

## Belegworkflow

Datum: 15.06.2026 | Version: 1.4

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

## Inhalt

1.	Einleitung und Zielbestimmung.....	2
1.1	Ziel des Dokuments.....	2
1.2	Geltungsbereich.....	2
1.3	Definitionen und Abkürzungen.....	3
1.4	Referenzen.....	3
2.	Systemüberblick.....	3
2.1	Kurzüberblick.....	3
2.2	Abgrenzung.....	3
2.3	Grobe Systemfunktionen.....	4
2.4	UML-Bezug.....	4
3.	Stakeholder und Kontext.....	4
4.	Funktionale Anforderungen.....	4
4.1	Angebot erstellen.....	5
4.2	Angebot in Rechnung überführen.....	5
4.3	Rechnung und Rechtskonformität.....	5
4.4	Übergreifende Belegregeln.....	5
5.	Nicht-funktionale Anforderungen.....	6
5.1	Benutzbarkeit.....	6
5.2	Datensicherheit und Zuverlässigkeit.....	6
5.3	Revisionssicherheit.....	6
6.	Daten und Schnittstellen.....	6
6.1	Designgrundsätze für Datentypen.....	6
6.2	Datenobjekte.....	6
6.3	Interne Schnittstellen.....	9
6.4	Externe Schnittstellen.....	9
7.	Systemarchitektur (logisch, grob).....	10
7.1	Klassendiagramm.....	10
7.2	Sequenzdiagramm.....	11
8.	Testbare Abnahmekriterien.....	11
9.	Traceability: Lastenheft (LH) - Pflichtenheft (PH).....	13
10.	Modultestplan.....	14
11.	Anhänge.....	16
11.1	Abkürzungen.....	16
11.2	Begriffe.....	16

## Freigabeübersicht

Autor	Freigebenden	Prüfer
Taha Erdogan, Kutlu Patir, Sarav Guli	Taha Erdogan, Kutlu Patir, Sarav Guli	Prof. Dr. Marmitt, Gerd
Entwickler	Entwickler	Modulverantwortlicher
15.06.2026	15.06.2026	Datum, Unterschrift

## Dokumentenhistorie

Version	Datum	Autor	Grund der Änderung
1.0	10.06.2026	Taha Erdogan, Kutlu Patir, Sarav Guli	Grundstruktur
1.1	11.06.2026	Taha Erdogan, Kutlu Patir, Sarav Guli	Anforderungen
1.2	12.06.2026	Taha Erdogan, Kutlu Patir, Sarav Guli	Daten und Schnittstellen
1.3	13.06.2026	Taha Erdogan, Kutlu Patir, Sarav Guli	UML und Tests
1.4	15.06.2026	Taha Erdogan, Kutlu Patir, Sarav Guli	Finale Überarbeitung

## 1. Einleitung und Zielbestimmung

### 1.1 Ziel des Dokuments

Ziel dieses Pflichtenhefts ist die vollständige und testbare Spezifikation der Komponente Belegworkflow. Die Komponente beschreibt die systemgestützte Erstellung von Angeboten sowie die Überführung eines vorhandenen Angebots in eine rechtskonforme Rechnung gemäß § 14 UStG.

Das Dokument konkretisiert die Anforderungen des Lastenhefts aus Sicht des Auftragnehmers. Es dient als verbindliche Grundlage für Implementierung, Schnittstellenabstimmung und Komponententest der Belegworkflow-Komponente.

### 1.2 Geltungsbereich

Dieses Dokument gilt für die Komponente Gruppe K - Belegworkflow. Die Gesamtanwendung wird arbeitsteilig in mehrere Kernbereiche spezifiziert. Dieses Pflichtenheft beschreibt ausschließlich die Fachlogik, Datenhaltung und Schnittstellen des Belegworkflows.

Die Komponente K kapselt die Erstellung und Speicherung von Angeboten, die Übernahme von Angebotsdaten in Rechnungen, die automatische Summenberechnung, die Vergabe lückenloser Rechnungsnummern, den Schreibschutz finaler Belege sowie den PDF-Export von Angeboten und Rechnungen.

Nicht Bestandteil dieses Dokuments sind die Pflege von Kundenstammdaten, die Pflege von Produkt- und Bestandsdaten sowie die grafische Benutzeroberfläche. Diese Bereiche werden über definierte Schnittstellen angebunden und hier nur soweit beschrieben, wie sie für den Belegworkflow erforderlich sind.

### 1.3 Definitionen und Abkürzungen

Fachbegriffe wie Fakturierung, Angebot, Rechnung, Snapshot-Prinzip, GoBD und DSGVO werden im Lastenheft Team 3 definiert und gelten für dieses Dokument unverändert. Dokumentspezifische Abkürzungen sind in Kapitel 11 aufgeführt.

### 1.4 Referenzen

- Lastenheft "Fakturierungssystem", Team 3, Version 1.0, 15.05.2026
- § 14 UStG - Pflichtangaben einer Rechnung
- GoBD - Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form
- DSGVO - Datenschutz-Grundverordnung der Europäischen Union
- Vorlesungsunterlagen Software Engineering 1 (SoSe 2026), Foliensatz "Lasten- und Pflichtenheft"

## 2. Systemüberblick

### 2.1 Kurzüberblick

Die Anwendung ist eine Einzelplatz-Desktop-Anwendung mit lokaler Datenhaltung. Die Komponente Belegworkflow stellt die Fachlogik für kaufmännische Belege bereit. Sie besitzt keine eigene Oberfläche, sondern wird durch die Benutzeroberfläche aufgerufen und nutzt Kunden- sowie Produktdaten über Schnittstellen.

Im Mittelpunkt stehen zwei Vorgänge: Erstens kann ein Angebot für einen vorhandenen Kunden mit mindestens einer Produktposition erstellt werden. Zweitens kann aus einem vorhandenen Angebot eine Rechnung erzeugt werden, wobei Kundendaten, Positionen und Mengen übernommen werden und eine eindeutige Rechnungsnummer vergeben wird.

### 2.2 Abgrenzung

Im Umfang dieser Komponente:

- Erstellung und Speicherung von Angeboten mit Kunde, Positionen, Summen und Status.
- Überführung eines vorhandenen Angebots in eine Rechnung.
- Automatische Berechnung von Netto-, Umsatzsteuer- und Bruttosumme.
- Snapshot-Speicherung von Produktbezeichnung, Einzelpreis und Steuersatz in Dokumentpositionen.

- Vergabe lückenloser und fortlaufender Rechnungsnummern.
- Schreibschutz für versendete bzw. festgeschriebene Belege.
- PDF-Export von Angebot und Rechnung in das lokale Dateisystem.
- Bereitstellung von Referenzprüfungen für Kunden und Produkte.

Nicht im Umfang dieser Komponente:

- Anlegen, Ändern oder Löschen von Kundendaten.
- Anlegen, Ändern oder Löschen von Produkt- und Bestandsdaten.
- Gestaltung und technische Umsetzung der grafischen Benutzeroberfläche.
- Cloud-Speicherung, Mehrbenutzerbetrieb oder externe Buchhaltungsschnittstellen.
- Aktive Spezifikation von Auftragsbestätigung und Lieferschein, da der für Gruppe K relevante Lastenheft-Workflow direkt Angebot zu Rechnung beschreibt.

### 2.3 Grobe Systemfunktionen

Aufruf durch Benutzeroberfläche -> Abruf von Kunden- und Produktdaten -> Validierung der Pflichtdaten -> Erstellung des Angebots -> Berechnung der Summen -> Speicherung im lokalen Datenbestand -> spätere Auswahl des Angebots -> Erzeugung der Rechnung mit Datenübernahme -> Vergabe der Rechnungsnummer -> Speicherung und optionaler PDF-Export.

### 2.4 UML-Bezug

Die logische Architektur der Komponente wird in Kapitel 7 dargestellt. Das Klassendiagramm beschreibt die fachlichen Klassen, Services und Schnittstellen des Belegworkflows. Das Sequenzdiagramm beschreibt den Ablauf, bei dem aus einem vorhandenen Angebot eine Rechnung erzeugt und anschließend als PDF exportiert wird.

## 3. Stakeholder und Kontext

Stakeholder, Benutzerrollen und Systemkontext sind im Lastenheft Team 3 definiert und gelten für dieses Pflichtenheft grundsätzlich unverändert.

Maßgeblicher Akteur ist der Einzelanwender bzw. die Einzelanwenderin. Diese Person nutzt die Anwendung, um Angebote zu erstellen und aus bestehenden Angeboten Rechnungen zu erzeugen.

Angrenzende Komponenten und Systeme sind die grafische Benutzeroberfläche, die Kundenverwaltung, die Produkt- und Bestandsverwaltung, das lokale Dateisystem sowie die Druck- und PDF-Exportfunktion. Die Kommunikation erfolgt über fachlich definierte Schnittstellen, nicht über direkte Datenbankzugriffe der Oberfläche.

## 4. Funktionale Anforderungen

Die folgenden Anforderungen leiten sich direkt aus den Lastenheft-Anforderungen BA-03, BA-04 sowie den Geschäftsregeln GR-01 bis GR-06 ab. Nicht benötigte Funktionen wurden nicht zusätzlich spezifiziert, um den Belegworkflow nicht unnötig zu überladen.

## 4.1 Angebot erstellen

F-01: Das System MUSS es ermöglichen, ein Angebot für einen vorhandenen Kunden mit mindestens einer Produktposition zu erstellen.

F-02: WENN ein Angebot gespeichert wird, DANN MUSS das System eine eindeutige Angebotsnummer, das Erstellungsdatum, ein Gültigkeitsdatum und den Status OFFEN setzen.

F-03: Das System MUSS für jedes Angebot die Nettosumme, Umsatzsteuersumme und Bruttosumme automatisch aus den Dokumentpositionen berechnen.

F-04: WENN eine Produktposition in ein Angebot übernommen wird, DANN MUSS das System Produktbezeichnung, Netto-Einzelpreis und Steuersatz als Snapshot in der Dokumentposition speichern.

## 4.2 Angebot in Rechnung überführen

F-05: WENN aus einem Angebot eine Rechnung erzeugt wird, DANN MUSS das System den Status des Angebots auf UEBERFUEHRT setzen und eine Rückreferenz zwischen Rechnung und Angebot speichern.

F-06: Das System MUSS es ermöglichen, aus einem vorhandenen Angebot eine Rechnung zu erzeugen, wobei Kundendaten, Positionen und Mengen unverändert aus dem Angebot übernommen werden.

F-07: WENN eine Rechnung erzeugt wird, DANN MUSS das System automatisch die nächste freie, fortlaufende und lückenlose Rechnungsnummer vergeben.

## 4.3 Rechnung und Rechtskonformität

F-08: Das System MUSS in jeder Rechnung die im Lastenheft geforderten Rechnungsdaten führen: Rechnungsnummer, Rechnungsdatum, Leistungsdatum, Kundenreferenz, Angebotsreferenz, Positionen, Netto-/Steuer-/Bruttosumme, Zahlungsziel, Status und die Pflichtangaben gemäß § 14 UStG.

F-09: Das System MUSS für jede Rechnung Netto-, Umsatzsteuer- und Bruttosumme automatisch berechnen und als BigDecimal-Werte speichern.

F-10: WENN eine Rechnung versendet oder festgeschrieben wurde, DANN MUSS das System jede inhaltliche Änderung an Rechnungsnummer, Kundendaten, Positionen und Beträgen verweigern.

## 4.4 Übergreifende Belegregeln

F-11: Das System MUSS es ermöglichen, gespeicherte Angebote und Rechnungen als druckbare PDF-Dateien in das lokale Dateisystem zu exportieren.

F-12: Das System MUSS über Serviceoperationen prüfen können, ob ein Kunde oder Produkt in einem aktiven oder archivierten Beleg referenziert wird, damit Löschsperrern gemäß GR-04 unterstützt werden.

## 5. Nicht-funktionale Anforderungen

Die nicht-funktionalen Anforderungen werden mit den IDs NF-01 bis NF-04 geführt. Die Benennung ist bewusst einfach gehalten und leitet sich aus den Qualitätsanforderungen Q-01 bis Q-03 des Lastenhefts ab.

### 5.1 Benutzbarkeit

NF-01: Die Erstellung eines Angebots und die anschließende Erzeugung einer Rechnung aus diesem Angebot MUSS durch eine erstmalige Anwender:in ohne externe Hilfe innerhalb der im Lastenheft geforderten Kernprozess-Zeit von maximal 10 Minuten durchführbar sein.

### 5.2 Datensicherheit und Zuverlässigkeit

NF-02: Das System MUSS gespeicherte Angebote, Rechnungen und Dokumentpositionen dauerhaft lokal speichern, sodass sie nach einem Neustart der Anwendung unverändert verfügbar sind.

NF-03: Das System MUSS Belegdaten ausschließlich lokal verarbeiten und speichern. Eine automatische Übertragung von Kunden-, Produkt- oder Belegdaten an externe Dienste ist nicht zulässig.

### 5.3 Revisionssicherheit

NF-04: Das System MUSS finalisierte Rechnungen systemseitig vor nachträglicher Manipulation schützen. Der Schreibschutz ist auf Datenmodell- und Serviceebene durchzusetzen und darf nicht allein von der Benutzeroberfläche abhängen.

## 6. Daten und Schnittstellen

### 6.1 Designgrundsätze für Datentypen

- Geldbeträge werden als `java.math.BigDecimal` mit Scale 2 und kaufmännischer Rundung `RoundingMode.HALF_UP` geführt. `double` und `float` sind für Geldbeträge unzulässig.
- Steuersätze werden als `java.math.BigDecimal` geführt, z. B. 0.19 oder 0.07.
- IDs und Belegnummern werden als `String` geführt, da sie Präfixe und führende Nullen enthalten können.
- Datumswerte werden als `java.time.LocalDate` geführt.
- Mengen und Bestände werden als `int` geführt.
- Positionslisten werden als `List(Dokumentposition)` geführt.

### 6.2 Datenobjekte

#### 6.2.1 Schnittstellenobjekt: Kundendaten

Attribut	Java-Datentyp	Beschreibung / Validierung
kundenId	String	Eindeutige Kunden-ID, Referenz auf vorhandenen Kunden.

nameFirmenname	String	Name oder Firmenbezeichnung; für Belege erforderlich.
anschrift	String	Vollständige Anschrift; für Rechnungslegung erforderlich.
steuernummerUStId	String	Steuernummer oder USt-IdNr.; sofern im Kundenstamm gepflegt.
eMail	String	Optionale E-Mail-Adresse.
telefon	String	Optionale Telefonnummer.

### 6.2.2 Schnittstellenobjekt: Produktdaten

Attribut	Java-Datentyp	Beschreibung / Validierung
produktId	String	Eindeutige Produkt-ID, Referenz auf vorhandenes Produkt.
bezeichnung	String	Produktbezeichnung, wird als Snapshot in Dokumentposition übernommen.
nettoEinzelpreis	BigDecimal	Netto-Einzelpreis mit Scale 2.
mehrwertsteuersatz	BigDecimal	Mehrwertsteuersatz als Faktor, z. B. 0.19.
bestand	int	Aktueller Bestand; wird im Belegworkflow nur lesend berücksichtigt.
beschreibung	String	Optionale Produktbeschreibung.

### 6.2.3 Klasse: Dokumentposition

Attribut	Java-Datentyp	Beschreibung / Validierung
produktReferenz	String	Referenz auf produktId.
bezeichnung	String	Snapshot der Produktbezeichnung zum Erstellungszeitpunkt.
menge	int	Menge der Position; muss größer als 0 sein.
einzelpreisNetto	BigDecimal	Snapshot des Netto-Einzelpreises, Scale 2.
steuersatz	BigDecimal	Snapshot des Mehrwertsteuersatzes als Faktor.
positionssummeNetto	BigDecimal	Berechnetes Attribut: einzelpreisNetto × menge, Scale 2.

### 6.2.4 Abstrakte Basisklasse: Dokument

Attribut	Java-Datentyp	Beschreibung / Validierung
belegnummer	String	Eindeutige Nummer des Belegs.
datum	LocalDate	Erstellungsdatum.
kundenReferenz	String	Referenz auf kundenId.
positionen	List(Dokumentposition)	Mindestens eine Position.
summeNetto	BigDecimal	Summe aller Positionssummen, Scale 2.
summeSteuer	BigDecimal	Summe der Steuerbeträge, Scale 2.
summeBrutto	BigDecimal	summeNetto + summeSteuer, Scale 2.
festgeschrieben	boolean	true, wenn der Beleg nicht mehr inhaltlich geändert werden darf.

### 6.2.5 Konkrete Spezialisierungen

Klasse	Zusätzliche Attribute (Java-Typ)	Beschreibung
Angebot	gueltigBis: LocalDate status: AngebotStatus	Angebot mit Bindefrist. Statuswerte: OFFEN, UEBERFUEHRT, VERWORFEN.
Rechnung	rechnungsdatum: LocalDate leistungsdatum: LocalDate zahlungsziel: LocalDate angebotsNr: String status: RechnungStatus	Rechnung aus Angebot. Statuswerte: OFFEN, BEZAHLT. Der Schreibschutz wird zusätzlich über festgeschrieben abgebildet.

## 6.3 Interne Schnittstellen

Die Schnittstellen sind Java-nah formuliert, damit sie als direkter Input für den Komponententest genutzt werden können. Sie beschreiben keine konkrete Persistenztechnologie.

```
public interface KundenService {  
    Optional(Kundendaten) findeKunde(String kundenId);  
}
```

```
public interface ProductService {  
    Optional(ProduktDaten) findeProdukt(String produktId);  
}
```

```
public interface DokumentService {  
    Angebot erstelleAngebot(String kundenId, List<PositionEingabe> positionen, LocalDate
```

```

gueltigBis);
    Rechnung erstelleRechnungAusAngebot(String angebotsNr, LocalDate rechnungsdatum,
        LocalDate leistungsdatum, LocalDate zahlungsziel);
    void finalisiereRechnung(String rechnungsNr);
    Path exportierePdf(String belegNr, Path zielDatei) throws IOException;
    boolean istKundeReferenziert(String kundenId);
    boolean istProduktReferenziert(String produktId);
}

public interface BelegnummernGenerator {
    String naechsteAngebotsnummer(int jahr);
    String naechsteRechnungsnummer(int jahr);
}

public interface PdfExporter {
    void exportiere(Dokument dokument, Path zielDatei) throws IOException;
}

```

## 6.4 Externe Schnittstellen

Die ID-Bezeichnungen IF-01 bis IF-03 werden aus dem Lastenheft übernommen. Sie werden nicht neu nummeriert, damit die Traceability zwischen Lastenheft und Pflichtenheft eindeutig bleibt.

ID	Schnittstelle	Zweck und technischer Rahmen
IF-01	Benutzerschnittstelle (GUI)	Ruft die Serviceoperationen des Belegworkflows auf. Die konkrete Gestaltung der Oberfläche ist nicht Teil dieses Dokuments.
IF-02	Lokales Dateisystem	Persistente Speicherung der Belege und Ablage exportierter PDF-Dateien. Der Zugriff erfolgt z. B. über <code>java.nio.file.Path</code> .
IF-03	Druck-/Export-Schnittstelle	Erzeugung druckbarer PDF-Dokumente für Angebote und Rechnungen.

## 7. Systemarchitektur (logisch, grob)

Die Komponente folgt einer einfachen schichtenbasierten Architektur. Die Benutzeroberfläche ruft den `DokumentService` auf. Dieser enthält die fachliche Logik, validiert Eingaben, ruft Kunden- und Produktdaten über Schnittstellen ab, berechnet Summen, fordert Belegnummern an und delegiert Speicherung sowie PDF-Export an technische Hilfskomponenten.

## 7.1 Klassendiagramm

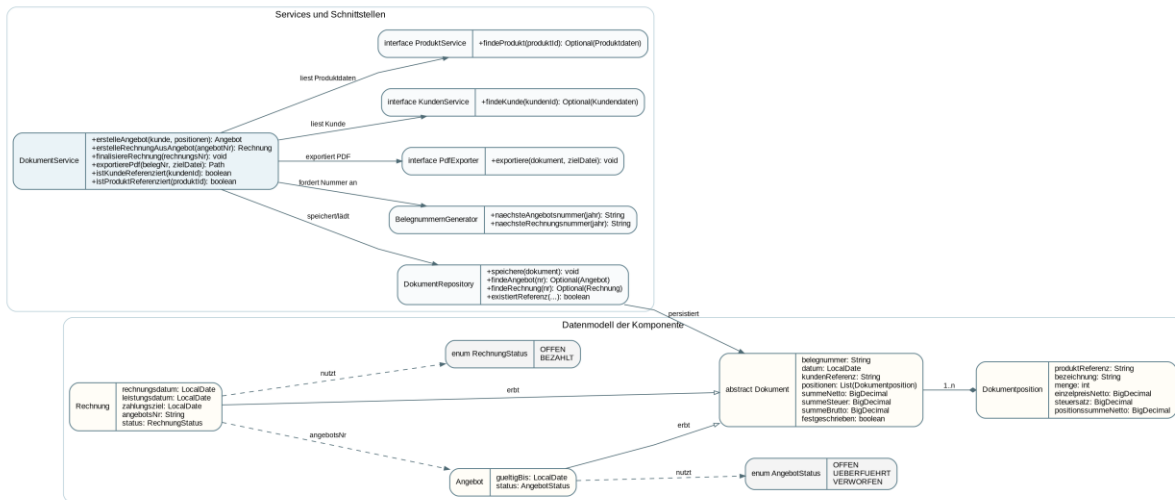


Abbildung 1: UML-Klassendiagramm der Komponente Belegworkflow

Abbildung 1 zeigt die zentralen Klassen und Schnittstellen der Komponente. Der DokumentService bildet den fachlichen Einstiegspunkt. Angebot und Rechnung erben gemeinsame Attribute aus der abstrakten Klasse Dokument. Dokumentposition speichert Produktdaten als Snapshot, sodass spätere Produktänderungen bestehende Belege nicht verändern.

## 7.2 Sequenzdiagramm

Sequenzdiagramm: Rechnung aus vorhandenem Angebot erzeugen und exportieren

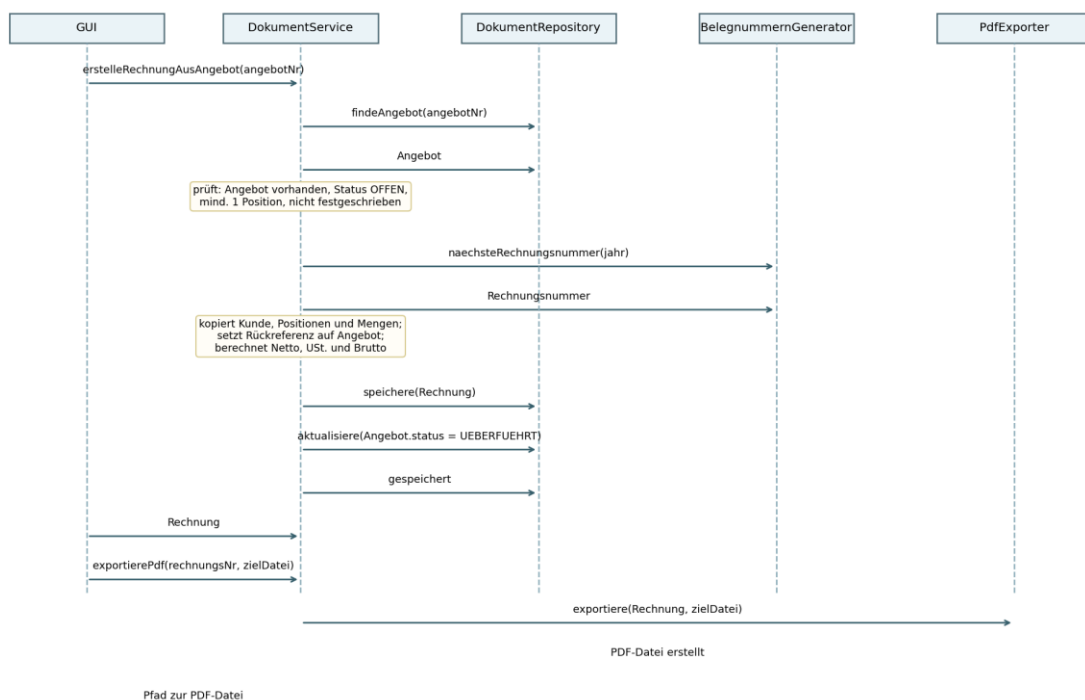


Abbildung 2: UML-Sequenzdiagramm "Rechnung aus Angebot erzeugen"

Abbildung 2 beschreibt den Hauptablauf aus BA-04. Die Benutzeroberfläche fordert die Rechnungserzeugung an. Der DokumentService lädt das Angebot, prüft den Status, übernimmt die Angebotsdaten, fordert eine neue Rechnungsnummer an, speichert die Rechnung und setzt das Angebot auf UEBERFUEHRT. Der PDF-Export erfolgt danach als eigener Schritt.

## 8. Testbare Abnahmekriterien

Die folgenden Abnahmekriterien verwenden die ID AC-01 bis AC-07. AC steht für Abnahmekriterium. Ein zusätzlicher Präfix ist nicht erforderlich, da dieses Dokument nur die Komponente Belegworkflow beschreibt.

ID	Bezug	Vorbedingung	Aktion	Erwartetes Ergebnis
AC-01	F-01, F-02, BA-03	Ein Kunde und mindestens ein Produkt existieren.	Ein Angebot mit einer Produktposition wird erstellt.	Das Angebot wird mit Angebotsnummer, Datum, Gültigkeitsdatum, Positionen und Status OFFEN gespeichert.
AC-02	F-03, GR-06	Ein Angebot enthält mehrere Positionen.	Das Angebot wird gespeichert.	Netto-, Steuer- und Bruttosumme werden korrekt berechnet und gespeichert.
AC-03	F-04, GR-03	Ein Produkt besitzt Preis 100.00 und Steuersatz 0.19.	Produkt wird in ein Angebot übernommen und danach der Produktpreis extern geändert.	Die Dokumentposition behält den ursprünglichen Preis und Steuersatz als Snapshot.
AC-04	F-06, BA-04	Ein gespeichertes Angebot im Status OFFEN existiert.	Aus dem Angebot wird eine Rechnung erzeugt.	Die Rechnung übernimmt Kunde, Positionen und Mengen und speichert die Angebotsreferenz.
AC-05	F-07, GR-01	Die letzte Rechnungsnummer eines Jahres ist bekannt.	Eine neue Rechnung wird erzeugt.	Das System vergibt die nächste freie, lückenlose Rechnungsnummer und verwendet keine alte Nummer erneut.
AC-06	F-08, F-10, Q-03	Eine Rechnung wurde erzeugt.	Die Rechnung wird finalisiert; danach wird eine Positionsänderung versucht.	Die Rechnung enthält die Pflichtdaten und lehnt die nachträgliche Änderung ab.
AC-07	F-11, IF-03	Ein Angebot oder eine Rechnung ist gespeichert.	Der Beleg wird als PDF exportiert.	Eine druckbare PDF-Datei wird im lokalen Dateisystem erzeugt.

## 9. Traceability: Lastenheft (LH) - Pflichtenheft (PH)

Die Traceability ordnet die für den Belegworkflow relevanten Lastenheft-Anforderungen den Anforderungen und Kapiteln dieses Pflichtenhefts zu. Anforderungen außerhalb des direkten Belegworkflow-Scopes werden nur über Schnittstellen berücksichtigt.

LH-ID	Beschreibung aus dem Lastenheft	Zuordnung im Pflichtenheft
BA-01	Kundendatensätze anlegen und für Angebote/Rechnungen verwenden.	Nicht direkte Umsetzung; Verwendung über Kundendaten/KundenService in Kapitel 6.2.1 und 6.3.
BA-02	Produktdaten mit Bezeichnung, Preis und Bestand speichern und in Dokumenten verwenden.	Nicht direkte Umsetzung; Verwendung über Produktdaten/ProduktService in Kapitel 6.2.2 und F-04.
BA-03	Angebot für vorhandenen Kunden mit mindestens einer Produktposition erstellen.	F-01 bis F-04, AC-01 bis AC-03.
BA-04	Aus einem vorhandenen Angebot eine Rechnung erzeugen.	F-05 bis F-10, AC-04 bis AC-06.
Q-01	Intuitive Bedienung der Kernprozesse innerhalb von 10 Minuten.	NF-01.
Q-02	Datensicherheit und Zuverlässigkeit; Daten bleiben erhalten und geschützt.	NF-02, NF-03, IF-02.
Q-03	Finale Rechnungen vor Manipulation schützen.	F-10, NF-04, AC-06.
GR-01	Fortlaufende, lückenlose Rechnungsnummern.	F-07, BelegnummernGenerator, AC-05.
GR-02	Unveränderlichkeit finaler Belege.	F-10, NF-04, AC-06.
GR-03	Snapshot-Prinzip für Preis und Steuersatz.	F-04, Dokumentposition, AC-03.
GR-04	Referenzielle Integrität für Kunden und Produkte.	F-12, DokumentService.istKundeReferenziert(), DokumentService.istProduktReferenziert().
GR-05	Dokumentenketten-Konsistenz bei Folgedokumenten.	F-05, F-06, Rechnung.angebotsNr, AC-04.
GR-06	Automatische Summenberechnung.	F-03, F-09, AC-02.
IF-01	Benutzerschnittstelle.	IF-01 in Kapitel 6.4; Serviceaufrufe über DokumentService.

IF-02	Lokales Dateisystem.	IF-02 in Kapitel 6.4; DokumentRepository/Persistenz.
IF-03	Druck-/Export-Schnittstelle.	F-11, IF-03, PdfExporter, AC-07.
AK-05	Aus bestehendem Angebot wird eine rechtskonforme Rechnung erzeugt.	AC-04, AC-05, AC-06.

## 10. Modultestplan

Die folgenden 10 Testfälle sind so formuliert, dass sie mit JUnit und Mocks für KundenService, ProductService und PdfExporter umgesetzt werden können.

TC	Abgedeckte PH-Anforderung	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-K-01	F-01, F-02	Kunde K-1 und Produkt P-1 existieren.	erstelleAngebot(K-1, Position P-1 Menge 2, gueltigBis)	Angebot wird gespeichert; Angebotsnummer, Datum, gueltigBis und Status OFFEN sind gesetzt.
TC-K-02	F-03, F-09	Positionen: 100.00 @ 0.19 und 50.00 @ 0.07.	Summenberechnung beim Speichern	summeNetto = 150.00, summeSteuer = 22.50, summeBrutto = 172.50.
TC-K-03	F-04, GR-03	Produkt P-1 hat Preis 100.00 und Steuersatz 0.19.	Produkt in Angebot übernehmen, danach Produktpreis im Mock ändern.	Dokumentposition behält einzelpreisNetto 100.00 und steuersatz 0.19.
TC-K-04	F-06, GR-05	Angebot AN-2026-000001 mit Kunde und 2 Positionen existiert.	erstelleRechnungAusAngebot(AN-2026-000001, ... )	Rechnung enthält dieselben Kundendaten, Positionen und Mengen; angebotsNr ist gesetzt.
TC-K-05	F-07, GR-01	Letzte Rechnungsnummer ist R-2026-000009.	Neue Rechnung erstellen.	Neue Rechnungsnummer lautet R-2026-000010 und ist String.
TC-K-05	F-05, GR-05	Angebot ist OFFEN.	Rechnung aus Angebot erzeugen.	Angebotsstatus wird UEBERFUEHRT.

06				
TC -K- 07	F-10, NF-04	Rechnung existiert und wird finalisiert.	Nach finalisiereRechnung() wird eine Position geändert.	Änderung wird mit IllegalStateException verweigert.
TC -K- 08	F-08	Rechnung wurde aus Angebot erzeugt.	Rechnung validieren.	Rechnungsnummer, Rechnungsdatum, Leistungsdatum, Kundenreferenz, Positionen, Summen, Zahlungsziel und Status sind vorhanden.
TC -K- 09	F-11, IF-03	Gespeicherte Rechnung existiert; PdfExporter ist gemockt.	exportierePdf(rechnungsNr, zielDatei)	PdfExporter.exportiere(...) wird mit Rechnung und Path aufgerufen; Rückgabe enthält Zielpfad.
TC -K- 10	F-12, GR-04	Repository enthält einen Beleg mit Produkt P-1 und Kunde K-1.	istProduktReferenziert(P-1), istKundeReferenziert(K-1)	Beide Aufrufe liefern true; unbekannte IDs liefern false.

## 11. Anhänge

### 11.1 Abkürzungen

Abkürzung	Bedeutung
F	Funktionale Anforderung
NF	Nicht-funktionale Anforderung
IF	Interface / Schnittstelle
AC	Abnahmekriterium
TC	Testfall
BA	Benutzeranforderung im Lastenheft
Q	Qualitätsanforderung
GR	Geschäftsregel
PH	Pflichtenheft
LH	Lastenheft

GUI	Graphical User Interface
UML	Unified Modeling Language
UStG	Umsatzsteuergesetz
GoBD	Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form
DSGVO	Datenschutz-Grundverordnung

## 11.2 Begriffe

Begriff	Bedeutung
Belegworkflow	Fachlicher Ablauf zur Erstellung von Angeboten und Rechnungen.
Angebot	Kaufmännisches Dokument mit Kunde, Positionen, Preisen und Summen vor Rechnungsstellung.
Rechnung	Dokument zur Zahlungsforderung gegenüber dem Kunden mit Pflichtangaben gemäß § 14 UStG.
Dokumentposition	Einzelne Belegzeile mit Produktreferenz, Menge, Preis, Steuersatz und Positionssumme.
Snapshot-Prinzip	Preis, Steuersatz und Bezeichnung werden beim Hinzufügen zur Position kopiert und bleiben später unverändert.
Festgeschrieben	Zustand, in dem ein Beleg nicht mehr inhaltlich verändert werden darf.
BigDecimal	Java-Datentyp für präzise Geldbeträge.
LocalDate	Java-Datentyp zur Speicherung eines Datums ohne Uhrzeit.