



Technisch-Naturwissenschaftliche  
Fakultät

# Expertise Management System

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Masterstudium

NETZWERKE UND SICHERHEIT

Eingereicht von:

Thomas Schneider, Bsc.

Angefertigt am:

Institut für Informationsverarbeitung und Mikroprozessortechnik

Beurteilung:

o. Univ.-Prof. Dr. Jörg R. Mühlbacher

Mitwirkung:

Assoz.Prof. Priv.-Doz. Mag. iur. Dipl.Ing. Dr. Michael Sonntag

Linz, März 2013

## *Zusammenfassung*

In Zusammenarbeit wurde mit der Firma AXAVIA Software GmbH und deren Produkt AXAVIASeries ein Gutachtenverwaltungssystem implementiert. Dieses System stellt alle organisatorischen und funktionalen Merkmale bereit, die notwendig sind, um Gutachten zu erstellen, zu warten und abschließend in gedruckter Form zur Verfügung zu stellen.

Es wurde der Arbeitsablauf eines Gutachters untersucht und mit Hilfe dieser Information, wird ein solcher Prozess über das System abgewickelt. Weitere Aspekte sind die Aufzeichnung von Zeiten, die für ein Gutachten aufgewendet wurden, die Möglichkeit aus Vorlagen Dokumente, Checklisten zu generieren, sowie alle notwendigen Dateien zu klassifizieren und leicht auffindbar in einem geschlossenen Gutachtenprojekt wiederzufinden. Ein Schlagwortverzeichnis inklusive Begriffserklärung kann automatisiert für ein Gutachten eingebunden werden und die Rechnungslegung wird auch unterstützt. Nach Abschluss des Gutachtens, kann dieses Projekt mit all seinen Informationen und Dateien verschlüsselt, archiviert und als Datei aus dem System exportiert werden.

Das entwickelte Programm wird als Client – Server Konzept umgesetzt und basiert auf einem Dokumentenmanagementsystem, um alle Daten und Dateien in einem zusammengehörigen System zu halten. Als Arbeitsergebnis wird dieses System über die AXAVIA Homepage frei verfügbar gestellt und mit Installations- & Bedienungsanleitung versehen.

## *Abstract*

In cooperation with AXAVIA Software GmbH and it's corresponding product AXAVIASeries an expertise management system was developed. This system should consist of all organizational and functional features which are needed to create, maintain and place an expertise to it's disposal.

The program will be developed as a client – server concept and implemented based on a document managemant system, so that all data and files are kept within one system. It examines the work of an expert and this information will be transfered into the workflow oft the program.

Other aspects are recording time which are needed for an expertise, the possibility to create documents and checklists from templates. Also the ability to classify files and make them searchable within the expertise project. A glossary directory inclusive definition can be added automatically to an expertise and the financial reporting is also supported. After closing such an expertise project it should be archived and exported as an encrypted file.

As work result this whole system including installation and operating manuals will be available from the AXAVIA homepage.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>3</b>
<b>1 Aufgabenstellung .....</b>	<b>5</b>
1.1 Server-basiertes System .....	5
1.2 Dokumentenmanagementsystem.....	5
1.3 Vorlagen.....	6
1.4 Automatisches Glossar .....	7
1.5 Zeitaufzeichnung .....	8
1.6 Rechnungslegung .....	9
1.7 Archivieren .....	9
1.8 Protokollierung der Vorgänge.....	10
<b>2 Vergleich existierender Systeme .....</b>	<b>11</b>
2.1 GutachterForm 6.0 Plus .....	11
2.2 Gutachten Manager 6.0 .....	12
2.3 SV-OfficeManager 2.0.....	14
2.4 GutachterSoft 2.1.8 .....	16
2.5 Resümee.....	18
<b>3 Prozessablauf für die Gutachenerstellung .....</b>	<b>19</b>
3.1 CRM/DMS .....	20
3.2 Projekt.....	21
3.3 Time & Invoice .....	22
3.4 Config.....	23
<b>4 Entwicklungsumgebung.....</b>	<b>24</b>
4.1 Datenbank.....	25
4.2 Entwicklungssprache .....	25
<b>5 Frontend AXAVIASeries.....</b>	<b>25</b>
5.1 Allgemein .....	26
5.2 Überblick .....	29
5.3 Modellebene.....	34
5.4 Datenebene.....	40
5.5 Berechtigungsverwaltung .....	46
5.6 Dokumentenmanagement.....	64
<b>6 Umsetzung und Arbeitsweise .....</b>	<b>67</b>
6.1 Konfiguration .....	67

6.2	Parteien erzeugen .....	75
6.3	Gutachtenprojekt erzeugen .....	78
6.4	Schlagwortverzeichnis generieren .....	88
6.5	Zeitaufzeichnung .....	92
6.6	Rechnung erstellen .....	94
6.7	Projekt archivieren .....	96
6.8	Webclient .....	98
<b>7</b>	<b>Implementierungsdetails .....</b>	<b>99</b>
7.1	Objekte und Attribute .....	99
7.2	Ansichten & Formulare .....	101
7.3	Ereignisimplementierung .....	104
7.4	Softwareerweiterung .....	105
<b>8</b>	<b>Verbesserungen .....</b>	<b>107</b>
8.1	Dokumentenmanagementsystem .....	107
8.2	Rechnungslegung .....	107
8.3	Archivieren .....	108
8.4	Archivbrowser .....	108
<b>9</b>	<b>Resümee .....</b>	<b>108</b>
<b>10</b>	<b>Literaturverzeichnis .....</b>	<b>110</b>
<b>11</b>	<b>Abbildungsverzeichnis .....</b>	<b>111</b>
<b>12</b>	<b>Anhänge .....</b>	<b>117</b>
12.1	Systemanforderung .....	117
12.2	Installationsanleitung .....	117
12.3	Bedienungsanleitung .....	126
12.4	Lebenslauf .....	155
12.5	Eidesstattliche Erklärung .....	157

# 1 Aufgabenstellung

Das Thema dieser Diplomarbeit befasst sich mit dem Schreiben von Gutachten unter Berücksichtigung der Abwicklung mit einem Computersystem.

Gutachten können sowohl von Privatpersonen als auch von Behörden in Auftrag gegeben werden. Dabei wird ein Sachverständiger bestellt, welcher zu einer Sachfrage, mittels seiner erworbenen Lebenserfahrung bzw. seines Erfahrungsschatzes als Experte, ein Gutachten abgibt.

Zielstellung dieser Arbeit ist es, vorhandene Produkte zu vergleichen und ein eigenständiges System basierend mit einer vorhandenen Software zu entwickeln und erweitern. Dabei sind nachfolgende Punkte in die Software einzuarbeiten.

## 1.1 Server-basiertes System

Es gibt eine zentrale Anlaufstelle für Daten, diese Daten können von ein oder mehreren Benutzern bearbeitet werden. Wobei je nach Art des Klienten mehr oder weniger Funktionalität zu Verfügung steht, siehe Abbildung 1.

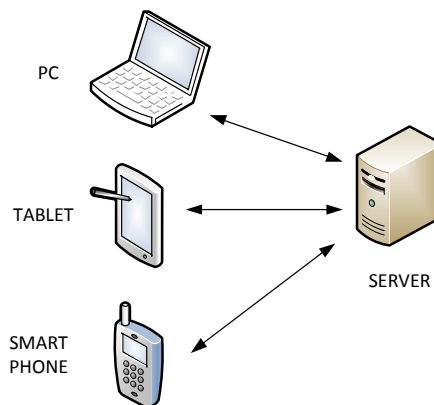


Abbildung 1: Server-basiertes System

Der Grund dafür liegt darin, dass ein Gutachter auch Besichtigungstermine wahrnehmen muss. Über ein Smartphone oder Tablet-PC ist es möglich Bilder vor Ort aufzunehmen und diese in das System zu übertragen. Am Desktop-PC können diese übertragenen Informationen weiter verarbeitet werden. Aber auch der andere Weg ist möglich, dass zum Beispiel Adressinformationen auf einem Smartphone abgerufen werden können, um diese zur Navigation zu nutzen.

Da es einen Server gibt, auf den alle Klienten zugreifen und alle Daten und Dateien abgespeichert sind, muss sichergestellt werden, dass dieser ausfallsicher eingebunden bzw. dass eine Sicherheitskopie vorhanden ist auf die im Falle eines Systemabsturzes zugegriffen werden kann. Auch die Art des Zugriffes muss gesichert sein, damit der Datenverkehr nicht entschlüsselt werden kann.

## 1.2 Dokumentenmanagementsystem

Dokumentenmanagementsysteme (=DMS) dienen zur Verwaltung und Klassifizierung jeglicher Art von Dokumenten. Dokumente können in freidefinierbaren Strukturen

abgelegt werden ähnlich die einer Ordnerstruktur in einem Dateisystem, wobei in einem DMS Dokumente zu mehreren Strukturen verknüpft sein können, siehe Abbildung 2.

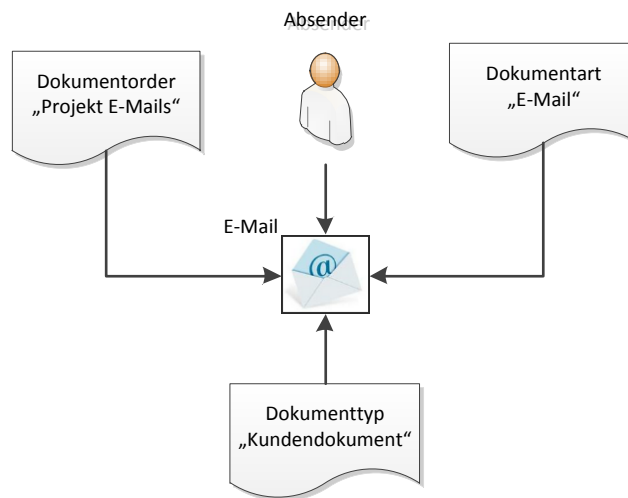


Abbildung 2: Dokumentenmanagementsystem

Der Mehrwert eines solchen Systems liegt unter anderem beim Auffinden von Dokumenten. Metadaten können aus Dateien extrahiert, aber auch neue hinzugefügt werden. Für Dokumente ist es oft wichtig zu wissen wie es sich entwickelt, also wie die einzelnen Revisionen aussehen und welche Veränderungen vorgenommen wurden.

Die Integration selbst soll so einfach als möglich ausfallen. Einfache Drag & Drop Aktionen sollen ausreichen, um Dateien in das System einzuchecken, oder aber auch wieder zu entfernen. Das Bearbeiten der einzelnen Dateien soll über einen Doppelklick möglich sein, wobei definiert werden kann, welches Programm gestartet wird. Nach Beendigung der Bearbeitung einer Datei soll das Speichern dieser ausreichen, um die aktuelle Version dieser im System zu haben.

Damit mit externen Programmen Dokumente oder im Allgemeinen Dateien bearbeitet werden können, müssen diese zur Bearbeitung auf das Dateisystem geschrieben werden. Dabei muss der Bearbeiter vom Betriebssystem Rechte besitzen, um auf dieses Verzeichnis zuzugreifen, aber auch die Datei selber zu bearbeiten.

Probleme können entstehen, wenn Benutzer direkt über das Betriebssystem auf diese Dateien zugreifen, oder diese sogar verändern.

### 1.3 Vorlagen

Es soll möglich sein, Vorlagen für Rechnungen, Gutachten, Checklisten, etc. im System zu hinterlegen und diese bei Bedarf anzuwenden.

Wobei Informationen, die in der Datenbank hinterlegt sind, dafür verwendet werden, diese als Inhalt in z.B. Textdokumenten einzuarbeiten. Bestehende Dokumente können bei Änderungen des Datenbankinhaltes einfach aktualisiert werden.

Ein Beispiel für eine Word Vorlage könnte die Briefkopfinformation (siehe Abbildung 3) von Dokumenten sein.

- Anrede / Titel
- Vor- & Nachname
- Adressdaten
- E-Mail
- Telefon / Fax

## Kontaktinformation

<b>Firmenname</b>	: Testfirma
<b>Adresse</b>	: Teststraße 10 A-4020 Linz
<b>Telefon</b>	: 12345
<b>Fax</b>	: 54321
<b>Ansprechpartner</b>	: Herr Dipl.Ing. Max Muster
<b>Email Adresse</b>	: tom.schuster@testfirma.com
<b>Tel. Durchwahl</b>	: 12345-5
<b>Fax Nummer</b>	: 54321

Abbildung 3: Beispiel Briefkopfinformation

Dabei werden Datenbankattribute verwendet und diese über Textfelder im Dokument ausgefüllt.

Aber auch Objektvorlagen, wie beispielsweise Rechnungen mit den einzelnen Rechnungspositionen, können verwendet und angepasst werden. Dafür soll es einen eigenen Vorlagebereich geben, in dem diese Dokumente und Objekte angepasst werden können.

Damit nicht von jedem diese Vorlagen verändert werden können, soll es zwei Rollen im System geben. Eine administrative Rolle, die alle Einstellungen und Vorlagen abändern und eine Anwenderrolle, die diese dann benutzen darf. Damit wird eine klare Trennung zwischen Konfiguration und Anwendung hervorgerufen.

## 1.4 Automatisches Glossar

Nachdem ein Gutachten geschrieben wurde, kann automatisiert der Inhalt dieses untersucht werden. Wörter, die in einem Gutachten oft vorkommen werden extrahiert und global in der Datenbank abgelegt. Nachdem diese dann definiert wurden, können die jeweiligen Begriffe, mit vorheriger Auswahl, als Anhang in einem Gutachtendokument angegeben werden.

Ein Glossar ist wie ein Lexikon aufgebaut, wobei nur Begriffe und fremdartigen Wörter eines Werkes, in diesem Fall eines Gutachtens, zusammengefasst und erklärt werden. Da in Gutachten immer wieder gleiche Ausdrücke vorkommen, macht es Sinn diese global abzuspeichern. Diese Einträge werden strukturiert und nach Auswahl im System hinterlegt, siehe Abbildung 4.

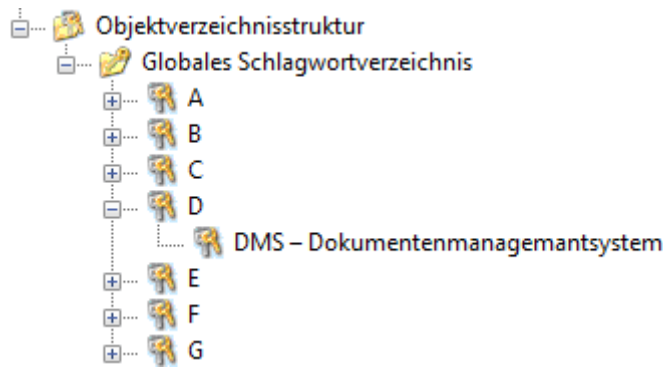


Abbildung 4: Beispiel Schlagwortverzeichnis

Für weitere Gutachten stehen diese Fachbegriffen dann bereits zur Verfügung und können nach Auswahl automatisiert angehängt werden.

Wie soll ein Algorithmus erkennen, was ein Fachbegriff ist und was nicht? Deshalb wird es nötig sein ein Art Black- & Whitelist von Einträgen zu erstellen. Damit können Wörter wie „der, die, das“ vorgefiltert werden, damit diese nicht als Auswahl erscheinen. Ein weiteres Problem könnte die Doppeldeutigkeit von Begriffen werden.

Es wird auf jeden Fall nötig werden, ein globales Glossarverzeichnis zu führen, in dem Begriffen definiert werden können. Diese globalen Einträge werden dann für die jeweilige Expertise übernommen, können aber dann noch einmal falls nötig, speziell für das eine Gutachten, angepasst werden.

## 1.5 Zeitaufzeichnung

Eine andere Aufgabenstellung ist, dass zu Gutachten Zeitaufzeichnungseinträge gemacht werden können, siehe Abbildung 5. Basierend auf diesen Einträgen kann eine Rechnung erstellt werden.

Zeit von/bis	Zeitkonto	Kostenträger	Bemerkung
08:00 – 09:00	Reisekosten	Gutachtenprojekt X	Anreise
09:00 – 10:00	Mühewaltung	Gutachtenprojekt X	Ortsbesichtigung
10:00 – 12:00	Mühewaltung	Gutachtenprojekt X	Sicherstellung d. Beweise
12:30 – 14:00	Reisekosten	Gutachtenprojekt Y	Anreise
14:00 – 16:00	Verhandlungsteilnahme	Gutachtenprojekt Y	Gerichtstermin

Abbildung 5: Beispiel Zeitaufzeichnung

Damit für die spätere Rechnungslegung unterschieden werden kann, welche Art von Gebühr in Rechnung gestellt werden darf, wird es auch unterschiedliche Zeitkonten geben. Diese sind im nächsten Kapitel noch einmal aufgelistet.

Als Kostenträger für den jeweiligen Eintrag wird ein Gutachtenprojekt gewählt an dem man gerade arbeitet. Außerdem notwendig ist es eine Art Bemerkungsfeld für jeden Eintrag zu haben um zu spezifizieren was man tat.

Diese Einträge sollen über einen Kalender eingetragen werden können, wobei zu beachten ist das es mehrere Einträge pro Tag geben kann welche auch zu unterschiedlichen Gutachten gehören. Ersichtlich sollte auch sein, wie viel Zeit und welche Kosten ein Gutachten verursacht.



Zusätzlich zu Zeitaufzeichnungseinträgen wird es auch Auslagenersatzeinträge geben, diese dienen dazu, um vorfinanzierte Geldbeträge, welche zur Lösung des Gutachtens notwendig waren, in Rechnung zu stellen. Dazu gehören etwaige Gebrauchsgegenstände, Hilfsmittel oder aber auch Personen welche man im Zuge einer Expertise benötigte.

## **1.6 Rechnungslegung**

Die Rechnungslegung, basiert, wie bereits im vorherigen Kapitel erwähnt, auf den Zeitaufzeichnungen welche im System hinterlegt werden und den jeweiligen Gutachten zu dem diese zugeordnet sind.

Dieser Rechnungslegung liegt das österreichische Gebührenanspruchsgesetz zugrunde [1], folgende Gebühren sind dabei zu berücksichtigen:

- Reisekosten
- Kostenersatz (Hilfskräfte, Kosten durch Tätigkeiten)
- Entschädigung für Zeitversäumnis
- Mühewaltung
- Teilnahme an Verhandlungen
- Aktenstudium

Rechnungen werden mit dem Kunden verknüpft, für den ein Gutachten erbracht wurde. Eine Rechnung besteht aus ein oder mehreren Rechnungspositionen. Jeder einzelnen Position kann über Vorlagen oder frei bearbeitbar hinterlegt werden.

- Nettopreis
- Umsatzsteuer
- Kurztext
- Langtext

Die Rechnung selbst summiert dann die einzelnen Positionen auf und es kann ein Rechnungsbericht erstellt werden. Alle notwendigen Informationen werden dabei über Datenbankattribute in diesen Bericht übergeben.

Mehrwertsteuersätze können zu einzelnen Positionen auch angegeben werden und ist die Einspruchsfrist einer Rechnung verjährt soll diese archiviert werden können.

## **1.7 Archivieren**

Nachdem ein Gutachten abgeschlossen wurde, sollen alle Informationen die diesem zugrunde liegen, archiviert werden, siehe Abbildung 6.

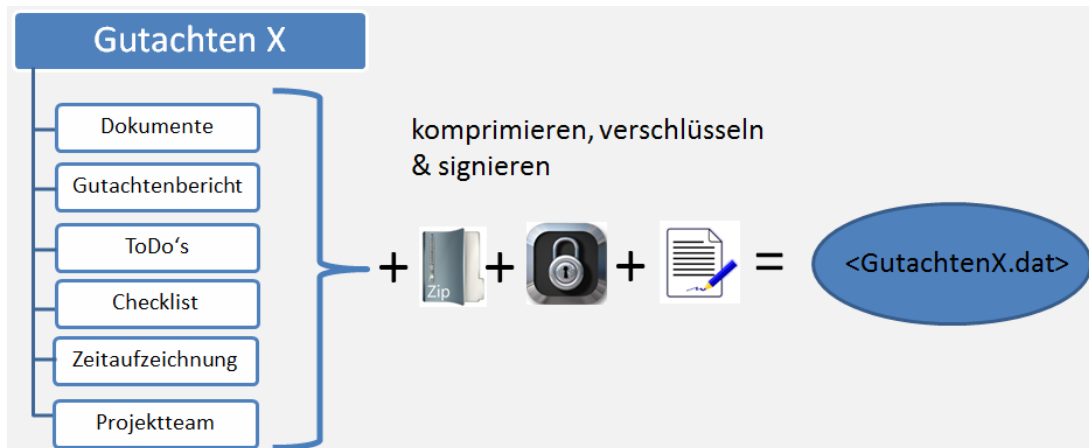


Abbildung 6: Beispiel Archivierung

Alle Datenbankobjekte und Dateien (Dokumente, Bilder, Protokolle, etc.) sollen komprimiert und verschlüsselt in eine Datei exportiert werden können. Ein späteres Importieren soll auch möglich sein.

Archivieren ist deshalb notwendig, damit die Größe der Datenbank in einem angemessenen Rahmen bleibt, da ja nicht nur Daten in die Datenbank eingetragen werden, sondern diese auch als Dokumentenmanagementsystem fungiert.

Für das Komprimieren der Daten kommt ein gängiges Verfahren zu tragen (ZIP-Format), das als Container dient, um alle Daten und Dateien zusammenzufassen. Beim Verschlüssen dieses Containers wird ein Algorithmus gewählt, der sowohl schnell als auch sicher ist, nämlich das symmetrische Blockchiffren-Verfahren AES 256. Für das anschließende Signieren dieser Datei, kann ein Hash Wert über den Inhalt der Datei errechnet (SHA-256) und anschließend im System hinterlegt werden.

Das Verschlüsselungspasswort wird nicht in Klartext im System hinterlegt, sondern nur dessen Hash Wert, der beim anschließenden Importieren dem eingegebenen Kennwort gegenübergestellt wird. Erst nach Richtigkeit des Passwortes und dem Integritätstest der Datei wird das Importieren möglich.

## 1.8 Protokollierung der Vorgänge

Außerdem sollen alle Vorgänge im System protokolliert werden und zwar von wem und wann Änderungen vorgenommen wurden. Jegliche Korrespondenz, welcher Art auch immer, sollen im System abgelegt werden können, seien es Besprechungsprotokolle oder E-Mails.

Es soll leicht ersichtlich und nachvollziehbar sein, wer und wann Änderungen an Daten vorgenommen hat.

## 2 Vergleich existierender Systeme

Um Vergleiche anstellen zu können werden verschiedene am Markt befindliche Systeme untersucht und mittels einer Matrix anschaulich gegenübergestellt. Die horizontale Achse der Matrix beschreibt die Merkmale dieser Systeme laut Aufgabenstellung und die vertikale Achse die Software selbst. Im letzten Unterkapitel findet man die Auswertung.

Außerdem wird aus den untersuchten Systemen ein Konzept entwickelt, wie ein Gutachten aufzubauen ist und wie der Prozess zur Erstellung einer Expertise aussehen könnte.

### 2.1 GutachterForm 6.0 Plus

Um die Software GutachterForm [2] zu erhalten, kann diese über die Homepage bestellt werden, leider ist keine direkte Verknüpfung zum Herunterladen der Software vorhanden und man muss über E-Mail in Kontakt treten.

#### 2.1.1 Systemanforderung

Betriebssystem:	Microsoft Windows XP / Vista / 7
Software:	Microsoft Office 2003 / 2007 / 2010
Speicherplatz:	50MB

#### 2.1.2 Installation

Die Installation erfolgt über eine Setupdatei und außer Installationsverzeichnis sind keine Einstellungen vorzunehmen.

Anzumerken ist, dass dies das einzige Programm ist, welches nicht über einen eigenen Client verfügt sondern sich in das Microsoft Office Produkt integriert. Es wird ein eigenes Untermenü in Microsoft Word eingetragen, „Add-Ins“ heißt es unter Microsoft Word 2010.

Es gibt sowohl Menübefehle als auch eine benutzerdefinierte Symbolleiste, in der Schnellzugriffe zu den Funktionen aus dem Menü definiert werden.

#### 2.1.3 Merkmale

Wie bereits erwähnt, integriert sich diese Software in das Microsoft Word, siehe Abbildung 7. Über Menübefehle können eine Vielzahl von vordefinierten Formularen erzeugt werden.

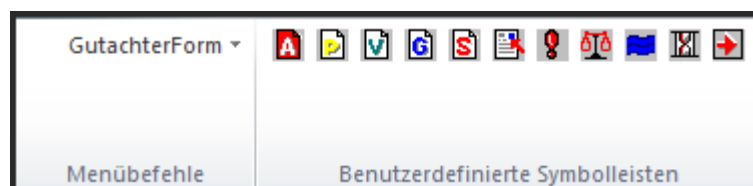


Abbildung 7: Softwarevergleich GutachterForm 6.0 Plus

Als Hauptgutachten zählen:

- Privatgutachten
- Versicherungsgutachten
- Gerichtsgutachten
- Sonstige Formulare

Wobei nach Auswahl eines dieser Gutachten wiederum ein Menü erscheint um eine Vorlage auszuwählen. Von Auftragsbestätigungen, Einladungsschreiben, Ortsterminmitteilungen bis hin zur Kostennote und Zahlungserinnerung. Insgesamt gibt es mehr als 60 Berichtsvorlagen, welche über mehr oder weniger Textfelder verfügen. Über einen eigenen Menüpunkt werden Stammdaten eingegeben, wie die Namen und Adressen der beteiligten Parteien oder Bankverbindungen usw. Genau diese Daten werden dann in die vorher erwähnten Textfelder verknüpft.

Das Programm nutzt die Skriptsprache Visual Basic, welche in den Microsoft Office Produkten verfügbar ist, um Makros und Programmcode auszuführen. Die Daten zu den jeweiligen Gutachten werden als Text in Dateien abgelegt und über das Gutachtenkennzeichen referenziert.

#### **2.1.4 Fazit**

Dem Programm ist ein 56 seitiges Handbuch beigelegt, das man unbedingt durchlesen muss, da es nicht intuitiv bedienbar ist und man nicht immer genau erkennen kann welche Textfelder es in einer Vorlage gibt. Dokumente die aus den Vorlagen erzeugt werden, können dann abgespeichert werden, ein Aktualisieren eines bestehenden Berichts ist nicht mehr möglich.

Positiv ist zu bewerten, dass es sehr viele Vorlagen gibt aus denen man auswählen kann. Leider gibt es keine Möglichkeit Zeiten aufzuzeichnen, die man für ein Gutachten aufgewendet hat. Da es kein eigenständiges Programm ist, vermisst man auch alle anderen Punkte aus den Aufgabenstellungen.

## **2.2 Gutachten Manager 6.0**

Eine kostenlose Testversion des Gutachten-Managers [3] kann direkt über einen Downloadlink auf der Homepage des Produktes bezogen werden.

### **2.2.1 Systemanforderung**

Betriebssystem:	Microsoft Windows 2000 / XP / Vista / 7
Software:	Microsoft Office 2000 bis 2010 aktuelle .NET
Speicherplatz:	50MB

### **2.2.2 Installation**

Nach dem herunterladen der ca. 17MB großen Installationsdatei wird diese direkt ausgeführt. Durch die Installation führt ein einfaches Menü, welches außer dem akzeptieren des Lizenzvertrages nur den Installationsordner ändern lässt.

Nach der Installation findet man ein neues Desktopsymbol auf dem Bildschirm, welches durch Doppelklicken das eigenständige Programm ausführt.

### 2.2.3 Merkmale

Das Programm, siehe Abbildung 8, arbeitet im Hintergrund mit einer Microsoft Access Datenbank und einem vordefinierten Bereich auf der Festplatte, in dem jedes Gutachten einen Ordner anlegt und alle Dateien dort abgelegt werden. Gesteuert werden die Anfragen an diese Datenbank über das .NET Framework.

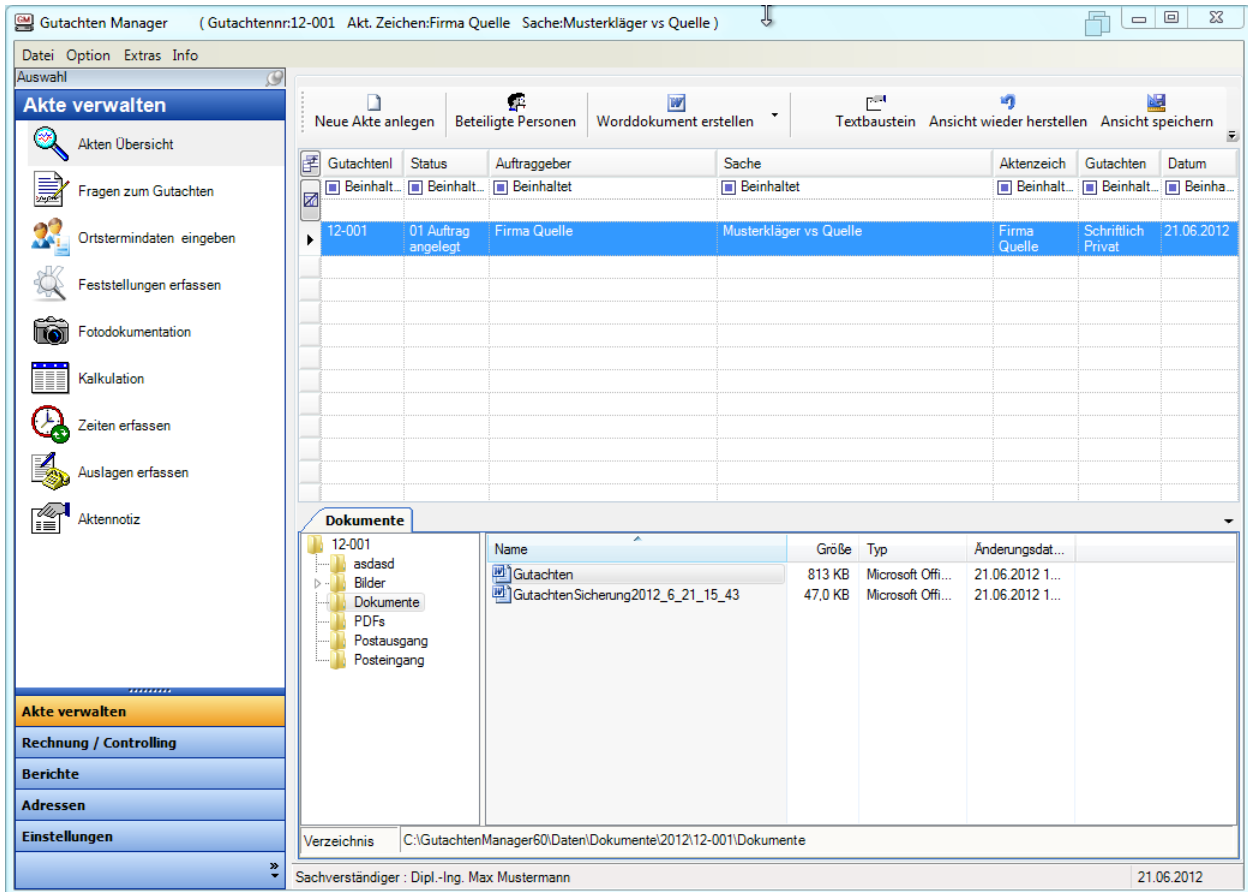


Abbildung 8: Softwarevergleich Gutachten Manager 6.0

Beim ersten Start können die eigenen Daten, die Mitarbeiter, die Gebühren, Auslagenparameter definiert und vorhandene Vorlagen, welche im Word Format vorliegen, verändert werden. Dann ist man bereit einen Akt, so wird ein Gutachten in diesem Programm bezeichnet, zu erzeugen.

Es gibt drei Arten von Gutachten welche zur Auswahl stehen:

- Gerichtsgutachten
- Privatgutachten
- Versicherungsgutachten

Nach Angabe der Art des Gutachtens werden die beteiligten Personen abgegrenzt, die dafür nötig sind. Diese können entweder neu angelegt, oder über bereits eingetragene Stammdaten bezogen werden.

Des weiteren können Angebot- & Auftragsbestätigungen, Fragen & Antworten, Ortsbesichtigungen, Fotodokumentation, Feststellungen, Zeitaufzeichnungen, Auslagen und Aktennotizen je Akt angelegt und verarbeitet werden.

Das Gutachtendokument selbst integriert dann Teile dieser Informationen in ein einheitliches Gesamtdokument, wobei eine Strukturierung vorgegeben wird. Diese kann aber dann manuell erweitert werden. Ein Aktualisieren des bestehenden Berichtes ist nicht möglich, es wird immer eine Sicherungskopie des vorhandenen angelegt. Texte welche man von Hand eingegeben hat, müssen wieder beispielweise über Copy & Paste eingefügt werden.

Auch die Rechnungslegung kann über die Software erfolgen, wobei diese die vorher definierten Zeiten & Auslagen berücksichtigt.

Eine Kalenderfunktion ist auch enthalten, in der Termine & Erinnerungen eingetragen werden können.

Eine Datensicherungsfunktion ist auch enthalten, jedoch kann nur immer das Gesamtsystem, also inklusive aller Gutachten und allen Dateien gesichert werden. Es existiert dann eine 1:1 Kopie. Einstellungen zu länderspezifischen Mehrwertsteuersätzen konnte ich nicht finden.

#### **2.2.4 Fazit**

Ein kompaktes System, welches umfangreiche Funktionen bietet. Das einzige Manko welches festzustellen war, ist das mehrere Personen nicht gleichzeitig damit arbeiten können und Dateien, welche über das System erzeugt werden, auch über den normalen Dateimanager zugänglich sind.

### **2.3 SV-OfficeManager 2.0**

Diese Software [4] wird aus Zeitmangel des Entwicklers nicht mehr aktiv entwickelt, es kann sich aber jeder die letzte Version dieses Produktes über die Homepage herunterladen und lizenzieren.

#### **2.3.1 Systemanforderungen**

Betriebssystem:	Microsoft Windows 2000 / XP [Vista] / 7
Software:	Microsoft Office 2003 [2007, 2010] min. .NET 1.1 & 2.0
Speicherplatz:	20MB

#### **2.3.2 Installation**

Die 4,5MB große Installationsdatei wird gestartet und nach der Wahl des Zielordners wird sofort die Software gestartet. Ein Desktopsymbol ist auch hier wieder vorhanden.

#### **2.3.3 Merkmale**

Wie beim „Gutachten Manager“ arbeitet auch dieses Programm wieder mit einer Microsoft Access Datenbank und einer dedizierten Ordnerwahl für alle Dateien, siehe Abbildung 9.

Auch die Konfiguration ist ähnlich, beim ersten Start wird der Sachverständige gebeten sein Daten, Briefunterschrift, Bankverbindung, Nummernkreise aber auch seinen länderspezifischen Mehrwertsteuersatz einzugeben.

Es gibt sowohl Privat- als auch Gerichtsgutachten zur Auswahl, wobei auch danach alle Personen welche zu diesem Akt gehören angelegt werden. Auch ein Notizbereich ist vorhanden.

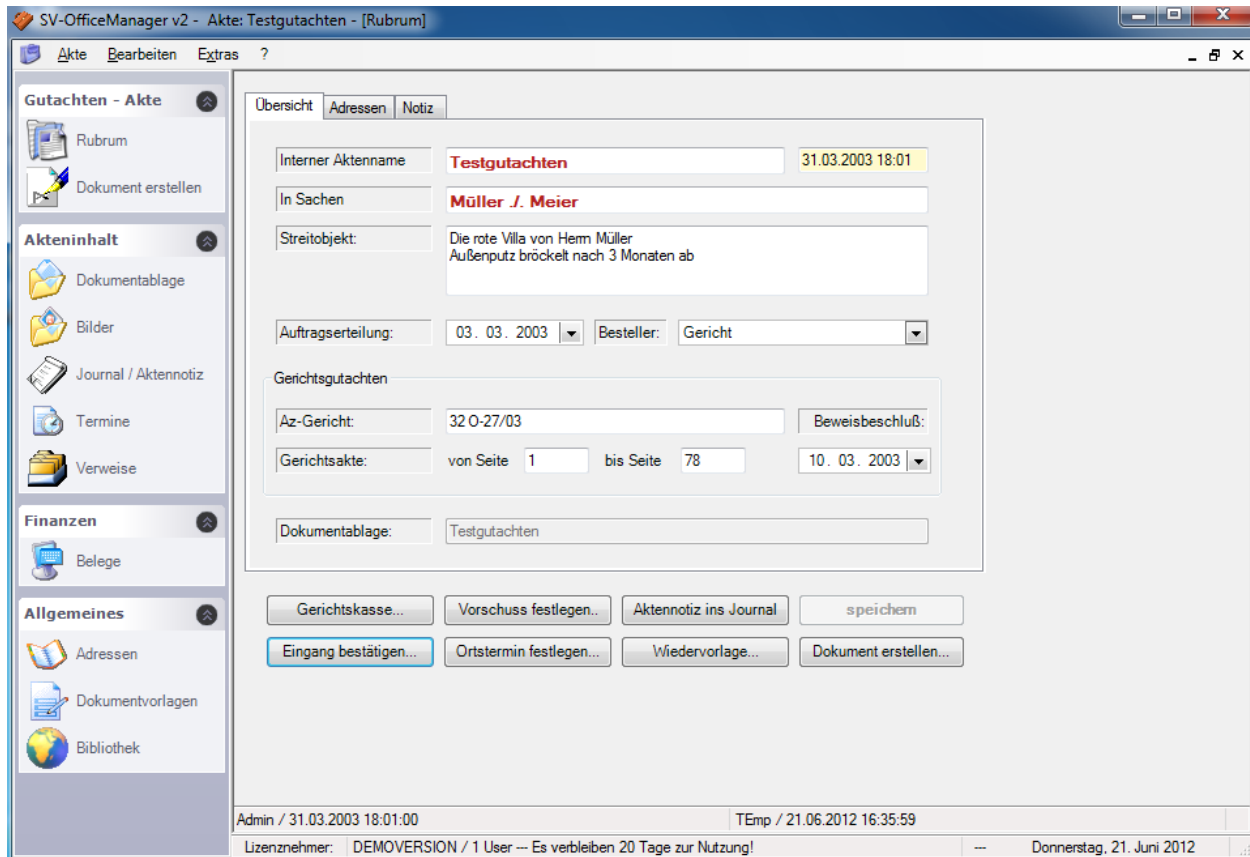


Abbildung 9: Softwarevergleich SV-OfficeManager 2.0

Im Journal können Gebühren eingetragene werden, also Reisekosten, Aktenstudium, etc. und im Status wird definiert ob dieser Eintrag zu berechnen ist. Über die Funktion „Dokument erstellen“ kann dann eine „Kostennote“ mit den jeweiligen Positionen erzeugt werden.

Aber auch andere Dokumente können erstellt werden, unter anderem:

- Adresslisten
- Ortsbesichtigungscheckliste
- Ortsbesichtigungseinladung
- Akteneingang Ablehnung / Annahme
- Standardbrief
- Gutachten (vordefiniert)

Leider war es mir nicht möglich auch Funktionen über das Menü anzusprechen, da das Programm jedes Mal einen Absturz erlitt.

### 2.3.4 Fazit

Da dieses Produkt seit 2010 nicht mehr weiterentwickelt wird, kam es zu diversen Programmabstürzen, welche, so glaube ich, auf Neuerungen im Betriebssystem & der .NET Schnittstelle basierten.

Es fehlt auch an einer Erinnerungsfunktion für Termine, Datensicherung und dem einfachen Anpassen von Vorlagen.

## **2.4 GutachterSoft 2.1.8**

Über die Homepage [5] kann eine vollfunktionsfähige, auf 30 Tage begrenzte, Version bezogen werden.

### **2.4.1 Systemanforderungen**

Betriebssystem:	Microsoft Windows XP / Vista / 7
Software:	Microsoft Office 2000 bis 2010 min. .NET 1.1 & 2.0
Speicherplatz:	500MB

### **2.4.2 Installation**

Die 22MB große Installationsdatei wird ausgeführt und nach dem akzeptieren der Lizenzvereinbarung und Auswahl des Installationsverzeichnis, befindet sich ein Symbol auf dem Desktop.

### **2.4.3 Merkmale**

Auch dieses Programm, siehe Abbildung 10, nutzt im Hintergrund eine Microsoft Access Datenbank um Informationen zu hinterlegen, in Kombination mit einem Ablageordner für jegliche Dateien.

Nach dem ersten Start des Programmes wird man über einen Einstellungsassistenten gebeten den Ort des Arbeitsordners einzugeben. Danach startet das System und über die Grundeinstellungen können verschiedene Optionen getroffen werden. Stammdaten, Softwareeinstellungen, Auswahllisten, etc. um nur Einige zu nennen.

Die Startmaske ist sehr einfach gehalten. Man wählt, wie auch bei den anderen Programmen zwischen verschiedenen Arten eines Gutachtens. Es ist möglich nach Gutachten zu suchen bzw. neue anzulegen und bei Doppelklick auf diesen öffnet sich eine neue Maske, wobei in den einzelnen Reitern alle Informationen einzugeben sind.





Abbildung 10: Softwarevergleich GutachterSoft 2.1.8

Die einzelnen Reiter sind nach Daten und Inhaltsverzeichnis des Gutachten gegliedert, das heißt es werden zuerst die Allgemeine Informationen eingetragen, wie die Parteien selbst (Auftraggeber, Kläger, Beklagter, etc.) heißen. Danach kommt der eigentliche Inhalt des Gutachtens:

- Beweisthema
- Fragestellungen
- Ortstermine
- Feststellung
- Zusammenfassung

Aus Vorlagen kann das Aussehen dieses Gutachten verändert werden, wie beispielsweise das Titelblatt.

Es können auch Rechnungen erstellt werden, wobei auch hier aus Vorlagen die einzelnen Positionen der Rechnung vordefiniert übernommen werden können. Über den Status können sowohl Rechnungen als auch das gesamte Gutachten gesteuert werden.

#### 2.4.4 Fazit

Eine sehr ordentliche und nicht überladene Oberfläche lassen das Programm einfach bedienen. Gelingen ist auch die Abarbeitung eines Gutachtens, was fehlt ist eine E-Mail Integration um Briefwechsel zu dokumentieren und mit zu protokollieren

## 2.5 Resümee

	Serverbasiertes System	DMS	Vorlagen	Autom.Glossar	Zeitaufzeichnung	Rechnungslegung	Archivierung	Protokollierung
GutachterForm Plus 6.0	-	-	X	-	-	-	-	-
Gutachten Manager 6.0	-	O	X	-	X	X	O	-
SV-Office Manager 2.0	-	O	X	-	O	O	-	-
GutachterSoft 2.0	-	-	X	-	-	X	O	-

Wie in der Matrix zu sehen ist kann keines der untersuchten Systeme alle Aufgabenstellungen abdecken, sondern erfüllt nur einzelnen Punkte(X) davon und dann auch nur teilweise (O).

Aufgefallen ist auch, dass die Struktur des Gutachtens vorgeschrieben wurde von den einzelnen Programmen, ein manuelles Ändern war zwar in allen Systemen möglich, jedoch konnte der Bericht danach nicht mehr aktualisiert werden. Alle manuellen Änderungen wären dabei verloren gegangen.

Teilweise vorhanden waren ein Dokumentenmanagement System, Zeitaufzeichnungen, Rechnungslegung und die Archivierung. Hier sei anzumerken, dass zwei Systeme über eine Art des DMS verfügten und zwar, dass die Dateien innerhalb des Programmes zugänglich waren. Auch das Archivieren beschränkte sich auf die Sicherung der gesamten Datenbank, ein anschließendes Komprimieren und Verschlüsseln war keine Option.

Nur die Aufgabenstellung der Vorlagen werden von allen Probanden erfüllt, wobei anzumerken ist, dass bei keinem der Systeme das Ändern dieser einen einfachen Prozess darstellt. Es werden sowohl die Skriptsprachen Visual Basic als auch Textfelder eingesetzt, um Daten welche in Dateien bzw. Datenbanken sind in die Arbeitsergebnisse (Gutachten, Rechnungen, etc.) zu importieren.

Nicht zu finden war ein Serverbasierendes System, Automatische Glossar und jegliches Protokollieren der Vorgänge.

### 3 Prozessablauf für die Gutachenerstellung

Mit Hilfe des vorherigen Kapitels und Informationen von Gutachtern, wird nun ein Prozessablauf (Abbildung 11) definiert, wie die Vorgehensweise sein könnte um ein Gutachten zu erstellen. Dabei wird dieser Prozess in vier Abschnitte unterteilt, in jedem dieser Abschnitte können einzelne oder mehrere Punkte der Aufgabenstellung abgewickelt werden.

Die einzelnen Prozesse kommunizieren jeweils mit den anderen, unterliegen aber einem Ablauf, so kann eine Rechnung nur erzeugt werden, wenn auch Zeiten aufgezeichnet werden. Zeiten müssen einem Gutachten zugeordnet sein, wobei das Gutachten selbst einen Auftraggeber hat und nur diesem kann überhaupt die Rechnung ausgestellt werden.

Einer der wichtigsten Punkte für das Erstellen eines Gutachtens ist die Abgrenzung des Auftraggebers, der beteiligten Personen und ob es ein Privat- oder Gerichtsgutachten ist. Zuerst müssen also Daten von Firmen, Gerichten und Personen vorliegen damit diese einem Gutachten zugeordnet werden können, all diese Prozesse werden im Customer Relationship Management (=CRM) erledigt.

Das eigentliche Gutachten selbst besteht, je nachdem welche Art es ist, aus verschiedenen Kapiteln in dem Aufgabenstellung, Ist- & Sollzustand, Ortsbesichtigungen, usw. schriftlich beschrieben werden. Informationen die zu diesem Gutachten führen, also alle Besprechungsprotokolle, Dateien und Bilder, sollten auch in dem Akt verfügbar sein. Dafür definieren wir den Prozess Projekt in dem alle Informationen, Quellen, Parteien und Korrespondenz zu einem Gutachten gespeichert sind.

Während des Arbeitens an einem Gutachten, müssen alle Zeiten und Auslagen aufgezeichnet und diesem zugeordnet sein, damit später eine ordnungsgemäße Rechnung erstellt wird. Dieser Prozess wird im Abschnitt Time & Invoice definiert.

Der letzte Prozess stellt das Konfigurieren des Systems dar. Einstellungen über Zeitkonten, Dateitypen aber auch das Ändern von Vorlagen soll möglich sein.

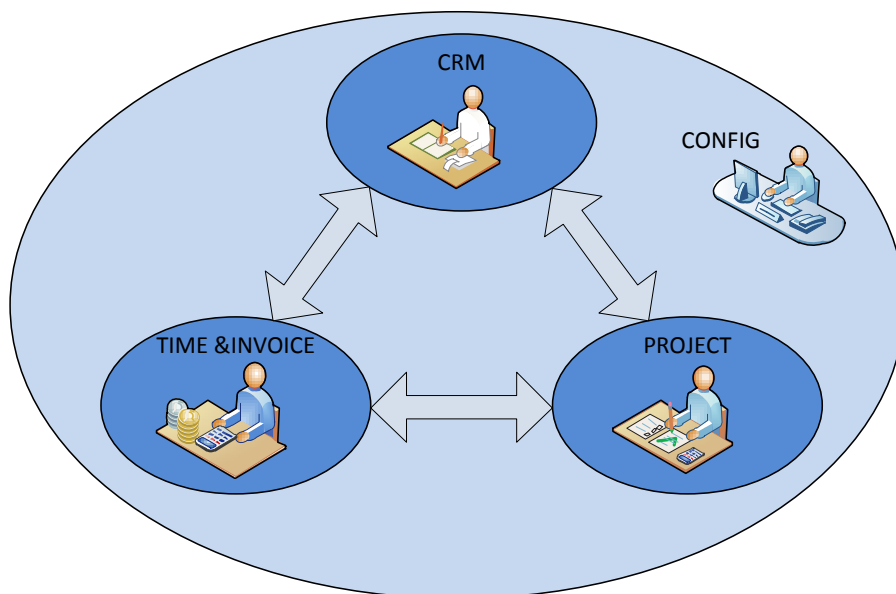


Abbildung 11: Prozessablauf

### 3.1 CRM/DMS

Start eines jeden Gutachtens ist das Erstellen der Stammdaten bezüglich der Parteien die involviert sind. Zuerst müssen Kunden & Personen im System hinterlegt sein, bevor ein Gutachten erzeugt werden kann. Der Startprozess eines jeden Gutachtens ist das Erstellen, Suchen oder Bearbeiten eines Kunden, siehe Abbildung 12.

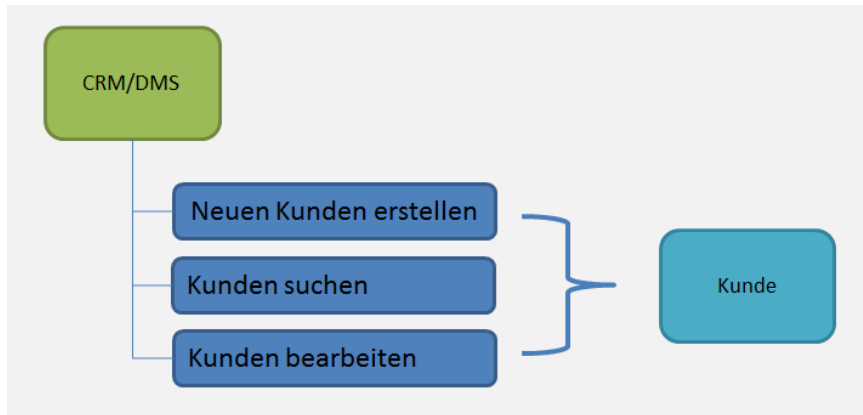


Abbildung 12: Prozessablauf CRM/DMS Allgemein

Dabei wird als Kunde eine Firma, Einzelperson oder Gericht bezeichnet und als Ansprechpartner eine Person, die mit diesem Geschäftspartner in Verbindung steht, wie in Abbildung 13 ersichtlich ist.

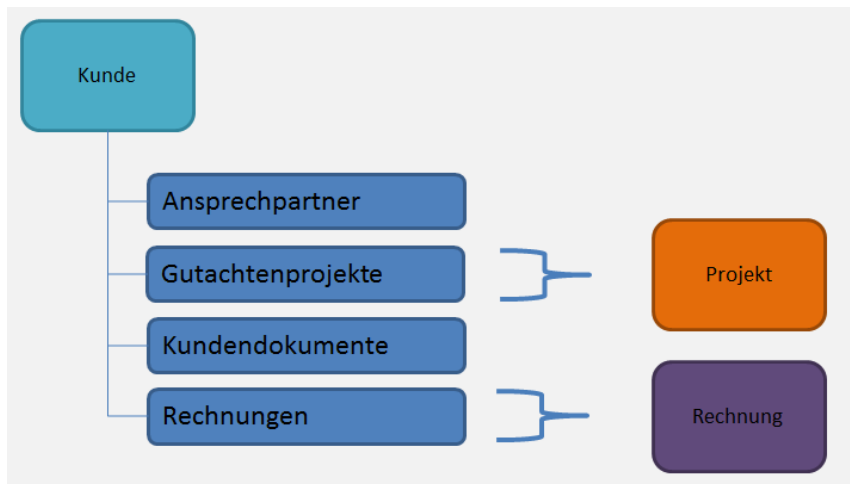


Abbildung 13: Prozessablauf CRM/DMS Kunde

Zu diesen Kunden können Ansprechpartner, Gutachtenprojekte, Kundendokumente und Rechnungen erzeugt und verknüpft werden. Dadurch ist ersichtlich, welche Projekte in Bearbeitung bzw. abgewickelt wurden. Rechnungen werden auch mit dem Kunden verknüpft und zeigen den Status dieser an. Über das Gutachtenprojekt und den Rechnungen kann ein neuer Prozess angestoßen werden.

Die nachstehende Tabelle zeigt, welche Attribute für die einzelnen Objekte vorgesehen sind.

Kunde	Firmenname, Land, PLZ, Ort, Straße, Telefon, Fax, Internet
-------	--

	Adresse, Bemerkungsfeld
Ansprechpartner	Anrede, Titel, Vorname, Nachname, E-Mail, Telefon
Gutachtenprojekt	Projektnummer, Projektstatus, Projektverantwortlicher, Bezeichnung, Thema, Projektart
Kundendokumente	Bezeichnung, Dateiname, Dokumentart, Dokumenttyp
Rechnung	Rechnungsnummer, Rechnungsdatum, Ansprechpartner Rechnung, Bearbeiter, Summe

Da Objekte wie Ansprechpartner, Gutachtenprojekt, Kundendokumente und Rechnungen mit dem Kunden verknüpft sind, erhalten diese alle Information welche direkt am Kunden eingetragen wurden.

### 3.2 Projekt

Wurde der Auftraggeber des Gutachtens erstellt, kann ein Gutachtenprojekt zu diesem Kunden initiiert werden. Alle nötigen Dokumente, involvierte Personen, Checklisten können strukturiert und abgeschlossen zu einem Projekt abgelegt werden, siehe Abbildung 14.

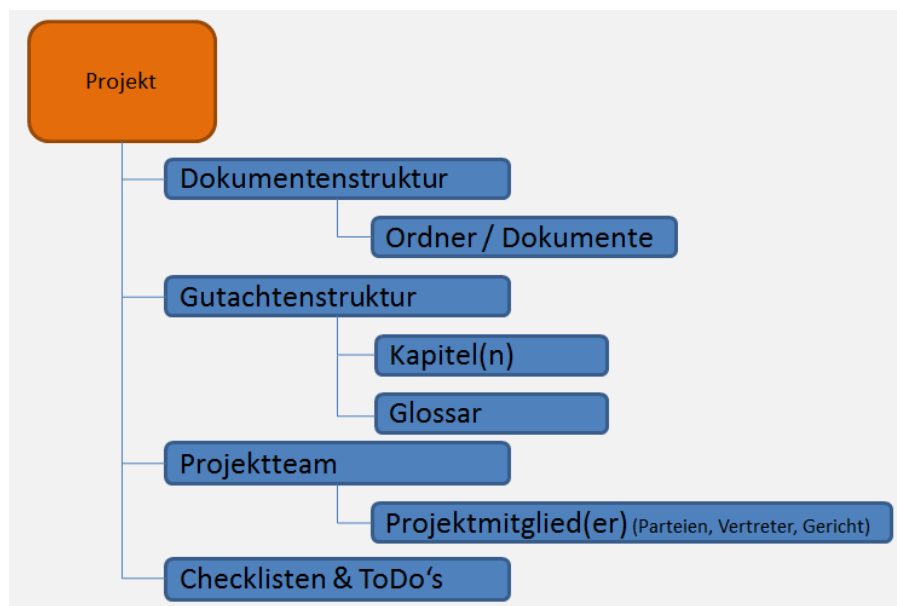


Abbildung 14: Prozessablauf Projekt

Die Dokumentenstruktur dient um Dokumente, wie beispielsweise Protokolle, Spezifikationen, E-Mails aber auch das Gutachten selbst abzulegen. Die Gutachtenstruktur kann mit seinen Kapiteln beliebig verschachtelt werden und zu einem späteren Zeitpunkt wird das Glossar angelegt. Das Gutachtenstruktur-Objekt stellt die Wurzel für das Gutachten dar, dort wird auch der Titelseiteninhalt definiert. Im Projektteam ist nicht nur der Auftraggeber definiert, sondern auch alle anderen beteiligten Parteien, dazu ist es nötig, diese im Prozess CRM/DMS zu erzeugen. Die Checklisten dienen um Aufgaben, die für ein Gutachten notwendig sind aufzuzeigen und diese dann abzuarbeiten.

Der Prozess Projekt beschäftigt sich also mit allen Daten, welche für ein Gutachten relevant sind.

Die nachstehende Tabelle zeigt, welche Attribute für die einzelnen Objekte vorgesehen sind.

Gutachtenprojekt	Projektnummer, Projektstatus, Projektverantwortlicher, Bezeichnung, Thema, Projektart
Dokumentenstruktur	Bezeichnung
Dokumente	Bezeichnung, Dateiname, Dokumentart, Dokumenttyp
Gutachtenstruktur	Bezeichnung, Titelseiteninhalt
Kapitel	Kapitelnummer, Bezeichnung, Inhalt
Glossar	Bezeichnung, Definition
Projektteam	Bezeichnung
Projektmitglied	Personenname, Rolle im Projekt
Checklisten	Bezeichnung, Status

### 3.3 Time & Invoice

Um später eine Rechnung ausstellen zu können, ist es nötig alle Zeiten und Auslagen aufzuzeichnen. Dieser Prozess kann parallel zu den anderen abgearbeitet werden, da es erforderlich ist, Zeiten und Rechnungen auch während eines Projektes einzuarbeiten.

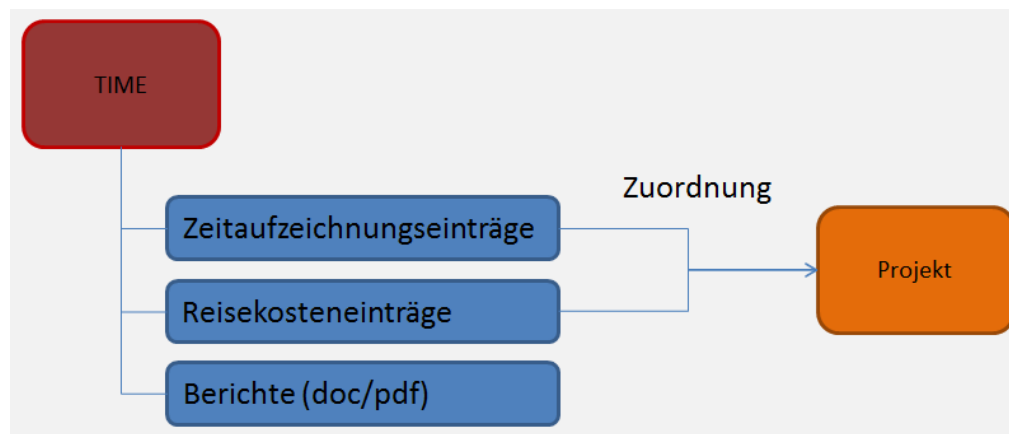


Abbildung 15: Prozessablauf Zeitaufzeichnung

Zeitaufzeichnungseinträge, Reisekosteneinträge und Auslagen (Abbildung 15) welche aufgewendet werden, sind einem Projekt zuzuordnen. Damit wird es möglich Rechnungen zu erstellen.

Einer Rechnung (Abbildung 16) werden Rechnungspositionen erstellt, welche sich aus den einzelnen Gebühren eines Sachverständigen zusammensetzen. Wie bereits aus dem Prozess CRM/DMS ersichtlich, sind Rechnungen dem Auftraggeber zugeordnet.

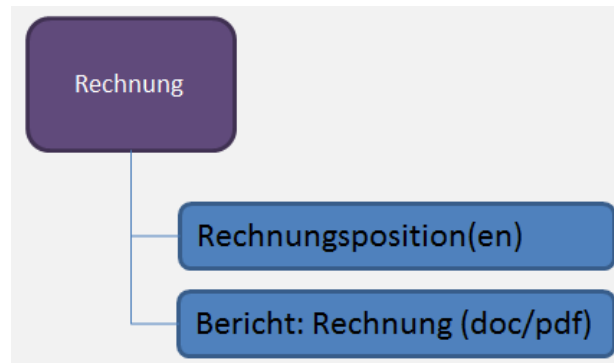


Abbildung 16: Prozessablauf Rechnungslegung

Die nachstehende Tabelle zeigt, welche Attribute für die einzelnen Objekte wesentlich sind.

Zeitaufzeichnungseinträge	Mitarbeiter, Datum, Zeit von, Zeit bis, Pause, Projekt, Kontentyp,
Speseneinträge	Mitarbeiter, Datum, Auslagenersatzeintrag, Projekt
Rechnung	Rechnungsnummer, Rechnungsdatum, Ansprechpartner Rechnung, Bearbeiter, Summe
Rechnungspositionen	Positionsnummer, Bezeichnung, Menge, Mengeneinheit, Preis

### 3.4 Config

Der Prozess Config (Abbildung 17) ist ein eigenständiger Prozess, der zum Initialisieren des Systems und zur Änderung von Basisinformationen dient. Damit zum Beispiel auf den einzelnen Berichten die Kopf- & Fußzeile stimmt, ist es notwendig, dass gewisse Unternehmensinformationen definiert werden. Auch Angaben zum Sachverständigen selbst, müssen in diesem Arbeitsablauf erzeugt und hinterlegt sein.

Die einzelnen Vorlagen und Textbausteine wie Rechnungsbericht, Gutachtenbericht, Zeitaufzeichnungsbericht, usw. sind auch hier hinterlegt und können gegebenenfalls angepasst werden. Das globale Glossarverzeichnis kann ebenfalls, da es durch einzelne Gutachten anwächst, verändert und gewartet werden.

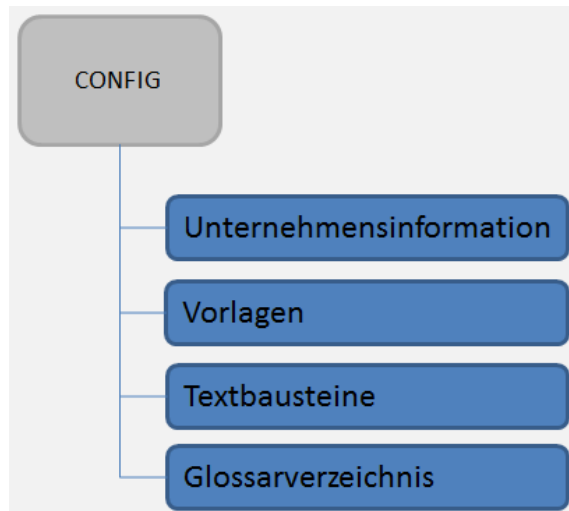


Abbildung 17: Prozessablauf Konfiguration

Die nachstehende Tabelle zeigt, welche Attribute für die einzelnen Objekte bedeutsam sind.

Unternehmensinformation	Firmenname, Dokumenttypen, Kontentypen, Länder, Gebühren, Sachverständiger
Vorlagen & Bausteine	Rechnung, Gutachten, Zeitaufzeichnung
Glossarverzeichnis	Bezeichnung, Definition

## 4 Entwicklungsumgebung

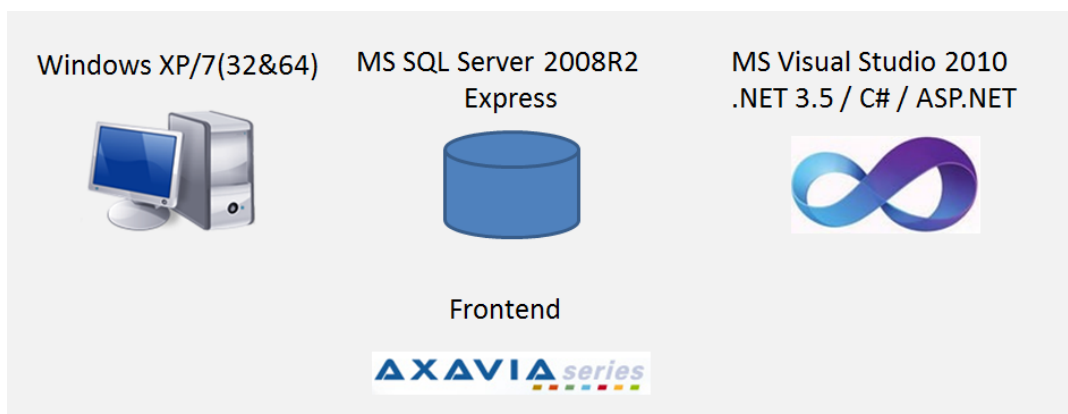


Abbildung 18: Entwicklungsumgebung

Das System wird für das Betriebssystem Windows 7 (64bit) entwickelt. Da die Komponenten aber auch abwärtskompatibel sind, kann auch ab Windows XP damit gearbeitet werden, siehe Abbildung 18. Es liegt sowohl ein 32bit als auch 64bit Client der Installation bei.

Als Datenbank kommt der Microsoft SQL Server Express 2008R2 [6] zum Einsatz. Die verwendete Programmiersprache ist C# [7] unter Berücksichtigung der .NET 3.5 [8] und ASP.NET [9] Bibliotheken.



Das Frontend für die Darstellung der Daten heißt AXAVIASeries [10] und wird von der Firma AXAVIA GmbH vertrieben und entwickelt. Dieses wird in einem eigenen Kapitel behandelt.

## 4.1 Datenbank

Wie bereits erwähnt, wird die Datenbank über die frei verfügbare Version des Microsoft SQL Express 2008R2 zu Verfügung gestellt. Alle Daten, außer Dateien, werden in dieser gespeichert. Es ist aber auch denkbar eine andere Version des Microsoft SQL Servers zu verwenden, solange die Version größer ist als SQL Server 2000.

Es ist sowohl möglich die Datenbankinstanz lokal als auch im Netzwerk zu betreiben, da das Frontend über eine eigene Schnittstelle darauf zugreift.

Wie diese Datenbank installiert und konfiguriert wird kann im Anhang nachgelesen werden.

## 4.2 Entwicklungssprache

Die integrierte Entwicklungsumgebung (=IDE) ist Microsoft Visual Studio 2010. Als Programmiersprache wird C# angewandt, auch die beiden Bibliotheken .NET 3.5 für den Client und ASP.NET für die Webapplikation werden genutzt.

Der Client verfügt über alle Funktionalitäten, die über die Aufgabenstellung definiert wurden. Die Webapplikation kann auf Projekte zugreifen, um wichtige Informationen auszulesen, aber auch Fotos und andere Dateien, welche mit Hilfe eines Smartphones oder Tablets bei der Beweissicherung aufgenommen wurden, hochzuladen.

# 5 Frontend AXAVIASeries

AXAVIASeries ist eine durchgängige Projektierungssoftware für kaufmännische und technische Projekte und verwaltet alle charakteristischen Daten. Es gibt sieben voneinander unabhängige Module, die entweder einzeln oder in Kombination zu erwerben sind, diese kommunizieren eng miteinander und arbeiten prozessübergreifend.

Als Basis dient eine Datenbank, basierend auf Microsoft SQL Server, diese wird über eine interne Schnittstelle der Software angesprochen, um die Daten graphisch darzustellen. Es wird also nie direkt mit der Datenbank gearbeitet, sondern nur mit Hilfe des AXAVIASeries-Client und dessen Funktionalität, dazu zählen u.a.:

- Datensätze definieren, erzeugen, löschen
- Verknüpfungen definieren, erzeugen, lösen
- Ansichten definieren, öffnen

Dank der zentralen Anlaufstelle für Informationen wird sichergestellt, dass jeder Benutzer des Systems immer mit den aktuellsten Daten arbeitet.

Die Benutzerverwaltung in AXAVIASeries wird durch ein „Role based access control“-Zugriffsmodell gelöst. Dabei werden Rechte an Rollen, und des weiteren Benutzer an Rollen, geknüpft, um sicherzustellen, dass nur Personen mit den notwendigen Zugriffsrechten Informationen einsehen dürfen.

Des Weiteren kann die Benutzeroberfläche individuell und dynamisch verändert werden, um sicherzustellen, dass nur gewisse Rollen im System, Zugriff auf Daten erhalten. Für diese Arbeit wurden zwei statische Rollen definiert. Die des Konfigurators, welcher nur

eine Ansicht zum Konfigurieren des Systems hat und jene des Benutzers, welcher dann über den definierten Prozess sein Gutachten abarbeitet. Es sind auch dynamische Rollen im System und zwar kann der Benutzer welcher ein Projekt erstellt hat, weitere Personen mit Hilfe des Projektteams Zugriff zu diesem gewähren. Dies ist insbesondere in einem Mehrbenutzersystem hilfreich.

Es stehen verschiedene Ansichtsdialoge (Baumansicht, Eigenschaftsansicht, Tabellen & Suchtabellen, Formulare, Kalender, etc.) zur Verfügung, wobei die einzelnen Dialoge untereinander Daten austauschen.

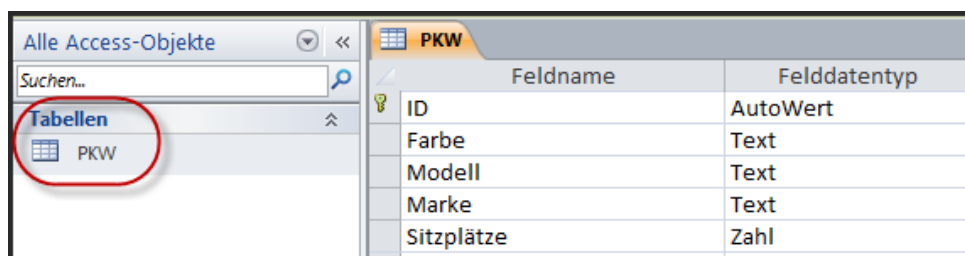
Das Frontend AXAVIAseries arbeitet mit zwei Ebenen und zwar der Modell- und Datenebene.

## 5.1 Allgemein

Um zu verstehen wie beide Ebenen miteinander kommunizieren wird in diesem Abschnitt das Verfahren vorgestellt, wie AXAVIAseries in einer relationalen Datenbank das Objektorientierte Modell abbildet.

### 5.1.1 Relationale Datenbank

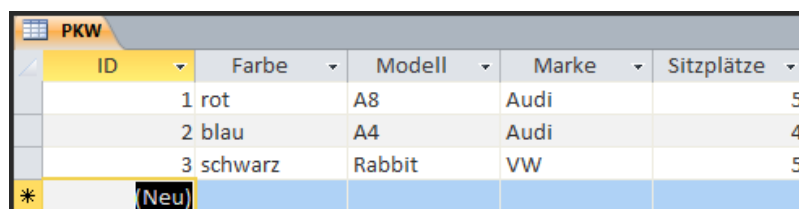
Eine relationale Datenbank (=rel.DB) besteht aus einer oder mehreren Tabellen, wobei für jede ein oder mehrere Attribute definiert werden. Im nächsten Bild (Abbildung 19) wird eine Microsoft Access Tabelle „PKW“ und die Attribute „Farbe, Modell, Marke, Sitzplätze“ erzeugt.



Feldname	Felddatentyp
ID	AutoWert
Farbe	Text
Modell	Text
Marke	Text
Sitzplätze	Zahl

Abbildung 19: Relationale Datenbank Modell MS Access

Dies könnte man auch als Modellebene betrachten, die definiert welche Werte für einen PKW eingetragen werden können. Die nächste Abbildung 20 zeigt die Datenebene, welche drei Datensätze beinhaltet.



ID	Farbe	Modell	Marke	Sitzplätze
1	rot	A8	Audi	5
2	blau	A4	Audi	4
3	schwarz	Rabbit	VW	5
*	(Neu)			

Abbildung 20: Relationale Datenbank Daten MS Access

Es wird also für jede Art von Objekt eine eigene Tabelle mit den dazugehörigen Attributen erzeugt.

### 5.1.2 Objektorientiertes Prinzip

Das Objektorientierte Prinzip (=OOP), siehe dazu Abbildung 21, beschreibt unter anderem Klassen, Attribute und Instanzen dieser.

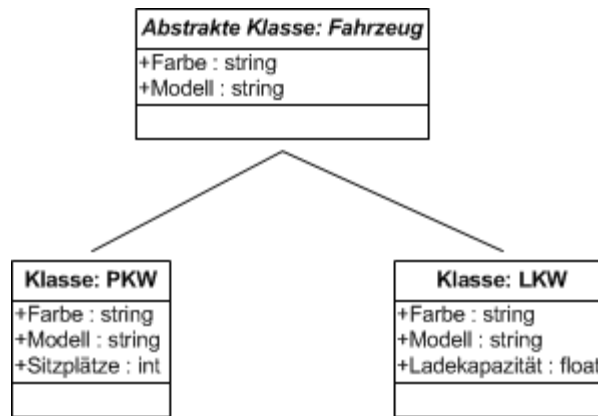


Abbildung 21: Objektorientiertes Prinzip

Es wird eine abstrakte Klasse „Fahrzeug“ mit zwei Attributen definiert und davon zwei abgeleitete, welche jeweils das Attribut „Sitzplätze“ und „Ladefähigkeit“ enthält. Über das Konzept der Vererbung wurden die Attribute der abstrakten auch auf die abgeleitete Klassen übergeben. Dieses UML-Diagramm definieren wir als Modellebene. Die Datenebene zeigt das nächste Bild (Abbildung 22), es werden Instanzen dieser Klassen angelegt.

Instanzen der Klasse  
PKW & LKW



Abbildung 22: Instanzen von Klassen

### 5.1.3 Überführung des Objektorientiertes Prinzip in rel. Datenbank

Um genau dieses Prinzip der Vererbung über eine Datenbank abzubilden, ist es nötig zwei Tabellen zu erzeugen.

Die Objektdefinitionstabelle (T\_OBJDEF) enthält drei Attribute, wobei einer davon auf anderen Eintrag in genau dieser Tabelle zeigen kann.

Die Attributdefinitionstabelle (T\_ATTRDEF) enthält Informationen über den Feldtypen und zeigt auf einen Eintrag in der Objektstabelle, siehe Abbildung 23.

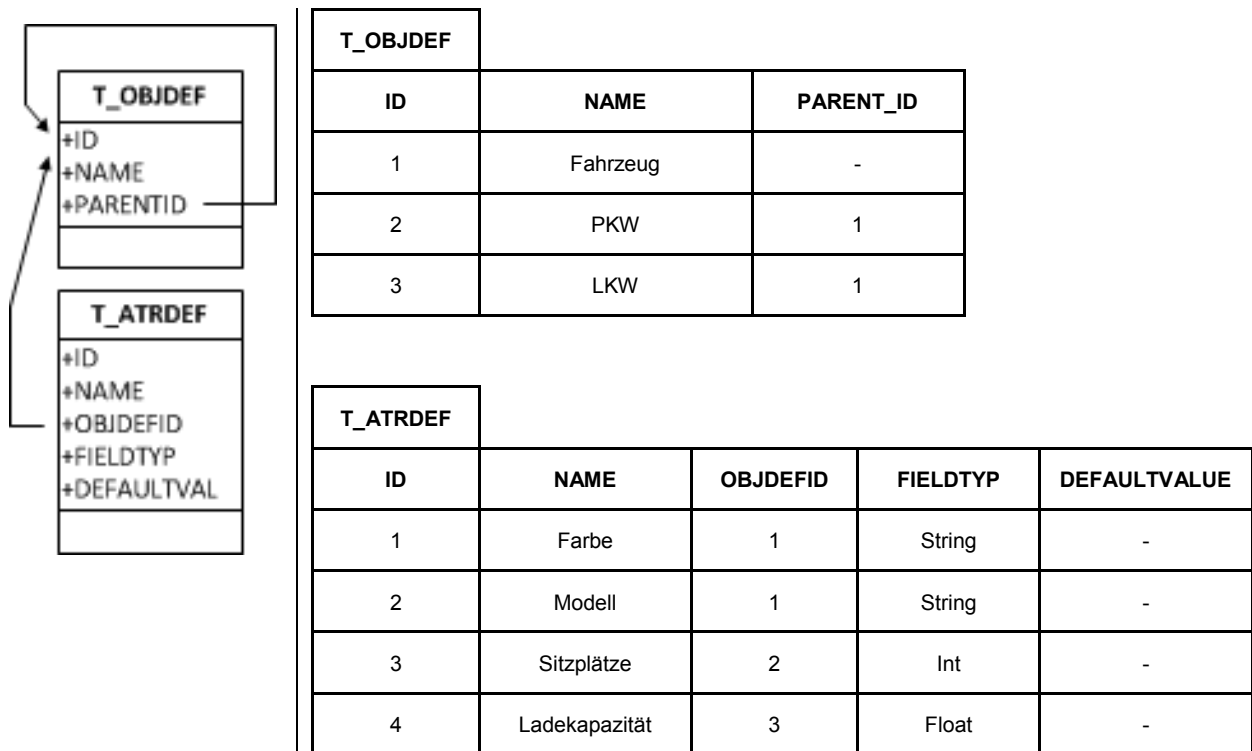


Abbildung 23: Objektorientiertes Prinzip mit rel. Datenbank

Über diese Verfahren kann die Modellebene definiert werden.

Damit man alle Attribute erhält welche für die Klasse „PKW“ definiert sind, sucht man sich dessen OBJDEFID, welche „2“ ist und alle Attribute dieser in der Attributtabelle. Da die Klasse „PKW“ einen Eintrag in der PARENT\_ID enthält, wiederholt man diesen Vorgang für jeden Eintrag in der Objekttable.

Damit nun auch Instanzen dieser Klassen erzeugt werden können, benötigt man eine Tabelle für Objekte und je nach Feldtypen eigene Tabellen.

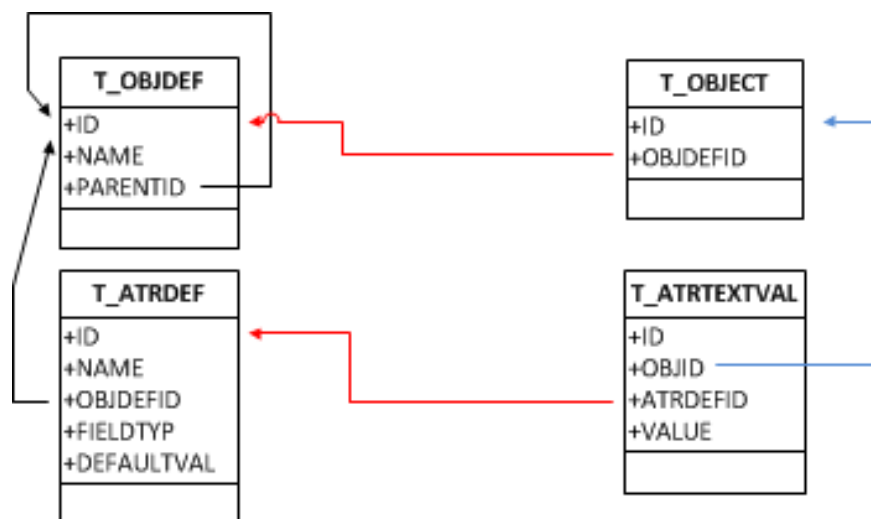


Abbildung 24: Instanz Information zu Klassen

In Abbildung 24 sieht man die beiden Definitionstabellen und für die Datenebene nötig die Objekttable (T\_OBJECT) sowie Texttable (T\_ATRTEXTVAL), die alle Daten des Feldtypen „string“ beinhaltet.

Folgende Vorteile ergeben sich aus diesem Verfahren:

- Erweiterbarkeit
  - Es ist sehr einfach neue Klassen zu definieren bzw. bestehende abzuleiten und mit neuen Attributen zu versehen.
- ACID Prinzip von Datenbanken
  - Atomarität, Konsistenzhaltung, Isolation und Dauerhaftigkeit der Daten wird durch die Datenbank gewährleistet.
- Speicher nur nötig wenn Daten vorhanden sind
  - Durch die Modelleben können Objekte mit dutzenden Attributen definiert werden, jedoch wird nur durch explizites Eintragen eines Wertes auch Speicherplatz benötigt, die Standardwerte ergeben sich aus der Attributdefinition
- Funktionalität lässt sich über das Frontend steuern
  - Listenfelder, Checkboxes, Auswahlfelder, usw. werden durch das Frontend definiert und haben keinen Einfluss auf die Datenbank.

Die Kehrseite ist, dass die SQL-Statements hoch optimiert werden müssen und einzelnen Tabellen, wie die des Feldtypen „string“, sehr schnell wachsen.

## 5.2 Überblick

AXAVIASeries unterteilt sich in einen Windows Client, es ist aber auch möglich, dass über eine Web Applikation auf Daten zugegriffen wird, siehe Abbildung 25. Je nachdem ob Windows Client oder Web Applikation, stehen verschiedene Ansichten & Dialoge zu Verfügung.

Über den Businesslayer, der aus Modulen und den Contentserver besteht, kann auf den Datalayer zugegriffen werden. Dieser Datalayer kontrolliert, optimiert und reiht die eintreffenden SQL Statements, die dann zum SQL Server weitergeleitet werden.

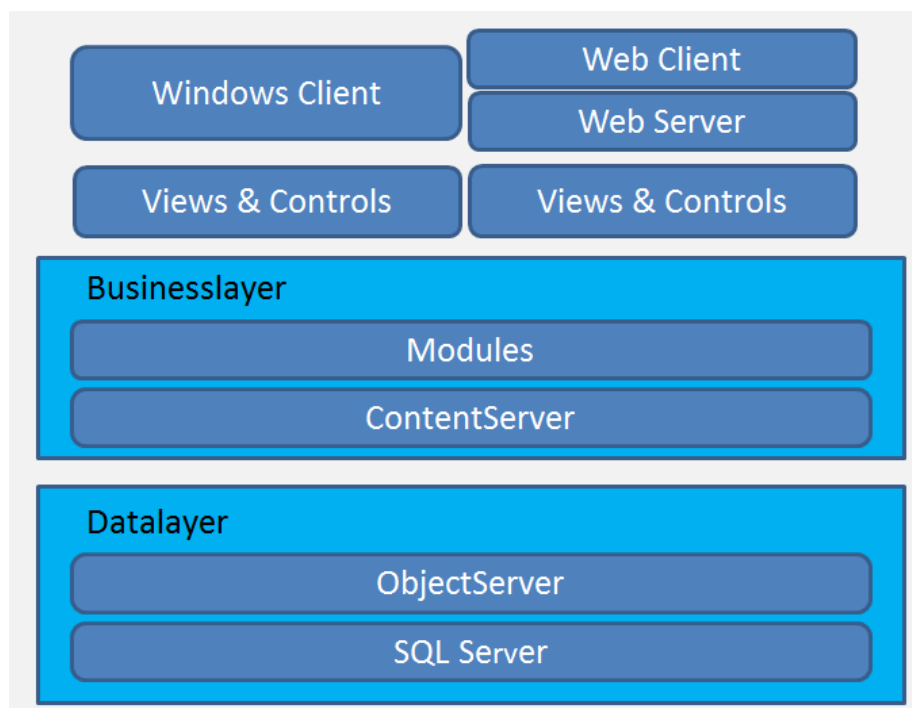


Abbildung 25: Überblick Frontend AXAVIASeries

## 5.2.1 Windows Client

Der Windows Client besteht aus einer oder mehreren Ansichten, die sich wiederum aus unterschiedlichen Dialogen zusammensetzen. Dabei zeigt das nächste Bild (Abbildung 26) die strukturierte Darstellung der Konfigurationsansicht „EXPERTISE\_Configuration“ in AXAVIASeries, die für diese Arbeit erstellt wurde. In 1) sieht man die einzelnen Dialoge, die verwendet werden und 2) zeigt die einzelnen Formularkonfigurationen der einzelnen Objekte. Die Information, wie Dialoge mit einander kommunizieren, ist als Binärform in den Attributen gespeichert.

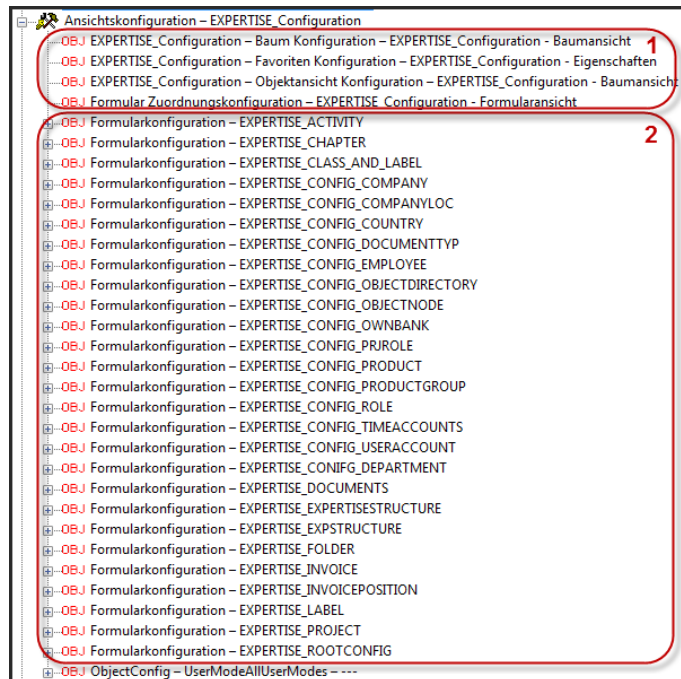


Abbildung 26: Ansichtskonfiguration EXPERTISE\_Configuration

Das Ergebnis sieht man im nächsten Bild (Abbildung 27), wobei links die Baumansicht und rechts der Dialog für das Formular „EXPERTISE\_CONFIG\_COUNTRY“ dargestellt ist.

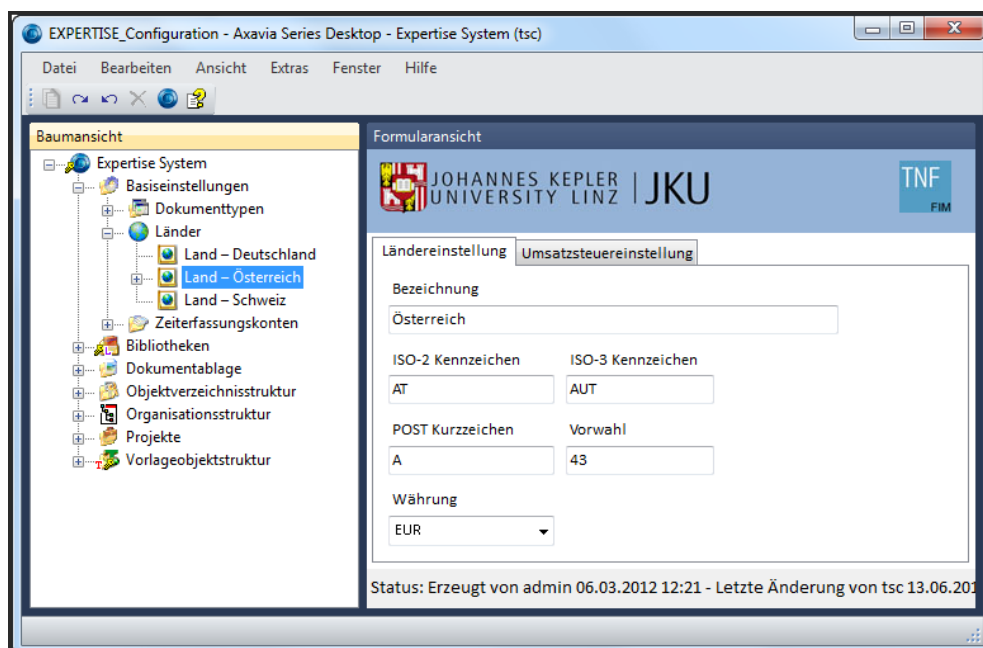


Abbildung 27: Ansicht EXPERTISE\_Configuration

## 5.2.2 Web Client

Damit auch über einen Browser auf das System zugegriffen werden kann, ist es notwendig einen Web Server (IIS) zu installieren in dem eine ASP.NET Anwendung läuft. Diese regelt den Zugriff auf den SQL Server und besitzt das allgemeine Layout (CSS). Die einzelnen Textfelder, Buttons, etc. und wie sich die Webseite verhält kann in AXAVIASeries konfiguriert werden. In Abbildung 28 ist zu sehen, wie sich die Webseite „Dokumente „ verhält und welche Elemente diese beinhaltet.

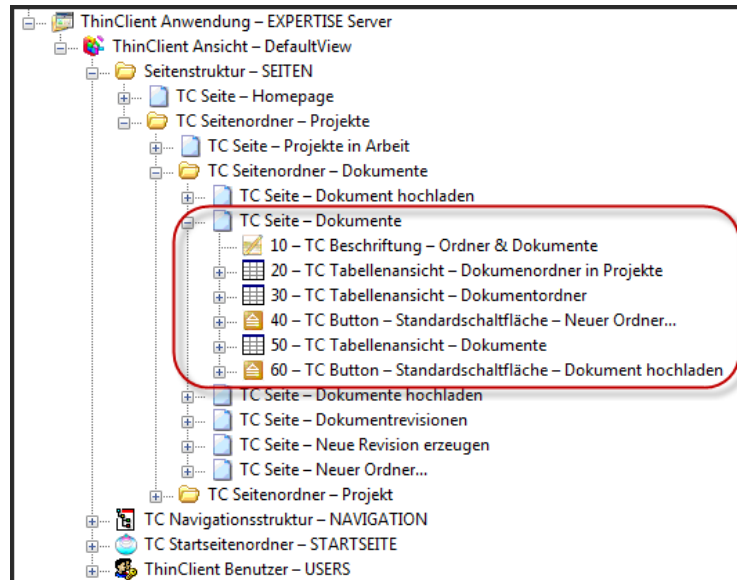


Abbildung 28: Web Client – Konfiguration

Wie diese Konfiguration nun im Browser aussieht wird in nächsten Bild (Abbildung 29) ersichtlich. Die einzelnen Elemente wie Beschriftungen, Tabellen und Schaltflächen sind wie in der Webkonfiguration dargestellt. Das Verhalten der einzelnen Elemente wird in den Attributen gesteuert.

Bezeichnung	Status	Letzte Änderung von	Letzte Änderung am	Revision	
moving_walls	In Arbeit	tsc	21.06.2012 09:45	moving_walls.jpg	0
2011-09-13 19.41.07	In Arbeit	tsc	21.06.2012 09:45	2011-09-13 19.41.07.jpg	0
2011-08-04 13.52.57	In Arbeit	tsc	21.06.2012 09:45	2011-08-04 13.52.57.jpg	0
2011-08-23 10.49.37	In Arbeit	tsc	21.06.2012 09:45	2011-08-23 10.49.37.jpg	0
2011-08-03 14.52.03	In Arbeit	tsc	21.06.2012 09:45	2011-08-03 14.52.03.jpg	0

Abbildung 29: Web Client - Ordner & Dokumente

### 5.2.3 Businesslayer

Diese Schicht besteht aus den einzelnen Modulen, sowie dem Contentserver und wurde in der Programmiersprache C# mit der .NET Bibliothek erstellt. Anwender nutzen nur die Funktionalität in dieser Schicht, für Administratoren ist es aber möglich über eine API und einem eigenen Control direkten Einfluss auf das Verhalten der Software zu nehmen bzw. dieses anzupassen. Der Contentserver erweitert die Funktionalität des Objectserver um:

- Spezielle Feldtypen
  - Mehrsprachigkeit
  - Einheitsfelder
  - Benutzerbezogene Felder
  - Clientbezogene Felder
- Steuerelementtypen
  - Auswahlfelder mit/ohne Suchfunktionen
  - Texteditoren
  - Datum/Zeit
  - Datei- / Pfadauswahl
- Formelbasierte Inhalte
  - Attribut berechnet sich einen Wert aus andere(n) Attribut(en).
  - Attribut setzt Abfrage ab.
- Kommandoorientierte Ereignisse
  - ObjectCreated, ObjectDeleted , usw.

Damit aber auch real mit der Software gearbeitet wird und es keine Plattform zum prototyping ist, gibt es sieben Module mit denen Arbeitsprozesse in Unternehmen abwickelt werden können.

Folgende Module wurden in und für die Software bereits entwickelt:

- ECM – Organisation und Dokumentenmanagement
  - Hilft bei der Abwicklung interner Prozesse und stellt die Basis für ein Dokumentenmanagement bereit.
- CRM – Marketing und Vertrieb
  - Damit können gezielt Informationen transparent, gruppiert und leicht auffindbar abgelegt werden.
- PM – Projekt und Prozessmanagement
  - Koordiniert Anforderungen, Aufgaben und Arbeitspakete und teilt diese Personenkreise zu.
- ERP – Beschaffung und Auftragsabarbeitung
  - Stellt den Prozess der Angebotsphase bis zur Rechnungslegung dar.
- PPS – Fertigungsplanung



- Steuert, überwacht und plant Ressourcen und Materialien zum Erzeugen von Produkten.
- EDB – Engineering und Konstruktion
  - Strukturierte Abbildung von Komponenten für den Anlagenbau, Verfahrenstechnik und Automatisierungstechnik.
- PLM – Wartung und Instandhaltung
  - Dokumentiert und überwacht den Lebenszyklus von Komponenten.

Die einzelnen Module beinhalten sowohl eine oder mehrere grafische Benutzeroberflächen, modulspezifische Funktionen, Dokumentation, Standardberechtigungen und Beispieldaten. Es wird also das Aussehen und die Funktionalität über diese Module bzw. in der Modellebene gesteuert bzw. erweitert.

Ein weiterer wichtiger Punkt ist, dass die einzelnen Module die Objekte und deren Attribute in der Modellebene definieren, erst damit wird es möglich, dass der Benutzer Instanzen davon bearbeitet.

Wie die Software angepasst werden kann und wie diese für das Thema in dieser Diplomarbeit gemacht wurde wird in den nächsten Abschnitten erklärt.

#### 5.2.4 Datalayer

Die unterste Schicht, der Objectserver, stellt Algorithmen bereit, um eintreffende Abfragen an den SQL Server, die durch die Benutzeroberfläche ausgelöst werden, zu optimieren. Außerdem wird ein Cache für Objekte und Attribute in diesem Layer genutzt, damit gleiche Abfragen nicht erneut ausgeführt werden.

Um dies besser zu verstehen, definieren wir eine Benutzeroberfläche, die aus drei Ansichtsdialoge (wie in Abbildung 30 ersichtlich) besteht. Eine Baumansicht, die Informationen über das markierte Objekt, sowohl in einem Eigenschaftsdialog, als auch in einen Tabellendialog sendet.

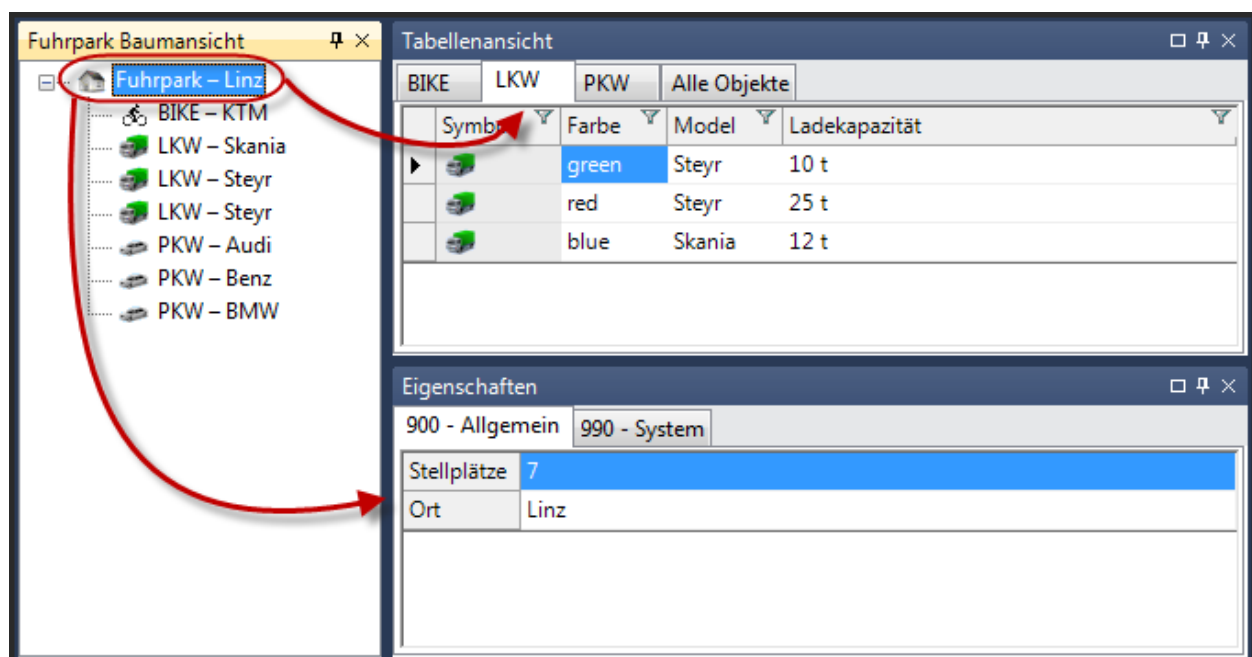


Abbildung 30: Datalayer Beispielsicht

Als Objekt dient uns die Klasse „Fuhrpark“, der unter anderem die Attribute „Bezeichnung, Stellplätze und Ort“ enthält. Wird diese Ansicht initialisiert sieht man nur Daten in der Baumansicht, die anderen beiden Controls sind leer.

In der Baumansicht müssen für das Objekt „Fuhrpark“ zwei Attribute geladen werden und zwar „Bezeichnung und Ort“, in diesem Fall „Fuhrpark - Linz“. Beim Aufklappen dieses Objektes werden weitere Objekte vom Typ „PKW, LKW und BIKE“ wiederum mit dem Attribut „Bezeichnung und Model“ über den SQL Server abgefragt.

Durch das Markieren eines Objektes im Baumdialog werden die beiden anderen Dialoge aktiv und zeigen Informationen zu diesem Objekt an.

Wobei der Eigenschaftsdialog, je nachdem welche Attribute sich im jeweiligen Reiter befinden, diese nun lädt bzw. Informationen welche bereits im Cache vorhanden sind, übernimmt. Auch der Tabellendialog zeigt nun die untergeordneten Objekte des markierten Wurzelobjektes im Baum mit den Attributen an, wobei unterschiedliche Klassen in Reiter unterteilt werden.

Eine weitere Funktionalität des Datalayers ist das Erzeugen eines Objektorientierten Metamodelles aus der relationalen Datenbank, wie bereits im vorherigen Kapitel beschrieben wurde. Anzumerken sei noch, dass diese Schicht der Software in „managed Code“ C# mit der .NET Bibliothek programmiert ist.

All diese Funktionen sind im Datalayer zu finden und bleiben dem Benutzer gänzlich verborgen.

### 5.3 Modellebene

Die Modellebene definiert Objekte, Attribute, Ereignisse und wie sich diese zu anderen Objekten verhalten.

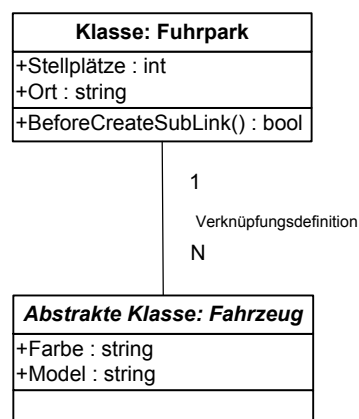


Abbildung 31: Modellebene

In Abbildung 31 sind zwei Objektdefinitionen (Fuhrpark und Fahrzeug), Attributdefinitionen (Stellplätze, Ort, Farbe, Model) und deren Verknüpfungsdefinitionen (1:N) definiert. Dadurch wird erreicht, dass ein oder mehrere Instanzen der abstrakten Klasse Fahrzeug, also auch abgeleitete Klassen davon (z.B.: PKW, LKW, etc.) zu einer Fuhrparkinstanz verknüpft werden können.

Des Weiteren ersichtlich ist ein Ereignis, auf der Objektdefinitionen „Fuhrpark“ und zwar „BeforeCreateSubLink“. Dieses Ereignis könnte dazu dienen, um sicherzustellen das noch genügend Stellplätze vorhanden sind bevor eine neue Instanz eines Fahrzeuges zum Fuhrpark verknüpft wird.

### 5.3.1 Objektdefinition

Alle Objekte, die später in den verschiedenen Ansichten zu Verfügung stehen, können in einer eigenen Modellansicht (Abbildung 32) als Objektdefinitionen definiert werden. Zu diesen Objektdefinitionen werden dann Attributdefinitionen erzeugt, mit den jeweiligen Datentypen. Zuletzt erstellt man noch Verknüpfungsdefinitionen, die Aussagen treffen, welche Objektdefinitionen miteinander verknüpft werden dürfen.

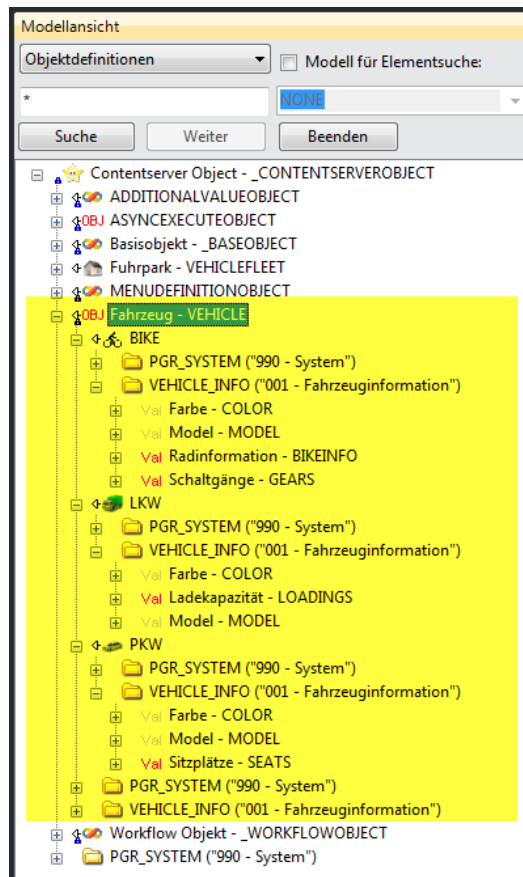


Abbildung 32: Modellansicht der Objektdefinitionen und Attribute

Objektdefinitionen können von anderen abgeleitet werden, dadurch werden alle übergeordnet erstellten Attribut- & Verknüpfungsdefinitionen vererbt, eine Mehrfachvererbung ist nicht möglich. In diesem Bild sieht man drei Objektdefinitionen „Fahrzeug - VEHICLE“, welche als abstrakte Klasse fungiert und zwei Attributdefinitionen „Farbe & Model“ enthält. Die abgeleiteten Klassen „BIKE, LKW, und PKW“ erben diese Attributdefinitionen und enthalten selbst noch jeweils „Radinformation, Schaltgänge, Ladekapazität und Sitzplätze“. Das UML – Diagramm dazu, ohne der Klasse „BIKE“ wurde bereits im Kapitel 5.1.2 dargestellt.

Als Eigenschaften zu Objektdefinitionen kann definiert werden, ob diese abstrakt oder auch nicht ist, was bedeuten würde, ob Instanzen in der Datenebene überhaupt erzeugbar bzw. sichtbar sind. Zu den weiteren einstellbaren Eigenschaften zählen unter anderem:

- Bezeichnung & Beschreibung
- Icon & Erzeugbar in der Symbolleiste einer Ansicht
- Methoden & Ereignisse

### 5.3.2 Attributdefinition

Die Attribute besitzen selbst verschiedene Eigenschaften und ein Verhalten bei Manipulationen.

Eigenschaften	
Allgemein	System
Aggregationsfunktion	Keine
Anzeige-Einheit	---
Attribute	
Ausrichtung	Links
Basis-Einheit	
Bearbeitungsmodus	Erzeugen, Nur Lesen, Löschen, Initiieren, Ändern
Bearbeitungs-Steuerelement	Eingabefeld
Beschreibung	DE ---
Bezeichnung	DE Schaltgänge
Datenbank - Index	nein
Eigenschaft - Abfrage	---
Eigenschaft - Abfrageparameter	DE ---
Eigenschaft - Art	Standard
Eigenschaft - Ausfülldefinition	---
Eigenschaft - Auswahllistendefinition	---
Eigenschaft - Schnellverbindung-Vb.Nr.	---
Eigenschaft - Verhalten beim Importieren	Importieren
Eigenschaft - Verhalten beim Kopieren	Zurücksetzen auf Standardwert falls Wert gleich Standardwert
Eindeutig	False
Eingabeverpflichtung	kann
Falls Werte überschreibbar werden	bereits überschriebene Werte beibehalten
Feldgröße	4
Feldtyp	Long
Formatierung	---
Formel	---
Gruppenliste	
Listenwerte	---
Listenwerte	DE ---
Maximum	---
Mehrfachwert-Dimension	Nein
Minimum	---
Modellbezeichnung	NONE
Name	GEARS
Offline - Konfliktbehandlung	letzte Änderung
Offline - Verhalten beim Abgleich	Abgleichen
Reiterliste	VEHICLE_INFO
Schrittweite	---
Sequenz-Nummer	1

Abbildung 33: Definition der Eigenschaften eines Attributes „GEARS“

In Abbildung 33 ist die Eigenschaft eines Attributes „GEARS“ der Klasse „BIKE“ zu sehen, wobei dies die interne Bezeichnung des Feldes ist. Als sprechende Bezeichnung wurde „Schaltgänge“ gewählt, der Feldtyp ist „LONG“ und 4 Byte groß. Als Steuerelement für die Bearbeitung wurde ein Eingabefeld gewählt. Es ist aber auch möglich ein Auswahlfeld daraus zu generieren, das sein Werte entweder statisch oder dynamisch lädt, wie im nächsten Bild (Abbildung 34) abgebildet ist.

Eigenschaften		
Favoriten	001 - Fahrzeuginformation	990 - System
Farbe	---	
Model	---	
Schaltgänge		
	1	
	2	
	3	
	4	
	5	
	6	

Abbildung 34: Auswahlfeld des Schaltgänge Attributes

Wie bereits erwähnt ist es auch möglich, dass Attribute definiert werden, die sich ihren Inhalt aus anderen Attributen zusammenrechnen, bzw. sogar über Abfragen erzeugen.

Dazu kann in den Eigenschaften einer Attributdefinition eine Formel hinterlegt werden, diese nutzt das Prinzip von Reflection in .NET und der Programmiersprache C# um zur Laufzeit Code zu generieren, wie aus den nächsten beiden Bildern ersichtlich ist (Abbildung 35 und Abbildung 36). Dazu wurde eine Attributdefinition „Radinformation“ auf der abgeleiteten Klasse „BIKE“ erzeugt, die folgende Formel enthält.

Der Inhalt eines Attributes kann entweder statisch sein, die Daten werden also in der Datenbank gespeichert und beim Anzeigen ausgelesen, oder dynamisch generiert, dazu überschreibt man die Methode „GetContent()“. In diesem Beispiel werden in das Attribut „Radinformation“ drei statische Werte von „Farbe, Model und Schaltgänge“ aus der Datenbank abgefragt und als „string“ zurückgeliefert.

```

1 public object GetContent()
2 {
3     string color = this.Object.Values["COLOR"].Contents.ToString();
4     string model = this.Object.Values["MODEL"].Contents.ToString();
5     string gears = this.Object.Values["GEARS"].Contents.ToString();
6
7     return "Bike with the color " + color + " from model " + model + " has " + gears + " gears!!";
8 }

```

Abbildung 35: Formel für Attribut "Radinformation"

Die aktuellen Daten für dieses Feld werden beim Anzeigen dieses aktualisiert, wie im folgenden Bild zu sehen ist.

Eigenschaften	
Favoriten	001 - Fahrzeuginformation 990 - System
Farbe	red
Model	KTM
Schaltgänge	6
Radinformation	Bike with the color red from model KTM has 6 gears!!

Abbildung 36: Auswertung für Attribut "Radinformation"

Zusammenfassend können für Attributdefinitionen unter anderem folgende Einstellungen vorgenommen werden.

- Bezeichnung & Beschreibung
- Steuerelementtypen
  - Eingabefeld, Auswahlfeld, Texteditor, etc.
- Statisch oder dynamische Bindung von Daten
  - Listenfelder, Abfragen, berechnete Auswahllisten, etc.
- API Framework
  - Intellisense

- Formeln
- Abfragen
- Flags
  - Muss / Soll Feld
  - Standardwert, indizierbar, überschreibbar (falls berechnet), etc.

### 5.3.3 Verknüpfungsdefinition

Eine Verknüpfungsdefinition könnte wie folgt aussehen: Man erstellt eine Objektdefinition „Fuhrpark - VEHICLEFLEET“ unter welche man alle Objekte der Klasse „Fahrzeug - VEHICLE“ verknüpfen kann.

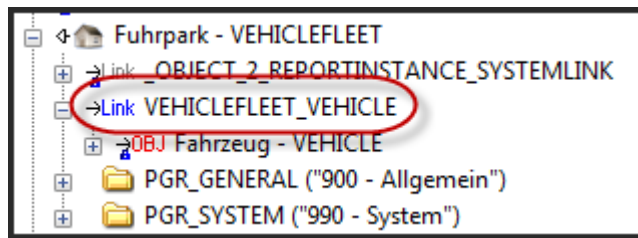


Abbildung 37: Verknüpfungsdefinition

Die Abbildung 37 zeigt eine Verknüpfungsdefinition „VEHICLEFLEET\_VEHICLE“, die die Objektdefinition „Fahrzeug - VEHICLE“ darunter gelistet hat. Diese bedeutet, dass auch alle abgeleiteten Objektdefinitionen von „Fahrzeug - VEHICLE“ über genau diese Verknüpfungsdefinition verbunden werden können. Dazu kommen die Eigenschaften dieser.

Eigenschaften	
Allgemein	System
Aggregierend	nein
Attribute	Sichtbar
Beim Einfrieren Übergeordnete ->	ignorieren
Beim Einfrieren Untergeordnete ->	kopieren
Beim Exportieren Übergeordnete ->	verknüpfen
Beim Exportieren Untergeordnete ->	verknüpfen
Beim Kopieren Übergeordnete ->	verknüpfen
Beim Kopieren Untergeordnete ->	ignorieren
Beim Kopieren: aus Vorlage Übergeordnete ->	ignorieren
Beim Kopieren: aus Vorlage Untergeordnete ->	ignorieren
Beim Löschen Übergeordnete ->	verknüpfen
Beim Löschen Untergeordnete ->	ignorieren
Beim Revisionieren Übergeordnete ->	verknüpfen
Beim Revisionieren Untergeordnete ->	verknüpfen
Berechtigungen vererben	---
Beschreibung	DE ---
Bezeichnung	DE ---
Kardinalität A	1:N
Kardinalität B	1:1
Modellbezeichnung	999-FIM
Name	VEHICLEFLEET_VEHICLE
Priorität für Verbindungserstellung	---
Verbindungsnummer	0

Abbildung 38: Eigenschaften einer Verknüpfungsdefinition

Im Bild (Abbildung 38) oben ist zu sehen, wie sich Objekte welche übergeordnet bzw. untergeordnet bei den jeweiligen Operationen (Kopieren, Löschen, etc.) verhalten sollen.

Es kann also eingestellt werden, ob auch Objekte, die sich unter dem eigentlich zu kopierenden Objekt befinden automatisch mitkopiert bzw. vorselektiert werden. Die Kardinalität A wiederum sagt aus, ob sich ein oder mehrere „Fahrzeug – VEHICLE“ Instanzen unter der Instanz „Fuhrpark – VEHICLEFLEET“ befinden dürfen. Kardinalität B definiert den umgekehrten Weg, in diesem Fall ob es auch möglich wäre das ein und dasselbe „Fahrzeug – VEHICLE“ Instanz zu mehreren „Fuhrpark – VEHICLEFLEET“ Instanzen verknüpft werden könnte. Die Einstellungsmöglichkeiten für die Kardinalität sind „1:1 oder 1:N“, deswegen gibt es zwei davon.

### 5.3.4 Ereignisdefinitionen

Auf Objektdefinitionen ist es möglich vordefinierte Methoden und Ereignisse zu programmieren. Damit kann das Verhalten bei Interaktion mit Instanzen von Objektdefinitionen gesteuert werden. Im nächsten Bild (Abbildung 39) ist zu sehen welche Ereignisse auf jede Objektdefinition anwendbar sind.

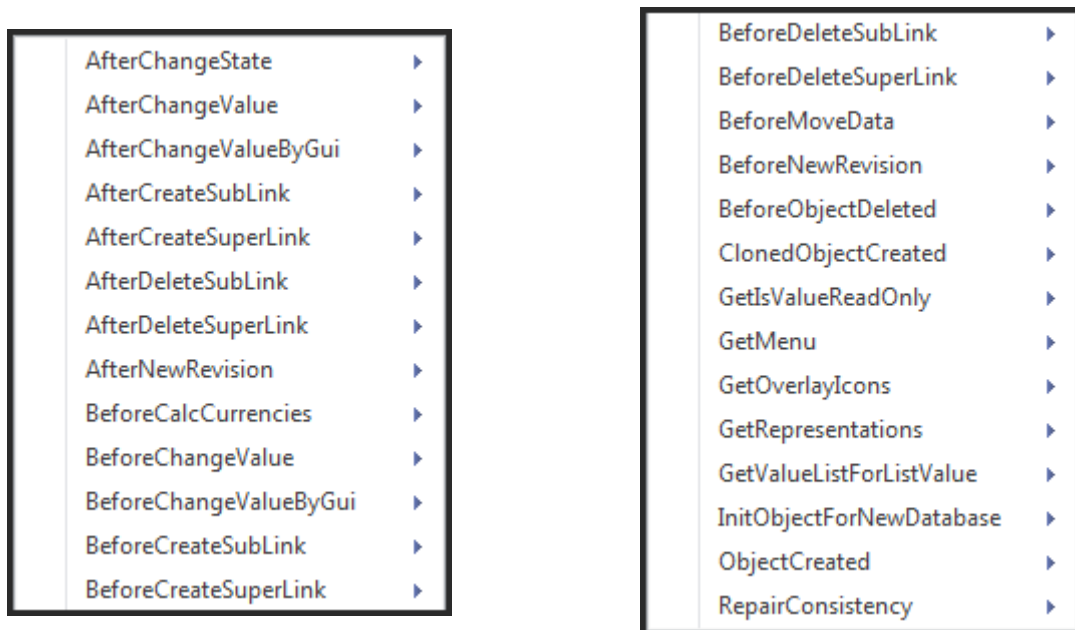


Abbildung 39: Ereignisse auf Objektdefinitionen

Um ein besseres Verständnis dafür zu bekommen, wird das nächste Programmstück dargestellt (Abbildung 40), das als „BeforeCreateSubLink“ definiert wurde.

```

1 public Boolean BeforeCreateSubLink_140194E7090E400(DataObject dataObject, DataObject subobj,
2
3     double freeSpace = (double)(int)dataObject.Values["SPACE"].GetContents(0);
4
5     if (freeSpace != 0)
6     {
7         Query q = new Query(new string[]{"VEHICLE"}, true, LinkDirectionEnum.Sub);
8         ArrayList vehicles = dataObject.LinkedObjects(q);
9         int vehicleCount = vehicles.Count;
10
11         if (freeSpace > vehicleCount)
12             return true;
13         else
14         {
15             cancelText = "No more free space in vehicle fleet!";
16             return false;
17         }
18     }
19     return false;
20

```

Abbildung 40: Beispiel Ereignis "BeforeCreateSubLink"

Dabei wurde auf der Objektdefinition „VEHICLELEET“ eine neue Attributdefinition „SPACE“ vom Feldtyp „LONG“ erzeugt.

Bevor abgeleitete Instanzen der Objektdefinition „VEHICLE“ zu einem Objekt „VEHICLEFLEET“ verknüpft werden, wird überprüft ob noch genug Platz vorhanden ist. Falls ja, wird diese Operation erlaubt und wenn nicht genügend Platz im Fuhrpark verfügbar ist, eine Fehlermeldung ausgegeben.

### 5.3.5 Zusammenfassung

Die Modellebene erlaubt Entwickler und Administratoren nicht nur die Software um Funktionalität zu erweitern bzw. anzupassen, auch die eigentliche Benutzeroberfläche lässt sich ohne Änderungen am Programmcode vornehmen zu müssen, abändern oder sogar neue zu erstellen. Dazu mehr im Kapitel 5.4.6.

Das Paradigma sieht vor dass, angelehnt am Metamodel der Objektorientierten Programmierung Klassen, in unserem Fall Objektdefinitionen, erstellt werden. Die Klassen beinhalten Attribute, in AXAVIASeries Attributdefinitionen und Methoden welche als Ereignisdefinition erzeugt sind.

Relationen zwischen Objekten werden über Verknüpfungsdefinitionen dargestellt.

## 5.4 Datenebene

Die Datenebene (Abbildung 41) zeigt über unterschiedliche Benutzeroberflächen dieser Daten an und lässt genau jene definierten Manipulationen auf diese Objekte zu.



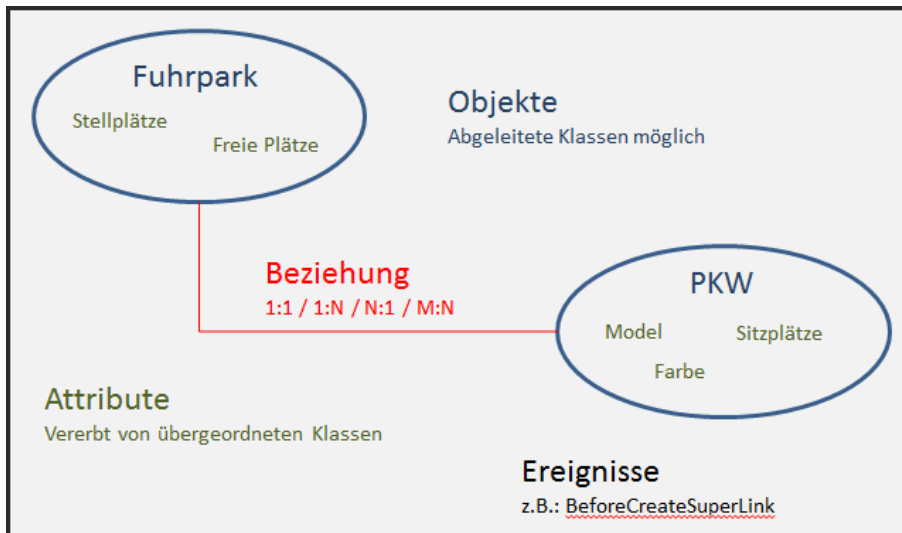


Abbildung 41: Übersicht Datenebene

Über die Modellebene wird abgegrenzt welche Objekte, Attribute, Ereignisse und Verknüpfungen es geben soll. Die Datenebene stellt die Instanzen dieser Definitionen nun anschaulich dar und versteckt alle nötigen Abfragen auf die Datenbank. Es existieren verschiedene Controls, die je nach Zweck in einer Ansicht angeordnet werden können und untereinander kommunizieren. In den nächsten Abschnitten dieses Kapitels werden die Controls vorgestellt, die für die Aufgabenstellung dieser Arbeit nötig waren.

#### 5.4.1 Baumansicht

Die Baumansicht, im englischen „Treeview“ genannt, ist eines der meisten verwendeten Control's im Microsoft Betriebssystem Windows, wenn es darum geht Informationen strukturiert anzuzeigen und darin zu navigieren. Um es kurz zu erklären wie diese in AXAVIASeries verwendet wird, betrachten wir die bereits in der Modellebene veranschaulichten Objektedefinitionen (Abbildung 42).

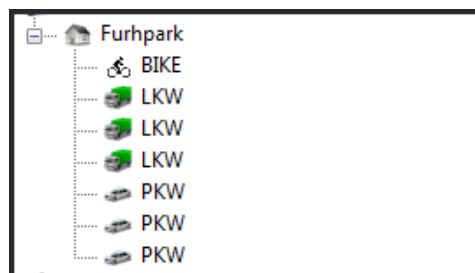


Abbildung 42: Beispiel Baumansicht

Als Wurzelobjekt dient eine Instanz der Objektedefinition „Fuhrpark“, darunter zu finden sind abgeleitete Instanzen von „VEHICLE“. Darüber hinaus ist einstellbar welche Attribute als Information im Baum dargestellt werden, in diesem Fall zeigt es nur ein Attribut an und zwar das des Klassennamens.

Es wäre aber auch möglich für die jeweilige Klasse unterschiedliche Attribute darstellen zu lassen bzw. die Klasse „BIKE“ gar nicht anzuzeigen.

#### 5.4.2 Eigenschaftsdialog

Dieser Dialog (Abbildung 43) enthält verschiedene Reiter, in dem je nach Modellierung gruppiert, die Attribute abgelegt sind.

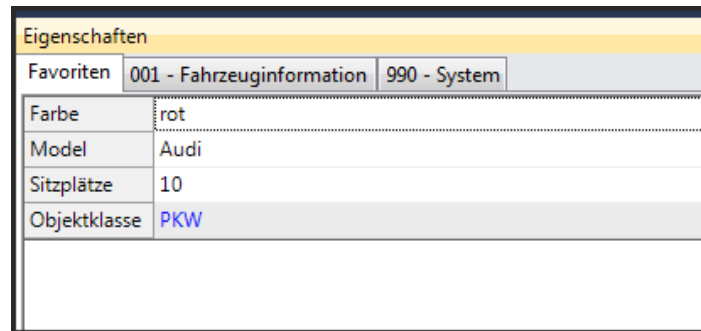


Abbildung 43: Beispiel Eigenschaftsdialog

Einer dieser Reiter „Favoriten“ kann speziell je Ansicht angepasst werden, in diesen werden jene Attribute aus den anderen Reitern zusammengelegt, die für die Art der Bearbeitung wichtig sind.

### 5.4.3 Formulardialog

Für jede Objektdefinition kann auch ein Formular (Abbildung 44) erzeugt und in der Datenbank abgespeichert werden.

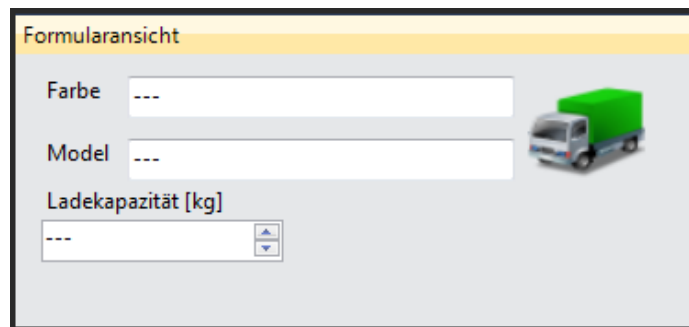


Abbildung 44: Beispiel Formulardialog

Wie aus Microsoft Visual Studio bekannt, gibt es auch in AXAVIASeries einen Formulardesigner (Abbildung 45) in dem die Attribute, die es auf der Objektdefinition gibt, per Drag & Drop in das Formular gezogen werden.

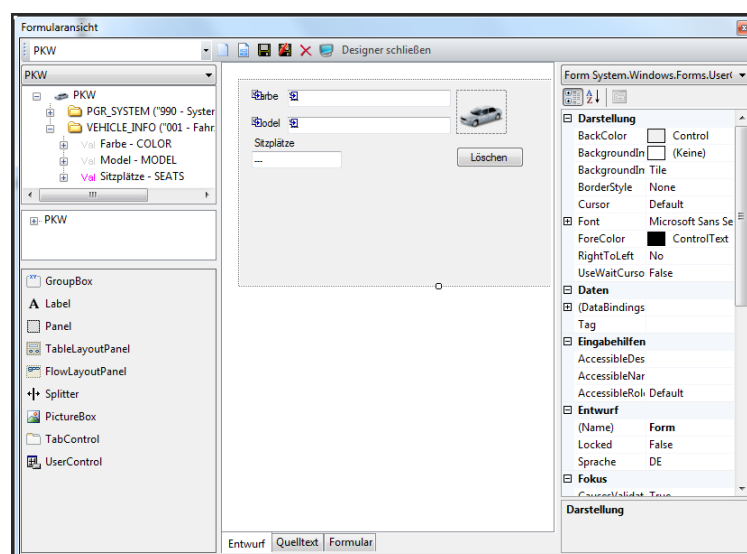


Abbildung 45: Beispiel Formulardesigner

Formulare können als Vorlage Masterformulare beinhalten, dies bedeutet, dass beispielsweise für eine abstrakte Klasse, wie es „VEHICLE“ ist, alle Attribute verarbeitet

und in einem eigenen Formular abspeichert. Erstellt man ein Formular für eine abgeleitete Klasse wie „PKW“, kann als Masterformular, das des „VEHICLES“ genommen werden und es müssen nur noch jene Attribute, die nur auf der abgeleiteten Klasse existieren, falls nötig, eingearbeitet werden.

#### 5.4.4 Tabellenansicht

In Tabellen (Abbildung 46) können mehrere Instanzen auch von unterschiedlichen Objektdefinitionen dargestellt werden.

Symbol	Farbe	Model	Sitzplätze
	rot	Audi	10
	---	---	---
	---	---	---

Abbildung 46: Beispiel Tabellenansicht

Einstellbar sind darüber hinaus welche Attribute der jeweiligen Objektdefinition angezeigt werden sollen. Auch Funktionen, die aus Microsoft Excel bekannt sind, können genutzt werden, wie:

- Gruppierungen
- Filter
- Sortierungen
- Suchen & Ersetzen
- Kopieren, Ausschneiden & Einfügen

#### 5.4.5 Kalenderdialog

Um Objekte welche Datumsfelder enthalten anschaulich darstellen zu können, gibt es den Kalenderdialog (Abbildung 47). Er besteht aus einem individuell anpassbaren Zeitintervall indem jeweils Tag, Woche, Monat dargestellt werden, ähnlich wie es aus Microsoft Outlook bekannt sein sollte. Eine Navigationsbar auf der Seite, dient dazu um zwischen Monate und Jahre navigieren zu können.

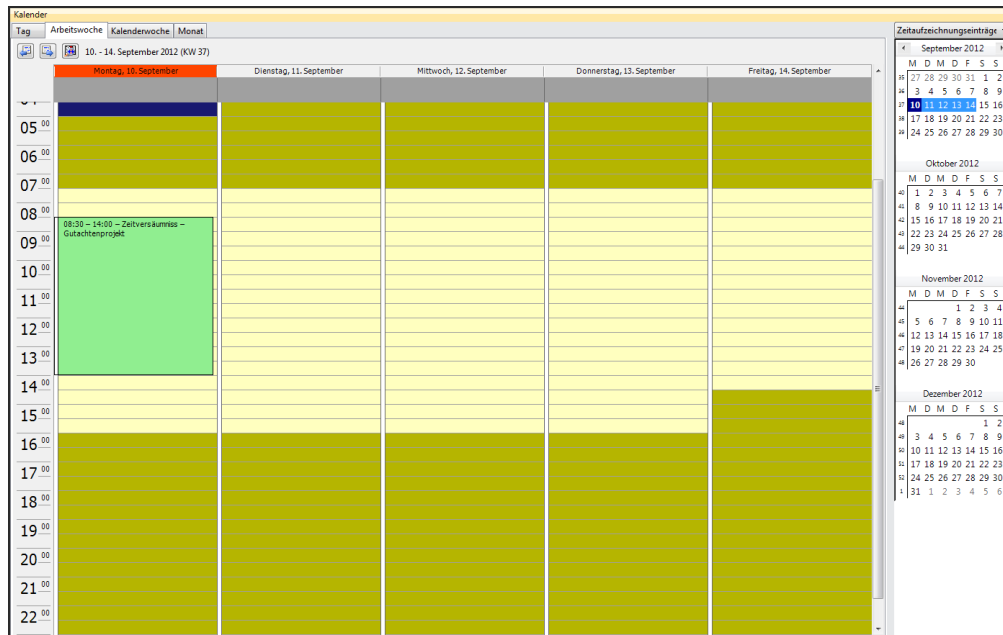


Abbildung 47: Beispiel Kalendertdialog

Über eine vordefinierte Abfrage werden genau die Informationen (Termine, Zeitaufzeichnungseinträge, etc.) im Kalender angezeigt, die von Bedeutung sind. Auch hier kann ansichtsbezogen definiert werden, welche Attribute für die angezeigte Klassendefinitionen wichtig sind.

#### 5.4.6 Ansichten

All diese Controls, welche vorgestellt wurden, können in einer Ansicht (Abbildung 48) frei wählbar zusammengesetzt werden. Dabei ist es möglich, jegliche Art von Funktionalität, die aus bekannten Designergeneratoren aus Programmiersprachen bekannt sind, auch in einer Ansicht abzuspeichern. Dazu zählen:

- Reiter von Dialogen
- Dialoge fixieren, anheften
- Dialoge aus- & einrollen, Vollbild
- Konfiguration der Größe & Position

Im nächsten Bild sieht man eine Ansicht mit drei Dialogen:

- Baumansicht (links)
- Tabellenansicht (rechts oben)
- Formularansicht (rechts unten)

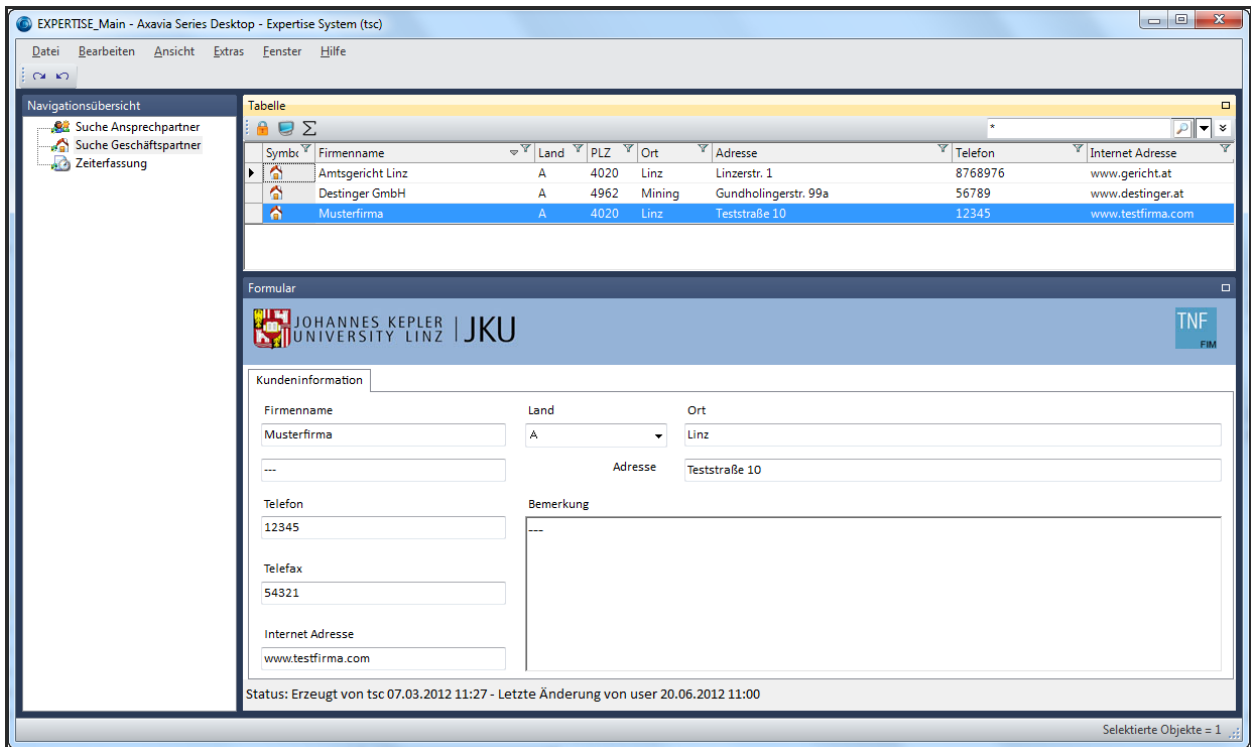


Abbildung 48: Beispiel einer Ansicht aus Dialogen

Auch die Kommunikation in der Ansicht wird gesteuert, so können Dialoge sowohl Daten empfangen, als auch Daten an andere Dialoge verschicken.

		an								
		Eigenschaften	Baumansicht	Tabellenansicht	Modellansicht	Grafikansicht	Suchtabelle	Übersicht	Formularansicht	Berechtigungen
von	Eigenschaften	■	■	■	■	■	■	■	■	■
	Baumansicht	➔	■	■	■	■	■	■	■	■
	Tabellenansicht	➔	■	■	■	■	■	■	■	■
	Modellansicht	➔	■	■	■	■	■	■	■	■
	Grafikansicht	➔	■	■	■	■	■	■	■	■
	Suchtabelle	■	■	■	■	■	■	■	■	■
	Übersicht	➔	■	■	■	■	■	■	■	■
	Formularansicht	■	■	■	■	■	■	■	■	■
	Berechtigungen	■	■	■	■	■	■	■	■	■

Abbildung 49: Beispiel der Kommunikation zwischen Dialogen

In dieser Matrix (Abbildung 49) ist zu erkennen, wie die einzelnen Dialoge ihre Daten an andere senden, sowohl einzelne Objekt (Baumansicht → Eigenschaften) als auch mehrere Objekte (Baumansicht → Tabellenansicht) sind zu erkennen.

Ein weiteres Charakteristikum ist das Konfigurieren dynamischer Symbolleisten, so ist es möglich genau zu justieren, welche Objekte erstellt werden können, wenn ein spezifische Objekt in einem Dialog markiert ist.

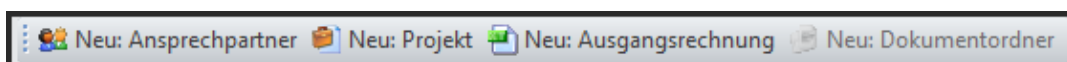


Abbildung 50: Beispiel Symbolleiste

In dieser Symbolleiste (Abbildung 50) sieht man, dass drei von vier Objekterzeugungen möglich sind.

Im Sinne der Benutzerfreundlichkeit ist es möglich in jedem Dialog zu definieren, welche Objektedefinitionen mit welchen Attributen überhaupt sichtbar sind und es kann zwischen drei Arbeitsmodi gewählt werden:

- Anwender
- Experte
- Customizer

Je nachdem welcher Modus genutzt wird, sind mehr oder weniger Kontextmenüeinträge verfügbar bzw. lassen sich dadurch Optionen und Merkmale ein- oder ausschalten. Im nächsten Bild (Abbildung 51) ist das gleiche Menü jeweils im Expertenmodus (links) und Anwendermodus (rechts) zu sehen. Der Modus „Customizer“ dient zum Administrieren und Modellieren, damit erhält man schnellen Zugriff auf die Objekt-, Attribut-, Verknüpfungs- und Methodendefinition.

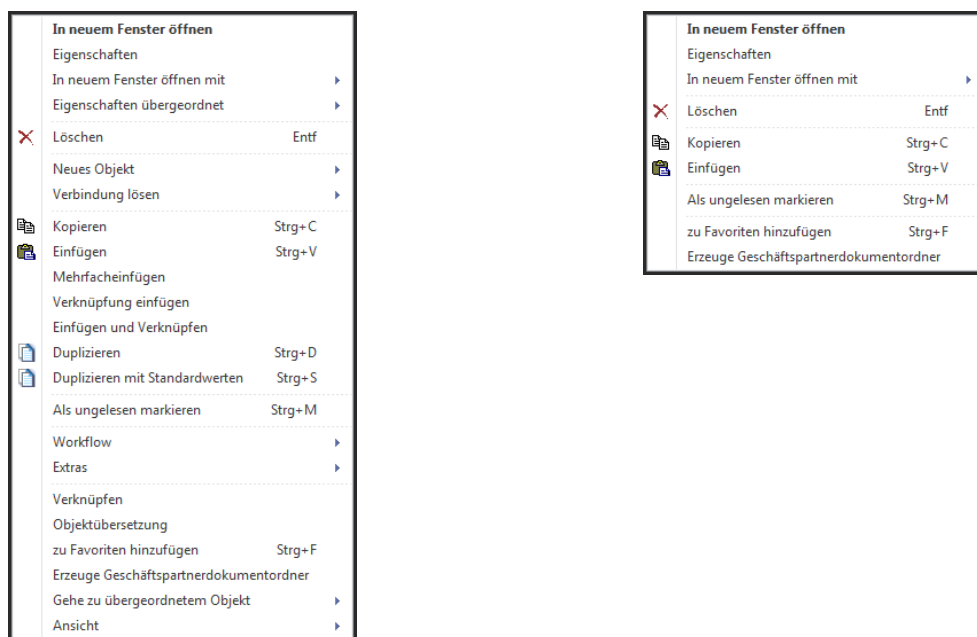


Abbildung 51: Beispiel Kontextmenüeinträge Anwender und Experte

Dadurch entsteht ein GUI, welche je nach Einsatzgebiet angepasst werden kann. Zu erwähnen sei noch, dass genau diese frei konfigurierbare Ansichtskonfiguration nicht fest in der Software programmiert ist, sondern auch diese als Objekte mit Attributen in der Datenbank gespeichert sind. Dadurch wird jeweils zur Laufzeit die Benutzeroberfläche aufgebaut und kann durch Administratoren verändert und angepasst werden.

Eine andere Art zur Sichtbarkeitssteuerung ist die Rechteverwaltung, auf welche im nächsten Kapitel näher eingegangen wird.

## 5.5 Berechtigungsverwaltung

Als Berechtigungssystem kommt eine modifizierte Version der Rollenbasierten Zugriffskontrolle (=RBAC) zum Einsatz. Unter Berechtigung sei eine Aussage verstanden, dass einer Rolle zu einem Objekt eine gewisse Anzahl von Rechten zuspricht. In AXAVIASeries kann ein Objekt sich in unterschiedlichen Zuständen befinden, wie „In Arbeit“ oder „Archiviert“ und je nach Zustand sind mehr oder weniger Operation auf dieses Objekt zulässig, siehe Abbildung 52.

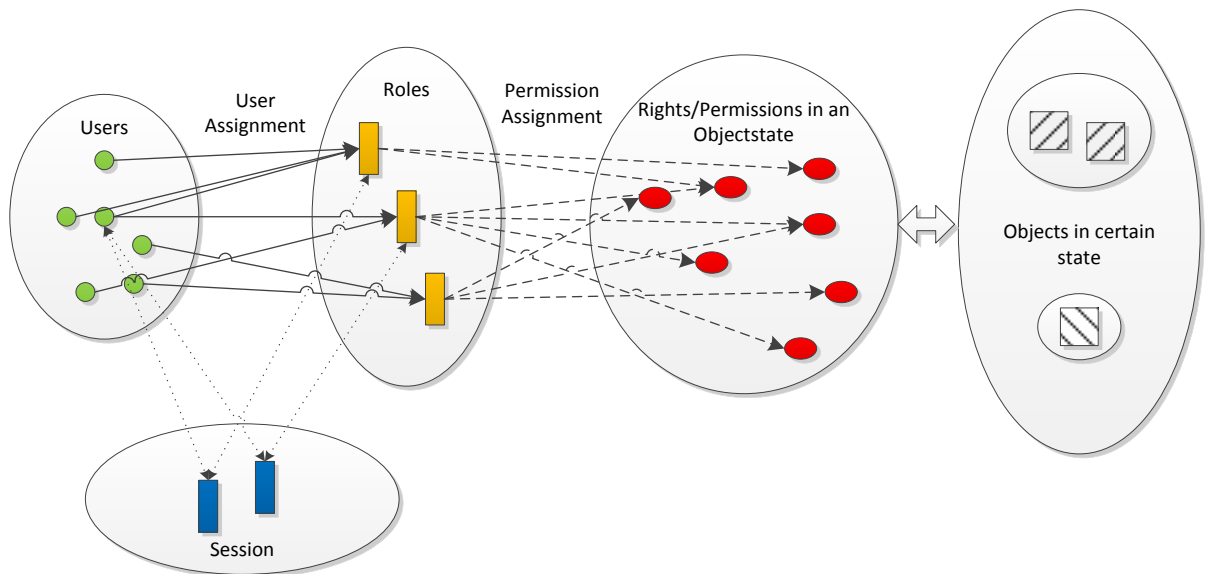


Abbildung 52: Role Based Access Control mit Objektstatus

Einzelne Benutzer bekommen Rollen im System zugewiesen, dabei ist es möglich dass der gleiche Benutzer mehrere Rollen erhält. Den Rollen wiederum werden Rechte eingeräumt. Zu einem bestimmten Zeitpunkt im System (=Session) ist eine Operation genau dann auf ein Objekt erlaubt, wenn eine Rolle zum Zustand des Objektes das Recht dazu hält. Es wurde also der Zustand (Workflow-State) eines Objektes in das RBAC System mit integriert.

Eine weitere spezielle Funktionalität ist die Verfügbarkeit von Objektmengen. Es ist möglich Objekte in der Datenbank einem Bereich (Cluster) zuzuordnen. Beispiele dazu wären alle Objekte im Vorlagebereich, Bibliotheksbereich, etc. aber auch alle Objekte welche sich in einem Projekt also dem Projektbereich befinden. Zu diesen Objektmengen gibt es Bereichsrechte.

Die Auswertung, ob eine Operation auf ein Objekt zulässig ist, wird in einem System welches zwischen Klassen, Instanzen, Vererbung, Beziehungen, etc. unterscheidet nicht trivial. Die nächsten Unterkapitel sollen einen Einblick geben wie dieses abgewandelte RBAC in AXAVIASeries umgesetzt wurde.

### 5.5.1 Begriffserklärung

Bevor mit der Erklärung der Rechte im System begonnen werden kann ist es notwendig einige Begriffe zu erklären.

#### Bereich (Cluster)

AXAVIASeries arbeitet im Hintergrund mit einer Datenbank und stellt diese Daten über Ansichten dar. Aus Gründen der Geschwindigkeit wurde das Konzept von Bereichen eingeführt. Folgende Bereiche existieren, siehe Abbildung 53. Dabei ist es möglich Berechtigungen auf einen gesamten Bereich zu setzen, alle in diesem Bereich enthaltenen Objekte erben diese. In dieser Arbeit wurde der Rolle „Benutzer“ das Löschen und Erzeugen neuer Objekte im Bibliotheks- und Vorlagebereich untersagt, nur die Rolle „Gutachtenkonfigurator“ kann diese anpassen.

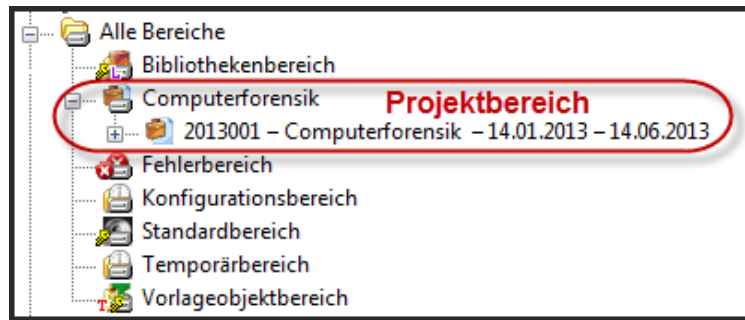


Abbildung 53: Bereiche in AXAVIASeries

Objekte, also Instanzen von Objektedefinitionen werden je nachdem wo diese erzeugt wurden einem Bereich zugeordnet. Auch auf Datenbankebene werden diese Objekte über ein spezielles Attribut markiert, damit kann ein Index zu den jeweiligen Bereichen automatisch erstellt werden.

Sucht man Objekte in einem Projektbereich, kann über diesen Index zugegriffen werden und es müssen nicht alle Objekte dieser Klasse in der gesamten Datenbank überprüft werden.

Als Beispiel für diese Arbeit sind alle Daten und Dateien, die für ein Gutachten notwendig sind in einem eigenen Projektbereich abgelegt. Stammdaten zu Geschäftspartner, Ansprechpartner, Zeitaufzeichnung und Rechnungslegung sind im Standardbereich. Aus den Vorlagen- und Bibliotheksbereich stammen die Berichte, Rechnungsvorlagen, Checklisten, etc., die dann in Projekte übernommen werden können. Im Konfigurationsbereich sind die Einstellungen der Ansichten zu finden, der Temporärbereich wird genutzt um Daten zur Berichterstellung zusammenzutragen. Falls Objekte nur noch lose, also keine Verknüpfungen zu anderen Objekten mehr haben, werden diese im Fehlerbereich gesammelt.

### Bereichsübergreifende Verknüpfung

Man spricht von einer bereichsübergreifenden Verknüpfung, wenn Objekte, welche sich in unterschiedlichen Bereichen befinden, eine Verknüpfung zueinander besitzen. Im Gegensatz dazu haben Objekte im selben Bereich auch deren Verknüpfung in diesem.

Als Beispiel siehe Abbildung 54, für eine bereichsübergreifende Verknüpfung (3) dient uns die Rechnungsposition „Reisekosten“ (2) aus dem Standardbereich, die über die Auswahl einer Produktnummer mit einem Artikel (1) aus dem Bibliotheksbereich verknüpft wurde. Die Verknüpfung dient in diesem Fall um Rechnungstext, Preis, Umsatzsteuer, etc. auf die Rechnungsposition zu referenzieren.

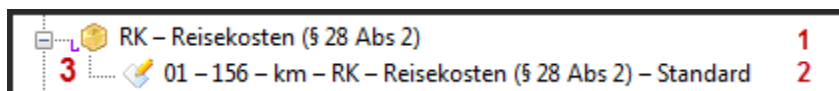


Abbildung 54: Bereichsübergreifende Verknüpfung

Mit Hilfe einer Einstellung auf der Verknüpfungsdefinition wird gesteuert ob sich Objekte, welche sich in unterschiedlichen Bereichen befinden überhaupt mit einander verknüpft werden können. Die Schwierigkeit liegt dabei in der internen Abwicklung der Daten auf Datenbankebene, da sich Rechte und Suchindizes diesem Verhalten anpassen müssen.

Wie die Auswertung der Rechte dazu definiert ist, kann im Kapitel 5.5.10 und 5.5.11 entnommen werden.



## Benachbarte Objekte

Man spricht von „benachbarten Objekten“, wenn auf der Verknüpfungsdefinition das Attribut „Berechtigung vererben“ gesetzt wurde. Folgende Einstellungen sind möglich:

- Nach unten vererben (Superobjekt ist zum Subobjekt benachbart)
- Nach oben vererben (Subobjekt ist zum Superobjekt benachbart)
- Keine Vererbung (kein Objekt ist zum Anderen benachbart)

Als Beispiel wurde im Expertise Management System die Einstellung „Nach unten vererben“ für alle Verknüpfungen vorgenommen, in dem Dokumente oder Dokumentordner zu einem anderen Dokumentordner verknüpft sind. Dies bedeutet, dass nur der Wurzelordner mit Zugriffsrechten versehen werden muss, siehe Abbildung 55.

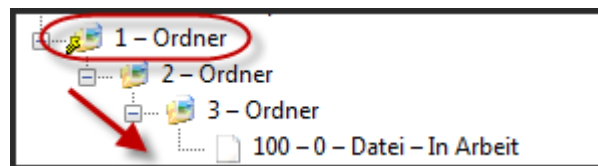


Abbildung 55: Ordnerberechtigungen

## Standardeinstellung am Recht

Es gibt unterschiedliche Rechte im System (Abbildung 59), um zu verstehen was die „Standardeinstellung am Recht“ heißt, wird das Recht „Instanz lesen“ im Detail betrachtet, siehe Abbildung 56.

Bezeichnung	DE	Instanz: Lesen
Für Objektdefinitionen		ja
Für Wertdefinitionen		ja
Für Objekte		ja
Für Werte		ja
Für Verbindungsdefinitionen		ja
Für Verbindungen		ja
Für Bereichsobjekte		nein
Für Bereichsobjektdefinitionen		nein
Für Nachbarobjekte		nein

Abbildung 56: Standardeinstellung am Recht: Instanz lesen

Diese Standardeinstellung gilt dann, wenn keine anderen Berechtigungen auf den jeweiligen Objekten gesetzt sind. Standardmäßig ist es jedem erlaubt alle Objektdefinitionen, Wertdefinitionen, Verbindungsdefinitionen, Objekte, Werte, Verbindungen zu lesen, nur für Bereichsobjektdefinitionen, Bereichsobjekte und Nachbarobjekte ist es nicht erlaubt.

Wie bereits erwähnt, hat nur die Rolle „Gutachtenkonfigurator“ Zugriff auf die Ansicht „EXPERTISE\_Configuration“. Es wurde also direkt auf dem Objekt der Ansichtskonfiguration die Rolle „Benutzer“ ausgesperrt.

Berechtigungen		
Berechtigungen	Besitzer/Objektabhängige Rollenzuteilung	Effektive Berechtigungen
Berechtigungen von "EXPERTISE_Configuration"		
Rolle/Benutzer	Objektstatus	
Rolle - Benutzer	---	
Recht	Standardberechtigung	Berechtigung
Instanz: Lesen	✓	✗
Instanz: Schreiben	✓	✗
Instanz: Löschen	✓	✗

Abbildung 57: Berechtigung auf Ansicht

In Abbildung 57 sieht man für die Rolle „Benutzer“ die Standardberechtigung und in der Spalte rechts daneben, wie diese Rechte verweigert werden.

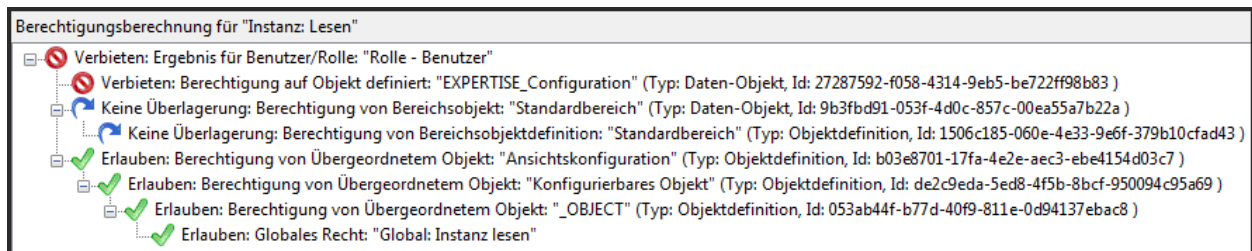


Abbildung 58: Berechtigungsauswertung auf Ansicht

Dazu wird noch der Auswertungsbaum in Abbildung 58 für „Instanz lesen“ abgebildet. Zu sehen sind die drei maßgeblich verantwortlichen Teilbäume, die zuständig sind ob ein Recht „Instanz lesen“, in diesem Fall einem Ansichtskonfigurations-Objekt“, den Zugriff erlaubt oder verbietet.

Die Auswertungsbaume werden noch ausführlicher in den weiterführenden Kapiteln beschrieben.

### 5.5.2 Allgemeine Einstellungen in AXAVIASeries

Die Standardeinstellung der Rechte und Rollen findet man in der Benutzerverwaltung.

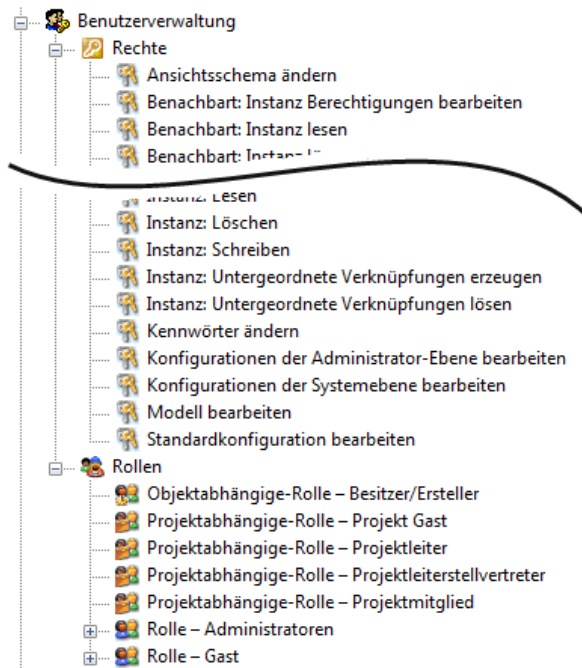


Abbildung 59: Rechte und Rollen in AXAVIAseries

Wie im Bild (Abbildung 59) zu sehen ist, gibt es eine Vielzahl von Rechten im System, zu diesem wird eine Standardeinstellung abgegeben. Dazu wird ein Singleton-Objekt (= ein Objekt das nur einmal in der Datenbank vorkommt) als Objekt gewählt, wo genau diese Rechte für die jeweiligen Rollen veränderbar sind.

Des Weiteren ersichtlich sind Rollen im System, einige davon sind statisch, andere dynamisch. Statische Rollen verändern sich in einer Sitzung nicht mehr, wobei dynamische Rollen zur Laufzeit änderbar sind. Als Beispiel könnte ein Benutzer Leserechte in einem Projekt erhalten. Da sich alle Objekte, die sich in einem Projekt befinden zu einem Bereich gehören, genügt es das Bereichsrecht dieser Projektabhängigen Rolle auf „Lesen“ zu setzen und Benutzer, die dieser Rolle zugeordnet sind erhalten das Recht dazu.

Für jedes Recht wird ein Standardwert eingestellt und ob es auf diesem Singleton-Objekt veränderbar ist. Im nächsten Bild (Abbildung 60) ist ersichtlich, wie auf diesem Objekt für die Rolle „Administrator“ die Standardeinstellung für „Modell bearbeiten“ von „Verboten“ auf „Erlaubt“ verändert wurde.

Berechtigungen		
Berechtigungen	Besitzer/Objektabhängige Rollenzuteilung	Effektive Berechtigungen
Berechtigungen von "Expertise System"		
Rolle/Benutzer	Objektstatus	
Rolle - Administrator	---	
Rolle - Anwalt Beklagter	---	
Rolle - Anwalt Kläger	---	
Rolle - Auftraggeber	---	
Rolle - Beklagter	---	
Rolle - Benutzer	---	
Rolle - Besitzer/Ersteller	---	
Rolle - Gutachtenkonfigurator	---	
Rolle - Kläger	---	
Rolle - Projektmitglied	---	
Recht	Standardberechtigung	Berechtigung
Modell bearbeiten	⊘	✓
Berechtigungseditor anzeigen	⊘	✓
Grundmodelle aktualisieren	⊘	✓
Kennwörter ändern	⊘	✓
Offlinedatenbanken erzeugen	⊘	✓
Alle Offlineregistrierungen bearbeiten	⊘	✓
Instanz: Lesen	✓	↻
Instanz: Schreiben	✓	↻
Instanz: Löschen	✓	⊘
Instanz: Berechtigungen bearbeiten	✓	↻

Abbildung 60: Berechtigungen für Rolle Administrator

Im nächsten Bild (Abbildung 61) sieht man, gegenübergestellt die Rolle „Administrator“ und „Benutzer“.

Berechtigung: Instanz lesen	Erlauben
Berechtigung: Instanz erzeugen	Erlauben
Berechtigung: Instanz schreiben	Erlauben
Berechtigung: Instanz löschen	Erlauben
Berechtigung: Instanz Berechtigungen bearbeiten	Erlauben
Berechtigung: Untergeordnete Verknüpfungen erzeugen	(Erlauben) keine Überladung
Berechtigung: Untergeordnete Verknüpfungen lösen	(Erlauben) keine Überladung
Berechtigung: Definition erzeugen	(Erlauben) keine Überladung
Berechtigung: Definition schreiben	(Erlauben) keine Überladung
Berechtigung: Definition löschen	(Erlauben) keine Überladung
Berechtigung: Formular designen	Erlauben
Berechtigung: Grafik designen	Erlauben
Berechtigung: Modell bearbeiten	Erlauben
Berechtigung: Grundmodelle aktualisieren	Erlauben
Berechtigung: Berechtigungseditor anzeigen	Erlauben
Berechtigung: Kennwörter ändern	Erlauben
Berechtigung: Ansichtsschema ändern	Erlauben
Berechtigung: Konfiguration der Systemebene bearbeiten	Erlauben
Berechtigung: Konfiguration der Administrator-Ebene bearbeiten	Erlauben
Berechtigung: Standardkonfiguration bearbeiten	Erlauben
Berechtigung: Offlinedatenbank erzeugen	Erlauben
Berechtigung: Alle Offlineregistrierungen bearbeiten	Erlauben

Berechtigung: Instanz lesen	Erlauben
Berechtigung: Instanz erzeugen	Erlauben
Berechtigung: Instanz schreiben	Erlauben
Berechtigung: Instanz löschen	Erlauben
Berechtigung: Instanz Berechtigungen bearbeiten	(Verboten) keine Überladung
Berechtigung: Untergeordnete Verknüpfungen erzeugen	(Erlauben) keine Überladung
Berechtigung: Untergeordnete Verknüpfungen lösen	(Erlauben) keine Überladung
Berechtigung: Definition erzeugen	(Verboten) keine Überladung
Berechtigung: Definition schreiben	(Verboten) keine Überladung
Berechtigung: Definition löschen	(Verboten) keine Überladung
Berechtigung: Formular designen	(Verboten) keine Überladung
Berechtigung: Grafik designen	(Verboten) keine Überladung
Berechtigung: Modell bearbeiten	(Verboten) keine Überladung
Berechtigung: Grundmodelle aktualisieren	(Verboten) keine Überladung
Berechtigung: Berechtigungseditor anzeigen	(Verboten) keine Überladung
Berechtigung: Kennwörter ändern	(Verboten) keine Überladung
Berechtigung: Ansichtsschema ändern	(Verboten) keine Überladung
Berechtigung: Konfiguration der Systemebene bearbeiten	(Verboten) keine Überladung
Berechtigung: Konfiguration der Administrator-Ebene bearbeiten	(Verboten) keine Überladung
Berechtigung: Standardkonfiguration bearbeiten	(Verboten) keine Überladung
Berechtigung: Offlinedatenbank erzeugen	(Verboten) keine Überladung
Berechtigung: Alle Offlineregistrierungen bearbeiten	(Verboten) keine Überladung

Abbildung 61: Rolle Administrator und Benutzer Gegenüberstellung

### 5.5.3 Rechte im System

Folgende Rechte sind im AXAVIASeries System bereits für alle enthaltenen Rollen vordefiniert.

- Instanz lesen / schreiben / löschen / Berechtigung der Instanz bearbeiten
  - Global, Lokal, Bereich, Benachbarte Objekte
- Untergeordnete Verknüpfungen erzeugen / löschen
  - Global, Lokal, Bereich
- Definition erzeugen / schreiben / löschen
  - Global
- Formulare erarbeiten
- Grafiken erarbeiten

- Modell bearbeiten
- Grundmodell aktualisieren
- Berechtigungseditor anzeigen
- Kennwörter ändern
- Ansichtsschema ändern
- Standardkonfiguration bearbeiten

#### 5.5.4 Auswertung der Rechte

Aus Performancegründen sind Berechtigungen welche am Objekt selbst definiert werden auch dort in binärer Form abgelegt, da dann genau diese Einstellung zu tragen kommt, siehe dazu 5.5.7.

Die Auswertung der Rechte innerhalb einer Rolle und zwischen den Rollen hängt nicht mehr ausschließlich von der Standardeinstellung am Recht selbst ab. Fließen für ein Objekt von mehreren Teilbäumen (siehe Abbildung 63) die Berechtigungseinstellungen ein, so wird in der gemeinsamen Rolle die resultierende Berechtigung „minimiert“.

Dies bedeutet dass bei nur einem vererbten „Verbot“ die ergebende Berechtigung auch ein „Verbot“ ist. Für die Auswertung eines konkreten Benutzers reicht es, wenn sich dieser in mindestens einer Rolle befindet, die ein „Erlauben“ enthält. Nur wenn für alle Rollen in denen sich ein Benutzer befindet, weder ein „Verbot“ noch „Erlauben“ also ein „keine Überlagerung“ evaluiert wird, wird die Standardeinstellung vom Recht verwendet.

Berechtigungsrechnung für "Instanz: Löschen"

- ✓ Erlauben: Ergebnis für Benutzer/Rolle: "Thomas Schneider" **4**
- ⊕ Keine Überlagerung: Ergebnis für Benutzer/Rolle: "Thomas Schneider"
- ⊕ Erlauben: Ergebnis für Benutzer/Rolle: "Rolle - Administrator"
  - ⊕ Keine Überlagerung: Berechtigung von Bereichsobjekt: "Projektbereich" (Typ: Daten-Objekt, Id: ba54203f-f117-4e57-a396-b2a66de5ead6) **3**
  - ✓ Erlauben: Berechtigung von Übergeordnetem Objekt: "Projekt" (Typ: Objektdefinition, Id: 457fbb8c-4194-4d1f-84a6-aa5a4271d64e)
- ⊕ Erlauben: Ergebnis für Benutzer/Rolle: "Rolle - Benutzer"
  - ⊕ Keine Überlagerung: Berechtigung von Bereichsobjekt: "Projektbereich" (Typ: Daten-Objekt, Id: ba54203f-f117-4e57-a396-b2a66de5ead6) **2**
  - ✓ Erlauben: Berechtigung von Übergeordnetem Objekt: "Projekt" (Typ: Objektdefinition, Id: 457fbb8c-4194-4d1f-84a6-aa5a4271d64e)
- ⊕ Verboten: Ergebnis für Benutzer/Rolle: "Rolle - Gutachtenkonfigurator"
  - ⊖ Verboten: Berechtigung auf Objekt definiert: "Computerforensik" (Typ: Daten-Objekt, Id: d5028e3c-ed2b-4d20-9f4b-c242f515f1dd) **1**
  - ⊕ Keine Überlagerung: Berechtigung von Bereichsobjekt: "Projektbereich" (Typ: Daten-Objekt, Id: ba54203f-f117-4e57-a396-b2a66de5ead6)
  - ✓ Erlauben: Berechtigung von Übergeordnetem Objekt: "Projekt" (Typ: Objektdefinition, Id: 457fbb8c-4194-4d1f-84a6-aa5a4271d64e)

Abbildung 62: Instanz: Löschen definiert auf einem Projekt

Als Beispiel, siehe Abbildung 62 habe ich der Rolle „Gutachtenkonfigurator“ das Recht „Instanz löschen“ für ein Projekt „Computerforensik“ (siehe 1) genommen. Zu sehen ist das diese Rolle standardmäßig das Recht hätte, die Berechtigung wird aber innerhalb dieser Rolle „minimiert“.

Die Rolle „Benutzer“ (siehe 2) hat das Recht dieses Objekt zu löschen, genauso wie die Rolle „Administrator“ (siehe 3).

Die gesamte Auswertung ob dieser Benutzer „Thomas Schneider“ das Objekt löschen darf oder nicht (siehe 4) wird erlaubt da dieser in mindestens einer Rolle ist, in diesem Fall „Administrator“ und „Benutzer“, welche es erlaubt.

Zusammenfassend wird definiert:

- Recht innerhalb einer Rolle wird „minimiert“
- Ein „Erlauben“ eines Benutzers in mindestens einer Rollen erlaubt ihm die Aktion
- Ansonsten wirkt die Standardeinstellung

### 5.5.5 Auswertung in einer Rolle

Um zu zeigen was „minimiert“ bedeutet definieren wir zwei Begriffe.

#### „=“ gesetzte Berechtigung

Wird bei der Auswertung auf einem Objekt eine gesetzte Berechtigung gefunden, so ist das Ergebnis für den Teilzweig gleich der gesetzten Berechtigung. Es werden keine weiteren Berechtigungen von übergeordneten Objekten berücksichtigt.

#### „min“ minimale Berechtigung

Es werden alle einfließenden Berechtigungen der übergeordneten Objekte ausgewertet. Ist mindestens eine Berechtigung ein „Verbot“, so ist das Ergebnis ein Verbot. Sind alle Berechtigungen ein „keine Überladung“ ist das Ergebnis „keine Überladung“. Bei keinem Verbot und mindestens einem „Erlaubt“ ergibt sich „Erlaubt“.

	„keine Überladung“	„Erlauben“	„Verboten“
„keine Überladung“	„keine Überladung“	„Erlauben“	„Verboten“
„Erlauben“	„Erlauben“	„Erlauben“	„Verboten“
„Verboten“	„Verboten“	„Verboten“	„Verboten“

Die Tabelle zeigt sowohl in der horizontalen als auch in der vertikalen zwei Rechteinstellungen und wie über die „min“-Operationen das Ergebnis folgt.

Alle nun folgenden Auswertungsbäume beschreiben jeweils die Auswertung innerhalb einer Rolle.

Anzumerken sei auch noch das es drei Teilbäume gibt:

- Berechtigungsvererbung auf Objekte
- Berechtigung der Bereichsrechte
- Rechte aus der Modellebene

## 5.5.6 Auswertung für einen Wert

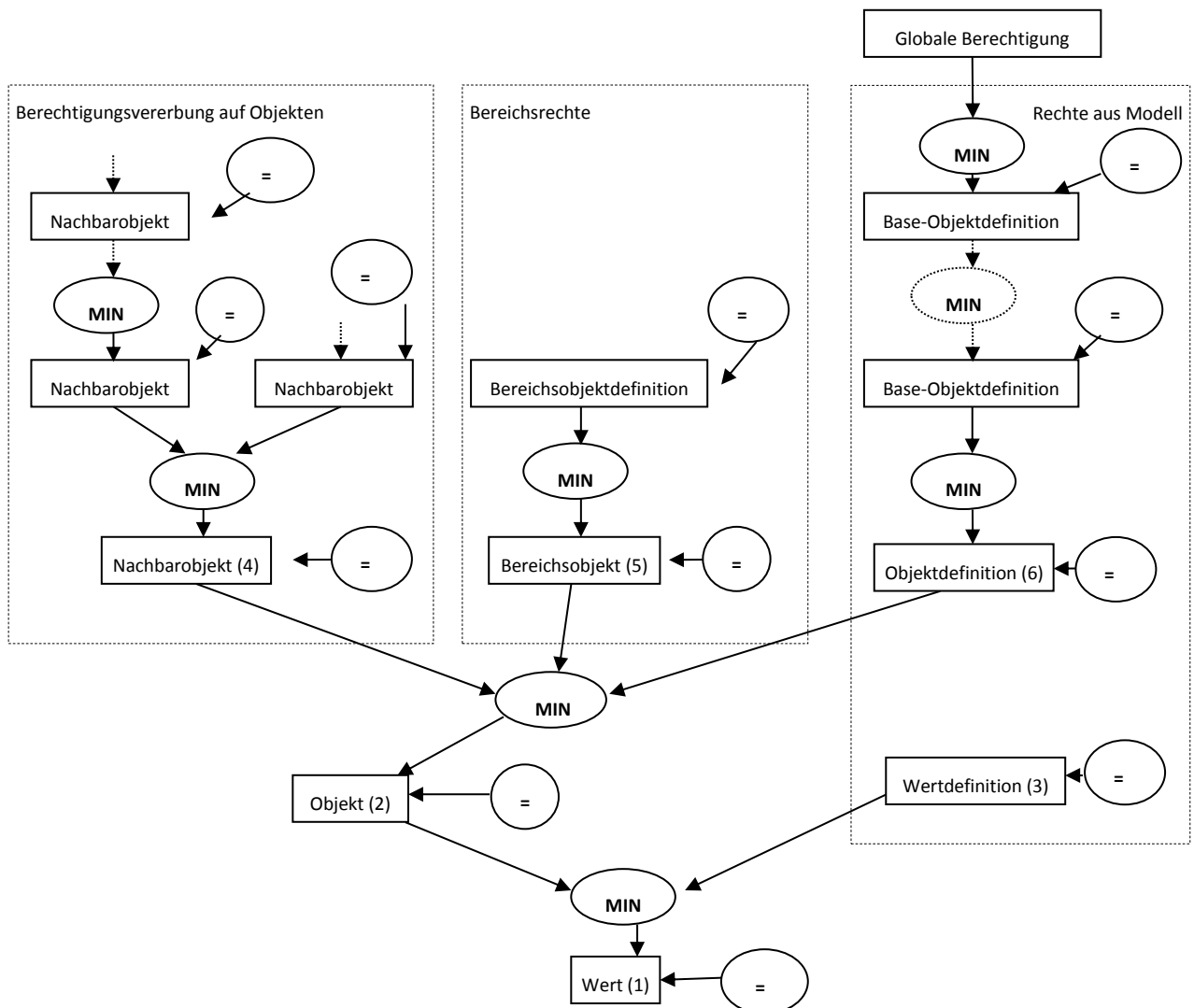


Abbildung 63: Auswertung für einen Wert

Beispiel (Abbildung 63) für das Recht „Instanz lesen“ eines Wertes (Attributes):

- I. Ist für den Wert die Berechtigung gesetzt?
  - a. Ja → Fertig, es gilt der gesetzte Wert, siehe (1)
- II. Berechtigung des zugehörigen „Objekt“, siehe (2) und der „Wertdefinition“, siehe (3) auswerten, beide Ergebnisse mit der „min“ – Operation verknüpfen → Fertig.
- III. Das Ergebnis für die Berechtigung der Wertdefinition, siehe (3) ist die gesetzte Berechtigung darauf.
- IV. Falls am Objekt eine Berechtigung definiert ist, wird diese gesetzte Berechtigung verwendet, siehe (2). Sonst wird eine „min“ – Operation für die Nachbarobjekte, siehe (4), ein eventuell vorhandenes Bereichsobjekt, siehe (5) und der Objektdefinition, siehe (6) ausgeführt.
  - a. Nachbarobjekte, gesetzte Berechtigung bzw. „min“ – Operation vorhandener Nachbarobjekte

- b. Bereichsobjekt, gesetzte Berechtigung bzw. „min“ – Operation der Bereichsobjektdefinition
- c. Objektdefinition, gesetzte Berechtigung bzw. „min“ – Operation der vererbten Berechtigung der übergeordneten Objektdefinitionen bis hin zu der „Globalen“ – Berechtigung.

Dazu ein Beispiel, ähnlich wie in Abbildung 62, wird für den Wert „Basisverzeichnis“ (1), welches sich auf dem Objekte „Expertise System“ (3) befindet, das Recht „Instanz schreiben“ für die Rolle „Benutzer“ entzogen, Abbildung 64 und zwar am Objekt selbst.

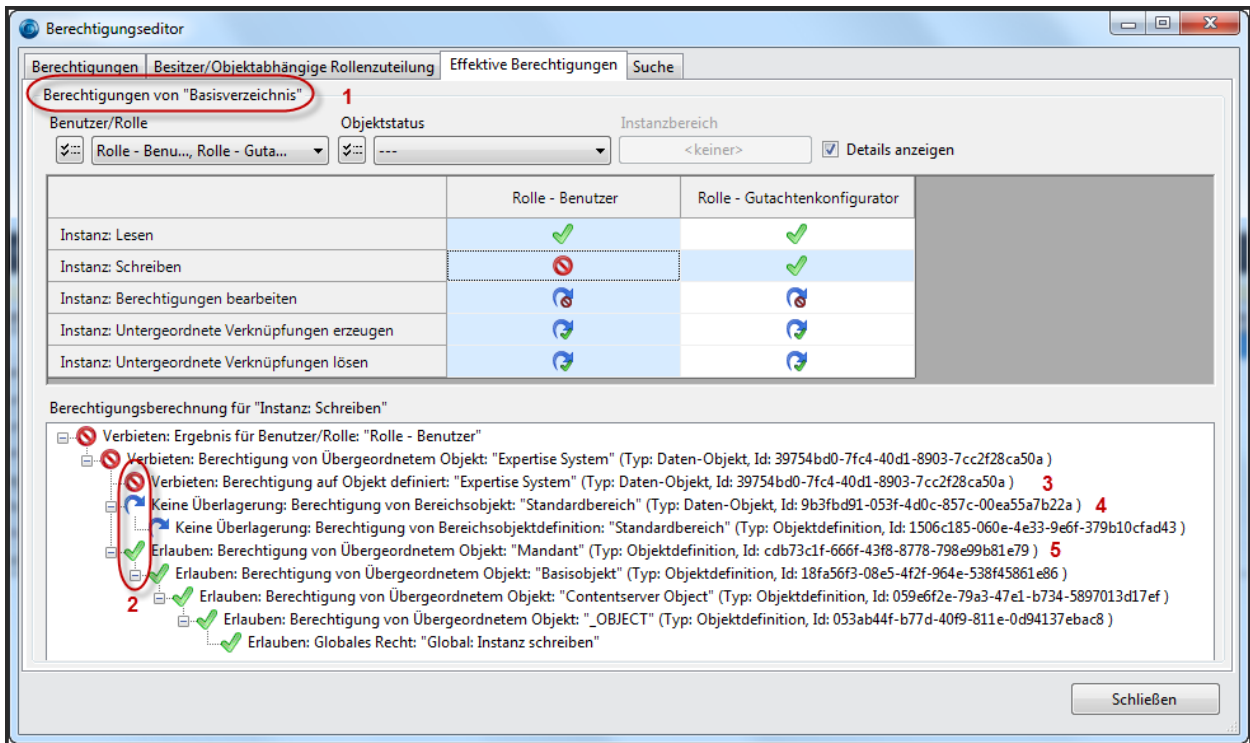


Abbildung 64: Basisverzeichnis, Schreiben verboten auf Objektebene

Zwei der drei Teilbäume sind in diesem Beispiel aktiv, da es keine „benachbarten Objekte“ gibt. In (2) sieht man, dass die Auswertung „minimiert“ (3,4 & 5) wurde und das Verbot bis auf den Wert „Basisverzeichnis“ weiter geleitet wird. Falls die Berechtigung nicht explizit gesetzt wurde zeigen die Symbole und ob die Aktion für die gewählte Rolle standardmäßig erlaubt oder verboten ist.

Zur Veranschaulichung setze ich die Berechtigung im nächsten Bild (Abbildung 65) nun direkt auf dem Wert „Basisverzeichnis“ (nicht auf dem Objekt) auf „Erlauben“ für die Rolle „Benutzer“.

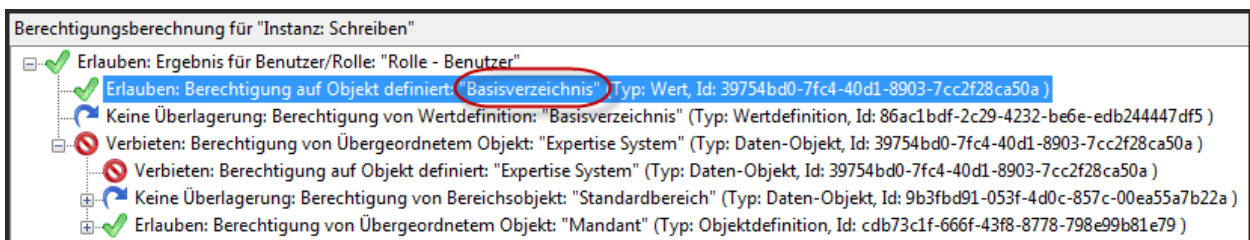


Abbildung 65: Basisverzeichnis, Schreiben erlaubt auf Attributebene



Alle Attribute können durch das Verbiehen des Rechtes „Instanz schreiben“ auf dem Objekte schreibgeschützt werden, nur ein explizites setzen des Rechtes auf einem Werte überschreibt diese.

### 5.5.7 Auswertung für ein Objekt

Der Auswertungsbaum für ein Objekt ist im Auswertungsbaum für Werte enthalten, siehe daher letztes Kapitel.

### 5.5.8 Auswertung für eine Wertdefinition

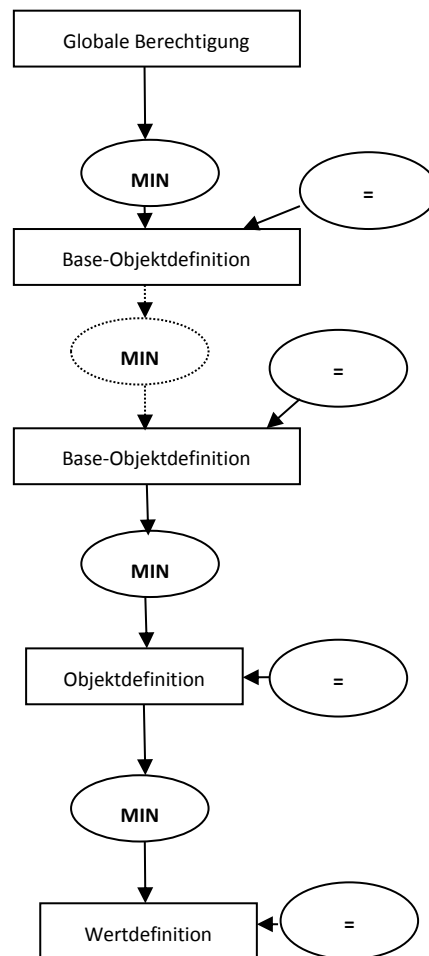


Abbildung 66: Auswertung für Wertdefinition

Da hier immer nur ein übergeordnetes Objekt existiert, reichen die „min“-Operationen die Berechtigungen vom übergeordneten Objekt einfach an das untergeordnete Objekt weiter. Somit ergibt sich die einfache Regel:

Die Berechtigung für ein Modellobjekt ergibt sich immer aus der nächstgelegenen gesetzten Berechtigung bzw. aus der globalen Berechtigung, falls keine Berechtigung gesetzt wurde oder aus der Standardberechtigung vom Recht, wenn auch keine globale Berechtigung definiert wurde, siehe dazu Abbildung 66.

### **5.5.9 Auswertung für eine Objektdefinition**

Der Auswertungsbaum für eine Objektdefinition ist in der Auswertung für eine Wertdefinition enthalten, siehe dazu das letzte Kapitel.

### **5.5.10 Auswertung für Verknüpfungen**

Auf Verknüpfungen können keine direkten Berechtigungen gesetzt werden. Die Berechtigungen für Verknüpfungen ergeben sich aus dem Modell, dem Bereich und der verknüpften Objekte.

Beim Löschen einer Verknüpfung werden die beiden Rechte „Instanz löschen“ und „Untergeordnete Verknüpfungen lösen“ geprüft.

Entsprechend werden für das Anlegen von Verknüpfungen die Rechte „Instanz erstellen“ und „Untergeordnete Verknüpfungen erstellen“ geprüft.

Der Bereich einer Verknüpfung ist nur dann definiert, wenn das übergeordnete und untergeordnete Objekt sich im selben Bereich befinden. d. h.: Bereichsrechte wirken sich nur auf Verknüpfungen aus, die sich zwischen Objekten innerhalb des Bereiches befinden. Auf bereichsübergreifende Verknüpfungen werden daher keine Bereichsrechte wirksam.

Verknüpfungen, welche Rechte von benachbarten Objekten vererben, erben die Berechtigung vom übergeordneten benachbarten Objekt. d. h. für Verknüpfungen die „nach unten“ vererben wird die Berechtigung vom übergeordneten Objekt und für „nach oben“ vererbende Verknüpfungen, die Berechtigung vom unterordneten Objekt berücksichtigt.

Die Berechtigungen für „Untergeordnete Verknüpfungen lösen/erstellen“ müssen jeweils am übergeordneten Objekt vorhanden sein. Für diese beiden Rechte existieren kein globales und kein Standardrecht. d. h. es kann effektiv nur ein Verbot über die beiden Rechte gesetzt werden, da ein „Erlauben“ auch aus dem Nichtvorhanden sein der Berechtigung folgt.

Evaluierungsbaum (Abbildung 67) für „Instanz löschen“ einer Verknüpfung:

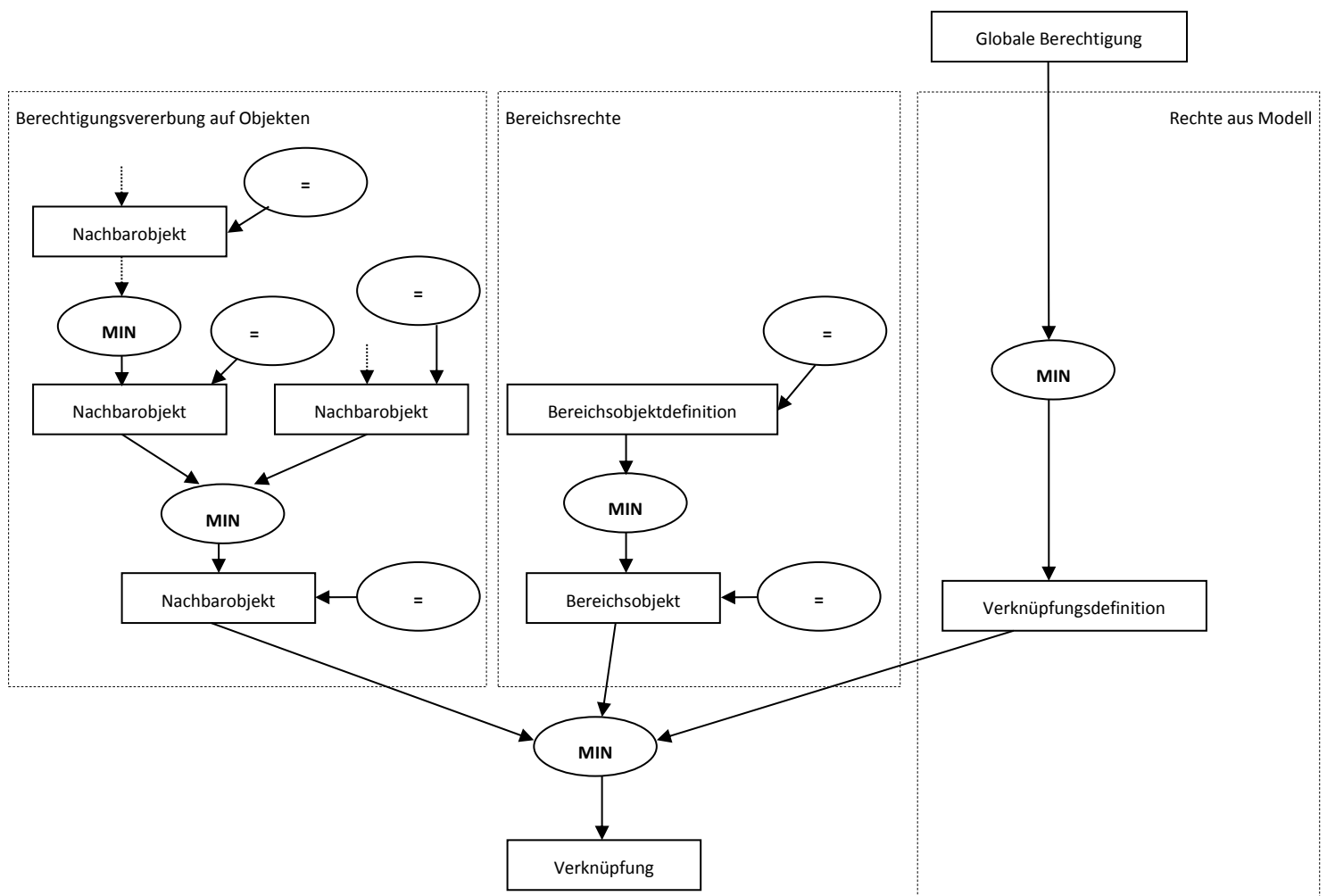


Abbildung 67: Auswertung für Verknüpfungen

### 5.5.11 Auswertung für Verknüpfungsdefinition

Beispiel (Abbildung 68): Definition lesen / schreiben / löschen

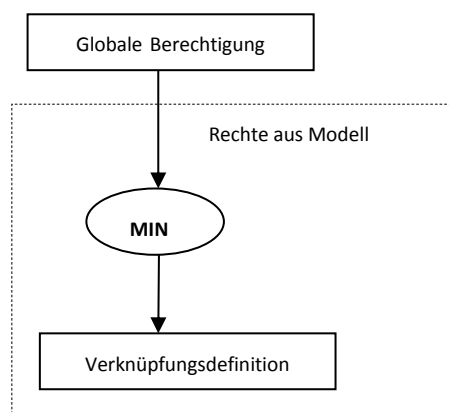


Abbildung 68: Auswertung für Verknüpfungsdefinition

### 5.5.12 Effektive Berechtigung für einen Benutzer

Für die effektiven Berechtigungen für einen Benutzer werden alle Rollen, in denen sich ein Benutzer befindet, ausgewertet. Befindet sich ein Benutzer in mindestens einer

Rolle, die ein „Erlauben“ erhält, folgt für den Benutzer ein „Erlauben“, dazu siehe Abbildung 69.

Ausnahme:

Auf Instanz-Ebene (Objekte, Werte) können neben den Berechtigungen für Rollen auch Berechtigungen direkt für einzelne Benutzer gesetzt werden. Die Auswertung für den Benutzer selbst erfolgt wie für eine Rolle. Bevor die einzelnen Rollen des Benutzers ausgewertet werden, werden die Berechtigungen für den Benutzer selbst ausgewertet. Ergibt sich dabei eine Berechtigung ungleich „keine Überlagerung“ (d. h: ein „Erlauben“ oder ein „Verboten“) so werden keine weiteren Rollen mehr ausgewertet. Werden somit Rechte direkt für Benutzer definiert, so gelten diese unabhängig von den weiteren Rollen in der sich der Benutzer befindet. Daher kann ein Benutzer explizit über direkte Benutzerberechtigungen ein „Verbot“ erhalten, auch wenn er sich in weiteren Rollen befindet, die zu einem „Erlauben“ evaluieren.

Die effektiven Berechtigungen und das Ergebnis der Auswertungsbaume können über einen Berechtigungseditor für Benutzer und Rollen eingesehen werden.

Benutzer/Rolle	Objektstatus	Instanzbereich
Thomas Schneider	---	<keiner>

Instanz: Berechtigungen bearbeiten	Objektstatus	Instanzbereich
Instanz: Untergeordnete Verknüpfungen erzeugen	✓	
Instanz: Untergeordnete Verknüpfungen lösen	✓	
Bereich: Instanz lesen	✓	
Bereich: Instanz schreiben	✓	
Bereich: Instanz löschen	✓	
Bereich: Instanz erzeugen	✓	
Bereich: Instanz Berechtigungen bearbeiten	⊘	
Bereich: Untergeordnete Verknüpfungen erzeugen	✓	

Berechtigungsbaum für "Bereich: Instanz löschen"

- ✓ Erlauben: Ergebnis für Benutzer/Rolle: "Thomas Schneider"
  - ⊘ Keine Überlagerung: Ergebnis für Benutzer/Rolle: "Thomas Schneider"
  - ✓ Erlauben: Ergebnis für Benutzer/Rolle: "Rolle - Administrator"
    - ✓ Erlauben: Berechtigung von Bereichsobjekt: "Vorlageobjektbereich" (Typ: Daten-Objekt, Id: ce9f1872-dd61-402e-b579-c798460544a1)
    - ✓ Erlauben: Berechtigung auf Objekt definiert: "Vorlageobjektbereich" (Typ: Daten-Objekt, Id: ce9f1872-dd61-402e-b579-c798460544a1)
    - ⊘ Keine Überlagerung: Berechtigung von Bereichsobjektdefinition: "Vorlageobjektbereich" (Typ: Objektdefinition, Id: 58ac7403-1ce6-438a-8a61-4de6dd1b7196)
  - ⊘ Keine Überlagerung: Ergebnis für Benutzer/Rolle: "Rolle - Gutachtenkonfigurator"
  - ⊘ Verboten: Ergebnis für Benutzer/Rolle: "Rolle - Benutzer"
    - ⊘ Verboten: Berechtigung von Bereichsobjekt: "Vorlageobjektbereich" (Typ: Daten-Objekt, Id: ce9f1872-dd61-402e-b579-c798460544a1)
    - ⊘ Verboten: Berechtigung auf Objekt definiert: "Vorlageobjektbereich" (Typ: Daten-Objekt, Id: ce9f1872-dd61-402e-b579-c798460544a1)
    - ⊘ Keine Überlagerung: Berechtigung von Bereichsobjektdefinition: "Vorlageobjektbereich" (Typ: Objektdefinition, Id: 58ac7403-1ce6-438a-8a61-4de6dd1b7196)

Abbildung 69: Effektive Berechtigung für einen Benutzer

Im Bild zu sehen ist wie das Recht „Bereich: Instanz löschen“ auf dem „Vorlagebereich“ ausgewertet wird.

Der Benutzer „Thomas Schneider“ befindet sich sowohl in der Rolle „Administrator“ als auch „Benutzer“.

- Der Benutzer selbst hat „keine Überlagerung“ als Ergebnis.
- Rolle „Administrator“ hat als Ergebnis „Erlauben“, das über die „Berechtigung am Objekt“ definiert wurde.
- Rolle „Gutachtenkonfigurator“ hat „keine Überlagerung“ definiert.
- Rolle „Benutzer“ verbietet das Löschen, da es am Objekt selbst definiert ist.

Für das gesamte Ergebnis ergibt sich ein „Erlauben“.

### 5.5.13 Benutzer-, Objekt- und Bereichsabhängige Rollen

Für Besitzer/Ersteller-Berechtigungen und für projektabhängige Berechtigungen wurde das Konzept von Objekt- und Bereichsabhängigen Rollen implementiert.

Die Mitgliedschaft eines Benutzers zu einer Rolle definiert sich wie folgt:

- für Benutzerrollen: statisch über die Zuordnung von Benutzern zu Benutzerrollen (wie bereits vorhanden)
- für objektabhängige Rollen: „dynamisch“ über die Zuordnung von Benutzern zu objektabhängigen Rollen auf den einzelnen Objekten
- für bereichsabhängige Rollen: „dynamisch“ über die Zuordnung von Benutzern zu bereichsabhängigen Rollen auf den einzelnen Bereichsobjekten

Die Zugehörigkeit eines Benutzers zu einer Objekt-/Bereichsabhängigen Rolle ist abhängig von dem Objekt für welches die Berechtigungen geprüft werden. Die Rollenzugehörigkeit wird vor Beginn der Abarbeitung der einzelnen Auswertungsbäume ermittelt. d.h. die Rollenzugehörigkeit bestimmt sich durch das „Start-Objekt“ der Berechtigungsevaluierung und wird während der Auswertung der Berechtigungen von übergeordneten Objekten nicht mehr geändert.

Anders als bei Benutzerrollen können zu „Dynamischen“-Rollen neben Benutzern auch Benutzerrollen als Mitglieder hinzugefügt werden.

Die Berechtigungen für „Dynamische“-Rollen können auf den Objekten wie für Benutzerrollen definiert werden.

#### Objektabhängige Rolle

Die Zugehörigkeit von Benutzern / Benutzerrollen zu objektabhängigen Rollen wird auf den einzelnen Objekten gespeichert. Bearbeitet und eingesehen werden diese über den Berechtigungsesitor (Abbildung 70).

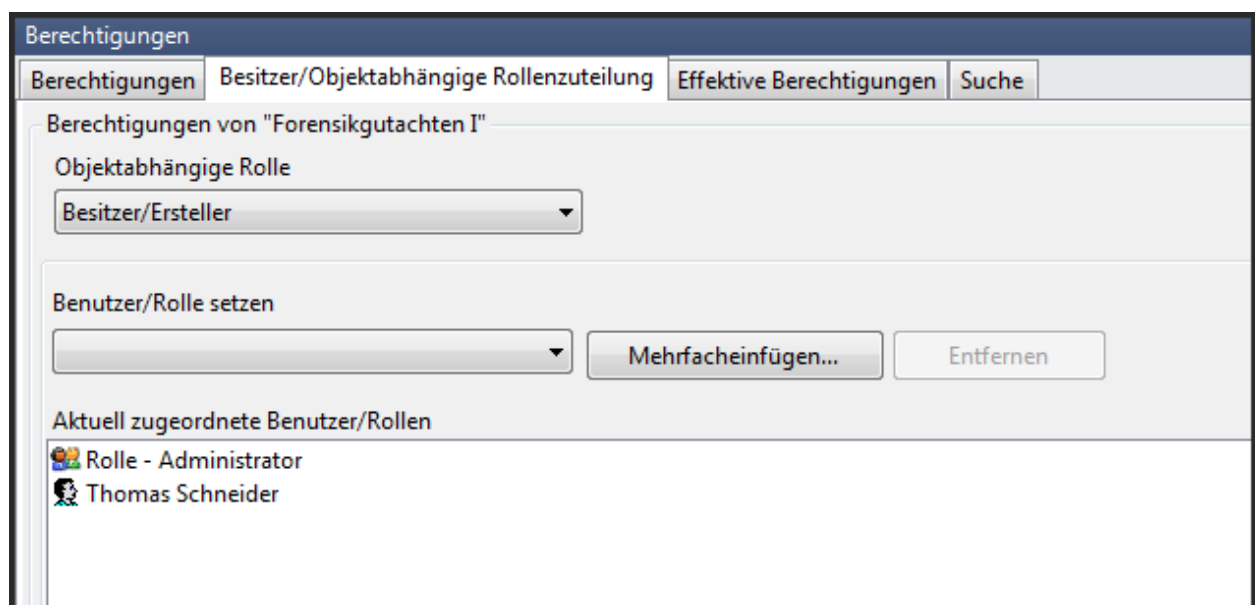


Abbildung 70: Objektabhängige Rollen

## **Objektabhängige Rolle „Besitzer/Ersteller“**

Die objektabhängige Rolle „Ersteller/Besitzer“ verhält sich mit einer Ausnahme wie gewöhnliche objektabhängige Rollen. Das Standardmitglied für neue Objekte ist immer der Ersteller des Objektes.

Dieser Rolle sind standardmäßig alle Manipulationen auf Instanz-Ebene gewährt, d.h. der Ersteller eines Objektes in der Datenbank hat volle Kontrolle darüber.

## **Bereichsabhängige Rollen**

Bereichsabhängige Rollen können wie objektabhängige Rollen über den Berechtigungseditor bearbeitet werden. Eine Zuordnung kann jedoch nur auf Bereichsobjekten bzw. auf deren stellvertretenden Objekten, wie z. B. dem Projekt für Projektbereichsobjekte, erfolgen.

## **Projektabhängige Rollen**

Die vordefinierten projektabhängigen Rollen „Projektleiter“, „Projektleiterstellvertreter“, „Projektmitglied“ und „Projektgast“ sind projektbereichsabhängige Rollen, die nur zu Projektbereichen definiert werden können. Die Zuordnung kann manuell über den Berechtigungseditor erfolgen oder besser über die Projektmitgliedschaft. Bei Änderungen an den Projektmitgliedern werden automatisch die Projektrollenzugehörigkeiten für das Berechtigungssystem gesetzt. Hierfür muss aber der ändernde Benutzer die Berechtigung „Instanz: Berechtigungen bearbeiten“ auf dem Projektbereichsobjekt besitzen. Sind diese Berechtigung nicht vorhanden, dann wird die Eigenschaft „Rolle im Projekt“ als schreibgeschützt dargestellt.

### **5.5.14 Besondere Rechte**

#### **Rechte für benachbarte Objekte**

Die Berechtigungen für die Nachbarrechte „Benachbart: Instanzberechtigungen bearbeiten“, „Benachbart: Instanz lesen“, „Benachbart: Instanz schreiben“ und „Benachbart: Instanz löschen“ können auf Objekten definiert werden. Die benachbarten Objekte eines Objektes sind jene Objekte, die über vererbende Verknüpfungen zum Objekt verknüpft sind. Beispiel: Ist die Verknüpfungsdefinition zwischen Dokumentordnern nach unten vererbend eingestellt, dann sind die untergeordneten Dokumentordner eines Dokumentordners zu diesem benachbart, siehe 5.5.1 Absatz „Benachbarte Objekte“.

Sind keine Berechtigungen zu Nachbarrechten gesetzt, dann vererben sich die gewöhnlichen Instanzrechte wie z. B. „Instanz: Schreiben“. Ist jedoch auf einem Objekt eine Nachbarberechtigung gesetzt, so wird diese als entsprechende Instanzberechtigung an die benachbarten Objekte vererbt.

Beispiel: Auf einem Ordner „Projekte“ ist die Instanzberechtigung „Instanz: schreiben“ auf „erlauben“ gesetzt. Falls keine Nachbarberechtigung „Benachbart: Instanz schreiben“ am Ordner „Projekte“ definiert ist, dann wird „Instanz: schreiben“ mit erlaubt auf die Unterordner vererbt. Ist jedoch auf „Projekte“ die Berechtigung „Benachbart: Instanz schreiben“ auf verboten gesetzt, dann wird für die Unterordner die Berechtigung „Instanz: schreiben“ mit „verboten“ vererbt. Der „Projekte“ Ordner könnte somit geändert werden, die enthaltenen Ordner aber nicht.

## Rechte für Bereiche

Die Berechtigungen für die Bereichsrechte „Bereich: ...“ können auf den Bereichsobjekten bzw. auf deren stellvertretenden Objekten wie dem Projekt und auf Bereichsobjektdefinitionen definiert werden. Wird für ein Objekt z. B. das Instanzrecht „Instanz: schreiben“ geprüft, so wird auf einem eventuell vorhandenem Bereichsobjekt die Berechtigung für „Bereich: Instanz schreiben“ berücksichtigt. Wird auf dem Bereichsobjekt keine entsprechende Bereichsberechtigung gefunden, so wird weiter eine Bereichsberechtigung auf der Bereichsobjektdefinition gesucht. Falls auch dort keine Berechtigungen gefunden werden, wird das Bereichsrecht mit „Erlauben“ angenommen und hat somit keinen Einfluss auf das Ergebnis. Es werden keine globalen Bereichsberechtigungen und Standardberechtigungen vom Recht verwendet.

Beispiel:

Für die Rolle „Benutzer“ wird auf der Objektdefinition „Projektbereich(\_PROJECTCLUSTEROBJECT)“ die Berechtigung für das Recht „Bereich: Instanz schreiben“ auf verboten gesetzt. Damit kann die Rolle „Benutzer“ keine Objekte in Projektbereichen schreiben. Auf einzelnen Projektbereichen kann jedoch die Berechtigung für „Bereich: Instanz schreiben“ auf „erlauben“ gesetzt werden. Auf solchen Projektbereiche könnten „Benutzer“ nun schreibend zugreifen. Normalerweise werden Benutzer jedoch Zugriff auf „ihre“ Projekte durch Projekttrollen erhalten und nicht durch auf Bereiche gesetzte Berechtigungen.

## Rechte für untergeordnete Verknüpfungen

Die Rechte „Untergeordnete Verknüpfungen lösen/erzeugen“ sind gewöhnliche Instanzrechte. Es existieren daher Globale-, Bereichs- und Instanzrechte. Weiter werden diese auf Objekten vererbt. Das System prüft vor der Erstellung bzw. dem Entfernen einer Verknüpfung auf das entsprechende Recht am übergeordnet verknüpften Objekt. Mit diesen Rechten ist es z. B. möglich eine Objektstruktur über Vererbung vor Veränderungen zu schützen, wobei aber die Berechtigungen für die Eigenschaften der Objekte unabhängig eingestellt werden können.

Die Berechtigungen für untergeordnete Verknüpfungen werden, anders wie bei den Nachbarrechten, auch auf nicht vererbende Verknüpfungen angewandt.

### 5.5.15 Zusammenfassung

Da man sowohl Objekte, Werte und Beziehungen (Verknüpfungsdefinitionen) in der Modellebene definieren kann und später in der Datenebene mit Instanzen dieser arbeitet, ist es notwendig sowohl die Berechtigungen von der Modellebene in die Datenebene zu prüfen.

Auch Objektmengen können in AXAVIASeries gebildet werden, ähnlich wie Dokumentordner und deren Dokumente in einem Betriebssystem, kann für die Wurzel dieser Ordnerstruktur eine Berechtigung gesetzt und an den Inhalt vererbt werden.

Auch dass der Status eines Objektes in die Auswertung einfließt, ist gegenüber dem allgemeinen RBAC-Modell neu und muss für dieses System extra behandelt werden. Dies wird beispielweise in der Abarbeitung von Dokumenten verwendet, welche bevor sie beim Kunden landen über das Vier-Augen-Prinzip geprüft werden müssen. Der oder die Bearbeiter des Dokumentes können nach Statuswechsel von „In Arbeit“ auf „Zur Prüfung“ keine Änderungen mehr vornehmen. Da davon ausgegangen wird das nur ein Gutachter im System arbeitet wurde dieser Aspekt nicht weiter behandelt.

Wie die Berechtigungen im Endeffekt ausgewertet werden, ist nicht trivial und benötigt eine gewisse Einarbeitung in das Thema, jedoch können dann komplexe Berechtigungskonzepte in das System eingearbeitet werden.

Im Zusammenhang mit dieser Arbeit wurden zwei Rollen erstellt und deren Berechtigungen wie folgt angepasst.

### **Gutachtenkonfigurator**

Diese Rolle dient dazu, um Einstellungen im System vorzunehmen. Es können Vorlagen, Zeiterfassungskonten, Dokumentablagestruktur, Gebührenarten, Dokumenttypen, etc. verändert und angepasst werden. Es ist aber auch möglich bestehenden Gutachten zu lesen und gewissen Strukturen, wie beispielweise Checklisten in den Vorlagenbereich zu übernehmen.

Jedoch ist es nicht möglich in Gutachten zu arbeiten oder diese zu löschen, diese Rolle dient alleine zur Konfiguration des Systems. Es steht auch nur eine Ansicht, in dem Einstellungen vorgenommen werden können, zu Verfügung.

### **Benutzer**

Die Benutzerrolle besitzt keinerlei Möglichkeit Einstellungen am System vorzunehmen. Sie dient zum Abarbeiten von Gutachten, wobei dies in vier unterschiedlichen Ansichten möglich ist. Jede dieser Ansicht hilft dabei einen oder mehrere Schritte aus dem Prozessablauf auszuführen. Wie diese Ansichten aufgeteilt sind kann aus der Bedienungsanleitung, die sich im Anhang befindet, eingesehen werden.

Das System ist auch Mehrbenutzerfähig, es könnten also mehrere Personen gleichzeitig damit arbeiten. Gutachter, welche Projekte erstellen können anderen Kollegen Zugriff auf deren Gutachten erlauben, entweder nur lesend oder auch schreiben, dies kann über das Projektteam gesteuert werden.

## **5.6 Dokumentenmanagement**

Das Datei- oder Dokumentenmanagement in AXAVIAseries steht eng in Beziehung mit dem Betriebssystem. Im System gibt es eine Objektdefinition „\_DOCUMENT“, diese stellt die Basis aller Dateien und Dokumente dar. Die Objektdefinition „\_FOLDER“ ist das Äquivalent zu einem Ordner im Betriebssystem, es kann eine Instanz von „\_DOCUMENT“ nur einem „\_FOLDER“ zugeordnet sein, siehe Abbildung 71.



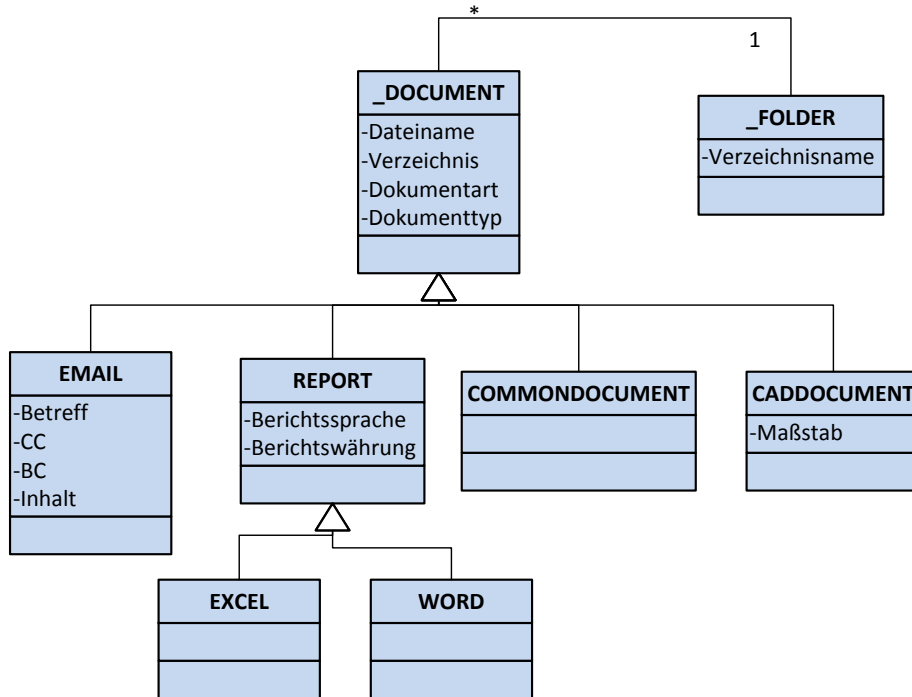


Abbildung 71: Dokumentenklassen in AXAVIAseries

Jegliche Instanzen dieser Klassen stehen nicht für die Datei selbst, die sich auf dem Dateisystem befindet, sondern stellen nur eine Referenz auf diese Datei dar. Sie verhalten sich ähnlich wie „Verknüpfungen“, die man aus dem Microsoft Betriebssystem Windows kennt. Der Unterschied zu Verknüpfungen ist jedoch die Möglichkeit, dass die Referenz, also die Instanz mit unterschiedlichsten Attributen ausgestattet werden kann und je nach Art der Objektdefinition diese mit Funktionalität erweitert wird.

Über den Dokumenttyp ist es möglich, Einstellungen vorzunehmen, die mittels externer Programme gestartet wird. Die Dokumentart kann dafür verwendet werden, um Dateien zu klassifizieren:

- WORD: Angebot, Rechnung, Bestellung, etc.
- EMAIL: Intern, Extern
- COMMONDOCUMENT: Bild, Aktenscan, etc.
- CADDOCUMENT: AutoCAD, SolidWorks, etc.

### 5.6.1 Arbeiten mit Dateien

Nach der Installation von AXAVIAseries wird auf einem Konfigurationsobjekt ein Eintrag für das Basisverzeichnis (Abbildung 72) erstellt. Dieses Verzeichnis steht für die Wurzel aller im System angelegten Verzeichnisse und Dateien.

Basisverzeichnis	D:\SQLServer\DATABASE Workdirs\EXPERTISE
Basisverzeichnis Projekte	D:\SQLServer\DATABASE Workdirs\EXPERTISE\projects
Basisverzeichnis Dokumentablage	D:\SQLServer\DATABASE Workdirs\EXPERTISE\documents
Basisverzeichnis Dokumentbibliothek	D:\SQLServer\DATABASE Workdirs\EXPERTISE\documentLibraries
Basisverzeichnis Vorlageobjekte	D:\SQLServer\DATABASE Workdirs\EXPERTISE\default

Abbildung 72: Basisverzeichnis der Ordner und Dateien

Wird also ein neues Projekt erzeugt, ergibt sich daraus ein eigener Ordner in dem /projects – Bereich, wobei auch hier wieder Ordnerstrukturen angelegt werden können.

Dateien können via Drag & Drop aus dem Windows-Explorer nach AXAVIASeries „kopiert“ werden. Es wird immer eine Kopie der Datei angelegt und physikalisch in das jeweilige Verzeichnis kopiert. Es ist auch möglich das Verzeichnisse inklusive deren Inhalt kopiert wird, dabei wird die Ordnerstruktur beibehalten.

### 5.6.2 Referenz auf eine Datei

Wie bereits erwähnt wird standardmäßig nicht der Inhalt einer Datei in die Datenbank kopiert, lediglich eine Referenz auf die Datei wird in der Datenbank erstellt, siehe Abbildung 73.

Die Größe der Referenz in der Datenbank hängt von der Anzahl der Attribute ab, die zur Objektdefinition gehören.

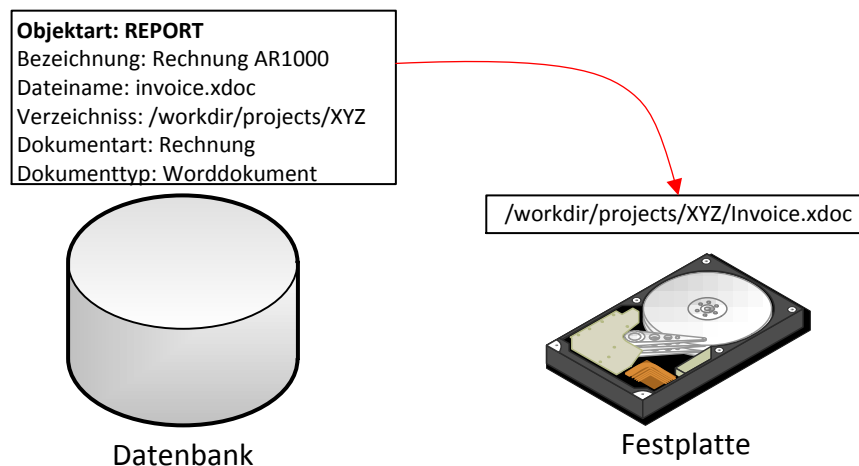


Abbildung 73: Dateireferenz

Vorteil dieser Methode ist, dass egal wie groß die eigentliche Datei ist, es keine Auswirkungen auf die Datenbankgröße hat. Auch der Datenbankserver wird nicht durch Aktionen beansprucht, die das Ein- und Auslesen solcher Binärdaten verursachen könnte. Natürlich bringt dies aber auch Probleme mit sich, da sichergestellt werden muss, dass sowohl Datenbank, als auch die Dateien auf dem physikalischen Speicher vor Zerstörung sicher sein müssen.

Außerdem benötigt jeder Benutzer, der mit der Datenbank arbeitet auch die Freigabe auf das Basisverzeichnis und dadurch ist es wiederum möglich, dass Benutzer über die Verzeichnisattribute, falls sichtbar, auf der Referenz direkt mit einem Dateexplorer auf die Datei zugreifen und diese verändern können. Damit dies vermieden wird sollten genau diese Attribute von normalen Benutzern verborgen bleiben.

### 5.6.3 Ein- & Auslesen von Dateien

Es ist aber auch möglich nicht nur die Referenz einer Datei in der Datenbank zu halten, sondern auch deren Inhalt. Beim Einlesen einer Datei wird ihr Inhalt in einen sogenannten BLOB (Binary Large Object) innerhalb der Datenbank abgelegt und der physikalische Inhalt gelöscht. Dadurch ist es möglich, Dateien vor externer Manipulation zu schützen.

Beim Auslesen wird der Inhalt aus der Datenbank wieder auf der Festplatte hergestellt. Jegliche Bearbeitung (z.B.: ein Doppelklick, zum Ausführen eines externen Programms) einer Datei bewirkt das Auslesen dieser automatisch. Dies ist bedingt durch die Abarbeitung durch externe Programme wie beispielsweise Microsoft Word, welche einen physikalischen Speicherort einer Datei benötigen um den Inhalt zu laden. Damit wird gewährleistet das AXAVIASeries mit allen externen Programmen kompatibel ist. Nach dem Bearbeiten des Programms kann die Datei wieder in die Datenbank eingelesen werden.

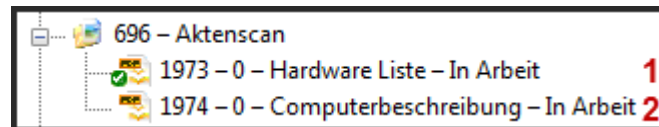


Abbildung 74: Ein- und Ausgelesene Datei

In Abbildung 74 sieht man, ob eine Datei eingelesen (2) oder ausgelesen (1) ist anhand eines „Overlay-Icon“, das durch ein grünes Häkchen dargestellt wird.

#### 5.6.4 Verbesserungen

Eine der wesentlich angedachten Verbesserung für das Dokumentenmanagement in AXAVIASeries ist die Verwendung von einem Vault. Ein Vault ist ein speziell geschützter Bereich auf einer Festplatte. Dieser Bereich wird verschlüsselt abgelegt und es ist nicht mehr möglich, über externe Programme auf Verzeichnisstrukturen oder Dateien innerhalb dieses Bereiches zuzugreifen.

Eine ähnliche Methode wird von TrueCrypt [11], ein bekanntes Open-Source on-the-fly Verschlüsselungssystem verwendet. Es können dabei ganze Partitionen aber auch Container, Datei welche als virtuelles Laufwerk dient, erzeugt werden, die automatisch alle darin enthaltenen Dateien verschlüsselt. Dabei bleibt die Datei voll manipulationsfähig für externe Programme, wie Microsoft Word, wenn das richtige Passwort zur Entschlüsselung verwendet wird.

AXAVIASeries befindet sich derzeit in einer Evaluierungsphase ob es dieses Verfahren integriert werden kann.

## 6 Umsetzung und Arbeitsweise

Das folgende Kapitel zeigt, mittels eines realitätsnahen Anwendungsfalls, wie die einzelnen Aufgabenstellungen abgearbeitet werden.

Dabei wird sich ein Benutzer zuerst über die Rolle des Konfigurators in das System einloggen und Einstellungen vornehmen. Nachdem Änderung an Vorlagen und des Benutzers vorgenommen wurde, wird ein Gutachtenprozess durchgespielt.

Die Kunden, beziehungsweise die in ein Gutachten verwickelten Parteien, werden mit Ansprechpartnern im System erzeugt und ein Forensikgutachten wird abgewickelt. Dabei werden Dateien und Dokumente im Projektbereich des Gutachtens verwaltet, ein Gutachten erstellt und Zeitaufzeichnungen getätigt.

Zum Schluss wird über den Aufwand eine Rechnung generiert und das Projekt archiviert.

### 6.1 Konfiguration

Nach der Installation müssen zuerst Einstellungen zum Bearbeiter der Software, externe Programmstartparameter, Ländereinstellungen, Zeitkonten,

Rechnungsvorlagen, Allgemeine Dokumentenablagestruktur, Schlagwortverzeichnis, Berichtvorlagen, usw. konfiguriert werden.

Dafür logt sich der Benutzer mit dem Konfigurationsaccount in die Software ein.

Benutzername: „expconfig“

Passwort: „expconfig“

### 6.1.1 Basiseinstellungen

Eine der wichtigsten Einstellung wird auf dem Wurzelobjekt (Abbildung 75) vorgenommen und zwar wo sich das Hauptverzeichnis auf der Festplatte befinden soll, in dem Dateien zur Bearbeitung abgelegt werden. Auch der Zähler für Projekte und Rechnungen wird hier definiert.

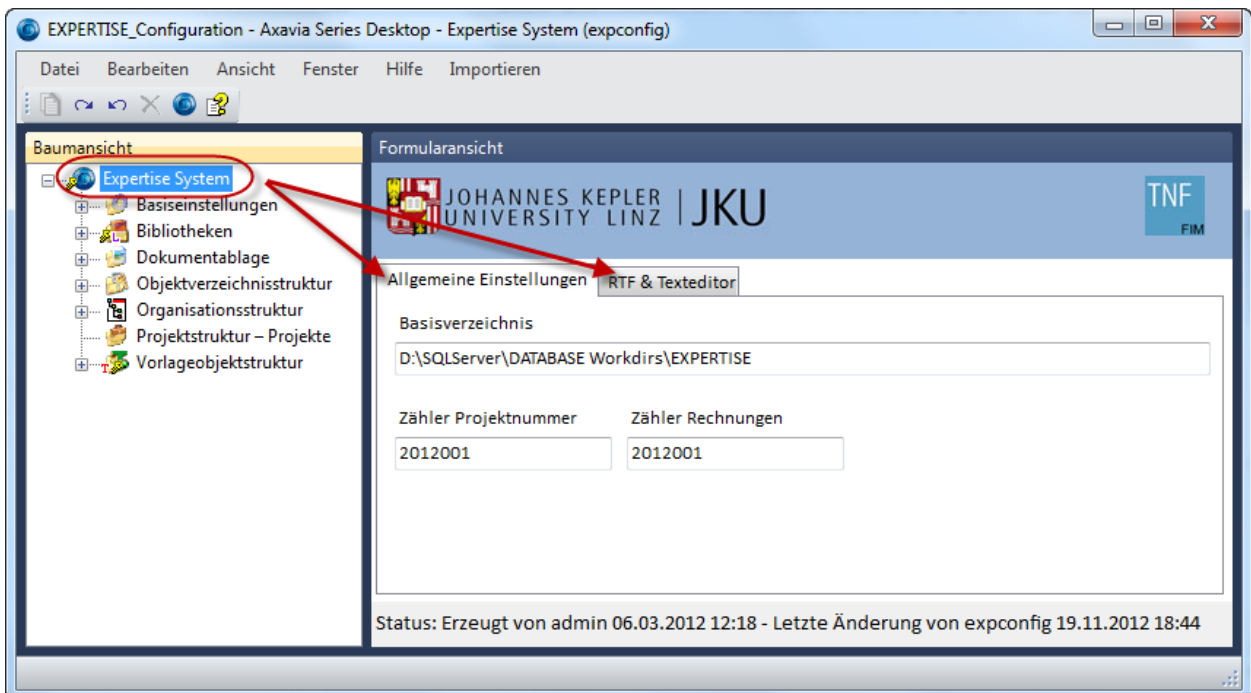


Abbildung 75: Basiseinstellungen

Im Reiter „RTF & Texteditor“ wird die Schriftart und Größe definiert, die später als Standard für das Gutachten gelten soll.

### 6.1.2 Protokollierung

In der Formularansicht ist bereits zu sehen (Abbildung 76), dass alle Änderungen protokolliert werden, jedes einzelne Objekt besitzt Informationen des Erzeugers und der letzten Änderung. Auch Änderungen auf Attributen sind einsehbar, bewegt man den Mauszeiger über ein Feld findet man in der Statuszeile die Information dafür, hier am Beispiel „Zähler Projektnummer“.

Zähler Projektnummer (PROJECT\_COUNTER) - letzte Änderung am "19.11.2012 18:44:44" von "expconfig"

Abbildung 76: Protokollierung von Vorgängen

### 6.1.3 Objektvorlagen

Weitere wichtige Einstellungen der Dokumenttypen, welches die Parameter bzw. Programme sind, mit welchem Dokumente bearbeitet werden.

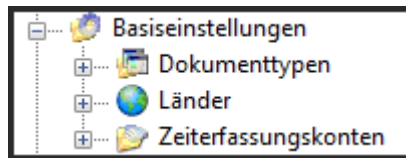


Abbildung 77: Objektvorlagen Basiseinstellung

In den Ländereinstellungen wird die Vorwahl, Währung, Kurzzeichen und die Umsatzsteuereinstellungen getätigt. Die Zeiterfassungskonten dienen der Zeitaufzeichnung, wobei hier Konten für Mühewaltung, Aktenstudium, Reisekosten, etc. erzeugt werden können um geleistete Stunden nach Art der Tätigkeit zu gruppieren, dazu siehe Abbildung 77.

In den Bibliotheken (Abbildung 78) ist es möglich, eine Art Materialstamm von Leistungen zu erzeugen, die der Gutachter später in Rechnung stellen kann.

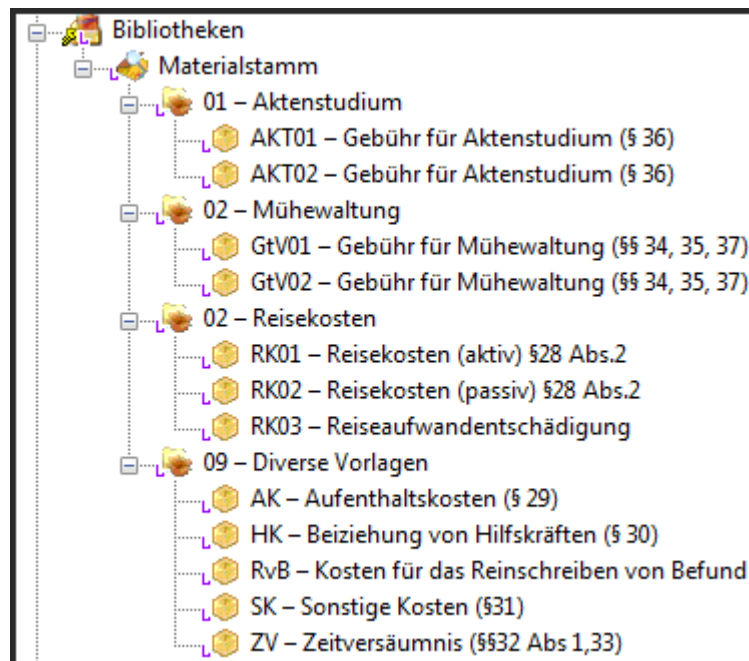


Abbildung 78: Objektvorlagen Bibliothek

Jede dieser Vorlagen kann individuell angepasst werden, siehe Abbildung 79.

Produktkonfiguration	
Produktnummer	Mengeneinheit
AKT02	Stk.
Produkttext kurz	Basispreis
Gebühr für Aktenstudium (§ 36)	40,00 €
Umsatzsteuergruppe	
0 - Keine Umsatzsteuer	
Produkttext Lang	
erster Band (€ 7,60 bis € 44,90) weitere(r) Band(Bände) (bis € 39,70)	

Abbildung 79: Beispiel Anpassung der Vorlage

#### 6.1.4 Dokumentenmanagementsystem

Die globale Dokumentenstruktur (Abbildung 80) kann auch verändert werden. Zu Beginn gibt es nur einen Hauptordner der Geschäftspartnerdokumente (=Kunden), eine für Allgemeine Dokumente, Ausgangsrechnungsberichte und Zeiterfassungsberichte.

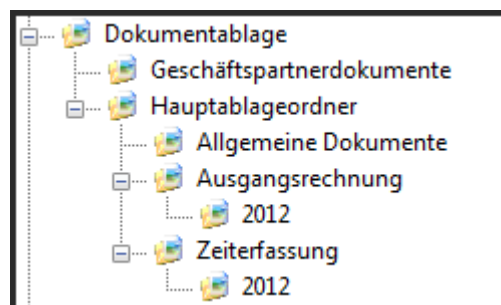


Abbildung 80: Globale Dokumentenstruktur

Nachdem Kunden erzeugt wurden, ist es möglich, einen Ordner, der direkt unter den „Geschäftspartnerdokumenten“ verknüpft wird, automatisch anzulegen, zu dem die Kundenrelevante Dokumente abgelegt sein können.

Anzumerken sei noch, dass für Gutachtenprojekte eigene Ablagestrukturen definiert sind, in diese Ordner gehören alle Dateien, die spezifisch für dieses Gutachten notwendig sind.

#### 6.1.5 Schlagwortverzeichnis

Das globale Schlagwortverzeichnis (Abbildung 81) kann auch gewartet bzw. eingesehen und falls notwendig Änderungen vorgenommen werden. Diese Schlagwörter werden später als Anhang im Gutachten automatisch verwaltet.

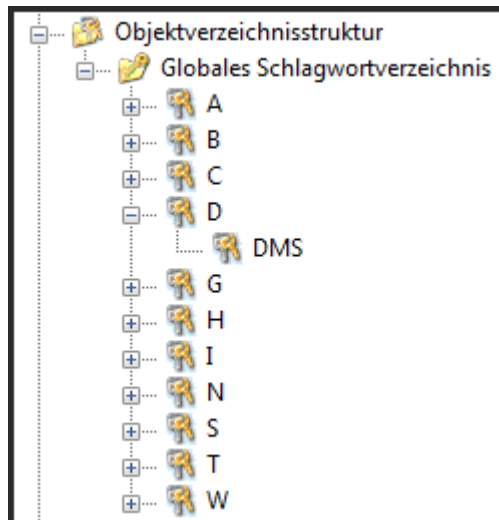


Abbildung 81: Globales Schlagwortverzeichnis

Hier zu sehen ist der Eintrag für das Schlagwort „DMS“, siehe Abbildung 82.

Globales Schlagwort	
Knotenname	<input type="text" value="DMS"/>
Beschreibung	<input type="text" value="Dokumentenmanagement System"/>

Abbildung 82: Beispiel Schlagwort "DMS"

### 6.1.6 Berichtvorlagen

Damit bei Berichten in den Kopf- und Fußzeilen die richtige Information vom Gutachter, Unternehmen und der Bankinformation enthalten ist, ist es wichtig, dass diese Informationen im System hinterlegt werden. Dies geschieht in der Organisationsstruktur.

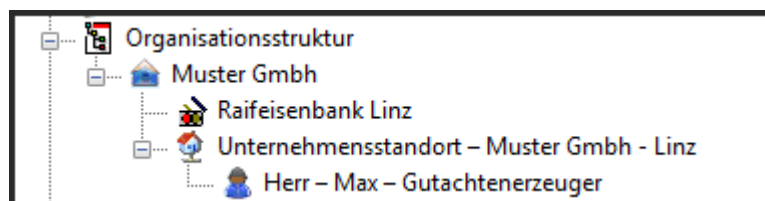


Abbildung 83: Organisationsstruktur

Unser fiktives Unternehmen trägt den Namen „Muster GmbH“. Es wird ein Bankkonto angelegt und die zugehörigen Daten eingetragen (Abbildung 83 und Abbildung 84).

Bankdaten	
Bank	
Raifeisenbank Linz	
Bank Land	Bank PLZ
A	4040
Bank Ort	
Linz	
Bank Adresse	
Linzer Strasse 45	
Bankkontonummer	
99999999	
Bankleitzahl	
BLZ11111	
BIC Nummer	
BIC9308	
IBAN-Nummer	
IBAN12313	

Abbildung 84: Bankdatendialog

Da es für ein Unternehmen mehrere Standorte geben kann, die auch unterschiedliche Bankverbindungen haben können, ist es notwendig diese einzutragen und am jeweiligen Standort auszuwählen (Abbildung 85). Dabei werden Informationen vom Unternehmen auf die einzelnen Standorte vererbt, die aber falls notwendig überschrieben werden können.

Unternehmensstandort	Bankverbindung
Firmenname/Unternehmen	
Muster GmbH	
Unternehmensstandort	
Muster GmbH - Linz	
Adresse Unternehmensstandort	
Hafenstr. 47-51	
Land Unternehmensstandort	PLZ Unternehmensstandort
A	4020
Ort Unternehmensstandort	
Linz	
E-Mail Unternehmensstandort	
office@muster.com	
Internetadresse Unternehmensstandort	
www.muster.com	

Abbildung 85: Unternehmensstandortdialog

Zum jeweiligen Standort wird dann ein Mitarbeiter angelegt, dieser wird dem Benutzerkonto „user“ zugeordnet (Abbildung 86). Dieses Benutzerkonto besitzt die



nötigen Berechtigungen um später Kunden, Ansprechpartner und Gutachten zu erzeugen.

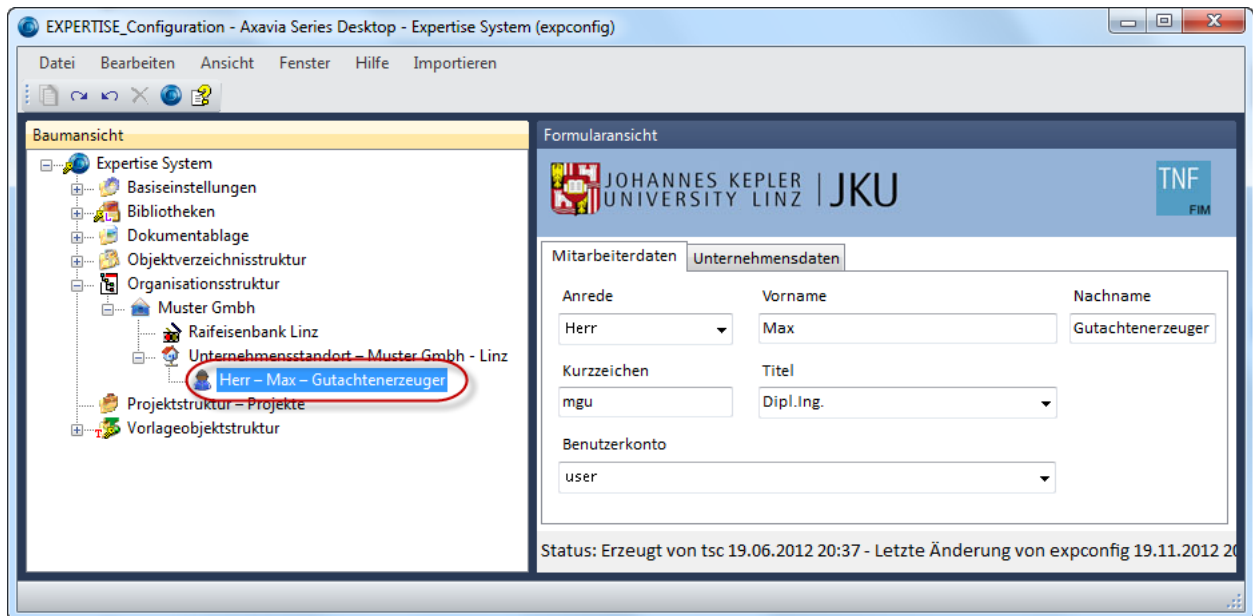


Abbildung 86: Mitarbeiter mit Benutzerkonto verknüpfen

Am Ende können noch diverse Vorlagen (Abbildung 87) angepasst werden, dazu zählen:

- Ausgangsrechnungen
- Checklisten
- Gutachten
- Dokumente
- Berichte
- Importvorlagen

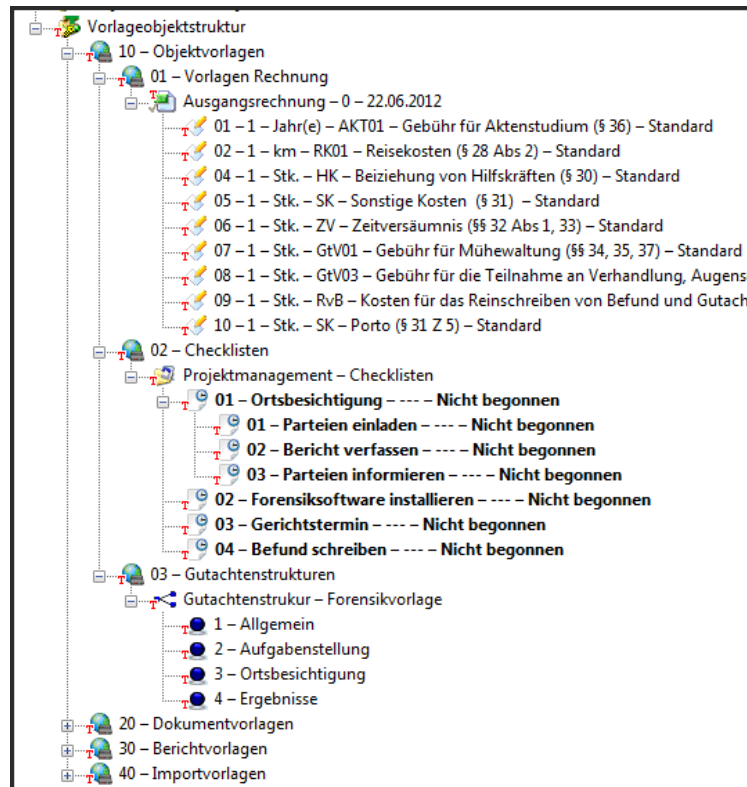


Abbildung 87: Übersicht der unterschiedlichen Vorlagen

### 6.1.7 Zusammenfassung

Um später mit dem Programm richtig arbeiten zu können, ist es sehr wichtig Einstellungen und Vorlagen zu definieren, aber auch zu warten. Nach den ersten Gutachtenprojekten können abgearbeitete Gutachten, Checklisten, Rechnungen, usw. direkt aus dem Projekt in diese Vorlagenstrukturen kopiert werden, um diese später bei neuen Projekten anzuwenden.

Folgende Punkte aus der Aufgabenstellung werden beim Konfigurieren abgearbeitet:

- Server-basiertes System
- Dokumentenmanagementsystem
- Vorlagen
- Automatisches Glossar (bearbeiten)
- Protokollierung der Vorgänge

## 6.2 Parteien erzeugen

Nachdem alle nötigen Einstellungen vorgenommen wurden, meldet man sich in das System mit dem Benutzer „user“ und dessen Passwort „user“ ein. Diesem Benutzer wurde in den Konfigurationseinstellungen (Objektvorlagen) ein Mitarbeiter „Max Gutachtenerzeuger“ zugeordnet.

Für den weiteren Verlauf erzeugen wir diverse Parteien mit Ansprechpartnern:

- Landesgericht Linz
  - Richter Rainer Nimmervoll
- Firma Müller & Meyer
  - Toni Müller
  - Thomas Meyer
- Familie Mustermann
  - Stefan Mustermann

In den nächsten beiden Kapiteln, wird erläutert wie die Partei „Landesgericht Linz“ mit seinem Ansprechpartner „Richter Rainer Nimmervoll“ erzeugt wird.

### 6.2.1 Kunden / Parteien

Nach dem Starten des Programmes befindet man sich in der Startansicht „EXPERTISE\_Main“, siehe Abbildung 88.

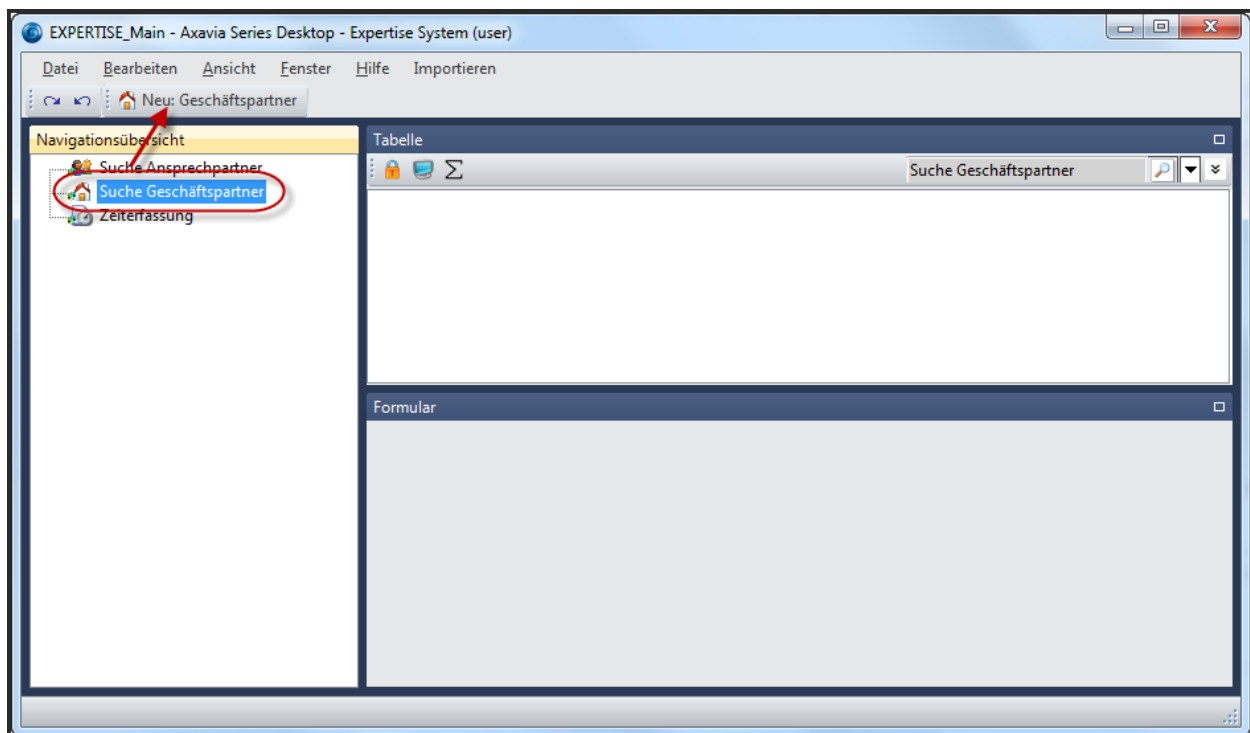



Abbildung 88: Startansicht "EXPERTISE\_Main"


Markiert man „Suche Geschäftspartner“ erhält man die Möglichkeit, über die Symbolleiste einen neuen Kunden / Partei anzulegen. In der Tabelle rechts daneben ist es möglich bestehende Stammdaten zu suchen.

Über die Symbolleiste „Neu Geschäftspartner“ erhält man ein Formular in dem notwendige Informationen zu dieser Partei eingetragen werden (Abbildung 89).

Formular



JOHANNES KEPLER  
UNIVERSITY LINZ | JKU



Kundeninformation

<b>Firmenname</b> <input style="width: 95%;" type="text" value="Landesgericht linz"/> <input style="width: 95%;" type="text" value="---"/>	<b>Land</b> <input style="width: 95%;" type="text" value="A"/>	<b>Ort</b> <input style="width: 95%;" type="text" value="Linz"/>
<b>Telefon</b> <input style="width: 95%;" type="text" value="+43 (0)57 60121"/>	<b>PLZ</b> <input style="width: 80%;" type="text" value="4040"/>	<b>Adresse</b> <input style="width: 95%;" type="text" value="Fadingerstraße 2"/>
<b>Telefax</b> <input style="width: 95%;" type="text" value="+43 (0)57 60121-12930"/>	<b>Bemerkung</b> <div style="border: 1px solid black; padding: 5px; font-size: 0.9em;"> Parteienverkehr: bitte telefonisch bei Gericht erfragen  Amtstag: Dienstag von 8.00 bis 12.00 Uhr  Bankverbindung: PSK-Nr 00005450064  Bankleitzahl: 60000  IBAN: AT886000000005450064  BIC: OPSKATWW  DVR-Nr: 0000550868 </div>	
<b>Internet Adresse</b> <input style="width: 95%;" type="text" value="---"/>		

Abbildung 89: Kundeninformation

## 6.2.2 Ansprechpartner

Es wird auch ein Ansprechpartner „Richter Rainer Nimmervoll“ zu dieser Partei angelegt, dies geschieht in der eigenen Ansicht „EXPERTISE\_Businesspartner“, siehe Abbildung 90. Um in diese Ansicht zu wechseln sucht man sich die jeweilige Partei in der Tabelle, markiert diese und mit einem Doppelklick öffnet sich die Ansicht.

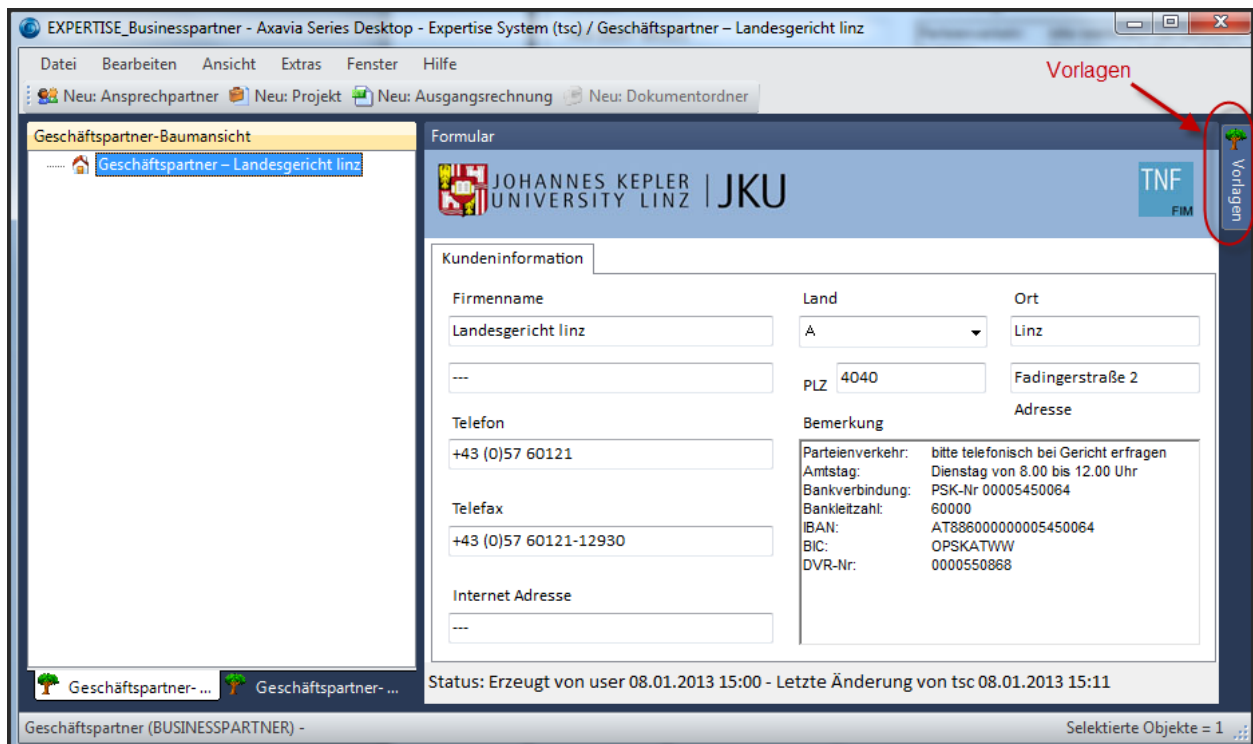


Abbildung 90: Geschäftspartneransicht "EXPERTISE\_Businesspartner"

In der Symbolleiste erstellt man nun den Ansprechpartner und trägt die Daten zu diesem ein (Abbildung 91).

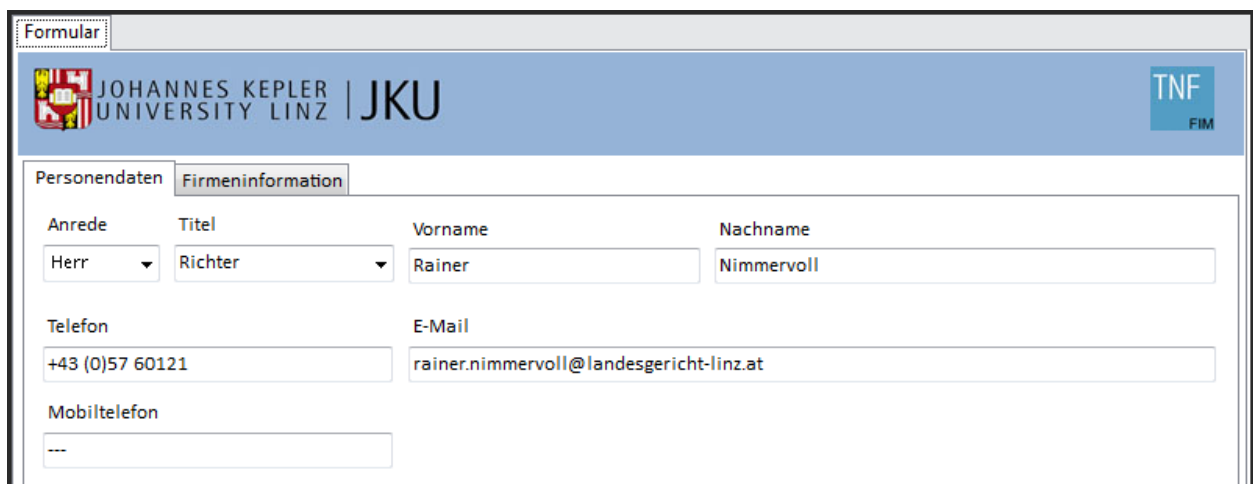


Abbildung 91: Ansprechpartner Formulardialog

Dieser Vorgang wird nun für alle Parteien wiederholt.

### 6.2.3 Dokumentablage

Zu den jeweiligen Parteien kann ein Ordner mit diversen Unterordnern angelegt werden in dem Allgemeine Dokumente hinterlegt sind. Um dies zu bewerkstelligen, genügt es einen Kunden zu markieren und mit Hilfe des Kontextmenüs „Erzeuge Geschäftspartnerdokumentordner“ zu wählen.

In der Ansicht „EXPERTISE\_Businesspartner“ können über die Symbolleiste weitere Ordner angelegt werden (Abbildung 92).

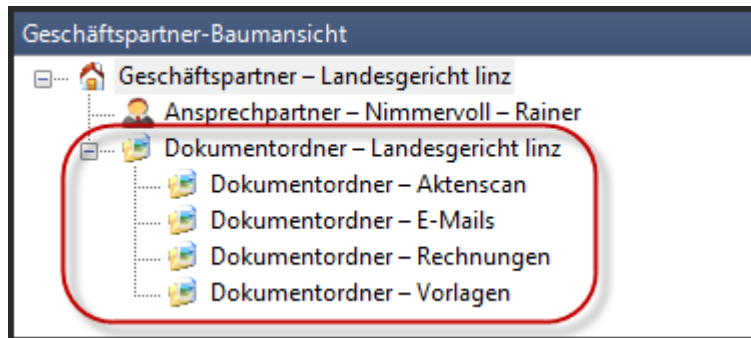


Abbildung 92: Dokumentablagestruktur zu einem Geschäftspartner

Dateien werden dann über Drag & Drop in den jeweiligen Ordner kopiert und verknüpft, oder es können aus dem Vorlagen Dokumente übernommen werden.

#### 6.2.4 Zusammenfassung

In der Startansicht sucht man oder erzeugt man seine Kunden / Parteien und ordnet diesen dann Ansprechpartner zu, dies sind die Stammdaten im System. Projekte und Rechnungen werden später diesen Geschäftspartnern zugeordnet.

Es ist auch möglich Dokumente zu organisieren und diese den jeweiligen Geschäftspartnern zuzuordnen, wobei noch einmal unterschieden wird zwischen Dokumenten zu Geschäftspartner und dem jeweiligen Gutachtenprojekt, dazu aber im nächsten Kapitel mehr.

Folgende Punkte aus der Aufgabenstellung wurden berücksichtigt:

- Dokumentenmanagementsystem
- Vorlagen
- Protokollierung der Vorgänge

### 6.3 Gutachtenprojekt erzeugen

Das eigentliche Gutachten kann dann zum jeweiligen Auftraggeber erzeugt werden. Da ein Gutachten aus vielen unterschiedlichen Themen besteht wie Termine, Dokumente, Notizen, Schriftliche Ausarbeitung, Schriftverkehr, Checklisten usw. wird der Begriff „Gutachtenprojekt“ geschaffen.

Ein Gutachtenprojekt ist ein abgeschlossener Bereich, in dem alle Themen gesammelt werden, d.h. dass alle Informationen zu einem Gutachten zusammen getragen werden und auf einem Blick zugänglich sind. Dies macht deshalb Sinn, damit zu einem späteren Zeitpunkt die ganzen Daten und Informationen einfacher archiviert werden können.

Im folgenden Beispiel wird gezeigt, wie für die Familie Mustermann ein Gutachtenprojekt mit allen nötigen Zusatzinformationen angelegt wird. Dazu erzeugt oder sucht man sich den jeweiligen Auftraggeber des Gutachtens, in der Geschäftspartneransicht „EXPERTISE\_Businesspartner“ kann dann ein neues Projekt erzeugt werden (Abbildung 93).

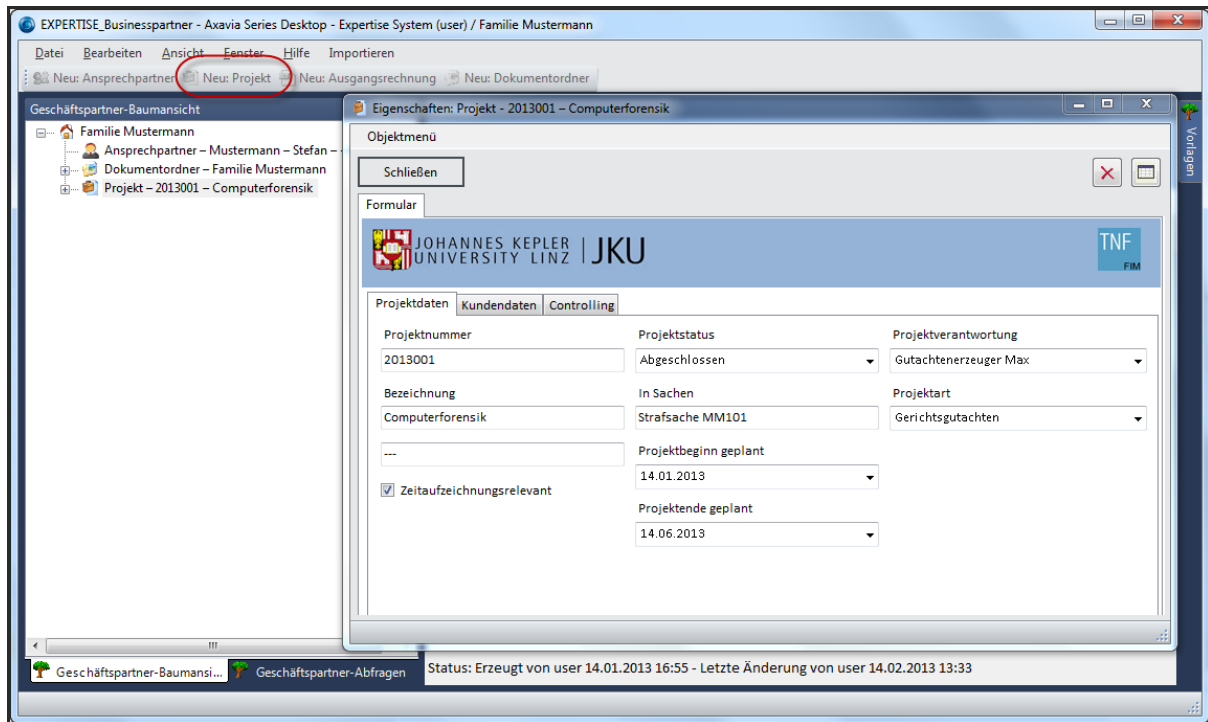


Abbildung 93: Gutachtenprojekt erzeugen

Zum Projekt selber können diverse Einstellungen gemacht werden, von der Projektnummer, Projektstatus, Projektverantwortlicher, Bezeichnung, In Sachen, Projektart und ob es zeitaufzeichnungsrelevant ist, siehe Abbildung 94. In den zwei weiteren Reitern „Kundendaten“ und „Controlling“ kann der Ansprechpartner zum Projekt definiert werden und Informationen abgefragt werden, wie viele Stunden bereits auf das Projekt verbucht wurden.

Nachdem man diese Daten eingegeben hat und mit „Schließen“ den Dialog beendet kann mit einem Doppelklick auf dieses Projekt eine eigene Ansicht geöffnet werden, in dem das Projekt als Wurzel in der Baumansicht angezeigt wird.

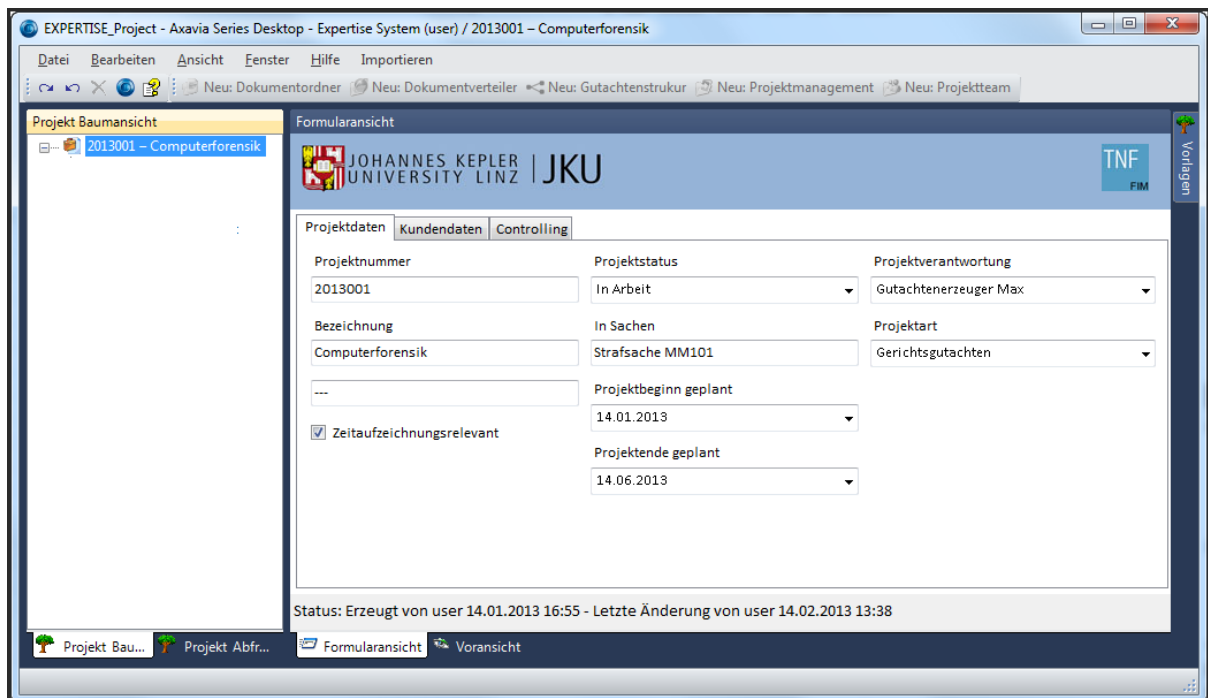


Abbildung 94: Projektinformationen eintragen

Es können noch einmal alle eingegebenen Daten geprüft und verändert werden, außerdem ist es möglich über die Symbolleiste diverse Informationen zum Gutachten abzulegen.

### 6.3.1 Dokumentablage

Wie bereits bei Geschäftspartnern, ist es möglich ganze Dokumentenstrukturen zu Gutachtenprojekten zu erzeugen bzw. über die Vorlagen zu erstellen (Abbildung 95). Falls bereits auf dem Dateisystem Ordner existieren, können auch diese per Drag & Drop in das System kopiert werden.

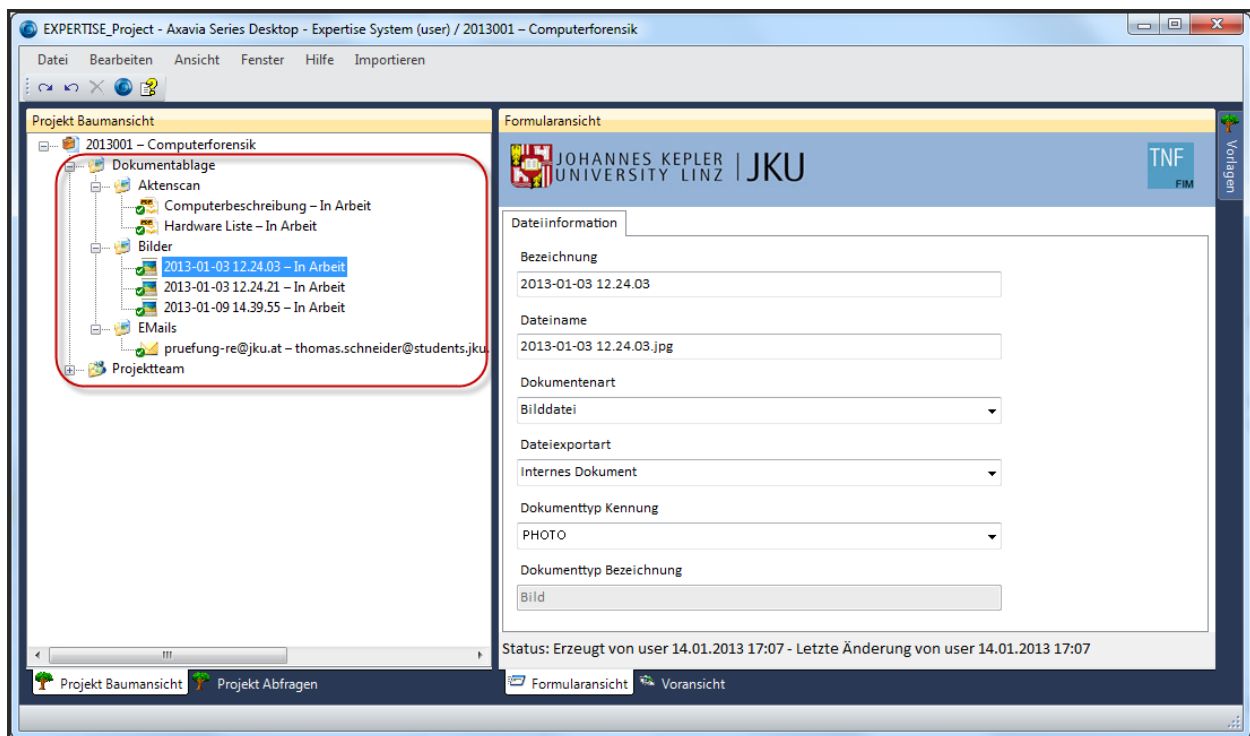


Abbildung 95: Dokumentablagestruktur zu einem Projekt

Im Bild zu sehen ist eine Dokumentablage für das Projekt in dem unterschiedliche Dateitypen in das System kopiert wurden. Es können ganz einfach neue Ordner über die Symbolleiste erstellt und benannt werden.

### 6.3.2 Projektteam erstellen

Anfangs sollten alle beteiligten Parteien, die für dieses Gutachten notwendig sind, erzeugt werden. Dafür wird über die Symbolleiste zuerst ein „Neu: Projektteam“ erzeugt zu dem dann „Neu Projektmitglieder“ eingefügt werden können.

Dabei können auch Personen aus anderen Kunden integriert werden, indem zuerst eine Geschäftsbeteiligung erzeugt wird, in dem der Kunden ausgewählt wird. Nachdem ist es möglich einen der Ansprechpartner als Projektmitglied zu wählen, siehe Abbildung 96.





Abbildung 96: Projektteam zu einem Gutachten erstellen

Zu den Parteien können Notizen, Telefoninformationen (Geschäftskontakte) und E-Mails (Nachrichten) erstellt werden, wie im nächsten Bild (Abbildung 97) zu sehen ist.

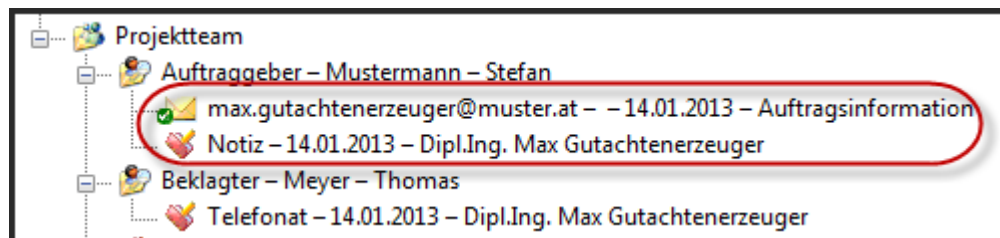


Abbildung 97: Geschäftskontakte und Nachrichten zum Projektteam

### 6.3.3 Checklisten erstellen

Über den Knoten Projektmanagement ist es möglich, Aktivitäten zu dem Projekt zu generieren, welche abgearbeitet werden müssen. Diese können entweder über die Symbolleiste, oder aber auch über vordefinierte Vorlagen eingefügt werden.

Im nächsten Bild (Abbildung 98) ist dargestellt, wie aus den Vorlagen die Aktivität „Ortsbesichtigung“ inklusive deren gegliederten Unteraktivitäten in das Projektmanagement des Projektes übernommen wird.

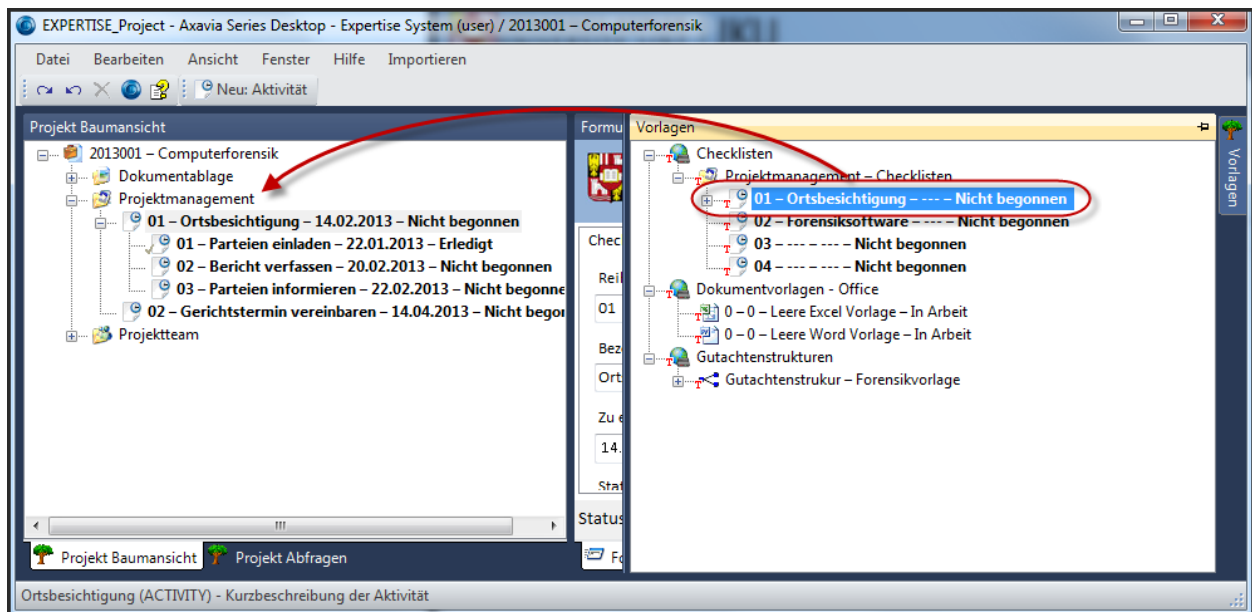


Abbildung 98: Checklisten aus Vorlage erzeugen

In der Baumansicht „Projekt Abfragen“ ist es des Weiteren möglich, dass man die Aktivitäten, die noch offen sind sucht (Abbildung 99).

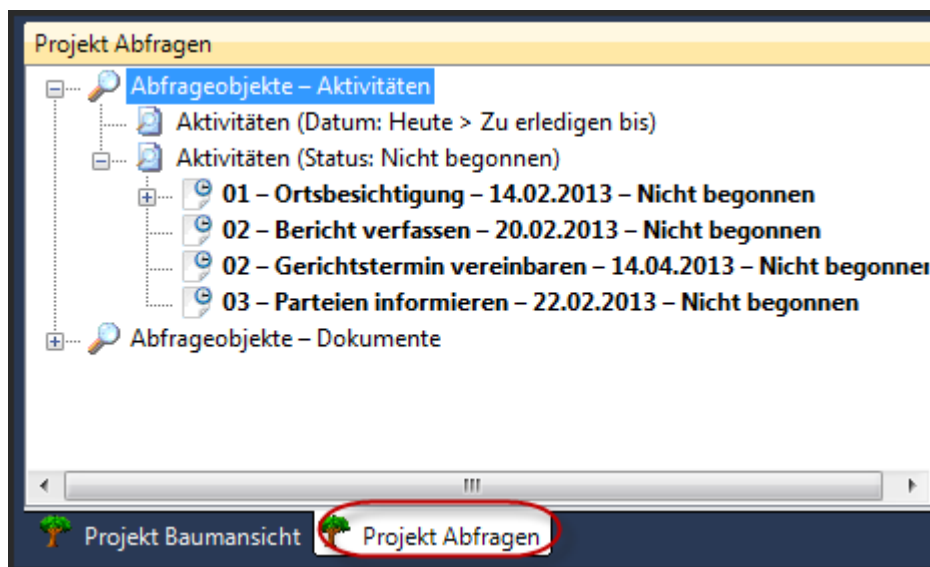


Abbildung 99: Abfragen zu einem Projekt

### 6.3.4 Gutachten erzeugen

Die Überlegung zum Erzeugen eines Gutachtens war jene, dass kapitelweise Informationen zusammengetragen werden. Die einzelnen Kapitel können auch von mehreren Gutachtern unabhängig erstellt werden, es ist auch denkbar, dass dies gleichzeitig auf einem Gutachtenprojekt passiert, solange ein geregelter Arbeitsprozess eingehalten wird. Das gesamte Gutachten besteht dann aus der Summe der einzelnen Kapitel. Dafür war es notwendig eine Gutachtenstruktur mit Kapiteln im System zu modellieren (Abbildung 100).

Das Gutachten kann entweder komplett neu aufgebaut werden, oder es besteht auch die Möglichkeit, dass ein vordefiniertes Gerüst aus den Vorlagen übernommen wird.

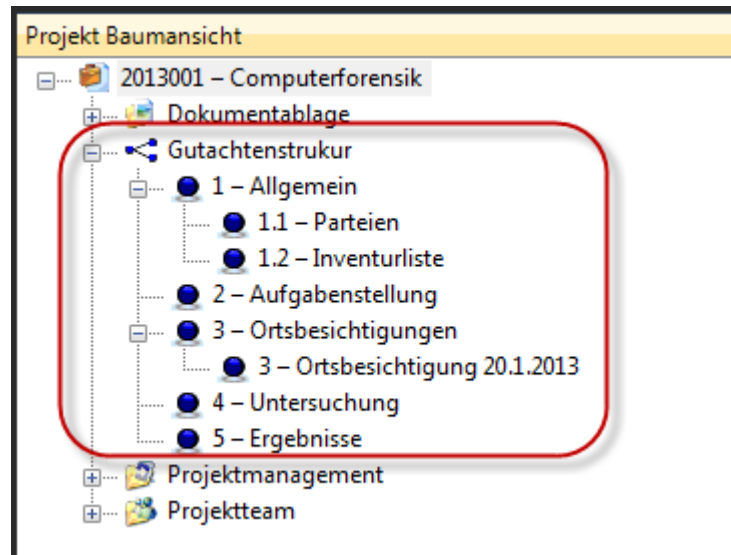


Abbildung 100: Gutachtenstruktur mit Kapiteln

Für jedes Kapitel, aber auch für das Gutachtenstruktur - Objekt, ist es möglich einen Editor zu öffnen und einen Inhalt zu generieren. Der zu Verfügung stehende Texteditor, welcher mit einem Doppelklick in das Formularfenster geöffnet wird, ist in der Lage Text, Tabellen, Bilder, usw. abzuspeichern.

Das Gutachtenstruktur – Objekt dient außerdem auch als Titelseite, in der auch Informationen aus der Datenbank übernommen werden können. Dabei ist im nächsten Bild zu sehen wie aus dem Projekt – Objekt die Information „In Sachen“ in das Gutachten übernommen wird. Es wird der interne Name des Attributes benötigt, in diesem Fall „EXPERTISEISSUE“. Der interne Name kann über die Symbolleiste beim Bewegen des Mauszeigers über ein Attribut entnommen werden (Abbildung 101).

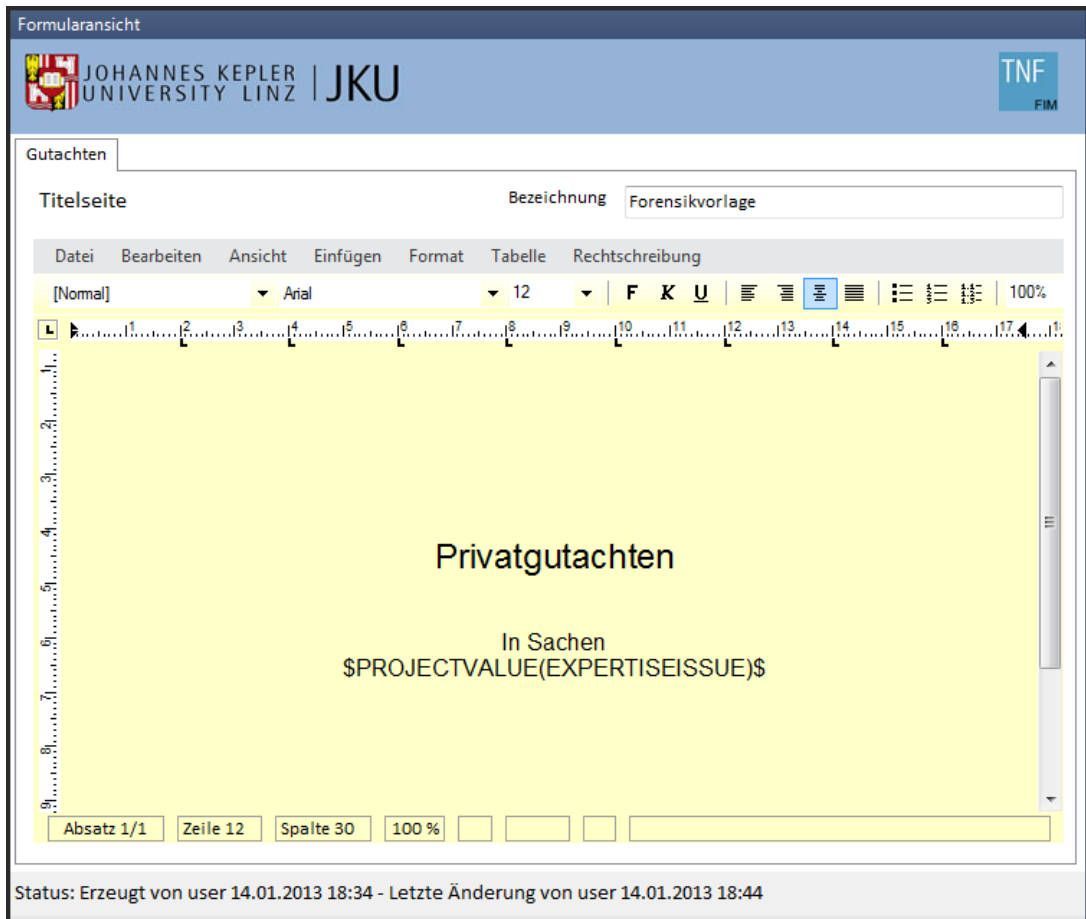


Abbildung 101: Titelseite eines Gutachtens

Für Kapitel (Abbildung 102) kann noch übernommen werden, ob davor ein Seitenumbruch stattfinden soll oder nicht. Auch die Sortierung kann über die Kapitelnummer eingestellt werden, beim Erstellen des Berichtes wird genau diese berücksichtigt.

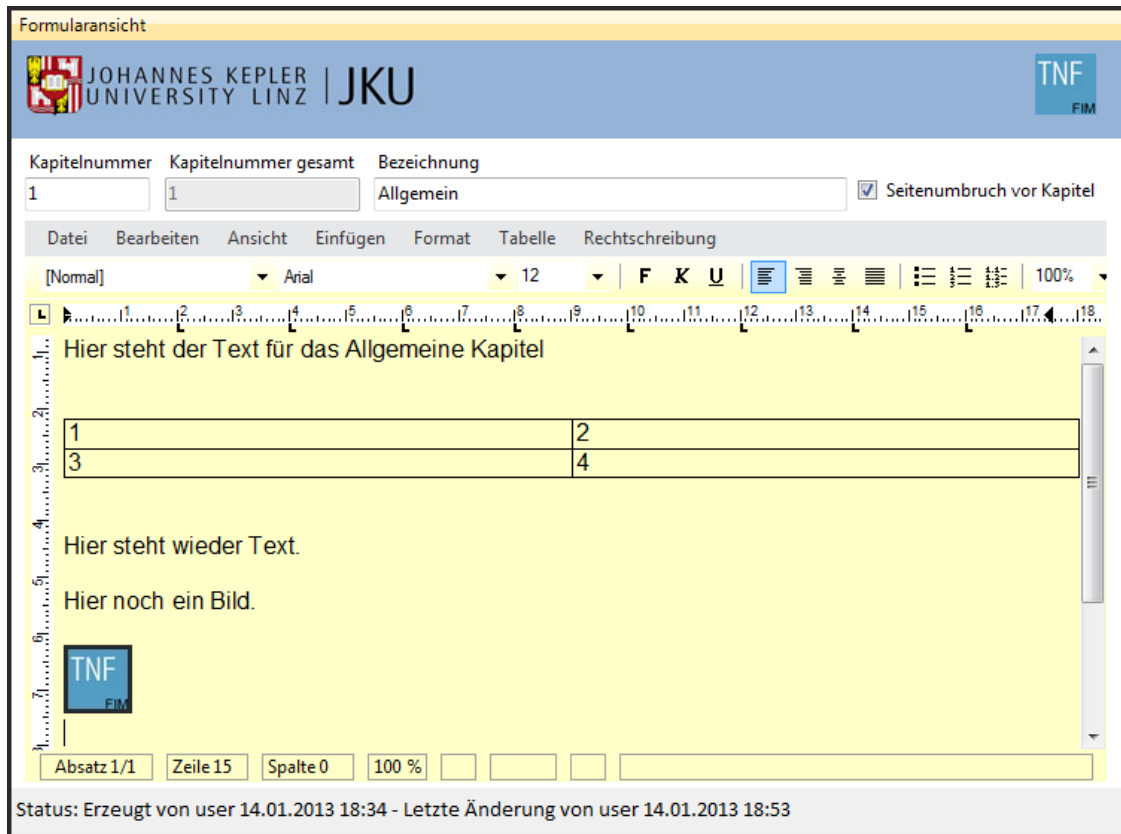


Abbildung 102: Formulario dialog mit Texteditor eines Kapitels

Ein Gutachtenbericht, der alle Kapitel zusammenfasst, kann erzeugt und abgespeichert werden in dem man die Gutachtenstruktur markiert und dann über das Kontextmenü „Erzeuge..“ oder „Voransicht Standardbericht - Gutachtenbericht“ wählt.

### 6.3.5 Zusammenfassung

Das Gutachtenprojekt trägt alle notwendigen Informationen, Notizen, Dokumente und die Kommunikation zu einem Fall zusammen und hilft diese strukturiert zu halten, wie im nächsten Bild (Abbildung 103) ersichtlich ist.

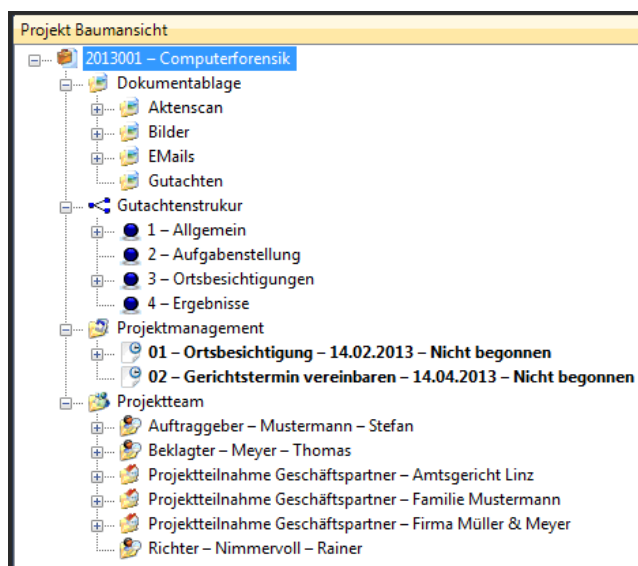


Abbildung 103: Übersicht Objekte in einem Projekt

Über Abfragen welche im System hinterlegt und erweiterbar sind, können dann gezielt Daten durchsucht und angezeigt werden. Ein Beispiel dafür wäre „Alle E-Mails“ oder „Alle Dokumente vom Typ: Aktenscan“.

Aus der Aufgabenstellung wurde berücksichtigt:

- Dokumentenmanagementsystem
- Vorlagen
- Protokollierung der Vorgänge

Nach der Berichterstellung, siehe Abbildung 104, ist die definierte Gutachtenstruktur mit seinen verschachtelten Kapiteln auch in Microsoft Word zu erkennen und wird über das Inhaltsverzeichnis abgebildet.

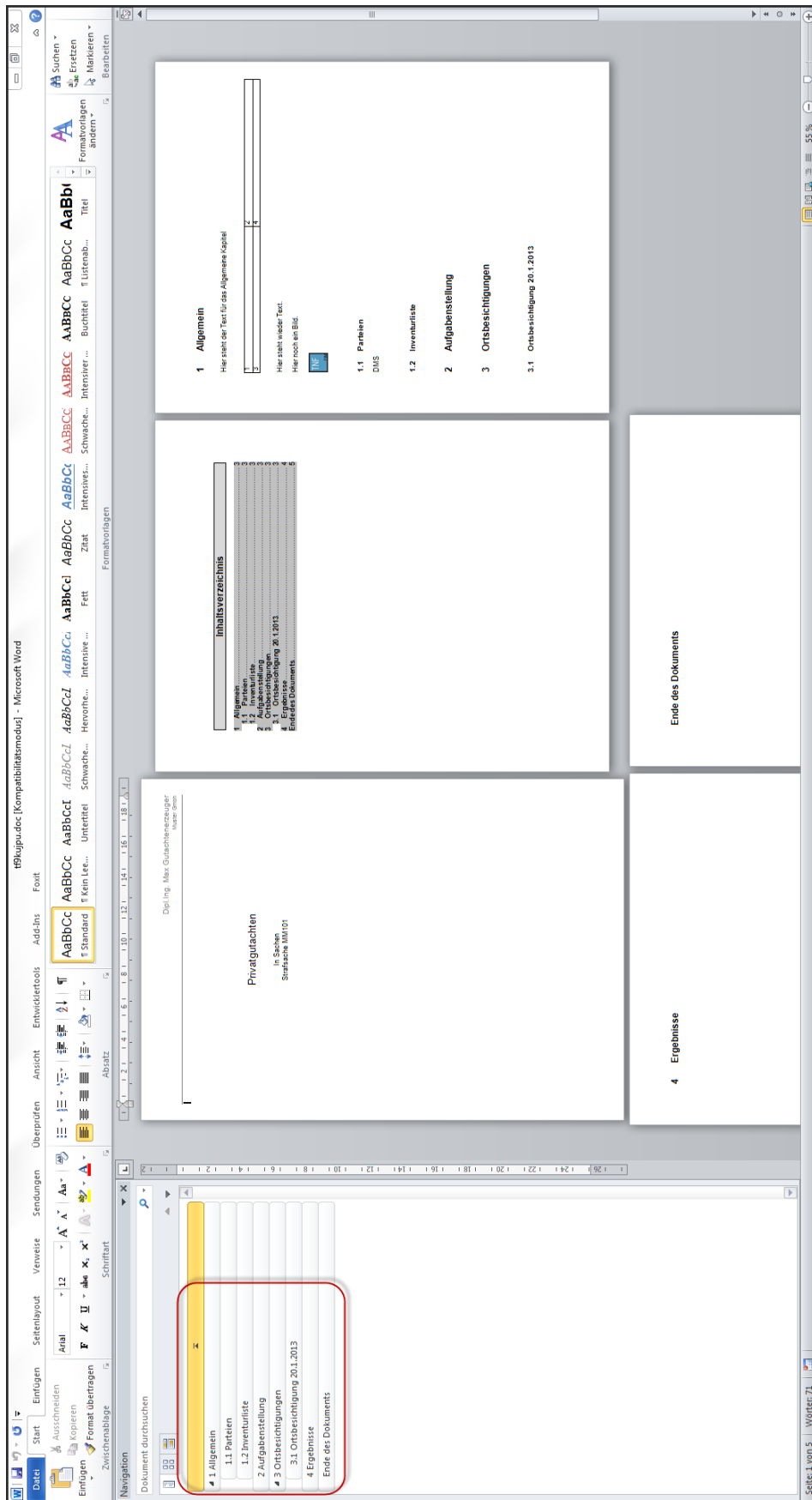


Abbildung 104: Gutachtenbericht

## 6.4 Schlagwortverzeichnis generieren

Aus der Aufgabenstellung geht hervor, dass es möglich sein soll, ein Schlagwortverzeichnis mit gewählten Wörtern automatisiert zu erstellen, das dann im Gutachten als Anhang mitgeliefert wird. Nachdem beziehungsweise während, für ein Gutachten die Kapitel verfasst werden, kann der Gutachter Wörter „beschlagworten“. Dazu ist es nötig das Wort im Texteditor zu markieren und mittels Kontextmenü „Beschlagwortung“ zu wählen.

Es genügt ein Wort einmal in einem Kapitel zu „beschlagworten“, damit wird dieses dann auch im Schlagwortverzeichnis eingetragen. Um zu sehen welche Wörter in einem Kapitel als Schlagwort markiert wurden, ist es möglich über das Texteditormenü „Ansicht → Schlagwörter“ diese farblich dazustellen.

Schlagwörter werden global in der Datenbank eingetragen, es ist also möglich bereits erklärte Schlagwörter ohne diese erneut definieren zu müssen in sein Gutachten aufzunehmen, wenn diese „beschlagwortet“ wurden.

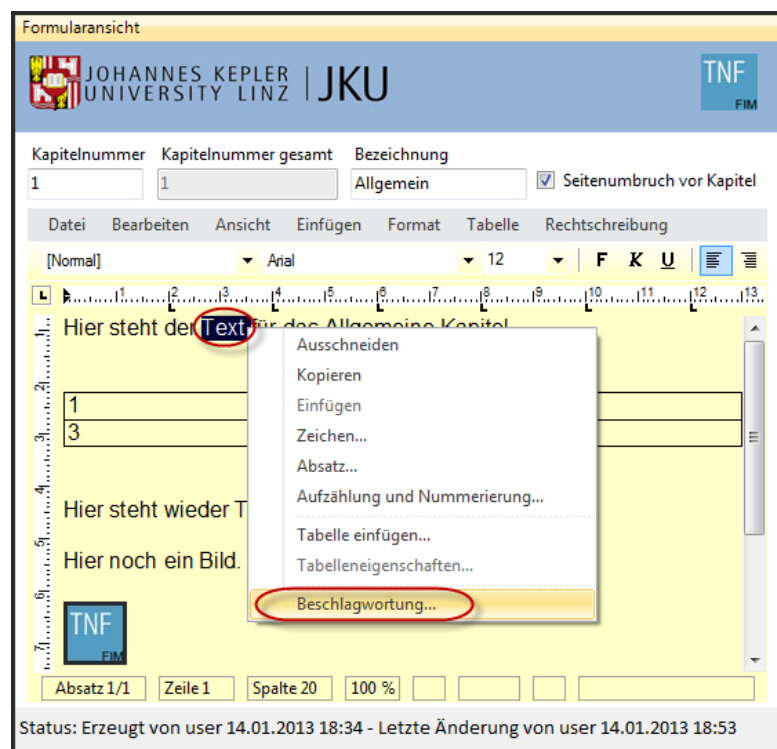


Abbildung 105: Wörter im Texteditor als Schlagwort definieren

Im Bild (Abbildung 105) ist zu sehen wie das Wort „Text“ zu einem Schlagwort definiert wird.



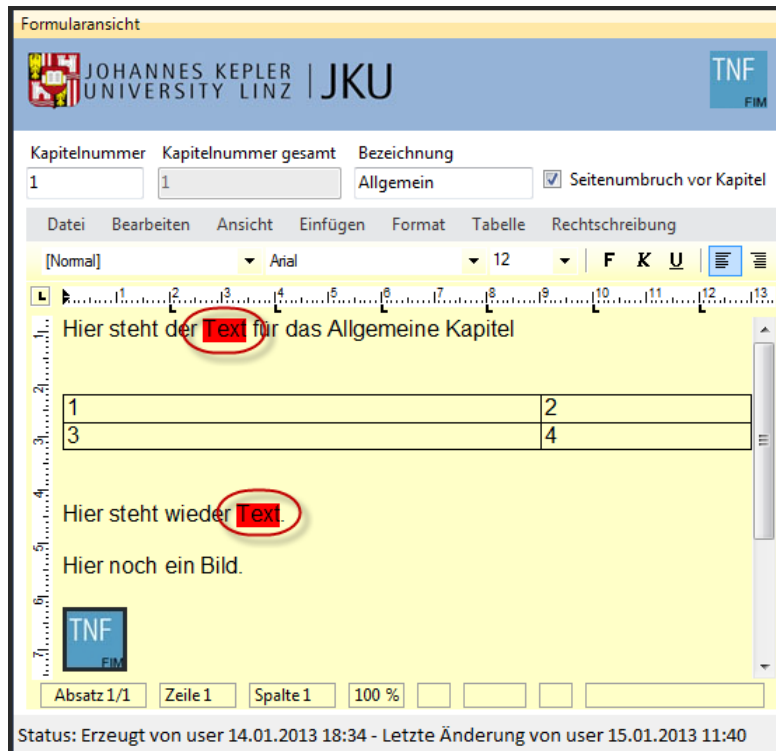


Abbildung 106: Darstellung der Schlagwörter in einem Kapitel

Über die Ansicht (Abbildung 106) können Schlagwörter eines Kapitels dargestellt werden. Nachdem Kapitelweise Schlagwörter definiert wurden, kann das Schlagwortverzeichnis zu einem Gutachten erstellt werden. Um diesen Prozess anzustoßen ist es nötig die Gutachtenstruktur zu markieren und über das Kontextmenü „Schlagwortverzeichnis erstellen / updaten“.

Dadurch wird ein Algorithmus in Gang gesetzt, der alle Wörter aus den Kapiteln welche „beschlagnahmt“ wurden, geordnet auflistet.

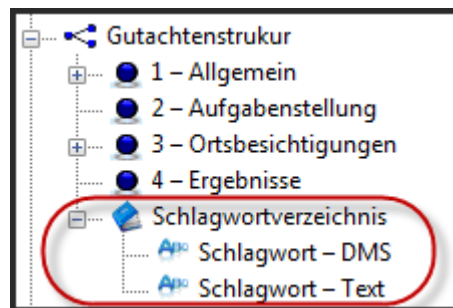


Abbildung 107: Schlagwortverzeichnisstruktur

Zu sehen sind in Abbildung 107 zwei Schlagwörter, die im Gutachten gekennzeichnet wurden, das Schlagwort „DMS“ ist bereits in der Datenbank enthalten und hält die gewählte Definition parat (Abbildung 108).

Schlagworteintrag
Bezeichnung
DMS
Beschreibung
Dokumentenmanagement System

Abbildung 108: Beschreibung für ein Schlagwort

Für das Schlagwort „Text“ wurde noch keine Beschreibung gefunden, diese kann nun direkt im Gutachtenprojekt definiert werden. Um die Beschreibung des Schlagworts auch global im System zu speichern, ist es notwendig dies dem System mitzuteilen. Dafür ist es nur notwendig das Schlagwort „Text“ im der Baumansicht zu markieren und über das Kontextmenü den Eintrag „Globales Schlagwort updaten“.

Um Schlagwörter aus dem Gutachten zu löschen, kann ebenfalls das Kontextmenü verwendet werden und zwar mit dem Eintrag „Schlagwort aus Gutachten entfernen“. Es ist also nicht notwendig zu wissen, in welchem Kapitel man ein Wort „beschlagwortet“ hat.

Der Gutachtenbericht selbst ist so vorbereitet, dass die Schlagwörter als Anhang sortiert angegeben werden.

#### 6.4.1 Zusammenfassung

Leider war es nicht möglich, voll automatisch Begriffe als Schlagwörter in einem Gutachten zu generieren, da von mir keine einfache Methodik gefunden wurde, wie ich Wörter ohne Benutzerinput als Schlagwort feststelle. Auch eine Art Black- & Whitelist von Wörtern hätte zuerst definiert und gewartet werden müssen, bevor Schlagwörter automatisch generiert werden hätten können. Zusätzlich hätte die Doppeldeutigkeit von Wörtern Probleme bereitet.

Mein gewählter Kompromiss war also, dass ich dem Gutachtenerzeuger die Auswahl überlasse, welche Wörter er „beschlagworten“ möchte. Er beschreibt dann diese Wörter und legt sie global in der Datenbank ab. Falls er dieses Wort dann noch einmal in einem neuen Gutachten verwenden möchte, genügt es dieses nur zu markieren, die Definition ist ja bereits im System vorhanden.

Aus der Aufgabenstellung wurde berücksichtigt:

- Vorlagen
- Automatisches Glossar
- Protokollierung der Vorgänge

Wird der Bericht mit einem Schlagwortverzeichnis erstellt so wird auch dieses im Inhaltsverzeichnis dargestellt (Abbildung 109) und in einem eigenen Kapitel sind diese Schlagwörter mit ihrer Beschreibung aufgelistet.

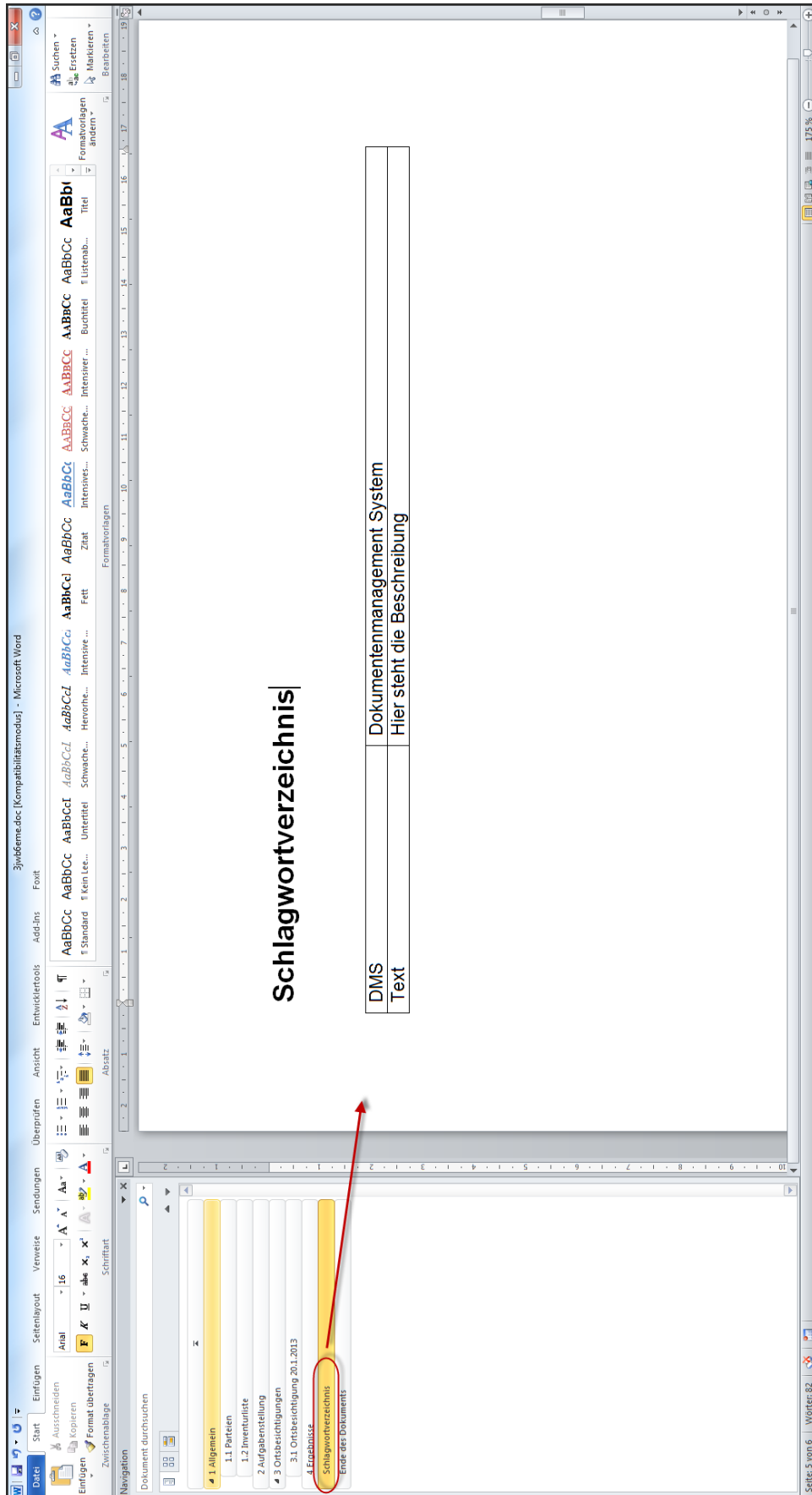


Abbildung 109: Gutachtenbericht mit Schlagwortverzeichnis

## 6.5 Zeitaufzeichnung

Ein weiterer Punkt der Umsetzung war die Zeitaufzeichnung, die direkten Einfluss auf die Rechnungslegung hat. Wie bereits im Kapitel „Gutachtenprojekt erzeugen“ erwähnt, muss zuerst ein Projekt auf „Zeitaufzeichnungsrelevant“ gesetzt werden, erst dann ist es möglich dass auf ein Projekt gebucht wird. Es existiert auch ein Feld „Pauschalbetrag“ auf dem Projekt im Reiter „Controlling“, welches dafür verwendet werden kann falls ein Pauschalbetrag zu einem Gutachten vereinbart wurden. Über dieses Feld kann kontrolliert werden ob die gebuchten Stunden und Auslagenersatzeinträge diesen Pauschalbetrag nicht übersteigen.

Um Zeiten zu verbuchen, wechselt man in die Ansicht „EXPERTISE\_Time“ in dem man in der Hauptansicht „EXPERTISE\_Main“ einen Doppelklick auf den Knoten „Zeiterfassung“ macht.

Dabei wurden sowohl „Zeitaufzeichnungseinträge“, als auch „Auslagenersatzeinträge“ berücksichtigt. Es können mehrere Zeitaufzeichnungseinträge mit unterschiedlicher Kostenart (Reisekosten, Mühewaltung, Aktenstudium, etc.) generiert werden, in einer Tabellenansicht werden diese tageweise aufsummiert.

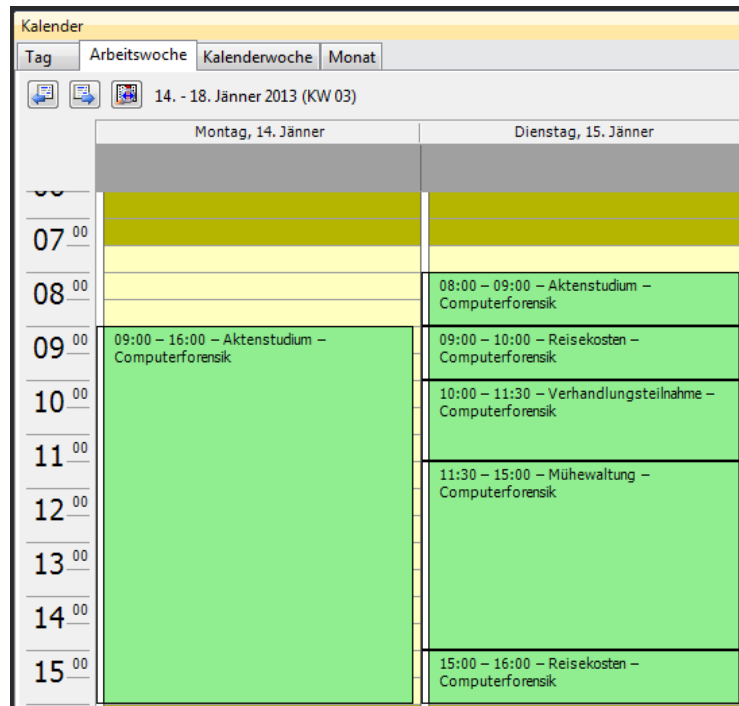


Abbildung 110: Kalendrierdialog mit Zeitaufzeichnungseinträgen

Zu sehen sind in diesem Bild (Abbildung 111) diverse Zeitaufzeichnungseinträge, wobei Einträge zum selben Tag automatisch aufsummiert und falls vorhanden Pausen abgezogen werden.

14.01.2013	Montag	09:00	16:00	7,00 h	6,50 h
15.01.2013	Dienstag	08:00	16:00	8,00 h	7,50 h

Abbildung 111: Tabellendarstellung der Zeitaufzeichnungseinträge

Einträge können entweder über die Symbolleiste, oder aber auch mittels markieren eines Zeitfensters im Kalender erzeugt (Abbildung 110) und später durch verschieben des gesamten Eintrages oder der An- und Endzeiten verändert werden. Die wichtigsten Eigenschaften eines Eintrages sieht man im nächsten Bild (Abbildung 112).

Abbildung 112: Formulardialog zu Zeitaufzeichnungseinträgen

Es können auch „Auslagenersatzeinträge“ über die Symbolleiste generiert werden. Diese Objekte müssen auch einem Projekt zugeordnet sein, damit später im „Controlling“ der Betrag auftaucht, siehe Abbildung 113.

Abbildung 113: Formulardialog zu Auslagenersatzeinträge

### 6.5.1 Zusammenfassung

Es wurde eine einfache Methode gesucht damit Zeiten zu Projekten zugeordnet werden können. Dabei war es notwendig eine eigene Ansicht mit diversen Dialogen zu erstellen, in dem Einträge definiert und zugeordnet werden konnten. Die Tabellen dienen zum aufsummieren und darstellen der einzelnen Einträge, der Kalender kann zum Erstellen dieser.

Aus der Aufgabenstellung wurde berücksichtigt:

- Zeitaufzeichnung
- Protokollierung der Vorgänge

## 6.6 Rechnung erstellen

Aus der Aufgabenstellung ging hervor, dass es auch möglich sein soll Rechnungen im System zu generieren. Rechnungen werden Geschäftspartnern zugeordnet, d.h. um Rechnungen erstellen zu können, muss man sich in der Ansicht „EXPERTISE\_Businesspartner“ befinden.

In diesem Fallbeispiel wurde für den Kunden „Familie Mustermann“ ein Privatgutachten erstellt. In der Hauptansicht wird „EXPERTISE\_Main“ gesucht und durch einen Doppelklick in die Geschäftspartneransicht gewechselt. Hier ist es möglich über das markierte Projekt im Reiter „Controlling“, die Informationen aus der Zeiterfassung abzulesen (Abbildung 114).

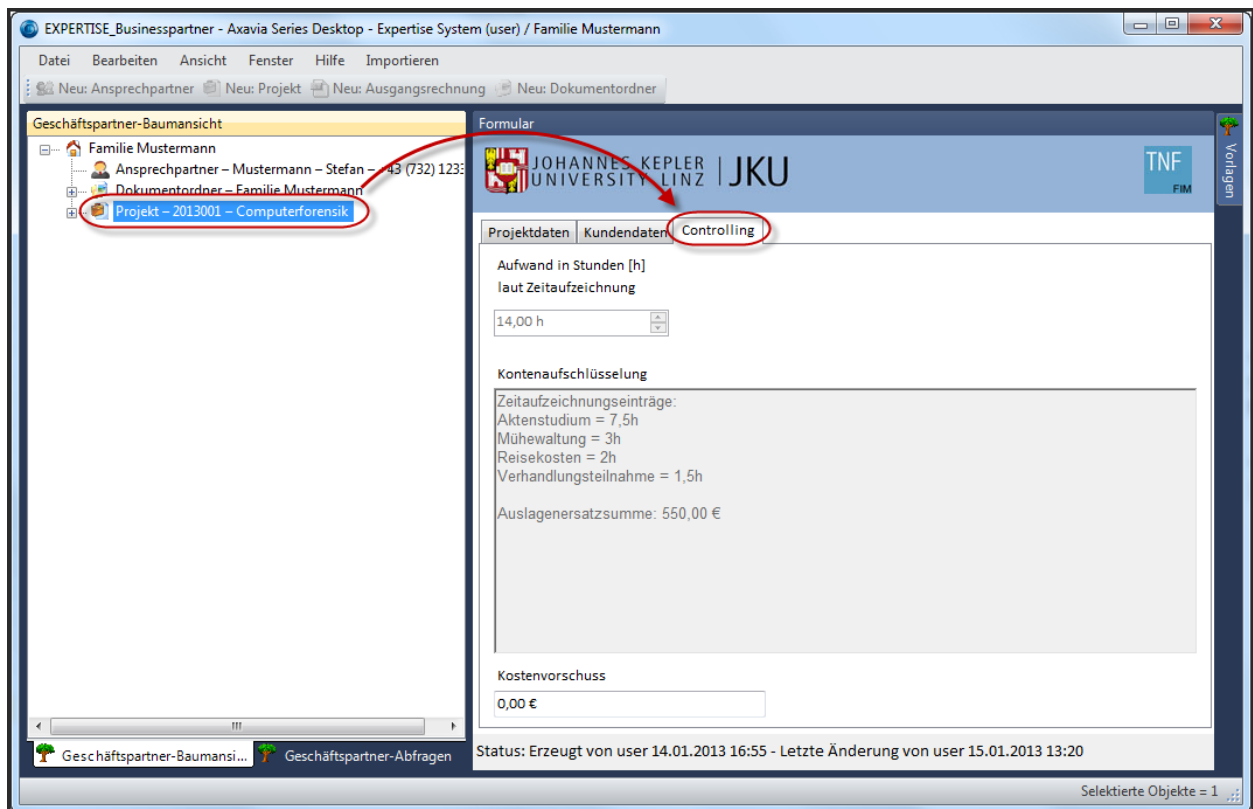


Abbildung 114: Reiter "Controlling" zu einem Projekt

Um eine Rechnung zu erstellen, kann entweder eine ohne Vorlagen erstellt werden. Dazu markiert man den Kundenknoten, hier „Familie Mustermann“ und wählt über die Symbolleiste „Neu: Ausgangsrechnung“. Es existiert auch die Möglichkeit Ausgangsrechnungen mit vordefinierten Positionen aus den Vorlagen zu übernehmen.

Zu einer Ausgangsrechnung können dann diverse Positionen verknüpft werden, dazu wählt man den Schaltknopf „Neue Rechnungsposition“ im Formulardialog, oder übernimmt wiederum einzelne Positionen aus den Vorlagen (Abbildung 115).

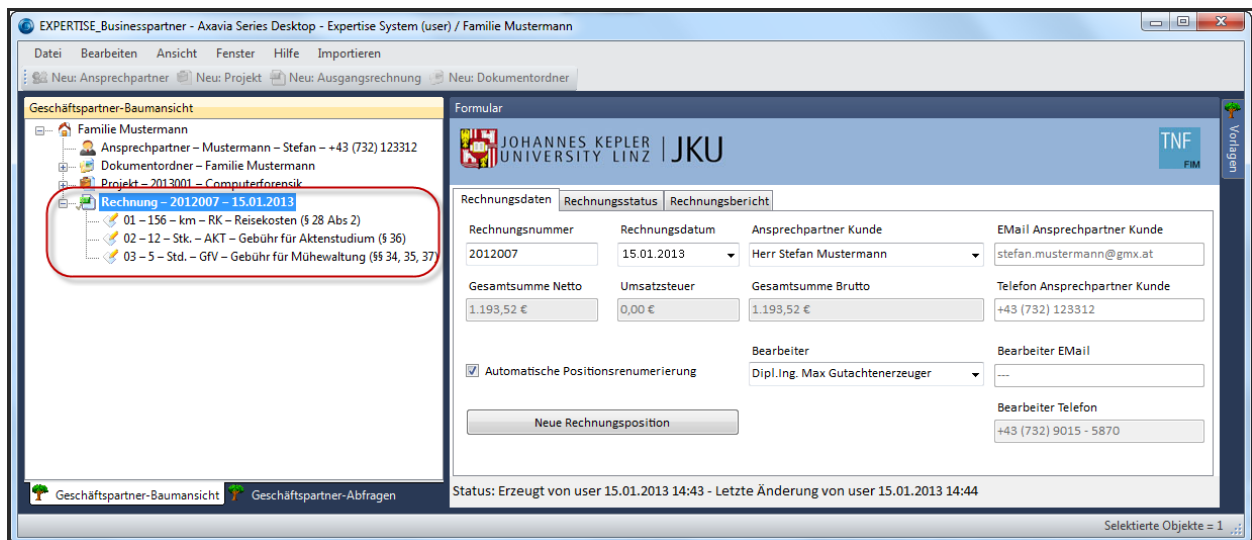


Abbildung 115: Rechnung mit Rechnungspositionen

Über die Reiter „Rechnungsstatus“ und „Rechnungsbericht“ kann angegeben werden, bis wann die Rechnung zu zahlen ist bzw. wie der Kopf- und Fußtext der Rechnung aussehen soll.

Im Rechnungsstatus ist es auch möglich das Rechnungen archiviert werden, dies ist nur möglich nachdem eine Rechnung auf „Komplett bezahlt“ gesetzt wurde. Setzt man zusätzlich das Feld „Objekt ist archiviert“, erscheint diese Rechnung das nächste Mal beim Öffnen der Geschäftspartneransicht nicht mehr. Um archivierte Rechnungen wieder anzuzeigen, kann dies über den Menüeintrag „Ansicht → Archivierte Objekte darstellen“ erledigt werden.

Sind alle Positionen eingetragen wird der Rechnungsbericht auf der Ausgangsrechnung über das Kontextmenü erzeugt und in einem Ordner ihrer Wahl abgespeichert. Diverse Suchabfragen in der Baumansicht „Geschäftspartner-Abfragen“ hilft bei der Suche nach offenen Rechnungen.

### 6.6.1 Zusammenfassung

Eine einfache Rechnungslegung wurde auch implementiert. Es ist möglich neue aber auch vordefinierte Rechnungen mit einzelnen Positionen im System zu hinterlegen. Die Information wie viele Stunden man für einzelne Tätigkeiten verbraucht hat, ist auf dem Projekt im Reiter „Controlling“ ablesbar.

Es wurden bereits, laut der Gebührenverordnung für Gutachter, diverse Vorlagepositionen in der Datenbank hinterlegt, die durch einfaches Auswählen übernommen werden. Eine Verbesserung wäre noch, wenn die gebuchten Stunden auf einem Projekten automatisch in Rechnungspositionen umgewandelt würden, jedoch müsste dazu vorher, je nach Gutachter, definiert sein wie sich die einzelnen konfigurierbaren Zeitkonten (Reisekosten, Mühewaltung, etc.) auf die diversen Gebühren verhalten sollen. Mit meiner Herangehensweise ist es jedoch möglich, dass sich Gutachter aus unterschiedlichen Branchen über das Projektcontrolling die Rechnungspositionen zwar manuell erstellen, doch mit den Daten aus der Zeitaufzeichnung füllen.

Aus der Aufgabenstellung wurde berücksichtigt:

- Vorlagen
- Rechnungslegung

- Protokollierung der Vorgänge

## 6.7 Projekt archivieren

Einer der letzten Punkte der Aufgabenstellung ist das Archivieren eines Projektes. Hat man ein Projekt abgeschlossen, sollen alle zusammengetragenen Informationen verschlüsselt und komprimiert in einer Datei abgelegt werden. Diese Datei kann dann gesondert gesichert werden.

Um ein Projekt zu Archivieren, ist es notwendig, dass es den Status „Abgeschlossen“ trägt, erst danach lässt es sich über das Kontextmenü der Eintrag „Projekt archivieren“ ausführen, siehe dazu Abbildung 116.

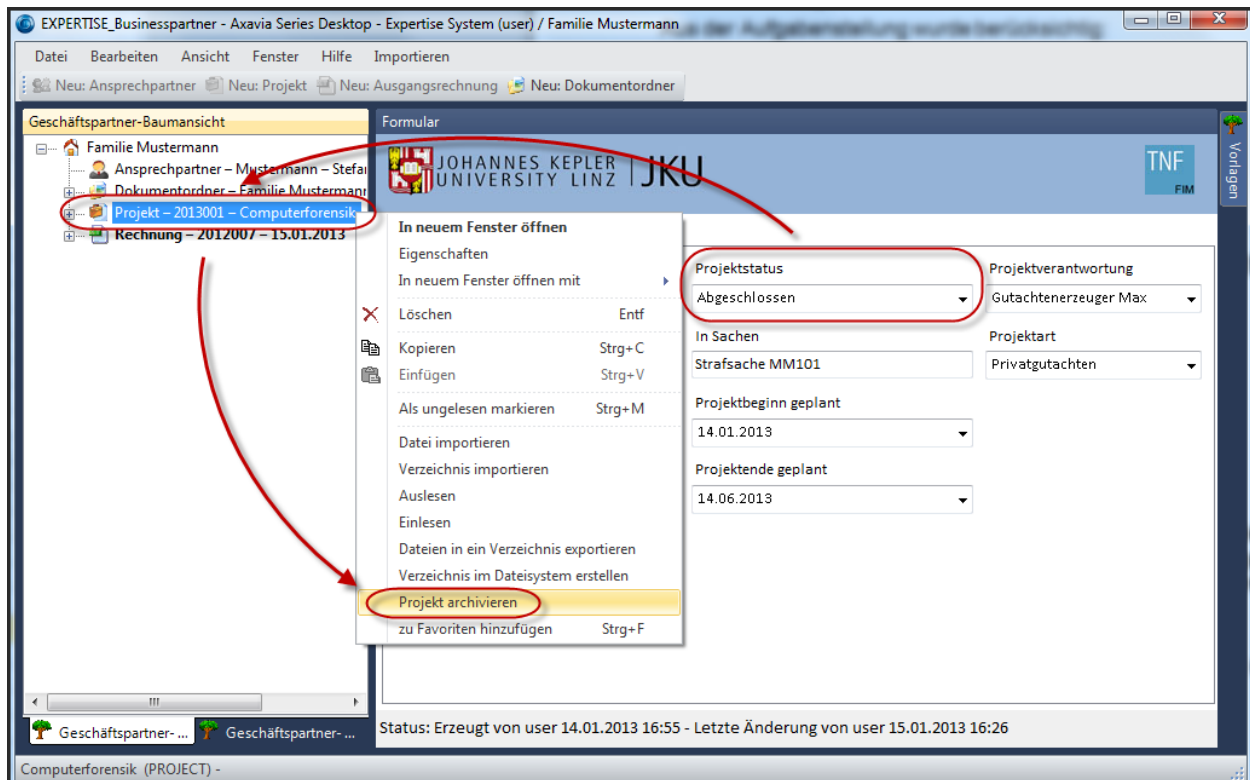


Abbildung 116: Projekt archivieren

Nach Ausführen des Befehls wird man zur Eingabe eines Passwortes aufgefordert und es wird nach dem Speicherort und Dateiname gefragt. Das Programm sucht sich alle projektrelevanten Daten und Dateien zusammen, komprimiert und verschlüsselt diese.

In der Datenbank bleibt nur ein Objekt zurück und zwar das Projekt selbst (Abbildung 117), alle Felder darauf werden schreibgeschützt. Es ist dann noch möglich zu sehen welche Projekte archiviert wurden, jedoch kann auf den Inhalt dieser nicht mehr zugegriffen werden.

Zeitaufzeichnungseinträge sind davon nicht betroffen, da diese zum verbleibenden Projekt verknüpft sind. Auch Rechnungen werden durch das Archivieren eines Projektes nicht manipuliert, da dieser direkt zu einem Kunden (Geschäftspartner) assoziiert sind.



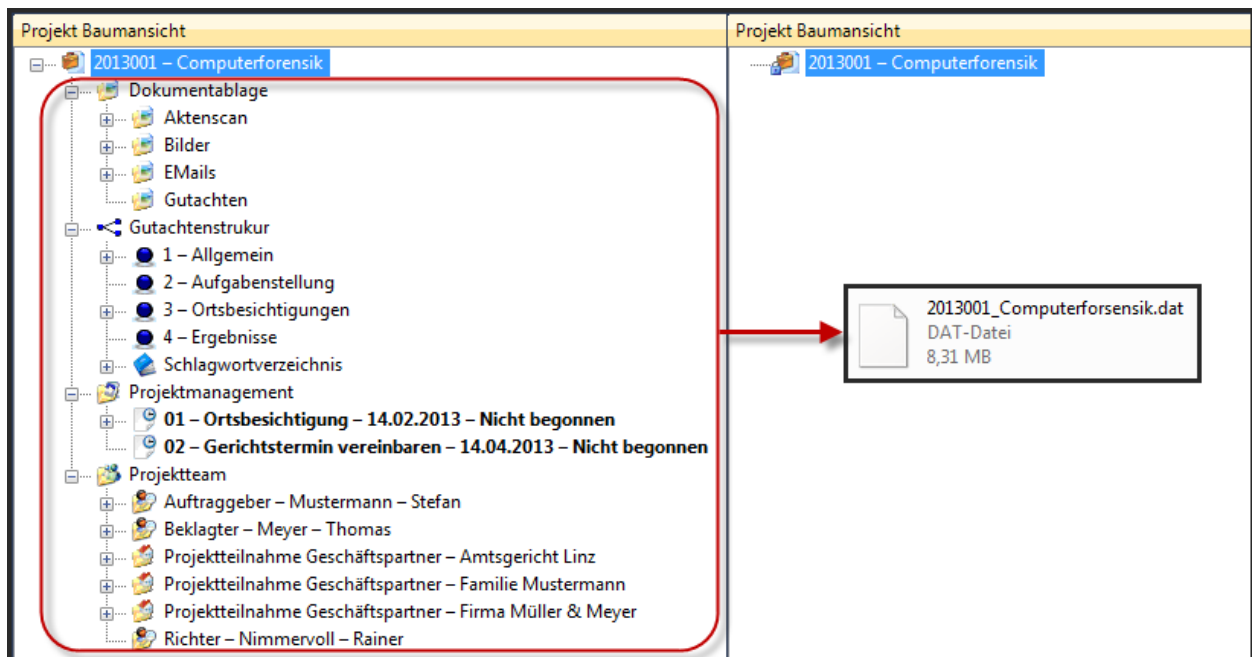


Abbildung 117: Archivierung der Objekte eines Projektes in eine Datei

Ein Projekt kann wieder importiert werden, in dem der umgekehrte Weg durchgespielt wird. Projekt markieren und über das Kontextmenü „Archiviertes Projekt importieren“ wählen. Alle Objekte, Verzeichnisse und Dateien werden, nachdem es wieder eine Passwortabfrage gab und sichergestellt wird, dass keine Manipulation an der Datei vorgenommen wurde, zurück ins System gespielt.

Durch das Archivieren wird einerseits die Datenbankgröße auf ein Minimum gehalten und es ist einfacher eine einzelne Datei zu sichern, als immer die gesamte Datenbank.

### 6.7.1 Zusammenfassung

Das Archivieren eines Projektes war auch eine Anforderung dieser Arbeit. Dabei sollte sichergestellt werden, dass nachdem ein Projekt als Datei vorhanden ist dieses nur von der Person wieder in das System eingespielt werden kann, welche auch das Passwort dazu kennt. Das Passwort dient zum Verschlüsseln, dabei wird dem Passwort noch ein „Salt“-Wert hinzugefügt und der gesamte Ausdruck als Hashwert (SHA256) in der Datenbank hinterlegt. Der „Salt“-Wert wird vom jeweiligen Projekt abgeleitet.

Ein weiterer wichtiger Punkt war die Verschlüsselung dieser Datei, welche über den symmetrischen Blockchiffre Algorithmus AES256 (Rijndael) erreicht wurde. Vor dem Verschlüsseln wird der gesamte Inhalt eines Projektes zusammen getragen und komprimiert um den Speicherplatz zu minimieren.

Auch die Datenintegrität war ein Thema welches berücksichtigt werden musste, dafür wurde nach dem Komprimieren und Verschlüsseln ein Hashwert (SHA256) der Datei generiert und mit dem Hashwert des Passwortes zu dem verbleibenden Projekt-Objekt gespeichert.

Aus der Aufgabenstellung wurde berücksichtigt:

- Archivieren
- Protokollierung der Vorgänge

## 6.8 Webclient

Als Zusatzaufgabe wurde auch ein Webclient eingerichtet. Dafür war es nötig auf dem SQL Server auch einen IIS Server zu konfigurieren. Wie der Aufbau der Webseiten aussieht kann direkt in AXAVIASeries gesteuert werden.

Der Nutzen eines Webserver war, dass es auch möglich ist Dateien, z. B. Bilder die vor Ort bei einer Besichtigung angefertigt werden, direkt zu einem Projekt hochzuladen. Die nächsten Bilder zeigen eine Fallstudie, wie so etwas aussehen könnte und unterstützen das in der Aufgabenstellung geforderten Thema des „Server-basierten Systems“.

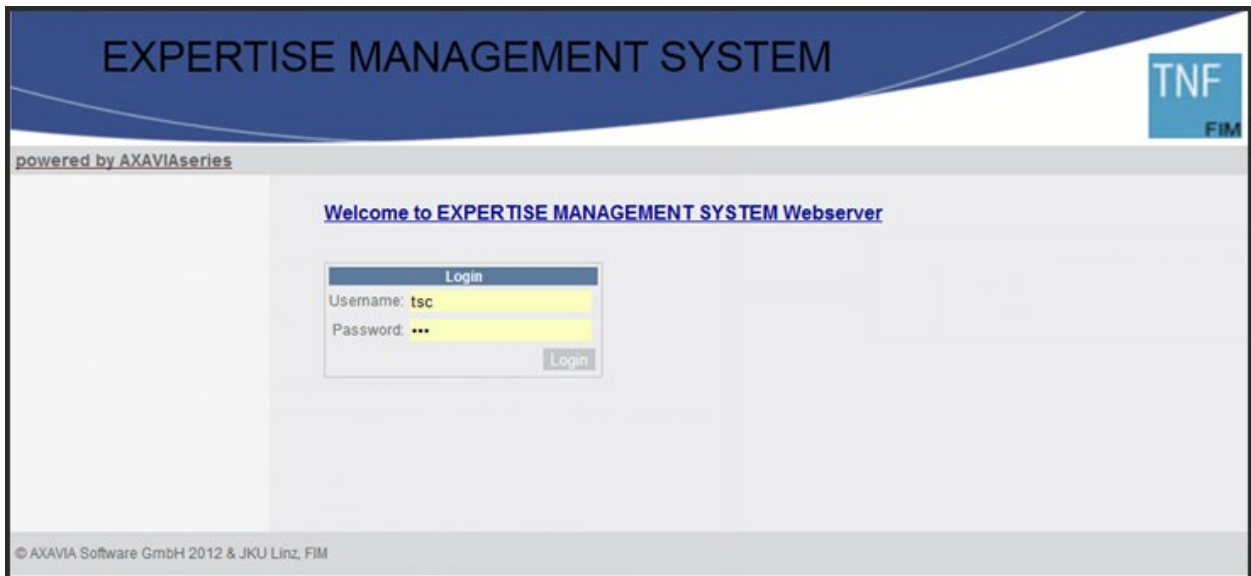


Abbildung 118: Webclient Startansicht

Hier sieht man den Willkommensbildschirm zum Einloggen in das System (Abbildung 118).



Abbildung 119: Webclient Projektauswahl

In Abbildung 119 sieht man alle Projekte, die sich im Status „In Arbeit“ befinden. Man wählt ein Projekt aus, navigiert in den Unterordner in welchen man eine Datei hochladen möchte.

Im letzten Bild (Abbildung 120) ist der Dialog zum Hochladen einer Datei, in den Ordner „Bilder“, dargestellt.



Abbildung 120: Webclient Dokument hochladen

### 6.8.1 Zusammenfassung

Diese Fallstudie soll zeigen, wie einfach es ist, nicht nur über einen Computer sondern auch über Smartphones oder Tablets Daten in das System zu integrieren. Dabei ist es egal auf welches System iOS oder Android gesetzt wird, da jedes dieser Betriebssysteme über einen integrierten Browser darauf zugreifen kann.

## 7 Implementierungsdetails

In diesem Kapitel werden einige Details zur Implementierung beantwortet. Dabei werde ich, das „Projekt“-Objekt, „Kontenaufschlüsselungs-Attribut“ und diverse Programmstücke präsentieren, alle anderen wurden aus dem System über den Modellexport in eine Datei exportiert, in der alle Änderungen auf Modellebene enthalten sind. Diese Datei wird auch auf der DVD mit ausgeliefert und kann über einen XML - Viewer geöffnet werden.

### 7.1 Objekte und Attribute

Für den Zweck dieser Arbeit wurden etwa 40 Klassen mit durchschnittlich zehn bis 20 Attributen teilweise neu erzeugt oder verändert. Jedes einzelne Attribut musste über einen Feldtypen definiert und gegebenenfalls mit Formeln zur Berechnung seines Wertes hinterlegt werden. Es war außerdem notwendig die Beziehungen zwischen den einzelnen Klassen zu generieren.

Zusätzlich wurden diverse Ereignisse implementiert, damit Änderungen zur Laufzeit Auswirkung auf Objekte, Verknüpfungen und Attribute haben.

#### 7.1.1 Klasse „Projekt“

Ein Problem stellte sich heraus, wie alle Informationen zu einem Gutachten organisiert werden sollten. Die Lösung dazu war, dass ein Objekt als „Informationsträger“ dienen sollte. Diese Klasse „PROJECT“ (Abbildung 121) dient zum Zusammenfassen aller für ein Gutachten notwendigen Informationen. Das heißt alle Objekte sind direkt oder indirekt, über Strukturierungselemente, mit dieser Klasse verbunden. Das Endergebnis ist in Abbildung 103 zu sehen.

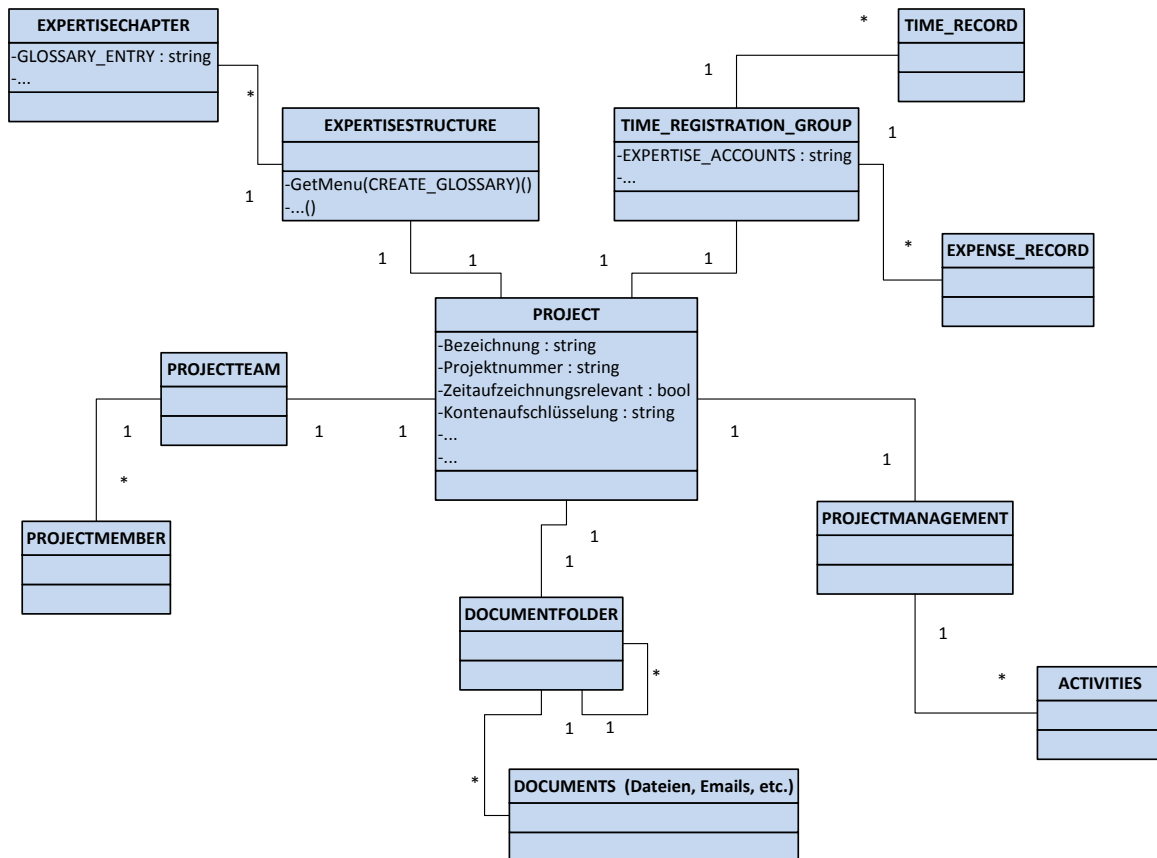


Abbildung 121: Klasse "Projekt" und Verknüpfungsdefinitionen

### 7.1.2 Attribut „Kontenaufschlüsselung“

Damit zu einem Projekt sichtbar wird, wie viele Stunden bereits verbucht wurden und die Summe aller Auslagenersatzeinträge sichtbar wird, wurde das Attribut „Kontenaufschlüsselung“ erzeugt. Der Unterschied zu statischen Attributen, die ihren Wert direkt in der Datenbank gespeichert haben, ist, dass der Inhalt dieses Attributes zur Laufzeit berechnet wird.

Der Feldtyp ist ein „String“ und es ist eine Formel hinterlegt, siehe Abbildung 122. Die Methode „GetContent()“ wird überschrieben und liefert nicht mehr den Datenbankinhalt, sondern sucht ein verknüpftes Objekt „TIME\_REGISTRATION\_GROUP“ (3) und falls vorhanden, liefert sie den Inhalt des Attributes „EXPERTISE\_ACCOUNTS“ (6).

```

1 public object GetContent()
2 {
3     DataObject timeRegGroup = this.Object.FirstSubObject("TIME_REGISTRATION_GROUP");
4
5     if (timeRegGroup != null)
6         return timeRegGroup.Values["EXPERTISE_ACCOUNTS"].Contents;
7     else
8         return System.DBNull.Value;
9 }
  
```

Abbildung 122: Formel für das Attribut "Kontenaufschlüsselung"

In Abbildung 123 sieht man die eigentliche Implementierung auf dem Objekt „TIME\_REGISTRATION\_GROUP“. Es werden alle Zeitaufzeichnungseinträge „TIME\_RECORDS“ eines Projektes nach Kontentyp zusammengesucht und die Stunden aufsummiert. Dies geschieht über die Query-Klasse (4) und in der foreach-

Schleife (8) werden die Attribute ausgelesen, gruppiert (13) und sortiert (24). Auch die Summe aller Auslagenersatzeinträge werden in das Attribut (30) geschrieben.

```

1 public object GetContent()
2
3 // TIME ENTRIES
4 Query timeQuery = new Query (new string[] { "TIME_RECORD" }, true, LinkDirectionEnum.Sub);
5 System.Collections.ArrayList timeEntries = this.Object.LinkedObjects(timeQuery);
6 Dictionary<string, double> accounts = new Dictionary<string, double>();
7
8 foreach (DataObject timeEntry in timeEntries)
9 {
10     string zeitkonto = timeEntry.Values["TIME_PERIOD_ACCOUNT_LABEL_NOEDIT"].Contents.ToString();
11     double zeitaufwand = (double)timeEntry.Values["TIME_PERIOD_WORKING_HOURS"].GetContents(0.0);
12
13     if (accounts.ContainsKey(zeitkonto))
14     {
15         double old_zeitaufwand = (double)accounts[zeitkonto];
16         zeitaufwand += old_zeitaufwand;
17         accounts[zeitkonto] = zeitaufwand;
18     }
19     else
20         accounts.Add(zeitkonto, zeitaufwand);
21 }
22
23 // Order By either KEY/VALUE
24 accounts = accounts.OrderBy(x => x.Key).ToDictionary(x => x.Key, x => x.Value);
25 string returnAccounts = "Zeitaufzeichnungseinträge:\n";
26 foreach (string key in accounts.Keys)
27     returnAccounts += key+" = "+accounts[key].ToString()+"h\n";
28
29 // EXPENSE ENTRIES
30 if ((double)((MultiValue)this.Object.Values["SUM_EXPENSES"]).GetContents(0.0) > 0)
31     returnAccounts += "\nAuslagenersatzsumme: "+ this.Object.Values["SUM_EXPENSES"].DisplayContents();
32
33 return returnAccounts;
34

```

Abbildung 123: Beispielimplementierung der Kontenaufschlüsselung

## 7.2 Ansichten & Formulare

Insgesamt wurden fünf Ansichten mit je vier bis fünf Dialogen konfiguriert, wobei über die Berechtigung definiert wurde, wer in diesen Ansichten arbeiten darf. Dabei musste gelöst werden, wie die einzelnen Ansichten mit dem Prozess eines Gutachtens zusammenspielen. Für viele dieser Lösungen genügte es einen Baum-, Suchtabellen- & Formulardialog in einer Ansicht zu konfigurieren. Als Beispiel dient uns wieder das Projekt-Objekt, das über eine eigene Ansicht verfügt (Abbildung 94). Folgende Dialoge wurden dabei verwendet.

### 7.2.1 Projekt Baumansicht

In diesem Dialog sind alle Daten zusammengefasst die zur Abwicklung eines Gutachtens notwendig sind, siehe Abbildung 103.

### 7.2.2 Projekt Abfragen

Hierbei handelt es sich um einen eigenen Baumdialog, der mit vordefinierten Abfragen bestückt ist. In Abbildung 124 sind diverse Abfragen zu Aktivitäten und Dokumenten zu sehen.

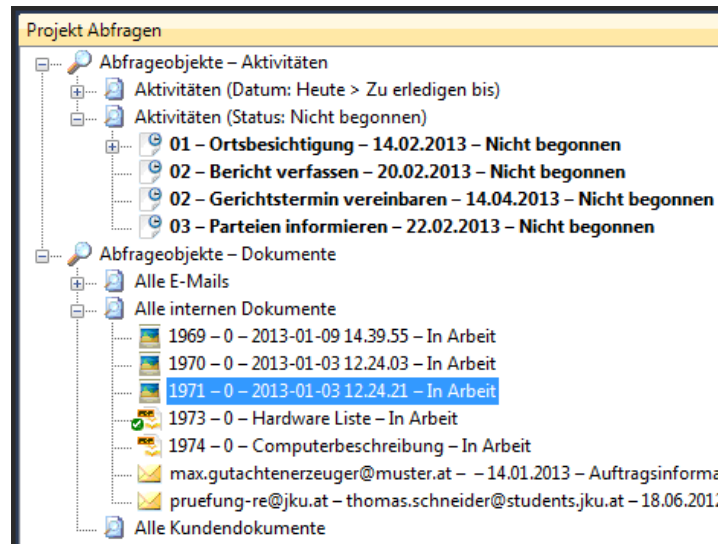


Abbildung 124: Projektabfragedialog

### 7.2.3 Formulardialog

Durch die Markierung der Objekte in den vorher beschriebenen Baumdialogen, werden unterschiedliche Formulare in diesem Dialog aktiv und zeigen deren Attribute an, als Beispiel siehe Abbildung 75.

### 7.2.4 Voransichtsdialog

Es besteht die Möglichkeit sich eine Voransicht von diversen Dokumententypen (Bilder, Adobe Acrobat Reader) anzeigen zu lassen (Abbildung 125).

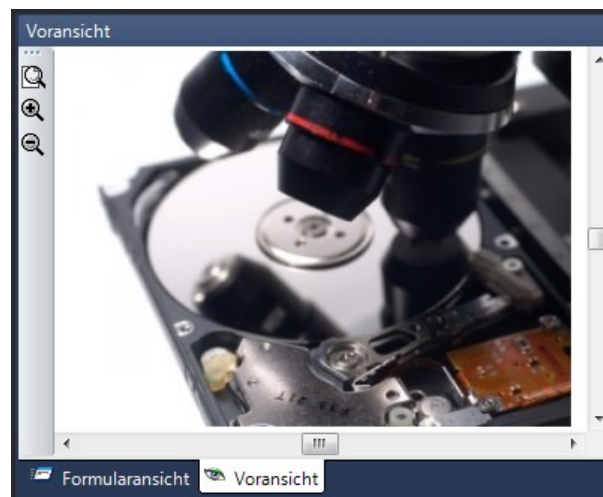


Abbildung 125: Voransichtsdialog

### 7.2.5 Vorlagendialog

Über den Vorlagendialog (Abbildung 126) können mittels Drag & Drop bereits vordefinierte Objekte in das Projekt kopiert werden.

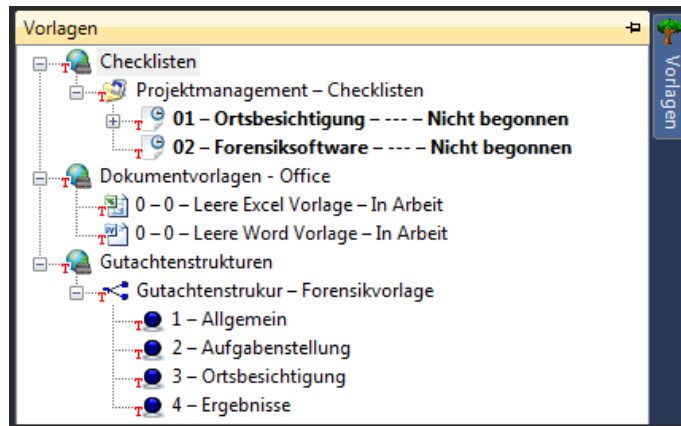


Abbildung 126: Vorlagedialog

## 7.2.6 Formular-Designer Objekt „Projekt“

Im Formulardialog – Designer wurden zirka 40 Formulare bearbeitet, damit jede Klasse ihre Attribute darstellen kann.

Da es auf einigen Objekten sehr viele Attribute gibt, war es notwendig diese logisch zu gruppieren, damit das Formular übersichtlich blieb. Das Gruppieren wurde über die Einführung von „Reiter“ gelöst. Am Projekt wurden drei Reiter (Projektdaten, Kundendaten und Controlling) erzeugt und deren zugehörige Attribute eingefügt.

Im nächsten Bild (Abbildung 127) sieht man wie alle relevanten Projektdaten eines Projektes in einem Reiter zusammengefasst werden. Es wird das Attribut „In Sachen“ in den Reiter „Projektdaten“ geschoben und das Aussehen noch angepasst.

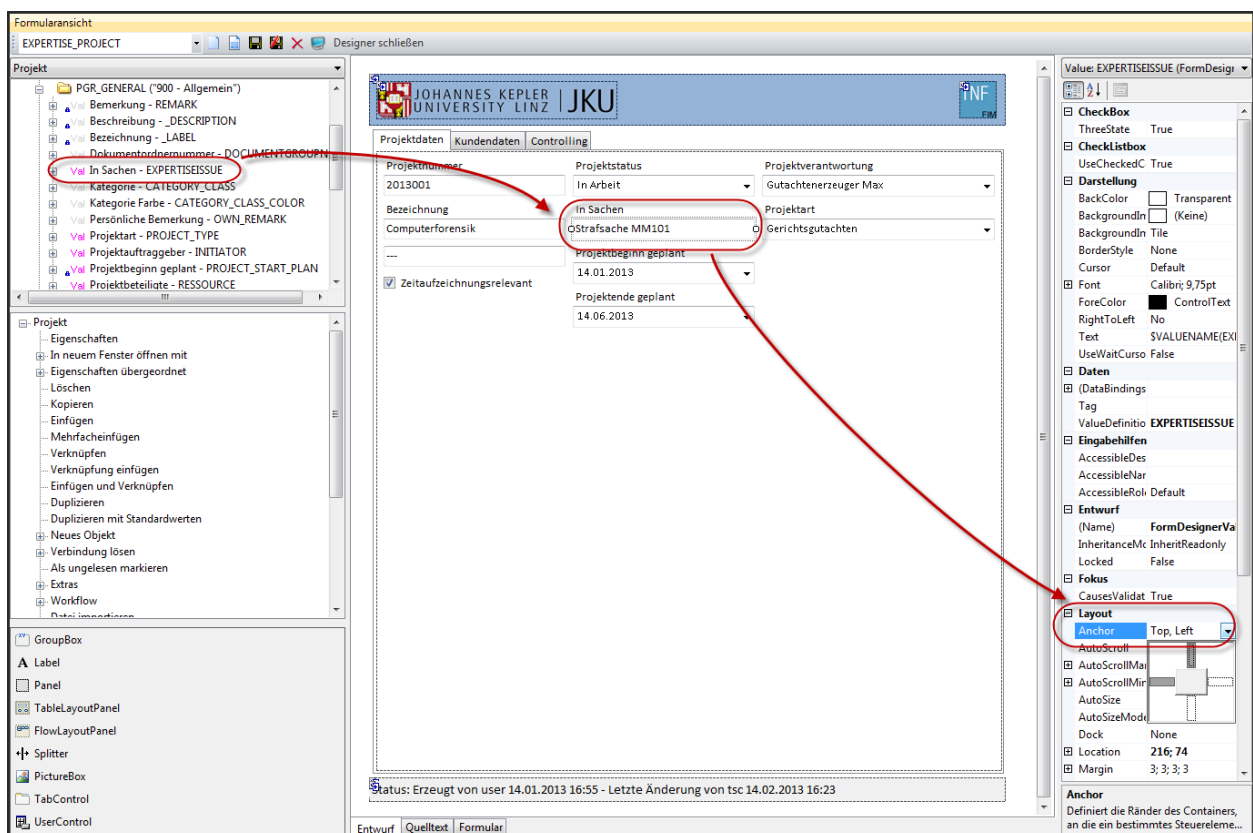


Abbildung 127: Formulardesigner für Objekt "Projekt"

## 7.3 Ereignisimplementierung

Es besteht auch die Möglichkeit auf Ereignisse zu reagieren. Beispiele dafür wären:

- Attributinhalt ändert sich
- Objekte werden erstellt oder gelöscht
- Objekte werden miteinander verknüpft
- Kontextmenüerweiterung

Am Beispiel zum Erstellen des Schlagwortverzeichnisses wird das Kontextmenü erweitert.

### 7.3.1 GetMenu „Schlagwortverzeichnis erstellen“

Hierfür wurde der Texteditor, der als Steuerungselement auf Attributen ausgewählt werden kann, um diverse Menüs und Operationen erweitert, damit Schlagwörter definiert und auch sichtbar gemacht werden können. Die Schlagwörter zu den Kapiteln wurden in der Datenbank global abgelegt. Diverse Ereignisse steuern dann das Erzeugen oder Updaten von den Schlagwörtern und schaffen eine Struktur in der diese gegebenenfalls bearbeitet werden. Siehe dazu Bedienungsanleitung EXPERTISE\_Project.

Das nächste Programmstück (Abbildung 128) zeigt die Implementierung des Kontextmenüs „Schlagwortverzeichnis erstellen / updaten“ das auf dem Objekt „Gutachtenstruktur“ aufrufbar ist. Dabei werden alle Gutachtenkapitel nach Schlagwörtern durchsucht und jene die es bereits global in der Datenbank mit ihrer, falls vorhandenen, Definitionen zur Gutachtenstruktur abgelegt.



```

1 public void GetMenu_75092CD9DCEE4E7(List<DataObject> dataObjs, Develop.Content.ContentServer.Menu.Menu menu)
2 {
3     Menu.MenuItem item = new Menu.MenuItem("EXPERTISE_GLOSSAR");
4     item.Label = this.Translate(item.ItemName);
5     menu.Add(item);
6
7     if (dataObjs.Count != 1)
8         return;
9
10    item.OnExecuteWithItem += delegate(Menu.MenuItem mItem)
11    {
12        // get all glossaryentries for this expertise, store keyword and Expertise-DataObjects where it was found
13        ArrayList expertiseChapters = dataObjs[0].LinkedObjects(new Query(new string[]{"EXPERTISECHAPTER"}, false), true);
14        Dictionary<string, ArrayList> expertiseGlossaryEntries = new Dictionary<string, ArrayList>();
15
16        // ArrayList glossaryEntries = new ArrayList();
17        char[] delimiters = new char[] {'|'};
18        foreach (DataObject expertiseChapter in expertiseChapters)
19        {
20            Value glossaryEntryValue = expertiseChapter.Values["OBJECTDIRECTORYKEYWORDS"];
21            if (glossaryEntryValue != null)
22            {
23                string[] glossaryString = glossaryEntryValue.Contents.ToString().Split(delimiters, StringSplitOptions.RemoveEmptyEntries);
24                if (glossaryString.Length != 0)
25                {
26                    // glossaryEntries.AddRange(glossaryString); //OBS
27                    foreach (string glossaryToken in glossaryString)
28                    {
29                        if (!expertiseGlossaryEntries.ContainsKey(glossaryToken))
30                        {
31                            ArrayList tempList = new ArrayList();
32                            tempList.Add(expertiseChapter);
33                            expertiseGlossaryEntries.Add(glossaryToken, tempList);
34                        }
35                        else
36                            expertiseGlossaryEntries[glossaryToken].Add(expertiseChapter);
37                    }
38                }
39            }
40        }
41        //if (glossaryEntries.Count == 0) //OBS
42        //    return;
43        if (expertiseGlossaryEntries.Count == 0)
44            return;
45
46        // create GlossaryStructure for Expertise if not present.
47        DataObject glossaryStructure = dataObjs[0].FirstSubObject("GLOSSARYSTRUCTURE");
48        if (glossaryStructure == null)
49        {
50            ObjectDefinition glossaryStructureDefinition = Application.App.GetObjectDefinition("GLOSSARYSTRUCTURE");
51            glossaryStructure = glossaryStructureDefinition.DataObjects.Add(dataObjs[0].ClusterNr);
52            glossaryStructure.CreateSuperLinkTo(dataObjs[0]);
53        }
54
55        // populate my global globalObjectDirectoryEntries
56        ArrayList globalObjectDirectoryNodes = Application.App.GetDataObjects(new Query(new string[]{"OBJECTDIRECTORYNODE"}, false, LinkDirectionEnum.Any));
57        Dictionary<string, DataObject> globalObjectDirectoryEntries = new Dictionary<string, DataObject>();
58        foreach (DataObject globalObjectDirectoryNode in globalObjectDirectoryNodes)
59        {
60            string key = globalObjectDirectoryNode.Values["NODE_LABEL"].Contents.ToString().ToLower();
61            if (!globalObjectDirectoryEntries.ContainsKey(key))
62                globalObjectDirectoryEntries[key] = globalObjectDirectoryNode;
63        }
64
65        // grab all existing ones and make a existingGlossaryDictionary
66        ArrayList existingGlossaryEntries = glossaryStructure.LinkedObjects(new Query(new string[]{"GLOSSARYENTRY"}, false, LinkDirectionEnum.Sub));
67        Dictionary<string, DataObject> existingGlossaryDictionary = new Dictionary<string, DataObject>();
68        foreach (DataObject existingGlossaryEntry in existingGlossaryEntries)
69        {
70            string key = ((MultiValue)existingGlossaryEntry.Values["_LABEL"])[0].ToString();
71            if (!existingGlossaryDictionary.ContainsKey(key))
72                existingGlossaryDictionary[key] = existingGlossaryEntry;
73        }
74
75        foreach (KeyValuePair<string, ArrayList> kvp in expertiseGlossaryEntries)
76        {
77            string keyToAdd = kvp.Key;
78            ArrayList expChapters = kvp.Value;
79
80            if (existingGlossaryDictionary.ContainsKey(keyToAdd))
81                continue;
82
83            DataObject glossaryEntryObject = Application.App.GetObjectDefinition("GLOSSARYENTRY").DataObjects.Add(dataObjs[0].ClusterNr);
84            glossaryEntryObject.CreateSuperLinkTo(glossaryStructure);
85            ((MultiValue)glossaryEntryObject.Values["_LABEL"])[0] = keyToAdd;
86            if (globalObjectDirectoryEntries.ContainsKey(keyToAdd.ToLower()))
87                glossaryEntryObject.Values["_DESCRIPTION"].CopyContents(globalObjectDirectoryEntries[keyToAdd.ToLower()].Values["_DESCRIPTION"]);
88        }
89    }
90 }

```

Abbildung 128: Beispielimplementierung zum Schlagwortverzeichnis

Zuerst wird das Kontextmenü erweitert (3), danach wird jedes Kapitel auf Schlagwörter untersucht und in einer Liste abgespeichert (13-40). Falls Schlagwörter vorhanden sind wird zum Gutachten eine Schlagwortstruktur angelegt (46-53) und gegebenenfalls mit den bereits global gefundenen Definition ausgefüllt (55-88).

## 7.4 Softwareerweiterung

Auch das Programm AXAVIASeries musste für den Zweck dieser Arbeit erweitert werden. Einer der Aufgabenstellungen war es ein Gutachten, samt Dateien und

Informationen, verschlüsselt zu archivieren. Da ein Projekt das Wurzelobjekt darstellt, war es sinnvoll es auf dieser Ebene zu implementieren. Es wurde unter anderem eine Menüerweiterung „Projekt archivieren“ erstellt, die folgende Aufgaben übernommen hat:

1. Verschlüsselungspasswort abfragen und in der Datenbank abspeichern
2. Alle projektrelevanten Daten zusammensuchen
3. Diese Daten in eine Datei speichern
4. Datei komprimieren
5. Datei verschlüsseln
6. Dateisignatur in der Datenbank abspeichern

Dazu sehen wir uns im nächsten Kapitel den Punkt 5 genauer an.

### 7.4.1 Datei verschlüsseln

Das Programmstück (Abbildung 129) zeigt das Verschlüsseln einer bereits komprimierten Datei, mit dem Passwort (key) und einem „Salt“-Wert (salt), der vom Projekt abgeleitet wurde.

```
1 private void EncryptFile(string inputFile, string outputFile, string key, string salt)
2 {
3     // compressed FileStream
4     FileStream fsIn = new FileStream(inputFile, FileMode.Open, FileAccess.Read);
5     // encrypted FileStream
6     FileStream fsEncrypted = new FileStream(outputFile, FileMode.Create, FileAccess.Write);
7
8     byte[] saltInBytes = System.Text.Encoding.Default.GetBytes(salt);
9     // Derives a password from key + salt via PBKDF1-Algorithmus
10    PasswordDeriveBytes pdb = new PasswordDeriveBytes(key, saltInBytes);
11
12    RijndaelManaged RCrypted = new RijndaelManaged();
13    RCrypted.Key = pdb.GetBytes(32);
14    RCrypted.IV = pdb.GetBytes(16);
15    CryptoStream cs = new CryptoStream(fsEncrypted, RCrypted.CreateEncryptor(), CryptoStreamMode.Write);
16
17    // encrypt in chunks
18    int bufferLen = 8192;
19    byte[] buffer = new byte[bufferLen];
20    int bytesRead;
21    long lengthToGo = 0;
22    long length = fsIn.Length;
23
24    bool closeProgressWriter = false;
25    bool dummy;
26    string titleText = Application.App.TranslationServer.TranslateLabel("GUI_EncFile_TitleText");
27    string text = Application.App.TranslationServer.TranslateLabel("GUI_EncFile_Text");
28
29    try
30    {
31        do
32        {
33            // read a chunk of data from the input file
34            bytesRead = fsIn.Read(buffer, 0, bufferLen);
35
36            // Progressbar
37            lengthToGo += bytesRead;
38            if (Application.App.RegisteredIODelegates.ModalProgressWriter != null)
39            {
40                Application.App.RegisteredIODelegates.ModalProgressWriter(titleText, text, 100.0 * length / lengthToGo, false, false, out dummy);
41                closeProgressWriter = true;
42            }
43
44            // encrypt it
45            cs.Write(buffer, 0, bytesRead);
46        } while (bytesRead != 0);
47
48        cs.FlushFinalBlock();
49        cs.Close(); //fsEncrypted is closed with cs
50        fsIn.Close();
51    }
52    catch (Exception e)
53    {
54        Application.App.RegisteredIODelegates.MessageShower("Ausnahmefehler: \n"+e.Message, "Ausnahme", IODelegates.MessageInput.OK, IODelegates.MessageType.Error);
55    }
56    finally
57    {
58        if (closeProgressWriter)
59            Application.App.RegisteredIODelegates.ModalProgressWriter("", "", 100.0, false, true, out dummy);
60    }
61 }
62 }
```

Abbildung 129: Beispielimplementierung Dateiverschlüsselung

## 7.4.2 Hashmethoden

Zum Schluss (Abbildung 130) noch die beiden Hilfsmethoden „SHA256fromString“ und „SHA256fromFile“, die zum Speichern des Passwortes und zur Integritätsprüfung der Datei eingesetzt werden.

```
1 private string SHA256fromString(string inputString, string salt)
2 {
3     SHA256 sha256 = new SHA256CryptoServiceProvider();
4     byte[] textToBytes = System.Text.Encoding.Default.GetBytes(inputString+salt);
5     byte[] result = sha256.ComputeHash(textToBytes);
6
7     return System.BitConverter.ToString(result);
8 }
9
10 private string SHA256fromFile(string inputFile)
11 {
12
13     SHA256CryptoServiceProvider SHA256hasher = new SHA256CryptoServiceProvider();
14     FileStream fs = new FileStream(inputFile, FileMode.Open, FileAccess.Read);
15     byte[] byteHash = SHA256hasher.ComputeHash(fs);
16     fs.Close();
17
18     return System.BitConverter.ToString(byteHash);
19 }
```

Abbildung 130: Hashmethoden SHA256

## 8 Verbesserungen

Natürlich wurden während der Fertigstellung dieser Arbeit auch Themen sichtbar, die verbesserungswürdig sind, diese sind in den nun folgenden Kapiteln erklärt.

### 8.1 Dokumentenmanagementsystem

Derzeit werden Datei - Objekte in der Datenbank erstellt und mit Attributen versehen, der Inhalt dieser Dateien befindet sich aber immer noch außerhalb im Dateisystem, es wird also nur eine Referenz auf diese Datei erzeugt. Der Vorteil ist das alle Operationen auf diese Datei vom Betriebssystem vorgenommen werden, ein Doppelklick auf ein Worddokument aus AXAVIASeries wird über das Betriebssystem weitergeleitet, welches entscheidet mit welcher Applikation es zu öffnen ist. Der große Nachteil dieses Verfahrens ist, dass auch normale Betriebssystembenutzer, die Zugang zum Pfad dieser Datei haben, diese im Dateisystem manipulieren können, ohne das AXAVIASeries davon etwas bemerkt.

Die Lösung dafür wurde bereits in Kapitel 5.6.4 besprochen, dass ein „Vault“ eingerichtet werden könnte, in dem man nur über das System Zugriff erreicht.

### 8.2 Rechnungslegung

Ein Softwareassistent, der Rechnungspositionen aus existierenden Zeitaufzeichnungseinträgen erzeugt, könnte noch implementiert werden. Wenn diese einzelnen Positionen dann zu Zeiten zugeordnet wären, könnte im Controlling nachlesbar sein, welche Stunden bereits in Rechnung gestellt wurden.

Es wäre nötig ein neues Objekt einzuführen zum Beispiel: „Auftrag“, dieses würde dann als Kommunikation zwischen dem Projekt und den Rechnungen dienen. Man könnte über dieses Objekt dann auch das Controlling abwickeln und hätte eine Zuordnung von

Rechnungen und Projekten auf einem Auftrag. Damit wäre auch eine Möglichkeit gefunden eine Warnpflicht bei Überschreitung des Kostenrahmens einzuführen. Diese könnte den Benutzer über einen Benachrichtigungsdialog darüber informieren.

### **8.3 Archivieren**

Zurzeit wird das .NET Framework des XML-Streams verwendet um sowohl Objekte und deren Attribute als auch Dateien zu exportieren. Bei Dateien werden diese jedoch in die Datenbank eingelesen, bevor sie wieder in einen XML-Stream verpackt werden, dies kann auf 32-bit Betriebssystemen zu Speicherproblemen führen und kostet unnötig Zeit.

Die angedachte Lösung wäre nur Objekte und deren Attribute aus der Datenbank mit Hilfe des XML-Streams zu exportieren und Dateien, die bereits auf dem Dateisystem existieren als Anhang zu verknüpfen.

### **8.4 Archivbrowser**

Es kann nötig sein dass, nachdem ein Projekt archiviert wurde, nur einige Dateien aus der Archivdatei extrahiert werden sollen. Derzeit ist es nur möglich das gesamte Projekt wieder zu importieren und danach können Dateien in das Dateisystem exportiert werden.

Ein Archivbrowser, welcher die Ordnerstrukturen inklusive der Dateien anzeigt und exportieren lässt wäre zweckmäßig.

## **9 Resümee**

Es wurden alle Aufgabenstellungen aus dem ersten Kapitel abgearbeitet. Beim Vergleich existierender System wurden Themen, welche überall zu finden waren, untersucht und in das eigene Programm eingearbeitet. Außerdem wurde ein Prozessablauf definiert, wie Gutachter mit solch einem System arbeiten können.

Im Kapitel Entwicklungsumgebung und Frontend AXAVIAseries zeigte ich, auf welcher Betriebssystemumgebung und mit welchen Mitteln ein bestehendes System so verändert werden kann um die Aufgabenstellung abzudecken.

Das Kapitel „Umsetzung und Arbeitsweise“ soll einen Überblick geben wie der Prozess ablaufen könnte, Gutachten über das System AXAVIAseries abzuwickeln. Natürlich kann nicht ein vorgefertigt konfiguriertes System zusammengestellt werden, da sich je nach Art des Gutachtens Vorlagen, Arbeitsschritte, etc. ändern. Es wurde aber darauf geachtet, dass diese bei Bedarf einfach, über ein eigenes Benutzerkonto mit eigenen Rechten, angepasst werden können.

Die Trennung zwischen den beiden Rollen Benutzer und Konfigurator wurde deshalb von mir so gewählt, dass der mit mehr Rechten ausgestattete Konfigurator Veränderungen und Einstellungen am System vornehmen darf. Der Benutzer davon aber nichts bemerkt und nur über einen vorgegebenen Weg den Prozess durchlaufen kann. Benutzer können zwar Vorlagen übernehmen, aber keine Werte in den Attributen dieser Vorlagen verändern, oder sogar ganze Vorlageobjekte löschen. Der Konfigurator hingegen kann im System zwar Projekte sehen, aber auch nur lesend bearbeiten. Dies bedeutet, er kann keine Änderungen an Projekten vornehmen jedoch ist es ihm möglich Gutachtenstrukturen mit deren Kapiteln, Checklisten, etc. aus Projekten in den Vorlagebereich zu kopieren. Es wird dadurch möglich, dass bei neuen Gutachten diese Vorlagen vom Benutzer in das neue Projekt kopiert werden können.

Die Software mit allen Anpassungen muss nun durch Benutzung evaluiert werden, damit Verbesserungen einfließen können. Grundsätzlich ist das System auch dafür ausgelegt, dass mehrere Gutachter damit arbeiten können und beispielsweise die Stammdaten der Geschäfts- und Ansprechpartner mit einander teilen. Aber alleine die Rechteinstellungen für Benutzer ist ein heikles Thema, da es mehrere Gutachter in dem System geben könnte, wobei zu definieren ist, ob Projekte untereinander sichtbar sein sollen oder nicht, derzeit sind sie es.

Auch die Konfiguration von Berichtvorlagen ist eine eigene Aufgabe, welche in dieser Arbeit nicht berücksichtigt wurde. Es existiert eine eigene Skriptsprache wie Werte aus der Datenbank übernommen, oder wie Abfragen definiert werden. Die von mir erstellen Berichte für Gutachten, Rechnungen und Zeiterfassung wurden so einfach als möglich gestaltet und enthalten alle notwendigen Informationen. Andere Vorlagen wie Briefvorlagen, Faxvorlagen, Aktivitäten, Zeitkonten, Rechnungspositionen, etc. können durch den Konfigurator im System neu erstellt, von bestehenden Projekten übernommen oder angepasst werden.

## 10 Literaturverzeichnis

- [1] „GEBÜHRENANSPRUCHSGESETZ (GebAG),“ [Online]. Available: <http://www.gerichts-sv.at/gebanspruch.html>. [Zugriff am August 2012].
- [2] „GutachterForm 6.0 Plus,“ [Online]. Available: <http://www.gutachterform.de/>. [Zugriff am August 2012].
- [3] „Gutachten Manager 6.0,“ [Online]. Available: <http://www.gutachterform.de/>. [Zugriff am August 2012].
- [4] „SV-OfficeManager 2.0,“ [Online]. Available: <http://www.sv-officemanager.de/>. [Zugriff am August 2012].
- [5] „GutachterSoft 2.1.8,“ [Online]. Available: <http://www.gutachtersoft.de/>. [Zugriff am August 2012].
- [6] L. D. Rajesh George, Wrox's SQL Server™ 2005 Express Edition Starter Kit, Indianapolis: Wiley Publishing, Inc., 2006.
- [7] T. Nash, Accelerated C# 2010, New York: Springer Verlag, 2010.
- [8] J. Mayo, C# 3.0 Unleashed with .net 3.5, Indianapolis: SAMS, 2008.
- [9] S. M. M. D. S. Daniele Bochicchio, ASP.NET 4.0 in Practice, Shelter Island: Manning Publications, 2011.
- [10] „AXAVIA Software GmbH,“ [Online]. Available: <http://www.axavia.com/>. [Zugriff am Oktober 2012].
- [11] „TrueCrypt,“ [Online]. Available: <http://www.truecrypt.org/>. [Zugriff am Februar 2013].

## 11 Abbildungsverzeichnis

Abbildung 1: Server-basiertes System .....	5
Abbildung 2: Dokumentenmanagementsystem .....	6
Abbildung 3: Beispiel Briefkopfinformation .....	7
Abbildung 4: Beispiel Schlagwortverzeichnis.....	8
Abbildung 5: Beispiel Zeitaufzeichnung .....	8
Abbildung 6: Beispiel Archivierung .....	10
Abbildung 7: Softwarevergleich GutachterForm 6.0 Plus .....	11
Abbildung 8: Softwarevergleich Gutachten Manager 6.0.....	13
Abbildung 9: Softwarevergleich SV-OfficeManager 2.0.....	15
Abbildung 10: Softwarevergleich GutachterSoft 2.1.8 .....	17
Abbildung 11: Prozessablauf.....	19
Abbildung 12: Prozessablauf CRM/DMS Allgemein .....	20
Abbildung 13: Prozessablauf CRM/DMS Kunde.....	20
Abbildung 14: Prozessablauf Projekt.....	21
Abbildung 15: Prozessablauf Zeitaufzeichnung.....	22
Abbildung 16: Prozessablauf Rechnungslegung .....	23
Abbildung 17: Prozessablauf Konfiguration .....	24
Abbildung 18: Entwicklungsumgebung.....	24
Abbildung 19: Relationale Datenbank Modell MS Access .....	26
Abbildung 20: Relationale Datenbank Daten MS Access .....	26
Abbildung 21: Objektorientiertes Prinzip.....	27
Abbildung 22: Instanzen von Klassen.....	27
Abbildung 23: Objektorientiertes Prinzip mit rel. Datenbank.....	28
Abbildung 24: Instanz Information zu Klassen.....	28
Abbildung 25: Überblick Frontend AXAVIAseries .....	29
Abbildung 26: Ansichtskonfiguration EXPERTISE_Configuration .....	30
Abbildung 27: Ansicht EXPERTISE_Configuration.....	30
Abbildung 28: Web Client – Konfiguration.....	31
Abbildung 29: Web Client - Ordner & Dokumente .....	31
Abbildung 30: Datenlayer Beispielansicht .....	33
Abbildung 31: Modellebene .....	34
Abbildung 32: Modellansicht der Objektdefinitionen und Attribute.....	35
Abbildung 33: Definition der Eigenschaften eines Attributes „GEARS“ .....	36
Abbildung 34: Auswahlfeld des Schaltgänge Attributes.....	36

Abbildung 35: Formel für Attribut "Radinformation" .....	37
Abbildung 36: Auswertung für Attribut "Radinformation" .....	37
Abbildung 37: Verknüpfungsdefinition .....	38
Abbildung 38: Eigenschaften einer Verknüpfungsdefinition.....	38
Abbildung 39: Ereignisse auf Objektdefinitionen .....	39
Abbildung 40: Beispiel Ereignis "BeforeCreateSubLink".....	40
Abbildung 41: Übersicht Datenebene .....	41
Abbildung 42: Beispiel Baumansicht .....	41
Abbildung 43: Beispiel Eigenschaftsdialog .....	42
Abbildung 44: Beispiel Formulardialog .....	42
Abbildung 45: Beispiel Formulardesigner .....	42
Abbildung 46: Beispiel Tabellenansicht .....	43
Abbildung 47: Beispiel Kalenderdialog .....	44
Abbildung 48: Beispiel einer Ansicht aus Dialogen.....	45
Abbildung 49: Beispiel der Kommunikation zwischen Dialoge.....	45
Abbildung 50: Beispiel Symbolleiste.....	45
Abbildung 51: Beispiel Kontextmenüeinträge Anwender und Experte .....	46
Abbildung 52: Role Based Access Control mit Objektstatus .....	47
Abbildung 53: Bereiche in AXAVIASeries .....	48
Abbildung 54: Bereichsübergreifende Verknüpfung .....	48
Abbildung 55: Ordnerberechtigungen.....	49
Abbildung 56: Standardeinstellung am Recht: Instanz lesen.....	49
Abbildung 57: Berechtigung auf Ansicht.....	50
Abbildung 58: Berechtigungsauswertung auf Ansicht.....	50
Abbildung 59: Rechte und Rollen in AXAVIASeries .....	51
Abbildung 60: Berechtigungen für Rolle Administrator .....	52
Abbildung 61: Rolle Administrator und Benutzer Gegenüberstellung .....	52
Abbildung 62: Instanz: Löschen definiert auf einem Projekt .....	53
Abbildung 63: Auswertung für einen Wert .....	55
Abbildung 64: Basisverzeichnis, Schreiben verboten auf Objektebene.....	56
Abbildung 65: Basisverzeichnis, Schreiben erlaubt auf Attributebene .....	56
Abbildung 66: Auswertung für Wertdefinition.....	57
Abbildung 67: Auswertung für Verknüpfungen .....	59
Abbildung 68: Auswertung für Verknüpfungsdefinition .....	59
Abbildung 69: Effektive Berechtigung für einen Benutzer.....	60
Abbildung 70: Objektabhängige Rollen .....	61



Abbildung 71: Dokumentenklassen in AXAVIAseries .....	65
Abbildung 72: Basisverzeichnis der Ordner und Dateien .....	65
Abbildung 73: Dateireferenz .....	66
Abbildung 74: Ein- und Ausgelesene Datei .....	67
Abbildung 75: Basiseinstellungen .....	68
Abbildung 76: Protokollierung von Vorgängen.....	68
Abbildung 77: Objektvorlagen Basiseinstellung.....	69
Abbildung 78: Objektvorlagen Bibliothek .....	69
Abbildung 79: Beispiel Anpassung der Vorlage .....	70
Abbildung 80: Globale Dokumentenstruktur .....	70
Abbildung 81: Globales Schlagwortverzeichnis .....	71
Abbildung 82: Beispiel Schlagwort "DMS" .....	71
Abbildung 83: Organisationsstruktur.....	71
Abbildung 84: Bankdatendialog .....	72
Abbildung 85: Unternehmensstandortdialog .....	72
Abbildung 86: Mitarbeiter mit Benutzerkonto verknüpfen .....	73
Abbildung 87: Übersicht der unterschiedlichen Vorlagen .....	74
Abbildung 88: Startansicht "EXPERTISE_Main" .....	75
Abbildung 89: Kundeninformation.....	76
Abbildung 90: Geschäftspartneransicht "EXPERTISE_Businesspartner" .....	77
Abbildung 91: Ansprechpartner Formulardialog .....	77
Abbildung 92: Dokumentablagestruktur zu einem Geschäftspartner.....	78
Abbildung 93: Gutachtenprojekt erzeugen .....	79
Abbildung 94: Projektinformationen eintragen .....	79
Abbildung 95: Dokumentablagestruktur zu einem Projekt .....	80
Abbildung 96: Projektteam zu einem Gutachten erstellen .....	81
Abbildung 97: Geschäftskontakte und Nachrichten zum Projektteam .....	81
Abbildung 98: Checklisten aus Vorlage erzeugen .....	82
Abbildung 99: Abfragen zu einem Projekt .....	82
Abbildung 100: Gutachtenstruktur mit Kapiteln.....	83
Abbildung 101: Titelseite eines Gutachtens .....	84
Abbildung 102: Formulardialog mit Texteditor eines Kapitels .....	85
Abbildung 103: Übersicht Objekte in einem Projekt.....	85
Abbildung 104: Gutachtenbericht .....	87
Abbildung 105: Wörter im Texteditor als Schlagwort definieren .....	88
Abbildung 106: Darstellung der Schlagwörter in einem Kapitel .....	89

Abbildung 107: Schlagwortverzeichnisstruktur .....	89
Abbildung 108: Beschreibung für ein Schlagwort .....	90
Abbildung 109: Gutachtenbericht mit Schlagwortverzeichnis .....	91
Abbildung 110: Kalenderdialog mit Zeitaufzeichnungseinträgen .....	92
Abbildung 111: Tabellendarstellung der Zeitaufzeichnungseinträge .....	92
Abbildung 112: Formulardialog zu Zeitaufzeichnungseinträgen .....	93
Abbildung 113: Formulardialog zu Auslagenersatzeinträge .....	93
Abbildung 114: Reiter "Controlling" zu einem Projekt .....	94
Abbildung 115: Rechnung mit Rechnungspositionen .....	95
Abbildung 116: Projekt archivieren .....	96
Abbildung 117: Archivierung der Objekte eines Projektes in eine Datei .....	97
Abbildung 118: Webclient Startansicht .....	98
Abbildung 119: Webclient Projektauswahl .....	98
Abbildung 120: Webclient Dokument hochladen .....	99
Abbildung 121: Klasse "Projekt" und Verknüpfungsdefinitionen .....	100
Abbildung 122: Formel für das Attribut "Kontenaufschlüsselung" .....	100
Abbildung 123: Beispielimplementierung der Kontenaufschlüsselung .....	101
Abbildung 124: Projektabfragedialog .....	102
Abbildung 125: Voransichtsdialog .....	102
Abbildung 126: Vorlagedialog .....	103
Abbildung 127: Formulardesigner für Objekt "Projekt" .....	103
Abbildung 128: Beispielimplementierung zum Schlagwortverzeichnis .....	105
Abbildung 129: Beispielimplementierung Dateiverschlüsselung .....	106
Abbildung 130: Hashmethoden SHA256 .....	107
Abbildung 131: Installation SQL Server .....	118
Abbildung 132: Funktionsauswahl SQL Server .....	118
Abbildung 133: Instanzkonfiguration SQL Server .....	119
Abbildung 134: Serverkonfiguration SQL Server .....	119
Abbildung 135: Datenbankmodulkonfiguration SQL Server .....	120
Abbildung 136: Dienste in Windows 7 .....	120
Abbildung 137: Anmeldeeigenschaften SQL Managementstudio .....	121
Abbildung 138: Serverrollen des SQL Benutzerkontos .....	121
Abbildung 139: Leere Datenbank erstellen .....	122
Abbildung 140: Vorlagedatenbank wiederherstellen .....	122
Abbildung 141: Datenbankverbindung einrichten .....	123
Abbildung 142: Verzeichnis für Datenbankverbindungsdateien .....	123

Abbildung 143: Datenbankverbindung bearbeiten.....	124
Abbildung 144: Datenbankverbindungs-Manager.....	124
Abbildung 145: ACC-Datei mit den Datenbankinformationsverbindungen .....	125
Abbildung 146: AXAVIASeries Anmeldedialog.....	125
Abbildung 147: Splashscreen und Anmeldedialog .....	126
Abbildung 148: Konfigurationsansicht "EXPERTISE_Configuration".....	127
Abbildung 149: Basiskonfiguration .....	129
Abbildung 150: Bibliothekseinstellungen .....	130
Abbildung 151: Dokumentablage.....	131
Abbildung 152: Organisationsstruktur konfigurieren .....	132
Abbildung 153: Vorlagen übernehmen aus einem Projekt.....	133
Abbildung 154: Dateien und Ordner importieren .....	134
Abbildung 155: Rechnungsbericht erstellen .....	135
Abbildung 156: Dokument speichern - Dialog.....	135
Abbildung 157: Bericht mit Startobjekt verknüpft .....	135
Abbildung 158: Dokumentrevisionen .....	136
Abbildung 159: Ansicht "EXPERTISE_Main" .....	137
Abbildung 160: Neuen Geschäftspartner erzeugen.....	138
Abbildung 161: Geschäftspartner suchen.....	139
Abbildung 162: Ansicht "EXPERTISE_Businesspartner".....	140
Abbildung 163: Geschäftspartnerdokumentorder erzeugen .....	141
Abbildung 164: Neuen Ansprechpartner erstellen .....	141
Abbildung 165: Dokument speichern - Dialog.....	142
Abbildung 166: Neuen Geschäftskontakt erzeugen.....	142
Abbildung 167: Neues Projekt erstellen.....	143
Abbildung 168: Neue Ausgangsrechnung erzeugen .....	144
Abbildung 169: Neue Rechnungsposition erzeugen .....	145
Abbildung 170: Rechnung mit Positionen .....	145
Abbildung 171: Ansicht "EXPERTISE_Project" .....	146
Abbildung 172: Dokumentordnerstruktur in einem Projekt .....	147
Abbildung 173: Dokumentenverteilerstruktur.....	147
Abbildung 174: Gutachtenstruktur .....	147
Abbildung 175: Texteditor für Kapitel.....	148
Abbildung 176: Schlagwörter definieren und anzeigen.....	149
Abbildung 177: Gutachtenstruktur mit Schlagwortverzeichnisstruktur .....	149
Abbildung 178: Projektmanagement mit Aktivitäten .....	150

Abbildung 179: Projektteam mit Parteien .....	150
Abbildung 180: Projekt verschlüsseln und Datelexport - Dialog .....	151
Abbildung 181: Ansicht "EXPERTISE_TIME" .....	152
Abbildung 182: Zeitaufzeichnungseintrag im Kalenderdialog erstellen.....	153
Abbildung 183: Zeitaufzeichnungseintrag - Eigenschaftendialog .....	153
Abbildung 184: Auslagenerstarteintrags - Dialog.....	154

## 12 Anhänge

### 12.1 Systemanforderung

Die nächsten Kapitel behandeln die Installation des Microsoft SQL Servers und der AXAVIASeries Applikation.

Es wird davon ausgegangen das noch keine, der beiden notwendigen Softwares auf Ihrem System installiert ist. Als Betriebssystem dient Microsoft Windows 7 Professional (64bit).

Betriebssystem:	Microsoft Windows 7 (32 & 64bit) Microsoft Windows XP (32 & 54bit)
Software:	Microsoft SQL Server 2008R2 Microsoft Office 2007 bis 2010 AXAVIASeries 6.6 Build 45xx .NET Framework 4.0
Speicherplatz:	500MB

### 12.2 Installationsanleitung

#### 12.2.1 Installation Microsoft SQL Server 2008R2 Express

Der SQL Server von Microsoft kann in verschiedene Versionen erworben werden, nur eine davon ist kostenfrei und zwar die „Express“ Edition. Unter folgender URL kann diese Version mit der SQL Management Erweiterung bezogen werden:

<http://www.microsoft.com/de-de/download/details.aspx?id=23650>

Folgende Einschränkungen müssen in Kauf genommen werden:

- Nur eine CPU bzw. ein Kern wird genutzt
- Max 1GB Arbeitsspeicher wird reserviert
- Datenbankgröße ist max. 10GB

Eine Upgrade-Funktionalität ist aber gegeben und man kann ohne Probleme auf die Standard bzw. Enterprise umsteigen.

Für unsere Zwecke ist die Version mit den Verwaltungswerkzeugen zu verwenden, je nachdem die 32bit oder die 64bit Version. Nach Auswählen von „Neuinstallation“ und Zustimmung zu den Lizenzbedingungen wird die Installation gestartet (Abbildung 131).

Falls Probleme bei der Installation auftreten dann besuchen sie diese Webseite:

<http://social.msdn.microsoft.com/Forums/de-DE/sqlserverde/thread/96df45fe-0863-492f-999c-f58813d2b60f>

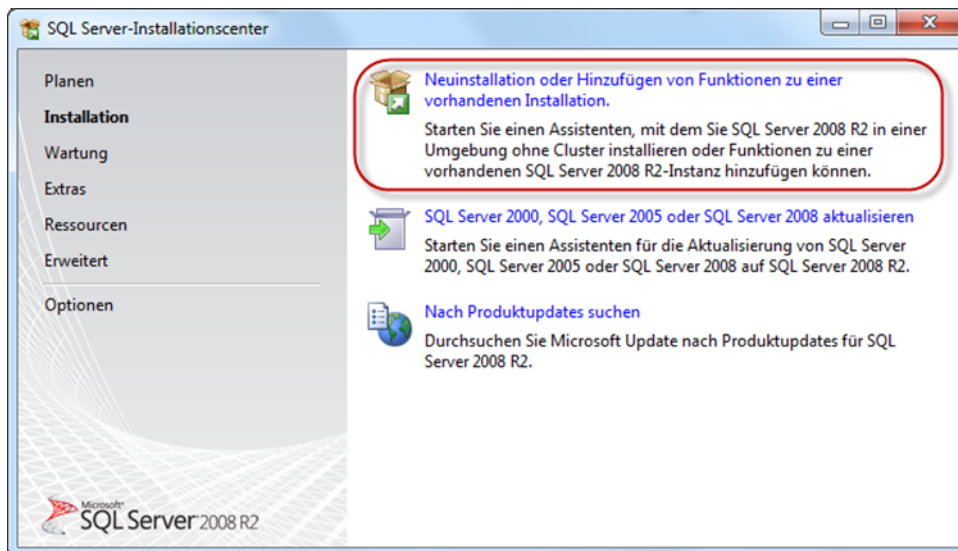


Abbildung 131: Installation SQL Server

Folgende Kernfunktionen (Abbildung 132) müssen installiert werden:

- Datenbankmoduldienste
- Verwaltungstools - Einfach

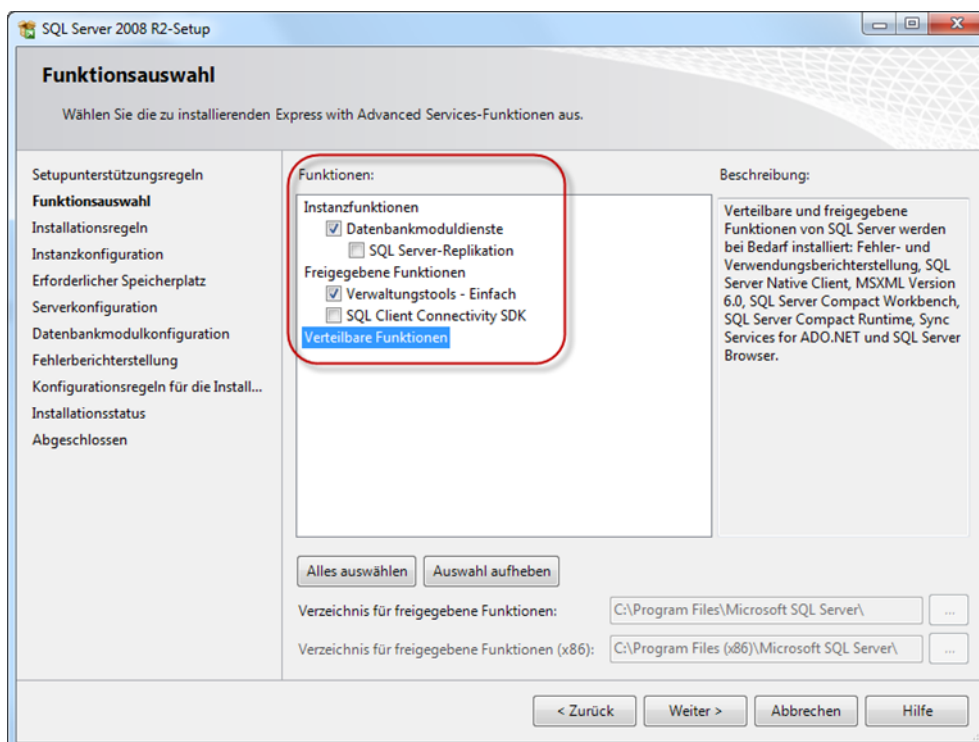


Abbildung 132: Funktionsauswahl SQL Server

Bei der Instanz Konfiguration (Abbildung 133) wird „Standardinstanz“ ausgewählt. Sollte bereits eine oder mehrere SQL Instanzen vorhanden sein, wählt man hier eine „Benannte Instanz“ und gibt dieser einen neuen Namen.

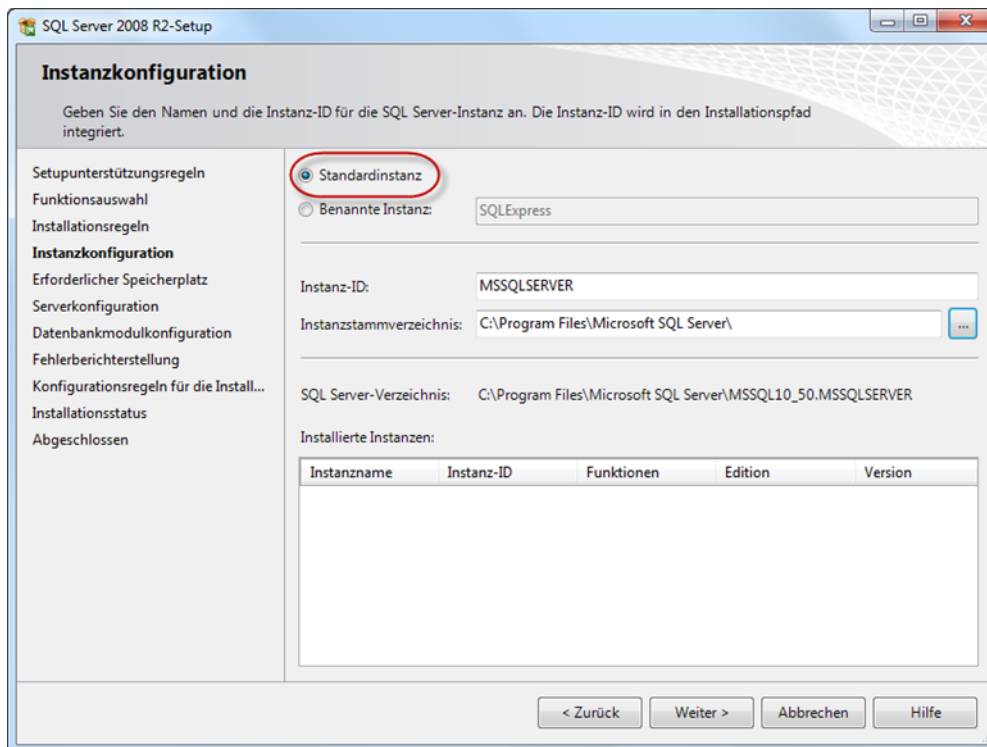


Abbildung 133: Instanzkonfiguration SQL Server

Bei der Serverkonfiguration (Abbildung 134) ist es dem Benutzer überlassen, ob der „SQL Server Database Engine“ Dienst automatisch oder manuell gestartet wird. Der „SQL Server Browser“ wird nur benötigt falls mehrere SQL Server Instanzen installiert wurden, da dieser koordiniert zu welchem er die Clients verbinden soll.

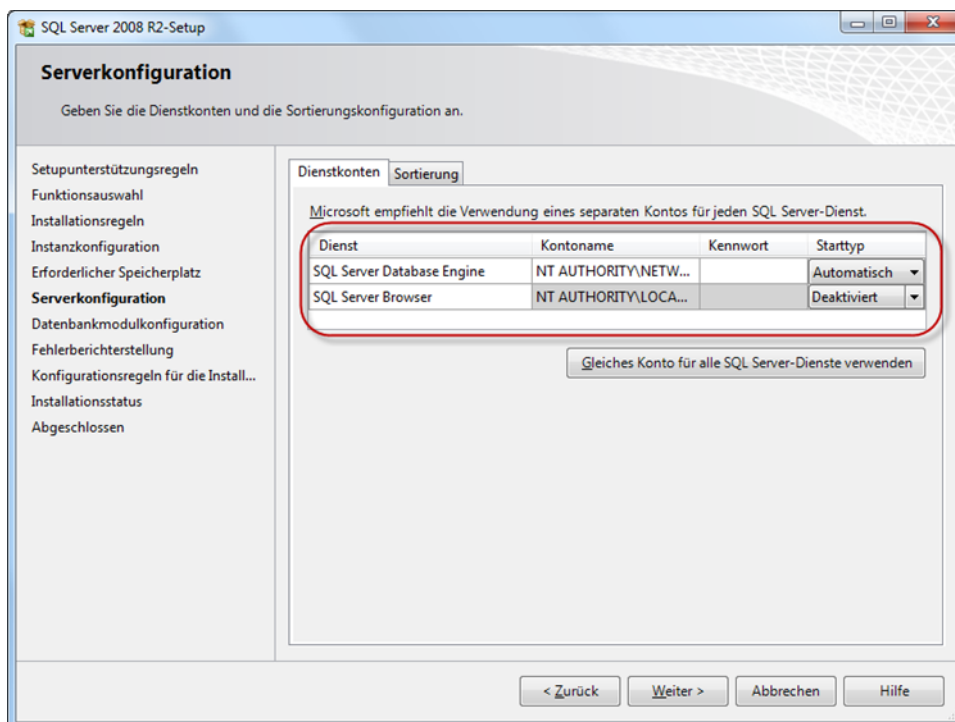


Abbildung 134: Serverkonfiguration SQL Server

Bei der Datenbankmodulkonfiguration (Abbildung 135) den „Gemischten Modus“ auswählen und ein Passwort vergeben.

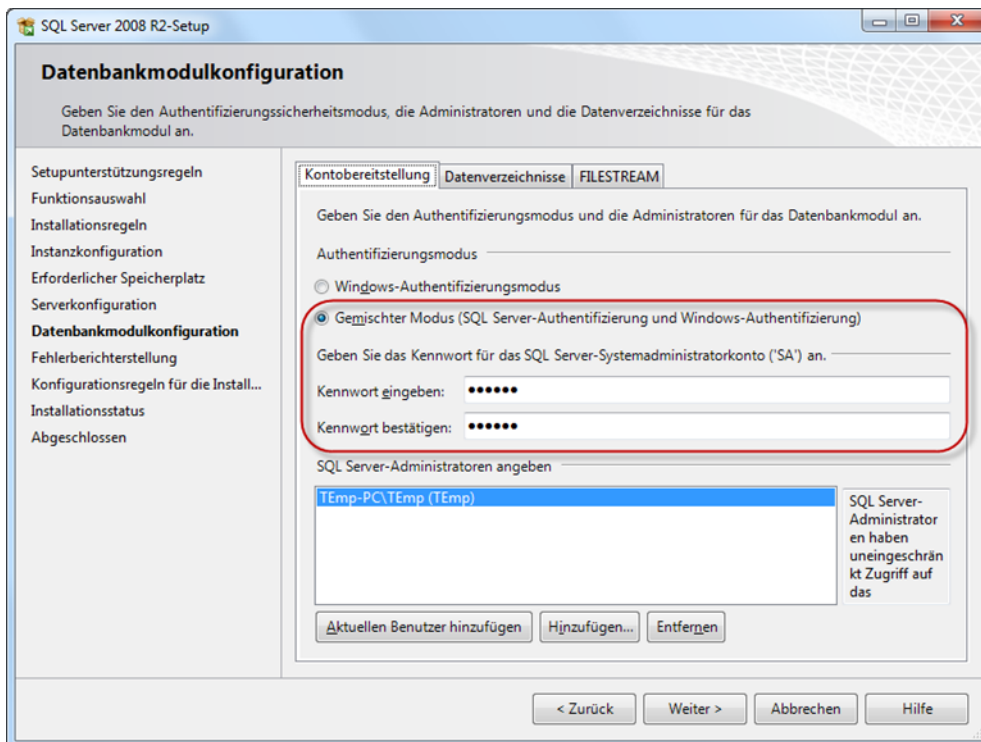


Abbildung 135: Datenbankmodulkonfiguration SQL Server

Je nachdem ob die Datenbank automatisch oder manuell gestartet wurde, findet man nun deren Status in der Dienste-Übersicht von Windows 7, siehe Abbildung 136.

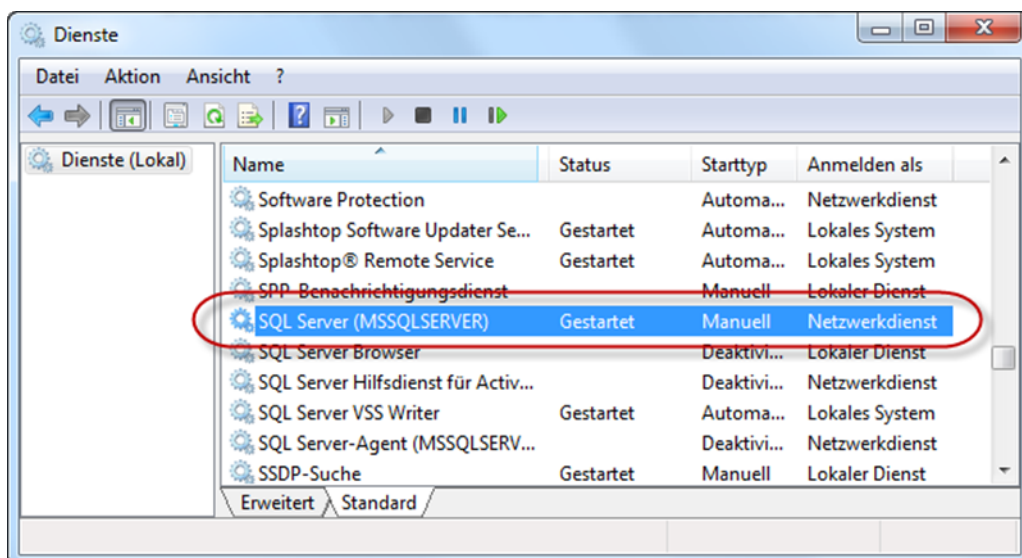


Abbildung 136: Dienste in Windows 7

### 12.2.2 Konfiguration Microsoft SQL Server 2008R2 Express

Nach der Installation des SQL Server, sollte aus sicherheitstechnischen Gründen noch ein expliziter Benutzer angelegt werden, welcher mit dem Server kommunizieren kann.

Dazu wird das „Microsoft Server Management Studio“ gestartet, dass mit dem Server mitinstalliert wurde und mit dem Datenbankserver verbunden.

Unter Sicherheit → Anmeldungen → Kontextmenü (Neue Anmeldung) erstellt man einen neuen Benutzer (Abbildung 137).



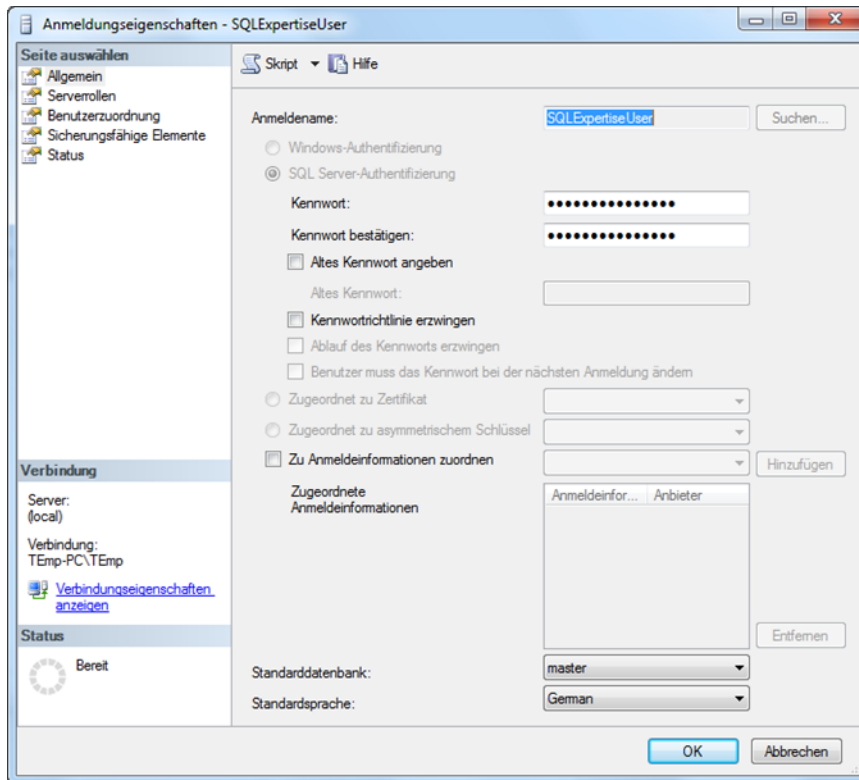


Abbildung 137: Anmeldeeigenschaften SQL Managementstudio

Als Serverrolle gibt man diesem Benutzer das Recht, dass er Datenbanken erstellen kann (Abbildung 138). Mit dieser SQL-Rolle bekommt der Benutzer auf Datenbanken, die von ihm erstellt wurde, alle Rechte die notwendig sind.

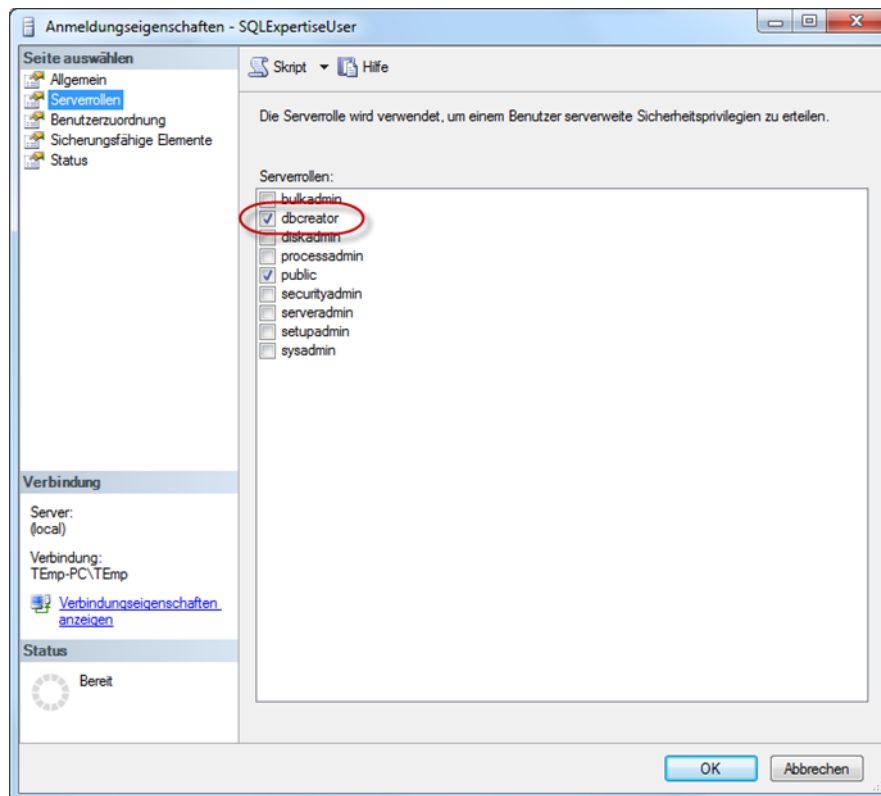


Abbildung 138: Serverrollen des SQL Benutzerkontos

### 12.2.3 Importieren der Vorlagedatenbank

Damit sofort mit der Arbeit begonnen werden kann, wird zuerst eine leere Datenbank am SQL Server erzeugt, siehe Abbildung 139.

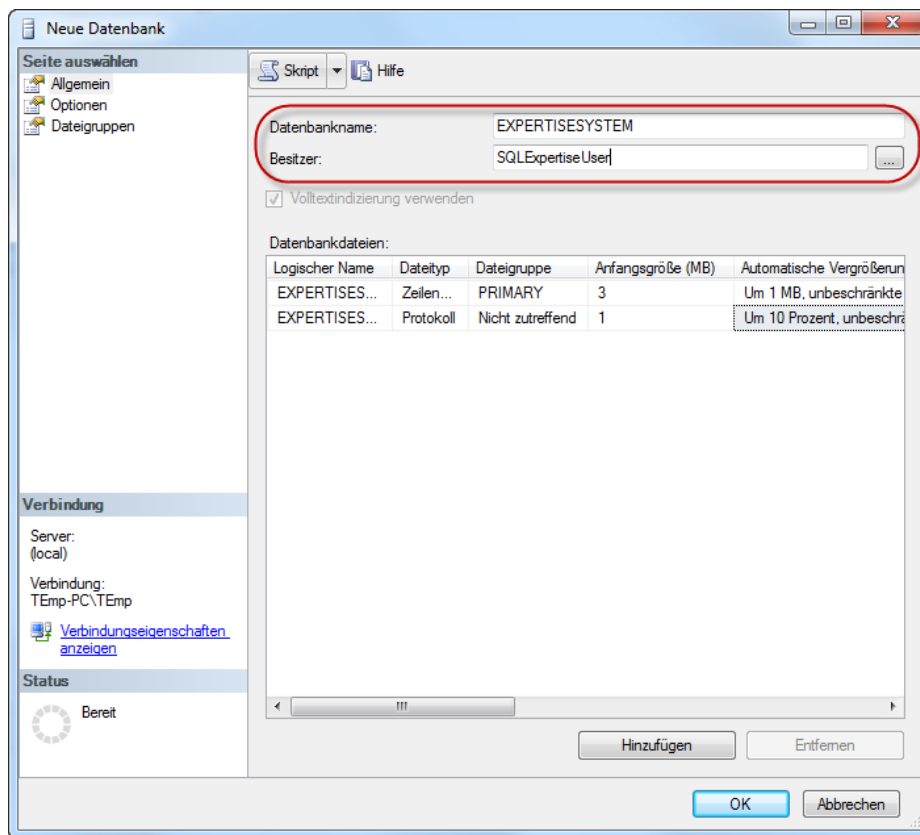


Abbildung 139: Leere Datenbank erstellen

Danach kann diese leere Datenbank über das Menü „Wiederherstellen“ mit der Vorlagedatenbank überschrieben werden (Abbildung 140).

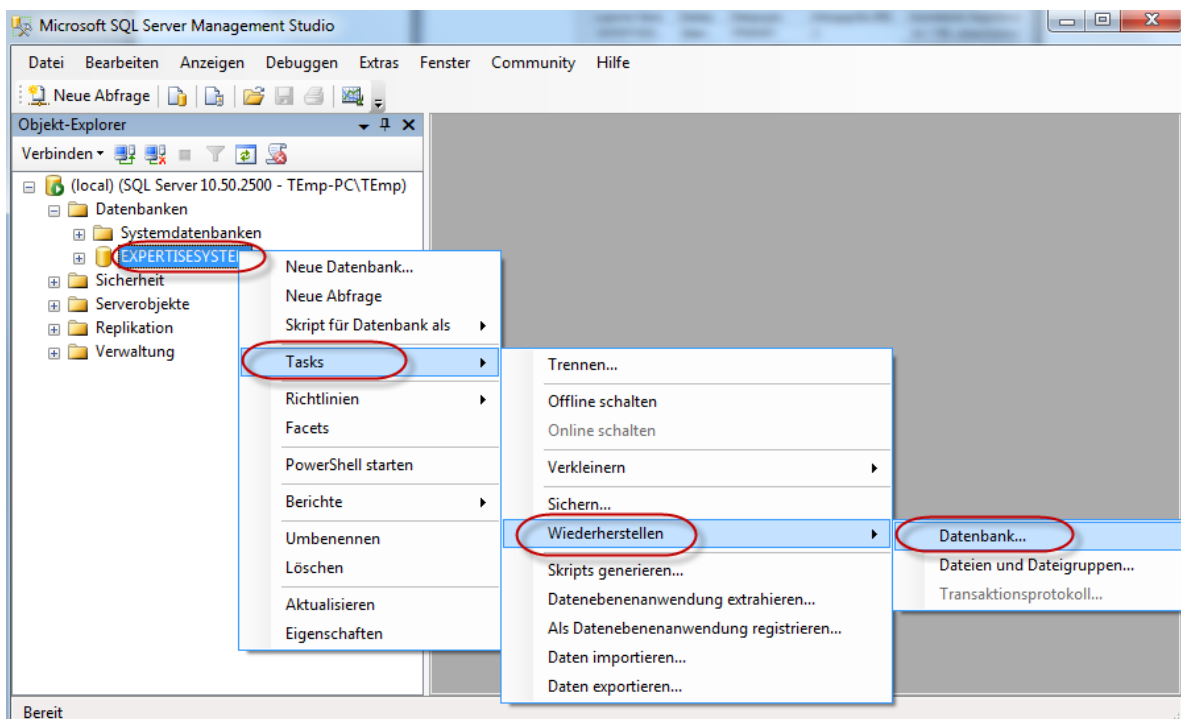


Abbildung 140: Vorlagedatenbank wiederherstellen

## 12.2.4 Installation AXAVIAseries

Man startet die „Expertise\_setup.exe“ Datei und folgt den Anweisungen.

Nach dem Installieren und darauffolgenden Starten des Programmes wird man gefragt, wo die Datenbankverbindungsinformationen abgelegt werden soll (Abbildung 141).

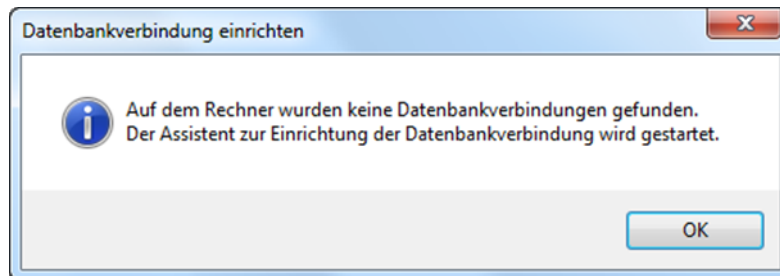


Abbildung 141: Datenbankverbindung einrichten

Falls mehrere Personen mit der Software arbeiten, ist es vorteilhaft diese Verbindungsinformationen auf einem Netzwerkpfad abzulegen. Dadurch erspart man sich administrative Tätigkeiten, falls sich diese ändern da nur noch an einer Stelle die Information angepasst werden muss, siehe Abbildung 142.

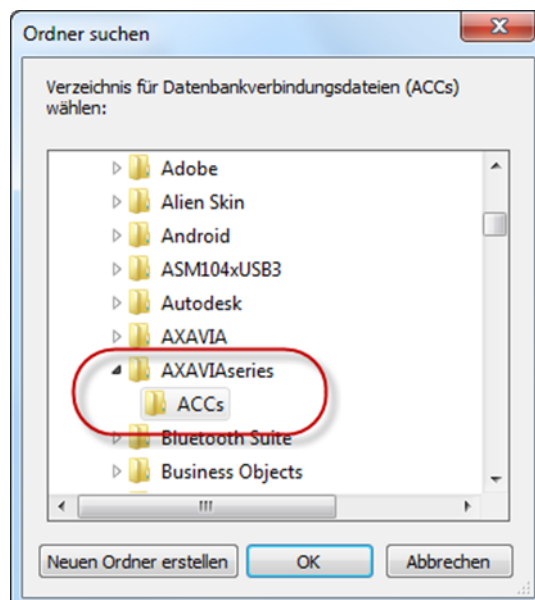


Abbildung 142: Verzeichnis für Datenbankverbindungsdateien

Als nächster Schritt muss die Verbindungsinformation erstellt werden. Im „Datenbankverbindung bearbeiten“ Fenster drücken Sie auf „Weiter“ um alle Information einzusehen (Abbildung 143).

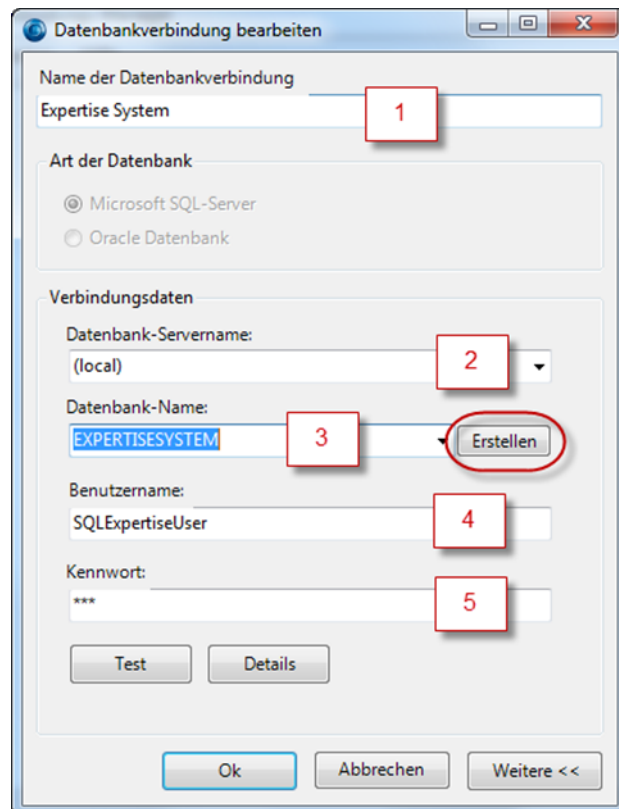


Abbildung 143: Datenbankverbindung bearbeiten

1. Verbindungsname
2. SQL Server
3. Datenbankname
4. SQL Benutzer
5. SQL Benutzerkennwort

Nachdem alle Informationen eingetragen wurden, wird durch das Drücken von „Erstellen“ eine neue Datenbank am SQL-Server angelegt. Im Datenbankverbindungsmanager (Abbildung 144) scheint nun diese neue Verbindung auf.

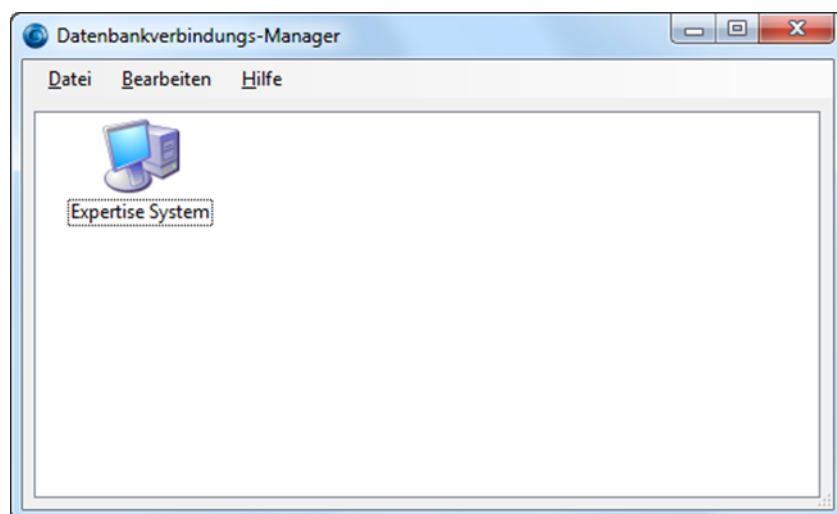


Abbildung 144: Datenbankverbindungs-Manager

Außerdem wurde diese Datenbankinformation (Abbildung 145) mit dem gleichen Namen am Dateisystem abgelegt.

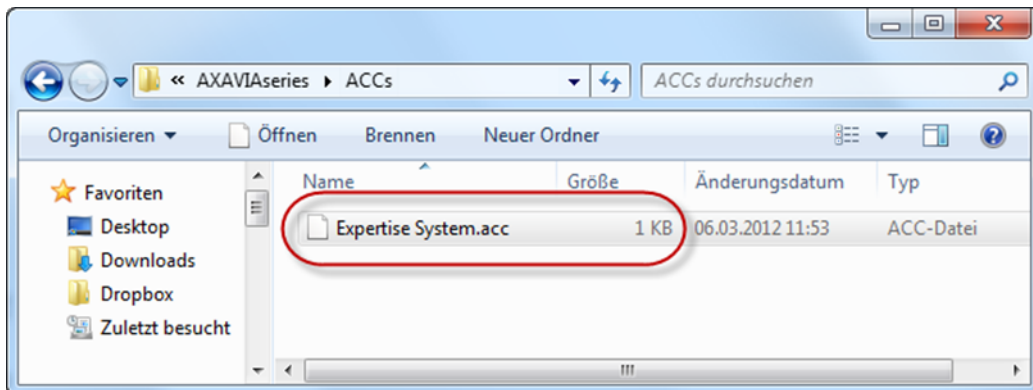


Abbildung 145: ACC-Datei mit den Datenbankinformationsverbindungen

Schließen Sie anschließend das „Datenbankverbindungs-Manager“ Fenster, um sich das erste Mal in die neu erstellte Datenbank einzuloggen (Abbildung 146).

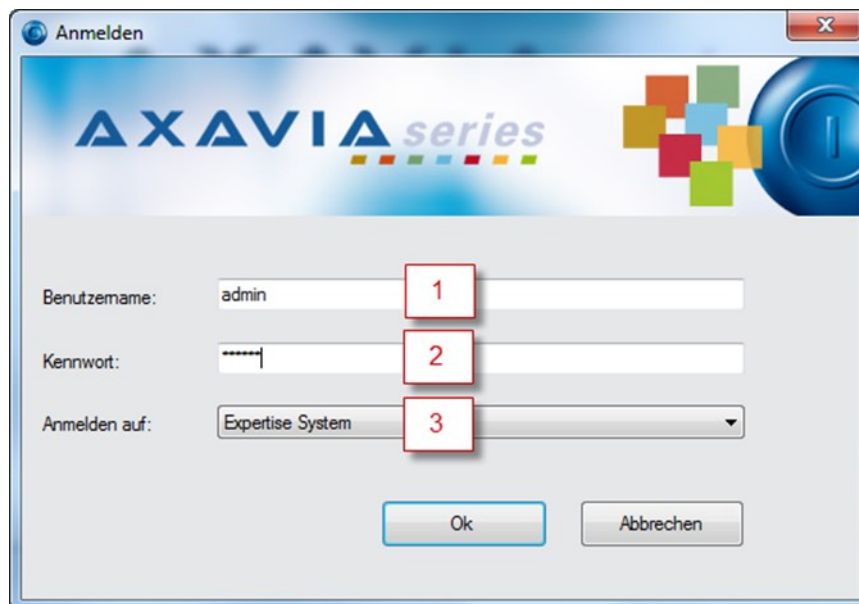


Abbildung 146: AXAVIAseries Anmeldedialog

1. Benutzername:
  - a. Gutachtenkonfigurator: „expconfig“
  - b. Benutzer: „user“
2. Kennwort:
  - a. Gutachtenkonfigurator: „expconfig“
  - b. Benutzer: „ user“
3. Anmelden auf: <Wählen Sie die zuvor erstellte Datenbankinformation>

## 12.3 Bedienungsanleitung

Expertise Managementsystem (EMS) wurde in Kooperation mit der Firma AXAVIAseries GmbH und dem Institut für Informationsverarbeitung und Mikroprozessortechnik an der Universität Linz entwickelt.

Die Ansprechpartner sind:

- Prof. Michael Sonntag ([michael.sonntag@jku.at](mailto:michael.sonntag@jku.at))
- Thomas Schneider ([thomas.schneider@axavia.com](mailto:thomas.schneider@axavia.com))


Die Aufgabenstellungen dieser Arbeit waren:

- Server-basiertes System
- Dokumentenmanagementsystem
- Vorlagen
- Automatisches Glossar für das Gutachten
- Zeitaufzeichnung
- Rechnungslegung
- Protokollieren aller Vorgänge

Es wurden diverse Gutachtensysteme untersucht und ausgehend dieser Information eine eigene Implementierung mit AXAVIAseries ausgearbeitet. Die wichtigsten Vorgaben entnimmt man den Aufgabenstellungen. Die Software wird als Fallstudie frei zugänglich gemacht. Es existiert eine eigener Installationssoftware, Vorlagedatenbank, Installations- und Bedienungsanleitung, die über die Webseite von AXAVIAseries bezogen werden kann.

In den nachfolgenden Kapiteln wird gezeigt, wie man diese Software bedient und die einzelnen Punkte der Aufgabenstellung bewältigt werden.

### 12.3.1 Starten des Programmes

Nach der Installation des Programmes doppelklickt man auf das Starticon . Das Startlogo erscheint, gefolgt vom Login-Dialog (Abbildung 147).

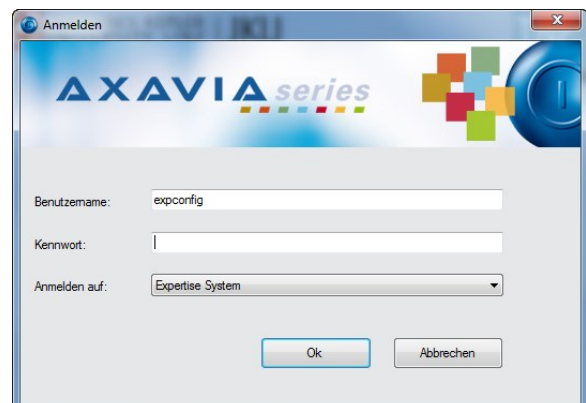
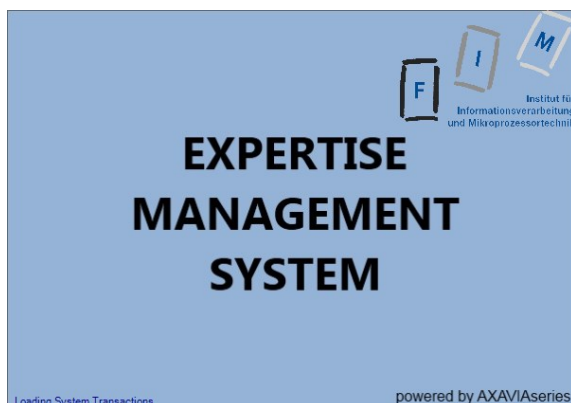


Abbildung 147: Splashscreen und Anmeldedialog

Das System erhielt zwei vordefinierte Benutzer, einer ist für die Konfiguration zuständig und der andere dient zum Benutzen des Systems.

Je nachdem wie man sich einloggt erhält man eine unterschiedliche Sichtweise auf die Daten. Die Benutzer sind:

- expconfig
- user

Das Standardpasswort dieser Benutzer ist jeweils der Login Name. Beim ersten Starten des Programms loggt man sich mit dem Konfigurationsbenutzer ein, um Einstellungen vorzunehmen.

### 12.3.2 Konfigurieren des Programmes

Nachdem das Programm geladen wurde, findet man sich in der Konfigurationsansicht wieder (Abbildung 148).

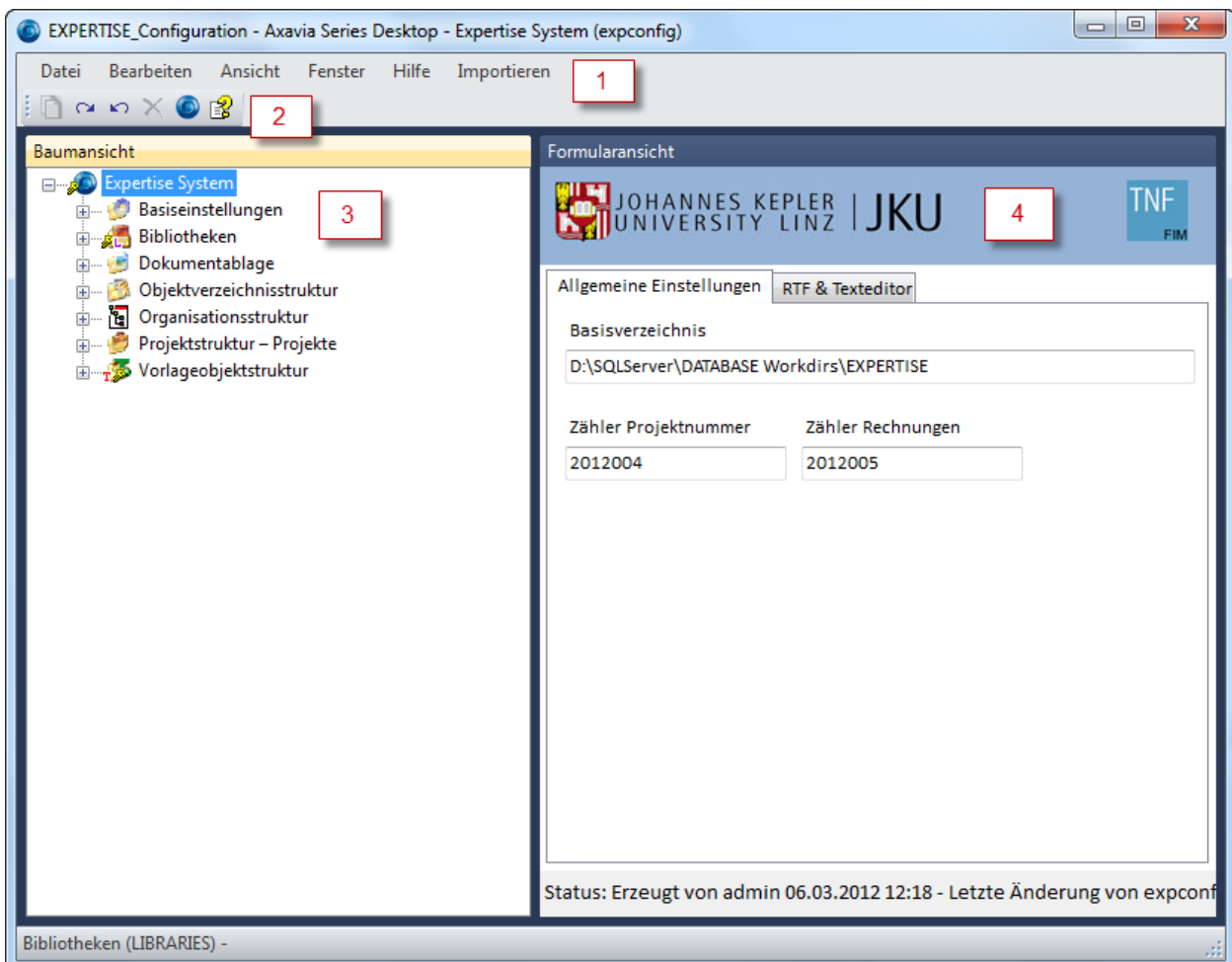


Abbildung 148: Konfigurationsansicht "EXPERTISE\_Configuration"

#### 1) Menüleiste

In der Menüleiste sind die Funktionen ersichtlich um das Programm zu beenden, ein neues Fenster zu öffnen, oder um über den Punkt „Importieren“ neue Updates einzuspielen.

#### 2) Symbolleiste

Die Symbolleiste dient um Aktionen über einen Kurzbefehl auszuführen, aber auch um zu erfahren welches Objekte, auf dem jeweilig markierten Objekt in der Baumansicht, neu erzeugbar ist.

3) Baumansicht

Eine strukturierte Übersicht über alle Objekte, welche in der geöffneten Ansicht zu sehen sind.

4) Formularansicht

Zeigt die Attribute des markierten Objektes in der Baumansicht an. Als Fußzeile im Formular wird vom aktuellen Objekt immer der Status angezeigt. Wann und vom wem wurde es erzeugt und selbiges über die letzte Änderung.

Es ist notwendig, sich mittels der Baumansicht durch die nachfolgenden Punkte zu navigieren und die notwendigen Daten in der Formularansicht einzutragen.



## Expertise System

Die wichtigste Einstellung, die vorzunehmen ist, ist die Angabe eines Basisverzeichnisses in welchem alle Dateien abgelegt werden, wenn diese über das interne Dokumentenmanagementsystem bearbeitet werden. Im gleichen Reiter kann auch noch angegeben werden, mit welcher Zahl die Projekt- & Rechnungsnummer gestartet werden soll. Der zweite Reiter „RTF & Texteditor“ dient dazu um zu konfigurieren welche Schriftart, Aufzählungszeichen und Stile es in den Texteditor geben soll.

Als Standardfont wurde Arial 12 sowohl für ein RTF Eingabefeld als auch für den featurereicheren Texteditor gewählt.

## Basiseinstellungen

Zuerst müssen die Basiseinstellungen (Abbildung 149) vorgenommen werden.

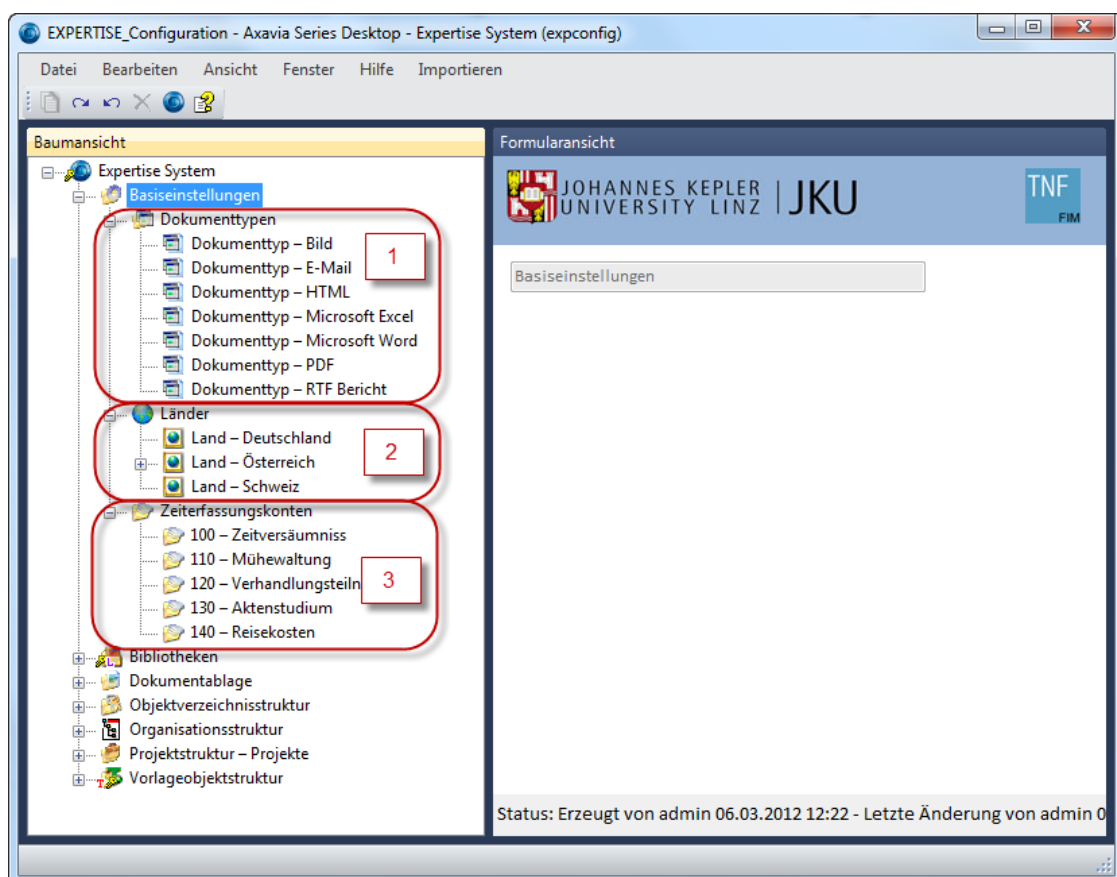


Abbildung 149: Basiskonfiguration

- 1) Die Dokumenttypen sind für das Dokumentenmanagementsystem relevant, da diese angeben, welche externen Programme gewählt werden sollen, um diese Dateien zu öffnen. Das Programm kann über das Attribut „Anwendung für Bearbeitung“ definiert werden oder, falls das Feld leer gelassen wird, wird das Programm verwendet, welches im Betriebssystem definiert ist.
- 2) Neue Länder können hinzugefügt werden bzw. bearbeitet werden. Die Hauptaufgabe dieser Einträge findet man im Reiter „Umsatzsteuereinstellungen“, welche für Rechnungen und deren Rechnungspositionen interessant werden.
- 3) Zeiterfassungskonten können angelegt werden, diese werden für die Zeitaufzeichnung benötigt.

## Bibliothek

In der Bibliothek (Abbildung 150) können vordefinierte Produktinformationen für die jeweiligen Rechnungspositionen abgelegt werden. Später wenn Rechnungspositionen erstellt werden, können über eine Produktnummer Daten in die Rechnung übernommen werden.

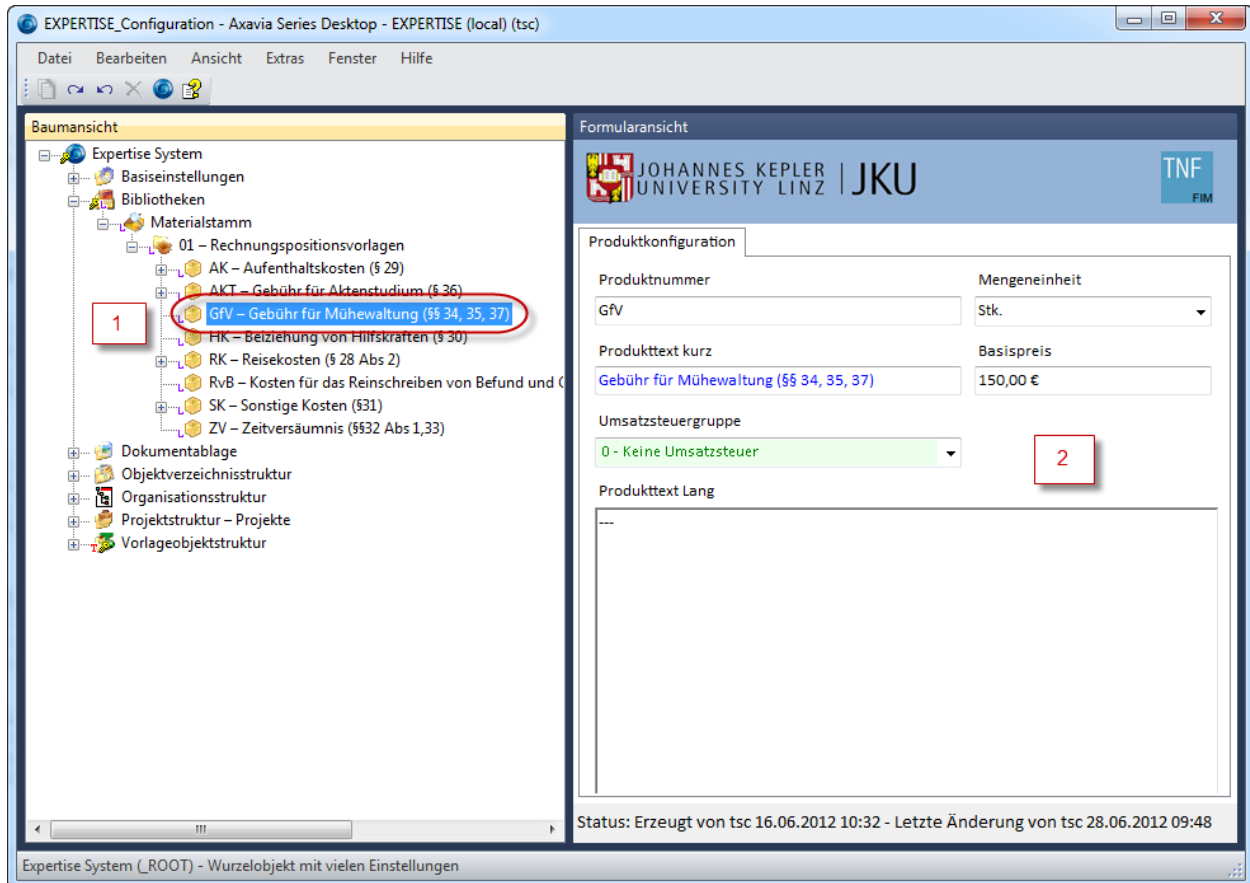


Abbildung 150: Bibliothekseinstellungen

- 1) In der Baumansicht können neue „Produkte“ über die Symbolleiste erstellt werden.
- 2) In der Formularansicht definiert man:
  - a. Produktnummer
  - b. Mengeneinheit
  - c. Basispreis
  - d. Umsatzsteuergruppe
  - e. Produkttext kurz
  - f. Produkttext lang

## Dokumentablage

Das Dokumentenmanagementsystem in AXAVIASeries erstellt aus dem Basisverzeichnis (siehe Kapitel

Expertise System) eine Ordnerstruktur. Sowohl für global verwendete Dateien wie Geschäftspartnerdokumente, als auch für jedes Gutachtenprojekt einen eigenen wo alle, für das jeweilige Gutachten, notwendigen Dokumente abgelegt werden.

Die hier ersichtlichen Ordnerstrukturen (Abbildung 151) sind für die globale Ablage gedacht.

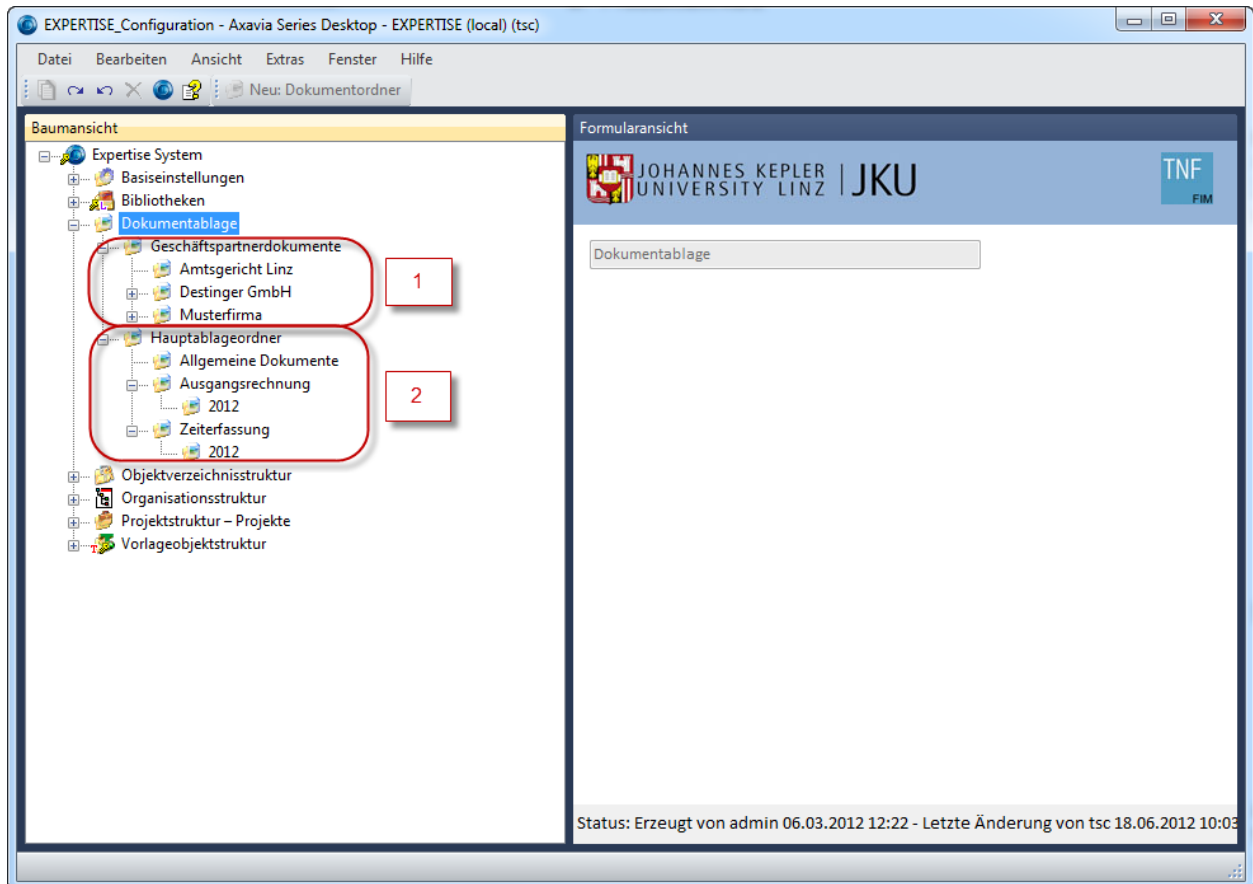


Abbildung 151: Dokumentablage

#### 1) Geschäftspartnerdokumente

Jeder Kunde welcher im System hinterlegt ist, sollte einen eigenen Ordner erhalten in dem Kundenspezifische Dokumente abgelegt werden können. Um für jeden einen Ordner anzulegen kann eine Funktion auf dem Wurzelobjekt „Expertise System“ verwendet werden. Über das Kontextmenü ist der Punkt „Erzeuge Dokumentordner für jeden Geschäftspartner“ aufrufbar.

#### 2) Hauptablageordner

Beim Erzeugen der Berichte für Rechnungen und Zeitaufzeichnungen, werden als Vorauswahl zur Speicherung die abgebildeten Ordner angegeben.

## Organisationsstruktur

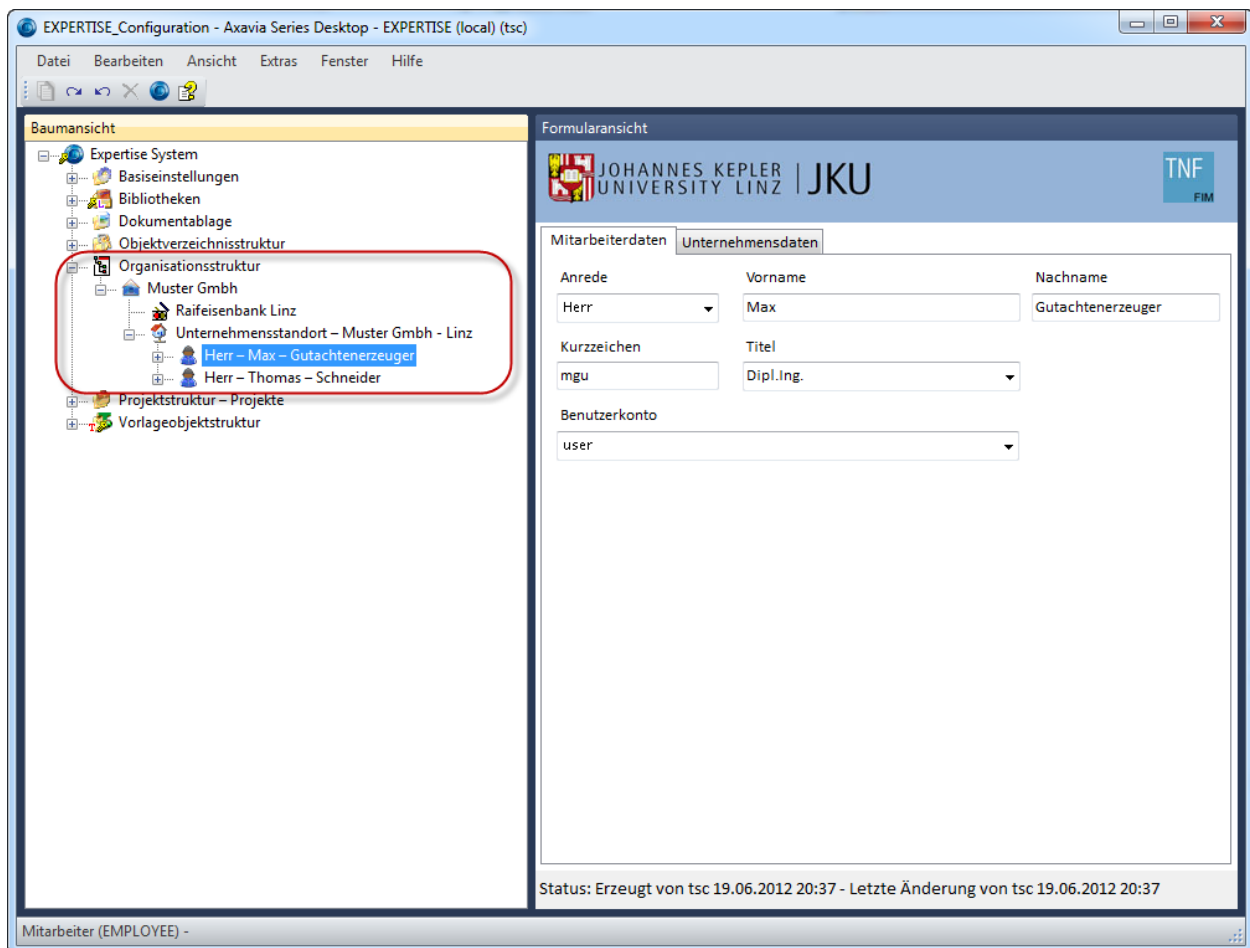


Abbildung 152: Organisationsstruktur konfigurieren

In der Organisationsstruktur (Abbildung 152) werden alle Daten eingegeben, dadurch werden die Person und das Unternehmen auf den jeweiligen Berichten (Rechnungen, Gutachten) dargestellt.

Über die Symbolleiste kann man auch Bankverbindungen erstellen, die dann für die jeweiligen Unternehmensstandorte auswählen können.

Als Person ändert man die Vorlage des „Max Gutachtenerzeugers“, dieser wurde bereits mit dem Benutzerkonto „user“ verknüpft.

### Projektstruktur & Vorlagen

In der Konfigurationsansicht ist es möglich alle Gutachtenprojekte (Abbildung 153) im System zu sichten und bereits ausgearbeitete Checklisten & Gutachten als Vorlagen zu übernehmen.

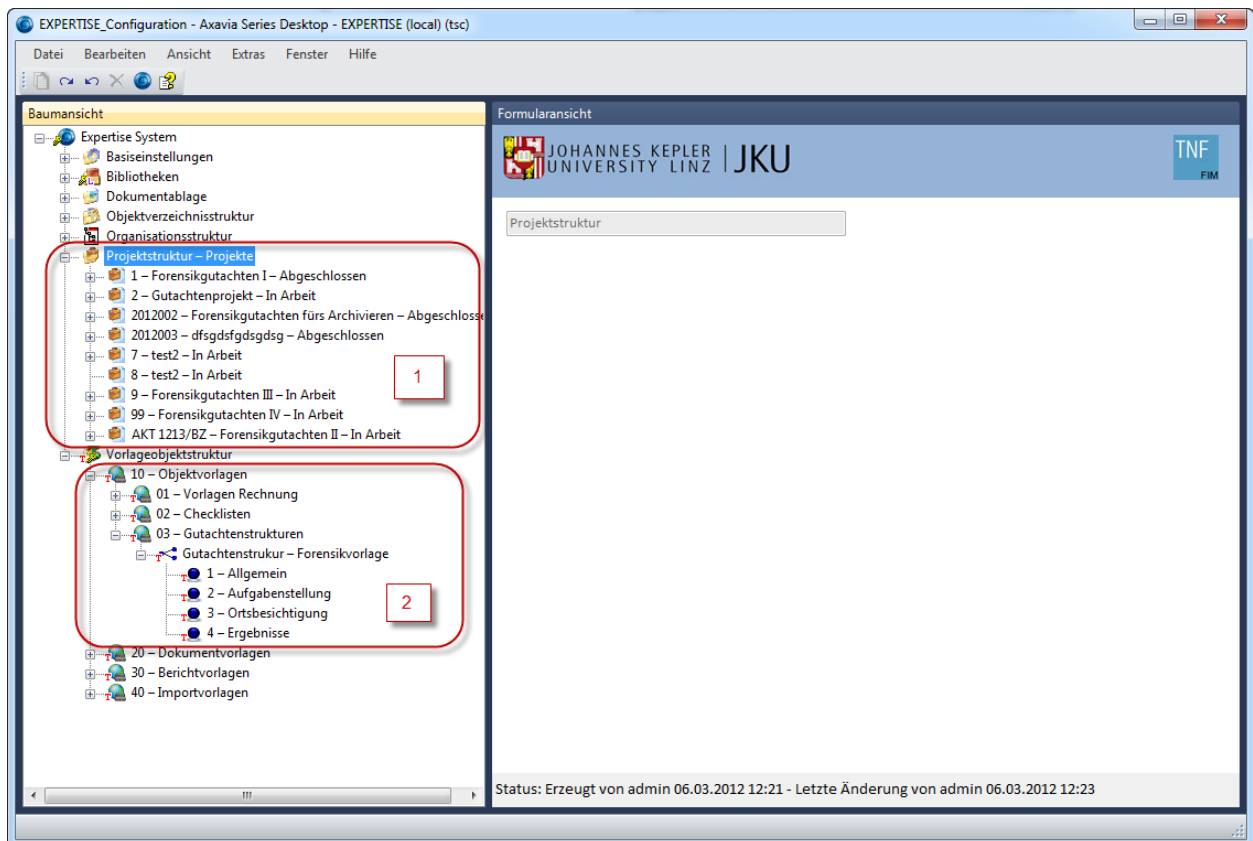


Abbildung 153: Vorlagen übernehmen aus einem Projekt

- 1) Das Projekt, das bereits fertige Daten enthält, wird gesucht und in den Vorlagenbereich (z.B.: eine fertige Gutachtenstruktur inklusive Kapitel) kopiert.
- 2) Im Vorlagebereich können im Abschnitt Objektvorlagen, vorgefertigte Informationen aus den Projekten übernommen werden.

Die Dokumentvorlage, Berichtvorlage & Importvorlage sollten nicht verändert werden, falls doch Anpassungen notwendig sind, sollte Herr Schneider kontaktiert werden, da diese Dateivorlagen teilweise mit einer Skriptsprache hinterlegt wurden.

### 12.3.3 Allgemeine Informationen

#### Dateien & Ordner

Dateien, die mit dem System verwaltet werden, sind physikalisch auch auf der Festplatte vorhanden, damit diese auch bearbeitet werden können. Im System selbst, sieht man nur ein Referenzobjekt auf diese Datei.

#### Dateien & Ordner importieren

Dateien und Ordner können über Drag & Drop in das System kopiert werden, dabei wird das Original nicht verändert.

Dateien können unter alle Objekte verknüpft werden, wobei zu beachten ist, dass, wenn eine Datei oder Ordner nicht auf einen Dokumentordner im System verknüpft wird, ein Auswahldialog erscheint und ein Dokumentordner gewählt werden muss.

Es ist aber auch möglich, wenn ein Dokumentordner markiert ist, über das Kontextmenü Dateien und sogar ganze Verzeichnisstrukturen zu importieren, siehe Abbildung 154.

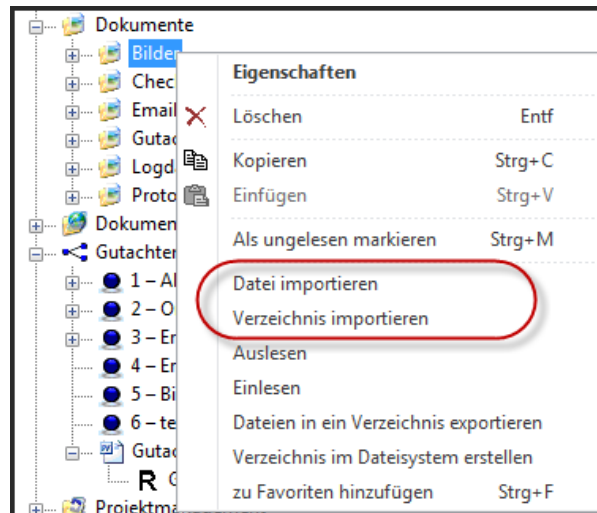


Abbildung 154: Dateien und Ordner importieren



### Dateien & Ordner exportieren

Wählt man eine Datei oder Dokumentordner aus so kann man diese über das Kontextmenü in ein zu wählendes Verzeichnis exportieren.

### Dateien & Ordner Ein- und Auslesen

Jede Datei & Dokumentordner kann eingelesen, also in die Datenbank gespeichert und ausgelesen, wieder ins Dateisystem gespeichert werden, die Bezeichnung der Datei wird als Dateinamen verwendet. Ein- und Auslesen ist das Kopieren des Inhaltes in die Datenbank bzw. auf den physikalischen Datenträger zurück in das Dateisystem.

Dabei wird beim Einlesen die Datei im Dateisystem nicht gelöscht, sondern standardmäßig nur schreibgeschützt.

Eine ausgelesene Datei ist durch das Overlay-Icon  gekennzeichnet. Nach dem Einlesen, sieht das Icon anders aus .

### Berichte erstellen

Folgende Berichte sind im Programm enthalten:

- Rechnungsbericht
- Gutachtenbericht
- Aktivitätenbericht
- Monatsübersicht Stunden
- Auslagenersatzbericht

Berichte können immer über das Kontextmenü aufgerufen werden, dafür ist es erforderlich, ein Objekt auf dem ein Bericht gestartet werden kann (z.B.: eine Rechnung), zu markieren (Abbildung 155).

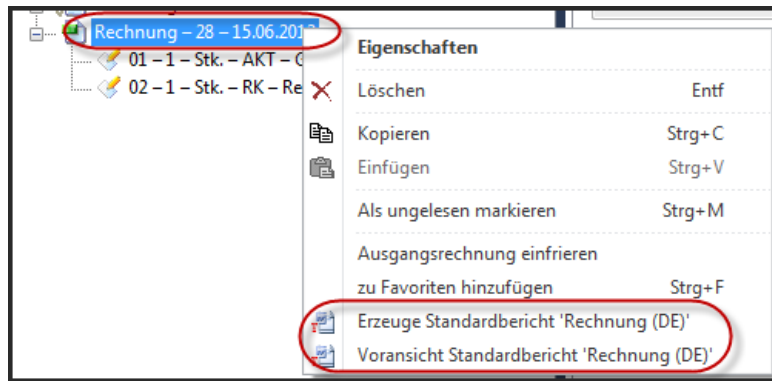


Abbildung 155: Rechnungsbericht erstellen

Es gibt sowohl die Möglichkeit den Bericht zu „Erzeugen“ oder über „Voransicht“ diesen einmal anzuzeigen.

Wählt man „Erzeugen“, erscheint ein Dialog in dem ausgewählt werden kann, wo dieser Bericht abgespeichert werden soll (Abbildung 156).

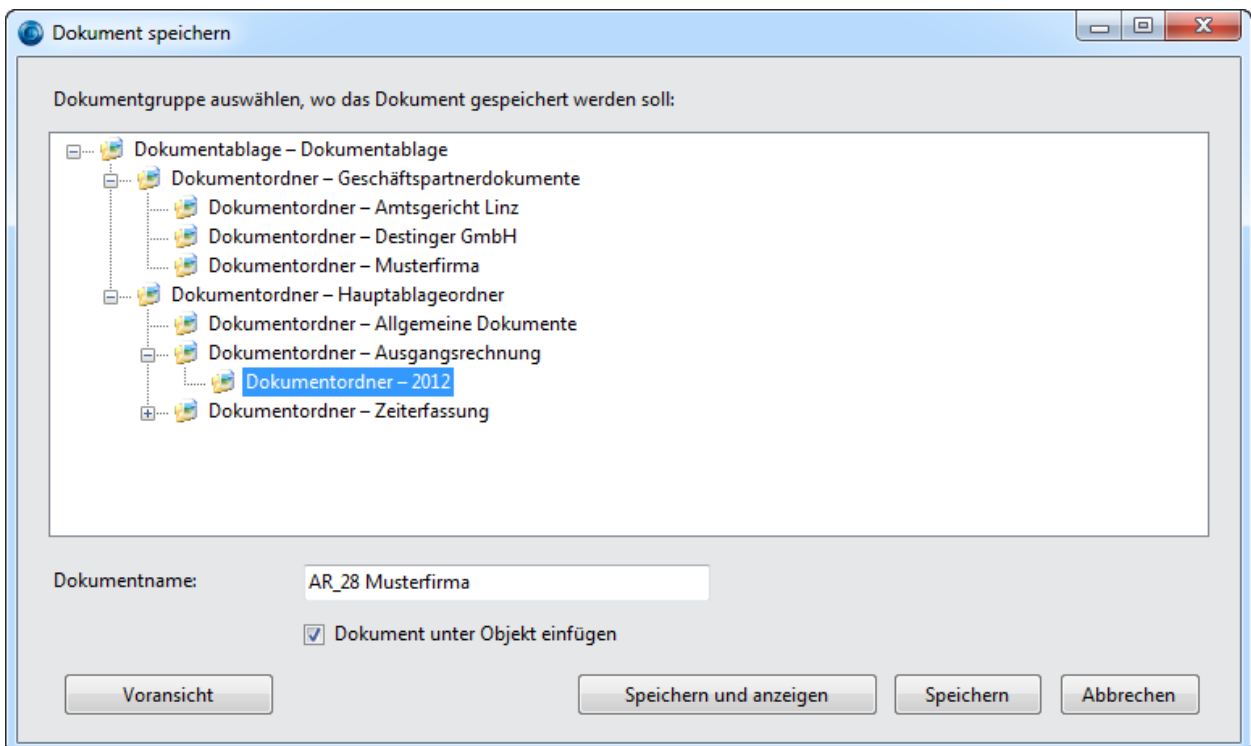


Abbildung 156: Dokument speichern - Dialog

Nun wählt man das Verzeichnis in dem der Bericht physikalisch abgelegt wird. Der Name der Datei kann noch angepasst werden und die Einstellung „Dokument unter Objekt einfügen“ dient dazu, dass der Bericht mit der Rechnung verknüpft wird (Abbildung 157).



Abbildung 157: Bericht mit Startobjekt verknüpft

Ein Doppelklicken öffnet diesen Bericht.

## Berichte aktualisieren

Ein Bericht kann auch über das Kontextmenü aktualisiert werden, falls sich etwas auf den Objekten ändert welche für die Informationen im Bericht zuständig sind. Dabei bleiben Speicherort und Dokumentinformationen erhalten.

Beim Aktualisieren wird man gefragt, ob man eine Revision erzeugen will, so können mehrere Versionen des Berichtes verwaltet werden (Abbildung 158).

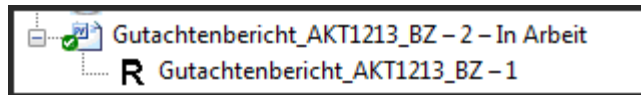


Abbildung 158: Dokumentrevisionen

Um sich eine Revision anzusehen, ist es möglich über das Kontextmenü „Öffnen“ zu wählen, dabei wird diese schreibgeschützte Datei geöffnet.

### 12.3.4 Ansichten im Programm

Es gibt insgesamt fünf verschiedene Ansichten, wobei vier für den Anwender und eine zum Konfigurieren des System sind.

Konfigurator:

- EXPERTISE\_Configuration

Benutzer:

- EXPERTISE\_Main
- EXPERTISE\_Businesspartner
- EXPERTISE\_Project
- EXPERTISE\_Time

Im Nachfolgenden werden alle Benutzeransichten behandelt und alle Funktionen durchgenommen, die in diesen Ansichten vorkommen.



## EXPERTISE\_Main

Diese dient als Startansicht für den Benutzer, siehe dazu Abbildung 159.

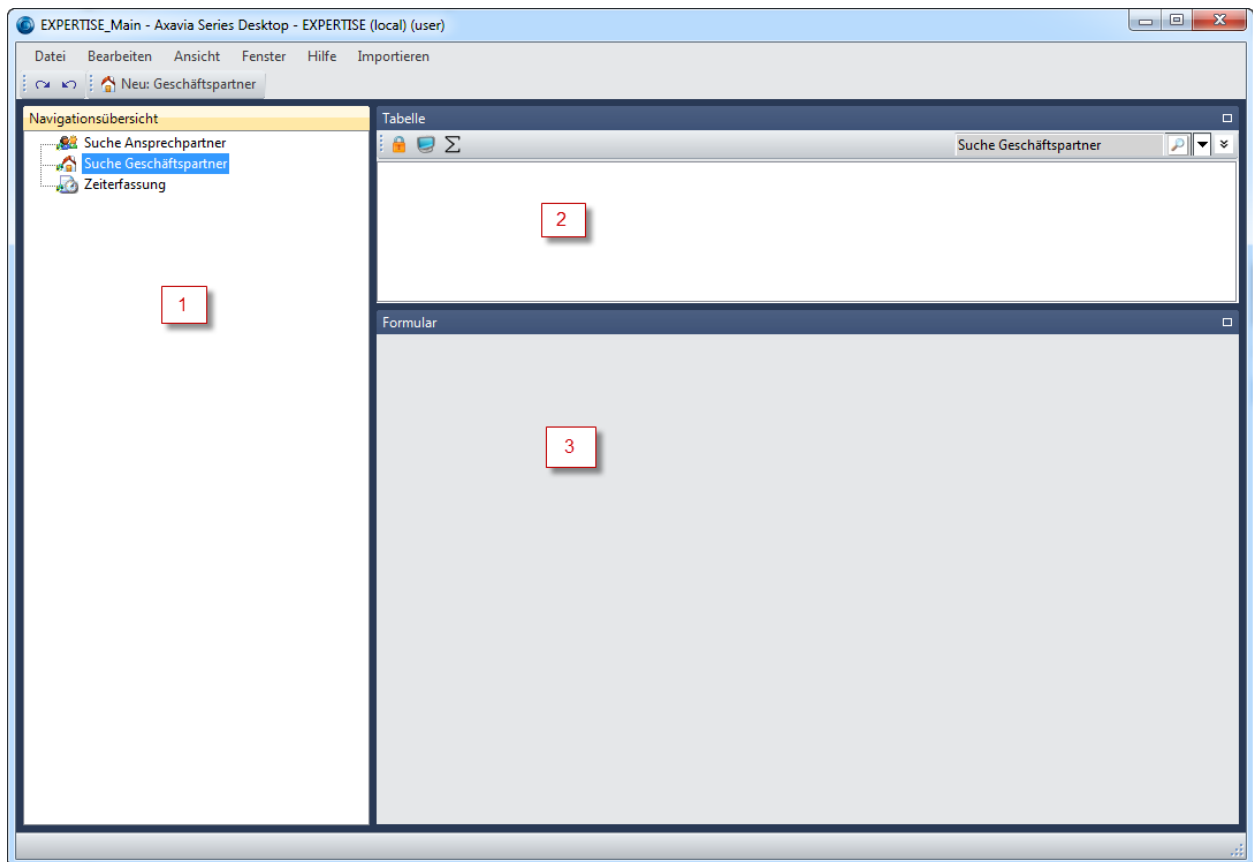


Abbildung 159: Ansicht "EXPERTISE\_Main"

- 1) Navigationsübersicht
- 2) Tabelle
- 3) Formular

Folgende Funktionen können in dieser Ansicht getätigt werden.

### Neuen Geschäftspartner anlegen

Dazu wird in der Navigationsübersicht „Suche Geschäftspartner“ ausgewählt und in der Symbolleiste wird ein „Neu: Geschäftspartner“ Icon angezeigt. Ein Klick darauf, erstellt einen neuen Kunden (Abbildung 160) und es kann damit begonnen werden, Unternehmensdaten einzugeben.

The screenshot shows a software window titled "Eigenschaften: Geschäftspartner". At the top, there is a "Objektmenü" with a "Schließen" button. Below this is a "Formular" tab. The header of the form area features the logo of Johannes Kepler University Linz (JKU) and the logo for TNF FIM. The main form is titled "Kundeninformation" and contains the following fields:

- Firmenname: Two text input fields.
- Land: A dropdown menu.
- Ort: A text input field.
- Adresse: A text input field.
- Telefon: A text input field.
- Telefax: A text input field.
- Internet Adresse: A text input field.
- Bemerkung: A large text area for notes.

Abbildung 160: Neuen Geschäftspartner erzeugen

Es ist kein explizites „Speichern“ notwendig, sondern „Schließen“ beendet den Eingabeprozess.

### Suchen nach Geschäftspartner

Durch das Markieren in der Navigationsansicht „Suche Geschäftspartner“ (Abbildung 161), kann in der Tabelle im Feld „Suche Geschäftspartner“ entweder ein „\*“ für alle in der Datenbank befindlichen Kunden, oder Teile des Firmennamens eingegeben werden.

Die Tabelle listet alle gefundenen Treffer auf. Wird nun in der Tabelle einer dieser Treffer markiert, wird die detaillierte Information in der Formularansicht dargestellt. Ein Doppelklick öffnet mit dem gewählten Objekt eine neue Ansicht „EXPERTISE\_Businesspartner“.

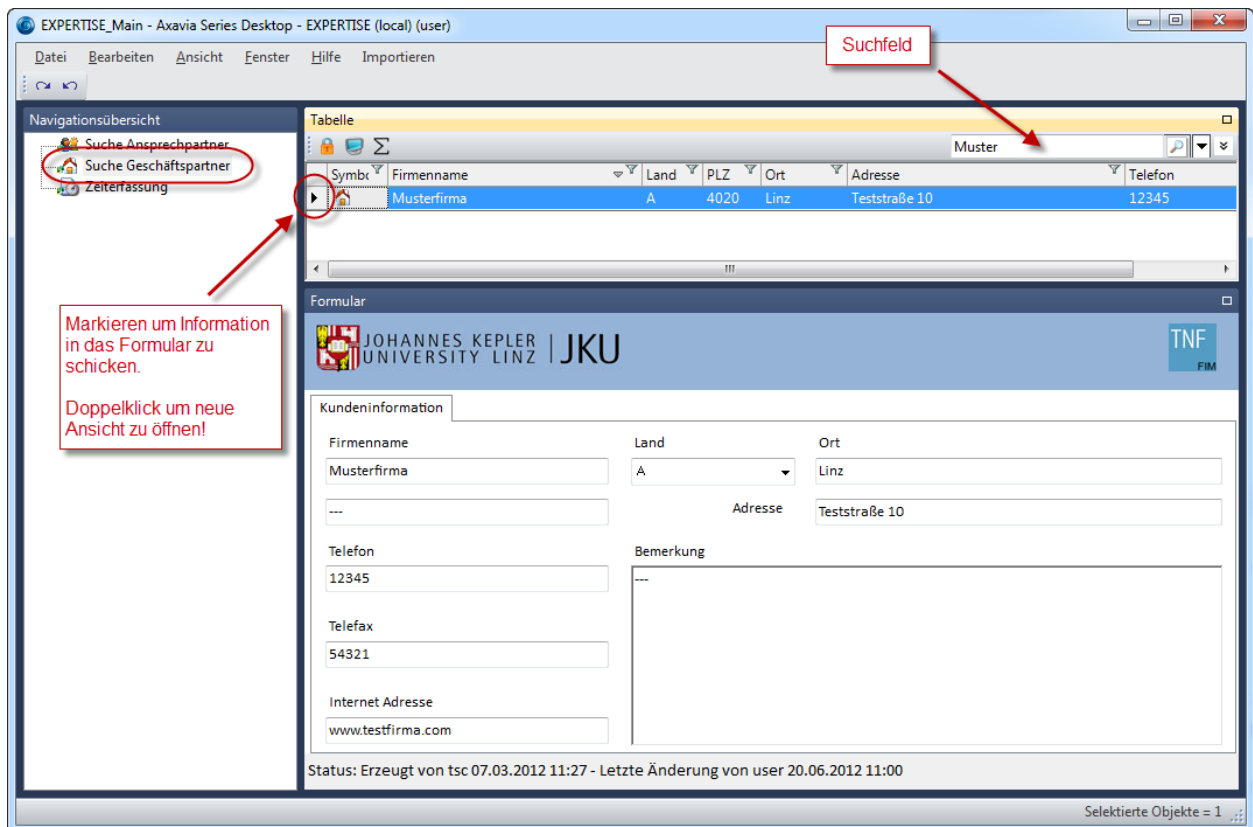


Abbildung 161: Geschäftspartner suchen

### Suchen nach Ansprechpartner

In der Ansicht „EXPERTISE\_Businesspartner“ wird es möglich nach Ansprechpartnern, die zu Kunden zugeordnet sind anzulegen. In dieser Ansicht ist es möglich auch nur nach einem einzigen Ansprechpartner zu suchen.

Die Schritte werden exakt so durchgeführt, wie im Kapitel Geschäftspartner, auch das Markieren und Doppelklicken funktioniert ebenso.

### Zeiterfassung

Beim Klicken auf die Zeiterfassung in der Navigationsübersicht, werden alle Zeitaufzeichnungseinträge & -tage in der Tabelle dargestellt. Selektieren man einen Eintrag in der Tabelle, werden die Informationen über diesen Tag / Eintrag in das Formular geschickt.

Beim Doppelklick auf die „Zeiterfassung“ öffnet sich eine neue Ansicht EXPERTISE\_Time, die genutzt werden kann, um Zeiten und Auslagen einzutragen.

### EXPERTISE\_Businesspartner

In dieser Ansicht befinden sich Angaben zu den Kunden (Abbildung 162):

- Ansprechpartner
- Kundendokumentorder
- Gutachtenprojekte
- Rechnungen

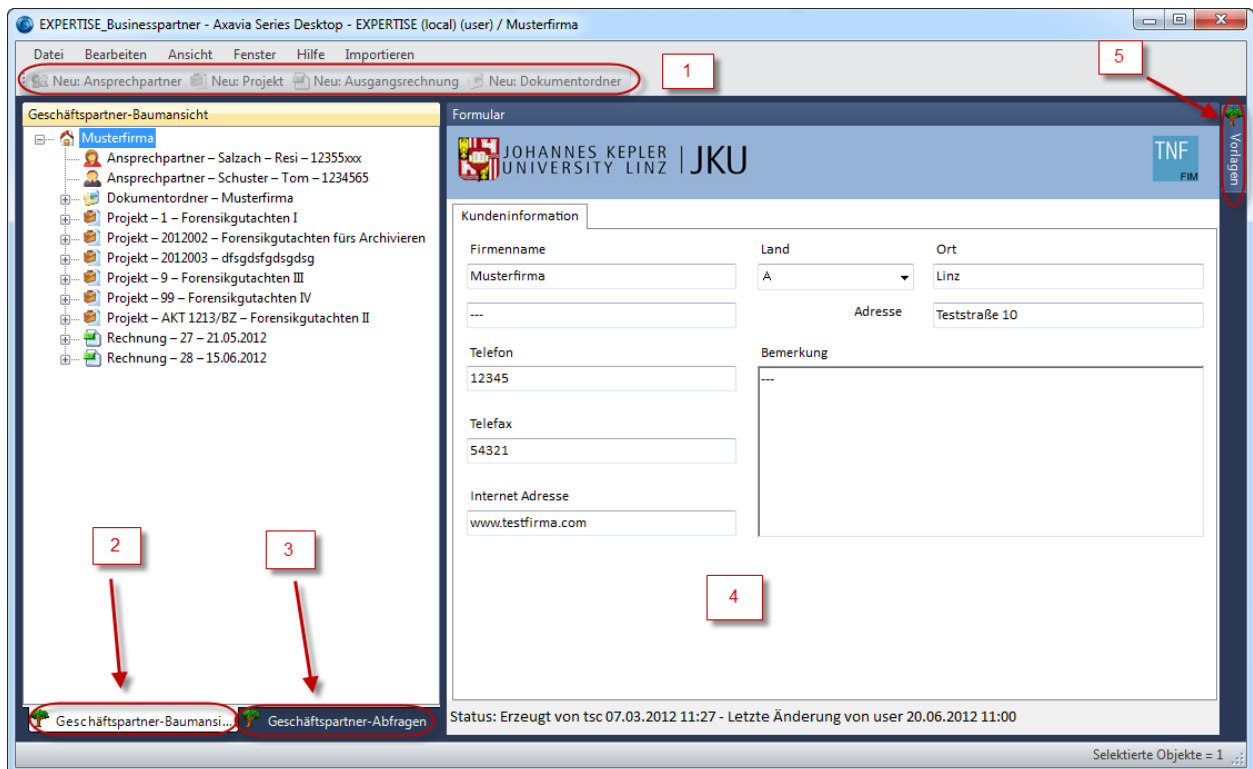


Abbildung 162: Ansicht "EXPERTISE\_Businesspartner"

- 1) Die Symbolleiste, in der je nachdem welches Objekt im Baum markiert, neue Objekte angelegt werden können.
- 2) Standardbaumansicht, beim Markieren eines Objektes ändert sich die Funktionalität der Symbolleiste und im Formular werden die Daten zu dem selektierten Objekt angezeigt.
- 3) Abfragebaumansicht  
Es wurden zwei Abfragen eingebaut, welche zum jeweiligen Kunden alle offenen Rechnungen & Projekte darlegt.
- 4) Formularansicht, zeigt Information zu den markierten Objekten im Baum an.
- 5) Vorlagen, derzeit ist es möglich leere Microsoft Office Vorlagen und vorgefertigte Ausgangsrechnungen auf einen Dokumentordner bzw. Kunden zu ziehen. Zum Ändern dieser, befolgt man die Ausführungen im Kapitel Projektstruktur & Vorlagen.

Folgende Funktionen können in dieser Ansicht getätigt werden.

### Kundendokumentordner anlegen

Durch das Markieren der Wurzel im Baum und über das Kontextmenü, ist es möglich, einen kundenspezifischen Ordner anzulegen (Abbildung 163).

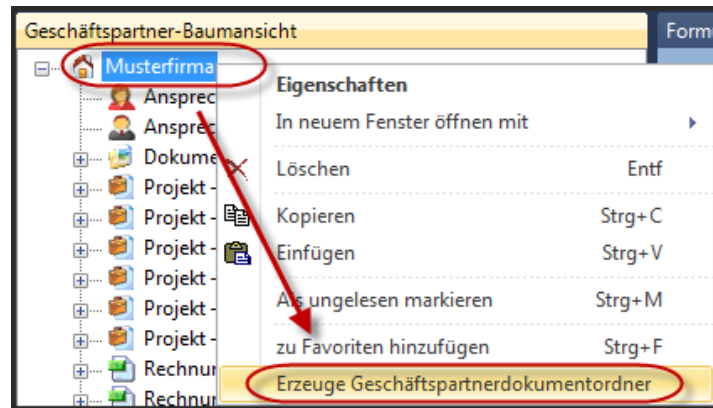


Abbildung 163: Geschäftspartnerdokumentordner erzeugen

Wurde dieser Hauptordner erstellt, können über die Symbolleiste weitere Unterordner generiert werden. Diese Ordner dienen dazu, um allgemeine Dokumente, welche zwar Kundenspezifisch aber nicht Gutachtenspezifisch sind, direkt zu hinterlegen.

### Neuen Ansprechpartner anlegen

Dazu markiert man die Wurzel im Baum, die den Kunden darstellen und man klickt in der Symbolleiste auf „Neu: Ansprechpartner“.

Abbildung 164: Neuen Ansprechpartner erstellen

Im Formular können Anrede, Titel, etc. angegeben werden und im zweiten Reiter „Firmeninformationen“ findet man nochmals die Zusammenfassung der Kundendaten (Abbildung 164).

Wiederum ist ein explizites Speichern nicht notwendig, da die Daten in Echtzeit in die Datenbank übertragen werden.

### Neue Nachricht(E-Mail) senden

Wenn ein Ansprechpartner markiert wird, erscheint in der Symbolleiste ein neuer Eintrag „Neu: Nachricht“.

Es wird nur Microsoft Outlook 2003-2010 unterstützt, genauere Informationen zum Installieren der Schnittstelle können im Kapitel Installation AXAVIASeries nachgelesen werden.

Nachdem „Senden“ gedrückt wurde, wird vom Programm nach dem Speicherort gefragt. Dazu wird ein globaler Ordner bzw. der Kundendokumentordner gewählt (Abbildung 165). Es liegt im Ermessen des Benutzers, wo diese Nachricht abgespeichert wird.

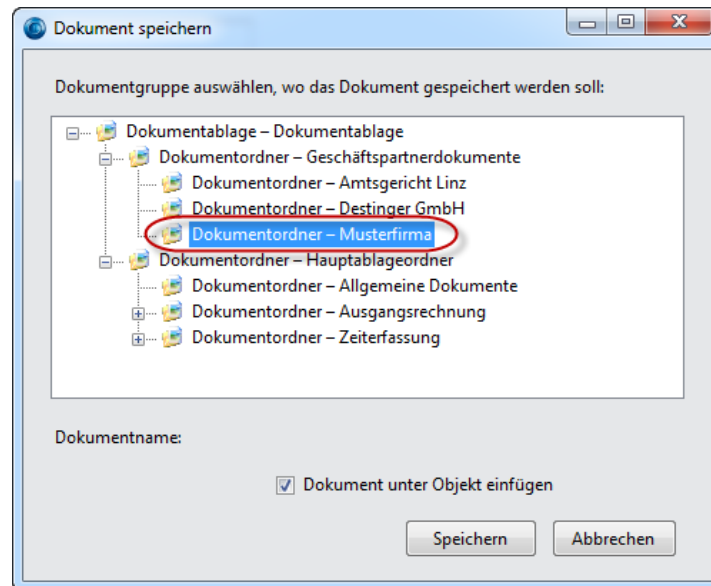


Abbildung 165: Dokument speichern - Dialog

### Neuen Geschäftskontakt anlegen

Ein Geschäftskontakt (Abbildung 166) kann ein Telefonat oder Notiz zu einem Ansprechpartner sein. Dazu muss ein Ansprechpartner markiert werden und über die Symbolleiste wird die Optionen zum Erzeugen angezeigt „Neu: Geschäftskontakt“.

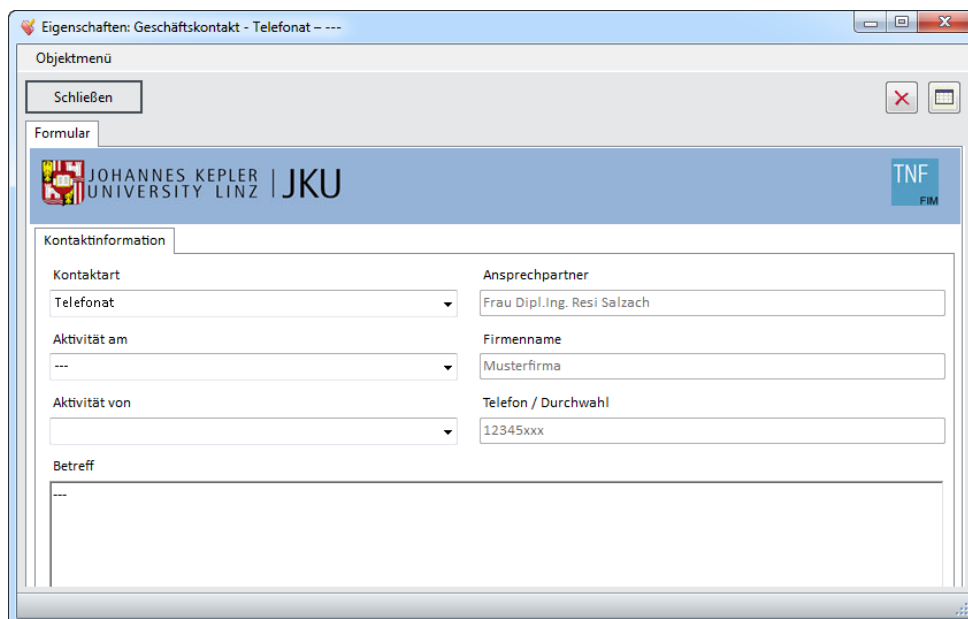


Abbildung 166: Neuen Geschäftskontakt erzeugen

Im Formular kann die Kontaktart, Aktivität von/am und einen Betreff eingeben werden.

## Neues Projekt erstellen

Nach markieren des Kunden im Baum erhält man die Möglichkeit über die Symbolleiste, ein neues Gutachtenprojekt zu erstellen.

The screenshot shows a software window titled 'Eigenschaften: Projekt - Projekt - 2012004'. It features a 'Formular' tab with three sub-tabs: 'Projektdaten', 'Kundendaten', and 'Controlling'. The 'Projektdaten' tab is selected and contains the following fields:

Projektnummer	Projektstatus	Projektverantwortung
2012004	In Arbeit	Gutachtenerzeuger Max
Bezeichnung	In Sachen	Projektart
---	---	Privatgutachten
---	Projektbeginn geplant	
<input type="checkbox"/> Zeitaufzeichnungsrelevant	28.06.2012	
	Projektende geplant	
	28.07.2012	

Abbildung 167: Neues Projekt erstellen

Im Reiter Projektdaten (Abbildung 167) können die allgemeinen Projektdaten eingetragen werden. Der Reiter Kundendaten zeigt Daten des Kunden an und es kann unter den vorher definierten Kundenansprechpartnern, ein Projektansprechpartner ausgewählt werden.

Der letzte Reiter Controlling dient um Einsicht zu nehmen, wie viele Stunden und Auslagen auf dieses Projekt gebucht wurden und um einen Kostenvorschuss zu definieren. Es kann auch ein Pauschalbetrag zu einem Gutachten vereinbart werden, welcher in diesem Reiter im Feld „Pauschalbetrag“ hinterlegt sein kann.

Es ist darauf zu achten, dass das Häkchen „Zeitaufzeichnungsrelevant“ gesetzt wird, damit Zeitaufzeichnungseinträge und Auslagen auf dieses Projekt gebucht werden können.

Um später in die Projektansicht „EXPERTISE\_Project“ zu wechseln genügt es auf das Objekt „Projekt“ im Baum einen Doppelklick zu machen.

## Neue Ausgangsrechnung erzeugen

Rechnungen können über die Symbolleiste durch Markieren des Kunden und anschließenden „Neu: Ausgangsrechnung“, oder aber auch aus den Vorlagen via Drag & Drop erzeugt werden.

Falls eine über die Symbolleiste erzeugt wird, müssen die einzelnen Rechnungspositionen erzeugt werden, dies geschieht über das Formular mit dem

Button „Neue Rechnungsposition“. Wird eine Rechnung über die Vorlagen importiert werden auch alle darin enthaltenen Position mitkopiert.

Rechnungsnummer	Rechnungsdatum	Ansprechpartner Kunde	EMail Ansprechpartner Kunde
27	21.05.2012	Frau Dipl.Ing. Resi Salzach	resi.salzach@asdasd.com
Gesamtsumme Netto	Umsatzsteuer	Gesamtsumme Brutto	Telefon Ansprechpartner Kunde
120,00 €	0,00 €	120,00 €	12355
<input checked="" type="checkbox"/> Automatische Positionsrenumerierung		Bearbeiter	Bearbeiter EMail
		Ing. Thomas Schneider	thomas.schneider@students.jku.at
			Bearbeiter Telefon
			+43 (732) 9015 - 5870

Abbildung 168: Neue Ausgangsrechnung erzeugen

Es gibt wiederum drei Reiter (Abbildung 168), der erste „Rechnungsdaten“ zeigt alle relevanten Daten und Preise an. Im nächsten „Rechnungsstatus“ gibt man ein Fälligkeitsdatum ein. Auch ob die Rechnung bereits bezahlt ist wird hier definiert und nachdem kann eine Rechnung auf „Archiviert“ gesetzt werden, dadurch wird diese nicht mehr unter dem Geschäftspartner dargestellt.

Über die Menüleiste „Ansicht → Archivierte Objekte darstellen“ können auch diese wieder sichtbar gemacht werden. Es kann notwendig sein, die Ansicht zu aktualisieren, dies geschieht mit der Taste F5.

Im letzten Reiter können ein Kopf- und Fußtext definiert werden, die vor und nach den einzelnen Positionen im Rechnungsbericht eingefügt werden.

Über den Button „Neue Rechnungspositionen“ werden die einzelnen zu verrechneten Gebühren angelegt.



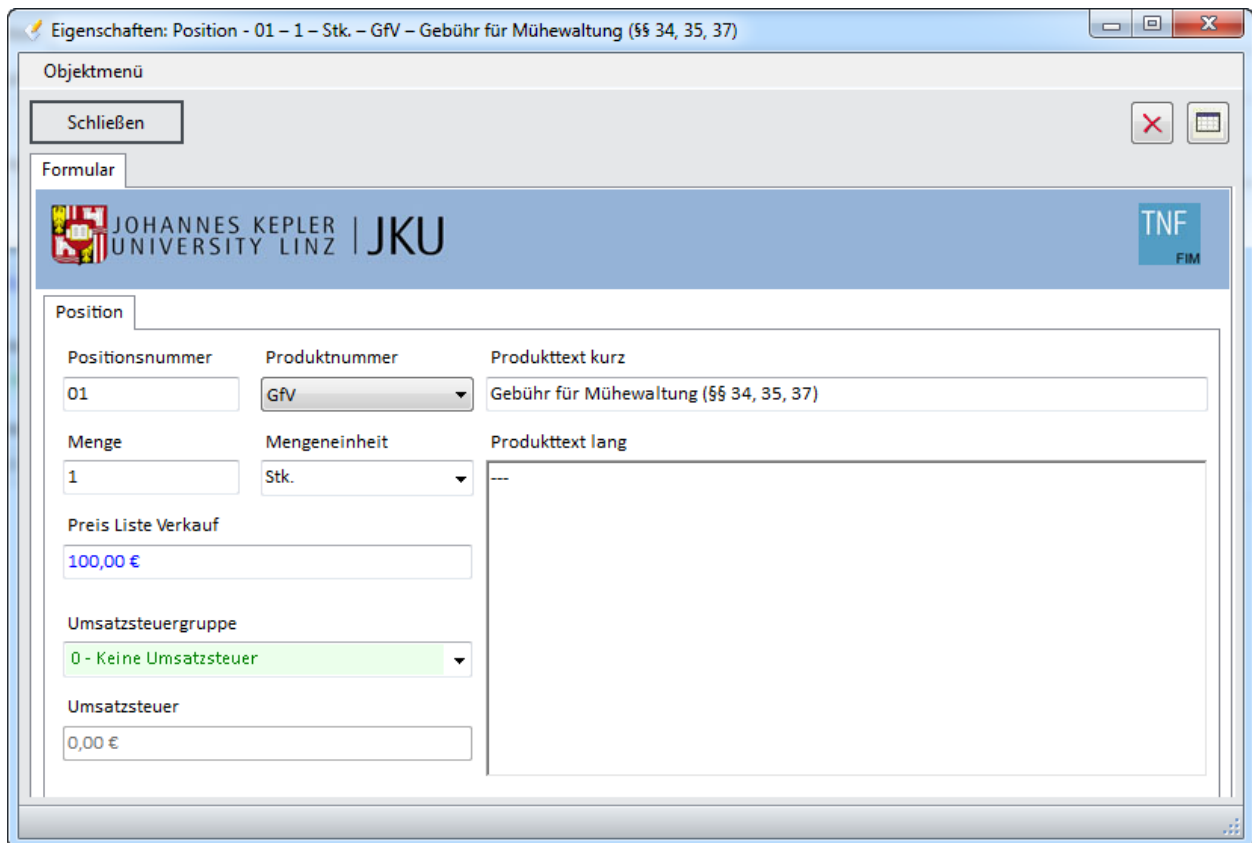


Abbildung 169: Neue Rechnungsposition erzeugen

Die Positionsnummer kann angepasst werden. Über die Produktnummer können in der Konfiguration definierte Produkte gesucht und ausgewählt werden. Die restlichen Werte werden dann automatisiert eingesetzt, siehe dazu Abbildung 169.

Im Baum sieht dann eine Rechnung wie folgt aus (Abbildung 170):

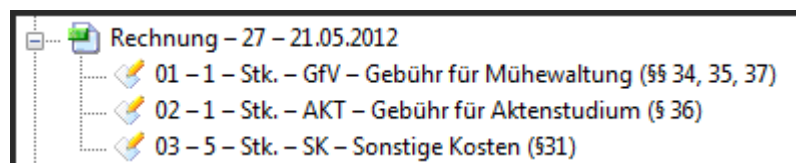


Abbildung 170: Rechnung mit Positionen

### Rechnungsbericht erzeugen

Über das Kontextmenü kann, wenn eine Rechnung markiert wird, ein Rechnungsbericht erzeugt und abgespeichert werden.

### EXPERTISE\_Project

Die Hauptansicht um an einem Gutachten zu arbeiten. Um in diese Ansicht zu gelangen, muss zuerst ein Geschäftspartner (=Auftraggeber) erstellt werden (siehe Kapitel: Neuen Geschäftspartner anlegen). Danach erstellt man eine neues Projekt (siehe Kapitel: Neues Projekt erstellen). Über einen Doppelklick auf dieses Projektobjekt gelangt man in die Ansicht EXPERTISE\_Project (Abbildung 171).

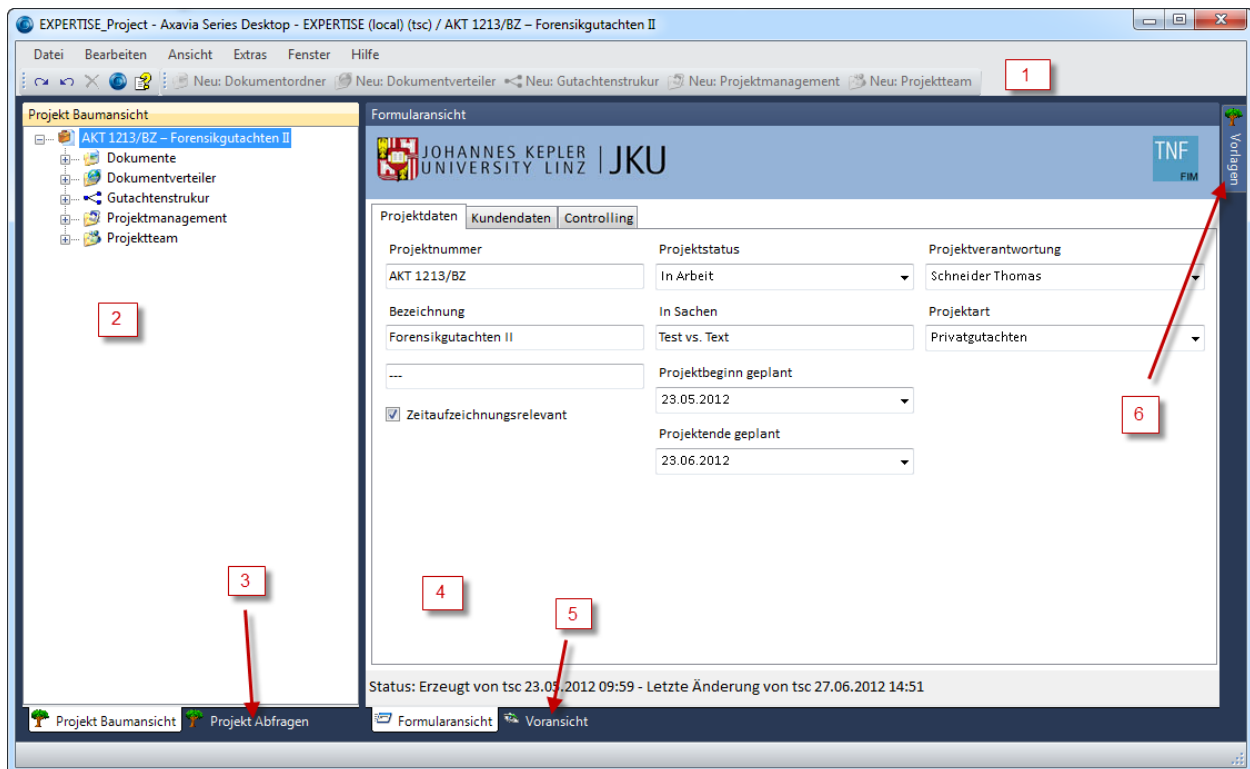


Abbildung 171: Ansicht "EXPERTISE\_Project"

- 1) Symbolleiste, diese dient zum Erzeugen von:
  - a. Dokumentordner
  - b. Dokumentverteiler
  - c. Gutachten
  - d. Aktivitäten / Checkliste
  - e. Projektteam
- 2) Projekt Baumansicht, in welcher alle Informationen zu einem Gutachten strukturiert dargestellt werden.
- 3) Projekt Abfragen, zur Suche von Dokumenten innerhalb eines Projektes
- 4) Formularansicht, zeigt Informationen der markierten Objekte aus der Baumansicht an.
- 5) Voransicht, kann Bilder, RTF-Dokumente & PDF-Dokumente (Adobe Reader muss installiert sein) darstellen.
- 6) Vorlagen, können benutzt werden, um bereits definierte Aktivitäten, leere Office-Dokumente, sowie fertig strukturierte Vorlagegutachten zu importieren.

### Dokumentordner erstellen

Über die Symbolleiste können einzelne oder auch verschachtelte Dokumentordner innerhalb eines Projektes erstellt werden. Alle für ein Gutachten relevanten Dokumente sollten hier abgelegt werden (Abbildung 172).

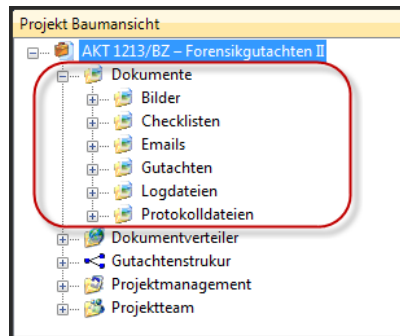


Abbildung 172: Dokumentordnerstruktur in einem Projekt

### Dokumentverteiler erstellen

Falls mitprotokolliert werden soll, welche Dokumente an welche Personen verschickt werden, kann diese Methode angewandt werden. Dafür ist es notwendig zuerst einen Dokumentverteiler zu erstellen, in diesem Dokumentverteiler können über die Symbolleiste Dokumentverteilereinträge erzeugt werden (Abbildung 173).

Zu einem Eintrag können Dokumente & Projektmitglieder per Drag & Drop verknüpft sein und dann ist es möglich ein Nachricht (E-Mail) zu generieren. Das Programm fragt nach, ob die Dateien komprimiert als Anhang zur Nachricht als Attachment abgelegt werden sollen. Alle Projektmitglieder, die zu diesem Eintrag verknüpft wurden, werden automatisch als E-Mail Empfänger eingetragen.

Nach dem Versenden der Nachricht wird die E-Mail, nachdem diese in einem Projektordner abgespeichert wurde, auch zum Dokumentverteilereintrag verknüpft.

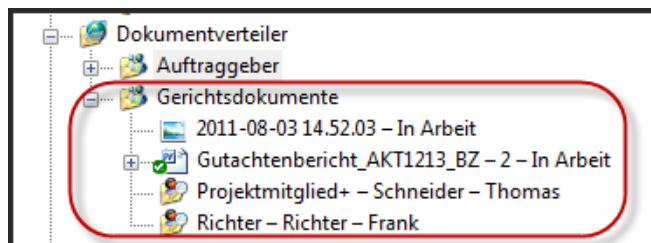


Abbildung 173: Dokumentenverteilerstruktur

### Gutachtenstruktur & -kapitel erzeugen

Um ein neues Gutachten zu erzeugen, markiert man das Projekt und wählt „Neue Gutachtenstruktur“ aus der Symbolleiste. Auf diesem Gutachtenstrukturobjekt, das die Titelseite im Bericht darstellt, können nun einzelne Gutachtenkapitel hinterlegt werden (Abbildung 174).

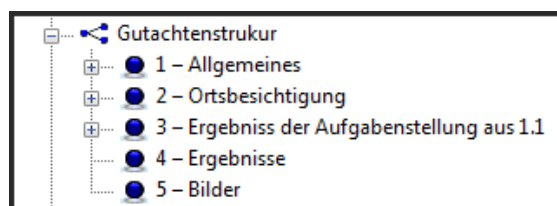


Abbildung 174: Gutachtenstruktur

Diese Kapitel können beliebig verschachtelt sein und stellen das Inhaltsverzeichnis im Gutachtenbericht dar. Änderungen an Kapitelnummer und der Überschrift können im

Formular getätigt werden. Es kann auch ausgewählt werden, ob vor einem Kapitel ein Seitenumbruch durchgeführt werden soll, oder nicht.

Im Formular selbst ist ein Doppelklick auf die Fläche notwendig um den Texteditor zu aktivieren (Abbildung 175).

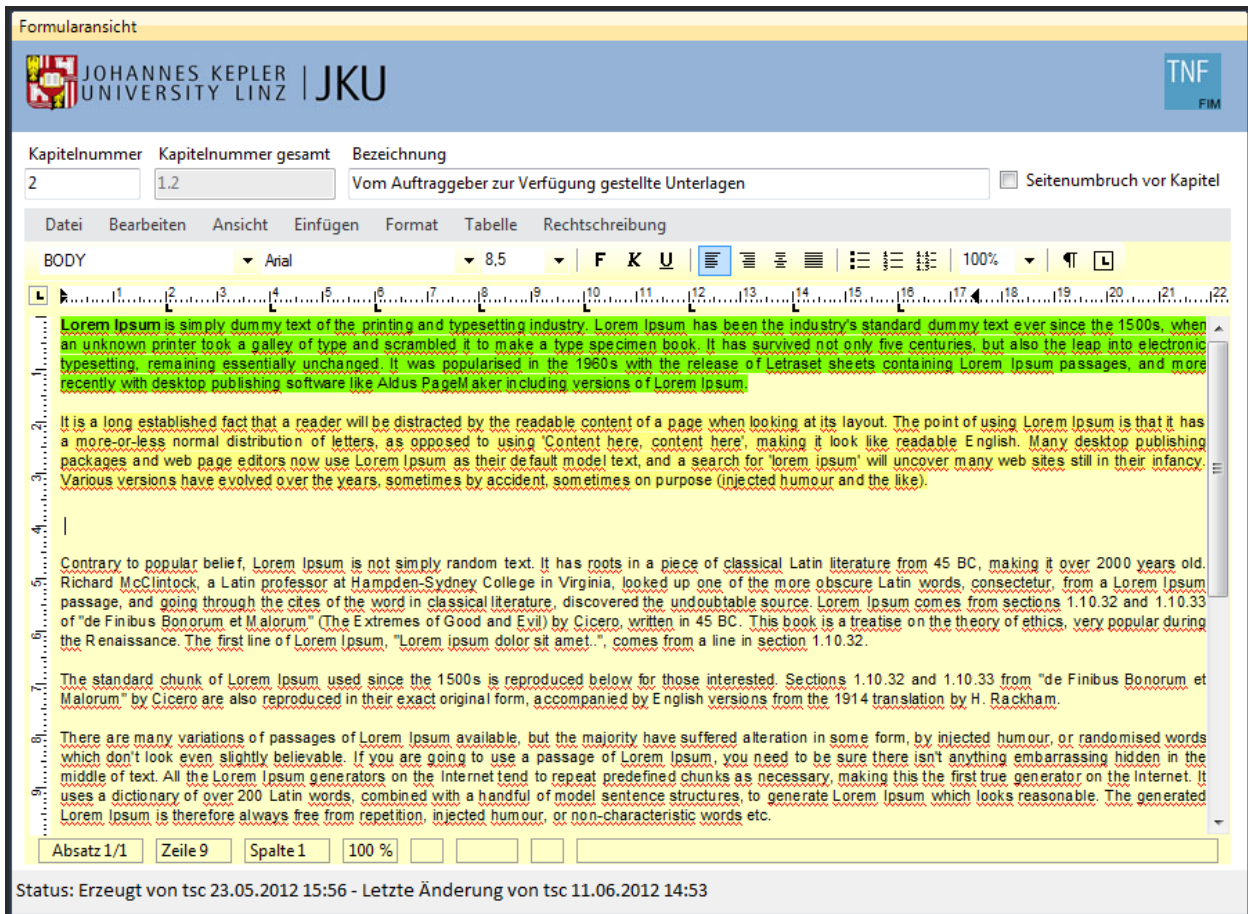


Abbildung 175: Texteditor für Kapitel

Es können Tabellen, Bilder und Text in diesem Texteditor eingefügt werden, die dann automatisch in der Datenbank abgelegt werden, ein explizites Speichern ist nicht notwendig.

### Schlagwörter definieren & anzeigen

Im Text ist es dann möglich, einzelne Wörter als Schlagwörter zu definieren, dazu markiert man das Wort und wählt über das Kontextmenü „Beschlagwortung“.

Über das Menü „Ansicht“ im Texteditor ist es möglich, die Schlagwörter welche man markiert hat, jeweils für das gewählte Gutachtenkapitel anzeigen zu lassen (Abbildung 176).

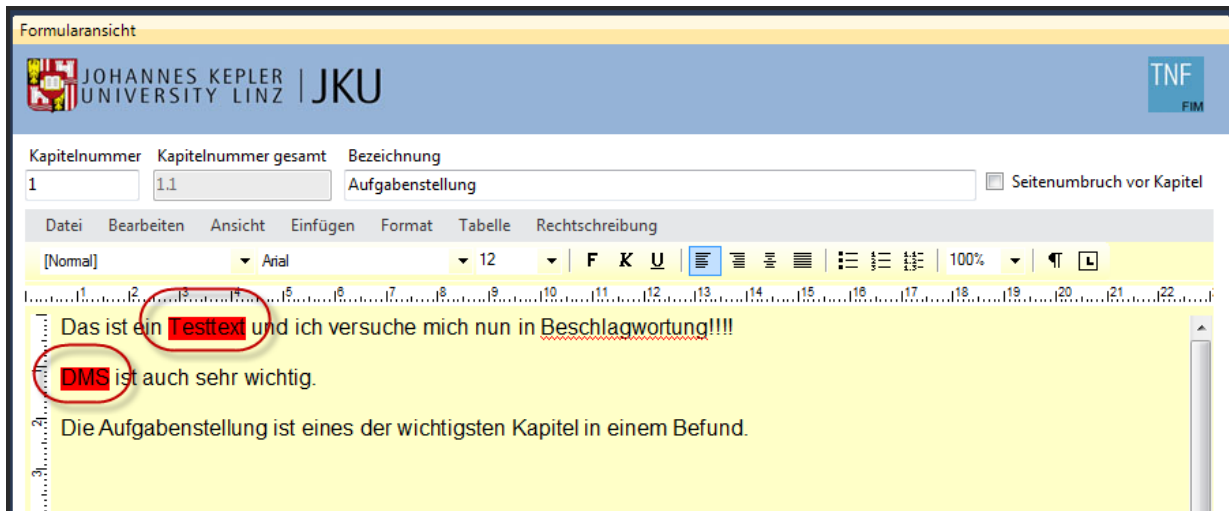


Abbildung 176: Schlagwörter definieren und anzeigen

### Schlagwörter im Bericht

Um Schlagwörter mit deren Definition im Bericht als Anhang anzuzeigen, genügt es die Gutachtenstruktur zu markieren und über das Kontextmenü „Schlagwortverzeichnis erstellen / updaten“ zu wählen (Abbildung 177).

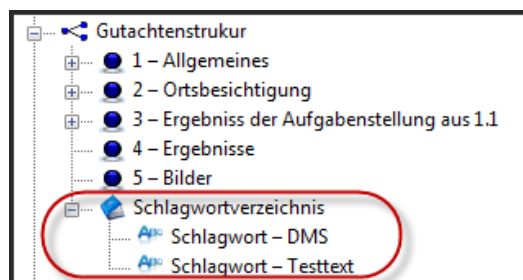


Abbildung 177: Gutachtenstruktur mit Schlagwortverzeichnisstruktur

Ist ein Schlagwort in der Datenbank bereits vorhanden, wird automatisch die Beschreibung im Formular sichtbar. Definiert man ein Schlagwort das erste Mal, so ist es möglich diese Definition über das Kontextmenü global abzuspeichern.

Auch das Entfernen eines Schlagwortes ist möglich, indem man das Kontextmenü auf dem Schlagwort ausführt.

### Gutachtenbericht erzeugen

Um die einzelnen Kapitel und Schlagwörter in einen Gutachtenbericht zu kombinieren, markiert man die Gutachtenstruktur und wählt über das Kontextmenü das Erzeugen bzw. Voransicht des Gutachtenberichtes.

Der erzeugte Bericht wird nach Auswahl eines Dokumentordners, auch noch zur Gutachtenstruktur verknüpft.

### Checklisten erzeugen

Damit Aktivitäten, die zu einem Gutachten gehören, nicht verschwinden, ist es ratsam sich eine Checkliste zu generieren, bzw. diese über die Vorlagen zu importieren.

Als Strukturierungsobjekt dient das Projektmanagement, das über die Symbolleiste, nach Auswahl des Projektes generiert wird. In diesem Projektmanagement ist es

möglich Aktivitäten zu erzeugen, diese können auch verschachtelt werden (Abbildung 178).



Abbildung 178: Projektmanagement mit Aktivitäten

Die einzelnen Aktivitäten können mit einer Reihenfolgen, Beschreibung, Datum und Status versehen werden. Über die Baumansicht Abfragen stehen noch zwei vordefinierte Abfragen zur Verfügung.

### Projektteam erzeugen

Unter einem Gutachtenprojekt können auch alle beteiligten Personen in Form von Projektmitgliedern abgelegt werden.

Dabei ist zu beachten, dass über Projektteilnahme Geschäftspartner auch Personen von anderen Kunden eingefügt werden können, ansonsten stehen nur die Ansprechpartner des Projekt-Geschäftspartners zur Auswahl (Abbildung 179).

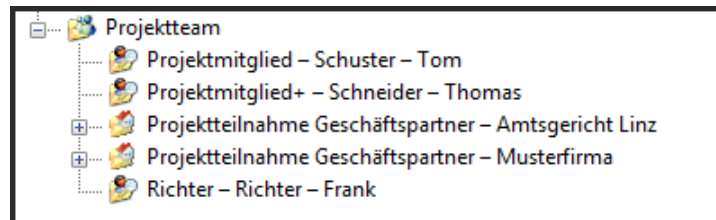


Abbildung 179: Projektteam mit Parteien

### E-Mail & Geschäftskontakt

Auch in dieser Ansicht ist es möglich Notizen, Telefonate und E-Mails zu versenden. Genauere Informationen dazu im Kapitel: Neue Nachricht(E-Mail) senden & Neuen Geschäftskontakt anlegen, der einzige Unterschied ist, das diese nicht zu Ansprechpartnern angelegt werden, sondern zu Projektmitgliedern.

### Projekt archivieren & importieren

Ist ein Gutachten abgeschlossen, kann dieses komprimiert und verschlüsselt in eine Datei abgelegt werden (Abbildung 180). Das Projekt – Objekt selbst bleibt im System erhalten und falls nötig, kann die Archivierung wieder rückgängig gemacht werden.

Um ein Projekt zu archivieren, wird es markiert und über das Kontextmenü der Befehl „Projekt archivieren“ aufgerufen. Ein Projekt kann nur archiviert werden, wenn der Status auf „Abgeschlossen“ geändert wird. Es wird nach einem Passwort, das auch zur Verschlüsselung dient, und nach dem Speicherort der Datei gefragt.

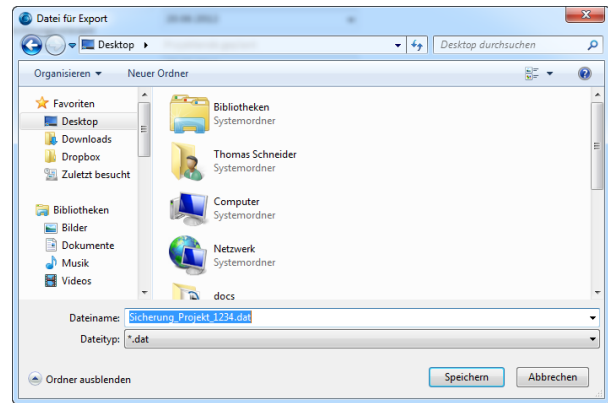
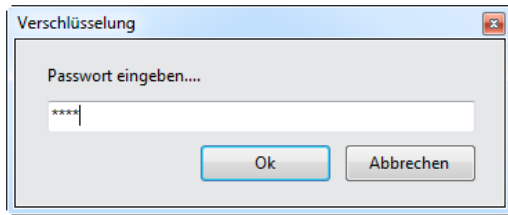



Abbildung 180: Projekt verschlüsseln und Datelexport - Dialog

Nach dem Archivieren befindet sich nur noch das Projekt, sichtbar mit dem Overlay Icon eines Schlosses, im System . Damit ein archiviertes Projekt wieder in das System importiert werden kann, ruft man das Kontextmenü auf dem Projekt auf und wählt „Archiviertes Projekt importieren“.

Achtung: Wenn das Passwort vergessen bzw. die Datei in seiner Integrität verändert wird, kann das Gutachtenprojekt nicht mehr hergestellt werden.

## EXPERTISE\_Time

Diese Ansicht (Abbildung 181) dient zum Eintragen von Zeiten und Auslagen, wobei die Zeiten direkt über das Kalendercontrol eingetragen werden, indem ein Zeitbereich markiert wird und die Auslagen über ein Formular definiert sind.

Die Zeiten und Auslagen werden sowohl einem Projekt zugeordnet, welches das Attribut „Zeitaufzeichnungsrelevant“ gesetzt hat und Zeiten können auch noch einem Zeitkonto zugehörig sein.

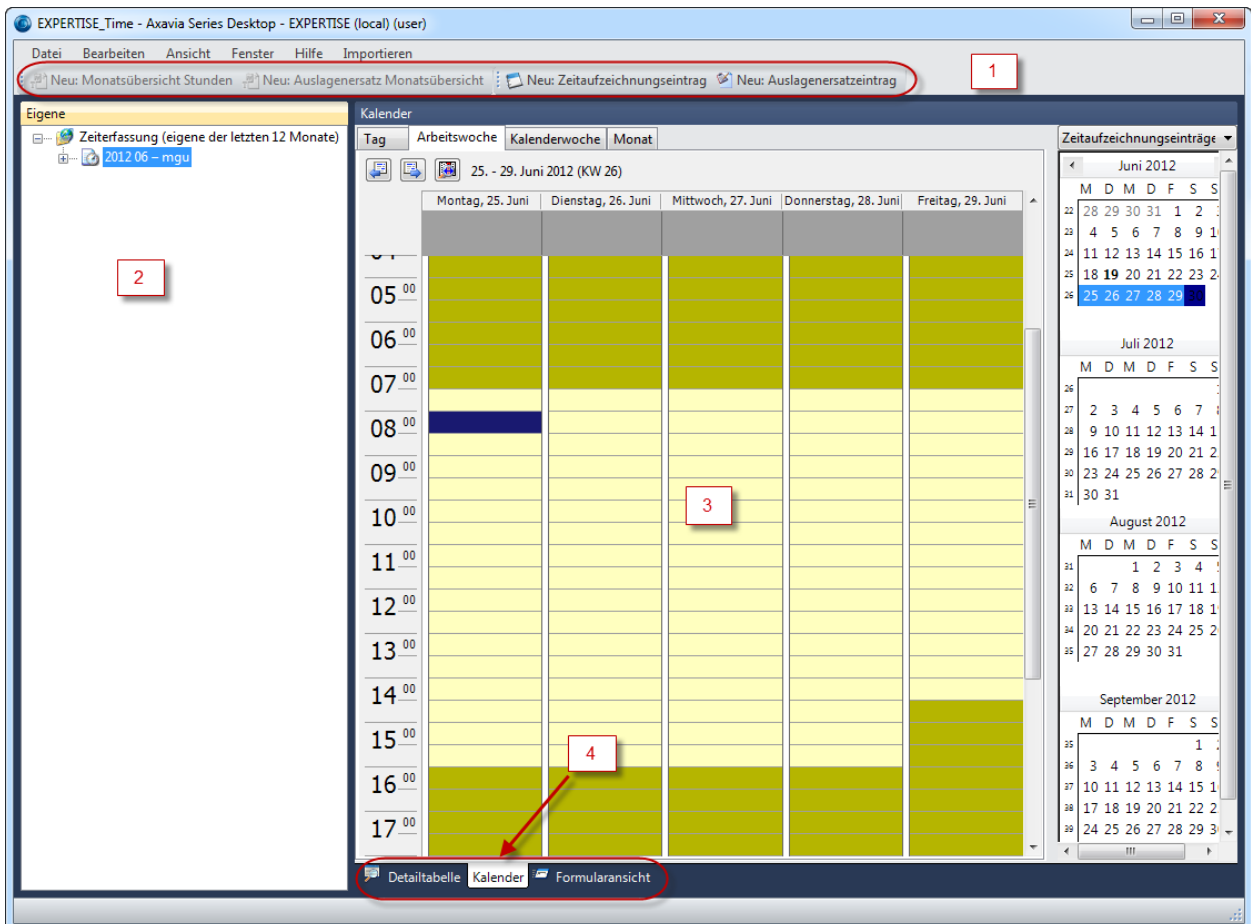


Abbildung 181: Ansicht "EXPERTISE\_TIME"

- 1) Symbolleiste, in der man Zeitaufzeichnungs- & Auslagenersatzeinträge erzeugen kann aber auch, wenn ein Monat in der Baumansicht markiert ist, jeweils ein Bericht für Stunden und Auslagen erzeugt wird.
- 2) Baumansicht, hier wird eine Abfrage die letzten zwölf Monate angezeigt, wobei unter den Monaten die Tage & Einträge sichtbar sind. Diese sind nötig, um später in der Detailtabelle eine Auflistung dieser sichtbar zu machen.
- 3) Kalendercontrol, kann verwendet werden um Zeitaufzeichnungseinträge zu generieren, indem ein Zeitbereich markiert wird und über das Kontextmenü der Befehl „Zeitaufzeichnungseintrag“ gewählt wird.
- 4) Es gibt auch noch zwei weitere Reiter, der eine zeigt in tabellarischer Form die Einträge an, in der auch eine Aggregationsfunktion zum Aufsummieren enthalten ist. Der Formularreiter kann verwendet werden, um die in der Baumansicht markierten Einträge zu sichten.



## Zeitaufzeichnungseintrag erstellen

Es gibt zwei Möglichkeiten einen Eintrag zu erzeugen. Entweder über die Symbolleiste, wobei hier definiert werden muss, von wann bis wann dieser Eintrag gilt, oder direkt im Kalenderdialog. Die zweite Möglichkeit setzt für die erwähnten Attribute bereits die richtigen Werte.

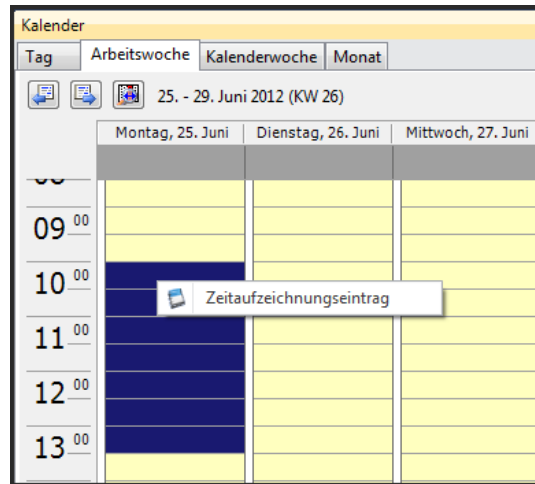
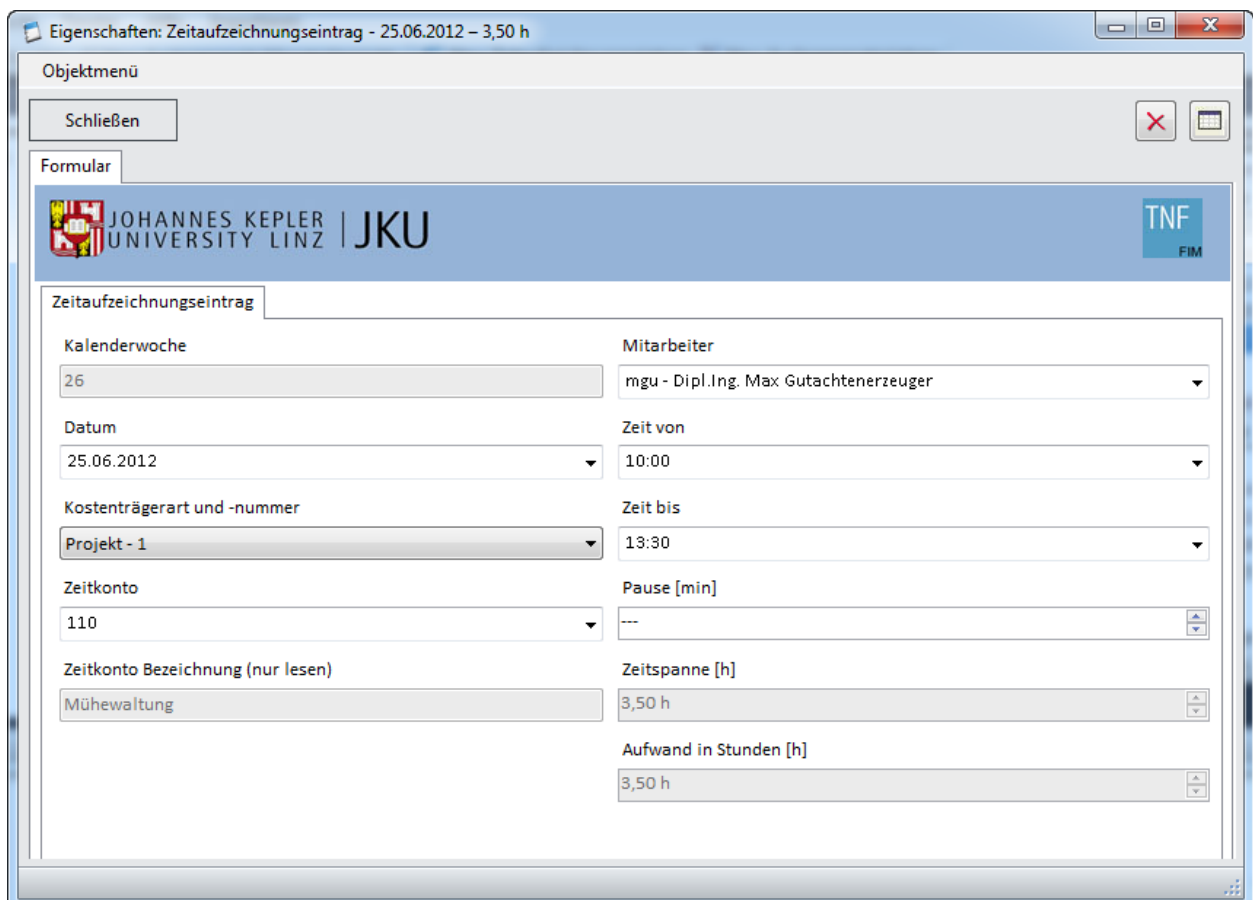


Abbildung 182: Zeitaufzeichnungseintrag im Kalenderdialog erstellen

Wählt man den Bereich aus, kann man über das Kontextmenü den Eintrag erzeugen (Abbildung 182). Es erscheint ein Dialog (Abbildung 183) in dem noch die fehlenden Informationen eingetragen werden können.

The image shows a dialog box titled 'Eigenschaften: Zeitaufzeichnungseintrag - 25.06.2012 - 3,50 h'. It has an 'Objektmenü' at the top with a 'Schließen' button. Below is a 'Formular' section with the JKU logo and TNF FIM branding. The form contains the following fields:

- Kalenderwoche: 26
- Mitarbeiter: mgu - Dipl.Ing. Max Gutachtenerzeuger
- Datum: 25.06.2012
- Zeit von: 10:00
- Kostenträgerart und -nummer: Projekt - 1
- Zeit bis: 13:30
- Zeitkonto: 110
- Pause [min]: ---
- Zeitkonto Bezeichnung (nur lesen): Mühewaltung
- Zeitspanne [h]: 3,50 h
- Aufwand in Stunden [h]: 3,50 h

Abbildung 183: Zeitaufzeichnungseintrag - Eigenschaftendialog

## Auslagensatzeintrag erstellen

Dieser wird über die Symbolleiste erzeugt. Nach dem Erzeugen trägt man die Auslagensumme, Auslagentext ein und verknüpft diesen Betrag zu einem Projekt (Abbildung 184).

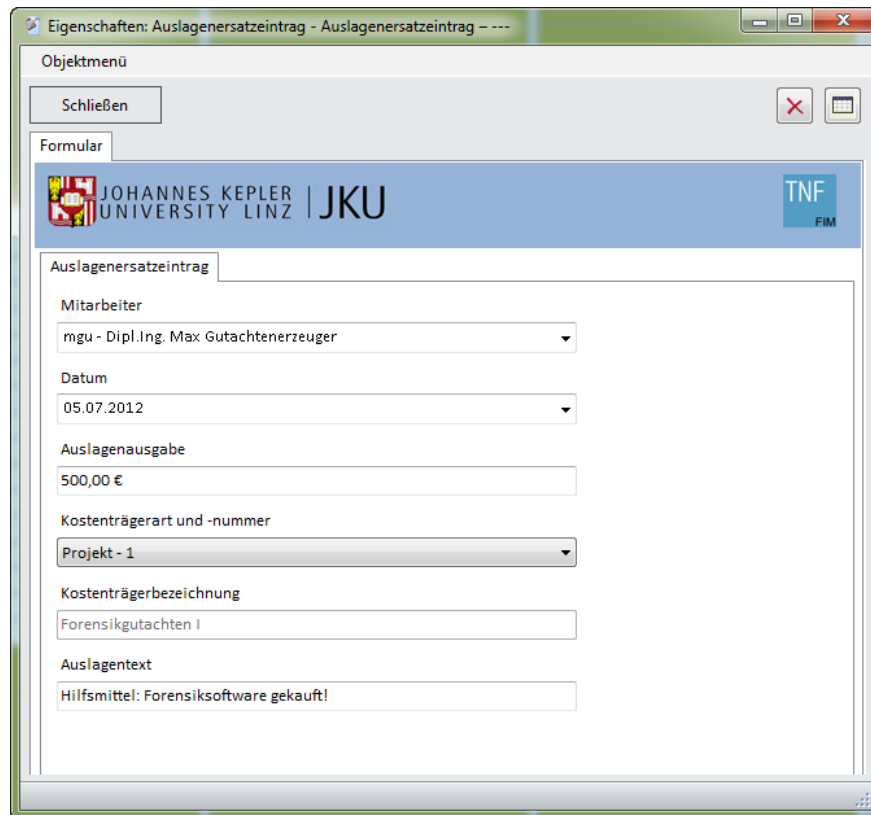


Abbildung 184: Auslagenerstarteintrags - Dialog

## Monatsstundenbericht erzeugen

Um eine Übersicht Ihrer Stunden zu erhalten, markiert man in der Baumansicht einen Monat und wählt, entweder über die Symbolleiste oder das Kontextmenü den Monatsreport aus.

## Auslagensatzbericht erzeugen

Auch für die Auslagen kann man sich einen Bericht erstellen lassen, in dem die gleiche Vorgehensweise wie für den Monatsbericht durchführt.

## 12.4 Lebenslauf

### Schneider Thomas

geboren 10. März 1976 in Vöcklabruck

Dorf 8, 4845 Rutzenmoos

[thomas.schneider@axavia.com](mailto:thomas.schneider@axavia.com)

### Ausbildung

- Volksschule Regau 1982-1986
- Sporthauptschule Vöcklabruck 1986-1990
- HTL Vöcklabruck Maschinenbau 1990-1995
- Studium der Informatik an der JKU Linz seit 1995

### Berufserfahrung

#### Ferialpraktika während HTL

- Stangl Technik Regau (Montagearbeiten)
- Doubrava Attnang Puchheim (Arbeiten Bohrwerk)
- 3 Jahre TKV Regau (Technischer Zeichner)

#### Berufstätigkeiten während / neben dem Studium

- Netzwerkadministrator Studentenheim JKH Linz 1997-2002
  - Aufbau / Installation und Wartung des Netzwerkes für 400+ Studenten
- Geringfügige Beschäftigung: samo Maschinenbau GmbH von 2003 – 2009
  - Entwicklung der internen Auftragsabarbeitung
  - Netzwerkkonfiguration & Systemadministrator
  - Service & Wartung der IT
- Freier Dienstnehmer: AXAVIA Software GmbH 2005 - 2010
  - CAPE „Computer-Aided Project Engineering“ für Siemens VAI Linz, Prozess- & Modellentwicklung für Mechanisches- & Elektrisches Engineering von Stranggussanlagen.
  - p-PoD für IPS Orbital Schweißtechnik GmbH, Prozess- & Modellentwicklung um Fortschritt, Qualität & Produktivität bei Projekten sicherzustellen.
  - EDB – Engineering Database Einführung bei diversen anderen Unternehmen
  - PM / CRM / DMS Prozessentwicklung

- Angestellter: AXAVIA Software GmbH seit 2010
  - Projektabwicklung im Bereich Anlagenbau
  - Entwicklung und Implementierungen von Prozessen im Anlagenbau
  - Integration von Geschäftsprozessen (Auftragsabarbeitung) im Anlagenbau

## **12.5 Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtliche und sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Masterarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

21.03.2013



(Schneider Thomas)