Fachkonzept

-Projekt-  
*RentACar Autovermietung*

****

-Projektmitglieder-  
*Erwin Braun, Alexander Steffen und Gerrit Böselager*

-Datum-  
*16. August 2012*

Inhaltsverzeichnis

1 Anforderungsdefinition 2

1.1 Muss-Funktionalitäten 2

1.2 Optionale Funktionalitäten 3

1.3 Abgrenzungskriterien 3

2 Planung und Fachkonzeption 4

2.1 Use-Case Diagramm 4

2.2 Entity Relationship Model (ERM) 5

2.3 Klassendiagramm 6

2.4 Prototypen 7

2.5 Projektplan 10

# Anforderungsdefinition

Als Ziel gilt die Umsetzung einer Software zu Verwaltung einer Autovermietung. In der Software müssen Standardprozesse, wie z.B. Suche nach Verfügbaren Fahrzeugen, Reservierung/Buchung eines Fahrzeugs und Preiskalkulation, abgebildet werden.

Aus technischer Sicht ist darauf zu achten, dass als Middleware Webservices zum Einsatz kommen sollen.

Die Funktionalitäten werden im Folgenden kurz konkret beschrieben.

## Muss-Funktionalitäten

Die Muss-Funktionalitäten dokumentieren die Funktionen, die explizit vom Auftraggeber verlangt werden und unbedingt umzusetzen sind.

* Aufrufbar über einen herkömmlichen Browser
* Autosuche (Verfügbarkeit)
  + Potentielle Kunden können verfügbare Mietwagen zu einem bestimmten Zeitpunkt und Ort anzeigen lassen.
  + Ergebnis ist eine Liste verfügbarer Autos.
* Autodetailansicht
  + Dem Kunden wird ein Fahrzeugsteckbrief mit Detailinformationen gezeigt.
  + Beispielsweise: Anzahl Türen, PS, Farbe, Typ, Modell, Kraftstoff, usw.
* Autoreservierung (bzw. -buchung)
  + Nach Auswahl eines Fahrzeuges kann der Kunde eine Reservierung durchführen.
  + Die Angabe von persönlichen Kundendaten (z.B. Vorname, Nachname, E-Mail, Anschrift) bei der Reservierung ist notwendig.
  + Auf Grundlage der Reservierungsinformationen muss eine Preiskalkulation durchgeführt.
* Plausibilitätsprüfung bei der Eingabe
  + Die angegebene E-Mail-Adresse des Kunden wird auf Plausibilität untersucht.
* Speicherung der Daten in einer Datenbank
  + Die Daten werden in einer MySQL-Datenbank gespeichert.
* Webservices als Middleware
  + Um die Unabhängigkeit von Programmiersprache und Betriebssystem (Interoperabilität) zu gewährleisten, sollen als Middleware Webservices eingesetzt werden.

## Optionale Funktionalitäten

Zusätzlich zu den Muss-Funktionalitäten können die folgend gelisteten Features optional umgesetzt werden.

* Filtermöglichkeit
  + Der Kunde hat die Möglichkeit die Ergebnisliste der verfügbaren Fahrzeuge nach Automarke, Typ, Modell zu filtern.
* Autostandort anzeigen lassen
  + Der Standort des Fahrzeugs kann in einer Maps-Ansicht angezeigt werden.
* Login für Bestandskunden
  + Kunden, die bereits in Vergangenheit Fahrzeuge angemietet haben, können sich mit ihren Logindaten (E-Mail; Passwort) am System anmelden und haben so Einblick in aktuelle Reservierungsdetails.

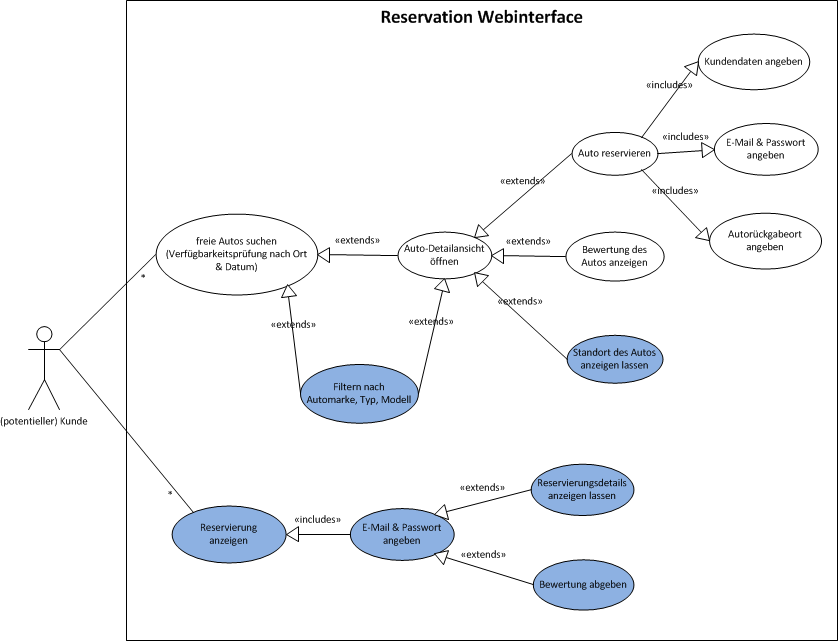
## Abgrenzungskriterien

Folgende Abgrenzungskriterien wurden für das Projekt definiert.

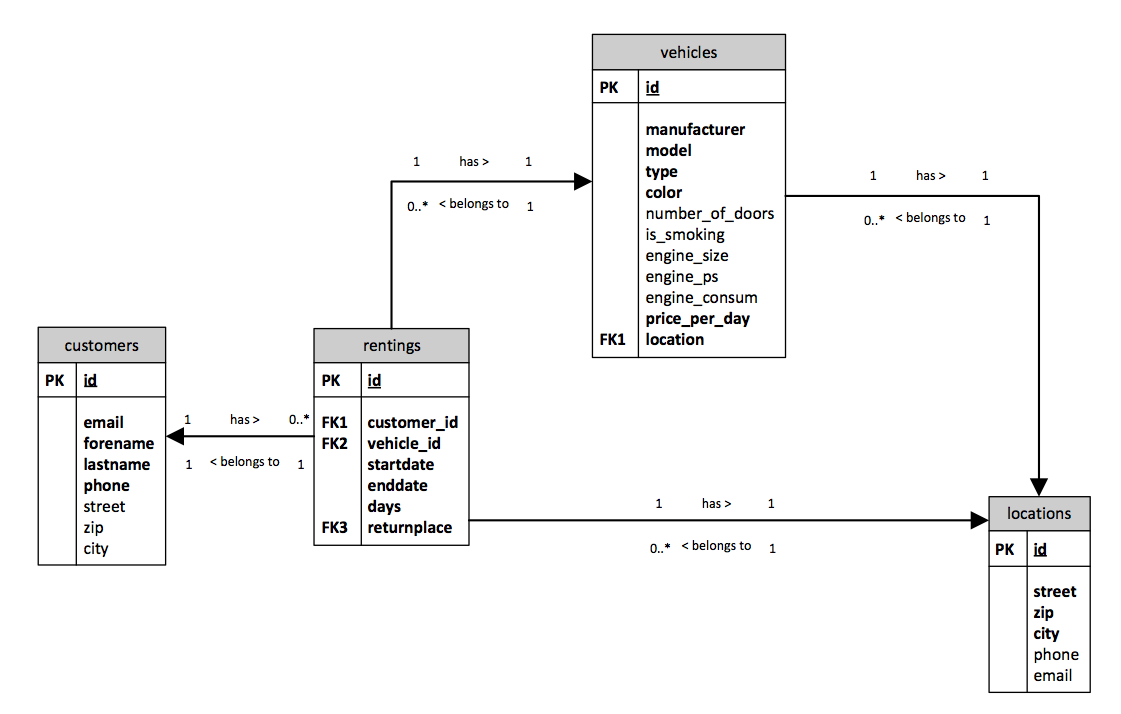
* Ein Backend zur Pflege von Stammdaten (Fahrzeuge, Standorte und Kunden) ist nicht notwendig.
* Schnittstellen zu externen Systemen (beispielsweise zu ERP oder CRM-Systemen) sind nicht vorgesehen.
* Die Datenübertragung zwischen Server und Client geschieht auf unverschlüsseltem Wege.
* Passwörter werden unverschlüsselt in der Datenbank abgespeichert.
* Mehrsprachigkeit ist nicht vorgesehen.

# Planung und Fachkonzeption

## Use-Case Diagramm



## Entity Relationship Model (ERM)

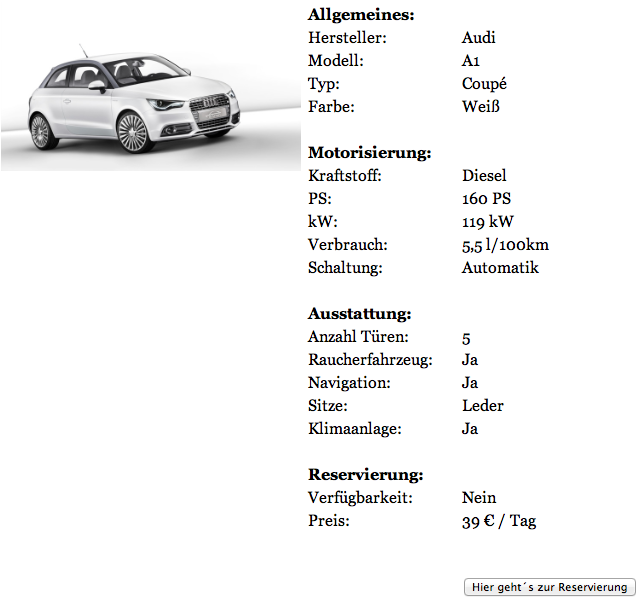


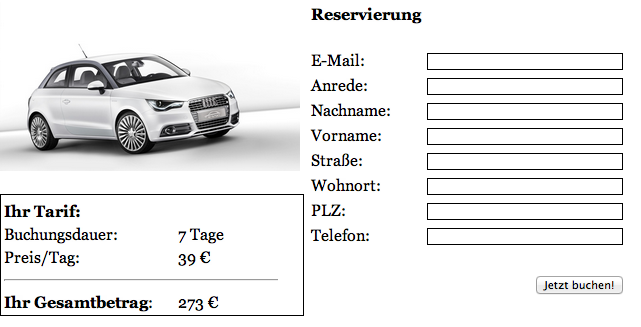
## Klassendiagramm

## Disk1:Users:gboeselager:Develop:Github:RentACar:Projektdokumente:Klassendiagramm.png

## Prototypen

**Startseite (Suchmaske)****Fahrzeugliste der (verfügbaren) Fahrzeuge**

**Detailansicht für ein Fahrzeug**

**Reservierung eines Fahrzeuges**

**Bestätigungsseite für die Reservierung**

## Projektplan

