

Use Case Name:

Projektausschreibung aufgeben

System: Coll@HBRS

Akteure:

Unternehmer (primärer Akteur)

Externes Cloud-System (für persistente Speicherung der Ausschreibung)

Vorbedingung:

- Registrierter Unternehmer hat im System Coll@HBRS eingeloggt (Use Case: „Benutzer authentifiziert sich“; die Registrierung ist im Use Case „Unternehmer registriert sich“ erfolgt).
- Der Unternehmer hat auf der Startseite von Coll@HBRS die Seite zur Einstellung einer Projektausschreibung angefordert, das System hat die Seite entsprechend dargestellt.

Grundlegender Ereignisfluss:

1. Der Unternehmer spezifiziert in einem Unterbereich dieser Unterseite die Daten für seine Projektausschreibung, die ein Student erfüllen muss. Dabei bestimmt er *innerhalb* von *diesem* Anwendungsfall auch die Art der Projektausschreibung. Folgende Arten von Projektausschreibungen werden unterstützt: Praxisprojekt, Festanstellung oder Freelancer. Falls die Eingabe getätigt wurde, navigiert der Unternehmer zur nächsten Eingabeseite. Das System zeigt die Eingabeseite an.
2. Der Unternehmer schreibt einen Beschreibungstext der Projektausschreibung innerhalb dieser Eingabeseite. Falls die Eingabe fertig ist, navigiert er zur nächsten Seite, die das System ihm anzeigt. Alternativ kann er zur letzten Eingabeseite zurück navigieren, um dort Änderungen vorzunehmen (→ Rücksprung Schritt 2).
3. Der Unternehmer kann in einer Gesamtübersicht seine eingegebenen Daten lesen. Er muss die Korrektheit der Daten bestätigen. Falls er Fehler in seinen Beschreibungen feststellt, kann diese in der Gesamtübersicht korrigieren. Falls keine Fehler vorhanden sind, schickt er den Auftrag zur Einstellung der Projektausschreibung an das System.
4. Das System empfängt die eingegebenen Daten des Auftrags für die Projektausschreibung und verifiziert die Gültigkeit der Daten. **(ToDo Nr. 2)**
5. Das System erstellt auf Basis der korrekten Daten eine Projektausschreibung und überführt die Angaben aus diesem in ein externes Informationssystem eines Cloudanbieters.
6. Das System ermittelt die Bankverbindung des Unternehmers und speichert diese temporär zur weiteren Verwendung. **(durch Einbeziehung des Use Case „Bankverbindung verwalten“), vgl. TODO Nr. 3).**
7. Das System ermittelt den Betrag für die Projektausschreibung und bucht von dem Konto des Unternehmens über die Einbeziehung des Use Case „Bankeinzug durchführen“ den Betrag per Bankeinzug ab. Der Unternehmer wird mit der neuen Projektausschreibung assoziiert. **(Alternativer Ereignisfluss unten zu spezifizieren, vgl. TODO Nr. 4. Hinweis für die Modellierung: innerhalb des Supplier**

Use Cases wird der Auftrag an das externe Buchungssystem „Banking-Software“, welches nur mit dem Supplier Use Case assoziiert ist, weitergereicht).

8. Das System schickt dem Unternehmer eine Bestätigung zurück auf eine neue Ausgabeseite, dass die Projektausschreibung erfolgreich eingestellt werden konnte. Die Bestätigung enthält außerdem eine Mitteilung, dass das System den Betrag erfolgreich abbuchen konnte.

Alternativer Ereignisfluss Nr.1:

(Alternativer Ereignisfluss spezifizieren, vgl. TODO Nr. 2)

4-1: System erkennt fehlerhafte Eingaben -> zeigt Fehlermeldung (bei dem Unternehmer).

4-2: System zeigt fehlerhafte Daten in der Gesamtübersicht an. Es besteht die Möglichkeit die fehlerhaften Daten neu einzugeben.

4-3: Nach neuer Eingabe -> zurück zu Schritt 4.

Falls wieder fehlerhaft -> zurück zu 4-1.

TODO Nr. 3:

Use Case Name: Bankverbindungen verwalten

Akteur: Unternehmer

Vorbedingungen: - Unternehmer ist eingeloggt.

-Seite „Bankverbindung verwalten“ ist geöffnet.

Ereignisfluss:

1. System bietet zwei Optionen:
 - a. Bankverbindung auswählen.
 - b. Bankverbindung eingeben.
2. Bankverbindung auswählen:
 - a. Falls keine vorhanden, eine Meldung wird gezeigt und der Unternehmer soll neu eingeben.
 - b. Falls vorhanden, eine Liste wird angezeigt, der Unternehmer soll eine Möglichkeit aussuchen, und das System liefert die Auswahl zurück.
3. Bankverbindung eingeben:
 - a. System zeigt Eingabemaske.
 - b. Unternehmer gibt Daten ein und das System speichert diese.

Alternativer Flow: Bei ungültigen Eingaben zeigt das System einen Fehler an, der Unternehmer korrigiert die Daten, und anschließend wird der Eingabeprozess erneut durchlaufen.

Nachbedingung: Bankverbindung wurde ggf. hinzugefügt, also keine Daten gehen verloren.

Alternativer Ereignisfluss Nr.2:

(Alternativer Ereignisfluss spezifizieren, vgl. TODO Nr. 4)

7-1: Falls das System einen Fehler im Bankeinzug meldet, sendet das System eine Meldung an dem Unternehmer, die Bankverbindung zu überprüfen.

7-2: Unternehmer korrigiert die Bankdaten und schickt die korrigierten Daten zurück an das System.

7-3: Das System versucht erneut den Bankeinzug mit Bankeinzug durchführen.

7-4: Falls das erfolgt, weiter zu Schritt 8.

Falls wieder fehlerhaft -> zurück auf Schritt 7-2.

Nachbedingung:

TODO Nr. 5!

Die Projektausschreibung ist persistent gespeichert und für Studierende sichtbar.

AUFGABE 6.1.c:

User Story 1: Als Unternehmer möchte ich eine Liste relevanter Skills auswählen können, damit das System passende Studierende für mein Projekt identifizieren kann.

User Story 2: Als Unternehmer möchte ich eine übersichtliche Liste der Studierenden erhalten, deren Skills zu meinem Projekt passen, damit ich fundierte Entscheidungen treffen kann.

User Story 3: Als Unternehmer möchte ich die Bedeutung einzelner Skills priorisieren können, damit das System meine Projektanforderungen genauer gewichten kann.

User Story 4: Als Unternehmer möchte ich automatisch einen ersten Projektentwurf basierend auf den gefundenen Skills erhalten, damit ich schnelle und strukturierte Ausgangsbasis für Projektplanung habe.

User Story 5: Als Unternehmer möchte ich den automatisch erzeugten Projektentwurf anpassen und verfeinern können, damit das finale Projekt exakt meinen Vorstellungen entspricht.