



LIBRARY – DESIGN

Tatjana Russo & Jan Löliger

06.04.2022



Inhaltsverzeichnis

1. Beschreibung	2
2. UseCase	3
2.1 UseCase – Diagramm	3
2.2 UseCase – Beschreibung	3
3. Domain Model	7
4. CRC-Karten.....	8
5. Klassendiagramm.....	10
6. Überarbeitung	11
6.1 Use Case Diagram	11

1. Beschreibung

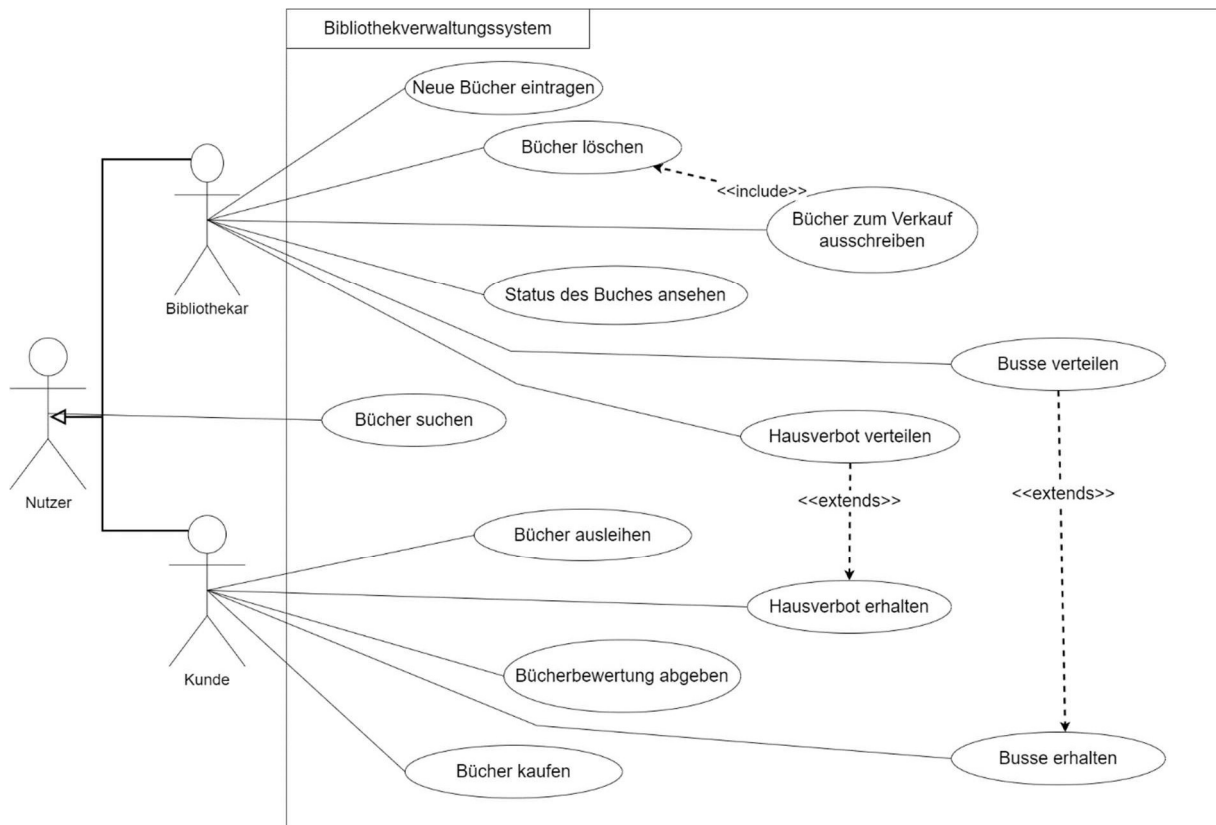
Bibliothekverwaltungssystem

Bei unserer Applikation kann man als Bibliothekar einsehen, was für Bücher es gibt in der Bibliothek, welcher Kategorie sie angehören und ob sie jemand geliehen hat. Man kann ebenfalls als Bibliothekar neue Bücher eintragen, löschen und für den Verkauf ausschreiben.

Kunden der Bibliothek können zu Büchern eine Bewertung abgeben und Bücher ausleihen. Kunden können für verspätete Abgaben & Sachbeschädigung eine Busse und oder Hausverbot erhalten.

2. UseCase

2.1 UseCase – Diagramm



2.2 UseCase – Beschreibung

Use Case #1	Neue Bücher eintragen
Actors	Bibliothekar
Pre-Condition	Buch existiert
Description of Use Case in detail	Bibliothekar kann neue Bücher im System hinzufügen
Post-Condition	Verfügbar in der Bibliothek
Exceptions (what can go wrong, how will the system respond?)	Falscher ISBN -> Meldung

Use Case #2	Bücher löschen
Actors	Bibliothekar
Pre-Condition	Buch ist eingetragen

Description of Use Case in detail	Bibliothekar kann bereits vorhandene Bücher wieder aus dem System entfernen
Post-Condition	Nicht mehr verfügbar in der Bibliothek
Exceptions (what can go wrong, how will the system respond?)	Buch ist nicht eingetragen -> Meldung

Use Case #3	Status des Buches ansehen
Actors	Bibliothekar
Pre-Condition	Buch ist eingetragen
Description of Use Case in detail	Bibliothekar kann den Status eines Buches ansehen. Geliehen oder nicht geliehen.
Post-Condition	-
Exceptions (what can go wrong, how will the system respond?)	-

Use Case #4	Bücher zum Verkauf ausschreiben
Actors	Bibliothekar
Pre-Condition	Buch ist eingetragen
Description of Use Case in detail	Bibliothekar kann ein Buch zum Verkauf ausschreiben
Post-Condition	Buch wird aus dem System entfernt
Exceptions (what can go wrong, how will the system respond?)	Buch existiert nicht in der Bibliothek -> Meldung

Use Case #5	Busse verteilen
Actors	Bibliothekar
Pre-Condition	Buch ist beschädigt / zu spät abgegeben worden.
Description of Use Case in detail	Bibliothekar kann bei Sachbeschädigung oder verspäteter Abgabe eine Busse verteilen
Post-Condition	Busse wird von Kunde bezahlt
Exceptions (what can go wrong, how will the system respond?)	-

Use Case #6	Hausverbot verteilen
Actors	Bibliothekar
Pre-Condition	Zweite Busse verteilt und oder schwere Sachbeschädigung

Description of Use Case in detail	Bibliothekar kann bei erneuter Busse oder schwerer Sachbeschädigung ein Hausverbot verteilen.
Post-Condition	Kunde wird auf «blacklist» gesetzt.
Exceptions (what can go wrong, how will the system respond?)	-

Use Case #7	Busse erhalten
Actors	Kunde
Pre-Condition	Buch ist beschädigt / zu spät abgegeben worden.
Description of Use Case in detail	Kunde erhält eine Busse
Post-Condition	Busse wird von Kunde bezahlt
Exceptions (what can go wrong, how will the system respond?)	-

Use Case #8	Hausverbot erhalten
Actors	Kunde
Pre-Condition	Zweite Busse verteilt und oder schwere Sachbeschädigung
Description of Use Case in detail	Kunde erhält einen Hausverbot
Post-Condition	Kunde wird auf «blacklist» gesetzt
Exceptions (what can go wrong, how will the system respond?)	-

Use Case #9	Bücher ausleihen
Actors	Kunde
Pre-Condition	Buch existiert.
Description of Use Case in detail	Kunde kann ein Buch oder mehrere Bücher ausleihen
Post-Condition	Buch ist nicht mehr verfügbar
Exceptions (what can go wrong, how will the system respond?)	Buch existiert nicht und oder ist nicht verfügbar

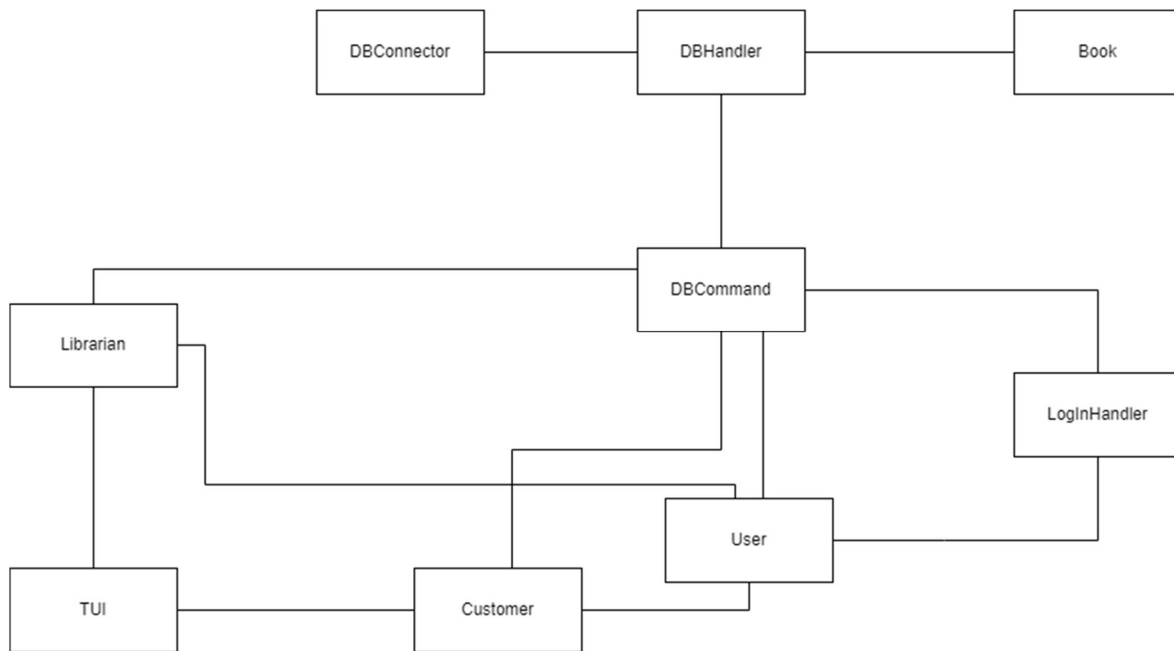
Use Case #10	Buchbewertung abgeben
Actors	Kunde
Pre-Condition	Buch existiert und wurde bereits vom Kunden ausgeliehen.
Description of Use Case in detail	Kunde kann eine Bewertung über ein bereits ausgeliehens Buch abgeben
Post-Condition	Buch erhält Bewertung

Exceptions (what can go wrong, how will the system respond?)	Buch existiert nicht und oder wurde nicht ausgeliehen
--	---

Use Case #11	Bücher kaufen
Actors	Kunde
Pre-Condition	Buch existiert und ist zum Verkauf ausgeschrieben
Description of Use Case in detail	Kunde kann ein Buch kaufen
Post-Condition	Buch ist nicht mehr verfügbar
Exceptions (what can go wrong, how will the system respond?)	Buch wurde nicht zum Verkauf ausgeschrieben bzw. existiert nicht.

Use Case #12	Bücher suchen
Actors	Nutzer
Pre-Condition	-
Description of Use Case in detail	Nutzer kann ein Buch suchen.
Post-Condition	Buch wird angezeigt oder falls es nicht existiert werden ähnliche Bücher ausgegeben.
Exceptions (what can go wrong, how will the system respond?)	-

3. Domain Model



4. CRC-Karten

Book	
<ul style="list-style-type: none"> • Getter and Setter for Book data, such as author, genre, title 	<ul style="list-style-type: none"> • DBHandler • User • Librarian • Customer • DBCommand

DBConnector	
<ul style="list-style-type: none"> • connects to the library db 	<ul style="list-style-type: none"> • DBHandler

DBHandler	
<ul style="list-style-type: none"> • Handles database requests 	<ul style="list-style-type: none"> • DBConnector • Book • DBCommand

User	
<ul style="list-style-type: none"> • Handles book search requests 	<ul style="list-style-type: none"> • DBCommand • TUI • LoginHandler • Book

Librarian	
<ul style="list-style-type: none"> • can add, remove, sell and see the status of books. • can give out fines and or ban customer accounts 	<ul style="list-style-type: none"> • DBCommand • TUI • Book

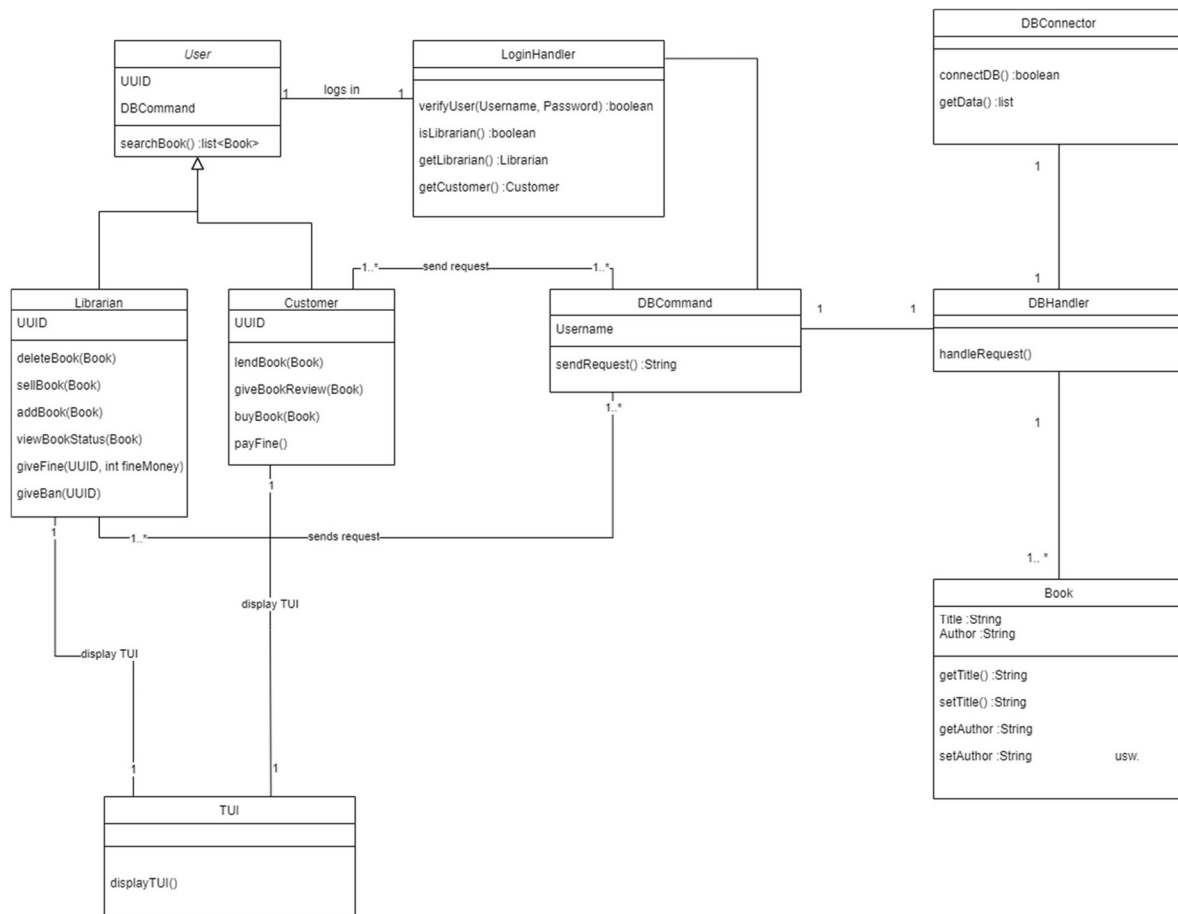
Customer	
<ul style="list-style-type: none"> • Can borrow books • Can send in a book review • Can receive and pay off a fine 	<ul style="list-style-type: none"> • DBCommand • TUI • Book

LoginHandler	
<ul style="list-style-type: none"> • handles log in requests 	<ul style="list-style-type: none"> • User • TUI • DBCommand

TUI	
<ul style="list-style-type: none"> • Text User Interface 	<ul style="list-style-type: none"> • LoginHandler • User • Librarian • Customer

DBCommand	
<ul style="list-style-type: none"> • Creates SQL-command 	<ul style="list-style-type: none"> • DBHandler • Book • User • Customer • Librarian • LoginHandler

5. Klassendiagramm



6. Überarbeitung

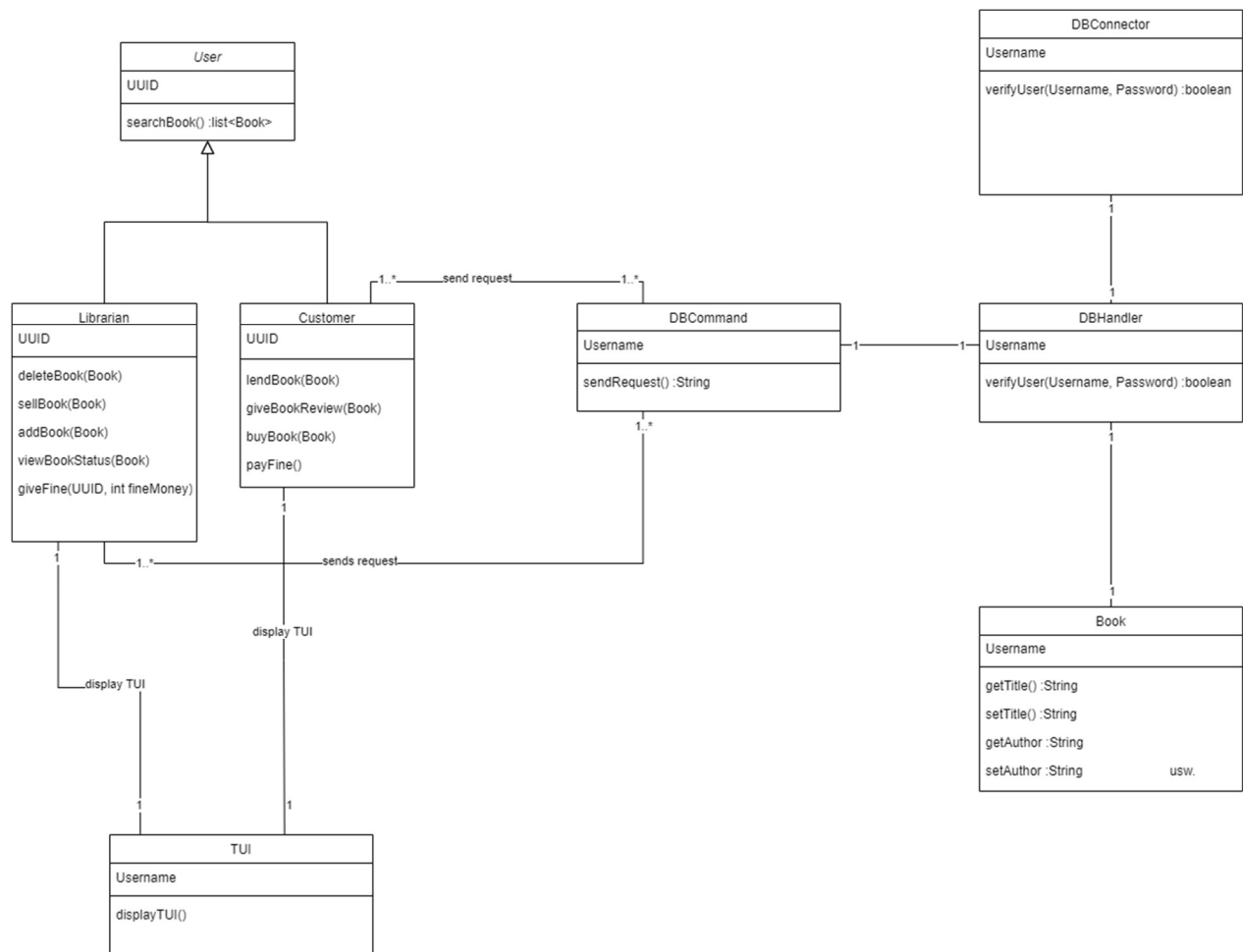
6.1 Use Case Diagramm



Durch Zeitmangel mussten wir zwei bzw. drei Use Cases streichen.

1. Bücherbewertung abgeben als Kunde:
Da dieses Use Case eher weniger wichtig war für uns und wir mehr Priorität auf die anderen setzten, mussten wir dieses streichen, da es im Vergleich zu den anderen viel zeitaufwendiger war und wir auch an der Datenbank ändern durchführen mussten.
2. Hausverbot geben (und auch bekommen):
Wir mussten uns zwischen „Busse verteilen“ und „Hausverbot verteilen“ entscheiden, da wir zu wenig Zeit hatten und entschieden uns für die Busse, da diese eine höhere Bedeutung für uns hat und in der Praxis öfter verwendet wird als das Hausverbot.

6.2 Klassendiagramm



LoginHandler:

Es hat für uns eher weniger Sinn ergeben eine extra Klasse für das Login zu erstellen. Wir haben diese Methode einfach bereits im DBHandler implementiert.

7. Testfälle

Testnr.	1
Name:	Buch erstellen
Beschreibung:	Das Buch wird in der Datenbank erstellt.
Erwartetes Ergebnis	Buch wird mit den Daten eingelesen.
Tatsächliches Ergebnis:	Buch wird mit den Daten eingelesen.

Testnr.	2
Name:	Buch löschen
Beschreibung:	Buch wird aus der Datenbank gelöscht.
Erwartetes Ergebnis	Buch existiert nicht mehr
Tatsächliches Ergebnis:	Buch existiert nicht mehr

Testnr.	3
Name:	Buch zum Verkauf ausschreiben
Beschreibung:	Bibliothekar kann ein Buch zum Verkauf ausschreiben.
Erwartetes Ergebnis	Buch wird aus der Datenbank gelöscht.
Tatsächliches Ergebnis:	Buch wird gelöscht

Testnr.	4
Name:	Buch suchen
Beschreibung:	Buch nach seiner ID, Titel, Autor etc suchen.
Erwartetes Ergebnis	Buch wird ausgegeben
Tatsächliches Ergebnis:	Buch wird ausgegeben

Testnr.	5
Name:	Nutzer hinzufügen
Beschreibung:	Einen neuen Nutzer/Kunden hinzufügen
Erwartetes Ergebnis	Nutzer kann sich registrieren & wird in der Datenbank abgespeichert
Tatsächliches Ergebnis:	Nutzer wird in der Datenbank abgespeichert

8. Arbeitsaufteilung

Wir haben uns die Arbeit nach Klassen aufgeteilt. Meist fingen wir zusammen an und einer beendete es.

Was?	Wer?
Designphase	Tatjana, Jan
Dokumentation	Tatjana
Erstellen der Datenbank mit SQL	Tatjana
DBHandler, DBConnector	Jan
DBCommands	Jan, Tatjana
User => Librarian, Customer	Jan, Tatjana
Book	Jan, Tatjana
TUI	Jan
Main	Jan, Tatjana