

PFLICHTENHEFT

OctaVIA AG – Pflichtenheft	
<Projektname>	OctaVIA AG Schulungsprojekt
<Projektnummer>	

Hinweis: Bitte Beschriften Sie Tabellen und Abbildungen um diese anschließend in einem Verzeichnis aufzulisten.

Änderungshistorie			
Version	Datum	Änderung	Name
0.1	29.07.13	Inhalt	Andreas Dinkel
0.2	30.07.13	Inhalt	Andreas Dinkel
0.3	31.07.13	Formatierung, Ergänzung	Andreas Dinkel
0.3	01.08.13	Ergänzung	Andreas Dinkel
0.4	02.08.2013	Korrektur	Daniel Kawaletz

Impressum

Copyright © <DocProperty Status Date> by OctaVIA AG
<DocProperty Author>, Kassel

Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

PFLICHTENHEFT

Herausgeber	Octavia AG		
Dateiname	OctaVIA_Pflichtenheft_ver_04		
Dokumentnummer			
Dokumentenbezeichnung			
Version	04		
Stand	31.07.2013		
Status	In Bearbeitung		
Autor	Daniel Kawaletz, Andreas Dinkel		
Inhaltlich geprüft von	Daniel Kawaletz		
Freigegeben von	"Hier Name, EMail eingeben" "Hier Ort, Datum eingeben"		
Ansprechpartner	Andreas Dinkel	+49 561 31000-546	andreas.dinkel@octavia.de
Kurzinfo	"OctaVIA AG internes Schulungsprojekt. Pflichtenheft."		

PFLICHTENHEFT

Inhaltsverzeichnis

1	Ausgangssituation.....	6
2	Aufgabenstellung	6
2.1	Wunschkriterien	7
3	Implementierungsentwurf	7
3.1	Gesamtsystem	7
3.2	Implementierung von Komponente <Betriebsmittel-Management>.....	8
3.3	Implementierung von Komponente <Kunde>	8
3.4	Implementierung von Komponente <Betriebsmittel>.....	8
3.5	Implementierung von Komponente <Verträge>.....	8
3.6	Klassendiagramm.....	9
3.6.1	Beschreibung der Eigenschaften der Klasse: Kunde.....	9
3.6.2	Beschreibung der Funktionalität der Klasse: Kunde.....	10
3.6.3	Beschreibung der Eigenschaften der Klasse: Vertrag	10
3.6.4	Beschreibung der Funktionalität der Klasse: Vertrag	10
3.6.5	Beschreibung der Eigenschaften der Klasse: Betriebsmittel	11
3.6.6	Beschreibung der Funktionalität der Klasse: Betriebsmittel	11
3.6.7	Beschreibung der Funktionalität der Klasse: Controller	11
4	Datenmodell.....	12
4.1	Diagramm	12
5	Funktionale (Detail-) Anforderungen	13
5.1	Anforderungspaket Geschäftsfälle	13
5.1.1	Detailanforderung: Geschäftsfall 1.....	13
5.1.2	Detailanforderung: Geschäftsfall 2.....	16
5.1.3	Detailanforderung: Geschäftsfall 3.....	18
5.1.4	Detailanforderung: Geschäftsfall 4.....	20
5.1.5	Detailanforderung: Geschäftsfall 5.....	22
5.1.6	Detailanforderung: Geschäftsfall 6.....	23
6	Messages.....	24
7	GUI	26
7.1	Startbildschirm	26
7.2	Verträge anzeigen.....	27
7.2.1	Verträge hinzufügen.....	28
7.3	Kunden anzeigen	29
7.3.1	Kunden hinzufügen	30
7.3.2	Kunden bearbeiten.....	31
7.4	Betriebsmittel anzeigen.....	32
7.4.1	Betriebsmittel hinzufügen	33
7.4.2	Betriebsmittel bearbeiten	34
8	Suchhilfen.....	34
9	Technische Architektur	36
10	Nichtfunktionale Anforderungen	36
10.1	Zuverlässigkeit	36
10.2	Benutzerfreundlichkeit	36
11	Beteiligte	37
11.1.1	Kunde/Auftraggeber/Interne Beteiligte.....	37
11.1.2	Anwender.....	37
12	Umfang der Anforderung.....	37
13	Offene Punkte	38
14	Testanforderungen	38
15	Abkürzungsverzeichnis.....	38
16	Quellenverzeichnis	38

PFLICHTENHEFT

Abbildungsverzeichnis

Abbildung 1: Komponentendiagramm - Betriebsmittel-Management	7
Abbildung 2: Klassendiagramm	9
Abbildung 3: Datenmodell	12
Abbildung 4: Flussdiagramm - Kunden anzeigen	13
Abbildung 5: Flussdiagramm - Betriebsmittel anzeigen	14
Abbildung 6: Flussdiagramm - Vertrag anzeigen	15
Abbildung 7: Flussdiagramm - Betriebsmittel anlegen	16
Abbildung 8: Flussdiagramm - Kunden anlegen	17
Abbildung 9: Flussdiagramm - Betriebsmittel bearbeiten	18
Abbildung 10: Flussdiagramm - Kunden bearbeiten	19
Abbildung 11: Flussdiagramm - Betriebsmittel löschen	20
Abbildung 12: Flussdiagramm - Kunden löschen	21
Abbildung 13: Flussdiagramm - Vertrag anlegen	22
Abbildung 14: Flussdiagramm - Kosten anzeigen	23
Abbildung 15: Skizze - „Startbildschirm“ der Anwendung	26
Abbildung 16: Skizze - Ansicht "Verträge anzeigen"	27
Abbildung 17: Skizze – Ansicht „Verträge hinzufügen“	28
Abbildung 18: Skizze - Ansicht "Kunden anzeigen"	29
Abbildung 19: Skizze - Ansicht "Kunden hinzufügen"	30
Abbildung 20: Skizze - Ansicht "Kunden bearbeiten"	31
Abbildung 21: Skizze - Ansicht "Betriebsmittel anzeigen"	32
Abbildung 22: Skizze - Ansicht "Betriebsmittel hinzufügen"	33
Abbildung 23: Skizze - Ansicht "Betriebsmittel bearbeiten"	34
Abbildung 24: Systemarchitektur	36

PFLICHTENHEFT

Tabellenverzeichnis

Tabelle 1: Eigenschaften - Klasse "Kunde"	10
Tabelle 2: Funktionalität - Klasse "Kunde"	10
Tabelle 3: Eigenschaften - Klasse "Vertrag"	10
Tabelle 4: Funktionalität - Klasse "Vertrag"	10
Tabelle 5: Eigenschaften - Klasse "Betriebsmittel"	11
Tabelle 6: Funktionalität - Klasse "Betriebsmittel"	11
Tabelle 7: Funktionalität - Klasse "Controller"	11
Tabelle 8: Kardinalität der Beziehungen	12
Tabelle 9: Anforderungspaket Geschäftsfälle	13
Tabelle 10: Geschäftsfall 1 – Kunden anzeigen	14
Tabelle 11: Geschäftsfall 1 - Betriebsmittel anzeigen	14
Tabelle 12: Geschäftsfall 1 - Vertrag anzeigen	15
Tabelle 13: Geschäftsfall 2 - Betriebsmittel anlegen	16
Tabelle 14: Geschäftsfall 2 - Kunden anlegen	17
Tabelle 15: Geschäftsfall 3 - Betriebsmittel bearbeiten	18
Tabelle 16: Geschäftsfall 3 - Kunden bearbeiten	19
Tabelle 17: Geschäftsfall 4 - Betriebsmittel löschen	20
Tabelle 18: Geschäftsfall 4 - Kunden löschen	21
Tabelle 19: Geschäftsfall 5 - Vertrag anlegen	23
Tabelle 20: geschäftsfall 6 - Kosten berechnen / Vertrag anzeigen	23
Tabelle 21: Definierte Messages	25
Tabelle 22: Suchhilfen	35
Tabelle 23: Kunde/Auftraggeber/Interne Beteiligte	37
Tabelle 24: Anwender	37
Tabelle 25: Umfang der Anforderung	37
Tabelle 26: Offene Punkte	38
Tabelle 27: Abkürzungsverzeichnis	38

PFLICHTENHEFT

1 Ausgangssituation

Innerhalb der OctaVIA AG werden im Rahmen der anfallenden zu erledigenden Aufgaben und Anforderungen an die Mitarbeiter Betriebsmittel zur Verfügung gestellt. Diese Betriebsmittel sind kostenfrei und je nach Bedarf an die Mitarbeiter vergeben. Das Sekretariat ist hierbei zuständig für die Verteilung, Anschaffung und Wartung der Betriebsmittel. Im Folgenden soll ein Konzept zur Verbesserung des Ausgabe-, Rückgabe- und Überwachungsprozesses der Betriebsmittel vorgestellt werden. Hierbei ist zu berücksichtigen, dass ab der Einführung der Software, Kosten pro Betriebsmittel anfallen die auf den Kunden, in dem Fall der Mitarbeiter, gebucht werden.

Aktuell werden alle Mietobjekte mit Hilfe einer Excel-Tabelle verwaltet. Innerhalb des Excel werden die Objekte mit Datum und Mitarbeiterkürzel versehen, um eine Zuordnung vorzunehmen.

2 Aufgabenstellung

Um die Bereitstellung der Betriebsmittel innerhalb der OctaVIA AG zu verbessern fordert das Sekretariat ein Programm welches die bereitgestellten Objekte verwaltet. Da ein SAP / R3 vorhanden ist, soll das Programm innerhalb des SAP / R3 betrieben werden. Um die Einarbeitung so gering wie möglich zu halten, soll das Programm der ersten Phase mit Hilfe einer Transaktion innerhalb des SAP GUI gestartet werden kann können.

Jeder Kunde soll einen Kundentyp (Mitarbeiter, Privat, Einmalkunde, etc) besitzen, sowie eine entsprechend im System hinterlegte Adresse (Anschrift: Name, Vorname, Anrede, Straße, Hausnummer, PLZ, Ort).

Jeder Kunde kann beliebig viele Verträge abschließen, zu jedem Vertrag ist genau ein Betriebsmittel zugeordnet. Objekte können innerhalb einer Zeitspanne nicht mehrfach vermietet werden.

Betriebsmittel müssen gegebenenfalls gewartet werden. Entsprechend muss ein Intervall hinterlegt werden, nach wie vielen Miettagen ein Betriebsmittel zur Wartung muss. Jedes Betriebsmittel besitzt einen Status.

Die Status sind wie folgt definiert:

- Grün: mindestens noch 15 Tage bis zur nächsten Wartung
- Gelb: 15-1 Tage bis zur nächsten Wartung
- Rot: 0 Tage oder Wartungsdatum überschritten

Wenn Betriebsmittel den Status Rot besitzt, darf keine weitere Vermietung möglich sein. Erst durch das zurücksetzen des Wartungsintervalls, kann die Vermietung wieder möglich gemacht werden.

Innerhalb der Oberfläche soll es möglich geben, die Daten nach Excel zu exportieren, d.h. der Datenbestand, die Entleihungen, die Kunden sowie die verfügbaren Betriebsmittel sind jeweils in eine separate Excel Tabelle exportierbar.

Anzuwendende Technologien:

- GUI: Klassisches Dynpro
- Programmiersprache: ABAP OO
- Techniken: Objektorientierte Klassenstruktur, MVC, Interfaces
- Zielsystem: IED
- Entwicklungspaket: Z_NEWOBV

Beschreibung Aufbau:

Es soll ein Framework entwickelt werden, welches mit Hilfe des MVC Entwurfsmusters an eine beliebige grafische Oberfläche angebunden werden kann. Um die zu entwickelnden Funktionen in der ersten Version

PFLICHTENHEFT

bereitzustellen, soll die Oberfläche als klassische Dynpro umgesetzt werden. Der Controller stellt hierbei alle Funktionen bereit um eine beliebige GUI anzubinden. Durch die Einbindung eines Interfaces in alle Klassen ist zu gewährleisten das zentrale Funktionen (speichern, sperren, entsperren, validieren, wurde geändert, ist neu, usw.) mit eingebunden werden.

2.1 Wunschkriterien

In der Zweiten Phase der Software, soll die Anwendung innerhalb einer Weboberfläche realisiert werden.

Anzuwendende Technologien:

- GUI: Web Dynpro (erst im Release 2.0 umzusetzen)
- Programmiersprache: ABAP OO
- Techniken: Objektorientierte Klassenstruktur, MVC, Interfaces
- Zielsystem: IED
- Entwicklungspaket: Z_NEWOBV

3 Implementierungsentwurf

3.1 Gesamtsystem

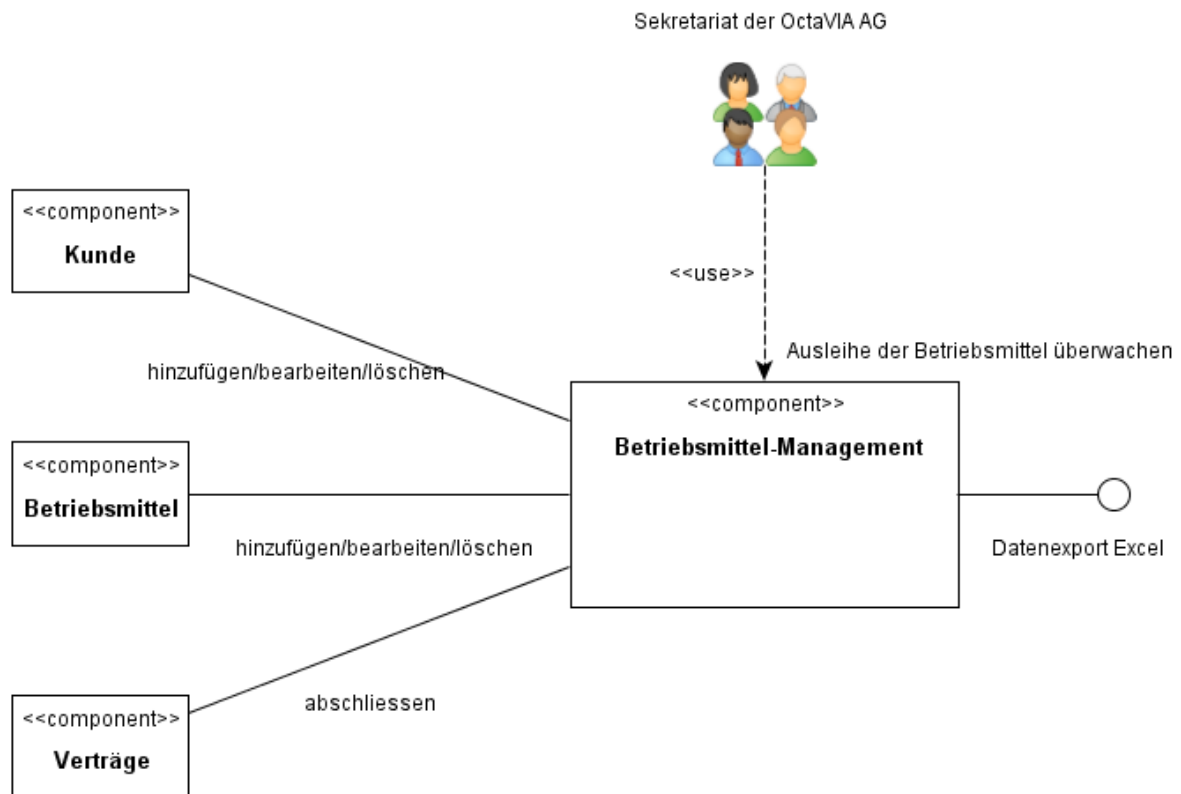


Abbildung 1: Komponentendiagramm - Betriebsmittel-Management

PFLICHTENHEFT

Sekretariat der OctaVIA AG wird beim Benutzen der Betriebsmittel-Management-Komponente den aktuellen Stand der Verträge, Kunden und Betriebsmittel ansehen können. Außerdem wird es möglich sein, über dieselbe Komponente neue Kunden, bzw. Betriebsmittel anlegen, bearbeiten und löschen, sowie beim Ausleihen eines Betriebsmittels durch Kunden ein Vertrag abgeschlossen werden können.

Dazu wird es möglich sein über die Betriebsmittel-Management-Komponente Datenbestände ins Microsoft Excel-Format zu exportieren.

3.2 Implementierung von Komponente <Betriebsmittel-Management>

Die Betriebsmittel-Management-Komponente umfasst in sich die Funktionalität der grafischen Oberfläche „View“, die Klasse „Controller“, Interface und den Datenexport. Die Klasse Controller stellt die Funktionalität für die Interaktion zwischen der grafischen Oberfläche und den Komponenten Kunde, Betriebsmittel und Verträge zur Verfügung.

3.3 Implementierung von Komponente <Kunde>

Zur Komponente <Kunde> gehört die Klasse „Kunde“ und die Datenbanktabelle „Kunde“. Die Klasse „Kunde“ stellt die Funktionalität und die Eigenschaften für den Kunden zur Verfügung. In der Datenbanktabelle „Kunde“ werden die erstellten Kunden gespeichert.

3.4 Implementierung von Komponente <Betriebsmittel>

Zur Komponente <Betriebsmittel> gehört die Klasse „Betriebsmittel“ und die Datenbanktabelle „Betriebsmittel“. Die Klasse „Betriebsmittel“ stellt die Funktionalität und die Eigenschaften für die Betriebsmittel zur Verfügung. In der Datenbanktabelle „Betriebsmittel“ können die erstellten Betriebsmittel gespeichert und gepflegt werden.

3.5 Implementierung von Komponente <Verträge>

Zum Inhalt der Komponente <Verträge> gehören die Klasse und die Datenbanktabelle „Vertrag“. Die Klasse „Vertrag“ stellt die Funktionalität und die Eigenschaften für Vertrag dar. Abgeschlossene Verträge mit dem Kunden werden in der zugehörigen Datenbanktabelle „Vertrag“ gespeichert.

PFLICHTENHEFT

3.6 Klassendiagramm

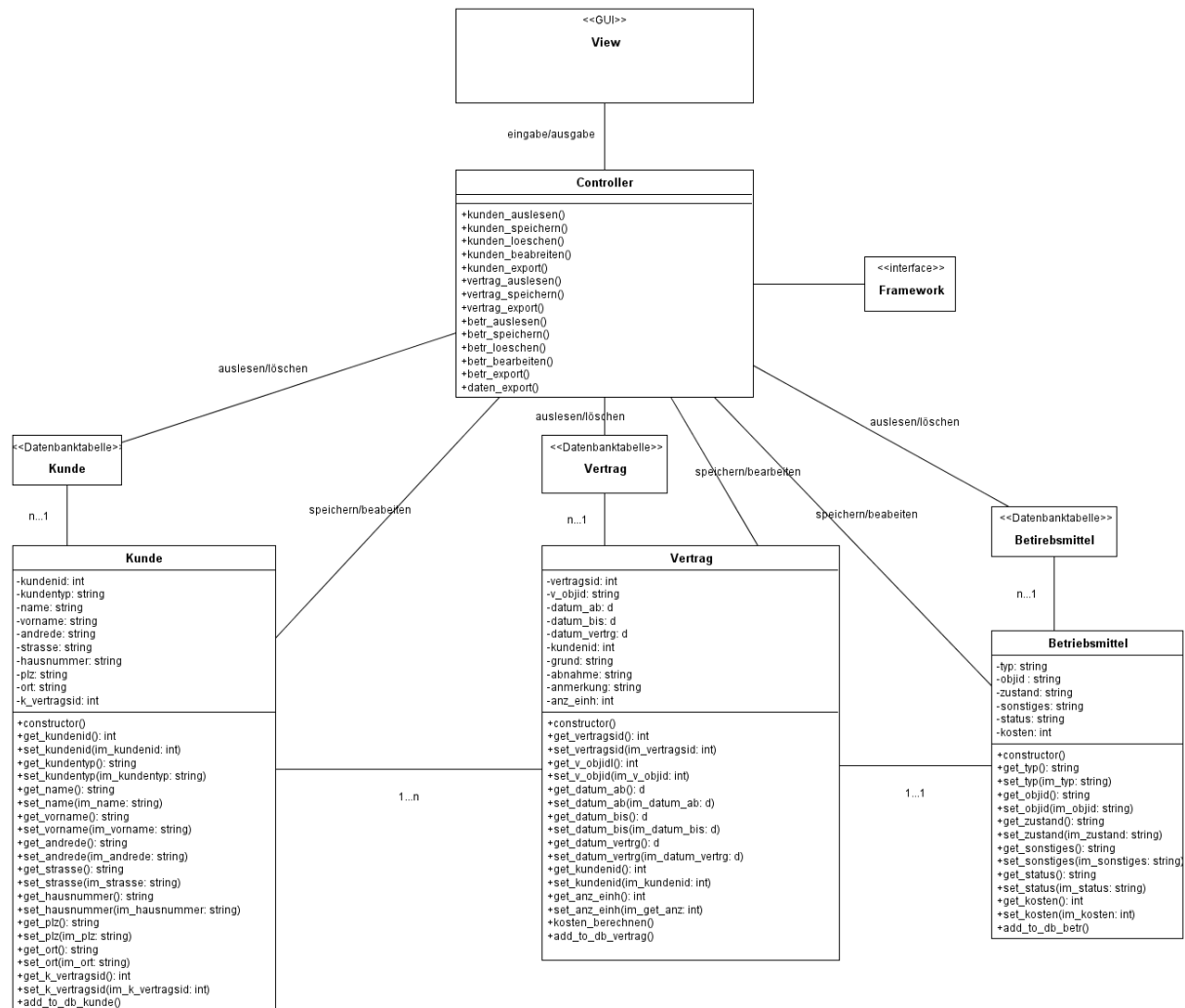


Abbildung 2: Klassendiagramm

3.6.1 Beschreibung der Eigenschaften der Klasse: Kunde

Attributname	Typ	Beschreibung
kundenid	int	Jeder Kunde besitzt eine einzelne ID die den Kunden eindeutig identifiziert. Die ID wird über die Nummernkreise der Kunden zu geordnet.
kundentyp	string	Kann die Werte: Mitarbeiter, Privat und Einmalkunde annehmen
name	string	Nachname des Kunden
vorname	string	Vorname des Kunden
anrede	string	Kann die Werte: Herr, Frau, Dr., Dr. Ing., B. Sc., M.Sc. annehmen.
strasse	string	Straße
hausnummer	string	Hausnummer
plz	string	Postleitzahl

PFLICHTENHEFT

ort	string	Ort
k_vertragsid	int	Tabelle mit VertragsID der abgeschlossenen Verträge des Kunden

Tabelle 1: Eigenschaften - Klasse "Kunde"

3.6.2 Beschreibung der Funktionalität der Klasse: Kunde

Methodenname	Typ	Beschreibung
get_k_vertragsid()	int	Listet die Verträge einzelner Kunden auf.
set_k_vertragsid(im_k_vertragsid)	int	Speichert die VertragsID der abgeschlossenen Verträge des Kunden in einer Tabelle
add_to_db_kunde()		Fügt den neu erstellten Kunden zur DB: Kunde hinzu

Tabelle 2: Funktionalität - Klasse "Kunde"

3.6.3 Beschreibung der Eigenschaften der Klasse: Vertrag

Attributenname	Typ	Beschreibung
vertragsid	int	Jeder Vertrag besitzt eine einzelne ID die den Vertrag eindeutig identifiziert. Die ID wird über die Nummernkreise dem Vertrag zu geordnet.
v_objid	int	Dieses Attribut enthält die dazugehörige ObjID des Betriebsmittels zum Vertrag
datum_ab	d	Datum: Ab wann der Vertrag gültig ist
datum_bis	d	Datum: Bis wann der Vertrag gültig ist
datum_vertrag	d	Datum: Wann der Vertrag abgeschlossen ist
kundenid	int	KundenID des Kunden auf den der Vertrag läuft
grund	string	Grund des Ausleihens
abnahme	string	Name der Person bei der Abnahme des Betriebsmittels
anmerkung	string	Anmerkung zum Vertrag
anz_einh	int	Anzahl der Einheiten für die Betriebsmittel, wonach die Kosten für das Ausleihen für den Kunden verbucht werden. Das Attribut kann Anzahl der Kilometer, Anzahl der Stunden und ohne Gebühr aufnehmen.

Tabelle 3: Eigenschaften - Klasse "Vertrag"

3.6.4 Beschreibung der Funktionalität der Klasse: Vertrag

Methodenname	Typ	Beschreibung
get_v_objid	int	Rückgabe des Betriebsmittels eines Vertrags
set_v_objid(im_v_objid)	int	Setzen einer ObjID des Betriebsmittels in einen Vertrag
kosten_berechnen()		Berechnung der Kosten mit Hilfe der anz_einh.
add_to_db_vertrag()		Fügt neu erstellten Vertrag zur DB: Vertrag hinzu

Tabelle 4: Funktionalität - Klasse "Vertrag"

PFLICHTENHEFT

3.6.5 Beschreibung der Eigenschaften der Klasse: Betriebsmittel

Attributename	Typ	Beschreibung
typ	string	Beschreibt den Typ des Betriebsmittels. Kann die Werte Auto, Buch, Beamer etc. aufnehmen
objid	string	Jedes Betriebsmittel besitzt eine einzelne ID die die Betriebsmittel eindeutig identifiziert. Die ID wird manuell gepflegt.
zustand	string	Zustand des Betriebsmittels
sonstiges	string	Zusätzliche Informationen zum Betriebsmittel
status	string ????	Die Status sind wie folgt definiert: - Grün: mindestens 15 Tage bis zur nächsten Wartung - Gelb: 15-1 Tage bis zur nächsten Wartung - Rot: 0 Tage oder Wartungsdatum überschritten
kosten	int	Kosten des Betriebsmittels pro Einheit

Tabelle 5: Eigenschaften - Klasse "Betriebsmittel"

3.6.6 Beschreibung der Funktionalität der Klasse: Betriebsmittel

Methodenname	Typ	Beschreibung
add_to_db_betr ()		Fügt neu erstelltes Betriebsmittel zur DB: Betriebsmittel hinzu

Tabelle 6: Funktionalität - Klasse "Betriebsmittel"

3.6.7 Beschreibung der Funktionalität der Klasse: Controller

Methodenname	Typ	Beschreibung
kunden_auslesen()		Kunden auslesen aus der DB: Kunde und Darstellen im View
kunden_speichern()		Neue Eingabe des Kunden aus dem View holen und ein Objekt mit dem neuen Kunden erstellen.
kunden_bearbeiten()		Bearbeiten eines bestehenden Kunden
kunden_loeschen()		Löschen eines bestehenden Kunden
kunden_export()		Kundendaten exportieren in Excel
vertrag_auslesen()		Verträge aus der DB: Vertrag auslesen und in der View darstellen
vertrag_speichern()		Eingaben des neuen Vertrags aus der View holen und ein neues Vertragsobjekt mit den gemachten Eingaben erstellen
vertrag_export()		Vertragsdaten exportieren in Excel
betr_auslesen()		Betriebsmittel aus der DB: Betriebsmittel auslesen und in der View darstellen
betr_speichern()		Eingaben des neuen Betriebsmittel aus der View holen und ein neues Betriebsmittelobjekt mit den gemachten Eingaben erstellen
betr_bearbeiten()		Bearbeiten eines bestehenden Betriebsmittels
betr_loeschen()		Löschen eines bestehenden Betriebsmittels
betr_export()		Betriebsmitteldaten exportieren in Excel
daten_export()		Gesamte Daten exportieren in Excel

Tabelle 7: Funktionalität - Klasse "Controller"

PFLICHTENHEFT

4 Datenmodell

Laut der Anforderung kann der Kunde mehrere Betriebsmittel ausleihen. Daraus ergibt sich, dass der Kunde auch mehrere Verträge abschließen kann. Jedem Vertrag kann jeweils höchstens ein Betriebsmittel zugeordnet werden. Aus dieser Überlegung ergibt sich folgendes Datenmodell:

4.1 Diagramm

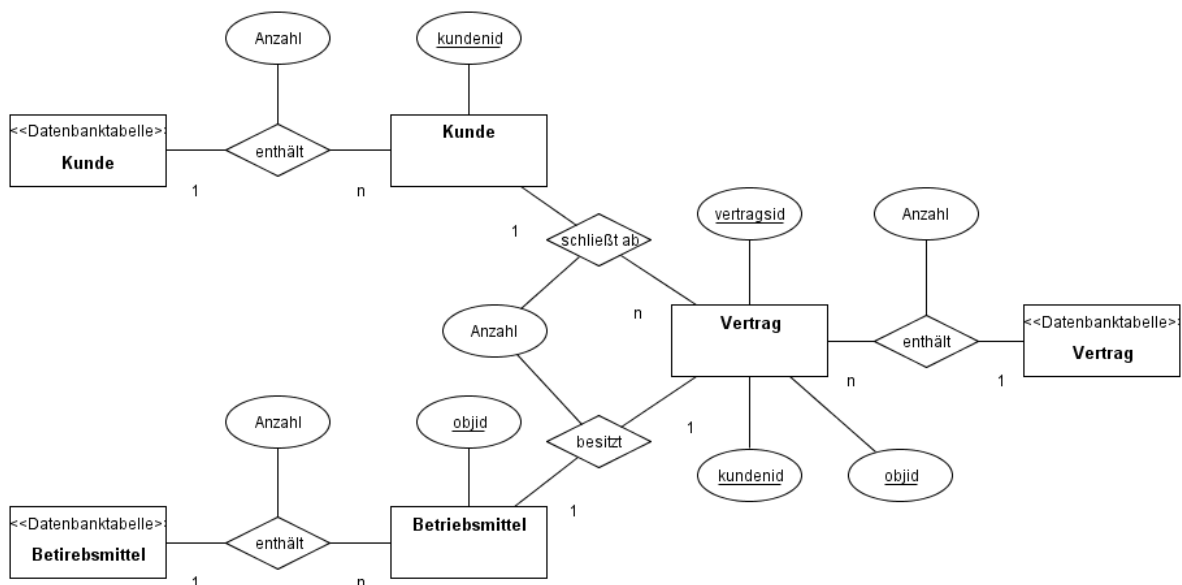


Abbildung 3: Datenmodell

Beziehungen	
Name der Beziehung	Kardinalität
< Kunde-Vertrag >	< 1 ... n >
< DB:Kunde-Kunde >	< 1 ... n >
< Vertrag-Betriebsmittel >	< 1 ... 1 >
< DB:Vertrag- Vertrag >	< 1 ... n >
< DB:Betriebsmittel-Betriebsmittel >	< 1 ... n >

Tabelle 8: Kardinalität der Beziehungen

PFLICHTENHEFT

5 Funktionale (Detail-) Anforderungen

5.1 Anforderungspaket Geschäftsfälle

Bearbeiter	Daniel Kawaletz, Andreas Dinkel
Geschäftsfall 1	Anzeige Kunde / Betriebsmittel / Mietverträge
Geschäftsfall 2	Anlegen neue Betriebsmittel sowie Kunden mit Konten und Adressen
Geschäftsfall 3	Bearbeiten von Betriebsmitteln sowie Kunden mit Konten und Adressen
Geschäftsfall 4	Löschen von Betriebsmitteln sowie Kunden mit Konten und Adressen
Geschäftsfall 5	Anlegen von neuen Mietverträgen mit Start und Enddatum inkl. Mietpreisfestlegung
Geschäftsfall 6	Anzeige der Kosten pro Mietvertrag

Tabelle 9:Anforderungspaket Geschäftsfälle

5.1.1 Detailanforderung: Geschäftsfall 1

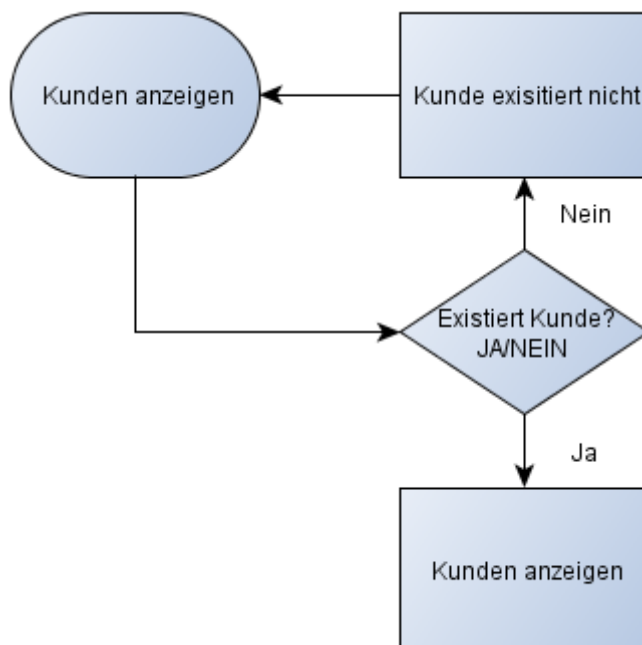


Abbildung 4: Flussdiagramm - Kunden anzeigen

PFLICHTENHEFT

Beteiligte Klasse	Aufgabe		Notwendige Attribute	Notwendige Kommunikationspartner
Controller	Name der Aufgabe	Beschreibung	kundenid:int	Name der Komponente
	kunden_auslesen()	Kunden aus der Datenbanktabelle auslesen		DB: Kunde
				GUI

Tabelle 10: Geschäftsfall 1 – Kunden anzeigen

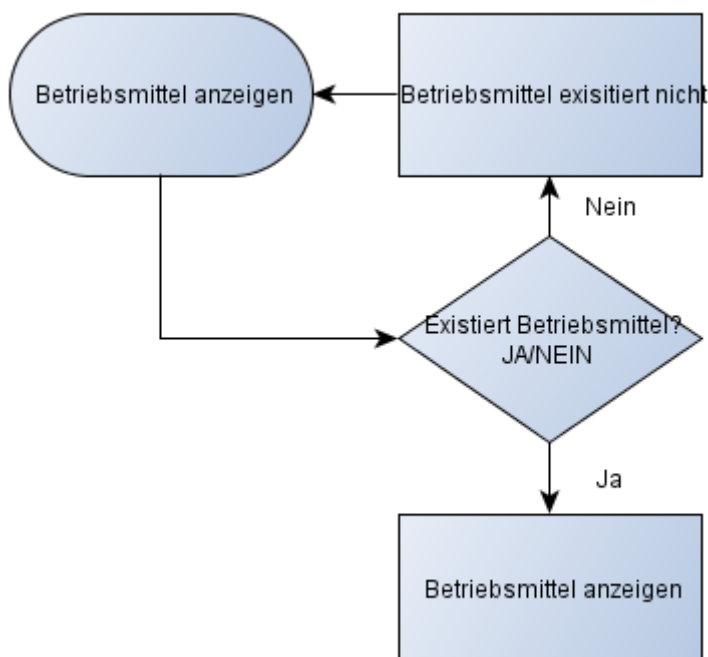


Abbildung 5: Flussdiagramm - Betriebsmittel anzeigen

Beteiligte Klasse	Aufgabe		Notwendige Attribute	Notwendige Kommunikationspartner
Controller	Name der Aufgabe	Beschreibung	objid:int	Name der Komponente
	betr_auslesen()	Betriebsmittel aus der Datenbanktabelle auslesen		DB: Betriebsmittel
				GUI

Tabelle 11: Geschäftsfall 1 - Betriebsmittel anzeigen

PFLICHTENHEFT

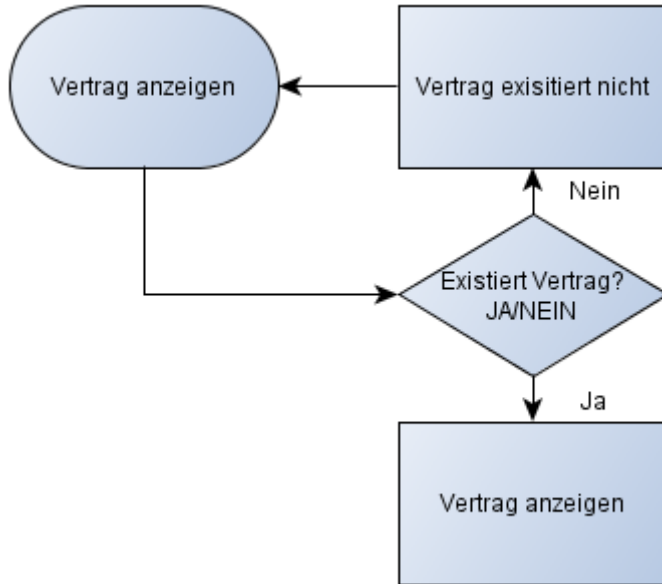


Abbildung 6: Flussdiagramm - Vertrag anzeigen

<i>Beteiligte Klasse</i>	<i>Aufgabe</i>		<i>Notwendige Attribute</i>	<i>Notwendige Kommunikationspartner</i>			
<i>Controller</i>	<i>Name der Aufgabe</i>	<i>Beschreibung</i>	<i>vertragsid:int</i>	<table border="1"> <thead> <tr> <th><i>Name der Komponente</i></th> </tr> </thead> <tbody> <tr> <td><i>DB: Vertrag</i></td> </tr> <tr> <td><i>GUI</i></td> </tr> </tbody> </table>	<i>Name der Komponente</i>	<i>DB: Vertrag</i>	<i>GUI</i>
	<i>Name der Komponente</i>						
<i>DB: Vertrag</i>							
<i>GUI</i>							
<i>vertrag_auslesen()</i>	<i>Vertrag aus der Datenbanktabelle auslesen</i>						

Tabelle 12: Geschäftsfall 1 - Vertrag anzeigen

PFLICHTENHEFT

5.1.2 Detailanforderung: Geschäftsfall 2

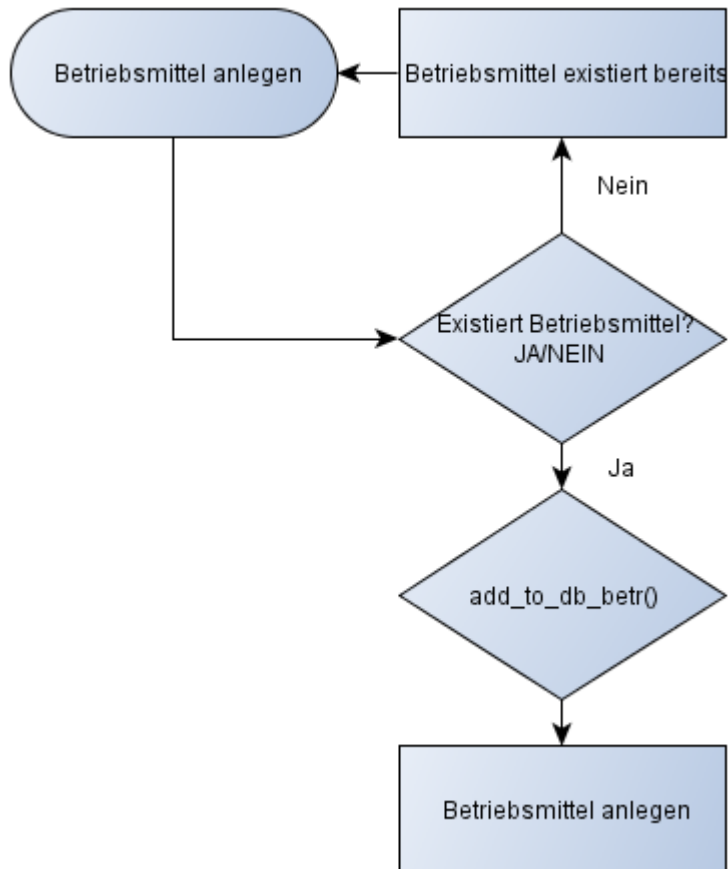


Abbildung 7: Flussdiagramm - Betriebsmittel anlegen

Beteiligte Klasse	Aufgabe		Notwendige Attribute	Notwendige Kommunikationspartner			
	Name der Aufgabe	Beschreibung					
Controller; Betriebsmittel	betr_speichern(), add_to_db_betr()	Betriebsmittel-Objekt erstellen und der Datenbanktabelle hinzufügen	typ:string objid:string zustand:string sonstiges:string status:string kosten:int	<table border="1"> <thead> <tr> <th>Name der Komponente</th> </tr> </thead> <tbody> <tr> <td>DB: Betriebsmittel</td> </tr> <tr> <td>GUI</td> </tr> </tbody> </table>	Name der Komponente	DB: Betriebsmittel	GUI
Name der Komponente							
DB: Betriebsmittel							
GUI							

Tabelle 13: Geschäftsfall 2 - Betriebsmittel anlegen

PFLICHTENHEFT

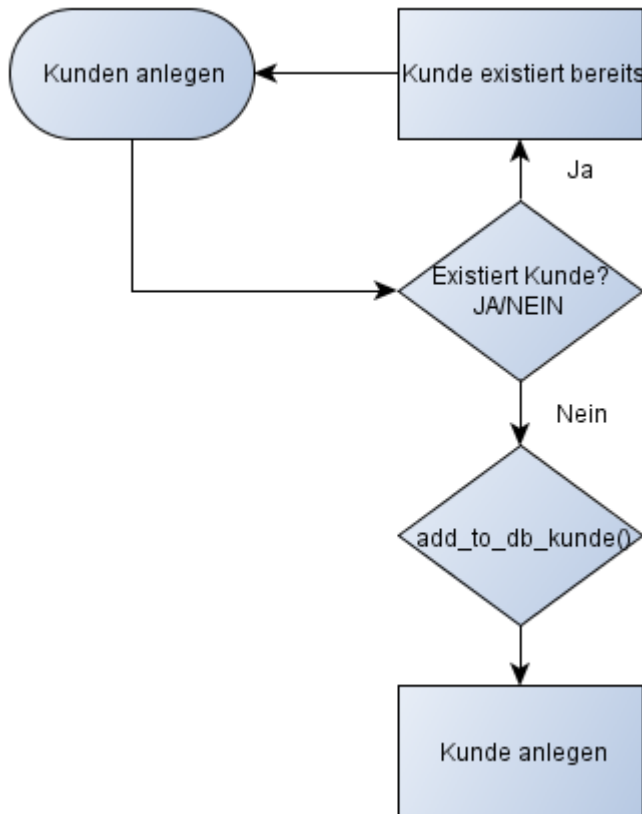


Abbildung 8: Flussdiagramm - Kunden anlegen

Beteiligte Klasse	Aufgabe	Notwendige Attribute	Notwendige Kommunikationspartner							
Controller; Kunde	<table border="1"> <thead> <tr> <th>Name der Aufgabe</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>kunden_speichern(), add_to_db_kunde()</td> <td>Kunde-Objekt erstellen und der Datenbanktabelle hinzufügen</td> </tr> </tbody> </table>	Name der Aufgabe	Beschreibung	kunden_speichern(), add_to_db_kunde()	Kunde-Objekt erstellen und der Datenbanktabelle hinzufügen	kundenid:int kundentyp:string name:string vorname:string anrede:string strasse:int hausnummer:string plz:string ort:string k_vertrags:int	<table border="1"> <thead> <tr> <th>Name der Komponente</th> </tr> </thead> <tbody> <tr> <td>DB: Kunde</td> </tr> <tr> <td>GUI</td> </tr> </tbody> </table>	Name der Komponente	DB: Kunde	GUI
Name der Aufgabe	Beschreibung									
kunden_speichern(), add_to_db_kunde()	Kunde-Objekt erstellen und der Datenbanktabelle hinzufügen									
Name der Komponente										
DB: Kunde										
GUI										

Tabelle 14: Geschäftsfall 2 - Kunden anlegen

PFLICHTENHEFT

5.1.3 Detailanforderung: Geschäftsfall 3

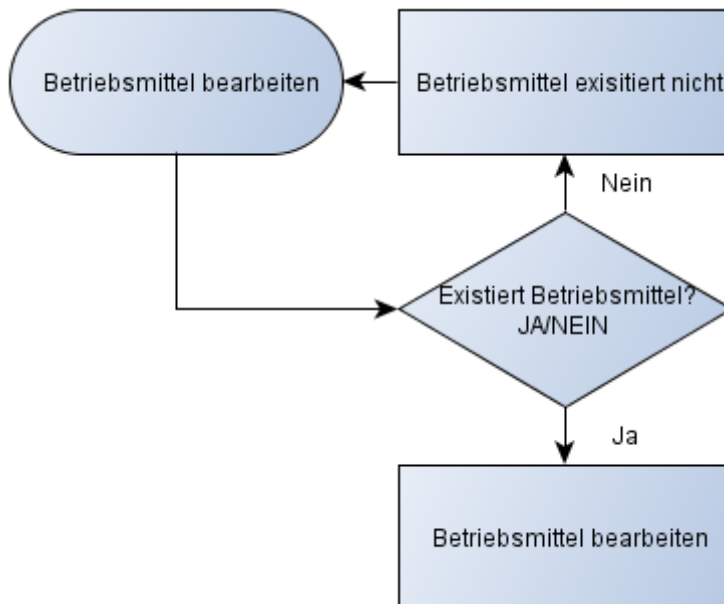


Abbildung 9: Flussdiagramm - Betriebsmittel bearbeiten

Beteiligte Klasse	Aufgabe		Notwendige Attribute	Notwendige Kommunikationspartner			
	Name der Aufgabe	Beschreibung					
Controller; Betriebsmittel	betr_bearbeiten()	Betriebsmittel-Objekt	typ:string zustand:string sonstiges:string status:string kosten:int	<table border="1"> <thead> <tr> <th>Name der Komponente</th> </tr> </thead> <tbody> <tr> <td>DB: Betriebsmittel</td> </tr> <tr> <td>GUI</td> </tr> </tbody> </table>	Name der Komponente	DB: Betriebsmittel	GUI
		Name der Komponente					
DB: Betriebsmittel							
GUI							
Eigenschaften ändern und der Datenbanktabelle hinzufügen							

Tabelle 15: Geschäftsfall 3 - Betriebsmittel bearbeiten

PFLICHTENHEFT

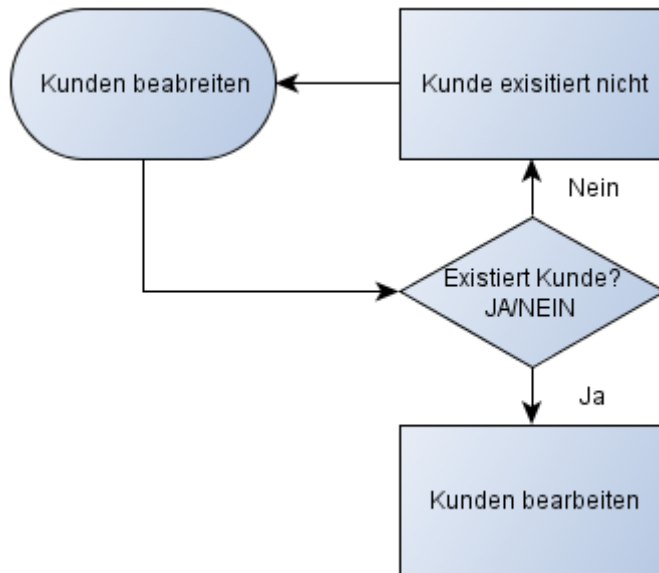


Abbildung 10: Flussdiagramm - Kunden bearbeiten

Beteiligte Klasse	Aufgabe		Notwendige Attribute	Notwendige Kommunikationspartner			
Controller; Kunde	Name der Aufgabe	Beschreibung	kundentyp:string name:string vorname:string anrede:string strasse:int hausnummer:string plz:string ort:string k_vertrags:int	<table border="1"> <thead> <tr> <th>Name der Komponente</th> </tr> </thead> <tbody> <tr> <td>DB: Kunde</td> </tr> <tr> <td>GUI</td> </tr> </tbody> </table>	Name der Komponente	DB: Kunde	GUI
	Name der Komponente						
DB: Kunde							
GUI							
kunden_bearbeiten()	Kunde-Objekt Eigenschaften verändern und der Datenbanktable le hinzufügen						

Tabelle 16: Geschäftsfall 3 - Kunden bearbeiten

PFLICHTENHEFT

5.1.4 Detailanforderung: Geschäftsfall 4

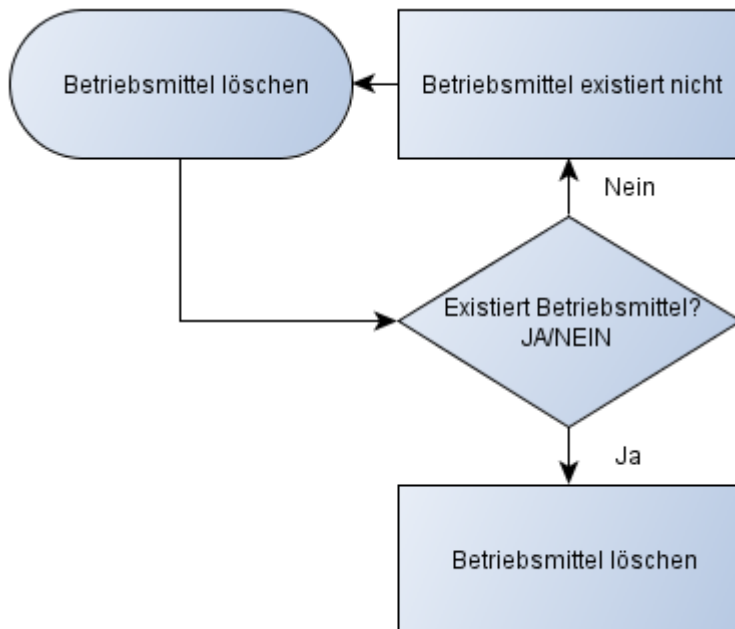


Abbildung 11: Flussdiagramm - Betriebsmittel löschen

Beteiligte Klasse	Aufgabe		Notwendige Attribute	Notwendige Kommunikationspartner
	Name der Aufgabe	Beschreibung		
Controller	betr_loeschen ()	Betriebsmittel aus der Datenbanktabelle löschen	objid:int	Name der Komponente DB: Betriebsmittel GUI

Tabelle 17: Geschäftsfall 4 - Betriebsmittel löschen

PFLICHTENHEFT

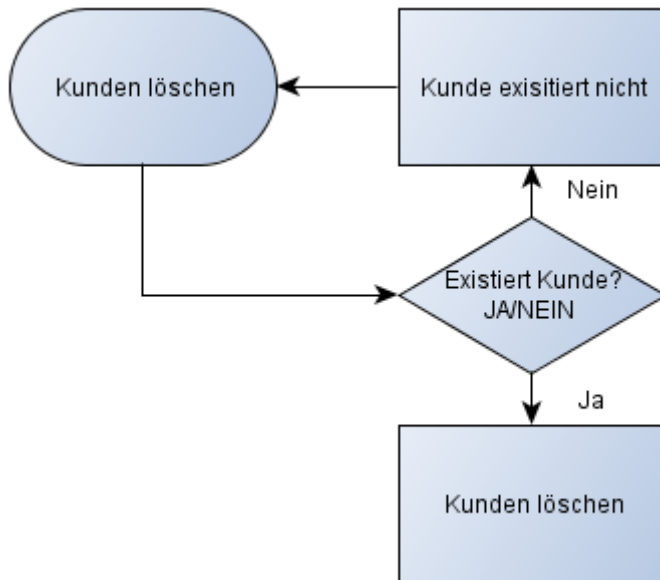


Abbildung 12: Flussdiagramm - Kunden löschen

Beteiligte Klasse	Aufgabe		Notwendige Attribute	Notwendige Kommunikationspartner	
	Name der Aufgabe	Beschreibung		Name der Komponente	
Controller	kunden_loeschen()	Kunden aus der Datenbanktabelle löschen	kundenid:int	DB: Kunde	GUI

Tabelle 18: Geschäftsfall 4 - Kunden löschen

PFLICHTENHEFT

5.1.5 Detailanforderung: Geschäftsfall 5

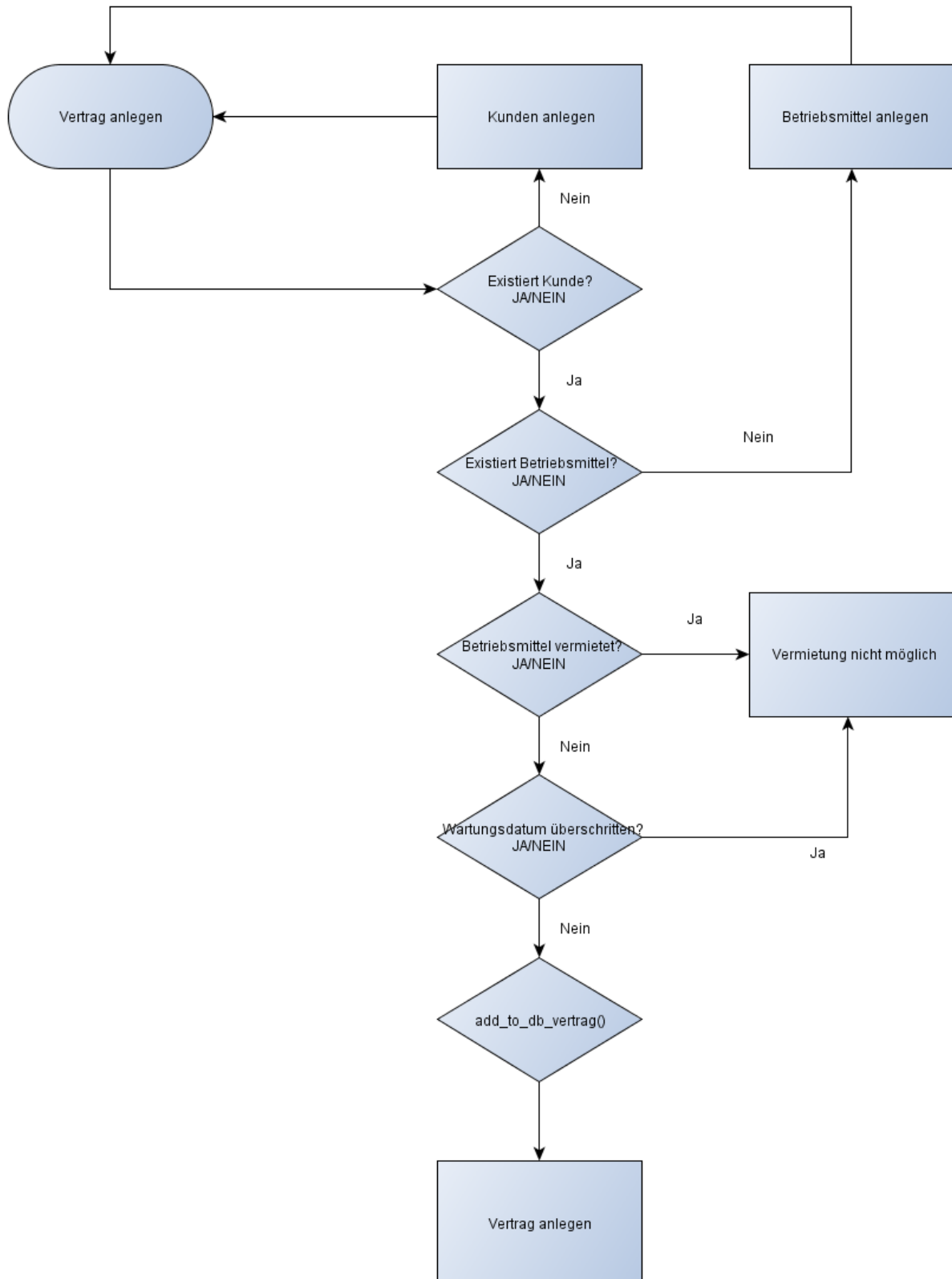


Abbildung 13: Flussdiagramm - Vertrag anlegen

PFLICHTENHEFT

Beteiligte Klasse	Aufgabe	Notwendige Attribute	Notwendige Kommunikationspartner									
Controller, Vertrag	<table border="1"> <thead> <tr> <th>Name der Aufgabe</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>vertrag_speichern(), add_to_db_vertrag()</td> <td>Vertrag-Objekt erstellen und in der Datenbanktabelle speichern</td> </tr> </tbody> </table>	Name der Aufgabe	Beschreibung	vertrag_speichern(), add_to_db_vertrag()	Vertrag-Objekt erstellen und in der Datenbanktabelle speichern	vertragsid:int v_objid:string datum_ab:d datum_bis:d datum_vertrag:d kundenid:int grund:string abnahme:string anmerkung:string anz_einh: int	<table border="1"> <thead> <tr> <th>Name der Komponente</th> </tr> </thead> <tbody> <tr> <td>DB: Vertrag</td> </tr> <tr> <td>GUI</td> </tr> <tr> <td>DB:Betriebsmittel</td> </tr> <tr> <td>DB:Kunde</td> </tr> </tbody> </table>	Name der Komponente	DB: Vertrag	GUI	DB:Betriebsmittel	DB:Kunde
Name der Aufgabe	Beschreibung											
vertrag_speichern(), add_to_db_vertrag()	Vertrag-Objekt erstellen und in der Datenbanktabelle speichern											
Name der Komponente												
DB: Vertrag												
GUI												
DB:Betriebsmittel												
DB:Kunde												

Tabelle 19: Geschäftsfall 5 - Vertrag anlegen

5.1.6 Detailanforderung: Geschäftsfall 6

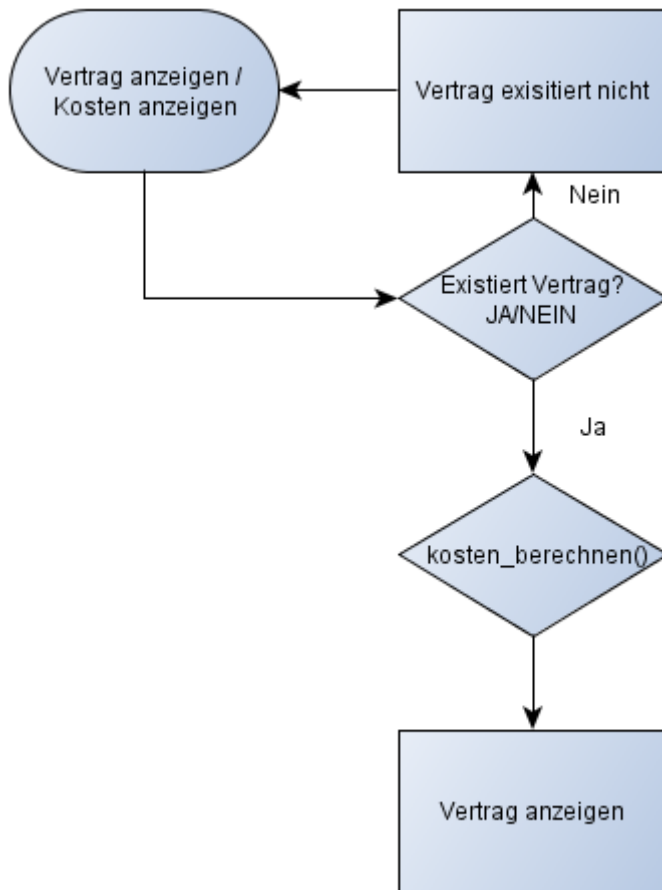


Abbildung 14: Flussdiagramm - Kosten anzeigen

Beteiligte Klasse	Aufgabe	Notwendige Attribute	Notwendige Kommunikationspartner							
Controller	<table border="1"> <thead> <tr> <th>Name der Aufgabe</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>kosten_berechnen(), vertrag_auslesen()</td> <td>Vertrag aus der Datenbanktabelle auslesen und Kosten darstellen</td> </tr> </tbody> </table>	Name der Aufgabe	Beschreibung	kosten_berechnen(), vertrag_auslesen()	Vertrag aus der Datenbanktabelle auslesen und Kosten darstellen	vertragsid:int	<table border="1"> <thead> <tr> <th>Name der Komponente</th> </tr> </thead> <tbody> <tr> <td>DB: Vertrag</td> </tr> <tr> <td>GUI</td> </tr> </tbody> </table>	Name der Komponente	DB: Vertrag	GUI
Name der Aufgabe	Beschreibung									
kosten_berechnen(), vertrag_auslesen()	Vertrag aus der Datenbanktabelle auslesen und Kosten darstellen									
Name der Komponente										
DB: Vertrag										
GUI										

Tabelle 20: geschäftsfall 6 - Kosten berechnen / Vertrag anzeigen

PFLICHTENHEFT

6 Messages

Geschäftsfall	Fehler	Feld	Art der Meldung	Message
Kunden anzeige	ja	kundenid	W	Anzeige &kundenid nicht möglich, da der Kunde nicht existiert
Betriebsmittel anzeigen	ja	objid	W	Anzeige der &objid nicht möglich, da das Betriebsmittel nicht existiert
Vertrag anzeigen	ja	vertragsid	W	Anzeige der &vertragsid nicht möglich, da der Vertrag nicht existiert
Betriebsmittel anlegen	ja	objid	W	Anlegen des Betriebsmittel &objid nicht möglich, da das Betriebsmittel bereits existiert
	nein	objid	I	Betriebsmittel &objid wurde erfolgreich angelegt
Kunden anlegen	ja	kundenid	W	Anlegen des Kunden &kundenid nicht möglich, da der Kunde bereits existiert
	nein	kundenid	I	Kunde &kundenid wurde erfolgreich angelegt
Betriebsmittel bearbeiten	ja	objid	W	Bearbeiten des Betriebsmittels &objid nicht möglich, da das Betriebsmittel nicht existiert
	nein	objid	I	Betriebsmittel &objid wurde erfolgreich bearbeitet
Kunden bearbeiten	ja	kundenid	W	Bearbeiten des Kunden &kundenid nicht möglich, da der Kunde nicht existiert
	nein	kundenid	I	Kunde &kundenid wurde erfolgreich bearbeitet
Betriebsmittel löschen	Ja	objid	W	Löschen des Betriebsmittels &objid nicht möglich, da das Betriebsmittel nicht existiert
	nein	objid	PopUp	Wirklich Betriebsmittel &objid löschen? Ja / Nein
	nein	objid	I	Betriebsmittel &objid wurde erfolgreich gelöscht
Kunden löschen	Ja	kundenid	W	Löschen des Kunden &kundenid nicht möglich, da der Kunde nicht existiert
	nein	kundenid	PopUp	Wirklich Kunden &kundenid löschen? Ja / Nein
	nein	kundenid	I	Kunde &kundenid wurde erfolgreich gelöscht
Vertrag anlegen	ja	kundenid	W	Anlegen des Vertrages &vertragsid nicht möglich, da der Kunde &kundenid nicht existiert
	ja	objid	W	Anlegen des Vertrages &vertragsid nicht möglich, da das Betriebsmittel &objid nicht existiert
	ja	v_objid	W	Anlegen des Vertrages &vertragsid nicht möglich, da das Betriebsmittel

PFLICHTENHEFT

Geschäftsfall	Fehler	Feld	Art der Meldung	Message
Vertrag anlegen	ja	v_objid	W	&objid vermietet ist
	ja	status, objid	W	Anlegen des Vertrages &vertragsid nicht möglich, da Wartungsdatum des Betriebsmittel &objid überschritten wurde
	ja	vertragsid	W	Anlegen des Vertrages &vertragsid nicht möglich, da der Vertrag bereits existiert
	nein	kundneid, objid, status	I	Vertrag &vertragsid wurde erfolgreich angelegt

Tabelle 21: Definierte Messages

PFLICHTENHEFT

7 GUI

7.1 Startbildschirm

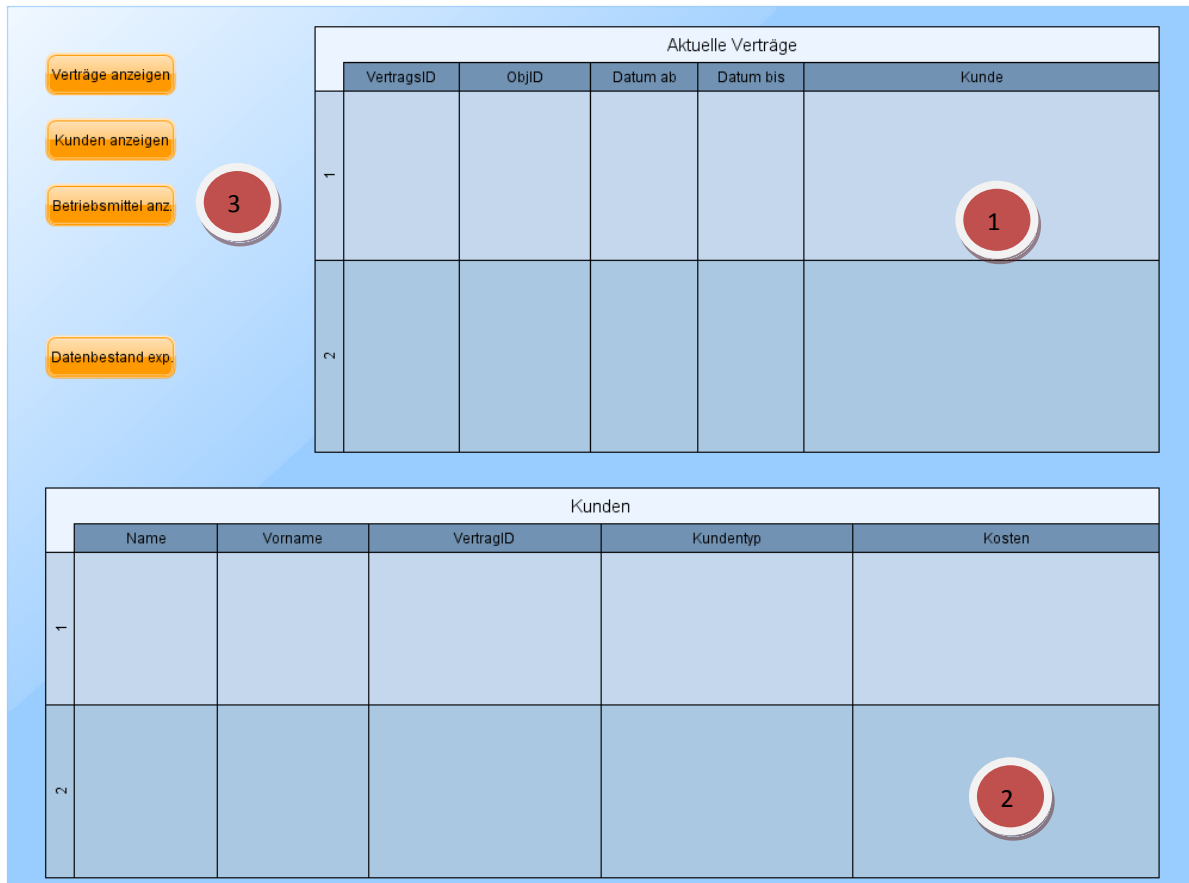


Abbildung 15: Skizze - „Startbildschirm“ der Anwendung

Beim Starten der Anwendung erscheint der Startbildschirm. Der Startbildschirm enthält eine Anzeige zu den laufenden Verträgen, siehe Punkt 1. Die zweite Anzeige dient zur Darstellung der Kunden und Kosten für die laufenden Verträge, siehe Punkt 2. Über die Steuerung des Startbildschirms lassen sich detaillierte Informationen zu Verträgen, Kunden und Betriebsmittel anzeigen. Außerdem wird ermöglicht den Datenbestand über die Schaltfläche „Datenbestand exportieren“ zu exportieren, siehe Punkt 3.

PFLICHTENHEFT

7.2 Verträge anzeigen

VertragsID: Datum von: bis:

Anzeigen Zurück

Verträge											
VertragsID	ObjID	Datum ab	Datum bis	Datum Vertrag	KundenID	Grund	Abnahme	Anmerkung	Anzahl Einheiten	Kosten	
1											
2											

Hinzufügen Exportieren

Abbildung 16: Skizze - Ansicht "Verträge anzeigen"

Dem Benutzer stehen zwei Auswahlkriterien für die Ausgabe der Verträge zur Verfügung, siehe Punkt 1. Über das Auswahlkriterium „VertragsID“ kann der Benutzer einen bestimmten Vertrag in der Tabelle, siehe Punkt 2 anzeigen lassen. Über Auswahlkriterium „Datum von bis“ kann er sich Verträge in einem bestimmten Zeitraum ausgeben lassen. Bleiben die Felder leer, werden alle Verträge in der Datenbanktabelle „Vertrag“ ausgegeben. Über die Schaltfläche „Hinzufügen“ gelangt der Benutzer zum Erstellen eines neuen Vertrages, siehe Punkt 3. Beim Anklicken der Schaltfläche „Exportieren“ wird der ganze Inhalt der Datenbanktabelle „Vertrag“ ins Excel exportiert, siehe Punkt 4.

PFLICHTENHEFT

7.2.1 Verträge hinzufügen

VertragsID: 1

ObjID:

Datum ab:

Datum bis:

Datum Vertrag:

KundenID:

Grund:

Anz. der Einheiten:

Abnahme:

Anmerkung:

Verträge										
VertragsID	ObjID	Datum ab	Datum bis	Datum Vertrag	KundenID	Grund	Abnahme	Anmerkung	Anz. Einh.	Kosten
1										
2									2	

Abbildung 17: Skizze – Ansicht „Verträge hinzufügen“

Der Benutzer muss die offenen Felder ausfüllen, siehe Punkt 1 und auf die Schaltfläche „Hinzufügen“ klicken um den Vertrag zu erstellen und in die Datenbanktabelle einzufügen. Nach dem erfolgreichen Erstellen des Vertrages erscheint er in der rechten Ansicht der Verträge, siehe Punkt 2.

PFLICHTENHEFT

7.3 Kunden anzeigen

KundenID: Ort:
 Kundentyp:

1

Kunden									
KundenID	Kundentyp	Name	Vorname	Andrede	Straße	Hausnummer	PLZ	Ort	
1									
2									

3

Abbildung 18: Skizze - Ansicht "Kunden anzeigen"

Die Kunden lassen sich über drei Auswahlkriterien KundenID, Ort und Kundentyp anzeigen, siehe Punkt 1. Die Informationen werden in der geeigneten Anzeigetabelle dargestellt, siehe Punkt 2. Dazu kann der Benutzer neue Kunden Hinzufügen, Bearbeiten, Löschen und Exportieren, siehe Punkt 3. Bearbeiten oder Löschen eines Kunden soll über die Markierung eines Kunden in der Anzeigetabelle und betätigen der Schaltfläche „Bearbeiten“ oder „Löschen“ geschehen.

PFLICHTENHEFT

7.3.1 Kunden hinzufügen

KundenID: 1

Kundentyp:

Name:

Vorname:

Anrede:

Straße:

Hausnummer:

PLZ:

Ort:

Kunden									
KundenID	Kundentyp	Name	Vorname	Anrede	Straße	Hausnummer	PLZ	Ort	
1									
							2		

Abbildung 19: Skizze - Ansicht "Kunden hinzufügen"

Der Benutzer muss die offenen Felder ausfüllen, siehe Punkt 1 und auf die Schaltfläche „Hinzufügen“ klicken um einen neuen Kunden zu erstellen und in die Datenbanktabelle „Kunde“ einfügen. Nach der erfolgreichen Erstellung des Kunden, erscheint der erstellte Kunde in der rechten Ansicht der Kunden-Tabelle, siehe Punkt 2.

PFLICHTENHEFT

7.3.2 Kunden bearbeiten

The screenshot shows a web interface for editing customer data. On the left side, there is a form with the following fields: KundenID (highlighted with a red circle '1'), Kundentyp, Name, Vorname, Anrede, Straße, Hausnummer, PLZ, and Ort. Below the form are two buttons: 'Bearbeiten' and 'Zurück'. On the right side, there is a table titled 'Kunden' with the following columns: KundenID, Kundentyp, Name, Vorname, Anrede, Straße, Hausnummer, PLZ, and Ort. A red circle '2' is placed over the 'Hausnummer' column in the table.

Abbildung 20: Skizze - Ansicht "Kunden bearbeiten"

Durch das Betätigen der Schaltfläche „Bearbeiten“ in der Ansicht „ Kunden anzeigen“, wird die KundenID fest vorgegeben und kann daher nicht mehr editiert werden, siehe Punkt 1. Die restlichen Angaben zum Bearbeiten des Kunden müssen gemacht werden. Durch das Anklicken der Schaltfläche „Bearbeiten“ werden die alten Werte des Kunden mit den neuen überschrieben und in der rechten Anzeigetabelle „Kunde“ dargestellt, siehe Punkt 2.

PFLICHTENHEFT

7.4 Betriebsmittel anzeigen

ObjektID: 1

Status:

Betriebsmittel					
ObjektID	Objekttyp	Zustand	Staus	Kosten	Sonstiges
1					
2					

Abbildung 21: Skizze - Ansicht "Betriebsmittel anzeigen"

Der Benutzer kann die Ansicht der Betriebsmittel über zwei Auswahlkriterien ObjektID und Status selektieren, siehe Punkt 1. Der Rest der Funktionen ist analog zum Kunden zu betrachten.

PFLICHTENHEFT

7.4.1 Betriebsmittel hinzufügen

ObjektID:

Objektyp:

Zustand:

Status:

Kosten:

Sonstiges:

Betriebsmittel				
ObjektID	Objektyp	Zustand	Staus	Sonstiges
1				
2				

Abbildung 22: Skizze - Ansicht "Betriebsmittel hinzufügen"

PFLICHTENHEFT

7.4.2 Betriebsmittel bearbeiten

Abbildung 23: Skizze - Ansicht "Betriebsmittel bearbeiten"

8 Suchhilfen

View	Feldname	Feld	Felder der Suchhilfe
Startbildschirm	-		-
Verträge anzeigen	VertragsID	vertragsid	kundenid, v_objjid, name, vorname
	Datum von	datum_von	-
	Datum bis	datum_bis	-
Vertrag hinzufügen	VertragsID	vertragsid	-
	ObjID	v_objjid	typ, status, kosten
	Datum ab	datum_ab	-
	Datum bis	datum_bis	-
	Datum Vertrag	datum_vertg	-
	KundenID	kundenid	name, vorname, kundentyp
	Grund	grund	-
	Anz. der Einheiten	anz_einh	-
	Abnahme	abnahme	-
Anmerkung	anmerkung	-	
Kunden anzeigen	KundenID	kundenid	name, vorname
	Kundentyp	kundentyp	Mitarbeiter, Privat, Einmalkunde

PFLICHTENHEFT

View	Feldname	Feld	Felder der Suchhilfe
Kunden anzeigen	Ort	ort	-
Kunden hinzufügen	KundenID	kundenid	-
	Kundentyp	kundentyp	Mitarbeiter, Privat, Einmalkunde
	Name	name	-
	Vorname	vorname	-
	Anrede	anrede	Herr, Frau, Dr., Dr. Ing., B. Sc., M.Sc.
	Straße	strasse	-
	Hausnummer	hausnummer	-
	PLZ	plz	-
Kunden bearbeiten	Ort	ort	-
	Kundentyp	kundentyp	Mitarbeiter, Privat, Einmalkunde
	Name	name	-
	Vorname	vorname	-
	Anrede	anrede	Herr, Frau, Dr., Dr. Ing., B. Sc., M.Sc.
	Straße	strasse	-
	Hausnummer	hausnummer	-
	PLZ	plz	-
Betriebsmittel anzeigen	Ort	ort	-
	ObjektID	objid	typ, zustand, kosten
Betriebsmittel hinzufügen	Status	status	grün, gelb, rot
	ObjektID	objid	-
	Objekttyp	typ	-
	Zustand	zustand	-
	Status	status	grün, gelb, rot
	Kosten	kosten	-
	Sonstiges	sonstiges	-
Betriebsmittel bearbeiten	Objekttyp	typ	-
	Zustand	zustand	-
	Status	status	grün, gelb, rot
	Kosten	kosten	-
	Sonstiges	sonstiges	-

Tabelle 22: Suchhilfen

PFLICHTENHEFT

9 Technische Architektur

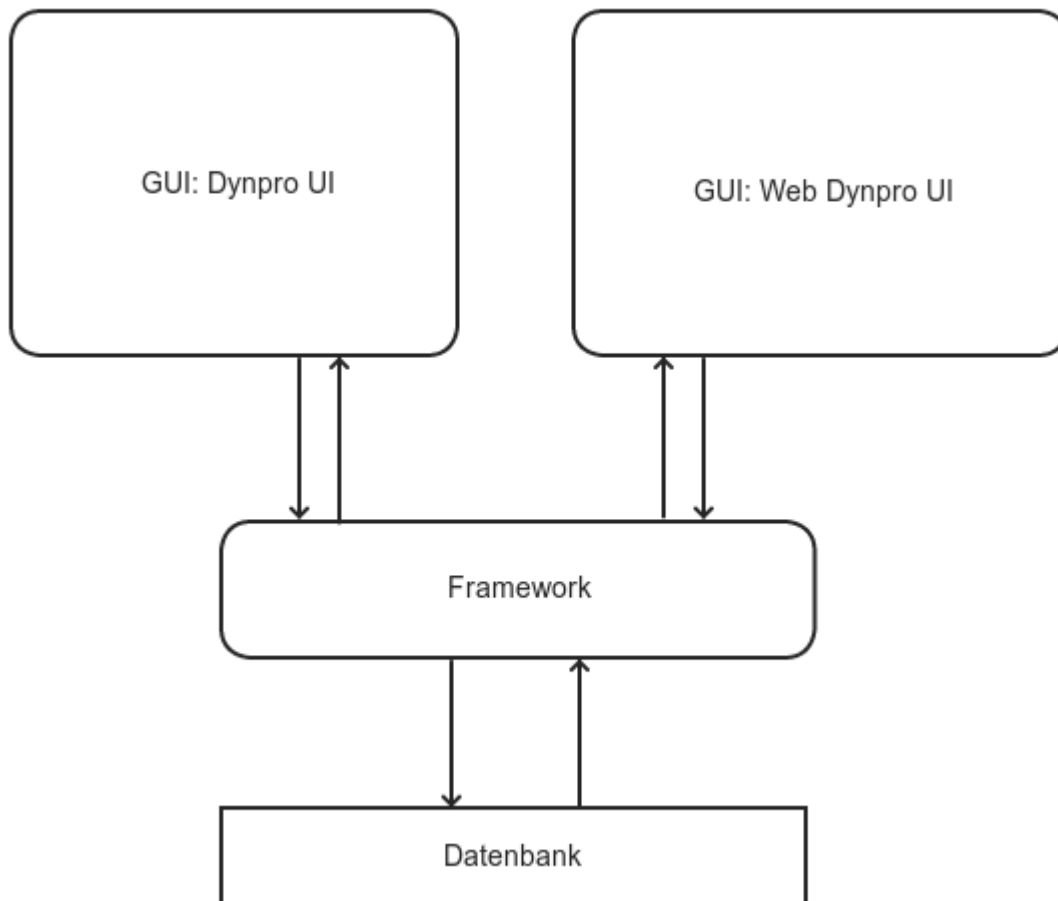


Abbildung 24: Systemarchitektur

10 Nichtfunktionale Anforderungen

Das Endprodukt soll den EN ISO 9241-11 Standard erfüllen. Ziel ist es, eine effektive, effiziente und zufriedenstellende Anwendung zu erstellen. Hierbei soll zusätzlich ein besonderes Augenmerk auf die Zuverlässigkeit, Wartbarkeit sowie die Skalierbarkeit gelegt werden.

10.1 Zuverlässigkeit

Das System muss jederzeit (bei jedweder Belastungsgröße) zuverlässig reagieren. Es darf nicht zu unkontrollierten Systemabstürzen oder Datenverlust kommen. Programme und Daten müssen gegen zufällige und unabsichtliche Veränderungen geschützt werden.

10.2 Benutzerfreundlichkeit

Der Anwender muss zeitnah (Antwortzeit < 0,5s) auf Fehler und falsche Eingaben hingewiesen werden. Er muss durch eine Hilfefunktion bei der Anwendung unterstützt werden. Die graphischen Oberflächen müssen übersichtlich, einheitlich strukturiert und robust sein und die geforderte Funktionalität anbieten. Sie müssen intuitiv bedienbar sein, das heißt der Anwender muss ohne Schulung, also nur mit der angebotenen Hilfefunktion, fähig sein mit dem System umzugehen. Bei der Erstellung der grafischen Oberflächen sind des

PFLICHTENHEFT

Weiteren die gängigen Normen und Richtlinien der Softwareergonomie zu berücksichtigen. Die Erstellung einer barrierefreien Benutzeroberfläche ist nicht zwingend gefordert.

11 Beteiligte

11.1.1 Kunde/Auftraggeber/Interne Beteiligte

Ansprechpartner	Name	OrgE	Telefon
Alle Themen	Felix Bodewald	OctaVIA AG	Siehe Adressbuch
Alle Themen	Stefan Schellhase	OctaVIA AG	Siehe Adressbuch

Tabelle 23: Kunde/Auftraggeber/Interne Beteiligte

11.1.2 Anwender

Nutzer	Geschätzte Anzahl an Personen	Nutzungsart
Germerott		Endnutzer
Hildalgo-Well		Endnutzer
Trebing		Endnutzer

Tabelle 24: Anwender

12 Umfang der Anforderung

Komponente	Zeit	Entwickler
Interface	???	Daniel Kawaletz, Andreas Dinkel
Klasse Kunde	2 Tage	Daniel Kawaletz, Andreas Dinkel
Klasse Vertrag	5 Tage	Daniel Kawaletz, Andreas Dinkel
Klasse Betriebsmittel	2 Tage	Daniel Kawaletz, Andreas Dinkel
DB: Kunde	0,5 Tag	Daniel Kawaletz, Andreas Dinkel
DB: Vertrag	0,5 Tag	Daniel Kawaletz, Andreas Dinkel
DB: Betriebsmittel	0,5 Tag	Daniel Kawaletz, Andreas Dinkel
Klasse Controller	5 Tage	Daniel Kawaletz, Andreas Dinkel
GUI: Dynpro	0,5 Tag	Daniel Kawaletz, Andreas Dinkel
Datenexport	0,5 Tag	Daniel Kawaletz, Andreas Dinkel
GUI: Web Dynpro	0,5 Tag	Daniel Kawaletz, Andreas Dinkel
Suchhilfen / Messages	1 Tag	Daniel Kawaletz, Andreas Dinkel
Test	0,5 Tag	Daniel Kawaletz, Andreas Dinkel

Tabelle 25: Umfang der Anforderung

Aus der Tabelle 11 geht hervor, dass für die Entwicklung und Test 18,5 Tage benötigt werden.

PFLICHTENHEFT

13 Offene Punkte

Kapitel / Seite	OPP/Anmerkung	Beschluss	Art	Verantwortlich	Erledigungsdatum
Interface			DS		
Kosten			DS		
Sperrkonzept			S		

Tabelle 26: Offene Punkte

Spalte „Art“: I = Inhaltlicher Fehler, B = Beschreibung fehlt, D = Detaillierung nicht ausreichend, S = Sonstiges

14 Testanforderungen

15 Abkürzungsverzeichnis

Abkürzung	Beschreibung
GUI	Graphical User Interface, deutsch: grafische Benutzeroberfläche
DB	Datenbanktabelle

Tabelle 27: Abkürzungsverzeichnis

16 Quellenverzeichnis