

## **Zusammenfassung**

In dem cids basierten System WUNDa, das in der Stadtverwaltung Wuppertal eingesetzt wird, wurde für ein Teilsystem (Abgabe der Produkte des Liegenschaftskatasters) eine Abrechnungskomponente entworfen und implementiert. Für den wichtigen Bereich der Vermessungsunterlagen umfassen diese Produkte zum einen Auszüge aus dem Amtlichen Liegenschaftskatasterinformationssystem ALKIS (Flurkarte, Flurstücks- und Eigentüternachweise, Punktlisten, NAS-Daten), zum anderen durch Scannen analoger Originale entstandene digitale Dokumente (Fortführungsrisse, Grenzniederschriften, Aufnahmepunktkarten (AP-Karten), Nivellementspunkt-Beschreibungen (NivP-Beschreibungen), AP-Übersichten etc.). Die erforderlichen online-Systeme zur Selbstentnahme dieser Produkte wurden über mehrere Vergaben seit 12/2008 in Form von WuNDA-Fachverfahren und -Fachthemen hergestellt.

Auf Grund fachrechtlicher Vorgaben und wegen datenschutzrechtlicher Belange müssen die von den Endkunden vorgenommenen Datenentnahmen durch die Stadt Wuppertal protokolliert werden. Dabei müssen einige vom externen Nutzer zu liefernde Angaben abgefragt werden, z. B. der Verwendungszweck der Daten - hieraus ergibt sich, ob die Datenentnahme kostenfrei oder kostenpflichtig ist - und die Nummer, unter der er die Datenentnahme in seinem Geschäftsbuch führt.

Um das volle Rationalisierungspotenzial der Selbstentnahmen ausschöpfen zu können, müssen auch gebührenpflichtige Datenentnahmen unterstützt werden. Dafür müssen die WuNDA-Beschreibungsseiten, aus denen heraus Datendownloads ermöglicht werden, um weitere Funktionen und Oberflächenelemente zur Gebührenberechnung, Gebührenanzeige und zum Akzeptieren der Gebührenforderung ergänzt werden. Jedes Herunterladen solcher Daten führt zu einem Protokolleintrag in der sog. „Billing“-Tabelle der WuNDA-Datenbank. Die Lösung ermöglicht es, die Prozesse rund um die Selbstentnahme von Vermessungsunterlagen medienbruchfrei zu gestalten und die betroffenen WuNDA-Teilsysteme mit dem Geschäftsbuch des Ressorts 102 zu verknüpfen.

Aus der Billing-Tabelle werden turnusmäßige Abrechnungen erzeugt, wobei je nach Kundengruppe und Verwendungszweck der Daten monatliche, quartalsweise und jährliche Abrechnungen möglich sind. Die Abrechnungen enthalten eine Liste aller Downloads des jeweiligen Nutzers mit den zugehörigen Gebühren bzw. den geschuldeten Entgelten. Die Unterstützung dieser Abrechnungen wird vom System genauso unterstützt wie die Anzeige der "Kontoinformationen" für den Endkunden der Datenendnahme.

Die Arbeit gibt zunächst einen Überblick über die abzugebenden Produkte und ihre Verwendung in den typischen Prozessen der Kunden der Stadtverwaltung. Anschließend wird die Integration des Billing Mechanismus in die Fachsysteme, die Preisermittlung und die Protokollierung der Produktentnahmen erläutert. Der Auswertungsteil der Software wird aus der Sicht der Endkunden und aus der Sicht der Fachabteilungen erläutert. Die Benutzerinteraktion und die betroffenen GUI Elemente werden beschrieben.

Im abschließenden Teil der Arbeit wird auf die wirtschaftlichen Aspekte (Einsparungen) in der Stadtverwaltung eingegangen und diese werden anhand eines Beispiels veranschaulicht. Im Ausblick wird die Integrationsmöglichkeit von cids/WuNDa in weitere Prozesse der Endkunden (Beschränkung auf ÖbVi, Sparkassen und Polizei) untersucht und die notwendigen weiteren Entwicklungsschritte skizziert.

Amtliche Geodaten der Stadt Wuppertal,  
© Ressort Vermessung, Katasteramt und Geodaten (**RV-102-** )

# 1 Einführung

In diesem einführenden Kapitel wird der aktuelle Zustand bezüglich WuNDa und seinen externen Benutzern beschrieben. Danach wird das, sich durch eine gewünschte Rationalisierung stellende, Problem erläutert und ein Lösungsansatz für dieses Problem wird vorgestellt.

## 1.1 Überblick über den aktuellen Zustand

Bei dem Wuppertaler Navigations- und Datenmanagementsystem (WuNDa) handelt es sich um ein Softwaresystem, das es der wuppertaler Stadtverwaltung ermöglicht auf geographisch-räumliche Informationen zuzugreifen [vgl. Cis13a]. Durch diesen Schwerpunkt findet WuNDa häufig Anwendung im Ressort 102, dem Ressort für Vermessung, Katasteramt und Geodaten.

Weiterhin stellt die Stadtverwaltung einen Online-Zugriff auf WuNDa bereit, so dass auch Benutzer, die keine Beamte der Stadtverwaltung sind, auf dieses System zugreifen können. Diese Gruppe von Benutzern wird im Folgenden externe Benutzer genannt. Zu diesen externen Benutzer gehören u. a. die Polizei, Sparkasse, Notare und insbesondere Öffentlich bestellte Vermessungsingenieure (ÖbVIs).

Die, für diese Arbeit wichtigste, Funktionalität des Online-Zugriffs ist die sogenannte Selbstentnahme von Produkten. Dabei können externen Benutzer amtliche Auszüge aus dem Liegenschaftskataster selbst beziehen. Dies bedeutet einerseits für die externen Benutzer, dass der Weg zum Katasteramt gespart wird und andererseits bedeutet es für das Katasteramt, dass die ansonsten benötigte Bearbeitungszeit eingespart wird [vgl. WUPd].

## 1 Einführung

Am Beispiel der ÖbVIs zeigt sich, dass es gebührenfreie und gebührenpflichtige Produkte gibt. So können die ÖbVIs bestimmte Produkte, die im Rahmen von der Anfertigung von Amtlichen Lageplänen und der Durchführung hoheitlicher Vermessungen benötigt werden, kostenfrei anfordern. Besteht allerdings Bedarf nach anderen Produkten so sind diese kostenpflichtig [vgl. WUPc]. Mehr Informationen hierzu befinden sich im Abschnitt 2.1.4.

Die Selbstentnahme unterstützt neben den kostenfreien Produkten auch kostenpflichtige Produkte. Dadurch wird es erforderlich, dass jedes bezogene Produkt protokolliert wird. Somit kann nach einem, vom Benutzer abhängigen, Abrechnungsturnus eine Abrechnung an die jeweiligen externen Benutzer gestellt werden. Eine solche Abrechnung für einen Benutzer enthält eine Auflistung seiner Produkte, die er während eines Abrechnungsturnus bezogen hat, und den entsprechend anfallenden Gebühren. Das Erstellen der Abrechnungen geschieht manuell durch einen Beamten des Katasteramtes, in dem sämtliche Produktprotokolle mit einem Tabellenkalkulationsprogramms so gefiltert werden, dass ausschließlich die gewünschten Protokolle für die Abrechnung eines bestimmten externen Benutzer und einen Abrechnungsturnus übrig sind. Durch dieses Verfahren ist das Erstellen der Abrechnungen arbeitsaufwändig und fehleranfällig [vgl. San13].

## 1.2 Rationalisierung

Die Selbstentnahme wurde einerseits wegen des Erlasses zur Online-Bereitstellung vom 2.2.2011 für die ÖbVIs eingeführt [vgl. WUPc], andererseits wegen eines Rationalisierungsplans, der zwischen dem Ressort 102 und dem Wuppertaler Stadtdirektor vereinbart wurde. Dieser Plan schreibt vor, dass bis Ende 2014 etwa zweieinhalb Vollzeitkräfte wegfallen und nicht wieder besetzt werden.

Der geplanten Rationalisierung steht allerdings im Weg, dass die Methode zur Erstellung der Abrechnungen arbeits- und zeitintensiv ist. Nichtsdestotrotz werden erhebliche Anstrengungen unternommen, um die Gruppe der externen Benutzer zu vergrößern. Durch das Wachsen dieser Gruppe, wächst allerdings auch der Aufwand der mit dem Erstellen der Abrechnungen einhergeht, folglich wird dieser noch zeitintensiver.

## 1 Einführung

Die Problemstellung lautet deshalb wie folgt: die Rationalisierung kann nur dann vollständig durchgeführt werden, falls die Methode zur Erstellung der Abrechnungen so verändert werden kann, so dass sie weniger zeitaufwändig ist [vgl. San13].

### 1.3 Lösungsansatz

Der, in dieser Arbeit verfolgte, Lösungsansatz ist es die Methode zur Erstellung der Abrechnungen möglichst zu automatisieren, so dass der Arbeitsaufwand für die Beamten minimiert wird.

Der naheliegendste Ansatz ist es WuNDa für diese Automatisierung zu erweitern, da über dieses System bereits die Selbstentnahme bereitgestellt wird und die dadurch anfallende Gebühren protokolliert werden. Dies bedeutet dass für das Erreichen des Ziels mindestens eine Übersicht und Auswahl für die einzelnen protokollierten Produkte zur Verfügung stehen muss, weiterhin muss ein automatisches Erstellen der Abrechnungen dieser Produkte möglich sein. Hierbei handelt es sich lediglich um minimal Anforderungen, im Kapitel 3 wird detaillierter auf die benötigten und zusätzlich gewünschten Erweiterungen von WuNDa eingegangen. Daraufhin wird im Kapitel 4 die Umsetzung der Erweiterungen in WuNDa beschrieben.

Im nächsten Kapitel wird die bereits vorgestellte Arbeit der externen Benutzer und die beschriebene Rationalisierung beleuchtet und in Verbindung gebracht.

**TODO: Stimmt das?**

### 1.4 Aussicht

Der Abschluss dieser Arbeit besteht aus Kapitel 5, hier werden die vorher gesammelten Informationen benutzt um Teile der Workflows ausfindig zu machen, die durch WuNDa unterstützt werden könnten. **TODO: Später erweitern und anpassen.**

## **2 Beleuchtung der Arbeitsweise von externen Benutzer**

In diesem Kapitel soll aufgezeigt werden, warum die geplante Rationalisierung mit dem aktuellen Stand von WuNDa nicht durchzuführen ist.

Um dies zu erreichen müssen in einem ersten Schritt einige fachliche Erläuterungen erfolgen. So werden einige Begriffe definiert und die Gebühren zu den Produkten erläutert. Daraufhin werden die Produkte aufgelistet, die von den externen Benutzern bezogen werden können.

Im Anschluss werden einige Workflows von externen Benutzern beleuchtet, woraufhin überprüft wird ob und wie WuNDa diese unterstützt. Demnach wird der aktuelle Stand von WuNDa bezüglich der bestehenden Problematik beschrieben.

Im letzten Teil des Kapitels wird mittels der zuvor gewonnen Erkenntnisse verdeutlicht, dass WuNDa dem Problem entgegenwirkend erweitert werden kann.

### **2.1 Fachliche Erläuterungen**

In diesem Abschnitt werden einige fachliche Begriffe und Zusammenhänge erläutert, die für das Verständnis dieses Kapitel von Bedeutung sind.

### 2.1.1 ÖbVI

Damit hoheitliche Vermessungen nicht nur durch Verwaltungen durchgeführt werden können, kann der Staat private Vermessungsingenieure mit hoheitlichen Aufgaben beleihen. Solche Freiberufler werden Öffentlich bestellte Vermessungsingenieure (ÖbVIs) genannt und sind berechtigt Katastervermessungen durchzuführen. Dies sind Vermessungen, die der Einrichtung und Fortführung des Liegenschaftskatasters und der Feststellung oder Abmarkung von Grundstücksgrenzen dienen [vgl. BDV99]. Weiterhin dürfen sie Grenzzeichen setzen oder entfernen [vgl. Wolb].

### 2.1.2 Festpunkte

Festpunkte sind Vermessungspunkte die laut dem sächsischen Staatsministerium für Umwelt und Landwirtschaft [Geoa] wie folgt definiert werden können: „Festpunkte der Grundlagenvermessung sind vermarkte, gesicherte und örtlich eingemessene Vermessungspunkte mit präzise bestimmten und im amtlichen Nachweis geführten Koordinaten, Höhen und Schwerewerten.“ Weiterhin wird zwischen Lage-, Höhen- und Schwerfestpunkten unterschieden [vgl. AdV13b].

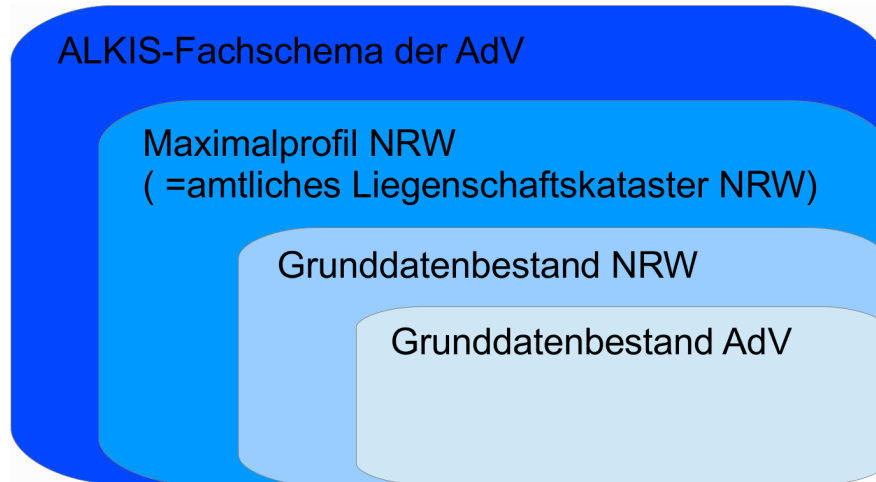
### 2.1.3 ALKIS und sein Datenmodell

Das Amtliche Liegenschaftskatasterinformationssystem (ALKIS) ersetzt das Automatisierte Liegenschaftsbuch (ALB) und die Automatisierte Liegenschaftskarte (ALK), indem es die Daten beider Systeme vereint. Die AdV hat ALKIS entwickelt und organisiert die Pflege des Datenmodells [vgl. AdV13a].

Das Datenmodell von ALKIS ist stufenweise aufgebaut. Die oberste Stufe ist das ALKIS-Fachschemata der AdV, dieses Fachschema bildet eine Grundlage mit der sämtliche Objekte, Attribute und Attributwerte aller Katasterbehörden Deutschlands abgebildet werden können. Die folgenden Stufen sind Reduktionen dieses Fachschemas, d. h. sie benutzen nur einen Teil dieser Grundlage. Weiterhin ist die jeweils folgende Stufe eine Reduktion der vorhergegangenen Stufe [vgl. Bez09c,



S. 1-4]. Eine Übersicht der einzelnen Stufen kann der Abbildung 2.1 entnommen werden.



**Abbildung 2.1: Erweiterungen der Datenbestände**

In Nordrhein-Westfalen (NRW) wurde das Maximalprofil NRW festgelegt, da hier einige Objekte aus dem ALKIS-Fachschema der AdV nicht benötigt werden. Alle Informationen die im Maximalprofil NRW geführt werden gehören zum amtlichen Liegenschaftskataster. Weiterhin dürfen in NRW nur diese Informationen innerhalb ALKIS geführt werden. Jedes Katasteramt darf frei wählen welche Informationen es aus dieser Stufe führen will, vorausgesetzt diese Informationen gehören nicht zu einer unteren Stufe. Die nächste Stufe bildet der Grunddatenbestand NRW dieser ist landesweit gültig und muss von jedem Katasteramt in NRW geführt werden. Die unterste Stufe bildet der Grunddatenbestand AdV, dieser ist bundesweit gültig und muss von jedem Katasteramt in Deutschland geführt werden [vgl. Bez09c, S. 1-4].

### 2.1.4 Gebühren der Produkte

In diesem Abschnitt wird das Zustandekommen der Gebühren der Produkte kurz beschrieben. Da die genaue Ermittlung der Gebühren nicht Bestandteil dieser Arbeit ist, wird hier nur auf das Nötigste eingegangen.

Die Gebühren der einzelnen Produkte wird durch eine Grundgebühr bestimmt, auf die verschiedene Rabatte angewendet werden können. Ein Rabatt kann z. B.

## 2 Beleuchtung der Arbeitsweise von externen Benutzer

ein Mengenrabatt sein oder ein Rabatt der durch eine Regelung gegeben werden muss.

Eine Regelung die insbesondere die ÖbVIs betrifft ist in der Vermessungs- und Wertermittlungsgebührenordnung (VermWertGebO) festgelegt. Laut dieser muss die Selbstentnahme der Produkte, die zur Durchführung bestimmter Amtshandlungen benötigt werden, kostenfrei sein. Zu diesen Handlungen gelten die Anfertigung von Amtlichen Lageplänen und die Durchführung hoheitlicher Vermessungen [vgl. WUPc]. Die Produkte die diese Bedingungen erfüllen werden im folgenden vermessungsrelevant genannt. Der Rabatt der bei diesen Produkten gegeben wird, wird „zweckabhängiger Rabatt“ genannt. **TODO: evt. noch MwSt erwähnen. Rabatt der Selbstentnahme erwähnen.**

Weitere Informationen können dem Dokument „Entgeltordnung Ressort 102“ [WUP13b] entnommen werden.

## 2.2 Auflistung der Produkte

In diesem Abschnitt werden die Produkte aufgelistet, welche die externen Benutzer selbst beziehen können. Diese Produkte sind in vier Produktklassen eingeteilt:

- ALKIS-Standardausgaben
- Standardausgaben WUP-kommunal
- WuNDa-Berichte
- Daten Liegenschaftskataster

Diese Produktklassen unterteilen die Produkte logisch und geben an von welcher Stelle diese definiert wurden.

### 2.2.1 ALKIS-Standardausgaben

In der Produktklasse ALKIS-Standardausgaben wird zwischen landesweit einheitliche Standardausgaben und kommunalen Produkten unterschieden. Bei den Standardausgaben handelt es sich um Auszüge aus dem Liegenschaftskataster deren Inhalt und Form landesweit festgelegt ist. Diese Ausgaben beinhalten nur Daten die im Grunddatenbestand NRW definiert sind [vgl. Bez09c, S. 8].

Bei den kommunalen Produkten handelt es sich um Auszüge aus dem Liegenschaftskataster, allerdings dürfen auch Daten verwendet werden, die nicht im Grunddatenbestand NRW definiert sind. Ein Beispiel eines kommunalen Produktes wäre eine modifizierte Standardausgabe, die mit Daten aus dem Maximalprofil NRW erweitert wurde. Weiterhin muss die Bezeichnung bzw. Beschriftung deutlich gekennzeichnet sein, so dass eine Verwechslung mit den Standardausgaben ausgeschlossen ist [vgl. Bez09c, S. 9]. Dies wird durch die Anhänge (*NRW*) bzw. (*kommunal*) garantiert.

Die Produkte der Produktklasse ALKIS-Standardausgaben können der Tabelle 2.1 auf den Seiten 11–14 entnommen werden.

**Tabelle 2.1: Produkte der Produktklasse ALKIS-Standardausgaben**

Produktname	Beschreibung
Flurstücksnachweis	„Im „Flurstücksnachweis“ werden alle flurstücksrelevanten Angaben beschrieben. Dies sind neben der „katastertechnischen Bezeichnung des Flurstücks“ „Gebietszugehörigkeit“, „Lage“, „Fläche“, „Tatsächliche Nutzung“, „Klassifizierung nach Straßen- oder Wasserrecht“, „Gebäude“, „Hinweise zum Flurstück“, „Buchungsart“ und „Buchung:“ [AdV06, S. 269]

## 2 Beleuchtung der Arbeitsweise von externen Benutzer

**Tabelle 2.1: Produkte der Produktklasse ALKIS-Standardausgaben**

Produktname	Beschreibung
Flurstücks- und Eigentumsnachweis (NRW)	„Der „Flurstücks- und Eigentumsnachweis“ enthält alle Angaben des „Flurstücksnachweises“ und darüber hinaus die „Eigentümer“ („Name“, „Adresse“,) und die „Angaben zu den Berechtigten.“ [AdV06, S. 269] Ein Beispielprodukt ist unter ?? auf Seite ?? zu finden.
Flurstücks- und Eigentumsnachweis (kommunal)	Erweiterung des Flurstücks- und Eigentumsnachweis (NRW). Beim Beispielprodukt ?? ist zu erkennen, dass mehr Informationen zu den Eigentümer angegeben sind.
Bestandsnachweis (NRW)	„Der „Bestandsnachweis“ enthält alle Grundstücke, die auf einem Buchungsblatt gebucht sind.“ [AdV06, S. 269] Ein Beispielprodukt ist unter ?? auf Seite ?? zu finden.
Bestandsnachweis (kommunal)	Der Bestandsnachweis (kommunal) enthält die gleichen Erweiterungen wie dies beim Flurstücks- und Eigentumsnachweis der Fall ist.
Grundstücksnachweis (NRW)	„Im „Grundstücksnachweis“ wird das unter einer laufenden Nummer im Buchungsblatt geführte Grundstück beschrieben. Neben den im Grundbuch gebuchten Grundstücken können dies auch von der Buchungspflicht befreite Grundstücke (§ 3 Abs. 2 GBO) und Grundstücke sein, die noch nicht im Grundbuch gebucht sind (Verzeichnis der Bodenordnung ersetzt den Grundbuchnachweis).“ [AdV06, S. 269]

**Tabelle 2.1: Produkte der Produktklasse ALKIS-Standardausgaben**

Produktname	Beschreibung
Flurkarte (NRW)	„Der „Auszug aus dem Liegenschaftskataster - Flurkarte NRW“ ist eine landesweit einheitliche Ausgabe aus dem Grunddatenbestand. Er enthält alle Flurstücke, Gebäude und Nutzungsarten ohne weitere topographische Inhalte.“ [WUP13a] In WuNDa wird die Flurkarte auch Liegenschaftskarte genannt, ein Beispieldokument kann unter ?? gefunden werden.
Stadtgrundkarte (kommunal)	Die Stadtgrundkarte (kommunal) ist eine umfangreiche Grundkarte, die, das Stadtbild prägende, Inhalte wiedergibt. Zusätzlich zur Flurkarte werden topographische Informationen wie zum Beispiel Fahrbahnbegrenzungen, Böschungen, Mauern, Bahngleise und Zäune, aber auch Balkone oder die Geschosszahl von Gebäuden. [vgl. WUP13a] Ein Beispieldokument kann unter ?? auf Seite ?? gefunden werden.
Stadtgrundkarte	Die Stadtgrundkarte ohne kommunale Ergänzungen. Diese Version wird nicht vertrieben, da die optischen Unterschiede, gegenüber der kommunalen Version, sehr gering sind. Aus diesem Grund wird nur eine Version angeboten.
Schätzungskarte (NRW)	Die Schätzungskarte basiert auf der Flurkarte und zeigt zusätzlich Schätzungsergebnisse einer Bodenschätzung [vgl. GG02]. Bei einer Bodenschätzung werden landwirtschaftlich genutzte Böden auf ihre Fruchtbarkeit hin geschätzt, so dass auf dieser Basis eine Besteuerung vorgenommen werden kann [vgl. Säc].

**Tabelle 2.1: Produkte der Produktklasse ALKIS-Standardausgaben**

Produktname	Beschreibung
Amtliche Basiskarte (NRW)	„Die Amtliche Basiskarte NRW (ABK NRW oder nur ABK) ist eine Übersichtskarte, die eine Verbindung zwischen der großmaßstäbigen Liegenschaftskarte und der Topographischen Karte 1:25000 (TK25) herstellt.“ [Wik13a]
Punktnachweis (PDF)	Beim Punktnachweis handelt es sich um ein Dokument das Informationen zu verschiedenen Vermessungspunkte enthält wie etwa Trigonometrische Punkte (TP), Nivellement Punkt (NivP) und Aufnahmepunkt (AP) [vgl. Inn05]. Ein Beispiel eines Punktnachweises kann ?? entnommen werden.
Punktliste (TEXT)	Die Punktliste enthält Informationen zu Vermessungspunkten, ähnlich wie der Punktnachweis, es handelt sich jedoch um eine einfache Textdatei.

### 2.2.2 Standardausgaben WUP-kommunal

Die ALKIS-Standardausgaben decken nicht alle Produkte ab, die benötigt werden. Aus diesem Grund musste die Stadtverwaltung Wuppertal weitere Produkte selbst definieren. Bei diesen Eigendefinition werden keine ALKIS-Objekte dargestellt und somit handelt es sich nicht um Auszüge aus dem Liegenschaftskataster. Der Inhalt der Produkte wird aus Rasterdatenbeständen bezogen und zeigen demnach Geodaten mit Bild. Obwohl es sich nicht um ALKIS-Standardausgaben handelt sind einige dieser Produkte vermessungsrelevant und somit in bestimmten Fällen für den ÖbVI gebührenfrei.

Die Produkte der Produktklasse Standardausgaben WUP-kommunal können der Tabelle 2.2 auf der nächsten Seite entnommen werden.

**Tabelle 2.2: Produkte der Produktklasse Standardausgaben WUP-kommunal**

Produktname	Beschreibung
Digitale Grundkarte (DGK)	Die DGK ist eine topografische Karte, die aus der Flurkarte und der Stadtgrundkarte abgeleitet wurde [vgl. WUPb]. Ein Beispiel einer DGK kann in ?? auf Seite ?? gefunden werden.
Digitale Grundkarte mit Höhenlinien	Eine DGK mit zusätzlichen Höhenlinien.
Orthofoto	Ein Orthofoto ist eine Mischung aus einem Luftbild und einer Karte [vgl. WUPa] Orthofotos „bieten kartenähnliche Genauigkeit und Maßstäbigkeit auf der Grundlage einer fotorealistischen Abbildung der Erdoberfläche.“ [WUPa]
Orthofoto mit Katasterdarstellung	Ein Orthofoto mit zusätzlicher Katasterdarstellung. Ein Beispiel eines solchen Produktes kann ?? entnommen werden.
NivP-Übersicht	Nivellement Punkte (NivP) werden auch Höhenfestpunkte genannt und dienen der Bestimmung von Höhen [vgl. Unn]. Eine NivP-Übersicht basiert auf einer topographischen Karte in der die einzelnen NivP eingetragen sind, so dass ihr Standort und ihre Nummerierung entnommen werden kann [vgl. AdV13c]. Ein Beispielprodukt kann ?? entnommen werden.

**Tabelle 2.2: Produkte der Produktklasse Standardausgaben WUP-kommunal**

Produktname	Beschreibung
AP-Übersicht	Die AP-Übersicht ist ähnlich der NivP-Übersicht wobei es sich bei den dargestellten Punkten um Aufnahmepunkte (AP) handelt. „Aufnahmepunkte sind“, laut Wikipedia [Wik13b], „Lagefestpunkte, die das TP-Netz unterster Ordnung (im Regelfall 4. Ordnung) verdichten und – gemeinsam mit diesen Trigonometrischen Punkten – einen koordinatenmäßigen Anschluss von Messungen an das übergeordnete Bezugssystem der Landesvermessung ermöglichen.“ Ein Beispiel einer AP-Übersicht kann unter ?? eingesehen werden.
Punktnummerierungsübersicht (PNÜ)	Eine topographische Karte, die die Punktnummern des Ausschnittes zeigt. Eine Beispielkarte ist ?? auf Seite ?? zu entnehmen.
Stadtgrundkarte mit Höhenlinien	Stadtgrundkarte mit zusätzlichen Höhenlinien.

### 2.2.3 WuNDa-Berichte

Die WuNDa-Berichte sind ebenfalls eigens definierte Produkte. Im Gegensatz zu den Standardausgaben WUP-kommunal enthalten sie allerdings keine Geodaten mit Bild, sondern sind Berichte die ein einzelnes Objekt beschreiben. Weiterhin sind sie ebenfalls nicht Bestandteil von ALKIS und auf Grund ihres Inhaltes vermessungsrelevant.

Die Produkte der Produktklasse Standardausgaben WuNDa-Berichte können der Tabelle 2.3 entnommen werden.



**Tabelle 2.3: Produkte der Produktklasse WuNDa-Berichte**

Produktname	Beschreibung
AP-Karte	AP-Karten werden angefertigt damit Aufnahme-punkte (AP) vor Ort aufgefunden werden können. In diese Karten werden Maße zu topographisch relevanten Gegenständen und die Entfernung zu Sicherheitspunkten angegeben [vgl. Sie]. Eine solche AP-Karte wird in ?? auf Seite ?? abgebildet.
Nivellement-Punkt	Auflistung von Informationen einzelner Nivellement Punkte (NivP), danach folgt für jeden NivP jeweils eine NivP-Beschreibung. Diese Beschreibung zeigt mittels einer Zeichnung den Standort des NivP. Eine solche NivP-Beschreibung kann in ?? gefunden werden.
Vermessungsriß	Bei einem Riß handelt es sich um eine technische Zeichnung, die nach den wichtigsten Linien eines Körpers angefertigt ist [vgl. Dud]. Ein Vermessungsriß stellt demnach die wichtigsten Informationen mit einer Zeichnung zu einer Vermessung bereit. Eine Art des Vermessungsrisse ist der Fortführungsriß, dieser wird bei der sogenannten Fortführung des Liegenschaftskatasters angelegt, also bei der Veränderung der eingetragenen Liegenschaften [vgl. Ver]. Im Fortführungsriß sind die Änderungen rot dargestellt und mit einem Pfeil markiert [vgl. SF99]. Ein solcher Fortführungsriß einer Gebäudeeinmessung kann in ?? gefunden werden.
Dokumente der Liegenschaftskatasterakten	Die Dokumente der Liegenschaftskatasterakten ergänzen die Vermessungsrisse. Zu ihnen gehören u. a. Koordinatenberechnungen und Grenzniederschriften. In letzteren erkennen die jeweiligen Eigentümer neu gezogene Grenzen an.

## 2.2.4 Daten Liegenschaftskataster

Die Produkte der Produktklasse Daten Liegenschaftskataster sind einzelne Dateien, die einen Teil des Liegenschaftskatasters abbilden. Die folgenden Dateiformate können ausgewählt werden:

**NAS** Die Normbasierte Austauschschnittstelle (NAS) ist das Datenaustauschformat von Alkis und umfasst neben den Fachobjekten auch Operationen zur Haltung von Bestandsdaten. NAS basiert auf Extensible Markup Language (XML), Geography Markup Language (GML) und Web Feature Service (WFS) [vgl. Geob].

**DXF** Das Drawing Interchange Format (DXF) ist ein Dateiformat das von dem Unternehmen Autodesk entwickelt wurde und in computer-aided design (CAD)-Applikationen (z. B. AutoCAD) als vektorbasierte Bilddatei Verwendung findet. DXF wurde mit dem Hintergrund entwickelt, dass es möglich sein soll AutoCAD Dokumente mit Programmen öffnen zu können, die nicht von Autodesk entwickelt wurden. Aus diesem Grund ist DXF eine ASCII-Textdatei [vgl. Fil13].

**GEOTIFF** „Ein GeoTIFF ist eine spezielle Form eines TIFF-Bildes, also ein Dateiformat zur Speicherung von Bilddaten [...]. Da das TIF-Format eine verlustfreie Speicherung zulässt, eignet es sich gut zur Verarbeitung von geographischen Daten, [...]“ [Wik13c] Weiterhin sind die „Bildinformationen [...] in jedem Programm darstellbar, welches den normalen TIFF-Standard unterstützt.“ [Wik13c] „Wesentliche Eigenschaft ist die enthaltene Georeferenzierung, die das Geotiff bei Verwendung im GIS an die richtige Stelle platziert, sodass es z.B. als Hintergrundbild für weitere Planzeichnungen dienen kann.“ [Frau Hähner] **TODO: Neu verfassen. Indirekte Zitierung.**

Die Produkte der Produktklasse Daten Liegenschaftskataster können der Tabelle 2.4 entnommen werden.

**Tabelle 2.4: Produkte der Produktklasse Daten Liegenschaftskataster**

Produktname	Beschreibung
-------------	--------------

**Tabelle 2.4: Produkte der Produktklasse Daten Liegenschaftskataster**

Produktname	Beschreibung
NAS-Daten (mit Eigentümern)	Vollständiger NAS-Datensatz, wird als XML-Datei ausgegeben.
NAS-Daten (ohne Eigentümer)	NAS-Datensatz ohne Eigentümer, wird als XML-Datei ausgegeben.
NAS-Daten (nur Punkte)	NAS Datensatz nur Punkte, wird als XML-Datei ausgegeben.
DXF-Daten (Stadtgrundkarte kommunal)	Die Stadtgrundkarte (kommunal) als DXF-Datei.
GEOTIFF-Daten (Stadtgrundkarte kommunal)	Die Stadtgrundkarte (kommunal) als GEOTIFF-Datei.

## 2.3 Beleuchtung typischer Workflows der externen Benutzer

Nachdem die einzelnen Produkte aufgelistet wurden, werden in diesem Abschnitt verschiedene Workflows von externen Benutzer beleuchtet und es wird gezeigt wie verschiedene Produkte verwendet werden. Im nächsten Abschnitt wird dann erläutert wie WuNDa diese Workflows im Augenblick unterstützt.

### 2.3.1 ÖbVI - Teilungsvermessung

Ein solcher Workflow ist die Teilungsvermessung, die durch einen ÖbVI durchgeführt wird. Durch die Teilungsvermessung wird ein neues Grundstück aus einem abgetrennten Teil eines bestehenden Grundstücks erstellt. Die folgende Beschreibung wurde nach [Kle] und [Jun13] erstellt.

## *2 Beleuchtung der Arbeitsweise von externen Benutzer*

Die Teilungsvermessung kann aus Sicht des ÖbVI in drei Teile zerlegt werden:

- Messungsvorbereitung
- Örtliche Vermessungsarbeiten
- Auswertung

Auf jeden dieser drei Schritte wird im Folgenden eingegangen.

### **Messungsvorbereitung**

Nachdem der ÖbVI die Auftragserteilung zur einer Teilungsvermessung durch den Eigentümer, Erwerber oder Bauträger erhalten hat, bezieht er die erforderlichen Katasterunterlagen vom Katasteramt und die Teilungsgenehmigung von der für das Baurecht zuständigen Baugenehmigungsbehörde.

Die benötigten Katasterunterlagen werden in Form der folgenden Produkte bezogen:

- Eigentümerangaben
- Flurkarte
- NAS-Bestandsdaten
- Fortführungsrisse
- AP-Karten
- AP-Übersichten
- Punktnummernübersichten

Sobald sämtliche Unterlagen vorhanden sind, kann die eigentliche Vorbereitung der Teilungsvermessung beginnen. In einem ersten Schritt müssen die zuvor erhaltenen NAS-Bestandsdaten in ein CAD-System sowie in ein Berechnungsprogramm (z. B. KAVDI) importiert werden.

Daraufhin können die Geräte, die im Außendienst benutzt werden, vorbereitet werden. Dazu gehört das Erstellen der Koordinatendateien mittels des Berechnungsprogramm, woraufhin diese vom Tachymeter und GPS-Gerät eingelesen

## *2 Beleuchtung der Arbeitsweise von externen Benutzer*

werden können. Danach kann die Übernahme der benötigten CAD-Dateien und des Berechnungsauftrags auf den Außendienstrechner erfolgen.

### **Örtliche Vermessungsarbeiten**

Die erste Aufgabe an Ort und Stelle ist das Untersuchen der alten Grenzen und das Aufsuchen der bereits vorhandenen Abmarkungen (Grenzsteine etc.). Danach kann der Lageanschluss durchgeführt werden. Beim Lageanschluss handelt es sich um einen „Anschluss einer Vermessung an das amtliche Bezugskoordinatensystem, [dieser ]wird durch differenzielle Beobachtung von Navigationssatelliten realisiert“ [Bie]. Diese Beobachtung erfolgt über SAPOS, dieses steht für Satellitenpositionierungsdienst der deutschen Landesvermessung und ermöglicht eine deutschlandweite Positionsbestimmung mittels Satelliten [vgl. SAP13, S. 2]. Die beim Katasteramt angeforderten AP-Karten und AP-Übersichten werden nur für Kontrollmessungen benötigt oder falls der Lageanschluss mittels SAPOS nicht möglich ist. Dies ist z. B. der Fall falls noch keine Koordinaten der Grenzpunkte vorliegen und somit die alten Vorgängervermessungen rekonstruiert werden müssen [vgl. Wola]. Danach kann der Aufmaß der neuen Vermessungspunkte erfolgen und die neue Grenze kann, wie gewünscht, durch neue Abmarkungen gekennzeichnet werden um den neuen Grenzverlauf festzulegen. Zu Schluss werden die Rohmessdaten zur weiteren Verarbeitung ausgegeben.

### **Auswertung**

Nach der örtlichen Arbeit folgt die Auswertung der ermittelten Rohmessdaten. Hierzu gehört es die Protokolle z. B. die Risse anzufertigen und die Berechnungen z. B. der neuen Flächen durchzuführen.

In einem ersten Schritt wird die sogenannte Punktnummernreservierung durchgeführt. Dabei werden für die neu zu erstellenden Punkte amtliche Punktnummern beim Katasteramt angefragt. Diese Nummern werden benötigt damit jeder Punkt eine eindeutige Kennung besitzt.

Der zweite Schritt der Auswertung besteht im Einlesen der Rohmessdaten in das Berechnungsprogramm **TODO: oder ins CAD? oder beides?**. Dieses kann den

## 2 Beleuchtung der Arbeitsweise von externen Benutzer

benötigten Fortführungsriß und die Skizze zur Grenzniederschrift automatisch erzeugen. Anschließend werde folgende Berechnungen zur Ermittlung der Koordinaten durchgeführt:

**Hierarchische Berechnung der Polaraufnahme** Mittels dieser Berechnung wird die Lage von Neupunkten bestimmt [vgl. Wik13d]. Bei Neupunkten handelt es sich um Punkte deren Koordinate noch nicht ermittelt wurden [vgl. Wik10].

**Flächenhafte Ausgleichung** Die flächenhafte Ausgleichung wird ebenfalls benutzt um neue Koordinaten zu ermitteln [vgl. Bez09b, S. 6].

Nachdem Fertigstellen der Berechnungen müssen die Daten aufbereitet werden ehe diese zurück an das Katasteramt gegeben werden können. Zu dieser Aufbereitung gehört die Vermessungspunktliste (VP-Liste), in der die zuvor ermittelten Koordinaten und zusätzlichen Informationen zu den Punkten eingetragen werden [vgl. Bez00, S. 62].

Um die Fortführung durchzuführen benötigt das Katasteramt diese drei erstellten Dokumente, also Fortführungsriß, Grenzniederschrift und VP-Liste. Zusätzlich können die Messergebnisse in digitaler Form eingereicht werden, welches dem Katasteramt die Fortführungsarbeit erleichtert. Dabei können die Punktdaten in Form einer Erhebungsdatei der Form NAS-ERH Stufe 1 zurückgegeben werden können. Bei NAS-Erhebungsdaten (NAS-ERH) Stufe 1 handelt es sich um einen Mindestumfang an Informationen der zu Vermessungspunkten bekannt sein muss [vgl. Bez13b].

Ehe die Erhebungsdatei erstellt werden kann muss darauf geachtet werden, dass jedes Katasteramt in NRW andere ALKIS-Datenbestände benutzen kann. Dies kann an der Attributart „abmarkung\_Marke“ zur Objektart AX\_Grenzpunkt verdeutlicht werden. Für diese Attributart gibt es im Grunddatenbestand NRW drei mögliche Werte:

**1000** Marke, allgemein

**9500** Ohne Marke

**9998** nach Quellenlage nicht zu spezifizieren

## *2 Beleuchtung der Arbeitsweise von externen Benutzer*

Das Maximalprofil NRW lässt allerdings weitere Werte zu, so z. B. „Stein, Grenzstein“ (1110) oder „Rohr mit Schutzkappe“ (1201) [vgl. Bez09a]. Kennt das Katasteramt die benutzten Werte nicht so müssen diese umgeschlüsselt werden, dies bedeutet dass im Zweifelsfall einer der drei Werte des Grunddatenbestandes NRW benutzt werden muss.

Nach Erstellung der Erhebungsdatei und der drei Dokumente können diese beim Katasteramt abgegeben werden. Dieses kann die Dokumente dann auf Richtigkeit überprüfen und archivieren. Mit Hilfe der Erhebungsdatei können die neu erstellten Punkte automatisch in ihr Softwaresystem importiert werden und weiterverarbeitet werden. Somit dienen sie als Basis für die Beamten des Katasteramtes, die die Teilungsvermessung dort übernehmen.

### **2.3.2 Polizei - Ermittlungen**

Die Polizei Wuppertal gehört auch zu den externen Benutzer von WuNDa. In verschiedenen Fahndungen müssen diese die Eigentümer von Flurstücken kennen, welche sie aus dem Produkt Flurstücks- und Eigentumsnachweis beziehen können. Ein anderes Beispiel ist die Erstellung von Unfallskizzen. Hier können Orthofotos Verwendung finden, da auf diesen die Fahrbahnmarkierungen sichtbar sind. Weiterhin können in solchen Fällen die Mess- und Zeichenwerkzeuge der WuNDa-Karte benutzt werden. Ein dritter Fall ist die Einsatzplanung, die auf Basis von geeigneter Datenquellen stattfindet. So können z. B. die Orthofotos auch hier wieder zu Einsatz kommen.

Ungeachtet dessen bereichert dieser Zugriff die Arbeit der Polizeibeamten, da die benötigten Informationen auch außerhalb der Öffnungszeiten des Katasteramtes bezogen werden können. Weiterhin besitzt jeder Beamte, aus Datenschutzgründen, seinen eigenen Benutzernamen. Auf diese Weise kann nachvollzogen werden, welcher Beamte welche Produkte bezogen hat.

## 2.4 Aktuelle Unterstützung der Workflows durch das System

In diesem Abschnitt wird beschrieben wie WuNDa die oben beschriebenen Workflows prinzipiell unterstützt. Um nicht auf sämtliche Details der einzelnen Workflows einzugehen, werden diese in drei Phasen eingeteilt:

- Beschaffung
- Verarbeitung
- Fortführung

Weiterhin werden vor allem die Schritte erläutert die eine Erstellung einer Abrechnung nach sich ziehen. Somit wird hier nicht beschrieben wie jedes einzelne Produkt in WuNDa erstellt werden kann.

### 2.4.1 Beschaffung

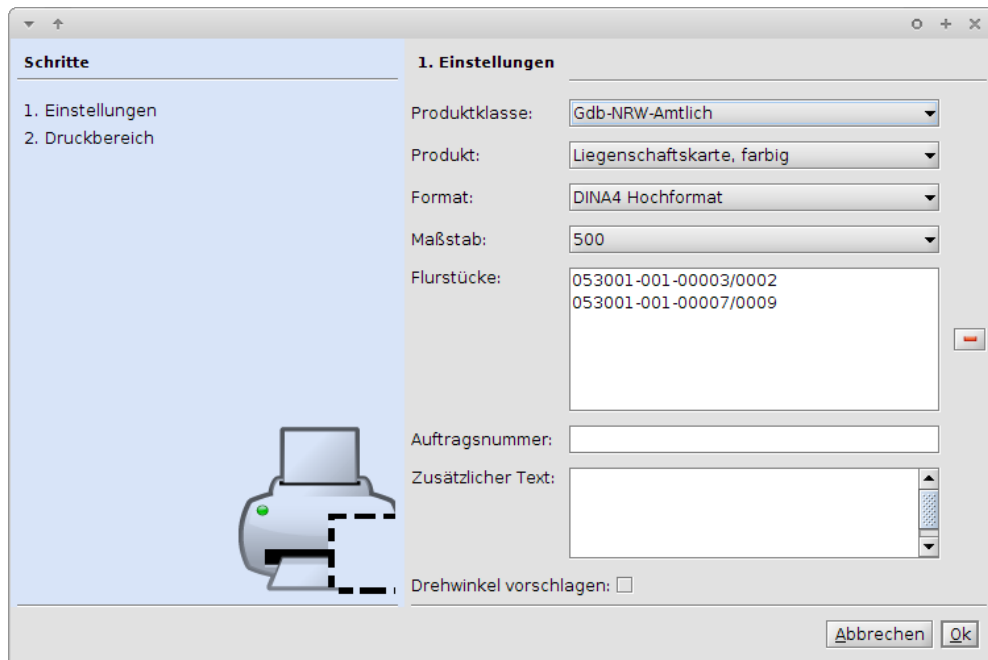
WuNDa unterstützt die externen Benutzer bei der Beschaffung der, für ihren Arbeitsschritt benötigten, Produkte. Durch die Selbstentnahme der Produkte wird das Anfragen der Produkte beim Katasteramt erspart, welches vorher die eigentliche Beschaffung übernehmen musste. Mittels der Selbstentnahme können die Produkte, die im Abschnitt 2.2 aufgelistet sind, bezogen werden. Die Benutzer können je nach Produkt verschiedene Einstellungen treffen. So können Sie etwa einen Bereich in der Karte oder Objekte auswählen, die in dem jeweiligen Produkt angezeigt werden.

Die Selbstentnahme wird im Folgenden exemplarisch vorgeführt, indem ein ÖbVI eine farbige Liegenschaftskarte bezieht. Ein Beispielprodukt der Liegenschaftskarte kann unter ?? gefunden werden. Hier sei daran erinnert, dass es sich bei der Liegenschaftskarte um das Produkt der Flurkarte handelt.

Zum Beziehen der Liegenschaftskarte muss in WuNDa der sogenannten ALKIS-Druckdialog (siehe Abbildung 2.2 auf der nächsten Seite) geöffnet werden. Nach dem Öffnen des Dialogs kann der Benutzer die Produktklasse auswählen. In



## 2 Beleuchtung der Arbeitsweise von externen Benutzer



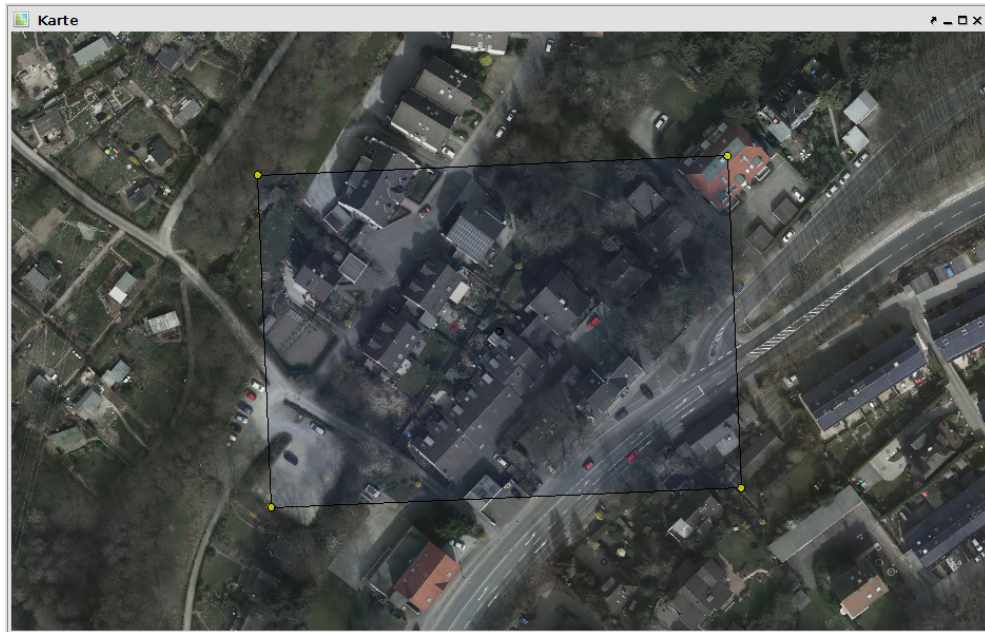
**Abbildung 2.2: ALKIS-Druckdialog**

diesem Fall ist dies „Gdb-NRW-Amtlich“, welches eine Teilmenge der ALKIS-Standardausgaben darstellt. Je nachdem welche Produktklasse ausgewählt wurde, ändern sich die Auswahlmöglichkeiten der Produkte. Wie bereits erwähnt, wird hier die farbige Liegenschaftskarte ausgewählt. Eine weitere wichtige Eingabe ist das Flurstück-Feld, hier können ein oder mehrere Flurstücke per drag-and-drop hereingezogen werden, diese werden dann später im Produkt dargestellt. Weitere Eingabefelder sind das Format (z. B. DIN A4), den Maßstab (hier 1:500), die Auftragsnummer und ein Kommentarfeld. Die Option „Drehwinkel vorschlagen“ beeinflusst die Rotation des später angezeigten Druckbereiches.

Bestätigt der Benutzer mit „Ok“ seine Eingabe so gelangt er zur Karte in WuNda, wobei der Druckbereich angezeigt wird (siehe Abbildung 2.3 auf der nächsten Seite). Dieser Druckbereich befindet sich über den zuvor ausgewählten Flurstücken und kann verschoben, in der Größe verändert und rotiert werden. Mit einem Doppelklick auf den Druckbereich, wird gelangt er zum Downloadprotokoll-Dialog (siehe Abbildung 2.4 auf Seite 27).

Das Downloadprotokoll-Dialog hat mehrere Aufgaben, so

## 2 Beleuchtung der Arbeitsweise von externen Benutzer



**Abbildung 2.3: Auswahl des Druckbereiches**

- nimmt er weitere Eingaben entgegen,
- er zeigt die fällige Gebühr des Downloads an,
- und der Benutzer kann die Gebühr bestätigen.

Die Eingaben bestehen aus dem Pflichtfeld der Geschäftsbuchnummer, dem optionalen Feld der Projektbezeichnung und der Auswahl des Verwendungszwecks. Für die Abrechnungen ist der Verwendungszweck besonders relevant, da dieser verschiedene Rabatte nach sich ziehen kann. Mehr Informationen zum Zustandekommen der Rabatte kann dem Abschnitt 2.1.4 entnommen werden.

Im Berechnungsfeld kann der Benutzer nachvollziehen wie die Gebühr zu seinem Download zustande kommt. Hier wird das gewählte Produkt und dessen Grundpreis angegeben, woraufhin der Rabatt abgezogen wird, um schließlich die fällige Gebühr anzuzeigen.

Die auswählbare Verwendungszwecke eines ÖbVI sind u. a. „Vermessungsunterlagen (amtlicher Lageplan TS3)“ und „Vermessungsunterlagen (hoheitliche Vermessung TS4)“. In diesem Beispiel ist der Benutzer ein ÖbVI der ein vermessungsrelevantes Dokument bezieht, somit fällt, nach Abzug der Rabatte, eine Gebühr

## 2 Beleuchtung der Arbeitsweise von externen Benutzer

Downloadprotokoll (oebvi...@ÖbVI - gesetzlich@WUNDA\_BLAU)

Geschäftsbuchnummer: 4711

Projektbezeichnung:

Verwendungszweck: Vermessungsunterlagen (amtlicher Lageplan TS 3)

Berechnung:

Produkt: Flurkarte (NRW) DIN A4	
1 Karte max. DIN A3 (a 20,00 €)	
	20,00 €
zweckabhängiger Rabatt (100%) :	-20,00 €
	0,00 €

Gebühr: 0,00 €

Abbrechen Ok

**Abbildung 2.4: Downloadprotokoll-Dialog**

von 0 € an. Bezieht der Kunde ein kostenpflichtiges Produkt selbst, so fällt auf jeden Fall der sogenannte Selbstentnahmerabatt an.

Mit dem Bestätigen des Dialoges akzeptiert der Benutzer die Gebühr und der Download des Produktes wird gestartet. Nach dem Abschluss des Downloads kann der Benutzer die heruntergeladene Datei wie gewohnt benutzen. Im Falle der Liegenschaftskarte wird eine PDF-Datei heruntergeladen, ein Beispiel hiervon ist auf Seite ?? zu finden.

Zeitgleich mit dem Starten des Downloads wird dieser protokolliert. Dies bedeutet, dass relevante Informationen zu dem Download abgespeichert werden. Das Abspeichern erfolgt in jedem Fall, also auch wenn die Gebühr 0 € beträgt oder der Download fehlschlägt. Insbesondere der letzte Fall bereitet beim Erstellen der Abrechnung Probleme, wie dies dem Abschnitt 2.5.1 auf Seite 30 zu entnehmen ist, da diese Protokolle storniert werden müssen.

Kurzum der externe Benutzer erhält durch die Selbstentnahme verschiedene Vor-

## *2 Beleuchtung der Arbeitsweise von externen Benutzer*

teile. Einerseits stehen ihm die Dokumente sofort zur Verfügung und dies auch außerhalb der Öffnungszeiten des Katasteramtes. Andererseits erhält er durch die Selbstentnahme ein Rabatt, da das Katasteramt die, ansonsten benötigte, Bearbeitungszeit einspart.

### **2.4.2 Verarbeitung**

Bei der Verarbeitung der Produkte oder sonstigen weiteren Tätigkeiten unterstützt WuNDa den externen Benutzer nicht.

### **2.4.3 Fortführung**

Bei der Fortführung wird der externen Benutzer mittels der von WuNDa angebotenen Punktnummernreservierung unterstützt. Ansonsten wird dem Benutzer in diesem Bereich keine Funktionalität angeboten.

Welche Erweiterungen von WuNDa denkbar und sinnvoll sind, wird im Kapitel **TODO: Ref** ausführlich beschrieben. Zu Schluss dieses Kapitels wird auf die Problemstellung aus Abschnitt 1.2 eingegangen, demnach wird detaillierter beschrieben warum die geplante Rationalisierung nicht durchgeführt werden kann.

## **2.5 Hohe Bearbeitungszeit**

Bislang wurde angegeben, dass die geplante Rationalisierung auf Grund der hohen Bearbeitungszeit die durch die Selbstentnahme entsteht nicht durchgeführt werden kann. Um dies zu verdeutlichen wird im nächsten Abschnitt die Erstellung der Abrechnung beschrieben.

### 2.5.1 Manuelles Erstellen der Abrechnung

Die Abrechnungen werden von einem Beamten des Katasteramts nach jedem Abrechnungsturnus (z. B. quartalsweise, jährlich) manuell erstellt, wobei jeder externe Benutzer zu einen anderen Abrechnungsturnus gehören kann. Weiterhin muss für jeden Benutzer eine eigene Abrechnung erstellt werden.

Im folgenden werden die Download-Protokolle auch Buchungen genannt, da sich die Sichtweise auf die Protokolle von der Sicht der externen Benutzer auf die Sicht des Rechnungswesens geändert hat. Zum Erstellen der einzelnen Abrechnungen müssen die Buchungen, also die Protokolle der Downloads, aus der Datenbank bezogen werden um weiterverarbeitet zu werden. Dies geschieht indem die Buchungen sämtlicher Kunden nach dem gewünschten Zeitraum gefiltert werden, um dann in eine Excel-Datei importiert zu werden. Demnach befindet sich in einer Spalte dieser Datei Informationen wie z. B. der Benutzername, der Download-Name und die zu bezahlende Gebühr.

Im folgenden Arbeitsschritt muss der Beamte die einzelnen Buchungen für jeden Benutzer von dieser Excel-Datei in eine jeweils eigene Excel-Datei kopieren. Eine solche Excel-Datei repräsentiert die Abrechnung an die einzelnen Kunden des Katasteramtes.

Diese Vorgehensweise hat einige Nachteile. Ein Nachteil ist, dass durch den manuellen Kopiervorgang eine sehr lange Bearbeitungszeit anfällt. Diese wird in Zukunft noch verlängert, da ein Ausbau der externen Benutzer geplant ist, woraufhin die Größe der Anfangsdatei anwächst.

Ein weiterer Nachteil ist, dass die Protokolle nur auf Login-Ebene angelegt werden, d. h. es wird nur der Name gespeichert, mit dem sich der externe Benutzer in WuNDa anmeldet. Dies ist unproblematisch falls es pro Kunde nur einen Login-Namen gibt. Dies ist allerdings nicht immer der Fall, weil verschiedene Kunden mehrere Login-Namen besitzen. Aus diesem Grund müssen beim Erstellen der Abrechnung die Login-Namen aus den Protokollen den entsprechenden Kunden zugeordnet werden. Demnach muss der Beamte, beim Erstellen der Abrechnung an einen bestimmten Kunden, darauf achten diese Zuordnung richtig und vollständig zu machen. Dies ist zwar machbar, verkompliziert den Arbeitsschritt aber

## *2 Beleuchtung der Arbeitsweise von externen Benutzer*

ungemein.

Ein weiterer Nachteil ist, dass keine Bearbeitungen der Buchungen selbst vorgenommen werden kann. Somit werden bereits abgerechnete Buchungen nicht als abgerechnet markiert. Weiterhin können keine Änderungen z. B. am Auftragsnamen der Buchung vorgenommen werden und die Buchung kann nicht als storniert markiert werden. Solche Bearbeitungen müssen vom Beamten selbst notiert und beim Erstellen der Abrechnungen berücksichtigt werden.

Kurzum die Vorgehensweise ist durch die manuelle Bearbeitung und den aufgeführten Nachteilen sehr aufwändig und fehleranfällig. Dieses Problem wird durch die Vergrößerung der Anzahl der externen Benutzer noch verschärft.

Aus diesen Gründen soll WuNDa einige Erweiterungen erhalten, die diese Probleme beseitigen, in dem es diesen Vorgang möglichst automatisiert und auch Funktionalitäten mit einer verwandten Thematik anbietet. Diese Erweiterungen sollen den externen Benutzern, aber vor allem dem Katasteramt Zeitersparnisse einbringen und werden im nächsten Kapitel spezifiziert.

## 3 Spezifikation

In diesem Kapitel werden die gewünschten Erweiterungen von WuNDa spezifiziert. Diese Erweiterungen werden der Übersicht halber zuerst aufgelistet und danach detaillierter beschrieben.

Die Informationen zu diesem Kapitel sind aus einer Zusammenarbeit zwischen dem Ressort 102 und Cismet entstanden. Dabei entstanden die Anforderungen nicht auf Anhieb, sondern es wurde von den Grundideen ausgegangen, so dass ein erstes Datenmodell und erste Mock-Ups entstanden. Basierend auf diesen Dokumenten wurden die Ideen, in mehreren Schritten, verfestigt und die konkreten Anforderungen an WuNDa entstanden. Dieses Kapitel fasst diese Anforderungen in ihrer schlussendlichen Form zusammen.

### 3.1 Gewünschte Erweiterungen

Dieser Abschnitt soll als Übersicht der gewünschten Erweiterungen dienen. Dabei werden zunächst einige Begriffe kurz erklärt, die wichtigsten Klassen erwähnt und anschließend werden die gewünschten Erweiterungen aufgelistet.

Bei der Spezifikation werden möglichst nur allgemein bekannte technische Begriffe verwendet, die wenigen spezifischeren Ausnahmen werden hier erklärt. **TODO: ref zum technischen Umfeld?; Gehört das hierhin? Abschnitt mit dem Cids-Umfeld vor die Spezifikation ziehen? Dies bedeutet allerdings einen Bruch der Sicht. Kapitel 2,3 fachliche Sicht, Kapitel 4 technische Sicht.**

**Klasse** Mit Klasse ist eine Cids-Klasse gemeint. Objekte von diesen Klassen werden in einer Datenbank gespeichert. Aus diesen Datenbankeinträgen lassen sich Java-Objekte erstellen, die auch CidsBean genannt werden. Diese

### 3 Spezifikation

Objekte können in einem Java-Programm benutzt und verändert werden. Anschließend können sie wieder in die Datenbank zurückgeführt werden.

**Renderer** Eine solche Klasse kann einen Renderer besitzen. Bei einem Renderer handelt es sich um eine Beschreibungsseite in einem Java-Programm, die Informationen bezüglich eines Objektes dieser Klasse anzeigt.

**Editor** Eine Cids-Klasse kann außerdem einen Editor besitzen. Ein Editor ist von dem Anzeigemechanismus ähnlich dem Renderer, im Gegensatz hierzu können die Informationen zu einem Objekt jedoch verändert und in der Datenbank gespeichert werden. Mit einem Editor ist es in der Regel auch möglich neue Objekte einer Cids-Klasse anzulegen.

Um die Spezifikation umzusetzen muss mit solchen Cids-Klassen gearbeitet werden, dabei sind die drei neu eingeführten Klassen Kunden, Kunden-Logins und Kundengruppe sowie die bereits bestehende Klasse Buchungen am relevantesten für das automatische Erstellen der Abrechnungen und werden im Abschnitt 3.2 stärker beleuchtet. Die restlichen neuen Klassen werden gesondert im Abschnitt 3.2.7 aufgeführt und beschrieben.

Damit im nächsten Abschnitt die Zusammenhänge der Erweiterungen aufgeführt werden können, ohne dass sämtliche Erweiterungen bereits erläutert wurden, werden diese hier aufgelistet. Die Liste ist in Kategorien eingeteilt, wobei eine Kategorie die logische Zuordnung der Erweiterung darstellt. So finden sich unter den Kategorien die neuen Klassen Kunde und Kundengruppe, aber auch die bereits bestehende Klasse Buchung.

- Allgemeines
  - Berechtigungen
  - Hinzufügen des Mehrwertsteuer-Satzes bei Gebühren
- Kunde
  - Kundenrenderer und Editor
  - Erneutes Herunterladen von Produkten
  - Ansicht der eigenen Buchungen durch externe Benutzer



### 3 Spezifikation

- Kundengruppen
  - Kundengruppenrenderer und Editor
  - Erstellen von E-Mails an Kundengruppen
- Buchungen
  - Buchungrenderer und Editor
  - Stornieren und Ändern einzelner Buchungen
  - Suchdialog zu den Buchungen
- Berichte
  - Buchungsbeleg
  - Rechnungsanlage
  - Geschäftsstatistik
- Renderer und Editoren für sonstige neue Klassen

## 3.2 Spezifikation der Erweiterungen

In diesem Abschnitt werden die oben genannten Erweiterungen spezifiziert. Dabei werden die Erweiterungen in der Regel zuerst erläutert und danach mittels möglichst wenigen und einfachen Sätzen spezifiziert.

### 3.2.1 Berechtigungen

Durch die Erweiterungen können vertrauliche Informationen betrachtet und verändert werden. Aus diesem Grund dürfen nur einige ausgewählte Benutzer Zugriff auf diese Erweiterungen erhalten. Bei den Benutzern handelt es sich um die WuNDa-Administratoren und einigen Beamten die für die Kundenverwaltung zuständig sind. Die einzige Ausnahme hiervon bildet die Ansicht der Kunden auf ihren eigenen Kundenrenderer. Siehe hierzu auch den Unterabschnitt „Ansicht der eigenen Buchungen durch externe Benutzer“ im Abschnitt 3.2.3.

### 3 Spezifikation

Demnach sollen nur die WuNDa-Administratoren und einige Beamten der Kundenverwaltung Zugriff auf die gewünschte Erweiterungen erhalten. Wie bereits erwähnt, soll es hiervon nur eine einzige Ausnahme geben.

#### 3.2.2 Hinzufügen des Mehrwertsteuer-Satzes bei Gebühren

Bei der bisherigen Implementierung wurde bei den Buchungen jeweils nur der Brutto-Preis abgespeichert. Dies ist nach der Umstellung nicht mehr ausreichend, da die Abrechnung mittels WuNDa erstellt werden soll. Für diese Erstellung ist der Netto-Preis und der Mehrwertsteuer-Satz eines Produktes von Nöten. Weiterhin soll aus Gründen der Nachvollziehbarkeit ebenfalls der Brutto-Preis mit gespeichert werden.

Demnach soll für eine Buchung der Netto-Preis, der Mehrwertsteuer-Satz sowie der Brutto-Preis gespeichert werden. Diese neuen Daten sollen ebenfalls im Downloadprotokoll-Dialog angezeigt werden.

#### 3.2.3 Kunde

Im aktuellen Stand werden die Kunden des Ressort 102 nicht in WuNDa verwaltet, durch die gewünschten Erweiterungen wird dies jedoch zwingend notwendig. Daher wird eine Klasse benötigt die es ermöglicht Kunden zu verwalten und darzustellen.

Für die Klasse Kunde sollen folgende Felder bereitstehen:

**Name** Der Name des Kunden, dies kann z.B. eine Organisationsbezeichnung sein.

**Vermessungsstellennummer** Dieses Feld enthält die Zulassungsnummer einer Vermessungsstelle. Eine Vermessungsstellen ist eine Behörde oder Person, die berechtigt ist Liegenschaftsvermessungen durchzuführen und Abmarkungen vorzunehmen. Zu den Vermessungsstellen zählen z.B. das Katasteramt oder ÖbVI [vgl. Kre].

### 3 Spezifikation

**Vertragskennzeichen** Das Kennzeichen des Vertrages der dem Kunde die Benutzung von WuNDa erlaubt.

**Vertragsende** Das Auslaufdatum dieses Vertrages.

**Abrechnungsturnus** Der zeitliche Abstand wann ein Kunde seine Abrechnung erhält. (z. B.. quartalsweise, jährlich)

**Direktkontakt** Eine E-Mail-Adresse mit der, der Kunde sofort erreicht werden kann. Dies kann zum Beispiel die Adresse des Geschäftsleiters sein.

**Weiterverkaufsvertragskennzeichen** Das Kennzeichen des Vertrages der dem Kunden das Weiterverkaufen von bezogenen Produkten erlaubt.

**Weiterverkaufsvertragsende** Das Auslaufdatum des Weiterverkaufsvertrags.

**Branche** Die Branche in der der Kunde tätig ist. Ein Kunde kann in genau einer Branche tätig sein. Hierbei handelt es sich um eine neue Klasse.

**Produkte** Die Softwareprodukte die ein Kunde benutzt. Ein Kunde kann mehrere Produkte besitzen. Hierbei handelt es sich ebenfalls um eine neue Klasse.

Neben den Kunden sollen die externen Benutzer in WuNDa abgebildet werden, hierfür wird die Klasse Kunden-Login eingeführt. Somit repräsentiert ein Kunden-Login einen WuNDa-Benutzer. Weiterhin soll ein Kunden-Login genau einem Kunden zugeordnet sein.

Zusammenfassend benötigt ein Kunden-Login folgende Felder:

**Name** Der Name des Kunden-Logins. Dies ist ebenfalls der Name mit dem sich der externe Benutzer in WuNDa anmeldet.

**Kontakt** Die E-Mail-Adresse des externen Benutzers.

**Kunde** Die Zuordnung des Kunden-Logins zu genau einem Kunden.

### 3 Spezifikation

#### **Kundenrenderer**

Der Kundenrenderer soll Buchungen eines Kunden tabellarisch darstellen. Dabei sollen verschiedene Filtermöglichkeiten angegeben werden können, so dass nur Buchungen angezeigt werden, die die Kriterien erfüllen. Die Filterergebnisse sollen in einem kurzen Satz zusammengefasst werden. Für die angezeigten Buchungen soll es möglich sein den Bericht Rechnungsanlage und Buchungsbeleg zu erstellen.

Die Buchungen sollen zeitlich gefiltert werden können. Dabei soll es eine Schnellauswahl für die Buchungen des aktuellen Tages, eines beliebigen Monats und eines beliebigen Quartals geben. Weiterhin soll ein Datumsbereich als Filter benutzt werden können. Die Buchungen sollen auch nach ihrer Geschäftsbuchnummer, ihrem Projekt und dem Kunden-Login, der die Buchung angelegt hat, gefiltert werden können. Weitere Filter sind der Kostentyp, sprich ob eine Buchung kostenfrei oder kostenpflichtig ist, sowie der Verwendungszweck der Buchung. Zusätzlich besteht das Kriterium dass sämtliche Filter eine neutrale Einstellung haben müssen, bei der sie keine filternde Funktionalität übernehmen.

In der Tabelle müssen die Geschäftsbuchnummer, die Projektbezeichnung, der Verwendungszweck, das Art des bezogenen Produktes, die Netto-Gebühr, der MwSt.-Satz, das Erstellungsdatum sowie der Kunden-Login der Buchung angezeigt werden.

Die einzelnen Filtermöglichkeiten und der Aufbau der Tabelle kann dem Mock-up des Kundenrenderers 3.1 auf der nächsten Seite entnommen werden.

#### **Ansicht der eigenen Buchungen durch externe Benutzer**

Die Funktionalität des Kundenrenderers soll dem Kunden selbst nicht vorenthalten werden. Aus diesem Grund soll es für die externen Benutzer eines Kunden möglich sein auf den eigenen Kundenrenderer zuzugreifen. Hierbei muss sichergestellt werden, dass die externen Benutzer nur den Kundenrenderer des eigenen Kunden öffnen können und somit keine Einsicht in die Buchungen der anderen Kunden erhalten. Weiterhin dürfen sie den Bericht Rechnungsanlage nicht erstellen können, da es sich bei diesem um einen Teil der Abrechnung handelt.

### 3 Spezifikation

[illegible]

### Abbildung 3.1: Mockup des Kundenrendereres

Demnach sollen externe Benutzer nur den eigenen Kundenrenderer öffnen können und es soll ihnen nicht möglich sein den Bericht Rechnungsanlage zu erstellen.

## Erneutes Herunterladen von Produkten

Die Produkte, die über den ALKIS-Druckdialog erzeugt werden, werden von einem externen Server bezogen. Durch dessen längere Initialisierungsphase passiert es häufiger, dass Produkte fehlerhaft heruntergeladen werden. Da der Download dieser Produkte trotzdem protokolliert wird, wie im Abschnitt 2.4.1 auf Seite 27 bereits erwähnt wurde, muss der externe Benutzer diese Buchung stornieren lassen und den Download erneut durchführen. Zwar gehört eine vereinfachte Stornierung auch zu den gewünschten Erweiterungen, trotzdem soll es für den Benutzer

### 3 Spezifikation

möglich sein ein Download nochmals anzufordern, ohne dass er eine Buchung stornieren lassen muss. Weiterhin soll durch diesen Vorgang keine neue Buchung entstehen. Aus diesem Grund soll es für den Benutzer möglich sein, Produkte die bereits heruntergeladen wurden, erneut herunterzuladen. Einschränkend soll dies nur für Produkte möglich sein, die am gleichen Tag und über den ALKIS-Druckdialog bezogen wurden.

Dieses erneute Herunterladen soll wie folgt umgesetzt werden. Bei einer Buchung, die am gleichen Tag erfolgte, soll in der Tabelle im Kundenrenderer das Datumsfeld eine zusätzliche Funktion als Auslöser zum erneuten Herunterladen des Produkts erhalten. Wenn das Datumsfeld ebenfalls als Auslöser dient, dann soll es blau unterstrichen sein, wie dies der Abbildung 3.1 auf der vorherigen Seite zu entnehmen ist. Bei einem Klick auf ein solches Datumsfeld soll das bereits protokollierte Produkt erneut heruntergeladen werden, ohne dass eine neue Buchung erstellt wird.

#### **Kundeneditor**

Mit Hilfe des Kundeneditors soll es möglich sein einen neuen Kunden anzulegen, bzw. Informationen zu einem bereits existierenden Kunden zu ändern.

#### **3.2.4 Kundengruppen**

Bei den Kundengruppen handelt es sich um Mengen von Kunden. Diese Klasse wird benötigt um Kunden mit ähnlichen Eigenschaften zu gruppieren. Die Kundengruppen besitzen kein Wissen über ihren Inhalt, sondern müssen manuell erstellt und verwaltet werden.

Die Funktionsweise einer Kundengruppe soll an dem Beispiel Gruppe „Quartalsweise“ erläutert werden. Diese Gruppe enthält sämtliche Kunden die den Abrechnungsturnus „Quartal“ besitzen. Wird ein neuer Kunde erstellt, mit eben dem Abrechnungsturnus, so muss dieser manuell der Kundengruppe hinzugefügt werden. Diese Kundengruppe hat den Nutzen, dass beim Erstellen der Abrechnung für ein Quartal die benötigten Kunden nicht zusammengesucht werden müssen, sondern bereits in dieser Gruppe gesammelt sind.

### 3 Spezifikation

Kurzum eine Kundengruppe soll eine Menge von Kunden sein, die manuell erstellt und verwaltet werden muss. Eine Kundengruppe soll folgende Informationen enthalten:

- Name
- Beschreibung
- Liste der Kunden

#### **Kundengruppenrenderer**

Der Kundengruppenrenderer soll die Buchungen eines jeden Kunden einer Kundengruppe aggregieren und tabellarisch darstellen. Seine Funktionsweise ist ähnlich dem des Kundenrenderers und besitzt deshalb auch einen ähnlichen Aufbau. Demnach können wieder verschiedene Filtermöglichkeiten angegeben werden und die Berichte Rechnungsanlage, Buchungsbeleg können erstellt werden. Zusätzlich steht hier auch der Bericht Geschäftsstatistik zur Auswahl.

Da im Vergleich zu dem Kundenrenderer nicht ein einzelner Kunde dargestellt wird, sondern mehrere, werden hier nicht sämtliche Buchungen angezeigt. Deshalb werden die Buchungen für die einzelnen Kunden zusammengefasst und somit in aggregierter Form in der Tabelle dargestellt. Die in der Tabelle angezeigten Informationen sind der Kundename, der aggregierte Brutto-Preis und die Anzahl der kostenpflichtigen Downloads. Weiterhin soll jeder Eintrag in der Tabelle ein Auswahlfeld besitzen. Mit diesem Feld kann der Benutzer auswählen, ob die Buchungen eines Kunden beim Erstellen eines Berichtes benutzt werden sollen, oder nicht.

Die Buchungen können genau wie im Kundenrenderer zeitlich und nach Verwendungszweck gefiltert werden. Weiterhin kann nach dem Abrechnungsturnus des Kunden gefiltert werden und kostenfreie Buchungen können ausgeblendet werden.

Eine frühes Mock-up des Kundengruppenrenderer kann in Abbildung 3.2 auf der nächsten Seite betrachtet werden.

### 3 Spezifikation

Aggregationsrenderer "Kunde"

Filtermöglichkeiten

Heute

Monatlich

Quartal

Datumsbereich

von: 23.08.2013

bis: 17.10.2013

Aggregierte Buchungen zwischen dem 23.08.2013 und dem 17.10.2013

Auswahl für Berichte	Kundenname	aggregierter Preis	Anzahl Buchungen
<input checked="" type="checkbox"/>	Testkunde 1	30 €	5
<input type="checkbox"/>	Testkunde 2	45 €	10

Rechnungsanlage

Geschäftstatistik

Abbildung 3.2: Mockup des Kundengruppenrenderers

#### Kundengruppeneditor

Mittels des Kundengruppeneditor sollen neue Kundengruppen angelegt werden können und bereits bestehende Kundengruppen sollen bearbeitet werden können. Zu diesen Bearbeitungen gehört z. B. das Hinzufügen bzw. Entfernen eines Kunden aus der Kundengruppe.

#### Erstellen von E-Mails an Kundengruppen

In verschiedenen Fällen müssen Benachrichtigung-E-Mails an alle oder verschiedene Kunden versendet werden. Dies muss u. a. bei geplanten Systemausfällen, die durch Wartungsarbeiten verursacht werden, beim Einführen von neuen Funktionen oder sonstigen neuen Funktionalitäten wie z. B. neue Kartenthemen erfolgen. Im aktuellen Stand besitzt jeder Beamte, der solche E-Mails versenden soll,



### 3 Spezifikation

einen eigenen Verteiler, den er manuell aktuell halten muss. Nach dem Hinzufügen/Entfernen eines Kunden oder dem Ändern einer E-Mail-Adresse kann es somit zu Unvollständigkeits in den einzelnen Verteiler kommen, da diese noch nicht aktualisiert wurden. Mittels der Kundengruppen können diese, von den Beamten einzeln verwalteten, Verteiler durch eine zentrale Lösung ersetzt werden, die für jeden Beamten den gleichen Stand besitzt.

Der gewünschte Lösungsansatz ist die Möglichkeit für eine Kundengruppe auszuwählen ob eine E-Mail an die jeweiligen Direktkontakte oder an jeden Kunden-Login der Kunden innerhalb der Gruppe erstellt werden soll. Nach dieser Auswahl soll eine E-Mail Vorlage gewählt werden können. Diese Vorlage soll den Betreff und den Text der E-Mail enthalten. Weiterhin soll sie zusätzlich die To-Adressaten enthalten, dies sind in der Regel Adressen von Mitarbeiter aus dem Katasteramt. Diese Mitarbeiter sollen die später versendete E-Mail als Kontrolle erhalten. Nach der Auswahl der Vorlage soll ein E-Mail-Erstellungsfenster des Standard-E-Mail-Programms des Benutzers geöffnet werden. In diesem Erstellungsfenster sollen die Daten der Vorlage eingesetzt werden. Weiterhin sollen die ausgewählten E-Mail-Adressen (sprich die Direktkontakte bzw. die Adressen aller Kunden-Logins) als BCC-Adressen hinzugefügt werden.

Daraufhin kann der Benutzer die E-Mail wie gewohnt bearbeiten und z. B. den bereits eingesetzten Text beliebig verändern und schlussendlich die E-Mail an sämtlich eingetragene Kontakte versenden.

#### **3.2.5 Buchungen**

Bei den Buchungen handelt es sich um die bereits mehrmals erwähnten Download-Protokolle. Im aktuellen Stand können die Buchungen nicht in WuNDA dargestellt oder bearbeitet werden.

##### **Suchdialog zu den Buchungen**

Dieser Suchdialog soll es ermöglichen Buchungen über verschiedene Suchkriterien zu finden und anschließend im Buchungsrenderer oder -editor zu öffnen. Diesen Renderer und Editor soll nur über den Umweg über den Suchdialog geöffnet

### 3 Spezifikation

werden können. Die Suchkriterien, die angegeben werden können, sind der Kundenname, der Abrechnungsturnus des Kunden, die Geschäftsbuchnummer und das Projekt der Buchung sowie das Kunden-Login, mit dem die Buchung angelegt wurde. Weiterhin soll, es wie im Kundenrender, möglich sein zeitliche Angaben, den Verwendungszweck und den Kostentyp anzugeben. Zusätzlich kann angegeben werden ob stornierte oder abgerechnete Buchungen gefunden werden sollen.

Der prinzipielle Aufbau des Suchdialogs kann dem Mock-up 3.3 entnommen werden.

#### **Buchungseditor und Buchungsrenderer**

Beim Buchungseditor und dem Buchungsrenderer soll es sich um die gleiche Ansicht handeln, wobei der Renderer keine Benutzereingabe entgegen nimmt. Der Buchungseditor soll es nicht erlauben neue Buchungen zu erstellen, da diese nur über den Downloadprotokoll-Dialog nach dem Beziehen eines Produktes angelegt werden. Weiterhin soll es im Buchungseditor nur möglich sein die Geschäftsbuchnummer und die Projektbezeichnung zu ändern. Außerdem soll eine Buchung storniert werden können. Siehe hierzu auch die zwei folgenden Abschnitte.

#### **Änderung von einzelnen Buchungen**

Wie bereits erwähnt dürfen nicht sämtliche Werte einer Buchung im Buchungseditor verändert werden sondern es sollen nur die Werte Geschäftsbuchnummer und Projektbezeichnung verändert werden können. Diese beiden Felder werden vom externen Benutzer händisch eingegeben und deshalb sind Fehleingaben vorstellbar, die von Beamten des Katasteramtes verbessert werden müssen.

Weiterhin soll die jeweils letzte Änderung protokolliert werden. Auf eine Historie mit den gemachten Änderungen wird bewusst verzichtet, erfolgen also zwei Änderungen so wird das Protokoll der erste Änderung überschrieben. In dieser Protokollierung sollen folgende Werte erfasst werden:

- Veränderter Wert (Geschäftsbuchnummer oder Projektbezeichnung)

### 3 Spezifikation

Buchungen-Suche

Filtermöglichkeiten

Kundenname

Testkunde

Geschäftsbuchnummer

1807/12

Projekt

Buntenbeck

Benutzer

Pepperl

Platzhaltersymbole:

% eine beliebige Anzahl von Zeichen

\_ ein Zeichen

Heute

Monatlich

Quartal

Datumsbereich

von:

23.08.2013

bis:

17.10.2013

Verwendungszweck

☒ eigener Gebrauch (einmalig) (eigG)

☐ Vermessungsunterlagen (amtlicher Lageplan TS 3)

☐ Vermessungsunterlagen (hoheitliche Vermessung

☐ Vermessungsunterlagen (sonstige) (VU s)

Kostenart

☒ kostenpflichtig

☒ kostenfrei

Suchen

Abbildung 3.3: Mockup des Buchungssuchdialog

### 3 Spezifikation

- alter Wert
- neuer Wert
- Benutzer der die Änderung vornimmt
- Zeitstempel

Diese Felder werden der Klasse Buchung hinzugefügt, so dass jede Buchung sein eigenes Änderungsprotokoll mit speichert.

#### **Stornierung von einzelnen Buchungen**

Wie oben bereits erwähnt soll es über den Buchungseitor möglich sein Buchungen zu stornieren. Beim Stornieren einer Buchung soll ein Grund ausgewählt werden können. Dieser gibt an, warum eine Buchung storniert werden musste. Um diese Auswahl darzustellen wird die Klasse Storno-Grund eingeführt, diese ist in 3.2.7 beschrieben.

Bei dem Vorgang der Stornierung soll ein Beamte ein Storno-Grund auswählen können und die Stornierung selbst anstoßen. Daraufhin soll die betroffene Buchung als storniert markiert werden und das aktuelle Datum, der Benutzer, der die Stornierung vornimmt, sowie der Storno-Grund sollen mit der Buchung abgespeichert werden.

#### **3.2.6 Berichte**

Wie bereits erwähnt sollen im Kundenrenderer und im Kundengruppenrenderer verschiedene Berichte bezüglich den dargestellten Kunden und Buchungen erstellt werden. In diesem Abschnitt werden diese drei Berichte beleuchtet. **TODO: Müssen noch besser spezifiziert werden.**

### 3 Spezifikation

#### **Buchungsbeleg**

Dieser Bericht soll die im Renderer angezeigten Buchungen tabellarisch darstellen. Dabei ist der Bericht in einen Kopfbereich und eine Tabelle mit den Buchungen eingeteilt.

Im Kopfbereich des Berichts sollen sich folgende Basisinformationen befinden:

- Kundenname
- Vertragsnummer
- Abrechnungszeitraum

Die Tabelle soll folgende Felder der Buchungen enthalten: **TODO: MwSt?**

- Gebühr
- Geschäftsbuchnummer
- Projektbezeichnung
- Zeit
- Verwendungszweck
- Produktbezeichnung

In den Renderern soll es möglich sein auszuwählen, ob im Bericht kostenlose Downloads angezeigt werden sollen oder nicht. Falls im Renderer keine Buchungen ausgewählt sind, so soll der Bericht nicht erstellt werden.

#### **Rechnungsanlage**

Die Rechnungsanlage ist eine Erweiterung des Buchungsbeleg. Diese besitzt zwischen dem Kopfbereich und der Tabelle ein zusätzliches Statistikfeld. Weiterhin ist die Rechnungsanlage ein Teil der Abrechnung, so dass die angezeigten Buchungen, beim Erstellen des Berichts, als abgerechnet markiert werden. Dies ist auch der Grund weshalb der Bericht im Kundenrenderer nicht von den externen Benutzern erstellt werden darf. Eine Vorlage eines solchen Berichts kann der Abbildung ?? auf Seite ?? entnommen werden.

### 3 Spezifikation

In den Renderern soll es möglich sein auszuwählen, ob im Bericht kostenlose Downloads angezeigt werden sollen oder nicht. Weiterhin soll festgelegt werden können, ob die Buchungen tatsächlich als abgerechnet markiert werden sollen, oder nicht.

#### **Geschäftsstatistik**

Die Geschäftsstatistik ist ein Bericht der Statistiken von den Downloads von verschiedenen Kunden darstellt. Durch die jeweilige Darstellung mehrerer Kunden, kann der Bericht nur im Kundengruppenrenderer erstellt werden. Die angezeigten Statistiken stellen Informationen zu der Aktivität der Benutzer, den Kostenarten der Produkte, den Verwendungszwecken, den Angaben für den Jahresbericht, sowie eine Übersicht über die bezogenen Produkte dar. Eine Vorlage des Berichts kann den Abbildungen ?? bis ?? auf Seite ?? entnommen werden.

#### **3.2.7 sonstige neu benötigte Klassen**

Der Hauptgrund warum diese, hier aufgeführten, Klassen eingeführt werden ist, die Unterstützung des Benutzers bei der Eingabe von den jeweiligen Werten. Mit diesen Klassen kann dem Benutzer eine vorgefertigte Auswahl an Elementen angeboten werden. Somit müssen die entsprechenden Felder nicht händisch in der GUI eingetragen werden. Diese Auswahl beschleunigt einerseits das Eintragen von Werten und verhindert andererseits Fehleingaben.

Zum Einführen der Klasse Kunde und zum Erweitern der Buchungen werden folgende zusätzlichen Klassen benötigt:

**Abrechnungsturnus** Der Abrechnungsturnus gibt an wie häufig ein Kunde eine Abrechnung erhält. Sein einziges Feld ist die Bezeichnung des Abrechnungsturnus (z. B. Jahr, Quartal)

**Produkt** Ein Produkt repräsentiert die verschiedenen Softwareprodukte, die ein Kunde benutzen kann. Seine Felder sind eine Bezeichnung des Produktes, eine Beschreibung sowie der konkrete Client, mit dem sich der Benutzer anmelden kann. Beim Client handelt es sich ebenfalls um eine Klasse.

### 3 Spezifikation

**Client** Ein Client kann von verschiedenen Produkten benutzt werden. Sein einziges Feld ist seine Bezeichnung.

**Stornogrund** Ein Stornogrund ist der Grund, warum eine Buchung storniert wurde. Sein einziges Feld ist die Bezeichnung des Grundes.

Bei diesen Klassen sollen die sogenannten Standardrenderer und -editoren verwendet werden. Diese werden für jede Klasse automatisch erstellt und müssen nicht eigens implementiert werden. Mit dem Standardeditor ist es möglich neue Objekte einer Klasse anzulegen und sämtliche Felder der Klasse verändern. Diese Anforderungen genügen allerdings für die hier aufgelisteten Klassen.

## 4 Realisierung

In diesem Kapitel wird die Realisierung der oben beschriebenen Erweiterungen erläutert. Im ersten Teil wird der aktuelle technische Stand erläutert, anschließend wird im zweiten Teil die Realisierung selbst beschrieben.

### 4.1 Aktueller technischer Stand

In diesem Abschnitt soll der aktuelle technische Stand aufgezeigt werden, dies dient einerseits der Verständnis bezüglich des zweiten Abschnitts dieses Kapitels. Andererseits soll somit verdeutlicht werden von welcher Basis ausgegangen wurde und welche Funktionalitäten neu implementiert werden mussten.

Die Informationen die für diesen Abschnitt benötigt wurden, stammen aus [Cisa] und [Cisb].

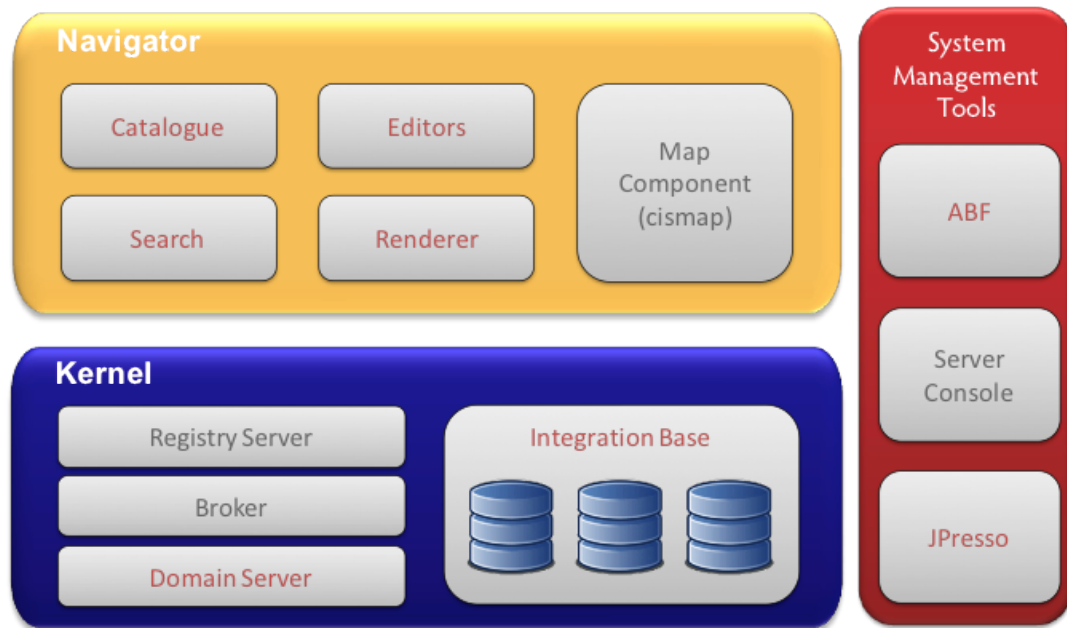
#### 4.1.1 Cids-Umgebung

In der bisherigen Arbeit wurde WuNda mehr oder weniger als einheitliches System betrachtet, aus technischer Sicht trifft dies jedoch nicht zu. Tatsächlich ist WuNda ein System, das sich aus mehreren Bestandteilen zusammensetzt und innerhalb der Cids-Umgebung realisiert wurde. Dabei ist es bei der Cids-Umgebung möglich, diese für besondere Zwecke zu erweitern, was auch der Fall bei WuNda ist. Die Hauptbestandteile dieser Umgebung sind der Navigator, der Kernel und verschiedene Entwicklungswerkzeuge. Eine Übersicht der Cids-Umgebung kann der Abbildung 4.1 auf der nächsten Seite entnommen werden. Diese Hauptbestandteile sind in weitere Unterkomponenten eingeteilt. Die in der Abbildung 4.1



#### 4 Realisierung

nicht ausgegrauten Komponenten sind für diese Arbeit relevant und werden im Folgenden detaillierter erläutert. Zunächst wird allerdings der Hauptzweck und der Aufbau des Cids-Umfeldes beschrieben. **TODO: Werden sie wirklich besser beschrieben?**



**Abbildung 4.1: Cids Komponenten**

Der Hauptzweck dieses Entwicklungsumfeldes ist das Erstellen von Informationssystemen, insbesondere Geoinformationssysteme (GIS). Dabei stehen verschiedene, häufig benötigte, Funktionalitäten bereits zur Verfügung. Dazu gehören eine Benutzerverwaltung sowie das Anzeigen, Bearbeiten und Suchen von Informationen. Außerdem können für die Benutzer Berechtigungen festgelegt werden, die ihnen den Zugriff auf Informationen erlauben oder verbieten. Weiterhin wird eine interaktive zweidimensionale Ansicht zu geographischen Daten, die sogenannte Cimap, bereit gestellt.

Durch das Cids-Umfeld kann ein verteiltes System erstellt werden, dessen Architektur auf dem Client-Server-Prinzip basiert, wobei es aus mehreren Clients, Server, sowie Datenbanken bestehen kann. Diese Bestandteile können in verschiedenen Ausprägungen nebeneinander existieren. In Wuppertal steht so z. B. neben dem WuNDa auch die VerDIS-Fachanwendung zur Verfügung. VerDIS dient zur Berechnung der Abwassergebühr. Dabei kann VerDIS auf Informationen aus

WuNDa zurück greifen. Um die Logik der Systeme zu trennen und um eine Skalierbarkeit zu gewährleisten, besitzt jedes System einen eigenen Domain Server und eine eigene Integration Base. Bei einer solchen Integration Base handelt es sich um eine Datenbank, in der neben den üblichen Daten auch Meta-Daten gespeichert werden. Durch diese Unterteilung der Daten können in der Integration Base beliebige Objekte abgebildet werden, weiterhin können dadurch auch Beziehungen zwischen den sogenannten Meta-Klassen und den Objekten abgebildet werden. Eine Integration Base erlaubt es zum Beispiel die Benutzer und Benutzergruppen eines Systems sowie deren Zugriffsrechte auf Objekte abzubilden. **TODO: ref auf abschnitt mit ABF. bildliche darstellung** Bei einem Domain Server handelt es sich um eine Schnittstelle zur Integration Base und hat die Aufgabe aus deren Daten konkrete Klassen und Objekte zu erstellen. Weiterhin ist der Domain Server auch für die Erstellung des Katalogs zuständig. Eine Beschreibung des Katalogs kann im Abschnitt 4.1.2 gefunden werden. In der Cids-Umgebung gibt es weiterhin die sogenannten Broker. Diese dienen den Clients als Proxy zu den einzelnen Domain Servern. Sie verstecken somit die Komplexität des verteilten Systems, indem sie die Anfragen der Clients an die entsprechenden Domain Server weiterleiten. Darüber hinaus gibt es noch den Registry Server. Dieser gewährleistet das Zusammenspiel der einzelnen Server, indem er z. B. die Namensauflösung für die restlichen Server bereitstellt.

In den folgenden Abschnitten des Kapitels werden die Bestandteile der Cids-Umgebung näher beleuchtet, die am relevantesten für diese Arbeit waren.

### 4.1.2 Navigator

Beim Navigator handelt es sich um eine erweiterbare graphische Benutzeroberfläche für GIS. So ist der Client von WuNDa eine Erweiterung des Navigators. Die für diese Arbeit wichtigsten Unterkomponenten sind der Katalog, die Suche sowie der Renderer und der Editor.

## Katalog

Beim Katalog handelt es sich um die Hauptnavigationsmöglichkeit des Navigators. Dieser Baum wird in der Integration Base mittels Meta-Daten beschrieben und wird bei Bedarf vom Domain Server erstellt und im Navigator angezeigt. Dieses Erstellen erfolgt dynamisch und ermöglicht es sämtliche Unterknoten, bis hin zur Objektebene, aus der Datenbank zu beziehen und zu erzeugen.

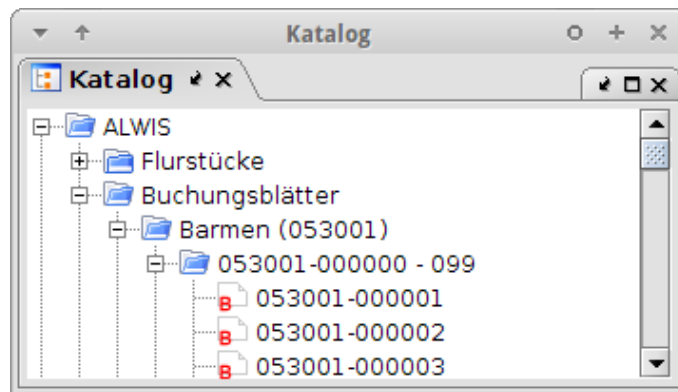


Abbildung 4.2: WuNda-Katlog eines ÖbVIs

Um den Aufbau des Katalogs mittels eines Beispiels zu verdeutlichen, kann der Katalog eines ÖbVI in WuNda der Abbildung 4.2 entnommen werden. Bei dem Knoten ALWIS<sup>1</sup> handelt es sich um einen sogenannten dynamischen Knoten. Ein solcher Knoten besitzt eine SQL-Query, mit der seine Unterknoten ermittelt und erstellt werden können. Diese Unterknoten können ebenfalls solche Queries besitzen, so dass diese wiederum eigene Unterknoten mit eigenen Queries besitzen. Auf diese Weise kann der Baum dynamisch gefüllt werden, wobei seine Blätter, in der Regel, konkrete Objekte darstellen. Ein Vorteile dieser Herangehensweise ist die Aktualität des Katalogs, da dieser zu jedem Zeitpunkt neu erstellt werden kann. Ein weiterer Vorteil ist das Anzeigen sämtlicher relevanten Kategorien und Objekte in einer zentralen Stelle. Durch dieses Anzeigen der kompletten Struktur wird eine unübersichtliche Struktur, die aus Menüs und verschiedenen Untermenüs besteht, vermieden.

<sup>1</sup>ALWIS (Alle Liegenschaftskataster-WuNda-Informationssysteme) fasst mehrere Teilsysteme von WuNda zusammen. Unter ALWIS finden sich die ALKIS-Benutzungskomponenten und alle weiteren Datenbestände die für die Katasterbenutzung relevant sind, so z.B. das Vermessungsregister [vgl. WUPc].

### Suche

Bei der Suche handelt es sich um eine Schnittstelle zum Kernel, die das Suchen von Objekten ermöglicht. Diese Suche kann an verschiedene Anwendungsfälle angepasst werden. Eine Suche kann aus der Eingabe von Suchkriterien in einer Maske bestehen, woraufhin nach passenden Objekten gesucht wird. Ein anderer Anwendungsfall der Suche kann die Karte (cimap) als Suchkriterium miteinbeziehen. Ein Beispiel einer solchen Suche wäre das Festlegen eines Bereiches in der Karte, woraufhin nach sämtlichen Objekte, einer bestimmten Klasse, gesucht wird, die sich in diesem Bereich befinden.

### Renderer und Editor

Die Renderer und Editoren wurden bereits in Abschnitt 3.1 beschrieben und werden deshalb hier nicht weiter erläutert.

### 4.1.3 ABF

Der Administrators' Best Friend (ABF) ist ein Werkzeug, mit dem die Meta-Daten der Integration Base visualisiert und bearbeitet werden können. Mit ihm können Klassen (vergleiche Abschnitt 3.1) und ihre Attribute, sowie Benutzer und Benutzergruppen angelegt und verwaltet werden.

Der ABF bietet ebenfalls einige Funktionalitäten, die das Erweitern des Katalogs vereinfachen. Die beiden Hauptfunktionalitäten dabei sind eine Vorschau des Kataloges, wobei auch die dynamischen Knoten aufgelöst und deren Unterknoten angezeigt werden. Weiterhin können die Informationen bezüglich jeden Knotens angezeigt, so z. B. die vergebenen Rechten und ihre SQL-Query. Diese Informationen können auch für die nicht dynamisch erstellten Knoten verändert werden. Eine weitere Funktionalität ist das Anlegen und Verwalten der verschiedenen Berechtigungen für die Benutzer. Da diese Funktionalität des ABF eine wichtige Rolle in dieser Arbeit spielen, werden sie im Folgenden näher beleuchtet.

### Berechtigungen

In diesem Unterabschnitt wird kurz auf das Rechtesystem der Cids-Umgebung eingegangen. In dieser Arbeit werden zwei Arten von Rechte benötigt. Die Lese- und Schreibrechte auf Klassen, Attribute und Knoten des Katalogs sowie die Aktionsrechte.

**Rechte auf Objekte** Für jede Klasse, Attribut oder Knoten lassen sich drei Standardverhalten festlegen, wobei immer ein Verhalten benutzt werden muss. Damit nicht für jedes Objekt ein Verhalten explicit gesetzt werden muss, wurde die Fallback Strategie eingeführt. Diese Strategie ermöglicht es z. B. ein Verhalten als Standard-Server-Verhalten festzulegen, woraufhin jede Klasse ohne explizit gesetztes Verhalten auf dieses zurückgreift.

Die verfügbaren Standardverhalten sind:

**Standard** Besitzt ein Objekt dieses Verhalten so bedeutet dies, dass jeder Benutzer lesen und niemand schreiben darf.

**Wiki** Bei diesem Standardverhalten darf jeder lesen und schreiben.

**Secure** Bei diesem Standardverhalten darf niemand lesen und schreiben

Für jedes Objekt können diese Rechte für verschiedene Benutzergruppen erweitert oder eingeschränkt werden. Für eine Klasse mit dem Verhalten „Secure“ kann z. B. festgelegt werden, dass die Administratoren Lese- und Schreibrechte besitzen. Im Gegensatz hierzu kann einer Benutzergruppe für eine Klasse mit dem Verhalten „Standard“, die Leserechte entzogen werden. [vgl. Ric10]

**Aktionsrecht** In manchen Fällen sind die Rechte auf Objekte zu grobkörnig und feinere Einschränkungen müssen definiert werden können. In diesen Fällen können die Aktionsrechte, auch Action Tags genannt, benutzt werden. Ein Aktionsrecht besitzt jeweils einen eindeutigen Namen sowie eine Liste mit Besitzern. In dieser Liste können sich ganze Benutzergruppen oder einzelne Benutzer aus einer Benutzergruppe befinden.

## 4 Realisierung

Ein Aktionsrecht kann benutzt werden um, im laufenden Programm, zu überprüfen, ob der aktuell angemeldete Benutzer das Recht hat eine bestimmte Aktion durchzuführen. Für jede solche Aktion muss es demnach je ein solches konfiguriertes Aktionsrecht geben. Der Benutzer hat das Recht die Aktion durchzuführen, falls er ein Besitzer des dazugehörigen Aktionsrechtes ist. Zu solchen Aktionen gehört zum Beispiel die Anzeige bestimmter Suchen oder das Aktivieren/Deaktivieren von Benutzungselementen in einem Renderer.

### 4.1.4 JPresso

JPresso ist ein ETL (Extraction, Transaction & Loading)-Werkzeug das eine graphische Benutzeroberfläche besitzt und es ermöglicht verschiedene Datenbestände in eine Datenbank zu importieren. Bei diesem Importieren können die Zieltabellen und Spalten für die Datenquellen frei gewählt werden und es können beliebige Datenmanipulationen vorgenommen werden. Weiterhin bietet JPresso eine Normalisierungsfunktion an. Durch diese Funktion wird die Verbindung zwischen verschiedenen, bereits bestehenden Tabellen oder Datenquellen aufgedeckt und sie werden automatisch mittels ihrer entsprechenden ID referenziert. [vgl. Cis13b]

## 4.2 Technische Realisierung

Dieser Abschnitt beschreibt die Realisierung der in Kapitel 3 beschriebenen gewünschten Erweiterungen in WuNDa.

### 4.2.1 Datenbankmodell

**TODO: DB-Modell ist noch so naja** Um die in Kapitel 3 beschriebenen Anforderungen zu realisieren, muss in einem ersten Schritt das Datenbankmodell der verwendeten Datenbank angepasst werden. Dies wird zum einen benötigt da die, durch die neuen Erweiterungen, anfallenden Daten abgespeichert werden müssen. Zum anderen müssen aus den Daten, die im Client benötigt, CidsBeans erstellt

## 4 Realisierung

werden können. Dies ist notwendig, da die verschiedenen Erweiterung quasi alle auf den Funktionalitäten der CidsBeans aufbauen.

Durch die objektrelationale Abbildung die zwischen den Tabellen der Datenbank und den CidsBeans passiert wird im Folgenden von Klassen geredet. Dabei sind auch die Tabelle der Datenbank und ihr Aufbau gemeint, da deren Struktur nicht oder nur geringfügig von der Struktur der erstellten Klassen abweicht.

### Ausgangslage

In einem ersten Schritt wurden die bereits vorhandenen, benötigten Klassen betrachtet. Hierbei wurde festgestellt, dass einzig die Klasse der Buchungen, die BILLING-Klasse vorhanden ist. Diese Klasse ist in der Abbildung 4.3 abgebildet.

**TODO: beschreiben**

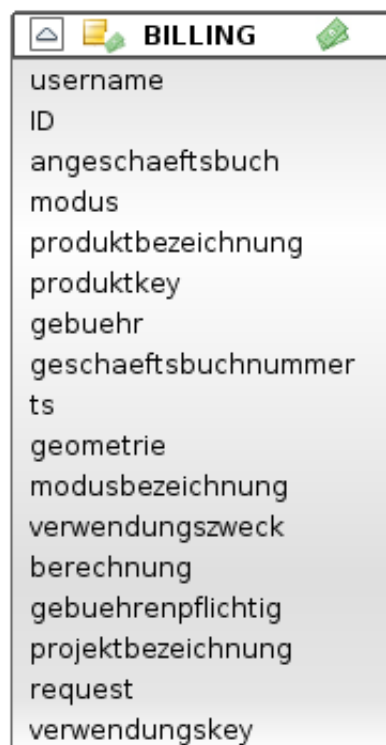


Abbildung 4.3: Ausgangslage des Datenbankmodells

### Weitere Schritte

Wie bereits in Kapitel 3 beschrieben, waren sämtliche Anforderungen nicht von Anfang an bekannt, sondern wurden schrittweise erweitert. Dieser Prozess ist am deutlichsten am Datenbankmodell zu betrachten, welches mit den Anforderungen gewachsen ist. Da die einzelnen Schritte für das Verständnis der Arbeit nicht relevant sind, werden diese nicht im Detail erläutert. Um das Anwachsen des Datenbankmodells trotzdem zu verdeutlichen, wird eine der ersten Versionen kurz beschrieben, um anschließend die schlussendliche Version abzubilden und zu kommentieren.

Für den ersten Schritt wurden die anfänglich existierenden Anforderungen analysiert und die erste Version des Datenbankmodells wurde erstellt. Die Abbildung 4.4 auf der nächsten Seite zeigt einen recht frühen Stand des Datenbankmodells. Dennoch sind die meisten Klassen die in den Spezifikationen aufgeführt werden bereits vorhanden und die Verbindungen zwischen den einzelnen Klassen bestehen ebenfalls bereits.

### Letzter Stand

Der letzte Stand des Datenbankmodell kann in Abbildung 4.5 auf Seite 59 betrachtet werden. **TODO: Schönere DB-Modell beschreibung** Im Vergleich zu der vorherigen Version haben sämtliche Klassen das Präfix „BILLING\_“ erhalten. Die Buchungen-Klasse hat die Felder zum Speichern der letzten Änderung erhalten. Weiterhin wurde ihr Feld „gebuhr“ in „netto\_summe“ umbenannt und sie hat die Felder „brutto\_summe“ und „mwst\_satz“ erhalten. Außerdem wurde das Feld „abgerechnet“ hinzugefügt. Die Kunden-Login-Klasse hat ein eigens Kontaktfeld erhalten. Die Kunden-Klasse hat ein Feld „name\_intern“ erhalten. In sämtlichen Klassen wurden die Bezeichnungsfelder in „name“ umbenannt. Somit erfüllen sie eine Namenskonvention und der Wert dieser Felder wird automatisch in Java bei der toString()-Methode ausgegeben. Einige Felder wurden umbenannt, so dass auf den ersten Blick erkennbar ist, dass es sich um besondere Felder handelt. Dies ist der Fall bei Array-Felder oder 1:n Felder. Siehe z. B. „produkte\_arr“ und „benutzer\_n“ der Kundenklasse. Weiterhin wurde die



## 4 Realisierung

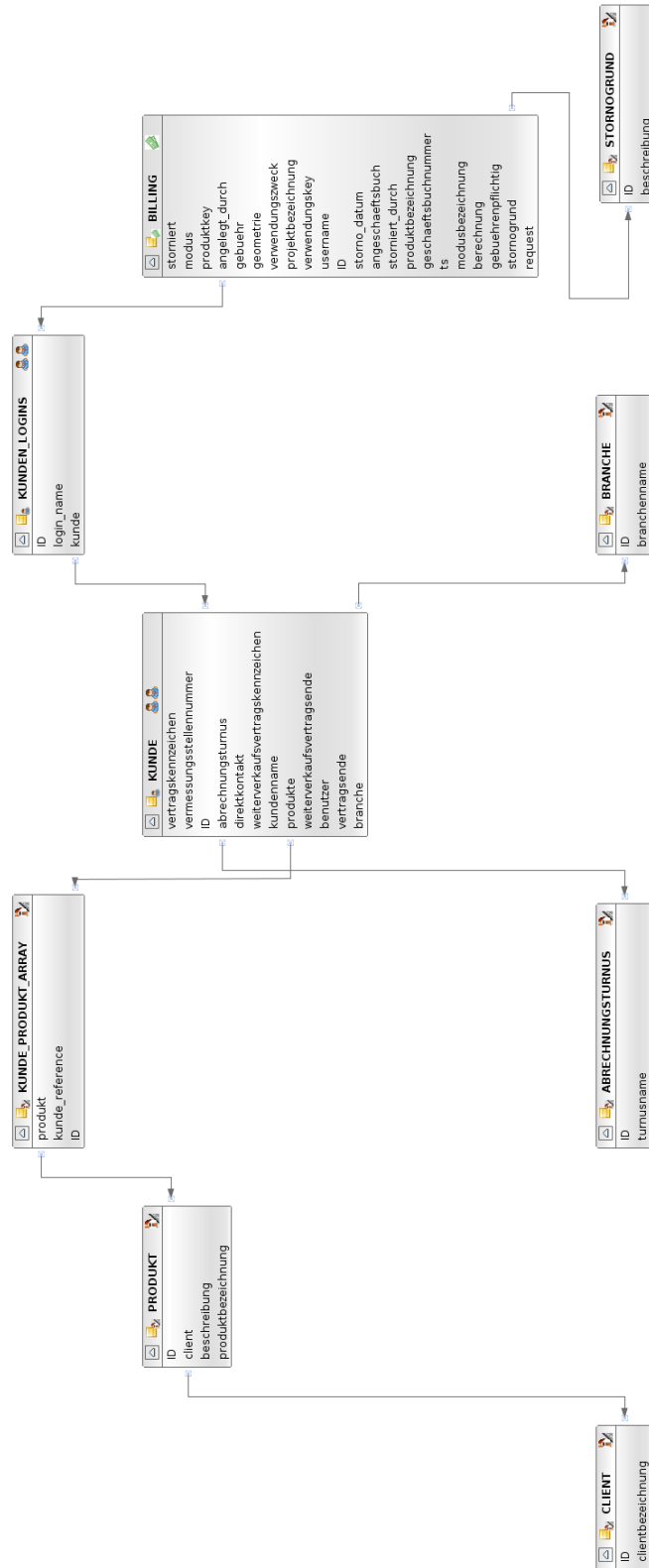


Abbildung 4.4: Zweiter Stand des Datenbankmodells

Kundengruppen-Klasse eingeführt, dies zeigt z. B. dass diese erst recht spät bei dem Ermitteln der Anforderungen aufgetreten ist.

### 4.2.2 Import von neuen Daten

Nachdem Erstellen des Datenbankmodell musste die Datenbank entsprechend erweitert werden. Eine gute Möglichkeit dies im Cids-Umfeld zu realisieren bietet der bereits erwähnte ABF. Mit diesem Werkzeug wurden die benötigten Klassen angelegt und die entsprechenden Tabellen konnten in die Datenbank eingespielt werden.

Für die weitere Entwicklung der Funktionalitäten wurden Anfangsdaten benötigt, die vom R102 in Form einer Excel-Datei geliefert wurde. Diese Datei enthielt die Informationen zu tatsächlichen Kunden und Kunden-Logins, wobei z. B. deren Abrechnungsturnus und Produkte angegeben war. Diese Datei wurde in mehrere CSV-Dateien aufgeteilt, die dann in eine Hilfsdatenbank eingespielt wurden. Mit Hilfe von JPresso konnten die Daten aus der Hilfsdatenbank in die eigentliche Datenbank importiert werden. Weiterhin konnten mit Hilfe dieses Werkzeugs die Buchungen aus der „BILLING“-Tabelle in die neue „BILLING\_BILLING“-Tabelle überführt werden. Bei diesem Vorgang wurde vermehrt auf die Normalisierungsfunktion von JPresso zurückgegriffen um die Fremdschlüssel zwischen den neuen Tabellen zu realisieren.

Ein besonderer Fall beim Importieren war der MwSt.-Satz, da dieser bei den bereits angelegten Buchungen nicht existierte. Dieser konnte beim Importieren nicht auf einen Wert von 0,0 % gesetzt werden, da dies fachlich falsch wäre. Dies liegt daran, dass es Produkte mit einem anderen MwSt.-Satz gibt als 0,0 %, so haben z. B. Orthofotos einen MwSt.-Satz von 19,0 %. Gäbe es nun Orthofotos mit 0,0 % und andere mit 19,0 % so würde dies wahrscheinlich für Verwirrung beim Endbenutzer sorgen, da es sich um einen inkonsistenten Fall handelt. Eine andere Lösung wäre das Anpassen der Datenbestände gewesen, also das Umändern der Netto- und Brutto-Summen sowie des MwSt.-Satzes. Hiergegen wurde sich entschieden, da es sich um kritische Daten handelt und ein kleiner Fehler finanzielle Auswirkungen für R102 oder einen Kunden hätte. Aus diesem Grund wurde beim Importieren der MwSt.-Satz auf `null` gesetzt. Durch diesen Sonderfall

## 59



## 4 Realisierung

kann der MwSt.-Satz, beim Anzeigen, anders behandelt werden und es entsteht keine Verwirrung beim Benutzer. Weiterhin kann bei Berechnungen `null` genau wie der Satz 0,0% behandelt werden, so dass die Netto-Summe immer gleich der Brutto-Summe ist. Hierdurch ist gewährleistet, dass am ursprünglichen Preis nichts verändert wird.

### 4.2.3 Anpassung des Downloadprotokoll-Dialoges

Durch die Änderung der Struktur der Buchungen-Klasse musste auch die Klasse des Downloadprotokoll-Dialoges `BillingPopup` angepasst werden. Dabei sollte an der eigentlichen Funktionalität von `BillingPopup`, die bereits im Abschnitt 2.4.1 beschrieben wurde, nichts geändert werden. Lediglich die graphische Oberfläche wurde dahingehend verändert, dass die Netto- und Brutto-Summen der anfallenden Gebühren sowie der anfallende MwSt-Satz angezeigt werden. Ein Screenshot des überarbeiteten Downloadprotokoll-Dialoges kann in Abbildung 4.6 gefunden werden.

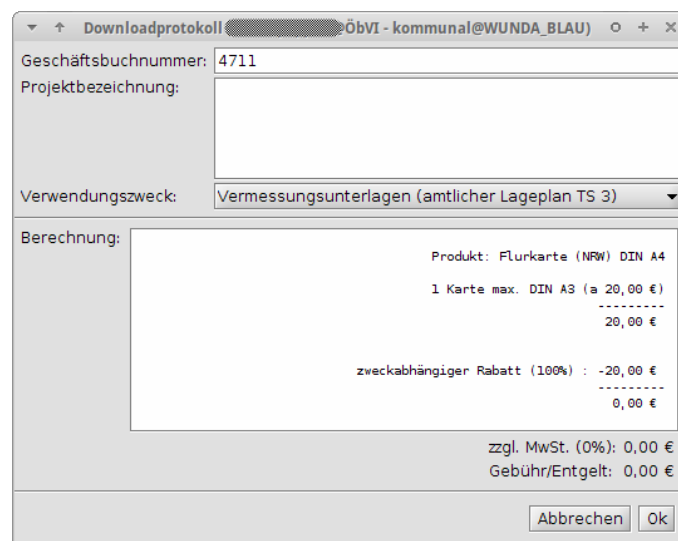


Abbildung 4.6: Downloadprotokoll-Dialog mit MwSt.-Satz

#### 4.2.4 Zeitliche Filter und Verwendungszweck-Filter

Beim Betrachten der benötigten Filter in der Spezifikation fällt auf, dass ähnliche Filtermöglichkeiten in den verschiedenen Renderer und der Suche benötigt werden. Aus diesem Grund wurde der zeitliche Filter und Verwendungszweck-Filter als eigene `JPanel` angelegt, so dass diese mehrfach verwendet werden können.

##### Zeitlicher Filter

Das eigenständige `JPanel TimeFilterPanel` gibt dem Benutzer die Möglichkeit nichts, einen Tag oder einen Zeitspanne auszuwählen. Dabei werden ihm Schnellwahlmöglichkeiten für den aktuellen Tag, einen Monat oder ein Quartal angeboten. Weiterhin kann er mittels der Angabe eines von-Datums und eines bis-Datums einen Zeitspanne festlegen. Bei Bedarf kann die aktuelle Einstellung des Panels abgefragt werden, wobei der Rückgabewert zwei `Date`-Variablen sind. Sind beide Variablen `null` so wurde keine Zeit ausgewählt. Ist nur das von-Datum belegt so wurde ein einzelner Tag ausgewählt. Sind beide Variablen belegt so wurde eine Zeitspanne festgelegt.

Während dem Testen der Renderer kam die zusätzliche Anforderung auf, dass die Datumsauswahl jeweils eine Schnellauswahl für das Jahr benötigt. Das gewünschte Result kann in der Abbildung 4.7 betrachtet werden.

Da die verwendete Komponente `JXDatePicker` dies offiziell nicht anbietet musste ein Workaround gefunden werden, welches unter [Rad13] gefunden wurde. Um diesen Workaround umzusetzen musste ein sogenanntes `TakeoffHook` implementiert werden. Dies ist eine Klasse die mittels des Lookup-Mechanismus beim Starten von WuNDa gefunden und ausgeführt wird. Somit wird beim Starten des Programmes das `TakeoffHook JXDatePickerHeaderTakeoff` ausgeführt, das die zwei Zeilen aus Listing 4.1 beinhaltet.



**Abbildung 4.7: Datumsauswahl**

Listing 4.1: innerhalb JXDatePickerHeaderTakeoff

```
1 UIManager.put(CalendarHeaderHandler.uiControllerID, "org.  
    jdesktop.swingx.plaf.basic.SpinningCalendarHeaderHandler");  
2 UIManager.put(SpinningCalendarHeaderHandler.  
    ARROWS_SURROUND_MONTH, Boolean.TRUE);
```

In der ersten Zeile wird angegeben dass für die `JXDatePicker` eine Kopfzeile verwendet wird, die eine Schnellauswahl des Jahres zulässt. In der zweiten Zeile wird die Anordnung des rechten Pfeils geändert, dieser würde sich ansonsten ganz rechts außen befinden. Damit die Kopfzeile einer `JXDatePicker`-Instanz `datePicker` entsprechend geändert wird, muss die Zeile aus Listing 4.2 ausgeführt werden.

Listing 4.2: Setzen der Kopfzeile

```
1 datePicker.getMonthView().setZoomable(true)
```

Genau diese Zeile enthält den problematischen Teil dieses Workaround, da das `Zoomable`-Attribut von `JXMonthView`, eine Anzeige von Monaten, über die ein Datum gewählt werden kann, noch nicht richtig implementiert ist und keine Auswirkungen haben dürfte [vgl. Swi09]. Dennoch wird hierdurch die gewünschte Kopfzeile angezeigt.

### Verwendungszweck-Filter

Der Verwendungszweck-Filter ist ebenfalls ein eigenständiges `JPanel` und wurde in der Klasse `VerwendungszweckPanel` implementiert. In diesem Panel werden die möglichen Verwendungszwecke untereinander angezeigt, wobei jedes eine `Checkbox` besitzt, mit der es an oder abgewählt werden kann. Weiterhin ist es nicht möglich sämtliche Verwendungszwecke abzuwählen. Die Besonderheit dieses Panel besteht darin, dass die Verwendungszwecke nicht hart kodiert sind, sondern aus der `billing.json` Datei geladen werden. Diese JavaScript Object Notation (JSON)-Datei beinhaltet Informationen bezüglich den Buchungen. Sie enthält neben den Verwendungszwecken auch die Gebühren und MwSt.-Sätze zu den einzelnen Produkten.

### 4.2.5 Suche nach Buchungen

Beim Analysieren der Anforderungen ist ebenfalls aufgefallen, dass sowohl die Kunden- und Kundengruppenrenderer als auch der Suchdialog zu den Buchungen eine Möglichkeit benötigen um in der Datenbank nach Buchungen mit bestimmten Kriterien zu suchen. Aus diesem Grund wurde auf die Such-Funktionalität der Cids-Umgebung zurückgegriffen und es wurde eine sogenannte Server Search implementiert. Um diesen Mechanismus umzusetzen wurde die Klasse `CidsBillingSearchStatement` erstellt. Diese erbt von der, bereits bestehenden, Klasse `AbstractCidsServerSearch` und besitzt somit die Grundfunktionalität zu einer solchen Server Search. Um diese Server Search in den Renderern oder dem Suchdialog auszuführen müssen lediglich die gewünschten Kriterien einem Objekt der Klasse `CidsBillingSearchStatement` übergeben werden und die eigentliche Suche muss gestartet werden. Daraufhin können die gefundenen Buchungen dort weiterverarbeitet und angezeigt werden.

`CidsBillingSearchStatement` kann die Geschäftsbuchnummer, die Projektbezeichnung und den Kundennamen als Kriterien entgegennehmen. Dabei wird jeweils überprüft, ob das angegebene Kriterium ein Teilwort des jeweiligen Feldes ist. Für diese Kriterien können außerdem die beiden Wildcards `'_'` und `'%'` für ein einzelnes Zeichen bzw. für eine beliebige Anzahl von Zeichen benutzt werden. Als weiteres Kriterium kann ein Kunden-Login angegeben werden. In dem Fall wird nur nach solche Buchungen gesucht, die mit jenem Kunden-Login angelegt wurden. Weitere Kriterien sind der Zeitbereich in denen die Buchungen erstellt werden mussten, sowie der Kostentyp, sprich ob die einzelne Buchung kostenpflichtig oder kostenfrei ist. Weiterhin kann angegeben werden, ob stornierte oder abgerechnete Buchungen gefunden werden sollen. Außerdem kann eine Liste mit Kunden und einem, optionalen, Abrechnungsturnus übergeben werden. In dem Fall werden nur Buchungen von jenen Kunden mit eben diesem Abrechnungsturnus gefunden. Das einzige Pflichtkriterium ist eine Information bezüglich den Kunden. So muss ein Kundename oder eine Liste mit Kunden angegeben werden.

### 4.2.6 Kunden

In diesem Abschnitt werden die graphischen Oberflächen beschrieben, die den Kunden betreffen.

#### Kundenrenderer

Die Funktionalität des Kundenrenderers wurde bereits größtenteils in dem Abschnitt 3.2.3 beschrieben. Der dort nicht beschriebene Knopf „Ergebnisse anzeigen“ wurde hinzugefügt, damit die Suche bei Bedarf ausgeführt werden kann. Dieser wurde während der Entwicklung hinzugefügt, da sich die automatisch ausgeführte Suche bei jeder Änderung der Filter als unpraktikabel herausstellte. Dies liegt an der Dauer der Suche, die für eine solche Funktionalität zu ausgedehnt ist, obwohl diese, in den meisten Fällen, nur ein paar Sekunden beträgt. Bei Betätigen des Knopfes wird ein Objekt der Klasse `CidsBillingSearchStatement` mit den verschiedenen Kriterien aus den Filtern gefüllt und die eigentliche Suche wird ausgeführt. Die Ergebnisse dieser Suche werden dann in der Tabelle angezeigt und eine entsprechende Zusammenfassung der Suchergebnisse wird über der Tabelle ausgegeben. Weiterhin können die beiden Berichte Rechnungsanlage und Buchungsbeleg erstellt werden, wobei die im Abschnitt 3.2.6 aufgeführten Bedingungen gelten. So dürfen externe Benutzer die Rechnungsanlage nicht erstellen und es kann ausgewählt werden, ob kostenfreie Buchungen in den Berichten angezeigt werden sollen oder nicht. Ein Screenshot des Kundenrenderers kann in Abbildung 4.8 auf der nächsten Seite eingesehen werden.

#### Kundeneditor

Der Kundeneditor, der in Abbildung 4.9 auf Seite 66 dargestellt ist, ermöglicht es einen neuen Kunden anzulegen oder die Daten eines bereits bestehenden Kunden zu bearbeiten. Bei den meisten Felder handelt es sich um gewöhnliche Texteingaben oder Auswahlmöglichkeiten. Allerdings konnte bei den Felder für die Produkte und Kundengruppen auf die Drag-and-Drop Funktionalität der Cids-Umgebung zurückgegriffen werden. Somit können Objekte dieser Klassen aus dem



## 4 Realisierung

**Kundenportal ÖbVI**

Filtermöglichkeiten

Ignorieren Heute Monat Quartal Datumsbereich

Quartal: Oktober - Dezember

Jahr: 2013

Kostentyp

☒ Kostenfrei

☒ Kostenpflichtig

Verwendungszweck

☒ Wiederverkauf (einmalig)

☒ eigener Gebrauch (gebührenbefreit)

☒ eigener Gebrauch (einmalig)

☒ Vermessungsunterlagen (amtlicher Lageplan TS 3)

☒ Vermessungsunterlagen (sonstige)

☒ Vermessungsunterlagen (hoheitliche Vermessung TS 4)

Ergebnisse anzeigen

**Buchungen**

Der Bruttopreis beträgt 222,75 € für 85 Buchungen zwischen dem 01.10.2013 und dem 31.12.2013. (Abgerechnete oder stornierte Buchungen werden nicht angezeigt.)

Geschäftsbuchnummer	Projektbezeichnung	Verwendung	Produkt	Preis (netto)	MwSt-Satz	Datum	Benutzer
13000-108		VU al	Vermessungsriß	0,00 €		-19.12.2013	
13-114		VU hv	NAS-Daten (nur Pun...	0,00 €		-05.10.2013	
13-133		VU hv	AP-Karte	0,00 €		-02.10.2013	
13-141		VU hv	Vermessungsriß	0,00 €		-05.10.2013	
13-141		VU hv	Flurstücks- und Eige...	0,00 €		-05.10.2013	
13-141		VU hv	Punktnummerierung...	0,00 €		-05.10.2013	
13-114		VU hv	NAS-Daten (nur Pun...	0,00 €		-07.10.2013	
13-114		VU hv	NAS-Daten (nur Pun...	0,00 €		-08.10.2013	
13-114		VU hv	NAS-Daten (ohne Eig...	0,00 €		-08.10.2013	
13-114		VU hv	NAS-Daten (nur Pun...	0,00 €		-10.10.2013	
13-133		VU hv	NAS-Daten (nur Pun...	0,00 €		-11.10.2013	

Rechnungsanlage ☒ Kostenfreie Downloads ausblenden

☐ Downloads als abgerechnet kennzeichnen

Buchungsbeleg ☒ Kostenfreie Downloads ausblenden

Abbildung 4.8: Kundenrenderer

Katalog in den Editor gezogen werden und die Beziehung zwischen den jeweiligen Objekten wird automatisch erstellt.

Dieser Drag-and-Drop-Mechanismus bot sich auch bei den Kunden-Logins an, allerdings wird in dem Fall die Beziehung nicht automatisch erstellt, da es sich um eine 1:n-Beziehung handelt. Aus diesem Grund musste in das entsprechende Kunden-Login-Objekt aus der Datenbank bezogen werden und die Beziehung gesetzt werden.

### Kunden-Login Editor und Renderer

Auf Grund der 1:n-Beziehung funktionierte das automatische Erstellen des Standardeditors und -renderers nicht für die Kunden-Login-Objekte. Aus diesem Grund musste ein Editor erstellt werden, mit dem der Name und die E-Mail-Adresse angegeben sowie ein Kunde ausgewählt werden können. Beim Renderer handelt es

## 4 Realisierung

The screenshot shows a software window titled 'Attributeditor'. Inside, there's a header bar with 'Kunde: [redacted] Immobilien KG'. Below this, a form is displayed with various fields for customer data. The fields are organized into two main sections: a top section with individual fields and a bottom section with a large text area and a 'Logins' field. The 'Kundengruppe' field at the bottom is set to 'alle\_gekündigten\_Kunden'.

Name:	[redacted]
interner Name:	[redacted]
Vertragskennzeichen:	NV_Web_2-2008
Vertragsende:	31.12.2012
Weiterverkaufsvertragskennzeichen:	[redacted]
Weiterverkaufsvertragsende:	[redacted]
Vermessungsstellennummern:	[redacted]
Direktkontakt:	[redacted]
Branche:	Immobilienmakler
Abrechnungsturnus:	Quartal
Produkte:	[redacted]
Logins:	[redacted]
Kundengruppe:	alle_gekündigten_Kunden

Abbildung 4.9: Kundeneditor

sich um die gleiche Oberfläche, mit dem Unterschied, dass sie keine Eingaben entgegen nimmt.

### 4.2.7 Buchungen

Für die Buchungen musste ein Renderer, ein Editor und ein Suchdialog erstellt werden.

#### Buchungseditor und -renderer

Beim Buchungseditor und -renderer handelt es sich um die gleiche graphische Oberfläche, wobei beim Buchungseditor die Buchung verändert werden kann. Die veränderbaren Werte sind die Geschäftsbuchnummer und die Projektbezeichnung, wobei die letzte Änderung jeweils protokolliert werden muss. Weiterhin kann im

## 4 Realisierung

Editor eine Buchung, unter Angabe eines Stornogrundes, storniert werden. Die genauere Beschreibung dieser Oberflächen findet sich unter Abschnitt 3.2.5 und kann unter Abbildung 4.10 betrachtet werden.

The screenshot shows a web application titled 'Attributeditor'. The main header displays 'Buchung: ÖbVI' followed by a blurred ID and the date '- 10-069 - 28.11.2013'. Below this, there are three main sections:

- mögliche Änderungen**: Contains two input fields: 'Geschäftsbuchnummer:' with the value '10-069' and 'Projektbezeichnung:' with the value 'Baustelle 13'.
- Storno**: Contains a checkbox 'Buchung ist storniert', a dropdown menu for 'Stornogrund:', and a button 'Buchung stornieren'.
- Informationen**: Contains a list of details:
  - Abgerechnet: ☐
  - Modus: Selbstentnahme (ÖbVI)
  - Angelegt durch: [blurred]
  - Verwendungszweck: Vermessungsunterlagen (hoheitliche Vermessung TS 4)
  - letzte Änderung: Änderung von admin@Billing\_intern@WUNDA\_BLAU am 31.01.2014  
Attribut: Projektbezeichnung  
Alter Wert:  
Neuer Wert: Baustelle 13
  - Berechnung: Produkt: Punktliste (TEXT)  
6 Punkte unrabattiert (a 0,20 €)  
-----  
1,20 €  
zweckabhängiger Rabatt (100%) : -1,20 €  
-----  
0,00 €
  - Produkt: Punktliste (TEXT)
  - Erstellt am: 28.11.2013
  - Netto Summe [€]: 0.0
  - MwSt [%]:

Abbildung 4.10: Buchungseditor

### Suchdialog zu den Buchungen

Der Suchdialog basiert auf der Beschreibung aus Abschnitt 3.2.5 und bietet eine Möglichkeit verschiedene Kriterien bezüglich Buchungen anzugeben, woraufhin nach Buchungen mit jenen Kriterien gesucht werden kann. Die graphische Oberfläche der Suche wurde in der Klasse `BillingWindowSearch` umgesetzt. Diese Klasse kann über einen Lookup-Mechanismus gefunden werden und wird automatisch als Eintrag des Fenster-Menüs von WuNda angezeigt. Über diesen Eintrag kann der Suchdialog geöffnet und geschlossen werden.

Die eigentliche Suche wurde über die Server Search, die auf Seite 63 beschrieben wurde, umgesetzt. Wobei die gefundenen Ergebnisse anschließend in den Suchergebnissen von WuNda angezeigt werden. Ein Beispiel der Suchergebnisse kann in der Abbildung 4.11 betrachtet werden, dabei wurde das Kontextmenü für den ersten Eintrag geöffnet. Über dieses Menü kann der Editor der ausgewählten Buchung geöffnet werden. Da es sich bei dem Kontextmenü um eine Standardkomponente der Cids-Umgebung handelt, befindet sich hier auch die Option „Objekt löschen“. Diese wurde jedoch so eingestellt, dass ein Löschen nicht möglich ist. Ein Screenshot des Suchdialog selbst kann in Abbildung 4.12 auf der nächsten Seite gefunden werden.

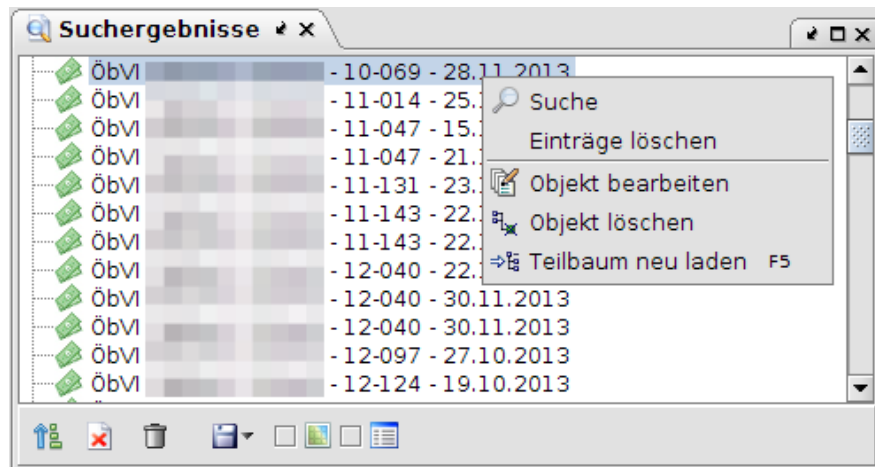


Abbildung 4.11: Suchergebnisse mit Buchungen und Kontextmenu

#### 4 Realisierung

**Buchungen-Suche**

**Filtermöglichkeiten**

Kundenname:

Abrechnungsturnus:

Geschäftsbuchnummer:

Projekt:

Benutzer:

**Wildcards:**  
% eine beliebige Anzahl von Zeichen  
\_ ein einzelnes Zeichen

Ignorieren Heute Monat Quartal **Datumsbereich**

von:

bis:

**Verwendungszweck**

- ☒ Wiederverkauf (einmalig)
- ☒ eigener Gebrauch (gebührenbefreit)
- ☒ eigener Gebrauch (einmalig)
- ☒ Vermessungsunterlagen (amtlicher Lageplan TS 3)
- ☒ Vermessungsunterlagen (sonstige)
- ☒ Vermessungsunterlagen (hoheitliche Vermessung TS 4)

**Kostentyp**

- ☒ Kostenfrei
- ☒ Kostenpflichtig

**Abgerechnet/Storniert**

- ☐ nur abgerechnete Buchungen anzeigen
- ☐ nur stornierte Buchungen anzeigen

Abgerechnete oder stornierte Buchungen werden nicht angezeigt.

Suchen

Abbildung 4.12: Suchdialog der Buchungen

## 4.2.8 Kundengruppen

### Kundengruppenrenderer und Kundengruppeneditor

Wie bereits beschrieben bietet der Kundengruppenrenderer einige Filterkriterien, so dass nicht gewünschte Buchungen nicht in die Ergebnisse miteinbezogen werden. Im Gegensatz zum Kundenrenderer werden die einzelnen Buchungen nicht angezeigt, sondern diese werden für den jeweiligen Kunden zusammengefasst und diese Zusammenfassung wird in der Tabelle dargestellt. Zum Erstellen der Rechnungsanlage und des Buchungbeleges stehen die gleichen Optionen zur Verfügung, wie im Kundenrenderer. Zusätzlich kann die Geschäftsstatistik als Bericht erstellt werden. Ein Screenshot des Kundengruppenrenderer kann in der Abbildung 4.13 gefunden werden.

**Kundengruppe Abrechnung\_Downloads\_quartalsweise: beinhaltet 9 Kunden**

**Filtermöglichkeiten**

Ignorieren Heute Monat Quartal Datumsbereich

Monat: Januar

Jahr: 2014

Abrechnungsturnus: [Dropdown]

☒ Kostenfreie Downloads ausblenden

**Verwendungszweck**

- ☒ Wiederverkauf (einmalig)
- ☒ eigener Gebrauch (gebührenbefreit)
- ☒ eigener Gebrauch (einmalig)
- ☒ Vermessungsunterlagen (amtlicher Lageplan TS 3)
- ☒ Vermessungsunterlagen (sonstige)
- ☒ Vermessungsunterlagen (hoheitliche Vermessung TS 4)

**Buchungen**

Aggregierte Buchungen zwischen dem 01.01.2014 und dem 31.01.2014. (Abgerechnete oder stornierte Buchungen werden nicht angezeigt.)

Auswahl für Berichte	Kundenname	aggregierter Preis (brutto)	kostenpflichtige Downloads
<input checked="" type="checkbox"/>	ObVI	107,64 €	4
<input checked="" type="checkbox"/>	ObVI	104,90 €	4
<input checked="" type="checkbox"/>	ObVI	45,63 €	2
<input checked="" type="checkbox"/>	ObVI	91,53 €	2
<input checked="" type="checkbox"/>	ObVI	48,96 €	1

**Rechnungsanlage** ☒ Kostenfreie Downloads ausblenden  
☐ Downloads als abgerechnet kennzeichnen

**Buchungsbeleg** ☒ Kostenfreie Downloads ausblenden

**Geschäftsstatistik**

Abbildung 4.13: Kundengruppenrenderer

## 4 Realisierung

Der Kundengruppeneditor ist in Abbildung 4.14 zu sehen. Mit diesem Editor kann eine neue Kundengruppe erstellt oder eine alte bearbeitet werden. Bei den Feldern für den Namen und die Beschreibung handelt es sich um normale Texteingaben, wohingegen die Kunden wieder per Drag-and-Drop hinzugefügt werden können.

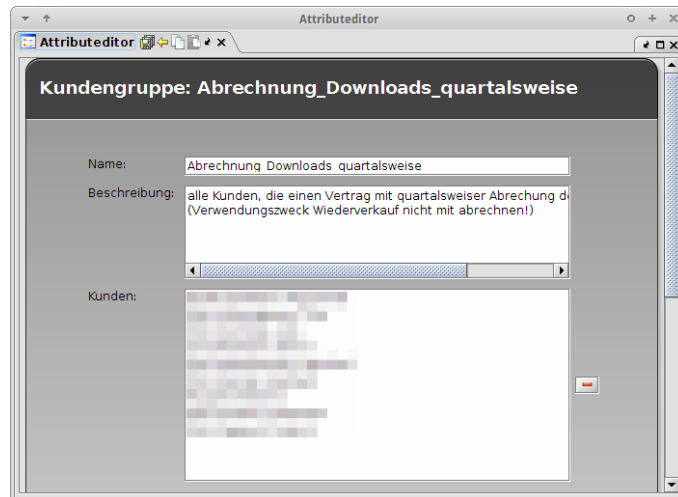


Abbildung 4.14: Kundengruppeneditor

### E-Mail an Kundengruppe erstellen

Wird im Katalog das Kontextmenü einer Kundengruppe geöffnet, dann erscheinen die zwei zusätzlichen Auswahlmöglichkeiten „E-Mail an Ansprechpartner“ und „E-Mail an Nutzer“. Siehe hierzu auch die Abbildung 4.15 auf der nächsten Seite. Mittels diesen beiden Optionen lässt sich ein Vorlagen-Dialog öffnen, mit dem eine Vorlage ausgewählt werden kann. Wird dieser Dialog bestätigt so wird ein E-Mail-Verfassen-Fenster des, im Betriebssystem, eingestellten Standard-E-Mail-Programms geöffnet. Dieses Fenster enthält die Werte aus der Vorgabe und kann wie gewohnt bearbeitet werden. Je nach Auswahl der Option werden unterschiedliche E-Mail-Adressen eingesetzt. Wurden die Ansprechpartner ausgewählt, so werden nur die Direktkontakte der Kunden innerhalb der Kundengruppe der E-Mail hinzugefügt. Andernfalls werden die Kontakte sämtlicher Kunden-Logins die sich in dieser Gruppe befinden benutzt. Um die restlichen E-Mail-Adressen den jeweiligen Empfängern nicht bekannt zu machen, werden diese als BCC der E-Mail hinzugefügt.

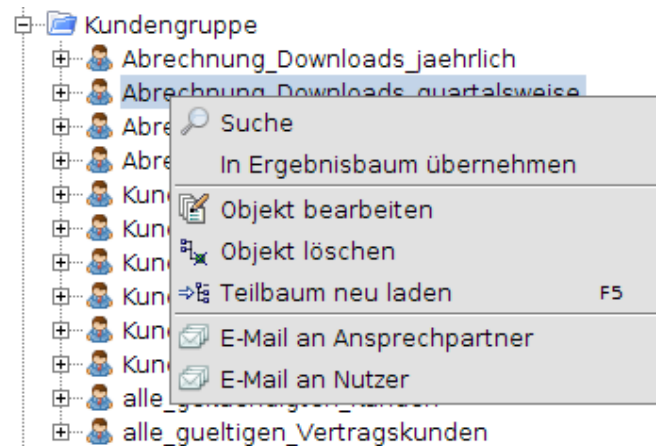


Abbildung 4.15: Kontextmenu einer Kundengruppe

Die Vorlagen die in dem Vorlagen-Dialog angezeigt werden, werden aus der Datei `email_templates.json` bezogen. Eine Vorlage besitzt die folgenden vier Felder:

**Name** Der Name der Vorlage. Dieser wird in dem Auswahlfeld des Vorlagen-Dialogs angezeigt.

**Betreff** Der Betreff der E-Mail.

**To-Adressat** Dieser Empfänger wird ebenfalls der E-Mail hinzugefügt und soll als Kontrolle dienen.

**Inhalt** Der eigentliche Inhalt der E-Mail.

Um das E-Mail-Verfassen-Fenster zu öffnen, wurde die Klasse `EMailComposer` implementiert. Ein Objekt dieser Klasse nimmt die benötigten Informationen entgegen und erstellt eine URI nach dem `mailto` Schema [HMZ98]. Diese URI kann der Java-Methode `Desktop.getDesktop().mail()` [Ora] übergeben werden, welches dann das E-Mail-Verfassen-Fenster, mit den entsprechenden Vorgaben, öffnet.

### 4.2.9 Berichte

**TODO:** Noch nicht fertig. Spezifikation der Berichte fehlt noch teils. Die Berichte wurden mit dem Werkzeug iReport erstellt. Diese Berichte können aus Java



heraus dynamisch generiert werden, so dass ihnen beliebige Parameter mitgegeben werden. Weiterhin kann diesen Berichten auch Listen mitgegeben werden und dort verarbeitet werden. So kann jede Zeile der Liste nach einer in iReport angegebenen Vorgabe angezeigt werden.

Beim Erstellen der Berichte in iReport wurde sich an die Vorgaben aus Abschnitt 3.2.6 gehalten. Durch die eher eingeschränkte Funktionalität von iReport konnten die Buchungen nicht als Liste übergeben werden, sondern die benötigten Daten mussten in WuNDa vorgerechnet werden und als Parameter übergeben werden.

### 4.2.10 Buchungen-Knoten im Katalog

Um die Navigation innerhalb des Buchungsthemas zu gewährleisten wurde der Katalog um einen Buchungen-Knoten erweitert. Dessen Aufbau ist, aus Sicht eines Administrators, in Abbildung 4.16 dargestellt. Diese Erweiterung wurde mit Hilfe des ABF durchgeführt. Dies war insbesondere deshalb nützlich, da es auch bei den Unterknoten, die sich direkt unter dem Knoten „Buchungen“ befinden, um dynamische Knoten handelt. Somit konnte im ABF direkt überprüft werden, ob die Query richtig angegeben wurde, da dort die automatisch erstellten Unterknoten als Vorschau angezeigt werden.

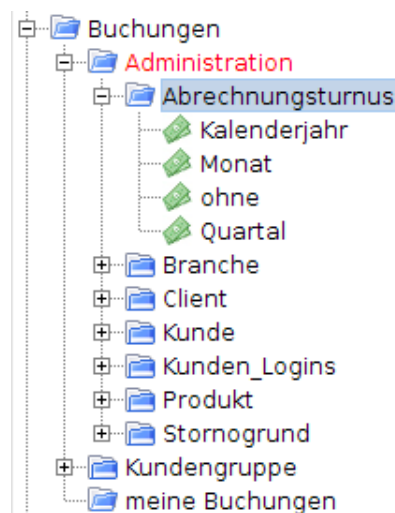


Abbildung 4.16: Aufbau des Katalogs für die Buchungen

Der Unterknoten „Administration“ enthält wiederum Unterknoten mit Objekten, die zur Verwaltung des Buchungsthemas benötigt werden. So kann mittels des Aufrufen des Kontextmenüs auf den verschiedenen Unterknoten ausgewählt werden ob neue Objekte erstellt oder alte Objekte bearbeitet werden sollen. Das gleiche gilt für den Unterknoten „Kundengruppe“, dieser steht nicht unterhalb der „Administration“ da dieser vorallem benutzt wird um die Kunden- und Kundengruppenrenderer aufzurufen. Mit ihm können aber auch die Kundengruppen verwaltet werden. Ein besonderer Fall stellt der Unterknoten „meine Buchungen“ dar, der für die Administratoren leer ist. An dieser Stelle sehen die externen Benutzer, den Kunden dem sie zugeordnet sind und können den entsprechenden Renderer öffnen.

### 4.2.11 Anlegen der Berechtigungen

Die verschiedenen, benötigten Berechtigungen wurden mit Hilfe des ABFs erstellt. In einem ersten Schritt wurde eine neue Benutzergruppe angelegt, danach wurden die Rechte der Benutzergruppen auf die Klassen festgelegt und anschließend wurden noch Aktionsrechte angelegt.

#### Neue Benutzergruppe

Im Abschnitt 3.2.1 wurde beschrieben, dass nur die WuNDa-Administratoren und verschiedene Beamte auf das Buchungen-Thema zugreifen dürfen. Für diese Beamte wurde eine Benutzergruppe „Billing Intern“ angelegt, die danach die Rechte erhalten hat, so dass die Benutzer die Buchungen verwalten können.

Für die Kunden gibt es keine eigene Benutzergruppe, sondern sie sind nach ihrer Art in verschiedene Benutzergruppen eingeteilt. Diese Benutzergruppen sind „Unternehmen“, „Behörde“, „ÖbVI - gesetzlich“ und „ÖbVI - kommunal“. Diese haben im Nachhinein die Rechte erhalten, so dass sie ihren eigenen Kundenrenderer, wie im Abschnitt 3.2.3 beschrieben, betrachten können.

## Rechte der Klassen

Sämtliche Klassen mit dem Präfix „Billing\_“ haben das Standardverhalten „Secure“ erhalten. Damit wird sichergestellt, dass niemand auf diese Klassen Zugriff erhält, dem die Rechte nicht explizit zugewiesen wurden. In einem nächsten Schritt haben die Administratoren und die Benutzer der Benutzergruppe „Billing Intern“ auf diesen Klassen Lese- und Schreibrechte erhalten. Anschließend haben die verschiedenen Benutzergruppen der Kunden ihre benötigten Rechte erhalten. Eine Übersicht zu den vergebenen Rechten kann der Tabelle 4.1 entnommen werden. In dieser Tabelle sind die Administratoren und die Billing Intern Gruppe, der Vollständigkeit halber, ebenfalls aufgelistet.

**Tabelle 4.1: Vergebene Rechte an die Klassen.**

	<b>Administratoren und Billing Intern</b>	<b>Kunden</b>
Abrechnungsturnus	r/w	r
Billing	r/w	r/w
Branche	r/w	r
Client	r/w	r
Kunde	r/w	r
Kunde_Kundengruppe_Array	r/w	-
Kunde_Produkt_Array	r/w	r
Kunden_Logins	r/w	r
Kundengruppe	r/w	-
Produkt	r/w	r
Stornogrund	r/w	r

r = Leserecht, w = Schreibrecht, - = keine Rechte

## Rechte des Kataloges

Damit die angelegten Katalog-Knoten nur von den gewünschten Benutzergruppen ausgewählt und benutzt werden können, mussten für die Knoten ebenfalls Rechte gesetzt werden. Eine Übersicht zu den vergebenen Rechten kann der Tabelle 4.2 auf der nächsten Seite entnommen werden. In dieser Tabelle wird das gesetzte

#### 4 Realisierung

Standardverhalten der Knoten ebenfalls aufgelistet. Das Standardverhalten „Secure“ funktioniert auf die gleiche Art und Weise wie dies auch bei den Klassen der Fall ist. Einem Knoten kann zusätzlich eine Klasse zugewiesen werden, auf die er sich bezieht. Daraufhin kann als Standardverhalten festgelegt werden, dass der Knoten die Rechte seiner Klasse übernimmt. In diesen Fällen wurde in der Tabelle Klammern um die Rechte gesetzt, um zu verdeutlichen dass diese übernommen wurden.

**Tabelle 4.2: Vergebene Rechte an die Katalogknoten.**

	<b>Standardverhalten</b>	<b>Administratoren und Billing Intern</b>	<b>Kunden</b>
Buchungen	Secure	r	r
Administration	Secure	r	-
Kundengruppe	Verhalten von „Kundengruppen“	(r/w)	(-)
meine Buchungen	Verhalten von „Kunde“	(r/w)	(r)

r = Leserecht, w = Schreibrecht, - = keine Rechte, () = übernommene Rechte  
Hierbei ist zu bemerken, dass die Rechte von den Kunden für die Klasse „Kundengruppe“ nicht auf lesend gesetzt werden darf. Wäre dies der Fall so könnten sämtliche Kunden die Kundenrenderer der anderen Kunden öffnen und somit Einsicht in deren Buchungen erhalten.

#### **Einschränkungen bezüglich der Rechnungsanlage und der Buchungen-Suche**

Nach dem Setzen der Rechte haben die externe Benutzer zu viele Berechtigungen. Nämlich das Erstellen der Rechnungsanlage und das Öffnen der Buchungen-Suche. Die zusätzlich benötigten Einschränkungen werden mit Hilfe der folgenden zwei Aktionsrechten realisiert:

- custom.billing.reports
- custom.billing.search

## 4 Realisierung

Diesen Aktionsrechte wurden die Benutzergruppen „Administratoren“ und „Billing Intern“ als Attribute hinzugefügt.

Die Einschränkungen bezüglich der Rechnungsanlage kann wie folgt realisiert werden. Im laufenden Programm wird im Kundenrenderer abgefragt, ob der aktuell angemeldete Benutzer ein Besitzer des Aktionsrechtes „custom.billing.reports“ ist. Je nach Resultat dieser Abfrage können somit die benötigten GUI-Elemente an- oder ausgeschaltet werden.

Für die Anzeige der Suche wurde eine Cids-Funktionalität verwendet, die das Interface `ActionTagProtected` liefert. Um diese Funktionalität umzusetzen muss die Klasse `BillingWindowSearch` dieses Interface implementieren. Dadurch muss in `BillingWindowSearch` die Methode `checkActionTag()` realisiert sein, die ein `boolean` zurückgibt. Somit fragt diese Methode ab, ob der aktuell angemeldete Benutzer das Aktionsrecht „custom.billing.search“ besitzt, um dieses Ergebnis daraufhin zurück zu geben. Cids-intern wird diese Methode aufgerufen und die Suche wird entsprechend den Berechtigungen im Such-Menü angezeigt oder nicht.

### **Ansicht des eigenen Kundenrenderers**

Die letzte Einschränkung die umgesetzt werden musste war, dass ein externer Benutzer nur den eigenen Kundenrenderer öffnen kann.

In einem ersten Schritt wurde für jeden Kunden ein eigenes Aktionsrecht angelegt. Der Namensaufbau eines solchen Aktionsrechtes ist: `custom.billing.tree.<interner_Name_des_Kunden>`, wobei `<interner_Name_des_Kunden>` durch den tatsächlichen internen Namen eines Kunden ersetzt wird. Die Attribute dieses Aktionsrecht sind die Benutzer die zu einem Kunden gehören. Wobei ein Benutzer für jede Benutzergruppe, in der er sich befindet, eingetragen werden muss. Um dies zu verdeutlichen wird ein konkretes Beispiel angeführt: der Kunde ÖbVI Olivier hat den internen Namen „olivier“. Weiterhin hat er einen Login-Name „oebvi\_olivier“ der sich in den Benutzergruppen „ÖbVI - gesetzlich“ und „ÖbVI - kommunal“ befindet. Sein Aktionsrecht muss demnach folgenden Aufbau haben, wobei die Unterelemente dessen Attribute darstellen (Login-Name – Benutzergruppe):

## 4 Realisierung

- custom.billing.tree.olivier
  - oebvi\_olivier – „ÖbVI - gesetzlich“
  - oebvi\_olivier – „ÖbVI - kommunal“

In einem nächsten Schritt wurde die SQL-Query des dynamischen Knotens „meine Buchungen“ angepasst. Diese Query beinhaltet in der Regel Parameter, die zur Erstellung eines Unterknotens benötigt werden, wie etwa die ID oder den anzuzeigenden Namen. Zusätzlich zu den üblichen Parameter wurde hier das Parameter „additionaltreepermissiontag“ hinzugefügt. Dessen Wert besteht aus dem String „custom.billing.tree.“ konkateniert mit dem jeweiligen internen Namen des Kunden. Wichtig zu erwähnen ist, dass somit der Aktionsrechtsname und der Wert des Parameters für einen Kunden gleich ist.

Im letzten Schritt musste die Klasse `VirtualTree` erweitert werden. Sie wertet für jeden zu erstellenden Kundenknoten das Parameter „additionaltreepermissiontag“ aus und überprüft, ob der aktuell angemeldete Benutzer ein Besitzer des Aktionsrechtes mit dem gleichen Namen ist. Nur in dem Fall bei dem dies zutrifft wird der Knoten dem Katalog hinzugefügt. Aus diesem Grund muss der oben genannte Aufbau des Aktionsrechtsnamens eingehalten werden, da sonst die Überprüfung im `VirtualTree` immer fehlschlägt und der Kundenknoten nicht angezeigt wird. Weiterhin ist somit sichergestellt, dass ein Kundenknoten eines anderen Kunden nicht angezeigt wird, weil der aktuell angemeldete Benutzer nicht Besitzer dessen Aktionsrechtes ist.

Auf diese Weise sieht ein Kunde jeweils nur seinen eigenen Knoten und kann hierüber nur seinen eigenen Kundenrenderer öffnen.

### 4.3 Vergleich Realisierung und Spezifikation

- Durch diese Realisierung ist es möglich das zu tun was in den Anforderungen beschrieben wurde
- D.h. durch die Rationalisierung kann durchgeführt werden

## 5 Zukünftige Unterstützung der Workflows durch das System

- Oben wurde beschrieben warum die ent. Benutzer Daten anfordern müssen, was der Grund der gewünschten Änderungen war.
- Nun wird ein Schritt weiter gedacht

### 5.1 Überprüfung der Workflows

- Überprüfung dieser Workflows auf andere potentielle automatisierbare Teile
- Dabei Konzentration auf:
  - ÖbVI
  - Polizei
  - Sparkasse
- Durch Aufdecken solcher auto. Teile können Folgeaufträge generiert werden, falls diese relevant sind. (Geld sparen)
- Beim Aufdecken wird z.b. geachtet auf
  - Beziehen der Daten
  - Verarbeiten der Daten
  - Rücksenden der Daten
- Kriterien sind u.a. Komfort, Zeitaufwand

## *5 Zukünftige Unterstützung der Workflows durch das System*

- Betrachten dieser gefundenen auto. Teile

### **5.1.1 Öffentlich bestellter Vermessungsingenieur**

### **5.1.2 Polizei**

### **5.1.3 Sparkasse**

## **5.2 Lösungsansätze zur Unterstützung**

- Beschreiben dieser gefundenen Lösungsansätze, für jeden relevanten auto. Teil

## **5.3 Bewertung der Lösungsansätze**

- Einschätzen dieser Lösungsansätze auf Machbarkeit und Rentabilität
- Favorisieren der Lösungsansätze



# Abkürzungen

**ABF** Administrators' Best Friend

**AdV** Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland

**ALB** Automatisierte Liegenschaftsbuch

**ALK** Automatisierte Liegenschaftskarte

**ALKIS** Amtliche Liegenschaftskatasterinformationssystem

**ALWIS** Alle Liegenschaftskataster-WuNDa-Informationssysteme

**AP** Aufnahmepunkt

**CAD** computer-aided design

**CSV** Comma-separated values

**DGK** Digitale Grundkarte

**DXF** Drawing Interchange Format

**GIS** Geoinformationssystem

**GML** Geography Markup Language

**JSON** JavaScript Object Notation

**NAS** Normbasierte Austauschchnittstelle

**NivP** Nivellement Punkt

**NRW** Nordrhein-Westfalen

**ÖbVI** Öffentlich bestellter Vermessungsingenieur

**PNÜ** Punktnummerierungsübersicht

**SAPOS** Satellitenpositionierungsdienst der deutschen Landesvermessung

**TP** Trigonometrische Punkte

**VermWertGebO** Vermessungs- und Wertermittlungsgebührenordnung

## *5 Zukünftige Unterstützung der Workflows durch das System*

**VP-Liste** Vermessungspunktliste

**WFS** Web Feature Service

**WuNDa** Wuppertaler Navigations- und Datenmanagementsystem

**XML** Extensible Markup Language