



Entwicklung einer Carsharing-Anwendung

Programmentwurf 2022

im Rahmen der Prüfung zum
Bachelor of Science (B.Sc.)

des Studienganges Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

Clemens Richter, Johannes Peters

Abgabedatum:	20. Juni 2022
Bearbeitungszeitraum:	27.12.2021 - 20.06.2022
Kurs:	TINF20B1
Gutachter der Dualen Hochschule:	Dr.-Ing. Richard Lutz

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Quellcodeverzeichnis	VI
1 Einleitung	1
2 Lastenheft	2
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
3 Aufgaben	7
3.1 Analyse	7
3.2 Sequenzdiagramm und Aktivitätsdiagramm	7
3.3 Entwurf	8
3.4 Implementierung	9
4 Vereinfachung für den Programmentwurf	11
5 Analyse	12
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

6	Use-Case Diagramm	39
6.1	Vorüberlegung	39
6.2	Use-Case Analyse	41
6.3	Use-Case Diagramm	46
7	Analyse-Klassen-Diagramm	48
7.1	Klassen	48
7.2	Diagramm	49
8	Sequenzdiagramm	56
8.1	Aktionsbetrachtung: Buchung eines Fahrzeugs	57
8.2	Pseudo-Code	58
8.3	Diagramme	62
8.3.1	Buchung anlegen	62
8.3.2	Fahrt antreten	64
8.3.3	Fahrt beenden	65
9	Aktivitätsdiagramm	67
9.1	Pseudo-Code	67
9.2	Standort anlegen	73
9.3	Fahrzeug anlegen	75
9.4	Fahrzeug einem Standort zuordnen	77
10	Entwurfsklassendiagramm	79
10.1	Diagramm	79
10.2	Pakete	80
10.2.1	model.buchung	80
10.2.2	model.fahrzeug	80
10.2.3	model.kunde	80
10.2.4	model.standort	81
10.2.5	model.utils	81
10.2.6	app	81
10.2.7	util	81
10.2.8	swe-utils	82
10.2.9	database	82
10.2.10	view.gui	82
10.3	Entwurfsmuster	82
10.3.1	EntityManager	83
10.3.2	Builder	83
10.3.3	EntityFactory	83
10.3.4	Singleton	83
10.3.5	Beobachter	84
10.3.6	Kompositum	84

11 GUI-Entwurf	85
11.1 Diagramm	86
11.2 Mock-Up	87
11.2.1 Fahrzeuge verwalten	87
11.2.2 Standorte verwalten	92
11.2.3 Buchung verwalten	94
12 Besonderheiten	97
12.1 Detailtiefe	97
12.2 Benutzeroberfläche	98
12.3 Pseudocode	99
12.4 Übersichtlichkeit	99
12.5 Kundenorientierung	100

Abkürzungsverzeichnis

Abbildungsverzeichnis

6.1	Use-Case Diagramm	46
6.2	Use-Case 4: Fahrzeug verwalten	47
7.1	Analyseklassendiagramm	49
8.1	Buchung eines Termins	63
8.2	Antritt eines gebuchten Termins	64
8.3	Abschluss eines gebuchten Termins	65
8.4	Erstellung der Mahnungen	66
9.1	Aktivitätsdiagramm: Standort anlegen	73
9.2	Aktivitätsdiagramm: Fahrzeug anlegen	75
9.3	Aktivitätsdiagramm: Fahrzeug einem Standort zuordnen	77
10.1	Entwurfsklassendiagramm: Allgemein	79
11.1	Dark Theme Farbschema	85
11.2	Entwurfsklassendiagramm: GUI	86
11.3	Mock-Up: Liste an Fahrzeugen	88
11.4	Mock-Up: Detailansicht eines Fahrzeugs	88
11.5	Mock-Up: Löschen eines Fahrzeugs	89
11.6	Mock-Up: Bearbeiten eines Fahrzeugs	90
11.7	Mock-Up: Anlegen eines Fahrzeugs	92
11.8	Mock-Up: Liste an Standorten	93
11.9	Mock-Up: Detailansicht von Standorten	94
11.10	Mock-Up: Listenübersicht aller Buchungen	95
11.11	Mock-Up: Detailansicht beim Erstellen einer Buchung	96

Quellcodeverzeichnis

8.1	Szenario Buchung anlegen	58
8.2	Pseudocode für das Anlegen eines Kunden	60
8.3	Pseudocode für das Stornieren einer Buchung	61
9.1	Szenario Standort mit Fahrzeugen anlegen	67
9.2	Methode um Standort anzulegen	68
9.3	Methode um Fahrzeug anzulegen	70
9.4	Methode um ein Fahrzeug einem Standort zuzuordnen	71

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 ein Kunde, der die Dienstleistung unserer Firma in Anspruch nimmt. In diesem Sinne beschreibt ein Mitglied 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?

Es 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. Außerdem soll es für Organisatoren nicht möglich sein Bilder hochzuladen.

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 Buchungsdaten 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?

In der Anwendung sollen nur anwendungsbezogene Personaldaten wie Rechte innerhalb der Anwendung etc. verwaltet werden.

Welche Kundendaten werden verwaltet?

- Vor- und Nachname
- Adresse
- Kontaktdaten
- Vertragsdaten (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.?

Dies ist leider nicht möglich, da die zu zahlende Summe erst nach der Fahrt sicher feststeht. Daher wird standardmäßig per Rechnung gezahlt. In Zukunft soll es auch möglich sein per SEPA-Lastschriftmandat zu zahlen.

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.

- 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, da sie nicht in Kontakt mit der Desktopanwendung kommen sollen. Also ist eine Betrachtung des Kunden in Zusammenhang mit der Benutzeroberfläche der Desktopanwendung irrelevant.</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. Bei einer firmeninternen Umfrage wurde ermittelt, dass die Lieblingsfarbe unserer Mitarbeiter rosa-rot ist. Unsere Mitarbeiter würden sich dementsprechend sehr über ein rosa-rotes Farbschema freuen.</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">• Textgröße• Fenstergröße• Schriftart• Dark- oder Lightmodus <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. Bezüglich der Fenstergröße soll man die gängigsten Größen, wie auch bei anderen Programmen wählen 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 data-bbox="355 302 1340 383">Verwaltet werden sollen Mitarbeiter, Fahrzeuge, Standorte, Kunden, Buchungen, Rechnungen, Änderungen, Stornierungen, Mahnungen usw.</p> <p data-bbox="355 398 983 432">Welche Gruppe hat Zugriff auf welche Daten?</p> <ul data-bbox="400 519 1340 869" style="list-style-type: none"><li data-bbox="400 519 1340 553">• Administrator: Hat Zugriff auf alle Datenbestände und Funktionen<li data-bbox="400 593 1340 674">• Organisator: Fahrzeuge (bis auf Bilder), Kunden, Buchungen, Rechnungen<li data-bbox="400 714 1134 748">• Personalmitarbeiter: Personal/Kundenstammdaten<li data-bbox="400 788 1340 869">• Kunde: aktuell noch keine Zugriffe, später aber Fahrzeuge, Standorte, eigene Buchungen <p data-bbox="355 909 1340 1037">Welche Daten sollen für Mitarbeiter, Fahrzeuge, Standorte, Kunden, Buchungen, Rechnungen, Änderungen, Stornierungen und Mahnungen erhoben werden?</p> <ul data-bbox="400 1124 1340 1995" style="list-style-type: none"><li data-bbox="400 1124 1070 1158">• Mitarbeiter: Rollen und Kontaktmöglichkeiten<li data-bbox="400 1198 1340 1279">• Fahrzeug: Kennzeichen, Typ/Bezeichnung, Kategorie, Farbe, Baujahr, Buchungen<li data-bbox="400 1319 1310 1352">• Standort: Adresse, ansässige Fahrzeuge, maximale Fahrzeugzahl<li data-bbox="400 1393 687 1426">• Kunde: siehe 5.2.4<li data-bbox="400 1467 715 1500">• Buchung: siehe 5.2.3<li data-bbox="400 1541 1340 1668">• Rechnung: Inhalt der Buchung (Fahrzeug, Dauer), Gesamtkosten, Steuern, Rechnungsadresse, Rechnungsnummer, gefahrene Kilometer<li data-bbox="400 1709 1340 1792">• Änderung: Änderungszeitpunkt und Person, welche verantwortlich für Änderung ist<li data-bbox="400 1832 1340 1912">• Stornierung: Es ist nur relevant, ob eine Buchung storniert wurde oder nicht<li data-bbox="400 1953 794 1986">• Mahnung: siehe Rechnung
--------	--

/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 1097">Nein das gibt es nicht. Allerdings soll es Rabattaktionen geben, welche sich auf ganze Fahrzeugklassen beziehen, geben.</p> <p data-bbox="352 1106 1098 1144">Soll es auch Eventwochen und Rabattaktionen geben?</p> <p data-bbox="352 1153 1355 1285">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 1294 1355 1382">Es muss möglich sein, jederzeit erkennen zu können, welche angestellte Person einen Datensatz angelegt, geändert oder gelöscht hat.</p> <p data-bbox="352 1391 1161 1429">Müssen alle Änderungen sichtbar sein oder nur die Letzte?</p> <p data-bbox="352 1438 1355 1525">Für unseren Anwendungsfall reicht es aus, wenn nur die letzte Änderung zu sehen ist.</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.</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>Buchungen sollen auch spontan möglich sein. Allerdings können sie dann nicht mehr 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> <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 3 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>Es soll ein mal pro Woche ein komplettes Backup erstellt werden. Zusätzlich können von Administratoren jederzeit 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.

Die 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.

Die 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.

Die 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. Außerdem 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, da die Rechnung 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))

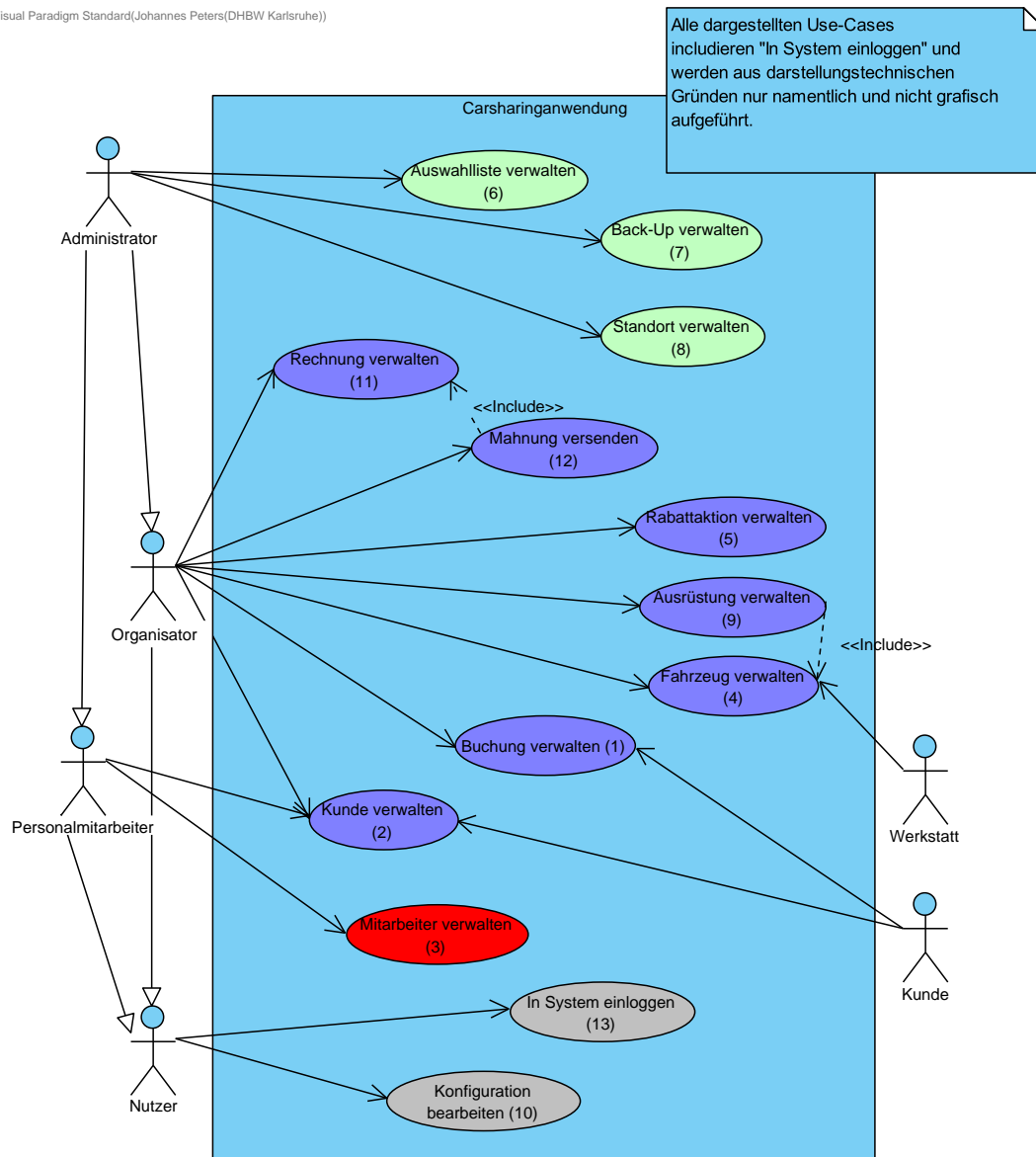


Abbildung 6.1: Use-Case Diagramm

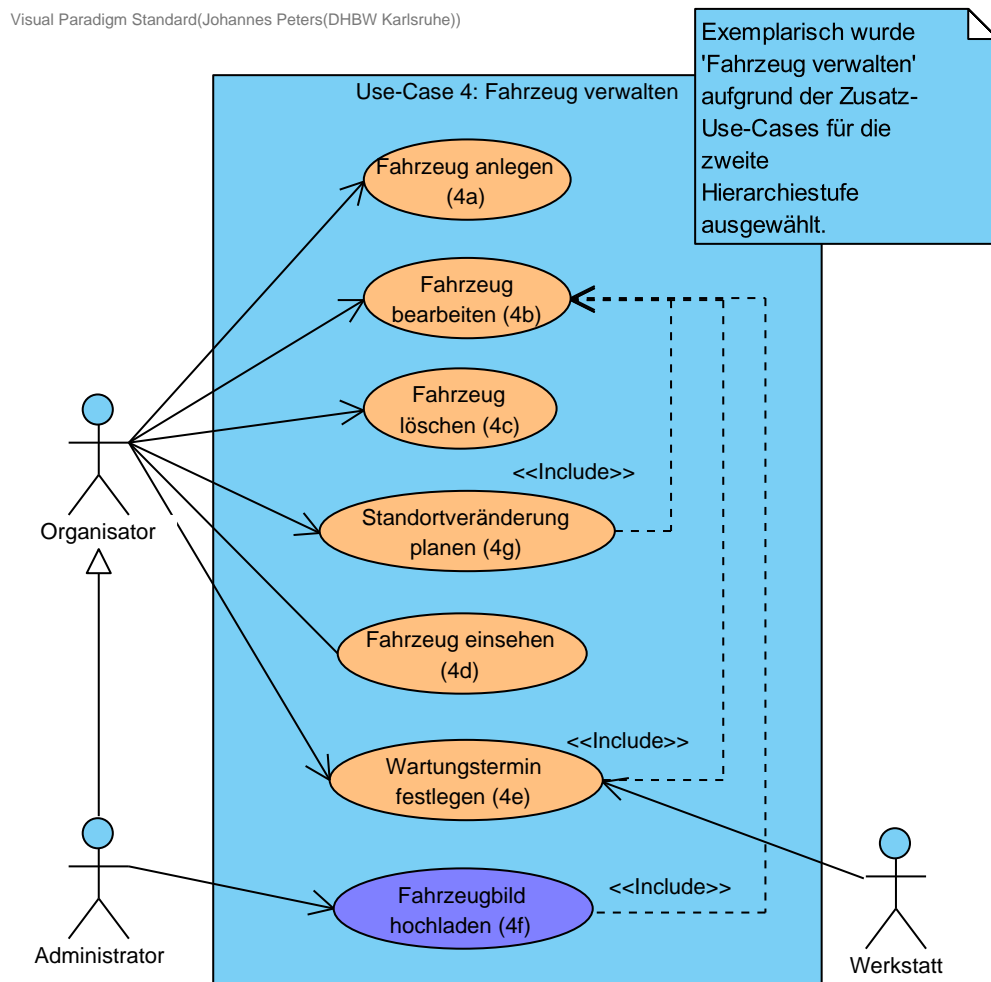


Abbildung 6.2: Use-Case 4: Fahrzeug verwalten

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

7.2 Diagramm

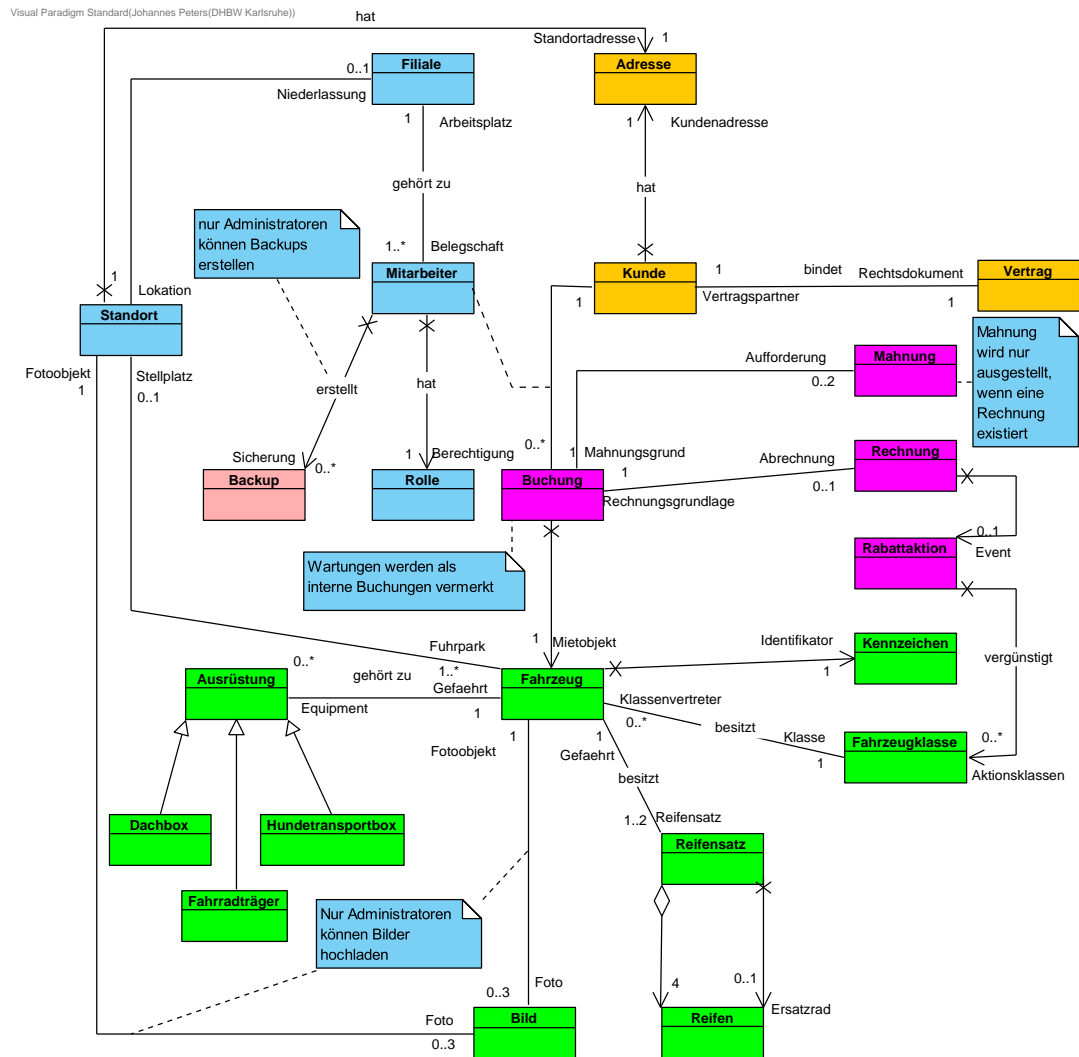


Abbildung 7.1: Analyseklassendiagramm

Kunde: Sobald sich ein neuer Kunde in einer der Filialen registrieren lässt, wird ein neuer Kunde angelegt. Die Kunden-Klasse führt alle nötigen Informationen zur Identifikation und Ausstellung von Rechnungen als Attribute zusammen. Von diesen Informationen sind Finanzdaten ausgenommen, da diese laut Anforderung durch ein anderes System verwaltet werden. Zu den Identifikationsinformationen gehört eine Anschrift, welche durch eine Referenz auf die Adresse-Klasse eingebracht wird.

Dabei kann jeder Kunde nur eine Adresse haben. Umgekehrt könnte zwar auch die gleiche Adresse auf mehrere Kunden verweisen (z.B. wenn mehrere Kunden im selben Haus wohnen wie es bei einer Familie der Fall sein könnte), doch ist diese Referenz für das Programm unwichtig und somit wird die Assoziation nur unidirektional dargestellt.

Für eine rechtliche Grundlage muss jeder Kunde einen Vertrag unterzeichnen, der digital eingescannt und abgespeichert werden soll. Die bidirektionale Verbindung weist jedem Kunden genau einen Kundenvertrag zu und natürlich wurde jeder Kundenvertrag von nur einem Kunden unterzeichnet.

Ein Kunde kann mehrere Buchungen durchführen, die er aber zum aktuellen Projektstand nicht selbst tätigen soll. Der Kunde äußert einen Terminwunsch an einen Mitarbeiter und der Mitarbeiter legt für den Kunden eine Buchung an, die auf den Kunden verweist. Ein Kunde darf mehrere Buchungen anlegen lassen, aber eine Buchung verweist nur auf einen Kunden. Aus diesem Grund fungiert die Mitarbeiter-Klasse als Koordinator.

Adresse: Die Adresse wird in Straße, Hausnummer, Wohnungszusatz, Ort und Postleitzahl aufgeteilt. Mit der Adress-Klasse werden die Adressen der Kunden und Filialen dargestellt. Da die Adresse-Klasse jeweils nur als Bestandteil anderer Klassen fungiert sind alle Assoziationen unidirektional auf die Adress-Klasse verweisend.

Vertrag: Die Vertrags-Klasse bildet die Verträge ab, welche die Kunden bindet. Der Vertrag soll unterschrieben eingescannt und als PDF-Datei im System abgelegt werden. Die einzigen zusätzlichen Angaben sind der Pfad und ein Vertragsdatum. Ein extra Dateiname wird nicht abgespeichert, da dieser dem Dateipfad entnommen werden kann.

Mitarbeiter: Die Klasse 'Mitarbeiter' stellt die Mitarbeiter des Unternehmens dar. Die Darstellung eines Mitarbeiters im System dient der Zugriffsbeschränkung der Anwendung mittels Rollen und der Identifikation von im System eingetragene Veränderungen. Jedem Mitarbeiter wird unidirektional eine Rolle zugewiesen. Da die Anwendung nicht auf Basis

einer Datenbank arbeitet und auf Abfragen wie die Gruppierung der Mitarbeiter nach vergebenen Rollen irrelevant ist, wurde nur eine unidirektionale und keine bidirektionale Assoziation angenommen.

Jeder Mitarbeiter hat einen festen Arbeitsplatz, welcher durch eine Referenz auf die Klasse 'Filiale' gekennzeichnet ist. Umgekehrt sind in einer Filiale mehrere Mitarbeiter beschäftigt.

Bereits bei der Klasse 'Kunde' angesprochen fungiert der Mitarbeiter als Koordinator zwischen Kunde und Buchung.

Weiterhin können Mitarbeiter, unter der Einschränkung, dass der Mitarbeiter die 'Admin'-Rolle hat, Backups erstellen. Der Ausdruck 'können' impliziert, dass nicht jeder Mitarbeiter ein Backup anlegt und dass auch der selbe Mitarbeiter mehrere Backups erstellen kann.

Rolle: Mit den Rollen sollen die Rechte innerhalb der Anwendung verteilt werden können. Dazu stellt die Klasse 'Rolle' neben dem Rollennamen auch ein Array aus Objekten, welches angibt, ob ein Nutzer mit dieser Rolle Zugriff auf die entsprechende Funktion bzw. auf eine GUI-Komponente haben soll. Die Klasse 'Rolle' referenziert keine anderen Klassen, sondern wird nur von der Klasse 'Mitarbeiter' referenziert.

Buchung: Die Klasse 'Buchung' 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 Fahrzeug gemietet werden soll. Pro Buchung soll ein Kunde nur ein einziges Fahrzeug buchen können.

Nach Abschluss einer Fahrt bzw. nach Ablauf eines Buchungszeitraums wird automatisch eine Rechnung erstellt. Die dadurch entstehende Assoziation zwischen Buchung und Rechnung ist bidirektional. Eine Buchung weist entweder keine Rechnung vor, solange der Termin noch offen ist oder genau eine Rechnung, wenn der Termin abgeschlossen wurde.

Sofern die Rechnung im vorgegebenen Zeitrahmen beglichen wird, ist damit der Buchungsvorgang abgeschlossen, doch sollte eine Zahlung überfällig sein, erzeugt das System auf Basis der Buchung eine oder zwei Mahnungen. Jede Mahnung verweist dabei eindeutig auf die zugehörige Buchung. Eine dritte Mahnung wird nicht ausgestellt, denn anstelle

dieser soll das schweizer Bankkonto eingefroren werden und andere rechtliche Schritte werden eingeleitet.

Fahrzeug: Die Klasse Fahrzeuge ist für das Abbilden der Fahrzeuge zuständig und das Kernobjekt der Anwendung. Jedem Fahrzeug wird genau eine Fahrzeugklasse zugeordnet und in diesem Fall ist die Assoziation bidirektional, da umgekehrt z.B. auch eine Suche nach Fahrzeugen einer bestimmten Fahrzeugklasse möglich sein soll.

Jedes Fahrzeug im Fuhrpark des Unternehmens benötigt eine rechtlich eindeutige Identifikation mittels Nummernschild, weshalb jedes Fahrzeug auf genau ein Kennzeichen verweist. Eine umgekehrte Verknüpfung ist nicht nötig für ein einfaches Buchungssystem.

Um die Fahrtüchtigkeit der Fahrzeuge zu allen Jahreszeiten gewährleisten zu können, müssen die Fahrzeuge mit den passenden Saisonreifen ausgerüstet sein. Um dies im Überblick zu behalten, verweist ein Fahrzeug auf einen Reifensatz. Jedes Fahrzeug hat einen Sommer- und Winterreifensatz. Um die Zuordnung zu genau dem zugehörigen Fahrzeug sicherzustellen, ist die Verbindung bidirektional.

Für ein optisch ansprechendes GUI gibt es die Möglichkeit beim Anlegen eines neuen Fahrzeuges auch ein oder mehrere Bilder hochzuladen. Dieses Angebot ist optional und sollte kein Bild hochgeladen werden, dann wird ein 'Placeholder'-Bild eingefügt. Das Hochladen von Bildern ist nach Aufgabenanalyse und -anforderung ebenfalls als eine Administratorfunktion deklariert. Um nicht übermäßig viele Bilder hochzuladen, ist ein Limit auf drei Bilder gesetzt.

Beim Ausleihen der Fahrzeuge für besondere Anlässe kann es auch sein, dass zusätzliche Ausrüstung erwünscht ist. Die Verbindung zur Ausrüstung ist als bidirektional gewählt, nicht nur vom Fahrzeug aus ersichtlich sein soll welche Ausrüstung vorhanden ist, sondern auch um zu verhindern, dass die selbe Ausrüstung im System ausversehen auf mehreren Fahrzeugen angebracht werden soll.

Die letzte Assoziation für ein Fahrzeug verweist auf den Standort. Um ein Fahrzeug zu buchen, muss bekannt sein, wo dieses Fahrzeug parkt. Von der andern Seite aus betrachtet gibt es am Standort nur eine gewisse Anzahl an Parkplätzen, sodass bekannt sein muss, welche Fahrzeuge sich zu dem bestimmten Zeitpunkt vor Ort befinden.

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. Dabei ist es denkbar, dass eine Fahrzeugklasse existiert, das Carsharing-Unternehmen aber keinen Klassenvertreter im Fuhrpark besitzt.

Bild: Die Klasse Bild stellt allgemein alle Bilder in der Anwendung dar. Jedes Bild ist einem anderen Element (zum Beispiel Fahrzeug, Standort, ...) zugeordnet. Als weitere Attribute besitzt diese Klasse den Titel des Bildes sowie den Pfad, an welchem das Bild liegt.

Reifensatz: Ein Reifensatz dient der Unterscheidung zwischen Sommer- und Winterreifen und verweist auf ein bestimmtes, zugehöriges Fahrzeug, sodass jedes Fahrzeug zwei Reifensätze haben sollte. Da es z.B. auch Ganzjahresreifen gibt, kann ein Fahrzeug entweder ein oder zwei Reifensätze vorweisen. Jeder Reifensatz besteht 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, Saison 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.

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 in Form eines PDF-Dokuments zu erstellen und an den Kunden per E-Mail zu versenden. Solch eine Rechnung referenziert dabei immer auf die zugrundeliegende Buchung.

Der Preis auf der Rechnung kann von Rabattaktionen beeinflusst werden. Daher steht die Rechnung in Assoziation zur Rabattaktion.

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. Ähnlich der Erstellung einer

Rechnung wird ein PDF-Dokument erzeugt, welches im Fall einer Mahnung jedoch manuell ausgedruckt und mittels Post versendet wird. Da eine Mahnung stets auf einer Buchung basiert, wird die zugrundeliegende Buchung referenziert.

Standort: Das Unternehmen ist an verschiedensten Standorten tätig. Grundlegend ist ein Standort eine Betriebsfläche des Carsharing-Unternehmens und ist in allen Fällen eine Lokation, an der Fahrzeuge ausgeliehen werden können. Die Anzahl der Fahrzeuge ist hierbei nicht beschränkt.

Damit der Standort für die Kunden erreichbar ist, wird eine Adresse vorausgesetzt, auf die eine Klasseninstanz des Standortes wie auch bei Kunde unidirektional verweist.

Es ist bei größeren oder wichtigeren Standorten möglich, dass sich dort auch eine Filiale des Unternehmens befindet, an die sich Kunden wenden können. Ein Standort **kann** eine Filiale referenzieren, doch umgekehrt **muss** eine Filiale allein wegen der Adressreferenz auf einen Standort verweisen.

Des Weiteren wird der Standort durch eine Beschreibung und einen Namen beschrieben.

Filiale: Die Klasse 'Filiale' stellt eine Filiale des Unternehmens dar. Dafür wird die Filiale mit dem Standort in Assoziation gesetzt.

In jeder Filiale sind Mitarbeiter des Unternehmens angestellt. Das heißt, dass die Belegschaft einer Filiale mindestens ein Mitarbeiter ist, wobei theoretisch nach oben keine Schranke gesetzt ist. Als wichtige Information hat diese Klasse das Attribut 'Öffnungszeiten'.

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

Es ist möglich, dass z.B. der Rabatt nicht global auf alle Fahrzeuge gewährleistet werden soll, sondern auf einzelnen oder mehrere Fahrzeugklassen. Die dabei entstehende Assoziation ist unidirektional, da eine Rabattaktion auf Fahrzeugklassen verweist, doch die Fahrzeugklasse nichts von einer Verbindung zu einer Rabattaktion kennen muss, da der Preisnachlass erst zur Rechnung erstellt wird.

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. Es gab die Überlegung, ob auch noch der Mitarbeiter verknüpft werden soll, der das Backup anlegt, doch muss diese Aufgabe nicht vom System übernommen werden, da die erstellte Backupdatei im Dateisystem diese Information des Erstellers automatisch speichert.

Ausrüstung: Die Klasse 'Ausrüstung' umfasst die Ausrüstung eines Fahrzeugs. Dabei handelt es sich nicht um die Ausstattung des Fahrzeugs (Klimaanlage, Panoramaschiebedach, etc.), sondern um Zusätze wie 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.

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 dem der Fahrradträger beladen werden darf.

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

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

Im folgenden Abschnitt wird der Pseudocode für das Anlegen einer Buchung anhand der Methode BUCHUNG-ANLEGEN beschrieben. Da das Anlegen einer Buchung einen großen Umfang hat werden bestimmte Abschnitte in separate Methoden ausgelagert, welche ebenfalls erläutert werden.

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

```
29         Organisator wählt "-" bei Rabattaktion aus.
30         ENDE WENN
31     ENDE SOLANGE
32
33     Organisator klickt auf "Buchung überprüfen"
34     WENN Fahrzeug nicht verfügbar
35         SOLANGE Fahrzeug nicht verfügbar
36             Zeige Fehlermeldung "Das gewählte Fahrzeug ist in ↗
                ↳ diesem Zeitraum nicht verfügbar"
37             Organisator wählt Fahrzeug aus
38         ENDE SOLANGE
39     ENDE WENN
40     Buchung speichern
41
42     WENN Kunde möchte Buchung stornieren
43         FUEHRE AUS BUCHUNG-STORNIEREN
44     ENDE WENN
45 ENDE BUCHUNG-ANLEGEN
```

Listing 8.1: Szenario Buchung anlegen

Wenn der Kunde einen Terminwunsch bei einem Organisator äußert startet die Methode BUCHUNG-ANLEGEN. Zuerst öffnet der Organisator das Buchungsformular. Daraufhin wird dem Organisator angezeigt, dass er alle Eingabefelder ausfüllen muss, um eine Buchung anzulegen. Durch Interaktion mit dem Kunden füllt der Organisator nun das Formular nach den Wünschen des Kunden aus. Dazu wird der Kunde, ein Standort sowie Fahrzeug, ein Start- und Endtermin ausgewählt. Optional ist auch das auswählen einer Rabattaktion möglich. Falls der Kunde keinen Rabattcode hat, wählt der Organisator als Rabattaktion '-' aus. Wenn der Kunde noch nicht registriert ist muss zuerst die Methode KUNDE-ANLEGEN ausgeführt werden. Diese Methode wird im Anschluss genauer erklärt. Um Fehler bei der Eingabe zu vermeiden, wird stets überprüft, ob der Starttermin vor dem Endtermin liegt und bei der Auswahl des Fahrzeuges muss der Organisator zuerst den Standort angeben. Nachdem der Standort angegeben wurde, werden alle Fahrzeuge des entsprechenden Standortes geladen. Somit wird verhindert, dass der Organisator ein Fahrzeug von einem verkehrtem Standort auswählt. Sobald alle Felder ausgefüllt sind kann der Organisator auf 'Buchung überprüfen' klicken. Daraufhin

wird überprüft, ob das Fahrzeug für den angegebenen Zeitraum verfügbar ist. Sollte das Fahrzeug nicht verfügbar sein, so wählt der Organisator ein neues Fahrzeug aus bis der Zeitraum passt. Wenn das Fahrzeug verfügbar ist, wird die Buchung gespeichert. Nun hat der Kunde noch die Option die Buchung wieder zu stornieren. Sollte sich der Kunde rechtzeitig dazu entscheiden die Buchung zu stornieren, so wird die Methode 'BUCHUNG-STORNIEREN' ausgeführt. Falls der Kunde die Buchung nicht stornieren möchte terminiert die Methode.

```
1  BEGINN KUNDE-ANLEGEN
2      Formular für das Anlegen eines Kunden zeigen
3      Solange nicht alle Felder ausgefüllt sind
4          Zeige Fehlermeldung "Bitte alle Felder ausfüllen"
5          Organisator trägt Vorname des Kunden ein
6          Organisator trägt Nachname des Kunden ein
7          Organisator trägt Adresse des Kunden ein
8          Organisator trägt Mail-Adresse des Kunden ein
9          Organisator trägt Telefonnummer des Kunden ein
10         Organisator trägt Nummer des Schweizer Bankkontos des ↗
           ↘ Kunden ein
11         Organisator klickt auf "Kunde anlegen"
12         WENN Schweizer Bankkonto valide
13             Sende Verifizierungs-Mail an die angegebene ↗
                 ↘ Mail-Adresse
14             Kunde klickt auf "Verifizieren"
15             Kunde Speichern
16         SONST
17             Sende Mail mit Anzeige wegen Betrug an Polizei
18             Abbruch des Anlegens
19         ENDE WENN
20 ENDE KUNDE-ANLEGEN
```

Listing 8.2: Pseudocode für das Anlegen eines Kunden

Diese Methode wird ausgeführt, wenn der Kunde noch nicht im System gespeichert ist. Zuerst öffnet sich hierbei das Formular für das Anlegen eines neuen Kunden. Solange der Organisator nicht alle Felder ausgefüllt hat, wird eine Fehlermeldung angezeigt, dass alle Eingabefelder ausgefüllt sein müssen um einen Kunden anzulegen. In Zusammenarbeit

mit dem Kunden trägt der Organisator nun den Vornamen, Nachnamen, die Adresse, die Mail-Adresse, die Telefonnummer und die Nummer des Schweizer Bankkontos des Kunden ein. Wenn alle Felder ausgefüllt wurden, kann der Organisator auf 'Kunde anlegen' klicken. Daraufhin wird das Schweizer Bankkonto auf Validität überprüft, bevor eine Mail an die angegebene Mail-Adresse gesendet wird. Dadurch wird eine Zwei-Faktor-Authentifizierung ermöglicht. Der Kunde klickt nun auf 'Verifizieren' in der Mail, woraufhin die Daten gespeichert werden. Falls das Schweizer Bankkonto nicht valide ist, wird automatisch eine Mail an die Polizei geschickt, um Anzeige wegen Betrugs zu erstatten.

```
1  BEGINN BUCHUNG-STORNIEREN
2      Organisator klickt auf "Kunden"
3      Zeige Übersicht aller Kunden an
4      Organisator wählt Kunde aus
5      Zeige Kundendetails
6      Organisator wählt Buchung aus
7      Zeige Buchungsdetails
8      Organisator klickt "Stornieren"
9      Zeige Warnmeldung "Wollen Sie die Buchung wirklich ↯
        ↵ stornieren?"
10     Organisator klickt erneut "Stornieren"
11     Buchung wird als storniert markiert
12  ENDE BUCHUNG-STORNIEREN
```

Listing 8.3: Pseudocode für das Stornieren einer Buchung

Die Methode BUCHUNG-STORNIEREN wird immer dann ausgeführt, wenn eine Buchung bereits angelegt wurde und der Kunde sich dazu entscheidet, diese Buchung rückgängig zu machen. Hierzu klickt der Organisator zuerst auf 'Kunden'. Dadurch wird die Übersicht aller Kunden geöffnet. Nun kann der Organisator den Kunden auswählen, der seine Buchung stornieren möchte. Sobald der Organisator einen Kunden ausgewählt hat wird die Detailansicht für diesen Kunden geöffnet. Hier sind auch die Buchungen des Kunden zu sehen. Der Organisator wählt eine Buchung aus. Daraufhin werden die Details der Buchung angezeigt. Hier kann der Organisator auf 'Stornieren' klicken. Wenn der Organisator auf 'Stornieren' klickt wird ihm eine Warnmeldung angezeigt, bei welcher er bestätigen muss, dass er die Buchung stornieren möchte. Klickt der Organisator erneut auf stornieren, so wird der Datensatz der Buchung als storniert markiert.

8.3 Diagramme

8.3.1 Buchung anlegen

Das Anlegen einer Buchung wird in Abbildung 8.1 auf Seite 63 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.

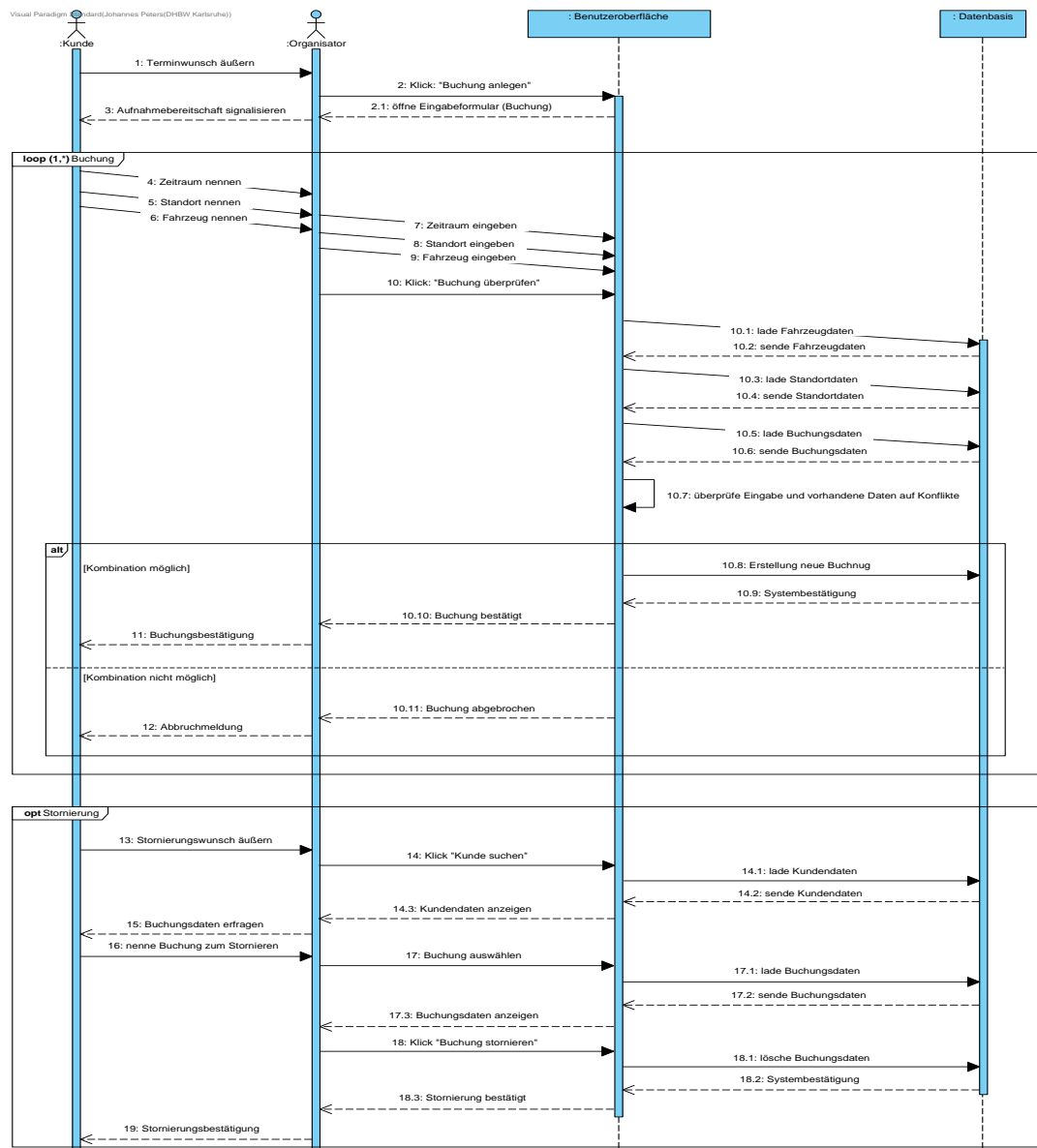


Abbildung 8.1: Buchung eines Termins

8.3.2 Fahrt antreten

Das Antreten einer Fahrt wird in Abbildung 8.2 auf Seite 64 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.

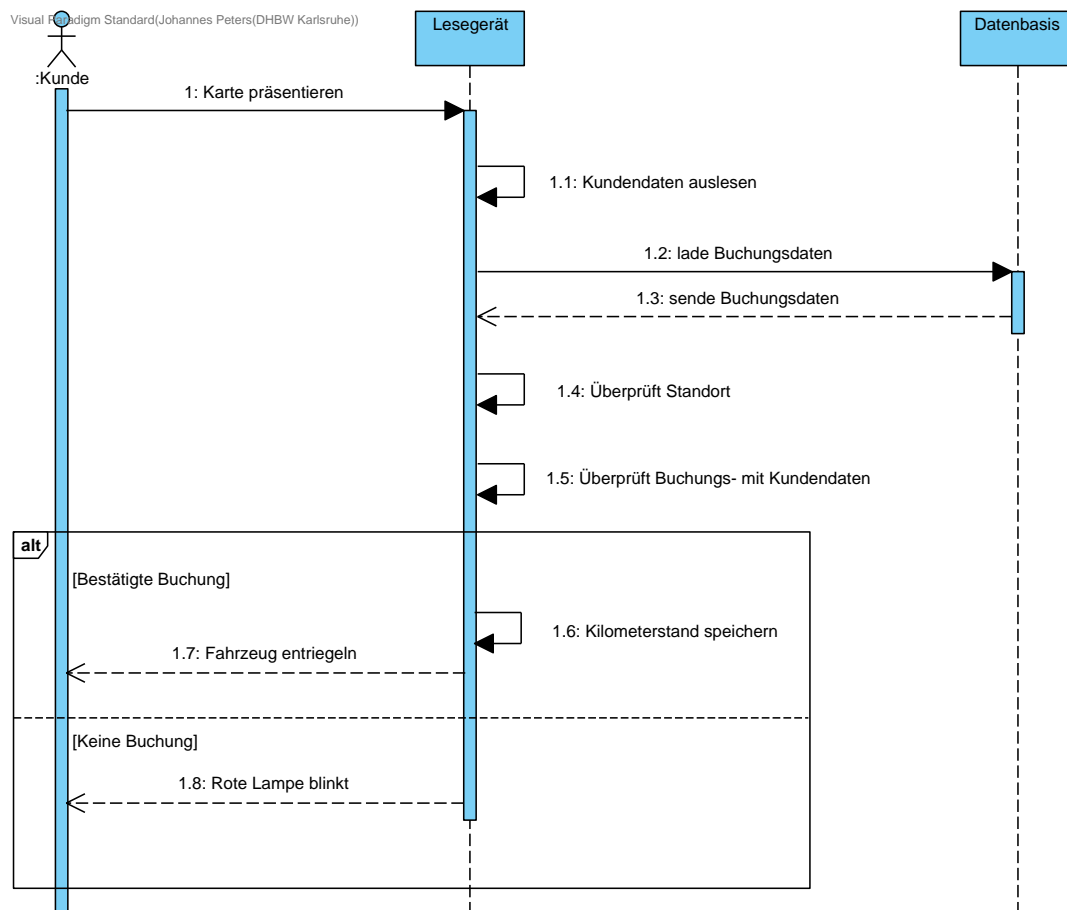


Abbildung 8.2: Antritt eines gebuchten Termins

8.3.3 Fahrt beenden

Das Beenden einer Fahrt wird in Abbildung 8.3 auf Seite 65 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.

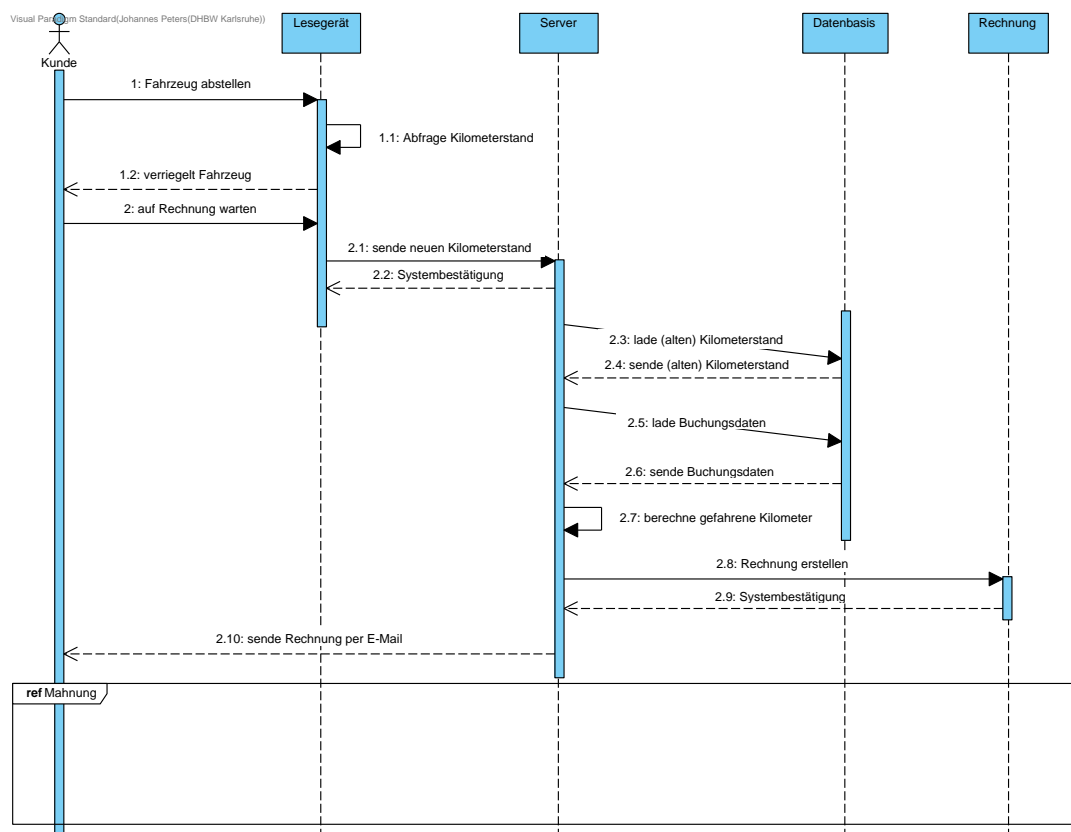


Abbildung 8.3: Abschluss eines gebuchten Termins

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 65 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.

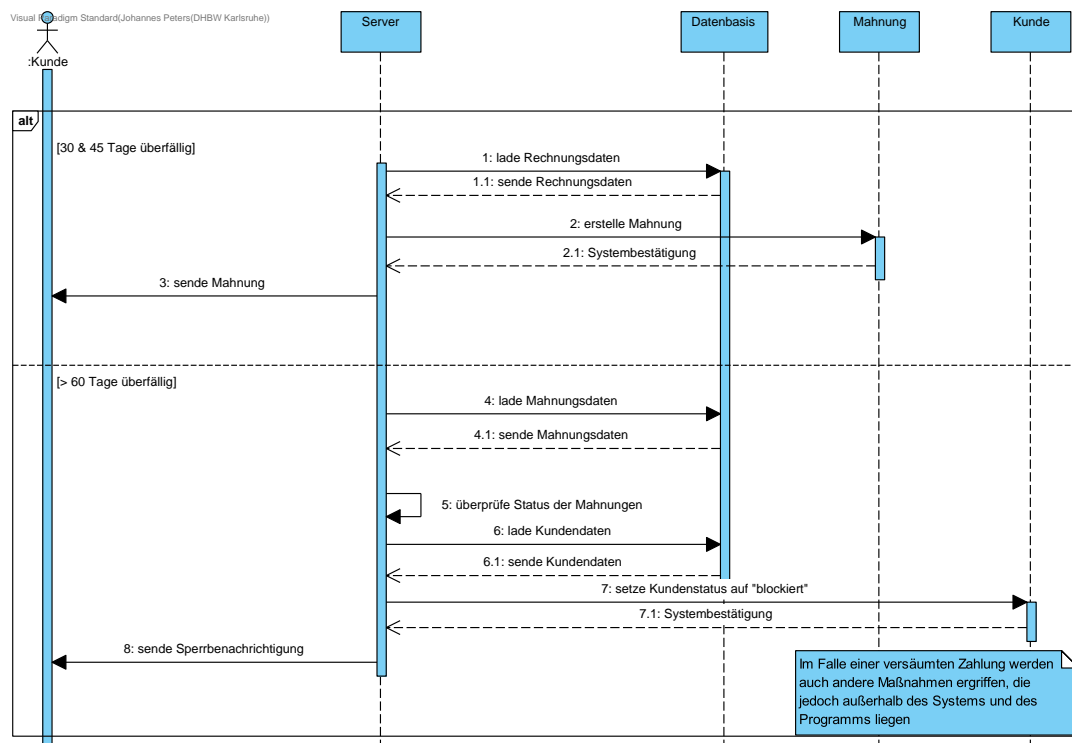


Abbildung 8.4: Erstellung der Mahnungen

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

Im folgenden Diagramm wird der Pseudocode für das Anlegen eines Standortes mit neuen Fahrzeugen erläutert. Da dies einen großen Umfang hat, wurde der Pseudocode in mehrere Methoden unterteilt, welche separat erklärt werden. Zu diesen Methoden gehören STANDORT-MIT-FAHRZEUGEN-ANLEGEN, STANDORT-ANLEGEN, FAHRZEUG-ANLEGEN und ZUORDNEN.

Die Methode STANDORT-MIT-FAHRZEUGEN-ANLEGEN ist die Ausgangsmethode, welche die anderen Methoden aufruft. Für das Anlegen eines neuen Standortes mit Fahrzeugen muss zuerst der Standort angelegt werden. Dazu wird die Methode STANDORT-ANLEGEN ausgeführt. Daraufhin können Fahrzeuge angelegt werden und dem Standort zugeordnet werden. Dies geschieht mithilfe der Methoden FAHRZEUG-ANLEGEN und ZUORDNEN. Diese beiden Methoden können solange wiederholt werden, bis alle Fahrzeuge die dem Standort zugeordnet werden sollen auch dem Standort zugeordnet sind oder die maximale Fahrzeuganzahl des Standortes erreicht wurde.

```
1  BEGINN STANDORT-MIT-FAHRZEUGEN-ANLEGEN
2      FUEHRE AUS STANDORT-ANLEGEN
3      SOLANGE Nicht alle Fahrzeuge angelegt und zugeordnet UND ↯
           ↳ maximale Fahrzeugzahl des Standortes nicht überschritten
4          FUEHRE AUS FAHRZEUG-ANLEGEN
5          FUEHRE AUS ZUORDNEN
6      ENDE SOLANGE
```

7 **ENDE** STANDORT-MIT-FAHRZEUGEN-ANLEGEN

Listing 9.1: Szenario Standort mit Fahrzeugen anlegen

Die erste Methode, welche in STANDORT-MIT-FAHRZEUGEN-ANLEGEN aufgerufen wird, ist STANDORT-ANLEGEN. Hier wird davon ausgegangen, dass der Mitarbeiter noch nicht angemeldet ist. Daher muss der Mitarbeiter sich erst mit seinem Nutzernamen und Passwort anmelden. Gelingt die Anmeldung, so landet der Mitarbeiter auf der Startansicht. Von da aus kann der Mitarbeiter zur Standortansicht navigieren. Dort kann ein neuer Standort erstellt werden. Dazu muss der Mitarbeiter nur auf das '+' oder auf den Link 'neuer Standort' klicken. Dadurch wird das Formular zum Anlegen eines Standortes geöffnet. Hier füllt der Mitarbeiter nun ähnlich wie beim Anlegen eines Kunden auf Seite 60 alle Eingabefelder aus. Dazu gehören Name, Beschreibung, Adresse und die maximale Anzahl an Fahrzeugen. Falls der Mitarbeiter die Administrator-Rolle hat, kann er noch Bilder hochladen. Anschließend kann der Standort gespeichert werden. Bevor der Standort aber in die Datenbasis gespeichert wird, wird zuerst überprüft, ob der Standort bereits existiert.

```
1 BEGINN STANDORT-ANLEGEN
2     Mitarbeiter gibt Nutzernamen und Passwort ein
3     Mitarbeiter klickt auf anmelden
4     WENN Nutzernamen nicht existiert ODER Passwort falsch
5         Zeige Fehlermeldung "Fehlerhafte Anmeldung"
6     SONST
7         Zeige Startseite
8     ENDE WENN
9     Mitarbeiter klickt auf Standorte
10    Zeige Standort-GUI
11    Mitarbeiter klickt auf Button "+" oder auf Link "neuen ↗
        ↳ Standort"
12    Zeige Formular für das Anlegen eines Standortes
13
14    SOLANGE nicht alle Felder ausgefüllt sind
15        Zeige Fehlermeldung "Bitte füllen Sie alle Felder aus, ↗
            ↳ um einen neuen Standort anzulegen."
16        Mitarbeiter gibt Namen an
17        Mitarbeiter gibt Beschreibung an
```

```
18      Mitarbeiter gibt Straße an
19      Mitarbeiter gibt Hausnummer an
20      Mitarbeiter gibt Postleitzahl an
21      Mitarbeiter gibt Stadt an
22      Mitarbeiter gibt maximale Fahrzeuganzahl an
23  ENDE SOLANGE
24
25  WENN Mitarbeiter hat Administrator-Rolle
26      WENN Mitarbeiter möchte Bilder hochladen
27          Mitarbeiter klickt auf "Bild hochladen"
28          Mitarbeiter wählt Bild aus
29      ENDE WENN
30  ENDE WENN
31
32  Mitarbeiter klickt auf "Standort speichern"
33
34  WENN Standort bereits vorhanden
35      Zeige Fehlermeldung "Standort bereits vorhanden"
36  SONST
37      Zeige Bestätigung für das Speichern des Datensatzes
38  ENDE WENN
39  ENDE STANDORT-ANLEGEN
```

Listing 9.2: Methode um Standort anzulegen

Nachdem der Standort angelegt wurde müssen zunächst die Fahrzeuge angelegt werden. Dies geschieht mit der Methode FAHRZEUG-ANLEGEN. Zuerst wird hier überprüft, ob der Mitarbeiter bereits angemeldet ist. Falls dies nicht der Fall ist, muss sich der Mitarbeiter erst anmelden. Um ein Fahrzeug anzulegen muss der Mitarbeiter zunächst zur Fahrzeugansicht navigieren. Dort kann der Mitarbeiter wie beim Anlegen des Standortes (siehe Seite 68) das Anlegeformular öffnen. Hier muss das Kennzeichen, die Farbe, das Modell, die Beschreibung, die Marke, das Baujahr, die Fahrzeugklasse sowie ein Refiensatz angegeben werden. Außerdem kann optional auch Ausrüstung hinzugefügt werden. Falls der Mitarbeiter die Administrator-Rolle hat kann er Bilder für das Fahrzeug hochladen. Im Anschluss kann das Fahrzeug gespeichert werden, indem der Mitarbeiter auf 'Speichern'

klickt. Falls dieses Fahrzeug bereits existiert wird hier ebenfalls eine Fehlermeldung angezeigt. Ansonsten wird eine Bestätigung des Speicherns angezeigt.

```
1  BEGINN FAHRZEUG-ANLEGEN
2      WENN Mitarbeiter noch nicht angemeldet
3          Mitarbeiter gibt Nutzernamen und Passwort ein
4          Mitarbeiter klickt auf anmelden
5      WENN Nutzernamen nicht existiert ODER Passwort falsch
6          Zeige Fehlermeldung "Fehlerhafte Anmeldung"
7      SONST
8          Zeige Startseite
9      ENDE WENN
10 ENDE WENN
11 Mitarbeiter klickt auf "Fahrzeuge"
12 Zeige Fahrzeug-GUI
13 Mitarbeiter klickt auf Button "+" oder auf Link "neues ↗
    ↳ Fahrzeug"
14 Zeige Formular für das Anlegen eines Standortes
15
16 SOLANGE nicht alle Felder ausgefüllt sind
17     Zeige Fehlermeldung "Bitte füllen Sie alle Felder aus, ↗
        ↳ um einen neuen Standort anzulegen."
18     Mitarbeiter gibt Kennzeichen an
19     Mitarbeiter gibt Farbe an
20     Mitarbeiter gibt Modell an
21     Mitarbeiter gibt Beschreibung an
22     Mitarbeiter wählt Marke aus
23     Mitarbeiter gibt Baujahr an
24     Mitarbeiter wählt Fahrzeugklasse aus
25     Mitarbeiter wählt Reifensatz aus
26 ENDE SOLANGE
27
28 SOLANGE Fahrzeug verfügt über nicht zugewiesene Ausrüstung
29     Mitarbeiter wählt Ausrüstung aus
30 ENDE SOLANGE
31
32 WENN Mitarbeiter hat Administrator-Rolle
```

```
33      WENN Mitarbeiter möchte Bilder hochladen
34          Mitarbeiter klickt auf "Bild hochladen"
35          Mitarbeiter wählt Bild aus
36      ENDE WENN
37  ENDE WENN
38
39  Mitarbeiter klickt auf "Fahrzeug speichern"
40
41  WENN Fahrzeug bereits vorhanden
42      Zeige Fehlermeldung "Fahrzeug bereits vorhanden"
43  SONST
44      Zeige Bestätigung für das Speichern des Datensatzes
45  ENDE WENN
46 ENDE FAHRZEUG-ANLEGEN
```

Listing 9.3: Methode um Fahrzeug anzulegen

Nun muss das Fahrzeug nur noch einem Standort zugeordnet werden. Dies geschieht mit der Methode ZUORDNEN. Dabei wird zuerst wieder überprüft, ob der Nutzer noch angemeldet ist. Falls nicht muss sich der Mitarbeiter erst wieder im System anmelden. Wenn der Mitarbeiter angemeldet ist, muss er zuerst zur Ansicht für die Fahrzeuge navigieren. Dort kann er das entsprechende Fahrzeug auswählen. Daraufhin werden die Fahrzeugdetails angezeigt. Dort ist auch ein Knopf 'Bearbeiten' zu finden. Wenn der Mitarbeiter auf diesen klickt, kann er die Daten des Fahrzeuges bearbeiten und auch einen Standort auswählen. Sobald ein Standort ausgewählt wurde, klickt der Mitarbeiter auf 'Speichern'. Der Datensatz wird dann gespeichert und dem Mitarbeiter wird eine Benachrichtigung angezeigt, dass die Änderung erfolgreich war.

```
1  BEGINN ZUORDNEN
2      WENN Mitarbeiter noch nicht angemeldet
3          Mitarbeiter gibt Nutzernamen und Passwort ein
4          Mitarbeiter klickt auf anmelden
5      WENN Nutzernamen nicht existiert ODER Passwort falsch
6          Zeige Fehlermeldung "Fehlerhafte Anmeldung"
7      SONST
8          Zeige Startseite
9      ENDE WENN
```

```
10      ENDE WENN
11      Mitarbeiterklickt auf "Fahrzeuge"
12      Zeige Fahrzeug-GUI
13      Mitarbeiter wählt Fahrzeug aus
14      zeige Fahrzeugdetails
15      Mitarbeiter klickt auf "Bearbeiten"
16      Fahrzeugdetails können bearbeitet werden
17      Mitarbeiter wählt Standort aus
18      Mitarbeiter klickt auf "Speichern"
19      Datensatz speichern
20      Zeige Bestätigung für das Speichern des Datensatzes
21 ENDE ZUORDNEN
```

Listing 9.4: Methode um ein Fahrzeug einem Standort zuzuordnen

9.2 Standort anlegen

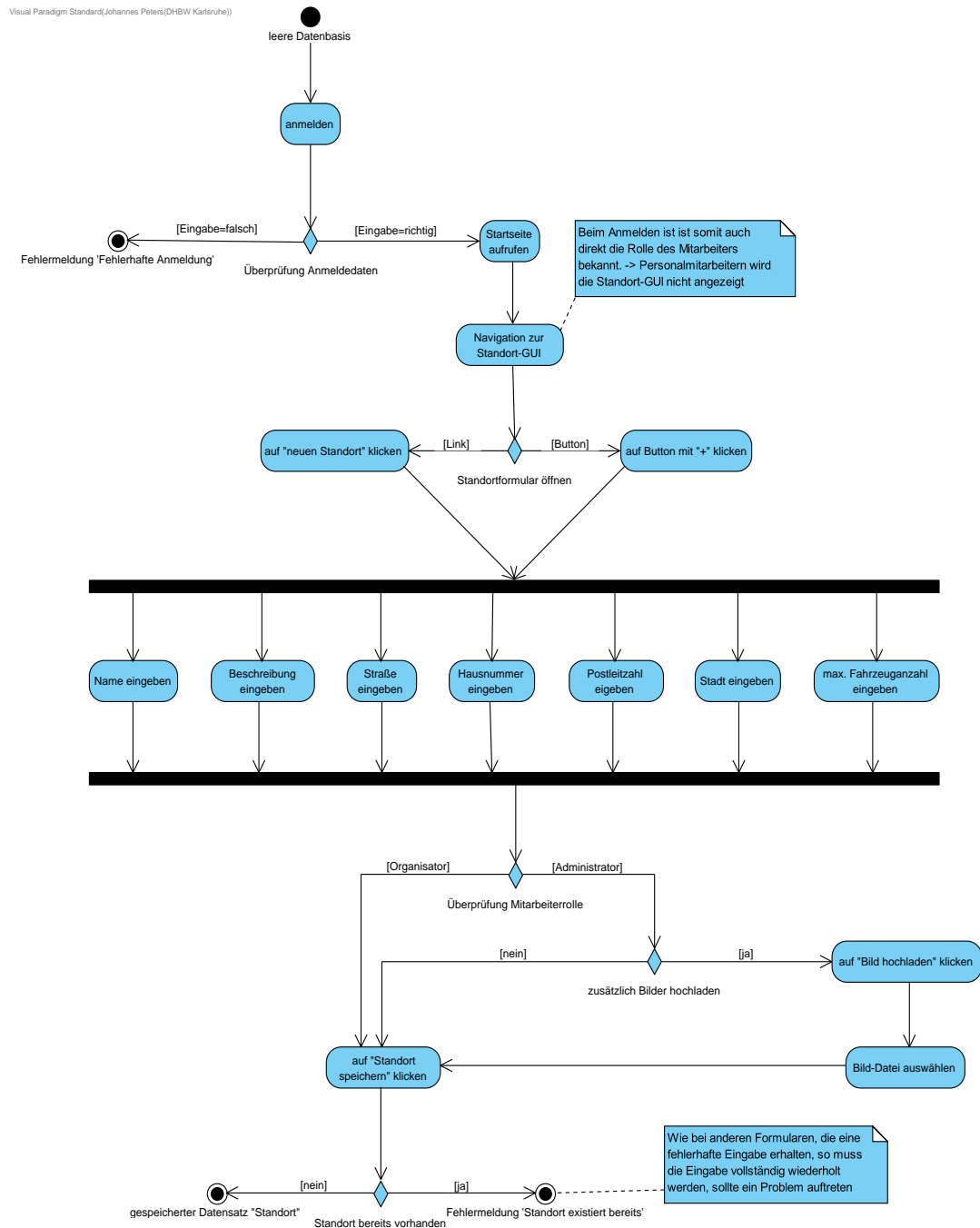


Abbildung 9.1: Aktivitätsdiagramm: Standort anlegen

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.

9.3 Fahrzeug anlegen

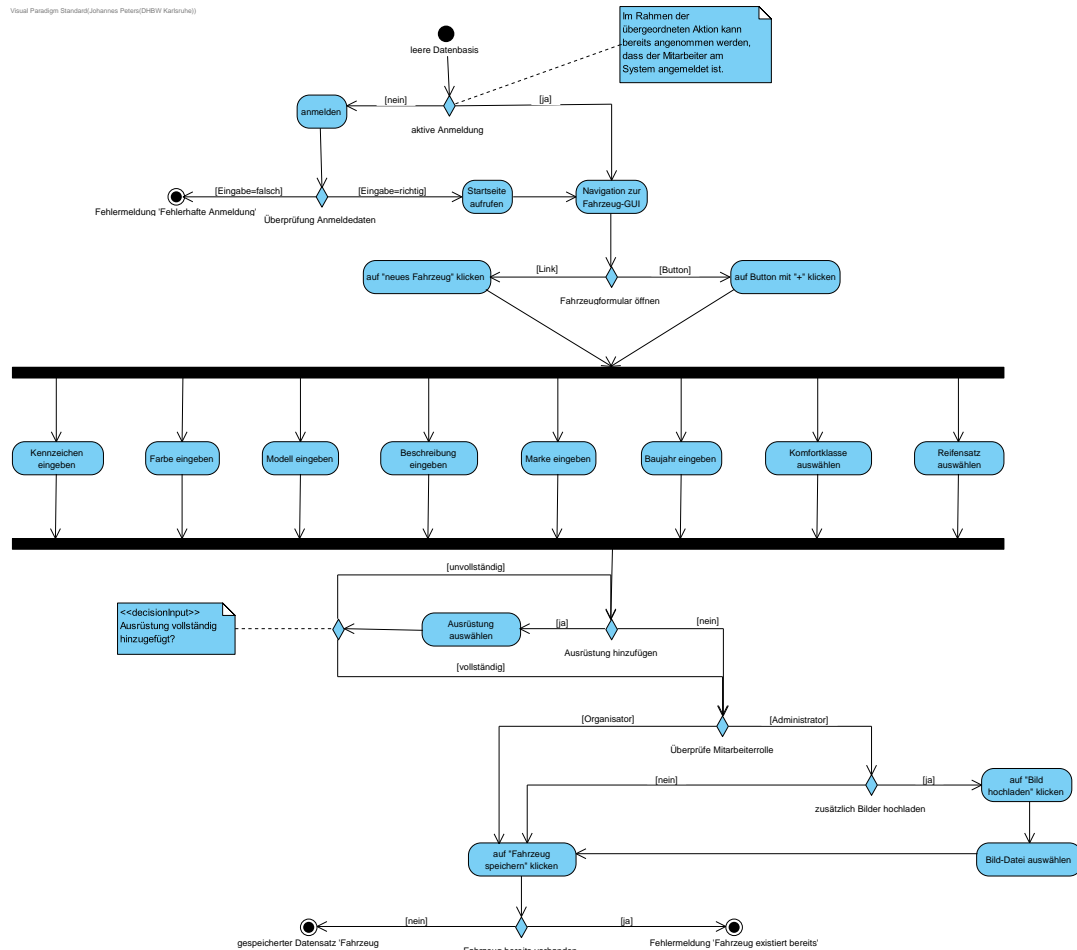


Abbildung 9.2: Aktivitätsdiagramm: Fahrzeug anlegen

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 'eingeben', 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 Hundetransportbox 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.

9.4 Fahrzeug einem Standort zuordnen

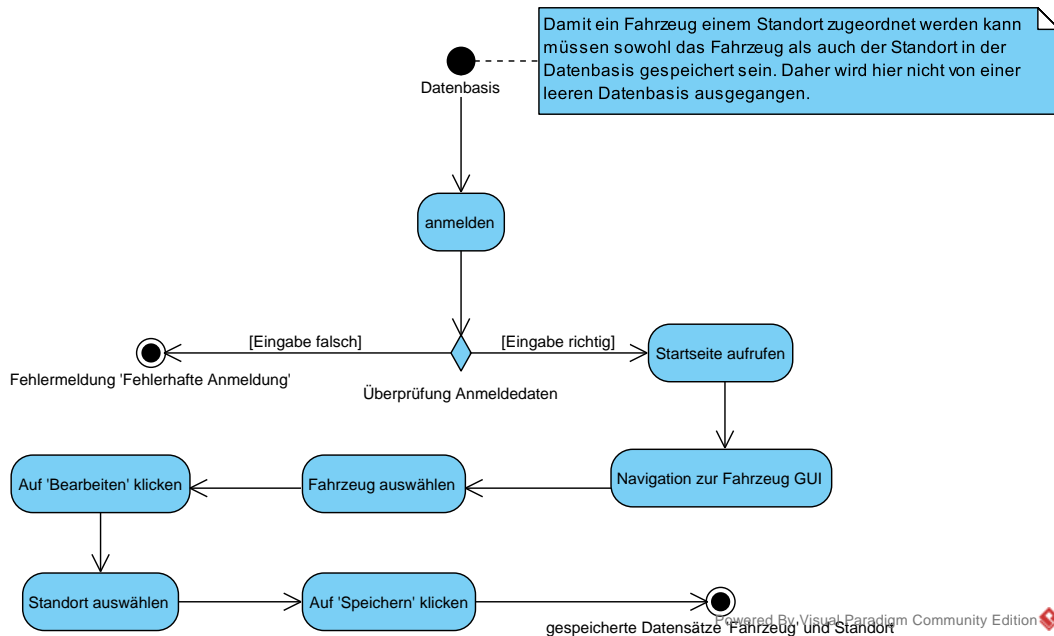


Abbildung 9.3: Aktivitätsdiagramm: Fahrzeug einem Standort zuordnen

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 Entwurfsklassendiagramm

Im folgenden Kapitel wird das Entwurfsklassendiagramm (EKD) genauer erklärt. Das EKD erweitert das Analyseklassendiagramm um Pakete, Methoden, Attribute und weitere Elemente, welche für die Umsetzung notwendig sind.

10.1 Diagramm

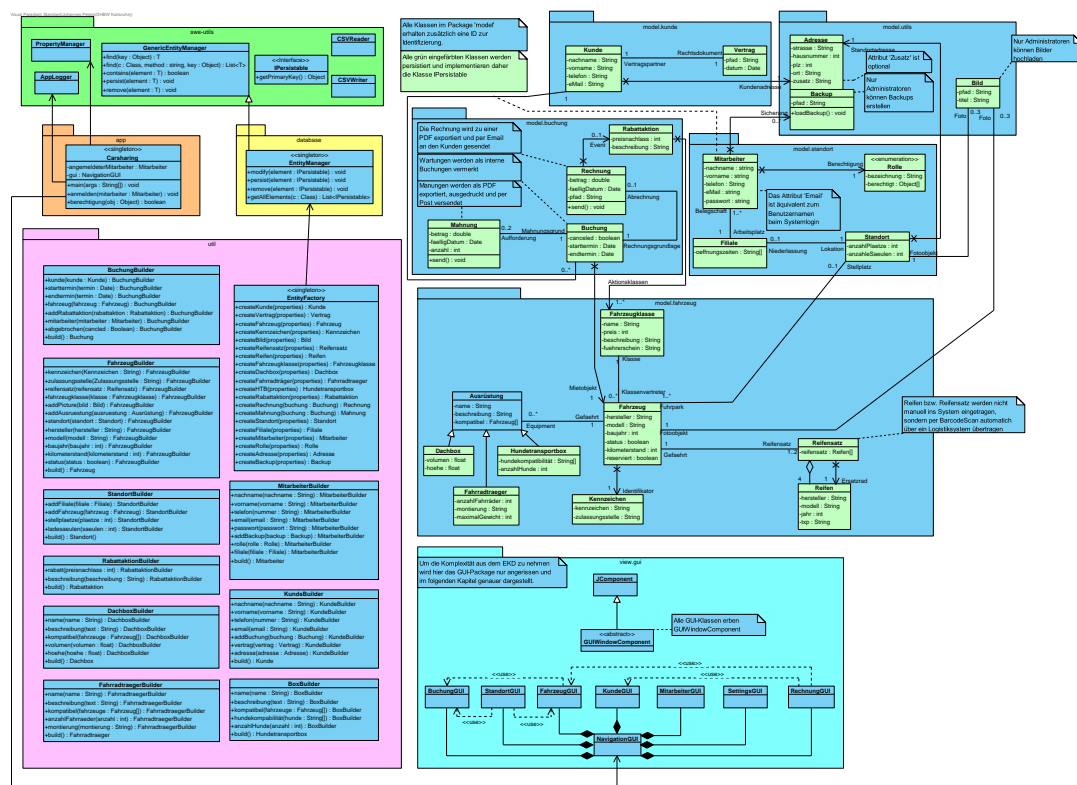


Abbildung 10.1: Entwurfsklassendiagramm: Allgemein

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, doch wird das Senden der Mahnung manuell durch Mitarbeiter übernommen, da diese per Post an den Kunden gesendet werden soll.

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' 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 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. Auf die extra Verwendung eines Dateinamen wurde verzichtet, da dieser aus dem Pfad herausgelesen werden kann. 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 bzw. GUIs ein Mitarbeiter zugreifen darf. Der Standort hat als Attribute die Anzahl der Parkplätze und Ladesäulen, welche zur Verfügung stehen und die Filiale besitzt die Öffnungszeiten als Attribut. Die Präsenz des Mitarbeiters im System dient nur der Rollenverteilung im System, d.h. es werden keine HR-Aufgaben darüber abgewickelt. Mitarbeiter zeichnen sich neben Vor- und Nachnamen durch eine Betriebs-E-Mail-Adresse und Betriebstelefonnummer aus. Außerdem wird noch das Passwort des Mitarbeiters gespeichert, damit dieser sich bei der Anwendung über die E-Mail als Benutzernamen anmelden kann.

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 Klassen 'Backup' und 'Bild'.

10.2.6 app

Hier befindet sich die zentrale Klasse 'Carsharing', welche die Main-Methode für das Starten der Anwendung beinhaltet. Diese Klasse wird als Singleton implementiert.

10.2.7 util

Im Package 'util' befindet sich die 'EntityFactory', welche für das Erzeugen von Instanzen der Klassen aus den Model-Paketen, welche in der Datenbasis gespeichert werden, verantwortlich ist. Für die Buchungen, Fahrzeuge, Standorte, Mitarbeiter, Kunden, Rabattaktionen, Dachboxen, Fahrradträger und Hundetransportboxen werden extra Builder implementiert, da es sich hierbei um besonders komplexe Klassen handelt.

10.2.8 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'. Das Paket wurde vom Auftraggeber zur einfacheren Implementierung mitgeliefert.

10.2.9 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.

10.2.10 view.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 85 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 Programmmentwurf 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. Basierend auf einer Mitarbeiterumfrage haben wir uns für ein rötlich-rosa Farbschema entschieden, sodass die Farben (in Hexwerten) **#DE639A**, **#E388B1**, **#D7A6B3**, **#F1E2E2**, **#707070** ihre Anwendung fanden.

Der Nutzer soll die Möglichkeit haben das Design der Benutzeroberfläche auf ein 'Dark Theme' umzustellen. Durch die Auswahl des 'Dark Themes' soll die Benutzeroberfläche der Anwendung dunkler dargestellt werden. Dazu haben wir uns zusätzlich für folgendes Farbschema entschieden (siehe Abbildung 11.1).





			
#2E0249	#570A57	#A91079	#F806CC
rgb(46, 2, 73)	rgb(87, 10, 87)	rgb(169, 16, 121)	rgb(248, 6, 204)

Abbildung 11.1: Dark Theme Farbschema

11.1 Diagramm

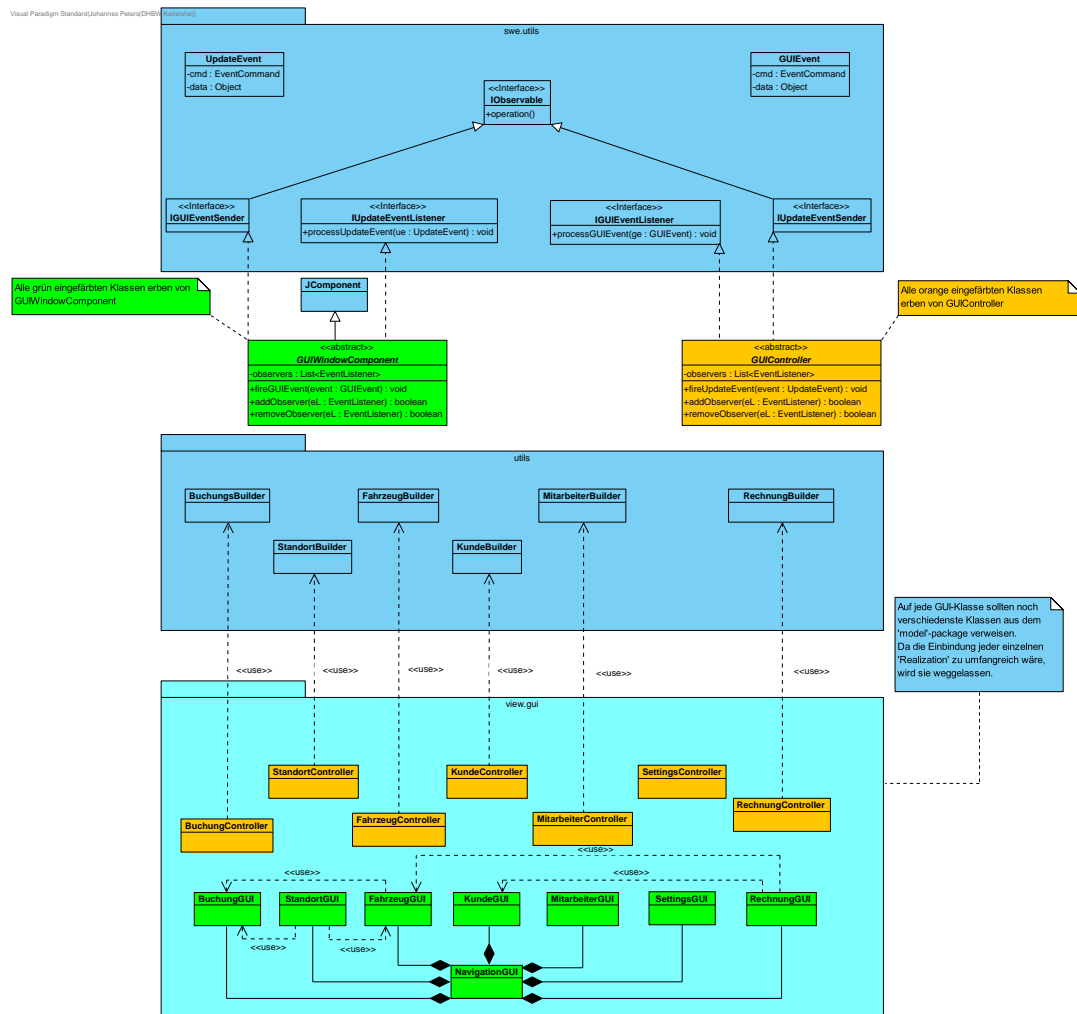


Abbildung 11.2: Entwurfsklassendiagramm: GUI

11.2 Mock-Up

Der Aufbau der Benutzeroberfläche ist an die Vorgehensweise bei mobilen Android Anwendungen angelehnt und basiert auf dem Konzept von 'Activity' und 'Fragments'. Die NavigationGUI dient dabei als 'Activity' und stellt eine Navigationsleiste bereit, die es den Anwendern ermöglicht zwischen verschiedenen Verwaltungsbereichen hin und her zu wechseln. Diese Navigationsleiste befindet sich vertikal angeordnet auf der linken Seite des Bildschirms, während der gesamte obere Bereich den Schriftzug des Unternehmens und das Firmenlogo zeigt. Die Hauptfläche des Bildschirms erscheint als das 'Fragment', welches je nach Verwaltungsbereich unterschiedlich gestaltet ist. Um Rücksicht auf die technisch weniger versierten Mitarbeiter zu nehmen, soll die Aufteilung der Fragment-Fläche bei den unterschiedlichen Bereichen so ähnlich wie möglich sein. Grundsätzlich besteht daher jede Fragment-Seite aus zwei Feldern. Das linke Feld zeigt eine Liste, in der sich meist alle Einträge des betroffenen Verwaltungsobjekts befinden, das rechte Feld dient der Detailansicht bei Auswahl eines Datensatzes.

11.2.1 Fahrzeuge verwalten

Sobald ein Anwender zur Fahrzeugverwaltung navigiert, erhält er eine Übersicht an allen vorhanden Fahrzeugen aus der Datenbasis. Diese Datenabfrage wird automatisch beim Aufrufen der GUI ausgeführt. Wie auch bei vielen anderen Listenanwendungen gibt es die Möglichkeit nach bestimmten Dateneinträgen zu Suchen oder zu Filtern. Das durch den Klick auf den Button oder das Enter in der Texteingabe wird ein GUIEvent erzeugt, welches eine modifizierte Datenabfrage an die Datenbasis absetzt. Mittels eines UpdateEvents wird die Liste neu geladen und die geänderte Datenauswahl wird in der Liste angezeigt. Sofern ein Fahrzeugeintrag ausgewählt wird, soll auf der rechten Bildschirmseite die Detailansicht des Fahrzeuges erscheinen. Auch hier wird mittels eines GUIEvents durch das Klicken auf den entsprechenden Listeneintrag auf eine Mauseingabe gewartet, welches die Detailinformationen auf den Bildschirm schreibt.



Abbildung 11.3: Mock-Up: Liste an Fahrzeugen



Abbildung 11.4: Mock-Up: Detailansicht eines Fahrzeugs

Fahrzeug löschen

Exemplarisch für Fahrzeuge, doch übertragbar auch alle anderen Datenobjekte, ist die Manipulationen der Datensätze möglich. Zu den Manipulationsmöglichkeiten gehören 'Anlegen', 'Bearbeiten' und 'Löschen'. Die Interaktion zum Löschen des ausgewählten Fahrzeug-Datensatzes erfolgt über ein einfaches Pop-Up-Fenster. Der Anwender muss diesen Dialog bestätigen, wenn er den Datensatz löschen möchte. Da die bestehende Datenbasis durch das Löschen bearbeitet wird, erzeugt diese Aktion ein UpdateEvent und entfernt den betroffenen Datensatz aus der Datenbasis. Nach dem Löschen wird der Anwender auf die Fahrzeug-Startseite zurückverwiesen, jedoch wird in der Liste der gelöschte Eintrag durch das ausgelöste UpdateEvent entfernt.



Abbildung 11.5: Mock-Up: Löschen eines Fahrzeugs

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 z.B. Preis werden aus der Klasse 'Fahrzeugklasse' hinzugezogen, Kilometerstand wird 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, ist die Bearbeitung mittels Checkbox-Liste vorgesehen. Sofern alle Bearbeitungsschritte durchgeführt wurden. Beim Speichern der Änderungen wird ein Dateneintrag in der Basis abgeändert und diese Veränderung soll über den Controller als UpdateEvent wieder in der Fahrzeug-Detailseite angezeigt werden.

Carsharing Dödelhausen

- [🏠 Startseite](#)
- [📍 Standorte](#)
- [🚗 Fahrzeuge](#)
- [👤 Kunden](#)
- [📅 Buchungen](#)
- [🔖 Aktionen](#)
- [👥 Mitarbeiter](#)
- [⚙️ Back-Ups](#)

▼
+

	Marke	Modell	Preis/h	Buchung	
<input type="checkbox"/>	Porsche	Taycan	10,50€/h	Buchung	●
<input type="checkbox"/>	Ford	Focus	2,50€/h	Buchung	●
<input type="checkbox"/>	Ford	Mustang GT	8,50€/h	Buchung	●

[+ Fahrzeug hinzufügen](#)

Status: verfügbar

Hersteller: Porsche

Klasse: Sportwagen

Baujahr: 2020

Reifen: ▼

[Buchungen](#)

Modell: Taycan

Preis: 10.50€/h

Kilometerstand: 15.000 km

Ausrüstung: ▼

[speichern](#)

[abbrechen](#)

[anlegen](#)

[Einstellungen](#)

Abbildung 11.6: Mock-Up: Bearbeiten eines Fahrzeugs

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. Äquivalent zum Vorgehen beim Bearbeiten eines Fahrzeugs wird auch hier erst mit Aktivierung des 'Speichen'-Buttons eine Aktion ausgelöst, die über den Controller den FahrzeugBuilder bemüht und einen neuen Eintrag in der Datenbasis erzeugt. Sobald der Eintrag physisch abgeschlossen wurde, wird der neue Eintrag im Programm instanziiert und über ein UpdateEvent in der GUI angezeigt.



Abbildung 11.7: Mock-Up: Anlegen eines Fahrzeugs

11.2.2 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 Funktionsweise und Interaktion mit anderen Programmbestandteilen ist dabei absolut identisch zu den Verwaltungsaufgaben bei 'Fahrzeugen'.

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) oder als einfachere Möglichkeit einfach ein Bild), auf der alle Standorte mit Markierungen eingetragen sind.

Sofern man eine Markierung auf der Karte oder einen Listeneintrag auswählt, wird man zu einer Standort-Detail-Seite geleitet, indem ein GUIEvent ausgelöst wird. 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, auch hier sind Fotos 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. Da eine aktive Datenabfrage erfolgen muss um zu ermitteln welche Fahrzeuge aktuell an dem ausgewählten Standort vorzufinden sind, wird dieser GUI-Bestandteil per UpdateEvent geladen und angezeigt.



Abbildung 11.8: Mock-Up: Liste an Standorten



Abbildung 11.9: Mock-Up: Detailansicht von Standorten

11.2.3 Buchung verwalten

Auf Anfrage des Auftraggebers erfolgte auch eine nähere Ausarbeitung der Buchungsverwaltung. Die Buchungsverwaltungs-Oberfläche kann auf zwei Arten adressiert werden. Einerseits ist die Navigation über die Navigationsleiste möglich. In diesem Fall gelangt der Anwender zu einer einfachen Übersicht in Form der bereits bekannten Liste. Sortiert nach BuchungsID (welche laufend vergeben werden) kann nach bereits vorhandenen Buchungen gefiltert und gesucht werden. Sollte man eine Buchung anlegen wollen gibt es den entsprechenden 'Anlegen'-Button, welcher ein GUIEvent zur Erstellung aufruft. Die andere Möglichkeit ist die Verwendung des 'Buchen'-Links bei Fahrzeugen. In diesem Fall wird der Anwender automatisch auf die BuchungsGUI weitergeleitet und mittels eines UpdateEvents werden die Informationen des ausgewählten Wagens automatisch auf der rechten Seite des Bildschirms angezeigt. Die linke Seite des Bildschirms ist zu Anfang nur ein leerer Bereich, doch existiert ein Eingabefeld. Sofern eine existierende KundenID in dieses Feld eingetragen wird, werden die Daten des Kunden durch ein GUIEvent aufgerufen und angezeigt. Bereits mit Fahrzeug und Kundendaten ausgestattet, wird nur noch der Buchungstermin als Information benötigt. Ein Date- & Timepicker stellt die

Funktion bereit den gewünschten Zeitraum auszuwählen und beim Schließen des Pickers lädt das damit verbundene GUIEvent die Zeitraumdaten und trägt sie in der GUI ein. Die Erstellung der Buchung wird mit der Bestätigung des 'buchen'-Buttons abgeschlossen, indem der Controller die Anforderung an den BuchungsBuilder weiterleitet, welcher ein Objekt der Klasse 'Buchung' erzeugt und speichert.



Abbildung 11.10: Mock-Up: Listenübersicht aller Buchungen

The image shows a web application mock-up for 'Carsharing Dödelhausen'. It features a pink sidebar with navigation links: Startseite, Standorte, Fahrzeuge, Kunden, Buchungen, Aktionen, Mitarbeiter, Back-Ups, and Einstellungen. The main content area is divided into three sections. The top section has a 'KundenID' input field. The middle section displays customer details for 'Dooley Tom' with contact information and a booking date. The right section shows car details for a 'Porsche Taycan' with its status, price, and specifications. At the bottom right are 'buchen' and 'abbrechen' buttons.

Carsharing Dödelhausen

KundenID

Kunde

Name: Dooley Tom

Telefon: +49 800 1721212

E-Mail: t.dooley@hoonk.com

Termin

Termin: [Kalender](#)

Start: 13.05.2022 13:30

Ende: 13.05.2022 17:30

Status: verfügbar

Hersteller: Porsche **Modell:** Taycan

Klasse: Sportwagen **Preis:** 10.50€/h

Baujahr: 2020 **Kilometerstand:** 15.000 km

[Reifen](#) [Ausrüstung](#)

buchen abbrechen

Abbildung 11.11: Mock-Up: Detailansicht beim Erstellen einer Buchung

12 Besonderheiten

Im folgenden Kapitel werden dem Leser die Besonderheiten der entworfenen Software näher gebracht.

12.1 Detailtiefe

Damit der Kunde bei der Buchung eines Fahrzeuges so viele Informationen wie nur möglich über das Mietobjekt erhält, wurde vor allem bei der Modellierung des Fahrzeuges auf eine große Detailtiefe geachtet. Dies ermöglicht nicht nur dem Kunden eine bessere Entscheidung über das passende Fahrzeug zu treffen, sondern vereinfacht auch die Instandhaltung und die Bestandsaufnahme für das Carsharing selbst, da alle wichtigen Informationen zu den Fahrzeugen abgebildet werden können. Dazu zählen unter anderem die Reifen, welche zum einen als Reifensatz aber auch als einzelne Reifen modelliert wurden. Außerdem wurde jegliche Ausrüstung der Fahrzeuge durch dedizierte Klassen detailliert beschrieben. Auch die Wartungen können problemlos dargestellt werden, indem eine Buchung für die Werkstatt angelegt wird.

Damit die Anwendung im Einsatz vollfunktional ist, wurden auch Rabattaktionen mit eingebunden, welche es dem Carsharing erlauben die Preise für einzelne Fahrzeugklassen temporär zu senken. Dies wird automatisch mit in die Rechnung für den Kunden einbezogen. Diese Rechnungen werden auch automatisch durch unser System versendet und falls diese nicht bezahlt werden sollten, stehen wir bereits in engem Kontakt mit der schweizerischen Nationalbank (SNB), um ein reibungsloses Einfrieren der Bankkontos zu ermöglichen.

12.2 Benutzeroberfläche

Auf die detailreiche der modellierten Benutzeroberfläche ist besondere Aufmerksamkeit zu werfen, da sich hier sogar an einer Mitarbeiterumfrage bezüglich der Farbauswahl orientiert wurde. Für die Benutzeroberfläche wurden insgesamt neun unterschiedliche Ansichten entworfen, welche als Orientierung für die Implementierung dienen. Hierbei wurde jeder Schlüsselaspekt der Anwendung und noch mehr ersichtlich, indem neben dem Anlegen von Buchungen sowie die Übersichten der Fahrzeuge und Buchungen auch die Detailansichten für sowohl Fahrzeuge als auch Standorte bereits modelliert wurden. Zudem ist für die Fahrzeuge jeweils eine Ansicht zum Anlegen, Löschen und Bearbeiten erstellt wurden. Dabei wurden auch Details wie das Logo, oder eine Karte für die Standorte berücksichtigt. Ebenfalls berücksichtigt wurde das hohe Durchschnittsalter beziehungsweise der hohe Anteil der digitalen Analphabeten unter den Mitarbeitern des Kunden. Um diesen prozentual gesehen nicht unerheblichen Teil der Mitarbeiter einen einfachen Umgang mit der Anwendung zu gewährleisten, wurde besonders auf Benutzerfreundlichkeit geachtet. Dazu wurden extra eigens hergestellte Icons zusätzlich zum Text in der Navigation hinzugefügt. Bei der Navigation wurde zudem komplett auf die Verwendung von Transaktionen (vgl. SAP Systeme) verzichtet, sodass alles über die Navigationsleiste am linken Fensterrand erreichbar ist. Außerdem wurde darauf geachtet, dass alle Funktionen mit nicht mehr als drei Klicks zu erreichen sind. Die wohl wichtigste Designentscheidung in Bezug auf das hohe Durchschnittsalter ist aber wohl die dreifache Abfrage, ob sich der Nutzer sicher ist, ob er das Element löschen möchte. Denn im höheren Alter kann es durchaus schon mal vorkommen, dass man mal mit der Maus ausrutscht und ausversehen etwas löscht. Des Weiteren wird in den Formularen überall wo es möglich ist ein Auswahlfeld anstelle einer Texteingabe verwendet, damit die Mitarbeiter nicht die kleinen Tastaturtasten suchen müssen. Dadurch werden zahlreiche Tippfehler präventiv vermieden und auch der Zeitaufwand durch das Zwei-Finger-Suchsystem minimiert. Eine weitere Designentscheidung für den Komfort bei der Bedienung ist, dass alles in einem Fenster dargestellt wird und keine weiteren sinnlosen Fenster geöffnet werden. So werden nur für zusätzliche Abfragen Popups verwendet.

Die erwähnten Ansichten wurden alle als Teil eines interaktiven Prototyps entwickelt. Der Prototyp wurde mithilfe von Adobe XD erstellt und ist mit angehängt. Um den Prototyp öffnen zu können ist Adobe XD vorausgesetzt.

Da die Mitarbeiter des Carsharings verschiedene Vorlieben bezüglich der Optik haben, wurden verschiedene Einstellungsmöglichkeiten für die Benutzeroberfläche eingeplant. So sind die Schriftart, Textgröße und Fenstergröße anpassbar auf die Ansprüche des entsprechenden Mitarbeiters. Zusätzlich steht die Möglichkeit zur Verfügung, zwischen 'Light Theme' und 'Dark Theme' zu wechseln. Dadurch soll es möglich sein, dass jeder Mitarbeiter die Anwendung individuell nach seinen Bedürfnissen einrichten kann.

12.3 Pseudocode

Sowohl für das Anlegen einer Buchung, als auch das Anlegen eines Standortes mit Fahrzeugen wurde, neben der Dokumentation durch Sequenzdiagramme beziehungsweise Aktivitätsdiagramme, bereits detaillierter Pseudocode geschrieben. Dieser bildet den Programmablauf genaustens ab und erleichtert somit die Implementierung erheblich. Dabei wurde auch auf Modularisierung geachtet, indem einzelne Programmabschnitte gezielt in eigene Methoden ausgelagert wurden. Dadurch wird der Ablauf noch einfacher verständlich und auf einzelne Teilschritte heruntergebrochen.

12.4 Übersichtlichkeit

Um die Komplexität der Modellierung zu reduzieren, wurde sich dazu entschieden bestimmte Funktionen zu vereinfachen beziehungsweise zusammenzufassen. So wird zum Beispiel jeglicher Import und Export von Daten mithilfe der Klasse 'Backup' dargestellt. Ein weiteres Beispiel ist die Klasse 'Buchung', welche sowohl für die Termine der Kunden als auch für die Wartungstermine bei der Werkstatt verwendet wird. Dadurch wird auf unnötige Komplexität beim Entwurf der Software verzichtet und eine einfachere Bedienung durch weniger Menüs gewährleistet.

12.5 Kundenorientierung

Bei der Planung der Software wurde sich nicht nur an den Bedürfnissen der Mitarbeiter des Carsharings orientiert, sondern auch an den Bedürfnissen der Kunden. Dies merkt man zum einen an der bereits erwähnten Detailtiefe, welche dem Kunden so viele Informationen wie möglich zu jedem Fahrzeug bereitstellt, und zum anderen an der Reservierungsfunktion. Wenn ein Kunde ein Fahrzeug buchen möchte, wird ein Zeitstempel im System hinterlegt, welcher das Fahrzeug für 15 Minuten reserviert, sodass kein anderer Kunde das Fahrzeug in diesem Zeitraum buchen kann. Dadurch wird zusätzlich noch verhindert, dass zwei Mitarbeiter das gleiche Fahrzeug gleichzeitig buchen.