

Software Engineering 1 | Pflichtenheft Fakturierungssystem |

Team 3 - Gruppe L |

Bestandsmanagement

Datum: 12.06.2026 | Version: 1.0

INHALTSVERZEICHNIS

1. EINLEITUNG UND ZIELBESTIMMUNG	1
1.1 ZIEL DES DOKUMENTS	1
1.2 GELTUNGSBEREICH	2
1.3 DEFINITIONEN UND ABKÜRZUNGEN	2
1.4 REFERENZEN	2
2. SYSTEMÜBERBLICK	2
2.1 KURZÜBERBLICK	3
2.2 ABGRENZUNG (WAS GEHÖRT DAZU/ WAS NICHT)	3
2.3 GROBE SYSTEMFUNKTIONEN	3
2.4 UML- BEZUG	4
2. STAKEHOLDER UND KONTEXT	4
4. FUNKTIONALE ANFORDERUNGEN	4
4.1 ARTIKEL ANLEGEN UND VERWALTEN	5
4.2 ARTIKEL SUCHEN UND AUSWÄHLEN	5
4.3 LAGERBESTANDSVERWALTUNG	5
4.4 LÖSCHEN VON ARTIKELN	5
5. NICHT-FUNKTIONALE ANFORDERUNGEN	5
5.1 DATENSICHERHEIT UND ZUVERLÄSSIGKEIT	6
5.2 BENUTZBARKEIT	6
6. DATEN UND SCHNITTSTELLEN	6
6.1 ARCHITEKTONISCHE DESIGNGRUNDSÄTZE FÜR DATENTYPEN:	6
6.2 GLOBALE STATUS-ENUMERATIONS	6
6.2.1 Klasse: Stammdatenobjekte (Kunde und Produkt)	7
6.2.2 Klasse: Dokumentposition	8
6.2.3 Abstrakte Basisklasse: Dokument	8
6.2.4 Konkrete Spezialisierungen (Vererbung von Dokument)	9
6.4 INTERNE SCHNITTSTELLEN	9
6.5 EXTERNE SCHNITTSTELLEN	10
6.5.1 IF-Satzschablone (Spezifikation nach Standard)	11
7. SYSTEMARCHITEKTUR (LOGISCH, GROB)	11
7.1 KLASSENDIAGRAMM	11
7.2 SEQUENZDIAGRAMM	12
8. TESTBARE ABNAHMEKRITERIEN	13
9. TRACEABILITY: LASTENHEFT (LH) – PFLICHTENHEFT (PH)	14
10. MODULTESTPLAN	16
11. ANHÄNGE	17

11.1 Abkürzungen	17
11.2 Begriffe	18

Freigabeübersicht

Autor	Freigebenden	Prüfer
		Prof. Dr. Marmitt, Gred
Entwickler	Entwickler	Modulverantwortlicher
12.06.2026	12.06.2026	Datum, Unterschrift

Dokumentenhistorie

Version	Datum	Autor	Grund der Änderung
1.0	10.06.2026	Feyza Yaz	Erstellung des Pflichtenhefts und Kapitel 1-3
1.1	10.06.2026	Derin Abdullah, Melike Caliskan	Bearbeitung von Kapitel 4-7
1.2	11.06.2026	Feyza Yaz	Bearbeitung von Kapitel 8
1.3	12.06.2026	Melike Caliskan	Bearbeitung von Kapitel 9
1.4	12.06.2026	Derin Abdullah	Bearbeitung von Kapitel 10
1.5	13.06.2026	Derin Abdullah, Melike Caliskan, Feyza Yaz	Bearbeitung und Fertigstellung des Pflichtenhefts

1. Einleitung und Zielbestimmung

1.1 Ziel des Dokuments

Ziel dieses Pflichtenheft-Teils ist die vollständige und testbare Spezifikation der Bestandsverwaltung. Dies umfasst die persistente lokale Speicherung, Pflege und Bereitstellung von Produktdaten (Bezeichnung, Netto-Einzelpreis, Mehrwertsteuersatz, Bestand).

Zudem spezifiziert dieses Dokument die Durchsetzung von Geschäftsregeln auf Datenbank-/Logikebene, insbesondere die Gewährleistung der referenziellen Integrität

(Löschsperrern gemäß GR-04), sowie die Bereitstellung einer internen Schnittstelle, über die andere Systemkomponenten auf den Produktkatalog zugreifen können.

1.2 Geltungsbereich

Dieses Dokument gilt für die Komponente Gruppe L — Bestandsmanagement. Die Gesamtanwendung wird arbeitsteilig in vier Kernbereichen spezifiziert. Jede Arbeitsgruppe steuert einen eigenen Pflichtenheft-Teil bei, die das Gesamtsystem bilden:

Gruppe	Komponente	Eigenes Pflichtenheft
J	Kundenverwaltung	separat
L	Bestandsmanagement	Dieses Dokument
K	Belegworkflow	separat
I	Benutzeroberfläche	separat

Die Komponente L kapselt ausschließlich die Fachlogik und Datenhaltung für die Produkte (Anforderung BA-02). Sie enthält keine grafischen Oberflächen (GUI), sondern bietet definierte Services (Schnittstellen) an. Diese werden von der Benutzeroberfläche (für Such- und Listenansichten) und dem Belegworkflow (zur Übernahme von Preis-Snapshots in Dokumente gemäß GR-03) aufgerufen. Die Dialogführung und Darstellung der Produkte auf dem Bildschirm werden arbeitsteilig von der UI-Gruppe spezifiziert.

1.3 Definitionen und Abkürzungen

Fachbegriffe (z. B. Fakturierung, Stammdaten, Snapshot-Prinzip, Reversionssicherheit) sind im Glossar des Lastenhefts (§ 8.1 und § 8.2) definiert und gelten für dieses Dokument unverändert. Dokumentspezifische Abkürzungen siehe Kapitel 11.

1.4 Referenzen

- Lastenheft „Fakturierungssystem“, Team 3, Version 1.0, 15.05.2026
- Pflichtenheft-Teil „Kundenverwaltung“ (separat gepflegt)
- Pflichtenheft-Teil „Belegworkflow / Dokumentenzyklus“ (separat gepflegt)
- Pflichtenheft-Teil „Programmoberfläche“ (separat gepflegt)
- Vorlesungsunterlagen Software Engineering 1 (SoSe 2026), insb. Foliensatz „Lasten- und Pflichtenheft“

2. Systemüberblick

2.1 Kurzüberblick

Die Anwendung ist eine Einzelplatz-Stand-Alone-Desktop-Anwendung mit lokaler Datenhaltung (keine Cloud, kein Server). Die Komponente **Bestandsmanagement (Gruppe L)** bildet das datentechnische Fundament für alle Artikel und Produkte

des Systems. Sie stellt keine eigene grafische Benutzeroberfläche bereit, sondern kapselt ausschließlich die Fachlogik und die persistente Datenspeicherung der Produktdaten. Über definierte Service-Schnittstellen macht sie diese Funktionalität für die anderen Systemkomponenten (Benutzeroberfläche und Dokumentenzyklus) zugänglich.

2.2 Abgrenzung (Was gehört dazu/ was nicht)

Im Umfang dieser Komponente:

- Fachlogik und persistente lokale Speicherung der Produktdaten (Produkt-ID, Bezeichnung, Netto-Einzelpreis, Mehrwertsteuersatz, Bestand) gemäß BA-02.
- Fachliche Validierung beim Anlegen und Ändern von Produkten (z. B. Prüfung auf negative Bestände oder Preise).
- Durchsetzung der referenziellen Integrität (Löschsperre nach GR-04): Prüfung und Verweigerung von Löschvorgängen, falls ein Produkt bereits in aktiven oder archivierten Belegen referenziert wird.
- Bereitstellung einer internen Schnittstelle (ProduktService) für Such- und Lesezugriffe (für die GUI-Gruppe) sowie zur Bereitstellung der Preis- und Steuersatz-Snapshots für die Belegerstellung (GR-03).

Nicht im Umfang dieser Komponente:

- Grafische Benutzeroberfläche (GUI) für Formulare, Suchfelder oder Tabellenansichten der Produkte (Darstellung übernimmt die UI-Gruppe).
- Fachlogik und Persistenz der Kundenstammdaten (Kunden-Gruppe).
- Fachlogik des Belegworkflows, Summenberechnungen für Rechnungen und Dokumenten-Speicherung (Dokumenten-Gruppe).
- Netzwerkanbindungen, Cloud-Speicherung oder Mehrbenutzerverwaltung (gemäß Nicht-Zielen des Lastenhefts).

2.3 Grobe Systemfunktionen

Da diese Komponente als Service für andere Module fungiert, laufen die Systemfunktionen nachfolgendem Muster ab: Aufruf durch UI- oder Dokumenten-Komponente (z. B. "Produkt speichern" oder "Produkt suchen") → Durchführung der fachlichen Validierung und Integritätsprüfung (GR-04) → Ausführung der Datenbank-/Dateioperation → Rückgabe des Ergebnisses, des angeforderten Objekts oder Werfen einer Fehlermeldung an die aufrufende Komponente

2.4 UML- Bezug

Ein gemeinsames Use-Case-Diagramm aller Gruppen in der Einleitung des Gesamtdokuments gibt den Überblick über die Akteure und systemweiten Ziele. Die für Gruppe L relevanten und abgedeckten Use Cases sind fachlich: "Produkt anlegen",

"Produkt bearbeiten", "Produkt suchen" und "Produkt löschen", sowie die systeminterne Beteiligung am Use Case "Beleg erstellen" (durch Bereitstellung der Artikeldaten). Die detaillierte logische Architektur und das Datenmodell (Klassendiagramm) dieser Komponente folgen in Kapitel 6 und 7.

2. Stakeholder und Kontext

Stakeholder, Benutzerrollen und der Systemkontext sind im Lastenheft (Kapitel 2 und 3) definiert und gelten für dieses Pflichtenheft unverändert.

Für die Teil-Komponente L (Bestandsmanagement) gilt spezifisch:

Maßgebliche Akteure:

- **Einzelanwender: in** — natürliche Person (Selbstständige: r, Freiberufler: in, Kleinstunternehmer: in).
- *Hinweis für diese Komponente:* Da das Bestandsmanagement ausschließlich die reine Fachlogik und Datenhaltung kapselt, interagiert die Anwender: in nicht direkt mit diesem Modul. Die Bedienung erfolgt indirekt über die Komponente Programmoberfläche (Gruppe D).

Angrenzende Systeme/Komponenten:

- **Intern – Programmoberfläche (Gruppe D):** Ruft die von uns definierten Dienste (ProduktService) auf, um Listen, Produktsuchen und Formulare zum Anlegen von Produkten grafisch darzustellen.
- **Intern – Prozess / Dokumentenzyklus (Gruppe A):** Fragt bei der Belegerstellung (z. B. Angebot oder Rechnung) die aktuellen Produktdaten und Steuersätze aus dem Bestandsmanagement ab, um diese gemäß dem Snapshot-Prinzip (GR-03) unveränderlich im jeweiligen Beleg zu fixieren.
- **Extern – Lokales Dateisystem (IF-02):** Die Komponente L greift zur Umsetzung von IF-02 direkt auf das lokale Dateisystem zu, um die Produktdaten (Bezeichnung, Preis, Bestand) persistent abzuspeichern, sodass die Datenschutzvorgaben (DSGVO) gewahrt bleiben und die Daten ohne Internetverbindung zur Verfügung stehen.

4. Funktionale Anforderungen

4.1 Artikel anlegen und verwalten

F-01: Das System MUSS es der Anwender:in ERMÖGLICHEN, einen Artikel mit Artikelnummer, Bezeichnung, Netto-Einzelpreis, Mehrwertsteuersatz, Lagerbestand und optionaler Beschreibung anzulegen.

F-02: WENN ein Artikel gespeichert wird, DANN MUSS das System die Artikeldaten dauerhaft speichern.

F-03: Das System MUSS es ERMÖGLICHEN, bestehende Artikelstammdaten zu bearbeiten.

F-04: WENN Änderungen an einem Artikel gespeichert werden, DANN MUSS das System die aktualisierten Daten übernehmen und anzeigen.

F-05: Das System MUSS für jeden Artikel eine eindeutige Artikelnummer verwalten.

F-06: Das System MUSS den aktuellen Netto-Einzelpreis, Mehrwertsteuersatz und Lagerbestand eines Artikels anzeigen.

4.2 Artikel suchen und auswählen

F-07: Das System MUSS es ERMÖGLICHEN, Artikel anhand ihrer Artikelnummer oder Bezeichnung zu suchen und auszuwählen.

4.3 Lagerbestandsverwaltung

F-08: WENN der Lagerbestand eines Artikels den Wert 0 erreicht, DANN MUSS das System den Artikel als nicht verfügbar kennzeichnen.

F-09: WENN ein negativer Lagerbestand eingegeben wird, DANN MUSS das System den Speichervorgang VERWEIGERN und eine Fehlermeldung ANZEIGEN.

4.4 Löschen von Artikeln

F-10: Das System MUSS es ERMÖGLICHEN, bestehende Artikel zu löschen.

F-11: WENN ein Artikel gelöscht werden soll UND dieser bereits in einem aktiven oder archivierten Dokument referenziert ist, DANN MUSS das System den Löschvorgang VERWEIGERN und eine Fehlermeldung ANZEIGEN.

5. Nicht-funktionale Anforderungen

5.1 Datensicherheit und Zuverlässigkeit

NF-REL-01: Das System MUSS Artikel-, Preis- und Lagerbestandsdaten dauerhaft speichern, sodass diese nach einem Neustart der Anwendung unverändert verfügbar sind.

NF-SEC-01: Das System MUSS die lokal gespeicherten Artikeldaten so im Dateisystem ablegen, dass sie vor unbefugtem Zugriff geschützt sind.

5.2 Benutzbarkeit

NF-USE-01: Das Anlegen eines neuen Artikels MUSS von einem erstmaligen Anwender:in OHNE EXTERNE HILFE IN WENIGER ALS 3 MINUTEN abgeschlossen werden können.

NF-USE-02: Das System MUSS fehlerhafte Eingaben (z.B. negativer Preis oder negativer Lagerbestand) so kennzeichnen, dass mindestens 80% der Testpersonen den Fehler ohne externe Hilfe korrigieren können

6. Daten und Schnittstellen

6.1 Architektonische Designgrundsätze für Datentypen:

- Geldbeträge & Steuersätze: Werden ausnahmslos als `java.math.BigDecimal` mit einem Scale von 2 bzw. 4 und kaufmännischer Rundung (`RoundingMode.HALF_UP`) geführt. Die Nutzung von `double` oder `float` ist wegen inhärenter Gleitkomma-Rundungsfehler unzulässig.
- Identifikatoren & Belegnummern: Werden als alphanumerische String -Typen mit festem Format, Präfix und führenden Nullen implementiert (z. B. "R-2026-000124"), um Flexibilität für Formatänderungen zu gewährleisten und mathematische Operationen auszuschließen.
- Datumswerte: Werden konsequent über die moderne API als `java.time.LocalDate` abgebildet.
- Mengen: Werden für physikalische Stückzahlen als ganzzahliger Primitivtyp `int` deklariert.

6.2 Globale Status-Enumerations

```
public enum AngebotStatus {  
    OFFEN, UEBERFUEHRT, VERWORFEN  
}  
public enum RechnungStatus {  
    OFFEN, BEZAHLT  
}
```

6.2.1 Klasse: Stammdatenobjekte (Kunde und Produkt)

Attribute	Java-Datentyp	Beschreibung/Validierung
-----------	---------------	--------------------------

Kunde (Komponente Kundenstamm)		
kundenId	String	Eindeutig, fortlaufende Kundennummer (Primärschlüssel). Alphanumerisch.
nameFirmenname	String	Vollständiger Name des Kunden oder Firmenbezeichnung (Pflichtfeld).
anschrift	String	Vollständiger Postenanschrift (Straße, Hausnummer, PLZ, Ort, Land).
steuernummerUStId	String	Steuernummer oder Umsatzsteuer-Identifikationsnummer für Rechnungslegung.
eMail	String	Validierte E-Mail-Adresse
telefon	String	Telefonnummer im internationalen Format.
Produkt (Komponente Produktstamm)		
produktId	String	Eindeutig, fortlaufende Produktnummer (Primärschlüssel).
bestand	int	Aktuell verfügbarer Lagerbestand (darf gemäß F-09 nicht negativ sein).
bezeichnung	String	Standardbezeichnung des Produkts (Pflichtfeld).
nettoEinzelpreis	BigDecimal	Standard-Nettopreis des Produkts (Scale 2, RoundingMode.HALF_UP).
mehrwertsteuersatz	BigDecimal	Standard-Mehrwertsteuersatz als Faktor (z.B. 0.19 oder 0.07). Scale 2.
beschreibung	String	Optionale, detaillierte Produktbeschreibung (nullfähig).

6.2.2 Klasse: Dokumentposition

Repräsentiert eine spezifische Zeile innerhalb eines beliebigen Belegs. Sie realisiert das funktionale Snapshot-Prinzip (GR-03).

Attribut	Java-Datentyp	Beschreibung / Validierung
produktReferenz	String	Fremdschlüssel auf Produkt.produktId .
bezeichnung	String	Snapshot: Kopie der Produktbezeichnung zum Erstellzeitpunkt des Belegs.
menge	int	Stückzahl der geordneten Position. Muss strikt > 0 sein.
einzelpreisNetto	BigDecimal	Snapshot: Der zum Erstellzeitpunkt gültige Netto- Einzelpreis (Scale 2).
steuersatz	BigDecimal	Snapshot: Der zum Erstellzeitpunkt gültige Steuersatz als Faktor (Scale 2).
positionssummeNetto	BigDecimal	Berechnetes Attribut: einzelpreisNetto × menge (Scale 2).

6.2.3 Abstrakte Basisklasse: Dokument

Bündelt die gemeinsamen Attribute und Strukturen aller Entitäten der Dokumentenkette.

Attribut	Java-Datentyp	Beschreibung / Validierung
belegnummer	String	Eindeutiger Belegschlüssel, vom System lückenlos generiert (GR-01).
datum	LocalDate	Erstelldatum des jeweiligen Dokuments.
kundenReferenz	String	Fremdschlüssel auf Kunde.kundenId (Zuweisung des Vertragspartners).
positionen	List<Dokumentposition>	Kollektion der Belegzeilen. Kardinalität: Mindestens 1 Position erforderlich.
summeNetto	BigDecimal	Summe aller Positionssummen: S positionssummeNetto (Scale 2).
summeSteuer	BigDecimal	Summe der Steuerbeträge: S(positionssummeNetto * steuersatz) (Scale 2).
summeBrutto	BigDecimal	Gesamtbetrag des Belegs: summeNetto + summeSteuer (Scale 2).

6.2.4 Konkrete Spezialisierungen (Vererbung von Dokument)

Klasse	Zusätzliche Attribute (Java-Typ)	Beschreibung / Funktionale Kettenreferenz
Angebot	gueltigBis: LocalDate status: AngebotStatus	Enthält eine Bindefrist. Status verwaltet den Lebenszyklus (Kardinalität zu AB =).
Auftragsbestaetigung	angebotNr: String status: DokumentStatus	angebotNr hält die Rückreferenz auf das Angebot (GR-05). Ist optional bei Direkteingabe (Kardinalität zu Lieferschein =).
Lieferschein	Lieferdatum: LocalDate auftragsNr: String	auftragsNr verweist auf die AB. Preisausschluss: Die Positionen spiegeln nur Bezeichnung und Menge wider (keine Preise!).
Rechnung	leistungsdatum: LocalDate zahlungsziel: LocalDate lieferscheinNr: String status: RechnungStatus	lieferscheinNr stellt lückenlose Kette her. Muss zwingend alle rechtlichen Pflichtangaben gem. § 14 UStG enthalten.

6.4 Interne Schnittstellen

Zur softwareseitigen Entkopplung der Systemkomponenten werden folgende Java-Interfaces spezifiziert:

```
package de.fakturierung.komponenten.interfaces;
```

```
import java.time.LocalDate;  
import java.nio.file.Path;  
import java.util.Optional;
```

```
// Lesender und entkoppelter Zugriff auf den Kundenstamm
```

```
public interface KundenService {
```

```
    /**
```

```
     * Sucht einen Kunden anhand seiner ID.
```

```

    * @return Optional mit Kunde, leer falls kundenId nicht existiert (GR-04)
    */
    Optional<Kunde> findeKunde(String kundenId);
}

// Lesender und entkoppelter Zugriff auf den Produktstamm
public interface ProductService {
    /**
     * Sucht ein Produkt anhand seiner ID.
     * @return Optional mit Produkt, leer falls produktId nicht existiert (GR-04)
     */
    Optional<Produkt> findeProdukt(String produktId);
}

// PDF-Generierungskomponente für den Dokumentenexport (IF-03)
public interface PdfExporter {
    /**
     * Exportiert ein konkretes Dokument als PDF in das lokale Dateisystem (IF-02).
     */
    void exportiere(Dokument dokument, Path zielDatei) throws java.io.IOException;
}

// Komponenteninterner Generator zur Erfüllung von GR-01
public interface BelegnummernGenerator {
    /**
     * Liefert die nächste lückenlose, fortlaufende Nummer für den Belegtyp.
     * Berücksichtigt GoBD-Konformität und das aktuelle Geschäftsjahr.
     */
    String naechsteNummer(Belegtyp typ, int jahr);
}

public enum Belegtyp {
    ANGEBOT, AUFTRAGSBESTAETIGUNG, LIEFERSCHEIN, RECHNUNG
}

```

6.5 Externe Schnittstellen

ID	Schnittstelle	Zweck und technischer Rahmen
IF-01	Benutzerschnittstelle (GUI)	Interaktive, grafische Desktop-Oberfläche (z. B. JavaFX) zur vollständigen, validierten Bedienung aller Anwendungsmodule durch den Einzelanwender.

IF-02	Lokales Dateisystem	Persistente Speicherung aller Stamm- und Bewegungsdaten (JSON/XML oder lokale eingebettete DB) sowie strukturierte Dateiablage exportierter PDF-Dokumente.
IF-03	Druck-/Export-Schnittstelle	Engine zur programmatischen Generierung druckbarer PDF-Dokumente auf Basis von Dokumentenobjekten sowie direkter Anbindung an den lokalen Betriebssystem-Drucker.

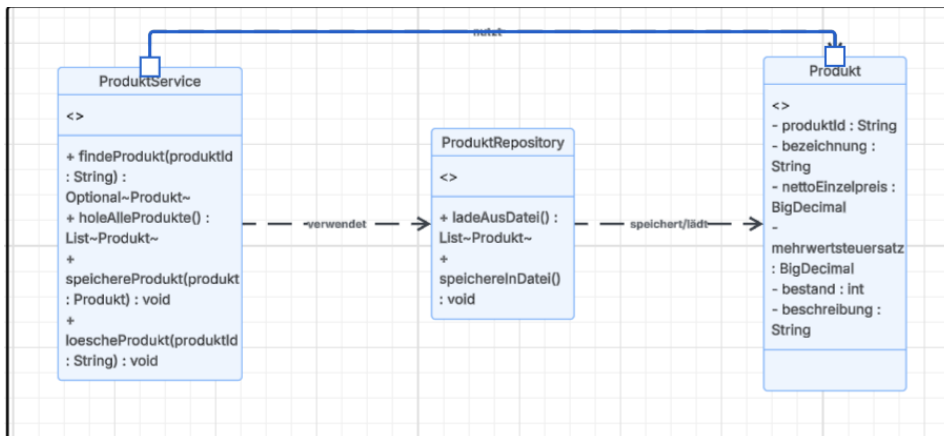
6.5.1 IF-Satzschablone (Spezifikation nach Standard)

- **ID:** IF-02 (Lokales Dateisystem)
- **Schablone:** Das System MUSS eine Datei-Schnittstelle bereitstellen, die es der Persistenzschicht ERMÖGLICHT, auf dem lokalen Dateisystem alle Stamm- und Bewegungsdaten (wie z. B. Produkte) revisionssicher zu persistieren sowie PDF-Exportdokumente strukturiert abzulegen. Die Schnittstelle MUSS standardisierte und plattformunabhängige Dateipfad-Zugriffe (z. B. `java.nio.file.Path`) verwenden.

7. Systemarchitektur (logisch, grob)

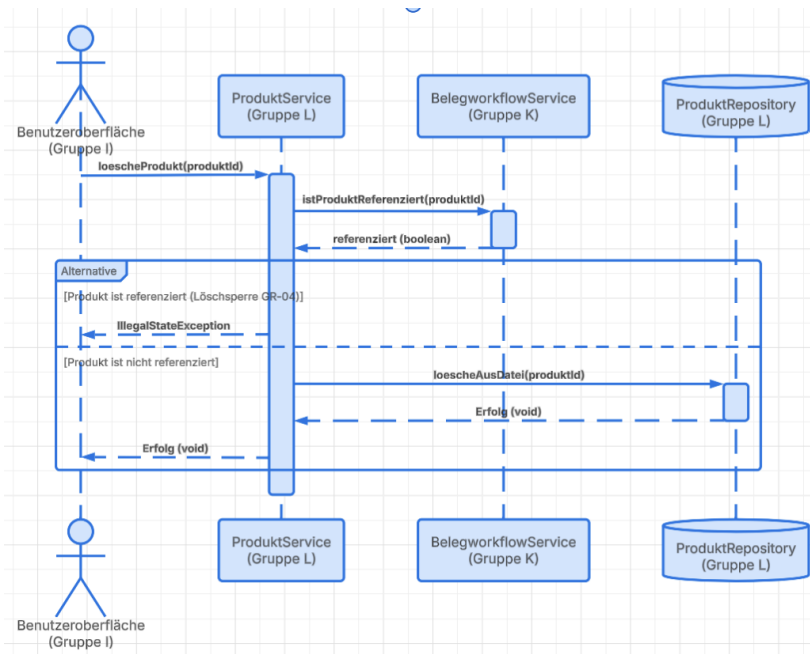
Die Komponente Bestandsmanagement (Gruppe L) kapselt die Fachlogik und die lokale Datenspeicherung für alle Produkte. Sie folgt einer schichtenbasierten Architektur: Ein Serviceinterface (ProduktService) nimmt Anforderungen der Benutzeroberfläche (Gruppe I) und des Belegworkflows (Gruppe K) entgegen. Die Geschäftslogik validiert die Eingaben (z.B. Prüfung auf referenzielle Integrität gemäß GR-04) und gelegiert die dauerhafte Datenspeicherung an eine Persistenzschicht (z.B. ProduktPrpository), welche direkt auf das lokale Dateisystem (IF-02) zugreift. Da die Komponente keine GUI besitzt, existieren hier keine reinen View- oder Controller- Klassen für die Benutzeroberfläche.

7.1 Klassendiagramm



Beschreibung zu Abbildung 1: das Klassendiagramm zeigt das Datenmodell und die Serviceschicht der Produktverwaltung. Die zentrale Fachklasse ist Produkt mit den im Lastenheft definierten Attributen produktId, bezeichnung, nettoEinzelpreis, mehrwertsteuersatz, bestand und beschreibung. Das Interface ProduktService definiert die von außen nutzbaren Methoden zur Manipulation der Daten (z. B. findeProdukt, speichereProdukt, loescheProdukt). Eine technische Persistenz-Klasse (ProduktRepository) ist für die Dateioperationen zuständig und kapselt den lokalen Speicherzugriff (IF-02) auf die Festplatte, getrennt von der Fachlogik.

7.2 Sequenzdiagramm



Beschreibung zu Abbildung 2: Das Sequenzdiagramm stellt den Ablauf beim Löschen eines Produkts dar, bei dem zwingend die Geschäftsregel GR-04 (Referenzielle Integrität) eingehalten werden muss. Die Benutzeroberfläche (Gruppe I) ruft die Methode loescheProdukt(produktId) des ProduktService auf. Der Service prüft zunächst durch einen Aufruf an die Service-Schnittstelle des Belegworkflows (Gruppe K), ob dieses

Produkt bereits in einem aktiven oder archivierten Beleg verwendet wird. Meldet die Komponente Belegworkflow eine Referenz zurück, bricht der ProductService den Vorgang ab und wirft eine Fehlermeldung (z. B. IllegalStateException), um die Löschsperre durchzusetzen. Ist das Produkt nicht referenziert, wird der Löschbefehl an das ProduktRepository delegiert, welches den Datensatz dauerhaft aus dem Dateisystem entfernt.

8. Testbare Abnahmekriterien

Die folgenden Abnahmekriterien überprüfen die korrekte Umsetzung der funktionalen Kernanforderungen der Komponente Bestandsmanagement (Gruppe L). Da diese Komponente keine eigene GUI besitzt, beziehen sich die Aktionen auf die von außen aufrufbaren Service- Schnittstellen (ProductService).

AC-L-01(zu F-01, F-02, NF- REL- 01) - Artikel anlegen und persistent Speichern

- Vorbedingung: Das System (bzw. Die Persistenzschicht) ist betriebsbereit.
- Aktion: Über den ProductService wird ein neues Produkt mit allen Pflichtattributen (inklusive Netto-Einzelpreis, Mehrwertsteuersatz und einem Lagerbestand > 0) zum Speichern übergeben. Das System wird danach simuliert neu gestartet.
- Erwartetes Ergebnis: Das System nimmt das Produkt fehlerfrei an und legt es im lokalen Dateisystem ab (IF-02). Nach dem Neustart ist das Produkt unverändert über die Suchfunktion des ProductService abrufbar.

AC-L-02 (zu F-10, F-11, GR-04) - Artikel anlegen und persistent speichern

- Vorbedingung: Das System (bzw. die Persistenzschicht) ist betriebsbereit.
- Aktion: Über den ProductService wird ein neues Produkt mit allen Pflichtattributen (inklusive Netto-Einzelpreis, Mehrwertsteuersatz und einem Lagerbestand > 0) zum Speichern übergeben. Das System wird danach simuliert neu gestartet.
- Erwartetes Ergebnis: Das System nimmt das Produkt fehlerfrei an und legt es im lokalen Dateisystem ab (IF-02). Nach dem Neustart ist das Produkt unverändert über die Suchfunktion des ProductService abrufbar.

AC-L-02 (zu F-10, F-11, GR-04) - Durchsetzung der Löschsperre (Referenzielle Integrität)

- Vorbedingung: Es existiert ein gespeichertes Produkt im System, welches bereits von der Komponente Belegworkflow (Gruppe K) in einer aktiven oder archivierten Rechnung referenziert wird.
- Aktion: Es wird der Befehl gegeben, dieses Produkt über `loescheProdukt(produktId)` aus dem System zu entfernen.

- Erwartetes Ergebnis: Das System verweigert den Löschvorgang zwingend, wirft eine Fehlermeldung (z. B. IllegalStateException) und das Produkt bleibt im lokalen Dateisystem vollständig erhalten.

AC-L-03 (zu F-06, F-07) - Artikel suchen und anzeigen

- Vorbedingung: Der Datenbestand enthält mindestens 100 Produkte (gemäß Q-01).
- Aktion: Es wird eine Suchanfrage an den ProductService mit einer bekannten Artikelnummer oder Bezeichnung gestellt.
- Erwartetes Ergebnis: Das System liefert den korrekten Datensatz inklusive des aktuellen Netto-Einzelpreises, des Mehrwertsteuersatzes und des Lagerbestands zurück

AC-L-04 (zu F-08, F-09) - Validierung des Lagerbestands

- Vorbedingung: Ein Produkt ist zur Bearbeitung ausgewählt.
- Aktion: Es wird versucht, den Lagerbestand des Produkts auf einen negativen Wert (z. B. -5) zu ändern und zu speichern.
- Erwartetes Ergebnis: Das System verweigert den Speichervorgang, zeigt eine Validierungsfehlermeldung an und der vorherige (gültige) Bestand bleibt unverändert erhalten. Erreicht der Bestand den Wert 0, wird das Produkt zudem korrekt als "nicht verfügbar" gekennzeichnet.

9. Traceability: Lastenheft (LH) – Pflichtenheft (PH)

LH-Anforderung	Beschreibung (LH)	PH-Anforderung(en)
BA-01	Kundendatensätze anlegen und verwalten.	Datenstruktur `Kunde` / Schnittstelle `KundenService` *(Scope Gruppe J)*
BA-02	Produktdaten mit Bezeichnung, Preis und Bestand speichern und in Dokumenten verwenden.	**F-01, F-02, F-03, F-04, F-05, F-06** (Artikelstammdaten und Bestandsverwaltung)
BA-03	Angebot für vorhandenen Kunden mit mindestens einer Produktposition erstellen.	Klasse `Produkt` und Struktur `Dokumentposition` via Schnittstelle `ProductService`
BA-04	Aus einem vorhandenen Angebot eine Rechnung erzeugen (Belegworkflow).	Datenstrukturen `Angebot`, `Rechnung` und `Dokumentposition` im Gesamtdatenmodell

Q-01	Usability: Intuitive Gestaltung, Kernprozesse ohne Schulung in < 10 Min bedienbar.	**NF-USE-01, NF-USE-02** (Eingabefreundlichkeit & interaktive Validierung)
Q-02	Datensicherheit & Zuverlässigkeit: Daten bleiben dauerhaft erhalten und sind vor unbefugtem Zugriff geschützt.	**NF-REL-01, NF-SEC-01** (Sicherung über `ProduktRepository`)
Q-03	Revisionssicherheit (GoBD): Final erzeugte Rechnungen systemseitig vor Manipulation schützen.	Datenmodell-Schreibschutz über Status `RechnungStatus.BEZAHLT`
GR-01	Fortlaufende lückenlose Rechnungsnummern (GoBD-konform).	Interface `BelegnummernGenerator` mit Methode `naechsteNummer(...)`
GR-02	Unveränderlichkeit finaler Belege (Storno-Prinzip).	Dokumentenstatus-Sperren im Klassen- und Sequenzdiagramm
GR-03	Snapshot-Prinzip (Preisübernahme): Beim Hinzufügen einer Produktposition zu einem Dokument werden Einzelpreis und Steuersatz im Dokument eingefroren.	Entität `Dokumentposition` mit entkoppelten Attributen `einzelpreisNetto` und `steuersatz`
GR-04	Referenzielle Integrität: Kunden- und Produktdatensätze dürfen nicht gelöscht werden, wenn sie in aktiven oder archivierten Belegen referenziert sind.	**F-10, F-11** (Löschprüfung über `BelegworkflowService.istProduktReferenziert()`)
GR-05	Dokumentenketten-Konsistenz: Ein Folgedokument übernimmt die Daten des Vorgängerdokuments und verweist über eine eindeutige Referenz auf dieses.	Generalisierung über Basisklasse `Dokument` und UUID-Kettenreferenzen
GR-06	Summenberechnung: Nettosumme = Summe (Menge × Einzelpreis); USt.-Betrag = Nettosumme × Steuersatz; Bruttosumme = Nettosumme + USt.-Betrag.	Berechnungslogik in der Entität `Dokument` mittels Datentyp `BigDecimal`
IF-01	Benutzerschnittstelle (GUI) zur Bedienung des Systems durch den Einzelanwender.	Java-Interface `ProduktService` als Programmschnittstelle zur UI *(Scope Gruppe I)*
IF-02	Lokales Dateisystem: Die Speicherung aller Stamm- und Bewegungsdaten erfolgt persistent im lokalen Dateisystem des Rechners.	Klasse `ProduktRepository` ** mit `ladeAusDatei()` und `speichereInDatei()` via `java.nio.file.Path`
IF-03	Druck-/Export-Schnittstelle zur Erzeugung von Dokumenten (Angebote, Rechnungen) im PDF-Format.	Deklaration des Java-Interface `PdfExporter` mit Methode `exportiere(Dokument, Path)`

10. Modultestplan

TC	Abgedeckte PH-Anforderung	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-L-01	F-01, F-02, F-05	Leeres Repository	speichereProdukt(produkt) mit neuem Produkt (Preis 10.00, Bestand 50)	Produkt wird fehlerfrei gespeichert. Ein anschließendes findeProdukt(id) liefert exakt dieses Produkt zurück.
TC-L-02	F-05 (Fehlerfall)	Produkt "P-100" existiert bereits im Repository	speichereProdukt(produkt) mit einem neuen Produkt, das ebenfalls die ID "P-100" besitzt	Speichern wird verweigert (z. B. IllegalArgumentException), das alte Produkt "P-100" bleibt unverändert.
TC-L-03	F-03, F-04, F-06	Produkt "P-101" existiert (Preis: 10.00)	speichereProdukt(produkt) mit geänderten Werten für "P-101" (neuer Preis: 15.00)	Update ist erfolgreich. Nächster Abruf zeigt den aktualisierten Preis und die aktualisierten Werte.
TC-L-04	F-07	Repository enthält Produkte "P-A" und "P-B"	Aufruf von findeProdukt("P-A")	Das System liefert den korrekten Datensatz für "P-A" zurück; Suche nach unbekannter ID liefert Optional.empty().
TC-L-05	F-09 (Fehlerfall)	Beliebiger Zustand	speichereProdukt(produkt) mit bestand = -5	Speichervorgang wird abgebrochen, System wirft eine IllegalArgumentException.
TC-L-06	F-08	Produkt "P-102" ist in Bearbeitung	speichereProdukt(produkt) mit bestand = 0	Speichern ist erfolgreich; das Produkt wird intern als "nicht verfügbar" gekennzeichnet (Attributabfrage liefert false).
TC-L-07	F-10	Produkt "P-103" existiert. Mock von Gruppe K (istProduktReferenziert) liefert false.	loescheProdukt("P-103")	Löschvorgang ist erfolgreich. Ein anschließendes findeProdukt("P-103") liefert kein Ergebnis.
TC-L-08	F-11 (Löschsperre)	Produkt "P-104" existiert. Mock von Gruppe K	loescheProdukt("P-104")	System wirft zwingend eine IllegalStateException (Löschsperre). Das

		(istProduktReferenziert) liefert true.		Produkt "P-104" bleibt im Repository unverändert erhalten.
TC-L-09	F-06, F-07	Repository enthält exakt 3 verschiedene Produkte	Aufruf von holeAlleProdukte()	Das System liefert eine Liste zurück, die exakt diese 3 Produkte enthält.
TC-L-10	F-01 (Fehlerfall)	Beliebiger Zustand	speichereProdukt(produkt) mit einem ungültigen negativen Netto-Einzelpreis (z.B. -10.00)	Speichervorgang wird abgebrochen, System wirft eine IllegalArgumentException, da Pflichtattribute valide sein müssen.

11. Anhänge

11.1 Abkürzungen

Abkürzung	Bedeutung
F-01 bis F-11	Funktionale Anforderungen
NF	Nicht-funktionale Anforderung
NF-REL	Nicht-funktionale Anforderung – Zuverlässigkeit
NF-SEC	Nicht-funktionale Anforderung – Sicherheit
NF-USE	Nicht-funktionale Anforderung – Benutzbarkeit
IF	Interface / Schnittstelle
USt	Umsatzsteuer
ID	Identifizier (eindeutige Kennung)
GR	Geschäftsregel
PH	Pflichtenheft
LH	Lastenheft
GUI	Graphical User Interface (grafische Benutzeroberfläche)
UML	Unified Modeling Language
DSGVO	Datenschutz-Grundverordnung

11.2 Begriffe

Begriffe	Bedeutung
Produkt-ID	Eindeutige Kennung eines Produkts
Lagerbestand	Aktuell verfügbare Menge eines Produkts
Netto-Einzelpreis	Verkaufspreis ohne Mehrwertsteuer
Mehrwertsteuersatz	Steueranteil eines Produkts (z. B. 19 %)
Persistenz	Dauerhafte Speicherung von Daten
ProduktService	Service-Schnittstelle für Produktzugriffe

ProduktRepository	Komponente zur Speicherung und zum Laden von Produktdaten
Referenzielle Integrität	Produkte dürfen nicht gelöscht werden, wenn sie bereits in Dokumenten verwendet werden
BigDecimal	Java-Datentyp für präzise Geldbeträge
LocalDate	Java-Datentyp zur Speicherung eines Datums