

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>IV</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Quellcodeverzeichnis</b>	<b>VI</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Lastenheft</b>	<b>2</b>
2.1 Zielsetzung . . . . .	2
2.2 Anwendungsbereiche . . . . .	2
2.3 Zielgruppen, Benutzerrollen und verantwortlichkeiten . . . . .	2
2.4 Zusammenspiel mit anderen Systemen . . . . .	3
2.5 Produktfunktionen . . . . .	4
2.6 Produktdaten . . . . .	5
2.7 Produktleistungen . . . . .	6
2.8 Qualitätsanforderungen . . . . .	6
<b>3 Aufgaben</b>	<b>7</b>
3.1 Analyse . . . . .	7
3.2 Sequenzdiagramm und Aktivitätsdiagramm . . . . .	7
3.3 Entwurf . . . . .	8
3.4 Implementierung . . . . .	9
<b>4 Vereinfachung für den Programmentwurf</b>	<b>11</b>
<b>5 Analyse</b>	<b>12</b>
5.1 Einleitung . . . . .	12
5.2 Lastenheft . . . . .	14
5.2.1 Zielsetzung . . . . .	14
5.2.2 Anwendungsbereiche . . . . .	17
5.2.3 Zielgruppen, Benutzerrollen und Verantwortlichkeiten . . . . .	18
5.2.4 Zusammenspiel mit anderen Systemen . . . . .	20
5.2.5 Produktfunktionen . . . . .	24
5.2.6 Produktdaten . . . . .	35
5.2.7 Produktleistungen . . . . .	36
5.2.8 Qualitätsanforderungen . . . . .	37

<b>6</b>	<b>Use-Case Diagramm</b>	<b>39</b>
6.1	Vorüberlegung . . . . .	39
6.2	Use-Case Analyse . . . . .	41
6.3	Use-Case Diagramm . . . . .	46
<b>7</b>	<b>Analyse-Klassen-Diagramm</b>	<b>48</b>
7.1	Klassen . . . . .	48
7.2	Diagramm . . . . .	53
<b>8</b>	<b>Sequenzdiagramm</b>	<b>54</b>
8.1	Aktionsbetrachtung: Buchung eines Fahrzeugs . . . . .	55
8.2	Pseudo-Code . . . . .	56
8.3	Diagramme . . . . .	59
8.3.1	Buchung anlegen . . . . .	59
8.3.2	Fahrt antreten . . . . .	61
8.3.3	Fahrt beenden . . . . .	62
<b>9</b>	<b>Aktivitätsdiagramm</b>	<b>64</b>
9.1	Pseudo-Code . . . . .	64
9.2	Diagramme . . . . .	67
9.2.1	Standort anlegen . . . . .	68
9.2.2	Fahrzeug anlegen . . . . .	70
9.2.3	Fahrzeug einem Standort zuordnen . . . . .	72
<b>10</b>	<b>Entwurfsklassendiagramm</b>	<b>74</b>
10.1	Diagramm . . . . .	74
10.2	Pakete . . . . .	75
10.2.1	model.buchung . . . . .	75
10.2.2	model.fahrzeug . . . . .	75
10.2.3	model.kunde . . . . .	75
10.2.4	model.standort . . . . .	76
10.2.5	model.utils . . . . .	76
10.2.6	app . . . . .	76
10.2.7	swe-utils . . . . .	76
10.2.8	database . . . . .	77
10.2.9	gui . . . . .	77
10.3	Entwurfsmuster . . . . .	77
10.3.1	EntityManager . . . . .	77
10.3.2	Builder . . . . .	78
10.3.3	EntityFactory . . . . .	78
10.3.4	Singleton . . . . .	78
10.3.5	Beobachter . . . . .	78
10.3.6	Kompositum . . . . .	79

<b>11 GUI-Entwurf</b>	<b>80</b>
11.1 Fahrzeuge verwalten . . . . .	80
11.1.1 Fahrzeug löschen . . . . .	82
11.1.2 Fahrzeug bearbeiten . . . . .	83
11.1.3 Fahrzeug anlegen . . . . .	84
11.2 Standorte verwalten . . . . .	85

# Abkürzungsverzeichnis

# Abbildungsverzeichnis

6.1	Use-Case Diagramm . . . . .	46
6.2	Use-Case 4: Fahrzeug verwalten . . . . .	47
7.1	Analyseklassendiagramm . . . . .	53
8.1	Buchung eines Termins . . . . .	60
8.2	Antritt eines gebuchten Termins . . . . .	61
8.3	Abschluss eines gebuchten Termins . . . . .	62
8.4	Erstellung der Mahnungen . . . . .	63
9.1	Aktivitätsdiagramm: Standort anlegen . . . . .	68
9.2	Aktivitätsdiagramm: Fahrzeug anlegen . . . . .	70
9.3	Aktivitätsdiagramm: Fahrzeug einem Standort zuordnen . . . . .	72
10.1	Aktivitätsdiagramm: Fahrzeug einem Standort zuordnen . . . . .	74
11.1	Mock-Up: Liste an Fahrzeugen . . . . .	81
11.2	Mock-Up: Detailansicht eines Fahrzeugs . . . . .	81
11.3	Mock-Up: Löschen eines Fahrzeugs . . . . .	82
11.4	Mock-Up: Bearbeiten eines Fahrzeugs . . . . .	83
11.5	Mock-Up: Anlegen eines Fahrzeugs . . . . .	84
11.6	Mock-Up: Liste an Standorten . . . . .	85
11.7	Mock-Up: Detailansicht von Standorten . . . . .	86

# Quellcodeverzeichnis

8.1	Szenario Event anlegen . . . . .	56
9.1	Szenario Standort mit Fahrzeugen anlegen . . . . .	64

# 1 Einleitung

Für unsere Carsharing-Organisation *Citycar\*BaDö* benötigen wir ein neues Buchungssystem, um dem wachsenden Bedarf an gemeinsam genutzten Fahrzeugen gerecht werden zu können.

*Citycar\*BaDö* hat inzwischen fast 500 Mitglieder, denen eine Fahrzeugflotte von ca. 60 Fahrzeugen an ungefähr 30 Standorten in und um Bad Dödelhausen zur Verfügung steht.

Bisher werden die Mitgliedschaften und Vermietungen mit einem inzwischen in die Jahre gekommenen Buchungssystem verwaltet, das sich recht umständlich bedienen lässt. Zwar können Buchungen online erfolgen, aber wir würden gerne zusätzliche Informationen für die Online-Kunden zur Verfügung stellen und eine Erweiterung der alten Software lohnt sich nicht.

## 2 Lastenheft

### 2.1 Zielsetzung

Ziel des Entwicklungsauftrags ist eine Software für die Verwaltung aller Daten, die für die Verwaltung der Fahrzeuge, Kunden sowie Buchungen unserer Carsharing-Organisation anfallen und benötigt werden.

Alle Daten sollen zentral gespeichert werden, da mehrere Benutzer gleichzeitig auf die Daten und Termine zugreifen werden.

Ein selektiver Import und Export von Daten über lesbare Dateien muss für Backups und zum Datenaustausch möglich sein.

Eine intuitive, leicht bedienbare Benutzeroberfläche setzen wir als selbstverständlich voraus. Es sollen keine besonderen Computerkenntnisse zur Bedienung der Software erforderlich sein.

### 2.2 Anwendungsbereiche

Die Software soll ausschließlich für die Verwaltung von Fahrzeugen, Kunden, Ausrüstung, Fahrzeugstandorte und Angestellten und den damit direkt verbundenen Elementen verwendet werden. Sie soll im Alltag auf Desktop-Rechnern und Laptops eingesetzt werden.

### 2.3 Zielgruppen, Benutzerrollen und verantwortlichkeiten

Es soll verschiedene Benutzerrollen geben:

- Organisatorinnen und Organisatoren pflegen die jeweiligen Buchungsdaten und Fahrzeuge.



- Personalmitarbeiter pflegen Mitarbeiterdaten im System
- Eine hauptverantwortliche Person (Administrator) hat Vollzugriff auf sämtliche Daten, vor allem für deren Import und Export sowie deren Backup.
- Es gibt keine Gruppen oder Abteilungen, die verwaltet werden müssen.

## 2.4 Zusammenspiel mit anderen Systemen

Die Daten über die Angestellten (Gehälter bzw. Löhne, Steuern, Kranken- und Rentenversicherung usw.) werden separat durch ein vorhandenes Personalbuchhaltungsprogramm verwaltet und müssen hier nicht berücksichtigt werden. Die finanztechnischen Daten werden über unser vorhandenes Finanzsystem erfasst und müssen hier ebenfalls nicht berücksichtigt werden.

Die Software soll aus zwei Teilen bestehen:

- Für die Mitarbeiter im Büro soll eine Desktop-Anwendung erstellt werden, mit denen die Datenbestände verwaltet werden können. Es sollen auch Buchungen erstellt werden können für Kunden, die keine Online-Buchungen machen wollen und persönlich in der Carsharing-Filiale erscheinen.
- Eine neue Web-Seite soll unseren Online-Kunden ermöglichen, nach einer Authentifizierung alle Standorte anzeigen zu lassen sowie natürlich die dort befindlichen Fahrzeuge, welche für einen anzugebenden Zeitbereich online gebucht werden können.

Die Web-Seite soll mit dem ersten Teilauftrag noch nicht programmiert werden, allerdings benötigen wir ein klares Konzept, wie diese Web-Seite realisiert werden soll (Schnittstellen usw.).

Möglichst alle Daten sollen vom alten in das neue System übertragen werden.

## 2.5 Produktfunktionen

/LF10/	<p>Der jeweilige Benutzer muss die Möglichkeit haben, über eine grafische Benutzeroberfläche alle für ihn relevanten Daten einfach und übersichtlich zu verwalten.</p> <p>Es sollen zahlreiche Konfigurationsdaten gespeichert und beim nächsten Start des Programms verwendet werden (z.B. aktuelle Größe und Position des Fensters). Daneben sollen einige Elemente vor dem Start konfigurierbar sein (z.B. Überschriften, Schriftarten und -größen usw.).</p>
/LF20/	<p>Verwaltet werden sollen Mitarbeiter, Fahrzeuge, Standorte, Kunden, Buchungen, Rechnungen, Änderungen, Stornierungen, Mahnungen usw.</p> <p>Es muss möglich sein, jederzeit erkennen zu können, welche angestellte Person einen Datensatz angelegt, geändert oder gelöscht hat.</p>
/LF30/	<p>Buchungen haben eine Start- und einen Endtermin, Terminüberschneidungen müssen vermieden werden, um die Verfügbarkeit sicherzustellen.</p>
/LF40/	<p>Unsere Kunden haben neben ihren Kontaktdaten auch Vertragsunterlagen für die Teilnahme am Carsharing, die die Höhe des Eigenanteils für einen Schadensfall der Versicherung sowie die Höhe der Teilnahme-Kautions enthält. Diese Vertragsunterlagen werden von uns eingescannt und sollen als Dokument mit den Kundendaten gespeichert werden. Daneben wird jedem Kunden eine Karte zum Öffnen und Schließen der Fahrzeuge ausgehändigt.</p> <p>Ein kleiner Prozessor im Fahrzeug sendet nach Fahrtende (Terminende) die exakte Start- und Ende-Zeit sowie die gefahrenen Kilometer an einen Server. Diese Daten sollen von dem Server nach Buchungsende geholt und zur Berechnung der Kosten für die Buchung (und somit für die Rechnung) verwendet werden.</p>

/LF50/	<p>Die Fahrzeuge selbst gehören unterschiedlichen Kategorien an: Kleinfahrzeuge, Mittelklassefahrzeuge, gehobene Mittelklasse und Transportfahrzeuge.</p> <p>Allen Kategorien sind eine bestimmte Höhe der Stunden-Mietpauschale und die Kosten pro gefahrenem km zugeordnet. Alle Werte sollen konfigurierbar sein.</p> <p>Alle Fahrzeuge werden regelmäßig von Fremdfirmen gewartet. Die entsprechenden Dienstleistungen sollen den Fahrzeugen chronologisch zugeordnet werden.</p>
/LF60/	<p>Einem Standort können ein oder mehrere Fahrzeuge zugeordnet sein. Ein Fahrzeug ist immer nur einem Standort zugeordnet.</p>
/LF70/	<p>Nach jeder Fahrt werden sofort die Rechnungen erstellt und dem Kunden per E-Mail zugesandt. Die Bezahlung der Rechnungen erfolgt über Bankeinzug, was durch das Finanzbuchhaltungssystem (FBH) erledigt wird und hier nicht betrachtet werden muss. Allerdings muss über die vorhandene Schnittstelle des FBH der Stand der Rechnungsbegleichung abgefragt werden, damit über das neue System erkennbar ist, ob und wann eine Rechnung bezahlt wurde.</p>
/LF80/	<p>Buchungen können bis 10 Stunden vor Antritt der Fahrt storniert werden.</p>
/LF90/	<p>Zur einfacheren Eingabe der Daten soll es Auswahllisten für deren Eigenschaften geben, wo immer es möglich ist. Die Auswahllisten sollen auf einfache Weise erweiterbar sein.</p>
/LF100/	<p>Sämtlichen Elementen sollen mehrere Bilder mit Titel zugeordnet werden können, die zentral auf einem Verzeichnis liegen sollen</p>

## 2.6 Produktdaten

/LD10/	<p>Die Daten sollen zunächst in einer zentralen Datenbasis (lesbare Dateien) abgespeichert und später in eine Datenbank überführt werden.</p>
--------	---

## 2.7 Produktleistungen

/LL10/	Die Anzahl der zu verwaltenden Elemente wird auf ca. 100.000 geschätzt.
/LL20/	Um bei HW- und SW-Anschaffungen und -neuerungen flexibel zu bleiben, ist auf Plattformunabhängigkeit besonders zu achten.

## 2.8 Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität	X			
Zuverlässigkeit		X		
Effizienz		X		
Benutzbarkeit (auch Gestaltung)	X			
Wartbarkeit			X	
Übertragbarkeit (Portabilität)			X	

## 3 Aufgaben

Es handelt sich hier um eine vereinfachte Verwaltungs-Software. Einzelne Lastenheftpunkte sind bewusst offengehalten. Denken Sie darüber nach, welche Informationen zusätzlich sinnvoll oder auch notwendig sind. Recherchieren Sie evtl. nach einzelnen Zusammenhängen im Internet.

### 3.1 Analyse

Für die Analyse sind zu erstellen:

- Analyse des Lastenhefts (Fragen und Antworten).
- Ein Use-Case-Diagramm der gesamten Anwendung incl. Beschreibung.
- Eine Verfeinerung des Use-Case-Diagramms incl. Beschreibung. (nach Absprache)
- Ein Analyse-Klassendiagramm incl. Beschreibung (Untersuchen Sie dabei den Einsatz geeigneter Analysemuster)
- Einfache GUI-Skizzen (Mockups) von mindestens zwei wesentlichen GUI-Komponenten (Hauptseite, Tabs, etc.). Die Skizzen können mit einem einfachen Grafikprogramm erstellt werden. Auch sorgfältige Handzeichnungen sind erlaubt. Keine Login-GUI skizzieren!

### 3.2 Sequenzdiagramm und Aktivitätsdiagramm

Erstellen Sie ein Sequenzdiagramm und ein Aktivitätsdiagramm (incl. Beschreibung) für folgende Szenarios (ein AD für das eine Szenario, ein SD für das andere Szenario):

- Die Aktion „Standort mit neuen Fahrzeugen anlegen“ durchführen. Ausgehend von einem neuen Standort und **leerer Datenbasis** werden dessen gesamte Daten

erfasst und in das System eingetragen. (dies wird als Gebrauchsanweisung für die Evaluation Ihrer Implementierung dienen)

- Die Aktionen „Buchung eines Fahrzeugs“ durchführen. Hierbei soll eine komplette Buchung inklusive Beendigung der Fahrt und bezahlen der Rechnung modelliert werden.

Die Bewertung Ihrer Diagramme erfolgt auf der Basis der Nutzung der UML-Elemente, auf Ihrer Kreativität sowie dem Detaillierungsgrad des jeweiligen Diagramms.

Fassen Sie bei beiden Diagrammen die Eingabe aller primitiven Attribute eines Elements (Float, String, Integer, ...) in einer einzigen Aktion zusammen (z.B. „Attribute eintragen“).

Für das Sequenzdiagramm ist das gewählte Szenario ausführlich zu entwickeln (idealerweise mit Pseudocode oder einer anderen Modellierungsmethode Ihrer Wahl). Es sind sämtliche referenzierten Elemente zu berücksichtigen, die zugeordnet werden können.

In allen Fällen wird eine (noch) leere Datenbasis angenommen. Denken Sie an geeignete Diagrammverfeinerungen.

### 3.3 Entwurf

Abzuliefern sind hier (alle Diagramme und GUIs jeweils mit Beschreibung):

- Entwurfsklassendiagramm (Untersuchen Sie dabei den Einsatz geeigneter Entwurfsmuster)
- GUI-Modellierung: Es ist das Kommunikationsschema eines Teils der während der Analyse skizzierten GUI mit **UML** zu modellieren. Die Anwendung selbst soll dabei nach dem einfachen Model-View-Control-Muster aufgebaut sein. Dazu sind mindestens ein Controller, die erforderlichen Modellklassen sowie eine unabhängige GUI (View) erforderlich.
- Die meisten GUI-Elemente werden über eine einfache kleine Java-Bibliothek zur Verfügung gestellt (swe-utils.jar), deren GUI-Komponenten in das Klassendiagramm zu integrieren sind, wenn sie verwendet werden.

- Die GUI-Modellierung kann in einem separaten Diagramm mit den relevanten (gewählten bzw. benötigten) Modellklassen erfolgen, falls das Entwurfsklassendiagramm sonst zu komplex werden würde.

## 3.4 Implementierung

Es ist eine einfache Java-Applikation zu implementieren, die es ermöglicht, Carsharing-Daten anzulegen, zu ändern und zu löschen.

Zur Realisierung wird die oben bei der Entwurfsaufgabe erwähnte Java-Bibliothek zur Verfügung gestellt (*swe-utils.jar*), die neben mehreren GUI-Komponenten einen *CSVReader*, einen *CSVWriter* sowie mehrere Interfaces bereitstellt (in den Packages *event* und *model*).

Daneben ist eine Mini-Test-Applikation gegeben, die die Funktionsfähigkeit der GUI-Komponenten demonstriert (Start mit `java -jar swe-utils.jar`). Details sind der Java-Dokumentation der Bibliothek zu entnehmen.

Zur leichteren und zukunftsicheren Evaluation Ihres Programmentwurfs soll die Java-Applikation als eine Desktop-Applikation mit CSV-Dateien (alternativ XML oder JSON) als zentrale Datenbasis realisiert werden, die von beliebigen Rechnern aus gestartet wird. Dabei sind mehrere Dateien analog zu Datenbanktabellen zu erzeugen.

### Einzelne Aufgaben

- Hauptaufgabe ist die Realisierung einer MVC-Applikation mithilfe des Observer-Patterns entsprechend des vorgegebenen GUI-Entwurfs und der gegebenen Java-Bibliothek.
- Die Erzeugung der Instanzen soll in einer Entity-Factory erfolgen und zur Verwaltung der Instanzen ist ein Entity-Manager zu realisieren (beides siehe Vorlesung).
- Beim Anlegen einer Buchung muss für die Zuordnung von Hilfsmitteln sichergestellt sein, dass es keine zeitlichen Überschneidungen gibt (LF30+LF40).
- Es muss eine ausführbare JAR-Datei abgegeben werden, die mit  
„java -jar SWE-PE-2022\_Carsharing\_<name1>\_<name2>.jar OPTIONEN“

gestartet werden kann. Hierfür ist ein BASH-Skript namens *startApp* zu erstellen

- Geprüft wird das Anlegen einer Buchung mit der Zuordnung aller zugehörigen Elemente. Nach dem Anlegen wird die Applikation erneut gestartet und geprüft, ob alle Daten korrekt abgespeichert und beim Laden wieder zugeordnet werden.

### Verwendung von CSV-Dateien

- Die Daten sollen in CSV-Dateien vorliegen und können mittels den gegebenen Bibliotheksklassen *CSVReader* und *CSVWriter* gelesen bzw. beschrieben werden. Zur Vereinfachung können die Daten jeweils komplett geschrieben werden.
- Abgegeben werden soll ein ZIP-File (oder TAR-File) mit allen Java- und CSV-Dateien (letztere gesammelt in einem eigenen Verzeichnis):

„SWE-PE-2022\_Carsharing\_\_<n1>\_\_<n2>.zip (tar oder tar.z)

- Als OPTIONEN in der Startanweisung soll der Pfad zu den CSV-Dateien sowie zu einer Properties-Datei angegeben werden können:

„java -jar SWE-PE-2022\_Carsharing\_\_<n1>\_\_<n2>.jar -d <csvpath> -p <prop-file>”



## 4 Vereinfachung für den Programmentwurf

1. Es muss nicht dafür gesorgt werden, dass auf dieselben Daten bzw. CSV-Dateien nicht gleichzeitig zugegriffen werden kann, d.h. es ist kein *Locking*-Mechanismus erforderlich.
2. Eine Protokollierfunktion und ein Login-Vorgang sind für die Anwendung nicht erforderlich (in der Realität natürlich schon!).
3. Zeitliche Überschneidungen sind natürlich bei allen Buchungen möglich und müssten sowohl beim Anlegen als auch bei Änderungen von Terminen berücksichtigt werden. Im Programmentwurf sollte dies in der Modellierung berücksichtigt werden, bei der Implementierung ist jedoch nur eine Überprüfung bei der Auswahl des Starts und Endes der Buchung erforderlich.
4. Konfigurationsdaten (LF 10) sollen exemplarisch für wenige Elemente änderbar sein (Angabe der realisierten Elemente!)
5. Alle Elemente, die zu einer Buchung zugeordnet werden können, müssen nicht interaktiv erzeugbar, sondern können bereits in CSV-Dateien vorhanden sein. Verwenden Sie dabei realistische Attributwerte!

## 5 Analyse

Text in Grün steht für die Fragen der Analyse.

Text in Blau steht für die Antworten der Analyse.

### 5.1 Einleitung

Für unsere Carsharing-Organisation *Citycar\*BaDö* benötigen wir ein neues Buchungssystem, um dem wachsenden Bedarf an gemeinsam genutzten Fahrzeugen gerecht werden zu können.

*Citycar\*BaDö* hat inzwischen fast 500 Mitglieder, denen eine Fahrzeugflotte von ca. 60 Fahrzeugen an ungefähr 30 Standorten in und um Bad Dödelhausen zur Verfügung steht.

Bisher werden die Mitgliedschaften und Vermietungen mit einem inzwischen in die Jahre gekommenen Buchungssystem verwaltet, das sich recht umständlich bedienen lässt.

Existiert ein Buchhaltungssystem?

Ja, wir verwenden bereits ein extra Buchhaltungssystem für die Verwaltung aller finanztechnischen Aspekte

Welches Buchungssystem haben Sie verwendet?

Wir verwenden ein in die Jahre gekommenes SAP System von 1993.

Was war daran so umständlich?

Neben einer sehr unübersichtlichen und optisch unansprechenden Oberfläche ist die Navigation innerhalb der Anwendung nur mit Transaktionscodes möglich. Sobald man sich in das System eingearbeitet hat, ist es einfacher, doch unsere Mitarbeiter müssen trotzdem regelmäßig bestimmte Codes nachschlagen, wenn Aufgaben anstehen, die weniger zum Alltagsgeschäft gehören. Wir hätten dahingehend gerne einen übersichtliche und auf einen Blick einsehbare Navigation.

Was macht ein Mitglied aus, Eigenschaften hat es?

Ein Mitglied ist in dem Sinne nichts weiter als ein Kunde bei anderen Unternehmen, der die Dienstleistung unserer Firma in Anspruch nimmt. In diesem Sinne beschreibt es eine reale Person, mit allen Informationen, die zur Abbildung nötig sind. Besonders wichtig ist uns in dabei die Mail-Adresse, der Name sowie Vorname, die Adresse und eine Telefonnummer. Die Mitglieder sollen außerdem eindeutig identifizierbar sein.

Welche Anforderungen müssen erfüllt werden, damit man als Mitglied (ins System) aufgenommen werden kann?

Unsere Mitglieder sind natürlich von der SCHUFA als liquide eingestuft und müssen zur Anmeldung eine Bescheinigung vorlegen. Das Mindestalter ist 21 und man muss aus der Probezeit heraus sein. Für die Abwicklung der Transaktionen wird weiterhin ein Schweizer Bankkonto vorausgesetzt.

Zwar können Buchungen online erfolgen, aber wir würden gerne zusätzliche Informationen für die Online-Kunden zur Verfügung stellen und eine Erweiterung der alten Software lohnt sich nicht.

Welches Budget steht für das Projekt zur Verfügung?

In Anbetracht des Projektumfangs gedenken wir ein Budget von 8.697.370 Mauritius-Rupien (umgerechnet 187 000 €) bereitzustellen.

Was für zusätzliche Funktionen und Informationen konnte das alte Programm speziell nicht erfüllen?

Neben den Navigationsproblemen war der Datenexport nur in Form von Druckaufträgen bei uns möglich. Das würden wir gerne im Zusammenhang mit der Datensicherung modifizieren und auch andere Verfahren verwenden, die den Export vereinfachen. Die Informationsdarstellung erfolgte auch nur, indem man sich die betreffenden Texte als .txt-Dateien auf den lokalen Rechner herunterladen konnte. Wir wollen die Informationen in der Anwendung einsehen können.

## 5.2 Lastenheft

### 5.2.1 Zielsetzung

Ziel des Entwicklungsauftrags ist eine Software für die Verwaltung aller Daten, die für die Verwaltung der Fahrzeuge, Kunden sowie Buchungen unserer Carsharing-Organisation anfallen und benötigt werden.

Alle Daten sollen zentral gespeichert werden, da mehrere Benutzer gleichzeitig auf die Daten und Termine zugreifen werden.

Welche Hardware liegt für diese zentrale Speicherung vor (Server, Speicher, Betriebssystem)?

Im Zusammenhang mit unserem alten System haben wir einen eigenen Server in unserer Firmenzentrale stehen. Laut Angaben unserer IT-Abteilung besitzt der Server 32 GB RAM und 10 TB Speicher. Das laufende Betriebssystem ist Windows XP. Sollten im Rahmen der Entwicklung Hardware-Anschaffungen nötig sein, haben wir in unserem Budget einen Puffer eingeplant. Es wäre jedoch gut, wenn wir die bereits vorhandene Hardware auch integrieren könnten.

In welcher Form, in welchem Format sollen die Daten zentral gespeichert werden?

Dahingehend haben wir keine speziellen Anforderungen. Wir würden die Methode bevorzugen, die effizient und zugleich auch kostengünstig ist. Eine Datenbank wäre denkbar.

Wie viele Benutzer verwenden die Daten gleichzeitig?

Das kann man so pauschal leider nicht sagen. Die Zugriffszahlen hängen von der Tageszeit, Jahreszeit und dem aktuellen Geschehen ab. Auch wenn wir wachsende Kundenzahlen verbuchen ist es bis jetzt jedoch eher unüblich, dass z.B. mehrere Kunden gleichzeitig bei uns in den Filialen sind.

Wie soll mit Kollisionen, gleichzeitigen und gegensätzlichen Zugriffen auf Daten und Terminen umgegangen werden?

Sollte es nicht möglich sein diese Situationen zu umgehen? Ansonsten sollen diese Probleme vom System gelöst werden, damit unsere Mitarbeiter normal mit der Anwendung arbeiten können.

Welche Form soll die Software haben? Soll es zum Beispiel eine Webapp oder eine Desktopanwendung sein?

Siehe Zusammenspiel mit anderen Systemen

Ein selektiver Import und Export von Daten über lesbare Dateien muss für Backups und zum Datenaustausch möglich sein.

Was soll denn ein selektiver Import sein?

Zugegeben ist in diesem Satz die Formulierung unter Umständen unglücklich gewählt. Es soll möglich sein, dass Datensätze z.B. aus bereits bestehenden Backups wieder in die Anwendung eingespielt werden. Dabei sollen die gesamten betroffenen Datensätze ausgetauscht werden, aber nicht zwingend die gesamte Datenbasis.

Was verstehen Sie unter selektiven Export?

Es soll möglich sein, dass Datensätze ausgewählt werden. Z.B. einzelne Mitglieder oder eine Gruppe an Mitgliedern, die gewisse Kriterien erfüllen.

Was wollen Sie selektieren können, was nicht?

An sich sollen alle Datensätze selektiert werden können, die Mitgliederdaten, Termin- und Buchungsdaten, sowie Rechnungsdaten enthalten. Kurz gesagt alle Daten, die auch für eine sinnvolle Erstellung von Backups benötigt werden.

Wo und wie werden die Backups gespeichert?

Wir haben einige externe Festplatten, die wir für das Überspielen der Backups an den Server anschließen. Die Aufbewahrung erfolgt im Firmentresor im Büro der Geschäftsführung.

Wie oft werden Backups durchgeführt?

Ein mal wöchentlich soll ein neues Backup erstellt werden.

Sollen die Backups automatisch oder manuell durchgeführt werden?

Die Backups sollen automatisch von der Anwendung ausgeführt werden. Der Zeitpunkt des Backups soll von einem Administrator einstellbar sein. Außerdem soll ein Administrator auch jederzeit manuell ein Backup erstellen können.

Sollen im Backup immer nur die Änderungen zum vorherigen Stand gespeichert werden oder soll immer ein vollständiges Backup erstellt werden?

Es sollen immer vollständige Backups erstellt werden.

Wie lange soll ein Backup aufbewahrt werden?

Die Backups sollen jeweils für ein Jahr gespeichert werden.

Gibt es redundante Notfallsysteme?

Nein, es gibt keine zusätzlichen Systeme zum einzigen Server im Keller.

In welche Form wollen Sie die Daten für den Datenaustausch exportieren?

Wir kennen uns mit den technischen Aspekten nicht so tief auf, da dies in unserem alten System nicht in diesem Umfang möglich war. Es sollte aber ein Format sein, dass auch für die Mitarbeiter lesbar sein sollte.

Zu welchen Zwecken betreiben Sie Datenaustausch?

Neben dem Export zur Erzeugung von Sicherungskopien haben wir noch eine andere wichtige Verwendung. Wir wollen verschiedene Analyseanwendungen verwenden, die uns die aktuelle Geschäftssituation darstellen und analysieren kann. Dazu brauchen wir die Möglichkeit die Daten in einer von beiden Seiten unterstützten Form von einem Programm ins andere zu transferieren.

Mit welchen Sicherheitsrichtlinien sollen die Backup-/Datenaustauschdateien versehen werden?

Da die Backups und der Datenaustausch keinerlei Kundenkontakt aufweist, sind bis jetzt keinerlei Sicherheitsvorkehrungen ergriffen worden. Es wäre jedoch gut, wenn nur die Mitarbeiter in der Lage sind, diese Dateien und Daten zu betrachten.

Eine intuitive, leicht bedienbare Benutzeroberfläche setzen wir als selbstverständlich voraus. Es sollen keine besonderen Computerkenntnisse zur Bedienung der Software erforderlich sein.

Was ist Ihre Definition von “keine besonderen Computerkenntnisse“?

Mit einem Firmendurchschnittsalter von 47 Jahren beschäftigen wir statistisch betrachtet überdurchschnittlich viele digitale Quereinsteiger, die noch in einer Zeit ohne Handy

und Computer aufgewachsen sind und daher nicht intuitiv mit der Technologie umgehen können.

Was verstehen Sie unter leicht bedienbar, wollen Sie Icons oder Labeltexte?

Die Bedeutung von Icons hat in unserem alten System einerseits zu Verwirrung gesorgt, aber mittlerweile sind unsere Mitarbeiter daran gewöhnt. Eine ausgewogene und verständliche Kombination aus beiden Teilen wäre hier gefordert. Es wäre gut, dass sie gleiche oder ähnliche Symbole wie im SAP System verwenden.

Muss die Benutzeroberfläche auch barrierefrei sein?

Aktuell beschäftigen wir keine Menschen mit Sehbeeinträchtigung, doch soll dieser Fall nicht ausgeschlossen sein, ist aber aktuell nicht zu fokussieren.

Muss es besondere Farb-Modi für z.B. Farbenblindheit, etc. geben?

Auf Anfrage unserer IT-Abteilung wäre es gut, wenn die Anwendung auch so ein “Dark Theme“ besitzt

## 5.2.2 Anwendungsbereiche

Die Software soll ausschließlich für die Verwaltung von Fahrzeugen, Kunden, Ausrüstung, Fahrzeugstandorte und Angestellten und den damit direkt verbundenen Elementen verwendet werden. Sie soll im Alltag auf Desktop-Rechnern und Laptops eingesetzt werden.

Welches Betriebssystem wird erwünscht oder gefordert?

Da Windows XP doch bereits seit Jahren veraltet ist, wäre es gut, Windows 10 zu verwenden.

Es ist wirklich keine mobile Nutzung erwünscht, um spontan von unterwegs Autos buchen zu können?

Ob eine mobile Anwendung benötigt wird, wird von unserer Marktforschungs-Abteilung momentan evaluiert und kann für dieses Projekt vernachlässigt werden.

Gibt es echtzeitkritische Aspekte in der Anwendung, die besonders hervorgehoben werden müssen?

Das System sollte schon an einem schnellen Ablauf orientiert sein. Es ist nötig, dass die Terminbuchungen und Rechnungsbelege nahezu sofort umgesetzt werden, damit es keine Probleme bei der Terminverarbeitung geben kann. Zusätzlich soll das Buchen von einem Fahrzeug mithilfe eines Reservation-Zeitfensters nur von einer Person auf einmal zugelassen werden (Ähnlich wie zum Beispiel bei Ticketmaster oder Eventim). Das Fenster soll 15 Minuten lang sein, dann soll die Reservierung freigegeben werden, falls sie nicht beendet wurde.

Soll das Programm im Browser aufgerufen werden oder eine Anwendung sein?

Bevor wir uns neben der Desktopanwendung für eine umfangreiche Digitalisierung mit Webanwendung und mobiler Handy-App entscheiden, wollen wir zuerst diese Anwendung im Unternehmensalltag integrieren. Im weiteren Verlauf ist jedoch eine Webanwendung gewünscht. Es wäre also erforderlich, dass einige Vorschläge und Konzepte bzgl. einer Einbindung einer Webanwendung bereits erstellt werden.

Was ist unter Ausrüstung zu verstehen?

Unter Ausrüstung verstehen wir zusätzliches Equipment, welches mit dem Auto gemietet werden kann. Das mieten der Ausrüstung sorgt dabei für keinen Aufpreis. Zur Ausrüstung zählen Dachboxen, Fahrradträger und Hundetransportboxen. Dieses Spektrum wird vorerst nicht erweitert.

### 5.2.3 Zielgruppen, Benutzerrollen und Verantwortlichkeiten

Es soll verschiedene Benutzerrollen geben:

- Organisatorinnen und Organisatoren pflegen die jeweiligen Buchungsdaten und Fahrzeuge.

Wer ist Organisator?

Als Organisator\*innen zählen alle Mitarbeiter der zuständigen Abteilung, also nicht die Personalmitarbeiter. Dazu gehören die Mitarbeiter\*innen, die in den Filialen mit Kundenkontakt arbeiten, da diese die Buchungsdaten eingeben müssen.

Kann jeder Organisator alles?



Organisatoren haben vollen Zugriff auf Buchungs- und Fahrzeugdaten, Lesezugriff auf Rechnungsdaten und keinen Zugriff auf etwaige Personaldaten.

Woraus bestehen die Buchungsdaten?

- Referenz auf das jeweilige Kundenkonto
- Zeitraum der Buchung
- das gebuchte Fahrzeug
- der Standort
- Verweis auf die zum Kundenkonto gehörenden Rechnungsdaten

Wie sollen die Daten gespeichert werden?

Die Daten sollen als zusammengehörender Datensatz wie auch alle anderen Daten gespeichert werden. (siehe oben)

- Personalmitarbeiter pflegen Mitarbeiterdaten im System

Pflegen die Personalmitarbeiter auch ihre eigenen Daten?

Da wir vollstes Vertrauen in unsere Mitarbeiter haben, ist die Bearbeitung der eigenen Mitarbeiterdaten möglich.

Gibt es Einschränkungen, ob bestimmte Daten nicht bearbeitet werden dürfen?

Wichtig ist, dass vergebene IDs, Rechnungsnummern, Buchungsnummern, etc. nach Erstellung und Eintragung nicht mehr geändert werden dürfen und eindeutig sind.

Dürfen Personalmitarbeiter auch andere Daten einsehen, z.B. Buchungsdaten?

Wir arbeiten nach Vorbild aus anderen Firmen nach dem Need-to-know-Prinzip. Somit ist es nicht nötig, dass die Personalmitarbeiter auf andere Datensätze als die Personalstammdaten editierend zugreifen können.

- Eine hauptverantwortliche Person (Administrator) hat Vollzugriff auf sämtliche Daten, vor allem für deren Import und Export sowie deren Backup.

Vergibt der Admin die Rechte für die Anderen? Wenn nein: Wer verteilt sonst die Rechte?

Die Aufgabe der Administratoren ist die umfassende Verwaltung der Software. Dazu gehört die Rollenzuweisung und somit ist ein Administrator auch für die Verteilung der Zugriffsrechte zuständig.

Heißt Vollzugriff auch unbeschränktes ändern?

Vollzugriff bedeutet unbeschränkter Umgang mit allen Daten. Davon ausgeschlossen sind jedoch die bereits erwähnten Daten zur Identifizierung verschiedener Objekte wie Buchungen oder Rechnungen.

Welche Sicherheitsvorkehrungen sollen getroffen werden, sodass kein Datenmissbrauch möglich ist?

Es muss mindestens zwei Administratoren geben und bei Zuweisung oder Änderung von Rechten und Rollen muss ein anderer Administrator dieser Änderung zustimmen, bevor sie umgesetzt wird.

- Es gibt keine Gruppen oder Abteilungen, die verwaltet werden müssen.

Wenn es keine unterschiedlichen Abteilungen gibt, woran wird die Rollenvergabe festgemacht?

Wir möchten, dass die Rollenvergabe händisch erfolgt. Der Administrator soll einzelnen Personen oder einer Mehrfachauswahl die Rollen anpassen können. Wichtig dabei ist, dass es zu Beginn die benötigten Systembenutzer gibt, mit denen man die initiale Rollenvergabe durchführen kann.

Welche Rolle soll Kunden zugewiesen werden?

Für die Verwendung der Desktopanwendung ist es nicht nötig, dass es einen extra Kundenzugriff gibt, doch in Berücksichtigung einer möglichen Webanwendung, wo auch die Mitglieder individuell ihre Buchungen tätigen können, sollten die Mitglieder dann einfach die Rolle Kunde erhalten.

#### 5.2.4 Zusammenspiel mit anderen Systemen

Die Daten über die Angestellten (Gehälter bzw. Löhne, Steuern, Kranken- und Rentenversicherung usw.) werden separat durch ein vorhandenes Personalbuchhaltungsprogramm verwaltet und müssen hier nicht berücksichtigt werden. Die finanztechnischen Daten

werden über unser vorhandenes Finanzsystem erfasst und müssen hier ebenfalls nicht berücksichtigt werden.

Welche Personaldaten werden dann verwaltet?

Die benannten Personaldaten beziehen sich auf die Daten der Mitglieder, also alle Menschen, die unsere Dienstleistung des Car-Sharings nutzen.

Welche Kundendaten werden verwaltet?

- Vor- und Nachname
- Adresse
- Kontaktdaten
- Vertragsdaten (als .pdf)
- Führerscheinkopie (als .pdf)
- bisherige Buchungen
- Zahlungsdaten

Die Software soll aus zwei Teilen bestehen:

- Für die Mitarbeiter im Büro soll eine Desktopanwendung erstellt werden, mit denen die Datenbestände verwaltet werden können. Es sollen auch Buchungen erstellt werden können für Kunden, die keine Online-Buchungen machen wollen und persönlich in der Carsharing-Filiale erscheinen.

Wie soll bei Kunden abgerechnet werden?

Die Abrechnung soll erst nach Abschluss der Fahrt vollständig erstellt werden, es wird also auf eine Voranzahlung verzichtet. Die Kosten bestehen aus einer festgelegten Entsperrgebühr, die für jedes Fahrzeug gleich ist. Diese Entsperrgebühr ist immer fällig, wenn der Termin vorher nicht storniert wurde. Auf diese Gebühr wird nach Abschluss der Fahrt ein Betrag für Dauer und gefahrene Kilometer addiert.

Gibt es Online-Banking, Kartenzahlung, etc.?

Für Online-Buchungen soll natürlich die Möglichkeit bestehen einfach per Kreditkarte oder PayPal zu bezahlen.

Soll für Kunden, die in der Filiale buchen, auch eine Buchung auf Rechnung möglich sein?

In der Filiale ist es nur möglich auf Rechnung zu Zahlen, da die Kosten ja erst nach der Fahrt errechnet werden.

Gibt es noch weitere Zahlungsmethoden, die verwendet werden können?

Für Onlinebuchungen oder in der Filiale soll die Möglichkeit existieren per SEPA-Lastschrift automatisch die Kosten einzubeziehen. Dazu muss offiziell ein Mandat vom Mitglied aus erteilt werden, wie es auch bei anderen Produkten gang und gäbe ist.

- Eine neue Web-Seite soll unseren Online-Kunden ermöglichen, nach einer Authentifizierung alle Standorte anzeigen zu lassen sowie natürlich die dort befindlichen Fahrzeuge, welche für einen anzugebenden Zeitbereich online gebucht werden können.

Soll die Buchung (visuell) identisch zu der Desktop-App stattfinden oder für Webanwendungen angepasst sein?

Da heutzutage eine Webseite modern und ansprechend aussehen muss, ist es denkbar, dass das Design und die Darstellung in der Webanwendung stark von der Desktopanwendung abweichen kann. Die Darstellung der Karte mit den verfügbaren Fahrzeugen soll jedoch gleich sein.

Soll vorher der Zeitbereich angegeben und nur verfügbare Fahrzeuge angezeigt werden oder soll erst ein Fahrzeug und dann der verfügbare Zeitraum gezeigt werden?

Wir denken, dass erst die Auswahl des Zeitraums und dann die Angabe der verfügbaren Fahrzeuge die bessere Wahl ist. Diese Reihenfolge der Darstellung ermöglicht einen einfacheren Überblick über die Daten. Natürlich soll es daneben aber auch möglich sein, dass man in der detaillierten Übersicht der einzelnen Fahrzeuge zu einem Buchungsüberblick gelangt, sollten Kunden ein spezielles Fahrzeug benötigen.

Die Web-Seite soll mit dem ersten Teilauftrag noch nicht programmiert werden, allerdings benötigen wir ein klares Konzept, wie diese Web-Seite realisiert werden soll (Schnittstellen usw.).

Wie detailliert sollen die Mock-ups erstellt werden?

Da aktuell die Desktopanwendung im Hauptfokus liegt, ist ein vollständig ausgebauter Prototyp nicht nötig. Es muss verständlich sein wie die Webanwendung aufgebaut sein soll, wie die Struktur ist und welche Architektur und Struktur darunter angedacht ist, doch die Ausarbeitung irgendwelcher Farbschema oder Button-Designs ist hierfür irrelevant.

Möglichst alle Daten sollen vom alten in das neue System übertragen werden.

Wie sollen die Daten übertragen werden, wenn ein ordentlicher Datenexport im alten System nicht unterstützt wird?

Es scheint so, als müssen diese Daten manuell übertragen werden, was durch unsere Administratoren erledigt werden kann.

### 5.2.5 Produktfunktionen

/LF10/	<p>Der jeweilige Benutzer muss die Möglichkeit haben, über eine grafische Benutzeroberfläche alle für ihn relevanten Daten einfach und übersichtlich zu verwalten.</p> <p>Wie werden Kunden von Mitarbeitern unterschieden? Unterscheidung durch Login?</p> <p>Abgesehen von den vergebenen Rollen, sollen sich die Nutzer im System mittels Benutzername und Kennwort anmelden. Wir verwenden intern IDs für unsere Mitarbeiter, es wäre also gut, wenn wir diese als Benutzername verwenden könnten. Kunden haben keine solchen IDs, also entweder müssten sie auch welche erhalten oder sich mit der E-Mail anmelden.</p> <p>Wie sind die Benutzergruppen definiert?</p> <p>Durch die unterschiedlichen Rechte sollen die unterschiedlichen Nutzergruppen leicht abgewandelte Benutzeroberflächen verwenden können.</p> <p>Sollen firmenspezifische Symbole, Farbmuster, etc. integriert werden oder ist dies eher nebensächlich?</p> <p>Da die Desktopanwendung intern verwendet wird, ist ein Firmen-Farbkonzept eher überflüssig, auch wenn zumindest die Darstellung des Logos wünschenswert ist. Für die Webanwendung dagegen, die direkten Kundenkontakt haben wird, ist die Verwendung eines Farbschemas wichtig, wobei unsere Firmenfarben Schwarz-Gelb sind.</p> <p>Es sollen zahlreiche Konfigurationsdaten gespeichert und beim nächsten Start des Programms verwendet werden (z.B. aktuelle Größe und Position des Fensters). Daneben sollen einige Elemente vor dem Start konfigurierbar sein (z.B. Überschriften, Schriftarten und -größen usw.).</p>
--------	--

/LF10/	<p>Kann die Größe von Überschrift und restlichem Text insgesamt frei gewählt werden oder nur im Zusammenhang? Groß, mittel, klein... (Bei freier Wahl könnte eine kleinere Schriftgröße für die Überschrift gewählt werden, wie für den Text).</p> <p>Um das Gesamterscheinungsbild in einem ästhetischen Rahmen zu halten, sollten die Überschriften nur in Kombination mit den Texten vergrößert werden können. Das hat vor allem den Grund, dass einige unserer Mitarbeiter nicht mehr so gut sehen können und daher eine größere Schrift hilfreich ist.</p> <p>Welche Konfigurationsdaten genau? (Farbschema, etc.?)</p> <ul style="list-style-type: none"><li>• Textgröße</li><li>• Schriftart</li><li>• Dark- oder Lightmodus</li></ul> <p>Sollen die Werte dazu frei auswählbar oder in einem gewissen Rahmen vorgegeben sein?</p> <p>Nein, es soll in allen Fällen eine Auswahl geben. Für die Textgröße soll zwischen klein, normal und groß entschieden werden können. Als Schriftart soll man "Calibri", "Arial", "Times New Roman" und für unsere älteren Mitarbeiter auch "Old English Text MT" auswählen können. Beim Farb-Modus gibt es nur die Option zwischen "Hell" oder "Dunkel" zu wählen.</p> <p>Sollen die Konfigurationsdaten anwendungsspezifisch oder benutzerspezifisch sein?</p> <p>Eigentlich sitzt in unseren Filialen immer der gleiche Mitarbeiter am eigenen Rechner, daher reicht es, wenn die Einstellungen für die installierte Anwendung gespeichert werden.</p>
--------	--

/LF20/	<p>Verwaltet werden sollen Mitarbeiter, Fahrzeuge, Standorte, Kunden, Buchungen, Rechnungen, Änderungen, Stornierungen, Mahnungen usw.</p> <p>Welche Gruppe hat Zugriff auf welche Daten?</p> <ul style="list-style-type: none"><li>• Administrator: Hat Zugriff auf alle Datenbestände und Funktionen</li><li>• Organisator: Fahrzeuge, Kunden, Buchungen, Rechnungen</li><li>• Personalmitarbeiter: Personal/Kundenstammdaten</li><li>• Kunde: aktuell noch keine Zugriffe, später aber Fahrzeuge, Standorte, eigene Buchungen</li></ul> <p>Welche Daten soll für Mitarbeiter, Fahrzeuge, Standorte, Kunden, Buchungen, Rechnungen, Änderungen, Stornierungen und Mahnungen erhoben werden?</p> <ul style="list-style-type: none"><li>• Mitarbeiter: nicht nötig, dafür haben wir unser eigenes System</li><li>• Fahrzeug: Kennzeichen, Typ/Bezeichnung, Kategorie, Farbe, Baujahr, Buchungen</li><li>• Standort: Adresse, ansässige Fahrzeuge</li><li>• Kunde: siehe 5.2.4</li><li>• Buchung: siehe 5.2.3</li><li>• Rechnung: Inhalt der Buchung (Fahrzeug, Dauer), Gesamtkosten, Steuern, Rechnungsadresse, Rechnungsnummer, gefahrene Kilometer</li><li>• Änderung: Benutzer, der die Änderung vorgenommen hat, betroffene Buchung, Inhalt der Änderung, Zeitpunkt der Änderung</li><li>• Stornierung: Benutzer, der die Stornierung vorgenommen hat, betroffene Buchung, Zeitpunkt der Stornierung, Grund der Stornierung</li><li>• Mahnung: Rechnung</li></ul>
--------	---



---

/LF20/	<p data-bbox="352 300 1050 338">Worin unterscheiden sich Buchung und Rechnung?</p> <p data-bbox="352 347 1355 528">Eine Buchung ist die Terminvereinbarung, die zum Ausleihen eines Autos getroffen wird und auch im System eingetragen wird, um das betroffene Fahrzeug für einen bestimmten Zeitraum zu blockieren. Eine Rechnung ist die zahlungspflichtige Aufforderung nach Abschluss der Leihdauer.</p> <p data-bbox="352 537 1114 575">Wird eine Rechnung aus den Buchungsdaten generiert?</p> <p data-bbox="352 584 1355 716">Die Rechnung soll aus einer Kombination von Buchungsdaten und den Daten der gefahrenen Strecke und dem verbrauchten Treibstoff erstellt werden.</p> <p data-bbox="352 725 1114 763">Gibt es eine Mindestlänge für einen gebuchten Termin?</p> <p data-bbox="352 772 1118 810">Die Mindestlänge für einen Termin beträgt 15 Minuten.</p> <p data-bbox="352 819 1123 857">Gibt es eine Maximallänge für einen gebuchten Termin?</p> <p data-bbox="352 866 1241 904">Die Maximallänge soll technisch auf 30 Tage beschränkt werden.</p> <p data-bbox="352 913 1355 1001">Gibt es Rabatte für längere Buchungen? Ab wann gilt eine Buchung als länger?</p> <p data-bbox="352 1010 1355 1240">Es ist bei uns schon öfter vorgekommen, dass Kunden ein Fahrzeug (meistens Kastenwagen) für länger als einen Tag brauchen, da sie beispielsweise Umziehen und keine Firma in Auftrag geben wollen. In solchen Fällen wollen wir einen Rabatt ermöglichen, der am Ende der Abrechnung erstattet wird. Die genaue Preisliste kann gerne nachgeliefert werden.</p> <p data-bbox="352 1249 1098 1288">Soll es auch Eventwochen und Rabattaktionen geben?</p> <p data-bbox="352 1296 1355 1429">Ja, es soll zum Beispiel zu besonderen Anlässen Rabattaktionen durchgeführt werden. Solche Ereignisse wären zum Beispiel Firmenjahrestage, besondere Mitgliederanzahlen und kulturelle Ereignisse.</p> <p data-bbox="352 1438 1355 1525">Es muss möglich sein, jederzeit erkennen zu können, welche angestellte Person einen Datensatz angelegt, geändert oder gelöscht hat.</p> <p data-bbox="352 1534 1161 1572">Müssen alle Änderungen sichtbar sein oder nur die Letzte?</p> <p data-bbox="352 1581 1331 1619">Ja, es sollen alle Änderungen sichtbar sein wie in einem Ticket-System.</p>
--------	--

---

---

/LF30/	<p>Buchungen haben eine Start- und einen Endtermin, Terminüberschneidungen müssen vermieden werden, um die Verfügbarkeit sicherzustellen.</p> <p>Was, wenn der vorherige Kunde sich verspätet? Wird ein Puffer eingeplant?</p> <p>Da wir den maximalen Nutzen aus der Autovermietung ziehen wollen, soll kein Zeitpuffer berücksichtigt werden. Kunden, die ihren gebuchten Rahmen überziehen werden mit Strafgeldern abgeschreckt. Sollte jedoch absehbar sein, dass sich ein Kunde verspätet und kein Termin ist danach gebucht, kann der Zeitraum auch spontan durch einen Anruf bei uns in einer Filiale oder (später) online verlängert werden. Sollte ein anderer Kunde auf das Auto warten, soll ihm eine Alternative angeboten werden.</p> <p>Gibt es Wartungsfenster zwischen Buchungen für Reinigungen, etc.?</p> <p>Nicht direkt. Wir gehen davon aus und haben diese Vorgabe auch in unseren AGB festgehalten, dass die Kunden sorgsam mit den Fahrzeugen umgehen. Wir haben Mitarbeiterteams, die zwischen den Standorten pendeln und einmal pro Woche die Fahrzeuge reinigen, wenn sie mal vor Ort sind. Dazu setzen die Organisatoren einfach einen Blocker in die Terminbuchung der Fahrzeuge, wenn eine Reinigung geplant ist.</p> <p>Was, wenn Mängel während der Fahrt auftreten?</p> <p>Mängel vor oder nach der Fahrt sind unverzüglich zu melden. Sollte ein Kunde einen Mangel nicht melden und der nächste Benutzer entdeckt ihn, wird die Schuld dafür automatisch dem Vormieter zugeschoben. Normale Gebrauchsspuren sind unvermeidlich, aber mutwilliges beschädigen oder selbstverschuldete Schäden werden im Nachhinein in Rechnung gestellt.</p> <p>Sollen nicht verfügbare Fahrzeuge als solche gekennzeichnet werden, z.B. durch eine graue Darstellung?</p> <p>Gerade nicht verfügbare Fahrzeuge sollen einfach ausgegraut dargestellt werden, man soll jedoch weiterhin Termine buchen können, die außerhalb der verhinderten Zeitspanne liegen.</p>
--------	--

---

/LF30/	<p>Wie soll die Befüllung der Tanks gewährleistet werden?</p> <p>Dafür sind die Kunden verantwortlich. Jedes Auto besitzt eine Tankkarte, mit der der Tank wieder aufgefüllt werden kann. Es ist in der Verantwortung der Kunden den Tank mindestens halbvoll zurückzugeben. Die Tankkosten werden letztendlich auf die gefahrenen Kilometer heruntergerechnet und in der Rechnung verrechnet.</p> <p>Kann es passieren, dass der Standort des Fahrzeuges geändert werden muss?</p> <p>Ja, das kann passieren. Wenn sich zu viele Fahrzeuge an einem Standort befinden, sollten diese wieder annähernd gleich auf andere Standorte verteilt werden. Dazu reicht das Einplanen eines kleinen Blockers im Buchungskalender.</p>
/LF40/	<p>Unsere Kunden haben neben ihren Kontaktdaten auch Vertragsunterlagen für die Teilnahme am Car-Sharing, die die Höhe des Eigenanteils für einen Schadensfall der Versicherung sowie die Höhe der Teilnahme-Kaution enthält. Diese Vertragsunterlagen werden von uns eingescannt und sollen als Dokument mit den Kundendaten gespeichert werden. Daneben wird jedem Kunden eine Karte zum Öffnen und Schließen der Fahrzeuge ausgehändigt.</p> <p>Wie werden Schließberechtigungen verwaltet?</p> <p>Am Auto soll ein Kartenscanner vorliegen, der die Karte einscannet. Im System ist ja bekannt welche Termine aktuell gebucht sind, wenn die präsentierte Karte mit dem Kunden, der den Termin gebucht hat, übereinstimmt, dann wird das Fahrzeug automatisch entriegelt. Während der gebuchten Zeit kann das Auto normal verschlossen und auch wieder mit der Karte entsperrt werden.</p> <p>Ist die Karte spezifisch für das Fahrzeug?</p> <p>Nein, die Karte ist kundenspezifisch.</p> <p>Welche Daten sind auf der Kundenkarte gespeichert?</p> <p>Es muss auf der Karte ein eindeutiges Merkmal zur Identifikation des Kunden geben, zum Beispiel eine Kundennummer oder ähnliches. Weiterhin muss noch eine Sicherheitsvorkehrung existieren, die das einfache Fälschen der Karten verhindert, auch wenn die Freigabe vollständig systemseitig stattfindet.</p>

/LF40/	<p>Wo und in welcher Form sind die Kundendaten gespeichert? Magnetband, Chip oder QR-Code?</p> <p>Die Daten sollen auf einem Chip gespeichert sein. Das Kartenlesegerät befindet sich im Auto und die Karte muss im verschlossenen Zustand von außen präsentiert werden.</p> <p>Sollen die Daten auf der Karte bearbeitet werden können?</p> <p>Nein, da mit der Karte ausschließlich der Kunde identifiziert werden soll, ist es nicht nötig, dass diese Daten bearbeitet werden.</p> <p>Wie lange müssen die Vertragsunterlagen und Buchungsbelege aufbewahrt werden?</p> <p>Buchungen müssen nur so lange aufbewahrt werden, bis der Buchungstermin beendet ist. Vertragsunterlagen und Rechnungsbelege müssen jedoch 10 Jahre lang vorliegen.</p> <p>Welche Art der Langzeitspeicherung gibt es? Magnetband, Festplatten, etc.?</p> <p>Bis jetzt haben wir die Speicherung mittels Festplatten durchgeführt. Es wäre gut, wenn diese Art beibehalten werden könnte.</p> <p>Ein kleiner Prozessor im Fahrzeug sendet nach Fahrtende (Terminende) die exakte Start- und Ende-Zeit sowie die gefahrenen Kilometer an einen Server. Diese Daten sollen von dem Server nach Buchungsende geholt und zur Berechnung der Kosten für die Buchung (und somit für die Rechnung) verwendet werden.</p> <p>Soll dafür http oder ein eigenes Protokoll verwendet werden? Muss Verschlüsselung beachtet werden?</p> <p>Es sollte auf alle Fälle ein gewisses Maß an Sicherheit vorhanden sein. Die technische Umsetzung würden wir ihnen überlassen.</p> <p>Was passiert, wenn das Fahrzeug zu Terminende noch fährt?</p> <p>Es ist natürlich sehr ungünstig, dass das Auto direkt auf der Straße anhält, daher soll der Fahrbetrieb normal weitergehen, doch werden für die Zeitüberschreitungen zusätzliche Gebühren erhoben. Damit der Fahrer informiert wird, dass er das gebuchte Terminfenster überschritten hat, soll eine Warnleuchte in der Fahrerkabine angehen.</p>
--------	--

/LF50/	<p>Die Fahrzeuge selbst gehören unterschiedlichen Kategorien an: Kleinfahrzeuge, Mittelklassefahrzeuge, gehobene Mittelklasse und Transportfahrzeuge.</p> <p>Allen Kategorien sind eine bestimmte Höhe der Stunden-Mietpauschale und die Kosten pro gefahrenem km zugeordnet. Alle Werte sollen konfigurierbar sein.</p> <p>Wie sollen die Daten konfigurierbar bleiben? Sollen Preisänderungen über die Datenbankebene oder mittels GUI erfolgen?</p> <p>Die Preisänderung soll auf Datenbankebene mithilfe einer Konfigurationsdatei geschehen.</p> <p>Gibt es Events wie Angebotswochen?</p> <p>Dies ist in Zukunft auf jeden Fall geplant, jedoch Bedarf dies auch nur einer Änderung der Preise in der Konfigurationsdatei.</p> <p>Können Mitglieder Prämien sammeln, die zu Vergünstigungen oder ähnliches führen?</p> <p>Unsere Kunden können für eine Fahrt im Wert von mindestens 10 € Payback-Punkte sammeln. Dies ist jedoch unabhängig von der zu implementierenden Anwendung.</p> <p>Ist diese Stunden-Mietpauschale nur für die Fahrzeugart treffend oder unterscheidet sie sich auch innerhalb dieser Kategorien je nach Auto?</p> <p>Die Pauschale ist für alle Fahrzeuge innerhalb einer Kategorie gleich.</p> <p>Alle Fahrzeuge werden regelmäßig von Fremdfirmen gewartet. Die entsprechenden Dienstleistungen sollen den Fahrzeugen chronologisch zugeordnet werden.</p> <p>Soll es eine Rechnungsverwaltung geben? Werden die Rechnungsdaten im gleichen System gespeichert oder müssen diese separat verwaltet werden?</p> <p>Die Rechnungsdaten werden separat verarbeitet. Es soll lediglich vermerkt werden, welcher Service wann durchgeführt wurde.</p> <p>Was ist mit chronologischer Zuordnung gemeint?</p> <p>Die Dienstleistungen sollen nach Datum geordnet angezeigt werden. Die aktuellste Dienstleistung soll dabei zuerst angezeigt werden.</p>
--------	---

/LF60/	<p>Einem Standort können ein oder mehrere Fahrzeuge zugeordnet sein. Ein Fahrzeug ist immer nur einem Standort zugeordnet.</p> <p>Welche Standorte gibt es?</p> <p>Da wir planen großflächig nach Fidschi zu expandieren muss das Anlegen neuer Standorte in der Anwendung möglich sein. Die bereits bestehenden Standorte werden dann bei der Inbetriebnahme von unseren Praktikanten angelegt und müssen demzufolge nicht betrachtet werden.</p> <p>Gibt es ein (Live-) Fahrzeugtracking?</p> <p>Da dies ein Eingriff in die Privatsphäre unserer Kunden wäre möchten wir auf diese Funktion verzichten.</p>
/LF70/	<p>Nach jeder Fahrt werden sofort die Rechnungen erstellt und dem Kunden per E-Mail zugesandt.</p> <p>Könnten bei der Sendung von Rechnungsdaten per E-Mail als Klartext eventuell DSGVO-Probleme auftreten?</p> <p>Ja, aber da unser Firmensitz in Dubai liegt ist uns das egal. Wir machen das einfach so wie Telegram.</p> <p>Was ist die Zeitraumdefinition für “sofort“?</p> <p>Wenn der Kunde die Fahrt beendet soll automatisiert eine Rechnung erstellt werden, welche direkt an den Kunden geschickt wird. Die Rechnung sollte maximal 10 Minuten nach dem Ende der Fahrt beim Kunden ankommen.</p>

/LF70/	<p>Die Bezahlung der Rechnungen erfolgt über Bankeinzug, was durch das Finanzbuchhaltungssystem (FBH) erledigt wird und hier nicht betrachtet werden muss. Allerdings muss über die vorhandene Schnittstelle des FBH der Stand der Rechnungsbegleichung abgefragt werden, damit über das neue System erkennbar ist, ob und wann eine Rechnung bezahlt wurde.</p> <p>Wie regelmäßig soll die Abfrage stattfinden? Ein mal pro Tag, mehrmals täglich?</p> <p>Es soll einen Aktualisierungsknopf geben, welcher eine Anfrage an die Schnittstelle sendet und die Daten aktualisiert. Zusätzlich sollen alle 30 Minuten die Daten automatisch abgefragt werden.</p> <p>Gibt es für die Rechnungen bereits Rechnungsnummern, oder müssen diese erst erstellt werden? Welches System wird für die Nummernvergabe verwendet?</p> <p>Die Vergabe der Rechnungsnummern geschieht automatisch durch das FBH und muss daher nicht betrachtet werden.</p>
/LF80/	<p>Buchungen können bis 10 Stunden vor Antritt der Fahrt storniert werden.</p> <p>Wie soll die Meldung des Systems aussehen, wenn die 10h-Marke unterschritten wird?</p> <p>Es soll ein Pop-up-Fenster geöffnet werden, welches den Nutzer darüber informiert.</p> <p>Soll die Funktion "Stornieren" nicht mehr angezeigt werden, falls die 10h-Marke unterschritten wird?</p> <p>Der Knopf zum Stornieren soll weiterhin angezeigt werden, jedoch soll er entweder deaktiviert sein oder beim Betätigen soll eine Fehlermeldung per Pop-up angezeigt werden.</p> <p>Impliziert diese Einschränkung auch, dass Buchungen unter der 10h-Grenze nicht mehr möglich sind oder einfach nur nicht storniert werden können?</p> <p>Ja. Da die Buchung von unseren Mitarbeitern freigeschaltet werden muss, ist auch hierfür eine Grenze von 10h einzubauen.</p>

/LF90/	<p>Zur einfacheren Eingabe der Daten soll es Auswahllisten für deren Eigenschaften geben, wo immer es möglich ist. Die Auswahllisten sollen auf einfache Weise erweiterbar sein.</p> <p>Welche Eingabefelder gibt es?</p> <p>Eingabefelder treten bei jeder Interaktion auf. Bei der Auswahl eines Buchungszeitraums, bei der Fahrzeug- oder Standortwahl, bei Filteroptionen, bei der Konfiguration der Anwendung. Bei dieser Auflistung sind auf alle Fälle Auswahllisten möglich. Weiterhin gibt es Eingabefelder bei der Eingabe von Kundendaten und auch die Administratoren müssen bei der Rollenvergabe Eingabefelder ausfüllen.</p> <p>Wie soll die Listenerweiterung vorgenommen werden? Reicht ein einfacher "Edit"-Knopf?</p> <p>Es soll eine Bearbeitungsansicht für das komplette Formular geben, welche über einen "Edit"-Knopf erreichbar sein soll. In dieser Ansicht soll für jede Auswahlliste ein Textinput mit einem Knopf zum Hinzufügen bereitgestellt werden.</p> <p>Wer hat die Berechtigungen, diese Liste zu erweitern?</p> <p>Die Berechtigungen für die Erweiterung der Auswahllisten sind auf die Administratoren beschränkt.</p> <p>Ist eine Mehrfachauswahl oder Mehrfachbearbeitung möglich?</p> <p>Nein. Eine Mehrfachauswahl soll nicht möglich sein.</p>
/LF100/	<p>Sämtlichen Elementen sollen mehrere Bilder mit Titel zugeordnet werden können, die zentral auf einem Verzeichnis liegen sollen</p> <p>Welche Vorgaben gibt es für die Bilder? Bildformat, Dateiformat, etc.</p> <p>Die Bilder sollen im JPEG- oder PNG-Format vorliegen.</p> <p>Wer hat die Berechtigungen, um Bilder hochzuladen?</p> <p>Das sollen nur die Administratoren können.</p> <p>Ist ein einheitliches Design für die Seiten vorgegeben?</p> <p>Das Design wird nicht vorgegeben aber es sollte auf allen Seiten einheitlich sein.</p> <p>Gibt es eine maximale Anzahl an Bildern pro Element?</p> <p>Die Anzahl der Bilder pro Element soll auf 20 beschränkt sein.</p> <p>Gibt es eine maximale Dateigröße für die Bilder, um Ladezeiten und Speicher einzugrenzen?</p> <p>Die Dateigröße ist auf 10 MB pro Bild beschränkt.</p>



### 5.2.6 Produktdaten

/LD10/	<p>Die Daten sollen zunächst in einer zentralen Datenbasis (lesbare Dateien) abgespeichert und später in eine Datenbank überführt werden.</p> <p>Wo ist diese Datenbasis, wie sieht diese Datenbasis aus?</p> <p>Die Datenbasis besteht aus CSV-Dateien, welche vorerst lokal bereitgestellt werden.</p> <p>Welches Dateiformat wird benötigt?</p> <p>Als Dateiformat wird CSV benötigt.</p> <p>Was bedeutet lesbar?</p> <p>Lesbar bedeutet in diesem Zusammenhang, dass die Leserechte für diese Dateien nicht eingeschränkt beziehungsweise für die Anwendung freigegeben sind.</p> <p>Um welche Daten handelt es sich?</p> <p>Bei den Daten handelt es sich um die Daten der Mitglieder sowie deren Buchungen.</p> <p>In was für eine Datenbank sollen die Daten überführt werden?</p> <p>Die Daten sollen in Zukunft in einer SQL-Datenbank gespeichert werden. Jedoch ist dies noch nicht in naher Zukunft geplant und kann deshalb vorerst vernachlässigt werden.</p> <p>Wann ist später, sollen Vorbereitungen für die Überführung getroffen werden?</p> <p>Die Überführung soll kurz vor der Inbetriebnahme der Anwendung geschehen, jedoch sehen wir keine Notwendigkeit dafür Vorbereitungen zu treffen, da unsere IT-Abteilung kompetent genug ist dies eigenständig umzusetzen.</p> <p>Wie sieht die Backupstrategie für Datenbasis und Datenbank aus?</p> <p>Um die Datenbank kümmert sich wie bereits erwähnt unsere IT-Abteilung. Da die Datenbasis sowieso abgelöst werden soll, müssen hier keine Backups erstellt werden.</p> <p>Gibt es einen Multiuser-Kontext für die "Datenbasis"? Wie sollen gleichzeitige Zugriffe geregelt sein?</p> <p>Nein. Da die Datenbasis nur lokal implementiert wird, muss vorerst kein Multiuser-Kontext beachtet werden.</p>
--------	---

### 5.2.7 Produktleistungen

/LL10/	<p>Die Anzahl der zu verwaltenden Elemente wird auf ca. 100.000 geschätzt. Kann mit der Elementzahl in den lesbaren Dateien der zentralen Datenbasis sinnvoll umgegangen werden?</p> <p>Die zentrale Datenbasis ist für eine sehr große Anzahl von Elementen ausgelegt, damit auch im Falle eines großen Wachstums der Datenmenge weiterhin ein leichter Umgang mit den Daten möglich ist.</p> <p>Ist das eine Schätzung für den “aktuellen“ Satz? Wird das zukünftige Wachstum beachtet? Werden es signifikant mehr?</p> <p>Ja. Die 100.000 Elemente sind nur eine Schätzung. Bei der letzten Erhebung, welche vor einem Monat stattgefunden hat, waren es 96.690 Elemente. Unser Unternehmen rechnet mit einem massiven Wachstum, sobald die neue Anwendung in Betrieb genommen wurde.</p>
/LL20/	<p>Um bei HW- und SW-Anschaffungen und -neuerungen flexibel zu bleiben, ist auf Plattformunabhängigkeit besonders zu achten.</p> <p>Gibt es Vorzüge wie die Plattformunabhängigkeit angegangen werden soll?</p> <p>In Zusammenhang mit der Plattformunabhängigkeit bevorzugen wir eine Java-Applikation, da Java auf allen Firmensystemen bereits installiert ist. Soll die Applikation auch mobil genutzt werden? Wird Kompatibilität für Mobilgeräte benötigt?</p> <p>Nein. Unsere Firma stellt momentan keine mobilen Endgeräte für die Mitarbeiter zur Verfügung, sondern nur Desktop-Rechner. Daher ist Kompatibilität mit mobilen Endgeräten nicht gefordert.</p> <p>Ist eine Cloudlösung/ oder -speicherung denkbar?</p> <p>Nein. Unser altes Personal ist im Umgang mit Cloudlösungen nicht geschult. Das Internet ist für uns alle Neuland.</p>

### 5.2.8 Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität	X			
Zuverlässigkeit		X		
Effizienz		X		
Benutzbarkeit (auch Gestaltung)	X			
Wartbarkeit			X	
Übertragbarkeit (Portabilität)			X	

Gibt es für die Zuverlässigkeiten bestimmten Konventionen, Standards oder Vorschriften, die eingehalten werden müssen? SLA?

Nein. Wir sind zufrieden, wenn die Anwendung ohne Störungen verwendet werden kann.

Was bedeutet gute Funktionalität, was soll dabei beachtet werden? Stabilität, unkomplizierte Bedienbarkeit, besonders viele Funktionen oder nur die minimal benötigten Funktionen?

Als funktional betrachten wir eine Anwendung, welche unkompliziert zu bedienen ist und mit allen Anwendungsszenarien umgehen kann.

Was macht die Anwendung zuverlässig? Stabilität oder Fehlertoleranz?

Die Anwendung soll einfach nicht abstürzen. Dies bezieht sich natürlich auch auf den Umgang mit fehlerhaften Eingaben.

Was macht die Anwendung effizient? Geschwindigkeit oder sparsamer Ressourcenumgang?

Die Effizienz bezieht sich auf sparsamen Ressourcenumgang. In diesem Zusammenhang soll das Speichern unnötiger Daten vermieden werden.

Benutzbarkeit soll sehr gut sein, welche Anforderungen stehen dahinter? Durch was wird das Programm leicht bedienbar? Mit welchen Hilfsmitteln kann die Bedienbarkeit erleichtert werden (Shortcuts, Gesten)?

Alle Funktionen des Programms sollen mit maximal 3 Klicks erreichbar sein und es soll auf Transaktionen verzichtet werden. Shortcuts werden nicht benötigt.

Gestaltung soll sehr gut sein? Ist das nötig, wenn in den Büros eigentlich nur die Mitarbeiter arbeiten und die Kunden gar nicht in den Kontakt mit der Anwendung treten?

Die Bedienung der SAP GUI bereitet unseren Mitarbeitern momentan große Kopfschmerzen. Daher wünschen wir uns eine ansprechendere Benutzeroberfläche mit vereinfachter Bedienung, um die Mitarbeiterzufriedenheit zu erhöhen.

Portabilität ist normal relevant, aber die Plattformunabhängigkeit ist besonders zu beachten? Wie ist das zu verstehen?

Die Anwendung soll plattformunabhängig auf allen gängigen Betriebssystemen eingesetzt werden können. Außerdem kann es in seltenen Fällen vorkommen, dass die Anwendung auf ein anderes System übertragen werden soll. Allerdings muss kein besonderer Fokus auf die Erleichterung dieser Übertragung gelegt werden.

## 6 Use-Case Diagramm

Im folgenden Kapitel werden die Use-Cases und Akteure der zu implementierenden Anwendung zuerst aufgelistet und anschließend genauer erläutert. Danach folgen die auf den Use-Cases und Akteuren beruhenden Diagramme.

### 6.1 Vorüberlegung

1. Buchung verwalten
  - a) Buchung eines Termins
  - b) gebuchten Termin bearbeiten
  - c) Stornierung eines Termins
  - d) Buchung einsehen
2. Kunde verwalten
  - a) neuen Kunden Anlegen
  - b) bestehenden Kunden löschen
  - c) bestehenden Kunden bearbeiten
  - d) bestehenden Kunden einsehen
3. Mitarbeiter verwalten
  - a) neuen Mitarbeiter anlegen
  - b) bestehenden Mitarbeiter löschen
  - c) bestehenden Mitarbeiter einsehen
  - d) Mitarbeiterrolle bearbeiten
4. Fahrzeug verwalten
  - a) neues Fahrzeug anlegen
  - b) bestehendes Fahrzeug bearbeiten
  - c) bestehendes Fahrzeug löschen
  - d) bestehendes Fahrzeug einsehen
  - e) Wartungstermin festlegen

- f) Fahrzeugbild hochzuladen
- g) Standortveränderung eines Fahrzeugs einplanen
- 5. Rabattaktion verwalten
  - a) neue Rabattaktion anlegen
  - b) bestehende Rabattaktion löschen
  - c) bestehende Rabattaktion einsehen
- 6. Auswahlliste bearbeiten
- 7. Back-Up verwalten
  - a) neues Back-Up erstellen
  - b) Daten importieren
  - c) Daten exportieren
- 8. Standort verwalten
  - a) neuen Standort anlegen
  - b) bestehenden Standort bearbeiten
  - c) bestehenden Standort löschen
  - d) bestehenden Standort einsehen
- 9. Ausrüstung verwalten
  - a) neuen Ausrüstungsgegenstand anlegen
  - b) bestehenden Ausrüstungsgegenstand bearbeiten
  - c) bestehenden Ausrüstungsgegenstand löschen
  - d) bestehenden Ausrüstungsgegenstand einsehen
- 10. Konfiguration der Benutzeroberfläche ändern
- 11. Rechnung verwalten
  - a) neue Rechnung anlegen
  - b) bestehende Rechnung archivieren
  - c) bestehende Rechnung einsehen
- 12. Mahnung versenden
- 13. In System einloggen

## 6.2 Use-Case Analyse

### Akteure:

- **Nutzer:** Der Nutzer ist ein Akteur, der im Rahmen von konzeptionellen Vorüberlegungen entstanden ist. An sich gibt es keinen 'Nutzer' im System, doch beschreibt dieser Akteur die Use-Cases, die sowohl von 'Personalmitarbeiter' als auch von 'Organisator' und 'Administrator' durchgeführt werden.
- **Personalmitarbeiter:** Ein Personalmitarbeiter ist ein Angestellter des Car-Sharing-Unternehmens, welcher sich um die Personal-Verwaltung kümmert. Hauptsächlich arbeitet er mit dem zusätzlichen Verwaltungssystem, doch hat er in der Desktopanwendung die Aufgabe der Verwaltung aller Mitarbeiter und Kundendaten.
- **Organisator:** Organisatoren sind die Mitarbeiter des Unternehmens, die aktiv in der Filiale arbeiten und Kundenkontakt pflegen. Organisatoren bearbeiten alle Anfragen von Kunden in der Filiale und ermöglichen einen ununterbrochenen Geschäftsablauf.
- **Administrator:** Administratoren sind für die technischen Aspekte der Firmenarbeit und das Funktionieren der Systeme zuständig. Sollten Probleme und Fehler auftreten, so ist es die Aufgabe der Administratoren diese Fehler zu beheben. Die Datensicherung fällt auch in diesen Bereich. Weiterhin sollen Administratoren Zugriff auf alle anderen Daten haben und auch die Rollen der Mitarbeiter verwalten.
- **Kunde:** Den Kunden betrifft im Konzept der Desktopanwendung keine systembezogenen Bedeutung. In Zukunft soll der Kunde mithilfe einer Webanwendung seine Termine online vereinbaren können. Diese Webanwendung existiert momentan jedoch noch nicht und deren Implementierung ist auch noch nicht geplant. Daher müssen die Kunden zur Terminvereinbarung in eine Filiale kommen, wo ein Organisator die Buchung in das System einträgt. Da der Organisator ohne die Interaktion mit dem Kunden bestimmte Use-Cases, wie zum Beispiel die Terminvereinbarung, nicht erfüllen kann, wird auch der Kunde in dieser Analyse betrachtet.
- **Werkstatt:** Eine Werkstatt hat keinen direkten Zugriff auf die Anwendung, jedoch wird die Werkstatt, ähnlich wie der Kunde, für den Datenfluss bestimmter Use-Cases benötigt. Daher wird die Werkstatt in dieser Analyse ebenfalls erwähnt.

**Use-Cases:**

Die Car-Sharing-Anwendung, welche im Rahmen des Programmentwurfs erstellt werden soll ist ausschließlich eine Verwaltungssoftware. Ein Großteil der Interaktionsmöglichkeiten mit dem Programm ist das Erstellen und Bearbeiten von verschiedensten Datensätzen. In den Tabellen 6.1, 6.2 und 6.3 von Seite 43 bis 45 erfolgt zu jedem erstellten Use-Case eine wörtliche Erklärung. Wichtig zu betrachten ist dabei, dass alle aufgeführten Verwaltungsaufgaben mit Ausnahme einiger Fälle auf dem gleichen Konzept basieren: neuen Datensatz erstellen, bestehenden Datensatz bearbeiten, bestehenden Datensatz löschen oder bestehenden Datensatz einsehen.

Die analytische Darstellung der Use-Cases in Diagrammform erfolgt später in zwei Schritten. Der erste Ansatz ist ein Überblick über die grob definierten Use-Cases. Die erwähnten Use-Cases, die 'verwalten' enthalten sind eine Zusammenfassung aus den Unterfunktionen 'anlegen, bearbeiten, einsehen und löschen'. In einigen Fällen kommen weitere Unter-Use-Cases hinzu. Exemplarisch dafür wird ein zweites Diagramm erstellt, welches diese Aufteilung genauer beschreiben wird. Die Erstellung von allen auftretenden Hierarchiestufen der Use-Case-Diagramme ist im Rahmen des Programmentwurfs nicht zielführend, müsste in einem realen Projekt aber vollständig ausgebaut werden.



Use-Case 1:	Ein Kunde kann vor Ort in der Filiale einen Termin buchen. Dazu wird der Terminwunsch des Kunden entgegengenommen und der Filialmitarbeiter trägt den Termin als Buchung ins System ein. Zusätzlich ist es nötig, bestehende Buchungen bearbeiten zu können, sollte sich z.B. der Zeitraum der Buchung auf Wunsch des Kunden ändern. Es existiert auch der Fall, dass eine Buchung storniert wird, was bis zu 10h vor dem gebuchten Termin möglich ist. Die Mitarbeiter sollen natürlich auch bestehende Buchungen einsehen können
Use-Case 2	In Anlehnung an die Erkenntnis, dass alle Verwaltungsaufgaben gleiche Grundaufgaben abdecken müssen, trifft dies auch auf die Verwaltung der Kundendatensätze zu. Ein Unternehmen hat immer einen dynamischen Kundenstamm. Es kommen neue Kunden hinzu, bestehende Kunden benötigen eine Datenanpassung (z.B. bei Änderung von Kontakt- oder Addressinformationen), die Daten müssen für die Mitarbeiter einsehbar sein und auf Kundenwunsch kann der Kunden-Account auch geschlossen werden.
Use-Case 3	Neben den Kunden muss es auch einen Datenstamm mit Mitarbeiterdaten geben. Im Lastenheft wurde hervorgehoben, dass die Datenverwaltung der Mitarbeiter (Gehalt, Zahlungsinformationen, etc.) in einem gesonderten System stattfindet. Bei der Mitarbeiterverwaltung in der Desktopanwendung ist ausschließlich die Verwaltung auf Rollenverteilung vorgesehen. Trotzdem werden dafür die Mitarbeiter auch in diesem System als Datensatz benötigt, aber in einem viel geringeren Umfang als im anderen System.

Tabelle 6.1: Use-Cases

Use-Case 4	Die 'Vermietung' von Fahrzeugen ist der Kernbestandteil des Business-Konzepts des Car-Sharing-Unternehmens. Aus diesem Grund ist der Umfang der Verwaltung von Fahrzeugen umfangreicher als bei anderen Verwaltungsobjekten. Neben den mittlerweile als Standard-Aufgaben klassifizierten Aufgaben benötigen Fahrzeuge weiterhin Wartungstermine (beispielsweise für Reparaturen oder Reifenwechsel). Die Fahrzeuge sind an Standorten abgestellt. Sollten sich zu viele Fahrzeuge an einem Standort befinden, so müssen die Fahrzeuge wieder auf andere Standorte verteilt werden. Schließlich ist noch vorgesehen, dass jeder Fahrzeug-Datensatz Bilder enthält, die das Fahrzeug zeigen. Auch diese Bilder müssen im System hochgeladen werden.
Use-Case 5	Aus verschiedensten Anlässen sollen Rabattaktionen möglich sein. Diese Rabattaktionen müssen grafisch dargestellt werden, sowie müssen alle damit verbundenen Konditionen einsehbar sein. Sollte eine Rabattaktion auslaufen, müssen die damit verbundenen Informationen auch wieder aus dem System entfernt werden.
Use-Case 6	Die Benutzeroberfläche der Anwendung soll laut Lastenheft überall wo es sinnvoll ist auf sogenannte Auswahllisten anstelle von Texteingabefeldern zurückgreifen. Diese Listen sollen ebenfalls einfach editierbar sein, um die Auswahlmöglichkeiten schnell anpassen zu können.
Use-Case 7	Der geforderte Datenimport und -export wurde unter dem Use-Case 'Back-Up verwalten' zusammengefasst. Exportierte Daten sollen entweder direkt als Back-Up verwendet oder für Analyseanwendungen weiterverwendet werden. Der Datenimport bezieht sich hauptsächlich auf das Einspielen von Back-Up Datensätzen.

Tabelle 6.2: Use-Cases

Use-Case 8	Für die Verwaltung von Standorten sind die vier Grundverwaltungsaufgaben 'anlegen', 'bearbeiten', 'löschen' und 'einsehen' vorgesehen. Auch wenn die Standorte weniger dynamisch sind als andere Datensätze, so ist es doch möglich, dass das Unternehmen neue Fahrzeugstandorte aufbaut oder alte Standorte bei zu wenig Auslastung schließt.
Use-Case 9	Die zu verwaltende Ausrüstung ist fahrzeugbezogen. Da das Car-Sharing Unternehmen einen großen Wert auf Kundenzufriedenheit legt und dafür auch mehr anbieten möchte als andere Car-Sharing Unternehmen gibt es zusätzliche Ausrüstung (z.B. Fahrradträger, Dachbox oder Hundetransportbox) die bei verschiedensten Autos an- und abmontiert werden kann.
Use-Case 10	Eine Anforderung an die Desktopanwendung ist die Konfiguration der Benutzeroberfläche. Schriftgröße, Schriftart und Farbmodus sollen wählbar sein. Die Konfiguration findet anwendungsweit (also auf einzelnen Rechnern mit dieser Anwendung) statt.
Use-Case 11	Nach jeder abgeschlossenen Fahrt eines Kunden soll automatisch eine Rechnung erstellt werden. Da die Rechnung ein rechtsgültiges Dokument sein muss, ist die Bearbeitung des generierten Dokuments nicht möglich und auch das Löschen der Rechnung soll nicht möglich sein. Nach Abzahlung der offenen Rechnung soll nur die Archivierung erfolgen, welche für 10 Jahre aufgehoben werden muss.
Use-Case 12	Sollte die Zahlung einer ausstehenden Rechnung überfällig sein, soll eine Mahnung zur Zahlung erstellt werden, welche an den Kunden geschickt wird. Die exakten finanziellen Abläufe sind wegen des bereits bestehenden Finanzbuchhaltungssystems für diese Arbeit nicht relevant.
Use-Case 13	Voraussetzung für alle Verwaltungsaufgaben ist eine erfolgreiche Anmeldung am System. Je nach zugewiesener Rolle und der damit verbundenen Rechte dürfen verschiedene Datensätze verwaltet werden oder nicht.

Tabelle 6.3: Use-Cases

## 6.3 Use-Case Diagramm

Visual Paradigm Standard(Johannes Peters(DHBW Karlsruhe))

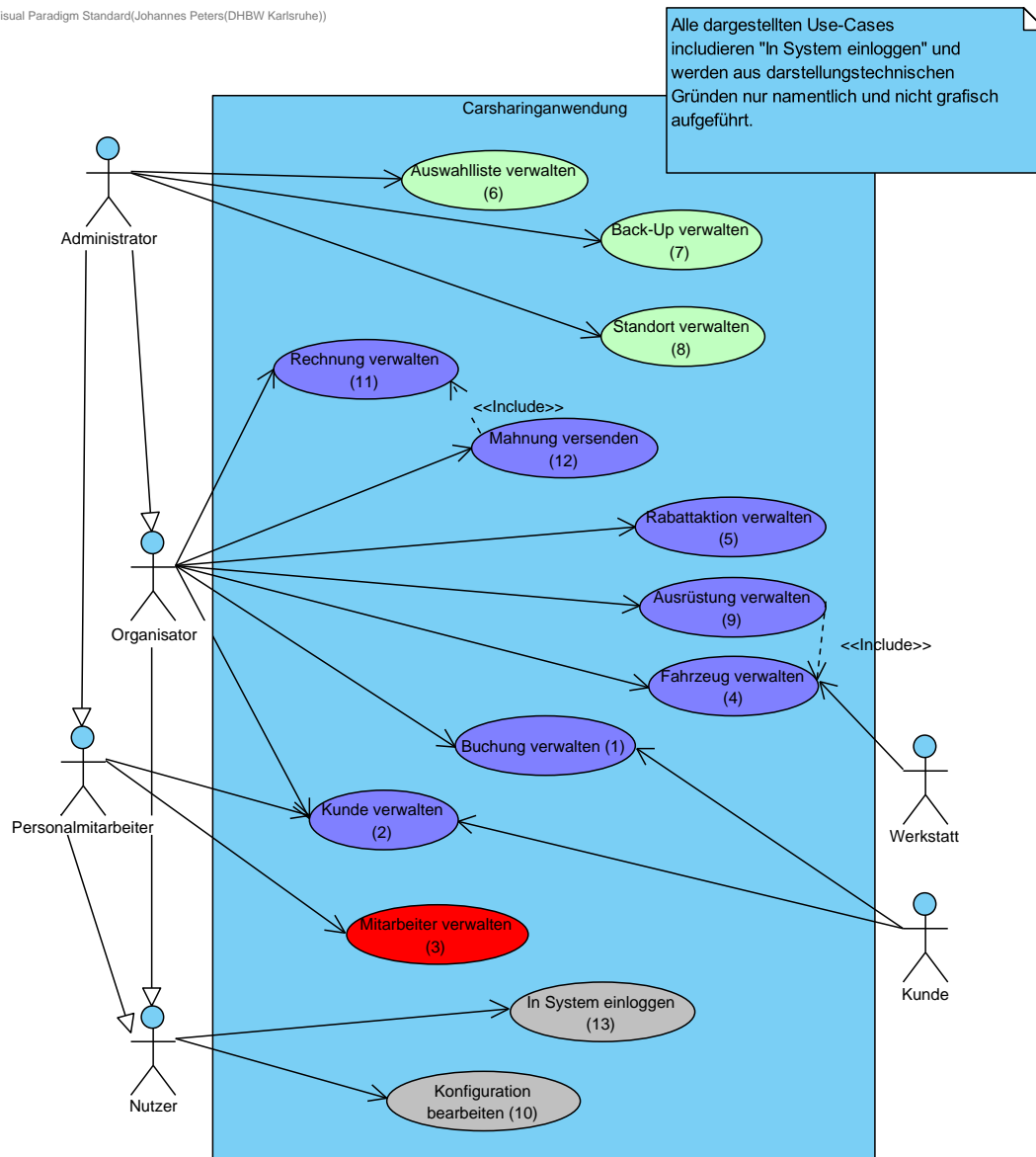


Abbildung 6.1: Use-Case Diagramm

Visual Paradigm Standard(Johannes Peters(DHBW Karlsruhe))

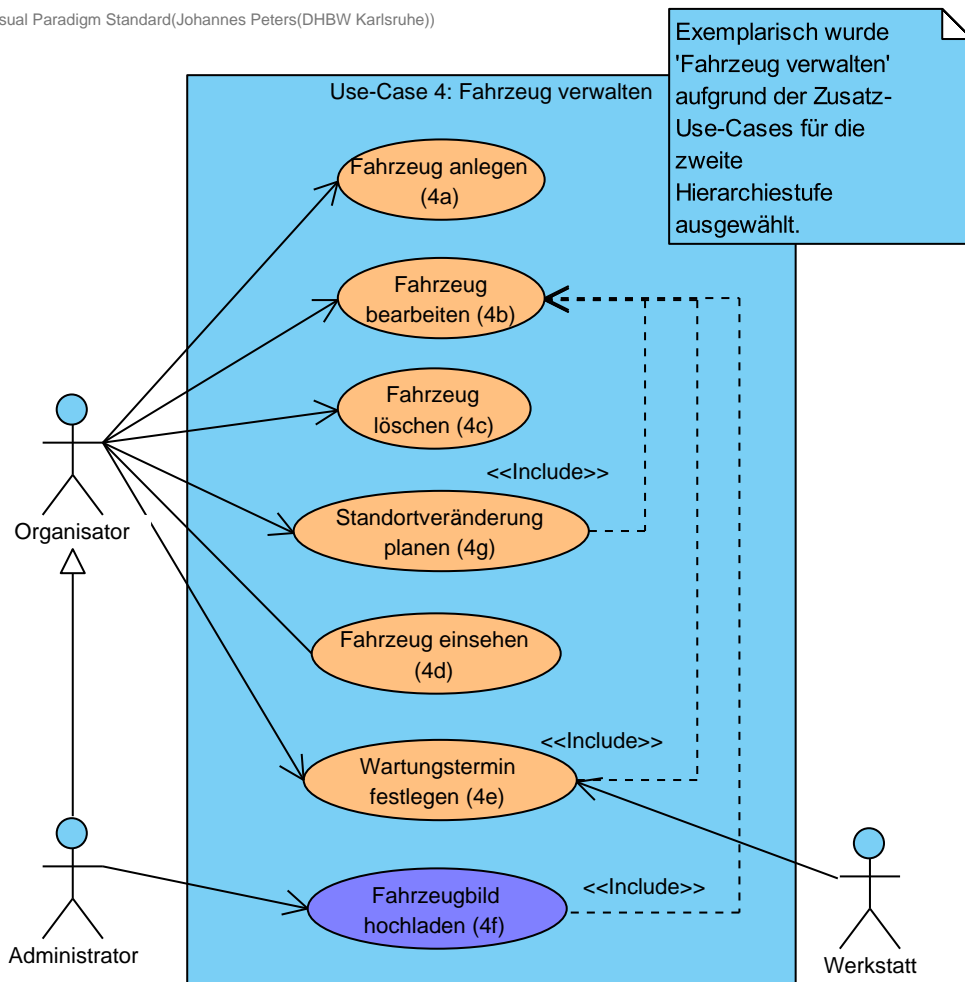


Abbildung 6.2: Use-Case 4: Fahrzeug verwalten

# 7 Analyse-Klassen-Diagramm

In diesem Kapitel wird zuerst auf die Klassen, welche für die Implementierung nötig sind, eingegangen bevor die Abhängigkeiten zwischen diesen Klassen nochmal übersichtlich mittel eines Analyse-Klassen-Diagramms dargestellt werden.

## 7.1 Klassen

- Kunde
- Adresse
- Vertrag
- Mitarbeiter
- Rolle
- Buchung
- Fahrzeug
- Fahrzeugklasse
- Bild
- Reifensatz
- Reifen
- Kennzeichen
- Ausrüstung
- Rechnung
- Mahnung
- Standort
- Filiale
- Rabattaktion
- Backup
- Hundetransportbox

**Kunde:** Die Kunden-Klasse stellt die Kunden dar und beinhaltet Attribute für alle Informationen, welche für die Identifikation und die Ausstellung von Rechnungen notwendig sind. Dazu besitzt die Klasse jeweils Assoziationen zu Adresse und Vertrag. Zusätzlich wird für die Identifikation eine ID gespeichert. Damit die Mitarbeiter mit dem Kunden einfacher in Kontakt treten können, wird ebenfalls eine Mail-Adresse und eine Telefonnummer gespeichert.

**Adresse:** Die Adresse wird in Straße, Hausnummer, Ort und Postleitzahl aufgeteilt. Mit der Adress-Klasse werden die Adressen der Kunden und Filialen dargestellt. Die Klasse 'Adresse' referenziert auf keine Klasse, sondern wird nur von anderen Klassen referenziert.

**Vertrag:** Die Vertrags-Klasse bildet die Verträge ab, welche die Kunden bindet. Dazu wird neben dem Vor- und Nachnamen der Pfad gespeichert, unter welchem die Vertrags-Datei (PDF) zu finden ist. Für die Zuordnung im System zwischen Vertrag und Kunde referenziert der Vertrag die Klasse Kunde.

**Mitarbeiter:** Die Klasse 'Mitarbeiter' stellt die Mitarbeiter des Unternehmens dar. Da die Mitarbeiter unterschiedliche Aufgaben und dementsprechend unterschiedliche Rollen in der Anwendung haben, steht die Mitarbeiter-Klasse in Assoziation zur Klasse 'Rolle'. Um einen Überblick über Anzahl der Mitarbeiter der Verschiedenen Standorte und Filialen zu erhalten gibt es ebenfalls eine Assoziation zur Klasse 'Filiale'. Für die Sicherung dieser und allen anderen Daten des Systems sollen Backups erstellt werden können, welche auf einen Mitarbeiter zurückführbar sein sollen. Daher muss die Mitarbeiter-Klasse auch in Assoziation zur Klasse 'Backup' stehen.

**Rolle:** Mit den Rollen sollen die Rechte innerhalb der Anwendung verteilt werden können. Dazu stellt die Klasse 'Rolle' neben dem Rollennamen für alle Funktionen der Anwendung jeweils ein Attribut bereit, welches vom Datentyp Boolean ist und angibt, ob ein Nutzer mit dieser Rolle Zugriff auf die entsprechende Funktion haben soll. Die Klasse 'Rolle' referenziert keine anderen Klassen, sondern wird nur von der Klasse 'Mitarbeiter' referenziert.

**Buchung:** Die Buchung wird als Koordinator-Klasse zwischen dem Kunden und dem Mitarbeiter dargestellt. Diese Klasse stellt einen vereinbarten Termin für die Mietung eines Autos dar. Dazu wird der Mitarbeiter, welcher die Buchung angelegt hat zusammen mit dem Kunden und dem Datum sowie der Uhrzeit des Termins gespeichert. Außerdem steht die Klasse 'Buchung' in Assoziation mit der Fahrzeug-Klasse, um festlegen zu können,

welches Auto gemietet werden soll. Da bei einer Buchung eines Fahrzeugs der Preis durch Rabattaktionen beeinflusst werden kann, steht die Buchungs-Klasse in Assoziation mit der Klasse 'Rabattaktion'. Nach dem Termin soll dem Kunden eine Rechnung mit den Buchungsdaten und anschließend eine Mahnung, falls der Kunde nicht rechtzeitig überweist, geschickt werden, weshalb es zusätzlich jeweils noch eine Assoziation zu 'Buchung' und 'Mahnung' geben muss. Eine Buchung referenziert in dieser Hinsicht immer nur eine Rechnung und maximal zwei Mahnungen, da nur ein mal eine Rechnung gesendet wird und falls der Kunde nach der zweiten Mahnung nicht zahlt keine weiteren Mahnungen mehr gesendet werden. Anstelle einer weiteren Mahnung lassen wir das schweizer Bankkonto des Kunden einfrieren.

**Fahrzeug:** Die Klasse Fahrzeuge ist für das Abbilden der Fahrzeuge zuständig. Jedes Fahrzeug ist dabei einer Fahrzeugklasse zugeordnet. Für die Darstellung eines Fahrzeugs referenziert die Klasse 'Fahrzeug' zudem die Klassen 'Kennzeichen', 'Reifensatz', 'Bild' und 'Ausrüstung'. Dabei kann jeweils nur ein Kennzeichen und Reifensatz referenziert werden. Für Bilder und Ausrüstung gibt es jedoch keine Obergrenze.

**Fahrzeugklasse:** Die Fahrzeugklassen sind für das Unterteilen der Fahrzeuge in verschiedene Kategorien zuständig. Diese Kategorien werden mit der Klasse 'Fahrzeugklasse' dargestellt. Eine Kategorie zeichnet sich dabei durch einen Namen, einen Preis und den benötigten Führerschein aus. Solch einer Kategorie sind beliebig viele Fahrzeuge zugeordnet.

**Bild:** Die Klasse Bild stellt die Bilder der Fahrzeuge dar. Jedes Bild ist einem Fahrzeug zugeordnet, weshalb die Klasse 'Bild' in Assoziation mit der Klasse 'Fahrzeug' steht. Als weitere Attribute besitzt diese Klasse den Titel des Bildes sowie den Pfad, an welchem das Bild liegt.

**Reifensatz:** Ein Reifensatz ist einem Fahrzeug zugeordnet, weshalb zwischen diesen beiden Klassen eine Assoziation besteht. Jeder Reifensatz besteht zudem aus vier Reifen und potenziell einem weiteren Reifen (Ersatzreifen).

**Reifen:** Die Klasse Reifen stellt die Reifen der Fahrzeuge dar. Ein Reifen wird hierbei durch den Hersteller, das Modell, den Zeitpunkt der Herstellung, die Profiltiefe und die Fahrtrichtung beschrieben.



**Kennzeichen:** Das Kennzeichen dient der Identifikation eines Fahrzeugs. Um das Kennzeichen eines Fahrzeugs in der Anwendung darzustellen wird die Klasse 'Kennzeichen' implementiert, welche die Zulassungsstelle, die Art des Kennzeichens und eine Zeichenkette für das eigentliche Kennzeichen als Attribute enthält.

**Ausrüstung:** Die Klasse 'Ausrüstung' stellt die Ausrüstung eines Fahrzeugs dar. Dabei handelt es sich nicht um die Ausstattung des Fahrzeugs (Klimaanlage, Panoramaschiebedach, ...), sondern um Fahrradträger, Dachbox und Hundetransportbox. Die Klasse 'Ausrüstung' fungiert dabei als Oberklasse der Ausrüstungsgegenstände und besitzt als Attribute die Kompatibilität mit den verschiedenen Fahrzeugen und das Fahrzeug, welches den Gegenstand momentan ausgerüstet hat. Da die Ausrüstung immer einem Fahrzeug zugeordnet ist, wird stets ein Fahrzeug referenziert.

**Rechnung:** Eine Rechnung fällt immer nach dem Abschluss einer Fahrt an. Die Klasse 'Rechnung' bildet solch eine Rechnung ab und beinhaltet Methoden um basierend auf der Buchung eine Rechnung zu erstellen und an den Kunden zu versenden. Solch eine Rechnung referenziert dabei immer auf die zugrundeliegende Buchung.

**Mahnung:** Wenn der Kunde die Rechnung nicht im vorgegebenen Zeitraum begleicht, wird eine Mahnung versendet. Die Klasse 'Mahnung' bildet solch eine Mahnung ab und beinhaltet Methoden, um basierend auf der Buchung eine Mahnung zu erstellen und um diese anschließend an den Kunden zu versenden. Da eine Mahnung stets auf einer Buchung basiert, wird die zugrundeliegende Buchung referenziert.

**Standort:** Das Unternehmen ist an verschiedensten Standorten tätig. Um diese Standorte abzubilden wird eine Klasse 'Standort' implementiert, welche in Assoziation mit den Klassen 'Fahrzeug' und 'Filiale' steht. Dadurch können dem Standort alle Mitarbeiter und Fahrzeuge zugeordnet werden. Des Weiteren wird der Standort durch eine Beschreibung und einen Namen beschrieben. Die Anzahl der Fahrzeuge und Filialen ist hierbei nicht beschränkt.

**Filiale:** Die Klasse 'Filiale' stellt eine Filiale des Unternehmens dar. Dafür wird die Filiale sowohl mit dem Standort als auch mit den Mitarbeitern sowie einer Adresse in Assoziation gesetzt. Außerdem besitzt diese Klasse die Anzahl der Mitarbeiter als Attribut. Eine Filiale ist stets immer einem Standort zugeordnet und hat genau eine Adresse sowie mindestens einen Mitarbeiter.

**Rabattaktion:** Mit der Klasse 'Rabattaktion' werden die Rabattaktionen dargestellt. Dazu besitzt diese Klasse den prozentualen Preisnachlass und den Namen der Rabattaktion als Attribute.

**Backup:** Die Klasse Backup ist für die Darstellung der Backups zuständig. Dafür wird zum einen das Datum und die Uhrzeit der Erstellung des Backups und zum anderen der Pfad, unter welchem das Backup abgelegt wurde, als Attribut bereitgestellt.

**Hundetransportbox:** Die Hundetransportbox ist eine der Unterklassen der Ausrüstung, welche die Klasse 'Ausrüstung' um folgende Attribute erweitert: Das Maximalgewicht des Hundes, Höhe, Breite und Länge sowie das Volumen.

**Fahrradträger:** Ein Fahrradträger ist ebenfalls ein Teil der Ausrüstung und somit eine Unterklasse der Klasse 'Ausrüstung'. Die Klasse Fahrradträger hat zusätzlich Attribute für die Anzahl der Fahrräder, welche mit dem Fahrradträger Transportiert werden können, und das Maximale Gewicht, mit welchem der Fahrradträger beladen werden darf.

**Dachbox:** Die letzte Unterklasse der Ausrüstung ist die Dachbox, welche zusätzlich Attribute für das Volumen und die Abschließbarkeit der Dachbox besitzt.

## 7.2 Diagramm

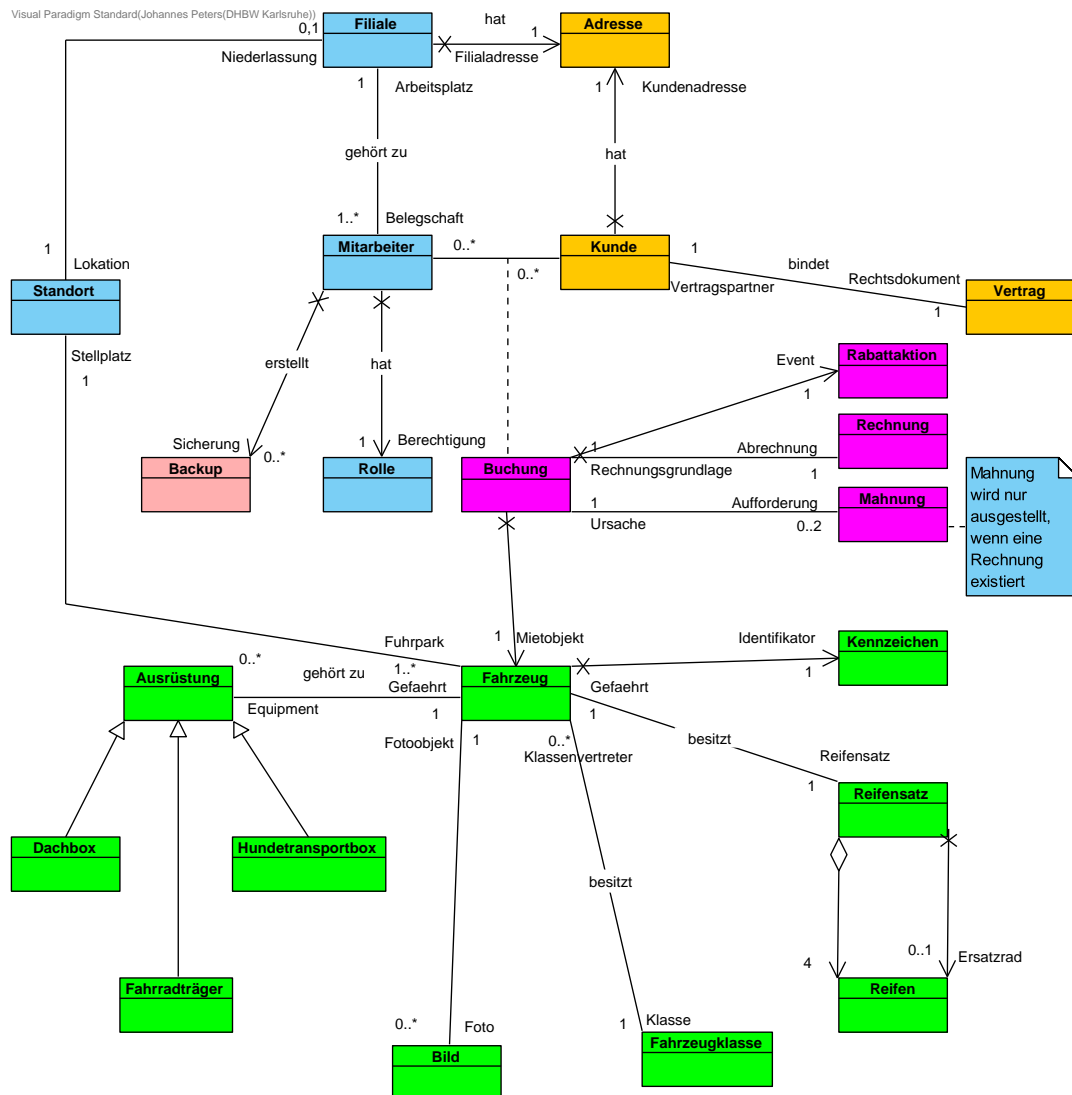


Abbildung 7.1: Analyseklassendiagramm

## 8 Sequenzdiagramm

Im Folgenden wird der Vorgang eines vollständigen Buchungsdurchgangs betrachtet, um final mittels Sequenzdiagrammen das Vorgehen visuell darzustellen.

Bei der Erarbeitung wird aufgrund des Umfangs ein Teil der Detailtiefe entfernt. Die exakte Kommunikation mit der Datenbasis (in Realität mit einer Datenbank und im Projekt mit den CSV-Dateien) wird nicht betrachtet. Bei der Interaktion mit der Anwendung wird die Unterteilung in verschiedene GUIs unter dem Begriff 'Benutzeroberfläche' zusammengefasst. Es wird ebenfalls davon ausgegangen, dass für den Sequenzablauf benötigte Datensätze wie Fahrzeuge, Standorte, Kunden und Mitarbeiter bereits vorhanden sind und nicht erst angelegt werden müssen. Davon auszugehen, dass mit einer vollständig leeren Datenbasis begonnen wird ist für die Abbildung des Buchungsablaufs nicht zielführend. Aus diesem Grund wird angenommen, dass ausschließlich die Datenbasis der Buchungen, Rechnungen und Mahnungen leer ist. Weiterhin wird in den Sequenzdiagrammen auf die Verwendung von Funktionen verzichtet und Vorgänge werden umschrieben oder umgangssprachlich aufgeführt.

## 8.1 Aktionsbetrachtung: Buchung eines Fahrzeugs

Eine Buchung soll angelegt werden. Abgesehen vom eigentlichen Buchungsdurchgangs soll auch die Abrechnung des Buchungstermins einbezogen werden. Ausgehend davon lassen sich mehrere Teil-Abläufe identifizieren: Buchung eines Fahrzeugs, Stornierung einer Buchung, Antreten des gebuchten Termins, Beendigung der Fahrt mit Erstellung einer Rechnung und notfalls Mahnungen. Die Buchung und die Stornierung lassen sich zusammenfassen, gleiches gilt für die Beendigung und die Rechnungsausstellung. Da die Buchung über die Desktopanwendung ausschließlich in der Filiale stattfinden kann, ist ein Akteur der Organisator, der zweite Aktuer ist der Kunde.

Für die Buchung und Stornierung interagieren die Akteure mit der Benutzeroberfläche und der Datenbasis. Beim Fahrtantritt findet die Interaktion mit dem Kartenlesegerät und der Datenbasis ab. Das Ende der Fahrt inkludiert neben dem Lesegerät und der Datenbasis einen Server, die Rechnung und Mahnung, sowie den Kunden als Datensatz im System und nicht nur als Akteur.

## 8.2 Pseudo-Code

```
1  "Buchung anlegen"
2
3  BEGINN BUCHUNG-ANLEGEN
4      Organisator klickt "Buchung anlegen"
5      Buchungsformular öffnet sich
6
7      SOLANGE nicht alle Felder ausgefüllt sind
8          Zeige Fehlermeldung "Bitte alle Felder ausfüllen"
9
10         WENN Kunde bereits im System vorhanden
11             Organisator wählt Kunde aus
12         SONST
13             FUEHRE AUS KUNDE-ANLEGEN
14             Organisator wählt Kunde aus
15         ENDE WENN
16
17         Organisator wählt Start- und Endetermin der Buchung
18         SOLANGE gewählter Endetermin vor dem Starttermin liegt
19             Zeige Fehlermeldung "Starttermin muss vor ↯
20                 ↰ Endetermin liegen"
21             Organisator wählt Start- und Endetermin der Buchung
22         ENDE SOLANGE
23
24         Organisator wählt Standort aus
25         Fahrzeuge des gewählten Standortes laden
26         Fahrzeuge des gewählten Standortes werden angezeigt
27         Organisator wählt Fahrzeug aus
28
29         WENN Kunde verfügt über Rabattcode
30             Organisator wählt Rabattaktion aus
31         SONST
32             Organisator wählt "-" bei Rabattaktion aus.
33         ENDE WENN
34     ENDE SOLANGE
```

```
34
35     Organisator klickt auf "Buchung überprüfen"
36     WENN Fahrzeug nicht verfügbar
37         SOLANGE Fahrzeug nicht verfügbar
38             Zeige Fehlermeldung "Das gewählte Fahrzeug ist in ↗
                 ↳ diesem Zeitraum nicht verfügbar"
39             Organisator wählt Fahrzeug aus
40         ENDE SOLANGE
41     ENDE WENN
42     Buchung speichern
43
44     WENN Kunde möchte Buchung stornieren
45         FUEHRE AUS BUCHUNG-STORNIEREN
46     ENDE WENN
47 ENDE BUCHUNG-ANLEGEN
48
49
50 BEGINN KUNDE-ANLEGEN
51     Formular für das Anlegen eines Kunden zeigen
52     Solange nicht alle Felder ausgefüllt sind
53         Zeige Fehlermeldung "Bitte alle Felder ausfüllen"
54         Organisator trägt Vorname des Kunden ein
55         Organisator trägt Nachname des Kunden ein
56         Organisator trägt Adresse des Kunden ein
57         Organisator trägt Mail-Adresse des Kunden ein
58         Organisator trägt Telefonnummer des Kunden ein
59         Organisator trägt Nummer des Schweizer Bankkontos des ↗
                 ↳ Kunden ein
60         Organisator klickt auf "Kunde" anlegen
61         WENN Schweizer Bankkonto valide
62             Sende Verifizierungs-Mail an die angegebene ↗
                 ↳ Mail-Adresse
63             Kunde klickt auf "Verifizieren"
64         SONST
65             Sende Mail mit Anzeige wegen Betrug an Polizei
66             Abbruch des Anlegens
```

```
67          ENDE WENN
68 ENDE KUNDE-ANLEGEN
69
70
71 BEGINN BUCHUNG-STORNIEREN
72     Organisator klickt auf "Kunden"
73     Zeige Übersicht aller Kunden an
74     Organisator wählt Kunde aus
75     Organisator wählt Buchung aus
76     Organisator klickt "Stornieren"
77     Organisator klickt erneut "Stornieren"
78     Buchung wird aus Datenbasis gelöscht
79 ENDE BUCHUNG-STORNIEREN
```

Listing 8.1: Szenario Event anlegen



## 8.3 Diagramme

### 8.3.1 Buchung anlegen

Das Anlegen einer Buchung wird in Abbildung 8.1 auf Seite 60 dargestellt. Die Sequenz beginnt damit, dass der Kunde einen Terminwunsch gegenüber einem Mitarbeiter äußert. Der Organisator öffnet daraufhin das Formular für das Anlegen einer Buchung und teilt dem Kunden mit, dass die Buchung nun entgegengenommen werden kann.

Für die eigentliche Buchung nennt der Kunde dem Organisator zuerst den Zeitraum, den Standort und das Fahrzeug. Der Organisator trägt diese Daten in das Formular ein und klickt anschließend auf 'Buchung überprüfen'. Daraufhin lädt die Anwendung die Fahrzeug-, Standort- sowie Buchungsdaten und gleicht diese mit der Eingabe des Organisators ab. Falls es eine Kollision mit einer anderen Buchung gibt, wird die Buchung abgebrochen. In diesem Fall nennt der Kunde andere Buchungsdaten, bis es keine Kollision mehr gibt. Sollte es keine Kollision geben, wird die Buchung in der Datenbasis hinterlegt und das System bestätigt die Buchung. Daraufhin teilt der Organisator dem Kunden mit, dass die Buchung erfolgreich war.

Optional kann der Kunde die Buchung nun stornieren. Dazu muss der Kunde gegenüber dem Organisator den Wunsch äußern, die Buchung zu stornieren. Der Organisator erfragt daraufhin den Namen des Kunden und gibt diesen in der entsprechenden Suchleiste ein. Die Anwendung lädt daraufhin die Kundendaten aus der Datenbasis und zeigt sie dem Organisator an. Im Anschluss erfragt der Organisator, welche Buchung der Kunde gern stornieren möchte. Nachdem der Kunde die Buchung spezifiziert hat, wählt der Organisator die entsprechende Buchung aus. Daraufhin lädt die Anwendung die Buchungsdaten für die spezifizierte Buchung aus der Datenbasis und zeigt diese Daten an. Der Organisator klickt anschließend auf 'Buchung stornieren', wodurch die Buchung aus der Datenbasis gelöscht wird. Der Organisator erhält daraufhin eine Bestätigung für das Löschen der Buchung und teilt dem Kunden mit, dass die Buchung erfolgreich storniert wurde.

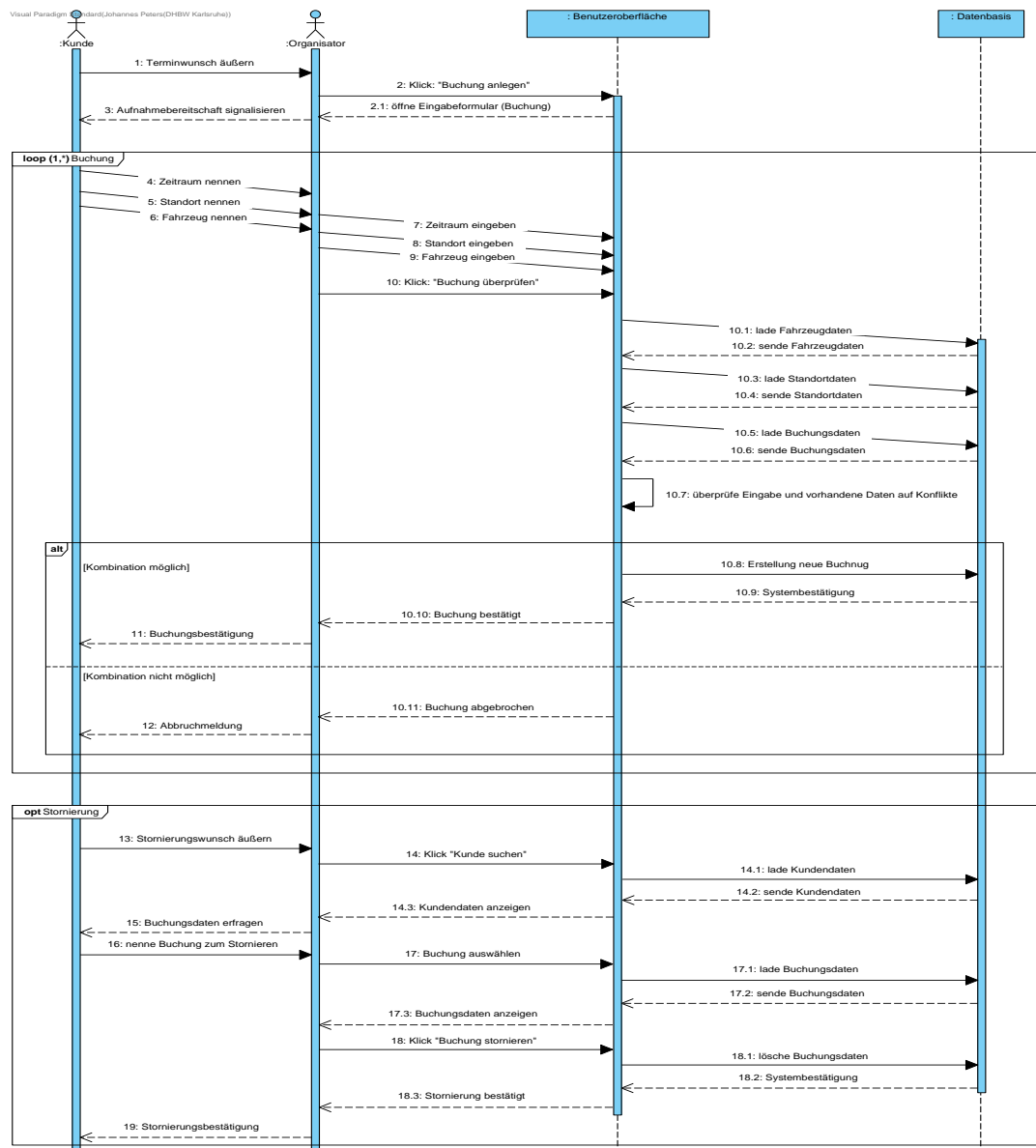


Abbildung 8.1: Buchung eines Termins

### 8.3.2 Fahrt antreten

Das Antreten einer Fahrt wird in Abbildung 8.2 auf Seite 61 dargestellt. Hierbei lässt der Kunde zuerst seine Kundenkarte vom Lesegerät einscannen, welches die Kundendaten ausliest und im Anschluss die Buchungsdaten aus der Datenbasis lädt. Nachdem die Buchungsdaten geladen wurden, wird zunächst der Standort überprüft, bevor die Kundendaten mit den Buchungsdaten abgeglichen werden. Bei einer übereinstimmung wird zuerst der Kilometerstand für die Abrechnung gespeichert und danach wird das Fahrzeug entriegelt. Nun kann der Kunde seine Fahrt beginnen.

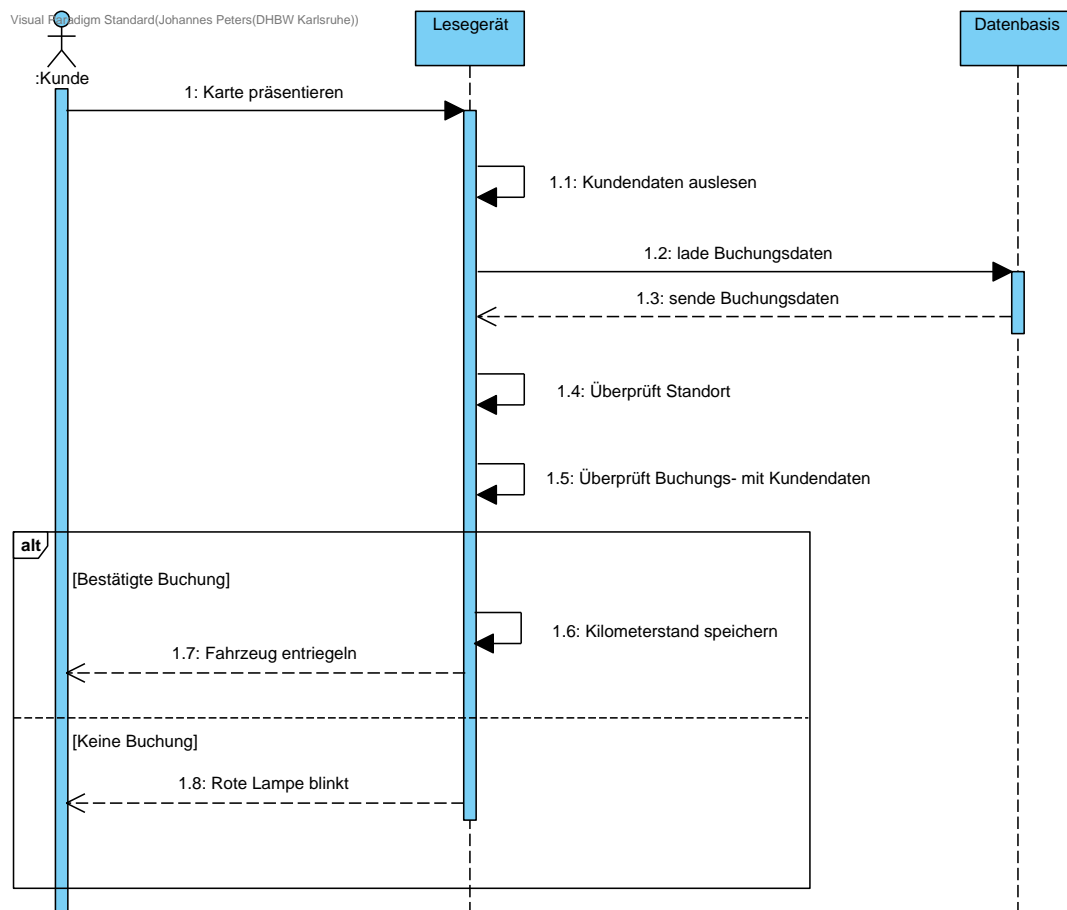


Abbildung 8.2: Antritt eines gebuchten Termins

### 8.3.3 Fahrt beenden

Das Beenden einer Fahrt wird in Abbildung 8.3 auf Seite 62 dargestellt. Sobald der Kunde das Fahrzeug abstellt wird der Kilometerstand vom Lesegerät abgefragt und das Fahrzeug wird verriegelt. Während der Kunde auf seine Rechnung wartet wird der neue Kilometerstand an den Server gesendet, welcher nach Abfrage des alten Kilometerstands und den Buchungsdaten die gefahrenen Kilometer berechnet. Basierend auf den Buchungsdaten und den gefahrenen Kilometern wird eine Rechnung erstellt, welche vom Server per Mail an den Kunden geschickt wird.

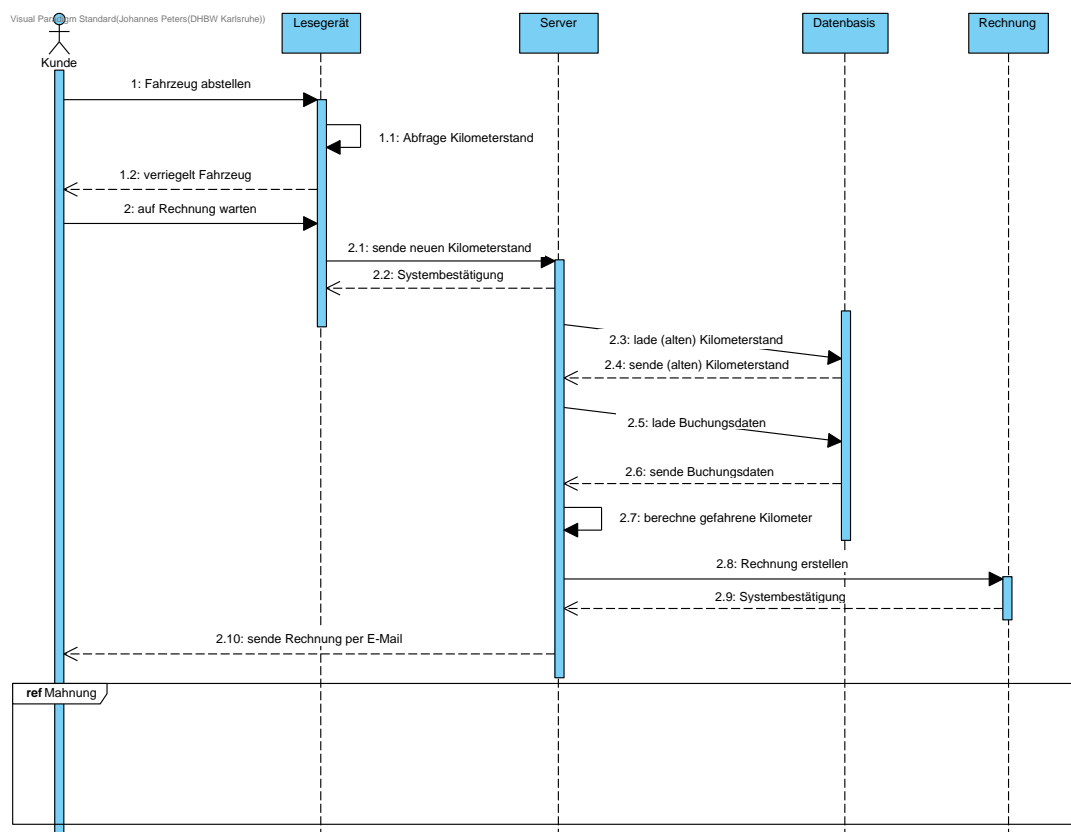


Abbildung 8.3: Abschluss eines gebuchten Termins

Falls die Rechnung vom Kunden ignoriert wird, erhält der Kunde nach 30 und nach 45 Tagen eine Mahnung. Dies wird in Abbildung 8.3 auf Seite 62 dargestellt. Um eine Mahnung zu versenden, lädt der Server zuerst die Rechnungsdaten der überfälligen Rechnung von der Datenbasis. Im Anschluss erstellt der Server basierend auf der Rechnung

eine Mahnung und sendet diese per Mail an den Kunden. Sollte der Kunde nach 60 Tagen nicht gezahlt haben, so wird er für vom Carsharing-Angebot ausgeschlossen. Dazu lädt der Server zuerst die Mahnungsdaten aus der Datenbasis, um den Status der Mahnung zu überprüfen. Wenn die Rechnung zu diesem Zeitpunkt nicht beglichen wurde, werden die Kundendaten aus der Datenbasis geladen und aktualisiert, sodass der Kunde für zukünftige Buchungen gesperrt ist. Sobald der Kunde blockiert ist, sendet der Server eine Mail an den Kunden mit der Information, dass der Kunde in Zukunft vom Carsharing-Angebot ausgeschlossen ist und sein Schweizer Bankkonto eingefroren wurde.

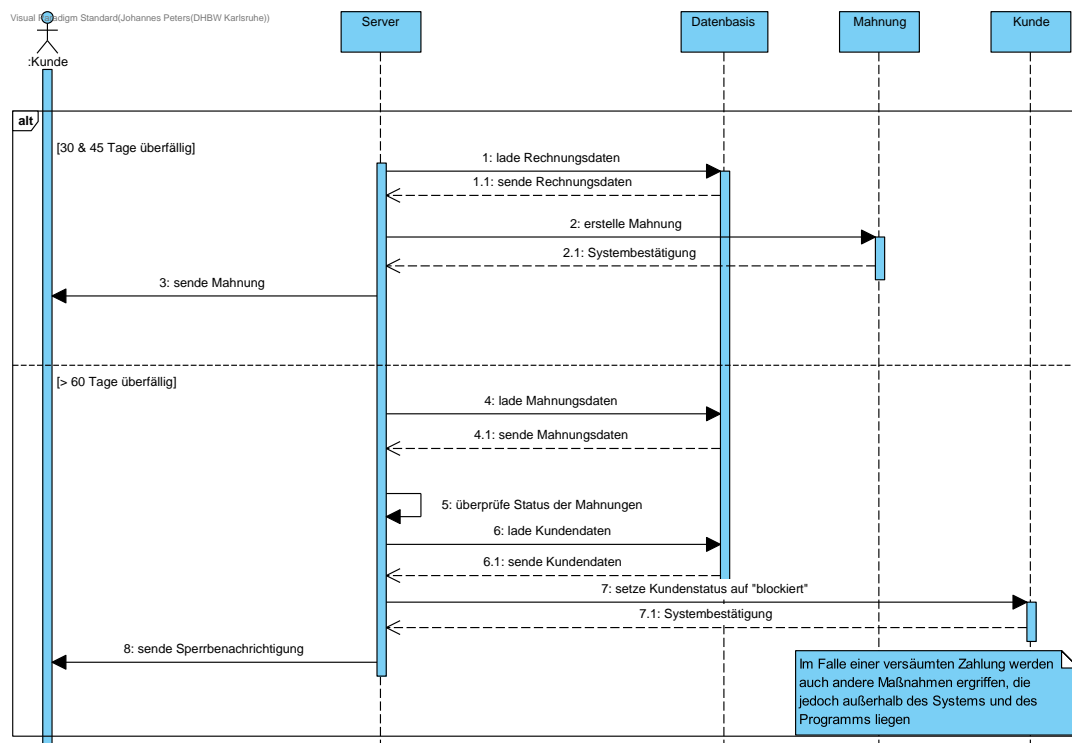


Abbildung 8.4: Erstellung der Mahnungen

## 9 Aktivitätsdiagramm

In diesem Kapitel erfolgt die Erarbeitung der Aktion 'Standort mit neuen Fahrzeugen anlegen' in Form von Aktivitätsdiagrammen. Da hierbei von einer vollständig leeren Datenbasis ausgegangen werden soll, muss jedoch die Aktion aufgegliedert werden. Die Unteraktionen 'Standort anlegen', 'Fahrzeug anlegen' und letztendlich 'Fahrzeug einem Standort zuordnen' müssen somit voneinander getrennt betrachtet werden.

### 9.1 Pseudo-Code

```
1  "Standort mit Fahrzeugen anlegen"
2
3  BEGINN STANDORT-MIT-FAHRZEUGEN-ANLEGEN
4      FUEHRE AUS STANDORT-ANLEGEN
5      FUEHRE AUS FAHRZEUG-ANLEGEN
6      FUEHRE AUS ZUORDNEN
7  ENDE STANDORT-MIT-FAHRZEUGEN-ANLEGEN
8
9
10 BEGINN STANDORT-ANLEGEN
11     Mitarbeiter gibt Nutzernamen und Passwort ein
12     Mitarbeiter klickt auf anmelden
13     WENN Nutzernamen nicht existiert ODER Passwort falsch
14         Zeige Fehlermeldung 'Fehlerhafte Anmeldung'
15     SONST
16         Zeige Startseite
17     ENDE WENN
18     Mitarbeiter klickt auf Standorte
19     Zeige Standort-GUI
20     Mitarbeiter klickt auf Button '+' oder auf Link 'neuen ↗
        ↳ Standort'
21     Zeige Formular für das Anlegen eines Standortes
```

```
22
23     SOLANGE nicht alle Felder ausgefüllt sind
24         Zeige Fehlermeldung 'Bitte füllen Sie alle Felder aus, ↵
           ↳ um einen neuen Standort anzulegen.'
25         Mitarbeiter gibt Name an
26         Mitarbeiter gibt Beschreibung an
27         Mitarbeiter gibt Straße an
28         Mitarbeiter gibt Hausnummer an
29         Mitarbeiter gibt Postleitzahl an
30         Mitarbeiter gibt Stadt an
31         Mitarbeiter gibt maximale Fahrzeuganzahl an
32     ENDE SOLANGE
33
34     WENN Mitarbeiter hat Administrator-Rolle
35         WENN Mitarbeiter möchte Bilder hochladen
36             Mitarbeiter klickt auf 'Bild hochladen'
37             Mitarbeiter wählt Bild aus
38         ENDE WENN
39     ENDE WENN
40
41     Mitarbeiter klickt auf 'Standort speichern'
42
43     WENN Standort bereits vorhanden
44         Zeige Fehlermeldung 'Standort bereits vorhanden'
45     SONST
46         Zeige Bestätigung für das Speichern des Datensatzes
47     ENDE WENN
48 ENDE STANDORT-ANLEGEN
49
50
51
52 BEGINN FAHRZEUG-ANLEGEN
53     WENN Mitarbeiter noch nicht angemeldet
54         Mitarbeiter gibt Nutzernamen und Passwort ein
55         Mitarbeiter klickt auf anmelden
56         WENN Nutzernamen nicht existiert ODER Passwort falsch
```

```
57         Zeige Fehlermeldung 'Fehlerhafte Anmeldung'
58     SONST
59         Zeige Startseite
60     ENDE WENN
61 ENDE WENN
62 Mitarbeiterklickt auf 'Fahrzeuge'
63 Zeige Fahrzeug-GUI
64 Mitarbeiter klickt auf Button '+' oder auf Link 'neues ↗
    ↘ Fahrzeug'
65 Zeige Formular für das Anlegen eines Standortes
66
67 SOLANGE nicht alle Felder ausgefüllt sind
68     Zeige Fehlermeldung 'Bitte füllen Sie alle Felder aus, ↗
        ↘ um einen neuen Standort anzulegen.'
69 Mitarbeiter gibt Kennzeichen an
70 Mitarbeiter gibt Farbe an
71 Mitarbeiter gibt Modell an
72 Mitarbeiter gibt Beschreibung an
73 Mitarbeiter wählt Marke aus
74 Mitarbeiter gibt Baujahr an
75 Mitarbeiter wählt Fahrzeugklasse aus
76 Mitarbeiter wählt Reifensatz aus
77 ENDE SOLANGE
78
79 SOLANGE Fahrzeug verfügt über nicht zugewiesene Ausrüstung
80     Mitarbeiter wählt Ausrüstung aus
81 ENDE SOLANGE
82
83 WENN Mitarbeiter hat Administrator-Rolle
84     WENN Mitarbeiter möchte Bilder hochladen
85         Mitarbeiter klickt auf 'Bild hochladen'
86         Mitarbeiter wählt Bild aus
87     ENDE WENN
88 ENDE WENN
89
90 Mitarbeiter klickt auf 'Fahrzeug speichern'
```



```
91
92     WENN Fahrzeug bereits vorhanden
93         Zeige Fehlermeldung 'Fahrzeug bereits vorhanden'
94     SONST
95         Zeige Bestätigung für das Speichern des Datensatzes
96     ENDE WENN
97 ENDE FAHRZEUG-ANLEGEN
98
99
100
101 BEGINN ZUORDNEN
102     WENN Mitarbeiter noch nicht angemeldet
103         Mitarbeiter gibt Nutzernamen und Passwort ein
104         Mitarbeiter klickt auf anmelden
105         WENN Nutzernamen nicht existiert ODER Passwort falsch
106             Zeige Fehlermeldung 'Fehlerhafte Anmeldung'
107         SONST
108             Zeige Startseite
109         ENDE WENN
110     ENDE WENN
111     Mitarbeiter klickt auf 'Fahrzeuge'
112     Zeige Fahrzeug-GUI
113     Mitarbeiter wählt Fahrzeug aus
114     Mitarbeiter klickt auf 'Bearbeiten'
115     Mitarbeiter wählt Standort aus
116     Mitarbeiter klickt auf 'Speichern'
117     Zeige Bestätigung für das Speichern des Datensatzes
118 ENDE ZUORDNEN
```

Listing 9.1: Szenario Standort mit Fahrzeugen anlegen

## 9.2 Diagramme

### 9.2.1 Standort anlegen

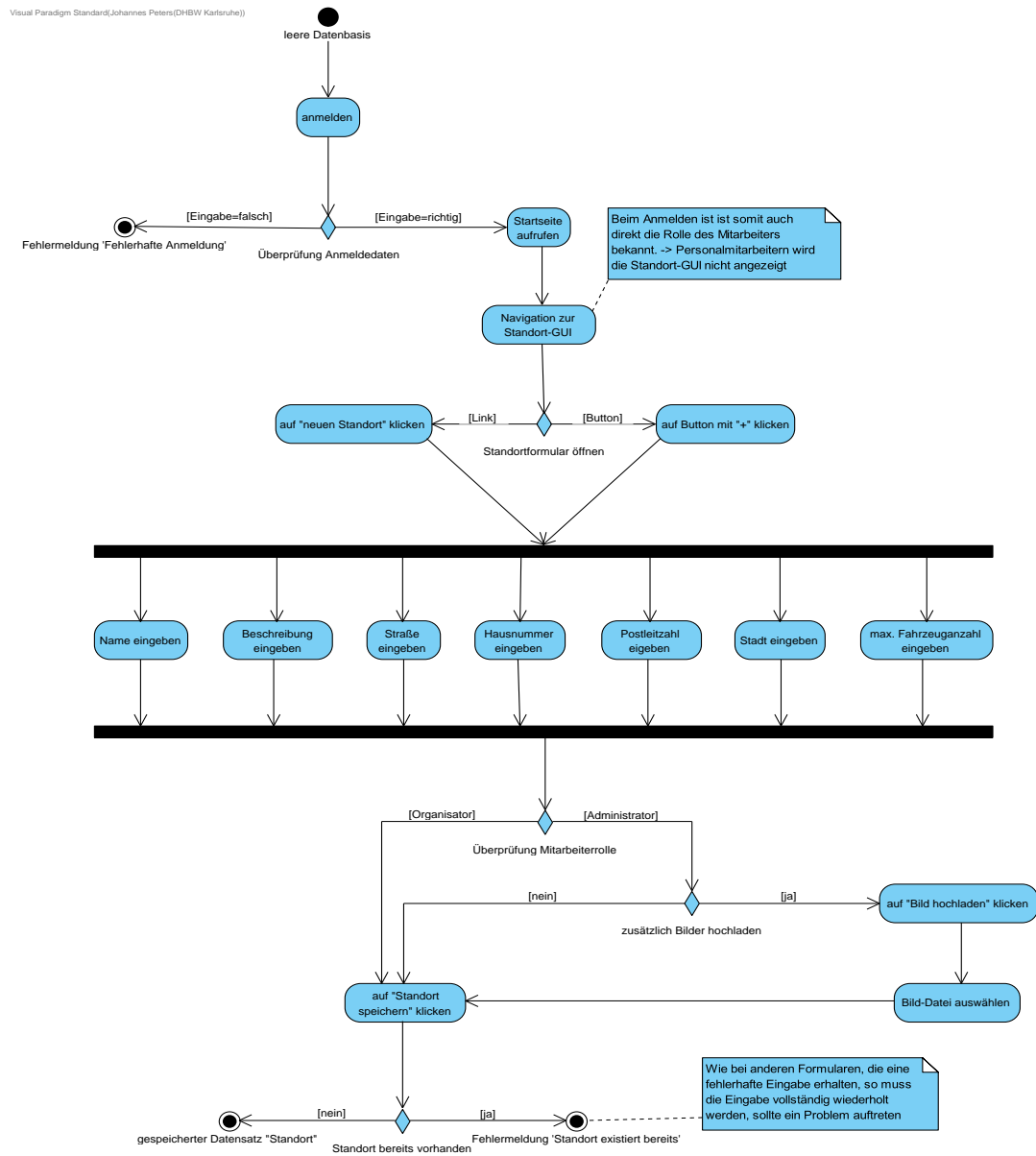


Abbildung 9.1: Aktivitätsdiagramm: Standort anlegen

Wie bei jeder anderen auf dem System ausgeführten Aktion muss auch beim Anlegen eines neuen Standorts zuerst eine Anmeldung vollzogen werden. Sobald ein Mitarbeiter im System angemeldet ist stehen mittels seines Benutzerprofils auch die Rechte des Nutzers griffbereit. Personalmitarbeiter soll es nicht möglich sein, Standorte anzulegen oder andere Aktionen mit solchen Datensätzen auszuführen, sodass diesen Mitarbeitern eine andere Benutzeroberfläche gezeigt wird wie den Organisatoren und Administratoren.

Nachdem zu der entsprechenden GUI navigiert wurde und der Mitarbeiter einen neuen Standort anlegen möchte, gibt es zwei Möglichkeiten zum entsprechenden Eingabeformular zu gelangen. Um für die Mitarbeiter, deren IT-Kenntnisse teils mangelhaft sind, mehrere Vorschläge darzustellen gibt es einmal den offensichtlichen Text-Button mit 'neuen Standort anlegen', für die etwas bewanderten Mitarbeiter, zusätzlich einen '+'-Button. Sobald sich das Eingabeformular geöffnet hat, müssen die Felder ausgefüllt werden.

In welcher Reihenfolge dies ausgeführt wird, ist irrelevant, weshalb im Diagramm diese Aktionen auf parallel dargestellt werden. Sobald alle Texteingaben abgeschlossen wurden, besteht noch die Option Bilder vom Standort hochzuladen, sozusagen ein Titelbild für den Detaileintrag des Standorts. Das Hochladen von Bildern dürfen nur die Administratoren ausführen, sodass zuerst eine Rollenabfrage stattfindet. Sollte ein Administrator diesen Standort anlegen, so wird ihm die Möglichkeit angezeigt und sollte er Bilder hinzufügen möchten, kann er über einen 'Standard-File-Choose-Dialog' Bilder von seinem Computer auswählen und in das Programm laden. Sofern dies erledigt wurde, oder auch nicht, ist es möglich den neuen Standorteintrag zu speichern.

Bevor der Standort gespeichert wird, erfolgt die Überprüfung, ob dieser Standort bereits vorhanden ist. Sollte das Ergebnis negativ sein, wird der Standort gespeichert. Ist jedoch in der Planung oder Organisation ein Fehler aufgetreten und der Standort existiert bereits, erscheint eine Fehlermeldung und die Aktion wird abgebrochen. An sich hätte man im Diagramm anstelle des Endzustandes 'Fehlermeldung' auch eine Schleife einbauen können, die solange läuft wie die Eingabe inkorrekt ist oder bis die Aktion händisch abgebrochen wird. In diesem Fall wird jedoch davon ausgegangen, dass allein auf Basis der Funktionsart eines Eingabeformulars die Eingabefelder geleert werden und die Eingabe wiederholt werden muss. Außer der Abfrage, ob der Standort bereits existiert, findet keine andere logische Überprüfung statt. Sofern sich die Adresse nicht doppelt wird der Datensatz gespeichert. Sollte die Fehlermeldung erscheinen, ist es auch sehr unwahrscheinlich, dass der Mitarbeiter ein zweites Mal die Erstellung des Standortes versucht.

## 9.2.2 Fahrzeug anlegen

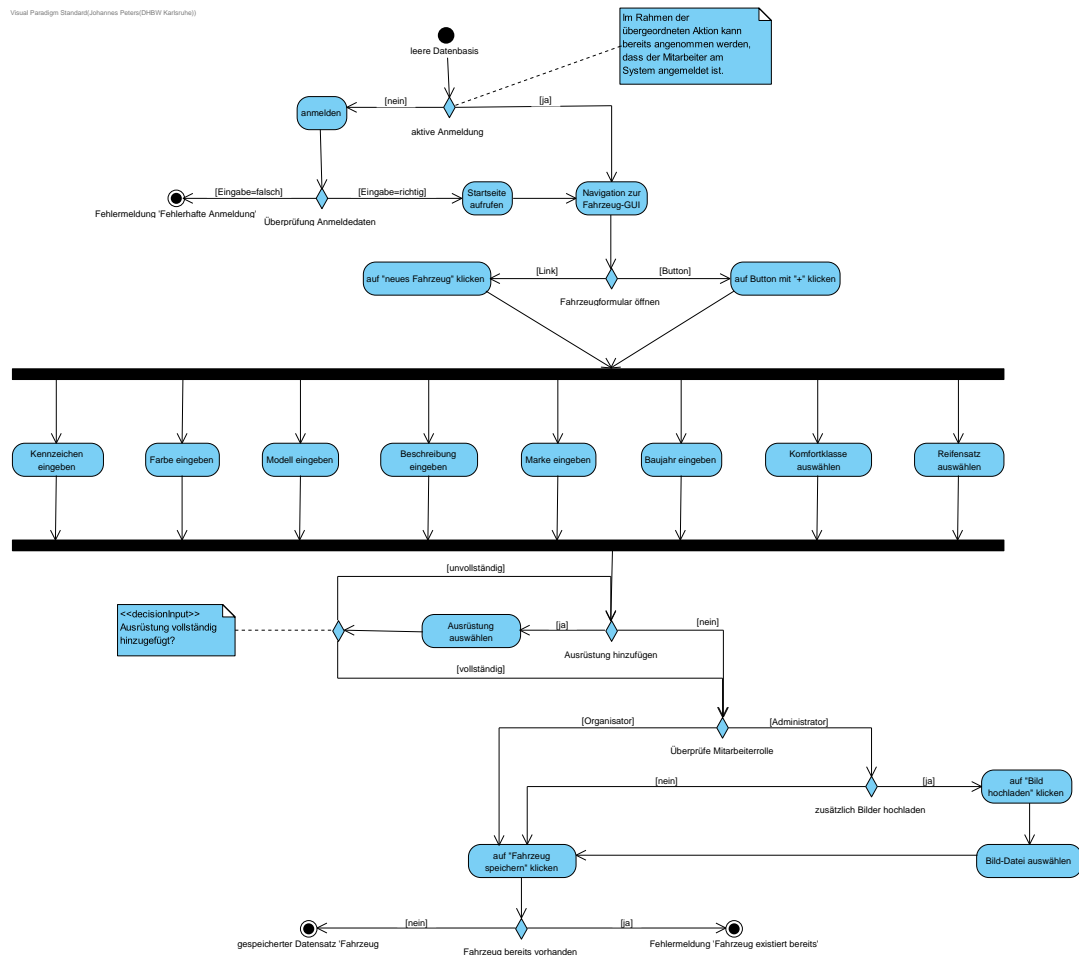


Abbildung 9.2: Aktivitätsdiagramm: Fahrzeug anlegen

Die eigentliche Aktion, die mit den Aktivitätsdiagrammen analysiert werden soll, ist das Anlegen eines Standortes mit Fahrzeugen. Nur zum besseren Verständnis wurde diese Aktion aufgeteilt, d. h., dass der gesamte Vorgang eigentlich flüssig hintereinander abläuft und der Mitarbeiter, der sich bereits für das Anlegen eines Standortes angemeldet hat, immer noch im System eingeloggt befindet. Theoretisch könnte eine Unterbrechung möglich sein und wurde daher exemplarisch in dieses Diagramm eingebaut.

Viele Bestandteile des Diagramms sind dem Standort-Diagramm sehr ähnlich, da das Erstellen und Eingeben von Datensätzen in allen Anwendungsfällen ähnlich ist.

Sollte der Mitarbeiter noch im System tätig sein, so wird er sich bereits irgendwo auf den Benutzeroberflächen befinden und muss sich im Verwaltungsbereich der Anwendung zur Fahrzeug-GUI navigieren. Sollte der Mitarbeiter, aus welchen Gründen auch immer, nicht mehr im System angemeldet sein, so muss dieser sich zuerst wieder anmelden.

Die Formularbearbeitung ist identisch zum Aufbau bei den Standorten. Da es auch hier irrelevant ist in welcher Reihenfolge die Daten eingegeben werden, wird Parallelisierung verwendet, um dies darzustellen. Die meisten Eingabeaktionen enthalten "eingegeben", doch gibt es einige Aktionen, die "auswählen" lauten und somit eine Auswahlmöglichkeit in Form von Drop-Down-Listen implizieren.

Einem Fahrzeug kann Ausrüstung hinzugefügt werden, was durch eine Decision-Node dargestellt wird. Da die Ausrüstung von Fahrzeugen aus Objekten wie Fahrradträger, Dachbox oder Hundetransportskäfig besteht, ist es möglich, dass ein Fahrzeug mehrere Ausrüstungsgegenstände hat. Um eine variable Anzahl an Ausrüstungsgegenständen einem Fahrzeug zuweisen zu können gibt es mittels weiterer Decision-Nodes einen Rückverweis, der solange wiederholt werden kann, bis die hinzugefügte Ausrüstung als vollständig erachtet wird.

Das Hinzufügen eines Bildes für die Detail-Seite der Fahrzeuge ist bei Fahrzeugen sogar wichtiger als bei einem Standort, aber die Funktionsweise ist trotzdem identisch. Dabei ist das Hinzufügen von Ausrüstungsobjekten und Bildern optional und kann auch später durch Bearbeitung hinzugefügt werden. Bei der Speicherung wird ebenfalls überprüft, ob das Fahrzeug bereits existiert, dabei soll später das Kennzeichen als Vergleichswert verwendet werden.

### 9.2.3 Fahrzeug einem Standort zuordnen

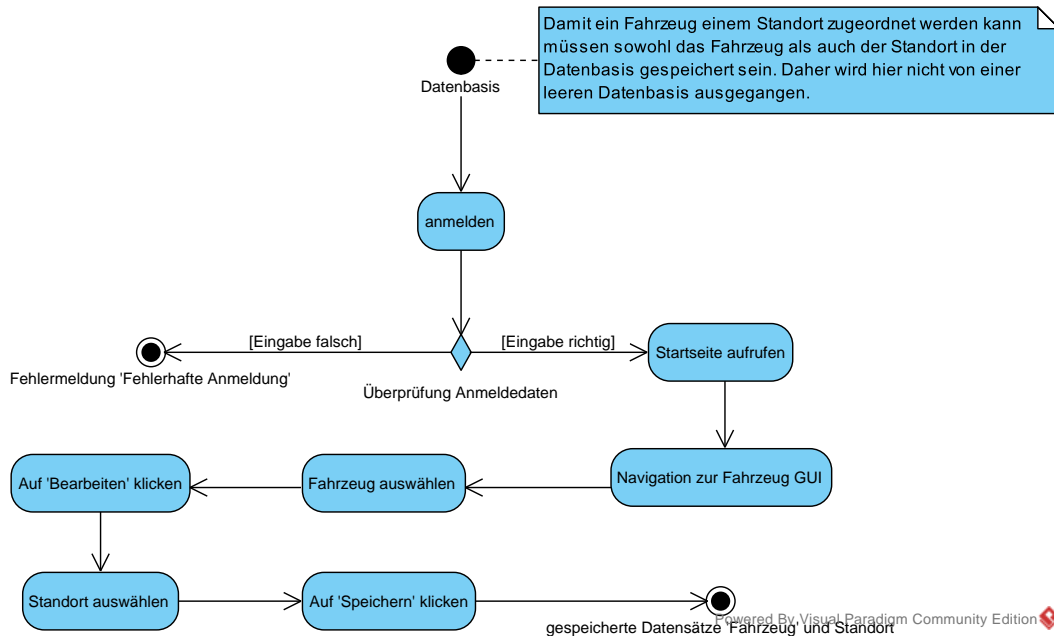


Abbildung 9.3: Aktivitätsdiagramm: Fahrzeug einem Standort zuordnen

Nachdem die Standorte und Fahrzeuge angelegt wurden, können die Fahrzeuge einem Standort zugewiesen werden. Die Zuweisung wird in Abbildung 9.3 dargestellt. Dazu wird davon ausgegangen, dass der Mitarbeiter noch nicht im System angemeldet ist. Daher ist der erste Schritt die Anmeldung im System. Sollte die Eingabe bei der Anmeldung fehlerhaft sein so wird eine Fehlermeldung angezeigt und der Nutzer kann nicht auf die Anwendung zugreifen. Falls die Anmeldung erfolgreich ist, wird dem Nutzer die Startansicht angezeigt. Um nun ein Fahrzeug einem Standort zuzuweisen, muss der Nutzer zunächst auf die Fahrzeug-GUI navigieren. Hier kann das Fahrzeug, welches

einem Standort zugewiesen werden soll, ausgewählt werden. Nachdem der Nutzer auf 'Bearbeiten geklickt hat, ist die Auswahl eines Standortes möglich. Sobald der Nutzer auf 'Speichern' klickt, wird sowohl der Datensatz des Standortes als auch der Datensatz des Fahrzeugs angepasst und gespeichert.





## 10.2 Pakete

### 10.2.1 model.buchung

Dieses Paket enthält alle Klassen aus dem Analyseklassendiagramm, welche der Klasse 'Buchung' zugeordnet werden können. Dazu gehören die Klassen 'Buchung', 'Rabattaktion', 'Rechnung' und 'Mahnung'. Damit auch die stornierten Buchungen noch eingesehen werden können, verfügt jede Buchung über ein Attribut, welches indiziert, ob die Buchung storniert wurde. Die Rabattaktion verfügt über ein Attribut für den prozentualen Preisnachlass. Die Rechnung ist durch Attribute für den Betrag der Rechnung und deren Fälligkeit definiert. Dazu gibt es eine Methode, um die Rechnung zum Kunden zu senden. Die Mahnung unterscheidet sich bezüglich der Attribute und methoden nicht von der Rechnung.

### 10.2.2 model.fahrzeug

Dieses Paket enthält alle Klassen, welche sich auf das Fahrzeug beziehen. Dazu gehören die Klassen 'Fahrzeug', 'Kennzeichen', 'Reifensatz', 'Reifen', 'Fahrzeugklasse', 'Bild' sowie 'Ausrüstung' und die Unterklassen der Ausrüstung ('Dachbox', 'Fahrradträger' und 'Hundetransportbox'). Das Fahrzeug besitzt Attribute, welche Auskunft über den Hersteller, das Baujahr und das Modell geben. Die Klasse 'Bild' speichert lediglich den Speicherort des Bildes. Die restlichen Klassen des Paketes verfügen jeweils über Attribute, um das Fahrzeug beziehungsweise den dargestellten Bestandteil des Fahrzeuges genauer zu beschreiben.

### 10.2.3 model.kunde

Dieses Paket enthält nur die Klasse 'Kunde' und die Klasse 'Vertrag' da dies Bereits alle Klassen sind, welche einen direkten Bezug zum Kunden haben. Die Klasse 'Vertrag' enthält nur den Pfad, unter welchem der Vertrag gespeichert wurde. Die Klasse 'Kunde' enthält hingegen alle Informationen, welche für die Kontaktierung des Kunden notwendig sind.

### 10.2.4 model.standort

Dieses Paket enthält alle Klassen, welche einen Standort sowie dessen Mitarbeiter darstellen. Dazu gehören neben der Klasse 'Standort' auch die Klassen 'Filiale', 'Mitarbeiter' und 'Rolle'. Die Rolle definiert dabei auf welche Objekte ein Mitarbeiter zugreifen darf. Sowohl für die Standorte als auch für die Filialen wird für statistische Erhebungen die Anzahl der Mitarbeiter angegeben. Der Mitarbeiter wird ähnlich wie der Kunde hauptsächlich durch Kontaktinformationen dargestellt. Jedoch wird außerdem noch das Gehalt des Mitarbeiters gespeichert.

### 10.2.5 model.utils

In diesem Paket befinden sich Klassen, welche von verschiedenen anderen Klassen verwendet werden. Dazu gehört neben der Klasse 'Adresse' auch die Klasse 'Backup'.

### 10.2.6 app

Hier befindet sich zum einen die zentrale Klasse 'Carsharing', welche die Main-Methode für das Starten der Anwendung beinhaltet. Diese Klasse wird als Singleton implementiert. Zum anderen befindet sich hier die 'EntityFactory' und der 'BuchungBuilder', welche für das Erzeugen von Instanzen der Klassen aus den Model-Paketen, welche in der Datenbasis gespeichert werden. Für die Buchungen wird extra ein Builder implementiert, da es sich hierbei um eine besonders komplexe Klasse handelt.

### 10.2.7 swe-utils

Dieses Paket enthält eigentlich noch wesentlich mehr Klassen, welche jedoch nicht mit dargestellt werden, da sie für die Anwendung irrelevant sind. Die enthalten Klassen beinhalten den 'CSVReader' und 'CSVWriter', welche im EntityManager für das Speichern der Daten eingesetzt werden. Für diese Persistierung der Daten ist das Interface 'IPersistable' enthalten. Außerdem befindet sich hier auch der 'PropertyManager' und der 'AppLogger'.

### 10.2.8 database

Dieses Paket beinhaltet nur eine Klasse. Dabei handelt es sich um den generischen 'EntityManager', welcher von der Klasse 'GenericEntityManager' aus dem Paket 'swe-utils' erbt. Diese Klasse wird als Singleton implementiert und ist für die Verwaltung der Entitäten verantwortlich.

### 10.2.9 gui

Dieses Paket beinhaltet alle Klassen, welche die Benutzeroberfläche betreffen. Dazu gehören neben einer Vielzahl von Klassen, welche auf 'GUI' enden auch die abstrakte Klasse 'GUIComponent' und die Klasse 'JComponent'. 'GUIComponent' erbt hierbei von 'JComponent' und alle auf 'GUI' endenden Klassen erben von 'GUIComponent'. Diese Klassen, welche auf 'GUI' enden stellen die einzelnen Ansichten der Anwendung dar. Im Sinne der Übersichtlichkeit wurde in diesem Diagramm auf die Detailtiefe bezüglich der Benutzeroberfläche verzichtet. Diese wird im nächsten Kapitel ab Seite 80 genauer beschrieben.

## 10.3 Entwurfsmuster

Für die Implementierung der Carsharingsoftware verwenden wir verschiedene Entwurfsmuster, welche das Auftreten von Problemen verhindern sollen. Solche Entwurfsmuster sind wiederverwendbare Schablonen zur Problemlösung. Im folgenden Abschnitt werden die verwendeten Entwurfsmuster genauer erläutert und erklärt, wie diese für die Implementierung eingesetzt werden.

### 10.3.1 EntityManager

Bezüglich des EntityManagers wurde sich dafür entschieden eine generische Klasse zu implementieren, welche alle gespeicherten Entitäten verwaltet. Dies hat den Vorteil, dass die Verwaltung der Entitäten sehr simpel dargestellt und Implementiert werden kann.

### 10.3.2 Builder

Für das Erstellen von komplexeren Klassen setzen wir in unserem Entwurf sogenannte Builder ein. Zu diesen Klassen gehören die Buchung, das Fahrzeug, der Standort sowie der Mitarbeiter. Durch den Builder wird ein schrittweises Erstellen von Instanzen dieser Objekte ermöglicht, wodurch die Erstellung dieser Objekte deutlich vereinfacht wird.

### 10.3.3 EntityFactory

Die EntityFactory wird verwendet, um Instanzen einzelner Klassen zu erstellen. Dazu verfügt die EntityFactory für jede Klasse, für welche eine Instanz erzeugt werden soll, über eine Methode, welche nach der Persistierung des Objektes mithilfe des EntityManagers eine Instanz zurückgibt. Um solche Instanzen zu erstellen, müssen den Methoden verschiedene Parameter übergeben werden. Diese Parameter werden im Diagramm zur Vereinfachung als 'properties' dargestellt.

### 10.3.4 Singleton

Ein Singleton stellt sicher, dass es zu einer Klasse nur genau ein einziges Objekt gibt. Diese Singletons werden in unserem Entwurf an verschiedenen Stellen eingesetzt. Bei diesen Stellen handelt es sich um die Klasse 'Carsharing', den EntityManager und die EntityFactory. Mit der Verwendung dieses Entwurfsmusters wird die Verhinderung des Auftretens von Dateninkonsistenzen beabsichtigt.

### 10.3.5 Beobachter

Mit Beobachtern kann auf die Veränderung von Objekten reagiert werden. Solche Beobachter wurden in unserem Entwurf lediglich passiv implementiert. Das heißt, dass die Beobachter nicht aktiv den Zustand eines konkreten Objektes abfragen, sondern nur dann benachrichtigt werden, wenn Änderungen aufgetreten sind. Dadurch ergibt sich eine größere Vielfalt der Abstraktionsmöglichkeiten und die Wiederverwendbarkeit von einzelnen Komponenten.

### **10.3.6 Kompositum**

Das Kompositum wird genutzt um Teil-Ganzes-Hierarchien darzustellen. In unserem Entwurfsklassendiagramm wird dies zum einen bei der Modellierung des Reifensatzes eines Fahrzeuges und zum anderen auch bei der Modellierung der Benutzeroberfläche genutzt. Ein Reifensatz besteht hierbei aus vier oder fünf Reifen, je nachdem, ob ein Ersatzrad vorhanden ist und bei der Benutzeroberfläche sind die Unteransichten ein Teil der Hauptansicht.

# 11 GUI-Entwurf

Eine Software kann noch so gut geplant und durchstrukturiert sein, wenn die Benutzeroberfläche jedoch unübersichtlich ist und nicht bedient werden kann, dann ist der gesamte Rest hinfällig. Daher ist es nötig sich bereits im Vorfeld eine mögliche Front-End-Gestaltung zu überlegen. In Anbetracht der Tatsache, dass der Programmentwurf in Java umgesetzt werden wird, haben wir beim GUI-Entwurf auf die Gestaltung von zu umständlichen und modernen Bestandteilen, wie man sie heute auf den meisten Webseiten sehen kann, verzichtet und uns auf die grundlegenden Bestandteile fokussiert.

Übergeordnet soll in der gesamten Anwendung ein einheitliches Design vorliegen. Darum haben wir uns bereits zu Beginn auf ein Farbschema abgestimmt. Aus unerklärlichen Gründen hat uns ein rötlich-rosa Farbschema überzeugt, sodass die Farben (in Hexwerten) **#DE639A**, **#E388B1**, **#D7A6B3**, **#F1E2E2**, **#707070** ihre Anwendung fanden.

## 11.1 Fahrzeuge verwalten

Grundsätzlich sollen alle Verwaltungsbereiche der Anwendung den selben oder einen sehr ähnlichen Aufbau haben, damit auch Mitarbeiter ohne besondere IT-Kenntnisse die Anwendung verwenden können. Neben einer Headline, die eine Überschrift und das Firmenlogo präsentiert, gibt es an dem linken Bildschirmrand die Navigationsleiste, mit der Mitarbeiter sich durch die unterschiedlichen Verwaltungsaufgaben klicken können. Der Hauptbereich einer jeden Seite wird in zwei Felder aufgeteilt. Das linke Feld besteht aus einer Liste, in der alle Einträge zum betroffenen Verwaltungsobjekt angezeigt werden, in diesem Fall alle Fahrzeuge. Bei Auswahl eines Fahrzeuges erscheint im rechten Feld eine Detail-Anzeige aller nötiger Informationen, die es zu dem Listeneintrag gibt. Neben einem Filter- und Suchfeld ist natürlich auch die Manipulation der Datensätze vorgesehen, sodass man einerseits neue Fahrzeuge anlegen kann. Wie in vorhergehenden Analysen angedacht gibt es dabei einerseits einen einfachen Button mit 'anlegen' und für erfahrenere Nutzer einen Button mit '+'. Sollte ein Listeneintrag ausgewählt sein, ist auch die Bearbeitung oder das Löschen des ausgewählten Datensatzes mittels entsprechender Buttons möglich.



Abbildung 11.1: Mock-Up: Liste an Fahrzeugen



Abbildung 11.2: Mock-Up: Detailansicht eines Fahrzeugs

### 11.1.1 Fahrzeug löschen

Die Interaktion zum Löschen des ausgewählten Fahrzeug-Datensatzes erfolgt über ein einfaches Pop-Up-Fenster. Der Anwender muss dieses Bestätigen, wenn er den Datensatz löschen möchte, um versehentliches Löschen auszuschließen. Nach dem Löschen wird der Anwender auf die Fahrzeug-Startseite zurückverwiesen, jedoch wird in der Liste der gelöschte Eintrag entfernt.

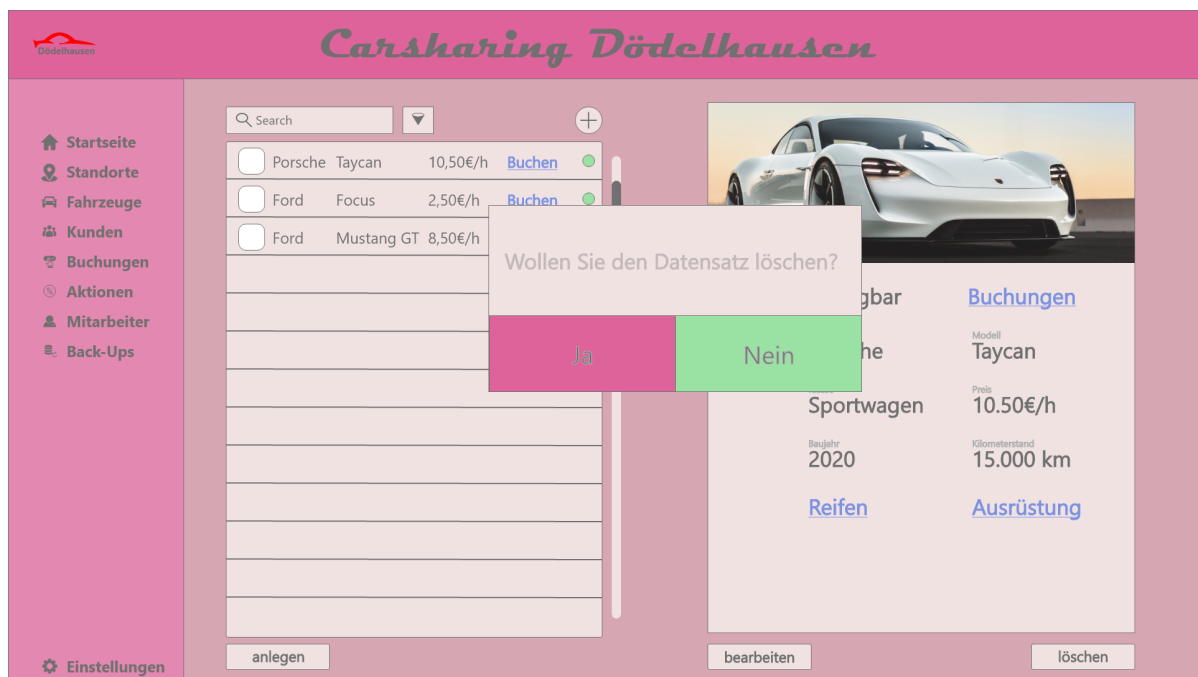


Abbildung 11.3: Mock-Up: Löschen eines Fahrzeugs



### 11.1.2 Fahrzeug bearbeiten

Das Bearbeiten eines Datensatzes ist so einfach wie möglich gestaltet. Der einzige Unterschied von der Bearbeitungs-GUI zur Detail-GUI ist, dass alle Felder, die bearbeitet werden können sollen, entweder zu einem Eingabefeld oder einer Drop-Down-Liste werden. Im folgenden Beispiel werden nur die Felder 'Reifen' und 'Ausrüstung' bearbeitbar. Da es sehr unwahrscheinlich ist, dass sich bei einem bereits vorhandenen Fahrzeug der Hersteller, das Modell, die Fahrzeugklasse oder das Baujahr ändern, soll es nicht bearbeitbar sein. Informationen wie Preis, Kilometerstand werden vom System vorgegeben und einfach nur angezeigt. Es ist aber möglich bei einem Fahrzeug die Reifen zu wechseln, z.B. von Sommer- auf Winterreifen oder auch die Ausrüstung zu ändern, da z.B. eine Dachbox angebracht und wieder abmontiert werden kann. Da bei beiden Feldern keine freie Eingabe möglich sein soll, sondern nur aus Vorhandenem ausgewählt werden soll, ist die Bearbeitung mittels Checkbox-Liste vorgesehen.



Abbildung 11.4: Mock-Up: Bearbeiten eines Fahrzeugs

### 11.1.3 Fahrzeug anlegen

Die letzte ausstehende Interaktionsmöglichkeit ist das Anlegen eines neuen Fahrzeugs. Auch hier wird nur die rechte Fläche für das Erstellen verwendet. Wie beim Bearbeiten erfolgt die Eingabe über einfache Eingabefelder oder Listen. Da das Fahrzeug noch nicht angelegt wurde, hat es auch keinen Status. Informationen über Hersteller, Modell, Baujahr und Kilometerstand sind frei einzutragen, wobei es ggf. auch möglich ist Felder wie Hersteller und Baujahr als Auswahlliste zu realisieren. Die Einordnung in die Klasse erfolgt ausschließlich über eine Auswahlliste und mit Auswahl eines Eintrages wird auch der Preis des Fahrzeuges gesetzt. Reifen und Ausrüstung sind Checkbox-Listen.

Als Vorbereitung auf die Webanwendung für Kunden und zur optisch schöneren Darstellung in der Anwendung ist es möglich ein Bild vom Fahrzeug hochzuladen. Diese Aktion ist nur für einen Admin möglich und kann jederzeit nachgeholt werden. Sollte kein Bild beim Anlegen hochgeladen werden, wird ein neutrales Platzhalter-Foto verwendet.

The mock-up shows the 'Carsharing Dödelhausen' application. On the left is a sidebar with navigation links. The main area has a top section with a search bar and a table of vehicles. Below this is a form for creating a new vehicle, and at the bottom are buttons for 'anlegen', 'speichern', and 'abbrechen'.

Hersteller	Modell	Preis
Porsche	Taycan	10,50€/h
Ford	Focus	2,50€/h
Ford	Mustang GT	8,50€/h

Form fields for creating a new vehicle:

- Status: ---
- Hersteller:
- Modell:
- Klasse:
- Preis: ---
- Baujahr:
- Kilometerstand:
- Reifen:
- Ausrüstung:

Abbildung 11.5: Mock-Up: Anlegen eines Fahrzeugs

## 11.2 Standorte verwalten

Es ist erwünscht, dass von mindestens zwei wesentlichen GUI-Komponenten Skizzen erstellt werden. Neben der 'Verwaltung von Fahrzeugen' wird die 'Verwaltung von Standorten' genauer ausgebaut. Der Seitenaufbau ist identisch zur Fahrzeugverwaltung. Auf der Standort-Startseite befindet sich auf der linken Hälfte eine Liste mit allen eingetragenen Firmen-Standorten. Auch die zusätzlichen Interaktionsmöglichkeiten wie Suchen, Filtern und Anlegen sind identisch zu anderen Verwaltungsbereichen. Die rechte Seite der Hauptfläche wird von einer interaktiven Karte eingenommen (möglich wäre z.B. eine Einbindung von [google.de/maps](https://www.google.de/maps)), auf der alle Standorte mit Markierungen eingetragen sind.



Abbildung 11.6: Mock-Up: Liste an Standorten

Sofern man eine Markierung auf der Karte oder einen Listeneintrag auswählt, wird man zu einer Standort-Detail-Seite geleitet. Ebenfalls identisch vom Aufbau zur Fahrzeug-Detail-Ansicht findet man auf der rechten Seite eine detaillierte Darstellung des Standortes. Dazu gehören Standortadresse, Parkplatzanzahl, das Vorhandensein von E-Ladesäulen und einer Filiale und ggf. deren Öffnungszeiten. Ebenfalls ist ein Foto angedacht. Die

Fläche auf der linken Bildschirmseite ist eine Liste, doch dieses Mal werden in dieser Liste alle Fahrzeuge angezeigt, die normalerweise am Standort vorhanden sind.



Abbildung 11.7: Mock-Up: Detailansicht von Standorten