

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
1.1	Grundlegendes . . . . .	3
1.1.1	Betreuender Professor . . . . .	3
1.1.2	Studenten . . . . .	3
1.2	Motivation . . . . .	4
1.3	Ideen zur Projektarbeit . . . . .	4
1.3.1	Tippspiel . . . . .	4
1.3.2	Videosoftware . . . . .	4
1.4	Zielsetzung und Einsatzbereich . . . . .	5
<b>2</b>	<b>Konzepte und Aufbau</b>	<b>6</b>
2.1	Aufbau . . . . .	6
2.1.1	Beschreibung des Gesamtsystems . . . . .	6
2.1.2	das Kundenmodul im Detail . . . . .	9
2.1.3	das Versandmodul im Detail . . . . .	11
2.1.4	das Verwaltungsmodul im Detail . . . . .	12
2.2	verschiedene Versionen . . . . .	12
<b>3</b>	<b>Technologien</b>	<b>14</b>
3.1	Datenbank . . . . .	16
3.2	Versionsverwaltung . . . . .	18
3.2.1	Concurrent Versions System - CVS . . . . .	18
3.2.2	Subversion . . . . .	18
3.3	Entwicklungsumgebung . . . . .	19
3.3.1	JBuilder . . . . .	19
3.3.2	Netbeans . . . . .	19
3.3.3	Eclipse . . . . .	19
3.4	grafischen Benutzerschnittstellen in Java . . . . .	20
3.4.1	Abstract Window Toolkit - AWT . . . . .	20
3.4.2	Swing . . . . .	20
3.4.3	Standard Widget Toolkit - SWT . . . . .	20

## INHALTSVERZEICHNIS

---

3.5	Java-Web-Anwendungen . . . . .	21
3.5.1	Java Server Faces - JSF . . . . .	21
3.5.2	Struts . . . . .	21
3.6	Persistenzschichten in Java . . . . .	22
3.6.1	Java Data Objects . . . . .	22
3.6.2	Hibernate . . . . .	22
<b>4</b>	<b>Implementierung</b>	<b>23</b>
4.1	Versionsverwaltung mit Subversion . . . . .	24
4.2	Entwicklungsumgebung mit Eclipse . . . . .	25
4.3	grafischen Benutzerschnittstellen mit SWT . . . . .	26
4.4	Java-Web-Anwendungen mit Struts . . . . .	27
4.5	Persistenzschichten mit Hibernate . . . . .	28
<b>5</b>	<b>Zusammenfassung</b>	<b>29</b>
<b>A</b>	<b>Protokoll vom 11. Mai 2004</b>	<b>30</b>
	<b>Literaturverzeichnis</b>	<b>32</b>

# Kapitel 1

## Einführung

### 1.1 Grundlegendes

#### 1.1.1 Betreuender Professor

Hochschule Harz  
Prof. Dr. Sigurd Günther  
Friedrichstr. 57- 59  
38855 Wernigerode  
sguenther@hs-harz.de

#### 1.1.2 Studenten

Remo Griesch	Stefan Forstner	Philipp Schneider
Strasse	Strasse der Jugend 22	Kastanienring 16
Ort	04880 Dommitsch	04316 Leipzig
Romeodied@gmx.de	fossiosi@web.de	provirent@phil-schneider.de

## 1.2 Motivation

Im Rahmen des Studiums an der Fachhochschule Harz in Wernigerode muß jeder Student des Studiengangs Kommunikationsinformatik eine Projektarbeit abgeben. Dies bedeutet, daß der Student eine Aufgabe (meist Programmieraufgabe) alleine oder in einem kleinen Team bewältigen muss. Die Professoren der Hochschule bieten dabei viele interessante Projektarbeiten an, sind jedoch offen für eigene Vorschläge der Studenten.

Da schon in den Teamprojekten<sup>1</sup> *Labmin*<sup>2</sup> und *German Team Sony Aibo*<sup>3</sup> eine interessante Aufgabe von den Studenten gelöst wurde, sollte das dort erlernte Wissen vertieft und weiter ausgebaut werden.

## 1.3 Ideen zur Projektarbeit

### 1.3.1 Tippspiel

Die erste Idee dieser Projektarbeit war die Umsetzung eines Tippspiels in Java, passend zu den damaligen Fussball-Europameisterschaft in Portugal. Diese Idee wurde im JavaMagazin<sup>4</sup> in mehreren Ausgaben aufgegriffen und verschiedene Ansatzmöglichkeiten diskutiert. Die Idee unseres Tippspiel war dabei eine Webanwendung mit Datenbankbindung. Nutzer dieses Systems sollten sich in verschiedenen Tippgemeinschaften, mit je einem Tippgemeinschaftsverwalter, zusammen tun und gemeinsam die EM 2004 tippen. Das Tippspiel sollte jedoch nicht nur auf die EM 2004 zugeschnitten sein, sondern auch für andere Fußballereignisse tauglich sein. Zusätzlich kam von unserer Seite die Idee, eine Webanwendung zur Verwaltung der Bundesligaergebnisse. Ein Tippspiel System sollte dann auf diese Daten zurückgreifen und so ein Bundesligatippspiel darstellen können.

Dieser Gedanke wurde jedoch aus verschiedenen Gründen verworfen. Zum einen war es nicht unsere Idee, sondern die des Javamagazin's und zum anderen wussten wir nicht sofort was bei diesem System alles zu realisieren war. Auch war uns die Funktionsweise dieses Tippspiels nicht sofort klar.

### 1.3.2 Videosoftware

Da jeder von uns schon einmal ein Video in einer Videothek ausgeliehen, kam uns der Gedanke einer Onlinevideothek. Solche Videotheken gab es mittlerweile schon

---

<sup>1</sup> Auch das Teamprojekt ist Bestandteil des Studiums. Beim Teamprojekt müssen mehrere Studenten (7-15) gemeinsam eine Programmieraufgabe umsetzen.

<sup>2</sup> <http://labmin.de.vu>

<sup>3</sup> <http://www.der-baer.com/projects.htm>

<sup>4</sup> [Frotscher 2004a][Frotscher 2004b][Frotscher 2004c]

wie bspw Amango<sup>5</sup>, Netleih<sup>6</sup>, Invdeo<sup>7</sup> und Verleihshop<sup>8</sup>. Bei genauer Betrachtung dieser Onlinevideotheken, fragten wir uns wie solch eine Videothek technisch funktioniert. Da wir gerade auf der Suche nach einem idealen Projekt waren, hatten wir damit eins gefunden.

Es sollte versucht werden eine Online-Videothek mit entsprechenden Modulen zu realisieren.

## 1.4 Zielsetzung und Einsatzbereich

### Muss noch überarbeitet und zusammengefasst werden

Zielsetzung dieses Projektes ist dabei Erfahrung mit verschiedenen neuen Technologien zu sammeln und selbständig an einem Projekt zu arbeiten. Sowohl die eigene Gedanken, Ideen, Planung und auch Realisierung dieses Projektes sollten uns auf eine spätere Eigenverantwortung im Berufsleben vorbereiten. Das Projekt sollte dabei keine vollständige und fehlerfreie Implementierung darstellen. Uns war bewußt, dass wir nur einen einfachen Prototypen einzelner Module realisieren könnten.

Diese Software ist sowohl für kleine als auch für grosse Unternehmen gedacht. Dabei ist es unwichtig, ob es sich um eine reine OnlineVideothek oder um eine richtige Videothek, die jetzt auch per Versand ihre Videos verleihen möchte, handelt. Durch weitere Module kann die Software so erweitert werden, dass die Software auch für eine richtige Videothek geeignet ist.

---

<sup>5</sup> <http://www.amango.de>  
<sup>6</sup> <http://www.netleih.de>  
<sup>7</sup> <http://www.invdeo.de/>  
<sup>8</sup> <http://www.verleihshop.de>

# Kapitel 2

## Konzepte und Aufbau

### 2.1 Aufbau

#### 2.1.1 Beschreibung des Gesamtsystems

Bei einer klassischen Videothek besucht der Kunde das Ladengeschäft der Videothek und stöbert dabei nach Videos, die er gerne an diesen Abend schauen möchte. Dabei muss die Videothek eine möglichst große Ladenfläche besitzen um die Videos dem Kunden zu präsentieren. Nachdem der Kunde sich für ein Video entschieden hat, nimmt er entweder die leere Verpackung oder ein Plastikschild mit einer Nummer zum Verleihschalter der Videothek. Nachdem der Kunde seine Kundenkarte vorzeigt und durch sein Passwort oder seine Unterschrift verifiziert wurde, sucht der Mitarbeiter anhand einer Nummer in der Leerverpackung oder des Plastikschilds das entsprechende Video heraus, markiert dieses Video im System und gibt es dem Kunden. Dies ist der klassische Ablauf in einer Videothek.

Bei einer Online-Videothek kann der Kunde, durch den Versand der Videos, keine Videos für den gleichen Abend ausleihen. Er ist gezwungen, sich einige Tage vorher für ein oder mehrere Videos zu entscheiden. Der Ablauf unterscheidet sich von einer klassischen Videothek. Der Kunden „besucht“ die Webseite der Videothek und sucht im Angebot nach Filmen die er sich ausleihen möchte. Nachdem die Verfügbarkeit überprüft wurde, legt er Videos in seinem Warenkorb ab. Durch Eingabe seines Benutzernamens und das zugehörige Passwort wird der Kunde verifiziert. In dem Lager der Videothek nimmt ein Mitarbeiter die Bestellung über einen Monitor oder eine ausgedruckte Liste entgegen und bearbeitet die Bestellung. Dabei sucht dieser die Videos für den Kunden heraus, nimmt die Videos in das System auf und versendet die Videos zu dem Kunden per Post. Der Kunde erhält seine gewünschten Videos, kann diese sich anschauen und schickt diese nach einer bestimmten Zeit an die Videothek zurück.

Das hier gewünschte System soll dabei eine komplette Videothek ersetzen. Die

Online-Videothek benötigt nur noch ein Lager für die zu verleihenden Videos und wenige Mitarbeiter für den Versand der Videos und die Verwaltung der Videothek. Die Software wird dabei wie in Abbildung 2.1 auf der nächsten Seite zu sehen ist, von zwei verschiedenen Personenkreisen benutzt, dem Kunden und dem Mitarbeiter. Der Kunde kann die in der Abbildung dargestellten Aktionen ausführen, wie bspw. betrachten und bestellen von Videos. Der Mitarbeiter kann dabei das System verwalten und Bestellungen der Kunden bearbeiten.

Nach einiger Überlegung wurde festgestellt, dass die Software aus drei anstatt zwei Modulen bestehen muss, wie in Abbildung 2.2 auf Seite 9 zu sehen ist. Bei den Mitarbeitern der Online-Videothek muss in Verwaltung und Versand/Lagen unterschieden werden, da diese unterschiedlichen Aufgaben von unterschiedlichen Mitarbeitern bearbeitet werden. Das **Kundenmodul** ist die Internetpräsenz der Videothek und repräsentiert das Unternehmen nach aussen. Auf dieser dynamischen Webseite kann der Kunde die vorhandenen Videos durchstöbern und detaillierte Informationen zu den Videos erhalten. Nach erfolgreicher Anmeldung im System kann der Kunde die Verfügbarkeit des jeweiligen Videos kontrollieren und auf Wunsch Videos ausleihen. In einem zusätzlichen Menüpunkt kann er seine bestellten Videos betrachten und sich ggf. Rechnungen ausdrucken. Das **Versandmodul** stellt die benötigte Software für das Lager und den Versand zur Verfügung. Mit deren Hilfe kann ein Mitarbeiter der Online-Videothek Videos für den Versand vorbereiten. D.h. der Mitarbeiter bekommt eine Liste mit Bestellungen von Kunden (elektronisch oder auf Papier) und arbeitet diese ab. Damit der Mitarbeiter nicht jedes mal die Kundennummer und Nummern der Videos eintippen muss, wird seine Arbeit durch Barcodes und Barcodescanner unterstützt. Mit dessen Hilfe markiert er Videos für einen bestimmten Kunden und eine bestimmte Bestellung und versendet diese. Dem System teilt der Mitarbeiter dadurch mit, dass bestimmte Videos nicht mehr verfügbar sind und von einem bestimmten Kunden ausgeliehen wurde. Rechnungen, Versandetiketten und eventuelle Lieferscheine werden dabei automatisch mit Hilfe eines Druckers erstellt. Zusätzlich bietet das Versandmodul die Möglichkeit, zurückgekommene Videos der Kunden wieder in das System aufzunehmen. Somit wurde das Video wieder vom Kunden zurückgegeben und es kann im System als vorhanden/ausleihbar markiert werden oder gleich an den nächsten Kunden weitergeschickt werden. Das **Verwaltungsmodul** hilft den Mitarbeitern in der Verwaltung bei der Organisation der Online-Videothek. Es können Kundendaten und Rechnungen betrachtet und ggf. gedruckt werden. Das Videosortiment kann bearbeitet und inhaltliche Änderungen (z.B. Sonderangebote) an dem Kundenmodul vorgenommen werden. Weiterhin besteht die Möglichkeit ausführliche Reports & Statistiken zu erstellen und zu betrachten.

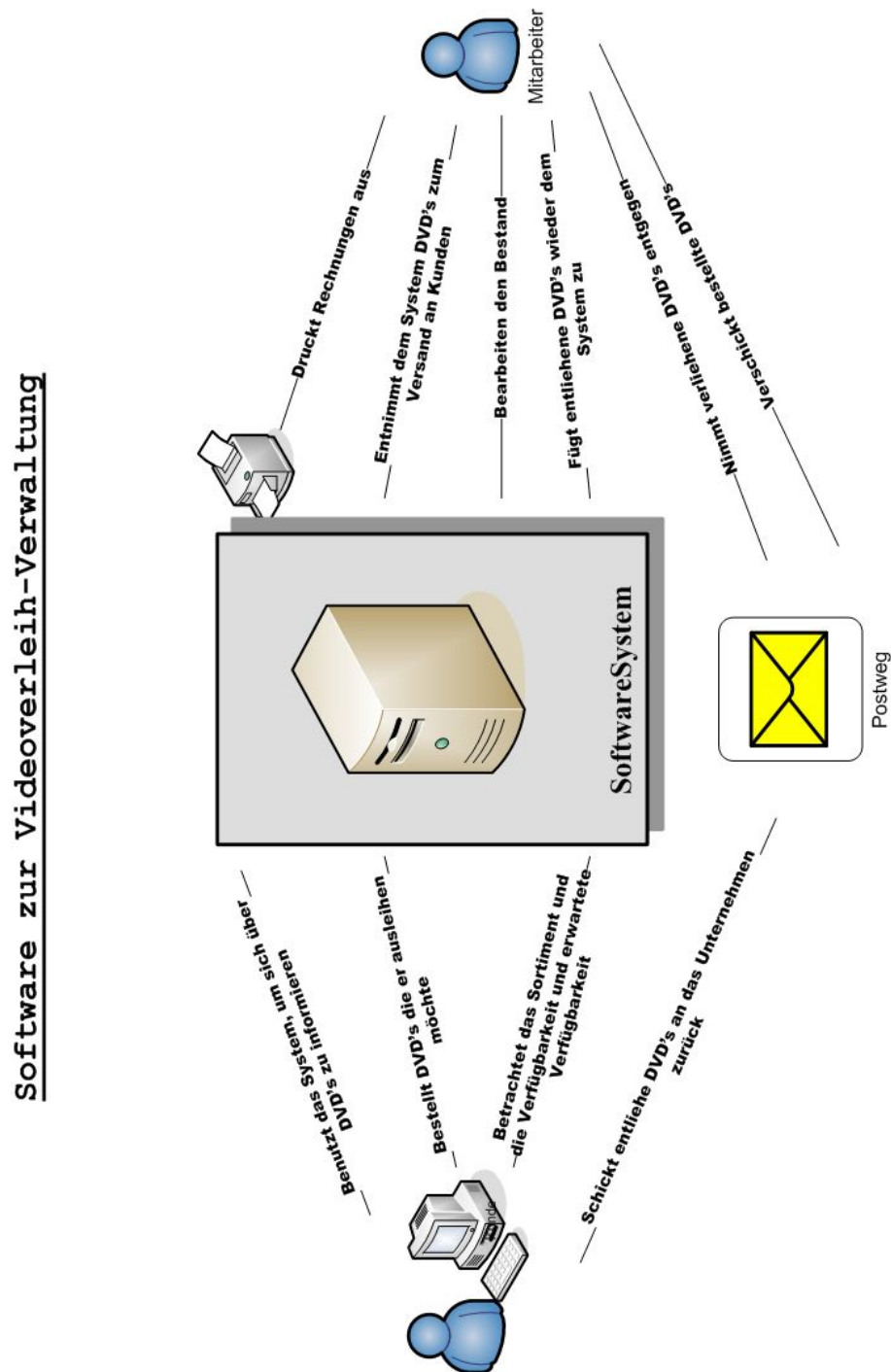


Abbildung 2.1: Erste Gedanken zu der Videosoftware



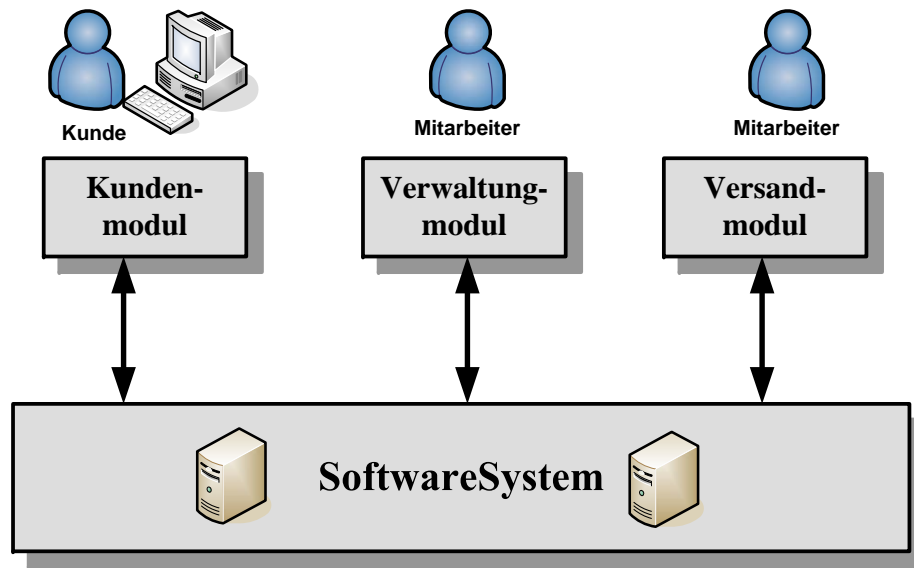


Abbildung 2.2: Die drei Module der Software

### 2.1.2 das Kundenmodul im Detail

Das Kundenmodul ist das wichtigste Element der Anwendung, denn dieses Modul wird vom Kunden verwendet und dieser bestimmt über Erfolg oder Misserfolg der Online-Videothek und damit über die Anwendung. Das Kundenmodul wird, wie bereits erklärt, als Webanwendung entwickelt. Somit muss der Nutzer keine spezielle Software auf seinem Rechner verwenden und kann die Software somit überall verwenden. Grundsätzlich wird bei der Anwendung zwischen zwei verschiedenen Kundengruppen unterschieden: dem *Interessenten* und dem *angemeldeten Kunden*. Ein *Interessent* ist ein Websurfer, der sich über das Angebot der Online-Videothek interessiert, ohne bisher ein Video ausgeliehen zu haben. Ist der Interessent von dem Angebot überzeugt und möchte ein Video ausleihen, muss er sich im System registrieren. Dazu gehört neben einem Benutzernamen und Passwort auch seine vollständige Adresse, für den Versand der Videos bzw. Rechnungen, und seine Bankverbindung bzw. Kreditkartendaten, für die Bezahlung. Ist Kunde in das System eingeloggt, ist er ein angemeldeter Kunde. Je nach Kundengruppe präsentiert sich die Webseite mit anderen Funktionalitäten. Zuerst sollen die Funktionen und Möglichkeiten eines Interessenten erklärt werden. Der Interessent hat kann sich auf der Webseite der Online-Videothek sowohl über das Angebot an Videos, als auch über die Online-Videothek informieren. Über verschiedene Webseiten erhält er Einblick in die für ihn interessante Funktionsweise der Online-Videothek und allgemeinen Daten, wie Impressum, Kontaktdaten und Allgemeine Geschäftsbe-

dingungen. Über eine intelligente Suche und Navigationselementen kann er nach Videos suchen bzw. stöbern. Die Videos sind kategorisiert bzw. geordnet. Der Interessent kann auch detaillierte Informationen über einzelne Filme erhalten. Dabei kann er sich über die Features des Filmes informieren, Kommentare bzw. Bewertungen lesen und ähnliche Filme betrachten. Die dabei verwendete Suche wird als intelligente Suche bezeichnet, weil diese Suche auch ähnliche bzw. verwandte Filme findet und sowohl im Titel, als auch in der Beschreibung der Filme nach dem Stichwort sucht. Es könnte auch realisiert werden, dass die Suchergebnisse nach Trefferquoten geordnet werden bzw. das Suchfeld automatisch das Wort nach häufigen Suchbegriffen ergänzt.

Ist der User eingeloggt hat er die gleiche Funktionsweise wie ein Interessent, nur mit erweiterten Funktionen. In der Ergebnissliste der Suche oder der Anzeigeliste der Kategorien, welche beide gleich aufgebaut sind, befinden sich neben jedem Video Elemente zum bestellen und zum Überprüfen der Verfügbarkeit. Jedes Video, also jeder Film, ist in einer gewissen Anzahl vorhanden. Mit Hilfe der Überprüfung der Verfügbarkeit kann der Kunde überprüfen, ob noch ein Video dieses Filmes für ihn zum Ausleihen vorhanden ist, bzw. wann das nächste Video wieder verfügbar ist. Je nach Status der Verfügbarkeit kann der Kunde das Video in seinem Warenkorb legen und damit ausleihen, oder vormerken lassen. Bei einer Vormerkung wird er entweder per Email informiert, dass das Video vorhanden ist, oder das Video wird schnellstmöglich an den Kunden versendet. Hat der Kunden Videos in seinen Warenkorb abgelegt, kann dieser am Ende seiner Bestellung zur Kasse gehen und somit die Videos ausleihen. Weiterhin hat der Kunde die Möglichkeit eine so genannte *Wunschliste* zu Erstellen. Diese Liste kann eine bestimmte Anzahl an Videos anschauen, die der Kunde schauen möchte. Das System arbeitet diese Liste automatisch ab, indem es dem Kunden eine bestimmte Anzahl an Videos mit einmal zusendet und bei Rücksendung durch den Kunden die nächsten Filme automatisch an den Kunden sendet. Somit hat der Kunden z.B. die Möglichkeit 50 Filme zu bestimmen, die er gerne anschauen möchte. Das System sendet ihm immer zwei Filme mit einmal zu, die er sich anschauen kann. Sendet er diese zwei Filme zurück, bekommt er die nächsten zwei Filme, bis die Liste abgearbeitet wurde. Weiterhin bietet das Kundenmodul dem Kunden die Möglichkeiten, ältere und aktuelle Bestellungen bzw. Vorbestellungen zu betrachten, Rechnungen auszudrucken und allgemeine Einstellungen an seinem Profil vorzunehmen. Das Kundenmodul soll dabei einfach und ohne Bedienungsanleitung zu bedienen sein, der Kunden soll sich schnell zu recht finden und schnell zu seinem Ziel, einer Bestellung, kommen.

### 2.1.3 das Versandmodul im Detail

Das Versandmodul ist der Bestandteil der Videothek in der die Arbeit geschieht. Hier werden Videos herausgesucht, verpackt und an den Kunden versendet. Dabei wird der Mitarbeiter soweit wie Möglich von der Technik unterstützt. Bei dieser Technik handelt es sich hauptsächlich um Barcodescanner.

Ein Barcodescanner ist ein Lesegerät, das mit Hilfe eines Lasers einen Strichcode auf einer Verpackung oder einem Blatt liest. Dieser Strichcode kann dabei je nach Art und Weise eine Nummer oder einen Wort repräsentieren. Diese Strichcodes sind aus dem heutigen Alltag nicht wegzudenken, denn sie sind auf jeder Verpackung vorhanden und werden in Supermärkten als Preisschild verwendet. Dabei hat jedes Produkt eine eindeutige Nummer, zu der mit Hilfe einer Datenbank ein Preis zugeordnet wird. Neben Strichcodes gibt es noch zweidimensionale Codes, die mehr Informationen speichern können.

An einem Zentralen Rechner werden die Bestellungen der Kunden ausgedruckt. Solch ein Ausdruck besteht aus einer Seite, auf der oben zwei Adresstiketten und unten einem Lieferschein vorhanden sind. Das erste Adresstickett ist das für den Versand zu dem Kunden, das auf den Umschlag der Bestellung geklebt wird. Das zweite ist dasjenige, welches der Kunde auf dem Umschlag zur Rücksendung der Video klebt. Der Lieferschein ist sowohl für den Mitarbeiter als auch für den Kunden wichtig. Der Mitarbeiter liest mit Hilfe eines Barcodescanners die Bestellnummer ein. Aus dieser Rechnungsnummer kann der Computer sowohl auf den Kunden als auch auf die Videos schließen, die versendet werden sollen. Der Mitarbeiter sucht, mit Hilfe einer weiteren Software den Lagerort der jeweiligen Videos heraus und scannt deren eindeutige Nummer. Damit sind die Videos im System nicht mehr verfügbar und dem Kunden zugeordnet. Anschließend verpackt er die Videos in einen entsprechenden Umschlag und verschickt die Videos mit dem Ausdruck an dem Kunden. Bei diesem Vorgang gibt es mehrere Möglichkeiten wie der Mitarbeiter vorgehen kann. Damit der Mitarbeiter nicht für jede Bestellung durch das Lager gehen muss, kann das System eine Liste mit Videos erstellen, für die nächsten 10 Bestellungen. Somit sucht der Mitarbeiter diese Videos heraus und bearbeitet dann nacheinander diese zehn Bestellungen. Eine weitere Möglichkeit wäre, das der Mitarbeiter und das Lager durch ein automatisiertes Lager ersetzt wird. Dabei übernimmt ein Roboter in einem Lager die Aufgabe des Suchen und Finden der Videos. Schickt der Kunde die Videos an die Online-Videothek zurück, nimmt ein Mitarbeiter die Videos entgegen. Zuerst scannt er dabei die Bestellnummer und die eindeutige Nummer des jeweiligen Videos. Zusätzlich muss er den Zustand der Videos überprüfen und im System eintragen. Anschließend sind die Videos wieder im System verfügbar und können wieder eingeordnet werden oder an den nächsten Kunden versendet werden.

## 2.2 verschiedene Versionen

---

Das komplette Lager könnte durch spezielle Maschinen vollständig automatisiert werden.

### 2.1.4 das Verwaltungsmodul im Detail

- Supporttelefon - bestellungen betrachten, verändern - kundendaten betrachten - neue Filme ins System aufnehmen - neue Videos(dvds, vhs) ins system aufnehmen - preise ändern - sonderaktionen hinzufügen - statistiken (auslastung) betrachten,exportieren und drucken

Die Verwaltungssoftware wird von Mitarbeitern der Videothek bedient. Mit Hilfe dieser ist es möglich das Video-Sortiment zu bearbeiten und zu vervollständigen. Weiterhin ist es möglich mit der Verwaltungssoftware Rechnungen von Kunden, Bestellungen oder ähnliches zu bearbeiten. Zusätzlich steht eine ausführliche Statistik zu einzelnen Kunden, einzelnen Videos und Genre zur Verfügung.

## 2.2 verschiedene Versionen

Die einfache Version des System wird von uns implementiert. Diese Version soll die wichtigsten Bestandteile des Systems enthalten, jedoch nicht zu komplex sein. Es soll außerdem möglich sein, diese Version durch Updates mit zu erweitern, sodaß eine komplexes System entstehen kann.

Diese einfache Version beinhaltet die Kundensoftware(Webanwendung) und die Lagersoftware in einfacher Form. Dies bedeutet für die Lagersoftware, dass keine grafische Oberfläche vorhanden ist. Als Ausgabemedium dient die Konsole, die man mit einem Dos Fenster vergleichen kann.

Eine Verwaltungssoftware ist nicht geplant. Diese wird jedoch zu einem späteren Zeitpunkt auch als Consolenanwendung realisiert. Die einfache Version ist als Testversion gedacht, bei der die Daten bereits in der Datenbank vorhanden sind, somit ist eine Verwaltungssoftware erstmal überflüssig.

- Kundenmodul

- 

- Verwaltungsmodul

- grafische Benutzeroberfläche die unter Windows und Linux verfügbar ist
  - neue DVD's hinzufügen bzw. bearbeiten & löschen
  - ausführliche Statistiken

## 2.2 verschiedene Versionen

---

- 
- Versandmodul
  - grafische Benutzeroberfläche die unter Windows und Linux verfügbar ist
  - automatischer Ausdruck der Versandlabels und Rechnungen

# Kapitel 3

## Technologien

In diesem Abschnitt werden kurz die zu verwendeten Technologien verwendet.

### Datenbank - erste Ideen

- Kunden
  - kundenid *Integer Autoincrement Primary Key*
  - name *VARCHAR(200)*
  - vorname *VARCHAR(200)*
  - strasse *VARCHAR(200)*
- benutzer
  - benutzerid *Integer Autoincrement Primary Key*
- dvds
  - dvdid *Integer Autoincrement Primary Key*
- genre
  - genreid *Integer Autoincrement Primary Key*
- artikel
  - artikelid *Integer Autoincrement Primary Key*
- verleih

- 
- verleihid *Integer Autoincrement Primary Key*
  - preis
    - preisid *Integer Autoincrement Primary Key*

## Prototyp

Bevor versuchen ein fertiges Produkt zu realisieren und daran vermutlich scheitern werden, haben wir beschlossen einen einfachen Prototyp zu programmieren. Dieser soll die wichtigsten Merkmale besitzen und zu Demonstrationszwecken dienen. Jedoch soll es auch möglich sein, diesen Prototypen zu einem fertigen Produkt fertig zu entwickeln. Der Prototyp soll also nicht quick & dirty programmiert werden.

Der Prototyp soll hauptsächlich die Kundenseite implementieren. D.h. er soll eine Webanwendung bereitstellen, bei der der Kunde bzw. Interessent sich registrieren kann und die Videothek benutzen kann. Dies bedeutet er kann sich die vorhanden Videos anschauen (Informationen zu diesen), kann sich die Verfügbarkeit anschauen, Video ausleihen, Rechnungen ansehen bzw. ausdrucken und eine Liste mit all seinen bisherigen Bestellungen anschauen.

Das Modul für die Verwaltung der DVD's ist in diesem Prototypen noch nicht vorgesehen.

Das Modul für das Versenden und Empfangen ist nur in einfacher Variante vorgesehen. Der Mitarbeiter der Onlinevideothek bekommt eine kleine Anwendung auf Konsolenbasis ohne grafische Oberfläche.

### 3.1 Datenbank

Zum Einsatz soll eine OpenSource Datenbank kommen. Gedanken an eine kommerzielle Datenbank kam aus Gründen der Lizenzkosten nicht auf.

Zu Auswahl standen mehrere OpenSource Datenbanken. MySql <sup>1</sup>, SAP DB <sup>2</sup>, HSQL DB <sup>3</sup> und Firebird <sup>4</sup>.

- **MySql**

- + sehr verbreitet
- + einige Erfahrung
- + gut Dokumentiert & große Community
- +
  - schlechtes Lizenzmodell
  - zu bekannt
  - keine Trigger
  - meist nur im privat bzw. klein Unternehmer Einsatz

- **SAP DB**

- +
  - schlechte Skalierbarkeit, da der Datenbank Speicherbereich im Vorfeld festgelegt werden muss
  - schlechte Erfahrung

- **HSQLDB**

- + reine JavaDatenbank
- + sehr klein
- + kann als reine Speicher Datenbank verwendet werden (Daten nur im Arbeitsspeicher)
- + kann als Applikations Datenbank verwendet werden (nur eine Applikation benutzt die Datenbank)

---

<sup>1</sup> <http://dev.mysql.com/downloads/mysql/4.0.html>

<sup>2</sup> <http://dev.mysql.com/downloads/maxdb/7.5.00.html>

<sup>3</sup> <http://hsqldb.sourceforge.net>

<sup>4</sup> <http://firebird.sourceforge.net>



### 3.1 Datenbank

---

- nicht für große Applikationen geeignet
- 

- **Firebird**

- + geringe Erfahrung durch Studium
- + sehr klein
- + gute grafische Tools
- + Original Sourcen kommen von Borland
- + Interbase Datenbank seit mehreren Jahren im Professionelle einsatz
- schlechtes Lizenzmodell
- 

Wir haben uns für die Firebird Datenbank entschieden, da es keine wirkliche Konkurrenz im Open Source Bereich gibt.

HSQL scheidet schon aus, weil es nicht für grosse Datenmengen geeignet ist. Bei der SAP DB muss der benötigte Speicherplatz der Datenbank vorher bekannt sein, was bei unserem Projekt nicht der Fall ist. MYSQL unterstützt keine Triggers und ist zu bekannt, d.h. MySql kann und sollte jeder Informatiker kennen und benutzt haben.

Firebird ist für uns relativ neu und die Erfahrungen die wir in der Vorlesung „Datenmanagment 2“ bekommen haben, war sehr positiv. Da diese Datenbank ursprünglich von Borland kommt, ist diese Datenbank auch nicht so neu, wie viele Denken.

Es soll aber schon am Anfang des Projektes bedacht werden, dass die Datenbank zu einem späteren Zeitpunkt eventuell mit einer professionelle Datenbank<sup>5</sup> ausgetauscht werden könnte. Deswegen muss schon am Anfang eine hohe Abstraktions-ebene vorhanden sein, so dass eventuelle Datenbankspezifische Elemente (Klassen) sehr einfach ausgetauscht werden können.

---

<sup>5</sup> z.B. DB2 von IBM

## 3.2 Versionsverwaltung

Eine Versionsverwaltung

<http://better-scm.berlios.de/comparison/comparison.html> [Wikipedia 2005, Versionsverwaltung]

### 3.2.1 Concurrent Versions System - CVS

### 3.2.2 Subversion

## 3.3 Entwicklungsumgebung

### 3.3.1 JBuilder

### 3.3.2 Netbeans

### 3.3.3 Eclipse

## **3.4 grafischen Benutzerschnittstellen in Java**

### **3.4.1 Abstract Window Toolkit - AWT**

### **3.4.2 Swing**

### **3.4.3 Standard Widget Toolkit - SWT**

### 3.5 Java-Web-Anwendungen

#### 3.5.1 Java Server Faces - JSF

#### 3.5.2 Struts

## 3.6 Persistenzschichten in Java

### 3.6.1 Java Data Objects

### 3.6.2 Hibernate

# Kapitel 4

## Implementierung

### 4.1 Versionsverwaltung mit Subversion

<sup>1</sup>

<http://www.phil-schneider.de> [Frotscher 2004c] Verweis auf eine Literaturquelle *hervorgehoben*

**Fett**

Typewriter

---

<sup>1</sup> So wird eine Fussnote gemacht



## 4.2 Entwicklungsumgebung mit Eclipse

## 4.3 grafischen Benutzerschnittstellen mit SWT

## 4.4 Java-Web-Anwendungen mit Struts

## 4.5 Persistenzschichten mit Hibernate

# Kapitel 5

## Zusammenfassung

Könnte von weiteren Studenten weitergeführt werden, da das gesamte Projekt in einzelne Module geteilt werden kann.

- suchmodul apache lucence - barcodemodul - lagerverteilung der Videos und kürzeste Wegstreckenberechnung - reports und statistik erstellung mit eclipse birt
- GUI (swt) tool für das Lager mit touchscreen

# Anhang A

## Protokoll vom 11. Mai 2004

Drei Frameworks stehen zur Auswahl

- Apache Cocoon
- Apache Struts
- Apache Tapestry

Jeder erstellt eine einfache (Web)Anwendung mit Hilfe eines dieser Frameworks folgende Komponenten sollen/muessen enthalten sein:

- einfache LoginSeite (über Datenbank)
- Liste aller Videos in Datenbank anzeigen
- EingabeMaske für neues Labor
- Validierung der Eingabedaten
- dynamische Navigation
- eventuell ein Bild für den Status der einzelnen Bilder (dynamisches Bild??)

Für diese BeispielAnwendung sollen möglichst viele Elemente des jeweiligen Framework verwendet werden. Wichtig ist dabei der Umgang und die Bedienbarkeit des Systems.

Wiederverwendbarkeit einzelner Module.

Design und Logik Trennung vorhanden? Kann das Design einfach/schnell ausgetauscht werden.

Es geht dabei nicht um ein 100Design spielt keine wichtige Rolle, es sollte jedoch beachtet werden, dass dieses später vom Kunden ausgetauscht werden möchte.

- 
- Struts: Stefan
  - Cocoon: Philipp
  - Tapestry: Remo
- 
- Namen für das Projekt finden
  - Link mit Beispiel Webseiten rumschicken

# Literaturverzeichnis

## **Frotscher 2004a**

FROTSCHER, Thilo: Der Ball ist rund. In: *Javamagazin* 04 (2004), S. 95

## **Frotscher 2004b**

FROTSCHER, Thilo: Das naechste Spiel ist immer das schwerste. In: *Javamagazin* 05 (2004), S. 89

## **Frotscher 2004c**

FROTSCHER, Thilo: Das Runde muss ins Eckige. In: *Javamagazin* 06 (2004), S. 60

## **Wikipedia 2005**

WIKIPEDIA: *Wikipedia - Enzyklopädie*. Version: 2005. <http://de.wikipedia.org>. – Online Ressource