



23. Mai 2024

## **Übungen zur Vorlesung Software Engineering I Sommersemester 2024**

### **Übungsblatt Nr. 6**

(Abgabe in Teams von max. 3 Personen bis: Mittwoch, den 5. Juni 2024, **9:00 Uhr**)

#### **Bitte beachten Sie:**

Wegen des Feiertags am 30.5. finden Vorlesung und Übung in der kommenden Woche nicht statt. Die Abgabefrist für diese Übung wird deshalb auf den 5.6. verlängert!

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

Im Rahmen des Projekts aus der vorangegangenen Übung wurde folgender textueller Use Case entwickelt, wobei einige TODOs offengeblieben sind:

**Use Case:** Projektausschreibung einstellen

**System:** NullPointer Collaboration Portal

**Akteure:** **TODO #1**

#### **Vorbedingung:**

- Registrierter Unternehmer hat sich im System eingeloggt (Use Case: Login eines registrierten Benutzers).
- Unternehmer hat Button zur Einstellung einer Projektausschreibung betätigt, das System zeigt Formular dafür an.

#### **Regulärer Ereignisfluss:**

1. Der Unternehmer bestimmt die Art der Projektausschreibung, es gibt die folgenden Arten: „Festanstellung“ oder „Selbständiger“. Danach gelangt er durch einen Button auf die nächste Eingabeseite, die vom System angezeigt wird.
2. Der Unternehmer erstellt eine Beschreibung innerhalb der Eingabeseite. Dann gelangt er mit einem Button auf die nächste Eingabeseite. Er hat die Möglichkeit, von dort zurückzunavigieren, um die Beschreibung noch einmal anzupassen (Rücksprung Schritt 2).
3. Der Unternehmer erhält eine Übersicht aller eingegebenen Daten. Er kann diese Daten in der Übersicht noch korrigieren, wenn ihm Fehler auffallen. Stimmen die

Daten, kann er den Auftrag zur Einstellung mit einem Button an das System abschicken.

4. Das System erhält den Einstellauftrag des Unternehmers und verifiziert die Gültigkeit der Daten. (TODO #2)
5. Das System erstellt eine Projektausschreibung und speichert diese in einer externen Datenbank eines Cloud-Speichers.
6. Das System ermittelt die Bankdaten des Unternehmers und speichert diese via des Use Case „Bankdaten ermitteln“ vorübergehend zur weiteren Verwendung. (TODO #3)
7. Das System ermittelt die Kosten für die Einstellung der Projektausschreibung und bucht vom Konto des Unternehmens via Use Case „Bankeinzug durchführen“ den Betrag ab. Das System speichert die Zuordnung der neuen Projektausschreibung zum Unternehmer. TODO #4: Alternativer Ereignisfluss zu spezifizieren (Hinweis: Innerhalb des hier eingebundenen Supplier Use Cases wird der Auftrag an das externe Buchungssystem „Banking-Software“ weitergereicht).
8. Das System sendet an den Unternehmer eine Bestätigung in Form einer neuen Bestätigungsseite, dass die Projektausschreibung eingestellt wurde und die Kosten abgebucht wurden.

Alternativer Ereignisfluss Nr. 1:  
(Zu spezifizieren, vgl. TODO #2)

Alternativer Ereignisfluss Nr. 2:  
(Zu spezifizieren, vgl. TODO #4)

Nachbedingung:  
TODO #5

Bitte bearbeiten Sie nun die folgenden Aufgaben:

### Aufgabe 1.1:

Bearbeiten Sie die offengebliebenen TODOs:

- **TODO #1:** Nennen Sie die Akteure, die mit diesem Use Case assoziiert sind.
- **TODO #2:** Spezifizieren Sie im gleichen textuellen Use Case einen alternativen Ereignisfluss bezüglich des Ereignisses Nr. 4, falls bei der Verifikation der Ausschreibung ein Fehler auftritt. Der Unternehmer soll dann die fehlerhaften

Daten angezeigt bekommen und die Möglichkeit erhalten, diese zu korrigieren. Hat er dies getan, soll zum Ereignis Nr. 4 zurückgesprungen werden.

- **TODO #3:** Erstellen Sie die Beschreibung eines neuen textuellen Use Cases „Bankdaten ermitteln“, der überprüft, ob der Unternehmer schon seine Bankdaten hinterlegt hat. Wenn nicht, soll er diese eingeben, so dass sie gespeichert werden können. In beiden Fällen sollen dann die Bankdaten zurückgegeben werden. Ein Unternehmer kann eine beliebige Anzahl von Bankdaten besitzen.
- **TODO #4:** Spezifizieren Sie im gleichen textuellen Use Case einen alternativen Ereignisfluss, der im Zuge von Ereignis Nr. 7 ausgeführt wird, wenn der Use Case „Bankeinzug durchführen“ eine Fehlermeldung ausgibt. Der Unternehmer soll dann im Rahmen des alternativen Ereignisflusses des Basis-Use-Case „Projektausschreibung einstellen“ die Bankdaten überprüfen und korrigieren. War das erfolgreich und der Bankeinzug konnte mit den korrigierten Daten durchgeführt werden, soll zum Ereignis Nr. 8 gesprungen werden. Im Use Case „Bankeinzug durchführen“ findet keine Interaktion mit dem Unternehmer statt.
- **TODO #5:** Spezifizieren Sie eine testbare Nachbedingung.

### Aufgabe 1.2:

Erstellen Sie auf Basis der gegebenen textuellen Use Cases (einschließlich der Erweiterungen der TODOs) **ein** UML-basiertes Use-Case-Modell zur Darstellung der Funktionalität des NullPointer Collaboration Portals. Die in der Vorbedingung angedeuteten Use Cases für den Benutzer-Login (Benutzer ist abstrakter Akteur!) sowie für die Registrierung eines Unternehmers sollten Sie ebenfalls berücksichtigen.

### Aufgabe 2 (Entwicklung Objektorientiertes Analysemodell (OOA), 15 Punkte):

Setzen Sie die in der Vorlesung vorgestellte Abbott-Methode ein, um aus den textuellen Use Cases (einschließlich der Erweiterungen aus den TODOs) Artefakte für eine objektorientierte Modellierung abzuleiten. Entwickeln Sie ein Klassendiagramm zur Darstellung eines Objektorientierten Analysemodells (OOA). Jede Klasse sollte mit einem Objekttyp (Entity, Boundary, Control oder Data Transfer Object) annotiert werden.

Dazu einige Hinweise:

- Modellieren Sie die Beziehungen ausgehend von der zentralen Control-Klasse, die sich aus dem Use Case „Projektausschreibung einstellen“ ergibt.
- Die Modellierung der Beziehungen zwischen der zentralen Control-Klasse und den Boundary-Klassen sollten Sie möglichst kompakt halten. Gehen Sie davon aus, dass die Boundary-Klasse „Eingabe Projektausschreibung“ mit der zentralen Control-Klasse interagiert. Alle anderen Boundary-Klassen, die Masken

darstellen, können Sie als „Sub Boundaries“ der Boundary-Klasse „Eingabe Projektausschreibung“ annehmen und müssen Sie nicht explizit in Ihr Klassendiagramm integrieren.

- Die Boundary-Klassen, welche Masken aus den Supplier Use Cases darstellen, müssen nicht modelliert werden.
- Modellieren Sie nur diejenigen Datentransferobjekte (DTOs), die sich explizit aus dem Text des Use Cases ergeben. Berücksichtigen Sie keine DTOs zur Kommunikation mit Boundaries, die Schnittstellen zu Fremdsystemen darstellen.
- Modellieren Sie die Beziehungen zwischen Entities mittels Assoziationen einschließlich der zugehörigen Multiplizitäten. Modellieren Sie die Abhängigkeiten von Control-Klassen zur Entities nur ausgehend von der zentralen Control-Klasse, die sich aus dem Use Case „Projektausschreibung einstellen“ ergibt.
- Die Control-Klassen, die sich aus den Vorbedingungen ergeben, müssen nicht berücksichtigt werden.

Bitte speichern Sie alle Ergebnisse als PDF und laden Sie sie auf LEA hoch.