Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Prof. Dr. Sascha Alda
Moritz Balg, Dr. Rita Cornely

20. November 2024

Übungen zur Vorlesung Software Engineering I
WS 2024 / 2025

Übungsblatt Nr. 6
(Abgabe Aufgabe 6-1: Mittwoch, den 27. November 2024, **10:00 Uhr**
Aufgabe 6-2: Mittwoch, den 4. Dezember 2024, **10:00 Uhr**)

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

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

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 mit diesem Use Case assoziiert sind.

- **TODO Nr 2:** Spezifizieren sie im *gleichen* textuellen Use Case einen alternativen Ereignisfluss bezüglich des Ereignisses Nr. 4, falls bei der Verifikation einer Projektausschreibung durch das SYSTEM ein Fehler auftritt. In der Gesamtübersicht soll dem Unternehmer dabei die fehlerhaften Daten angezeigt werden. Er hat dort auch die Möglichkeit, die fehlerhaften Daten neu einzugeben. Nach erfolgreicher Eingabe soll zum Ereignis Nr. 4 zurückgesprungen werden.

- **TODO Nr. 3:** Fügen sie einen *neuen* textuellen Supplier-Use-Case „Bankverbindungen verwalten“ hinzu. In einer Option „Bankverbindung auswählen“ soll der Unternehmer aus einer bereits eingegebenen Bankverbindung auswählen. Die gewählte Bankverbindung sollte dann als Ergebnis zurückgeliefert werden. In einer weiteren Option sollte der Unternehmer eine neue Bankverbindung eingeben können. Weitere Optionen brauchen Sie nicht zu spezifizieren Ein Unternehmer kann eine beliebige Anzahl von Bankverbindungen besitzen. Beide Optionen sollen in dem Haupt-Ereignisfluss spezifiziert werden. Formulieren Sie zudem in dem Supplier Use Cases ein Alternativ-Szenario zur Behandlung von Fehlern (z.B. falsche IBAN eingegeben).

- **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, dass der Bankeinzug nicht durchgeführt werden konnte. Der Unternehmer soll dann, im Rahmen des alternativen Ereignisflusses des Basis-Use-Case „Projektausschreibung aufgeben“,

die Bankverbindung noch mal überprüfen und diese ggf. korrigieren. Falls der Vorgang dann erfolgreich abgeschlossen werden kann, so soll zum Ereignis Nr. 8 navigiert werden. In dem Use-Case „Bankeinzug durchführen“ findet *keine* Interaktion mit dem Unternehmer statt.

- **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-basiertes Use-Case-Modell zur Darstellung der Funktionen für das System Coll@HBRS. Die in der Vorbedingung angedeuteten Use Cases für das Login eines Benutzers (Benutzer als abstrakten Akteur berücksichtigen!) sowie für die Registrierung eines Unternehmers sollten Sie ebenfalls berücksichtigen.

c.)
Ein weiterer zentraler Use-Case der Plattform lautet „Projekt auf Basis von Skills planen“. Entwickeln Sie dazu fünf aussagekräftige User-Stories, welche diesen Use-Case konkretisieren. Die User-Stories sollten in Anlehnung an die INVEST-Merkmale entwickelt werden.

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

Wenden sie auf die textuellen Use Cases (inkl. den Erweiterungen aus der Aufgabe 6-1) die Abbott-Methode an, und entwickeln sie ein Klassendiagramm zur Darstellung des objektorientierten Analyse-Modells (OOA). Jede Klasse sollte mit einem Objekttyp, analog wie in der Vorlesung eingeführt, annotiert werden. Modellieren sie auch die wichtigsten Beziehungen zwischen den Klassen, Methoden brauchen sie keine zu modellieren. Hier einige Hinweise zur Modellierung, die sie beachten sollten (vgl. Hinweise in Kap. 4, Abschnitt 4):

- Modellieren sie vorrangig die Beziehungen ausgehend von *der zentralen Control-Klasse*, die sich aus dem Use Case „Projektausschreibung aufgeben“ 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 „Eingabe Projektausschreibung“ 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 „Eingabe Projektausschreibung“ 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 „Projektausschreibung aufgeben“ ergibt.

- Die Control-Klassen, die sich aus der Vorbedingung ergeben, brauchen Sie nicht zu berücksichtigen.