



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 2022/2023

Prof. Dr. Sabine Glesner
Milko Monecke
Simon Schwan

Übungsblatt 9

Aufgabe 1: Nicht-funktionale Anforderungen

Die folgende Anforderungsbeschreibung wurde euch für die Entwicklung eines Programms geliefert. 



Eine Autowerkstatt möchte die Abfertigung ihrer Aufträge komfortabel mit einer Software verwalten. Dazu kann ein Mitarbeiter im System Kundys anlegen und ihnen Fahrzeuge zuordnen. Für ein neues Kundy wird 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 vom Mitarbeiter als beendet markiert werden. In diesem Fall wird das Kundy automatisch vom System benachrichtigt. Außerdem wird für den Auftrag vermerkt, welches Mitarbeiter ihn beendet hat.

Um Missbrauch vorzubeugen, müssen sich die Mitarbeiters am Browser mit ID und Passwort sicher anmelden. Ein Administrator kann Mitarbeiters anlegen und entfernen.

- a) Was ist der Unterschied zwischen funktionalen und nicht-funktionalen Eigenschaften?
- b) Überlegt, welche nicht-funktionalen Anforderungen aus dem Text oben hervorgehen.

Schlüssel:

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

- c) Welche Eigenschaften sind für dieses System wahrscheinlich auch noch wichtig?

Lösung:

- a) Funktionale Anforderungen beschreiben, was das System bei bestimmten Eingaben (ggf. nicht) tun soll.
Nicht-funktionale Anforderungen beschreiben, wie (im Sinne von Eigenschaften) das System bestimmte Dinge machen sollte. Oft treffen nicht-funktionale Eigenschaften nicht nur einzelne Module, sondern betreffen das gesamte Systeme.
- b) Folgende kann man finden:
- Produktanforderung - Usability: Abfertigung soll komfortabel sein.
 - Produktanforderung - Security: Anmeldung soll sicher sein.

Aufgabe 2: Requirements Engineering



- a) Ermittelt die im Text von Aufgabe 2 enthaltenen Use-Cases.
- b) Diskutiert, welche Anforderungen das Kundy wahrscheinlich zusätzlich an das System haben wird bzw. welche der Beschreibungen unklar sind.
- c) Erstellt aus zwei Use-Cases User Stories. Sind Vorteile der User Stories ersichtlich?

Lösung: Folgende Use-Cases gibt es:

- Kundy anlegen
- Fahrzeug erstellen (und zuordnen)
- Auftrag anlegen (implizit)
- Auftrag beenden
- Als Mitarbeiter anmelden
- Mitarbeiter registrieren
- Mitarbeiter entfernen

Und die hier fehlen wahrscheinlich:

- Kunde aktualisieren/deaktivieren
- Fahrzeug löschen (bzw. deaktivieren)

- Auftrag stornieren
- Passwort ändern
- Offene Aufträge auflisten

Etwas weiter gehend:


- Rechnung erstellen
- Bestellung ändern

Mögliche User Stories:

- Als Mitarbeiter möchte ich neue Kundys im System anlegen können und ihnen Fahrzeuge zuordnen.
- Als Administratör möchte ich Mitarbeiter anlegen, verwalten und entfernen.

Vorteile User Stories: Aus Sicht des Anwendys (was braucht das Benutzy vs. was sollte das System liefern), mehr Informationen im Text möglich im Gegensatz zu Schlüsselwort, Anwender bekannt; Informelle Struktur ermöglicht freie Darstellung relevanter Informationen.

Aufgabe 3: Strukturierte Anforderungsspezifikation

Modelliert einige der in Aufgabe 3 beschriebenen Anwendungsfälle (Use-Cases) in Form von strukturierten Spezifikationen. Überlegt euch sinnvolle Attribute zur Strukturierung. 

Lösung:

Beispielanwendungsfall: Auftrag anlegen.

Funktion Auftrag anlegen

Beschreibung Ein neuer Auftrag vom richtigen Typ wird im System angelegt. Diesem Auftrag ist ein Fahrzeug zugeordnet.

Input Auftrag (Reparatur, Inspektion, Reifenwechsel), Kundy, Fahrzeug/Kennzeichen, Preis, Beschreibung

Output Erfolgs-/Misserfolgs-Nachricht

Aktion Auftrag im System anlegen.

Vorbedingung Kundy existiert, Fahrzeug mit Kennzeichen existiert und gehört diesem Kundy.

Nachbedingung Auftrag existiert und ist dem Fahrzeug zugewiesen.

Beispielanwendungsfall: Mitarbeiter entfernen

Funktion Mitarbeiter entfernen

Beschreibung Ein Mitarbeiter wird aus dem System entfernt

Input Mitarbeiter-ID

Output Erfolgs-/Misserfolgs-Nachricht

Aktion Löschen des Mitarbeiters im System

Vorbedingung Der Mitarbeiter mit der ID existiert

Nachbedingung Der Mitarbeiter mit der ID existiert nicht

Für die anderen Anwendungsfälle analog.