2025

Use-Case-Dokumentation für die Anwendung „Auftragscockpit“

AUFTRAGS COCKPIT

FHDW: MODUL: SOFTWARE MODELING & ARCHITECTURE
VON ARDA ARIF ERMAYA & NIKOLAS PERING (2025)

1.0 Use Case: Login:

1. Erfolgsfall:

- (a) Benutzer navigiert zur Startseite der Anwendung „Auftragscockpit“ unter dem folgenden Link: http://localhost:8080/login.html
- (b) System zeigt das Menü „Willkommen zurück“ mit den Eingabefeldern für E-Mail und Passwort an.
- (c) Benutzer befüllt beide Felder mit gültigen Daten, die für den Login benötigt werden.
- (d) System validiert die Eingaben als vollständig und gültig, belässt alle Eingaben sichtbar.
- (e) Benutzer drückt auf den Button „Anmelden“.
- (f) System verarbeitet die eingegeben Daten und prüft, ob die Anmeldedaten korrekt sind. Ist dies der Fall, so kommt der Benutzer auf die eigentliche Willkommensseite, wo er weitreichende Funktionalitäten zur Auswahl stehen hat (http://localhost:8080/index.html).

2. Fehlerfall:

- (a) Benutzer navigiert zur Startseite der Anwendung „Auftragscockpit“ unter dem folgenden Link: http://localhost:8080/login.html
- (b) System zeigt das Menü „Willkommen zurück“ mit den Eingabefeldern für E-Mail und Passwort an.
- (c) Benutzer befüllt beide Felder mit Daten, von denen er annimmt, dass diese Korrekt sind und klickt auf den Button „Anmelden“
- (e) System gibt den Benutzer eine Fehlermeldung aus „Ungültige Zugangsdaten. Bitte versuche es erneut.“ Und lässt die zuvor eingegeben Daten bestehen
- (f) Der Benutzer hat nun zwei Möglichkeiten: Entweder gibt er die Benutzerdaten erneut ein und versucht es ein zweites Mal, oder er klickt auf das „Augensymbol“ in „Passwort“ und erschließt sich selbst, wo der Fehler bei der Eingabe des Passwortes gelegen hat.

Akzeptanzkriterien:

- Nur korrekt eingegebene Logindaten führen den Benutzer auf die nächste Seite weiter
- Sollten falsche Logindaten eingegeben worden sein, so hat der Benutzer die Möglichkeit seine Eingaben x-beliebig oft anzupassen.

1.1 Use Case: Artikel anlegen

1. Erfolgsfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“ mit den Eingabefeldern Artikelnummer, Name, Einkaufspreis (EUR), Verkaufspreis (EUR), Kategorie (5 Vorauswahlen), Lagerbestand, Beschreibung sowie dem Umschalter „Artikel ist aktiv“.
- (c) Benutzer befüllt jedes Feld mit gültigen Daten (z.B. Artikelnummer „A-1001“, Name „Standardartikel“, Preise als Dezimalzahl, eine der vorgegebenen Kategorien, Lagerbestand als Zahl, optionale Beschreibung und setzt „Artikel ist aktiv“ auf aktiv.
- (d) System validiert die Eingaben als vollständig und gültig, belässt alle Eingaben sichtbar.
- (e) Benutzer drückt auf den Button „Artikel speichern“.
- (f) System persistiert den Artikel und zeigt ihn in der Tabelle „Alle Artikel“ mit genau den eingegeben Werten und aktivem Status an.

2. Fehlerfall:

- (a) Benutzer navigiert zur Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“ mit allen Eingabefeldern.
- (c) Benutzer füllt alle Pflichtfelder aus, lässt aber das Dropdown „Kategorie“ unausgewählt.
- (d) System validiert und erkennt die fehlende Kategorie, belässt alle bisher eingegebenen Werte im Formular.
- (e) Benutzer klickt auf Artikel „speichern“.
- (f) System verhindert das Speichern, gibt einen Validierungsfehler „Wählen Sie ein Element in der Liste aus.“ aus und lässt alle bisherigen Eingaben unverändert sichtbar.

Akzeptanzkriterien:

- POST/Save mit den oben beschriebenen gültigen Werten führt zu HTTP 200/201 und einem neuen Datensatz in der Artikeltabelle/Repository
- Der gespeicherte Datensatz enthält exakt die gesendeten Fehler (inkl. aktiv = true) und wird von der List-API bzw. der „Alle Artikel“-Ansicht zurückgegeben.
- POST/Save mit Kategorie = null/leer liefert http 400/422 mit einer Fehlermeldung „Kategorie ist erforderlich“.
- Nach der Fehlermeldung liefert Reload des Formularzustands dieselben eingegeben Werte (keine Datenverluste), abgesehen von der weiterhin fehlenden Kategorie

1.2 Use Case: Preisfelder validieren

1. Erfolgsfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer gibt in „Einkaufspreis (EUR)“ und „Verkaufspreis (EUR)“ gültige Dezimalzahlen ein; alle übrigen Pflichtfelder sind korrekt befüllt.
- (d) System validiert beide Preise als numerisch und formatiert sie ggf. korrekt.
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System persistiert den Artikel mit den Preisen und zeigt ihn in „Alle Artikel“ an.

2. Fehlerfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer gibt im Feld „Einkaufspreis (EUR)“ oder „Verkaufspreis (EUR)“ einen nicht-numerischen Wert ein (z. B. „e“), andere Felder gültig.
- (d) System erkennt das ungültige Format, markiert das betroffene Preisfeld und behält alle übrigen Eingaben bei.
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System bricht das Speichern ab und meldet einen Formatfehler („Geben Sie eine Zahl ein“) für das jeweilige Preisfeld.

Akzeptanzkriterien:

- POST/Save mit gültigen Dezimalpreisen liefert HTTP 200/201 und der Artikel erscheint in der „Alle Artikel“-Ansicht mit korrekter Dezimaldarstellung.
- POST/Save mit nicht-parsbarem Dezimalwert ergibt HTTP 400/422 und eine Feld-Fehlermeldung für das betroffene Preisfeld.
- Nach Validierungsfehler bleiben alle anderen eingegebenen Werte im Formular sichtbar.

1.3 Use Case: Artikelnummer-Eindeutigkeit

1. Erfolgsfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer gibt eine noch nicht verwendete Artikelnummer ein (case-insensitiv geprüft) und füllt alle weiteren Felder gültig aus.
- (d) System erkennt, dass die Artikelnummer noch nicht existiert.
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System persistiert den Artikel mit der eingegebenen Artikelnummer und zeigt ihn in „Alle Artikel“ an.

2. Fehlerfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer gibt eine bereits existierende Artikelnummer ein (case-insensitiv) und füllt alle anderen Felder gültig aus.
- (d) System erkennt den Konflikt durch Prüfung der vorhandenen Artikel.
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System lehnt das Speichern ab und zeigt „Fehler beim Speichern des Artikels: Serverfehler 400 […]“.

Akzeptanzkriterien:

- POST/Save mit neuer einzigartiger Artikelnummer liefert HTTP 200/201 und legt einen neuen Datensatz an.
- POST/Save mit Duplikat (nur andere Groß-/Kleinschreibung) liefert HTTP 400 mit Fehlermeldung „Fehler beim Speichern des Artikels: Serverfehler 400 […]“.
- Datenbestand bleibt unverändert, Formular zeigt weiter alle eingegebenen Werte inkl. der fehlerhaften Artikelnummer.

1.4 Use Case: Artikelstatus steuern

1. Erfolgsfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer füllt alle Pflichtfelder gültig aus und setzt „Artikel ist aktiv“ auf inaktiv.
- (d) System validiert die Eingaben und markiert den Status als inaktiv.
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System persistiert den Artikel als inaktiv und zeigt ihn in „Alle Artikel“ entsprechend an.
- (d) Wenn der Benutzer nun die PDF-Export-Funktion benutzt, wird der Artikel als „Inaktiv“ dargestellt

2. Fehlerfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer setzt „Artikel ist aktiv“ auf inaktiv, lässt aber ein Pflichtfeld (z. B. Name) leer.
- (d) System erkennt das fehlende Pflichtfeld und markiert es als fehlerhaft.
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System bricht das Speichern ab und weist auf das fehlende Pflichtfeld hin.

Akzeptanzkriterien:

- POST/Save mit aktiv = false und vollständigen Pflichtfeldern liefert HTTP 200/201 und speichert einen Artikel mit aktiv = false.
- POST/Save mit aktiv = false und fehlendem Pflichtfeld liefert HTTP 400/422 und markiert das Feld korrekt.
- Formularzustand bleibt nach der Fehlermeldung vollständig erhalten.
- Das erzeugte PDF zeigt den Status eines Artikels korrekt an

1.5 Use Case: Optionale Beschreibung

1. Erfolgsfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer lässt „Beschreibung“ leer, alle anderen Pflichtfelder gültig.
- (d) System akzeptiert das leere Feld.
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System persistiert den Artikel ohne Beschreibung und zeigt ihn in „Alle Artikel“ an.

2. Fehlerfall:

- (a) Benutzer navigiert zum Menüpunkt „Artikel“.
- (b) System zeigt das Menü „Neuen Artikel anlegen“.
- (c) Benutzer lässt „Beschreibung“ leer.
- (d) System würde fälschlicherweise eine Pflichtverletzung erkennen (unerwünscht).
- (e) Benutzer klickt auf „Artikel speichern“.
- (f) System bricht ab und meldet, dass eine Beschreibung erforderlich sei (unerwünscht).

Akzeptanzkriterien:

- POST/Save ohne Beschreibung liefert HTTP 200/201 und speichert Beschreibung = null/leer.
- Der Datensatz erscheint ohne Beschreibung in „Alle Artikel“.
- Falls das Backend Beschreibung fälschlich als Pflicht markiert, liefert POST/Save HTTP 400/422 → dieser Zustand gilt als Regression (Test soll sicherstellen, dass dies nicht mehr vorkommt).

1.6 Use Case: Artikel bearbeiten

1. **Erfolgsfall**:
   • (a) Benutzer navigiert zum Menüpunkt „Artikel“ und wählt einen bestehenden Artikel aus der Liste „Alle Artikel“.
   • (b) System lädt die aktuellen Felder in ein Bearbeitungsformular (Artikelnummer ebenfalls bearbeitbar).
   • (c) Benutzer ändert Name, Preis, Kategorie, Lagerbestand oder Aktiv-Status.
   • (d) System erkennt alle geänderten Felder als gültig.
   • (e) Benutzer klickt auf „Änderungen speichern“.
   • (f) System speichert die Änderungen und zeigt den aktualisierten Datensatz in der Artikelliste.
2. **Fehlerfall**:
   • (a) Benutzer navigiert zum Menüpunkt „Artikel“ und wählt einen bestehenden Artikel aus der Liste „Alle Artikel“.
   • (b) Benutzer bearbeitet einen ausgewählten Artikel und möchte dessen Artikelnummer ändern. Dabei gibt er eine bereits vorhandene Artikelnummer unwissentlich an.
   • (c) Backend meldet PUT 400 „Bad Request“ und zeigt grafisch die folgende Fehlermeldung dar „Fehler beim Speichern des Artikels: Serverfehler: 400 […]“.
   • (d) Formular bleibt sichtbar, keine Daten werden gespeichert, der Benutzer bekommt die Möglichkeit eine andere Artikelnummer, die nicht vergeben ist zu wählen.

Akzeptanzkriterien:
• Erfolgreicher PUT liefert HTTP 200 und gibt den aktualisierten Artikel zurück.

• PUT auf eine bereits existierende Artikelnummer liefert HTTP 400 (Bad Request) und persistiert keine Änderungen.
• System zeigt geänderte Werte korrekt in der Liste an.

1.7 Use Case: Lagerbestand anpassen

1. **Erfolgsfall**:
   • (a) Benutzer öffnet einen bestehenden Artikel und wählt „Bestand anpassen“.
   • (b) System zeigt aktuellen Bestand und die Eingabemöglichkeit einer Bestandsänderung (positiv oder negativ).
   • (c) Benutzer gibt eine gültige Anzahl ein (z. B. +25 oder −5 bei ausreichendem Bestand).
   • (d) Benutzer klickt „Änderungen speichern“.
   • (e) System berechnet den neuen Lagerbestand und speichert ihn.
   • (f) Die Artikelliste zeigt den aktualisierten Lagerbestand.
2. **Fehlerfall**:
   • (a) Benutzer gibt eine Dezimalzahl (bspw. 12,5) als Änderung ein, die den Bestand auf eine ungerade Zahl ändern würde.
   • (b) System erkennt den Fehler und zeigt die Meldung „Geben Sie einen gültigen Wert ein. Die nächsten Werte sind 12 und 13“.
   • (c) Benutzer klickt „Änderungen speichern“.
   • (d) System verweigert Speicherung, bisherige Eingaben bleiben sichtbar.

Akzeptanzkriterien:
• Gültige Anpassung liefert HTTP 200 und der neue Bestand wird korrekt gespeichert.
• Ungültige Anpassungen (Dezimalzahlen) liefern HTTP 400 und verhindern Änderungen.
• Formular zeigt nach Fehler weiterhin alle Eingaben.

1.8 Use Case: Artikel löschen

1. **Erfolgsfall**:
   • (a) Benutzer klickt in der Tabelle „Alle Artikel“ auf einen bestimmten Artikel und dann auf „Löschen“.
   • (b) System zeigt einen Bestätigungsdialog „Soll der Artikel „[Name des Artikels]“ (ID: [ID des Artikels]) wirklich gelöscht werden?“.
   • (c) Benutzer bestätigt den Löschvorgang mit „OK“.
   • (d) System löscht den Artikel.
   • (e) Artikelliste aktualisiert sich nach Aktualisierung der Website ohne den gelöschten Eintrag anzuzeigen in der Tabelle „Alle Artikel“.
2. **Fehlerfall**:
   • (a) Benutzer versucht einen Artikel zu löschen, der nicht existiert oder bereits gelöscht wurde.
   • (b) System liefert HTTP 404 „Artikel mit der ID [ID] wurde nicht gefunden“.
   • (c) Liste bleibt unverändert.

Akzeptanzkriterien:
• Erfolgreiches DELETE liefert HTTP 200/204 und entfernt den Datensatz dauerhaft.
• DELETE mit ungültiger ID liefert HTTP 404; keine Änderungen am Datenbestand.
• UI zeigt nach erfolgreichem Löschen die aktualisierte Liste.

---

2. Use Case: Lieferant anlegen

1. **Erfolgsfall**:
   • (a) Benutzer navigiert zu „Lieferanten“ und begibt sich zum Menüpunkt „Neuen Lieferanten anlegen“.
   • (b) System zeigt Felder für Firmenname, Ansprechpartner, E-Mail, Telefon und Adresse an.
   • (c) Benutzer trägt gültige Daten ein.
   • (d) System erkennt alle Eingaben als gültig.
   • (e) Benutzer klickt auf „Lieferant speichern“.
   • (f) System legt den Lieferanten an und zeigt ihn in der Tabelle „Alle Lieferanten“ an.
2. **Fehlerfall**:
   • (a) Benutzer lässt eine der beiden Pflichtfelder (Firmenname oder E-Mail) leer bzw. ungültig.
   • (b) System zeigt Fehlermeldung („Füllen Sie dieses Feld aus.“) und markiert das entsprechende Feld, welches noch ausgefüllt werden muss.
   • (c) Benutzer klickt „Lieferant speichern“.
   • (d) Backend verweigert Speicherung, bisherige Eingaben bleiben sichtbar.

Akzeptanzkriterien:
• Erfolgreicher POST liefert HTTP 201 und gibt den neuen Supplier zurück.
• Ungültige Pflichtfelder führen zu HTTP 400 und verhindern Speicherung.
• UI behält eingegebene Daten bei Validierungsfehlern.

2.1 Use Case: Lieferant bearbeiten

1. **Erfolgsfall**:
   • (a) Benutzer wählt bestehenden Lieferanten aus der Liste „Alle Lieferanten“ aus.
   • (b) System lädt aktuelle Daten in ein Bearbeitungsformular.
   • (c) Benutzer ändert z. B. Telefonnummer oder Adresse und klickt auf „Änderungen speichern“.
   • (e) Backend aktualisiert den Lieferanten.
   • (f) UI zeigt die neuen Werte in der Lieferantenliste „Alle Lieferanten“ an.
2. **Fehlerfall**:
   • (a) Benutzer bearbeitet eine nicht existierende Lieferanten-ID.
   • (b) Backend liefert 404 „Lieferant nicht gefunden“.
   • (c) Formular bleibt sichtbar, keine Speicherung.

Akzeptanzkriterien:
• PUT mit gültigen Änderungen liefert HTTP 200 und aktualisiert den Datensatz.
• PUT mit ungültiger ID liefert HTTP 404; keine Speicherung.
• UI zeigt aktualisierte Daten korrekt an.

---

3. Use Case: Lieferantenbestellung erfassen

1. **Erfolgsfall**:
   • (a) Benutzer öffnet „Bestellwesen“ und begibt sich zum Menüpunkt „Neue Bestellung aufgeben“.
   • (b) System zeigt Felder für „Lieferant auswählen“ und „Bestellpositionen“ (Artikel, Menge, automatisch übernommener EK-Preis/Stück €) an.
   • (c) Benutzer wählt existierenden Lieferanten und fügt mindestens eine gültige Position hinzu mit einer Menge, die auch im Lager vorhanden ist.
   • (d) Benutzer fügt mind. Eine oder mehrere Bestellposition hinzu und sobald seine Bestellung vollständig ist, klickt er auf „Bestellung absenden“.
   • (e) Backend erstellt die Bestellung mit Status „BESTELLT“.
   • (f) Bestellung erscheint in der tabellarischen Übersicht „Offene & Aktuelle Bestellungen“.
2. **Fehlerfall**:
   • (a) Benutzer wählt einen existierenden Artikel, vergisst aber die Quantität („Menge“) beim der Bestellposition mit anzugeben.
   • (b) Frontend liefert die Fehlermeldung „Füllen Sie dieses Feld aus.“.
   • (c) Eingaben bleiben sichtbar, Bestellung wird nicht erzeugt und der Benutzer hat die Chance eine Menge die Bestellt werden soll auszuwählen.

Akzeptanzkriterien:
• Erfolgreicher POST liefert HTTP 201 mit Bestell-ID, Lieferant, Datum, Betrag und aktuellen Status (nach Bestellung zuerst auf „BESTELLT“).
• Fehlende/ungültige Referenzen liefern HTTP 404 und verhindern Speicherung.
• UI zeigt eingegebene Werte nach Fehler weiterhin an.

3.1 Use Case: Status einer Lieferantenbestellung ändern

1. **Erfolgsfall**:
   • (a) Benutzer begibt sich in der Tabelle „Offene & Aktuelle Bestellungen“ im Reiter „Bestellwesen“ zu einer bestehenden Bestellung.
   • (b) System zeigt unter „Aktionen“ die Möglichkeit an, eine „Lieferung [zu] empfangen“.
   • (c) Benutzer wählt diese Option und erhält vom System eine Nachricht „Soll der Bestellauftrag (ID: [ID des Auftrags] als GELIEFERT markiert und der Lagerbestand erhöht werden?“. Diese bestätigt der Benutzer – insofern zutreffend – mit „OK“.
   • (d) Backend speichert den neuen Status.
   • (e) Bestellung wird mit aktualisiertem Status „GELIEFERT“ in der Tabelle „Offene & Aktuelle Bestellungen “angezeigt.
2. **Fehlerfall**:
   • (a) Benutzer wählt „Lieferung empfangen“ für einen nicht zulässigen Statuswechsel. Daraufhin klickt der Benutzer „Abbrechen“.
   • (b) Alter Status bleibt bestehen.

Akzeptanzkriterien:
• Gültiger Statuswechsel liefert HTTP 200 und aktualisiert den Datensatz.
• Ungültiger Statuswechsel liefert HTTP 4xx; keine Änderung im Bestand.
• UI zeigt aktualisierten oder unveränderten Status entsprechend an.

3.3 Use Case: Lieferantenbestellung löschen

1. **Erfolgsfall**:
   • (a) Benutzer begibt sich in der Tabelle „Offene & Aktuelle Bestellungen“ im Reiter „Bestellwesen“ zu einer bestehenden Bestellung. Er markiert eine Bestellung und klickt „Löschen“.
   • (b) System zeigt den Bestätigungsdialog „Soll der Bestellauftrag (ID: [ID des Auftrags]) wirklich gelöscht werden?“
   • (c) Benutzer bestätigt diese Anzeige mit „OK“.
   • (d) Backend löscht die Bestellung.
   • (e) Liste zeigt den Eintrag – auch ohne Aktualisierung der Website - nicht mehr an.
2. **Fehlerfall**:
   • (a) Benutzer versucht, nichtexistierende Bestellung zu löschen.
   • (b) Backend liefert HTTP 404.
   • (c) UI bleibt unverändert.

Akzeptanzkriterien:
• Erfolgreiches DELETE liefert HTTP 200/204.
• DELETE mit ungültiger ID liefert 404; keine Änderungen.
