

Anforderungsspezifikation

Problemstellung und Zielsetzung

In vielen Haushalten gibt es Bücher, die derzeit nicht gelesen, aber auch nicht dauerhaft abgegeben werden sollen. Gleichzeitig besteht häufig das Interesse, andere Werke aus der Nachbarschaft auszuleihen. Die Buchtausch-App soll dieses Bedürfnis aufgreifen und durch eine gemeinsame Datenbanklösung ermöglichen, dass Nutzer ihre Bücher verleihen und gleichzeitig Bücher anderer Personen ausleihen können. Ziel ist die Entwicklung einer relationalen Datenbank, die den gesamten Lebenszyklus einer Ausleihe abbildet – von der Erfassung der bibliographischen Daten bis hin zu Ausleihvorgängen, Bewertungen und Kommunikationsmöglichkeiten.

Besonderer Wert wird auf Datenschutz gelegt: Während der Benutzername öffentlich sichtbar ist, bleiben persönliche Daten wie Name, Adresse und E-Mail geschützt und sind nur für Transaktionen oder durch Administratoren einsehbar.

Rollen

- Benutzer: natürliche Person mit Konto, kann Ausleiher, Verleiher und/oder Administrator sein.
- Verleiher: stellt eigene Buch-Exemplare in die Datenbank ein und verwaltet deren Verleihstatus.
- Ausleiher: leiht verfügbare Exemplare aus, gibt diese zurück und kann Bewertungen abgeben.
- Administrator: pflegt Daten, unterstützt Benutzer bei Problemen und sichert die Konsistenz der Datenbank.

Aktionen der Rollen

Ein Benutzer registriert sich mit persönlichen Daten wie Name, Benutzername, Adresse, E-Mail und Passwort. Der Benutzername dient als öffentliche Kennung, während die übrigen Daten vertraulich bleiben. Benutzer können die Datenbank durchsuchen und nach Kriterien wie Titel, Autor, Genre, Sprache, Ort oder ISBN filtern. Außerdem können sie Bewertungen in Form von Sternen und Kommentaren abgeben, ihre eigenen Einträge bearbeiten oder löschen und über ein Chatsystem mit anderen Nutzern kommunizieren. Für den Support steht eine Hilfeanfrage-Funktion zur Verfügung.

In der Rolle des Verleihers kann ein Benutzer Bücher einstellen. Neben allgemeinen Angaben wie Titel, Autor, Verlag, Sprache, Erscheinungsjahr, Genre und ISBN werden auch Informationen zu Zustand, Standort, Abholmöglichkeit, Versandoptionen und maximaler Ausleihdauer gespeichert. Der Verleiher verwaltet zudem den Verleihstatus jedes Exemplars (z. B. verfügbar, ausgeliehen oder reserviert).

Als Ausleiher kann ein Benutzer eine Ausleihe starten und diese mit Start- und Enddatum erfassen. Nach Rückgabe bestätigt er die Transaktion. Zusätzlich können

Exemplare reserviert werden, um sie für einen bestimmten Zeitraum vorzumerken.

Der Administrator verwaltet Benutzerkonten, legt Testaccounts an, kann Nutzer sperren oder löschen und hat die Möglichkeit, Daten zu korrigieren. Außerdem stellt er die Systemintegrität durch Konsistenzprüfungen sicher und kann fehlerhafte Buch- oder Ausleiheinträge beenden.

Datenanforderungen

Zentrale Entität ist der Benutzer mit Stammdaten wie Benutzername, Name, E-Mail, Passwort-Hash und Adresse. Da Adressen mehrfach genutzt werden können (z. B. bei Benutzern und Verlagen), werden sie in einer separaten Entität gespeichert.

Ein BuchWerk bildet die bibliographische Einheit ab und enthält Informationen wie ISBN, Titel, Erscheinungsjahr, Sprache, Verlag, Autoren und Genre. Autoren und Genres stehen jeweils in einer m:n-Beziehung zu den Werken, die über Hilfsentitäten (WerkAutor, WerkGenre) aufgelöst werden. Aus einem Werk können mehrere BuchExemplare entstehen, die einzelnen Verleihern gehören und Attribute wie Zustand, Verleihstatus, maximale Ausleihdauer und Abholort besitzen.

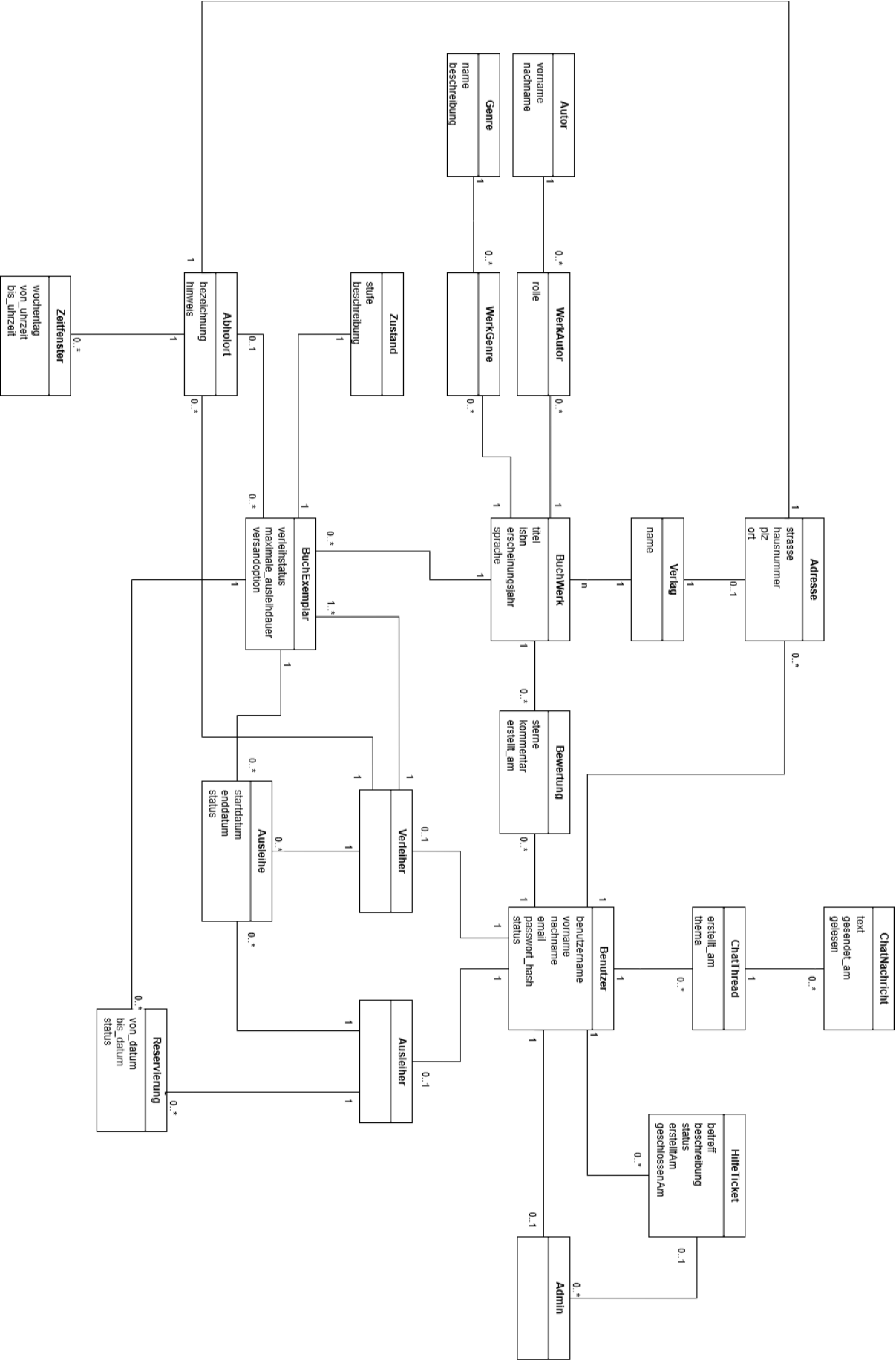
Der Prozess der Ausleihe stellt eine Dreifachbeziehung dar: Er verbindet einen Ausleiher, einen Verleiher und ein konkretes BuchExemplar miteinander. Eine weitere Dreifachbeziehung ergibt sich bei der Reservierung, die einen Ausleiher, ein Exemplar und einen Zeitraum zusammenführt. Ebenso sorgt das Bewertungssystem dafür, dass jedes Werk von einem Benutzer nur einmal bewertet werden kann.

Darüber hinaus erweitern ChatThreads, ChatNachrichten und HilfeTickets das Modell um Kommunikations- und Supportfunktionen. Diese bleiben gegenüber den Kernfunktionen sekundär, erhöhen aber die Praxistauglichkeit.

Zusammenfassung

Die Buchtausch-App benötigt eine normalisierte Datenbankstruktur, die Rollen, Bücher, Ausleihen, Reservierungen, Bewertungen und Kommunikation abbildet. Durch die Trennung zwischen bibliographischen Werken und konkreten Exemplaren sowie durch Entitäten wie Autor, Genre und Verlag wird Redundanz vermieden. Das Modell enthält mehrere Dreifachbeziehungen (z. B. Ausleihe, Reservierung) und bildet so den realen Ablauf der App realistisch ab. Damit erfüllt die Datenbank die Anforderungen, Bücher effizient zu verwalten, Datenschutz sicherzustellen und Transaktionen transparent abzubilden.

ER Model



Datenwörterbuch

Benutzer

- benutzer_id (INT, PK, AUTO_INCREMENT)
- benutzername (VARCHAR (50), UNIQUE, NOT NULL)
- vorname (VARCHAR (100), NOT NULL)
- nachname (VARCHAR (100), NOT NULL)
- email (VARCHAR (255), UNIQUE, NOT NULL)
- passwort_hash (VARCHAR (255), NOT NULL)
- adresse_id (INT, FK → adresse)
- status (VARCHAR (10), NOT NULL)

Admin

- admin_id (INT, PK, FK → benutzer.benutzer_id)
Hinweis: Spezialisierung von Benutzer, kennzeichnet Administratorrechte.

Verleiher

- verleih_id (INT, PK, FK → benutzer.benutzer_id)
Hinweis: Spezialisierung von Benutzer, kennzeichnet Benutzer, die Exemplare verleihen.

Ausleiher

- ausleiher_id (INT, PK, FK → benutzer.benutzer_id)
Hinweis: Spezialisierung von Benutzer, kennzeichnet Benutzer, die Exemplare ausleihen.

Adresse

- adresse_id (INT, PK)
- strasse (VARCHAR (120))
- hausnummer (VARCHAR (10))
- plz (VARCHAR (10))
- ort (VARCHAR (120))

BuchWerk

- werk_id (INT, PK)
- titel (VARCHAR (255), NOT NULL)
- isbn (CHAR (13), UNIQUE, NOT NULL)
- erscheinungsjahr (INT)
- sprache (VARCHAR (40))
- verlag_id (INT, FK → verlag)

Verlag

- verlag_id (INT, PK)
- name (VARCHAR (255), UNIQUE, NOT NULL)
- adresse_id (INT, FK → adresse)

Autor

- autor_id (INT, PK)
- vorname (VARCHAR (100))
- nachname (VARCHAR (100))

WerkAutor (m:n)

- werk_id (INT, FK → buch_werk, PK-Teil)
- autor_id (INT, FK → autor, PK-Teil)
- rolle (VARCHAR (100))

Genre

- genre_id (INT, PK)
- name (VARCHAR (80), UNIQUE)
- beschreibung (TEXT)

WerkGenre (m:n)

- werk_id (INT, FK → buch_werk, PK-Teil)
- genre_id (INT, FK → genre, PK-Teil)

BuchExemplar

- exemplar_id (INT, PK)
- werk_id (INT, FK → buch_werk)
- verleih_id (INT, FK → verleih)
- zustand_id (INT, FK → zustand)
- verleihstatus (VARCHAR (20))
- max_ausleihdauer (INT)
- abholort_id (INT, FK → abholort)
- versandoption (BOOLEAN)

Zustand

- zustand_id (INT, PK)
- stufe (VARCHAR (30), NOT NULL, z. B. „neu“, „gut“, „gebraucht“)
- beschreibung (TEXT)

Abholort

- abholort_id (INT, PK)
- verleih_id (INT, FK → verleih)
- adresse_id (INT, FK → adresse)
- bezeichnung (VARCHAR (120), NOT NULL, z. B. „Haustür“, „Packstation“)
- hinweis (TEXT)

Zeitfenster

- zeitfenster_id (INT, PK)
- abholort_id (INT, FK → abholort)
- wochentag (INT, 0–6, So–Sa)
- von_uhrzeit (TIME)
- bis_uhrzeit (TIME)

Ausleihe (Dreifachbeziehung: Ausleiher – Verleiher – Exemplar)

- ausleihe_id (INT, PK)
- exemplar_id (INT, FK → buch_exemplar)
- verleih_id (INT, FK → verleih)
- ausleiher_id (INT, FK → ausleiher)
- startdatum (DATE)
- enddatum (DATE)
- status (VARCHAR (20))

Reservierung (Dreifachbeziehung: Ausleiher – Exemplar – Zeitraum)

- reservierung_id (INT, PK)
- exemplar_id (INT, FK → buch_exemplar)
- ausleiher_id (INT, FK → ausleiher)
- von_datum (DATE)
- bis_datum (DATE)
- status (VARCHAR (20))

Bewertung

- bewertung_id (INT, PK)
- benutzer_id (INT, FK → benutzer)
- werk_id (INT, FK → buch_werk)
- sterne (INT, 1–5)
- kommentar (VARCHAR (255))
- erstellt_am (TIMESTAMP)

ChatThread

- thread_id (INT, PK)
- erstellt_am (TIMESTAMP)
- thema (VARCHAR (255))

ChatNachricht

- nachricht_id (INT, PK)
- thread_id (INT, FK → chat_thread)
- sender_id (INT, FK → benutzer)
- text (TEXT, NOT NULL)
- gesendet_am (TIMESTAMP)
- gelesen (BOOLEAN, DEFAULT FALSE)

HilfeTicket

- ticket_id (INT, PK)
- ersteller_id (INT, FK → benutzer)
- betreff (VARCHAR (255))
- beschreibung (VARCHAR (2000))
- status (VARCHAR (20), z. B. „offen“, „in Bearbeitung“, „geschlossen“)
- erstellt_am (TIMESTAMP)
- geschlossen_am (TIMESTAMP)