



Technische Universität Berlin

Software and Embedded Systems Engineering Group

Prof. Dr. Sabine Glesner

www.sese.tu-berlin.de

Sekr. TEL 12-4

Ernst-Reuter-Platz 7

10587 Berlin



Softwaretechnik und Programmierparadigmen WiSe 2023/2024

Prof. Dr. Sabine Glesner
Simon Schwan
Julian Klein



Übungsblatt 12

Eine Autowerkstatt möchte die Abfertigung ihrer Aufträge komfortabel mit einer Software verwalten. Dazu können Mitarbeiter:innen (Employee) im System Kunden und Kundinnen (Customer) anlegen und ihnen Fahrzeuge zuordnen. Für neue Kunden und Kundinnen werden jeweils ein Name, eine Telefonnummer und eine Rechnungsadresse gespeichert und die Fahrzeuge werden mit Kennzeichen und Typ registriert.




Ein Auftrag kann entweder eine Inspektion, ein Reifenwechsel oder eine Reparatur sein. Einem neuen Auftrag wird ein Preis, ein Fahrzeug und automatisch ein Datumstempel zugewiesen. Eine Reparatur erhält außerdem eine genaue Tätigkeitsbeschreibung. Ein Auftrag kann von Mitarbeitenden als beendet markiert werden. In diesem Fall wird der Kunde/die Kundin automatisch vom System benachrichtigt. Außerdem wird für den Auftrag vermerkt, welche/r Mitarbeiter:in ihn beendet hat.

Um Missbrauch vorzubeugen, müssen sich Mitarbeitende am Browser mit ID und Passwort sicher anmelden. Ein/e Administrator:in kann Mitarbeitende anlegen und entfernen.





Schlüssel:

-  Ein ergänzendes Video wird zur Vor- oder Nachbereitung veröffentlicht.
-  Wird im Tutorium besprochen.

1. Use-Case-Diagramm

- a) Erstellt aus den Anforderungen ein Use-Case-Diagramm, in dem beteiligte Akteure/Aktuerinnen und Anwendungsfälle enthalten sind. 
- b) Überlegt, ob Beziehungen zwischen Use-Cases die Beschreibung der Anforderungen sinnvoll erweitern würden.  

2. Klassendiagramme

- a) Erstellt aus dem Text ein Klassendiagramm, das alle Informationen aus dem Anforderungstext enthält, sofern sie abbildbar sind. 
- b) An welchen Stellen hattet ihr verschiedene Möglichkeiten zur Modellierung des Klassendiagramms? Diskutiert alternative Entwurfsentscheidungen.  
- c) Überlegt, wie Stereotype nach dem *entity-boundary-control* Pattern auf unser Beispiel angewendet werden könnten. 

3. Implementierung

Diskutiert wie eine mögliche Implementierung eures Klassenmodells in Java aussehen könnte. 