



Projekt Studiengang Medieninformatik

Codeload

Download-Tool für Musik-/Plattenlabels

Autor: Heiko Stumpf

Betreuer: Prof. Dr. Fridtjof Toenniessen

Wintersemester 2010

Erstellt am: 20. Februar 2010

Hochschule der Medien Stuttgart



Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	1
1.2	Codeload	2
1.3	Codes	3
1.4	Download-Site	3
2	Projektmandat	5
2.1	Projektdefinition	5
2.1.1	Projektzielsetzung	5
2.1.2	Scope	5
2.1.3	Abgrenzung	5
2.1.4	Rahmenbedingungen	5
2.1.4.1	Organisation	5
2.1.4.2	Technik	6
2.2	Beziehung zu anderen Projekten	6
2.3	Qualitätserwartungen	6
2.4	Business Case	6
2.5	Toleranzen	8
2.6	Referenzmaterial	8
2.7	Vorschlag Projektorganisation	8
3	Projektbeschreibung	9
3.1	Hintergrund des Projektes	9
3.2	Projektdefinition	9
3.2.1	Scope	9
3.2.2	Projektziele	10
3.2.3	Umfang und Abgrenzung	10
3.2.4	Einschränkungen	11
3.2.5	Rahmenbedingungen	11
3.2.5.1	Organisation	11
3.2.5.2	Technik	11
3.2.6	Projektorganisation	12
3.2.7	Schnittstellen zu anderen Projekten	12
3.3	Business Case	12
3.4	Kunden	13
3.4.1	Kundenerwartungen	13
3.5	Abnahmekriterien	14
3.6	Kosten / Budget	14
3.7	Projekttoleranz	14
3.8	Risiken	15
3.9	Entscheidungsvorschlag	15

4	Projettlösungsansatz	16
4.1	Erforderliche Qualifikationen	16
4.2	Alternativen	16
4.3	Infrastruktur	16
4.3.1	Entwicklungsinfrastruktur	16
4.3.2	Produktivinfrastruktur	17
4.4	Eigenentwicklung und fertige Lösungen	17
4.4.1	Frameworks	17
4.4.2	Content- Management-Systeme	17
4.4.3	Architektur Entscheidung	17
4.4.4	Entwicklungsprinzipien	18
4.5	Security und Vertraulichkeit	18
4.6	Wartungs- und Betriebsanforderungen	18
4.6.1	Betriebsanforderungen	18
4.6.2	Wartungsanforderungen	19
4.7	Vereinbarungen	19
5	Produktbeschreibung	20
5.1	Produktstrukturplan	20
5.2	Produktflussdiagramm	21
5.3	Managementprodukte	22
5.4	Spezialistenprodukte	22
5.5	Architektur/Framework	22
5.5.1	Codeload WebPortal	23
5.5.1.1	Sessions	23
5.5.1.2	Benutzer- & Rechteverwaltung	23
5.5.1.3	Klasse Session Manager	24
5.5.1.4	Content-Verwaltung	25
5.5.1.5	Codeload Wizard	25
5.5.1.6	Klasse Wizard	26
5.5.1.7	GUI / Benutzeroberfläche / Navigation	26
5.5.2	Codes	28
5.5.2.1	Klasse Code	28
5.5.3	Download-Site	29
5.5.4	Download Engine	29
5.5.5	Template System	30
5.5.6	Klassendiagramme	31
5.5.7	Datenbank	32
5.6	Qualitätsmanagement	33
5.6.1	Testing	33
5.6.2	Codekriterien/Notationen	33
5.6.3	Projektdateistruktur	34
6	Projektplan	35
6.1	Projekt Produkte	35

6.1.1	Codeload Version 1.0	35
6.1.2	Präsentation	35
6.1.3	Dokumentation.....	35
6.2	Gant-Diagramm	36
6.3	Change Management	37
7	Risikomanagement	37
8	Kommunikation	39
8.1	Kommunikationsplan.....	39
8.2	Sitzungsprotokolle	39
9	Fazit.....	41
9.1	Probleme	41
9.2	Zusammenarbeit	42
9.3	Projektmanagementtool	42
9.4	Schlussfazit	43
10	Quellangaben	45
10.1	Internet.....	45
10.2	Bücher	45
10.3	Freie Bibliotheken.....	45
10.4	Programme und Tools	45

1 Einführung

1.1 Motivation

Viele Musiklabels beschwerten sich seit den letzten Jahren immer häufiger über die rückgängigen Verkaufszahlen von Tonträgern, die unter anderem durch illegale Musik-Downloads¹ im Internet verursacht werden. Dies schadet in erster Linie nicht nur den Musiklabels sondern auch dem Künstler an sich.

Anstatt den Trend zu erkennen, haben sich viele Musiklabels zunächst geweigert, den Vertrieb ihrer Musik auf das Internet umzulegen. Das Potenzial, dass im Verkauf von Musikdateien über das Internet liegt, wurde viel zu spät erkannt und das Internet viel zu lang als Feind anstatt als Vertriebs- und Marketingplattform angesehen. Inzwischen ist den Musiklabels bewusst geworden, dass das Internet immer stärker den regulären Verkauf von Tonträgern in den Geschäften verdrängen und ablösen wird. Darüber hinaus bietet das Internet eine gute Möglichkeit, effizientes Marketing für neue Künstler und deren Lieder zu betreiben.

Oft ist es immer noch so, dass eine strikte Trennung zwischen einem Web-Angebot für Musik (z.B. iTunes Store, musicload etc.)² und dem Verkauf von Tonträgern in Geschäften stattfindet. Es kommt zwar vor, dass Free-Downloads beim Kauf eines Tonträgers angeboten werden, doch entstehen oft Probleme beim Anbieten dieser Downloads, da in den meisten Fällen keine zentrale Plattform zum Erstellen und Verwalten dieser Downloads existiert. Fehler im Druck (z. B. bei Download-Code, Links, etc.) oder die Möglichkeit für Missbrauch sind keine Seltenheit. Darüber hinaus verursacht die Abstimmung zwischen dem Druck der Download-Codes und der Online-Bereitstellung der Musikdateien über verschiedene Plattformen zusätzliche Kosten und einen nicht unwesentlichen Mehraufwand an Verwaltungsarbeit, da viele Abläufe noch von Hand durch mehrere Mitarbeiter erfolgen.

Hier setzt **CodeLoad** an und bietet den Musiklabels eine zentrale Plattform für alle Vorgänge rund um die Erstellung und Verwaltung der Codes³ und Download-Sites⁴. Dadurch lassen sich Kosten sparen und die Fehlerraten stark minimieren.

¹ engl. Download: Übertragen Dateien aus dem Internet/Netzwerk auf den eigenen Computer(Client)

² Online Portale für entgeltliche Bereitstellung von Musik

³ Zufällige generierte Zeichenfolge z.B.: X4f7G8

⁴ Webseite auf der der Kunde z.B. Musik downloaden kann

1.2 Codeload

Die Ursprüngliche Idee für **Codeload** entstand im März 2009, als im Rahmen einer CD/Schallplatten Veröffentlichung der Band CrisisNeverEnds nach Möglichkeiten gesucht wurde, für potenzielle Käufer einen Kaufanreiz zu schaffen und für Fans der Band zusätzliches Material zur Verfügung zu stellen. Da es sich hierbei um ausschließlich digitales Material (Mp3⁵ und PDF⁶) handelte, sollte dieses zum Download angeboten werden.

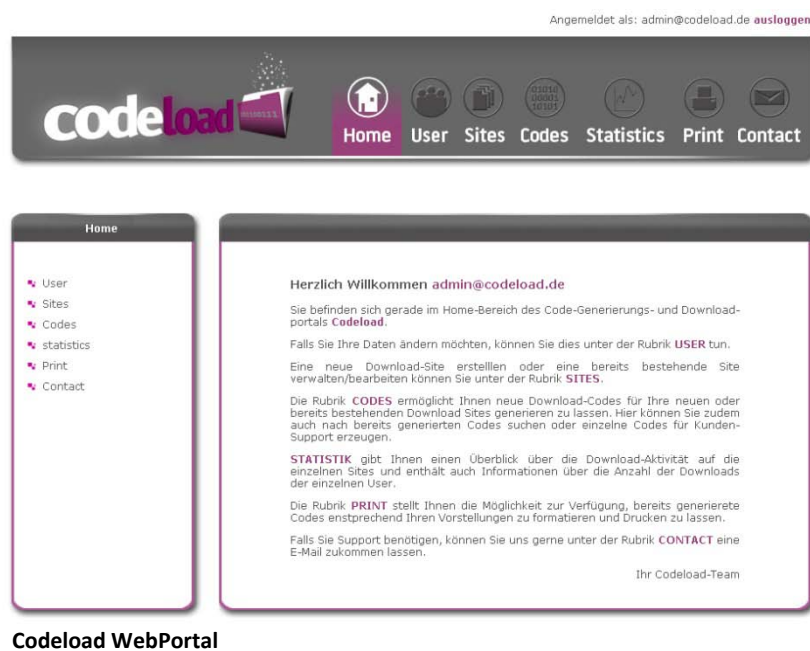
Um den Download vor unbefugten Zugriffen zu schützen und nur den Käufern der Tonträger den Download zu ermöglichen, sollte dem Tonträger ein Code beigelegt werden, der für den Download berechtigt. Nach eingehender Recherche konnte leider kein Anbieter im Internet gefunden werden welcher einen Dienst anbietet, der die genannten Kriterien erfüllt. Somit musste die Lösung selbst implementiert werden.

Aufgrund der „Einmaligkeit“ des Projekts wurde viel manuell bewerkstelligt. Die Codes wurden über ein selbst entwickeltes C++ Programm erstellt und in eine MySQL⁷ Datenbank importiert. Für die Download Verifizierung wurde eine simple HTML Seite mit einer Eingabemaske für den Code in PHP⁸ geschrieben. Die HTML Seite, in der weiteren Dokumentation „Download-Site“ genannt, konnte den eingegeben Code mit der Datenbank abgleichen und bei Korrektheit des Codes, die Dateien zum Download freigeben.

Nach der Veröffentlichung des Tonträgers gab es Anfragen von kleineren und mittelständischen Musik- und Plattenlabels, die für Ihre zukünftigen Veröffentlichungen ebenfalls diese Art von Downloads anbieten wollten. Allerdings mussten diese Anfragen abgelehnt werden, da es zeitlich nicht möglich war eine Automatisierung des Vorgangs zu entwickeln beziehungsweise, die Entwicklung für jede Veröffentlichung zu portieren und anzupassen. Aufgrund nicht abnehmender Anfragen und Interesse Seitens der Musiklabels soll dieser Automatisierungsvorgang nun als studienbegleitende Projektarbeit umgesetzt werden.

Codeload vereint nun alle oben genannten Entwicklungsschritte in einer Webplattform und ermöglicht es dem Label nach erfolgreicher Anmeldung innerhalb von 2 Minuten eine Download-Site nach eigenen Wünschen zu

Angemeldet als: admin@codeload.de [ausloggen](#)



Codeload WebPortal

⁵ Komprimiertes Audio Datenformat zur Wiedergabe von Musik

⁶ Portable Document Format: Plattform unabhängiges Dateiformat für Dokumente

⁷ relationales Datenbankverwaltungssystem

⁸ Hypertext Preprocessor, Scriptsprache für dynamische Webseiten

gestalten, eine Vielzahl von Codes für diese Download-Site zu erstellen, Dateien zum Downloaden bereitzustellen und die Codes auf Etiketten bzw. Flyer zu drucken oder in XML Form einer Druckerei zugänglich zu machen. Alle weiteren Funktionen wie z.B. die Anzeige von Statistiken über alle loads werden im weiteren Verlauf dieser Dokumentation beschrieben.

1.3 Codes

Um Downloads zu schützen bzw. nur einer bestimmten Gruppe von Usern zugänglich zu machen, bedarf es einer gewissen Art der Authentifizierung um für einen Download berechtigt zu sein. Für **CodeLoad** wurde auf das Prinzip der Code Verifizierung zurück gegriffen. Ist der Kunde im Besitz eines gültigen Codes, der ihm im Voraus z.B. durch den Kauf eines Tonträgers ausgehändigt/beigelegt wurde, wird der Download anhand des Codes gestartet. Der Code dient hierfür als eine Art Ticket.

Natürlich gehören hierzu auch Sicherheitsmaßnahmen um den Missbrauch wie z.B. die Weitergabe an Dritte zu verhindern.

Diese Sicherheitsmaßnahmen werden ebenfalls im weiteren Verlauf dieser Dokumentation ausführlich beschrieben.



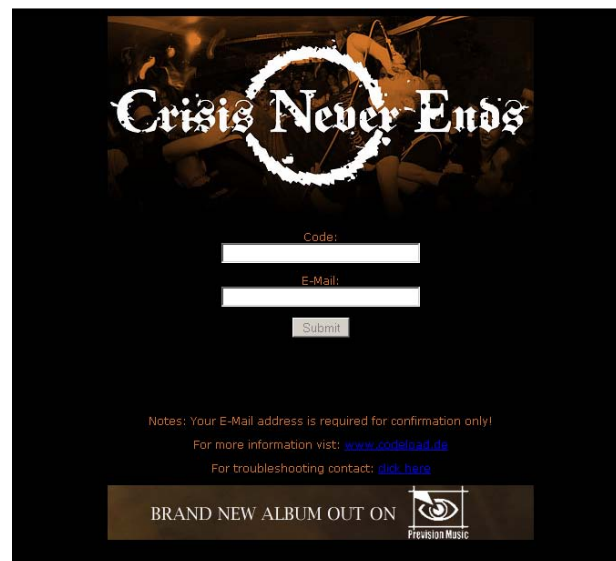
Booklet eines Tonträgers mit Download-Code

1.4 Download-Site

Um den Code einzulösen, benötigt der Kunde eine Webseite, die es ihm ermöglicht, den Code über eine Eingabemaske einzugeben und den Download zu starten. Das eigentliche **CodeLoad**-Portal für die Erstellung dieser Seiten sieht der Kunde nicht, für ihn ist allein die Download-Site der zentrale Anlaufpunkt um den Download zu starten.

Die Download-Site veranlasst die Prüfung des eingegeben Codes, interagiert hierbei mit der Datenbank in der die Codes gespeichert sind und gibt bei Gültigkeit den Download, der für die Download-Site hinterlegt ist, frei.

Ein Hauptaufgabenmerk liegt hier besonders auf der Gestaltung der Download-Site. Diese lässt sich individuell gestalten. Mit Schrift- und Hintergrundfarben, Bildern und Werbebannern durch die das Label die Site personalisieren und an ein Corporate Design des Künstlers anpassen kann.



Download-Site mit Eingabefeld für Code und Email Adresse

Auf die weiteren Details der Download-Sites, deren Erstellung und Personalisierung wird im weiteren Verlauf der Dokumentation ausführlich eingegangen

2 Projektmandat

2.1 Projektdefinition

Viele Musiklabels sind auf der Suche nach neuen Lösungen für den Download von Musikdateien. Einige Musiklabels bieten zusätzlich an, dass der Kunde beim Erwerb eines Tonträgers (CD, Schallplatte, usw.), den dieses Label vertreibt, einen Code ausgehändigt bekommt, mit dem er oder sie weitere Lieder oder andere digitale Medien des Künstlers auf einer Website downloaden kann. Für diese Musik-/Plattenlabels soll ein "Download-Tool" entwickelt werden, das es ihnen ermöglicht, die Download-Codes über dieses Tool einfach und schnell generieren zu lassen. Zusätzlich soll es dem Label ermöglicht werden, diese Codes über das Tool zu verwalten und auch auszudrucken. Darüber hinaus soll es auf dieser Webplattform auch möglich sein, einzelne Download-Seiten für die Tonträger zu erstellen und zu verwalten. Ziel ist es, ein Online-Portal für Code-Generierung und Musikdownloads zu entwickeln und bereitzustellen, das sich durch leichte Bedienbar- und hohe Benutzerfreundlichkeit auszeichnet.

2.1.1 Projektzielsetzung

Das Projektziel ist die Entwicklung eines Online-Download- und Code-Generierungsportals für Musikdateien. Da es sich bei dem Produkt um eine Webanwendung handelt, muss zudem ein hohes Maß an Sicherheit dieser Anwendung gewährleistet sein. Die Kunden eines Musiklabels sollen mit Hilfe dieser Webanwendung in der Lage sein, nach Eingabe eines Codes schnell und unkompliziert Musikdateien oder andere digitalen Formate von einem Server herunterladen zu können.

2.1.2 Scope

Das Projekt besitzt ein definiertes Ende und einen Starttermin, der sich aus der Genehmigung des Projektbetreuers ergibt. Innerhalb dieses Zeitraums muss das Projekt bearbeitet und abgeschlossen werden. Der Abgabe- und Präsentationstermin ist der 28.1.2010. Die Ziele sind aus der Projektzielsetzung 2.1.1 zu entnehmen.

2.1.3 Abgrenzung

Das Projekt umfasst nicht die folgenden Punkte:

- Suchmaschinenoptimierung
- Werbe- und Akquisefunktion
- Entwickeln eines Preisbildungsmodell für die Umlage der Kosten auf die Kunden

2.1.4 Rahmenbedingungen

2.1.4.1 Organisation

Das Projekt muss in einem zeitlichen beschränkten Rahmen durchgeführt werden. Für die Abgabe- und Präsentation ist Termin der Media Night an der Hochschule der Medien in Stuttgart am

28.1.2010 vorgesehen. Um das Projekt erfolgreich abzuschließen, muss dieser Termin eingehalten werden.

2.1.4.2 Technik

- Hardware:
Die Entwicklung des Projekts wird allein mit privater Hardware bewerkstelligt.
- WebServer:
Um das Portal betreiben zu können, wird ein WebServer mit PHP 5 Interpreter und MySQL Datenbank benötigt. Hierfür wurde extra eine Domain (www.codeload.de) gemietet um die abschließenden Tests unter realen Bedingungen durchführen zu können und das Portal später betreiben zu können.
- Software
Für die Entwicklung des Webportals und der Gestaltung des Designs wurde ausschließlich Open Source Software eingesetzt (u.a. Ubuntu 9.10, NetBeans 6.8, GIMP 2.6)
- W3C
Während der Entwicklung soll vor allem auch darauf geachtet werden, das Portal W3C konform zu entwickeln und vorgeschriebene Standards einzuhalten (XHTML usw.)

2.2 Beziehung zu anderen Projekten

Bis zu diesem Zeitpunkt sind keine Beziehungen zu anderen Projekten bekannt und im weiteren Projektverlauf auch nicht vorgesehen.

2.3 Qualitätserwartungen

Für Web-Anwendungen gibt es mittlerweile definierte Standards, die auch für dieses Projekt eingehalten werden müssen, um die Zufriedenheit und Qualitätsanforderungen der Kunden zu entsprechen. Folgende Qualitätserwartungen sollen durch dieses Projekt abgedeckt werden:

- Web-Anwendung muss fehlerfreie Funktion aufweisen
- Web-Anwendung muss alle gängigen Browser unterstützen
- Schutz der Kundendaten vor unbefugtem Zugriff
- Einfache und intuitive Bedienbarkeit
- Ladezeiten sollen möglichst gering gehalten werden, gute Performance

2.4 Business Case

Die Entwicklung von **CodeLoad** dient in erster Line der Entwicklung und Schulung des Entwicklerteams. Da ein großes Interesse an Webtechnologien wie PHP und JavaScript besteht, ist die Umsetzung dieses Projekts prädestiniert um einen tieferen Einblick in Design-, Architektur- und Performance Konzepte für dynamische Webanwendungen zu erhalten. Des Weiteren soll nicht außer Acht ge-

lassen werden, dass neben der erfolgreichen Entwicklung eines Projekts, auch Erfahrungen im Bereich des Projektmanagements gesammelt werden sollen.

Jedes neue Projekt bringt Herausforderungen mit sich, die im Voraus meistens nicht bekannt sind und gelöst werden müssen. Daher sollte das Entwickler-Team nicht nur technische Aspekte und Hintergründe erlernen, sondern auch die Art und Weise, Probleme anzugehen und erfolgreich zu lösen.

Das Prinzip Downloads anzubieten ist nichts Neues. Der Markt ist mittlerweile überflutet von einer Vielzahl an Sharehostern⁹ (rapidshare, megaupload usw.). Diese Sharehoster bieten dem User die Möglichkeit, Dateien hochzuladen und als Download anderen Internet-Usern zur Verfügung zu stellen. Es ist keine Seltenheit, dass sich diese Download Links nach kürzerer Zeit in unzähligen Foren befinden und jeder Internet-User diese Dateien herunterladen könnte. Hier findet keine Berechtigungsverifizierung statt und Urheberrechtsverletzungen sind nicht auszuschließen.

Die Verifikation eines Downloads über einen Code stellt ebenfalls keine neuartige Idee dar. Allerdings gibt es im Internet keinen Sharehoster, der jedem interessierten Kunden die Möglichkeit anbietet Downloads zur Verfügung zu stellen und diese anhand eines Codes freizuschalten. Diese Dienstleistungen müssen seit je her speziell für einen Kunden implementiert und in sein System (Webseite, Webshop o.ä.) implementiert werden. Es fehlt also ein Anbieter der diese Dienstleistung anhand einer Webplattform global anbietet.

CodeLoad bietet in erster Linie Musik- und Plattenlabels diesen Dienst an, um ihre Veröffentlichungen optimal vermarkten und vertreiben zu können. Theoretisch kann jeder Veröffentlichung zusätzliches digitales Material zum Download angeboten werden, welches über den beigelegten Code heruntergeladen werden kann. Für jede Veröffentlichung oder Kampagne des Musik-/Plattenlabels wird eine spezielle Download-Site erstellt, die auf das Design des Tonträgers angepasst werden kann.

Um die Schnittstelle zwischen Erstellung der Codes und der endgültigen Verteilung an die Kunden optimal zu unterstützen, wurde speziell für Druckereien ein XML Export entwickelt. Die Codes können nach dem Druck des Booklets¹⁰, in einem zweiten digitalen Druckverfahren auf das bereits gedruckte Booklet aufgedruckt werden (personalisierter Druck).

Durch den Schutz der Downloads über einen Code wird die unfreiwillige Verbreitung der Dateien verhindert, wodurch keine Urheberrechtsverletzung im Bereich der öffentlichen Zugänglichmachung entstehen. Das Musik- oder Plattenlabel, welches die Nutzungsrechte an der Verbreitung vom Urheber hält, kontrolliert auf diesem Wege die Verbreitung und die Bereitstellung an den urhebergeschützten Materialien jedes Künstlers.

Um die entstehenden Kosten zu decken, die durch anfallenden Traffic und monatliche Domain bzw. Webspace Kosten entstehen, soll ein Tarif Modell entwickelt werden, welches erlaubt die Kosten auf die Kunden und deren Nutzerverhalten umzulegen. Das Tarif-Modell ist jedoch nicht Teil dieses Projekts und wird erst im produktiven Betrieb des Projektes umgesetzt.

⁹ Internetdienstleister die sich auf den Download größerer Dateien spezialisiert haben

¹⁰ Beiheft eines Tonträgers mit Informationen über den Künstler, Songtexte usw.

2.5 Toleranzen

Geplant ist es, das Projekt nach seinen definierten Zielen zu implementieren und abzuschließen. Sollten während der Entwicklungsphase Schwierigkeiten bzw. zeitlich kritische Probleme auftreten, können diese abgeändert werden. Das Projektziel darf dadurch nicht gefährdet werden.

2.6 Referenzmaterial

Wie schon in der Einleitung beschrieben, stammt diese Projektidee aus einer bereits produktiven Version, die hier als Prototyp betrachtet werden soll. Dieser Prototyp dient als grober Leitfaden für die spätere Produkt Entwicklung. Hierbei wird allerdings nur die Idee verwendet. Schon entwickelte Module des Prototyps fließen nicht in dieses Projekt mit ein, da die Qualität und Sicherheit des Codes nicht ausreichend ist.

2.7 Vorschlag Projektorganisation

Rolle:	Name	E-Mail Adresse
Projektbetreuer	Prof. Dr. Fridtjof Toenniessen	toenniessen@hdm-stuttgart.de
Projektmanager	Tomislav Biscic	tb051@hdm-stuttgart.de
Projektteam	Heiko Stumpf	hs046@hdm-stuttgart.de
Anvisierte Kunden	Musik- und Plattenlabels	-

3 Projektbeschreibung

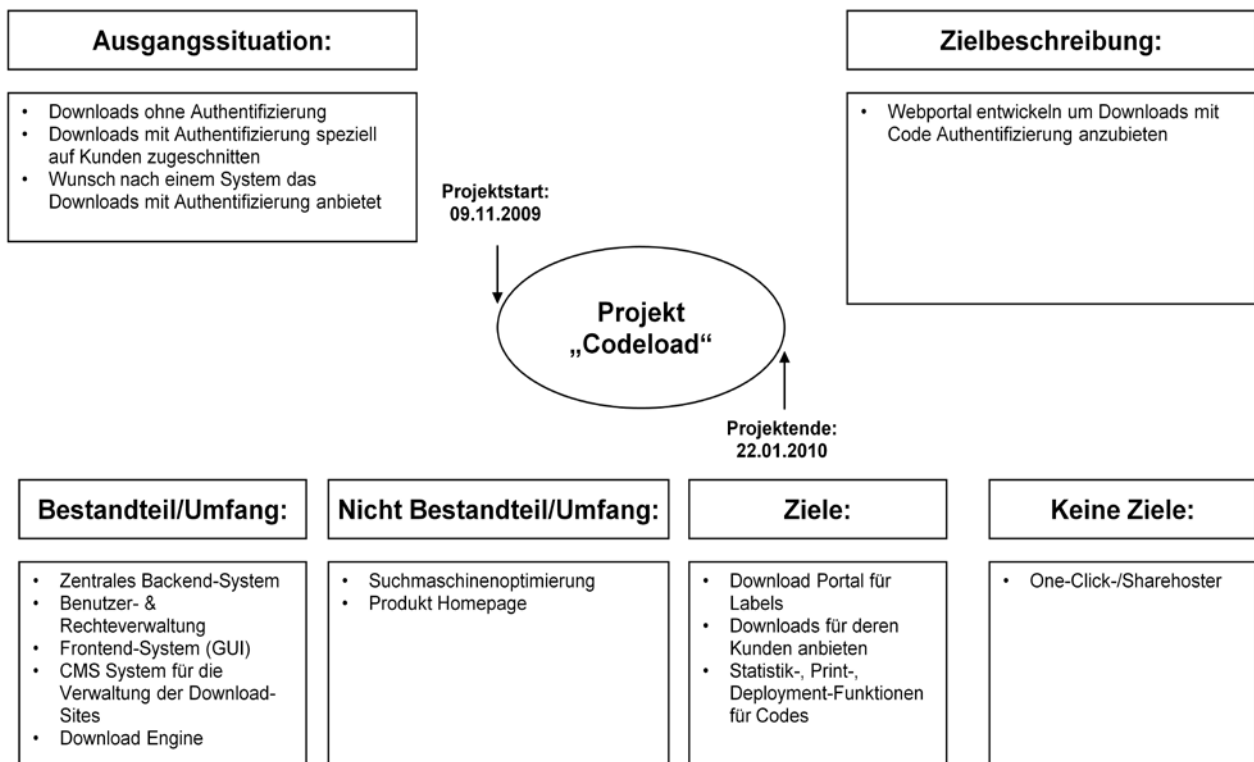
3.1 Hintergrund des Projektes

Viele Musiklabels sind auf der Suche nach neuen Lösungen für den Download von Musikdateien. Einige Musiklabels bieten an, dass der Kunde beim Erwerb eines Tonträgers (CD, Schallplatte, usw.), den dieses Label vertreibt, zusätzlich einen Code ausgehändigt bekommt, mit dem er oder sie weitere Lieder oder andere digitale Medien des Künstlers auf einer Website downloaden kann. Für diese Musik-/Plattenlabels soll ein "Download-Tool" entwickelt werden, das es ihnen ermöglicht, die Download-Codes über dieses Tool einfach und schnell generieren zu lassen. Zusätzlich soll es dem Label ermöglicht werden diese Codes über das Tool zu verwalten und auch auszudrucken. Darüberhinaus soll es auf dieser Web-Plattform auch möglich sein, einzelne Download-Sites für die Tonträger zu erstellen und zu verwalten. Ziel ist es ein Online-Portal für Code-Generierung und Musikdownloads zu entwickeln und bereitzustellen, das sich durch leichte Bedienbar- und hohe Benutzerfreundlichkeit auszeichnet.

3.2 Projektdefinition

3.2.1 Scope

Project Scope Definition



3.2.2 Projektziele

Das Projektziel ist die Entwicklung eines Online-Download- und Code-Generierungsportals für Musikdateien. Da es sich bei dem Produkt um eine Webanwendung handelt, muss zudem ein hohes Maß an Sicherheit dieser Anwendung gewährleistet sein. Die Kunden eines Musiklabels sollen mit Hilfe dieser Webanwendung in der Lage sein nach Eingabe eines Codes schnell und unkompliziert Musikdateien oder andere Digitalen Formate von einem Server herunterladen zu können.

3.2.3 Umfang und Abgrenzung

Umfang des Projektes ist es ein Webportal für Musik- und Plattenlabels zu entwickeln mit dem Sie schnell und unkompliziert Download-Codes und Download-Sites erstellen können.

Folgender Umfang ist für das Projekt vorgesehen:

- **Entwickeln eines zentralen Backend-Systems**
Das Backend-System enthält neben dem Webserver auf dem PHP Scripte ausführbar sein müssen, eine MySQL Datenbank.
- **Benutzer- und Rechteverwaltung**
Für das Webportal soll eine Benutzer- und Rechteverwaltung implementiert werden um Zugriffsrechte auf Daten klassifizieren und einschränken zu können. Des Weiteren soll ein Administrator implementiert werden um global Zugriff auf alle Daten des Systems zu erhalten.
- **Entwicklung eines Frontend-Systems**
Das Frontend-System dient der interaktiven Bedienung der Funktionen des Systems und steht in Verbindung mit dem Backend-System.
- **CMS¹¹ System zur Erstellung von Download-Sites**
Für die Erstellung und Verwaltung (z.B. ändern und löschen) der Download-Sites muss ein Content Management System implementiert werden um den Funktionsumfang zu garantieren.
- **Codegenerierung**
Für das System muss ein Codegenerierungs-Algorithmus implementiert werden der skalierbar Codes nach Kundenwünschen erstellt (Groß-/Kleinschreibung)
- **Statistiken**
Um den Status aller Downloads abfragen zu können, muss eine Statistikfunktion implementiert werden.
- **Veröffentlichung**
Um die Codes veröffentlichen zu können bedarf es gewisser Funktionen die dies sicherstellen (Druck, digitaler Versand).
- **Download Engine**
Für die Downloads und deren Verifizierung muss eine Logik implementiert werden. Diese muss auch ohne das vorhandene Portal funktionsfähig sein und mit der Datenbank interagieren.

¹¹ Content Management System: Dient zur Verwaltung von Webseiten und deren Inhalt

Dieses Projekt grenzt folgende Themen aus:

- Suchmaschinenoptimierung
- Webseite um den Dienst im Internet zu bewerben und zu Präsentieren (keine Akquise Tätigkeiten)
- Entwickeln eines Tarif-Modells für die Umlegung der Kosten auf die Kunden

3.2.4 Einschränkungen

Um das Online-Download-Portal für Musikdateien nutzen zu können, wird ein Computer mit internetfähigem Browser benötigt. Die Nutzung des Download-Portals mit mobilen Endgeräten ist bisher nicht vorgesehen kann aber in einem Folgeprojekt realisiert werden.

3.2.5 Rahmenbedingungen

3.2.5.1 Organisation

Das Projekt muss in einem zeitlichen beschränkten Rahmen durchgeführt werden. Für die Abgabe und Präsentation des Projekts ist der Termin der Media Night an der Hochschule der Medien in Stuttgart am 28.1.2010 vorgesehen. Um das Projekt erfolgreich abzuschließen muss dieser Termin eingehalten werden.

3.2.5.2 Technik

- Hardware:
Die Entwicklung des Projekts wird allein mit privater Hardware bewerkstelligt.
- WebServer:
Um das Portal betreiben zu können wird ein WebServer mit PHP 5 Interpreter und MySQL Datenbank benötigt. Hierfür wurde extra eine Domain (www.codeload.de) gemietet um die abschließenden Tests unter realen Bedingungen durchführen zu können und das Portal später betreiben zu können.
- Software
Für die Entwicklung des Webportals und der Gestaltung des Designs wurde ausschließlich Open Source Software eingesetzt (u.a. Ubuntu 9.10, NetBeans 6.8, GIMP 2.6)
- W3C
Während der Entwicklung soll vor allem auch darauf geachtet werden das Portal W3C konform zu entwickeln und vorgeschriebene Standards einzuhalten (XHTML usw.)

3.2.6 Projektorganisation

Für die Kontrolle der Meilensteine und um die erfolgreiche Umsetzung des Projekts zu gewährleisten, wurden in regelmäßigen Abständen Statusmeetings zwischen dem Projektmanager Tomislav Bisic und dem Projektteam bestehend aus Heiko Stumpf durchgeführt.

Rolle:	Name	E-Mail Adresse
Projektbetreuer	Prof. Dr. Fridtjof Toenniessen	toenniessen@hdm-stuttgart.de
Projektmanager	Tomislav Bisic	tb051@hdm-stuttgart.de
Projektteam	Heiko Stumpf	hs046@hdm-stuttgart.de

3.2.7 Schnittstellen zu anderen Projekten

Dieses Projekt wird als Standalone Projekt betrieben und steht daher nicht in Verbindung zu anderen Projekten. Schnittstellen zu weiteren Systemen oder Anwendungen werden deshalb nicht benötigt und sind deshalb auch nicht in der Planung enthalten. Eine weitere Schnittstellenanbindung könnte jedoch in einem Folgeprojekt für Mobile-Endgeräte definiert und realisiert werden.

3.3 Business Case

Die Entwicklung von **CodeLoad** dient in erster Line der Entwicklung und Schulung des Entwicklerteams. Da ein großes Interesse an Webtechnologien wie PHP und JavaScript besteht, ist die Umsetzung dieses Projekts prädestiniert um einen tieferen Einblick in Design-, Architektur- und Performance Konzepte für dynamische Webanwendungen zu erhalten. Des Weiteren soll nicht außer Acht gelassen werden, dass neben der erfolgreichen Entwicklung eines Projekts, auch Erfahrungen im Bereich des Projektmanagements gesammelt werden sollen.

Jedes neue Projekt bringt Herausforderungen mit sich, die im Voraus meistens nicht bekannt sind und gelöst werden müssen. Daher sollte das Entwickler-Team nicht nur technische Aspekte und Hintergründe erlernen, sondern auch die Art und Weise, Probleme anzugehen und erfolgreich zu lösen.

Das Prinzip Downloads anzubieten ist nichts Neues. Der Markt ist mittlerweile überflutet von einer Vielzahl an Sharehostern¹² (rapidshare, megaupload usw.). Diese Sharehoster bieten dem User die Möglichkeit, Dateien hochzuladen und als Download anderen Internet-Usern zur Verfügung zu stellen. Es ist keine Seltenheit, dass sich diese Download Links nach kürzerer Zeit in unzähligen Foren befinden und jeder Internet-User diese Dateien herunterladen könnte. Hier findet keine Berechtigungsverifizierung statt und Urheberrechtsverletzungen sind nicht auszuschließen.

Die Verifikation eines Downloads über einen Code stellt ebenfalls keine neuartige Idee dar. Allerdings gibt es im Internet keinen Sharehoster, der jedem interessierten Kunden die Möglichkeit anbietet Downloads zur Verfügung zu stellen und diese anhand eines Codes freizuschalten. Diese Dienstleistungen müssen seit je her speziell für einen Kunden implementiert und in sein System (Webseite,

¹² Internetdienstleister die sich auf den Download größerer Dateien spezialisiert haben

Webshop o.ä.) implementiert werden. Es fehlt also ein Anbieter der diese Dienstleistung anhand einer Webplattform global anbietet.

CodeLoad bietet in erster Linie Musik- und Plattenlabels diesen Dienst an, um ihre Veröffentlichungen optimal vermarkten und vertreiben zu können. Theoretisch kann jeder Veröffentlichung zusätzliches digitales Material zum Download angeboten werden, welches über den beigelegten Code heruntergeladen werden kann. Für jede Veröffentlichung oder Kampagne des Musik-/Plattenlabels wird eine spezielle Download-Site erstellt, die auf das Design des Tonträgers angepasst werden kann.

Um die Schnittstelle zwischen Erstellung der Codes und der endgültigen Verteilung an die Kunden optimal zu unterstützen, wurde speziell für Druckereien ein XML Export entwickelt. Die Codes können nach dem Druck des Booklets¹³, in einem zweiten digitalen Druckverfahren auf das bereits gedruckte Booklet aufgedruckt werden (personalisierter Druck).

Durch den Schutz der Downloads über einen Code wird die unfreiwillige Verbreitung der Dateien verhindert, wodurch keine Urheberrechtsverletzung im Bereich der öffentlichen Zugänglichmachung entstehen. Das Musik- oder Plattenlabel, welches die Nutzungsrechte an der Verbreitung vom Urheber hält, kontrolliert auf diesem Wege die Verbreitung und die Bereitstellung an den urhebergeschützten Materialien jedes Künstlers.

Um die entstehenden Kosten zu decken, die durch anfallenden Traffic und monatliche Domain bzw. Webspaces entstehen, soll ein Tarifmodell entwickelt werden, welches erlaubt die Kosten auf die Kunden und deren Nutzerverhalten umzulegen. Das Tarifmodell ist jedoch nicht Teil dieses Projekts und wird erst im produktiven Betrieb des Projektes umgesetzt.

3.4 Kunden

Wie schon in der Einleitung beschrieben, ist dieses Projekt aus den Anfragen von Platten- und Musiklabels entstanden und wurde daher in dieser Version speziell auf deren Bedürfnisse zugeschnitten und anhand deren Interessen umgesetzt. Es ist aber durchaus für jeden benutz- und anwendbar der einer speziellen Gruppe von Usern Downloads anbieten möchte. Dies können sowohl Bands, die Ihre Tonträger ohne einen Plattenvertrag vertreiben möchten aber auch Unternehmen sein, die für Werbe- oder Marketing Kampagnen Downloads anbieten möchten.

3.4.1 Kundenerwartungen

Um den Kundenerwartungen gerecht zu werden wurden vor dem eigentlichen Entwicklungsstart von CodeLoad zuerst Informationen über gewünschte Funktionalitäten der potentiellen Kunden eingeholt. Einheitlich wurde auf eine einfache Erstellung der Codes und Downloads-Sites Wert gelegt. Weiterhin sollte die Funktion als Host¹⁴ übernommen werden und an eine zuverlässige und skalierbare Druckfunktion implementiert werden. Zu guter Letzt sollte eine konstante Verfügbarkeit des Services vorhanden sein.

¹³ Beiheft eines Tonträgers mit Informationen über den Künstler, Songtexte usw.

¹⁴ bereitstellen von Webspaces zur Speicherung von Dateien im Internet

3.5 Abnahmekriterien

Folgende Punkte wurden als Abnahmekriterien definiert:

- Projekt muss im zeitlichen Rahmen abgeschlossen werden
- Webanwendung muss auf Fehler getestet werden
- Webanwendung muss die gängigen Sicherheitskriterien erfüllen, sprich gegen kriminelle Einwirkungen von außerhalb geschützt werden
- Gewährleistung der Datenintegrität angemeldeter User
- Einmaligkeit eines Codes muss garantiert werden
- Code darf nur einem einzigen Kunden zugeordnet werden dürfen
- Unterstützung folgender Webbrowser: Internet Explorer (ab Version 6), Mozilla Firefox (ab Version 2.0), Opera (ab Version 5.0)
- Gute Performance auch bei mehreren gleichzeitigen Zugriffen durch verschiedene User
- Zuverlässigkeit der Downloads
- Gewährleistung der Verfügbarkeit der Downloads
- Ansprechendes Erscheinungsbild (Design, GUI)
- Statistiken über Downloadaktivitäten
- Verteilung der Codes in physikalischer sowohl Digitaler Form (Druck bzw. XML, E-Mail)
- Benutzerverwaltung mit Rechtesystem soll implementiert werden

3.6 Kosten / Budget

Dieses Projekt wurde im Umfeld einer studienbegleitenden Projektarbeit umgesetzt. Alle anfallenden Arbeitsstunden wurden durch das Projektteam geleistet und stehen somit in keinem Zusammenhang für anfallende Kosten. Da komplett auf Open-Source Software entwickelt bzw. gestaltet wurde, fallen somit auch keine Lizenzgebühren an.

Die Entwicklung fand komplett auf eigener Hardware statt, wofür ebenfalls keine zusätzlichen Kosten entstanden sind. Lediglich für endgültigen Test der Produktivumgebung und das finale deployment des Systems fallen Kosten für einen Webserver inkl. Domain an. Dies wird vom Projektteam, aus eigenem Interesse an der Fortführung dieses Projektes, selbst übernommen.

Jeder gestartete Download erzeugt Traffic¹⁵, dies galt es bei er Auswahl des Tarifes zu beachten. Aufgrund des geringen Bekanntheitsgrades des Projektes und der gegenwärtigen Entwicklungsphase, kann das Traffic-Problem aktuell noch außer Acht gelassen werden. Sobald das System mehr als 10 Kunden besitzt sollte auf einen höheren Tarif ohne Traffic Limit gewechselt werden.

Von Seiten der Hochschule wurde für dieses Projekt auch kein Budget eingeplant bzw. vergeben.

3.7 Projekt toleranz

Geplant ist es, das Projekt nach seinen definierten Zielen zu implementieren und abzuschließen. Sollten während der Entwicklungsphase Schwierigkeiten bzw. zeitlich kritische Probleme auftreten, können diese, falls das Projektziel dadurch nicht gefährdet ist, abgeändert werden. Hierfür bedarf es

¹⁵ Datenaufkommen in Computernetzen

allerdings der Rücksprache mit dem Projektleiter. Alle Änderungen werden im Change Request Protokoll festgehalten und finden sich in Abschnitt 6.3 Change Management wieder.

3.8 Risiken

Um im Voraus Gefahren bei der Entwicklung bzw. Umsetzung des Projekts entgegen zu wirken wurden zu Beginn mögliche Risiken definiert und hierfür Lösungen vorgeschlagen.

Eine detaillierte Auflistung aller Risiken finden Sie in Abschnitt 7 Risikomanagement wieder.

3.9 Entscheidungsvorschlag

Vor Projektstart wurden die Risiken des Projektes genau analysiert und Gegenmaßnahmen definiert mit denen diese Risiken minimiert oder sogar ausgeschlossen werden können.

Folgende Punkte sollen im Gegensatz zu den Risiken stehen und den Entscheidungsvorschlag für die Projektrealisierung erleichtern

- Konstante Kontrolle des Arbeitsfortschritts
- Kritische Programmiererelemente werden Priorisiert bearbeitet
- Professoren und Wissenschaftliche Mitarbeiter sollen bei der Beantwortung offener Fragen zu Technologien und deren Umsetzungen behilflich sein
- Planung und Durchführung des Projekts durch einen Projektleiter
- Kein wirtschaftlicher Schaden bei nicht Erfolg des Projekts

Aufgrund der entstehenden Zusammenarbeit mit dem Projektleiter Tomislav Bisic und der Unterstützung durch den Projektbetreuer Prof. Dr. Fridtjof Toenniessen fiel die Entscheidung auf die Umsetzung dieses Projektes.

4 Projektlösungsansatz

4.1 Erforderliche Qualifikationen

Für **CodeLoad** kommen folgende Script- und Beschreibungssprachen zum Einsatz:

- XHTML
- PHP
- JavaScript
- SQL
- CSS

Kenntnisse in den Scriptsprachen PHP und JavaScript sind zu Beginn des Projekts nur rudimentär vorhanden. Beide Script Sprachen sollten aber keinerlei Hürden darstellen, da sie bezüglich Syntax und Aufbau stark an C++ bzw. Java ähneln, in welchen das Projektteam mehrere Jahre Erfahrung aufweisen kann. In den restlichen erwähnten Technologien bestehen fundierte Kenntnisse.

4.2 Alternativen

Das System hätte serverseitig ebenfalls mit Java Server Pages implementiert werden können. Vorteile wären hierbei die Debugging Möglichkeiten und die bereits vorhanden Kenntnisse des Projektteams in der Programmierung mit Java.

PHP hat sich aber als Standard für dynamische Webseiten durchgesetzt und soll deshalb auch als gesetzte Scriptsprache für dieses Projekt betrachtet werden. In erster Linie geht es bei diesem Projekt auch darum, einen gewissen Lerneffekt zu erzielen und eine neue Scriptsprache zu lernen.

4.3 Infrastruktur

Für das Projekt sind gewissen Voraussetzungen zu erfüllen um das Produkt zu entwickeln, zu testen und im späteren Zeitpunkt in Betrieb zu nehmen.

4.3.1 Entwicklungsinfrastruktur

Um auch während der Entwicklungsphase Ressourcen, Kosten und Zeit zu sparen, wird hierfür eine spezielle Entwicklungsumgebung benötigt. Diese Entwicklungsumgebung basiert auf einer virtuellen Ubuntu 9.10 Umgebung, in der ein Apache Webserver, ein PHP Interpreter und eine MySQL Datenbank installiert wurden. Dieser Mehraufwand für die Einrichtung der virtuellen Maschine rechnet sich während des eigentlichen Entwicklungsprozesses wieder, da zum einen auch offline an dem Projekt gearbeitet werden kann und der permanente Dateitransfer auf den Server zum Testen der Skripte hinfällig wird.

Zusätzlich soll in der virtuellen Linux Umgebung die Versionierungssoftware Subversion installiert werden, um den Überblick über alle Dateiveränderungen zu behalten. Dies ist durchaus sinnvoll, da

es vorkommen kann, dass das Projekt aufgrund anderer Studententätigkeiten still steht und zu einem späteren Zeitpunkt vergangene Arbeiten leichter aufgenommen werden können.

4.3.2 Produktivinfrastruktur

Für die Produktivumgebung soll eine passende Domain und zugehöriger Webespace bei einem Provider gemietet werden, um das Projekt am Ende der Durchführungsphase in einem realen Umfeld testen und schließlich an der Media Night präsentieren zu können.

Sowohl die Produktiv- als auch die Entwicklungsinfrastruktur müssen über identische Konfigurationen und Versionsnummern der einzelnen verwendeten Scriptsprachen (z.B. PHP 5.2.9) verfügen um später Portierungen des Codes aus dem Wege zu gehen.

4.4 Eigenentwicklung und fertige Lösungen

Im Bereich der dynamischen Webentwicklung gibt es mittlerweile eine Vielzahl von Frameworks (Zend Framework, Symfony) und Content-Management-Systeme (Typo3, Joomla) auf dem Markt, die zum größten Teil auch kostenlos zur Verfügung stehen und nach eigenen Bedürfnissen angepasst werden können. Es bedarf aber einer gewissen Einarbeitungszeit in die Architektur und den Code, um das Verständnis für das entsprechende Framework zu erlangen. Gleiches gilt hier für Content-Management-Systeme. Zu Beginn des Projekts muss also die Frage gestellt werden, ob auf ein bereits vorhandenes Framework oder Content-Management-System zurückgegriffen werden oder ob eine Eigenentwicklung implementiert werden soll.

4.4.1 Frameworks

Ein Framework für eine dynamische Webanwendung hat mehrere unterschiedliche Funktionen. Es enthält eine Sammlung von Klassen die z.B. für die Benutzeranmeldung und Umsetzung von Benutzerrechten eingesetzt werden können. Es bedarf dabei einer gewissen Zeit um mit dem Framework umgehen zu können und seine Eigenschaften effizient auszunutzen.

4.4.2 Content- Management-Systeme

Content-Management-Systeme sind Anwendungen, die für die Erstellung und Verwaltung von Websites verwendet werden. Mit ihnen ist es möglich, ohne Kenntnisse im Umgang mit HTML Seiten, diese zu erstellen und zu verwalten bzw. ihren Inhalt zu kontrollieren und gegebenenfalls zu verändern. Sie verhalten sich ähnlich wie Frameworks, nur lässt sich das System nicht mehr modifizieren.

4.4.3 Architektur Entscheidung

Für dieses Projekt wurde bewusst auf den Einsatz eines Frameworks oder eines Content-Management-Systems verzichtet. Dies hat in erster Linie nichts mit der Einarbeitungszeit in das entsprechende Framework zu tun, sondern viel mehr mit dem Lernfaktor, der entsteht wenn ein solch „kleines“ Framework selbst entwickelt wird. Diese Entscheidung wurde mit der Kenntnis gefällt, dass durchaus ein erheblicher Mehraufwand auf das Projektteam zukommt und das erstellte Framework eventl. nicht an die Performance eines fertigen, ausgereiften Frameworks herankommt.

4.4.4 Entwicklungsprinzipien

Objektorientiertes Programmieren ist seit Jahren das Werkzeug der Wahl um skalierbare und wiederverwendbare Applikationen zu entwickeln. PHP bietet seit der Version 4 und mittlerweile in der Version 5 weitreichende Unterstützung für Objektorientiertes Programmieren. **CodeLoad** soll komplett über den Objektorientierten Ansatz entwickelt und modular aufgebaut werden.

Dies ermöglicht dem Entwickler, geschriebene Programmcodes komfortabel über das komplette System hinweg zu verwenden.

4.5 Security und Vertraulichkeit

- **Authentifizierung:**
Die User des Portals müssen sich mit einem Benutzernamen und Passwort am System anmelden
- **Sicherheitsbestimmungen für Webanwendungen:**
Alle Sicherheitsbestimmungen für Webanwendungen (Schutz vor Cross-Site Scripting, SQL-Injections usw.) müssen eingehalten werden, da ansonsten Daten unbefugt heruntergeladen oder gelöscht werden können und wirtschaftlichen Schaden verursachen. Im schlimmsten Fall wird die Datenbank durch SQL Injections gelöscht, was einen kompletten Ausfall des Systems zur Folge hätte.
- **Datenbank Backup:**
Sollte die Datenbank durch den Entwickler gelöscht werden oder Inkonsistenzen aufweisen, muss ein tägliches inkrementelles Backup der Datenbank angefertigt.

4.6 Wartungs- und Betriebsanforderungen

4.6.1 Betriebsanforderungen

- Für das System ist es wichtig, dass der Server eine geringe Downtime hat. Das heißt der Server sollte nur zu Wartungszwecken neu gestartet werden. Der komplette Datenverkehr zur Erstellung der Download-Sites, also das WebPortal an sich und die Download-Sites die für den Kunden sichtbar sind, laufen über diesen Webserver. Es muss gewährleistet werden, dass der Kunde praktisch zu jeder Tages- und Nachtzeit einen Download starten kann. Eine größere Ausfallzeit des Dienstes wirkt sich schlecht auf das Image des Künstlers bzw. des Musik-/Plattenlabels aus. Des Weiteren legt es ein schlechtes Bild auf **CodeLoad** und schreckt neue Kunden ab.
- Um Downloads auch ohne das Portal anbieten zu können, müssen beide Systeme, sprich das Portal zur Generierung der Codes und die einzelnen Download-Sites, unabhängig voneinander funktionieren und als Standalone Anwendungen implementiert werden. Soll das Portal zu Wartungszwecken kurz geschlossen werden, ist so der Download Betrieb ohne weiteres möglich.
- Um schnell mit Daten wie z.B. neu erstellte Codes oder Benutzerinformationen umgehen zu können, wird als Datenbankverwaltungssystem eine MySQL Datenbank verwendet.
- Um die Datenbank zu entlasten, muss die grundlegende Architektur intelligent genug skaliert werden um ständige Datenbankzugriffe wie z.B. auf Benutzer & Gruppen Informatio-

nen zu speichern. Dies wird über Sessionvariablen geregelt, die solange der Benutzer angemeldet ist die nötigen Informationen speichern.

4.6.2 Wartungsanforderungen

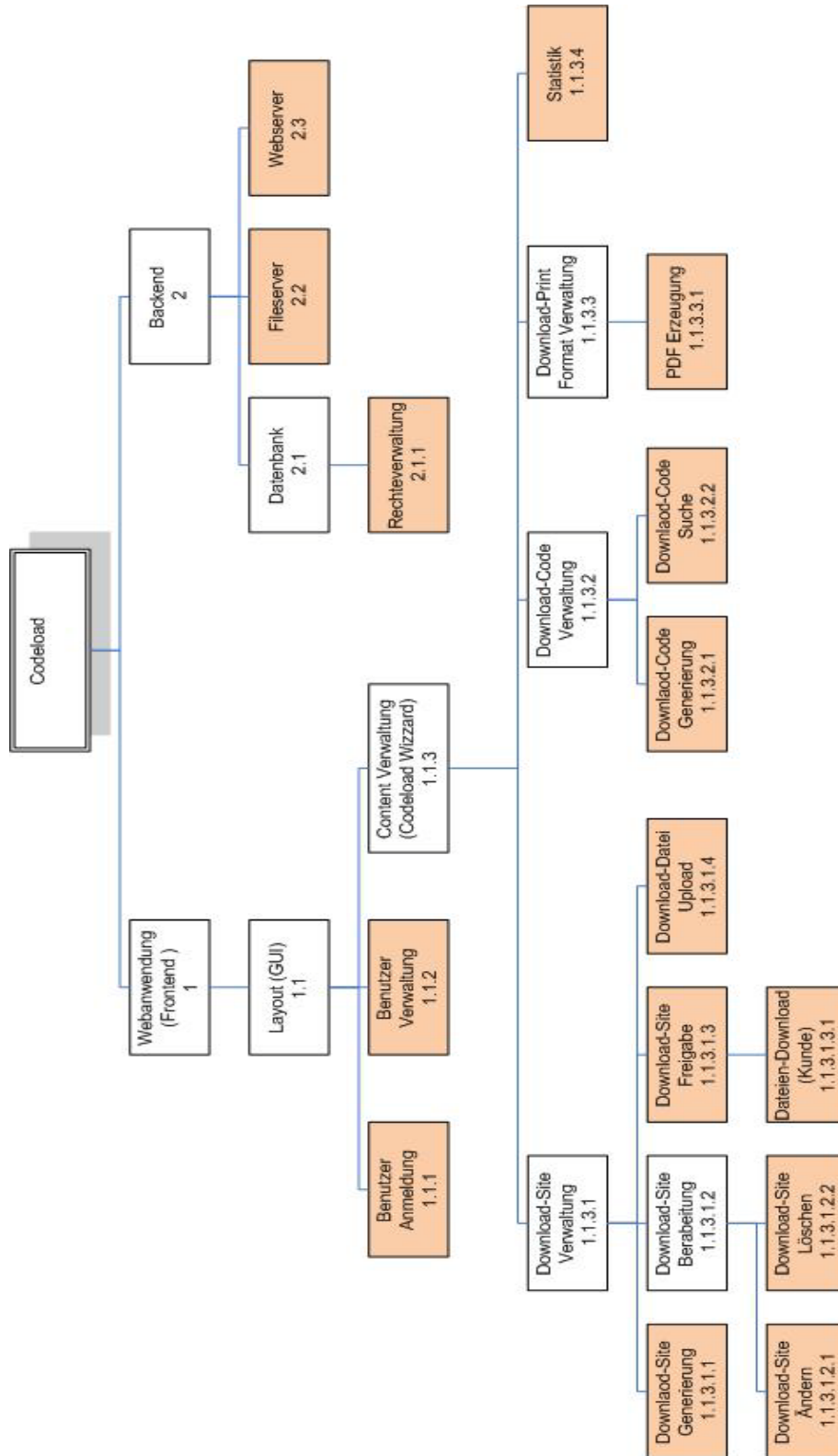
Falls der Server einmal ausfallen sollte, wäre es wünschenswert den Dienst auf einem Ersatzserver weiter zur Verfügung zu stellen. Dies setzt aber einen Abgleich mit der Datenbank voraus, was zusätzlich einen erhöhten Entwicklungs- und Kostenaufwand zur Folge hätte und für dieses Projekt nicht umgesetzt werden kann.

4.7 Vereinbarungen

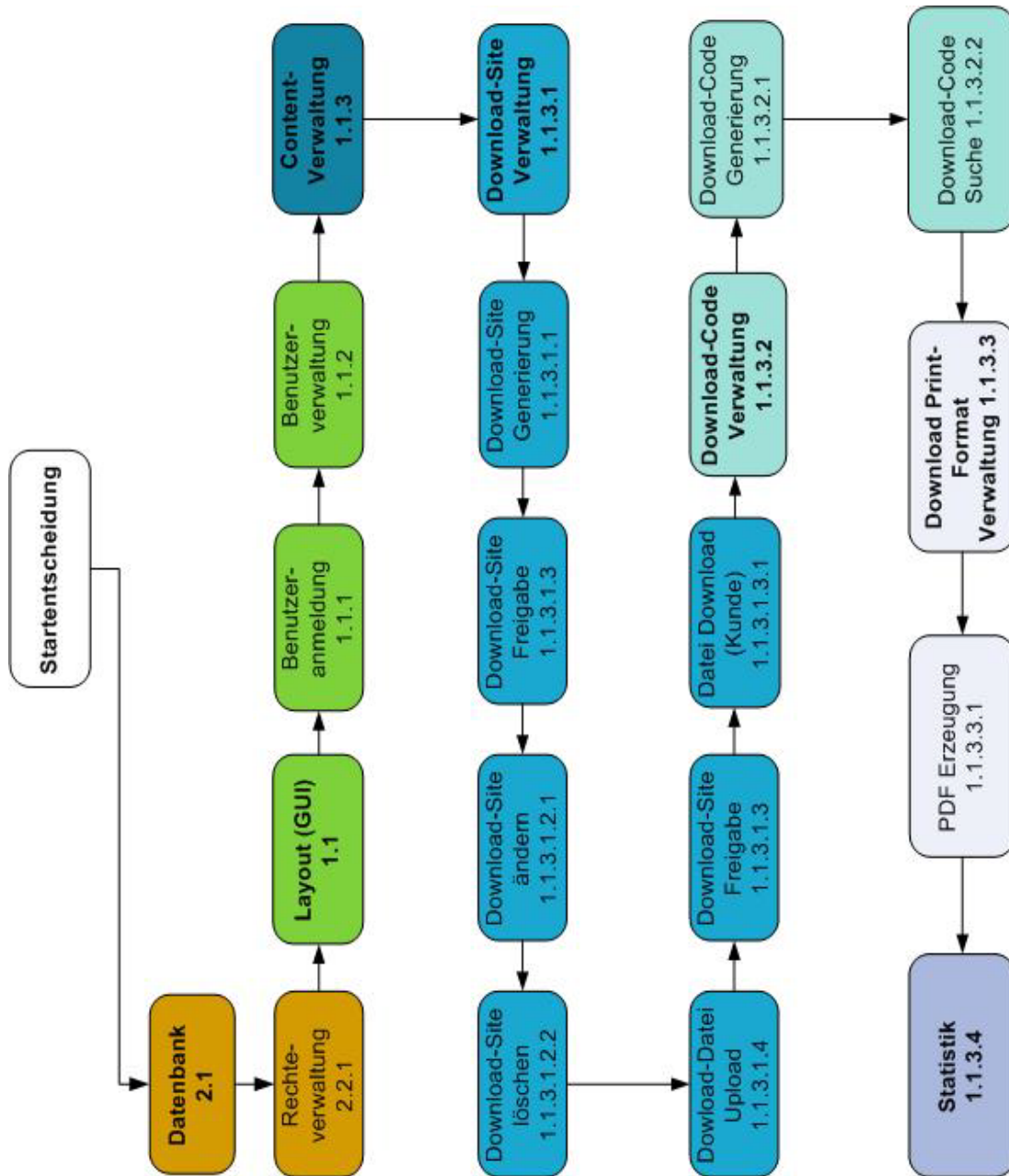
CodeLoad basiert auf einer eigenständigen Idee des Entwicklers und soll zu einem späteren Zeitpunkt als Dienst im Internet angeboten werden. Es besteht zwar von außerhalb Interesse an dieser Lösung, jedoch wurden keine mündlichen oder schriftlichen Vereinbarungen mit zukünftigen Kunden oder Firmen abgeschlossen.

5 Produktbeschreibung

5.1 Produktstrukturplan



5.2 Produktflussdiagramm



5.3 Managementprodukte

Die Managementprodukte dieser Dokumentation gliedern sich folgendermaßen auf:

- Einführung
- Projektbeschreibung
- Projektlösungsansatz
- Produktbeschreibung
- Projektplan
- Projektflussdiagramm
- Risikomanagement
- Fazit

Anhand dieser, im oberen Abschnitt der Dokumentation erwähnten Punkte wird die Organisation, Planung und Theorie des vorliegenden Projekts beschrieben. Diese Punkte beschreiben vor Projektstart die Realisierbarkeit des Projektes, dienen während der Durchführung als Leitfaden und nach Abschluss des Projekts als Tätigkeitsnachweiß der Projektinhalte. Detaillierte Informationen über die genannten Punkte befinden sich in ihren jeweiligen Abschnitten.

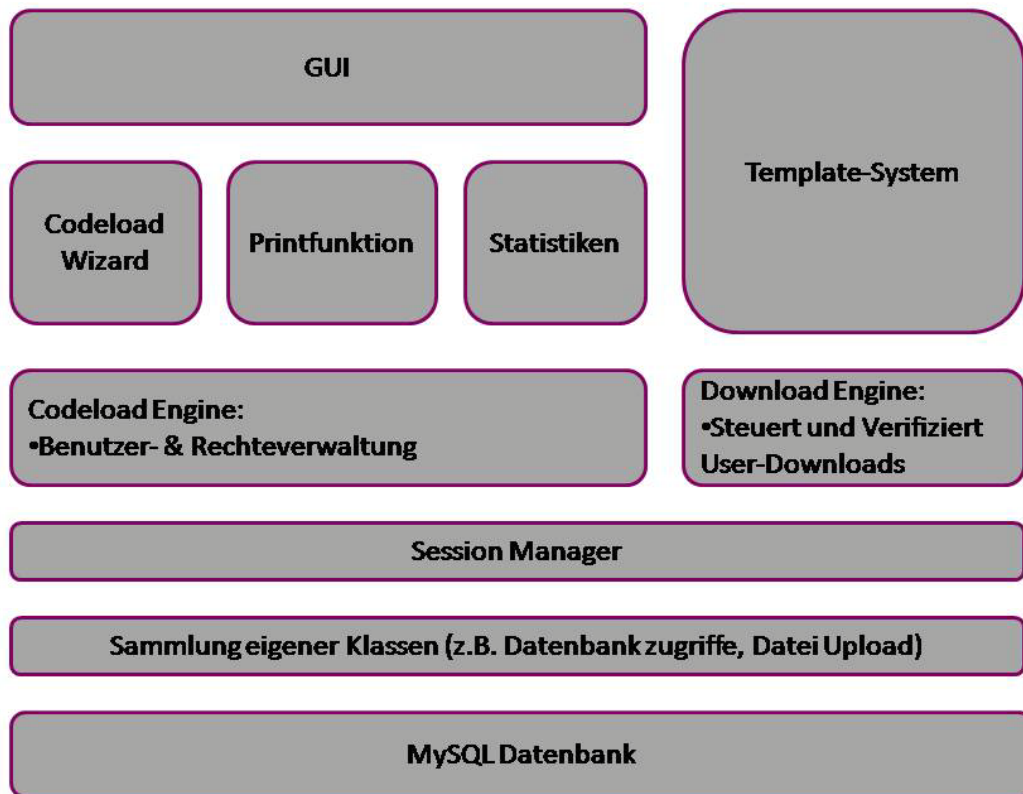
5.4 Spezialistenprodukte

Im folgenden Abschnitt soll auf die wichtigsten Produkte dieses Projekts eingegangen werden. Nach PRINCE2¹⁶ beschreiben diese Produkte das finale Produkt nach Projektabschluss.

5.5 Architektur/Framework

Im Projektplan wurde schon detailliert über die Entscheidung ein Framework einzusetzen oder nicht, ausführlich beschrieben. Um den Lerneffekt für dieses Projekt zu maximieren wurde ein eigenes „kleines“ Framework implementiert. Die nachfolgende Grafik soll die Architektur des Frameworks und des Projektes verdeutlichen. Auf die einzelnen Komponenten wird im weiteren Verlauf der Produktbeschreibung eingegangen.

¹⁶ Prince2: „Projects in Controlled Environments“ – Projektmanagement Methode



5.5.1 Codeload WebPortal

Die zentrale Anlaufstelle für die Erstellung und Verwaltung der Download-Sites sowie deren zugehörigen Codes erfolgt über das **CodeLoad** WebPortal. Über dieses Portal ist es außerdem noch möglich, weitere Zusatzfunktionalitäten wie Statistiken oder Druckfunktionen abzurufen.

5.5.1.1 Sessions

Für **CodeLoad** ist es notwendig, den User, sei es der User des Systems oder der User der per Code Downloads startet, zu erkennen und seine „Identität“ zu speichern. Im System muss sich der User durch die Benutzerverwaltung identifizieren und seine Identität wird während seiner Sitzung konstant gespeichert. Seine Identität und Gruppenzugehörigkeit ist wichtig um die Gruppenrichtlinien zu gewährleisten. Für den Download User wird ebenfalls eine Sitzung angelegt, um ihn bei einem eventl. Abbruch des Downloads nochmals zu erkennen und den Download erneut zu starten, obwohl der Code schon eingelöst wurde. Genauere Informationen zu den zwei Sitzungsvarianten sind im Abschnitt Benutzerverwaltung bzw. Download Engine zu entnehmen.

5.5.1.2 Benutzer- & Rechteverwaltung

Für das **CodeLoad** System ist es notwendig, eine Benutzer- & Rechteverwaltung zu implementieren. Die Verwaltung der Codes und Downloadseiten darf nur für die entsprechende Gruppe (Label/Plattenfirma) bzw. den Benutzer der zu dieser Gruppe gehört möglich sein.

Durch den Benutzer wird über seine Email Adresse und sein Passwort der Login sichergestellt, durch die Gruppe werden alle Inhalte im System anhand der Gruppen ID dargestellt. So wird sichergestellt, dass das Erstellen und Ändern von Seiten und Codes nur im Gruppenkontext stattfindet.

5.5.1.3 Klasse Session Manager

Die Motivation hinter der CSessionManager Klasse ist, nach der Instanziierung ein Objekt zu besitzen das die komplette Benutzer, Gruppen und Rechte Verwaltung übernimmt:

- die Erzeugung einer eindeutigen SessionID
- speichern der Session in der Datenbank
- die Speicherung aller anfallenden Variablen für eine Seite um diese auch für nachfolgende Seiten zur Verfügung zu stellen

Problem

Der verbindungslose Aufbau von HTTP hat bei größeren Projekten bzw. zusammenhängenden Webseiten den Nachteil, dass Informationen für nachfolgende Seiten verloren gehen. Der Server vergisst sämtliche Informationen, die er aus der letzten Transaktion ermittelt hat. Ob ein Nutzer die Seite ein zweites Mal aufruft oder ob es sich um zwei verschiedene Nutzer handelt, kann nicht ohne weiteres unterschieden werden.

PHP bietet für dieses Probleme die Möglichkeit der Sitzungsverfolgung an. Dabei bekommt der Nutzer der Webseite eine eindeutige ID zugeteilt, die Session ID. Dies kann ohne eigene Implementierungen „Out of the Box“ mit PHP auf 3 verschiedene Wege erfolgen.

- Setzen von Cookies
- Übermittlung der Daten durch den Query-/ Parameterstring, Per HTTP GET (<http://www.codeload.de/index.php?var=wert&var2=zweiterwert>)
- Übermittlung der Daten durch den HTTP Header per HTTP POST

Alle 3 Methoden haben eines gemeinsam, dass sie nur bedingt sicher sind. Der Query-/ Parameterstring kann durch den User in der Adressleiste verändert und der HTTP Header durch „dritte“ abgefangen und somit die Session übernommen werden. Cookies können vom Benutzer deaktiviert werden und stehen somit der Sessionverwaltung nicht mehr zur Verfügung. Um diese Probleme zu umgehen wurde die Sessionverwaltung für **CodeLoad** in die MySQL Datenbank verlagert.

Umsetzung

Die Sessionverwaltung wird durch die abstrakte Klasse CSessionManager komplett in eine MySQL Datenbank verlagert. PHP bietet dafür die Möglichkeit dem bereits vorhanden Session Handler neue Callback Funktionen zu übergeben. Diese Callback Funktionen werden durch den Webserver bei der Erstellung und Beendigung der Session oder nach Abarbeitung des Scripts aufgerufen.

Da es für **CodeLoad** zweimal nötig ist Sessions zu speichern, sowohl für das Webportal als auch die Download Engine wurde die Klasse CSessionManager als abstrakt deklariert. D.h alle weiteren Klas-

sen die mit Sessions arbeiten müssen, können von dieser Klasse ableiten und ihre eigenen Implementierungen vornehmen.

Das Klassendiagramm für die Printfunktion befindet sich in Abschnitt 5.5.6 Klassendiagramme.

5.5.1.4 Content-Verwaltung

Die Content-Verwaltung umfasst folgende Punkte, welche ebenfalls aus dem Framework Architektur Diagramm zu entnehmen sind:

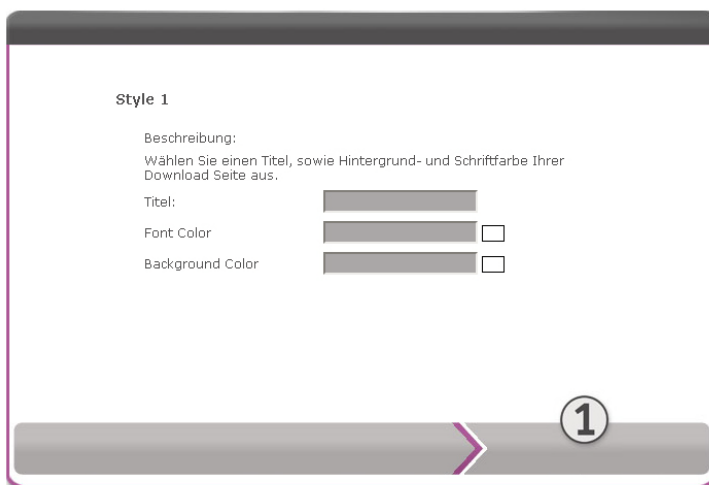
- **CodeLoad Wizard**¹⁷
Auf den Wizard wird in Punkt 4.5.1.4 noch genauer eingegangen
- **Printfunktion**
Um PDF Dokumente für den Druck der Codes zu erstellen wurde eine Wrapper Klasse für die frei verfügbare FPDF Library geschrieben. Für Informationen über die FPDF Library wird auf die Homepage der Bibliothek im Anhang verwiesen.
Das Klassendiagramm für die Printfunktion befindet sich in Abschnitt 5.5.6 Klassendiagramme.
- **Statistiken**
Anhand von Datenbankabfragen, in Zeiträume gegliedert und nach Download-Sites sortiert, lassen sich verschiedene Statistiken anzeigen.
Das Klassendiagramm für die Statistik befindet sich in Abschnitt 5.5.6 Klassendiagramme.

5.5.1.5 CodeLoad Wizard

Grundidee

Ein wesentlicher Bestandteil von **CodeLoad** ist das Erstellen unterschiedlicher Webobjekte, sprich Codes, Download Sites oder PDF Dateien für den Druck. Diese Objekte haben jeweils bestimmte Ei-

genschaften. Bei der Erstellung der Codes sind dies z.B. die maximale Anzahl, die Länge oder die Syntax (Groß- und Kleinbuchstaben) der Codes. Für die Download-Sites lässt sich die Hintergrund- und Schriftfarbe anpassen, Bilder- und die Download Dateien hochladen. Um den Benutzer durch diesen Erstellungsprozess zu führen, also die Definition dieser Eigenschaften vorzunehmen wird dies anhand eines Assistenten durchgeführt – der **CodeLoad Wizard**.



CodeLoad Wizard für Download-Sites

¹⁷ Assistent der durch einen Erstellungsprozess führt

5.5.1.6 Klasse Wizard

Beim **CodeLoad**-Wizard handelt es sich um eine abstrakte Basisklasse, die die allgemeinen Funktionalitäten eines Assistenten kapselt. In diesem Fall implementiert die Klasse die Logik für das Vor- und Zurückspringen der einzelnen Schritte und die Speicherung der einzelnen Eingaben. Sollte also bei der Ausführung des Assistenten eine Eingabe vergessen werden, so lässt sich zu diesem Punkt navigieren ohne die restlichen Eingaben zu verlieren, was bei Web Anwendungen durchaus nicht üblich ist. Ist der Wizard abgeschlossen, bekommt der User vor der endgültigen Speicherung seiner Eingabe Daten ein Übersichtsfenster präsentiert. Eine weitere Eigenschaft ist, dass der Wizard komplett als Formularfeld aufgebaut ist. Er sendet also durch den Next Button die Eingaben an sich selbst, die Logik des Wizards regelt dabei das Speichern und lädt die nächste darzustellende Seite.

Wie schon in der allgemeinen Beschreibung des Wizards erwähnt, wird der Wizard für jeden allgemeinen Erstellungs-, Änderungs- und Lösprozess angewendet.

Um dem Wizard nun eine spezielle Funktion zu übertragen, z.B. die Erstellung einer Download-Site, wird von ihm abgeleitet und nun die speziellen Eigenschaften die für eine Download-Site nötig sind, implementiert. Es muss in diesem Fall nur die Methode für das Speichern der Daten angepasst werden. Da der Download-Site-Wizard dies anders Hand habt als z.B. die Modifizierung einer bereits vorhandenen Download-Site (SQL Befehle: INSERT INTO, UPDATE).

Wie stellt der Wizard nun die einzelnen Seiten dar? Der Wizard ist mehr als Darstellungsrahmen zu verstehen, die einzelnen Seiten sind eigenständige HTML Seiten welche nur in den Wizard eingehängt werden und durch bestimmte positionierungseigenschaften der Reihe nach aufgerufen werden.

Das Klassendiagramm für den Wizard befindet sich in Abschnitt 4.5.4

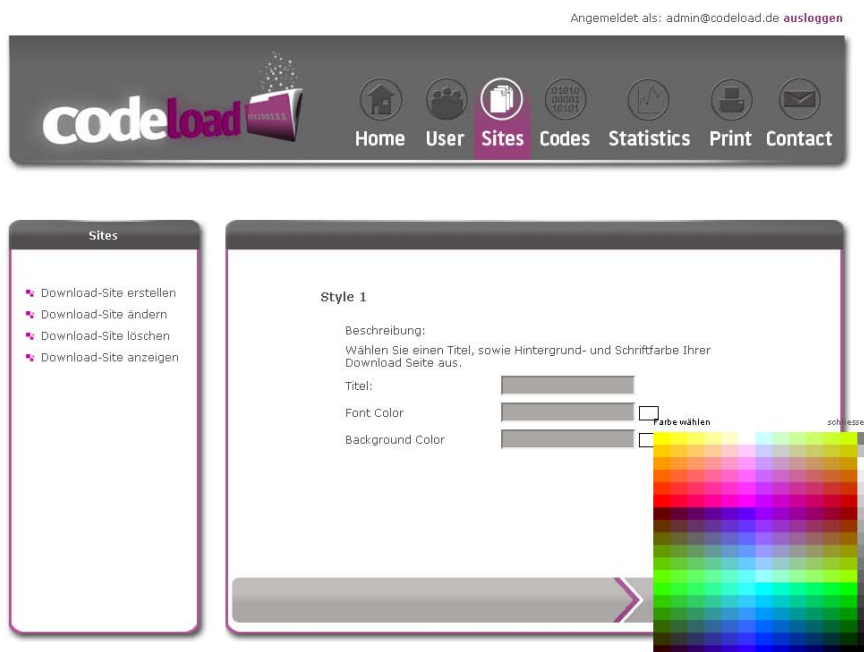
5.5.1.7 GUI / Benutzeroberfläche / Navigation

Für das WebPortal eine Primär/Sekundärnavigation implementiert. Jeder Primärmenüpunkt enthält weitere Untermenüpunkte die in der Sekundärnavigation dargestellt werden. Dies ist wie folgt aufgeteilt:

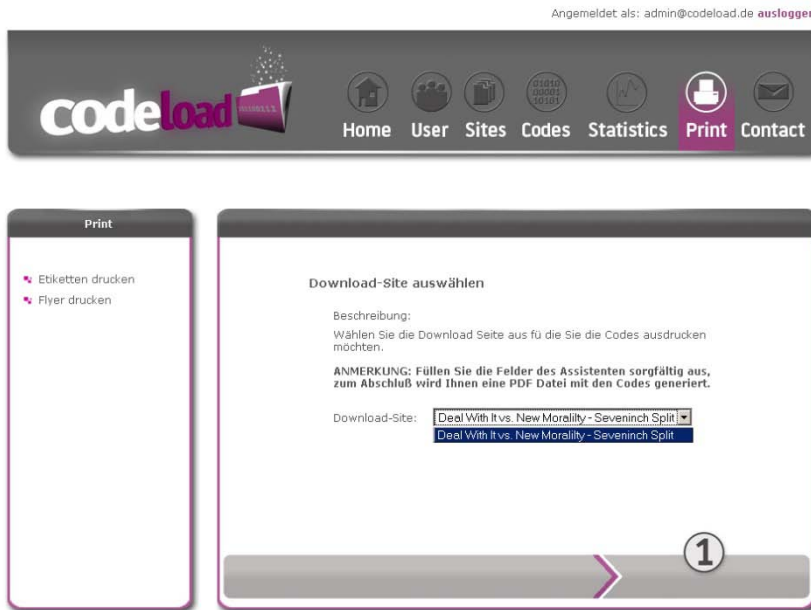
- Home
- User
 - Benutzername ändern
 - Passwort ändern
- Sites
 - Download-Site erstellen
 - Download-Site ändern
 - Download-Site löschen
 - Download-Site anzeigen

- Codes
 - Code suchen
 - Code erstellen
- Statistics
 - Statistik anzeigen
- Print
 - Etiketten drucken
 - Flyer drucken
 - XML Export
- Contact

Um einen Einblick in das Web-Portal und die Menünavigation zu bekommen wurden zwei Screenshots der Dokumentation beigefügt.



Rubrik Sites



Rubrik Print

5.5.2 Codes

Eine der Hauptfunktionen von **CodeLoad** ist es, Codes zu generieren und über diese Downloads zur Verfügung zu stellen. Dabei muss das System in erster Linie die Codes erstellen, in die Datenbank eintragen und einer Download Seite zuordnen. Neben der Verwaltung im System müssen die Codes durch die Download Engine verifiziert und eingelöst werden. Weitere Funktionen wie Drucken von Codes werden dabei durch andere Klassen zur Verfügung gestellt.

Sollen Codes generiert werden, gibt es dafür mehrere Einstellungsmöglichkeit.

- Länge: Kann zwischen 5 und 10 Zeichen gewählt werden
- Zeichen: Zahlen und Großbuchstaben
- Zeichen: Zahlen und Groß-/Kleinbuchstaben (Case Sensitive)

Für jedes einzelne Zeichen eines Codes wird zufällig unterschieden ob eine Zahl oder ein Buchstabe gewählt werden soll. Daher kann es auch durchaus vorkommen das Codes, nur aus Zahlen oder nur aus Buchstaben bestehen.

5.5.2.1 Klasse Code

Grundidee:

Um die Funktionalität der Code Erstellung und Verwaltung zu kapseln und auch anderen Stellen des Systems zur Verfügung zu stellen, wurde der komplette Funktionsumfang der Codes in die Klasse **CCode** implementiert. So benötigt z.B. die Funktion Drucken des Systems die Codes als Array oder der Wizard die Methoden zum erstellen von Codes.

Problem:

Für die Erstellung der Codes gibt es keinen fertigen Algorithmus der bereits in einer PHP Library implementiert ist. Der Benutzer des Systems soll die Codes für seine Ansprüche erstellen können und dafür Auswahlmöglichkeiten haben. So soll zum Beispiel die Länge der Codes (Anzahl Zeichen) freiwählbar sein, oder ob die Codes aus Zahlen und Groß- /Kleinbuchstaben bestehen oder nur aus Zahlen und Großbuchstaben.

Bei der Erstellung der Codes muss weiterhin beachtet werden, dass jeder erstellte Code eindeutig im System ist. Kein Kunde soll die Möglichkeit besitzen, seinen Code auch für eine andere Download Seiten zu verwenden. Die Chance, mit dem richtigen Code die richtige Download Seite zu wählen in der der Code ebenfalls vorhanden ist geht zwar gegen Null, trotzdem muss dies ausgeschlossen werden.

Umsetzung:

Die Codeverwaltung wird vollständig von der Klasse CCode implementiert. Sie enthält dabei Methoden für die Erstellung, Verwaltung und Verifizierung. Die Klasse enthält zusätzlich Sicherheits- bzw. Schutzmechanismen um die maximale Anzahl für eine Download-Site nicht zu überschreiten oder um die Datenbank beim Vergleichen nach bereits vorhanden Codes nicht zu überlasten.

Das Klassendiagramm der Code Klasse befindet sich in Abschnitt 5.5.6 Klassendiagramme.

5.5.3 Download-Site

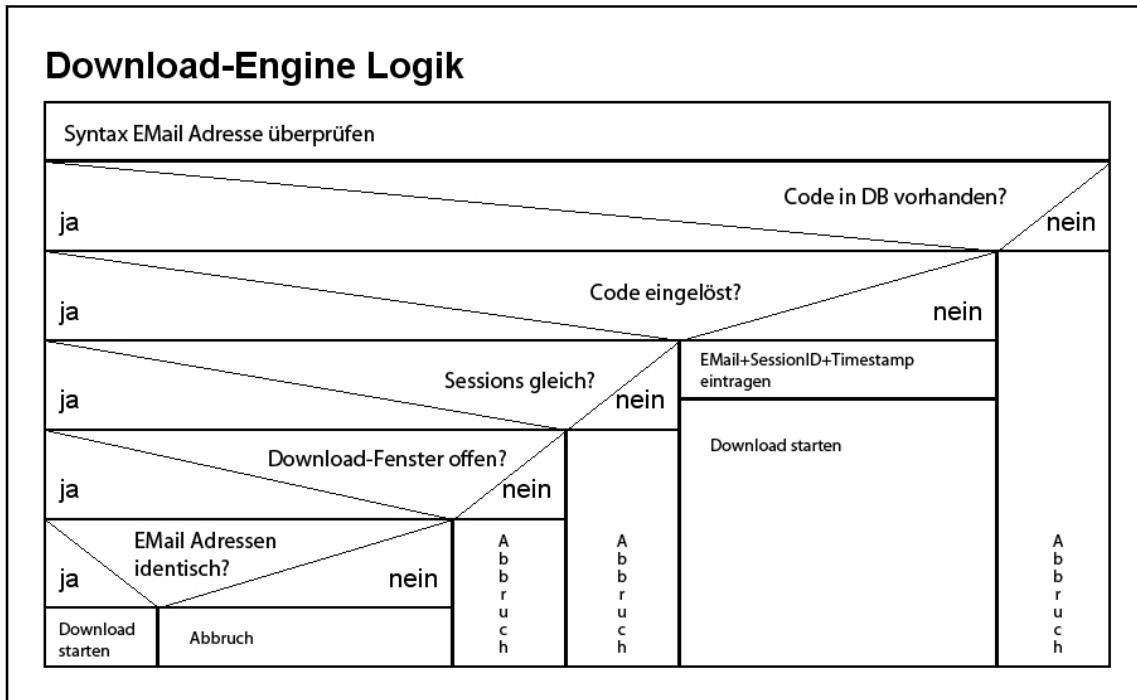
Wie schon im Abschnitt Wizard kurz angeschnitten, werden die einzelnen Download-Sites über den Wizard erstellt. Das heißt eigentlich werden hier keine „realen“ HTML Seiten erstellt sondern nur die Eigenschaften einer Site festgelegt sowie die entsprechenden Dateien wie Bilder, Banner und Download Dateien auf den Server übertragen und in der Datenbank vermerkt. Eine Download-Site bekommt eine eindeutige Seriennummer vom System zugeteilt, mit der Sie eindeutig identifiziert werden kann und über eine URL erreichbar ist. Mehr Details werden im Abschnitt 4.5.5 Template-System beschrieben.

5.5.4 Download Engine

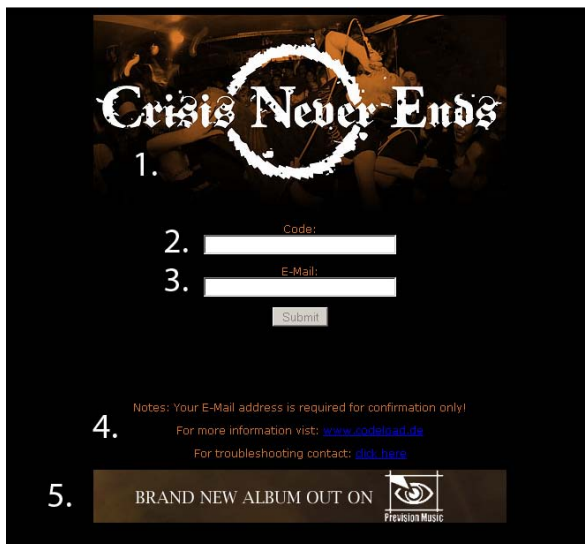
Die Download Engine und das **CodeLoad** Portal sind zwei voneinander unabhängige Programm Module. Sie haben nur die Datenbank als Basis gemeinsam. Die Download-Engine implementiert die Logik die für die Downloads verantwortlich ist. Nach erfolgreicher Verifizierung des Codes stößt sie den Download an und trägt den Code mit zusätzlichen Userinformationen in die Datenbank als eingelöst ein. Dabei wird auf z.B. folgende Logik verifiziert:

- Der Code muß in der Datenbank vorhanden sein, also durch das **CodeLoad** Portal erstellt worden sein
- Die Email Adresse zur Bestätigung des Downloads muss gültig sein
- Es wird geprüft ob der Code schon eingelöst wurde
- Falls der Code eingelöst wurde, wird das Download-Zeitfenster geprüft

Die komplette Logik ist in einem Strukturdiagramm dargestellt und enthält alle Verifizierungsschritte der Download Engine:



5.5.5 Template System



Das Template-System ist für die Darstellung der einzelnen Download-Sites zuständig. Ruft ein Benutzer eine spezielle Download-Site über eine URL auf. Wird das Template-System aufgefordert die entsprechende Download-Site aufzubauen und darzustellen. Wie das geschieht, soll im Folgenden kurz beschrieben werden

Der Kunde bekommt mit dem Code eine URL ausgehändigt unter der er die Download-Site aufrufen kann. Diese URL sieht folgendermaßen aus: <http://www.codeload.de/x6tp>

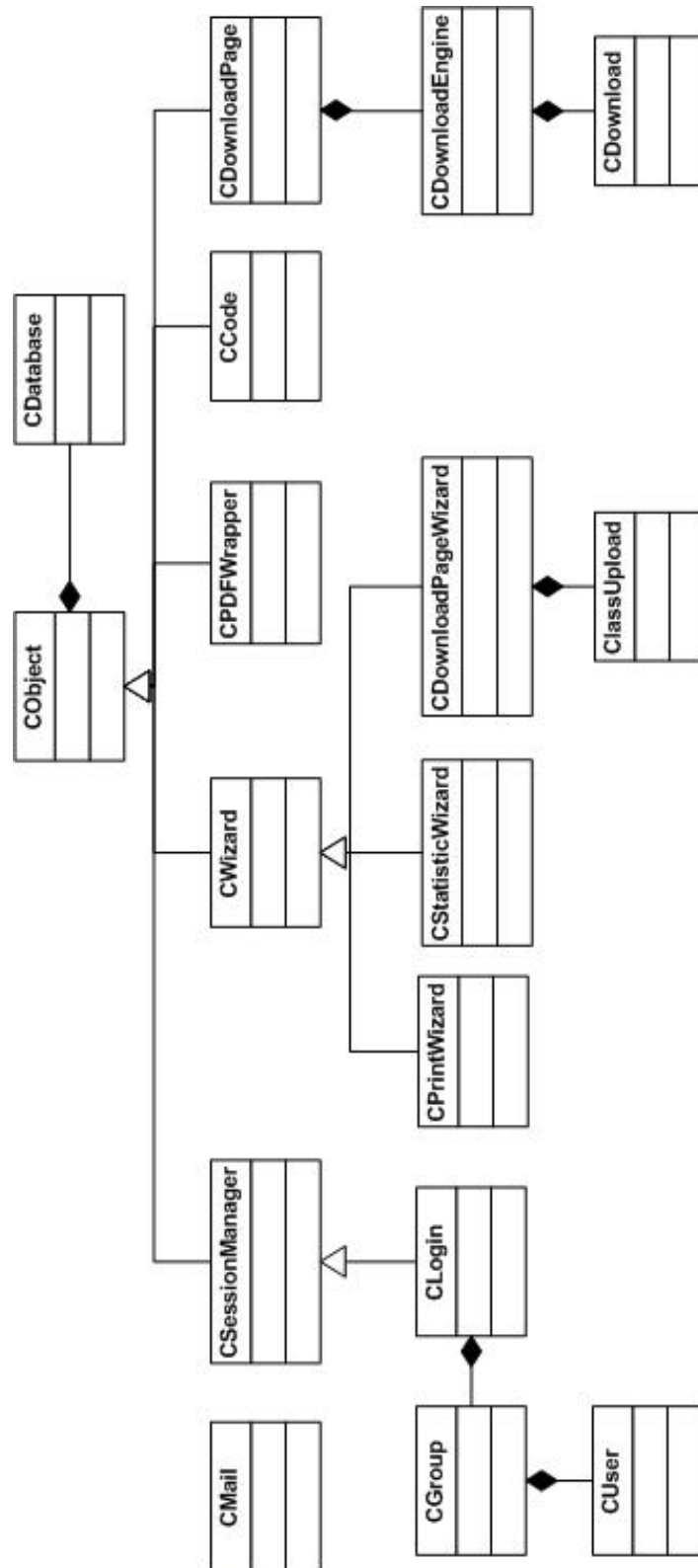
Das Zeichenkürzel am Ende repräsentiert die Seriennummer der Download-Site. Anhand dieser

Seriennummer weiß das Template-System welche Inhalte darzustellen sind. Alle durch den Download-Wizard spezifizierten Eigenschaften werden geladen und dargestellt.

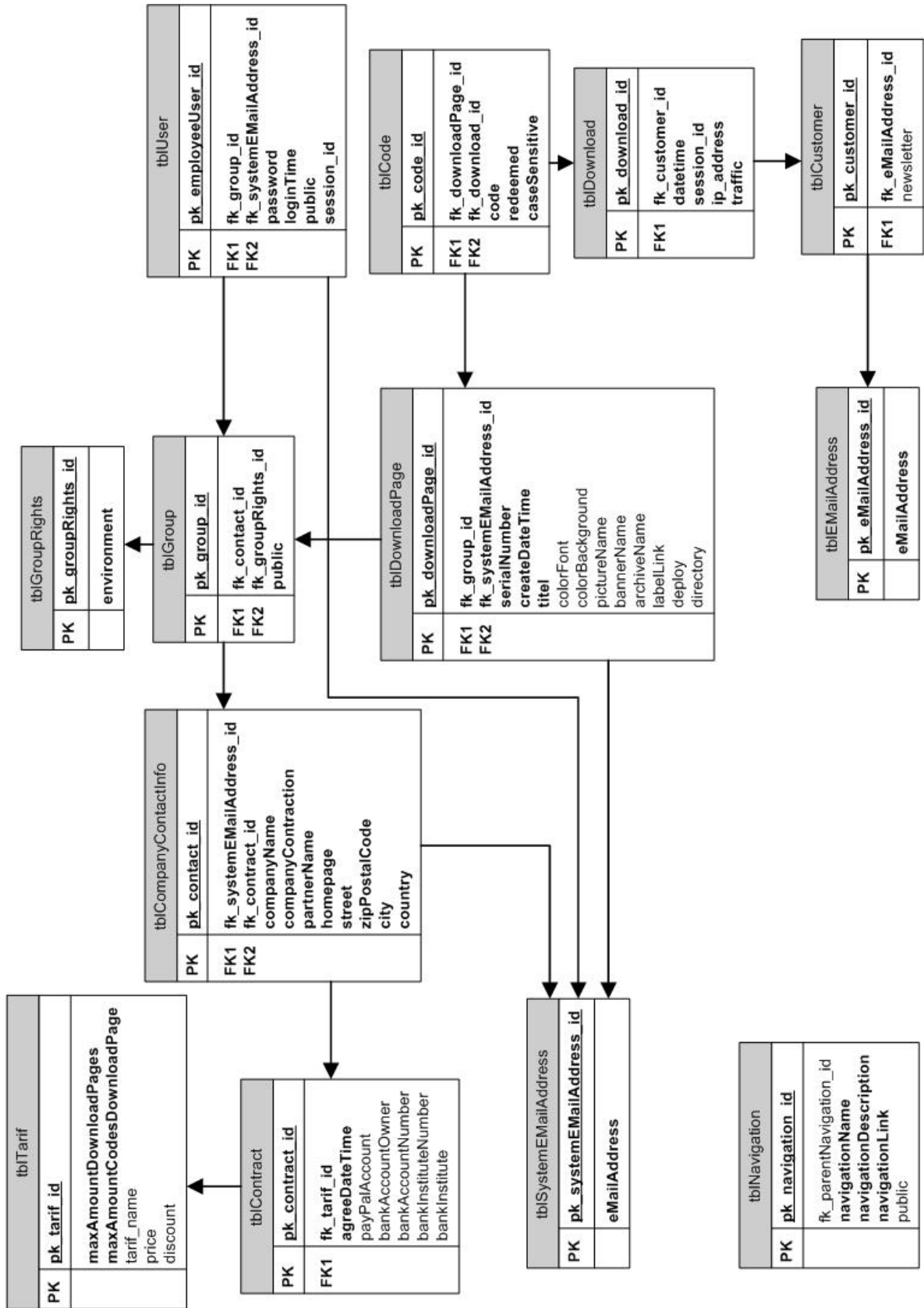
1.) Allgemeines Bild oder Logo 2.) Eingabefeld für den Code

3.) Eingabefeld für die Email Adresse 4.) Informationen zum Label und Kontakt bei Problemen 5.) Banner

5.5.6 Klassendiagramme



5.5.7 Datenbank



5.6 Qualitätsmanagement

5.6.1 Testing

Um die Qualität des endgültigen Produkts zu garantieren wurden Qualitätskriterien zu Beginn des Projekt definiert. Um deren Einhaltung zu garantieren wurden die entwickelten Produkte während und nach jedem abgeschlossenen Entwicklungszyklus auf ihre Funktion und Richtigkeit getestet. Dabei war das komplette Projektteam an diesen Tests beteiligt.

Folgende Testes wurden durchgeführt und an die Projektteilnehmer verteilt:

Kontrollstruktur	Zweck	Tester	Häufigkeit
Code-Tests	Funktionalität, Vollständigkeit und Richtig prüfen	Heiko Stumpf	Ständig und nach Bedarf
Integrationstests	Zur Kontrolle der implementierten Funktionalitäten	Heiko Stumpf	Während der Implementierung im regelmäßigen Rhythmus
Abschluss-tests	Die Abschluss-tests dienen der Qualitätssicherung	Heiko Stumpf Tomislav Basic	Nach Fertigstellung des Projekts, nach der Implementierung aller Funktionalitäten

Folgende Produkte wurden zur Qualitätssicherung eingesetzt:

- Subversion (SVN)
- Firebug
- XDebug
- Zend Debugger

5.6.2 Codekriterien/Notationen

Um einen sauberen und wartbaren Code zu schreiben, wurde gewisse Notationsregeln eingehalten und während des Projekts strikt umgesetzt. Diese Notationsregeln erzeugen mehr Schreibarbeit, haben sich aber in der Vergangenheit bewährt um Codes verständlicher und strukturierter zu gestalten.

PHP Variablen:

Obwohl PHP keine Typisierung von Datentypen wie in C oder Java kennt, wurden aus Gewohnheit und Übersichtlichkeit die Variablen nach der Ungarischen Notation benannt. Dabei wird jeder Variable ein Präfix, das den Datentyp genau identifiziert, vorangestellt.

Integer: \$iVariable

String: \$sVariable2

Boolean: \$bVariable

PHP Klassenvariablen:

Klassenvariablen erhalten zu der bereits oben genannten Ungarischen Notation noch zusätzlich das Präfix „m_“. Somit wird eindeutig eine Member Variable (Attribut) einer Klasse identifiziert.

Integer: private \$m_iVariable

PHP Klassendefinition:

Bei der Definition der Klassen wurde ebenfalls ein Präfix verwendet. Diese Definition wurde aus der Klassendefinition der Microsoft Foundation Class (MFC) übernommen. Klassen werden mit einem vorangestellten „C“ definiert, wobei das C für Class steht

Klasse Test: Class CTest{}

MySQL Tabellendefinition:

Allen Tabellen ist das Präfix „tbl“ vorangestellt um z.B. Fehler in Abfragen schneller identifizieren zu können

Tabelle User: tblUser

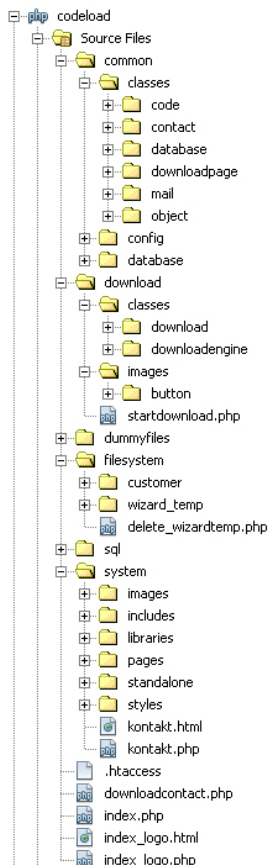
MySQL Primär- und Fremdschlüssel:

Obwohl es für die Spaltennamen keine von mir spezielle Notationsregeln gibt wird den Primär- bzw. Fremdschlüssel die Präfixe „pk_“ für den Primärschlüssel und „fk_“ für den Fremdschlüssel angehängt und jeweils mit dem Postfix „_id“ abgeschlossen.

Primärschlüssel : pk_user_id

Fremdschlüssel: fk_download_id

5.6.3 Projektdateistruktur



6 Projektplan

6.1 Projekt Produkte

Im folgenden Abschnitt sind die einzelnen Bestandteile des kompletten Projekts aufgelistet und als Teilprodukte dargestellt.

6.1.1 Codeload Version 1.0

Nach Abschluss dieses Projektes liegt dieses als erste Version zur Verfügung und stellt eine lauffähige Umgebung zur Verfügung die den definierten Anforderungen, sowie den Qualitäts- und Kundenanforderungen entsprechen.

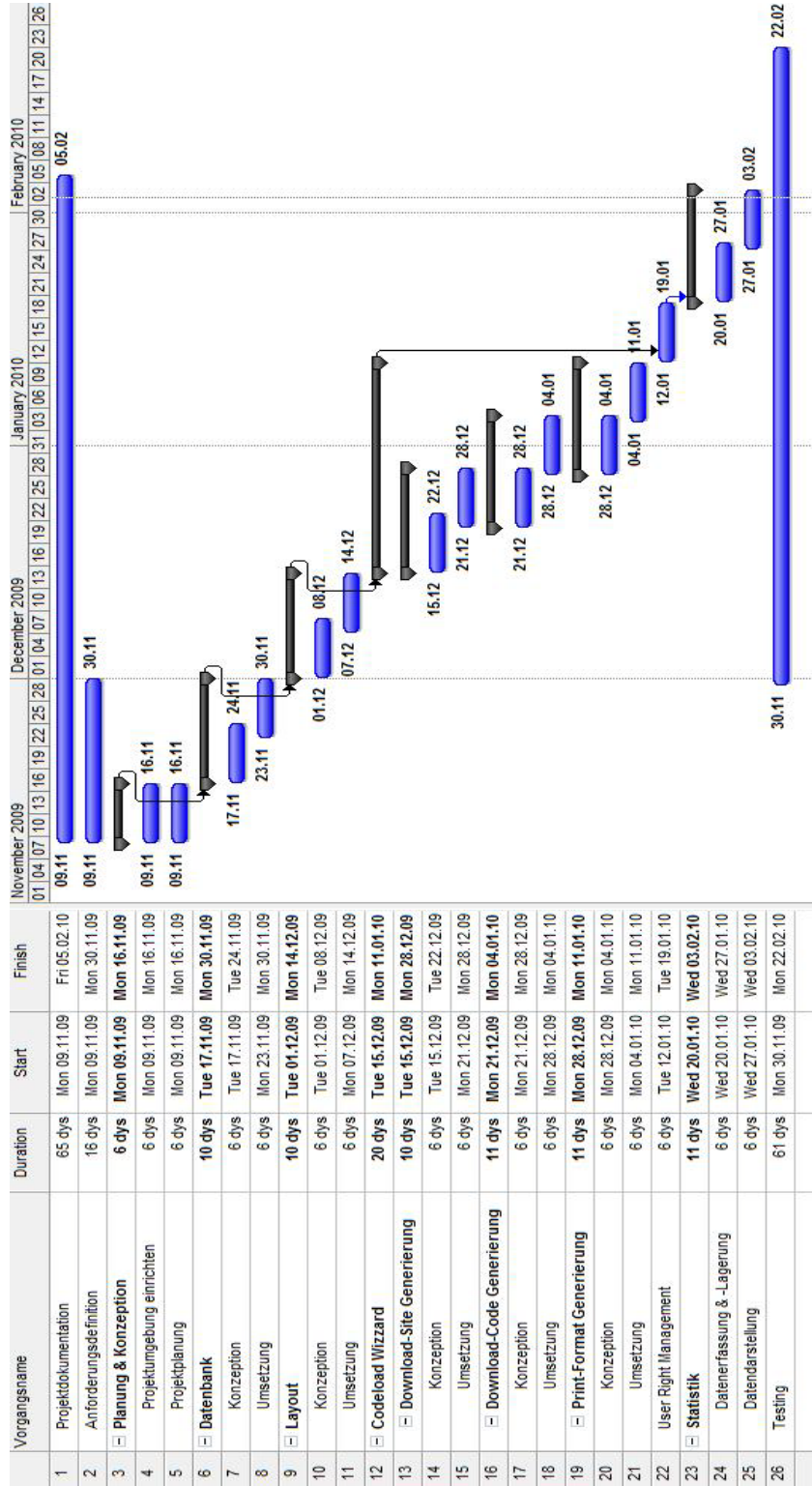
6.1.2 Präsentation

Teil dieses Projektes ist es, eine Präsentation für alle Professoren, wissenschaftlichen Mitarbeiter und interessierten Studenten zu erstellen und das Projekt am Medieninformatik Präsentationstag zu präsentieren. Hierfür wurde eine 8 minütige Präsentation erstellt, die einen groben Überblick über das Projekt, darin aufgetretene Probleme und die Vorgehensweisen bzw. Umsetzung erläutern soll.

6.1.3 Dokumentation

Diese Dokumentation ist Bestandteil des Projektes und dient als Nachschlagewerk sowie der Nachvollziehbarkeit aller Projektaktivitäten. Diese Dokumentation wurde im Rahmen der Vorlesung IT-Projektmanagement erstellt und besteht aus Produkten nach dem Prince2 Vorgehensmodell.

6.2 Gant-Diagramm



6.3 Change Management

Im Folgenden werden die Change Requests/Anforderungs-Änderungen aufgeführt die im Laufe des Projekts aufgetreten sind. Eine Änderung des Funktionsumfangs bedarf der Rücksprache des Projektleiters und wurden nur stattgegeben falls es sicher hierbei um Show Stopper oder zeitkritische Aktionen handelte die das Projektziel nicht gefährdeten.

ID	1
Aktueller Zustand	Zugriffsrechte beschränkt auf Webportal-Admin, Gruppen-User, Kunden
Gewünschter Zustand	Zugriffsrechte für Webportal-Admin, Gruppen-Admin, Gruppen-User, Kunden
Kommentar, Hinweise	Durch den CR gibt es für eine Gruppe nur noch einen User, eine Mehr-User Integration wird ausgeschlossen.
Geschätzter Aufwand	-5 Tage
Einschätzung der Notwendigkeit	Lässt sich aus Zeitgründen nicht umsetzen. Andere Funktionen haben Priorität. Fehlen der Funktion ist kein Show Stopper

7 Risikomanagement

Nr.	Risikofaktor	Auswirkung	Gegenmaßnahme	Wahr.	Status
1	Fremdeingriffe durch SQL-Injections und Cross-Site-Scripting	Datenbank kann gelöscht werden, Übernahme des Systems durch Dritte	System von Beginn an auf Sicherheitslücken untersuchen und entsprechend programmieren	4	erledigt
2	Server überlastet	Server nimmt Anfragen nicht mehr entgegen	Optimierung der Skripte	2	Noch nicht eingetreten
3	System wird gehacked	Dienst kann nicht mehr angeboten werden	System offline schalten und Sicherheitslücken schließen	3	Noch nicht eingetreten
4	Einarbeitung in unbekannte Themen	Einarbeitung und einlesen in Dokumentationen verhindert das planmäßige Vorankommen des Projekts	Puffer einplanen bei unbekanntem Themen	5	erledigt
5	Ausfall eines Entwicklers	Projektstopp	Weitere Entwickler	2	offen
6	Nicht alle gängigen Browser können unterstützt werden	System läuft nicht auf allen gängigen Browsern	Optimierung der Skripte auf Browserkompatibilität	4	Erledigt
7	Zu hoher Traffic durch	Nachzahlung der	Auf Flatrate-	4	Noch nicht

	Downloads wird erzeugt	Überschreitung	Vertrag umsteigen		eingetreten
8	System ist zu unperformant	System kann viele parallele Anfragen nicht verarbeiten	Architektur Umbau	3	Noch nicht eingetreten
10	System wird wegen Urheberrechtsverletzungen (illegale Downloads) abgemahnt	Musik-/Plattenlabel muss für die Urheberrechtsverletzung einstehen, vertraglich festhalten bei Anmeldung	genaue Anschrift des Kunden bei Anmeldung ermitteln, Haftung ausschließen, Haftung an Plattenlabel übertragen, bei Verstoß sofort Download entfernen	3	Noch nicht eingetreten
11	Funktionen lassen sich aus Zeitgründen nicht umsetzen	Kann Stop des Projekts bedeuten	Erneute Priorisierung der funktionalen Anforderungen	3	erledigt

Eintrittswahrscheinlichkeiten:

- 1 unwahrscheinlich (<0,1 %)
- 2 geringe Wahrscheinlichkeit (0,1 – 1%)
- 3 möglich (1-10%)
- 4 wahrscheinlich (10%-50%)
- 5 höchstwahrscheinlich (>50%)

8 Kommunikation

8.1 Kommunikationsplan

Informations-empfänger	Informationsgehalt	Informationsquelle	Häufigkeit	Methode
Tomislav Biscic	Status- und Fortschrittsbericht über Projektstand	Heiko Stumpf	Regelmäßig	Meeting
Tomislav Biscic	Anforderungsänderungen/ Change Request	Heiko Stumpf	Bei Bedarf	Meeting
Heiko Stumpf	Termine, Einhaltung Meilensteine	Tomislav Biscic	Regelmäßig	Meeting
Heiko Stumpf	Sitzungsprotokolle	Tomislav Biscic	Regelmäßig	Meeting

8.2 Sitzungsprotokolle

Die folgenden Sitzungsprotokolle wurden nach den Sitzungsprotokollen des Projektleiters Tomislav Biscic erstellt und enthalten die für dieses Projekt wichtigen Punkte in gekürzter Form

Status: o=offen, a=in Arbeit, e=erledigt, s=ständig

Sitzungsprotokoll 1		Datum: 09.11.2009		
Datum/offen seit	Thema	Bemerkung	Status	
09.11.2009	Projektbesprechung und -vorstellung	Bei offenen Fragen Teammitglieder kontaktieren	s	
09.11.2009	Erste Projektplanung	Zeitplanung, Ressourcenplanung	a	
09.11.2009	Anforderungen	Anforderungen definieren und festlegen	o	

Sitzungsprotokoll 2		Datum: 16.11.2009		
Datum/offen seit	Thema	Bemerkung	Status	
09.11.2009	Projektbesprechung und -vorstellung	Bei offenen Fragen Teammitglieder kontaktieren	s	
09.11.2009	Erste Projektplanung	Zeitplanung, Ressourcenplanung	e	
09.11.2009	Anforderungen	Anforderungen definieren und festlegen	a	
16.11.2009	Datenbank	Konzeption und Umsetzung	o	
09.11.2009	Layout	Konzeption und Umsetzung	o	

Sitzungsprotokoll 3		Datum: 07.12.2009		
Datum/offen seit	Thema	Bemerkung	Status	
09.11.2009	Projektbesprechung und -vorstellung	Bei offenen Fragen Teammitglieder kontaktieren	s	

09.11.2009	Anforderungen	Anforderungen definieren und festlegen	e
16.11.2009	Datenbank	Konzeption und Umsetzung	e
16.11.2009	Layout	Konzeption und Umsetzung	a
07.12.2009	Download-Wizard	Konzeption und Umsetzung	o

Sitzungsprotokoll 4		Datum: 12.01.2010	
Datum/offen seit	Thema	Bemerkung	Status
09.11.2009	Projektbesprechung und -vorstellung	Bei offenen Fragen Teammitglieder kontaktieren	s
16.11.2009	Layout	Konzeption und Umsetzung	e
07.12.2009	Download-Wizard	Konzeption und Umsetzung	e
12.01.2010	Download-Engine	Konzeption und Umsetzung	o

Sitzungsprotokoll 5		Datum: 19.01.2010	
Datum/offen seit	Thema	Bemerkung	Status
09.11.2009	Projektbesprechung und -vorstellung	Bei offenen Fragen Teammitglieder kontaktieren	s
16.11.2009	Erfassung und Lagerung der Daten für Darstellung der Statistik	Konzeption und Umsetzung	o

Sitzungsprotokoll 6		Datum: 05.01.2010	
Datum/offen seit	Thema	Bemerkung	Status
09.11.2009	Projektbesprechung und -vorstellung	Bei offenen Fragen Teammitglieder kontaktieren	s
16.11.2009	Erfassung und Lagerung der Daten für Darstellung der Statistik	Konzeption und Umsetzung	e

9 Fazit

9.1 Probleme

Während der Entwicklung von Software kommt es früher oder später zu Problemen die gelöst werden müssen um den Fortschritt des Projektes zu gewährleisten. Alle Probleme die einen gewissen Schwierigkeitsgrad aufwiesen und nicht innerhalb einer gewissen toleranten Zeitspanne zu lösen waren, wurden an den Projektleiter eskaliert, falls erforderlich in den regelmäßigen Meetings diskutiert und protokolliert. Im Folgenden soll ein Überblick über die vorhandenen Probleme während der Entwicklung des Systems verschafft werden.

Die aufgeführten Probleme sind aus den Issue Logs des Projektleiters Tomislav Bisic entnommen.

ID	1				
Aktueller Zustand	Web-Sites werden je nach Browser unterschiedlich dargestellt				
Gewünschter Zustand	Web-Sites sollen wenn möglich auf allen Browsern identisch aussehen				
Hinweise	Optimierung von CSS soll sofern möglich identische Darstellung auf allen Browsern garantieren				
Geschätzter Aufwand	3 MTage				
Bewertung	Muss erledigt werden				
Auswirkungen auf Softwarequalität	Hoch	Priorität	Hoch	Show Stopper	Ja

ID	2				
Aktueller Zustand	Anmeldevorgang funktioniert nur unter Mozilla Firefox				
Gewünschter Zustand	Anmeldevorgang muss auf allen Browsern funktionieren				
Hinweise	Anpassen der SessionManager Klasse				
Geschätzter Aufwand	2 MTage				
Bewertung	Muss erledigt werden				
Auswirkungen auf Softwarequalität	Hoch	Priorität	Hoch	Show Stopper	Ja

ID	3				
Aktueller Zustand	Nachfolgende Bearbeitungsschritte sind bei allen Wizard Varianten möglich in denen eine Download-Site ausgewählt werden muss, selbst wenn diese nicht ausgewählt wurde				
Gewünschter Zustand	Bearbeitungsschritte dürfen nur noch Auswahl der Download-Site erfolgen				
Hinweise	Sperrern des Next-Button, Wizard Klasse anpassen				
Geschätzter Aufwand	2 MTage				
Bewertung	Muss erledigt werden				

Auswirkungen auf Softwarequalität	Hoch	Priorität	Hoch	Show Stopper	Ja
--	------	------------------	------	---------------------	----

ID	4				
Aktueller Zustand	Eingabedaten werden nicht temporär gespeichert, solange der Wizard ausgeführt wird (Übergabe nicht möglich)				
Gewünschter Zustand	Eingabedaten-Übergabe zum nächsten Bearbeitungsschritt soll gewährleistet sein				
Hinweise	Eingabedaten müssen über die Übergabeparameter verwaltet werden (Globales Session Array)				
Geschätzter Aufwand	2 MTage				
Bewertung	Muss erledigt werden				
Auswirkungen auf Softwarequalität	Hoch	Priorität	Hoch	Show Stopper	Ja

ID	5				
Aktueller Zustand	Transaktionen mit der Datenbank sind nicht möglich				
Gewünschter Zustand	Transaktionen mit der Datenbank müssen möglich sein um Konsistenz der erzeugten Codes zu garantieren				
Hinweise	MySQL-Bibliothek kann keine Transaktionen. Umstieg auf MySQLi Klasse (i = improved) ermöglicht Transaktionen				
Geschätzter Aufwand	4 MTage				
Bewertung	Muss erledigt werden				
Auswirkungen auf Softwarequalität	Mittel	Priorität	Mittel	Show Stopper	Nein

9.2 Zusammenarbeit

Mit Beginn des Projekts wurde eine Zusammenarbeit des Entwicklerteams mit dem Master Studenten Tomislav Bisic vereinbart, der im Rahmen der Vorlesung „agiles Projektmanagement“ die Projektleitung übernahm. Zu Beginn des Projektes wurde das Projekt von Entwicklerseite mit Ideen und Funktionen „vollgestopft“. Diese Euphorie musste durch Herrn Bisic zu Recht gebremst werden, da hier die zeitlichen Ressourcen einfach zu knapp bemessen waren.

Aufgrund der regelmäßigen Meetings und Absprachen wurde dem Projektteam eine gewisse Sicherheit gegeben sich größtenteils auf die eigenen Aufgaben konzentrieren zu können und diese erfolgreich zu absolvieren.

Über die durchweg positive Zusammenarbeit möchte ich mich hiermit nochmals bedanken.

9.3 Projektmanagementtool

Um die Kommunikation zwischen Entwicklerteam und Projektleitung so gut wie möglich zu gestalten, um Status- und Meeting Dokumente für jeden Projektteilnehmer auf dem aktuellen Stand zur Verfügung zu stellen wurde auf das Projektmanagementtool Redmine zurückgegriffen. Ohne dieses

Managementtool wäre die Kommunikation zwischen den einzelnen Parteien womöglich nicht so gut und reibungslos abgelaufen.

9.4 Schlussfazit

Dieses Projekt wurde aus mehreren Gründen begonnen und durchgeführt. Einige Gründe wurden im Verlauf der Dokumentation schon aufgeführt, welche hier nochmals genauer aufgeführt werden sollen.

- Es bestand ein reales Interesse an diesem Projekt von Musik- und Plattenlabels. das Projekt kann also nach erfolgreicher Umsetzung weitergeführt und ausgebaut werden.
- Ich interessiere mich schon seit längerem für die Web-Entwicklung, leider kommt dieser Bereich meiner Meinung nach im Studium zu kurz. Privat fehlt mir leider die Zeit mich damit intensiver zu beschäftigen, was ich gerne tun würde. Hierfür war dieses Projekt ein idealer Anlass um Einblicke in die dynamische Web-Entwicklung zu bekommen.
- Ich habe mich entschieden mein Praxissemester in einer Internet/Web Agentur zu absolvieren, hierfür war es Voraussetzung Grundkenntnisse im Umgang mit PHP und JavaScript zu besitzen, welche ich mir zu Beginn des Projekts aneignen konnte.

Wie bei jedem Projekt werden Entscheidungen getroffen die man im ersten Moment für richtig im späteren Verlauf aber eventl. für falsch ansieht. Im Nachhinein würde ich 2 Themen anderes angehen.

- Framework:
Die Entscheidung, auf komplett ein Framework zu verzichten und ein eigenes zu schreiben war für meinen persönlichen Lerneffekt sehr wertvoll. Ich habe dadurch viel über PHP und Architektur Prinzipien für dynamische Webanwendungen gelernt und möchte dies nicht missen. Allerdings habe ich gemerkt das diese Entscheidung sehr viel zeit in Anspruch genommen hat und die Architektur oft re-engineered und überarbeitet werden musste. Endgültig habe ich es aber geschafft das Framework nach meinen eigenen Vorstellungen zu entwickeln.
Für zeitkritische Projekte, welches Projektarbeiten nun mal sind, würde ich in Zukunft mehr auf fertige Lösungen zurückgreifen um dadurch mehr Zeit für funktionale Themen zu haben.
- Design:
Beim Design habe ich zu stark auf Grafikelemente wert gelegt, dies sieht zwar schön aus, erschwert allerdings die Entwicklung mit CSS. In Zukunft werde ich dafür mehr mit gewöhnlichen HTML und CSS Stilen und Eigenschaften arbeiten. Auch habe ich unterschätzt wie viel Arbeit in ein gutes und „funktionierendes“ Design investiert werden muss. Nicht umsonst gibt es Grafiker die sich speziell mit dem Design von Webseiten auseinandersetzen. Auch hier habe ich wieder einiges gelernt. Nicht nur den Umgang mit Grafikprogrammen sondern auch Grundsätzliche Dinge wie z.B. die Farbwahl oder Auswahl einer passenden Schriftart.

Abschließend möchte ich sagen, dass dies nicht mein letztes Web Projekt war. Während meiner Ausbildung zum Fachinformatiker habe ich viel mit C++ programmiert, hier kam mir einfach der visuelle Teil zu kurz. Dies ist ein Punkt der mich an der Entwicklung von Webseiten und –Anwendung

fasziniert, da hier neben der eigentlichen Funktionalität auch ein großer Wert auf gestalterische Bereiche wert gelegt wird.

Wendlingen, den 20.02.2010

Heiko Stumpf

10 Quellangaben

10.1 Internet

- <http://www.pc-erfahrung.de/linux/netzwerk/subversion-unter-ubuntu.html>
- <http://www.howtogeek.com/howto/ubuntu/install-subversion-with-web-access-on-ubuntu/>
- <http://davidwinter.me.uk/articles/2006/02/16/subversion-over-apache-2-on-ubuntu/>
- <http://php.net/docs.php>
- http://www.zdnet.de/anwendungsentwicklung_eigener_error_handler_mit_php_story-20000201-39147726-2.htm
- <http://www.fukurama.org/wordpress/category/technik/linux-ubuntu/page/5/>
- <http://www.alvit.de/css-showcase/css-navigation-techniques-showcase.php>
- <http://superfluosbanter.org/archives/2003/12/the-art-of-navigation/>
- <http://devzone.zend.com/article/687#Heading1>
- <http://www.tsql.de/php/aufruf-der-authentifizierung>

10.2 Bücher

- Einstieg in SQL von Marcus Throll, Oliver Bartosch
- PHP 5. Grundlagen und Profiwissen von Jörg Krause
- PHP Sicherheit von Christopher Kunz und Stefan Esser
- CSS-Praxis: Modernes Webdesign mit CSS von Kai Laborenz
- GIMP - ab Version 2.6 - Für digitale Fotografie, Webdesign und kreative Bildbearbeitung von Bettina K. Lechner
- JavaScript das Umfassende Handbuch von Christian Wenz

10.3 Freie Bibliotheken

- <http://www.fpdf.org/>

10.4 Programme und Tools

- Ubuntu 9.10
- NetBeans 6.8
- Apache WebServer
- MySQL
- Subversion
- Gimp 2.6
- Virtual Box
- Mozilla Firefox Browser
- Opera Browser
- Microsoft Internet Explorer