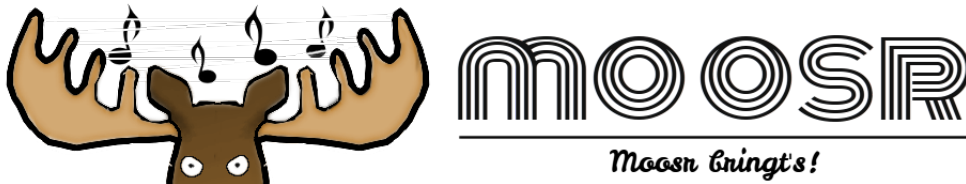


HOCHSCHULE HOF, UNIVERSITY OF APPLIED
SCIENCES

WEBTECHNOLOGIE UND WEBMARKETING MIT OPENSOURCE
SOFTWARE



A music metadata search service

Beteiligte Studenten:

Christopher PAHL,

MN: #01271710

Christoph PIECHULA,

MN: #02746310

Dozent:

Dr. Alois KASTNER-MARESCH

8. Januar 2013

Inhaltsverzeichnis

I. Die Idee	5
1. Vorwort	6
1.1. Motivation	6
1.2. Idee	6
1.3. Anmerkung	6
II. Webmarketing	7
2. Untersuchung (20 Fragen)	8
2.1. Welche besonderen Eigenschaften, Stärken, Alleinstellungsmerkmale hat (m)eine Leistung?	8
2.2. Welche besonderen Alleinstellungsmerkmale und Vermarktungskonzepte haben Mitbewerber?	8
2.3. Wie können wir unsere Leistung durch eine Spezialisierung abgrenzen und unsere Stärken optimal zur Geltung bringen?	9
2.4. Was ist die erfolgversprechendste Zielgruppe?	10
2.5. Welche Medien nutzt die Zielgruppe?	10
2.6. Wer sind die wichtigsten Meinungsführer?	10
2.7. Was sind die brennenden Probleme der Zielgruppe?	10
2.8. Welche Trojanische Pferde können wir entwickeln?	10
2.9. Welche Überraschung können wir Meinungsführern bieten?	11
2.10. Wie können wir ein Angebot mit PR bekannt machen und nicht nur durch Anzeigenwerbung?	11
2.11. Welche Risiken gibt es dafür, dass die Zielgruppe die Leistung nicht nutzen könnte? Was kann potentielle Kunden eventuell abschrecken?	11
2.12. Welche Kooperationsstrategie können wir verfolgen?	11

2.13. Können wir einen Markennamen generieren?	12
2.14. Ist eine Intel-Inside Strategie möglich?	12
2.15. Welches Key-Visual / Key-Theme können wir verwenden?	12
2.16. Können wir eine Strategie entwickeln, wie wir mit PR und Vorträgen an die Zielgruppe oder deren Meinungsführer direkt rankommen?	12
2.17. Wie können wir einen Markenaufbau zum Nulltarif erreichen?	13
2.18. Wie können wir Mundpropaganda unterstützen?	13
2.19. Welchen Claim sollten wir verwenden? Die treffende Message bewirkt Wunder!	13
2.20. Welches Leitbild und welche Ziele verfolgen wir für die Ansprache der Zielgruppe?	13
3. Keyvisual und Claim	14
3.1. Key-visual	14
3.2. Claim	15
4. Markt und Wettbewerb	16
5. Finanzierung	17
5.1. Lizenz	17
5.2. Mögliche Finanzierungsszenarien	17
III. Technik	19
6. Konzept Web	20
6.1. Allgemein	20
6.2. Farbklima	21
6.3. Webservice Kategorien	21
6.4. Footer-Bereich	29
7. Google Optimierung	31
7.1. Duplicate Content vermeiden	31
7.2. Content Seiten über Navigation erreichbar	32
7.3. Artikel über mehrere Seiten strecken	32
7.4. Festlegbare Linktexte	32
7.5. Einfache URLs erzeugen	33

7.6.	Editierbare Metadaten	33
7.7.	Übersicht über Artikel/Keywords	34
7.7.1.	Developers:	34
7.7.2.	FAQ:	34
7.7.3.	Shop:	34
7.7.4.	About us:	34
7.7.5.	Suche nach Lyrics:Farin Urlaub + Lieber Staat	34
7.7.6.	Blogübersicht:	34
7.7.7.	Blogpost - Bandvorstellung: Coppelius:	35
7.7.8.	Blogpost - Knorkator auf Haltestelle Woodstock Festival:	35
7.7.9.	Blogpost - Das lose Album - Eine Rezession:	35
7.7.10.	Blogpost - Moosr says Hello World:	35
8.	Technische Implementierung	36
8.1.	Vorwort - technische Implementierung	36
8.2.	Verwendete Frameworks	36
8.2.1.	Flask	36
8.2.2.	libglyr	36
8.2.3.	sqlite3	36
8.3.	Historisches	37
8.4.	Implementierung	37
9.	Weiteres	38
9.1.	Github-Repository	38
9.2.	Filmeabend	38

Teil I.

Die Idee

1. Vorwort

1.1. Motivation

Im Netz gibt es eine riesige Menge an Seiten die sich darauf spezialisiert haben Informationen jeglicher Art zu Musikstücken, Künstlern und Ähnlichem zu liefern.

Die Reichweite geht dabei von Songtexten bis hin zu Tagging-Informationen die von Musikabspielprogrammen ausgelesen werden kann. Es gibt Streamingdienste wie last.fm, oder Spotify die neben der eigentlichen Musik auch entsprechende Metadaten anbieten. Im Falle von last.fm sogar mit einer gut nutzbaren Webschnittstelle.

Allerdings sind die ganzen Informationen mehr oder minder über das ganze Web fragmentiert. Last.fm bietet lediglich Coverart und Künstlerbilder an, lyrics.wikia.com nur Lyrics und ein anderer Provider nur Biographien und Reviews.

1.2. Idee

Unsere Idee ist nun eine Metadatensuche zu bauen, die möglichst viele Metadatatypen bündelt und dabei auch über eine Web-API nutzbar ist.

Um diese Idee abzurunden, wollen wir uns einen Namen in der Indie Musik Szene schaffen und bieten neben der Suchdienstleistung zugleich einen „Treffpunkt“ für Indie Musik Liebhaber, Liedermacher und freischaffende Künstler an.

1.3. Anmerkung

Wir wollen bereits jetzt darauf hinweisen dass im Blog genannte Personen und Ereignisse zum Teil fiktiv sind. Auch das angebotene Forum und der Webshop existiert nur auf dem Papier. Auch viele Links, wie zB auf „unsere“ Flickr oder Facebook Seite existieren nicht, da dies den Rahmen der Studienarbeit gesprengt hätte. Die meisten Blogposts sind allerdings an reale Bands angelehnt.

Teil II.

Webmarketing

2. Untersuchung (20 Fragen)

2.1. Welche besonderen Eigenschaften, Stärken, Alleinstellungsmerkmale hat (m)eine Leistung?

Unsere Dienstleistung bietet eine zentrale Anlaufstelle für Indie Musik Liebhaber, Liedermacher und Künstler und kombiniert diese mit einer Metadaten Suchmaschine die Cover-Art Suche, Songtext-Suche, Biographie-Suche und Artistphoto-Suche in einer Dienstleistung bündelt.

Anbieter mit ähnlichen Dienstleistungen:

- <http://www.allcdcovers.com>
- <http://www.albumart.org> (keine soziale Komponente)
- http://lyrics.wikia.com/Lyrics_Wiki
- <http://www.last.fm>

2.2. Welche besonderen Alleinstellungsmerkmale und Vermarktungskonzepte haben Mitbewerber?

Last.fm Der Anbieter Last.fm spezialisiert sich hauptsächlich auf Musikstreaming. Desweiteren bietet er eine Metadatenschnittstelle an, die Artist-Biographien, Cover-Art und Artist-Photos liefern kann. Kompatible Clients und Webseiten können daher diese anzeigen. Zudem bietet er eine soziale Komponente an, indem pro User ein Profil vorhanden ist, auf dem einsehbar ist was dieser gerade hört bzw. mag.

alldcovers Dieser Anbieter hat sich auf besonders hochauflösende Cover-Art Images spezialisiert. Außerdem bietet er zudem Bilder von der CD Rückseite und dem Inlet an. Er ist vollkommen Community-basiert.

albumart Der Anbieter Albumart ähnelt unserem Suchangebot noch am ehesten. Allerdings beschränkt er sich lediglich auf CD und DVD Cover-Art. Er bietet zudem eine API.

lyrics.wikia Dieses Angebot ist in einer Wiki-ähnlichen Struktur organisiert. Leider bietet dieser Anbieter nur Songtexte an. Allerdings verlinkt er auf andere Angebote und integriert soziale Dienste wie Twitter und Facebook.

Wir sollten erwähnen, dass es noch eine große Anzahl weiterer Webseiten gibt. Allerdings haben wir die oberen exemplarisch als Vergleich ausgewählt.

2.3. Wie können wir unsere Leistung durch eine Spezialisierung abgrenzen und unsere Stärken optimal zur Geltung bringen?

Unsere Dienstleistung konzentriert sich auf die Bereitstellung einer leichtgewichtigen Webschnittstelle die sowohl andere Webseiten als auch Desktopclients nutzen können. Das Hauptaugenmerk liegt hier auf einer besonders guten Dokumentation sowie einer einfachen Integration der Dienste in verschiedenen externen Produkten.

Allerdings versteht sich unser Angebot als eine Art „Metaprovider“, der sich auf andere Dienstleister wie beispielsweise last.fm stützt und diese bündelt.

Als Anlaufstelle für eine große Community fühlen wir uns verpflichtet diese neben unserem Suchdienst über Neuigkeiten jeglicher Art auf dem Laufenden zu halten.

Auf technischer Seite legen wir großen Wert auf Transparenz und die Integration freier Software. So steht der Quellcode der Anwendung offen auf Github¹ zur Verfügung.

¹<https://www.github.com/studenkittens/flascat>

2.4. Was ist die erfolversprechendste Zielgruppe?

Als Zielgruppe sehen wir primär musikinteressierte Benutzer und Anbieter von Musik-Abspielsoftware (und damit deren Nutzerbasis). Die Zielgruppe ist prinzipiell ein junges experimentierfreudiges Publikum das sich schnell für Neues begeistern lässt.

2.5. Welche Medien nutzt die Zielgruppe?

Primär nutzt die Zielgruppe das Internet. Andere Medien sind für unsere Dienstleistung zu vernachlässigen.

2.6. Wer sind die wichtigsten Meinungsführer?

Frage trifft auf unsere Dienstleistung nicht zu, da es in dem Sinne keine Meinungsführer gibt.

2.7. Was sind die brennenden Probleme der Zielgruppe?

Die Zielgruppe hat das Problem Metadaten an einer zentralen Stelle im Netz aufzufinden. Unsere Dienstleistung soll diese Angebotslücke schließen und zusätzlich als Treffpunkt für gleichgesinnte Musikliebhaber und freischaffende Künstler dienen.

Wie oben bereits erwähnt, ist die Verteilung der Communities stark dezentral. Wir wollen das nach Möglichkeit soweit wie möglich zentralisieren und bieten ein Forum sowie Links auf andere Seiten.

2.8. Welche Trojanische Pferde können wir entwickeln?

Im Moment gibt es kein Angebot das verschiedene Metadaten maschinenlesbar in einem Angebot bündelt. Unser Angebot kann auf anderen Webseiten leicht durch die leichtgewichtige Webschnittstelle eingebunden werden.

Ein zwingender Nutzen würde sich für Musicplayer ergeben, da dort Client-Bibliotheken für unseren Service zur Verfügung stehen.

2.9. Welche Überraschung können wir Meinungsführern bieten?

Frage trifft auf unsere Dienstleistung nicht zu, da keine Meinungsführer vorhanden.

2.10. Wie können wir ein Angebot mit PR bekannt machen und nicht nur durch Anzeigenwerbung?

Durch den Einsatz von sozialen Medien wie Twitter oder Facebook kann ein positives Meinungsbild suggeriert werden. Desweiteren kann der Service von verschiedenen Fachseiten getestet und empfohlen werden. Aufgrund der hohen Verfügbarkeit unserer Dienstleistung die sich über verschiedene soziale Netze von Facebook bis hin zu Flickr streut sollte es problemlos möglich sein unsere Dienstleistung einem breitem Publikum zugänglich zu machen.

2.11. Welche Risiken gibt es dafür, dass die Zielgruppe die Leistung nicht nutzen könnte? Was kann potentielle Kunden eventuell abschrecken?

Ein mögliches Risiko wäre eine zu komplexe entwickler-unfreundliche API. Um dieses Risiko zu minimieren wird darauf geachtet die API und Seite nach offenen Standards zum implementieren.

Für normale User hätte ein kompliziertes und überladenes Design eine abschreckende Wirkung, weswegen wir uns hier auf Minimalismus konzentrieren.

2.12. Welche Kooperationsstrategie können wir verfolgen?

Da wir außer einem externen Shop und sozialen Diensten nicht auf andere Partner zurückgreifen trifft die Frage nicht auf unser Angebot zu.

2.13. Können wir einen Markennamen generieren?

Diese Frage trifft nicht direkt zu da es sich bei moosr in erster Linie um ein nicht kommerzielles Produkt handelt wo die Etablierung eines Markennamens zweitrangig ist.

2.14. Ist eine Intel-Inside Strategie möglich?

Ja, eine Intel-Inside Strategie wäre denkbar, wenn verschiedene Musikplayeranbieter moosr als Metadatensuchmaschine nutzen. Diese Situation könnte man durch die Implementierung von Wrapperlibraries für verschiedene Programmiersprachen positiv beeinflussen.

Mögliche „Intel-Inside“-Claims für Playeranbieter:

- *Moosr bringt's - Englisch alternativ: Moosr delivers*
- *Moosr inside*

2.15. Welches Key-Visual / Key-Theme können wir verwenden?

Eine weiche, helle Farbgebung mit grün/braun Farbtönen die ein skandinavisches „Feeling“ vermittelt um die zunehmend in letzter Zeit aus den skandinavischen Ländern kommende Indie Musik Bewegung visuell zu unterstreichen.

Weitere Details zum KeyVisual/Claim finden sich im entsprechenden Kapitel 3.

2.16. Können wir eine Strategie entwickeln, wie wir mit PR und Vorträgen an die Zielgruppe oder deren Meinungsführer direkt rankommen?

Die Frage trifft nicht auf unsere Dienstleistungen zu.

2.17. Wie können wir einen Markenaufbau zum Nulltarif erreichen?

Wir setzen auf Mundpropaganda, siehe nächste Frage.

2.18. Wie können wir Mundpropaganda unterstützen?

Bei genügend Mundpropaganda und positiver Rezension verbreitet sich die Dienstleistung automatisch über verschiedene Plattformen sowie z.B. auch Linux-Distributionen.

Dies sollte bei der hohen Verfügbarkeit in sozialen Netzen und der engen Bindung zur Community kein Problem darstellen.

2.19. Welchen Claim sollten wir verwenden? Die treffende Message bewirkt Wunder!

Als Claim wurde „moosr bringt’s.“ gewählt. Da dies ein kurzer, einprägsamer Begriff ist der unsere Dienstleistung auf den Punkt bringt. Wir informieren über Neuigkeiten und liefern Metadaten.

2.20. Welches Leitbild und welche Ziele verfolgen wir für die Ansprache der Zielgruppe?

Frage trifft nicht auf unsere Dienstleistung zu.

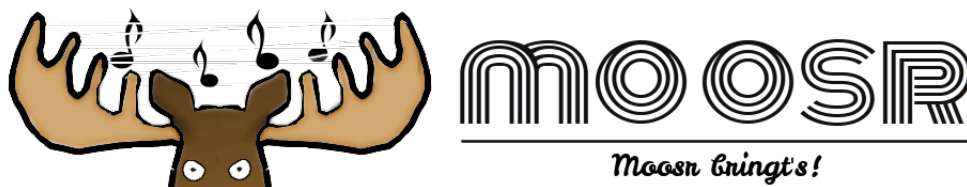
3. Keyvisual und Claim

Beim Key-Visual wurde als Logo ein Elch mit Saiten und Musiknoten im Geweih gewählt. Die Wahl eines Tieres als Logo ist hier leicht an das Vorbild Napster angelehnt welches eine Katze mit Kopfhörern zeigt.



Es wurde bewusst ein Elch ausgewählt, dieser soll symbolisch die anfangs überwiegend aus den skandinavischen Ländern antreibende Kraft der Indie Musik Bewegung visuell darstellen. Desweiteren sollen die Saiten im Geweih den Zusammenhalt der Community widerspiegeln, der grimmige Blick ist eine „Kampferklärung“ an die Mainstream Musik Branche. Der comichaft Ansatz soll hier primär ein junges Publikum ansprechen.

3.1. Key-visual



Beschreibung Das Logo besteht aus einem comichaft gezeichneten Elchkopf, der zwischen den Lücken seines Geweihs Saiten aufgespannt hat. Zwischen diesen sind 4 Musiknoten in unterschiedlicher Größe und Ausrichtung dargestellt.

Neben dem Maskottchen ist der verwendete Schriftzug zu finden. In großer Schrift (Font: Monoton 110) ist dabei der Name unser Dienstleistung geschrieben, darunter in Schreibschrift (Font: Leckerli One 28) unser Claim.

Der grimmige Blick ist bewusst gewählt worden um uns von anderen ewig-glücklichen Maskottchen abzugrenzen.

3.2. Claim

Der Claim soll unsere Dienstleistung möglichst prägnant zusammenfassen. Das „bringt's“ bezieht sich dabei auf das Liefern von Neuigkeiten und Metadaten verschiedener Art. Er ist möglichst kurz ausgewählt worden, und kann auch leicht ins Englische übersetzt werden („moosr delivers“).

Ein solch kurzer Claim kann auch auf Fanartikel wie T-Shirts oder kleine Werbgeschenke wie Kugelschreiber gedruckt werden.

4. Markt und Wettbewerb

Dieser Punkt ist bei uns Bestandteil der „20 Fragen“.

Siehe dazu Kapitel 2.

5. Finanzierung

5.1. Lizenz

Primär ist es uns wichtig den Bekanntheitsgrad zu steigern und eine große Community zu gewinnen welche auch daran interessiert ist die Dienstleistung weiter auszubauen und zu verbessern. Die direkte Kommunikation zu der Community soll eine Weiterentwicklung der Dienstleistung gewährleisten und dabei gleichzeitig die Interessen der „Kunden“ wahren.

Aus diesem Grund haben wir uns für Freie Lizenzen entschieden um so jedem die Möglichkeit zu bieten die Dienstleistung zu verbessern.

Alle Quelltexte stehen unter GPLv3 Lizenz.

<http://www.gnu.org/licenses/gpl-3.0.html>

Das *Design* sowie alle weiteren Medien unter der Creative Commons Lizenz.

<http://de.creativecommons.org/>

5.2. Mögliche Finanzierungsszenarien

Da unsere Dienstleistung nicht kommerziellen Ursprungs ist trifft die Frage der Finanzierung nicht direkt zu. Allerdings gibt es mehrere Möglichkeiten die laufenden Kosten (welche durch die Bereitstellung des Servers bedingt sind) zu decken:

Shop Als primäre Finanzierungsquelle um die Serverkosten zu decken bieten wir über Drittanbieter wie

<http://www.zazzle.de/>

<http://www.cafepress.de/>

Fanartikel wie T-Shirts, Aufkleber etc. an. Durch diese Maßnahme wird zusätzlich als positiver Nebeneffekt Werbung geschaffen.

Spenden Durch Einbindung von Diensten wie Flattr (ein Flattr-Button) oder ein PayPal Spendenkonto können auf freiwilliger Basis Spenden in variabler Höhe geleistet werden. Insbesondere durch den auf der Seite sich befindlichen Flattr-Button hoffen wir dabei auf Mirco-Spenden.

Weiteres Einnahmequellen wären mögliche kostenpflichtige API Anpassungen für „Kunden“ kommerzieller Musikplayer und Contentanbieter (SaaS - Software as a Service). Wir erhoffen uns durch die freigewählten Lizenzen, dass kommerzielle Dienstleister ermutigt werden unseren Dienst zu finanzieren und somit zur Entwicklung direkt oder indirekt beitragen.

Der Dienst an sich ist prinzipiell selbsttragend, da außer den Unterhaltungskosten für Domain und Server keinerlei laufende Kosten anfallen sollten.

Teil III.

Technik

6. Konzept Web

6.1. Allgemein

Unsere Dienstleistung konzentriert sich ähnlich der Unix Philosophie auf eine bestimmte Tätigkeit und versucht diese möglichst gut zu realisieren. Aufgrund dieser Tatsache ist es uns sehr wichtig den User nicht mit „Thematikfremden“ Themen zu nerven. Aufgrund unserer Spezialisierung werden fachfremde Kategorien wie z.B. der Fanartikel Webshop an Drittanbieter „outgesourct“.

Um eine einfache sowie angenehme Usability zu gewährleisten konzentrieren wir uns sehr stark darauf das *Design* nach dem KISS-Prinzip so einfach wie möglich zu halten und den Benutzer nicht mit Werbebannern oder sinnfreien Informationen zu belästigen.

6.2. Farbklima

Das an den Key-Visual angepasste Farbklima sieht wie folgt aus:








Farbe	Name	Farbwerte	Einsatzgebiet
	a	HEX #FDF9EF RGB 253, 249, 239	Hintergrundfarbe für Contentbereich
	b	HEX #F4F1E5 RGB 244, 241, 229	Hintergrund
	c	HEX #544738 RGB 84, 71, 56	Standard Schriftfarbe
	d	HEX #98BF21 RGB 152, 191, 33	Buttonflächen
	e	HEX #C2C5BE RGB 194, 197, 190	Footer Hintergrundfarbe
	f	HEX #BBBBBB RGB 187, 187, 187	Tagcloud Schriftfarbe
	g	HEX #303030 RGB 48, 48, 48	Tagcloud Farbe

Tabelle 6.1.: Farbschema

6.3. Webservice Kategorien


Folgende Hauptkategorien wurden auf moosr umgesetzt:

Layoutübersicht



moosrdata Hier befindet sich die der Kernpunkt unserer Dienstleistung, die Metadaten Suchmaschine. Diese Seite bietet dynamischen Content in Form einer *Tagcloud* mit den zu einem bestimmten Zeitpunkt am häufigsten gesuchten Metadaten. Die Suche ist auch in der Übersicht unter 6.3 gezeigt.


Bei einer Suche wird ein angepasstes Layout angezeigt. Hier als Beispiel das Layout bei der Anzeige von Coverart:




moosr

Moosn bringt's!

[Go!](#) [Lyrics ▼](#) [1 Items ▼](#)




Provider: google
Size: 600x600
Source-URL: images.handy.de
Cached in MoosrDB™: False



Provider: lastfm
Size: 300x300
Source-URL: userserve-ak.last.fm
Cached in MoosrDB™: True

Textuelle Resultate werden folgendermaßen angezeigt:



moosr

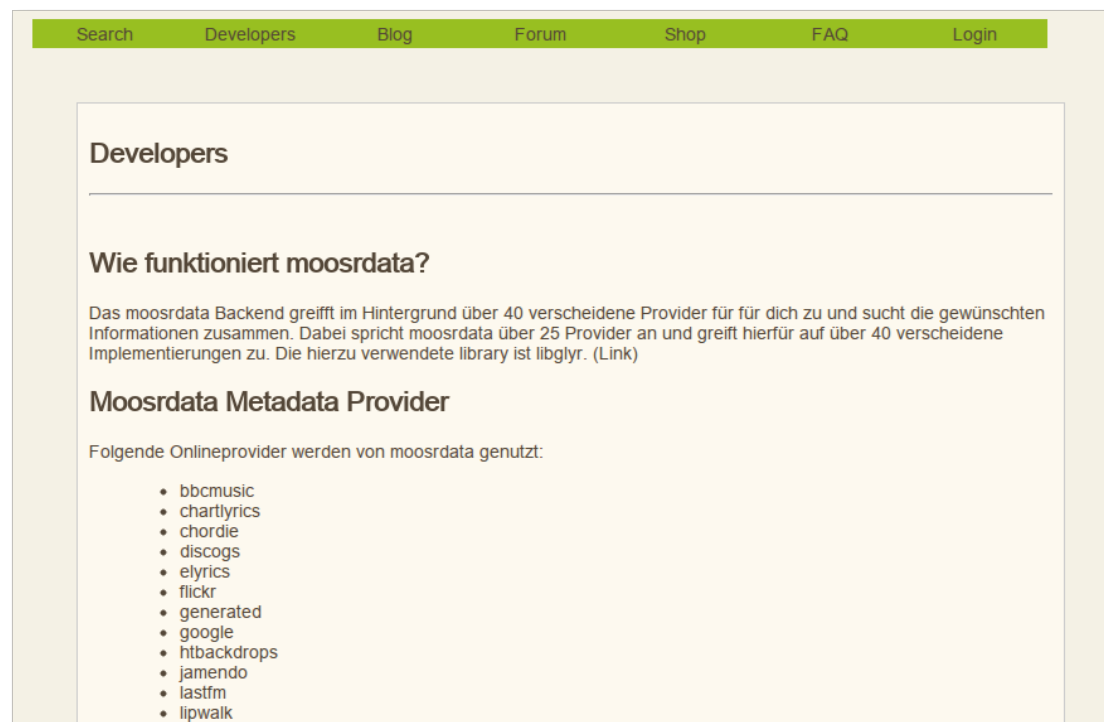
Moosn bringt's!

[Go!](#) [Lyrics ▼](#) [1 Items ▼](#)

Lieber Staat
Ich fühle mich so rundum wohl in dir
Lieber Staat
Es weht ein Wind von Freiheit hier
Du erklärst mir immer wieder, was erlaubt ist und was nicht
Lenkst mein Leben jeden Tag und bist furchtbar fürsorglich
Ach, was wär ich ohne dich

Danke, dass du mich regierst
Danke, dass du mich regierst
Und dass du mich nicht ignorierst

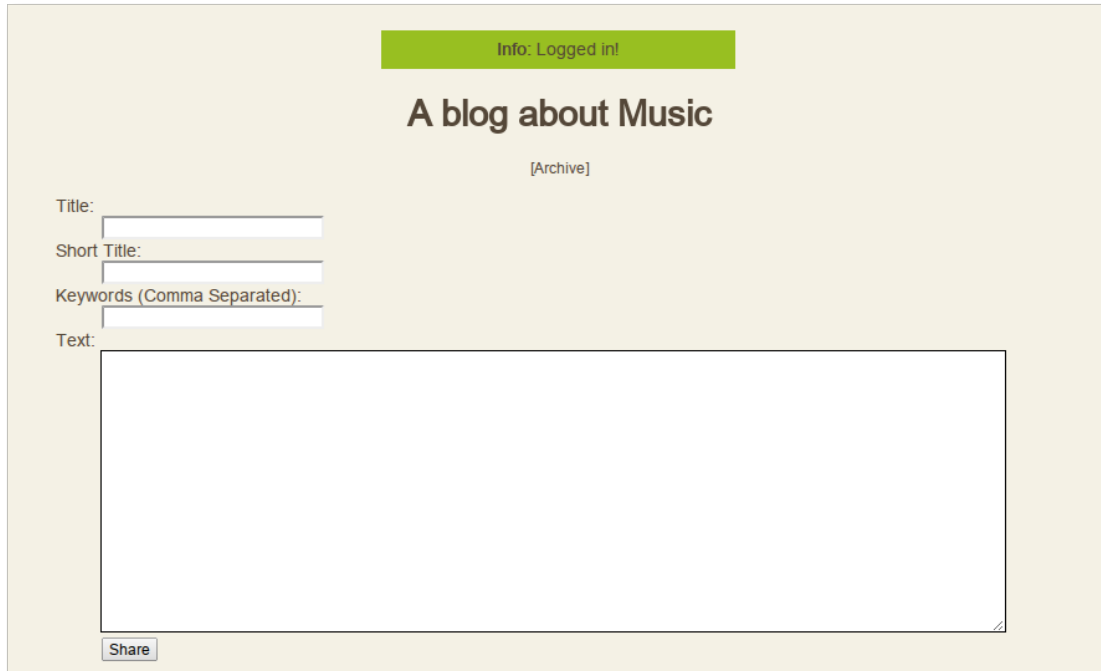
Developers



Die Kategorie *Developers* ist eine Seite mit statischem Content. Hier werden für interessierte Entwickler alle Informationen zu unserer HTML Darstellung und JSON API erläutert.

Blog Der *Blog* ist eine Seite mit wachsendem Content. Über diese Seite wollen wir unsere Community auf dem Laufenden halten über Newcomer aus der Szene und Berichte über Alben, Festivals etc. verfassen. Die „Blogsoftware“ wurde von uns in Flask implementiert damit diese die SEO Anforderungen optimal beherrscht.

Die Loginmaske erlaubt ein SEO-gerechtes Eintragen von Artikeln:



The screenshot shows a web form for creating a blog post. At the top, there is a green status bar that says "Info: Logged in!". Below this is the title "A blog about Music" and a link "[Archive]". The form contains four input fields: "Title:", "Short Title:", "Keywords (Comma Separated):", and "Text:". The "Text:" field is a large text area. At the bottom left of the form is a "Share" button.

Die Beiträge sind dabei in reStructuredText zu verfassen. Beim Posten werden diese zu HTML konvertiert. Hierbei ergibt sich ein fest vorgegebenes Header Schema was eine manuelle Definition der Header überflüssig macht. Siehe dazu die reStructuredText Dokumentation:

<http://docutils.sourceforge.net/docs/user/rst/quickstart.html>

Momentan enthält der *Blog* vier von uns verfasste Beiträge die ebenfalls gut von Google gerankt werden sollen. Hierbei wurde darauf geachtet, dass ein primäres und mehrere sekundäre Keywords wie gewünscht festgelegt wurden und diese mehrmals im Text wiederholt wurden. Diese wurden beim Verfassen des Blogeintrags als Keywords festgelegt und somit vom Blogsystem als Meta-Keywords in den HTML Quelltext geschrieben.

```
<meta name="keywords" content="Coppelius, Band, Kammercore, Klassik" />
```


Es folgt ein Screenshot der Blogübersicht. Es wird nicht der vollständige Beitrag angezeigt sondern jeweils nur ein Bild und ein Teaser:

A blog about Music

[Archive]

Das Niveau - Das lose Album: Eine Rezension


by Elch [Posted on 19/12/2012 - 20:23]



Die Band, die Barden Sie werden „Die Barden aus Berlin“ genannt und ihr Motto ist „Das Niveau sinkt für Sie“. Es geht um die beiden Jungs ... [\[READ MORE\]](#)

Bandvorstellung: Coppelius

by Elch [Posted on 19/12/2012 - 20:22]



In der Übersicht ist die Detailansicht des Beitrags verlinkt. Darin ist der vollständige Beitrag sichtbar:

Bandvorstellung: Coppelius by Elch [Posted on 19/12/2012 - 20:22]

Heute in unserer Vorstellungsrunde stellen wir euch die koppelianische Kappelle vor.

Geschichtliches

Laut eigener Aussage hat sich die Band vor 200 Jahren in Wien gegründet, damals noch als klassische Kammermusikkappelle. Allerdings wurden sie kurz darauf von einem Blitz getroffen und touren seitdem elektrisch verstärkt um das Weitenrund.



Das Archiv bietet eine Übersicht über alle vorhanden Posts, und listet das Datum, den Autor, den Kurztitel sowie die Keywords auf. Durch das Archiv sollen auch ältere, nicht mehr andersweitig verlinkte Posts Linkkraft erhalten.

Blog Archive				
Posted On	Short Titel	Author	Keywords	Full Titel
19/12/2012 - 20:23	review-dasniveau	Elch	Niveau, Barde, Lied	Das Niveau - Das lose Album: Eine Rezession
19/12/2012 - 20:22	band-coppelius	Elch	Coppelius, Band, Kammercore, Klassik	Bandvorstellung: Coppelius
19/12/2012 - 20:20	festival-knorkator	Elch	Knorkator, Band, Festival	Knorkator auf Haltestelle Woodstock Festival
19/12/2012 - 20:19	hello-world	Elch	Moosr, Community, Metadata, Blog, Indie	Moosr says Hello World!

Webshop In der Kategorie *Webshop* befinden sich Seitenlinks und Beschreibungen zu Webshops mit Tickets und Fanartikeln. Diese Seite enthält hauptsächlich statischen Content, daher wurde auf einen Screenshot verzichtet. Siehe 6.3.

Forum Das *Forum* ist der Haupttreffpunkt für die Community. Hier können sich alle nach Lust und Laune austauschen. Hier ist der Content dynamisch und wächst mit der Anzahl der Besucher und Beiträge. Als Forumsystem würde sich hier z.B. FluxBB

<http://fluxbb.org/>

gut eignen. Siehe <https://bbs.archlinux.de/> für eine beispielhafte Umsetzung mit FluxBB.

Da in unserer Version die Seite noch ein Platzhalter ist wurde auf ein Screenshot verzichtet.

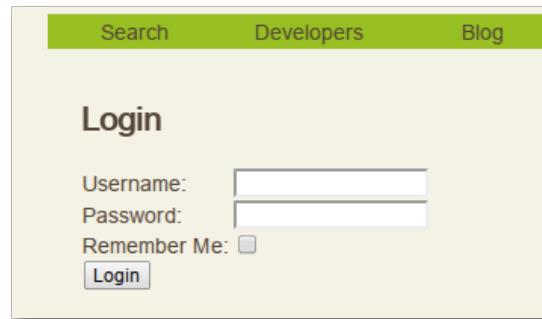
About Us In der Kategorie *About Us* erläutern wir dem „Kunden“ das Spektrum unserer Dienstleistung. Diese Seite ist zugleich unsere *Startseite*. Diese Seite enthält wieder statischen Content der unsere Dienstleistung klar und direkt beschreibt.

Die Seite enthält statischen Content, und ähnelt daher 6.3.

F.A.Q. Der *F.A.Q.* Bereich bietet Interessierten Antworten auf die häufig gestellten Fragen. Der Content ist statisch und ändert sich kaum.

Die Seite enthält statischen Content, und ähnelt daher 6.3.

Login

A screenshot of a web login form. At the top, there is a green navigation bar with three links: 'Search', 'Developers', and 'Blog'. Below this, the word 'Login' is displayed in a large, bold font. Underneath, there are three input fields: 'Username:', 'Password:', and 'Remember Me:'. The 'Remember Me:' field has a small checkbox next to it. At the bottom of the form, there is a 'Login' button.

Hier ist der *Loginbereich* der primär für *Redakteure* interessant ist die Beiträge auf der Blogseite veröffentlichen möchten. Auf der Loginseite selbst ist kein weiterer zusätzlicher Content vorhanden.

Jeglicher Content in allen Kategorien wird „per Hand“ verfasst. Kopieren von Texten ist verboten. Dadurch wollen wir die Originalität unseres Contents gewährleisten.

6.4. Footer-Bereich

In grafischer Hinsicht findet sich eine „Waldszene“. Diese haben wir vom „Gnome Reference Manual“ ¹ übernommen. Der Footer Bereich ist oben in der Übersicht abgebildet (Siehe 6.3). Im *Footer* befinden sich Links zu folgenden Seiten:

Social Im *Social* Bereich befinden sich Links auf die folgenden sozialen Netze in welche unsere Dienstleistung Vorganden ist:

- Flickr (Bildergalerien von Festivals und Community-Treffs, Bilder für den Blog)
- Google+ (Unser Auftritt bei Google+)
- Facebook (Unser Auftritt bei Facebook)
- Twitter (Unser Microblogging Seite um kurz über Änderungen auf unserer Seite zu informieren)

¹<http://developer.gnome.org/glib/>

- E-Mail Kontakt
- RSS Feed zum Blog

Siehe auf der linken Seite in der Übersicht.

Wir haben uns bewusst für die Auslagerung von großen Bildergalerien auf Flickr entschieden weil wir so Bilderreihen von Festivals oder Community-Treffs online stellen können ohne dass hierbei Datenvolumen-Kosten auf unserer Seite entstehen. Gleichzeitig sparen wir Speicherplatz auf unserem Server.

Die Einbindung von Flickr und den weiteren sozialen Netzen hat neben dem „Mundpropaganda Werbeeffect“ den Vorteil, dass hier gleichzeitig ein „Hotspot“ auf unsere Seite generiert wird was unser Google Ranking verbessert.

Sitemap Die Sitemap bietet nochmal für den Google-Crawler die Möglichkeit in alle wichtigen Bereiche unserer Seite zu kommen.

Featured Posts Unter „Featured Posts“ befinden sich einige von Hand ausgewählte Blogposts, die hier verlinkt werden.

External Links Hier sind Links zu den von uns verwendeten Libraries wie z.B. libgylr. Dadurch informieren wir nicht nur über die Technologien die wir verwenden sondern ermöglichen es der Community auch indirekt an moosr über die Entwicklung an Thirdparty-Libraries mitzuarbeiten.

Zudem finden sich hier externe Anbieter die unser Angebot ergänzen.

7. Google Optimierung

7.1. Duplicate Content vermeiden

Durch den Einsatz von Flask als Webframework haben wir volle Kontrolle über die Festlegung einzelner URLs. Die Definition einer View-Funktion erfordert die Angabe einer URL. Doppelte URLs sind in Flask nicht erlaubt. Innerhalb der Templates ist dabei nur eine Referenzierung dieser URL via der *url_for* Funktion möglich. Es folgt ein Beispiel für eine View Funktion, und ein Fragment eines Templates:

```
# File: app.py
# Aufruf bei GET www.moosr.org/search
@app.route('/search')
def view_search_page():
    return render_template('search.html')

<!-- File: search.html -->
<!-- {{ Text }} wird durch Template Engine ersetzt -->
<a href="{{ url_for('view_search_page') }}">Link to Search Page</a>
<!-- ... -->
```

Durch die konstante Festlegung und dem konsistenten Zugriff wird effektiv externer Duplicate Content vermieden.

Um internen Duplicate Content zu vermeiden wird als Leitlinie stets zwischen Beiträgen verlinkt. Texte zu kopieren ist verboten. Desweiteren ist auf unserer Seite ohnehin der Content jeweils unabhängig voneinander.

Der einzige Fall in dem Interner Duplikat Content vorkommt ist die verkürzte Blogansicht, bei der nur ein Teaser des Blogbeitrags vorhanden ist. Die Blogübersicht ist

allerdings mittels des meta-Tags aus der Indexierung ausgenommen. Der Link zu den eigentlichen Einträgen wird hingegen befolgt!

```
<meta name="robots" content="noindex, follow, noarchive">
```

7.2. Content Seiten über Navigation erreichbar

„Punkt-Punkt-Punkt-Problem“ Wir haben keine Seite bei der eine Unterteilung in Subseiten zwingend notwendig ist. Für eine nähere Begründung siehe Punkt 7.3.

„Seite zurück/Seite vor“-Problem Um zu vermeiden, dass ältere Artikel ein schlechteres Ranking bekommen als neue Artikel werden Links zu alten Artikeln in einem Archiv „aufbewahrt“, welches wiederum auf der Blogübersicht verlinkt ist.

7.3. Artikel über mehrere Seiten strecken

Aus Gründen der Usability haben wir auf den Punkt verzichtet. Unsere Blogposts werden in der Übersicht nur mittels eines *Teasers* gezeigt, bei dem auf eine Detailansicht verlinkt ist. Zudem verlinken wir innerhalb des Archives auf alle Blogposts.

Momentan wächst die Postübersicht bis ins theoretisch „unendliche“ an. Unsere Seite bietet sonst kein Content an der sich auf mehrere Seiten ausbreiten ließe.

7.4. Festlegbare Linktexte

Dieses Verhalten wurde vollkommen implementiert. Siehe hier den Screenshot zum Blog unter Kapitel 6.3.

Beispielsweise kann man bei Blogbeiträgen einen langen und einen kurzen Titel bestimmen. Der lange Titel wird als Überschrift für die Detailansicht und für die Blogübersicht benutzt. Der kurze Titel wird in der Url und innerhalb von den Link-Texten benutzt.

Unwichtige Seiten wie beispielsweise das Impressum haben im Headtag ein Metatag stehen:

```
<meta name="robots" content="noindex, follow, noarchive">
```

Dadurch werden diese vom Indizieren ausgenommen, die darin enthaltenen Links werden aber dennoch ausgewertet.

7.5. Einfache URLs erzeugen

Unsere URL-Struktur basiert auf folgenden Kriterien:

- keine Query-Parameter via *?key=value*
- beschreiben die Seite kurz und prägnant.
- keine IDs in der URL. Kurze Titel sind zu bevorzugen.
- möglichst kurz, ohne viele Zwischenebenen.

Beispiele aus unserer URL-Struktur:

1. */blog* Blogübersicht.
2. */blog/entry/knorkator* Blogeintrag mit dem Kurztitel „Knorkator.“ Der tatsächliche Name des Eintrags ist dabei länger. Der kurze Titel eignet sich für Linktexte und als Bestandteil der URL.
3. */developers, /about-us* Statische Contentseite.
4. */search* Dynamische Metadatensuchseite.
5. */query/artistphoto/1/Avril%20Lavigne* Anzeige der Suchresultate eines Künstlerbildes von Avril Lavigne.
6. */api/artistphoto/1/Avril%20Lavigne* Wie oben, aber Anzeige eines JSON Dokuments, statt einer HTML Repräsentation.

7.6. Editierbare Metadaten

Jeder Blogeintrag hat einen definierten langen Titel, einen kurzen, prägnanten Titel der für Linktexte und URLs eingesetzt werden kann, sowie einige Meta-Tags:

Statische Seiten haben ähnliche Attribute, haben allerdings keinen kurzen Titel. Ein Beispiel:

```
<!-- Kurzbeschreibung der Seite -->
<meta name="description" content="Guide for API Usage" />
<!-- Keywords innerhalb des Contents -->
<meta name="keywords" content="API, Manual, Reference, Metadata, Developer" />
<!-- Erlaube ausdrücklich Indizierung -->
<meta name="robots" content="all"/>
```

7.7. Übersicht über Artikel/Keywords

Es folgt eine Übersicht über die optimierten Artikel mit den dazugehörigen Keywords. Das zu erst erwähnte Keyword ist dabei das primäre Keyword. Zudem findet sich in Klammern hinter dem primären Keyword die Anzahl desselben im Text.

7.7.1. Developers:

URL: <http://moosr.nullcat.de/developers>

Keywords: API (3), Manual, Reference, Metadata, Developer

7.7.2. FAQ:

URL: <http://moosr.nullcat.de/faq>

Keywords: Moosr (7), Musik, Fragen, Suchfeld

7.7.3. Shop:

URL: <http://moosr.nullcat.de/shop>

Keywords: Konzert (4), Festival, Links, Merchandise

7.7.4. About us:

URL: <http://moosr.nullcat.de/about-us>

Keywords: Musik (7), Community, Indie, Fan

7.7.5. Suche nach Lyrics:Farin Urlaub + Lieber Staat

Stellvertretend für andere Such-Seiten ausgewählt.

URL: <http://moosr.nullcat.de/query/lyrics/1/Farin%20Urlaub%20%2B%20Lieber%20Staat>

Keywords: Lyrics (2), Farin Urlaub (1), Lieber Staat

7.7.6. Blogübersicht:

URL: <http://moosr.nullcat.de/blog>

Keywords: Indie, Band, Artist, Festival

7.7.7. Blogpost - Bandvorstellung: Coppelius:

URL: <http://moosr.nullcat.de/blog/entry/band-coppelius>

Keywords: Coppelius (4), Band, Kammercore, Klassik

7.7.8. Blogpost - Knorkator auf Haltestelle Woodstock Festival:

URL: <http://moosr.nullcat.de/blog/entry/festival-knorkator>

Keywords: Knorkator (5), Band, Festival

7.7.9. Blogpost - Das lose Album - Eine Rezession:

URL: <http://moosr.nullcat.de/blog/entry/album-das-niveau>

Keywords: Niveau (7), Barde, Lied Das Niveau

7.7.10. Blogpost - Moosr says Hello World:

URL: <http://moosr.nullcat.de/blog/entry/hello-world>

Keywords: Community (2), Metadata, Blog, Indie

8. Technische Implementierung

8.1. Vorwort - technische Implementierung

Als technische Grundlage für moosr wird ein Webframework verwendet, da dieses an die benötigten Bedürfnisse (Suchmaschine) am relativ gut angepasst werden kann.

Als Webframework haben wir uns mit Absprache mit Herrn Dr. Alois Kastner-Maresch auf das Python Microwebframework Flask geeinigt da wir das Framework in der Vorlesung *Wiederverwendungsbasierte Entwicklung von Systemen* in einer Studienarbeit erarbeiteten und das dort gewonnene Wissen (Flask und Python) praktisch in der Vorlesung „Webtechnologie und Webmarketing mit Open Source“ umsetzen möchten.

8.2. Verwendete Frameworks

8.2.1. Flask

Das Microwebframework benutzt *Jinja* als Template Renderengine und *Werkzeugs* als WSGI Middleware. Die Pakete selbst können mit pip gezogen werden.

Da die Webpräsentation primär als Dienstleistung zu sehen ist kommen bei der Implementierung neben dem Framework selbst hauptsächlich auf die beiden libraries *libglyr* und *sqlite3* zum Einsatz.

8.2.2. libglyr

Libglyr ist eine library zum auffinden von Musikmetadaten.

<https://github.com/sahib/glyr>

8.2.3. sqlite3

Sqlite3 ist eine sehr weit genutzte und bekannte embedded Datenbank, welche für unseren Einsatz optimal ist.

<http://www.sqlite.org/>

8.3. Historisches

Wie bereits erwähnt ging die aktuelle Seite aus einer Studienarbeit im Fach „Wiederverwendungsbasierte Entwicklung von Systemen“ hervor. Die Metadatensuchmaschine wurde im Rahmen dieser implementiert. Siehe Screenshot:



8.4. Implementierung

In Kapitel 6 (Siehe 6) sind bereits viele technische Details erläutert.

Tagcloud Die implementierte Tagcloud holt sich immer aus der Datenbank die zu einem bestimmten Zeitpunkt am meist gesuchten Schlagwörter raus. Dadurch wird die Linkkraft auf bestimmten Ergebnisse dynamisch angepasst wodurch möglicherweise Google-User die nach diesem aktuell populären Ergebnis googlen auf unsere Seite geleitet werden.

9. Weiteres

9.1. Github-Repository

Bei Interesse steht der vollständige Sourcecode auf Github zur Verfügung.

`https://github.com/studentkittens/flascat`

9.2. Filmeabend

Gegen Ende der Fertigstellung unserer Seite sind wir zufälligerweise auf der Suche nach einem Film, für einen gemütlichen Filmeabend unter Studenten, auf folgende Seite über die Google-Suche „find similar movies“ gestoßen:

`http://www.tastekid.com/`

Wir fanden es wichtig diese hier mit aufzunehmen weil die Seite vom Aufbau wie auch von der API her sehr viele Ähnlichkeiten zu unserer Dienstleistung hat und bei der Suche nach „similar movies“ diese direkt auf dem 1. Platz gelistet wird. Hier hat man es anscheinend ohne übermäßig viel Content auf der Seite unterzubringen problemlos geschafft durch direkte und sinnvolle Informationen ein gutes Ranking zu erreichen.