

# **Projektdokumentation**

## **BookPoint für Microsoft Office SharePoint 2007**

Bericht zur betrieblichen Projektarbeit von

**Robert Mühsig**

zur Erlangung des Abschlusses

**als Fachinformatiker**

**Fachrichtung**

**Anwendungsentwicklung**

Betrieb: **T-Systems Multimedia Solutions GmbH**

Riesaer Straße 5

01129 Dresden

Betreuer: **Dieter Bauckmeier**

Ausführungszeitraum: 5. – 19. April 2007

letzte Änderungen: 19. April 2007

# Inhaltsverzeichnis

1.	Erläuterung des Projektauftrages .....	1
1.1	Projektumfeld.....	1
1.2	Aufgabenstellung .....	1
1.3	Auftraggeber und Ansprechpartner.....	2
1.4	Änderung gegenüber dem Projektantrag.....	2
2.	Erläuterung des Projektauftrages .....	2
2.1	Problemanalyse .....	2
2.1.1	Ist-Analyse.....	2
2.1.2	Soll-Konzept.....	3
2.2	Ressourcenplanung .....	3
2.2.1	Zeitliche Projektplanung.....	3
2.2.2	Kosten und Preise .....	3
2.3	Funktionsanalyse.....	4
2.3.1	Allgemeine Leistungsmerkmale .....	4
2.3.2	Informationsspeicherung im SharePoint.....	4
3.	Projektdurchführung.....	5
3.1	Wahl der Art des Clientprogramms .....	5
3.2	Entwicklungsumgebung.....	5
3.3	Realisierung .....	6
3.3.1	SharePoint Seiten- und Listenkonfiguration.....	6
3.3.2	Softwareentwurf .....	6
3.3.2.1	Backend.....	7
3.3.2.1.1	SharePoint-Anbindung.....	7
3.3.2.1.2	Amazon-Anbindung.....	8
3.3.2.2	Testanwendung.....	9
3.3.2.3	Frontend .....	9
4.	Testphase .....	11
5.	Projektabschluss .....	11
5.1	Erfüllung des Projektziels .....	11
5.2	Mögliche Verbesserungen.....	12
5.3	Aufgetretene Probleme .....	12
6.	Quellenangabe .....	13
7.	Anlagenverzeichnis.....	14

## **1. Erläuterung des Projektauftrages**

### **1.1 Projektumfeld**

Das Projekt wurde im Projektfeld 33 des Bereiches (Business Unit – BU) „Tele-Applications & Services“ der Multimedia Solutions GmbH realisiert, dieses Projektfeld ist für Portallösungen verantwortlich.

Die Multimedia Solutions GmbH (MMS) ist ein Tochterunternehmen der T-Systems und bietet ihren Kunden multimediale Softwarelösungen, um Prozessabläufe zu optimieren oder neue Geschäftsmodelle im Internet zu entwickeln, darunter sind eCommerce-, Portal- und andere webbasierende Lösungen.

Um die Verwaltung und die tägliche Arbeit zu vereinfachen, wird bereits in der MMS intern eine Portallösung auf Basis von Microsoft SharePoint Portal Server 2003 eingesetzt.

Die webbasierende Portallösung von Microsoft integriert sich vollständig in die IT-Infrastruktur. Des Weiteren bieten die Microsoft Office Programme komfortable Möglichkeiten, um Daten und Dokumente auf dem SharePoint Portal Server zu bearbeiten. Neben einer umfangreichen Dateiverwaltung verfügt der SharePoint auch über andere Funktionen, wie z.B. ein CMS (Content Management System).

Ende letzten Jahres hat Microsoft die neue Version des SharePoint veröffentlicht: Microsoft Office SharePoint Server 2007. Die neue Version bringt neben neuen Funktionen zusätzlich auch mehr Möglichkeiten für das Erstellen eigener Anwendungen mit.

### **1.2 Aufgabenstellung**

Um Kunden die Möglichkeiten des SharePoint 2007 zu zeigen, soll ein Prototyp erstellt werden, welcher insbesondere auf die Möglichkeit eingeht, weitere Anwendungen mit dem SharePoint zu verbinden. Dabei sollen zugleich neuste „Web 2.0“ Features genutzt werden.

Damit die Anwendung auch einen Mehrwert für die MMS bietet, wurde das Projekt aufgrund folgender Situation erstellt:

Wie in vielen Software-Unternehmen, gibt es auch in der MMS eine große Anzahl an Fachbüchern. Diese werden in der MMS nicht zentral gelagert, sondern befinden sich beim jeweiligen Mitarbeiter am Arbeitsplatz. Damit eine gewisse Ordnung gewahrt bleibt, ist momentan auf dem SharePoint Portal Server 2003 eine Liste eingerichtet. In dieser wird erfasst, welches Buch ein Mitarbeiter hat. Leider ist die Handhabung und Pflege dieser Liste sehr umständlich, daher soll nun eine einfachere Lösung geschaffen werden, um die dezentralen Bücherbestände auf dem SharePoint zu bearbeiten. Da eine Migration vom alten SharePoint auf die neue Version geplant ist, soll die Anwendung auf Basis von Microsoft Office SharePoint 2007 erstellt werden.

### **1.3 Auftraggeber und Ansprechpartner**

Da dieses Projekt intern umgesetzt wird, ist der Auftraggeber auch gleichzeitig der Ansprechpartner für das Projekt. Während der Projektdurchführung ist Dieter Bauckmeier der Ansprechpartner und übernimmt auch die Projektabnahme.

### **1.4 Änderung gegenüber dem Projektantrag**

Gegenüber dem Projektantrag wurden durch ein Gespräch mit dem Ansprechpartner einige Änderungen vorgenommen.

Der auf dem Projektantrag geschriebene Projektname „Librarysolution for Microsoft SharePoint 2007“ wurde geändert. Diese Umbenennung hatte mehrere Gründe. Die offizielle Bezeichnung des SharePoints lautet „Microsoft Office SharePoint 2007“ und nicht nur „Microsoft SharePoint 2007“. Damit der Applikationsname später auch einen engeren Bezug zum SharePoint hat und das Wort „Solution“ im SharePoint-Bereich einer anderen Bedeutung zugeordnet ist, wurde der Projektname in „BookPoint für Microsoft Office SharePoint 2007“ geändert.

Der Punkt, dass nur der Besitzer des Buches Buchinformationen ändern kann, wurde im Projekt nicht umgesetzt. Grund hierfür war, dass ein komplexes Rollenkonzept für die komfortable Administration erforderlich wäre und der Fokus des Projektes auf die Integration der List-Webservices lag. Da außerdem viele Mitarbeiter nur wenig Zeit haben, ihre eigenen Bücher einzupflegen, ist es sinnvoller, dass auch andere Personen, wie z.B. Assistenten, diese Bücher in das System eintragen können. Aus den oben genannten Gründen wurde der Punkt nicht realisiert.

## **2. Erläuterung des Projektauftrages**

### **2.1 Problemanalyse**

#### **2.1.1 Ist-Analyse**

Die Liste auf dem SharePoint ist nur sehr umständlich zu erreichen, da die Navigation und Seitengestaltung sehr komplex gestaltet ist. Das Hinzufügen neuer Bücher ist auch mit einem bestimmten Zeitaufwand verbunden, da alle Daten, wie ISBN, Beschreibung, Autor und Titel, händisch eingegeben werden müssen.

Durch diese Mängel ist die Akzeptanz der Bücherliste zurückgegangen, sodass viele Mitarbeiter ihre Bücherbestände nicht in den SharePoint einpflegen.

Somit gibt es keine aktuelle und zentrale Stelle, wo aufgelistet ist, wer welches Buch hat. Es muss mit vielen Leuten Kontakt gehalten werden, damit es nicht zu Doppelkäufen von neuen Büchern kommt.

### **2.1.2 Soll-Konzept**

Es soll eine einfach zu bedienende und schnelle Clientanwendung erstellt werden, welche einen guten Überblick über die Bücherbestände geben soll.

Die Bücherbestände werden nach wie vor in einer Liste auf dem SharePoint gespeichert, dabei soll die neue Version des SharePoints genutzt werden.

Um das Eintragen zu vereinfachen sollen die Webservices von Amazon genutzt werden, da Amazon sehr viele Bücher listet. Durch Eingabe der ISBN sollen die wichtigsten Buchinformationen, wie Titel, Beschreibung und Autor, abgerufen und in die Eingabemaske geschrieben werden.

Im Pflichtenheft (Anlage III) werden unter anderem diese Angaben konkretisiert.

## **2.2 Ressourcenplanung**

### **2.2.1 Zeitliche Projektplanung**

Die Zeitplanung ist als Balkendiagramm nach Gantt aufgestellt. Termine und Zeiträume für die einzelnen Phasen können aus dem Diagramm entnommen werden.

Die Zeitplanung ist in der Anlage IV dargestellt.

### **2.2.2 Kosten und Preise**

Die Hardware und Software, welche für die Realisierung nötig waren, wurden von der Multimedia Solutions GmbH gestellt, sodass keine Kostenaufstellung möglich ist.

Eine Preisaufstellung ist ebenfalls nicht möglich, da dieser Prototyp nur intern Verwendung findet.

## 2.3 Funktionsanalyse

### 2.3.1 Allgemeine Leistungsmerkmale

Aufbauend auf dem Soll-Konzept der Anwendung, werden nun die allgemeinen Funktionsmerkmale beschrieben:

- Anbindung an den SharePoint:
  - o Es sollen sich nur autorisierte Nutzer über die Anwendung am SharePoint anmelden können, dazu werden die Login-Daten vom SharePoint genutzt.
  - o Die Daten werden in einer gewöhnlichen SharePoint Liste auch ohne die Anwendung durch einen normalen Browser abrufbar und lesbar sein.
- Integration der Amazon Webservices:
  - o Um die Eingabe der Bücher zu erleichtern, werden die Amazon Webservices eingebunden.

Diese Angaben werden unter anderem im Pflichtenheft (Anlage III) konkretisiert.

### 2.3.2 Informationsspeicherung im SharePoint

Da der Microsoft Office SharePoint Server 2007 als Datenspeicher dient, werden die Buchinformationen in einer Liste, ähnlich einer herkömmlichen Excel Liste, gespeichert.

Um die Daten auch ohne die Anwendung lesbar zu machen, wurde folgende Struktur erdacht:

- Titel: Buchtitel
- Untertitel: Buchuntertitel
- Beschreibung: Beschreibungstext für das Buch
- Kommentar: Ein Kommentarfeld für evtl. Bemerkungen zum Buch
- Autor: Buchautor, dabei sollte nur der „Hauptautor“ benannt werden
- Kategorie: Eine frei wählbare Kategorie für die Bücher, um diese später besser zu organisieren
- Bewertung: Auswahloption für Buchbewertung („Empfehlenswert“/„Normal“/„Schlecht“)
- Sprache: Auswahloption für die Sprache des Buches („deutsch“/„englisch“)
- Anzahl: Buchanzahl, standardmäßig ist die Anzahl auf eins, kann aber bei mehreren Exemplaren geändert werden
- Ansprechpartner: Die Person, welche das Buch besitzt bzw. es verwahrt.

### 3. Projektdurchführung

#### 3.1 Wahl der Art des Clientprogramms

Da für die Nutzer eine schnelle Anwendung geschaffen werden soll, welche leicht erreichbar und die Informationen auf einfache Weise darstellen soll, fiel die Wahl auf eine Windows-Anwendung.

Als Alternative für die Windows-Anwendung gab es noch eine direkte Webanwendung im SharePoint sowie eine reine Webanwendung.

Der Vorteil einer Windows-Anwendung ist, dass die Entwicklungszeit für eine komfortable grafische Benutzeroberfläche wesentlich kürzer und deren Erstellung einfacher ist, als bei einer Webanwendung. Neben der Entwicklungszeit wurde auch die Usability in Betracht gezogen. Heutige Webanwendungen sind dank neuer Techniken zwar Clientanwendungen ähnlich, aber reagieren langsamer, sodass der Nutzer immer eine bestimmte Wartezeit hat.

Da es auch nicht gewünscht wurde, einen weiteren Webserver für eine Webanwendung bereit zu stellen und eine direkte Integration in den SharePoint äußerst umständlich und zeitaufwendig gewesen wäre, fiel die Wahl auf eine Windows-Anwendung, welche ohne Installation auf jedem Windows-PC mit .NET Framework 2.0 lauffähig ist.

#### 3.2 Entwicklungsumgebung

Das Projekt wurde auf einem Microsoft Virtual PC Image entwickelt, da in der MMS der SharePoint Office Server 2007 erst in Planung ist. Auch für eine evtl. Präsentation bei einem Kunden kann dieses Image schnell gezeigt werden.

Das Image lag bereits fertig konfiguriert von einer Microsoft Schulung vor. Auf dem virtuellen PC ist ein Microsoft Windows Server 2003 Enterprise Edition als Betriebssystem installiert.

Neben dem aktivierten IIS 6.0 als Webserver wird ein Microsoft SQL Server 2005 als Datenbank für den SharePoint genutzt. Bei der Schulung wurde eine englische Testversion des SharePoint Servers 2007 installiert. Für den Betrieb des SharePoint Servers musste auch das Microsoft .NET Framework 3.0 installiert sein.

Da die Installation der Standardinstallationsroutine entspricht und der SharePoint bereits fertig konfiguriert vorlag, wird in der Dokumentation nicht näher auf den eigentlichen Server eingegangen.

Neben dem SharePoint wurde auf dem virtuellen PC auch ein Visual Studio 2005 Professional Edition installiert, mit welcher die Windows-Anwendung erstellt wurde.

Als Host für den virtuellen PC dient ein Windows – XP PC mit 3 GHz und 2 GB RAM, wobei dem virtuellen PC 1 GB Arbeitsspeicher zugewiesen wurde.

### 3.3 Realisierung

#### 3.3.1 SharePoint Seiten- und Listenkonfiguration

Damit die Anwendung später mit verschiedenen Nutzern getestet werden konnte, wurde neben dem „Administrator“ Account noch zwei weitere Accounts auf dem SharePoint eingerichtet: RMU & STST. Die Accountnamen sind Mitarbeiterkürzel der MMS, wobei RMU mein eigenes Kürzel ist und STST für eine Kollegin steht.

Für das Projekt wurde ein neuer Arbeitsbereich, eine „*Sitecollection*“, mit dem Namen „**Abschlussprojekt**“ auf den SharePoint angelegt. Diese „*Sitecollection*“ ist eine Art Container für weitere einfache Seiten oder Formulare.

Danach musste noch unter dem „Abschlussprojekt“ eine einzelne Seite „**BookPoint**“ hinzugefügt werden, weil die hierarchische Struktur des SharePoint eine Liste nur an einer einzelnen Seite erlaubt.

Diese Hierarchie kann man auch gut in der URL des Projektes nachvollziehen:

***http://wum56083/abschlussprojekt/bookpoint/***

*Server Sitecollection Site*

Es wurde eine neue Liste „Buchliste“ angelegt und alle Spalten wurden, wie in Punkt 2.3.2 beschrieben, erstellt. Dabei wurden noch alle Spalten der „*Default View*“, also der Standardansicht, hinzugefügt. Der Webservice holt immer nur Daten von einem bestimmten „*View*“ ab, daher ist es wichtig, alle Daten in einem „*View*“ zugänglich zu machen.

Diese Informationen und Hintergrundinformationen zum SharePoint finden Sie in der technischen Dokumentation (Anlage II). Weitere Einstellungen wurden am SharePoint nicht vorgenommen.

#### 3.3.2 Softwareentwurf

Das Projekt wurde mit Nutzung des Microsoft .NET Framework 2.0 und der Programmiersprache C# erstellt, somit konnten verschiedene Programmteile sehr einfach realisiert werden.

Um eine flexible Lösung zu schaffen, die auch möglicherweise in späteren Projekten eingebunden werden kann, wurde die Windows-Anwendung in zwei Teile gegliedert.

Für das Testen der Funktionalität wurde noch eine Testanwendung erstellt.

Genauere Informationen über den Softwareentwurf ist in der technischen Dokumentation (Anlage II) beschrieben.



### 3.3.2.1 Backend

Damit das Backend auch für andere .NET Projekte genommen werden kann, wurde ein Klassenbibliotheksprojekt in Visual Studio 2005 angelegt. Nach dem Kompilieren einer Klassenbibliothek generiert Visual Studio eine dll-Datei, welche z.B. auch in ASP.NET Projekten genutzt werden kann.

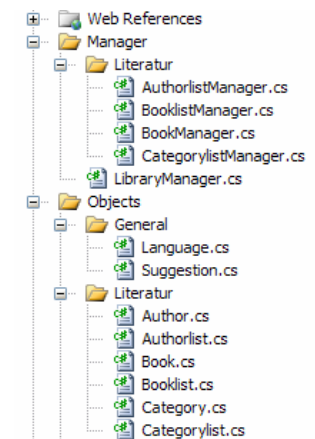
Projektname: ***MMS.BookPoint.Definitions***

Dateiname: ***MMS.BookPoint.Definitions.dll***

*Muster-Projektname von Microsoft: Firma.Produktname.Produktteil*

Innerhalb des Backends wurde noch eine Gliederung vorgenommen. Um einen einfachen Zugriff später auf die Listeneinträge zu ermöglichen, wurden unter dem Namespace ***MMS.BookPoint.Definitions.Manager*** die eigentlichen Zugriffsklassen erstellt.

Für die Objekthaltung wurde der Namespace ***MMS.BookPoint.Definitions.Objects*** angelegt. In diesem Namespace befinden sich die einzelnen Klassen, welche die Daten halten und zwischenspeichern. Die eigentliche Logik, wie das Befüllen und das Verändern der Objekte, übernehmen die Manager.



*Projektstruktur des Backends*

Der „*LibraryManager*“ ist für die Authentifikation und für den Verbindungsaufbau und die Speicherung von den Verbindungsdaten zuständig. Die anderen Manager, wie z.B. der „*BookManager*“ greifen über den „*LibraryManager*“ auf den SharePoint zu und führen die jeweilige Operation aus.

In den folgenden Punkten werden die Zugriffe auf die Webservices behandelt. Eine genauere Beschreibung über die einzelnen Objekte und Manager befindet sich in der Anlage II.

#### 3.3.2.1.1 SharePoint-Anbindung

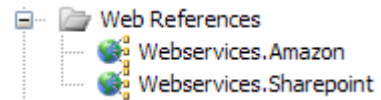
SharePoint verfügt über mehrer Webservices, welche für die Anbindung weiterer Anwendungen verwendet werden können.

Die URL für das Einfügen des List-Webservice war folgende:

***http://wum56083/abschlussprojekt/bookpoint/\_vti\_bin/Lists.asmx***

*Server Sitecollection Site Webservice List-Webservice*

Der Webservice wurde unter dem Namespace „*MMS.BookPoint.Definitions.Webservices.SharePoint*“ als “Web References” in Visual Studio hinzugefügt. In der Abbildung sind bereits alle beiden Webservices integriert.



*Integrierte Webservices*

Im nächsten Punkt wird beschrieben, wie der Amazon Webservice eingebunden wurde.

Nach dem Einfügen des Webservice wurden alle Methoden und Klassen, wie in der WSDL des Webservice beschrieben, automatisch erstellt und unter dem Namespace „*MMS.BookPoint.Definitions.Webservices.Sharepoint*“ verfügbar gemacht.

Nach dem Hinzufügen wurden die jeweiligen Klassen und Methoden in den einzelnen Managern eingebunden.

Auf die genauen Parameter wird in der technischen Dokumentation (Anlage II) eingegangen.

### **3.3.2.1.2 Amazon-Anbindung**

Für die Nutzung des Amazon Webservice, wurde ein Developer Account unter „<http://www.amazon.com/webservices>“ angelegt. Mit diesem Developer Account ist es möglich, den Webservice kostenfrei zu nutzen.

Um die Methoden anzuwenden, wurde der Webservice über Visual Studio dem Projekt hinzugefügt. Die Adresse für die WSDL befindet sich unter dieser URL:

**<http://webservices.amazon.com/AWSECommerceService/DE/AWSECommerceService.wsdl>**

Wie auch beim SharePoint Webservice, wurden automatisch nach dem Einfügen des Amazon-Webservice die Klassen und Methoden unter dem Namespace „*MMS.BookPoint.Definitions.Webservice.Amazon*“ verfügbar gemacht.

Aus zeitlichen Gründen wurde kein „*AmazonManager*“ implementiert. Da die Funktion des SharePoints im Frontend nur beim Eintragen und Editieren zum Tragen kommt, bleibt trotzdem eine Übersicht erhalten. Die Webservice Methode zum Verbinden zu Amazon.de und zum Abrufen von Buchdaten wurde direkt in der Frontend Funktion implementiert.

Die genauen Parameter werden in der technischen Dokumentation näher erläutert.

### 3.3.2.2 Testanwendung

Die Klassenbibliothek wurde in einer Testanwendung getestet.

Dazu wurde im Visual Studio ein neues Konsolenprojekt „*MMS.BookPoint.TestApp*“ angelegt.

Projektname: *MMS.BookPoint.TestApp*

Dateiname: *MMS.BookPoint.TestApp.exe*

In dieser Anwendung wurde die „*MMS.BookPoint.Definitions.dll*“ hinzugefügt.

Es wurden während der Entwicklungsphase in der Testapplikation geprüft, ob die Verbindung zu den einzelnen Webservices hergestellt werden konnte und ob das Abrufen und das Editieren von den SharePoint-Daten funktionierte.

### 3.3.2.3 Frontend

Das Frontend ist eine Windows-Forms Applikation. Die Windows-Forms wurden mit dem .NET Framework eingeführt und ersetzt die alten MFC Bibliothek.

Dafür wurde ein neues Windows-Forms Projekt in Visual Studio 2005 erstellt.

Projektname: *MMS.BookPoint.WinApp*

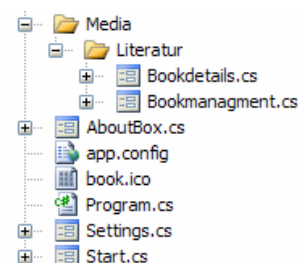
Dateiname: *MMS.BookPoint.WinApp.exe*

Wie in der Testapplikation wurde auch in diesem Projekt das Backend hinzugefügt.

Die „*MMS.BookPoint.Definitions.dll*“ wurde dem Projekt hinzugefügt und unter dem Namespace „*MMS.BookPoint.Definitions*“ konnten nun alle Namespaces, Klassen und Webservices des Backends genutzt werden.

Das Frontend besteht aus vier Hauptkomponenten:

- Loginfenster (*Settings.cs*)
- Übersichtsfenster (*Start.cs*)
- Anlegen und Bearbeiten der Bücher (*Bookmanagment.cs*)
- Detailsansicht für die Bücher (*Bookdetails.cs*)



Strukturübersicht Frontend

Die genauen Benutzeraktionen werden in der Kundendokumentation (Anlage I) gezeigt. Technische Details können der Anlage II entnommen werden.

Im Login-Fenster (*Settings.cs*) wurden Eingabefelder für den Benutzernamen und das Passwort erstellt. Da es mehrere Buchlisten auf den SharePoint geben kann, wurde noch ein Eingabefeld für die SharePoint URL hinzugefügt. Der „*LibraryManager*“ aus dem Backend wird nun eingebunden. Weiterhin wird der Verbindungsaufbau mit den angegebenen Daten geprüft.

Die Buchdaten werden nach dem erfolgreichen Login im Übersichtsfenster (*Start.cs*) geladen. Für das Darstellen der Daten wurde ein „*Datagridview*“ erstellt. Damit die Übersichtlichkeit nicht verloren geht, wird in dem „*Datagridview*“ nur die Titel und Untertitel der Bücher gezeigt, sowie zwei Buttons für das Bearbeiten (*Bookmanagment.cs*) und für die Detailansicht (*Details.cs*). Das „*Datagridview*“ wird über den „*BooklistManager*“ befüllt, dieser ruft alle Bücher über seine „*Load*“-Methode aus dem SharePoint ab.

Neben dem „*Datagridview*“ wurde noch Platz für andere Optionsmenüs gelassen. In diesem Optionsmenü wurde der Button „*Neues Buch anlegen*“ (*Bookmanagment.cs*) erstellt.

Damit später auch andere Medien, wie DVDs o.Ä., in dem Programm verwaltet werden können, wurden ein „*TabControl*“ genutzt. Für die Bücher wurde ein Tab (oder auch Reiter) „*Bücher*“ angelegt.

In der Detailansicht (*Details.cs*) wurden alle Buchinformationen ohne Editierungsfunktion dargestellt. Dabei wurde für jede Information ein entsprechendes „*Label*“ oder eine „*Textbox*“ erstellt.

Das Anlegen neuer Bücher und das Editieren von bestehenden Buchdaten erfolgt im „*Bookmanagment*“-Fenster. Es können dort alle Informationen eingegeben und später wieder bearbeitet werden. Für die Informationen wurden mehrere „*Textbox*“-Elemente und Auswahlelemente („*Combobox*“) erstellt. Für eine einfache Auswahl des Autors und der Kategorie wurden diese Auswahlelemente eingefügt. Bestehende Kategorien und Autoren können somit leicht ausgewählt werden. Beim Bearbeiten eines bereits bestehenden Buches werden alle Angaben bereits in die entsprechenden Eingabefelder geschrieben und die anderen Auswahlfelder, wie z.B. der Autor, vorselektiert.

Damit alle benötigten Daten eingegeben werden müssen, wurde noch eine Fehlerüberprüfung eingebaut.

Die Icons für das Projekt stammen aus einer freien Icon Sammlung von der MMS. Die Icons haben verschiedene Quellen, sind aber völlig frei verwendbar und auch in kommerziellen Bereich nutzbar.

Nach dem Erstellen des Designs der einzelnen Fenster wurde das Backend entsprechend eingebunden. Eine genauere Integrationsbeschreibung wurde in der technischen Dokumentation (Anlage II) angefügt.

## 4. Testphase

In der ersten Testphase wurde die Anbindung an den SharePoint geprüft.

Um das Abrufen von den Einträgen in der SharePoint Liste zu prüfen, wurden zuerst mehrere Einträge direkt auf dem SharePoint gemacht.

Nach der erfolgreichen Anmeldung über die Software sollte man alle eingetragenen Bücher sehen. Um das Bearbeiten der Bücher zu testen, veränderte ich mehrere Bücherdaten. Das Löschen wurde getestet, indem ich absichtlich „fehlerhafte“ Daten in das System eintrug und diese wieder löschte. Zwischen den einzelnen Schritten wurde immer wieder geprüft, ob die Änderungen im SharePoint übernommen wurden.

Die zweite Testphase umfasste die Anbindung an Amazon. Bei Eingabe einer ISBN soll auf Amazon.de nach dem Buchinformationen gesucht werden.

Es wurden mehrere gültiger ISBNs eingegeben und geschaut, ob die Ergebnisse mit den tatsächlichen Buchinformationen übereinstimmen.

Bei Eingabe einer ungültigen ISBN öffnete sich eine Fehlermeldung, dass das Buch mit der angegebenen ISBN nicht gefunden wurde.

Nach Ablauf der Testphase komme ich zu folgendem Schluss:

Die Anwendung läuft bei einer korrekt erstellen SharePoint Liste ohne kritische Fehler. Auf andere kleinere Fehler wird der Nutzer mit einer Fehlermeldung hingewiesen und die Anwendung läuft stabil weiter. Der Amazon.de Webservice wurde korrekt implementiert und liefert bei einer gültigen ISBN die entsprechenden Daten aus der Amazon.de-Datenbank.

## 5. Projektabschluss

### 5.1 Erfüllung des Projektziels

Die Entwicklung eines schnellen und komfortablen Clients mit Amazon- und SharePoint-Anbindung ist erfolgreich durchgeführt und somit ist das Projektziel voll erfüllt. Es wurde eine Lösung geschaffen, mit welcher die Bearbeitung der SharePoint-Liste in einer besseren Form möglich ist.

Es wurde eine erweiterbare und flexible Lösung geschaffen, mit welcher das Backend ebenso in anderen Projekten genutzt werden kann.

Neben der Kundendokumentation (Anlage I) liegt eine technische Dokumentation (Anlage II), das Pflichtenheft (Anlage III) und der Zeitplan (Anlage IV) bereit.

## 5.2 Mögliche Verbesserungen

Damit die Einträge auch ohne die Anwendung lesbar sind, ist die Liste, anderes als es in einem Datenbankentwurf realisiert werden sollte, sehr einfach gehalten.

Wenn zwei Bücher, welche vom gleichen Autor geschrieben wurden, in die Liste eingetragen wurden, dann wird auch der Autor doppelt gespeichert. Derselbe Effekt kommt auch bei den Kategorien zum Tragen.

Dies könnte verbessert werden, indem man mehrere „Listen“ anlegt und diese miteinander verknüpft. Dies wäre dann ähnlich wie in einem Datenbankentwurf nach der dritten Normalform.

## 5.3 Aufgetretene Probleme

Laut den Angaben in der Planung, sollte der Buchtitel in der SharePoint Liste unter dem Spaltennamen „Titel“ gespeichert werden. Da allerdings eine englische SharePoint Version zur Verfügung steht ist das „Title“ Attribut zwingend erforderlich und darf auch nicht geändert werden. Der Buchtitel wurde in der Spalte „Title“ gespeichert, da ansonsten der Webservice einen Fehler verursacht.

Eine mögliche Lösung für das Problem ist, die deutsche Version des SharePoint zu nutzen.

Das SDK (*Software Development Kit*) des Microsoft Office SharePoint 2007 wird noch ständig erweitert, da das Produkt erst seit kurzem auf dem Markt ist. Dadurch mussten einige Funktionen länger als geplant getestet werden, bis die richtige Anwendung herausgefunden wurde.

## 6. Quellenangabe

### Bücher

Charles Petzold: **Windows Forms-Programmierung mit Visual C sharp 2005**, Der schnelle und kompakte Leitfaden zur Entwicklung von Windows Forms mit C#; Microsoft Press Deutschland; 2006

Erich Gamma et Al.: **Entwurfsmuster**, Elemente wiederverwendbarer Software; Addison Wesley Verlag; 2004

### Internet

Microsoft: <http://msdn.microsoft.com/vcsharp/>, Microsoft Developer Network (MSDN); 2007

Microsoft: <http://msdn2.microsoft.com/en-us/library/ms441339.aspx>, MSDN Windows SharePoint Services 3.0 SDK; 2007

Microsoft: <http://msdn2.microsoft.com/en-us/library/ms550992.aspx>, MSDN Microsoft Office SharePoint Server 2007 SDK; 2007

## 7. Anlagenverzeichnis

I	Kundendokumentation
II	Technische Dokumentation
III	Pflichtenheft
IV	Zeitplanung
V	Glossar



# **Anlage I: Kundendokumentation**

## **BookPoint für Microsoft Office SharePoint Server 2007**

Firma: T-Systems Multimedia Solutions GmbH  
Riesaer Straße 5  
01129 Dresden

Erstellt von: Robert Mühsig  
Anschrift: Wiesentorstraße 10  
01612 Nünchritz

Version: 1.0  
Letzte Änderung: 19. April 2007

# Inhaltsverzeichnis

1.	Hinweise für die Anwendung .....	1
2.	Bedienungsanleitung.....	1
2.1	Anmeldung.....	1
2.2	Übersichtsseite .....	1
2.3	Neue Bücher anlegen .....	2
2.4	Bücher editieren & löschen.....	4
2.5	Buchdetails.....	5
2.6	Kontakt.....	5
3.	Ansicht der SharePoint-Liste.....	5

## 1. Hinweise für die Anwendung

Für das Verwenden der Anwendung muss das Microsoft .NET 2.0 Framework installiert sein, da verschiedene Funktionen daraus genutzt werden.

Die SharePoint-Liste muss, wie in der technischen Dokumentation beschrieben, von Ihrem SharePoint-Administrator angelegt sein.

Wenn diese Voraussetzungen erfüllt sind, kann die Anwendung direkt gestartet werden. Es ist keine Installation nötig, somit kann die Software auch später wieder vollständig gelöscht werden.

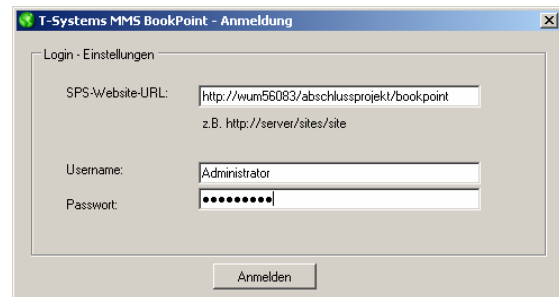
## 2. Bedienungsanleitung

### 2.1 Anmeldung

Nach dem Starten der Anwendung erscheint das Anmeldefenster.

Der Eintrag „SPS-Website-URL“ steht für den Pfad auf der Seite, wo die Buchliste liegt.

Damit nur authentifizierte Nutzer Zugriff haben, muss man sich mit Passwort und Nutzernamen anmelden.



*Anmeldefenster*

Falls der angegebene Benutzername oder das Passwort ungültig sind oder der Nutzer keine Rechte auf der Seite hat, wird eine Fehlermeldung ausgegeben.

### 2.2 Übersichtsseite

Nach dem erfolgreichen Anmelden am SharePoint wird automatisch eine aktuelle Liste geladen.

Die Übersichtsseite besteht aus einer Menüleiste, einer Optionsspalte und der eigentlichen Tabelle.

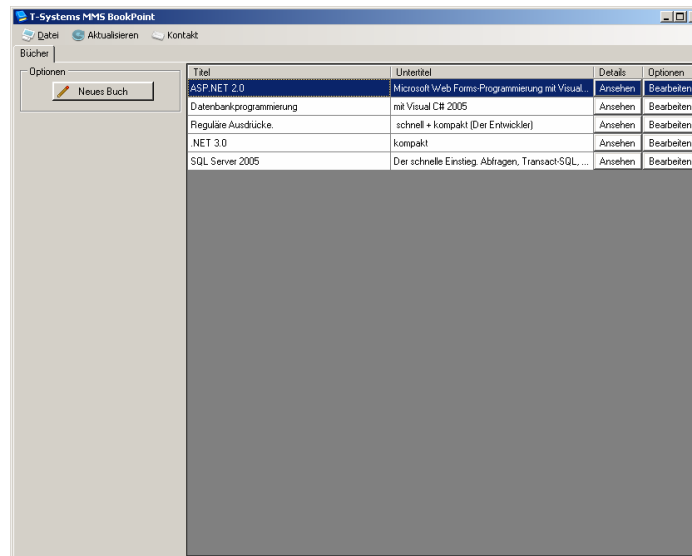
Die Menüleiste umfasst drei Punkte:

- Datei
- Aktualisieren
- Kontakt

Unter dem Menüpunkt Datei befindet sich, ähnlich vielen anderen Windows-Anwendungen, eine Beenden-Option. Eine andere Möglichkeit die Anwendung zu beenden, ist das „X“ oben rechts.

Mit dem Menüpunkt „Aktualisieren“ kann man die aktuelle Liste vom SharePoint abrufen. Die Liste wird automatisch nach dem Login oder einen anderen Aktion abgerufen. Da aber auch andere Benutzer Bücher verändern können, kann die Liste über den Menüpunkt manuell nachgeladen werden.

Der Menüpunkt Kontakt umfasst Kontaktinformationen um an Support für diese Anwendung zu kommen.



*Übersichtsseite der Anwendung*

In der Mitte des Fensters befindet sich die eigentliche Liste. Auf der linken Seite befindet sich das Optionsmenü mit dem „Neues Buch“-Button.

In der Liste sieht man alphabetisch sortiert die aktuellen Bücher welche auf dem SharePoint hinterlegt sind. Dargestellt werden die Titel und Untertitel der Bücher.

Um mehr Informationen zum Buch zu erhalten befindet sich rechts neben den Titel und Untertitel der „Ansehen“-Button.

Über den „Bearbeiten“-Button gelangt man zum Editierfenster für das jeweilige Buch.

### 2.3 Neue Bücher anlegen

Um neue Bücher anzulegen, klicken Sie auf der Übersichtsseite auf den „Neues Buch“-Button.

Es öffnet sich ein neues Fenster in welchen sich die Buchinformationen eingeben lassen.

Auf der linken Seite kann man die allgemeinen Buchinformationen hinterlegen, sowie Angaben zur Bewertung und Sprache, in welche das Buch geschrieben wurde, machen.

Falls mehrere Bücher vorhanden sind, kann die Checkbox „Mehrere Exemplare“ angeklickt werden. Nach dem anklicken wird eine neues Eingabefeld sichtbar in der man die Anzahl der Bücher eingeben kann.

Auf der rechten Seite befinden sich die Auswahlboxen für den Autor und die Kategorie.

Alle bereits eingetragenen Autoren und Kategorien werden automatisch als Auswahl angeboten. Falls ein Autor oder eine Kategorie noch nicht angelegt wurde, kann man diese jeweils hinzufügen.

Pro Buch kann jedoch immer nur ein Autor und eine Kategorie angegeben werden.

*Ausgefüllte Eingabemaske*

Folgende Angaben sind zwingend einzutragen: ISBN, Titel, Mitarbeiter, Autor, Sprache, Bewertung und Kategorie.

Um das Anlegen zu erleichtern kann man Buchinformationen von Amazon.de laden. Dazu muss eine gültige ISBN eingegeben werden. Dabei sollte die ISBN nur aus Zahlen und ohne Bindestriche bestehen. Beim Klick auf den „Amazon!“-Button werden die Buchinformationen abgerufen. Falls keine Buchinformationen vorhanden sind, wird eine Fehlermeldung ausgegeben.

Nach dem Eintragen aller gewünschten Buchinformationen fügt man das Buch über den „Hinzufügen“-Button in die SharePoint-Liste auf dem SharePoint ein.

Das Fenster wird danach automatisch geschlossen und eine aktuelle Buchliste, mit dem eben eingegeben Buch, wird vom SharePoint geladen.

Falls kein Buch eingegeben werden soll, genügt ein Klick auf das „X“ oben rechts oder auf den „Zurück“-Button.

## 2.4 Bücher editieren & löschen

Bücher können editiert werden, indem in der Übersichtsseite der „Bearbeiten“-Button gedrückt wird.

Es öffnet sich das Fenster zum Bearbeiten der Bücher.

Das Fenster ist ähnlich strukturiert wie das Fenster zum Anlegen neuer Bücher.

*Fenster zum Editieren eines Buches*

Im Fenstertitel erscheint das aktuell bearbeitete Buch und alle vorhandenen Buchdaten werden in den Eingabefeldern hinterlegt.

Um ein Buch aus der SharePoint-Liste zu entfernen muss der „Löschen“-Button gedrückt werden.

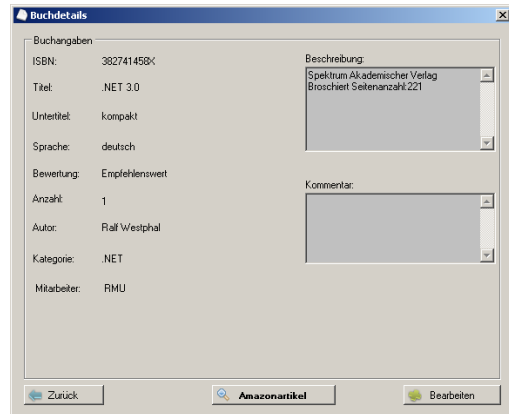
Falls Buchinformationen aus Amazon.de geholt werden sollen, kann dies wie beim Eintragen über den „Amazon!“-Button geschehen.

Die geänderten Informationen werden erst bei einem Klick auf „Einstellungen speichern“ auf dem SharePoint gesichert. Falls das Fenster geschlossen wird, werden alle geänderten Einstellungen verworfen.

## 2.5 Buchdetails

Um mehr Informationen zu einem Buch zu bekommen, klicken Sie in der Übersichtsseite bei dem jeweiligen Buch auf den „Anschauen“-Button. Alle eingetragenen Buchdaten werden in dem neuen Fenster angezeigt.

Falls man das Buch bearbeiten möchte, gelangt man über den „Bearbeiten“-Button zu der Editierseite des Buches. Über den „Amazonartikel“-Button wird bei Amazon.de geprüft ob es ein Buch mit der angegebenen ISBN gibt. Wenn das Buch auch im Amazon.de Produktkatalog vorhanden ist, wird automatisch der Standardbrowser gestartet und die Amazon.de Artikelseite aufgerufen.



The screenshot shows a window titled 'Buchdetails' with the following fields:

Buchangaben	Beschreibung
ISBN: 382741458X	Spektrum Akademischer Verlag Brochüre-Seitenanzahl:221
Titel: .NET 3.0	
Untertitel: Kompakt	
Sprache: deutsch	
Bewertung: Empfehlenswert	
Anzahl: 1	Kommentar:
Autor: Ralf Westphal	
Kategorie: .NET	
Mitarbeiter: RMU	

Buttons at the bottom: Zurück, Amazonartikel, Bearbeiten

*Buchdetailsfenster*

## 2.6 Kontakt

Durch einen Klick auf den Menüpunkt „Kontakt“ auf der Übersichtsseite öffnet sich das Kontaktfenster.

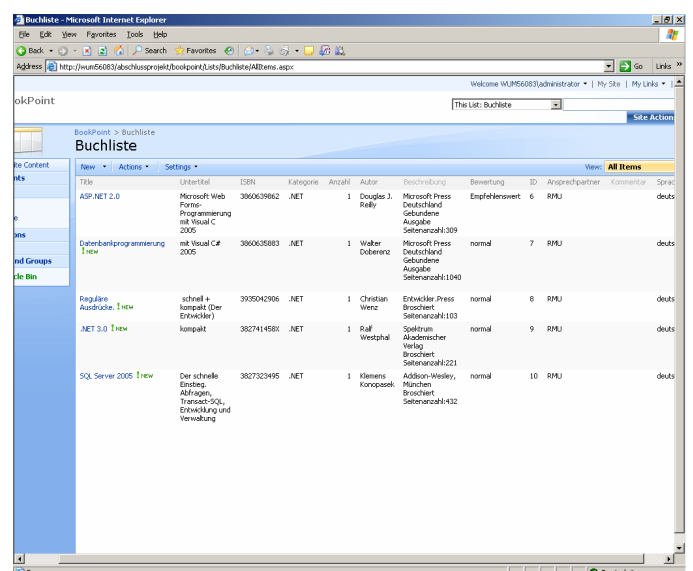
Dieses Fenster enthält Kontaktinformationen, falls es zu Problemen oder Störungen mit der Anwendung kommt.

## 3. Ansicht der SharePoint-Liste

Um die Liste auch ohne die Anwendung anzuschauen, kann man die komplette Buchliste mit allen Inhalten auch direkt im Browser ansehen.

Dazu muss man die gleiche Seitenadresse angeben, wie beim Login. Unter dieser Adresse befindet sich dann die „Buchliste“.

In dieser Buchliste sind alle Buchinhalte in tabellarischer Form gespeichert. Diese Daten können direkt im Browser sortiert und bearbeitet werden. Es können auch neue Bücher über die Listentoolbar („New“, „Actions“, „Settings“) erzeugt werden.



The screenshot shows a SharePoint web page titled 'Buchliste' with a table of book entries. The table has columns for Title, Untertitel, ISBN, Kategorie, Anzahl, Autor, Beschreibung, Bewertung, ID, Ansprechpartner, and Sprach.

Titel	Untertitel	ISBN	Kategorie	Anzahl	Autor	Beschreibung	Bewertung	ID	Ansprechpartner	Sprach
ASP.NET 2.0	Microsoft Web Forms Programmierung mit Visual C# 2005	3800639862	.NET	1	Douglas J. Reilly	Microsoft Press Deutschland gebundene Ausgabe Seitenanzahl:309	Empfehlenswert	6	RMU	deutsch
Datenbankprogrammierung	mit Visual C# 2005	3800635883	.NET	1	Walter Doberner	Microsoft Press Deutschland gebundene Ausgabe Seitenanzahl:1040	normal	7	RMU	deutsch
Regulare Ausdrücke	schnell + kompakt (Der Entwickler)	3925042906	.NET	1	Christian Wernz	Erwäcker Press Brochüre Seitenanzahl:103	normal	8	RMU	deutsch
.NET 3.0	Kompakt	382741458X	.NET	1	Ralf Westphal	Spektrum Akademischer Verlag Brochüre Seitenanzahl:221	normal	9	RMU	deutsch
SQL Server 2005	Der schnelle Einstieg, Aufgaben, Transact-SQL, Entwicklung und Verwaltung	3827320495	.NET	1	Klemens Koppasch	Addison-Wesley, München Brochüre Seitenanzahl:432	normal	10	RMU	deutsch

*Ansicht im SharePoint*

Für eine genaue Beschreibung des SharePoint, lesen Sie bitte die Microsoft Office SharePoint Server 2007 Hilfe.

# **Anlage II: Technische Dokumentation**

## **BookPoint für Microsoft Office SharePoint Server 2007**

Firma: T-Systems Multimedia Solutions GmbH  
Riesaer Straße 5  
01129 Dresden

Erstellt von: Robert Mühsig  
Anschrift: Wiesentorstraße 10  
01612 Nünchritz

Version: 1.0  
Letzte Änderung: 19. April 2007



# Inhaltsverzeichnis

1.	Komponentenübersicht .....	1
2.	Projektdatei .....	2
3.	Microsoft Office SharePoint Server 2007 .....	2
3.1	Systembeschreibung .....	2
3.2	Konfiguration .....	3
4.	Windows-Clientanwendung .....	4
4.1	Übersicht .....	4
4.2	Softwaredesign .....	4
4.2.1	Backend .....	5
4.2.2	Webservices .....	9
4.2.2.1	SharePoint .....	9
4.2.2.2	Amazon .....	10
4.2.3	Benutzeroberfläche .....	11

## 1. Komponentenübersicht

Die Anwendung „BookPoint“ setzt sich aus drei einzelnen Komponenten zusammen.

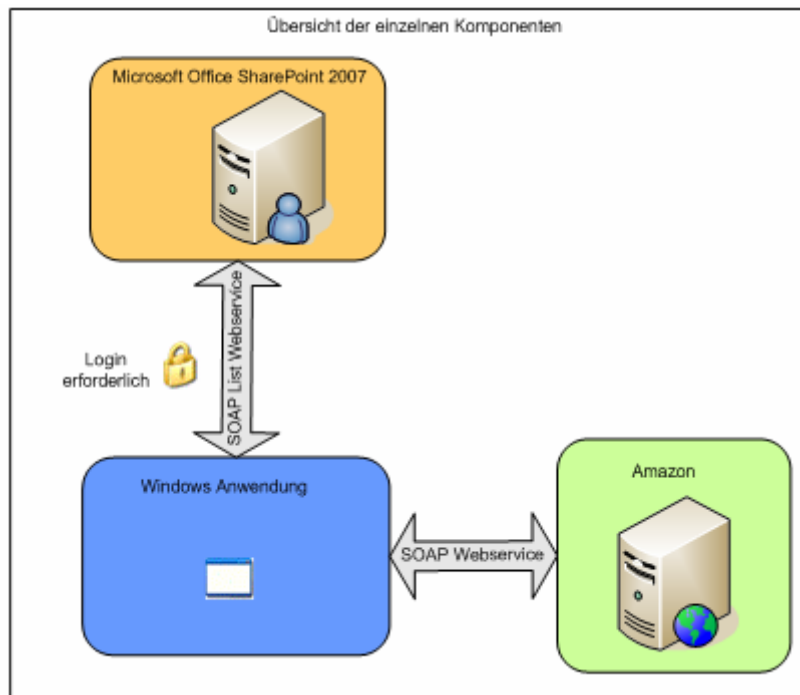
Der Microsoft Office SharePoint Server 2007 ist für die Datenhaltung und die Authentifizierung zuständig. Er bildet die Basis für die Windows-Anwendung. Die Daten auf dem SharePoint können grundsätzlich auch über den Browser abgerufen und verändert werden.

Über die Windows Anwendung kann sich der Nutzer anmelden und es werden die aktuellen Daten vom SharePoint geladen. Alle gemachten Änderungen in der Windows-Anwendung werden ebenso auf dem SharePoint ausgeführt.

Dies geschieht automatisch direkt nach erfolgter Aktion um einen inkonsistenten Datenbestand zu vermeiden.

Als dritte Komponente tritt der Webservice von Amazon.de in Erscheinung. Dieser Webservice erleichtert das Eingeben von Buchdaten und ist daher nur als Hilfssystem anzusehen.

Da dies aber wichtiger Bestandteil des Projektauftrages war, werden die Anbindungen zu Amazon.de ebenso behandelt wie die SharePoint-Anbindung.



*Übersicht über die einzelnen Komponenten*

Alle drei Komponenten sind verbunden über die einzelnen Webservices. Eine direkte Kopplung von SharePoint und Amazon wäre zwar grundsätzlich möglich, ist aber unnötig, da nur die Windows-Anwendung die Amazon Webservices nutzt.

Die Kommunikation der einzelnen Komponenten erfolgt über SOAP.

Damit nur autorisierte Nutzer die Liste bearbeiten, muss sich jeder Nutzer über die Windows-Anmeldung mit seinen Login-Daten am SharePoint anmelden. Die Anmeldedaten werden über den Webservice verschlüsselt übertragen.

## **2. Projektdaten**

Das Projekt wurde komplett auf einem Microsoft Virtual PC Image erstellt, da in der T-Systems Multimedia Solutions GmbH noch kein SharePoint 2007 installiert ist.

Virtual PC emuliert einen virtuellen Rechner, ähnlich wie das Programm VMWare.

Das Image benötigt den Microsoft Virtual PC 2004.

## **3. Microsoft Office SharePoint Server 2007**

### **3.1 Systembeschreibung**

Der Microsoft Office SharePoint 2007 dient als Basis für die Anwendung. Neben den Daten, welche dort gehalten werden, beruht auch das Login-System darauf.

Konfiguriert wird der SharePoint über sein webbasierendes Frontend, welches mit den aktuellen Browsern (Microsoft Internet Explorer 6/7 und Mozilla Firefox) betrachtet werden kann.

Da der SharePoint eine Vielzahl an Funktionen bietet, wird in den Dokumentationen nicht näher auf den kompletten SharePoint eingegangen.

Mehr Informationen findet man unter dieser Adresse:

***<http://office.microsoft.com/sharepoint/>***

### 3.2 Konfiguration

Der SharePoint ist hierarchisch aufgebaut. Jeder berechtigte Nutzer, kann neue „Sites“ oder „Sitecollections“ anlegen.

Die Seitenstruktur lässt sich am anschaulichsten anhand der URL erklären:

***http://wum56083/abschlussprojekt/bookpoint/***

*Server      Sitecollection      Site*

„wum56083“ ist der Rechner auf dem der SharePoint installiert ist.

„abschlussprojekt“ ist eine „Sitecollection“. Eine SharePoint „Sitecollection“ wird als eine Art übergeordnete Seite gesehen, welches ein bestimmtes Thema von einem anderen abgrenzt.

„bookpoint“ ist eine ganz konkrete „Site“. Eine Seite ist ein kleiner Teil einer „Sitecollection“ und kann wiederum mehrere SharePoint Elemente enthalten.

Ein Element kann z.B. eine Liste, ähnlich einer Excel Liste, sein.

Für den Zugriff auf die SharePoint Liste ist es wichtig, dass die Tabelle „**Buchliste**“ heißt und die Spalten wie folgt konfiguriert sind:

Spaltenname	Typ
Title* **	Single line of text
Untertitel	Single line of text
ISBN*	Single line of text
Beschreibung	Multiple lines of text
Kommentar	Multiple lines of text
Autor*	Single line of text
Kategorie*	Single line of text
Sprache	Single line of text
Bewertung	Single line of text
Anzahl	Number
Ansprechpartner*	Single line of text

\* = Pflichtfeld

\*\* = Im englischen SharePoint muss es „Title“ heißen.

## 4. Windows-Clientanwendung

### 4.1 Übersicht

Die Clientanwendung benötigt das Microsoft .NET Framework 2.0 und kann aus dem Virtual PC Image kopiert werden.

Die Anwendung bedarf keiner Installation auf dem System.

### 4.2 Softwaredesign

Die Software wurde in zwei Teile getrennt um später die einzelnen Teile flexibler einzusetzen.

Die komplette Software wurde mit Microsoft Visual Studio 2005 erstellt. Dabei wurden zwei Projekte angelegt. Zum Testen der einzelnen Funktionalitäten wurde noch ein drittes Projekt erstellt.

Da die Testapplikation nur in der Entwicklungszeit zum Einsatz kam und nur die Funktionsaufrufe testete, wird in der Dokumentation keinen Bezug mehr darauf genommen.

#### Auflistung der Teilprojekte:

Backend - Klassenbibliothek: **MMS.BookPoint.Definitions**

Frontend – Windows Forms Projekt: **MMS.BookPoint.WinApp**

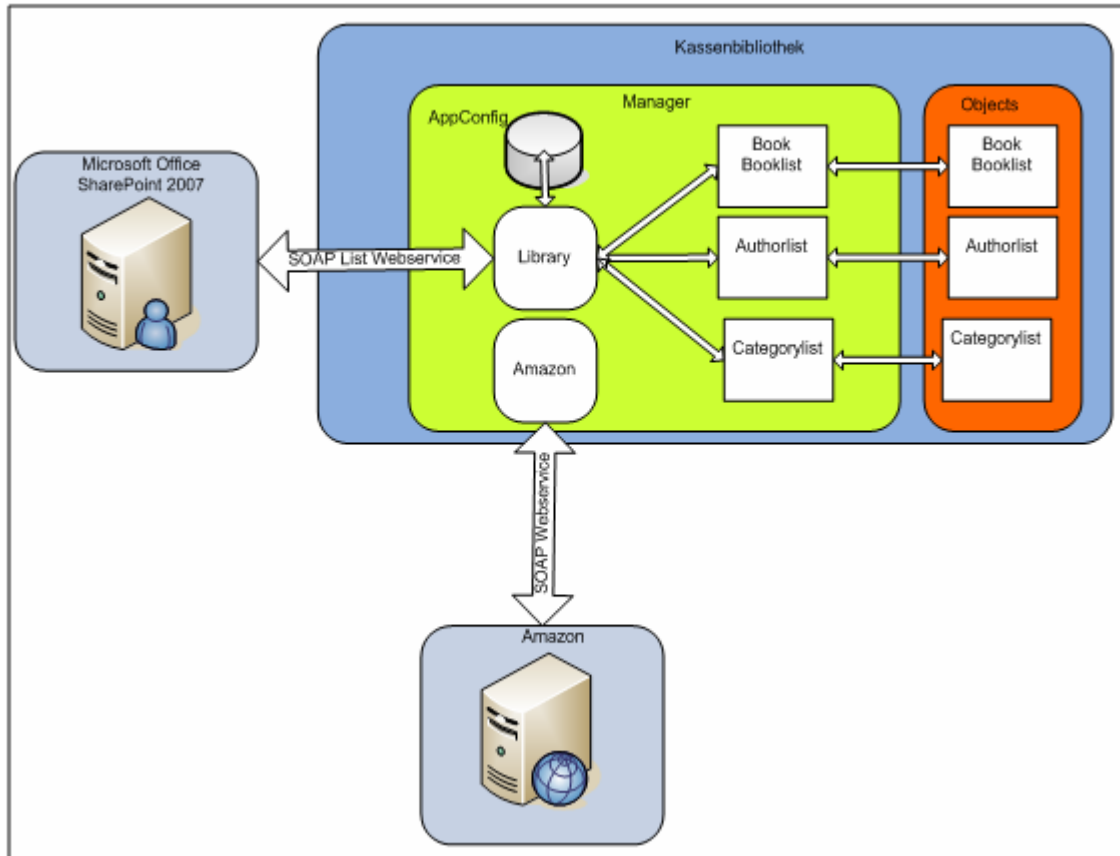
Test Program – Konsolenapplikation: **MMS.BookPoint.TestApp**

Für die Benennung wurden die .NET Coding-Style Regeln beachtet. Microsoft stellte diese Regel auf, damit in einem Projekt mehr Überblick herrscht, welche Klassen und Funktionen von welcher Firma stammen. Der Projektname, welcher automatisch den Namespace darstellt, hält sich an das Muster:  
*„Firma.Produktname.Produkteil“*

### 4.2.1 Backend

Für das Backend wurde ein neues Projekt „**MMS.BookPoint.Definitions**“ angelegt.

Das Backend ist eine Programmbibliothek, welche die Kommunikation zum SharePoint sowie zu den Amazon Webservices übernimmt und auch die eigentliche Logik enthält.



Überblick über das Backend der Windows-Anwendung

Neben den SOAP Anbindungen zu Amazon und dem SharePoint besteht auch das Backend aus zwei Teilen.

Die Klassen und Methoden der jeweiligen Webservices wurden unter diesen Namespaces zur Verfügung gestellt:

SharePoint-List-Webservice: **MMS.BookPoint.Definitions.Webservice.Sharepoint**

Amazon-Webservice: **MMS.BookPoint.Definitions.Webservice.Amazon**

Der Namespace **MMS.BookPoint.Definitions.Manager** ist für den Zugriff auf die einzelnen Objekte verantwortlich. Der „*LibraryManager*“ übernimmt den Verbindungsaufbau zum SharePoint. Dabei speichert er temporär intern die Anmeldedaten des Nutzers in die „*AppConfig*“. Diese XML Datei ist in .NET Applikationen zum Zwischenspeichern von bestimmten Daten zuständig.

## SharePoint-Verbindungsabau

Der Verbindungsaufbau geschieht über die statische „*CheckConnection()*“-Methode des „*LibraryManager*“. Dabei muss ein gültiger Benutzername, Benutzerpasswort und SharePoint-URL übermittelt werden. Wenn eine Verbindung erfolgreich aufgebaut wurde, wird die Verbindung im statischen „*LibraryListService*“ Objekt gehalten. Die Verbindungsdaten werden durch den Webservice verschlüsselt übertragen. Wenn keine Verbindung zustande kommt, gibt die Methode „*false*“ zurück.

Die Abbildung verdeutlicht den Zugriff auf den List-Webservice:

```
public bool Connecting()
{
    if (LibraryManager.CheckConnection())
    {
        XmlNode Value = LibraryManager.LibraryListService.GetList("Buchliste");
        return true;
    }
    else
    {
        return false;
    }
}
```

*Zugriffsprinzip auf den List-Webservice*

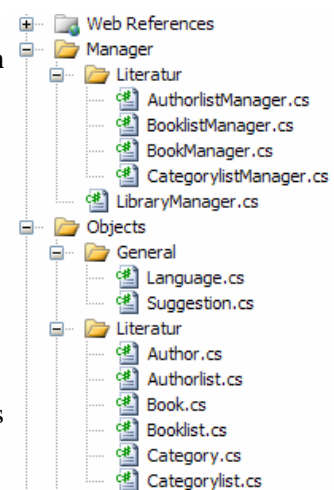
## Manager & Objects

Der Zugriff auf die einzelnen Elemente der SharePoint-Liste wird über die „*Manager*“ geregelt. Die „*Manager*“ können die Daten laden und verändern, dabei speichern sie diese Daten in die jeweiligen „*Objects*“.

Die Manager sind im Namespace „*MMS.BookPoint.Definitions.Manager*“ zu finden. In dem Namespace „*MMS.BookPoint.Definitions.Objects*“ sind die „*Objects*“ hinterlegt. Unter den beiden Namespaces wurde noch eine logische Einteilung vorgenommen. „*Literatur*“ steht dabei für den Namespace der Bücher. Falls später noch andere Medien eingebunden werden, können so einfach neue Namespaces angelegt werden.

Da alle Daten, wie Autoren und Kategorien, direkt an ein Buch gebunden sind, gibt es für diese beiden Objekttypen keinen eigenen Manager, welcher Änderungen an diesen Objekten erlauben. Über den „*AuthorlistManager*“ und den „*CategorylistManager*“ können jedoch alle bereits bestehenden Autoren und Kategorien geladen werden. Alle Buchdaten welche in der SharePoint-Liste gespeichert und können über den „*BooklistManager*“ abgerufen werden.

Für das Bearbeiten, Löschen, Laden und Anlegen eines einzelnen Buches hat der „*BookManager*“ entsprechende Methoden.



*Projektstruktur des Backends*

## Manager-Übersicht

### (static) LibraryManager:

Stellt die Verbindung zum SharePoint her und speichert diese im „*LibraryListService*“-Objekt.

Methoden: *static CheckConnection()* – Rückgabewert: *Boolean*

### (static) BooklistManager:

Stellt Methoden bereit, um ein „*Booklist*“-Objekt zu befüllen und zu verändern.

Methoden: *static Load* – Rückgabewert: *Booklist*

Lädt alle Bücher aus der SharePoint-Liste in ein „*Booklist*“-Objekt.

### (static) AuthorlistManager:

Stellt Methoden bereit, um ein „*Authorlist*“-Objekt zu befüllen und zu verändern.

Methoden: *static Load* – Rückgabewert: *Authorlist*

Lädt alle Autoren Daten aus der SharePoint-Liste in ein „*Authorlist*“-Objekt.

### (static) CategorylistManager:

Stellt Methoden bereit, um ein „*Categorylist*“-Objekt zu befüllen und zu verändern.

Methoden: *static Load* – Rückgabewert: *Categorylist*

Lädt alle Kategoriedaten aus der SharePoint-Liste in ein „*Categorylist*“-Objekt.

### (static) BookManager:

Stellt Methoden bereit, um ein „*Book*“-Objekt zu befüllen und zu verändern.

Methoden: *static Load* – Rückgabewert: *Book*

Lädt ein einzelnes Buch aus der SharePoint-Liste anhand der ID in ein „*Book*“-Objekt.

*static Create* – Rückgabewert: *boolean*

Erstellt einen neuen Datensatz in der SharePoint-Liste anhand des übergebenden Buches.

*static Update* – Rückgabewert: *boolean*

Editiert einen bestehenden Datensatz in der SharePoint-Liste anhand des übergebenden Buches.

*static Delete* – Rückgabewert: *boolean*

Löscht einen bestehenden Datensatz in der SharePoint-Liste anhand des übergebenden Buches.



## Objects-Übersicht

### Book:

Repräsentiert ein reales Buch und speichert verschiedene Buchdaten.

Attribute:

- *int id* – Listenelement ID
- *string isbn* – ISBN des Buches
- *string title* - Buchtitel
- *string subtitle* - Buchuntertitel
- *string description* - Buchbeschreibung
- *string comment* - Kommentar
- *int count* - Anzahl
- *string authorlist* - Autor
- *string categorylist* - Kategorie
- *Suggestion suggestion* - Bewertung
- *Language language* - Sprache
- *string contactPerson* – Ansprechperson für dieses Buch

### Suggestion:

Aufzählung („Enums“) an Bewertungseinheiten.

- „Empfehlenswert“
- „Normal“
- „Schlecht“

### Language:

Aufzählung („Enums“) an wählbaren Sprachen.

- „deutsch“
- „englisch“

### Booklist:

Repräsentiert eine Liste (eine „Collection“) an „Book“-Objekten.

Erbt von der .NET Klasse „List“ und nimmt nur Objekte vom Typ „Book“ auf:

*Booklist* : *List*<*Book*>

### Author:

Repräsentiert einen Autor und speichert dessen Namen.

Attribute:

- *string name* – Name des Autors

### Authorlist:

Repräsentiert eine Liste (eine „Collection“) an „Author“-Objekten.

Erbt von der .NET Klasse „List“ und nimmt nur Objekte vom Typ „Author“ auf:

*Authorlist : List<Author>*

### Category:

Repräsentiert eine Kategorie und speichert dessen Namen.

Attribute:

- *string name* – Name der Kategorie

### Categorylist:

Repräsentiert eine Liste (eine „Collection“) an „Category“-Objekten.

Erbt von der .NET Klasse „List“ und nimmt nur Objekte vom Typ „Category“ auf:

*Categorylist : List<Category>*

## **4.2.2 Webservices**

### **4.2.2.1 SharePoint**

Der SharePoint 2007 bietet, wie auch der SharePoint 2003, eine Reihe von Webservices, welche es ermöglichen, bestimmte Funktionen und Daten in andere Programme zu integrieren.

Für die Anwendung „BookPoint“ war es das Ziel, Daten aus einer Liste abzurufen und zu bearbeiten.

Dafür gibt es den sogenannten „List Webservices“. Die WSDL befindet sich unter folgender URL:

***http://server/sitecollection/site/\_vti\_bin/Lists.asmx***

Im Projekt BookPoint ist sie daher unter folgenden Pfad zu finden:

***http://wum56083/abschlussprojekt/bookpoint/\_vti\_bin/Lists.asmx***

Nach Einbinden des Webservice in Visual Studio 2005 werden automatisch alle Methoden und Klassen, welche der SharePoint-List Webservice bereitstellt, unter dem Namespace „MMS.BookPoint.Definitions.Webservice.Sharepoint“ generiert und verfügbar gemacht.

Die Methoden des SharePoint Webservices verwenden XML Elemente für die Ein- und Ausgabe. Microsoft hat für dafür einen eigenen XML Dialekt „CAML“ entworfen. Über „CAML“ (Collaboration-Application-Markup-Language) können viele Funktionen des SharePoints angesprochen werden.

Für den Zugriff auf die Bücher der Liste wurde die „*LibraryListService.GetListItems*“-Methode genommen. Das Bearbeiten, Löschen und Anlegen neuer Bücher erfolgt über die „*LibraryListService.UpdateListItems*“-Methode. Die „*UpdateListItems*“-Methode nimmt „CAML“ Elemente entgegen. Da die „CAML“-Elemente sehr komplex sind und Microsoft eine ausführliche Dokumentation bereitstellt, wird hier nicht näher darauf eingegangen.

#### 4.2.2.2 Amazon

Der Amazon Webservice ist für alle frei und kostenlos verfügbar. Bedingung hierbei ist, dass man sich unter <https://aws-portal.amazon.com> registriert.

Die genaue Bezeichnung für den Webservice, der es ermöglicht auf Produktinformationen zuzugreifen, nennt sich „Amazon E-Commerce Service“.

Die WSDL um auf die Datenbestände von Amazon.de zuzugreifen befindet sich unter dieser Adresse:

**<http://webservices.amazon.com/AWSECommerceService/DE/AWSECommerceService.wsdl>**

Man kann neben Amazon.com und Amazon.de auch auf Amazon.fr, Amazon.ca, Amazon.co.jp, und Amazon.co.uk zugreifen. Für Amazon.com gibt es diese Adresse:

**<http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl>**

In diesem Projekt wurde allerdings nur von der deutschen WSDL gebrauch gemacht.

Ähnlich wie bei den SharePoint Webservices werden nach dem Einbinden der Amazon Webservices die benötigten Methoden und Klassen für den Zugriff generiert.

Diese wurden unter dem Namespace „*MMS.BookPoint.Definitions.Webservice.Amazon*“ verfügbar gemacht.

Aus zeitlichen Gründen wurde kein „*AmazonManager*“ implementiert. Da die Funktion des SharePoints im Frontend nur beim Eintragen und Editieren zum Tragen kommt, bleibt trotzdem eine Übersicht erhalten, da die Webservice -Methode zum Verbinden zu Amazon.de und zum Abrufen von Buchdaten direkt in die Frontend Funktion geschrieben wurde.

Um die Buchdaten abzurufen, wurde die Methode „*ItemSearch*“ implementiert. Diese Methode nimmt ein Objekt der Klasse „*ItemSearchRequest*“ als Parameter entgegen. In diesem Objekt hinterlegt man seine genauen Suchkriterien.

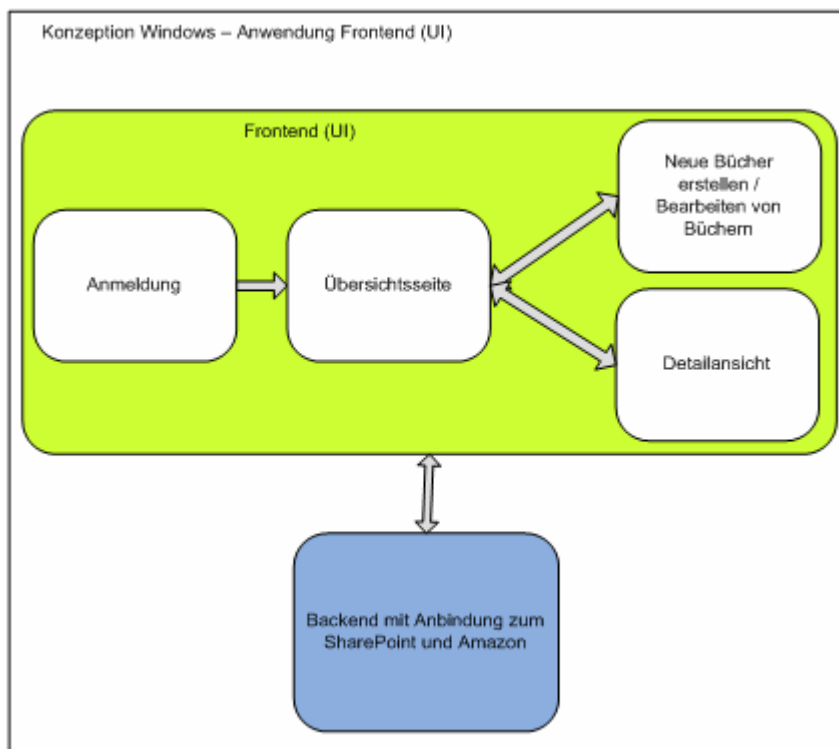
Das Suchkriterium in diesem Projekt war die ISBN Nummer, diese kann man als „*Keyword*“ in dem „*ItemSearchRequest*“ hinterlegen.

Rückgabewert der „*ItemSearch*“ Methode ist ein Objekt der Klasse „*ItemSearchResponse*“. In diesem Objekt ist eine Auflistung aller möglichen Artikel die gefunden wurden. Da pro ISBN nur ein Buch hinterlegt ist und über das Frontend nur eine ISBN eingegeben werden kann, findet diese Methode entweder ein einzelnes Buch oder keins.

### 4.2.3 Benutzeroberfläche

Das Frontend ist für die Darstellung der einzelnen Fenster verantwortlich und greift auf das Backend zurück.

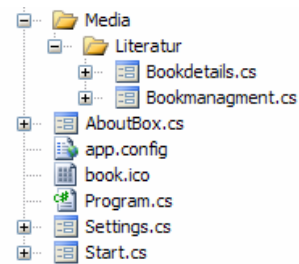
Die Frontend-Daten wurden in einem neuen Projekt „**MMS.BookPoint.WinApp**“ erstellt. Damit später die Funktionen des Backends genutzt werden können, wurde die „*MMS.BookPoint.Definitions.dll*“ als Verweis in das Projekt eingebunden.



Übersicht über den Ablauf der Benutzerinteraktionen

Das Frontend besteht aus vier Hauptkomponenten:

- Loginfenster (*Settings.cs*)
- Übersichtsfenster (*Start.cs*)
- Anlegen und Bearbeiten der Bücher (*Bookmanagment.cs*)
- Detailsansicht für die Bücher (*Bookdetails.cs*)



Strukturübersicht Frontend

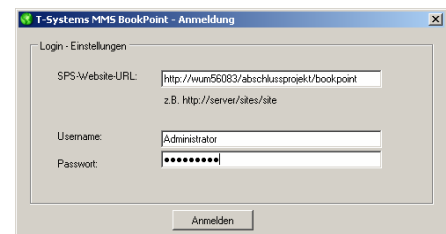
Der genaue Ablauf der Benutzerinteraktionen wird in der Kundendokumentation beschrieben.

### Technischer Ablauf & Implementation des Backends

Da nur berechtigte Nutzer auf die Liste zugreifen dürfen, ist der Einstiegspunkt für den Benutzer das Anmeldefenster (*Settings.cs*).

Das Loginfenster besteht daher aus drei Eingabefeldern, wobei die SharePoint URL standardmäßig auf diese URL eingestellt wurde:

***<http://wum56083/abschlussprojekt/bookpoint>***



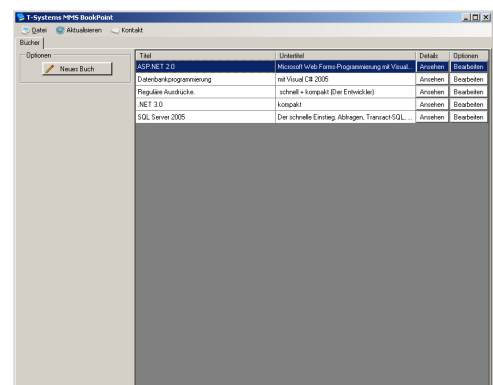
Loginfenster

Über den Anmeldebutton werden die Daten dem „LibraryManager“ übergeben und er versucht eine Verbindung zum SharePoint aufzubauen. Gelingt dies nicht, erscheint eine Fehlermeldung.

Nach der erfolgreichen Anmeldung bekommt er eine Übersichtseite (*Start.cs*) angezeigt, welche ihm in einer Liste einen Überblick über die bestehenden Bücher gibt.

Die Übersichtsseite besteht aus drei Teilen. Wie in den meisten Windows-Anwendungen bietet auch die BookPoint Anwendung eine Menüleiste. Die Menüleiste selbst enthält nur wenige Punkte und wurde angelegt, um evtl. später Optionsmenüs zu bieten.

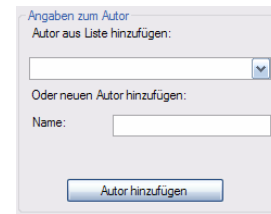
Damit die Anwendung möglicherweise später auch andere Medien, wie DVDs, verwalten sollte, wurde ein *TabControl* verwendet. Tabs oder auch Reiter können genutzt werden um Inhalte besser voneinander zu trennen.



Übersichtsseite



Da einige Bücher vom gleichen Autor erstellt wurden und viele Bücher einer Kategorie zugeordnet werden können, wurde für diese beiden Sachen eine Combobox erstellt. Über die Klassen „*AuthorlistManager*“ und „*CategorylistManager*“ aus dem Backend werden diese Comboboxen befüllt. Falls ein Autor oder eine Kategorie noch nicht in der Liste auftaucht, kann man beide noch hinzufügen.



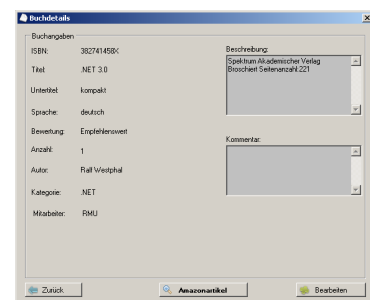
*Combobox für die Autoren*

Wenn ein neues Buch hinzugefügt oder ein bestehendes Buch verändert oder gelöscht wird, wird eine aktuelle Liste vom SharePoint geholt um Dateninkonsistenz zu vermeiden.

Da man häufiger nur Buchdetails anschauen will, als ein Buch bearbeiten, wurde dafür der „Detail“-Button in dem Datagridview erstellt.

Beim Klick auf den Button wird das aktuelle Buch dem Detailfenster „*Bookdetails.cs*“ übergeben. Ähnlich wie bei dem Editionsfenster, werden keine neuen Daten vom SharePoint geholt, sondern aus dem Objekt übernommen.

Die Daten werden auch ähnlich dargestellt. Von dem Detailfenster kann man auch zum Editionsfenster gelangen. Da bestimmte Buchdaten nicht aus Amazon.de direkt übernommen werden können, wurde noch ein Linkbutton zum Amazon-Artikel hinzugefügt. Wenn das Buch eine gültige ISBN hat und auch bei Amazon.de gelistet ist, wird automatisch der Standardbrowser mit der entsprechenden Amazon-Seite geladen.



*Detailsseite*

# **Anlage III: Pflichtenheft**

BookPoint für Microsoft Office SharePoint Server 2007

Firma: T-Systems Multimedia Solutions GmbH  
Riesaer Straße 5  
01129 Dresden

Erstellt von: Robert Mühsig  
Anschrift: Wiesentorstraße 10  
01612 Nünchritz

Version: 1.0  
Letzte Änderung: 10. April 2007



# Inhaltsverzeichnis

1.	Einleitung.....	1
1.1	Situationsbeschreibung .....	1
1.2	Auftragegeber .....	1
2.	Zielbestimmungen .....	2
2.1	Musskriterien .....	2
2.2	Abgrenzung.....	2
3.	Funktionsumfang .....	3
3.1	Zu speichernde Daten.....	3
3.2	Benutzerinteraktionen .....	3
4.	Konzeption.....	4
5.	Technische Spezifikationen .....	4
5.1	Serverseitig .....	4
5.2	Clientseitig .....	5
6.	Qualitätsmerkmale.....	5
6.1	Erweiterbarkeit.....	5
6.2	Fehlerbehandlung.....	5
6.3	Wartbarkeit .....	5
6.4	Dokumentiert .....	5
7.	Liefertermin .....	5

## **1. Einleitung**

Um Kunden die Möglichkeiten des SharePoint 2007 zu zeigen, soll ein Prototyp erstellt werden, welche insbesondere auf die Möglichkeit eingeht, weitere Anwendungen mit den SharePoint zu verbinden. Dabei sollen zugleich neuste „Web 2.0“ Features genutzt werden.

### **1.1 Situationsbeschreibung**

Wie in vielen Software Unternehmen, gibt es auch in der T-Systems Multimedia Solutions GmbH (MMS) eine große Anzahl an Fachbüchern. Diese werden in der MMS nicht zentral gelagert, sondern befinden sich beim jeweiligen Mitarbeiter am Arbeitsplatz. Um eine gewisse Ordnung hineinzubringen ist momentan auf dem intern genutzten Microsoft SharePoint Portal Server 2003 eine Art Excel Liste eingerichtet. In dieser wird erfasst, welches Buch ein Mitarbeiter hat. Leider ist die Handhabung und Pflege dieser Liste sehr umständlich, daher soll nun eine einfachere Lösung geschaffen werden, um die dezentralen Bücherbestände auf dem SharePoint zentral zu bearbeiten. Die Lösung soll auf den Microsoft Office SharePoint Server 2007 aufbauen, da eine Migration der Daten von dem alten SharePoint auf die neue Version geplant ist.

### **1.2 Auftraggeber**

Auftraggeber ist Dieter Bauckmeier und ist zuständig für die Projektabnahme.

## **2. Zielbestimmungen**

### **2.1 Musskriterien**

Es soll eine Clientanwendung werden, welche leicht zu bedienen ist und die Bücherbestände auf dem SharePoint bearbeitet.

Die Clientanwendung kann im Browser oder auf dem Desktop laufen.

Dabei sollte eine Desktopvariante ohne Installationsaufwand auf jedem Windows XP PC mit aktuell installierten .NET Framework funktionsfähig sein.

Falls die Clientanwendung eine Webapplikation wird, muss diese mit dem Microsoft Internet Explorer 6/7 und den Mozilla Firefox kompatibel sein.

Die Anwendung muss es erlauben, eine aktuelle Liste aus dem SharePoint zu beziehen und zu bearbeiten. Dazu gehört das Editieren oder das Löschen von bereits bestehenden Büchern. Auch das Anlegen neuer Bücher soll leicht über diese Anwendung realisiert werden.

Damit die Pflege der Buchliste möglichst wenig Zeit in Anspruch nimmt, soll der Webservice von Amazon.de genutzt werden. Dieser soll die manuelle Eingabe der Bücherinformationen möglichst gering halten.

Es muss bei der Anwendung gewährleistet sein, dass nur berechtigte Nutzer auf die SharePoint-Liste Zugriff haben. Hierfür soll das Authentifizierungssystem des SharePoints genutzt werden.

Neben dem eigentlichen Projekt soll eine Projektdokumentation, eine Kundendokumentation und eine technische Dokumentation angefertigt.

### **2.2 Abgrenzung**

Die Anwendung soll keine vollständige Bibliotheksanwendung mit Ausleihfunktion sein. Ziel ist es, die Anbindung an den SharePoint umzusetzen und auch neuste „Web 2.0“ Features zu nutzen.

### **3. Funktionsumfang**

#### **3.1 Zu speichernde Daten**

Die Anwendung muss es erlauben folgende Buchdaten zu speichern:

- Titel
- Untertitel
- Bewertung
- Sprache
- Kommentar
- Beschreibung
- Autor
- Kategorie
- Ansprechpartner

Dabei braucht jeweils nur eine Kategorie und ein Autor gespeichert werden. Kommentare, Untertitel und Beschreibung sollen keine Pflichtfelder sein.

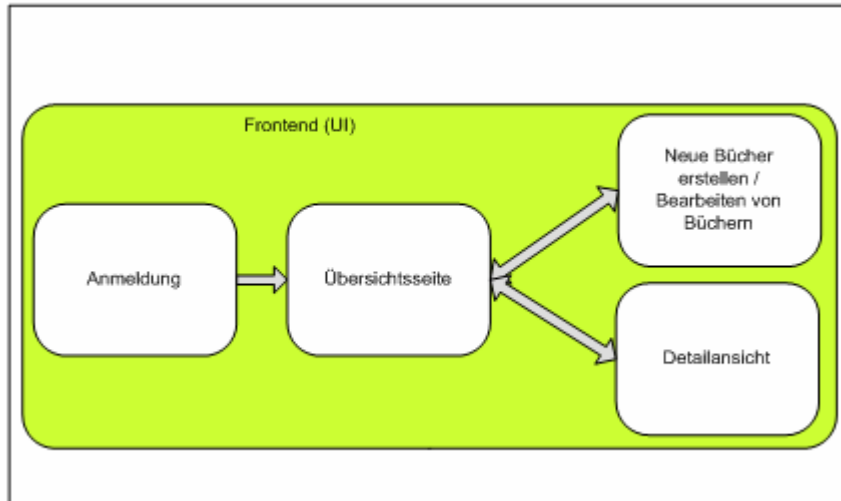
#### **3.2 Benutzerinteraktionen**

Die Clientanwendung muss es erlauben die Informationen auf einfache Weise einzugeben, anzuschauen und zu bearbeiten. Zudem soll es möglich sein Bücher vollständig zu löschen.

Damit auch nur autorisierte Nutzer die Buchdaten aus der SharePoint-Liste bearbeiten können, soll das Login-System des SharePoints genutzt werden. Nur autorisierte Nutzer sollen die Buchdaten abrufen und bearbeiten können.

## 4. Konzeption

Die Applikation soll einfach und linear in der Benutzerführung aufgebaut sein.



*Konzeption Frontend*

Der Einstiegspunkt der Software muss die Anmeldung sein, damit nur berechtigte Nutzer Zugriff haben sollen. Die Übersichtsseite soll grobe Informationen über den momentanen Buchbestand auf dem SharePoint geben. Damit die Übersichtlichkeit gewahrt bleibt, sollen in der Übersichtsseite nur der Titel und der Untertitel sichtbar sein.

Das Erstellen neuer Bücher und das Editieren von bestehenden Büchern soll in derselben Eingabemaske erfolgen.

Die Detailansicht soll ähnlich aufgebaut sein, wie die Editieransicht, nur ohne die Möglichkeit die Buchdaten zu ändern.

## 5. Technische Spezifikationen

### 5.1 Serverseitig

Als Basis für die Anwendung soll ein Microsoft Office SharePoint Server 2007 dienen. Das Betriebssystem für den SharePoint ist der Windows Server 2003.

Da die MMS intern den SharePoint 2007 noch nicht eingesetzt wird, soll der Server als Image für den Microsoft Virtual PC bereit gestellt werden.

Auf dem SharePoint soll eine einfache Liste eingerichtet werden um die Daten zu speichern. Die Buchdaten sollen dabei in geeignete Spalten gespeichert werden.

## **5.2 Clientseitig**

Die Anwendung muss auf allen aktuellen PCs mit Microsoft Windows XP mit installierten .NET Framework 2.0 ohne Installationsaufwand laufen.

Falls eine Webapplikation erstellt wird, soll diese kompatibel zu Microsoft Internet Explorer 6 / 7 und Mozilla Firefox sein.

## **6. Qualitätsmerkmale**

### **6.1 Erweiterbarkeit**

Die Software-Architektur soll erweiterbar und flexibel konstruiert werden um evtl. später Programmteile in anderen Applikationen zu verwenden.

### **6.2 Fehlerbehandlung**

Programmfehler in der Anwendung sollen abgefangen und durch Fehlermeldungen beschrieben werden.

### **6.3 Wartbarkeit**

Damit der Quellcode gut auch von anderen SharePoint-Entwicklern weiterentwickelt werden kann, soll dieser nach den allgemeinen .NET Konventionen entwickelt werden und an geeigneten Stellen kommentiert werden.

### **6.4 Dokumentiert**

Um die Weiterentwicklung zu gewährleisten, soll das Projekt ausreichend dokumentiert werden.

## **7. Liefertermin**

Die fertige Anwendung mit den Dokumentationen soll bis spätestens 20. April beim Auftraggeber eingereicht werden.

# **Anlage IV: Zeitliche Planung**

## **BookPoint für Microsoft Office SharePoint Server 2007**

Firma: T-Systems Multimedia Solutions GmbH  
Riesaer Straße 5  
01129 Dresden

Erstellt von: Robert Mühsig  
Anschrift: Wiesentorstraße 10  
01612 Nünchritz

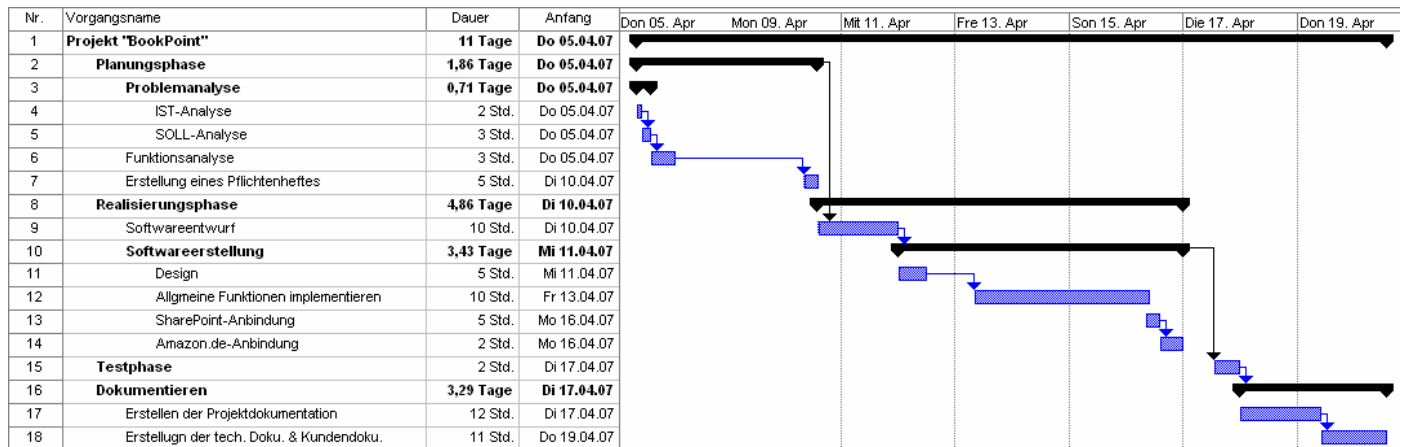
Version: 1.0  
Letzte Änderung: 10. April 2007

## Zeitliche Planung

Die zeitliche Projektplanung wurde mit Hilfe von Microsoft Project 2003 erstellt.

Die Zeitplanung aus dem Projektantrag konnte nicht vollständig eingehalten werden, da bereits vor dem eigentlichen Projektbeginn Dokumentationen der Webservices und des neuen SharePoint gesammelt wurden. Einige Funktionen waren nur unzureichend beschrieben, sodass sich Planänderungen ergeben hatten.

In der Abbildung sind alle relevanten Projektphasen und Termine ersichtlich.



Projektablauf – Zeitliche Planung (Gantt-Balkendiagramm)



# Anlage V: Glossar

## BookPoint für Microsoft Office SharePoint 2007

Firma: T-Systems Multimedia Solutions GmbH  
Riesaer Straße 5  
01129 Dresden

Erstellt von: Robert Mühsig  
Anschrift: Wiesentor Straße 10  
01612 Nünchritz

Version: 1.0  
Letzte Änderung: 19. April 2007

Begriff	Erläuterung
ASP.NET	serverseitige Technologie zum Erstellen von Webanwendungen auf Basis des Microsoft.NET-Frameworks
C#	Objektorientierte Programmiersprache von Microsoft für .NET
CAML	XML Dialekt von Microsoft für die Nutzung der Funktionen des SharePoints
DLL-Datei	Dynamic Link Library, Programmbibliothek
IIS	Internet Information Server, Webserver für Windows
Namespace	Namensraum, ermöglicht eine Grobstrukturierung der Klassen
PC	Personal Computer
SDK	Software Development Kit, Entwickler Dokumentationen & Tools
SOAP	Netzwerkprotokoll für den Datenaustausch
SP	Service Pack
SPS	SharePoint-Site, eine Seite auf dem SharePoint
URL	Uniform Resource Locator, einheitlicher Ortsangeber für Ressourcen
Visual Studio	Entwicklungsumgebung von Microsoft
Webservice	Stellt Funktionen eines bestimmten Dienstes über eine Schnittstelle zur Verfügung
Windows 2003 Server	Serverbetriebssystem von Microsoft
WSDL	Webservice-Description-Language, beschreibt einen Webservice mit seinen Methoden und Klassen
XML	Extensible Markup Language, erweiterbare Auszeichnungssprache