutzer die E-Mail Adresse, Messenger Daten und Adressdaten einsehen dürfen. Über das Feld ***active*** (0 = nicht aktiviert, 1 = aktiviert) wird gesteuert ob das Benutzerkonto aktiv ist, d.h. das sich der User einloggen kann.
- **instrument:** Instrumente die ein User spielen kann
- **user2instrument:** Zuordnung von Instrumenten zu Benutzern (n:m)
- **user2band:** Zuordnung eines Benutzers zu Bands
- **online:** Verwaltung welche Benutzer online sind für die Online-Statistik
- **band**: Hier werden alle Bands und die dazugehörigen Informationen wie Website, Stadt, Bandbild etc. gespeichert. Des Weiteren hat eine Band verschiedene Zustände die über die Felder ***active*** (0 = nicht aktiviert, 1 = aktiviert) und ***granted*** (0 = nicht freigeben, 1 = freigegeben) gesteuert werden.
- **style:** Musikrichtungen die eine Band spielen kann
- **band2style:** Zuordnung von Musikrichtungen zu Bands (n:m)
- **locations:** Veranstaltungsorte für die Konzerte eingetragen werden können.
- **shows:** Hier werden alle eingetragenen Konzerte und die dazu gehörenden Informationen wie Datum, Beginn, Eintritt etc. gespeichert.
- **band2veranstaltung:** Zuordnung einer Band zu einer Veranstaltung
- **news:** Hier werden die News gespeichert die eine Band eingetragen hat

Des Weiteren haben wir zum Aufbau des Menüs und den hinter den Menüpunkten liegenden Aktionen folgende Tabellen benutzt.

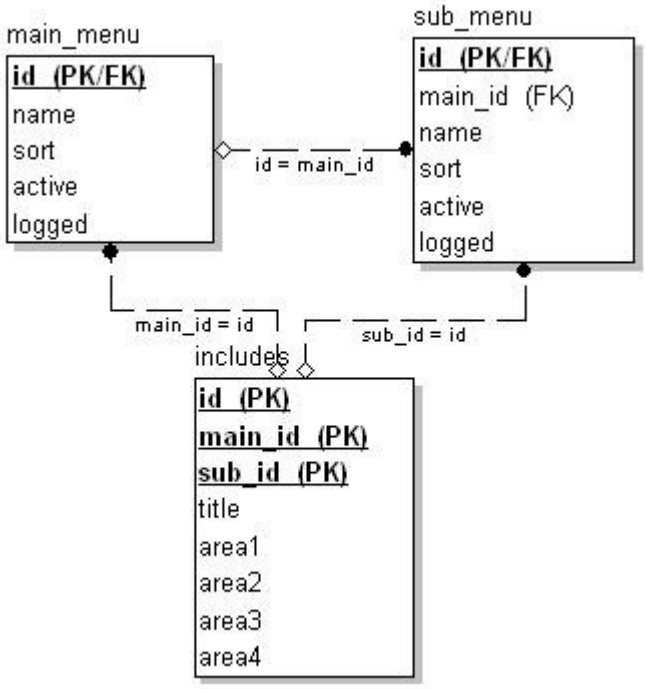

Abbildung 1: ER-Modell Navigationstabellen

Beschreibung der Tabellen

- **main_menu:** Speichern der Hauptmenupunkte
- **sub_menu:** Speichern der Untermenupunkte
- **includes:** Speichern der Dateien die beim Aufruf des entsprechenden Menupunkts angezogen werden (mehr dazu unter 2.3)

Die Tabellen main_menu und sub_menu haben jeweils noch die Felder ***logged*** und ***active***.

Das Feld logged kann folgende drei Werte annehmen

- -1: Menupunkt wird nur nicht eingeloggten Benutzern angezeigt (z.B. login)
- 0: Menupunkt wird immer angezeigt, egal ob eingeloggt oder nicht
- 1: Menupunkt wird nur eingeloggten Benutzern angezeigt (z.B. mygrunz)

Über das Feld active kann gesteuert werden, ob der Menupunkt in der Navigation angezeigt wird oder nicht.

2.3 Seitenaufbau

Das Layout und die komplette Funktionalität des Bandportals laufen über die Datei index.php ab. Nachfolgend werden wir Schritt für Schritt erläutern was beim Aufruf der Datei index.php passiert.

1. Inkludieren der Konfigurationsdatei und Funktionsdatei
2. Sessionverwaltung
3. Aufbau der Datenbankverbindung
4. Unterscheidung ob Besucher oder eingeloggter Benutzer, Anlegen entsprechender Session Objekte
5. Aufbau der Navigation
6. Abhängig von den übergebenen Parametern mm (main menu id) und sm (submenu id) wird der in der Tabelle includes hinterlegte Content angzeigt (siehe 2.3.1)
7. Schließen der Datenbankverbindung

Funktionsdatei
In dieser Datei haben wir Funktionen ausgelagert die in jedem Modul verwendet werden können. Des Weiteren ist hier die bei PHP wichtige Funktion __autoload definiert. Diese Funktion kümmert sich um das Laden von Klassen.

Konfigurationsdatei
Diese Datei dient zur Konfiguration der Laufzeitumgebung des Bandportals. Hier werden z.B. die Einstellungen für die Datenbankverbindung und für Pfade hinterlegt.

Durch die Verwendung der zentralen Datei index.php werden Funktionen, Variablen und Konstanten bereitgestellt, die in jedem Modul benötigt werden. Somit haben wir die Wiederholung von Code in den Modulen vermieden.

2.3.1 Einbinden der Module

Beim Aufbau der Module haben wir die Logik und Anzeige getrennt. Dazu haben wir die zwei Packages php und html verwendet. In diesen Packages wurden jeweils zwei gleichnamige Dateien angelegt.

Die Datei im php Package kümmert sich um die Logik, d.h. das Instanziieren von Objekten, Datenbankabfragen, Auswerten der Benutzereingaben usw. Im html package liegt die Datei die sich um die Präsentation der Daten kümmert.

Welches Modul beim Aufruf der index.php angezeigt werden soll ist in der includes Tabelle hinterlegt. Über die Übergabeparameter mm und sm an index.php wird der entsprechende Dateiname aus der Datenbank gelesen und die Dateien aus dem php und html Packages in index.php inkludiert.

Durch das Grundgerüst, dass durch die index.php Datei geschaffen wird, standen uns beim Einbinden der Module natürlich die global definierten Funktionen, Konstanten und Sessionobjekte für Datenbankaktionen und Benutzerverwaltung zur Verfügung.

Mit diesem Konzept konnten wir aber genau so gut nur eine Datei im package html anlegen die statischen Inhalt hat. Es wird dann zwar versucht auch die Datei aus dem php Package zu inkludieren. Ist diese nicht vorhanden wird einfach nichts inkludiert.

Dieses Konzept durch index.php und einzelne Module ermöglichte uns mehr Übersichtlichkeit und besseres Verständnis des Codes. Des Weiteren konnte sich so der Designer nur mit den Dateien im html package beschäftigen.

2.4 Objektorientierung

In diesem Kapitel werden die wichtigsten Klassen bzw. Typen von Klassen, die zum Einsatz gekommen sind, und ihre Zusammenhänge (siehe auch das UML – Diagramm im Anhang) erläutert.

2.4.1 SetGet

Die Klasse SetGet ist die Oberklasse einer großen Anzahl von anderen Klassen. Der Grund dafür sind die 3 nützlichen Methoden, die die Klasse besitzt:

- **set()**: Mit Hilfe dieser Methode kann der Wert jeder Instanzvariable gesetzt werden. Dabei wird der Name der Variablen als String übergeben.
- **get():** Diese Methode kann den Wert einer jeden Instanzvariable liefern. Dabei wird der Name der gewünschten Variablen als String übergeben.
- **append():** Die Methode ***append()*** kann den Wert jeder Instanzvariable um einen anderen Wert erweitern. So können z.B. einem String beliebig oft andere Strings angehängt werden.

Die Methoden der SetGet Klasse bieten eine sehr einfache Möglichkeit an, Schreib- und Lesezugriff auf geschützte Instanzvariablen zu realisieren.

2.4.2 Datenbankklassen

Für Datenbankzugriffe wurden folgende 2 Klassen bzw. Klassentypen entworfen:

- **MySQL:** Diese Klasse bietet Methoden zum Auf- und Abbau von Datenbankverbindungen und zur Datenmanipulation an. Mögliche Fehlermeldungen werden mit Hilfe der ***Logger*** Klasse protokolliert.
- **Query / Subklassen von Query:** Mit Hilfe der Query-Klassen werden sämtliche Datenbankabfragen aufgebaut. Sie eignen sich sehr gut für Datenbankabfragen, die abhängig von Benutzereingaben on-the-fly generiert werden. Da jede Abfrage aus mehreren Teilen besteht, kann man gezielt die Teile anpassen, die variabel sind. So kann z.B. der WHERE – Abschnitt einer Aktualisierungsabfrage erweitert bzw. ersetzt werden, ohne dass die UPDATE – und SET – Abschnitte geändert werden. Natürlich kann man der Datenmanipulationsmethode des Datenbankobjekts auch einen String übergeben.

2.4.3 Benutzerklassen

Bei der Benutzerverwaltung wird zwischen Besucher (***Visitor***), eingeloggter Benutzer (***User***) und Administrator unterschieden.

- **Visitor:** Jeder Besucher des Bandportals ist ein ***Visitor***. Die Klasse bietet Methoden an, mit deren Hilfe Information über den Besucher ermittelt werden können. Diese Informationen werden z.B. für die Online-Statistik benötigt.
- **User:** Sobald sich ein ***Visitor*** ins Portal einloggt hat, wird er automatisch zum ***User***. Die Klasse bietet Methoden zum Einloggen und Benutzernamenverwaltung an. Jedes User – Objekt enthält auch ein Profile – Objekt (siehe 2.4.4), dass das Profil des eingeloggten Benutzers abbildet.

2.4.4 Profile / Subklassen von Profile

Die von der Klasse Profile erbenden Klassen bilden, wie der Name schon sagt, Profile ab.

- **UserProfile:** Die Klasse enthält alle Informationen über einen eingeloggten Benutzer. Anlegen, Aktualisierung und Löschung von Benutzerdaten wird durch die Methoden dieser Klasse ermöglicht.
- **BandProfile:** Diese Klasse kümmert sich um die Daten der angemeldeten Bands. Dazu zählen z.B. die Bandmitglieder.

2.4.5 Index / Subklassen von Index

Die Index – Klassen repräsentieren im Grunde Aufzählungen von verschiedenen Objekten. Mit ihrer Hilfe können auf einfache Weise Listen gebaut werden, z.B. Benutzerliste, Bandliste, Instrumentliste usw.

Ein spezieller Fall der Index – Klassen sind die LimitedIndex – Klassen. Sie bilden auch Listen ab. Diese Listen sind allerdings beschränkt. So kann z.B. eine Liste aller Instrumente gebaut werden, die ein bestimmter Benutzer spielt bzw. nicht spielt.

2.4.6 Weitere Klassen

Zusätzlich wurden auch Klassen entworfen, die sich keiner der oben genannten Kategorien zuordnen lassen. Diese werden nachfolgend aufgelistet.

- **Logger:** Die Logger – Klasse bietet Protokollierungsfunktionalität an. Mit Ihrer Hilfe können z.B. Fehlermeldungen protokolliert werden, die bei der Ausführung einer Datenbankabfrage auftreten.
- **Image:** Eine Klasse für Bilder. Sie ist für die Benutzer – und Bandprofile von Bedeutung.
- **Menu:** Die Klasse wird zum Aufbau der Navigation verwendet.
- **Band:** Objekte dieser Klasse bilden die angemeldeten Bands ab.
- **News:** Dies ist ein Klasse zur Verwaltung von Nachrichten
- **Show:** Konzerte und andere Veranstaltungen werden mit Hilfe dieser Klasse verwaltet.
- **Location:** Mit Hilfe der Location – Klasse werden Veranstaltungsorte abgebildet.
- **HTML:** Diese Klasse enthält einige nützlich statische Methoden, die für die Generierung von HTML – Elementen (Quelltext) zuständig sind. So kann z.B. eine Auswahlliste mit Daten aus der Datenbank dynamisch generiert werden. Dadurch wird der Code deutlich kürzer und übersichtlicher.

2.5 Feeds

Für Shows und News können jeweils RSS Feeds generiert werden. Die Feeds können mit einem entsprechenden News Reader abonniert werden. RSS Feeds sind XML Dateien die wir mit Hilfe des PHP PEAR-Pakets XML_Tree erzeugen. Wird der Feed URL eine Band-Id übergeben, so wird der Feed nur für die entsprechende Band generiert. Ansonsten wird der Feed für alle Bands generiert.

2.6 Externe Libraries

Einige Dinge haben wir mit der Hilfe von zusätzlichen Libraries gelöst.

2.6.1 E-Mail Versand

Für den E-Mail Versand haben wir die Möglichkeiten des Zend Frameworks[1] genutzt. Dieses stellt Klassen und Methoden zum Erstellen und Versenden von E-Mails zur Verfügung. Ursprünglich wollten wir mit diesem mächtigen Framework auch die Feeds generieren. Die Möglichkeiten um Feeds zu generieren stellten sicher allerdings als schlecht dokumentiert heraus was die Nutzung unmöglich machte. Evtl. werden wir das Framework wieder entfernen und eine eigene Klasse zum E-Mail Versand schreiben.

2.6.2 Texteingabe

Zur Texteingabe nutzen wir das Open Source Skript spaw[2]. Dieses stellt komfortable Möglichkeiten für die Eingabe von Text zur Verfügung. Es bietet einen fein konfigurierbaren WYSIWYG Editor an. Dieser kann auch so konfiguriert werden das Tabellen, Grafiken und anderen HTML Elemente eingefügt werden können. Wir nutzen das Skript momentan aber nur zur simplen Texteingabe für die News.

[1] http://framework.zend.com/
[2] http://sourceforge.net/projects/spaw/

3 Module

In diesem Kapitel werden wir etwas auf die wichtigsten Module eingehen und welche Aufgaben diese erfüllen.

3.1 Online-Statistik (online_stat)

Die Online-Statistik zeigt an, wie viele Benutzer gerade online sind. Dabei wird zwischen eingeloggten Benutzern und Besuchern der Seite unterschieden. Des Weiteren wir durch die Online-Statistik ein Zähler der bisherigen Besucher der Seite umgesetzt.

3.2 Benutzer

3.2.1 Registrierung (register)

Jeder Besucher der Website hat die Möglichkeit sich zu registrieren. Dadurch bekommt er die Möglichkeit ein Band zu registrieren und ein Benutzerprofil anzulegen.

Die Registrierung läuft so ab, dass der Benutzer seinen gewünschten Benutzernamen, ein Passwort und seine E-Mail Adresse angibt. Sind die Angaben korrekt bekommt er eine Bestätigung in Form einer E-Mail zugeschickt. Die E-Mail enthält einen Link zur Aktivierung des Profils. Klickt der Benutzer diesen Link an, so wird sein Profil aktiviert und er Benutzer kann sich daraufhin mit Benutzername und Passwort einloggen.

3.2.2 Passwort vergessen (request_password)

Hat ein registrierter Benutzer sein Passwort vergessen so kann er ein neues anfordern. Dazu muss er seinen Benutzernamen und seine E-Mail Adresse angeben. Stimmt die Kombination aus Benutzername und E-Mail Adresse überein, so wird an die E-Mail Adresse ein neues Passwort geschickt. Mit diesem kann der Benutzer sich anschließend einloggen.

3.2.3 Benutzerprofil (user_profile, user_contact, user_instruments)

Über das Benutzerprofil kann der Benutzer persönliche Daten eingeben und sich anderen Benutzern dadurch präsentieren. Er kann ein Profilbild hochladen, Kontaktdaten angeben sowie die Instrumente die er spielt hinterlegen.

3.3 Bands

3.3.1 Band-Registrierung (my_bands)

Allen registrierten Benutzern steht die Möglichkeit zur Verfügung eine Band bzw. mehrere Bands zu registrieren. Dazu gibt der Benutzer in einem Formular Bandname, Stadt, PLZ und E-Mail an. Sind die Eingaben korrekt wird eine E-Mail an die eingegebene E-Mail Adresse sowie eine E-Mail an den Administrator geschickt. Der Administrator kann dann über die Administrationsoberfläche die Daten prüfen und entscheiden ob die Band freigegeben werden soll.

3.3.1.1 Warum Bands freigeben?

Da nur Bands aus der Region der Grunz unterstützt werden sollen muss eine Überprüfung der Band durch den Administrator erfolgen. Dadurch soll verhindert werden das auch Bands die nichts mit der Grunz zu tun haben die Funktionen für Bands nutzen können.

3.3.2 Bandprofil und Bandservices

Wurde eine Band durch den Administrator freigegeben, so können Daten zur Band eingegeben werden. Des Weiteren können die freigegeben Bands die Services Shows und News nutzen.

3.3.3 Bandservices

3.3.3.1 Shows (my_shows)

Dieser Service stellt den Bands die Möglichkeit zur Verfügung ihre Konzerttermine einzutragen und zu verwalten.

Die Band kann unter anderem aus vordefinierten Veranstaltungsorten auswählen oder einen eigenen eingeben.

3.3.3.1.1 Veranstaltungsorte

Durch den Administrator können Veranstaltungsorte eingetragen werden (dieses Modul ist noch nicht in der Administration implementiert). Diese können dann durch die Band beim Eintragen einer Show ausgewählt werden.

Trägt eine Band einen Veranstaltungsort manuell ein, so wird dieser auch in die Datenbank geschrieben. Der Eintrag wird aber als nicht aktiv markiert.

Dahinter steckt die Idee, dass der Administrator die von Bands neu eingetragenen Veranstaltungsorte prüfen kann und evtl. die Informationen zu diesen noch ergänzt. Somit baut sich nach und nach eine umfangreiche Veranstaltungsortdatenbank auf.

3.3.3.2 News (my_news)

Hier können Bands Neuigkeiten eintragen und diese verwalten. Durch den Einsatz des WYSIWIG Editors spaw kann der User seinen Text auch anschaulich gestalten.

3.4 Administration

Der Administrationsbereich dient zur Verwaltung von Usern und Bands.
Bislang können

- Band- und Userprofile betrachtet werden
- User aktiviert und deaktiviert werden
- Bands freigeschaltet werden
- Bands aktiviert und deaktiviert werden

In Zukunft soll es möglich sein:

- News anlegen/bearbeiten/löschen
- Statische Seiten (Content) anlegen/bearbeiten/löschen
- Eintragen/Bearbeiten/Löschen von Veranstaltungsorten
- Bearbeiten/Löschen von Usern und Bands

Der Administrationsbereich ist bislang nur durch eine .htaccess/.htpasswd Kombination geschützt bzw. zugänglich (dies ist auf applic.mi.hdm-stuttgart.de nicht funktionsfähig).

4 Design

4.1 Formatierung und Layout

Die Formatierung der Seite erfolgt selbstverständlich mit Cascading Stylesheets (CSS). Die zugrunde liegende Anordnung der Elemente der Seite wird jedoch durch die Verwendung von Tabellen festgelegt. Dies soll gewährleisten, dass Besucher, die z.B. an einer Sehschwäche leiden, die Seite auch ohne Stylesheet sinnvoll nutzen können (vgl. Abb. 1).

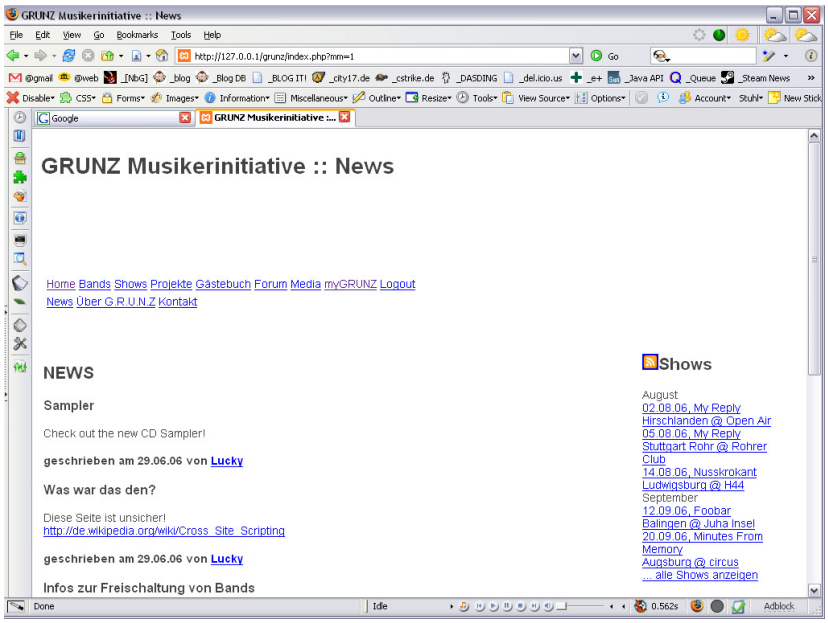

Abbildung 1: GRUNZ ohne CSS

Abbildung 2: GRUNZ mit CSS (light.css)

Abbildung 3: GRUNZ mit altem CSS (styles.css)

Des Weiteren erleichterte die Verwendung von Tabellen die Interoperabilität v.a. mit nicht standardkonformen Browsern wie dem Microsoft Internet Explorer (auch Version 7 Beta).

4.1.1 Anordnung der Seite

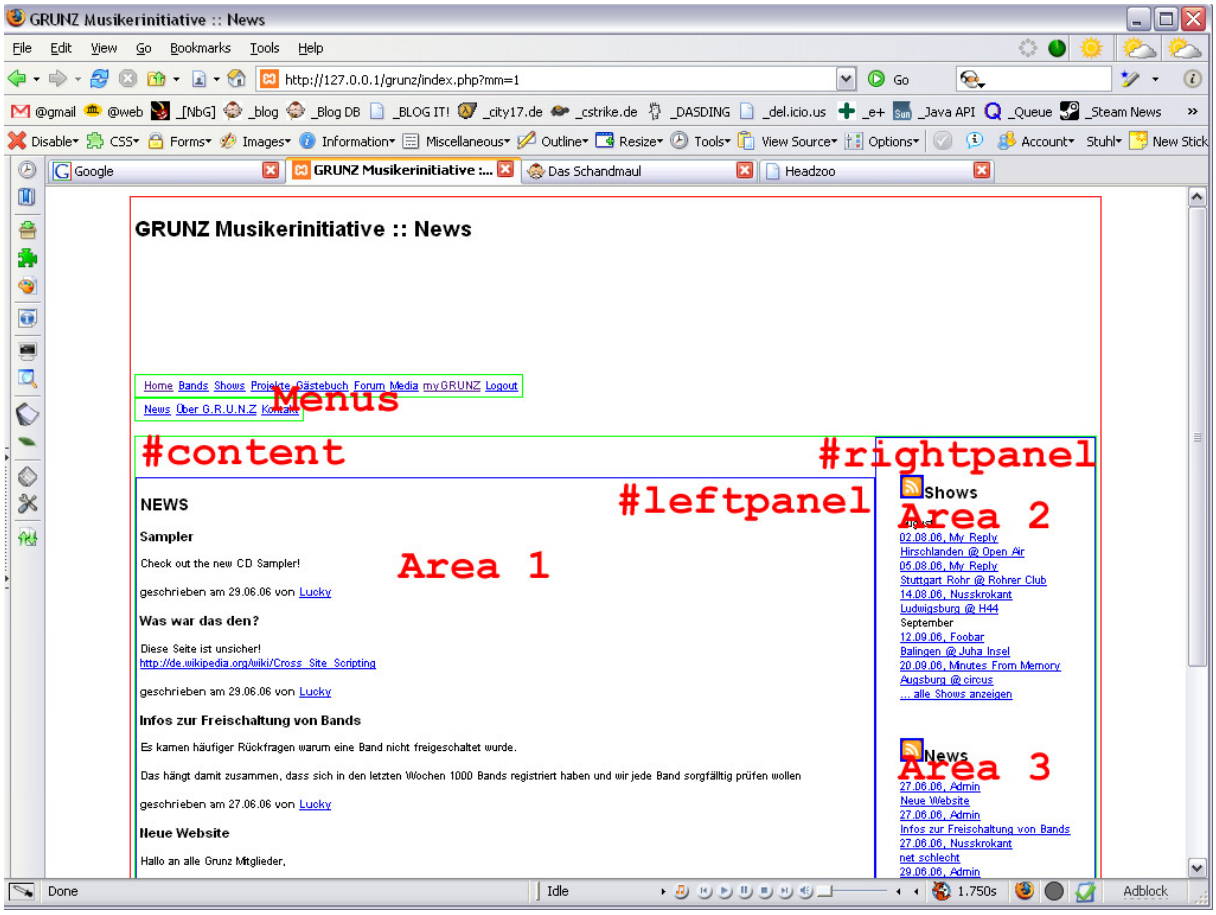

Abbildung 4: Anordnung der Seitenelemente mit Hilfe von Tabellen

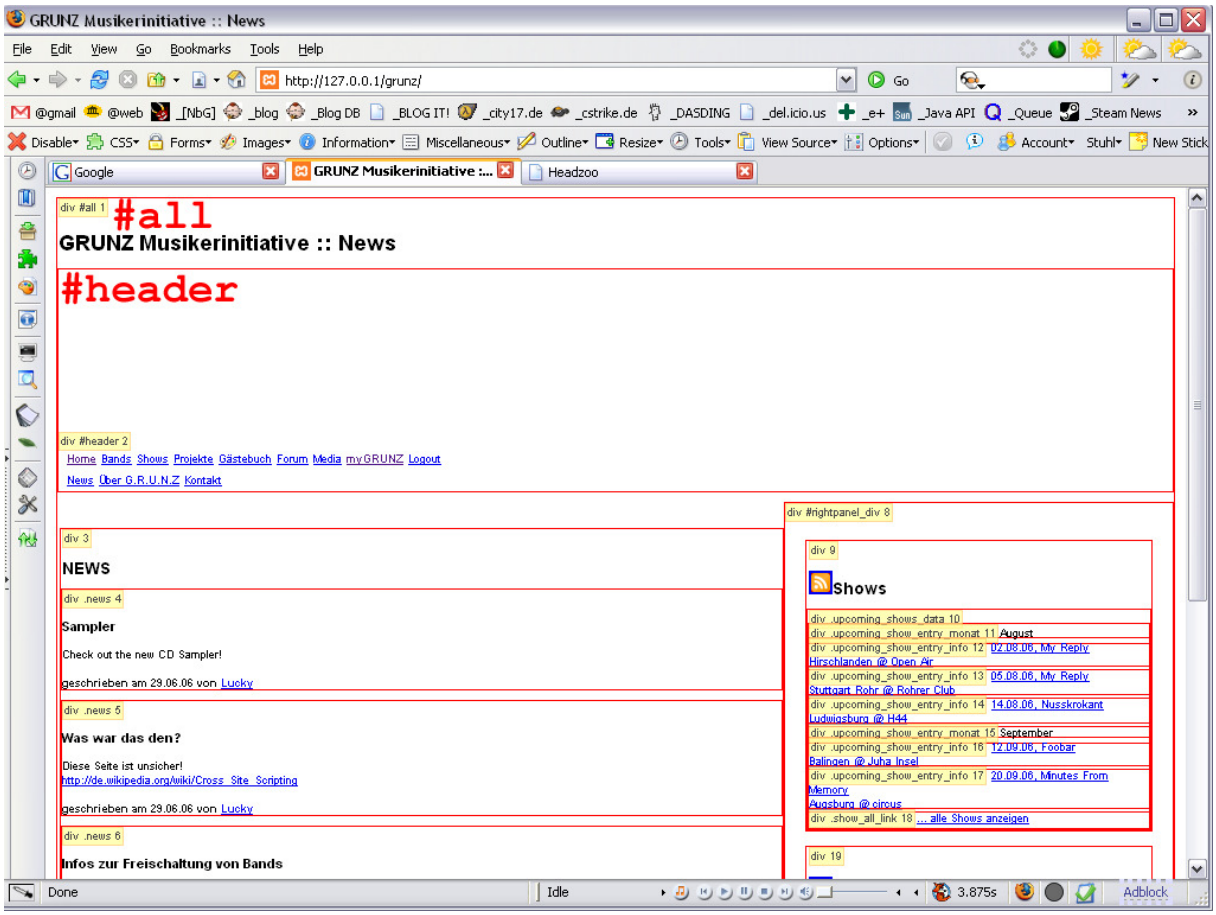

Abbildung 5: Anordnung der Layer zur Formatierung

Aus Abbildung 4 werden die verwendeten Tabellen erkenntlich. Zudem wurden die in den zur Ausrichtung verwendeten Tabellenzellen enthaltenen Inhalte mit einem Layer (, vgl. Abb. 5) umfasst, da die Formatierungsmöglichkeiten von Layern in CSS weitaus mächtiger sind.
Die äußere Tabelle (in Abb. 4 rot umrandet) dient ausschließlich zur Zentrierung der Seite. Der darin enthaltene Layer mit der ID #all dient zur Formatierung des Rahmens und des Hintergrunds.
Der Layer mit der ID #header dient zur Darstellung des Logos und des Schriftzuges (bei der Verwendung von CSS wird die in Abbildung 5 zu sehende Überschrift nicht dargestellt) und enthält das Navigationsmenü.
Die Tabelle mit der ID #content umfasst den eigentlichen Inhalt und ist weiterhin unterteilt in zwei Tabellen mit den IDs #leftpanel und #rightpanel. Diese dienen zur Aufteilung der Inhalte in logische Areas (vgl. Abb. 4). Area 1 innerhalb der Tabelle #leftpanel dient zur Darstellung des Inhalts des aktuellen Menüpunkts. In den Areas 2 bis n (derzeit 2 bis 4) innerhalb der Tabelle #rightpanel können abhängig davon weitere, verwandte Inhalte angezeigt werden.

4.1.2 Allgemeines zur Formatierung mit CSS

Innerhalb der Stylesheet-Datei werden zunächst allgemeine und allgemein gültige Formatierungen vorgenommen. Darunter fallen Schriftformatierung (Stil, Größe, Farbe) für das gesamte Dokument, Absätze, Überschriften und Links sowie z.B. Rahmenformatierung für verlinkte Bilder.
Im Anschluss erfolgt dann die Formatierung spezieller Elemente, sofern notwendig. Hierbei folgt man am besten dem DOM-Tree des Dokuments.

Um an dieser Stelle nicht zu tief in die Formatierungsmöglichkeiten von Stylesheets einzudringen, ein Beispiel anhand der Überschrift <h2> (vgl. Abbildungen 1- 3 Überschriften „News“ und „Shows“):

1. Der Schriftstil für das gesamte Dokument wird definiert durch Formatierung des <body>-Tags:
```
body
{
font-family:arial,sans;
color:#444;
font-size:11px;
...
}
```
2. Die Farbe für alle Überschriften wird festgelegt mit
```
h1, h2, h3, h4, h5
{
color:#555;
}
```
3. Das spezielle Erscheinungsbild der Überschrift <h2> wird definiert:
```
h2
{
font-size:2em;
border:2px solid #900;
background:#e00 url(img/bg_red.gif);
color:#fff;
padding:7px 5px 7px 5px;
margin-bottom:10px;
margin-top:0px;
}
```
4. Das spezielle Erscheinungsbild der Überschrift <h2> innerhalb der Tabelle #rightpanel wird angepasst:
```
#rightpanel h2
{
font-size:13px;
margin:0px;
padding:3px 5px;
}
```

4.1.3 Umgehung von Inkompatibilitäten des Internet Explorer 6

Ein kleines Problem das in unserem Falle auftrat ist die Unfähigkeit des IE6, transparente PNGs darzustellen. Um dies zu kompensieren wird eine weitere Inkompatibilität des IE6 bezügl. des sog. Kind-Selektors (>)[3] genutzt.

Die Funktionsweise dieses CSS-Hacks[4] am Beispiel des in Abbildung 3 zu sehenden alternativen Stylesheets (styles.css), wobei der dunkle Hintergrund der Tabelle mit der ID #content ein Transparentes PNG ist:

```
#content
{
        clear:both;
        margin-left:0px;
```

[3] http://de.selfhtml.org/css/layouts/browserweichen.htm#kind_selektor
[4] Bezeichnung für ein mehr oder weniger simples Workaround, meist in Verbindung mit IE6. Vgl. auch http://de.wikipedia.org/wiki/Hack

```
        background:url(img/bg_trans_ie.jpg);
        border:7px solid #600;
}

#all > #content
{
        background:url(img/bg_trans.png);
}
```

Dabei werden zuerst im Block `#content{…}` allgemein gültige Formatierungen und das Hintergrundbild, das im IE6 angezeigt werden soll, definiert.
Im Block `#all > #content{…}`, der Aufgrund der Verwendung des Kind-Selektors > von IE6 nicht interpretiert wird, ist das transparente Hintergrundbild definiert.

5 Sicherheit

Um automatisierte Registrierungen durch Skripte zu verhindern, muss ein User bei der Registrierung einen zufällig generierten Wert eingeben der in Form einer Grafik angezeigt wird. Dieses so genannte Captcha[5] sollte von einer Maschine nicht ausgewertet werden können.

Ein Sicherheitsloch haben wir bei der Registrierung nicht berücksichtigt. Die Auswahl des Passworts wird von uns nicht überprüft. So kann z.B. ein User ein Passwort das gleich seinem Benutzername ist wählen.

Weitere Sicherheitsvorkehrungen haben wir bis jetzt nicht getroffen da uns die Gefahren die beispielsweise durch Cross-Site Scripting[6] und SQL-Injektion[7] ausgehen nicht bewusst waren. Diese Sicherheitslücken müssen wir natürlich noch schließen. Deshalb müssen wir den Code diesbezüglich nochmals prüfen und überarbeiten.

[5] http://de.wikipedia.org/wiki/Captcha
[6] http://de.wikipedia.org/wiki/Cross_Site_Scripting
[7] http://de.wikipedia.org/wiki/SQL_Injection

6 Dokumentation

Wir haben uns bemüht während der Programmierung den Code gut zu dokumentieren. Ähnlich wie javadoc gibt es für PHP das Tool phpDocumentor[8] mit welchem aus den vorhandenen Quelldateien eine anschauliche, gut strukturierte Dokumentation generiert werden kann. Diese findet sich im Anhang.

Damit ein Kommentar von phpDocumentor berücksichtigt wird schreibt man einfach

```
/**
* Kommentar
* @package Demo
*/
```

d.h. es muss auf den einleitenden Stern bei einem Kommentar gleich ein weiterer folgen. Im Kommentar Block kann dann mit den phpDocumentar Tags wie @param, @return etc. gearbeitet werden.

Die Kommentare in unserem Code haben wir leider nicht konsequent genug durchgezogen. Deshalb sieht das von phpDoc generierte Ergebnis leider etwas dürftig aus. Daraus haben wir gelernt wie wichtig es ist sofort während der Programmierung die entsprechenden Kommentare zu machen, auch wenn dies einige Zeit in Anspruch nimmt.

[8] http://www.phpdoc.org/

7 Fazit

Unsere Projektplanung war nicht optimal. Deshalb kamen wir am Ende des Projekts sehr ins schwitzen. Es gab noch viel zu tun, aber die Zeit wurde immer knapper. Des Weiteren haben wir uns zu viel vorgenommen. Einige Ideen haben sich am Anfang sehr gut angehört, aber als es um die Umsetzung ging wurde uns erst bewusst wie viel Aufwand ein kleine Idee eigentlich bedeutet. Deshalb ist uns durch das Projekt die Wichtigkeit einer guten Projektplanung bewusst geworden und wir werden uns mit dem Thema Projektplanung nochmals grundlegend beschäftigen.

Mit dem Erreichten sind wir zufrieden und haben im Verlauf des Projektes sehr viel dazu gelernt. Natürlich hätten wir gerne noch einige Features mehr implementiert, aber durch die knappe Zeit war uns dies leider nicht mehr möglich.

Das erste Feedback der Grunz war sehr gut und sie können sich gut vorstellen des es bei den Bands und Besuchern der Seite sehr gut ankommen wird. Natürlich gab es noch die ein oder andere Anmerkung was sie noch ändern würden, aber das waren eher Kleinigkeiten.

Die nächsten Schritte werden die Implementierung weiterer Module sowie der Test mit ausgewählten Usern und Bands sein.

8 Anhang

- ER-Diagramm
- UML Diagramm
- phpDoc im HTML Format
- Sourcecode als zip
- Alternativ kann der Sourcecode auch vom Subversion Server der HdM ausgecheckt werden. https://version.mi.hdm-stuttgart.de/svn/grunz_bandportal