

Modultestplan

Desktop-Fakturierungsanwendung — Gesamtsystem (alle Komponenten)

Team 1

Inhaltsverzeichnis

Dokumentenhistorie	2
1. Einleitung	3
1.1 Zweck des Dokuments	3
1.2 Rahmenbedingungen	3
2. Testfälle	4
2.1 Komponente A — Prozess / Dokumentenzyklus	4
2.2 Komponente B — Produktverwaltung	7
2.3 Komponente C — Kundenverwaltung	9
2.4 Komponente D — Programmoberfläche	10
2.5 Gemeinsame Infrastruktur (Paket gemeinsam)	12
2.6 Performance-/Lastnachweise (PerformanceTest)	13
3. Testumfang-Übersicht	14
4. Abkürzungen	15

Autor	Prüfer	Freigebender
Strubel, Lucas	Prof. Dr. Marmitt, Gerd	Prof. Dr. Marmitt, Gerd
Team 1 (Gesamtsystem)	Modulverantwortlicher	Modulverantwortlicher
15.06.2026	15.06.2026	15.06.2026

Freigabevermerk: Dieses Dokument ist nach Prüfung und Freigabe durch den Modulverantwortlichen verbindliche Grundlage für den Modultest des Gesamtsystems *Desktop-Fakturierungsanwendung* (Komponenten A–D, gemeinsame Infrastruktur sowie Performance-Nachweise).

Dokumentenhistorie

Version	Datum	Autor	Grund der Änderung
1.0	15.06.2026	Lucas Strubel	Initiale Erstellung (ausgliedert aus Pflichtenheft Gruppe A v1.2)
2.0	15.06.2026	Lucas Strubel	Konsolidierung aller Komponenten-Testfälle (A–D, gemeinsame Infrastruktur, Performance) in einen projektweiten Modultestplan

1. Einleitung

1.1 Zweck des Dokuments

Dieser Modultestplan spezifiziert die Modul-/Komponententests des **Gesamtsystems Desktop-Fakturierungsanwendung**. Er fasst die Testfälle aller vier Komponenten (A – Prozess/Dokumentenzyklus, B – Produktverwaltung, C – Kundenverwaltung, D – Programmoberfläche), der gemeinsam genutzten Infrastruktur (Paket gemeinsam) sowie die übergreifenden Performance-/Lastnachweise in einem Dokument zusammen. Die Testfälle leiten sich aus den funktionalen Anforderungen, Daten/Schnittstellen und testbaren Abnahmekriterien der jeweiligen Pflichtenhefte ab und überführen diese in deterministische, mit JUnit 5 umgesetzte Testfälle. Quelle der Wahrheit sind die Testklassen unter `src/test/java/de/team1/faktura/...`

1.2 Rahmenbedingungen

Die folgenden Testfälle sind deterministisch (feste Ein-/Ausgaben) und mit JUnit 5 umgesetzt. Geldbeträge werden als `java.math.BigDecimal` mit **Scale 2** erwartet (`assertEquals(new BigDecimal("119.00"), ...)` bzw. `compareTo`). Die jeweils benachbarten Komponenten werden im Modultest durch **Stubs/Mocks** ersetzt:

- **Komponente A** ersetzt die Schnittstellen der Gruppen B (Produkte) und C (Kunden) sowie den PDF-Export durch Stubs.
- **Komponente B** ersetzt die Schnittstelle `ProduktReferenzPruefung` (Gruppe A) durch einen Stub.
- **Komponente C** ersetzt die Schnittstelle `KundenReferenzPruefung` (Gruppe A) durch einen Stub.
- **Komponente D** ersetzt die Service-Schnittstellen der Gruppen A–C durch Stubs/Mocks.

Die Abkürzungen für die abgedeckten Pflichtenheft-Anforderungen (Spalte *Abgedeckte PH-Anf.*) sind **komponentenlokal** zu lesen: F-01 in Abschnitt 2.2 bezeichnet eine Anforderung des Pflichtenhefts B, nicht des Pflichtenhefts A.

2. Testfälle

2.1 Komponente A — Prozess / Dokumentenzyklus

2.1.1 Dokumentenzyklus (DokumentzyklusTest)

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes 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(R ECHNUNG, 2026)	liefert R-2026-000124 (lückenlos)
TC-05	F-12 (Format)	Zähler = 7, Jahr 2026	naechsteNummer(R ECHNUNG, 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	setzePositionen(...) / Änderung	wirft IllegalStateException

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-09	F-19, F-20	Rechnung im Status OFFEN	storniere()	Status = STORNIERT; nicht in offeneRechnungen(); storniertAm und storniertVon gesetzt
TC-10	F-22, GR-05	Angebot AN-2026-00001 mit Kunde + 2 Positionen	erzeugeFolgebeleg(angebot) (AB erzeugen)	AB übernimmt Kunde/Positionen/Mengen; v orgaengerNr = "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	erstelleRechnung (...)	Speichern abgelehnt; Validierungsfehler benennt fehlendes Pflichtfeld (Kunde bzw. Position)
TC-13	F-11, F-12, F-13	Kunde + 1 Position vorhanden	vollständige Rechnung erstellen	Rechnung gespeichert, Nummer R-2026-... vergeben, alle § 14 UStG-Pflichtangaben gesetzt

2.1.2 Belegpersistenz (JsonDokumentRepositoryTest)

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-14	IF-01 (GoBD)	Repository mit gespeicherter Rechnung R-2026-00 0001 (OFFEN) und Angebot AN-2026-00 0001 (ENTWURF)	Repository neu instanzieren (Neustart)	beide Belege geladen (al le().size() == 2); polymorphe Typen Rech nung/Angebot bleiben erhalten; Positionen ≠ nu ll

2.1.3 PDF-Export (PdfBoxPdfExporterTest)

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-15	F-15, IF-01	Vollständige Rechnung R-2026-00 0001	exportiere(rechn ung, ziel)	PDF-Datei existiert am Zielpfad
TC-16	F-15 (Ro- bustheit)	Rechnung R-2026-00 0099 mit Altdaten-Position (pr oduktReferenz/einz elpreisNetto/posit ionssummeNetto = n ull)	exportiere(rechn ung, ziel)	kein Fehler (assertDoes NotThrow); PDF-Datei existiert
TC-17	F-23 (Ro- bustheit)	Rechnung R-2026-00 0098 mit null -Positionssummen	Summen berechnen	summeNetto = summeSteuer = summeBrutto = 0.00 (Scale 2)

2.1.4 CSV-Datenexport der Belege (DokumentCsvExportTest)

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-18	Q-08, IF-04	Repository mit 2 Rechnungen (je 1 Position); eine storniert (Datum 2026-06-10, Benutzer „Anwender“)	exportiereCsv(zi el)	CSV mit Kopfzeile bele gnummer;belegtyp;dat um;status... + je Position eine Zeile (3 Zeilen); enthält R-2026- 000001 sowie STORNIERT /Anwender

2.2 Komponente B — Produktverwaltung

PH-Anf.-Nummern beziehen sich auf das Pflichtenheft Gruppe B. Die Schnittstelle ProduktReferenzPruefung (Gruppe A) wird durch einen Stub ersetzt.

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-01	F-01, F-02	Höchste Produktnummer P-00 0041	Produkt („Beratungsstunde“, 80.00, 0.19) speichern	Produkt persistiert; Produktnummer = P-000 042
TC-02	F-02 (Format)	Zähler = 7	naechsteNummer()	liefert P-000007 (führende Nullen, Strin g)
TC-03	F-03	gültiges Produkt	Einzelpreis -1.00	Speichern abgelehnt (Validierungsfehler „Einzelpreis“)
TC-04	F-03	gültiges Produkt	Steuersatz 0.15	Speichern abgelehnt (unzulässiger Steuersatz)
TC-05	F-04, NF- USE-01	Produkt ohne Bezeichnung	speichere()	Speichern abgelehnt; Validierungsfehler benennt „Bezeichnung“

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-06	F-05	Produkt P-000042 mit Preis 80.00	Preis auf 95.00 ändern, speichern	gespeichertes Produkt hat einzelpreisNetto = 95.00
TC-07	F-07	Produkt P-000042	Änderungsversuch der Produktnummer auf P-999999	wirft IllegalArgumentException / Änderung abgelehnt
TC-08	F-08	Produkt unverknüpft (Stub: istProduktReferenziert → false)	loescheProdukt("P-000011") mit Bestätigung	Produkt entfernt; nicht mehr in alleSortiertNachBezeichnung()
TC-09	F-09, F-10	Stub: istProduktReferenziert("P-000010") → true	loescheProdukt("P-000010")	Löschen abgelehnt; Produkt weiterhin vorhanden; Hinweis erzeugt
TC-10	F-11	Produkte „Zaun“, „Anker“, „Mast“	alleSortiertNachBezeichnung()	Reihenfolge: „Anker“, „Mast“, „Zaun“
TC-11	F-12	Produkt „Beratungsstunde“	suche("BERATUNG")	Trefferliste enthält „Beratungsstunde“ (case-insensitive, Teilstring)
TC-12	F-12	Produkt P-000042	suche("P-000042")	Trefferliste enthält genau dieses Produkt
TC-13	F-14	Kein Produkt P-999999 vorhanden	findeProdukt("P-999999")	liefert null
TC-14	F-15	3 Produkte im Bestand	exportiereCsv(ziel)	CSV-Datei mit Kopfzeile + 3 Datenzeilen, Semikolon-getrennt, UTF-8

2.3 Komponente C — Kundenverwaltung

PH-Anf.-Nummern beziehen sich auf das Pflichtenheft Gruppe C. Die Schnittstelle KundenReferenzPruefung (Gruppe A) wird durch einen Stub ersetzt.

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-01	F-01, F-02	Höchste Kundennummer K-000016	Kunde („Muster GmbH“, „Hauptstr. 1“, „68163“, „Mannheim“) speichern	Kunde persistiert; Kundennummer = K-000017
TC-02	F-02 (Format)	Zähler = 7	naechsteNummer()	liefert K-000007 (führende Nullen, String)
TC-03	F-03, NF-USE-01	Kunde ohne Ort	speichere()	Speichern abgelehnt; Validierungsfehler benennt „Ort“
TC-04	F-03	Kunde mit leerem Namen (" ")	speichere()	Speichern abgelehnt; Validierungsfehler benennt „Name“
TC-05	F-04	Kunde mit E-Mail "max.mustermann"	speichere()	Speichern abgelehnt (ungültiges E-Mail-Format)
TC-06	F-04	Kunde mit E-Mail "max@beispiel.de"	speichere()	Kunde gespeichert (gültiges Format)
TC-07	F-05	Kunde K-000017 mit Ort „Mannheim“	Ort auf „Heidelberg“ ändern, speichern	gespeicherter Kunde hat ort = „Heidelberg“
TC-08	F-07	Kunde K-000017	Änderungsversuch der Kundennummer auf K-999999	wirft IllegalArgumentException / Änderung abgelehnt

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-09	F-08	Stub: anzahlVerknuepfterDokumente → 0	loescheKunde("K-000011") mit Bestätigung	Kunde entfernt; nicht mehr in alleSortiertNachName()
TC-10	F-09, F-10, GR-04	Stub: anzahlVerknuepfterDokumente("K-000010") → 3	loescheKunde("K-000010")	Löschen abgelehnt; Kunde weiterhin vorhanden; Hinweis enthält Anzahl 3
TC-11	F-11	Kunden „Zimmer“, „Albrecht“, „Maier“	alleSortiertNachName()	Reihenfolge: „Albrecht“, „Maier“, „Zimmer“
TC-12	F-12	Kunde „Muster GmbH“	suche("MUSTER")	Trefferliste enthält „Muster GmbH“ (case-insensitive, Teilstring)
TC-13	F-12, F-14	Kunde K-000017 vorhanden; K-999999 nicht	suche("K-000017"); findeKunde("K-999999")	Treffer enthält K-000017; findeKunde liefert null
TC-14	F-15	3 Kunden im Bestand	exportiereCsv(ziel)	CSV-Datei mit Kopfzeile + 3 Datenzeilen, Semikolon-getrennt, UTF-8

2.4 Komponente D — Programmoberfläche

PH-Anf.-Nummern beziehen sich auf das Pflichtenheft Gruppe D. Getestet wird die GUI-freie Controller- und Modell-Schicht; die Service-Schnittstellen der Gruppen A–C werden durch Stubs/Mocks ersetzt.

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-01	F-09	Wizard neu gestartet	aktuellerSchritt lesen	KUNDE_WAEHLEN (erster Schritt)

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-02	F-09	Schritt 1 mit gewähltem Kunden	weiter() 4-mal mit gültigen Eingaben	Schrittfolge: POSITIONEN _ERFASSEN → DATEN_BE STAETIGEN → ZUSAMMEN FASSUNG → SPEICHERN
TC-03	F-10	Schritt 1, kein Kunde gewählt (kundenNr = null)	weiter()	Wechsel verhindert; Meld ung (FEHLER, "Kunde", ...) erzeugt
TC-04	F-10	Schritt 2, leere Positionsliste	weiter()	Wechsel verhindert; Meldung benennt „Position“
TC-05	F-10	Schritt 2, Position mit menge = 0	weiter()	Wechsel verhindert; Meldung benennt „Menge“
TC-06	F-11	Schritt 3 erreicht; Kunde K-000017, 1 Position erfasst	zurueck() bis Schritt 1	kundenNr und position en unverändert erhalten
TC-07	F-12	Schritt 4; Stub Dokum entService liefert Summen 200.00/38.00/238.00	Zusammenfassung erzeugen	Zusammenfassung enthält Kunde, Positionen, Mengen, 200.00/38.00/238.00, Rechnungsdatum, Zahlungsziel
TC-08	F-13	Schritt 5; gültiges Modell	speichern()	genau ein Aufruf erstel leRechnung(...) am Mock; Erfolgsmeldung enthält gelieferte Rechnungsnummer
TC-09	F-13 (Feh- lerfall)	Stub erstelleRechn ung wirft Validierungsfehler „Rechnungsdatum“	speichern()	keine Erfolgsmeldung; M eldung (FEHLER, "Rech nungsdatum", ...) dargestellt (F-05/F-16)

Abgedeckte				
TC	PH-Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-10	F-14	Dokumentliste mit Rechnungen in OFFEN, VERSENDET, STORNIERT	verfügbare Aktionen je Rechnung ermitteln	<i>Stornieren</i> nur bei Status OFFEN aktiviert
TC-11	F-15	Rechnung R-2026-000124	storniere() ohne Bestätigung; danach mit Bestätigung	ohne Bestätigung: kein Service-Aufruf; mit Bestätigung: genau ein Aufruf storniere("R-2026-000124")
TC-12	F-08	Beleg im Status VERSENDET	Änderungsaktionen ermitteln	alle inhaltlichen Änderungsaktionen deaktiviert; PDF-Export aktiviert
TC-13	F-06	Belege mit Status OFFEN (2×) und STORNIERT (1×)	Statusfilter OFFEN anwenden	Liste enthält genau die 2 offenen Belege
TC-14	F-03	Stub KundenService.suche("Muster") liefert 1 Treffer	Suchbegriff „Muster“ eingeben	Controller delegiert an KundenService.suche(...); Trefferliste enthält genau diesen Kunden
TC-15	F-03 (D-F-03)	Stub KundenService mit 1 Kunden „Muster GmbH“	kundenListe(""), (" "), (null), ("unbekannt")	leerer/fehlender Suchbegriff: gesamter Bestand (1); „Muster“: 1 Treffer; „unbekannt“: 0 Treffer

2.5 Gemeinsame Infrastruktur (Paket gemeinsam)

Querschnittliche Dienste, die von allen Komponenten genutzt werden. Der EreignisBus (Observer-Muster) ist in der Architektur der Komponente A (Pflichtenheft A §7) verortet; JsonPersistenz ist die gemeinsame atomare JSON-Ablage hinter allen Repositories (IF-

01).

2.5.1 Ereignisbenachrichtigung (EreignisBusTest)

Abgedeckte				
TC	Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
INF-01	Observer (PH-A §7)	EreignisBus mit 2 Beobachtern für Bereich KUNDEN	melde(KUNDEN) 2×	beide Beobachter genau 2× benachrichtigt
INF-02	Observer	Beobachter für KUNDE N und DOKUMENTE	melde(DOKUMENTE)	nur DOKUMENTE -Beobachter (1×); KUNDEN -Beobachter 0×
INF-03	Observer (Robustheit)	EreignisBus ohne Beobachter	melde(PRODUKTE)	wirkungslos, keine Exception

2.5.2 Atomare JSON-Persistenz (JsonPersistenzTest)

Abgedeckte				
TC	Anf.	Vorbedingung	Eingabe	Erwartetes Ergebnis
INF-04	IF-01 (atomar)	leeres Verzeichnis	schreibeAtomar(datei, ["a", "b"])	Zieldatei existiert; keine .tmp-Restdatei; Inhalt = ["a", "b"]
INF-05	IF-01	Datei mit Bestand ["alt"]	schreibeAtomar(datei, ["neu1", "neu2"])	Inhalt vollständig ersetzt = ["neu1", "neu2"]
INF-06	IF-01	Zielpfad mit fehlendem Unterordner	schreibeAtomar(unterordner/datei, ["a"])	fehlende Elternverzeichnisse angelegt; Datei existiert

2.6 Performance-/Lastnachweise (PerformanceTest)

Übergreifende Lastnachweise gemäß Lastenheft. Referenzgröße: 5.000 Kunden, 5.000 Produkte, 1.000 Belege (vorab geseedet; das Befüllen fließt nicht in die Messung ein).

Die Spalte *Erwartetes Ergebnis* nennt die einzuhaltende Zeitschranke.

ID	Abgedeckte			Erwartetes Ergebnis (Zeitschranke)
	Anf.	Vorbedingung	Eingabe	
Q-04	Q-04 (Start)	Seeding 5.000 Kunden / 5.000 Produkte / 1.000 Belege (je JSON)	drei Repositories laden	Laden in ≤ 5 s
Q-02	Q-02 (Suche)	Bestand wie Q-04	Kunden- /Produktsuche + Auflistung	abgeschlossen in ≤ 1 s
Q-03	Q-03 (PDF)	Rechnung mit 50 Positionen	exportiere(rec hnung, ziel)	PDF-Erstellung in ≤ 2 s
Q-08	Q-08 (Export)	Bestand wie Q-04	Vollexport Kunden + Produkte + Belege als CSV	abgeschlossen in ≤ 30 s

3. Testumfang-Übersicht

Testklasse	Komponente	Anzahl
dokumente/DokumentzyklusTest	A – Prozess/Dokumentenzyklus	13
dokumente/JsonDokumentRepositoryTest	A – Belegpersistenz (IF-01/GoBD)	1
dokumente/PdfBoxPdfExporterTest	A – PDF-Export (F-15)	3
dokumente/DokumentCsvExportTest	A – CSV-Export Belege (Q-08)	1
produkte/ProduktverwaltungTest	B – Produktverwaltung	14
kunden/KundenverwaltungTest	C – Kundenverwaltung	14
gui/OberflaechenControllerTest	D – Programmoberfläche	15

Testklasse	Komponente	Anzahl
gemeinsam/EreignisBusTest	Gemeinsame Infrastruktur (Observer)	3
gemeinsam/JsonPersistenzTest	Gemeinsame Infrastruktur (Persistenz)	3
PerformanceTest	Querschnitt (Q-02/Q-03/Q-04/Q-08)	4
Summe		71

Damit sind **71 Testfälle** über alle vier Komponenten, die gemeinsame Infrastruktur und die Performance-Nachweise spezifiziert. Sie decken die funktionalen Kernregeln, die zentralen Geschäftsregeln (GR-01...GR-06) sowie die Qualitäts-/Performanceanforderungen (Q-02, Q-03, Q-04, Q-08, Q-09) ab.

4. Abkürzungen

Abkürzung	Bedeutung
TC	Testfall (Test Case)
INF	Infrastruktur-Testfall (Paket gemeinsam)
F	Funktionale Anforderung (Pflichtenheft, komponentenlokal)
NF	Nicht-funktionale Anforderung (Pflichtenheft)
IF	Schnittstelle (Interface)
GR	Geschäftsregel (Lastenheft)
Q	Qualitätsanforderung (Lastenheft)
PH	Pflichtenheft
CSV	Comma-Separated Values (offenes Exportformat)
GoBD	Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern