

L01: Pflichtenheft

Gruppe 2 Team 1:

Mitglied:

Matrikelnummer:

Mariella Heidi Kuimo Tchouanmoe

5163016

Ghulam Reza Moradi

5139697

Alireza Hosseini

5133583

Jame's Junior Tchana II

5165841

Jakob Spohler

5094026

Inhaltsverzeichnis

1 Einführung	3
1.1 Umfeld – Kundenbeziehung, vertragliche und inhaltliche Zuordnung dieses Dokumentes	3
1.2 Ausgangssituation und Ziele	3
1.2.1 Ausgangssituation	3
1.2.2 Ziele	3
1.3 Umfang und Art des vorgeschlagenen Systems	3
1.4 Definitionen, Akronyme und Abkürzungen	3
2 Bezug zu Dokumenten	4
2.1 Anwendbare Dokumente	4
2.2 Referenzierte Dokumente	4
3 Voraussetzungen (Reza)	4
3.1 Hardware-Umgebung	4
3.2 Software-Umgebung	4
3.3 Entwicklungshilfsmittel	5
3.4 Verwendete Fremdprodukte	5
4 Randbedingungen (Alireza)	5
4.1 Technische Randbedingungen	5
4.2 Terminliche Randbedingungen	5
4.3 Implementierungs-/Entwicklungsvorschriften	5
4.4 Rechtliche Randbedingungen	5
5 Funktionsumfang	7
5.1 Anwendungsfälle	7
5.1.1 Use Case Diagramm 1 : Authentifikation (Alireza)	7
5.1.2 Use Case Diagramm 2: Ausleihstation verwalten (Jakob)	9
5.1.3 Use Case Diagramm 3: Reservierung verwalten und Abrechnung erstellen (Reza)	11
5.1.4 Use Case Diagramm 4: Karte benutzen (Kuimo Mariella)	13
5.1.4.1 Karte benutzen	14
5.1.4.2 Karte sperren	14
5.1.5 Use Case Diagramm 5: Mitgliedschaft verwalten (Jame's)	16
5.1.6 Use Case Diagramm 6: Fahrzeugdaten ändern (Alireza)	18
5.2 Funktionale Anforderungen	20
5.3 Schnittstellen Anforderungen (Alireza)	24
5.4 Bedienbarkeits/Operationalitäts-Anforderungen (Jakob)	24
5.5 Qualitätsanforderungen (Reza)	25
5.5.1 Zuverlässigkeitsanforderungen	25
5.5.2 Leistungs-/Performanceanforderungen	25
5.5.3 Sicherheitsanforderungen	25
5.5.4 Kompatibilitäts-/Portabilitätsanforderungen	25
5.6 Konfigurationen, Ausbaustufen, Varianten (Jame's)	25
5.7 Grenzen und Einschränkungen – Was das System nicht kann (Mariella Kuimo)	26
6 Lieferumfang – Liste der zu liefernden HW-/SW-Einheiten (Mariella Kuimo)	26

Das Produkt ist eine kompakte Datei, die mit zwei Server und einer zugehörigen Datenbank geliefert wird.	26
7 Funktionsprüfung/Abnahmekriterien(Mariella Kuimo)	26
7.1 Funktionsprüfung	26
7.2 Abnahmekriterien	27
7.3 Testhilfsmittel	27

1 Einführung

1.1 Umfeld – Kundenbeziehung, vertragliche und inhaltliche Zuordnung dieses Dokumentes

Der Auftraggeber Move beauftragt den Auftragnehmer Softwaregroup 2 ein Softwaresystem zu entwickeln. Dieses Dokument beschreibt die zu erbringende Leistung im Detail und deren Umfang.

1.2 Ausgangssituation und Ziele

1.2.1 Ausgangssituation

Der Auftraggeber möchte in das Carsharing-Geschäft einsteigen, die benötigten Autos sind bereits vorhanden.

1.2.2 Ziele

Der Auftragnehmer soll ein webbasiertes Softwaresystem entwickeln, das eine automatische Abwicklung der Reservierungen, Verwaltung der Fahrzeuge, Mitglieder und Abrechnungen ermöglicht.

1.3 Umfang und Art des vorgeschlagenen Systems

Das vorgeschlagene System ist webbasiert und soll die Daten persistent in einer Datenbank speichern. Es umfasst mindestens die Aufgaben einer automatischen Abwicklung der Reservierung, Verwaltung der Fahrzeuge, Mitglieder und Abrechnungen.

1.4 Definitionen, Akronyme und Abkürzungen

FA = funktionale Anforderung

NFA = nicht funktionale Anforderung

UC = Use Case

SA = Schnittstellenanforderung

BOA = Bedienbarkeits/Operationalitäts-Anforderungen

QA = Qualitätsanforderung

V-xx = Version xx

TI4 = Technische Informatik Viertes Semester

2 Bezug zu Dokumenten

2.1 Anwendbare Dokumente

Herkunft	Ref-der Anwendbaren Dokumente	Link
TI4 Softwaretechnik-2 Aulis Hochschule Bremen	Frequently Asked Questions-Laborprojekt Carsharing HSB@ Prof.Dr.-Ing Jasminka Matevska	Fragen
TI4 Softwaretechnik-2 Laborskript	Projektbeschreibung-HSB@ Prof.Dr.-Ing Jasminka Matevska	Lastenheft

2.2 Referenzierte Dokumente

Herkunft	Referenz und Autor	Link
Wissenschaftliche Dienste des Deutschen Bundestag	Rahmenbedingungen für Carsharing und Roller Unternehmen	Link zum wissenschaftlichen Dienst
Java Coding Richtlinie	Coding-Styles	Link zu den Coding-Styles
Gesetzgebungsakte und andere Rechtsinstrumente	DSGVO@ Rat der Europäische Union	Link zum DSGVO Dokument
TÜV SÜD	Änderung am Fahrzeug	Link zum TÜV SÜD Dokument
Betriebs-Berater Geldverkehr	Recht der Zahlungsdienste	Link zum Dokument Betriebsberater
Gesetze für Carsharing	Bundesministerium der Justiz	Gesetz zur Bevorrechtigung des Carsharing

3 Voraussetzungen (Reza)

3.1 Hardware-Umgebung

- Internetfähiger Rechner mit Bildschirm.
- Ausreichend Rechen- und Festplattenkapazität.
- Da jedes Mitglied ein RFID-Karte bekommen soll, um das reservierte Fahrzeug bei Abholung öffnen und bei Rückgabe schließen kann, werden RFID-Karte und die entsprechende Hardware wie RFID-Reader/Writer benötigt.

3.2 Software-Umgebung

- Webbrowser

3.3 Entwicklungshilfsmittel

- Visual Paradigm 16.2

- Jira Software von Atlassian
- Apache NetBeans 12.3
- IntelliJ IDEA 2020
- PostgreSQL-13
- Spring Boot 2.7

3.4 Verwendete Fremdprodukte

Es wird eine gängige Webumgebung wie Web-Server, SMTP-Server, Domain und SSL-Zertifikat benötigt, um auf das Softwaresystem zuzugreifen und bereitzustellen. Der Kunde kann entscheiden bei welchem Hosting-Anbieter die o.g Materialien bestellt werden.

4 Randbedingungen (Alireza)

4.1 Technische Randbedingungen

Es sind keine technische Randbedingungen bekannt.

4.2 Terminliche Randbedingungen

12.05.2022: Es soll ein vollständiges Pflichtenheft dem Kunden vorgelegt werden

20.05.2022: Der Projektplan soll dem Kunden vorgelegt werden.

30.05.2022: Ein Systementwurf soll dem Kunden vorgelegt werden.

15.06.2022: Die erste Version der Software soll implementiert werden.

30.06.2022: Der Projektvorschriften soll vor einem Plenum präsentiert werden.

4.3 Implementierungs-/Entwicklungsvorschriften

Um eine gute Qualität des Produkts zu gewährleisten, Erweiterbarkeit des Produktes zu vereinfachen und kommende Fehler effizienter zu beseitigen, sollen einheitliche Coding-Styles im gesamten Projekt verwendet werden. Coding-Styles beschreibt Namenskonventionen und Formatierungen im Quellcode.

Unser Entwicklungs-Team orientiert sich an diesen [Java Coding Richtlinien](#).

4.4 Rechtliche Randbedingungen

→ Datenschutz

Da die Speicherung kundenspezifischer Daten in dem Endprodukt vorkommt, ist eine Prüfung der zu speichernden Daten auf rechtliche Korrektheit sehr empfohlen. Hierzu verwendet unser Team das referenzierte Dokument [DSGVO](#). Unser Entwicklungs-Team achtet insbesondere auf Privacy by Design und Privacy by Default.

→ Umbau des Fahrzeuges

Der Umbau der Schließanlage der Fahrzeuge kann unter Umständen zum Entzug der Betriebserlaubnis für ein Fahrzeug führen. Denn ein Fehler in der Fahrzeugelektronik kann zu einem

Unfall führen. Daher wird empfohlen, nach Umbau die Fahrzeuge einer neuen Prüfung zu unterziehen. Hierzu verwenden wir das referenzierte Dokument [TÜV SÜD](#).

→ Service Level Agreements (SLA)

Für die bessere Service und Wartungsarbeit werden die folgenden Vereinbarungen getroffen. Falls nach der Installation des System ein Fehler auftaucht wird versucht diese Fehler innerhalb von 24 Stunden zu beheben. Dazu wird von unserem Mitarbeiter zuerst eine Fehleranalyse durchgeführt. Nachdem herausgefunden wurde was der Fehler ist, werden unsere Fachleute den Fehler beheben. Außerdem können Kunden sich per E-Mail melden, wenn sie einen Fehler entdecken. Dazu machen wir ein Ticket in unserem Ticketsystem und werden innerhalb von zwei Woche den Fehler beseitigen.

Für Systemausfälle wird dabei eine Systemverfügbarkeit von 95% garantiert. Dies wird bei uns durch unterschiedlich Methoden wie z.B. Nutzung von RAID-Systemen, Backups und Firewalls usw. erfolgt.

Wenn der Kunde Funktionalitäten der Software verlangt die nicht in diesem Dokument vorhanden sind so sind diese in einem separaten Vertragsschluss zu behandeln.

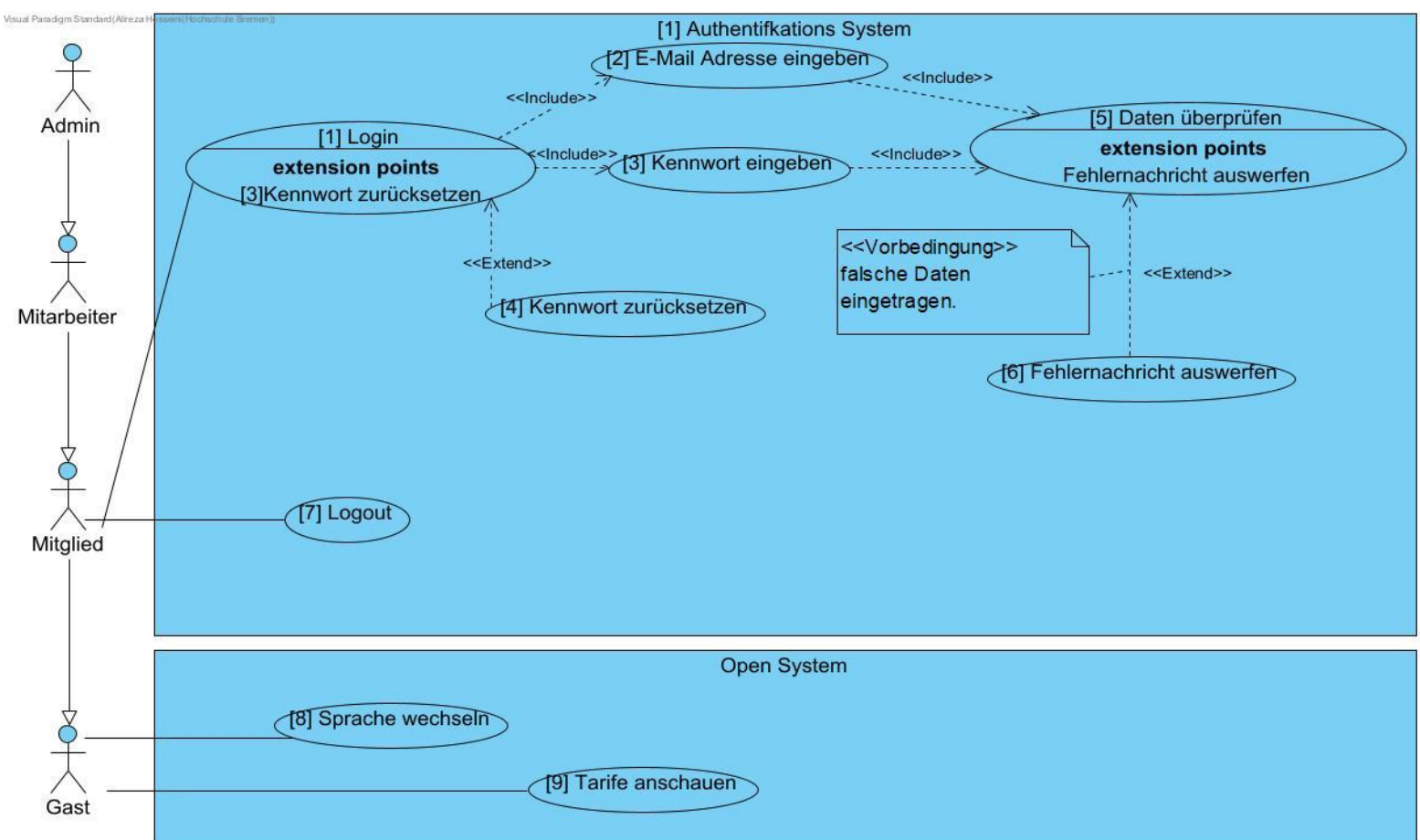
Die hier beschriebenen Wartungsvertragsbedingungen sind gültig für 5 Jahre nach Vertragsschluss.

5 Funktionsumfang

5.1 Anwendungsfälle

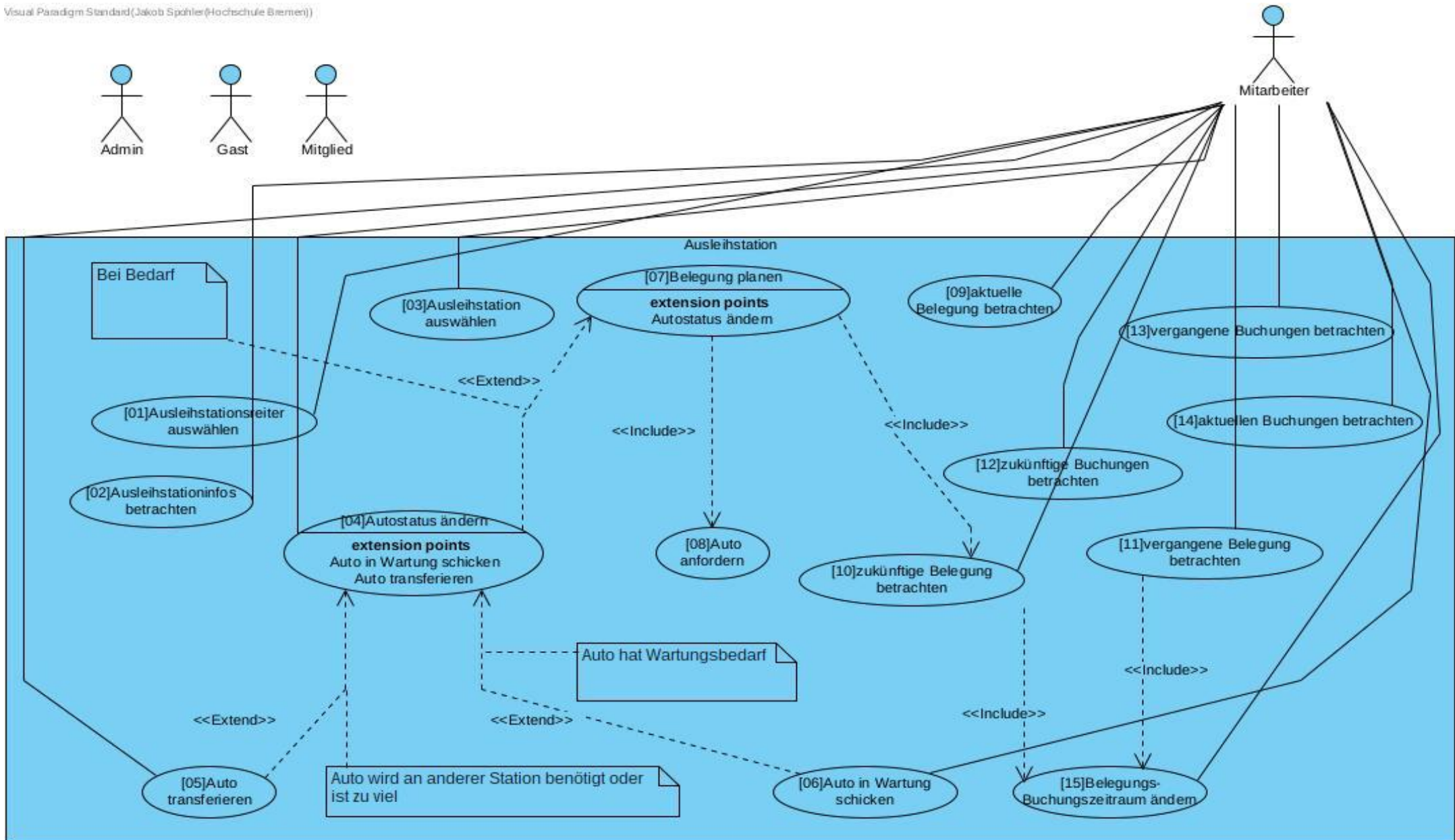
5.1.1 Use Case Diagramm 1 : Authentifikation (Alireza)

Das authentifikations System ist die erste Funktion des Programms auf die die Benutzer treffen. Alle wichtigen Aktionen können von Admin, Mitarbeiter und Mitglied nur dann durchgeführt werden, wenn sie sich einloggen. Der Unterschied zwischen dem Akteur „Gast“ und „Mitglied“ besteht darin, dass das Mitglied ein Konto im System hat und sich ein- bzw. ausloggen. Der Gast ist nur in der Lage, die Sprache zu wechseln, um sich Tariffe anzuschauen.



Anwendungsfall	UC-1
Name	Login
Initiierender Akteur	Mitglied
Weitere Akteure	Mitarbeiter, Admin
Kurzbeschreibung	Der User (Mitglied, Mitarbeiter oder Admin) kann sich ins System einloggen.
Vorbedingung	Es soll ein Konto angelegt sein.
Nachbedingung	Nach der gewünschten Aktion soll der User sich ausloggen können.
Ablauf	Der User gibt seine richtige E-Mail-Adresse und Kennwort ein. Dann ist er im System.
Alternativen	-
Ausnahmen	Das Kennwort kann zurückgesetzt werden, wenn der User es vergessen hat.
Benutzte Anwendungsfälle	UC2 , UC3 , UC4
Spezielle Anforderungen	<ul style="list-style-type: none"> -Wenn der User sein Kennwort eingibt, soll er es nicht sehen können. - wenn der User sein Kennwort 5 Mal falsch eingibt, soll sein Konto gesperrt werden. - automatische Abmeldung nach 10 Minuten, wenn der User keine Aktion durchführt.
Annahmen	-
Offene Themen	-
Referenzen	<p>DSGVO-Dokument</p> <p>Da es sich um einen sicherheitskritischen Bereich handelt, ist es gut zu wissen, wie soll mit diesen sensiblen Daten (Kennwort, ...) umgegangen werden soll.</p>
Datenanforderungen	Login-Daten soll im System geprüft werden.
Nichtfunktionale Anforderungen	-

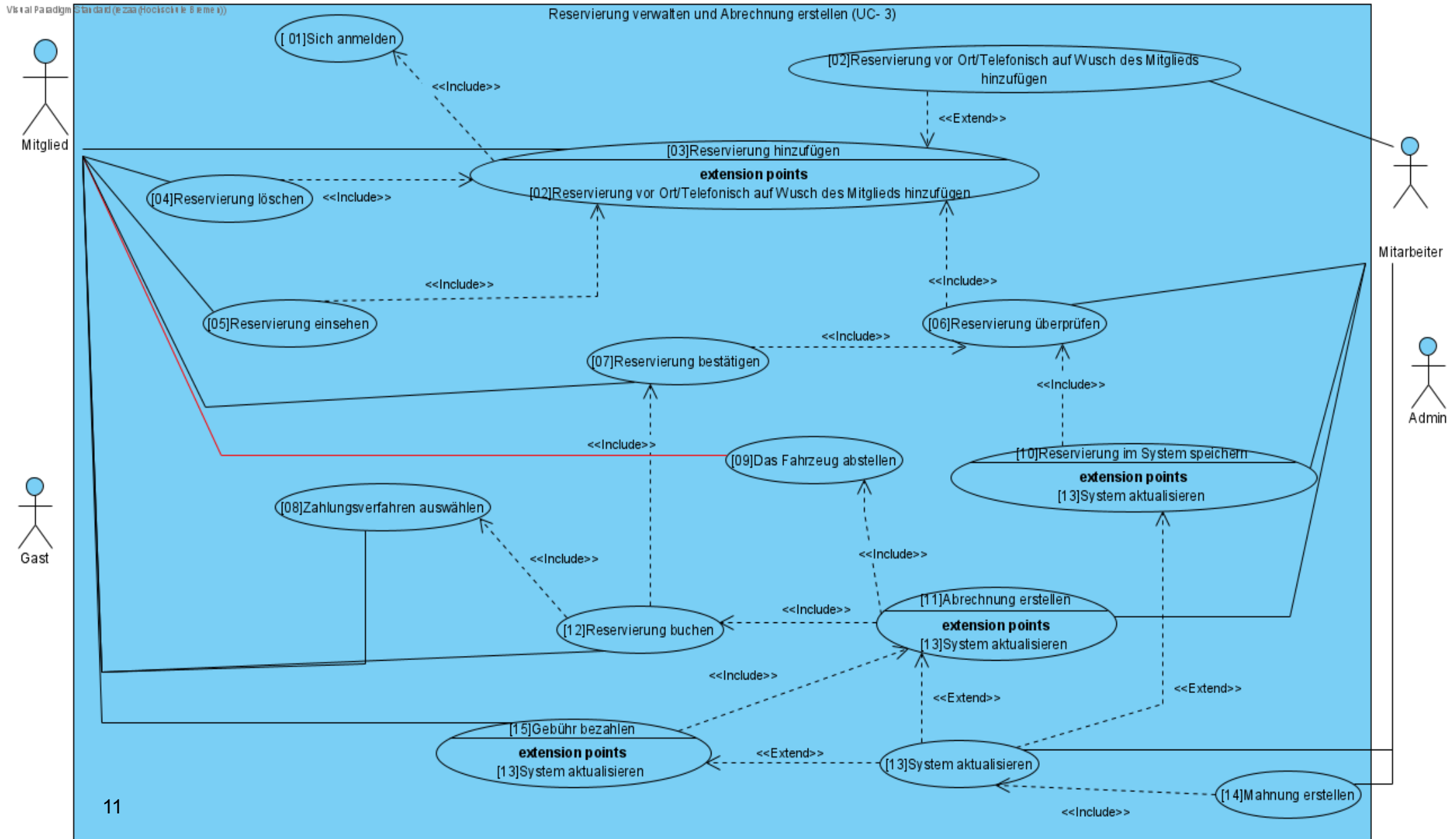
Visual Paradigm Standard (Jakob Spöhrer (Hochschule Bremen))



Der Ausleihstationenreiter in der Webanwendung kann nur ausgewählt werden wenn man als Mitarbeiter oder Admin eingeloggt ist. Von hier aus kann dann die Belegung einer Ausleihstation mit Autos anhand der bevorstehenden Buchungen geplant werden. Es können außerdem die vergangenen Buchungen und die dazugehörigen Belegungen betrachtet werden. Um die Belegung korrekt planen zu können können benötigte Autos von anderen Stationen angefordert und überschüssige Autos zu anderen Stationen transferiert werden. Autos können von hier aus auch in die Wartung geschickt werden.

Anwendungsfall	UC-2
Name	Ausleihstation verwalten
Initiierende Akteur	Mitarbeiter
Weitere Akteure	-
Kurzbeschreibungen	Ausleihstation wird verwaltet, Belegung kann überprüft werden und Station kann mit neuen Autos ausgestattet werden oder Autos können entfernt werden
Vorbedingung	Mitarbeiter muss eingeloggt sein und Ausleihstationen Reiter ausgewählt haben
Nachbedingung	-
Ablauf	<ul style="list-style-type: none"> - Ausleihstationenreiter auswählen - Ausleihstation wählen - Ausleihstationinfos betrachten - Belegung betrachten - Zeitraum evtl ändern für Betrachtung - Belegung anhand der Infos planen für die nächsten Tage - Autos von anderer Station anfordern - vorhandene Autos eventuell in Wartung schicken oder zu anderen Stationen weil zu viel
Alternativen	-
Ausnahmen	Es gibt keine Buchungen und alle Autos sind standardmäßig verfügbar, dadurch besteht kein Bedarf Autos anzufordern oder wegzuschicken.
Benutzte Anwendungsfälle	-
Spezielle Anforderungen	-
Annahmen	-
Referenzen	Laborprojekt Carsharing System Seite 6 und Seite 7 Punkt 5
Nichtfunktionale Anforderungen	-

5.1.3 Use Case Diagramm 3: Reservierung verwalten und Abrechnung erstellen (Reza)



Das Mitglied kann eine Reservierung hinzufügen (UC-03), indem es sich per Webschnittstelle anmeldet (UC-01). Das Mitglied kann auch in eine Filiale des Unternehmens gehen und dort mit Hilfe eines Mitarbeiters die Reservierung hinzufügen (UC-02), welche durch eine Extend-Beziehung dargestellt wurde. Ebenfalls kann das Mitglied die Reservierung über ein Callcenter hinzufügen. Nachdem das Mitglied eine Reservierung hinzugefügt hat, kann es seine Reservierung einsehen (UC-04) und ggf. löschen (UC-05).

Nachdem die Reservierung hinzugefügt wurde, wird die Reservierung vom Mitarbeiter überprüft (UC-06) und im System abgespeichert (UC-10), welche durch die Include-Beziehungen dargestellt wurden. Dabei kann der Mitarbeiter das System aktualisieren (UC-13). Anschließend kann das Mitglied seine Reservierung buchen, indem er vorher seine Reservierung bestätigt hat (UC-07) und ein Zahlungsverfahren ausgewählt hat (UC-08), welche durch die Include-Beziehungen dargestellt wurden. Nachdem das Mitglied das Fahrzeug in einem definierten Bereich abgestellt hat (UC-09), kann der Mitarbeiter eine Abrechnung erstellen (UC-11) und das System aktualisieren (UC-13).

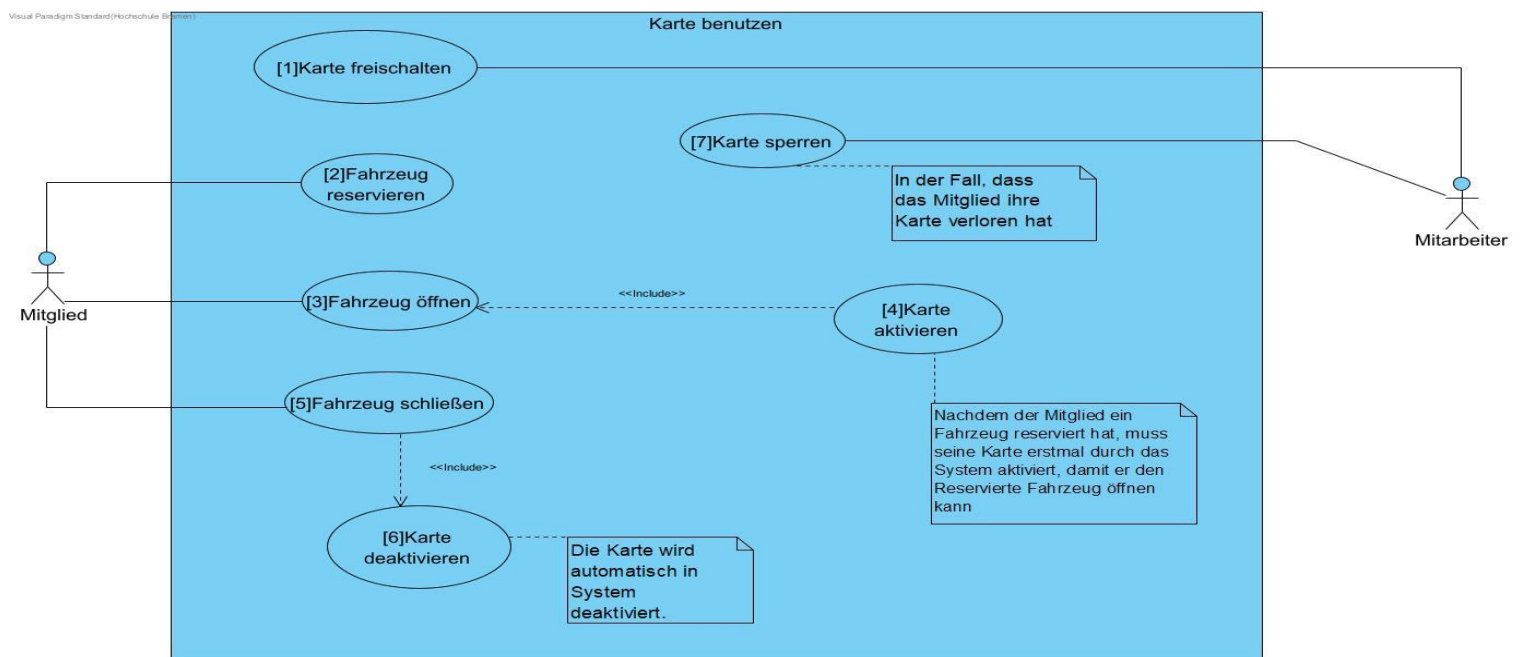
Falls das Mitglied seine Gebühr nicht bezahlt (UC-15) hat, kann der Mitarbeiter eine Mahnung erstellen (UC-14). Da die Verwaltung der Reservierung nur von einem Mitglied oder durch einen Mitarbeiter und die Erstellung der Abrechnung nur von einem Mitarbeiter durchgeführt werden können, sind nur zwei Akteure in diesem Use Case aktiv.

Reservierung hinzufügen:

Anwendungsfall	UC-3
Name	Reservierung hinzufügen
Initiierende Akteur	Mitglied
Weitere Akteure	keine
Kurzbeschreibungen	Das Mitglied fügt eine neue Reservierung hinzu.
Vorbedingung	Das Mitglied hat ein Konto und macht die Reservierung über die Web-Schnittstelle
Nachbedingung	Das Mitglied hat eine neue Reservierung hinzugefügt
Ablauf	<ul style="list-style-type: none"> • Anmelden • Abholzeit auswählen • Abgabezeit auswählen • Abholstandort auswählen • Abstellstandort auswählen • Tarife auswählen • Fahrzeug auswählen • Anfrage senden
Alternativen	Wenn das Mitglied den Prozess abbricht werden die Daten verworfen.
Ausnahmen	Wenn das Mitglied die gewünschten Daten (Fahrzeug, Abstell- und Abholstandort, Abgabe- und Abholzeit und Tarife) im System nicht finden kann, dann wird der Prozess abgebrochen und die Daten verworfen. In diesem Fall kann das Mitglied die Reservierung in einer Filiale/Telefonisch durch einen Mitarbeiter erledigen.

Benutzte Anwendungsfälle	<ul style="list-style-type: none"> • Sich Anmelden (UC 01) • Reservierung vor Ort/Telefonisch auf Wunsch des Mitglieds hinzufügen (UC 02)
Spezielle Anforderungen	Die Daten müssen verschlüsselt übermittelt werden.
Annahmen	Es gibt eine Überprüfung , dass die eingegebenen Daten einer Korrektheit entsprechen z.B. keine doppelte Buchung eines Fahrzeuges im selben Zeitraum.
Referenzen	Kundenanforderung 3,4,5
Datenanforderungen	E-Mail-Adresse, Fahrzeugdaten , Ort und Zeit
Nichtfunktionale Anforderungen	Eine strukturierte grafische Benutzeroberfläche (GUI), die die korrekte Eingabe offensichtlich macht.

5.1.4 Use Case Diagramm 4: Karte benutzen(Kuimo Mariella)



Das Mitglied kann die Karte nutzen, um das Fahrzeug öffnen und schließen, das er reserviert hat. Aber er muss natürlich schon ein aktiviertes Mitgliedschaftskonto haben. Nach der Aktivierung seines Konto bekommt er eine RFD- Karte, die freigeschaltet wird (UC1). Dann wenn der Kunde (Mitglied) ein Fahrzeug nutzen möchte, muss er vorher eine Reservierung machen (UC2). Nach der erfolgreichen Buchung des Fahrzeugs kann das Mitglied das gewählte Fahrzeug öffnen (UC3). Damit der Kunde dieses ausgewählte Fahrzeug öffnen kann, muss seine Karte nur für diese Reservierung erstmal automatisch durch das System aktiviert (UC4). Wenn die Buchungszeit schon fertig ist, muss das Mitglied das Fahrzeug zurückgeben und es wieder mit der Karte schließen (UC5). Danach wird ihre Karte wieder automatisch durch das System deaktiviert (UC6).

Das Mitglied kann auch ihre Karte verloren und in diesem Fall wird der Mitarbeiter die verlorene Karte sperren (UC7).

5.1.4.1 Karte benutzen

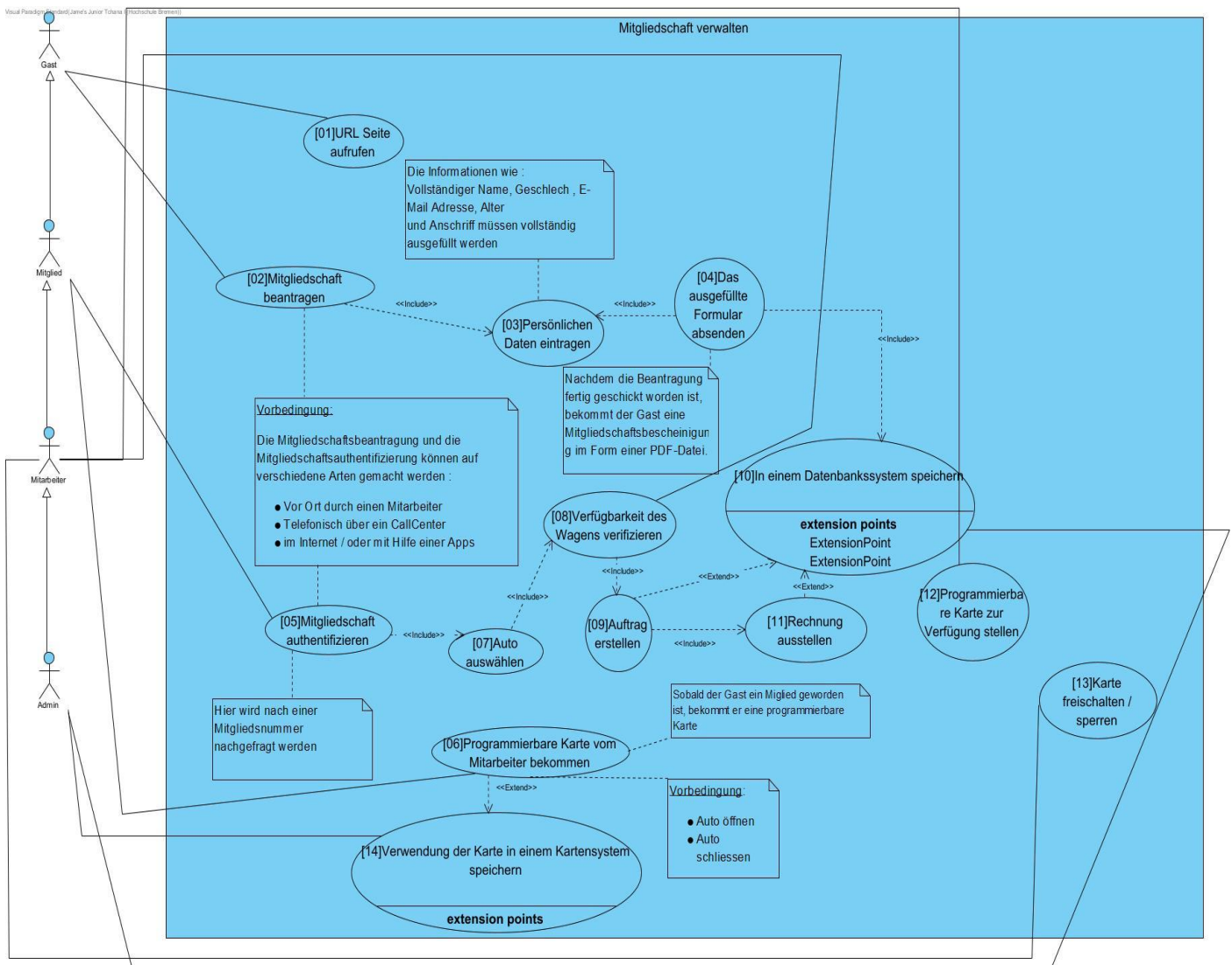
Da das Diagramm nicht so viel Use Case hat, wurde eine Schablone für das ganze Diagramm gemacht.

Anwendungsfall	UC-5
Name	Karte benutzen
Initiierende Akteur	Mitglied
Weitere Akteure	Mitarbeiter
Kurzbeschreibungen	Das Mitglied benutzt die Karte für die Öffnung und das Schließen von Reservierten Fahrzeuge.
Vorbedingung	<ul style="list-style-type: none"> - Mitgliedskonto haben - Karte muss freigeschaltet sein
Nachbedingung	<ul style="list-style-type: none"> - Karte deaktiviert für dieses Fahrzeug
Ablauf	<ul style="list-style-type: none"> · Fahrzeug reservieren · Karte wird aktiviert · reserviertes Fahrzeug öffnen · Nach der Fahrt Fahrzeug bei Rückgabe schließen. · Karte wird deaktiviert
Alternativen	keine
Ausnahmen	Karte ist gesperrt. Wenn das Mitglied seine Karte verliert, wird die Karte gesperrt.
Benutzte Anwendungsfälle	. UC1, UC2, UC3, UC4, UC5, UC6, UC7
Spezielle Anforderungen	-
Annahmen	-
Referenzen	Kundenanforderungen 7
Nichtfunktionale Anforderungen	-

5.1.4.2 Karte sperren

Anwendungsfall	UC-5.07
Name	Karte sperren
Initiierende Akteur	Mitarbeiter
Weitere Akteure	Mitglied
Kurzbeschreibungen	Der Mitarbeiter kann die Karte eines Mitglieds sperren, wenn dieses ihre Karte verloren hat.
Vorbedingung	<ul style="list-style-type: none"> - Mitgliedskonto ist aktiviert - Karte muss freigeschaltet sein - Das Mitglied hat Bescheid gesagt, dass seine Karte verloren ist. - Mitarbeiter ist eingeloggt
Nachbedingung	<ul style="list-style-type: none"> - Karte ist gesperrt - Die Karte kann nicht mehr genutzt werden - Kunde kann eine andere Karte bekommen
Ablauf	<ul style="list-style-type: none"> · Mitarbeiter loggt sich ein · Mitarbeiter sperrt die Karte · Mitarbeiter bestellt eine neue Karte
Alternativen	keine
Ausnahmen	Wenn das Mitglied Schulden hat, kann es keine neue Karte bekommen, bevor es die Schulden ausgleichen.
Benutzte Anwendungsfälle	UC7
Spezielle Anforderungen	-
Annahmen	-
Referenzen	-
Nichtfunktionale Anforderungen	-

5.1.5 Use Case Diagramm 5: Mitgliedschaft verwalten(Jame's)



Der Gast kann auf die Webseite zugreifen ohne dass er sich anmelden muss. Allerdings braucht er eine Mitgliedschaft, um ein Fahrzeug reservieren zu können. Dafür muss er eine Mitgliedschaft beantragen (UC-02). Die Beantragung kann nicht nur vor Ort durch einen Mitarbeiter, oder telefonisch durch ein Callcenter durchgeführt werden, sondern auch online über die Webseite des Carsharings Unternehmen. Sobald er die Beantragung abgeschickt hat und eine Bescheinigung erhalten hat, kann er sich als Mitglied einloggen (UC-05), ein Auto auswählen (UC-06), und gucken ob das ausgewählte Auto verfügbar ist oder nicht (UC-07). Außerdem bekommt er auch von dem Mitarbeiter eine programmierbare Karte, nach dem er eine Mitgliedsbescheinigung gekriegt hat. Dank der erhaltenen Karte kann das Mitglied die Autos benutzen. Dafür muss diese programmierbare Karte vorher freigeschaltet sein und dann nach der Benutzung des Autos wieder versperret sein (UC-13).

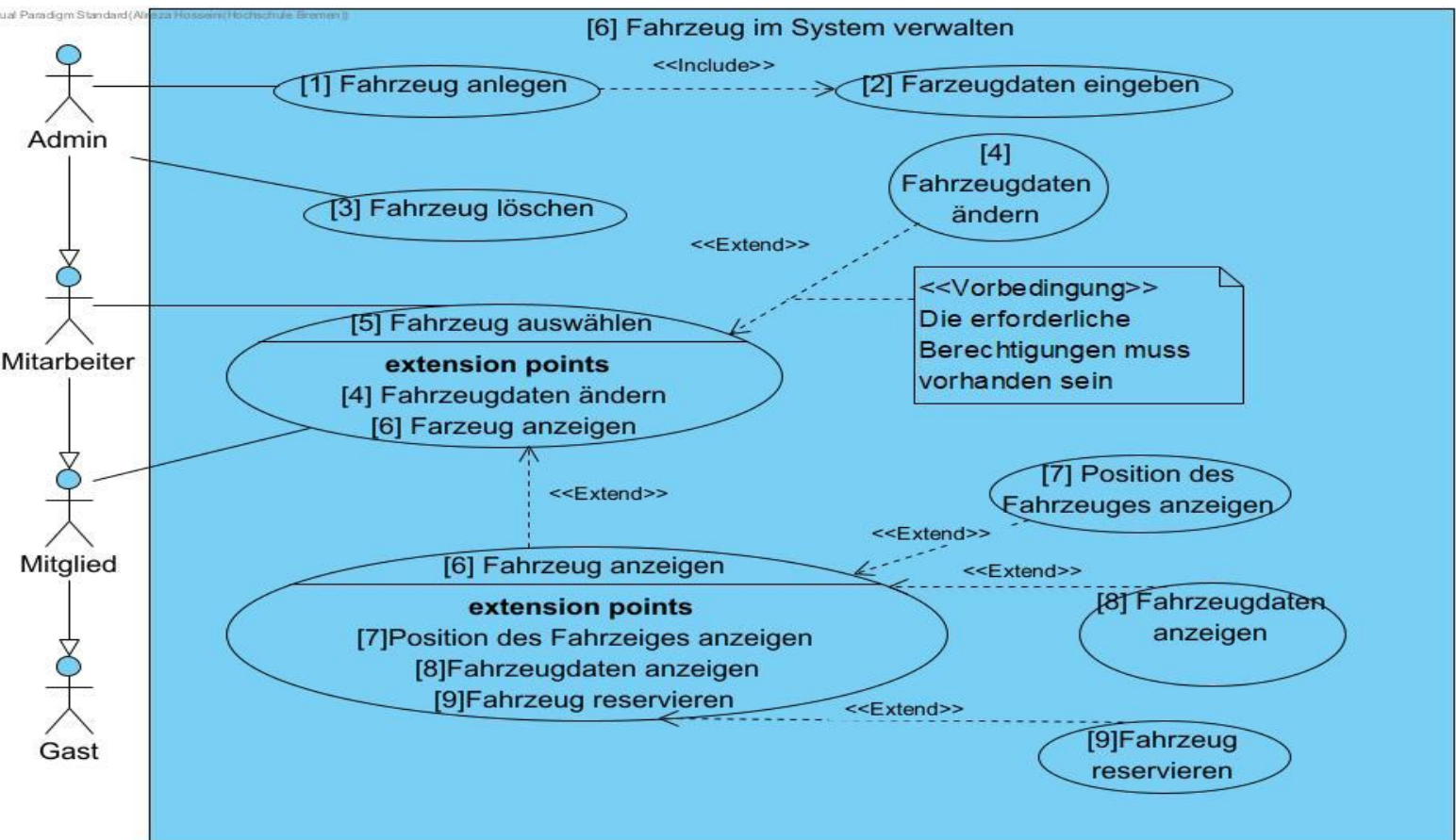
Auf der anderen Seite soll der Mitarbeiter für jede Reservierung einen Auftrag erstellen (UC-09) und Rechnungen ausstellen (UC-11), die in einem Datenbanksystem gespeichert werden müssen (UC-10).

Darüber hinaus stellt der Mitarbeiter dem Mitglied eine programmierbare Karte zur Verfügung, deren Verwendung auch in einem Kartensystem gespeichert werden sollte.

Der Admin kümmert sich um die Verwaltung der verschiedenen Sicherungssysteme (UC-10) und (UC-14).

Anwendungsfall	UC-4
Name	Mitgliedschaft verwalten
Initiierende Akteur	Gast
Weitere Akteure	Mitglied, Mitarbeiter, Admin
Kurzbeschreibungen	Der Gast möchte Mitglied werden, um die Vorteile des System nutzen zu können.
Vorbedingung	- noch kein Mitgliedskonto vorhanden - Der Gast macht eine Mitgliedschaftsbeantragen.
Nachbedingung	Mitgliedschaftsbescheinigung erhalten Programmierbare Karte bekommen
Ablauf	<ul style="list-style-type: none"> - URL Website aufrufen - Mitgliedschaft beantragen - persönlichen Daten eintragen - ausgefülltes Formular absenden - Mitgliedschaftsbescheinigung in Form einer PDF-Datei bekommen - RFID signals Karte bekommen - Auto auswählen - Autosinformationen herausrufen - Verfügbarkeit des Autos überprüfen - Antrag senden - Auto öffnen und schließen
Alternativen	keine
Ausnahmen	keine
Benutze Anwendungsfälle	<ul style="list-style-type: none"> - Sich einloggen - Mitgliedschaft beantragen (vor Ort durch einen Mitarbeiter, oder Telefonisch über ein CallCenter, oder auch einfach Online / mit Hilfe einer s.
Speziellen Anforderungen	<ul style="list-style-type: none"> • Alle Daten müssen in einer Datenbank gespeichert werden. • Die Verwendung der RFID signal Karte muss im Kartensystem archiviert werden.
Annahmen	-
Referenzen	Kundenanforderungen 2,3,4,6,7
Nicht funktionalen Anforderungen	-

5.1.6 Use Case Diagramm 6: Fahrzeugdaten ändern (Alireza)



Dieses System wird benutzt, um Fahrzeugdaten im System zu pflegen. Die Admins können ein neues Fahrzeug dem System hinzufügen oder vom System löschen. Die Mitarbeiter können die Fahrzeugdaten im System korrigieren, wenn sie die Berechtigung bekommen haben dies zu tun. Die Mitglieder können die Standortdaten oder eine allgemeine Übersicht des gemieteten Fahrzeuges abfragen.

Anwendungsfall	UC-13
Name	Fahrzeugdaten ändern
Initiierender Akteur	Mitarbeiter
Weitere Akteure	Admin
Kurzbeschreibung	Die Daten des Fahrzeuges können geändert werden.
Vorbedingung	<ul style="list-style-type: none"> - Der Mitarbeiter hat das Recht zum Ändern der Daten. - Das Fahrzeug ist bereits im System angelegt.
Nachbedingung	-Die geänderten Daten werden gespeichert.

Ablauf	Der Admin oder die Mitarbeiter mit den bestimmten Rechten können ein Fahrzeug auswählen und die Daten ändern.
Alternativen	Das Fahrzeug kann vom System gelöscht werden und nochmal mit den richtigen Informationen angelegt werden.
Ausnahmen	-
Benutzte Anwendungsfälle	UC14 , UC13
Spezielle Anforderungen	-
Annahmen	Der Administrator oder der Mitarbeiter hat sich im System eingeloggt
Offene Themen	-
Referenzen	-
Datenanforderungen	Die geänderten Daten werden im System gespeichert.
Nichtfunktionale Anforderungen	-

5.2 Funktionale Anforderungen

Anforderungs-Nr.	Beschreibung	Use-Case	Verifikationsmethode	Klassifizierung der Anforderung
FA-1	Das Mitglied soll den Abstellstandort auswählen können.	UC3.01	T	muss
FA-2	Das Mitglied soll den Abholstandort auswählen können.	UC3.01	T	muss
FA-3	Das Mitglied soll die Abholzeit angeben können.	UC3.01	T	muss
FA-4	Das Mitglied soll die Abgabezeit eingeben können.	UC3.01	T	muss
FA-5	Das Mitglied soll das Fahrzeug auswählen können.	UC3.01	T	muss
FA-6	Das System soll die Reservierung überprüfen können.	UC3.06	T	muss
FA-7	Das Mitglied soll seine Gebühr über folgende Zahlungsmethode bezahlen können. <ul style="list-style-type: none"> • PayPal • Kreditkarte Sofortüberweisung • Zahlung per Lastschrift 	UC3.08	T	kann
FA-8	Die Reservierung soll persistent im System gespeichert werden.	UC3.10	I,RoD,T	muss
FA-9	Das Mitglied soll seine Reservierung sehen können	UC3.05	T	muss
FA-10	Das Mitglied soll seine Reservierung löschen können	UC3.04	T	muss
NFA-1 1	Es sollen drei Tarife (Basic, Ermäßigt und Exklusiv) vorhanden sein.	UC1.09	I,RoD	soll
FA-12	Das Mitglied soll die Karte automatisch zugeschickt bekommen.	-	I	soll

FA-13	Das Mitglied soll ein Fahrzeug reserviert haben.	UC5.02	I	soll
FA-14	Das Mitglied soll das Fahrzeug mit der Karte öffnen können	UC5.03	I	soll
FA-15	Die Karte soll aktiviert worden sein	UC5.04	T	soll
FA-16	Das Mitglied soll das Fahrzeug mit der Karte schließen können.	UC5.05	T	soll
FA-17	Die Karte soll deaktiviert worden sein.	UC5.06	T	soll
FA-18	Der Mitarbeiter soll Ausleihstationenreiter auf Website auswählen können.	UC2.01	T	soll
FA-19	Der Mitarbeiter soll die Informationen über die Ausleihstation betrachten können.	UC2.02	I	soll
FA-20	Der Mitarbeiter soll eine Ausleihstation auswählen können.	UC2.03	T	soll
FA-21	Der Mitarbeiter soll den Status eines Autos in einer Ausleihstation ändern können.	UC2.04	T	soll
FA-22	Der Mitarbeiter soll ein Auto von einer Ausleihstation zu einer anderen Ausleihstation weg transferieren können.	UC2.05	T	soll
FA-23	Der Mitarbeiter soll ein Auto in einer Ausleihstation in die Wartung schicken können.	UC2.06	T	soll
FA-24	Der Mitarbeiter soll die Belegung der Ausleihstation planen können.	UC2.07	T	soll
FA-25	Der Mitarbeiter soll ein Auto für eine Ausleihstation anfordern können.	UC2.08	T	soll
FA-26	Der Mitarbeiter soll die aktuelle Belegung einer Ausleihstation mit Autos betrachten können.	UC2.09	T	soll
FA-27	Der Mitarbeiter soll die zukünftige Belegung einer Ausleihstation mit Autos betrachten können.	UC2.10	T	soll

FA-28	Der Mitarbeiter soll die vergangene Belegung einer Ausleihstation mit Autos betrachten können.	UC2.11	T	soll
FA-29	Der Mitarbeiter soll die zukünftigen Buchungen einer Ausleihstation betrachten können.	UC2.12	T	soll
FA-30	Der Mitarbeiter soll die vergangenen Buchungen einer Ausleihstation betrachten können.	UC2.13	T	soll
FA-31	Der Mitarbeiter soll die aktuellen Buchungen einer Ausleihstation betrachten können.	UC2.14	T	soll
FA-32	Der Mitarbeiter soll den Betrachtungszeitraum von Belegung und Buchungen einer Ausleihstation ändern können.	UC2.15	T	soll
FA-33	Der Mitarbeiter soll immer die Belegung und Buchung der Ausleihstation des gleichen Zeitraumes angezeigt bekommen können.	UC2.16	I	soll
FA-34	Der User (Mitarbeiter, Admin und Mitglied) soll sich im System anmelden können.	UC1.01	T	muss
FA-35	Bei der Eingabe des Kennwortes soll das Kennwort nicht sichtbar sein.	UC1.03	T	kann
FA-36	Das Kennwort soll verschlüsselt gespeichert sein.	UC1.04	I	soll
FA-37	Der User kann sich nicht mehr einloggen, wenn er 5 Mal falsche Daten eingegeben hat.	UC1.01	T	kann
FA-38	Der User soll sich abmelden können	UC1.07	T	muss
FA-39	Das Kennwort soll zurückgesetzt werden	UC1.04	T	soll
FA-40	Wenn der User falsche Kennwort oder E Mail eingegeben hat, soll er eine Fehlermeldung sehen.	UC1.06	T	soll
FA-41	Ein Gast soll die Sprache wechseln können.	UC1.08	T	kann

FA-42	Ein Admin soll ein Fahrzeug im System anlegen können	UC6.01	T	soll
FA-43	Ein Admin soll ein Fahrzeug im System löschen können.	UC6.03	T	soll
FA-44	Ein User soll ein Fahrzeug im System auswählen können.	UC6.05	T	soll
FA-45	Ein Admin soll die Daten eines Fahrzeuges ändern können.	UC6.04	T	soll
FA-46	Ein Mitarbeiter soll die Daten eines Fahrzeug ändern können, wenn er die Berechtigung hat.	UC6.04	T	kann
FA-47	Ein User soll die Fahrzeugdaten sehen können	UC6.08	T	soll
FA-48	Das Softwaresystem soll an das Buchhaltung des Unternehmens verbinden werden.	-	T	kann
FA-49	Der Gast soll eine Mitgliedschaft beantragen können	UC4.03	T	muss
FA-50	Der Mitarbeiter soll dem Mitglied eine programmierbare Karte zur Verfügung stellen können.	UC4.12	T	soll
FA-51	Die Aktivität der Karte soll in der Datenbank gespeichert werden können	UC4.14	T	soll
FA-52	Die Mitarbeiter sollen die Mitglieder verwalten können(CRUD-Funktionen)	UC4.10	T	muss
FA-53	Der Admin soll die Karten verwalten können (CRUD-Funktionen)	UC4.14	T	soll
FA-54	Die Karte soll freigeschaltet worden sein.	UC5.01	T	soll
FA-55	Der User soll automatisch abgemeldet werden, wenn er 5 Minuten lang keine Aktion durchführt.	UC1,07	T	kann

5.3 Schnittstellen Anforderungen (Alireza)

→ Web-Schnittstelle

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
SA-1	Durch die Web-Schnittstelle sollen neue Einträge in der Datenbank angelegt werden.	T	muss
SA-2	Die existierende Einträge sollen geändert oder gelöscht werden können.	T	muss
SA-3	Durch eine Abfrage soll die Karte eines Kunden geprüft werden und bei bestehenden Reservierung soll das zugehörige Fahrzeug aufgeschlossen werden.	T	muss

→ Fahrzeug

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
SA-4	Das Fahrzeug soll Requests von der Web-Schnittstelle erhalten können.	T	muss
SA-5	Das Fahrzeug soll auf die entsprechenden Request reagieren, den Zustand des Fahrzeugs Systems ändern können und ggf. eine Response senden.	T	muss
SA-6	Der Standort eines Fahrzeug soll abgefragt werden können.	T	soll

→ Buchhaltungssystem

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
SA-7	Ein Anbindung an das bestehende Buchhaltungssystem kann implementiert werden.	T	kann

5.4 Bedienbarkeits/Operationalitäts-Anforderungen(Jakob)

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
BOA-1	Die Anwendung soll im Chrome-Browser aufgerufen werden können.	T	muss

5.5 Qualitätsanforderungen (Reza)

5.5.1 Zuverlässigkeitsanforderungen

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
QA-1	Das Gesamtsystem soll eine Verfügbarkeit von 95% aufweisen.	A	Muss
QA-2	Die Software soll ausfallsicher sein.	T	soll

5.5.2 Leistungs-/Performanceanforderungen

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
QA-3	Jede Datenbankabfrage soll in maximal 30 Sekunden verarbeitet werden.	T	soll

5.5.3 Sicherheitsanforderungen

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
QA-4	Alle Daten sollen verschlüsselt übermittelt werden.	T,A	muss

5.5.4 Kompatibilitäts-/Portabilitätsanforderungen

Anforderungs Nr.:	Beschreibung	Verifikationsmethode	Klassifizierung der Anforderung
QA-5	Das Softwaresystem, das über eine URL zu erreichen ist, soll von unterschiedlichen Geräten (Smartphone, Tablets und Stand-PCs) aufgerufen werden kann	T	soll

5.6 Konfigurationen, Ausbaustufen, Varianten(Jame's)

In der ersten version **V-01** ist nach der Implementierungsphase (25.06.2022) ein Softwaresystem zu liefern. Dies besitzt nur die Basisanforderungen des Produktsystems und leistet seinen ersten Einsatz.

D.h. dadurch können Fahrzeuge von einem festen Mietparkplatz in der Stadt oder von einer größeren Fläche, die auf einem öffentlichen Straßenraum geparkt ist, abgeholt werden. Feste Mietstationen befinden sich häufig an Flughafen Knotenpunkten, Bahnhöfen, Straßenbahn Knotenpunkten, Busbahnhöfen usw. schneller und einfacher Zugriff für Kunden.

In der Version **V-02** werden nach Bewertungen der Benutzer unkritische Systemanforderungen in dem Softwaresystem übertragen.

In der Version **V-03** wird eine Android/Desktop oder IOS-Applikation mit vollständigen Anforderungsgrenzen.

5.7 Grenzen und Einschränkungen – Was das System nicht kann(Mariella Kuimo)

- Das System kann nicht ein Fahrzeug an Jemanden ausleihen, der sich nicht in Europa befindet. Da die gesamte Nutzung der Software nur in Europa anwendbar und nutzbar ist.
- Das System ist nicht Barrierefrei.

6 Lieferumfang – Liste der zu liefernden HW-/SW-Einheiten(Mariella Kuimo)

Das Produkt ist eine kompakte Datei, die mit zwei Server und einer zugehörigen Datenbank geliefert wird.

Die zu liefernden Hardware Einheiten sind:

- zwei Server, um die Applikation deployen zu können.
- Raspberry Pi 4G
- UMTS Antenne
- GPS-Sensor
- Scanner
- RFID-Karte

Die zu liefernden Software Einheiten sind:

- Web-Anwendung/Applikation

7 Funktionsprüfung/Abnahmekriterien(Mariella Kuimo)

7.1 Funktionsprüfung

Die Abnahmekriterien sind die Anforderungen des Pflichtenheftes und werden durch die folgenden Tests geprüft.

- **Unit-Tests** : Um die einzelnen Methoden und Funktionen der Klassen, der Komponenten sowie der Module, die in der Software sind. Der Test wird automatisiert durchgeführt.
- **Integrationstests** : hier wird getestet, ob die verschiedene Anwendungen der genutzte Module ohne Probleme interagieren (Datenaustausch). Dadurch wird sichergestellt, dass die Schnittstellen fehlerfrei zusammenarbeiten. Der Test wird sowohl automatisiert als auch manuell durchgeführt.
- **Funktionstests** : um die Ausgabe von Aktionen zu verifizieren. Wird manuell durchgeführt.
- **Akzeptanztests** wird manuell durchgeführt
- **Leistungstests** : Um das Verhalten des Systems zu überprüfen bzw. die Zuverlässigkeit, Stabilität und Verfügbarkeit. Der Test wird automatisiert durchgeführt.

Die Software wird mit dem **White-Box** sowie **Black-Box** Testverfahren, Unit Test getestet.

Der White-Box Test dient dazu die Programmstruktur der Software zu testen das heißt zu testen ob die unterschiedlichen Funktionalitäten der Software wie gewünscht verlaufen. Da der White-Box auf der Kenntnis der inneren Struktur des Systems basiert ist.

Mit dem Black-Box Test wird das äußerliche Verhalten der Software ohne Kenntnis der Softwarestruktur getestet.

NB: Am Ende sollen alle Testen die Ausfallsicherheit, die Funktionalität, die Zuverlässigkeit, die Benutzbarkeit, die Effizienz, die Änderbarkeit, die Übertragbarkeit der Software prüfen und verifizieren.

7.2 Abnahmekriterien

An dem Liefertag muss die erste Version der Applikation geliefert werden, dazu müssen alle Server konfiguriert sowie alle Hardware installiert werden. Nach der Installation, wird die initiale Version nur über Grundfunktionen des Systems verfügen. Wichtig ist, dass der Auftraggeber bei Abnahme jeder Version anwesend sein.

Die andere Versionen werden mit Abnahmetest getestet und genehmigt. Der Auftraggeber muss eventuell wieder anwesend sein. Es wird dabei die Anwendungsfallsbezogene Funktionstests durchgeführt, welche der richtige Anwendung der Applikation beweisen können.

7.3 Testhilfsmittel

- JUnit und Mockito für das Backend. JUnit dient für das White und Black-Box Testverfahren sowie für den Integrationstest.
- npm Test für das Frontend.