

# Masterarbeit

zur Erlangung des akademischen Grades  
Master

**Technische Hochschule Wildau**  
**Fachbereich Wirtschaft, Informatik, Recht**  
**Studiengang Bibliotheks-informatik (M. Sc.)**

**Thema (deutsch):** Konzeption und Entwicklung eines datengetriebenen  
Unterstützungssystems für Etatplanung und Mittelallokation einer  
hybriden Spezialbibliothek

**Thema (englisch):** Design and development of a data-driven support system for budget  
planning and resource allocation of a hybrid library

Autor/in: Peter Breternitz  
Seminargruppe: BIM/17  
Betreuer/in: Dipl.-Informatiker Sascha Szott  
Zweitgutachter/in: Dr. Frank Seeliger  
Eingereicht am: 10.03.2021



Konzeption und Entwicklung eines datengetriebenen  
Unterstützungssystems für Etatplanung und Mittelallokation einer  
hybriden Spezialbibliothek

*von*

Peter Breternitz



## ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## ZUSAMMENFASSUNG

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



# INHALTSVERZEICHNIS

1	EINFÜHRUNG	1
1.1	Problemstellung . . . . .	1
1.2	Ziel der Arbeit . . . . .	3
1.3	Verwandte Arbeiten . . . . .	5
1.4	Gliederung der Arbeit . . . . .	7
2	THEORETISCHE GRUNDLAGEN	9
2.1	Bibliothek und Statistik . . . . .	9
2.2	Datenvisualisierung . . . . .	13
2.3	Business-Intelligence-Systeme . . . . .	17
3	AUSGANGSSITUATION	23
3.1	Bibliothek . . . . .	23
3.1.1	Allgemeines . . . . .	23
3.1.2	Organisatorische Einbettung . . . . .	24
3.1.3	Informationsdienstleistungen . . . . .	24
3.1.4	Evaluation der Informationsdienstleistungen . . . . .	25
4	KONZEPTION EINER LÖSUNG	29
4.1	Anforderungsanalyse . . . . .	29
4.1.1	Vision und Ziele . . . . .	30
4.1.2	Rahmenbedingungen . . . . .	30

4.1.3	Funktionale Anforderungen . . . . .	31
4.1.4	Nicht-funktionale Anforderungen . . . . .	33
4.1.5	Anwendungsfälle . . . . .	34
5	DISKUSSION DER UMSETZUNG	41
5.1	Implementierung . . . . .	41
5.1.1	Technische Details der Implementierung . . . . .	41
5.1.2	Systemarchitektur . . . . .	45
5.1.3	Teilsysteme . . . . .	47
5.2	Demonstration der Funktionalität . . . . .	65
5.2.1	Technische Voraussetzungen . . . . .	65
5.2.2	Daten-Import . . . . .	65
5.2.3	Dashboard . . . . .	66
5.3	Bewertung . . . . .	73
5.3.1	Datenlage . . . . .	74
5.3.2	Umgesetzte Anforderungen der Anforderungsanalyse . . . . .	76
5.3.3	Umgesetzte Anwendungsfälle . . . . .	77
6	SCHLUSS	83
	TABELLENVERZEICHNIS	87
	ABBILDUNGSVERZEICHNIS	89
	QUELLCODEVERZEICHNIS	91
	AKRONYME	93
	LITERATURVERZEICHNIS	95



# 1 EINFÜHRUNG

Als Antwort auf die anhaltende Unsicherheit im Bereich der öffentlichen Gesundheit, die die Ausbreitung der COVID-19-Pandemie hervorgerufen hat, entwickelte zu Beginn des Jahres 2020 ein Team um die Professorin Lauren Gardner der *Johns Hopkins University* ein Dashboard. Dieses visualisiert die gemeldeten Fälle der COVID-19-Pandemie weltweit. Das Dashboard wurde entwickelt, um Forschenden, Gesundheitsbehörden und der breiten Öffentlichkeit ein benutzerfreundliches Instrument an die Hand zu geben, mit dem sich der Ausbruch leicht verfolgen lässt. Visualisiert durch eine Weltkarte und unterschiedlich großen Punktmarkierungen zeigt das Dashboard die gegenwärtige Ausbreitung. Zusätzlich werden Zahlen der bestätigten COVID-19-Fälle, der Todesfälle und der Genesungen für alle betroffenen Länder angezeigt[Vgl. [DDG20](#), S. 533]. In der Anfangszeit des Dashboards wurden die Daten noch manuell gesammelt und bearbeitet. Mittlerweile wurde ein halb-automatischer Prozess implementiert. Durch die übersichtliche Darstellung auf dem Dashboard können rasch Informationen über die Gefahrenlage von einer breiten Öffentlichkeit rezipiert werden. Daraus kann ein Verständnis der derzeitigen Maßnahmen bei der Öffentlichkeit entstehen sowie die Informationen in eigene Handlungsoptionen integriert werden.

## 1.1 PROBLEMSTELLUNG

Ausgehend von ökonomischen, informationstechnologischen und marktpolitischen Veränderungen in den vergangenen Jahrzehnten, sind Bibliotheken dazu veranlasst, ihr Budget

hinsichtlich der Informationsbedarfe ihrer Nutzer:innen behutsamer zu planen und sich in zunehmenden Maße gegenüber ihren Unterhaltsträgern zu rechtfertigen. Die Relevanz von bibliothekarischen Statistiken ist in diesem Zusammenhang größer geworden. Deswegen ist es wichtig, Daten aus den bibliothekarischen Bereichen zu aggregieren, statistisch zu erheben und auszuwerten, um auf Basis der daraus erzielten Erkenntnisse handeln zu können. Die Transparenz von statistischen Daten sorgt für eine bessere Grundlage in den Verhandlungen mit den Stakeholdern einer Bibliothek. Zudem wird durch sie der Einsatz des Bibliotheksbudgets zielgerichteter auf die Bedürfnisse der Nutzer:innen zugeschnitten. Dazu ist es zweckmäßig, alle anfallenden Daten für die Budgetplanung und Mittelallokation einer Bibliothek zentral zu sammeln und mit geeigneten statistischen Methoden und Verfahren langfristig auszuwerten. Um den Wert dieser aus den Daten gewonnenen Information elegant den Stakeholdern zu kommunizieren und zu präsentieren, können geeignete Verfahren der Datenvisualisierung zum Einsatz kommen. Die technische Realisierung kann durch gewöhnliche Tabellenkalkulationsprogramme erfolgen. Um den mitunter hohen Zeitaufwand und die Fehleranfälligkeit manueller Prozesse einerseits zu minimieren und den Automatisierungsgrad hinsichtlich der Aggregation und Auswertung der bibliothekarischen Daten andererseits zu erhöhen, können aber auch andere technische Umsetzungen eingesetzt werden. In Bereichen der Wirtschaft kommen Business Intelligence (BI)-Systeme zum Einsatz, die IT-basiert Entscheidungsfindungen unterstützen (siehe dazu [Abschnitt 2.3](#)).

Es gibt bereits eine Vielzahl kommerzieller Lösungen für den Bibliotheksbereich, die auf *BI-Systemen* basieren. Zu nennen wären *AlmaAnalytics* für das Next-Generation-Library-System *Alma* von *ExLibris* [[Lib20b](#)], *BibControl* vom Online Computer Library Center (OCLC) [[OCL20](#)], *CollectionHq* von *Baker & Taylor* [[BT20](#)] oder *Libinsight* von *SpringShare* [[Spr20](#)]. Darüber hinaus gibt es *Business Intelligence-Systeme*, die von Bibliotheken für Reporting, Datenanalyse und Datenvisualisierung adaptiert werden, wie zum Beispiel *Tableau* von der Firma *Tableau Software* [[Sof20](#)], *Crystal Reports* von *SAP*

[SAP20] oder *Microsoft BI* von *Microsoft* [Mic20]. Diese Applikationen sind entweder an bestimmte Bibliothekssysteme zurückgebunden, limitiert in ihren Funktionen oder zu generisch.

Für die Spezialbibliothek des Max-Planck-Institut für empirische Ästhetik (MPI EA) begründet sich die Notwendigkeit für eine an *BI-Systemen* angelehnte Applikation durch das Fehlen eines zentralen Nachweissystems für bibliothekarische Statistiken in der Bibliothek. Überdies wird sowohl vom Hessisches Bibliotheksinformationssystem (HeBis)-Verbund, deren Mitglied die Bibliothek ist, als auch von der max-planck-digital-library (mpdl) keine Systeme in dieser Richtung angeboten. Ebenso ist ungewiss, wann die Ablösung des schon betagten Bibliothekssystems hin zu einem neuen Next-Generation-Library-System in *HeBis* stattfinden wird und ob es ein Modul zur statistischen Datenerhebung mitbringen wird. Des Weiteren ist das Erfordernis, bibliothekarische Geschäftsprozesse zu evaluieren und die Servicedienstleistungen bezüglich der Ziele der Institution noch weiter zu optimieren, von großer Relevanz. Die zu entstehende Applikation könnte hierbei helfen, systematisches Controlling einzuführen und das Bibliotheksmanagement weiter zu professionalisieren. Mit dem Ende der Konsolidierungsphase der Bibliothek, die im Zuge des *MPI EA* 2013 gegründet wurde, tritt sie ein in eine Phase, in der ab dem Jahr 2021 Budgetplanungen eine größere Rolle spielen werden.

## 1.2 ZIEL DER ARBEIT

Das Ziel der Arbeit ist die Schaffung eines Dashboards für die Budgetplanung in der Spezialbibliothek des *MPI EA*. In Anlehnung an *BI-Systeme* soll ein ganzheitliches System als Proof-of-Concept entstehen, mit dem systematisch die relevanten Daten der hybriden Spezialbibliothek aggregiert, statistisch mit geeigneten und modernen Datenvisualisierungen analysiert und ausgewertet werden sollen. Darüber hinaus soll es möglich sein, aus dem System ausgewählte Resultate automatisiert als Standardbericht zu exportieren, um

## 1 Einführung

diese als *factsheet* gegenüber Stakeholdern der Bibliothek präsentieren zu können. Um künftigen Anforderungen gewachsen zu sein, soll das Dashboard modulbasiert programmiert werden und dadurch leicht erweiterbar sowie eventuell von anderen Bibliotheken nachnutzbar sein.

Es gibt einen wachsenden Markt für Systeme, die solche Anwendungen möglich machen. Dieser wächst im Schatten von Data-Science und Big Data. Der Markt verfügt über ausgereifte und mächtige Frameworks, die statistische Auswertungen mit wenig Programmieraufwand erlauben. Der höhere zeitliche Aufwand bei solchen statistischen Auswertungen liegt demgegenüber einerseits in der Aufbereitung der Daten, insbesondere dann, wenn die Daten aus heterogenen Datenquellen kommen. Das setzt dementsprechend eine genaue Analyse der Daten voraus. Andererseits kann die Zusammenführung der einzelnen statistischen Auswertungen in eine Applikation eine nicht leicht zu nehmende Hürde darstellen. Die Entwicklung von interaktiven Dashboards oder ähnlichen Web-Anwendungen ist ebenfalls leichter geworden, da Kenntnisse in einer Programmiersprache durchaus ausreichend sind, um eine solche Applikation zu programmieren.

Folgende Aufgaben und Zwecke soll das System lösen können: Die Daten sollen aus den heterogenen Datenquellen mit einem Extract, Transform, Load (ETL)-Prozess bearbeitet und in geeigneter Form gespeichert und bereitgestellt werden. Die Auswertungen erfolgen mittels statistischer Verfahren, die aus den Formulierungen der Fragen an die Daten abgeleitet werden. Dabei sollen Verfahren der deskriptiven und explorativen Statistik zum Einsatz kommen. Die Auswertungen werden dann über das interaktive Dashboard visuell dargestellt. Die visuelle Darstellung der Daten soll nach einstellbaren Parametern schrittweise verfeinert werden können.

Ein wesentlicher Zweck des Dashboards ist das Monitoring der laufenden Budgetkosten für die Medien. So sollen die Kosten für das laufende Jahr auch im Vergleich zu den Vorjahren dargestellt werden können. Ein anderer wichtiger Baustein sind die Ausleihzahlen der Medien. Diese können anhand der Dimensionen der Zeit oder der Aufstellungssystematik

analysiert werden. Diese Analysen können sich nach Zeitraum beziehungsweise nach Bestandssegmenten bis hinunter auf Titelebene als kleinste Ebene ausdifferenzieren. Ebenso vorgesehen ist eine Analyse der Neuerwerbungen nach Zeit anhand der Dimension der Aufstellungssystematik.

Der Standardbericht greift auf vorberechnete Auswertungen und Darstellungen zurück und generiert anfrageorientiert einen Bericht, der die wichtigsten Key Performance Indicators (KPI) der Bibliothek enthält. Er wird im Format PDF verteilt und kann ohne bibliothekarisches Domänenwissen gelesen werden.

Das Dashboard ist für das Monitoring und die Analyse der Daten durch die Bibliothek gedacht. So kann die Bibliotheksleitung die Ausgaben für Medien nach Lieferanten überwachen und gegebenenfalls steuernd eingreifen. Anhand der Analyse der Daten kann die Bibliotheksleitung das Jahresbudget besser steuern. Am Wachstum des Bestandes beziehungsweise der Bestandssegmente können die Bibliotheksmitarbeiter:innen ablesen, welche Bestandsgruppe wie schnell wächst. Der Standardbericht als aggregierte Form wichtigster *KPI* ist für die Kommunikation und Präsentation gegenüber der Institutsleitung gedacht.

## 1.3 VERWANDTE ARBEITEN

Es gab in den letzten Jahren Versuche an Universitätsbibliotheken Dashboards zu entwickeln, um daten-getriebene Entscheidungen zu unterstützen. So wurde an den *New York University Health Sciences Libraries* ein Dashboard entwickelt, das versucht, möglichst viele Metriken aus bibliothekarischen Dienstleistungen aufzunehmen. Die Architektur des Dashboards besteht aus drei Hauptteilen. Die Daten werden mit Import-Skripten aus den verschiedenen Datenquellen bezogen, mit einem *ETL-Prozess* bearbeitet und in ein Data Warehouse geladen. Das Data Warehouse stellt eine einfache MySQL-Datenbank dar. Die Daten werden aus dieser mit einem Mix von PHP/Javascript-Skripten in einem

## 1 Einführung

Dashboard mit unterschiedlichen Diagrammen dargestellt [MH12]. Ebenfalls wurde an der *James C. Kirkpatrick Library der University of Central Missouri* ein Dashboard zur Unterstützung Evidenz-basierter Entscheidungen der Bibliotheksleitung entwickelt. Vom Entwicklungs-Team wurde sich gegen kommerzielle Lösungen entschieden und ein Dashboard mit dem Ruby-on-Rails-Framework entwickelt. Es zeigt verschiedene Daten wie Study-Room- oder Computer-Usage-Data [Lib20a]. Horne-Popp, Tessone und Welker geben detaillierten Einblick in die Designentscheidungen und Problematiken während der Entwicklung dieses Tools [vgl. HTW18, 194 ff.].

An der *Universitätsbibliothek der Technischen Universität Berlin* wird *Alma Analytics* benutzt, was eine Business-Intelligence-Lösung des Bibliothekssystemanbieters *ExLibris* darstellt. Golas beschreibt die Neuimplementierung der statistischen Abfragen nach Einführung des cloudbasierten Bibliotheksmanagementsystems *Alma* an der Universitätsbibliothek. Neu implementiert wurden unter anderem Trends nach der Regensburger Verbundklassifikation (RVK), die RVK-Schlagwörter-Zuordnung, Statistiken über Ausleihen und Vormerkungen [Gol18].

In eine ähnliche Richtung wie das zu entwickelnde Tool, geht die an der *Technischen Hochschule Wildau* entwickelte Software *BibloVis*. *BibloVis* basiert auf einem modular aufgebauten Client-Server-Modell und ermöglicht die Einbindung und Visualisierung von Daten wie der Katalognutzung, Raumauslastungen und anderen bibliothekarischen Service-Angeboten. Die Grundlage bildet die Auswertungen von CSV- und XML-Dateien [BS15].

Der Einsatz von *Tableau* an den *Ohio State University Libraries* wird von Murphy beschrieben. Anhand von zwei Projekten werden die Einsatzmöglichkeiten insbesondere der Datenvisualisierungsoptionen dieser Software dargestellt. Unter anderem wurden die Transaktionsprotokolldateien auf Bibliotheksleitfäden untersucht, die von der Bibliothek zu den Universitätskursen herausgegeben wurden. Diese wurden über einen Zeitraum von 2009-2012 ausgewertet und mit verschiedenen Datenvisualisierungen dargestellt [vgl. Mur13, 469 f.].

Ein gutes Beispiel für ein datengetriebenes Unterstützungssystem findet sich bei Spielberg, der sich mit dem Thema der Bestandspflege an der *Universitätsbibliothek Essen* befasst und eine Applikation (weiter-)entwickelt hat, die die Fachreferent:innen bei der Aussonderung und Erwerbung von Medien unterstützt [Spi17].

Ebenso finden sich in der Fachliteratur Ansätze, die vorrangig anhand einzelner Fragestellungen hinsichtlich der Bestandsentwicklung[Hug16] oder anderer bibliothekarischer Servicedienstleistungen[KM20; KWC06; Mey18] verschiedene statistische Analysen vollzogen und diese visualisiert haben.

Fast alle Projekte sind an größeren Bibliotheken mit ganz unterschiedlichen softwaretechnischen Herangehensweisen[FF16; WH13] und Zielen[Phe12] entstanden.

Zudem fehlt ein Beispiel in der Literatur, das holistisch alle relevanten Daten, die in den Geschäftsgängen und Servicedienstleistungen insbesondere einer Spezialbibliothek entstehen, aggregiert, auf diesen Daten automatisch statistische Analysen ausführt und diese mit modernen Visualisierungstechniken interaktiv darstellt.

## 1.4 GLIEDERUNG DER ARBEIT

Im Kapitel 2 werden die theoretischen Grundlagen für die folgenden Kapitel gelegt. Das Kapitel beschreibt den Rahmen, in dem die Entwicklung eines Dashboards eingebettet ist. Dabei wird herausgestellt, wie wichtig Statistik im bibliothekarischen Bereich ist, was Datenvisualisierungen sind und warum sie eingesetzt werden sollen und welche Anleihen *BI-Systeme* für das zu entstehende System liefern können. Im Kapitel 3 wird die Bibliothek vorgestellt und darauf eingegangen, welche bibliothekarischen Statistiken bereits erhoben wurden. Nachdem die Ausgangssituation bestimmt wurde, wird mit der Anforderungsanalyse im Kapitel 4 das generierte Wissen von Kapitel 2 aufgegriffen und die Konzeption einer Lösung vorgestellt. Im Kapitel 5 wird die Umsetzung diskutiert. Bevor das System bewertet wird, wird das Design und die Implementierung vorgestellt. Das Fazit wird im

## 1 Einführung

Kapitel 6 mit dem Stand der Umsetzung, den *lessons learned* und einem Ausblick auf Themen, die noch bearbeitet werden könnten, gezogen.



## 2 THEORETISCHE GRUNDLAGEN

In diesem Kapitel wird der theoretische Rahmen für die weiteren Kapitel gelegt. Im ersten Abschnitt werden die Grundlagen der Budgetplanung und Mittelallokation im Zusammenhang mit bibliothekarischen Statistiken erläutert. Der darauffolgende Abschnitt handelt von Datenvisualisierungen und deren Einsatz für Datenrepräsentationen und Datenpräsentation. Abschließend wird das Modell der *Business Intelligence-Systeme* als dritter Baustein zur Lösung der Problemstellung dieser Arbeit vorgestellt.

### 2.1 BIBLIOTHEK UND STATISTIK

Die Etatplanung von Bibliotheken richtet sich nach deren Informations- und Versorgungsauftrag. Seit Beginn der 1990er Jahre müssen sich Bibliotheken mit den Auswirkungen einer veränderten Medienlandschaft auseinandersetzen. Sie kämpfen mit einem größer werdenden Informationsangebot, den steigenden Preisen auf dem Publikationsmarkt, den zunehmenden Kommerzialisierungstendenzen in der Verlagslandschaft und neuen Medientypen. Zu nennen wären hier konkret: die Explosion der Zeitschriftenpreise im Bereich der Science, Technology, and Medicine (STM), die Konzentration auf wenige Verlage, und dem Aufkommen von ebooks. Demgegenüber steigen Bibliotheksetats nur mäßig. Demzufolge geht ein Kaufkraftverlust der Bibliotheken einher [vgl. [Mor15](#), 164 ff.]. Bibliotheken haben gegen diesen Trend Instrumente entwickelt, um den Informationsauftrag trotz dieser Widrigkeiten zu erfüllen. So entstehen seit Mitte der 1990er Jahre von Bund und Ländern geförderte Konsortien, um den Kostendruck auf Bibliotheken insbe-

sondere im Bereich der elektronischen Fachinformationen zu mildern. Neue Geschäftsmodelle werden zur Abfederung der Kosten entwickelt, um Preisnachlässe bei großen Verlagen zu erzielen [vgl. [Mor15](#), 169 ff.]. Das Projekt *Deal* – ein Projekt der Hochschulrektorenkonferenz (HRK) in Zusammenarbeit mit den wissenschaftlichen Einrichtungen in Deutschland – konnte so in den vergangenen Jahren Verträge mit den Verlagen *Springer* und *Wiley* erfolgreich aushandeln [vgl. [Dea20](#)].

Um den Veränderungen des Publikationsmarktes lokal in der Bibliothek zu begegnen, wird es immer wichtiger, das Bibliotheksbudget und die Mittelallokation kosteneffizient zu planen. Dies geschieht bisher in größeren Bibliotheken durch Etatbedarfs- und Etatverteilungsmodelle [vgl. [Mor15](#), 172 ff.]. Diese Modelle basieren auf der statistischen Erhebung von bibliothekarischen Kennzahlen.

Bibliotheksstatistik reflektiert das Gestern, Heute und Morgen, indem sie die bibliothekarischen Servicedienstleistungen evaluiert und den zukünftigen Zielen und Aufgaben anpasst [vgl. [Jil04](#), 2 f. vgl. [Lai13](#), S. 462]. Im deutschen Bibliothekswesen gibt es die umfangreiche Deutsche Bibliotheksstatistik (DBS). Träger der *DBS* sind das Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (hbz), das Kompetenznetzwerk für Bibliotheken (KBN), die Kultusministerkonferenz (KMK) sowie die teilnehmenden Bibliotheken. Aufgabe der *DBS* ist die jährliche statistische Datenerhebung von Bibliothekskennzahlen. Seit 1999 werden die Daten nur noch online erfasst, ausgewertet und präsentiert [vgl. [SB08](#), S. 2]. Daneben gab es den Bibliotheksindex (BIX), der ursprünglich für die Leistungsmessung in Öffentlichen und Wissenschaftlichen Bibliotheken konzipiert wurde. Dieser wurde 2015 aber aufgrund von Finanzierungsproblemen eingestellt.

Bibliothekskarische Kennzahlen werden durch quantitative und qualitative Evaluationsverfahren erhoben. Diese Verfahren sind auf den Bestand der Bibliothek zentriert. Bestand ist nach Johannsen und Mittermaier „... die Gesamtheit aller Medien, die eine Bibliothek ihren Nutzern anbietet, sei es, dass sie diese 'physisch' besitzt, sei es, dass sie entsprechende Nutzungsrechte erworben hat.“ [[JM15](#), S. 252] Als Typen der Bestandsevaluation sind

sammelungs-, nutzungsbezogene und nutzer:innenbezogene Evaluationen zu nennen [vgl. Joh14, S. 302]. Basiert die sammlungs- und nutzungsbezogene Evaluation auf quantitativen Daten, greift die nutzer:innenbezogene Evaluation zumeist auf qualitative Daten zurück [vgl. BS04, 461 ff.].

Die sammlungsbezogene Evaluation betrifft die Größe des Bestandes und das Wachstum über die Jahre. Die Bestimmung der Bestandsstärke- und -tiefe, der Ausgewogenheit in den Bestandssegmenten sind Ziele der sammlungsbezogenen Evaluation. Ebenfalls lässt sich die Frage nach der aktuellsten Literatur im Bestand oder in einem Segment durch die sammlungsbezogene Evaluation klären [vgl. Lyo10, 48 f.].

Nutzungsbezogene Evaluation umfasst die Lesesaalnutzung, die Ausleihe vor-Ort, die Nutzung des Fernleihservices oder der Dokumentenlieferdienste sowie die Online-Nutzung von elektronischen Ressourcen [vgl. JM15, 254 ff.]. Die Frage nach den Zugriffsstatistiken auf elektronischen Ressourcen beansprucht in der nutzungsbezogenen Evaluation einen größer werdenden Raum. Die internationale Organisation *Counting Online Usage of Networked Electronic Resource* (COUNTER) gibt dazu die COUNTER-Statistiken heraus. Mitglieder der Organisation sind Verlage, Bibliotheken und Zwischenhändler. Die COUNTER-Statistiken sind mittlerweile der Quasi-Standard für die Zugriffsstatistiken auf elektronische Ressourcen geworden. Diese werden getrennt nach Art der Informationsressourcen in verschiedenen Reports herausgegeben. [vgl. JM15, 260 ff.]. Mittlerweile ist die fünfte Iteration der COUNTER-Statistiken COP 5 erschienen [vgl. Cou20]. Die Bibliotheken sind bei dem Bezug von diesen Statistiken auf die Unterstützung der Verlage angewiesen. Diese stellen unregelmäßig die COP 5-Statistiken zur Verfügung.

Ziele der nutzungsbezogenen Evaluation sind die Identifizierung von ausleihträchtigen Medienbeständen (Vormerkungs- und Rennerlisten) und die De-Akquisition schlecht oder gar nicht genutzter Titel. Ebenso kann die Evaluation von Fernleih- und Dokumentenlieferungen Hinweise auf Bestandslücken liefern. Als Konsequenz aus den COUNTER-Statistiken kann die Abbestellung von elektronischen Ressourcen resultieren.

## 2 Theoretische Grundlagen

Die nutzer:innenbezogene Evaluation ist auf den Nutzer:innenkreis der Bibliothek und deren Informationsbedürfnisse zentriert. Nutzer:innenbezogene Evaluation benutzt qualitative Daten, die sie aus Befragungen erhebt [vgl. JM15, 255 ff. vgl. Joh14, S. 302].

Die einzelnen Evaluationen vermitteln ein umfassendes Gesamtbild der Bibliothek und deren Service-Dienstleistungen. Die datengetriebenen Evaluationsauswertungen bieten Hinweise auf Optimierungen der bibliothekarischen Service-Dienstleistungen. Die Auswertungen können durch die Bibliotheksleitung aufgenommen werden und in strategische (zukünftige) Entscheidungen einfließen. So kann ein detailliertes Erwerbungsprofil und somit eine gezieltere Erwerbungspolitik entstehen. Dadurch wird das Management der Ressourcen effektiver und effizienter [vgl. Joh14, S. 297]. Gegenüber Stakeholdern kann auf der Grundlage der Evaluationen gezielt um Budget verhandelt werden.

*The purpose of the statistics is to give the management of the library or another decision-maker a satisfactory and correct picture about the situation of the library as a support to them - the statistics are the mirror of the library!* [Lai13, S. 463]

Um ein ansprechendes und korrektes Bild der Situation der Bibliothek zu präsentieren, helfen sorgsam ausgewählte Datenvisualisierungen.

## 2.2 DATENVISUALISIERUNG

Datenvisualisierungen sind wirkmächtig. Sie stellen einen Weg dar, statistische Informationen effizient zu kommunizieren [vgl. [Tuf19](#), S. 15], indem sie Daten mit visuellen Reizen ausstatten, die vom menschlichen Auge aufgenommen und vom menschlichen Gehirn schnell verarbeitet werden können [vgl. [Few09](#), S. 32]. Zusammenhänge, Trends und Ausnahmen einer großen Datenmenge sind in einer Zahlenkolonne schwieriger zu entdecken als mit einer geeigneten Datenvisualisierung. Datenvisualisierungen ermöglichen den visuellen Vergleich von verschiedenen Informationen. Sie können eine große Anzahl von Datensätzen kompakt darstellen. Datenvisualisierungen können nicht nur die Informationen von verschiedenen Blickwinkeln anzeigen, sondern die Informationen auch mit unterschiedlicher Granularität darstellen [vgl. [ML13](#), S. 245]. Visualisierungen benötigen Daten. Daten benötigen Visualisierungen, um ihren Wert besser präsentieren zu können [vgl. [Kir19](#), S. 16].

Im Folgenden werden der inhaltliche Bezugsrahmen und die Merkmale von Datenvisualisierungen näher erläutert. Datenvisualisierungen sind Verfahren der deskriptiven und explorativen Statistik beziehungsweise der explorativen Datenanalyse. Im Allgemeinen bilden sowohl die deskriptive Statistik als auch die explorative Datenanalyse keine Hypothesen. Beide treffen nur Aussagen zu vorliegenden Datensätzen. Dennoch gibt die explorative Statistik Hinweise für eine mögliche Hypothesenbildung in der weiterführenden Analyse. Die Datenvisualisierung hat sich aus den explorativen Verfahren zu einem eigenständigen Fachgebiet der Statistik beziehungsweise der Informatik entwickelt [vgl. [Bec+16](#), 28 f.].

Der Begriff der Datenvisualisierung umschreibt die visuelle Repräsentation und Präsentation von Daten, um das Verständnis zu verbessern [vgl. [Kir19](#), 15 ff.]. Er wird in Teilen der Literatur als Oberbegriff für „*Information visualization*“ und „*Scientific Visualization*“ verwendet [vgl. [Few09](#), S. 11].

Datenvisualisierungen haben das Ziel, die Analyse, Exploration und Entdeckung der Daten zu ermöglichen. Sie sollen das Verständnis der dargestellten Daten erleichtern und sind anders als Infographiken<sup>1</sup> nicht primär dafür geschaffen, Geschichten über die Informationen zu erzählen [vgl. Kir19, 20 ff.]. Sie werden vielmehr als Werkzeuge verstanden, die es ermöglichen sollen, Entscheidungen aus den visualisierten Daten zu ziehen [vgl. Cai16, S. 31].

In der Fachliteratur finden sich verschiedene Eigenschaften von Datenvisualisierungen. Cairo führt fünf Eigenschaften auf: *truthful*, *functional*, *beautiful*, *insightful* und *enlightening*. Datenvisualisierungen basieren auf gründlicher und ernsthafter Forschung (*truthful*). Sie sind funktional, das heißt sie bemühen sich die Daten genau darzustellen (*functional*). Indem Datenvisualisierungen schwer entdeckbare Beweise offenbaren, sind sie aufschlussreich (*insightful*). Darüber hinaus sollen sie für die Zielgruppe attraktiv sein (*beautiful*). Zudem sind Datenvisualisierungen aufklärend (*enlightening*), da sie Veränderungen im Denken anstoßen können [vgl. Cai16, S. 45].

Die visuelle Repräsentation der Daten erfolgt unter Verwendung graphischer Markierungen (*marks*) wie Punkt-, Linien und Balkensymbolen. Die Eigenschaften dieser Markierungen wie ihre Form, Größe oder Farbe kodieren die darunter liegenden Datenwerte. Die so kodierten Datenwerte werden dann in Diagrammen dargestellt [vgl. Kir19, 135 ff.].

Die visuelle Datenrepräsentation wird von verschiedenen Faktoren beeinflusst. Grundsätzlich ist zu überlegen, ob die Daten in Diagrammen oder Tabellen repräsentiert werden sollen. Daran anschließend ist die Frage zu klären, welche Art der Beziehung zwischen den Daten gezeigt werden soll. Für die Auswahl der Diagramme ist es wichtig zu bestimmen,

---

<sup>1</sup> Infographiken haben die Aufgabe Nachrichten zu kommunizieren. Sie bestehen aus einer Mischung von Diagrammen, Karten, Illustrationen und Text. Klarheit und Tiefe der Darstellungen sind dabei wichtig [vgl. Cai16, S. 31]. Sie werden auch als „*Explanation Graphics*“ bezeichnet und bestimmen sich dadurch, dass sie Geschehen und Ereignisse graphisch darstellen. Historisch sind Infographiken mit dem Medium der Printzeitungen und Printzeitschriften verbunden [vgl. Kir19, S. 27].

ob ein Kategorienvergleich, eine Zeitreihe, eine Rangfolge, eine relative Häufigkeit oder eine Korrelation dargestellt werden soll [vgl. Few12, S. 137].

Für die Auswahl der richtigen Datenvisualisierung sind des Weiteren die unterschiedlichen Datentypen von großer Relevanz. Datentypen „...define the nature of the values held under each variable and about each item in your dataset.“ [Kir19, S. 99] Eine Variable (Merkmal) kann qualitativ (kategorial) oder quantitativ (metrisch) sein. Die [Abbildung 2.1](#) zeigt die statistischen Datentypen nach der Nominal-, Ordinal-, Intervall-, Ratio-Systematik (NOIR) [vgl. BS10, 12 ff.] mit möglichen Aussagegehalten und Beispielen.

2



Abbildung 2.1: Statistische Datentypen mit Aussagegehalten und Beispielen

Unterschieden werden die Merkmale ferner nach diskret und stetig. Ein diskretes Merkmal kann auf der Basis der natürlichen Zahlen abzählbar viele Merkmalsausprägungen annehmen. So ist zum Beispiel die Größe des Medienbestandes einer Bibliothek ein diskretes Merkmal, da es keine halben oder viertel Medien gibt. Im Gegensatz dazu können die Merkmalsausprägungen eines stetigen Merkmals jeden beliebigen Wert annehmen. So ist

<sup>2</sup> Manchmal ist auch nur die Unterscheidung zwischen nominalen, ordinalen und metrischen Merkmalen in der wissenschaftlichen Literatur anzutreffen [vgl. Cle11, S. 20]. Unter Berücksichtigung der großen Vielfalt (variety) der Daten, schlägt Kirk eine Erweiterung der NOIR-Systematik um einen textuellen Datentyp vor [vgl. Kir19, S. 100].

zum Beispiel die Raumtemperatur ein stetiges Merkmal. Qualitative Merkmale können nur diskret sein, während quantitative Merkmale sowohl diskret als auch stetig sein können [vgl. [Kir19](#), 102 f.]. Für kategoriale Merkmale eignen sich Balken- und Kreisdiagramme eher als Liniendiagramme, da kategoriale Daten nur diskret sein können.

Der Einsatz von Datenvisualisierungen wird außerdem bestimmt von der Größe der Datenmenge. Die Darstellung einer großen Datenmenge kann zum Beispiel durch die Wahl eines bestimmten Diagrammes überladen wirken. So stößt die Darstellung einer Datenmenge mit vielen Kategorien durch Balken- und Kreisdiagramme in der Übersichtlichkeit an Grenzen und kann den Wert der zu erzielenden Aussage verwischen [Vgl. [Few12](#), 5 ff.].

Daten können softwareseitig mit Tabellenkalkulationsprogrammen visualisiert werden. In Data-Science-Projekten können außerdem verschiedene Frameworks zum Einsatz kommen. Populär sind die Bibliotheken Matplotlib, Seaborn, Plotly für die Programmiersprache Python. Für die Programmiersprache R gibt es ebenfalls vielfältige Möglichkeiten der Visualisierung mit ggplot. Einen Überblick zeigt die Webseite „The Chart maker“ [[WK20](#)]. Eine mächtige und verbreitete JavaScript-Bibliothek zur Datenvisualisierung ist D3.js.

Daten entstehen in wissenschaftlichen Experimenten, aus statistischen Erhebungen wie dem Census, auf Kassenzetteln, in Smartphones, in Unternehmen oder in Einrichtungen wie Bibliotheken. In Unternehmen werden Datenvisualisierungen eingesetzt, um auf der Managementebene die operative Planung und Entscheidungsfindung zu unterstützen. Dabei ist Datenvisualisierung ein Bestandteil eines ganzheitlichen Prozess, der unter dem Begriff Business Intelligence (BI) fungiert.



## 2.3 BUSINESS-INTELLIGENCE-SYSTEME

Business Intelligence (BI)<sup>3</sup> bezeichnet allgemein Konzepte und Methoden zur Entscheidungsfindung, die auf erfassten Informationen beruhen. *BI-Systeme* gehören zu den Management-Support-Systemen, die seit den 1960er Jahren in Unternehmen im Einsatz sind [vgl. Gro20, S. 83].

Populär wurde der Begriff *BI* in den 1990er Jahren. Die Verbreitung des Begriffs fiel zusammen mit der Entwicklung einheitlich strukturierter und dauerhaft verfügbarer Datenbanken in Unternehmen, sogenannten Data Warehouses (DWH). Grund für diese Entwicklung waren die immer neuen Informationsbedürfnisse auf Managementebene, die schnell befriedigt werden sollten. Die Einrichtung dieser Datenbanken zielt ab auf eine umfassende Informationsversorgung durch Verdichtung der Unternehmensdaten [vgl. AM17, 268 ff.].

*BI* beschreibt einen integrierten, unternehmensspezifischen, IT-basierten Gesamtansatz zur Unterstützung betrieblicher Entscheidungen [vgl. AM17, S. 270]. Ein Hauptmerkmal der *BI* ist nach Linden die Entscheidungsunterstützung der Managementebene [vgl. Lin16, S. 111]. Die Abgrenzung zu operativen Anwendungssystemen wie Online Transaction Processing (OLTP)-Systemen ist laut Abts und Mülder ein weiteres wichtiges Merkmal der *BI-Systeme* [vgl. AM17, S. 267].

Die wesentlichen Schichten eines *BI-Systems* umfassen die Bereiche der internen und externen Datenlieferanten, den Komplex der Datenintegration und -aufbereitung, das Gebiet der Datenspeicherung - und bereitstellung im *Data Warehouse* und den Zweig der Datenanalyse und -präsentation [vgl. Lin16, 126 ff. vgl. KBM10, S. 8].

Die Schichten der *BI-Systeme* können in einem Referenzarchitekturmodell abgebildet werden. Die Referenzarchitektur kann auch die Datenflüsse zwischen den verteilten Systeme-

---

<sup>3</sup> Synonym wird auch manchmal der Begriff Analytische Informationssysteme oder Business-Intelligence-Systeme gebraucht

Einheiten und Komponenten erfassen [vgl. Lin16, 126 ff.]. Eine schematische Darstellung in Form eines solchen Referenzarchitekturmodell zeigt die [Abbildung 2.2](#).



Abbildung 2.2: Schichten eines Business-Intelligence-Systems

Von den internen und externen Datenlieferanten werden die Daten im Bereich der Datenintegration und -aufbereitung mithilfe von *ETL-Prozessen* bearbeitet. In einem ersten Schritt werden die Daten aus den *OLTP-Systemen* oder aus anderen Datenquellen extrahiert. Der anschließende Transformationsprozess wandelt die Daten in ein homogenes Format um. Dabei handelt es sich um einen vierstufigen Prozess. Die Daten werden mit Hilfe zum Teil automatischer Verfahren bereinigt, harmonisiert, verdichtet (aggregiert) und angereichert. Bereinigt werden die Daten unter anderem von syntaktischen und semantischen Mängeln. Unterschiedliche Codierungen der Daten werden durch die Harmonisierung beseitigt. Des Weiteren werden die Daten verdichtet, das heißt es werden Summationen von den Daten auf verschiedenen Ebenen durchgeführt und gespeichert.

Die Berechnung und Speicherung wichtiger Kennzahlen geschieht durch das Verfahren der Anreicherung [vgl. Gro20, S. 86; vgl. AM17, 277 f.]. Schließlich werden die Daten in einem Ladeprozess in das *Data Warehouse* geladen [vgl. Lin16, 129 ff.].

Die Datenspeicherung und -bereitstellung erfolgt im *DWH*. Dieser Bereich ist nach Linden zentral für die *BI-Referenzarchitektur* [vgl. Lin16, S. 135]. Ein *Data Warehouse* ist ein logisch zentralisiertes Datenhaltungssystem. Dieses ist physisch von den operativen Anwendungssystemen getrennt und stellt eine harmonisierte Datenbasis für betriebswirtschaftliche Analysen bereit [vgl. MB00, S. 135].<sup>4</sup> Ein *Data Warehouse* zeichnet sich durch die vier Merkmale themenorientiert, zeitorientiert, integriert und nicht-volatil aus [vgl. Inm05, 29 f. vgl. AM17, 271 f. vgl. Lin16, 136 f.]. Die Speicherung der Daten erfolgt nach Themenschwerpunkten und orientiert sich am Informationsbedarf des Unternehmens. Die zeitorientierte Speicherung der Daten ermöglicht Zeitreihenanalysen auf historischen Daten. Für die Schaffung einer homogenen Datenbasis integriert das *DWH* die Daten aus heterogenen Datenquellen [vgl. Lin16, S. 136].

Die Datenintegration in das *Data Warehouse* kann auf Grundlage multidimensionaler Datenmodelle wie sogenannten Datenwürfeln erfolgen. Multidimensionale Datenwürfel können n-Dimensionen haben. Sie bestehen mit Fakten und Dimensionen aus zwei Elementen. Fakten sind Kennzahlen, die in den Zellen des multidimensionalen Würfels enthalten sind. Dimensionen sind Entitäten, die um einen Fakt angeordnet sind. Die Dimensionen spannen die Kanten des multidimensionalen Datenwürfels auf. Die Betrachtung der Kennzahlen ist so in einem multidimensionalen Datenmodell anhand verschiedener Dimensionen möglich [vgl. Far11, 13 ff., 21 f. vgl. KBM10, 66 f.].

Ein *Data Warehouse* bietet des Weiteren eine nicht-volatile Speicherung der Daten an. Damit ist sichergestellt, dass auf diesen Daten längerfristige Analysen durchgeführt werden können [vgl. Lin16, S. 136]. In die Konzeptionierung eines *Data Warehouses* sollten deswegen Archivierungskonzepte und zudem Überlegungen zu den Aktualisierungszyklen

---

<sup>4</sup> Alternativen zum *Data Warehouse* wären verschiedene *Data Marts*, die kleinere Datenspeichereinheiten darstellen und sich inhaltlich an späteren Abfrage- und Auswertungszwecken orientieren.

der Daten für das *DWH* miteinfließen. Die Archivierungskonzepte sorgen dafür, dass veraltete Datenbestände gesichert und komprimiert werden. Die Aktualisierungszyklen legen fest, in welcher periodischen Abfolge die Daten aktualisiert werden, entweder zu Zeitpunkten der Änderungshäufigkeit der Daten im operativen Anwendungssystem, in periodischen Zeitabständen oder aber auch in Echtzeit [vgl. [Lin16](#), S. 137].

Bei der Umsetzung mit Datenbanktechnologien besteht die Möglichkeit, das multidimensionale Modell in einer relationalen Datenbank umzusetzen. Realisiert werden kann dies mit einer logischen Datenmodellierung durch ein Star- oder Snowflake-Schema [vgl. [Lin16](#), 177 f.].

Die letzte Schicht eines *BI-Systems* umfasst die Datenanalyse und -präsentation. Die Datenanalyse kann verschiedene Auswertungskonzepte aufweisen. Die Auswertungskonzepte bieten spezifische Funktionen für Analysen an. Im Rahmen herkömmlicher *BI-Systeme* werden Online Analytical Processing (OLAP)-Verfahren angewendet. *OLAP* ist eine Anfragetechnik für die Analyse multidimensionaler oder relationaler Daten. Diese Technik erlaubt es, Daten mit *OLAP*-Funktionen wie Drill-Down, Roll-up, Slice oder Dice auf verschiedenen Stufen mit unterschiedlichen Sichtweisen darzustellen. So ist eine Verfeinerung der Analyseergebnissen (Drill-Down) und deren Aggregation möglich. Darüber hinaus können Teilmengen durch Slicing und Dicing gebildet werden [vgl. [AM17](#), 283 f.]. Durch *OLAP*-Anwendungen können die multidimensionalen Datenstrukturen interaktiv ausgewertet werden.

Ein anderes Verfahren ist Data Mining. Data Mining benutzt verschiedene statistische und mathematische Verfahren, um Muster und Trends in vor allem großen Datenmengen zu entdecken. Klassische Data-Mining-Aufgaben sind Ausreißer-Erkennung, Klassifikation, sowie die Cluster-, Assoziations- und Regressionsanalysen [vgl. [HKP12](#), 15 ff.]. Data Mining wird entweder synonym gesetzt mit dem Prozess der Knowledge Discovery in Databases (KDD) oder als Teilphase dieses Prozesses beschrieben [vgl. [HKP12](#), S. 6; vgl.

[Lin16](#), 142 f.]. Data-Mining-Verfahren können als zusätzliche Auswertungsverfahren zu *OLAP*-Verfahren hinzutreten.

Die Datenpräsentation umfasst die strukturierte und visuelle Darstellung der zuvor angewendeten Analyseverfahren. Die Präsentation der Daten kann durch Reporting oder Dashboards erfolgen. Über Reporting-Tools können Standardberichte generiert werden. Diese statischen Berichte können nicht verändert werden. Für die Darstellung der Daten in diesen Berichten werden Tabellen, Listen und Diagramme verwendet. Der Bericht kann bereitgestellt werden in PDF-Format [vgl. [Ban16](#), S. 114].

Im Gegensatz dazu ermöglichen Dashboards einen interaktiven Zugang über eine Benutzungsoberfläche zu den relevanten Informationen. Nach Few ist ein Dashboard ein „... visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.“ [[Few06](#), S. 26] Dashboards werden im unternehmerischen Rahmen auch Performance Dashboards genannt [vgl. [Lin16](#), S. 154]. Diese werden verschieden klassifiziert. Sie lassen sich nach Reichweite und Zweck einteilen. Es gibt operative, taktische (analytische) oder strategische Dashboards.

Operative Dashboards überwachen und kontrollieren gegenwärtige Geschäftsprozesse. Durch die hohe Aktualisierungsfrequenz der Daten kann schnell in die betrieblichen Prozesse steuernd eingegriffen werden [vgl. [Eck11](#), 11 f. vgl. [Few06](#), 30 f.].

Taktische oder analytische Dashboards konzentrieren sich auf verschiedene Bereiche des Unternehmens und bieten Möglichkeiten einer anforderungsgerechten und lösungsorientierten Analyse durch eine Bewertung der Daten auf mehreren Detailebenen.

Strategische Dashboards repräsentieren hochaggregierte Kennzahlen, die langfristige Ziele und deren Erreichungsgrade visualisieren. Sie werden zur Kommunikation und Kollaboration auf oberster Managementebene genutzt [vgl. [Lin16](#), 155 f.].

Die Grenzen zwischen den Dashboards bezüglich Reichweite und Zweck sind aber nicht trennscharf. So kann es Überlappungen zwischen strategischen, taktischen und operativen Dashboards geben [vgl. [Eck11](#), S. 121].

Neuere Entwicklungen in den *BI-Systemen* führen weg von der strikten Trennung zwischen OLTP und OLAP und zur Auflösung der *DWH*. Anstelle derer treten Data Lakes. Data Lakes speichern im Gegensatz zu den Data Warehouses die Rohdaten in strukturierter oder unstrukturierter Form. Mitunter wird dabei auf die aufwändigen *ETL-Prozesse* verzichtet [vgl. [Gro20](#), S. 86], und die Daten werden zum Analysezeitpunkt bearbeitet.

## 3 AUSGANGSSITUATION

Im folgenden Kapitel wird die wissenschaftliche Spezialbibliothek des *Max-Planck-Institutes für empirische Ästhetik* porträtiert, um die Ausgangslage zu umreißen. Anschließend werden die bibliothekarischen Informationsdienstleistungen der Bibliothek skizziert und der Frage nachgegangen, welche statistischen Daten aggregiert und ausgewertet wurden. Diese Daten stellen die Basis für die spätere Konzeption und Entwicklung eines datengetriebenen Unterstützungssystems dar.

### 3.1 BIBLIOTHEK

#### 3.1.1 ALLGEMEINES

Die Spezialbibliothek wurde im Zuge der Gründung des *MPI EA* in Frankfurt im Jahr 2013 gegründet. Die Aufgabe des Institutes ist die interdisziplinäre Erforschung empirischer Fragestellungen der Ästhetik. Das Institut besteht derzeit aus drei Abteilungen *Sprache und Literatur*, *Musik* und *Neurowissenschaften* sowie einigen Forschungsgruppen.

Die Bibliothek ist eine Serviceeinrichtung des Institutes und dient mit ihren Informationsdienstleistungen der Forschung. Zentral ist dabei die Informationsversorgung der Forschenden. Die benötigten Informationen sind Bücher, Zeitschriften, Zeitschriftenartikel sowohl in gedruckter als auch in elektronischer Form. Der Bibliotheksbestand ist somit hybrid. Er besteht sowohl aus gedruckten als auch Online-Medien sowie audiovi-

suellen Materialien. An Bestand umfasst die Bibliothek zirka 11.000 Bücher, 30 laufende Zeitschriften, knapp über 200 audiovisuelle Medien sowie Online-Datenbanken und Online-Zeitschriften.

#### 3.1.2 ORGANISATORISCHE EINBETTUNG

Um alle Informationsbedarfe der Forscher:innen zu befriedigen, wird die Bibliothek in ihren Aufgaben von der *max-planck-digital-library* unterstützt. Deren Portfolio umfasst vorrangig die zentrale Lizenzierung von relevanten elektronischen Informationsressourcen, die Bereitstellung von Softwarelösungen, das Betreiben des Publikationsrepositoriums *Pu-Re.MPG* der *Max-Planck-Gesellschaft* (MPG) sowie das Vorantreiben von Open-Access-Initiativen.

Darüber hinaus ist die Spezialbibliothek Teil des *HeBis-Verbundes*. Seit Ende 2014 finden die Geschäftsprozesse der Katalogisierung und der Erwerbung im *Zentralsystem* (CBS) und im Lokalsystem *Lokalsystem* (LBS) vom *OCLC* statt. Im *Online-Katalog* (OPAC) befinden sich Bücher, ausgewählte E-Books und Zeitschriften (Print und Online) der Institutsbibliothek. Lokal lizenzierte Datenbanken finden sich dagegen nicht im Katalog. Das *LBS* wird gehostet und betreut vom Lokalsystem-Team Frankfurt. Als Service-Leistungen werden der Bibliothek besondere Funktionalitäten für das *Zentralsystem* und Statistiken aus dem *LBS* von dem Lokalsystem-Team bereitgestellt.

#### 3.1.3 INFORMATIONSDIENSTLEISTUNGEN

Das Bibliotheks-Team des *MPI EA* ist verantwortlich für den Ablauf und Organisation der bibliothekarischen Informationsdienstleistungen. Eine Übersicht der Informationsdienstleistungen, aufgeschlüsselt nach den Basisfunktionen einer Bibliothek [Rös+19, S. 204 f.], zeigt [Tabelle 3.1](#). Die zentralen Informationsdienstleistungen der Spezialbibliothek bestehen aus der Sammeltätigkeit und dem Benutzungsservice. Seit der Institutsgründung wird neben dem nutzungsorientiertem Bestandaufbau ebenfalls eine planmäßige



Bestandsentwicklung betrieben. Das Erwerbungsprofil der Bibliothek leitet sich aus dem Forschungsauftrag des Institutes ab und umfasst dementsprechend die Erwerbung von Informationsressourcen, die sich den theoretischen und empirischen Fragestellungen der Ästhetik widmen.

Die Dienstleistungsbereiche der Benutzung sind zuständig für die Organisation der Fern- und Ortsleihe von Informationsressourcen, die nicht in das Erwerbungsprofil der Spezialbibliothek fallen. Ferner sind diese für die Informationsbeschaffung sowohl über Dokumentenlieferdienste als auch für die Akquise von einzelnen Zeitschriftenaufsätzen zuständig.

Basisfunktion	Beschreibung
Benutzung	Ausleihe, Lesesaalnutzung, Organisation der Lieferdienste (Fern und Ortsleihe, Dokumentenlieferdienste)
Management techn. Infrastruktur	<i>PuRe.MPG</i> , Medien-Datenbank
Ordnen	Aufstellungssystematik (RVK)
Sammeln und Erschließen	geplanter Bestandsaufbau, Integrierter Geschäftsgang Medienerwerbung und Medienschließung, besondere Materialien
Vermitteln	Literaturrecherche, Nutzung elektronischer Ressourcen, Urheberrecht und Publikationsberatung

Tabelle 3.1: Informationsdienstleistungen nach Basisfunktionen der Spezialbibliothek

Weitere Informationsdienstleistungen sind die Betreuung des Publikationsrepositoriums *PuRe.MPG* des Institutes, spezielle Beratungsdienstleistungen zum Urheberrecht und zum Publishing sowie klassische Auskunft- und Informationsdienste. Seit Beginn 2016 geschieht die Ausleihe der Medien über ein Selbstverbuchungssystem.

### 3.1.4 EVALUATION DER INFORMATIONSDIENSTLEISTUNGEN

Zu fast jeder Informationsdienstleistung der Spezialbibliothek werden quantitative Daten elektronisch generiert. [Tabelle 3.2](#) zeigt Daten, die bereits jetzt in der ein oder anderen Form aggregiert und ausgewertet werden. Die Tabelle stellt nach den Evaluationstypen

### 3 Ausgangssituation

dar, ab wann und wie regelmäßig die Statistiken erfasst werden. Ferner bietet sie einen Überblick darüber, in welchem Format die Daten vorliegen, über die Quellen aus der die Daten stammen und ob die Daten bereits systematisch ausgewertet und/oder visualisiert werden.

Evaluationstyp	Basisfunktion	Daten	Zeitraum	Turnus	Quelle	Format	Syst. Auswertung	Visualisierungen
Nutzungsbezogen	Benutzung	Ausleihzahlen Bibliotheksbestand	2016-	monatlich	LBS	Mail, xlsx	nein	-
Nutzungsbezogen	Benutzung	Ausleihzahlen Lieferdienste	2015-	monatlich	intern	xlsx	ja	teilweise, Liniendiagramm
Nutzungsbezogen	Benutzung	Besonders nachgefragte Medien (OPAC)	2017-	monatlich	LBS	Mail, txt	nein	-
Nutzungsbezogen	Sammeln	COP 5-Statistiken elektr. Ressourcen	2013-	unbekannt	mpdl	csv, tsv, txt	nein	-
Nutzungsbezogen	Benutzung	Lesesaalnutzung	2015-	wöchentlich	intern	xlsx	nein	-
Sammlungsbezogen	Sammeln	Budget nach Kostenstellen	2018-	monatlich	LBS	Mail, txt	ja	-
Sammlungsbezogen	Sammeln	Umsatz nach Lieferanten	2018-	monatlich	LBS	Mail, txt	ja	Balken- und Kreisdiagramm
Sammlungsbezogen	Sammeln	Größe und Art des Bestandes	2014-	jährlich	LBS, intern	csv	nein	-
Sammlungsbezogen	Sammeln	Neuerwerbungslisten	2014-	monatlich	LBS, intern	tsv	nein	-

Tabelle 3.2: Liste der Dienstleistungsbereiche zu denen statistische Daten erhoben werden

Intern erfasst die Bibliothek monatlich die Daten der Ausleihe über die Lieferdienste. Unterschieden wird in der Erfassung nach Medientypen, nach Ausleihort und Ausleihart. Zudem werden wöchentlich Statistiken zur Nutzungshäufigkeit des Lesesaals geführt. Jährlich wird die Bestandsgröße nach Medientyp für die Buchhaltung ermittelt.

Die Ausleihzahlen des Bibliotheksbestandes werden bei Bedarf durch das Lokalsystem-Team ermittelt und an die Bibliothek geschickt. Diese liegen kumulativ nach Ausleihanzahl des einzelnen Titels oder nach Jahr und der Identifikationsnummer des Titeldatensatzes im CBS vor. Ebenfalls stehen die Ausleihzahlen als Rohdaten, in denen jede Titelausleihe aufgeführt wird, zur Verfügung.

Anzahl von cum_loans		Spaltenbeschriftungen					
Zeilenbeschriftungen		2016	2017	2018	2019	2020 (Leer)	Gesamtergebnis
5	74290844			1			1
6	74294621			1	1		2
7	74294826					1	1
8	74294926			1	1	1	3
9	74294971			1	1		2
10	74295032				1	1	2
11	74295067			1	1		2
12	74295100		1	1	1	1	4
13	74295652		1	1	1	1	4
14	74295755		1	1		1	4
15	74298047		1	1			2

Abbildung 3.1: Anzahl der Titelausleihen nach Identifikationsnummer und Jahr

Monatlich bekommt die Bibliothek kumulative Budget- und Umsatzübersichten der Kostenstellen und der Lieferanten zugeschickt. Die Budgetübersicht gibt den monatlichen Stand der Budgets (Bindungen und Ausgaben) der einzelnen Kostenstellen zwischen den ACQ-Jahresübergängen aus.<sup>5</sup> Die Kostenstellen bilden die einzelnen Abteilungen und zum Teil die Forschungsgruppen des Institutes ab.

Die Umsatzübersicht summiert die monatlichen Ausgaben nach Lieferanten bezogen auf das Kalenderjahr. Kriterium ist hierbei das Erfassungsdatum der Rechnung.

Bearbeitet werden nur die Umsatzübersichten der Lieferanten, um die Umsatzverteilung nach Lieferanten zu steuern. [Abbildung 3.2](#) zeigt ein Beispiel<sup>6</sup> vom November 2020 für die Budget- und Umsatzübersichten des Monats Oktober 2020, die vom Lokalsystem-Team als email zur Verfügung gestellt werden.

Budgetuebersicht Bibliothek MPI fuer Empirische Aesthetik						Stand: Nov 1 2020 6:16AM	
Stand: Nov 1 2020 6:16AM							
Budgetuebersicht alle Budgets						Lieferant	Umsatz (EUR)
S	Bezeichnung	Ansatz	Bindungen	Ausg. ges.	Bestellvol.		
bet m 20	Betriebsrat	0.00	0.00	0.00	0.00	Antiquariat	0.00
bib m 20	Bibliothek	0.00	0.00	0.00	0.00	Schildkröte Buchhandlung	0.00
bib z 20	Bibliothek	0.00	0.00	0.00	0.00	Rosa Arbeiter	0.00
fam m 20	Fachgeschichte Musik	0.00	0.00	0.00	0.00	HDGDL Buch	0.00
hmm z 20	Histories of Music Mind and Body	0.00	0.00	0.00	0.00	Eschenbach Verlag	0.00
inf m 20	IT	0.00	0.00	0.00	0.00	Sappho bookshop	0.00
lit m 20	Literatur und Sprache	0.00	0.00	0.00	0.00	Unser Glück Verlag	0.00
mus m 20	Musik	0.00	0.00	0.00	0.00	Schwalben Medien	0.00
neu m 20	Neurowissenschaft	0.00	0.00	0.00	0.00	Golden Leaves Musikverlag	0.00
vew l 20	Verwaltung	0.00	0.00	0.00	0.00	Tassen	0.00
vew m 20	Verwaltung	0.00	0.00	0.00	0.00	schwarze Botin	0.00
wid m 20	Wissenschaftliche Dienste	0.00	0.00	0.00	0.00	Buchhandlung Wal	0.00
						FuP Informationen	0.00
						Leslie Ben Ron	0.00
						Lic Vinyl	0.00
Summe		0.00	0.00	0.00	0.00	Summe	0.00

Abbildung 3.2: Monatliche Budget- und Umsatzübersicht

Die Ausgaben für die lokal lizenzierten Datenbanken fehlen in der Aufstellung der Ausgaben und werden in einer Tabelle extra geführt.

<sup>5</sup> Der ACQ-Jahresübergang variiert und der Zeitpunkt wird von der Bibliothek vereinbart mit dem Lokalsystem-Team. Der Jahresübergang geschieht meistens in den ersten beiden Januarwochen des Jahres.

<sup>6</sup> Die Zahlen wurden zurückgesetzt.

### 3 Ausgangssituation

Die *Counter 5-Statistiken* (COP 5) der Verlage werden auf einen internen Portal von der *mpdl* dem Institut zur Verfügung gestellt. Diese Statistiken verzeichnen den Zugriff innerhalb der IP-Adressbereiche des Institutes auf die elektronische Ressourcen, die konsortial durch die MPG lizenziert wurden. Darunter fallen ebooks der Verlage *Springer*, *Wiley* oder *De Gruyter*. Bisher wurden diese von der Bibliothek noch nicht gesichtet und ausgewertet.

Eine proaktive und systematische Auswertung der Entwicklung der Bestandsgröße, der Ausleihzahlen und der *COP 5*-Statistiken findet nicht oder nur unzureichend statt. Auch wird das Potential wie in [Abschnitt 2.1](#) beschrieben hinsichtlich der Budgetplanung nicht ausgeschöpft.

## 4 KONZEPTION EINER LÖSUNG

Ausgehend von der Analyse der Ausgangssituation werden im Folgenden Anforderungen für das zu entwickelnde System herausgearbeitet. Alle Anforderungen dienen als Grundlage für die Entwicklung des Proof-of-Concepts. Schließlich werden 7 Anwendungsfälle für das System im [Unterabschnitt 4.1.5](#) beschrieben, die es zu erfüllen hat. Anhand der Anforderungen und der Anwendungsfälle erfolgt im [Kapitel 5](#) die Bewertung des Proof-of-Concepts.

### 4.1 ANFORDERUNGSANALYSE

Die Vision und die Ziele des Systems werden zunächst kurz formuliert. Im Anschluss daran, werden die Rahmenbedingungen erarbeitet. Danach werden sowohl die funktionalen Anforderungen als auch die nicht-funktionalen Anforderungen festgelegt.

Die Anforderungen werden nach dem *MoSCoW-Prinzip* priorisiert. Hierbei wird unterschieden in Muss-Anforderungen (M - must), in Soll-Anforderungen (S - should), in Kann-Anforderungen (C - could) und in Anforderungen, die im Zuge der Implementierung des Proof-of-Concepts (noch) nicht umgesetzt werden (W - would / won't). Soweit nötig wurden für die Rahmenbedingungen und für die nicht-funktionalen Anforderungen Messbarkeitskriterien formuliert.

### 4.1.1 VISION UND ZIELE

Die Bibliothek des *MPI EA* soll durch das System in die Lage versetzt werden, ihr Budget effizient und bedarfsgerecht zu planen. Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen können sich relevante *Key Performance Indicators* durch das System mit Datenvisualisierungen anzeigen lassen. Relevante *KPI* sind die Umsatz- und Budgetübersichten, die Ausleihzahlen, das Bestandswachstum und die Neuerwerbungen. Ausgewählte *KPI* werden an die Institutsleitung als Standardreport verteilt.

### 4.1.2 RAHMENBEDINGUNGEN

Die Rahmenbedingungen legen die organisatorischen Anforderungen für das zu entwickelnde System fest. Darunter fallen die Anwendungsbereiche, die unterschiedlichen Zielgruppen und die technischen Anforderungen.

ID	Beschreibung	Messbarkeit	Priorisierung
R1	Zielgruppen für das System sind die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen.	-	M
R2	Die Institutsleitung ist die Zielgruppe für den Standardbericht, der aus dem System erzeugt wird.	-	M
R3	Das System wird als Desktop-Anwendung in einer Büroumgebung eingesetzt.	Das System wird nicht optimiert für mobile Endgeräte wie Smartphones oder Tablets.	M
R4	Das System wird bedarfsorientiert gestartet und beendet.	-	M
R5	Der Betrieb des Systems läuft unbeaufsichtigt.	-	M
R6	Die Entwicklungsumgebung kann identisch mit der Produktivumgebung sein.	Vergleich der Entwicklungsumgebung mit der Produktivumgebung.	C
R7	Die Softwareanforderungen sind für das System dokumentiert.	Überprüfen, ob entsprechendes Dokument vorhanden ist.	M
R8	Die Hardwareanforderungen sind für das System dokumentiert.	Überprüfen, ob entsprechendes Dokument vorhanden ist.	M
R9	Das System läuft in einem geschützten Bereich im internen Netzwerk des Institutes, der nur für das Bibliothekspersonal einsehbar ist.	Testen des Zugriffs von außerhalb des instituts-eigenen IP-Adressbereich und von bibliotheks-fremden Mitarbeiter:innen des Instituts.	W
R10	Das System wird auch von anderen Bibliotheken mit einer ähnlichen Datenlage eingesetzt.	Testen des Systems durch eine andere Bibliothek	W

Tabelle 4.1: Rahmenbedingungen

### 4.1.3 FUNKTIONALE ANFORDERUNGEN

Im Folgenden werden die funktionalen Anforderungen an das System formuliert. Das System strukturiert sich in die Bereiche des *ETL-Prozesses* und der Datenspeicherung, der Datenanalyse, der Datenpräsentation und des Standardberichtes.

Die [Tabelle 4.2](#) beschreibt die Anforderungen an das System bezüglich des *ETL-Prozesses* und der Datenspeicherung.

ID	Beschreibung	Priorisierung
F1	Das System importiert die Daten mithilfe von Skripten von einem lokalem Verzeichnis.	M
F2	Das System bereinigt automatisch die Daten von syntaktischen Fehlern und vollzieht Formatanpassungen (einheitliches Zeichenformat, einheitliches Dateiformat).	M
F3	Das System harmonisiert die Daten automatisch (Erkennen und Harmonisierung von unterschiedlichen Codierungen der Datenwerte, Erkennen und Zusammenführung von Synonymen).	M
F4	Das System ergänzt die Daten mit zusätzlichen Daten aus anderen Datenquellen (z.B. RVK-Systematik).	M
F5	Das System speichert automatisch die Daten an einem definierten Speicherort.	M
F6	Das System speichert automatisch redundant die Daten für ein Backup an einem definierten Ort.	S
F7	Das System löscht nach dem Importprozess automatisch die Daten aus dem lokalen Verzeichnis.	C
F8	Das System dokumentiert zur Fehlererkennung den <i>ETL</i> - und den Speicherprozess in einer log-Datei.	W

Tabelle 4.2: Funktionale Anforderungen - ETL-Prozess und Datenspeicherung

[Tabelle 4.3](#) legt die Anforderungen für die Datenanalyse an das System fest.

ID	Beschreibung	Priorisierung
F10	Das System bietet eine Graphische Benutzeroberfläche (GUI) an.	M
F11	Das System analysiert die Daten automatisch mit deskriptiven und explorativen Methoden der Statistik nach Kriterien, die in den Anwendungsfällen formuliert sind.	M
F12	Das System filtert die Daten nach bestimmten Kriterien, die in den Anwendungsfällen formuliert sind.	M
F13	Das System bietet die Filterung der Daten über die <i>GUI</i> auf Basis von Filtern, die von den Benutzer:innen ausgewählt werden, an.	M
F14	Das System analysiert die gefilterten Daten automatisch mit deskriptiven und explorativen Methoden der Statistik nach Kriterien, die in den Anwendungsfällen formuliert sind.	M

Tabelle 4.3: Funktionale Anforderungen - Datenanalyse

#### 4 Konzeption einer Lösung

Die Anforderungen für die Datenpräsentation und die Erstellung des Standardberichtes sind in [Tabelle 4.4](#) aufgelistet.

ID	Beschreibung	Priorisierung
F15	Das System bietet für die betreffenden Daten Datenvisualisierungen an.	M
F16	Die angebotenen Datenvisualisierungen sind hauptsächlich Linien- und Balkendiagramme.	M
F17	Das System bietet für die betreffenden Daten eine Auswahl an diesen Datenvisualisierungen an.	S
F18	Das System erstellt bedarfsorientiert ein PDF-Dokument mit den relevanten <i>KPI</i> .	M
F19	Das System speichert bedarfsorientiert Diagramme der relevanten <i>KPI</i> in einem platzsparenden Bildformat.	C
F20	Das System greift auf Diagrammbilder und Texte für die Generierung des PDF-Dokumentes automatisch zu.	M
F21	Das System öffnet das PDF-Dokument automatisch.	M

Tabelle 4.4: Funktionale Anforderungen - Datenpräsentation und Standardbericht



## 4.1.4 NICHT-FUNKTIONALE ANFORDERUNGEN

In der [Tabelle 4.5](#) sind die nicht-funktionalen Anforderungen, die Qualitätskriterien an das System beschreiben, aufgelistet.

ID	Beschreibung	Messbarkeit	Priorisierung
NF1	Das System ist portierbar auf eine andere Plattform.	Testen mit unterschiedlichen Systemen (OS, Browser).	M
NF2	Das System ist leicht erlernbar.	Schulungsdauer weniger als 1 Stunde.	M
NF3	Der Zugriff auf das System erfolgt passwortgeschützt.	-	W
NF4	Das System ist modular aufgebaut. Die einzelnen Module sind testbar.	Module sind unabhängig voneinander testbar.	S
NF5	Zur Programmierung des Systems wird freie Software (Programmiersprachen) genutzt.	Überprüfen, ob Software freie Lizenz enthält.	M
NF6	Die eingesetzte Software ist weitverbreitet und geeignet für diese Aufgabe.	Überprüfen der Verbreitung und Eignung der Software.	M
NF7	Das System ist einfach zu warten.	-	S
NF8	Das System ist unter einer Open-Source-Lizenz zu entwickeln.	Vergabe einer freien Lizenz für das zu entwickelnde System.	M
NF9	Die Reaktionszeit des Systems auf Benutzungsanfragen über die GUI beträgt weniger als 2 Sekunden.	Tests mit dem System.	S
NF10	Auf das System kann in vollem Umfang von mehreren Endgeräten gleichzeitig zugegriffen.	Tests mit mehreren Endgeräten, die gleichzeitig auf System zugreifen.	S
NF11	Das Layout des Dashboards ist strukturiert, übersichtlich und selbsterklärend.	Erfassen relevanter Informationen in weniger als 5 Sekunden.	M
NF12	Die Entwicklung des Systems erfolgt mit modernen Technologien.	-	M
NF13	Das Layout des Dashboards ist am Corporate Design des Institutes auszurichten.	-	W
NF14	Die verschiedenen Diagrammtypen werden zielgerichtet eingesetzt.	-	M
NF15	Optische Gestaltungsmerkmale und Farben der Datenvisualisierungen werden zur Verdeutlichung der Information eingesetzt.	-	M
NF16	Optische Effekte oder Animationen der Datenvisualisierungen sind sparsam einzusetzen.	-	M
NF17	Die Darstellung der Informationen ist auf die Zielgruppen zugeschnitten.	-	M
NF18	Die Datenvisualisierung zielt auf die Beantwortung der spezifischen Anwendungsfälle.	-	M

Tabelle 4.5: Nicht-funktionale Anforderungen

### 4.1.5 ANWENDUNGSFÄLLE

Im folgenden Abschnitt werden die Anwendungsfälle dargestellt, die das System beantworten soll. Für die Entwicklung des Proof-of-Concepts wurde eine Auswahl aus den bibliothekarischen Dienstleistungsbereichen getroffen. Dabei wird auf die in [Tabelle 3.2](#) aufgeführte Evaluationstypen referiert. Jeder Anwendungsfall enthält den Namen des Anwendungsfalls (Titel), den bibliothekarischen Evaluationstyp, die beteiligten Akteur:innen, das zu erreichende Ziel und den Inhalt. Zudem werden Vorbedingungen und die Anforderungen aufgeführt. Das Systemverhalten wird zu den jeweiligen Punkten ebenfalls beschrieben.

#### Anwendungsfall 1

	Beschreibung	Systemverhalten
Titel	Ausleihzahlen Bibliotheksbestand	-
Evaluationstyp	Nutzungsbezogen	-
Akteur:innen	Bibliotheksleitung, Bibliotheksmitarbeiter:innen	-
Ziel	Anzeige der Ausleihzahlen des Bestandes nach Jahr.	Das System generiert eine Anzeige mit den jeweiligen Parametern.
Vorbedingungen	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen haben Zugriff auf das System.	Das System ist im Betrieb.
Inhalt	Die Bibliotheksleitung oder die Bibliotheksmitarbeiter:innen lassen sich auf der <i>GUI</i> die Ausleihzahlen pro Jahr anzeigen.	<p>Das System filtert die betreffenden Datensätze nach Jahr. Das System zeigt diese Datensätze mit Datenvisualisierungen an.</p> <p>Das System zeigt darüberhinaus die ausleihstärksten Titel absteigend nach Anzahl und Jahr an.</p> <p>Das System zeigt darüberhinaus die Verteilung der ausgeliehenen Titel in der <i>RVK</i>-Fachsystematik pro Jahr an.</p> <p>Das System zeigt im Vergleich mit dem Bestand die Top-Fachsystematikgruppen an.</p> <p>Das System zeigt die Verteilung der Ausleihe unterschieden in Bibliotheksbestand und Buchservice an.</p>
Anforderungen	R1, R4, F2-F4, F10-F17, NF14-NF18	-

Tabelle 4.6: Anwendungsfall 1 - Ausleihzahlen Bibliotheksbestand

## Anwendungsfall 2

	Beschreibung	Systemverhalten
Titel	Ausleihzahlen bibliotheksinterne Lieferdienste	-
Evaluationstyp	Nutzungsbezogen	-
Akteur:innen	Bibliotheksführung, Bibliotheksmitarbeiter:innen	-
Ziel	Anzeige der Ausleihzahlen der bibliotheksinternen Lieferdienste.	Das System generiert eine Anzeige mit den jeweiligen Parametern.
Vorbedingungen	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen haben Zugriff auf das System.	Das System ist im Betrieb.
Inhalt	Die Bibliotheksleitung oder die Bibliotheksmitarbeiter:innen wählen den gewünschten Zeitraum aus und lassen sich die Ausleihzahlen bibliotheksinterner Lieferdienste auf der <i>GUI</i> anzeigen.	<p>Das System filtert die betreffenden Datensätze nach Gesamtzeitraum und Jahr. Das System zeigt diese Datensätze mit Datenvisualisierungen an.</p> <p>Das System zeigt die Nutzung der verschiedenen internen Lieferservices nach Gesamtzeitraum, Jahr und Monat an.</p> <p>Das System zeigt darüberhinaus die verschiedenen internen Lieferservices nach Nutzung der Institutsabteilungen an.</p> <p>Das System zeigt darüberhinaus die Entwicklung in der Nutzung der verschiedenen Lieferservices durch die Abteilungen nach Jahr und im Gesamtzeitraum an.</p>
Anforderungen	R1, R4, F2-F4 F10-F17, NF9, NF14-NF18	-

Tabelle 4.7: Anwendungsfall 2 - Ausleihzahlen bibliotheksinterne Lieferdienste

#### 4 Konzeption einer Lösung

##### Anwendungsfall 3

	Beschreibung	Systemverhalten
Titel	Lesesaalnutzung	-
Evaluationstyp	Nutzungsbezogen	-
Akteur:innen	Bibliotheksleitung, Bibliotheksmitarbeiter:innen	-
Ziel	Anzeige der Nutzung des Lesesaals während der Öffnungszeiten.	Das System generiert eine Anzeige mit den jeweiligen Parametern.
Vorbedingungen	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen haben Zugriff auf das System.	Das System ist im Betrieb.
Inhalt	Die Bibliotheksleitung oder die Bibliotheksmitarbeiter:innen lassen sich die Nutzung des Lesesaals auf der <i>GUI</i> anzeigen.	Das System filtert die betreffenden Datensätze nach Gesamtzeitraum und Jahr. Das System zeigt diese Datensätze mit Datenvisualisierungen an.  Das System zeigt die Nutzung des Lesesaals nach Monat und Jahr, gruppiert in vier Zeitschichten an.
Anforderungen	R1, R4, F2, F3, F10-F17, NF9, NF14-NF18	-

Tabelle 4.8: Anwendungsfall 3 - Lesesaalnutzung

## Anwendungsfall 4

	Beschreibung	Systemverhalten
Titel	Neuerwerbungen	-
Evaluationstyp	Sammlungsbezogen	-
Akteur:innen	Bibliotheksführung, Bibliotheksmitarbeiter:innen	-
Ziel	Anzeige der Anzahl der Neuerwerbungen nach <i>RVK</i> -Fachsystematiken pro Monat.	Das System generiert eine Anzeige mit den jeweiligen Parametern.
Vorbedingungen	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen haben Zugriff auf das System.	Das System ist im Betrieb.
Inhalt	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen wählen einen Monat auf der <i>GUI</i> aus.	Das System filtert die betreffenden Datensätze nach Monat. Das System zeigt diese Datensätze mit Datenvisualisierungen an.
	-	Das System zeigt die Neuerwerbungen des laufenden Jahres an.
	-	Das System zeigt die jährliche Bestandsentwicklung pro Monat an.
Anforderungen	R1, R4, F2-F4, F10-F17, NF9, NF14-NF18	-

Tabelle 4.9: Anwendungsfall 4 - Neuerwerbungen

#### 4 Konzeption einer Lösung

##### Anwendungsfall 5

	Beschreibung	Systemverhalten
Titel	Bestandswachstum	-
Evaluationstyp	Sammlungsbezogen	-
Akteur:innen	Bibliotheksführung, Bibliotheksmitarbeiter:innen	-
Ziel	Anzeige des Wachstums des Bibliotheksbestandes insgesamt und nach einzelnen <i>RVK</i> -Fachsystematiken.	Das System generiert eine Anzeige mit den jeweiligen Parametern.
Vorbedingungen	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen haben Zugriff auf das System.	Das System ist im Betrieb.
Inhalt	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen lassen sich auf der <i>GUI</i> das Bestandswachstum insgesamt und nach <i>RVK</i> -Fachsystematiken anzeigen.	<p>Das System filtert die betreffenden Datensätze für die <i>RVK</i>-Systematikstellen. Das System zeigt diese Datensätze mit Datenvisualisierungen an.</p> <p>Das System zeigt die Top<i>RVK</i>-Fachsystematiken des Bestandes pro Jahr an.</p> <p>Das System zeigt darüberhinaus die Gesamtzahl der Titel nach Jahren an.</p> <p>Das System zeigt darüberhinaus die Anzahl der Titel nach Medienart an.</p> <p>Das System zeigt die Top<i>RVK</i>-Fachsystematiken des Bestandes insgesamt an.</p>
Anforderungen	R1, R4, F2-F4, F10-F17, NF9, NF14-NF18	-

Tabelle 4.10: Anwendungsfall 5 - Bestandswachstum

## Anwendungsfall 6

	Beschreibung	Systemverhalten
Titel	Umsatz- und Budgetübersicht	-
Evaluationstyp	Sammlungsbezogen	-
Akteur:innen	Bibliotheksführung, Bibliotheksmitarbeiter:innen	-
Ziel	Anzeige der Umsatz- und Budgetübersicht für den Gesamtzeitraum und das laufende Jahr.	Das System generiert eine Anzeige mit den jeweiligen Parametern.
Vorbedingungen	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen haben Zugriff auf das System.	Das System ist im Betrieb.
Inhalt	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen können sich den Lieferanten auswählen und den Gesamtumsatz und den Umsatz pro Jahr ansehen.	Das System filtert die betreffenden Datensätze für den Lieferanten. Das System zeigt diese Datensätze mit Datenvisualisierungen an. Das System zeigt den Umsatz im laufenden Jahr und den Jahresdurchschnittsumsatz dieses Lieferanten. Das System zeigt die umsatzstärksten Lieferanten im Gesamtzeitraum an. Das System zeigt den Gesamtumsatz im Gesamtzeitraum und im laufenden Jahr.
	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen können sich die Budgetübersicht ansehen.	Das System filtert die betreffenden Datensätze für die Kostenstellen. Das System zeigt diese Datensätze mit Datenvisualisierungen an.
	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen können sich den Verlauf des Budgets und der Umsätze über den Gesamtzeitraum und über das laufende Jahr anzeigen lassen.	Das System zeigt das Budget und den Umsatz für den Gesamtzeitraum und für das laufende Jahr an.  Das System zeigt darüberhinaus Entwicklung für das Budget und den Umsatz über den Gesamtzeitraum an. Das System zeigt die kostenintensivsten Kostenstellen im Gesamtzeitraum an.
Anforderungen	R1, R4, F2, F3, F10-F17, NF9, NF14-NF18	-

Tabelle 4.11: Anwendungsfall 6 - Umsatz- und Budgetübersicht

#### 4 Konzeption einer Lösung

##### Anwendungsfall 7

	Beschreibung	Systemverhalten
Titel	Standardbericht	-
Evaluationstyp	-	-
Akteur:innen	Bibliotheksleitung, Bibliotheksmitarbeiter:innen, Institutsleitung	-
Ziel	Generierung eines Standardberichts mit den relevanten <i>KPI</i> der Bibliothek zur Vorlage bei der Institutsleitung.	Das System generiert eine Anzeige mit den jeweiligen Parametern.
Vorbedingungen	Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen haben Zugriff auf das System.	
Inhalt	Die Bibliotheksmitarbeiter:innen lösen ein Skript zur Generierung des Standardberichtes aus.	Das System greift auf Bilder und Texte an einem definierten Speicherort zu.
		Das System generiert aus diesen Bildern und Texten automatisch ein PDF-Dokument.
		Das System öffnet das PDF-Dokument automatisch.
	Die Bibliotheksleitung oder die Bibliotheksmitarbeiter:innen speichern das PDF-Dokument an einem Speicherort.	Speicherung des PDF-Dokument systemunabhängig.
	Die Bibliotheksleitung oder die Bibliotheksmitarbeiter:innen verteilen das PDF-Dokument an die Institutsleitung.	-
Anforderungen	R2, F18-F21, NF14-NF18	-

Tabelle 4.12: Anwendungsfall 7 - Standardbericht



# 5 DISKUSSION DER UMSETZUNG

Im folgendem Kapitel wird das Design des Prototypen besprochen. Dabei wird zunächst auf die technischen Details der Implementierung eingegangen. Danach folgt die Vorstellung der Systemarchitektur und der zugehörigen Teilsysteme. Es wird aufgezeigt, wie die einzelnen Teilsysteme funktionieren und ineinander greifen. Dies geschieht unter anderem an ausgewählten Programmcode-Beispielen.<sup>7</sup>

Im Anschluss daran wird die praktische Funktionsweise des Systems skizziert. Neben den technischen Voraussetzungen wird dabei die Vorgehensweise des Datenimports beschrieben und auf das Layout und die Darstellung der Daten im Dashboard eingegangen. Abschließend findet eine Bewertung des Standes der Umsetzung des Systems statt anhand der Anforderungen und der Anwendungsfälle, die im [Kapitel 4](#) formuliert sind. Dabei wird auf die allgemeine Bewertung und auf Datengrundlage kurz eingegangen.

## 5.1 IMPLEMENTIERUNG

### 5.1.1 TECHNISCHE DETAILS DER IMPLEMENTIERUNG

Das Proof-of-Concept wurde mittels der Programmiersprache Python umgesetzt. Python ist eine höhere Programmiersprache. Sie ist weitverbreitet [vgl. [Lou21](#)] und besitzt eine einfache Syntax. Besonderheiten von Python sind der Verzicht auf geschweifte Klammern

---

<sup>7</sup> Für den vollständigen Programmcode und dessen Dokumentation sei auf das GitHub-Repositorium verwiesen.<sup>8</sup>

und Interpunktionen nach Anweisungen. Die Anweisungen sind durch Einrückungen strukturiert und nicht durch öffnende und schließende Klammern getrennt. Des Weiteren zeichnet sich Python durch eine dynamische Typisierung aus und ist dadurch sowohl für Skripte als auch für die schnelle Entwicklung von Anwendungen geeignet. Python erlaubt die Aufteilung von Programmen in Modulen, die in anderen Python-Programmen wiederverwendet werden können [vgl. [Pyt21a](#)]. Sie ist für alle wichtigen Plattformen frei verfügbar und verfügt über eine große Dokumentation [vgl. [Pyt21c](#)]. Wie andere Programmiersprachen besitzt Python eine umfangreiche Standardbibliothek. Dazu gibt es noch eine Vielzahl von Modulen, Programmen und Werkzeugen von Drittanbietern [vgl. [Pyt21b](#)].

Genutzt wurde für das Projekt insbesondere die Python-Bibliothek pandas, die eine Open-Source-Bibliothek für Datenanalyse und Datenmanipulation darstellt [vgl. [Pan21](#)]. Wichtige Konzepte dieser Bibliothek sind der pandas Dataframes und die pandas Series, die pandas Objekte sind. Der pandas Dataframe ist eine zweidimensionale tabellarische Datenstruktur mit beschrifteten Achsen (Spalten und Reihen). In ihn werden die Daten unter anderem aus csv- oder xlsx-Dateien über pandas Funktionen geladen. Eine pandas Series kann Bestandteil eines Dataframes sein. Der Datentyp entspricht eines eindimensionalen Arrays. [Abbildung 5.1](#) zeigt anhand der Umsatzdaten die ersten fünf Reihen eines pandas Dataframe und einer pandas Series desselben Dataframes.

	Lieferant	Umsatz (EUR)	Datum	Lieferant	Abk.		
0	Antiquariat	0.00	2020-10-01	Antiquariat		0	Antiquariat
1	Schildkröte Buchhandlung	0.00	2020-10-01	Schildkröte Buchhandlung		1	Schildkröte Buchhandlung
2	Rosa Arbeiter	0.00	2020-10-01	Rosa Arbeiter		2	Rosa Arbeiter
3	HDGDL Buch	0.00	2020-10-01	HDGDL Buch		3	HDGDL Buch
4	Eschenbach Verlag	0.00	2020-10-01	Eschenbach Verlag		4	Eschenbach Verlag

Abbildung 5.1: pandas Dataframe (links) und pandas Series (rechts)

Zugriff auf die Series erfolgt über den Spaltenkopf 'Lieferant Abk.' der die Series über diesen anspricht. Sowohl der Dataframe als auch die Series haben einen Index, über den auf die einzelnen Werte zugegriffen werden kann.

Auf dem Dataframe, der als sogenannter Container für die Series dient, können verschiedene Operationen der Datenanalyse und -manipulation erfolgen. Ähnlich wie bei Abfragen einer Datenbank lassen sich verschiedene Funktionen wie `sort()`- oder `groupby()` in Zusammenhang mit Aggregatfunktionen wie `mean()`, `median()` oder `sum()` auf den Series durchführen. Auch gibt es eine Vielzahl von Funktionen, um Datasets rasch zu explorieren. Zum Beispiel bietet sich Jupyter Notebook als Entwicklungsumgebung dazu hervorragend an, da include dieser der Code zeilenbasiert ausgeführt werden kann.<sup>9</sup> Die Werte der Zellen können verschiedene Datentypen annehmen. Zum Beispiel können die Datumswerte der Spalte 'Datum' in `datetimeformat`-Objekte umgewandelt werden.

In Verbindung mit pandas ist Python neben R im wissenschaftlichen Kontext für Data-Science-Projekte deshalb sehr beliebt.

Es gibt eine Vielzahl an Bibliotheken für Python, die für die Entwicklung von interaktiven Datenvisualisierungen verfügbar sind. Zu nennen wären Bokeh [vgl. [Van21](#)], Altair [vgl. [Alt21](#)] und die Graphic Libraries von Plotly [vgl. [Plo21f](#)]. Für das Projekt wurde sich für Plotly Express entschieden und dieses hauptsächlich genutzt. Plotly Express ist eine Graphische Bibliothek für Python, R und Javascript. Es ist eine Weiterentwicklung (Wrapper) der Bibliothek Plotly Graph Objects. Plotly Express besitzt eine einfachere Syntax bei fast annäherender Feature-Gleichheit mit Plotly Graph Objects.[[Plo21f](#)]. So können mit wenigen Zeilen Python-Code interaktive Grafiken erzeugt werden. An interaktiven Basisfunktionalitäten bietet Plotly Express zum Beispiel Hover-Informationen der Datenpunkte, Zoom-In und Zoom-Out-Möglichkeiten, das Aus- und Abwählen von Balken oder Linien in den entsprechenden Diagrammen. Plotly Express kann pandas Dataframes oder pandas Series als Datenobjekte erwarten und verarbeiten. Dabei können die

---

<sup>9</sup> Für das vorliegende Projekt wurden insbesondere Teile der Datenanalyse damit entwickelt und getestet.

Spaltenköpfe die Achsen darstellen und die einzelnen Werte der Spalten die Datenpunkte bilden. Die Rendering-Engine für die Diagramme beruht auf dem JavaScript Framework D3.js. Plotly Express ist frei verfügbar. In Kombination mit der Bibliothek Dash für die Entwicklung von interaktiven Dashboards lässt sich Plotly Express gut anwenden, da beide von der selben Firma entwickelt werden und aufeinander abgestimmt sind.

Mit der Bibliothek Dash wurde das Dashboard realisiert. Dash baut auf den Technologien Flask, plotly.js und React.js auf. Das Framework ermöglicht die Erstellung interaktiver Webapplikationen oder 'Analytical Applications' in Python ohne hierfür mit Javascript programmieren zu müssen [vgl. [Plo21c](#)]. Es gibt verschiedene Dash-Komponenten, die unterschiedliche Funktionen erfüllen. `Dash_html_components` stellen Klassen für alle HTML-Tags bereit. Die Schlüsselwortargumente dieser Klasse beschreiben HTML-Attribute wie `style`, `className` und `id` [vgl. [Plo21d](#)]. Andere Komponenten sind `dash_core_components` oder `dash_dependencies` [vgl. [Plo21b](#)]. Während die `dash_core_components` Steuerelemente und Graphen erzeugen, regeln die `dash_dependencies` über Callback-Dekorator-Funktionen die Interaktion zwischen den einzelnen Komponenten. So kann zum Beispiel in der Dash-WebApplikation das Verhalten eines Diagramms von den Werten eines Dropdown-Menü gesteuert werden. Dabei bewältigt Dash alle Javascript-Anforderungen im Front- und im Backend [vgl. [Plo21a](#)]. `Dash_bootstrap_components` ist eine weitere Bibliothek, die zum Einsatz in diesem Projekt kommt. Sie unterstützt die grafische Umsetzung des Dashboardes [vgl. [Fac21](#)].

[Tabelle 5.1](#) zeigt einen kurzen Überblick über die Versionsnummern der genutzten Programmiersprache und der hauptsächlich genutzten Bibliotheken sowie deren Open-Source Lizenzen.

Name	Version	Lizenz	Webseite
Python	3.7.9	Open Source (PSF)	<a href="https://docs.python.org/3.7/">https://docs.python.org/3.7/</a>
pandas	1.1.5	3-Clause-BSD-License	<a href="https://pandas.pydata.org/pandas-docs/version/1.1.5/">https://pandas.pydata.org/pandas-docs/version/1.1.5/</a>
Plotly	4.14.1	MIT-License	<a href="https://plotly.com/python/">https://plotly.com/python/</a>
Dash	1.18.1	MIT-License	<a href="https://dash.plotly.com/">https://dash.plotly.com/</a>

Tabelle 5.1: Liste der grundlegenden Programmiersprache und Frameworks

### 5.1.2 SYSTEMARCHITEKTUR

Das System teilt sich in drei Teilsysteme auf, die im [Unterabschnitt 5.1.3](#) näher beschrieben werden. In der folgenden [Abbildung 5.2](#) wird die Systemarchitektur des Projektes gezeigt.

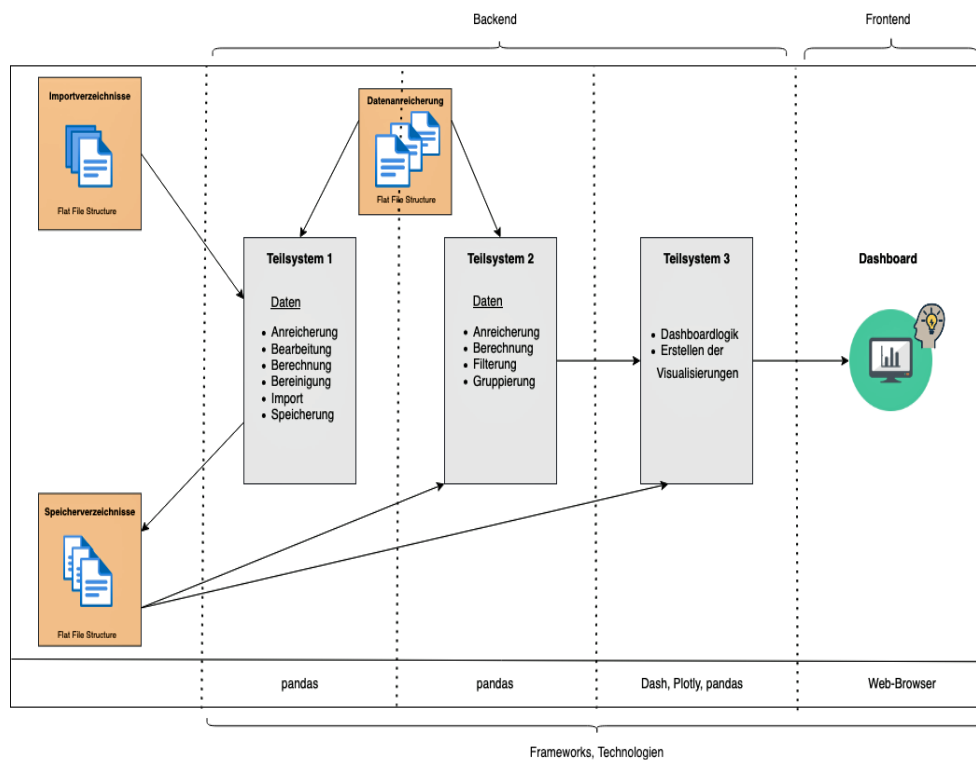


Abbildung 5.2: Systemarchitektur

## 5 Diskussion der Umsetzung

Teilsystem 1 ist für die erste Bereinigung der Daten und den Import der Daten in ein einheitliches Datenformat verantwortlich. Teilsystem 2 greift auf den Speicherort der importierten Dateien zu und bereitet die Daten für die Anzeige im Dashboard vor, indem es mit der Python-Bibliothek pandas Daten vorfiltert, gruppiert und Berechnungen auf den Daten ausführt. Teilsystem 3 ist für das Layout des Dashboards, die Erstellung der Diagramme, die Anordnung der Diagramme im Dashboard und die Bereitstellung von Interaktionen mit dem Dashboard im Frontend verantwortlich. Teilsystem 1 und Teilsystem 2 wurde objektorientiert programmiert, da hier der Anspruch bestand, den Programmcode für verschiedene Daten wiederzuverwenden. Teilsystem 3 besteht vorerst nur aus Funktionen, die die Anzeige im Frontend ermöglichen. [Tabelle 5.2](#) zeigt die drei Teilsysteme mit einer Kurzbeschreibung der Hauptaufgabe.

Teilsystem Name		Hauptaufgabe
1	Import	Import und erste Bereinigung der Daten aus heterogenen Datenquellen.
2	Datenbearbeitung	Aufbereitung der Daten für die graphische Darstellung im Dashboard.
3	Darstellung	Umwandlung der Daten in Datenvisualisierungen und Implementierung der Dashboard-Logik.

Tabelle 5.2: Liste der Teilsysteme

## 5.1.3 TEILSYSTEME

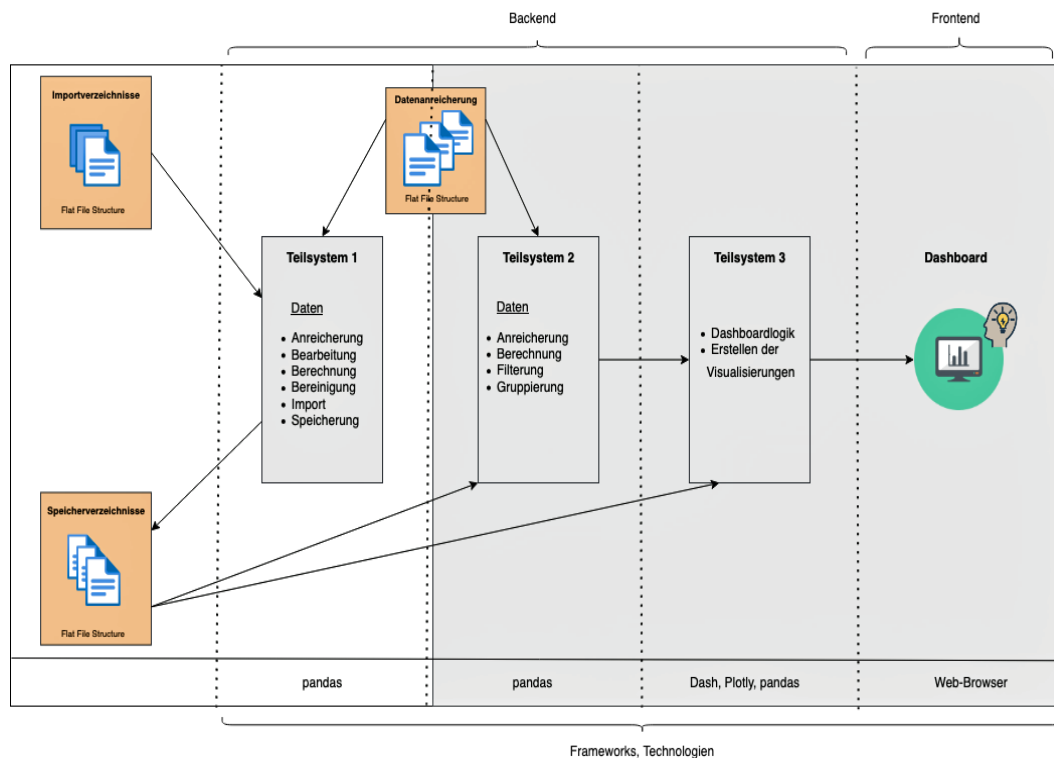
*Teilsystem 1 Import*

Abbildung 5.3: Systemarchitektur Teilsystem 1

Das *Teilsystem 1 Import* ist verantwortlich für den Import der Daten im Rohformat aus vordefinierten Importverzeichnissen in vordefinierte Zielverzeichnisse. Das Ziel ist einerseits die Daten automatisch ohne Informationsverlust zu importieren und andererseits diese mit notwendigen Daten für die weitere Analyse anzureichern und erste Bereinigungen der Daten durchzuführen. Die Daten werden zum Abschluss im csv-Format abgespeichert. Mit dem Teilsystem soll eine einheitliche Datengrundlage für die spätere Bearbeitung und Darstellung der Daten garantiert werden. Für das csv-Format wurde sich aufgrund seiner flachen und einfachen Struktur, der guten Lesbarkeit und seiner weiten Verbreitung entschieden. Das *Teilsystem 1 Import* besteht aus vier Klassen. Die

## 5 Diskussion der Umsetzung

Abbildung 5.4 zeigt die einzelnen Klassen mit ihren Methoden. Die Klassen im *Teilsystem 1 Import* sind auf die Daten aus fremden Quellen zugeschnitten (Budget Umsatz, Neuerwerbungslisten, Ausleihe (Anwendungsfall 2, 4-6)). Dennoch können mit mit ihnen auch die bibliotheksinternen Daten wie Lesesaalnutzung (Anwendungsfall 3) bearbeitet werden.

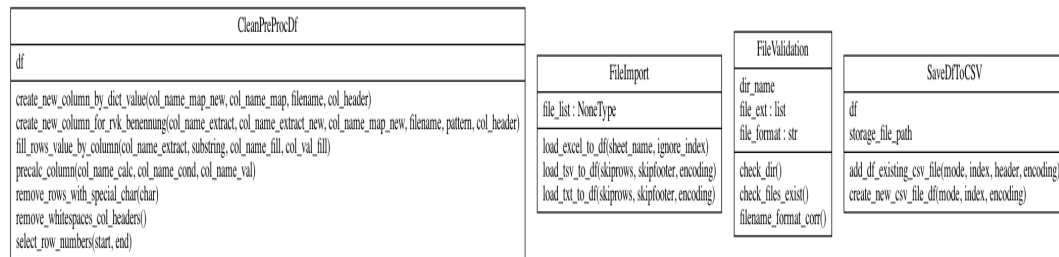


Abbildung 5.4: Klassendiagramm - Teilsystem 1 Import

Instantiiert werden die einzelnen Klassen für die jeweiligen konkreten bibliothekarischen Daten in eigenen Python-Skripten. So gibt es Skripte für Budget, Umsatz, Ausleihe und Bestand/Neuerwerbungen. Diese verwenden für die Daten passende Methoden und führen den Import durch. Den Ablauf des Teilsystems zeigt schematisch die [Abbildung 5.5](#).

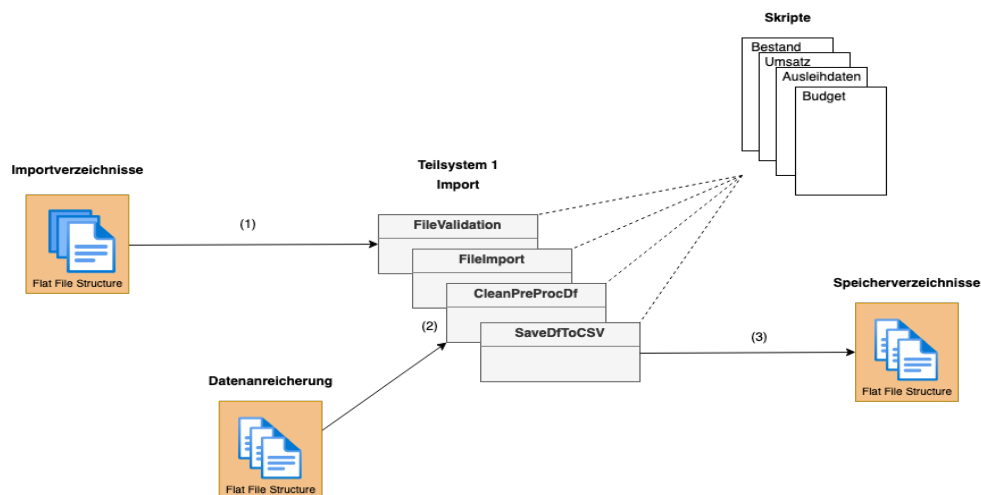


Abbildung 5.5: Datenfluss - Teilsystem 1 Import



(1) Für den ersten Schritt sind die Klassen `FileValidation` und `FileImport` verantwortlich. Die Dateien werden aus einem lokalen Verzeichnis in ein pandas Dataframe geladen. Dabei wird mit den Methoden der Klasse `FileValidation` überprüft, ob sowohl das Verzeichnis als auch die Dateien existieren. Des Weiteren wird sichergestellt, dass die Dateinamen einem definierten semantischen Format wie dem Datumsformat `YYYY_MM_DD` und einem Dateiformat wie `txt`-, `xlsx`- oder `tsv`-Format entsprechen. Wenn die Daten nicht diesen Vorgaben entsprechen, werden sie nicht in das pandas Dataframe geladen. Da die Daten unterschiedlich aufgebaut und in unterschiedlichen Dateiformaten vorliegen, werden beim Laden in den Dataframe jeweils verschiedene Methoden angewandt. Dabei werden spezifische pandas-Funktionen eingesetzt, die diese Datenformate parsen können. Das Parsen der unterschiedlichen Dateiformate leitet die Datentypen der Werte in den Spalten aus den vorliegenden Werten der Spalten automatisch ab.<sup>10</sup>

Beim Ladeprozess der Dateien in den pandas Dataframe wird mit den Methoden `load_txt_to_df()` oder `load_tsv_to_df()` der Klasse `FileImport` der Dateiname von der Datei extrahiert und in einer neu geschaffenen Spalte des Dataframes im Datumsformat `YYYY-DD-MM` gespeichert. Anhand der Werte dieser Spalte werden später verschiedene Datenanalysen und Datenvisualisierungen vollzogen.<sup>11</sup> Verantwortlich für die Extrahierung des Dateinamens ist die in `utils.py` ausgelagerte Funktion `date_from_filename()`, die den Dateinamen als Argument entgegennimmt.<sup>12</sup>

(2) Das geladene pandas Dataframe wird im zweiten Schritt durch die Klasse `CleanPreProcDf` aufgenommen und durch verschiedene Methoden dieser Klasse manipuliert. Beispielhaft ist hier die Methode `create_new_column_for_rvk_benennung()` zu nennen, die aus einer Spalte einen Substring unter Zuhilfenahme eines regulären Ausdrucks extra-

---

<sup>10</sup>Das funktionierte für das vorliegende Projekt sehr gut, sodass bei diesem Prozess nicht manuell eingegriffen wurde.

<sup>11</sup>Dieses Verfahren wird bei den Daten ausgeführt, die einer zusätzlichen Datumsspalte bedürfen.

<sup>12</sup>In der Datei `utils.py` sind noch andere Funktionen als stand-alone-functions für den Datenimport und Datenbearbeitung gruppiert, da diese das Objekt `self` nicht verändern. Zu finden ist die Datei im Projekt in dem Verzeichnis `src`.

hiert, diesen in eine neue Spalte schreibt und ihn um zusätzliche Informationen anreichert. Diese Methode wird sowohl bei den Neuerwerbungs/Bestandsdaten als auch bei den Ausleihdaten zweimal angewendet, um die *RVK*-Fachsystematiken der Titelsignaturen zu extrahieren und durch die *RVK*-Benennungen anzureichern. Zunächst werden die Untergruppen der Fachsystematiken aus der Signatur extrahiert, in einer neuen Spalte gespeichert und um die entsprechenden Benennungen der Untergruppen der *RVK* angereichert. Danach werden die Fachsystematiken aus den Untergruppen extrahiert und ebenfalls um die entsprechenden Benennungen angereichert. Mit der [Abbildung 5.6](#) lässt sich dieser Prozess an der Signatur 'CP 2000 zit 2018' nachvollziehen.

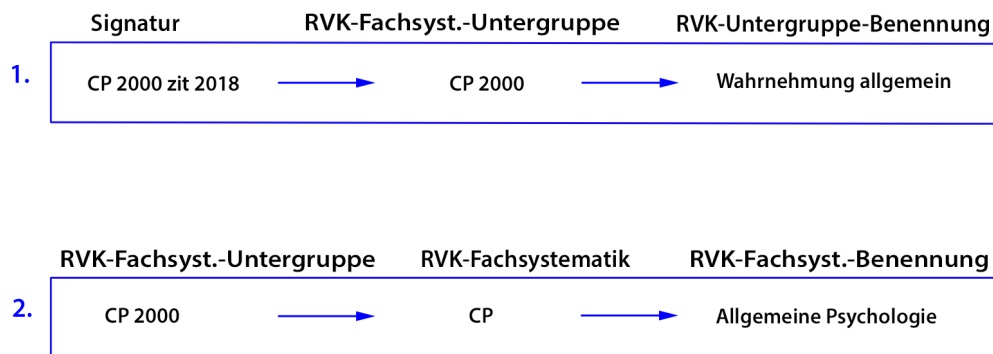


Abbildung 5.6: Beispiel Extraktion Fachsystematik

Zu diesem Zweck liegt eine csv-Datei der *RVK* zu Grunde, die mit einem Python-Skript aus einer *RVK*-XML-Datei[Vgl. [RVK21](#)] entstanden ist. In dieser liegen die Fachsystematiken und die Untergruppen mit ihren Benennungen vor.<sup>13</sup> Diese wird bei jedem Aufruf der Methode mit den Werten der neu entstanden Spalte gemappt. Die Benennungen werden ebenfalls in je einer Spalte gespeichert. Ziel ist es, die Neuerwerbungs- und Bestandsdaten später nach den *RVK*-Fachsystematiken und deren Untergruppen auszuwerten.

<sup>13</sup>Manuell ergänzt wurde diese noch um selbstgeschaffene Fachsystematiken der Institutsbibliothek.

Bei Umsatz- und Budgetdaten entstehen ebenfalls neue Spalten. In Bezug auf die Darstellung der Daten im Dashboard werden hier die Namen der Lieferanten und Kostenstellen in lesbarer Form in einer neuen Spalte gespeichert. Des Weiteren gibt es in der Klasse `CleanPreProcDf` Methoden, die für die Entfernung von Reihen mit bestimmten syntaktischen Zeichen wie Bindestrichen verantwortlich sind oder die eine Vielzahl unnötiger Leerzeichen in Spaltenköpfen löschen und durch ein Leerzeichen ersetzt. Ferner werden erste Berechnungen auf den Daten ausgeführt, um valide Daten für die Auswertung zu haben. Dies betrifft die Ausleihdaten und wird durch die Methode `precalc_column()` realisiert.<sup>14</sup>

(3) Nach dem Transformationsprozess wird das veränderte pandas Dataframe als csv-Datei in einem vorher definierten Speicherordner gespeichert. Verantwortlich ist dabei die Klasse `SaveDfToCsv` mit den zwei Methoden `add_df_existing_csv_file()` und `create_new_csv_file_df()`. Es wird durch diese Methoden entweder eine neue csv-Datei mit dem Dataframe erstellt oder der Dataframe wird an eine bereits vorhandene csv-Datei angehängt. [Abbildung 5.7](#) zeigt anhand einer Umsatz-Datei, die im txt-Format gespeichert wurde, die Transformation in ein csv-Format. Zur besseren Lesbarkeit der csv-Datei wird die Darstellung in einem Tabellenformat angezeigt.

---

<sup>14</sup>Im Arbeitsprozess der Medienschließung validiert die Bibliothek die RFID-Etiketten der Medien durch einmalige Ausleihe der Medien am Selbstverbucher. Deswegen wird die Anzahl der Ausleihe pro Medium um eins reduziert, wenn sie ein oder mehrmals ausgeliehen wurden.

## 5 Diskussion der Umsetzung

Stand: Nov 1 2020 6:16AM

Lieferant	Umsatz (EUR)
Antiquariat	0.00
Schildkröte Buchhandlung	0.00
Rosa Arbeiter	0.00
HDGDL Buch	0.00
Eschenbach Verlag	0.00
Sappho bookshop	0.00
Unser Glück Verlag	0.00
Schwalben Medien	0.00
Golden Leaves Musikverlag	0.00
Tassen	0.00
schwarze Botin	0.00
Buchhandlung Wal	0.00
FuP Informationen	0.00
Leslie Ben Ron	0.00
LiC Vinyl	0.00
	Umsatz (EUR)
Summe	0.00

	Lieferant	Umsatz (EUR)	Datum	Lieferant Abk.
0	Antiquariat	0.00	2020-10-01	Antiquariat
1	Schildkröte Buchhandlung	0.00	2020-10-01	Schildkröte Buchhandlung
2	Rosa Arbeiter	0.00	2020-10-01	Rosa Arbeiter
3	HDGDL Buch	0.00	2020-10-01	HDGDL Buch
4	Eschenbach Verlag	0.00	2020-10-01	Eschenbach Verlag
5	Sappho bookshop	0.00	2020-10-01	Sappho bookshop
6	Unser Glück Verlag	0.00	2020-10-01	Unser Glück Verlag
7	Schwalben Medien	0.00	2020-10-01	Schwalben Medien
8	Golden Leaves Musikverlag	0.00	2020-10-01	Golden Leaves Musikverlag
9	Tassen	0.00	2020-10-01	Tassen
10	schwarze Botin	0.00	2020-10-01	schwarze Botin
11	Buchhandlung Wal	0.00	2020-10-01	Buchhandlung Wal
12	FuP Informationen	0.00	2020-10-01	FuP Informationen
13	Leslie Ben Ron	0.00	2020-10-01	Leslie Ben Ron
14	LiC Vinyl	0.00	2020-10-01	LiC Vinyl

Abbildung 5.7: Monatliche Umsatzübersicht nach Ablauf Teilsystem 1 Import

Die Daten liegen nun in den Speicherzeichnissen vor und können nun durch das *Teilsystem 2 Datenbearbeitung* weiterbearbeitet werden.

## Teilsystem 2 Datenbearbeitung

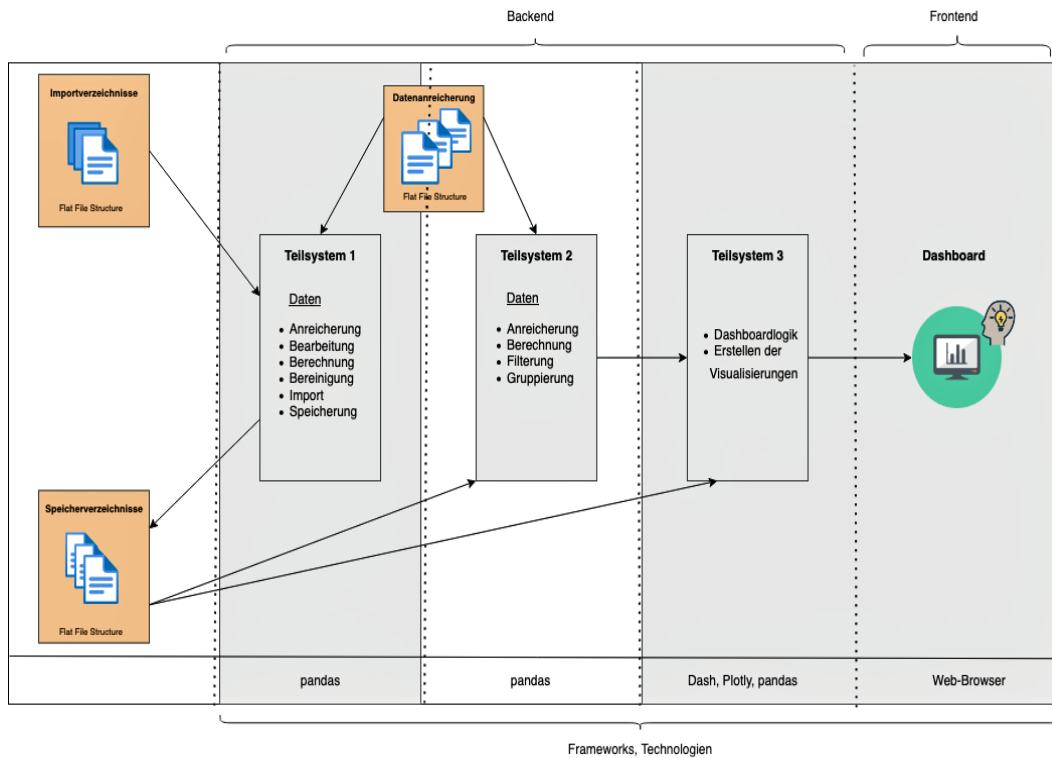


Abbildung 5.8: Systemarchitektur Teilsystem 2

Das *Teilsystem 2 Datenbearbeitung* hat das Ziel, die Daten für die Darstellung im Dashboard vorzubereiten. Der weitere Zweck dieses Teilsystem ist, die Vorbereitung der Daten von der Erstellung der Datenvisualisierungen zu trennen. Ein Grund hierfür ist, den Programmcode im Teilsystem 3 nicht mit dem Programmcode der Datenmanipulation zu überfrachten. Deswegen werden in dem Teilsystem Datenbearbeitung Daten mit pandas so bearbeitet, dass entweder Teilmengen der eigentlichen Daten oder Ergebnisse mathematischer Operationen durch das Teilsystem 3 zu Datenvisualisierungen nur noch weiterverarbeitet werden müssen. Die Vorbereitung der Daten umfasst einzelne Berechnungen wie `mean()` oder `sum()`. Ferner werden im Teilsystem 2 die Sortierung, Gruppierung und Filterung der Daten nach bestimmten Aspekten vollzogen, die in den Anwendungsfällen im [Unterabschnitt 4.1.5](#) formuliert sind.

## 5 Diskussion der Umsetzung

Für das *Teilsystem 2 Datenbearbeitung* ist das Modul `data_prep` zentral. Die Klassenstruktur dieses Moduls ist eine Basisklasse, von der vier Kindklassen erben. Es gibt Kindklassen für Umsatz- und Budgetdaten, für Bestands- und Neuerwerbungsdaten sowie für die Daten der Ausleihe und der Lesesaalnutzung. Aufgrund der ähnlichen Datenstruktur der Umsatz- und Budgetdaten genügt eine Klasse für beide. Der Gesamtbestand und die monatlichen Neuerwerbungen berufen sich auf denselben Datenbestand, sodass hier ebenfalls auf eine Aufteilung auf mehrere Klassen verzichtet wurde. Für jede Kindklasse wurden Methoden geschrieben, die auf den spezifischen Bibliotheksdaten verschiedene Manipulationen ausführen. Die Ergebnisse werden von jeder Methode zurückgeben.

Abbildung 5.9 zeigt die Basisklasse und die vier Kindklassen mit ihren Methoden.

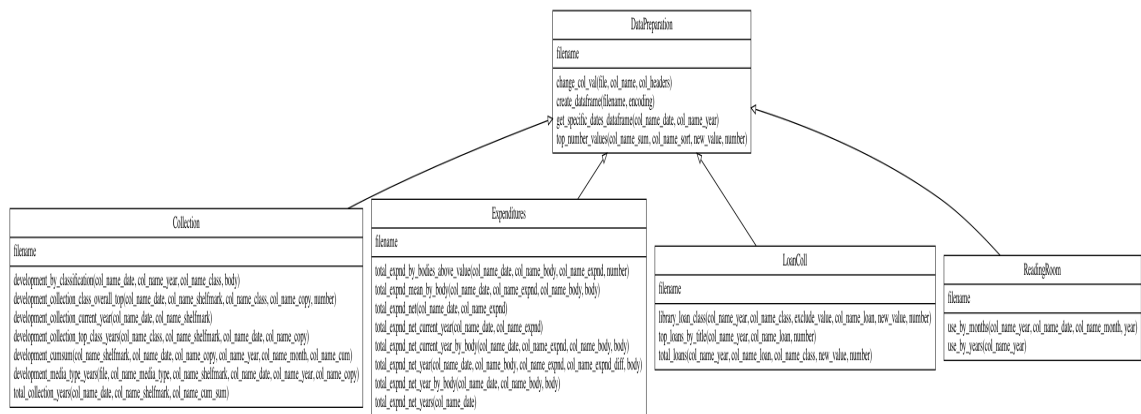


Abbildung 5.9: Klassendiagramm - Teilsystem 2 Datenbearbeitung

Der Ablauf innerhalb des Teilsystems besteht aus zwei Schritten.

(1) Das Laden der einzelnen csv-Dateien in dem pandas Dataframe aus dem Zielverzeichnis wird durch die Methode `create_dataframe()` der Basisklasse geregelt. Beim Aufrufen des Konstruktors beziehungsweise der `init`-Methode der Klasse wird die Methode `create_dataframe()` automatisch mit aufgerufen. Da die Methode und die Konstruktor-Properties der Basisklasse an die Kindklassen vererbt werden, kann in den Kindklassen das Objekt ebenfalls mit dem pandas Dataframe initiiert werden. Die Objekte werden im nächsten Schritt durch die spezifischen Methoden der Kindklassen bearbeitet.

(2) Das Ergebnis der Manipulation der pandas Dataframes ist auf die Darstellung der Daten im Dashboard ausgerichtet. Als Input der einzelnen Methoden der Kindklassen werden verschiedene Parameter verlangt, mit deren Hilfe der Dataframe entweder manipuliert wird oder auf ihm Berechnungen ausgeführt werden können. Um die Daten für das Teilsystem 3 vorzubereiten, können verschiedene Transformationsschritte auf den Daten in den Methoden ausgeführt werden. Die Rückgabewerte der Methoden sind veränderte pandas Dataframes, pandas Series oder einzelne Ergebnisse mathematischer Operationen.

Anhand der folgenden Methode `total_expnd_net_current_year()` soll der letzte Schritt verdeutlicht werden. Die Methode der `Expenditures`-Klasse bestimmt im Allgemeinen die Summe von Werten einer Spalte, gefiltert nach einem Wert einer anderen Spalte eines pandas Dataframes. Mit dieser Methode soll so zum Beispiel konkret der Gesamtumsatz des laufenden Jahres bestimmt werden.

---

```
def total_expnd_net_current_year(self, col_name_date, col_name_expnd):
    ...
    self._date_max = self._df[col_name_date].max()
    self._df = self._df.set_index(col_name_date)
    self._df = self._df.loc[self._date_max]

    return self._df[col_name_expnd].sum()
```

---

Quellcode 5.1: Beispiel Methode `Exenditures` class

Als Input erwartet die Methode in der Methodensignatur zwei Spaltennamen als Parameter: den Namen der Datumsspalte `col_name_date`, nach der gefiltert wird, und den Namen der Umsatz-Spalte `col_name_expnd`, auf der die Berechnung stattfindet. Der Transformationsprozess im Methodenkörper teilt sich in drei Schritte auf. Da die monatlichen Umsatzdaten im Jahr als akkumulierte Daten vorliegen, interessieren nur die letzten vorhandenen Datensätze des laufenden Jahres. Deswegen wird nach diesen gefiltert und ein Dataframe von diesen erstellt. Dementsprechend wird mit der pandas-Funktion `max()`

## 5 Diskussion der Umsetzung

zunächst der maximale Wert in der Datumsspalte bestimmt und der Variable `_date_max` zugewiesen. Danach wird die Datumsspalte als Index gesetzt. Im dritten Schritt wird der Dataframe mit den Reihen, die der Variable `_date_max` entsprechen, erstellt. Dies geschieht mit der pandas-Funktion `.loc`, die auf den Index der Reihen des Dataframes zugreift. Zum Schluss wird auf Basis der Umsatz-Spalte des Dataframes die Summe mit der pandas-Funktion `sum()` berechnet und als Rückgabewert zurückgeliefert. Der Rückgabewert kann nun vom *Teilsystem 3 Darstellung* weiterverarbeitet werden.

### Teilsystem 3 Darstellung

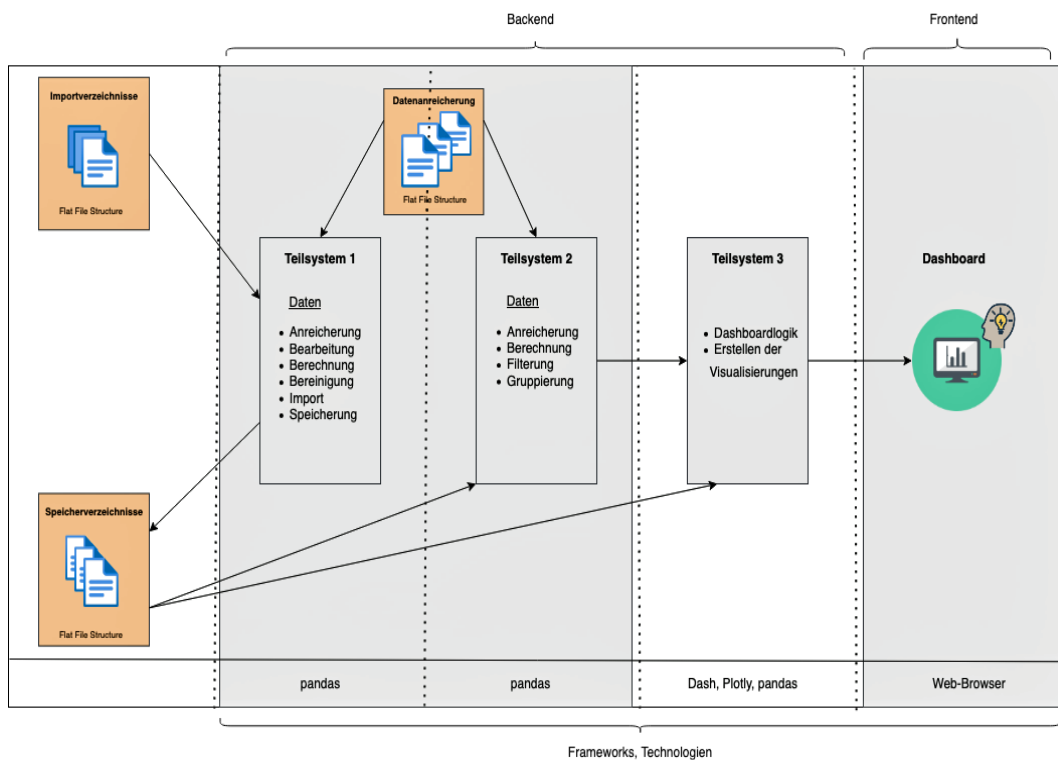


Abbildung 5.10: Systemarchitektur Teilsystem 3

Für die Erschaffung des Dashboards mit seinen Datenvisualisierungen und Interaktionen ist das *Teilsystem 3 Darstellung* verantwortlich. In ihm werden die Ergebnisse des



Teilsystems 2 zu Datenvisualisierungen verarbeitet und die Dashboard-Logik bereitstellt. Dazu werden die Bibliothek Plotly Express für die Datenvisualisierungen und die Bibliothek Dash zur Umsetzung des Dashboards genutzt.

Die Struktur der Dashboard-App entspricht einem Multi-App-Dashboard. Aufgrund der Vielzahl an Datenvisualisierungen wurde sich gegen eine Single-Page-Lösung entschieden. Deswegen besteht das Dashboard aus drei einzelnen Tabs, auf die die einzelnen Datenvisualisierungen aufgeteilt sind.

Es gibt verschiedene Möglichkeiten Multi-App-Dashboard-Projekte zu strukturieren. Für das vorliegende Projekt wurde sich für die in [Abbildung 5.11](#) gezeigte Struktur entschieden.

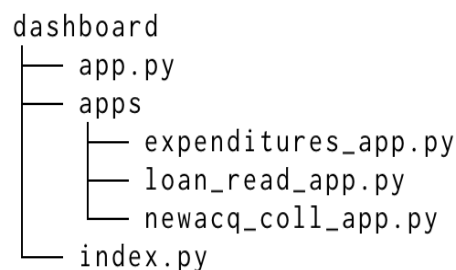


Abbildung 5.11: Struktur Dash Multi-Page App

Dies ist eine Möglichkeit, wie sie auf der Webseite von Dash für Multi-App-Projekte vorgeschlagen wird [vgl. [Plo21h](#)].

Die einzelnen Tabs wurden inhaltlich um bibliothekarische Tätigkeiten zu gruppieren. So werden in der `expenditures_app.py` Datenvisualisierungen erstellt, die Budget- und die Umsatzdaten visualisieren, während mit Hilfe der `loan_read_app.py` Ausleih- und Lesesaalnutzungsdaten dargestellt werden. Mit der `newacq_coll_app.py` wird ermöglicht, Daten aus dem Bereich der Bestandsentwicklung und der Ausleihe zu präsentieren.

Für jeden Tab gibt es eine separate Datei. Jede einzelne Tab-Datei besteht einerseits aus mehreren Funktionen für die Erschaffung von Datenvisualisierungen der Ergebnisse aus Teilsystem 2.<sup>15</sup> Innerhalb dieser Funktionen werden mit den Plotly-Funktionen `Plot-`

<sup>15</sup>Zur besseren Lesbarkeit heißt die Gruppe dieser Funktionen im Folgenden `fig_data()`.

ly Graph Object Figures geschaffen. Andererseits bestehen die Dateien aus mehreren verschiedenen Funktionen für die Dash-Komponenten wie zum Beispiel Dropdown-Menüs, Cards oder Diagrammen.<sup>16</sup> Diese Funktionen binden die `fig_data()`-Funktionen so ein, dass die Plotly Graph Object Figures im Dashboard zur Anzeige gebracht werden können. Weiterhin sind die `html_fig()`-Funktionen zum Teil mit Dekorator-Callback-Funktionen verknüpft, die es ermöglichen, mit dem Dashboard zu interagieren. Ferner enthalten die Dateien jeweils eine Layoutfunktion, die das gesamte Layout des Tabs bündelt.

Neben den Funktionen für die Dash-Komponenten und der Datenvisualisierungen, werden in den tab-Dateien noch die benötigten Objekte aus dem Modul `data_prep` instanziiert und die Methoden der Kindklassen aus diesem Modul auf diesen angewendet. In der Regel geschieht dies außerhalb der Funktionen. Für die Callback-Funktionalität werden aber die Objekte und die Methoden des Moduls `data_prep` innerhalb einiger `html_fig()` aufgerufen. Anhand der Erstellung von Plotly Graph Object Figures wird exemplarisch auf den Aufbau und die Funktionsweise der Funktionen in den Tab-Dateien im Folgenden näher eingegangen.<sup>17</sup>

Schematisch zeigt die [Abbildung 5.12](#) die Ablauflogik in den Tab-Dateien ohne die callback-Funktionen. Die Objekte werden zunächst in Plotly Graph Objects umgewandelt. Diese wiederum werden in Dash-Objekte transformiert und diese werden letztlich in einem Tab-Layout zusammengefasst.

---

<sup>16</sup>Ebenfalls zur besseren Lesbarkeit heißt die Gruppe dieser Funktionen im Folgenden `html_fig()`.

<sup>17</sup>Auf die Erzeugung von anderen Dash-Elementen wie Cards wird dabei aufgrund der Übersichtlichkeit der Darstellung nicht Bezug genommen. Diesem liegt ein ähnlicher Prozess zu Grunde. Es werden aus den berechneten Ergebnissen der Objekte direkt Dash-Komponenten mit Hilfe der `dash_bootstrap_components` erstellt.

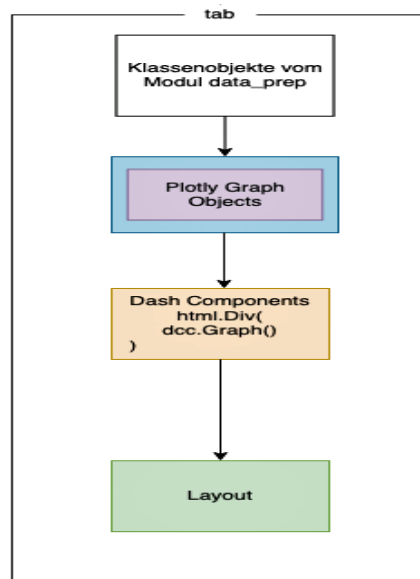


Abbildung 5.12: Ablauf Tab

Die grundsätzliche Logik der Funktionsaufrufe sieht in den Tab-Dateien ohne die Callback-Funktionen wie folgt aus:

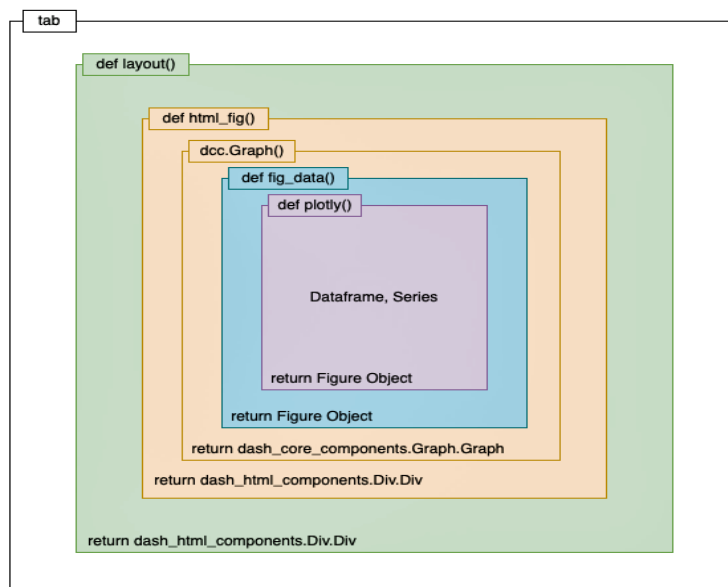


Abbildung 5.13: Funktionsaufrufe Tab

In der Plotly-Funktion, die das `Plotly Graph Object Figure` erzeugt, werden die durch die Methoden der Kindklassen bearbeiteten Objekte (Dataframes, Series) als Parameter entgegengenommen. Zusätzlich müssen noch weitere Parameter übergeben werden, die für die jeweiligen Diagrammtypen wichtig sind. Andere Parameter, die das Layout festlegen, sind dagegen optional [vgl. [Plo21g](#)].

Im folgenden Beispiel wird in der Funktion `fig_total_expnd()` aus `expenditures_app.py` ein horizontal gekipptes Figure-Objekt mit dem pandas Dataframe `df_total_expnd` durch den Aufruf der Plotly-Funktion `bar()` erstellt. Es werden zusätzlich in dem Funktionsaufruf noch die x-Achse und y-Achse mit den Werten der Spalten 'Umsatz (EUR)' und 'Datum' festgelegt.

---

```
def fig_total_expnd():  
    ...  
    fig = px.bar(df_total_expnd,  
                 x='Umsatz (EUR)',  
                 y='Datum',  
                 orientation='h',  
                 ...  
    )  
    ...
```

---

Quellcode 5.2: Funktion `fig_total_expnd()` Auszug 1

Weitere Funktionen von Plotly Express bearbeiten die `Plotly Graph Object Figures` in den `fig_data()`-Funktionen. So kann durch die `update_layout`-Funktion unter anderem der Achsentitel, die Höhe und die Breite der Datenvisualisierung festgelegt werden. Als Rückgabewert der Funktionen `fig_data()` werden die `Plotly Graph Object Figures` zurückgegeben.

---

```
def fig_total_expnd():  
    ...  
    fig.update_layout(title_x=0.5,
```

```

    xaxis_title='Umsatz in Euro',
    yaxis_title='Jahr',
    height=500)
fig.update_xaxes(nticks=20)
return fig

```

---

#### Quellcode 5.3: fig\_total\_expnd() Auszug 2

In dem Beispiel werden die Titel der x- und y-Achse, sowie die Größe des Diagramm-Objektes festgelegt. Schließlich wird noch die x-Achse mit der Funktion `update_xaxes()` skaliert.

Die `fig_data()`-Funktionen werden durch die Graph-Komponente der `dash_core_components` innerhalb der `html_fig()`-Funktionen aufgerufen. Diese Komponente ist für die Umsetzung der interaktiven Datenvisualisierungen zuständig. Ebenfalls kann von der Graph-Komponente eine `id` als Parameter entgegengenommen werden. Diese ist unter anderem wichtig für die eindeutige Adressierung durch die Callback-Funktionen. Zudem werden in den `html_fig()`-Funktionen durch die `dash_html_components` die Eigenschaften und das Aussehen der Div-Objekte definiert. Dabei werden die Properties unter anderem in der externen css-Datei `layout.css` definiert, die in dem Verzeichnis `assets` des Dashboard-Verzeichnisses liegt.

---

```

def html_fig_total_expnd():
    ...
    return html.Div(
        [
            html.Div(
                [
                    dcc.Graph(
                        id='gesamtumsatz',
                        figure=fig_total_expnd()
                    ),

```

## 5 Diskussion der Umsetzung

```
    ], className="six columns chart_div", style={'margin-top': '20px', 'margin-  
-left': '10px'}  
    ),  
    ]  
    )
```

---

Quellcode 5.4: `html_fig_total_expnd()`

Die `html_fig()`-Funktionen werden schließlich in einer Layoutfunktion eingebunden, die alle Funktionen dieser Art für das Gesamtlayout des Tab bündelt. Als Rückgabewert returniert sie ebenso wie die `html_fig()`-Funktionen ein Div-Objekt der `dash_html_components`.

Die Callback-Funktionen wurden für zwei Dropdown-Menüs in zwei Tab-Dateien implementiert. Die Werte der Dropdown-Menüs werden aus den unique Werten einer Dataframespalte erstellt. Der Callback ist als Dekorator für jeweils eine Funktion implementiert. In der Implementierung wird ein Callback ausgelöst, wenn ein Wert über das Dropdown-Menü ausgewählt wird. Im Programmcode wird das folgendermaßen für ein Diagramm der Lesesaalnutzung in der `loan_read_app.py` umgesetzt.

---

```
@app.callback(  
    Output(component_id='use_by_month', component_property='figure'),  
    [Input(component_id='my-id2', component_property='value')])  
def update_output_div(input_value):  
    ...  
    return fig_use_by_month(input_value)
```

---

Quellcode 5.5: `html_fig_total_expnd()`

Der Callback übergibt den Wert (`input_value`) des Dropdown-Menüs an die Funktion `update_output_div()`. Die Funktion gibt das Ergebnis einer `fig_data()`-Funktion mit

diesem Wert als Argument zurück.<sup>18</sup> Der Callback `@app.callback` übergibt das zurückgegebene Ergebnis an die im Output angegebene Komponente.

`Input()` und `Output()` nehmen die `id` einer Komponente und die Eigenschaft einer Komponente als Argumente entgegen. Die Inputkomponente ist in dem Beispiel die Dropdown-Komponente, während die Outputkomponente die `dcc.Graph`-Komponente der `html_fig_use_by_month()` darstellt. Beide werden über die eindeutige `component_id` adressiert. Multiple Inputs and Outputs sind ebenfalls möglich. So sind in der `expenditures_app.py` jeweils zwei Diagramme und Zahlenwerte für den Umsatz von den Werten einer Dropdown-Liste abhängig. In der folgenden Abbildung ist die Ablauflogik in den Tab-Dateien mit der Callback-Funktion zu sehen.

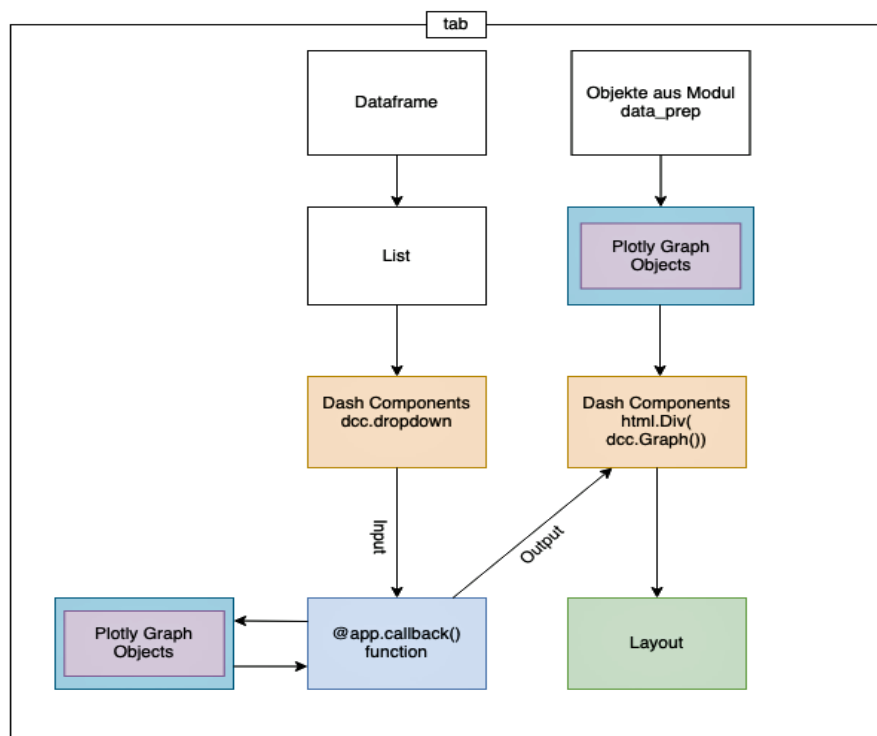


Abbildung 5.14: Ablauf Tab mit Callback

<sup>18</sup>Diese Funktionen übergeben den Methoden der Kindklassen aus dem Modul `data_prep` das Argument. Diese erzeugen einen Dataframe basierend auf den übergebenden Argument und geben mit Hilfe der Plotly-Funktionen ein `Plotly Graph Object Figure` der gefilterten Daten zurück.

## 5 Diskussion der Umsetzung

Zusammengesetzt wird das Layout des Dashboards durch den Aufruf der Layout-Funktion der einzelnen Tab-Dateien in der `index.py`. Die `index.py` definiert zudem das Layout des gesamten Dashboards. Hier werden auch die Anzahl und die Eigenschaften der Tabs festgelegt. Die Dekorator-Callback-Funktion `@app.callback()` steuert die Auswahl der Tabs und ruft die jeweiligen Tabs beziehungsweise deren Layouts auf. [Abbildung 5.15](#) zeigt im Allgemeinen die Funktionsweise einer Tab-Datei mit der `index.py`.

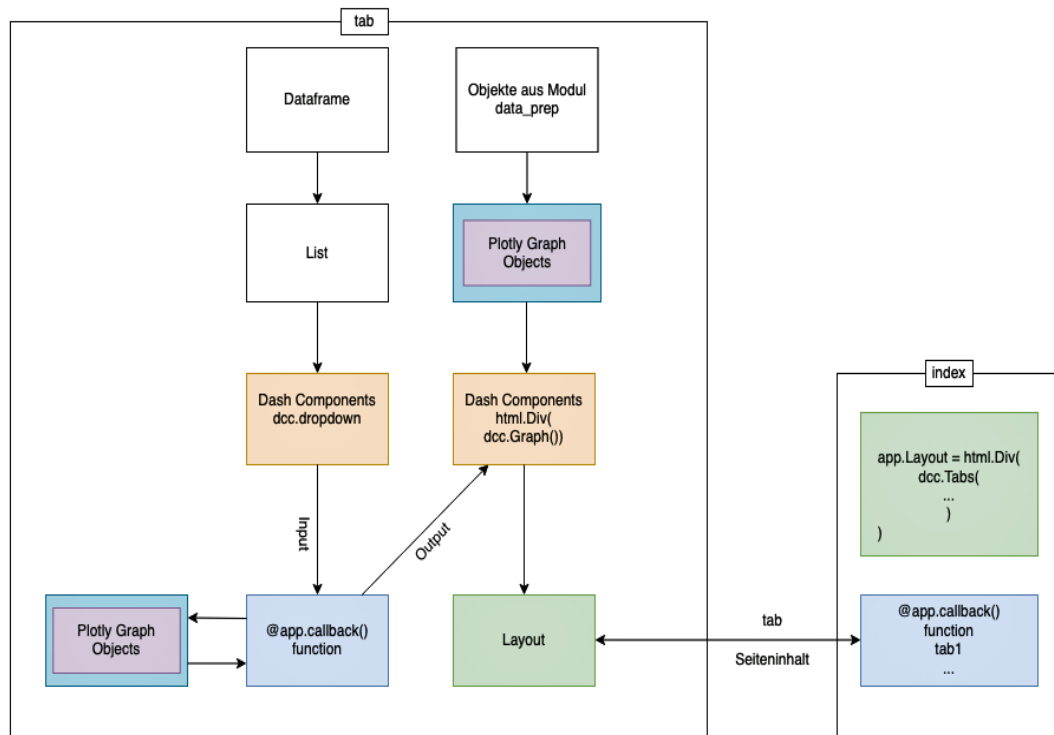


Abbildung 5.15: Ablauf Tab mit index

Der Einstiegspunkt zur Ausführung des Dashboards ist die Datei `index.py`. Mit dieser Datei wird das Dashboard mittels `Flask-Webserver` gestartet. Zur Vermeidung zirkulärer Importe ist die Dash-Instanz in der separaten `app.py` definiert[vgl. [Plo21h](#)].



## 5.2 DEMONSTRATION DER FUNKTIONALITÄT

Im Folgenden wird die Funktionsweise des Systems dargelegt. Dabei wird zunächst auf Voraussetzungen eingegangen und dann spezifisch auf das Teilsystem 1 Import und schließlich der praktische Import der Daten skizziert. Da die anderen beiden Teilsysteme im Backend angesiedelt sind und keine Schnittstelle nach außen bieten, wird auf diese beiden nicht eingegangen. Teilsystem 3 ist soweit von Interesse, da aus diesem das Dashboard gestartet wird. Das Dashboard ist die grafische Umsetzung des Teilsystem 2 und Teilsystem 3. Beim Dashboard wird noch kurz auf das Layout eingegangen, bevor die Funktionsweise und die Datenvisualisierungen besprochen werden.

### 5.2.1 TECHNISCHE VORAUSSETZUNGEN

Der Programmcode zum Projekt ist auf github zu finden.<sup>19</sup> Dort gibt es weitere Informationen zur Installation. Das System wurde auf den Betriebssystemen macOS Big Sur, Linux in der Ubuntu-Distribution 20.10. getestet. Das Dashboard kann mit auf den aktuellen Versionen der Web-Browser Google Chrome, Firefox und Safari dargestellt werden. Als Hardware-Anforderungen wird ein Intel Dual Core i5 (Haswell) mit 128 GB Festplatte und 8 GB Arbeitsspeicher angegeben. Mit dem Programmcode werden keine bibliothekarischen Daten mitgeliefert.

### 5.2.2 DATEN-IMPORT

Der Import der Daten findet über Skripte statt. Diese liegen im Projektverzeichnis `src/instances`. Für Budget, Umsatz, Neuerwerbungen und Ausleihdaten liegen einzelne Python-Skripte bereit, die manuell über die Kommandozeile aufgerufen werden können. Zudem gibt es noch ein shell-Skript, dass die vier Skripte zusammen anstösst. Dieses muss ebenfalls

---

<sup>19</sup><http://pfad/zu/github/repo>

manuell aufgerufen werden. Nach erfolgreichem Import der Dateien aus den Importverzeichnissen wird eine einfache Erfolgsmeldung auf der Kommandozeile ausgegeben.

Die Pfade zu den lokalen Verzeichnissen für Import und Speicherung der Daten sind als Konstanten zentral in der `configuration.py` im Projektverzeichnis hinterlegt. Dort sind auch noch andere Pfadkonstanten definiert, die auf Dateien für die Datenanreicherung verweisen, welche das Teilsystem 1 und das Teilsystem 2 unterstützen. Wichtig ist, dass die zu importierenden Daten über Dateinamen einer gewissen Semantik und über ein gewisses Format verfügen, sonst werden sie nicht importiert (Siehe auch [Unterabschnitt 5.1.3](#) Teilsystem 1).

### 5.2.3 DASHBOARD

Gestartet wird die Dashboard-Application auf der Kommandozeile, indem `index.py` im Projektverzeichnis `dashboard` aufgerufen wird. Diese startet den Flask-Webserver mit der Dashboard-App. Der Webserver ist in der `index.py` so eingestellt, dass er an allen Netzwerkschnittstellen lauscht.

```
Dash is running on http://0.0.0.0:8050/
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
```

Abbildung 5.16: Start Flask Webserver

Das Dashboard kann mit der angegebenen Adresse vom Web-Browser geöffnet werden. [Abbildung 5.17](#) zeigt schematisch die Layout-Struktur des Dashboards.

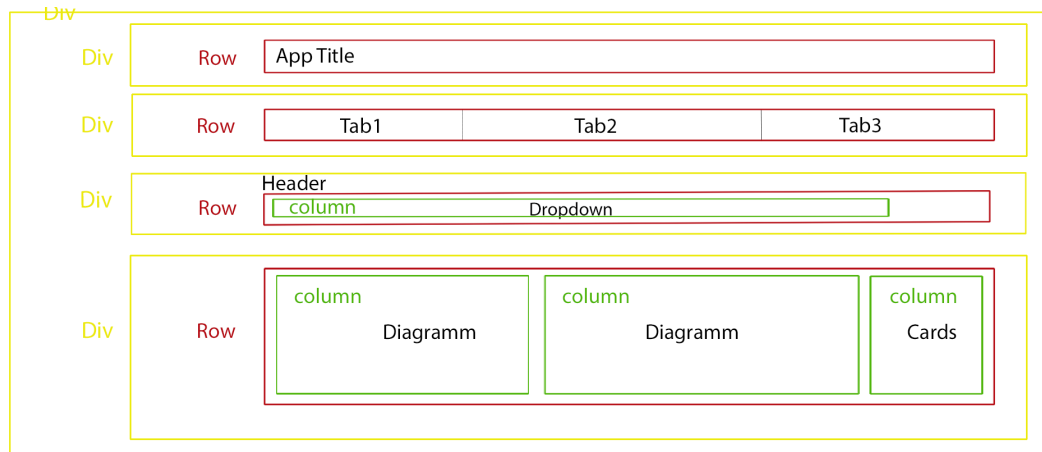


Abbildung 5.17: Struktur Layout

Die Layout-Struktur besteht aus Divs, Rows, Header und Columns, und den Steuerelementen wie Dropdown-Menüs sowie den Darstellungselementen wie Diagrammen oder Cards. Die einzelnen Rows werden durch Div-Container bemantelt. Diese sind auf dem html-Body angesiedelt. Während die ersten beiden Reihen zentral in der `index.py` festgelegt werden, werden die anderen Reihen in den einzelnen Tab-Dateien definiert. Diese beherbergen bis zu drei Columns-Elemente, in denen die Steuer- und Datenvisualisierungselemente enthalten sind. Die Columns-Elemente sind in Abhängigkeit der darzustellenden Daten unterschiedlich groß. Die Größe wird festgelegt in den einzelnen `fig_data()`-Funktionen des Teilsystems 3. Die Header gelten als inhaltlicher Trenner zwischen den einzelnen Datenvisualisierungen und dienen der schnelleren Orientierung. Die Header sind mit der jeweils nachfolgenden Row verknüpft. Das Dashboard besteht aus drei Tabs: *Umsatz und Budget*, *Lesesaal und Ausleihe*, *Neuerwerbungen und Bestand*. Der Wechsel zwischen den Tabs geschieht durch das einmalige Klicken mit der Maus auf dem Tab-Titel.

## 5 Diskussion der Umsetzung

### Tab1 - Umsatz und Budget

Nachdem die Adresse im Browser geöffnet wurde, wird der Tab *Umsatz und Budget* aufgerufen. Dieser ist als default-value im Dashboard-Layout der `index.py` eingestellt.<sup>20</sup>

Den Tab *Umsatz und Budget* zeigt [Abbildung 5.18](#).

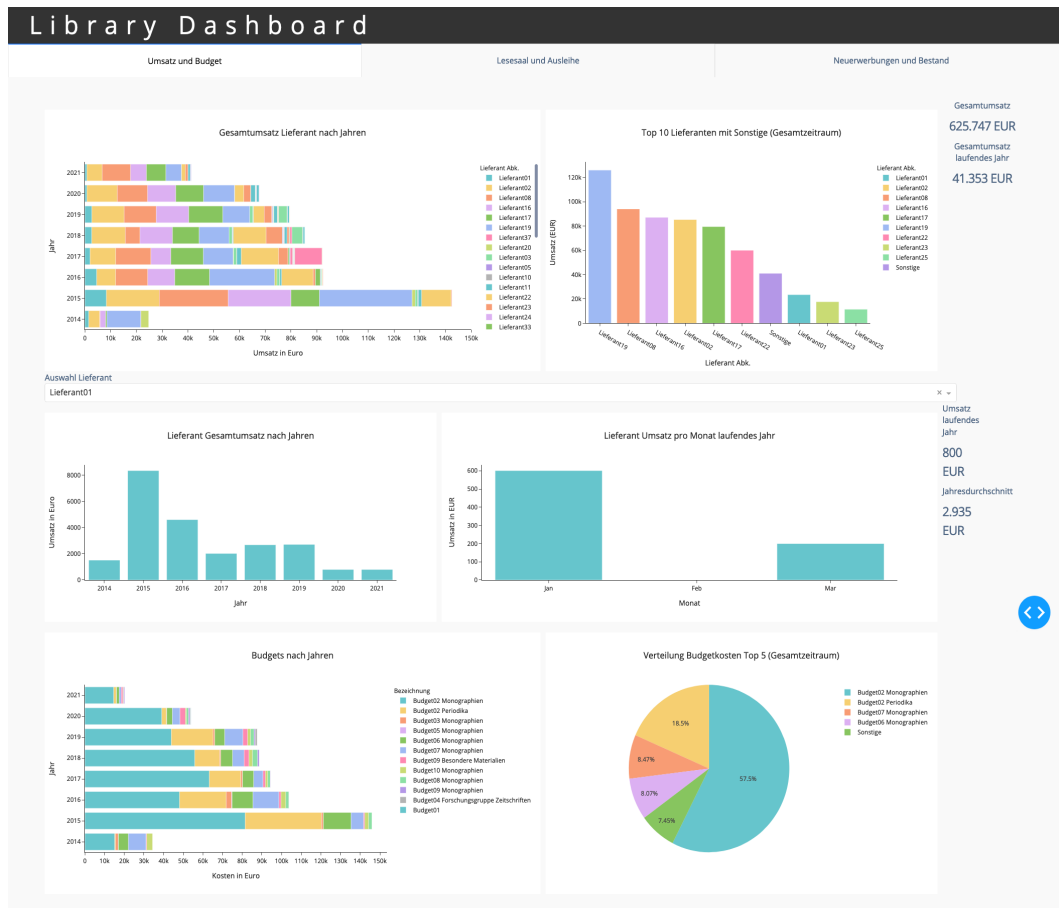


Abbildung 5.18: Tab1

<sup>20</sup>Für die Darstellung der Umsatz- und Budgetdaten werden sowohl die Kostenstellen als auch die Lieferanten maskiert dargestellt. Es wurde außerdem mit Testdaten gearbeitet, die bis März 2021 gehen.

Folgende Diagramme und Steuerelemente werden im Tab *Umsatz und Budget* dargestellt.

Row	Titel	Beschreibung	Datenset	Darstellung	Interaktivität auf dem Dashboard
1	Gesamtumsatz Lieferant nach Jahren	Die Einzelwerte der Lieferanten pro Jahr werden dargestellt.	Umsatzdaten	gestapeltes Balkendiagramm (horizontal)	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
	Top 10 Lieferanten mit Sonstige (Gesamtzeitraum)	Die 9 Lieferanten, bei denen der Umsatz am stärksten ist, werden dargestellt. Die restlichen werden in der Kategorie Sonstiges gruppiert.	Umsatzdaten	Balkendiagramm	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
	Gesamtumsatz	-	Umsatzdaten	numerischer Wert	-
	Gesamtumsatz laufendes Jahr	-	Umsatzdaten	numerischer Wert	-
2	Auswahl Lieferant	Dropdown-Menü mit eindeutigen Werten der Dataframe-Spalte 'Lieferanten Abk.'	Umsatzdaten	Dropdown-Menü	Auswahl von Werten aus einer Liste. Dadurch werden vier Darstellungen beeinflusst.
3	Lieferant Gesamtumsatz nach Jahren	Der Gesamtumsatz eines Lieferanten nach Jahren wird dargestellt.	Umsatzdaten	Balkendiagramm	Auswahl des Lieferanten über das Dropdown-Menü
	Lieferant Umsatz pro Monat laufendes Jahr	Der Gesamtumsatz eines Lieferanten nach Monaten für das laufende Jahr wird dargestellt.	Umsatzdaten	Balkendiagramm	Auswahl des Lieferanten über das Dropdown-Menü
	Umsatz laufendes Jahr (für einen Lieferanten)	-	Umsatzdaten	numerischer Wert	Wert verändert sich durch Auswahl des Lieferanten über das Dropdown-Menü
	Jahresdurchschnitt (Lieferant)	-	Umsatzdaten	numerischer Wert	Wert verändert sich durch Auswahl des Lieferanten über das Dropdown-Menü
4	Gesamtbudget Kostenstellen nach Jahren	Die Einzelwerte der Kostenstellen pro Jahr werden dargestellt.	Budgetdaten	gestapeltes Balkendiagramm (horizontal)	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
	Budgetkosten Top 5 Kostenstellen mit Sonstige (Gesamtzeitraum)	Die 4 Kostenstellen, bei denen die Budgetkosten am höchsten sind, werden dargestellt. Die restlichen werden in der Kategorie Sonstiges gruppiert.	Budgetdaten	Kreisdiagramm	Plotly-Interaktivität (Aus- und Einblenden von Anteilen, Hover-Informationen)

Tabelle 5.3: Übersicht Darstellung Tab Umsatz und Budget



Folgende Diagramme, Tabellen und Steuerelemente werden im Tab *Lesesaal und Ausleihe* dargestellt.

Row	Titel	Beschreibung	Datenset	Darstellung	Interaktivität auf dem Dashboard
1	Auswahl Jahr	Dropdown-Menü mit eindeutigen Werten der Dataframe-Spalte 'Jahr'.	Ausleihdaten	Dropdown-Menü	Auswahl von Werten aus einer Liste. Dadurch werden eine Darstellung beeinflusst.
2	Monatliche Anzahl der Nutzer:innen nach Servicezeiten	Es wird der monatliche Verlauf nach den vier Service-Zeiten dargestellt.	Lesesaaldaten	Liniendiagramm	Auswahl des Zeitraums (Jahr) über Dropdown-Menü. Plotly-Interaktivität (Aus- und Einblenden von Linien, Hover-Informationen)
	Jährliche Lesesaalnutzung nach Servicezeiten	Es wird die jährliche Nutzung des Lesesaal in den Service-Zeiten dargestellt.	Lesesaaldaten	Balkendiagramm	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
3	Jährliche Ausleihe nach den Top 10 RVK-Fachsystematiken mit Sonstige	Die 9 Fachsystematiken, bei denen die Ausleihanzahl am höchsten ist werden dargestellt. Die übrigen Fachsystematiken im Bestand werden in der Kategorie Sonstiges gruppiert.	Ausleihdaten	gestapeltes Balkendiagramm (horizontal)	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
	Gesamtverteilung Ausleihe Buchservice / Bibliothek	Es wird die prozentale Verteilung zweier Werte dargestellt.	Ausleihdaten	Kreisdiagramm	Plotly-Interaktivität (Aus- und Einblenden von Anteilen, Hover-Informationen)
4	Tabellarische Darstellung der 5 besonders nachgefragten Titel nach Jahren	-	Ausleihdaten	Tabelle mit den Spalten Jahr, Signatur, Titel und Anzahl der Ausleihen.	-

Tabelle 5.4: Übersicht Darstellung Tab Lesesaal und Ausleihe

## 5 Diskussion der Umsetzung

### Tab3 - Neuerwerbungen und Bestand

Den Tab *Neuerwerbungen und Bestand* zeigt [Abbildung 5.20](#).



Abbildung 5.20: Tab3



Folgende Diagramme werden im Tab *Neuerwerbungen und Bestand* dargestellt.

Row	Titel	Beschreibung	Datenset	Darstellung	Interaktivität auf dem Dashboard
1	Monatliche Neuerwerbungen laufendes Jahr	Es werden die monatlichen Neuerwerbungen des laufenden Jahres dargestellt.	Bestandsdaten	Balkendiagramm	-
	Jährliche Bestandsentwicklung nach Monaten	Darstellung des akkumulierten Bestandswachstums für die einzelnen Jahre	Bestandsdaten	Liniendiagramm	Plotly-Interaktivität (Aus- und Einblenden von Linien, Hover-Informationen)
2	Bestandswachstum pro Jahr und Gesamt	Darstellung des absoluten und des relativen Wachstums nach Jahren.	Bestandsdaten	überlagertes Balkendiagramm	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
	Top 10 Systematikgruppen pro Jahr	Die jährlichen Top 10 Fachsystematikgruppen im Bestand werden dargestellt.	Bestandsdaten	gestapeltes Balkendiagramm	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
3	Top 10 Systematikgruppen Ausleihen	Es werden die Top 10 Fachsystematiken der Ausleihe vom kleinsten Wert zum höchsten des Bestandes (ohne Buchservice) dargestellt.	Ausleihdaten	Balkendiagramm	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)
	Top 10 Systematikgruppen Bestand	Es werden die Top 10 Fachsystematiken im Bestand vom höchsten Wert zum kleinsten dargestellt.	Bestandsdaten	Balkendiagramm	Plotly-Interaktivität (Aus- und Einblenden von Balken, Hover-Informationen)

Tabelle 5.5: Übersicht Darstellung Tab Lesesaal und Ausleihe

## 5.3 BEWERTUNG

Das Ziel des vorliegenden Projektes war es, ein Proof-of-Concept eines Dashboards zu entwickeln, das wesentliche bibliothekarische Daten sammelt, statistisch mit geeigneten und modernen Datenvisualisierungen analysiert und auswertet. Als Ergebnis steht ein Dashboard, das die bibliothekarischen Datensets aus den drei Bereichen Umsatz und Budget, Lesesaal und Ausleihe sowie der Bestandsentwicklung darstellt. Damit ist die Umsetzung des Proof-of-Concepts gelungen. Vor allem ist es gelungen, die vorhandenen Datensets soweit zu importieren, dass auf diesen Manipulationen und Berechnungen durchgeführt werden können. Ein Anliegen war es, die Daten an einem zentralen Ort in einem einheitlichen Datenformat zu speichern. Dieses Anliegen konnte ebenfalls umgesetzt werden. Ebenso ist es gelungen, eine grafische Oberfläche für die visualisierten Daten in Form von Diagrammen und Tabellen zu erstellen.

Die Entwicklung der einzelnen Teilsysteme war unterschiedlich zeitintensiv. Der hohe Aufwand für das Teilsystem 2 stach gegenüber den anderen beiden heraus.

Die Programmiersprache Python und insbesondere die Bibliothek pandas haben sehr zum Gelingen dieses Projektes beigetragen. Das Zusammenspiel zwischen Dash und Plotly für die Erstellung einer 'Analytical Application' war sehr gut, da hier ebenfalls nur Python angewendet wird.

Leider war in der Bearbeitungszeit für das vorliegende Projekt keine Raum mehr für die Entwicklung des *Teilsystems 4 Standardbericht*. Dieses sollte ein Dokument erstellen, der die wichtigsten KPI der Bibliothek enthält und den stakeholdern der Bibliothek übergeben werden kann.

### 5.3.1 DATENLAGE

In dem vorliegenden Projekt wurden hauptsächlich die Bibliotheksdaten aus dem hebis-Verbund sowie aus dem LBS verarbeitet. Das ermöglicht den Bibliotheken, die vom hebis-Verbund die gleichen Daten wie die Institutsbibliothek bekommen, das vorliegende System leicht zu implementieren. Der Workflow für den Import, die Weiterverarbeitung und die Darstellung der Umsatz-, Budget-, und Bestandsdaten kann durch das System abgedeckt werden, die hierfür benötigten Daten der Bibliothek regelmäßig zur Verfügung gestellt werden. Die Titel-Ausleihdaten werden nicht automatisch geliefert. Dadurch entsteht eine Schieflage zwischen Bestandsdaten und Ausleihdaten in Bezug auf die Bestandsgröße. Deshalb müssten im Regelbetrieb die bibliotheksinternen Prozesse und das System angepasst werden, um einen regelmäßigen Abzug der Ausleihdaten verarbeiten zu können. Zur Zeit ist das System für die vorliegenden Daten zugeschnitten.

Hindernisse lagen in den Daten selbst, die überwunden werden mussten. Zu nennen wären hier fehlende Daten in den Umsatz- und Budgetdaten. Diese fehlenden Daten können Fehler in der Darstellung im Dashboard erzeugen.<sup>21</sup> Des Weiteren wird bei dem

---

<sup>21</sup>Ebenso fehlen noch relevante Lieferanten, die sich nicht im LBS finden lassen.

monatlichen Abzug der Neuerwerbungsdaten aus dem CBS Duplikate mit exportiert. Diese Duplikate entstehen durch die Mehrfachexemplare, die an einem Datensatz hängen. Diese lassen sich zwar als Duplikate relativ einfach über die Signatur erkennen und entfernen. Darunter würden dann aber auch alle ebooks fallen, da sie die gleiche Signatur aufweisen.

Mit diesen Ausnahmen innerhalb der Daten musste und wurde ein Umgang gefunden. Diese haben mitunter alle Teilsysteme affektiert und die Programmierung dieser mitbestimmt.

In Bezug auf die Darstellung der Daten im Dashboard stellt die Vielzahl von diskreten Werten ein Problem dar. Die Grenze der Darstellbarkeit der Daten in Form von Diagrammen ist erreicht, wenn zu viele diskrete Werte dargestellt werden müssen. Das betrifft zum Beispiel die Vielzahl an Lieferanten in dem Diagramm 'Gesamtumsatz Lieferant nach Jahren' im Tab *Umsatz und Budget*. Dort kann anhand der Farbe in den gestapelten Balkendiagramm nicht mehr zwischen den einzelnen Lieferanten unterschieden werden, da die Color-Palette nur eine bestimmte Anzahl an diskreten Werten darstellen kann [Vgl. [Plo21e](#)].<sup>22</sup> Ebenso betrifft dieses Problem die Auswertung der Daten nach den *RVK*-Fachsystematiken beziehungsweise nach deren Untergruppen. Erste Datenanalysen haben verdeutlicht, dass es eine nicht darstellbare Anzahl an Fachsystematiken im Bestand der Bibliothek gibt, die visuell nicht mehr darstellbar ist. Aufgrund dieser Darstellungsprobleme wurden noch andere Diagramme erstellt, die die Anzahl der darzustellenden Werte reduziert. So wurde ein Diagramm erstellt, das eine bestimmte Anzahl der umsatzstärksten Lieferanten im Gesamtzeitraum zeigt. Bei den *RVK*-Fachsystematiken wurde ebenso verfahren.

---

<sup>22</sup>Auch wäre die Frage zu stellen, inwieweit hier die Darstellung aller Lieferanten sinnvoll im Hinblick auf die Informationsrezeption.

### 5.3.2 UMGESetzte ANFORDERUNGEN DER ANFORDERUNGSANALYSE

Im folgenden wird sich auf die Muss-Anforderungen der Anforderungsanalyse konzentriert, die durch das Proof-of-Concept umgesetzt werden sollten. Dabei werden die Rahmenbedingungen, die funktionalen sowie die nicht-funktionalen Anforderungen betrachtet.

Im Bereich des ETL-Prozesses und der Datenspeicherung ([Tabelle 4.2](#)) wurden die Anforderungen F1, F2, F4, und F5 erfüllt. Die Anforderung F3 musste nicht umgesetzt werden, da die Daten bereits so vorlagen, dass auf ihnen ohne automatische Harmonisierung weitergearbeitet werden konnte. Bedingt wurde dennoch diese Anforderung durch die Datenanreicherung abgedeckt.

Die funktionalen Anforderungen der Datenanalyse ([Tabelle 4.3](#)) wurden vollständig erfüllt (F10, F11, F12, F13, F14). Die funktionalen Anforderungen F15 und F16 des Bereiches Datenpräsentation und Standardbericht ([Tabelle 4.4](#)) wurden ebenso erfüllt. Es gibt eine Vielzahl an Datenvisualisierungen, die den Benutzer:innen auf dem Dashboard angeboten werden. Diese bestehen neben Kreisdiagrammen und einer Tabelle hauptsächlich aus Linien- und Balkendiagrammen. Die Erzeugung des Standardberichts wurde nicht mehr implementiert. Dementsprechend wurden die diesbezüglichen Anforderungen (F18, F20, F21) nicht umgesetzt.

Von den nicht-funktionalen Anforderungen ([Tabelle 4.5](#)) wurden folgende erfüllt: NF1, NF5, NF6, NF8, NF11, NF12, NF15, NF16, NF17, NF18. Da das System nur ein Proof-of-Concept darstellt, wurde es noch nicht anderen Benutzer:innen vorgestellt, deswegen kann die Anforderung NF2 'Das System ist leicht erlernbar' noch nicht als erfüllt gelten.

Die Grenze der Darstellbarkeit in Diagrammen betrifft die Umsetzung der nicht-funktionalen Anforderung NF14, die nicht vollständig realisiert werden konnte. Da nur eine bestimmte Anzahl an Werten übersichtlich präsentiert werden kann, gilt es die Auswahl der Diagramme beziehungsweise der in ihnen dargestellten Daten zu evaluieren.

### 5.3.3 UMGESETZTE ANWENDUNGSFÄLLE

Die Anwendungsfälle 1, 3, 4, 5, 6 wurden bearbeitet. Die Anwendungsfälle 2 und 7 konnten in dem Bearbeitungszeitraum nicht mehr bearbeitet werden, da die Zeit im Bearbeitungszeitraum nicht mehr ausgereicht hat. Gründe hierfür waren unter anderem eine komplexere Tabellenstruktur (Anwendungsfall 2).

#### *Anwendungsfall 1*

Titel: Ausleihzahlen Bibliotheksbestand

Das Ziel des *Anwendungsfall 1* ist die Darstellung der Anzahl der Ausleihen. Die Ergebnisse befinden sich in dem *Tab2 - Lesesaal und Ausleihe* des Dashboards.

Anforderung des Anwendungsfalls	Titel des Diagramms / Tabelle im Dashboard	Bemerkung
Das System filtert die betreffenden Datensätze nach Jahr. Das System zeigt diese Datensätze mit Datenvisualisierungen an.	-	Durch die Teilsysteme 2 und 3 gewährleistet.
Das System zeigt darüberhinaus die Verteilung der ausgeliehenen Titel in der RVK-Fachsystematiken pro Jahr an.	Jährliche Ausleihe nach den Top 10 RVK-Fachsystematiken mit Sonstige	Die Anzahl der RVK-Fachsystematiken kann im Backend als Parameter eingestellt werden.
Das System zeigt darüberhinaus die ausleihstärksten Titel absteigend nach Anzahl und Jahr an.	Tabellarische Darstellung der 5 besonders nachgefragten Titel nach Jahren	Die Anzahl der Titel kann im Backend als Parameter eingestellt werden.
Das System zeigt im Vergleich mit dem Bibliotheksbestand die Top-Fachsystematikgruppen an.	Top 10 Systematikgruppen Ausleihen <sup>23</sup>	Die Anzahl der RVK-Fachsystematiken kann im Backend als Parameter eingestellt werden.
Das System zeigt die Verteilung der Ausleihe unterschieden in Bibliotheksbestand und Buchservice an.	Gesamtverteilung Ausleihe Buchservice / Bibliothek	-

Tabelle 5.6: Anwendungsfall 1 - Umgesetzte Anforderungen

## 5 Diskussion der Umsetzung

### Anwendungsfall 3

Titel: Lesesaalnutzung

Das Ziel des *Anwendungsfall 3* ist Anzeige der Nutzung des Lesesaals während der Öffnungszeiten. Die Ergebnisse befinden sich in dem *Tab2 - Lesesaal und Ausleihe* des Dashboards.

Anforderung des Anwendungsfalls	Titel des Diagramms / Tabelle im Dashboard	Bemerkung
Das System filtert die betreffenden Datensätze nach Jahr. Das System zeigt diese Datensätze mit Datenvisualisierungen an.	-	Durch die Teilsysteme 2 und 3 gewährleistet.
Das System zeigt die Nutzung des Lesesaals nach Monat und Jahr, gruppiert in vier Zeitschichten an.	Monatliche Anzahl der Nutzer:innen nach Servicezeiten	Die Jahre können im Dropdown-Menü im Frontend ausgewählt werden.

Tabelle 5.7: Anwendungsfall 3 - Umgesetzte Anforderungen

---

<sup>23</sup>Dieses Diagramm befindet sich in dem Dashboard Tab *Neuerwerbungen und Bestand* als zur verständlichen Gegenüberstellung zu den Bestandsdaten.

*Anwendungsfall 4*

Titel: Neuerwerbungen

Das Ziel des *Anwendungsfall 4* ist die Anzeige der Anzahl der Neuerwerbungen nach *RVK*-Fachsystematiken pro Monat. Die Ergebnisse befinden sich in dem *Tab3- Neuerwerbungen und Bestand* des Dashboards.

Anforderung des Anwendungsfalls	Titel des Diagramms / Tabelle im Dashboard	Bemerkung
Das System filtert die betreffenden Datensätze. Das System zeigt diese Datensätze mit Datenvisualisierungen an.	-	Durch die Teilsysteme 2 und 3 gewährleistet.
Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen wählen einen Monat auf der <i>GUI</i> aus.	-	Diese Interaktion wurde nicht zusätzlich implementiert, da die Darstellung durch die zwei anderen Diagramme als genügend betrachtet wurde.
Das System zeigt die Neuerwerbungen des laufenden Jahres an.	Monatliche Neuerwerbungen laufendes Jahr	-
Das System zeigt die jährliche Bestandsentwicklung pro Monat an.	Jährliche Bestandsentwicklung nach Monaten	Der Verlauf der Bestandsentwicklung der einzelnen Jahre wird akkumuliert angezeigt.
Das System zeigt darüberhinaus die Anzahl der Titel nach Medienart an.	-	Diese Anforderung wurde zwar im Programmcode implementiert, aber nicht im Dashboard umgesetzt, da das gedruckte Buch als Medium dominiert.

Tabelle 5.8: Anwendungsfall 4 - Umgesetzte Anforderungen

## 5 Diskussion der Umsetzung

### Anwendungsfall 5

Titel: Bestandswachstum

Das Ziel des *Anwendungsfall 5* ist die Anzeige Wachstums des Bibliotheksbestandes insgesamt und nach einzelnen *RVK*-Fachsystematiken. Die Ergebnisse befinden sich in dem *Tab3 - Neuerwerbungen und Bestand* des Dashboards.

Anforderung des Anwendungsfalls	Titel des Diagramms / Tabelle im Dashboard	Bemerkung
Das System filtert die betreffenden Datensätze. Das System zeigt diese Datensätze mit Datenvisualisierungen an.	-	Durch die Teilsysteme 2 und 3 gewährleistet.
Das System zeigt die Top- <i>RVK</i> -Fachsystematiken des Bestandes pro Jahr an.	Top 10 Systematikgruppen pro Jahr	Die Anzahl der Fachsystematiken kann im Backend als Parameter eingestellt werden.
Das System zeigt darüberhinaus die Gesamtzahl der Titel nach Jahren an.	Bestandswachstum pro Jahr und Gesamt	-
Das System zeigt die Top- <i>RVK</i> -Fachsystematiken des Bestandes insgesamt an.	Top 10 Systematikgruppen Bestand	Die Anzahl der Fachsystematiken kann im Backend als Parameter eingestellt werden.

Tabelle 5.9: Anwendungsfall 5 - Umgesetzte Anforderungen



*Anwendungsfall 6*

## Titel: Umsatz- und Budgetübersicht

Das Ziel des *Anwendungsfall 6* ist die Anzeige der Umsatz- und Budgetübersicht für den Gesamtzeitraum und das laufende Jahr. Die Ergebnisse befinden sich in dem *Tab1 - Umsatz und Budget* des Dashboards.

Anforderung des Anwendungsfalls	Titel des Diagramms / Tabelle im Dashboard	Bemerkung
Das System filtert die betreffenden Datensätze. Das System zeigt diese Datensätze mit Datenvisualisierungen an.	-	Durch die Teilsysteme 2 und 3 gewährleistet.
Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen können sich den Lieferanten auswählen und den Gesamtumsatz und den Umsatz pro Jahr ansehen.	Gesamtumsatz Lieferant nach Jahren, Lieferant Gesamtumsatz nach Jahren, Lieferant Umsatz pro Monat laufendes Jahr	Die einzelnen Lieferanten können im Dropdown-Menü im Frontend ausgewählt werden.
Das System zeigt den Umsatz im laufenden Jahr und den Jahresdurchschnittsumsatz dieses Lieferanten.	Umsatz laufendes Jahr (für einen Lieferanten), Jahresdurchschnitt (Lieferant)	Die einzelnen Lieferanten können im Dropdown-Menü im Frontend ausgewählt werden.
Das System zeigt die umsatzstärksten Lieferanten im Gesamtzeitraum an.	Top 10 Lieferanten mit Sonstige (Gesamtzeitraum)	Die Anzahl der Lieferanten kann im Backend als Parameter eingestellt werden.
Das System zeigt den Gesamtumsatz im Gesamtzeitraum und im laufenden Jahr	Gesamtumsatz, Gesamtumsatz laufendes Jahr	-
Die Bibliotheksleitung und die Bibliotheksmitarbeiter:innen können sich die Budgetübersicht ansehen.	Gesamtbudget Kostenstellen nach Jahren	-
Das System zeigt die kostenintensivsten Kostenstellen im Gesamtzeitraum an.	Budgetkosten Top 5 Kostenstellen mit Sonstige (Gesamtzeitraum)	Die Anzahl der Kostenstellen kann im Backend als Parameter eingestellt werden.

Tabelle 5.10: Anwendungsfall 6 - Umgesetzte Anforderungen



## 6 SCHLUSS

Mit der vorliegenden Arbeit sollte ein Dashboard als Proof-of-Concept entwickelt werden. Vom Datenimport über die Datenaufbereitung bis hin zur Darstellung im Dashboard wurde ein System geschaffen, das für die vorhandenen Datensets funktioniert. Das System stellt eine solide Basis dar, mit der neue Anforderungen implementiert werden können.

Wie im Kapitel 5 dargelegt ist, wurden fast alle Anforderungen und Anwendungsfälle umgesetzt. Offen blieben die Anwendungsfälle 2 und 7, die aus zeitlichen Gründen nicht mehr bearbeitet werden konnten. Neben diesen beiden Anwendungsfällen, wäre eine Weiterentwicklung des Systems über den Projektzeitraum hinaus denkbar und wünschenswert. Die Weiterentwicklung betrifft alle Teilsysteme sowie die Darstellung im Dashboard. Zu der Weiterentwicklung gehört auch, zusätzliche Daten in die statistische Auswertung miteinzubeziehen. Die Nutzung elektronischer Ressourcen spielt innerhalb der *MPG* eine wichtige Rolle. So könnte die Auswertung von *COP 5*-Statistiken weiteren Aufschluss darüberbieten, welche elektronischen Ressourcen wie stark genutzt werden. Ebenfalls könnte die Auswertung der Herkunft der zur Verfügung gestellten wissenschaftlichen Artikel für die Wissenschaftler:innen durch Dokumentliefersdienste Indizien geben, welche Zeitschriftenabonnements zusätzlich angeschafft oder lizenziert werden könnten. Dafür müssen aber erst einmal die Informationen zusammengetragen werden. Im Bereich des Datenimports könnte der Import zeitbasiert erfolgen. So könnte über einen CronJob einmal im Monat die Daten vollautomatisch importiert werden. Dabei wäre wichtig, die Ergebnisse des Import-Prozesses in einer log-Datei zu protokollieren. Umgesetzt werden

könnte die Protokollierung mit der Python Bibliothek Logging. Weiterhin könnte darüber nachgedacht werden, ob perspektivisch, die Flatfile-Structure durch ein Datenbanksystem zu ersetzen. Außerdem wäre zu überlegen, ob nicht auch die unterschiedlichen Datensets zusammengeführt werden und die Daten nach mehreren Dimensionen ausgewertet werden können, was durch die vorliegende Arbeit nicht geleistet werden konnte. Ein Beispiel für die Zusammenführung wären die Bestands- und Ausleihdaten.

Verbesserungen im Bereich der Datenaufbereitung sind die Auswertungen der *RVK*-Fachsystematiken mit den entsprechenden Benennungen zu vollziehen. Diese würden die Aussagekraft der Diagramme für ein fachfremdes Publikum erhöhen und verbessern. Auch wäre denkbar die Daten nach anderen Anforderungen zu untersuchen und mit fortgeschritteneren statistischen Methoden auszuwerten. Zum Beispiel wäre die Korrelation zwischen Bestands- und Ausleihdaten interessant. Diese könnten dann innerhalb einer Datenvisualisierung gezeigt werden. Die Erscheinung und die Darstellung im Dashboard könnten ebenfalls überarbeitet werden. Eine sinnvolle Erweiterung wäre die Anpassung des Dashboard-Layouts mit Responsive-Design-Technologien an mobile Geräte. Dies könnte geschehen durch einen vermehrten Einsatz der `dash_bootstrap_components`. Die Funktionalitäten von Plotly Express wurden für die Diagramme nicht ausgeschöpft. Hier bedürfen die Diagramme mitunter noch Feineinstellungen, die während der Bearbeitungszeit des Proof-of-Concepts nicht umgesetzt werden konnten. Ferner könnten weitere Interaktionen implementiert werden. Die `Dash_core_components` bieten noch andere interaktive Elemente wie zum Beispiel Slider für die Begrenzung der zeitliche Begrenzung. Solche interaktiven Komponenten sind nicht kompliziert zu implementieren. Sie sorgen aber dafür, dass sich der Programmcode deutlich vergrößert. Dessen ungeachtet sollte ebenso der Programmcode überarbeitet werden. Diese wäre notwendig in dem `data_prep`-Modul, dass aus einer Datei die die Basis- und den Kindklassen enthält. In Python ist die Struktur mehrerer Klassen in einer Modul durchaus üblich. Viele Module der Python Standardbibliothek haben viele Klassen in einer Datei. Diese sind einfacher zu importieren.

Sollte sich aber die Codebase durch die Hinzunahme zusätzlicher Daten weiter vergrößern, wäre es sinnvoll die Klassen in weiteren Dateien aufzusplitten. Auch der Programmcode an sich könnte sicherlich vereinfacht und optimiert werden nach dem DRY-Prinzip (Don't repeat yourself). Dies betrifft die Implementation der Dashboard-Logik. Da die Diagramme in dem Dashboard ähnliche Parameter empfangen, ließen sich hier vermutlich Funktionen / Methoden für einen Diagrammtyp entwickeln. Ferner wäre es schön, wenn die Layoutlogik (Hintergrundfarbe, Größe) von dem Erstellen der Diagramme separiert werden könnte. Generell muss auch zwischen Lesbarkeit, kondensierter Komplexität[Vgl. [Ous18](#)] und expliziten Programmcode abgewogen werden. Ebenfalls sollten die Methoden und Funktionen noch ausführlicher getestet werden mit zum Beispiel der Bibliothek `pytest`. Ziel wäre es, den Programmcode robuster zu machen. Für das Testen in diesem Sinne, blieb ebenfalls keine Zeit während der Bearbeitung.

Viel Zeit wurde auf die Datenanalyse verwendet. Dagegen viel es leicht die Datenvisualisierungen und das Dashboard zu erstellen. Die Analyse der Daten aus vielen Bereichen war dann viel. Ebenso war es am Anfang schwierig mit Python und `pandas`. Da es das erste große Projekt mit Python und den `pandas`, `plotly` und `dash` war, war der Beginn der Entwicklung schwierig. Nichtsdestotrotz war es die richtige Wahl dieses Projekt mit den ausgewählten Sprache und den Bibliotheken umzusetzen, da sie sehr mächtig und flexibel sind. Dies zeichnen sich unter anderem dadurch aus, dass es mehrere Lösungsalternativen für ein Problem geben kann. Gerade als Anfänger da schon den Überblick verlieren kann. Während insbesondere der Programmierung gab es eine Lernkurve, die als bereichernd empfunden wurde. Die Planung für ein Projekt mit abgesteckten Zeitrahmen ist sehr wichtig und hat bis auf ein paar Ausnahmen sehr gut funktioniert. Mit dem Fortschreiten des Projektes hat man eine bessere Aufwandseinschätzung gewonnen, insbesondere in der Programmierung, aber auch im schreiben des theoretischen Teils. So zum Beispiel bei manchen Fehlern, die frühzeitig gemacht wurden, und die erst spät auffallen. Das verlangt eine Flexibilität ab, aber auch einen Mut zur Lücke.



# TABELLENVERZEICHNIS

3.1	Informationsdienstleistungen nach Basisfunktionen der Spezialbibliothek . . . . .	25
3.2	Liste der Dienstleistungsbereiche zu denen statistische Daten erhoben werden . . . . .	26
4.1	Rahmenbedingungen . . . . .	30
4.2	Funktionale Anforderungen - ETL-Prozess und Datenspeicherung . .	31
4.3	Funktionale Anforderungen - Datenanalyse . . . . .	31
4.4	Funktionale Anforderungen - Datenpräsentation und Standardbericht	32
4.5	Nicht-funktionale Anforderungen . . . . .	33
4.6	Anwendungsfall 1 - Ausleihzahlen Bibliotheksbestand . . . . .	34
4.7	Anwendungsfall 2 - Ausleihzahlen bibliotheksinterne Lieferdienste . .	35
4.8	Anwendungsfall 3 - Lesesaalnutzung . . . . .	36
4.9	Anwendungsfall 4 - Neuerwerbungen . . . . .	37
4.10	Anwendungsfall 5 - Bestandswachstum . . . . .	38
4.11	Anwendungsfall 6 - Umsatz- und Budgetübersicht . . . . .	39
4.12	Anwendungsfall 7 - Standardbericht . . . . .	40
5.1	Liste der grundlegenden Programmiersprache und Frameworks . . . .	45
5.2	Liste der Teilsysteme . . . . .	46
5.3	Übersicht Darstellung Tab Umsatz und Budget . . . . .	69

5.4	Übersicht Darstellung Tab Lesesaal und Ausleihe . . . . .	71
5.5	Übersicht Darstellung Tab Lesesaal und Ausleihe . . . . .	73
5.6	Anwendungsfall 1 - Umgesetzte Anforderungen . . . . .	77
5.7	Anwendungsfall 3 - Umgesetzte Anforderungen . . . . .	78
5.8	Anwendungsfall 4 - Umgesetzte Anforderungen . . . . .	79
5.9	Anwendungsfall 5 - Umgesetzte Anforderungen . . . . .	80
5.10	Anwendungsfall 6 - Umgesetzte Anforderungen . . . . .	81



# ABBILDUNGSVERZEICHNIS

2.1	Statistische Datentypen mit Aussagegehalten und Beispielen . . . . .	15
2.2	Schichten eines Business-Intelligence-Systems . . . . .	18
3.1	Anzahl der Titelausleihen nach Identifikationsnummer und Jahr . . . . .	26
3.2	Monatliche Budget- und Umsatzübersicht . . . . .	27
5.1	pandas Dataframe (links) und pandas Series (rechts) . . . . .	42
5.2	Systemarchitektur . . . . .	45
5.3	Systemarchitektur Teilsystem 1 . . . . .	47
5.4	Klassendiagramm - Teilsystem 1 Import . . . . .	48
5.5	Datenfluss - Teilsystem 1 Import . . . . .	48
5.6	Beispiel Extraktion Fachsystematik . . . . .	50
5.7	Monatliche Umsatzübersicht nach Ablauf Teilsystem 1 Import . . . . .	52
5.8	Systemarchitektur Teilsystem 2 . . . . .	53
5.9	Klassendiagramm - Teilsystem 2 Datenbearbeitung . . . . .	54
5.10	Systemarchitektur Teilsystem 3 . . . . .	56
5.11	Struktur Dash Multi-Page App . . . . .	57
5.12	Ablauf Tab . . . . .	59
5.13	Funktionsaufrufe Tab . . . . .	59
5.14	Ablauf Tab mit Callback . . . . .	63
5.15	Ablauf Tab mit index . . . . .	64

5.16	Start Flask Webserver . . . . .	66
5.17	Struktur Layout . . . . .	67
5.18	Tab1 . . . . .	68
5.19	Tab2 . . . . .	70
5.20	Tab3 . . . . .	72

# QUELLCODEVERZEICHNIS

5.1	Beispiel Methode Exenditures class . . . . .	55
5.2	Funktion fig_total_expnd() Auszug 1 . . . . .	60
5.3	fig_total_expnd() Auszug 2 . . . . .	60
5.4	html_fig_total_expnd() . . . . .	61
5.5	html_fig_total_expnd() . . . . .	62



# AKRONYME

BI	Business Intelligence
BIX	Bibliotheksindex
CBS	Zentralsystem
COP 5	Counter 5
COUNTER	Counting Online Usage of NeTworked Electronic Resource
CSV	Comma-separated values
DBS	Deutsche Bibliotheksstatistik
DWH	Data Warehouse
ETL	Extract, Transform, Load
GUI	Graphische Benutzeroberfläche
hbz	Hochschulbibliothekszenrum NRW
HeBis	Hessisches Bibliotheksinformationssystem
KBN	Kompetenznetzwerk für Bibliotheken

## *Akronyme*

KDD	Knowledge Discovery in Databases
KMK	Kultusministerkonferenz
KPI	Key Performance Indicators
LBS	Lokalsystem
mpdl	max-planck-digital-library
MPG	Max-Planck-Gesellschaft
MPI EA	MPI für empirische Ästhetik
NOIR	Nominal-, Ordinal-, Intervall-, Ratio-Systematik
OCLC	Online Computer Library Center
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OPAC	Online Public Access Catalog
PuRe.MPG	Publikationsrepositorium der Max-Planck-Gesellschaft
RVK	Regensburger Verbundklassifikation
STM	Science, Technology, and Medicine
XML	Extensible Markup Language

# LITERATURVERZEICHNIS

- [Alt21] Altair. *Altair: Declarative Visualization in Python — Altair 4.1.0 documentation*. 2021. URL: <https://altair-viz.github.io/index.html> (besucht am 27. 01. 2021).
- [AM17] D. Abts und W. Mülder. *Grundkurs Wirtschaftsinformatik : eine kompakte und praxisorientierte Einführung*. 9., erweiterte und aktualisierte Auflage. Springer, Wiesbaden, 2017. xvi, 758 Seiten.
- [Ban16] C. Bange. „Werkzeuge für analytische Informationssysteme“. In: *Analytische Informationssysteme : Business Intelligence-Technologien und -Anwendungen*. 5., vollständig überarbeitete Auflage. Springer Gabler, Berlin, 2016, S. 114–126.
- [Bec+16] T. Becker, R. Herrmann, V. Sandor, D. Schäfer, und U. Wellisch. *Stochastische Risikomodellierung und statistische Methoden*. Statistik und ihre Anwendungen. Springer, Berlin, 2016. xiv, 375 Seiten.
- [BS04] J. C. Blake und S. P. Schleper. „From data to decisions“. *Library Collections, Acquisitions, & Technical Services* 28:4, 2004, S. 460–464. DOI: [10.1080/14649055.2004.10766018](https://doi.org/10.1080/14649055.2004.10766018). (Besucht am 01. 09. 2020).
- [BS10] J. Bortz und C. Schuster. *Statistik für Human- und Sozialwissenschaftler : mit ... 163 Tabellen*. 7., vollst. überarb. und erw. Aufl. Springer-Lehrbuch. Springer, Berlin, 2010, xvi, 655 Seiten.

- [BS15] M. Block und F. Seeliger. *BibloVis - Ein webbasierter One-Stop Shop zur zentralen Verwaltung und Visualisierung heterogener Nutzungsdaten aus dem Bibliothekskontext*. Type: Konferenzabstract. 104. Deutscher Bibliothekartag (Nürnberg, 2015), 2015.
- [BT20] Baker und Taylor. *Select, Manage and Promote your collection*. 2020. URL: <https://www.collectionhq.com/> (besucht am 28. 09. 2020).
- [Cai16] A. Cairo. *The truthful art : data, charts, and maps for communication*. New Riders, [San Francisco, CA], 2016. xvii, 382 Seiten.
- [Cle11] T. Cleff. *Deskriptive Statistik und moderne Datenanalyse : eine computergestützte Einführung mit Excel, PASW (SPSS) und STATA*. 2., überarbeitete und erweiterte Auflage. Gabler Verlag, Wiesbaden, 2011. xxvi, 227 Seiten.
- [Cou20] Counter. *Abstract | Project Counter*. Abstract | Project Counter. 2020. URL: <https://www.projectcounter.org/code-of-practice-five-sections/foreword/> (besucht am 08. 09. 2020).
- [DDG20] E. Dong, H. Du, und L. Gardner. „An interactive web-based dashboard to track COVID-19 in real time“. *The Lancet Infectious Diseases* 20:5, 2020, S. 533–534. DOI: [10.1016/S1473-3099\(20\)30120-1](https://doi.org/10.1016/S1473-3099(20)30120-1).
- [Dea20] Deal. *Projekt Deal*. Projekt DEAL – Bundesweite Lizenzierung von Angeboten großer Wissenschaftsverlage. 2020. URL: <https://www.projekt-deal.de/> (besucht am 08. 09. 2020).
- [Eck11] W. W. Eckerson. *Performance dashboards : measuring, monitoring, and managing your business*. Second edition. John Wiley & Sons, Inc., Hoboken, New Jersey, 2011. xvii, 318 Seiten.
- [Fac21] Faculty. *Dash bootstrap components*. 2021. URL: <https://dash-bootstrap-components.opensource.faculty.ai/> (besucht am 27. 01. 2021).



- [Far11] K. Farkisch. *Data-Warehouse-Systeme kompakt : Aufbau, Architektur, Grundfunktionen*. Springer, Wiesbaden, 2011. xi, 122 Seiten.
- [Few06] S. Few. *Information dashboard design : the effective visual communication of data*. O'Reilly & Associates, Sebastopol, CA, 2006. viii, 211 Seiten.
- [Few09] S. Few. *Now you see it : simple visualization techniques for quantitative analysis*. Analytics Press, Oakland, Calif., 2009. xi, 327 Seiten.
- [Few12] S. Few. *Show me the numbers : designing tables and graphs to enlighten*. Second edition. El Dorado Hills, Calif, 2012. xviii, 351 Seiten.
- [FF16] J. L. Finch und A. R. Flenner. „Using data visualization to examine an academic library collection“. *College and Research Libraries* 77:6, 2016, S. 765–778. DOI: [10.5860/crl.77.6.765](https://doi.org/10.5860/crl.77.6.765). (Besucht am 30. 08. 2020).
- [Gol18] U. Golas. „Statistische Abfragen mit Alma für die Fachreferatsarbeit“. *o-bib. Das offene Bibliotheksjournal / Herausgeber VDB* 5:4, 2018. DOI: [10.5282/o-bib/2018H4S44-57](https://doi.org/10.5282/o-bib/2018H4S44-57). (Besucht am 30. 08. 2020).
- [Gro20] K.-D. Gronwald. *Integrierte Business-Informationssysteme : ganzheitliche, geschäftsprozessorientierte Sicht auf die vernetzte Unternehmensprozesskette ERP, SCM, CRM, BI, Big Data Analytics*. 3., überarbeitete Auflage. Springer Vieweg, Berlin, 2020. XIX, 177 Seiten.
- [HKP12] J. Han, M. Kamber, und J. Pei. *Data Mining : concepts and techniques*. Third edition. Elsevier, Amsterdam, 2012. xxxv, 703 Seiten.
- [HTW18] L. Horne-Popp, E. Tessone, und J. Welker. „If you build it, they will come: creating a library statistics dashboard for decision-making“. In: *Developing In-House digital tools in library spaces*. Hrsg. von L. Costello. IGI Global/-Information Science Reference, Hershey, PA, 2018, S. 177–203.

- [Hug16] M. Hughes. „A long-term study of collection use based on detailed Library of Congress Classification: a statistical tool for collection management decisions“. *Collection Management* 41:3, 2016, S. 152–167. DOI: [10.1080/01462679.2016.1169964](https://doi.org/10.1080/01462679.2016.1169964). (Besucht am 30. 08. 2020).
- [Inm05] W. H. Inmon. *Building the data warehouse*. 4. ed. Wiley, Indianapolis, IN, 2005. xxviii, 543 Seiten.
- [Jil04] C. Jilovsky. „Library Statistics: reflecting yesterday, today and tomorrow“, 2004. URL: [https://www.caval.edu.au/assets/files/Research\\_and\\_Advocacy/Library\\_Statistics-reflecting\\_yesterday\\_today\\_and\\_tomorrow-Northumbria\\_2005.pdf](https://www.caval.edu.au/assets/files/Research_and_Advocacy/Library_Statistics-reflecting_yesterday_today_and_tomorrow-Northumbria_2005.pdf) (besucht am 01. 09. 2020).
- [JM15] J. Johannsen und B. Mittermaier. „Bestands- und Beschaffungsevaluierung“. In: *Praxis Handbuch Bibliotheksmanagement*. Hrsg. von R. Griebel, H. Schäffler, und K. Söllner. Bd. 1. De Gruyter, Berlin, 2015, S. 252–269.
- [Joh14] P. Johnson. *Fundamentals of collection development and management*. Third edition. Ala edition, Chicago, 2014. xiv, 554 Seiten.
- [KBM10] H.-G. Kemper, H. Baars, und W. Mehanna. *Business intelligence - Grundlagen und praktische Anwendungen : eine Einführung in die IT-basierte Managementunterstützung*. 3., überarb. und erw. Aufl. Vieweg + Teubner, Wiesbaden, 2010. ix, 298 Seiten.
- [Kir19] A. Kirk. *Data visualisation : a handbook for data driven design*. 2nd edition. Sage, Los Angeles, 2019. 312 Seiten.
- [KM20] A. Kutlay und C. Murgu. „Shiny Fabric: a lightweight, Open-Source-Tool for visualizing and reporting library relationships“. *Code4Lib* 47, 2020. URL: <https://journal.code4lib.org/articles/14938>.

- [KWC06] J. E. Knievel, H. Wicht, und L. S. Connaway. „Use of Circulation Statistics and Interlibrary Loan Data in Collection Management“. 2006 67:1, 2006, S. 35–49. DOI: [10.5860/crl.67.1.35](https://doi.org/10.5860/crl.67.1.35). (Besucht am 29. 08. 2020).
- [Lai13] M. Laitinen. „Library statistics with confidence: facts from figures with no fear“. *Qualitative and Quantitative Methods in Libraries* 2:4, 2013, S. 459–467. URL: <http://www.qqml-journal.net/index.php/qqml/article/view/122/122>.
- [Lib20a] J. C. K. Library. *JCKL Statistics Dashboard - James C. Kirkpatrick Library - University of Central Missouri*. 2020. URL: <https://library.ucmo.edu/stats/dashboard> (besucht am 28. 09. 2020).
- [Lib20b] E. Libris. *Alma Analytics - Data-Rich & Actionable Decision Support | Ex Libris*. 2020. URL: <https://www.exlibrisgroup.com/products/alma-library-services-platform/alma-analytics/> (besucht am 28. 09. 2020).
- [Lin16] M. Linden. *Geschäftsmodellbasierte Unternehmenssteuerung mit Business-Intelligence-Technologien : Unternehmensmodell - Architekturmodell - Datenmodell*. Springer Gabler, Wiesbaden, 2016. xxiv, 403 Seiten.
- [Lou21] M. Loukides. *Where Programming, Ops, AI, and the Cloud are headed in 2021*. O'Reilly Media. 2021. URL: <https://www.oreilly.com/radar/where-programming-ops-ai-and-the-cloud-are-headed-in-2021/> (besucht am 26. 01. 2021).
- [Lyo10] L. E. Lyons. „Collection evaluation : selecting the right tools and methods for your library“. In: *Library data : empowering practice and persuasion*. Hrsg. von D. Orcutt. Libraries Unlimited, Santa Barbara, Calif., 2010, S. 37–51.

- [MB00] H. Mucksch und W. Behme, Hrsg. *Das Data Warehouse-Konzept : Architektur - Datenmodelle - Anwendungen : mit Erfahrungsberichten*. 4., vollständig überarbeitete und erweiterte Auflage. Gabler, Wiesbaden, 2000. xix, 541 Seiten.
- [Mey18] A. Meyer. „Using R and the Tidyverse to generate library usage reports“. *Code4Lib* 39, 2018. URL: <https://journal.code4lib.org/articles/13282> (besucht am 30. 08. 2020).
- [MH12] E. Morton-Owens und K. Hanson. „Trends at a glance: a management dashboard of library statistics“. *Information Technology and Libraries* 31, 2012. DOI: [10.6017/ital.v31i3.1919](https://doi.org/10.6017/ital.v31i3.1919). (Besucht am 30. 08. 2020).
- [Mic20] Microsoft. *Datenvisualisierung*. 2020. URL: <https://powerbi.microsoft.com/de-de/> (besucht am 28. 09. 2020).
- [ML13] R. M. Müller und H.-J. Lenz. *Business Intelligence*. Springer, Berlin, 2013. xxii, 306 Seiten.
- [Mor15] M. Moravetz-Kuhlmann. „Erwerbungspolitik, Etatplanung und Mittelallokation in wissenschaftlichen Bibliotheken“. In: *Praxis Handbuch Bibliotheksmanagement*. Hrsg. von R. Griebel, H. Schäffler, und K. Söllner. Bd. 1. De Gruyter, Berlin, 2015, S. 161–183.
- [Mur13] S. A. Murphy. „Data visualization and rapid analytics: applying Tableau Desktop to support library decision-making“. *Journal of Web Librarianship* 7:4, 2013, S. 465–476. DOI: [10.1080/19322909.2013.825148](https://doi.org/10.1080/19322909.2013.825148).
- [OCL20] OCLC. *BibControl: Statistik und Reporting für Bibliotheken*. 2020. URL: <https://www.oclc.org/de/bibcontrol.html> (besucht am 28. 09. 2020).
- [Ous18] J. K. Ousterhout. *A philosophy of software design*. First Edition. Yaknyam Press, Palo Alto, CA, 2018. x, 178 Seiten.

- [Pan21] Pandas. *Pandas - Python Data Analysis Library*. Pandas. 2021. URL: <https://pandas.pydata.org/> (besucht am 26. 01. 2021).
- [Phe12] E. Phetteplace. „Effectively visualizing library data“. *Reference & User Services Quarterly* 52:2, 2012, S. 93–97. DOI: [10.5860/rusq.52n2.93](https://doi.org/10.5860/rusq.52n2.93). (Besucht am 30. 08. 2020).
- [Plo21a] Plotly. *Dash callbacks*. 2021. URL: <https://dash.plotly.com/basic-callbacks> (besucht am 27. 01. 2021).
- [Plo21b] Plotly. *Dash core components*. 2021. URL: <https://dash.plotly.com/dash-core-components> (besucht am 27. 01. 2021).
- [Plo21c] Plotly. *Dash documentation & user guide*. 2021. URL: <https://dash.plotly.com/introduction> (besucht am 27. 01. 2021).
- [Plo21d] Plotly. *Dash html components*. 2021. URL: <https://dash.plotly.com/dash-html-components> (besucht am 27. 01. 2021).
- [Plo21e] Plotly. *Discrete Colors : Python*. 2021. URL: <https://plotly.com/python/discrete-color/> (besucht am 02. 02. 2021).
- [Plo21f] Plotly. *Plotly open source graphing libraries*. 2021. URL: <https://plotly.com/graphing-libraries/> (besucht am 27. 01. 2021).
- [Plo21g] Plotly. *Plotly.graph\_objects.Bar : 4.14.3 documentation*. 2021. URL: [https://plotly.com/python-api-reference/generated/plotly.graph\\_objects.Bar.html](https://plotly.com/python-api-reference/generated/plotly.graph_objects.Bar.html) (besucht am 01. 02. 2021).
- [Plo21h] Plotly. *URL routing and multiple apps*. 2021. URL: <https://dash.plotly.com/urls> (besucht am 29. 01. 2021).
- [Pyt21a] Python. *6. Modules — Python 3.7.9 documentation*. 2021. URL: <https://docs.python.org/3.7/tutorial/modules.html> (besucht am 26. 01. 2021).

- [Pyt21b] Python. *PyPI : the Python package index*. 2021. URL: <https://pypi.org/> (besucht am 26. 01. 2021).
- [Pyt21c] Python. *Welcome to Python.org : the official home of the Python Programming Language*. Welcome to Python.org. 2021. URL: <https://www.python.org/> (besucht am 26. 01. 2021).
- [Rös+19] H. Rösch, J. Seefeldt, K. Umlauf, und P. Engelbert, Hrsg. *Bibliotheken und Informationsgesellschaft in Deutschland : eine Einführung*. 3., neu konzipierte und aktualisierte Auflage. Harrassowitz Verlag, Wiesbaden, 2019. xiii, 329 Seiten.
- [RVK21] RVK. *RVK Download - Regensburger Verbundklassifikation Online*. 2021. URL: <https://rvk.uni-regensburg.de/regensburger-verbundklassifikation-online/rvk-download> (besucht am 02. 02. 2021).
- [SAP20] SAP. *Pixel-perfect report creation, data analysis and report distribution*. 2020. URL: <https://www.crystalreports.com/> (besucht am 28. 09. 2020).
- [SB08] R. M. Schmidt und B. Bauer. „Deutsche Bibliotheksstatistik (DBS): Konzept, Umsetzung und Perspektiven für eine umfassende Datenbasis zum Bibliothekswesen in Deutschland: 10 Fragen von Bruno Bauer an Ronald M. Schmidt, Leiter der DBS“. *GMS Medizin - Bibliothek - Informatio* 8:1, 2008, S. 1–7. URL: <http://www.egms.de/en/journals/mbi/2008-8/mbi000102.shtml> (besucht am 08. 09. 2020).
- [Sof20] T. Software. *Software für Business Intelligence und Analytics*. 2020. URL: <https://www.tableau.com/de-de> (besucht am 28. 09. 2020).
- [Spi17] E. T. Spielberg. „Der FachRef-Assistent : personalisiertes, fachspezifisches und transparentes Bestandsmanagement“. Master Thesis. 2017. xiii, 107 Seiten. URL: <http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:hbz:79pbc-opus-9888> (besucht am 30. 08. 2020).

- [Spr20] SpringShare. *LibInsight - analyze library services and make more informed service decisions*. 2020. URL: <https://springshare.com/libinsight/> (besucht am 28. 09. 2020).
- [Tuf19] E. R. Tufte. *The visual display of quantitative information*. 2. Aufl. Graphics Press, Cheshire, CT, USA, 2019, 197 Seiten.
- [Van21] B. Van de Ven. *Bokeh*. 2021. URL: <https://bokeh.org/> (besucht am 27. 01. 2021).
- [WH13] L. K. Wiegand und B. Humphrey. „Visualizing library statistics using Open Flash Chart 2 and Drupal“. *Code4Lib* 19, 2013. URL: <https://journal.code4lib.org/articles/7812> (besucht am 30. 08. 2020).
- [WK20] A. Witherley und A. Kirk. *The Chartmaker directory*. 2020. URL: <https://chartmaker.visualisingdata.com/> (besucht am 28. 09. 2020).





# 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.

---

Ort, Datum

---

Unterschrift