



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Mehdi Afridi

Entwicklung eines webbasierten IPC/IPL Network  
Project Management System für die Firma  
AEROTEC

Mehdi Afridi

Entwicklung eines webbasierten IPC/IPL Network  
Project Management System für die Firma  
AEROTEC

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Bettina Buth  
Zweitgutachter : Prof. Dr. rer. nat. Wolfgang Gerken

Abgegeben am 8. Januar 2009

**Mehdi Afridi**

**Thema der Bachelorarbeit**

Entwicklung eines webbasierten IPC/IPL Network Project Management System für die Firma AEROTEC

**Stichworte**

Web-Anwendung, JBoss, Java EE, EJB, JSF, CMS, ECM Reporting, Testen

**Kurzzusammenfassung**

Diese Arbeit beschäftigt sich mit der Entwicklung eines webbasierten Systems. Dazu werden in einer Analysephase Gründe für eine Neuentwicklung zusammengetragen und eine Marktübersicht bereits existierender Systeme erstellt. Dabei soll unter Berücksichtigung aller Anforderungen auf prototypischer Weise ein System entwickelt werden, die die Anforderungen des Altsystem und die Anforderungen des neuen Systems bereitstellt.

**Mehdi Afridi**

**Title of the paper**

Development of a IPC/IPL Network Project Management System solution for the company AEROTEC

**Keywords**

Web application, JBoss, Java EE, EJB, JSF, CMS, ECM, Reporting, Testing

**Abstract**

This thesis is about the further development of a webbased system. For that purpose, reasons for a new development are collected and a market overview of already existing systems is created. In consideration of all requirements, the goal of this prototypical development is to supply the requirements of the old and new system.

## **Danksagung**

An dieser Stelle, möchte ich die Gelegenheit nutzen und mich bei allen bedanken, die mich während der Entstehung dieser Arbeit unterstützt haben. Besonders bedanken möchte ich mich bei meiner Betreuerin Frau Prof. Dr. Bettina Buth für ihr Engagement und Unterstützung dieser Arbeit.

Mein Dank geht ebenfalls an Herrn Prof. Wolfgang Gerken, der sich bereit erklärt hat, die Zweitkorrektur zu übernehmen. Des Weiteren geht ein Dank an Herrn Torben Fuß für die Ermöglichung dieser Arbeit und für die fachliche Betreuung des Projektes. Ebenfalls möchte ich Frau Ellen Hinrichsen, Herrn Sven Arendt und Herrn Alkiviadis mein Dank aussprechen.

Zuletzt möchte ich mich meiner Familie für die hervorragende Unterstützung während des gesamten Studiums danken.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>7</b>
<b>Abbildungsverzeichnis</b>	<b>8</b>
<b>1 Einleitung</b>	<b>10</b>
1.1 Motivation . . . . .	10
1.2 Aufgabe und Ziel . . . . .	11
1.3 Gliederung der Arbeit . . . . .	11
<b>2 Grundlagen</b>	<b>12</b>
2.1 Content Management System . . . . .	12
2.2 Java EE . . . . .	13
2.2.1 Java Server Faces (JSF) . . . . .	14
2.2.2 EJB . . . . .	15
2.3 JBoss . . . . .	17
2.4 Jasper Report . . . . .	17
<b>3 Analyse</b>	<b>20</b>
3.1 Systemidee . . . . .	20
3.2 Bewertung des Altsystems . . . . .	25
3.3 Neuplanung . . . . .	27
3.4 Anforderungsspezifikation . . . . .	28
3.4.1 Funktionale Anforderung . . . . .	28
3.4.2 Nichtfunktionale Anforderungen . . . . .	30
3.5 Akteure . . . . .	31
3.6 Nutzungsszenarien . . . . .	36
3.6.1 Kapitel . . . . .	36
3.6.2 Basket . . . . .	44
3.6.3 Admin . . . . .	44
3.7 Untersuchung existierende CMS-System . . . . .	47
3.7.1 Typo3 . . . . .	47
3.7.2 Joomla! . . . . .	48
3.7.3 Drupal . . . . .	50

3.7.4	Fazit . . . . .	52
3.8	Auswahl eines Report Tools . . . . .	53
<b>4</b>	<b>Entwurf</b>	<b>54</b>
4.1	Fachliche Architektur . . . . .	54
4.1.1	Datenmodell . . . . .	55
4.1.2	Rollen-/Rechtesystem . . . . .	58
4.2	Technische Architektur . . . . .	61
4.2.1	Drei Sichten Architektur . . . . .	61
4.2.2	Klassenmodell . . . . .	62
<b>5</b>	<b>Realisierung</b>	<b>64</b>
5.1	Startseite . . . . .	64
5.2	Aufteilung in Komponenten . . . . .	66
5.2.1	Menu Komponente . . . . .	67
5.2.2	User Komponente . . . . .	69
5.2.3	Work Order Komponenten . . . . .	70
5.2.4	Figur Komponenten . . . . .	75
5.2.5	Report Komponenten . . . . .	78
5.2.6	Admin Komponenten . . . . .	79
<b>6</b>	<b>Testkonzept</b>	<b>83</b>
6.1	Komponententest . . . . .	83
6.2	Integrationstest . . . . .	86
6.3	Systemtest . . . . .	86
6.4	Abnahmetest . . . . .	87
<b>7</b>	<b>Zusammenfassung</b>	<b>88</b>
7.1	Fazit . . . . .	88
7.2	Ausblick . . . . .	89
	<b>Glossar</b>	<b>90</b>
	<b>Literaturverzeichnis</b>	<b>91</b>

# Tabellenverzeichnis

3.1	Nutzungsszenario Kapitel erstellen . . . . .	37
3.2	Nutzungsszenario Kapitel bearbeiten . . . . .	39
3.3	Nutzungsszenario Kapitel löschen . . . . .	40
3.4	Nutzungsszenario Figur bearbeiten . . . . .	42
3.5	Nutzungsszenario Basket ansehen . . . . .	44
3.6	Nutzungsszenario Benutzer erstellen . . . . .	45
3.7	Nutzungsszenario Benutzer bearbeiten . . . . .	45
3.8	Nutzungsszenario Benutzer löschen . . . . .	46
3.9	Nutzungsszenario Benutzer-Rollen verwalten . . . . .	46

# Abbildungsverzeichnis

2.1	iReport	18
2.2	JasperReport Ablaufdiagramm	19
3.1	Airbus A380	23
3.2	Daten aus dem SAP-System	24
3.3	Altsystem	26
3.4	Akteure Arbeitsvorbereiter	32
3.5	Akteure, Redakteure	33
3.6	Akteure Projektleiter	34
3.7	Akteure Admin	35
3.8	Akteure SuperUser	35
3.9	Nutzungsszenario Kapitel Erstellen	36
3.10	Nutzungsszenario Kapitel Verwalten	38
3.11	Nutzungsszenario Figur bearbeiten	41
3.12	Nutzungsszenario Admin	44
3.13	Typo3	48
3.14	Joomla!	49
3.15	Drupal	51
4.1	Komponenten	54
4.2	Firmen Struktur	56
4.3	Auftrag Struktur	57
4.4	Figur Struktur	58
4.5	Schichten Architektur	61
4.6	Paketdiagramm	63
5.1	Login Panel	65
5.2	Startseite Arbeitsvorbereiter	65
5.3	Startseite Redakteure, Projektleiter und Admin	66
5.4	Menu Admin	67
5.5	Menü Projektleiter	68
5.6	Menü Redakteure	68
5.7	Menü Arbeitsvorbereiter	69

## Abbildungsverzeichnis

---

5.8	User Settings . . . . .	70
5.9	Basket . . . . .	72
5.10	Edit Work Order . . . . .	73
5.11	Create Workorder Step 1 . . . . .	74
5.12	Create Workorder Step 2 . . . . .	75
5.13	Figure Step 1 . . . . .	76
5.14	Figure Step 1 . . . . .	77
5.15	Reports . . . . .	78
5.16	Admin View User . . . . .	80
5.17	Admin Create User . . . . .	81
5.18	Admin User Role . . . . .	82

# 1 Einleitung

## 1.1 Motivation

Im Zuge verändernder informationeller und zeitlicher Bedingungen entwickelte sich das World Wide Web (WWW) von seiner anfänglichen Form, der einfachen Text-Anwendung, zu einer komplexen Client-Server-Applikation Plattform. Da das WWW eine ideale Infrastruktur für verteilte Anwendungen bietet, ergänzt es nicht nur herkömmliche Desktop-Applikationen, sondern löst diese teilweise ab. Um bei dieser Inflation mitzuhalten, sind entsprechend Strategien und Werkzeuge notwendig.

Content-Management-Systeme (CMS), ist in diesem Zusammenhang ein fester Bestandteil geworden. Ein CMS erfüllt zunächst die Aufgabe, Inhalte zu verwalten und bereitzustellen. Inhalte können einfache Webseiten, aber auch umfangreiche und komplex strukturierte Informationen sein, die an unterschiedlichen Stellen einer Organisation auftreten. Diese Inhalte müssen dabei teilweise in unterschiedlichen Ausgabeformaten verfügbar gemacht werden. Demzufolge ist CMS heutzutage ein wichtiges Medium, das häufig zur Kommunikation und Informationsverarbeitung im Unternehmen genutzt wird.

Dementsprechend und parallel zu beobachten ist, dass die Anforderungen in dieser steigenden Entwicklung, eine Optimierung der Geschäftsprozesse erfordern. Die Aktuellen Marktbedingungen drängen dazu, das Geschäftsprozesse und deren technische Implementierung schlank und flexibel gestaltet werden, um die dazu gehörenden Kosten zu reduzieren und Reaktionszeiten auf veränderte Marktbedingungen zu verkürzen. Nur so, kann ein langfristiger, unternehmerischer Erfolg gesichert werden.

Um eben solche Geschäftsprozesse informationstechnisch umzusetzen bietet Enterprise Content Management (ECM), die entsprechenden Prinzipien, die Beispielsweise Methoden, Technologien und Software-Komponenten wären.

Die Motivation dieser vorliegenden Arbeit ist, Erfahrungen, die bei der Ablösung des bisherigen Altsystems, durch die neue Entwicklung oder die Weiterentwicklung eines bereit existierenden Systems, zu sammeln.

## 1.2 Aufgabe und Ziel

Ziel dieser Arbeit ist, den Zyklus der Ablösung des bisherigen Altsystems zu konzipieren. Das Altsystem, das auf einer netzwerkbasierter Plattform besteht, soll durch eine neue webbasierte Plattform abgelöst werden.

Diese Arbeit beschäftigt sich mit der Ausarbeitung eines Konzepts für ein neues System. Dabei soll durch geeignete Analysen, ein Konzept des Systems erarbeitet und in einer Marktübersicht mögliche einsetzbare Systeme gefunden werden. Zum Abschluss dieser Arbeit soll ein Prototyp erstellt werden, der die Anforderungen des Altsystems und den Funktionsumfang des neuen, zu entwickelnden Systems, beinhaltet.

## 1.3 Gliederung der Arbeit

- *Kapitel „Grundlagen“* beschäftigt sich mit der Erläuterung des Content-Management-System Konzepts. Es werden die wichtigsten technologischen Standards erläutert, die in dem neuem System zu den Einsätzen kommen.
- *Kapitel „Analyse“* entwickelt durch Analyse des Altsystems die Anforderungen die an das neue System gestellt werden. Dazu werden die Anforderungen des neuen Systems ermittelt und anhand existierender Systeme verglichen. Außerdem wird das neue System anhand von Nutzungsszenarien erläutert.
- *Kapitel „Entwurf“* befasst sich mit der grundlegenden Entwurfsentscheidung die sich auf der Basis der Analyse des Fachkonzepts modelliert. Außerdem werden die Komponenten aus der fachlichen/technischen Sicht beschrieben und das Konzept für die Oberflächenstruktur dargestellt.
- In *Kapitel „Realisierung“* wird der entwickelte Prototyp vorgestellt. Zusätzlich werden die einzelnen Komponenten des Systems beschrieben.
- In *Kapitel „Testkonzept“* wird ein Testplan aufgestellt und durchgeführt, um die Qualität des Gesamtsystems zu messen.
- In *Kapitel „Zusammenfassung“* wird abschließend die Arbeit bewertet und ein Ausblick auf mögliche Weiterentwicklung gegeben.

## 2 Grundlagen

In diesem Kapitel werden die grundlegenden Technologien, die diese Arbeit beinhaltet, erläutert. Vorerst wird auf das Content Management System eingegangen, um den theoretischen Aspekt des Systems abzudecken. Zum Abschluss wird das Java EE Konzepte erklärt.

### 2.1 Content Management System

Ein Content Management System (CMS) ist eine Technologie zur Verwaltung des Inhaltes einer Internetseite oder anderer Informationsangebote, deren wichtigsten Aufgaben in folgenden Kategorien gegliedert werden können:

- Beschaffung und Erstellung von Inhalten
- Präsentation und Publikation von Inhalten
- Aufbereitung und Aktualisierung von Inhalten
- Management und Organisation von Inhalten
- Verteilung und Integration von Inhalten

Durch eine klare Trennung von Inhalt und Layout können Redakteure, ohne dass sie Programmierkenntnisse oder Erfahrungen besitzen, selbständig mithilfe von Editoren, Veränderungen am Inhalt vornehmen.

Der zusätzliche Vorteil eines CMS ist, dass die Betreuung der Webinhalte nicht mehr nur durch einen Webmaster im Unternehmen oder extern von einer Agentur vorgenommen werden können, sondern durch alle Benutzer, die eine Zugangsberechtigung haben.

### **Enterprise Content Management (ECM)**

Im Vergleich zu einem Content Management System ermöglicht ein Enterprise Content Management nicht nur eine professionelle, effiziente und aktuelle Gestaltung von umfangreichen Webauftritten.

Die AIIM, die internationale Enterprise Content Management Assoziation, hat 2003 eine eingängige Definition geliefert, was unter ECM zu verstehen sei (AIIM International, 2003):

*„Enterprise Content Management (ECM) is the technologies used to capture, manage, store, preserve, and deliver content and documents related to organizational processes. ECM tools and strategies allow the management of an organization's unstructured information, wherever that information exists.“*

Somit kann ein Enterprise Content Management als ein Konzept bzw. ein strategisch-organisatorisches Modell definiert werden, welches den Umfang einer einzigen Technologie oder gar einer Software übertrifft. Darüberhinaus organisiert, vereinheitlicht und vereinfacht ein ECM alle Unternehmensinformationen speziell auf die individuellen Bedürfnisse des Unternehmens, zur Verwaltung und Bearbeitung aller unstrukturierten Informationen.

## **2.2 Java EE**

Java EE steht für Java Enterprise Edition und ist ein Standard für verteilten und mehrschichtigen Anwendungen der Programmierung. Verteilt ist eine Anwendung immer dann, wenn nicht alle Komponenten auf dem gleichen Computer, oder um noch genauer zu sein, nicht in der gleichen Java VM (Virtual Maschine) laufen. Die Enterprise Edition baut auf der Standard Edition (Java SE) auf und erweitert diese um eine Reihe von Technologien, die im Folgenden näher betrachtet werden.

Ziel von Java EE ist die Aufteilung der Anwendungslogik in einen Client- und einen Serverteil. Alle fachlichen Funktionen sollen zentral auf einem Server zur Verfügung gestellt werden, auf den alle Clients zugreifen. Dort finden auch alle Datenbankzugriffe statt. Damit die Serverkomponenten erreichbar und verwaltbar sind, werden sie unter die Kontrolle eines Applikationsservers gestellt, der die technischen Funktionalitäten für Sicherheit, Transaktion, Namen- und Verzeichnisdienste zu Verfügung stellt (Assisi, 2005).

Ferner werden die wichtigsten Komponenten der Java EE-Architektur, die zum Einsatz kommt, dargestellt:

- Enterprise Java Beans (EJB) definieren die Geschäftslogik einer Anwendung und ermöglichen Zugriff auf persistente Daten
- Java Servlet API ermöglicht dynamische Webanwendungen
- Java Message Service (JMS), eine API für asynchronen Nachrichtenaustausch
- Java Authentication and Autorisation Service (JAAS) ermöglicht eine Authentifikation und das Bereitstellen von Zugriffsrechten in Java Programmen
- Java Server Faces (JSF) ermöglicht dem Benutzer Komponenten für Benutzerschnittstellen in Webseiten auf einfache Art und Weise zu erstellen

Im Folgenden soll nun, die für die Entwicklung der Prototypen, verwendet er Technologie EJB, näher erläutert werden.

### 2.2.1 Java Server Faces (JSF)

Für die Ausgabe von dynamischen HTML-Seiten eines Webservers verwendet man im Javaumfeld üblicherweise Java Server Pages (JSP). Damit wird die Einbettung von Javacode- und somit von dynamischen Inhalten - in HTML-Seiten ermöglicht. Die Vermischung von Java- und HTML-Code führt allerdings schnell zu einer unübersichtlichen Programmstruktur, die die Entwicklung deutlich verkompliziert.

Um die Entwicklung von Webseiten zu vereinfachen, wurden Web-Frameworks wie zum Beispiel Java Server Faces (JSF) entwickelt. JSF benutzt das Model-View-Controller-Architekturmuster (MVC). In der Anwendung werden die Daten (Model), die Präsentation (View) und die Programmsteuerung (Controller) voneinander getrennt. JSF ist hauptsächlich für die Präsentation zuständig und verfolgt zusätzlich einen komponentenbasierten Ansatz, mit dem man in einfacher Art und Weise Komponenten für Benutzerschnittstellen in Webseiten einbinden und die Navigation definieren kann.

Die wichtigsten Merkmale von JSF sind:

- Verwaltung der Zustände über mehrere Seiten über einen Komponentenbaum
- Browser-spezifische Generierung von HTML
- Server-seitige Formularüberprüfung durch Standard-Validatoren
- Fehlerbehandlung durch Messages
- Internationalisierung durch Ressource Bundles

### 2.2.2 EJB

EJBs (Enterprise JavaBeans) stellen einen wichtigen Bestandteil von Java EE dar. EJBs sind Komponenten, die Server-seitig in einem speziellen EJB-Container ablaufen und die vorrangig zur Implementierung von Geschäftslogik und Persistenz verwendet werden. Der EJB-Container arbeitet als Vermittlungsschicht zwischen Client und Datenbank und kümmert sich sowohl um die Speicherung der Daten, als auch um deren Verfügbarkeit für jeden autorisierten Client.

Die Vorteile von EJBs sind:

- Standardisierung, Wartungsfreundlichkeit
- Verteilte Anwendungen möglich
- Verteilte Transaktionen
- Definierte Sicherheitsmechanismen
- Hohe Ausfallsicherheit erreichbar
- Hohe Skalierbarkeit

Bei den EJBs unterscheidet man zwischen drei verschiedene Arten von Beans:

#### **Session Beans**

Session Beans sind EJBs, die einem Client während einer Sitzung zu Verfügung stehen. Nach der Beendigung der Session ist es nicht mehr möglich auf die Session Beans zu zugreifen. Die Hauptfunktion der Session Beans liegt darin, Code auf einem oder mehreren Server auszuführen. Der Vorteil solcher Session Beans ist, dass dadurch eine erhöhte Sicherheit gewährleistet wird. Demnach muss eine Session Bean ein Interface implementieren, damit bestimmt wird welche Methoden veröffentlicht werden. Da der Aufrufende nur auf dieses Interface verweist, erkennt er die Klasse nicht, in welches das Interface implementiert wurde

Die Session Beans werden zusätzlich in zwei Beans unterteilt:

- **Stateful Session Beans**

Die Statful Beans sind zustandsbehafteten Beans d.h., dass sie ihren Zustand zwischen den einzelnen Methodenaufrufen beibehalten. Dies entspricht einem lokalen

Objekt, dessen Felder ebenfalls ihren Zustand so lange beibehalten, wie das Objekt existiert (Assisi, 2005).

- **Stateless Session Beans**

Die Stateless Beans sind zustandslose Beans die im Gegensatz zu dem Stateful Beans eine Funktionssammlung darstellen, daraus ergibt sich, dass sie ihren Zustand und die Inhalte, deren Variablen nicht dauerhaft sind und nicht speichern. Der Vorteil der zustandslosen Beans ist, dass sie vom Container besonders ressourcenschonend verwendet werden können. Dadurch, dass sie keinen Zustand besitzen, stehen sie allen Client zur Verfügung und können parallel aufgerufen werden.

### **Entity Beans**

Die Hauptaufgabe der Entity Beans ist die Persistenz, also das dauerhafte Speichern über mehrere Sitzungen hinweg. Entity Beans repräsentieren oft ein Objekt aus der realen Welt, die als POJOs (Plain old Java Objects) im System realisiert werden. Die Daten einer Entity Bean entsprechen einer Zeile einer Datenbanktabelle oder mehrerer Tabellen. Anders als eine Session Bean kann eine Entity Bean mehrere Clients gleichzeitig bedienen.

### **Message-driven Beans**

Message-driven Beans werden für die Implementierung asynchroner Kommunikation in Java EE-basierten Systemen eingesetzt. Sie sind in der Lage asynchrone Nachrichten entgegenzunehmen und sie zu verarbeiten. Dabei gibt es zwei verschiedene Möglichkeiten der Kommunikation, die das Java Message Service (JMS) bereitstellt.

- **Punkt-zu-Punkt**  
Besteht darin, dass durch die Message Queue genannte Warteschlange Punkt-zu-Punkt Verbindung zwischen Sender und Empfänger aufgebaut werden, bei denen der Empfänger aus der Queue Nachrichten ausliest.
- **Message Topic**  
Die Message Topic Möglichkeit funktioniert nach dem Publisher/Subscriber-Prinzip. Ein Sender verschickt eine Nachricht an ein Topic und alle die dieses Topic abonniert haben. Demnach kann eine Nachricht mehreren Empfängern zugestellt werden.

Ein Messaging-Service garantiert dabei, dass eine Nachricht vom Client exakt einmal bei einer Nachrichten-getriebenen Komponente landet. Der Unterschied zu den beiden anderen Beans ist, dass ein Message-driven Bean nicht direkt aufgerufen werden kann und sie ist für einen Aufrufenden nicht sichtbar ist.

## 2.3 JBoss

Der JBoss Applikation Server ist die Implementierung eines Applikation-Servers nach dem J2EE-Standard und Teil des JBoss Middleware-Frameworks. JBoss ist freie Software unter der LGPL-Lizenz und kann inklusive Quelltext kostenlos von der Herstellerwebseite heruntergeladen werden. JBoss ist neben BEA Weblogic und IBM Websphere einer der drei verbreitetsten J2EE-Applicationserver. Viele benutzen den JBoss Applikation Server auch zum Test der J2EE-Applikationen.

Als Gründe für die Verwendung des JBoss Applikations-Servers werden neben Kostenfaktoren, vor allem die Stabilität gegenüber anderen Plattformen, höhere Entwicklerproduktivität, Standardkonformität und die gute Integration mit weiteren Open-Source-Projekten (wie Hibernate, jBPM, JBoss Portal und anderen) genannt.

Der JBoss Applikation Server bietet einzelne „Dienste“ wie JAAS, einem Authentifizierungsdienst und einem Hibernate-Modul, mit dem es möglich ist den Zustand eines Objekts in einer relationalen Datenbank zu speichern und aus entsprechenden Datensätzen wiederum Objekte zu erzeugen, ohne die DB-Zugriffe explizit in SQL programmieren zu müssen (Red Hat Inc.) .

## 2.4 Jasper Report

JasperReports ist ein bekannter und großer Report Tool, was seit September 2001 vom Teodor Danciu entwickelt worden ist und seit dem ständig weiter entwickelt wird. Außerdem enthält es eine Javabibliothek, welche sich problemlos in Java- Anwendungen integrieren lässt. Diese Bibliothek stammt von der Firma JasperSoft, welche 2005 gegründet wurde.

JasperReports ist ein in Java geschriebenes Open-Source Reporting Tool. Es ist in der Lage aus den gängigsten Datenquellen auf Basis von Formatvorlagen druckfertige Inhalte auf Drucker, Bildschirm oder in Dateien des Formats PDF, HTML, XML, XLS, CSV, RTF, TXT und ODF auszugeben (JasperSoft, 2006). Es eignet sich für den Einsatz in Desktop Java Applikationen, Servlet-, EJB-Container oder Hibernate-Implementierung. JasperReports benutzt für das Layout eine eigene XML-Beschreibungssyntax. Die Daten werden aus verschiedenen Datenquellen, wie Relationales Datenbanken oder Arrays und Collections mit beliebigen Java Objekten, bezogen.

Eine der Stärken von JasperReports ist die rekursive Unterstützung von Subreport. Diese ermöglicht es, lange Berichte seitenweise aufzubereiten. So können überlange Reportergebnisse optimal dargestellt werden (ADELCHI 2004).

## 2 Grundlagen

JasperSoft bietet eine grafische Oberfläche namens iReport (siehe Abbildung 2.1), um zunächst Reports zu gestalten, oder Report-Templates zu erstellen. Diese Templates können durch die SQL Anweisung oder durch EJBs, welche die gewünschten Daten zurückliefern gefüllt werden. Anschließend kann zur Laufzeit durch den Jasper Compiler von JasperReports ein Exemplar dieses Templates erstellt werden, dass alle gewünschten Daten enthalten und durch den JasperExport- oder JasperPrintManager auf den Monitor, Druckansicht bringen oder exportieren lassen (siehe Abbildung 2.2).

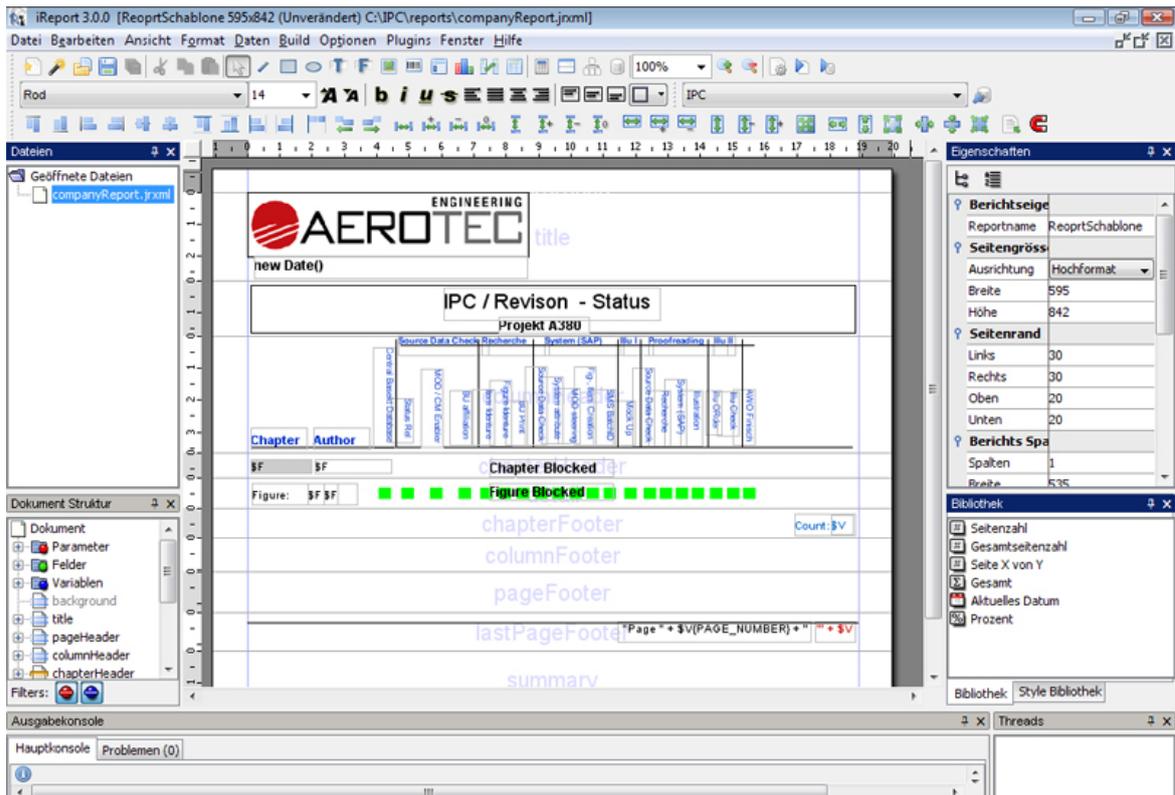


Abbildung 2.1: iReport

In Abbildung 2.1 wird durch das Tool iReport ein Report-Template dargestellt.

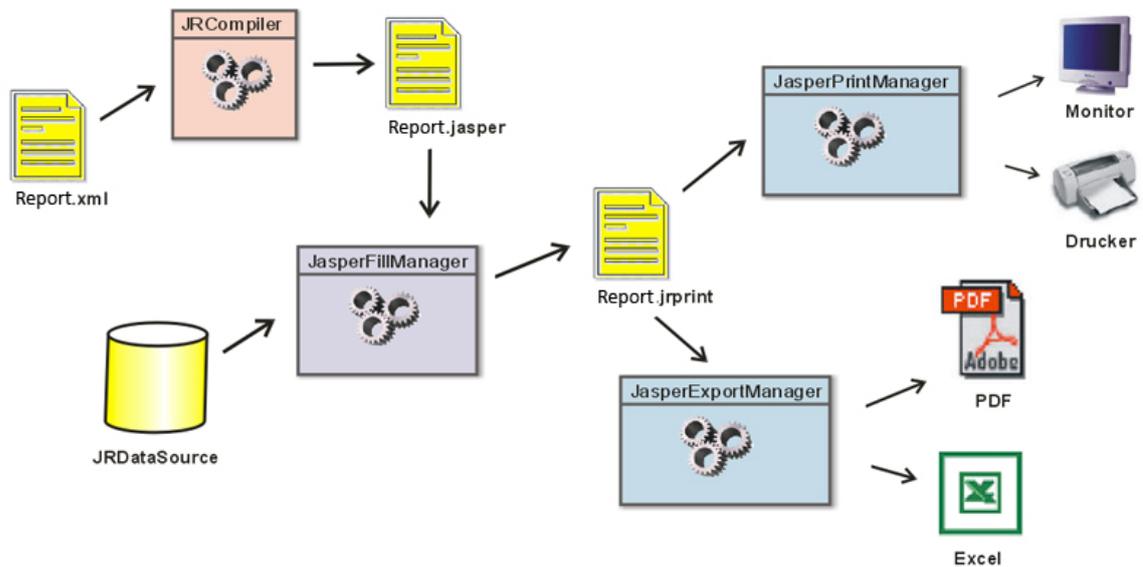


Abbildung 2.2: JasperReport Ablaufdiagramm

Anhand der Abbildung 2.2 JasperReport werden deutlich, dass der Ablauf eines Jasper-Reports in drei Abschnitten gegliedert wird. Der erste Abschnitt befasst sich mit der Erstellung bzw. Definierung des Report-Template (XML-Datei). Im zweiten Abschnitt wird das XML-Dokument vom JasperCompileManager verarbeitet. In Java Quellcode wird die Datei kompiliert und gespeichert (Report.jasper). Die erstellte Jasper-Datei wird nun mit dem JasperFillManager mit den vorgegebenen Daten des Templates gefüllt. Die dadurch resultierende Jrprint-Datei kann nun im letzten Schritt mit dem JasperPrintManager oder dem JasperExportManager in gängige Dateiformate oder in Druckversionen überführt werden.

# 3 Analyse

Ziel dieses Kapitels ist es, durch angemessene Analysen, die Anforderungen an das neue System zu identifizieren. Dazu wird in erster Linie der Funktionsumfang des Altsystems bestimmt und bewertet. Im Anschluss daran, werden diese Anforderungen in einer Marktübersicht mit bereits verfügbaren CMS-Systemen verglichen, um eine Entscheidung zu treffen, ob ein existierendes System eingesetzt werden kann oder ob eine Neuentwicklung geplant werden sollte.

## 3.1 Systemidee

Das augenblickliche System wird von den Anwendern verwendet, um für das Flugzeug Airbus A380, das Handbuch IPC (Illustration Part Catalog), Kapiteln (Chapters) und deren auftragsbezogene Daten und Information zur Selbstverwaltung anzulegen. Die Plattform soll den Anwender zusätzlich ermöglichen ihre auftragsbezogene Daten zu kontrollieren, analysieren und zu steuern.

Um den Kapitel- und Figurenaufbau übersichtlich gestalten zu können, orientieren sich die Redakteure an die vordefinierte Frame und Zonenaufteilung (siehe Abbildung 3.1).

Die wichtigsten Funktionen des Anwenders sind momentan:

### **Erstellen von Work Orders**

Ein Work Order ist eine Interne Darstellung eines Kapitels und deren auftragsbezogene Daten, die nach der Erstellung des Kapitels eine eindeutige Work Order ID erhält. Demnach repräsentiert ein Work Order einen Kompletten Kapitel und den dazugehörigen Aufträge.

Um ein Kapitel (Chapter) und die dazugehörigen Aufträge (AWOs) für das Handbuch IPC anlegen zu können, muss der Anwender die erforderlichen Daten aus dem SAP-System manuell beschaffen. Für das Anlegen des Kapitels werden die Daten Chapter-Nummer und die Required - Start/End Datum aus dem SAP-System benötigt. Des Weiteren werden die

Daten Auftragsnummer (AWO), Bauunterlagennummer (LBU), Modifikationsnummer (MOD) und die Beschreibung (Description) benötigt um einen Auftrag zu dem Kapitel anzulegen (siehe Abbildung 3.2).

Durch die Beschaffung der erforderlichen Daten und dem Zuweisen eines zuständigen Redakteurs, ist es dem Anwender nun möglich, ein Kapitel im Altsystem anzulegen. Nach dem erfolgreichen Anlegen des Kapitels wird dem erstellten Kapitel, eine Interne Work Order Nummer vergeben.

Außerdem hat der Anwender die Möglichkeit einen Timeschedule (Zeitplan) durch Festlegen des Anfangs Datum zu erstellen. Nach dem Anlegen des Timeschedule werden die Should (soll) Daten der einzelnen Milestones berechnet.

#### **Verwalten von Workorders**

Damit ein Anwender einen Work Order bearbeiten/verwalten kann, muss dem Anwender einen Work Order vorher zugewiesen worden sein. Ihm ist es nun möglich die angelegten

- Awos (Aufträge)
- Timeschedule (Zeitplan)
- Chapter Setup (Kapitel Aufbau)

*In diesem Abschnitt ist es dem Redakteur möglich für das Flugzeug Airbus A380 das ausgewählte Kapitel für die vordefinierten Frame und Zonebereiche einzuteilen(sieh Abbildung 3.1).*

- Figuren

*In diesem Abschnitt ist es dem Redakteur möglich für das jeweilige Kapitel Figuren anzulegen. Die Figuren stellen wiederum Bereich in den vorher definierten Framebereiche (Chapter Setup) dar.*

- Blocking Point

*Das Thema Blocking Point ermöglicht dem Redakteur Fehlzustände zu definieren und resultierende Aktionen daraus abzuleiten. Die Archivierung der Blocking Points eines Auftrages wird z.B. als Basis zur Begründung von Terminüberschreitungen verwendet.*

zu bearbeiten oder welche hinzuzufügen.

#### **Verwalten von Figuren**

Ein Redakteur hat momentan die Möglichkeit Figuren zu verwalten. Durch Auswählen einer Figur ist es dem Anwender anschließend Möglich

- Status

*In dem Bereich Status setzt der Redakteur den Fortschritt der Einarbeitung der Figur fest. Der Status kann zur jeder Zeit variiert werden. Des Weiteren ist es dem Redakteur durch die Belegung des Status möglich eine Übersicht der Figur zu beziehen.*

- Setups

*In dem Abschnitt Figur Setup ist es dem Redakteur möglich durch die vorher festgelegten Frame und Zonebereiche im Chapter Setup, die Figuren an den jeweiligen Bereichen zu koordinieren (siehe Abbildung 3.1).*

- Illustration

*Die Illustration beschäftigt sich mit der Vereinfachung der technischen Bauunterlagen und Darstellung, die anhand von diversen Darstellungsmöglichkeiten dargestellt werden können.*

- Additional Source Data

*Die Additional Source Data ermöglicht dem Redakteur zusätzliche verwendete Bauunterlagen zu archivieren, die nicht über die AWOs beauftragt wurden sind.*

- Items

*In dem Bereich Items wird dem Redakteur ermöglicht die jeweilige Figur in ihre einzelnen Komponenten aufzuteilen und festzuhalten.*

- Blocking Points

der ausgewählten Figur zu verwalten.

#### **Einsicht auf entsprechenden Basket**

Einzelnen Anwender ist es möglich durch den Basket eine Auflistung seiner persönlichen Daten zu bekommen. Der Basket besteht aus einer Zusammenfassung der persönlichen Work Order, Figuren, Illustrationen, AWOs und Proofreadings (siehe Abbildung 3.3).

#### **Einsehen der jeweiligen Fertigstellungsgrade der Aufträge (Reports)**

Jeder Anwender hat die Möglichkeit sich die vordefinierten Reports zu betrachten (siehe Abbildung 3.3).

### 3 Analyse

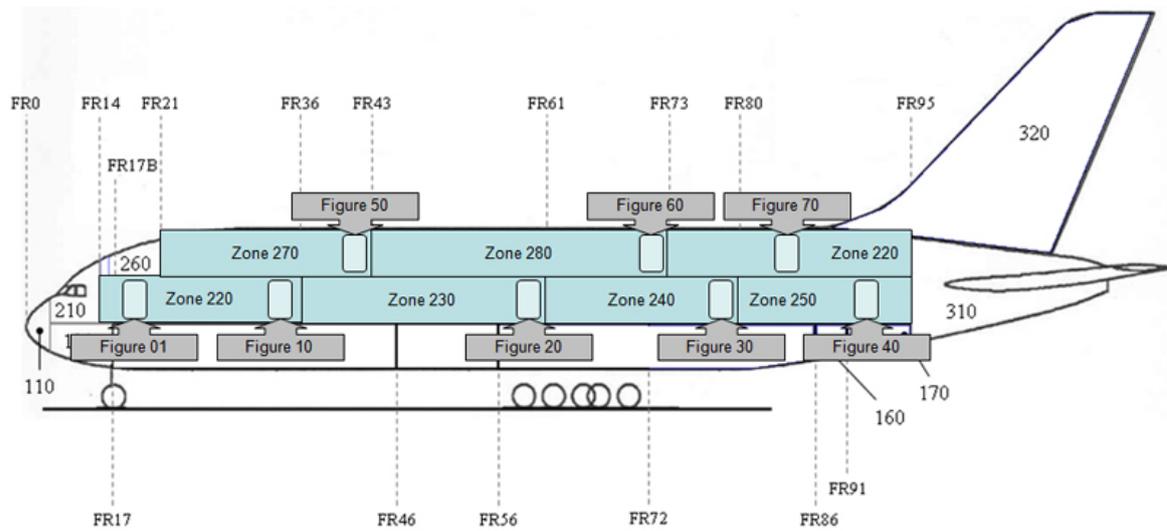


Abbildung 3.1: Airbus A380

Abbildung 3.1 verdeutlicht eine Möglichkeit einer Aufteilung des Flugzeug Airbus A380 in vordefinierten Frame und Zonebereiche

### 3 Analyse

The screenshot displays the SAP Work Order Data interface. At the top, there is a navigation bar with icons for Home, Partner, and other functions. The main header area includes the following information:

- Notification: **AWO** 10060266 Z1 257871\_inc. T77090 L257-71903-004-00
- Status: NOPR OSTS **Chapter** ACCE
- Order: [Empty field]

Below the header, there are several tabs: Header, Work Order Data, Split Work Order, Actions Follow-Up, and Partners. The 'Work Order Data' tab is active, showing the following sections:

- Start/End Dates:** Required Start: 28.08.2008, Required End: 30.09.2008, Priority: Low, Revision: [Empty field]
- Incorporation data:** Criterion/Event: L-MPM67657T77090-PC, ACD: 02.09.2008, ACD Requested: [Empty field] Change
- Other data:** Workload: [Empty field], Master W/O: 000010050691, W/O Structure: [Empty field], Rejection cause: [Empty field]
- Subject:** C.Baskets: TLD- AP11 GE / IPC-WP11 Authoring, Description: 257871\_inc. T77090 L257-71903-004-00

The description field contains the following text:

This workorder concerns to the incorporation of FINs 5067MM190MX 5068MM190MX ,5069MM190MX,5099MM190MX by using design solution L257-71903-004-00 for Batch ID

257871 (10)

Please incorporate the new partnumbers outlined in attached TRS T77090

Abbildung 3.2: Daten aus dem SAP-System

Anhand Abbildung 3.2 wird verdeutlicht, welche Daten aus dem SAP-System beschaffen werden müssen um einen Kapitel (Work Order) und den dazugehörigen Auftrag (AWOs) für das Handbuch IPC anlegen zu können.

## 3.2 Bewertung des Altsystems

Bei der Vorstellung und Bearbeitung des momentanen Systems, sind auf Anhieb Defizite in der Bedienbarkeit der Masken, vom zuständigen Mitarbeiter beobachtet wurden, die einen klaren Aufbau und eine einfache Bedienung nicht ermöglichen.

Da einem Benutzer, der sich nicht mit dem System ausführlich beschäftigt und keine Schulung über diese Materie besucht hat, kann nicht problemlos einen Work Order erstellen. Der Benutzer müsste sich über mehrere Masken durch das System klicken. Hat man die Maske des Create Work Orders entdeckt, wird einem nicht gleich bewusst, ob die Create Work Order Maske zum anlegen oder zum bearbeiten von Work Orders dient, da die Maske schon mit Daten eines Work Order gefüllt sind. Des Weiteren könnten bei dem Benutzer Verwirrungen im Bezug auf die Funktionen des Baskets auftreten. Schwierigkeiten könnte es beim editieren dieser Datei geben, da nicht ganz klar wird, ob sie allein zur Auflistung der auftragbezogenen Daten des einzelnen Benutzers dient, oder es auch möglich sein könnte, dies durch anklicken der einzelnen aufgelisteten Daten zu erreichen.

Ein wesentlicher Aspekt über den nicht hinweg gedacht werden kann, dass man bei der Verwaltungsmaske von Work Order und Figur vergebens nach der Möglichkeit sucht, einen Aufbau der jeweiligen Figur bzw. Kapitels zu definieren, da sie sich nicht in der jeweiligen Maske befinden, sondern es existieren für den jeweiligen Figurenaufbau/ Kapitelaufbau einzelne Masken, die wiederum zusätzlichen Defizite, wie die Create Work Order Maske aufweisen. Somit ist ein klarer Aufbau, der einzelnen Prozesse anhand von umstrukturierten Masken und unübersichtliche Informationen, nicht gegeben.

Des Weiteren wurden bei der Betrachtung des Datenmodells, Verletzungen mehrere Normalformen (NF) entdeckt, die sich durch zusammengesetzte, mengenwertige oder geschachtelte Wertebereiche aufmerksam machten, was zur Folge hat, dass sie abhängig voneinander sind. Durch die Verletzung der Normalformen, werden Redundanzen der Daten nicht vermieden, was bedeutet, dass ein erhöhter Speicherplatz benötigt wird, was wiederum die Konsequenz wäre, dass das Durchsuchen und Analysieren der Daten verlängert würde und es bei Änderungen von Daten zu Inkonsistenzen führen könnte. Als Beispiel für die zusammengesetzte Wertebereiche sei das Attribut Adresse genannt, sie darf nicht als einzelner Attribut verwendet werden, sondern sollte in PLZ, Ort, Straße, Hausnummer aufgeteilt werden, um Redundanzen und Abhängigkeiten zu vermeiden.

Ein momentan großer Kritikpunkt ist, dass keine Benutzerverwaltung und Authentifizierung Möglichkeit verfügbar sei und somit gewährleistet wird, dass jeder Anwender Einsicht auf die Daten des anderen Anwenders bekommt und jegliche Veränderung der Daten vornehmen könnte, was Manipulation zur Folge hat. Ferner ist es dem Anwender nicht möglich Aufgaben zu delegieren, Zuständigkeiten zu definieren, die Identifizierung und Verifizierung zu gewährleisten.

Als Wunsch für das neue System wurde eine webspezifische Anwendung mit Benutzerverwaltung genannt, die die gängigen Sicherheitsmerkmale mit sich bringen und die Defizite des jetzigen Systems aufgehoben werden.

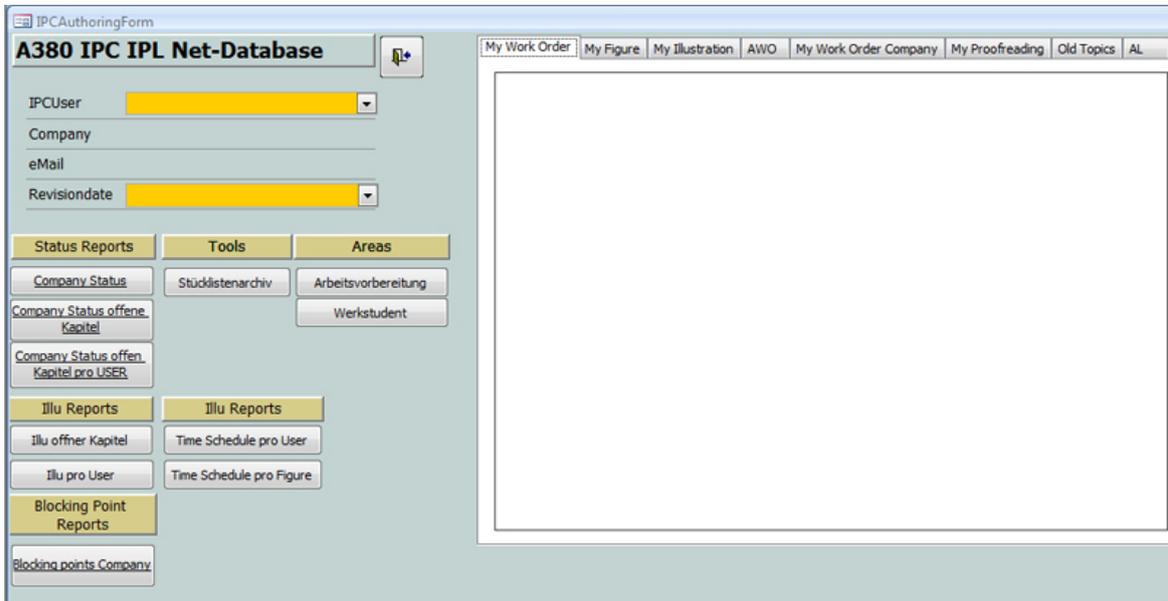


Abbildung 3.3: Altsystem

Abbildung 3.3 stellt die Startmaske des jetzigen Systems dar. Anhand dieser Abbildung wird noch mal verdeutlicht das ein Anwender keine Möglichkeit hat sich am System zu Authentifizieren und dadurch jedem Anwender möglich ist durch auswählen des IPCUsers Einsicht auf die Daten des anderen Anwenders zu bekommen.

### 3.3 Neuplanung

Der Analyse des Altsystems nach zu beurteilen, ist ein neues System zwingend notwendig, das vor allem die neuen Funktionalitäten:

#### Sicherheit

- die eine Authentifikation und Autorisation für den Anwender bietet um sich dem System zu authentifizieren.
- Wiederum soll bei der Autorisation die Rolle des Anwenders überprüft werden und ihm die Anwendung auf die Rolle angepasst werden

#### Benutzerverwaltung

- sodass einem Anwender Rollen zugeteilt / entzogen und deaktiviert / aktiviert werden können
- einzelne Bereiche bzw. Inhalte sollte mit Rechten versehen werden können
- zudem sollten die Rollen dynamisch mit Rechten angepasst werden können

ermöglicht.

Zusätzlich sollte eine deutliche Verbesserung des:

#### Datenmodell

- in dem eine optimierte Version des jetzigen Datenmodell erstellt wird, um Redundanzen der Daten zu vermeiden

und in Benutzerfreundlichkeit und Bedienbarkeit liefern.

Das neue System sollte alle jetzigen genutzten Funktionalitäten und die neuen Anforderungen mit sich bringen, diese aber so komfortabel zu lösen, dass jederzeit zusätzliche Anforderungen, die bisher noch nicht eingeplant waren, integrieren werden können

## 3.4 Anforderungsspezifikation

In diesem Abschnitt werden die Anforderungen an das neu zu entwickelnde System zusammengefasst, die sich aus der Fehleranalyse des Altsystems und aus dem Gespräch mit dem zuständigen Projektleiter des Altsystems ergaben. Diese Anforderungen werden in zwei Abschnitte unterteilt:

### **Funktionale Anforderung:**

diese legen fest, welche Aufgaben das neu zu entwickelnde System erbringen soll.

### **Nichtfunktionale Anforderung:**

legen fest, welche Eigenschaften das System bezüglich der Qualitätsanforderungen haben sollen.

### 3.4.1 Funktionale Anforderung

#### 1. Benutzer-Kennung

Ein bereits registrierter Benutzer kann sich über die Login-Seite des Systems schnell und bequem anmelden. Dazu ist seine Kennung erforderlich:

- sein Benutzername
- sein Passwort

Der angemeldete Benutzer kann sich jeder Zeit wieder vom System abmelden.

#### 2. Anlegen von Work Orders

Der Benutzer muss die Möglichkeit haben, für seine Firma abhängige Work Orders anzulegen. Außerdem muss der Benutzer auch die Möglichkeit besitzen AWO's und Timeschedule zu dem angelegten Work Order hinzufügen zu können. Dies ist nur der Fall, wenn dem Benutzer die Rechte zum Anlegen von Work Order besitzt. Diese Lizenzen werden im Vorfeld vergeben und geklärt.

### 3. Verwalten von Work Orders

Der Benutzer sollte die Möglichkeit haben, die einzelnen Komponenten AWO's, Figuren, Chapter Setup, Timeschedule und Blocking Points eines Work Orders abzuspeichern, hinzuzufügen, zu editieren und ggf. zu löschen.

### 4. Basket

Der Benutzer muss die Möglichkeit haben, in Bezug auf seinen persönlichen auftragsbezogene Daten einzusehen. Zudem wird ihm Zutritt zu den Daten der Benutzer der dazugehörigen Firma gewährt, soweit ihm die Rechte zugewiesen worden sind. Wichtige Voraussetzung hierfür ist die Privilegierung durch die dazugehörigen Rechte.

### 5. Verwalten von Figuren

Der Benutzer sollte die Möglichkeit haben, durch ein Auswählen von benutzerabhängige Figuren, deren Eigenschaften wie Details, Setup, Illustration, Additional Source Data, Blocking Points und Items abzuspeichern, zu editieren, hinzuzufügen und ggf. zu löschen. Auch hier ist es nur der Fall, wenn dem Benutzer die Rechte zugeteilt wurden.

### 6. Reports

Der Benutzer sollte die Möglichkeit habe, in vordefinierte Fertigstellungsgrad-Reports einzusehen und diese in ein geeignetes Format wie PDF-, Word- und Excel- abzuspeichern, um darauf zurückgreifen zu können und diese zu analysieren. Auch hier ist es nur der Fall, wenn dem Benutzer die Rechte zugeteilt wurden.

### 7. Benutzerverwaltung

Einem firmenspezifischen Administrator muss es möglich sein, in einer Auflistung firmenspezifische Usern hinzuzufügen, zu editieren und ggf. zu löschen. Außerdem muss der Administrator Usern Rollen zuweisen und die Möglichkeit haben sie auf aktiv / inaktiv zu setzten.

### 8. Rechte- / Rollenverwaltung

Der Super User muss die Möglichkeit haben, den vordefinierten Rollen manuell

Rechte hinzuzufügen und ggf. zu entziehen. Zusätzlich sollte die Möglichkeit bestehen einzelne Felder bzw. bestimmte Bereiche mit Rechte zu versehen, damit nur bestimmte Rollen Einsicht auf jeweilige Felder bzw. bestimmte Bereiche besitzen.

#### 9. Mandantenfähigkeit

Das System sollte mandantenfähig sein, d.h. es sollte möglich sein, auf einer Installation des Systems mehrere Unternehmen arbeiten zu lassen, die sich gegenseitig jedoch nicht beeinflussen und nur Zugriff auf ihre eigenen Daten haben dürfen. Der Datenschutz muss zu jeder Zeit gewährleistet sein.

### 3.4.2 Nichtfunktionale Anforderungen

#### 1. Bedienbarkeit

Das System sollte so einfach zu bedienen sein, dass sich jeder Benutzer auf Anhieb zu Recht findet. Es sollte einen klaren, simplen Aufbau zur Bedienung der einzelnen Module vorweisen können.

#### 2. Zuverlässigkeit

Das System soll selbst bei fehlerhaften Eingaben sachgemäß reagieren und insgesamt stabil laufen. Erzeugte Fehler sollten das System nicht zum Abstürzen führen, vielmehr sollte das System intelligent darauf reagieren und den Benutzer mit einer Fehlermeldung darauf aufmerksam machen.

#### 3. Vertraulichkeit

Die Vertraulichkeit der Daten des Einzelnen Benutzers sollten sehr strikt eingehalten werden. Nur Benutzer die Zugang bzw. sich beim System Authentifiziert haben, sollten nur die Möglichkeit besitzen explizit Einsicht auf ihre Persönliche Daten zu haben. Innerhalb des Systems sollten Sichtbarkeit der Daten durch Rechte/Rollen Zuweisung Einschränkungen zu Verfügung stehen.

#### 4. Erweiterbarkeit

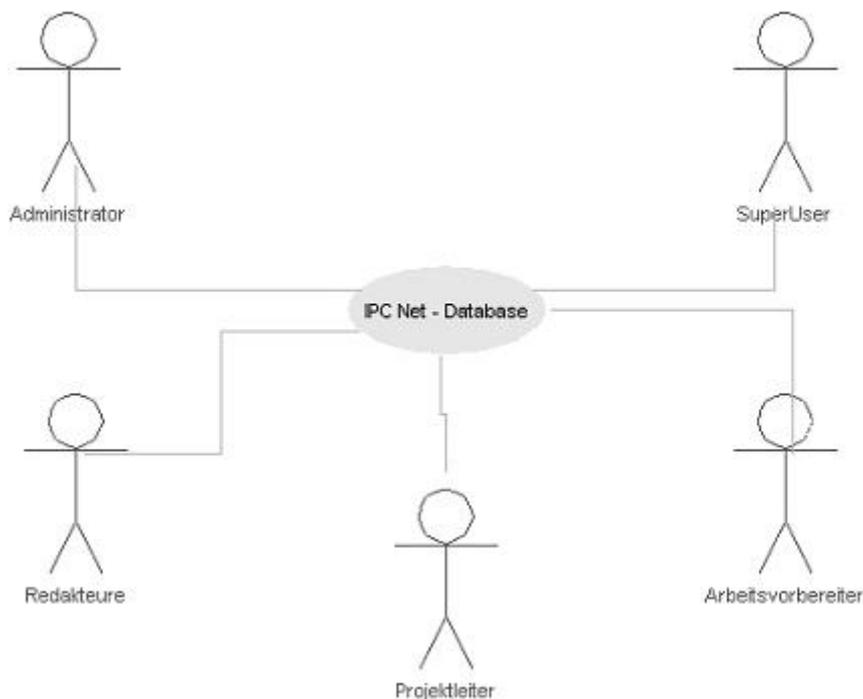
Das System sollte so Modular aufgebaut sein das Erweiterungen jeder Zeit problemlos implementiert werden können.

#### 5. Rahmenbedingungen

Alle Technologien die verwendet werden, sollten die Möglichkeit bieten die Java EE Technologien zu verwenden. Zusätzlich sollten das Report Tool die Möglichkeit bieten die erforderlichen Daten für die Reports aus den EJB zu erstellen und in gängige Formate wie PDF-, Word- und Excel- exportieren zu können.

### 3.5 Akteure

In diesem Abschnitt werden Akteure für das neue System und ihrer Aufgaben definiert. Anhand von UseCase Diagrammen, werden die Akteure zusätzlich veranschaulicht. Die genaue Beschreibung der einzelnen Aufgaben erfolgt im Kapitel 3.6.



### Arbeitsvorbereiter

Der „Arbeitsvorbereiter“ kann auftragsbezogenen Daten und Information zur Selbstverwaltung seiner Firma anlegen d.h. er kann Kapitel, AWO's und Time Schedule anlegen und einem Redakteure zuweisen.

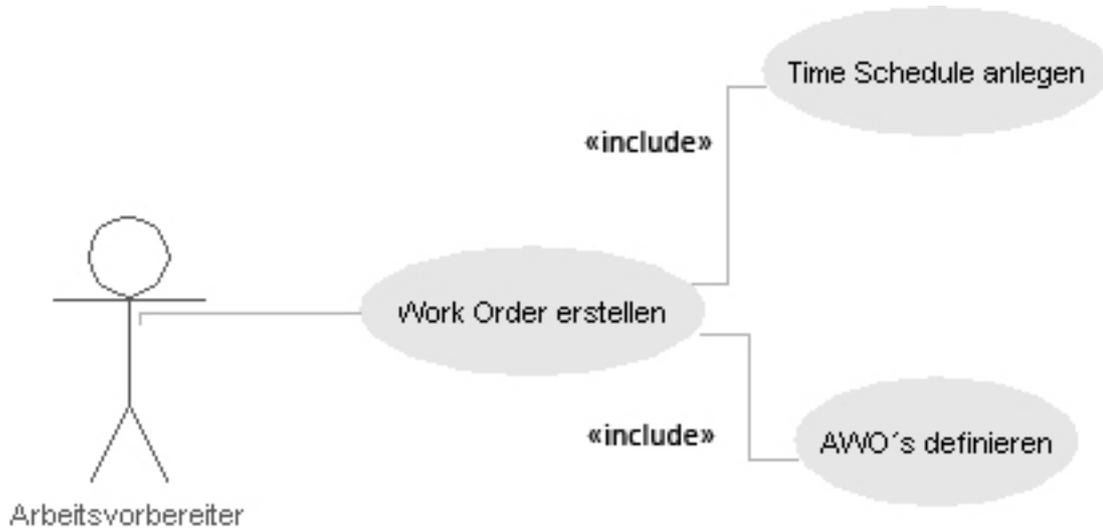


Abbildung 3.4: Akteure Arbeitsvorbereiter

### Redakteure

Der „Redakteur“ ist in der Lage, die auftragsbezogenen Daten und Informationen zur Selbstverwaltung, die im vom Arbeitsvorbereiter zugeordnet worden sind, zu bearbeiten oder zu löschen. Des Weiteren können die Redakteure Aufträge abschließen, Aufträge zur Abrechnung vorbereiten und die jeweiligen Figuren, AWO's und Kapitel, als erledigt markieren. Darüber hinaus hat er Einsicht auf seinem persönlichen Basket.

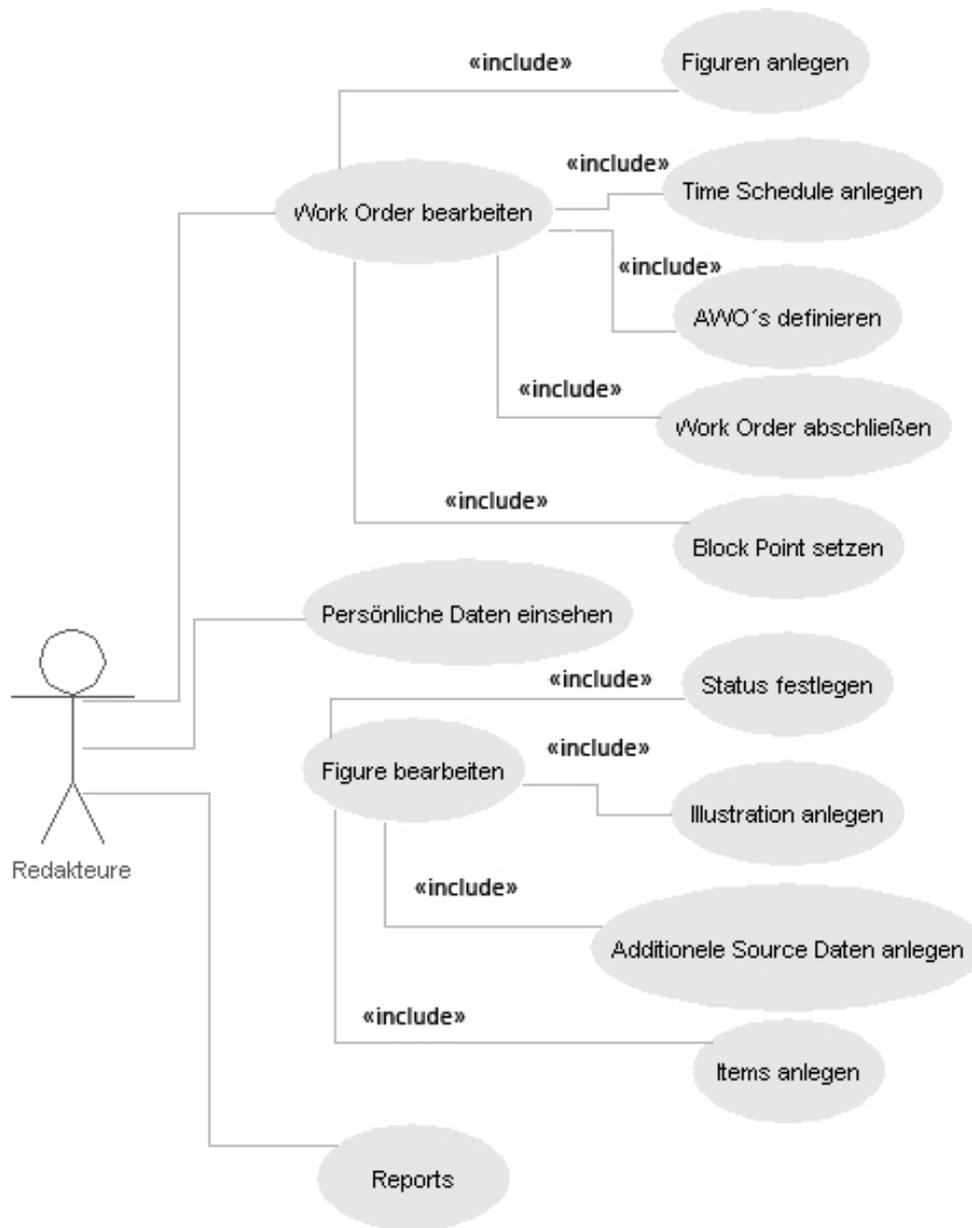


Abbildung 3.5: Akteure, Redakteure

### Projektleiter

Der „Projektleiter“ verfügt über die gleichen Berechtigungen, wie die Redakteure und hat darüber hinaus, die komplette Übersicht über die persönlichen Daten der Mitarbeiter in seiner Firma. Außerdem kann der „Leiter“ Aufträge zur Abrechnung vorbereiten und in deren Fertigstellungsgrade einsehen.

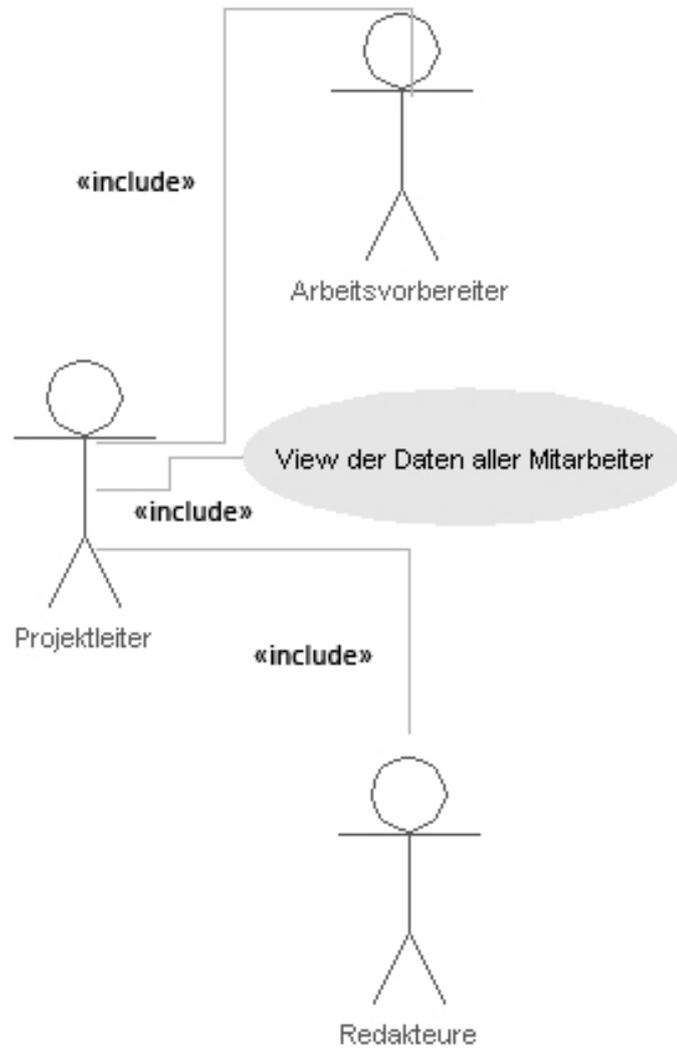


Abbildung 3.6: Akteure Projektleiter

### Administrator

Der firmenspezifische Administrator kann User für die Firma anlegen, löschen und verfügt darüber hinaus über dieselben Berechtigungen wie der „Projektleiter“.

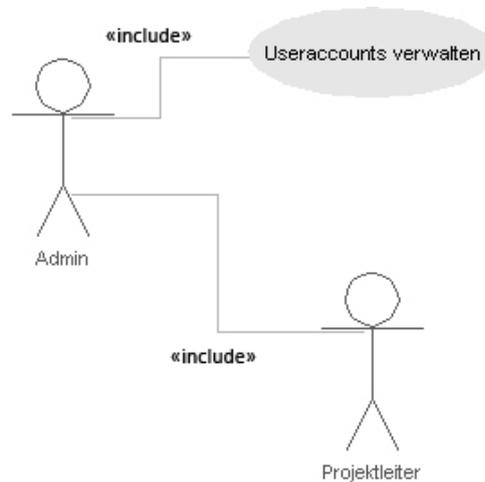


Abbildung 3.7: Akteure Admin

### SuperUser

Der SuperUser ist der Ansprechpartner bei technischen Problemen. Er sorgt dafür, dass das Informationssystem zur Verfügung steht und richtet für die jeweiligen Firmen firmenspezifische Administratoren ein. Des Weiteren kann der SuperUser bei Bedarf sämtliche Daten des Informationssystems bearbeiten.

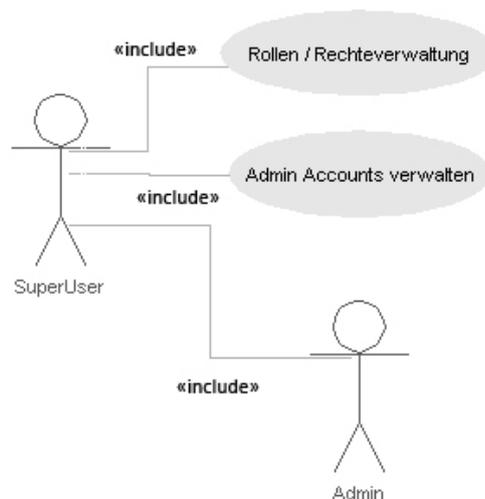


Abbildung 3.8: Akteure SuperUser

## 3.6 Nutzungsszenarien

### 3.6.1 Kapitel

#### Erstellen

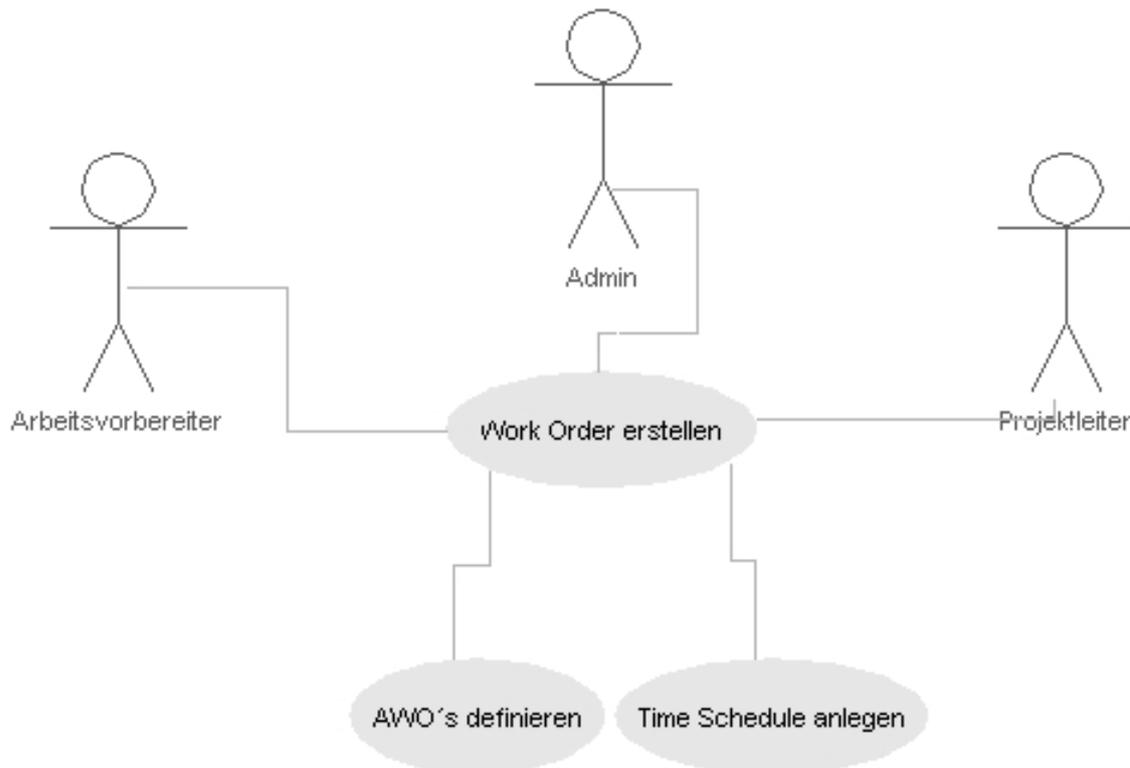


Abbildung 3.9: Nutzungsszenario Kapitel Erstellen

Bevor ein Arbeitsvorbereiter, Projektleiter oder der Admin ein Kapitel (WorkOrder) und die dazugehörigen Aufträge (AWOs) im neuen System anlegen kann, muss er sich die Daten aus dem SAP-System manuell beschaffen. Im Anschluss muss der Akteur sich erfolgreich am System authentifizieren. Um ein Kapitel erfolgreich anzulegen, muss dem Auftrag ein Benutzer zugewiesen werden, der für die weitere Bearbeitung zuständig ist. Anschließend ist es ihnen möglich, zu dem erstellten Kapitel, einen AWO's und einen Timeschedule anzulegen.

### 3 Analyse

Name:	<b>Kapitel erstellen</b>
Kurzbeschreibung:	Ein Kapitel(WorkOrder) wird angelegt und einem Benutzer zugewiesen
Akteur:	Arbeitsvorbereiter, Projektleiter, Admin
Auslöser:	Auswahl des Menüpunkts Kapitel erstellen(Create Work Order)
Vorbedingungen:	Akteur hat sich erfolgreich am System Authentifiziert, Kapitel existiert im SAP System
Eingehende Information:	Daten des Kapitel(Firma, Projekt, Kapitel Nr, Revisonstart-/Enddatum,AWO - LBU - MOD Nummer)
Ergebnisse:	Kapitel, Kapitelzuweisung
Nachbedingungen:	Work Order Nummer wird generiert. Der Auftrag wird im persönlichen Basket des Benutzers angezeigt. Akteure haben zusätzlich die Möglichkeit AWO´s und einen Timeschedule zu dem erstellten Kapitel hinzuzufügen.
Ablauf:	<ol style="list-style-type: none"> <li>1. Akteur holt sich die Daten aus dem SAP System und trägt sie in dem neuen System ein</li> <li>2. Validierung: Bei nicht Vervollständigung der Pflichtfelder wird der Akteur vom System hingewiesen sie auszufüllen.</li> <li>3. Kapitel erstellen: Kapitel wird erst erstellt wenn es noch nicht im System existiert.</li> <li>4. Zugewiesener Benutzer erhält eine Nachricht des zugewiesenen Kapitels.</li> <li>5. Auftrag(AWO) erstellen: Akteur hat nun zusätzlich die Möglichkeit dem Kapitel Aufträge hinzuzufügen.</li> <li>6. Timeschedule erstellen: Akteur hat nun zusätzlich die Möglichkeit dem Kapitel ein Timeschedule hinzuzufügen</li> </ol>

Tabelle 3.1: Nutzungsszenario Kapitel erstellen

**Verwalten**

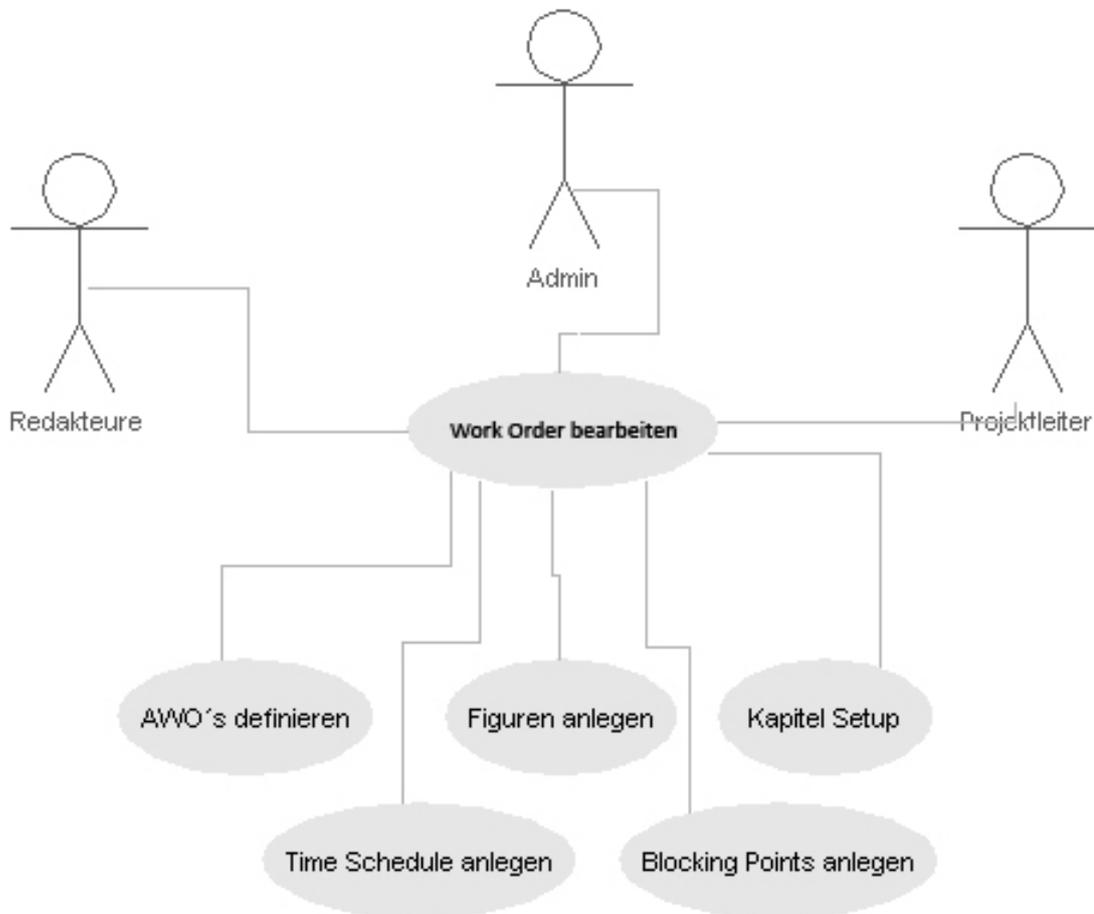


Abbildung 3.10: Nutzungsszenario Kapitel Verwalten

Bevor die Akteure, Redakteure, Projektleiter oder Admins die Möglichkeit haben, die ihm zugewiesen Kapitel zu verwalten, muss der Akteur sich erfolgreich gegenüber dem System authentifizieren.

### 3 Analyse

---

Name:	<b>Kapitel bearbeiten</b>
Kurzbeschreibung:	Der Akteur bearbeitet Kapitel die ihm zugeordnet worden sind
Akteur:	Redakteur, Projektleiter, Admin
Auslöser:	Durch auswählen des Edite Button des jeweiligen Kapitels
Vorbedingungen:	Akteur wurde erfolgreich am System Authentifiziert. Zu bearbeitender Kapitel wurde ausgewählt.
Eingehende Information:	Daten ändern
Nachbedingungen:	Kapitel wurde bearbeitet
Ablauf:	<ol style="list-style-type: none"><li>1. Kapitel wurde ausgewählt</li><li>2. Bearbeitung/hinzufügen von Daten: Aufträgen, Timeschedule, Figure, Chapter Setup und Blocking Points</li><li>3. Daten werden erst dann angelegt wenn sie noch nicht für die jeweiligen Kapitel existieren</li><li>4. Validierung: Bei nicht Vervollständigung der Pflichtfelder wird der Akteur vom System drauf hingewiesen.</li></ol>

Tabelle 3.2: Nutzungsszenario Kapitel bearbeiten

### 3 Analyse

---

Name:	<b>Kapitel löschen</b>
Kurzbeschreibung:	Kapitel wird aus dem System entfernt und aus dem Basket des jeweiligen Akteur
Akteur:	Redakteur, Projektleiter, Admin
Auslöser:	Kapitel wurde abgerechnet
Vorbedingungen:	Akteure wurde erfolgreich am System authentifiziert. Zu löschendes Kapitel wurde ausgewählt.
Eingehende Information:	Löschungsauftrag
Ergebnisse:	Kapitel wird vom System entfernt
Nachbedingungen:	Kapitel wurde gelöscht
Ablauf:	<ol style="list-style-type: none"><li>1. Kapitel wurde ausgewählt</li><li>2. Akteur wird darauf hingewiesen ob er wirklich das jeweilige Kapitel löschen möchte.</li><li>3. Bestätigen des Löschvorgangs</li></ol>

Tabelle 3.3: Nutzungsszenario Kapitel löschen

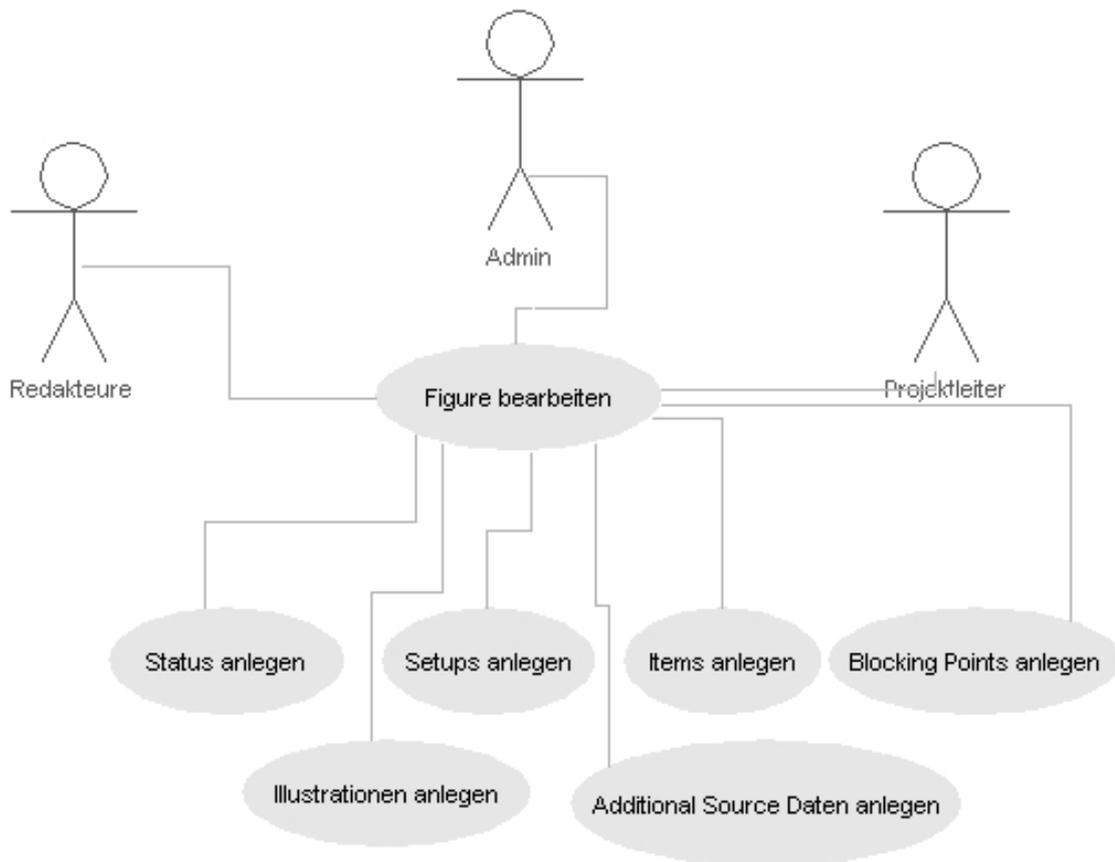


Abbildung 3.11: Nutzungsszenario Figur bearbeiten

Bevor ein Redakteur, Projektleiter oder Admin eine Figur bearbeiten kann, muss der Akteur sich am System erfolgreich authentifizieren. Des Weiteren sollte eine Figur für ein Kapitel existieren/erstellt worden sein. Anschließend ist es dem Akteur möglich, die ausgewählte Figur zu bearbeiten. Der Akteur hat nun die Möglichkeit, die ausgewählte Figur einen Status, Illustrationen, Setups, Items, Additional Source Daten oder Blocking Points zu verwalten.

### 3 Analyse

Name:	<b>Figur bearbeiten</b>
Kurzbeschreibung:	Akteur bearbeitet persönliche Figur
Akteur:	Redakteur, Projektleiter, Admin
Auslöser:	Durch auswählen des Edite Button des jeweiligen Figur
Vorbedingungen:	Akteur hat sich erfolgreich am System authentifiziert. Die zu bearbeitende Figur wurde ausgewählt.
Eingehende Information:	Daten ändern
Nachbedingungen:	Figur wurde verwaltet
Ablauf:	<ol style="list-style-type: none"><li>1. Figur wurde ausgewählt</li><li>2. Bearbeitung/hinzufügen von Daten: Status, Illustrationen, , Items, Additional Source Daten, Setup und Blocking Points</li><li>3. Daten werden erst dann angelegt wenn sie noch nicht für die jeweiligen Figuren existieren</li><li>4. Validierung: Bei nicht Vervollständigung der Pflichtfelder wird der Akteur vom System drauf hingewiesen.</li></ol>

Tabelle 3.4: Nutzungsszenario Figur bearbeiten

Die weiteren, aufgeführten Daten in Kapitel und Figur sind identisch mit den bereits erarbeiteten Nutzungsszenarien. Deswegen gilt auch hier, für die unten aufgeführten Daten von Kapitel und Figur die Nutzungsszenarien erstellen, bearbeiten und löschen. Deshalb werde ich nicht weiter darauf eingehen und sie nicht mehr auflisten.

#### **Kapitel**

- AWO's
- Time schedule
- Blocking Point
- Kapitel Setup
- Figure

#### **Figur**

- Status
- Setup
- Item
- Blocking Point
- Additional Source Data
- Illustration

### 3.6.2 Basket

Name:	<b>Basket ansehen</b>
Kurzbeschreibung:	Der Akteur bekommt einen Überblick auf seine persönlichen Daten
Akteur:	Redakteur, Projektleiter, Admin
Auslöser:	Durch auswählen des Menüpunkts Basket
Vorbedingungen:	Benutzer hat sich erfolgreich am System authentifiziert
Eingehende Information:	Auswahl persönliche Daten
Ergebnisse:	Benutzer hat einen Überblick auf seine persönliche Daten bekommen
Ablauf:	1. Liste der persönlichen Daten werden angezeigt.

Tabelle 3.5: Nutzungsszenario Basket ansehen

### 3.6.3 Admin

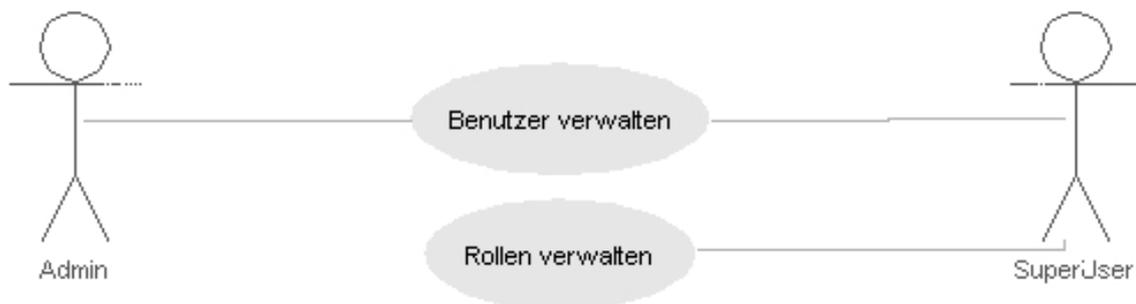


Abbildung 3.12: Nutzungsszenario Admin

Bevor der Akteur, „Admin“ seiner administrative Aufgabe nachgehen kann, muss er sich erfolgreich im System authentifizieren. Dem Admin ist nach der Authentifizierung möglich firmenspezifische Benutzer zu erstellen und ihnen eine Rolle zu zuweisen. Er verwalten diese und kann sie ggf. löschen. Des Weiteren erhält er eine komplette Sicht auf seinen firmenspezifische Benutzer und deren Work Orders.

Der SuperUser dagegen muss dieselben Kriterien wie der Admin erfüllen und hat des Weiteren die Möglichkeit Benutzer-Rollen zu verwalten.

### 3 Analyse

Name:	<b>Benutzer erstellen</b>
Kurzbeschreibung:	Erstellen von Benutzern
Akteure:	Admin, SuperUser
Auslöser:	Durch auswählen des Menüpunkts Admin
Vorbedingungen:	Benutzer hat sich erfolgreich am System Authentifiziert
Eingehende Information:	Daten des neuen Mitarbeiter/Benutzers
Ergebnisse:	Mitarbeiter/Benutzer wurde erstellt
Ablauf:	<ol style="list-style-type: none"> <li>1. Einrichten des Account für den neuen Benutzer</li> <li>2. Benutzer Rolle zuweisen</li> </ol>

Tabelle 3.6: Nutzungsszenario Benutzer erstellen

Name:	<b>Benutzer bearbeiten</b>
Kurzbeschreibung:	Der Akteur bearbeitet die persönlichen Daten eines Benutzers
Akteure:	Admin, SuperAdmin
Auslöser:	Auswählen des Edite Button des jeweiligen Benutzer
Vorbedingungen:	Benutzer hat sich erfolgreich am System authentifiziert. Gewünschter Benutzer ist ausgewählt
Eingehende Information:	zu ändernde Daten
Nachbedingungen:	Persönliche Daten des Benutzers wurden aktualisiert
Ablauf:	<ol style="list-style-type: none"> <li>1. Gewünschter Benutzer auswählen</li> <li>2. Daten des Benutzers verändern</li> <li>3. Daten sichern</li> </ol>

Tabelle 3.7: Nutzungsszenario Benutzer bearbeiten

### 3 Analyse

Name:	<b>Benutzer löschen</b>
Kurzbeschreibung:	Benutzer wird aus dem System gelöscht
Akteure:	Admin, SuperAdmin
Auslöser:	Auswählen des Delete Button des jeweiligen Benutzer
Vorbedingungen:	Benutzer hat sich erfolgreich am System authentifiziert. Gewünschter Benutzer ist ausgewählt
Eingehende Information:	Löschauftrag
Nachbedingungen:	Benutzer wurde erfolgreich gelöscht
Ablauf:	<ol style="list-style-type: none"><li>1. Gewünschter Benutzer auswählen</li><li>2. Bestätigen des Löschvorgangs</li></ol>

Tabelle 3.8: Nutzungsszenario Benutzer löschen

Name:	<b>Benutzer-Rollen verwalten</b>
Kurzbeschreibung:	Benutzer-Rollen bearbeiten
Akteure:	SuperAdmin
Auslöser:	Auswählen des Menüpunkts Admin
Vorbedingungen:	Benutzer hat sich erfolgreich am System authentifiziert. Gewünschte Benutzer-Rolle ist ausgewählt
Eingehende Information:	Aktualisierung der Benutzer-Rolle
Ergebnisse:	Benutzer-Rolle wurde um Rechte erweitert/eingegrenzt
Ablauf:	<ol style="list-style-type: none"><li>1. Gewünschte Benutzer-Rolle auswählen</li><li>2. Rechte erweitern/eingrenzen</li><li>3. Aktualisierung der Benutzer-Rolle bestätigen</li></ol>

Tabelle 3.9: Nutzungsszenario Benutzer-Rollen verwalten

## 3.7 Untersuchung existierende CMS-System

Bei der Untersuchung des Marktes ergab sich eine große Sammlung von CMS-Systeme. Laut CmsMatrix.org (Cms Matrix) sind über 500 CMS-Systeme registriert. Aus diesem Grund fällt die Auswahl auf die größten bekanntesten und beliebteste Open Source CMS-Systeme Typo3, Joomla! und Drupal das von den Kunden laut Benchpark.com ermittelt worden ist(Benchpark).

### 3.7.1 Typo3

Name:	Typo3
Website:	<a href="http://typo3.org/">http://typo3.org/</a>
Online-Demo:	<a href="http://typo3.tecnas.de/typo3">http://typo3.tecnas.de/typo3</a>
Version:	5.0
Programmiersprache:	PHP
Datenbank:	MySQL

Der TYPO3 ist ein, im Jahr 1998 erschienenenes, Open-Source-Entwicklungsframework für mittlere bis große Websites, das von Kasper Skårhøj entwickelt wurde. TYPO3 basiert auf dem Server seitige Skriptsprache PHP. Als Datenbank wird häufig MySQL eingesetzt, möglich sind aber auch etwa PostgreSQL oder Oracle. Das System wird von dem ursprünglichen Autor und einer Community aktiv weiterentwickelt(Andreas Stöckl, 2006).

TYPO3 kann mit Plugins (extensions, Erweiterungsmodulen) um viele Funktionen und ganze Anwendungsprogramme erweitert werden, ohne dass ein eigener Programmcode geschrieben werden muss. Derzeit sind etwa 3000 Erweiterungen verfügbar, darunter fallen, z.B. News, Shop-Systeme und Diskussionsforen. Vorteilhaft ist die Mehrsprachigkeit von Front- und Backend, sowie die aktive Anwender- und Entwicklergemeinde aus aller Welt.

### 3 Analyse

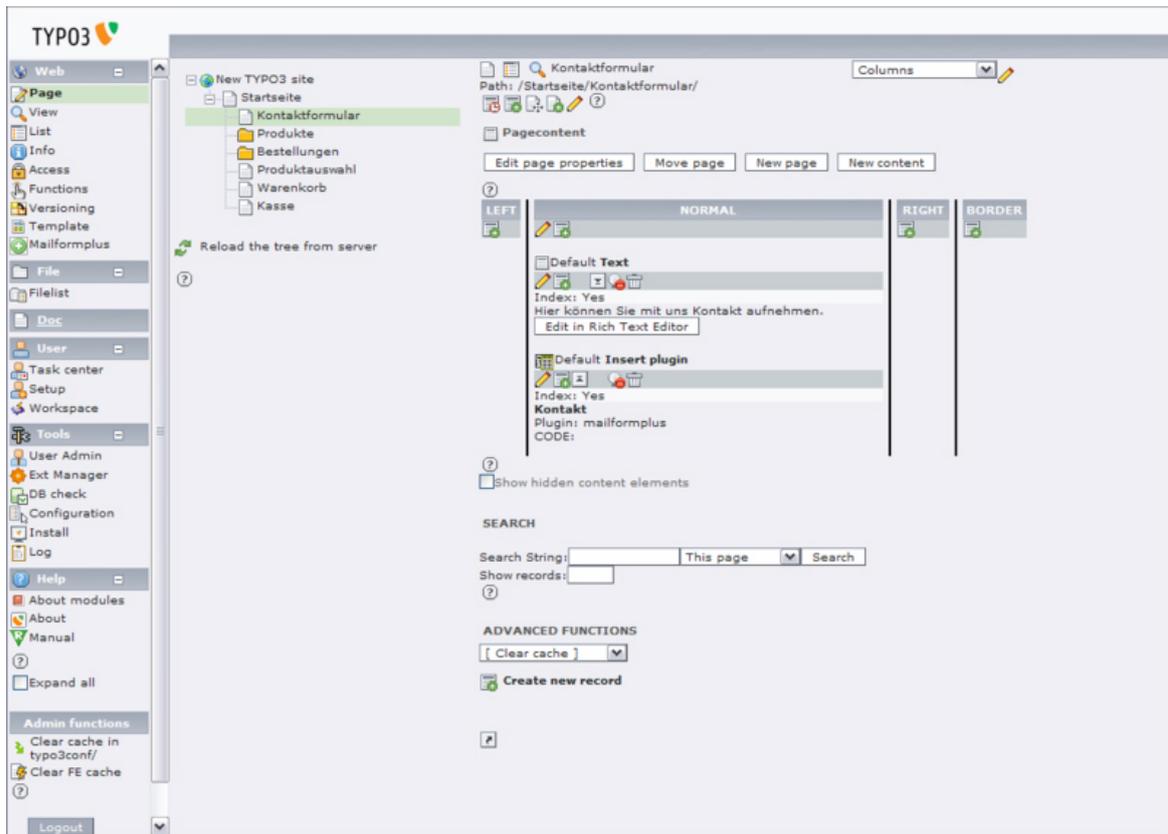


Abbildung 3.13: Typo3

Abbildung 3.13 stellt das CMS System Typ3 dar

### 3.7.2 Joomla!

Name:	Joomla!
Website:	<a href="http://www.joomla.de/">http://www.joomla.de/</a>
Online-Demo:	<a href="http://joomla.tecnas.de">http://joomla.tecnas.de</a>
Version:	5.1.5
Programmiersprache:	PHP
Datenbank:	MySQL

Joomla! ist eine Verzweigung des Open Source CMS Mambo. Dieses wurde im Jahr 2001 von der australischen Firma Miro entwickelt und unter einer Open Source Lizenz freigegeben. Am 17. August 2005 verfasste das ursprüngliche Mambo-Entwicklerteam eine Erklärung für die Mambo Community. In diesem Brief teilen sie der Nutzergemeinde die Trennung

des Teams von der australischen Software-Firma Miro International mit. Nach dieser Trennung im September 2005 entsandt Joomla! 1.0.8, die auf der Open Source Web Content Management System Mambo 4.5.2.3 Code weiterentwickelt wurden ist (Anja Ebersbach).

Joomla! basiert wie TYPO3 auf der Server-seitigen Skriptsprache PHP und verwendet häufiger eine MySQL Datenbank. Auf der Clientseite genügt ein normaler Browser, wobei dieser, den Administrationsbereich, den Editor und den JavaScript unterstützen muss. Joomla! kann wie TYPO3 mit Plugins (extensions, Erweiterungsmodulen) um viele Funktionen und ganze Anwendungsprogramme erweitert werden, ohne dass ein eigener Programmcode geschrieben werden muss.

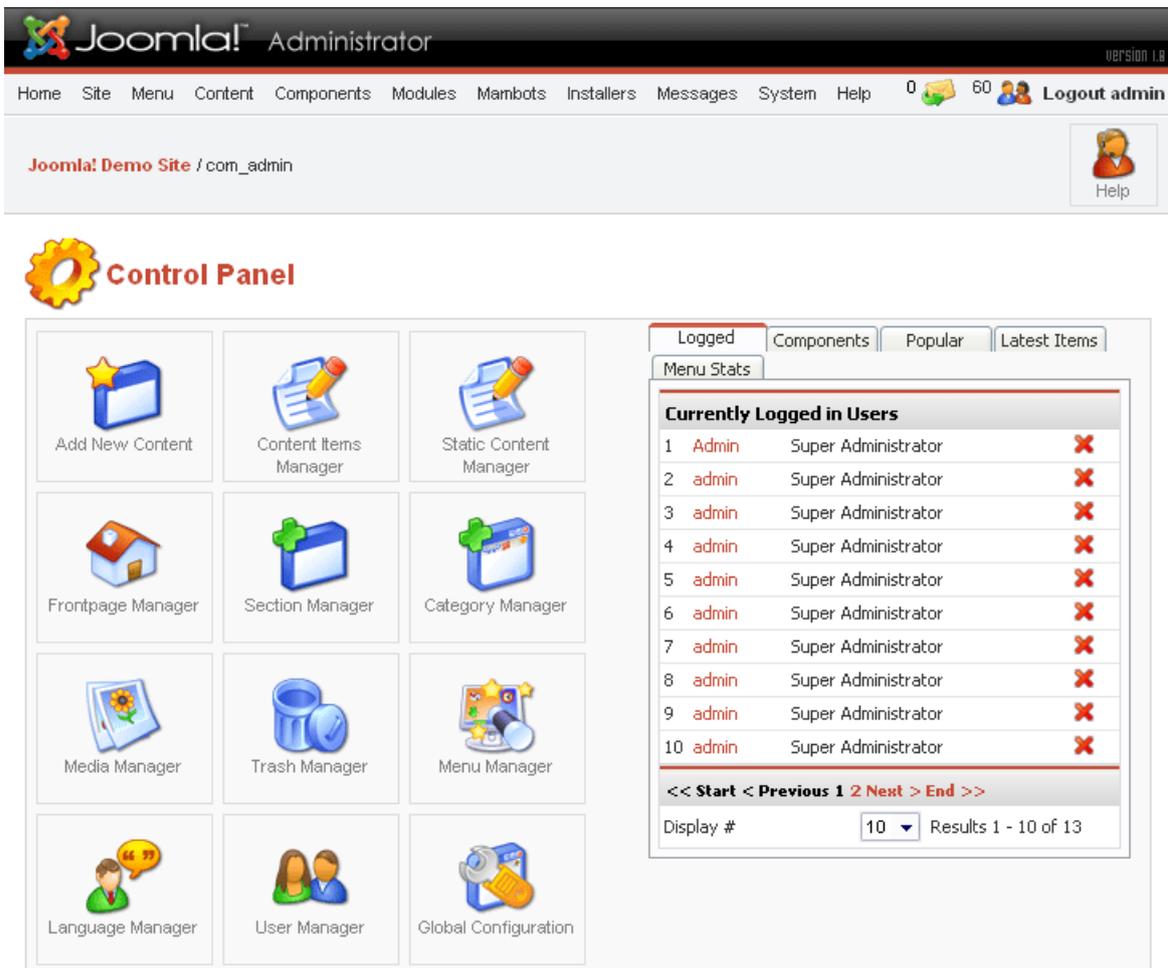


Abbildung 3.14: Joomla!

Abbildung 3.14 stellt das CMS System Joomla dar

### 3.7.3 Drupal

Name:	Drupal
Website:	<a href="http://www.drupal.org/">http://www.drupal.org/</a>
Version:	6.6
Programmiersprache:	PHP
Datenbank:	MySQL

Drupal wurden von den Studenten Dries Buytaert und Hans Snijder entwickelt. Ziel war es, eine News-Seite zu erarbeiten, die es ermöglicht Neuigkeiten und andere Informationen auszutauschen. Im Laufe der Zeit hat sich die Seite vergrößert. Demnach ist auch eine Vielfalt von Anwendungsmöglichkeiten gegeben und mit neuen Features versehen. Im Jahr 2001, hat man sich dazu entschlossen, den Code zu veröffentlichen, um andere dazu zu bewegen, sich an der Entwicklungsarbeit zu beteiligen. So wurde Drupal zu einer Open-Source-Software. Drupal verfolgt ein stark modulares Konzept. Zusätzlich dazu gibt es eine große Menge von Modulen, die von Benutzern beigesteuert wurden. Dadurch erhält Drupal eine hohe Flexibilität, jedoch erhält man keine Garantie, dass alle möglichen Kombinationen von Modulen einwandfrei funktionieren und zu benutzen sind (Mercer, 2006). Sie basiert wie Typo3 und Joomla! auf der Server-seitigen Skriptsprache PH. Im Gegensatz zu Typo3 und Joomla! ist Drupal datenbankunabhängig und unterstützt z.B. MySQL oder PostgreSQL.

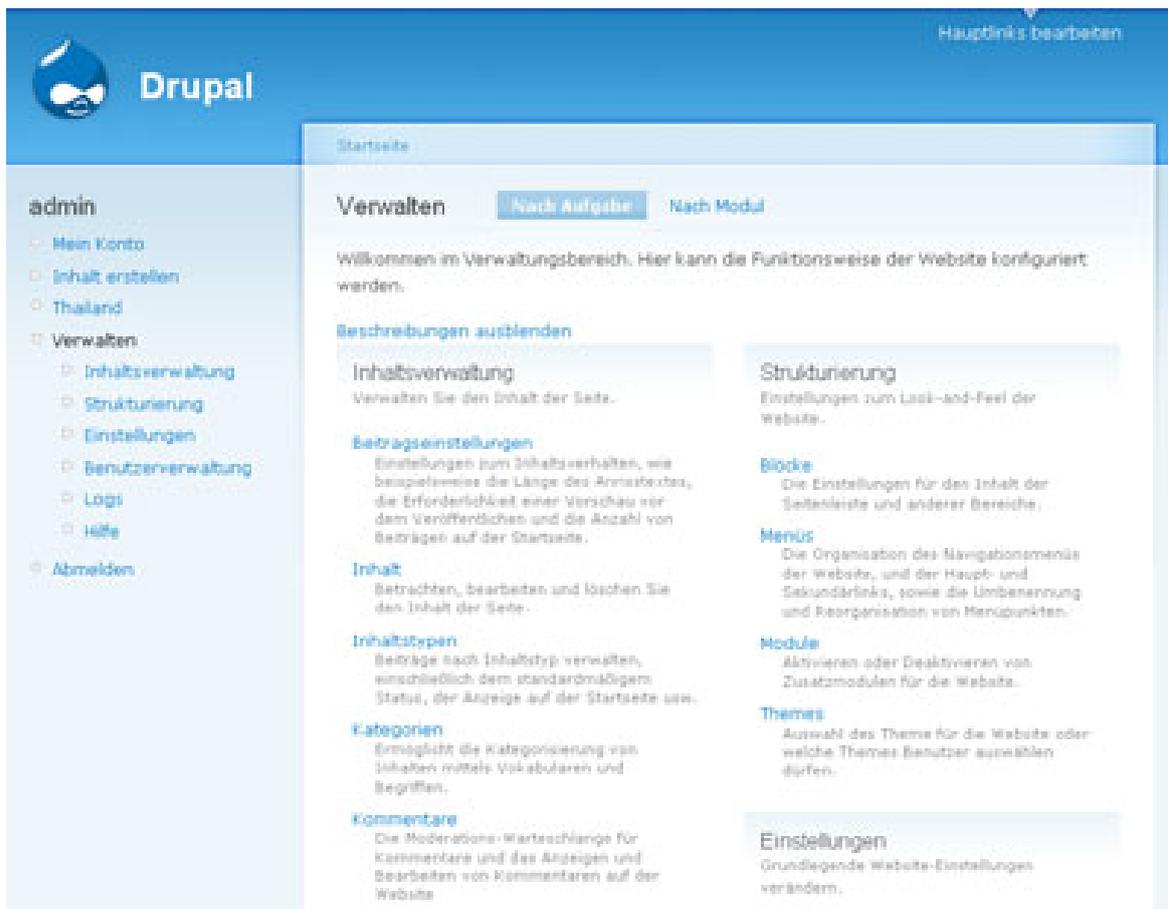


Abbildung 3.15: Drupal

Abbildung 3.15 stellt das CMS System Drupal dar

### 3.7.4 Fazit

Zusammengefasst, bieten die obengenannten CMS-Systeme, die Möglichkeit alle definierten Anforderungen, das neu zu entwickelnde System, zu realisieren. Zu kritisieren ist allerdings, dass diese Systeme meist zu viele Funktionalitäten anbieten, sodass die gesamte Anwendung schwergewichtiger wird. Dies hat zur Folge, dass die Administratoren, die die Administrativen Bereiche betreuen, im Umgang mit dem CMS, umfangreiche Lektüre und je nach Umfang auch eine Schulung im Umgang mit der Software benötigen. Infolgedessen müssen die Administratoren einen hohen Arbeitsaufwand betreiben, um das Verständnis für die Struktur der Systeme zu bekommen, dass wiederum sehr hohe Kosten bedeuten würden. Ein weiterer Aspekt der zu bemängeln ist, ist den Überblick bei der Erweiterung des Systems, durch die großen/unübersichtlichen Auswahl, an verschiedenen Modulen, verlieren zu können und sich mit den einzelnen Modulen auseinander setzen zu müssen.

Ein zusätzlicher Kritikpunkt bei dem CMS-System Typo3 ist, dass die Entwickler sich zusätzlich, der firmeneigenen Scriptsprache, Typoskript, Kenntnisse aneignen müssen, um umfangreiche Layouts anlegen zu können.

Gegenüber Joomla! ist die fehlende bzw. mangelnde Rechteverwaltung zu kritisieren. Administratoren können zwar Rollen erstellen, diesen können aber nur vordefinierten Rechtegruppen zugewiesen werden. Die Rechtegruppen kann man weder anpassen noch selber erstellen. Diese Einschränkung macht es unmöglich z.B. saubere Rechte auf Navigationsebenen, einzelne Inhalte bzw. Bereiche, Module oder Dokumente zu vergeben. Somit könnten die Anforderung Rechte-/Rollenverteilung nicht erfüllt werden.

Abschließend der Analyse der CMS-Systeme, wird dennoch eine spezifische Entwicklung für das neue System, an die neuen Anforderungen angepasst. Die Gründe sind folgende:

Durch einer spezifischen Lösung eines individualisierten Systems, würde eine Plattform entstehen, mit welcher die erforderlichen Informationen und Steuerungen von Prozessen genau an den Bedürfnissen des Auftraggebers abgestimmt und angepasst werden könnten. Darüberhinaus könnten Daten- und Dokumentenredundanz vermieden werden, Zugriffe individualisierter und einheitlicher geregelt werden, beliebige Informationen unabhängig von Quelle und Nutzung bereitgestellt und auch Dienste oder Anwendungen werden gleichförmig zur Verfügung gestellt. Des Weiteren würde ein solches System keine überflüssigen Funktionalitäten bieten, sodass es zu einer schlanken, flexiblen und leichtgewichtigen Lösung führen würde.

Demnach würde ein derartiges System alle Unternehmensinformationen, auf die individuelle Bedürfnisse des Unternehmens zur Verwaltung und Bearbeitung aller unstrukturierten Informationen organisieren, vereinheitlichen, vereinfachen und spezialisiert werden.

Dennoch könnten solch einer Eigenentwicklung Nachteile mit sich bringen, die bei folgenden Punkten sich sichtbar machen könnten:

#### **Zeitverlust**

Eine Eigenentwicklung benötigt unter Umständen viel Zeit, das durch Steigende Entwicklungsaufwendungen bedeutet meist automatisch eine Terminverschiebung.

#### **Qualitätsprobleme**

Software lässt sich nicht fehlerfrei entwickeln. Speziell bei komplexen Lösungen kann es lange dauern, bis nach einer Pilotphase, in der das System unter realen Bedingungen im Einsatz ist, die notwendige Stabilität erreicht ist.

#### **Abhängigkeit**

Bei der Eigenentwicklung würde die Abhängigkeit von den Mitarbeitern steigen, sodass die neu gewonnen Kompetenz des Mitarbeiters sein Marktwert steigen lässt, was sich in steigenden Gehaltsforderungen oder Kündigungen ausdrücken kann. Mit jedem Softwareentwickler, der das Unternehmen verlässt, geht ein Stück des teuer bezahlten Know-hows verloren.

## **3.8 Auswahl eines Report Tools**

Bei der Untersuchung von Report Tools sollten die Aspekte der nichtfunktionalen Anforderungen, der Rahmenbedingung, zur Unterstützung der EJB- Spezifikation und Export des Reports in gängige Formate wie PDF-, Word- und Excel- Dokument berücksichtigt werden.

Bei der Analyse von dem Report Tools, Birt, Pentaho und JasperReport, ist die Entscheidung der Auswahl durch die vordefinierte Aspekte schnell gefallen.

JasperReport ist derzeit das einzige Reportingtool, das die Anforderungen erfüllt. Es bietet eine hohe Anzahl an Ausgabeformaten und die dazugehörige Möglichkeit, auf Daten aus EJBs und Hibernate -Implementierung zurückzugreifen.

Obwohl Birt die Anforderung des Lesens der Daten aus EJBs anbietet, ermöglicht es jedoch nur die Reports in HTML und PDF- Formaten zu exportieren.

Aus diesem Grund fällt die Entscheidung auf das Report Tool „JasperReport“.

## 4 Entwurf

In diesem Kapitel wird der Architekturentwurf des neu zu entwickelnden Systems aus der fachlicher und der technischer Sicht beschrieben. Dazu werden die Komponenten und deren Beziehung untereinander beschrieben.

### 4.1 Fachliche Architektur

Das neu zu entwickelnde System teilt sich in folgende Komponenten auf:

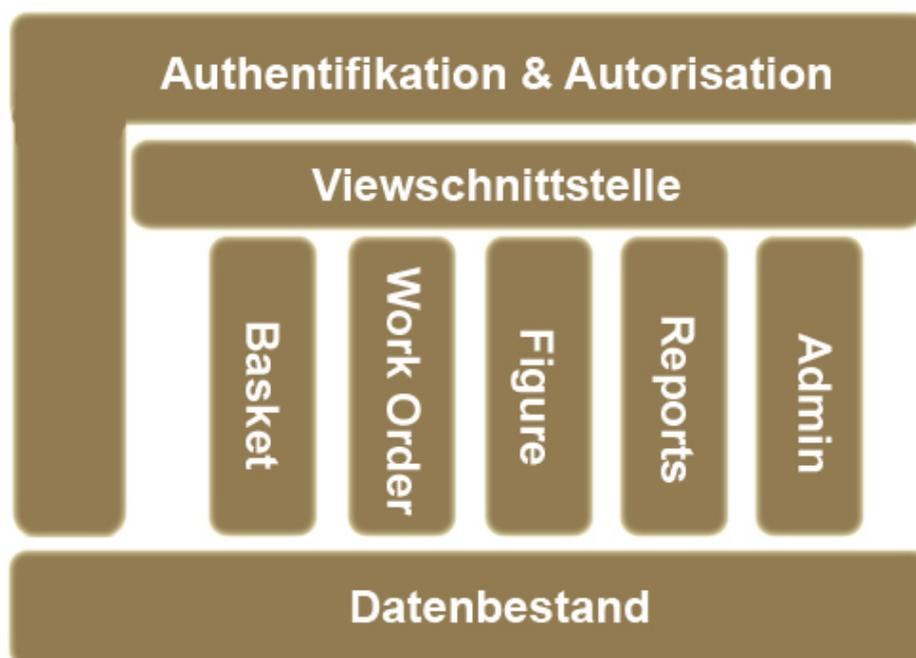


Abbildung 4.1: Komponenten

### **Authentifikation & Autorisation**

ist für die Authentifizierung des Benutzers gegenüber dem System verantwortlich. Dies geschieht mittels der Logins Seite, auf dem der Benutzer beim ersten Zugriff auf das System umgeleitet wird. Des Weiteren wird bei der Autorisation die Rolle des jeweiligen Benutzers überprüft und ihm durch erfolgreichen Login der Zugang erteilt oder verweigert.

### **Die Komponenten**

Basket, Auftrag, Figur, Report und Admin bieten die Möglichkeit, die dazugehörigen Daten vom System durch den Datenbestand einzusehen, löschen oder zu verwalten. Außerdem ist es auch möglich, durch die Admin Komponente Einstellung an dem System vorzunehmen, wenn der Benutzer die Rechte-/ Rolle zugewiesen bekommen hat. Dazu gehören die Rechte-/Rollenverwaltung, Benutzerverwaltung. Die Komponenten werden im Kapitel Klassenmodell 4.2.2 näher erläutert.

### **Datenbestand**

ist dafür zuständig für die dauerhafte Speicherung der bereits angelegten Daten und der Speicherung von neuen Daten. Die anderen Komponenten greifen lesend bzw. schreibend auf diese zu. Der Datenbestand bildet somit die Basis des zu entwickelnden Systems.

### **Viewschnittstelle**

durch die Viewschnittstelle wird dem Benutzer der komplette Funktionsumfang seiner Rolle zur Verfügung gestellt.

## **4.1.1 Datenmodell**

In diesem Abschnitt wird die Anwendungslogik des Datenbank-Schemas erläutert. Wegen der Komplexität des Datenmodells, werden die Kernsegmente in ein Modul für

### **Firmenprofil**

das die Beziehungen der Benutzer und den Mitarbeiter darstellt.

### **Auftrag**

das die Beziehungen zwischen dem einzelnen Unterabschnitte der Aufträge verdeutlicht.

und **Figur** die, die Beziehung deren Unterabschnitte und die Beziehung zwischen den Work Orders darstellt.

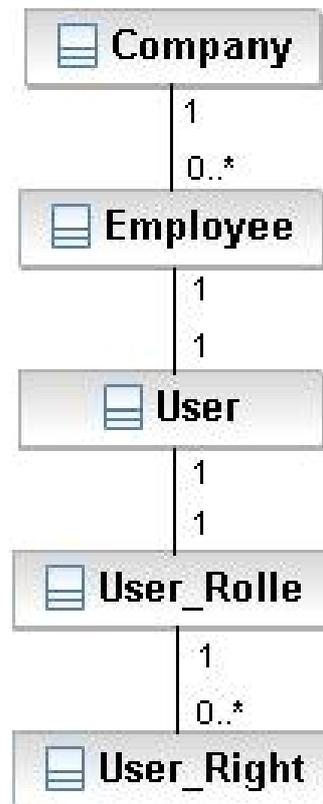


Abbildung 4.2: Firmen Struktur

Abbildung 4.2 zeigt das Modell der Firmen-Struktur. Der SuperUser ist für die Verwaltung der Firmen(Company) zuständig und für die Erstellung der firmenspezifische Administratoren. Außerdem ist er auch für die Verwaltung der Rollen(UserRolle) und deren Rechte(UserRight) zuständig. Eine Firma kann mehrere Mitarbeiter(Employee) haben, die ausschließlich nur für die Firma tätig sind. Der firmenspezifische Administrator hat die Möglichkeit für seine Firma Mitarbeiter(Employee) mit einer der vordefinierten Rollen(UserRole) zu erstellen und sie zu verwalten. Ein Mitarbeiter(Employee) wird als Benutzer(User) im System eingerichtet.

Der Vorteil einer solchen Einstellung ist, dass das System durch einen SuperUser immer einen firmenspezifischen Administrator anlegen muss, der wiederum seine Firma verwaltet. Zudem ist durch die Rollenverwaltung gewährleistet, dass jeder Mitarbeiter bestimmte Rechte besitzt und nur auf seine persönlichen Daten Zugriff erhält.

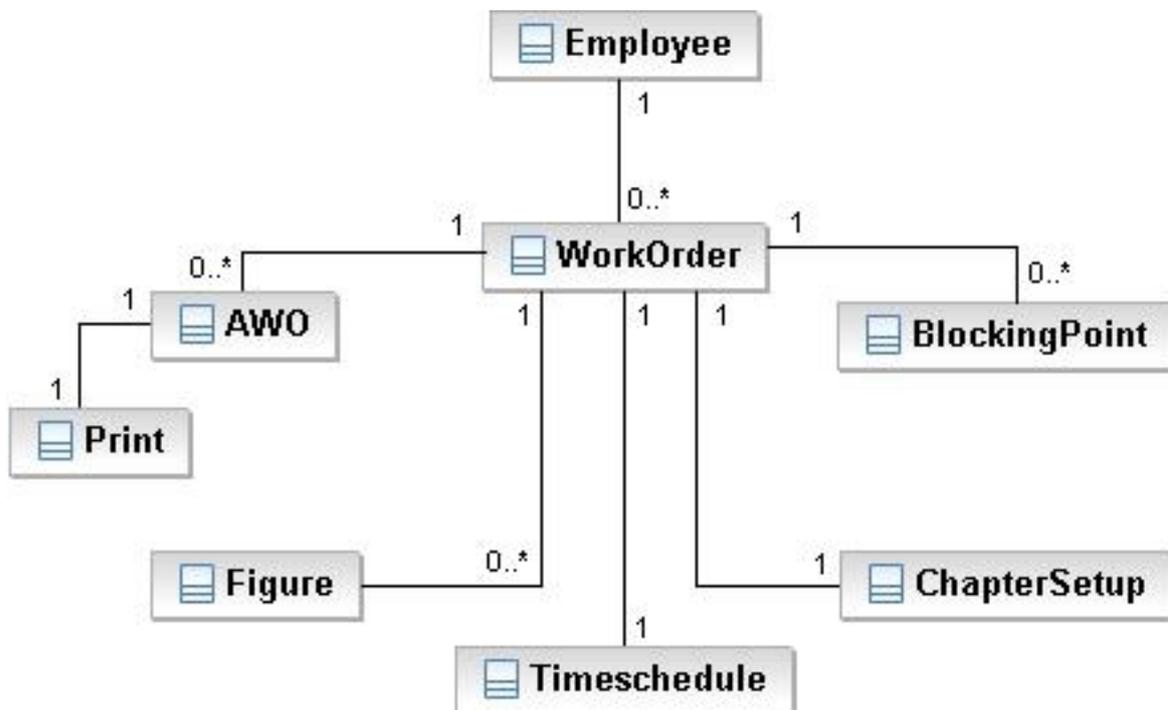


Abbildung 4.3: Auftrag Struktur

Abbildung 4.3 zeigt das Modell der Kapitel(Work Order)-Struktur. Ein Mitarbeiter(Employee) kann mehrere Kapiteln(WorkOrders) für seine Firma(Company) haben. Ein Kapitel besteht aus keine oder mehrere AWO's, die einen Print besitzt, aus keine oder mehrere Figuren, aus einen oder keinen Timeschedule, aus einen ChapterSetup und aus keinen oder mehreren BlockingPoints.

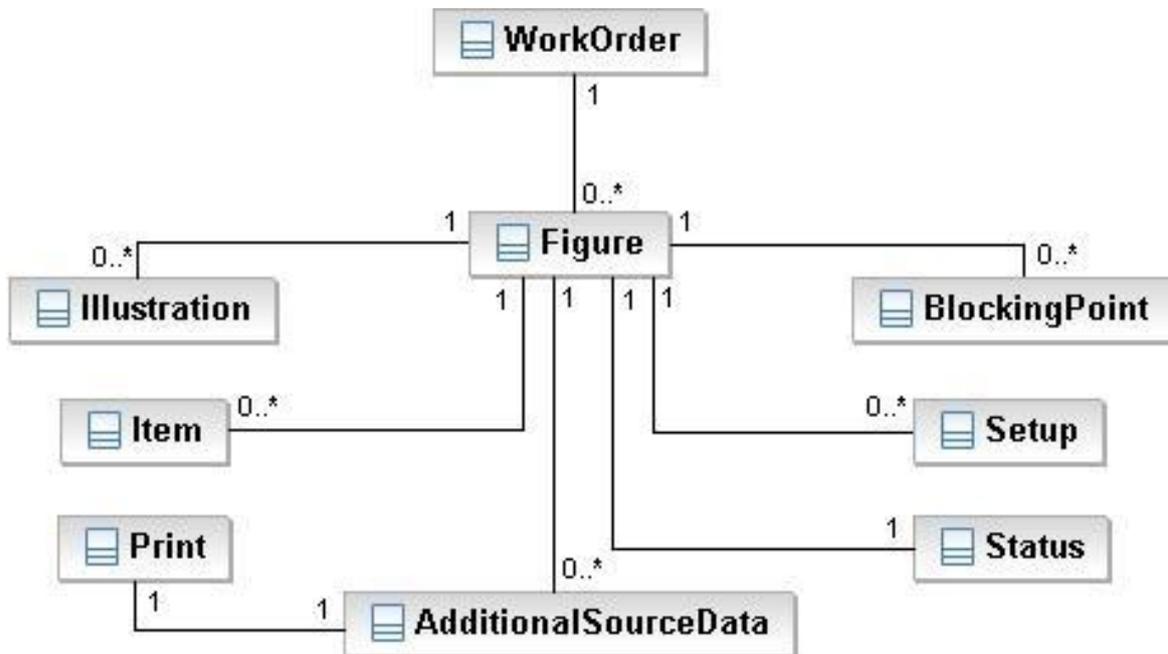


Abbildung 4.4: Figur Struktur

Abbildung 4.4 zeigt das Modell der Figur-Struktur. Ein Kapitel (Work Order) kann mehrere Figuren besitzen. Eine Figur kann einen Status und aus kein oder mehrere Illustrationen, Blocking Points, Setups, Additional Source Data, Items bestehen.

#### 4.1.2 Rollen-/Rechtesystem

Da das System über eine rollenbasierte Benutzerverwaltung verfügen soll, werden im Folgenden die benötigten Rollen aus Kapitel 3 realisiert.

Die wichtigste Benutzerrolle im System, ist die Rolle des SuperUsers. Diese Rolle ist in der Lage, firmenspezifische Administratoren anzulegen, zu verwalten und ggf. zu löschen. Des Weiteren dürfen diese Benutzer neue Benutzer im System registrieren und entfernen. Zur Zeit der Auslieferung des Vertriebssystems muss mindestens ein Benutzer eingerichtet sein, der diese Rolle besitzt, da sonst eine Konfiguration des System und die Einrichtung neuer Benutzer nicht möglich ist.

Das Rechtesystem des neuen Systems soll die Möglichkeit haben dynamisch und Mehrfachverteilung von Gruppierungen bereit zu stellen.

Aus diesem Grund wird ein Rechtssystem implementiert das auf eine Bitmaske mit Bitoperationen basiert. Der Vorteil dieses Rechtssystem ist, dass es nicht hierarchisch strukturiert

### Realisierung

Um dieses Rechtssystem zu realisieren verwendet man für jedes Recht eine Bitmaske. Durch setzen von einzelnen Bits können einzelne Rechte definiert werden.

Recht 1: 0000 0001  
Recht 2: 0000 0010  
Recht 3: 0000 0100  
Recht 4: 0000 1000

Anhand dieses Beispiels sieht man, das man in einem Byte 8 verschiedene Rechte speichern kann. Sollten wir mehr Rechte benötigen, brauchen wir einen größeren Datentypen, das Prinzip bleibt aber das gleiche.

Dabei repräsentieren die einzelnen Bits von rechts die Dualzahlen:

$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$   
0000 0001 = 1

Wir können nun einzelne Bits durch die Bit Operationen OR setzen, AND vergleichen und durch XOR entfernen.

0000 0100 (4) OR  
0000 0001 (1)  
-----  
0000 0101 (5)

Wie man hier sieht, haben wir 2 Zahlen mit dem Bitoperator OR verknüpft und somit das erste Bit gesetzt. Alle anderen Bits bleiben so wie gehabt. Die Zahlen in den Klammern zeigen die Dualzahlen an.

0000 0101 (5) AND

0000 0100 (4)

---

0000 0100 (4)

Wollen wir nun prüfen, ob ein bestimmtes Bit gesetzt ist, verknüpfen wir unsere Zahl mit der Bit folge, welche das gesetzte Bit enthält. Wenn wieder unsere Prüfbitfolge rauskommt, ist das Bit gesetzt.

0000 0101 (5) XOR

0000 0100 (4)

---

0000 0001 (1)

Mit XOR können wir bestimmte Bits nun entfernen. Verknüpfen wir unsere obere Zahl mit dem Recht, das wir entziehen wollen(unter Zahl), bekommen wir dann unser neues Recht mit dem entzogenen Recht.

Nun können wir den jeweiligen Rollen dynamisch Rechte zuweisen oder entfernen, und die jeweiligen Dualzahlen im System für die Rolle hinterlegen.

## 4.2 Technische Architektur

### 4.2.1 Drei Sichten Architektur

Das neue System wurde auf einer Drei-Schichten Architektur aufgebaut. Um eine Drei-Schichten Architektur gewährleisten zu können, wird das System in ihre jeweiligen Aufgabengebiete eingeteilt, sodass das System in Präsentation, Business Logik und Datenhaltung Komponenten unterteilt wird (siehe Abbildung 4.5). Diese Architektur bietet vor allem Vorteile bei der Entwicklung und Wartung, da man die Funktionalität der einzelnen Schichten relativ einfach austauschen kann, ohne die anderen auch mit ändern zu müssen.

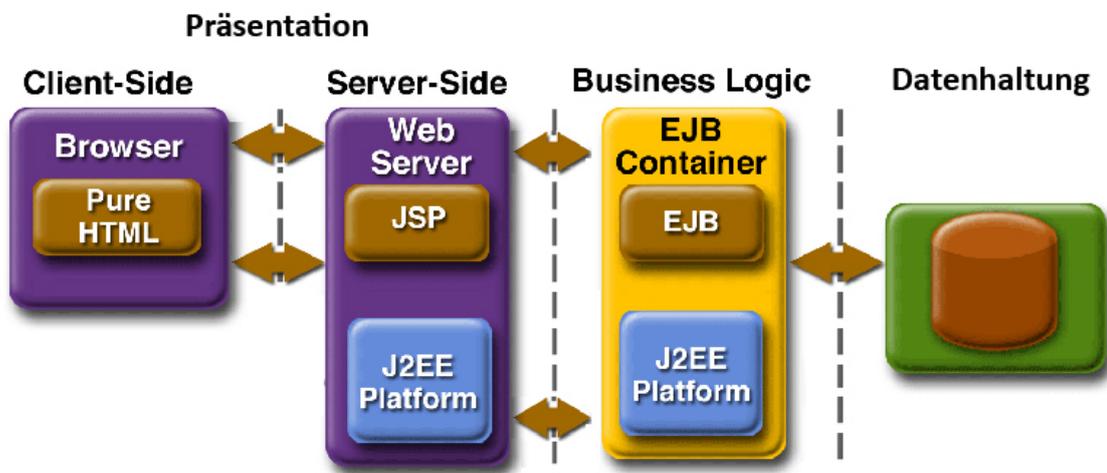


Abbildung 4.5: Schichten Architektur

#### Präsentation

Anhand der Abbildung 4.5 wird es deutlich, dass die Präsentationssicht sich in eine Serverseitige und in eine Client-seitige Komponente aufteilt. Die Server-Seite ist für die Generierung der Webseiten (JSP) zuständig. Die Client-Seite hingegen ist dafür zuständig die Generierten JSP Seiten in puren HTML darzustellen.

#### Business Logik

Die Business Logik beinhaltet die Komplette Logik des Systems die in Java-Klassen repräsentiert werden.

#### Datenhaltung

Für die Datenhaltung wird ein relationales Datenbanksystem (RDMS) eingesetzt. Die Business Logik, die auch die Persistenzabbildung beinhaltet, wird in einem Anwendungsserver ablaufen. Die Kommunikation der Business Logik mit der Datenbank erfolgt über JDBC

Schnittstelle.

Durch die Aufteilung des Systems in der oben genannten Architektur hat den Vorteil, dass die Wartbarkeit und Skalierbarkeit erhöht werden. Somit wird durch solch einer Aufteilung leicht möglich, die einzelnen Schichten auf verschiedene Computer zu verteilen. Der Datenbestand kann somit auf einem separaten Computer verlagert werden, um die Performance des Systems zu steigern. Auch können einzelne Schichten unter Einhaltung der Schnittstellen ausgetauscht werden. Die webbasierte Präsentationsschicht, könnte z.B. durch einen Java Client ersetzt werden, ohne das Anpassungen in den übrigen Schichten benötigt werden.

### 4.2.2 Klassenmodell

Das Paketdiagramm (siehe Abbildung 4.6) zeigt die Aufteilung der Klassen des Systems in Pakete und deren Beziehungen untereinander und zeigt daher die statische Sicht auf das System.

Das Paket Entities beinhaltet alle identifizierten Datenobjekte, die einfache Java- Objekte sind, aber vom Anwendungs-Server auch für die Persistenzabbildung genutzt werden. Diese Objekte werden durch das gesamte System transportiert und daher von den anderen Pakten importiert. Sie werden in den Schichten Anwendungslogik und Präsentation genutzt. Als Beispiel sei hier die Klasse `WorkOrder` genannt, die im System ein abgebildetes Kapitel und ihrer Aufträge darstellt. Die Klasse bietet nur Getter und Setter Methoden an, um Zugriff zu ihren Variablen zu ermöglichen.

Das Paket Services beinhaltet die Interfaces und Klassen, die für die Präsentationsschicht Methoden zur Verfügung stellen, die z.B. das Lesen aus und das Schreiben in die Datenbank realisiert werden. Nach Außen wird die Implementierung eines Services nur durch sein Interface angesprochen und bleibt somit austauschbar. Diese Interfaces bieten zudem eine Fassade, die Implementierung vor den oberen Schichten versteckt. Dieses Paket bildet die Anwendungslogik-Schicht. Als Beispiel sei hier das Interface `WorkOrderServiceBean` genannt, die Methode definiert, um z.B. `WorkOrders` aus der Datenbank zu lesen, in die Datenbank zu schreiben oder aus der Datenbank zu löschen. Die Klasse `WorkOrderServiceBean` implementiert dieses Interface und realisiert dessen Methoden. Präsentation enthält die Klassen, die in der Präsentationsschicht notwendig sind, um auf die Services zuzugreifen und die Daten anzuzeigen oder anzulegen. Die Klasse `ServiceLocator` implementiert das Service Locator Pattern (Sun Microsystems Inc., 2002) und bietet somit einen zentralen Zugriff auf die Services. Die Seiten (JSF), die angezeigt werden, sind in dieser Darstellung ausgelagert, gehören jedoch zur Präsentationsschicht. Das Paket Präsentation bildet zusammen mit den JSF Seiten die Präsentationsschicht.

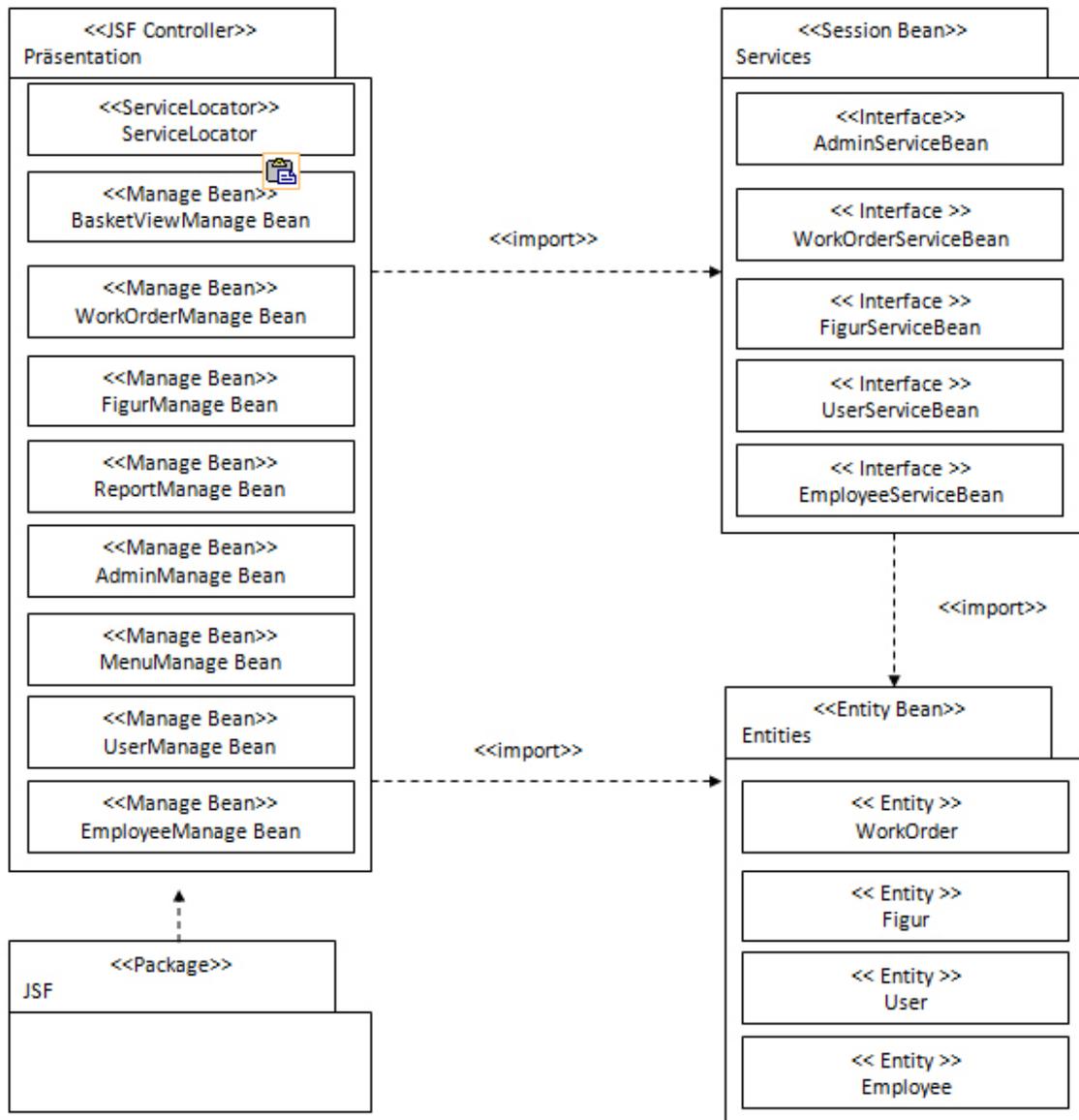


Abbildung 4.6: Paketdiagramm

Abbildung 4.6 zeigt die Aufteilung der Klassen des Systems in Pakete und deren Beziehungen untereinander und zeigt daher die statische Sicht auf das System.

# 5 Realisierung

Die Aufgabe dieser Arbeit ist ein bestehendes Vertriebssystem durch ein neu konzipiertes webbasiertes System abzulösen.

Der Prototyp wird mit dem Ziel entwickelt, die entwickelten Konzepte des in den vorigen Kapiteln einzubringen. In fachlicher Hinsicht sollen mögliche Lücken im Konzept aufgedeckt und geschlossen werden. In technischer Hinsicht soll, durch Entscheidungen bei der Implementierung, die Grundlage für das zu entwickeltes System sein.

Der Prototyp wurde als Java EE Anwendung mit webbasierter Oberfläche realisiert. Zum Betreiben der Prototypen ist folgende Ablaufumgebung notwendig. Als Applikation Server wird JBoss in der Version 4.2.2 verwendet. Des Weiteren wird als Datenbank eine MySQL Installation in der Version 5.0 vorausgesetzt. Auf der Client-Seite wird ein Web-Browser benötigt, der JavaScript fähig ist.

## 5.1 Startseite

Die Startseite des Prototypen wurde so entwickelt, dass sie nach einen erfolgreichen Authentifikation einen guten Einstieg in das neue System bietet(siehe Abbildung 5.1).

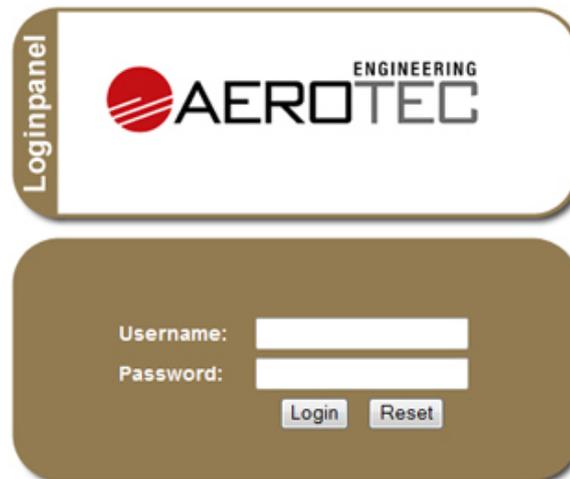


Abbildung 5.1: Login Panel

Die Startseite ist hervorragend geeignet, um wichtige Informationen anzuzeigen, damit jeder Benutzer mit seine Arbeit auf dieser Seite beginnen kann. Die Information die auf der Startseite angezeigt wird ist Rollen abhängig, zu dem Zweck das jeder Benutzer laut seiner Rolle die wichtigste Information erhält. Der Benutzer mit der Rolle „Arbeitsvorbereiter“, dessen Aufgabe ist Aufträge zu erstellen, wird ihm ermöglicht sofort mit seiner Aufgabe zu beginnen und Aufträge zu erstellen (siehe Abbildung 5.2).

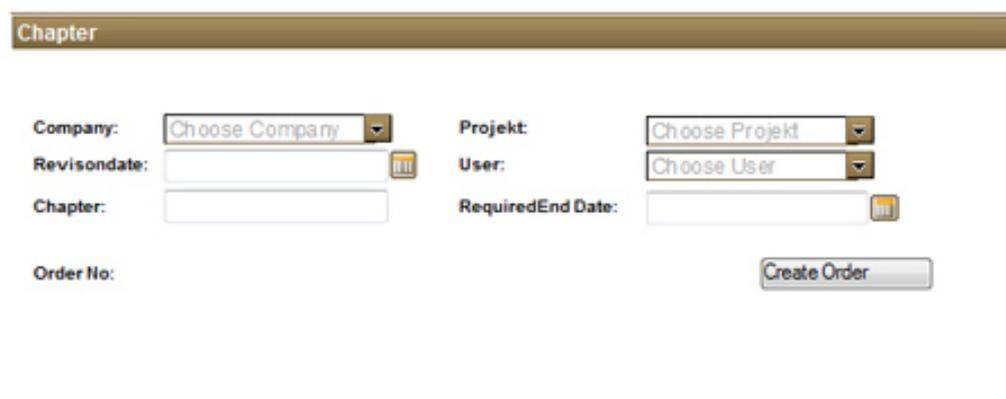


Abbildung 5.2: Startseite Arbeitsvorbereiter

Dagegen sehen die Benutzer mit den Rollen Redakteure, Projektleiter und Admin die Information seiner persönliche auftragsbezogene Daten, somit er durch auswählen / anklicken der Work Orders mit der Verwaltung seiner Daten loslegen kann (siehe Abbildung 5.3).

[My WorkOrder](#)   [My Figure](#)   [My Illustration](#)   [My Awo](#)   [My Proofreading](#)

---

>> My WorkOrder

OrderNo ↕	Author	ChapterNo ↕	Projekt	CreateDate ↕	startDate ↕	endDate ↕	
Aerotec- IPC- 323212- 15.08.2008	Aerotec	323212	IPC	18.08.2008	18.08.2008		<a href="#">Edit</a> <a href="#">remove</a>

---

Abbildung 5.3: Startseite Redakteure, Projektleiter und Admin

## 5.2 Aufteilung in Komponenten

Das neue System wird in Komponenten aufgeteilt, sodass eine logische Trennung vorgenommen werden kann. Dadurch wird die Komplexität der Abhängigkeiten innerhalb des Systems reduziert, die zum einen eine klare Benutzerführung erlaubt, Zyklen vermeidet und sich zum anderen auf die Qualität der Entwicklung bzw. Wartung des Systems niederschlägt. Dies hat den Vorteile sowohl für das Verständnis wie für die Wartung des Systems. Somit wird auch dem geforderten Model-View-Controller Ansatz gerecht.

Somit werden folgende Komponenten im neuen System realisiert:

- Menü - Komponente
- User - Komponente
- Baske t - Komponente
- Work Order - Komponente
- Figur - Komponente
- Report - Komponente

- Admin - Komponente

Jedes dieser Komponente beinhaltet den Umfang an Funktionen, um die gegebenen Nutzungsszenarien abzubilden. Die Komponenten sind auch im Sinne der Objektorientierung mit starker Kohäsion ausgelegt, so dass in jeder Komponente, nur die für diese Aufgaben notwendigen Funktionalitäten zur Verfügung stehen. Eine einheitliche Gestaltung der einzelnen Komponente sorgt für einen hohen Wiedererkennungswert und unterstützt den Anwender bei der Bewältigung seiner Aufgaben. Die angezeigten, komponentenabhängigen Daten werden in Bereiche zusammengefasst, die sich in diesem System Cluster nennen. In diesen Clustern werden verwandte Informationen übersichtlich präsentiert. Eine Durchsetzung der Cluster ist in jedem Modul realisiert.

### 5.2.1 Menu Komponente

Die Menu - Komponente beinhaltet sämtliche Information über den Aufbau der Menustruktur. Die Rolle des eingeloggten Benutzers wird aus der Datenbank selektiert und in einem Menü-Cluster gruppiert. Die Menüpunkte, soweit die Rechte vorhanden sind, werden nach dem Prinzip wie in Kapitel 4.1.2 mit der AND Funktion überprüft. Nachdem die Überprüfungen für die Menüpunkte anhand der Rechte des Menu-Clusters durchgelaufen sind werden die einzelnen Menüpunkte aktiviert oder deaktiviert, sodass dann bei der Deaktivierung nicht mehr möglich ist diesen Menüpunkt anzuklicken. Außerdem wird noch die Navigation der einzelnen Menüpunkte bestimmt, sodass bei der Aktivierung des Menüpunktes die dazugehörige Weiterleitung eingerichtet wird. Bei der Deaktivierung des Menüpunktes wird die Weiterleitung auf die aktuell besuchte Seite eingerichtet.

In der Abbildung 5.4 wird die Menustruktur der Rolle des Admins gezeigt. Die Admin - Rolle besitzt alle Rechte der Menü Punkte, sodass alle Menüpunkte aktiviert und deren Weiterleitung eingerichtet worden sind.

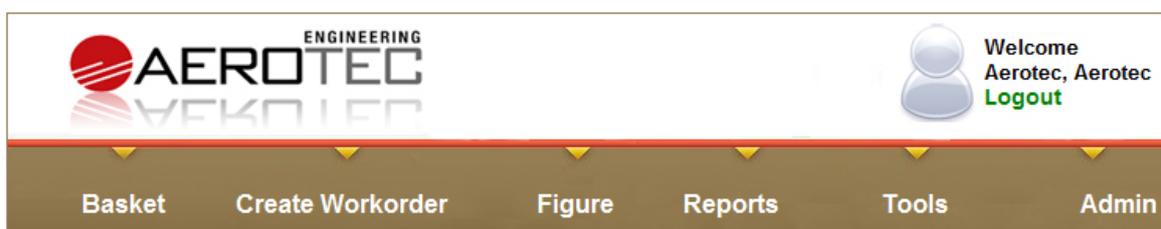


Abbildung 5.4: Menu Admin

In der Abbildung 5.5 wird die Menustruktur der Rolle des Projektleiters gezeigt. Die Projektleiter - Rolle besitzt die Rechte der Menüpunkte Basket, Create WorkOrder, Figur und Reports,

sodass die Menüpunkte aktiviert und deren Weiterleitung eingerichtet werden. Der Menüpunkte Admin wird deaktiviert da die Rolle Projektleiter nicht die Rechte für das Menüpunkte besitzt.

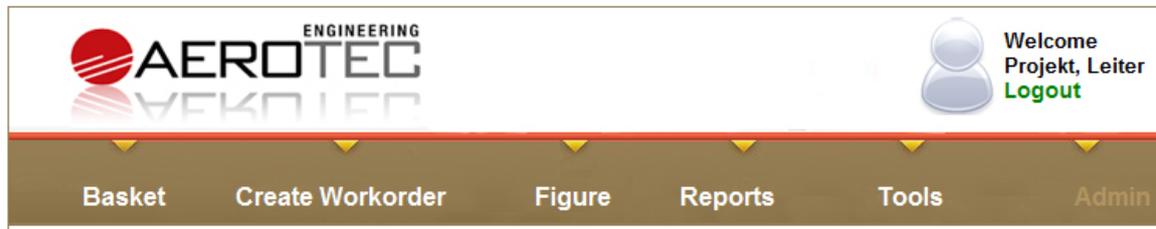


Abbildung 5.5: Menü Projektleiter

In der Abbildung 5.6 wird die Menüstruktur der Rolle des Redakteurs gezeigt. Die Redakteure - Rolle besitzt die Rechte der Menüpunkte Basket, Figur und Reports, sodass die Menüpunkte aktiviert und deren Weiterleitung eingerichtet werden. Der Menüpunkte Admin und Create WorkOrder werden deaktiviert da die Rolle Redakteure nicht die Rechte für die Menüpunkte besitzt, sodass die Redakteure nicht die Möglichkeit hat Aufträge zu erstellen und keine Administrative Möglichkeiten besitzen.

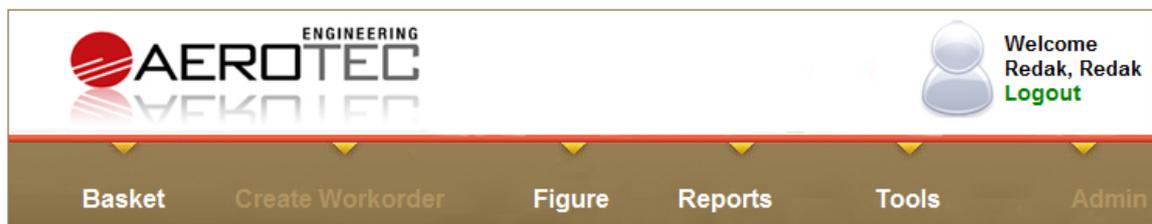


Abbildung 5.6: Menü Redakteure

In der Abbildung 5.7 wird die Menüstruktur der Rolle des Arbeitsvorbereiters gezeigt. Die Arbeitsvorbereiter - Rolle besitzt nur die Create WorkOrder rechte, sodass der Menüpunkte aktiviert und seine Weiterleitung eingerichtet wird. Der Menüpunkte Admin, Report, Figur, Basket werden deaktiviert da die Rolle Arbeitsvorbereiter nicht die Rechte für die Menüpunkte besitzt, sodass der Arbeitsvorbereiter nur die Möglichkeit besitzt Aufträge zu erstellen.



Abbildung 5.7: Menü Arbeitsvorbereiter

## 5.2.2 User Komponente

Die User Settings - Komponente beinhaltet sämtliche Information über die User Settings und die Möglichkeit sie zu pflegen. Die User Settings beinhaltet den zurzeit eingeloggten Benutzer, sodass dem jeweiligen Benutzer die Möglichkeit bietet seine persönlichen Daten zu verwalten. Die User Information des eingeloggten Users wird aus der Datenbank selektiert und in einem User Cluster zusammengefasst.

Folgende Möglichkeiten bietet diese Komponente:

- Ändern des Passwort
- Ändern der Email
- Ändern der Telefonnummer

Durch bestätigen des „Save Settings“ Button werden die Informationen Passwort, in einem MD5 Hash konvertiert, Email und die Telefonnummer anhand des User Cluster aktualisiert und an die Datenbank übertragen. Notwendige Felder, die zum Anlegen erforderlich sind, werden mit einer entsprechenden Validierung dieser Felder den Anwender nötigenfalls auf das Fehlen hingewiesen

Erreicht wird die User Setting über anklicken des Usericon der oben rechts zu sehen ist (siehe Abbildung 5.8)

The screenshot shows the 'User Settings' page for Aerotec Engineering. At the top left is the Aerotec logo with the word 'ENGINEERING' above it. To the right, there is a user profile icon and the text 'Welcome Aerotec, Aerotec' with a 'Logout' link. Below the header is a navigation bar with the following items: 'Basket', 'Create Workorder', 'Figure', 'Reports', 'Tools', and 'Admin'. The main content area is titled 'Aerotec, Aerotec' and contains a form with the following fields: 'Email:' with the value 'Aerotec@aerotec.de', 'Phone:' with the value '12345', 'Password:', and 'Confirm Password:'. A 'Save Setting' button is located below the form.

Abbildung 5.8: User Settings

### 5.2.3 Work Order Komponenten

Ein Work Order ist ein Kapitel der nur einem Benutzer zugewiesen werden kann. Ein Kapitel besteht in wesentlichen aus Aufträgen, Timeschedule, Figuren, BlockingPoint und Chapter Setup. Durch anklicken des Menüpunktes Basket erreicht man die Work Order - Komponente (siehe Abbildung 5.9). Hier gelangt der jeweilige Benutzer auf die Übersicht seiner sämtlichen Work Order bezogene Daten. Die Work Order Daten des eingeloggtem Users werden aus der Datenbank für den jeweiligen Users selektiert und in einen WorkOrder-Cluster gruppiert. Durch auswählen der Reiter MyWorkOrder, MyFigure, MyIllustration, MyAwo, oder MyProofreading wird der WorkOrder-Cluster nach den jeweiligen Reiterpunkt sortiert und

in Listenform für den User angezeigt. Um die jeweiligen Work Orders zu verwalten oder zu löschen reicht ein einfacher Klick auf den jeweiligen Edit / Remove Button.

Die Work Order - Komponente bietet folgende Möglichkeiten an:

### **Anlegen eines Kapitel**

Ein Kapitel wird durch bestätigen des Create Work Order Button angelegt. Notwendige Felder, die zum Anlegen erforderlich sind, werden mit eine entsprechende Validierung dieser Felder den Anwender nötigenfalls auf das Fehlen hingewiesen

Erreicht wird die Funktion Anlegen eines Kapitels über dem Menüpunkt Create Work Order (siehe Abbildung 5.11).

Nach dem bestätigen des Create Work Order Button und der erfolgreichen Validierung wird ein WorkOrder-Cluster erstellt und in der Datenbank gespeichert. Anschließend hat der User noch die Möglichkeit für den aktuellen angelegten WorkOrder Aufträge und Timeschedule anzulegen(siehe Abbildung 5.12), die bei erfolgreicher Erstellung an dem WorkOrder-Cluster hinzugefügt werden und in der Datenbank gespeichert werden.

### **Bearbeiten eines Kapitel**

Bearbeitet wird ein Kapitel, in dem man in die Editeansicht eines Kapitels gelangt. Erreicht wird diese Editeansicht durch den Menüpunkt Basket um an die jeweiligen benutzerspezifischen Kapitel zu gelangen (siehe Abbildung 5.9). Durch anklicken des jeweiligen Edit Button des Kapitels wird für das jeweilige Kapitel eine temporäre WorkOrder-Cluster mit den jeweiligen Daten aus der Datenbank selektiert und der Editansicht übergeben(siehe Abbildung 5.10).

In der Editeansicht gibt es nun die Möglichkeiten für das jeweilige temporären WorkOrder-Cluster Daten hinzuzufügen oder zu verwalten:

- Awo's
- Timeschedule
- Figure
- Blocking Points
- Chapter Setup

Nach der Erstellung oder Verwaltung der einzelnen Daten des temporären WorkOrder-Cluster werden die Daten in der Datenbank hinzugefügt bzw. aktualisiert.

The screenshot displays the Aerotec Engineering web application interface. At the top left is the Aerotec logo with the text 'ENGINEERING AEROTEC'. At the top right, there is a user profile icon and the text 'Welcome Aerotec, Aerotec Logout'. Below the header is a navigation bar with the following menu items: 'Basket', 'Create Workorder', 'Figure', 'Reports', 'Tools', and 'Admin'. The main content area features a 'Choose User' dropdown menu and a 'Show' button. Below this, there are links for 'My WorkOrder', 'My Figure', 'My Illustration', 'My Awo', and 'My Proofreading'. The 'My WorkOrder' section is expanded, showing a table with the following data:

OrderNo	Author	ChapterNo	Projekt	CreateDate	startDate	endDate	
Aerotec-IPC-323212-15.08.2008	Aerotec	323212	IPC	18.08.2008	18.08.2008		<a href="#">Edit</a> <a href="#">remove</a>

Abbildung 5.9: Basket

**ENGINEERING AEROTEC**

Welcome Aerotec, Aerotec  
Logout

Basket Create Workorder Figure Reports Tools Admin

**Chapter**

Company:	Aerotec	Projekt:	IPC
Revisondate:	15.08.2008	User:	Aerotec
Chapter:	323212	RequiredEnd Date:	20.08.2008
Order No:	Aerotec-IPC-323212-15.08.2008		

AWO »

Figure »

Chapter Setup »

Timeschedule »

Blocking Point »

Abbildung 5.10: Edit Work Order

The screenshot displays the 'Create Workorder' interface. At the top left is the 'AEROTEC ENGINEERING' logo. At the top right, a user profile icon is shown with the text 'Welcome Aerotec, Aerotec' and a 'Logout' link. Below the header is a navigation bar with the following items: 'Basket', 'Create Workorder', 'Figure', 'Reports', 'Tools', and 'Admin'. The main content area is titled 'Chapter' and contains the following form fields:

- Company:** A dropdown menu with the text 'Choose Company'.
- Projekt:** A dropdown menu with the text 'Choose Projekt'.
- Revisondate:** A text input field with a calendar icon on the right.
- User:** A dropdown menu with the text 'Choose User'.
- Chapter:** A text input field.
- RequiredEnd Date:** A text input field with a calendar icon on the right.
- Order No.:** A text input field.

A 'Create Order' button is located at the bottom right of the form area.

Abbildung 5.11: Create Workorder Step 1

The screenshot shows the 'Create Workorder Step 2' interface. At the top left is the 'AEROTEC ENGINEERING' logo. At the top right, it says 'Welcome Aerotec, Aerotec' and 'Logout'. Below the logo is a navigation bar with buttons for 'Basket', 'Create Workorder', 'Figure', 'Reports', 'Tools', and 'Admin'. The main content area is titled 'Chapter' and contains a form with the following fields:

- Company: Aerotec (dropdown)
- Revisondate: 10.09.2008 (calendar icon)
- Chapter: 89000 (text input)
- Projekt: IPC (dropdown)
- User: Viktor (dropdown)
- RequiredEnd Date: 18.09.2008 (calendar icon)

Below the form, the 'Order No.' is displayed as 'Aerotec-IPC-89000-10.09.2008'. There is a 'Save Order' button. At the bottom, there are two expandable sections: 'AWO' and 'Timeschedule'.

Abbildung 5.12: Create Workorder Step 2

## 5.2.4 Figur Komponenten

Die Figur Komponente wird über den Menüpunkt „Figur“ erreicht. Durch auswählen einer Figur und bestätigen des „Show“ Buttons wird die Figur Maske durch Erstellung einer temporären Figur-Cluster, die die Daten für die ausgewählte Figur aus der Datenbank selektiert, geladen(siehe Abbildung 5.13). Nun hat der Benutzer die Möglichkeit die Daten der Figur zu verwalten. Die Figur Daten bestehen aus den Komponenten Status, Setup, Illustration, Addition Source Data, Blocking Point und Items. Sie können nun verwaltet oder ggf. gelöscht werden (siehe Abbildung 5.14). Jeder der einzelnen Daten besitzt ihre eigene Komponente die folgende Möglichkeit bietet:

- Erstellen
- Bearbeiten
- Löschen

Nach der Verwaltung der jeweiligen Komponenten werden die Daten anhand der Datenbank aktualisiert.



Abbildung 5.13: Figure Step 1

**ENGINEERING AEROTEC**

Welcome Aerotec, Aerotec Logout

Basket Create Workorder Figure Reports Tools Admin

Figure Management

Figure: Aerotec PC-323212-19.0 8.20.08-Fig-05-C Show

Revisendate: Aug 14, 2008 ready for proofreading:  Proofreader accept:   
Chapter: 323212 Proofreader: Aerotec Proofreader finish:   
Company: Aerotec Auto SAP Status valid:   
Figure valid:   
Figure Blocked Y/N:

save Figure Details

Status  
Setup  
Illustration  
Additional Source Data  
Blocking Point  
Item

Abbildung 5.14: Figure Step 1

### 5.2.5 Report Komponenten

Die Report Komponente beinhaltet alle Reportvorlagen, die mit der grafische Oberfläche iReport erstellt worden sind. Die Reportvorlagen werden per Laufzeit mit dem gewünschtem Inhalt gesetzt und dem Benutzer ermöglicht das Ergebnis des Reports als ein PDF Dokument anzuschauen. Die Report Komponente wird über den Menüpunkt „Reports“ erreicht (siehe Abbildung 5.15).

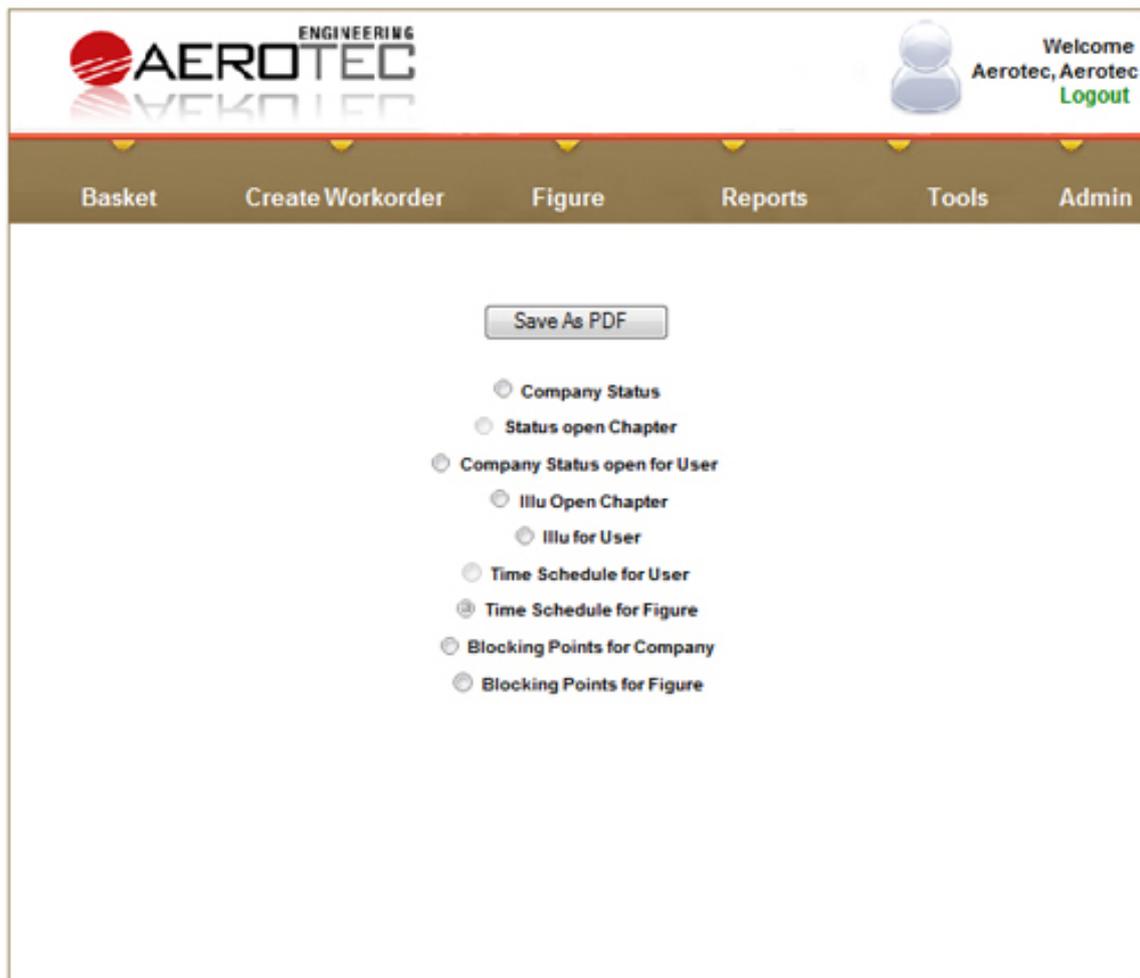


Abbildung 5.15: Reports

## 5.2.6 Admin Komponenten

Die Admin Komponente beinhaltet die sämtliche Information des Administrativen teils. Bei der Admin Komponente werden die Rollen des jeweiligen Benutzers zusätzlich überprüft, da die Rolle Admin und die Rolle SuperUser für unterschiedliche Aufgaben des Administration zuständig sind, besitzen sie auch unterschiedliche Rechte.

Die Rechte werden in vier Untermenüpunkte View User, Create User und User Rolle aufgeteilt. Sobald ein Benutzer die Rechte besitzt, werden die die Untermenüpunkte für den jeweiligen Benutzer aktiviert. Somit hat der Benutzer dann die Möglichkeit Administrative Aufgaben unter den Menüpunkten nach zu kommen.

Zu den einzelnen Admin Untermenüpunkte kommt man über den Menüpunkt Admin.

### **View User**

Bei dieser Komponente handelt es sich um Überblick der einzelne Benutzer die im System registriert sind. Der firmenspezifische Admin hat nur einen Überblick auf seine Benutzer, wobei der SuperUser den Überblick auf sämtliche Benutzer hat. Anhand dieser Rechteverteilung werden nur die jeweiligen Benutzer aus der Datenbank gelesen und als Listenform aufgelistet. Durch anklicken des Edit/Remove Button des zugehörigen Benutzers ist es nun möglich den jeweiligen Benutzer zu löschen oder zu bearbeiten (siehe Abbildung 5.16).

### **Create User**

Unter diesem Menüpunkt ist es für dem Admin und dem SuperUser möglich Benutzer mit Rollen zu verweisen und anzulegen, allerdings ist dem Admin nur möglich firmenspezifische Benutzer anzulegen(siehe Abbildung 5.17).Bevor der Benutzer angelegt wird, wird überprüft ob der Benutzer schon im System existiert. Existiert der Benutzer nicht wird der Benutzer im System hinzugefügt und in der Datenbank abgespeichert. So bald der Benutzer existiert wird der jeweiligen Admin darauf hingewiesen das ein Benutzer mit den Daten schon im System existiert.

### **User Rolle**

Unter diesem Menüpunkt ist es dem SuperUser möglich die Benutzer-Rollen zu verwalten bzw. um Rechte zu erweitern oder zu entziehen (siehe Abbildung 5.18). Durch auswählen der jeweiligen Rolle wird eine temporäre Rolle-Cluster erstellt und mit den Rechten der jeweiligen Rolle aus der Datenbank geladen und in Listenform angezeigt. Durch auswählen bzw. entfernen der einzelnen Rechte für die jeweilige Rolle wird die Temporäre Rolle-Cluster bearbeitet. Abschließend durch bestätigen die Updates Rolle Buttons wird die Rolle-Cluster der Datenbank übergeben.

## 5 Realisierung

ENGINEERING  
**AEROTEC**  
AEROTEC

Welcome  
Aerotec, Aerotec  
Logout

Basket Create Workorder Figure Reports Tools Admin

View User Create User User Role EmailHost

ID	Firstname	Lastname	
12	Sven	Arendt	<a href="#">Edit</a> <a href="#">remove</a>
13	Viktor	Zequiraj	<a href="#">Edit</a> <a href="#">remove</a>
14	Christoph	Albert	<a href="#">Edit</a> <a href="#">remove</a>
15	Sakir	Aylar	<a href="#">Edit</a> <a href="#">remove</a>
16	Gunnar	Heidkamp	<a href="#">Edit</a> <a href="#">remove</a>
17	Gunar	Hofmann	<a href="#">Edit</a> <a href="#">remove</a>
18	Mohsen	Shirazi	<a href="#">Edit</a> <a href="#">remove</a>

Abbildung 5.16: Admin View User

**ENGINEERING AEROTEC**

Welcome Aerotec, Aerotec  
Logout

Basket Create Workorder Figure Reports Tools Admin

View User Create User User Role EmailHost

Username:

Firstname:

Lastname:

Email:

Phone:

Password:

Company  
Choose Company

Role  
Choose Role

Activity

Save User

Abbildung 5.17: Admin Create User

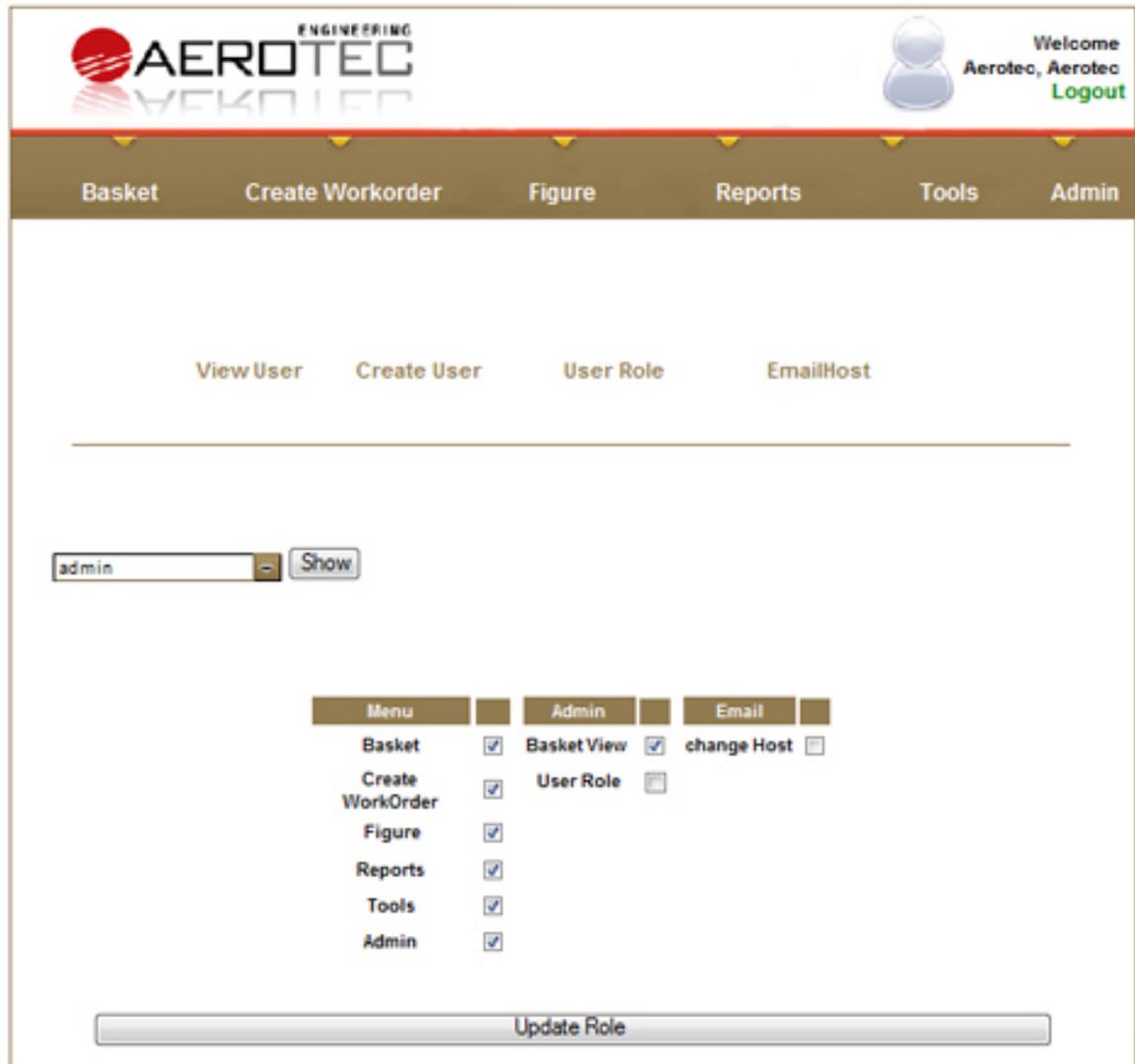


Abbildung 5.18: Admin User Role

# 6 Testkonzept

Um zu gewährleisten, dass die entwickelte Software für einen Produktiveinsatz bereit ist, sind verschiedene Tests notwendig. Diese Tests dienen als Beweis für eine lauffähige Software. Dabei ist das Ziel dieser Tests, eine möglichst hohe Testabdeckung zu erreichen und somit die Qualität der Software zu messen.

Das neue System besteht aus folgenden Komponenten:

- Basket
- Work Order
- Figur
- Report
- Admin

Die Komponenten sollen in den folgenden Teststufen intensiv geprüft werden. Dabei sollen keine Fehler entstehen, die zu einem Absturz der gesamten Anwendung führen. Es wird Testdaten Manuel in der Datenbank gespeichert um die Testfalle durchführen zu können.

## 6.1 Komponententest

Im Komponententest werden die erstellten Softwarebausteine unmittelbar nach der Programmierphase erstmalig einem systematischen Test unterzogen. Wichtigste Aufgabe des Komponententests ist die Sicherstellung, dass das jeweilige Testobjekt die laut seiner Spezifikation geforderte Funktionalität korrekt und vollständig realisiert.

Um Korrektheit und Vollständigkeit der Implementierung zu prüfen, wird die Komponente einer Reihe von Black Box Testfälle unterzogen, d.h. die Funktionen bzw. Methoden der Testobjekte werden ohne die Spezifikation der jeweiligen Implementierung zu kennen, auf bestimmte Ein-/Ausgabe-Kombination überprüft. Bei Entdeckte Fehler wird der jeweilige Fehler behoben und auf ein Neues getestet um Folgefehler zu vermeiden.

Da die Komponenten in dieser Teststufe keine Abhängigkeiten untereinander besitzen, werden alle Datenbank zugriffe wie z.B. lesen und schreib Funktionen der einzelnen Komponenten überprüft.

Als Beispiel wird die Komponente Work Order genannt, die die Funktionen `addWorkOrder()`, `deleteWorkOrder()` und `getAllWorkOrderForUser()` enthält. Die einzelnen Funktionen werden nun anhand eines Beispiel Szenarien getestet.

Die folgende Tabelle zeigt einen Ausschnitt der Work Order Tabelle, wo wir nun unser Beispiel Szenario für die einzelnen Funktionen, für den User mit der UserID „1“ anwenden.

**Tabelle Work Order**

WorkOrderID	UserID
123456	1
312343	1
787878	2
999999	1
888000	2

**addWorkOrder()**

Nun wird anhand der `addWorkOrder` Funktion der Work Order „001100“ der noch nicht in der oben aufgelistete Tabelle existiert, für den User mit der UserID „1“ hinzugefügt.

WorkOrderID	UserID
123456	1
312343	1
787878	2
999999	1
888000	2
<b>001100</b>	<b>1</b>

Die folgende Tabelle zeigt das der neue Datensatz erfolgreich in der Datenbank hinzugefügt wurden ist.

**deleteWorkOrder()**

Anhand der Funktion `deleteWorkOrder()` wird nun Versucht den WorkOrder „999999“ zu löschen.

WorkOrderID	UserID
123456	1
312343	1
787878	2
888000	2
001100	1

Die folgende Tabelle zeigt nun Ergebnisse der Tabellen nach dem erfolgreichem löschen den Datensatz

**getAllWorkOrderForUser()**

Mit der Funktion getAllWorkOrderForUser() wird für den User mit der UserID „1“, versucht nun alle Work Order Datensätze des Users zu erhalten.

WorkOrderID	UserID
123456	1
312343	1
001100	1

Die folgende Tabelle zeigt das dass Ergebnis der Eingabe entspricht.

Um eine vollständige Testabdeckung für die jeweiligen Funktion zu erhalten, werden in zusätzliche Testfälle versucht falsche Datensätze einzugeben, existierende Datensätze zu erzeugen, nicht existierende Datensätze zu löschen und nicht vorhandene User WorkOrder zu zuweisen um die Reaktion der Funktion auf Fehler zu überprüfen. Die Funktionen reagieren alle anhand der Anforderungen und weisen den Benutzer auf die falsche Eingabe der Werte, ohne einen System Absturz zu erzeugen.

## 6.2 Integrationstest

Nachdem die ersten Komponenten getestet wurden erfolgt die Zusammenfügung einzelnen Komponenten zu Teilsystemen. Beim Integrationstest wird davon ausgegangen, dass die einzelnen Komponenten sehr gut getestet wurden und nun die Arbeit auf das Testen von Komponentenschnittstellen und das Zusammenwirken der Komponenten zu konzentrieren, um Schnittstellenfehler aufzudecken.

Die Bausteine werden z. B. in der (zufälligen) Reihenfolge ihrer Fertigstellung integriert. Sobald eine Komponente ihren Komponententest absolviert hat, wird geprüft, ob sie zu einer anderen schon vorhandenen und bereits getesteten Komponente oder zu einem teilintegrierten Teilsystem passt. Wenn ja, werden beide Teile integriert.

Die Teilsysteme werden wie im Vorherigen Abschnitt Komponententest anhand von Black Box Testfälle getestet, um zu gewährleisten dass die integrierten Komponenten fehlerfrei mit einander kommunizieren.

## 6.3 Systemtest

Nach abgeschlossenem Integrationstest folgt vor Auslieferung des Systems der Systemtest . Der Systemtest ist die Testphase, bei der das gesamte System gegen die gesamten Anforderungen (funktionale und nicht funktionale Anforderungen) getestet wird. Der Systemtest betrachtet das System hingegen aus der Perspektive des Kunden und des späteren Anwenders. Durch den Systemtest wird validiert, ob die Anforderungen vollständig und angemessen umgesetzt wurden. Der Systemtest wird auf der Testumgebung der späteren Produktivumgebung möglichst nahe kommt getestet.

Um alle funktionale-/nicht funktionale Anforderungen zu testen, wurde das jetzige System anhand der Nutzungsszenarien in Kapitel 3.6 getestet, damit die Anforderungen aus der Perspektive des späteren Anwenders auch wirklich realisiert wurden sind.

Nach umfangreichen testen der funktionale-/nicht funktionale Anforderungen und Anwendungsfällen, wurden alle Anforderungen in dem Prototyp realisiert.

## 6.4 Abnahmetest

Vor Inbetriebnahme der Software erfolgt nun als abschließender Test noch ein so genannter Abnahmetest. Hierbei stehen die Sicht und das Urteil des Kunden bzw. Anwenders im Vordergrund. Der Abnahmetest ist unter Umständen der einzige Test, den der Kunde nachvollziehen kann oder an dem er direkt beteiligt ist. Der Abnahmetest ist eine spezielle Form des Systemtests, der beim Kunden durchgeführt wird, damit der Kunde überprüfen kann ob er das bestellte Softwaresystem als mangelfrei betrachtet und den Entwicklungsvertrag bzw. die vertraglich geschuldete Leistung als erfüllt ansieht.

Momentan wurde das neue System eingerichtet und dem Kunden übergeben, um zu gewährleisten dass alle Erwartungen aus Sicht des Kunden erfüllt wurden sind. Bevor das neue System endgültig in Betrieb eingesetzt werden kann, wird anhand von Feedback der Benutzer das System kontinuierlich ausgebaut, um zu gewährleisten dass keine Fehler in den Geschäftsprozesse enthalten sind und dass es alle Anforderungen des Benutzers erfüllt.

# 7 Zusammenfassung

## 7.1 Fazit

Ziel dieser Arbeit war es ein neues Konzept für das Altsystem zusammen zu stellen. Darüber hinaus sollten die alten und die neuen Anforderungen in einem Prototyp umgesetzt werden. Bei der Analyse des Altsystems wurden neue Anforderungen für das neu zu entwickelnde System untersucht und erfasst. Die Anforderungen wurden anhand von existierende Lösungen die auf dem Markt sind verglichen, um gegeben falls eine Lösung zu verwenden. Bei dem Vergleich eines existierenden Systems ist keine angemessen Einsetzbarkeit festgestellt worden und deshalb wurde eine Neuentwicklung angestrebt. Allerdings wurde bei dem Vergleich von Report-Tools anhand der Anforderungen ein Tool analysiert und ausgewählt.

Bei der Realisierung der Architektur, wurde eine Architektur entworfen, die bei der Entwicklung und Wartung der einzelnen Schichten relativ einfach auszutauschen sind, ohne die anderen auch mit ändern zu müssen. Durch die Wahl dieser Architektur wird die Wartbarkeit und die Erweiterbarkeit des Systems gesteigert.

In einem anschließenden Test wurden Fehler beseitigt und die gesamte Anwendung wurden nach den neuen Anforderungen optimiert, so dass einem Produktiveinsatz für die Testphase nichts mehr in Wege steht.

Der Prototyp der zurzeit in der Testphase eingeführt wurden ist, läuft stabil und aus Sicht des Benutzers wurden die Anforderungen der Benutzerfreundlichkeit und der Bedienbarkeit gut umgesetzt.

## 7.2 Ausblick

Das neue System, das zurzeit in der Testphase liegt, wird durch Feedback der Benutzer kontinuierlich ausgebaut.

Momentan wurden Testdaten für das neue System angelegt um dem Benutzer die Möglichkeit zu bieten Anwendungsfälle durch zu arbeiten oder es wurden Testdaten vom Benutzer aus angelegt. Allerdings sollte nach Beendigung der Testphase eine Möglichkeit in Aussicht gestellt werden um die Daten des Altsystems verlustfrei zu übernehmen, sodass dass neue System endgültig in Betrieb genommen werden kann.

Bevor der entwickelte Prototyp des Handbuchs IPC/IPL endgültig für den Betrieb eingesetzt werden kann, muss zunächst sichergestellt werden, dass alle gestellten Anforderungen des Benutzers erfüllt wurden sind. Bis zur Einführung des Systems muss sichergestellt sein, dass es mindestens die Funktionalität bietet, um eine problemlose Arbeit damit zu ermöglichen.

Darüber hinaus sind noch Handbuch übergreifende Weiterentwicklungen eingeplant, sodass dass System nicht nur für das Handbuch IPC/IPL eingesetzt werden soll. So könnte das System um weitere Handbücher erweitert werden.

# Glossar

**CMS** Content Management System

**EJB** Enterprise Java Beans

**Java SE** Java Standard Edition

**Java VM** Java Virtual Machine

**Java EE** Java Enterprise Edition

**RDMS** Relationales Data Management System

**HTML** Hypertext Markup Language

**JDBC** Java Database Connectivity

**JSP** JavaServer Pages

**JSF** Java Server Faces

**MDB** Message Driven Bean

**POJO** Plain old Java Object

# Literaturverzeichnis

- [ADELCHI 2004 ] ADELCHI 2004, Missio: *JasperReports.*, Diplomarbeit. – URL <http://www.gruntz.ch/courses/sem/ss04/jasperreports.pdf>
- [AIIM International 2003] AIIM INTERNATIONAL: *Enterprise Content Management (ECM).* 2003. – URL <http://www.aiim.org.uk/index1.asp>
- [Andreas Stöckl 2006] ANDREAS STÖCKL, Frank Bongers Frank B.: *Einstieg in TYPO3 4.0.* Galileo Pres, 2006. – ISBN 3-8984-2836-2,
- [Anja Ebersbach ] ANJA EBERSBACH, Radovan K.: *Das Handbuch für Einsteiger.* Bd. Joomla!. Galileo Computing. – ISBN 3-89842-632-7
- [Assisi 2005] ASSISI, Ramin: *J2EE mit Eclipse 3 und JBoss.* München : Carl Hanser Verlag München Wien, 2005. – ISBN 3-44-22739-3
- [Balzert 2000] BALZERT, Helmut: *Lehrbuch der Software-Technik.* Bd. 2. Software-Entwicklung. Heidelberg ; Berlin : Spektrum, Akad. Verlag, 2000. – ISBN 3-8274-0480-0
- [Benchpark ] BENCHPARK: *HighText Verlag Graf und Treplin OHG.* – URL ([http://www.benchpark.com/e-commerce.htm?offset=1&sc=Benchmark.CNT\\_ASSESSMENTS&so=1&sub=&val=&abc=](http://www.benchpark.com/e-commerce.htm?offset=1&sc=Benchmark.CNT_ASSESSMENTS&so=1&sub=&val=&abc=)). – Abruf: 22.08.2008
- [Cms Matrix ] CMS MATRIX: *Plain Black Corporation.* – URL <http://cmsmatrix.org/matrix>. – Abruf: 22.08.2008
- [Dierk Harbeck ] DIERK HARBECK: *EJB 3 professional.* – URL <http://www.ejb3buch.de/cms/EJBGlossar/M>. – Abruf: 24.09.2008
- [Gorus ] GORUS, Oliver: *Web Content Management.* Bonn : Galileo Business. – ISBN 3-89842-157-0
- [JasperSoft 2006] JASPERSOFT: *JasperReports Datasheet.* 2006. – URL <http://www.jaspersoft.com/downloads/Datasheet/jasperreports-0206.pdf>
- [Mercer 2006] MERCER, David: *Drupal der schnelle Einstieg.* Bd. Drupal. Paperback, 2006. – ISBN 978-3-8266-1695-2

- [Märsch 2006] MÄRSCH, Friedhelm: *EJBs und J2EE*. W3L, 2006. – ISBN 3-937137-10-6
- [Red Hat Inc. ] RED HAT INC.: *Jboss Application Server*. – URL <http://www.jboss.org/>
- [Sun Microsystems Inc. 2002] SUN MICROSYSTEMS INC.: *Core J2EE Patterns*. 2002. – URL <http://java.sun.com/blueprints/corej2eepatterns/Patterns/>. – Abruf: 26.05.2008
- [Sun Microsystems Inc. 2008] SUN MICROSYSTEMS INC.: *Java 2 Platform, Enterprise Edition (J2EE) Overview*. 2008. – URL <http://java.sun.com/j2ee/overview.html>. – Abruf: 26.05.2008
- [Wikipedia 2008] WIKIPEDIA: *Java EE*. 2008. – URL <http://de.wikipedia.org/wiki/JavaEE>. – Abruf: 07.06.2008

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 8. Januar 2009

Ort, Datum

Unterschrift