Software Engineering 1 | Modultestplan | Kundenverwaltung

Fakturierungssystem | Team 3

Datum: 29.06.2026

Weiterleitung zum Git: https://gitty.informatik.hs-mannheim.de/3028363/SE1_Team_3

Inhalt

Freigabeübersicht

Autor	Freigebenden	Prüfer
Khazanovych, Christian	Prof. Dr. Marmitt, Gerd	Prof. Dr. Marmitt, Gerd
Team 3	Modulverantwortlicher	Modulverantwortlicher
29.06.2026	29.06.2026	29.06.2026

Dokumentenhistorie

Version	Datum	Autor	Grund der Änderung
1.0	29.06.2026	Khazanovych, Christian	Erstmalige Erstellung des Modultestplan

1. Testfälle

1.2 Modultestplan – Gruppe J - Kundenverwaltung

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	naechsteKundenId() 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 findenKunden“K-000001“) aufrufen	Datensatz ist vollständig und unverändert verfügbar
TC-06	F-05, F-06, F-07	Kunde „K-000001“ mit E-Mail alt@mail.de ist gepseichert	E-Mail auf neu@mail.de ändern und speichern	findenKunde(„K-000001“) gibt Datensatz mit E-Mail neu@mail.de zurück, ohne Neustart der Anwendung
TC-07	F-09	Drei Kunden gespeichert: „Khazanovych GmbH“, „Winkler AG“, „Senger KG“	sucheNachName (“Kh“) aufrufen	Rückgabe enthält ausschließlich „Khazanovych GmbH“, die anderen beiden Einträge sind nicht enthalten

TC-08	F-10	Kunde „K-000001“ ist gespeichert	findeKunde(“K-000001“) über KundenService aufrufen	Vollständiger Kundendatensatz wird zurückgegeben
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

1.3 Modultestplan – Gruppe L - Bestandsmanagement

TC	Abgedeckte PH-Anforderung	Vorbedingung	Eingabe	Erwartetes Ergebnis
TC-L-01	F-01, F-02, F-05	Leeres Repository	speichereProdukt(produkt) mit neuem Produkt (Preis 10.00, Bestand 50)	Produkt wird fehlerfrei gespeichert. Ein anschließendes findeProdukt(id) liefert exakt dieses Produkt zurück.
TC-L-02	F-05 (Fehlerfall)	Produkt "P-100" existiert bereits im Repository	speichereProdukt(produkt) mit einem neuen Produkt, das ebenfalls die ID "P-100" besitzt	Speichern wird verweigert (z. B. IllegalArgumentException), das alte Produkt "P-100" bleibt unverändert.
TC-L-03	F-03, F-04, F-06	Produkt "P-101" existiert (Preis: 10.00)	speichereProdukt(produkt) mit geänderten Werten für "P-101" (neuer Preis: 15.00)	Update ist erfolgreich. Nächster Abruf zeigt den aktualisierten Preis und die aktualisierten Werte.
TC-L-04	F-07	Repository enthält Produkte "P-A" und "P-B"	Aufruf von findeProdukt("P-A")	Das System liefert den korrekten Datensatz für "P-A" zurück; Suche nach unbekannter ID liefert Optional.empty().
TC-L-05	F-09 (Fehlerfall)	Beliebiger Zustand	speichereProdukt(produkt) mit bestand = -5	Speichervorgang wird abgebrochen, System wirft eine IllegalArgumentException.
TC-L-06	F-08	Produkt "P-102" ist in Bearbeitung	speichereProdukt(produkt) mit bestand = 0	Speichern ist erfolgreich; das Produkt wird intern als "nicht verfügbar" gekennzeichnet (Attributabfrage liefert false).

TC-L-07	F-10	Produkt "P-103" existiert. Mock von Gruppe K (istProduktReferenziert) liefert false	loescheProdukt("P-103")	Löschvorgang ist erfolgreich. Ein anschließendes findeProdukt("P-103") liefert kein Ergebnis.
TC-L-08	F-11 (Löschsperre	Produkt "P-104" existiert. Mock von Gruppe K (ist ProduktRefernziert) liefert true	loescheProdukt("P-104")	System wirft zwingend eine IllegalStateException (Löschsperre). Das Produkt "P-104" bleibt im Repository unverändert erhalten.
TC-L-09	F-06, F-07	Repository enthält exakt 3 verschiedene Produkte	Aufruf von holeAlleProdukte()	Das System liefert eine Liste zurück, die exakt diese 3 Produkte enthält.
TC-L-10	F-01 (Fehlerfall)	Beliebiger Zustand	speichereProdukt(produkt) mit einem ungültigen negativen Netto-Einzelpreis (z.B. -10.00)	Speichervorgang wird abgebrochen, System wirft eine IllegalArgumentException, da Pflichtattribute valide sein müssen.

1.4 Modultestplan – Gruppe K - Belegworkflow

Die folgenden 10 Testfälle sind so formuliert, dass sie mit JUnit und Mocks für KundenService, ProduktService 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-06	F-05, GR-05	Angebot ist OFFEN.	Rechnung aus Angebot erzeugen.	Angebotsstatus wird UEBERFUEHRT.
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.

1.5 Modultestplan – Gruppe I - Benutzeroberfläche

Der Modultestplan beschreibt Testfälle für das Modul Benutzeroberfläche. Die Testfälle prüfen, ob die Oberfläche die geforderten Funktionen korrekt anzeigt, Eingaben richtig behandelt und dem Anwender verständliche Rückmeldungen gibt.

Die Tests können manuell über die Oberfläche durchgeführt werden. Einzelne Prüfungen, zum Beispiel Validierungen, können zusätzlich automatisiert auf Controller, ViewModel oder Service Ebene getestet werden.

Testfall ID	**Anforderung**	**Testziel**	**Vorbedingung**	**Testschritte**	**Erwartetes Ergebnis**
MT-UI-01	PH-UI-01	Start des Hauptfensters prüfen	Anwendung ist installiert	Anwendung starten	Hauptfenster wird angezeigt und Navigation ist sichtbar
MT-UI-02	PH-UI-01	Navigation zur Kundenmaske prüfen	Hauptfenster ist geöffnet	Menüpunkt Kunden anklicken	Kundenübersicht wird angezeigt
MT-UI-03	PH-UI-01	Navigation zur Produktmaske prüfen	Hauptfenster ist geöffnet	Menüpunkt Produkte anklicken	Produktübersicht wird angezeigt
MT-UI-04	PH-UI-02	Kunde mit gültigen Daten speichern	Kundenmaske ist geöffnet	Name und Anschrift eingeben, Speichern anklicken	Kunde wird gespeichert und Erfolgsmeldung erscheint
MT-UI-05	PH-UI-02, PH-UI-07	Kunde ohne Pflichtfeld ablehnen	Kundenmaske ist geöffnet	Name oder Anschrift leer lassen, Speichern anklicken	Speicherung wird verhindert und Fehlermeldung erscheint
MT-UI-06	PH-UI-03	Produkt mit gültigen Daten speichern	Produktmaske ist geöffnet	Bezeichnung, Preis, Steuersatz und Bestand eingeben, Speichern anklicken	Produkt wird gespeichert und Erfolgsmeldung erscheint
MT-UI-07	PH-UI-03, PH-UI-07	Negativen Produktpreis ablehnen	Produktmaske ist geöffnet	Negativen Preis eingeben,	Speicherung wird verhindert und

Testfall ID	Anforderung	Testziel	Vorbedingung	Testschritte	Erwartetes Ergebnis
				Speichern anklicken	Fehlermeldung erscheint
MT-UI-08	PH-UI-03, PH-UI-07	Negativen Bestand ablehnen	Produktmaske ist geöffnet	Negativen Bestand eingeben, Speichern anklicken	Speicherung wird verhindert und Fehlermeldung erscheint
MT-UI-09	PH-UI-04	Dokumentenübersicht anzeigen	Es existiert mindestens ein Dokument	Dokumentenbereich öffnen	Dokumente werden tabellarisch angezeigt
MT-UI-10	PH-UI-05	Angebot mit Kunde und Position erstellen	Kunde und Produkt existieren	Kunde auswählen, Produktposition hinzufügen, Speichern anklicken	Angebot wird erstellt und Summenbereich wird angezeigt
MT-UI-11	PH-UI-05, PH-UI-07	Angebot ohne Produktposition ablehnen	Kunde existiert	Kunde auswählen, keine Position hinzufügen, Speichern anklicken	Speicherung wird verhindert und Fehlermeldung erscheint
MT-UI-12	PH-UI-06	Rechnung über Oberfläche erzeugen	Passendes Angebot existiert	Dokument öffnen, Aktion für Rechnungserstellung anklicken	Rechnung wird erzeugt und Erfolgsmeldung erscheint
MT-UI-13	PH-UI-08	Erfolgsmeldung prüfen	Gültige Eingabe liegt vor	Daten speichern	Verständliche Erfolgsmeldung wird angezeigt
MT-UI-14	PH-UI-08	Fehlermeldung prüfen	Ungültige Eingabe liegt vor	Speichern anklicken	Verständliche Fehlermeldung wird angezeigt

Testfall ID	Anforderung	Testziel	Vorbedingung	Testschritte	Erwartetes Ergebnis
MT-UI-15	PH-UI-09	Schreibschutz finalisierter Rechnung prüfen	Finalisierte Rechnung existiert	Rechnung öffnen und Bearbeitung versuchen	Eingabefelder sind deaktiviert und Rechnung kann nicht geändert werden
MT-UI-16	PH-UI-10	PDF Export starten	Dokument existiert	Dokument öffnen, Export Button anklicken	Dialog zur Auswahl des Speicherorts wird geöffnet
MT-UI-17	PH-Q-01	Bedienbarkeit prüfen	Testperson kennt das System nicht	Testperson legt Kunde, Produkt und Dokument an	Aufgaben werden innerhalb von maximal 10 Minuten ohne Hilfe abgeschlossen
MT-UI-18	PH-Q-05	Offline Nutzung prüfen	Internetverbindung ist deaktiviert	Anwendung starten und Hauptfunktionen öffnen	Oberfläche ist ohne Internetverbindung nutzbar