

# Masterarbeit

zur Erlangung des akademischen Grades

Master

[überarbeitete Version]

**Technische Hochschule Wildau**

**Fachbereich Wirtschaft, Informatik, Recht**

**Studiengang Bibliotheks-informatik (M. Sc.)**

Thema (deutsch): Entwicklung eines Konzepts für eine App zur Überwindung des Medienbruchs beim Stöbern am Regal in der Bibliothek der TH Wildau

Thema (englisch): Concept for an application to directly integrate electronic resources in the research at the library shelves of the Technical University of Applied Sciences Wildau

Autorin: Dana Geier  
Matrikel-Nr.: 50077469  
Seminargruppe: BIM/18  
Betreuer: Prof. Dr. rer. nat. Janett Mohnke  
Zweitgutachter: Dr. Frank Seeliger  
Spätestmögliche Abgabe: 25.08.2021  
Eingereicht am: 25.08.2021

**Dana Geier**

Entwicklung eines Konzepts für eine App zur Überwindung des Medienbruchs beim Stöbern am Regal  
in der Bibliothek der TH Wildau

Masterarbeit, 25.08.2021

Gutachter: Prof. Dr. rer. nat. Janett Mohnke, Dr. Frank Seeliger

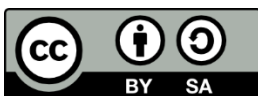
**Wildau Institute of Technology**

Technische Hochschule Wildau

Hochschulring 1

15754 Wildau

Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz.



# Zusammenfassung

Zielstellung der Arbeit ist die Entwicklung eines Konzepts für eine Anwendung zur Überwindung des Medienbruchs zwischen physischen und elektronischen Ressourcen am Regal für die Bibliothek der Technischen Hochschule Wildau. Als Ausgangspunkte dafür werden ein bereits bestehender Prototyp sowie die Ergebnisse eines Kickoff-Meetings mit Kolleg/innen der Hochschulbibliothek und des iCampus-Teams der Technischen Hochschule Wildau genutzt.

Zunächst wird dabei die Problematik des Medienbruchs in Bibliotheken erläutert und bestehende Ansätze zur Lösung des Problems skizziert. Es folgen bibliothekarische Vorbetrachtungen zur Standortermittlung, Sacherschließung und Aufstellung in Bibliotheken gefolgt von kurzen technischen Vorbetrachtungen zur Art der App und verwendbaren Frameworks.

Anschließend wird der Ist-Zustand dargestellt. Dazu werden die Ausgangsvoraussetzungen an der Hochschulbibliothek sowie Funktionen und Architektur des bestehenden Prototyps beschrieben.

Schließlich werden die Stakeholder ermittelt und anhand Personas beschrieben, bevor die funktionalen und nicht funktionalen Anforderungen erfasst werden.

Hauptteil der Arbeit stellt das Konzept für die Anwendung dar. Im darauffolgenden Kapitel wird die Implementierung der Integration entliehener Bände, der Vormerkfunktion sowie der Durchschaltung zum E-Book beschrieben.

In den Schlussbetrachtungen schließlich wird ein Fazit gezogen, aufgefallene Probleme erläutert und ein Ausblick auf Weiterentwicklungsmöglichkeiten geworfen.

## Abstract

Purpose of this master thesis is to create a concept for an application to directly integrate electronic resources in the research at the library shelves of the Technical University of Applied Sciences Wildau. This concept is based on an existing prototype and the results of a kickoff-meeting with colleagues from the library and the iCampus-Team of the Technical University of Applied Sciences Wildau.

First of all the problem of the integration of electronic resources in the research at the library shelves and solutions for that in different libraries will be presented. Furthermore library topics like locating, subject cataloguing and shelving as well as technical topics like the type of the app or different frameworks are outlined.

Subsequently the actual state is described. Therefore the requirements of the library and the different functions and architecture of the existing prototype are specified.

In the next chapter the stakeholder are determined and described followed by the establishing of the functional and non-functional requirements.

The concept for the application forms the largest chapter of this thesis. The following chapter contains the description of the implementation of the integration of lent print books, the reservation-function and the possibility to directly open fulltext-PDFs in the application.

Finally there is a conclusion and perspectives for the further development of the app will be presented.

# Abkürzungsverzeichnis

API	Application Programming Interface
App	Application, hier für Web-Anwendung
CSN	Cutter-Sanborn-Notation/-Nummer
CSS	Cascading Style Sheets
DDC	Dewey Decimal Classification
DOI	Digital Object Identifier
EPUB	Akronym für <i>electronic publication</i>
GESIS	Lebniz-Institut für Sozialwissenschaften
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
ISBN	International Standard Book Number
ISMN	International Standard Music Number
JSON	JavaScript Object Notation
KOBV	Kooperativer Bibliotheksverbund Berlin-Brandenburg
LDAP	Lightweight Directory Access Protocol
NFC	Nearfield Communication
PDA	Patron Driven Acquisition
PDF	Portable Document Format
POI	Point of Interest
QR-Code	Quick Response-Code
REST	Representational State Transfer
RFID	Radio Frequency Identification
RVK	Regensburger Verbundklassifikation
SLNP	Simple Library Network Protocol
SOAP	<i>(ursprünglich)</i> Simple Object Access Protocol
TH	Technische Hochschule
UI	User Interface

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
Wilbert	Wildauer Bücher+E-Medien Recherche-Tool
XML	Extensible Markup Language

## Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei den Kollegen der Technischen Hochschule Wildau bedanken, die mir im Rahmen der Erstellung der Arbeit stets hilfreich zur Seite standen.

Ganz herzlicher Dank soll zunächst an die beiden Betreuer der Arbeit gehen. Vielen Dank für die zeitnahen Antworten auf meine Rückfragen, die sehr gute inhaltliche Betreuung und die ständige Erreichbarkeit.

Zu nennen sind an dieser Stelle auch vor allem die Kollegen der Hochschulbibliothek: Vielen Dank an Herrn Kissig, für die zahlreichen Hinweise, die Umsetzung von neuen bzw Einführung in bestehende Schnittstellen und die gemeinsame Fehlersuche bei der Installation der App auf dem Server der Bibliothek. Ein großer Dank geht auch an Frau Petri, die mir mit Tipps und Tricks zur praktischen Arbeit mit den QR-Codes weitergeholfen hat und Rückfragen zu E-Book-Paketen beantwortet hat. Vielen Dank auch an Frau Keidel für Rückfragen zur Erwerbung von elektronischen Ressourcen.

Danken möchte ich darüber hinaus auch den Kollegen des iCampus-Teams der TH Wildau, die mir stets bei Rückfragen zu vorhandenen Schnittstellen weitergeholfen haben.

Ein besonderer Dank geht an Christopher Grätz und Omar Attia, die mich bei der Einarbeitung in die vorhandene Version der Anwendung unterstützt haben.

Dank geht darüber hinaus an Herrn Kant vom KOBV, der mir stets hilfreich für Rückfragen zur Solr-Schnittstelle des KOBV zur Verfügung stand.

Last but not least möchte ich mich bei meinem privaten Umfeld bedanken, dass ihr mich in der Zeit der Erarbeitung der Arbeit unterstützt und immer wieder motiviert habt!

## Hinweise zum Lesen der Arbeit

Aus Gründen der besseren Lesbarkeit wird in der Arbeit ausschließlich das generische Maskulinum verwendet. Weibliche und andere Geschlechteridentitäten werden dabei ausdrücklich mitgemeint.

Zitate werden in Fußnoten angegeben. Bei der ersten Zitation erfolgen Angaben zum Verfasser, gefolgt von Angaben zum Erscheinungsjahr in Klammern sowie Titel und Titelzusatz sowie Seitenangaben. Bei Beiträgen aus Sammelwerken sowie Zeitschriftenaufsätzen sind zusätzlich noch die Angaben zum übergeordneten Werk (hier: Sammelwerk bzw Zeitschrift) bei der ersten Nennung angegeben. Die zweiten und weiteren Zitationen eines Werkes erfolgen in der Kurzform *Verfasser/Herausgeber (Jahr), Seitenzahl*. Bezieht sich eine Fußnote auf den gleichen Titel wie die vorherige Fußnote, wird dies mit *Ebd, S. 123* angegeben. Stimmt auch die Seitenzahl mit der vorherigen Fußnote überein, wird dies lediglich mit *Ebd.* angegeben.

Die vollständigen bibliographischen Angaben zu allen Angaben finden sich am Ende der Arbeit in alphabetischer Reihenfolge (nach Verfassern) in einem Literaturverzeichnis.

Codebeispiele sind als Listings gekennzeichnet.

In der Arbeit verwendete Abkürzungen sind im Abkürzungsverzeichnis am Anfang der Arbeit aufgelistet. In der Arbeit werden diese bei erstmaliger Verwendung ausgeschrieben und anschließend die Abkürzung in Klammern genannt. Bei allen weiteren Verwendungen wird die nur noch die Abkürzung genannt.

Darüber hinaus findet sich am Ende der Arbeit ein Glossar, in dem sich zu ausgewählten Begriffen nochmals Erläuterungen finden. Bei der Erstnennung im Text wird anhand eines ✎ gekennzeichnet, dass es zu dem Begriff einen Glossareintrag gibt.

Die Arbeit als PDF-Version findet sich auf der beiliegenden CD-ROM. Sämtliche Quellcodes werden in einem Git-Repository bereitgestellt. Weitere Hinweise dazu finden sich in Anhang A.6



# Inhaltsverzeichnis

<b>Zusammenfassung</b> .....	<b>i</b>
<b>Abstract</b> .....	<b>i</b>
<b>Abkürzungsverzeichnis</b> .....	<b>iii</b>
<b>Danksagung</b> .....	<b>v</b>
<b>Hinweise zum Lesen der Arbeit</b> .....	<b>vi</b>
<b>1 Einleitung</b> .....	<b>1</b>
1.1 Motivation .....	1
1.2 Zielsetzung .....	3
1.3 Abgrenzung .....	3
1.4 Aufbau der Arbeit .....	4
<b>2 Stöbern und die Integration des Digitalen ins Physische</b> .....	<b>5</b>
2.1 Der Medienbruch online und am Regal .....	5
2.2 Lösungsansätze .....	7
<b>3 Bibliothekarische und technische Vorüberlegungen</b> .....	<b>11</b>
3.1 Standortermittlung von Nutzern in Bibliotheken .....	11
3.1.1 iBeacons .....	11
3.1.2 RFID bzw. NFC .....	12
3.1.3 Mobile Tagging mit QR-Codes .....	13
3.1.4 Empfehlung .....	15
3.2 Sacherschließung und Aufstellung in Bibliotheken .....	15
3.2.1 Sacherschließung .....	15
3.2.2 Anforderungen an die Aufstellung in wissenschaftlichen Bibliotheken .....	18
3.3 Formate von E-Ressourcen in Bibliotheken .....	18
3.4 Native oder Web-App? .....	19
3.5 Mögliche Frameworks zur Entwicklung von Web Apps .....	21
3.6 Zusammenfassung .....	23
<b>4 Ist-Stand</b> .....	<b>24</b>
4.1 Die Hochschulbibliothek der TH Wildau .....	24
4.2 Recherche- und Informationstools .....	25
4.2.1 Wilbert .....	25
4.2.2 Online-Katalog .....	26
4.2.3 Campus App Unidos Wildau .....	26
4.2.4 Weitere digitale Services .....	28
4.3 Der Prototyp der App E-Books-To-Go .....	29

4.3.1	Umgesetzte Funktionalitäten.....	30
4.3.2	Technische Umsetzung/Systemarchitektur .....	34
4.4	Zusammenfassung.....	40
<b>5</b>	<b>Anforderungsanalyse/Requirements Engineering .....</b>	<b>41</b>
5.1	Stakeholder.....	41
5.2	Erstellung der Personas .....	42
5.2.1	Persona 1 – Student.....	42
5.2.2	Persona 2 – Dozentin.....	42
5.2.3	Persona 3 – Bibliotheksmitarbeiter .....	43
5.3	Use-Cases.....	43
5.4	Anforderungen.....	44
5.4.1	Anforderungen funktional.....	45
5.4.2	Anforderungen nicht funktional.....	47
<b>6</b>	<b>Konzept .....</b>	<b>48</b>
6.1	Technische Gesamtarchitektur.....	48
6.1.1	Frontend .....	49
6.1.2	Backend.....	50
6.2	Zu integrierende Medientypen.....	51
6.3	Standorterkennung .....	52
6.3.1	Integration von QR-Codes in die App .....	52
6.3.2	Erstellung von QR-Codes .....	54
6.3.3	Unterbringung der QR-Codes am Regal.....	54
6.4	Schnittstelle für die Ermittlung der Ressourcen.....	55
6.5	Ermittlung von zum Nutzerstandort passenden E-Ressourcen .....	59
6.5.1	DDC versus RVK bei E-Book-Central.....	59
6.5.2	Treffermengen bei Nutzung der RVK-Notation.....	62
6.5.3	Sonderfall E-Journals.....	63
6.5.4	Anfragen an Solr-Proxy .....	64
6.6	Ermittlung der „neusten“ E-Books.....	64
6.7	Ermittlung entliehener Print-Monographien.....	65
6.8	Layout .....	67
6.8.1	Allgemeine Anforderungen.....	67
6.8.2	Darstellung der E-Books durch ihre Cover .....	68
6.8.3	Trefferlisten .....	70
6.8.4	Detaildarstellung.....	71
6.9	Filter- und Sortiermöglichkeiten.....	72
6.10	Authentifizierung .....	73
6.11	Vormerkung .....	74
6.12	Empfehlungen .....	75

6.12.1 Anzeigen von Empfehlungen für eigenen Studiengang anzeigen .....	75
6.12.2 Anzeigen von Empfehlungen für andere Studiengänge .....	77
6.12.3 Senden von E-Mail-Empfehlungen .....	77
6.12.4 Einzelne Medien bewerten – einige Ideenskizzen .....	79
6.13 Direkter Zugang/Durchschaltung zu E-Books .....	80
6.14 Integration von einfacher Suchfunktion .....	81
6.15 Datenschutz .....	81
6.16 Ideen für die Benennung der Anwendung .....	82
<b>7 Umsetzung ausgewählter Aspekte .....</b>	<b>83</b>
7.1 Vom Wilbert-HTML zum Solr-API-XML .....	83
7.2 Auf den QR-Codes zu hinterlegende Informationen .....	86
7.3 Erstellung von QR-Code-Listen .....	88
7.4 Direkter Zugang zum E-Book aus der App heraus .....	91
7.5 Integration entliehener Werke .....	99
7.6 Vormerkung .....	104
7.7 Zusammenfassung technische Umsetzung .....	108
<b>8 Abschlussbetrachtungen .....</b>	<b>109</b>
8.1 Umgesetzte Anforderungen .....	109
8.2 Fazit zur praktischen Umsetzung .....	110
8.3 Problemstellungen .....	110
8.4 Ausblick .....	112

<b>Literaturverzeichnis.....</b>	<b>I</b>
<b>Abbildungsverzeichnis.....</b>	<b>VII</b>
<b>Listingverzeichnis .....</b>	<b>IX</b>
<b>Tabellenverzeichnis.....</b>	<b>X</b>
<b>Glossar .....</b>	<b>XI</b>
<b>A. Anhang.....</b>	<b>XIV</b>
<b>A.1 Ergebnisse des Get Togethers am 13.11.2020.....</b>	<b>XIV</b>
<b>A.2 E-Mail an die Mailingliste InetBib.....</b>	<b>XVI</b>
<b>A.3 Öffentliche Etherpad-Antworten zur Inetbib-Umfrage.....</b>	<b>XVII</b>
<b>A.4 Übersicht RVK-Notationen pro Regalboden (Bereich Informatik).....</b>	<b>XXIII</b>
<b>A.5 E-Mail mit Anforderungen an API zur Ermittlung entliehener Print-Monographien.....</b>	<b>XXVIII</b>
<b>A.6 CD-ROM zur Arbeit .....</b>	<b>XXIX</b>
<b>Selbständigkeitserklärung .....</b>	<b>XXX</b>

# 1 Einleitung

Waren es einst Zettel- und dann Online-Kataloge (Online Public Access Catalogues (OPACS) ↗), so sind es heute Discovery-Systeme ↗, die einem in den Sinn kommen, wenn man ans Auffinden von Medien in Bibliotheken denkt. Als Beispiele sollen an dieser Stelle nur kurz Primo (Discovery-System der Firma Alma) oder Lukida (Discovery-Service der VZG) genannt werden.

Bereits seit Aufkommen der ersten elektronischen Ressourcen und dem Entstehen des Medienbruchs, also dem „dauernden Wechsel zwischen elektronischem und gedrucktem Text“<sup>1</sup>, wurde versucht, selbigem durch die Integration elektronischer Ressourcen in die Online-Kataloge entgegenzuwirken, um den Nutzern einen vollständigen Überblick über den Bestand zu gewährleisten. Es finden sich zahlreiche Publikationen zu Möglichkeiten, physische Bestände digital auffind- bzw. abbildbar zu machen, von Discovery-Systemen bis hin zu Blended Shelves, welche das analoge Bibliotheksregal digital abbilden sollen.<sup>2</sup>

Stellen Discovery-Systeme zwar die wohl wichtigste Möglichkeit in Bibliotheken dar, auf relevante Medien zu stoßen, jedoch bei weitem nicht die einzige. Eine weitere Möglichkeit, die für den eigenen Forschungszweck relevante Literatur auffindig zu machen, stellt das Stöbern am Regal dar.

Doch wie sieht es mit dem Medienbruch beim Stöbern am Regal aus? Erstmals intensiv mit dem Thema beschäftigte sich Jasmin Taubert in ihrer Masterarbeit „Absentia in Praesentia: Zur Präsentation und Vermittlung digitaler Medien im physischen Raum“ aus dem Jahr 2013.

Schaut man sich wissenschaftliche Bibliotheken an, so macht ein Großteil der Flächen der Bibliotheken neben (Gruppen-)Arbeitsplätzen, die nach wie vor an Bedeutung gewinnen, die Aufstellung des gedruckten Bestandes aus.

Tessa Högele beschreibt die Bedeutung des Stöberns am Regal folgendermaßen:

*„In Freihandbibliotheken ist es möglich, Wissen zu entdecken, auf das man bei einer zielgerichteten Suche in einem Onlineportal nicht gestoßen wäre. Das Wissen mag nicht immer direkt für die Hausarbeit oder den Essay verwertbar sein, an dem man eigentlich arbeitet. Unnötige Lektüre ist es deshalb nicht. Ganz vielleicht stößt man sogar auf ein Buch, das in Kombination mit dem Thema, zu dem man eigentlich recherchiert, eine Explosion auslöst. Das einen neuen Gedankengang zu dem Forschungsthema entfacht, eine eigene, neuartige Idee entstehen lässt.“<sup>3</sup>*

## 1.1 Motivation

Die Bibliothek der Technischen Hochschule (TH) Wildau, welche im Jahr 2012 als „Bibliothek des Jahres“ ausgezeichnet wurde, gilt als eine kreative, innovations- und experimentierfreudige Bibliothek,

---

<sup>1</sup> Vgl. Mittler (1996): Die Rolle der Bibliothek, <http://www.imn.htwk-leipzig.de/~bastian/unendl/mittler.txt>, 19.01.2021.

<sup>2</sup> Vgl. z.B. Kleiner (2013): Blended Shelf.

<sup>3</sup> Vgl. Högele (2018): Warum es der Kreativität schadet, wenn Universitätsbibliotheken ihre Bücherregale abschaffen, <https://ze.tt/universitaetsbibliotheken-in-denen-man-nicht-nach-buechern-stoebeln-kann-sind-tot/>, 03.05.2021.

welche durch eine Vielzahl zukunftsweisender Serviceleistungen Akzente setzt.<sup>4</sup> Gemeinsam mit der Hochschule werden verschiedenste Projekte angegangen, unter anderem mit dem Team des RoboticLab des Studiengangs Telematik.

Ein weiteres Beispiel für eine solche Kooperation ist die Entwicklung der App<sup>5</sup> „E-Books<sup>6</sup> To Go“: Im Wintersemester 2019/2020 wurde von Christopher Grätz und Omar Attia im Rahmen eines Praktikums während des Telematik-Studiums ein Prototyp einer webbasierten Anwendung entwickelt, welche das Ziel hat, „den Medienbruch zwischen gedruckten und elektronischen Büchern zu überwinden.“<sup>7</sup> Resultat des Praktikums war der Prototyp der Anwendung „E-Books To Go“.

Die Anwendung soll die Herausforderung aufgreifen, die Freihandaufstellung beibehalten zu können, um

*„dieses intelligente, abenteuerliche Herumwandeln am Regal weiterhin [...] ermöglichen zu können [...] und dann aber natürlich die Leser immer darauf aufmerksam machen, dass da im Regal Lücken sind [...] das sind Angebote, die sind digital, die können wir nicht ins Regal stellen. Das ist die Herausforderung der Zukunft.“<sup>8</sup>*

Mit Hilfe der App soll aber nicht nur auf die Lücken aufmerksam gemacht werden, sondern diese auch gefüllt werden. Die App soll es Nutzern auch beim Stöbern am Regal ermöglichen, passend zum Themenstandort einen Einblick in die vielfältigen digitalen Angebote der Bibliothek zu erhalten.

Neben der Überwindung des Medienbruchs soll eine erhöhte Präsenz der elektronischen Ressourcen in den Köpfen der Nutzer, die die Bibliothek vor Ort nutzen, erreicht werden. Dadurch wiederum kann eine Steigerung der Nutzungszahlen des digitalen Bestandes erhöht werden. Nicht zuletzt kann die Anwendung als Marketinginstrument das Image der Hochschulbibliothek der TH Wildau als innovative und moderne Bibliothek steigern.<sup>9</sup>

Bibliotheken stehen darüber hinaus vor der Herausforderung, neben ihren traditionellen Aufgaben wie bspw. Erwerbung und Ausleihe, den Nutzern „ihre qualitativ hochwertig aufbereiteten Daten und Informationen [...] über webbasierte, also von überall zugängliche, für mobile Geräte aufbereitete, intuitiv anwendbare und attraktive Dienste anzubieten.“<sup>10</sup> Zielstellung sollte es vor allem sein, „die Fachliteratur so zu transportieren, dass sie die Nutzerinnen und Nutzer auf dem für sie direktesten Wege erreicht.“<sup>11</sup>

---

<sup>4</sup> Vgl. Deutscher Bibliotheksverband e.V. (n. N.): Bibliothek des Jahres 2012, <https://www.bibliotheksverband.de/dbv/auszeichnungen/bibliothek-des-jahres/preistraeger/2012.html>, 13.02.2021.

<sup>5</sup> Im Rahmen der Arbeit werden die Begriffe App und (mobile) Anwendung synonym verwendet.

<sup>6</sup> Diese Arbeit folgt für den Begriff E-Book der Rechtschreibung gemäß Duden ( Vgl. Bibliographisches Institut (2021): Duden: E-Book, [https://www.duden.de/rechtschreibung/E\\_Book](https://www.duden.de/rechtschreibung/E_Book), 17.08.2021) In direkten Wortziten wird die Schreibweise der jeweiligen Quelle wiedergegeben.

<sup>7</sup> Vgl. Grätz (2020b): Praktikumsbericht iLibrary, S. 1.

<sup>8</sup> Ulrich Johannes Schneider in einem Interview mit dem Radiosender MDR Figaro am 10.03.2011. Zitiert nach: Vgl. Ackermann (2011): Die E-Books am Regal finden, [http://www.opus-bayern.de/bib-info/volltexte//2011/998/pdf/E\\_Books\\_per\\_QR.pdf](http://www.opus-bayern.de/bib-info/volltexte//2011/998/pdf/E_Books_per_QR.pdf).

<sup>9</sup> Vgl. Taubert (2016): Absentia in Praesentia? Absentia in Praesentia? Zur Präsentation von digitalen Beständen im physischen Raum, Folie 4.

<sup>10</sup> Vgl. Goltz (2015): Mobile Applikationen für Bibliotheken im deutschsprachigen Raum, S. 2.

<sup>11</sup> Ebd., S. 3.

## 1.2 Zielsetzung

Im Rahmen dieser Masterarbeit soll u.a. auf Grundlage des bestehenden Prototyps ein allgemeines Konzept für eine Anwendung für eine App zur Überwindung des Medienbruchs beim Stöbern am Regal in der Bibliothek der TH Wildau entworfen werden.

Das zu betrachtende Szenario geht von einer vor-Ort-Nutzung der Anwendung durch Hochschulangehörige (Studenten, Dozenten, Mitarbeiter) aus.

In späteren Szenarien könnte die Anwendung in die Campus-App Unidos oder das 3D-System der Bibliothek integriert werden und so ggf. zusätzlich den Nutzern außerhalb der Räumlichkeiten zur Verfügung stehen.

Im Rahmen der Arbeit sollen darüber hinaus neue Funktionen im bereits bestehenden Prototyp umgesetzt werden. Dazu gehören unter anderem die Anzeige entliehener Werke sowie eine Vormerkfunktion derer. Darüber hinaus soll es die Möglichkeit geben, Volltext-PDFs von E-Books direkt ohne Umweg über die Anbieterseite öffnen zu können.

## 1.3 Abgrenzung

Im Rahmen der Arbeit soll ein Konzept entwickelt werden, welches an die Bedingungen und Voraussetzungen an der Hochschulbibliothek der TH Wildau angepasst ist. Zielstellung soll es nicht sein, ein generelles Konzept für wissenschaftliche oder Hochschulbibliotheken zu erstellen.

Darüber hinaus ist Schwerpunkt der Arbeit das Konzept für die Anwendung und nicht die vollständige Umsetzung der im Konzept beschriebenen Funktionalitäten. Der bestehende Prototyp soll lediglich um ausgewählte Funktionalitäten erweitert werden, die in der Umsetzungsdokumentation beschrieben werden.

Beschäftigt sich das Konzept ggf. mit weiteren Medientypen, sind im Prototyp vorerst nur E-Books integriert. Auch die Erweiterung um ausgeliehene Bände bezieht sich auf Monographien.

Da der Schwerpunkt der Arbeit auf der Betrachtung der technischen Umsetzungsmöglichkeiten liegt, wird an dieser Stelle auf eine qualitative oder quantitative Bedarfsanalyse im Rahmen von Umfragen oder Interviews verzichtet. Für die Betrachtung der Lösungsmöglichkeiten der Problematik wurden stattdessen Kollegen über die Mailingliste Inetbib angeschrieben mit der Bitte, Ihren Umgang mit der Problematik in einem Etherpad kurz zu schildern.<sup>12</sup>

Für die Umsetzung des Prototyps wurde mit der Solr-Test-API für die TH Wildau gearbeitet. Für den Produktivbetrieb muss entsprechend auf die live Solr-API umgestiegen werden.

---

<sup>12</sup> Das öffentliche Etherpad findet sich unter: [https://yopad.eu/p/E-Books\\_Stoebem\\_Bibliotheken-365days](https://yopad.eu/p/E-Books_Stoebem_Bibliotheken-365days). Die E-Mail an die Mailingliste findet sich im Anhang.

## 1.4 Aufbau der Arbeit

Zu Beginn der Arbeit soll unter anderem im Hinblick auf die Ergebnisse der Masterarbeit von Frau Taubert (heute Praessler)<sup>13</sup> ein Einblick in die Problematik gegeben werden sowie verschiedene Lösungsansätze der Integration des Digitalen ins Physische anderer Einrichtungen kurz dargestellt werden. Darüber hinaus sollen unterschiedliche Umsetzungsmöglichkeiten für die verschiedenen bestehenden Problemstellungen wie die Standortermittlung des Nutzers in der Bibliothek betrachtet werden.

Anschließend soll der Ist-Zustand an der Hochschulbibliothek an der TH Wildau beschrieben sowie der Prototyp der App vorgestellt werden. Dabei soll die technische Architektur und die verwendeten Frameworks<sup>14</sup> beschrieben sowie deren Nutzung begründet werden.

Im Hauptteil wird aufbauend auf User Stories eine Anforderungsanalyse entwickelt.<sup>14</sup> Dem schließt sich das Konzept an, in welchem die ermittelten Anforderungen ausführlich beschrieben und Lösungsmöglichkeiten erarbeitet werden. Weiterhin werden benötigte Daten und Schnittstellen betrachtet.

Im praktischen Teil der Arbeit soll der bestehende Prototyp der App sowohl um die Möglichkeit des direkten Zugangs zum E-Book (hier: Volltext-PDF), als auch um die Anzeige entliehener Werke (ggf. mit der Möglichkeit, eine Vormerkung abzusetzen) erweitert werden.

Abschließend soll ein Fazit gezogen und bei der Umsetzung aufgetretene Probleme zusammengefasst sowie ein Ausblick auf weitere Entwicklungsmöglichkeiten gegeben werden.

---

<sup>13</sup> Vgl. Taubert (2012): Absentia in Praesentia? Zur Präsentation und Vermittlung digitaler Medien im physischen Raum.

<sup>14</sup> Da der Schwerpunkt der Arbeit auf der Entwicklung des eigentlichen Konzepts sowie der Erweiterung des Prototyps liegt, wurden die Hauptanforderungen an die App im Voraus besprochen. Auf eine qualitative oder quantitative Analyse der Zielgruppen und der Anforderungen wird deshalb im Rahmen dieser Arbeit verzichtet.



## 2 Stöbern und die Integration des Digitalen ins Physische

Im Rahmen dieses Kapitels soll nach einer einführenden Beschreibung der Problematik des Medienbruchs eine Betrachtung bisheriger Konzepte zur Integration elektronischer Ressourcen beim Stöbern am Regal in Bibliotheken erfolgen.

### 2.1 Der Medienbruch online und am Regal

Bereits Mittler hat vor 25 Jahren den Medienbruch im Bereich der wissenschaftlichen Arbeit beschrieben als "de[n] ständigen Wechsel zwischen elektronischem und gedrucktem Text."<sup>15</sup> Heute findet sich der Medienbruch im Bereich der Bibliotheken in zweierlei Hinsicht:

a) bei der Online-Recherche:

Bei der Literaturrecherche von einem anderen Standort aus als der Bibliothek sind Printmedien nicht sofort verfügbar. Um das Medium vollständig einsehen zu können, muss dieses vor Ort in der Bibliothek eingesehen bzw. ausgeliehen werden. Eine zeit- und ortsunabhängige Verfügbarkeit ist hier nicht gegeben. Zum Teil wird versucht, diesem mittels Kataloganreicherung bspw. durch Inhaltsverzeichnisse oder Abstracts ein wenig entgegenzuwirken, um es Nutzern zu erleichtern, die Eignung des Buches für das eigene Thema besser abschätzen zu können. Wird jedoch die komplette Ressource benötigt, führt kein Weg an dem Besuch in der Bibliothek vorbei.

b) beim Stöbern am Regal:

E-Ressourcen<sup>16</sup> sind meist überhaupt nicht sichtbar. Die Suche nach diesen muss erneut im Katalog bzw. im Discovery-System durchgeführt werden. Auch entlehene Werke sind beim Stöbern am Regal nicht einsehbar, sodass hier nicht von einem umfassenden Bestandsüberblick gesprochen werden kann. Selbst Hinweise auf verfügbare digitale Ressourcen finden sich nur zum Teil.

Die Bedeutung des Stöberns am Regal/Browsings in Bibliotheken verdeutlicht unter anderem in einer Studie im Rahmen des *Project Information Literacy* in den USA aus dem Jahr 2009. Dabei wurde eine Benutzerumfrage mit dem Titel „*How College Students Seek Information in the Digital Age*“ zu unterschiedlichen Feldern der Informationssuche durchgeführt.<sup>17</sup> Mehr als die Hälfte der Befragten antworteten dabei auf die Frage, welche Angebote der Bibliothek sie bei der Literaturrecherche für Lehrveranstaltungen nutzen, dass sie die Möglichkeit nutzen, selbst am Regal zu stöbern (siehe Abb. 1).

---

<sup>15</sup> Vgl. Mittler (1996).

<sup>16</sup> Unter E-Ressourcen werden im Rahmen dieser Arbeit über das Internet abrufbare Veröffentlichungen mit unveränderlichem Textbestand, verstanden, welche sich wie Bücher zitieren lassen (Vgl.: Gantert (2008), S. 111). Darunter fallen neben monografisch erscheinenden (z.B. E-Books) auch fortlaufende elektronische Ressourcen wie Online-Zeitschriften.

<sup>17</sup> Vgl. Head; Eisenberg (2009): *Lessons Learned: How College Students Seek Information in the Digital Age*, [http://ctl.yale.edu/sites/default/files/basic-page-supplementary-materials-files/how\\_students\\_seek\\_information\\_in\\_the\\_digital\\_age.pdf](http://ctl.yale.edu/sites/default/files/basic-page-supplementary-materials-files/how_students_seek_information_in_the_digital_age.pdf), 03.05.2021.

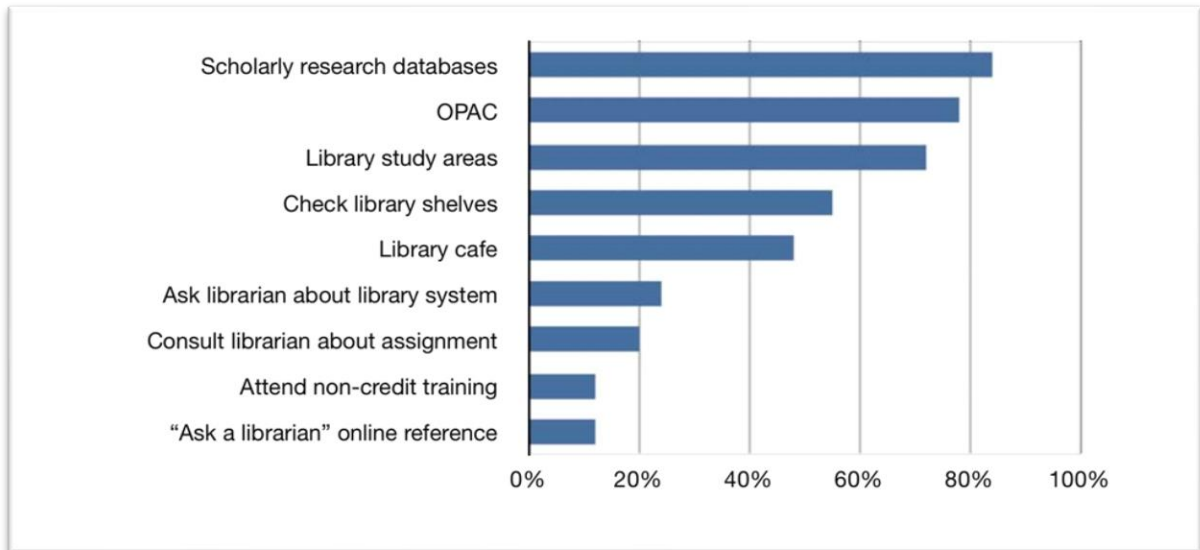


Abbildung 1: Nutzung von Bibliotheken für die Suche nach seminarrelevanter Literatur  
 Quelle: Head, Eisenberg (2009), S. 22 [FIGURE 10: Usage of Libraries for Course-Related Research]

Auch die Ergebnisse einer Umfrage im Rahmen des Projekts *Blended Library*, in dem „Wissenschaftler der Arbeitsgruppen Mensch-Computer Interaktion und Datenbanken & Informationssysteme am Standort Konstanz sowie dem Institut für Wissensmedien am Standort Tübingen den Einsatz, Einfluss und Mehrwert von neuartiger Informations- und Kommunikationstechnologie (IKT) im Bereich der Wissensarbeit [erforschen]“,<sup>18</sup> bestätigen dies. Hier gaben über 50% der Befragten an, dass Sie die Möglichkeit des Stöberns am Regal nutzen, um weiterführende Literatur ausfindig zu machen (siehe Abb. 2).

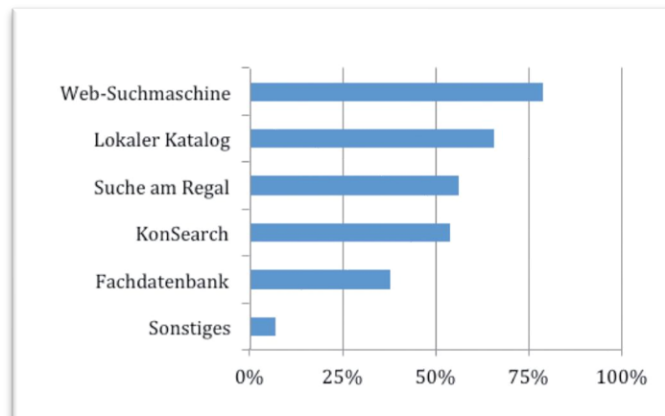


Abbildung 2: Nutzung der Recherchemittel für die Suche nach weiterführender Literatur  
 Quelle: Reiterer, Rädle et al (2016), S. 11 [Abb. 4: Auswertung der Frage „In welchen Katalogen/Suchmasken suchen Sie nach weiterführender Literatur?“]

<sup>18</sup> Vgl. Universität Konstanz (n. N.): Blended Library, <https://hci.uni-konstanz.de/forschung/forschungsprojekte/abgeschlossene-projekte/blended-library/>, 19.06.2021.

Die Tatsache, dass mehr und mehr Entwicklungen versuchen, den Vorgang des Stöberns am Regal bspw. mit Hilfe eines Blended Shelves<sup>19</sup> digital abzubilden, sowie die praktische Umsetzung dessen (bspw. im Katalog der Universität Konstanz, siehe Abb. 3), bestätigen die Bedeutung des Vorgangs für Bibliotheksnutzer.

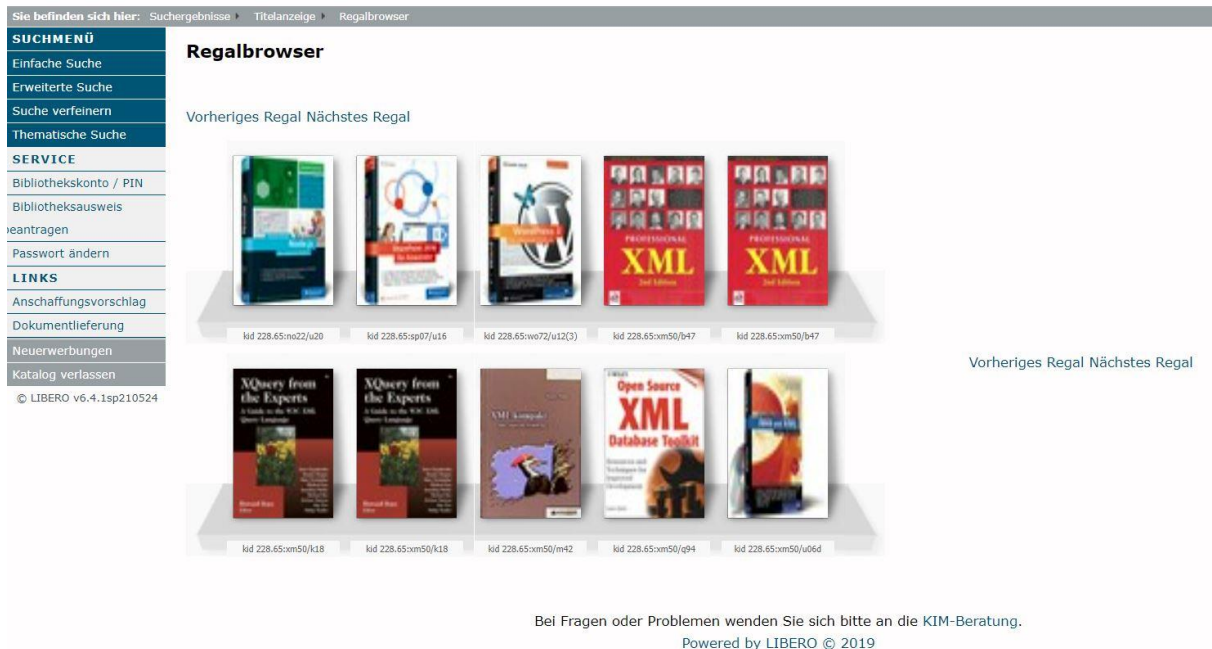


Abbildung 3: Blended Shelf im Katalog der Universität Konstanz

## 2.2 Lösungsansätze

Im Kontext der Thematik *Integration des Digitalen ins Physische* untersuchte Janin Taubert im Jahr 2013 im Rahmen ihrer Masterarbeit Möglichkeiten, den digitalen Angeboten der Bibliothek mehr Präsenz im physischen Raum der Bibliothek zu verleihen.<sup>20</sup> Dazu führte sie eine Literaturuntersuchung und mehrere qualitative Interviews sowohl mit Personen aus der Bibliotheksbranche als auch aus fachfremden Sparten durch. Als Ergebnis stellte sich unter anderem heraus, „dass die Erhöhung der Nutzung der elektronischen Medien und die Gewinnung neuer Nutzer für das E-Medien-Angebot [als Motivation für die Integration des Digitalen ins Physische] eine große Rolle spielen“<sup>21</sup> Ziel war hier also auch die Steigerung der Nutzungszahlen der E-Ressourcen. Darüber hinaus sollte über die Integration digitaler Bestände verhindert werden, dass Nutzer den Eindruck eines „mangelhafte[n] oder veraltete[n] Medienangebot[s]“<sup>22</sup> gewinnen. Es sollte ein „neuer Sucheinstieg in den digitalen Bestand“<sup>23</sup> geboten werden.

Vor allem betont wurde auch die Möglichkeit des Stöberns am Regal ohne zielgerichtetes Informationsbedürfnis, welches sich durch die Vor-Ort-Präsentation ergibt:

<sup>19</sup> Vgl. Kleiner (2013).

<sup>20</sup> Vgl. Taubert (2012).

<sup>21</sup> Ebd., S. 58.

<sup>22</sup> Ebd.

<sup>23</sup> Ebd., S. 14.

*„Im Vergleich zur Suche nach Medien und Informationen im Internet wird die Stärke von Vor-Ort-Präsentationen – egal ob im Buchladen oder in der [...] Bibliothek – gerade darin gesehen, dass man ohne ein bestimmtes, vorstrukturiertes Informationsinteresse im Angebot stöbern kann.“<sup>24</sup>*

Schließlich sei es auch die „Immaterialität der E-Medien [, die] nach einer erhöhten Präsenz in der Vermittlung des Angebots [verlangt].“<sup>25</sup>

Als Gefahr wurde hingegen die „Imitation des Physischen und dessen unreflektierte Übertragung auf die Präsentation digitaler Medien“<sup>26</sup> angesehen. Als Nachteil wurde weiterhin der zusätzliche Aufwand genannt.<sup>27</sup> Dies gilt es im Rahmen der Erstellung des Konzepts zu berücksichtigen.

Die Studie ergab, dass viele Bibliotheken physische Platzhalter oder Attrappen nutzen, um bspw. auf das parallele digitale Angebot einer Zeitschrift hinzuweisen.<sup>28</sup> Bei diesen handelt es sich zwar um eine schnell umzusetzende Möglichkeit, welche jedoch zu Platzproblemen führen kann, da die Platzhalter separaten Raum im Regal benötigen. Darüber hinaus ist der Weg zum eigentlichen digitalen Inhalt umständlich und langwierig.<sup>29</sup>

Als weitere Lösungsmöglichkeit stellten sich QR-Codes heraus, welche im Gegensatz zu physischen Platzhalten und Attrappen einen „schnelle[n] und direkte[n] Zugriff auf die Inhalte“<sup>30</sup> ermöglichen. Eine separate Eingabe einer URL ist nicht notwendig. Durch Scannen des Codes mittels Handy-Kamera gelangt man direkt zum Angebot. Darüber hinaus lassen sich QR-Codes schnell und einfach umsetzen und erfordern nur wenig Arbeitsaufwand und geringe Kosten, da Software zum Generieren von QR-Code kostenfrei zur Verfügung steht und die Materialkosten sehr gering sind.<sup>31</sup> In der Regel verwiesen die QR-Codes direkt auf den Volltext der parallelen digitale Ausgabe eines Buches oder auf Übersichtsseiten einer Zeitschrift. Als problematisch wurde der Aspekt genannt, dass nicht alle Anbieter ihre Inhalte für die mobile Nutzung optimiert bereitstellen.<sup>32</sup> Zu beachten sei außerdem, dass die angebrachten QR-Codes eine bestimmte Mindestgröße aufweisen müssen. Darüber hinaus wurde angemerkt, dass nicht für jede digitale Printausgabe ein QR-Code am Regal angebracht werden könne, da es sonst zu unübersichtlich werde.<sup>33</sup>

Auf weitere in der Arbeit von Frau Taubert vorgestellte Möglichkeiten der Integration des Digitalen ins Physische soll an dieser Stelle nicht weiter eingegangen werden, da sie das Stöbern im Regal nicht betreffen.

Die Bibliothek der Georg-Simon-Ohm-Hochschule in Nürnberg verfolgte einen weiteren Ansatz mit QR-Codes: Die dort verwendeten QR-Codes verweisen nicht auf ein bestimmtes E-Book, sondern auf eine

---

<sup>24</sup> Vgl. Taubert (2012): Absentia in Praesentia? Zur Präsentation und Vermittlung digitaler Medien im physischen Raum, S. 11.

<sup>25</sup> Vgl. Hommes (2009): Vandalen haben keine Chance mehr. In: *BuB*, Jg. 61, Nr. 09, S. 620–623, S. 623. Zitiert nach Taubert (2012), S. 15

<sup>26</sup> Vgl. Taubert (2012), S. 16.

<sup>27</sup> Ebd., S. 17.

<sup>28</sup> Ebd., S. 18.

<sup>29</sup> Vgl. Taubert (2016), Folie 9.

<sup>30</sup> Ebd., Folie 12.

<sup>31</sup> Vgl. Taubert (2012), S. 27.

<sup>32</sup> Vgl. Taubert (2016), Folie 12.

<sup>33</sup> Vgl. Taubert (2012), S. 25.

Gruppe systematisch zusammengefasster E-Books.<sup>34</sup> Die QR-Codes befinden sich hier nicht direkt am Regal, sondern „an der Regalstirnseite passend zur jeweiligen Systemstelle“<sup>35</sup>

*„Voraussetzung ist hier, dass man die jeweilige URL hat, die alle im OPAC mit einer bestimmten Notation erschlossenen E-Medien aufruft. Dies wiederum setzt voraus, dass man E-Books systematisch erschließt.“<sup>36</sup>*

Zu beachten sei hier der zusätzliche Aufwand durch die systematische Erschließung der Medien mit der verwendeten Klassifikation.<sup>37</sup> Problematisch ist an dieser Stelle der „Trend zur Übernahme von Fremddaten, die keine klassifikatorische Erschließung vorsehen.“<sup>38</sup>

In der Bibliothek der Universität Bern gibt es darüber hinaus das Projekt cUBe („see You (cU) in der Universitätsbibliothek (UB) Bern (Be)!“),<sup>39</sup> welches „physische und digitale Bibliotheksräume“<sup>40</sup> verbindet. Dabei weisen rote Würfel auf das physische Web, also „zum analogen Angebot im Raum passende digitale Angebote“<sup>41</sup>, hin.

*„Der standort- bzw. kontextbasierte Zugang zu Online-Angeboten kann über QR-Code, Kurz-URL oder Beacon geschehen. Während QR-Codes und Kurz-URLs gescannt bzw. abgetippt werden müssen, werden die Signale der Beacons, d.h. der Bluetooth-Miniatursender, einfach über eine App empfangen.“<sup>42</sup>*

Darüber hinaus werden QR-Codes auch in Online-Katalogen verwendet, damit sich Nutzer die Informationen zum Medium unkompliziert auf ihrem Smartphone anzeigen lassen können, so bspw. in der Universitätsbibliothek Bayreuth. Mittels QR-Code können Informationen zu Standort, Signatur, Titel und Verfasser als Textinformation an das Smartphone übertragen werden.<sup>43</sup> An der Bibliothek der Technischen Universität Wien verweisen QR-Codes auf Neuerscheinungslisten aus dem jeweiligen Fachbereich.<sup>44</sup>

Der Einsatz von QR-Codes, um auf digitale Parallelausgaben hinzuweisen, zeichnet sich bis heute fort.<sup>45</sup> So werden bspw. in der Universitäts- und Landesbibliothek (ULB) Düsseldorf zu diesem Zweck (zunächst für ca 40 Titel im Bereich der Lehrbuchsammlung der naturwissenschaftlichen Fächer) Buchstützen mit Titelfotos und QR-Codes verwendet. An der Bibliothek der Freien Universität (FU) Berlin werden QR-Codes (auf Vertretern) genutzt, um in Ausstellungsregalen auf aktuelle Online-Ressourcen hinzuweisen. Auf die Umfrage über die Mailingliste Inetbib hin mehrfach genannt wurde auch die Nutzung von (mittels Makro erzeugten) QR-Codes, um auf Parallelausgaben von Printzeitschriften hinzuweisen. Die QR-Codes enthielten z.B. Verlinkungen zum Anbieter oder zur Elektronischen Zeitschriftenbibliothek. Hier wurde ebenfalls geäußert, dass ein Problem bei der Umsetzung die Art der Sacherschließung ist. Weitere

---

<sup>34</sup> Vgl. Ackermann (2011).

<sup>35</sup> Vgl. Taubert (2012), S. 26.

<sup>36</sup> Ebd.

<sup>37</sup> Ebd.

<sup>38</sup> Ebd.

<sup>39</sup> Vgl. Universität Bern (n. N.): cUBe · Online-Informationen erkunden, [https://www.ub.unibe.ch/ub/cube/index\\_ger.html](https://www.ub.unibe.ch/ub/cube/index_ger.html), 20.06.2021.

<sup>40</sup> Ebd.

<sup>41</sup> Ebd.

<sup>42</sup> Ebd.

<sup>43</sup> Vgl. Universitätsbibliothek Bayreuth (n. N.): Neu im Katalog: Anzeige von QR-Codes, [https://www.ub.uni-bayreuth.de/de/news\\_archiv/neuerungen-IG-6/index.html](https://www.ub.uni-bayreuth.de/de/news_archiv/neuerungen-IG-6/index.html), 20.06.2021.

<sup>44</sup> Vgl. Slama (2012): QR-Code-Hinweise in der Lehrbuchsammlung, <https://www.tuwien.at/tuwien/aktuelles/news/news/qr-code-hinweise-in-der-lehrbuchsammlung>, 20.06.2021.

<sup>45</sup> Für den folgenden Absatz siehe Ergebnisse des Etherpads, (siehe Anhang A.1)

Anwendungsgebiete von QR-Codes in deutschsprachigen Bibliotheken finden sich in einer von Viola Voss zusammengestellten Liste.<sup>46</sup>

Zusammenfassend haben sowohl die Arbeit von Frau Taubert, als auch die eigene Umfrage über die Mailingliste gezeigt, dass viele Lösungsansätze zur Überwindung des Medienbruchs beim Stöbern am Regal vorhanden sind. Eine wichtige Rolle spielen dabei QR-Codes, um schnell und unmittelbar auf ein Medium zugreifen zu können, ohne eine URL abtippen zu müssen. Darüber hinaus sind diese schnell und kostengünstig zu erstellen. Häufig werden QR-Codes genutzt, um auf parallele digitale Zeitschriften oder E-Book-Angebote aufmerksam zu machen. Als besonderes Beispiel wurde die Bibliothek der Georg-Simon-Ohm-Hochschule in Nürnberg beschrieben. Hier werden QR-Codes nicht direkt an der Regalfrontseite angebracht, sondern an der Systematikübersicht an der Stirnseite. Verwiesen wird hier nicht auf einen bestimmten Titel, sondern über die Systematik auf thematisch passende E-Books. Als wesentliche Voraussetzung wurde genannt, dass die entsprechenden Ressourcen systematisch erschlossen sein müssen.

---

<sup>46</sup> Vgl. Voß (2011): QR-Codes im bibliothekarischen Einsatz, <https://netbib.hypotheses.org/78620092>.

## 3 Bibliothekarische und technische Vorüberlegungen

Nachdem bisher Lösungsansätze verschiedener Bibliotheken zur Problematik skizziert wurden, sollen in den folgenden Abschnitten einige bibliothekarische und technische Aspekte genauer betrachtet werden. Stellte sich im vorigen Kapitel heraus, dass zum Zweck der Standortermittlung des Nutzers hauptsächlich mit QR-Codes gearbeitet wird, soll nun untersucht werden, ob ggf. auch andere Methoden genutzt werden könnten. Dazu soll ein Blick auf die Standortermittlung von Nutzern in Bibliotheken geworfen und betrachtet werden, inwiefern sich diese Methoden für die zu erstellende Anwendung eignen könnten. Weiterhin sollen Methoden zur Bestandsaufstellungen in (wissenschaftlichen) Bibliotheken skizziert werden, um herauszufinden, welche Voraussetzungen für die Überwindung des Medienbruchs am Regal diesbezüglich bestehen. Die letzten Abschnitte des Kapitels beschäftigen sich mit technischen Vorüberlegungen. Zunächst sollen Formate von e-Ressourcen in Bibliotheken und deren Anzeige auf mobilen Geräten skizziert werden. Anschließend sollen verschiedene Arten von Apps dargestellt sowie mögliche Frameworks zur Umsetzung betrachtet werden.

### 3.1 Standortermittlung von Nutzern in Bibliotheken

*„Insgesamt ist der Ruf nach einem funktionierenden und nicht zu kostenintensiven Indoor-Ortungssystem für Bibliotheken, das durch ihre Besucher auf dem eigenen Smartphone genutzt werden kann und die eben erwähnte Orientierung unterstützt, allgegenwärtig.“<sup>47</sup>*

GPS-Outdoor-Systeme sind mittlerweile nahezu eine Selbstverständlichkeit für Smartphone-Nutzer, lassen sich jedoch in Gebäuden nicht so sicher und genau nutzen.<sup>48</sup> Doch welche Möglichkeiten der Ortung eines Nutzers in einer Bibliothek gibt es und inwiefern können diese ggf. dabei helfen, dem Nutzer thematisch passend zum Regalstandort zusätzliche e-Ressourcen anzubieten?

Werden Standorte gedruckter Medien in Lesesälen von Bibliotheken mehr und mehr in virtualisierten Orientierungssystemen wie V:Scout<sup>49</sup> oder Mapongo<sup>50</sup> abgebildet und in die Recherchertools von Bibliotheken eingebunden, gibt es für die Standorterkennung von Nutzern bisher wenige Ansätze.

#### 3.1.1 iBeacons

In der Hochschulbibliothek der TH Wildau werden zur Indoor-Standorterkennung von Nutzern iBeacons verwendet. Dabei handelt es sich um eine von Apple entwickelte, auf Bluetooth Low Energy (BLE) basierende Sender-Empfänger-Technologie. Als Sender dienen dabei die sogenannten iBeacons, welche regelmäßig Signale aussenden (hier die eindeutige ID des Senders). Als Empfänger kann jedes BLE-fähige

---

<sup>47</sup> Vgl. Azmitia; Mohnke; Wiechers (2016): Wo bin ich? iBeacons im Einsatz in der Bibliothek der TH Wildau. In: *Bibliothek - Forschung und Praxis*, Jg. 40, Nr. 3, S. 395–404, S. 396.

<sup>48</sup> Ebd., S. 395.

<sup>49</sup> Vgl. ARTEC (n. N.): V:SCOUT: Die innovative und visuelle Orientierungshilfe für Ihre Besucher & Mitarbeiter, <https://vscout.biz/>, 20.06.2021.

<sup>50</sup> Vgl. mapongo (n. N.): mapongo Gebäudeinformationssysteme, <http://www.mapongo.de/libraries>, 20.06.2021.

Smartphone genutzt werden (iOS- oder androidbasiert). Die iBeacons werden gezielt in den Räumlichkeiten der Bibliothek verteilt.

*„Die genaue Position jedes Beacons wird, zusammen mit einer eindeutigen Identifikationsnummer (kurz ID), in einem zum System gehörenden Softwaresystem verwaltet. [...] Mit der Annahme, dass ein Empfänger nur Signale von Beacons in seiner Nähe empfängt, können diese Informationen zur Berechnung der Position des Empfängers im Raum genutzt werden.“<sup>51</sup>*

Zur Überwindung des Medienbruchs beim Stöbern am Regal würden darüber hinaus genaue Standortinformationen zu jedem Regal benötigt werden. Diese müssten zudem mit Daten darüber, welche Themenbereiche im Regal untergebracht sind, angereichert werden. Zusammen mit diesen Informationen könnte ein Abgleich der Standorte stattfinden und dem Nutzer die passenden Ressourcen angezeigt werden.

Zum Problem werden an dieser Stelle jedoch nah beieinander- oder gegenüberstehende Regale. Steht man genau mittig zwischen zwei Regalen empfängt man möglicherweise die IDs von zwei iBeacons und würde somit ggf. thematisch unpassende E-Ressourcen des gegenüberstehenden Regals angezeigt bekommen. Auch einzelne Böden eines Regals ließen sich mit der iBeacon-Technologie nicht abbilden lassen. Nicht zu vernachlässigen sind die für die Einrichtung eines solchen Systems benötigten Vorarbeiten. Hier zu nennen ist einerseits die korrekte Konfiguration (Feinabstimmung der Signale) der iBeacons, aber auch das Ermitteln der Standortdaten zu den einzelnen Regalen.

### 3.1.2 RFID bzw. NFC

Eine weitere Idee stellt die Nutzung einer anderen Sender-Empfänger-Technologie dar, die bereits in Bibliotheken genutzt wird: RFID (Radio Frequency Identification). Ein RFID-System umfasst zwei Komponenten: ein Transponder (hier der RFID-Tag), welcher sich am zu erkennenden Objekt befindet und ein Lesegerät. Auf dem Transponder befinden sich gespeicherte Informationen.<sup>52</sup> Sobald dieser nun in den Empfangsbereich des Lesegeräts gelangt, können die Daten auf dem Transponder über Radiowellen (Hochfrequenz, 13,56 MHz) gelesen werden.<sup>53</sup>

Können RFID-Tags in einem Bereich von ca 30 – 50 cm gelesen werden,<sup>54</sup> liegt die Reichweite bei NFC (Near Field Communication) bei maximal zehn cm.<sup>55</sup> Die im Grunde auf RFID-basierende NFC-Technologie findet vor allem auf neuen Smartphones Anwendung.<sup>56</sup> Bereits Plappert machte 2016 auf die Möglichkeiten aufmerksam:

*„So können sich [die Nutzer] über ihre NFC-fähigen Smartphones und Tablets sowie weitere mobile Endgeräte zukünftig in der Bibliothek authentifizieren, Bücher direkt am Regal ausleihen sowie Informationen*

---

<sup>51</sup> Vgl. Azmitia; Mohnke; Wiechers (2016), S. 399.

<sup>52</sup> Vgl. Kern (2011): RFID für Bibliotheken, S. 3.

<sup>53</sup> Für eine weitergehende Einführung in den RFID-Einsatz in Bibliotheken siehe Kern (2011): RFID für Bibliotheken.

<sup>54</sup> Ebd., S. 11.

<sup>55</sup> Vgl. Bauer (2016): NFC: Reichweite und Geschwindigkeit, [https://praxistipps.chip.de/nfc-reichweite-und-geschwindigkeit\\_48444](https://praxistipps.chip.de/nfc-reichweite-und-geschwindigkeit_48444), 27.06.2021.

<sup>56</sup> Vgl. Krautz (2014): Einflüsse von NFC-Smartphones auf das RFID-Bibliothekssystem. In: Seeliger, F. et al. (Hg.), S. 209–221, S. 209.



*zum gewünschten Medium aus dem Internet abrufen oder die RFID-Technologie als Orientierungs- und Leitsystem innerhalb der Bibliothek nutzen.“<sup>57</sup>*

So könnte die NFC-Technologie genutzt werden, Informationen zum Inhalt eines Mediums, welche bspw. in Form einer Notation auf dem Chip gespeichert werden, mit dem Smartphone auszulesen. Derzeit werden Informationen, welche auf RFID-Tags gespeichert werden, in Form des Dänischen Datenmodells gespeichert. Dieses beinhaltet exemplarspezifische Informationen wie Ausleihstatus, Barcode etc. Titelspezifische Informationen zum Inhalt eines Mediums liegen dabei jedoch nicht direkt vor. Diese müssten zunächst über ein separates Modul aufwändig ermittelt und verarbeitet werden.

### 3.1.3 Mobile Tagging mit QR-Codes

Eine dritte Möglichkeit, auf die an dieser Stelle eingegangen werden soll, stellt das *mobile Tagging* dar, welches auf „die Interaktion zwischen einem Nutzer, einem mobilen Gerät und einem smart object“<sup>58</sup> abzielt. Dabei spielen drei Komponenten eine Rolle:

- „maschinenlesbare Daten, die an Objekten angebracht werden (tags)
- das Arbeiten mit den Tags und den in ihnen enthaltenen Daten (Kodieren, Auslesen, Weiterverarbeiten, tagging) [...]
- Ortsunabhängigkeit sowohl für die Tags als auch für das Lesegerät (mobile)<sup>59</sup>

Im Falle optischer Codes bedeutet dies, dass anstelle des mühsamen und fehleranfälligen Abtippens einer URL lediglich ein Programm zum Tag-Auslesen gestartet und der Tag abfotografiert werden müsste. Das Programm öffnet anschließend bspw. den Browser mit der im Tag codierten URL.

Als Vorteile sind an dieser Stelle vor allem die Einfachheit und die geringen Kosten zu nennen: Für die Erstellung der Codes stehen zahlreiche (auch kostenfreie) Angebote im Internet bereit. Die erstellten Codes müssen lediglich ausgedruckt werden. Auf Anwenderseite wird nur ein mobiles Endgerät mit Kamera benötigt.<sup>60</sup>

Im Bereich der optischen Codes werden vier Typen (nach Art der Codierung der Informationen) unterschieden (siehe Tabelle 1):

*„Während in eindimensionalen Codes, den „altbekanntem“ Strichcodes, Informationen in einer Richtung (also in einer Dimension) verschlüsselt werden, werden für 2D-Codes ähnlich einem Koordinaten-System zwei Dimensionen zur Informationsspeicherung genutzt. Wenn man diese zwei Dimensionen um den Faktor Farbe als dritte und den Faktor Zeit als vierte Dimension ergänzt, ergeben sich 3D- bzw. 4D-Codes. Mit höherer Dimensionszahl können mehr Daten kodiert werden.“<sup>61</sup>*

---

<sup>57</sup> Vgl. Plappert (2016): RFID in Bibliotheken. Eine erste Bilanz und zukünftige Anwendungsoptionen. In: *Bibliotheksdienst*, Jg. 50, Nr. 6, S. 531–534, S. 534.

<sup>58</sup> Vgl. Voß (2011): Mobile Tagging. In: *B.I.T.online*, Jg. 14, Nr. 4, S. 347–353, S. 347.

<sup>59</sup> Ebd.

<sup>60</sup> Ebd., S. 348.

<sup>61</sup> Ebd.














Typ	1D-Codes	2D-Codes		3D-Codes	4D-Codes
Kodierung	eindimensional	zweidimensional		2D + Farbe	3D + Zeit
Beispiele	Codabar, Code39, Code128, EAN, Interleave, Pharmacode, Zielcode	<i>Stapelcodes:</i> Codablock-F, PDF147 <i>Matrixcodes:</i> Aztec, BeeTagg, Blotcode, DataMatrix, Maxicode, JagTag, QR-Code <i>Sonderformen:</i> Shotcode, Array Tag, 3-DI /Accu-Code, Dandelion Code		Microsoft Tag	unsynchronisierter 4D-Barcode
	Code128	Codablock-F	PDF147	Microsoft Tag	unsynchronis. 4D-Barcode
					
	EAN	Aztec	BeeTagg	QR-Code	
					
	Pharmacode Zweispur	DataMatrix	Maxicode	Shotcode	
					

Tabelle 1: Mobile Tagging - Übersicht  
 Quelle: Voß (2011), S. 348 [Tabelle 1: Optische Codes]

Im Bereich des optischen Mobile-Tagging finden überwiegend 2D-Codes Anwendung, da die Kapazität dieser ausreichend für die zu speichernde Datenmenge ist. Außerdem steht für die Erstellung dieser 2D-Codes lizenzfreie Software zur Verfügung.<sup>62</sup> Hier sind es vor allem QR-Codes, die Anwendung finden. Die Abkürzung QR steht dabei für *quick response* und bezieht sich auf die schnelle Auslesbarkeit des Codes.<sup>63</sup> Als Charakteristikum für diese gelten die drei sich in den Ecken befindenden Quadrate. Bezüglich der Speicherung der Informationen wird zwischen direkten und indirekten Codes unterschieden. Während auf den direkten Codes die eigentlichen Informationen gespeichert werden, enthalten indirekte Codes lediglich eine ID, welche zunächst (bspw. mit Hilfe einer Datenbank) aufgelöst werden muss, um schließlich zur eigentlichen Information zu gelangen.<sup>64</sup> QR-Codes selbst können mehrere Tausend Zeichen umfassen und eignen sich somit bspw. zum Abbilden von alphanumerischen Texten, URLs oder digitalen Visitenkarten.<sup>65</sup>

Bei der Anwendung von QR-Codes ist vor allem auf eine ausreichende Größe zu achten: Zur Mindestgröße schreibt Voß bspw., dass „die gewünschte Scan-Distanz geteilt durch zehn“<sup>66</sup> empfohlen wird.

QR-Code-Reader stehen mittlerweile (zumeist kostenfrei) für alle gängigen Betriebssysteme mobiler Geräte zur Verfügung.<sup>67</sup>

<sup>62</sup> Ebd.  
<sup>63</sup> Ebd., S. 349.  
<sup>64</sup> Ebd., S. 348.  
<sup>65</sup> Ebd., S. 349.  
<sup>66</sup> Ebd., S. 350.  
<sup>67</sup> Ebd.

### 3.1.4 Empfehlung

Nach genauerer Betrachtung verschiedener derzeit genutzter Möglichkeiten der Standorterkennung von Nutzern in Bibliotheken und deren Anwendbarkeit für die zu entwickelnde App lässt sich sagen, dass sich QR-Codes aufgrund der vergleichsweise einfachen und vor allem kostengünstigen Umsetzung und auch unkomplizierten Anwendung am besten eignen.

Sind iBeacons zwar sehr gut nutzbar, um den Standort eines Nutzers in der Bibliothek abzubilden, wären diese jedoch für den Zweck der App nur mit einem nicht zu vernachlässigendem Einrichtungsaufwand umsetzbar. Darüber hinaus wäre eine Abbildung verschiedener Regalebenen bereits nicht mehr möglich. Auch der Einsatz der NFC-Technologie wäre nicht praktikabel, da die benötigten Informationen über Medieninhalte nicht auf den verwendeten Tags enthalten sind. Der Haupteinsatzzweck der RFID-Technologie liegt schließlich im Bereich der Ausleihe.

## 3.2 Sacherschließung und Aufstellung in Bibliotheken

Nachdem sich im vorhergehenden Abschnitt herausgestellt hat, dass QR-Codes am besten geeignet sind, um den Standort des Lesers hinsichtlich der zu entwickelnden App zu ermitteln, soll nun untersucht werden, welche Daten genutzt werden können, um eine Verbindung zwischen Standort des Lesers und dem dort untergebrachten Themenbereich herzustellen. Dazu sollen im folgenden Abschnitt sowohl die Erschließung als auch die Aufstellung in Bibliotheken beschrieben werden. Vor allem die Erschließung elektronischer Ressourcen im Vergleich zur Erschließung gedruckter Bestände soll betrachtet werden, um mögliche Probleme frühzeitig zu erkennen.

### 3.2.1 Sacherschließung

Um ihren Bestand für ihre Nutzer bestmöglich zugänglich zu machen, erschließen Bibliotheken diesen formal und sachlich.

*„Durch die formale Katalogisierung der [Bestände] einer Bibliothek sollen Antworten auf die drei folgenden Fragen ermöglicht werden:*

- (1) Ist ein bestimmtes Werk in der Bibliothek vorhanden?*
- (2) Welche Werke eines bestimmten Verfassers oder Urhebers besitzt die Bibliothek?*
- (3) Welche verschiedenen Ausgaben eines bestimmten Werkes besitzt die Bibliothek?“<sup>68</sup>*

Hier geht es also um die Beschreibung anhand formaler Elemente wie bspw. Verfassernamen oder Titel.

Bei der Sacherschließung hingegen ist das Ziel die „inhaltliche Beschreibung und Erschließung“.<sup>69</sup> Dabei wird zwischen zwei Verfahren unterschieden: Werden bei der verbalen Sacherschließung natürlichsprachliche Bezeichnungen (wie Stich- und Schlagwörter, z.T. auch sogenannte syntaktische Verfahren wie Abstracts) verwendet, wird im Bereich der klassifikatorischen Sacherschließung überwiegend mit Klassifikationssystemen, welche auf hierarchisch geordneten Systemstellen (Notationen) beruhen, gearbeitet.<sup>70</sup>

---

<sup>68</sup> Vgl. Gantert; Hacker (2008): Bibliothekarisches Grundwissen, S. 165.

<sup>69</sup> Ebd., S. 177.

<sup>70</sup> Ebd.

„Während die verbale Sacherschließung vor allem eine „punktueller“ Suche nach Publikationen zu einem begrenzten, genau definierten Thema ermöglicht, weisen Klassifikationen die Veröffentlichungen im Kontext ihres Fachgebietes nach.“<sup>71</sup>

Als Vorarbeit für das zu entwickelnde Konzept lohnt es sich also, vor allem den Bereich der klassifikatorischen Sacherschließung noch einmal genauer zu betrachten, um entscheiden zu können, auf Grundlage welcher Informationen dem Nutzer passende Ressourcen angezeigt werden sollen. Der Vorteil beim Stöbern am Regal liegt ja schließlich darin, zufällig neue interessante Funde zu machen, die zum Fachgebiet gehören.

Jede Publikation wird bei der Erschließung einer oder mehrere Klasse/n oder Systemstelle/n zugeordnet. Inhaltlich zusammengehörige Publikationen sind über die gleiche Notation auffindbar.

Ein Beispiel für eine in vielen deutschen Hochschulbibliotheken auch als Aufstellungssystematik verwendete Systematik ist die Regensburger Verbundklassifikation (RVK), welche in den 1960er Jahren an der neu gegründeten Universitätsbibliothek Regensburg entwickelt wurde.<sup>72</sup> Mittlerweile wird sie von über 140 Anwendern in Deutschland, Österreich, Italien, Liechtenstein und der Schweiz genutzt.<sup>73</sup>

Bei der RVK handelt es sich um eine alphanumerische Klassifikation, welche sich aus zwei Großbuchstaben für Gruppe und Untergruppe und einer drei- bis sechsstelligen Zahl, welche Schlüssel für zeitliche/regionale Gliederung, Form, Literaturtyp u.Ä. sind, zusammensetzt.

Am Beispiel der Notation ST 250 baut sich diese folgendermaßen auf:<sup>74</sup>

<i>SQ – SU</i>	<i>Informatik</i>
<i>ST</i>	<i>Monografien</i>
<i>ST 230 – ST 285</i>	<i>Software und -entwicklung</i>
<i>ST 240 – ST 250</i>	<i>Programmiersprachen</i>
<i>ST 250</i>	<i>Einzelne Programmiersprachen (A-Z)</i>

Dabei sind einige Teilsystematiken tiefer gegliedert als andere. Wird die RVK als Aufstellungssystematik genutzt, setzt sie sich aus drei Teilen zusammen: Lokalkennzeichen, Notation und individualisierender Teil (wie bspw. Angaben zu Verfasser/Titel/Thema, Band, Auflage, Exemplar mit Hilfe der Cutter-Sanborn-Notation/-Nummer (CSN)).<sup>75</sup>

Eine vor allem im Bereich der Naturwissenschaften weltweit eingesetzte Notation ist die Dewey Decimal Classification (DDC),<sup>76</sup> bei welcher die Notationen lediglich aus Ziffern bestehen. Eine Einteilung des

---

<sup>71</sup> Ebd.

<sup>72</sup> Ebd., S. 190.

<sup>73</sup> Vgl. Universitätsbibliothek Regensburg (n. N.): Was ist die RVK?, <https://rvk.uni-regensburg.de/2-uncategorised/141-rvk>, 27.06.2021.

<sup>74</sup> Vgl. Universitätsbibliothek Regensburg (n. N.): RVK Online, <https://rvk.uni-regensburg.de/regensburger-verbundklassifikation-online>, 27.06.2021.

<sup>75</sup> Vgl. Bayreuther; Peisl (2012): Regeln für die Signaturvergabe der UB Regensburg, [https://rvk.uni-regensburg.de/images/stories/fruit/sigreg/regeln\\_fr\\_die\\_signaturvergabe\\_version\\_2012\\_02\\_17.pdf](https://rvk.uni-regensburg.de/images/stories/fruit/sigreg/regeln_fr_die_signaturvergabe_version_2012_02_17.pdf), 14.08.2021. S. 2.

<sup>76</sup> Vgl. Gantert; Hacker (2008), S. 189.

menschlichen Wissens erfolgt hierbei jeweils in Zehnergruppen. Es gibt also zehn Hauptgruppen (bezeichnet mit 0 – 9), welche durch Hinzufügen einer weiteren Ziffer (0 – 9) wiederum in Untergruppen zerteilt wird.

Die Klassen der dritten Ebene sehen hier zum Beispiel für Programmiersprachen folgendermaßen aus:<sup>77</sup>

0	<i>Informatik, Informationswissenschaft &amp; allgemeine Werke</i>
00	<i>Informatik, Wissen und Systeme</i>
005	<i>Computerprogrammierung, Computerprogramme &amp; Daten</i>

Nach genauerer Betrachtung der verbalen und der klassifikatorischen Sacherschließung kann man sagen, dass sich grundsätzlich beide Typen eignen, um dem Nutzer zum Thema passende Bestände anzeigen zu lassen. Wichtig an dieser Stelle ist es jedoch, zu betrachten, inwiefern elektronische Ressourcen über entsprechende Informationen im Titeldatensatz verfügen.

Gerade E-Books werden häufig in größeren Paketen gekauft werden und somit nicht einzeln von Hand katalogisiert. Hier werden also nicht wie bei gedruckten Medien per Autopsie manuell Schlagworte oder Klassifikationen vergeben, sondern es werden die vom Anbieter mitgelieferten Daten übernommen. Eine manuelle Ergänzung der Daten geschieht hier meist nicht mehr. Durch Verfahren wie Patron Driven Acquisition (PDA), bei dem „der Beschaffungsentscheid dem Bibliotheksnutzer übertragen [wird]“<sup>78</sup> werden herkömmliche Geschäftsgänge zum Teil umgangen.

Problematisch bei der Erschließung von elektronischen Ressourcen im Vergleich zu Printmedien ist die Qualität der durch den Anbieter mitgelieferten Metadaten:

*„Von Sacherschließungsdaten, wie sie in vielen deutschen, meist wissenschaftlichen Bibliotheken genutzt werden (z.B. Schlagwörter nach der Schlagwortnormdatei (SWD) oder Notationen der Regensburger Verbundklassifikation (RVK)) ist meistens keine Rede, höchstens Daten der Dewey Decimal Classification (DDC) können erwartet werden.“<sup>79</sup>*

An der Bibliothek der Technischen Hochschule Wildau verfügen derzeit lediglich ca 30% der E-Books über eine klassifikatorische Erschließung in Form von RVK-Notationen.<sup>80</sup>

Auch diese Tatsache muss bei der Entwicklung des Konzepts beachtet werden.

---

<sup>77</sup> Vgl. DNB (n. N.): DDC-Übersichten, [<sup>78</sup> Vgl. Mumenthaler \(2018\): E-Books, <https://ebooksgrundlagen.pressbooks.com/>, 18.06.2021.](https://www.dnb.de/DE/Professionell/DDC-Deutsch/DDCuebersichten/ddcUebersichten.html#:~:text=Die%20Dewey%20Dezimalklassifikation%20(DDC),in%20die%20Grundstruktur%20der%20DDC%20., 27.06.2021.</a></p></div><div data-bbox=)

<sup>79</sup> Vgl. Rüter (2009): Elektronische Ressourcen im Geschäftsgang, S. 11.

<sup>80</sup> Diese Zahl äußerte Frau Keidel (Stellvertretende Leiterin der Bibliothek, zuständig für Erwerbung, Bestandserschließung und Management elektronischer Ressourcen) beim Kick-Off-Treffen zum Projekt am 13.11.2020, siehe Anhang A.1

### 3.2.2 Anforderungen an die Aufstellung in wissenschaftlichen Bibliotheken

Um den Nutzern einer Bibliothek ein Stöbern am Regal zu ermöglichen, sind an die Aufstellung des Bestands bestimmte Bedingungen geknüpft, welche im folgenden Abschnitt dargestellt werden.

In Hochschulbibliotheken wird allgemein das Ziel verfolgt, einen Großteil des Bestandes den Nutzern in Form einer Freihandaufstellung zur Verfügung zu stellen, „um so den unmittelbaren Kontakt zwischen Lesern und Medien am Standort zu ermöglichen.“<sup>81</sup> Damit ist eine der wesentlichen Grundvoraussetzungen für das Stöbern am Regal erfüllt. Gibt es darüber hinaus für den Nutzer nicht zugängliche Magazinbestände, könnten Überlegungen angestellt werden, diese in die App zu integrieren, damit der Nutzer sich diese anzeigen lassen kann.

Neben der Zugänglichkeit spielt aber auch die Anordnung der Bestände eine Rolle: Voraussetzung fürs Stöbern am Regal ist, dass thematisch zusammengehörige Publikationen zusammenstehen.

In Bibliotheken kann grob zwischen systematischer und mechanischer Aufstellung unterschieden werden. Erfolgt die mechanische Aufstellung (*numerus currens*) nach dem Eingang der Medien in den Bestand der Bibliothek in zufälliger Reihenfolge ohne Rücksicht auf den Inhalt, erfolgt bei der systematischen Aufstellung eine Einordnung der Medien nach einem System der Wissenschaften. Die Reihenfolge der Publikationen im Bibliotheksregal entspricht der Abfolge der Sachgebiete in der verwendeten Systematik bzw. Klassifikation.<sup>82</sup>

*„Die Vorteile der systematischen Aufstellung liegen vor allem darin, dass die Literatur zu einem Thema an einer Stelle aufgestellt ist und man daher am Regal rasch einen Überblick über die [gedruckten] Bücher zu einem bestimmten Sachgebiet gewinnen kann.“<sup>83</sup>*

Voraussetzung für die Anwendbarkeit der zu erstellenden App ist also die systematische Aufstellung.

### 3.3 Formate von E-Ressourcen in Bibliotheken

Um E-Books und andere elektronische Ressourcen auf mobilen Endgeräten aufrufen zu können, ist eine wesentliche Voraussetzung, dass die Formate, in denen die E-Ressourcen von Bibliotheken erworben werden, auf diesen dargestellt werden können. Deshalb sollen nun kurz die überwiegend von Bibliotheken genutzten beiden E-Book-Formate PDF und EPUB dargestellt werden.

In seinem 2012 publizierten Artikel über durch E-Books ausgelöste Veränderungsprozesse in wissenschaftlichen Bibliotheken<sup>84</sup> stellte Mumenthaler heraus, dass Verlage sowohl E-Journals als auch E-Books auf einer Webseite anbieten. Dort werden dann einzelne Artikel (einer Zeitschrift) oder einzelne Kapitel (eines E-Books) als PDF-Dokument (Portable Document Format) zum Download bereitgestellt. Einige Anbieter wie Springer bieten auch Volltexte als PDF-Dokumente an.

---

<sup>81</sup> Vgl. Gantert; Hacker (2008), S. 227.

<sup>82</sup> Ebd., S. 230.

<sup>83</sup> Ebd., S. 231.

<sup>84</sup> Vgl. Mumenthaler (2012): E-Books als Katalysator. In: *Bibliothek - Forschung und Praxis*, Jg. 64, Nr. 9, S. 604–608.

Das Portable Document Format (PDF) ermöglicht es, elektronische Dokumente „unabhängig von der verwendeten Soft- oder Hardware originalgetreu wiederzugeben.“<sup>85</sup> Das Öffnen eines PDFs auf mobilen Endgeräten ist ohne die Installation zusätzlicher Software möglich, da die bereits vorinstallierten Programme entsprechende Funktionalitäten zur Verfügung stellen. Das Format erfreut sich zwar einer großen Reichweite, wird jedoch für seine „starre, meist an eine Printversion angepasste Formatierung“<sup>86</sup> kritisiert.

*„In den letzten Jahren haben einige Wissenschaftsverlage damit begonnen, E-Books auch im Format EPUB anzubieten (z.B. Springer oder De Gruyter).“<sup>87</sup>*

„Bei EPUB handelt es sich um einen offenen Standard für die Publikation elektronischer Bücher“, welcher mit Ausnahme von Amazons Kindle von allen E-Book-Readern unterstützt wird. Vorteil gegenüber dem PDF-Format sind die Integration von Multimediaelementen (Bilder, Audiodateien und Videos) und interaktiven Elementen und die dynamische Anpassung der Bildschirmgröße des mobilen Endgeräts.<sup>88</sup>

### 3.4 Native oder Web-App?

Im Bereich der technischen Vorüberlegungen ist vor allem die Frage nach dem Typ der zu entwickelnden Anwendung zu beantworten. Dazu soll nun der Begriff der App und die verschiedenen Typen kurz betrachtet werden.

*„Eine mobile Applikation, kurz App, ist Anwendungssoftware für ein mobiles Gerät, z.B. ein Smartphone oder ein Tablet. Neben nativen Applikationen gibt es webbasierte mobile Dienste, sogenannte Web Apps, die zusätzlich ausgestattet mit dem richtigen CSS-Design und betriebssystemgeeignetem Framework weitgehend agieren wie native Apps (=Hybrid App).“<sup>89</sup>*

Bei einer nativen App handelt es sich um eine Anwendung, die einer Software ähnlich auf dem jeweiligen Gerät installiert wird. Durch die direkte Nutzung des Gerätespeichers kann diese einfach und schnell mit anderen Anwendungen auf dem Gerät zusammenarbeiten, wodurch eine zeitnahe und unmittelbare Reaktionszeit ermöglicht wird. Durch vertrautes Aussehen, Nutzungsgefühl und Funktionsweise wird Zuverlässigkeit vermittelt. Was im ersten Eindruck als großer Vorteil erscheint, bringt jedoch einige Nachteile mit sich: Native Apps müssen für jede Plattform separat entwickelt werden. Darüber hinaus müssen sie im App Store der jeweiligen Plattform (ggf.. kostenpflichtig) eingestellt werden. Native Apps können nicht direkt und unmittelbar zur Verfügung gestellt werden, da sie vor der eigentlichen Veröffentlichung im Store eine detaillierte Prüf- und Testphase durchlaufen müssen.<sup>90</sup>

*„Web Apps – oder besser webbasierte Anwendungen – [hingegen] sind grundsätzlich nichts anderes, als eine für mobile Endgeräte optimierte Webseite (meist unter Verwendung von Responsive Design-Ansätzen), die das Endgerät erkennt und den Inhalt dafür optimiert darstellt. [...] Der Unterschied zwischen [einer] Web App und einer einfachen mobilen Website liegt im Einsatz von CSS3, HTML5 und JavaScript bei der*

---

<sup>85</sup> Vgl. IONOS (2019): Alles rund um E-Books - Teil 3: E-Book-Formate, <https://www.ionos.de/digitalguide/online-marketing/verkaufen-im-internet/e-book-formate/>, 27.06.2019.

<sup>86</sup> Vgl. IONOS (2019): Alles rund um E-Books - Teil 3: E-Book-Formate, <https://www.ionos.de/digitalguide/online-marketing/verkaufen-im-internet/e-book-formate/>, 27.06.2019.

<sup>87</sup> Vgl. Mumenthaler (2018).

<sup>88</sup> Vgl. IONOS (2019).

<sup>89</sup> Vgl. Goltz (2015), S. 7.

<sup>90</sup> Ebd.

*Programmierung. Damit werden das Erstellen dynamischer Oberflächen und Effekte – ähnlich wie bei nativen Anwendungen – ermöglicht.“<sup>91</sup>*

Web-Apps bieten zahlreiche Vorteile. Für die Nutzung einer solchen Anwendung ist jedes Endgerät geeignet, das über einen Browser verfügt. Eine separate Installation ist nicht notwendig. Da die App lediglich einmal entwickelt werden muss, sind die Entwicklungskosten im Vergleich zur nativen App darüber hinaus wesentlich geringer. Auch der Update-Vorgang gestaltet sich wesentlich einfacher. Zu beachten ist jedoch, dass bei Web Apps der Zugriff nur auf bestimmte Kommunikationsmittel wie Kamera oder Kalender, nicht aber auf alle Ressourcen des Geräts, wie bspw. den gesamten Datenspeicher des Geräts, möglich sind.<sup>92</sup>

*„Hardware Features, die in Web Apps nicht umsetzbar sind, können mit Hilfe spezieller Frameworks, die auf die Betriebssysteme der mobilen Endgeräte spezialisiert sind, und JavaScript zu sogenannten Hybrid Apps optimiert werden. Diese unterscheiden sich dann kaum noch von einer nativen Lösung.“<sup>93</sup>*

Tabelle 2 listet die unterschiedlichen Eigenschaften von Web-Apps und nativen Apps, auch im Vergleich zu hybriden Apps, nochmal auf.

	Native App	Web App	Hybrid App
<b>Unterstützung</b>	Jeweils ein Betriebssystem	Alle Betriebssysteme	Alle Betriebssysteme via Framework und JavaScript
<b>Entwicklungsaufwand</b>	Aufwand muss pro Betriebssystem mit unterschiedlichen Frameworks, Programmiersprachen und Tools betrieben werden. Updates müssen dem gleichen Entwicklungsprozess folgen.	Apps werden in HTML5 programmiert. Ein passendes CSS passt die Applikation an.	Apps werden in HTML5 programmiert. Ein passendes CSS passt die App an. Anschließend wird die App via betriebssystemoptimierendem Framework an das jeweilige Gerät angepasst.
<b>Installation</b>	Native Apps werden über den jeweiligen App Store des Betriebssystemanbieters heruntergeladen. Ein Update der App erfordert, dass der Benutzer dieses über den App Store lädt.	Web Apps können überall veröffentlicht und jederzeit aktualisiert werden. Beim Besuch des Dienstes wird die aktuelle Version automatisch geladen. Als Bookmark erscheint ein Symbol auf dem Startbildschirm.	Hybride Apps werden über den jeweiligen App Store des Betriebssystemanbieters heruntergeladen. Ein Update der App erfordert, dass der Benutzer dieses über den App Store lädt.
<b>Integration (Push-News, Zugriff auf andere Applikationen)</b>	Vollständige Integration ins System, Nutzung aller Hardwarekomponenten möglich	Teilweise Integration ins System möglich, wenige Hardwarekomponenten können genutzt werden (z.B. die Verarbeitung bereitgestellter Geodaten)	Fast vollständige Integration ins System, Nutzung aller Hardwarekomponenten möglich (Speicherbeschränkungen)
<b>Performance</b>	sehr hoch	normal	hoch
<b>User-Experience</b>	Passt sich nahtlos an die Bedienung an. Zugriff auf Gerätespeicher ermöglicht viele Funktionen und die Interaktion mit anderen installierten Applikationen.	Kann, abhängig vom CSS und der mobilen Netzanbindung, sehr gut sein. Das „Look and Feel“ ist jedoch meist anders als bei nativen Applikationen auf dem Gerät.	Passt sich beinahe nahtlos an die Bedienung an. Zugriff auf limitierten Gerätespeicher ermöglicht viele Funktionen und die Interaktion mit anderen installierten Applikationen.

Tabelle 2: Vergleich verschiedener App-Typen

<sup>91</sup> Ebd., S. 8.

<sup>92</sup> Ebd.

<sup>93</sup> Ebd.



Quelle: Goltz (2015), S. 9 [Vergleich der Besonderheiten verschiedener App-Typen]

Da der Etat für innovative Entwicklungen in Bibliotheken meist eher gering ausfällt, zeichnet sich eine Tendenz zur Umsetzung von Web Apps ab. Auch die schnelle Umsetzbarkeit und die Möglichkeit der unmittelbaren Publikation der App spricht für diese. Wie schon Frau Goltz schreibt: „Das Preis-/Leistungsverhältnis stimmt.“<sup>94</sup>

### 3.5 Mögliche Frameworks zur Entwicklung von Web Apps

Für die technische Umsetzung sollen nun verschiedene Web-Frameworks betrachtet werden, die für die Umsetzung des Frontends genutzt werden können.

Die Entwicklung einer Web App kann auf unterschiedlichste Art und Weise angegangen werden. So ist es unter anderem möglich, diese von Grund auf mit Hilfe von HTML, CSS und JavaScript selbst zu implementieren. Für die Lösung verschiedener Grundprobleme bei der Entwicklung von Web-Apps stehen mittlerweile jedoch zahlreiche Frameworks zur Verfügung.

*„Ein Framework ist eine Softwareplattform, die die Struktur und Architektur des künftigen Softwareprodukts bestimmt. Jedes Framework enthält ein vorgefertigtes “Gerüst” – die Vorlagen, Standardmodule und APIs, die dem Entwickler zur Verfügung stehen. Während das Framework grundlegende Softwaremodule und Technologien bietet, programmiert der Entwickler alle weiteren, projektspezifischen Komponenten und Funktionen.“<sup>95</sup>*

Schaut man sich die Popularität verschiedener Web-Frameworks an, stellt man fest, dass Vue.js, auch wenn es sich um ein vergleichsweise junges Framework handelt, in den letzten Jahren stark an Popularität gewonnen hat (siehe roter Graph in Abbildung 7):

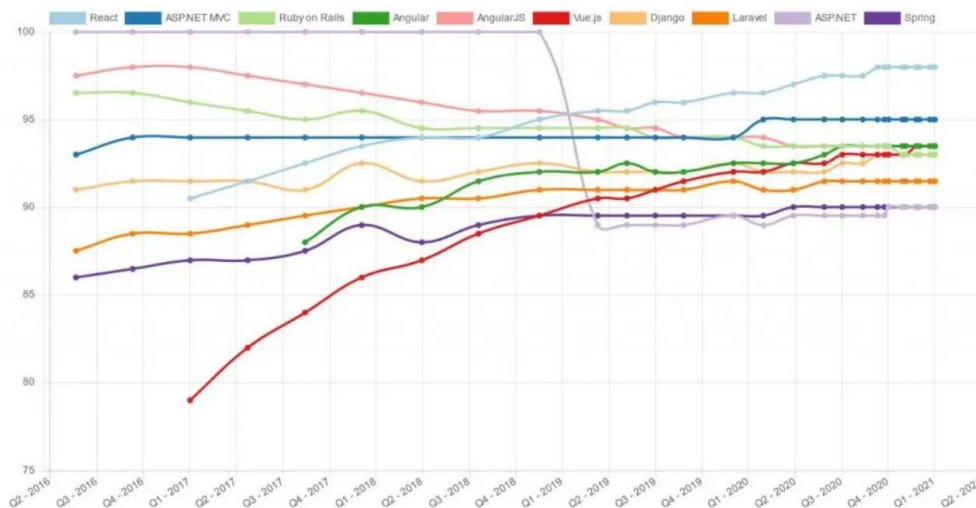


Abbildung 4: Beliebtheit von Web-Frameworks  
Quelle: hotframeworks (n.N.)

<sup>94</sup> Vgl. Goltz (2015): Mobile Applikationen für Bibliotheken im deutschsprachigen Raum, S. 9.

<sup>95</sup> Vgl. Landwehr (2020): Was ist ein Framework? - Definition und Erklärung, <https://it-talents.de/it-wissen/framework/>, 27.06.2021.

Im Rahmen seiner Bachelorarbeit hat Christopher Grätz verschiedene Frontend-Frameworks verglichen, wobei sich ergeben hat, dass Vue.js trotz seines vergleichsweise jungen Alters vor allem in den Bereichen Dokumentation, Einarbeitungszeit und Integrationsfähigkeit Pluspunkte sammeln konnte:

Parameter	Angular	AngularJS	Bootstrap	Meteor	React	VueJS
<b>Erscheinungsjahr</b>	2016	2009	2011	2012	2013	2014
<b>Entwickler</b>	Google	Google	Twitter	Meteor	Facebook	Community
<b>Lizenz</b>	MIT	MIT	MIT	MIT	MIT	MIT
<b>Architektur</b>	MVC	MVVM	entfällt	MVVM	MVC, MVVM	MVVM
<b>Sprache</b>	HTML, CSS, TypeScript	HTML, CSS, JavaScript	HTML, CSS, JavaScript	HTML, CSS, JavaScript	HTML, CSS, JavaScript	HTML, CSS, JavaScript
<b>Dokumentation</b>	++	+	+	+	+	++
<b>Einarbeitungszeit</b>	-	-	++	+	-	++
<b>Desktop</b>	++	++	++	++	++	++
<b>Mobile</b>	++	++	+	++	++	++
<b>Community</b>	++	+	+	+	++	+
<b>Development Speed</b>	+	+	++	++	+	++
<b>Integrationsfähigkeit</b>	o	o	-	o	o	++
<b>Beliebtheit für Web-Apps</b>	o	+	-	-	++	o
<b>Summe</b>	<b>8</b>	<b>7</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>13</b>

*Tabelle 3: Vergleich von Web-Frameworks*

*Quelle: Grätz 2020a, S. 43 [Tabelle 4.3: Vergleich von Web-Frameworks] <sup>96</sup>*

<sup>96</sup> Erläuterung zur Bewertungsskala: „++“ : sehr gut, „o“: befriedigend, „-“: sehr schlecht

## 3.6 Zusammenfassung

Wie sich im Kapitel gezeigt hat, gibt es bereits verschiedene Ansätze, den Standort des Nutzers in einer Bibliothek zu ermitteln. Hier ist vor allem das an der TH Wildau genutzte System der iBeacons zu nennen. Nicht zuletzt aufgrund des enormen Einrichtungsaufwandes ist es für das Ziel, den Bibliotheksnutzern möglichst genaue, zum eigenen Standort passende, thematische Vorschläge für elektronische Ressourcen, nur bedingt geeignet. Auch bei nahe beieinander oder gegenüber stehenden Regalen kann es zu Ungenauigkeiten kommen. Details bis hin zu einzelnen Regalebenen lassen sich darüber hinaus nicht mehr abbilden. Als unkompliziert einzurichtende und anzuwendende Möglichkeit stellten sich hingegen optische Tags wie bspw. QR-Codes heraus.

Gezeigt hat sich weiterhin, dass bestimmte Anforderungen an die Aufstellung des Bestandes im Lesesaal der Bibliothek bestehen. Ideal ist hier eine systematische Freihand-Aufstellung. So ließen sich passende E-Ressourcen mit Hilfe der Systematikstelle(n) des Regals ermitteln.

Um den Nutzern geeignete Ressourcen bereitzustellen, bieten sich Recherchen mit Hilfe von Schlagwörtern oder Notationen an. Probleme haben sich hier jedoch im Bereich der Erschließung von E-Ressourcen gezeigt. Im Vergleich zu Printmedien werden diese häufig nicht mehr einzeln per Autopsie erschlossen. Da gerade E-Books häufig in Paketen gekauft werden, wird nicht mehr jeder einzelne Datensatz inspiziert, sondern die vom Verlag gelieferten Fremddaten übernommen, was zur Folge hat, dass auch im Bereich der Sacherschließung zum Teil nur unzureichende Informationen vorliegen.

Bei der Betrachtung verschiedener mobiler Apps hat sich herausgestellt, dass sich gerade für Bibliotheken aufgrund der einfachen Möglichkeit der Veröffentlichung und der wesentlich günstigeren Entwicklungskosten (im Vergleich zu nativen Apps) die Erstellung von Web-Apps empfiehlt.

Der Vergleich verschiedener Frameworks für die Erstellung von Web-Frameworks zeigte, dass das vergleichsweise junge Tool Vue.js sich als vielversprechend vor allem im Bereich Dokumentation und Einarbeitungszeit herausstellte.

Nachdem nun allgemeine Aspekte betrachtet wurden, soll im nächsten Kapitel der Ist-Zustand an der Bibliothek der TH Wildau beschrieben werden.

## 4 Ist-Stand

### 4.1 Die Hochschulbibliothek der TH Wildau

Die Hochschulbibliothek der TH Wildau, welche im Jahr 2012 vom Deutschen Bibliotheksverband zur Bibliothek des Jahres gekürt wurde,<sup>97</sup> stellt „elektronische und gedruckte Informationsdienstleistungen für Studierende und Angehörige der TH Wildau zur Verfügung.“<sup>98</sup>

*„Die Hochschulbibliothek als zentraler Ort für Informationsversorgung versteht sich zudem nicht nur als Lernort, der den Hochschulangehörigen und interessierten Besuchern eine entsprechende Infrastruktur bereitstellt, sie stellt sich zudem den gesellschaftlichen und technischen Veränderungen und begegnet diesen mit innovativen Entwicklungen an Dienstleistungen und Lösungen.“<sup>99</sup>*

Sie zeichnet sich also vor allem durch die Umsetzung innovativer Projekte aus. Spezialisiert hat sich die Bibliothek vor allem auf den Einsatz der RFID-Technologie. Neben den Bereichen Mediensicherung und Selbstverbuchung wird diese seit 2012 auch für die Inventur genutzt. Durch die Ausstattung von Rückstelltischen mit RFID wird es darüber hinaus ermöglicht, Kenntnisse über die Nutzungsfrequenz von Lesesaalbeständen zu erhalten.<sup>100</sup> Bekanntheit erlangte die Bibliothek vor allem über den Einsatz des humanoiden Roboters Wilma, welcher im Februar 2019 vom Stifterverband mit der Hochschulperle Future Skills ausgezeichnet wurde.<sup>101</sup> "Durch Automatisierung und Digitalisierung erreichen die Serviceangebote der TH Wildau eine neue Qualitätsstufe."<sup>102</sup> Erreicht wird dies vor allem durch die Kooperation der Bibliothek mit den Studierenden und Professoren der Hochschule, bspw. mit dem Team des RoboticLab des Studiengangs Telematik.

Organisatorisch ist die Bibliothek der Technischen Hochschule Wildau zugeordnet, welche im Jahr 2020 ca. 3.800 Studierende verzeichnete. Davon waren ca. 3.300 eingeschriebene Nutzer der Hochschulbibliothek. Hinzu kommen ca. 350 Nutzer aus dem Bereich wissenschaftliches Personal und 670 externe Nutzer im Bezugsjahr 2020. Geöffnet hat die Bibliothek an 55 Öffnungsstunden. Der Bestand umfasst ca. 81.000 gedruckte Bände und über 67.000 Einheiten digitaler Bestand. Hinzu kommen ca. 100 gedruckte Zeitschriften- und ca. 12.700 lizenzierte und digitale laufende Zeitschriftentitel. Auf den Bestand entfielen im Jahr 2020 ca. 19.000 Entleihungen (2019: 30.000).<sup>103</sup>

---

<sup>97</sup> Vgl. Deutscher Bibliotheksverband e.V. (n. N.).

<sup>98</sup> Vgl. TH Wildau (2021): Hochschulbibliothek, <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/>, 13.02.2021.

<sup>99</sup> Vgl. Azmitia; Mohnke; Wiechers (2016), S. 396.

<sup>100</sup> Vgl. TH Wildau | Bibliothek (2021): Projekte, <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/ueber-die-bibliothek/projekte/>, 13.02.2021.

<sup>101</sup> Vgl. Stifterverband (2019): Bibliotheksroboter Wilma: Hochschulperle des Monats Februar geht nach Wildau in Brandenburg, [https://www.stifterverband.org/pressemitteilungen/2019\\_02\\_27\\_hochschulperle\\_bibliothekroboter\\_wilma](https://www.stifterverband.org/pressemitteilungen/2019_02_27_hochschulperle_bibliothekroboter_wilma), 13.02.2021.

<sup>102</sup> Ebd.

<sup>103</sup> Vgl. hbz (n. N.): Deutsche Bibliotheksstatistik: variable Auswertung, <https://www.bibliotheksstatistik.de/vaAttribute>, 16.07.2021.

Die Aufstellung der Bestände erfolgt als Freihandbestand nach Fachgebieten (inklusive Lehrbuchsammlungen). Abschlussarbeiten und Zeitschriften sind dabei separat aufgestellt.

## 4.2 Recherche- und Informationstools

Im folgenden Abschnitt sollen die bestehenden Recherche- und Informationstools, die derzeit an der TH Wildau genutzt werden, vorgestellt werden.

### 4.2.1 Wilbert

Um einen schnellen Überblick über die Bestände der Bibliothek zu erhalten, steht den Nutzern an erster Stelle das Discovery-System Wilbert (Wildauer Bücher+E-Medien Recherche-Tool) zur Verfügung.<sup>104</sup> Hier sind sowohl Printmedien als auch e-Ressourcen verzeichnet. Gesucht werden kann nach Monographien, Zeitschriften und Artikeln. Derzeit kann in über sieben Millionen Datensätzen recherchiert werden.<sup>105</sup> Wilbert steht, wie alle digitalen Angebote der TH Wildau, in deutscher und englischer Sprache zur Verfügung. Neben der einfachen Suche gibt es eine erweiterte Suche, womit gezielt in den Feldern Titel, Autor/Körperschaft, Verlag/Verlagsort, ISBN/ISSN/ISMN, Schlagwort, Erscheinungsjahr und RVK recherchiert werden kann. Nach Absetzen der Suchabfrage werden verschiedene Filtermöglichkeiten, wie bspw. Erscheinungszeitraum, Sprache oder Schlagwörter, angeboten. Im jeweiligen Titelsatz finden sich bei E-Ressourcen zusätzlich die entsprechenden Zugangsverlinkung zum Volltext. Darüber hinaus lassen sich in Wilbert derzeit entliehene Werke vormerken. Über verschiedene Einstellungsmöglichkeiten lassen sich zahlreiche Standardwerte, die für die Suche genutzt werden sollen, konfigurieren, so bspw. die Anzahl oder die Sortierung (auf- oder absteigend, nach Relevanz, Jahr, Titel oder Autor) der Treffer pro Seite. Darüber hinaus können Einstellungen zum Layout (Anzeige der Filter rechts oder links) oder zum Export von Daten gemacht werden.

Im Bereich der Printressourcen bietet Wilbert die Möglichkeit, sich den Standort des Bandes im Lesesaal der Bibliothek anzeigen zu lassen:

---

<sup>104</sup> Vgl. TH Wildau | Bibliothek (n. N.): WILBERT — Wildauer Bücher+E-Medien Recherche-Tool, <https://wilbert.kobv.de/?plv=2>, 13.02.2021.

<sup>105</sup> Ebd.

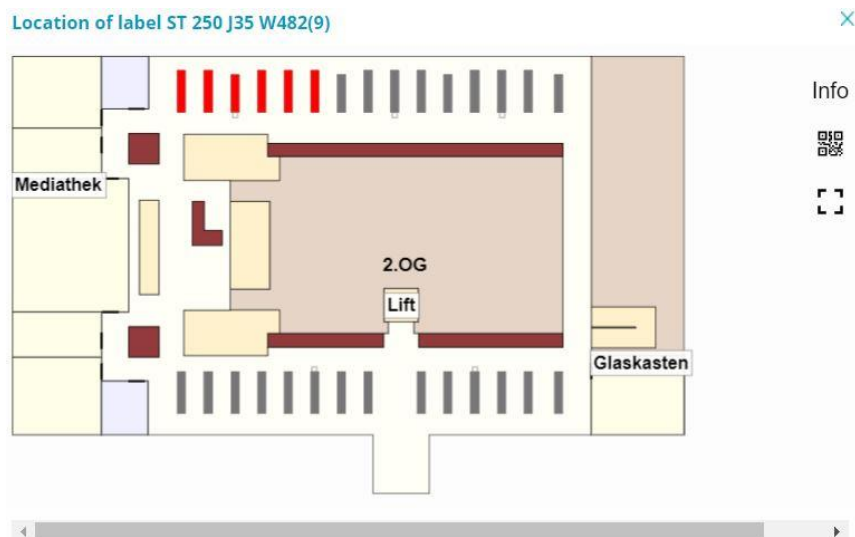


Abbildung 5: Standortanzeige gedruckter Medien in Wilbert

## 4.2.2 Online-Katalog

Auch wenn er nicht mehr unmittelbar auf der Startseite der Homepage der Bibliothek beworben wird, so steht dennoch der klassische Online-Katalog zur Verfügung.<sup>106</sup> Hier können zusätzlich Recherchen auf Exemplarebene durchgeführt werden, so dass nach der Signatur eines Bandes gesucht werden kann. Darüber hinaus ist hier eine Suche in verschiedenen Indizes möglich (z.B. Verfasser, Verlag oder Schlagwort). Der Zugriff auf das Bibliothekskonto erfolgt ebenfalls hier. Im Gegensatz zu Wilbert finden sich hier jedoch keine Artikelnachweise.<sup>107</sup>

## 4.2.3 Campus App Unidos Wildau

Seit April 2015 steht allen Studenten und Mitarbeitern mit der Campus App Unidos Wildau ein mehrsprachiges, mobiles Informationssystem sowohl für Android-, als auch iOS-basierte mobile Endgeräte zur Verfügung.<sup>108</sup> Ziel der Anwendung ist es, „alle relevanten Informationen zum Hochschulleben über eine zentrale Plattform zugänglich zu machen“<sup>109</sup> Der Name der App spiegelt zugleich den Charakter der App wieder: *unidos*, aus dem Spanischen übersetzt, steht für „zusammen/vereint“.

*„Unidos Wildau ist eine App für alle Angehörige[n] der Technischen Hochschule Wildau, deren Weiterentwicklung auch entscheidend von den verschiedenen Ideen aller getragen wird.“<sup>110</sup>*

Darüber hinaus bezieht sich die Kombination von „uni“ und „dos“ auf die zweite Version der App.<sup>111</sup>

<sup>106</sup> Siehe <https://btwoz2.bib-bvb.de/InfoGuideClient.btwsis/search.do?methodToCall=start>

<sup>107</sup> Vgl. TH Wildau | Bibliothek (n. N.): Wilbert/Onlinekatalog, <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/recherche/wilbert/>, 08.08.2021.

<sup>108</sup> Vgl. Azmitia; Mohnke; Wiechers (2016): Campus App Unidos Wildau - ein ständiger Begleiter für den Alltag an der Technischen Hochschule Wildau. In: Barton, T. et al. (Hg.), S. 227–242, S. 227.

<sup>109</sup> Ebd., S. 229.

<sup>110</sup> Ebd., S. 228.

<sup>111</sup> Vgl. Informationen zum Namen der App unidos in der App selbst.

Hervorgegangen ist die Anwendung aus dem Vorgängerprojekt *iLibrary Wildau*, welche zunächst lediglich als „Bibliotheksschnittstelle konzipiert und umgesetzt war.“<sup>112</sup> An der Entwicklung von *unidos* waren insbesondere Studenten und Mitarbeiter des Studiengangs Telematik beteiligt.

Als Anwendung für den gesamten Campus bietet *unidos* verschiedenste Funktionen für unterschiedliche Zielgruppen. Für Hochschulangehörige stehen (nach Anmeldung) folgende Funktionen zur Verfügung:

- Anzeige von Stundenplänen und Terminen über Auswahl der Seminargruppen für Studierende und Lehrende
- Verwaltung ausgeliehener Bücher aus der Bibliothek
- Buchung von Arbeitsräumen zur Gruppenarbeit
- Anzeige des Mensaplans
- direkter Zugriff auf andere webbasierte Anwendungen und Angebote der Hochschule wie die e-learning-Plattform oder den Panorama-Campusrundgängen
- Zugriff auf E-Mail-Dienst der Hochschule

Darüber hinaus stehen zahlreiche Mitteilungsdienste zur Verfügung. Nutzer können dabei die verschiedenen Kanäle der Hochschuleinrichtungen abonnieren. Benachrichtigungen erfolgen dabei über Push-Nachrichten auf dem Sperrbildschirm.<sup>113</sup>

Für Besucher oder potenzielle Studenten und Erstsemester bietet die App

*„[m]ultimediale Inhalte und virtuelle Rundgänge, die sowohl mit Informationen über Studienangebote und Einrichtungen der Hochschule einen Überblick über die Hochschule geben als auch das Kennenlernen des Campus mittels Panoramaeinblicken ermöglichen.“<sup>114</sup>*

Um die App trotz der zahlreichen Funktionalitäten zu untergliedern, wurden diese in vier Bereiche aufgeteilt:

- Studium und Lehre
- Campus
- Bibliothek
- Mitteilungen

Der Datenaustausch der Anwendung ist über Webservices organisiert. Dies ermöglicht es, Änderungen und Erweiterungen an Datensätzen möglichst einfach durchführen zu können.<sup>115</sup> Zur Kommunikation mit diesen werden Daten im JSON-Format ausgetauscht, „weil dessen Implementierung - im Vergleich zu anderen Netzwerkprotokollen wie SOAP (in Kombination mit XML) – in verschiedenen Programmiersprachen mittlerweile ohne externe Bibliotheken leicht umzusetzen ist. Hinzu kommt, dass das JSON-Format leicht menschenlesbar ist, was das Verstehen der Dokumentation für Entwickler erleichtert.“<sup>116</sup>

---

<sup>112</sup> Ebd.

<sup>113</sup> Ebd., S. 230.

<sup>114</sup> Ebd.

<sup>115</sup> Ebd., S. 235.

<sup>116</sup> Ebd., S. 236.

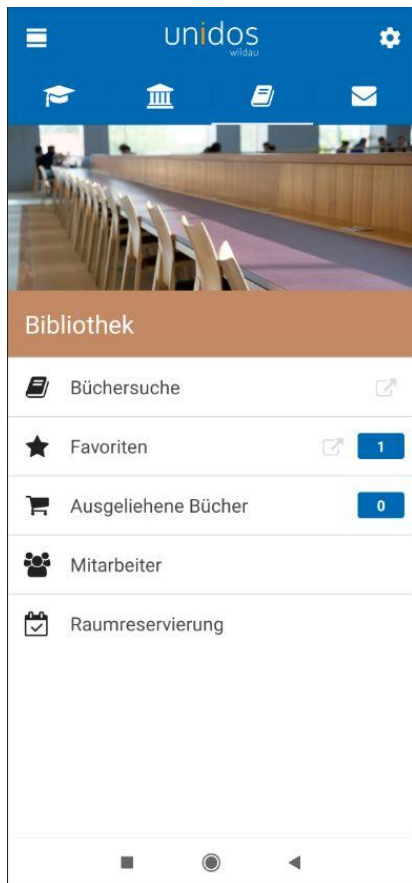


Abbildung 6: Ansicht der unidos-App für den Bereich "Bibliothek"

Die Webservices stehen REST-konform zur Verfügung. Das heißt, dass jede Ressource über eine URI (Uniform Resource Identifier) adressierbar ist und darüber angesprochen werden kann.<sup>117</sup>

Um die Sicherheit der Kommunikationsschnittstellen zu gewährleisten, werden alle Informationen über HTTPS übertragen. Das zugehörige Zertifikat ist mit der kryptografischen Hashfunktion SHA-256 signiert.<sup>118</sup> Darüber hinaus ist für die Abfrage von fast allen Diensten ein Token erforderlich. Dabei melden sich die Studenten mit ihrem Hochschul-Account an, welcher dann vom Hochschulrechenzentrum überprüft wird. Im Falle einer erfolgreichen Anmeldung wird ein entsprechender Token ausgestellt.<sup>119</sup>

*„Hochschuldienste, die als alleinstehende Anwendungen bereits für die mobile Nutzung optimiert sind und daher ein fertiges User-Interface mitbringen (Moodle, E-Mail-Client), sind als externe Links markiert und werden im Frame eines Views aufgerufen. Sie laufen als eigenständige webbasierte Anwendung, sind aber durch die Rahmensteuerung in Unidos integriert.“<sup>120</sup>*

Für die Umsetzung der Push-Dienste werden Services von Google und Apple genutzt.<sup>121</sup>

Gerade durch seine einfache Erweiterbarkeit stehen die Chancen sehr gut, dass sich Unidos auch in Zukunft immer „aktuell an den Bedürfnissen seiner Nutzer orientieren kann“<sup>122</sup> und sich somit weiterhin als Hilfsmittel für Hochschulangehörige und Gäste der Technische Hochschule beweisen kann, nicht zuletzt durch sein „modernes, übersichtliches und visuell sehr ansprechendes Design [welches] den Spaß an der Nutzung der Anwendung [fördert].“<sup>123</sup>

nes, übersichtliches und visuell sehr ansprechendes Design [welches] den Spaß an der Nutzung der Anwendung [fördert].“<sup>123</sup>

#### 4.2.4 Weitere digitale Services

Über die Grenzen von Wildau hinweg ist der humanoide Bibliotheksroboter Wilma bekannt, ein NAO-Roboter der Firma Softbank Robotics. Wilma agiert dabei als Serviceroboter und unterstützt das Bibliothekspersonal durch Bibliothekseinführungen und -rundgänge. Auch die Umsetzung des 24/7-Konzepts, welches Studenten und Mitarbeitern den Zugang zur Bibliothek rund um die Uhr ermöglicht, wäre ohne

<sup>117</sup> Ebd.

<sup>118</sup> Ebd.

<sup>119</sup> Ebd.

<sup>120</sup> Ebd., S. 234.

<sup>121</sup> Ebd., S. 236.

<sup>122</sup> Ebd., S. 241.

<sup>123</sup> Ebd.



Wilma nicht denkbar gewesen. Neben der Beantwortung von verschiedenen Service-Fragen bspw. zu Druck- oder Kopierfragen kann Wilma Witze erzählen.<sup>124</sup>

Im Rahmen einer Bachelorarbeit im Studiengang Telematik wurde von Christopher Grätz darüber hinaus ein Prototyp für eine 3D-Darstellung der Bibliothek der TH Wildau entwickelt. Als Hauptzielgruppe sollen mit der App vor allem Gäste der Hochschulbibliothek angesprochen werden, aber auch „Interessierte, die sich nicht auf dem Campus der TH Wildau befinden und die Bibliothek erkunden und dessen Angebot an [vor allem digitalen] Dienstleistungen [...] kennen lernen möchten“<sup>125</sup>. Zielstellung ist primär die Präsentation der Bibliothek und ihrer innovativen Dienstleistungen nach außen hin. Darüber hinaus kann die App Hochschulangehörigen helfen, neue Dienstleistungen der Bibliothek zu entdecken, wenn sie nicht vor Ort sind. Dazu beinhaltet die Anwendung verschiedene Points of Interest (POIs):<sup>126</sup>

Hier integriert ist auch der bereits bestehende Prototyp der App „E-Books To Go“, welche im Rahmen eines Praktikums erstellt wurde und das Stöbern am Regal um die Anzeige von E-Books ergänzt. Ausgewählte Funktionalitäten dieser Anwendung (siehe Kapitel 4.3) könnten dann über das 3D-System genutzt werden. Die technische Implementation der E-Book-App in das 3D-System ist über einen iFrame umgesetzt. Damit dies umgesetzt werden kann, muss die einzubindende Ressource über eine URL (Uniform Resource Locator) adressierbar sein.<sup>127</sup>

Darüber hinaus stehen den Studierenden der technischen Hochschule zahlreiche weitere (mobile) Dienstleistungen zur Verfügung. So wurde im Rahmen der Corona-Pandemie eine Web-App zur Kontaktnachverfolgung entwickelt: die „THWI Kontaktnachverfolgung“, welche seit Oktober 2020 zur Verfügung steht.<sup>128</sup> Bei der Anwendung handelt es sich um „eine Eigenentwicklung des iCampus-Teams [...], die] in enger Abstimmung und kürzester Zeit mit dem Krisenstab der TH Wildau und dem Vizepräsidenten für Digitalisierung und Qualitätsmanagement entstanden [ist].“<sup>129</sup>

### 4.3 Der Prototyp der App E-Books-To-Go

Im Rahmen eines Praktikums von zwei Telematik-Studenten (Christopher Grätz und Omar Attia) im Wintersemester 2019/2020 entstand ein erster Prototyp einer Anwendung, die als Ziel hat, den Nutzern beim Stöbern am Regal die (durch die Bibliothek lizenzierten) E-Books anzeigen zu können und somit zur Überwindung des Medienbruchs beizutragen. Im Rahmen einer Bachelorarbeit von Omar Attia wurde der Prototyp um weitere Funktionalitäten erweitert.

---

<sup>124</sup> Vgl. TH Wildau | Hochschulbibliothek (n. N.): Bewerbung zur Bibliothek des Jahres 2019 : Praktische Anwendungen, <https://icampus.th-wildau.de/bewerbung-bdj-2019/praktische-anwendungen.html>, 08.08.2021.

<sup>125</sup> Vgl. Grätz (2020a): Bibliothekarische Dienstleistungen für die Überwindung des Medienbruchs von physischen und elektronischen Medien, S. 4.

<sup>126</sup> Ebd., S. 52.

<sup>127</sup> Ebd., S. 56.

<sup>128</sup> Diese ist erreichbar unter <https://icampus.th-wildau.de/kontaktnachverfolgung/>

<sup>129</sup> Vgl. TH Wildau (n. N.): Web-App "THWi Kontaktnachverfolgung", <https://www.th-wildau.de/studieren-weiterbilden/neuigkeiten-und-veranstaltungen/kontaktnachverfolgung/>, 01.07.2021.

Im Folgenden sollen zunächst die grundlegenden Funktionalitäten der App beschrieben werden, bevor anschließend die technische Umsetzung dieser skizziert wird. Gegebenenfalls können hieraus weitere Anforderungen abgeleitet werden.

### 4.3.1 Umgesetzte Funktionalitäten

Zielstellung der App „E-Books To Go“ ist die zum Regalstandort der Nutzer passende Anzeige von E-Books, die durch die Hochschulbibliothek der TH Wildau lizenziert wurden. Somit soll den Nutzern auch beim Stöbern am Regal ein möglichst umfassender Bestandsüberblick geboten werden.

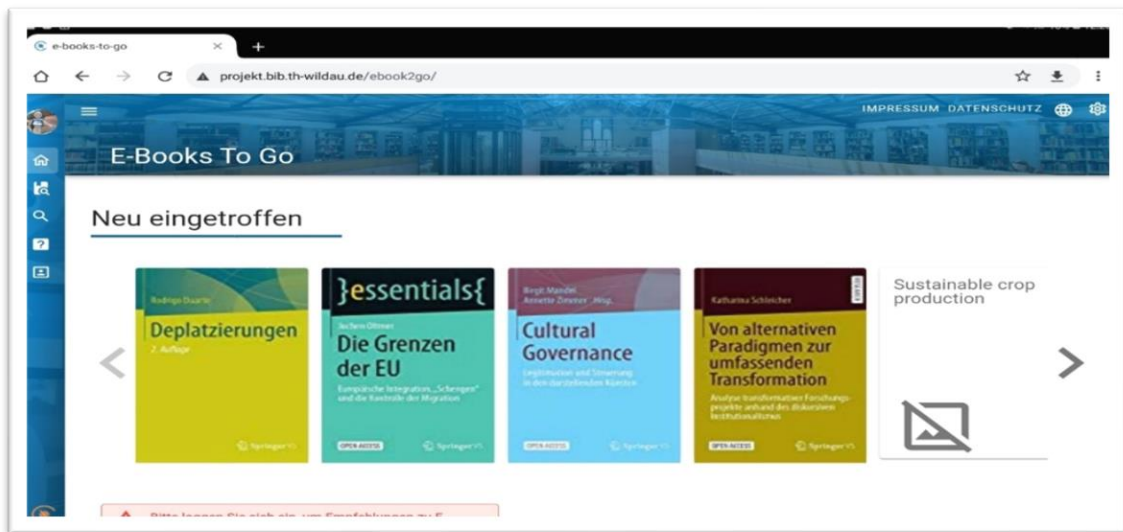


Abbildung 7: Startseite der App "E-Books To Go" (Desktop)

Das User-Interface (UI) der App steht in deutscher und englischer Sprache zur Verfügung und ist auf verschiedenen mobilen Endgeräten (Tablet, Smartphone, ...) im Browser aufrufbar (responsiv gestaltet). Um eine möglichst hohe Usability zu erreichen, wurde das Design mit Komponenten des Material Design umgesetzt.

*„Material is a design system created by Google to help teams build high-quality digital experiences for Android, iOS, Flutter, and the web. [It] is inspired by the physical world and its textures, including how they reflect light and cast shadows. Material surfaces reimagine the mediums of paper and ink.“<sup>130</sup>*

Beim Aufrufen der App werden in einer Art Dashboard die "neu eingetroffenen" E-Books der Bibliothek angezeigt (siehe Abb. 7). Die Darstellung der E-Books erfolgt durch ihre Cover in einer Art Slideshow, bei denen die Cover nebeneinander abgebildet werden. Diese ist für Desktop, Tablet und Smartphone (siehe Abb. 8) umgesetzt.

<sup>130</sup> Vgl. Google (n. N.): Material Design - Introduction, <https://material.io/design/introduction>, 02.07.2021.



Abbildung 8: Startseite der App "E-Books To Go" (Smartphone)

Nach einem Login mit der Hochschulkenung werden darüber hinaus E-Book-Empfehlungen dargestellt, welche sich aus den Literaturhinweisen des Modulhandbuchs für den eigenen Studiengang ergeben. Auch diese werden in Form einer Slideshow präsentiert.

Header und Navigationsmenü finden sich auf allen Seiten der Anwendung. Im Header der App lässt sich die Sprache der App auswählen. Über ein Navigationsmenü, welches sich am linken Rand befindet, können die weiteren Funktionalitäten der Anwendung erreicht werden. Zudem ist für den Nutzer jederzeit erkennbar, ob er angemeldet ist, indem im Falle einer aktiven Session sein Name und seine Rolle (bspw. „Student“) angezeigt wird.

Im Folgenden sollen die einzelnen Funktionen erläutert werden:

1.) Passende E-Books zum Regalbereich finden – „E-Books finden“

Um sich zum Standort passende E-Books anzeigen zu lassen, besteht die Möglichkeit, den eigenen Standort mittels Einlesen eines indirekten QR-Codes ermitteln zu lassen. Durch Weiterleitung der im QR-Code hinterlegten ID kann über eine Datenbank die zugehörige RVK-Notation des Regals ermittelt und mit Hilfe dieser die zum Thema passenden E-Books in der App angezeigt werden. Dargestellt werden die E-Books auch hier durch ihr Cover. Anders als auf dem Dashboard erfolgt die Anzeige hier in Form einer Liste (siehe Abb. 9). Auf dem Smartphone werden die einzelnen Treffer untereinander angezeigt.

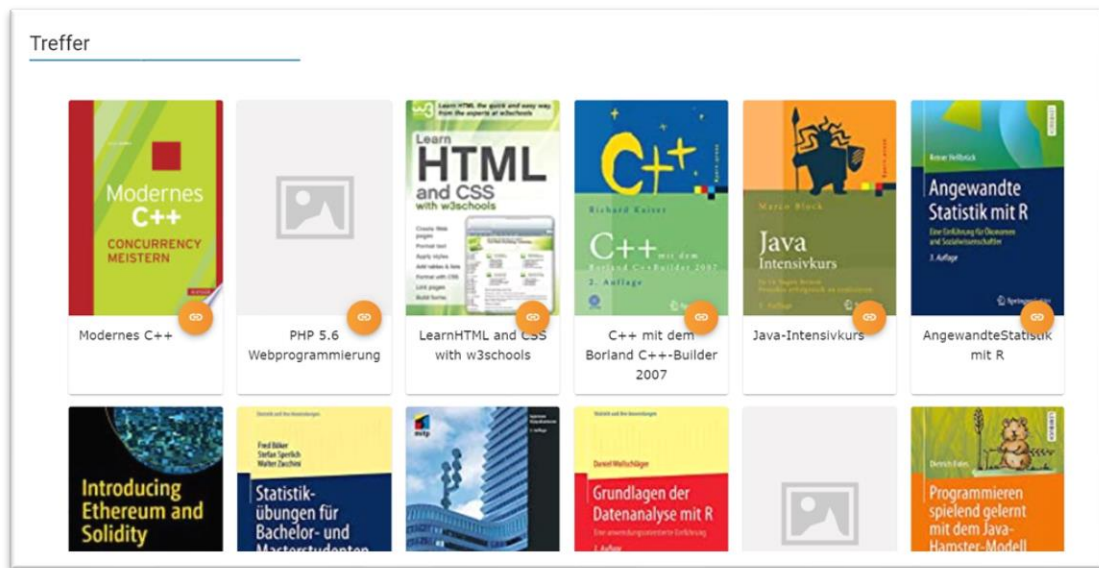


Abbildung 9: Anzeige der Trefferliste nach Scannen eines QR-Codes (Desktop)

Im Gegensatz zur Anzeige in der Slideshow sind die einzelnen Datensätze hier um die Angabe des Titels in Textform unterhalb des Covers ergänzt. Bereits aus der Liste heraus besteht die Möglichkeit, zum Volltext einzelner Titel zu gelangen. Die Verknüpfung führt (wie in Wilbert) auf die Anbieterseite des E-Books. Sollte dort ein Volltext im PDF-Format zum E-Book zur Verfügung stehen, muss dies auf der Anbieterseite geöffnet werden. Es erfolgt also bisher keine „direkte Durchschaltung“ zum E-Book. Über die Trefferliste gelangt man aber auch zur Detailansicht der E-Books (siehe Abb. 9).

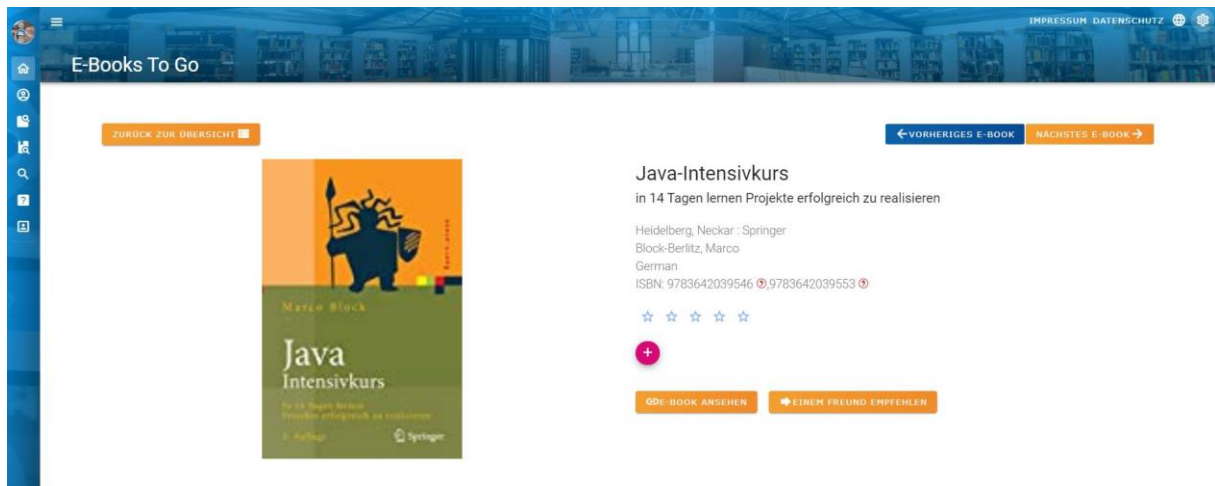


Abbildung 10: Detailansicht E-Book

Diese stellt neben dem Titel weitere bibliographische Informationen wie Verfasser, Erscheinungsort und Verlag, Sprache des E-Books sowie ISBNs zur Verfügung. Auch aus der Detailansicht heraus kann der Volltext (über die Anbieterseite) aufgerufen werden. Nach Anmeldung mit der Hochschulkennung kann zudem eine Empfehlung für ein bestimmtes E-Book als E-Mail verschickt werden. Aus der Detailansicht heraus kann das vorherige bzw. nächste E-Book aus der Liste über eine Art Blätterfunktion aufgerufen werden.

## 2.) aktive Suchmöglichkeit nach E-Books – „E-Books suchen“

Die App bietet die Möglichkeit, selbst eine thematische Suche nach E-Books durchzuführen. Eine Eingrenzung der Suchergebnisse ist möglich nach Publikationsjahr-Intervall, Sprache (deutsch/englisch) oder Anzahl der Treffer (10, 20, 50, 100). Die Sortierung der Suchergebnisse kann nach Relevanz, Autor (auf-/absteigend), Erscheinungszeitraum (auf-/absteigend) oder Titel (auf-/absteigend) erfolgen und kann bereits vor Absenden der Suchabfrage eingestellt werden. Die Darstellung der Treffer erfolgt hier, wie auch bei der Funktion „E-Books finden“ in einer Liste.

## 3.) Anzeige von Empfehlungen zum eigenen Studiengang – „Alle Empfehlungen“

Über den Menüpunkt „Alle Empfehlungen“ besteht nach Anmeldung(!) die Möglichkeit, sich semesterweise E-Book-Empfehlungen zu den Modulen des eigenen Studiengangs anzeigen zu lassen. Ermittelt werden diese anhand der Literaturhinweise aus dem Modulhandbuch. Nach Auswahl eines Semesters werden diese modulweise aufgelistet. Enthalten diese mehr E-Books, als in einer Reihe aufgelistet werden können, erfolgt die Anzeige in Form einer Slideshow (siehe Abb. 11). Um zum E-Book zu gelangen, muss zunächst die Detailansicht aufgerufen werden.

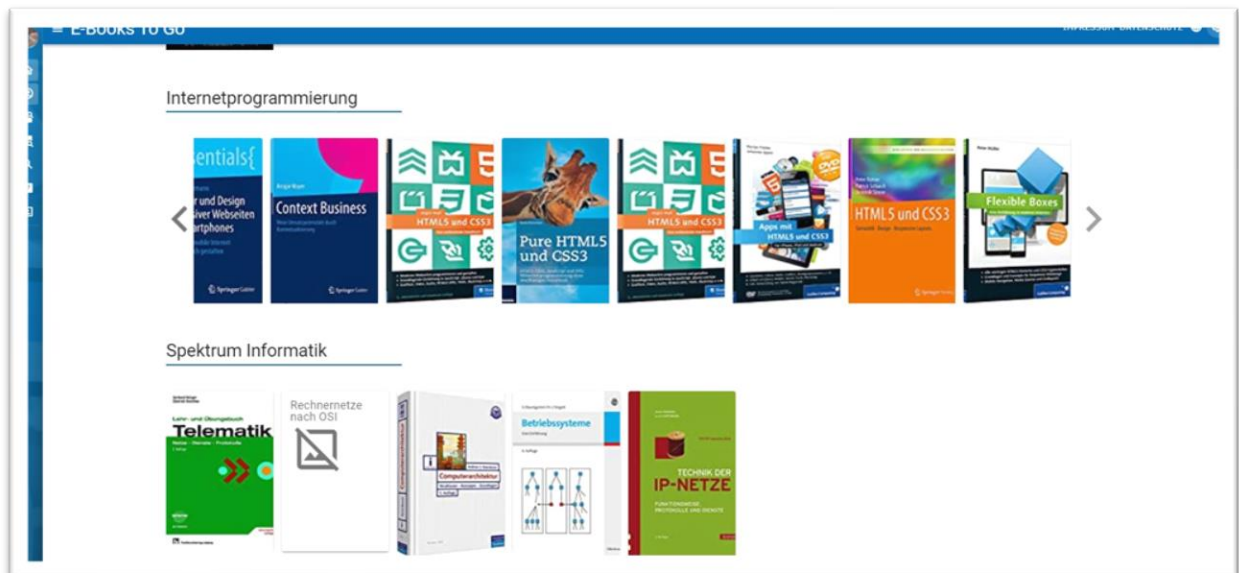


Abbildung 11: E-Book-Empfehlungen zu einzelnen Modulen des eigenen Studiengangs

## 4.) Suche nach Empfehlungen – „Empfehlungen suchen“

Darüber hinaus besteht die Möglichkeit, nach Literaturempfehlungen anderer Studiengänge zu suchen. Dazu müssen zunächst Studiengang, Semester und Modul ausgewählt werden, bevor die Treffer in Form einer Slideshow angezeigt werden.

## 5.) Login

Über den Login-Button, welcher sich an oberster Stelle im Navigationsmenü befindet, hat ein Nutzer jederzeit die Möglichkeit, sich mit seiner Hochschulkennung anzumelden. Nach erfolgreicher Anmeldung stehen die oben beschriebenen zusätzlichen Funktionalitäten zur Verfügung.

Über eine Hilfeseite werden die Funktionen der App erklärt.

### 4.3.2 Technische Umsetzung/Systemarchitektur

Im folgenden Abschnitt soll nun die grundlegende Systemarchitektur des Prototyps beschrieben werden.

*„Die Architektur des Systems ist ein entscheidender Aspekt, damit Performance und Skalierbarkeit gewährleistet sind.“<sup>131</sup>*

Die Anwendung besteht aus zwei Teilsystemen (Frontend<sup>132</sup> und Backend<sup>133</sup>), welche mit Hilfe JavaScript-basierter Frameworks umgesetzt wurden. Laut Sebastian Springer ist JavaScript aufgrund der Vielzahl der verfügbaren Frameworks und Bibliotheken „eine der erfolgreichsten Programmiersprachen der Welt“<sup>132</sup>.

Jedes der Teilsysteme Front- und Backend hat dabei spezifische Aufgaben. Während das Frontend quasi die Benutzerschnittstelle darstellt, erledigt das Backend im Hintergrund auszuführende Aufgaben wie die Generierung der im Frontend darzustellenden E-Books. Wie in Abbildung 12 zu erkennen ist, erfolgt eine ständige Kommunikation zwischen Frontend und Backend.

Die Web-App wird vom Nutzer als Client gestartet. Sowohl während des Startvorgangs der App als auch nach Interaktionen des Nutzers mit der Anwendung erfolgt das Laden relevanter Daten aus dem Backend.<sup>133</sup>

---

<sup>131</sup> Vgl. Grätz (2020b), S. 19.

<sup>132</sup> Vgl. Springer (2018): Node.js, S. 30.

<sup>133</sup> Vgl. Grätz (2020b), S. 20.

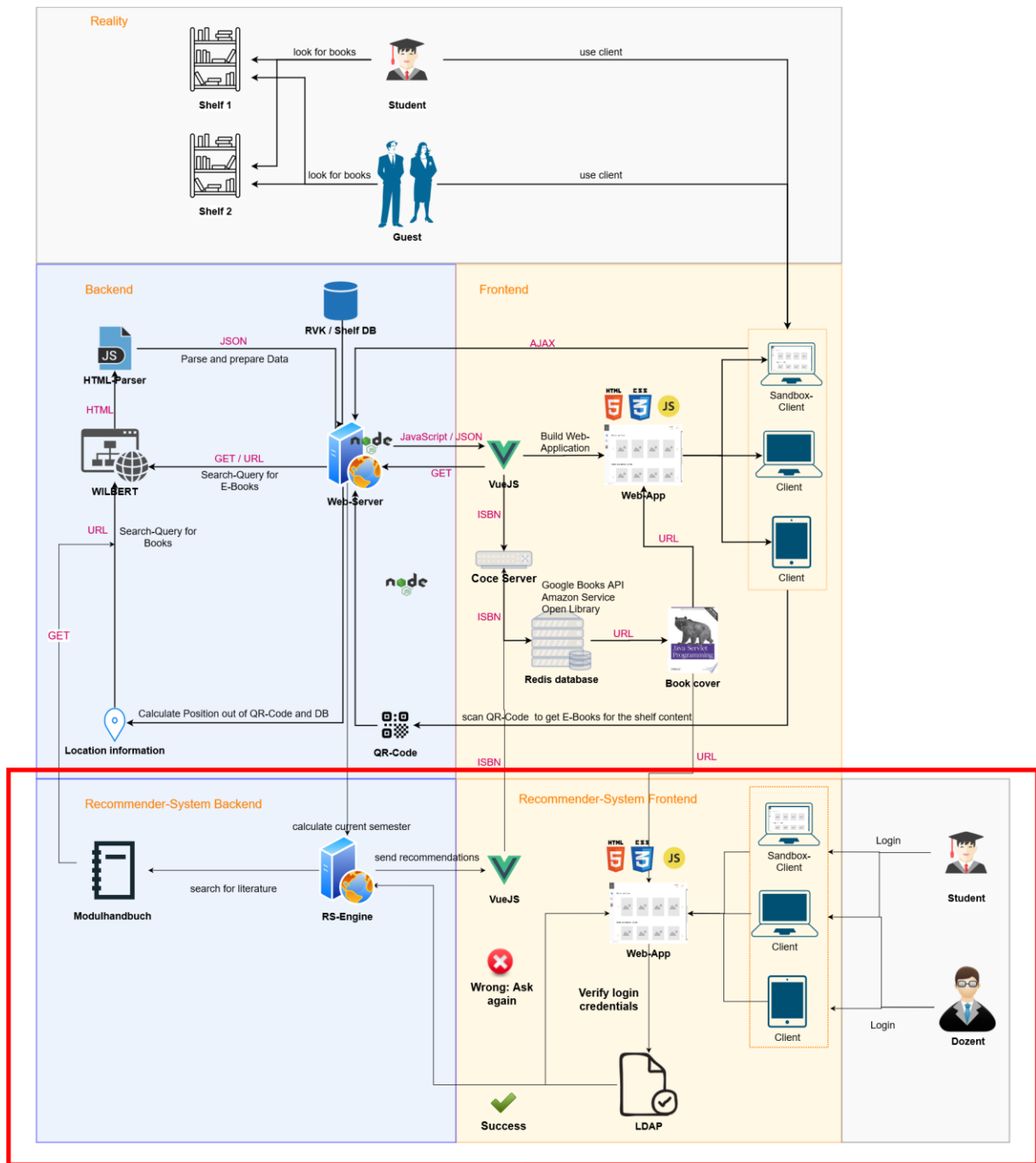


Abbildung 12: Architektur des bestehenden Prototyps  
 Quelle: Attia (2020a) [Abbildung 5.1.1: Architektur des Systems]

### 4.3.2.1 Frontend

*„Das Frontend zählt zu den wichtigsten Komponenten des Systems, da hier alle Interaktionen des Nutzers vorgenommen werden [...]“<sup>134</sup>*

Es ist verantwortlich für die Interaktion mit dem Nutzer und stellt somit die entscheidende Nutzerschnittstelle dar. Mit Hilfe des Frontends soll ein *UI* erstellt werden, welches die in Kapitel 4.3.1 beschriebenen Funktionalitäten für den Nutzer in ansprechender Weise darstellt und zugänglich macht. Darüber hinaus steht es in Kommunikation mit verschiedenen Endpunkten des im Rahmen des Backend implementierten Node-Servers.

Die technische Umsetzung des Frontends erfolgte mit Hilfe des JavaScript-Frameworks *Vue.js* „(pronounced /vju/, like view)“<sup>135</sup>.

*„Der Kern von Vue.js konzentriert sich auf den View-Layer einer Webanwendung und ist daher besonders performant und schlank. Durch zahlreiche Erweiterungen kann [es] aber auch als vollumfängliches Web Application Framework für die Entwicklung komplexer Single-Page-Anwendungen genutzt werden.“<sup>136</sup>*

Gerade der progressive Grundgedanke und die „umfangreiche und einfach zu verstehende Dokumentation [...] des Frameworks sind die Hauptgründe der großen Beliebtheit von *Vue.js* in der Community.“<sup>137</sup>

*Vue.js* folgt dem Prinzip des Separation of Concerns. Ziel dabei ist, dass eine Komponente für genau eine Aufgabe verantwortlich ist und somit die Zuständigkeiten der einzelnen Komponenten strikt voneinander getrennt sind und diese universell einsetzbar sind. Abgeschlossene Komponenten werden durch den Grundsatz der Single File Components erreicht. Dabei werden HTML, CSS und JavaScript in eine Datei geschrieben und nicht wie bei anderen Frameworks getrennt abgelegt. Kleine, abgeschlossene Komponenten lassen sich dadurch gut testen, warten und leichter erweitern.<sup>138</sup>

Wie bereits beschrieben, ist *Vue.js* selbst ein sehr schlankes Framework, lässt sich jedoch durch zahlreiche Pakete erweitern. Auch für den Prototyp wurden verschiedene Erweiterungen genutzt:

- *Vuetify*

ist ein UI-Framework mit zahlreichen Komponenten zur Umsetzung von Googles Material Design Spezifikation, welches auf *Vue.js* aufsetzt.<sup>139</sup>

*„Vuetify takes a mobile first approach to design which means your application just works out of the box – whether it’s on a phone, tablet, or desktop computer.“<sup>140</sup>*

---

<sup>134</sup> Vgl. Grätz (2020b): Praktikumsbericht iLibrary, S. 23.

<sup>135</sup> Ebd.; Vgl. *Vue.js*(n. N.): Introduction, <https://vuejs.org/v2/guide/>, 16.05.2021.

<sup>136</sup> Vgl. Peterke (2019): *Vue.js kurz & gut*, Klappentext.

<sup>137</sup> Vgl. Böhner (2019): Einführung in *Vue.js*: Was steckt wirklich hinter dem progressiven Framework?, <https://www.hosteurope.de/blog/einfuehrung-in-vue-js-was-steckt-wirklich-hinter-dem-progressiven-framework/>, 03.07.2021.

<sup>138</sup> Ebd.

<sup>139</sup> Vgl. *Vuetify* (n. N.): Introduction: What is *Vuetify*, <https://vuetifyjs.com/en/introduction/why-vuetify/>, 03.07.2021.

<sup>140</sup> Ebd.



- Vuex

stellt ein Modul zur zentralen Datenspeicherung dar (auch als Store bezeichnet), welche zwischen den verschiedenen Komponenten ausgetauscht werden

- Router

„dient der Navigation zwischen den einzelnen „views“ bzw. Seiten der Web-App“<sup>141</sup>

- I18n

ist ein Tool zu Unterstützung bei der Implementierung der Mehrsprachigkeit

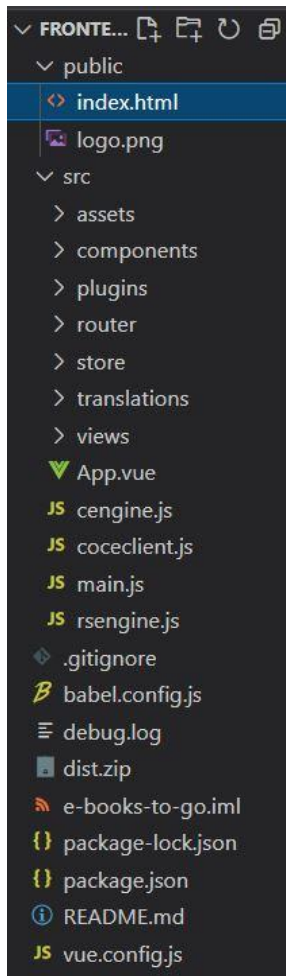


Abbildung 13: Struktur des Frontends in der IDE Visual Studio Code

Als Ausgangspunkt der gesamten Anwendung dient die Datei „index.html“ im Verzeichnis public. Darin finden sich neben der Definition des Grundgerüsts der Website vor allem der Einstiegspunkt für die Web-App - `<div id=“app“></div>`.<sup>142</sup> In den Dateien main.js und App.vue erfolgt die Definition der eigentlichen „App“ als neue Vue-Instanz. Darüber hinaus wird hier das Grundgerüst für die App definiert, welches Header und Navigationsmenü umfasst. Auch die Router-View, welche „für die einzelnen Seiten der Web-App zuständig ist und diese lädt“<sup>143</sup>, wird hier eingebunden.

Das eigentliche Projekt setzt sich aus Views (Seiten) und Components (Komponenten) zusammen. Views bilden dabei die einzelnen Seiten der Web-App ab, z.B.:

- /home => öffnet zugehörige View zur Anzeige der Startseite/ des Dashboards der neu eingetroffenen E-Books
- /findEBooks => öffnet zugehörige View zur Anzeige der zugehörigen Seite mit eingebundenem QR-Code-Reader
- /SearchForEBooks => öffnet zugehörige View, welche die Suche nach verschiedenen Themenbereichen ermöglicht
- /book => öffnet die View, in der die Detailansicht abgebildet ist
- /help => öffnet die View, in der die Hilfeseite implementiert ist

Das Wechseln zwischen den verschiedenen Seiten der Web-App wird durch den Vue-Router ermöglicht. In ihm sind die Routen zu den einzelnen Views definiert.

Die Views wiederum setzen sich aus verschiedenen Komponenten zusammen, welche für die Umsetzung der Funktionalitäten der einzelnen Seiten verantwortlich sind (z.B. Language-Picker, Book-Slider, Book-Grid).

Durch Modularisierung können die einzelnen Komponenten mehrfach in verschiedenen Views genutzt

<sup>141</sup> Vgl. Grätz (2020b), S. 24.

<sup>142</sup> Vgl. Grätz (2020a), S. 64.

<sup>143</sup> Vgl. Grätz (2020b), S. 25.

und somit doppelter Code vermieden werden.<sup>144</sup> Darüber hinaus wird dadurch eine verbesserte Wartbarkeit erreicht.<sup>145</sup>

Über den Komponenten steht der Vue-Store (Vuex):

*„Hier werden zentral verschiedene Variablen und Arrays gespeichert, welche in verschiedenen Komponenten benötigt werden. Damit sind die Daten konsistent gespeichert und abrufbar.“<sup>146</sup>*

Er ist dafür verantwortlich, *„die Variablen und deren Inhalt[e] auch in anderen Komponenten bekannt zu machen.“<sup>147</sup>* Mit Hilfe von Gettern und Settern können die Werte gelesen oder gesetzt werden.<sup>148</sup>

Wichtig für die Mehrsprachigkeit der App sind die Sprachdateien im Verzeichnis „translations“, welche alle zu übersetzenden Elemente enthalten und mit Hilfe des i18n-Pakets integriert werden. In den Skripten wird mit Platzhaltern gearbeitet, bspw. `{{ $t('name') }}`, die dann automatisch mit der ausgewählten Sprache besetzt werden.<sup>149</sup>

Angaben zum Layout der App sind größtenteils in eine CSS-Datei ausgelagert. Hier finden sich die Definitionen zu einzelnen Elementen der Komponenten.<sup>150</sup> Generell wird mit Hilfe von Material Design-Elementen gearbeitet, welche über die Einbindung der Vuetify-Bibliothek genutzt werden können.

Auch die Login-Funktionalität ist im Frontend umgesetzt. Nach Eingabe von Nutzernamen und Passwort wird versucht, eine Authentifizierung durchzuführen, indem eine Axios-Get-Anfrage an den zentralen Verzeichnisdienst der TH Wildau via LDAP (Lightweight Directory Access Protocol) gesendet wird. Bei erfolgreicher Authentifizierung werden die für die weiteren Funktionen der App benötigten Daten in den Store übernommen. Die Benutzerdaten werden zusätzlich im lokalen Speicher des Browsers gesichert, sodass die Daten auch nach Neuladen der App zur Verfügung stehen.<sup>151</sup>

Wurde im allerersten Prototyp die Abfrage verschiedener APIs (mittels ISBN) nach Buchcovern über eine Anfrage an einen Coce-Server umgesetzt, werden nun zwei APIs (Google-Books-API und Amazon-http-Service) vom Frontend aus einzeln angefragt. Bei Nutzung des Coce-Servers wäre zusätzlich die Open-Library-Books-API angefragt worden. Aufgrund der aufwendigen Implementation des Coce-Servers wurde sich jedoch für eine manuelle Umsetzung der Abfrage der verschiedenen APIs und gegen den Einsatz des Coce-Servers entschieden.

---

<sup>144</sup> Ebd.

<sup>145</sup> Vgl. Grätz (2020a), S. 65.

<sup>146</sup> Ebd., S. 66.

<sup>147</sup> Vgl. Grätz (2020b), S. 25.

<sup>148</sup> Ebd.

<sup>149</sup> Ebd., S. 24.

<sup>150</sup> Ebd.

<sup>151</sup> Vgl. Attia (2020a): Entwicklung eines Recommender-Systems für E-Book-Empfehlungen in der Hochschulbibliothek der TH Wildau, S. 39.

### 4.3.2.2 Backend

Das Backend besteht aus einem Node.js-Server und einer Datenbank.

*„Node.js ist eine serverseitige, eventbasierte JavaScript-Laufzeitumgebung, die von Entwickler Ryan Dahl als Anwendung für die Entwicklung von skalierbaren Netzwerkanwendungen konzipiert wurde.“<sup>152</sup>*

Der Node-Server selbst verwendet das Framework express.js, welches die Umsetzung eines Webservers wesentlich vereinfacht.<sup>153</sup>

Als Datenbank wird die NoSQL-Datenbank MongoDB verwendet. Dort sind die Verknüpfungen von auf dem QR-Code hinterlegten Regal-IDs und zugehörigen RVK-Notationen in einem JSON-ähnlichen Format hinterlegt.

Hauptaufgabe des Node-Servers ist die Herstellung einer Verbindung zu Wilbert, um passende E-Books sowie deren Metadaten zu ermitteln. Da zum Zeitpunkt der Erstellung des Prototyps noch keine passende API zum Auslesen dieser zur Verfügung stand, „wird von Wilbert nur der HTML-Content geladen und geparkt.“<sup>154</sup> Nach Aufbereitung der Daten werden diese an das Frontend zurückgesendet.

Der Parser für die Metadaten stellt die wichtigste Komponente im Backend dar. Zum Abruf der Daten werden je nach Funktionalität die entsprechenden Wilbert-URLs aufgerufen. Der Aufbau der URLs folgt dabei einem festen Schema:


Der Einstiegspunkt stellt die Adresse von Wilbert dar: <<https://wilbert.kobv.de>>. An diesen wird die Suchmethode (simple oder advanced) ergänzt. Gefolgt wird diese vom Suchbegriff, eingeleitet durch <fq=...>. Darüber hinaus können weitere Parameter wie bspw. *hitsPerPage* für die Anzahl der Treffer pro Seite oder *sortOrder* und *sortCrit* zu Angaben zur Sortierreihenfolge der Treffer, angehängt werden.

Eine Recherche nach einem Titel kann bspw. über die einfache Suche umgesetzt werden. Eine Wilbert-Anfrage sieht dann folgendermaßen aus:

*[https://wilbert.kobv.de/simpleSearch.do?fq=datenlieferant%3Afhw\\_ebooks&fq=search\\_space%3Abooks&query=test&plv=2](https://wilbert.kobv.de/simpleSearch.do?fq=datenlieferant%3Afhw_ebooks&fq=search_space%3Abooks&query=test&plv=2)*

Für die Abfrage von E-Books zu einer bestimmten RVK-Notation wird hingegen eine erweiterte Suche ausgeführt:

*[https://wilbert.kobv.de/advancedSearch.do?page=1&sortCrit=score&sortOrder=desc&hitsPerPage=50&fq=search\\_space%3Abooks&fq=datenlieferant%3Afhw\\_ebooks&fl=rvk&v1=ST+250](https://wilbert.kobv.de/advancedSearch.do?page=1&sortCrit=score&sortOrder=desc&hitsPerPage=50&fq=search_space%3Abooks&fq=datenlieferant%3Afhw_ebooks&fl=rvk&v1=ST+250)*

Zum Abrufen der HTML-Daten wird auf dem Server die JavaScript-Bibliothek Axios genutzt. Zum Parsen wird das cheerio-Modul genutzt, welches auf JQuery  aufsetzt „und damit ein einfaches und schnelles [P]arsen ermöglicht.“<sup>155</sup> Je nach im Frontend aufgerufener Funktionalität werden die entsprechenden Abfragen in Wilbert ausgeführt und die benötigten Daten abgerufen.

---

<sup>152</sup> Vgl. Arocom (n. N.): Fachbegriffe einfach erklärt : Node.js, <https://www.arocom.de/fachbegriffe/webentwicklung/nodejs>, 03.07.2021.

<sup>153</sup> Vgl. Grätz (2020b), S. 26.

<sup>154</sup> Ebd., S. 20.

<sup>155</sup> Ebd., S. 26.

Um passende E-Books zum Regalbereich zu erhalten, stellt das Frontend eine Anfrage an den Endpunkt „/shelf/:shelfid“. Die durch den QR-Code-Reader des Frontend gesammelten Informationen zu Regal-ID und der Art der verwendeten Systematik werden vom Frontend aus an das Backend übermittelt. Das Backend extrahiert die Regal ID und führt mit Hilfe dieser eine Anfrage an die Mongo-Datenbank im Backend aus. Mit Hilfe der ermittelten RVK-Notation wird schließlich die Wilbert-Abfrage ausgeführt und die geparsten und aufbereiteten Daten ans Frontend zurückgesendet.

Ein weiterer Endpunkt ist bspw. „/newbooks“. Wird dieser Endpunkt angesprochen, wird vom Backend aus folgende Wilbert-Abfrage ausgeführt:

```
https://wilbert.kobv.de/latestBooks.do?fq=datenlieferant%3afhw_ebooks&fq=search_space%3a  
books&index=internal&plv=2
```

Auch die Datei „rsengine“ (Recommender-System-Engine) aus dem Vue-Framework beinhaltet Funktionalitäten, die dem Backend zuzurechnen sind: Sie beinhaltet Methoden für die Berechnung des aktuellen Semesters, in dem sich ein Student gerade befindet und die Generierung/Extraktion der E-Book-Empfehlungen aus den Modulhandbüchern.

## **4.4 Zusammenfassung**

Wie sich gezeigt hat, steht die Hochschulbibliothek der TH Wildau neuen technologischen Entwicklungen stets offen gegenüber. Gerade durch die enge Zusammenarbeit mit der Hochschule (hier vor allem das iCampus-Team) können neue Projekte umgesetzt werden. So wurde bereits ein Prototyp einer App zur Überwindung des Medienbruchs am Regal entwickelt: „E-Books To Go“. Dieser bietet verschiedene Funktionen wie die Anzeige der „neuesten“ E-Books oder E-Books passend zur RVK-Notation des Regals (nach Scannen eines QR-Codes). Darüber hinaus sind auch schon Recommender-Funktionen untergebracht. Die Beschreibung der technischen Architektur soll dabei helfen, technische Anforderungen zu generieren. Kenntnisse dieser sind Voraussetzung für die Implementation neuer Funktionen wie der Anzeige entliehener Bände.

Als hochschulweites Informationssystem hat sich Unidos bewährt. Auch hier finden sich bibliotheksbezogene Funktionen. Als bibliotheksspezifisches Recherchesystem steht vor allem das Discovery-System Wilbert zur Verfügung. Beim Stöbern am Regal soll sich jedoch die zu entwickelnde Anwendung als Hilfsmittel bewähren.

## 5 Anforderungsanalyse/Requirements Engineering

Im folgenden Kapitel sollen mit Hilfe der Erstellung von Personas Anforderungen an die zu entwickelnde Anwendung ermittelt und analysiert werden, die später im Rahmen des Konzepts näher ausgeführt werden. Hierbei geht es zunächst nur um die Formulierung fachlicher Anforderungen, aus denen später technische Lösungsansätze erarbeitet werden.

*„Ziel ist das Finden aller relevanten Anforderungen für die Entwicklung eines Produkts mit der Motivation, das Produkt effizient und möglichst ohne Fehler zu entwickeln.“<sup>156</sup>*

### 5.1 Stakeholder

Zunächst sollen dabei die Stakeholder erläutert werden. Stakeholder „sind die relevanten Beteiligten, Betroffenen, Nutzer und Interessierten an der Software(entwicklung). [...] Sie liefern direkt oder indirekt Informationen über die Ziele, die Rahmenbedingungen, den Kontext, die Anforderungen und auch die Risiken des Produkts bzw. Systems.“<sup>157</sup> Niebisch bezeichnet Stakeholder als „Anforderungsinhaber“<sup>158</sup>.

Die Zielgruppe der Anwendung wird in erster Linie durch die Nutzer der Bibliothek der Technischen Hochschule Wildau präsentiert. Angesprochen werden sollen also "Studierende und Angehörige"<sup>159</sup> der Hochschule. Somit stellen diese einen Stakeholder dar. Zur Zielgruppe der Angehörigen können bspw. Dozenten und Mitarbeiter gezählt werden. Repräsentiert werden die Studierenden im Rahmen der Arbeit durch Christopher Grätz und Omar Attia. Beide sind Master-Studenten an der TH Wildau und verantwortlich für die Entwicklung des bestehenden Prototyps der App.

Darüber hinaus zählen die Mitarbeiter der Hochschulbibliothek zu den Stakeholdern. Für die Anforderungsanalyse vertrat Dr. Frank Seeliger als Leiter der Hochschulbibliothek deren Anforderungen. Mitarbeiter und Leitung der Bibliothek verfügen über zahlreiche Kenntnisse über das Rechercheverhalten ihrer Nutzer. Zudem ist es auch das Bibliothekspersonal, welches im aktiven Einsatz der App mit Fragen und Problemstellungen der Nutzer konfrontiert sein wird. Besonderer Ansprechpartner aus der Gruppe der Mitarbeiter der Bibliotheksmitarbeiter stellt Jan Kissig dar, der für die IT-Betreuung der Bibliothek verantwortlich ist. Er trägt unter anderem die Verantwortung für die Administration der Bibliotheksserver und ist später für die Installation des Systems verantwortlich.

Als dritte Gruppe der Stakeholder ist das iCampus-Team der Technischen Hochschule Wildau zu nennen, welche "Studium und Lehre an der Technischen Hochschule Wildau mit unterschiedlichen Angeboten [unterstützen], die Studierenden und Mitarbeitern relevante Informationen zum Hochschulleben zugänglich machen."<sup>160</sup> Zum Team gehören unter anderem Prof. Dr. Jannett Mohnke und Alfredo Azmitia. Die

---

<sup>156</sup> Vgl. Grande (2011): 100 Minuten für Anforderungsmanagement, S. 11.

<sup>157</sup> Vgl. Balzert (2009): Lehrbuch der Software-Technik, S. 504.

<sup>158</sup> Vgl. Niebisch (2013): Anforderungsmanagement in sieben Tagen, S. 7.

<sup>159</sup> Vgl. TH Wildau (2021).

<sup>160</sup> Vgl. TH Wildau | iCampus (n. N.): iCampus Wildau, <https://icampus.th-wildau.de/icampus/>, 05.05.2021.

Anforderungen des iCampus Wildau-Teams beziehen sich vor allem auf technische Anforderungen sowie Unterstützung bei der Nutzung und Bereitstellung bestimmter Schnittstellen.

## 5.2 Erstellung der Personas

Zur Annäherung an die Zielgruppen sollen nun Personas erstellt werden. Im Rahmen dieser Arbeit sollen dafür Proto-Personas genutzt werden. Diese werden auf Grundlage des „*vorhandenen Wissens über die Zielgruppe erstellt*.“<sup>161</sup> Im Mittelpunkt stehen dabei vor allem die Bedürfnisse eines „archetypischen Nutzer[s], welcher für eine Nutzergruppe mit ähnlichen Anforderungen an den Funktionsumfang und das Interaktionsdesign steht.“<sup>162</sup> Proto-Personas können eine Hilfestellung bei der Priorisierung von Entwicklungszielen sein.<sup>163</sup> Betrachtet werden können Kategorien wie persönliche Attribute, Kontext, Ziele, Aufgaben und Frustrpunkte.<sup>164</sup>

### 5.2.1 Persona 1 – Student

Maximilian Maus, 20 Jahre alt, ist Bachelorstudent der Telematik im vierten Semester an der TH Wildau. Er nutzt die Bibliothek und ihre Angebote, um Literatur für Hausarbeiten und Prüfungsvorbereitungen auszuleihen. Vor Corona nutzte er auch die Arbeitsplätze in der Bibliothek und stöberte zwischen den Regalen. Dabei ließ er sich gern von den Titeln rechts und links vom gesuchten Band inspirieren, um neue Ideen für seine Arbeiten gewinnen zu können. Mit zunehmender Zahl online zur Verfügung stehender Ressourcen, die durch die Bibliothek angeboten werden, bemängelt Maximilian, dass ihm diese beim Stöbern am Regal nicht zur Verfügung stehen. Momentan muss er eine erneute Recherche in Wilbert durchführen.

Ab und an tauscht er sich mit seinen Kommilitonen über gelesene Literatur aus. Er profitiert gern von Literaturempfehlungen, die er durch seine Mitstudenten oder Dozenten erhält. Die Modulhandbücher empfindet er allerdings als sehr lang und nicht sonderlich attraktiv zum Lesen. Ihm ist gar nicht bewusst, dass diese auch Literaturempfehlungen für die für sein Studium zu belegenden Module enthalten.

Durch sein Studium arbeitet Maximilian tagtäglich mit dem Laptop. Auf dem Campus nutzt er vor allem die Unidos-App, zum Lesen von E-Mails, Verlängern von Ausleihen aus der Bibliothek oder um das Mittagsangebot in der Mensa ausfindig zu machen.

### 5.2.2 Persona 2 – Dozentin

Jochen Muster ist 45 Jahre alt und bereits seit einigen Jahren Dozent im Bereich Datenbanken an der TH Wildau. Er nutzt neben dem Online-Katalog die Bibliothek vor Ort, um sich einen Überblick über neue

---

<sup>161</sup> Vgl. t2informatik (n. N.): Personas: Die Entwicklung und kontinuierliche Verwendung, <https://t2informatik.de/wissen-kompakt/personas/>, 03.07.2021.

<sup>162</sup> Vgl. Steimle; Wallach (n. N.): Proto Persona, <http://www.collaborative-uxdesign.com/scoping/protoperonas>, 03.07.2021.

<sup>163</sup> Vgl. t2informatik (n. N.).

<sup>164</sup> Vgl. Steimle; Wallach (n. N.).

Literatur zu einem bestimmten Fachgebiet anzusehen. Er genießt die Atmosphäre in der Bibliothek und stöbert gern zwischen den Regalen.

Jochen hat sein Laptop und sein Smartphone immer bei sich. Auf seinem Laptop sind Präsentationen für seine Seminare gespeichert, welche er für die Studenten auf der Lernplattform Moodle bereitstellt. Darüber hinaus ist er begeisterter Nutzer der Unidos-App. Über diese ruft er bspw. regelmäßig seine E-Mails ab.

Er stellt häufig fest, dass die Studenten keine Kenntnis von den Literaturempfehlungen in den Modulhandbüchern haben. Dabei stellen diese aus seiner Sicht wichtige Hilfsmittel für Projektarbeiten und Prüfungsvorbereitungen dar.

### **5.2.3 Persona 3 – Bibliotheksmitarbeiter**

Linus Leitner ist 47 Jahre alt und Mitarbeiter der Bibliothek der Technischen Hochschule Wildau. Er hegt großes Interesse daran, neue Services für die Nutzer der Bibliothek umzusetzen. Zu diesem Zweck haben er und sein Team in Vergangenheit bereits häufig mit dem iCampus-Team der TH Wildau kooperiert. Gern beobachtet er, wie die Studenten zwischen den Regalen entlangstöbern und sich zusätzlich zum gesuchten Band weitere Titel anschauen und für ihre Arbeiten nutzen.

Als Leiter der Bibliothek ist er unter anderem für das Controlling verantwortlich. Dabei hat er festgestellt, dass die Nutzungszahlen der E-Ressourcen der Bibliothek nicht in gleichem Maße zunehmen wie die Erwerbungszahlen in diesem Bereich. Deshalb würde er sich wünschen, dass auch in der Bibliothek selbst mehr auf diese aufmerksam gemacht wird.

Er bemängelt zudem, dass die Studierenden beim Stöbern am Regal keinen vollständigen Überblick über die Bestände der Bibliothek der TH Wildau erhalten. Einerseits fallen E-Ressourcen beim Stöbern am Regal aus dem Sichtfeld, aber auch entliehene Bände sind nicht sichtbar. Er würde sich eine Möglichkeit wünschen, dass die Nutzer der Bibliothek beim Stöbern am Regal einen Überblick über den gesamten Bestand der Bibliothek zu einem bestimmten Themengebiet erhalten könnten, inklusive elektronischer Ressourcen und ausgeliehener Bände. Für die Anzeige entliehener Bände würde er sich zudem eine direkte Möglichkeit wünschen, die Bände vormerken zu können, ohne eine erneute Recherche in Wilbert ausführen zu müssen.

## **5.3 Use-Cases**

Um die Anforderungen der verschiedenen Stakeholder zu bündeln, sollen im Folgenden Use-Cases für die Anwendung definiert werden.

Das System soll die Anzeige von E-Ressourcen passend zum Regalstandort des Nutzers ermöglichen, um so das Stöbern am Regal durch die Anzeige derer zu ergänzen.

Für E-Ressourcen soll im Falle von E-Books die Möglichkeit einer Durchschaltung direkt zum PDF ohne Umweg über die Verlagsseite möglich sein.

Das System soll eine Anzeige entliehener Bände ermöglichen, um auch im gedruckten Bereich einen möglichst vollständigen Bestandsüberblick zu einem bestimmten Thema zu ermöglichen.

Bei entliehenen Bänden soll zudem die Möglichkeit gegeben sein, diese direkt aus der Anwendung heraus vormerken zu können.

Für alle Listen soll die Möglichkeit bestehen, auch nach Absenden der Suchabfrage verschiedene Filter- und Sortierungen anzuwenden.

Darüber hinaus soll das System über Recommender-Funktionalitäten verfügen:

Das System soll

... dem Nutzer passend zum eigenen Studiengang Literaturhinweise (modulbasiert) anzeigen

... dem Nutzer die Möglichkeit bieten, sich Literaturhinweise aus dem Modulhandbuch zu anderen Studiengängen anzeigen zu lassen

... dem Nutzer die Möglichkeit geben, E-Ressourcen für bestimmte Module zu empfehlen

## 5.4 Anforderungen

Anhand der vorliegenden Use-Cases und Personas sollen nun Anforderungen formuliert werden.

Bei der Erstellung der Anforderungen muss darauf geachtet werden, dass es sich bei Apps um Fokussierungssysteme handelt:

*„Sie fokussieren sich auf eine bestimmte Problemlösung und wollen diese besonders gut umsetzen. Sie ermöglichen dem Nutzer nicht den Zugang zu allen möglichen Optionen, sondern sie reduzieren diese. Sucht der Nutzer nach einer anderen Problemlösung, gibt es dafür – wenn möglich – eine andere App.“<sup>165</sup>*

Das heißt, dass bei der Erstellung der App genau darauf geachtet wird, dass im Umfang der Funktionalitäten kein Konkurrenzprodukt zum Discovery-System Wilbert entworfen werden, sondern eine App für eine spezifische Zielstellung entwickelt werden soll. Hauptzielstellung hier ist die Unterstützung beim Stöbern am Regal. Für spezifischere Recherchen kann ggf. auf Wilbert verwiesen werden.

Unterschieden werden soll zwischen funktionalen und nicht-funktionalen Anforderungen.

*„Eine funktionale Anforderung legt eine vom Softwaresystem oder einer seiner Komponenten bereitstellende Funktion oder bereitzustellenden Service fest.“<sup>166</sup> Nicht funktionale Anforderungen hingegen „beschreiben Aspekte, die typischerweise mehrere oder alle funktionale Anforderungen betreffen bzw. überschneiden (cross-cut).“<sup>167</sup>*

Zum Teil ist jedoch eine strikte Unterscheidung nicht möglich, sodass es zu Überschneidungen kommen kann.

Im Rahmen der Anforderungen wird im Bereich der Priorisierung zwischen muss, kann und soll unterschieden. Bei Priorisierungen mit Kleinbuchstaben am Ende handelt es sich um *eine* technische Umsetzungsmöglichkeit. Diese sind in der Priorisierung mit soll gekennzeichnet und sollen im Rahmen des (zu erweiternden) Prototyps Anwendung finden. Andere Umsetzungsmöglichkeiten sind hier langfristig denkbar. Die genauere Ausführung der Anforderung findet sich im Konzept (siehe Kapitel 6).

---

<sup>165</sup> Vgl. Deeg (2018): Die nächste digitale Welle. In: *BuB*, Jg. 70, Nr. 05, S. 248–250, S. 248.

<sup>166</sup> Vgl. Balzert (2009), S. 456.

<sup>167</sup> Ebd., S. 463.



### 5.4.1 Anforderungen funktional

ID	Name	Beschreibung	Priorität
F_01	Ortungsfunktion/Nutzerstandort ermitteln	Das System muss den Nutzerstandort im Verhältnis zu bestimmten Bibliotheksregalen ermitteln.	muss
F_01a	Ortungsfunktion/Nutzerstandort ermitteln	Das System soll den Nutzerstandort im Verhältnis zu bestimmten Bibliotheksregalen mit Hilfe von QR-Codes ermitteln.	soll
F_02	QR-Code-Generator	Das System soll durch ein Tool zur Erstellung von QR-Codes unterstützt werden.	soll
F_03	Anbringung QR-Codes	Die QR-Codes müssen auf eine Art am Regal angebracht werden, dass sie dem Nutzer ins Auge fallen und nicht übersehen werden.	muss
F_04	Zum Nutzerstandort passende E-Ressourcen ermitteln	Das System muss zum Nutzerstandort (thematisch) passende E-Ressourcen <i>ermitteln</i> .	muss
F_05	Zum Nutzerstandort passende E-Ressourcen anzeigen	Das System muss die zum Nutzerstandort passenden E-Ressourcen in geeigneter Form <i>anzeigen</i> .	muss
F_06	Entlehene (gedruckte) Monographien ermitteln	Das System muss die zu einer RVK-Notation gehörenden entlehene Exemplare <i>ermitteln</i> .	muss
F_07	Entlehene (gedruckte) Monographien anzeigen	Das System muss die zu einer RVK-Notation gehörenden entlehene Exemplare anzeigen.	muss
F_08	Vormerkfunktion	Entlehene (gedruckte) Bände müssen vorgemerkt werden können.	muss
F_09	Vormerkfunktion	Das System muss dem Nutzer eine Rückmeldung geben, sollte bei der Vormerkung ein Fehler aufgetreten sein.	muss
F_10	Neueste E-Books in der Bibliothek ermitteln	Das System muss die durch die Bibliothek der TH Wildau neu erworbenen E-Books <i>ermitteln</i> .	muss
F_11	Neueste E-Books in der Bibliothek anzeigen	Das System muss die Möglichkeit bieten, die durch die Bibliothek neu erworbenen E-Books <i>anzuzeigen</i> .	muss
F_12	Authentifizierung	Das System muss die Möglichkeit bieten, dass Nutzer sich mit ihrer Hochschulkennung und dem dazugehörigen Passwort authentifizieren können.	muss
F_12a	Authentifizierung	Für die Nutzerauthentifizierung soll das System den zentralen Verzeichnisdienst der TH Wildau über LDAP verwenden.	soll
F_13	Authentifizierung	Das System muss dem Nutzer eine Rückmeldung geben, sollte bei der Authentifizierung ein Fehler aufgetreten sein.	muss
F_14	Dashboard	Beim Starten der App soll ein Dashboard präsentiert werden, welches die neuesten E-Books im Bestand der Bibliothek anzeigt.	soll
F_15	Authentifizierung	Nach Authentifizierung sollen auf dem Dashboard Literaturempfehlungen aus dem Modulhandbuch für das aktuelle Semester angezeigt werden.	soll
F_16	Präsentation der Treffer	Das System soll die gefundenen Medien in einer Trefferliste mit ihrem Cover präsentieren.	soll
F_17	Präsentation der Treffer	Unterschiedliche Medientypen sollen voneinander unterscheidbar abgebildet werden.	soll

ID	Name	Beschreibung	Priorität
F_18	Filtermöglichkeit	Das System soll die Möglichkeit bieten, die Ergebnisse anhand von Filtern einzugrenzen.	soll
F_19	Rechercheverlauf	Das System kann die Möglichkeit bieten, eine Suchhistorie abzubilden.	kann
F_20	Sortierung der Treffer	Das System bietet verschiedene Möglichkeiten, die Treffer zu sortieren, mindestens <ul style="list-style-type: none"> <li>- nach Autor</li> <li>- nach Titel</li> <li>- nach Erscheinungsjahr</li> </ul>	soll
F_21	Anzeige der Gesamtzahl der Treffer	Zu den Trefferlisten wird die Gesamtzahl der Treffer angezeigt.	soll
F_22	Detailansicht	Das System bietet zu jedem Medium eine Detailansicht, welche über einen Klick auf einen Treffer in der Liste zu erreichen ist. Diese beinhaltet Informationen zu Titel, Untertitel, Autor, Verlag und Erscheinungsjahr.	muss
F_23	Detailansicht	Zusätzlich zu den bibliographischen Informationen (siehe F_16) kann die Detailansicht eine Kurzbeschreibung des Inhalts beinhalten	kann
F_24	Navigation	Das System muss eine übersichtliche Navigation zum Erreichen der verschiedenen Funktionalitäten an zentraler Stelle beinhalten.	muss
F_25	Durchschaltung	Das System soll die Möglichkeit bieten, ohne Umleitung über die Verlagsseite, direkt zum Volltext von E-Books zu gelangen.	soll
F_26	Versenden von Empfehlungen per E-Mail	Das System soll nach Authentifizierung die Möglichkeit bieten, einzelne Titel als Empfehlung per E-Mail zu versenden.	soll
F_27	Literaturempfehlungen aus dem Modulhandbuch ermitteln	Das System soll Literaturempfehlungen aus dem Modulhandbuch, die sich im Bestand der TH Wildau befinden, <i>ermitteln</i> .	soll
F_28	Literaturempfehlungen aus dem Modulhandbuch anzeigen	Das System soll Literaturempfehlungen aus dem Modulhandbuch, die sich im Bestand der TH Wildau befinden, <i>anzeigen</i> .	soll
F_29	Einzelne Titel bewerten	Das System soll nach Authentifizierung die Möglichkeit bieten, einzelne Titel zu bewerten.	soll
F_30	Empfehlungen von Nutzer speichern	Das System kann Empfehlungen von Nutzern speichern.	kann
F_31	Empfehlungen anderer Nutzer ermitteln	Das System kann Empfehlungen, die andere Nutzer im System gegeben haben, <i>ermitteln</i> .	kann
F_32	Empfehlungen anderer Nutzer anzeigen	Das System kann Empfehlungen, die andere Nutzer im System gegeben haben, <i>anzeigen</i> .	kann
F_33	Thematische Suche	Das System soll die Möglichkeit bieten, einfache thematische Suchen ausführen zu können.	soll
F_34	Anzeige leerer Trefferlisten	Das System muss dem Nutzer anzeigen, wenn keine Treffer zu einer Anfrage gefunden wurden.	muss
F_35	Anzeige von Ladezeiten	Das System muss dem Nutzer Ladezeiten des Systems anzeigen.	muss
F_36	Tutorial mit Wilma	Wilma kann ein Tutorial für die Nutzung der Anwendung bereitstellen.	kann

ID	Name	Beschreibung	Priorität
F_37	Verknüpfung zum vorherigen/nächsten Treffer aus der Liste	Das System muss die Möglichkeit bieten, eine Navigation von einem Treffer zum nächsten anzubieten, ohne erneut in die Trefferliste gehen zu müssen.	muss

Tabelle 4: Funktionale Anforderungen

#### 5.4.2 Anforderungen nicht funktional

ID	Name	Beschreibung	Priorität
NF_01	Web-App	Das System muss als Web-App konzipiert und umgesetzt sein, sodass eine Integration in bspw. die Unidos-App und das 3D-System möglich ist.	muss
NF_02	Responsivität	Das System muss sowohl auf PCs/Laptops, als auch auf Smartphones und Tablets benutzbar sein.	muss
NF_03	Anpassbarkeit an Corporate Design der TH Wildau	Das System muss so gestaltet sein, dass es als Produkt der TH Wildau erkennbar ist.	muss
NF_04	Antwortzeitverhalten	Das System sollte nicht länger als fünf Sekunden zum Laden einer Seite benötigen.	soll
NF_05	Skalierbarkeit	Der Aufbau des Systems muss modular erfolgen, damit neue Funktionalitäten einfach ergänzt werden können.	muss
NF_06	Bedienungskonzept	Das System muss den Nutzer in maximal zwei Navigationsschritten zum Ziel führen.	soll
NF_07	Separates CSS	Die CSS-Dateien müssen vom Frontend komplett getrennt sein, damit durch den Austausch dieser ein neues Design der App erzielt werden kann.	soll
NF_08	Mehrsprachigkeit	Das System muss in mehreren Sprachen zur Verfügung stehen, mindestens Deutsch und Englisch.	muss
NF_09	Hilfefunktion	Das System muss über eine Hilfefunktion verfügen, die die grundlegenden Funktionalitäten erklärt.	muss

Tabelle 5: Nicht-funktionale Anforderungen

## 6 Konzept

Nachdem im vorherigen Kapitel nach einer Beschreibung der Stakeholder und der Use-Cases die Anforderungen ermittelt und priorisiert wurden, soll nun das eigentliche Konzept entwickelt werden. Zunächst soll dafür die Gesamtarchitektur des Systems beschrieben werden. Anschließend werden die einzelnen in der Anforderungsanalyse ermittelten notwendigen Funktionalitäten sowie Möglichkeiten für die technische Umsetzung derer genau beschrieben. Im Rahmen dieser Arbeit werden lediglich ausgewählte Funktionen implementiert (siehe Kapitel 7).

### 6.1 Technische Gesamtarchitektur

Generell ist bei der Entwicklung von Apps die Entscheidung zu treffen, ob eine native oder eine Web-App entwickelt werden soll. Wie in Kapitel 3.4 dargestellt, sind für Bibliotheken aufgrund der Entwicklungskosten und der schnellen und unkomplizierten Möglichkeit der Veröffentlichung der App vor allem Web-Apps geeignet. Dabei sind hauptsächlich HTML-, CSS- und JavaScript-Kenntnisse notwendig, um Anwendungen sowohl für Android-, als auch iOS-Geräte zu entwickeln. Zur Nutzung der App auf dem Endgerät ist lediglich ein Webbrowser notwendig. Eine separate Installation und umfangreiche Update-Abläufe entfallen.

Wie bereits im Prototyp umgesetzt, soll das System aus einem Front- und einem Backend bestehen. Während das Frontend für die Interaktion mit dem Benutzer verantwortlich ist, ist das Backend für die Ermittlung und Bereitstellung wichtiger Informationen zuständig. Beide Komponenten befinden sich dabei in ständiger Kommunikation miteinander (siehe Abb. 14).

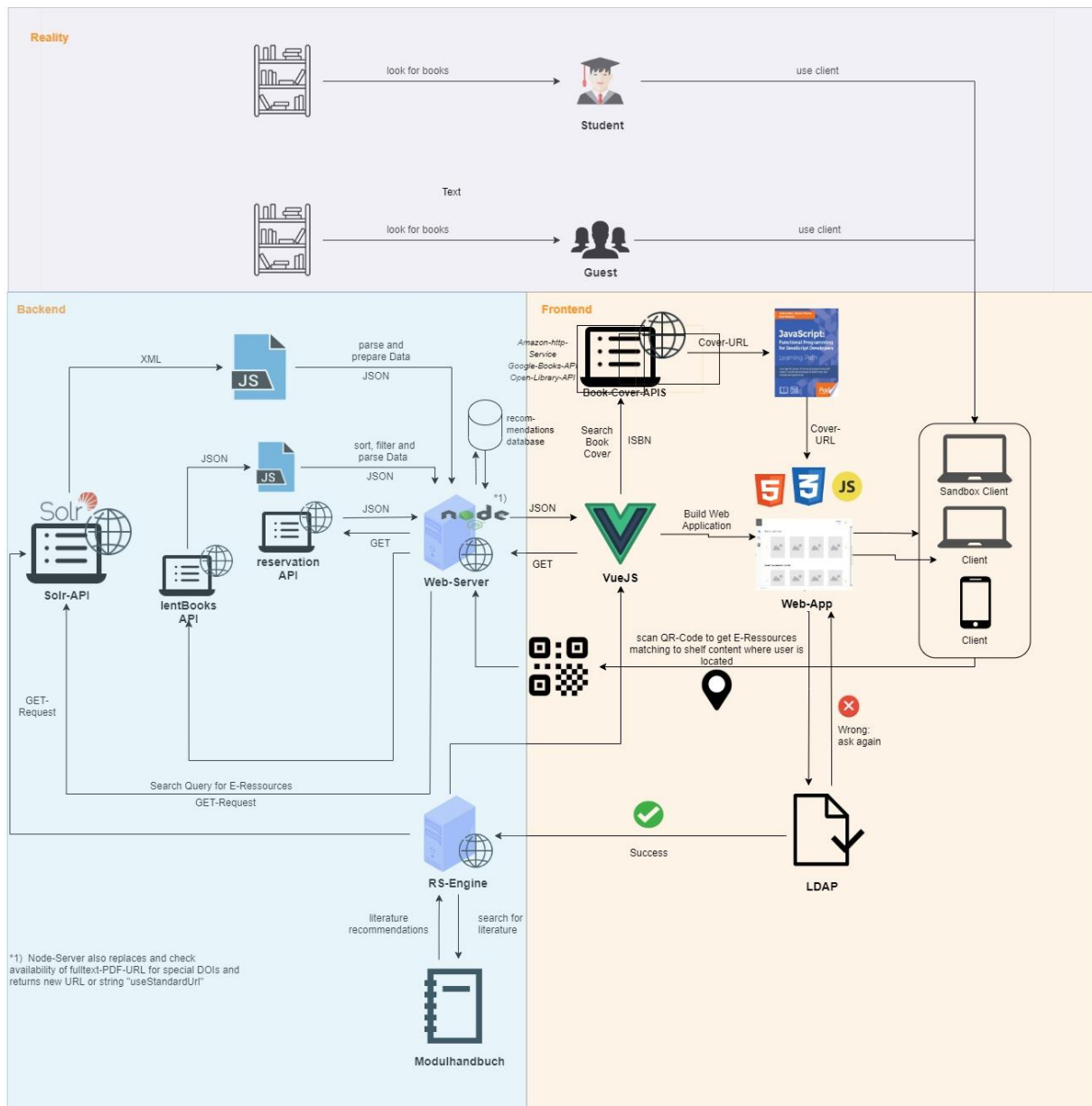


Abbildung 14: Gesamtarchitektur des Systems  
Eigene Abbildung

### 6.1.1 Frontend

Für die Umsetzung von Frontends von Web-Apps gibt es unterschiedlichste Herangehensweisen. Ist es möglich, die App von Grund auf selbst mit HTML, CSS und JavaScript zu gestalten, stehen mittlerweile auch zahlreiche Frameworks zur Verfügung, die viele Grundfunktionalitäten bereits umgesetzt haben. Wie in Kapitel 5.3 aufgezeigt, empfiehlt sich aufgrund der hohen Lernkurve und der guten Dokumentation bspw. Vue.js besonders für die Umsetzung. Der schlanke Kern des Frameworks lässt sich dabei einfach um verschiedene Pakete erweitern. Auch der bereits bestehende Prototyp basiert auf Vue.js.

Wichtig bei der Erstellung des Frontends ist das Ziel, eine Oberfläche mit hoher Usability zu erstellen. Zu diesem Zweck soll das Frontend mit Komponenten des Material Designs erstellt werden. Dieses versucht, die Nutzererfahrungen von Printmedien aufzugreifen und diese bei der Erstellung digitaler Oberflächen

einzubringen.<sup>168</sup> Für das Framework Vue.js gibt es ein eigenes Paket für die Umsetzung von Komponenten des Material Designs: Vuetify



Abbildung 15: Navigation im Prototyp der App E-Books to go

Darüber hinaus spielt zu diesem Zweck eine „sprechende“ Benennung der Menüpunkte eine besondere Rolle. Die im bestehenden Prototyp verwendeten Bezeichnungen „E-Books finden/Find E-Books“ (für die Suche nach Medien passend zum Regalstandort durch Scannen eines QR-Codes) bzw. „E-Books suchen/Search E-Books“ (für eine „eigene“ Suche unabhängig vom Regalstandort) (siehe Abb. 15) erschweren eine eindeutige Identifikation der dahinterstehenden Funktionen.

Wichtig bei der Umsetzung des Frontends ist außerdem Anforderung NF\_02: die App soll responsiv gestaltet sein und auf verschiedenen Endgeräten wie Laptop, Tablet und Smartphone nutzbar sein. Vue.js erfüllt all diese Anforderungen.

Neben der Darstellung der Nutzeroberfläche der App ist das Frontend auch verantwortlich für die Kommunikation mit dem zentralen Verzeichnisdienst der TH Wildau, welche während der Anmeldung eines Nutzers stattfindet. Diese wird nicht ins Backend ausgelagert, um nicht unnötig Nutzerdaten zwischen Front- und Backend hin und her zu schicken. Die für weitere Funktionen der App notwendigen Daten werden im Bereich des Frontends gespeichert.

## 6.1.2 Backend

Das Backend besteht im Wesentlichen aus einem Server und einer bzw. zwei Datenbank/en.

In einer Datenbank werden die Informationen zu Buchempfehlungen der Nutzer hinterlegt. In der zweiten Datenbank können, je nach Umsetzung, ggf. die in einem Regalbereich untergebrachten RVK-Notationen gespeichert werden.

Der Server des Prototyps wurde auf Grundlage des Node.js-Frameworks erstellt. Dieses ermöglicht die Erstellung des Servers mit der Sprache JavaScript, welche auch für die Erstellung des Frontends genutzt wird, sodass im Rahmen der verwendeten Programmiersprachen Einheitlichkeit erreicht werden kann. JavaScript, welches ursprünglich clientseitig genutzt wird, kann mit Hilfe des Frameworks Node.js auch serverseitig genutzt werden. Auf der Grundlage von Node.js entstanden wiederum Frameworks, welche unter anderem eine vergleichsweise einfache Erstellung verschiedener Endpoints ermöglicht, welche später vom Frontend aus angefragt werden können. Endpunkte müssen zum Beispiel für folgende Funktionen bereitgestellt werden:

- Ermittlung der neuesten E-Books (siehe Anforderung F\_10)
- Ermittlung von zum Nutzerstandort passenden E-Ressourcen (siehe Anforderung F\_04)
- Ermittlung von Medien nach Absetzen der „einfachen“ Suche (siehe Anforderung F\_33)
- Ermittlung von entliehenen Bänden (siehe Anforderung F\_06)

---

<sup>168</sup> Vgl. Google (n. N.).

- Ermittlung von Covern zu den ermittelten Medien (für Umsetzung der Anforderung F\_16)
- Senden einer E-Mail mit einer Empfehlung (siehe Anforderung F\_26)
- Ermittlung von Empfehlungen anderer Nutzer (F\_30 – F\_31)

## 6.2 Zu integrierende Medientypen

Der Bestand der Bibliothek der Hochschule der TH Wildau setzt sich aus verschiedensten Medientypen zusammen. Während der Prototyp der App sich auf die Präsentation von E-Books fokussiert, sollen im Rahmen der Weiterentwicklung der App ggf. weitere Medientypen integriert werden. Im Bereich der E-Ressourcen spielen neben E-Books vor allem im naturwissenschaftlichen Bereich E-Journals bzw. einzelne Artikel eine große Rolle.

Für die Leser standen im Jahr 2020 in den Regalen der Bibliothek der TH Wildau ca 82.000 Monographien und ca 100 laufend bestellte Zeitschriftentitel zum Stöbern bereit. Hinzu kommt jedoch der elektronische Bestand. Dieser setzt sich zusammen aus ca 67.000 Einheiten *Digitaler Bestände insgesamt (ohne elektronische Zeitschriften und Zeitungen)* und ca 12.700 Einheiten *lizensierter und digitalisierter elektronischer Zeitschriften und Zeitungen*.<sup>169</sup>

Im Bereich der elektronischen Medien (hier gemeint sind E-Books und E-Journals) machen die E-Journals einen Anteil von nahezu einem Fünftel aus, sodass diese langfristig in die App integriert werden sollten. Dafür spricht auch die Tatsache, dass Forschungsergebnisse gerade im technisch-naturwissenschaftlichen Bereich zunächst in Artikeln in Fachzeitschriften publiziert werden, bevor sie in eigenständigen Publikationen veröffentlicht werden.

In Wilbert finden sich zusätzlich zu E-Books und E-Journals über sieben Millionen Aufsätze.<sup>170</sup> Aufgrund der Vielzahl der recherchierbaren Artikel bietet sich hier eine differenziertere Suche über Wilbert an. Da das Ziel der App nicht die Nachbildung der Suchfunktionalitäten in Wilbert sein soll, soll an dieser Stelle auf die Integration von Artikeln verzichtet werden. Ggf. kann in der App ein Hinweis untergebracht werden, dass Artikel nicht in die Suche integriert sind. Stattdessen kann auf Wilbert verwiesen werden.

Für den Nutzer, der am Regal stöbert, sind jedoch neben den elektronischen Medien auch entliehene Printmedien nicht direkt sichtbar. Um den Nutzern wie bei der Katalogrecherche auch beim Stöbern am Regal einen umfassenden Bestandsüberblick zu ermöglichen, sollten diese ebenfalls in die App integriert werden.

---

<sup>169</sup> Vgl. hbz (n. N.).

<sup>170</sup> Siehe

[https://wilbert.kobv.de/simpleSearch.do?sortCrit=year&sortOrder=desc&hitsPerPage=100&query=%3A\\*&plv=2](https://wilbert.kobv.de/simpleSearch.do?sortCrit=year&sortOrder=desc&hitsPerPage=100&query=%3A*&plv=2), letzte Abfrage 16.07.2021

## 6.3 Standorterkennung

Wie sich in der Vorbetrachtung herausgestellt hat, eignen sich direkte oder indirekte QR-Codes als Methode des mobile Taggings am besten für die angestrebte Zielstellung. Dafür sprechen vor allem die mit wenig Aufwand verbundene und preisgünstige Erstellung dieser, verbunden mit der einfachen Anwendbarkeit auf Nutzerseite.

Gerade aufgrund dieser Argumente soll die Standorterkennung in der App zunächst auf Grundlage des Scannens eines QR-Codes erfolgen, der zuvor am Regal angebracht wurde. Später kann dies ggf. durch andere Verfahren ersetzt werden.

### 6.3.1 Integration von QR-Codes in die App

Wichtig ist, dass dem Nutzer einfach verständlich ist, wie er in der Anwendung zu den gewünschten Ergebnissen kommt. So kann bspw., wie im Prototyp bereits umgesetzt, die Aktivierung der Kamera über einen Button erfolgen.

Ein Platzhalter (bspw. mit einem QR-Code-Template) kann helfen, dem Nutzer vor Aktivierung der Kamera anzuzeigen, an welchem Bereich des Screens der Kamerabereich sichtbar sein wird (siehe Abb. 15).



Abbildung 16: Mockup Menü mit QR-Template

Darüber hinaus ist es wichtig, dem Nutzer zu signalisieren, wann das Scannen des QR-Codes erfolgreich war. Dafür eignen sich optische Tools. Auf ein akustisches Signal sollte verzichtet werden, um andere Nutzer nicht zu stören.

Ggf. kann ein Hinweis erfolgen, dass nach Aktivierung der Kamera das mobile Endgerät lediglich so positioniert werden muss, dass sich der QR-Code gut im Sichtfeld der Kamera befindet und keine weitere Aktion (wie das Bestätigen eines Buttons o.Ä.) notwendig sind.

Je nachdem, wie die QR-Codes an den Bibliotheksregalen angebracht werden (statisch oder dynamisch, siehe Kapitel 6.3.3) werden direkte oder indirekte QR-Codes genutzt. Beide Möglichkeiten sollen an dieser Stelle skizziert werden (siehe Abb. 17).



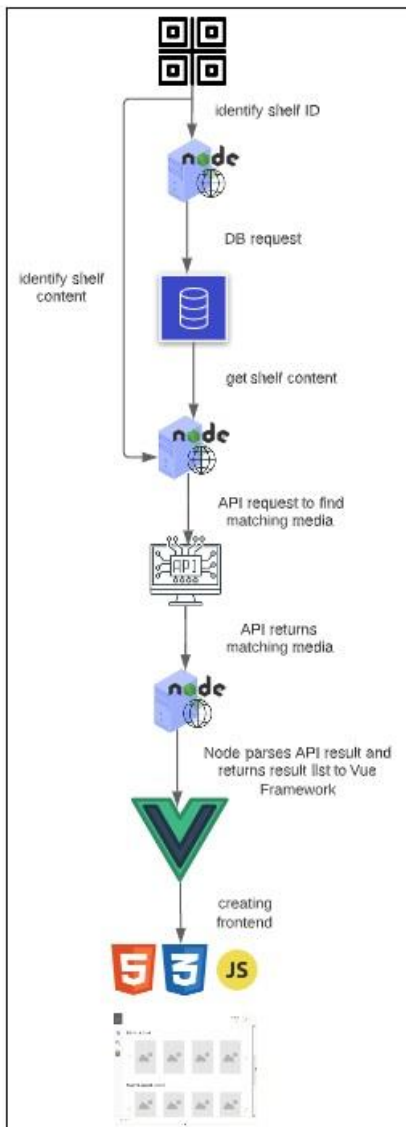


Abbildung 17: Ablauf Ermittlung Regalstandort  
Eigene Abbildung

Werden die QR-Codes statisch am Regal untergebracht, bezeichnen diese einen vorher definierten Regalbereich. Der Inhalt dieses Bereichs kann sich jedoch stetig durch Umstellungen oder neu hinzugekommene Bände ändern. Deshalb wird in dem Fall auf dem QR-Code lediglich eine ID, bspw. eine Regal-ID, gespeichert. Hier wird also mit indirekten QR-Codes gearbeitet. Diese muss im Nachhinein aufgelöst werden und durch die entsprechenden Informationen zur Ermittlung des zum Regalbereich passenden Themenbereichs ersetzt werden. Das Matching der Informationen kann bspw. mit einer Datenbank erfolgen, in der die Regal-IDs und die zugehörigen Informationen zu den im Regalbereich stehenden Medien erfasst sind.

Im Falle von dynamisch am Regal angebrachten direkten QR-Codes wandern diese quasi mit dem Bestandsbereich, den sie bezeichnen. In dem Fall kann mit direkten QR-Codes gearbeitet werden, da sich die zu ermittelnden Informationen zum Themenbereich des Bestandsbereichs hier nicht ändern. Ein separates Matching der Informationen bspw. mit Hilfe einer Datenbank ist hier nicht notwendig.

Die mit Hilfe des Barcodes erworbenen Informationen werden dann in beiden Fällen in der App weiterverwendet, um die passenden Medien anhand einer API oder ähnlicher Tools abzufragen (siehe Kapitel 6.4 – 6.6).

### 6.3.2 Erstellung von QR-Codes

Für die Erstellung von QR-Codes gibt es verschiedene kostenfreie oder kostenpflichtige Tools (je nach Anforderung) im Internet, z.B. QR Code Generator<sup>171</sup> oder qrcodmonkey<sup>172</sup>. Andernfalls ist es auch möglich, bspw. auf Basis von JavaScript, einen eigenen Generator zu erstellen oder Tools zur Generierung von Etiketten zu nutzen.

Da unabhängig von der Verwendung direkter bzw. indirekter QR-Codes eine Vielzahl dergleichen benötigt wird, ist es wichtig, ein Tool zu finden, welches mehrere QR-Codes bspw. anhand einer vorgegebenen Liste von Inhalten generieren kann.

Darüber hinaus sollten zusätzlich zu den eigentlichen QR-Codes auch Beschriftungen dieser möglich sein, sodass zusätzlich bspw. die hinterlegte/n RVK-Notation/en ablesbar sind.

### 6.3.3 Unterbringung der QR-Codes am Regal

Wie bereits in Kapitel 6.3.1 erwähnt, können die QR-Codes statisch oder dynamisch an den Bibliotheksregalen angebracht werden.

Werden die Codes statisch (unveränderlich) angebracht, werden indirekte QR-Codes genutzt, die einen Identifier für den zu bezeichnenden Regalbereich enthält. Bspw. mit Hilfe einer Datenbank werden dem Identifier dann die Informationen zum thematischen Inhalt des Regals zugeordnet. Durch in Bibliotheken vorkommende Rückungen (in der Bibliothek der TH Wildau z.B. vor allem durch das Umsignierungsprojekt von GHB-Systematik auf RVK) und ständig neu hinzukommende Bände, sind die Informationen, welche Bände und somit welche thematischen Bereiche in einem Regal untergebracht sind, nicht statisch, sondern befinden sich stets im Wandel. Dies hat zur Folge, dass die Datenbank stets auf dem aktuellen Stand gehalten werden muss und ein zusätzlicher nicht zu vernachlässigender Verwaltungsaufwand für das Bibliothekspersonal entsteht, um die Daten für die Anwendung aktuell zu halten.

Darüber hinaus müssen die Regalflächen in regelmäßige (unabhängig von der Aufstellung) Abschnitte eingeteilt werden. Auf die thematischen Inhalte der Regale kann an dieser Stelle keine Rücksicht genommen werden, da diese sich stetig ändern. Abschnitte können z.B. Regalreihen, einzelne Regale oder Regalböden sein. Werden ganze Regalreihen mit einem QR-Code abgebildet, ist dies inhaltlich viel zu ungenau. Auch bei der Abgrenzung eines Regals kann die Einordnung zum Teil noch viel zu grob sein. Bildet man jedoch einzelne Regalböden ab, müssen Nutzer sich bücken, wollen sie bspw. den QR-Code des untersten Regalbodens abfotografieren. Der Nutzerkomfort geht an dieser Stelle verloren. Für Themengebiete die sich über mehrere Regalböden erstrecken (Bücher zum Fachgebiet Java-Programmierung erstrecken sich bspw. über 15 Regalböden/4 Regale) würden bei einer Verteilung von statischen QR-Codes auf Regalböden 15 Stück benötigt, die auf das gleiche Fachgebiet verweisen würden.

Würde man für diesen Fachbereich dynamische QR-Codes verwenden, würde die Anbringung eines Codes (idealerweise beim Beginn des Fachgebiets) ausreichen. Beim Weiterrücken des Bestands würde der Code einfach mit weitergesetzt werden. Eine zusätzliche regelmäßige Datenpflege ist nicht notwendig. In der

---

<sup>171</sup> Siehe <https://www.qrcode-generator.de/>

<sup>172</sup> Siehe <https://www.qrcode-monkey.com/de/>

Bibliothek der TH Wildau könnte man bspw. die ohnehin an den Regalböden angebrachten Hinweise zum Beginn einer neuen Systematikstelle um einen QR-Code ergänzt werden (siehe Abb. 18). Alternativ könnten sogenannte „Schildwackler“ (siehe Abb. 19) oder, wenn genug Platz im Regal vorhanden ist, Beschriftungswürfel (siehe Abb. 20) genutzt werden.

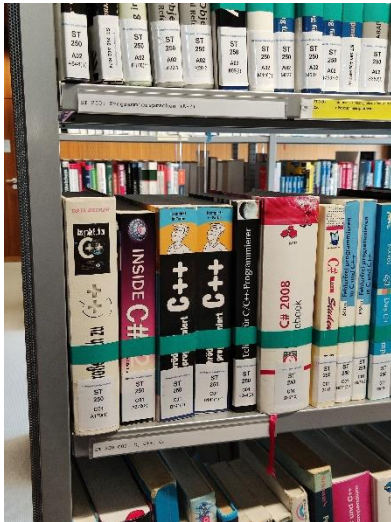


Abbildung 18: *Abbildung 21: Beschriftung der Regale mit Systematikstellen in Wildau*



Abbildung 19: *Schildwackler Flexi*  
Quelle: ekz (2020), S. 14



Abbildung 20: *Beschriftungswürfel*  
Quelle: ekz (2020), S. 214

Nicht zuletzt darf nicht vergessen werden, dass zu viele QR-Codes die Gefahr innehaben, in einem Dschungel aus QR-Codes unterzugehen und gar nicht mehr durch die Nutzer wahrgenommen werden. Zielstellung sollte aber sein, dass die QR-Codes den Lesern ins Auge fallen und zur Nutzung der Web-App anregen.

Nach Rücksprache mit den Kollegen der Hochschulbibliothek wurde darüber hinaus nochmals die Wichtigkeit betont, dass bei der Verwaltung der Daten der QR-Codes unbedingt darauf zu achten ist, dass kein zusätzlicher Arbeitsaufwand entstehen sollte.

## 6.4 Schnittstelle für die Ermittlung der Ressourcen

Zugriff auf die gedruckten und elektronisch lizenzierten Inhalte der Bibliothek der TH Wildau kann man über das Discovery-System Wilbert erhalten.

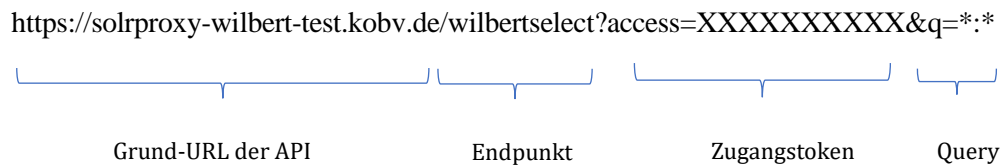
Im bestehenden Prototyp wurden Anfragen direkt an dieses gestellt. Mittels eines entsprechenden Parsers wurden die notwendigen Informationen aus der HTML-Response entnommen und diese für die App weiterverarbeitet. Da die Inhalte der HTML-Response jedoch bereits vorverarbeitet für die Anzeige und Nutzung im Discovery-System sind, soll stattdessen eine API mit Zugriff auf die nicht vorverarbeiteten Daten genutzt werden. Für Informationen zum Bestand der Bibliothek der TH Wildau wird vom Kooperativen Bibliotheksverband Berlin-Brandenburg (KOBV) auf einem Solr-Proxy-Server eine REST-API

angeboten, an welche Anfragen gestellt werden können. Da es sich nicht um eine offene Schnittstelle handelt, benötigt der Nutzer einen Key, welcher bei allen Anfragen mitzusenden ist.<sup>173</sup>

Ähnlich wie bei Anfragen an Wilbert können über die Schnittstelle anhand verschiedener Parameter die Ressourcen der Bibliothek der TH Wildau abgefragt werden. Die Antwort wird im XML-Format ausgeliefert. Diese kann dann mittels Parser auf die benötigten Informationen durchsucht werden.

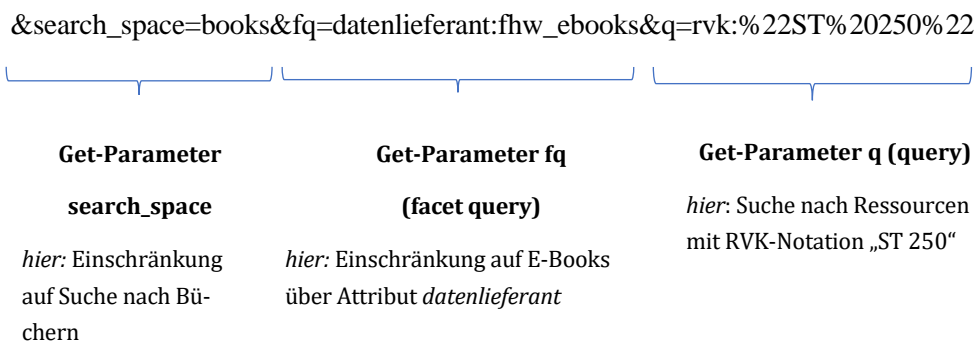
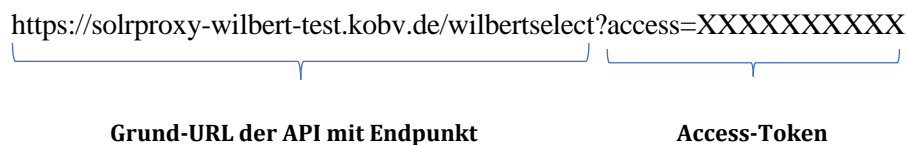
Standardmäßig werden die ersten zehn Treffer zurückgeliefert. Sollten mehr Treffer zurückgegeben werden, so ist dies mit Hilfe des Parameters „rows“ anzugeben.

Der Grundaufbau einer Anfrage sieht folgendermaßen aus:



Je nach Anfrage können zusätzlich über den GET-Parameter *fq* weitere Spezifizierungen der Suchanfrage, bspw. zu *search\_space* und *datenlieferant*, vorgenommen werden.

Eine Anfrage nach E-Books zur RVK-Notation *ST 250* sieht folgendermaßen aus:



Die Schnittstelle liefert eine Antwort im XML-Format (siehe Listing 1).

<sup>173</sup> Bei den beschriebenen Anfragen wird der Key aus Sicherheitsgründen anonymisiert.

```
<response>
[...]
```

<result name="response" numFound="173" start="0" maxScore="9.513494">

<str name="uid">fhw\_ebooks\_249713</str>

<str name="search\_space">books</str>

<str name="datenlieferant">fhw\_ebooks</str>

<arr name="seiten">

<str>1 Online-Ressource (XI, 156 S. 216 Abb)</str>

</arr>

<arr name="book\_url\_plain">

<str>https://doi.org/10.1007/978-3-662-61309-2 </str>

</arr>

<arr name="autor">

<str>Silberbauer, Christian</str>

</arr>

<arr name="isbn">

<str>9783662613092 </str>

</arr>

<arr name="edition">

<str>2., aktualisierte und erweiterte Auflage</str>

</arr>

<arr name="identnr">

<str>249713</str>

</arr>

<arr name="schlagwort">

<str>Java (Programmiersprache)</str>

<str>Objektorientierung</str>

<str>Objektorientierte Programmierung</str>

<str>Java (Computer program language)</str>

<str>Programming languages (Electronic computers)</str>

<str>Software engineering</str>

<str>Java</str>

<str>Programming Languages, Compilers, Interpreters</str>

<str>Software Engineering</str>

<str>Java (Programmiersprache)</str>

<str>Objektorientierung</str>

<str>Objektorientierte Programmierung</str>

</arr>

<str>Berlin, Heidelberg : Springer Berlin Heidelberg</str>

[...]

<arr name="subject">

<str>inf</str>

</arr>

<arr name="language">

<str>ger</str>

</arr>

<arr name="titel">

<str>Einstieg in Java und OOP : Grundelemente, Objektorientierung, De-sign-Patterns und Aspektorientierung</str>

</arr>

<arr name="shingle\_title\_1">

<str>Einstieg in Java und OOP : Grundelemente, Objektorientierung, De-sign-Patterns und Aspektorientierung</str>

</arr>

[...]

<str name="materialart">online</str>

<arr name="book\_url">

<str>https://doi.org/10.1007/978-3-662-61309-2@A@Volltext </str>

</arr>

<arr name="rvk">

<str>ST 231</str>

<str>ST 250</str>

```

</arr>
<arr name="titel_suche">
<str>Einstieg in Java und OOP Grundelemente, Objektorientierung, Design-Patterns
und Aspektorientierung</str>
</arr>
<arr name="book_doi">
<str>10.1007/978-3-662-61309-2 </str>
</arr>
<arr name="publikationsjahr_intervall">
<str>7979:2020-2022</str>
</arr>
<str name="insertion_date">2020-10-07</str>
<str name="publikationsjahr_sort">2020</str>
<arr name="autor_bkz">
<str>Silberbauer, Christian</str>
</arr>
<arr name="fulltext_url">
<str>https://doi.org/10.1007/978-3-662-61309-2 </str>
</arr>
<arr name="person">
<str>Silberbauer, Christian</str>
</arr>
<arr name="publikationsjahr_anzeige">
<str>2020</str>
</arr>
<arr name="shingle_author_1">
<str>Silberbauer, Christian</str>
</arr>
<arr name="publisher">
<str>Berlin, Heidelberg : Springer Berlin Heidelberg</str>
<str>Berlin, Heidelberg : Springer Vieweg</str>
</arr>
<arr name="publikationsjahr_facette">
<str>2020</str>
</arr>
<arr name="ddc">
<str>005.133</str>
</arr>
[...]
<float name="score">9.5305</float>
</doc>
</response>

```

*Listing 1: Antwort der Solr-API (Ausschnitt)*

## 6.5 Ermittlung von zum Nutzerstandort passenden E-Ressourcen

Wichtigstes Ziel der Anwendung in Bezug auf die Zahl der anzuzeigenden Treffer sollte es sein, den Nutzer nicht mit zu vielen Treffern zu überfordern. Andererseits sollte versucht werden, wirklich alle zum Themenbereich passenden E-Ressourcen zu ermitteln. Dies stellt sich jedoch aufgrund der z.T. recht unterschiedlich vergebenen Sacherschließungsattribute als nicht ganz einfach heraus.

Zur Ermittlung der thematisch passenden E-Ressourcen spielen vor allem die Erschließung von Print- und E-Ressourcen sowie die Art der Aufstellung eine Rolle.

Wie im Kapitel 3.2.2 dargestellt, ist Voraussetzung für die Anwendung einer App zur Überwindung des Medienbruchs am Regal eine systematische Aufstellung. In der Bibliothek der TH Wildau stehen die Monographien im Lesesaal systematisch nach RVK-Notationen für die Nutzer zugänglich aufgestellt. Somit ist diese Voraussetzung erfüllt.

Die inhaltliche Erschließung von Printmedien erfolgt mit Hilfe der Vergabe von Notationen und Schlagwörtern. Bei der Erschließung von E-Ressourcen liegen diese Informationen jedoch nur zum Teil vor, sodass lediglich 30% der E-Books mit RVK-Notationen erschlossen sind.<sup>174</sup> Ursache dafür ist, dass diese häufig in großen Paketen gekauft werden. Dadurch werden E-Ressourcen, anders als gedruckte Bände, nicht mehr einzeln manuell per Autopsie erschlossen, sondern die bibliographischen Daten automatisch in die Kataloge übernommen werden.

### 6.5.1 DDC versus RVK bei E-Book-Central

Vor allem E-Books von E-Book-Central (ca 11.500 E-Books)<sup>175</sup> sind anstelle einer RVK-Notation mit einer DDC-Notation versehen. Da die Aufstellung der Bände jedoch anhand der RVK erfolgt, müsste hier zunächst ein Mapping der DDC-Notation auf die jeweilige RVK-Notation erfolgen. Hierfür kann bspw. das Cocoda-Mapping-Tool genutzt werden,<sup>176</sup> für welches eine offene API angeboten wird. Beim Testen stellte sich jedoch heraus, dass es nur sehr wenige eingetragene RVK-DDC-Mappings gibt.

---

<sup>174</sup> Siehe Anlage A.1, Protokoll zum Kickoff-Meeting

<sup>175</sup> Vgl.:

[https://wilbert.kobv.de/simpleSearch.do?fq=search\\_space%3abooks&sortCrit=year&sortOrder=desc&hitsPerPage=100&index=internal&query=%3a\\*&plv=2](https://wilbert.kobv.de/simpleSearch.do?fq=search_space%3abooks&sortCrit=year&sortOrder=desc&hitsPerPage=100&index=internal&query=%3a*&plv=2), 24.07.2021

<sup>176</sup> Vgl.: <https://coli-conc.gbv.de/cocoda/rvk/>, 24.07.2021

From	To	Creator	Date	Download	Mappings
RVK	DDC	GESIS		JSKOS CSV	68
RVK	DDC	GESIS		JSKOS CSV	771
RVK	DDC	GESIS		JSKOS CSV	1.335
RVK	DDC	Manuela Queitsch		JSKOS CSV	407
RVK	DDC	GESIS		JSKOS CSV	1.162
RVK	DDC	GESIS		JSKOS CSV	5.347
RVK	DDC	GESIS		JSKOS CSV	227
RVK	DDC	VZG	2015	JSKOS CSV	23

9 Concordances Total: 9.697

Abbildung 21: Abfrage RVK-DDC-Mappings vom Cocoda Mapping Tool  
 Quelle: Verbundzentrale des Gemeinsamen Bibliotheksverbundes (GBV) (n. N.)

Wie auf Abbildung 21 zu erkennen ist, stammt ein Großteil der Einträge vom Leibniz-Institut für Sozialwissenschaften (GESIS). Zum Bereich Informatik (RVK-Notationsbereich SQ – SU) finden sich dementsprechend nur sehr wenige Mappings. Anders sieht es aus, wenn man nach Mappings von DDC auf RVK sucht. Hier stammt ein Großteil der Einträge von der VZG (über 8.700 von insgesamt 9.200 Einträgen).<sup>177</sup>

So kann über die API quasi auf umgekehrten Wege versucht werden, zu einer bestimmten RVK-Notation passende DDC-Notationen zu suchen. Anstatt von auf der RVK-Notation Mappings auf die DDC-Notation zu recherchieren, wird nach DDC-Notationen gesucht, die auf die entsprechende RVK-Notation mappen. Eine entsprechende Suchanfrage sähe folgendermaßen aus:

<https://coli-conc.gbv.de/api/mappings?fromScheme=DDC>

URL der API

Endpoint

Get-Parameter *fromScheme*  
 - enthält Namen der Systematik, von der aus das Mapping durchgeführt werden soll

[https://coli-conc.gbv.de/api/mappings?fromScheme=DDC&to=http%3A%2F%2Frvk.uni-regensburg.de%2Fnt%2FST%252027\\*](https://coli-conc.gbv.de/api/mappings?fromScheme=DDC&to=http%3A%2F%2Frvk.uni-regensburg.de%2Fnt%2FST%252027*)

Get-Parameter *to*

- enthält Systematikname und Notation, zu der das Mapping durchgeführt werden soll als URI (hier trunkierte Suche nach RVK-Notation/en ST 27\*)

<sup>177</sup> Vgl.: <https://coli-conc.gbv.de/cocoda/app/?fromScheme=http%3A%2F%2Furi.gbv.de%2Fterminology%2Frvk%2F&toScheme=http%3A%2F%2Fdewey.info%2Fscheme%2Fedition%2Fe23%2F>, 25.07.2021



Als Antwort erhält man ein JSON-Objekt, welches in Listing 2 ausschnittsweise dargestellt ist. Dieses enthält neben weiteren Informationen die ermittelte DDC-Notation.

```
[...]
{
  "toScheme": {
    "notation": [
      "RVK"
    ],
    "uri": "http://bartoc.org/en/node/533"
  },
  "from": {
    "memberSet": [
      {
        "notation": [
          "346.08210285"
        ],
        "uri": "http://dewey.info/class/346.08210285/e23/"
      }
    ]
  },
  "to": {
    "memberSet": [
      {
        "uri": "http://rvk.uni-regensburg.de/nt/ST%20278",
        "notation": [
          "ST 278"
        ]
      }
    ]
  },
  "fromScheme": {
    "notation": [
      "DDC"
    ],
    "uri": "http://bartoc.org/en/node/241"
  },
  "partOf": [
    {
      "uri": "http://coli-conc.gbv.de/concordances/ddc_rvk_recht"
    }
  ],
  "creator": [
    {
      "prefLabel": {
        "de": "VZG"
      }
    }
  ],
  [...]
}
```

Listing 2: Antwort der Cocoda-API (Ausschnitt)

Hält man sich jedoch vor Augen, dass die RVK insgesamt ca 860.000 Einträge (Stand: 2018)<sup>178</sup> umfasst, wird klar, dass das Cocoda-Tool hier nur sehr begrenzt Abhilfe schaffen kann. (Die DDC umfasst ca 48.000 Einträge, Stand: 2005).<sup>179</sup>

Ein weiteres Tool zur Ermittlung von DDC-Notationen ist das Tool Classify von OCLC. Hier könnten u.a. mit Hilfe von ISBNs DDC-Notationen ermittelt werden.<sup>180</sup> Problematisch wäre hier allerdings, dass sich diese auf einen spezifischen Band beziehen würden und somit unklar ist, ob der thematische RVK-Kontext, in dem der Band aufgestellt ist, noch korrekt abgebildet wird.

## 6.5.2 Treffermengen bei Nutzung der RVK-Notation

Nach Betrachtung der Erschließungsattribute zeigt sich, dass (wie im bestehenden Prototyp umgesetzt) für die Ermittlung der zum Regalstandort passenden E-Books die RVK-Notation geeignet sein könnte. Ggf. kann ein Mapping auf zugehörige DDC-Notationen erfolgen, sodass auch die E-Books des Anbieters E-Book-Central gefunden werden. Wie sich gezeigt hat, mangelt es hier derzeit allerdings noch an Mapping-Tools mit hinreichend großem Datenbestand.

Schaut man sich nun den Regalbereich zum Fachgebiet Informatik genauer an, fällt besonders an der RVK-Systemstelle *ST 250* auf, dass sich diese über sieben Regale<sup>181</sup> (948 Exemplare, Stand: 13.04.2021)<sup>182</sup> erstreckt. Eine Kataloganfrage nach E-Books zur RVK-Notation *ST 250* liefert 173 Treffer.<sup>183</sup>

Bei genauerer Betrachtung der Signaturen in diesem Bereich fällt auf, dass diese hier neben der RVK-Notation und Informationen zu Verfasser und Exemplar zusätzlich spezifischere Informationen zum Thema enthalten. Die Signatur *ST 250 X70 E77* setzt sich bspw. folgendermaßen zusammen:

ST 250	RVK-Notation für Monographien zu einzelnen Programmiersprachen (A-Z)
X70	Kürzel für den Unterbereich XML
E77	Verfasserkürzel (hier: Espasito, Dino)

Der RVK-Bereich *ST 250* gliedert sich insgesamt noch einmal in 35 Untergruppen, zu welcher Bestand im Umfang von einem Band (z.B. X 84, S 46 oder P 89) bis hin zu über 380 Bänden (J 35, Programmiersprache Java) gehören.

---

<sup>178</sup> Vgl. Universitätsbibliothek Regensburg (2019): Über 50 Jahre RVK : Näheres zum Werdegang, <https://rvk.uni-regensburg.de/44-ueber-die-rvk-regensburger-verbundklassifikation/geschichte-rvk-regensburger-verbundklassifikation-historie-entwicklung/57-werdegang-der-rvk>, 25.07.2021.

<sup>179</sup> Vgl. Reiner (2007): Automatic Analysis of Dewey Decimal Classification Notations, <https://monarch.qucosa.de/api/qucosa%3A18781/attachment/ATT-0/>, 25.07.2021.

<sup>180</sup> Vgl.: <http://classify.oclc.org/classify2/>

<sup>181</sup> Siehe Anhang A5

<sup>182</sup> Siehe [rvk\\_notationen\\_lesesaal\\_2021\\_04\\_13\\_rvk\\_bereich\\_ST\\_250.txt](#)

<sup>183</sup> Vgl.:

[https://wilbert.kobv.de/advancedSearch.do?fq=datenlieferant%3afhw\\_ebooks&sortCrit=score&sortOrder=desc&hitsPerPage=10&fq=search\\_space%3abooks&index=internal&f1=rvk&v1=ST+250&conj1=and&f2=author&v2=&conj2=and&f3=title&v3=&conj3=and&f4=title&v4=&conj4=and&f5=title&v5=&conj5=and&f6=title&v6=&conj6=and&f7=title&v7=&plv=2](https://wilbert.kobv.de/advancedSearch.do?fq=datenlieferant%3afhw_ebooks&sortCrit=score&sortOrder=desc&hitsPerPage=10&fq=search_space%3abooks&index=internal&f1=rvk&v1=ST+250&conj1=and&f2=author&v2=&conj2=and&f3=title&v3=&conj3=and&f4=title&v4=&conj4=and&f5=title&v5=&conj5=and&f6=title&v6=&conj6=and&f7=title&v7=&plv=2), 24.07.2021

Gerade diese untere Ebene, die zum Teil in der Signatur von Printbänden zu finden ist, ist allerdings bei RVK-Notationen der E-Books nicht mehr abgebildet. Die Erschließung von E-Books erfolgt nur im Rahmen der RVK-Notationen, spezifische Informationen (wie sie hier bei den Printbänden zu finden sind) sind darüber hinaus nicht mehr erhalten. Hier würde es sich anbieten, die bei der Aufstellung an die RVK-Notation gehangene Zusatzinformation als Schlagwort an die Suche anzuhängen, sodass im Falle einer Suche nach einem E-Book zur Programmiersprache Java die RVK-Notation *ST 250* genutzt wird und die Suchergebnisse durch die Nutzung des Schlagwortes „Java“ entsprechend weiter eingeschränkt werden.<sup>184</sup>

Neben dem Problem einer zu großen Trefferzahl kann es passieren, dass für bestimmte RVK-Notationen gar keine passenden E-Ressourcen zu finden sind. Dies ist z.B. für die Notation *ST 166* (Prozessrechner) der Fall. Hier könnte ggf. versucht werden, die natürlichsprachliche Bezeichnung der Notation stattdessen als Schlagwort zu recherchieren. Gegebenenfalls könnte auch überlegt werden, zusätzlich benachbarte Notationen anzufragen, sofern diese inhaltlichen Bezug haben oder dem Nutzer die Möglichkeit bieten, dies aktiv anstoßen. Beim Stöbern am Regal sieht der Nutzer schließlich unmittelbar anschließend Bände der benachbarten RVK-Systemstellen.

Das Problem bei einer punktuellen Suche nach RVK-Notationen, wie sie bisher beschrieben wurde ist, dass es E-Ressourcen geben kann, die mit RVK-Notationen erschlossen sind, zu welchen es keinen Printbestand gibt. Dieses Problem ließe sich mit Hilfe einer Bereichssuche lösen. Anstatt punktuell nach bestimmten Notationen zu suchen, könnten Ressourcen zu einer bestimmten Range gesucht werden. Hier wäre es wichtig, geeignete Ranges zu finden, sodass die Treffermengen weder zu groß noch zu klein sind und vor allem inhaltlich noch zusammenpassen.

An dieser Stelle zeigt sich bereits, dass allein die RVK-Notation als Schnittstelle zwischen gedrucktem und elektronischem Bestand zu einem bestimmen Fachbereich zum Teil nicht ausreichend sein kann.

### 6.5.3 Sonderfall E-Journals

Schaut man sich nun weiterhin die in Wilbert hinterlegten Sacherschließungs-Informationen von E-Journals an, stellt man fest, dass anders als bei E-Books keine spezifische RVK-Notation hinterlegt ist, sondern lediglich ein übergeordnetes Thema. Dabei handelt es sich um die über Buchstaben abgebildeten übergeordneten Themenbereiche der RVK-Systematik (z.B. *ST*). Erklären lässt sich dies dadurch, dass Zeitschriften anders als Monographien (sei es elektronisch oder gedruckt) Artikel zu verschiedenen Themengebieten beinhalten, die sich nicht so spezifisch abbilden lassen, wie eine einmalig erscheinende Publikation. Zusätzlich zu dieser Information sind E-Journals (ebenso wie Monographien) mit Schlagwörtern versehen, welche als Fremddaten aus der Zeitschriftendatenbank (ZDB) und der Elektronischen Zeitschriftenbibliothek (EZB) übernommen werden. Zusätzlich finden sich zum Teil Informationen zur DDC.

Würde nun eine Suche im Bereich der RVK-Notation mit „ST“ durchgeführt werden, würde dies das Problem der bereits bei den E-Books beschrieben zu großen Trefferzahlen nochmals verschärfen. Eine

---

<sup>184</sup> Vgl.:

[https://wilbert.kobv.de/advancedSearch.do?fq=datenlieferant%3afhw\\_ebooks&sortCrit=score&sortOrder=desc&hitsPerPage=10&fq=search\\_space%3abooks&index=internal&f1=rvk&v1=ST+250&conj1=and&f2=keyword&v2=Java&conj2=and&f3=title&v3=&conj3=and&f4=title&v4=&conj4=and&f5=title&v5=&conj5=and&f6=title&v6=&conj6=and&f7=title&v7=&plv=2](https://wilbert.kobv.de/advancedSearch.do?fq=datenlieferant%3afhw_ebooks&sortCrit=score&sortOrder=desc&hitsPerPage=10&fq=search_space%3abooks&index=internal&f1=rvk&v1=ST+250&conj1=and&f2=keyword&v2=Java&conj2=and&f3=title&v3=&conj3=and&f4=title&v4=&conj4=and&f5=title&v5=&conj5=and&f6=title&v6=&conj6=and&f7=title&v7=&plv=2), 24.07.2021

Suche nach einer vollständigen RVK-Notation (z.B: *ST 250* anstelle von *ST*) würde jedoch wiederum ggf. dazu führen, dass unter den Treffern keine Zeitschriftentitel enthalten sind. Dieses Problem ließe sich durch die Ausführung zweier separater Suchanfragen lösen (für Zeitschriften und Monographien), bei dem beide Treffermengen anschließend zusammengeführt werden. Für die ggf. vorher ermittelten DDC-Notationen könnte zusätzlich eine dritte Anfrage ausgeführt werden.

#### 6.5.4 Anfragen an Solr-Proxy

Sind alle für die Abfrage/n notwendigen Informationen ermittelt, kann diese zusammengestellt und abgesendet werden.

Eine kombinierte Abfrage nach E-Books zur RVK-Notation *ST 250* kombiniert mit dem Schlagwort *Java* würde folgendermaßen aussehen:

```
https://solrproxy-wilbert-test.kobv.de/wilbertselect?access=XXXXXXXXXX
&search_space=books&fq=datenlieferant:fhw_ebooks&
q=rvk:%22ST%20250%22+schlagwort:java
```

Eine Abfrage nach Zeitschriftentiteln zum Fachgebiet Informatik, bspw. zum Thema *Datenbanken* könnte folgendermaßen aussehen:

```
https://solrproxy-wilbert-test.kobv.de/wilbertselect?access=XXXXXXXXXX
&fq=search_space:journals+schlagwort:datenbank*&q=topic:SQ-SU
```

### 6.6 Ermittlung der „neusten“ E-Books

Um die neuesten E-Books zu ermitteln, bietet Wilbert die Abfragemöglichkeit „latestbooks“:

```
https://wilbert.kobv.de/latestBooks.do?page=2&sortCrit=score&sortOrder=desc&fq=search_s
pace%3abooks&fq=datenlieferant%3afhw_ebooks&index=internal&plv=2
```

Ermittelt werden diese auf Grund der Einträge im Feld „insertion\_date“ stehenden Daten. Problematisch bei diesem Feld ist allerdings, dass der Inhalt des Feldes nicht nur beim neu-hinzufügen des Datensatzes beschrieben wird, sondern auch beim Ändern eines Datensatzes angepasst wird. Demzufolge muss der Inhalt des Feldes nicht unbedingt dem Datum entsprechen, an dem die Ressource in die Bibliothek gelangt ist. Grund dafür ist, dass das Mapping hier auf Grundlage des MARC-Feldes 005 (Datum und Zeit der letzten Transaktion) erfolgt.

Da die API vom Solr-Proxy die Funktion „latestbooks“ aus Wilbert nicht direkt anbietet, soll versucht werden, mit Hilfe der vorhandenen Attribute die Suche nachzustellen:

In Ermangelung passenderer Feldinhalte, soll hier trotzdem mit Hilfe des Feldes „insertion\_date“ aus Solr gearbeitet werden, um die neuesten E-Books abzubilden.

Eine Anfrage an die Solr-API ließe sich bspw. auf einen festgelegten Zeitraum ausführen. Möchte man bspw. die neuesten E-Books der letzten zwei Wochen ermitteln, wird zunächst über das Frontend das

entsprechende Startdatum der Anfrage ermittelt, das Enddatum kann offen gelassen werden. Die entsprechende Solr-API-Anfrage sähe dann folgendermaßen aus:

```
https://solrproxy-  
wilbert.kobv.de/wilbertselect?access=XXXXXXXXXX&q=datenlieferant:fhw_ebooks&fq=search  
h_space:books+insertion_date%3a%5b2021-07-13%20TO%202021-07-26%5d
```

Diese Anfrage würde E-Books mit einem Insertion-Date zwischen 13. und 26.07. ermitteln. Der zweite Wert kann dabei trunkiert werden.

Problematisch an dieser Anfrage ist jedoch, dass zuvor eine Festlegung eines bestimmten Zeitrahmens erfolgen muss, welcher abgefragt wird. Entscheidet man sich bspw. für eine Woche, kann es nach Feiertagen wie Weihnachten und Neujahr durchaus passieren, dass keine E-Books mit entsprechendem Eintrag im Feld `insertion_date` ermittelt werden können. Zudem kann es passieren, dass zu bestimmten Zeiträumen keine neuen E-Books eingespielt wurden. Dies hätte eine leere Liste auf dem Dashboard, welches zeitgleich die Startseite der Anwendung ist, zur Folge.

Um dies zu verhindern, sollte eine Abfrage eines statischen Zeitrahmens vermieden werden.

Stattdessen könnte eine Sortierung der Treffer absteigend nach `insertion_date` erfolgen. Um den Nutzer nicht mit einer zu großen Treffermenge zu überfordern, kann diese bspw. über den Parameter `rows` auf 20 Treffer beschränkt werden. Eine entsprechende Anfrage sähe folgendermaßen aus:

```
https://solrproxy-  
wilbert.kobv.de/wilbertselect?access=w21%234ICUb&q=datenlieferant:fhw_ebooks&fq=search  
_space:books&sort=insertion_date%20desc&rows=20
```

## 6.7 Ermittlung entliehener Print-Monographien

Die Ermittlung entliehener Printmedien bezieht sich immer auf den Standort, sodass auch hier eine Abfrage mit Hilfe der RVK-Notation erfolgen sollte. Für diesen Zweck stand bisher keine entsprechende Schnittstelle zur Verfügung. Über Wilbert und die Solr-API lassen sich entsprechende Informationen nicht abfragen, da diese keine Exemplardaten wie Signaturen oder Verfügbarkeitsinformationen abbilden. Zudem handelt es sich bei der Suche anhand einer bestimmten RVK-Notation um eine recht spezifische Eingrenzung der Medien, die standardmäßig nicht über die SISIS-Schnittstelle SLNP (Simple Library Network Protocol) umgesetzt ist.

Nach Übermittlung der entsprechenden Anforderungen wurde durch den IT-Beauftragten der Hochschulbibliothek eine entsprechende Schnittstelle erstellt.<sup>185</sup> Als Antwort liefert die API ein Array mit den Treffern als JSON-Objekte.

Die RVK-Notation, nach der die Ergebnisse gefiltert werden sollen, werden als GET-Parameter übergeben. Eine Anfrage sieht dann bspw. für die RVK-Notation ST 250 folgendermaßen aus:

```
https://kobz22.bib-bvb.de/cgi-bin/btw-cgi/ebooktogo/checkedOutItems.pl?key=ST250
```

---

<sup>185</sup> Die E-Mail mit den Anforderungen an die Schnittstelle findet sich in Anhang A.6.

Die Antwort sähe dann folgendermaßen aus:

```
{
  "returnVal" : "ok",
  "requestId" : "ST250",
  "timestamp" : 1627321548,
  "items" : [
    {
      "status" : 4,
      "place" : "München [u.a.]",
      "autor_avs" : "Sedgewick, Robert",
      "key" : 106112,
      "holds" : 0,
      "checkedOutItems" : "1",
      "isbn" : "978-3-86894-076-3",
      "publisher" : "Pearson",
      "sign" : "ST 250 J35 S448",
      "zusatz" : " ",
      "barcode" : "201227078",
      "mediatype" : 0,
      "title_avs" : "Einführung in die Programmierung mit Java",
      "totalItems" : "1",
      "verfasser" : "Sedgewick, Robert;Wayne, Kevin",
      "title" : "Einführung in die Programmierung mit Java",
      "duedate" : "Dec 24 2021 12:00AM",
      "year" : "2011"
    },
    {
      "status" : 4,
      "place" : "Bonn",
      "autor_avs" : "Ackermann, Philip",
      "key" : 239398,
      "holds" : 0,
      "checkedOutItems" : "1",
      "isbn" : "978-3-8362-3838-0",
      "publisher" : "Rheinwerk",
      "sign" : "ST 250 J35 A182(1)",
      "zusatz" : "das umfassende Handbuch",
      "barcode" : "201900778",
      "mediatype" : 0,
      "title_avs" : "JavaScript",
      "totalItems" : "1",
      "verfasser" : "Ackermann, Philip",
      "title" : "JavaScript",
      "duedate" : "Dec 24 2021 12:00AM",
      "year" : "2016"
    }
  ],
  [...]
}
```

*Listing 3: Antwort der API für entliehene Bände (Ausschnitt)*

Einen Sonderfall stellen hier jedoch die Bände dar, die zum Semesterapparat gehören. Diese gehören zum Präsenzbestand der Bibliothek. Das heißt, dass von den als Semesterapparat gekennzeichneten Exemplaren immer ein Band vorhanden ist, da diese nicht ausgeliehen werden dürfen. Im Bibliothekssystem ist die Zugehörigkeit zum Semesterapparat durch eine Ausleihe auf diesen umgesetzt, sodass auch vorhandene Semesterapparate als entliehene Medien angezeigt werden würden. Dies hat zur Folge, dass entsprechende Titel über die API zurückgegeben werden, obwohl auf jeden Fall ein Exemplar im Regal stehen müsste. Hier muss ggf. mit entsprechenden Filtern gearbeitet werden.

## 6.8 Layout

Um eine nutzerfreundliche Webanwendung zu gestalten, ist ein übersichtliches, klar strukturiertes Layout von großer Bedeutung. Da es sich um eine mobile Anwendung handelt, sollte zudem Responsivität gewährleistet sein.

### 6.8.1 Allgemeine Anforderungen

Um eine möglichst große Usability zu gewährleisten, bietet es sich an, bewährte Konzepte, bspw. zum Seitenaufbau, zu berücksichtigen. Im Bereich von Webangeboten hat sich eine Gliederung in Header, Navigationsmenü, Hauptbereich und Footer durchgesetzt (siehe Abb. 22).

Im Header findet sich der Titel der Seite und ggf. ein Logo. „In der Navigation werden die Links zu den Unterseiten der Webs[e]ite bereitgestellt.“<sup>186</sup> Der Hauptbereich nimmt den größten Platz ein. Hier findet sich der eigentliche Inhalt der Webseite. Im Footer finden sich häufig „Informationen zum Impressum, Copyright und Kontakt[...]“.<sup>187</sup>

Bei zahlreichen Seiten hat sich die Unterbringung der Navigationsleiste auf der linken oder rechten Seite etabliert.

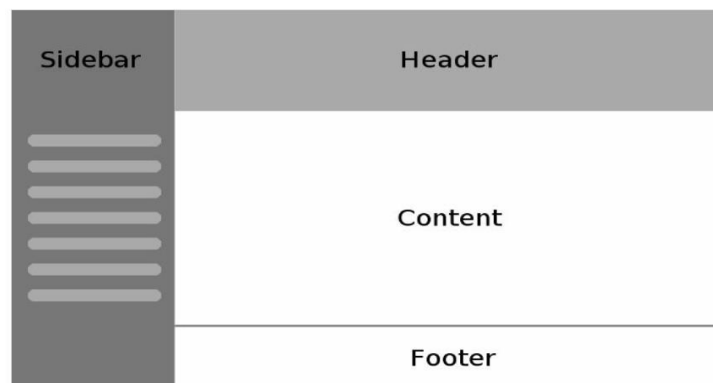


Abbildung 22: Seitenaufbau einer Webseite  
Quelle: Diaz (2013)]

Neben der Strukturierung der Seite bietet es sich zudem an, einzelne Elemente und Gewohnheiten aus der analogen Welt auch im Digitalen anzuwenden. Dieses Ziel verfolgt das Material Design, welches von Google entwickelt wurde. Für Vue.js werden mit Hilfe der Bibliothek Vuetify entsprechende Material-Design-Elemente bereitgestellt, um eine möglichst hohe Usability zu erreichen. Die Anwendung wird deshalb empfohlen.

Neben der Positionierung der Navigation ist es vor allem wichtig, sprechende Namen für die einzelnen Menüpunkte zu verwenden.

---

<sup>186</sup> Vgl. Mozilla (28.07.2021): Struktur in die Webseite bringen, [https://developer.mozilla.org/de/docs/Learn/HTML/Introduction\\_to\\_HTML/Document\\_and\\_website\\_structure](https://developer.mozilla.org/de/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure), 28.07.2021.

<sup>187</sup> Ebd.

Für die angebotenen Funktionalitäten werden folgende Benennungen vorgeschlagen:

- Dashboard: „Home“/“Home“
- Ermittlung passender E-Books mit Hilfe von QR-Code:  
„QR-Code scannen“/„Scan QR-Code“
- Suchfunktion: „E-Books suchen“/“Search E-Books“
- Hilfefunktion: „Hilfe“/“Help“
- Kontakt: „Kontakt“/“Contact“
- Impressum: „Impressum“/“Imprint“
- Datenschutz: „Datenschutz“/“Privacy“

Nach Anmeldung stehen zudem folgende Menüpunkte bereit:

- Literaturempfehlungen zum eigenen Studiengang:  
„Studiengangsbezogene Empfehlungen“/“Recommendations for your degree programme“
- Browsen nach Literaturempfehlungen anderer Studiengänge:  
„Mehr Empfehlungen“/“More recommendations“

Wichtig bei der Umsetzung des Layouts generell ist darüber hinaus die Integration des Corporate Designs der Bibliothek bzw. der Technischen Hochschule, welcher die Bibliothek organisatorisch zugeordnet ist. So könnte bspw. das Logo der Technischen Hochschule Wildau im Header untergebracht werden oder das Farbkonzept der Hochschule aufgegriffen werden.

Weiterhin ist bei der Entwicklung des Layouts darauf zu achten, dass die Anwendung responsiv nutzbar bleibt, sodass hier bspw. für einzelne Bildschirmgrößen Elemente anders angeordnet werden oder anstatt eines Sliders eine Liste verwendet wird.

## 6.8.2 Darstellung der E-Books durch ihre Cover

Um den Nutzern die gefundenen Medien ansprechend zu präsentieren, sollten diese in den Trefferlisten durch ihre Cover abgebildet werden. Bevor die Listen erzeugt werden können, müssen also die Cover zu den Treffern ermittelt werden.

Leider finden sich in der Antwort der Solr-API keine Informationen dazu, sodass die notwendigen Daten über andere Schnittstellen ausfindig gemacht werden müssen. Wilbert selbst nutzt für die Ermittlung von Buchcovern lediglich die Google-Books-Schnittstelle.<sup>188</sup> Problematisch dabei ist jedoch, dass sich Regeln und Nutzung einer API ändern können. Empfohlen wird deshalb die Nutzung von mindestens zwei Schnittstellen zur Ermittlung von Covern. Dadurch ist auch sichergestellt, dass bei Ausfall einer API trotzdem weiterhin Cover ermittelt werden können. Durch Abfrage mehrerer Schnittstellen kann zudem aus einem größeren Datenpool geschöpft werden und bei Vorhandensein mehrerer Cover das nach den eigenen Anforderungen beste ausgewählt werden.<sup>189</sup>

---

<sup>188</sup> Vgl. Attia (2020b): Praktikumsbericht, S. 3.

<sup>189</sup> Vgl. Attia (2020b): Praktikumsbericht, S. 2.



Für die Ermittlung von Buch-Covern könnten folgende Schnittstellen (ohne Anmeldung)<sup>190</sup> genutzt werden:

- Google-Books-API
- Open-Library-Books-API
- (Amazon-Product-Advertising-API)
- Amazon-http-Service

Ziel der Google-Books-API ist es, Buchinhalte besser über das Web auffindbar zu machen.<sup>191</sup> Über folgende Schnittstelle können bspw. anhand der ISBN (10- oder 13-stellig) Buch-Cover ermittelt werden:

```
https://books.google.com/books?bibkeys=ISBN:0451526538&jscmd=viewapi&callback=mycallback
```

Als Antwort werden ein oder mehrere JSON-Objekt/e generiert. Für die obige Anfrage erhält man folgende Antwort:

```
mycallback({"ISBN:0451526538":{"bib_key":"ISBN:0451526538","info_url":"https://books.google.com/books?id=7osYCoz_gdgC\u0026source=gbs_ViewAPI","preview_url":"https://books.google.com/books?id=7osYCoz_gdgC\u0026source=gbs_ViewAPI","thumbnail_url":"https://books.google.com/books/content?id=7osYCoz_gdgC\u0026printsec=frontcover\u0026img=1\u0026zoom=5","preview":"noview","embeddable":false,"can_download_pdf":false,"can_download_epub":false,"is_pdf_drm_enabled":false,"is_epub_drm_enabled":false}});
```

Listing 4: Antwort der Google-Books-API

Im Feld *thumbnail\_url* findet sich die URL zum Cover. Schaut man sich die URL genauer an, stellt man fest, dass zunächst noch die Unicode-Zeichenkette „\u0026“ für das Ampersand („&“) decodiert werden muss. Anschließend steht einem die URL zum Buchcover bereit.

Buch-Cover werden auch durch die Open Library Covers API bereitgestellt. Anfragen sind wie bei der Google Books API anhand der ISBN oder weiterer Parameter wie OCLC-ID möglich. Cover werden in drei Größen angeboten:

*S*: Small, suitable for use as a thumbnail on a results page on Open Library,  
*M*: Medium, suitable for display on a details page on Open Library and,  
*L*: Large<sup>192</sup>

<sup>190</sup> Zusätzlich bietet auch buchhandel.de einen solchen Service. Dieser steht allerdings nur nach Anmeldung bereit, siehe: <https://www.buchhandel.de/angebot-fuer-bibliotheken>, 14.08.2021

<sup>191</sup> Vgl. Google Developers (n. N.): Google Books API : Integrate with the Google Books repository, <https://developers.google.com/books>, 27.07.2021.

<sup>192</sup> Vgl. Open Library (2021): Open Library Covers API, <https://openlibrary.org/dev/docs/api/covers>, 27.07.2021.

Zu beachten ist, dass es eine Zugriffslimitierung gibt: Derzeit sind 100 Requests/IP innerhalb von fünf Minuten zugelassen. Werden mehr Anfragen gesendet, wird automatisch ein „403 Forbidden“-Status gesendet.<sup>193</sup>

Um Buch-Cover zu erhalten, fragt man eine URL mit folgendem Muster an:

*[http://covers.openlibrary.org/b/\\$key/\\$value-\\$size.jpg](http://covers.openlibrary.org/b/$key/$value-$size.jpg)*

Eine Anfrage mit einer ISBN für ein Cover im Format M sieht dann folgendermaßen aus:

*<http://covers.openlibrary.org/b/isbn/9780385472579-M.jpg>*

Als Antwort erhält man das Cover bzw. ein leeres Image, wenn kein Cover gefunden werden konnte.

Darüber hinaus gibt es die Amazon-Product-Advertising-API (APA). Für die Nutzung der API ist jedoch ein API-Key erforderlich. Die API steht also nicht frei zur Verfügung und soll deshalb nicht eingesetzt werden.

Statt der Nutzung der APA können über einen http-Service von Amazon mit einem GET-Request an URLs nach folgendem Muster einzelne Book-Cover abgefragt werden:

*[https://images-na.ssl-images-amazon.com/images/P/\[ISBN\].01.MZZZZZZZZZ.jpg](https://images-na.ssl-images-amazon.com/images/P/[ISBN].01.MZZZZZZZZZ.jpg)*

Für die Abfrage kann hier nur die ISBN verwendet werden. Es kann immer nur ein Cover pro Abfrage ermittelt werden.

Als Antwort erhält man entweder das Cover oder aber ein 1px großes, undefiniertes Bild.<sup>194</sup>

Zusätzlich zum Cover könnte bspw. auch die Seitenzahl durch einen entsprechenden senkrechten Balken links vom Cover abgebildet werden. Problematisch sind hier jedoch die vorhandenen Umfangsangaben zu E-Books. Leider enthalten nicht alle E-Books Angaben zur Seitenzahl, so dass dies nicht vollständig umgesetzt werden könnte.

### 6.8.3 Trefferlisten

Zielstellung ist hier, dass dem Nutzer ein möglichst vollständiger Überblick über die sich zu einem bestimmten Themengebiet im Bestand der Bibliothek vorhandene Literatur präsentiert wird. Wenn der Nutzer vor Ort in der Bibliothek ist, ist es für ihn ggf. erst einmal nicht relevant, ob es sich dabei um (entliehene) gedruckte oder elektronische Ressourcen handelt. Gedruckte (*hier: entliehene Bände*) und elektronische Ressourcen können somit in einer gemeinsamen Trefferliste angezeigt werden. Sollen nur elektronische oder gedruckte Ressourcen angezeigt werden, so bietet sich die Eingrenzung mit Hilfe eines Filters an. Jedoch sollten die verschiedenen Ressourcentypen (*hier: E-Book, E-Journal oder Printbuch*) in den Trefferlisten optisch voneinander differenziert werden können. Hinzu kommt, dass bei einer E-Ressource die Möglichkeit angeboten werden kann, über eine Verlinkung direkt aus der Trefferliste heraus zur E-Ressource geleitet zu werden (ggf. via Durchschaltung, siehe Kapitel 6.13).

---

<sup>193</sup> Vgl. Open Library (2021): Open Library Covers API, <https://openlibrary.org/dev/docs/api/covers>, 27.07.2021.

<sup>194</sup> Vgl. Attia (2020b), S. 9.

Die Darstellung der verschiedenen Medien soll auch hier mit Hilfe der Cover erfolgen.

Zur Darstellung von Trefferlisten bieten sich verschiedene Möglichkeiten an. Einerseits könnten Slider genutzt werden, die das horizontale Navigieren (auf einem Smartphone seitwärts *Wischen*) ermöglichen. Andererseits besteht die Möglichkeit, die Ressourcen in Form einer Liste anzuordnen. In dem Fall werden Medien, die nicht mehr in eine Reihe passen, in die nächste Reihe angeordnet. Auf dem Smartphone würde dies einem *nach unten Wischen* entsprechen.

#### 6.8.4 Detaildarstellung

Zusätzlich zur Anzeige des Covers sollen hier weitere Informationen zum Medium dargestellt werden. Dazu gehören:

- Angaben zu/m Verfasser/n
- Titel und Titelzusatz
- Erscheinungsjahr
- Verlag und Verlagsort

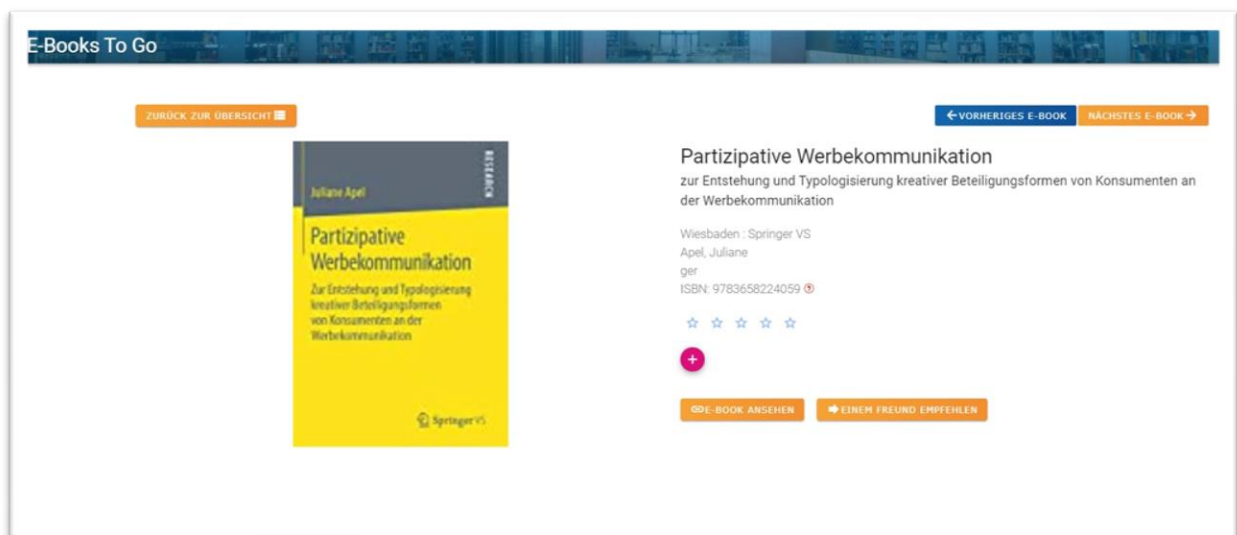


Abbildung 23: Mockup zur Einzelansicht eines E-Books

Hier werden, abhängig vom Medientypen, verschiedene Funktionen angeboten. Bei E-Ressourcen erfolgt hier die Verlinkung zum Volltext, ggf. via Durchschaltung zum PDF (siehe Kapitel 6.13). Bei ausgeliehenen Printmedien besteht an dieser Stelle nach Authentifizierung die Möglichkeit, eine Vormerkung aufzugeben.

Für alle Medientypen kann ggf. zudem die Möglichkeit angeboten werden, das Medium zu bewerten oder per E-Mail als Empfehlung zu versenden.

In der Einzelanzeige sollte eine Verknüpfung zurück zur Trefferliste untergebracht werden. Ideal wäre darüber hinaus die Möglichkeit, in der Detailansicht Verknüpfungen zum vorherigen bzw. nächsten Treffer aus der Liste zu visualisieren, sodass direkt zwischen den Treffern gewechselt werden kann, ohne jedes Mal wieder zurück in die Trefferliste gehen zu müssen.

## 6.9 Filter- und Sortiermöglichkeiten

Wie bereits in Kapitel 6.5.2 beschrieben, kann es nicht immer verhindert werden, dass es zu großen Treffermengen kommt. Umso wichtiger ist es, dem Nutzer auch nach Absetzen der Suchanfrage die Möglichkeit zu geben, die Treffer nochmals zu filtern. Dabei sind verschiedene Filtermethoden und Herangehensweisen denkbar.

Zunächst kann nach Einstellen der Filter eine erneute Suchanfrage an die API mit den entsprechenden Attributen geschickt werden. Andererseits ist es möglich, Sortieroperationen auf dem bereits vorhandenen Treffer-Array auszuführen.

Als Filter sind verschiedene Bereiche denkbar. Da beim Absetzen einer Suchanfrage nach Einlesen eines QR-Codes zunächst alle Medientypen in einer Trefferliste abgebildet werden sollen, ist in dem Fall ein möglicher Filter nach Medientyp denkbar: E-Book, entliehener Printband, Printzeitschrift und E-Journal. Wobei letztere schwer zu differenzieren sind, da bei Zeitschriften zum Teil einige Jahrgänge als gedruckte Bände und andere online vorliegen. Deshalb sollte hier nur der Medientyp Journal angegeben werden.

Darüber hinaus bietet es sich für die Listen, die nach einfacher Suche und nach Einlesen des QR-Codes erzeugt werden, an, nach bestimmten Erscheinungszeiträumen zu filtern. Die Solr-API bietet hier die Möglichkeit, ab dem Jahr 2000 in 5-Jahres-Schritten zu filtern.

Eine Abfrage nach E-Books zur RVK-Notation *ST 250* mit Filter auf die Erscheinungsjahre *2015-2019* würde folgendermaßen aussehen.

```
https://solrproxy-wilbert-  
test.kobv.de/wilbertselect?access=XXXXXXXXXX&fq=datenlieferant:fhw_ebooks&fq=publikati  
onsjahr_intervall:%227984%3a2015-2019%22&q=rvk:%22ST%20250%22
```

Darüber hinaus sollten bei der freien Suche Sortiermöglichkeiten der Treffer angeboten werden. Wie im bestehenden Prototyp bereits umgesetzt, sollten Sortierungen nach folgenden Kriterien erfolgen:

- Relevanz (Standard)
- Autor
- Titel
- Erscheinungsjahr

Die entsprechenden Parameter für die Solr-API sehen folgendermaßen aus:

- Sortierung nach Verfasser (auf- bzw. absteigend)  
*„sort=autor\_sort%20asc“* bzw. *„sort=autor\_sort%20desc“*
- Sortierung nach Titel (auf- bzw. absteigend)  
*„sort=titel\_sort%20desc“* bzw. *„sort=titel\_sort%20asc“*
- Sortierung nach Erscheinungsjahr  
*„sort=publikationsjahr\_sort%20desc“* bzw.  
*„sort=publikationsjahr\_sort%20asc“*

## 6.10 Authentifizierung

Bestimmte Funktionen der Anwendung sollen nur für Hochschulangehörige zur Verfügung stehen: die Möglichkeiten der Bewertung einzelner Ressourcen sowie der Vormerkung entliehener Bände. Dazu ist eine Anmeldung mit den Nutzerdaten der Hochschule notwendig. Diese sind in einem zentralen Verzeichnis hinterlegt, auf welches man mittels des Lightweight Directory Access Protocols (LDAP) zugreifen kann.

Zunächst müssen dafür die für die weiteren Funktionalitäten benötigten Daten analysiert werden. Diese können nach Abfrage der Schnittstelle im Store der Anwendung gespeichert werden, so dass diese auch später zur Verfügung stehen.

Für den Empfehlungsdienst werden hier vor allem Daten zum Studiengang, in dem ein Student eingeschrieben ist, benötigt. Für die Auslösung einer Vormerkung wird die Bibliotheks-ID benötigt, welcher nicht der Hochschulkennung entspricht. Um dem Prinzip der Datensparsamkeit zu folgen, sollen keine weiteren Daten gespeichert werden.

Für Anfragen des LDAPs der Technischen Hochschule Wildau steht eine entsprechende API zur Verfügung. Die benötigten Informationen können anschließend in der Anwendung gespeichert werden, solange der Nutzer angemeldet ist.

Untergebracht werden könnte die Authentifizierungsfunktion als oberster Punkt in der Navigationsleiste. Die Eingabe der Nutzerdaten kann in einem Popup-Fenster ermöglicht werden.

Nach erfolgreicher Anmeldung ändert sich der obere Bereich der Navigationsleiste. Anstelle der Login-Funktion steht nun die Logout-Funktion zur Verfügung. Zusätzlich werden außerdem der Name und die Rolle, bspw. Student, angezeigt.

Sollte bei der Authentifizierung etwas schiefgelaufen sein, sollte der Nutzer in entsprechender Art und Weise darüber informiert werden. Bei Eingabe eines falschen Passwortes könnte diese Information bspw. direkt im Popup-Fenster, in welchem die Anmeldedaten eingegeben werden sollen, angezeigt werden. Zur Hervorhebung, dass etwas nicht geklappt hat, könnte die Mitteilung bspw. in roter Schrift hervorgehoben werden (siehe Abb. 24).

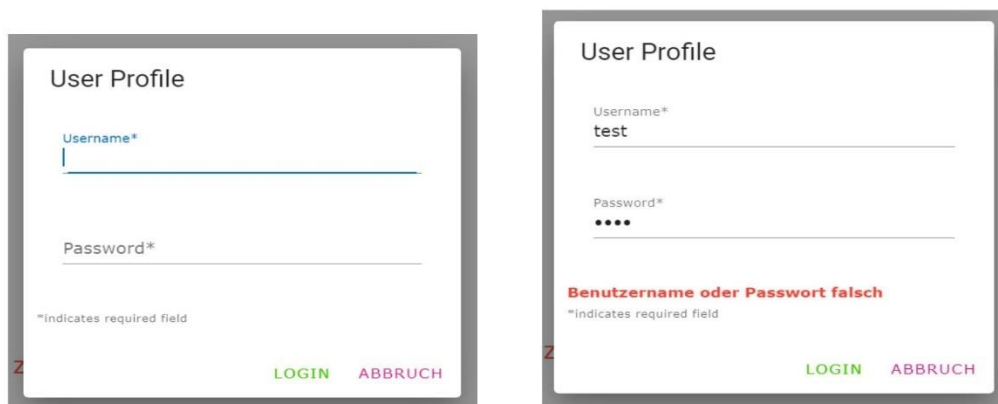


Abbildung 24: Anmeldefenster des Prototyps der App  
Quelle: Attia (2020a), S. 53 [Abbildung 6.2.1: Anmeldefenster von E-Books-To-Go]

## 6.11 Vormerkung

Bereits Wilbert nutzt einen Webservice, der es ermöglicht, aus dem Discovery-System heraus Vormerkungen auf entliehene Bände aufzugeben. Leider lässt sich dieser im Rahmen der zu entwickelnden Anwendung nicht nachnutzen, da über Wilbert Daten an den Webservice weitergegeben werden, die im Falle der Web-App nicht vorhanden sind. Deshalb wurde durch den IT-Verantwortlichen der Bibliothek auf Grundlage von SIP2-Anfragen an das Bibliothekssystem SISIS eine API entwickelt, mit der die Aufgabe von Vormerkungen aus der Anwendung heraus möglich ist. Für die Nutzung der API ist die Autorisierung mit einer entsprechenden Kennung notwendig.

Darüber hinaus ist es wichtig, dass die API teilqualifizierte Vormerkungen ausführt, also die Vormerkung sich auf einen Titel und nicht auf ein bestimmtes Exemplar beziehen.<sup>195</sup> Vorteil dabei ist, dass der Nutzer das Exemplar bereitgestellt bekommt, das als nächstes zurückgegeben wird.

Bei Aufgabe der Vormerkung über die API ist neben dem Barcode des vorzumerkenden Mediums auch die *library\_id* des Nutzers anzugeben. Diese kann der LDAP-API-Antwort entnommen und im System für die Zeit der Anmeldung hinterlegt werden, sodass bei Auslösung einer Vormerkung darauf zugegriffen werden kann. Den Barcode der entsprechenden Medien enthält man über die Antwort der API zur Ermittlung der entliehenen Bände (siehe Kapitel 6.7).

Die Funktion könnte in der Detailansicht einzelner Medien untergebracht werden und sollte aufgrund der benötigten Informationen nur nach Anmeldung bereitstehen.

Nach Aktivieren der Funktion sollte sicherheitshalber ein Popup-Fenster erscheinen, um die gewünschte Aktion nochmals zu bestätigen. Damit könnte vermieden werden, dass die Vormerk-Funktion versehentlich aktiviert wird.

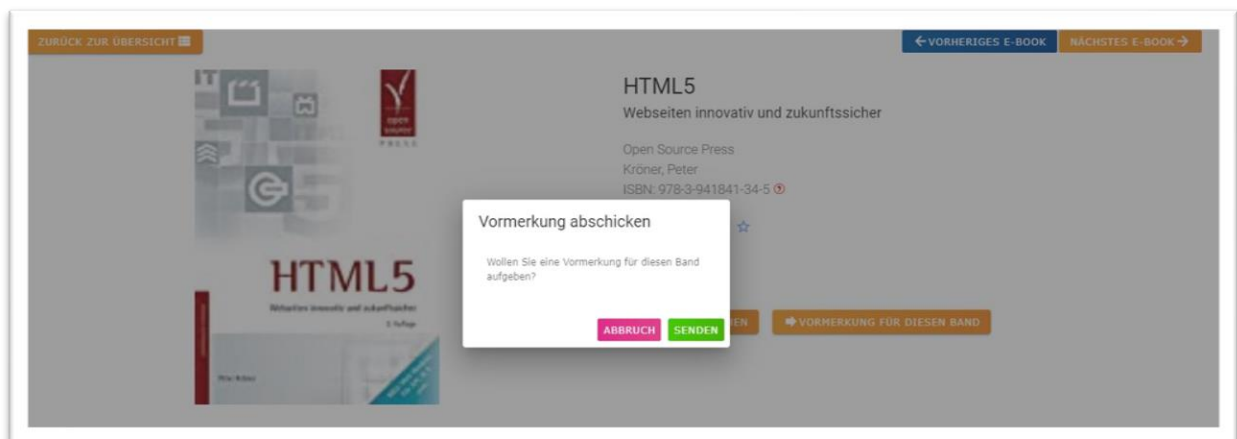


Abbildung 25: Mockup zur Bestätigung des Vormerk-Wunsches im Popup-Fenster

Nach Absenden der Vormerkung sollte der Nutzer entsprechend über Erfolg oder Misserfolg des Vormerkwunsches informiert werden. Dies könnte ähnlich zur Authentifizierung umgesetzt werden, indem

<sup>195</sup> Vollqualifizierte Vormerkungen hingegen beziehen sich auf ein ganz bestimmtes Exemplar.

die entsprechenden Informationen auch hier bei Misserfolg im Popup-Fenster angezeigt werden. Bei Erfolg kann bspw. der Schriftzug *Vormerkung erfolgreich* eingeblendet werden.

Auf die Integration der Möglichkeit, eine Vormerkung zu stornieren soll an dieser Stelle verzichtet werden, da Zielstellung der App nicht die Nachbildung sämtlicher Funktionen des Bibliothekskontos sein soll.

## 6.12 Empfehlungen

Die Möglichkeit, über das Internet Empfehlungen auszutauschen, entstand bereits mit der Entstehung des Web 2.0. Immer mehr Online-Angebote generieren Empfehlungen für die Nutzer. Um den Nutzern der zu entwickelnden Anwendung einen zusätzlichen Service anzubieten, können auch hier verschiedene Arten von Empfehlungen angeboten werden.

### 6.12.1 Anzeigen von Empfehlungen für eigenen Studiengang anzeigen

Für die Vor- und Nachbereitung von Seminaren erhalten Studenten Literaturempfehlungen zu einzelnen Seminaren. Diese sind im Modulhandbuch hinterlegt.

In den Modulhandbüchern der Technischen Hochschule Wildau finden sich darüber hinaus Literaturhinweise zu den einzelnen Seminaren. Somit könnten diese als Quelle für Literaturempfehlungen zum eigenen Studium dienen.

<b>Pflichtliteratur:</b>
<b>Gumm, H. &amp; Sommer, M.</b> (2013). <i>Einführung in die Informatik</i> . München: Oldenbourg. <b>Ratz, D.</b> (2011). <i>Grundkurs Programmieren in Java</i> . München [u.a.]: Hanser. <b>Mössenböck, H.</b> (2014). <i>Sprechen Sie Java?: Eine Einführung in das systematische Programmieren</i> . dpunkt.verlag GmbH.
<b>Empfohlene Literatur:</b>
<b>Hunt, A. &amp; Thomas, D.</b> (2004). <i>Unit-Tests mit JUnit</i> . München: Hanser. <b>Cormen, T.</b> (2007). <i>Algorithmen</i> . München [u.a.]: Oldenbourg. <b>Rupp, C. &amp; Queins, S.</b> (2012). <i>UML 2 glasklar</i> . München: Hanser.

Abbildung 26: Literaturhinweise zum Modul Programmierung des Masterstudiengangs  
Bibliotheksinformatik  
Quelle: TH Wildau (2019), S. [11]

Durch das iCampus-Team wurde ein spezieller Webservice für Abfrage von Informationen aus den Modulhandbüchern erstellt.<sup>196</sup> Für die Abfrage werden zwei Schlüsselwerte benötigt: die Seminargruppe des Studenten und den Login-Token. Beide Informationen sind in der Antwort der Authentifizierungs-API enthalten.

<sup>196</sup> Vgl. Attia (2020a): Entwicklung eines Recommender-Systems für E-Book-Empfehlungen in der Hochschulbibliothek der TH Wildau, S. 31.

Grundsätzlich kann die Literatur im Modulhandbuch durch die Dozenten auf drei verschiedene Arten hinterlegt werden:

- (1) Library: Hinterlegung einer ID (hier: Catkey), nach der direkt in Wilbert und über die Solr-API gesucht werden kann
- (2) Citation: Manuelle Eintragung von Titel und Autor
- (3) Custom: Freitexteingabe der Informationen, daher keine einheitliche Struktur<sup>197</sup>

Während Angaben vom Typ *Library* einfach recherchiert werden können, müssten die Angaben vom Typ *Citation* oder *Custom* zunächst geparkt werden, um die geeigneten Informationen bspw. zu Autor oder Titel eines Bandes zu ermitteln.

Anschließend kann über die Solr-API eine entsprechende Abfrage abgesetzt werden, um zu ermitteln, ob sich die Literatur im Bestand der Bibliothek der Technischen Hochschule Wildau befindet und weitere bibliographische Informationen zu ermitteln. Abschließend würden auch hier wieder die verschiedenen APIs zur Ermittlung der Cover abgefragt werden.

Im Bereich des Dashboards sollten dabei Literaturempfehlungen zum aktuellen Semester, in dem der Student sich derzeit befindet, angezeigt werden. Dafür ist es notwendig, dieses zunächst zu ermitteln.

Darüber hinaus soll es möglich sein, zum eigenen Studiengang alle Literaturempfehlungen (unabhängig vom aktuellen Semester) anzeigen zu lassen. Dazu soll es einen eigenen Punkt in der Navigationsleiste geben.

Nach Auswahl des aktuellen Semesters werden, sortiert nach Modul, die ermittelten Literaturhinweise aufgelistet:

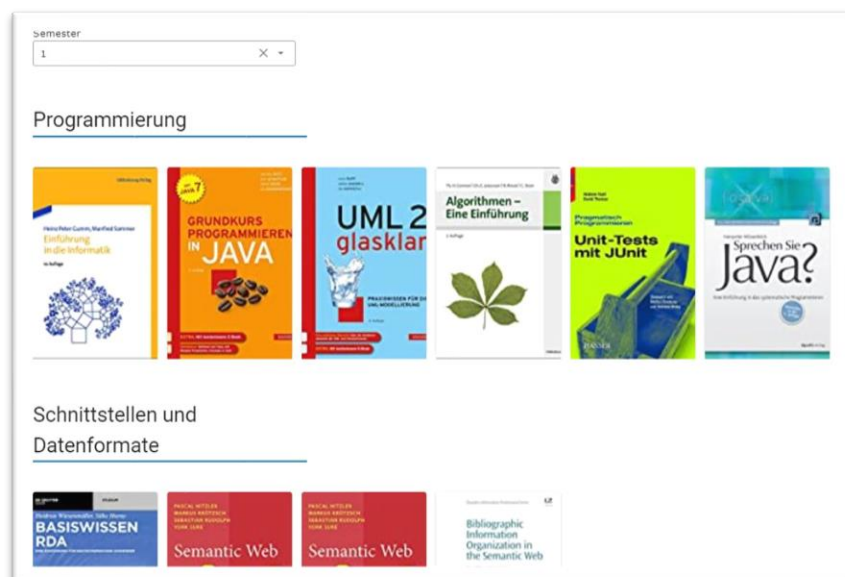


Abbildung 27: Mockup zu Literaturempfehlungen zum eigenen Studiengang

<sup>197</sup> Vgl. Attia (2020a): Entwicklung eines Recommender-Systems für E-Book-Empfehlungen in der Hochschulbibliothek der TH Wildau, S. 33.



## 6.12.2 Anzeigen von Empfehlungen für andere Studiengänge

Damit sich Studenten auch über Literaturempfehlungen anderer Studiengänge informieren können, soll darüber hinaus die Möglichkeit bestehen, sich Literaturhinweise anderer Studiengänge anzeigen zu lassen. Die Auswahl eines bestimmten Studiengangs könnte dabei mit Hilfe eines Dropdown-Menüs umgesetzt werden. Die einzelnen Studiengänge müssten dafür im System hinterlegt werden. Für die Ermittlung der Module, die pro Semester angeboten werden, könnte wiederum die Modulhandbuch-API genutzt werden.

## 6.12.3 Senden von E-Mail-Empfehlungen

Nach Anmeldung in der App soll zudem die Möglichkeit bestehen, E-Mail-Empfehlungen an Kommilitonen zu versenden. Die Funktion könnte in der Detailansicht untergebracht werden. Die Anzeige sollte nur erfolgen, wenn sich der Nutzer authentifiziert hat. Die Eingabe der gewünschten Zieladresse könnte über ein Popup-Fenster erfolgen, welches sich nach Anklicken des Buttons öffnet (siehe Abb. 28).

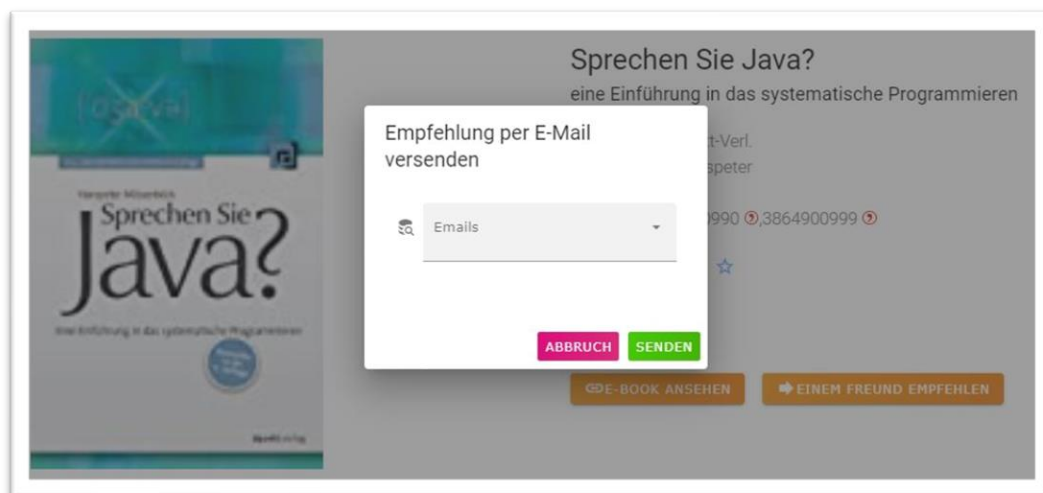


Abbildung 28: Mockup für Versenden einer E-Mail-Empfehlung

Alternativ kann der Schlitz für die Eingabe der Zieladresse auch direkt in die Seite eingebunden werden:

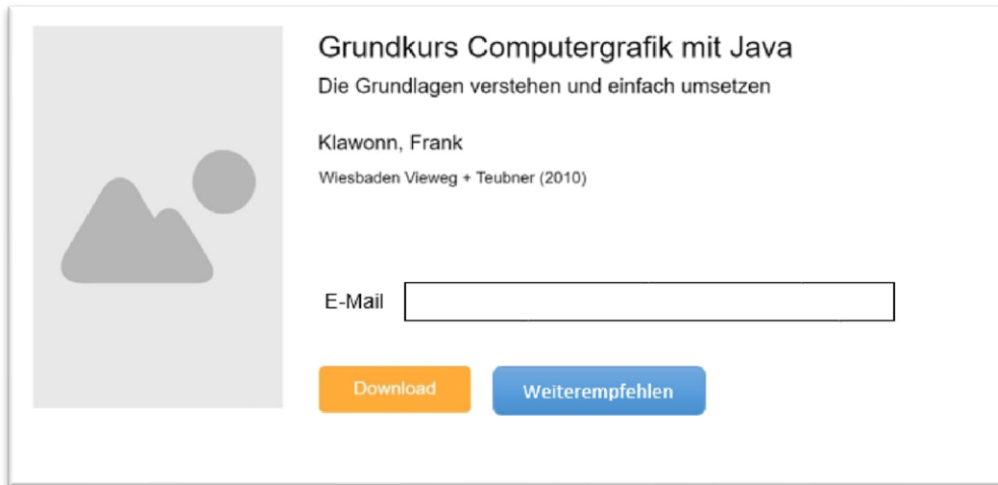


Abbildung 29: Mockup für Versenden einer Buchempfehlung per E-Mail  
 Quelle: Attia (2020a), S. 37 [Abbildung 5.5.1: Mockup für Empfehlung eines Buchs]

Darüber könnte ggf. eine Autovervollständigung von E-Mail-Adressen angeboten werden, sofern eine geeignete Schnittstelle zur Verfügung steht. Ideal wäre eine Auto-Vervollständigung ab dem dritten Buchstaben, da die Trefferliste andernfalls zu lang werden würde (siehe Abb. 30).



Abbildung 30: Mockup für Autovervollständigung

Für die technische Umsetzung könnte die Node.js-Bibliothek Nodemailer genutzt werden, welches das Versenden von E-Mails ermöglicht, ohne einen zusätzlichen Mailserver einrichten zu müssen.<sup>198</sup>

Auch für die Darstellung der Inhalte in der E-Mail gibt es verschiedene Möglichkeiten. Neben einem allgemeinen kurzen Hinweistext, woher die Empfehlung kommt und was es damit auf sich hat, gibt es zur Präsentation der bibliographischen Daten verschiedene Möglichkeiten:

<sup>198</sup> Vgl. Reinman (n. N.): Nodemailer, <https://nodemailer.com/about/>, 27.07.2021.

Einerseits ist es möglich, diese in einer tabellarischen Übersicht aufzulisten. Darüber hinaus besteht jedoch auch die Möglichkeit, neben dem allgemeinen Hinweistext lediglich einen Link auf die Empfehlung zu nutzen. Diese würde dann innerhalb der App angezeigt werden.

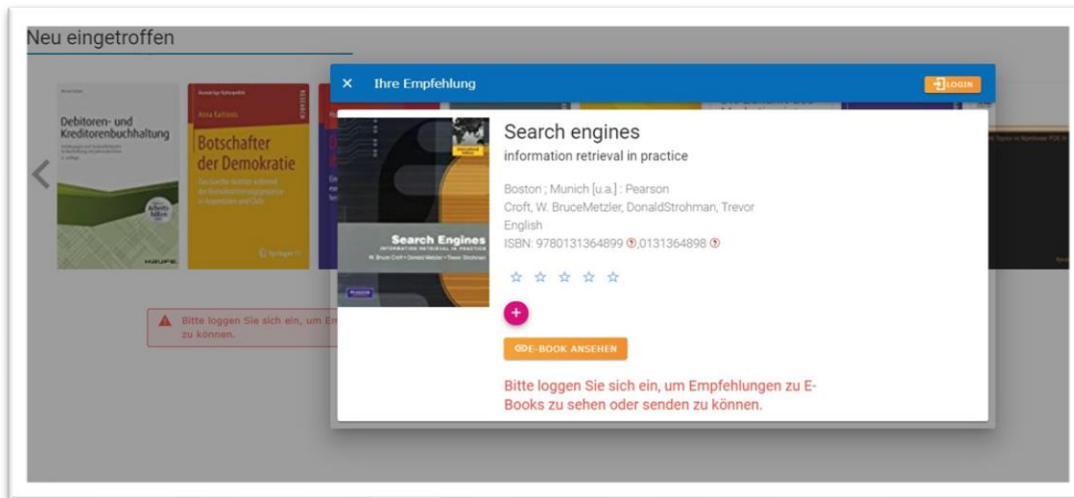


Abbildung 31: Mockup für Anzeige der Empfehlung aus E-Mail in der App selbst

Vorteil der zweiten Variante wäre, dass dadurch gleich auf die App selbst verwiesen wird und dadurch diese ggf. einen höheren Bekanntheitsgrad erreicht. Evtl wird der Nutzer neugierig und schaut sich weitere Funktionen der Anwendung an und wird somit zur Nutzung derer angeregt.

#### 6.12.4 Einzelne Medien bewerten – einige Ideenskizzen

Um Buchempfehlungen nicht nur einem einzelnen Kommilitonen per E-Mail sondern ggf. mehreren Studenten zeitgleich zugänglich zu machen und diese langfristig zur Verfügung zu stellen, wäre es darüber hinaus denkbar, einzelne Medien durch die Nutzer bewerten zu lassen. Bewertungssysteme finden sich heute in zahlreichen Online-Shops wie Amazon oder Thalia.<sup>199</sup>

Hierbei sind verschiedene Bewertungskriterien denkbar. Einerseits ist es möglich, dass Studenten Literatur in Abhängigkeit zu ihrem Studiengang bzw. ausgewählten Modulen bewerten könnten. Diese Informationen könnten dann bspw. zusätzlich zu den Literaturempfehlungen der Dozenten aus dem Modulhandbuch herangezogen werden.

Vorstellen könnte man sich dabei eine Bewertung in Form von ein bis fünf Sternen, wie bei Amazon.de oder thalia.de<sup>200</sup>, und zusätzlich zwei Dropdownlisten aus denen ein oder mehrere Studiengänge und ein oder mehrere Module ausgewählt werden können, für die die Bewertung gelten sollen. Um dies zu erweitern, könnte ggf. zusätzlich ein Bemerkungsfeld angeboten werden.

Damit würde man Literaturempfehlungen erhalten, die die Eignung eines Mediums in Bezug auf einen bestimmten Studiengang bzw. sogar ein bestimmtes Modul abbilden könnten.

<sup>199</sup> Siehe z.B. amazon.de oder thalia.de

<sup>200</sup> Bei thalia.de werden anstelle von Sternen fünf entsprechend gestaltete Smileys zur Abgabe der Bewertung angeboten.

Zu bedenken dabei ist jedoch, dass die Informationen in einer entsprechenden Datenbank hinterlegt werden müssen, welche gut skalierbar ist. Hierfür würde sich bspw. die NoSQL-Datenbank MongoDB eignen, derzeit eine der beliebtesten Datenbanken.<sup>201</sup> Die Bewertungen könnten dabei mit folgenden Informationen hinterlegt werden:

- Anzahl Sterne
- Studiengang/Studiengänge
- Ggf. Modul/e
- Ggf. Bemerkungen

Darüber hinaus ist zu überlegen, wie verhindert werden kann, dass ein Student/Dozent mehrere Bewertungen abgibt.

Um Informationen zur Precision der gefundenen Medien anhand des Regalstandortes zu erhalten, könnte man ggf. auch abfragen, inwiefern die gefundenen Treffer nach Scannen des QR-Codes thematisch zum Regalstandort gepasst haben. Dies würde eines expliziten Relevance-Feedbacks entsprechen. Hier könnte man ebenfalls Bewertungssysteme von „nützlich“/„nicht nützlich“ bis hin zu n Sternen verwenden. Diese Funktion könnte auch Nutzern bereitgestellt werden, die sich nicht angemeldet haben. Wie bereits Attia schreibt, ist allerdings sehr fraglich, inwieweit die Möglichkeit tatsächlich genutzt wird.<sup>202</sup> Die Daten wären jedoch sehr interessant für die Weiterentwicklung der Ermittlung der zum Regalstandort passenden E-Ressourcen.

Zu bedenken bei der Implementation entsprechender Komponenten sind darüber hinaus die von der Anwendung für die Bereitstellung eines entsprechenden Recommender-Systems benötigten Ressourcen:

*„Einige dieser Systeme benötigen einen riesigen Datenspeicherplatz, und ihre Prozesse bzw. Algorithmen sind sehr kompliziert, was die Gesamtarbeitslast der Anwendung erhöht.“<sup>203</sup>*

## 6.13 Direkter Zugang/Durchschaltung zu E-Books

Als weiterer Service für die Nutzer der App soll der Zugang zu den Volltexten der E-Books erleichtert werden. Im aktuellen Prototyp ist analog zu Wilbert eine Verlinkung zum Volltext über die Anbieterseite umgesetzt. Um zum Volltext-PDF zu gelangen, muss dies nochmal separat geöffnet werden. Als Service für die Nutzer der App soll bei Öffnen des Volltext-Links (sofern ein Volltext-PDF angeboten wird) dies direkt im Browser geöffnet und der Umweg über die Anbieterseite vermieden werden.<sup>204</sup>

Voraussetzung für die Bereitstellung dieser Funktion ist zunächst, dass die Anbieter der E-Books ihre Volltexte im PDF-Format bereitstellen. Nur dann ist es möglich, dieses direkt im Browser zu öffnen.

---

<sup>201</sup> Vgl. Hill (2021): Was die NoSQL-Datenbank kann : MongoDB FAQ, <https://www.computerwoche.de/a/was-die-nosql-datenbank-kann,3548993>, 28.07.2021.

<sup>202</sup> Vgl. Attia (2020a), S. 19.

<sup>203</sup> Vgl. Attia (2020a): Entwicklung eines Recommender-Systems für E-Book-Empfehlungen in der Hochschulbibliothek der TH Wildau, S. 20.

<sup>204</sup> Da die Anwendung der zu entwickelnden App auf die Nutzung in der Bibliothek zielt und davon ausgegangen wird, dass die Nutzer sich im Hochschulnetz befinden, soll an dieser Stelle nicht weiter auf einzelne Authentifizierungsmethoden für Online-Angebote wie Shibboleth eingegangen werden.

Darüber hinaus müssen die URLs zu den PDF-Dateien aus den Anbieterseiten in einem festen Muster ableitbar sein. Aus der Anbieter-URL zu einem DeGruyter-Dokument

*<https://www.degruyter.com/document/doi/10.1524/9783486594058/html>*

ließe sich bspw. die URL zum PDF nach einem bestimmten Muster ableiten:

*<https://www.degruyter.com/document/doi/10.1524/9783486594058/pdf>*

An dieser Stelle wäre eine Ersetzung der URL problemlos möglich, der letzte Teil des URL-Pfades würde einfach von *html* zu *pdf* ersetzt werden können.

Weiterhin ist zu prüfen, ob die Anbieter die Volltexte in einem PDF-Dokument oder kapitelweise bereitstellen. Werden nur einzelne Kapitel zum Download angeboten, kann nicht gezielt auf ein bestimmtes Kapitel gelinkt werden. An dieser Stelle ist der Umweg über die Anbieterseite nicht zu umgehen.

Da die Anbieter ggf. die Speicherorte der Volltexte ändern ist darüber hinaus stets zu prüfen, ob die erstellte URL tatsächlich zugänglich ist. Ist dies nicht der Fall, sollte der Nutzer alternativ auf die Anbieterseite geleitet werden.

## 6.14 Integration von einfacher Suchfunktion

Zusätzlich zur Ermittlung der zum Standort thematisch passenden Ressourcen soll eine minimale Suchfunktion zur Verfügung stehen, mit der einfache thematische Suchen durchgeführt werden können. Zielstellung ist hier allerdings nicht, sämtliche Funktionalitäten von Wilbert nachzubilden. Fokus der App liegt schließlich auf der Ermittlung der zum Regalstandort passenden Medien. Die freie Suche ist quasi ein *nice-to-have*. Für spezifische Recherchen kann ggf. eine Verlinkung zum Discovery-System eingebunden werden. Eine entsprechende Anfrage könnte als Freitextsuche durchgeführt werden. Die zugehörige Anfrage der Solr-API für eine Suche nach dem Thema *JavaScript* sieht folgendermaßen aus:

*[https://solrproxy-wilbert-test.koby.de/wilbertselect?access=XXXXXXXXXX&fq=datenlieferant:fhw\\_ebooks&q=javascript](https://solrproxy-wilbert-test.koby.de/wilbertselect?access=XXXXXXXXXX&fq=datenlieferant:fhw_ebooks&q=javascript)*

Dabei wäre auch eine Übergabe mehrerer Terme möglich.

## 6.15 Datenschutz

Da mit personenbezogenen Daten gearbeitet wird, ist es wichtig, dass vor allem das Prinzip der Datensparsamkeit beachtet wird. Zudem muss der Nutzer darüber informiert werden, welche Daten zu welchem Zweck und für welchen Zeitraum in der Anwendung verarbeitet und gespeichert werden (siehe Kapitel 14 EU DSGVO).<sup>205</sup>

Die entsprechenden Informationen können bspw. im Impressum der Anwendung untergebracht werden. Ggf. kann zu diesem Zweck im Anmeldefenster eine Checkbox implementiert werden, die auf die

---

<sup>205</sup> Vgl. dejure.org Rechtsinformationssysteme GmbH (n. N.): Datenschutz-Grundverordnung : Artikel 14 Informationspflicht, wenn die personenbezogenen Daten nicht bei der betroffenen Person erhoben wurden, <https://dejure.org/gesetze/DSGVO/14.html>, 28.07.2021.

Verarbeitung personenbezogener Daten nach Anmeldung hinweist. Ist diese Checkbox nicht angehakt, ist die Anmeldung in der App nicht möglich.

Darüber hinaus ist eine Absprache mit dem entsprechenden Datenschutzbeauftragten notwendig.

## **6.16 Ideen für die Benennung der Anwendung**

Derzeit wird für den Prototyp der Name „E-Books To Go“ verwendet. Sollte es zur Integration weiterer Ressourcen wie E-Journals kommen, ist dieser Name inhaltlich nicht mehr korrekt. Auch die Integration der entliehenen Printbände führt dazu, dass der Name nicht mehr treffend ist. Alternativvorschläge könnten evtl über eine Umfrage an der Hochschule ermittelt werden. Vorschläge könnten sein „Bibshelf Expander – more than you see“ oder ganz kurz: „Bibshelf+“.

## 7 Umsetzung ausgewählter Aspekte

Nachdem im vorherigen Kapitel das Konzept erarbeitet wurde, soll nun auf die Umsetzung einzelner Funktionen eingegangen werden.

Die Umsetzung der Funktionalitäten erfolgt aufbauend auf dem bereits bestehenden Prototypen „E-Books To Go“. Zur Einarbeitung in die bestehenden Skripte fand große Unterstützung durch Christopher Grätz und Omar Attia statt. Es sollen vor allem die Anpassungen und Erweiterungen beschrieben werden.<sup>206</sup> Um die Gesamtfunktionalität zu beschreiben, wird zum Teil auf die Umsetzung des bereits bestehenden Prototyp zurückgegriffen.

Zunächst soll dabei auf die Einbindung der Solr-API eingegangen werden, welche genutzt werden soll, da im derzeit genutzten HTML die Daten bereits für Wilbert aufbereitet wurden.

Anschließend geht es darum, die QR-Codes mit geeigneten Inhalten zu füllen, um wie im Konzept beschrieben, mit den Medien „mitwandernde“ QR-Codes erstellen zu können. Darüber hinaus soll eine Möglichkeit der möglichst einfachen Erstellung dieser gefunden werden.

Zusätzlich soll beschrieben werden, wie die „Durchschaltung“ zum E-Book-Volltext (ohne Umleitung über die Anbieterseite) erreicht werden kann.

Als neue Funktionalität der Anwendung sollen zudem zusätzlich zu den E-Books auch entliehene Print-Bücher angezeigt werden, welche aus der App heraus (nach Anmeldung) vorgemerkt werden können.

Für die Umsetzung des Projekts wurde mit der Integrated Development Environment (IDE) Visual Studio Code (VS Code) gearbeitet. Diese bietet u.a. Unterstützung in Form einer für Vue.js zugeschnittenen Autovervollständigung und Code-Highlighting sowie weitere Hilfswerkzeuge zur Entwicklung. Darüber hinaus wurde vor allem für die Frontend-Entwicklung die Vue.js-Extension für Chrome intensiv genutzt.<sup>207</sup>

### 7.1 Vom Wilbert-HTML zum Solr-API-XML

Da für die Entwicklung der neuen Funktionalität der Durchschaltung zu den Volltexten der E-Books Daten benötigt werden, die über die HTML-Antwort von Wilbert nicht ermittelt werden können, fiel schnell die Entscheidung, dass der Umstieg auf die Solr-API als Datenquelle notwendig ist. Außerdem finden sich hier die Originaldaten aus dem Index des Bibliotheksmanagementsystems.

Bei der Einarbeitung in die Solr-API und die Umsetzung der bisher an Wilbert erfolgten Anfragen stand neben Herr Kissig vor allem auch Herr Kant vom KOBV für Rückfragen zur Verfügung.

Die entsprechenden Abfragen der Solr-API geschehen im Backend durch den Node-Server. Nach Aufruf des entsprechenden Endpunktes durch das Frontend bspw. für die Ermittlung der neusten E-Books wird die Funktion *getInfo()* aufgerufen, welcher u.a. die zuvor erstellte Abfrage-URL für die Solr-API übergeben wird. Diese wiederum wandelt die Daten mit Hilfe der Funktion *parseStringPromise()* aus dem npm-

---

<sup>206</sup> Die Beschreibung des Ist-Standes der App findet sich in Kapitel 4.3

<sup>207</sup> Siehe <https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnanhbledajbpd>

Paket `xml2js`, welches vorher importiert wurde, zunächst vom XML- in das JSON-Format um, um die für die Anwendung benötigten Daten anschließend einfacher parsen zu können (siehe Listing 7).

```
/**
 * Get the XML-Content from a Solr-API-Request
 *
 * @param {any} response
 * @param {any} url the url to Solr-API to connect and get XML-Data
 */
var getInfo = async function getInformation(response, url) {
  console.log('call getInformation');

  axios.get(url, {
    /*
    proxy: {
      host: 'proxy.th-wildau.de',
      port: 8080
    }
    */
  }).then((res) => {
    //console.log(res.data);
    // response.send(res.data);
    parser.parseStringPromise(res.data, {mergeAttrs: true})
      .then(result => {
        // console.log(result);
        parseData(result)
          .then(result => response.send(result))
          .catch ( (error) => {
            console.log('ERROR');
            console.log(error.message);
          })
      })
      .catch ((error) => {
        console.log('ERROR');
        console.log(error.message);
      })
      //response.send(result)
  }).catch(function (error) {
    console.log('ERROR');
    console.log(error.message);
  })
};
```

Listing 5: Funktion `getInfo()` zur Ermittlung von E-Ressourcen

In der Funktion `parseData()` werden schließlich die für die Anwendung benötigten Metadaten zu den E-Books ermittelt. Dabei werden zunächst einzelne JSON-(E-Book-)Objekte erzeugt. Anschließend werden diese in Form eines Arrays an das Vue.js-Frontend zurückgegeben (siehe Listing 8).



```

records.forEach(record => {
  i += 1;
  let eBook = {
    title : '',
    year : '',
    publisher : '',
    author : '',
    language : '',
    isbn : '',
    hashtags : '',
    cover : '',
    pagesCount : '',
    url_ebook : '',
    key: ''
  }
  let bookArr = record.arr;
  bookArr.forEach(element => {
    var key = element.name[0];
    switch (key) {
      case 'fulltext_url':
        eBook.url_ebook = element.str[0];
        //console.log("URL");
        break;
      case 'book_doi':
        eBook.doi = element.str[0];
        break;
      case 'person':
        eBook.author = element.str.join('; ');
        break;
      case 'isbn':
        eBook.isbn = element.str.join(', ');
        break;
      case 'titel':
        eBook.title = element.str[0];
        break;
      case 'publikationsjahr_anzeige':
        eBook.year = element.str[0];
        break;
      case 'publisher':
        eBook.publisher = element.str[0];
        break;
      case 'language':
        eBook.language = element.str[0];
        break;
      case 'schlagwort':
        eBook.hashtags = element.str.join(';');
        break;
      case 'seiten':
        eBook.pagesCount = element.str[0];
        //console.log("Seitenanzahl");
        break;
      case 'identnr':
        eBook.key = element.str[0];
    }
  })
  //console.log(eBook);
  allRecords.push(eBook);
  //console.log(allRecords);
})
resolve(allRecords);
reject('ERROR') }
})
}

```

*Listing 6: Erstellen der E-Book-Objekte und Mapping mit Daten aus Solr-API*

## 7.2 Auf den QR-Codes zu hinterlegende Informationen

Wie im Konzept beschrieben, sollen an den Regalen „mitwandernde“ QR-Codes befestigt werden, um den Aufwand einer dauerhaften Aktualisierung der sich in einzelnen Regalen oder Regalebene befindenden Signaturen zu vermeiden. Gerade durch das Umsignier-Projekt, welches derzeit in der Bibliothek der TH Wildau stattfindet, würde dadurch ein ständiger Pflegeaufwand entstehen.

Im Rahmen der Masterarbeit soll exemplarisch für den Bereich Informatik (RVK-Notationsbereich SQ – SU) ein Entwurf für die benötigten QR-Codes entwickelt werden.<sup>208</sup>

Zu diesem Zweck wurde sich zunächst der Ist-Stand der Verteilung der einzelnen RVK-Notationen im Regal angeschaut.<sup>209</sup> Es stellte sich heraus, dass einzelne Notationen (z.B. *ST 250*, *ST 253* oder *ST 261*) über mehrere Regalböden und sogar Regale verteilt sind. Im Bereich der Informatik werden zusätzlich zur RVK-Notation zum Teil die Namen von Programmen als Kurz-Cutterung zur weiteren thematische Untergliederung genutzt. In der TH Wildau wird dies bspw. innerhalb der Notationen *ST 250* und *ST 261* genutzt. Für diese Bereiche wurde anschließend betrachtet, inwieweit nochmals eine thematische Untergliederung mittels Kurz-Cutterung für einzelne Programmnamen erfolgt ist. War dies der Fall, so wurden diese mit in die Übersicht aufgenommen.

Im Gegensatz dazu kommt es jedoch auch vor, dass für einzelne RVK-Notationen nur ein oder zwei Bände im Regal stehen (z.B. *ST 155* oder *ST 257*). Gerade an diesen Stellen ist anhand der Trefferzahlen der Solr-API zu prüfen, ob einzelne QR-Codes pro RVK-Notation hier angebracht sind oder ob stattdessen ein QR-Code für mehrere Notationen genutzt werden kann. Hierbei galt es jedoch zu beachten, dass die Notationen inhaltlich noch grob zusammenpassen. Als Ansatz wurde hier versucht, keine RVK-Notationen über numerische Zehnerbereiche hinaus zusammenzufassen. So wurden bspw. für die Notationen *ST 630* und *ST 640* separate QR-Codes erstellt, anstatt einen QR-Code mit beiden Notationen zu erstellen.

Anschließend wurden die Trefferzahlen der Solr-API ermittelt, um herauszufinden, wie viele E-Books bei Scannen der einzelnen QR-Codes erwarten. Fanden sich für einige Notationen nur Trefferzahlen im einstelligen Bereich (z.B. *ST 125*: 1 Treffer oder *ST 136*: 8 Treffer), so fanden sich für andere Systemstellen wiederum sehr große Trefferzahlen (z.B. *ST 230*: 135 Treffer oder *ST 610*: 85 Treffer). Hier galt es zu überlegen, ob diese hohen Trefferzahlen über einen QR-Code erreichbar sein sollten oder ob hier nochmal eine Trennung stattfinden sollte und wenn ja auf welche Weise.

Da die Erfahrung großer Suchmaschinen zeigt, dass hier nur die erste Seite, welche zehn Treffer anzeigt, angeschaut wird und auf dieser meist nur die ersten drei Treffer, wurde sich dafür entschieden, ab einer bestimmten Trefferzahl einen zweiten QR-Code für eine Notation zu erstellen. Hier wurde geprüft, welche Trefferstaffelung Wilbert anbietet: Hier lassen sich 5, 10, 20, 50 oder sogar 100 Treffer pro Seite anzeigen. Würde man im Falle der QR-Codes lediglich 20 Treffer anzeigen lassen, bestände die Gefahr, dass zu viele QR-Codes an den Regalen hängen. Da man für die Anzeige von 50 Treffern auf einem Smartphone doch schon eine ganze Weile scrollen müsste, wurde sich für das Mittelmaß von 40 Treffern entschieden. Dazu musste nun jedoch bestimmt werden, auf welcher Grundlage die Trennung geschehen sollte.

---

<sup>208</sup> Die (Excel-)Tabellen, die im Rahmen der Erarbeitung der Inhalte der QR-Codes erstellt wurden, finden sich auf der beiliegenden CD-ROM

<sup>209</sup> Siehe Anhang A.4

Grundsätzlich sollten die Treffer wie auch am Regal alphabetisch nach Verfassernamen sortiert werden. Zunächst wurde versucht, die Treffer auf Grundlage des ersten Buchstabens des Nachnamens des Verfassers einzuschränken, welcher sich im Solr-Feld *autor* befindet:

```
...q=rvk:"ST%20230"&fq=autor:%5b"N*"%20TO%20"Z*"%5d&sort=autor_sort%20asc...
```

Dies führte leider nicht zum Erfolg. In den Ergebnissen fanden sich Verfasser mit Anfangsbuchstaben von *A – M*. Anschließend wurde versucht, die Treffer anhand der Einträge im Feld *autor\_sort* einzugrenzen:

```
...q=rvk:"ST%20230"&fq=autor_sort:%5b"N*"%20TO%20"Z*"%5d&sort=autor_sort%20asc...
```

Hierbei fiel jedoch auf, dass die Treffermenge von zwei Anfragen nach Eingrenzung der Verfasser-Anfangsbuchstaben (bspw. von *A – M* und *N – Z*) nicht die gleiche Trefferzahl lieferten wie eine Anfrage ohne die Eingrenzung: Insgesamt (ohne Eingrenzung) liefert die Anfrage nach der RVK-Notation *ST 130* 50 Treffer. Grenzt man auf die Buchstaben *A – M* ein, so erhält man 14 Treffer.<sup>210</sup> Auch für die zweite Hälfte des Alphabets werden lediglich 14 Treffer geliefert.<sup>211</sup>

Ursache dafür ist vermutlich, dass die entsprechenden Einträge aufgrund der bereits beschriebenen Übernahme der bibliographischen Metadaten von den E-Book-Anbietern nicht für alle Datensätze ausgefüllt sind. Diese würden im Falle einer solchen Suche aus der Treffermenge herausfallen und könnten den Nutzern nicht präsentiert werden.

Als weitere Möglichkeit bietet es sich an, eine Trennung der Treffer anhand der Trefferzahlen vorzunehmen. Durch die alphabetische Sortierung könnten so 70 Treffer zum Beispiel zweigeteilt werden, in dem auf dem ersten QR-Code die Treffer von 1 – 35 und auf dem zweiten alle Treffer ab 36 angezeigt werden könnten. Eine entsprechende Anfrage der Solr-API sieht folgendermaßen aus:

Treffer 1 – 35: ...q=rvk:%22ST%20130%22&rows=35...<sup>212</sup>

Treffer 36 – Ende: q=rvk:%22ST%20130%22&start=35...<sup>213</sup>

Anschließend wurde auch hinsichtlich der Trefferzahlen geschaut, ob bestimmte RVK-Notationen zusammengefasst werden können.

Nicht zuletzt musste geprüft werden, ob tatsächlich alle möglichen RVK-Notationen über die QR-Codes abgebildet wurden, da es vorkommen kann, dass für einzelne Notationen keine Printbooks im Regal stehen, aber E-Books vorhanden sind. Schließlich sollten auch diese über die QR-Codes abgebildet werden.

---

<sup>210</sup> Vgl.: Die entsprechende Anfrage lautete:

```
„...fq=datenlieferant:fhw_ebooks&index=internal&q=rvk:%22ST%20130%22&fq=autor_sort:%5b%22A*%22%20TO%20%22N*"%22%5d&sort=autor_sort%20asc“ *Bei Einer Anfrage von A – M waren die Treffer mit Verfasser-Anfangsbuchstaben M nicht inbegriffen, deshalb musste auf N erweitert werden
```

<sup>211</sup> Vgl.: Die entsprechende Anfrage lautete:

```
„...fq=datenlieferant:fhw_ebooks&index=internal&q=rvk:%22ST%20130%22&fq=autor_sort:%5b%22N*"%22%20TO%20%22Z*"%22%5d&sort=autor_sort%20asc“
```

<sup>212</sup> Der Parameter „start“ kann hier entfallen, da automatisch mit dem ersten Treffer begonnen werden kann.

<sup>213</sup> Der Parameter „rows“ kann hier entfallen, wenn alle restlichen Treffer angezeigt werden sollen. Begonnen wird hier mit 35, da die Zählung mit 0 beginnt.

Es zeichnete sich ab, dass es vorkommen kann, dass sich für Notationen mit nur sehr wenigen im Regal stehenden Bänden trotzdem sehr große Treffermengen an E-Books ermitteln ließen. So kann es sein, dass an dieser Stelle trotzdem mehrere QR-Codes am Regal befestigt werden müssten, hier kann ggf. noch ein Hinweis auf die Verfasserbereiche unter dem QR-Code angebracht werden. Sollte es also u.a. aufgrund der großen Solr-API-Trefferzahlen doch dazu kommen, dass mehrere QR-Codes an einem Regalboden befestigt werden müssen, so könnten diese ggf. alle nebeneinander am rechten oder linken Rand mit einer entsprechenden Bezeichnung angebracht werden. Dies kann auch der Fall sein, wenn Bände zu verschiedenen RVK-Notationen auf einem Regalboden stehen, die auf verschiedene QR-Codes abgebildet wurden.

Für RVK-Bereiche mit Cutterung des Programmnamens wurde für einzelne Bereiche mit erheblichen Trefferzahlen (z.B. *ST 250 J35* -> Java, *ST 250 C01* => C, *ST 601 S69* -> SPSS) zunächst versucht, eine Suche anhand der genauen Notation (inklusive Cutterung) umzusetzen. Hier wurden jedoch nur sehr geringe Treffermengen erreicht. So wurde die Entscheidung getroffen, hier eine Suche mit Hilfe von Schlagwörtern durchzuführen. Da eine alphabetische Abbildung des Bestandes für Notationen, die sich auf einzelne Programme oder Programmiersprachen wenig sinnvoll ist, soll hier stattdessen am Regal ein Hinweis auf die freie Suche innerhalb der App angebracht werden. Nur Unterbereiche mit größeren Treffermengen sollen mit einem separaten QR-Code versehen werden.

Es zeigt sich also, dass eine Entscheidung getroffen werden muss, zwischen den pro Regalbereich beinhalteten RVK-Notationsbereichen durch Anbringung der QR-Codes in regelmäßigen Abständen und andererseits der Größe der hinter den QR-Codes stehenden Treffermengen. Diese sollten nicht zu groß gewählt werden, da davon ausgegangen wird, dass bei zu großen Treffermengen nicht mehr alle Treffer angeschaut werden. Zu bedenken dabei ist außerdem, dass an der Bibliothek der TH Wildau derzeit nur ca 30% der E-Books mit RVK-Notationen versehen sind. Mit zunehmender Zahl wird sich auch die Trefferzahl an E-Books pro Notation erhöhen, was die Problematik verschärfen wird.

### 7.3 Erstellung von QR-Code-Listen

Nachdem alle für die Absetzung entsprechender Abfragen an die Solr-API notwendigen Informationen für die Ermittlung zum Standort thematisch passender E-Books ermittelt wurden, konnte überlegt werden, wie diese in geeigneter Form auf dem QR-Code untergebracht werden konnten. Darüber hinaus war es wichtig, anhand einer vorgegebenen Liste eine Vielzahl von QR-Codes erzeugen zu können.

Betrachtet man die für die verschiedensten Abfragen benötigten Informationen, so sind dies:

- Abfrage anhand RVK-Notation oder Schlagwort
- RVK-Notation (ggf. von ... bis ...)
- ggf. Schlagwort
- Treffer von ...
- Treffer bis ...

Zunächst galt es, diese in einer Art und Weise auf dem QR-Code unterzubringen, dass das Backend diese entsprechend auswerten und die Solr-API-Anfragen erstellen und ausführen kann.

Die wichtigste Information ist, ob die Abfrage anhand RVK-Notation oder mit Hilfe eines Schlagworts erfolgen soll, denn darauf bauen alle weiteren Verzweigungen auf. An zweiter Stelle kann so bspw. die

RVK-Notation stehen. Für den Fall, dass die Abfrage der E-Books mit Hilfe von Schlagwörtern erfolgen soll, ist diese Angabe fakultativ. Spätestens für die Ermittlung der entliehenen Bände wird diese Information jedoch benötigt, sodass sie auf dem QR-Code enthalten sein muss. Für den Fall, dass die Suche mit Hilfe eines Schlagwortes erfolgen soll, muss dieses darüber hinaus angegeben werden. Um eine Eingrenzung der Treffermengen zu erreichen, soll zusätzlich angegeben werden, welcher Trefferbereich abgefragt werden soll (Treffer von, Treffer bis).

Für die App sollte der Aufbau des Inhaltes der QR-Codes folgendermaßen aussehen:

*Abfragetyp (RVK oder SW)/RVK-Notation (ggf. von ... bis ...)|Treffer von ...|Treffer bis ...|Schlagwort*

Diese Informationen könnten bspw. in einer Excel-Tabelle gesammelt werden und anschließend in einer Spalte automatisch zusammengeführt werden. Da das Office-Paket (inklusive Excel) in Bibliotheken meist als Standardsoftware vorhanden ist, wird hierfür keine zusätzliche Software benötigt. Die Datenpflege ist darüber hinaus einfach, da Excel zu den Standard-Office-Programmen gehört. Ein entsprechendes Template befindet sich auf der CD-ROM.<sup>214</sup>

Am Backend-Endpunkt `/booksqrcode/:qrCodeInfo` findet dann die entsprechende Auswertung der auf den Codes hinterlegten Informationen statt. Je nach gefundenen Informationen wird schließlich die Solr-API-Abfrage zusammengebaut (siehe Listing 9).

```
// Search for E-Books from QR-Code matching to user position
app.get('/booksqrcode/:qrCodeInfo', async function (req, res) {
  // console.log('\x1b[33m%s\x1b[0m', '-----');
  ----');
  console.log('got QR-Code information');
  let codeInfos = req.params.qrCodeInfo.replace(' ', '%20');
  // console.log(codeInfos);
  let infoArr = codeInfos.split('|');
  let localUrl = qrCodeUrl;
  let infotype = infoArr[0]; // either rvk or schlagwort
  let rvkNotation = infoArr[1];
  let startNr = infoArr[2];
  let nrRows = infoArr[3];
  let hashtag = infoArr[4];
  console.log('RVK-Notation: ' + rvkNotation);
  console.log('Hashtag: ' + hashtag);
  console.log('Treffer von: ' + startNr);
  console.log('Treffer bis: ' + nrRows);
  // searching for rvk-notation
  if (infotype == 'RVK') {
    localUrl += 'rvk:%22' + rvkNotation + '%22';
    // if there are more than 40 hits, there is another
    // qr-code, which returns more hits
    // startnr = 40,80, ...
  // searching for hashtag
  } else if (infotype == 'SW') {
    localUrl += 'schlagwort:' + hashtag;
    // if there are more than 40 hits, there is another
    // qr-code, which returns more hits
    // startnr = 40,80, ...
  }
```

<sup>214</sup> Siehe Datei *QR-Codes\_Template.xlsx*

```

    }
    if (startNr != "") {
        localUrl += '&start=' + startNr;
    }
    if (nrRows != "") {
        localUrl += '&rows=' + nrRows;
    } else {
        localUrl += '&rows=40'
    }
    // sorting results by author and return 40 results
    localUrl += '&sort=autor_sort%20asc';
    console.log(localUrl);
    tools_solr.getInfo(res, localUrl);
    // res.send(localUrl);
})

```

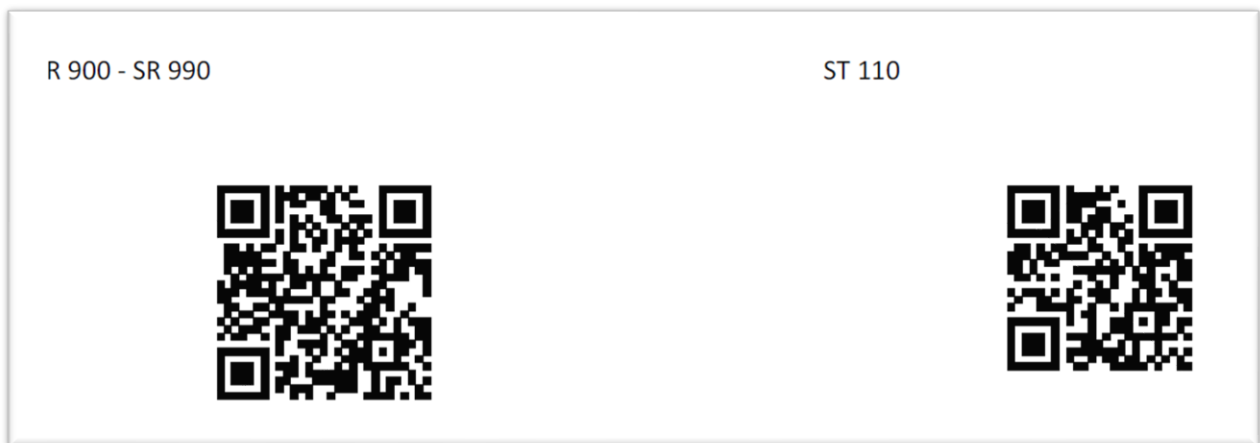
*Listing 7: Backend-Endpoint zur Erstellung und Ausführung der Solr-API-Abfrage*

Anschließend ging es darum, ein Tool zu finden, mit dem aus einer Liste von Textinformationen QR-Codes erstellt werden können. Kostenfreie Tools wie qrcodemonkey<sup>215</sup> oder QR Code Generator<sup>216</sup> bieten zwar die Möglichkeit, ohne Anmeldung einzelne QR-Codes zu erzeugen. Für die Massenerzeugung muss man sich jedoch ein Nutzerkonto erstellen.

Auch Word bietet die Möglichkeit, mit Hilfe der Serienbrief-Funktion QR-Codes zu erstellen. Dafür kann folgende Funktion verwendet werden:

*{ DisplayBarcode { Mergefield Feldname } QR }*

Diese erstellten QR-Codes variieren jedoch je nach hinterlegter Zeichenanzahl recht stark in der Größe (siehe Abb. 32). Eine feste Größenangabe ist hier leider nicht möglich.



*Abbildung 32: Größenvergleich verschiedener aus Word erzeugter QR-Codes*

Schließlich fiel die Entscheidung auf ein kostenfreies Online-Tool zur Erstellung von Etiketten von Avery.<sup>217</sup> Hier können auf Grundlage vorhandener Daten, bspw. im XLSX-Format (Excel), nach einmaliger Erstellung eines Templates zahlreiche Etiketten erstellt werden und bspw. als PDF exportiert werden.

<sup>215</sup> Vgl.: <https://www.qrcode-monkey.com/>

<sup>216</sup> Vgl.: <https://www.qrcode-generator.de/>

<sup>217</sup> Vgl.: <https://www.avery-zweckform.com/software/avery-design-print>

Nach Anmeldung kann die Vorlage ggf. gespeichert werden. So ist es auch möglich, schnell neue QR-Codes zu erzeugen (siehe Abb. 33).



Abbildung 33: Erstellung einer QR-Code-Vorlage mit dem Avery-Tool

## 7.4 Direkter Zugang zum E-Book aus der App heraus

Als weitere neue Funktion, die den Nutzern im Rahmen der Weiterentwicklung zur Verfügung gestellt werden soll, galt die direkte Durchschaltung zum E-Book aus der App heraus, ohne Umweg über die Anbieterseite.

Zunächst musste hierfür ermittelt werden, für welche E-Book-Anbieter dies überhaupt möglich ist. Als Grundlage hierfür diente die Übersicht der Hochschulbibliothek über die Anbieter, von denen E-Books bezogen werden.<sup>218-219</sup>

Um die Angaben nochmals zu verifizieren wurde zu jedem E-Book-Anbieter nach Treffern in Wilbert recherchiert, um analysieren zu können, wie die Volltext-URLs aussehen und wie die Ressourcen bereitgestellt werden, also ob als PDF oder EPUB-Format und ob ein Volltext oder nur einzelne Kapitel zum Download bereitstehen (siehe Tabelle 6).

Für die Anbieter, die einen Download von Volltexten als PDF ermöglichen, sollte nun geprüft werden, ob eine Weiterleitung möglich ist. Dies betraf konkret die folgenden Anbieter:

- De Gruyter<sup>220</sup>
- Herdt-Campus

<sup>218</sup> Vgl. TH Wildau | Bibliothek (n. N.): E-Books, <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/recherche/e-books/>, 03.08.2021.

<sup>219</sup> Auf die Betrachtung des Anbieters Langenscheidt als Anbieter von Wörterbüchern soll hier verzichtet werden.

<sup>220</sup> Nach eigenen Recherchen konnten für De Gruyter Volltext-PDFs gefunden werden, sodass auch hier betrachtet werden soll, ob eine Durchschaltung zum E-Book möglich ist.

- (Nomos eLibrary)<sup>221</sup>
- SpringerLink
- Taylor & Francis
- WISO

Beim Anbieter De Gruyter sehen die Volltext-URLs, auf welche aus Wilbert heraus verwiesen wird, bspw. folgendermaßen aus:

<https://www.degruyter.com/document/doi/10.1524/9783486594058/html>

Beim Aufruf des Volltextes wird hier folgende Seite geöffnet:

<https://www.degruyter.com/document/doi/10.1524/9783486594058/pdf>

Beim Vergleichen der beiden URLs fällt ein Muster auf. Durch Ersetzen des Dateitypens im letzten Teil des Pfades könnte also direkt auf das Volltext-PDF-Dokument verwiesen werden, ohne erst die Anbieterseite aufrufen zu müssen. Da der Button hier als Verlinkung auf das Dokument angelegt ist (siehe Listing 10), kann die über die Solr-API hinterlegte URL ggf. ersetzt werden.

```
<a href="/document/doi/10.1524/9783486594058/pdf" class="downloadPdf btn btn-
primary font-weight-bold mb-4 py-2 w-100">
  <span>Buch downloaden (PDF)</span>
  
</a>
```

*Listing 85: Umsetzung der Weiterleitung zum Volltext-PDF bei De Gruyter*

Bei Volltexten von E-Books des Anbieters Hanser erreicht man bspw. folgende URL:

<https://herdt-campus.com/product/SQL2016A>

Nach Anklicken des Download-Buttons erfolgt eine Weiterleitung auf folgende Seite:

<https://herdt-campus.com/product/download/NKtmkg13MiL45OrcmpqfAmR0mPfty6D/SQL2016A>

Der Link lässt bereits vermuten, was das HTML-Skript bestätigt: Zum Öffnen des Volltext-PDFs wird ein individueller Key erzeugt, der sich nicht anhand eines vorgegebenen Musters ersetzen lässt:


```
<form action="/product/get-download-link" method="post" target="_blank">
  <input type="hidden" name="match_code" value="SQL2016A">
  <button type="submit" class="btn btn-primary btn-lg btn-
  block">Download</button>
</form>
```

*Listing 9: Umsetzung der Weiterleitung zum Volltext-PDF bei Herdt Campus*

Folglich ist hier eine Weiterleitung zum Volltext-PDF beim Anbieter Herdt-Campus nicht möglich.

<sup>221</sup> Da für den Anbieter Nomos leider keine Titel gefunden wurden, bei denen die E-Books aus Wilbert heraus über Nomos bereitgestellt wurden, können hier leider keine Aussagen getroffen werden. Auch eine Recherche nach den von Nomos bereitgestellten Verlagen in Wilbert führte nicht zum Erfolg. Anscheinend werden die Daten nicht in Wilbert geladen.



E-Book-Anbieter	Daten in Wilbert enthalten?	Dokumente als PDF vorhanden?	Volltext-PDF vorhanden?	URL automatisiert ersetzbar? (Bsp)	Bemerkungen
Beck-Online	ja	nein	<b>nein</b>		
De Gruyter	ja	Ja	Nein	<a href="https://www.degruyter.com/document/doi/10.1524/9783486594058/html">https://www.degruyter.com/document/doi/10.1524/9783486594058/html</a>  <a href="https://www.degruyter.com/document/doi/10.1524/9783486594058/pdf">https://www.degruyter.com/document/doi/10.1524/9783486594058/pdf</a> ja	Lt Angabe TH-Wildau-Webseite Download nur kapitelweise möglich, nach eigenen Tests zur Verifikation der Informationen wurden auch Volltexte von DeGruyter gefunden.
Ebook Central	Ja	nein	<b>nein</b>	nein	Zum Lesen der Volltexte wird Adobe Digital Editions benötigt; Seitenlimit beim Download
Hanser eLibrary	ja	ja	<b>nein</b>	nein	
Herdt Campus	ja	ja	nein	<b>nein</b>	Einfaches Ersetzen der Ziel-URL nicht möglich
Nomos eLibrary	ja	ja	ja	unklar	Auch nach verschiedenen Rechercheansätzen konnten in Wilbert keine Beispiele gefunden werden.
OAPEN (Open access E-Books)	<b>unklar</b>	ja	Ja		Laut Rückmeldung der Kolleg/innen der Hochschulbibliothek unklar, ob E-Books in Wilbert enthalten sind
ScienceDirect	ja	ja	<b>nein</b>		


E-Book-Anbieter	Daten in Wilbert enthalten?	Dokumente als PDF vorhanden?	Volltext-PDF vorhanden?	URL automatisiert ersetzbar? (Bsp)	Bemerkungen
<b>SpringerLink</b>	Ja	ja	teilweise	<a href="https://link.springer.com/book/10.1007%2F978-3-8351-9115-0">https://link.springer.com/book/10.1007%2F978-3-8351-9115-0</a>  <a href="https://link.springer.com/content/pdf/10.1007%2F978-3-8351-9115-0.pdf">https://link.springer.com/content/pdf/10.1007%2F978-3-8351-9115-0.pdf</a> <b>wenn Volltext-PDF vorliegt, ja</b>	
<b>Taylor &amp; Francis</b>	ja	ja	ja	<b>nein</b>	Einfaches Ersetzen der Ziel-URL nicht möglich
<b>Vahlen (Plattform beck eLibrary)</b>	ja	ja	<b>nein</b>		<i>Zum Lesen der Volltexte wird Adobe Digital Editions benötigt; Seitenlimit beim Download</i>
<b>WISO</b>	ja	Ja	teilweise	<b>nein</b>	Einfaches Ersetzen der Ziel-URL nicht möglich

Tabelle 6 : Prüfung der Durchschaltungsmöglichkeiten zu Volltexten der verschiedenen Anbieter - Übersicht

Bei Volltexten von E-Books des Anbieters SpringerLink erreicht man bspw. folgende URL:

<https://link.springer.com/book/10.1007%2F978-3-8351-9115-0>

Bei Anklicken des Buttons zum Volltext wird man auf folgende URL weitergeleitet:

<https://link.springer.com/content/pdf/10.1007%2F978-3-8351-9115-0.pdf>

Bei Analyse der Volltext-URL erkennt man, dass im Vergleich zur Anbieter-URL das Format von „book“ auf „content/pdf“ abgewandelt wurde. Zudem wurde der Pfad um „.pdf“ ergänzt.

Bei Betrachtung des Quelltexts sieht man zudem, dass der Button als Verlinkung angelegt ist und keine zusätzliche Funktion zum Erstellen eines Keys oder Ähnlichem aufgerufen wird (siehe Listing 12).

```
<div>
  <a href="/content/pdf/10.1007%2F978-3-8351-9115-0.pdf" target="_blank"
  class="c-button c-button--blue c-button__icon-right test-download-book-options
  test-bookpdf-link" title="Download this book in PDF format" rel="noopener" data-
  track="click" data-track-action="Book download - pdf" data-track-label="">
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" version="1.1"><g stroke="none" stroke-width="1" fill="none"
  fill-rule="evenodd"><g fill="#fff">[...] </path></g></g></svg>
    <span class="hide-text-small">Download</span>
    <span class="hide-text-small">book</span>
    <span>PDF</span>
  </a>
</div>
```

Listing 106: Umsetzung der Weiterleitung zum Volltext-PDF bei Springer Link

Bei Volltexten von E-Books des Anbieters Taylor & Francis erreicht man bspw. folgende URL:

<https://www.taylorfrancis.com/books/mono/10.1201/9781351188319/hands-course-sensors-using-arduino-raspberry-pi-volker-ziemmann>

Nach Anklicken des Buttons zum Volltext-PDF öffnet sich das Volltext-PDF-Dokument nicht im Browser, sondern der Download beginnt direkt. Hier laufen Netzwerkprozesse im Hintergrund ab (siehe Abb. 33).

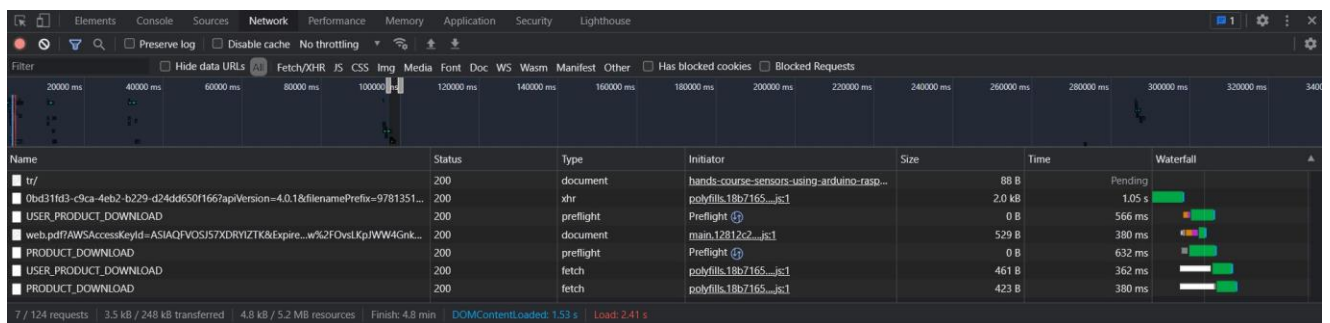


Abbildung 34: Screenshot der im Hintergrund ablaufenden Prozesse beim Download eines E-Books des Anbieters Taylor & Francis

Die E-Books des Anbieters WISO sind bspw. unter folgender URL erreichbar:

[https://www.wiso-net.de/document/EBOK,AEBO\\_\\_9783866187726239](https://www.wiso-net.de/document/EBOK,AEBO__9783866187726239)

Bei Anklicken des Buttons zum Download des Volltext-PDF-Dokuments werden JQuery-Funktionen aufgerufen (siehe Listing 12). Eine neue URL wird hier nicht aufgerufen, sodass auch hier keine Weiterleitung möglich ist.

```

<div class="moduleBigButton Pdf">
  <span class="button">
    <a id="g_262f847b2f614ad4b29773f8c6f6182c" href="#" class="ng-
secondary">Gesamtes Buch als PDF herunterladen (3 MB)</a>
    <script> jQuery(function(){
      jq('g_262f847b2f614ad4b29773f8c6f6182c').jasty("AjaxLink",
"init", [{"visible":true, "confirmTitle":"Hinweis",
"id":"g_262f847b2f614ad4b29773f8c6f6182c",
"params":{"srcId":"g_262f847b2f614ad4b29773f8c6f6182c",
"id":"EBOOK_9783866187726239", "index":["1"], "fileName":"EBOOK/2012/978-3-86618-
772-6.pdf", "type":-8, "isMobile":false}, "url":"/stream/downloadConsole"}]);
    })
  </script>
[...]
```

Listing 71: Umsetzung der Weiterleitung zum Volltext-PDF bei WISO

Eine Weiterleitung zum Volltext-PDF kann also nur für E-Books von De Gruyter und Springer Link erfolgen. Um diese zu identifizieren können zunächst die Informationen aus dem Feld „fulltext\_url“ der Solr-API genutzt werden. Es stellte sich jedoch heraus, dass dort nicht immer die Ziel-URLs der Anbieter hinter<sup>222</sup>legt sind. Stattdessen finden sich häufig URLs in der Form

<https://doi.org/10.1007/978-3-658-26290-7>

Diese enthalten einen Digital Object Identifier (DOI) und stellen eine persistente, zitierfähige Zugangsmöglichkeit zu digitalen Ressourcen dar. Eine DOI setzt sich aus folgenden Bestandteilen zusammen:

Präfix	{	10	=> DOI-Kennung
		1007	=> Verlagsnummer
Suffix	{	978-3-658-26290-7	=> Verweis auf einzelnes Objekt des Verlags

Über die DOI-URLs erfolgt eine Weiterleitung zur Anbieterseite.

Die Verwendung der DOI-URLs hat jedoch zur Folge, dass die eigentliche Anbieterseite nicht immer gleich erkenntlich ist. Die dahinterstehenden Anbieter können jedoch anhand der DOI ermittelt werden. Für De Gruyter und Springer Link konnten folgende DOIs ermittelt werden:

De Gruyter: 10.1515 und 10.1524  
Springer Link: 10.1007

Beim Anklicken des Buttons *E-Book ansehen* wird im Frontend die Funktion *link()* aufgerufen (siehe Listing 14). Um dem Nutzer anzuzeigen, dass der Volltext geöffnet wird, wird ein Ladesymbol angezeigt. Dies wird erreicht, in dem die Eigenschaft *fulltextLoading* der Anwendung auf *true* gesetzt wird. Bei der Umsetzung des Ladesymbols wurde ein wenig getrickst. Da ohne Wartezeit kein Ladesymbol erschien,

<sup>222</sup> Vgl. Schubert: DOI (Digital Object Identifier), <https://www.wiwi.uni-wuerzburg.de/fileadmin/12020700/DOI.pdf>, 22.08.2021.

wurde hier mit der Javascript-Funktion `setTimeout()` gearbeitet, welche das Zurücksetzen der Eigenschaft `fulltextLoading` zurück auf `false` etwas verzögert.

Anschließend wird anhand der DOI geprüft, ob es sich um E-Books handelt, die von Springer Link oder De Gruyter bereitgestellt werden. Ist dies der Fall, wird mit Hilfe der DOI der Backend-Endpunkt `/fulltext/:doi` aufgerufen.

```
link: function (url, doi) {
  this.setFulltextLoading(true)
  console.log('URL: ' + url)
  console.log('DOI:' + doi)
  console.log('decoded URI: ' + encodeURIComponent(doi))
  let fulltextUrl = BACKEND_URL + '/fulltext/' + encodeURIComponent(doi)
  console.log(fulltextUrl)
  if (doi !== '' && (doi.startsWith('10.1524') || doi.startsWith('10.1515')
  || doi.startsWith('10.1007'))) {
    axios.get(fulltextUrl)
      .then(response => {
        console.log(response.data)
        if (response.data === 'useStandardUrl') {
          window.open(url, '_blank')
        } else {
          window.open(encodeURIComponent(response.data))
        }
      })
      .catch(error => {
        console.log(error.message)
        window.open(url, '_blank')
      })
  } else {
    window.open(url, '_blank')
  }
  setTimeout(() => {
    this.setFulltextLoading(false)
  }, 3000)
  // this.setFulltextLoading(false)
}
```

*Listing 82: Frontend-Funktion link zum Öffnen der E-Book-Volltexte*

Dieser ist dafür verantwortlich, die URL zu ersetzen und zu prüfen, ob die ersetzte URL tatsächlich erreichbar ist, also ob der Statuscode 200 zurückgeliefert wird. Ist dies der Fall, liefert das Backend die ersetzte URL zurück. Ist die ersetzte URL nicht erreichbar oder an anderer Stelle ein Fehler aufgetreten, liefert das Backend die Zeichenfolge „useStandardUrl“ zurück (siehe Listing 15).

Je nach Antwort öffnet das Frontend dann die ersetzte oder bisher hinterlegte URL.

```

app.get('/fulltext/:doi', function (req, res) {
  let doi = req.params.doi;
  let url = ''
  if (doi.startsWith('10.1007')) {
    url = 'https://link.springer.com/content/pdf/' + encodeURIComponent(doi)
+ '.pdf'
    console.log('geänderte URL: ' + url)
    axios.get(url)
      .then( (response) => {
        console.log(response.status);
        if (response.status == 200) {
          res.send(encodeURIComponent(url))
        } else {
          res.send('useStandardUrl')
        }
      })
      .catch( (error) => {
        console.log('ERROR: ' + error.message);
        res.send('useStandardUrl')
      })

    } else if (doi.startsWith('10.1524')) {
      console.log('DOI: 1524');
      url = 'https://www.degruyter.com/document/doi/' + encodeURIComponent(doi)
+ '/pdf'
      console.log('geänderte URL: ' + url)
      axios.get(url)
        .then( (response) => {
          console.log(response.status);
          if (response.status == 200) {
            res.send(encodeURIComponent(url))
          } else {
            res.send('useStandardUrl')
          }
        })
        .catch( (error) => {
          console.log('ERROR: ' + error.message);
          res.send('useStandardUrl')
        })

    } else if (doi.startsWith('10.1515')) {
      console.log('DOI: 1515');
      url = 'https://www.degruyter.com/document/doi/' + encodeURIComponent(doi)
+ '/pdf'
      console.log('geänderte URL: ' + url)
      axios.get(url)
        .then( (response) => {
          console.log(response.status);
          if (response.status == 200) {
            res.send(encodeURIComponent(url))
          } else {
            res.send('useStandardUrl')
          }
        })
        .catch( (error) => {
          console.log('ERROR: ' + error.message);
          res.send('useStandardUrl')
        })
    } else {
      res.send('useStandardUrl')
    }
  });

```

*Listing 93: Prüfung der DOIs und Ersetzung der Volltext-URLs im Backend*

## 7.5 Integration entliehener Werke

Eine weitere Anforderung war die Integration entliehener Werke, um dem Nutzer der App einen noch vollständigeren Bestandsüberblick präsentieren zu können.

Diese Informationen lassen sich jedoch nicht durch eine Anfrage der Solr-API gewinnen, da diese lediglich Titelinformationen, jedoch keine Exemplardaten wie Signatur oder Ausleihstatus zurückliefert. Deshalb wurde von Herr Kissig, dem IT-Beauftragten der Hochschulbibliothek, nach Übermittlung der Anforderungen<sup>223</sup> eine entsprechende Schnittstelle entwickelt, welche eine Antwort im JSON-Format zurückgibt:

```
{
  "returnVal" : "ok",
  "requestId" : "ST250",
  "timestamp" : 1628323317,
  "items" : [
    {
      "status" : 4,
      "place" : "München [u.a.]",
      "autor_avs" : "Sedgewick, Robert",
      "key" : 106112,
      "holds" : 0,
      "checkedOutItems" : "1",
      "isbn" : "978-3-86894-076-3",
      "publisher" : "Pearson",
      "sign" : "ST 250 J35 S448",
      "zusatz" : " ",
      "barcode" : "201227078",
      "mediatype" : 0,
      "title_avs" : "Einführung in die Programmierung mit Java",
      "totalItems" : "1",
      "verfasser" : "Sedgewick, Robert;Wayne, Kevin",
      "title" : "Einführung in die Programmierung mit Java",
      "duedate" : "Dec 24 2021 12:00AM",
      "year" : "2011"
    }
  ],
}
```

*Listing 104: Antwort der API für entliehene Bände im JSON-Format (Ausschnitt)*

Da es darum ging, die entliehenen Medien thematisch passend zum Nutzerstandort abbilden zu können, war eine Anforderung an die API, dass sie mit Hilfe der RVK-Notation abfragbar ist. Dementsprechend soll eine Anzeige der entsprechenden Titel auch nur nach Einscannen eines QR-Codes angezeigt werden, nicht aber bei einer thematischen Suche.

Die Schnittstelle liefert alle entliehenen Exemplare zurück. Sind von einem Titel mehrere Exemplare ausgeliehen, so werden alle Exemplare von der Schnittstelle zurückgeliefert. Um dem Nutzer keine entliehenen Titel doppelt anzuzeigen, muss also die Treffermenge zunächst auf doppelte Treffer durchsucht und entsprechend gefiltert werden. Dies erfolgt mit Hilfe des Keys.<sup>224</sup> Das Filtern hat jedoch zur Folge, dass jeweils der zuerst in der Liste gefundene Band zurückgegeben wird. Problem ist an dieser Stelle, dass das der erste Band nicht unbedingt derjenige ist, welches als erstes zurückgegeben wird. Im Falle einer Vormerkung (siehe Kapitel 6.7) interessiert den Nutzer aber in der Regel, wann er den Band frühestens

---

<sup>223</sup> Siehe Anhang A.6

<sup>224</sup> Gemeint ist hier die Nummer des Titeldatensatzes im Katalog.

erhalten kann.<sup>225</sup> Deshalb ist es notwendig, vorm Filtern nach Keys zunächst die Treffer nach Duedate zu sortieren. Dieses ist mit dreistelligen Monatsbezeichnungen, gefolgt mit numerischen Angaben zum Tag, Jahr und zur Uhrzeit angegeben, z.B. „Dec 24 2021 12:00AM“. Um die Treffer also nach Duedate sortieren zu können, wurden die einzelnen Angaben zunächst voneinander gesplittet, um anschließend mehrere aufeinanderfolgende Sortierdurchgänge durchzuführen. Die Splittung wurde zunächst anhand eines Space-Zeichens versucht. Dabei traten jedoch Probleme mit einstelligen Tagesangaben auf, welche nicht in der Form „07“, sondern „ 7“ (mit vorangestelltem Space) angegeben waren. Deshalb wurden zunächst doppelte Spaces durch einfache ersetzt. Anschließend folgten die einzelnen Sortiervorgänge (siehe Listing 17):

- (1) Im ersten Sortierdurchgang wurde nach dem Jahr (vierstellig) sortiert.
- (2) Als Vorbereitung für den zweiten Sortiervorgang wurden zunächst die Angaben zu den Monaten mit Hilfe eines Hilfsarrays in numerische Werte umgewandelt, in dem die Monatskürzel auf die Indexwerte abgebildet und um 1 addiert wurden. Den einstelligen Indexwerten wurde anschließend noch eine 0 vorangestellt, damit es bei der Sortierung nicht zu Problemen kommt.
- (3) Im dritten Sortierdurchgang wurde nach dem Tag sortiert. Hier wurden wie bei den Monaten auch den einstelligen Tagesangaben zuvor noch eine Null vorangestellt, um eine korrekte Sortierung zu gewährleisten.

Für die Anzeige in der App wurden die einzelnen Elemente anschließend wieder in der Form „DD.MM.YYYY“ zusammengefügt

```
allRecords = allRecords.filter(record => record.mediatype !== 3);
allRecords.forEach( function (element, index, array) {
    array[index].duedate = element.duedate.replace(/\s\s+/g, ' ');
})
// sorting the array in order to duedate
// split array.duedate in three elements
// sorting three times: year, month, date
// split duedate in four elements (month, day, year, time)
allRecords.forEach( function (element, index, array) {
    array[index].duedate = element.duedate.split(' ')
})
// matching months to numbers
let months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
'Sep', 'Oct', 'Nov', 'Dec'];
allRecords.forEach( function (element, index, array) {
    array[index].duedate[0] = (months.indexOf(element.duedate[0]) +
1).toString()
    // change single digit day or month to double-digit
    // month
    if (array[index].duedate[0].length == 1) {
        array[index].duedate[0] = '0' + element.duedate[0]
    }
})
```

---

<sup>225</sup> Hier wird exemplarisch davon ausgegangen, dass der Nutzer den Band an erster Stelle vormerkt. Ist der entsprechende Titel bereits einmal vorgemerkt, gelten andere Rückgabefristen.



```

    }
    // day
    if (array[index].duedate[1].length == 1) {
        array[index].duedate[1] = '0' + element.duedate[1]
    }
})
// a) sorting by month
allRecords.sort((a,b) => {
    if (a.duedate[0] > b.duedate[0]) {
        return 1;
    }
    if (a.duedate[0] < b.duedate[0]) {
        return -1;
    }
    return 0;
})
// b) sorting by day
allRecords.sort((a,b) => {
    if (a.duedate[0] === b.duedate[0] && a.duedate[1] > b.duedate[1]) {
        return 1;
    }
    if (a.duedate[0] === b.duedate[0] && a.duedate[1] < b.duedate[1]) {
        return -1;
    }
    return 0;
})
// c) sorting by year
allRecords.sort((a,b) => {
    if (a.duedate[2] > b.duedate[2]) {
        return 1;
    }
    if (a.duedate[2] < b.duedate[2]) {
        return -1;
    }
    return 0;
})
// preparing duedate to display it in the app
allRecords.forEach( function (element, index, array) {
    array[index].duedate = element.duedate[1] + '.' + element.duedate[0] +
    '.' + element.duedate[2];
    // console.log(array[index].duedate);
})
// filter by key to get each title just once in the list
allRecords = allRecords.filter((record, index, self) => (
    index === self.findIndex((t) => (
        t.key === record.key
    )))
))
// sorting by title
allRecords.sort((a,b) => {
    if (a.title > b.title) {
        return 1;
    }
    if (a.title < b.title) {
        return -1;
    }
    return 0;
})
response.send(allRecords);
}
}

```

*Listing 115: Sortierung der Ergebnisse der API für entliehene Bände*

Ein weiteres Problem stellten die Exemplare dar, die einem Semesterapparat angehören. Bei diesen Bänden handelt es sich um Präsenzexemplare, die nicht ausleihbar sind. Die Zugehörigkeit zu einem Semesterapparat wird in SISIS allerdings durch die Ausleihe auf genau diesen abgebildet, sodass auch diese von der API zurückgeliefert werden. Durch die Angabe des Medientyps lassen sich diese allerdings gut erkennen und entsprechend filtern (siehe Listing 17).

Vom Frontend aus werden Informationen in der Form

„Kategorie | RVK-Notation (ggf.. von ... bis ...) | Start | Trefferzahl | ggf. Schlagwort“

ans Backend übergeben.

Wie beim Endpunkt „/BooksQRCode/:qrCodeInfo“ werden auch hier die benötigten Informationen aus dem QR-Code-Inhalt abstrahiert. Von Bedeutung für die Anzeige der entliehenen Bände ist nur die Information zur RVK-Notation. Wichtig ist hier, zu erkennen, ob es sich um eine einzelne RVK-Notation oder um eine Range handelt. Im Falle einer einzelnen RVK-Notation kann mit Hilfe von Axios direkt eine Anfrage an die API vorbereitet und ausgeführt werden. Ist hingegen eine Range angegeben, so sollen im Rahmen des Prototyps zumindest die erste und letzte Notation abgefragt werden. Dafür müssen zwei API-Abfragen durchgeführt und die Ergebnisse zusammengeführt (und anschließend gefiltert und sortiert werden).

```
app.get('/lentBooks/:qrCodeInfo', function(req, response){
  console.log('lentBooks-Request');
  console.log('got QR-Code information');
  let codeInfos = req.params.qrCodeInfo
  let infoArr = codeInfos.split('|');
  // let infotype = infoArr[0]; // either rvk or schlagwort
  let rvkNotation = infoArr[1].split(' - ');
  console.log('Array nach Split: ' + rvkNotation);
  rvkNotation.forEach( function( element, index, array) {
    array[index] = element.replace(/ /, '')
  })
  console.log('RVK-Notation: ' + rvkNotation);
  if (rvkNotation.length > 1) {
    let requestOne = axios.get(lentBooksUrl + rvkNotation[0])
    let requestTwo = axios.get(lentBooksUrl + rvkNotation[1])
    axios.all([requestOne, requestTwo])
      .then((responseArr) =>
        {
          console.log('Response 1: ' + responseArr[0].data.items);
          console.log('Response 2: ' + responseArr[1].data.items);
          let allResponses =
responseArr[0].data.items.concat(responseArr[1].data.items)
          console.log(allResponses);
          tools_lentItems.parseData(response, allResponses)
          // response.send('This is a test');
        })
  }
  } else {
    let localUrl = lentBooksUrl + rvkNotation[0];
    console.log(localUrl);
    let allRecords = [];
    axios.get(localUrl, {
      /*
      proxy: {
        host: 'proxy.th-wildau.de',
        port: 8080
      }
      */
    }).then((res) => {
```

```

        console.log(res.data);
        console.log(res.data.returnVal);
        console.log(res.data.requestId);
        tools_lentItems.parseData(response, res.data.items)
    }).catch((err) =>{
        console.log('ERROR');
        console.log(err.message);
    })
}
//response.send('Testausgabe');
})

```

Listing 126: Backend-Endpunkt zur Bereitstellung entliehener Bände

Die Anzeige der entliehenen Bände erfolgt in Form einer Liste:

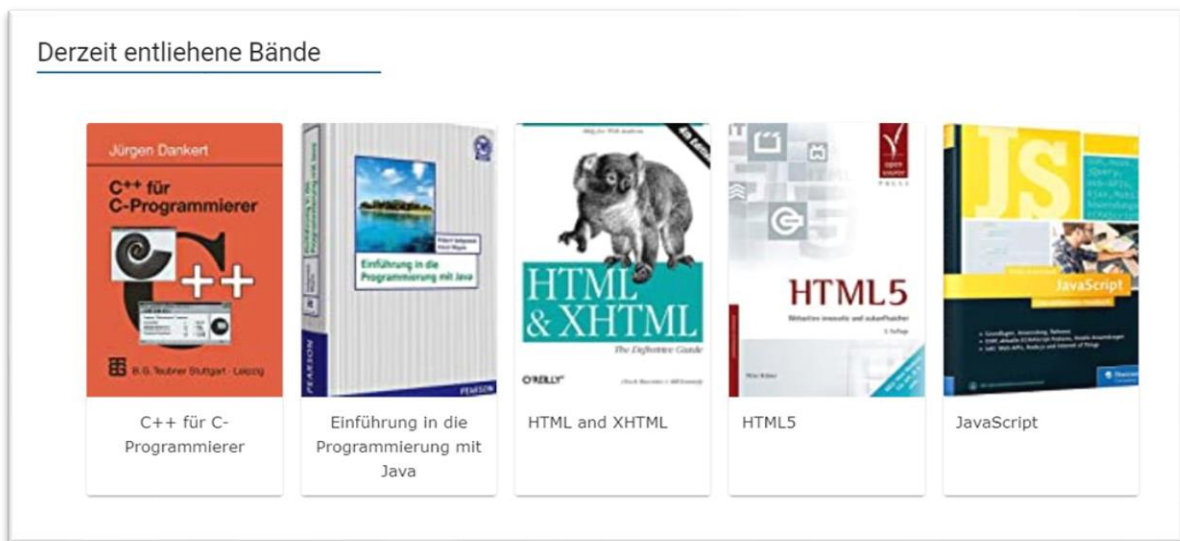


Abbildung 35: Anzeige entliehener Bände als Liste

Für den Fall, dass keine Bände für die angefragte/n RVK-Notation/en entliehen sind, erfolgt eine entsprechende Information an den Nutzer.



Abbildung 36: Hinweis, wenn keine entliehenen Bände gefunden wurden

Für die Detailansicht der entliehenen Bände wurde zudem eine neue Komponente erstellt, welche unter anderem das Due date anzeigt und nach Anmeldung in der App zusätzlich einen Button integriert, die Bände vorzumerken. Das angezeigte Due date bezieht sich auf den Band, der laut Bibliothekssystem als

nächstes zurückgegeben werden müsste/dessen Leihfrist als nächstes endet. Ist jedoch bereits eine Vormerkung auf diesen Titel hinterlegt, kann dieses Datum abweichen. Deshalb ist es wichtig, zusätzlich einen entsprechenden Hinweis zu hinterlegen.



Abbildung 37: Detailansicht der entliehenen Bände

## 7.6 Vormerkung

Nachdem der Button für die Vormerkfunktion im Frontend untergebracht war, galt es nun zu überlegen, wie die eigentliche Vormerkung optisch und funktional integriert werden sollte. Voraussetzung für die Aufgabe einer Vormerkung ist die Anmeldung in der App, welche bereits umgesetzt war. Ist der Nutzer nicht in der App angemeldet, erscheint statt des Vormerk-Buttons ein entsprechender Hinweis:

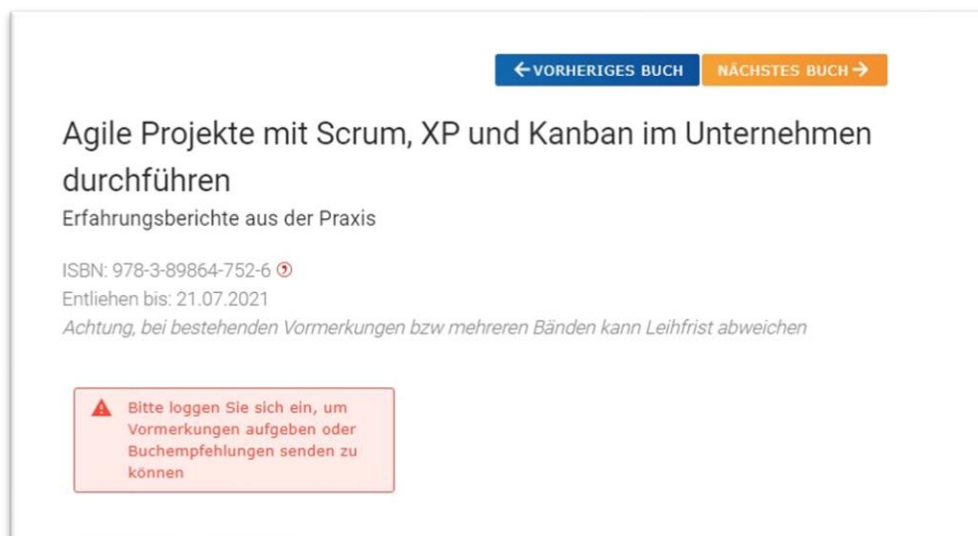


Abbildung 38: Anmeldehinweis für Aufgabe einer Vormerkung

Um zu verhindern, dass ein Nutzer versehentlich den Button „Vormerkung für diesen Band“ betätigt, soll dieses zunächst ein Popup-Fenster öffnen, in welchem der eigentliche Vormerkwunsch zunächst noch einmal bestätigt werden muss. Erst wenn dies geschehen ist, wird eine entsprechende Anfrage ans Backend gestellt:

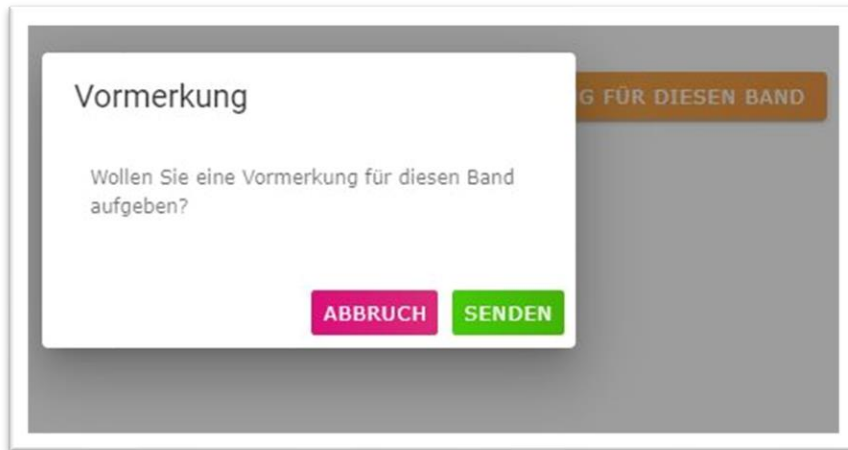


Abbildung 39: Popup zum Aufgeben einer Vormerkung

Auch Wilbert bietet die Möglichkeit, entlehene Bände vorzumerken. Die dort genutzte Funktionalität leitet derzeit zum konventionellen Online-Katalog der Hochschulbibliothek weiter, um die Bibliotheks-Nutzererkennung und das Bibliothekspasswort abzufragen. Eine Nachnutzung dieser ist von der Funktionalität her also nicht zielführend.

Stattdessen bietet die Hochschulbibliothek eine Schnittstelle an, die es ermöglicht, anhand des Medienbarcodes und der Bibliothekskennung (ohne Eingabe des Bibliothekspassworts) Vormerkungen aufzugeben. Die Nutzung der API steht nur autorisierten LDAP-Nutzern der TH Wildau zur Verfügung.

Die Schnittstelle führt teilqualifizierte Vormerkungen aus, welche sich im Gegensatz zu vollqualifizierten Vormerkungen nicht auf ein bestimmtes Exemplar, sondern auf einen Titel beziehen.<sup>226</sup> Somit wird das erste zurückgegebene Exemplar für die erste Vormerkung bereitgestellt.

Die notwendigen Informationen zum Barcode des vorzumerkenden Bandes<sup>227</sup> können der Schnittstelle für die vorgemerkten Medien entnommen und in der App gespeichert werden.

Da die Nutzung der Vormerkfunktion nur nach Anmeldung in der App angeboten werden sollte, können die Informationen zur Bibliotheks-ID des Nutzers im Rahmen der LDAP-Anmeldung gewonnen werden. Durch Omar Attia wurde im Rahmen seiner Bachelorarbeit bereits eine Funktion zur LDAP-Anmeldung im Frontend umgesetzt. Diese speichert u.a. folgende nutzerbezogene Informationen:

- Nutzernamen
- Rolle
- Studiengang
- E-Mail-Adresse

<sup>226</sup> Im Gegensatz dazu bezieht sich eine vollqualifizierte Vormerkung auf ein bestimmtes Exemplar.

<sup>227</sup> Trotz Angabe des Barcodes bezieht sich die Vormerkung auf den Titel und nicht den bestimmten Band!

- API-Token

Die LDAP-Schnittstelle, über die die Anmeldung umgesetzt wird, liefert darüber hinaus die Bibliotheks-ID, sodass auch diese in der App hinterlegt werden kann. Dazu wurde der State des Frontend entsprechend erweitert, sodass nun alle Informationen vorhanden waren, welche für die Vormerkung eines Bandes benötigt wurden.

Der eigentliche Aufruf der Vormerkfunktion geschieht wieder über das Backend. Hierzu wird aus dem Frontend heraus der Endpunkt

`/placeReservation/:cardNr/:barcode`

aufgerufen. Anschließend werden die entsprechenden Informationen an die API weitergegeben und das Ergebnis ausgewertet (siehe Listing 19). Wichtig ist, dass bei Aufruf der API für die Ausführung der Vormerkung die Daten zur Autorisierung im Header mitgegeben werden.

```
app.get('/placeReservation/:cardNr/:barcode', async function (req, res) {
  let cardNr = req.params.cardNr;
  let barcode = req.params.barcode;
  //console.log('\x1b[33m%s\x1b[0m', '-----
  ---');
  console.log('\x1b[31m%s\x1b[0m', '\nPlace a reservation in Sisis');
  console.log('library card number: ' + cardNr);
  console.log('barcode: ' + barcode);
  let resUrl = reservationURL + cardNr + '/' + barcode;
  console.log('URL: ' + resUrl);
  axios.get(resUrl, {
    headers: {
      'Authorization': `Basic ${base64.encode(config.res_api_user + ':' +
config.res_api_pw)}`
    },
  })
  .then( response => {
    console.log(response.data);
    res.send(response.data);
  })
  .catch (error => {
    console.log('ERROR: ' + error.message);
  })
})
```

*Listing 137: Backend-Endpunkt zum Aufgeben einer Vormerkung*

Als Antwort liefert die Vormerk-API einen JSON-String:

```
{ "ok": true, "available": false, "transactionDate": "2021-08-07T12:40:06.000Z", "expirationDate": "", "queuePosition": "1", "pickupLocation": "Halle 10 (Hochschulbibliothek)", "institutionId": "", "patronIdentifizier": "15007746916", "itemIdentifizier": "", "titleIdentifizier": "", "screenMessage": ["Kotulla, Andreas", "Management von Softwareprojekten", "Ausgabeort : 00 Theke", "Vormerkung o.k. Vormerkung an 1. Stelle"], "printLine": ["Kotulla, Andreas", "Management von Softwareprojekten", "Ausgabeort : 00 Theke", "Vormerkung o.k. Vormerkung an 1. Stelle"], "sequence": 0, "checksum": "8D5B", "raw": "161N20210807144006BW|BR1|BSHalle 10 (Hochschulbibliothek)|AO|AA15007746916|AB|AJ|AFKotulla, Andreas|AFManagement von Softwareprojekten|AFAusgabeort : 00 Theke|AFVormerkung o.k. Vormerkung an 1. Stelle|AGKotulla, Andreas|AGManagement von Softwareprojekten|AGAusgabeort : 00 Theke|AGVormerkung o.k. Vormerkung an 1. Stelle|AY0AZ8D5B\r" }
```

Listing 18: Antwort der Vormerkungs-API im JSON-Format

Das Backend liefert schließlich die Antwort der Vormerk-API oder eine Errormeldung an das Frontend zurück, welches die Informationen entsprechend verarbeitet und den State der App entsprechend anpasst, der für die Anzeige des Ergebnisses des Vormerkvorgangs verantwortlich ist (siehe Listing 21).

```
async placeReservation ({ commit }, reservationItem) {
  let url = BACKEND_URL + '/placeReservation/' + state.libraryID + '/' +
  reservationItem.barcode
  // console.log(url)
  commit('setReservationState', 'loading')
  await axios.get(url).then(function (response) {
    // console.log(response)
    // console.log(response.data)
    // console.log(response.data.screenMessage[2])
    // console.log(response.data.screenMessage[3])
    commit('setReservationResult', response.data)
    if (response.data.ok === true) {
      commit('setReservationState', 'success')
    } else {
      commit('setReservationState', 'failed')
    }
    if (response.data === 'API_ERROR') {
      commit('setReservationState', 'failedAPI')
    }
  }).catch((error) => {
    console.log('upps, something went wrong while placing a reservation')
    console.log(error.message)
    commit('setReservationState', 'failedAPI')
  })
},
```

Listing 19: Verarbeitung der Informationen der Vormerk-API im Frontend

Je nach Inhalt des *ReservationState* wird eine farblich entsprechend gefärbte Information über Erfolg oder Misserfolg der Vormerkung an den Nutzer weitergegeben. Zusätzlich werden die Daten des *screenMessage-Arrays* für die Weitergabe an den Nutzer verwendet (siehe Abb. 40).

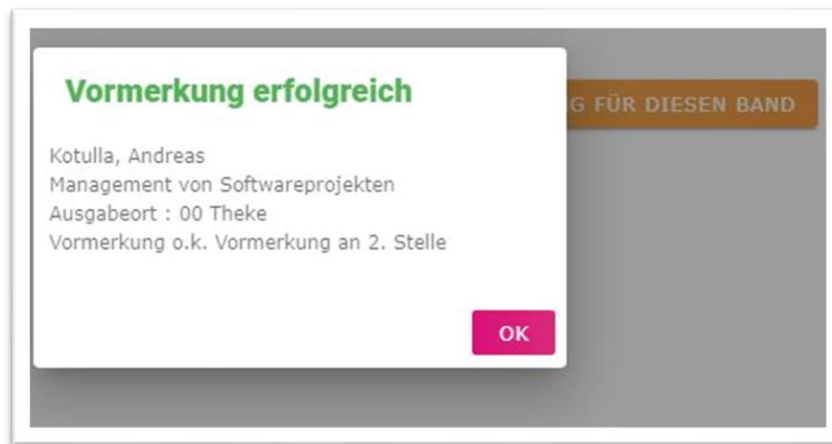


Abbildung 40: Information des Benutzers über eine erfolgreiche Vormerkung

## 7.7 Zusammenfassung technische Umsetzung

Bereits die Ermittlung und Entscheidung der Verteilung der QR-Codes über die Regale gestaltete sich schwierig. Durch die Entscheidung für „mitwandernde“ QR-Codes war im Voraus wesentlich mehr Planung und Vorbereitung notwendig. Da sich bei der Verteilung der Treffermengen für eine RVK-Notation auf mehrere QR-Codes an den aktuellen Treffermengen der Solr-API orientiert wurde, ist hier wahrscheinlich, dass mit wachsendem Bestand an E-Books ein Nachjustieren notwendig ist. Trotz des Versuchs, zum Teil (bei nicht zu hohen Trefferzahlen), RVK-Notationen zusammenzufassen, kommt es vor, dass auf einer Regalebene mehrere QR-Codes hängen. Wichtig ist an dieser Stelle vor allem die entsprechende Beschriftung dieser. Dies ist mit dem Avery-Tool individuell möglich. Als Datengrundlage kann hier bspw. eine Excel-Tabelle genutzt werden.

Die Integration der entliehenen Bücher bei der Suche nach Literatur nach RVK-Notationen hat erfolgreich funktioniert. Derzeit werden diese in einer separaten Liste unter den thematisch passenden E-Books aufgeführt. Langfristig könnte hier evtl. eine Liste mit sowohl entliehenen Print-Büchern als auch E-Books implementiert werden, wobei die verschiedenen Medientypen in der Liste entsprechend gekennzeichnet sind. Auch die Vormerkfunktion konnte dank der bereitgestellten Schnittstelle erfolgreich umgesetzt werden.

Lediglich die Durchschaltung zu den Volltexten der E-Books konnte lediglich für die Verlage Springer und De Gruyter umgesetzt werden. Andere Anbieter nutzen Verfahren zum Zugänglichmachen der Volltexte, die ein direktes Zugreifen aus der zu entwickelnden App heraus verhindern.



## 8 Abschlussbetrachtungen

Im Gegensatz zu den Masterarbeiten, die sich mit der Integration des Physischen (hier: Gedruckten) ins Digitale beschäftigen, sollte diese Arbeit wie die von Jasmin Taubert den Ansatz verfolgen, ein Konzept für die Integration des Digitalen ins Physische zu verfolgen. Zielstellung der Arbeit war die Entwicklung eines Konzepts für eine Anwendung zur Überwindung des Medienbruchs beim Stöbern am Regal für die Hochschulbibliothek der TH Wildau. Für diese sollten ausgewählte Funktionen aufbauend auf einem bestehenden Prototyp umgesetzt werden.

Im Rahmen der Masterarbeit wurden die Entwicklungsperspektiven des bestehenden Prototyps konzeptuell zusammengefasst. Hierbei wurden Problemstellungen und neue Perspektiven für die weitere Entwicklung der App aufgezeigt.

In den Abschlussbetrachtungen soll nun zunächst der weiterentwickelte Prototyp mit den aufgestellten funktionalen Anforderungen abgeglichen und so die Umsetzung der Anforderungen geprüft werden. Anschließend soll ein Fazit zur praktischen Umsetzung gezogen und aufgetretene Probleme zusammengefasst werden. Abschließend wird ein Ausblick auf weitere Entwicklungsmöglichkeiten geworfen.

### 8.1 Umgesetzte Anforderungen

Betrachtet man die funktionalen Anforderungen, so wurden von insgesamt 37 unterschiedlichen Anforderungen 23 erfolgreich in der App umgesetzt (F\_01, F\_02, F\_07 – F\_16, F\_20, F\_22, F\_24, F\_26 - F\_28, F\_33 – F\_35, F\_37). Ein Großteil der Funktionen fanden sich dabei bereits im bereits bestehenden Prototypen. Lediglich sechs der gestellten Anforderungen wurden bisher noch nicht umgesetzt: Dazu gehört bspw. die Anzeige verschiedener Medientypen in einer Liste, sodass diese auch optisch voneinander unterscheidbar sind (F\_17). Derzeit werden unterschiedliche Medientypen (hier: E-Book und entliehener Printband) in getrennten Listen dargestellt. Weiterhin noch nicht integriert sind die Abbildung der Suchhistorie (F\_19), die Anzeige der Gesamtzahl der Treffer (F\_21) sowie eine Anreicherung der Detailinformationen um eine Kurzbeschreibung des Inhalts bspw in Form eines Abstracts (F\_23). Auch eine Recommenderfunktion mit der Möglichkeit der Bewertung einzelner Titel ist noch nicht umgesetzt (F\_30 – F\_32). Zuletzt ist auch die Entwicklung eines Tutorials für die App durch Wilma noch offen (F\_36).

Fünf Anforderungen konnten zumindest zum Teil umgesetzt werden. Dazu gehört zunächst die Ermittlung (F\_04) und Anzeige (F\_05) der zum Nutzerstandort passenden E-Ressourcen. Eine Implementation ist zunächst nur für E-Books erfolgt. Auch die Integration entliehener Werke (F\_06) wurde zunächst nur für die Suche nach Bänden anhand der RVK-Notation umgesetzt. Filtermöglichkeiten sind vorerst nur für die Funktion „E-Books suchen“ implementiert. Zu klären ist, inwieweit diese Funktion bei der Abbildung von Titeln passend zum Nutzerstandort sinnvoll ist. Hier sind die Treffer derzeit alphabetisch nach Verfasseramen sortiert, passend zum Regalstandort. Zuletzt konnte die Durchschaltung zum Volltext aufgrund der technischen Gegebenheiten nur für einzelne Anbieter implementiert werden.

Insgesamt konnten also bereits ca 85% der funktionalen Anforderungen (ggf. zum Teil) umgesetzt werden.

## 8.2 Fazit zur praktischen Umsetzung

Durch die Nutzung zweier auf JavaScript aufbauender Frameworks für Backend und Frontend musste bei der Implementierung der neuen Funktionen nicht ständig gedanklich zwischen zwei Programmiersprachen mit möglicherweise anderen Paradigmen gewechselt werden.

Durch das Express-Framework, welches auf Node.js aufbaut, ließen sich relativ schnell neue Backend-Endpunkte und die dazugehörigen Funktionen hinzufügen.

Im Bereich des Frontends machte sich vor allem die komponentenbasierte Entwicklung einzelner Bestandteile sowie die daraus folgende Wiederverwendbarkeit dieser bezahlt. Durch die Einbindung der i18n-Bibliothek lässt sich mit Hilfe von entsprechenden Sprachdateien unproblematisch eine Anwendung in verschiedenen Sprachen entwickeln. Beim Hinzufügen neuer Beschriftungen müssen lediglich die entsprechenden Sprachdateien ergänzt werden.

Durch die Nutzung der Vuetify-Bibliothek, welche die Elemente des Material Designs abbildet, kann auf das Schreiben nicht immer ganz einfacher CSS-Styles verzichtet werden.

Durch die Größe der App macht sich die Nutzung der Vuex-Bibliothek bezahlt, welche ein zentrales State-Management ermöglicht. Durch die Bereitstellung eines „Single Source of Truth“ können alle Komponenten jederzeit auf die zentral vorgehaltenen States der Anwendung zugreifen. Ein kompliziertes Hin- und Herreichen von Properties zwischen einzelnen Komponenten ist dadurch nicht mehr notwendig.

## 8.3 Problemstellungen

Motivation der App ist die möglichst vollständige Präsentation des Bestandes der Hochschulbibliothek der TH Wildau zu einem bestimmten Themengebiet während des Stöberns am Regal. Leider stellte sich bereits bei der Auswahl der für die Zielstellung nutzbaren Metadaten heraus, dass es kein Metadatum gibt, über welches sowohl Print- als auch E-Books verfügen und welches den Inhalt des Mediums bezeichnet. Da die Aufstellung in der Hochschulbibliothek der TH Wildau anhand der RVK-Notation erfolgt galt es, diese als Grundlage zu nutzen, da diese quasi als Grundlage für die thematische Einordnung des Nutzerstandortes dienen. Die von verschiedensten Anbietern bereitgestellten E-Books weisen eine Vielfalt an Inhalterschließungs-Metadaten auf. Einige Anbieter liefern Informationen zu RVK-Notationen, andere erschließen eher auf Grundlage der DDC-Dezimalklassifikation. Durch das Zugrundelegen der RVK-Notation scheiden leider bereits die E-Books des Anbieters Ebook Central komplett aus. Langfristiges Ziel wäre hier entweder eine Anreicherung der bibliographischen Metadaten um RVK-Notationen oder ein entsprechendes Mapping-Tool, welches es ermöglicht, RVK-Notationen auf die passenden DDC-Notationen abzubilden. Die über das Cocoda Tool bereitgestellten Mappings reichen hierfür noch nicht aus.

Auch die Planung der Hinterlegung der Standortinformationen auf den QR-Codes stellte eine unerwartete Herausforderung dar. Wurde im Konzept eine Empfehlung gegen statische QR-Codes, die fest pro Regal oder Regalboden angebracht werden, ausgesprochen, da diese den Nachteil eines nicht zu vernachlässigendem dauerhaften Pflegeaufwandes mit sich brachten, zeigten sich während der Umsetzung schnell die Schwächen der „mitwandernden“ QR-Codes. Benötigt man für die statischen QR-Codes lediglich die Notation des ersten und letzten Bandes pro Regalbereich, so sind beim „mitwandernden“ QR-Code doch

weitaus mehr Vorüberlegungen notwendig. Zunächst muss entschieden werden, für welche Notationsbereiche QR-Codes erstellt werden müssen. Dabei muss zudem beachtet werden, dass nicht thematisch zu sehr entfernte Notationen unter einem QR-Code zusammengefasst werden. Ein weiterer wichtiger Aspekt ist anschließend die zu erwartende Trefferzahl. So würde es bspw. nicht viel Sinn machen, dem Nutzer über 100 Treffer zu einer Notation zu präsentieren, da die Wahrscheinlichkeit, dass ein Nutzer die Treffer durchschaut sehr gering ist. Darüber hinaus würden beim Laden von hohen Trefferzahlen auch entsprechende Lade- und somit Wartezeiten für den Nutzer entstehen.

Zu überlegen ist darüber hinaus, wie die Treffer sortiert werden sollten. Um sich an der Regalaufstellung zu orientieren, wurde für den Prototyp eine alphabetische Sortierung der Treffer nach Verfassernamen vorgenommen. Eine Sortierung nach Scorewert würde erst bei einer freien Suche Sinn machen. Bei der Suche lediglich nach RVK-Notation wäre der Scorewert immer gleich.

So zeigt sich, dass hier doch weitaus mehr Vorüberlegungen und Planungen anzustellen sind, als bei statischen QR-Codes. Ist der Pflegeaufwand im Vergleich zu statischen QR-Codes hier zwar geringer, ist der Planungs- und Vorbereitungsaufwand jedoch wesentlich größer. Vorteil ist allerdings, dass die anzuzeigenden Treffer individueller bestimmt und eingrenzbar sind. Für Bibliotheken, die bereits regal(boden)genaue Standortinformationen zu ihren Beständen im Lesesaal vorhalten, ist jedoch auch die Umsetzung statischer QR-Codes ohne große Vorarbeiten möglich.

Probleme haben sich unter anderen bei der Ermittlung vorgemerakter Bände ergeben. Hier wäre eine Erweiterung dahingehend denkbar, dass für RVK-Ranges Abfragen ermöglicht werden. Diesbezüglich wäre zu klären, ob die API zur Abfrage der entliehenen Bände entsprechend angepasst werden könne oder ob sämtliche RVK-Notation innerhalb eines bestimmten Bereichs ermittelt werden könnten.

Sorgfalt gefragt ist auch bei der Auswahl der zu integrierenden Medientypen. Bereits bei den konzeptionellen Überlegungen zur Integration von E-Journals zeigten sich zahlreiche Probleme. Im Konzept wurde von der Integration von elektronischen Aufsätzen abgeraten, da Discovery-Systeme hierfür die geeigneteren Werkzeuge sind. Schließlich soll eine App kein Multifunktionsstool darstellen, sondern ein Spezialwerkzeug für ausgewählte Funktionen.

Bereits bei der Entwicklung des Konzepts stellte sich darüber hinaus, dass für die Umsetzung zahlreicher Funktionen entsprechende Schnittstellen benötigt werden, über die man an die notwendigen Daten gelangen kann. Dies zeigte sich auch bei der Umsetzung der Integration entliehener Werke sowie der Implementation der Vormerkfunktion. Dies verdeutlicht einmal mehr, wie wichtig die Offenheit der neuen Generation der Bibliothekssysteme ist.

Bei der Integration der Durchschaltung zum E-Book zeigte sich leider, dass diese nur für zwei E-Book-Anbieter (SpringerLink und De Gruyter) umgesetzt werden konnte. Zwar stellen durchaus mehr Anbieter ihre E-Books als Volltexte im PDF-Format bereit, doch wird durch die Erzeugung von individuellen Token verhindert, dass ein Zugriff direkt auf den Volltext (ohne Umweg über die Anbieterseite) möglich ist.

## 8.4 Ausblick

Gerade durch den immensen Pflege- oder Planungs- und Einführungsaufwand, der mit der Standortermittlung der Nutzer durch die Nutzung von QR-Codes verbunden ist, wäre es ideal, hierfür langfristig Möglichkeiten der Texterkennung zu nutzen. So könnte über das Einlesen von Signaturetiketten, die am Buchrücken angebracht sind, automatisch anhand der Signatur die RVK-Notation gelesen werden.

Als weitere Funktion wäre eine Blätterfunktion zwischen verschiedenen RVK-Notationen denkbar. Nach Einscannen eines QR-Codes könnten so (ohne erneutes Einlesen eines neuen QR-Codes) die Treffer zu den benachbarten RVK-Notationen angezeigt werden.

Um lange Wartezeiten für den Nutzer auch bei großen Treffermengen zu vermeiden, ließe sich die Möglichkeit nutzen, zunächst immer zehn Treffer anzuzeigen und dann die nächsten Treffer nachzuladen, indem die nächste Anfrage an die API gestellt wird.

Wichtig wäre auch eine entsprechende Bewerbung der Anwendung. So könnte bspw. ein Tutorial für den Bibliotheksroboter Wilma erstellt werden, der die Funktion der App vorstellt. Um auch weiterhin alle Services der Hochschule zusammen zu präsentieren, würde es sich darüber hinaus anbieten, die Anwendung in die zentrale Hochschul-App Unidos zu integrieren.

Zusammenfassend kann gesagt werden, dass sich aufbauend auf dem bestehenden Prototyp zahlreiche spannende weitere Projekte für Erweiterungen herleiten lassen.

# Literaturverzeichnis

- Ackermann, C.** (2011): Die E-Books am Regal finden. Online verfügbar unter: [http://www.opus-bayern.de/bib-info/volltexte//2011/998/pdf/E\\_Books\\_per\\_QR.pdf](http://www.opus-bayern.de/bib-info/volltexte//2011/998/pdf/E_Books_per_QR.pdf).
- Arocom** (n. N.): Fachbegriffe einfach erklärt : Node.js. Online verfügbar unter: <https://www.arocom.de/fachbegriffe/webentwicklung/nodejs> [Zuletzt geprüft am: 03.07.2021].
- ARTEC** (n. N.): V:SCOUT: Die innovative und visuelle Orientierungshilfe für Ihre Besucher & Mitarbeiter. Online verfügbar unter: <https://vscout.biz/> [Zuletzt geprüft am: 20.06.2021].
- Attia, O.** (2020a): Entwicklung eines Recommender-Systems für E-Book-Empfehlungen in der Hochschulbibliothek der TH Wildau. Wildau.
- Attia, O.** (2020b): Praktikumsbericht. Wildau.
- Azmitia, A.; Mohnke, J. und Wiechers, H.** (2016): Campus App Unidos Wildau - ein ständiger Begleiter für den Alltag an der Technischen Hochschule Wildau. In: *Barton, T. et al. (Hg.): Campus App Unidos Wildau - ein ständiger Begleiter für den Alltag an der Technischen Hochschule Wildau.* Wiesbaden: Springer, S. 227–242.
- Azmitia, A.; Mohnke, J. und Wiechers, H.** (2016): Wo bin ich? iBeacons im Einsatz in der Bibliothek der TH Wildau. In: *Bibliothek - Forschung und Praxis*, Jg. 40, Nr. 3, S. 395–404.
- Balzert, H.** (2009): Lehrbuch der Software-Technik. Basiskonzepte und Requirements Engineering. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag.
- Bauer, C.** (2016): NFC: Reichweite und Geschwindigkeit. Online verfügbar unter: [https://praxistipps.chip.de/nfc-reichweite-und-geschwindigkeit\\_48444](https://praxistipps.chip.de/nfc-reichweite-und-geschwindigkeit_48444) [Zuletzt geprüft am: 27.06.2021].
- Bayreuther, I. und Peisl, B.** (2012): Regeln für die Signaturvergabe der UB Regensburg. Online verfügbar unter: [https://rvk.uni-regensburg.de/images/stories/fruit/sigreg/regeln\\_fr\\_die\\_signaturvergabe\\_version\\_2012\\_02\\_17.pdf](https://rvk.uni-regensburg.de/images/stories/fruit/sigreg/regeln_fr_die_signaturvergabe_version_2012_02_17.pdf) [Zuletzt geprüft am: 14.08.2021].
- Bibliographisches Institut** (2021): Duden: E-Book. Online verfügbar unter: [https://www.duden.de/rechtschreibung/E\\_Book](https://www.duden.de/rechtschreibung/E_Book) [Zuletzt geprüft am: 17.08.2021].
- Böhner, V.** (2019): Einführung in Vue.js: Was steckt wirklich hinter dem progressiven Framework? Online verfügbar unter: <https://www.hosteurope.de/blog/einfuehrung-in-vue-js-was-steckt-wirklich-hinter-dem-progressiven-framework/> [Zuletzt geprüft am: 03.07.2021].
- ComputerWeekly.de (Redaktion) und TechTarget** (n. N.): Definition Apache Solr. Online verfügbar unter: <https://www.computerweekly.com/de/definition/Apache-Solr> [Zuletzt geprüft am: 13.08.2021].

- Deeg, C.** (2018): Die nächste digitale Welle. Apps und Bibliotheken: Denken wir mal wieder neu. In: *BuB*, Jg. 70, Nr. 05, S. 248–250.
- dejure.org Rechtsinformationssysteme GmbH** (n. N.): Datenschutz-Grundverordnung : Artikel 14 Informationspflicht, wenn die personenbezogenen Daten nicht bei der betroffenen Person erhoben wurden. Online verfügbar unter: <https://dejure.org/gesetze/DSGVO/14.html> [Zuletzt geprüft am: 28.07.2021].
- Deutscher Bibliotheksverband e.V.** (n. N.): Bibliothek des Jahres 2012. Bibliothek der Technischen Hochschule Wildau (FH). Online verfügbar unter: <https://www.bibliotheksverband.de/dbv/auszeichnungen/bibliothek-des-jahres/preistraeger/2012.html> [Zuletzt geprüft am: 13.02.2021].
- Diaz, M.** (2013): Genesis Full Height Sidebar with Header above Content. Online verfügbar unter: <https://amethystwebsitedesign.com/genesis-full-height-sidebar-with-header-above-content/> [Zuletzt geprüft am 08.08.2021].
- DNB** (n. N.): DDC-Übersichten. Online verfügbar unter: [https://www.dnb.de/DE/Professionell/DDC-Deutsch/DDCuebersichten/ddcUebersichten.html#:~:text=Die%20Dewey%2DDezimalklassifikation%20\(DDC\),in%20die%20Grundstruktur%20der%20DDC%20.](https://www.dnb.de/DE/Professionell/DDC-Deutsch/DDCuebersichten/ddcUebersichten.html#:~:text=Die%20Dewey%2DDezimalklassifikation%20(DDC),in%20die%20Grundstruktur%20der%20DDC%20.) [Zuletzt geprüft am: 27.06.2021].
- Ekz.bibliotheksservice GmbH** (2020): Bibliotheksausstattung 2020/2021 : Alles für ein perfektes Ambiente. Reutlingen.
- Frontend GmbH** (n. N.): Was ist Frontend - was Backend? Online verfügbar unter: <https://www.frontend-gmbh.de/blog/frontend-und-backend/> [Zuletzt geprüft am: 16.05.2021].
- Fuchs Media Solutions:** SEO Lexikon: jQuery Begriffserklärung und Definition. Online verfügbar unter: <https://www.seo-analyse.com/seo-lexikon/j/jquery/> [Zuletzt geprüft am: 13.08.2021].
- Gantert, K. und Hacker, R.** (2008): Bibliothekarisches Grundwissen. 8., vollständig neu bearbeitete und erweiterte Auflage. München: K. G. Saur Verlag.
- Goltz, J.** (2015): Mobile Applikationen für Bibliotheken im deutschsprachigen Raum. Berlin: Konrad-Zuse-Zentrum für Informationstechnik Berlin.
- Google** (n. N.): Material Design - Introduction. Online verfügbar unter: <https://material.io/design/introduction> [Zuletzt geprüft am: 02.07.2021].
- Google Developers** (n. N.): Google Books API : Integrate with the Google Books repository. Online verfügbar unter: <https://developers.google.com/books> [Zuletzt geprüft am: 27.07.2021].
- Grande, M.** (2011): 100 Minuten für Anforderungsmanagement. Kompaktes Wissen nicht nur für Projektleiter und Entwickler. 1. Auflage. Wiesbaden: Springer.
- Grätz, C.** (2020a): Bibliothekarische Dienstleistungen für die Überwindung des Medienbruchs von physischen und elektronischen Medien. Wildau.

- Grätz, C.** (2020b): Praktikumsbericht iLibrary. E-Books To Go. Wildau.
- hbz** (n. N.): Deutsche Bibliotheksstatistik: variable Auswertung. Online verfügbar unter: <https://www.bibliotheksstatistik.de/vaAttribute> [Zuletzt geprüft am: 16.07.2021].
- Head, A. J. und Eisenberg, M. B.** (2009): Lessons Learned: How College Students Seek Information in the Digital Age. Project Information Literacy Progress Report. Online verfügbar unter: [http://ctl.yale.edu/sites/default/files/basic-page-supplementary-materials-files/how\\_students\\_seek\\_information\\_in\\_the\\_digital\\_age.pdf](http://ctl.yale.edu/sites/default/files/basic-page-supplementary-materials-files/how_students_seek_information_in_the_digital_age.pdf) [Zuletzt geprüft am: 03.05.2021].
- Hill, J.** (2021): Was die NoSQL-Datenbank kann : MongoDB FAQ. Online verfügbar unter: <https://www.computerwoche.de/a/was-die-nosql-datenbank-kann,3548993> [Zuletzt geprüft am: 28.07.2021].
- Högele, T.** (2018): Warum es der Kreativität schadet, wenn Universitätsbibliotheken ihre Bücherregale abschaffen. Online verfügbar unter: <https://ze.tt/universitaetsbibliotheken-in-denen-man-nicht-nach-buechern-stoebnern-kann-sind-tot/> [Zuletzt geprüft am: 03.05.2021].
- Hommes, K. P.** (2009): Vandalen haben keine Chance mehr. Veränderungen der Arbeitsorganisation und des Anforderungsprofils durch E-Medien. In: *BuB*, Jg. 61, Nr. 09, S. 620–623.
- hotframeworks** (n.N.): Find your new favourite web framework. Online verfügbar unter: <https://hotframeworks.com/> [Zuletzt geprüft am: 27.06.2021].
- IONOS** (2019): Alles rund um E-Books - Teil 3: E-Book-Formate. Online verfügbar unter: <https://www.ionos.de/digitalguide/online-marketing/verkaufen-im-internet/e-book-format/> [Zuletzt geprüft am: 27.06.2019].
- IT-Talents/Jesko** (2020): Was ist ein Framework? - Definition und Erklärung. Online verfügbar unter: <https://www.it-talents.de/blog/it-talents/frameworks> [Zuletzt geprüft am: 16.05.2021].
- Kern, C.** (2011): RFID für Bibliotheken. Heidelberg u.a.: Springer.
- Kleiner, E.** (2013): Blended Shelf. Ein realitätsbasierter Ansatz zur Präsentation und Exploration von Bibliotheksbeständen. Konstanz.
- Krautz, S.** (2014): Einflüsse von NFC-Smartphones auf das RFID-Bibliothekssystem. Eine Analyse des Bedrohungspotentials durch NFC-Smartphones und Beschreibung von möglichen Gegenmaßnahmen. In: *Seeliger, F. et al. (Hg.): Einflüsse von NFC-Smartphones auf das RFID-Bibliothekssystem. Eine Analyse des Bedrohungspotentials durch NFC-Smartphones und Beschreibung von möglichen Gegenmaßnahmen.* 3. Berlin: News & Media, S. 209–221.
- Landwehr, J.** (2020): Was ist ein Framework? - Definition und Erklärung. Online verfügbar unter: <https://it-talents.de/it-wissen/framework/> [Zuletzt geprüft am: 27.06.2021].
- mapongo** (n. N.): mapongo Gebäudeinformationssysteme. Online verfügbar unter: <http://www.mapongo.de/libraries> [Zuletzt geprüft am: 20.06.2021].

- Mittler, E.** (1996): Die Rolle der Bibliothek. Online verfügbar unter: <http://www.imn.htwk-leipzig.de/~bastian/unendl/mittler.txt> [Zuletzt geprüft am: 19.01.2021].
- Mozilla** (Mittwoch, 28. Juli 2021): Struktur in die Webseite bringen. Online verfügbar unter: [https://developer.mozilla.org/de/docs/Learn/HTML/Introduction\\_to\\_HTML/Document\\_and\\_website\\_structure](https://developer.mozilla.org/de/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure) [Zuletzt geprüft am: 28.07.2021].
- Mumenthaler, R.** (2012): E-Books als Katalysator. Veränderungsprozesse in wissenschaftlichen Bibliotheken - E-Books können tiefgreifende Umwälzungen auslösen. In: *Bibliothek - Forschung und Praxis*, Jg. 64, Nr. 9, S. 604–608.
- Mumenthaler, R.** (2018): E-Books. Grundlagen und Praxis. Online verfügbar unter: <https://ebooksgrundlagen.pressbooks.com/> [Zuletzt geprüft am: 18.06.2021].
- Niebisch, T.** (2013): Anforderungsmanagement in sieben Tagen. Der Weg vom Wunsch zur Konzeption. Berlin, Heidelberg: Springer-Verlag.
- Open Library** (2021): Open Library Covers API. Online verfügbar unter: <https://open-library.org/dev/docs/api/covers> [Zuletzt geprüft am: 27.07.2021].
- Peterke, L.** (2019): Vue.js kurz & gut. Heidelberg: dunkt.verlag GmbH.
- Plappert, R.** (2016): RFID in Bibliotheken. Eine erste Bilanz und zukünftige Anwendungsoptionen. In: *Bibliotheksdienst*, Jg. 50, Nr. 6, S. 531–534.
- Reiner, U.** (2007): Automatic Analysis of Dewey Decimal Classification Notations. Online verfügbar unter: <https://monarch.qucosa.de/api/qucosa%3A18781/attachment/ATT-0/> [Zuletzt geprüft am: 25.07.2021].
- Reinman, A.** (n. N.): Nodemailer. Online verfügbar unter: <https://nodemailer.com/about/> [Zuletzt geprüft am: 27.07.2021].
- Reiterer, H.; Rädle, R.; Buscher, S. und Müller J.** (2016): Blended Library – neue Zugangswege zu den Inhalten wissenschaftlicher und öffentlicher Bibliotheken. In: *Bibliothek - Forschung und Praxis*, Jg. 40, Nr. 1, S. 7-20.
- Rüter, C.** (2009): Elektronische Ressourcen im Geschäftsgang. Ein Praxisbeispiel an der Universitätsbibliothek der Humboldt-Universität. Berlin.
- Schlede, F.-M.; Bär, T. und Donner, A.** (2018): Was ist LDAP (Lightweight Directory Access Protocol)? Online verfügbar unter: <https://www.ip-insider.de/was-ist-ldap-lightweight-directory-access-protocol-a-581204/> [Zuletzt geprüft am: 13.08.2021].
- Schubert, D.**: DOI (Digital Object Identifier). Online verfügbar unter: <https://www.wiwi.uni-wuerzburg.de/fileadmin/12020700/DOI.pdf> [Zuletzt geprüft am: 22.08.2021].
- Slama, C.** (2012): QR-Code-Hinweise in der Lehrbuchsammlung. Online verfügbar unter: <https://www.tuwien.at/tu-wien/aktuelles/news/news/qr-code-hinweise-in-der-lehrbuchsammlung> [Zuletzt geprüft am: 20.06.2021].



- SoftSelect:** SoftSelect Glossar: Backend. Online verfügbar unter: <http://www.softselect.de/business-software-glossar/backend> [Zuletzt geprüft am: 13.08.2021].
- Springer, S.** (2018): Node.js. Das umfassende Handbuch. 3., aktualisierte und erweiterte Auflage. Bonn: Rheinwerk Verlag.
- Steimle, T. und Wallach, D.** (n. N.): Proto Persona. Online verfügbar unter: <http://www.collaborative-uxdesign.com/scoping/protopersonas> [Zuletzt geprüft am: 03.07.2021].
- Stifterverband** (2019): Bibliotheksroboter Wilma: Hochschulperle des Monats Februar geht nach Wildau in Brandenburg. Online verfügbar unter: [https://www.stifterverband.org/pressemitteilungen/2019\\_02\\_27\\_hochschulperle\\_bibliotheksroboter\\_wilma](https://www.stifterverband.org/pressemitteilungen/2019_02_27_hochschulperle_bibliotheksroboter_wilma) [Zuletzt geprüft am: 13.02.2021].
- t2informatik** (n. N.): Personas: Die Entwicklung und kontinuierliche Verwendung. Online verfügbar unter: <https://t2informatik.de/wissen-kompakt/personas/> [Zuletzt geprüft am: 03.07.2021].
- Taubert, J.** (2012): Absentia in Praesentia? Zur Präsentation und Vermittlung digitaler Medien im physischen Raum. Berlin.
- Taubert, J.** (2016): Absentia in Praesentia? Absentia in Praesentia? Zur Präsentation von digitalen Beständen im physischen Raum. Leipzig.
- TH Wildau** (n. N.): Web-App „THWi Kontaktnachverfolgung“. Online verfügbar unter: <https://www.th-wildau.de/studieren-weiterbilden/neuigkeiten-und-veranstaltungen/kontaktnachverfolgung/> [Zuletzt geprüft am: 01.07.2021].
- TH Wildau** (2019): Studiengang "Bibliotheksinformatik"; Master of Science; Modulkatalog. Online verfügbar unter: [https://www.wit-wildau.de/wp-content/uploads/2019/03/Modulhandbuch\\_Bibliotheksinformatik.pdf](https://www.wit-wildau.de/wp-content/uploads/2019/03/Modulhandbuch_Bibliotheksinformatik.pdf) [Zuletzt geprüft am 19.08.2021].
- TH Wildau** (2021): Hochschulbibliothek. Online verfügbar unter: <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/> [Zuletzt geprüft am: 13.02.2021].
- TH Wildau | Bibliothek** (n. N.): E-Books. Online verfügbar unter: <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/recherche/e-books/> [Zuletzt geprüft am: 03.08.2021].
- TH Wildau | Bibliothek** (n. N.): WILBERT — Wildauer Bücher+E-Medien Recherche-Tool. Online verfügbar unter: <https://wilbert.kobv.de/?plv=2> [Zuletzt geprüft am: 13.02.2021].
- TH Wildau | Bibliothek** (n. N.): Wilbert/Onlinekatalog. Online verfügbar unter: <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/recherche/wilbert/> [Zuletzt geprüft am: 08.08.2021].
- TH Wildau | Bibliothek** (2021): Projekte. Projekte der Hochschulbibliothek : RFID. Online verfügbar unter: <https://www.th-wildau.de/hochschule/zentrale-einrichtungen/hochschulbibliothek/ueber-die-bibliothek/projekte/> [Zuletzt geprüft am: 13.02.2021].

- TH Wildau | Hochschulbibliothek** (n. N.): Bewerbung zur Bibliothek des Jahres 2019 : Praktische Anwendungen. Online verfügbar unter: <https://icampus.th-wildau.de/bewerbung-bdj-2019/praktische-anwendungen.html> [Zuletzt geprüft am: 08.08.2021].
- TH Wildau | iCampus** (n. N.): iCampus Wildau. Online verfügbar unter: <https://icampus.th-wildau.de/icampus/> [Zuletzt geprüft am: 05.05.2021].
- Universität Bern** (n. N.): cUBe · Online-Informationen erkunden. Online verfügbar unter: [https://www.ub.unibe.ch/ub/cube/index\\_ger.html](https://www.ub.unibe.ch/ub/cube/index_ger.html) [Zuletzt geprüft am: 20.06.2021].
- Universität Konstanz** (n. N.): Blended Library. Online verfügbar unter: <https://hci.uni-konstanz.de/forschung/forschungsprojekte/abgeschlossene-projekte/blended-library/> [Zuletzt geprüft am: 19.06.2021].
- Universitätsbibliothek Bayreuth** (n. N.): Neu im Katalog: Anzeige von QR-Codes. Online verfügbar unter: [https://www.ub.uni-bayreuth.de/de/news\\_archiv/neuerungen-IG-6/index.html](https://www.ub.uni-bayreuth.de/de/news_archiv/neuerungen-IG-6/index.html) [Zuletzt geprüft am: 20.06.2021].
- Universitätsbibliothek Regensburg** (n. N.): RVK Online. Online verfügbar unter: <https://rvk.uni-regensburg.de/regensburger-verbundklassifikation-online> [Zuletzt geprüft am: 27.06.2021].
- Universitätsbibliothek Regensburg** (n. N.): Was ist die RVK? Online verfügbar unter: <https://rvk.uni-regensburg.de/2-uncategorised/141-rvk> [Zuletzt geprüft am: 27.06.2021].
- Universitätsbibliothek Regensburg** (2019): Über 50 Jahre RVK : Näheres zum Werdegang. Online verfügbar unter: <https://rvk.uni-regensburg.de/44-ueber-die-rvk-regensburger-verbundklassifikation/geschichte-rvk-regensburger-verbundklassifikation-historie-entwicklung/57-werdegang-der-rvk> [Zuletzt geprüft am: 25.07.2021].
- Verbundzentrale des Gemeinsamen Bibliotheksverbundes (GBV)** (n.N.): Cocoda Mapping Tool. Online verfügbar unter: <https://coli-conc.gbv.de/cocoda/app/?from-Scheme=http%3A%2F%2Furi.gbv.de%2Fterminology%2Frvk%2F&to-Scheme=http%3A%2F%2Fdewey.info%2Fscheme%2Fedition%2Fe23%2F> [Zuletzt geprüft am 25.07.2021].
- Voß, V.** (2011): Mobile Tagging, QR-Codes und ihre Anwendungsmöglichkeiten. In: *B.I.T.online*, Jg. 14, Nr. 4, S. 347–353.
- Voß, V.** (2011): QR-Codes im bibliothekarischen Einsatz. Online verfügbar unter: <https://net-bib.hypotheses.org/78620092>.
- Vue.js** (n. N.): Introduction. What is vue.js? Online verfügbar unter: <https://vuejs.org/v2/guide/> [Zuletzt geprüft am: 16.05.2021].
- Vuetify** (n. N.): Introduction: What is Vuetify. Online verfügbar unter: <https://vuetifyjs.com/en/introduction/why-vuetify/> [Zuletzt geprüft am: 03.07.2021].

# Abbildungsverzeichnis

Abbildung 1: Nutzung von Bibliotheken für die Suche nach seminarrelevanter Literatur .....	6
Abbildung 2: Nutzung der Recherchemittel für die Suche nach weiterführender Literatur .....	6
Abbildung 3: Blended Shelf im Katalog der Universität Konstanz.....	7
Abbildung 4: Beliebtheit von Web-Frameworks.....	21
Abbildung 5: Standortanzeige gedruckter Medien in Wilbert.....	26
Abbildung 6: Ansicht der unidos-App für den Bereich "Bibliothek" .....	28
Abbildung 7: Startseite der App "E-Books To Go" (Desktop).....	30
Abbildung 8: Startseite der App "E-Books To Go" (Smartphone).....	31
Abbildung 9: Anzeige der Trefferliste nach Scannen eines QR-Codes (Desktop).....	32
Abbildung 10: Detailansicht E-Book .....	32
Abbildung 11: E-Book-Empfehlungen zu einzelnen Modulen des eigenen Studiengangs.....	33
Abbildung 12: Architektur des bestehenden Prototyps .....	35
Abbildung 13: Struktur des Frontends in der IDE Visual Studio Code .....	37
Abbildung 14: Gesamtarchitektur des Systems .....	49
Abbildung 15: Navigation im Prototyp der App E-Books to go .....	50
Abbildung 16: Mockup Menü mit QR-Template.....	52
Abbildung 17: Ablauf Ermittlung Regalstandort Eigene Abbildung.....	53
Abbildung 18: Abbildung 21: Beschriftung der Regale mit Systematikstellen in Wildau.....	55
Abbildung 19: Schildwackler Flexi .....	55
Abbildung 20: Beschriftungswürfel.....	55
Abbildung 21: Abfrage RVK-DDC-Mappings vom Cocoda Mapping Tool .....	60
Abbildung 22: Seitenaufbau einer Webseite.....	67
Abbildung 23: Mockup zur Einzelansicht eines E-Books .....	71
Abbildung 24: Anmeldefenster des Prototyps der App.....	73
Abbildung 25: Mockup zur Bestätigung des Vormerk-Wunsches im Popup-Fenster.....	74
Abbildung 26: Literaturhinweise zum Modul Programmierung des Masterstudiengangs .....	75
Abbildung 27: Mockup zu Literaturempfehlungen zum eigenen Studiengang .....	76
Abbildung 28: Mockup für Versenden einer E-Mail-Empfehlung .....	77
Abbildung 29: Mockup für Versenden einer Buchempfehlung per E-Mail.....	78
Abbildung 30: Mockup für Autovervollständigung .....	78
Abbildung 31: Mockup für Anzeige der Empfehlung aus E-Mail in der App selbst.....	79
Abbildung 32: Größenvergleich verschiedener aus Word erzeugter QR-Codes.....	90
Abbildung 33: Erstellung einer QR-Code-Vorlage mit dem Avery-Tool .....	91
Abbildung 34: Screenshot der im Hintergrund ablaufenden Prozesse beim Download eines E-Books des Anbieters Taylor & Francis .....	95
Abbildung 35: Anzeige entliehener Bände als Liste .....	103
Abbildung 36: Hinweis, wenn keine entliehenen Bände gefunden wurden.....	103
Abbildung 37: Detailansicht der entliehenen Bände .....	104
Abbildung 38: Anmeldehinweis für Aufgabe einer Vormerkung.....	104

Abbildung 39: Popup zum Aufgeben einer Vormerkung .....	105
Abbildung 40: Information des Benutzers über eine erfolgreiche Vormerkung .....	108

# Listingverzeichnis

Listing 1: Antwort der Solr-API (Ausschnitt).....	58
Listing 2: Antwort der Cocoda-API (Ausschnitt).....	61
Listing 3: Antwort der API für entlehene Bände (Ausschnitt).....	66
Listing 4: Antwort der Google-Books-API .....	69
Listing 8: Umsetzung der Weiterleitung zum Volltext-PDF bei De Gruyter .....	92
Listing 10: Umsetzung der Weiterleitung zum Volltext-PDF bei Springer Link.....	95
Listing 11: Umsetzung der Weiterleitung zum Volltext-PDF bei WISO .....	96
Listing 12: Frontend-Funktion link zum Öffnen der E-Book-Volltexte .....	97
Listing 13: Prüfung der DOIs und Ersetzung der Volltext-URLs im Backend.....	98
Listing 14: Antwort der API für entlehene Bände im JSON-Format (Ausschnitt).....	99
Listing 15: Sortierung der Ergebnisse der API für entlehene Bände .....	101
Listing 16: Backend-Endpunkt zur Bereitstellung entliehener Bände.....	103
Listing 17: Backend-Endpunkt zum Aufgeben einer Vormerkung .....	106
Listing 18: Antwort der Vormerkungs-API im JSON-Format.....	107
Listing 19: Verarbeitung der Informationen der Vormerk-API im Frontend.....	107

# Tabellenverzeichnis

Tabelle 1: Mobile Tagging - Übersicht.....	14
Tabelle 2: Vergleich verschiedener App-Typen.....	20
Tabelle 3: Vergleich von Web-Frameworks.....	22
Tabelle 4: Funktionale Anforderungen.....	47
Tabelle 5: Nicht-funktionale Anforderungen.....	47
Tabelle 6 : Prüfung der Durchschaltungsmöglichkeiten zu Volltexten der verschiedenen Anbieter - Übersicht .....	94

# Glossar

## API

Eine API (kurz für Application Programming Interface) ist eine Programmierschnittstelle, die den Austausch von Daten zwischen einzelnen Programm(teil)en vereinheitlicht.

## Axios

Bei Axios handelt es sich um eine JavaScript-Bibliothek, welches es ermöglicht, HTTP-Anforderungen zu senden und die Antworten zu verarbeiten. Axios unterstützt sowohl server- als auch clientseitiges JavaScript.

## Backend

Als Backend bezeichnet man „den Teil eines IT-Systems, der sich mit der Datenverarbeitung im Hintergrund beschäftigt [...]“<sup>228</sup>

## Discovery-System

Im Gegensatz zum konventionellen OPAC (Online Public Access Catalogue) sind in Discovery-Systemen als One-Stop-Shops auch Aufsätze aus bspw. aus Fachdatenbanken recherchierbar. Auf Grundlage vergleichsweise einfacher Suchanfragen werden nach Ranking gelistete Trefferlisten erzeugt. Die Treffer erfüllen die Suchanfragen hier nur zum Teil.

## Framework

Ein Framework stellt eine Art „Programmiergerüst“ für Entwickler dar, welches „die Struktur und Architektur des künftigen Softwareprodukts bestimmt.“<sup>229</sup> Es enthält verschiedene Tools, mit denen sich „routinemäßige, sich wiederholende Aufgaben lösen“<sup>230</sup> lassen.

## Frontend

Bei einem Frontend handelt es sich um die Präsentationsebene einer Anwendung, also „den Teil einer Webapplikation, den der Betrachter sehen kann, zum Beispiel in Form von einer grafischen Benutzeroberfläche“.<sup>231</sup>

---

<sup>228</sup> Vgl. SoftSelect: SoftSelect Glossar: Backend, <http://www.softselect.de/business-software-glossar/backend>, 13.08.2021.

<sup>229</sup> Vgl. IT-Talents/Jesko (2020): Was ist ein Framework? - Definition und Erklärung, <https://www.it-talents.de/blog/it-talents/frameworks>, 16.05.2021.

<sup>230</sup> Ebd.

<sup>231</sup> Vgl. Frontend GmbH (n. N.): Was ist Frontend - was Backend?, <https://www.frontend-gmbh.de/blog/frontend-und-backend/>, 16.05.2021.

## **E-Book**

Unter E-Books werden elektronisch publizierte Bücher verstanden. „E-Books zeichnen sich in der Regel dadurch aus, dass sie wie ein gedrucktes Buch einen festen Textbestand anbieten, der – bis zu einer neuen Ausgabe – nicht mehr verändert oder aktualisiert wird, sie lassen sich daher auch wie gedruckte Bücher zitieren.“<sup>232</sup>

## **Get-Request/Get-Parameter**

Ein Get-Request ist eine HTTP-Methode zur Abfrage von Daten. Mit Hilfe von über die URL mitgesendeten GET-Parametern wird die Anfrage spezifiziert. Im Gegensatz dazu werden bspw. POST-Requests zum Senden von Daten genutzt.

## **JQuery**

JQuery ist eine frei zugängliche JavaScript-Bibliothek, welche „umfangreiche und komfortable Funktionen [bietet], mit deren Verwendung sich Inhaltselemente innerhalb des Document-Object-Models (DOM) manipulieren lassen.“<sup>233</sup>

## **LDAP**

LDAP (Lightweight Directory Access Protocol) ist ein „Netzwerkprotokoll zur Durchführung von Abfragen und Änderungen in einem verteilten Verzeichnisdienst.“<sup>234</sup>

## **OPAC**

Der Begriff OPAC (Online Public Access Catalogue) bezeichnet den konventionellen bibliothekarischen Online-Katalog, der eine Recherche nach Beständen einer Einrichtung ermöglicht. Gesucht werden kann nach Monographien, Reihen- oder Zeitschriftentiteln, nicht aber nach Aufsätzen (siehe Discovery-System). Abfragen werden unter Verwendung boolescher Operatoren erstellt. Die Treffer erfüllen die Suchanfrage hier exakt.

## **Precision**

Precision beschreibt den Anteil der Treffermenge, der relevant in Bezug auf das in der Suchanfrage formulierte Informationsbedürfnis ist. Je größer der Anteil relevanter Dokumente im Suchergebnis ist, desto größer die Precision.

## **Schlagwort**

„Ein Schlagwort [...] ist die möglichst genaue und vollständige [der deutschen Sprache entnommenen] Bezeichnung für den sachlichen Inhalt eines Dokuments, der – terminologisch kontrolliert – für die Erschließung und Recherche verwendet wird.“<sup>235</sup>

---

<sup>232</sup> Vgl. Gantert; Hacker (2008), S. 111.

<sup>233</sup> Vgl. Fuchs Media Solutions: SEO Lexikon: jQuery Begriffserklärung und Definition, <https://www.seo-analyse.com/seo-lexikon/j/jquery/>, 13.08.2021.

<sup>234</sup> Vgl. Schlede; Bär; Donner (2018): Was ist LDAP (Lightweight Directory Access Protocol)?, <https://www.ip-insider.de/was-ist-ldap-lightweight-directory-access-protocol-a-581204/>, 13.08.2021.

<sup>235</sup> Vgl. Gantert; Hacker (2008), S. 177.



## Solr

„Apache Solr ist eine Open-Source-Suchplattform auf Basis der Java-Bibliothek Apache Lucene, die alle Kernelemente einer Suchmaschine bereitstellt.“<sup>236</sup>

---

<sup>236</sup> Vgl. ComputerWeekly.de (Redaktion); TechTarget (n. N.): Definition Apache Solr, <https://www.computerweekly.com/de/definition/Apache-Solr>, 13.08.2021.

# A. Anhang

## A.1 Ergebnisse des Get Togethers am 13.11.2020

### Ergebnisse des Get Togethers zu E-Books-To-Go am 13.11.2020, 14.00 Uhr (bis 15.30 Uhr)

#### Teilnehmer/innen:

- Frank Seeliger
- Petra Keidel
- Janett Mohnke
- Christopher Grätz
- Omar Attia
- Dana Geier

#### Ist-Stand der App:

- An-/Abmeldung in E-Books-To-Go
- Anzeige der E-Books, die sich im Bestand der Bibliothek der TH Wildau befinden
  - o Standortermittlung über QR-Codes (auf Regalebene) umgesetzt für den Themenbereich Informatik
  - o Abruf der E-Books über Wilbert,
- Anzeige von Literaturempfehlungen auf Grundlage der belegten/zu belegenden Kurse (Datengrundlage ist hier das LDAP)
- Recommender-System: Möglichkeit des Versendens von Empfehlungen per E-Mail
- Projekt liegt auf iCampus-Git

#### Anstehend:

- Umzug der 3D-Visualisierung und E-Book-To-Go-App vom Development-Server auf live Server

#### Angesprochene Probleme:

- derzeit sind lediglich 50%<sup>237</sup> des E-Book-Bestandes mit einer RVK-Notation versehen
  - o wo können Daten anderweitig gezogen werden?
  - o Datengrundlage der Verlage sehr problematisch
- hohe Dynamik der Bestandsaufstellung der Hochschulbibliothek
- Verbindung zwischen Printmedien und E-Books in Nachweissystemen noch nicht so nachvollzogen -> Struktur in Wilbert nicht erkennbar
- derzeit sind E-Books mit RVK-Notation in Datenbank gespeichert:
  - o Problem: Datenpflege, da momentan pro QR-Code ein Regalfach hinterlegt ist

---

<sup>237</sup> Korrektur: Hier müsste es korrekt heißen: 30%, leider ist im Protokoll ein Tippfehler enthalten.

## Zukünftige Arbeitspakete

- (1) Möglichkeit, E-Books direkt via Link auf dem Smartphone öffnen zu können (zusätzlich zu Metadaten sollte auch Direktlink aufs E-Book enthalten sein)
  - Direkter Download von E-Books nur von Springer möglich
  - Problematisch: E-Book Central -> separate Anmeldung notwendig
- (2) Datenbasis für E-Books (RVK-Notation nicht ausreichend)
- (3) Standortzuordnung: bspw. via IBeacons anstelle von QR-Code
- (4) Präsentation entliehener Werke
- (5) Weiterentwicklung des Recommendersystems:
  - Langfristige Speicherung von Empfehlungen in Datenbank
  - Quellen: Modulhandbuch, Semesterapparat, Empfehlungen durch andere Studierende (Modulhandbuch deckt sich nicht unbedingt mit Semesterapparat)
- (6) Scannen der Signatur/des Barcodes, um weitere thematisch passende Medien (E-Books etc) angezeigt zu bekommen, („weg vom QR-Code (Regal) hin zu Orientierung an Signatur“)
  - Thema „Bildverarbeitung in Webanwendungen“
  - Ideal wäre Signatur mit QR-Code hintendran, welcher auf Buchrücken eingescannt werden kann (aber auch hier wäre problematisch, dass nicht alle Bände auf dem Buchrücken eine Signatur haben)
  - Nicht zu vergessen: hohe Regaldynamik v.a. in den Bereichen Recht und Wirtschaft
- (7) Langfristige Möglichkeit der Realisierung der „fluiden“ Bibliothek (Bände haben keinen festen Standort mehr)
  - Hintergrund: RVK ursprünglich reine Aufstellungssystematik
  - Vorteil in App: Anzeige des E-Books in mehreren Fachbereichen möglich
  - auch Printbestand könnte völlig neu durcheinandergewürfelt werden
- (8) Ausleihzahlen als Empfehlungskriterium
  - Problem: nicht alle Lieferanten liefern statistische Informationen
  - Weitere Idee war Empfehlung auf Grundlage von Bewertungen: schwierig, da diese oftmals einfach nicht abgegeben werden; alternativ: Nutzung von Bewertungen anderer Anbieter, z.B. Springer: ebenfalls schwierig, da z.T. Werbung mit eingeschleust wird
- (9) Anzeige von ausgeliehenen Printmedien
- (10) Langfristiges Ziel: „Digitales Schmökern“: Ausgangspunkt ist nicht mehr Recherche, sondern Stöbern (Recommender-Empfehlungen, ...)
  - Zufallsfund
  - Darstellung der Räume wie in Gamingszene, sodass man diese erkennen kann und Services der Bibliothek der TH Wildau erkunden kann

## A.2 E-Mail an die Mailingliste InetBib

Liebe Kolleg/innen,

im Rahmen meiner Masterarbeit im Studiengang Bibliotheks-informatik beschäftige ich mich mit der Weiterentwicklung einer Anwendung, welche das Stöbern am Regal durch die Anzeige von thematisch passenden E-Books unterstützen soll.

Um einen Überblick über den Umgang mit der Problematik des Medienbruchs beim Stöbern am Regal zu erhalten, würde mich interessieren, ob es in Ihren Bibliotheken Möglichkeiten gibt, die (durch Ihre Einrichtung lizenzierten) E-Ressourcen auch beim Stöbern am Regal angezeigt zu bekommen, und wenn ja, welche das sind.

Ich würde mich sehr freuen, wenn Sie sich die Zeit nehmen würden und diese im folgenden Etherpad (mit Nennung Ihrer Einrichtung) bis Ende Mai kurz beschreiben könnten. Auch wenn es keine derartigen Services gibt, würde mich das interessieren.

Das Etherpad finden Sie unter:

[https://yopad.eu/p/E-Books\\_Stoebern\\_Bibliotheken-365days](https://yopad.eu/p/E-Books_Stoebern_Bibliotheken-365days)

Vielen Dank und beste Grüße

Dana Geier

(Systembibliothekarin, Studiengang Bibliotheks-informatik)

## A.3 Öffentliche Etherpad-Antworten zur Inetbib-Umfrage

Stand: 24.07.2021

---

Sehr geehrte Frau Geier,

in der Gemeindebücherei Grasbrunn haben wir in jedem Regal einen QR-Code, der auf die Onleihe-Seite als alternative Medienquelle verweist. Aber nicht zu einzelnen Titeln.

Mit freundlichen Grüßen

Sabine Dunker

---

Liebe Frau Geier,

wir verwenden in Düsseldorf ebenfalls QR-Codes, die wir incl. Titelfinfos auf ein Art Buchstütze aufgeklebt haben. Gestartet sind wir ca. Ende 2019 bei den ca. 40 Lehrbuchsammlungstiteln der naturwissenschaftlichen Fächer, die gedruckt und elektronisch vorhanden sind. Es war als Test geplant und wir wollten die Nutzung zunächst evaluieren. Dann hat uns aber Corona mit Bibliotheksschließungen und kaum Studierenden auf dem Campus ein Strich durch die Rechnung gemacht. Ob wir das Projekt noch weiter verfolgen, ist noch nicht entschieden.

Viele Grüße

Ina Gedrath (ULB Düsseldorf)

---

Liebe Frau Geier,

im Jahr 2012 hatten wir einmal ein kleines Projekt zur Sichtbarmachung von E-Journals für einen Praktikanten aus dem höheren Bibliotheksdienst entwickelt. Für die bei uns sowohl in Print als auch elektronisch vorhandenen Zeitschriften wurden mithilfe eines Makros QR-Codes generiert. Schildchen mit diesen QR-Codes wurden dann an den Klappen der jeweiligen Auslage angebracht (also im Grunde die gleiche Methode wie von Frau Poeszus schon erwähnt). Leider haben wir nicht erfahren, ob dieser Service groß genutzt wurde.

Ich selbst habe auch immer mal wieder solche QR-Codes in Print-Ausstellungsregale integriert: um neben attraktiven neuen Print-Exemplaren zu einem bestimmten Thema auch auf aktuelle Online-Ressourcen hinzuweisen. Die per Makro erstellten QR-Codes wurden in eine kleine Liste mit solchen interessanten E-Books integriert, diese dann auf leichtem Karton ausgedruckt, laminiert und im Ausstellungsregal plaziert.

Hinweise auf E-Books in unserem Freihandbereich sind angesichts der Menge an neugekauften Online-Ressourcen schwierig umzusetzen und aktuell zu halten. (Denkbar wären hier höchstens am Anfang / Ende jeder großen Sachgruppe ein Hinweisschild mit den QR-Codes von einigen Online-Beispiel-Titeln mit Hinweis für weitere Titel im OPAC, das ist Abständen aktualisiert

werden müsste. Aber der Arbeitsaufwand steht in keinem Verhältnis zum Nutzen, da in einer Universitätsbibliothek viele Leute erst gezielt im OPAC schauen, also das Stöbern am Regal nicht die Regel ist.)

Alternative: Da wir für die E-Books RVK-Notationen vergeben, kann man sich im OPAC ein "virtuelles Regal" anzeigen lassen, wo Online und Print integriert werden.

Viel Erfolg für Ihre Arbeit

Angelika Krieser, FU Berlin

---

Sehr geehrte Frau Geier,

schauen Sie sich mal diese Präsentation an: [https://staatsbibliothek-berlin.de/fileadmin/user\\_upload/zentrale\\_Seiten/bestandsaufbau/pdf/Praessler\\_Versinnlichung\\_SBB\\_2015.pdf](https://staatsbibliothek-berlin.de/fileadmin/user_upload/zentrale_Seiten/bestandsaufbau/pdf/Praessler_Versinnlichung_SBB_2015.pdf)

Sie basiert auf einer Masterarbeit, die sich einem sehr ähnlichen Thema gewidmet hat. Hier finden Sie sicherlich viele Anhaltspunkte und zudem Bibliotheken, auf die Sie ggf. zugehen können.

Mit freundlichen Grüßen

Kornelia Abicht

---

Liebe Frau Geier,

da fallen mir spontan unsere Zeitschriften ein. Der aktuelle Jahrgang ist in Boxen hinterlegt und durch eine Klappe geschlossen. Wir haben die jeweiligen Titelblätter gescannt, auf der Klappe befestigt und falls wir Zugang zum E-Journal haben, wird zusätzlich ein QR-Code mit dem Link zur EZB generiert. Ein Beispiel sende ich Ihnen per Mail. Das Bild lässt sich leider hier nicht darstellen. Ich hoffe, dass so etwas auch interessant ist.

Für Monografien etc. haben wir keine Darstellung am Regal.

Die Einrichtung ist: Universität Koblenz-Landau, Campus Landau.

Viele Grüße und viel Erfolg für Ihre Masterarbeit!

Daniela Poeszus

---

Hallo Dana,

wir nutzen für Monografien kleine runde Schildchen mit der Info, dass das Buch im Katalog als e-Lizenz verfügbar ist, die wir wie ein Lesezeichen, gut sichtbar, in das Buch "hängen". Zeitschriften/Zeitungen beschildern wir ähnlich wie Frau Poeszus von der Uni Landau.

PS: Es ist schon ewig her, da haben wir im ersten Semester Bibliothekswissenschaften an der HU zusammen studiert, ich bin dann aber ausgestiegen. :-)

Viele Grüße und alles Gute, Elisabeth Herzig (ESCP Business School, Berlin)

---

Hallo Frau Geier,

wir haben unsere Print-Bücher mit einem "E-Book-Aufkleber" versehen - die Bücher mit Aufkleber sind auch als E-Book vorhanden.

Bei den Zeitschriften haben wir direkt neben den ZS-Auslage Boxen einen Wandbildschirm installieren lassen. Auf diesem sind alle unsere ZS, die es sowohl als Print als auch als E-Journal bei uns gibt. Man kann auf das Titelblatt einer ZS draufklicken und kommt dann direkt zu den Volltexten. Damit möchten wir die Nutzer animieren, die e-Journals verstärkt zu nutzen.

Viele Grüße und viel Erfolg wünscht das Team der DHBW Stuttgart Bibliothek

Für weitere Fragen stehen wir natürlich zur Verfügung.

Elke Nehm

---

Hallo Frau Geier,

leider Fehlanzeige. Wir hoffen, dass sich das im Rahmen des Neubaus 2023 ändert. Insofern wäre es nett, wenn Sie uns Ihre Ergebnisse mitteilen könnten; dann könnten wir die evtl. mit dort einfließen lassen als eine Art best practice.

Viele Grüße,

Torsten Haß / Hochschulbibliothek Ludwigshafen/Rhein

torsten.hass@hwg-lu.de

---

Liebe Frau Geier,

Sie sind sicher schon auf das Projekt "Blended Shelf" gestossen, das die UB Konstanz 2014 u.a. gemeinsam mit der UB Tübingen durchgeführt hatte. Hier sollten die vorhandenen gedruckten Bücher und E-Books auf einem Multi-Touch-Tisch optisch dargestellt werden. <https://www.kim.uni-konstanz.de/das-kim/ueber-das-kim/projekte/abgeschlossene-projekte/blended-shelf/>. M.W. ist die Software jetzt aber kommerziell weitergenutzt worden.

Viele Grüße

Randi Knorr, Information UB Tübingen

randi.knorr@uni-tuebingen.de

---

Hallo Frau Geier,

bei uns ist es so, dass E-Books leider nicht sachlich so erschlossen werden wie Print-Bestand.

Bei Print gibt es die Fachgruppe und die Signatur. Dort weiß man daher, wie die Sortierreihenfolge im Regal ist

und wir können eine virtual bookshelf Funktion in unserem Primo-Katalog zum Stöbern anbieten.

Bei E-Ressourcen können wir das leider nicht machen.

Sollten Sie auf eine geniale Lösung stoßen, wäre ich an Feedback dazu sehr interessiert.

Ansonsten noch viel Erfolg bei der Anfertigung der Masterarbeit.

Viele Grüße, Uwe Dierolf

Uwe.Dierolf@kit.edu

KIT-Bibliothek in Karlsruhe

---

Liebe Frau Geier,

das "Hybrid Bookshelf", das aus dem Projekt "Blended Shelf" hervorging, wurde bereits im Tübingen Eintrag weiter oben erwähnt. In Konstanz ist das Hybrid Bookshelf im Einsatz.

Informationen finden Sie auf der Seite der Firma "Picibird": <https://www.liberry.de/>

Bei Interesse nach mehr Informationen melden Sie sich gerne.

Viele Grüße, Isabell Leibing

Mail: [isabell.leibing@uni-konstanz.de](mailto:isabell.leibing@uni-konstanz.de)

Universität Konstanz, Kommunikations-, Informations-, Medienzentrum (KIM)

---

Liebe Frau Geier,

in der Bibliothek am RheinAhrCampus der Hochschule Koblenz arbeiten meine Kolleginnen und ich auch mit QR-Codes. Ähnlich wie Frau Krieser und Frau Poeszus gehen wir mit Zeitschriften um: Die aktuellen Jahrgänge liegen in Fächern, auf der Klappe befinden sich die Titelblätter - je nach dem mit QR-Code. Dieser führt allerdings in den meisten Fällen direkt zum eBook-Anbieter, nicht zur EZB.

Bei Büchern verfahren wir ähnlich wie Frau Krieser: Hier haben wir uns darauf konzentriert, vor allem dauernd vergriffene Bücher als eBooks zu kaufen. Sichtbar werden sie dann durch ein Schildchen im A6-Format, das im Regal genau dort hängt, wo auch das gedruckte Buch steht bzw. stehen würde. Auf den Schildchen ist sowohl das Cover und die Signatur als auch Hinweise zur eBook-Nutzung zu finden.

Viel Erfolg für Ihre Arbeit und liebe Grüße!

Lisa Sowa

---



Moin,

in den gedruckten Büchern, bei denen es die Erwerbung sieht, werden allgemeine Auch-als-E-Book-vorhanden-bitte-im-Katalog-schauen-Aufkleber auf die Rückseite des Vorderdeckels aufgeklebt. Bei den Zeitschriftenfächern gibt es einen A6-Zettel mit dem Hinweis auch als E-Ausgabe vorhanden inkl. QR-Code, der aber nur auf die EZB-Startseite führt. In einer Teilbibliothek (Bestand ~2000 Bände) habe ich vor einigen Jahren A5-Aufsteller mit 2-4 Titeln inkl. Cover und QR-Code in den Katalog in die Systemstellen gestellt, für die wir explizit spezielle englischsprachige E-Books gekauft haben um diese sichtbarer zu machen. Nutzung wurde nicht gemessen. Bei uns ist auch das Problem, dass es bei E-Books keine Signaturen bzw. Verknüpfungen mit den Aufstellungssystematiken gibt, sodass eine Browsing-Ansicht wie im Regal mit Verschränkung Print/Online nicht funktioniert. Es wäre für viele Fälle praktisch, wäre das anders. Z.b. wenn auch das E-Book auftaucht, wenn nach Signatur der Druckausgabe gesucht wird oder wenn es bei Werken mit mehreren Auflagen auch bei einer Druckausgabe, die wir nicht als E-Book haben ein Hinweis auftaucht, dass ein E-Book für eine ältere/neuere Auflage verfügbar ist oder wenn überhaupt deutlicher wäre wo wir E-Ressource und gedruckte Band haben. Ein sichtbarer Hinweis auf das E-Book im Katalog bei der gedruckten Ausgabe wäre schon toll. Damit es gleich klar ist, ohne das in der Titelaufnahme auf eine der Verknüpfungen geklickt werden muss und ich nie weiß, taucht das E-Book auf, taucht die Druckausgabe wieder auf, werde ich in den Verbundkatalog weitergeleitet (um nicht im nichts zu landen). Insbesondere unter den gegebenen Umständen wären Beratungsgespräche so viel entspannter. Wir kaufen in erster Linie E-Book-Pakete und da händisch auch rückwirkend was zu machen ist schon aufgrund der Menge unmöglich. Und automatisierte Verknüpfungen mit entsprechender Anzeige (z.b. von Seiten der Verbundzentrale eine Anreicherung von lizenzierten E-Book-Titeln mit der

Print-Signatur/einem Abrufzeichen, dass im OPAC beide Titel sichtbarer verbinden würde oder der Sacherschließung einer Druckausgabe) gibt es nicht. Da verstehe ich nicht warum das nicht irgendwie automatisiert geht/gemacht wird. Mich würde interessieren, ob so eine Browsing-Option die Druckausgabe/E-Book verschränkt was für Discovery-Systeme wäre als andere Art der Darstellung, neben einer Kurzliste und wie das umgesetzt werden könnte. Oder ob dann wenigsten dort in zweiter Instanz mit den vorhandenen Daten eine Anreicherung stattfinden könnte, die sichtbare Verknüpfungen (wie Anzeige Downloadmöglichkeit auch bei Print-Treffern) ermöglichen könnte.

Beste Grüße, Oliver Feig (Bibliothek der Hochschule Hannover)

---

Guten Tag,

da wir in den Zeitschriftenregalen im Lesesaal (in denen immer das Titelblatt der neuesten Ausgabe sichtbar ist) noch freie Plätze hatten, haben wir dort QR Codes von unseren einzeln lizenzierten E-Journals (auch mit einem Beispieltitelblatt) angebracht.

Beste Grüße,

Cornelia Panzenböck (FH des BFI Wien)

---

Guten Tag,

in 2 zwei besuchten Bibliotheken habe ich vor einiger Zeit schon, ein Paar Ideen gefunden.

In einer genfer geisteswissenschaftliche Universitätsbibliothek gab es plastifizierte A4 Blätter mit Bild und Zusammenfassung, Autoren Bio, sowie QR Code des elektronischen Buches. Die Blätter standen im Regal zwischen 2 Bücher, genau am themetischen Platz, den sie als Papierbuch gehabt hätten.

In einer amerikanischen medizinischen Bibliothek (Houston) gab es kleine Papiersignets mit praktischen Informationen zum mitnehmen, um ein neues elektronisches Dokument der Leserschaft bekannt zu machen.

Mit freundlichen Grüßen aus der BCU Lausanne, Béatrice Kamber Thiocone

---

Liebe Frau Geier,

Bücher, die auch online vorhanden sind, werden bei uns durch grüne Etiketten am Regal, auf denen Titel, Autor und der Name der jeweiligen Datenbank stehen, kenntlich gemacht. Die Etiketten kleben direkt unter den jeweiligen Print-Büchern. Es gibt jedoch (am Regal) bislang keinen Hinweis auf diejenigen Bestände, die ausschließlich online vorhanden sind. Bei unseren Zeitschriften gibt es bislang ebenfalls noch keinen Hinweis.

Viele Grüße aus den Zentralbibliotheken Jura der Juristischen Fakultät der Universität Göttingen!

Claudia Keller

---

Guten Tag Frau Geier,

in der Campusbibliothek der FU-Berlin hatten wir bis dato Platzhalter der Titel mit QR-Codes zur E-Ressource in der Lehrbuchsammlung aufgestellt. Dies galt auch für E-Only-Titel. Das sind - je nach Fachbereich - mittlerweile die meisten. Wir warfen diese nun zum aktuellen Semester raus, da es dem Suchverhalten unserer Studis längst nicht mehr entspricht. Die Studis sind mittlerweile alle mittels Katalogrecherche unterwegs und erwarten eigentlich eher ein E-book (dank Corona und Onlinesemestern noch mehr denn je).

Da es in Zukunft gilt, die Leute da abzuholen, wo sie sind (und das ist im Zweifelsfall nicht das Bibliotheksregal), finde ich übrigens die Idee von Stefan Wehrmeyer (OKFN, FragDenStaat) mit Webbrowserplugins zu Arbeiten gut. Vgl.: <https://twitter.com/hashtag/voebbot?> - Etwas in der Art ist nicht nur für E-Medien denkbar, sondern auch als Katalogerweiterung für physische Bestände und v.A. für ÖB interessant.

Grüße aus der Campusbibliothek der FU,

Florian Wieland

## A.4 Übersicht RVK-Notationen pro Regalboden (Bereich Informatik)

Stand: 08.07.2021

RegalID	RVK_von	RVK_bis
01-01-l-01	<i>RB 10103</i>	<i>SH 500</i>
01-01-l-02	<i>SK 110</i>	<i>SK 890</i>
01-01-l-03	<i>SK 910</i>	<i>SK 950</i>
01-01-l-04	<i>SK 955</i>	<i>SR 100</i>
01-01-l-05	SR 100	SR 200
01-01-l-06	SR 200	SR 800
01-01-l-07	SR 850	SR 870
01-01-l-08	SR 870	SR 870
01-01-l-09	SR 870	SS 4800
01-01-l-10	ST 110	ST 110
01-01-l-11	ST 110	ST 110
01-01-l-12	ST 110	ST 110
01-01-l-13	ST 120	ST 130
01-01-l-14	ST 130	ST 134
01-01-l-15	ST 134	ST 134
01-01-l-16	ST 134	ST 134
01-01-l-17	ST 134	ST 140
01-01-l-18	empty	Empty
01-01-r-01	ST 150	ST 150
01-01-r-02	ST 151	ST 166
01-01-r-03	ST 170	ST 175
01-01-r-04	ST 177	ST 177
01-01-r-05	ST 177	ST 190
01-01-r-06	empty	empty
01-01-r-07	ST 200	ST 200
01-01-r-08	ST 200	ST 200
01-01-r-09	ST 200	ST 200
01-01-r-10	ST 200	ST 200
01-01-r-11	ST 200	ST 200
01-01-r-12	empty	empty
01-01-r-13	ST 201	ST 201
01-01-r-14	ST 201	ST 201
01-01-r-15	ST 201	ST 201
01-01-r-16	ST 205	ST 205
01-01-r-17	ST 205	ST 205
01-01-r-18	empty	Empty
01-02-l-01	ST 206	ST 206
01-02-l-02	ST 206	ST 230
01-02-l-03	ST 230	ST 230
01-02-l-04	ST 230	ST 230
01-02-l-05	ST 230	ST 230
01-02-l-06	empty	empty

RegalID	RVK_von	RVK_bis
01-02-l-07	ST 230	ST 230
01-02-l-08	ST 230	ST 230
01-02-l-09	ST 230	ST 230
01-02-l-10	ST 230	ST 231
01-02-l-11	ST 231	ST 231
01-02-l-12	empty	empty
01-02-l-13	ST 231	ST 231
01-02-l-14	ST 232	ST 232
01-02-l-15	empty	empty
01-02-l-16	ST 232	ST 232
01-02-l-17	ST 233	ST 234
01-02-l-18	empty	Empty
01-02-r-01	ST 237	ST 240
01-02-r-02	ST 250   A02	ST 250   B10
01-02-r-03	ST 250   C01	ST 250   C01
01-02-r-04	ST 250   C01	ST 250   C01
01-02-r-05	ST 250   C01	ST 250   C01
01-02-r-06	empty	empty
01-02-r-07	ST 250   C01	ST 250   C01
01-02-r-08	ST 250   C01	ST 250   F60
01-02-r-09	ST 250   G88	ST 250   H85
01-02-r-10	ST 250   J35	ST 250   J35
01-02-r-11	ST 250   J35	ST 250   J35
01-02-r-12	empty	empty
01-02-r-13	ST 250   J35	ST 250   J35
01-02-r-14	ST 250   J35	ST 250   J35
01-02-r-15	ST 250   J35	ST 250   J35
01-02-r-16	ST 250   J35	ST 250   J35
01-02-r-17	ST 250   J35	ST 250   J35
01-02-r-18	empty	Empty
01-03-l-01	ST 250   J35	ST 250   J35
01-03-l-02	ST 250   J35	ST 250   J35
01-03-l-03	ST 250   J35	ST 250   J35
01-03-l-04	ST 250   J35	ST 250   J35
01-03-l-05	ST 250   J35	ST 250   J35
01-03-l-06	empty	empty
01-03-l-07	ST 250   J35	ST 250   J35
01-03-l-08	ST 250   J35	ST 250   J35
01-03-l-09	ST 250   J35	ST 250   J35
01-03-l-10	ST 250   J61	ST 250   L03
01-03-l-11	ST 250   L03	ST 250   L03
01-03-l-12	empty	empty
01-03-l-13	ST 250   L03	ST 250   P30
01-03-l-14	ST 250   P30	ST 250   P32
01-03-l-15	ST 250   P32	ST 250   P32
01-03-l-16	ST 250   P33	ST 250   P99
01-03-l-17	ST 250   R1	ST 250   V95
01-03-l-18	empty	Empty
01-03-r-01	ST 250   X70	ST 250   X70

RegalID	RVK_von	RVK_bis
01-03-r-02	ST 250   X70	ST 250   X84
01-03-r-03	ST 252	ST 252
01-03-r-04	ST 252	ST 252
01-03-r-05	ST 252	ST 252
01-03-r-06	empty	empty
01-03-r-07	ST 252	ST 253   A39
01-03-r-08	ST 253   A65	ST 253   A82
01-03-r-09	ST 253   C23	ST 253   F34
01-03-r-10	ST 253   G88	ST 253   J35
01-03-r-11	ST 253   J35	ST 253   J86
01-03-r-12	empty	empty
01-03-r-13	ST 253   M21	ST 253   N37
01-03-r-14	ST 253   N37	ST 253   S81
01-03-r-15	ST 253   T91	ST 253   T96
01-03-r-16	ST 253   T96	ST 253   Z91
01-03-r-17	ST 255	ST 257
01-03-r-18	empty	Empty
01-04-l-01	ST 260	ST 260
01-04-l-02	ST 261   A39	ST 261   L33
01-04-l-03	ST 261   M06	ST 261   S66
01-04-l-04	ST 261   S66	ST 261   U61
01-04-l-05	ST 261   U61	ST 261   U61
01-04-l-06	empty	empty
01-04-l-07	ST 261   U61	ST 261   U61
01-04-l-08	ST 261   U61	ST 261   U61
01-04-l-09	ST 261   U61	ST 261   U61
01-04-l-10	ST 261   U61	ST 261   U61
01-04-l-11	ST 261   U61	ST 261   U61
01-04-l-12	empty	empty
01-04-l-13	ST 261   V27	ST 261   W34
01-04-l-14	ST 261   W34	ST 261   W34
01-04-l-15	ST 265	ST 270
01-04-l-16	ST 270	ST 270
01-04-l-17	ST 270	ST 270
01-04-l-18	empty	Empty
01-04-r-01	ST 270	ST 270
01-04-r-02	ST 271   A06	ST 271   H57
01-04-r-03	empty	empty
01-04-r-04	ST 271   I44	ST 271   O57
01-04-r-05	ST 271   O57	ST 271   Q39
01-04-r-06	empty	empty
01-04-r-07	ST 271   S70	ST 271   S70
01-04-r-08	ST 273	ST 273
01-04-r-09	ST 276   A318	ST 276   G233
01-04-r-10	ST 276   G231	ST 276   P748
01-04-r-11	ST 276   R161	ST 276   W837
01-04-r-12	empty	empty
01-04-r-13	ST 277	ST 277
01-04-r-14	ST 277	ST 277

RegalID	RVK_von	RVK_bis
01-04-r-15	ST 277	ST 277
01-04-r-16	ST 278	ST 278
01-04-r-17	ST 280	ST 281
01-04-r-18	ST 281	ST 281
01-05-l-01	ST 281	ST 281
01-05-l-02	ST 284	ST 285
01-05-l-03	ST 285	ST 300
01-05-l-04	ST 300	ST 300
01-05-l-05	ST 301	ST 301
01-05-l-06	empty	empty
01-05-l-07	ST 301	ST 301
01-05-l-08	ST 302	ST 308
01-05-l-09	ST 320	ST 320
01-05-l-10	ST 321	ST 321
01-05-l-11	ST 321	ST 321
01-05-l-12	empty	empty
01-05-l-13	ST 323	ST 324
01-05-l-14	ST 325	ST 326
01-05-l-15	empty	empty
01-05-l-16	ST 330	ST 330
01-05-l-17	ST 330	ST 330
01-05-l-18	empty	Empty
01-05-r-01	ST 330	ST 331
01-05-r-02	ST 331	ST 340
01-05-r-03	ST 340	ST 351
01-05-r-04	ST 351	ST 351
01-05-r-05	ST 351	ST 361
01-05-r-06	empty	empty
01-05-r-07	ST 370	ST 371
01-05-r-08	ST 500	ST 505
01-05-r-09	ST 505	ST 505
01-05-r-10	ST 505	ST 505
01-05-r-11	ST 505	ST 505
01-05-r-12	empty	empty
01-05-r-13	ST 510   A74	ST 510   A74
01-05-r-14	ST 510   O32	ST 510   S05
01-05-r-15	ST 510   S05	ST 510   S05
01-05-r-16	ST 510   S05	ST 510   S05
01-05-r-17	ST 510   S05	ST 510   S05
01-05-r-18	empty	Empty
01-06-l-01	ST 510   S05	ST 510   S05
01-06-l-02	ST 510   S05	ST 510   S05
01-06-l-03	ST 510   S05	ST 510   S05
01-06-l-04	ST 510   S05	ST 510   S05
01-06-l-05	ST 510   S05	ST 510   S05
01-06-l-06	empty	empty
01-06-l-07	ST 510   S05	ST 515
01-06-l-08	ST 515	ST 515
01-06-l-09	ST 515	ST 515

<b>RegalID</b>	<b>RVK_von</b>	<b>RVK_bis</b>
01-06-l-10	ST 520	ST 530
01-06-l-11	ST 530	ST 530
01-06-l-12	ST 600	ST 600
01-06-l-13	ST 601	ST 601
01-06-l-14	ST 601	ST 601
01-06-l-15	ST 610	ST 640
01-06-l-16	ST 650	ST 670
01-06-l-17	ST 690	TZ 9950
01-06-l-18	empty	empty

## A.5 E-Mail mit Anforderungen an API zur Ermittlung entliehener Print-Monographien

Hallo Herr Kissig,

anbei meine Wünsche zur Antwort der Schnittstelle für die Abfrage der "entliehenen" Bände:

- Titel + Zusatz
- Verfasser
- Erscheinungsjahr
- Verlag
- Erscheinungsjahr
- Auflage
- ISBN
- RVK-Notation/en
- wenn möglich:
  - ausgeliehen bis (Leihfrist)
  - Anzahl Vormerkungen (bereits durch anderen Nutzer/andere Nutzerin vorgemerkt?)

[...]

Beste Grüße

Dana Geier

*[E-Mail vom 11.02.2021]*



## A.6 CD-ROM zur Arbeit

Im hinteren Einband der Arbeit befindet sich eine CD-ROM mit folgendem Inhalt:

- Text der Masterarbeit als PDF
- Tabellen zur Planung der Inhalte der QR-Codes als XLSX und PDF-Dateien:
  - Das erste Tabellenblatt („Ueberblick\_Signaturen“) beinhaltet die Signaturen aller Bände, die im Bereich Informatik stehen. Diesen sind die RVK-Notation und die Trefferanzahl der Solr-API zugeordnet (bei Abfragen mit RVK-Notation). Hier findet sich bereits eine erste Idee der möglichen Aufteilung auf QR-Codes.
  - Das zweite Tabellenblatt („QR-Codes\_Versuch\_1“) beinhaltet bereits genauere Ideen zur Aufteilung der RVK-Notationen unter Berücksichtigung der Regalaufstellung.
- Tabelle mit allen QR-Code-Inhalten für den Bereich Informatik („QR-Codes\_Daten“) als xlsx und PDF-Datei
- Vorlage zur Erstellung von QR-Codes als XLSX-Datei („QR-Codes\_Template“)
- Anleitung zum Avery-Drucktool als PDF- und docx-Datei
- PDF-Datei mit allen QR-Codes zum Bereich Informatik („QR-Codes\_Bereich\_Informatik.pdf“)

Die Quellcodes zur Arbeit befinden sich im Git-Repository des iCampus-Teams der TH Wildau. Für Rückfragen hierzu wenden Sie sich bitte an: [icampus@th-wildau.de](mailto:icampus@th-wildau.de)

# Selbständigkeitserklärung

Ich versichere, dass die vorliegende Arbeit von mir selbständig und ohne unerlaubte Hilfe angefertigt worden ist. Ich habe alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, durch Zitate bzw. Literaturhinweise als solche kenntlich gemacht.

Berlin, den

Dana Geier