

# Audit

|  |  |
| --- | --- |
| Mattias Zurbuchen |  |
| Julia Ebner |  |
|  |  |
| Firma: | Die Schweizerische Post |
|  |  |
| Datum: 20.03.2024 |  |

Inhaltsverzeichnis

[Audit 1](#_Toc162515223)

[TEIL 1 4](#_Toc162515224)

[Abstract 5](#_Toc162515225)

[Anforderungsanalyse 6](#_Toc162515226)

[Zielgruppe 6](#_Toc162515227)

[Pflichtenheft 6](#_Toc162515228)

[Lieferumfang 6](#_Toc162515229)

[Planung 7](#_Toc162515230)

[TEIL 2 8](#_Toc162515231)

[Sitemap 9](#_Toc162515232)

[Index Page (/) 9](#_Toc162515233)

[Audiobuch Page (/audibuecher) 9](#_Toc162515234)

[Audiobuch Erstellen Page (/audibuecher/create) 9](#_Toc162515235)

[Audiobuch Bearbeiten Page (audibuecher/edit/[i]) 9](#_Toc162515236)

[Users Page (/users) 9](#_Toc162515237)

[User erstellen Page (users/create) 9](#_Toc162515238)

[User Details Page (users/[i]) 10](#_Toc162515239)

[User bearbeiten Page (users/edit/[i]) 10](#_Toc162515240)

[Login (/login) 10](#_Toc162515241)

[Sign-Up (/sign-up) 10](#_Toc162515242)

[Impressum (/impressum) 10](#_Toc162515243)

[Layouts 11](#_Toc162515244)

[Farbkonzept 12](#_Toc162515245)

[TEIL 3 13](#_Toc162515246)

[Technische Dokumentation 14](#_Toc162515247)

[Testkonzept 14](#_Toc162515248)

[Testumgebung 14](#_Toc162515249)

[Testfälle 14](#_Toc162515250)

[Testprotokoll 17](#_Toc162515251)

[W3C Validation 19](#_Toc162515252)

[Fazit 19](#_Toc162515253)

# TEIL 1

## Abstract

Wir schreiben das Frontend, das ermöglichen soll, CRUD-Operationen auf einer Datenbank durchzuführen, welche eine Hörbuch-Bibliothek repräsentieren soll. Es existieren die Entitäten user, audiobuch, genre und ausleihe. Wir planen, CRUD für user und audiobuch zu erfassen. Weiter können sich die Besucher der Webseite registrieren und anmelden. Wenn ein Benutzer angemeldet ist, bekommt dieser administrative Rechte, anderenfalls nur Read auf den Audiobüchern.

## Anforderungsanalyse

### Zielgruppe

Das Produkt wird für den Bibliothekar unserer modellierten Bibliothek erstellt.

### Pflichtenheft

###### Funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| ID | Beschreibung | Art |
| FA-01 | Die Webseite soll dem Benutzer die Möglichkeit geben, sich mit Hilfe seines Benutzernamen und seines Passworts einzuloggen. | muss |
| FA-02 | Die Webseite soll ermöglichen, sich zu Registrieren. | muss |
| FA-03 | CRUD für Audiobücher | muss |
| FA-04 | CRUD für Users | muss |

Tabelle 1 Funktionale Anforderungen

###### Nichtfunktonale Anforderungen

|  |  |  |
| --- | --- | --- |
| ID | Beschreibung | Art |
| NFA-01 | Jede einzelne Seite des Projekts soll die HTML W3 Validation Error-Frei durchlaufen. | muss |
| NFA-02 | Einheitliches Styling | kann |
| NFA-03 | Seite ist intuitiv navigierbar | kann |
| NFA-04 | Pre-Rendering wird eingesetzt für users-Page und audiobuecher-Page | muss |

Tabelle 2 Nicht Funktionale Anforderungen

### Lieferumfang

Der Lieferumfang und die Vorgaben über Speicherort werden dem mitgegebenen Lastenheft entnommen.

## Planung

##### Siehe Excel

# TEIL 2

## Sitemap

Unsere Sitemap sieht folgendermassen aus:

Ein Bild, das Reihe, Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung 1 Sitemap

### Index Page (/)

Hier sind alle Audiobücher aufgelistet mit ihren Titeln und Autor, darunter kann man einen Mehr Details Link sehen. Zuunterst auf der Seite gibt es noch einen Erstellen Link um ein neues Audiobuch zu erstellen.

### Audiobuch Page (/audibuecher)

Klickt man auf den vorher erwähnten Mehr Details Link auf der Index Page gelangt man zu der Audiobuch Page. Dies zeigt nun ein Audiobuch mit seinen ganzen Details:

##### Titel, Autor, Länge, Genre, Erscheinungsjahr

Darunter gibt es noch drei Links mit den Titeln: Zurück, Bearbeiten und Löschen.

### Audiobuch Erstellen Page (/audibuecher/create)

Klickt man auf den vorher erwähnten Erstellen Link gelangt man zu der Audiobuch Erstellen Page. Das Formular weist folgende Felder auf:

##### Titel, Autor, Länge, Genre, Erscheinungsjahr

Darunter gibt es noch drei Links mit den Titeln: Erstellen, Zurück und Löschen

### Audiobuch Bearbeiten Page (audibuecher/edit/[i])

Klickt man auf den erwähnten Bearbeiten Link gelangt man zu der Audiobuch Bearbeiten Page. Dies zeigt nun ein Formular um das Audiobuch zu bearbeiten. Mit allen Feldern:

Titel, Autor, Länge, Genre, Erscheinungsjahr

Darunter gibt es einen Link mit dem Titel Zurück.

### Users Page (/users)

Die Startseite für die Users. Der Inhalt besteht aus allen vorhandenen Usern, es kann auf die Subpages navigiert werden. Erreichbar ist die Seite durch den Link in der Navbar.

### User erstellen Page (users/create)

Auf dieser Seite können neue User erstellt werden und eigentlich deckt sich die Funktionalität mit der /sign-up-Page. Der Inhalt ist ein Formular, über welches die Benutzerdaten eingegeben werden können.

### User Details Page (users/[i])

Die detaillierte Ansicht eines bestimmten Users, ist erreichbar über die /users Seite. Hier werden alle Benutzerdaten, ausser Passwort, angezeigt. Falls wir CRUD auch für die Ausleihen machen, werden wir auch die Ausleihen auf dieser Seite einbinden.

### User bearbeiten Page (users/edit/[i])

Erreichbar über die /users-Seite enthält ein Formular, welches vorbefüllt ist mit den Nutzerdaten des aktuellen Nutzers. Der Inhalt kann bearbeitet werden und beim Submit wird der entsprechende User bearbeitet.

### Login (/login)

Auf dieser Seite findet sich ein Formular mit Benutzername und Passwort und es findet beim Submit die Authentifikation statt.

### Sign-Up (/sign-up)

Auf dieser Seite kann, wie auch bei der Seite (/users/create) ein neuer Benutzer erstellt werden, analog findet sich ein Formular, welches mit den Benutzerdaten des zu erstellenden Benutzers befüllt werden können.

### Impressum (/impressum)

Beinhaltet ein standardmässiges Impressum, welches über den Footer aufgerufen werden kann.

## Layouts

Die Layouts der Seiten findet sich im anderen Abgabedokument, unter dem Namen «Entwurf-Pages-pdf»

### Farbkonzept

Wir halten unser Farbkonzept sehr simpel, also nur schwarz für Text weiss für den Hintergrund und grau für die Navigation oder Knöpfe.

Unser Farbkonzept:

|  |  |  |
| --- | --- | --- |
| 000000 | D9D9D9 | FFFFFF |

# TEIL 3

## Technische Dokumentation

Im Rahmen des Projekts kommen die folgenden Technologien zum Einsatz:

|  |  |
| --- | --- |
| **TECHNOLOGIE** | **VERSION** |
| Visual Studio Code | 1.87.2 |
| Java Script | ES6 |
| CSS | CSS2/CSS3 |
| Intelij | 232.10227.8 |
| Node.js | 20.11.1 |
| React | ^18 |
| Next | 14.1.3 |
| Npm | 10.5.0 |
| Postman | 10.24.3 |
| Chrome | 123.0.6312.86 |
| Git | 2.42.0.2 |
| Cypress | 13.7.1 |

## Testkonzept

### Testumgebung

Wir werden unsere Tests auf Windows machen.

Wir testen auf Chrome, Version ist in unserer Toolchain ersichtlich.

Wir verwenden für alle Tests Cypress, diese kann durch cypress open im Terminal geöffnet werden, dann sollte E2E ausgewählt werden und die Testumgebung (Chrome). Von dort aus findet man die specs, die Tests beinhalten.

### Testfälle

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 1 |
| Anforderungen | FA-01, FA-04 (Read für Users), NFA-03 |
| Vorbedingungen | Frontend und Backend wurden gestartet. Gültiger User vorhanden |
| Ablauf | * Im Browser <http://localhost:3000/> eingeben * Mit der Navigation auf Login navigieren. * Sich mit einem vorhandenen Benutzer anmelden:   Benutzername: m  Passwort: 1234   * Auf den Login Knopf klicken * Überprüfen, ob URL wieder <http://localhost:3000/> ist. * Überprüfen, ob Sign-up-Link aus der Navigation verschwunden ist. |
| Erwartetes Resultat | Benutzer kann sich erfolgreich Anmelden und wird auf die Index Seite zurückgeleitet, die Navbar-Links sind angepasst. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 2 |
| Anforderungen | FA-02, FA-04 (Create, Delete) NFA-03 |
| Vorbedingungen | Frontend und Backend wurden gestartet. Test 1 bestanden. |
| Ablauf | * Im Browser <http://localhost:3000/> eingeben * Mit der Navigation auf Sign-Up navigieren. * Sich mit neuen Angaben registrieren. * Auf den Login Knopf klicken * Überprüfen, ob die URL <http://localhost:3000/login> ist. * Sich mit den Benutzerdaten anmelden. * Auf <http://localhost:3000/users> navigieren. * Erstellten User suchen. * Auf ‘Mehr Details’ klicken * User Löschen * Überprüfen, ob wieder auf der <http://localhost:3000/users-Seite> * Überprüfen, ob der User gelöscht wurde. |
| Erwartetes Resultat | Benutzer kann sich registrieren, mit den neuen Daten anmelden, der User wird auf Users angezeigt und ist löschbar. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 3 |
| Anforderungen | FA-03 (Read, Update), NFA-03 |
| Vorbedingungen | Frontend und Backend wurden gestartet. Eingeloggt |
| Ablauf | * Im Browser <http://localhost:3000/> eingeben * Auf Mehr Details bei dem Audiobuch “Mastering the Art of French Cooking” klicken * Dort auf Bearbeiten * Autor für das Audiobuch ändern. * Nun auf Erstellen klicken * Überprüfen, ob wieder auf <http://localhost:3000> * Überprüfen, ob der neu gesetzte Autor wieder vorhanden ist. |
| Erwartetes Resultat | Der Autor des Buchs wurde verändert. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 4 |
| Anforderungen | FA-03 (Create, Delete) |
| Vorbedingungen | Frontend und Backend wurden gestartet. Testfall 1 bestanden.  Valider User vorhanden. |
| Ablauf | * Im Browser <http://localhost:3000/> eingeben * Auf Login navigieren. * Folgende Benutzerdaten eingeben und dann abschicken: Benutzername: ‘m’ Passwort: ‘1234’ * Auf ‘Erstellen’ drücken. * Audiobücherdetails angeben. * Erstellen drücken. * Erstelltes Buch auf Vorhandensein überprüfen. * Mehr Details und dann Löschen * Audiobuecher auf Absenz des gelöschten Buches untersuchen. |
| Erwartetes Resultat | Ein Audiobuch wird erstellt, dieses lässt sich auffinden und ist wieder löschbar. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 5 |
| Anforderungen | FA-01, NFA-03 |
| Vorbedingungen | Frontend und Backend wurden gestartet. |
| Ablauf | * Im Browser <http://localhost:3000/> eingeben * Mit der Navigation auf Login navigieren. * Sich mit nicht vorhandenen Benutzerdaten anmelden:   Benutzername: keinBenutzer  Passwort: IchbinkeinBenutzer   * Auf den Login Knopf klicken * Ist der Sign-Up-Link noch vorhanden in der Navigation. |
| Erwartetes Resultat | Der Benutzer kann sich **nicht** anmelden. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 6 |
| Anforderungen | FA-04 (Update) |
| Vorbedingungen | Frontend und Backend wurden gestartet. Ein User kann erstellt werden. |
| Ablauf | * Im Browser <http://localhost:3000/>sign-up eingeben * Das Formular mit Benutzerdaten befüllen. * Login * Navigieren nach <http://localhost:3000/users> * Den erstellten User finden, auf ‘Mehr Details’ * Auf ‘Bearbeiten’ klicken. * Benutzername und Passwort ändern. * Auf ‘Erstellen’ drücken. * Überprüfen, ob wieder auf der Seite <http://localhost:3000/users> * Sicherstellen, dass der Benutzername geändert wurde. |
| Erwartetes Resultat | Der Benutzername wurde wie gewünscht geändert. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 7 |
| Anforderungen | - |
| Vorbedingungen | Frontend und Backend wurden gestartet. |
| Ablauf | * Im Browser <http://localhost:3000/>users eingeben * Überprüfen, ob <http://localhost:3000/users> in der URL ist. |
| Erwartetes Resultat | Die URL sollte **nicht** <http://localhost:3000/users> sein. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall 8 |
| Anforderungen | FA-01 |
| Vorbedingungen | Frontend und Backend wurden gestartet. Log-in ist möglich.  Valide User ist vorhanden. |
| Ablauf | * Im Browser <http://localhost:3000/>login eingeben * Valide Benutzerdaten eingeben, dann auf Login drücken. Benutzername: m Passwort: 1234 * Überprüfen, ob nun Logout in der Navbar erscheint. * Überprüfen, dass Login nicht mehr vorhanden ist. * Log-out drücken. * Überprüfen, ob nun Login wieder vorhanden ist. |
| Erwartetes Resultat | Links passen sich an, wenn nicht eingeloggt oder wenn eingeloggt. |

## Testprotokoll

|  |  |  |
| --- | --- | --- |
| **Testfall Nr** | **Erfolgreich?** | **Bemerkung / Erklärung wenn nicht erfolgreich** |
| Testfall 1 | Ja |  |
| Testfall 2 | Ja |  |
| Testfall 3 | Ja |  |
| Testfall 4 | Ja |  |
| Testfall 5 | Ja |  |
| Testfall 6 | Ja |  |
| Testfall 7 | Ja |  |
| Testfall 8 | Ja |  |

## W3C Validation

Alle Seiten konnten nach kleinen Anpassungen Warnung und Error frei getestet werden. Die Anpassungen die gemacht werden musste waren bei der Seite \_documents.js von:

import { Html, Head, Main, NextScript } from "next/document";

export default function Document() {

return (

<Html lang="en">

<Head />

<body>

<Main />

<NextScript />

</body>

</Html>

);

}

Zu:

import { Html, Head, Main, NextScript } from "next/document";

export default function Document() {

return (

<Html lang="en">

<Head>

<title>Audit</title>

</Head>

<body>

<Main />

<NextScript />

</body>

</Html>

);

}

Der Validator gibt fälschlicherweise eine Fehlermeldung aus ( Error: Element head is missing a required instance of child element title.). Seine Diagnose ist in sich widersprüchlich, da er das Fehlen eines title Elements als Fehler meldet, den HTML 5.1-Entwurf jedoch unter anderem mit den Worten zitiert, dass ein Element nicht erforderlich sei, wenn „Titelinformationen von einer höheren Ebene verfügbar sind“. Protokoll".

## Fazit

Wir haben eine Planung für unser Projekt gemacht und konnten uns an diese halten, dazu hatten wir genug Zeit und sogar genug übrig, um weitere Aufgaben zu erfassen und verteilen. Ich denke, dass wir all unsere Anforderungen testen konnten.

Schlecht ist nichts gelaufen, wir hatten zwar Schwierigkeiten, das Backend zu integrieren, aber dabei wesentlich weniger, als die Allgemeinheit hatte, daher würde ich eher behaupten, dass es relativ gut gegangen ist.

Das Endergebnis ist sehr zufriedenstellend, wir konnten die ganzen Seiten erstellen, dabei mit dem Backend kommunizieren und viel über React und Next lernen.

Wie bereits erwähnt, haben wir gelernt, CRUD für Entitäten aus dem Frontend heraus zu implementieren, Seiten zu pre-rendern, die Handhabung von React und Next und ihren Komponenten und Hooks, was node und npm sind und auch der Umgang mit cypress.

Wir konnten unseren gesamten Projektumfang erfüllen und es ist alles vorhanden. Wir könnten noch White-Box-Test für die UX machen, aber es ist schwierig, Leute zu finden, die Zeit dafür haben.