**Dokumentation Car Rent**

Inhaltsverzeichniss

[1 Einführung 3](#_Toc18162205)

[1.1 Aufgabenstellung 3](#_Toc18162206)

[1.2 Qualitätsziele 3](#_Toc18162207)

[1.3 Stakeholder 3](#_Toc18162208)

[1.3.1 Kunde 3](#_Toc18162209)

[1.3.2 Admin Mitarbeiter 3](#_Toc18162210)

[1.3.3 Administrator der Software 3](#_Toc18162211)

[2 Randbedingungen 4](#_Toc18162212)

[2.1 Technische Randbedingungen 4](#_Toc18162213)

[2.2 Organisatorische Randbedingungen 4](#_Toc18162214)

[2.3 Konventionen 4](#_Toc18162215)

[3 Kontextabgrenzung 5](#_Toc18162216)

[3.1 Fachlicher Kontext 5](#_Toc18162217)

[3.1.1 Auto 5](#_Toc18162218)

[3.1.2 Reservierung 5](#_Toc18162219)

[3.1.3 Buchung 5](#_Toc18162220)

[3.1.4 Mitarbeiter 5](#_Toc18162221)

[4 Lösungsstrategie 6](#_Toc18162222)

[5 Verteilungssicht 7](#_Toc18162223)

[6 Konzepte 8](#_Toc18162224)

[7 Risiken 9](#_Toc18162225)

[8 Glossar 10](#_Toc18162226)

1. Einführung
   1. [Aufgabenstellung](http://www.dokchess.de/dokchess/arc42/1_einfuehrung/36.html)

Der Sachbearbeiter kann Kunden mit Namen und Adresse und Kundennummer im System verwalten, d.h. erfassen, bearbeiten, löschen und den Kunden mit dessen Namen oder Kundennummer suchen.

• Der Sachbearbeiter kann zudem die Autos von CarRent verwalten und nach denen suchen.

• Jedes Auto kann einer bestimmten Klasse zwischen Luxusklasse, Mittelklasse oder Einfachklasse zugeordnet werden und besitzt zudem eine Marke, einen Typ und eine eindeutige Identifikation.

• Jede Klasse besitzt eine Tagesgebühr.

• Bei einer neuen Reservation kann der Kunde ein Auto aus einer bestimmten Klasse wählen. Er muss zudem die Anzahl der Tage angeben, die er das Auto gerne mieten möchte. Dabei werden die Gesamtkosten berechnet. Wird die Reservation gespeichert, so wird sie mit einer Reservationsnummer ablegt.

• Bei Abholung des Autos wird die Reservation in einen Mietvertrag umgewandelt.

* 1. [Qualitätsziele](http://www.dokchess.de/dokchess/arc42/1_einfuehrung/37.html)

Dieses Projekt hat folgende Qualität Ziele.

Es muss eine einfachwartbare Software.

Zudem muss sie immer alle Auto und Reservierungen verwalten können.

* 1. [Stakeholder](http://www.dokchess.de/dokchess/arc42/1_einfuehrung/38.html)

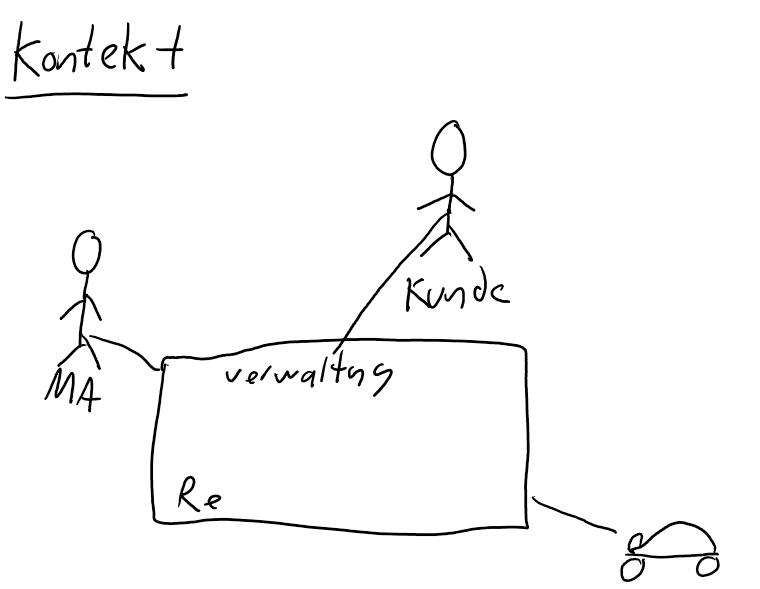
Es gibt Folgende Steakholder.

* + 1. Kunde
    2. Admin Mitarbeiter
    3. Administrator der Software

1. Randbedingungen
   1. Technische Randbedingungen

* Es ist eine Webservice Applikation.
* Die eine WebAPI anbietet.
* Die API kann dann von Browser angesprochen werden. Dieses ist aber nicht teil des Projektes
* Die Programmier Sprache ist C#
* Die Daten werden im Filesystem als cv gespeichet.
* Die gesundetet und empfangenden Daten sind im Json Format.
  1. Organisatorische Randbedingungen
* Es ist ein OpenSource Projekt
* Der Daten Austausche fom Programm Code erfolgt über Git Hub
* Eine erste Version wird bis a 15.09.2019 entwickelt.
* Entwicklungswerkzeuge ist Visual Studio 2017 und 2019 ¨
* Vorgehensmodell: Entwicklung risikogetrieben, iterativ und inkrementell- Zur Dokumentation der Architektur kommt arc42 zum Einsatz. Eine Architekturdokumentation gegliedert nach dieser Vorlage ist zentrales Projektergebnis.
  1. Konventionen
* Es gelten die Normal C# Codier Richtline
* Dokumentation ist auf Deutsch
* Kommentare sind auf Deutsch oder Englisch.

1. Kontextabgrenzung
   1. Fachlicher Kontext



* + 1. Auto

Ein Auto kann von eine über eine Class gebucht werden.

* + 1. Reservierung

Wenn ein Kunde ein Auto Buchen will Reserviert er dieses zu erst.

* + 1. Buchung

Wenn der Kunde eine Auto Buchen will dann wir die Reservierung in eine Buchung umgewandelt.

* + 1. Mitarbeiter

Der Ma Arbeitet mit der Software und erstellt Buchungen und Reservierungen.

1. Lösungsstrategie

Für die Lösung wurden mehrere Diagramm erstellt. Die halfen das System besser zu verstehen.

Dieses Diagramme sind zu finde unter “./ Aufgabe1/”

1. Verteilungssicht

Da es sich um eine WebAblikation handelt gibt es schon eine Aufteilung zwischen Server und Client.

Der Server ist für die Verarbeitung und Speicher der Daten verantwortlich.

Der Client also zum Beispiel ein Browser ist für die Darstellung der Daten verantwortliche .

Der Aufbau des System ist in ein Diagramm Dargestellt. Dieses findet man unter:

. \Aufgabe6\6.1.1 Deployment View.pdf

1. Konzepte

Um die Software zu erstellen wurden mehre Diagramme erstellt die befinde sich unter.

. \Aufgabe6\6.1.2 Logical View Class Diagram.mdj

. \Aufgabe6\6.1.2 Logical View Component Diagram.pdf

1. Risiken

Die folgenden Risiken wurden zu Beginn des Vorhabens identifiziert. Sie beeinflussten die Planung der ersten drei Iterationen maßgeblich. Seit Abschluss der dritten Iteration werden sie beherrscht. Dieser Architekturüberblick zeigt die Risiken inklusive der damaligen Eventualfallplanung weiterhin, wegen ihres großen Einflusses auf die Lösung.

* + 1. Speicherung der der Verknüpfungen.

Da wir am Anfang die Daten in mehrere CSV Datei speichern . Dieses sollte in der Implantation gelöst werden. Ist momentan leider noch nicht so. Dieses ist aber in der Anfangs Phase nicht so kitsch.

* + 1. Zugriffzeiten

Da wir mit einem CSV Speicher arbeiten der nicht sehr schnell ist könnte es bei vielen abfragen zu Performance Problem kommen.

* + 1. Erweiterbarkeit

Weil beim CSV die Reihenfolge der Datensätze wichtig ist ist der Erweiterung nicht ganz einfach.

* + 1. Fazit

Eine CSV Lösung ist nicht das besste und muss bei mehr Nutzung auf eine DB Lösung Ausgetauscht werden. Aber in der Start und Versuchs Phase ist sie Ausreichend.

1. Glossar

Gibt es noch nicht da noch nicht nötig.