

Pflichtenheft

Desktop-Fakturierungsanwendung — Gruppe A: Prozess /
Dokumentenzyklus

Team 1 – Gruppe A

Inhaltsverzeichnis

Dokumentenhistorie	3
1. Einleitung	4
1.1 Zweck des Dokuments	4
1.2 Ziel	4
1.3 Geltungsbereich	4
1.4 Definitionen und Abkürzungen	5
1.5 Referenzen	5
2. Systemüberblick	5
2.1 Kurzbeschreibung	5
2.2 Abgrenzung (Was gehört dazu / was nicht)	5
2.3 Grobe Systemfunktionen	6
2.4 UML-Bezug	6
3. Stakeholder und Kontext	6
4. Funktionale Anforderungen	7
4.1 Angebot (aus BA-09)	7
4.2 Auftragsbestätigung (aus BA-10)	8
4.3 Lieferschein (aus BA-11)	8
4.4 Rechnung (aus BA-12)	8
4.5 Geführte Rechnungserstellung (aus BA-13)	9
4.6 Rechnung stornieren (aus BA-14)	9
4.7 Übergreifende Prozessregeln	10
5. Nicht-funktionale Anforderungen	10
6. Daten und Schnittstellen	11
6.1 Datenobjekte und Datentypen	11
6.2 Schnittstellen	13
7. Systemarchitektur (logisch, grob)	14
7.1 Klassendiagramm	15
7.2 Sequenzdiagramm	15
8. Testbare Abnahmekriterien	17
9. Traceability LH ↔ PH	18
10. Modultestplan	19

11. Anhänge	22
11.1 Abkürzungen	22
11.2 Glossar	22
11.3 Referenzen	22

Autor	Prüfer	Freigebender
Strubel, Lucas Kaiser, Luca Gruppe A (Prozess)	Prof. Dr. Marmitt, Gerd Modulverantwortlicher	Prof. Dr. Marmitt, Gerd Modulverantwortlicher
09.06.2026	09.06.2026	09.06.2026

Freigabevermerk: Dieses Dokument ist nach Prüfung und Freigabe durch den Modulverantwortlichen verbindliche Spezifikationsgrundlage für die Implementierung und den Modultest der Komponente *Prozess / Dokumentenzyklus*.

Dokumentenhistorie

Version	Datum	Autor	Grund der Änderung
1.0	09.06.2026	Lucas Strubel, Luca Kaiser	Initiale Erstellung
1.1	13.06.2026	Lucas Strubel, Luca Kaiser	Einfügen der UML-Diagramme (Klassen- und Sequenzdiagramm)

1. Einleitung

1.1 Zweck des Dokuments

Dieses Pflichtenheft (System Requirements Specification, SRS) beschreibt aus Sicht des Auftragnehmers, **wie** die Komponente *Prozess / Dokumentenzyklus* der Desktop-Fakturierungsanwendung die Anforderungen des Lastenhefts (v1.3) erfüllt. Es konkretisiert die fachlichen Anforderungen in testbare Systemanforderungen und dient als direkte Grundlage für Design, Implementierung sowie den Komponenten- bzw. Modultestplan (Kapitel 10).

1.2 Ziel

Ziel dieses Pflichtenhefts ist die vollständige und testbare Spezifikation der Erzeugung, Verknüpfung und Statusführung der vier kaufmännischen Belegtypen (Angebot, Auftragsbestätigung, Lieferschein, Rechnung), der geführten Rechnungserstellung sowie der Rechnungstornierung.

1.3 Geltungsbereich

Dieses Dokument gilt für die Komponente **Gruppe A — Prozess / Dokumentenzyklus**. Die Gesamtanwendung wird arbeitsteilig in vier Komponenten entwickelt; jede Untergruppe pflegt ein eigenes Pflichtenheft:

Gruppe	Komponente	Eigenes Pflichtenheft
A	Prozess / Dokumentenzyklus	dieses Dokument
B	Verwaltung von Produkten	separat
C	Verwaltung von Kunden	separat
D	Programmoberfläche	separat

Die Komponente A **nutzt** Kunden (Gruppe C) und Produkte (Gruppe B) lesend über definierte interne Schnittstellen (Kapitel 6.2) und wird über die Programmoberfläche (Gruppe D) bedient. Stammdatenpflege (Anlegen/Ändern/Löschen von Kunden und Produkten) ist **nicht** Gegenstand dieses Dokuments.

1.4 Definitionen und Abkürzungen

Fachbegriffe (Angebot, Auftragsbestätigung, Lieferschein, Rechnung, Dokumentenzyklus, GoBD, DSGVO, ...) sind im Glossar des Lastenhefts (§ 8.1) definiert und gelten unverändert. Dokumentspezifische Abkürzungen siehe Kapitel 11.

1.5 Referenzen

- Lastenheft „Desktop-Fakturierungsanwendung“, Team 1, Version 1.3, 09.06.2026
 - Project Charter, Team 1, Version 1.3, 14.05.2026
 - § 14 UStG — Pflichtangaben einer Rechnung
 - GoBD — Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern
 - DSGVO — EU-Verordnung 2016/679
 - Vorlesungsunterlagen Software Engineering 1 (SoSe 2026), Foliensatz „Lasten- und Pflichtenheft“
-

2. Systemüberblick

2.1 Kurzbeschreibung

Die Anwendung ist eine **Einzelplatz-Stand-Alone-Desktop-Anwendung** mit **lokaler Datenhaltung** (keine Cloud, kein Server). Die Bedienung erfolgt über eine **minimale grafische Benutzeroberfläche** (Gruppe D), über die die Funktionalität des Dokumentenzyklus zugänglich gemacht wird. Erzeugte Belege werden lokal als **PDF** exportiert und können optional gedruckt oder per Standard-E-Mail-Client versendet werden.

Die Komponente *Prozess / Dokumentenzyklus* stellt die fachliche Kernlogik bereit: Belegerzeugung, automatische Summen- und Steuerberechnung, Vergabe eindeutiger Belegnummern, Verknüpfung aufeinanderfolgender Belege, Statusführung und Stornierung.

2.2 Abgrenzung (Was gehört dazu / was nicht)

Im Umfang dieser Komponente:

- Erstellen von Angebot, Auftragsbestätigung, Lieferschein, Rechnung
- Übernahme von Daten aus Vorgängerbelegen (Dokumentenzzyklus-Konsistenz)
- Automatische Berechnung von Netto-, Steuer- und Bruttobeträgen (Snapshot-Prinzip)
- Vergabe fortlaufender, lückenloser Belegnummern
- Geführte (schrittweise) Rechnungserstellung
- Statusführung und Stornierung von Rechnungen
- PDF-Export der Belege

Nicht im Umfang dieser Komponente:

- Anlegen/Ändern/Löschen von Kunden (Gruppe C) und Produkten (Gruppe B)
- Aufbau und Layout der GUI (Gruppe D)
- E-Rechnungsformate (ZUGFeRD/XRechnung), Mahnwesen, Buchhaltung (LH-Nichtziele)

2.3 Grobe Systemfunktionen

Erstellen von Belegen → Berechnen der Summen → Vergeben der Belegnummer → Persistieren → PDF-Export → Statuswechsel (inkl. Storno).

2.4 UML-Bezug

Ein gemeinsames Use-Case-Diagramm aller Gruppen gibt den Überblick über die Akteure und Ziele. Die für Gruppe A relevanten Use Cases sind: *Angebot erstellen*, *Auftragsbestätigung erstellen*, *Lieferschein erstellen*, *Rechnung erstellen (geführt)* und *Rechnung stornieren*. Die detaillierte logische Architektur dieser Komponente folgt in Kapitel 7.

3. Stakeholder und Kontext

Stakeholder und Systemkontext sind im Lastenheft (§ 2, § 3) beschrieben und gelten unverändert. Für diese Komponente ist der maßgebliche Akteur:

- **Anwender:in** — natürliche Person (Selbstständige:r, Freiberufler:in, Kleinunternehmer:in), die den Dokumentenzyklus eigenverantwortlich durchführt.

Angrenzende Systeme/Komponenten: lokales Dateisystem (Persistenz, PDF), optional Drucker und Standard-E-Mail-Client sowie intern die Komponenten Kundenverwaltung (C) und Produktverwaltung (B).

4. Funktionale Anforderungen

Die Anforderungen sind nach Belegtyp gruppiert und mit den Satzschablonen des Foliensatzes formuliert. Jede Anforderung ist eindeutig, vollständig, widerspruchsfrei und verifizierbar.

Belegnummern (übergreifend, GR-01): Belegnummern sind **eindeutig** und werden **vom System generiert** (nicht durch den Anwender eingegeben). Sie werden als `String` geführt, **nicht** als `int`, weil die Nummern ein festes Format mit führenden Nullen und Präfix besitzen (z. B. R-2026-000124); ein ganzzahliger Typ würde führende Nullen verlieren. Pro Belegtyp wird ein eigener, fortlaufender und **lückenloser** Zähler auf Basis der höchsten bisher vergebenen Nummer geführt. Präfixe: AN- (Angebot), AB- (Auftragsbestätigung), LS- (Lieferschein), R- (Rechnung).

4.1 Angebot (aus BA-09)

F-01: Das System MUSS es der Anwender:in ERMÖGLICHEN, ein Angebot für einen existierenden Kunden mit mindestens einer Position aus dem Produktkatalog zu erstellen.

F-02: WENN ein Angebot gespeichert wird, DANN MUSS das System eine eindeutige Angebotsnummer (Präfix AN-), das Erstelldatum und ein Gültigkeitsdatum setzen.

F-03: Das System MUSS für jedes Angebot die Netto-, Steuer- und Bruttosumme automatisch aus den Positionen berechnen (siehe F-23).

F-04: Das System MUSS es ERMÖGLICHEN, ein gespeichertes Angebot als PDF in das

lokale Dateisystem zu exportieren.

4.2 Auftragsbestätigung (aus BA-10)

F-05: Das System MUSS es ERMÖGLICHEN, eine Auftragsbestätigung zu erstellen, wobei Kunde, Positionen und Mengen aus einem vorhandenen Angebot übernommen werden können (siehe F-22).

F-06: WENN eine Auftragsbestätigung gespeichert wird, DANN MUSS das System eine eindeutige AB-Nummer (Präfix AB-) vergeben und — sofern aus einem Angebot erzeugt — eine Rückreferenz auf das Angebot speichern.

F-07: Das System MUSS es ERMÖGLICHEN, eine Auftragsbestätigung als PDF zu exportieren.

4.3 Lieferschein (aus BA-11)

F-08: Das System MUSS es ERMÖGLICHEN, einen Lieferschein mit Lieferdatum, Positionen und Liefermengen zu erstellen; Kunde und Positionen können aus einer Auftragsbestätigung übernommen werden (siehe F-22).

F-09: WENN ein Lieferschein gespeichert wird, DANN MUSS das System eine eindeutige Lieferscheinnummer (Präfix LS-) vergeben und — sofern aus einer Auftragsbestätigung erzeugt — eine Rückreferenz speichern.

F-10: Das System MUSS es ERMÖGLICHEN, einen Lieferschein als PDF zu exportieren.

4.4 Rechnung (aus BA-12)

F-11: Das System MUSS es ERMÖGLICHEN, eine Rechnung für einen Kunden mit mindestens einer Position zu erstellen.

F-12: WENN eine Rechnung gespeichert wird, DANN MUSS das System eine fortlaufende, lückenlose Rechnungsnummer (Präfix R-) auf Basis der höchsten bisher vergebenen Nummer vergeben (GR-01).

F-13: Das System MUSS in jeder Rechnung die Pflichtangaben gemäß § 14 UStG führen: Rechnungsnummer, Rechnungsdatum, Leistungsdatum, Kundendaten, Positionen mit

Einzel- und Gesamtbeträgen, Steuersatz und Steuerbetrag sowie Netto-, Steuer- und Bruttosumme.

F-14: WENN bei der Rechnungserstellung kein abweichendes Zahlungsziel angegeben ist, DANN MUSS das System ein Standard-Zahlungsziel von **14 Kalendertagen** ab Rechnungsdatum setzen (GR-06).

F-15: Das System MUSS es ERMÖGLICHEN, eine Rechnung als PDF zu exportieren.

4.5 Geführte Rechnungserstellung (aus BA-13)

F-16: Das System MUSS es ERMÖGLICHEN, die Rechnungserstellung schrittweise durchzuführen: (1) Kunde auswählen, (2) mindestens eine Produktposition mit Menge erfassen, (3) Rechnungsdatum und Zahlungsziel bestätigen, (4) Zusammenfassung prüfen, (5) speichern.

F-17: WENN die Schritteingabe abgeschlossen ist, DANN MUSS das System vor dem Speichern eine **Zusammenfassung** mit Kunde, allen Positionen, Mengen, Netto-/Steuer-/Bruttosumme, Rechnungsdatum und Zahlungsziel anzeigen.

F-18: WENN ein Pflichtfeld fehlt (kein Kunde, keine Position, kein Rechnungsdatum), DANN MUSS das System das Speichern ablehnen und das fehlende Pflichtfeld benennen (GR/Q-09).

4.6 Rechnung stornieren (aus BA-14)

F-19: Das System MUSS es ERMÖGLICHEN, eine gespeicherte Rechnung im Status OFFEN zu stornieren.

F-20: WENN eine Rechnung storniert wird, DANN MUSS das System ihren Status auf STORNIERT setzen, sie nicht mehr in der Liste offener Rechnungen führen und den Vorgang mit Datum protokollieren.

F-21: WENN eine Rechnung den Status STORNIERT hat, DANN MUSS das System jede weitere inhaltliche Änderung ablehnen.

4.7 Übergreifende Prozessregeln

F-22 (Dokumentenzyklus-Konsistenz, GR-05): WENN ein Beleg aus einem Vorgängerbeleg erzeugt wird, DANN MUSS das System Kunde, Positionen und Mengen übernehmen und eine eindeutige Rückreferenz auf den Vorgänger speichern.

F-23 (Steuer-/Summenberechnung, GR-03): WENN eine Position gespeichert wird, DANN MUSS das System Netto-, Steuer- und Bruttobetrag automatisch berechnen, wobei der zum Zeitpunkt der Belegerstellung gültige Steuersatz und Einzelpreis des Produkts als unveränderlicher **Snapshot** in der Position abgelegt werden.

F-24 (Unveränderlichkeit, GR-02 / Q-07): WENN ein Beleg den Status VERSENDET hat, DANN MUSS das System jede inhaltliche Änderung ablehnen; Korrekturen erfolgen ausschließlich über neue Belege (Storno-/Korrekturrechnung).

5. Nicht-funktionale Anforderungen

NF-PERF-01 (aus Q-03): Das System MUSS die Erstellung und den PDF-Export eines beliebigen Belegtyps INNERHALB VON 2 SEKUNDEN abschließen, bei Belegen mit bis zu 50 Positionen und einem Datenbestand gemäß Q-01 (bis 5.000 Kunden/Produkte).

NF-INT-01 (aus Q-07 / GR-02): Das System MUSS nach dem Status VERSENDET einer Rechnung **jede** inhaltliche Änderung ablehnen; zulässig bleiben ausschließlich Storno- bzw. Korrekturvorgänge über neue Belege.

NF-USE-01 (aus Q-05): Die geführte Erstellung einer vollständigen Rechnung an einen bestehenden Kunden MUSS von einer erstmaligen Anwender:in OHNE EXTERNE HILFE IN WENIGER ALS 10 MINUTEN im ersten Versuch abgeschlossen werden können (Nachweis durch Usability-Test mit mind. 5 Testpersonen).

NF-USE-02 (aus Q-09): Das System MUSS fehlende Pflichtangaben in den Belegformularen so markieren und benennen, dass mindestens 80 % der Testpersonen die fehlende Eingabe ohne externe Hilfe im ersten Korrekturversuch ergänzen können.

6. Daten und Schnittstellen

Dieses Kapitel ist direkter Input für den Modultestplan (Kapitel 10). Datentypen werden bereits als Java-Typen angegeben.

6.1 Datenobjekte und Datentypen

Designgrundsätze:

- **Geldbeträge** werden als `java.math.BigDecimal` mit **Scale 2** und kaufmännischer Rundung (`RoundingMode.HALF_UP`) geführt — **nicht** als `double` (Gleitkommata-Rundungsfehler wären für Beträge unzulässig).
- **Belegnummern** werden als `String` geführt (festes Format mit Präfix und führenden Nullen, z. B. "R-2026-000124") — **nicht** als `int`.
- **Datumswerte** werden als `java.time.LocalDate` geführt.
- **Mengen** werden als `int` (Stückzahl) geführt.
- **Steuersätze** werden als `BigDecimal` als Faktor geführt (z. B. 0.19).

```
enum DokumentStatus { ENTWURF, OFFEN, VERSENDET, STORNIERT }
```

Klasse Dokumentposition

Attribut	Java-Typ	Beschreibung
produktReferenz	String	Produktnummer des referenzierten Produkts (Gruppe B)
bezeichnung	String	Snapshot der Produktbezeichnung zum Erstellzeitpunkt
menge	int	Stückzahl (> 0)
einzelpreisNetto	BigDecimal	Snapshot Netto-Einzelpreis (Scale 2)

Attribut	Java-Typ	Beschreibung
steuersatz	BigDecimal	Snapshot Steuersatz, z. B. 0.19
positionssummeNetto	BigDecimal	einzelpreisNetto * menge (Scale 2)

Abstrakte Klasse Dokument

Attribut	Java-Typ	Beschreibung
belegnummer	String	eindeutig, vom System generiert
datum	LocalDate	Erstelldatum
kundenReferenz	String	Kundennummer (Gruppe C)
positionen	List<Dokumentposition>	mind. 1 Position
status	DokumentStatus	Lebenszyklus- Status
vorgaengerNr	String (optional, null)	Rückreferenz auf Vorgängerbeleg (GR-05)
summeNetto	BigDecimal	Summe aller Positionssummen (Scale 2)
summeSteuer	BigDecimal	Summe der Steuerbeträge (Scale 2)
summeBrutto	BigDecimal	summeNetto + summeSteuer (Scale 2)

Spezialisierungen (erben von Dokument)

Klasse	Zusätzliche Attribute (Java-Typ)
Angebot	gueltigBis: LocalDate
Auftragsbestaetigung	— (nutzt vorgaengerNr → Angebot)
Lieferschein	lieferdatum: LocalDate
Rechnung	leistungsdatum: LocalDate, zahlungsziel: LocalDate, storniertAm: LocalDate (optional)

6.2 Schnittstellen

Externe Schnittstellen:

ID	Schnittstelle	Zweck
IF-01	Lokales Dateisystem	Persistenz der Belege, Ablage exportierter PDF-Dokumente
IF-02	Druckersystem (optional)	Direkter Druck eines Belegs
IF-03	Standard-E-Mail-Client (optional)	Versand eines Belegs als PDF-Anhang

Interne Schnittstellen (zu anderen Komponenten), als Java-Interfaces skizziert:

// Lesender Zugriff auf Kundenstammdaten (Gruppe C)

```
public interface KundenService {
```

```
    Kunde findeKunde(String kundenummer); // null, wenn nicht vorhanden
```

```
}  
  
// Lesender Zugriff auf Produktstammdaten (Gruppe B)  
public interface ProductService {  
    Produkt findeProdukt(String produktnummer);  
}  
  
// PDF-Export eines Belegs (IF-01)  
public interface PdfExporter {  
    void exportiere(Dokument dokument, Path zielDatei);  
}
```

Belegnummern-Schnittstelle (komponenteninterner Dienst):

```
public interface BelegnummernGenerator {  
    // liefert die nächste lückenlose Nummer für den Belegtyp (GR-01)  
    String naechsteNummer(Belegtyp typ, int jahr);  
}
```

IF-Satzschablone (Beispiel IF-01): *Das System MUSS eine Datei-Schnittstelle bereitstellen, die es dem lokalen Dateisystem ERMÖGLICHT, Belege zu persistieren und PDF- Dokumente abzulegen. Die Schnittstelle MUSS gängige Dateipfade (java.nio.file.Path) verwenden.*

7. Systemarchitektur (logisch, grob)

Die Komponente folgt einer einfachen Schichtung: die GUI (Gruppe D) ruft den DokumentService (realisiert durch StandardDokumentService) auf, der die Fachlogik kapselt und die Dienste BelegnummernGenerator, KundenService, ProductService und PdfExporter nutzt. Belege werden über ein DokumentRepository im lokalen Dateisystem persistiert (realisiert als JSON-Ablage). Nach jeder schreibenden Operation meldet der DokumentService die Datenänderung über einen **EreignisBus** (Observer-Muster, Pa-

ket gemeinsam; melde(DatenBereich.DOKUMENTE)); die Modulansichten der Gruppe D abonnieren diesen Bus und aktualisieren sich automatisch.

7.1 Klassendiagramm

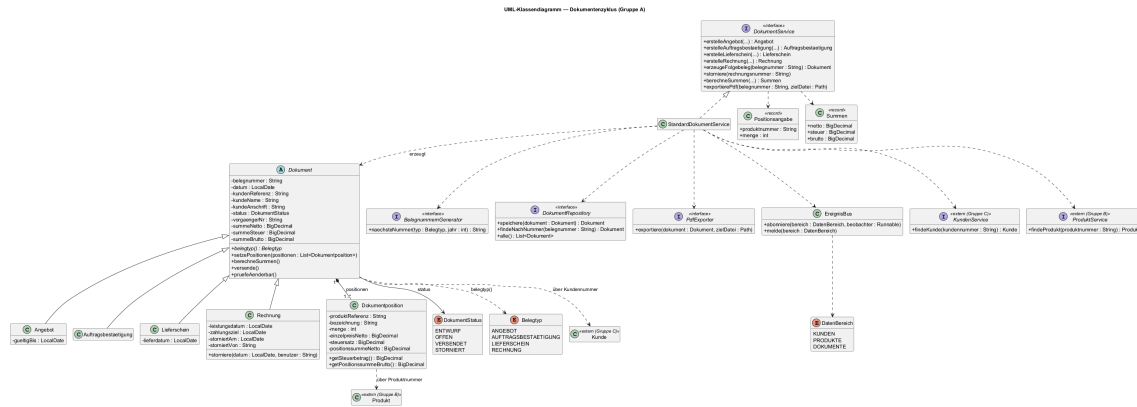


Abbildung 1: UML-Klassendiagramm Dokumentenzyklus (Gruppe A)

Beschreibung zu Abbildung 1: Das Klassendiagramm zeigt die abstrakte Oberklasse Dokument mit den Spezialisierungen Angebot, Auftragsbestaetigung, Lieferschein und Rechnung (Vererbung). Ein Dokument besteht aus einer bis vielen Dokumentposition-Objekten (Komposition zwischen Dokument und Dokumentposition, Multiplizität 1..*). Jede Dokumentposition referenziert ein Produkt (Gruppe B), ein Dokument referenziert einen Kunde (Gruppe C) — jeweils über die Stammdatenummer (lose Kopplung). Der DokumentService orchestriert die Erstellung und nutzt den BelegnummernGenerator (Vergabe lückenloser Belegnummern), die Schnittstellen KundenService/ProduktService (Stammdaten), den PdfExporter (PDF-Export) sowie den EreignisBus (Benachrichtigung der Modulansichten nach Datenänderungen, Observer-Muster). Der Status eines Belegs wird über das Enum DokumentStatus abgebildet.

7.2 Sequenzdiagramm

Beschreibung zu Abbildung 2: Das Sequenzdiagramm stellt den Ablauf Rechnung erstellen dar. Die Anwender:in löst über die GUI (Gruppe D) erstelleRechnung(kundenNr, positionen) am DokumentService aus. Dieser ermittelt über KundenService.findeKunde(...) und ProduktService.findeProdukt(...) die

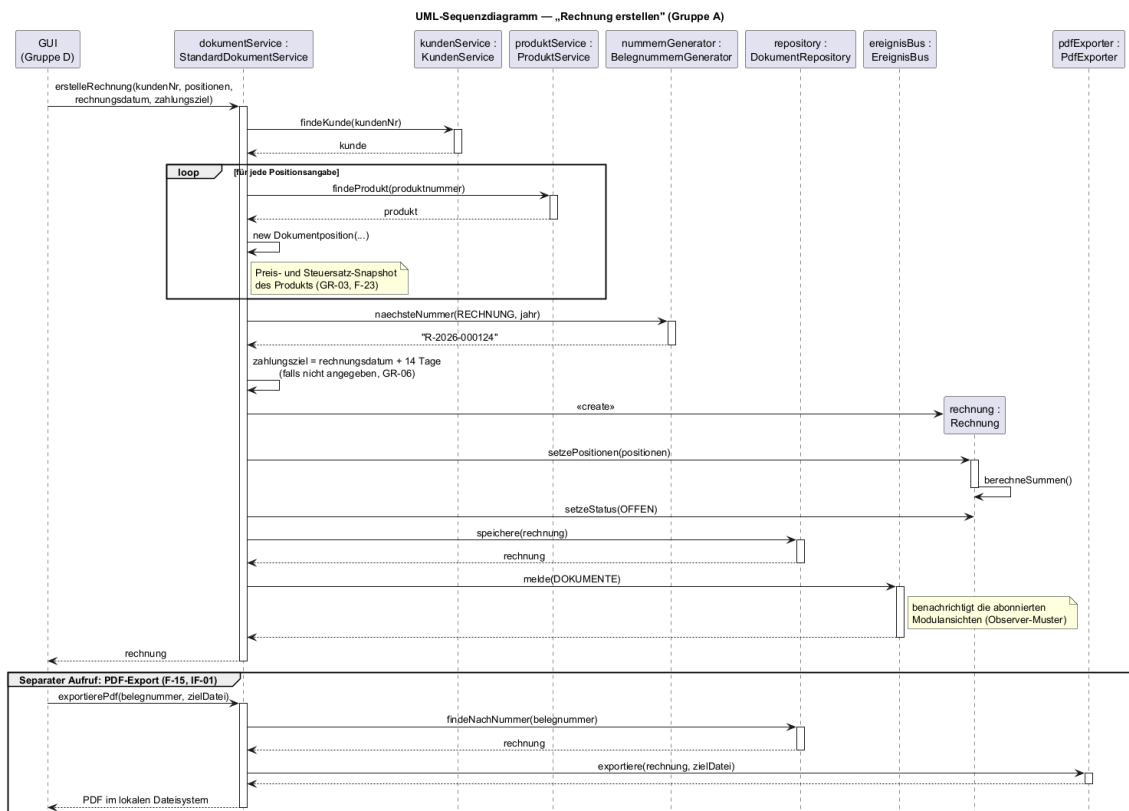


Abbildung 2: Abbildung 2: UML-Sequenzdiagramm „Rechnung erstellen“ (Gruppe A)

Stammdaten (Snapshot), berechnet je Position und in Summe Netto-, Steuer- und Bruttobetrag (F-23), fordert vom BelegnummernGenerator mit `naechsteNummer(RECHNUNG, jahr)` eine lückenlose Rechnungsnummer an (GR-01), setzt das Standard-Zahlungsziel (+14 Tage, GR-06), persistiert den Beleg über das `DokumentRepository` und meldet die Änderung über `EreignisBus.melde(DOKUMENTE)` an die abonnierten Modulansichten. Abschließend wird die gespeicherte Rechnung an die GUI zurückgegeben. Der PDF-Export ist ein **separater** Vorgang (im Diagramm unten dargestellt): über `exportierePdf(...)` lädt der `DokumentService` den Beleg erneut aus dem `DokumentRepository` und exportiert ihn mittels `PdfExporter.exportiere(...)` in das lokale Dateisystem (F-15, IF-01).

8. Testbare Abnahmekriterien

AC-A-01 (zu F-01–F-04, NF-PERF-01) — *Angebot erstellen und exportieren* Vorbedingung: Ein Kunde und 5 Produkte sind erfasst. Aktion: Anwender:in erstellt ein Angebot mit 5 Positionen und exportiert es als PDF. Erwartet: Das Angebot ist mit Angebotsnummer (AN-...) und korrekten Summen gespeichert; der PDF-Export ist in ≤ 2 Sekunden abgeschlossen.

AC-A-02 (zu F-05–F-07, F-22) — *Auftragsbestätigung aus Angebot* Vorbedingung: Ein Angebot liegt vor. Aktion: Anwender:in erstellt eine Auftragsbestätigung mit Übernahme aller Positionen. Erwartet: Die AB ist mit eindeutiger Nummer (AB-...), übernommenen Positionen/Mengen und Rückreferenz auf das Angebot gespeichert und als PDF exportierbar.

AC-A-03 (zu F-08–F-10, F-22) — *Lieferschein erstellen* Vorbedingung: Eine Auftragsbestätigung liegt vor. Aktion: Anwender:in erstellt einen Lieferschein mit Lieferdatum. Erwartet: Der Lieferschein ist mit eindeutiger Nummer (LS-...), Lieferdatum und allen Positionsdaten gespeichert und als PDF exportierbar.

AC-A-04 (zu F-11–F-15, F-23) — *Rechnung mit Pflichtangaben und Standard-Zahlungsziel* Vorbedingung: Kunde und mind. eine Position liegen vor; letzte Rech-

nungsnummer = R-2026-000123. Aktion: Anwender:in erstellt eine Rechnung mit Rechnungsdatum 09.06.2026 ohne abweichendes Zahlungsziel. Erwartet: Die Rechnung trägt die Nummer R-2026-000124, ein Zahlungsziel 23.06.2026 (+14 Tage), alle Pflichtangaben gem. § 14 UStG sowie korrekte Netto-/Steuer-/Bruttosummen.

AC-A-05 (zu F-16–F-18, NF-USE-01/02) — *Geführte Rechnungserstellung* Vorbedingung: Mind. ein Kunde und ein Produkt vorhanden. Aktion: Anwender:in durchläuft die geführte Erstellung (Kunde → Position+Menge → Datum/ Zahlungsziel → Zusammenfassung → speichern). Erwartet: Vor dem Speichern erscheint eine Zusammenfassung mit Kunde, Position, Menge, Summen, Rechnungsdatum und Zahlungsziel; fehlt ein Pflichtfeld, wird das Speichern abgelehnt und das fehlende Feld benannt.

AC-A-06 (zu F-19–F-21) — *Rechnung stornieren* Vorbedingung: Eine Rechnung im Status OFFEN existiert. Aktion: Anwender:in storniert die Rechnung. Erwartet: Status wird STORNIERT, die Rechnung erscheint nicht mehr in der Liste offener Rechnungen, der Vorgang ist mit Datum protokolliert; weitere Änderungen werden abgelehnt.

AC-A-07 (zu F-23, F-24, NF-INT-01) — *Snapshot und Unveränderlichkeit* Vorbedingung: Eine Rechnung mit einem Produkt ist erstellt; danach wird der Produktpreis geändert; eine zweite Rechnung im Status VERSENDET existiert. Aktion: Vergleich der ersten Rechnung mit dem geänderten Produktpreis; Änderungsversuch an der versendeten Rechnung. Erwartet: Die erste Rechnung behält den ursprünglichen Preis (Snapshot); der Änderungsversuch an der versendeten Rechnung wird abgelehnt.

9. Traceability LH ↔ PH

Jede für Gruppe A relevante Lastenheft-Anforderung ist mindestens einer Pflichtenheft-Anforderung zugeordnet.

LH-

Anforderung	Beschreibung (LH)	PH-Anforderung(en)
BA-09	Angebot erstellen	F-01, F-02, F-03, F-04
BA-10	Auftragsbestätigung erstellen	F-05, F-06, F-07, F-22

LH-		
Anforderung	Beschreibung (LH)	PH-Anforderung(en)
BA-11	Lieferschein erstellen	F-08, F-09, F-10, F-22
BA-12	Rechnung erstellen	F-11, F-12, F-13, F-14, F-15
BA-13	Geführte Rechnungserstellung	F-16, F-17, F-18
BA-14	Rechnung stornieren	F-19, F-20, F-21
GR-01	Lückenlose Rechnungsnummern	F-12 (Belegnummern-Regel)
GR-02	Unveränderlichkeit versendeter Dokumente	F-24, F-21, NF-INT-01
GR-03	Steuerberechnung (Snapshot)	F-23, F-03, F-13
GR-05	Dokumentenzklus-Konsistenz	F-22, F-06, F-09
GR-06	Standard-Zahlungsziel 14 Tage	F-14
Q-03	Performance PDF-Erstellung ≤ 2 s	NF-PERF-01
Q-05	Usability Ersterstellung Rechnung	NF-USE-01
Q-07	Unveränderlichkeit versendeter Rechnungen	NF-INT-01, F-24
Q-09	Pflichtfeldhinweise ≥ 80 %	NF-USE-02, F-18

Hinweis: GR-04 (Löschsperre für verknüpfte Kunden) liegt in der Verantwortung von Gruppe C; Komponente A nutzt Kundendaten nur lesend (IF/KundenService) und ist von dieser Regel betroffen, spezifiziert sie aber nicht.

10. Modultestplan

Die folgenden Testfälle sind deterministisch (feste Ein-/Ausgaben) und mit JUnit 5 umsetzbar. Geldbeträge werden als `BigDecimal` mit Scale 2 erwartet (`assertEquals(new BigDecimal("119.00"), ...)` bzw. `compareTo`).

Abgedeckte		Erwartetes		
TC	PH-Anf.	Vorbedingung	Eingabe	Ergebnis
TC-01	F-23, F-03	Position mit Netto 100.00 €, Steuersatz 0.19	berechne()	Steuer = 19.00, Brutto = 119.00 (Scale 2)
TC-02	F-23	Einzelpreis 50.00 €, Menge 3	Positionssumme berechnen	positionssummeNetto = 150.00
TC-03	F-03, F-13	Beleg mit 2 Positionen (150.00 € @ 0.19; 50.00 € @ 0.07)	Summen berechnen	summeNetto = 200.00, summeSteuer = 32.00, summeBrutto = 232.00
TC-04	F-12, GR-01	Letzte Rechnungsnummer R-2026-000123	naechsteNummer (RECHNUNG, 2026)	liefert R-2026-000124 (lückenlos)
TC-05	F-12 (Format)	Zähler = 7, Jahr 2026	naechsteNummer (RECHNUNG, 2026)	liefert R-2026-000007 (führende Nullen, String)
TC-06	F-14, GR-06	Rechnungsdatum 2026-06-09, kein Zahlungsziel	Rechnung erstellen	zahlungsziel = 2026-06-23 (+14 Tage)
TC-07	F-14	Rechnungsdatum 2026-06-09, Zahlungsziel 2026-07-31	Rechnung erstellen	zahlungsziel = 2026-07-31 (übernommen)
TC-08	F-24, NF-INT-01	Rechnung im Status VERSENDET	setzePosition(...) / Änderung	wirft IllegalStateException

Abgedeckte				Erwartetes
TC	PH-Anf.	Vorbedingung	Eingabe	Ergebnis
TC-09	F-19, F-20	Rechnung im Status OFFEN	storniere()	Status = STORNIERT; nicht in offeneRechnungen(); storniertAm gesetzt
TC-10	F-22, GR-05	Angebot AN-2026-000001 mit Kunde + 2 Positionen	ausAngebot (angebot (AB erzeugen))	AB übernimmt Kunde/ Positionen/Mengen; vorgaengerNr = "AN-2026-000001"
TC-11	F-23, F-24	Rechnung mit Produkt @ 50.00 €; danach Produktpreis → 80.00 €	erste Rechnung erneut lesen	einzelpreisNetto bleibt 50.00 (Snapshot unverändert)
TC-12	F-18, NF-USE-02	Rechnung ohne Kunde oder ohne Position	speichere()	Speichern abgelehnt; Validierungsfehler benennt fehlendes Pflichtfeld
TC-13	F-11, F-12, F-13	Kunde + 1 Position vorhanden	vollständige Rechnung erstellen	Rechnung gespeichert, Nummer vergeben, alle § 14 UStG-Pflichtangaben gesetzt

Damit sind 13 Testfälle (> 10) spezifiziert, die alle funktionalen Kernregeln (F-12, F-14, F-18, F-22, F-23, F-24) sowie die zentralen Geschäftsregeln (GR-01, GR-02, GR-03, GR-05, GR-06) abdecken.

11. Anhänge

11.1 Abkürzungen

Abkürzung	Bedeutung
F	Funktionale Anforderung (Pflichtenheft)
NF	Nicht-funktionale Anforderung (Pflichtenheft)
IF	Schnittstelle (Interface)
AC	Abnahmekriterium
TC	Testfall (Test Case)
BA	Benutzeranforderung (Lastenheft)
GR	Geschäftsregel (Lastenheft)
Q	Qualitätsanforderung (Lastenheft)
SRS	System Requirements Specification (Pflichtenheft)

11.2 Glossar

Es gilt das Glossar des Lastenhefts (§ 8.1) unverändert.

11.3 Referenzen

Siehe Kapitel 1.5.