



HTW Chur
Hochschule für Technik und Wirtschaft

Fachhochschule Ostschweiz
University of Applied Sciences

Entwurfsdokument

Projekt: Bücherverwaltungssystem BVS

Version 1.2



HTW Chur
Bücherverwaltungssystem BVS

Schöb Romana

Schneider Björn

Stadelmann Luca

Krähenbühl Daniel

Büchel Simon

Änderungsgeschichte

<u>Datum</u>	<u>Version</u>	<u>Autor</u>	<u>Beschreibung</u>
18.04.2008	1.0	SL	Entwurfsdokument erster Entwurf
01.05.2008	1.1	SL KR	Ergänzung Revision Entwurfsdokument
02.05.2008	1.2	SR, BS, RS	Ergänzung Revision Entwurfsdokument

Autoren sind mit ihren Kurzzeichen geführt, welche sich wie folgt ergeben:

SR Schöb Romana
BS Büchel Simon
SB Schneider Björn
KR Krähenbühl Daniel
SL Stadelmann Luca

Inhaltsverzeichnis

1	EINFÜHRUNG.....	5
1.1	Überblick	5
1.2	Inhalt	5
1.3	Referenzen	5
1.4	Begriffe und Abkürzungen	5
2	HINWEISE ZUM DESIGN	6
2.1	Annahme	6
2.2	Einschränkungen.....	6
2.3	Systemumgebung.....	6
2.4	Designmethoden.....	6
2.5	Risiken	6
3	ARCHITEKTUR.....	7
3.1	Überblick	7
3.2	Benutzer	8
3.3	Präsentationsschicht	8
3.4	Anwendungsschicht.....	8
3.5	Struktur Diagramm	9
3.6	Datei- und Ordnerstruktur.....	10
3.7	Klassendiagramm.....	11
3.8	Klassenbeschreibung	12
4	DESIGN.....	15
4.1	Datenbank	15
4.2	Benutzerschnittstelle	18
A	REFERENZEN	19

Abbildungsverzeichnis

Abbildung 1: 2- Schichtenarchitektur Model	7
Abbildung 2: Struktur Diagramm	9
Abbildung 3: Klassendiagramm.....	11
Abbildung 4: Relationales Datenbankmodell.....	15
Abbildung 5: Benutzerschnittstelle Beschreibung	18
Abbildung 6: Benutzerschnittstelle Webansicht	18

Tabellenverzeichnis

Tabelle 1: Begriffe	5
Tabelle 2: user	16
Tabelle 3: book	16
Tabelle 4: statistics	17

1 Einführung

1.1 Überblick

Dieses Dokument definiert das Design des Bücherverwaltungssystems (BVS). Das BVS wird in Form einer Web Applikation programmiert. Mit dieser Software wird es möglich sein, nicht mehr benötigte Bücher von Studierenden zu verkaufen und zugleich Bücher, die für die Vorlesungen nötig sind, zu kaufen.

Das Einlesen der Bücher mit einem Hand-Scanner ist nicht vorgesehen.

Die folgende Dokumentation richtet sich an die Projektmitarbeiter des BVS, sowie an die Betreuer der Studienarbeit.

1.2 Inhalt

Das Entwurfsdokument des BVS behandelt folgende Themen:

- Allgemeine Hinweise zum Design
- Architektur mit der Beschreibung der grundlegenden Komponenten und deren Zusammenspiel innerhalb des BVS
- Design mit einer ausführlichen Beschreibung der Architektur und der einzelnen Module
- Design der Benutzerschnittstellen

1.3 Referenzen

Siehe A Referenzen auf Seite 19 dieses Dokumentes.

1.4 Begriffe und Abkürzungen

BVS	Bücherverwaltungssystem
D & I	Design & Implementation
A & Q	Anforderungen & Qualität
W3-Konsortium	World Wide Web Konsortium
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
SQL	Structured Query Language

Tabelle 1: Begriffe

2 Hinweise zum Design

Die hier aufgeführten Annahmen, Einschränkungen, die Systemumgebung und Risiken sind bereits im Dokument „Anforderungsdokument_Buecherverwaltung.doc“ ausführlich beschrieben. Die wichtigsten Punkte sind mit einem Verweis auf die Anforderungsdokumentation gekennzeichnet.

2.1 Annahme

Siehe Anforderungsdokument Punkt 2.4.

2.2 Einschränkungen

Siehe Anforderungsdokument Punkt 2.3.

2.3 Systemumgebung

Siehe Anforderungsdokument Punkt 2.2.2 und 2.2.3.

2.4 Designmethoden

Bei der Entwicklung wird die in den Modulen Informatik I und II vermittelte Theorie angewendet. Als Kodierstandards beim Einsatz von HTML und CSS werden diejenigen des W3- Konsortiums verwendet. Für die Implementierung der dynamischen Webseiten wird die Script-Sprache PHP eingesetzt. Die objektbasierte Scriptsprache Javascript wird zusätzlich zu PHP für die Überprüfung der Formularfelder implementiert. Dabei richten wir uns an die im Dokument „Projektplan_Buecherverwaltung_1.2.doc“ aufgeführten Standards. Die Abfragesprache für das BVS-Portal ist MySQL Version 4.1.

2.5 Risiken

Sollte der Amazon Webservice bei der Eingabe der ISBN Nummer nicht mehr die gewünschten Rückgabewerte liefern, müsste der User die Eingabe der Nummer selber tätigen. Die grundsätzliche Funktion ist jedoch nie gefährdet, da die Funktion nur als eine Erleichterung ist für den User gedacht ist.

3 Architektur

3.1 Überblick

Das Bücherverwaltungssystem lässt sich, wie unten abgebildet, mit einem 2- Schichtenarchitektur-Model beschreiben. Dieses Model besteht aus einer Präsentationsschicht (GUI) und einer Anwendungsschicht (Anwendungslogik, System).

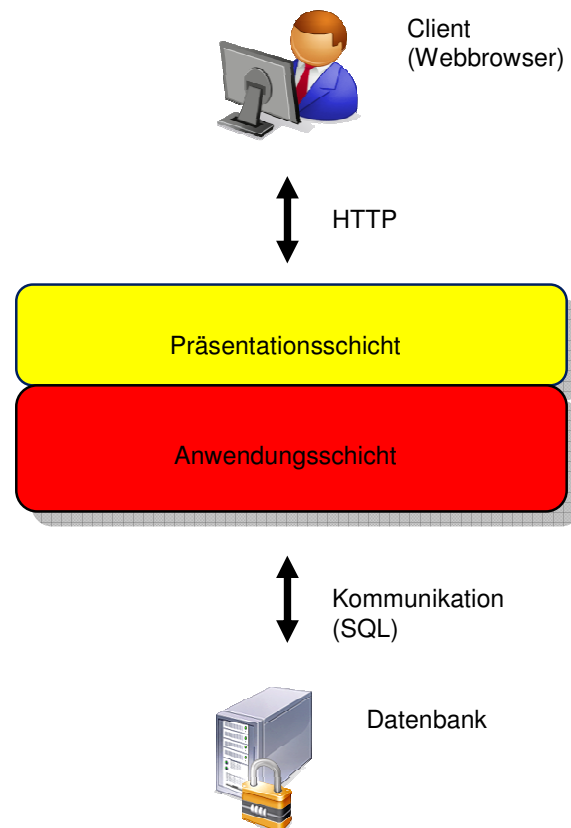


Abbildung 1: 2- Schichtenarchitektur Model

3.2 Benutzer

Ein Benutzer kann mit Hilfe eines Webbrowsers (Vgl. Kapitel 2.3) alle Eingaben vornehmen. Die Übertragung wird mit HTTP realisiert.

3.3 Präsentationsschicht

Zur Präsentationsschicht gehören Klassen und Code die HTML-Quellcode generieren, welcher beim Benutzer im Browser als grafische Oberfläche dargestellt wird. Im Browser des Benutzers wird Javascript eingesetzt, um vor dem Löschen eine Bestätigung abzufragen. Die Validierung der Eingaben des Benutzers erfolgt in der Anwendungsschicht.

3.4 Anwendungsschicht

Für die Ablaufsteuerung der Anwendung ist die Anwendungsschicht zuständig. Diese sorgt dafür, dass alle Informationen der Anwendung in der gewünschten Weise verarbeitet, transformiert oder aufbereitet werden. Darüber hinaus ist die Anwendungsschicht aber auch für die Konsistenz der Daten verantwortlich. Zum Beispiel werden alle Eingaben eines Benutzers auf Korrektheit geprüft. Um zu kontrollieren, wann die Daten eines Buchverkaufes abgelaufen, respektive überschritten sind, wird der Dienst von Cronjob benutzt, der auf dem Server läuft. Dieser kontrolliert täglich alle Daten und versendet wenn nötig mit Hilfe von PHP, die Mails an die Benutzer.

3.5 Struktur Diagramm

Das Struktur Diagramm gibt einen Überblick über die verwendeten Komponenten und die Gruppierung innerhalb der Ordner und Schichten des Bücherverwaltungssystems. Dabei wird zwischen Präsentationsschicht und Anwendungsschicht unterschieden.

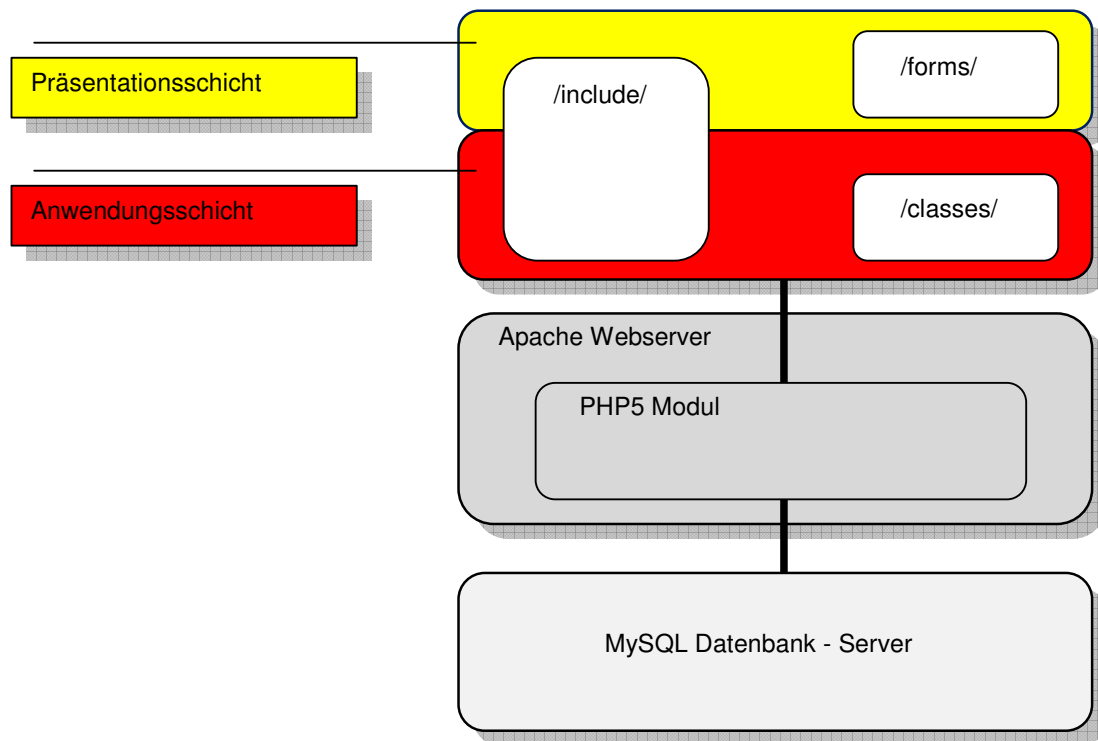


Abbildung 2: Struktur Diagramm

3.6 Datei- und Ordnerstruktur

Quelldateien werden in folgender Ordnerstruktur verwaltet:

/	Der Hauptordner enthält alle Unterordner des Bücherverwaltungsystems
/include/	Enthält die Klassen, die Initialisierungsdatei und die Logdatei <ul style="list-style-type: none">- book.class.php- user.class.php- sql.class.php- session.class.php- init.inc.php- statistics.log.php
/Web/	
/images/	Enthält die Bilddateien (.jpg) für das Design <ul style="list-style-type: none">- logo.jpg- book.jpg- footer_bg.jpg- navigation_icon.jpg- usw.
/include/	Enthält die für das Layout benötigten Dateien <ul style="list-style-type: none">- header.php- navigation.php- footer.php
/sites/	Enthält die einzelnen Seiten wie z.B. <ul style="list-style-type: none">- index.php- login.php- usw.
/style/	Enthält die CSS-Datei für das Layout. <ul style="list-style-type: none">- format.css

3.7 Klassendiagramm

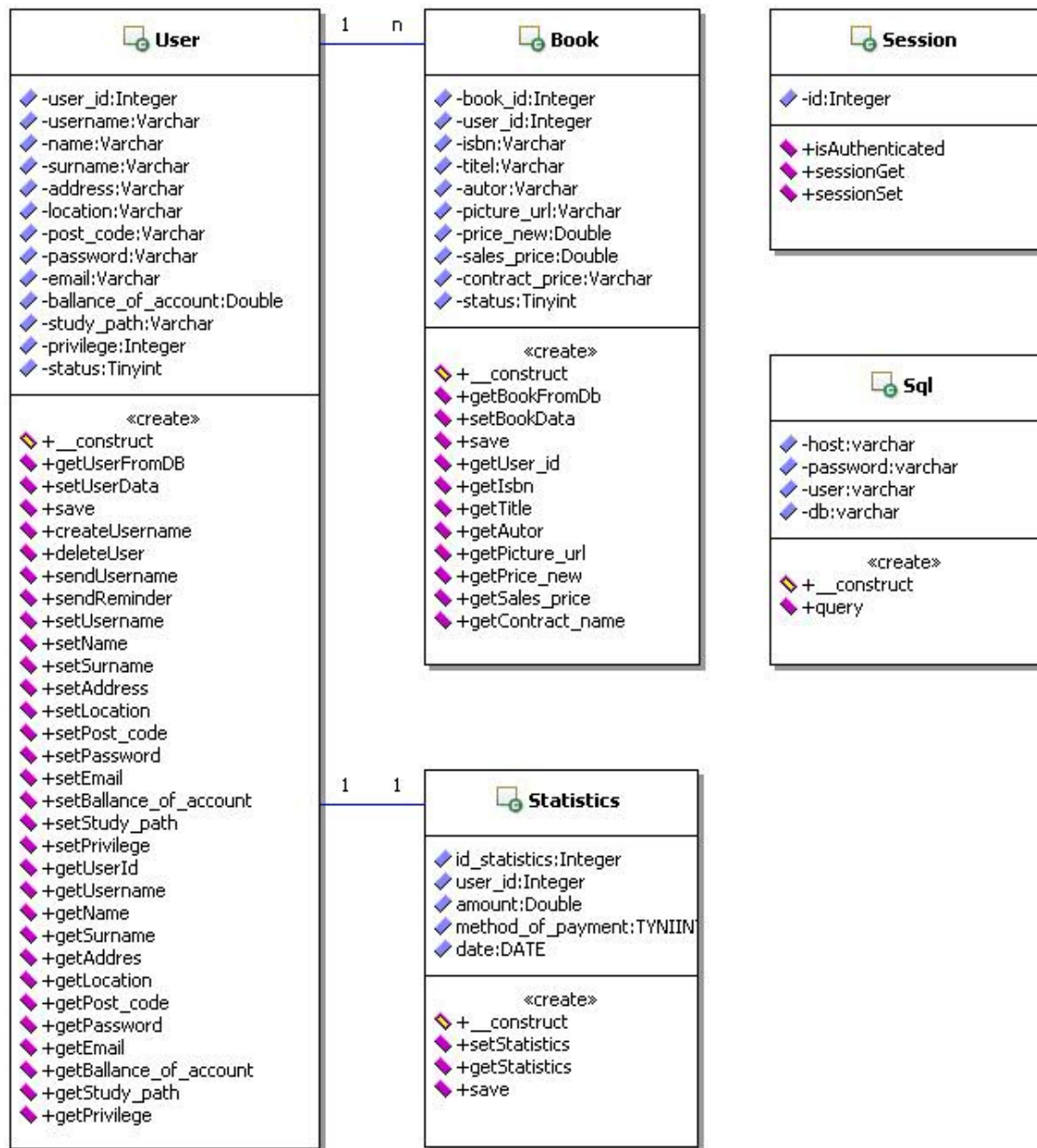


Abbildung 3: Klassendiagramm

3.8 Klassenbeschreibung

3.8.1 Klasse user

Die Klasse ist für die Verwaltung des Benutzers notwendig. Mit Hilfe der unten aufgeführten Methoden ist eine Benutzerverwaltung möglich.

Funktion getUserFromDB

Speichert die Daten aus der Datenbank in den Variablen der Userklasse

Funktion setUserData

Speichert die übergebenen Buchdaten in den Variablen der Userklasse

Funktion save

Speichert die Variablen der Userklasse in der Datenbank

Funktion createUsername

Erstellt aus den ersten drei Buchstaben des Vornamens und den ersten 3 Buchstaben des Nachnamens den Username

Funktion deleteUser

Setzt Status auf „0“ was bedeutet, dass der User gelöscht ist

Funktion sendUsername

Verschickt den vorher erstellten Username per Email an den Benutzer

Funktion sendReminder

Verschickt ein Erinnerungsmail, dass das Buch in der nächsten Zeit ablaufen wird

Funktion setUsername

Speichert Username in der Datenbank

Funktion setName

Speichert den Namen in der Datenbank

Funktion setSurname

Speichert den Vornamen in der Datenbank

Funktion setAddress

Speichert die Adresse in der Datenbank

Funktion setLocation

Speichert den Ort in der Datenbank

Funktion setPost_code

Speichert die PLZ in der Datenbank

Funktion setPassword

Speichert das Passwort in der Datenbank

Funktion setEmail

Speichert die Email-Adresse in der Datenbank

Funktion setBallance_of_account

Speichert den Kontostand in der Datenbank

Funktion setStudypath

Speichert Studiengang in der Datenbank

Funktion setPrivilege

Speichert den Status in der Datenbank (Benutzer, Bibliothekarin, Admin)

Funktion getUserid

Speicher die User ID aus der Datenbank in die entsprechende lokale Variable

Funktion getUsername

Speicher den Benutzernamen aus der Datenbank in die entsprechende lokale Variable

Funktion getName

Speicher den Namen aus der Datenbank in die entsprechende lokale Variable

Funktion getSurname

Speicher den Vornamen aus der Datenbank in die entsprechende lokale Variable

Funktion getAddress

Speicher die Adresse aus der Datenbank in die entsprechende lokale Variable

Funktion getLocation

Speicher den Ort aus der Datenbank in die entsprechende lokale Variable

Funktion getPost_code

Speicher die PLZ aus der Datenbank in die entsprechende lokale Variable

Funktion getPassword

Speicher das Passwort aus der Datenbank in die entsprechende lokale Variable

Funktion getEmail

Speicher die Email-Adresse aus der Datenbank in die entsprechende lokale Variable

Funktion getBalance_of_account

Speicher den Kontostand aus der Datenbank in die entsprechende lokale Variable

Funktion getStudy_path

Speicher die Studienrichtung aus der Datenbank in die entsprechende lokale Variable

Funktion getPrivilege

Speicher den Status aus der Datenbank in die entsprechende lokale Variable

3.8.2 Klasse book

Die Klasse book ist für die Verwaltung der Bücher zuständig. Sei es um ein Buch zu erstellen, bearbeiten oder zu löschen, die unten aufgeführten Methoden ermöglichen dies.

Funktion getBookFromdb

Speichert die Daten aus der Datenbank in den Variablen der Buchklasse

Funktion setBookData

Speichert die übergebenen Buchdaten in den Variablen der Buchklasse

Funktion save

Speichert die Variablen der Buchklasse in der Datenbank

Funktion getUser_id

Speicher die User ID aus der Datenbank in die entsprechende lokale Variable

Funktion getISBN

Speicher die ISBN aus der Datenbank in die entsprechende lokale Variable

Funktion getTitle

Speicher den Titel aus der Datenbank in die entsprechende lokale Variable

Funktion getAutor

Speicher den Autor aus der Datenbank in die entsprechende lokale Variable

Funktion getPicture_url

Speicher die Bild URL aus der Datenbank in die entsprechende lokale Variable

Funktion getPrice_new

Speicher den Neupreis aus der Datenbank in die entsprechende lokale Variable

Funktion getSales_price

Speicher den Verkaufspreis aus der Datenbank in die entsprechende lokale Variable

Funktion getContract_name

Speicher den Verlagsnamen aus der Datenbank in die entsprechende lokale Variable

3.8.3 Klasse session

Die Klasse session wird für die Sessions-Verwaltung verwendet. Sie beinhaltet folgende Methoden:

Funktion isAuthenticatet

Überprüft ob der Benutzer angemeldet ist. Wenn ja, wird er zum internen Account weitergeleitet, wenn nein auf die Login Seite verwiesen

Funktion sessionGet

Holt die aktuelle Session aus der Funktion session_start()

Funktion sessionSet

Ruft die Funktion session_start() auf, um einen neue Session zu starten

4 Design

4.1 Datenbank

4.1.1 Relationales Datenbankmodell

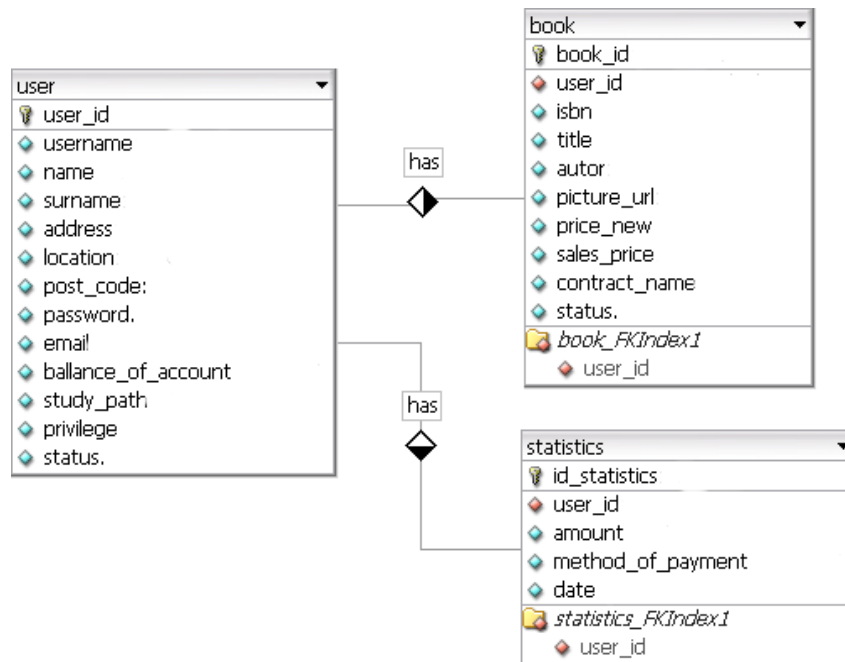




Abbildung 4: Relationales Datenbankmodell

Erläuterung:

-  steht für Primärschlüssel
-  steht für Fremdschlüssel

4.1.2 Datenbankschema

User

Spalten-Name	Typ	Beschreibung
user_id	int(11) NOT NULL auto_increment	Primary Key
username	varchar(30) NOT NULL	Benutzername
name	varchar(30) NOT NULL	Name
surname	varchar(30) NOT NULL	Vorname
address	varchar(30) NOT NULL	Adresse
location	varchar(30) NOT NULL	Ort
post_code	int(11) NOT NULL	PLZ
password	varchar(30) NOT NULL	Passwort
email	varchar(30) NOT NULL	Email
ballance_of_account	double NOT NULL	Kontostand
study_path	varchar(30) NOT NULL	Studienrichtung
privilege	int(11) NOT NULL	Berechtigung für das Einloggen
status	tinyint(4) NOT NULL	0 = gelöscht, 1= nicht gelöscht

Tabelle 2: user

Book

Spalten-Name	Typ	Beschreibung
book_id	int(11) NOT NULL auto_increment	Primary Key
user_id	int(11) NOT NULL	Benutzer ID
isbn	varchar(30) NOT NULL	ISBN-Nummer
title	varchar(30) NOT NULL	Titel des Buches
autor	varchar(30) NOT NULL	Autor des Buches
picture_url	varchar(255) NOT NULL	Bild-URL
price_new	double NOT NULL	Neupreis
sales_price	double NOT NULL	Verkaufspreis
contract_name	varchar(30) NOT NULL	Verlagsname
status	tinyint(4) NOT NULL	0 = gelöscht, 1= nicht gelöscht

Tabelle 3: book

Statistics

Spalten-Name	Typ	Beschreibung
id_statistics	int(11) NOT NULL auto_increment	Primary Key
user_id	int(11) NOT NULL	Benutzer ID
Amount	double NOT NULL	Betrag
method_of_payment	tinyint(4) NOT NULL	Art der Zahlung
Date	Datetime	Datum

Tabelle 4: statistics

4.2 Benutzerschnittstelle

Die Benutzerschnittstelle ist an das Design von my-fhhtwchur.ch angelehnt, damit ein schnelles Zurechtfinden gewährleistet sein sollte.

Die Gestaltung wird dabei mit einer Tabelle umgesetzt. Die oberste Zeile besteht aus einer Spalte, die jeweils mit einem Bild gefüllt ist, das links zentriert wird. Rechts erscheint das Feld für den Login.

Auf der linken Seite nach unten ist die Navigationsliste, welche sich dem entsprechenden Benutzer anpasst. Am Ende der Seite wird noch ein footer angefügt.

Breite: 724px Höhe: 115px Inhalt: Header Datei – Name: header.php	
Breite: 184px Höhe: optional Inhalt: Navigation Datei – Name: navigation.php	Breite: 540px Höhe: optional Inhalt: Content
Breite: 724px Höhe: 48px Inhalt: Footer Datei – Name: footer.php	

Abbildung 5: Benutzerschnittstelle Beschreibung



Abbildung 6: Benutzerschnittstelle Webansicht

A Referenzen