

Inhaltsverzeichnis

1	Einführung	3
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	5
2	Konzepte und Aufbau	6
2.1	Aufbau	6
2.1.1	kurze Beschreibung	6
2.1.2	Kundenmodul	7
2.1.3	Lagermodul / Versandmodul	7
2.1.4	Verwaltungsmodul	7
2.2	Einsatzbereich	9
2.3	verschiedene Versionen	9
3	Technologien	10
3.1	Datenbank	10
3.2	Versionsverwaltung	13
3.2.1	Concurrent Versions System - CVS	13
3.2.2	Subversion	13
3.3	Entwicklungsumgebung	13
3.3.1	JBuilder	13
3.3.2	Netbeans	13
3.3.3	Eclipse	13
3.4	grafischen Benutzerschnittstellen in Java	13
3.4.1	Abstract Window Toolkit - AWT	13
3.4.2	Swing	13

INHALTSVERZEICHNIS

3.4.3	Standard Widget Toolkit - SWT	13
3.5	Java-Web-Anwendungen	13
3.5.1	Java Server Faces - JSF	13
3.5.2	Struts	13
3.6	Persistenzschichten in Java	13
3.6.1	Java Data Objects	13
3.6.2	Hibernate	13
4	Implementierung	16
4.1	Versionsverwaltung mit Subversion	16
4.2	Entwicklungsumgebung mit Eclipse	16
4.3	grafischen Benutzerschnittstellen mit SWT	16
4.4	Java-Web-Anwendungen mit Struts	16
4.5	Persistenzschichten mit Hibernate	16
5	Zusammenfassung	17
A	Protokoll vom 11. Mai 2004	18
	Literaturverzeichnis	20

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 Teamprojekten¹ *Labmin*² und *German Team Sony Aibo*³ 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 JavaMagazin⁴ in mehreren Ausgaben aufgegriffen und verschiedene Ansatzmöglichkeiten diskutiert. Die Idee unseres Tippspiel war dabei eine Webanwendung mit Datenbankanbindung. 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

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

² <http://labmin.de.vu>

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

⁴ [Frotscher 2004a, b, c]

wie bspw Amango⁵, Netleih⁶, Invdeo⁷ und Verleihshop⁸. 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

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.

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

Kapitel 2

Konzepte und Aufbau

2.1 Aufbau

2.1.1 kurze Beschreibung

Das System soll eine komplette Videothek ersetzen. Die Videothek benötigt dann nur noch ein (grosses) Lager für die zu verleihenden Videos und einige Mitarbeiter für den Versand und die Verwaltung. Die Software soll aus drei Modulen bestehen. Das **Kundenmodul** ist die Internetpräsenz der Videothek. Auf dieser kann der Kunde die vorhandenen Videos durchstöbern und nach Anmeldung deren Verfügbarkeit kontrollieren und auf Wunsch einige Video ausleihen. Dabei kann ein angemeldeter Kunde zusätzlich seine Bestellungen und Rechnungen betrachten. Das **Lagermodul** stellt die Software für das Lager und den Versand dar. Mit deren Hilfe kann ein Mitarbeiter der Videothek Videos für den Versand vorbereiten. D.h. er bekommt eine Liste mit Bestellungen von Kunden und arbeitet diese ab. Dabei arbeitet er viel mit Barcodes und einem Barcodescanner. Mit dessen Hilfe markiert er Videos für einen bestimmten Kunden und eine bestimmte Bestellung und versendet diese. Dem System wird somit mitgeteilt, dass KundeA jetzt das Video1 für x Tage ausgeliehen hat und dieses Video am Montag dem 1.1.2004 dem Versandunternehmen übergeben wurde. Rechnungen, Versandetiketten und eventuelle Lieferscheine werden dabei automatisch mit ausgedruckt. Zusätzlich bietet das Lagermodul die Möglichkeit die zurückgekommenen Videos des 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 der Verwaltung bei der Organisation der Videothek. Es können Kundendaten und Rechnungen betrachtet und ggf. gedruckt/exportiert werden, das Videosortiment bearbeitet werden und inhaltliche Änderungen (z.B. Angebote) an dem Kundenmodul vorgenommen werden. Weiterhin besteht die Möglichkeit ausführ-

2.1 Aufbau

liche Reports & Statistiken zu betrachten/exportieren.

Abbildung 2.1 auf Seite 8 stellt dies in einfacher Weise grafisch dar.

2.1.2 Kundenmodul

Die Kundensoftware ist eine Webapplikation. Auf dieser „Webseite“ können sich sowohl Interessenten als auch Kunden über Videos informieren. Nach Registration kann man sich auch den Status von einzelnen Videos anschauen und verleihbare ausleihen. Neben Statistiken stehen dem Kunden auch ein Navigationspunkt mit seinen bisherigen Bestellungen und dem aktuellen Stand seiner Wunschliste zur Verfügung.

2.1.3 Lagermodul / Versandmodul

Ein Mitarbeiter der Videothek hat dann die Möglichkeit über eine eigene Software die gewünschten Videos zum Versand vorzubereiten. Dabei wird er so weit vom System unterstützt, dass er nur noch die fehlenden Roboterarme ersetzen muss. Mit Hilfe eines Barcodescanners kann der Mitarbeiter, auf dem bereits fertig ausgedrucktem Bestellschein, die Kundennummer und die gewünschten Videos erfassen. Der Barcodescanner erleichtert dabei nur die Eingabe von Kundennummern, Rechnungsnummern und Artikelnummern. Die Aufgaben des Mitarbeiters für eine Bestellung können wie folgt zusammengefasst werden:

- Bestellschein vom Drucker entnehmen
- Bestellnummer vom Bestellschein dem System mitteilen
- gewünschte Artikel aus dem Sortiment nehmen
- Artikelnummern einzeln dem System mitteilen *dadurch wird dem System mitgeteilt, daß diese Artikel verliehen sind*
- Artikel in einen Versandumschlag packen und ggf. Versandaufkleber vom Bestellschein entnehmen

2.1.4 Verwaltungsmodul

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.

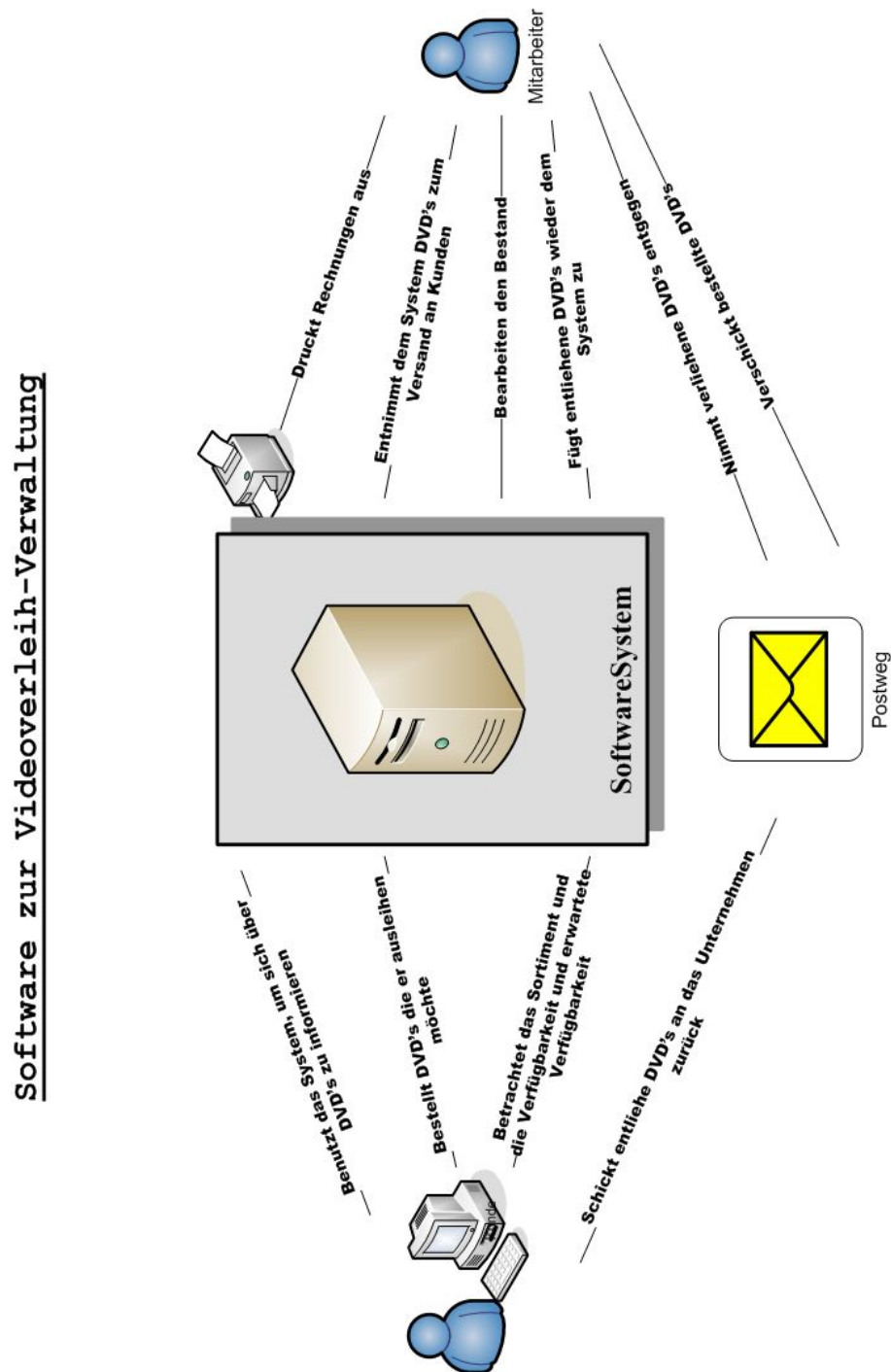


Abbildung 2.1: Erste Gedanken zu der Videosoftware

2.2 Einsatzbereich

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.

2.3 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
 -
- 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.

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 ¹, SAP DB ², HSQL DB ³ und Firebird ⁴.

- **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**

-
- ¹ <http://dev.mysql.com/downloads/mysql/4.0.html>
 - ² <http://dev.mysql.com/downloads/maxdb/7.5.00.html>
 - ³ <http://hsqldb.sourceforge.net>
 - ⁴ <http://firebird.sourceforge.net>

3.1 Datenbank

- +
- + Datenbank seit mehreren Jahren bei SAP im Einsatz
- 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)
- 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

3.1 Datenbank

Denken.

Es soll aber schon am Anfang des Projektes bedacht werden, dass die Datenbank zu einem späteren Zeitpunkt eventuell mit einer professionellen Datenbank⁵ ausgetauscht werden könnte. Deswegen muss schon am Anfang eine hohe Abstraktionsebene vorhanden sein, so dass eventuelle Datenbankspezifische Elemente (Klassen) sehr einfach ausgetauscht werden können.

⁵ z.B. DB2 von IBM

3.2 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

Datenbank - erste Ideen

- Kunden
 - kundenid *Integer Autoincrement Primary Key*
 - name *VARCHAR(200)*
 - vorname *VARCHAR(200)*
 - strasse *VARCHAR(200)*

3.6 Persistenzschichten in Java

- 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 vorge-

3.6 Persistenzschichten in Java

sehen. Der Mitarbeiter der Onlinevideothek bekommt eine kleine Anwendung auf Konsolenbasis ohne grafische Oberfläche.

Kapitel 4

Implementierung

4.1 Versionsverwaltung mit Subversion

¹

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

Fett

Typewriter

4.2 Entwicklungsumgebung mit Eclipse

4.3 grafischen Benutzerschnittstellen mit SWT

4.4 Java-Web-Anwendungen mit Struts

4.5 Persistenzschichten mit Hibernate

¹ So wird eine Fussnote gemacht

Kapitel 5

Zusammenfassung

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