

## Softwaretechnik und Programmierparadigmen WiSe 2014/2015

Prof. Dr. Sabine Glesner

Dr. Thomas Göthel

Lydia Mattick

Tutoren

thomas.goethel@tu-berlin.de

lydia.mattick@tu-berlin.de

### Übungsblatt 3

Ausgabe: 30.10. (Besprechung: 3.11. und 4.11.)

#### 1. Requirements Engineering - Use Cases

In der letzten Übung wurde ein erweiterter Ausschnitt eines Online-Shops vorgestellt. Erfasst alle wesentlichen Anwendungsfälle in Form von Use Cases. Überlegt, wo man sinnvoll extensions und inclusions verwenden könnte.

Nachdem eine Bestellung zusammengestellt und abgeschickt wurde soll es dem Kunden möglich sein, die Bestellung noch zurück zu ziehen. Diese Funktion soll gut auf der Webseite platziert sein und durch einen einfachen Klick mit einem Bestätigungsdialog erfolgen können. Dies ist aber nur möglich solange die Bestellung noch nicht versendet wurde. In diesem Fall wird die Bestellung aus dem System gelöscht und die entsprechenden Produkte wieder im System zurückgebucht.

Zu jeder Zeit soll es einem Mitarbeiter möglich sein, die Anzahl der verfügbaren Produkte im System zu erhöhen, wenn entsprechend Produkte im Großkundenbestellsystem bestellt wurden und im Lager vorliegen.

Produkte können von einem Mitarbeiter neu ins Sortiment aufgenommen oder entfernt werden. Beim Entfernen dürfen keine offenen Bestellungen mit diesem Produkt vorhanden sein. Beim Hinzufügen muss unbedingt darauf geachtet werden, dass ein Bild zum Produkt hinzugefügt wird, für das die Nutzungsrechte vorliegen.

Sobald die Auslieferung vom Spediteur bestätigt wurde, wird der Status der Bestellung auf ausgeliefert gesetzt.

## 2. Sequenzdiagramme

Modelliert nun zu den definierten Use Cases Sequenzdiagramme, die die Interaktion des Anwenders mit dem System als Ganzes beschreiben. Überlegt euch auch entsprechende Szenarien zur Behandlung von Fehlerfällen.

## 3. Klassen in UML

Zur Beschreibung der Struktur eines Systems werden in der UML meist Klassen (bzw. Klassendiagramme) verwendet. Für Klassen können dabei insbesondere Attribute und Methoden deklariert werden. Die Kategorie einer Klasse beschreibt ihre Aufgabe innerhalb des Systems.



Modelliert nun einige Klassen aus, basierend auf der folgenden Anforderungsbeschreibung.

Motorsport Gomez möchte die Abfertigung von Aufträgen mit einer Software verwalten.

Ein Kunde kann eine Reparatur oder ein Tuning für sein Motorrad vom Verwaltungspersonal in Auftrag geben lassen. Dazu wird das Kennzeichen des Motorrads im System vermerkt und der Auftrag erhält automatisch einen Datumsstempel. Wenn ein Auftrag beendet ist, kann ein Mitarbeiter diesen als beendet markieren.

Das Verwaltungspersonal soll die Möglichkeit haben, Kundendaten aufzunehmen sowie Mitarbeiterkontos zu erstellen. Kundendaten beinhalten Name, Adresse und Telefonnummer. Bei der Erstellung eines Mitarbeiterkontos wird die automatisch generierte Mitarbeiter-ID zurückgegeben.

## 4. Objekte in UML

Objekte (Objektdiagramme) sind Instanzen von Klassen (Klassendiagrammen). Modelliert nun einige Beispielobjekte für die oben erstellten Klassen. Diskutiert dabei insbesondere, wann ein Objekt zu einer Klasse “passt” und wann nicht.