Projekt: U-Car

Anforderungsspezifikationen

# Dokumenteninformation

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderungsgrund | Autorin |
| 26.03.23 | 1.0 | Erstellung | Wenzl |
| 01.04.23 | 1.1 | Ergänzung nichtfunktionaler Anforderungen | Wenzl |
| 13.04.23 | 1.2 | Ergänzung funktionaler Anforderungen, Use Cases und Use Case Diagramm | Kronberg |
| 20.04.23 | 1.3 | Aktualisierung UC fully dressed / Wartbarkeit | Kenmo |
| 03.05.23 | 1.4 | Aktualisierung nichtfunktionale Anforderungen | Wenzl |

## Inhalt

[1 Dokumenteninformation 2](#_Toc134009614)

[1.1 Änderungsgeschichte 2](#_Toc134009615)

[1.2 Inhalt 3](#_Toc134009616)

[2 Einführung 5](#_Toc134009617)

[2.1 Definitionen und Abkürzungen 5](#_Toc134009618)

[2.2 Referenzen 5](#_Toc134009619)

[3 Allgemeine Beschreibung 5](#_Toc134009620)

[3.1 Produktperspektive 5](#_Toc134009621)

[3.2 Produktfunktion 5](#_Toc134009622)

[3.3 Benutzercharakteristik 5](#_Toc134009623)

[3.4 Einschränkungen 6](#_Toc134009624)

[3.5 Annahmen 6](#_Toc134009625)

[3.6 Abhängigkeiten 6](#_Toc134009626)

[3.7 Use Case Überblick 6](#_Toc134009627)

[4 Spezifische Anforderungen 6](#_Toc134009628)

[4.1 Funktionale Anforderungen 6](#_Toc134009629)

[4.1.1 Funktionale Anforderung F1 – Kontoregistrierung und -anmeldung 6](#_Toc134009630)

[4.1.2 Funktionale Anforderung F2 – Fahrzeugregistrierung 6](#_Toc134009631)

[4.1.3 Funktionale Anforderung F3 – Fahrzeugsuche 7](#_Toc134009632)

[4.1.4 Funktionale Anforderung F4 – Fahrzeugbuchung 7](#_Toc134009633)

[4.1.5 Funktionale Anforderung F5 – Fahrzeugabholung und -rückgabe 7](#_Toc134009634)

[4.1.6 Funktionale Anforderung F6 – Rechnung 7](#_Toc134009635)

[4.1.7 Funktionale Anforderung F7 – Stornierung 7](#_Toc134009636)

[4.1.8 Funktionale Anforderung F8 – Administratorkonto 7](#_Toc134009637)

[4.2 Bedienbarkeit 7](#_Toc134009638)

[4.2.1 Bedienbarkeitsanforderung 1 7](#_Toc134009639)

[4.2.2 Bedienbarkeitsanforderung 2 7](#_Toc134009640)

[4.3 Zuverlässigkeit 8](#_Toc134009641)

[4.3.1 Zuverlässigkeitsanforderung 1 8](#_Toc134009642)

[4.4 Leistung 8](#_Toc134009643)

[4.4.1 Leistungsanforderung 1 8](#_Toc134009644)

[4.5 Wartbarkeit 8](#_Toc134009645)

[4.5.1 Wartbarkeitsanforderung 1 8](#_Toc134009646)

[4.6 Sicherheit 8](#_Toc134009647)

[4.6.1 Sicherheitsanforderung 1 8](#_Toc134009648)

[4.7 Schnittstellen 8](#_Toc134009649)

[4.7.1 Benutzerschnittstelle 8](#_Toc134009650)

[4.7.2 Datenbankschnittstelle 8](#_Toc134009651)

[4.8 Installation 8](#_Toc134009652)

[5 Use Cases 9](#_Toc134009653)

[5.1 Use Case – Diagramm 9](#_Toc134009654)

[5.2 Use Case UC1: Konto registrieren/anmelden 9](#_Toc134009655)

[5.3 Use Case UC2: Fahrzeug registrieren 10](#_Toc134009656)

[5.4 Use Case UC3: Fahrzeug suchen 11](#_Toc134009657)

[5.5 Use Case UC4: Fahrzeug buchen 12](#_Toc134009658)

[5.6 Use Case UC5: Fahrzeug abholen und zurückbringen 13](#_Toc134009659)

[5.7 Use Case UC6: Buchung stornieren 14](#_Toc134009660)

# Einführung

## Definitionen und Abkürzungen

Nutzer bezeichnet die Person ohne Berechtigungen.

Fahrzeugeigentümer bezeichnet die Person, die berechtigt ist, Fahrzeuge zu registrieren.

Wenn beide zusammen auftreten, stellt der Nutzer den Mieter und der Fahrzeugeigentümer den Vermieter dar, unabhängig von den anderen Berechtigungen des Nutzers.

## Referenzen

# Allgemeine Beschreibung

## Produktperspektive

U-Car ist eine Anwendung für Privatpersonen, die ihr Autos nicht oft nutzen oder ein Auto benötigen, aber keines besitzen. Fahrzeugeigentümer können ihre Fahrzeuge registrieren und für bestimmte Zeiträume zur Buchung freigeben. Andere Nutzer können eine Reservierung oder Buchung anfragen, die durch den Eigentümer bestätigt werden muss. Dadurch können sich die Nutzer für ihren gewünschten Zeitraum und gegen eine Gebühr ein Auto ausleihen, während die Fahrzeugeigentümer am Monatsende eine Gutschrift je nach Entleihdauer erhalten.

## Produktfunktion

* Anmeldung mit verschiedenen Berechtigungen für Privatpersonen, Fahrzeugeigentümer und Administratoren
* Ausgeben einer Liste von verfügbaren Fahrzeugen und Buchung eines Fahrzeugs
* Bestätigung von Buchungen durch Fahrzeugeigentümer
* Nutzerkonto mit Profilinformationen und Zahlungsdetails
* Registrierung von Fahrzeugen durch die Eigentümer
* Fahrzeugübergabe- und -rückgabeoptionen, einschließlich Standortinformationen und Zustandsberichten
* Ausstellen einer Rechnung für jede Buchung

## Benutzercharakteristik

Zielgruppe des Produkts sind einerseits Menschen, die ihr eigenes Auto nur selten nutzen und deswegen nach einer Möglichkeit suchen, es an andere zu vermieten. Andererseits richtet sich U-Car aber auch an diejenigen, die selbst kein Auto besitzen und deshalb nur für eine bestimmte Zeit ein Fahrzeug mieten wollen.

## Einschränkungen

Benachrichtigungen, die eigentlich per E-Mail an die Nutzer verschickt werden, werden stattdessen in ein Textdokument geschrieben.

## Annahmen

Die vom Nutzer eingegebenen Daten werden als korrekt angesehen.

## Abhängigkeiten

Unsere Anwendung ist von der funktionsfähigen Anbindung einer Datenbank abhängig.

## Use Case Überblick

* UC1: Konto registrieren/anmelden
* UC2: Fahrzeug registrieren
* UC3: Fahrzeug suchen
* UC4: Fahrzeug buchen
* UC5: Fahrzeug abholen und zurückbringen
* UC6: Buchung stornieren

# Spezifische Anforderungen

## Funktionale Anforderungen

### Funktionale Anforderung F1 – Kontoregistrierung und -anmeldung

Die Nutzer sollen sich registrieren/ anmelden können, um Zugang zum Carsharing-Dienst zu erhalten. Bei der Registrierung sollen sie ihre persönlichen Daten wie Name, Adresse, Geburtsdatum, E-Mail-Adresse und Passwort angeben können. Dabei soll das Passwort aus mindestens 8 Zeichen bestehen. Bei der Anmeldung sollen Nutzer ihren Namen und ihr Passwort eingeben können.

### Funktionale Anforderung F2 – Fahrzeugregistrierung

Der Fahrzeugeigentümer soll Fahrzeuge in seinem Profil registrieren können. Dabei soll er einige Angaben zu seinem Fahrzeug machen, wie z.B. das Fahrzeugmodell, die Anzahl der Sitzplätze und die Adresse. Bei der Adressauswahl soll das System bereits bestehende Stellplatzadressen anzeigen, um dem Fahrzeugeigentümer die Möglichkeit zu geben, aus diesen auszuwählen. Nach Bestätigung durch den Fahrzeugeigentümer soll das System das Fahrzeug und ggf. den neuen Stellplatz anlegen.

### Funktionale Anforderung F3 – Fahrzeugsuche

Die Nutzer sollen die im System gespeicherten verfügbaren Fahrzeuge suchen können. Sie sollen den gewünschten Standort, den Fahrzeugtyp und das Datum angeben können.

### Funktionale Anforderung F4 – Fahrzeugbuchung

Nachdem der Nutzer das gewünschte Fahrzeug gefunden hat, soll er dieses buchen können. Das System sollte dann den Fahrzeugeigentümer informieren und auf dessen Bestätigung warten, bevor das System das Fahrzeug endgültig bucht.

### Funktionale Anforderung F5 – Fahrzeugabholung und -rückgabe

Bei der Fahrzeugübernahme soll der Nutzer seine Startzeit und den aktuellen Kilometerstand ins System eingeben. Zusätzlich soll er die Möglichkeit zur Eingabe spezifischer Informationen, wie z.B. Tankfüllung oder Schäden, haben. Nach der Nutzung soll es der Nutzer die Endzeit, der Kilometerstand und ggf. weitere Informationen vom Nutzer in das System eingegeben können.

### Funktionale Anforderung F6 – Rechnung

Nach jeder Nutzung sollen der Nutzer und der Fahrzeugeigentümer automatisch eine Rechnung erhalten. Diese soll Angaben zum Nutzer, zum Fahrzeugeigentümer, zum Fahrzeugtyp, zum konkreten Fahrzeug und zu den gefahrenen Kilometern enthalten.

### Funktionale Anforderung F7 – Stornierung

Der Nutzer soll die Möglichkeit haben, seine aktuell vorgenommenen Buchungen einschließlich Buchungszeitraum und Fahrzeugtyp einzusehen und zu stornieren. Der Fahrzeugeigentümer soll eine Benachrichtigung über die geänderte Buchung erhalten.

### Funktionale Anforderung F8 – Administratorkonto

Das System soll in der Lage sein, dem Administrator eine Übersicht mit allen Informationen über die bisher erfolgten Buchungen, Nutzer, Fahrzeuge und Stellplätze darzustellen.

## Bedienbarkeit

### Bedienbarkeitsanforderung 1

Die Anwendung soll so gestaltet sein, dass eine Person zwischen 18 und 40 Jahren mit der für diese Altersgruppe typischen Technikaffinität die Bedienung innerhalb von fünf Minuten verstehen kann.

### Bedienbarkeitsanforderung 2

Die Anwendung soll mit Maus und Tastatur bedient werden.

## Zuverlässigkeit

### Zuverlässigkeitsanforderung 1

Beim Ausstellen einer Rechnung soll der Betrag in jedem Fall korrekt berechnet werden, sofern die Daten zu Fahrzeug und Fahrt korrekt eingegeben wurden.

## Leistung

### Leistungsanforderung 1

Das System soll nicht länger als drei Sekunden brauchen, um zu einer anderen Seite zu wechseln. Datenbankanfragen richten sich nach der Größe der Datenbank.

## Wartbarkeit

### Wartbarkeitsanforderung 1

Die Anwendung soll später noch gewartet oder verändert werden können. (Wie???)

## Sicherheit

### Sicherheitsanforderung 1

Für alle Nutzerkonten soll das Berechtigungslevel vermerkt werden, um die Verteilung der Administratorrechte zu sichern.

## Schnittstellen

### Benutzerschnittstelle

Der Benutzer soll die Anwendung am eigenen Rechner verwenden und über ein grafisches Interface steuern können.

### Datenbankschnittstelle

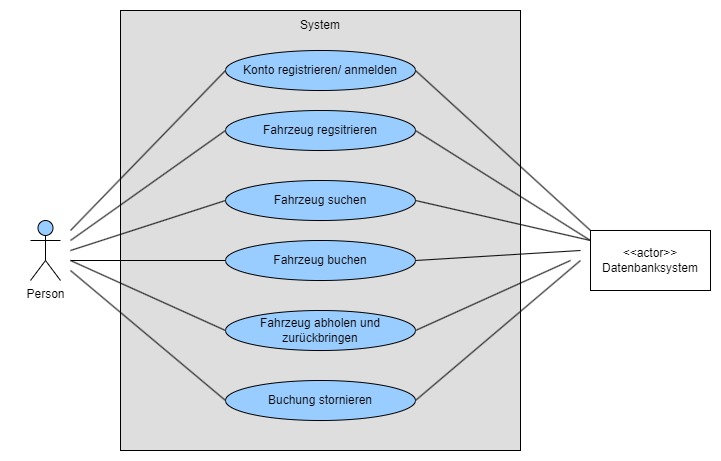
Zur Speicherung der Nutzer- und Fahrzeugdaten soll eine MySQL-Datenbank verwendet werden.

## Installation

Die Anwendung soll nach der Fertigstellung zum Download für Linux bereitstehen.

# Use Cases

## Use Case – Diagramm



## Use Case UC1: Konto registrieren/anmelden

**Primary Actor:** Nutzer

**Stakeholders and Interests:**

* Nutzer: möchte ein neues Konto erstellen/ sich anmelden
* Admin: will, dass die Registrierung oder Anmeldung erfolgreich läuft

**Preconditions:** Der Benutzer hat U-Car installiert.

**Postconditions:** Der Benutzer hat ein neues Konto erstellt/ ist nun in U-Car eingeloggt.

**Main Success Scenario:**

1. Der Benutzer gibt seine Daten (Name und Passwort/ Name, Adresse, Geburtsdatum, E-Mail-Adresse und Passwort) an.
2. Das System überprüft die eingegebenen Informationen auf ihre Richtigkeit und Validität.
3. Bei der Registrierung erstellt das System ein neues Konto für den Benutzer.
4. Das System leitet den Benutzer auf die Hauptseite der Carsharing-Software weiter.

**Extensions:**

2a. Die Daten sind nicht korrekt.

1. Das System benachrichtigt den Nutzer, dass die Daten nicht korrekt sind.

weiter mit 1.

**Special Requirements:**

**Technology und Data Variations List:**

**Frequency of Occurrence:** mehrmals täglich

**Open Issues:**

## Use Case UC2: Fahrzeug registrieren

**Primary Actor:** Fahrzeugeigentümer

**Stakeholders and Interests:**

* Fahrzeugeigentümer: will, dass sein Fahrzeug erfolgreich registriert wurde und alle Angaben stimmen
* Admin: möchte, dass das Fahrzeug und alle Informationen und im Fahrzeugkatalog gespeichert werden

**Preconditions:** Der Nutzer hat die Benachrichtigung des Fahrzeugeigentümer und ist angemeldet.

**Postconditions:** Das neue Fahrzeug und seine Daten wurden gespeichert und es wurde dem Fahrzeugkatalog hinzugefügt.

**Main Success Scenario:**

1. Der Fahrzeugeigentümer gibt Angaben zu seinem Fahrzeug, so wie Fahrzeugmodell, Kennzeichen, Farbe, Anzahlsitzplätze, mögliche Mietzeiträume, Anzahl der Sitzplätze an.
2. Das System zeigt alle bisher gespeicherten Stellplätze an.
3. Der Fahrzeugeigentümer wählt einen dieser Stellplätze aus.
4. Der Fahrzeugeigentümer gibt die Stellplatznummer ein und bestätigt alle Eingaben.
5. Das System speichert das neue Fahrzeug und alle weiteren angegebenen Informationen und fügt es dem Fahrzeugkatalog hinzu.

**Extensions:**

3a. Der Fahrzeugeigentümer will keinen dieser Stellplätze auswählen.

1. Der Fahrzeugeigentümer gibt eine andere Adresse ein.
2. Das System speichert die neue Adresse als Stellplatz.

Weiter bei 4.

**Special Requirements:**

**Technology und Data Variations List:**

**Frequency of Occurrence:** mehrmals täglich

**Open Issues:**

## Use Case UC3: Fahrzeug suchen

**Primary Actor:** Nutzer

**Stakeholders and Interests:**

* Nutzer: will, dass alle verfügbaren Fahrzeuge zu seiner Suche angezeigt werden

**Preconditions:** Der Nutzer hat sich erfolgreich angemeldet.

**Postconditions:** Der Nutzer hat eine Liste der verfügbaren Fahrzeuge erhalten, die seinen Suchkriterien entsprechen.

**Main Success Scenario:**

1. Der Nutzer gibt seine gewünschten Daten für die Suchfunktion, wie z.B. den Standort, den Fahrzeugtyp und das Datum, ein.
2. Das System zeigt eine Liste von Fahrzeugen an, die den Suchkriterien des Nutzers entsprechen.
3. Der Nutzer wählt das gewünschte Fahrzeug aus.

**Extensions:**

\*a. Der Nutzer bricht die Suche ab.

1. Das System zeigt das Suchfenster an.

2a. Es ist kein Fahrzeug mit den gewünschten Daten verfügbar.

1. Das System benachrichtigt den Nutzer und zeigt das Suchfenster an.

**Special Requirements:**

**Technology und Data Variations List:**

**Frequency of Occurrence:** Vor jeder Buchung, mehrmals pro Stunde

**Open Issues:**

## Use Case UC4: Fahrzeug buchen

**Primary Actor:** Nutzer

**Stakeholders and Interests:**

* Nutzer: will, dass das Fahrzeug zu dem Mietzeitpunkt gebucht ist und die Buchung in seiner Buchungsliste vermerkt ist
* Fahrzeugeigentümer: will, dass das Fahrzeug für den angegeben Zeitpunkt gebucht ist und er alle erforderlichen Nutzerdaten erhalten hat
* Admin: will, dass die Buchung erfolgreich läuft

**Preconditions:** Der Nutzer hat ein verfügbares Fahrzeug gefunden.

**Postconditions:** Das Fahrzeug ist für den Nutzer für die gewünschten Zeit gebucht und steht zu dem gebuchten Zeitpunkt zur Verfügung. Die Buchung ist in der Buchungsliste des Nutzers und der Buchungsliste für das Fahrzeug aufgelistet.

**Main Success Scenario:**

1. Das System zeigt das vom Nutzer ausgewählte Fahrzeug an und fragt nach einer Bestätigung der Auswahl.
2. Der Nutzer bestätigt seine Auswahl.
3. Das System bestätigt die Buchung und zeigt die Details der Reservierung an.
4. Das Fahrzeug ist für den Nutzer reserviert und ist zu dem gebuchten Zeitpunkt verfügbar.
5. Das System benachrichtigt den Fahrzeugeigentümer und übermittelt ihm alle erforderlichen Nutzerdaten.
6. Der Fahrzeugeigentümer bestätigt die Buchung.
7. Das System informiert den Nutzer über die Bestätigung.

**Extensions:**

2a. Ein anderer Nutzer hat das gleiche Fahrzeug zur gleichen Zeit gebucht

1. Das System zeigt eine Fehlermeldung an und fordert den Nutzer auf, eine andere Buchung vorzunehmen.
2. Das System zeigt die Suchliste an.

5a. der Fahrzeugeigentümer lehnt ab

1. Der Nutzer wird benachrichtigt und das Fahrzeug wird für den Mietzeitraum als frei gespeichert.

**Special Requirements:**

**Technology und Data Variations List:**

**Frequency of Occurrence:** mehrmals pro Stunde

**Open Issues:**

## Use Case UC5: Fahrzeug abholen und zurückbringen

**Primary Actor:** Nutzer

**Stakeholders and Interests:**

* Nutzer: will ein fehlerfreies System, das alle seine Anfragen schnell bearbeitet und in dem die Preise korrekt berechnet werden
* Fahrzeugeigentümer: will, dass alle Daten erfasst und die Preise korrekt errechnet werden
* Unternehmen: will das alle Daten bezüglich der Buchung korrekt gespeichert werden und dass das Fahrzeug danach wieder als frei angegeben wird und dass die Rechnung korrekt erstellt wurde und im System gespeichert wurde
* Bundesamt für Steuern: will Steuern von jeder Rechnung erhalten

**Preconditions:** Der angemeldete Nutzer hat ein Fahrzeug gebucht und die Buchung wurde vom Fahrzeugeigentümer bestätigt. Er befindet sich am Abholort.

**Postconditions:** Das System hat alle Daten bezüglich der Zeiten, des Kilometerstandes und der zusätzlichen Informationen gespeichert. Das Fahrzeug wird wieder als frei angezeigt, sofern nicht im Anschluss gebucht. Der Nutzer und der Fahrzeugeigentümer haben eine Rechnung erhalten.

**Main Success Scenario:**

1. Der Nutzer gibt die Startzeit und den aktuellen Kilometerstand ins System ein. Zusätzlich kann er spezifische Informationen, wie z.B. Tankfüllung oder Schäden ins System eingeben.
2. Das System speichert diese Informationen.
3. Nach der Nutzung gibt der Nutzer Endzeit, Kilometerstand und ggf. weitere Informationen ins System ein.
4. Das System speichert diese Informationen.
5. Das System erstellt eine Rechnung mit Informationen über den Nutzer, den Fahrzeugeigentümer, den Fahrzeugtyp, das konkrete Fahrzeug, die tatsächlich gebuchten Zeiten, die gefahrenen Kilometer und den Zahlungsinformationen.
6. Aus diesen Angaben berechnet das System den Preis und fügt dies ebenfalls der Rechnung hinzu.
7. Das System schickt diese Rechnung an den Nutzer und den Fahrzeugeigentümer.

**Extensions:**

**Special Requirements:**

**Technology und Data Variations List:**

**Frequency of Occurrence:** Bei jeder Buchung, mehrmals täglich

**Open Issues:**

* Was passiert, wenn die Buchungszeit überschritten wurde und ein anderer Nutzer das Fahrzeug für diese Zeit gebucht hatte?
* Was passiert, wenn der Nutzer einen Unfall hatte?
* Welche Auswirkungen in der Rechnung haben Fahrzeugschäden, die durch die Benutzung entstanden sind?

## Use Case UC6: Buchung stornieren

**Primary Actor:** Nutzer

**Stakeholders and Interests:**

* Nutzer: will das die Buchung korrekt storniert wurde
* Fahrzeugeigentümer: will über die Änderung benachrichtigt werden und sein Fahrzeug soll danach als „frei“ gelten
* Unternehmen: will das alle Kunden zufrieden sind, indem alle Prozesse korrekt ablaufen

**Preconditions:** Der Nutzer hat ein Fahrzeug gebucht und die Buchung wurde vom Fahrzeugeigentümer akzeptiert. Die Buchungszeit hat noch nicht begonnen.

**Postconditions:** Die Buchung wurde gelöscht. Das Fahrzeug gilt in dem Buchungszeitraum als nicht gebucht.

**Main Success Scenario:**

1. Der Nutzer sieht eine Übersicht seiner Buchungen.
2. Der Nutzer wählt eine noch nicht eingetretene Buchung aus und wählt Buchung stornieren im System aus.
3. Das System löscht die Buchung mit allen Informationen, setzt das Fahrzeug in dem Buchungszeitraum als nicht gebucht und benachrichtigt den Fahrzeugeigentümer.

**Extensions:**

2a. Der Nutzer wählt eine Buchung zu einer bereits vergangen Zeit aus.

1. Das System weist Nutzer darauf hin, dass diese Buchung bereits geschehen ist. Zurück zu 1.

**Special Requirements:**

**Technology und Data Variations List:**

**Frequency of Occurrence:** bei jeder hunderten Buchung

**Open Issues:**

* Wie viele Stunden vor dem Buchungsbeginn, darf der Nutzer die Buchung noch stornieren?