

Pflichtenheft - Dokumentenprozess Fakturierungssystem

Gruppe F - SE1 Team 2

Inhaltsverzeichnis

Freigabeuebersicht	2
Dokumentenhistorie	2
1. Einleitung	2
1.1 Zweck dieses Dokuments	2
1.2 Bezug zum Project Charter	2
1.3 Geltungsbereich	2
2. Systemueberblick fuer Gruppe F	3
3. Stakeholder und Kontext	3
4. Funktionale Anforderungen	3
5. Nicht-funktionale Anforderungen fuer das Modul	7
6. Daten und Schnittstellen	8
6.1 Datenobjekte und Java-Datentypen	8
6.2 Schnittstellen	9
7. Systemarchitektur	9
8. Testbare Abnahmekriterien	11
9. Traceability zwischen Lastenheft und Pflichtenheft	13
10. Modultestplan	13
11. Glossar	16
12. Referenzen	16

Freigabeuebersicht

Ersteller	Pruefer	Freigebender
Gruppe F: Andreas Ivanovic, Armin Omanovic, Alexander Teller SE1 Team 2 11.06.2026	Prof. Dr. Gerd Marmitt Hochschule Mannheim [Datum Review]	SE1 Team 2 SE1 Team 2 30.06.2026

Dokumentenhistorie

Version	Datum	Autor	Aenderung
1.0	11.06.2026	Gruppe F / Alexander Teller	Initiale Erstellung des Pflichtenheft-Teils und Modultestplans fuer den Dokumentenprozess

1. Einleitung

1.1 Zweck dieses Dokuments

Dieses Pflichtenheft beschreibt die konkrete Umsetzung des Moduls **Dokumentenprozess** im Fakturierungssystem. Es konkretisiert die Anforderungen aus dem Lastenheft und dem Project Charter fuer den Verantwortungsbereich der **Gruppe F**. Der Fokus liegt auf der fachlichen Kette:

Angebot -> Auftragsbestaetigung -> Lieferschein -> Rechnung

Das Dokument dient als Input fuer Implementierung, Komponententests und Integration mit Produktverwaltung, Kundenverwaltung und GUI.

1.2 Bezug zum Project Charter

Team 2 besteht aus den Gruppen E, F, G und H. Gruppe F ist laut Project Charter fuer den Dokumentenprozess verantwortlich. Der Dokumentenprozess umfasst die fachliche Kette Angebot -> Auftragsbestaetigung -> Lieferschein -> Rechnung. Das Projekt nutzt Java, JavaFX, SQLite und Gitty und orientiert sich am V-Modell.

1.3 Geltungsbereich

Dieses Pflichtenheft deckt nur den Dokumentenprozess ab. Produktverwaltung, Kundenverwaltung und GUI werden nur als Schnittstellen beschrieben, weil sie durch die Gruppen G, H und E verantwortet werden.

Nicht Teil dieses Moduls sind: vollstaendige Buchhaltung, E-Rechnung, Mehrbenutzerbetrieb, Cloud-Betrieb und externe ERP-/FiBu-Anbindungen.

2. Systemueberblick fuer Gruppe F

Das Modul Gruppe F stellt die Geschaeftslogik fuer Dokumente bereit. Die GUI ruft den DokumentService auf. Produkt- und Kundendaten werden ueber Lookup-Schnittstellen gelesen. Dokumente werden persistent ueber ein Repository gespeichert. Aus vorhandenen Dokumenten koennen Folgedokumente erzeugt werden, wobei Status, Referenzen und fachliche Regeln geprueft werden.

3. Stakeholder und Kontext

Stakeholder / Rolle	Interesse am Modul Dokumentenprozess
Auftraggeber Prof. Dr. Gerd Marmitt	Nachvollziehbare Umsetzung der Anforderungen und testbare Artefakte im V-Modell.
Gruppe F	Implementierung des Dokumentenworkflows und Bereitstellung testbarer Services.
Gruppe E - GUI	Benötigt Service-Methoden und DTOs fuer Anzeige, Eingabe und Export.
Gruppe G - Produktverwaltung	Liefert Produktdaten und muss pruefen koennen, ob Produkte in Dokumenten referenziert sind.
Gruppe H - Kundenverwaltung	Liefert Kundendaten und muss pruefen koennen, ob Kunden in Dokumenten referenziert sind.
Endnutzer	Moechte Angebote, Auftraege, Lieferscheine und Rechnungen korrekt erzeugen und wiederfinden.

4. Funktionale Anforderungen

PH-ID	LH-Quelle	Kurzname	Konkretisierung / Systemverhalten	Prioritaet	Abnahmekriterium
PH-DP-01	BA-DP-01, GR-04, GR-06	Angebot anlegen	Das System MUSS ueber DokumentService.erstelleAngebot(long kundenId, List<PositionCommand> positionen) ein Angebot fuer genau einen vorhandenen Kunden mit mindestens einer Position erzeugen. Pro Position werden Produktbezeichnung, Einzelpreis netto und MwSt.-Satz als Snapshot uebernommen.	Muss	Angebot ist gespeichert, besitzt Nummer A-yyyy-nnnn, Status OFFEN, Datum LocalDate.now() und korrekt berechnete Netto-, USt.- und Bruttosumme.
PH-DP-02	F-DP-02	Angebotsnummer vergeben	Das System MUSS Angebotsnummern ueber OfferNumber-Generator je Kalenderjahr fortlaufend und eindeutig vergeben.	Muss	Zwei nacheinander erzeugte Angebote erhalten unterschiedliche und aufeinanderfolgende Nummern.
PH-DP-03	F-DP-03	Angebot bearbeiten	Das System MUSS die Bearbeitung eines Angebots nur erlauben, solange AngebotsStatus.OFFEN gilt.	Muss	Bearbeitung bei OFFEN erfolgreich; bei UEBERFUEHRT wird IllegalStateException geworfen.
PH-DP-04	BA-DP-04, GR-01	Auftragsbestaetigung erzeugen	Das System MUSS aus genau einem offenen Angebot eine Auftragsbestaetigung mit eigener Nummer AB-yyyy-nnnn erzeugen und alle Positionen unveraendert uebernehmen.	Muss	AB referenziert Angebotsnummer, Positionen und Summen entsprechen dem Angebot, Angebot wird auf UEBERFUEHRT gesetzt.

PH-ID	LH-Quelle	Kurzname	Konkretisierung / Systemverhalten	Prioritaet	Abnahmekriterium
PH-DP-05	F-DP-05	Doppelte Auftragsbestaetigung verhindern	Das System MUSS vor der Erzeugung pruefen, ob fuer das Angebot bereits eine Auftragsbestaetigung existiert.	Muss	Zweiter Erzeugungsversuch wird mit Duplicate-DocumentException abgelehnt.
PH-DP-06	BA-DP-06, GR-01	Lieferschein erzeugen	Das System MUSS aus genau einer bestaetigten Auftragsbestaetigung einen Lieferschein mit Nummer LS-yyyy-nnnn und LocalDate Lieferdatum erzeugen.	Muss	Lieferschein referenziert AB, uebernimmt Produktbezeichnung und Mengen und erhaelt Status GELIEFERT.
PH-DP-07	F-DP-07	Lieferschein ohne Preise	Das System MUSS fuer Lieferscheinansicht und PDF nur Menge und Produktbezeichnung ausgeben, aber keine Einzelpreise, Netto-, USt.- oder Bruttowerte.	Muss	Export/DTO des Lieferscheins enthaelt keine Preisfelder.
PH-DP-08	BA-DP-08, F-DP-09, GR-01	Rechnung erzeugen	Das System MUSS aus genau einem gelieferten Lieferschein eine Rechnung mit Nummer R-yyyy-nnnn, Rechnungsdatum, Referenz auf den Lieferschein und Pflichtangaben nach UStG erzeugen.	Muss	Rechnung enthaelt Kunde, Leistungsdatum/Lieferdatum, Positionen, fortlaufende Nummer, Netto-, USt.- und Bruttosumme.

PH-ID	LH-Quelle	Kurzname	Konkretisierung / Systemverhalten	Prioritaet	Abnahmekriterium
PH-DP-09	F-DP-09	Pflichtangaben pruefen	Das System MUSS vor dem Speichern einer Rechnung RechnungPflichtangabenValidator.validate(Rechnung) ausfuehren.	Muss	Fehlende Pflichtangaben verhindern Speicherung und liefern Validierungsfehler.
PH-DP-10	F-DP-10, GR-02	Rechnungsnummer fortlaufend	Das System MUSS Rechnungsnummern fortlaufend, eindeutig und lueckenlos vergeben, sobald eine Rechnung erfolgreich gespeichert wird.	Muss	Drei erfolgreiche Rechnungen ergeben R-2026-0001, R-2026-0002, R-2026-0003.
PH-DP-11	F-DP-11, GR-03	Rechnung festschreiben	Das System MUSS eine gespeicherte Rechnung inhaltlich sperren (festgeschrieben=true).	Muss	Aenderungsversuche an Positionen, Summen oder Pflichtangaben werden abgelehnt.
PH-DP-12	F-DP-12	PDF-Export	Das System SOLL jedes Dokument ueber PdfExportService.export(Dokument) als PDF-Datei im lokalen Dateisystem erzeugen.	Soll	PDF-Datei existiert, ist nicht leer und enthaelt Dokumentnummer und Dokumenttyp.
PH-DP-13	F-DP-13	Dokumentenubersicht pro Kunde	Das System SOLL ueber DokumentRepository.findByKunde(long kundenId) alle Dokumente eines Kunden sortiert nach Datum und Dokumenttyp bereitstellen.	Soll	Alle Angebote, ABs, Lieferscheine und Rechnungen des Kunden werden ohne fremde Kundendokumente geliefert.

PH-ID	LH-Quelle	Kurzname	Konkretisierung / Systemverhalten	Prioritaet	Abnahmekriterium
PH-DP-14	GR-04	Preis-Snapshot sichern	Das System MUSS den Produktpreis und MwSt.-Satz beim Einfuegen in das Angebot als Snapshot in Dokumentposition speichern.	Muss	Spaetere Produktaenderung beeinflusst bestehende Dokumentpositionen nicht.
PH-DP-15	GR-05	Stammdatenreferenzen pruefbar machen	Das Dokumentenmodul MUSS fuer Produkt- und Kundenverwaltung eine Prueffunktion <code>isReferenced(...)</code> bereitstellen.	Muss	Referenzierte Produkte/Kunden koennen durch andere Module als verwendet erkannt werden.
PH-DP-16	NF-TEST-02, NF-MAINT-03	Testbare Logik ohne GUI	Die Dokumentenlogik MUSS in Service- und Domainklassen ohne JavaFX-Abhaengigkeit implementiert werden.	Muss	JUnit-Tests koennen DokumentService ohne Start der GUI ausfuehren.

5. Nicht-funktionale Anforderungen fuer das Modul

PH-NF-ID	LH-Quelle	Konkretisierung	Nachweis
PH-NF-DP-01	NF-PERF-01	Dokumentenaktionen bis 1.000 Kunden/Produkte muessen innerhalb von 1 Sekunde eine Rueckmeldung liefern, wenn Repositories mit Testdaten initialisiert sind.	Zeitmessung im Komponententest/Integrationstest < 1 s fuer Erzeugung eines Angebots mit 10 Positionen.
PH-NF-DP-02	NF-ARCH-01	Dokumente muessen nach Neustart aus SQLite/Dateispeicher vollstaendig wiederherstellbar sein.	Dokument erstellen, Repository neu initialisieren, Dokument unveraendert laden.

PH-NF-ID	LH-Quelle	Konkretisierung	Nachweis
PH-NF-DP-03	NF-ARCH-02	Datenhaltung des Dokumentenmoduls muss ueber DokumentRepository gekapselt sein.	Code-Review: Service nutzt Interface, keine direkten SQL-Aufrufe im Service.
PH-NF-DP-04	NF-MAINT-01, NF-MAINT-03	Dokumentenprozess wird im Paket de.hsma.se1.fakturierung.dokumentRepository umgesetzt und in Domain, Service, Repository und Export getrennt.	Repository-Struktur enthaelt getrennte Pakete; Abhaengigkeiten laufen von UI -> Service -> Repository.
PH-NF-DP-05	NF-TEST-02, NF-TEST-03	Die Geschaeftslogik des Dokumentenmoduls muss automatisiert testbar sein.	Mindestens 10 JUnit-Testfaelle fuer Services/Domainklassen laufen ohne GUI.

6. Daten und Schnittstellen

6.1 Datenobjekte und Java-Datentypen

Datenfeld / Objekt	Java-Datentyp	Wertebereich / Format	Beschreibung
DokumentType	enum	ANGEBOT, AUFTRAGSBESTAETIGUNG, LIEFERSCHEIN, RECHNUNG	Typisierung fuer Anzeige, Export und Repository.
Dokumentnummer	String	Format A-2026-0001, AB-2026-0001, LS-2026-0001, R-2026-0001	Fachlicher Identifikator; nicht als int, da Praefix/Jahr enthalten.
kundenId	long	z. B. 10001L	Referenz auf Kundenverwaltung, keine Kopie des gesamten Kundenobjekts im Dokumentservice erforderlich.
produktId	long	z. B. 20001L	Referenz auf Produktverwaltung.
bezeichnungSnapshot	String	max. 255 Zeichen	Produktbezeichnung zum Zeitpunkt der Angebotsanlage.
menge	int	> 0	Menge pro Dokumentposition.
einzelpreisNettoSnapshot	BigDecimal	>= 0, Skalierung 2	Geldbetrag; BigDecimal statt double wegen Rundungsfehlern.
mwstSatzSnapshot	BigDecimal	z. B. 0.19, 0.07	MwSt.-Satz als Dezimalwert.
datum, lieferdatum, rechnungsdatum	LocalDate	ISO-Format yyyy-MM-dd	Fachliche Datumswerte ohne Uhrzeit.

Datenfeld / Objekt	Java-Datentyp	Wertebereich / Format	Beschreibung
nettoSumme, ustSumme, bruttoSumme status	BigDecimal enum	Skalierung 2, Rundung HALF_UP OFFEN, UEBERFUEHRT, BESTAETIGT, GELIEFERT, FAKTURIERT, FESTGESCHRIEBEN	Berechnete Summen des Dokuments. Steuert erlaubte Folgeaktionen.
pflichtangaben	RechnungPflichtangaben	Objekt mit leistungserbringerName, anschrift, steuerNr, leistungsdatum usw.	Pflichtfelder der Rechnung.

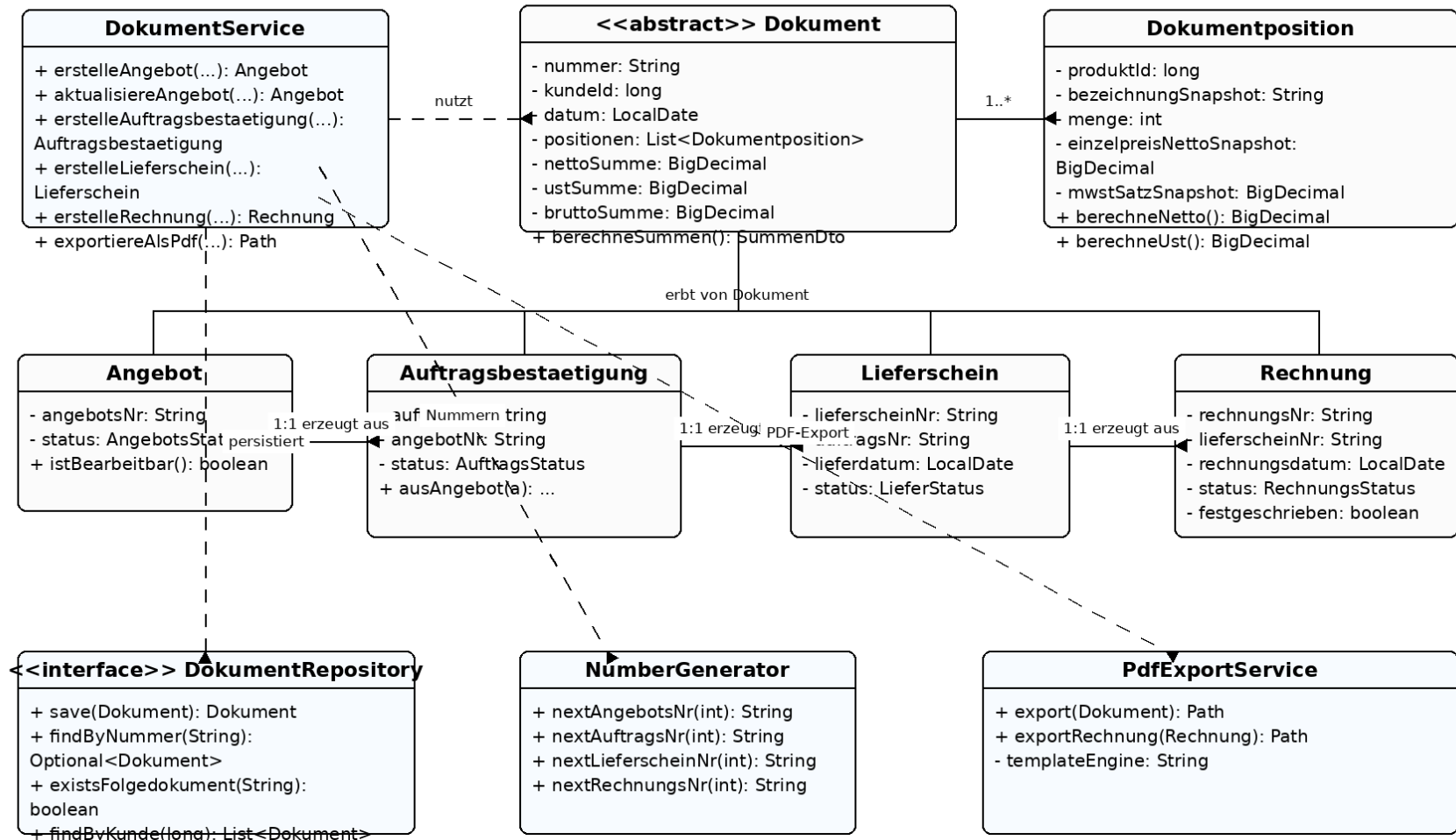
6.2 Schnittstellen

Schnittstelle	Art	Zweck	Operationen / Daten
DokumentService	Java-Service-Interface	UI/Controller ruft Dokumentenlogik auf.	Methoden: erstelleAngebot, erstelleAuftragsbestaeti- gung, erstelleLieferschein, erstelleRechnung, exportiereAlsPdf.
DokumentRepository	Java-Interface / SQLite-Implementierung	Persistenz von Dokumenten.	save, findByNummer, findByKunde, existsFolgedokument, existsReferencedProdukt, existsReferencedKunde.
ProductLookup	Java-Interface zu Gruppe G	Produktdaten fuer Angebotspositionen lesen.	findProduct(long produktId): Optional<ProductDto> mit String bezeichnung, BigDecimal einzelpreisNetto, BigDecimal mwstSatz.
CustomerLookup	Java-Interface zu Gruppe H	Kundendaten fuer Dokumentkopf und Rechnungspflichtangaben lesen.	findCustomer(long kundenId): Optional<CustomerDto>.
PdfExportService	Java-Service / lokale Datei-Schnittstelle	PDF-Dateien aus Dokumentdaten erzeugen.	Rueckgabe Path; Zielordner z. B. /exports.

7. Systemarchitektur

Das Dokumentenmodul folgt einer dreischichtigen Struktur: Praesentation/UI ruft Services auf, Services enthalten die Geschaeftslogik, Repositories kapseln die Datenhaltung. Die fachlichen Domaenenklassen sind von JavaFX unabhaengig und koennen isoliert getestet werden.

UML-Klassendiagramm - Gruppe F: Dokumentenprozess



Die Repositories kapseln SQLite/Dateispeicherung. ProductService und CustomerService werden ueber IDs/Snapshots angebunden.

Abbildung 1: UML-Klassendiagramm Gruppe F

Beschreibung zu Abbildung 1: Das Klassendiagramm zeigt die zentrale abstrakte Klasse Dokument mit den Spezialisierungen Angebot, Auftragsbestätigung, Lieferschein und Rechnung. Die Klasse Dokumentposition speichert Produktdaten als Snapshot. Der DokumentService nutzt Repositories, Nummerngenerator und PDF-Export, ohne selbst technische Persistenzdetails zu kennen.

Beschreibung zu Abbildung 2: Das Sequenzdiagramm beschreibt den Ablauf zur Rechnungserzeugung aus einem Lieferschein. Zuerst wird der Lieferschein geladen und sein Status geprüft. Danach wird eine Rechnungsnummer vergeben, die Rechnung berechnet, gespeichert, optional als PDF exportiert und an die GUI zurückgegeben. Falls der Lieferschein fehlt oder nicht den Status GELIEFERT besitzt, wird keine Rechnung erzeugt.

8. Testbare Abnahmekriterien

AC-ID	Abgedeckte PH-Anforderungen	Testbares Abnahmekriterium
AC-DP-01	PH-DP-01, PH-DP-02, PH-DP-14	Aus vorhandenem Kunden und zwei Produkten wird ein Angebot erzeugt. Erwartet werden korrekte Nummer, Status OFFEN, zwei Positionen, Preis-Snapshots und korrekte Summen.
AC-DP-02	PH-DP-03	Ein offenes Angebot kann geändert werden. Ein überführtes Angebot kann nicht mehr geändert werden.
AC-DP-03	PH-DP-04, PH-DP-05	Aus einem offenen Angebot wird genau eine Auftragsbestätigung erzeugt. Ein zweiter Versuch wird abgelehnt.
AC-DP-04	PH-DP-06, PH-DP-07	Aus einer bestätigten AB wird ein Lieferschein erzeugt. Der Lieferschein enthält Mengen und Bezeichnungen, aber keine Preise.
AC-DP-05	PH-DP-08, PH-DP-09, PH-DP-10, PH-DP-11	Aus einem gelieferten Lieferschein wird eine festgeschriebene Rechnung mit Pflichtangaben, lückenloser Nummer und korrekten Summen erzeugt.
AC-DP-06	PH-DP-12	Ein Dokument kann als PDF exportiert werden; die Datei existiert und enthält Dokumentnummer und Dokumenttyp.
AC-DP-07	PH-DP-13	Alle Dokumente eines Kunden werden kundenbezogen aufgelistet.
AC-DP-08	PH-DP-15	Produkt- und Kundenverwaltung können prüfen, ob Stammdaten in Dokumenten referenziert werden.

UML-Sequenzdiagramm - Rechnung aus Lieferschein erzeugen

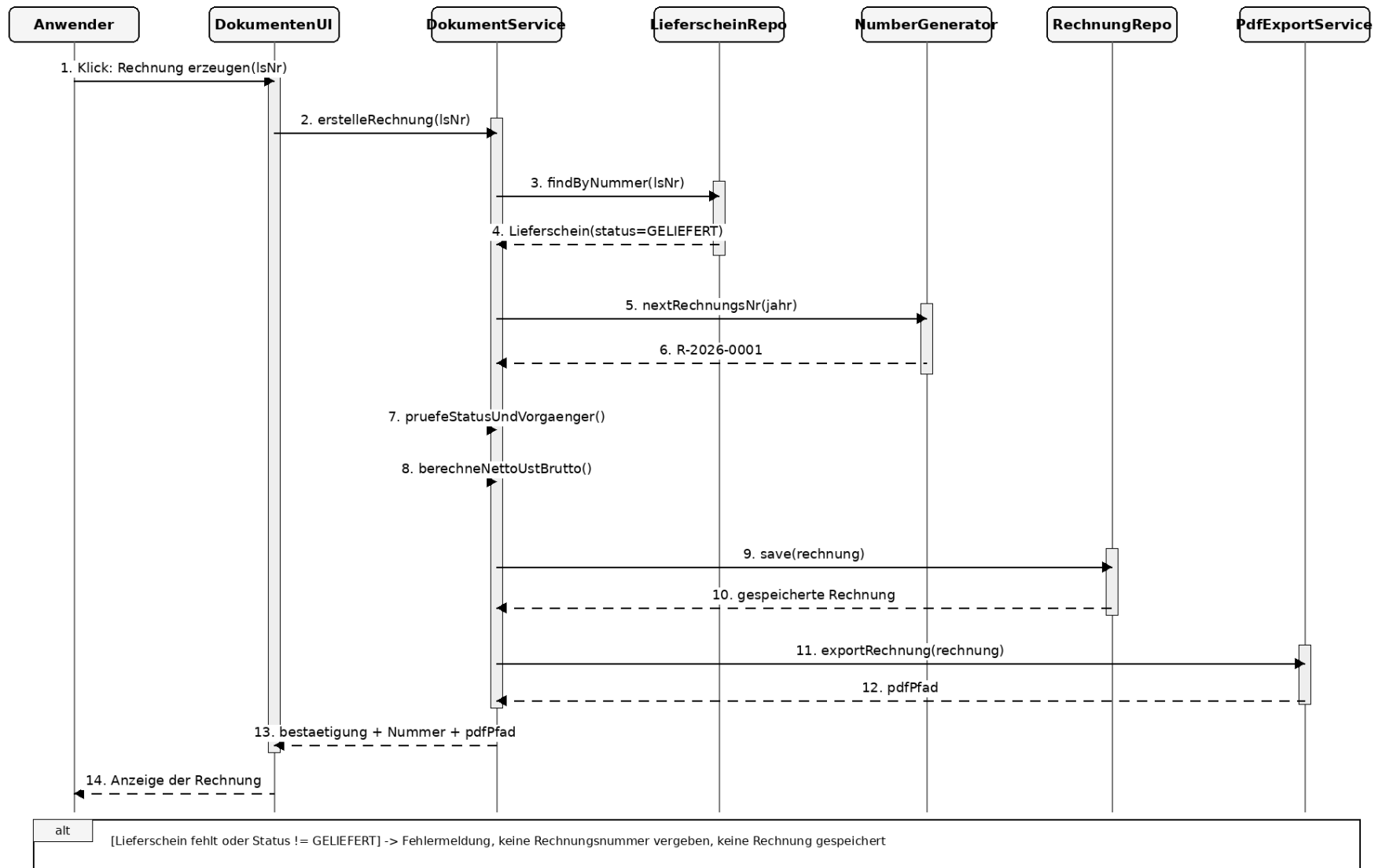


Abbildung 2: UML-Sequenzdiagramm Rechnung aus Lieferschein

9. Traceability zwischen Lastenheft und Pflichtenheft

LH-Anforderung	Kurzbeschreibung	PH-Anforderungen	Abnahmekriterium	Modultestfaelle
BA-DP-01	Angebot fuer Kunden anlegen	PH-DP-01, PH-DP-14	AC-DP-01	MT-DP-01, MT-DP-02, MT-DP-13
F-DP-02	Angebotsnummer eindeutig/fortlaufend	PH-DP-02	AC-DP-01	MT-DP-03
F-DP-03	Angebot nur bis Ueberfuehrung bearbeiten	PH-DP-03	AC-DP-02	MT-DP-04
BA-DP-04	Auftragsbestaetigung aus Angebot	PH-DP-04	AC-DP-03	MT-DP-05
F-DP-05	Maximal eine AB pro Angebot	PH-DP-05	AC-DP-03	MT-DP-06
BA-DP-06	Lieferschein aus AB	PH-DP-06	AC-DP-04	MT-DP-07
F-DP-07	Lieferschein ohne Preise	PH-DP-07	AC-DP-04	MT-DP-08
BA-DP-08	Rechnung aus Lieferschein	PH-DP-08	AC-DP-05	MT-DP-09
F-DP-09	Rechnungspflichtangaben	PH-DP-09	AC-DP-05	MT-DP-10
F-DP-10	Rechnungsnummer fortlaufend/lueckenlos	PH-DP-10	AC-DP-05	MT-DP-11
F-DP-11	Rechnung unveraenderlich	PH-DP-11	AC-DP-05	MT-DP-12
F-DP-12	PDF-Export	PH-DP-12	AC-DP-06	MT-DP-14
F-DP-13	Dokumentenuebersicht Kunde	PH-DP-13	AC-DP-07	MT-DP-15
GR-01	Dokumentenketten	PH-DP-04, PH-DP-06, PH-DP-08	AC-DP-03 bis AC-DP-05	MT-DP-05, MT-DP-07, MT-DP-09
GR-02	Rechnungsnummern eindeutig	PH-DP-10	AC-DP-05	MT-DP-11
GR-03	Rechnung nicht aenderbar	PH-DP-11	AC-DP-05	MT-DP-12
GR-04	Preisuebernahme Snapshot	PH-DP-14	AC-DP-01	MT-DP-13
GR-05	Stammdatenschutz	PH-DP-15	AC-DP-08	MT-DP-16
GR-06	Summenberechnung	PH-DP-01, PH-DP-08	AC-DP-01, AC-DP-05	MT-DP-02, MT-DP-09

10. Modultestplan

Die folgenden Testfaelle sind so formuliert, dass sie direkt mit JUnit 5 umgesetzt werden koennen. Fuer den isolierten Modultest werden Produkt- und Kundendaten durch Fake-Lookups bereitgestellt. Die GUI wird nicht gestartet.

Test-ID	Name	Abgedeckte PH-Anforderungen	Vorbedingung	Testschritte	Erwartetes Ergebnis	Umsetzungshinweis
MT-DP-01	Angebot anlegen - Happy Path	PH-DP-01	Kunde 10001 und Produkte 20001/20002 existieren.	erstelleAngebot(10001, [2x Produkt 20001, 1x Produkt 20002])	Angebot gespeichert, Status OFFEN, 2 Positionen, Nummer beginnt mit A-2026-.	JUnit: assertEquals, FakeProductLookup, InMemoryDocumentRepository
MT-DP-02	Summenberechnung Angebot	PH-DP-01, PH-DP-08	Positionen: 2 x 100,00 EUR mit 19 %, 1 x 50,00 EUR mit 7 %.	angebot.berechneSumme()	Netto 250,00; USt. 41,50; Brutto 291,50.	JUnit: BigDecimal-Vergleich mit compareTo oder skalierter Assertion
MT-DP-03	Fortlaufende Angebotsnummern	PH-DP-02	Leerer Nummernkreis 2026.	Zwei Angebote erzeugen.	Nummern A-2026-0001 und A-2026-0002.	JUnit: assertEquals, assertTrue(endsWith)
MT-DP-04	Angebot nach Ueberfuehrung sperren	PH-DP-03	Angebot wurde in AB ueberfuehrt.	aktualisiereAngebot aufrufen.	IllegalStateException; Positionen unveraendert.	JUnit: assertThrows
MT-DP-05	Auftragsbestaetigung erzeugen	PH-DP-04	Offenes Angebot mit 2 Positionen existiert.	erstelleAuftragsbestaetigung()	AB mit eigener Nummer, Angebotsreferenz und identischen Positionen gespeichert.	JUnit: assertEquals, Objektvergleich Positionen
MT-DP-06	Doppelte AB verhindern	PH-DP-05	Fuer Angebot existiert bereits eine AB.	erstelleAuftragsbestaetigung() erneut.	Exception; keine zweite AB im Repository.	JUnit: assertThrows, Repository-Count
MT-DP-07	Lieferschein erzeugen	PH-DP-06	AB im Status BESTAETIGT existiert.	erstelleLieferschein(Sm, Lieferdatum)	Sm (Auftragsnr, Lieferdatum, AB-Referenz und Status GELIEFERT gespeichert.	JUnit: assertEquals
MT-DP-08	Lieferschein enthaelt keine Preise	PH-DP-07	Lieferschein aus AB existiert.	LieferscheinDto oder Exportmodell erzeugen.	DTO enthaelt Menge und Bezeichnung, aber keine Preis-/Steuerfelder.	JUnit: Getter/DTO-Feldpruefung

Test-ID	Name	Abgedeckte PH-Anforderungen	Vorbedingung	Testschritte	Erwartetes Ergebnis	Umsetzungshinweis
MT-DP-09	Rechnung erzeugen - Happy Path	PH-DP-08, PH-DP-09	LS im Status GELIEFERT mit Kunde und Positionen existiert.	erstelleRechnung(R-..., Nr)	Rechnung mit R-..., Pflichtangaben, Summen und Status FESTGESCHRIEBEN gespeichert.	JUnit: assertAll
MT-DP-10	Pflichtangaben fehlen	PH-DP-09	Kunde ohne Rechnungsanschrift/Steuernummer in Testdaten.	erstelleRechnung(Validierungsfehler; keine Rechnung gespeichert.	Validierungsfehler; keine Rechnung gespeichert.	JUnit: assertThrows(ValidationException)
MT-DP-11	Rechnungsnummern lueckenlos	PH-DP-10	Drei unterschiedliche gelieferte Lieferscheine existieren.	Drei Rechnungen erzeugen.	Nummern enden auf 0001, 0002, 0003 ohne Luecke.	JUnit: Listenvergleich
MT-DP-12	Rechnung unveraenderlich	PH-DP-11	Rechnung wurde gespeichert/festgeschrieben	Aenderung von Position oder Summe versuchen.	Aenderung wird abgelehnt; geladene Rechnung bleibt unveraendert.	JUnit: assertThrows, erneutes Laden
MT-DP-13	Preis-Snapshot bleibt stabil	PH-DP-14	Angebot wurde mit Produktpreis 100,00 erzeugt. Danach Produktpreis auf 120,00 setzen.	Angebot erneut laden.	Positionspreis im Angebot bleibt 100,00.	JUnit mit FakeProductLookup-Preiswechsel
MT-DP-14	PDF-Export erzeugt Datei	PH-DP-12	Angebot oder Rechnung existiert.	exportiereAlsPdf(RolokentzPath; RolokentzPath)	Datei existiert und Groesse > 0 Byte.	JUnit mit temporaerem Verzeichnis @TempDir
MT-DP-15	Dokumentenuebersicht pro Kunde	PH-DP-13	Dokumente fuer Kunde 1 und Kunde 2 existieren.	findByKunde(1)	Nur Dokumente von Kunde 1, sortiert nach Datum.	JUnit: assertTrue(allMatch)
MT-DP-16	Stammdatenreferenz pruefen	PH-DP-15	Produkt 20001 ist in Angebot referenziert, Produkt 99999 nicht.	isProductReferenced(20001) und isProductReferenced(99999)	Edge(20001) true bzw. false.	JUnit: assertTrue, assertFalse

11. Glossar

Begriff	Bedeutung
Angebot	Preisangebot an einen Kunden, Ausgangspunkt der Dokumentenkette.
Auftragsbestaetigung	Verbindliche Bestaetigung der Annahme eines Angebots.
Lieferschein	Dokumentiert Lieferung; enthaelt keine Preise.
Rechnung	Festgeschriebenes Dokument mit Zahlungsforderung und Pflichtangaben.
Snapshot	Kopie von Produktpreis, Produktbezeichnung und MwSt.-Satz zum Zeitpunkt der Angebotsanlage.
Traceability	Rueckverfolgbarkeit von Lastenheft-Anforderung zu Pflichtenheft-Anforderung und Testfall.

12. Referenzen

- Project Charter - Fakturierungssystem, Version 1.0, 15.04.2026.
- Lastenheft - Fakturierungssystem, Version 1.0 alpha, 12.05.2026.
- Vorlesung Software Engineering 1: Lastenheft, Pflichtenheft, Traceability, V-Modell und Komponententests.