

# Pflichtenheft Benutzeroberfläche Fakturierungssystem

Software Engineering 1 Pflichtenheft zum Lastenheft Team 3 Modul: Benutzeroberfläche Version: 1.0 Datum: 12.06.2026

## Inhalt

1. Metainformationen
2. Einleitung und Bezug zum Lastenheft
3. Systemübersicht
4. Funktionale Anforderungen
5. Nichtfunktionale Anforderungen
6. Daten und Schnittstellen
7. Systemarchitektur
8. Abnahmekriterien
9. Traceability Matrix
10. Modultestplan
11. Glossar

## 1. Metainformationen

### 1.1 Dokumentinformationen

Feld	Inhalt
Dokumentname	Pflichtenheft Benutzeroberfläche Fakturierungssystem
Version	1.0
Datum	12.06.2026
Projekt	Lokales Fakturierungssystem
Modul	Benutzeroberfläche
Bezug	Lastenheft Team 3, Version 1.0
Gruppe	Gruppe I
Autorin	Meltem Bardakci

## 1.2 Freigabeübersicht

Rolle	Name	Datum
Autorin	Meltem Bardakci	12.06.2026
Prüfer	Prof. Dr. Marmitt	

## 1.3 Änderungshistorie

Version	Datum	Autorin	Änderung
1.0	12.06.2026	Meltem Bardakci	Initiale Erstellung des Pflichtenhefts für das Modul Benutzeroberfläche

## 2. Einleitung und Bezug zum Lastenheft

### 2.1 Zweck des Dokuments

Dieses Pflichtenheft beschreibt die technische Umsetzung der Benutzeroberfläche des Fakturierungssystems. Es basiert auf dem Lastenheft von Team 3 und konkretisiert die dort beschriebenen Anforderungen aus Sicht der Umsetzung.

Der Schwerpunkt dieses Pflichtenhefts liegt auf der Bedienung durch den Einzelanwender. Dazu gehören Navigation, Eingabemasken, Übersichtstabellen, Validierung, Fehlermeldungen, Statusanzeigen und die einfache Durchführung der Kernprozesse.

### 2.2 Ziel des UI Moduls

Ziel des UI Moduls ist eine übersichtliche und verständliche Desktop Oberfläche. Der Einzelanwender soll die wichtigsten Funktionen des Fakturierungssystems ohne lange Einarbeitung bedienen können.

Die Oberfläche soll folgende Aufgaben unterstützen:

- Kunden erfassen und anzeigen
- Produkte erfassen und anzeigen
- Angebote erstellen
- Auftragsbestätigungen und Lieferscheine anzeigen oder erzeugen
- Rechnungen aus vorhandenen Angeboten erzeugen
- Pflichtfelder prüfen
- Fehlermeldungen verständlich anzeigen
- finalisierte Rechnungen als schreibgeschützt kennzeichnen
- PDF Export über eine klare Bedienaktion starten

## 2.3 Abgrenzung

Dieses Pflichtenheft beschreibt nur das Modul Benutzeroberfläche. Die interne Speicherung, die vollständige Beleglogik und die steuerrechtliche Detailprüfung werden nur so weit beschrieben, wie sie für die Bedienoberfläche sichtbar sind.

Nicht Bestandteil dieses Moduls sind:

- externe Cloud Anbindung
- Mehrbenutzerverwaltung
- mobile Oberfläche
- Web Anwendung
- vollständige Buchhaltung
- Schnittstellen zu Finanzamt oder Buchhaltungssystemen

## 2.4 Referenzen

- Lastenheft Fakturierungssystem, Team 3, Version 1.0
- Vorlesungsunterlagen Software Engineering 1, Thema Lastenheft und Pflichtenheft
- Java Dokumentation zu Datentypen String, int, BigDecimal und List
- UML Grundlagen für Klassen und Sequenzdiagramme

## 2.5 Stakeholder und Kontext

Maßgeblicher Akteur ist der Einzelanwender bzw. die Einzelanwenderin des Fakturierungssystems. Diese Person nutzt die Benutzeroberfläche, um Kunden, Produkte und Dokumente aufzurufen, Daten einzugeben, Angebote zu erstellen, Rechnungen aus vorhandenen Angeboten zu erzeugen und PDF Dateien zu exportieren.

Angrenzende Komponenten sind die Kundenverwaltung, die Produktverwaltung, der DocumentService, der ValidationService, der MessageService und der PdfExportService. Die Benutzeroberfläche ruft diese Komponenten über definierte Schnittstellen auf und stellt deren Ergebnisse sichtbar dar.

Nicht Aufgabe der Benutzeroberfläche ist die dauerhafte Speicherung oder fachliche Berechnung der Belegdaten. Diese Aufgaben liegen bei den fachlichen Services.

## 3. Systemübersicht

Das Fakturierungssystem wird als lokale Desktop Anwendung umgesetzt. Der Einzelanwender arbeitet über eine grafische Benutzeroberfläche mit dem System.

Die Benutzeroberfläche ist die zentrale Schnittstelle zwischen Anwender und Systemfunktionen. Über die Oberfläche kann der Anwender Kunden, Produkte und Dokumente verwalten. Außerdem erhält er Rückmeldungen zu erfolgreichen Aktionen, fehlerhaften Eingaben und dem Status von Dokumenten.

Das UI Modul besteht aus folgenden Bereichen:

Bereich	Aufgabe
Hauptfenster	Einstiegspunkt der Anwendung
Navigation	Wechsel zwischen Kunden, Produkten und Dokumenten
Kundenmaske	Eingabe und Anzeige von Kundendaten
Produktmaske	Eingabe und Anzeige von Produktdaten
Dokumentenmaske	Erstellung und Anzeige von Angeboten, Auftragsbestätigungen, Lieferscheinen und Rechnungen
Validierungsanzeige	Anzeige fehlender oder ungültiger Eingaben
Statusanzeige	Anzeige von Erfolgsmeldungen, Warnungen und Fehlern
Exportdialog	Auswahl des Speicherorts für PDF Dateien
Schreibschutzanzeige	Kennzeichnung finalisierter Rechnungen

Die Oberfläche verarbeitet selbst keine dauerhafte Speicherung. Sie nimmt Eingaben entgegen, zeigt Daten an und übergibt Benutzeraktionen an die zuständigen Systemkomponenten.

#### 4. Funktionale Anforderungen

##### PH-UI-01 Hauptfenster und Navigation

Das System stellt ein Hauptfenster bereit, das nach dem Start der Anwendung angezeigt wird. Im Hauptfenster befindet sich eine Navigation zu den wichtigsten Bereichen.

Die Navigation enthält mindestens folgende Menüpunkte:

Menüpunkt	Ziel
Kunden	Öffnet die Kundenübersicht
Produkte	Öffnet die Produktübersicht
Dokumente	Öffnet die Dokumentenübersicht
Export	Öffnet Exportfunktionen oder startet den PDF Export bei geöffnetem Dokument

Die Navigation muss dauerhaft sichtbar oder schnell erreichbar sein. Der Anwender soll jederzeit erkennen können, in welchem Bereich er sich befindet.

Erfüllungskriterium: Nach dem Start der Anwendung ist das Hauptfenster sichtbar und der Anwender kann zwischen Kunden, Produkten und Dokumenten wechseln.

### PH-UI-02 Kundenmaske

Das System stellt eine Kundenmaske bereit. In dieser Maske kann der Einzelanwender Kundendaten erfassen, anzeigen und bearbeiten.

Die Kundenmaske enthält folgende Eingabefelder:

Feld	Datentyp	Pflichtfeld
Name oder Firmenname	String	Ja
Anschrift	String	Ja
Steuernummer oder USt IdNr	String	Nein
E-Mail	String	Nein
Telefon	String	Nein

Die Kundenmaske enthält folgende Schaltflächen:

Schaltfläche	Funktion
Speichern	Speichert den Kundendatensatz
Abbrechen	Verwirft die Eingabe
Bearbeiten	Aktiviert die Bearbeitung eines bestehenden Kunden
Zurück	Führt zurück zur Kundenübersicht

Erfüllungskriterium: Der Anwender kann Kundendaten über die Maske eingeben und speichern. Fehlende Pflichtfelder werden sichtbar markiert.

### PH-UI-03 Produktmaske

Das System stellt eine Produktmaske bereit. In dieser Maske kann der Einzelanwender Produktdaten erfassen, anzeigen und bearbeiten.

Die Produktmaske enthält folgende Eingabefelder:

<b>Feld</b>	<b>Datentyp</b>	<b>Pflichtfeld</b>
Bezeichnung	String	Ja
Netto Einzelpreis	BigDecimal	Ja
Mehrwertsteuersatz	BigDecimal	Ja
Beschreibung	String	Nein
Bestand	int	Ja

Die Produktmaske prüft, ob der Preis größer oder gleich 0 ist. Der Bestand darf nicht negativ sein.

Erfüllungskriterium: Der Anwender kann ein Produkt über die Maske anlegen. Ungültige Zahlenwerte werden abgelehnt und mit einer verständlichen Fehlermeldung angezeigt.

#### **PH-UI-04 Dokumentenübersicht**

Das System stellt eine Dokumentenübersicht bereit. Dort kann der Anwender vorhandene Dokumente einsehen.

Die Dokumentenübersicht zeigt mindestens folgende Informationen:

<b>Spalte</b>	<b>Beschreibung</b>
Dokumentnummer	Nummer des Dokuments
Dokumenttyp	Angebot, Auftragsbestätigung, Lieferschein oder Rechnung
Kunde	Zugeordneter Kunde
Datum	Erstellungsdatum
Status	offen, überführt, festgeschrieben, bezahlt oder verworfen
Bruttosumme	Gesamtbetrag, sofern für den Dokumenttyp relevant

Der Anwender kann Dokumente auswählen und öffnen.

Erfüllungskriterium: Vorhandene Dokumente werden tabellarisch angezeigt und können geöffnet werden.

#### **PH-UI-05 Dokumentenmaske zur Erstellung von Angeboten**

Das System stellt eine Dokumentenmaske zur Erstellung von Angeboten bereit. Der Anwender wählt einen vorhandenen Kunden und mindestens eine Produktposition aus.

Die Maske enthält folgende Bereiche:

Bereich	Inhalt
Kundenauswahl	Auswahl eines vorhandenen Kunden
Positionsbereich	Hinzufügen von Produkten mit Menge
Summenbereich	Anzeige von Netto, Steuer und Brutto
Aktionsbereich	Speichern, Abbrechen und PDF Export

Die Oberfläche verhindert, dass ein Angebot ohne Kunde oder ohne Produktposition gespeichert wird.

Erfüllungskriterium: Der Anwender kann ein Angebot über die Oberfläche erstellen. Die Summen werden sichtbar angezeigt.

#### **PH-UI-06 Rechnung aus Angebot erzeugen**

Das System stellt in der Oberfläche eine Aktion bereit, mit der der Anwender aus einem vorhandenen Angebot eine Rechnung erzeugen kann.

Der Workflow umfasst:

Ausgangsdokument	Folgedokument
Angebot	Rechnung

Die Oberfläche zeigt die Aktion zur Rechnungserstellung nur an, wenn ein passendes Angebot ausgewählt wurde. Die Benutzeroberfläche erzeugt die Rechnung nicht selbst, sondern ruft dafür die Methode `createInvoiceFromOffer(String offerId)` des `DocumentService` auf.

Erfüllungskriterium: Der Anwender kann über eine sichtbare Schaltfläche aus einem vorhandenen Angebot eine Rechnung erzeugen.

#### **PH-UI-07 Eingabevalidierung**

Das System prüft Eingaben spätestens beim Speichern. Fehlerhafte Eingaben werden sichtbar angezeigt.

Folgende Validierungen werden über die Oberfläche sichtbar gemacht:

Bereich	Prüfung
Kunde	Name und Anschrift dürfen nicht leer sein
Produkt	Bezeichnung, Preis, Steuersatz und Bestand müssen gültig sein
Dokument	Kunde und mindestens eine Position müssen vorhanden sein
Position	Menge muss größer als 0 sein
Rechnung	Finalisierte Rechnung darf nicht bearbeitet werden

Fehlerhafte Felder werden markiert. Zusätzlich wird eine kurze Fehlermeldung angezeigt.

Erfüllungskriterium: Ungültige Eingaben werden nicht gespeichert und der Anwender erhält eine verständliche Rückmeldung.

#### PH-UI-08 Fehlermeldungen und Erfolgsmeldungen

Das System zeigt klare Rückmeldungen an.

Beispiele für Erfolgsmeldungen:

Situation	Meldung
Kunde gespeichert	Kunde wurde erfolgreich gespeichert
Produkt gespeichert	Produkt wurde erfolgreich gespeichert
Angebot erstellt	Angebot wurde erfolgreich erstellt
Rechnung erzeugt	Rechnung wurde erfolgreich erzeugt

Beispiele für Fehlermeldungen:

Situation	Meldung
Pflichtfeld leer	Bitte füllen Sie alle Pflichtfelder aus
Preis negativ	Der Preis darf nicht negativ sein
Menge ungültig	Die Menge muss größer als 0 sein
Rechnung finalisiert	Diese Rechnung ist finalisiert und kann nicht bearbeitet werden

Erfüllungskriterium: Jede erfolgreiche oder fehlerhafte Aktion erzeugt eine verständliche Rückmeldung.

#### **PH-UI-09 Schreibschutzanzeige bei finalisierten Rechnungen**

Finalisierte Rechnungen werden in der Oberfläche schreibgeschützt angezeigt. Eingabefelder und Bearbeitungsfunktionen sind deaktiviert.

Die Oberfläche zeigt zusätzlich einen sichtbaren Hinweis an:

<b>Status</b>	<b>Anzeige</b>
Finalisiert	Diese Rechnung ist finalisiert und schreibgeschützt

Folgende Daten dürfen bei finalisierten Rechnungen nicht mehr bearbeitet werden:

- Rechnungsnummer
- Kundendaten
- Positionen
- Nettosumme
- Steuerbetrag
- Bruttosumme
- Leistungsdatum

Erfüllungskriterium: Eine finalisierte Rechnung kann über die Oberfläche nicht mehr verändert werden.

#### **PH-UI-10 PDF Export Bedienung**

Das System stellt eine sichtbare Exportfunktion bereit. Der Anwender kann aus einem geöffneten Dokument eine PDF Datei erzeugen.

Die Oberfläche zeigt einen Exportbutton an. Nach dem Klick öffnet sich ein Dialog zur Auswahl des Speicherorts.

Erfüllungskriterium: Der Anwender kann den PDF Export über die Oberfläche starten und einen lokalen Speicherort auswählen.

### **5. Nichtfunktionale Anforderungen**

#### **PH-Q-01 Benutzerfreundlichkeit**

Die Oberfläche muss so gestaltet sein, dass ein neuer Einzelanwender die wichtigsten Kernprozesse ohne technische Vorkenntnisse durchführen kann.

Die Kernprozesse sind:

- Kunde anlegen
- Produkt anlegen
- Angebot erstellen
- Rechnung aus vorhandenem Angebot erzeugen

Erfüllungskriterium: Ein neuer Anwender kann diese Prozesse in einem Usability Test ohne Hilfestellung innerhalb von maximal 10 Minuten durchführen.

### PH-Q-02 Übersichtlichkeit

Die Oberfläche muss klar strukturiert sein. Zusammengehörige Informationen werden gruppiert dargestellt.

Bereich	Darstellung
Kundendaten	In einer eigenen Kundenmaske
Produkt Daten	In einer eigenen Produktmaske
Dokumentdaten	In einer eigenen Dokumentenmaske
Summen	In einem sichtbaren Summenbereich
Fehler	In unmittelbarer Nähe zur Eingabe oder als Meldung

Erfüllungskriterium: Der Anwender erkennt ohne Suche, in welchem Bereich er sich befindet und welche Aktion möglich ist.

### PH-Q-03 Konsistenz

Schaltflächen, Begriffe und Statusanzeigen werden in der gesamten Oberfläche einheitlich verwendet.

Begriff	Verwendung
Speichern	Daten dauerhaft sichern
Abbrechen	Eingabe ohne Speicherung verlassen
Bearbeiten	Bestehende Daten ändern
Exportieren	PDF Datei erzeugen

Begriff	Verwendung
Finalisiert	Dokument ist schreibgeschützt

Erfüllungskriterium: Gleiche Aktionen werden in allen Masken gleich benannt.

#### PH-Q-04 Fehlertoleranz

Das System soll Fehleingaben verhindern oder frühzeitig erkennen. Fehlerhafte Eingaben dürfen nicht stillschweigend gespeichert werden.

Erfüllungskriterium: Bei jeder fehlerhaften Eingabe wird das Speichern verhindert und eine verständliche Fehlermeldung angezeigt.

#### PH-Q-05 Lokale Nutzbarkeit

Die Oberfläche muss ohne Internetverbindung nutzbar sein. Alle sichtbaren Funktionen des UI Moduls müssen lokal bedienbar sein.

Erfüllungskriterium: Die Anwendung startet und die Oberfläche ist ohne Internetverbindung nutzbar.

#### PH-Q-06 Lesbarkeit

Texte, Tabellen, Eingabefelder und Schaltflächen müssen gut lesbar sein. Pflichtfelder sollen eindeutig erkennbar sein.

Erfüllungskriterium: Pflichtfelder sind markiert und alle Beschriftungen sind verständlich formuliert.

### 6. Daten und Schnittstellen

#### 6.1 Datenobjekte in der Benutzeroberfläche

Die Benutzeroberfläche zeigt Daten aus den fachlichen Modulen an. Das UI Modul berechnet und speichert diese Daten nicht selbst dauerhaft. Es nimmt Eingaben entgegen, zeigt Daten an und übergibt Benutzeraktionen an die zuständigen Systemkomponenten.

Objekt	Sichtbare Daten in der Oberfläche
Kunde	Name, Anschrift, Steuernummer, E Mail, Telefon
Produkt	Bezeichnung, Netto Einzelpreis, Mehrwertsteuersatz, Beschreibung, Bestand
Dokumentposition	Produkt, Menge, Einzelpreis, Steuersatz, Positionssumme
Angebot	Angebotsnummer, Kunde, Datum, Positionen, Summen, Status
Auftragsbestätigung	AB Nummer, Kunde, Datum, Positionen, Status

Objekt	Sichtbare Daten in der Oberfläche
Lieferschein	Lieferscheinnummer, Kunde, Lieferdatum, Positionen, Status
Rechnung	Rechnungsnummer, Kunde, Rechnungsdatum, Leistungsdatum, Positionen, Summen, Zahlungsziel, Status

## 6.2 UI interne Datenmodelle

Für die Oberfläche werden interne UI Modelle verwendet. Diese Modelle dienen dazu, Eingaben aus Formularen zwischenspeichern und für die Validierung vorzubereiten.

UI Modell	Zweck
CustomerFormModel	Zwischenspeicherung der Eingaben in der Kundenmaske
ProductFormModel	Zwischenspeicherung der Eingaben in der Produktmaske
DocumentFormModel	Zwischenspeicherung der Eingaben in der Dokumentenmaske
ValidationResult	Rückgabe von Validierungsfehlern
UiMessage	Darstellung von Fehlern, Warnungen und Erfolgsmeldungen

### 6.2.1 Java nahe UI Modelle

Die folgenden Klassen beschreiben die Datenmodelle, die innerhalb der Benutzeroberfläche verwendet werden. Sie dienen zur Zwischenspeicherung von Formulareingaben und zur Übergabe an Validierungs- und Serviceklassen.

```
public class CustomerFormModel {
    private String name;
    private String anschrift;
    private String steuernummer;
    private String email;
    private String telefon;
}
```

```
public class ProductFormModel {
    private String bezeichnung;
```

```

private BigDecimal nettoPreis;

private BigDecimal mehrwertsteuerSatz;

private String beschreibung;

private int bestand;
}

public class DocumentFormModel {

    private String customerId;

    private List<DocumentPositionFormModel> positionen;

    private String status;
}

public class DocumentPositionFormModel {

    private String productId;

    private int menge;
}

```

```

public class ValidationResult {

    private boolean valid;

    private List<String> messages;
}

```

Diese Modelle dienen nur der Benutzeroberfläche. Die dauerhafte Speicherung und fachliche Verarbeitung erfolgen in den jeweiligen Services.

### **6.3 Schnittstellen des UI Moduls**

Das UI Modul kommuniziert mit mehreren fachlichen Services. Diese Services übernehmen die eigentliche Verarbeitung, Speicherung und Ausgabe der Daten.

#### **6.3.1 Java nahe Schnittstellendefinitionen**

Die folgenden Schnittstellen beschreiben, wie das UI Modul mit den fachlichen Komponenten kommuniziert. Die Angaben sind Java nah formuliert, damit sie als Grundlage für die spätere Implementierung und für Modultests verwendet werden können.

```
public interface CustomerService {  
  
    List<CustomerDto> loadCustomers();  
  
    CustomerDto saveCustomer(CustomerFormModel model);  
  
    CustomerDto updateCustomer(String customerId, CustomerFormModel model);  
  
}
```

```
public interface ProductService {  
  
    List<ProductDto> loadProducts();  
  
    ProductDto saveProduct(ProductFormModel model);  
  
    ProductDto updateProduct(String productId, ProductFormModel model);  
  
}
```

```
public interface DocumentService {  
  
    List<DocumentDto> loadDocuments();  
  
    OfferDto createOffer(DocumentFormModel model);  
  
    InvoiceDto createInvoiceFromOffer(String offerId);  
  
    DocumentDto loadDocument(String documentId);  
  
}
```

```
public interface ValidationService {  
  
    ValidationResult validateCustomer(CustomerFormModel model);  
  
    ValidationResult validateProduct(ProductFormModel model);  
  
    ValidationResult validateDocument(DocumentFormModel model);  
  
}
```

```
public interface PdfExportService {  
  
    Path exportDocumentAsPdf(String documentId, Path targetPath);  
  
}
```

Die Benutzeroberfläche greift nicht direkt auf die lokale Speicherung zu. Sie ruft stattdessen Services auf und erhält Datenobjekte oder Validierungsergebnisse zurück.

ID	Schnittstelle	Zweck
UI-IF-01	CustomerService	Speichern, Laden und Anzeigen von Kundendaten
UI-IF-02	ProductService	Speichern, Laden und Anzeigen von Produktdaten
UI-IF-03	DocumentService	Erstellen und Anzeigen von Dokumenten
UI-IF-04	PdfExportService	Export eines Dokuments als PDF Datei
UI-IF-05	ValidationService	Prüfung von Eingaben vor dem Speichern

#### 6.4 Eingabeformate und Datentypen

Die Oberfläche muss die Eingaben des Anwenders prüfen, bevor sie an die fachlichen Services übergeben werden.

Feld	Datentyp	Validierung
name	String	darf nicht leer sein
anschrift	String	darf nicht leer sein
email	String	optional, einfache Formatprüfung
telefon	String	optional
bezeichnung	String	darf nicht leer sein
nettoPreis	BigDecimal	muss größer oder gleich 0 sein
mehrwertsteuerSatz	BigDecimal	muss größer oder gleich 0 sein
bestand	int	darf nicht negativ sein
menge	int	muss größer als 0 sein
status	String	nur erlaubte Statuswerte

## 6.5 Rückmeldungen an die Oberfläche

Die Services geben Rückmeldungen an die Benutzeroberfläche zurück. Diese Rückmeldungen werden in der Oberfläche als Erfolgs-, Warn- oder Fehlermeldung angezeigt.

Rückmeldung	Bedeutung	Anzeige in der Oberfläche
success	Aktion war erfolgreich	Erfolgsmeldung
validationError	Eingabe ist unvollständig oder ungültig	Fehlermeldung am Formular
warning	Aktion ist möglich, aber kritisch	Warnmeldung
locked	Dokument ist schreibgeschützt	Hinweis auf Finalisierung
exportFinished	PDF Export wurde abgeschlossen	Erfolgsmeldung

## 7. Systemarchitektur

### 7.1 Beschreibung der Architektur

Das UI Modul wird in mehrere Komponenten aufgeteilt. Das Hauptfenster verwaltet die Navigation. Die einzelnen Views stellen die Masken für Kunden, Produkte und Dokumente bereit. Controller oder ViewModels verbinden die Oberfläche mit den fachlichen Services.

Die Benutzeroberfläche führt selbst keine dauerhafte Speicherung durch. Sie nimmt Eingaben entgegen, zeigt Daten an, prüft Eingaben und übergibt gültige Benutzeraktionen an die zuständigen Services.

Die Architektur besteht aus folgenden Bereichen:

Bereich	Aufgabe
MainWindow	Startpunkt der Oberfläche
NavigationController	Steuerung der Navigation
CustomerView	Anzeige und Eingabe von Kundendaten
ProductView	Anzeige und Eingabe von Produktdaten
DocumentView	Anzeige und Erstellung von Dokumenten
ValidationService	Prüfung von Eingaben

Bereich	Aufgabe
MessageService	Anzeige von Rückmeldungen
Fachliche Services	Verarbeitung, Speicherung und Export

## 7.2 UML Klassendiagramm

Abbildung 1: UML Klassendiagramm des UI Moduls

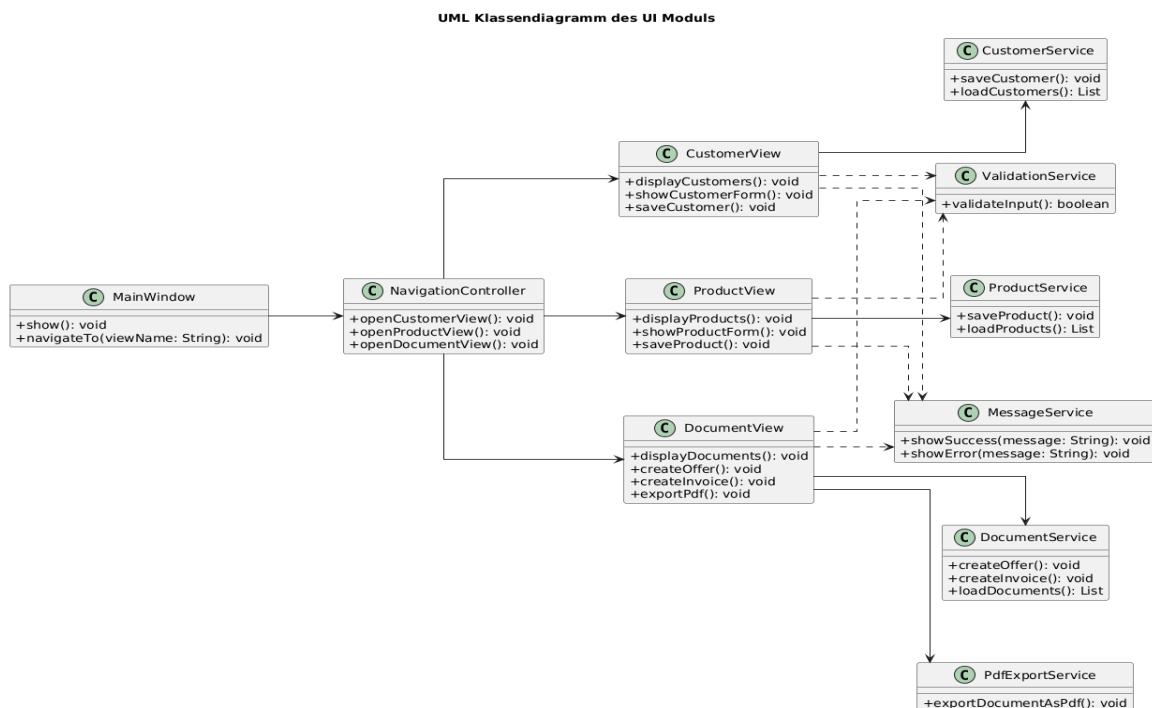


Abbildung 1 zeigt die wichtigsten Klassen des UI Moduls und deren Beziehungen zu den fachlichen Services.

## 7.3 Textuelle Beschreibung zum Klassendiagramm

Das Hauptfenster ist der Einstiegspunkt der Anwendung. Über den NavigationController kann der Anwender zwischen Kundenansicht, Produktansicht und Dokumentenansicht wechseln.

Die CustomerView ist für die Anzeige und Eingabe von Kundendaten zuständig. Die ProductView übernimmt die Anzeige und Eingabe von Produktdaten. Die DocumentView ist für Angebote, Auftragsbestätigungen, Lieferscheine, Rechnungen und den PDF Export zuständig.

Die FormModels speichern die Eingaben aus den Formularen vorübergehend. Der ValidationService prüft diese Eingaben vor dem Speichern. Das Ergebnis der Prüfung wird als ValidationResult zurückgegeben.

Der MessageService zeigt Erfolgsmeldungen, Warnungen und Fehlermeldungen an. Die fachlichen Services übernehmen die eigentliche Verarbeitung der Daten.

## 7.4 UML Sequenzdiagramm

Abbildung 2: UML Sequenzdiagramm zur Rechnungserstellung

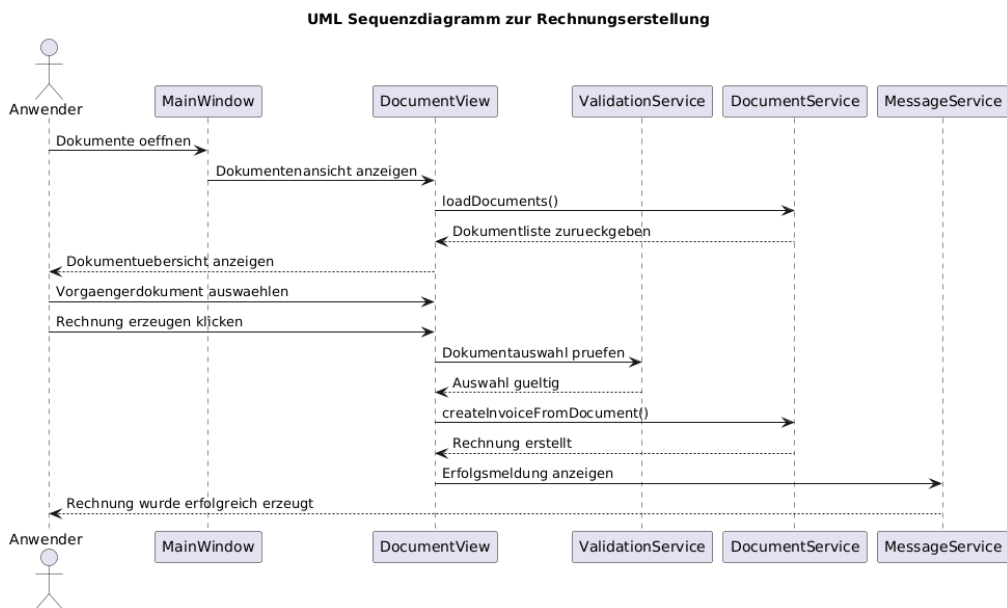


Abbildung 2 zeigt beispielhaft den Ablauf, wenn ein Anwender über die Benutzeroberfläche eine Rechnung aus einem vorhandenen Angebot erzeugt.

## 7.5 Textuelle Beschreibung zum Sequenzdiagramm

Der Anwender öffnet zuerst über das Hauptfenster die Dokumentenansicht. Die DocumentView fordert daraufhin über den DocumentService die vorhandenen Dokumente an. Anschließend werden die Dokumente in der Dokumentenübersicht angezeigt.

Danach wählt der Anwender ein vorhandenes Angebot aus und klickt auf die Aktion zur Rechnungserstellung. Die DocumentView prüft die Auswahl über den ValidationService. Wenn die Auswahl gültig ist, wird über den DocumentService eine Rechnung erzeugt.

Nach erfolgreicher Erstellung zeigt der MessageService eine Erfolgsmeldung in der Benutzeroberfläche an. Dadurch erhält der Anwender eine direkte Rückmeldung, dass die Rechnung erfolgreich erzeugt wurde.

## 8. Abnahmekriterien

Die Abnahmekriterien legen fest, wann das Modul Benutzeroberfläche als erfolgreich umgesetzt gilt. Sie beziehen sich auf die funktionalen und nichtfunktionalen Anforderungen dieses Pflichtenhefts.

ID	Bezug	Abnahmekriterium	Testmethode
AK-UI-01	PH-UI-01	Das Hauptfenster startet und zeigt eine Navigation zu Kunden, Produkten und Dokumenten an.	Manueller UI Test

ID	Bezug	Abnahmekriterium	Testmethode
AK-UI-02	PH-UI-02	Ein Kunde kann über die Kundenmaske eingegeben und gespeichert werden.	Funktionstest
AK-UI-03	PH-UI-03	Ein Produkt kann über die Produktmaske eingegeben und gespeichert werden.	Funktionstest
AK-UI-04	PH-UI-04	Vorhandene Dokumente werden in einer Tabelle angezeigt.	UI Test
AK-UI-05	PH-UI-05	Ein Angebot kann über die Dokumentenmaske mit Kunde und mindestens einer Position erstellt werden.	Workflow Test
AK-UI-06	PH-UI-06	Eine Rechnung kann aus einem vorhandenen Angebot über eine sichtbare Aktion erzeugt werden.	Workflow Test
AK-UI-07	PH-UI-07	Fehlende Pflichtfelder verhindern das Speichern und werden sichtbar markiert.	Negativtest
AK-UI-08	PH-UI-08	Erfolgs und Fehlermeldungen werden verständlich angezeigt.	UI Test
AK-UI-09	PH-UI-09	Finalisierte Rechnungen sind in der Oberfläche schreibgeschützt.	Funktionstest
AK-UI-10	PH-UI-10	Der PDF Export kann über einen sichtbaren Button gestartet werden.	Funktionstest
AK-UI-11	PH-Q-01	Ein neuer Anwender kann die Kernprozesse innerhalb von maximal 10 Minuten durchführen.	Usability Test
AK-UI-12	PH-Q-05	Die Oberfläche ist ohne Internetverbindung nutzbar.	Systemtest

## 9. Traceability Matrix

Die Traceability Matrix zeigt, welche Anforderungen aus dem Lastenheft durch welche Anforderungen im Pflichtenheft umgesetzt werden. Dadurch ist nachvollziehbar, dass die Benutzeroberfläche die geforderten Funktionen sichtbar unterstützt.

<b>Lastenheft ID</b>	<b>Inhalt aus dem Lastenheft</b>	<b>Pflichtenheft ID</b>
BA-01	Kundendatensätze anlegen	PH-UI-02, PH-UI-07, PH-UI-08
BA-02	Produktdaten speichern	PH-UI-03, PH-UI-07, PH-UI-08
BA-03	Angebot für vorhandenen Kunden mit Produktposition erstellen	PH-UI-05, PH-UI-07, PH-UI-08
BA-04	Rechnung aus vorhandenem Angebot erzeugen	PH-UI-06, PH-UI-09
Q-01	Benutzerfreundlichkeit und Bedienung innerhalb von 10 Minuten	PH-Q-01, PH-Q-02, PH-Q-03
Q-02	Datensicherheit und zuverlässige Datenverfügbarkeit	PH-UI-07, PH-Q-04, PH-Q-05
Q-03	Revisionsicherheit finaler Rechnungen	PH-UI-09
IF-01	Benutzerschnittstelle GUI	PH-UI-01 bis PH-UI-10
IF-03	Druck und Export Schnittstelle	PH-UI-10
GR-02	Unveränderlichkeit finaler Belege	PH-UI-09
GR-05	Dokumentenketten Konsistenz	PH-UI-06
GR-06	Summenberechnung	PH-UI-05

## **10. Modultestplan**

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, Speichern anklicken	Speicherung wird verhindert und Fehlermeldung erscheint

<b>Testfall ID</b>	<b>Anforderung</b>	<b>Testziel</b>	<b>Vorbedingung</b>	<b>Testschritte</b>	<b>Erwartetes Ergebnis</b>
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

### 10.1 Testdaten

Für die Modultests werden folgende Beispieldaten verwendet.

Objekt	Testdaten
Kunde	Max Mustermann, Musterstraße 1, 68161 Mannheim
Produkt	Beratung, Netto Einzelpreis 100,00 Euro, Mehrwertsteuer 19 Prozent, Bestand 10
Dokumentposition	Produkt Beratung, Menge 2
Rechnung	Finalisierte Rechnung mit Rechnungsnummer 1
Ungültiger Preis	minus 10,00 Euro

Objekt	Testdaten
Ungültiger Bestand	minus 5
Ungültige Menge	0

## 10.2 Bewertung der Testergebnisse

Ein Testfall gilt als bestanden, wenn das erwartete Ergebnis vollständig eintritt. Ein Testfall gilt als nicht bestanden, wenn eine Funktion nicht erreichbar ist, eine falsche Meldung angezeigt wird oder ungültige Eingaben gespeichert werden können.

Fehlgeschlagene Testfälle werden dokumentiert und nach der Fehlerbehebung erneut durchgeführt.

## 11. Glossar

Begriff	Erklärung
Benutzeroberfläche	Grafische Oberfläche, über die der Anwender mit dem System arbeitet
UI	User Interface, also Benutzeroberfläche
GUI	Graphical User Interface, grafische Benutzeroberfläche
Maske	Eingabebereich für bestimmte Daten, zum Beispiel Kundenmaske oder Produktmaske
Pflichtfeld	Feld, das ausgefüllt werden muss, damit eine Eingabe gespeichert werden kann
Validierung	Prüfung, ob eine Eingabe korrekt und vollständig ist
Fehlermeldung	Hinweis, dass eine Eingabe oder Aktion nicht gültig ist
Erfolgsmeldung	Hinweis, dass eine Aktion erfolgreich abgeschlossen wurde
Warnmeldung	Hinweis auf eine mögliche kritische Situation
Schreibschutz	Zustand, bei dem Daten nicht mehr verändert werden können
Finalisierte Rechnung	Rechnung, die abgeschlossen ist und nicht mehr verändert werden darf

<b>Begriff</b>	<b>Erklärung</b>
Dokumentenworkflow	Ablauf, bei dem aus einem Dokument ein Folgedokument entsteht
Angebot	Dokument, mit dem einem Kunden eine Leistung oder Ware angeboten wird
Auftragsbestätigung	Dokument, das einen angenommenen Auftrag bestätigt
Lieferschein	Dokument, das die Lieferung von Waren oder Leistungen dokumentiert
Rechnung	Dokument, mit dem eine Zahlung gefordert wird
Medienbruch	Unterbrechung eines Prozesses durch Wechsel in ein anderes System oder durch manuelle Übertragung
View	Sichtbarer Bereich der Benutzeroberfläche
Service	Fachliche Komponente, die Daten verarbeitet oder speichert
ViewModel	Objekt, das Daten zwischen Oberfläche und Logik vorbereitet
PDF Export	Erstellung einer PDF Datei aus einem Dokument
Traceability Matrix	Tabelle, die Anforderungen aus dem Lastenheft mit Anforderungen aus dem Pflichtenheft verbindet
Abnahmekriterium	Bedingung, anhand der geprüft wird, ob eine Anforderung erfüllt ist
Modultest	Test eines einzelnen Systembereichs, hier des UI Moduls
Usability Test	Test, bei dem geprüft wird, ob ein Anwender die Oberfläche einfach bedienen kann