



17. November 2021

Übungen zur Vorlesung Software Engineering I WS 2021 / 2022

Übungsblatt Nr. 6

(Abgabe Aufgabe 6-1: Mittwoch, den 24. November 2021, **09:00 Uhr**
Aufgabe 6-2: Mittwoch, den 1. Dezember 2021, **09:00 Uhr**)

Aufgabe 1 (Entwicklung UML Use Case Diagramm, 15 Punkte):

Im Rahmen des Projekts Coll@HBRS ist ein textueller Use Case „Jobangebot einstellen“ zur weiteren Beschreibung einer Job-Börse entwickelt worden. Diese finden sie auf dem LEA-Server im Abschnitt 4 (Word und PDF-Datei „Use Case Jobangebot einstellen“).

Ihre Aufgaben:

a.)

In dem Use Case sind einige TODOs (rot markiert) offengeblieben. Bitte bearbeiten sie diese entsprechend in dem Word-File (oder in einem entsprechend kopierten Format):

- **TODO Nr. 1:** Fügen sie Akteure hinzu, die *unmittelbar* mit diesem Use Case assoziiert sind.
- **TODO Nr. 2:** Fügen sie einen neuen textuellen Supplier Use Case hinzu, der an dieser Stelle (= Extension Point!) den Basis Use Case potentiell erweitern und somit abbrechen könnte, falls die Beschreibung des Jobangebots zu unspezifisch für den Studiengang Informatik ist (= Condition!). In dem Supplier Use Case soll der Unternehmer eine entsprechende Meldung über einen Ausgabedialog erhalten. Die Eingabe des Jobangebots kann dann komplett neu gestartet werden oder kann die Anwendung verlassen (Logout). Eine Vor- und Nachbedingung müssen sie in dem Supplier Use Case *nicht* spezifizieren!
- **TODO Nr. 3:** Fügen sie einen neuen textuellen Supplier Use Case „Bankdaten ermitteln“ hinzu, der überprüft, ob der Benutzer ggf. bereits im Vorfeld seine Bankdaten hinterlegt hat. Ansonsten sollte der Unternehmer seine Bankdaten erstmalig eingeben, damit diese abgespeichert werden können. In *beiden* Fällen sollen dann die Bankdaten zurückgegeben werden. Ein Benutzer kann eine beliebige Anzahl von Bankdaten besitzen.
- **TODO Nr. 4:** Spezifizieren sie im gleichen textuellen Use Case einen alternativen Ereignisfluss, der während des Ereignis Nr. 7 ausgeführt wird, falls der Use Case „Bankeinzug durchführen“ eine Fehlermeldung ausgibt. Der Unternehmer soll dann, im Rahmen des alternativen Ereignisflusses, die Bankdaten noch mal sichten und

diese ggf. korrigieren. Falls der Vorgang dann erfolgreich abgeschlossen werden kann, so soll zum Ereignis Nr. 8 navigiert werden.

- **TODO Nr. 5:** Spezifizieren sie eine testbare Nachbedingung.

b.)

Entwickeln sie aus den gegebenen textuellen Use Cases (inkl. der Erweiterungen, die sich aus den Todos ergeben) *ein* UML Use Case Modell zur Darstellung der Funktionen für das System Coll@HBRS. Von dem Akteur „Unternehmer“ aus brauchen sie nur die Assoziationen zu den Basis Use Cases zu berücksichtigen. Die in der Vorbedingung angedeuteten Use Cases für das Login sowie für die Registrierung brauchen sie nicht zu berücksichtigen).

Aufgabe 2 (Entwicklung UML Klassendiagramm, 15 Punkte):

Wenden sie auf die textuellen Use Cases (inkl. ihrer Erweiterungen) aus der Aufgabe 1 die Abbott-Methode an, und entwickeln sie ein Klassendiagramm zur Darstellung des Analyse-Modells. Jede Klasse sollte mit einem Objekttyp, analog wie in der Vorlesung eingeführt, annotiert werden. Modellieren sie auch die wichtigsten Beziehungen zwischen den Klassen. Anbei ein paar Hinweise zur Modellierung, die sie beachten sollten (siehe auch Hinweise in Kapitel 4, Abschnitt 4):

- Modellieren sie vorrangig die Beziehungen ausgehend von *der zentralen Control-Klasse*, die sich aus dem Use Case „Jobangebot einstellen“ ergibt.
- Die Modellierung der Beziehungen zwischen der zentralen Control-Klasse und den Boundary-Klassen sollten sie auf ein minimales Maß reduzieren. Gehen sie davon aus, dass die Boundary-Klasse „Einstellung eines Jobangebots“ diejenige Klasse ist, die mit der zentralen Control-Klasse interagiert. Alle übrigen Boundary-Klassen, die Masken repräsentieren, können sie als „Sub Boundaries“ der Boundary-Klasse „Einstellung eines Jobangebots“ annehmen und *nicht* explizit in ihr Klassendiagramm integrieren.
- Die Boundary-Klassen, welche Masken aus den Supplier Use Cases präsentieren, brauchen sie ebenfalls nicht zu modellieren.
- Modellieren sie nur die Datentransferobjekte („DTOs“), die sich explizit aus dem Text des Use Case ergeben. Datentransferobjekte zur Kommunikation mit Boundaries, welche Schnittstellen zu Fremdsystemen repräsentieren, brauchen sie *nicht* zu berücksichtigen.
- Modellieren sie die Beziehungen zwischen Entities mittels Assoziationen inklusive den zugehörigen Multiplizitäten. Modellieren sie die Abhängigkeiten von Control-Klassen zu Entities nur ausgehend von *der zentralen Control-Klasse*, die sich aus dem Use Case „Jobangebot einstellen“ ergibt.
- Methoden brauchen sie *keine* modellieren.