

## Pflichtenheft

### „Entwicklung einer Litera Bücherkatalog-Anwendung“

Stand: 23.12.2024

Auftraggeber:



Firma Litera

Schloss Straße 10,

01067 Dresden, Deutschland

Ansprechpartner: Edwin Hartmann

Kontaktdaten:

Telefon: +49 40 1864567

E-Mail: [info@litera.de](mailto:info@litera.de)

Auftragnehmer:



Firma NOVATECH SOLUTIONS

Technologiepark 15,

22303 Hamburg, Deutschland

Ansprechpartner: Anna Fischer

Kontaktdaten:

Telefon: +49 40 1234567

E-Mail: [info@novatech-solutions.de](mailto:info@novatech-solutions.de)

Autoren: Ranim Khallouf, Rama Alshaer, Marija Ratnikova, Lena Heberlein

# Inhaltsverzeichnis

<b>1. Zielbestimmung .....</b>	<b>4</b>
1.1 Muss-Kriterien .....	4
1.2 Kann-Kriterien .....	4
1.3 Abgrenzungskriterien .....	5
<b>2. Produkteinsatz .....</b>	<b>5</b>
2.1 Anwendungsbereich.....	5
2.2 Zielgruppen .....	5
2.3 Produktumgebung .....	5
2.3.1 Architektur .....	6
2.3.2 Technologie.....	6
2.3.3 Komponenten.....	6
2.3.4 Schnittstellen .....	6
2.4 Betriebsbedingungen .....	7
<b>3. Produktfunktionen/Anforderungen .....</b>	<b>7</b>
<b>3.1 Funktionale Anforderungen .....</b>	<b>7</b>
3.1.1 Beschreibung der FA mit Rollen innerhalb der Geschäftsprozesse.....	7
3.1.2 Aktivitäten mit Benutzerschnittstelle (UI) .....	8
3.1.3 Fachliches Klassendiagramm (domain model) / Produktdaten .....	11
<b>3.2 Nichtfunktionale Anforderungen .....</b>	<b>12</b>
3.2.1 Benutzbarkeit.....	12
3.2.2 Zuverlässigkeit.....	12
3.2.3 Effizienz .....	12
3.2.4 Softwarewartung.....	12
3.2.5 Sicherheit .....	12
3.2.6 Normen .....	12
<b>4. Testung .....</b>	<b>13</b>
4.1 Funktionstests.....	13
4.2 Integrationstests.....	14
4.3 Usability-Tests.....	14
4.4 Testumgebung.....	14
4.5 Testergebnisse .....	14
4.6 Modultests .....	14
<b>5. Monitoring/ Support bei Übergabe oder ähnliche Leistungen.....</b>	<b>15</b>
5.1 Bereitstellung des Repositories .....	15
5.2 Individueller Support .....	15
5.3 Erreichbarkeit und Kommunikationskanäle.....	16
5.4 Unterstützung bei der Übergabe.....	16
5.5 Schulung .....	16
5.6 Monitoring nach der Übergabe .....	16

<b>6. Dokumentation.....</b>	<b>17</b>
6.1 Anwenderdokumentation .....	17
6.2 Administratordokumentation .....	17
Da die Anwendung nur grundlegende administrative Eingriffe erfordert, umfasst die Administratordokumentation lediglich die wichtigsten Informationen zur Einrichtung, Konfiguration, Wartung und Fehlerbehebung.....	17
6.3 Entwicklerdokumentation .....	17
6.4 Weitere referenzierte Dokumente .....	17
<b>7. Vorgehen .....</b>	<b>17</b>
<b>8. Entwicklungsumgebung .....</b>	<b>18</b>
8.1 Tools.....	18
8.2 Hardware .....	18
8.3 Software .....	18
8.4 Orgware .....	18
<b>9. Glossar .....</b>	<b>18</b>

## 1. Zielbestimmung

Für den Buchladen Litera besteht die Notwendigkeit, einen digitalen Bücherkatalog zu erstellen, der es den Kunden ermöglicht, das angebotene Sortiment des Ladens bequem zu durchsuchen. Der Katalog sollte mehrere Kernfunktionen enthalten: die Suche nach bestimmten Filtern, die Anzeige detaillierter Informationen über Bücher sowie das Verfassen und Veröffentlichen von Rezensionen. In Zukunft ist eine Erweiterung der Plattform geplant, um einen Online-Shop für den Verkauf von Büchern zu integrieren.

### 1.1 Muss-Kriterien

ID	Name	Beschreibung
MK-SYS-0	Systemumgebung	Die Anwendung muss in Java entwickelt werden und auf Linux-Systemen (Ubuntu 22.04 LTS) sowie auf Windows 10 und höher ausführbar sein.
MK-FUN-01	Buchsuche	Nutzer müssen Bücher mindestens nach einem der Kriterien (Titel, Autor, Verlag, Genre) suchen können.
MK-FUN-02	Buchdetails anzeigen	Die Anwendung muss Details eines ausgewählten Buches anzeigen, einschließlich Titel, Autor, Genre, Verlag und Beschreibung.
MK-FUN-03	Rezensionen verwalten	Nutzer müssen Rezensionen für Bücher schreiben, anzeigen und speichern können.
MK-DAT-01	Datenspeicherung	Alle Daten (Bücher und Rezensionen) müssen in JSON- oder Textdateien gespeichert werden.
MK-SYS-02	Grafische Benutzeroberfläche	Eine einfache grafische Benutzeroberfläche (GUI).
MK-PERF-01	Reaktionszeit	Das System muss innerhalb von maximal 2 Sekunden auf Benutzeranfragen reagieren.
MK-DOK-01	Dokumentation	Eine grundlegende Benutzerdokumentation muss bereitgestellt werden, um die Nutzung des Systems zu erleichtern.
MK-SICH-01	Datensicherheit	Gespeicherte Daten müssen vor unberechtigtem Zugriff geschützt werden.
MK-WART-01	Wartbarkeit	Der Quellcode muss gut dokumentiert sein, um zukünftige Änderungen und Erweiterungen zu erleichtern

### 1.2 Kann-Kriterien

ID	Name	Beschreibung
KK-FUN-01	Erweiterte Filteroptionen	Nutzer können Bücher zusätzlich nach Erscheinungsjahr, Tags oder Bewertungen filtern.
KK-FUN-02	Mehrsprachige Unterstützung	Die Benutzeroberfläche kann auf Deutsch und Englisch, umgestellt werden.

KK-FUN-03	Bewertungen anzeigen	Nutzer können die durchschnittliche Bewertung eines Buches einsehen und auch Bewertungen hinzufügen.
KK-FUN-04	Anzeige ähnlicher Bücher	Das System kann ähnliche Bücher basierend auf Genre oder Autor empfehlen.
KK-SYS-01	Erweiterung auf andere Plattformen	Die Anwendung kann auch auf macOS(Ventura 13.0 oder höher) ausführbar sein.

### 1.3 Abgrenzungskriterien

ID	Name	Beschreibung
AK-FUN-01	Keine Online-Integration	Es wird keine Anbindung an externe Online-Services oder APIs (z. B. Google Books) realisiert.
AK-DAT-01	Keine relationale Datenbanksysteme	Die Daten werden nur in JSON- oder Textdateien gespeichert, keine Verwendung von relationalen Datenbanken (z. B. MySQL, PostgreSQL).
AK-FUN-02	Kein Kaufprozess	Die Funktion zum Kauf von Büchern wird nicht implementiert, sondern bleibt als mögliche zukünftige Erweiterung vorgesehen.
AK-FUN-04	Keine Benutzerkonten	Es wird keine Registrierung oder Verwaltung von Benutzerkonten implementiert.

## 2. Produkteinsatz

### 2.1 Anwendungsbereich

Die Firma Litera nutzt die Bücherkatalog-Anwendung, um ihren Buchkatalog effizient und übersichtlich darzustellen. Das System ermöglicht eine einfache Suche nach Büchern und relevanten Informationen.

### 2.2 Zielgruppen

#### **Bücherliebhaber:**

Möglichkeit, Lieblingsbücher einfach zu suchen, detaillierte Informationen zu erhalten und durch Rezensionen andere Leser zu inspirieren.

#### **Studierende:**

Unterstützung bei der Suche nach geeigneter Studienliteratur sowie beim Verfassen und Lesen akademischer Rezensionen.

#### **Lehrkräfte:**

Eine Plattform zur Verwaltung und Weitergabe von Buchinformationen und -anmerkungen an Schüler oder Kollegen.

#### **Bibliotheken und Bildungseinrichtungen:**

Ermöglicht eine Erweiterung der Buchkataloge von Bibliotheken und Geschäften, wodurch das Sortiment vielfältiger und umfassender gestaltet wird.

### 2.3 Produktumgebung

Das System wird in einer Umgebung mit den folgenden Voraussetzungen betrieben:

### 2.3.1 Architektur

- **Modulare Architektur:** Die Anwendung besteht aus unabhängigen Modulen wie Buchsuche, Anzeige von Buchdetails und Verwaltung von Rezensionen. Diese Module kommunizieren über standardisierte Schnittstellen.
- **Dateibasierte Speicherung:** Daten wie Buchinformationen und Rezensionen werden in JSON- oder Textdateien gespeichert, die lokal verarbeitet werden.

### 2.3.2 Technologie

- **Programmiersprache:** Java (mindestens Version 1.8 oder höher).
- **Ausführung:** Die Anwendung wird als Desktop-Anwendung mit einer grafischen Benutzeroberfläche (GUI) ausgeführt. Sie ermöglicht die Interaktion über ein benutzerfreundliches Interface mit klaren Menüs und Bedienelementen.
- **Softwareanforderungen:** Java Runtime Environment (JRE) muss auf dem Zielsystem installiert sein. Der Pfad zur Java-Umgebung muss korrekt gesetzt sein.
- **Hardwareanforderungen:** Keine spezifischen Anforderungen; die Anwendung läuft auf handelsüblichen Rechnern.

### 2.3.3 Komponenten

- **Kernkomponenten:**
  - **Suchmodul:** Ermöglicht die Suche nach Büchern anhand von Titel, Autor oder Genre.
  - **Anzeigemodul:** Zeigt Buchdetails wie Titel, Autor, Genre und Beschreibung an.
  - **Rezensionsmodul:** Verwalten von Benutzerrezensionen, einschließlich Schreiben, Anzeigen und Speichern.
- **Datenspeicherung:** JSON-basierte Speicherung von Büchern und Rezensionen.

### 2.3.4 Schnittstellen

- **Interne Schnittstellen:**
  - Module kommunizieren über Methodenaufrufe innerhalb der Anwendung.
- **Externe Schnittstellen:**
  - Es werden keine externen APIs oder Datenbanken integriert.

## 2.4 Betriebsbedingungen

Das System wird von der Firma Litera zur Verwaltung ihres Buchkatalogs verwendet. Hauptnutzer sind Mitarbeiter und Kunden, die auf die Such- und Rezensionsfunktionen zugreifen. Das System läuft auf **Windows- und Linux-Rechnern**. Eine Installation auf anderen Betriebssystemen ist nicht vorgesehen. Keine speziellen Hardwareanforderungen. Das System funktioniert auf den üblichen Arbeitsplätzen von Litera ohne dedizierte unterbrechungsfreie Stromversorgung (USV).

Die Anwendung wird in klimatisierten Büros auf gepflegter, zyklisch aktualisierter Hardware betrieben. Die Installation und Updates erfolgen durch die IT-Abteilung während Wartungszeiten.

## 3. Produktfunktionen/Anforderungen

### 3.1 Funktionale Anforderungen

ID	Name	Beschreibung
F1	Buchsuche	Nutzer können Bücher nach mindestens einem Kriterium (Titel, Autor, Genre, Verlag) suchen.
F2	Buchdetails anzeigen	Details wie Titel, Autor, Genre, Beschreibung, Verlag sollen angezeigt werden.
F3	Rezensionen verwalten	Nutzer können Rezensionen schreiben, anzeigen und speichern.

#### 3.1.1 Beschreibung der FA mit Rollen innerhalb der Geschäftsprozesse

##### Mitarbeiter von Litera

Mitarbeiterinnen und Mitarbeiter des Unternehmens Litera verwenden die Anwendung zur täglichen Pflege des Buchkatalogs. Dazu gehören unter anderem das Hinzufügen neuer Bücher, das Bearbeiten bestehender Einträge sowie das Moderieren von eingereichten Rezensionen durch Kunden. Sie stellen sicher, dass die dargestellten Inhalte aktuell, korrekt und qualitativ hochwertig sind. Darüber hinaus tragen sie zur strukturellen Ordnung der Daten bei und unterstützen bei der Kategorisierung nach Genre, Autor und Verlag.

##### Kunden

Kundinnen und Kunden nutzen die Anwendung primär zur Recherche innerhalb des Buchkatalogs. Mithilfe der intuitiven Suchfunktionen und Filteroptionen können sie gezielt nach Titeln, Autoren oder Genres suchen. Sie haben Zugriff auf ausführliche Buchinformationen sowie vorhandene Rezensionen anderer Nutzerinnen und Nutzer. Zudem können sie eigene Rezensionen verfassen und damit aktiv zur Bewertung und Weiterentwicklung des Katalogs beitragen.

##### Administrator

Die Administratorin oder der Administrator ist für den technischen Betrieb der Anwendung verantwortlich. Zu den Hauptaufgaben zählen die regelmäßige Wartung, das Einspielen von Updates sowie die Sicherung und Wiederherstellung von Daten. Darüber hinaus kümmern sie sich um die Konfiguration der Systempfade, die Anpassung der Anwendung an firmenspezifische Anforderungen und die

Überwachung der Systemleistung. Sie fungieren zudem als erste Ansprechperson bei technischen Problemen.

AF-01: Buchsuche	Ein Benutzer startet die Anwendung über die grafische Benutzeroberfläche und gibt Suchkriterien wie Titel, Autor, Verlag oder Genre in ein Suchfeld ein. Das System durchsucht den Katalog und zeigt eine Liste der passenden Bücher in einem Ergebnisfenster an. Der Benutzer kann die Liste scrollen und bei Bedarf die Suchkriterien ändern.
AF-02: Anzeige von Buchdetails	Nach einer erfolgreichen Buchsuche wählt der Benutzer ein bestimmtes Buch aus der Ergebnisliste durch einen Klick aus. Die Anwendung öffnet ein neues Fenster oder einen Dialog, in dem detaillierte Informationen zum Buch wie Titel, Autor, Verlag, Genre und Beschreibung angezeigt werden.
AF-03: Rezensionen lesen	Benutzer können Rezensionen zu einem Buch einsehen, die Anwendung zeigt eine Liste der vorhandenen Rezensionen in einem separaten Bereich an.
AF-04: Rezension schreiben	Benutzer können Rezensionen schreiben, indem sie in der Detailansicht eines Buches auf die Schaltfläche „Rezension schreiben“ klicken. Ein Eingabefeld erscheint, in das der Benutzer seine Rezension eingeben kann. Nach dem Klick auf „Speichern“ wird die Rezension in der JSON-Datei oder Datenbank gespeichert und zur späteren Anzeige hinzugefügt.

### 3.1.2 Aktivitäten mit Benutzerschnittstelle (UI)

Anwendungsfall ID	AF-01
AF Name	Buchsuche
Akteur	Benutzer der Anwendung
Vorbedingung	Die Anwendung ist gestartet, und der Buchkatalog ist verfügbar (z. B. JSON-Datei ist geladen).
Auslösendes Ereignis B	Der Benutzer öffnet das Suchmenü in der grafischen Benutzeroberfläche und wählt eine Suchkategorie aus (z. B. Titel, Autor, Verlag, Genre). Anschließend gibt er den Suchbegriff in das entsprechende Eingabefeld ein und bestätigt die Suche.
Nachbedingung Erfolg	Eine Liste passender Bücher wird im Ergebnisfenster der Anwendung angezeigt.
Nachbedingung Fehlschlag	Wenn keine Bücher gefunden wurden, wird eine entsprechende Meldung ausgegeben (z. B. „Nothing found.“).
Ablauf	<ol style="list-style-type: none"> <li>1. Benutzer startet die Anwendung.</li> <li>2. Der Benutzer wählt eine Suchkategorie (Titel, Autor, Verlag oder Genre) aus.</li> </ol>



	<ol style="list-style-type: none"> <li>Der Benutzer gibt den Suchbegriff in das Eingabefeld ein und startet die Suche.</li> <li>Die Anwendung zeigt eine Liste der passenden Bücher an.</li> </ol>
Benutzerschnittstelle	<ul style="list-style-type: none"> <li>Eingabe: Auswahl der Suchkategorie und Eingabe des Suchbegriffs über die grafische Benutzeroberfläche.</li> <li>Ausgabe: Anzeige der Ergebnisse in einem Listenformat im Ergebnisfenster.</li> </ul>

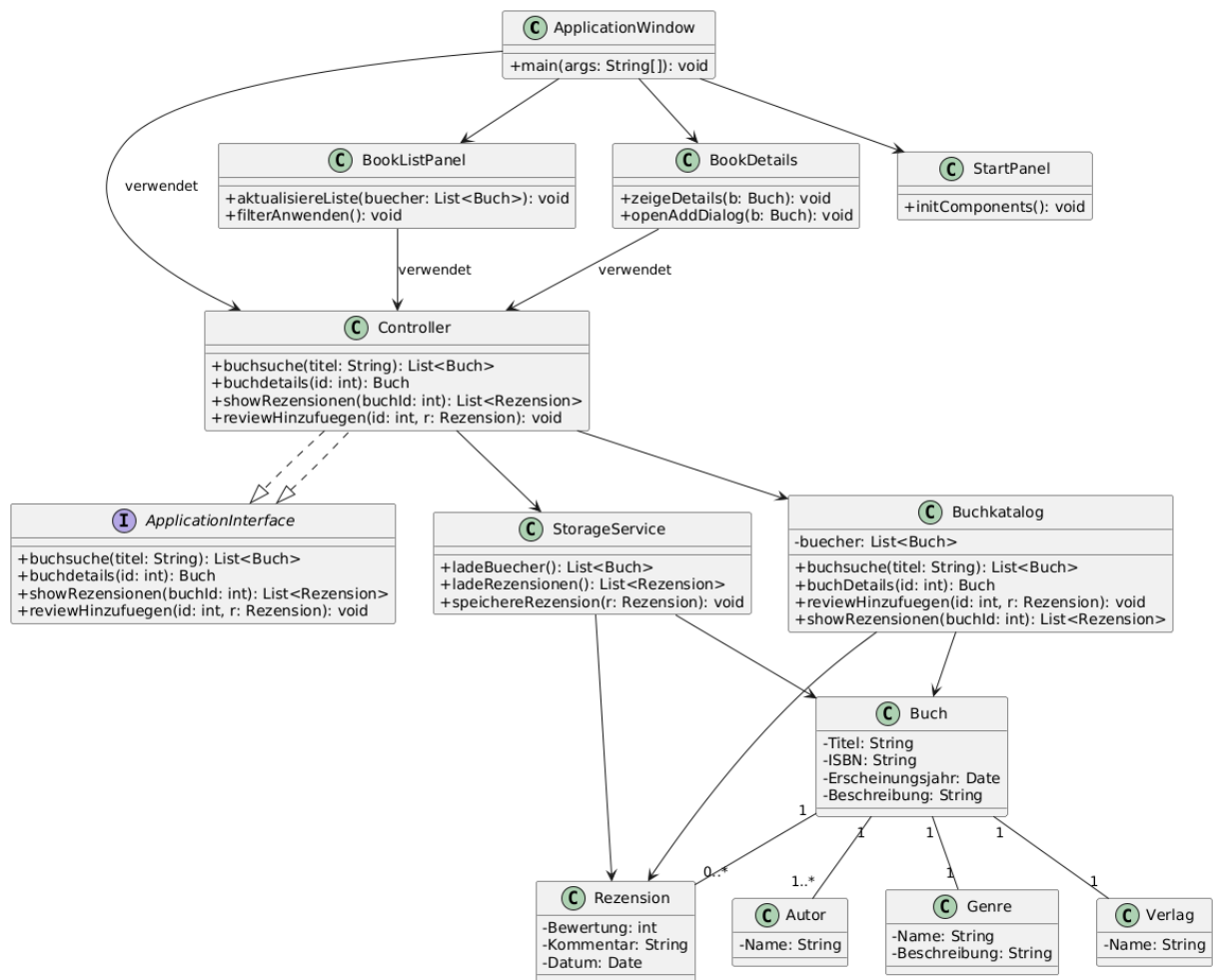
Anwendungsfall ID	AF-02
AF Name	Anzeige von Buchdetails
Akteur	Benutzer der Anwendung
Vorbedingung	Die Anwendung ist gestartet, und eine Suchanfrage wurde erfolgreich ausgeführt. Eine Liste der Suchergebnisse ist verfügbar.
Auslösendes Ereignis B	Der Benutzer wählt in der Ergebnisliste ein Buch durch Anklicken aus.
Nachbedingung Erfolg	Die Anwendung zeigt die Details des ausgewählten Buches (Titel, Autor, Genre und Beschreibung) in einem separaten Fenster an.
Nachbedingung Fehlschlag	Falls das Buch nicht verfügbar ist, wird eine Fehlermeldung angezeigt.
Ablauf	<ol style="list-style-type: none"> <li>Der Benutzer führt eine Buchsuche durch.</li> <li>Die Anwendung zeigt eine Liste der Suchergebnisse an.</li> <li>Der Benutzer wählt ein Buch aus der Ergebnisliste aus.</li> <li>Die Anwendung öffnet ein Detailfenster mit den Buchinformationen.</li> </ol>
Benutzerschnittstelle	<ul style="list-style-type: none"> <li>Eingabe: Auswahl eines Buches aus der Ergebnisliste durch Klicken.</li> <li>Ausgabe: Anzeige der Buchdetails in einem neuen Fenster.</li> </ul>

Anwendungsfall ID	AF-03
AF Name	Rezensionen lesen
Akteur	Benutzer der Anwendung

Vorbedingung	Ein Buch wurde ausgewählt, und Rezensionen sind verfügbar.
Auslösendes Ereignis B	Benutzer können Rezensionen zu einem Buch einsehen.
Nachbedingung Erfolg	Die Rezensionen werden in einem separaten Bereich unter den Buchdetails angezeigt.
Nachbedingung Fehlschlag	Falls keine Rezensionen vorhanden sind, wird die Nachricht „No reviews available“ angezeigt.
Ablauf	<ol style="list-style-type: none"> <li>1. Der Benutzer öffnet die Detailansicht eines Buches.</li> <li>2. Die Anwendung zeigt die verfügbaren Rezensionen an.</li> </ol>
Benutzerschnittstelle	<ul style="list-style-type: none"> <li>• Auswahl eines Buches aus der Ergebnisliste durch Klicken.</li> <li>• Ausgabe: Liste der Rezensionen wird in der Detailansicht angezeigt.</li> </ul>

Anwendungsfall ID	AF-04
AF Name	Rezension schreiben
Akteur	Benutzer der Anwendung
Vorbedingung	Ein Buch wurde ausgewählt.
Auslösendes Ereignis B	Der Benutzer klickt im Buchdetailfenster auf die Schaltfläche „Write review“.
Nachbedingung Erfolg	Die Rezension wird erfolgreich gespeichert und der Benutzer erhält eine Bestätigungsmeldung.
Nachbedingung Fehlschlag	Falls die Rezension nicht gespeichert werden kann, wird eine Fehlermeldung angezeigt.
Ablauf	<ol style="list-style-type: none"> <li>1. Benutzer wählt ein Buch aus.</li> <li>2. Der Benutzer klickt auf Option „Write a review“.</li> <li>3. Der Benutzer gibt die Rezension ein und klickt auf „Save“.</li> <li>4. Die Anwendung speichert die Rezension und zeigt eine Bestätigungsmeldung an.</li> </ol>
Benutzerschnittstelle	<ul style="list-style-type: none"> <li>• Eingabe: Eingabe der Rezension im Eingabefeld.</li> <li>• Ausgabe: Bestätigungsmeldung, dass die Rezension gespeichert wurde.</li> </ul>

### 3.1.3 Fachliches Klassendiagramm (domain model) / Produktdaten



## 3.2 Nichtfunktionale Anforderungen

### 3.2.1 Benutzbarkeit

Die Anwendung soll über eine intuitive grafische Benutzeroberfläche bedienbar sein. Die Menüs, Suchfelder und Schaltflächen sind klar strukturiert und leicht verständlich gestaltet, um die Nutzung so einfach wie möglich zu machen.

### 3.2.2 Zuverlässigkeit

Die Anwendung muss sicherstellen, dass alle Suchanfragen korrekt verarbeitet werden. Bei ungültigen Eingaben wird eine Fehlermeldung ausgegeben, ohne dass die Anwendung abstürzt. Die Datenintegrität bleibt gewährleistet (keine Datenverluste).

### 3.2.3 Effizienz

Suchergebnisse müssen innerhalb von maximal 2 Sekunden angezeigt werden. Die Verarbeitung von Rezensionen und Buchdaten erfolgt ohne merkbare Verzögerungen.

### 3.2.4 Softwarewartung

Der Quellcode wird gut dokumentiert, um zukünftige Änderungen und Erweiterungen zu erleichtern. Neue Funktionen wie Datenbankintegration, können problemlos hinzugefügt werden. Langfristig ist eine Unterstützung für mehrere Sprachen vorgesehen.

### 3.2.5 Sicherheit

Die gespeicherten Daten (Bücher und Rezensionen) dürfen nur von autorisierten Benutzern geändert werden. Es gibt keine spezielle Verschlüsselung, da die Anwendung lokal genutzt wird.

### 3.2.6 Normen

NF-N1 Codekonventionen	Die Anwendung folgt den <a href="#">Java Coding Conventions</a> sowie ergänzend dem <a href="#">Mozilla Java Style Guide</a> , um eine einheitliche Struktur und Lesbarkeit des Quellcodes sicherzustellen.
NF-N2 Dokumentation	Für die technische Dokumentation wird das Tool <i>Javadoc</i> verwendet. Die Dokumentation orientiert sich an den Empfehlungen der <a href="#">Javadoc-Spezifikation von Oracle</a> .
NF-N3 Build-Management	Als Build-Tool wird <a href="#">Apache Maven</a> eingesetzt, wodurch der Projektaufbau, das Dependency-Management sowie die Integration von Plugins (z. B. für Tests und Obfuskation) standardisiert und nachvollziehbar ist.
NF-N4 Dateiformat-Standard	Die Daten werden im JSON-Format gespeichert, das dem ECMA-404 Standard entspricht und damit leicht lesbar und austauschbar ist.

## 4. Testung

Für die Bücherkatalog-Anwendung wird ein systematisches Testverfahren angewendet, um sicherzustellen, dass alle definierten funktionalen und nichtfunktionalen Anforderungen erfüllt werden. Die Tests umfassen Funktionstests, Integrationstests und Usability-Tests.

### 4.1 Funktionstests

Die Funktionstests überprüfen die definierten Anwendungsfälle (AF) und stellen sicher, dass die Hauptfunktionen der Anwendung mit der grafischen Benutzeroberfläche korrekt arbeiten.

#### AF-01: Buchsuche

**Testziel:** Sicherstellen, dass Suchanfragen nach Titel, Autor oder Genre korrekte Ergebnisse liefern.

**Testfall:**

**Eingabe:** Der Benutzer öffnet die Anwendung, gibt im Suchfeld „Harry Potter“ ein.

**Erwartetes Ergebnis:** Eine Liste von Büchern mit dem Titel „Harry Potter“ wird im Ergebnisbereich angezeigt.

**Teststatus:** Erfolgreich, wenn die angezeigten Ergebnisse mit den gespeicherten Daten übereinstimmen.

#### AF-02: Anzeige von Buchdetails

**Testziel:** Sicherstellen, dass die Buchdetails korrekt angezeigt werden.

**Testfall:**

**Eingabe:** Der Benutzer klickt auf ein Buch in der Ergebnisliste.

**Erwartetes Ergebnis:** Ein neues Fenster mit den Buchdetails (Titel, Autor, Genre, Verlag, Beschreibung) wird angezeigt.

**Teststatus:** Erfolgreich, wenn die angezeigten Details den gespeicherten Daten entsprechen.

#### AF-03: Rezensionen lesen

**Testziel:** Sicherstellen, dass die gespeicherten Rezensionen korrekt angezeigt werden.

**Testfall:**

**Eingabe:** Der Benutzer klickt auf ein Buch in der Ergebnisliste.

**Erwartetes Ergebnis:** Eine Liste der Rezensionen wird unter den Buchdetails angezeigt.

**Teststatus:** Erfolgreich, wenn alle gespeicherten Rezensionen korrekt dargestellt werden.

#### AF-04: Rezension schreiben

**Testziel:** Sicherstellen, dass eine neue Rezension gespeichert wird.

**Testfall:**

**Eingabe:** Der Benutzer klickt im Detailfenster eines Buches auf „Write review“, gibt die Rezension ein und klickt auf „Save“.

**Erwartetes Ergebnis:** Die Rezension wird gespeichert und ist über die Funktion „Show reviews“ sichtbar.

**Teststatus:** Erfolgreich, wenn die neue Rezension korrekt angezeigt wird.

## 4.2 Integrationstests

Integrationstests überprüfen die Interaktion zwischen den verschiedenen Modulen der Anwendung.

**Testziel:** Sicherstellen, dass Module wie Buch, Genre, Autor und Rezension korrekt miteinander interagieren.

**Testfall:**

**Eingabe:** Der Benutzer sucht nach „Fantasy“, wählt ein Buch aus den Ergebnissen, zeigt die Details an und schreibt eine Rezension.

**Erwartetes Ergebnis:** Alle Schritte funktionieren ohne Fehler, und die Rezension wird korrekt gespeichert und angezeigt.

## 4.3 Usability-Tests

Die Usability-Tests überprüfen die Benutzerfreundlichkeit der grafischen Benutzeroberfläche

**Testziel:** Sicherstellen, dass die Benutzeroberfläche klar und intuitiv bedienbar ist.

**Testfall:**

**Szenario:** Ein neuer Benutzer führt eine Buchsuche durch, zeigt die Details eines Buches an und schreibt eine Rezension.

**Bewertungskriterien:** Verständlichkeit der Menüs und Schaltflächen, Klarheit der Ausgabe.

**Ergebnis:** Erfolgreich, wenn Benutzer die Aufgaben ohne externe Hilfe ausführen können und die Benutzeroberfläche als benutzerfreundlich bewertet wird.

## 4.4 Testumgebung

Die Tests werden in folgender Umgebung durchgeführt:

- **Betriebssysteme:** Windows 10, Linux Ubuntu 22.04
- **Java-Version:** Java 11 oder höher
- **Hardware:** Standard-PC mit mindestens 4 GB RAM und 1 GHz CPU

## 4.5 Testergebnisse

Nach Abschluss aller Tests werden die Ergebnisse dokumentiert, und fehlerhafte Funktionen werden korrigiert. Die Anwendung wird als erfolgreich getestet angesehen, wenn:

1. Alle definierten funktionalen Tests (AF-01 bis AF-04) bestanden sind.
2. Integrationstests ohne Fehler abgeschlossen wurden.
3. Benutzerfreundlichkeit mit einer positiven Bewertung abgeschlossen wurde.

## 4.6 Modultests

Die Modultests dienen dazu, einzelne Komponenten der Anwendung isoliert zu testen, bevor sie in das Gesamtsystem integriert werden. Sie gewährleisten, dass jede Klasse oder Methode gemäß den Anforderungen funktioniert.

**Getestete Module:**

**StorageService**

Ziel: Sicherstellen, dass Bücher und Rezensionen korrekt gelesen und geschrieben werden.

Testfälle:

- Lesevorgang mit existierender Datei.

- Lesevorgang mit fehlerhafter Datei.
- Schreiben einer neuen Rezension.  
Ergebnis: Erfolgreich, wenn beim Lesen/Schreiben keine Fehler auftreten und die Inhalte korrekt verarbeitet werden.

### **Buchkatalog**

Ziel: Überprüfung der Such- und Filterfunktionen.

#### **Testfälle:**

- Suche nach Titel.
- Suche nach Autor oder Genre.
- Hinzufügen und Abrufen von Rezensionen.

Ergebnis: Erfolgreich, wenn passende Bücher zurückgegeben und Rezensionen korrekt verwaltet werden.

### **Controller**

Ziel: Prüfung der Schnittstelle ApplicationInterface.

#### **Testfälle:**

- Übergabe von Suchbegriffen an Buchkatalog.
- Aufruf der Detailansicht und Rezensionsverwaltung.

Ergebnis: Erfolgreich, wenn Methodenaufrufe korrekt delegiert werden.

### **GUI-Komponenten (z. B. BookDetails, BookListPanel)**

Ziel: Testen der Ereignisverarbeitung und Anzeige.

#### **Testfälle:**

- Klick auf ein Buch öffnet Detailansicht.
- Schreiben einer Rezension löst Speichervorgang aus.

Ergebnis: Erfolgreich, wenn Aktionen korrekt ausgeführt und Informationen richtig angezeigt werden.

## **5. Monitoring/ Support bei Übergabe oder ähnliche Leistungen**

Im Rahmen des semesterbegleitenden Supports für die "Bücherkatalog-Anwendung" werden folgende Leistungen erbracht:

### **5.1 Bereitstellung des Repositories**

- Ein Git-Repository wird bereitgestellt, das alle relevanten Projektartefakte enthält, darunter:
  - Den vollständigen Quellcode der Anwendung.
  - Dokumentationen wie Anwender-, Entwickler- und Administratorendokumentationen.
  - Testprotokolle und -ergebnisse.
  - Historische Versionen der Anwendung zur Nachverfolgung von Änderungen.

### **5.2 Individueller Support**

- Während des Einsatzes der Anwendung steht ein individueller Support zur Verfügung, der folgende Leistungen umfasst:
- Regelmäßige Feedbackgespräche zur Klärung technischer Fragen und zur gemeinsamen Lösung möglicher Herausforderungen.

- Detaillierte Anleitungen zur Nutzung der Anwendung sowie zur Behebung technischer Probleme.

### **5.3 Erreichbarkeit und Kommunikationskanäle**

- Rufbereitschaft an fünf Tagen pro Woche während der Arbeitszeiten von 9.00 bis 17.00 (8 Stunden täglich).
- Kontaktaufnahme über folgende Kanäle:
  - E-Mail für technische Anfragen und Probleme.
  - Telefon für dringende Supportanfragen.
  - Eine FAQ-Seite im Repository zur Selbsthilfe bei häufigen Fragen.

### **5.4 Unterstützung bei der Übergabe**

- Um eine reibungslose Übergabe der Anwendung zu gewährleisten, wird Folgendes angeboten:
  - Unterstützung bei der Installation der Anwendung auf den Zielplattformen (Linux und Windows).
  - Gemeinsame Überprüfung der Systemanforderungen vor der Installation.
  - Erstellung eines Übergabeprotokolls, das alle Schritte dokumentiert.

### **5.5 Schulung**

- Eine Schulung wird für die Endnutzer bereitgestellt, die folgende Inhalte abdeckt:
  - Einführung in die Bedienung der grafische Benutzeroberfläche.
  - Anleitung zur Buchsuche, Anzeige von Details und Verwaltung von Rezensionen.
  - Hinweise zur Pflege und Sicherung der gespeicherten Daten.

### **5.6 Monitoring nach der Übergabe**

- Nach der Inbetriebnahme der Anwendung wird ein Monitoring durchgeführt, um:
  - Die Leistung der Anwendung zu überwachen und sicherzustellen, dass sie den Anforderungen entspricht.
  - Frühzeitig potenzielle Probleme zu identifizieren und zu beheben.
- Ein Monitoring-Bericht wird erstellt und dem Auftraggeber zur Verfügung gestellt.



## 6. Dokumentation

### 6.1 Anwenderdokumentation

Die Anwenderdokumentation wird als "liesmich.txt"-Datei in deutscher und "readme.txt"-Datei in englischer Sprache im Repository bereitgestellt. Sie enthält:

- Anweisungen zur Installation der Anwendung.
- Beispiele für die Nutzung der Benutzeroberfläche.
- Hinweise zur Fehlerbehebung bei typischen Problemen.

### 6.2 Administratordokumentation

Da die Anwendung nur grundlegende administrative Eingriffe erfordert, umfasst die Administratordokumentation lediglich die wichtigsten Informationen zur Einrichtung, Konfiguration, Wartung und Fehlerbehebung.

### 6.3 Entwicklerdokumentation

Die Entwicklerdokumentation wird mit Javadoc erstellt und als HTML-Dateien im Repository abgelegt. Sie enthält:

- Beschreibung der Codebasis und der einzelnen Module.
- Details zur JSON-Datenstruktur, einschließlich Beispieldateien und deren Schema.
- Erläuterung der verwendeten Schnittstellen und Klassenhierarchien.

### 6.4 Weitere referenzierte Dokumente

Alle relevanten Dokumente wie Lastenheft, Pflichtenheft, Anwender- und Entwicklerdokumentation sowie andere Projektreferenzen werden zentral im Repository gespeichert. Diese Dokumente werden regelmäßig aktualisiert, um den aktuellen Entwicklungsstand widerzuspiegeln.

## 7. Vorgehen

Das Projekt folgt einem iterativen Entwicklungsmodell:

- Erstellung eines funktionalen Prototyps für die grundlegenden Funktionen der Anwendung (z. B. Buchsuche, Anzeige von Buchdetails, Verwaltung von Rezensionen).
- Durchführung von Unit-Tests und Benutzerakzeptanztests zur Validierung der Anforderungen.

Meilensteine:

- **01.04.2025: Projektbeginn:**  
Start der Analysephase und Erstellung des Projektplans.
- **08.04.2025: Abschluss der Anforderungsanalyse:**  
Fertigstellung der Anforderungen und des Pflichtenhefts.
- **22.04.2025: Fertigstellung des Prototyps:**  
Entwicklung eines funktionalen Prototyps mit den Kernfunktionen.
- **20.05.2025: Abschluss der Implementierungsphase:**  
Integration aller Funktionen und Optimierung der Anwendung.

- **03.06.2025: Start der Testphase:**  
Beginn der Funktionstests, Integrationstests und Usability-Tests.
- **24.06.2025: Projektabschluss und Übergabe:**  
Übergabe der finalen Version inklusive Dokumentation und technischer Support.

Die Fortschrittskontrolle erfolgt wöchentlich. Soll- und Istwerte werden tabellarisch dokumentiert und regelmäßig im Projektbericht aktualisiert.

## 8. Entwicklungsumgebung

### 8.1 Tools

- **Entwicklungsumgebung:** VS-Code
- **Versionierung:** Git
- **Build-Tool:** Maven
- **Testframework:** JUnit
- **Dokumentationstool:** Javadoc

### 8.2 Hardware

- Standard-PC mit mindestens 8 GB RAM und 500 GB Speicherplatz.
- Betriebssysteme: Linux (Ubuntu 20.04) und Windows 10.

### 8.3 Software

- Java Development Kit (JDK) 11 oder höher.
- JSON-Library zur Datenspeicherung.

### 8.4 Orgware

- **Projektmanagement-Tool:** Jira oder Trello
- **Kollaboration:** Slack oder Microsoft Teams

## 9. Glossar

- **AG:** Auftraggeber
- **AN:** Auftragnehmer
- **FA:** Funktionale Anforderungen
- **UI:** User Interface (Benutzerschnittstelle)
- **JSON:** Datenformat zur strukturierten Datenspeicherung
- **Prototyp:** Erste, vereinfachte Version der Anwendung, um grundlegende Funktionen zu testen und Feedback zu sammeln
- **Release Candidate:** Letzte Testversion vor der finalen Veröffentlichung
- **Javadoc:** Tool zur Erstellung von Dokumentationen aus Java-Kommentaren

- **API:** Application Programming Interface, eine Schnittstelle, die es verschiedenen Softwarekomponenten ermöglicht, miteinander zu kommunizieren
- **GUI (Graphical User Interface):**  
Eine grafische Benutzeroberfläche, über die Benutzer mit der Anwendung interagieren können.
- **JSON-Library:**  
Eine Bibliothek zur Verarbeitung von JSON-Daten (JavaScript Object Notation).
- **Repository:**  
Ein zentraler Ort zur Speicherung und Verwaltung des Quellcodes und aller Projektartefakte, üblicherweise mithilfe eines Versionskontrollsystems wie Git.
- **Framework:**  
Eine Sammlung von vordefinierten Klassen und Funktionen, die die Entwicklung einer Anwendung erleichtern und beschleunigen.