

Bibliotheksverwaltung BSEVITA

Ein umfassendes MVC-Webprojekt für moderne Schulbibliotheken mit durchdachtem Datenmodell und praxisorientierten Designvorschlägen



Ausgangssituation und Anforderungen

Die BSEVITA-Schulbibliothek steht vor der Herausforderung, einen Bestand von ca. 5.000–9.000 Büchern effizient zu verwalten. Mit durchschnittlich 50 Ausleihen pro Tag ist ein robustes, benutzerfreundliches System erforderlich, das sowohl die Anforderungen der Schülerinnen und Schüler als auch die der Bibliotheksverwaltung erfüllt.

Das System muss eine intuitive Suchfunktion bieten, die nach Buchtitel, Autor bzw. Herausgeber sowie Sachgebiet filtert. Dabei werden für Schülerinnen und Schüler grundlegende Identifikationsdaten erfasst: Ausweisnummer (eindeutig), Vorname und Nachname. Die Ausweisnummer dient als zentrale Kennung und muss systemweit einmalig sein.

Schülerdaten

- Ausweisnummer (eindeutig)
- Vorname
- Nachname

Buchdaten

- Buchnummer (Format: xxxxx-jjjj)
- Sachgebiet, ISBN
- Titel, Autor/Herausgeber
- Verlag, Verlagsort
- Erscheinungsdatum

Tägliche Ausleihen

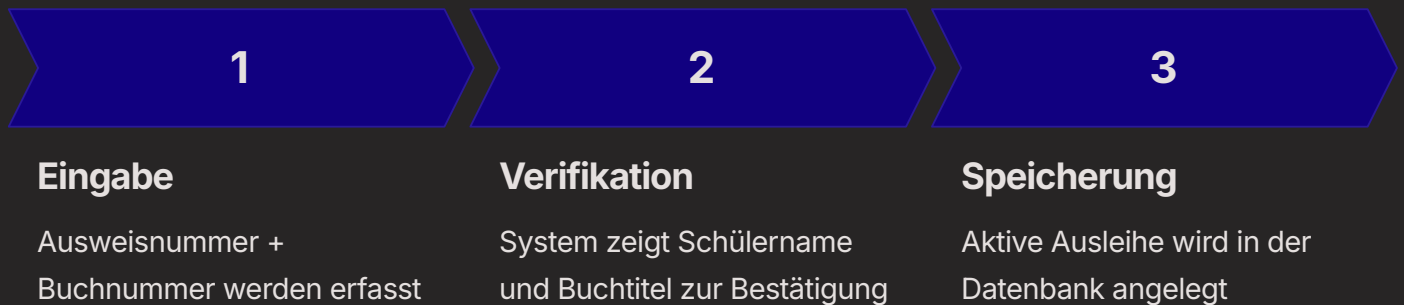
Durchschnittlich **50 Bücher** pro Tag

Bestand: **5.000–9.000** Bücher

Für Bücher werden umfangreiche Metadaten verwaltet, darunter die Buchnummer im speziellen Format „xxxxx-jjjj“ (fünf Ziffern, Bindestrich, vierstellige Jahreszahl), Sachgebiet, ISBN, Titel, Autor bzw. Herausgeber, Verlag, Verlagsort sowie Erscheinungsdatum. Diese Datenstruktur ermöglicht eine präzise Katalogisierung und schnelle Auffindbarkeit.

Prozessbeschreibung: Ausleihe und Rückgabe

Der Ausleihprozess ist auf Effizienz und Fehlerminimierung ausgelegt. Beim Ausleihen werden Ausweisnummer und Buchnummer eingegeben. Das System zeigt anschließend zur Verifikation zusätzliche Informationen an – beispielsweise den vollständigen Schülernamen und den Buchtitel. Diese Anzeige dient der Kontrolle und verhindert Fehleingaben, bevor die Ausleihe final gespeichert wird.



Bei der Rückgabe wird ausschließlich die Buchnummer eingegeben. Existiert eine aktive Ausleihe zu dieser Buchnummer, zeigt das System alle dazugehörigen Informationen in einem read-only Modus an. Die Bibliothekarin oder der Bibliothekar kann die Daten einsehen, jedoch nicht mehr bearbeiten. Nach erfolgter Rückgabe greift eine besondere Datenschutzanforderung: Personenbezogene Daten dürfen nicht dauerhaft in der Rückgabe- oder Statistikablage im Klartext gespeichert bleiben. Trotzdem muss das System statistische Auswertungen ermöglichen – beispielsweise die Anzahl der Ausleihen pro Monat oder Sachgebiet. Hierfür wird ein Konzept benötigt, das aggregierte, anonymisierte Daten speichert, ohne die Zuordnung zu einzelnen Personen zu bewahren.

Ein weiterer wichtiger Aspekt ist die On-Demand-Erfassung von Büchern. Da die vollständige Eingabe aller ca. 6.000 Bücher zu Projektbeginn nicht praktikabel ist, werden Buchstammdaten erst dann erfasst, wenn ein Buch zum ersten Mal ausgeliehen wird. Dies reduziert den initialen Aufwand erheblich und ermöglicht einen schrittweisen Aufbau der Datenbank während des laufenden Betriebs.

Fachliche Anforderungen an die Datenbank

Die Datenbank muss eine Reihe fachlicher und technischer Anforderungen erfüllen, um Integrität, Eindeutigkeit und Auswertbarkeit sicherzustellen. Diese Anforderungen bilden das Fundament für ein zuverlässiges Bibliothekssystem.

1

Eindeutigkeit Ausweisnummern

Ausweisnummern dürfen nicht doppelt vergeben werden. Jede Schülerin und jeder Schüler erhält eine systemweit einmalige Kennung.

2

Format Buchnummer

Buchnummern dürfen nicht doppelt vorkommen und müssen dem Format „xxxxx-xxxx“ entsprechen. Ein Check-Constraint in der Datenbank stellt dies sicher.

3

Aktive Ausleihe

Ein Buch darf zur selben Zeit maximal einmal aktiv ausgeliehen sein. Dies verhindert Konflikte und doppelte Ausleihen desselben Exemplars.

4

Reports und Anfragen

Das System muss verschiedene Reports liefern können, darunter Mahnlisten, Suchfunktionen und Übersichten über aktive Ausleihen.

Die Report-Anforderungen umfassen unter anderem eine Mahnliste, die alle Schülerinnen und Schüler auflistet, die seit mehr als drei Monaten ausgeliehene Bücher nicht zurückgegeben haben. Diese Liste wird quartalsweise durch die Bibliothekarin oder den Bibliothekar ausgeführt. Weiterhin muss eine Suche nach Sachgebiet, Autor bzw. Herausgeber und Buchtitel möglich sein. Eine Liste aller Schülerinnen und Schüler, die aktuell mindestens ein Buch ausgeliehen haben, sowie eine Suche nach Schülernamen runden die Anforderungen ab.

Diese fachlichen Vorgaben erfordern ein durchdachtes Datenmodell mit klaren Beziehungen, Constraints und Indizes. Unique-Constraints sorgen für Eindeutigkeit, Fremdschlüssel für referenzielle Integrität, und Indizes beschleunigen Such- und Reportabfragen erheblich.

Technische Rahmenbedingungen

Das Projekt wird als **ASP.NET Core MVC Webanwendung** umgesetzt. Diese Architektur trennt klar zwischen Model, View und Controller und ermöglicht eine wartbare, erweiterbare Codebasis. Die Datenhaltung erfolgt in **Microsoft SQL Server (MSSQL)**, einer robusten und weit verbreiteten relationalen Datenbank, die sich optimal für Schulumgebungen eignet.

01

ASP.NET Core MVC

Moderne Webanwendung mit klarer Trennung von Logik, Darstellung und Daten

02

Microsoft SQL Server

Zuverlässige Datenhaltung mit robusten Transaktions- und Integritätsfunktionen

03

Entity Framework Core

ORM mit Wahlfreiheit zwischen Code First (inkl. Migrationen) oder Database First (inkl. Scaffold)

04

MVC-Views

Einfache, übersichtliche Frontend-Gestaltung mit Fokus auf Funktion

Als Object-Relational Mapper (ORM) kommt **Entity Framework Core** zum Einsatz. Sie haben die Wahlfreiheit zwischen zwei Ansätzen: **Code First**, bei dem das Datenmodell im C#-Code definiert und über Migrationen in die Datenbank überführt wird, oder **Database First**, bei dem eine bestehende Datenbank mittels Scaffold in Entity-Klassen abgebildet wird. Beide Ansätze haben ihre Vorzüge – Code First bietet mehr Flexibilität bei der Modellierung, während Database First ideal ist, wenn das Datenbankschema bereits existiert.

Das Frontend-Design konzentriert sich auf einfache, übersichtliche MVC-Views. Es sind keine speziellen UI-Frameworks erforderlich; der Fokus liegt auf Funktion und Benutzerfreundlichkeit. Die Views sollten jedoch klar strukturiert sein und eine intuitive Navigation ermöglichen, damit Bibliothekspersonal und Schülerinnen und Schüler das System ohne lange Einarbeitungszeit nutzen können.

Aufgaben: Datenmodellierung und Implementierung

Die Projektumsetzung gliedert sich in mehrere aufeinander aufbauende Aufgaben, die vom konzeptionellen Datenmodell bis zur vollständigen MVC-Webanwendung reichen.



Datenmodellierung (ERD)

Erstellen Sie ein Entity-Relationship-Diagramm, das alle beschriebenen Entitäten und Beziehungen abbildet. Achten Sie auf fachliche Korrektheit, insbesondere die Trennung von Ausleihe und Statistik zur Erfüllung der Datenschutzerfordernisse. Abgabe als Bild oder PDF.



Relationenschema

Überführen Sie das ER-Diagramm in ein Relationenschema mit Tabellen, Primär- und Fremdschlüsseln, Kardinalitäten und notwendigen Unique-Constraints.



MSSQL-Implementierungsmodell

Erstellen Sie ein MSSQL-konformes Modell mit geeigneten Datentypen, IDENTITY-Definitionen, Constraints (Unique, Not Null, Check für Buchnummer-Format) und Indizes für Such- und Reportabfragen.



SQL-Script zur Datenbankanlage

Erstellen Sie ein vollständiges SQL-Script, das Datenbank und Tabellen anlegt, Constraints und Indizes erzeugt sowie referenzielle Integrität sicherstellt.



MVC-Webanwendung

Implementieren Sie eine ASP.NET Core MVC Anwendung mit CRUD-Funktionen für Schüler und Bücher, Ausleihe- und Rückgabeprozessen sowie Such- und Reportfunktionen.

Die MVC-Webanwendung umfasst mehrere Funktionsbereiche: Die **Schülerverwaltung (CRUD)** ermöglicht das Anlegen, Anzeigen, Bearbeiten und Deaktivieren bzw. Löschen von Schülerdaten mit Validierung der eindeutigen Ausweisnummer. Die **Buchverwaltung (CRUD / On-Demand)** erfasst Bücher mit Validierung der eindeutigen Buchnummer im vorgegebenen Format. Besonders wichtig ist, dass die Bucherfassung im Ausleihprozess möglich ist, falls das Buch noch nicht existiert.

Der **Ausleihprozess** erfasst Ausweisnummer und Buchnummer, zeigt Verifikationsdaten an und speichert die aktive Ausleihe, wobei verhindert wird, dass ein bereits ausgeliehenes Buch erneut ausgeliehen wird. Die **Rückgabe** erfolgt über die Buchnummer, zeigt bestehende Ausleihinfos read-only an und setzt die Datenschutzerfordernisse um. Abgerundet wird die Anwendung durch **Such- und Reportfunktionen**, die nach Sachgebiet, Autor, Titel und Schülernamen suchen sowie Mahnlisten und Übersichten über aktive Ausleihen generieren.

Design-Vorschläge für die Benutzeroberfläche

Eine durchdachte Benutzeroberfläche ist entscheidend für die Akzeptanz und Effizienz des Bibliothekssystems. Im Folgenden werden konkrete Design-Vorschläge für verschiedene Bereiche der Anwendung präsentiert, die sowohl Funktionalität als auch Benutzerfreundlichkeit in den Vordergrund stellen.

