



**Hochschule  
Bonn-Rhein-Sieg**  
University of Applied Sciences

**Praktische Hausarbeit zur Vorlesung**

**Software Engineering 2 (BWI)**

**Prof. Dr. Sascha Alda**

**SS 2020**

Kevin Stephan Pierzchala  
9019483

---

Für diese Hausarbeit wurde eine Software entwickelt (zutreffendes ankreuzen):

☒ JA

☐ NEIN

Falls eine Software für das Semesterprojekt entwickelt wurde (= JA): Der Source Code der Software wurde dokumentiert als (zutreffendes ankreuzen bzw. angeben):

☒ als ZIP-Datei auf LEA

☐ in einem Git-Repository unter dem Link:

---

## Inhaltsverzeichnis

1	Einleitung.....	4
	1.1 Problemstellung.....	4
	1.2 Zielsetzung.....	4
2	Analyse (Anforderungs-Artefakte).....	4
	2.1 Beschreibung des Scopes.....	4
	2.2 Use Case Modell .....	4
	2.3 OOA.....	5
	2.4 Stakeholder-Analyse .....	5
	2.5 Stakeholder Portfolio .....	6
	2.6 Qualitätsanforderungen Usability.....	6
	2.7 Utility Tree .....	7
3	Konzeption (Software-Architektur).....	8
	3.1 Kontext-Sicht.....	8
	3.2 Baustein-Sicht .....	9
	3.3 ER-Modell.....	10
	3.4 ATAM – Methode.....	11
	3.5 Persistenz.....	11
4	Implementierung.....	12
	4.1 Wichtige Implikationen für die Implementierung .....	12
	4.2 Berechnung der Metriken .....	12
	4.3 Umsetzung der Qualitätsanforderung Usability .....	13
	4.4 Snapshot der Anwendung .....	14
	4.5 Manuelle Akzeptanztests .....	17
	4.6 SQL-Schema der Datenbank .....	17
5	Zusammenfassung.....	18
	5.1 Fazit.....	18
	5.2 Was wurde nicht erreicht?.....	18
	5.3 Ausblick.. ..	18
6	Quellenverzeichnis.....	19

## 1. Einleitung

### 1.1 Problemstellung

Das Unternehmen CarLook Ltd möchte eine Suchplattform haben, mit der registrierte Endkunden nach Autos aus dem Fuhrpark des Unternehmens suchen können. Dies soll auf Grundlage des Frameworks Vaadin erfolgen.

### 1.2 Zielsetzung

Es wird eine prototypische Lösung implementiert, die Login, Registrierung, Suche und die Buchung von Autos erlaubt. Mithilfe einer Datenbank lassen sich auch Daten Persistenz abspeichern.

## 2. Analyse (Anforderungs-Artefakte)

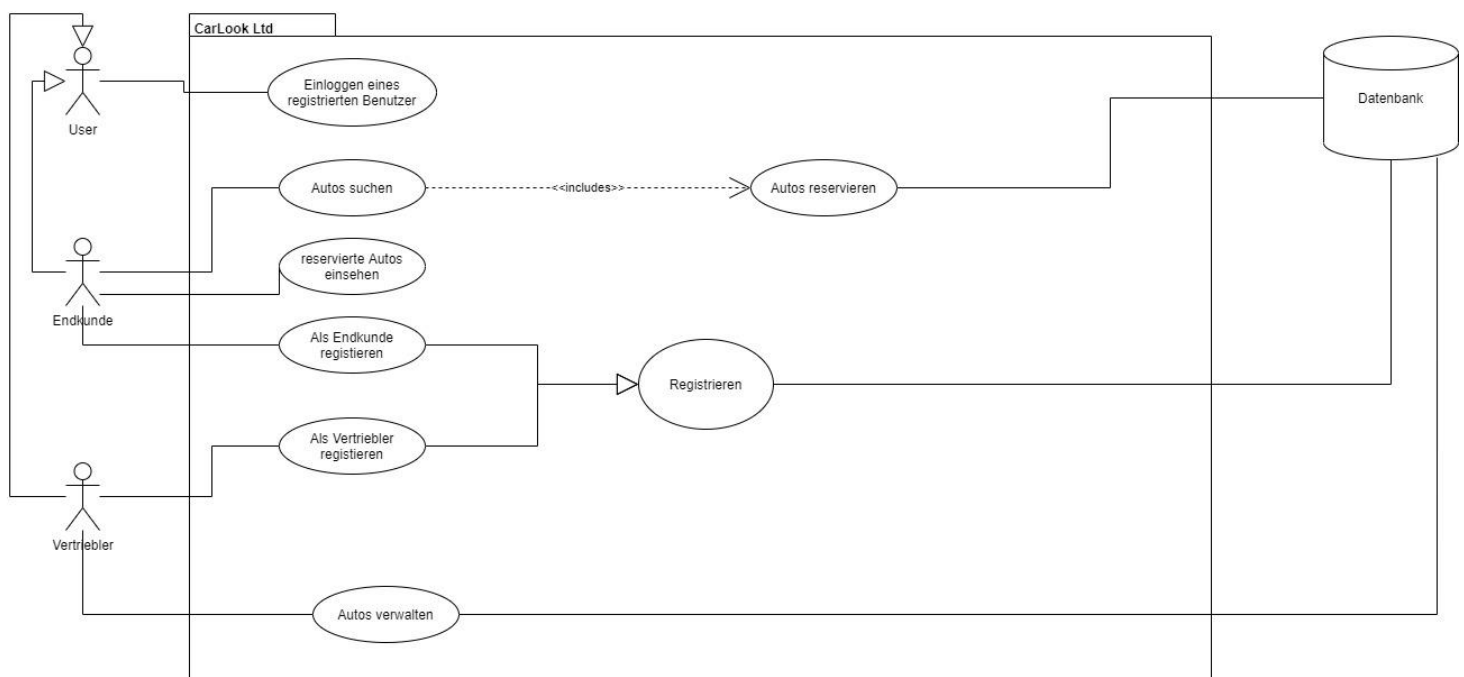
### 2.1 Beschreibung des Scopes

Der Umfang dieses Projekt ist es eine Software mithilfe des Vaadin Frameworks zu implementieren. Der Prototyp soll einen Login sowie eine Registrierung und eine Suche anbieten. Zudem sollen auch gewisse Heuristiken der Usability eingehalten werden und eine Datenbank Verbindung aufgebaut werden.

Zusätzlich sollen auch Software-Architektur und diverse Artefakte erstellt werden, die zur Dokumentation des Projekts dienen.

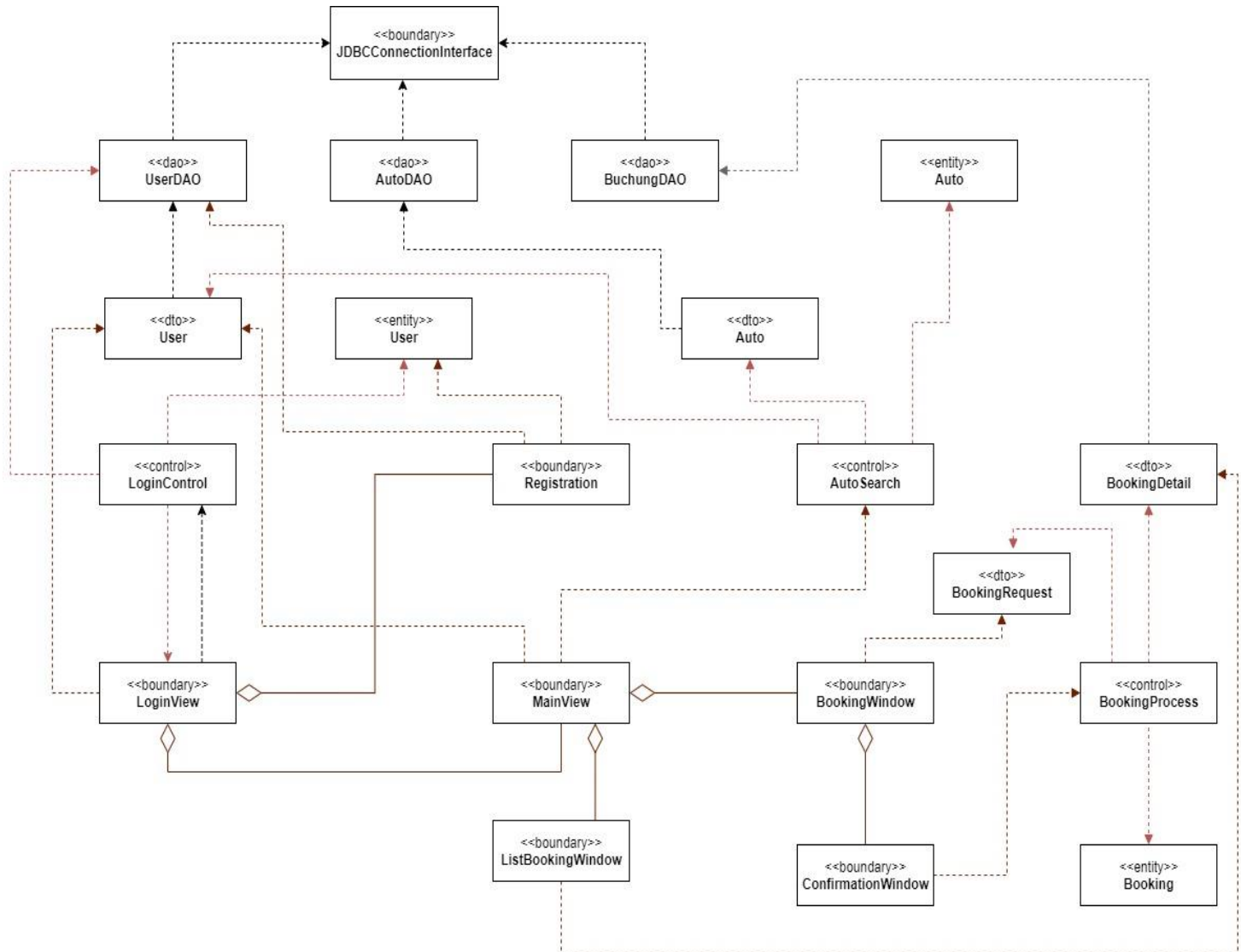
### 2.2 Use Case Modell

Dieses Use Case Modell für die Firma CarLook beschreibt die Anforderungen an den Prototypen, die umgesetzt werden können.



## 2.3 OOA

Das objektorientiertes Analyse-Modell dient zur Veranschaulichung der Software Architektur und deren Abhängigkeit zueinander.



## 2.4 Stakeholder-Analyse

Die folgende Tabelle erläutert die Einstellung und Auswirkung auf das Projekt des jeweiligen Stakeholders.

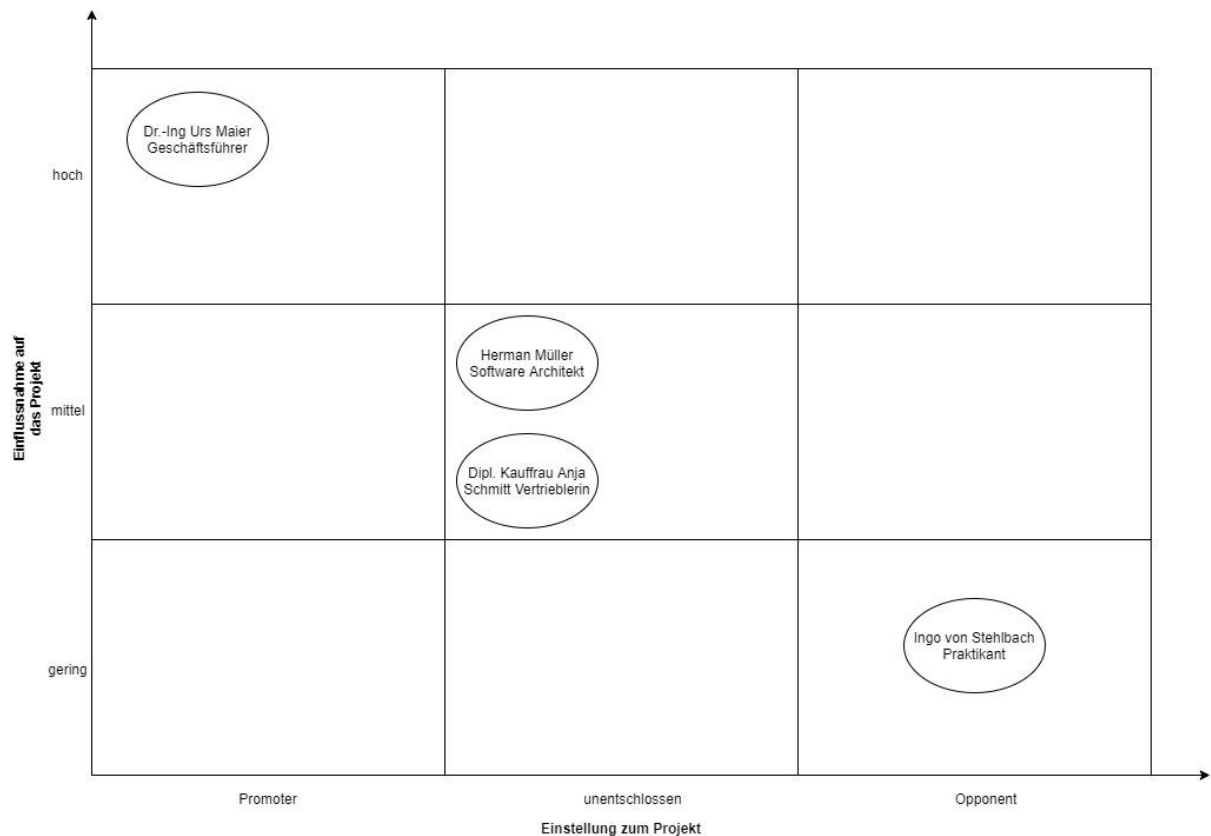
Eine Gesamtbeurteilung wurde für den Geschäftsführer sowie den Praktikanten durchgeführt.

Stakeholder						
Nr.	Person	Einstellung zum Projekt	Einfluss	Stakeholder Gruppe	Innovationsbereitschaft	Beurteilung
1	Urs Maier Geschäftsführer	gut = 2	3	Promoter	3	2+3+3 = 8
2	Anja Schmitt Vertrieblerin	neutral = 1	2	unentschlossen	2	
3	Herman Müller Software Architekt	neutral = 1	2	unentschlossen	1	
4	Ingo von Stehlbach Praktikant	schlecht = 0	1	Opponent	1	0 + 1 + 1 = 2

## 2.5 Stakeholder Portfolio

Das Stakeholder Portfolio für die vier Stakeholder dient zur Veranschaulichung der Einstellung zum Projekt.

Es stellt sich heraus, dass es einen Promoter, zwei unentschlossene und einen Opponent gibt.



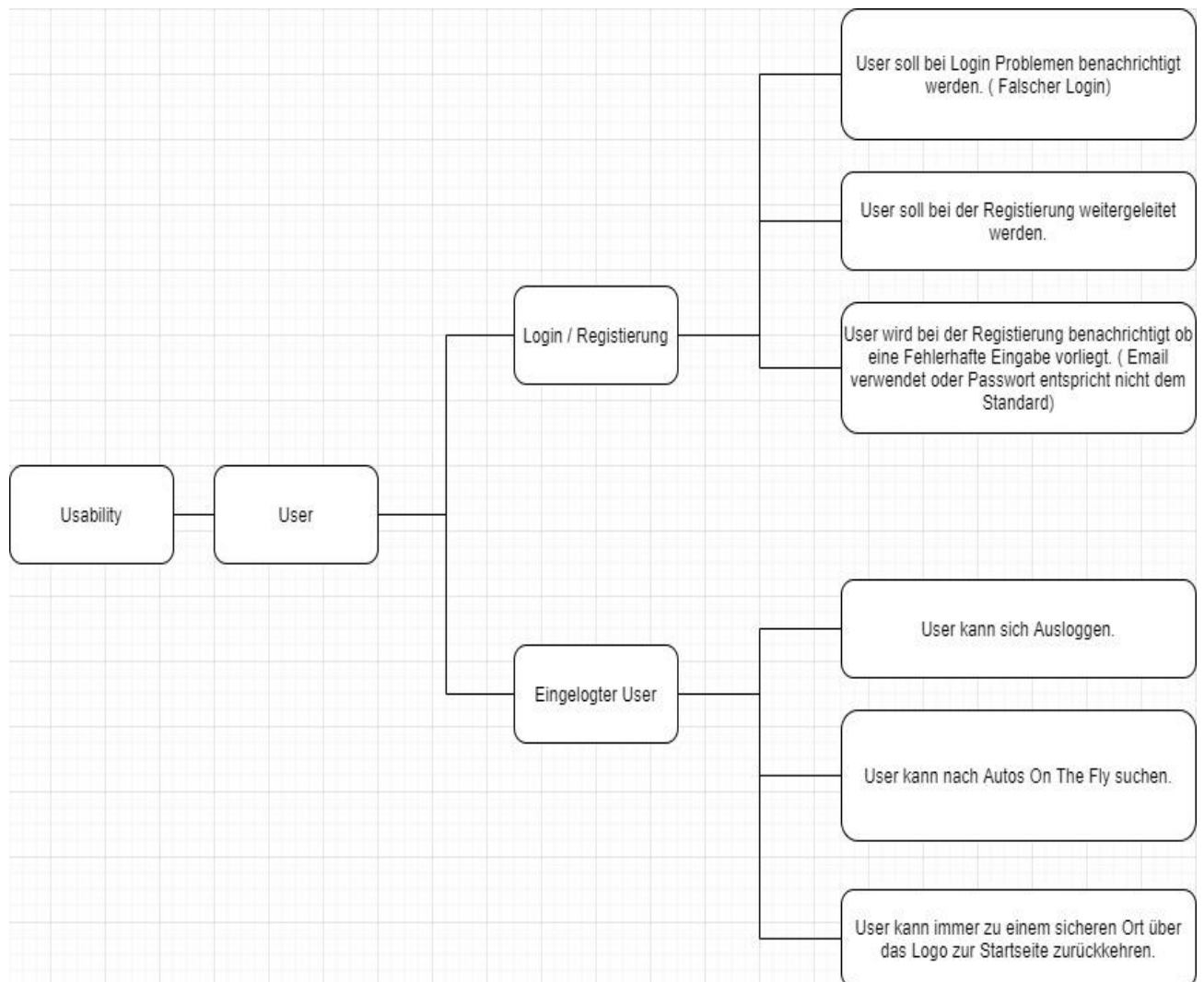
## 2.6 Qualitätsanforderung Usability

Es wurden folgende Standards der Usability beachtet:

- Verwendung von Tabellen für das Anzeigen von Suchergebnissen
- Strukturelle Wiederholung durch eine Navigationbar
- Ein Logo mit Startseite-funktion um an einen sicheren Ort zu gelangen
- Unterstützung bei Benutzerfehlern wie zum Beispiel falsche Login Daten
- Validierung von Benutzer Eingaben
  - Email check ob bereits in Verwendung
  - Passwort Korrektheit und Übereinstimmung checken

## 2.7 Utility Tree

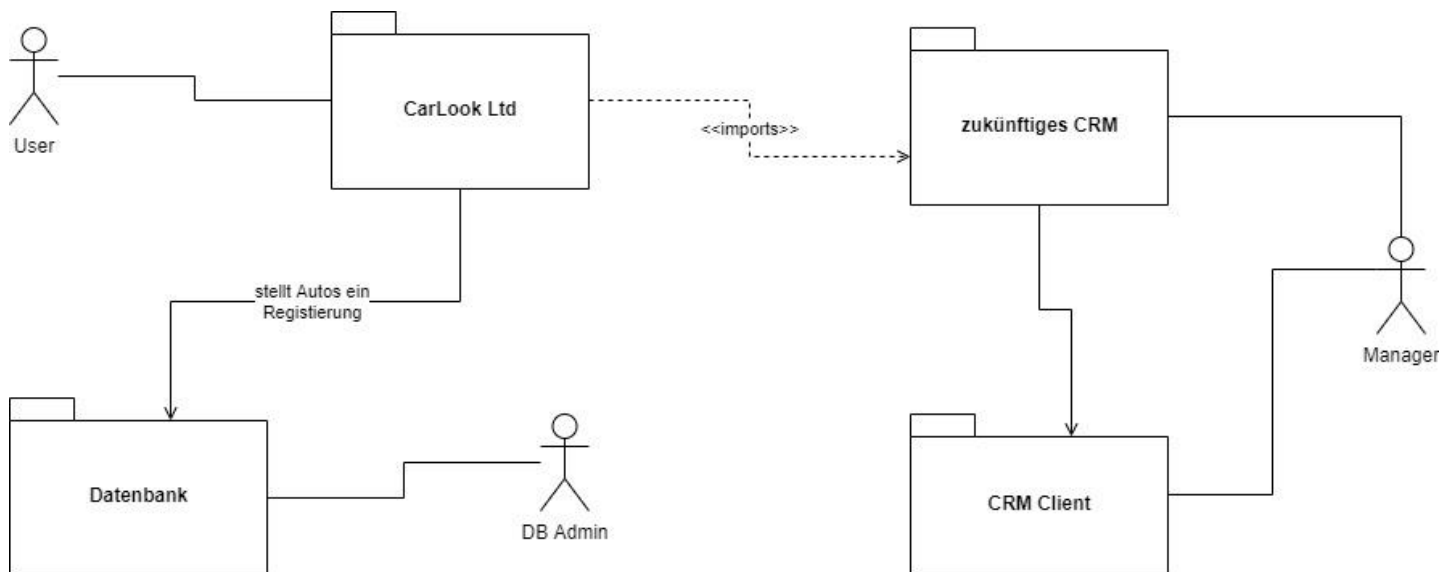
In diesem Diagramm werden die Refinements der Usability dargestellt, um Architekturentscheidungen zu erklären. Es wird unterschieden ob der User eingeloggt ist oder nicht. Ein nicht eingeloggter User kann sich einloggen oder Registrieren. Im eingeloggten Zustand kann der User nach Autos suchen oder Ausloggen. Falls der User zu einen Sicheren Ort zurückkehren möchte, hat er die Möglichkeit die Startseite Funktion zu benutzen.



### 3. Konzeption (Software-Architektur)

In diesem Abschnitt werden die Modelle für die Software-Architektur dargestellt.

#### 3.1 Kontext-Sicht

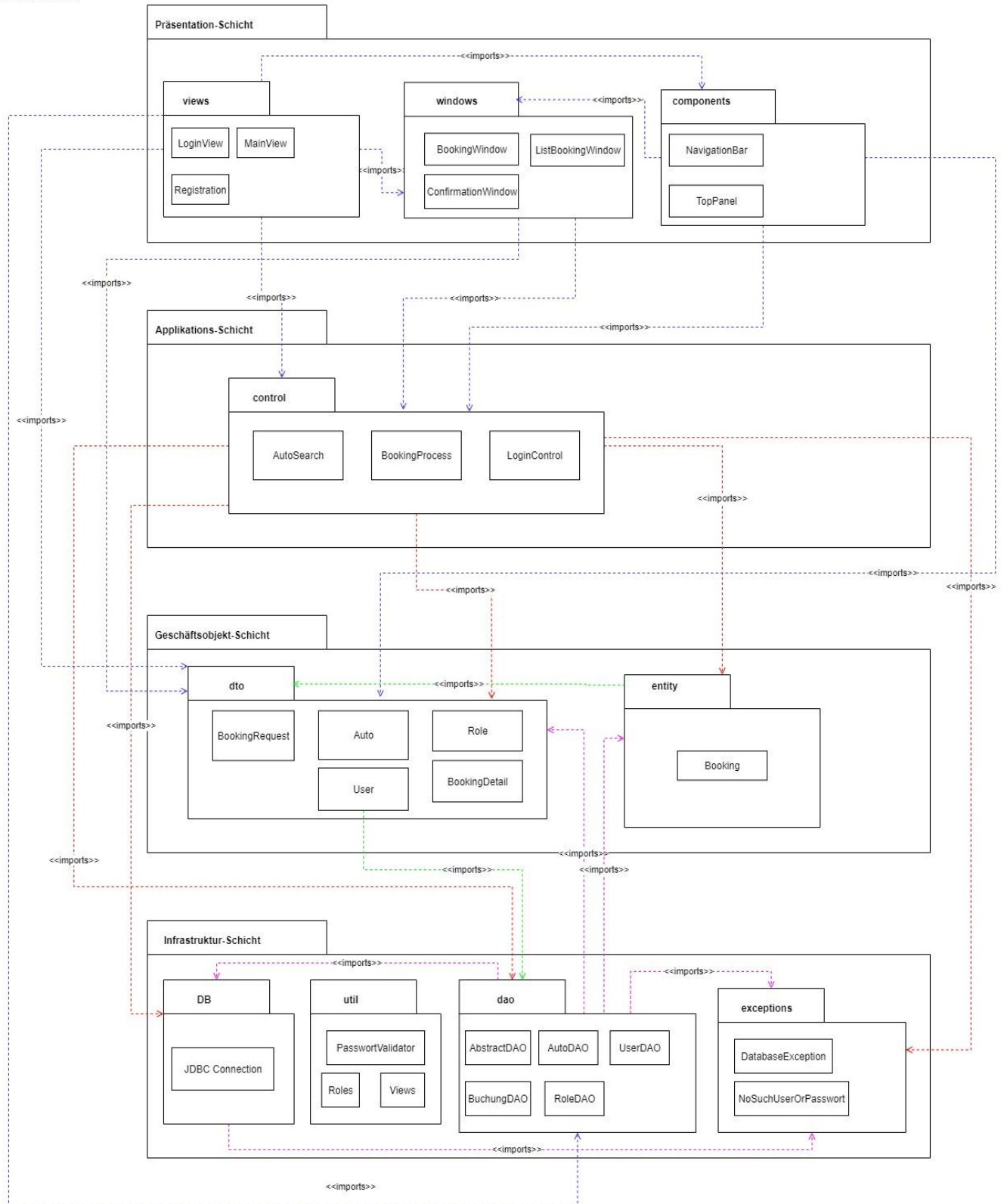




### 3.2 Baustein-Sicht

Die Baustein-Sicht unterteilt sich in vier Pakete, die in eine Präsentations-Schicht, Applikations-Schicht, Geschäftsobjekt-Schicht und eine Infrastruktur-Schicht aufgeteilt ist. Sie zeigt die notwendigen Imports für die Software Architektur.

Baustein Sicht von CarLook

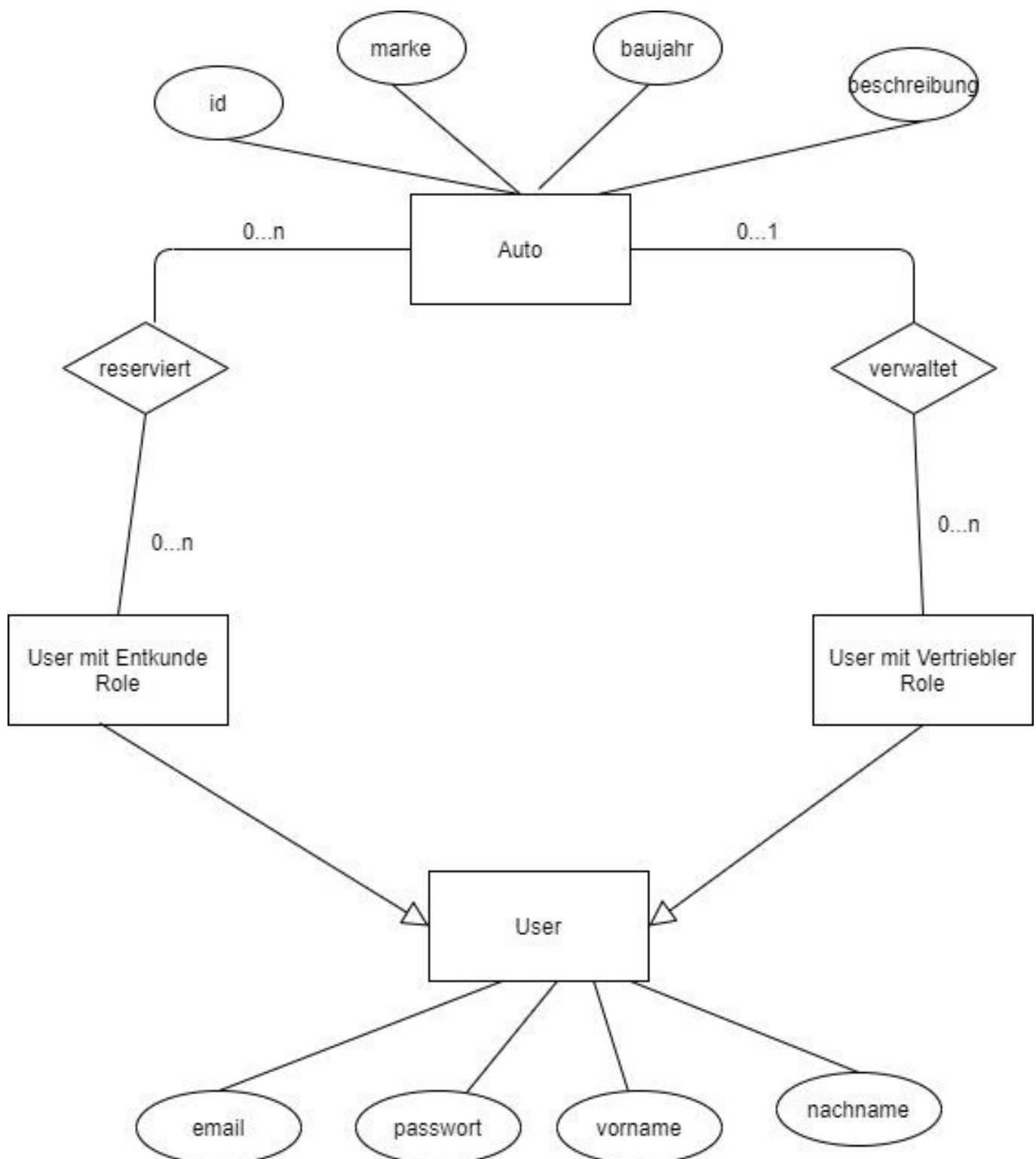


### 3.3 ER-Modell

Im folgenden ER-Modell werden die Relationen der Entities dargestellt.

Ein Endkunde User kann mehrere Autos oder keine Reservieren und ein Auto kann mehrere oder keinen User zugewiesen sein.

Der Vertriebler User kann mehrere Autos einstellen oder auch nicht. Zudem kann aber nur ein Auto immer nur zu einem Vertriebler gehören.



### 3.4 ATAM – Methode

Entscheidung für eine On-The-Fly suche:

- Es dient zur asynchronen Kommunikation zwischen dem Client und dem Server um, eine Chatty Application darzustellen.

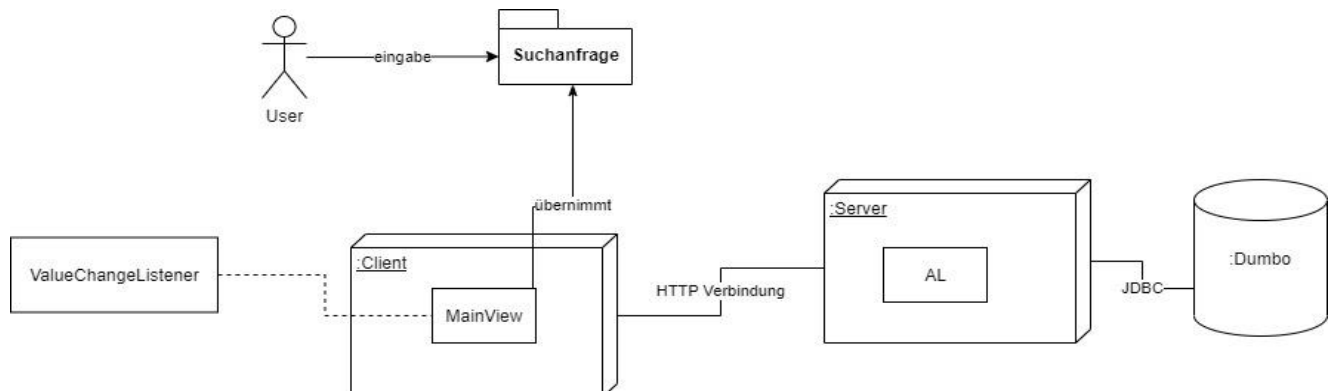
Risiken der Implementierung:

- Aufwendige Implementierung
- Komplexe Exceptions
- Jede Eingabe startet eine neue Suchanfrage (hohe Auslastung)

Voraussetzungen einer Implementierung:

- Verwendung eines Frameworks, das einen Valuechangelistener bereitstellt.
- Verwendung einer Datenbank die auch große Mengen an Anfragen verarbeiten kann.

Architektur Diagramm:



### 3.5 Persistenz (DAO)

Zur Persistenz wurde eine Datenbank mit einer JDBC Verbindung verbunden, die dadurch verwendet wird. Sie dient zur Dauerhaften Abspeicherung von Usern für den Login sowie die Registrierung. Der Vertriebler kann auch manuell Autos in die Datenbank einfügen und über die Suche vom Prototyp die Daten aufrufen.

## 4. Implementierung

### 4.1 Wichtige Implikationen für die Implementierung

Eine VPN-Verbindung wird benötigt, um Zugriff auf die Datenbank zu erlangen.

Die Datenbank Logindaten sind: ID: kpierz2s PW: kpierz2s

Die folgende Dependency dient zur Einbindung von postgresql, um eine Datenbank anzubinden.[1]

Zudem werden noch einige Änderungen der POM File benötigt und die Implementierung des Vaadin Frameworks 8.[2]

```
<dependencies>
  <dependency>
    <groupId>postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>9.1-901-1.jdbc4</version>
  </dependency>
```

### 4.2 Berechnung der Metriken

- Weighted Methods per Class (WMC)

WMC LoginView = ck(enter) + ck(setup)

ck (enter) = 1 (Einstiegspunkt) + 1 (if-Statement) + 1 (try-catch) = 3

ck (setup) = 1 (Einstiegspunkt) = 1

WMC(LoginView) = 3+1 = 4

- Depth of Inheritance (DIT)

DIT = 6

- Number of Children of a class (NOC)

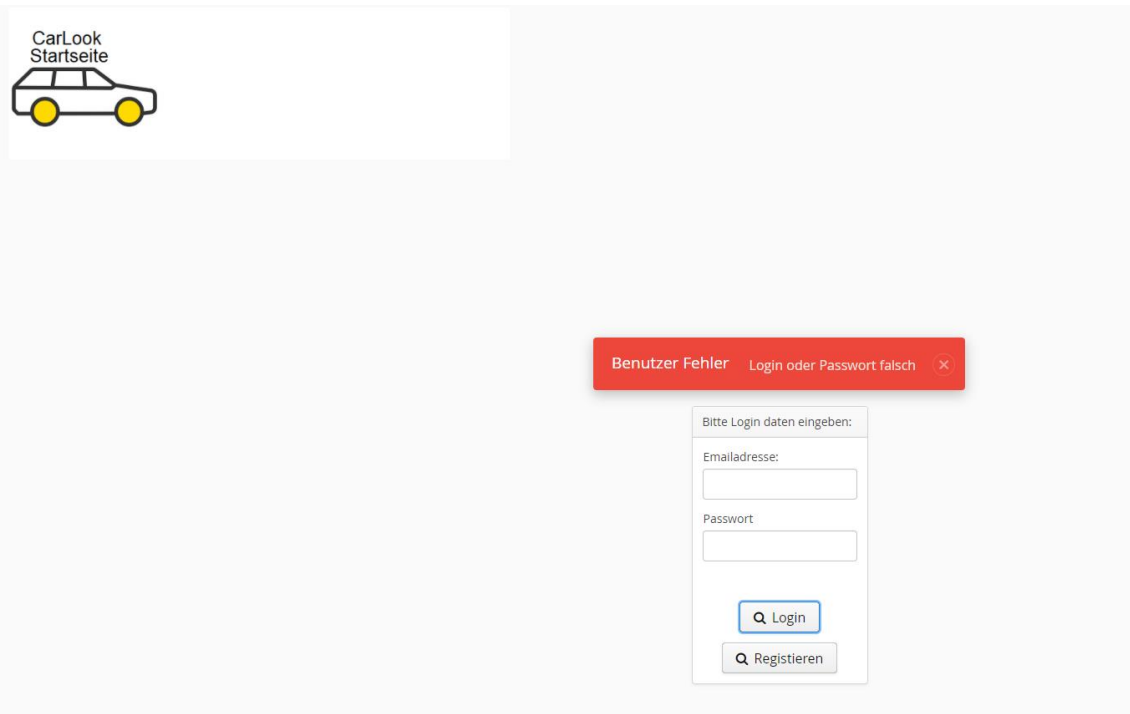
Von LoginView gibt es keine weiteren SubTypen.

NOC = 0

#### 4.3 Umsetzung der Qualitätsanforderung Usability

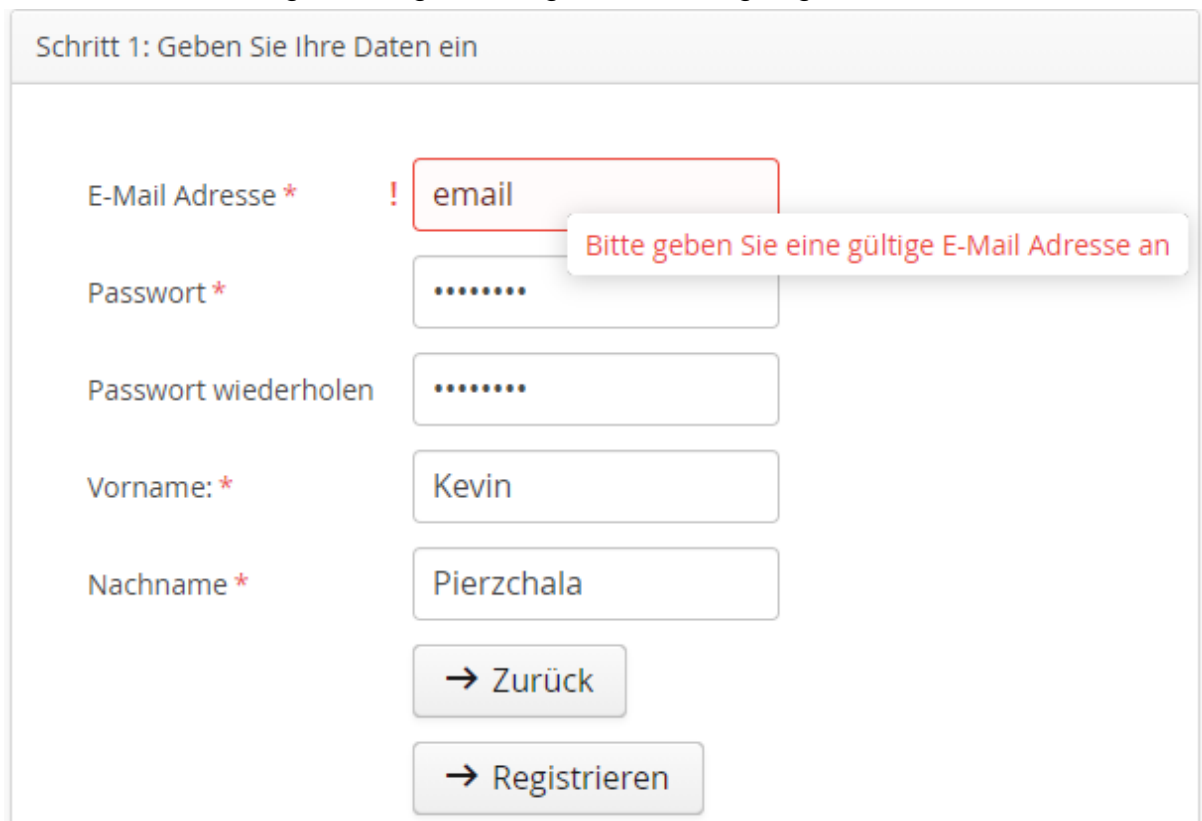
In diesem Bild wird ein Falscher Login demonstriert.

Der Benutzer bekommt eine Fehlermeldung, wenn der Login nicht existiert oder fehlerhaft ist.



The screenshot shows the 'CarLook Startseite' logo in the top left corner. On the right side, there is a red error message box that reads 'Benutzer Fehler Login oder Passwort falsch' with a close button. Below this, a login form titled 'Bitte Login daten eingeben:' contains fields for 'Emailadresse:' and 'Passwort'. At the bottom of the form are two buttons: 'Login' and 'Registrieren'.

Im nächsten Bild folgt die Registrierung mit einer ungültigen Email-Adresse.



The screenshot shows a registration form titled 'Schritt 1: Geben Sie Ihre Daten ein'. The form includes fields for 'E-Mail Adresse \*', 'Passwort \*', 'Passwort wiederholen', 'Vorname: \*', and 'Nachname \*'. The 'E-Mail Adresse' field contains the text 'email' and is highlighted with a red border. A red error message box is displayed over the field, stating 'Bitte geben Sie eine gültige E-Mail Adresse an'. At the bottom of the form, there are two buttons: '→ Zurück' and '→ Registrieren'.

Im folgenden Bild wurde auch ein nicht übereinstimmendes Passwort überprüft. Die Passwörter müssen auch mindestens 8 Zeichen lang sein, Zahlen enthalten und mindestens einen Großbuchstaben. Zum Beispiel: Test1234

Schritt 1: Geben Sie Ihre Daten ein

E-Mail Adresse \*

Passwort \*

Passwort wiederholen

Vorname: \*

Nachname \*


Die Passwörter stimmen nicht überein. ✕

#### 4.4 Snapshots der Anwendung

Die Login View des Prototyps:

Der Benutzer kann oben Links immer wieder zur Login View zurückkehren oder sich mit einem bestehenden User einloggen. Ansonsten kann er einen User Registrieren.

CarLook  
Startseite



Bitte Login daten eingeben:

Emailadresse:

Passwort

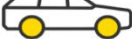
### Die Main View des Prototyps:

In der Main View kann der User sich oben rechts Ausloggen oder seine Buchungen einsehen.

Zudem dient das Logo nun als Startseite für die Main View, kann also so eine Sucheingaben resettet.

CarLook - das Autosuchsystem Willkommen: Kevin! [Menu](#)

CarLook  
Startseite



Suchfenster

Gebe eine AutoMarke ein:  [Suche](#)

Marke	ID	Baujahr	Beschreibung
BMW	1	2003	M3, Sportausstattung, Unfallfrei
BMW	2	2010	M5, Lederausstattung
BMW	3	1999	E36, Unfallauto
VW	4	2005	VW Golf, Unfallauto
VW	5	2007	VW Eos, Cabrio
PORSCHE	6	2017	911, Unfallfrei, 10000 KM
MERCEDES	7	2018	AMG, 1.Hand

[Buchungen](#)

### Die On-The-Fly Suche:

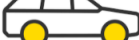
In der Main View kann der User auch On-The-Fly nach Autos suchen.

Sobald ein Buchstabe eingegeben wird, löst der Valuechangelistener eine Suchanfrage aus.

Alternativ kann man auch die Suchanfrage ausschreiben, wie zum Beispiel „BMW“ und dann den Suche Button anklicken oder die Enter Taste drücken.

CarLook - das Autosuchsystem Willkommen: Kevin! [Menu](#)

CarLook  
Startseite



Suchfenster

Gebe eine AutoMarke ein:  [Suche](#)

Marke	ID	Baujahr	Beschreibung
BMW	1	2003	M3, Sportausstattung, Unfallfrei
BMW	2	2010	M5, Lederausstattung
BMW	3	1999	E36, Unfallauto


[Buchungen](#)

## Reservierung von Autos:

Wenn ein User ein Auto reservieren möchte, dann kann er in der Tabelle ein Auto auswählen und dann den Buchen Button anklicken.

Es öffnet sich dann ein neues Fenster mit den ganzen Details, die nochmals mit den Buche Button bestätigt werden muss.

CarLook - das Autosuchsystem Willkommen: Kevin! [Menu](#)

CarLook Startseite 

Suchfenster

Gebe eine AutoMarke ein:  [Suche](#)

Marke	ID	Baujahr	Beschreibung
BMW	1	2003	M3, Sportausstattung, Unfallfrei
BMW	2	2010	M5, Lederausstattung
BMW	3	1999	E36, Unfallauto

[Buchen](#)

Buchung

Buchung für Auto: ID 1 Marke : BMW Baujahr: 2003 Beschreibung: M3, Sportausstattung, Unfallfrei


[Buche](#)

## Und die Einsicht der Reservierungen:

Um die Reservierungen einzusehen muss der User oben rechts das Drop-Down Menu anklicken und auf Buchungen gehen.

Es öffnet sich ein Fenster mit allen Reservierungen für diesen User.

CarLook - das Autosuchsystem Willkommen: Kevin! [Menu](#)

CarLook Startseite 

Suchfenster

Gebe eine AutoMarke ein:  [Suche](#)

Marke	ID	Baujahr	Beschreibung
BMW	1	2003	M3, Sportausstattung, Unfallfrei
BMW	2	2010	M5, Lederausstattung
BMW	3	1999	E36, Unfallauto

[Buchen](#)

Liste aller Buchungen

User	Auto ID	Booking ID
test@web.de	1	1
test@web.de	2	2
test@web.de	2	3
test@web.de	4	4
test@web.de	1	11

[Logout](#) [Buchungen](#)



#### 4.5 Manuelle Akzeptanztests

Für den Login wurde ein schriftlicher Akzeptanztest durchgeführt, um dem Tester einen Login zu ermöglichen, ohne eine Registrierung eines neuen Users. Es wurden positive und negative Äquivalenzklassen angegeben.

Test Case für das System:		CarLook
Use Case (oder User Story):		Login
Tester:		Kevin Pierzchala
Datum der Ausführung:		09.09.2020
Äquivalenzklasse:		pos_Ä1 (Login mit gültigen Eingaben)
Kategorie:		Positivtest

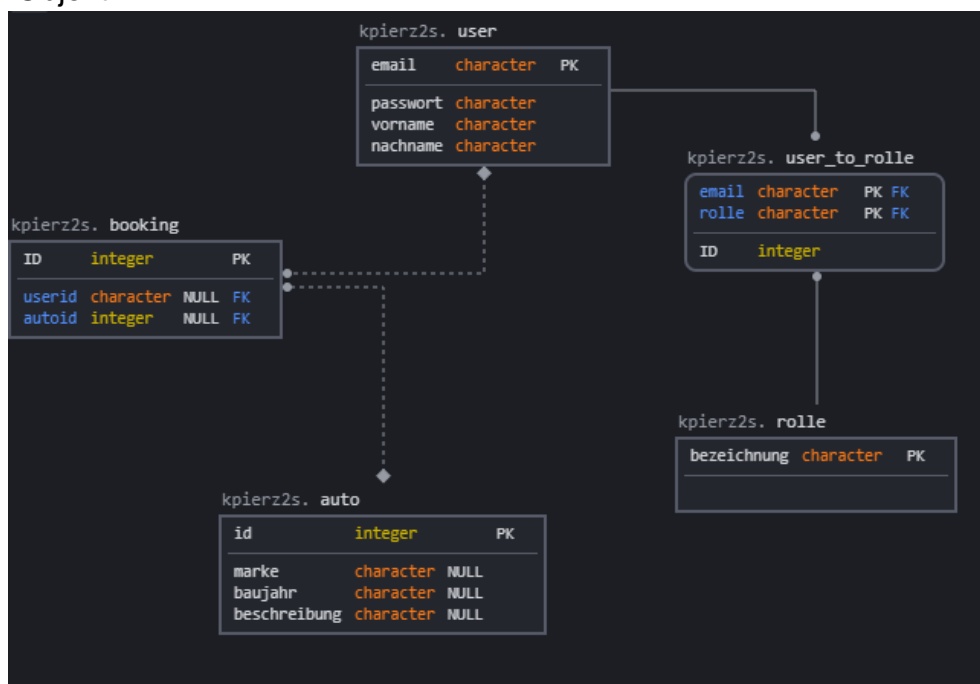
Testprozedur:					
Testschritt-Nr.	Testschritt (mit Input)	Soll-Ergebnis (Was soll erscheinen?)	Ist-Ergebnis (Was ist beim Test tatsächlich erschienen?)	Status (ok / false)	Anmerkungen
1	Startseite mit der URL http://localhost:8080/CarLook#!login aufrufen	Startseite der Anwendung CarLook erscheint.		ok	
2	Login durchführen (LoginID: email@test.de ; Passwort: Test1234) und mit Enter oder dem Login Button bestätigen.	Personalisierte Seite erscheint mit einer Suchleiste sowie einer MenuBar oben rechts zum Ausloggen.		ok	
Gesamtergebnis:				ok	

Zugehörige Äquivalenzklassen:

Parameter	Äquivalenzklasse	Repräsentant
Login ID	pos_Ä1 (Login mit gültigen Eingaben)	email@test.de
Passwort	pos_Ä1 (Login mit gültigen Eingaben)	Test1234
Login ID	neg_Ä2 (Login mit ungültigen Eingaben)	User
Passwort	neg_Ä2 (Login mit ungültigen Eingaben)	testtest

#### 4.6 SQL-Schema der Datenbank

Die Datenbank besteht aus 5 Objekten, wovon das Rolle Objekt noch nicht benötigt wird. Es dient zur Unterscheidung bei einer Zukünftigen Einbindung des Vertrieblers im User Objekt.



---

## 5. Zusammenfassung (max. 1 Seite)

### 5.1 Fazit

In dieser Hausarbeit wurden alle Artefakte bearbeitet und ein Prototyp erstellt, der alle Anforderungen implementiert. Darunter gehört der Login, die Registrierung, die Startseiten Heuristik, die Überprüfung von Login Informationen, die On-The-Fly Suche, die Reservierung von Autos, die Einsicht von Reservierten Autos. Eine Anbindung an eine Datenbank wurde auch umgesetzt, um die Persistenz zu erlauben.

### 5.2 Was wurde nicht erreicht?

Die Verwendung eines Git-Repository und die Einbindung eines Automatisierten Selenium Test.

### 5.3 Ausblick

Der gelieferte Prototyp kann in Zukunft noch um, die Vertriebler Sicht erweitert werden und die Anbindung an ein CRM-System.

---

## Quellenverzeichnis

[1] postgresql URL: <https://mvnrepository.com/artifact/org.postgresql/postgresql> (Last Access 15.09.2020)

[2] Vaadin URL: <https://mvnrepository.com/artifact/com.vaadin/vaadin-server> (Last Access 15.09.2020)