Pflichtenheft für das Webprojekt „Bibliotheksverwaltung“

# Zielbestimmung

## Musskriterien

* Büchersuche über Volltext
* Login für Bibliothekare
* Ausleihe und Rückgabe, wobei der Zugriff nur für Bibliothekare gestattet ist
* Buchanlage bei Neuanschaffung (ebenfalls nur für Bibliothekare zugänglich)
* Buchdaten ändern (nur Bibliothekare)
* Buch ausscheiden (nur Bibliothekare)

## Wunschkriterien

* Büchersuche über verschiedene Kategorien
* Büchersuche über verschiedene Verlage
* Responsive Web-Design

# Produkteinsatz

## Anwendungsbereiche

Grundsätzlich soll die Web-App die Verwaltung von Büchern vereinfachen. Nach erfolgreichem Login kann auf Daten über Bücher und Ausleihen, welche in einer Datenbank gespeichert werden, zugegriffen werden.

## Zielgruppen

Einerseits ist die App auf Bibliothekare ausgerichtet, die einfach über den Browser Bücherausleihen verwalten können. Andererseits können reguläre Benutzer nach Büchern suchen und filtern.

## Betriebsbedingungen

Im Rahmen des Projekts wird die Anwendung ausschließlich lokal auf einem PC oder Notebook laufen. Aufgrund der Aufgabenstellung ist es nicht unbedingt notwendig und hinsichtlich der zu benützenden Technologien aufwendig, die App zu hosten.

# Produktumgebung

## Software

* Betriebssystem: Linux-Server
* Nginx
* Datenbanksystem: MySQL +
* NodeJS
* **Frontend:** Vue.js mit InertiaJS zur Integration mit dem Laravel-Backend
* **Backend:** Laravel (PHP Framework) mit InertiaJS für serverseitiges Rendering und Routing
* Mindestens PHP 8.1

## Hardware

* Schulserver
* Genügend Speicherplatz (Festplattenspeicher), um die Datenbank betreiben zu können

## Produktschnittstellen

* InertiaJS regelt die Schnittstellen, die nötig sind, um das Laravel Backend mit dem Vue.js Frontend zu verknüpfen. Mithilfe dieser Schnittstelle kann das Frontend mit modernen JavaScript Frameworks programmiert werden, während gleichzeitig das Backend traditionell gehalten wird.

# Produktfunktionen

## Büchersuche

Diese Funktion kann öffentlich von jedem benutzt werden. Bücher sollen über folgende Arten gesucht und gefiltert werden können:

* Über Texteingabe (Titel und Beschreibung)
* Über Kategorien (Wunschkriterium)
* Über Verlage (Wundschkriterium)

## Login

Um den Zugriff auf Bibliothekare zu beschränken, muss man sich einloggen, um nachfolgende Funktionen benutzen zu können. Im Großen und Ganzen können nach dem Login Bücher verwaltet werden können.

## Ausleihe und Rückgabe

In der Datenbank werden Daten über die Ausleihe und Rückgabe eines Buches festgehalten. Bibliothekare können in der App eintragen, wenn eine Person ein Buch ausleiht, bzw. zurückgibt und in welchem Zeitraum er dieses Buch behalten darf.

## Buchanlage und Neuanschaffung

Zudem haben Bibliothekare die Möglichkeit, neue Bücher in das System einzutragen. Dies wird über eine Zeile in einer Tabelle in der Datenbank realisiert.

## Buchdaten ändern

Wenn ein Buch eingetragen ist, kann ein Bibliothekar dieses bearbeiten.

## Buch ausscheiden

Sollte ein Buch nicht mehr vorhanden sein in der Bibliothek, dann kann ein Bibliothekar dieses Buch aus dem System / der Datenbank löschen.

# Produktdaten

## Bücherdaten

* ISBN
* Titel
* Beschreibung
* Verlag
* Anschaffungspreis
* Kategorie

## Ausleihdaten

* Ausleiher
* Bibliothekar
* Buch
* Ausleihdatum
* Rückgabedatum

## Bibliothekar

* Vorname
* Nachname
* E-Mail
* Benutzername

## ER-Diagramm

Ein Bild, das Kinderkunst, Diagramm, Design enthält.

Automatisch generierte Beschreibung

## Beziehungen

### 1 zu 1 Beziehung zwischen Bücher und Ausleihdaten

Ein Buch kann jeweils nur mit einer Ausleihe in Beziehung stehen. Wenn ein Buch ausgeliehen wird, ist es so lange nicht mehr verfügbar, bis der Ausleiher es wieder zurückgebracht hat. Dementsprechend ist ein Buch mit einer Ausleihe in Relation, nicht mit mehreren. Umgekehrt gilt das gleiche.

### 1 zu n Beziehung zwischen Bibliothekar und Ausleihe

Ein Bibliothekar kann mehrere Bücher zu Ausleihe freigeben.

# Benutzungsoberfläche

Im Folgenden werden Skizzen angegeben, wie die Benutzeroberfläche grob gestaltet werden könnte:

## Bildersuche



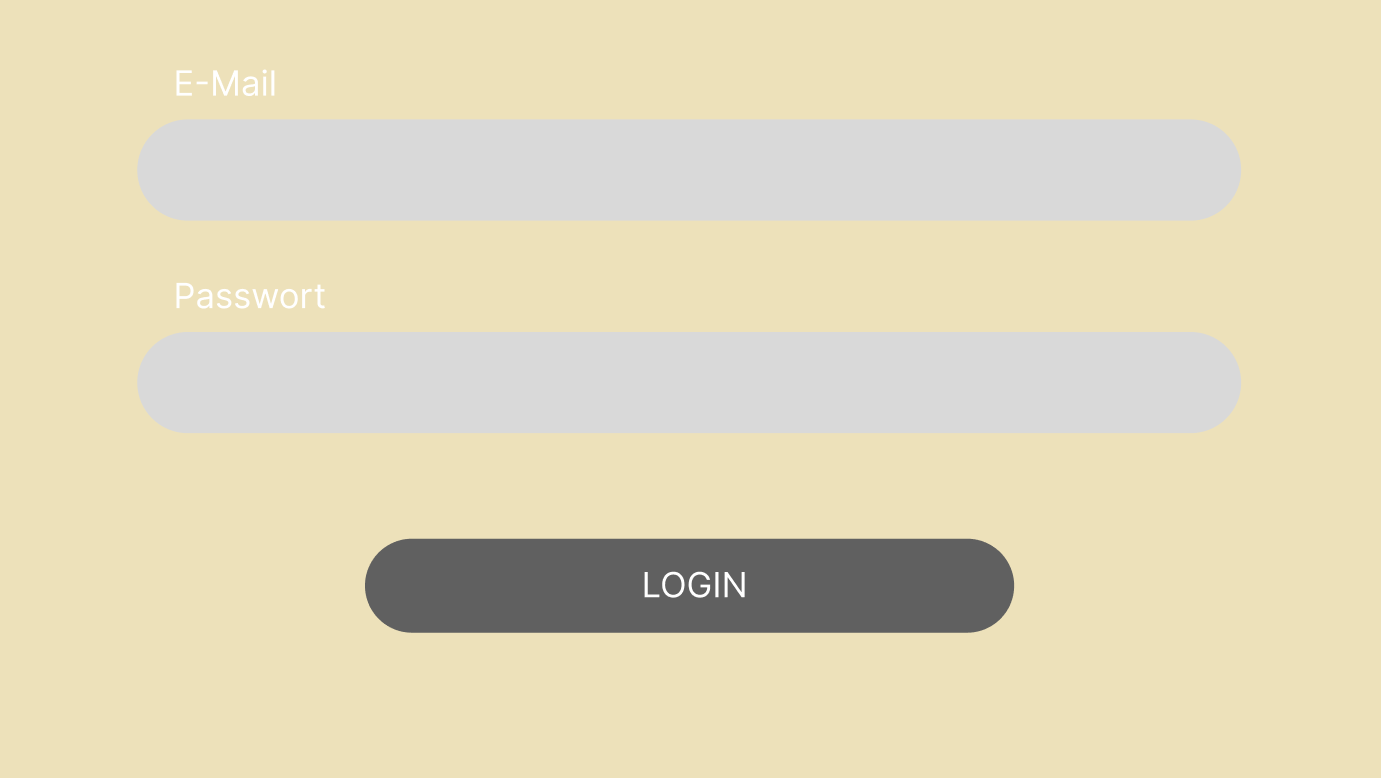
## Bücherverwaltung



## Bücherverleihung



## Login



# Qualitätsanforderungen

## Design

Vorausgesetzt wird, dass das Design für sich selbst spricht und nichts erklärt werden muss. Aus der Sicht des Benutzers müssen alle Funktionen ersichtlich sein. Wie in den Wunschkriterien angegeben, wäre es wünschenswert, dass die Anwendung diese Kriterien auch bei der Verwendung am Smartphone oder am Tablet erfüllt.

## Single Page Application (SPA)

Die Verwendung von Inertia.js und Vue.js ermöglicht das Erstellen einer sogenannten Single Page Application. Das bedeutet, dass beim Wechseln von Seiten (z.B. beim Klick auf einen Link zu einer Unterseite) das Browser-Fenster nicht neugeladen werden muss. Das Vue.js Frontend lädt nur die Inhalte herunter, die es benötigt, anstatt eine komplette HTML Datei vom Server zu empfangen. Dementsprechend werden die Inhalte am Client visualisiert und gerendered. Der Servercode muss nu die Inhalte verwalten und die Daten an die Clients weitergeben. Somit wird die Aufgabe des Servers von der des Clients getrennt, was den Code eventuell auch übersichtlicher gestaltet.

# Globale Testszenarien und Testfälle

## Test Büchersuche

Um zu überprüfen, ob die Büchersuche funktioniert, können mehrere Suchbegriffe in die Suche eingetragen werden. Wenn entsprechende Bücher ausgegeben werden, ist die Suche erfolgreich.

## Test Login / Authentifikation

Alle Seiten außer der Login- und Büchersuchseite dürfen nur für authentifizierte Bibliothekare erreichbar sein. Um dies zu testen, kann die Website einmal als Gast, also ohne Anmeldedaten eines Bibliothekars aufgerufen werden. Dann darf man nur die Login Seite und die Büchersuchseite sehen. Werden aber beim Login vorliegende Anmeldedaten eines registrierten Bibliothekars eingegeben (E-Mail und Passwort), dann kann auf alle Funktionen der Webapp zugegriffen werden.

## Test Ausleihe und Rückgabe

Als eingeloggter Bibliothekar kann nun überprüft werden, ob die Ausleihe und Rückgabe von Büchern wie vorgesehen funktioniert, indem ein Buch testweise ausgeliehen und wieder zurückgegeben wird. Wenn die Änderungen sowohl im Frontend als auch in der Datenbank direkt angezeigt werden, ist die Funktion vollständig.

## Test Buchanlage und Neuanschaffung

Dasselbe Verfahren wird mit der Buchanlage und Neuanschaffung angewandt. Mit einem Datenbanktool wie TablePlus können die Inhalte von Tabellen angezeigt werden.

## Test Buchdatenänderungen

Wenn bei einer Buchdatenänderung die Daten direkt in der Datenbank geändert werden, ist die Funktion vollständig.

## Test Buchausscheidung / Löschen von Büchern

Wenn beim Löschen von einem Buch in der Webapp die entsprechende Zeile aus der Datenbanktabelle gelöscht wird / verschwindet, ist die Funktion vollständig.

# Entwicklungsumgebung

## Software

* Betriebssystem: Windows 11
* Datenbanksystem: MySQL + DBngine + TablePlus (zur Visualisierung der Tabellen)
* NodeJS Runtime
* NPM (Node Package Manager)
* Vue3: Javascript Framework, wird mit NPM installiert
* PHP (mindestens 8.1)
* Composer
* Laravel: PHP Framework, wird mit Composer installiert
* Laravel Herd: Umgebung, speziell für Laravel ausgerichtet

## Hardware

* Privater Laptop / Privater PC / Schul-PC
* Ausreichend Speicherplatz (Festplattenspeicher), um die Datenbank betreiben zu können
* Mindestanforderungen für Windows 11, sowie PHP, NodeJS und MySQL

## Entwicklungsschnittstellen

* Wie auch in der Produktumgebung wird das Frontend mit dem Backend mithilfe von InertiaJS kommunizieren. Das Projekt wird mit Vite gebündelt, das heißt, die .vue Dateien werden durch Vite in JS Dateien kompiliert.