

Software Engineering 1 | Pflichtenheft | Gruppe J |  
Kundenverwaltung

Frakturierungssystem

Datum: 15.06.2026 | Version: 1.0

Weiterleitung zum Git: [https://gitty.informatik.hs-mannheim.de/3028363/SE1\\_Team\\_3](https://gitty.informatik.hs-mannheim.de/3028363/SE1_Team_3)

Inhalt

Dokumentenhistorie ..... 3

1. Einleitung und Zielsetzung..... 3

    1.1 Zweck des Dokuments..... 3

    1.2 Ziel ..... 3

    1.3 Geltungsbereich ..... 4

    1.4 Definitionen und Abkürzungen ..... 4

    1.5 Referenzen..... 4

2. Systemüberblick..... 5

    2.1 Beschreibung..... 5

    2.2 Abgrenzung (Was gehört dazu / was nicht) ..... 5

    2.3 Grobe Systemfunktionen..... 5

    2.4 UML-Bezug ..... 6

3. Stakeholder und Kontext ..... 6

    3.2 Akteur..... 6

    3.3 Angrenzende Komponente..... 6

4. Funktionale Anforderungen..... 7

    4.1 Kunden anlegen (aus BA-01) ..... 7

    4.2 Kunde bearbeiten (aus BA-01) ..... 7

    4.3 Kunde abrufen (aus BA-01) ..... 8

    4.4 Kunde löschen (aus BA-01, GR-04)..... 8

    4.5 Transaktionshistorie (aus BA-01)..... 8

5. Nicht-funktionale Anforderungen..... 8

    5.1 Benutzerfreundlichkeit (aus Q-01) ..... 8

    5.2 Datensicherheit und Zuverlässigkeit (aus Q-02) ..... 8

    5.3 Revisionssicherheit (aus Q-03) ..... 9

    5.4 Performance..... 9

6. Daten und Schnittstellen..... 9

6.1 Datenobjekte und Datentypen.....	9
6.2 Schnittstellen.....	10
6.3 Geschäftsregeln mit Datenbezug .....	11
7. Systemarchitektur (logisch, grob) .....	12
7.1 Klassendiagramm .....	12
7.2 Sequenzdiagramm.....	13
8. Testbare Abnahmekriterien .....	13
9. Traceability LH <-> PH .....	16
10. Modultestplan .....	17
11. Anhänge.....	18
11.1 Abkürzungen .....	18
11.2 Glossar.....	19

## Freigabeübersicht

Autor	Freigebenden	Prüfer
Khazanovych, Christian Winkler, Louis Senger, Robin	Prof. Dr. Marmitt, Gerd	Prof. Dr. Marmitt, Gerd
Gruppe J	Modulverantwortlicher	Modulverantwortlicher
15.06.2026	15.06.2026	15.06.2026

## Dokumentenhistorie

Version	Datum	Autor	Grund der Änderung
1.0	15.06.2026	Khazanovych, Christian Winkler, Louis Senger, Robin	Erstmalige Erstellung des Pflichtenhefts

# 1. Einleitung und Zielsetzung

## 1.1 Zweck des Dokuments

Dieses Dokument beschreibt auf technischer Ebene, wie die Anforderungen aus dem Lastenheft (Team 3, Version 1.4) für den Bereich Kundenverwaltung konkret umgesetzt werden. Während das Lastenheft festhält, was das System leisten soll, gibt dieses Pflichtenheft die Antwort auf das wie. Die hier festgelegten Systemanforderungen sind direkt prüfbar und fließen unmittelbar in die Implementierung sowie den abschließenden Modultest ein.

## 1.2 Ziel

Im Fokus steht die lückenlose Spezifikation aller Funktionen rund um die Verwaltung von Kundendaten, das Anlegen neuer Datensätze, das Bearbeiten bestehender Einträge, das Abrufen gespeicherter Informationen sowie das regelkonforme Löschen unter Berücksichtigung der Geschäftsregel GR-04 (referenzielle Integrität). Darüber hinaus wird definiert, wie diese Daten anderen Komponente, insbesondere dem Belegworkflow (Gruppe K), zugänglich gemacht werden, da jedes kaufmännische Dokument zwingend auf einem gültigen Kundendatensatz beruht.

## 1.3 Geltungsbereich

Dieses Pflichtenheft bezieht sich ausschließlich auf die Komponente Gruppe J – Kundenverwaltung. Die Gesamtanwendung wird in vier unabhängigen Teilbereiche entwickelt, für die jeweils ein eigenes Pflichtenheft existiert:

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

Themen wie die Produktverwaltung, der Aufbau der Benutzeroberfläche oder die Logik zur Beleggenerierung werden hier nicht behandelt. Die Komponente J stellt ihre Daten über eine klar definierte interne Schnittstelle bereit (siehe Kapitel 6.2), greift auf andere Komponente jedoch nicht aktiv zu.

## 1.4 Definitionen und Abkürzungen

Alle im Lastenheft (Abschnitt 8) eingeführten Begriffe und Kürzel gelten unverändert weiter und werden an dieser Stelle nicht wiederholt. Abkürzungen, die spezifisch für dieses Pflichtenheft sind, finden sich gesammelt in Kapitel 11.

## 1.5 Referenzen

Dokument / Quelle	Details & Version	Datum
Projekt Charter	Team 3, Version 1.2	15.04.2026
Lastenheft „Fakturierungsanwendung“	Team 3, Version 1.4	15.05.2026
Pflichtenheft	Gruppe J, Version 1.0	15.06.2026
Vorlesungsunterlagen Software Engineering 1	Hochschule Mannheim, Folienblock „Lasten- und Pflichtenheft“	Sommersemester 2026
GoBD	Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern in elektronischer Form	i. d. g. F. (in der geltenden Fassung)
DSGVO	Verordnung (EU) 2016/679 (Datenschutz- Grundverordnung)	i. d. g. F.
§ 14 UstG	Pflichtangabe einer Rechnung	i. d. g. F.

## 2. Systemüberblick

### 2.1 Beschreibung

Die Kundenverwaltung ist eine von vier Komponente der Fakturierungsanwendung und bildet das datentechnische Fundament des Gesamtsystems. Sie läuft vollständig lokal auf dem Rechner des Anwenders, benötigt keine Internetverbindung und speichert alle Daten auf dem lokalen Dateisystem. Über die grafische Oberfläche (Gruppe I) kann der Anwender Kundendatensätze anlegen, einsehen, bearbeiten und sofern keine aktiven Dokumentenreferenzen bestehen auch löschen. Andere Komponenten, insbesondere der Belegworkflow (Gruppe K), greifend lesend auf die hier verwalteten Daten zu.

### 2.2 Abgrenzung (Was gehört dazu / was nicht)

Im Umfang dieser Anwendung (In-Scope):

- Anlegen neuer Kundendatensätze mit allen relevanten Stammdaten
- Bearbeiten und Abrufen bestehender Einträge
- Löschen von Kundendatensätzen unter Prüfung referenzieller Integrität (GR-04)
- Bereitstellung der Kundendaten über eine interne Schnittstelle für andere Komponenten
- Anzeigen einer vollständigen Transaktionshistorie je Kunde

Explizit nicht im Umfang der Anwendung (Out-of-Scope / Nicht-Ziele):

- Verwaltung von Produkten oder Lagerbeständen (Gruppe L)
- Erstellung oder Verwaltung von Belegen jeglicher Art (Gruppe K)
- Aufbau und Layout der grafischen Oberfläche (Gruppe I)
- Cloud-Synchronisation, Netzwerkzugriff oder externe Datenbankanbindung (Lastenheft, Abschnitt 1.3)

### 2.3 Grobe Systemfunktionen

Der grundlegende Ablauf innerhalb der Komponente lässt sich wie folgt zusammenfassen:

Kunde anlegen -> Pflichtfelder validieren -> Datensatz persistieren -> In Übersicht anzeigen

Kunden bearbeiten -> Bestehenden Datensatz laden -> Änderungen validieren -> aktualisiert speichern

Kunde löschen -> Referenzprüfung (GR-04) -> Bei Freigabe löschen, sonst Hinweis ausgeben

Kundendaten abrufen -> Über interne Schnittstelle an anfragende Komponente zurückgeben

## 2.4 UML-Bezug

Die für diese Komponente relevanten Use Cases sind: Kunde anlegen, Kunde bearbeiten, Kunde löschen und Kundendaten abrufen. Das detaillierte Klassendiagramm sowie ein Sequenzdiagramm für den Ablauf „Kunden anlegen“ folgen in Kapitel 7.

## 3. Stakeholder und Kontext

Stakeholder und Systemkontext sind im Lastenheft (Abschnitt 2 und 3) grundlegend beschrieben und behalten dort ihre Gültigkeit. Für die Komponente Kundenverwaltung sind folgende Parteien besonders relevant:

ID	Stakeholder	Relevanz für diese Komponente
SH-01	Auftraggeber (Prof. Dr. Marmitt)	Definiert die fachlichen Anforderungen und bewertet die Umsetzung anhand der Abnahmekriterien
SH-02	Projektteam (Team 3)	Verantwortlich für die technische Umsetzung, andere Gruppe sind auf korrekte Kundendaten angewiesen
SH-03	Endnutzer	Nutzt die Kundenverwaltung direkt, um Datensätze zu erfassen und für Belege bereitzustellen

### 3.2 Akteur

Da die Anwendung explizit für den Einzelplatzbetrieb konzipiert ist, gibt es genau einen handelnden Akteur in dieser Komponente:

Einzelanwender:in – Die natürliche Person (z.B. Freiberufler:in oder Inhaber:in eines Kleinstunternehmens), die Kundendaten eigenverantwortlich erfasst, pflegt und für die Belegausstellung verwendet

### 3.3 Angrenzende Komponente

Die Kundenverwaltung steht mit folgenden Teilen des Gesamtsystems in Beziehung:

Komponente	Gruppe	Art der Beziehung
Benutzeroberfläche	I	Gruppe I stellt die GUI bereit, über die der Anwender mit der

		Kundenverwaltung interagiert
Belegworkflow	K	Gruppe K greift lesend auf Kundendaten zu, um Belege korrekt adressieren zu können
Bestandsmanagement	L	Keine direkte Abhängigkeit, beide Komponente liefern unabhängig voneinander Stammdaten
Lokales Dateisystem	-	Persistente Speicherung aller Kundendatensätze auf dem Rechner des Anwenders

## 4. Funktionale Anforderungen

Die funktionale Anforderungen beschreiben auf Systemebene, wie die Kundenverwaltung die im Lastenheft festgelegten Anforderungen BA-01 technisch umsetzt.

### 4.1 Kunden anlegen (aus BA-01)

- F-01: Das System MUSS es der Anwender:in ERMÖGLICHEN, einen neuen Kundendatensatz mit folgenden Pflichtfeldern anzulegen: Name oder Firmenname, vollständige Anschrift, Steuernummer oder USt-IdNr., E-Mail-Adresse und Telefonnummer.
- F-02: WENN ein neuer Kundendatensatz gespeichert wird, DANN MUSS das System automatisch eine eindeutige, fortlaufende Kunden-ID vergeben, die nachträglich nicht verändert werden kann.
- F-03: WENN ein Pflichtfeld beim Speichern leer ist, DANN MUSS das System den Speichervorgang ablehnen und dem Anwender klar anzeigen, welches Feld fehlt.
- F-04: WENN ein Kundendatensatz erfolgreich gespeichert wurde, DANN MUSS der Kunde unmittelbar in der Kundenübersicht erscheinen und für andere Komponenten (insbesondere den Belegworkflow, Gruppe K) abrufbar sein.

### 4.2 Kunde bearbeiten (aus BA-01)

- F-05: Das System MUSS es der Anwender:in ERMÖGLICHEN, alle Felder eines bestehenden Kundendatensatzes, mit Ausnahme der Kunden-ID, nachträglich zu bearbeiten und zu speichern.
- F-06: WENN eine Änderung an einem Kundendatensatz gespeichert wird, DANN MUSS das System sicherstellen, dass alle Pflichtfelder weiterhin befüllt sind, bevor die Aktualisierung übernommen wird.

- F-07: WENN ein Kundendatensatz bearbeitet wurde, DANN MUSS die aktualisierte Version in der Kundenübersicht und über die interne Schnittstelle sofort verfügbar sein, ohne dass ein Neustart der Anwendung erforderlich ist.

### 4.3 Kunde abrufen (aus BA-01)

- F-08: Das System MUSS es der Anwender:in ERMÖGLICHEN, die vollständige Kundenliste in einer übersichtlichen Listenansicht einzusehen.
- F-09: Das System MUSS eine Suchfunktion bereitstellen, mit der Kundendatensätze anhand von Name oder Kunden-ID gefunden werden können.
- F-10: Das System MUSS über eine interne Schnittstelle (KundenService) anderen Komponenten den lesenden Zugriff auf einzelne Kundendatensätze per Kunden-ID ermöglichen, ohne dass diese Komponenten direkten Zugriff auf die Datenhaltung benötigen.

### 4.4 Kunde löschen (aus BA-01, GR-04)

- F-11: Das System MUSS es der Anwender:in ERMÖGLICHEN, einen Kundendatensatz zu löschen, sofern dieser in keinem aktiven oder archivierten Beleg referenziert wird.
- F-12: WENN ein Lösversuch für einen Kunden ausgeführt wird, der in mindestens einem Beleg referenziert ist, DANN MUSS das System den Löschvorgang verweigern und eine verständliche Fehlermeldung ausgeben, die den Grund der Ablehnung benennt (GR-04).

### 4.5 Transaktionshistorie (aus BA-01)

- F-13: Das System MUSS je Kundendatensatz eine lückenlose Übersicht aller verknüpften Belege (Angebote und Rechnungen) anzeigen können, sodass die gesamte Geschäftsbeziehung nachvollziehbar bleibt.

## 5. Nicht-funktionale Anforderungen

### 5.1 Benutzerfreundlichkeit (aus Q-01)

- NF-USE-01: Das Anlegen eines neuen Kundendatensatzes MUSS von einer Person ohne technische Vorkenntnisse und ohne vorherige Einweisung innerhalb von maximal 3 Minuten erfolgreich abgeschlossen werden können. Der Nachweis erfolgt durch einen manuellen Usability-Test mit mindestens 5 Testpersonen.
- NF-USE-02: WENN ein Pflichtfeld nicht ausgefüllt wurde, MUSS das System die fehlerhafte oder fehlende Eingabe so hervorheben, dass mindestens 80 % der Testpersonen die Korrektur im ersten Versuch ohne fremde Hilfe vornehmen können.

### 5.2 Datensicherheit und Zuverlässigkeit (aus Q-02)

- NF-SEC-01: Das System MUSS sicherstellen, dass alle gespeicherten Kundendatensätze nach einem Neustart der Anwendung vollständig und unverändert wieder zur Verfügung stehen. Ein Datenverlust durch reguläres Beenden der Anwendung ist nicht zulässig.

- NF-SEC-02: Die Datenhaltung MUSS ausschließlich lokal auf dem Dateisystem des Anwenders erfolgen. Eine Übertragung von Kundendaten an externe Dienste oder Dritte ist systemseitig nicht vorgesehen und technisch auszuschließen (DSGVO-Konformität).

### 5.3 Revisionsicherheit (aus Q-03)

- NF-INT-01: Kundendatensätze, die in einem finalisierten Beleg referenziert werden, MÜSSEN vor versehentlichem Löschen geschützt sein. Das System MUSS einen entsprechenden Löschversuch erkennen und aktiv verweigern, ohne dass Daten verloren gehen (GR-04).

### 5.4 Performance

- NF-PERF-01: Suchanfragen innerhalb der Kundenliste MÜSSEN bei einem Datenbestand von bis zu 5.000 Kundendatensätzen in unter 1 Sekunde ein Ergebnis liefern.
- NF-PERF-02: Das Speichern sowie das Laden eines Kundendatensatzes MUSS in unter 2 Sekunden abgeschlossen sein, unabhängig von der Gesamtgröße des Datenbestands.

## 6. Daten und Schnittstellen

Dieses Kapitel bildet die technische Grundlage für den Modultestplan (Kapitel 10). Alle Datentypen werden bereits als Java-Typen angegeben, da das Pflichtenheft direkt Input für die Implementierung und den Komponententest ist.

### 6.1 Datenobjekte und Datentypen

Designgrundsätze zur Datentyp-Wahl:

- Kundennummer werden als `String` geführt, da sie ein festes Format mit führenden Nullen und Präfix besitzen. Ein ganzzahliger Typ würde führende Nullen verlieren
- Alle Pflichtfelder werden als `String` geführt
- Die Kunden-ID wird vom System automatisch vergeben und ist unveränderlich

Klasse: Kunde

Repräsentiert eine einzelne Position innerhalb eines kaufmännischen Dokuments

Attribut	Java-Typ	Beschreibung
kundenId	String	Eindeutige, vom System generierte Kundennummer, unveränderlich nach Erstellung
name	String	Name der natürlichen Person oder Firmenname, Pflichtfeld

strasse	String	Straße und Hausnummer, Pflichtfeld
plz	String	Postleitzahl, Pflichtfeld
ort	String	Wohnort oder Firmensitz, Pflichtfeld
steuerNummer	String	Steuernummer oder Ust-IdNr. Gemäß § 14 UstG, Pflichtfeld
email	String	E-Mail-Adresse des Kunden, Pflichtfeld
telefon	String	Telefonnummer des Kunden, Pflichtfeld
belegReferenzen	List<String>	Liste der Belegnummern, in denen dieser Kunde referenziert wird (für GR-04)

Enum: KundenStatus

Enum KundenStatus {AKTIV, ARCHIVIERT}

Wert	Bedeutung
AKTIV	Kundendatensatz ist aktiv und kann für neue Belege verwendet werden
ARCHIVIERT	Kunde ist nicht mehr aktiv, bestehende Belegreferenzen bleiben erhalten

## 6.2 Schnittstellen

Externe Schnittstellen:

ID	Schnittstellen	Zweck
IF-01	Lokales Dateisystem	Persistente Speicherung aller Kundendatensätze auf dem Rechner des Anwenders
IF-02	Benutzeroberfläche (Gruppe 1)	Darstellung der Kundenliste und Eingabemasken für den Anwender

Interne Schnittstellen (zu anderen Komponenten):

// Lesender Zugriff auf Kundenstammdaten, wird von Gruppe K (Belegworkflow) genutzt

```
public interface KundenService {  
    // Gibt einen einzelnen Kundendatensatz anhand der Kundennummer zurück  
    // Gibt null zurück, wenn kein Kunde mit dieser Nummer existiert  
    Kunde findeKunde(String kundenId);  
  
    // Gibt eine Liste aller gespeicherten Kunden zurück  
    List<Kunde> alleKunden();  
  
    // Gibt alle Kunden zurück, deren Name den Suchbegriff enthält  
    List<Kunde> sucheNachName(String suchbegriff);  
}
```

Kundennummer-Schnittstelle (komponenteninterner Dienst):

// Generiert die nächste eindeutige, lückenlose Kundennummer

```
public interface KundenIdGenerator {  
    // Liefert die nächste verfügbare Kundennummer im Format K-XXXXXX  
    String naechsteKundenId();  
}
```

### 6.3 Geschäftsregeln mit Datenbezug

Folgende Geschäftsregeln aus dem Lastenheft haben direkten Einfluss auf die Datenhaltung und müssen auf Implementierungsebene berücksichtigt werden:

Regel	Auswirkung auf die Daten
GR-04 (Referenzielle Integrität)	Von jedem Löschvorgang wird belegReferenz geprüft. Ist die Liste nicht leer, wird der Löschvorgang verweigert
Q-02 (Datenpersistenz)	Jede Änderung an einem Kundendatensatz wird sofort auf dem lokalen Dateisystem gespeichert, nicht erst beim Beenden der Anwendung

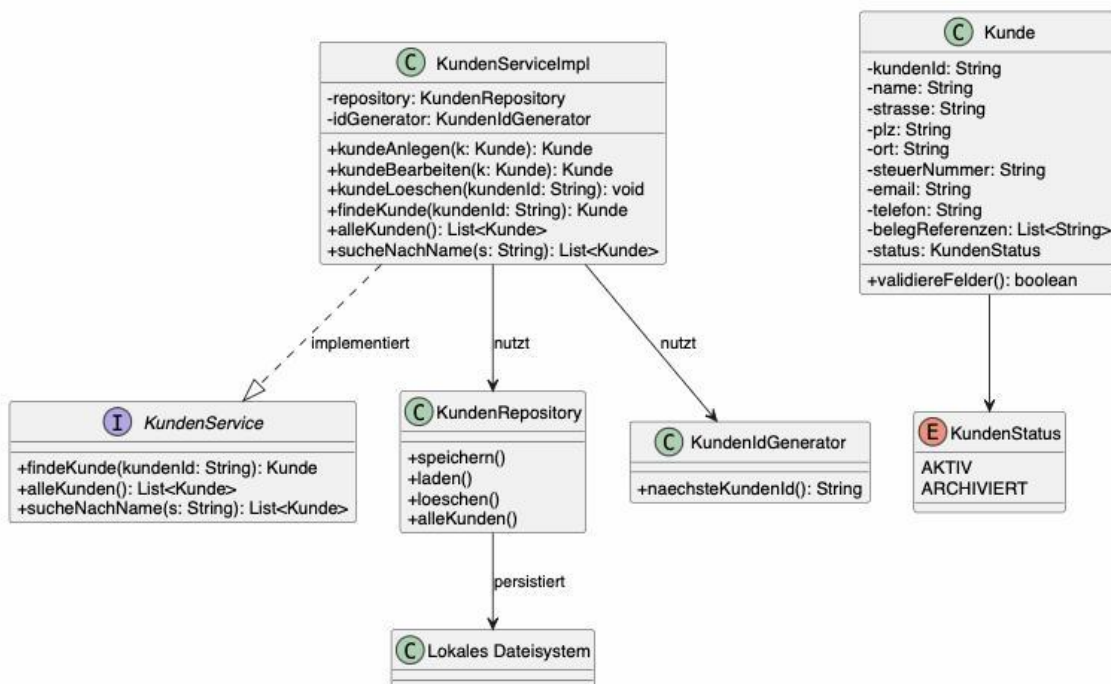
## 7. Systemarchitektur (logisch, grob)

Die Komponente Kundenverwaltung folgt einer klar geschichteten Struktur. Die Benutzeroberfläche (Gruppe I) kommuniziert ausschließlich über den `KundenService` mit der Fachlogik. Der `KundenService` kapselt alle Operationen rund um Kundendaten und nutzt dabei den `KundenIdGenerator` für die automatische Nummernvergabe sowie das `KundenRepository` für die persistente Speicherung auf dem lokalen Dateisystem. Andere Komponente, insbesondere der Belegworkflow (Gruppe K), greifen ebenfalls nur über das `KundenService`-Interface lesend auf Kundendaten zu, ohne direkten Zugriff auf die Datenhaltung zu haben.

### 7.1 Klassendiagramm

Abbildung 1: UML-Klassendiagramm Kundenverwaltung (Gruppe J)

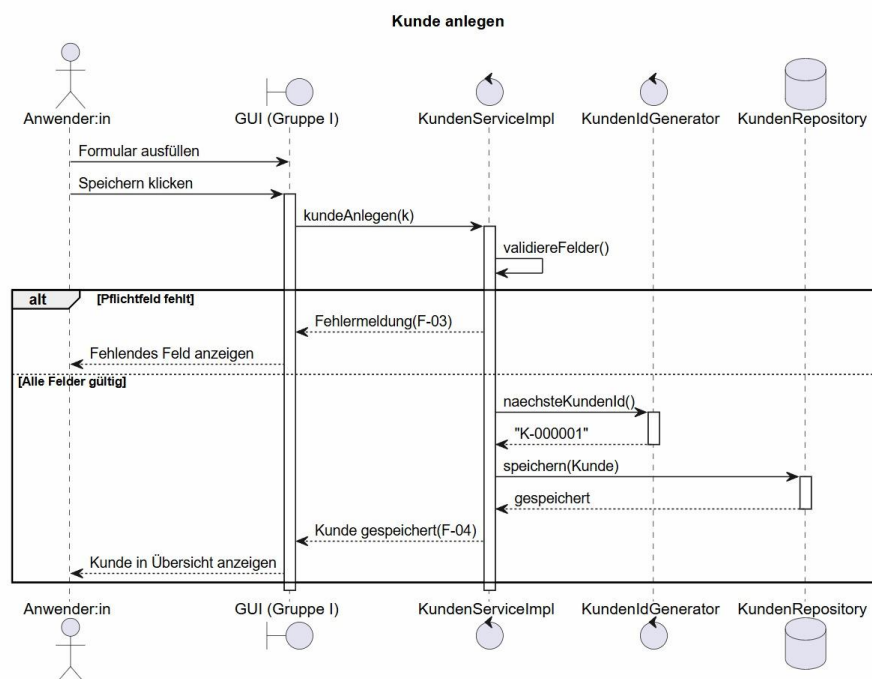
Beschreibung zu Abbildung 1: Das Klassendiagramm zeigt den strukturellen Aufbau der Kundenverwaltung. Die zentrale Klasse `Kunde` hält alle relevanten Stammdaten sowie eine Liste von Belegreferenzen, die für die Prüfung der referenziellen Integrität (GR-04) benötigt wird. Der `KundenStatus` wird als Enum abgebildet und unterscheidet zwischen aktiven und archivierten Kunden. Die Klasse `KundenServiceImp1` implementiert das Interface `KundenService` und stellt damit sowohl der Benutzeroberfläche als auch anderen Komponenten eine einheitliche Zugriffsebene bereit. Intern nutzt sie das `KundenRepository` für alles Lese- und Schreibzugriffe auf das lokale Dateisystem sowie den `KundenIdGenerator` für die automatische, lückenlose Vergabe von Kundennummern.



## 7.2 Sequenzdiagramm

Abbildung 2: UML-Sequenzdiagramm „Kunde anlegen“ (Gruppe J)

Beschreibung zu Abbildung 2: Das Sequenzdiagramm zeigt den vollständigen Ablauf beim Anlegen eines neuen Kunden. Die Anwender:in füllt das Formular über die GUI (Gruppe I) aus und löst den Speichervorgang aus. Die GUI delegiert den Aufruf an `KundenServiceImpl`, welche zunächst alle Pflichtfelder über `validiereFelder()` prüft. Schlägt die Validierung fehl, wird eine Fehlermeldung an die GUI zurückgegeben, die dem Anwender das fehlende Feld benennt (F-03). Sind alle Felder gültig, wird über den `KundenIdGenerator` eine neue, eindeutige Kundennummer angefordert (F-02). Anschließend wird der vollständige Datensatz über das `KundenRepository` auf dem lokalen Dateisystem persistiert. Nach erfolgreicher Speicherung erscheint der neue Kunde unmittelbar in der Kundenübersicht (F-04).



## 8. Testbare Abnahmekriterien

Die folgende Abnahmekriterien leiten sich direkt aus den funktionalen und nicht-funktionalen Anforderungen ab. Sie sind so formuliert, dass sie als Grundlage für den Modultestplan (Kapitel 10) dienen und eindeutig nachgewiesen werden können.

AC-J-01 (zu F-01, F-02, F-04) – Kunde erfolgreich anlegen

Vorbedingung: Die Anwendung ist gestartet, es sind noch keine Kunden vorhanden.

Aktion: Die Anwender:in füllt alle Pflichtfelder (Name, Anschrift, Steuernummer, E-Mail, Telefon) aus und speichert den Datensatz.

Erwartet: Der Kunde wird mit einer automatisch vergebenen, eindeutigen Kunden-ID im Format „K-XXXXXX" gespeichert und erscheint unmittelbar in der Kundenübersicht.

#### AC-J-02 (zu F-03, NF-USE-02) – Speichern bei fehlendem Pflichtfeld verweigern

Vorbedingung: Die Anwendung ist gestartet, das Formular zum Anlegen eines Kunden ist geöffnet.

Aktion: Die Anwender:in lässt mindestens ein Pflichtfeld leer und versucht zu speichern.

Erwartet: Der Speichervorgang wird abgelehnt. Das System hebt das fehlende Feld visuell hervor und benennt es namentlich, sodass die Korrektur ohne fremde Hilfe vorgenommen werden kann.

#### AC-J-03 (zu F-05, F-06, F-07) – Kundendaten bearbeiten

Vorbedingung: Mindestens ein Kundendatensatz ist gespeichert.

Aktion: Die Anwender:in öffnet den Datensatz, ändert die E-Mail-Adresse und speichert die Änderung.

Erwartet: Die aktualisierte E-Mail-Adresse ist sofort in der Kundenübersicht sowie über den KundenService abrufbar, ohne dass ein Neustart der Anwendung notwendig ist.

#### AC-J-04 (zu F-11, F-12, GR-04) - Löschen eines referenzierten Kunden verweigern

Vorbedingung: Ein Kunde ist gespeichert und wird in mindestens einem Beleg referenziert.

Aktion: Die Anwender:in versucht, diesen Kunden zu löschen.

Erwartet: Der Löschvorgang wird verweigert. Das System gibt eine verständliche Fehlermeldung aus, die den Grund der Ablehnung klar benennt. Der Kundendatensatz bleibt vollständig erhalten.

#### AC-J-05 (zu F-11) - Löschen eines nicht referenzierten Kunden

Vorbedingung: Ein Kunde ist gespeichert und wird in keinem Beleg referenziert.

Aktion: Die Anwender:in löscht diesen Kunden.

Erwartet: Der Datensatz wird vollständig aus der Kundenliste entfernt und ist auch über den KundenService nicht mehr abrufbar.

### AC-J-06 (zu F-08, F-09) - Suche nach Kunden

Vorbedingung: Mindestens 3 Kundendatensätze mit unterschiedlichen Namen sind gespeichert.

Aktion: Die Anwender:in gibt einen Namensbestandteil in das Suchfeld ein.

Erwartet: Nur die Kunden, deren Name den Suchbegriff enthält, werden in der Ergebnisliste angezeigt. Kunden ohne Übereinstimmung werden nicht angezeigt.

### AC-J-07 (zu F-10) - Interner Zugriff über KundenService

Vorbedingung: Ein Kunde mit der ID „K-000001“ ist gespeichert.

Aktion: Eine andere Komponente ruft `findeKunde("K-000001")` über das KundenService-Interface auf.

Erwartet: Das System gibt den vollständigen Kundendatensatz zurück. Bei einer nicht existierenden ID wird `null` zurückgegeben.

### AC-J-08 (zu NF-SEC-01, Q-02) - Datenpersistenz nach Neustart

Vorbedingung: Mehrere Kundendatensätze sind gespeichert.

Aktion: Die Anwendung wird vollständig geschlossen und anschließend neu gestartet.

Erwartet: Alle zuvor gespeicherten Kundendatensätze stehen nach dem Neustart vollständig und unverändert zur Verfügung.

### AC-J-09 (zu F-13) - Transaktionshistorie je Kunde

Vorbedingung: Ein Kunde ist in mindestens zwei Belegen referenziert.

Aktion: Die Anwender:in öffnet den Kundendatensatz und ruft die Transaktionshistorie ab.

Erwartet: Alle verknüpften Belegnummern werden vollständig und korrekt angezeigt.

### AC-J-10 (zu NF-USE-01, Q-01) - Usability-Test Kunde anlegen

Vorbedingung: Eine Testperson ohne Vorkenntnisse öffnet die Anwendung zum ersten Mal.

Aktion: Die Testperson legt ohne externe Hilfe einen vollständigen Kundendatensatz an.

Erwartet: Der Vorgang wird in unter 3 Minuten erfolgreich abgeschlossen. Der Test wird mit mindestens 5 Testpersonen wiederholt, von denen alle das Ziel innerhalb der Zeitvorgabe erreichen.

## 9. Traceability LH <-> PH

Jede für Gruppe J relevanten Lastenheft-Anforderungen ist mindestens einer Pflichtenheft-Anforderung zugeordnet. Damit wird sichergestellt, dass keine Anforderung aus dem Lastenheft in der Umsetzung verloren geht.

LH-Anforderung	Beschreibung (LH)	PH-Anforderung/en
BA-01	Kunde anlegen mit Namen, Anschrift und Kontaktinformationen	F-01, F-02, F-03, F-04
BA-01	Kunde bearbeiten	F-05, F-06, F-07
BA-01	Kunden abrufen und suchen	F-08, F-09, F-10
BA-01	Kunden löschen unter Berücksichtigung referenzieller Integrität	F-11, F-12
BA-01	Transaktionshistorie je Kunde	F-13
GR-04	Löschsperre für referenzierte Kundendatensätze	F-12, NF-INT-01
Q-01	Benutzerfreundlichkeit: Kernprozesse ohne Einarbeitung bedienbar	NF-USE-01, NF-USE-02
Q-02	Datensicherheit und Zuverlässigkeit: Persistente lokale Speicherung	NF-SEC-01, NF-SEC-02
Q-03	Revisionssicherheit: Schutz referenzierter Datensätze vor Löschung	NF-INT-01

### Hinweis zur Abgrenzung

Die Geschäftsregeln GR-01 bis GR-03 sowie GR-05 und GR-06 betreffen ausschließlich die Beleglogik und liegen damit in der Verantwortung von Gruppe K. Die Komponente J stellt Kundendaten lediglich lesend über den KundenService bereit und ist von diesen Regeln zwar indirekt betroffen, spezifiziert sie jedoch nicht.

Die funktionalen Anforderungen BA-02 bis BA-04 aus dem Lastenheft werden von den Gruppen L (Bestandsmanagement) und K (Belegworkflow) abgedeckt und sind nicht Gegenstand dieses Dokuments.

## 10. Modultestplan

Die folgenden Testfälle sind deterministisch formuliert, das bedeutet sie haben feste Eingaben und klar definierte erwartete Ausgaben. Alle Testfälle sind direkt mit JUnit 5 umsetzbar. String-Vergleiche werden mit `assertEquals`, Nullprüfungen mit `assertNull` und Ausnahmen mit `assertThrows` getestet

TC-ID	Abgedeckte PH-Anf.	Vorbedingung	Test-Eingabe (Aktion)	Erwartetes Ergebnis
TC-01	F-01, F-02	Keine Kunde vorhanden, KundenIdGenerator bei Startwert	Vollständiger Kundendatensatz mit allen Pflichtfeldern	Kunde wird gespeichert, vergebene ID lautet „K-000001“
TC-02	F-02	Letzter vergebener Zähler = 5	<code>naechsteKundenId()</code> aufrufen	Rückgabe „K-000006“ (führende Nullen, String-Format)
TC-03	F-03, NF-USE-02	Formular geöffnet, kein Kunde vorhanden	Kundendatensatz ohne E-Mail-Adresse speichern	Speichervorgang wird abgelehnt, Validierungsfehler benennt „E-Mail“ als fehlendes Pflichtfeld
TC-04	F-03	Formular geöffnet, kein Kunde vorhanden	Kundendatensatz ohne Name speichern	Speichervorgang wird abgelehnt, Validierungsfehler benennt „Name“ als fehlendes Pflichtfeld
TC-05	F-04, NF-SEC-01	Kunde „K-000001“ wurde gespeichert	Anwendung schließen und neu starten, dann <code>findenKunden("K-000001")</code> aufrufen	Datensatz ist vollständig und unverändert verfügbar
TC-06	F-05, F-06, F-07	Kunde „K-000001“ mit E-Mail <a href="mailto:alt@mail.de">alt@mail.de</a> ist gespeichert	E-Mail auf <a href="mailto:neu@mail.de">neu@mail.de</a> ändern und speichern	<code>findenKunde("K-000001")</code> gibt Datensatz mit E-Mail <a href="mailto:neu@mail.de">neu@mail.de</a> zurück, ohne Neustart der Anwendung
TC-07	F-09	Drei Kunden gespeichert: „Khazanovych GmbH“, „Winkler AG“, „Senger KG“	<code>sucheNachName("Kh")</code> aufrufen	Rückgabe enthält ausschließlich „Khazanovych GmbH“, die anderen beiden Einträge sind nicht enthalten
TC-08	F-10	Kunde „K-000001“ ist gespeichert	<code>findeKunde("K-000001")</code> über	Vollständiger Kundendatensatz wird zurückgegeben

			KundenService aufrufen	
TC-09	F-10	Kein Kunde mit ID „K-999999“ vorhanden	findeKunde(“K-999999“) über KundenService aufrufen	Rückgabe ist null
TC-10	F-11, F-12, GR-04	Kunde „K-000001“ ist in Beleg „R-2026-000001“ referenziert	kundeLoschen(“K-000001“) aufrufen	Löschvorgang wird mit einer IllegalStateException abgelehnt, Datensatz bleibt vollständig erhalten
TC-11	F-11	Kunden „K-000002“ ist in keinem Beleg referenziert	kundeLoschen(“K-000002“) aufrufen	Datensatz wird gelöscht, findeKunde(“K-000002“) gibt anschließend null zurück
TC-12	F-13	Kunde „K-000001“ ist in den Belegen „AN-2026-000001“ und „R-2026-000001“ referenziert	Transaktionshistorie für „K-000001“ abrufen	Rückgabe enthält genau die Belegnummern „AN-2026-000001“ und „R-2026-000001“, keine weiteren Einträge
TC-13	NF-PERF-01	5.000 Kundendatensätze sind gespeichert	sucheNachname(“Khazanovych“) aufrufen	Ergebnis wird in unter 1 Sekunde zurückgegeben

Damit sind 13 Testfälle spezifiziert, die alle funktionalen Kernregeln (F-01 bis F-13) sowie die relevanten Geschäftsregeln (GR-04) und Qualitätsanforderungen (NF-SEC-01, NF-USE-02, NF-PERF-01) abdecken

## 11. Anhänge

### 11.1 Abkürzungen

Abkürzung	Bedeutung
F	Funktionale Anforderung
NF	Nicht-funktionale Anforderung
IF	Schnittstelle (Interface)
AC	Abnahmekriterium (Acceptance Criterion)
TC	Testfall (Test Case)
BA	Benutzeranforderung
GR	Geschäftsregel

Q	Qualitätsanforderung
GUI	Graphical User Interface (Grafische Benutzeroberfläche)
DSGVO	Datenschutz-Grundverordnung der Europäischen Union
GoBD	Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern in elektronischer Form
UstG	Umsatzsteuergesetz
Ust-IdNr	Umsatzsteuer-Identifikationsnummer

## 11.2 Glossar

Es gilt das Glossar des Lastenhefts (Abschnitt 8.2) unverändert. Nachfolgend sind ausschließlich Begriffe aufgeführt, die spezifisch für dieses Pflichtenheft eingeführt wurden und im Lastenheft nicht enthalten sind