

Entwickler- dokumentation

Belegarbeit Software Engineering II

Gruppe 8 “Bibliotheksprozesse unterstützen”

1. Einleitung	2
2. Entwurf	2
3. Verwendete Programmiersprachen und Entwicklungsumgebungen	5
4. Datenbank	5
4.1. Datenbanktechnologie	5
4.2. Tabellen	5
4.2.1. Tabelle bibliotheksNutzer	7
4.2.2. Tabelle bibliotheksBuch	8
4.2.3. Tabelle ausleihe	8
5. Beschreibung der Klassen	9
5.1. Program.cs	9
5.2. Nutzer.cs	9
5.3. Form1.Designer.cs	10
5.4. Form1.cs	10
5.5. DBConnection.cs	10
5.6. DataBConnection.cs	10
5.7. Buch.cs	10
5.8. Ausleihe.cs	11
5.9. Form2.Designer	11
5.10. Form2	11

1. Einleitung

Diese Dokumentation soll die Weiterentwicklung und Wartung des Softwaresystems erleichtern und den Entwicklern ermöglichen, sich schnell und leicht in die bestehende Struktur des Softwaresystems einzuarbeiten.

2. Entwurf

Um eine spätere Erweiterung oder Änderung des Softwaresystems zu erleichtern sowie die Wiederverwendbarkeit der einzelnen Komponenten zu ermöglichen, wurde das Model-View-Controller-Konzept (MVC) verwendet. Dies wurde im Paketdiagramm abgebildet (siehe Abb. 1, S. 2).

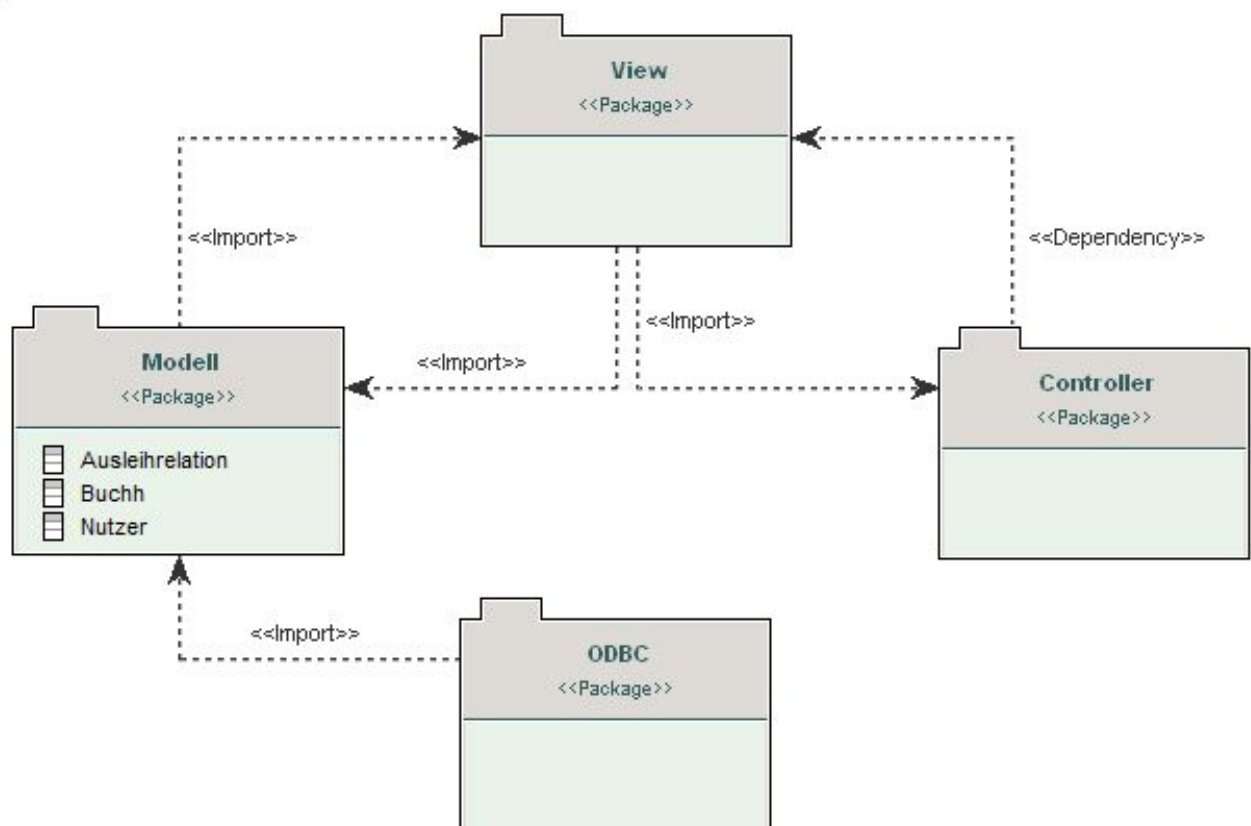


Abbildung 1: Paketdiagramm des Softwaresystems.

Daraus wurde das Klassendiagramm entwickelt, das die benötigten Klassen des Programmes mit Relationen abbildet (Abb. 2, S. 3).

Die zwei wichtigsten Funktionen des Programms sind in Sequenzdiagrammen dargestellt (Abb. 3 und Abb. 4, S. 4).

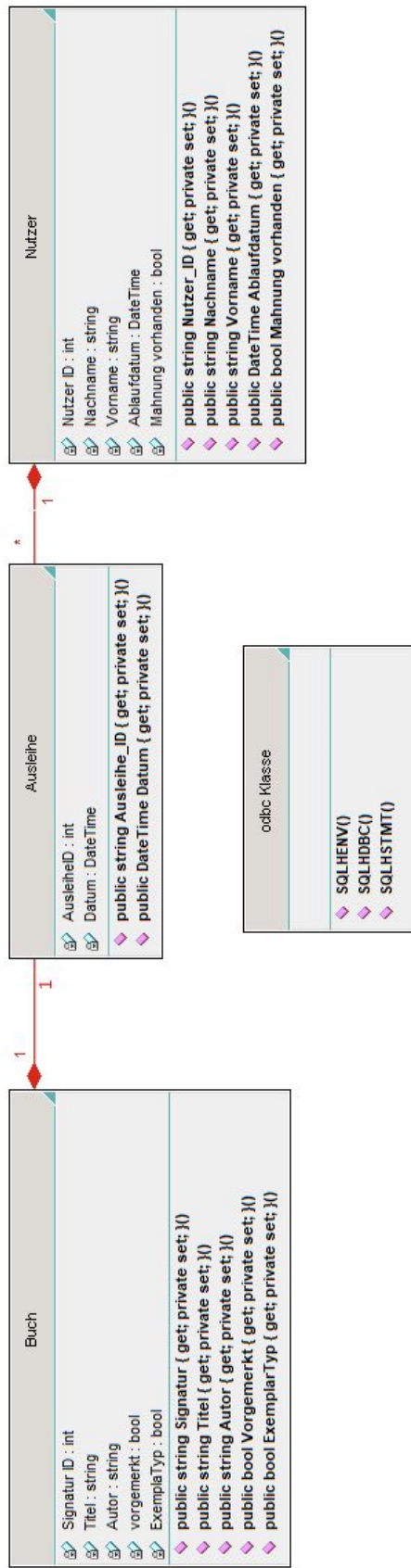


Abbildung 2: Klassendiagramm des Softwaresystems.

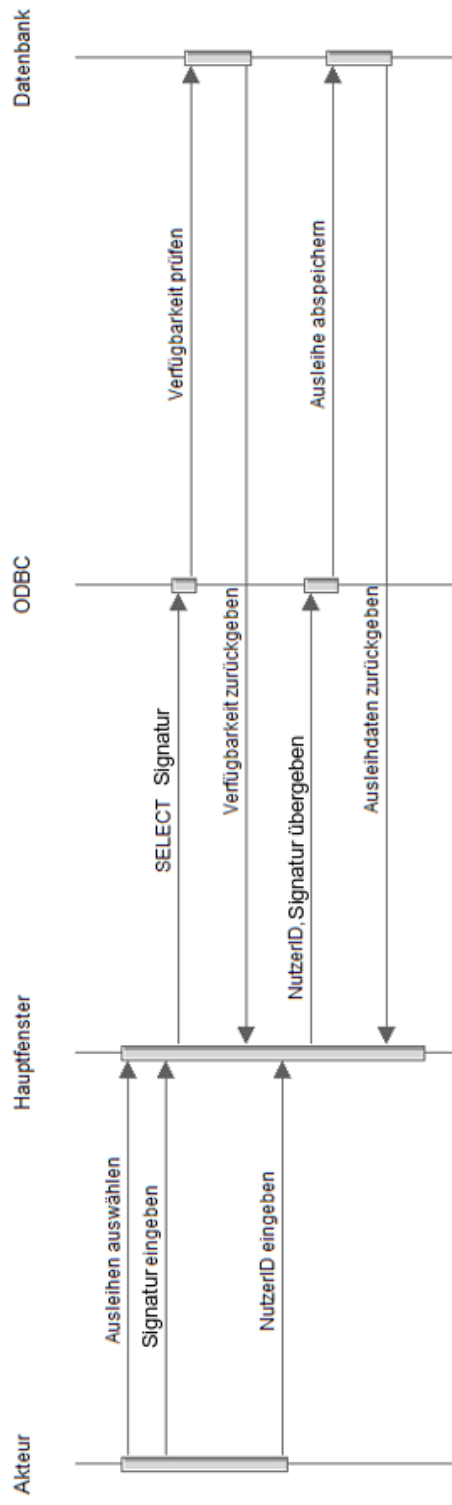


Abbildung 3: Sequenzdiagramm des Anwendungsfalls Buch ausleihen.

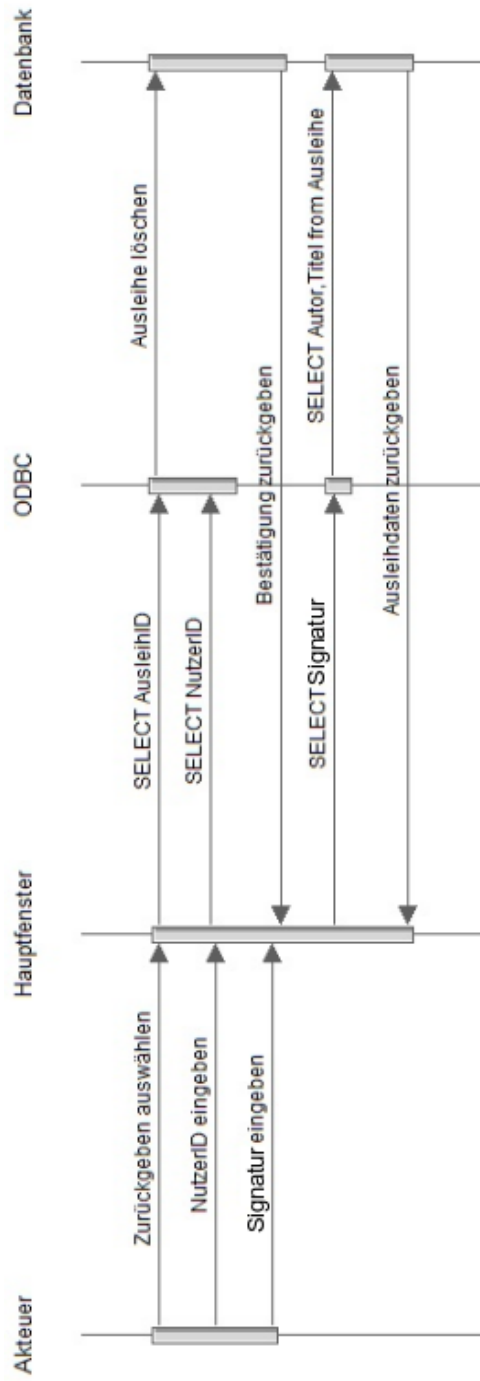


Abbildung 4: Sequenzdiagramm des Anwendungsfalls Buch zurückgeben.

3. Verwendete Programmiersprachen und Entwicklungsumgebungen

Für die Implementierung des Softwareprojekts wurde die Programmiersprache C# gewählt. Ausschlaggebend dafür war, dass sich C# einerseits als objektbasierte Programmiersprache sehr für die Implementierung der zu erstellenden Klassen eignet, andererseits ist der Implementierungsverantwortliche bereits im Umgang mit C # geübt.

Als Datenbankmodell wurde MySQL gewählt, da es als relationale Datenbank alle Möglichkeiten bietet, die unser Programm benötigt. Mit MySQL wurde bereits in anderen Lehrveranstaltungen Erfahrung gesammelt. In diesem Rahmen wurde bereits ein Datenbankserver für die Studenten angelegt, so dass dieser auch für die Umsetzung dieses Projekts genutzt werden konnte.

4. Datenbank

4.1. Datenbanktechnologie

Für die Datenbank wurde das relationale Datenbankmanagementsystem MySQL von Oracle verwendet und mittels ODBC an das Programm angebunden. Hier wurde der Datenbankserver idb45 der Fakultät Informatik/Mathematik der HTW Dresden genutzt, der für die Studenten angelegt wurde. Der Login erfolgt mittels der dazu ausgegebenen Nutzernummer und dem dazugehörigen Passwort und muss ggf. in der Klasse DBConnection manuell geändert werden.

4.2. Tabellen

In der Datenbank wurden die Tabellen bibliotheksNutzer, bibliotheksBuch und ausleihe angelegt. Die Tabelle ausleihe steht dabei mit bibliotheksNutzer und bibliotheksBuch in Relation (siehe ERM, Abb. 5, S. 6).

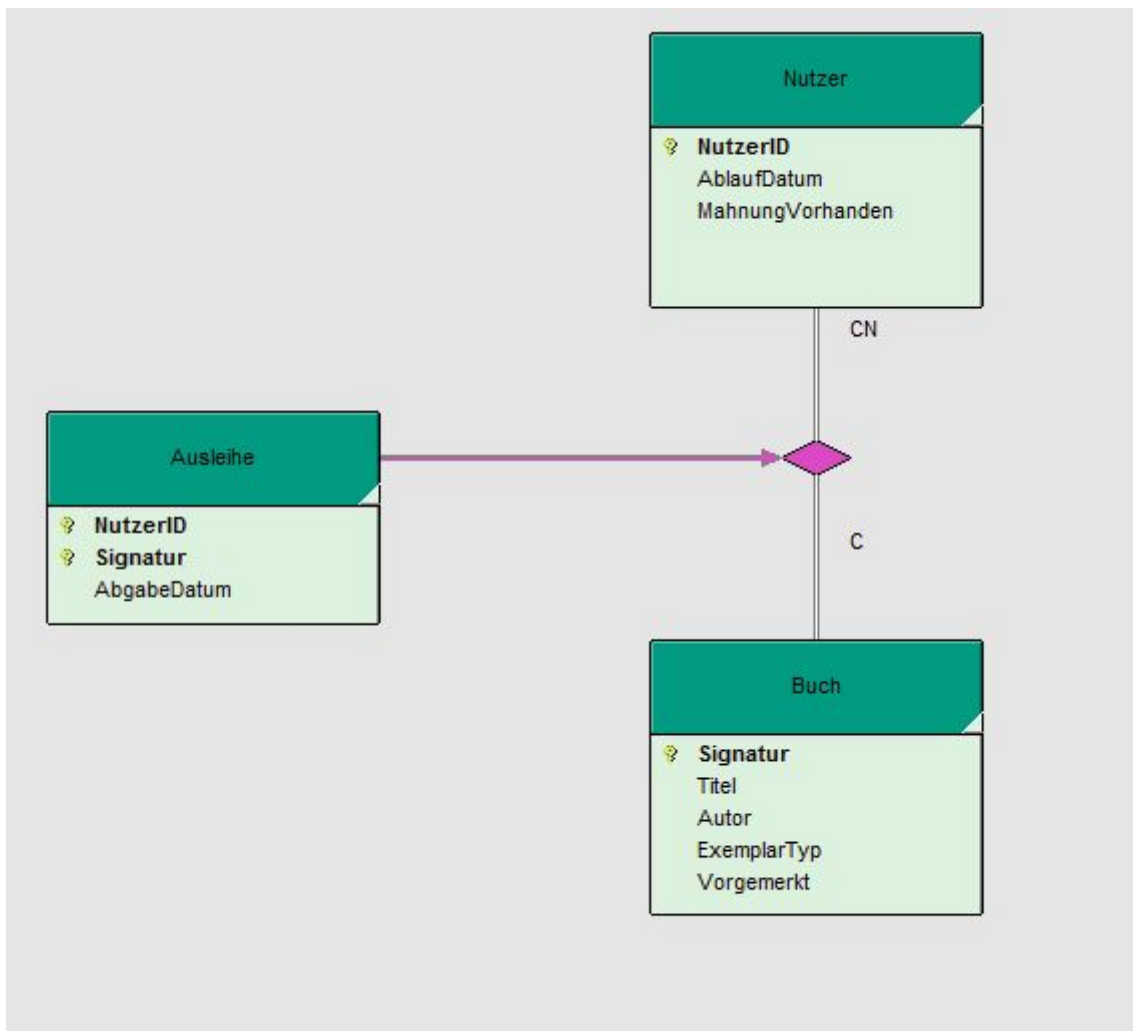


Abbildung 5: ERM der zugehörigen Datenbank.

4.2.1 Tabelle bibliotheksNutzer

Hier wird für jeden Benutzer der Bibliothek neben der eindeutigen Benutzer-ID, die den Primärschlüssel der Tabelle darstellt, das Ablaufdatum des Benutzerausweises und die Information darüber, ob der Benutzer seinen Verpflichtungen der Bibliothek gegenüber nachgekommen ist, abgespeichert. Diese Tabelle dient lediglich dazu, eine möglicherweise real vorhandene Nutzerverwaltung zu simulieren.

Attribut	Datentyp	Constraints	Bemerkung
nutzerid	CHAR(5)	NOT NULL PRIMARY KEY	enthält die eindeutige Benutzer-ID
abDatum	DATE		gibt den letzten gültigen Tag des Benutzerausweises an
mahnung	INT		gibt an, ob eine unbezahlte Mahnung vorliegt; 0 = keine Mahnung vorhanden, 1 = eine oder mehrere Mahnungen

4.2.2 Tabelle bibliotheksBuch

In dieser Tabelle wird für jedes Buch der Bibliothek die eindeutige Signatur gespeichert, die den Primärschlüssel der Tabelle darstellt. Daneben wird der Titel und Autor des Buches vermerkt. Das Attribut exTyp gibt Auskunft darüber, ob das Buch grundsätzlich ausleihbar ist, oder ob es sich um ein Belegexemplar handelt. Falls ein Buch vorgemerkt ist, wird im Attribut vorgemerkt die

Attribut	Datentyp	Constraints	Bemerkung
signatur	CHAR(8)	NOT NULL PRIMARY KEY	enthält die eindeutige Signatur eines Buches
titel	VARCHAR(50)		Titel des Buches
autor	VARCHAR(50)		Autor(en) des Buches
exTyp	INT		enthält Informationen darüber, ob das Buch grundsätzlich ausgeliehen werden kann; 0 = Präsenzexemplar, 1 = ausleihbar
vorgemerkt	CHAR(8)		enthält Informationen darüber, ob das Buch bereits vorgemerkt wurde; falls eine Vormerkung vorliegt, enthält dieses Attribut die 8-stellige Benutzer-ID des vormerkenden Benutzers, ansonsten NULL

4.2.3 Tabelle ausleihe

In dieser Tabelle wird die Ausleihe verbucht. Falls ein Nutzer ein Buch ausleiht, wird hier neben dem Rückgabedatum sowohl seine Benutzer-ID als auch die Signatur des ausgeliehenen Buches gespeichert. Diese beiden Attribute bilden den zusammengesetzten Primärschlüssel dieser Tabelle. nutzerid verweist als Fremdschlüssel auf bibliotheksNutzer(nutzerid), signatur verweist als Fremdschlüssel auf bibliotheksbuch(signatur).

Attribut	Datentyp	Constraints	Bemerkung
nutzerid	CHAR(5)	PRIMARY KEY FOREIGN KEY REFERENCES bibliotheksNutzer (nutzerid)	enthält die eindeutige BenutzerID; bildet mit der signatur den zusammengesetzten Primärschlüssel; zeigt als Fremdschlüssel auf bibliotheksNutzer(nutzerid)
signatur	CHAR(8)	PRIMARY KEY FOREIGN KEY (signatur) REFERENCES bibliotheksBuch(s ignatur)	die eindeutige Signatur eines Buches; bildet mit der nutzerid den zusammengesetzten Primärschlüssel; zeigt als Fremdschlüssel auf bibliotheksBuch(signatur)
rückgabeDatum	DATE		gibt den Tag an, an dem die Leihfrist endet

5. Beschreibung der Klassen

5.1. Program.cs

void Main()

Einstiegspunkt des Programms, Aufruf der Forms.

5.2. Nutzer.cs

public Nutzer

Konstruktor der Klasse Nutzer, Initialisiert das Objekt.

public string Nutzer_ID

Nutzer_ID setzen und bekommen.

public DateTime AblaufDatum

AblaufDatum setzen und bekommen.

public bool MahnungVorhanden

MahnungVorhanden setzen und bekommen.

public string NutzerDetails

Eine Zeichenkette zurückgeben bestehend aus Nutzer_ID, AblaufDatum und MahnungVorhanden

5.3. Form1.Designer.cs

Form1 wird mit ihren Objekten initialisiert.

5.4. Form1.cs

Jegliche Button funktionen werden implementiert.

5.5. DBConnection.cs

public DBConnection()

Verbindung wird zur Datenbank hergestellt.

5.6. DataBConnection.cs

Datenbank Verbindung wird kontrolliert. Jegliche SQL Abfragen werden in Methoden implementiert.

5.7. Buch.cs

public Buch

Konstruktor zum Initialisieren des Objektes.

public string Signatur_ID

Signatur_ID setzen und bekommen.

public string Autor

Autor setzen und bekommen.

public string Titel

Titel setzen und bekommen.

public bool ExemplarTyp

ExemplarTyp setzen und bekommen.

public bool Vorgemerkt

Vorgemerkt setzen und bekommen.

public string BuchDetails

Zeichenkette BuchDetails zurückgeben die enthält Autor, Titel, ExemplarTyp und Vorgemerkt.

5.8. Ausleihe.cs

public Ausleihe

Konstruktor der Objekte der Klasse Ausleihe initialisiert.

public string Nutzer_ID

Nutzer_ID setzen und bekommen.

public string Signatur_ID

Signatur_ID setzen und bekommen.

public DateTime RückgabeDatum

RückgabeDatum setzen und bekommen.

5.9. Form2.Designer

Die Form2 wird mit sämtlichen Elementen initialisiert.

5.10. Form2

Ein Button "Buchausleihe verwalten" enthält eine Funktion die zur Form1 weiterleitet.