

# *nhanced* **catalog**

## **Dokumentation:**

Konzeption und Implementierung eines Systems zur digitalen Umsetzung von PDF-Katalog-Seiten als EPAPER-Katalog in Java/Java-Applet

**Sommersemester 2007**

### **Betreuer:**

Prof.Dr.-Ing. Oliver Kretzschmar

### **Vorgelegt von:**

Carina Dubb (MIB 4, cd013)

Sebastian Holder (MIB 4, sh092)

Georgi Madjarov (MIB 4, gm013)

Adrian Staudt (MIB 4, as097)

Stuttgart, im Juli 2007

# Inhaltsverzeichnis

1	Einleitung.....	2
2	Motivation .....	2
3	Projektdefinition .....	3
3.1	Projektziel .....	3
3.1.1	Zielgruppe.....	3
3.2	Anforderungen .....	4
3.3	Projektumfang .....	5
4	Konzeption .....	6
4.1	Machbarkeitsstudie, Funktionalität .....	6
4.2	Technologien.....	7
5	Entwicklung.....	8
5.1	Architekturmuster.....	8
5.1.1	Admin-Tool .....	8
5.1.2	Customer-Applet .....	9
5.1.3	Querschnittsfunktionen/XML-Schnittstelle .....	9
5.2	Workflow und Realisierung der Konzeption.....	10
5.2.1	Admin-Tool .....	10
5.2.2	Customer-Tool .....	11
5.2.3	Querschnittsfunktionen/XML-Schnittstelle .....	12
5.3	Problematik.....	14
5.3.1	Admin-Tool .....	14
5.3.2	Customer-Tool .....	15
5.3.3	Querschnittsfunktionen/XML-Schnittstelle .....	16
5.4	Offene Punkte.....	18
5.4.1	Admin-Tool .....	18
5.4.2	Customer-Tool .....	18
5.4.3	Querschnittsfunktionen/XML-Schnittstelle .....	18
6	Fazit.....	19
A.	Anhang .....	20
A.1	Bedienung - Admin-Tool.....	20
A.2	Bedienung - Customer-Tool .....	21
A.3	Funktionalität - Admin-Tool .....	22
A.4	Funktionalität - Customer-Tool .....	24
B.	Anhang .....	27
B.1	Instanziierung der XML-Schnittstelle.....	27
B.2	Zusammenarbeit: Admin Applikation und XML-Schnittstelle.....	28
C.	Anhang - Protokollmeeting .....	29

## **1 Einleitung**

Das Software-Projekt „eCatalog“ definiert die Konzeption und die Implementierung eines Systems zur digitalen Umsetzung von PDF-Katalog-Seiten als EPAPER-Katalog mit Hilfe der Programmiersprache Java und der Umsetzung als Applet bzw. Applikation.

## **2 Motivation**

Ziel dieses Projekts ist die Erstellung eines Online-Einkaufsportals, basierend auf einem PDF-Katalog und dessen Anbindung an ein E-Commerce System.

Die Realisierung unseres Vorhabens soll das EPAPER Modell als Grundgerüst haben. Man soll einen virtuellen blätterbaren Online-Katalog erstellen, der mit dem hardcopy Original identisch ist.

Damit will man dem Kunden das Gefühl vermitteln, als ob er die Papierausgabe des Katalogs blättern würde.

## **3 Projektdefinition**

### **3.1 Projektziel**

- Das Geschäftsmodell hat folgende Ziele:
- Verkürzung der Zeiten (näher am Kunden)
- Flexibilität
- Kostensenkung
- Ständige Aktualität
- Immer verfügbar (24/7)
- Prozesse werden optimiert – „ONE CLICK AWAY“
- Crossmedia – falls ein Unternehmen mit seinen Produkten noch nicht im Web tätig ist

#### **3.1.1 Zielgruppe**

Als Zielgruppe für die eCatalog Software kommen alle Verlage in Frage, die die Bandbreite ihres Angebotsspektrums vergrößern wollen: Von Katalog-Zusammenfassung und -Herausgabe, bis zur Direktanbindung an ein eCommerce System.

eCatalog stellt den einfachsten Weg zu einem professionellen und kundenfreundlichen Internetauftritt mit der ganzen Produktpalette eines Unternehmens dar.

Auf Seiten der Nutzer (also der Käufer aus dem Katalog) besteht eine konkret umrissene Zielgruppe. Sie umfasst Personen, die mit dem Medium Internet und insbesondere der Online-Bestellung noch wenig Erfahrung haben und daher vor einem gewöhnlichen Online-Shop eher zurückschrecken. Hier bietet eCatalog einen leichten Übergang von der Bestellung aus gedruckten Katalogen zur Online-Abwicklung.

## **3.2 Anforderungen**

Unser „eCatalog“ System musste zwei Kriteriengruppen entsprechen.

### *Benutzerbezogene Anforderungen:*

- Usability
- Simple Handhabung selbst für nicht versierte User
- Durchgehende Prozesskette
- Übersichtlichkeit
- Ansprechendes Layout
- Direkter Zugriff zu einem Online-Shop

### *Technische Anforderungen:*

- Definition von Schnittstellen (BMEcat, osCommerce, eigene Schnittstellen)
- Auswahl der passenden Technologie (Java, Flash, etc)
- Webfähigkeit
- Persistenz
- Ausbaufähige Konzeptstruktur

### **3.3 Projektumfang**

Das Softwareprojekt hatte folgenden Umfang:

- Entwicklung eines Werkzeuges zur Definition von anklickbaren Artikel-Bereichen (z.B. Artikelbild + Bestellnummer + Artikelbeschreibung + Staffelpreis etc.) der PDF-Katalogseiten mit der Möglichkeit, den jeweiligen Artikelbereich einem Artikel-Datensatz eines Warenwirtschaftssystems zuzuordnen.
- Entwicklung eines Werkzeuges (JAVA-Applikation) zur Umsetzung einzelner PDF-Katalogseiten in eine webbasierte Präsentationsform inklusive suchbarer Metadaten sowie das Mapping der zuvor definierten anklickbaren Artikelbereiche zu interaktiven Elementen der webbasierten Präsentationsform.
- Entwicklung einer webbasierten Anwendung (JAVA-Applet) zur Nutzung der webbasierten Präsentationsform des Kataloges mit Funktionen zum Suchen, Blättern, Zoomen und Kennzeichnen eines markierten Artikels. Die Anwendung soll ergänzend ein Mini-Warenkorb-System beinhalten, in das gewählte Artikel übernommen werden können.
- Definition der XML-Schnittstelle zur Übergabe des befüllten Warenkorbes an ein E-Commerce-System, z.B. osCommerce, zur weiteren Bestellabwicklung.

## 4 Konzeption

Bei der Konzeption ließ sich „eCatalog“ in folgende Elemente aufteilen:

- Grobe Konzeption:
  - Machbarkeit
  - Funktionalität
  - Technologie-Suche
  - Schnittstellenplanung
  - Abstrahierung von Komponenten / Modul-Definition
- Feine Spezifikation:
  - Zwei-Modul-Ebene:
    - Customer-Tool
    - Admin-Tool
  - Einheitlicher XML Datenaustausch

### 4.1 Machbarkeitsstudie, Funktionalität

Zu Beginn der Konzeptionsphase machten wir uns Gedanken zur allgemeinen Machbarkeit des Projekts.

Viel Zeit wandten wir dafür auf, wie sich die Unterscheidung zwischen Artikeln und Produkten (zum Beispiel verschiedene T-Shirt Größen unter einer Artikel-ID) sinnvoll abbilden lässt. Wir mussten feststellen, dass diese Unterscheidung im BMEcat Standard nicht direkt gegeben ist.<sup>1</sup>

Wir entschieden uns dafür, trotzdem beides anzubieten, so dass bei der Administration eine Gruppierung in Produkte vorgenommen werden kann.

---

<sup>1</sup> Sowohl „PRODUCT“ als auch „ARTICLE“ existieren als Elemente in BMEcat – sie werden allerdings nicht unterschieden. Die verschiedenen Elementnamen stammen aus unterschiedlichen Versionen des Standards wobei „ARTICLE“ durch „PRODUCT“ ersetzt wurde.

Eine indirekte Möglichkeit der Schaffung von Produkten in oben beschriebenem Sinn besteht in der Verwendung von sogenannten „Catalog Groups“, die über „ARTICLE\_TO\_CATALOGGROUP\_MAP“ beziehungsweise „PRODUCT\_TO\_CATALOGGROUP\_MAP“ bei Erstellung der BMEcat XML gebildet werden können.

Die weiteren Schritte in der Konzeptionsphase waren nun das Entwickeln einer Struktur und Konzeption für die Speicherung der Daten. Das XML Format wurde dafür definiert und eine erste Klassenstruktur ausgearbeitet.

Eine detaillierte Diagramübersicht ist im Anhang B ersichtlich.

Parallel dazu begann die Planung zu den beiden Programmoberflächen.

Es sollte eine Administrations-Oberfläche geben (nachfolgend Admin-Tool genannt) und eine Oberfläche, die dem Kunden im Internet zur Verfügung steht (nachfolgend Customer-Tool genannt).

Das Admin-Tool musste die einzelnen Seiten darstellen sowie die XML-Inhalte visualisieren. Als Gerüst für die XML Datenstruktur wurde das BMECAT-Austauschformat genommen.

Das Customer-Tool sollte die einzelnen Seiten darstellen, sowie einen Warenkorb und eine Suchmöglichkeit beinhalten. Zudem sollte eine Anbindung zu osCommerce, ein Open-Source-Onlineshop-System, bestehen.

## **4.2 Technologien**

In unserem Projekt kamen mehrere Technologien zum Einsatz. Eine Anforderung war die Umsetzung des Projekts in der Programmiersprache Java. Mit dem eCatalog haben wir eine breite Palette an Implementierungstechniken abgedeckt, von Java Applets über Java Applications, Swing und SWT GUI Oberflächen, bis hin zu XML Spezifikationen und PHP.

Ein Hauptpunkt in der Technologieauswahl war die notwendige und aktuelle ePaper – Technik, die das Projekt prägte. Das heißt, dass Printprodukte auf dem Computermonitor dargestellt werden.



## **5 Entwicklung**

### **5.1 Architekturmuster**

Unser Softwareprojekt wurde in drei wesentliche Elemente unterteilt, die der Funktionalität und den Einsatzbereichen der einzelnen Projektmodule und der dazugehörigen Schnittstellen entsprachen.

#### **5.1.1 Admin-Tool**

Mit diesem Tool sollen folgende administrative Vorbereitungen für die spätere kundenseitige Visualisierung durchgeführt werden.

- Umwandlung des vorliegenden PDF-Katalogs in entsprechende Einzelseiten-JPGs mit drei unterschiedlichen Qualitätsstufen für die Zoomfunktion, sowie Erstellung von Thumbnails.
- Visualisierung der mitgelieferten BMEcat XML
- Grafische Markierung von Produkten/Artikeln für die spätere Anzeige im Customer-Tool.
- Zuweisung von Artikel-IDs auf die entsprechenden grafischen Markierungen (in Form von Polygonen) für die spätere Bestellmöglichkeit im Customer-Tool.
- Speicherung der vorgenommenen administrativen Einstellungen in einer eCat XML.
- Übergabe aller Parameter an das Customer-Applet.

### **5.1.2 Customer-Applet**

Die Kundenoberfläche stellt einen professionellen Internetauftritt dar mit direkter Anbindung an ein angeschlossenes E-Commerce System und verfügt über folgende Funktionalität:

- Webbasierte Darstellung des PDF-Katalogs nach der ePaper-Technik
- Auswahl von Artikeln mit direktem Verweis zu einem Warenkorbsystem
- Zoombarkeit der einzelnen Katalogseiten in drei Qualitätsstufen
- Bestellbarkeit der markierten Artikel
- Suchfunktionalität innerhalb des Katalogs nach Schlagwörtern und Artikelnummern
- Leichte und native Bedienbarkeit

### **5.1.3 Querschnittsfunktionen/XML-Schnittstelle**

Als Grundlage für die zwei beschriebenen Tools sind sehr ähnliche Funktionalitäten erforderlich. Daher existieren dafür einige Klassen zur einheitlichen Nutzung (Querschnittsfunktionen).

Im Einzelnen erfüllen diese die folgenden Funktionalitäten:

- Verwaltung von Meta-Informationen zum Gesamt-Katalog
- Verwaltung von Meta-Informationen zu den einzelnen Artikeln
- Zuordnung: Artikel zu Seiten
- Zuordnung: Polygone auf einer Seite zu Artikeln
- Kombinieren von Artikeln zu Produkten
- Suchfunktionalität

## **5.2 Workflow und Realisierung der Konzeption**

### **5.2.1 Admin-Tool**

Die Grundlage für ein eCat-Projekt bilden ein Druck-PDF eines Kataloges sowie die dazugehörige BMEcat XML (ein standardisiertes Austauschformat für Katalogdaten).

Beim Anlegen eines neuen Projekts wird eine entsprechende Ordnerstruktur mit der dazugehörigen Unterteilung der einzelnen Projektdateien erzeugt.

Aus dem PDF werden Einzelseiten-JPGs in verschiedenen Qualitätsstufen generiert. Dazu wird Ghostscript verwendet, ein kostenloser Interpreter für PDF und Postscript. Mit Ghostscript werden die Bilder der höchsten Qualitätsstufe (high-JPGs) generiert. Anschließend werden die restlichen JPGs in den Qualitätsstufen medium und low mit der Java-Library JMagick (ImageMagick, kostenloses Tool zur Bildanpassung) aus den high-JPGs erstellt.

Das GUI wurde mit dem Swing-Framework Jigloo von Cloudgarden erzeugt. Es erleichterte die genaue Positionierung der einzelnen Komponenten und somit konnte der Schwerpunkt bei der Gestaltung des Admin-Tools auf Usability gesetzt werden. Das Admin Tool besteht aus zusammenhängenden übersichtlichen Blöcken, die dem Benutzer helfen, eine schnelle Übersicht über das Programm zu bekommen und eine native Bedienung zu erreichen. Durch Tooltips bei den einzelnen Komponenten werden dem Anwender mit kleinen Hilfefenstern weitere Informationen zu dem entsprechenden Objekt angezeigt.

Auf der linken Seite des Admin-Tools werden die einzelnen Seiten des Katalogs mittels der erzeugten JPGs in einem JPanel dargestellt. Auf diese JPGs können nun in einer neuen Ebene produkt- beziehungsweise artikelpolygone gezeichnet werden.

Die BMEcat XML wird auf der rechten Seite mit Hilfe eines JAVA-JTrees grafisch abgebildet.

Bei der Zuweisung von Artikelnummer zu gezeichnetem Polygon werden im Hintergrund diese Informationen in einem DOM-Baum des eCat XMLs gespeichert. Beim Speichern des Projekts wird der DOM-Baum serialisiert und die eCat XML Datei in dem aktuellen Projektordner erstellt bzw. falls schon vorhanden, entsprechend aktualisiert.

### **5.2.2 Customer-Tool**

Da unser potenzieller Kunde schnellen Zugriff wünscht, war es unzumutbar diesem den Download einer Java-Anwendung anzubieten. Um dennoch alle Vorteile Javas auszunutzen verblieb lediglich die Applet-Variante zur Auswahl.

Zur Darstellung wurde ausschließlich Swing verwendet, sowie ein eigenes Button-Set, das dynamisch bei Bedarf geladen wird. Das Customer-Tool ist gut sichtbar in logische Module aufgeteilt, sodass der Benutzer für das Wesentliche aufnahmefähig bleibt.

Beim Laden des Applets wird die Katalogeinstiegsseite abgebildet und aus der eCat XML werden die Artikelkoordinaten ausgelesen und als bestellbare Einheiten dargestellt.

Die logischen Einheiten sind visuell gut getrennt, technisch gesehen aber eng miteinander verzahnt. Um diese Trennung im Quellcode organisieren zu können wurde das Observer-Pattern verwendet.

Beim Zoomen wird interaktiv die zugehörige Bilddatei nachgeladen und die Artikelkoordinaten neu berechnet. Die Netzwerkverbindung wird nur bei Bedarf verwendet. Ein eigener Cache sorgt für die nötige Performance beim Laden der kürzlich aufgerufenen Seiten.

Die Suche greift auf die XML-Schnittstelle (vgl. 5.2.3) zu und stellt grafisch das Ergebnis entsprechend dar.

Die im Warenkorb abgelegten Artikel werden in osCommerce exportiert. Dieses E-Commerce-System dient der Abschließung des Kaufverhältnisses und klärt die Zahlungsmodalitäten ab. Um osCommerce betreiben zu können bedarf es eines Webservers und einer MySQL Datenbank.

Da das Importieren eines Warenkorbes von osCommerce nicht unterstützt wird, musste ein externes PHP-Script die osCommerce-Session manipulieren um das Vorhaben erfolgreich abzuschließen.

Bevor das Applet genutzt werden kann, muss der Administrator alle benötigten Daten auf einen Webserver hochladen. Dies umfasst das Übertragen der Ordnerstruktur die vom Admin-Tool erstellt wurde. Hierbei werden die JPEG-Dateien in ihren unterschiedlichen Qualitätsstufen sowie die eCat XML (vgl. 5.2.3) auf einem Webserver übertragen.

Die html-Seite, die das Applet einbettet muss zunächst noch angepasst werden:

```
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>eCatalog - Ihr neues Katalogmodell</title>
</head>
<body>
  <applet code="ecatalog.customer.GUISupervisor.class" width="850" height="680">
    <param name="catalogid" value="Erstein">
    <param name="resourcepath" value="http://localhost">
  </applet>
</body>
</html>
```

Diese Parameter definieren welcher Katalog verwendet wird, und wo es sich befindet:

```
<param name="..." value="...">
```

### 5.2.3 Querschnittsfunktionen/XML-Schnittstelle

Hauptsächlich geht es bei den Querschnittsfunktionen um das Zuordnen, Abspeichern und Auslesen von Meta-Informationen (vgl. 5.1.3)

Die grundlegenden Daten aus denen auch der PDF-Katalog erzeugt wurde, liegen im BMEcat Format als XML Datei vor. Zur Arbeit mit dem Admin Tool ist es notwendig, diese Daten auslesen zu können. Für das Kunden-Applet hingegen soll die BMEcat XML nicht verwendet werden, da sie für diesen Zweck nicht genügend Informationen enthält (zum Beispiel Zuordnung Artikel zu Seiten), stattdessen aber andere Informationen, die für diese Anwendung gar nicht notwendig sind.

Daher wird während der Arbeit mit dem Admin Tool eine eigene XML Datei (eCat XML) erzeugt, die alle relevanten Informationen beinhaltet. Es werden allgemeine Daten des Katalogs gespeichert sowie Daten über die einzelnen Seiten mit ihren Artikeln, Produkten und jeweils zugeordneten Polygonen.

Für die Arbeit mit den XML Dateien wird die Apache Xerces DOM Implementation verwendet. Diese bietet relativ einfache Möglichkeiten zur Arbeit mit der speicherinternen baumartigen Darstellung des XML Dokuments. Zusätzlich existiert die Möglichkeit, Abfragen über den DOM Baum mit Hilfe von XPath zu machen.

Neben der Möglichkeit, Informationen abzuspeichern und Auszulesen, wird außerdem von beiden Programmen eine Suchfunktionalität benötigt. Diese muss jedoch über unterschiedliche Dateien suchen und auch andere Ergebnismengen zurückliefern. Konkret darf die Suche im Customer Applet nur solche Artikel zurück liefern, die auch tatsächlich bestellt werden können, und muss außerdem zusätzlich Angaben darüber enthalten, auf welcher Katalogseite der Artikel gefunden wurde. Da zum Zeitpunkt der Arbeit mit dem Admin-Tool die eCat XML Datei noch nicht (vollständig) existiert, erfolgt die Suche hier über die komplette BMEcat Datei und liefert gefundene Artikelnummern zurück. Für die Suche im Customer-Applet hingegen wird (während der Bearbeitung des Projekts im Admin-Tool) eine spezielle Suchdatei (eDescrip XML) erstellt, über die eine optimierte Suche möglich ist.

## **5.3 Problematik**

### **5.3.1 Admin-Tool**

Um das Print-PDF auszulesen und JPGs zu erstellen war die erste Idee, PDF-Box, eine Open Source JAVA-PDF-library, zu verwenden. Allerdings entsprachen die Ergebnisse beim Generieren der JPGs nicht den Qualitätsanforderungen: es gab Probleme mit Bildern, dem CMYK-Farbraum sowie Schriften, die fehlerhaft dargestellt wurden. Mit Ghostscript waren die Ergebnisse schließlich zufrieden stellend. Allerdings war mit Ghostscript keine Möglichkeit der Skalierung zu finden, deshalb musste schließlich ImageMagick eingesetzt werden, mit dem die JPGs entsprechend verändert werden konnten.

Um die Abfrage später in den Tools zu vereinfachen, wurde beschlossen, die Nummerierung der erstellten JPGs bei 1 statt 001 zu beginnen. Da nun Ghostscript erst die high-JPGs erstellt und ImageMagick über einen Konsolenaufruf den Inhalt des high-Ordners verarbeitet und nacheinander neu benannt, gab es das Problem, dass die Seiteninhalte ab den Zehner-Zahlen nicht mehr zur richtigen Seitenzahl passte. Die Lösung des Problems war die Nutzung der Java-Library von ImageMagick statt einem Konsolenaufruf. Die high-JPGs werden nun nacheinander in Schleife aufgerufen, umbenannt, entsprechend der Qualitätsstufe skaliert und in den zugehörigen Ordner kopiert.

Aufgrund der Tatsache, dass wir die Erstellung des Tools auf zwei Personen aufteilen mussten, wurde zu Beginn das Baukastenprinzip verwendet. Während das GUI entwickelt wurde, entstand der Teil mit der BMEcat XML-Darstellung separat. Dieser wurde so entwickelt, dass man ihn einfach an ein anderes GUI anhängen kann und so die Logik von der Darstellung getrennt wird. Entsprechend wurde mit dem Teil der JPG-Seitendarstellung verfahren.

Bei der Arbeit mit dem Tool, wenn man ein Projekt neu erstellt und es wieder geschlossen hat und anschließend gleich ein neues Projekt öffnen will, hat es das Problem gegeben, dass der heap-Speicher übergelaufen ist. Anschließend haben wir

herausgefunden, dass das Projekt nicht geschlossen werden darf, bevor die edescrip.xml nicht fertiggestellt ist (das dauert allerdings ein paar Minuten). Ist schon eine fertige edescrip.xml in dem aktuellen Projekt vorhanden, gibt es beim Öffnen und Schließen eines Projekts auch keine Probleme mehr.

Als großes Problem hat sich das Aufziehen von einzelnen grafischen Objekten (Polygone) erwiesen. Man musste diesbezüglich mehrere Abhängigkeiten beachten, die entscheidend für die Realisierung (Laden der JPGs, Zeichnen von Polygonen, Speichern der Objekte) waren. Als Lösung für dieses Problem wurde eine dreistufige Visualisierungsstruktur gewählt – In jeder Ebene wird entsprechender Inhalt geladen. (1. Ebene – weißer Hintergrund, 2. Ebene – JPG, 3. Ebene – BufferedImage Fläche für die Polygone). Diese komplexe Realisierung hat das Implementieren von recht simplen Funktionalitäten, wie Neuladen, Löschen, Vergrößern, Verkleinern, Wiederaufrufen, enorm erschwert, da man immer das entsprechende Objekt in der richtigen Ebene laden musste. Wichtig war auch, dass man schon bestehende Objekte aus den einzelnen grafischen Ebenen nicht löschen konnte, sondern vollständig überschreiben musste. Falls das nicht beachtet wurde, führte dies zu Verzerrungen der Objekte in dem JPanel.

Für das Aufziehen der Polygone wurden einzelne MouseListener verwendet, die die Unterscheidung zw. Produkten und Artikeln ausmachten. Die bedingte Abfrage dieser Listener war nicht einfach und man sah sich gezwungen, die Anzahl der Polygonpunkte zur Bildung der einzelnen grafischen Objekte, auf einen festen Wert (6 + 1) zu setzen.

Wenn nun ein Projekt geschlossen wird, soll auch das JPG, das auf das JPanel gezeichnet wurde, entfernt werden. Es konnte keine zufriedenstellende Lösung gefunden werden. Als annehmbare Lösung wird das JPG nun mit einem weißen Bild übermalt.

### **5.3.2 Customer-Tool**

Das größte Problem stellt der begrenzte Heapspeicher dar. Da es sich hierbei um ein Applet, also um das Starten eines Browser-Plugins handelt, gibt es auch keine Möglichkeit dem Applet einen großzügigeren Heapspeicher zuzuweisen. Der



standardmäßig zugewiesene Speicher muss ausreichen. Katalogseiten einer hochauflösten Ebene belegen auch mehr Speicher im RAM. Alle Seiten müssen bei jedem Aufruf neu geladen werden. Lediglich die geringste Qualitätsstufe wird gecacht.

Desweiteren wäre es geschickter gewesen, einmal geladene Bilder auf die Festplatte temporär zwischenspeichern, als diese erneut downloaden zu müssen. Da aber ein Applet seine Sandbox nicht verlassen darf, führt dieser weitere Punkt zu großen Latenzzeiten beim Laden von Katalogseiten. Diese Latenz merken vor allem Kunden mit geringer Bandbreite (z.B. ISDN).

Die einst kluge Wahl, auf mehreren Ebenen zu Zeichnen, entpuppte sich als Speicher- und Performanceproblem. Der ursprüngliche Gedanke war, dass nur die Layer neu gezeichnet werden müssen, in denen auch ein Update notwendig ist. Ändert sich nun zum Beispiel der Status eines Artikels nach „gehovert“, so wird dieses Polygon rötlich transparent gezeichnet. Die Idee war, dass nun nur die Polygon-Ebene neu gezeichnet wird. Hier spielte Java aber nicht mit, alle darunterliegenden Layer müssen ebenfalls explizit neu gezeichnet werden. Da addierende Transparenz in diesem Fall nicht gewünscht ist, muss zuvor die Katalogdoppelseite neu gezeichnet werden. Dies dauert so lange, dass das Nachzeichnen deutlich wahrnehmbar wird.

Anm.: Der Test, addierende Transparenz zuzulassen, bestätigte die Problemwahrnehmung. Bei der addierenden Transparenz muss nämlich die Katalogdoppelseite nicht neu gezeichnet werden. Diese Methode lief sehr performant.

### **5.3.3 Querschnittsfunktionen/XML-Schnittstelle**

Eine Herausforderung bei der Implementation der Querschnittsfunktionen war die Notwendigkeit, die jeweiligen Funktionalitäten für Admin-Tool und Customer-Applet bereitzustellen und dabei zu berücksichtigen, welche Informationen aus welchen XML Dateien zu welchem Zeitpunkt vorliegen, beziehungsweise benötigt oder erstellt werden. Zur Lösung des Problems sind die XML Handler Klassen in einer Vererbungsstruktur angelegt. Die Menge an Funktionen die vom Admin-Tool

benötigt wird ist eine Erweiterung dessen, was im Customer-Applet zur Verfügung steht.

Ein weiterer wichtiger Punkt war die Korrektheit der gespeicherten Daten. Es musste sichergestellt werden, dass am Ende der Arbeit mit dem Admin Tool eine korrekte eCat XML Datei das Resultat ist. Zu diesem Zweck wurden Prüffunktionen entwickelt, die bei fehlenden Daten entsprechende Exceptions werfen damit Korrekturen gemacht werden können.

Nach Implementierung sämtlicher XML handling Funktionalitäten war die Bereitstellung einer Suchmöglichkeit das verbleibende Problem. Sowohl für Admin-Tool als auch für das Customer-Applet werden XML Dateien durchsucht. Für jede Fundstelle muss die zugehörige Artikelnummer, sowie beim Customer-Applet zusätzlich die Seitenzahl, zurückgegeben werden. Diese Abfragen dauerten zunächst sehr lange – Grund war die unzureichende Geschwindigkeit bei der Auswertung von XPath Ausdrücken. Die manuelle Auswahl der Elemente durch DOM Befehle brachte die gewünschte Verbesserung.

## **5.4 Offene Punkte**

### **5.4.1 Admin-Tool**

- Usability der Dialoge verbessern
- Mehrere Projekte gleichzeitig öffnen
- In dem JPanel auch mit Maus navigieren anstatt nur mit Scrollbalken zu arbeiten
- Polygone einzeln markieren und löschen (geht bisher nur so, dass man in der XML die entsprechenden Tags entfernt)
- Polygone mit beliebiger Anzahl von Polygonpunkten aufziehen
- Ladeanimation, wenn JPG geladen oder geschlossen wird
- Automatischer FTP-Upload des Projekts für das Customer-Tool

### **5.4.2 Customer-Tool**

- Blätteranimation
- Anzeige von Produkten
- Unterstützung zur Artikelauswahl beim Bestellwunsch eines Produktes
- Druckmöglichkeit von Einzel- oder Doppelseiten
- Anzeige von Miniaturansichten der einzelnen Katalogseiten im Überblick. Somit wird ein gezieltes Themenfeld objekt-visuell schneller erfasst.

### **5.4.3 Querschnittsfunktionen/XML-Schnittstelle**

- Die Indizierung zur Bereitstellung von Suchfunktionalität im Kunden Applet sollte irrelevante Worte wie zum Beispiel, „der“, „die“, „das“, „und“, ... nicht mit indizieren.
- Geschwindigkeitsoptimierungen (zu viele XPath abfragen werden langsam)
- Skalierungsfähigkeit der XML Verarbeitung mit DOM müsste für einen Einsatz mit beliebigen Katalogdateien überprüft werden. Eventuell muss für das Lesen der BMEcat XML ein ressourcenschonender Parser verwendet werden.

## **6 Fazit**

Unser Team ist froh, dass es das von Prof. Kretzschmar in Auftrag gestellte Projekt erfolgreich abschließen konnte.

Während dieser 14 Wochen wurde extrem viel Wissen gesammelt und die Fähigkeiten im Bereich der Teamarbeit enorm gesteigert.

Der Projektumfang wurde von Seiten des Teams am Anfang stark unterschätzt, dennoch war der gut strukturierte Projektzeitplan von entscheidender Hilfe bei der rechtzeitigen Fertigstellung des Projekts.

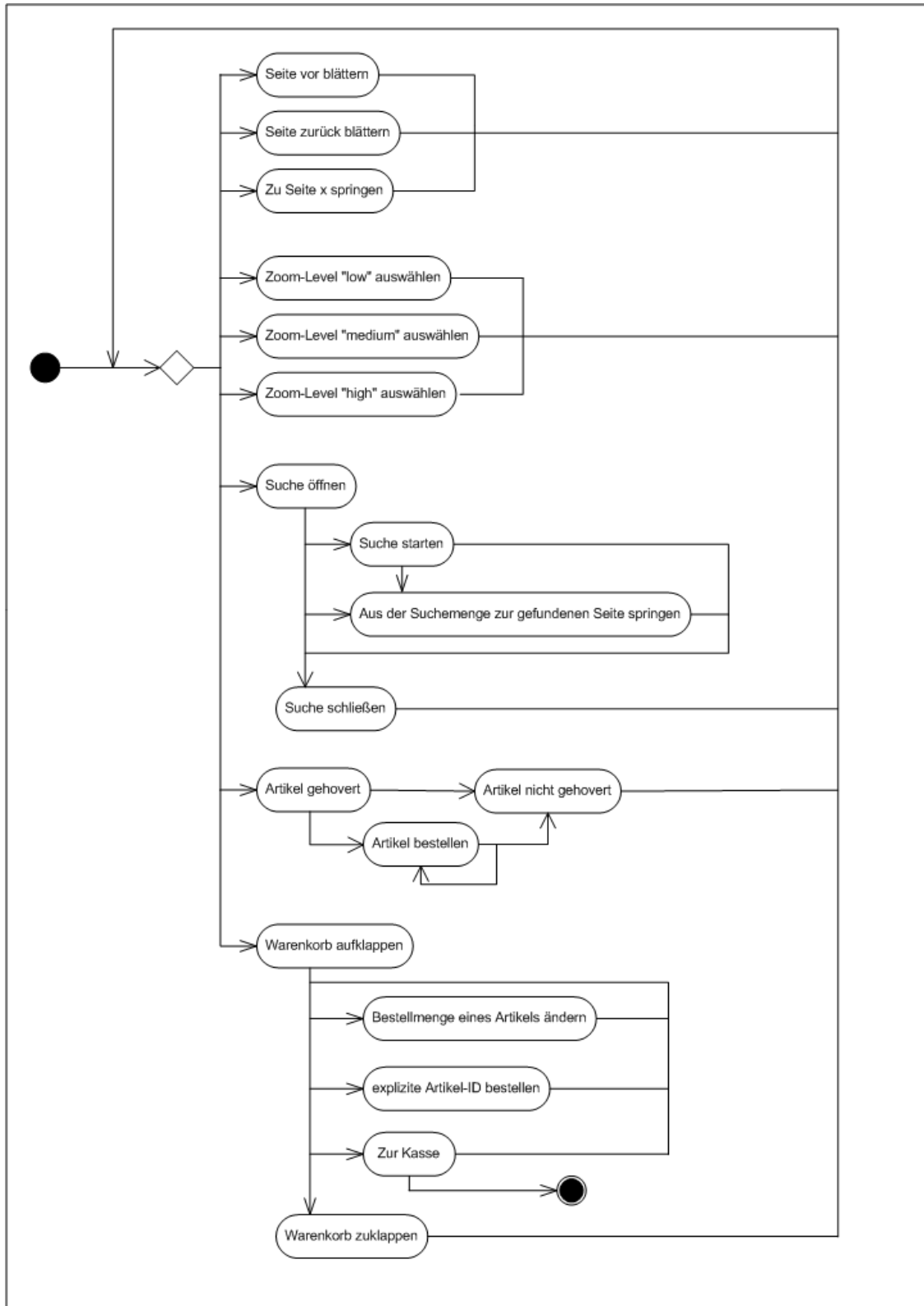
Eine wichtige Erkenntnis für das Team war, dass Theorie und Praxis oft stark voneinander abweichen können. Denn alle Planungen während der Konzeptionsphase sind später in den praktischen Umsetzungen von weiteren, ursprünglich nicht berücksichtigten Parametern abhängig und somit werden Korrekturen und sonstige Anpassungen erforderlich, um das Projektziel zu erreichen.

# A. Anhang

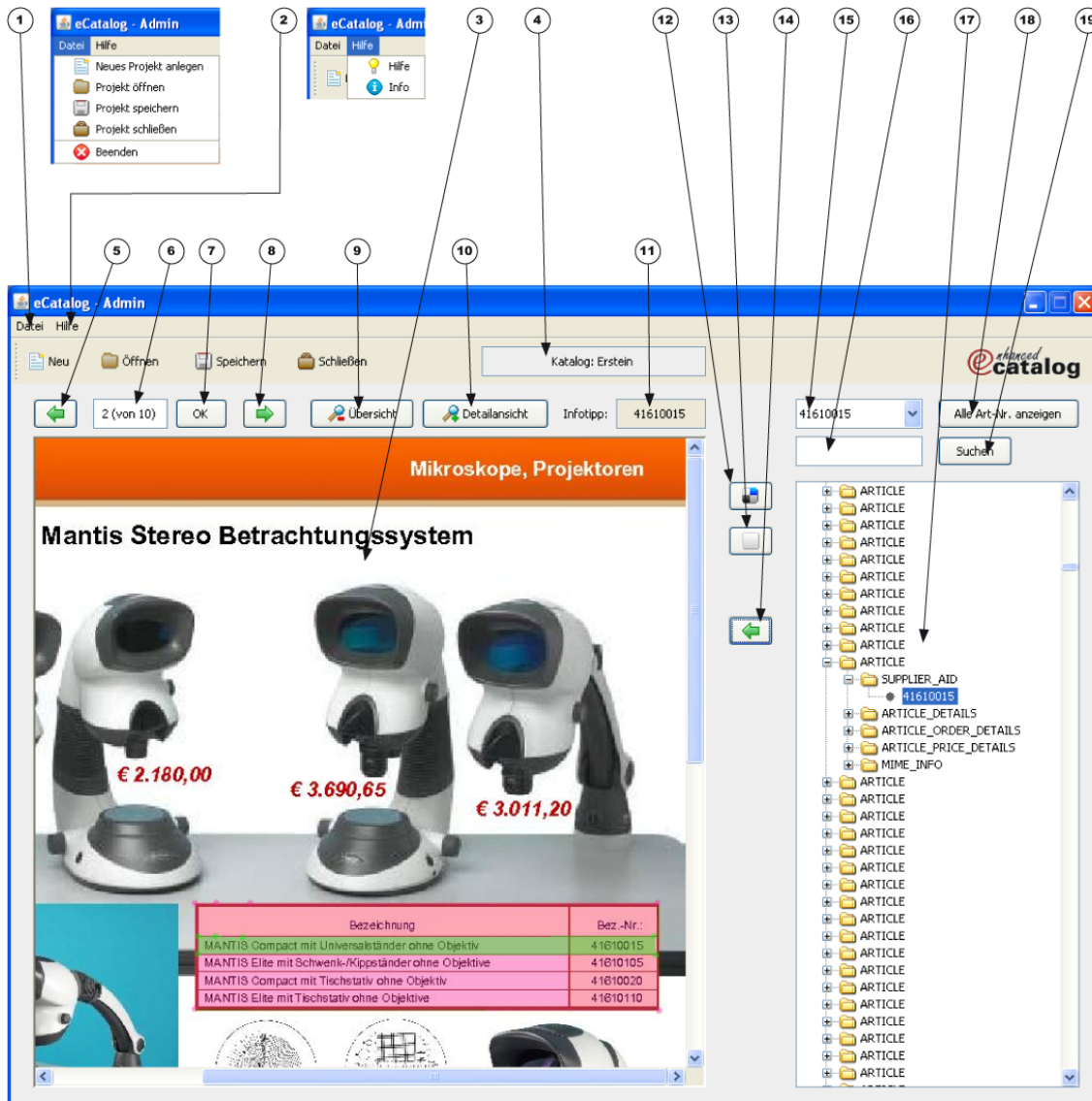
## A.1 Bedienung - Admin-Tool



## A.2 Bedienung - Customer-Tool



## A.3 Funktionalität - Admin-Tool



- ① Datei-Menü (Entsprechendes gilt für die Buttons in der Tool-Bar):
- „Neues Projekt anlegen“: ein Projekt wird erstellt inklusive generierter Projektstruktur und JPGs
  - „Projekt öffnen“: ein bestehendes Projekt wird geöffnet
  - „Projekt speichern“: alle Polygoninformationen (Koordinaten, Artikelnummer, Seitenzahl) werden in der eCat XML gespeichert
  - „Projekt schließen“: schließt das aktuelle Projekt
  - „Beenden“: die Admin-Applikation wird geschlossen

- ② Hilfe-Menü:
  - a. „Hilfe“: Informationen zum Vorgehen
  - b. „Info“: Informationen zu dem eCatalog-Programm
- ③ Darstellung der einzelnen Seiten
- ④ Name des Projekts
- ⑤ Eine Seite zurück blättern
- ⑥ Aktuelle Seite
- ⑦ Wenn eine aktuelle Seitennummer manuell eingegeben wird, bestätigt dieser Button die Eingabe und die entsprechende Seite wird dargestellt.
- ⑧ Eine Seite vorwärts blättern
- ⑨ Ganzseitenansicht
- ⑩ Vergrößerung der Seite
- ⑪ Anzeige der einem Polygon zugewiesenen Artikelnummer, wenn mit der Maus über ein Polygon gefahren wird
- ⑫ Ein rotes Produktpolygon zeichnen
- ⑬ Ein grünes Artikelpolygon zeichnen
- ⑭ Artikelnummer von XML dem aktuellen Polygon zuweisen
- ⑮ Darstellung vorhandener Artikelnummern aus dem BMEcat XML
- ⑯ Suchmöglichkeit nach einem bestimmten Begriff
- ⑰ Visualisierung der BMEcat XML in einem jTree
- ⑱ Alle Artikelnummern aus dem BMEcat XML in der Combo-Box (Punkt 15) auflisten
- ⑲ Mit diesem Button wird nach dem in Punkt 16 eingegebenen Begriff im XML gesucht und alle dazu gefundenen Artikelnummern in der Combo-Box (Punkt 15) aufgelistet.



## A.4 Funktionalität - Customer-Tool

The screenshot shows a web browser window displaying the ERSTEIN e-catalog. The main content area features a large product image of the 'Mantis Stereo Betrachtungssystem' with a price of € 2.190,00. Below it is a table of accessories and a detailed description. To the right, there is a grid of eyeglasses with various specifications and prices. A search dialog is open, showing the search term 'Bügel Schutzbrille' and the results 'Bügel Schutzbrille DIN EN 166 F' and 'Bügel Schutzbrille DIN EN 166'. At the bottom, there is a pagination bar with 'Zu Seite: 2' and 'OK' buttons. Numbered callouts 1-9 point to various UI elements: 1 points to the main product image, 2 to the product title, 3 to the product price, 4 to the product description, 5 to the product table, 6 to the product image, 7 to the search dialog, 8 to the search results, and 9 to the pagination bar.

### Hauptseite

- ① Die aktuellen JPEGs werden vom Server geladen und angezeigt. Mittels der gedrückten mittleren Maustaste kann in höheren Zoom-Stufen das Bild verschoben werden.
- ② Alle bestellbaren Artikel werden gelb hervorgehoben.
- ③ Gehovertete Artikel werden rot hervorgehoben.

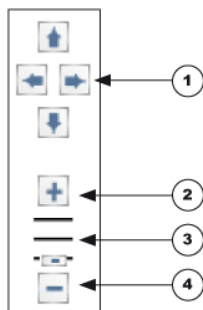
## ToolBar

- ④ Das Suchfenster wird geöffnet
- ⑤ Eine Seite zurückblättern
- ⑥ Zu beliebiger Seite x springen
- ⑦ Eine Seite vor blättern

## Suche

- ⑧ Suchbegriffsmaske
- ⑨ Suchmenge als Ergebnismenge.  
Mit einem Doppelklick auf ein Suchergebnis wird die zugehörige Seite geladen. Gefundene Artikel werden lila hervorgehoben.

## gMap-Navigation



- ① In höheren Zoom-Stufen lässt sich die Seite in alle Himmelsrichtungen bewegen.
- ② Die nächst höherer Zoom-Stufe anzeigen
- ③ Zoom-Stufe mittels des Reglers direkt auswählbar
- ④ Die nächst niedrigerer Zoom-Stufe anzeigen

The screenshot shows a web browser window with the URL 'http://localhost/'. The main content area displays the 'Mantis Stereo Betrachtungssystem' product page. A shopping cart is overlaid on the right side of the page, titled 'Warenkorb (3 Artikel, 7540,63 €)'. The cart contains the following items:

ArtikelNr	Bezeichnung	Menge	Preis
41610110	MANTIS Elite mit ...	2	7528,94 €
55640100	Schweißbrille a...	1	11,69 €

Below the cart, the total price is shown as 'Gesamtpreis: 7540,63 €' and a 'Zur Kasse' button is visible. A search bar at the bottom of the cart contains the article number '41610112' and an 'OK' button. Three numbered callouts (1, 2, 3) point to the quantity field, the search bar, and the 'Zur Kasse' button respectively.

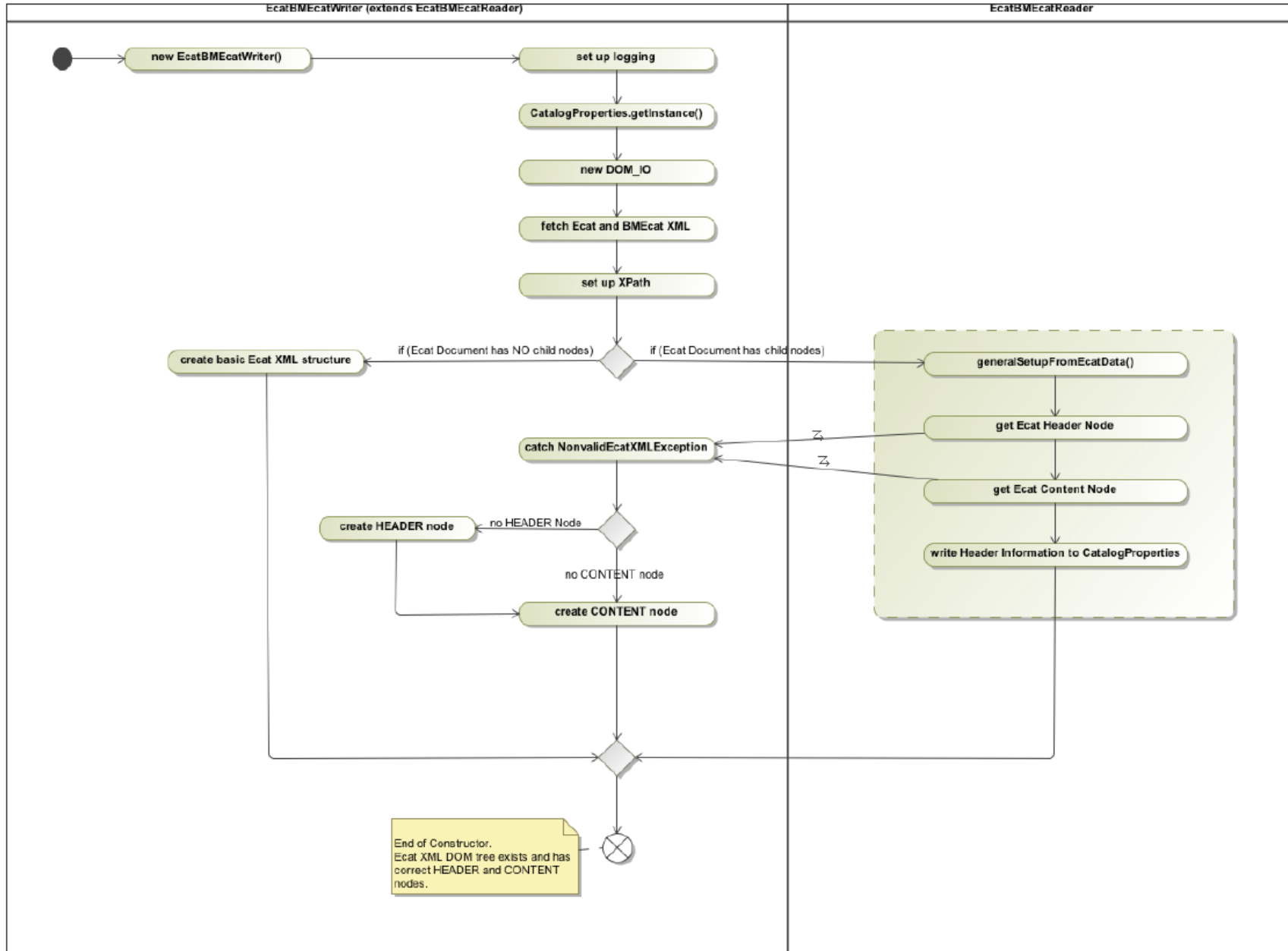
## Warenkorb

Mittels eines Doppelklicks auf einen bestellbaren (gelb hervorgehobene) Artikel, landet dieses in den Warenkorb.

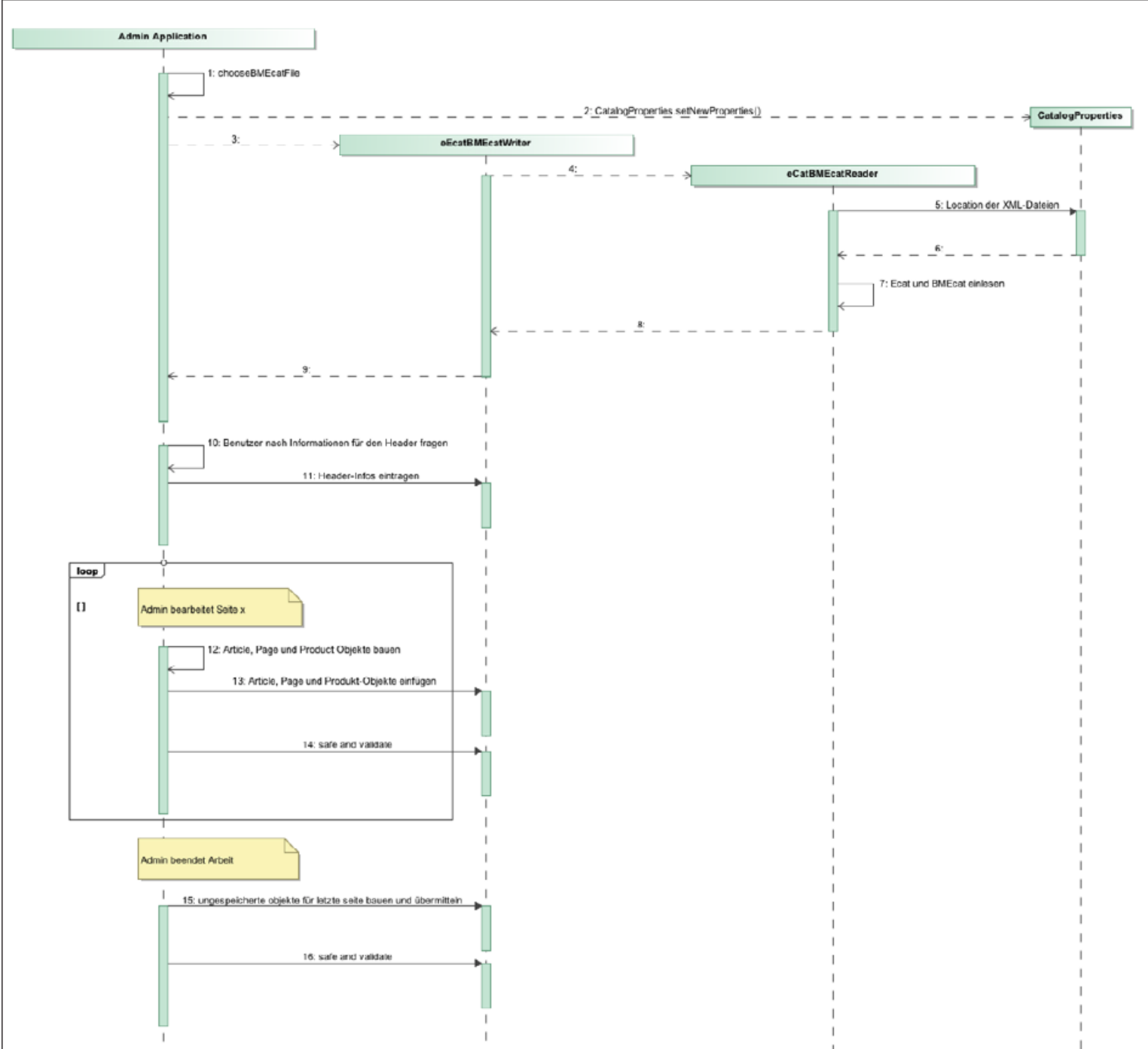
- ① Die Menge kann geändert werden
- ② Artikel können mittels ihrer ArtikelNr. in den Warenkorb gelegt werden
- ③ Warenkorb wird nach osCommerce exportiert, woraufhin die Bestellung abgeschlossen werden kann.

# B. Anhang

## B.1 Instanziierung der XML-Schnittstelle



## B.2 Zusammenarbeit: Admin Applikation und XML-Schnittstelle



Protokollführer: S. Holder		Termin: 20.03.07	Ort:	Dauer: 12.15-13.15 Uhr + 16.00-16.45 Uhr	
Anwesend: C. Dubb, S. Holder, G. Madjarov, A. Staudt					
Verteiler: dto.					
Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Aufgabenstellung	Allgemeines Diskutieren der Richtlinien und Punkte im Dokument "Projektmanagement im Rahmen der Software-Projekte im Studiengang Medieninformatik an der HdM Stuttgart" sowie der von Herrn Kretzschmar verfassten Aufgabenstellung	I		alle
2	Vorgehensweise	Diskutieren und Einteilen der Punkte unter "3.2 Team Kick-Off" (Richtlinien-Dokument). Ergebnisse daraus: - Grobe Festlegung Interessengebiete: <ul style="list-style-type: none"> <li>• Back-End (Konvertierung aus PDF, Administrations-Tool): C. Dubb, G. Madjarov</li> <li>• Front-End (Darstellung, Blätterfunktion, Zoom): A. Staudt, S. Holder</li> <li>• Schnittstellen zu osCommerce: A. Staudt</li> </ul> - Projektleiter: G. Madjarov, später eventuell Wechsel - Regelmäßige Treffen dienstags und des Weiteren nach Bedarf. - Termine für Treffen mit Herrn Kretzschmar sollen sich nach den Meilensteinen richten. - Verfassen der Protokolle erfolgt rotierend - Dateiaustausch über CVS (muss noch eingerichtet werden)	B		alle
3	Aufgabenverteilung	Diskussion über das weitere Vorgehen. Zuteilen weiterer Tätigkeiten an einzelne Mitarbeiter. - Darstellung des Business-Case: G. Madjarov - Zielvorstellungen FrontEnd: A. Staudt - Zielvorstellungen Administrations-Tool: C. Dubb	B		alle
4	Nächstes Meeting	Donnerstag, 22.03.07, 12 Uhr	B	22.03.07	alle
5	Offene ToDo's	- Einrichten CVS - Definition Schnittstellen - Definition Milestones	HB	22.03.07	alle

Protokollführer: G.Madjarov		Termin: 21.03.07	Ort:	Dauer: 12:00 bis 13:30 Uhr	
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt					
Verteiler: dto.					
Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Ziel /Business case	Endversion wurde von dem Team geprüft	I		Madjarov
2	Admin Portal	Konzeptvorstellung des Admin Portal	I		Dubb
3	Admin Portal	Diskussion über die Übernahme von der Original XML oder Erstellung einer eigenen	HB		Dubb
4		Technologieauswahl bei der Realisierung der Blätterungsfunktion: <ul style="list-style-type: none"> <li>▪ HTML (durch pdftohtml converter)</li> <li>▪ JPEG (mit 3 Zoom Stufen; Div Layer)</li> </ul>	I HB		
5		Visualisierung wird über JPEG realisiert(Positionierung und Auflösung über Koeffizienten)	B		
6		Problem erkannt: Übernahme/Konvertierung der Original –XML → konforme Datei(Abstrahierung der Details)	HB		
7	dotProjekt	Milestones im System einstellen, Protokollvorlage erstellen	A	Bis 27.03.07	Madjarov
8	Admin Portal	Konzept überarbeiten und abgeben ( im System einchecken)	A		Dubb
9	Nächstes Meeting	Verbliebene Fragen zu dem Admin Portal, Vorstellung des Kundenportals	I	27.03.07	
10					
11					

Protokollführer: C.Dubb	Termin: 27.03.07	Ort: HdM	Dauer: 12:15 bis 13:30 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Frontend	Konzept der Blätterfunktion und Warenkorb wurden vorgestellt (Zoom, Speichern, Drucken). Features: Übersichtsmöglichkeit aller Katalogseiten. Problem erkannt: - Seitenzahlanzeige muss mit der Seitenzahl im PDF übereinstimmen, . - zusätzliche Anzeigemöglichkeit bei Bestellvorgang, wenn von einem Produkt mehrere Varianten zu erhalten sind	I		Staudt
2	Admin Portal	Diskussion über die Admin-Oberfläche (keine Verwendung einer Datenbank sondern nur XML, Standard-XML soll aktualisiert werden können)	B		Dubb
3	dotProject	Probleme mit dotProjekt → Mail an Beate Schlitter	A		Madjarov
4	dotProject	Aktuelle Aufgaben in dotProject eintragen	A		Madjarov
5	Datenquelle	Woher bekommt das Team eine XML aus einem Warenwirtschaftssystem mit evtl. entsprechendem Katalog-PDF? → Mail an Herr Kretzschmar	A		Madjarov
6	Datenquelle	Eine mögliche Standard-XML mit evtl. entsprechender DTD erstellen	A	02.04.07	Holder
7	E-Commerce-System	Informationen über osCommerce sammeln	A	02.04.07	Staudt
8	Frontend	Informationen der Frontend-Konzeption in dotProject einstellen	A		Staudt
8	Zeitplan	Gedanken über Zeitplan machen	A	02.04.07	alle
9	Namen und Logo	Ideen für Namen und Logo sammeln	A	02.04.07	alle
10	Nächstes Meeting	osCommerce, Standard-XML, Zeitplan erstellen, Namen und Logo, Aufgabenverteilung	I	02.04.07	alle



Protokollführer: A. Staudt	Termin: 02.04.07	Ort: Café Stuttgart/Stadtmitte	Dauer: 16:15 bis 18:30 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Logo	Geschwungenes e, ähnlich dem @ zeichen. eCatalog, oder enhancedCatalog, evt. als Icon ein Buch in einem Monitor	A	10.04.07	Dubb
2	Namen	eCatalog	B		alle
3	Medianight	Kurzbeschreibung wurde für die Broschüre abgeseget	B		Madjarov
4		Anmeldung durchführen, und Nachbarstand beantragen (Doku 2 Projekt von Dubb und Madjarov) damit Laufwege gering sind	HB		Madjarov
5	Auftraggeber XML-Datei	Herr Kretzschmar hat keine XML-Datei parat, aber einen Ansprechpartner	I		Madjarov
6	SVN, Subclipse	SVN für die Versionierung benutzen, keine weitere Fremdsoftware	B		alle
7		Probleme bestehen beim Zugriff auf das SVN Repository → Mail an F. Schlitter	A	02.04.07	Madjarov
8	Zeitplan	Deadline für die Konzeptionsphase	B	10.04.07	Madjarov
9		Implementierungsphase 12. April bis 12. Juni	B	12.06.07	Madjarov
10		Abschätzung der benötigten Zeit für die einzelnen Module	B	10.04.07	alle
11	Aufgabenteilung	Kunden-Frontend	HB		Staudt, Holder
12		Warenkorb	HB		Staudt
13		Admin-Portal	HB		Dubb, Madjarov
14		Konvertierung der FremdXML zu unserem XML-Format	HB		Dubb, Madjarov

15	eCatalog-XML-Datei	Einigung auf grundlegende XML-Struktur	B		Holder
16		Die Struktur unseres XML-Formates wird granularer definiert Risiko: Das ganze Projekt beruht auf eine allgemein gültige XML-Datei. Nachträgliche notwendige Erweiterungen können das Projekt gefährden.	A	03.04.07	Holder
17		XML-Datei um Inhaltsverzeichnis erweitern (Allgemein: Verweise auf andere Seiten)	A	03.04.07	Holder
18	Kunden-Portal	Indizierung für Suchfunktion geklärt	B		Holder
19	Nächstes Meeting	Standard-XML, Unterlagen für H. Kretschmar, Beschluss über Logo und Name	I	03.04.07	alle
20	Übernächstes Meeting	osCommerce, Schnittstellen definieren, Klassenhierarchie, Standart-XML verabschieden	I	10.04.07	alle

Protokollführer: G.Madjarov	Termin: 03.04.07	Ort:	Dauer: 11:30 bis 13:30 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	XML Konzept	Konzept für die XML Datei wurde vorgestellt und Beispiele wurden eingebaut	I		Holder
2	LOGO	Das Projektlogo würde vorgestellt und eine Version wurde am 04.04.07 auch ausgesucht.	B		Dubb
3	XML Beispiele	Es wurden 4-5 verschiedene Versionen von dem XML Konzept erstellt. Als Endergebnis wurde eine Gesamtlösung von bestimmten Versionen zusammengebaut.	I		Dubb, Holder, Madjarov
4	Problematik	Als Problem wurde die Abhängigkeit zw. den Auswahlelementen entdeckt...(price, colour, etc)	BH		Dubb, Madjarov, Holder
5	Medianight Anmeldung	Anmeldung wurde gemacht und Logo wurde hochgeladen	I		Madjarov
6	BMECat	Email von Prof. Kretschmar bezüglich Standart...(weitergeleitet)	I, BH		Madjarov

Protokollführer: C. Dubb	Termin: 10.04.07	Ort: HdM	Dauer: 11:30 bis 13:00 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Besprechung mit Herr Kretzschmar	Mail an Herr Kretzschmar mit Verweis auf die Dokumente in dotProject bzw. oder die Frage, ob spezielle Dokumente für ihn angefertigt werden sollen	A	11.04.07	Madjarov
2	MediaNight	Projekt ist angemeldet	I		Madjarov
3	OS-Commerce	Recherchen: - riesige openSource-Software – ein komplettes Shopsystem (nicht nur reine Abwicklung); - unklar, ob XML-Schnittstelle frei Haus dabei ist, - wir brauchen eigentlich nur Abwicklung im Hintergrund → ist das möglich?	I		Staudt
5	OS-Commerce	Recherchen zusammenfassen und dokumentieren	A	11.04.07	Staudt
6	BMEcat	Es kam von Herr Kretzschmar die Information, dass wir als Input XMLs mit BMEcat-Basis. Die Frage steht im Raum, ob trotz der Standard-XML eine eigene XML erstellt werden soll, die nach Seiten sortiert ist	HB	17.04.07	alle
7	BMEcat	Analyse der Dokumentation	A	17.04.07	Holder
8	Admin-Oberfläche	Ist es möglich, automatisch aus dem PDF die Bestellnummer eines Artikels und dazu noch die Seite auszugeben, auf welcher der Artikel vorkommt? (PDFBox?)	A	17.04.07	Dubb
9	Interfaces	Alle machen sich Gedanken über notwendige Interfaces	A	17.04.07	alle
10	Interfaces	Programmieren	A	24.04.07	Staudt, Holder
11	Meeting	Nächstes Meeting am 17.04.2007 in der HdM um 12.00 Uhr	I	17.04.07	alle

Protokollführer: A. Staudt	Termin: 17.04.07	Ort: Bibliothek/HdM Stuttgart	Dauer: 12:20 bis 13:30 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	BMECat	Erklärung der DTD und der XML-Datei Zusätzlich 3DTDs für neue Kataloge	A	10.04.07	Holder
2		Problem: Keine Überordnung von Produkten und Artikel Es gibt lediglich Artikel, BMECat sieht hierfür keine Spezifizierung	BH		alle
2	eCat	Einführung einer eigenen eCat Datei, die für uns zusätzlich wichtige Attribute enthält unter anderem die Elemente, <page><artikel><coordinates>...	B		alle
3	PDF2Image	Verwendung von pdfBox da diese eine Java-Bibliothek anbietet um aus PDF Seiten JPEG Seiten zu generieren	B		Dubb
4	Suche	Die Suche wird über die XML-Datei ausgeführt	HB		alle
5	Interfaces	Interfaces als Grundgerüst für weitere Implantation entwerfen	HB	30.04.07	alle
6	Nächstes Meeting	Interfaces besprechen Evt. Lösung für das Artikel – Produkt Problem finden	I	24.04.07	alle

Protokollführer: S.Holder	Termin: 24.04.07	Ort: HdM	Dauer: 12:00 bis 13:00 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Allgemeine util Klassen	Vorstellung der allgemeinen Utility Klassen (Verarbeitung von XML und JPEG): Bisherige Ergebnisse, weitere Planungen	I		Staudt, Holder
2	Eclipse / Code-Verwaltung	Nutzung von Subversion funktioniert; Zeichnen der UML Diagramme mit Together	I, B		alle
3	eCat interne xml	Vorstellung eCat DTD und Beispiel XML-Datei	I		Holder
4	eCat interne xml	Erweitern der eCat DTD so dass auch Polygone abgespeichert werden können	A	27.04.07	Holder
5	Namenskonvention eCat	Die aus BMEcat erzeugte eCat soll den selben Namen mit dem Zusatz „_ecat“ erhalten (z. B. „Katalogname_ecat.xml“)	B		alle
6	Namenskonvention Java-Klassen	package: de.hdm-stuttgart.mi.projekte.eCatalog	B		alle
7	Unterscheidung Produkt/Artikel	Zusammenfassen von Artikeln zu Produkten in BMEcat nicht einfach möglich. In wie weit können wir die Unterscheidung Produkt/Artikel trotzdem realisieren?	I		alle
8	Unterscheidung Produkt/Artikel	weiteres Vorgehen, erarbeiten von Lösungsmöglichkeiten	A		Dubb, Madjarov
9	Meeting	Nächste Woche kein Meeting (1. Mai) weiteres Meeting per eMail vereinbaren	B		alle

Protokollführer: G. Madjarov	Termin: 15.05.07	Ort: HdM	Dauer: 12:00 bis 14:15 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Projektstand	Klassen und bisherigen Implementierungen werden vorgestellt Logger-Werk, Test-Werk, Exceptions, UML	I		Holder
2	Projektmanagement	Draft erstellen	A	15.06.07	Madjarov
3	Konzeption	Diskussion über die Zuordnung und Implementierung von Produkt-Artikel	B		Team
4	Admin	XML Baum, und Filter + PDF	A, I		Dubb
5	eCatXML	Erklärung und Erläuterung des Konzepts	I		Holder
6	GUI	Konzeption, Design, Implementierung Buttons -> Staudt	I, A		Madjarov

Protokollführer: C.Dubb	Termin: 22.05.07	Ort: HdM	Dauer: 12:15 bis 13:30 Uhr
Anwesend: C.Dubb, S.Holder, G.Madjarov, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Subversoin	Lib-Ordner erzeugen, in den alle .jar-Dateien zu finden sind	A		Dubb
2	Admin Portal	Vorstellung des Admin-GUIs	I		Madjarov
3	Admin Portal	Kein PDF/XML-Button: beim Anlegen eines neuen Projekts muss man das Verzeichnis definieren, sowie anschließend gleich die PDF-Datei und die XML-Datei auswählen. PDF und XML werden direkt in das ausgewählte Verzeichnis kopiert und dort werden die Unterordner mit den JPGs erstellt. ECat-Catalog-ID entspricht dann dem Verzeichnisnamen.	B		Dubb
4	Admin Portal	Statusbalken für das Erstellen der JPGs einbauen	A		Dubb
5	Admin Portal	Fenstergröße erst einmal fix, als Feature veränderbar	B		Madjarov
6	Admin Portal	Auflösungen für unterschiedliche Katalogformate anlegen (berechnen) → erstmal Feature	B		Dubb
7	Admin Portal/Frontend	Methode erstellen, die 4 Dimension-Werte als Eingabe erwartet (Pixelgrößen von Low/Medium/High/Thumbnails) um diese in die ECat-XML zu schreiben.	A		Holder
8					
8					
9					
10					



Protokollführer: C.Dubb	Termin:05.06.07	Ort: HdM	Dauer: 11:45 bis 13:15 Uhr
Anwesend: C.Dubb, S.Holder, A.Staudt			
Verteiler: dto.			

Nr.	Thema	Vorgang / Festlegung	Art	Termin	Verant.
1	Allgemein	Applikationssprache wird Deutsch sein	B		alle
2	Allgemein	CatalogProperties ändern -> sie müssen überschrieben werden dürfen, denn wenn ein neues Projekt im Gui geladen wird, müssen diese an die neuen Dateien angepasst werden	A		Holder
3	Admin Portal	Ordnerstruktur ändern: anpassen wie auf dem Customer-Server	A		Dubb
4	Admin Portal	Alle ausführbaren Klassen angeschaut	I		alle
5	Customer Portal	Alle ausführbaren Klassen angeschaut	I		alle